

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Estudo da performance numérica em problemas da mecânica dos sólidos utilizando o Método dos Elementos de Contorno

Carlos A. Reyna Vera-Tudela¹

Departamento de Matemática, PPG em Modelagem Matemática e Computacional, UFRRJ, Seropédica, RJ

Ronilson R. Pinho²

CEFET, Unidade Maria da Graça, Rio de Janeiro, RJ

Resumo. O Método dos Elementos de Contorno (MEC) é um método numérico computacional para a solução de sistemas de equações diferenciais, formuladas em forma integral. Pode ser aplicado em todas as áreas das ciências e engenharias. Os autores vêm desenvolvendo aplicações na área da mecânica dos sólidos, mecânica da fratura linear elástica em problemas estáticos e dinâmicos. A linguagem de programação preferencialmente utilizada pelos pesquisadores foi Fortran. Existem muitos dos códigos computacionais disponíveis escritos nesta linguagem, os mesmos não tem foco na eficiência computacional, ou seja os resultados são numericamente eficientes, mas computacionalmente deixam muito a desejar. Com o desenvolvimento dos recursos computacionais, são disponibilizadas novas ferramentas que ajudam a analisar os problemas além dos resultados numéricos. Neste trabalho é estudado um programa sequencial e estruturado, desenvolvido em linguagem de programação Fortran. São implementados códigos de paralelização com OpenMP e são realizados testes em computadores pessoais com processadores Pentium dual-core, i3 e i5. O sistema operacional Windows 7 na sua versão Professional 32 bits, e o compilador Fortran da Intel, na versão 11. São realizadas medições dos tempos de processamento e os resultados são apresentados em forma de tabelas.

Palavras-chave. Método dos Elementos de Contorno, Métodos Numéricos, Programação Paralela.

1 Introdução

Nas últimas décadas diversas técnicas numéricas foram desenvolvidas para resolver problemas das ciências e engenharias, especificamente problemas estáticos e dinâmicos da mecânica dos sólidos. O artigo publicado por Cheng e Cheng [1] apresenta uma pesquisa realizada usando o site *Web of Science* onde mostra que os três métodos numéricos mais referenciados são o Método dos Elementos Finitos, o Método das Diferenças Finitas e o Método dos Elementos de Contorno, nessa ordem. O método mais referenciado com 66237

¹candres@ufrj.br

²ronilsonpinho@gmail.com

entradas é o Método dos Elementos Finitos [2]. Segundo colocado nesta pesquisa, com 19531 entradas aparece o Método das Diferenças Finitas [3] e em terceira posição com 10126 entradas o Método dos Elementos de Contorno [4].

Nas cinco últimas décadas, o MEC vem ganhando muito destaque tanto em trabalhos de pesquisa pura quanto em simulações de engenharia. O MEC possui uma grande vantagem com relação aos demais métodos numéricos em análises de modelos com domínio infinito ou semi-infinito, pela simplicidade de discretizado do modelo. Além disto, o MEC apresenta resultados com ótima precisão em análise de problemas com altos gradientes, com concentração de tensões e em problemas de Mecânica da Fratura, por exemplo [5].

O MEC, consiste, basicamente, na transformação da equação diferencial que governa o problema em uma equação integral relacionando somente valores das variáveis no contorno que é discretizado em elementos. A partir desta discretização, as integrais de contorno são aproximadas por integrais efetuadas em todos os elementos possibilitando o conhecimento das incógnitas em qualquer ponto do contorno da região de interesse. Os sistemas de equações são menores quando comparados com os outros métodos mencionados. Para cálculo de valores no domínio é realizado em um pós-processamento utilizando apenas os valores calculados para o contorno.

Os microcomputadores pessoais têm, hoje em dia, incorporados a eles o poder de processamento das máquinas de grande porte por meio da utilização de múltiplos processadores, trabalhando sobre o mesmo programa ou executando programas de diferentes usuários ao mesmo tempo. Baseado nesta premissa, os autores realizaram uma experiência para medir os tempos de processamento computacional para três configurações diferentes de processadores.

