

APLICAÇÕES DO PARALELISMO NO MÉTODO DOS ELEMENTOS FINITOS

Elias C. Carrijo[†] e João B. de Paiva*

[†] Universidade Católica de Goiás
Departamento de Matemática e Física
Av. Universitária, Nº 1440, Setor Universitário, CEP 74605-010
Goiânia, Goiás – Brasil
e-mail: calixto@ucg.br

*Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia de Estruturas
Av Dr Carlos Botelho, 1465 CEP 13560-250
São Carlos, SP - Brasil

Palavras chaves: Método dos Elementos Finitos, paralelismo, matriz de rigidez, abordagem nodal.

Resumo *Neste trabalho apresenta-se uma proposta de paralelismo para o cálculo da matriz de rigidez no Método dos Elementos Finitos. No algoritmo convencional as contribuições dos elementos para a rigidez da estrutura são adicionadas elemento por elemento, utilizando-se um laço sobre o número total de elementos finitos da malha. Apresenta-se neste trabalho uma solução alternativa com abordagem nodal, cuja formulação apresenta ampla aplicação no paralelismo. Utiliza-se uma máquina com memória distribuída e biblioteca de troca de mensagens PVM.*

1. INTRODUÇÃO.

A análise estrutural foi extremamente beneficiada com o surgimento dos computadores e dos métodos numéricos. Problemas com solução analítica difícil passaram a dispor de ferramentas importantes na sua resolução. Com o desenvolvimento dessas máquinas, a uma taxa de dez vezes a cada cinco anos, pode-se tratar problemas com uma complexidade maior. Entretanto, as dificuldades encontradas no desenvolvimento dessas máquinas, tornaram evidentes as limitações arquiteturais das mesmas. Com a introdução do paralelismo esse gargalo das máquinas sequenciais foi superado, possibilitando grandes avanços na obtenção de máquinas com elevada capacidade de processamento. NOOR (1997) mostra a evolução das arquiteturas de computadores e as possibilidades que emergem dessa evolução.

Por outro lado, para um aproveitamento adequado desses equipamentos, é necessária uma mudança no modo de elaboração de programas. Os algoritmos que são usados com sucesso em máquinas sequenciais podem não apresentar a mesma eficiência em máquinas com arquitetura paralela. Diante dessa situação pesquisadores em análise numérica tem se dedicado à tarefa de adaptar e por vezes propor algoritmos numéricos que melhor se adequem a essas arquiteturas. Considerando-se então os métodos numéricos, mais especificamente o Método dos Elementos Finitos, quando se procura desempenho computacional com arquiteturas paralelas, os esforços serão direcionados a paralelizar suas etapas típicas, ou seja cálculo da matriz de rigidez, condições de contorno, resolução do sistema de equações e obtenção de esforços.

Os esforços empreendidos com intuito de se conseguir melhores resultados para os métodos numéricos em máquinas com arquitetura paralela, podem ser observados no trabalho versando sobre esse assunto. Nota-se uma enorme variedade de propostas, desde a paralelização das etapas típicas já citadas, até abordagens em subestruturas e decomposições de domínio. Para decomposições de domínio cita-se LAW (1986), CARTER (1989), EL-SAYED e HSI-UNG (1990) e para etapas típica tem-se FARAH e WILSON (1988), CHIEN e SUN (1989), GOEHLICH (1989), ADELI e KAMAL (1992), SCHMIT e LAI (1994), REZENDE (1995).

O presente trabalho apresenta uma implementação em paralelo para máquinas com memória distribuída, utilizando a biblioteca PVM (parallel virtual machine) para uma etapa típica do Método dos Elementos Finitos, o cálculo da matriz de rigidez da estrutura. Essa idealização foi originalmente proposta por REZENDE (1995) para máquinas com memória compartilhada.

