Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 3063-3075 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

UMA ESTRATÉGIA *MULTITHREADED* PARA O ACOPLAMENTO ENTRE O MÉTODO DOS ELEMENTOS FINITOS E DE CONTORNO

Edivaldo Figueiredo Fontes Júnior[a], José Antonio Fontes Santiago[a], José Claudio de Faria Telles[a] e Carlos Andrés Reyna Vera-Tudela[b]

[a]*Programa de Engenharia Civil, COPPE / Universidade Federal do Rio de Janeiro, CP 68506, CEP 21945-970, Rio de Janeiro - RJ - Brasil, fontesjunior@coc.ufrj.br*

[b]*Departamento de Matemática, Universidade Federal Rural do Rio de Janeiro, Caixa Postal 74517, CEP 23890-971, Seropédica, RJ, Brasil, candres@ufrrj.br*

Palavras Chave: *Multithreaded*, Elementos de Contorno, Elementos Finitos.

Resumo. O Método dos Elementos de Contorno (MEC) e o Método dos Elementos Finitos (MEF) são métodos numéricos reconhecidamente eficientes e largamente utilizados para resolver problemas nas mais diversas áreas do conhecimento. Neste trabalho é apresentado resumidamente uma formulação para o acoplamento entre o MEC e o MEF. Um esquema de programação *multithreaded* é utilizado para acelerar o tempo na montagem das matrizes do MEC e da solução do sistema final de equações. As melhores características das matrizes do MEF também são exploradas. Tais implementações paralelas são voltadas para computadores pessoais equipados com processadores *multi*-core, de forma a se obter o máximo de desempenho dos núcleos do processador. O nível de paralelização é obtido via programação *multithreading* através da plataforma de desenvolvimento Java. A formulação é desenvolvida para problemas da elasticidade linear bidimensionais, porém as técnicas aqui empregadas podem ser extendidas, com pequenas modificações, a outros tipos de aplicações numéricas onde é vantajoso o acoplamento entre o MEC e o MEF.

1 INTRODUÇÃO

A procura por ferramentas computacionais de alto desempenho, que apresentem uma interface amigável e a um custo reduzido, está nos desafios de muitos pesquisadores; mas por outro lado está a necessidade de muitos outros pesquisadores e/ou empresas que precisam realizar cálculos em diversos problemas das ciências e engenharia. As ferramentas comumente encontradas no mercado têm um custo elevado e fica inviável sua utilização em pesquisas e nas pequenas e médias empresas. Visando está necessidade é que os autores vêm trabalhando em um projeto ambicioso e de longo prazo com o objetivo de preencher este vazio e oferecer uma ferramenta computacional que permita resolver problemas físicos, com um ambiente de trabalho eficiente e amigável que permita a visualização científica dos resultados.

O acoplamento entre diferentes métodos numéricos, aplicados a diversas áreas de conhecimento, vem ganhando grande interesse por diversos pesquisadores, devido a possibilidade de se aproveitar as vantagens associadas a cada método, tornando o problema acoplado mais preciso numericamente e fácil de ser modelado. Como grande foco das pesquisas, está o acoplamento entre o Método dos Elementos de Contorno (MEC) e o Método dos Elementos Finitos (MEF). Existem inúmeras atrações para a realização deste acoplamento, desde a redução da malha do problema até os elevados graus de precisão numérica que podem ser obtidos.

O uso de arquiteturas de processadores com crescentes números de núcleos de processamento vem se tornando um padrão para equipar tanto computadores pessoais (PC's) como os mais modernos *clusters* de PC's. Tais processadores são chamados processadores *multi-core*. A grande disseminação desses processadores multi-núcleo observada recentemente justifica os esforços em se escrever novos programas ou reescrever programas existentes para estes tipos de arquiteturas, bem como a pesquisa de novos algoritmos que utilizem de forma eficiente as características de processamento destes processadores.

Portanto neste trabalho são apresentados alguns resultados obtidos com o desenvolvimento de uma estratégia de programação *multithreading* (Carver e Tai, 2006), para minimizar o gasto de tempo de processamento na análise numérica realizada com o acoplamento entre o MEC e o MEF (MEC-MEF) dentro do ambiente computacional gráfico interativo MEMEC (Fontes Júnior et al., 2008; Vera-Tudela et al., 2009), desenvolvido para simulação numérica de problemas de Elasticidade Linear através do Método dos Elementos de Contorno e Visualização Científica. A eficiência computacional ao se utilizar a programação *multithreading* na implementação de métodos numéricos em geral, é considerável quando se utiliza um processador com mais de um núcleo de processamento, estes que por sua vez estão ficando a cada vez mais populares.

2 ACOPLAMENTO ENTRE O MEC E O MEF

O desenvolvimento do acoplamento MEC-MEF tem como objetivo aliar os principais pontos fortes de cada método. Em muitas ocasiões é de grande interesse prático resolver parte do problema utilizando o MEC e outra parte utilizando o MEF. Por exemplo problemas onde existem singularidades, tornando muito interessante o uso do MEC na região onde são encontradas as maiores singularidades. Outros casos onde se é interessante a aplicação do acoplamento MEC-MEF são em problemas com região infinita ou semi-infinita, onde pode-se utilizar a capacidade do MEC em reduzir consideravelmente a malha que seria utilizada para discretizar a região infinita ou semi-infinita com o MEF e obtendo uma ótima precisão numérica, com pode ser visto nos trabalhos de Telles e Brebbia (1981) e Brebbia et al. (1984).