2 O Método dos Elementos de Contorno

As condições de equilíbrio para um problema elastostático com carga de domínio nula é representado pela conhecida Equação de Navier:

$$\mu u_{j,ii} + (\lambda + \mu) u_{i,ij} = 0, \quad (1)$$

onde u representa o campo de deslocamentos; μ e λ são as constantes de Lamê. A relação entre o tensor de deslocamentos e o tensor de tensões é dado pela expressão seguinte,

$$p_i = \sigma_{ij} n_j, \quad (2)$$

onde n_j é normal externa à superfície. A formulação tradicional do Método dos Elementos de Contorno consiste em ponderar a equação (1) por uma função de ponderação u^* , com características especiais e depois integrá-la no domínio. Por meio de um tratamento matemático adequado, que envolve integração por partes e tomando em consideração o princípio de reciprocidade, transforma-se esta equação integral de domínio em uma equação integral de contorno.

$$C_{ij}(\xi) u_j(\xi) + \int_{\Gamma} p_{ij}^*(\xi, x) u_j(x) d\Gamma(x) = \int_{\Gamma} u_{ij}^*(\xi, x) p_j(x) d\Gamma(x). \quad (3)$$

A solução fundamental para o problema de estado plano num médio elástico infinito é dado pela expressão seguinte,

$$u_{ij}^*(\xi, x) = \frac{-1}{8\pi(1-\nu)} \{ (3-4\nu) \ln(r) \delta_{ij} - r_{,i} r_{,j} \} \quad (4)$$

e

$$p_{ij}^*(\xi, x) = \frac{-1}{4\pi(1-\nu)r} \{ [(1-2\nu)\delta_{ij} + 2r_{,i} r_{,j}] \frac{\partial r}{\partial n} - (1-2\nu)(r_{,i} n_j - r_{,j} n_i) \} \quad (5)$$

onde r é a distância do ponto de aplicação da carga ao ponto em consideração [4]. A Solução Fundamental é um tipo de função utilizada para resolver equações diferenciais não-homogêneas sujeitas a condições iniciais ou condições de contorno determinadas, e descreve a resposta do sistema físico à função delta de Dirac, que representa um impulso aplicado no ponto x (magnitude unitária).

3 Implementação Numérica

Uma vez obtida a equação integral de contorno é preciso discretizá-la para então resolvê-la [6]. Para a discretização da equação (3) divide-se o contorno em J elementos Γ_j . A esses elementos estão associados valores nodais de deslocamentos e forças de superfície, em função dos quais a variação destas grandezas dentro do elemento depende das funções de interpolação N utilizadas e que por sua vez dependem do número de nós funcionais n dos elementos. Desta forma para deslocamentos tem-se a forma matricial seguinte:

$$u^{(j)} = Nu^{(n)}, \quad (6)$$

e para forças de superfície:

$$p^{(j)} = Np^{(n)}, \quad (7)$$

substituindo as equações (6) e (7) na equação (3) tem-se,

$$Cu^{(n)} = \sum_{j=1}^J \left(\int_{\Gamma_j} u^* Nd\Gamma \right) p^{(n)} - \sum_{j=1}^J \left(\int_{\Gamma_j} p^* Nd\Gamma \right) u^{(n)}. \quad (8)$$

O elemento de contorno Γ_j varia de acordo com a função de interpolação adotada, neste trabalho é considerada a função de interpolação linear com dois nós geométricos por elemento.

Durante a montagem do sistema indicado pela equação (8), cada uma das integrais será calculada numericamente sempre que o ponto fonte ξ não pertencer ao elementos Γ_j sobre o qual está sendo efetuada a integração. Este cálculo se dará através da integração numérica de Gauss dada por,

$$\int_{-1}^1 f(\eta) d\eta = \sum_{k=1}^{NPI} f(\eta_k) w_k, \quad (9)$$

onde η_k é a coordenada adimensional do k -ésimo ponto de integração, w_k é o fator de peso associado ao ponto k e NPI é o número total de pontos de integração utilizado. Desta forma trabalha-se com as parcelas da equação (8) como segue,

$$\int_{\Gamma_j} p^* N d\Gamma = \int_{\Gamma_j} p^* N |J| d\eta \cong \sum_{k=1}^{NPI} |J|_k w_k N_k p_k^*, \quad (10)$$

e

$$\int_{\Gamma_j} u^* N d\Gamma = \int_{\Gamma_j} u^* N |J| d\eta \cong \sum_{k=1}^{NPI} |J|_k w_k N_k u_k^*. \quad (11)$$

