

UMA TÉCNICA ELEMENTO POR ELEMENTO LIVRE DE MATRIZES PARA A AVALIAÇÃO DE RESÍDUOS

ÁLVARO L.G.A. COUTINHO*

JOSÉ L.D. ALVES*

e

PHILIPPE R.B. DEVLOO**

* *COPPE/UFRJ Programa de Engenharia Civil*

Caixa Postal 68506

21945-970 Rio de Janeiro, RJ - Brasil

** *Faculdade de Engenharia Civil-UNICAMP*

Caixa Postal 6021

13081-970 Campinas, SP - Brasil

SUMÁRIO

Neste trabalho propõe-se um novo melhoramento à técnica elemento por elemento (EPE), que se utiliza da deficiência de posto natural das matrizes de elemento. Dessa forma, o cálculo dos resíduos necessário na solução iterativa do sistema de equações pode ser efetuado sem a avaliação explícita das matrizes de elemento. Os produtos matriz por vetor são otimizados para minimizar o número de operações de ponto flutuante e a área de memória. A técnica resultante constitui-se em um método livre de matrizes no sentido em que não há necessidade de armazenamento para as matrizes de elemento, o que provê um método poderoso para a abordagem de problemas de grandes dimensões. O desempenho da técnica EPE Livre de Matrizes é avaliado na solução de problemas simétricos e não simétricos em diversos ambientes computacionais, desde estações de trabalho até supercomputadores paralelos.

SUMMARY

In this work a new improvement is proposed for the element by element technique (EBE), using the natural rank deficiency of the element matrices. With this technique the residual calculation, required for iterative solution methods can be done without the explicit evaluation of the element matrices. The matrix-vector products are optimized to minimize the number of floating point operations and memory requirements. The resulting technique constitutes in a matrix free method, as there is no need to store the element matrices, providing a powerful tool for the treatment of large problems. The performance of the EBE Matrix Free technique is evaluated in the solution of symmetric and non-symmetric problems in several different computational environments, from workstations to parallel supercomputers.

Recibido: Enero 1995

INTRODUÇÃO

Formulações implícitas de elementos finitos para a solução de problemas em engenharia (linear ou não linear, em regime permanente ou transiente) requerem a solução de um sistema de equações lineares em cada passo de tempo ou iteração. Tanto a matriz de coeficientes como o vetor independente são construídos a partir das contribuições dos elementos. A matriz resultante é esparsa e em perfil. A ordem de tais sistemas pode variar de milhares a milhões de equações, tornando imperativa uma escolha cuidadosa do método de solução. Métodos diretos baseados em alguma forma da eliminação de Gauss podem ser utilizados para a solução desses sistemas, mas logo se tornam proibitivos para problemas de interesse prático, particularmente em três dimensões. Estratégias empregando métodos iterativos são mais interessantes devido à sua menor demanda de memória. Entretanto, o número de operações necessário ao cálculo da solução dentro de uma tolerância pré-especificada não é conhecido a priori, sendo dependente de algumas propriedades do sistema e da escolha de um pré-condicionador adequado.

No âmbito do método dos elementos finitos, a implementação de estratégias iterativas de solução empregando técnicas elemento por elemento (EPE), tanto para a avaliação do resíduo como para a construção do pré-condicionador, oferece várias vantagens. A estrutura EPE permite a vetorização/paralelização de toda a estratégia de solução, apresenta uma baixa demanda de memória e é muito adequada para os métodos adaptativos e multimalha já que é insensível à ordem de numeração dos nós. As técnicas EPE tem sido utilizadas com sucesso na solução de sistemas de equações simétricos e não simétricos em várias aplicações, desde estruturas e sólidos até problemas de fluidos⁸.

Neste trabalho, propõe-se um novo melhoramento à técnica elemento-por-elemento, que se utiliza da deficiência de posto natural das matrizes de elemento. Cada matriz de elemento possui, dependendo da equação diferencial envolvida em sua formulação, um certo grau de singularidade. A Tabela I sumariza, para alguns tipos de elementos finitos, a correspondente deficiência de posto verificada. É possível, então, calcular os resíduos nos elementos minimizando-se o volume de operações de ponto flutuante necessárias e, em alguns casos, eliminando-se por completo o cálculo e armazenamento das matrizes de elemento propriamente ditas. É nesse sentido que a nova técnica EPE constitui-se em um método livre de matrizes, proporcionando uma estratégia vantajosa para a solução de problemas de grandes dimensões.