2.1 Formulação do MEC

Os problemas de Elasticidade aqui tratados são governados pela equação de equilíbrio de Navier, que pode ser escrita utilizando a notação cartesiana indicial para qualquer domínio Ω na forma:

$$G u_{j,kk} + \frac{G}{1-2\nu} u_{k,kj} + b_j = 0 \quad \text{em} \quad \Omega \tag{1}$$

sujeita as condições de contorno:

$$\begin{aligned} u &= \bar{u} \ \text{ em } \ \Gamma_2 \\ p &= \bar{p} \ \text{ em } \ \Gamma_1 \end{aligned} \tag{2}$$

onde u são os deslocamentos, p as forças de superfície, $\bar{u}$ e $\bar{p}$ são os valores prescritos no contorno $\Gamma = \Gamma_1 \cup \Gamma_2$, G é o módulo de cisalhamento, ν é o coeficiente de Poisson e b_j é a componente de forças de volume.

A dedução da formulação do MEC para a Eqs. (1) e (2), pode ser obtida via método dos resíduos ponderados (Brebbia et al., 1984). Dando origem a uma equação integral, que representa os deslocamentos no interior de um corpo, conhecida como Identidade de Somigliana:

$$u_i(\xi) = \int_\Gamma u^*_{ij}(\xi, x) p_j(x) d\Gamma - \int_\Gamma p^*_{ij}(\xi, x) u_j(x) d\Gamma \tag{3}$$

onde assume-se que $b_i = 0$ por simplicidade, u^*_{ij}, p^*_{ij} representam, os tensores da solução fundamental de Kelvin para o espaço infinito e $\xi \in \Omega$. O próximo passo é obter uma equação integral envolvendo somente variáveis sobre o contorno, isso pode ser feito avaliando-se o limite da Eq. (3) quando o ponto ξ tende a se aproximar do contorno Γ, resultando na equação:

$$c_{ij}(\xi) u_j(\xi) + \cancel{\int_\Gamma} p^*_{ij}(\xi, x) u_j(x) d\Gamma = \int_\Gamma u^*_{ij}(\xi, x) p_j(x) d\Gamma \tag{4}$$

onde o coeficiente c_{ij} é definido de acordo com a geometria do contorno no ponto $\xi \in \Gamma$ e a integral do lado esquerdo é calculada no sentido do valor principal de Cauchy.

A solução da equação integral de contorno, Eq. (4), é obtida discretizando-se o contorno do problema em NE elementos conexos, reescrevendo a Eq. (4) na seguinte forma:

$$c_{ij} u_j + \sum_{k=1}^{NE} \int_{\Gamma_k} p^*_{ij} u_j d\Gamma = \sum_{k=1}^{NE} \int_{\Gamma_k} u^*_{ij} p_j d\Gamma \tag{5}$$

Sobre cada elemento devemos descrever a geometria, os deslocamentos u_i e forças de superfície p_j por meio de funções de interpolação. Para o presente trabalho foram utilizadas funções de interpolação lineares. Portanto reescrevendo a Eq. (5) com a substituição das funções de interpolação obtemos:

$$c_i u_i + \sum_{k=1}^{NE} \mathbf{hu} = \sum_{k=1}^{NE} \mathbf{gp} \tag{6}$$

A Eq. (6) pode ser rearranjada e reescrita na forma matricial global como:

$$\mathbf{Hu} = \mathbf{Gp} \tag{7}$$

onde $\mathbf{H}$ e $\mathbf{G}$ são matrizes de influência obtidas pela integração sobre os elementos de contorno utilizando as soluções fundamentais. As matrizes $\mathbf{H}$ e $\mathbf{G}$ são da ordem de $(2N \times 2N)$ com N igual ao número de nós geométricos do problema discretizado. E $\mathbf{u}$ e $\mathbf{p}$ são vetores representando os deslocamentos e forças de superfície nodais respectivamente.

2.2 Desenvolvimento da Matriz de Rigidez de Elementos de Contorno

O desenvolvimento da matriz de rigidez de elementos de contorno tem como objetivo a sua combinação com a matriz de rigidez do MEF. Existem diversas formas para se analisar um problema utilizando um acoplamento do MEC com o MEF (Aour et al., 2007; Georgiou, 1981; de Paula, 1991; Zienkiewicz et al., 1977). O procedimento de acoplamento desenvolvido neste trabalho consiste em considerar a região discretizada com o MEC como um super-elemento finito (Fig. 1). Portanto deve-se transformar a região do MEC que fornece um sistema relacionando forças de superfície e deslocamentos nodais (Eq. (7)) em um sistema similar ao do MEF relacionando deslocamentos e forças nodais equivalentes (Eq. (14)).

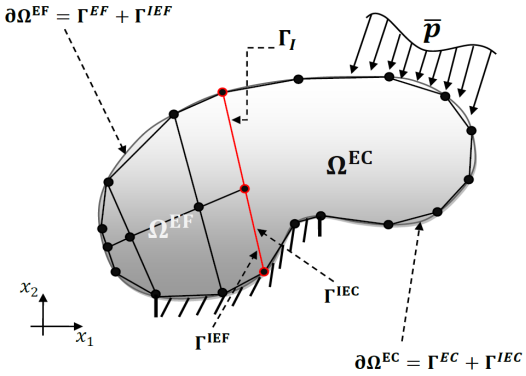

Figura 1: Discretização do problema acoplado.

