

**DECOMPOSIÇÃO DE DOMÍNIO AUTOMÁTICA DE MALHAS DO MÉTODO DE
ELEMENTOS FINITOS NO HIPERCUBO INTEL iPSC/860**

Manoel Rodrigues Justino Filho, Dsc.

Nelson Francisco Favilla Ebecken, D.Sc.

COPPE/UFRJ - Programa de Eng. Civil

Caixa Postal 68506 - CEP 21945-970 Rio de Janeiro-RJ

Email : coc35001@ufrj.bitnet

RESUMO

Apresenta-se um algoritmo heurístico de decomposição de domínio automática de malhas do método de elementos finitos, dirigido para métodos diretos de solução esparsos. O presente algoritmo produz carga computacional equilibrada entre processadores de um arquitetura com memória distribuída, com reduzido custo de comunicação para transferência de mensagens. Resultados numéricos da decomposição de domínio são apresentados para demonstrar a precisão e eficiência do algoritmo no supercomputador hipercubo Intel iPSC/860.

ABSTRACT

A heuristic algorithm for automatic domain decomposition of an irregular finite element mesh is presented, directed to direct sparse solution methods. The present algorithm produces computational workload balance between processors in a distributed memory machine, with reduced communication costs for message passing. Numerical results are presented to demonstrate the accuracy and efficiency of the algorithm on the Intel iPSC/860 hypercube supercomputer.

I Introdução

A decomposição de domínio automática é uma fase importante de subdivisão de uma malha irregular de elementos finitos em p subdomínios, onde p é o número de processadores alocados numa arquitetura paralela com memória distribuída.

Cada subdomínio da decomposição corresponde a uma subestrutura contendo nós e elementos internos e de fronteira. Na figura 1 tem-se uma decomposição de uma malha de elementos finitos em quatro subdomínios.

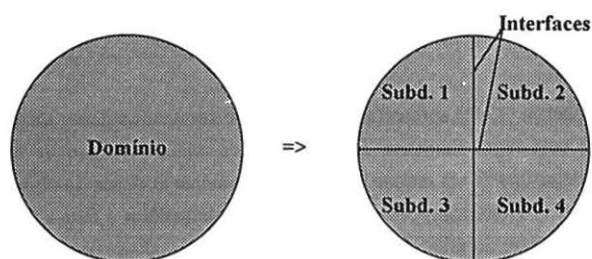


Figura 1 - Decomposição de domínio de uma malha de elementos finitos

Um mapeamento resultante da decomposição de domínio é dito ótimo quando produz carga computacional equilibrada entre os processadores alocados e a solução paralela é eficiente em relação à melhor solução sequencial do problema.

A decomposição ótima de uma malha irregular de elementos finitos pertence à classe de problemas do tipo NP-completo (não determinístico polinomial completo) de otimização da análise combinatória. Existem métodos heurísticos iterativos e diretos que fornecem soluções aproximadas para o problema de decomposição de domínio de malhas de elementos.

O método iterativo heurístico "simulated annealing" é utilizado por Kirkpatrick et alii (1983), para obter a solução aproximada de classe de problemas NP-completos da análise combinatória.

Flower et alii (1987) utiliza o método iterativo "simulated annealing" na decomposição de domínio de malhas de elementos finitos (MEF), cuja solução fica próxima da ótima, na faixa de 10 e 20% para propósitos práticos.

Farhat (1988) apresenta um método direto automático de decomposição de MEF, cujo critério heurístico é baseado no conceito de peso nodal mínimo e adjacência entre subdomínios consecutivos. O peso nodal é definido como sendo o número de elementos ainda não selecionados conectados ao nó. Os elementos conectados a um nó com peso nodal mínimo são selecionados primeiro.

Al-Nasra and Nguyen (1991) implementam uma decomposição de domínio automática de MEF, que também utiliza o conceito de peso nodal mínimo, sendo que a expansão dos subdomínios é controlada através da posição das coordenadas dos nós na malha.