Elemento Finito	Deficiência de Posto
Transferência de Calor (1D,2D,3D)	1
Viga	2
Treliça 3D	5
Elasticidade 3D	6
Casca	6

Tabela I. Deficiência de Posto de alguns Elementos Finitos

O restante do presente trabalho encontra-se organizado da seguinte forma. Na seção que se segue são resumidas as características essenciais da técnica EPE. Na seqüência, a nova técnica EPE é aplicada ao cálculo dos resíduos em treliças espaciais, geometricamente não-lineares. Na próxima seção, mostra-se como esses conceitos podem ser estendidos a formulações estabilizadas do método dos elementos finitos através da solução de um problema escalar puramente convectivo, discretizado segundo a formulação SUPG, em dois supercomputadores vetoriais/paralelos, o CRAY Y-MP e o CRAY C90. O artigo encerra-se com um resumo das principais conclusões desse trabalho.

1. TÉCNICAS ELEMENTO-POR-ELEMENTO

Deve-se ter em mente que, nas implementações de operações esparsas entre matrizes e vetores na forma elemento-por-elemento, nem a matriz \mathbf{A} nem o vetor independente \mathbf{b} são montados. O vetor de resíduos, \mathbf{r} , é construído a partir das contribuições individuais dos elementos segundo

$$\mathbf{r} = \sum_{e=1}^{nel} \mathbf{b}_e - \sum_{e=1}^{nel} \mathbf{A}_e \mathbf{u}_e \quad (1)$$

onde \mathbf{b}_e é o vetor independente a nível de elemento, \mathbf{A}_e é a matriz de elemento e \mathbf{u}_e são as componentes do vetor global da solução, \mathbf{u} , restrito aos graus de liberdade do elemento. A quantidade de elementos na malha é dada por nel . As operações EPE são em geral realizadas de acordo com os seguintes passos⁷:

1. Agrupar as componentes de um arranjo global para o arranjo correspondente aos graus de liberdade do elemento;
2. Efetuar as operações EPE;
3. Espalhar e acumular os resultados em um arranjo global.

Os cálculos envolvidos nos passos de 1 a 3 tem um conteúdo vetorial intrínseco, tornando as operações EPE adequadas para implementação nos supercomputadores atuais^{5,3}, como veremos adiante. No entanto, para os elementos finitos usuais, as dimensões dos arranjos envolvidos nos cálculos são relativamente pequenas. Nos supercomputadores vetoriais o desempenho global pode ser significativamente melhorado através do processamento blocado dos elementos, indexando os cálculos pelos elementos pertencentes ao bloco corrente. Uma vez que vários elementos podem compartilhar um nó comum (ou grau de liberdade) os resultados da operação de espalhamento e acumulação serão determinados pelo último elemento a ser processado. A solução para se evitar tal problema reside na reordenação dos elementos segundo blocos internamente disjuntos, onde nenhum par de elementos pertencentes a um mesmo grupo possui um nó comum. Esse enfoque é equivalente à aplicação de um algoritmo de coloração de malha. No âmbito de cada cor ou grupo pode-se também paralelizar as operações, o que torna esse esquema bem adequado aos computadores vetoriais/paralelos.

2. CÁLCULO EFICIENTE DE RESÍDUOS PARA TRELIÇAS ESPACIAIS GEOMETRICAMENTE NÃO LINEARES

Freqüentemente, estruturas reticuladas espaciais de grande porte requerem o emprego de uma formulação não-linear geométrica para a obtenção de uma melhor aproximação dos esforços normais nas barras. Métodos iterativos são mais adequados a esse cálculo, uma vez que aproximam o estado final de equilíbrio de uma forma contínua, sem a necessidade da montagem e fatoração da matriz de rigidez global. Já foi demonstrado que ao se considerar a estrutura especial da matriz Hessiana para treliças espaciais geometricamente não-lineares⁴, uma economia considerável pode ser obtida em termos de tempo de CPU em comparação à implementações que empregam a matriz de rigidez propriamente dita. A energia potencial interna de uma treliça espacial é dada por