Seja o sistema de Eqs. (7), multiplicando-o pela matriz inversa de $\mathbf{G}$ obtém-se:

$$\mathbf{G}^{-1}\,\mathbf{H}\,\mathbf{u} = \mathbf{p} \tag{8}$$

Agora deve-se transformar o vetor de forças de superfície $\mathbf{p}$ em um vetor de cargas nodais equivalente igual ao utilizado no MEF por meio de uma matriz de transformação $\mathbf{M}$. Considerando um elemento de contorno e, o vetor de cargas nodais equivalente $\mathbf{f}^{(e)}$ pode ser expresso em termos de um vetor de forças de superfície nodal $\mathbf{p}^{(e)}$ como:

$$\mathbf{f}^{(e)} = \mathbf{M}^{(e)}\mathbf{p}^{(e)} \tag{9}$$

onde $\mathbf{M}^{(e)}$ é a matriz de transformação que depende das funções de interpolação ψ_1 e ψ_2 do elemento linear aqui empregado:

$$\mathbf{M}^{(e)} = \begin{bmatrix} \int_{-1}^{1}\psi_1\psi_1 J\,d\xi & 0 & \int_{-1}^{1}\psi_1\psi_2 J\,d\xi & 0 \\ 0 & \int_{-1}^{1}\psi_1\psi_1 J\,d\xi & 0 & \int_{-1}^{1}\psi_1\psi_2 J\,d\xi \\ \int_{-1}^{1}\psi_2\psi_1 J\,d\xi & 0 & \int_{-1}^{1}\psi_2\psi_2 J\,d\xi & 0 \\ 0 & \int_{-1}^{1}\psi_2\psi_1 J\,d\xi & 0 & \int_{-1}^{1}\psi_2\psi_2 J\,d\xi \end{bmatrix} \tag{10}$$

(11)

Pré-multiplicando a Eq. (8) pela matriz $\mathbf{M}$, obtém-se:

$$\mathbf{M}\,\mathbf{G}^{-1}\,\mathbf{H}\,\mathbf{u} = \mathbf{M}\,\mathbf{p} = \mathbf{f}^{(EC)} \tag{12}$$

onde a matriz $\mathbf{M}$ é definida segundo um processo de *assembly* das matrizes locais $\mathbf{M}^{(e)}$ definidas pela Eq. (11). Comparando a Eq. (12) com a Eq. (14) pode-se definir a matriz de rigidez equivalente do Método dos Elementos de Contorno como:

$$\mathbf{K}^{(EC)} = \mathbf{M}\,\mathbf{G}^{-1}\,\mathbf{H} \tag{13}$$

A consequência ao se utilizar a Eq. (13) é que a região discretizada com o MEC pode ser tratada como um super-elemento finito e agora a matriz de rigidez equivalente do MEC pode ser acoplada da forma usual na matriz de rigidez do MEF, para a obtenção do sistema de equações global acoplado do problema discretizado com o MEC e o MEF.

2.3 Combinação do MEC com o MEF

A formulação do MEF para problemas correlatos aos aqui tratados com o MEC pode ser vista na referência (Hughes, 2000). Aqui é apresentado diretamente o sistema final de equações do MEF:

$$\mathbf{K}^{(EF)}\,\mathbf{u}^{(EF)} = \mathbf{f}^{(EF)} \tag{14}$$

onde $\mathbf{K}^{(EF)}$ é uma matriz simétrica positiva definida e esparsa, $\mathbf{u}^{(EF)}$ é o vetor de deslocamentos nodais e $\mathbf{f}^{(EF)}$ o vetor de forças nodais equivalentes.

Para a obtenção e uma combinação compatível entre o MEC e o MEF e que simule de forma correta as descontinuidades das forças de superfície na região do MEC, é necessário que a matriz de rigidez do MEC tenha somente um único grupo de equações para cada ponto nodal. Considerando a matriz de rigidez do MEC dada pela Eq. (13), é necessário a utilização da inversa da matriz $\mathbf{G}$. Quando o nó pertence a um contorno suave a matriz $\mathbf{G}$ é não singular e é garantida a existência de sua inversa. Porém ao se utilizar nós duplos nos pontos de descontinuidade geométrica do contorno, a matriz $\mathbf{G}$ torna-se singular. Aqui a matriz inversa de $\mathbf{G}$ é obtida diretamente, seja a matriz $\mathbf{G}$ singular ou não, pela técnica da pseudo inversa de Moore-Penrose (Demmel, 1997) a qual será utilizada neste trabalho.

Reescrevendo o sistema de Eqs. (12) com o auxílio da pseudo-inversa obtém-se

$$\mathbf{M}\,\mathbf{G}^{\dagger}\,\mathbf{H}\,\mathbf{u}^{(EC)} = \mathbf{f}^{(EC)} \tag{15}$$

onde $\mathbf{G}^{\dagger}$ é a pseudo inversa de Moore-Penrose. Portanto redefine-se a matriz de rigidez equivalente do MEC da Eq. (13) por:

$$\mathbf{K}^{(EC)} = \mathbf{M}\,\mathbf{G}^{\dagger}\,\mathbf{H} \tag{16}$$

De forma a obter um único grupo de equações para cada ponto nodal do contorno na Eq. (16) e fazer com que o vetor de forças nodais equivalentes do MEC considere o efeito acumulado das duas diferentes forças de superfície nos nós de descontinuidade as seguintes condições devem ser definidas respectivamente:

$$\mathbf{u}^{(EC)} = \mathbf{R}\hat{\mathbf{u}}^{(EC)} \tag{17}$$

e

$$\hat{\mathbf{f}}^{(EC)} = \mathbf{R}^{\mathrm{T}}\mathbf{f}^{(EC)} \tag{18}$$

onde $\hat{\mathbf{f}}^{(EC)}$ e $\hat{\mathbf{u}}^{(EC)}$ são os vetores que contém as forças de superfície e deslocamentos reduzidos respectivamente dos seus vetores originais e $\mathbf{R}$ é uma matriz de rotação definida esquematicamente dependendo da localização dos pontos nodais com descontinuidade na força de superfície.