Malone (1987) implementa uma decomposição de domínio automática de MEF para arquiteturas hipercubo, onde os elementos de cada subdomínio são selecionados através de uma reordenação em ordem ascendente da renumeração dos nós, após a utilização de um algoritmo de minimização de banda nodal.

Justino Filho e Ebecken (1992) apresentam um algoritmo automático direto de decomposição de MEF, que utiliza uma regra heurística de desempate de mínimo número de nós de interface, adjacência entre elementos de um mesmo subdomínio e máxima distância do último nível do grafo de elementos da MEF. Neste algoritmo a expansão dos subdomínios é feita a partir do elemento inicial do pseudo-diâmetro do grafo de elementos da MEF, dirigindo-se para o final deste pseudo-diâmetro.

Os algoritmos diretos automáticos de decomposição de domínio de MEF, que foram discutidos anteriormente, são adequados quando utilizados em programas que utilizem métodos diretos de solução de equações do tipo banda constante ou variável.

II Regras heurísticas para decomposição de domínio de MEF

Os métodos diretos automáticos de decomposição de domínio de MEF utilizam regras heurísticas para atender os seguintes objetivos:

- Carga computacional equilibrada entre processadores. Portanto cada processador deve ter uma distribuição equilibrada de nós e elementos;
- Número mínimo de nós de fronteira entre subdomínios adjacentes. Isto reduz a penalização de sincronização entre processadores e também minimiza a transferência de mensagens entre processadores de arquitetura do tipo hipercubo;
- Adjacência entre os elementos de um subdomínio, sempre que possível. Um subdomínio muito fragmentado possui muitos nós de fronteira;

- Minimização da largura de banda para os métodos de solução de banda constante ou variável, ou do preenchimento para os métodos de solução esparsos gerais;
- O tempo de execução do algoritmo de decomposição de domínio deve ser pequeno em relação ao tempo total da análise. Isto permite que o algoritmo seja executado no modo serial;
- O resultado do algoritmo de decomposição, para malhas irregulares quaisquer de elementos finitos, faz com que a versão paralela do programa de elementos finitos seja eficiente, em termos de tempo de execução, em relação à melhor versão sequencial do programa, rodando num único processador da máquina paralela.

III Algoritmo de decomposição de domínios de MEF para métodos esparsos

O algoritmo de decomposição proposto por Justino Filho e Ebecken (1992) e os demais descritos na introdução são mais adequados para métodos diretos de solução de equações, do tipo banda constante ou variável e o método frontal.

Para os métodos de solução esparsos gerais e o método multifrontal, utiliza-se o Algoritmo 1, apresentado a seguir, de acordo com Justino Filho (1993).

Definição complementar de variáveis:

- p - número de subdomínios ou processadores;
- i - subdomínio ou processador;
- ne - número de elementos presente na MEF;
- m - número de nós presentes na MEF;
- iel - variável de controle do número de elementos da MEF;
- nep_i - número de elementos do subdomínio ou processador i .
- $maxnep$ - máximo número de elementos por processador

Algoritmo 1 Decomposição de domínio automática de MEF

begin

Inicialização da estrutura de dados da decomposição de domínio;

Obtenção do grafo de elementos da MEF (lista de adjacências dos elementos);

Cálculo da lista de elementos conectados a cada nó da MEF;

Ordenação nodal da MEF pelo método de grau mínimo;

Pós-ordenação nodal topológica dos nós na árvore de eliminação;

Faz a ordenação dos elementos ("quick-sort") da MEF em ordem ascendente dos nós pós-ordenados anteriormente;

{ "Loop" sobre o número de subdomínios }

$iel = 0$;

