

## Implementação eficiente para auxiliar a construção do sistema matricial do MEFE

**Romulo P. Lucchini\***

**Werley G. Facco**

**Eduardo da Silva**

Instituto Federal do Espírito Santo, IFES  
29932-540, São Mateus, ES, Brasil

**Alex S. Moura**

Departamento de Economia, Universidade Federal de Juiz de Fora  
35010-177, Governador Valadares, MG, Brasil

### **RESUMO**

O Método dos Elementos Finitos Estendidos (MEFE) possui extensa aplicabilidade na modelagem e na solução de problemas eletromagnéticos com características especiais localizadas [1]. Tais características podem aparecer no domínio computacional do problema, o que altera a configuração da malha que cobre o domínio computacional.

O objetivo desse trabalho é apresentar uma implementação eficiente capaz de fornecer todas as informações a respeito dos elementos, arestas e nós da nova malha de elementos finitos gerada a partir da interface entre os diferentes meios.

**Palavras chave:** *Método dos Elementos Finitos Estendidos, Estrutura de Dados, Singularidade*

### **INTRODUÇÃO**

O MEFE caracteriza-se por permitir o enriquecimento local no espaço de aproximação da solução que envolve saltos, quinas ou singularidades em geral. O enriquecimento é realizado através do Método da Partição da Unidade [1]. O método é usado para a aproximação de soluções com singularidades em partes do domínio computacional do problema.

Levando em consideração que o MEFE faz uso de malhas de elementos finitos e que para uma classe de problemas estas malhas são cortadas por curvas que representam singularidades físicas. Neste trabalho, pretende-se por meio de estruturas de dados, ordenação e buscas binárias, melhorar o desempenho do MEFE, principalmente no que diz respeito ao Pré-processamento das informações da nova malha, que agora possui elementos que foram cortados pela singularidade.

A proposta é implementar um algoritmo que faça uso de estruturas de dados específicas, capaz de gerar todas as informações relacionadas aos dados da nova malha de elementos finitos.

Na próxima seção apresentamos a técnica utilizada para definir as informações dos nós e elementos da malha do MEF que é interceptada pela singularidade. Em seguida apresentamos os resultados obtidos com a técnica proposta por este trabalho, onde foi utilizada a formulação do problema exposto em [3].

### **PRÉ-PROCESSAMENTO DE DADOS**

Os dados de entrada serão duas matrizes, uma matriz **A** com informações dos elementos e uma matriz **B** com informações dos nós da malha de elementos finitos, respectivamente. Com o pré-processamento, pretende-se receber as matrizes dadas pelo malhador e retornar novas matrizes com informações a respeito dos elementos que foram cortados pela interface e dos novos nós. Essa tarefa vai se tornando complexa à medida que a malha vai sendo refinada. Para facilitar nosso trabalho, em nosso pré-processamento, as informações da matriz **A** e **B** serão alocadas em uma matriz **C**.

Uma função geral foi criada, chamada “interface”, essa função recebe como parâmetro uma coordenada “x” e retorna o valor da coordenada “y”, obtendo assim um ponto em função de x e

\* Bolsista de Iniciação Científica PIBIT/CNPq

y. Portanto, com uma comparação fácil entre as coordenadas da interface e as coordenadas dos nós, podemos descobrir se um determinado nó está acima ou abaixo da interface do problema. Com essas informações conseguimos descobrir quais elementos foram cortados ou não pela interface. Para calcular o valor das coordenadas de interseção, foi utilizada a função “seg2poly” do MatLab®. Essa função recebe como parâmetros duas matrizes, a primeira contém as coordenadas cartesianas do ponto de início e fim da interface, já a segunda contém as coordenadas cartesianas dos três nós do elemento analisado. Essa função retorna uma matriz com as coordenadas dos dois pontos de interseção que serão adicionados a uma matriz que vai conter as coordenadas de todos os novos nós. Para facilitar o trabalho, criamos uma função de ordenação da matriz, que irá ordenar de acordo com a coordenada “x”. O método implementado foi o ShellSort, que foi escolhido devido a sua fácil implementação e sua eficiência na ordenação comparado a alguns outros métodos [2].