Considerando o sistema de Eqs. (15), substituindo o vetor de deslocamentos $\mathbf{u}$ pela Eq. (17) e pré multiplicando-o por $\mathbf{R}^{\mathrm{T}}$ obtém-se:

$$\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{G}^{\dagger}\mathbf{H}\mathbf{R}\,\hat{\mathbf{u}}^{(EC)} = \mathbf{R}^{\mathrm{T}}\mathbf{f}^{(EC)} \tag{19}$$

e de forma compacta:

$$\hat{\mathbf{K}}^{(EC)}\,\hat{\mathbf{u}}^{(EC)} = \hat{\mathbf{f}}^{(EC)} \tag{20}$$

Assumindo que os sistemas de Eqs. (14) e (20) do MEF e do MEC respectivamente podem ser particionados de tal forma que os nós da interface MEC-MEF sejam separados dos outros nós correspondentes à região do MEF e MEC (Fig. 1). Pode-se reescrever os sistemas de Equações para o MEF e o MEC respectivamente, na forma:

$$\begin{bmatrix} \mathbf{K}^{(EF)} & \mathbf{K}^{(EFI)} \\ \mathbf{K}^{(IEF)} & \mathbf{K}^{(II)} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(EF)} \\ \mathbf{u}^{(I)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{(EF)} \\ \mathbf{f}^{(IEF)} \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} \hat{\mathbf{K}}^{(EC)} & \hat{\mathbf{K}}^{(ECI)} \\ \hat{\mathbf{K}}^{(IEC)} & \hat{\mathbf{K}}^{(II)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}^{(EC)} \\ \hat{\mathbf{u}}^{(I)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}^{(EC)} \\ \hat{\mathbf{f}}^{(IEC)} \end{bmatrix} \tag{22}$$

onde os sobre-escritos EC, EF e I representam elementos de contorno, elementos finitos e a interface MEC-MEF respectivamente.

Na fase de acoplamento do MEC com o MEF é utilizado um algoritmo de busca linear, com o objetivo de encontrar os nós pertencentes a interface comum entre o MEC e o MEF e criar uma rotulação global para todos os nós do problema MEC-MEF. Subsequentemente o sistema global do acoplamento MEC-MEF é montado utilizando as técnicas usuais de *assembly* do MEF para o acoplamento da matriz de rigidez do MEC, podendo ser escrita na forma:

$$\begin{bmatrix} \hat{\mathbf{K}}^{(EC)} & \hat{\mathbf{K}}^{(ECI)} & \mathbf{0} \\ \hat{\mathbf{K}}^{(IEC)} & \mathbf{K}^{(I)} & \mathbf{K}^{(EFI)} \\ \mathbf{0} & \mathbf{K}^{(IEF)} & \mathbf{K}^{(EF)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}^{(EC)} \\ \mathbf{u}^{(I)} \\ \mathbf{u}^{(EF)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}^{(EC)} \\ \hat{\mathbf{f}}^{(IEC)} + \mathbf{f}^{(IEF)} \\ \hat{\mathbf{f}}^{(EF)} \end{bmatrix} \tag{23}$$

ou escrevendo o sistema global do problema acoplado MEC-MEF em forma compacta:

$$\mathbf{K}\,\mathbf{u} = \mathbf{f} \tag{24}$$

Deve se observar que a matriz de rigidez do MEF $\mathbf{K}^{(EF)}$ é simétrica positiva definida e esparsa enquanto que a matriz de rigidez equivalente do MEC $\hat{\mathbf{K}}^{(EC)}$ é não simétrica e densa. Logo a matriz de rigidez global do problema acoplado $\mathbf{K}$ perde a característica de simetria da região do MEF. Algumas tentativas de simetrizar a matriz de rigidez equivalente do MEC via o método de colocação de forma aproximada foram consideradas. No trabalho de Georgiou (Georgiou, 1981), onde é feita uma análise da possibilidade de se obter uma matriz simétrica através da distribuição das diferenças entre os seus diversos elementos.

3 IMPLEMENTAÇÃO *MULTITHREADED* DO ACOPLAMENTO MEC-MEF

Nesta seção são apresentadas as ferramentas computacionais utilizadas para realizar o acoplamento MEC-MEF e alguns resultados de desempenho computacional obtidos com o desenvolvimento de uma estratégia *multithreaded* (Carver e Tai, 2006), para minimizar o gasto de tempo de processamento na análise numérica realizada com o acoplamento MEC-MEF, o qual é desenvolvido para simulação numérica de problemas da Elasticidade 2D e Visualização Científica. A eficiência computacional ao se utilizar a programação *multithreading* na implementação de

métodos numéricos em geral, é considerável quando se utiliza um processador com mais de um núcleo de processamento, estes que por sua vez estão ficando cada vez mais populares. Portanto o paralelismo é obtido via programação *multithreading*. O programa desenvolvido pode ser executado em processadores que possuam somente um núcleo (neste caso as *threads* executarão concorrentemente) em sistemas operacionais Unix ou Windows.

O código computacional para realizar o acoplamento MEC-MEF foi implementado em linguagem Java (Deitel e Deitel, 2005), a qual permite um desenvolvimento utilizando programação orientada a objetos, permitindo uma expansão mais fácil a outros tipos de problemas ou métodos numéricos que futuramente serão desenvolvidos. A linguagem JAVA é escolhida pois já existe um software denominado MEMEC (Fontes Júnior et al., 2008; Vera-Tudela et al., 2009) já desenvolvido pelos autores que é um ambiente computacional gráfico interativo, desenvolvido para simulação numérica de problemas da Elastostática 2D através do Método dos Elementos de Contorno e Visualização Científica (Fig. 2). Portanto o código para realizar o acoplamento MEC-MEF é incluído no softare MEMEC.