$maxnep = ne/p + \min(1, \text{mod}(ne/p))$;

for $i = 1$ **to** p **do**

$nep_i = 0$ {Número de elementos do subdomínio ou processador i }

while ($nep_i \leq maxnep$ **and** $iel \leq ne$) **do**

Seleciona o próximo elemento ordenado pelo "quick-sort", para o subdomínio i ;

$nep_i = nep_i + 1$;

$iel = iel + 1$

end-while;

end-for;

{Mapeamento dos nós entre os subdomínios}

Os nós internos aos subdomínios são mapeados nos subdomínios correspondentes;

Os nós de fronteira adjacentes a dois ou mais subdomínios são atribuídos ao subdomínio adjacente com menor número de nós mapeados, de modo a garantir que cada subdomínio tenha aproximadamente m/p nós alocados;

end.

O algoritmo 1 é executado no modo serial no "host" 80386 do hipercubo iPSC/860.

Após a decomposição de domínio, o "host" envia para cada processador alocado a estrutura de dados associada à decomposição de domínio.

IV Resultados numéricos

A plataforma fixa hipotética, descrita na tese de Justino Filho (1993), é utilizada para os resultados numéricos da decomposição de domínio. Na tese anterior existem resultados da decomposição de domínio para outros modelos de plataformas fixas.

A geometria do modelo da plataforma está apresentada na figura 2. Esta plataforma possui quatro pernas, 75 metros de altura, sendo que as dimensões transversais variam de 50 metros quadrados no topo e 58 metros quadrados na base da plataforma.

O estaqueamento é simulado por molas nodais desacopladas com comportamento elástico linear.

O modelo estrutural da MEF é composto por 2436 elementos de pórtico espacial e 2141 nós. O número total de equações do modelo é de 12846 e a matriz de rigidez global da estrutura é formada por 219525 termos não-nulos, após a reordenação nodal pelo método do grau mínimo. O refinamento utilizado na MEF, permite uma determinação mais precisa das cargas de colapso de estrutura, pelo método dos elementos finitos.

Na tabela 1 tem-se o resultado da decomposição de domínio, utilizando-se oito processadores na análise.

Tabela 1 Decomposição de domínio da MEF da figura 1 com 8 processadores

| Subdomínio | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Núm. de elem. | 305 | 305 | 305 | 305 | 305 | 305 | 305 | 301 |
| Núm. elem. interface | 64 | 52 | 61 | 60 | 69 | 71 | 47 | 68 |
| Núm. de nós | 265 | 268 | 268 | 272 | 270 | 267 | 267 | 264 |
| Núm. de nós interface | 21 | 17 | 25 | 22 | 26 | 21 | 20 | 22 |