A equação integral discretizada é aplicada repetidamente considerando o ponto fonte ξ situado coincidentemente com todos os pontos nodais existentes. Um sistema de $2n$ equações algébricas é gerado e envolve $2n$ valores nodais de deslocamento e força de superfície. Ainda é necessário levar este sistema para uma forma matricial e para isso coloca-se da forma a seguir,

$$\sum_{j=1}^J \left(\int_{\Gamma_j} p^* N d\Gamma \right) u^{(n)} = \sum_{j=1}^J h_j u^{(n)}. \quad (12)$$

Similarmente

$$\sum_{j=1}^J \left(\int_{\Gamma_j} u^* N d\Gamma \right) p^{(n)} = \sum_{j=1}^J g_j p^{(n)}. \quad (13)$$

O sistema fica reduzido na forma a seguir,

$$C u^{(n)} + \sum_{j=1}^J h_j u^{(n)} = \sum_{j=1}^J g_j p^{(n)}. \quad (14)$$

Resulta, então, um sistema de equações matriciais na forma

$$(C + \hat{H})u = Gp, \quad (15)$$

onde os vetores u e p contêm os valores de deslocamentos e forças de superfície em todos os pontos nodais. A matriz C é de banda pequena e pode ser incorporada a \hat{H} para formar,

$$Hu = Gp. \quad (16)$$

Após a introdução das condições de contorno, a equação (16) pode ser reescrita como,

$$Ay = b, \quad (17)$$

onde y representa os valores nodais incógnitos e b as contribuições dos valores prescritos no contorno.

4 Testes e Resultados

Para realizar os testes estuda-se o problema de uma barra engastada e tracionada como pode ser observado na figura (1). O método de resolução das equações diferenciais parciais é o Método dos Elementos de Contorno. São utilizados 36 elementos de contorno lineares, que correspondem a 76 nós de contorno e são considerados nós duplos nos quatro cantos da barra. Também é considerado um nó interno. As propriedades físicas do problema são: Módulo de Young $E = 1$, Coeficiente de Poisson $\nu = 0.0$, densidade $\rho = 1$. Para a integração numérica são utilizados 12 pontos de Gauss. O carregamento tem valor unitário distribuído pela superfície lateral direita da barra.

Após a resolução matemática do problema é gerado um código computacional desenvol-



Figura 1: Representação física da barra engastada e tracionada.

vido em linguagem Fortran [7].

Este programa deve ser estudado para determinar, que partes do código possuem o maior tempo de processamento computacional. Para isto foi utilizada a ferramenta *Intel Parallel Studio XE* [8]. Desta forma foi identificado que a formação das matrizes H e G possuem o maior tempo de processamento computacional. Outras partes do código também apresentam esta característica como, por exemplo, aquele que resolve o sistema de equações lineares. Neste trabalho o foco é a formação das matrizes H e G por terem uma relevância maior quando comparado com os outros tempos medidos.

Para melhorar o tempo de processamento computacional foi utilizada a programação paralela através da ferramenta OpenMP [9]. OpenMP (do inglês Open Multi-Processing) é uma interface de programação (API), portátil, baseada no modelo de programação paralela de memória compartilhada para arquiteturas de múltiplos processadores. OpenMP está disponível para uso com os compiladores C/C++ e Fortran, podendo ser executado em ambientes Unix e Windows (Sistemas Multithreads).

Feitas todas as implementações, foram realizados três estudos em diferentes configurações de computadores pessoais, mas que em comum oferecem a opção de processamento multithreading. Estes estudos estão resumidos nas tabelas 1, 2 e 3. Os resultados das tabelas apresentam o tempo de processamento computacional em segundos.