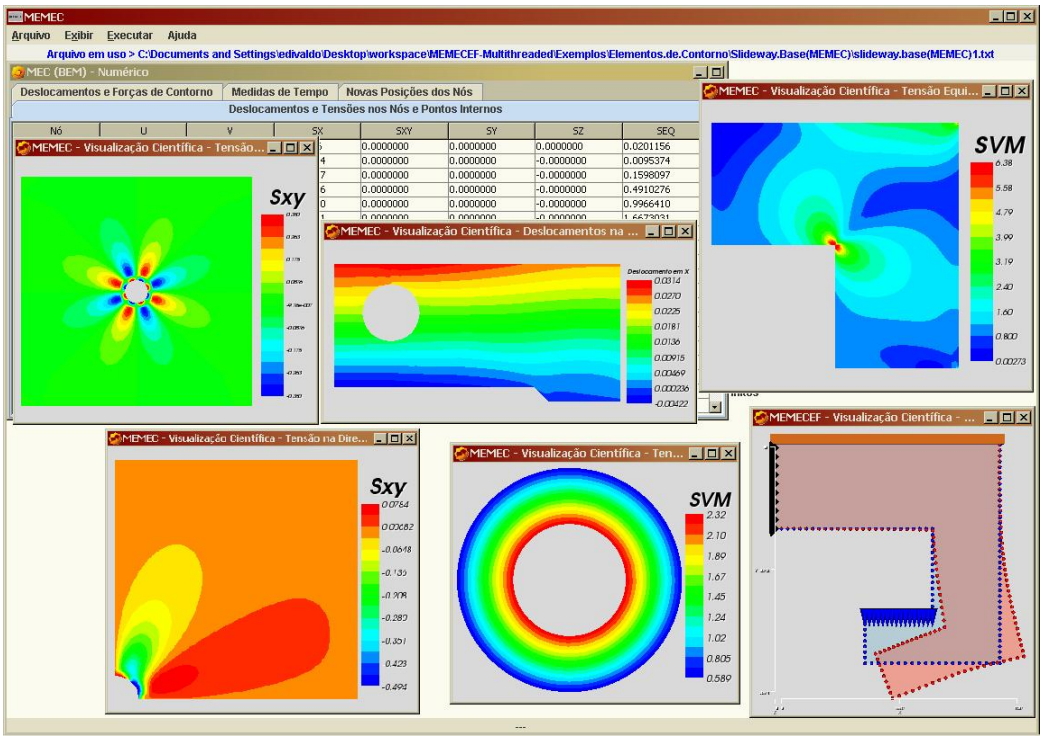

Figura 2: O software MEMEC.

O código computacional do MEC é uma versão traduzida para o Java do código desenvolvido em Fortran 77 por Telles (Brebbia et al., 1984) com algumas modificações tais como a estratégia de programação *multithreading* para as matrizes do MEC, a transformação de coordenadas Telles (Telles, 1987), a solução fundamental de Melan (Telles e Brebbia, 1981), algumas das técnicas de otimização apresentadas no trabalho de Cunha et al. (Cunha et al., 2008) são utilizadas, o *solver* do sistema de equações lineares direto é substituído pelo solver iterativo GMRES (método do resíduo mínimo generalizado) (Saad e Schultz, 1986) em sua versão *multithread* implementada por Wendykier (2009) e as classes referentes as estruturas de dados de matrizes esparsas e densas do conjunto de bibliotecas Colt (CERN, 2004) são empregadas para a construção da maioria das matrizes do MEC anteriormente definidas. O conjunto de bibliote-

cas COLT (CERN, 2004) é um conjunto de ferramentas livre desenvolvido em Java que possui ferramentas de álgebra linear para computação de alto desempenho utilizadas pelo *European Organization for Nuclear Research* (CERN).

O código computacional do MEF aqui empregado foi desenvolvido baseando-se nas referências (Hughes, 2000; Ribeiro, 2004; Bathe, 1996) para o cálculo dos deslocamentos nodais e na referência (Felippa, 2001) para o calculo das tensões nodais. E também é desenvolvido um código baseado no conjunto de bibliotecas VTK (Schroeder et al., 2004) para a análise gráfica dos resultados obtidos com o acoplamento MEC-MEF.

3.1 Vantagens da Programação *Multithreading*

Um programa escrito sem o uso de *threads*, executando em um computador com processador multi-núcleo não tem ganho de desempenho significativo se comparado a um processador com somente um núcleo de mesma velocidade. Porém o uso da programação *multithreading* em um processador multi-núcleo pode aumentar consideravelmente a performance de um programa, pois as *threads* serão executadas paralelamente, aumentando a eficiência dos processadores multi-núcleo. Teoricamente um programa que utiliza totalmente dois processadores pode obter um tempo de processamento próximo a metade do tempo do mesmo programa utilizando somente um processador. No entanto esse nível de *speedup* não pode ser obtido devido ao tempo gasto no gerenciamento correto das *threads* envolvidas no processo e a largura de banda de memória RAM ser compartilhada pelos núcleos do processador (Carver e Tai, 2006). As vantagens no uso de tais técnicas podem ser vistas e comprovadas com mais detalhes nos resultados apresentados mais a frente.

A principal desvantagem de programas *multithreaded* é que seu desenvolvimento é muito complexo, principalmente pelo fato das *threads* compartilharem a mesma área de memória. A implementação errada pode levar a ocorrência de erros que são difíceis de encontrar e resolver, tais como *deadlock* ou *race condition* (acesso a dados compartilhados entre *threads* simultaneamente), melhores explicados em (Oaks e Wong, 2004; Goetz et al., 2006), sendo necessário a utilização de técnicas de gerenciamento de *threads*, o que consome mais tempo de processamento. E a escalabilidade (representa a capacidade proporcional de aumento de performance de um código paralelo ao se adicionar processadores) é limitada pela arquitetura de memória.

3.2 Montagem *multithreaded* das Matrizes do MEC

Como o tempo de montagem das matrizes do MEC, na maioria das discretizações aqui empregadas, é superior ao da montagem da matriz de rigidez do MEF, as técnicas de programação *multithreading* são implementadas de forma a aumentar a eficiência para montagem das matrizes do MEC. Como pode ser visto no pseudo-código abaixo, onde é mostrado o trecho onde se gasta a maior parte do tempo para a montagem do sistema de equações (6), o MEC mostra-se um método numérico altamente paralelizável.

```
1  Loop nos Pontos de Colocação
2    Loop nos Elementos
3      Loop de Integração
4        ...
5        /*Cálculo das soluções fundamentais e suas derivadas
6          em todos os pontos de integração no elemento para
7          cada ponto de colocação*/
8        ...
9      Fim do Loop de Integração
10   Fim do Loop nos Elementos
11 Fim do Loop nos Nós
```

Cabe ressaltar que as técnicas de paralelização via técnicas *multithreading* foram implementadas depois de ser realizado uma otimização do código serial do MEC como um todo.

