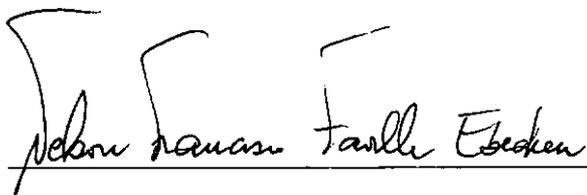


**ANÁLISE DO COLAPSO DE ESTRUTURAS OFFSHORE NO
SUPERCOMPUTADOR HIPERCUBO INTEL iPSC/860**

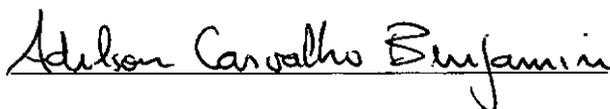
Manoel Rodrigues Justino Filho

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

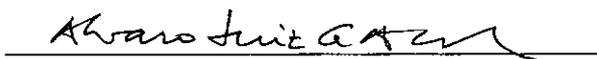
Aprovada por :



Prof. Nelson Francisco Favilla Ebecken, D.Sc.
(Presidente)



Eng. Adilson Carvalho Benjamin, D. Sc.



Prof. Alvaro L. G. A. Coutinho, D. Sc.



Prof. Edison Castro Prates de Lima, D. Sc.



Eng. José L. Drummond Alves, D. Sc.

RIO DE JANEIRO, RJ-BRASIL

MAIO DE 1993

JUSTINO FILHO, MANOEL RODRIGUES

Análise do Colapso de Estruturas Offshore no Supercomputador Intel iPSC/860

(Rio de Janeiro) 1993

ix, 115 p., 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Civil, 1993)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Computação Paralela
2. Arquiteturas Hipercubo
3. Análise não-linear de estruturas
4. Plataformas Offshore
5. Plasticidade Computacional
6. Decomposição de Domínio

I. COPPE/UFRJ II. Título (série)

*A minha esposa com carinho pelos bons
momentos que temos vivido juntos...*

AGRADECIMENTOS

Ao Prof. Nelson F. F. Ebecken pela valiosa orientação e pelo apoio demonstrado em todas as fases da tese.

À CAPES e ao RHAE-INFO do CNPq (parcial) pelo suporte financeiro desta pesquisa.

Aos professores pesquisadores na área de offshore do Programa de Engenharia Civil da COPPE, em especial aos colegas Alvaro L. G. A. Coutinho, Breno Pinheiro Jacob, Gilberto Ellwanger, José Luis Drummond Alves e Luiz Landau pelo apoio recebido.

Aos colegas do Núcleo de Computação Paralela (NCP) da COPPE, em particular ao Eng. Francisco C. Mota pela ajuda na utilização do supercomputador iPSC/860.

Ao Eng. Rafael Ochoa Lopez "*in memorium*" pela ajuda inicial na implementação do elemento elasto-plástico e ao Eng. Mário Ribeiro da Silva pela ajuda na parte gráfica.

A todos os professores do Programa de Engenharia Civil (PEC) da COPPE pelos ensinamentos recebidos durante as fases de mestrado e doutorado.

Aos amigos do PEC da COPPE pelo bom espírito de convivência, dos quais sempre guardarei boas lembranças.

Ao amigo Nilo Scoralick pela confecção dos desenhos da tese.

Ao Eng. Adilson C. Benjamin pela grande ajuda recebida na parte de Offshore e de análise não-linear de estruturas.

Ao Eng. Benjamim Ernani Diaz pelas diversas sugestões e críticas construtivas recebidas durante a elaboração desta tese.

Ao Eng. Antonio Vellasco Filho pela ajuda recebida na parte de editoração eletrônica da tese.

Aos colegas da área de Infra-estrutura da Promon Engenharia Ltda pela amizade, incentivo e compreensão mostrados no período em que precisei ficar afastado para a realização do doutorado.

A minha esposa pelo grande apoio e incentivo demonstrado, também pela ajuda importante na digitação e verificação ortográfica dos capítulos desta tese.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências (D. Sc.)

ANÁLISE DO COLAPSO DE ESTRUTURAS OFFSHORE NO SUPERCOMPUTADOR HIPERCUBO INTEL iPSC/860

Manoel Rodrigues Justino Filho

MAIO, 1993

Orientador : Nelson Francisco Favilla Ebecken

Programa : Engenharia Civil

Apresenta-se neste trabalho uma implementação computacional paralela, em um supercomputador com memória distribuída, de um programa de elementos finitos para a análise do colapso de estruturas offshore reticuladas.

A implementação utiliza métodos de decomposição de domínio e a solução multifrontal paralela para solução de sistemas esparsos.

A análise não-linear incremental iterativa da estrutura considera as estratégias de solução de controle de carga ou de deslocamento.

O comportamento não-linear dos membros da estrutura leva em conta o efeito elasto-plástico com endurecimento e grandes deslocamentos.

A avaliação da resistência última e a história do colapso de estruturas, nas condições intacta e danificada, é um dos tópicos chave das técnicas avançadas de projeto baseadas na confiabilidade estrutural.

Exemplos numéricos são apresentados para demonstrar a precisão e eficiência da estratégia de solução no supercomputador hipercubo Intel iPSC/860 .

Abstract of the thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctor in Science (D.Sc.)

**COLLAPSE ANALYSIS OF OFFSHORE STRUCTURES ON THE INTEL
iPSC/860 HYPERCUBE SUPERCOMPUTER**

Manoel Rodrigues Justino Filho

MAY, 1993

Thesis Supervisor: Nelson Francisco Favilla Ebecken

Department: Civil Engineering

A parallel computational implementation on distributed memory supercomputer of a finite element program for ultimate strength analysis of offshore space frame structures is presented in this thesis.

The computer implementation uses domain decomposition methods and the parallel multifrontal solution of sparse system of equations.

The incremental iterative nonlinear analysis of the structure considers both solution strategies of load and displacement controls.

The nonlinear behavior of structural members take into account the elastic-plastic effect with material strain-hardening and large displacements.

Evaluation of the ultimate strength and history of collapse of structures in their intact and damaged conditions is one of the key issues in advanced design techniques based on structural reliability.

Numerical examples are presented to demonstrate the accuracy and efficiency of the solution strategy on the Intel iPSC/860 hypercube supercomputer .

ÍNDICE

CAPÍTULO I INTRODUÇÃO

I.1 Considerações iniciais e objetivos	1
I.2 Histórico geral	2
I.3 Conteúdo	3

CAPÍTULO II SOLUÇÃO DIRETA DE SISTEMAS DE EQUAÇÕES ALGÉBRICAS LINEARES ESPARSAS EM MÁQUINAS SEQUENCIAIS

II.1 Introdução	5
II.2 O problema algébrico da eliminação de Gauss para matrizes densas	6
II.3 Métodos de matrizes banda e perfil	8
II.4 Método esparsos gerais	9
II.5 Método frontal para problemas de elementos finitos	12
II.6 Método multifrontal	14

CAPÍTULO III DECOMPOSIÇÃO DE DOMÍNIO AUTOMÁTICA DE MALHAS DE ELEMENTOS FINITOS

III.1 Introdução	23
III.2 Regras heurísticas para decomposição de domínio de MEF	24
III.3 Algoritmo de decomposição de domínio de MEF para métodos esparsos	25

CAPÍTULO IV IMPLEMENTAÇÃO PARALELA NO SUPERCOMPUTADOR HIPERCUBO INTEL iPSC/860

IV.1 Introdução	27
IV.2 Custos de comunicação	27
IV.3 Aferição de desempenho	28
IV.4 Lei de Amdahl	28

IV.5 Implementação paralela do método dos elementos finitos	29
---	----

CAPÍTULO V FORMULAÇÃO NÃO-LINEAR FÍSICA E GEOMÉTRICA

V.1 Introdução	44
V.2 Solução incremental iterativa	44
V.3 Métodos de Newton-Raphson	45
V.4 Critérios de convergência	46
V.5 Formulação do elemento de pórtico espacial	47

CAPÍTULO VI ESTUDOS DE CASOS

VI.1 Introdução	57
VI.2 Coluna biapoiada com imperfeição inicial	57
VI.3 Coluna biengastada com imperfeição inicial	59
VI.4 Viga biapoiada com carga distribuída	61
VI.5 Viga monoengastada com carga concentrada	62
VI.6 Pórtico plano	63
VI.7 Pórtico espacial	66
VI.8 Plataforma fixa hipotética	69
VI.9 Plataforma fixa da bacia de Campos	80
VI.10 Plataforma fixa do golfo do México	86

CAPÍTULO VII CONCLUSÃO

VII.1 Considerações gerais	92
VII.2 Contribuições	92
VII.3 Sugestões para futuros trabalhos	93

APÊNDICE A REVISÃO DA TEORIA DE GRAFOS E SEU USO EM MATRIZES SIMÉTRICAS ESPARSAS	95
---	-----------

APÊNDICE B DESCRIÇÃO DO HARDWARE DO iPSC/860	98
APÊNDICE C RECOMENDAÇÕES PRÁTICAS PARA PROGRAMAÇÃO NO iPSC/860	103
APÊNDICE D ALGORITMO DE CONTROLE DE DESLOCAMENTOS	105
BIBLIOGRAFIA	109

CAPÍTULO I

INTRODUÇÃO

I.1 Considerações iniciais e objetivos

A utilização de máquinas vetoriais e paralelas na solução de problemas complexos de engenharia tem sido intensa, devido à redução drástica no tempo de execução do processamento em relação às máquinas sequenciais usuais.

Para tirar proveito das vantagens das máquinas paralelas, os algoritmos sequenciais existentes precisam ser modificados, pois os compiladores para máquinas paralelas atualmente disponíveis fazem, em geral, a paralelização de "loops" (compiladores para multiprocessadores com memória compartilhada), além da vetorização, utilização de registradores e otimização de código.

As arquiteturas paralelas existentes podem ser subdivididas em máquinas com memória compartilhada e máquinas com memória distribuída. Nas máquinas com memória compartilhada, como, por exemplo, Cray Y-MP/8, Alliant FX/8 e Convex C3880, tem-se poucos processadores e os conflitos de acesso de memória pelos processadores têm que ser tratados na implementação. Nas máquinas com memória distribuída não existe o problema de conflito de memória e o número de processadores disponível é bem maior que os das máquinas com memória global. Por exemplo, o hipercubo Intel iPSC/860 pode ter até 128 processadores e na Connection Machine CM-5 tem-se configurações com até 1024 processadores (ver Meurant, 1992). A grande desvantagem das máquinas com memória distribuída é o custo de comunicação para transferência de mensagens entre os processadores, que está sendo reduzido cada vez mais nas novas versões destas arquiteturas.

A meta dos fabricantes de supercomputadores é atingir velocidades de processamento da ordem de Teraflops ($1 \text{ Teraflop} = 10^{12} \text{ flops}$) ainda nesta década, através do uso da tecnologia MPP ("Massively Parallel Processing").

O objetivo deste trabalho é apresentar uma implementação computacional paralela, no supercomputador hipercubo Intel iPSC/860, de um programa de elementos finitos para a análise do colapso de estruturas offshore reticuladas.

A implementação utiliza um método heurístico de decomposição de domínio, para subdivisão da carga computacional equilibrada entre processadores, procurando reduzir o custo de comunicação.

A solução multifrontal paralela é adotada para a solução direta de sistemas de equações algébricas lineares, armazenadas na forma esparsa.

A estratégia de solução incremental iterativa com controle de parâmetros de carga ou deslocamento é empregada na análise não-linear estrutural.

Na análise do comportamento não-linear dos membros da plataforma, considera-se o método plástico nodal com endurecimento do material e grandes deslocamentos.

A determinação da resposta não-linear da estrutura até a carga de colapso, nas condições intacta e danificada, é um dos tópicos chave das técnicas avançadas de projeto baseadas na confiabilidade estrutural.

A utilização da metodologia de confiabilidade estrutural pode aumentar a segurança e reduzir os custos de construção, inspeção e manutenção de estruturas offshore. O estudo do comportamento dinâmico de estruturas de águas profundas implica na análise aleatória, que envolve simulações de fenômenos ambientais complexos, com diversas fontes de não-linearidade e incertezas. Através do uso de programas de computador, é importante que os requisitos de segurança local e global sejam atendidos de uma maneira mais econômica possível.

Os métodos de confiabilidade são uma extensão dos métodos de otimização e ambos necessitam de utilização computacional intensiva, envolvendo muitos parâmetros e sistemas de grande complexidade.

A utilização da presente estratégia de solução não-linear na análise de confiabilidade pode possibilitar a avaliação de novos conceitos de projeto, para aplicação em estruturas offshore para águas profundas.

I.2 Histórico geral

Apresenta-se a seguir um resumo das principais referências bibliográficas analisadas, sobre aplicações do método dos elementos finitos em computação paralela e vetorial.

A tese de Farhat (1987) trata do multiprocessamento em mecânica computacional, com aplicações no supercomputador Intel iPSC de memória distribuída e no Cray X-MP-4 de memória compartilhada. Utiliza um método de decomposição de domínio bastante simples (ver também Farhat, 1988) e suficiente para o método de solução de banda variável ("skyline") em paralelo (ver também Farhat e Wilson, 1988 e Farhat, 1989).

A tese de Lucas (1987) faz a implementação de sistemas de equações planares de problemas de engenharia elétrica no supercomputador Intel iPSC/2, com memória distribuída. A distribuição das equações entre os processadores é feita com o método "nested dissection" incompleto e a solução do sistema de equações é feita pelo método multifrontal esparsos e também pelo método iterativo de gradiente conjugados.

Alves (1989), em sua tese, utiliza uma rede de transputer T800 da Imos, para implementar um programa de elementos para análise dinâmica transiente. Utiliza o método de solução de banda variável de Farhat e Wilson (1988) para a solução do sistema de equações.

Malone (1990) utiliza o supercomputador Intel iPSC/d5, com memória distribuída, para análise não-linear de cascas tridimensionais. A solução do sistema de equações em paralelo é o de banda variável ("skyline").

Zhang e Liu (1991) implementam uma versão paralela do método frontal serial de Irons (1970) no supercomputador Alliant FX/8, com memória compartilhada.

Barton e Scott (1991) apresentam resultados de uma implementação do método de banda variável, para a solução de grandes modelos de elementos finitos, no supercomputador Intel iPSC/860.

Poole et alii (1992) descrevem o impacto dos algoritmos sequenciais de soluções de equações com vetorização de alta eficiência no método dos elementos finitos.

Para finalizar, o artigo de Meaurant (1992) apresenta uma visão global da evolução da computação científica em computadores paralelos e vetoriais.

I.3 Conteúdo

Apresenta-se a seguir o conteúdo dos capítulos desta tese.