$$U = \frac{EA}{2l} \Delta l^2 \quad (2)$$

onde A é a área da seção transversal da treliça, E é o módulo de elasticidade, l é o comprimento da treliça e Δl é o alongamento. No âmbito de um sistema de coordenadas cartesiano, (x_1, y_1, z_1) e (x_2, y_2, z_2) são as coordenadas das extremidades da treliça na configuração de referência. Sejam (u_1, v_1, w_1) e (u_2, v_2, w_2) os deslocamentos de suas extremidades, medidos no mesmo sistema de coordenadas. Portanto

$$(\Delta x, \Delta y, \Delta z) = (x_2 - x_1, y_2 - y_1, z_2 - z_1) \quad (3)$$

e

$$l = (\Delta x^2 + \Delta y^2 + \Delta z^2)^{1/2} \quad (4)$$

$$(\Delta u, \Delta v, \Delta w) = (u_2 - u_1, v_2 - v_1, w_2 - w_1) \quad (5)$$

$$l + \Delta l = [(\Delta x + \Delta u)^2 + (\Delta y + \Delta v)^2 + (\Delta z + \Delta w)^2]^{1/2} \quad (6)$$

Tomando-se duas derivadas parciais em relação a $(\Delta u, \Delta v, \Delta w)$ chega-se à seguinte matriz Hessiana não-linear,

$$\frac{\partial^2 U}{\partial (\Delta u, \Delta v, \Delta w)^2} = \frac{EA}{l} \frac{\Delta l}{(l + \Delta l)} \mathbf{I}_3 + \frac{EA}{l} \left[\frac{1}{(l + \Delta l)} - \frac{\Delta l}{(l + \Delta l)^2} \right] \mathbf{d} \otimes \mathbf{d} \quad (7)$$

onde \mathbf{I}_3 é a matriz identidade de ordem três e $\mathbf{d}^T = (\Delta x + \Delta u, \Delta y + \Delta v, \Delta z + \Delta w)^T$. Nota-se claramente nesse caso, que a matriz Hessiana é a soma de duas matrizes de posto um e três, respectivamente. Ao serem calculados os produtos matriz por vetor, as operações serão efetuadas de acordo com a estratégia EPE, o que totaliza 36nel multiplicações e 30nel adições. Considerando-se a estrutura especial dos elementos,

o número de operações pode ser consideravelmente reduzido. As etapas genéricas da estratégia EPE, nesse caso, podem ser re-escritas da seguinte maneira:

1. Espalhe os deslocamentos nodais para um vetor local de deslocamentos, mantendo u , v e w em arranjos separados;
2. Calcule $\Delta u^T = (\Delta u, \Delta v, \Delta w)^T$;
3. Calcule $\Delta u^* = \Delta u(\Delta x + \Delta u)$, $\Delta v^* = \Delta v(\Delta y + \Delta v)$, $\Delta w^* = \Delta w(\Delta z + \Delta w)$ e some para obter $S = \Delta u^* + \Delta v^* + \Delta w^*$;
4. Obtenha $\mathbf{s}^T = S\mathbf{d}^T = (S^u, S^v, S^w)$;
5. Multiplique \mathbf{s}^T por $\frac{EA}{l} \left[\frac{1}{(l+\Delta l) - \frac{\Delta l}{(l+\Delta l)^2}} \right]$ para obter S^{su}, S^{sv}, S^{sw} ;
6. Calcule as forças nodais equivalentes :

$$F^u = \Delta u^* \frac{EA}{l} \left(\frac{\Delta l}{l+\Delta l} \right) + S^{su}$$

$$F^v = \Delta v^* \frac{EA}{l} \left(\frac{\Delta l}{l+\Delta l} \right) + S^{sv}$$

$$F^w = \Delta w^* \frac{EA}{l} \left(\frac{\Delta l}{l+\Delta l} \right) + S^{sw}$$
7. Monte as contribuições dos resíduos no elemento para os nós.

Utilizando os passos descritos acima o número de operações para efetuar um produto matriz por vetor é $12nel$ multiplicações e $5nel$ adições. O algoritmo utilizado para a solução as equações não-lineares é o método de Newton onde a inversão da matriz para as iterações de Newton é feita pelo método dos gradientes conjugados preconditionado. O preconditionador utilizado é a matriz bloco-diagonal onde cada bloco corresponde a uma matriz nodal 3×3 . Os resultados numéricos mostram que mesmo para estruturas moderadamente grandes o número de iterações não é excessivo. Para uma estrutura com 921 nós e 3516 barras, a análise não-linear geométrica com a presente formulação convergiu em 4 iterações. O número de iterações do método de gradientes conjugados preconditionado correspondente à análise não-linear completa é de 184. O desempenho global do algoritmo indicou a taxa de 2.5 Mflops em uma estação de trabalho IBM RS/6000 Modelo 320, onde 8.8 segundos de CPU foram gastos na rotina dos produtos matriz por vetor. Pode-se concluir que a combinação eficiente dos métodos iterativos com produtos matriz por vetor otimizados proporciona tempos de solução da ordem de segundos, mesmo para treliças de grande porte, abrindo a possibilidade do desenvolvimento de algoritmos de otimização de treliças que exijam a solução de várias configurações.