Primeiramente *m threads* são criadas e associadas a *n* núcleos de processamento (formando um processo *multithreaded*). Observe que pode-se utilizar $m \geq n$, porém com $m > n$ resulta em um maior gasto dos recursos da CPU, não havendo ganho significativo de desempenho. Portanto $m = n$ é sempre utilizado como número de *threads* para o cálculo dos coeficientes das matrizes do MEC.

A implementação de forma paralela da Equação Integral de Contorno (6), pode ser obtida da seguinte forma. Para cada ponto de colocação ξ correspondente a cada nó do contorno é associada uma *thread* a qual executará a tarefa de montagem das matrizes locais **g** e **h** para cada elemento de contorno, por meio de integração numérica no elemento linear do contorno, gerando um conjunto de duas equações para cada nó. Portanto para um processador com *n* núcleos, pode-se montar *n* conjuntos de equações paralelamente, para gerar as matrizes **H** e **G** definidas na Eq. (7). E finalmente a matriz de rigidez equivalente do MEC é obtida e acoplada de forma usual a matriz de rigidez do MEF concluindo a montagem do sistema de equações global (24).

3.3 Solução do Sistema de Equações Acoplado

A outra parte do código que também requer um grande esforço computacional é na solução do sistema de equações não simétrico e esparso do problema acoplado MEC-MEF. Para a solução do sistema de equações optou-se pelo uso do *solver* iterativo GMRES (Saad e Schultz, 1986) em sua versão multithread (Wendykier, 2009).

O acoplamento entre o MEC e o MEF para problemas de Elasticidade linear, fornece um sistema de equações esparso, não simétrico e real. Vários métodos de solução de sistemas de equações lineares baseados em pacotes tais como LAPACK e LINPACK foram testados (JAMA, .; MTJ, .; CERN, 2004; Wendykier, 2009). Neste trabalho optou-se pela utilização da biblioteca COLT (CERN, 2004) na sua versão *multithreaded* (Wendykier, 2009) dentre as outras bibliotecas citadas, pela sua eficiência devido a quantidade de *solvers* implementados de forma *multithread* e por possuir estruturas de dados para armazenamento de matrizes esparsas. Portanto o *solver* direto implementado anteriormente por Telles (Brebbia et al., 1984) é substituído pelo *solver* iterativo GMRES e utilizado em todo o trabalho.

No presente trabalho é utilizado o pré-condicionandor de Jacobi para melhorar o grau de condicionamento do sistema matricial MEC-MEF, formado pela diagonal da matriz **K** do sistema acoplado MEC-MEF, apresentando uma melhora significativa na convergência do GMRES. Como o objetivo do GMRES é para fins computacionais a sua implementação com reinício é utilizada, de forma que o tamanho da base do subespaço de Krylov utilizada seja muito menor que a ordem do sistema de equações original. O número de *restart* usualmente utilizado neste trabalho é 60.

O *solver* iterativo GMRES reinicializável em sua versão multithreaded (Wendykier, 2009) utilizando pré-condicionador diagonal, implementa de forma paralela a operação de multiplicação entre a matriz do sistema de equações e um vetor de aproximação, a qual consome cerca de 95% do tempo total de resolução não-paralela.

4 RESULTADOS

Baseado nas estratégias do uso de *threads* para a implementação do MEC, é medido o *speedup* para uma viga em balanço com carga uniformemente distribuída na extremidade (Fig.

3), para diversas discretizações, executados em um computador com processador Intel Core i7 940 (codinome *Bloomfield*). O processador *Bloomfield* possui *clock* de 2.93GHz com 4x256KB de *cache L2*, 8MB de *cache* L3 e 8GB de memória RAM. O exemplo aqui é analisado somente com o problema discretizado com o MEC, para validar o esquema de montagem paralela das matrizes **H** e **G**.

O código foi paralelizado usando os recursos de programação *multithreading* intrínsecos a liguagem de programação Java utilizando a versão 1.6 da maquina virtual. Utiliza-se aqui como métrica de desempenho o *speedup*. O speedup em um computador multinúcleo pode ser definido por

$$S = \frac{T_1}{T_n} \tag{25}$$

onde T_1 e T_n são as medidas de tempo ao se utilizar uma única *thread* e ao se utilizar *n threads* respectivamente. A Figura 4 mostra os *speedups* obtidos para a montagem da matriz e a solução do sistema de um problema com 8004 nós. Pode-se observar também na Fig. 4 uma maior eficiência na montagem da matriz sobre a solução do sistema. O que pode ser explicado, pelo fato de a montagem ser em sua maior parte limitada pela capacidade de processamento, e a solução do sistema pelo método GMRES é tanto limitada pela capacidade de processamento quanto pela quantidade de memória RAM do computador.

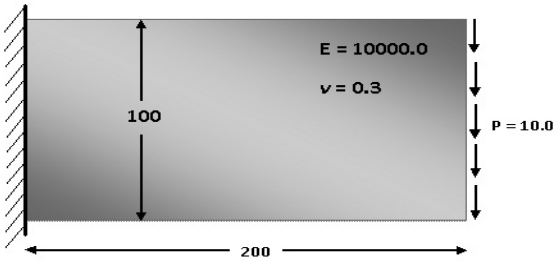

Figura 3: Viga em balanço com carga uniformemente distribuída na extremidade.