No capítulo II tem-se uma discussão dos métodos diretos de solução de sistemas de equações algébricas lineares, direcionados para computadores sequenciais. Dentro do contexto do método dos elementos finitos, destacam-se os métodos de banda constante e variável, métodos esparsos gerais, métodos frontal e multifrontal.

Apresenta-se no capítulo III o algoritmo de decomposição de domínio, utilizado para subdividir uma malha irregular de elementos finitos entre os processadores, de modo a garantir uma carga computacional balanceada e transferência de mensagens minimizada entre os processadores.

A descrição da implementação paralela do programa de elementos finitos, no supercomputador Intel iPSC/860, é feita no capítulo IV. Apresentam-se os algoritmos das etapas de cálculo de um programa de elementos finitos que foram paralelizadas.

No capítulo V apresenta-se a formulação não-linear física e geométrica do elemento de pórtico espacial utilizado. A não-linearidade geométrica considera o efeito de grandes deslocamentos e pequenas deformações. A não-linearidade física utiliza o método plástico nodal, com a consideração do efeito de endurecimento isotrópico do material.

No capítulo VI são feitos estudos de casos de modelos estruturais existentes na literatura e também de plataformas fixas, para avaliação de resultados e a eficiência da implementação paralela do programa de elementos finitos.

No capítulo VII apresentam-se as conclusões desta tese e algumas sugestões para o desenvolvimento de trabalhos futuros nesta linha de pesquisa.

No apêndice A tem-se uma revisão da teoria de grafos e o seu uso em matrizes simétricas esparsas.

No apêndice B apresenta-se uma descrição sucinta do hardware do iPSC/860.

No apêndice C são apresentadas algumas recomendações práticas para programação no supercomputador iPSC/860.

No apêndice D tem-se a formulação do algoritmo de controle de deslocamentos utilizado.

CAPÍTULO II

SOLUÇÃO DIRETA DE SISTEMAS DE EQUAÇÕES ALGÉBRICAS LINEARES ESPARSAS EM MÁQUINAS SEQUENCIAIS

II.1 Introdução

Este capítulo apresenta uma discussão dos métodos diretos de solução de sistemas de equações algébricas lineares, para máquinas sequenciais, onde a matriz dos coeficientes é de grande dimensão e com forma de armazenamento esparsa.

Os algoritmos analisados baseiam-se no método de eliminação de Gauss para a solução do sistema de equações:

$$Ax = b \quad (\text{II.1})$$

onde A é a matriz de coeficientes, x o vetor solução e b o vetor de coeficientes do lado direito do sistema de equações.

Os métodos diretos de solução podem ser agrupados em quatro categorias principais :

- métodos de matriz banda e perfil;
- métodos esparsos gerais;
- método frontal;
- método multifrontal.

Os métodos de solução diretos na forma esparsa envolvem a solução das seguintes fases:

- reordenação nodal;
- fatoração simbólica;
- fatoração numérica;
- solução.

Neste trabalho será dado ênfase ao método de Cholesky para solução de matrizes simétricas, esparsas positiva definidas, de grandes modelos estruturais de elementos finitos.

II.2 O problema algébrico da eliminação de Gauss para matrizes densas

II.2.1 Eliminação de Gauss

O método de eliminação de Gauss consiste em se transformar a matriz A numa matriz triangular superior, através da criação de zeros na primeira coluna, depois na segunda e assim sucessivamente. Para $k=1, 2, \dots, n-1$, as fórmulas de recorrência para a eliminação de Gauss são (Duff, Erisman and Reid, 1989):

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - (a_{ik}^{(k)} / a_{kk}^{(k)})a_{kj}^{(k)}, i, j > k \quad (\text{II.2})$$

e

$$b_i^{(k+1)} = b_i^{(k)} - (a_{ik}^{(k)} / a_{kk}^{(k)})b_k^{(k)}, i > k \quad (\text{II.3})$$

onde:

$$a_{ij}^{(1)} = a_{ij}, i, j = 1, 2, \dots, n \quad (\text{II.4})$$

Assume-se sempre que $a_{kk}^{(k)} \neq 0$ para $k=1, 2, \dots, n$. Estes coeficientes são chamados pivôs na eliminação de Gauss.

Usando a notação:

$$A^{(k)}x = b^{(k)} \quad (\text{II.5})$$

para o sistema de equações após $k-1$ estágios, $k=1, 2, \dots, n$, onde $A^{(1)} = A$ e $b^{(1)} = b$. A matriz final $A^{(n)}$ é triangular superior.

II.2.2 Teorema da decomposição

Os métodos de eliminação de variáveis utilizam o conceito de decomposição de uma matriz A no produto de matrizes triangulares e diagonais, em alguns casos. Dentre os métodos de eliminação existentes, destacam-se o método de eliminação de Gauss e o método de Cholesky.

Pelo teorema de decomposição, uma matriz quadrada não-simétrica pode ser representada da seguinte forma (Duff, Erisman and Reid, 1989):

$$A = \bar{L}D\bar{U} \quad (\text{II.6})$$

onde \bar{L} e \bar{U} são matrizes triangulares inferior e superior respectivamente, com os termos diagonais unitários, e D é uma matriz diagonal. Esta decomposição é única.

Comparando-se a eliminação de Gauss com a decomposição (II.6), tem-se que a matriz triangular superior $A^{(n)}$ da eliminação é igual a $(D\bar{U})$.

Variantes da decomposição (II.6) podem ser obtidas pela associação da matriz D com as matrizes \bar{L} ou \bar{U} . Na decomposição de Crout, tem-se a associação $(\bar{L}D)\bar{U}$ e de Doolittle $\bar{L}(D\bar{U})$.

Para matrizes simétricas, $\bar{U} = \bar{L}^t$, onde \bar{L}^t é a transposta da matriz \bar{L} , então a equação (II.6) pode ser reescrita na forma :

$$A = \bar{L}D\bar{L}^t \quad (\text{II.7})$$

onde:

$$d_j = a_{jj} - \sum_{k=1}^{j-1} \bar{l}_{jk}^2 d_k \quad (\text{II.8})$$

$$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} \bar{l}_{ik} d_k \bar{l}_{jk}) / d_j, i > j \quad (\text{II.9})$$

II.2.3 Método de Cholesky

II.2.3.1 Fatoração

Uma matriz A é dita positiva definida se $x^t Ax > 0$, para todo vetor não nulo x .

Substituindo na equação (II.6) $L = \bar{L}D^{1/2}$, tem-se o método de Cholesky para matrizes simétricas positiva definidas:

$$A = LL^t \quad (\text{II.10})$$

onde:

$$l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2)^{1/2} \quad (\text{II.11})$$

$$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}) / l_{jj}, i > j \quad (\text{II.12})$$

Na equação (II.10), L é uma matriz triangular inferior e os termos da diagonal são sempre positivos.

Dentre os vários algoritmos de minimização de largura de banda ou da altura de perfil, os mais conhecidos são o Reverse-Cuthill-McKee (ver George e Liu, 1981), Sloan (1989) e Duff, Reid e Scott (1989).

Uma implementação simples de um algoritmo de solução de banda variável é apresentado também por George e Liu (1981).

II.4 Método esparsos gerais

II.4.1 Introdução

Os métodos esparsos têm-se tornado atrativos, devido a redução da demanda de memória associada à forma de armazenamento de dados na forma esparsa, através de algoritmos eficientes para minimização do preenchimento do sistema de equações.

A solução direta de um sistema de equações algébricas lineares esparsas $Ax=b$, consiste das seguintes fases (George and Liu, 1981) :

- **Reordenação:** cálculo da matriz de permutação P , para reduzir o número de termos não-nulos dos fatores de Cholesky L de PAP^t ;
- **Fatoração simbólica:** determinação da estrutura de L e prepara uma estrutura de dados para armazenar a matriz A e a estrutura de termos não-nulos de L ;
- **Fatoração numérica:** cálculo dos fatores de Cholesky L da matriz PAP^t ;
- **Solução triangular:** solução dos sistemas triangulares $Ly=Pb$ ("forward"), $L^t z = y$ (retrosubstituição) e cálculo de $x = P^t z$.

A determinação de uma matriz de permutação P , que minimize o número de termos não-nulos de L , é uma classe de problema NP-completo (Yannakakis, 1981). Dentre os vários métodos existentes para a minimização do preenchimento ("fill-in") de uma matriz, destaca-se o método do grau mínimo ("minimum degree").

O algoritmo é chamado de grau mínimo devido a sua interpretação pela teoria de grafos, ou seja, para um grafo associado a uma matriz esparsa, esta técnica seleciona o nó, para a próxima eliminação, com um menor número de vizinhos conectados ao mesmo.

Liu (1985) apresenta melhorias no algoritmo de grau mínimo em relação às implementações anteriores (George e Liu, 1981).

O método de "nested dissection" também pode ser utilizado (George e Liu, 1981), porém o método de grau mínimo é mais eficiente na maioria dos casos.

II.4.2 Fatoração simbólica

Esta fase corresponde ao cálculo das posições do preenchimento de L e alocação de uma estrutura de dados estática, para acomodar os índices e os termos não-nulos de L .

George e Liu (1981) apresentam uma implementação eficiente da fatoração simbólica, em termos da matriz L , da seguinte forma:

A estrutura de uma coluna j para $2 \leq j \leq n$ de L é obtida a partir da união da estrutura esparsa da coluna j de PAP^t e todas as colunas anteriores de L , cujo primeiro termo não-nulo é a linha j .

Alguma economia no armazenamento de índices inteiros pode ser obtida, através da técnica de compressão de índice de Sherman, utilizada também por George e Liu (1981). Esta técnica utiliza a propriedade de que os índices de linha não-nulos de uma coluna podem ser reutilizados nas colunas subsequentes, através de vetores apontadores.

II.4.3 Fatoração numérica

A fatoração numérica é a fase que consome mais tempo para ser realizada.

Para a fatoração de Cholesky existem três famílias de algoritmos (Heath et alii, 1990) :

- **Cholesky por linha** : as linhas sucessivas de L são calculadas uma a uma;
- **Cholesky por coluna** : as colunas sucessivas de L são obtidas uma a uma, usando-se as colunas de L previamente calculadas;
- **Cholesky por submatriz** : as colunas sucessivas de L são obtidas uma a uma, e tão logo que uma coluna j é completada, o seu efeito nas colunas subsequentes é considerado de imediato (atualização do tipo "rank -1").

Na fatoração de Cholesky de matrizes esparsas, os dois métodos orientados por coluna, Cholesky por coluna e Cholesky por submatriz, são os mais utilizados nas implementações em máquinas vetoriais e paralelas (ver Heath et alii, 1990).

No método de Cholesky orientados por colunas, existem duas subtarefas fundamentais:

- *cmod* (j,k) - modificação da coluna j pela coluna k , $k < j$;
- *cdiv* (j) - divisão da coluna j por um escalar (diagonal da coluna j).

A estrutura de termos não-nulos de uma coluna j é definida como:

$$\text{struct}(L_{*j}) = \{k > j | l_{kj} \neq 0\} \quad (\text{II.17})$$

Analogamente, as colunas de uma linha i são:

$$\text{struct}(L_{i*}) = \{k < i | l_{ik} \neq 0\} \quad (\text{II.18})$$

Em termos de operações básicas, os algoritmos (II.1) e (II.2) são pseudo-códigos da fatoração de Cholesky por coluna e Cholesky por submatriz, respectivamente.

Algoritmo II.1 Fatoração de Cholesky esparsa por colunas

begin

for $j=1$ to n **do**

for $k \in \text{struct}(L_{j*})$

$\text{cmod}(j, k)$;

end-for;

$\text{cdiv}(j)$;

end-for ;

end.

Algoritmo II.2 Fatoração de Cholesky esparsa por submatriz

begin

for $k=1$ to n **do**

$\text{cdiv}(k)$;

for $j \in \text{struct}(L_{*k})$ **do**

$\text{cmod}(j, k)$;

end-for ;

end-for;

end.

Uma implementação do método de Cholesky esparsa por colunas, bastante conhecida, é dada por George e Liu (1981).

O método frontal e o multifrontal são variações do método de Cholesky por submatriz, os quais serão discutidos nos itens (II. 5) e (II.6) deste capítulo, respectivamente.

II.4.4 Solução triangular

A solução dos sistemas triangulares (fases de "forward" e retrossubstituição) necessita somente de uma pequena parcela do tempo total de solução gasto, para a solução de um sistema linear esparsa em máquinas seriais.

George e Liu (1981) apresentam uma implementação da solução triangular para a matriz L de Cholesky, armazenada na forma esparsa.

II.5 Método frontal para problemas de elementos finitos

O método frontal, implementado por Irons (1970), é uma variante do método de eliminação de Gauss para solução de sistemas de equações "out-of-core", em programas de elementos finitos, para solução de matrizes banda. A idéia principal do método frontal é executar as fases de montagem das equações e de eliminação de variáveis ao mesmo tempo. Irons (1970) sugere que, assim que os coeficientes de uma equação estejam totalmente montados, a eliminação desta equação pode ser realizada. Assim, a matriz dos coeficientes do sistema de equações nunca é totalmente montada na memória principal, uma vez que as equações eliminadas podem ser transferidas para memória secundária em disco.

Na memória principal ou frente permanecem somente as equações ativas em cada fase da fatoração. O número de equações presentes no frente é designado largura de frente. O tamanho máximo do problema a ser analisado é limitado pela máxima largura de frente, que é bem menor que a ordem do sistema de equações.

Para minimização da largura de frente, a ordenação dos elementos da malha de elementos finitos é importante, sendo a ordenação dos nós não relevante.

Na figura (II.2) tem-se a representação da decomposição do método frontal para uma matriz banda.

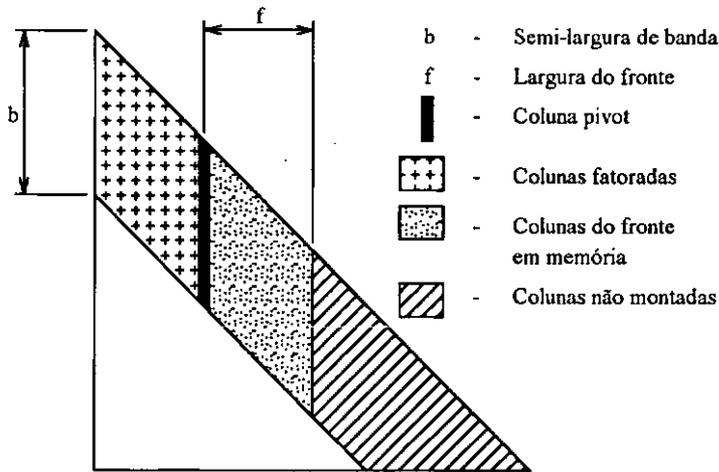


Figura II.2 Representação do método frontal para uma matriz banda simétrica triangular inferior.

O algoritmo (II.3) apresentado a seguir sumariza o método frontal.