2. CÁLCULO EM PARALELO DA MATRIZ DE RIGIDEZ DA ESTRUTURA.

Como se trata de uma das etapas fundamentais do MEF, vários trabalhos tem sido apresentados com intuito de propor um algoritmo eficiente para o cálculo em paralelo da matriz de rigidez da estrutura como LAW (1986), CHIEN e SUN (1989) e ADELI e KAMAL (1992). Esses trabalhos envolvem desde o cálculo paralelo da matriz de rigidez, até o processamento concorrente de toda estrutura com trocas de informações entre os processos em execução.

REZENDE (1995) apresenta um algoritmo alternativo que aloca os coeficientes de rigidez de cada elemento da malha para a estrutura através de um *loop* sobre os *pontos nodais da estrutura (abordagem nodal)*. Nesse algoritmo a matriz é calculada *linha por linha*, ou seja, *pa-*

ra cada ponto nodal da malha calculam-se os coeficientes de rigidez de todos os graus de liberdade correspondentes a esse ponto. No caso de programas para processamento em paralelo, esse algoritmo apresenta a vantagem de não incorrer no problema de dependência de dados. Desse modo, para se calcular a matriz de rigidez da estrutura basta dividir os pontos nodais da malha entre os processos em execução.

Para melhor compreensão deste algoritmo pode-se recorrer ao exemplo da figura 1, onde supõe-se a existência de dois graus de liberdade por nó. Nesse caso para o ponto nodal 4, o algoritmo calculará os coeficientes das linhas 7 e 8 da respectiva matriz de rigidez global, sendo que os elementos finitos que participarão deste processo serão os elementos 1, 2 e 3, ou seja, os elementos concorrentes a esse ponto nodal.

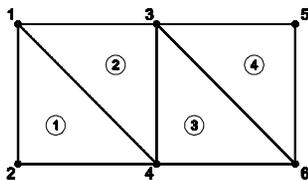
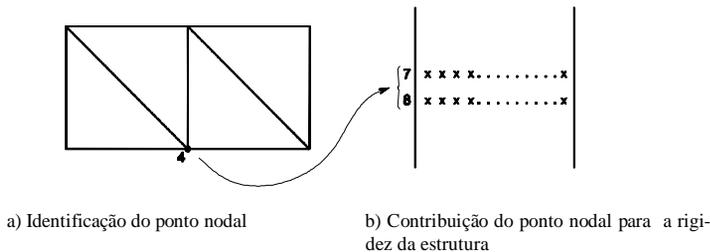


Figura 1. Exemplo de malha para elementos finitos triangulares

Esse procedimento está ilustrado na figura 2, onde se mostra qual a contribuição do respectivo ponto nodal para a rigidez da estrutura.



a) Identificação do ponto nodal b) Contribuição do ponto nodal para a rigidez da estrutura

Figura 2. Cálculo da matriz de rigidez com abordagem nodal

Uma vez que o problema da dependência de dados foi eliminado, então o cálculo de uma linha da matriz de rigidez é independente de outra linha. Por esse motivo o algoritmo proposto pode ser usado diretamente no paralelismo. Então, para uma máquina com memória distribuída e supondo-se a existência de dois processos em execução, pode-se, sem nenhuma dificuldade, alocar para um processo P1 os pontos nodais 1 a 3 e para outro processo P2 os pontos 4 a 6. No algoritmo 1 essa operação se resumiria em mudar os limites do *loop* na linha 13. Gráficamente pode-se representar esse procedimento através da figura 3:

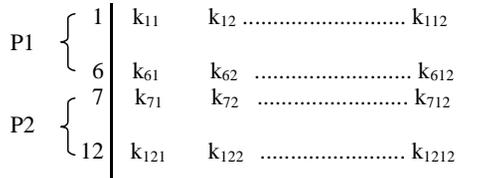


Figura 3 Cálculo da matriz de rigidez com dois processos concorrentes

Para se executar esse procedimento é necessário proceder a identificação dos elementos finitos que concorrem a um determinado ponto nodal. Essa identificação é efetuada com uma estrutura de dados auxiliar que corresponde às linhas 1 a 9 desse algoritmo alternativo, apresentado no quadro a seguir.