3. EQUAÇÃO DE CONVECÇÃO-DIFUSÃO ESCALAR

A equação de convecção-difusão escalar, em regime permanente, em duas dimensões, descreve fenômenos importantes e pode ser escrita na forma

$$v\nabla u - \nabla(k\nabla u) - f = 0, \quad \text{em } \Omega \subset \mathbb{R}^2 \quad (8)$$

onde v é um campo de velocidades com divergente nulo, k é o coeficiente de difusão e f representa os termos fonte no volume. As condições de contorno são dadas por

$$u = \bar{u} \quad \text{em } \Gamma_u \quad (9)$$

$$\mathbf{n} \cdot k \nabla u = g \quad \text{em } \Gamma_g \quad (10)$$

onde o contorno de Ω é tal que $\partial\Omega = \Gamma_u \cup \Gamma_g$ e \mathbf{n} é o vetor unitário normal à fronteira, orientado para o exterior do domínio.

A formulação de elementos finitos Streamline Upwind Petrov-Galerkin (SUPG) para o problema acima pode ser dada por

$$B(w^h, u^h) + \sum_{e=1}^{nel} \int_{\Omega} \tau \frac{v}{\|v\|} \nabla w^h \mathcal{L}(u^h) d\Omega = 0 \quad (11)$$

onde w^h é a função discreta de ponderação e u^h é a forma discreta de u . As funções w^h e u^h são definidas sobre os espaços usuais de elementos finitos. Os termos da formulação acima são

$$B(w^h, u^h) = \int_{\Omega} \nabla w^h \mathcal{L}(u^h) d\Omega - \int_{\Gamma_u} w^h g d\Gamma \quad (12)$$

$$\mathcal{L}(u^h) = v \cdot \nabla u^h - \nabla \cdot (k \nabla u^h) - f \quad (13)$$

A equação (12), após a integração por partes usual, corresponde ao termo de Galerkin. A segunda integral, equação (11), é o termo SUPG². A formulação é variacionalmente consistente no sentido em que para $u^h \rightarrow u$, o termo de estabilização se anula. O parametro τ é estabelecido para prover a quantidade correta de difusão ao longo das linhas de corrente. Considerando-se apenas elementos triangulares de lado reto e funções de interpolação lineares, a formulação SUPG conduz, após integração, às seguintes matrizes de elemento

$$\mathbf{K}_A = \frac{v_x}{6} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ y_{23} & y_{31} & y_{12} \\ y_{23} & y_{31} & y_{12} \end{bmatrix} + \frac{v_y}{6} \begin{bmatrix} x_{32} & x_{13} & x_{21} \\ x_{32} & x_{13} & x_{21} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \quad (14)$$

$$\mathbf{K}_D = \frac{k}{4A} \begin{bmatrix} y_{23}y_{23} + x_{32}x_{32} & y_{23}y_{31} + x_{32}x_{13} & y_{23}y_{12} + x_{32}x_{21} \\ y_{31}y_{23} + x_{13}x_{32} & y_{31}y_{31} + x_{13}x_{13} & y_{31}y_{12} + x_{31}x_{21} \\ y_{12}y_{23} + x_{21}x_{32} & y_{12}y_{31} + x_{12}x_{13} & y_{12}y_{12} + x_{12}x_{21} \end{bmatrix} \quad (15)$$

$$\mathbf{K}_{PG} = \frac{\tau}{4A \|v\|} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} v_x^2 & v_x v_y \\ v_y v_x & v_y^2 \end{bmatrix} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \quad (16)$$

onde $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$, $i, j = 1, 2, 3$ são as diferenças entre as coordenadas cartesianas dos nós do elemento, A é a área do elemento v_x e v_y são as velocidades do centróide do elemento:

$$v_x = \frac{v_x^1 + v_x^2 + v_x^3}{3}, \quad v_y = \frac{v_y^1 + v_y^2 + v_y^3}{3} \quad (17)$$