Algoritmo II.3 Método frontal de Irons

begin

for $i=1$ **to** número de elementos **do**

{Loop sobre todos os elemento do modelo}

Calcula a matriz de rigidez do elemento i ;

Faz a montagem da matriz de rigidez do elemento na matriz frontal do sistema ;

for $j=1$ **to** número de nós do elemento **do**

{Loop sobre o número de nós de incidência do elemento}

if último-elemento (j)= i **then**

{Verifica se o nó j ocorre pela última vez no elemento i }

Armazena os coeficientes de rigidez associados aos graus de liberdade do nó j e a parcela do vetor de cargas associada aos graus de liberdade ativos no frente ;

Processa a eliminação de Gauss para graus de liberdade associados ao nó j ;

end-if ;

end-for ;

end-for ;

Processa a retrossubstituição do sistema de equações;

end.

A eliminação de Gauss engloba também a fase de "forward" do vetor de cargas. Um importante aspecto do método frontal é que as operações de eliminação são executadas em uma matriz cheia, possibilitando a vetorização e a eliminação múltipla de variáveis.

O método frontal deve ter uma largura de frente pequena, para ser eficiente com relação aos demais métodos esparsos de eliminação de Gauss. Existem bons algoritmos de minimização de frente, tais como Sloan (1989) e Duff, Reid e Scott (1989).

II.6 Método multifrontal

II.6.1 Revisão geral

O método multifrontal é uma generalização do método frontal de Irons (1970) com a utilização de múltiplas matrizes frontais.

Este método foi implementado por Duff e Reid (1983) e representa um avanço significativo na solução direta de sistemas esparsos.

A economia no número de operações de ponto flutuante, em relação ao método frontal, em geral é maior no método multifrontal.

As matrizes frontais do método frontal e multifrontal são matrizes triangulares densas armazenadas na memória principal, com as seguintes vantagens :

- utilização de vetorização ;
- redução das atividades de paginação de máquinas com memória virtual ;
- utilização de memória "cache".

A implementação do método multifrontal, utilizando o conceito de eliminação em massa de variáveis, é utilizado por Ashcraft (1987) e por Amestoy e Duff (1989).

Os conceitos gerais de formulação e de implementação do método multifrontal são revistos por Liu (1992).

II.6.2 Revisão da fatoração de Cholesky por submatrizes

Considere-se a fatoração de uma matriz cheia A de ordem n , em LL' , usando o método da fatoração de Cholesky por submatrizes. Considerando que um passo da formulação do produto externo da fatoração de Cholesky pode ser expresso da seguinte forma (Liu, 1992):

$$A = \begin{pmatrix} d & v' \\ v & C \end{pmatrix} = \begin{pmatrix} \sqrt{d} & 0 \\ v/\sqrt{d} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & C - vv'/d \end{pmatrix} \begin{pmatrix} \sqrt{d} & v'/\sqrt{d} \\ 0 & I \end{pmatrix} \quad (\text{II.19})$$

onde d é o primeiro termo da diagonal, v um vetor com $(n-1)$ elementos, I uma matriz identidade e C uma submatriz de ordem $n-1$. A submatriz $C - vv'/d$ é a parte remanescente a ser fatorada, cuja fatoração é realizada sucessivamente usando a mesma técnica anterior.

A forma de fatoração anterior pode ser generalizada para a fórmula de blocos. Considera-se que $j-1$ passos de fatoração tenham sido realizados ($j > 1$), então esta fatoração parcial pode ser vista através de um bloco de partição 2×2 (Liu, 1992) :

$$A = \begin{pmatrix} B & V' \\ V & C \end{pmatrix} = \begin{pmatrix} L_B & 0 \\ VL_B^{-1} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & C - VB^{-1}V' \end{pmatrix} \begin{pmatrix} L_B' & L_B^{-1}V' \\ 0 & I \end{pmatrix} \quad (\text{II.20})$$

onde $B = L_B L_B'$ é a fatoração de Cholesky da submatriz B de ordem $j-1$.

O complemento de Schur $C - VB^{-1}V'$ representa a parte remanescente da matriz a ser fatorada. Nota-se que a submatriz $-VB^{-1}V'$ de ordem $(n-j+1)$ representa a atualização das primeiras $j-1$ linhas e colunas da submatriz C , para formar a submatriz $C - VB^{-1}V'$. A contribuição desta submatriz de atualização (com $j \geq 1$) pode ser expressa da seguinte forma:

$$-VB^{-1}V' = -(VL_B^{-1})(L_B^{-1}V') = - \sum_{k=1}^{j-1} \begin{pmatrix} l_{jk} \\ \vdots \\ l_{nk} \end{pmatrix} (l_{jk} \dots l_{nk}) \quad (\text{II.21})$$

Isto corresponde à soma da atualização do produto externo das $j-1$ primeiras colunas dos fatores de Cholesky. A noção da forma de atualização do produto externo é a base da definição das matrizes frontais e das matrizes de atualização da fatoração multifrontal.

II.6.3 Definição da árvore de eliminação

A estrutura da árvore de eliminação (Liu, 1990) é muito importante na organização do método multifrontal. Seja uma matriz A de ordem n , simétrica e positiva definida, com matriz de fator de Cholesky L . A árvore de eliminação da matriz A é definida como uma estrutura de n nós $\{1, 2, \dots, n\}$ tal que o nó p é o pai do nó j , se e somente se, $p = \min\{i > j / l_{ij} \neq 0\}$. Utiliza-se a notação $T(A)$ para representar a árvore de eliminação da matriz A .

Apresentam-se a seguir as propriedades importantes da árvore de eliminação (Liu, 1990 e 1992):

- **Propriedade 1** . Se um nó k é um descendente do nó j na árvore de eliminação, então a estrutura da coluna $(l_{jk} \dots l_{nk})^t$ está contida na estrutura da coluna $(l_{jj} \dots l_{nj})^t$;
- **Propriedade 2**. Se $l_{jk} \neq 0$ e $k < j$, então o nó k é um descendente do nó j na árvore de eliminação.

Na figura (II.3) tem uma malha de elementos finitos onde, por simplicidade, assume-se que cada nó tenha somente um grau de liberdade.

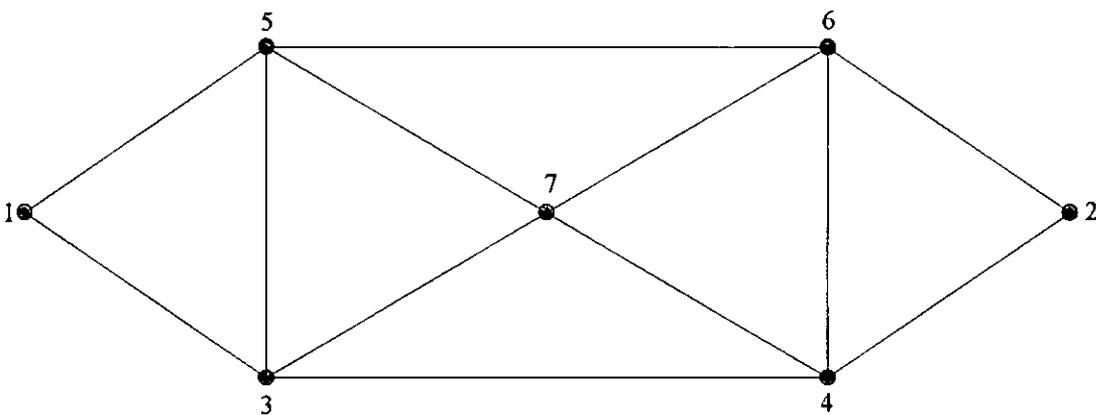


Figura II.3 Malha de elementos finitos com uma equação por nó.

Na figura (II.4) a seguir, tem-se a matriz A e a matriz de fatores de Cholesky L .

$$A = \begin{pmatrix} 1 & \bullet & & & & & \\ 2 & & \bullet & & & & \\ 3 & \bullet & & \bullet & & & \\ 4 & & \bullet & \bullet & \bullet & & \\ 5 & \bullet & & \bullet & & \bullet & \\ 6 & & \bullet & & \bullet & \bullet & \bullet \\ 7 & & & \bullet & \bullet & \bullet & \bullet \end{pmatrix} \quad L = \begin{pmatrix} 1 & \bullet & & & & & \\ 2 & & \bullet & & & & \\ 3 & \bullet & & \bullet & & & \\ 4 & & \bullet & \bullet & \bullet & & \\ 5 & \bullet & & \bullet & \circ & \bullet & \\ 6 & & \bullet & & \bullet & \bullet & \bullet \\ 7 & & & \bullet & \bullet & \bullet & \bullet \end{pmatrix}$$

\circ – preenchimento
 \bullet – termos não-nulos

Figura II.4 Matrizes de coeficientes A e de fatores de Cholesky L da malha da figura (II.3)

Nota-se, na figura (II.4), que existe preenchimento no termo l_{54} .

Na figura (II.5) a seguir tem-se a árvore de eliminação.

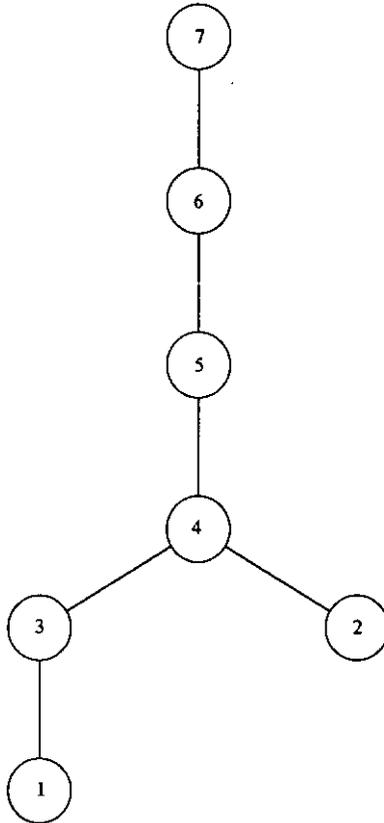


Figura II.5 Árvore de eliminação da matriz da figura (II.4)

Designa-se aqui a notação $T(j)$ para representar o conjunto de nós descendentes do nó j na árvore de eliminação, inclusive o nó j .

II.6.4 Definição das matrizes frontais e de atualização

Considera-se a j -ésima coluna de L com os índices de linha não-nulos em $L_{*,j}$, índices $\{j, i_1, \dots, i_r\}$, ou seja, a coluna j possui r termos não nulos fora da diagonal.

A matriz de atualização de subárvore na coluna j é definida como (Liu, 1992):

$$\bar{U}_j = - \sum_{k \in T(j) - \{j\}} \begin{pmatrix} l_{jk} \\ l_{i_1 k} \\ \vdots \\ l_{i_r k} \end{pmatrix} (l_{jk} l_{i_1 k} \dots l_{i_r k}) \quad (\text{II.22})$$

Isto é, a matriz \bar{U}_j contém as contribuições não nulas do produto externo das colunas precedentes, que são descendentes da coluna j na árvore de eliminação.

As matrizes frontais no método multifrontal usando fatoração de Cholesky são submatrizes triangulares cheias e de ordem bem menor que a ordem n da matriz A . Para cada variável, ou conjunto de variáveis supernodais, está associada uma matriz frontal.

A j -ésima matriz frontal de A é definida da seguinte forma:

$$F_j = \begin{pmatrix} a_{jj} & a_{j i_1} & \dots & a_{j i_r} \\ a_{i_1 j} & & & \\ \vdots & & 0 & \\ a_{i_r j} & & & \end{pmatrix} + \bar{U}_j \quad (\text{II.23})$$

A ordem $r+1$, da matriz frontal F_j e da matriz \bar{U}_j , é o número de não nulos da j -ésima coluna da matriz L .

A submatriz remanescente U_j a ser fatorada, após a realização de um passo de eliminação na matriz F_j , é dada por Liu (1992):

$$U_j = - \sum_{k \in T(j)} \begin{pmatrix} l_{i,k} \\ \vdots \\ l_{i,k} \end{pmatrix} (l_{i,k} \dots l_{i,k}) \quad (\text{II.24})$$

A matriz U_j possui uma linha e uma coluna a menos que a matriz \bar{U}_j . A matriz U_j é usada para formar a j -ésima matriz frontal F_j , enquanto que U_j é gerada após a realização de um passo de eliminação da matriz F_j .

II.6.5 Algoritmo de método multifrontal

Conforme visto anteriormente, a matriz frontal F_j é montada usando a coluna A_{*j} e as matrizes frontais remanescentes U_j dos nós filhos do nó j , caso existam, na árvore de eliminação.

Assume-se aqui que, durante a fase de montagem da matriz F_j , as matrizes de atualização U_j dos seus filhos estejam disponíveis numa pilha de trabalho (“stack”) em memória (Duff e Reid, 1983).

O algoritmo apresentado a seguir é constituído em termos da estrutura da árvore de eliminação e da pilha de trabalho, que armazena em memória as matrizes de atualização.

Algoritmo II.4 Método multifrontal para fatoração de Cholesky

begin

for $j=1$ **to** n **do**

 Aloca espaço para a matriz frontal F_j ;

 Sejam $\{f_1, f_2, \dots, f_t\}$ os filhos do nó j na árvore de eliminação ;

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

 Traz a matriz U_{f_i} do topo da pilha;

 Monta matriz U_{f_i} em F_j ;

end-for ;

Monta a coluna $A_{\bullet j}$ na matriz F_j ;

Processa um passo da fatoração de Cholesky para obter $L_{\bullet j}$ e U_j ;

Armazena a matriz U_j no topo da pilha de trabalho;

end-for ;

end.

A montagem das matrizes U_{f_i} na matriz F_j é feita adicionando-se os elementos não nulos de U_{f_i} nas posições apropriadas de F_j .

Nota-se que, se o nó j não é uma folha da árvore de eliminação (ou seja, $t > 0$), espaço em memória pode ser salvo, sobrepondo-se o espaço da matriz F_j com a matriz frontal existente no topo da pilha.

II.6.6 Armazenamento e manipulação das matrizes frontais e de atualizações

Para minimização do espaço da pilha de trabalho, em memória, para armazenamento das matrizes frontais e as matrizes de atualizações, existem algoritmos de reordenação topológica da árvore de eliminação que não alteram o preenchimento da matriz, após o algoritmo de ordenação primário ("minimum degree").

Um desses algoritmos é aquele que renumera um nó antes de seu pai na árvore de eliminação (Liu, 1990).

Uma descrição mais detalhada destes algoritmos é dada por Liu (1987, 1990 e 1992).

II.6.7 Eliminação em massa de variáveis

Uma importante propriedade da implementação multifrontal é a eliminação em massa de variáveis, também designada eliminação de variáveis supernodais (Ashcraft, 1987 e Liu, 1992).