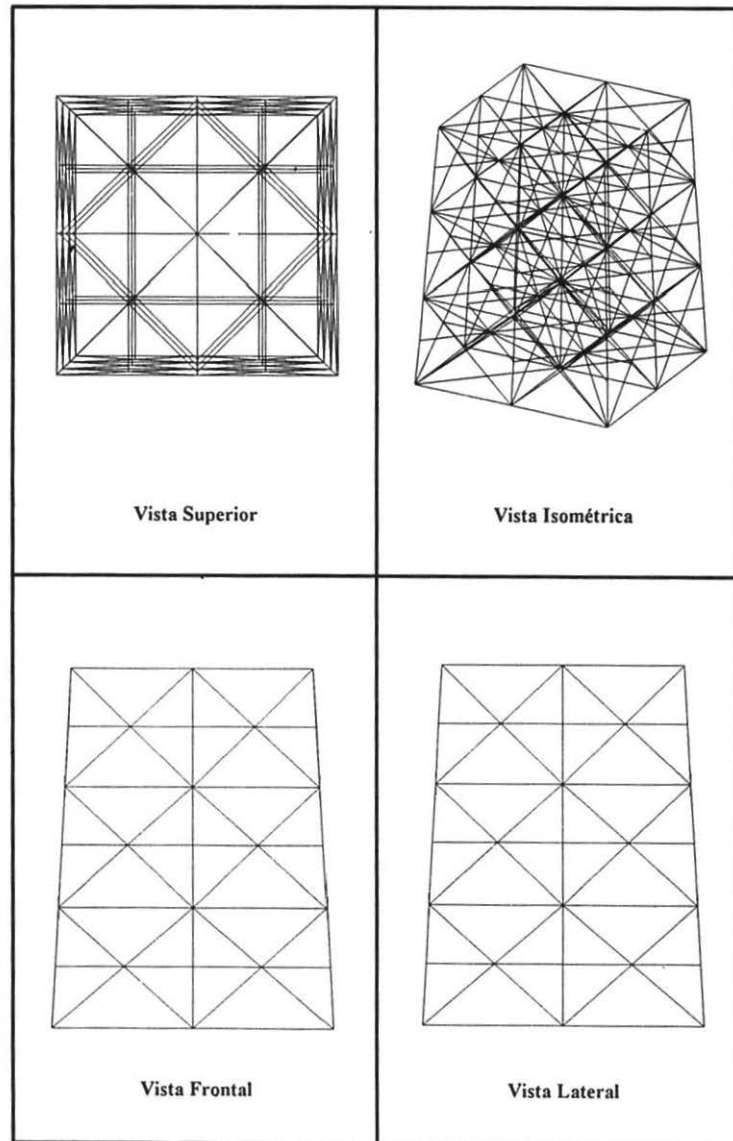


Figura 2 Plataforma fixa hipotética.

Analisando-se os resultados apresentados nas tabela 1, conclui-se que o algoritmo de decomposição de domínio utilizado produz uma distribuição equilibrada de nós e elementos entre os subdomínios. Também o número de nós de fronteira, em cada subdomínio, é pequeno em relação ao número de nós de cada subdomínio.

Na tabela 2 tem-se os tempos de execução da fase de pré-processamento da análise, executada no modo sequencial. O tempo de solução da fase de decomposição de domínio foi determinado para oito subdomínios.

Tabela 2 Tempos de execução (em segundos) da fase de pré-processamento sequencial da MEF da figura 2

| Fase | i80386 | i860 |
|---------------------------|--------|-------|
| Leit. e impr. geometria | 61.25 | 19.69 |
| Reord. nodal grau mínimo | 13.47 | 0.93 |
| Preench. da matriz global | 0.48 | 0.05 |
| Leit. e impr. de cargas | 66.90 | 20.01 |
| Decomp. de domínio | 91.60 | 6.04 |
| Alocação e carga do cubo | 2.38 | ---- |

Nota-se, nos tempos de execução da tabela 2, que a fase de pré-processamento executada sequencialmente, alocando-se um processador i860, é muito mais rápida do que quando executada no "host" i80386.

Nas fases de leitura e impressão de dados de geometria e cargas, a diferença entre os tempos de execução do i860 e do i80386 é menor do que as outras fases, por causa das operações de I/O envolvidas.

Na tabela 3 tem-se o tempo de execução de uma iteração da fase de solução de análise, utilizando-se 1, 2, 4 e 8 processadores.

Os tempos de execução foram obtidos com a função `dclock()` da biblioteca do iPSC/860, no processador supervisor responsável pelo controle da análise e pela comunicação com o "host".

Tabela 3 Tempos de execução (em segundos) de uma iteração típica de solução da análise da MEF da figura 2

| Fase | serial | 1 proc. | 2 proc. | 4 proc. | 8 proc. |
|---------------------------|--------|---------|---------|---------|---------|
| Mont. matriz global | 3.25 | 3.70 | 1.90 | 1.00 | 0.52 |
| Fator. multifrontal | 3.20 | 3.30 | 2.00 | 1.30 | 0.98 |
| Controle de desloc. | 1.24 | 1.20 | 0.98 | 0.82 | 0.71 |
| Cálc. forças int. resist. | 1.95 | 2.00 | 1.10 | 0.67 | 0.51 |

Nota-se na tabela 3 que os tempos de solução da versão paralela, rodando em um processador, estão bastante próximos do algoritmo sequencial, rodando também em um processador do cubo.