O método ShellSort é derivado do método de inserção e é baseado na diminuição dos incrementos, por causa disso, cada etapa na ordenação envolve a comparação de alguns elementos já um pouco ordenados, por isso a ordenação é eficiente. O tempo de execução desse método é proporcional ao número de elementos pela relação:  $t = O(n^{1.2})$  [3]. Um exemplo do funcionamento do método ShellSort é ilustrado pela Fig 1.

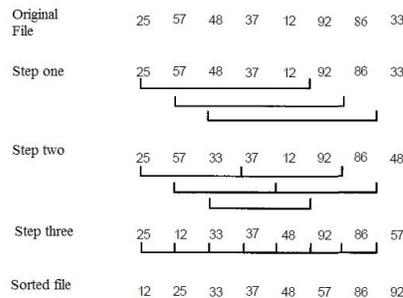


Figura 1. Como o método ShellSort opera para um arquivo qualquer [2].

O pré-processamento proposto neste trabalho foi confrontado com uma implementação que não faz uso de estruturas de dados.

### **RESULTADOS**

Para verificação, foi considerado um domínio unitário quadrado 2D, onde a tensão de 1V foi aplicada na placa superior e a placa inferior foi aterrada. O domínio foi coberto por uma malha triangular, com doze elementos, Fig. 2.

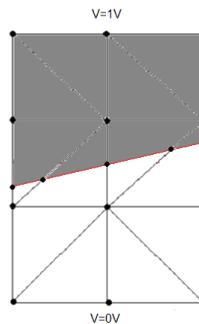


Figura 2. Modelo 2D de uma malha de doze elementos cortada por uma interface oblíqua.

Em todos os testes foram usadas a permissividade elétrica do meio superior igual  $\varepsilon_r = 10$  e a permissividade do meio inferior igual  $\varepsilon_r = 1$ . Com o uso das funções de forma triangulares e quadrangulares definidas em [4], o potencial elétrico foi calculado para o caso em que a interface não é paralela às placas do capacitor. As intercessões entre a malha de elementos finitos e a interface entre os dois meios é feita usando pré-processamento descrito na seção II, e o resultado do potencial calculado é demonstrado na Fig. 3.

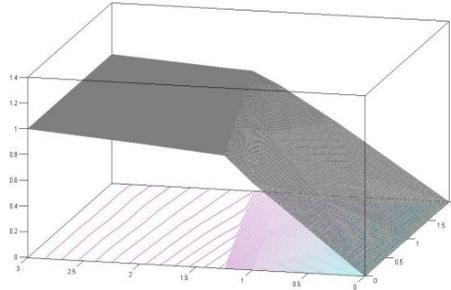


Figura 3. Potencial eletrostático calculado através do MEF.

Ambos pré-processamentos foram analisados em um computador com 8GB de memória RAM, 750GB de HD, processador Intel® Core™ i7, 2ª geração, 2.20 GHz, conseguimos os seguintes tempos de processamento:

- O tempo de processamento sem estruturas adequadas de dados para uma malha, Fig. 2 foi de 0,0624 segundos;
- Com o pré-processamento proposto neste trabalho o tempo de processamento, usando uma malha com **162 elementos triangulares** foi de 0,0468 segundos.

### CONCLUSÕES

Este trabalho possui como principal contribuição um pré-processamento eficiente dos dados na tarefa de construir as matrizes necessárias para o desenvolvimento do MEF. Como esperado, usando estruturas de dados é possível diminuir de maneira considerável o custo computacional. Com os métodos de ordenação, as comparações de informações se tornam muito rápidas, ou seja, o número de comparações é reduzido significativamente.

### AGRADECIMENTOS

Esse trabalho possui suporte pela FAPES, FAPEMIG, CAPES, e Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, Brasil.

### **Referências:**

- [1] I. Babuška, U. Banerjee, and J. E. Osborn. Generalized finite element methods: Main ideas, results and perspective. *Technical report, TICAM*, 2004.
- [2] Schildt, H. *C completo e total*. 3 ed, São Paulo: MAKRON Books, 1996, p. 511-523.
- [3] Tenenbaum, Aaron M., Langsam, Y. Moshe, J. A. *Estruturas de dados usando C*. São Paulo: MAKRON Books, 1995.
- [4] V. Rochus, L. V. Miegroet, D. J. Rixem, P. Duysinx. Electrostatic simulation using XFEM for conductor and dielectric interfaces, *International Journal for Numerical Methods in Engineering*, volume 85, Issue 10, 2011, pages 1207–1226.