Um supernó é definido como um conjunto de nós contínuos $\{j, j+1, j+t\}$, se:

$$\text{struct}(L_{\bullet k}) = \text{struct}(L_{\bullet k+1}) \cup \{k+1\}, j \leq k \leq j+t-1 \quad (\text{II.25})$$

ou seja, um supernó corresponde a um próximo bloco de colunas contíguas na matriz de fatores de Cholesky L , onde o bloco diagonal correspondente é triangular inferior cheio e

todas as colunas possuem estruturas de colunas com termos não nulos idênticos fora da diagonal.

Por exemplo, na árvore de eliminação da figura (II.5), os nós 4, 5, 6 e 7 constituem um supernó.

Na figura (II.6) a seguir, tem-se a árvore de eliminação supernodal da matriz da figura (II.4).

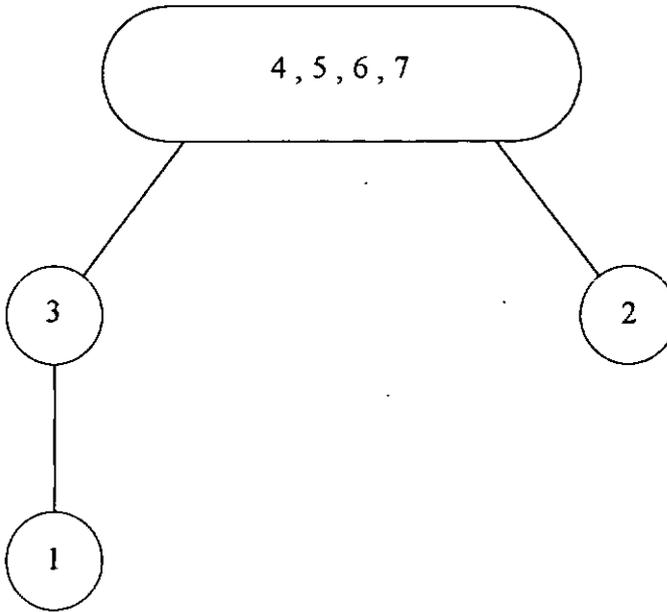


Figura II.6 Árvore de eliminação supernodal da matriz da figura (II.4).

II.6.8 Subestruturação e o método multifrontal

A técnica de subestruturação é bastante utilizada na solução de grandes modelos de elementos finitos.

A solução multifrontal "out-of-core" permite que as matrizes frontais sejam armazenadas temporariamente em memória auxiliar, para realizar a solução de grandes sistemas de equações.

A técnica de condensação estática pode ser estendida a subestruturas, que consistem de conjuntos de elementos finitos com nós internos e nós de fronteira. Na solução de modelos de elementos finitos por subestruturação, existem várias matrizes frontais uma para cada subestrutura e a solução multifrontal pode ser utilizada.

Após montagem das matrizes frontais das subestruturas, a condensação estática das variáveis internas das subestruturas pode ser realizada. As matrizes reduzidas resultantes, associadas às variáveis de fronteira, são designadas superelementos.

Estes superelementos são combinados numa única matriz frontal, para eliminação final das variáveis de fronteira do modelo.

Após a solução das variáveis de fronteira, a solução das variáveis internas em cada subdomínio pode ser realizada.

CAPÍTULO III

DECOMPOSIÇÃO DE DOMÍNIO AUTOMÁTICA DE MALHAS DE ELEMENTOS FINITOS

III.1 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.

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 (1992a) 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.

III.2 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. 3 Algoritmo de decomposição de domínios de MEF para métodos esparsos

O algoritmo de decomposição proposto por Justino Filho e Ebecken (1992a) e os demais descritos no item (III.1) 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 (III.1), apresentado a seguir.

Definição complementar de variáveis:

- p - número de subdomínios ou processadores;
- i - subdomínio ou processador;
- ne - número de elementos presente na MEF;
- nm - 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 III.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 nm/p nós alocados;

end.

O algoritmo (III.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.

No capítulo VI apresentam-se os resultados da decomposição de domínio para os exemplos analisados.

CAPÍTULO IV

IMPLEMENTAÇÃO PARALELA NO SUPERCOMPUTADOR HIPERCUBO iPSC/860

IV.1 Introdução

Neste capítulo apresenta-se a implementação computacional paralela do programa de elementos finitos no supercomputador Intel iPSC/860.

Uma descrição geral do hardware do iPSC/860 é feita no apêndice B.

As definições gerais da solução multifrontal sequencial apresentadas no capítulo II são reutilizadas aqui, para a implementação da solução multifrontal paralela.

No método dos elementos finitos, as seguintes fases de solução podem ser executadas em paralelo:

- Cálculo das matrizes dos elementos e montagem da matriz de rigidez global com estrutura de dados distribuída entre os nós do cubo;
- Cálculo do vetor de forças desequilibradas;
- Fatoração da matriz da estrutura;
- "Forward" das cargas;
- Retrossubstituição para obtenção dos deslocamentos nodais;
- Cálculo de resultantes de tensões nos elementos e montagem do vetor de forças resistentes internas da estrutura.

IV. 2 Custo de comunicação

Nas máquinas paralelas com memória distribuída, o custo de comunicação para transferência de mensagens entre processadores precisa ser considerado.

O custo de comunicação de transferência de m Mbytes, entre dois processadores, é calculado da seguinte maneira:

$$com(m) = \alpha_a + \beta_a m \quad (IV.1)$$

onde α_a é o tempo requerido para iniciar a mensagem e β_a é a taxa de transferência da mensagem.

Para o iPSC/860 $\alpha_d = 65 \times 10^{-6}$ segundos e $\beta_d = 2.8$ Mbytes por segundo no pico de largura de banda de hardware (ver Apêndice B).

IV.3 Aferição do desempenho

O "speed-up" de um algoritmo paralelo para um problema particular é definido da seguinte maneira:

$$S_p = T_{seq} / T_p \quad (IV.2)$$

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.

A versão sequencial do programa, rodando num nó do cubo, é utilizada para aferir a eficiência e o "speed-up" da implementação paralela.

Para alguns problemas, a versão sequencial mais rápida do algoritmo não pode ser paralelizada e, então, dois algoritmos distintos são considerados na determinação do "speed-up".

O "speed-up" pode ser utilizado para definir a eficiência da implementação paralela da seguinte maneira:

$$E_p = S_p / p = T_{seq} / (pT_p) \quad (IV.3)$$

O valor da eficiência E_p expressa a taxa de utilização dos processadores e tem-se que, $0 \leq E_p \leq 1$.

Os tempos de solução nos processadores do iPSC/860 foram determinados pela rotina `dclock()`, que fornece o tempo de solução, em dupla precisão, a partir da inicialização do cubo.

IV.4 Lei de Amdahl

Na maioria das situações práticas nem todas as operações de um algoritmo complexo podem ser executadas com mesmo desempenho. Quando duas partes de uma tarefa são executadas em diferentes velocidades, então o tempo total de CPU pode ser expresso como uma função destas velocidades.

Tem-se que a tarefa executada em menor velocidade pode influir desfavoravelmente na performance global. Este efeito é denominado como lei de Amdahl (ver Dongarra et alii, 1991).

No contexto do processamento paralelo, considera-se por simplicidade que uma tarefa constituída por operações básicas, executadas com uma mesma velocidade de computação, e que uma fração f destas operações pode ser executada em paralelo, em p processadores, enquanto que o resto do trabalho pode ser feito em um único processador (execução serial). Então, o tempo de execução da tarefa, utilizando p processadores, é:

$$T_p = fT_1/p + (1-f)T_1 \quad (\text{IV.4})$$

Assim, o "speed-up" neste caso é:

$$S_p = \frac{p}{f + (1-f)p} \quad (\text{IV.5})$$

Note que, para $f < 1$, então:

$$S_p \leq 1/(1-f) \quad (\text{IV.6})$$

A eficiência é dada por:

$$E_p = \frac{1}{f + (1-f)p} \quad (\text{IV.7})$$

A lei de Amdahl é utilizada para definir a viabilidade de implementação de um sistema numa máquina paralela.

IV.5 Implementação paralela do método dos elementos finitos

IV.5.1 Estrutura geral do programa em cada processador do cubo

O algoritmo de decomposição de domínio, executado no processador "host", é responsável pela distribuição equilibrada de elementos e nós do modelo de elementos finitos em cada processador do cubo.

O nó 0 do cubo é o processador supervisor responsável pelo controle da análise não-linear e pela comunicação com o "host".

Cada processador alocado no cubo, após receber a estrutura de dados correspondentes ao seu subdomínio, realiza em paralelo as seguintes tarefas:

- Cálculo das matrizes dos elementos do subdomínio e montagem da matriz de rigidez global com estrutura de dados pertencentes ao subdomínio;
- Cálculo do vetor das forças desequilibradas;
- Fatoração distribuída, com comunicação entre os processadores para os graus de liberdade de fronteira entre subdomínios;
- "Forward" distribuído das cargas, com comunicação entre os processadores, para os graus de liberdade de fronteira entre os subdomínios;
- Retrosubstituição distribuída, com comunicação entre os processadores para os graus de liberdade de fronteira entre subdomínios, para obtenção de deslocamentos nodais;
- Operação global de coleta do vetor solução, distribuído entre os processadores;
- Cálculo distribuído de resultantes de tensões em cada elemento do subdomínio do processador e montagem do vetor de forças resistentes internas, para os graus de liberdade associados aos processadores;
- Operação global de soma das contribuições do vetor de forças internas resistentes distribuídas entre os processadores.

O algoritmo (IV.1) apresenta a solução incremental iterativa realizada em cada processador do cubo.

Algoritmo IV.1 Solução incremental iterativa distribuída nos processadores do cubo

begin

$i=1$;

while ($i \leq$ número de passos de carga **and** critério de convergência = ok) **do**

repeat

Montagem da matriz de rigidez distribuída;

{Operação global de sincronização entre processadores do iPSC/860}

gsync();

Cálculo das forças desequilibradas;

```

{Operação global de sincronização entre processadores}

gsync();

Cálculo da fatoração de Cholesky distribuída;

gsync();

Cálculo do "forward" distribuído das cargas;

gsync();

Cálculo da retrosubstituição distribuída;

{Operação global de coleta do vetor solução  $U$ , distribuído entre
os processadores do iPSC/860}

gcolx( $U$ );

Cálculo de resultantes de tensões e do vetor de forças internas
resistentes distribuído;

{Operação global de soma das contribuições do vetor de forças
internas  $F_{int}$ , distribuído entre os processadores do iPSC/860};

gdsum( $F_{int}$ );

until (critério de convergência = ok or excedido limite de iterações);

i=i+1;

end-while;

```

end.

No algoritmo (IV.1) as funções de operações globais `gsync()`, `gcolx()` e `gdsum()` pertencem à biblioteca de funções de comunicação do iPSC/860 (ver Intel, 1991b).

IV.5.2 Montagem distribuída da matriz de rigidez da estrutura

A montagem da matriz de rigidez global, distribuída entre os processadores, é constituída pelo cálculo das matrizes dos elementos do subdomínio e montagem da matriz global.

As matrizes dos elementos de fronteira entre subdomínios são enviadas aos processadores responsáveis pelos subdomínios adjacentes a estes elementos.

O algoritmo (IV.2) apresenta o esquema da montagem da matriz global distribuída em cada processador.

Algoritmo IV.2 Montagem da matriz de rigidez global distribuída em cada processador

begin

for $i = 1$ **to** número de elementos do proc. p **do**

 Calcula a matriz de rigidez do elemento i ;

 Adiciona a matriz do elemento na matriz global;

if (elemento i é de fronteira) **then**

 Envia a matriz do elemento i para os processadores responsáveis pelos subdomínios de fronteira;

end-if;

end-for;

 Recebe as matrizes de fronteira de outros subdomínios;

 Adiciona as matrizes de fronteira na matriz global;

end.

IV.5.3 Fatoração paralela

A fatoração da matriz de rigidez global do sistema de equações é a fase que, em geral, mais consome tempo de execução.

O método multifrontal paralelo empregado neste trabalho foi utilizado por Duff (1986, 1989) para máquinas com memória compartilhada e por Lucas (1987), para máquinas com memória distribuída. Também a versão paralela do método frontal serial de Irons (1970), para máquinas com memória compartilhada, foi feita por Zhang e Liu (1991). A versão sequencial do algoritmo multifrontal de Duff e Reid (1983) utiliza:

- Reordenação das equações pelo método do grau mínimo;

- Pós-reordenação da árvore de eliminação para reduzir o espaço alocado em memória para as matrizes frontais;
- As matrizes frontais intermediárias são armazenadas em memória, utilizando uma estrutura de dados de pilha.

A paralelização do método multifrontal, a partir da organização da estrutura de dados da versão sequencial do algoritmo não é viável pelos seguintes motivos:

- A reordenação das equações é feita pelo método de "nested dissection" incompleto (Lucas, 1987) ou pelo método do grau mínimo;
- A pós-reordenação das equações tem por objetivo reduzir a altura da árvore de eliminação, para aumentar a paralelização da solução (Liu, 1989b);
- As matrizes frontais intermediárias não podem ser armazenadas em pilha e são utilizadas técnicas de alocação dinâmica tipo "binary buddy", utilizada por Duff (1989), ou então, o recurso de alocação dinâmica de memória, disponível no Fortran 90.