O *speedup* exposto na Fig. 4 para a discretização do problema exposto na Fig. 3 com 8000 elementos mostra um *speedup* superlinear para o número de *threads* igual a dois para o GMRES. Tal ganho de eficiência superlinear para o GMRES pode ser explicado pela arquitetura do processador *Core* i7 permitir uma melhor utilização da *Cache* L3 e o *overclock* automático utilizado (tecnologia *Turbo Boost*). Observando ainda o *speedup* do GMRES, pode-se ver uma perda de eficiência em relação ao *speedup* da montagem da matriz quando são utilizadas quatro ou mais *threads*. Esta perda de performance pode ser explicada pela tecnologia *Hyper Threading* que simula 8 núcleos de processamento para o sistema operacional, e quando mais que 4 *threads* são utilizadas, principamente para o GMRES, pode se dar frequentes faltas de dados na memória *cache*, fazendo com que o dado seja procurado na memória RAM. Para uma análise mais precisa sobre o desempenho ao se utilizar o processador *Core* i7 a tecnologia *Turbo Boost* deveria ser desabilitada no *setup* do computador.

A Figura 5 mostra o tempo total gasto para realizar a montagem da matriz do problema versus o número de elementos para o número de *threads* variando entre 1 e 8. Pode-se observar, por exemplo, que para 8000 elementos o tempo computacional da montagem das matrizes do MEC com 8 *threads* foi aproximadamente 4 vezes mais rápido do que quando utilizado uma única *thread*.

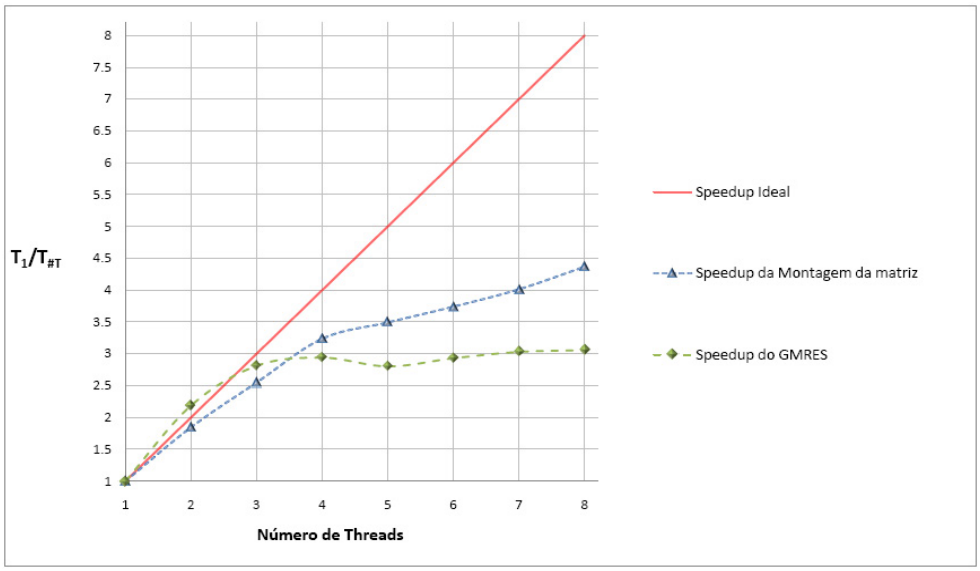

Figura 4: Speedup obtido para a montagem e a solução do sistema de equações do MEC de um problema discretizado com 8004 nós.

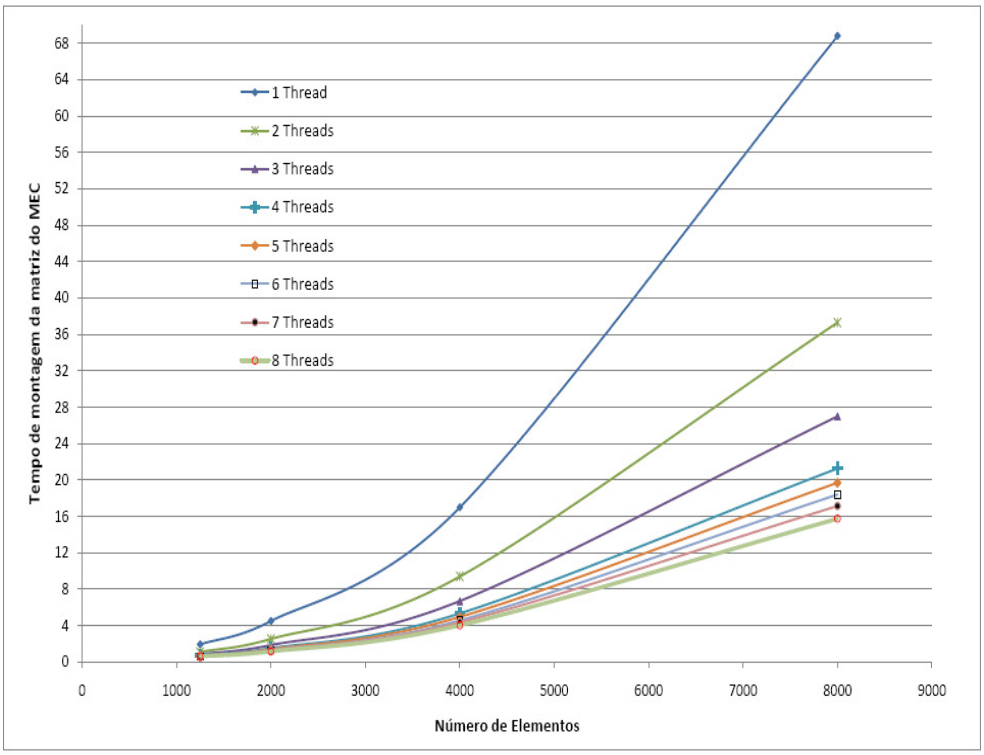

Figura 5: Tempo da montagem da matriz versus o número de elementos para 1, 2, ... , 8 threads.