A matriz \mathbf{K}_A corresponde ao termo convectivo de Galerkin e é fácil verificar que se trata de uma matriz de posto igual a um. A matriz de difusão de Galerkin, designada por \mathbf{K}_D , é uma matriz de posto igual a dois. A contribuição SUPG é representada pela matriz \mathbf{K}_{PG} que também é uma matriz de posto dois. A matriz final para o elemento é a soma dessas três matrizes. O sistema de equações resultante é em geral não simétrico. Considerando que se deseja de fato calcular os produtos matriz por vetor durante o processo iterativo de solução, ou nesse caso, os fluxos nos elementos, uma economia considerável em relação ao número de operações de ponto flutuante pode ser alcançada através da avaliação direta dos fluxos. Supondo que os valores da solução para os nós do elemento são dados por u_1, u_2 e u_3 os fluxos convectivos de Galerkin são calculados simplesmente por

$$F_1^A = \frac{v_x}{6} [y_{23}u_1 + y_{31}u_2 + y_{12}u_3] + \frac{v_y}{6} [x_{32}u_1 + x_{13}u_2 + x_{21}u_3] \quad (18)$$

$$F_2^A = F_1^A \quad (19)$$

$$F_3^A = F_1^A \quad (20)$$

e os fluxos difusivos de Galerkin como

$$F_1^D = \frac{k}{4A} [y_{23} (y_{23}u_1 + y_{31}u_2 + y_{12}u_3) + x_{32} (x_{32}u_1 + x_{13}u_2 + x_{21}u_3)] \quad (21)$$

$$F_2^D = \frac{k}{4A} [y_{31} (y_{23}u_1 + y_{31}u_2 + y_{12}u_3) + x_{13} (x_{32}u_1 + x_{13}u_2 + x_{21}u_3)] \quad (22)$$

$$F_3^D = - (F_1^D + F_2^D) \quad (23)$$

finalmente, os fluxos correspondentes ao termo SUPG são dados por

$$F_1^{PG} = \frac{\tau}{4A \|v\|} \left\{ \left[y_{23} (v_x^2 y_{23} + v_x v_y x_{32}) + x_{32} (v_x v_y y_{23} + v_x^2 x_{32}) \right] u_1 + \right. \\ \left. + \left[y_{23} (v_x^2 y_{31} + v_x v_y x_{13}) + x_{32} (v_x v_y y_{31} + v_y^2 x_{13}) \right] u_2 + \right. \\ \left. + \left[y_{23} (v_x^2 y_{12} + v_x v_y x_{21}) + x_{32} (v_x v_y y_{12} + v_y^2 x_{21}) \right] u_3 \right\} \quad (24)$$

$$F_2^{PG} = \frac{\tau}{4A \|v\|} \left\{ \left[y_{31} (v_x^2 y_{23} + v_x v_y x_{32}) + x_{13} (v_x v_y y_{23} + v_x^2 x_{32}) \right] u_1 + \right. \\ \left. + \left[y_{31} (v_x^2 y_{31} + v_x v_y x_{13}) + x_{13} (v_x v_y y_{31} + v_y^2 x_{13}) \right] u_2 + \right. \\ \left. + \left[y_{31} (v_x^2 y_{12} + v_x v_y x_{21}) + x_{13} (v_x v_y y_{12} + v_y^2 x_{21}) \right] u_3 \right\} \quad (25)$$