Na presente implementação paralela da solução multifrontal em máquina distribuída, as transferências de mensagens são feitas no modo síncrono, pela sua simplicidade de implementação e depuração, pois no método assíncrono a dificuldade de implementação é maior, embora possa existir uma redução da penalização de comunicação. A técnica "binary buddy" foi a escolhida para alocação dinâmica de memória para armazenamento das matrizes intermediárias em cada processador. A técnica de eliminação em massa de variáveis, ou eliminação supernodal, foi utilizada (Liu, 1992). O algoritmo (IV.3) da solução multifrontal paralela da fatoração de Cholesky é para ser executado em processadores com memória distribuída. Neste algoritmo, tem-se as seguintes definições:

- $ncol$ - número de colunas da matriz pertencentes ao processador p ;
- $mapeq(*)$ - é uma função que mapeia as equações em cada processador. Por exemplo, o processador responsável pela equação j é $p = mapeq(j)$;
- $nmod$ - vetor de dimensão $ncol$ que armazena inicialmente o número de filhos na árvore de eliminação para cada coluna. Por exemplo, se para uma coluna j tem-se $nmod_j = 0$, então a coluna j é uma folha na árvore de eliminação. Cada vez que um filho de uma coluna j na árvore de eliminação é processado, então o elemento $nmod_j$ é decrementado da unidade;

- *ncolmod* - variável que armazena, em cada instante, o número de colunas que já foram fatoradas;
- *pai(*)* - função que retorna o pai de cada equação na árvore de eliminação;
- *nelim* - número de colunas presentes em uma dada eliminação supernodal;
- *j,k* - índices das colunas L_{*j} e L_{*k} , respectivamente;
- *jlast* - última coluna da eliminação supernodal;
- U_j - matriz frontal de atualização associada à coluna *j*;
- F_j - matriz frontal associada a coluna *j*;

Algoritmo IV.3 Fatoração multifrontal paralela em processadores com memória distribuída.

begin

{Inicialização}

ncolmod = 0;

while (*ncolmod* < *ncol*) **do**

j = 0;

Procura uma coluna *j* tal que $nmod_j = 0$;

if (*j* > 0) **then**

{Existe uma coluna pronta para eliminação}

nelim = 1;

Calcula o número de variáveis supernodais associadas à coluna *j*
(*nelim* ≥ 1);

Aloca espaço na memória dinâmica para a matriz frontal
supernodal F_j do nó *j*;

Traz da memória dinâmica as matrizes frontais de atualização dos
filhos;

Processa a eliminação de variáveis supernodais associadas à coluna j , através da fatoração de Cholesky por submatrizes;

{Determinação da última coluna supernodal eliminada}

$jlast = j + nelim - 1;$

{Determina o processador pai da última coluna de eliminação supernodal}

$proc = mapeq(pai(jlast));$

if ($proc = p$) **then**

{O pai da matriz frontal de atualização U_j pertence ao processador p }

Aloca espaço na memória dinâmica para armazenar a matriz frontal supernodal de atualização U_j ;

else

{O pai da matriz frontal de atualização U_j pertence a outro processador}

Envia a matriz frontal supernodal de atualização U_j para o processador $proc$;

end-if;

Libera da memória dinâmica o espaço reservado para a matriz frontal F_j ;

$ncolmod = ncolmod + nelim;$

else

{Recebimento de matrizes frontais de atualização de outros processadores}

Espera o aviso de chegada de uma matriz frontal de atualização de um filho, cujo pai é uma coluna k pertencente a este processador;

Aloca espaço na memória dinâmica e recebe a mensagem contendo a matriz de atualização vinda de outro processador;

$$nmod_k = nmod_k - 1;$$

end-if;

end-while;

end.

No algoritmo (IV.3) cada processador trabalha independentemente, fatorando suas colunas internas, e a comunicação existe somente para as colunas de fronteira de subdomínios.

IV.5.4 "Forward" paralelo

A fase de "forward" consiste na obtenção do vetor y do sistema triangular inferior $Ly = b$. Na implementação computacional o vetor y e o vetor b ocupam as mesmas posições de memória.

No algoritmo (IV.4) tem-se o "forward" serial de um sistema esparso.

Algoritmo IV.4 "Forward" serial de um sistema esparso

begin

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

{Armazena y_i }

$$y_i = b_i / l_{ii};$$

for $j \in \text{struct}(L_{*j})$ **do**

$$b_j = b_j - y_i l_{ji};$$

end-for;

end-for;

end.

Como pode ser visto no algoritmo (IV.4), a fase de "forward" é um processo de atualização do vetor b pelas mesmas transformações lineares usadas para fatorizar a matriz A . Cada coluna j de L é multiplicada pelo elemento b_j e subtraída do vetor b . O "loop" externo do algoritmo (IV.4) seleciona a coluna que será usada para atualizar o vetor b . O "loop" interno contém as multiplicações acumulativas e pode ser vetorizado. Assim, a fase de "forward" é uma extensão da fase de atualização de decomposição de Cholesky, aplicada ao vetor b .

Para a fase de "forward" executada em processadores paralelos com memória distribuída, tem-se:

- As colunas da matriz L dos coeficientes de Cholesky e o vetor de coeficientes b encontram-se distribuídos entre os processadores;
- As colunas internas aos processadores podem atualizar o vetor b em paralelo, sem comunicação entre processadores;

Uma coluna j , que pertence a uma fronteira de um subdomínio de um processador, precisa receber as atualizações do vetor b de outros processadores responsáveis pelos subdomínios adjacentes. Após isto, o processador adiciona estas atualizações às suas próprias atualizações e transforma o vetor b com a coluna j .

O algoritmo (IV.5) é o pseudo-código da fase de "forward" em máquinas com memória distribuída. As definições de variáveis feitas na fase de fatoração são reutilizadas aqui. Definição complementar de variáveis:

- *temp* - vetor temporário auxiliar de atualização do vetor b ;
- *ncolfwd* - variável que armazena, em cada instante, o número de colunas com a operação de "forward" completada.

Algoritmo IV.5 "Forward" paralelo em máquina com memória distribuída

begin

{Inicialização}

ncolfwd = 0; {Número de colunas com "forward"}

Zera o vetor *temp*;

while (*ncolfwd* < *ncol*) **do**

$j = 0;$

Procura uma coluna j tal que $nmod_j = 0;$

if ($j > 0$) **then**

{Realiza o "forward" da equação j }

$aux = temp_j / l_{jj};$

for $k \in struct(L_{*j})$ **do**

$temp_k = temp_k - aux l_{kj};$

end-for

$b_j = aux;$ {Armazena a componente j }

$ncolfwd = ncolfwd + 1;$

{Obtém o processador responsável pelo pai da coluna j }

$proc = mapeq(pai(j));$

if ($proc = p$) **then**

{A coluna pai da coluna j pertence a este processador }

$k = pai(j);$

$nmod_k = nmod_k - 1;$

else

{O pai da coluna j pertence a outro processador}

Envia para o processador $proc$ os elementos do vetor $temp$
com índices $k \in struct(L_{*j});$

end-if;

else

{Recebimento de vetores de atualização de outros processadores}

Espera o aviso de chegada de um vetor de atualização de "forward";

Recebe o vetor de atualização de uma coluna k cuja coluna pai pertence ao processador p ;

Adiciona esta atualização ao vetor $temp$;

$l = pai(k)$;

$nmod_l = nmod_l - 1$;;

end-if;

end-while;

end.

No algoritmo (IV.5) o vetor b , após a fase de "forward", encontra-se distribuído entre os processadores e está pronto para a fase de retrossubstituição.

IV.5.5 Retrossubstituição paralela

A fase de retrossubstituição consiste na solução do vetor x para o sistema triangular superior $L'x = y$. Na implementação computacional o vetor y , obtido na fase de "forward", e o vetor x ocupam as mesmas posições de memória.

Como somente a matriz L triangular inferior é armazenada em memória, por condição de simetria, tem-se:

$$l'_{ij} = l_{ji} \quad (\text{IV.8})$$

No algoritmo (IV.6) tem-se a retrossubstituição serial de um sistema de equações esparso.

Algoritmo IV.6 Retrossubstituição serial para um sistema esparso

begin

for $i = ncol$ **downto** 1 **do**

$aux = y_i$;

for $j \in struct(L_{*i})$ **do**

$$aux = aux - x_j l_{ji};$$

end-for;

$$x_i = aux/l_{ii};$$

end-for;

end.

Para a fase de retrossubstituição, executada em processadores paralelos com memória distribuída, tem-se:

- As colunas da matriz L dos coeficientes de Cholesky e o vetor dos coeficientes y da fase de "forward" encontram-se distribuídos entre os processadores;
- As equações internas aos processadores podem ser resolvidas em paralelo, sem comunicação entre os processadores;
- As equações de fronteira precisam receber as contribuições do vetor x de outros processadores responsáveis pelos subdomínios adjacentes.

Uma equação j é dita interna a um subdomínio, de um processador p , quando:

$$mapeq(j) = p \quad (\text{IV.9})$$

$$mapeq(pai(j)) = p \quad (\text{IV.10})$$

e

$$filhos(j) = \{k \mid pai(k) = j\} \quad (\text{IV.11})$$

$$mapeq(k) = p, \text{ para todo } k \in filhos(j) \quad (\text{IV.12})$$

ou seja, para que uma equação seja interna a um processador, o seu pai e todos os seus filhos na árvore de eliminação, devem pertencer ao processador p .

Assim, na fase de retrossubstituição distribuída quando o pai de uma equação j pertence a outro processador, então esta equação deverá receber as contribuições do vetor solução x , que afetam os elementos do vetor y , do processador responsável pela equação. Se uma equação k é o pai da equação j e estas equações pertencem a processadores diferentes, então os elementos do vetor x a serem enviados para o processador $p = \text{mapeq}(j)$ são $\{x_i \mid i \in \{\text{struct}(L_{*k}) \cup k\}\}$. No algoritmo (IV.7) tem-se a retrossubstituição paralela. Neste algoritmo o vetor $mask$ é um vetor auxiliar de dimensão $ncol$, que é inicialmente igual a um, para todas as colunas cujo pai pertence ao processador p e zero, em caso contrário. As colunas de fronteira j terão valor $mask_j = 1$, quando receberem as atualizações do vetor solução do processador responsável pela coluna $\text{pai}(j)$.

Algoritmo IV.7 Retrossubstituição paralela em máquinas com memória distribuída.

begin

{Inicializa o vetor $mask$ }

for $j = 1$ **to** $ncol$ **do**

$mask_j = 0$;

$jpai = \text{pai}(j)$;

$proc = \text{mapeq}(jpai)$;

if ($proc = p$) **then**

$mask_j = 1$;

end-if;

end-for;

{Fase de retrossubstituição}

for $j = ncol$ **downto** 1 **do**

{Testa se a equação j está pronta para a retrossubstituição, $mask_j = 1$ }

while ($mask_j = 0$) **do**

Espera o aviso de chegada de um vetor contendo elementos do vetor x de outro processador, referentes a uma coluna k ,

Recebe os elementos $\{x_i \mid i \in \{struct(L_{*k}) \cup k\}\}$;

{Atualiza o vetor $mask$ para todas as colunas filhas da coluna k na árvore de eliminação}

for $l = j$ **downto** 1 **do**

if ($pai(l) = k$) **then**

$mask_l = 1$;

end-if;

end-for;

end-while;

{Processa a retrossubstituição para a coluna j }

$aux = y_j$;

for $i \in struct(L_{*j})$ **do**

$aux = aux - x_i l_{ij}$;

end-for;

$x_j = aux / l_{jj}$;

Determina o número de filhos da coluna j na árvore de eliminação;

Envia para os processadores responsáveis pelas equações filhas da equação j na árvore de eliminação os elementos

$\{x_i \mid i \in \{struct(L_{*j}) \cup j\}\}$;

end-for;

end.

IV.5.6 Cálculo de resultantes de tensões em paralelo

Na fase de cálculo de resultantes de tensões, cada processador calcula independentemente as resultantes de tensões de seus respectivos elementos e faz a adição das forças nodais internas resistentes dos elementos, para o vetor de forças nodais internas resistentes da estrutura. Após isto, a função `gdsum()` do cubo (ver Intel, 1991) faz a soma global, em dupla precisão, das contribuições do vetor de forças resistentes da estrutura armazenadas em cada processador.

O vetor de forças internas resistentes nodais da estrutura é utilizado para o cálculo do vetor de forças nodais desequilibradas da estrutura.

CAPÍTULO V

FORMULAÇÃO NÃO-LINEAR FÍSICA E GEOMÉTRICA

V.1 Introdução

Neste capítulo apresenta-se a formulação não-linear física e geométrica para o elemento finito de pórtico espacial.

As estratégias de solução de controle de carga-deslocamento, para a solução incremental iterativa, do sistema de equações não-lineares de elementos finitos, são também apresentadas.

Na análise elasto-plástica o endurecimento isotrópico do material é considerado no método plástico nodal.

V.2 Solução incremental iterativa

Considerando-se a estrutura em equilíbrio, numa etapa m de aplicação das cargas externas, tem-se:

$$F^m - P^m = 0 \quad (\text{V.1})$$

onde F^m é o vetor das forças internas da estrutura, obtido a partir da contribuição de todos os elementos finitos da estrutura na etapa de incremento de carga m e P^m é o vetor resultante das cargas aplicadas da estrutura na etapa m .

O vetor de carga externas da estrutura na etapa m é definido pela combinação linear de n vetores de referência de cargas, da seguinte maneira:

$$P^m = \lambda_1^m P_1 + \lambda_2^m P_2 + \dots + \lambda_n^m P_n \quad (\text{V.2})$$

onde P_l é o vetor de referência de carga para a carga l aplicada na estrutura e λ_l^m é coeficiente multiplicador da carga l na etapa de incremento de carga m .

Os coeficientes multiplicadores das cargas de referência possibilitam variar as cargas de acordo com as necessidades da análise não-linear, principalmente nas análises de cargas limites de instabilidade estrutural.

Na solução incremental iterativa, assume-se a premissa de que os deslocamentos e as forças internas resultantes de tensões da estrutura sejam conhecidos na etapa de carga m e deseja-se a solução para a etapa de incremento de carga $m+1$ escolhido. Então, tem-se a condição de equilíbrio:

$$F^{m+1} - P^{m+1} = 0 \quad (\text{V.3})$$

onde:

$$F^{m+1} = F^m + \Delta F^{m+1} \quad (\text{V.4})$$

onde ΔF^{m+1} é o incremento do vetor das forças internas correspondentes aos esforços internos resultantes de tensões e deslocamentos da etapa m para a etapa $m+1$.

O incremento do vetor de deslocamentos da estrutura ΔU^{m+1} pode ser obtido de modo aproximado, utilizando-se a matriz de rigidez tangente da estrutura (Bathe, 1982):

$$K^m \Delta U^{m+1} = P^{m+1} - F^m \quad (\text{V.5})$$

onde K^m é a matriz da tangente da estrutura, obtida a partir do estado geométrico e físico da estrutura na etapa m .

O vetor de deslocamentos da estrutura, na etapa de carga $m+1$, é dado por :

$$U^{m+1} = U^m + \Delta U^{m+1} \quad (\text{V.6})$$

V.3 Métodos de Newton-Raphson

Considera-se o método de Newton-Raphson com iterações dentro de uma etapa incremental de carga. As equações (V.5) e (V.6) podem ser reescritas na seguinte forma:

$$K^{m+1,i} \Delta U^{m+1,j+1} = P^{m+1} - F^{m+1,i} \quad (\text{V.7})$$

$$U^{m+1,j+1} = U^{m+1,j} + \Delta U^{m+1,j+1} \quad (\text{V.8})$$

onde i é uma iteração dentro da etapa de carga $m+1$. As condições iniciais são:

$$U^{m+1,0} = U^m \quad (\text{V.9})$$

$$F^{m+1,0} = F^m \quad (\text{V.10})$$

$$K^{m+1,0} = K^m \quad (\text{V.11})$$

Na equação (V.7) pode-se evitar o cálculo e a fatoração da matriz tangente a cada iteração. No método de Newton-Raphson modificado, a matriz tangente é calculada somente no início de cada passo de carga, mantendo-se inalterada nas iterações subsequentes.

No método das tensões iniciais a matriz tangente é calculada somente uma vez no início do processo incremental iterativo. Este método corresponde a uma linearização da resposta da estrutura em relação a configuração inicial da estrutura e pode resultar numa solução bastante demorada ou então divergente.