O "speed-up" e a eficiência são determinados pelas equações (1) e (2), respectivamente, a seguir :

$$S_p = T_{seq}/T_p \quad (1)$$

$$E_p = S_p/p = T_{seq}/(pT_p) \quad (2)$$

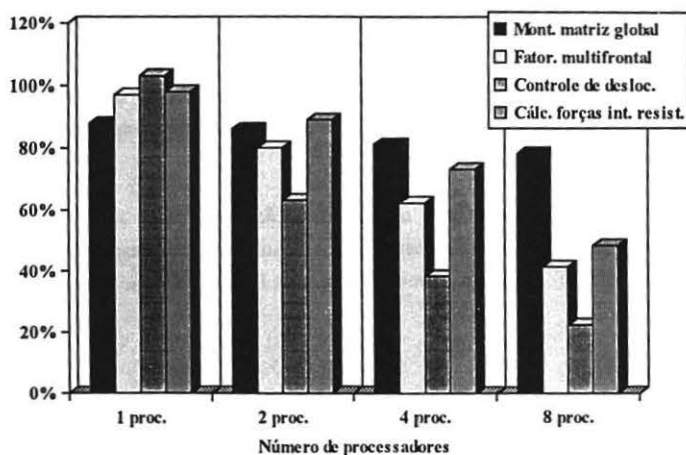
onde p é o número de processadores utilizados, T_{seq} é o tempo conseguido pela melhor versão sequencial do algoritmo, rodando num processador, e T_p é o tempo de solução gasto pelo algoritmo paralelo utilizando p processadores.

Na tabela 4 tem-se o "speed-up", S_p , e a eficiência, E_p , para a versão paralela do programa, para uma iteração típica da fase de solução do programa de elementos finitos (Justino Filho, 1993).

Tabela 4 "Speed-up" e eficiência de uma iteração típica da análise da MEF da figura 2

| Fase | 1 proc. | | 2 proc. | | 4 proc. | | 8 proc. | |
|---------------------------|---------|-------|---------|-------|---------|-------|---------|-------|
| | S_p | E_p | S_p | E_p | S_p | E_p | S_p | E_p |
| Mont. matriz global | 0.88 | 0.88 | 1.71 | 0.86 | 3.25 | 0.81 | 6.25 | 0.78 |
| Fator. multifrontal | 0.97 | 0.97 | 1.60 | 0.80 | 2.46 | 0.62 | 3.27 | 0.41 |
| Controle de desloc. | 1.03 | 1.03 | 1.27 | 0.63 | 1.51 | 0.38 | 1.75 | 0.22 |
| Cálc. forças int. resist. | 0.98 | 0.98 | 1.77 | 0.89 | 2.91 | 0.73 | 3.82 | 0.48 |

Observa-se na tabela 4, que o "speed-up" e a eficiência da implementação paralela diminuí com o aumento do número de processadores. Isto é explicado pelo alto custo de comunicação envolvido com o aumento do número de processadores utilizados.

**Gráfico 1** Eficiência de uma iteração típica da análise da MEF da figura 1

No entanto, os custos de comunicação envolvidos referem-se ao hardware protótipo atual do iPSC/860, que possui baixa taxa de transferência de dados de 2.8 Mbytes por segundo no pico, em relação ao seu alto potencial de cálculo aritmético.

No gráfico 1 tem-se a eficiência do algoritmo para uma iteração típica de um passo incremental da análise.

Na nova máquina comercial Paragon da Intel (ver Meurant, 1992), a taxa de transmissão de dados é da ordem de 200 Mbytes por segundo no pico, ou seja, os custos de comunicação ficam bastante reduzidos neste hardware.