$$F_3^{PG} = - (F_1^{PG} + F_2^{PG}) \quad (26)$$

Não há portanto a necessidade do cálculo e armazenamento das matrizes de elemento, tornando essa nova técnica EPE um método livre de matrizes. É interessante notar que esse enfoque, embora um método de elementos finitos, guarda uma acentuada semelhança com o método dos volumes finitos⁶. Para a avaliação da técnica EPE Livre de Matrizes selecionamos um problema simples de convecção escalar em um domínio quadrado $\Omega = [0, 1] \times [0, 1]$, com as condições de contorno, $u = 0$ em $y = 0$; $u = 0$ em $\{(x, y) \mid x = 0, y < 0.25\}$; $u = 1$ em $\{(x, y) \mid x = 0, 0.25 < y < 1\}$. Condições de fluxo nulo no contorno foram prescritas em $x = 1$ e $y = 1$. O campo de velocidades tem módulo unitário e forma um ângulo de 45 graus com o eixo x . O domínio foi discretizado com o emprego de várias malhas compostas respectivamente por 64x64, 128x128, 256x256 e 512x512 células, cada célula composta de dois elementos triangulares. Os sistemas de equações lineares não-simétricos foram resolvidos por dois métodos iterativos distintos, para que fossem avaliados os desempenhos relativos, o GMRES(k) e o Bi-CGSTAB. Para uma descrição desses métodos sugere-se a leitura de¹. A Tabela II apresenta um resumo do número de operações necessárias em cada passo desses algoritmos. Nessa tabela Av significa o número de produtos matriz por vetor (ou *matvec*, por acronismo), dot para os produtos internos e/ou normas, e saxpy para atualizações de vetores do tipo *saxpy*, de acordo com a terminologia BLAS.

Método	Av	dot	saxpy	Memória
GMRES(k)	1	k/2+2	k+4	kn+k2/2
Bi-CGSTAB	2	4	12	4n

Tabela II. Operações por iteração e demanda de memória (n é a ordem do sistema)

As várias malhas foram processadas em uma única CPU de um CRAY Y - MP, considerando um condicionamento diagonal à esquerda, que apresenta a complexidade computacional de uma *saxpy* adicional. As Tabelas III e IV listam os tempos de CPU em segundos, o número de iterações necessárias para se alcançar a tolerância desejada na solução e o número de Mflops (milhões de operações de ponto flutuante por segundo) para cada execução. Considerou-se a convergência alcançada para uma norma relativa dos resíduos inferior a 10^{-6} . O algoritmo GMRES(k) empregou um sub-espço de Krylov contendo cinco (5) vetores.

Os resultados obtidos demonstram que ambos algoritmos apresentam basicamente o mesmo desempenho. Foi observado que a maior parte do tempo de CPU é consumida na rotina *matvec*, a qual emprega a técnica EPE Livre de Matrizes. Para o problema de maior dimensão outros experimentos foram realizados em uma única CPU do supercomputador CRAY C90. Os resultados encontram-se listados na Tabela V, onde pode ser observado que para ambos algoritmos foi alcançado um alto desempenho.

É interessante comparar a demanda de memória entre a abordagem aqui apresentada e a da técnica EPE convencional, onde as matrizes de elemento são calculadas e armazenadas. A Tabela VI apresenta os números para essa comparação, onde 1 Mword corresponde a 1×10^6 palavras de 64bits.

Malha	CPU(s)	Iterações	Mflops
64 × 64	2.865	184	114.4
128 × 128	19.211	326	119.7
256 × 256	138.094	595	121.1
512 × 512	1016.480	1107	122.3

Tabela III. Resultados para o algoritmo GMRES(5)

Malha	CPU(s)	Iterações	Mflops
64 × 64	1.959	74	112.0
128 × 128	14.156	146	120.1
256 × 256	121.037	316	121.0
512 × 512	1022.055	675	122.2

Tabela IV. Resultados para o algoritmo Bi-CGSTAB

Método	CPU(s)	Mflops
GMRES(5)	423.428	293.5
Bi-CGSTAB	426.480	291.5

Tabela V. Informações da solução para a malha 512 × 512 no CRAY C90

Método	EPE Livre de Matrizes	EPE
GMRES(5)	9.38	12.32
Bi-CGSTAB	9.38	12.58

Tabela VI. Demanda de memória em Mwords para a malha 512 × 512

Deve ser observado que em ambos algoritmos a técnica EPE convencional requer cerca de 30% de memória a mais do que a alternativa EPE Livre de Matrizes. Entretanto, ambas as técnicas EPE são muito mais atraentes do que uma solução direta com armazenamento em perfil, a qual necessitaria, nesse caso, de uma área de memória de 269.22 Mwords, apenas para armazenar os termos da matriz global. Para finalizar, utilizando as facilidades de *autotasking* disponíveis no sistema C90, os códigos foram paralelizados empregando a estratégia delineada na Seção 1. Os custos de ambas as soluções, GMRES(5) e Bi-CGSTAB, foram dominados pelas operações *matvec* e apresentaram o mesmo desempenho global. A Tabela VII apresenta os ganhos relativos para a execução em paralelo, a eficiência computacional e a taxa de desempenho em Gflops para a solução empregando o Bi-CGSTAB em uma máquina

de 16 CPUs, respectivamente para toda a execução (job) e para a rotina *matvec*.