No método de Newton-Raphson com controle de deslocamento (ver Apêndice D), escolhe-se um deslocamento prescrito para um grau de liberdade nodal representativo da estrutura e neste método determina-se o coeficiente multiplicador do vetor de referência para uma das cargas na equação (V.2). Este método é bastante útil na determinação de carga de colapso em estruturas, pela sua facilidade de convergência, em relação ao método de Newton-Raphson com controle de carga.

V.4 Critérios de convergência

No processo incremental iterativo, com controle de carga ou deslocamento em cada iteração é preciso verificar se o algoritmo atingiu a convergência para as tolerâncias prescritas ou divergiu, excedendo o limite de iterações estabelecido por passo de carga.

Existem três critérios de convergência utilizados para teste de convergência (Bathe, 1982):

- critério de deslocamentos;
- critério de forças desequilibradas;
- critério de energia.

No critério de deslocamentos, compara-se a razão entre as normas Euclidianas dos deslocamentos incrementais e dos deslocamentos totais da etapa de carga $m+1$, na iteração i , com uma tolerância ε_d :

$$\left\| \frac{\Delta U^{m+1,i}}{U^{m+1,i}} \right\| \leq \varepsilon_d \quad (\text{V.12})$$

No critério de convergência das forças desequilibradas, compara-se a razão entre as normas Euclidianas das forças desequilibradas do passo de carga $m+1$, na iteração i e do início do incremento de carga $m+1$, com uma tolerância de forças ε_f :

$$\left\| \frac{P^{m+1} - F^{m+1,i-1}}{P^{m+1} - F^m} \right\| \leq \varepsilon_f \quad (\text{V.13})$$

O critério de convergência de energia, proposto por Bathe (1983), procura englobar ao mesmo tempo os critérios de convergência de deslocamentos e forças desequilibradas.

No presente estudo utiliza-se como critérios de convergência o atendimento simultâneo dos critérios de deslocamentos e de forças desequilibradas.

V.5 Formulação do elemento de pórtico espacial

V.5.1 Revisão teórica

A resposta não-linear de um elemento de pórtico espacial submetido a forças incrementais é considerada, utilizando-se a formulação Lagrangeana atualizada (Bathe 1982).

A matriz de rigidez tangente do elemento de pórtico espacial consiste na soma da matriz elástica K^e , da matriz geométrica K^g e de uma matriz de redução plástica K^p :

$$K = K^e + K^g + K^p \quad (\text{V.14})$$

A matriz elástica K^e é a matriz convencional com doze graus de liberdade, desprezando-se as deformações de cisalhamento.

A matriz de redução plástica é baseada no método plástico nodal e considera efeito de endurecimento isotrópico do material (Ueda e Fujikubo, 1992).

V.5.2 Formulação não-linear geométrica

Na formulação Lagrangeana atualizada para o elemento de pórtico espacial (Bathe e Bolourchi, 1979), para grandes deslocamentos e pequenas deformações, tem-se:

- o elemento pode sofrer grandes deflexões e rotações, porém as deformações extensionais e rotacionais devem ser pequenas em relação ao comprimento do elemento e da unidade, respectivamente;
- hipótese das seções planas de Navier-Bernoulli.

A matriz de rigidez geométrica é deduzida pelo princípio dos trabalhos virtuais, utilizando-se o segundo tensor de Piola-Kirchoff incremental de tensões e os tensores de deformação não-lineares incrementais de Green.

Um sistema de referência móvel local acompanha a deformação do elemento no sistema Lagrangeano atualizado (Bathe e Bolourchi, 1979).

A matriz de rigidez geométrica utilizada neste trabalho foi deduzida por Benjamin (1982).

Nesta formulação os deslocamentos transversais são expressos por uma função polinomial de terceira ordem e os axiais por uma função linear e podem substituir a teoria geral de vigas-colunas, utilizando modelos de elementos finitos bem discretizados.

V.5.3 Formulação não-linear elasto-plástica

V.5.3.1 Introdução

Apresenta-se uma formulação elasto-plástica para o elemento de pórtico espacial, utilizando-se o método plástico nodal e superfícies de escoamento de esforços resultantes de tensões.

A análise elasto-plástica perfeita tem sido estudada por Argyris et alii (1982), Orbison et alii (1982) e Ueda e Fujikubo (1991).

A análise elasto-plástica, incluindo efeito de endurecimento ("strain-hardening") do material, tem sido considerada por Hilmy e Abel (1985), Ueda e Rashed (1991), Ueda e Fujikubo (1992) e Fujikubo et alii (1991). No caso da análise elasto-plástica com endurecimento, a resistência última da estrutura é detectada pela presença de deformações plásticas excessivas.

V.5.3.2 Hipóteses básicas

As seguintes hipóteses básicas são assumidas na formulação elasto-plástica do elemento finito de pórtico espacial:

- As deformações plásticas são concentradas nos nós do elemento finito, enquanto que o interior do elemento permanece em regime elástico;
- A condição de plasticidade é expressa em termos de esforços resultantes de tensões, nos nós do elemento, como um potencial plástico (ver Chen e Han, 1988);
- A taxa de endurecimento, para os nós com deformações plásticas, é considerada igualando-se o trabalho plástico nodal com o trabalho obtido pela distribuição elasto-plástica real de tensões no interior do elemento (Ueda e Fujikubo, 1992);
- Somente o endurecimento isotrópico (Chen e Han, 1988) será considerado neste trabalho.

V.5.3.3 Função de escoamento

A seguinte função é utilizada como função de escoamento dos nós de extremidade do elemento, considerando o efeito de endurecimento isotrópico:

$$F(S, k(\bar{\varepsilon}^p)) = F(S) - k(\bar{\varepsilon}^p) \quad (\text{V.15})$$

onde $F(S)$ é a função de escoamento em termos de resultantes de tensão, k é a função de endurecimento linear em relação à deformação plástica equivalente $\bar{\varepsilon}^p$. Quando $\bar{\varepsilon}^p = 0$, $k = 1$. A função de escoamento utilizada para os nós de extremidade do elemento, desprezando-se a influência da torção, é a seguinte:

$$F(S) = \left| \frac{N}{N_p} \right| + \left| \frac{M_y}{M_{py}} \right| + \left| \frac{M_z}{M_{pz}} \right| \quad (\text{V.16})$$

sendo N o esforço normal; M_y e M_z os momentos fletores em relação os eixos y e z da seção, respectivamente; N_p o esforço axial de plastificação; M_{py} e M_{pz} os momentos fletores plásticos em relação aos eixos y e z , respectivamente.

Os esforços de plastificação de uma seção são calculados de seguinte forma:

$$\begin{aligned} N_p &= A \sigma_e \\ M_{py} &= Z_y \sigma_e \\ M_{pz} &= Z_z \sigma_e \end{aligned} \quad (\text{V.17})$$

sendo σ_e a tensão de escoamento longitudinal do material, A é a área da seção, Z_y e Z_z são os módulos plásticos de seção em relação aos eixos y e z , respectivamente.

A área e os módulos plásticos para seções usualmente utilizados são:

- Seção retangular:

$$A = bh; Z_y = hb^2 / 6; Z_z = bh^2 / 6 \quad (\text{V.18})$$

- Seção circular vazada com raio interno r e raio externo R :

$$A = \pi(R^2 - r^2); Z_y = Z_z = \frac{4}{3}(R^3 - r^3) \quad (\text{V.19})$$

- Seção elíptica vazada (a e b são as semi-distância externas em relação aos eixos Z e Y , respectivamente e as internas a_1, b_1 , respectivamente):

$$A = \pi(ab - a_1b_1); Z_y = \frac{4}{3}(a^2b - a_1^2b_1); Z_z = \frac{4}{3}(ab^2 - a_1b_1^2) \quad (\text{V.20})$$

V.5.3.4 Determinação da taxa de endurecimento seccional

Pela hipótese da seções planas, a deformação axial num ponto genérico da seção é dado por:

$$\varepsilon(y, z) = \varepsilon_0 + k_y z - k_z y \quad (\text{V.21})$$

sendo ε_0 a deformação centróide da seção, k_y e k_z as curvaturas da seção em relação aos eixos neutros y e z , respectivamente. Tem-se que:

$$\varepsilon_0 = N / EA$$

$$k_y = M_y / EI_y \quad (\text{V.22})$$

$$k_z = M_z / EI_z$$

sendo N o esforço normal atuante na seção; M_y e M_z os momentos fletores atuantes na seção em torno dos eixos y e z , respectivamente; A é a área da seção transversal; I_y e I_z são os momentos de inércia em relação aos eixos y e z da seção, respectivamente.

Considerando a função de escoamento como um potencial plástico, pelo princípio da normalidade de Drucker (ver Chen e Han, 1988), as deformações plásticas são normais à superfície de escoamento e ortogonais ao vetor de incremento de esforços resultantes de tensões, tem-se:

$$d\varepsilon^p = d\lambda \phi \quad (\text{V.23})$$

onde $d\lambda$ é a medida da magnitude das deformações plásticas e ϕ é o vetor gradiente da superfície de escoamento, dado por:

$$\phi = \partial F / \partial S \quad (\text{V.24})$$

Como $\bar{\varepsilon}^p$ é a deformação plástica generalizada equivalente, cujo incremento é dado igualando-se o trabalho realizado por $k(\bar{\varepsilon}^p)$, com o trabalho realizado pelo estado atual dos esforços resultantes de tensões atuantes na seção:

$$k(\bar{\varepsilon}^p) \cdot d\bar{\varepsilon}^p = S' d\varepsilon^p \quad (\text{V.25})$$

Assim, usando a equação (V.23) na equação anterior, tem-se:

$$d\bar{\varepsilon}^p = cd\lambda \quad (\text{V.26})$$

onde:

$$c = S' \phi / k(\bar{\varepsilon}^p) \quad (\text{V.27})$$

Para função de escoamento homogêneas, o coeficiente c é igual a 1 (Ueda e Fujikubo, 1992).

Para a condição da seção em escoamento, tem-se:

$$dF(S, k(\bar{\varepsilon}^p)) = dF(S) - dk(\bar{\varepsilon}^p) = \phi' dS - \frac{\partial k}{\partial \varepsilon_p} d\bar{\varepsilon}^p = 0 \quad (\text{V.28})$$

onde:

$$\frac{\partial k}{\partial \varepsilon_p} = H'_s \quad (\text{V.29})$$

e H'_s representa a taxa de endurecimento seccional e será obtida a seguir.

Considera-se que o incremento de forças resultantes de tensões, causadas pelo efeito de endurecimento, é dado por:

$$ds = \begin{Bmatrix} dN \\ dM_y \\ dM_z \end{Bmatrix} = H'_s \begin{Bmatrix} Ad\varepsilon_0^p \\ I_y dk_y \\ I_z dk_z \end{Bmatrix} \quad (\text{V.30})$$

onde H' é módulo plástico de endurecimento dado por:

$$H' = \frac{E E'}{E - E'} \quad (\text{V.31})$$

onde E é o modulo tangente elástico e E' o módulo tangente de endurecimento, correspondentes ao caso unidimensional de tensões, conforme mostrado na figura (V.1).

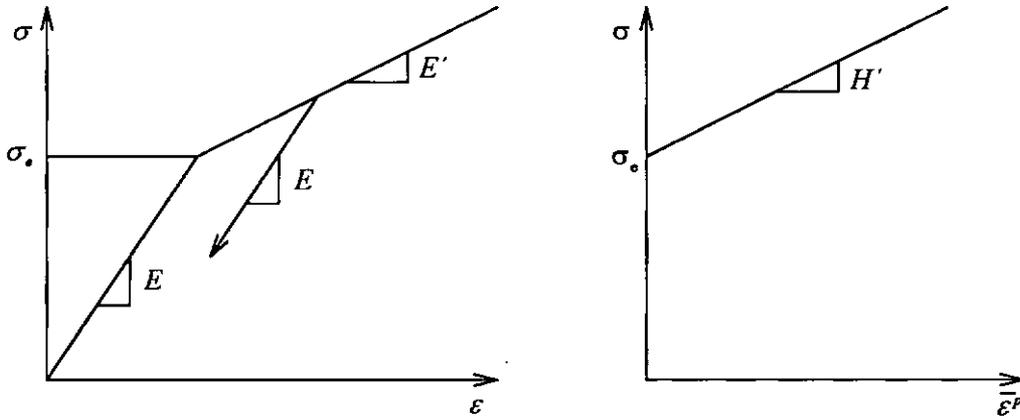


Figura V.1 Módulos tangente elástico e plástico para um caso unidimensional

Usando a expressão da superfície de escoamento (V. 16) tem-se:

$$\phi = \frac{\partial F}{\partial S} = \left\{ \frac{\text{sign}(N)}{N_p} \quad \frac{\text{sign}(M_y)}{M_{py}} \quad \frac{\text{sign}(M_z)}{M_{pz}} \right\}^t \quad (\text{V.32})$$

Substituindo as equações (V.26), (V.30) e (V.32) na equação (V.28), resulta:

$$H'_s = cH' \left(\frac{A}{N_p^2} + \frac{I_y}{M_{py}^2} + \frac{I_z}{M_{pz}^2} \right) \quad (\text{V.33})$$

V.5.3.5 Matriz tangente elasto-plástica

Considerando o comportamento inelástico do elemento, o vetor incremental de deslocamentos nodais, em uma dada configuração de equilíbrio, é dado pela soma das componentes elásticas e plásticas:

$$du = du^e + du^p \quad (\text{V.34})$$

onde o vetor elástico incremental de deslocamentos está relacionado com as forças nodais incrementais dS do elemento pela relação:

$$K^e du^e = dS \quad (\text{V.35})$$

onde K^e é a matriz de rigidez tangente elástica do elemento, considerando-se por simplicidade não-linearidade física e linearidade geométrica.

Designando as forças nodais no nó i do elemento de S_i , a condição de escoamento (V.16) do nó i , considerando o efeito de endurecimento, é:

$$F_i(S_i, k_i(\bar{\varepsilon}_i^p)) = F_i(S_i) - k_i(\bar{\varepsilon}_i^p) = 0 \quad (\text{V.36})$$

O incremento da deformação plástica $\bar{\varepsilon}_i^p$ equivalente para o nó i é dado por:

$$k_i(\bar{\varepsilon}_i^p) d\bar{\varepsilon}_i^p = S_i' d\varepsilon_i^p \quad (\text{V.37})$$

Como a função de escoamento F_i é um potencial plástico, pelo critério da normalidade de Drucker, tem-se que o vetor incremental de deslocamentos plásticos, do nó i , é dado por:

$$du_i^p = d\lambda_i \phi_i \quad (\text{V.38})$$

onde $d\lambda_i$ é a magnitude do incremento do deslocamento plástico e ϕ_i é o vetor gradiente da superfície de escoamento do nó i , dado por:

$$\phi_i = \partial F_i / \partial S_i \quad (\text{V.39})$$

Na equação (V.38) se o valor de $d\lambda_i$ for negativo, tem-se a condição de descarregamento elástico.