A fatoração multifrontal é a fase mais difícil de se conseguir eficiência na análise. A eficiência desta fase é igual a 41%, para oito processadores, que está aceitável dentro do objetivo final da redução do tempo total de solução da análise não-linear.

Os resultados obtidos nas tabelas 3 e 4 reforçam o potencial de paralelização do método dos elementos finitos em máquinas paralelas com memória distribuída.

V Conclusões

A solução de problemas complexos de engenharia exige a utilização de cálculo computacional intensivo.

As máquinas vetoriais e as máquinas paralelas com memória compartilhada ou distribuída, têm possibilitado a solução de grandes problemas de engenharia, com tempos de solução bem menores que os obtidos em máquinas sequenciais convencionais.

Dentro do contexto do método dos elementos finitos, os supercomputadores paralelos e vetoriais têm permitido a análise não-linear de modelos estruturais com a solução de grandes sistemas de equações.

Destaca-se neste trabalho a importância da decomposição de domínio automática de malhas irregulares de elementos finitos, voltada para métodos de solução diretos esparsos, para viabilizar a implementação paralela de um programa de elementos finitos, em arquiteturas com memória distribuída.

VI Bibliografia

- Al-Nasra, M. and Nguyen, D. T. (1991), "An algorithm for domain decomposition in finite element analysis", *Computer & Structures*, Vol. 39, No. 3/4, pp. 277-289.
- Ashcraft, C. (1987), "A vector implementation of the multifrontal method for large symmetric positive definite linear systems", Technical Report ETA-TR-51, Engineering Technology Applications Division, Boeing Computer Services, Seattle, Washington.
- Farhat, C. (1988), "A simple and efficient automatic FEM domain decomposer", *Computer & Structures*, Vol. 28, No. 5, pp. 579-602.

- Flower, J. , Otto , S. and Salama, M. (1987), "Optimal mapping of irregular finite element domains to parallel processors", in *Parallel Computations and Their Impact on Mechanics*, presented at the Winter Annual Society of Mechanical Engineers, AMD, Vol. 86, pp. 239-250, Boston, Massachusetts.
- Intel Corporation (1989), "i860 64-bit microprocessor programmer's reference manual", Intel Corporation, Santa Clara, California.
- Intel Corporation (1991a), "iPSC/2 and iPSC/860 programmer's reference manual", Intel Corporation, Santa Clara, California.
- Intel Corporation (1991b), "iPSC/2 and iPSC/860 user's guide", Intel Corporation, Santa Clara, California.
- Intel Corporation (1992), "iPSC i860 Fortran compiler user's guide", Intel Corporation, Santa Clara, California.
- Justino Filho, M. R. and Ebecken, N. F. F. (1992), "Ultimate strength analysis of offshore structures on the Intel iPSC/860 hypercube", *International Conference on Parallel Computing and Transputers Applications*, IOS Press/CIMNE, Barcelona, Part II, pp. 1089-1097.
- Justino Filho, M. R. (1993), "Análise do colapso de estruturas offshore no supercomputador hpercubo Intel iPSC/860", Tese de Doutorado, Programa de Engenharia Civil, COPPE/UFRJ, Rio de Janeiro, RJ, Maio.
- Kirkpatrick, S. , Gelatt Jr , C. D. and Vecchi, P. (1983), "Optimization by simulated annealing", *Science*, Vol. 220 , No. 4598, pp. 671-680.
- Malone, J. G. (1987), "Automated decomposition of finite element meshes for hypercube computers", In *Parallel Processing for Scientific Computing*, Proceedings of the Third SIAM Conference on Parallel Processing for Scientific Computing, Los Angeles, California.
- Meurant, G. (1992), "The evolution of scientific computing on paralled computers", I Escola de Computação Científica de Alto Desempenho, LNCC/CNPq, Rio de Janeiro, Março.