Em todas as medições, quer em modo vetorial e/ou paralelo, o desempenho verificado encontra-se no entorno de um terço do desempenho de pico do processador. Isso é muito razoável, uma vez que todos os algoritmos foram codificados em FORTRAN 77, fazendo uso das diretivas de compilação disponíveis. Nenhuma otimização de código foi realizada além daquela efetuada pelo compilador propriamente dito.

Tarefa	Ganho	Eficiência	Gflops
job	10.30	0.64	3.023
matvec	14.45	0.90	4.241

Tabela VII. Informações da solução em paralelo para a malha 512×512 no CRAY C90 - Bi-CGSTAB

CONCLUSÕES

Um novo melhoramento da técnica EPE foi proposto neste trabalho. Este se baseia na exploração cuidadosa da deficiência de posto das matrizes de elemento, resultando em uma economia considerável de operações de ponto flutuante e memória. Além disso, no caso em que as matrizes de elemento podem ser avaliadas explicitamente, um método EPE Livre de Matrizes pode ser desenvolvido, onde não há necessidade de armazenar as matrizes de elemento. Este enfoque requer um mínimo de memória, e fornece uma estratégia adequada para problemas de grandes dimensões, onde o emprego de métodos iterativos de solução é fortemente recomendado. Os experimentos numéricos efetuados mostram que a estratégia proposta permite a solução de grandes treliças espaciais, geometricamente não-lineares, em alguns segundos nas estações de trabalho correntes. Problemas não-simétricos de grande porte podem ser solucionados com desempenho da ordem de Gflops em um supercomputador vetorial/paralelo CRAY C916.

AGRADECIMENTOS

Os recursos computacionais do CRAY Y-MP 2E/232 foram oferecidos pelo Centro Nacional de Supercomputação da Universidade Federal do Rio Grande do Sul, Brasil. Agradecimentos especiais à Sra. Denise G. Ewald do Centro Nacional de Supercomputação, pelo apoio inestimável. Os autores gostariam também de manifestar sua gratidão ao Sr. Tibério Bulhões, da CRAY Brasil, pelo auxílio na paralelização de nosso código. Agradecimentos especiais ao Sr. S. Kremer, que nos forneceu acesso ao CRAY C90 em Eagan, Minnesota, Estados Unidos da América.

REFERÊNCIAS

1. R. Barret, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, “*Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods*”, SIAM, Philadelphia, USA, (1994).
2. A.N. Brooks, T.R.J. Hughes, “Streamline Upwind/Petrov-Galerkin Formulation for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equation”, *Comp. Meth. Appl. Mech. Engng.*, Vol. **32**, pp. 199–259, (1982).
3. A.L.G.A. Coutinho, J.L.D. Alves, N.F.F. Ebecken, “A Study of Implementation Schemes for Vectorized Sparse EBE Matrix-Vector Multiplication”, *Advances in Engineering Software and Workstations*, Vol. **13**, pp. 130–34, (1991).
4. P.R.B. Devloo, F.A. Menezes, “On the Development of a Finite Element Approximation of Two-Dimensional Elasticity”, *XV Congresso Ibero Latino-Americano sobre Métodos Computacionais em Engenharia*, Belo Horizonte, Brasil, Vol. **2**, pp. 1616–1625, (1994).
5. L.J. Hayes, P.R.B. Devloo, “A Vectorized Version of a Sparse Matrix-Vector Multiply”, *Int. J. for Num. Meth. in Engineering*, Vol. **23**, pp. 1043–56, (1986).
6. C. Hirsch, “*Numerical Computation of External and Internal Flows*”, Vol. **1**, John Wiley and Sons, Chichester, UK, (1989).
7. T.J.R. Hughes, R.M. Ferencz, J.O. Hallquist, “Large-Scale Vectorized Implicit Calculations in Solid Mechanics on a CRAY X-MP/48 Utilizing EBE Preconditioned Conjugate Gradients”, *Comp. Meth. Appl. Mech. Engng.*, Vol. **61**, pp. 215–48, (1987).
8. M. Papadrakakis, “*Solving Large-Scale Problems in Mechanics*”, John Wiley and Sons, Chichester, UK, (1993).