Os resultados são aqui apresentados somente para problemas analisados com o MEC, e são utilizados também para o acoplamento MEC-MEF, porém aqui não são apresentados tais resultados. Nas discretizações utilizadas para os diversos problemas, onde procura-se não se utilizar uma malha muito refinada para a região do MEC os tempos de execução ficam então dominados pela solução do sistema de equações.

Os resultados aqui obtidos com o processador *multi-core* equipado com quatro núcleos físicos de processamento, mostram um ótimo ganho de desempenho na versão *multithreaded* do código computacional para o acoplamento MEC-MEF. A implementação *multithreaded* tende a se tornar um padrão de programação, e o código aqui desenvolvido pode ser estendido a qualquer quantidade de núcleos de processamento. Tais implementações são fundamentais, posto que os modernos clusters de PC's possuem diversos processadores com a tecnologia *multi-core*.

5 CONCLUSÃO

Um procedimento para realizar o acoplamento entre o MEC e o MEF para problemas de Elasticidade bidimensional é apresentado, utilizando técnicas de programação *multithreading*.

O uso eficiente de processadores *multi-core*, com as técnicas de programação *multithreading* foi comprovado em um computador pessoal, para o programa de elasticidade bidimensional. As técnicas de programação *multithreading* são implementadas especialmente para o código do MEC e para resolução do sistema de equações com *solver* iterativo GMRES. Os *speedups* obtidos mostram uma ótima eficiência dos processadores *multi-core*.

Cabe ressaltar que além dos resultados satisfatórios, o código computacional aqui escrito em linguagem Java e o uso de bibliotecas de computação de alto desempenho gratuitas disponíveis, permite que sejam desenvolvidas implementações para diversos tipos de aplicações e ainda garante-se a portabilidade do código entre as várias plataformas existentes atualmente.

6 AGRADECIMENTOS

Este trabalho foi desenvolvido com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS

Aour B., Rahmani O., e Nait-Abdelaziz M. A coupled fem/bem approach and its accuracy for solving crack problems in fracture mechanics. *International Journal of Solids and Structures*, 44:2523–2539, 2007.

Bathe K. *Finite Element Procedures*. Prentice Hall, 1996.

Brebbia C.A., Telles J.C.F., e Wrobel L.C. *Boundary Element Techniques: Theory and Applications in Engineering*. Springer, Berlin, 1984.

Carver R.H. e Tai K.C. *Modern Multithreading: Implementing, Testing, and Debugging Multithreaded Java and C++/Pthreads/Win32 Programs*. John Wiley & Sons,, New Jersey, 2006.

CERN. *COLT - Open Source Libraries for High Performance Scientific and Technical Computing in Java*. CERN - European Organization for Nuclear Research, 2004. Http://acs.lbl.gov/ hoschek/colt/.

Cunha M.T.F., Telles J.C.F., e Ribeiro F.L.B. Streaming simd extensions applied to boundary element codes. *Advances in Engineering Software*, (39):888–898, 2008.

de Paula F.A. *Elementos de Contorno com equilíbrio :uma formulação consistente para problemas de potencial e elasticidade*. Doctor's Thesis, Prog. de Eng. Civil - COPPE - UFRJ, 1991.

Deitel H.M. e Deitel P.J. *Java - Como Programar*. Pearson Education, São Paulo, 6 edição, 2005.

Demmel J.W. *Applied Numerical Linear Algebra*. SIAM, 1997.

Felippa C.A. *Introduction to Finite Element Methods*. Lecture Notes for the course Introduction to Finite Elements Methods. University of Colorado, USA, 2001.

Fontes Júnior E.F., Barbosa M., Vera-Tudela C.A.R., e Telles J.C.F. O método de elementos de contorno e a visualização científica aliados à resolução de problemas da mecânica computacional. 2008.

Georgiou P. *The coupling of the direct boundary element method with the finite element displacement technique in elastostatics*. Doctor's Thesis, Southampton University, 1981.

Goetz B., Peierls T., Bloch J., Bowbeer J., Holmes D., e Lea D. *Java Concurrency in Practice*. Pearson Education, United States, 2006.

Hughes T.J.R. *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis*. Dover Publications, New York, USA, 2000.

JAMA. A java matrix package. . Http://math.nist.gov/javanumerics/jama.

MTJ. Matrix toolkits for java. . Http://rs.cipr.uib.no/mtj.

Oaks S. e Wong H. *Java Threads*. O'Reilly Media, 3 edição, 2004.

Ribeiro F.L. *Introdução ao Método dos Elementos Finitos*. COPPE-UFRJ, 2004. Notas de Aula do PEC.

Saad Y. e Schultz M.H. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *Society for Industrial and Applied Mathematics*, 7(3), 1986.

Schroeder W., Ken M., e Lorensen B. *The Visualization Toolkit*. Kitware, Inc, 3 edição, 2004.

Telles J.C.F. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *Int. J. Numer. Meth. Engng.*, 24:959–973, 1987.

Telles J.C.F. e Brebbia C.A. Boundary element solution for half-plane problems. *Int. J. Solids Structures*, 17(12):1149–1158, 1981.

Vera-Tudela C.A.R., Barbosa M., Fontes Júnior E.F., e Telles J.C.F. Desenvolvimento de software para a resolução de problemas da mecânica dos sólidos. *Congreso Interamericano de Computación Aplicada a la Industria de Procesos*, 1(9), 2009.

Wendykier P. *PARALLEL COLT - Open Source Libraries for High Performance Scientific and Technical Computing in Java*. 2009. Http://sites.google.com/site/piotrwendykier/software/parallelcolt.

Zienkiewicz O.C., Kelly D.W., e Bettess P. The coupling of the finite element method and boundary solution procedures. *Int. J. Numer. Meth. Engng.*, 11:355–375, 1977.