Algoritmo 1 Algoritmo alternativo de cálculo da matriz de rigidez

```

ALGORITMO ALTERNATIVO COM ABORDAGEM NODAL PARA  

CÁLCULO DA MATRIZ DE RIGIDEZ DA ESTRUTURA EM BANDA

NUMNP..... NÚMERO DE PONTOS NODAIS DA ESTRUTURA.
NUMEL..... NÚMERO DE ELEMENTOS DA ESTRUTURA.
NVARN..... NÚMERO DE VARIÁVEIS POR NÓ DO ELEMENTO.
NNOS..... NÚMERO DE NÓS DO ELEMENTO.
NP[I,J]..... INCIDÊNCIA DO ELEMENTO I.
NCONC(K)..... NÚMERO DE ELEMENTOS LIGADOS AO PONTO K.
ELEM_CONC[K,I]..... NÚMERO DO I-ÉSIMO ELEMENTO LIGADO AO PONTO K.
NO_CORRESP[K,I]..... NUMERAÇÃO LOCAL DO PONTO K PARA O I-ÉSIMO
ELEMENTO A ELE LIGADO.

1  ESTRUTURA DE DADOS AUXILIAR.
2  PARA I=1 ATÉ NUMEL FAÇA
3    PARA J=1 ATÉ NNOS FAÇA
4      P = NP(I,J)
5      NCONC(P) = NCONC(P)+1
6      ELEM_CONC(P,NCONC(P)) = I
7      NO_CORRESP(P,NCONC(P)) = J
8    FIM FAÇA
9  FIM FAÇA
10
11 CÁLCULO DA MATRIZ DA ESTRUTURA
12 NPAR=NVARN
13 PARA I=1 ATÉ NUMNP FAÇA
14   PARA J=1 ATÉ NCONC(I) FAÇA
15     ELEM=ELEM_CONC(IJ)
16     PON=NO_CORRESP(IJ)
17     CHAMA ROTINA PARA CALCULAR A MATRIZ DO ELEMENTO J
18     LINHA_BASE=NPAR*(NPAR-1)
19     PARA K=1 ATÉ NNOS FAÇA
20       PAUX=NP(ELEM,K)
21       COLUNA=NPAR*PAUX-(NPAR-1)
22     PARA M=1 ATÉ NPAR FAÇA
23       LINHA_LOCAL=NPAR*(PON-1)+M1
24       LINHA_GLOBAL=(LINHA_BASE-1)+M1
25     PARA N=1 ATÉ NPAR FAÇA
26       COLUNA_LOCAL=NPAR*(K-1)+N
27       COLUNA_GLOBAL=(COLUNA-1)+N

```

```

28          DIF=COLUNA_GLOBAL-LINHA_GLOBAL+I
29          SE DIF, MENOR QUE, 0 ENTÃO
30              MAT_G(LINHA_GLOBAL,DIF)=MAT_G(LINHA_GLOBAL,DIF)+
31                  MAT_L(LINHA_LOCAL,COLUNA_LOCAL)
32          FIM SE
33          FIM FAÇA
34          FIM FAÇA
35          FIM FAÇA
36          FIM FAÇA
37          FIM FAÇA
    
```

Para exemplificar esse procedimento, suponha-se a execução dessas instruções tomando-se como referência o nó 4 do exemplo da figura 1. O resultado dessa execução será:

Tabela 1. Estrutura auxiliar para nó 4.

Número do ponto nodal: 4		
Número de elementos concorrentes: 3		
NCONC(4) = 3		
Índice I	ELEM_CONC(4,I) Número do elemento I ligado ao ponto 4.	NO_CORRESP(4,I) Numeração local do ponto 4 do elemento I.
1	1	3
2	2	3
3	3	2

Essa identificação é realizada para todos os pontos nodais da estrutura. Assim, à medida que o algoritmo se posiciona em determinado ponto nodal, para calcular os coeficientes da matriz de rigidez da estrutura, todos os elementos concorrentes a esse ponto já estarão determinados e poderão ser utilizados nos cálculos restantes. Observe-se que, na linha 14 do algoritmo alternativo o *loop* é realizado até o número de elementos que concorrem a esse ponto nodal.