Tabela 1: Resultado para Pentium Dual-Core: Cpu de 2 GHz com 3 Giga RAM, SO Windows 7

Número de Threads	Teste 1	Teste2	Teste 3	Teste 4
1	359	360	359	358
2	186	185	186	186
3	264	263	263	264
4	235	239	235	238
5	260	260	259	261
6	247	242	246	247

Tabela 2: Resultado para Intel Core i3: Cpu de 1.47 GHz com 3 Giga RAM, SO Windows 7

Número de Threads	Teste 1	Teste2	Teste 3	Teste 4
1	420	418	420	418
2	246	238	236	228
3	199	196	197	196
4	231	186	182	182
5	267	272	274	268
6	296	328	278	312

Tabela 3: Resultado para Intel Core i5: Cpu de 3.4 GHz com 8 Giga RAM, SO Windows 7

Número de Threads	Teste 1	Teste2	Teste 3	Teste 4
1	167	167	165	164
2	86	87	84	84
3	60	60	60	60
4	46	46	45	45
5	65	66	65	66
6	66	70	73	66

5 Conclusões

Uma das primeiras conclusões que pode-se observar, nos estudos realizados, é o fato de que o aumento no número de threads não necessariamente implica em redução no tempo de processamento computacional. Constata-se que o tempo diminui a partir do momento em que o número de threads seja igual ao número de núcleos do processador, quando temos um número maior de threads que o número de núcleos no processador, o tempo da aplicação tende a aumentar, isso em função do escalonamento de threads.

Para os resultados com o processador Dual-Core pode-se observar que o melhor resultado

acontece com o uso de dois threads o que está em concordância com o fato deste processador possuir tecnologia de dois núcleos. O melhor resultado é para a configuração com processador Intel Core i5, com uma redução de aproximadamente 75% quando comparado com o processador Intel Core i3. Testes para o processador Intel Core i7 ainda são necessários.

Outras partes do programa também devem ser paralelizadas, o que deve melhorar ainda mais os resultados obtidos.

Para analisar o comportamento do exemplo apresentado seria interessante fazer outro estudo variando a discretização do contorno aumentando o número de elementos de contorno, variando o número de pontos de integração e até variando o número de pontos internos. Aumentando todos estes valores, o tempo de processamento computacional, também, aumenta.

Finalmente, deseja-se deixar registrado que este trabalho não pretende analisar, sob todos os rigores da teoria, um problema de computação de alto desempenho.

Referências

- [1] G. L. Diniz, J. F. C. A. Meyer e L. C. Barros, Solução numérica para um problema de Cauchy Fuzzy que modela o decaimento radioativo, *TEMA*, 23:63–72, 2001. DOI:10.1007/s40314-014-0163-6.
- [2] K. J. Bathe. *Finite Element Procedures*. Prentice Hall, USA, 2016.
- [3] G. E. Forsythe and W. R. Wasow. *Finite-Difference Methods for Partial Differential Equations*. Dover Publications, USA, 2004.
- [4] C. A. Brebbia, J. C. F. Telles and L. C. Wrobel. *Boundary Elements Techniques: Theory and Applications in Engineering*. Springer-Verlag, London, 1984.
- [5] C. A. R. Vera-Tudela and J. C. F. Telles, The dual reciprocity method and the numerical Green's function for BEM fracture mechanic problems, *Acta Mechanica*, 2016. DOI: 10.1007/s00707-015-1530-0.
- [6] C. A. R. Vera-Tudela, Formulações Alternativas do MEC para Problemas Elastodinâmicos de Mecânica da Fratura com o uso da Função de Green Numérica, Tese de Doutorado, COPPE-UFRRJ, 2003.
- [7] S. J. Chapman. *Fortran 95/2003 for Scientists and Engineers*. McGraw-Hill, USA, 2007.
- [8] S. Blair-Chappell and A. Stokes. *Parallel Programming with Intel Parallel Studio XE*. John Wiley and Sons, USA, 2012.
- [9] B. Chapman, G. Jost and R. Van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. MIT Press, USA, 2007.