Na condição de incremento de carga, a seguinte condição precisa ser satisfeita no nó i :

$$dF_i = \phi_i' dS_i - H_{ii}' d\bar{\varepsilon}_i^p = 0 \quad (\text{V.40})$$

onde H_{ii}' é a taxa de endurecimento da seção do nó i .

A relação existente entre a deformação plástica incremental equivalente $d\bar{\varepsilon}_i^p$ e $d\lambda_i$ é dada por:

$$d\bar{\varepsilon}_i^p = h_i d\lambda_i \quad (\text{V.41})$$

onde h_i é um parâmetro que converte o efeito de endurecimento, na região do interior do elemento, para a extremidade nodal do elemento (Ueda e Fujikubo, 1992).

Na figura (V.2) tem-se o elemento plastificado numa região de comprimento l_i^p em relação ao nó i . A determinação do coeficiente h_i será feita no próximo subitem deste capítulo.

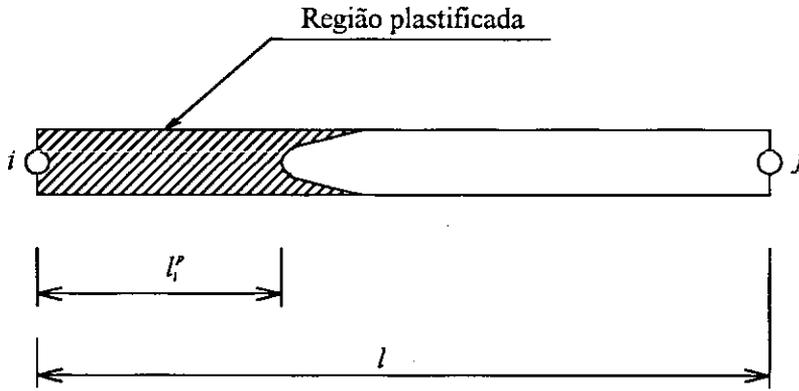


Figura V.2 Elemento parcialmente plastificado.

Usando a equação (V.41) na equação (V.40), tem-se:

$$dF_i = \phi'_i dS_i - H'_m d\lambda_i = 0 \quad (\text{V.42})$$

onde:

$$H'_m = h_i H'_{si} \quad (\text{V.43})$$

Se ambos os nós do elemento estão escoando, então as equações (V.38) e (V.42) tornam-se, respectivamente:

$$du^p = \phi d\lambda \quad (\text{V.44})$$

e

$$\phi^i dS - H'_n d\lambda = 0 \quad (\text{V.45})$$

onde:

$$\phi = \begin{bmatrix} \phi_i & 0 \\ 0 & \phi_j \end{bmatrix} \quad (\text{V.46})$$

$$H'_n = \begin{bmatrix} H'_m & 0 \\ 0 & H'_{nj} \end{bmatrix} \quad (\text{V.47})$$

$$d\lambda = \{d\lambda_i, d\lambda_j\}' \quad (\text{V.48})$$

Usando as equações (V.34) e (V.44), a equação (V.35) pode ser reescrita na forma:

$$dS = K^e(du - \phi d\lambda) \quad (V.49)$$

Usando a equação (V.45) na equação (V.49), então tem-se:

$$d\lambda = (H'_n + \phi' K^e \phi)^{-1} \phi' K^e du \quad (V.50)$$

Substituindo a equação (V.50) na equação (V.49), resulta:

$$dS = (K^e + K^{ep}) du \quad (V.51)$$

onde:

$$K^{ep} = -K^e \phi (H'_n + \phi' K^e \phi)^{-1} \phi' K^e \quad (V.52)$$

é a matriz tangente de redução elasto-plástica. Para a situação de material elasto-plástico perfeito, a matriz H'_n é nula.

V.5.3.6 Determinação da taxa de endurecimento nodal

Por causa do efeito de endurecimento, a região de plastificação estende-se numa região de comprimento l_i^p , em relação ao nó i do elemento, conforme mostrado na figura (V.2).

Para considerar o efeito de endurecimento no método de plastificação nodal, é necessário determinar a região plastificada no interior do elemento e concentrar nos nós de extremidade do elemento o efeito de endurecimento plástico.

A taxa de endurecimento nodal é obtida igualando-se o trabalho plástico nodal incremental com o trabalho elasto-plástico incremental, produzido pela distribuição real de tensões no interior do elemento (Ueda e Fujikubo, 1992). Assim, tem-se que:

$$\int_0^{l_i^p} k(x) d\bar{\varepsilon}^p(x) dx = d\lambda_i \phi'_i S \quad (V.53)$$

Aqui a distribuição da deformação plástica incremental equivalente, na região de plastificação de comprimento l_i^p , é dada por:

$$d\bar{\varepsilon}^p(x) = g(x) d\bar{\varepsilon}_i^p \quad (V.54)$$

onde $g(x)$ é uma função de distribuição de deformação elasto-plástica e $d\bar{\varepsilon}_i^p$ é a deformação plástica equivalente no nó i .

Substituindo a equação (V.54) na equação (V.53), tem-se:

$$d\bar{\varepsilon}_i^p \int_0^{l_i^p} k(x)g(x)dx = d\lambda_i \phi_i^t S \quad (\text{V.55})$$

Usando a equação (V.41) na equação (V.55), resulta:

$$h_i = \phi_i^t S / \int_0^{l_i^p} k(x)g(x)dx \quad (\text{V.56})$$

Ueda e Fujikubo (1992) obtêm a forma explícita da função $g(x)$ para o caso geral de vigas-colunas, onde o coeficiente h_i é obtido por integração numérica no interior do elemento.

Dentro do contexto do método dos elementos finitos, assume-se a hipótese de que a malha esteja suficientemente refinada, para poder representar com precisão o comportamento estrutural do modelo. Assim, as seguintes simplificações podem ser consideradas neste caso:

$$g(x) = 1 \quad (\text{V.57a})$$

$$\bar{\varepsilon}^p(x) = \bar{\varepsilon}_i^p \quad (\text{V.57b})$$

$$k(x) = k_i \quad (\text{V.57c})$$

Logo:

$$h_i = \frac{\phi_i^t S}{l_i^p k_i} \quad (\text{V.58})$$

Usando a equação (V.27) e considerando o comprimento de plastificação do nó i como sendo igual a $l/2$, tem-se finalmente:

$$h_i = \frac{cl}{2} \quad (\text{V.59})$$

CAPÍTULO VI

ESTUDOS DE CASOS

VI.1 Introdução

Neste capítulo apresentam-se estudos de casos de modelos estruturais existentes na literatura e também de algumas plataformas offshore fixas do tipo jaqueta, similares a plataformas existentes nas bacias brasileiras e do golfo do México.

Nos exemplos de jaquetas offshore, os carregamentos de cargas ambientais de corrente e vento foram consideradas como cargas nodais equivalentes. Estas cargas foram avaliadas por um software específico de determinação de cargas ambientais, pertencente ao programa de offshore do PEC da COPPE.

Nas análises de estruturas apresentando resposta pós-crítica com instabilidade, a fatoração de Cholesky precisou ser substituída pela fatoração de Crout.

Em função do contexto acadêmico desta pesquisa, as seguintes verificações de projeto das normas API (ver American Petroleum Institute, 1986 e 1989) não foram consideradas:

- flambagem local nas seções tubulares de pequena espessura;
- efeito de "punching shear" nas juntas tubulares;
- interação solo-estrutura e verificação da resistência das estacas da fundação;
- fadiga nas juntas tubulares;
- efeito dinâmico de cargas ambientais e de terremotos extremos;
- tensões residuais.

Os resultados da versão serial e paralela do programa NLFA ("Non-Linear Frame Analysis") desta tese são apresentados a seguir.

VI.2 Coluna biapoiada com imperfeição inicial

Neste item apresentam-se os resultados da análise não linear elasto-plástica perfeita da coluna com seção tubular da figura (VI.1).

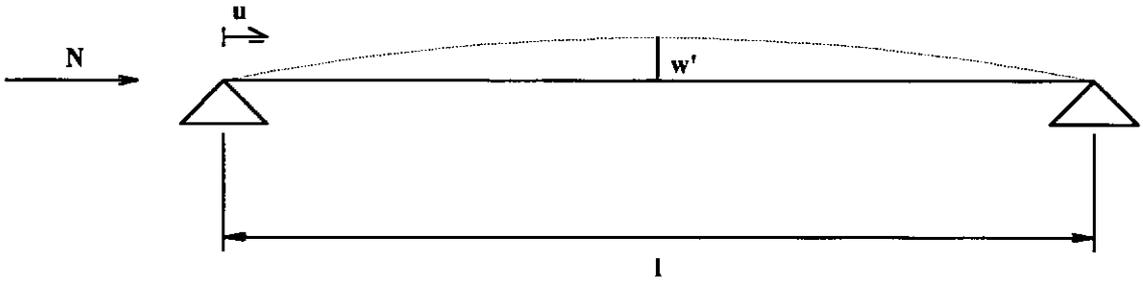


Figura VI.1 Coluna biapoiada com imperfeição inicial

Os dados da coluna tubular da figura (VI.1) são:

$$l/r = 120; w'/l = 0.0015; \sigma_y = 330 \text{ MPa}$$

onde r é o raio de giração da seção tubular.

O gráfico (VI.1) apresenta a resposta do programa NLFA e dos programas USFOS e FENRIS, apresentados por Soreide et alii (1986). Neste gráfico tem-se que $u' = \varepsilon_y l$.

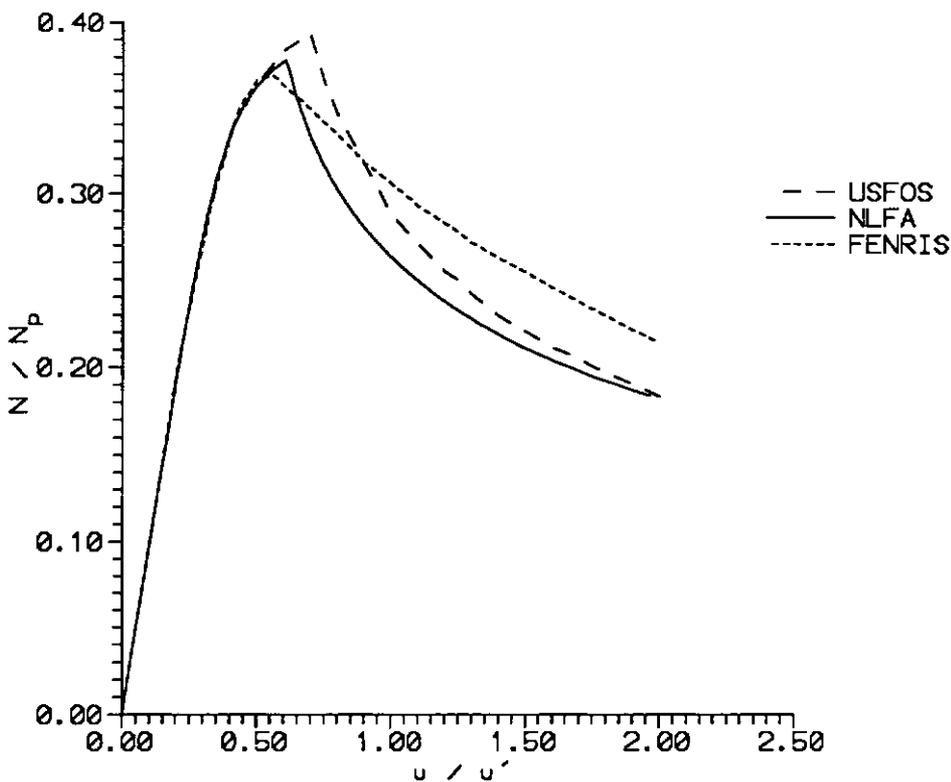


Gráfico VI.1 Curvas de carga versus encurtamento axial para a coluna da figura (VI.1)

O programa USFOS utiliza o método plástico nodal e teoria geral de viga-coluna, sendo que a coluna é modelada com um único elemento. O efeito da imperfeição inicial é simulado pela presença de carga lateral.

O programa FENRIS utiliza o método dos elementos finitos e a não-linearidade física utiliza a plasticidade computacional, com integração numérica de tensões ao longo da seção transversal do elemento.

O programa NLFA utiliza oito elementos finitos e o método plástico nodal da teoria elasto-plástica perfeita. A técnica de controle de deslocamentos é utilizada para obter a resposta pós-crítica da coluna. Na região elástica os resultados dos programas são praticamente iguais.

O programa FENRIS fornece a menor carga limite, porque considera na análise o efeito da plastificação gradual da seção transversal.

O programa NLFA apresenta carga limite menor que o programa USFOS, devido ao fato de que, com apenas um elemento de viga-coluna, o programa USFOS não consegue representar bem o efeito da imperfeição inicial.

Na região pós-crítica os programas NLFA e USFOS subestimam a análise do programa FENRIS, porque utilizam o método plástico nodal, que considera plastificação instantânea da seção transversal.

As diferenças de resultados, nas regiões da carga limite e pós-crítica, entre o método plástico nodal e a plasticidade computacional, com integração numérica de tensões na seção transversal, também acontecem nos exemplos de colunas analisadas por Ueda e Rashed (1991).

VI.3 Coluna biengastada com imperfeição inicial

A coluna biengastada, com imperfeição inicial, da figura (VI.2) é analisada pelo programa NLFA com a técnica de controle de deslocamentos.

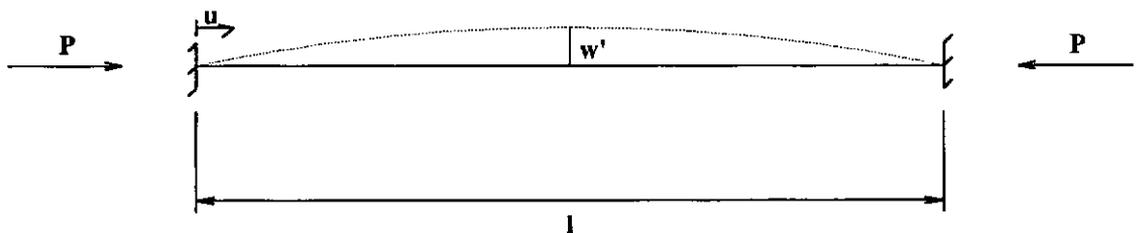


Figura VI.2 Coluna biengastada com imperfeição inicial

Os dados da coluna tubular da figura (VI.2) são:

$$l/r = 144 ; l = 572\text{cm} ; D = 11.4\text{cm} ; t = 2.3\text{mm} ; w'/l = 0.001 ; \sigma_y = 289\text{ MPa}$$

O gráfico (VI.2) apresenta os resultados do programa NLFA e também os resultados das respostas analítica e experimental de Chen e Han (1985).