3. RESULTADOS

Os resultados apresentados neste trabalhos foram obtidos em um IBM SP2 com três nós de processamento, localizado no Centro de Informática de São Carlos. Cada nó possui um processador Pentium de 66 MHz e 256 MB de memória local. Para análise desse algoritmo recorre-se aos parâmetros usados para verificação de algoritmos paralelizados. Os conceitos normalmente usados são *speed-up* e *eficiência*, apresentados a seguir:

Speed-up

É definido como o resultado do quociente entre o tempo necessário para se executar um exemplo com um processador e o tempo para n processadores, ou seja:

$$\text{Speed-up} = \frac{\text{tempo para 1 processador}}{\text{tempo para n processadores}} \quad (1)$$

Eficiência

Este parâmetro informa o quanto um programa está paralelizado. É definido por:

$$\text{Eficiência} = \frac{100 \text{Speed-up}}{\text{número de processadores}} \quad (2)$$

Na verificação desse algoritmo foram executados dois exemplos com 200 (exemplo 1) e 300 (exemplo 2) pontos nodais, respectivamente. Cada elemento finito possui 3 (tres) pontos nodais e 6 (seis) variáveis por nó, totalizando 1200 e 1800 graus de liberdade respectivamente; trata-se de um elemento finito de casca plano sendo utilizado para se executar uma placa. A disposição dos elementos finitos está dada na figura 4, onde todos os elementos finitos mantêm a mesma orientação. Como se trata apenas de tomada de tempo, não se necessita mudar essa orientação.

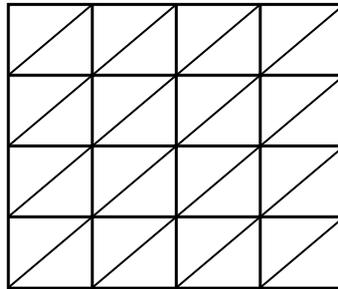


Figura 4. Disposição dos elementos finitos da malha.

Os resultados para estes exemplos estão dados nas figuras a seguir, onde estão mostrados os parâmetros *Speed-up* e *eficiência*

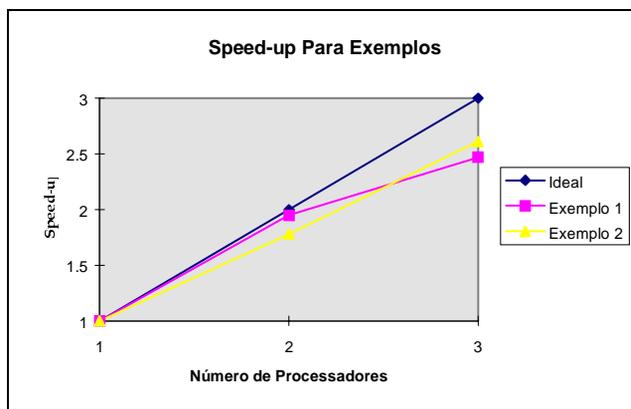


Figura 5. Speed-up para exemplos 1 e 2.

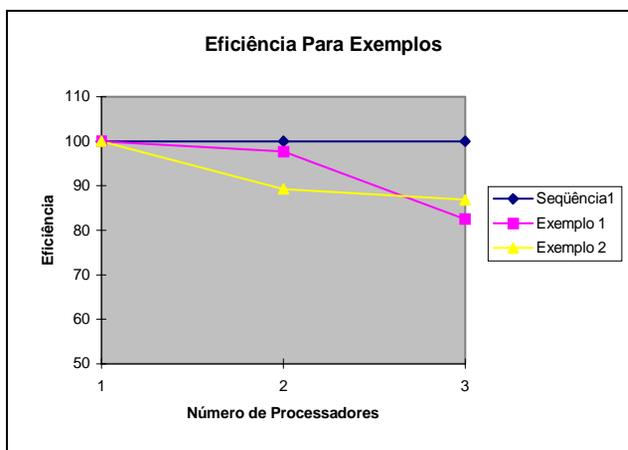


Figura 6. Eficiência para os exemplos 1 e 2.

Como pode ser observado, o desempenho do algoritmo tem uma melhora com o aumento da dimensão do problema. Isso pode ser explicado pelo esforço de paralelização inicial, que é constante para todos os exemplos, sendo que à medida que aumenta a dimensão da matriz aumenta o trabalho a ser efetuado em paralelo, com conseqüente melhora do desempenho do algoritmo.

No algoritmo proposto, um balanceamento de trabalho ruim seria verificado para as malhas onde alguns pontos nodais apresentassem elevada concentração de elementos finitos, e todos esses pontos estivessem sendo executados em um mesmo processador. Isso poderia ser evitado com uma disposição adequada da numeração da malha, sendo um trabalho específico de pré-processadores voltados para utilização no paralelismo.

4. CONCLUSÕES

Apresentou-se nesse trabalho um algoritmo alternativo para o cálculo em paralelo da matriz de rigidez de uma estrutura utilizando-se máquinas com memória distribuída e biblioteca de troca de mensagens PVM. Esse algoritmo apresenta vantagens para o paralelismo pelo fato de não incorrer no problema de dependência de dados, obtendo-se o paralelismo diretamente do loop mais externo de modo simples e objetivo. Pode ser usado tanto em máquinas com memória compartilhada, como em REZENDE (1995), quanto em memória distribuída, como neste trabalho. Os resultados dos exemplos para speed-up e eficiência indicam maior utilização do algoritmo, inclusive pela facilidade de implementação que o algoritmo possui.

5. BIBLIOGRAFIA

- [1] NOOR, A. K. (1997) , New computing systems and future high-performance computing enviroment and their impact on structural analysis and design. *Computers and Structures*, v. 64, n 1-4, pp. 1-30
- [2] LAW, H. K. (1986), A parallel finite element solution method. *Computers and Structures*, v. 23, n. 6, pp 845-858.
- [3] CARTER Jr, W. T.; SHAW, T. L.; LAW, H. K. (1989), A parallel finite element method and its prototype implementation on a hypercube. *Computers and Structures*, v. 31, n. 6, pp. 921-934.
- [4] EL-SAYED, M. E. M.; HSIUNG, C. K. (1990). Parallel finite element computation with separate substructures. *Computers and Structures*, v. 36, n. 2, pp. 261-265.
- [5] FARAH, C.; WILSON, E. (1988), A parallel active column solver. *Computers and Structures*, v. 28, n. 2, pp. 289-304
- [6] CHIEN, L. S.; SUN, C. T. (1989). Parallel processing techniques for finite element analysis of nonlinear large truss structures. *Computers and Structures*, v. 31, n. 6, pp. 1023-1029
- [7] GOELICH, D.; KOMZSIK, L.; FULTON, R. E. (1989). Application of a parallel equation solver to static fem problems. *Computers and Structures*, v. 31, n. 2. Pp. 121-129.

[8] ADELI, H.; KAMAL, O. (1992). Concurrent analysis of large structures – I. Algorithms. *Computers and Structures*, v. 42, n. 3, pp. 413-424.

[9] SCHMIT, L. A.; LAY, YE-CHEN, (1994). Structural optimization based on preconditioned conjugated gradient analysis methods. *International Journal for Numerical Methods in Engineering*, v. 37, n. 6, Mar. 30, pp. 943-964.

[10] REZENDE, M. N. (1995). *Processamento paralelo aplicado em análise estrutural*. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.