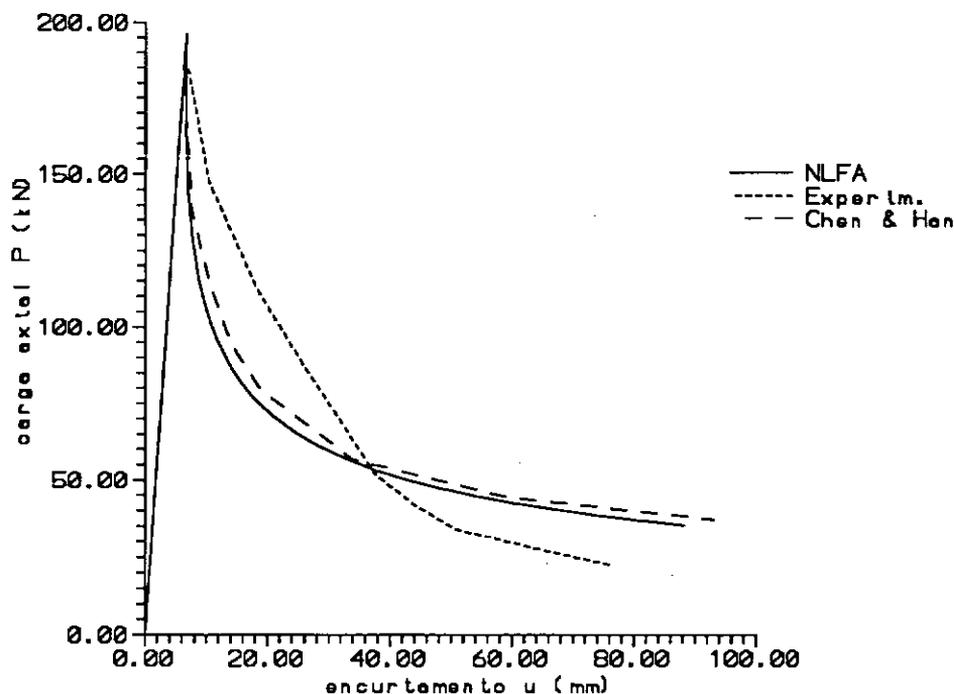


Gráfico VI.2 Curvas de carga versus encurtamento axial para a coluna da figura (VI.2)

A solução analítica de Chen e Han (1985) utiliza o método plástico nodal perfeito e a teoria geral de viga-coluna.

A carga limite apresentada pelo programa NLFA é de 196.2 kN e a carga limite de Chen e Han (1985) é igual a 192.2 kN, ao passo que a carga limite experimental é de 184.3 kN.

Os resultados do programa NLFA estão bastante próximos da solução de Chen e Han (1985), inclusive na região pós-crítica da análise.

Após a carga limite, a curva experimental apresenta uma resistência maior que a solução teórica, por causa do efeito de endurecimento do material. Porém, com o aumento das deformações inelásticas, ocorre flambagem local na seção tubular e a resposta da curva experimental fica abaixo da teórica.

Nas soluções analíticas do programa NLFA e de Chen e Han (1985), a curva apresenta uma concavidade dirigida para a esquerda, logo após a carga limite. Este fenômeno acontece em colunas curtas com uma pequena imperfeição inicial (ver Chen e Han, 1985, p. 119-120).

VI.4 Viga biapoiada com carga distribuída

A viga biapoiada da figura (VI.3) está submetida a uma carga transversal uniformemente distribuída.

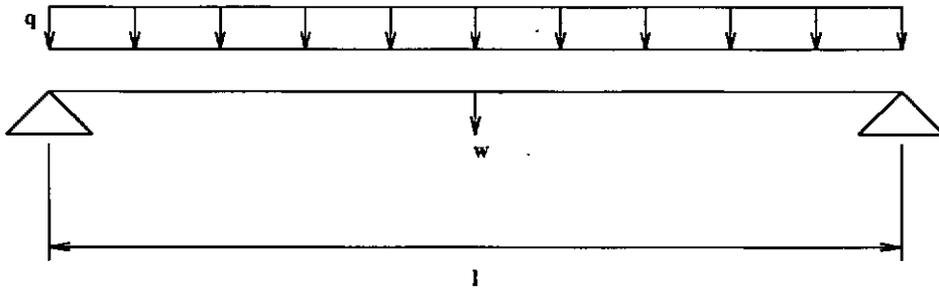


Figura VI.3 Viga biapoiada com carga distribuída

Os dados da viga da figura (VI.3), com seção transversal retangular, são:

$$b = 1 \text{ m}; h = 0.01 \text{ m}; l = 0.5 \text{ m}; E = 220 \text{ GN/m}^2; \sigma_y = 300 \text{ MN/m}^2$$

No programa NLFA foram utilizados 20 elementos finitos, utilizando condição de simetria do modelo. O carregamento uniformemente distribuído foi representado por cargas concentradas nodais equivalentes. Assume-se que o material tenha um comportamento elasto-plástico perfeito. O efeito da força de membrana foi considerado na análise não linear geométrica.

Os resultados da análise estão apresentados no gráfico (VI.3).

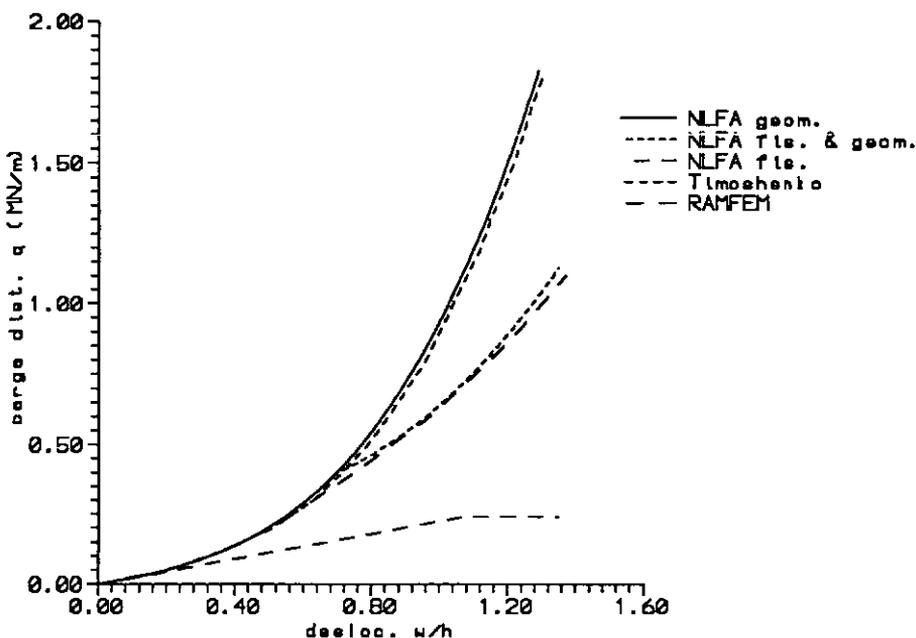


Gráfico VI.3 Curvas carga versus deslocamento transversal para a viga da figura (VI.3)

A curva da resposta não-linear geométrica do programa NLFA está próxima da solução de viga-coluna de Timoshenko e Gere (1963).

Também a resposta do programa, considerando somente não-linearidade física, produz uma carga de plastificação $q = 0.24 \text{ MN/m}$, que está de acordo com a teoria plástica de vigas com linearidade geométrica.

Na análise considerando não-linearidade física e geométrica, o ponto de início de plastificação da curva de resposta do programa NLFA está acima da mesma curva obtida pelo programa RAMFEM, apresentado por Bäcklund (1976). Isto é explicado pelo fato de que o programa RAMFEM utiliza a plasticidade computacional, com integração numérica de tensões na seção transversal.

Portanto, no programa RAMFEM a plastificação ocorre de modo gradual, após alguns pontos de uma seção transversal atingirem a tensão de escoamento. No entanto, no programa NLFA a plastificação ocorre de modo instantâneo, após atingir a superfície de escoamento do material no método plástico nodal.

VI.5 Viga monoengastada com carga concentrada

A viga monoengastada, com seção transversal retangular, da figura (VI.4) está submetida a uma carga concentrada de extremidade. Considera-se o material com comportamento elasto-plástico com endurecimento isotrópico do material.

A estrutura foi analisada assumindo-se comportamento geométrico linear, ou seja, pequenos deslocamentos.

A carga lateral P é aplicada com incrementos proporcionais na análise.

O programa de elementos finitos, considerado para comparação de resultados, utiliza dez elementos ao longo do comprimento da viga e vinte camadas de integração ao longo da altura da viga, de acordo com Ueda e Rashed (1991).

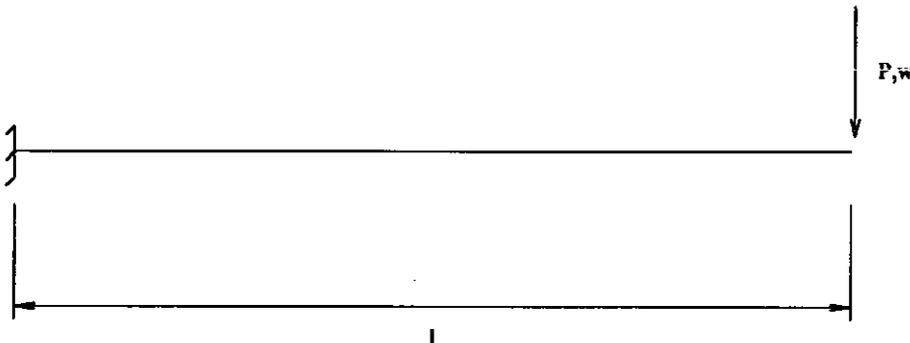


Figura VI.4 Viga monoengastada com carga concentrada

Os dados da viga da figura (VI.4) com seção transversal retangular são:

$b = 12 \text{ mm}$; $h = 20 \text{ mm}$; $l = 300 \text{ mm}$; $E = 210 \text{ kN / mm}^2$; $\sigma_y = 0.3 \text{ kN / mm}^2$; $H' = 10 \text{ kN / mm}^2$

No gráfico (VI.4) tem-se as curvas do programa NLFA e da solução do MEF (Ueda e Rashed, 1991).

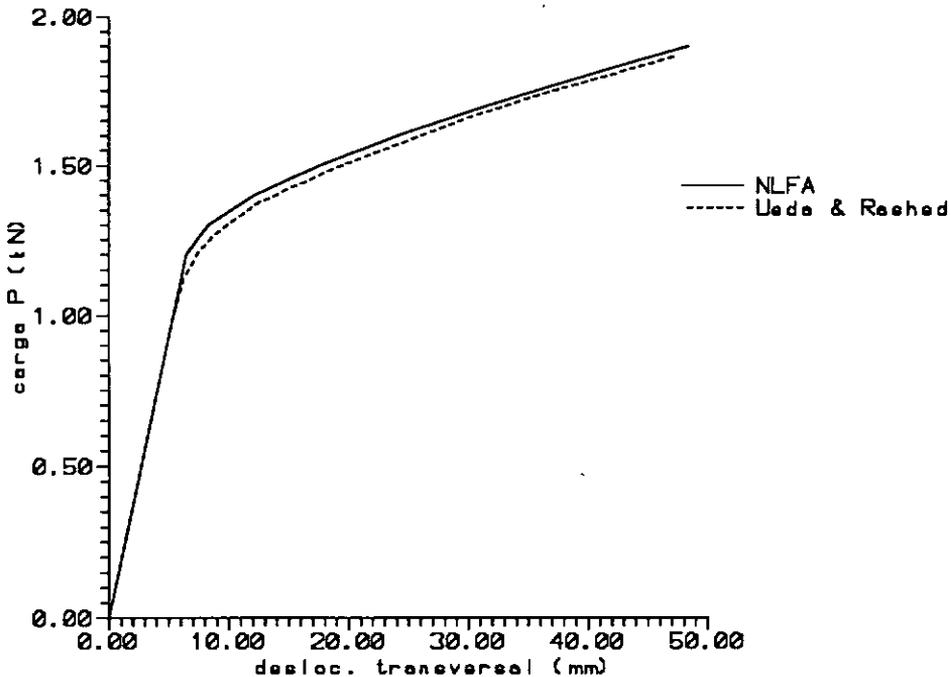


Gráfico VI.4 Curva carga versus desloc. transversal para a viga da figura (VI.4)

Nota-se que, na solução do programa NLFA, a transição plástica gradual da seção transversal não é considerada, porque o método plástico nodal considera uma capacidade de carga resistente maior no início da plastificação. Entretanto, com o incremento da região de plastificação extendendo-se para outras seções, o resultado do método plástico nodal aproxima-se da solução do método dos elementos finitos..

Na análise elasto-plástica perfeita, a carga limite de plastificação é igual a 1.2 kN.

VI.6 Pórtico plano

O pórtico plano da figura (VI.5) é utilizado para comparação de resultados numéricos com ensaios experimentais, apresentados por Moan et alii (1985).

O pórtico é constituído por uma viga transversal horizontal e por dois braços diagonais. O programa USFOS utilizou dois elementos para modelar a viga transversal e dois elementos para cada um dos braços diagonais.

O programa NLFA considerou dez elementos na viga transversal e vinte elementos nos braços diagonais, considerando a simetria do modelo. Imperfeições iniciais foram consideradas na análise com amplitudes máximas de 2 mm.

A técnica de controle de deslocamentos foi utilizada, utilizando como equação de controle o deslocamento horizontal do ponto médio do braço diagonal.

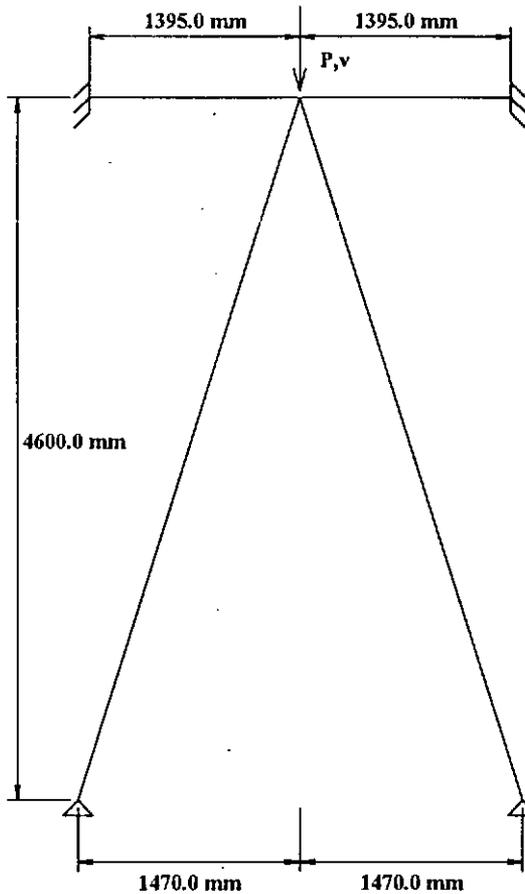


Figura VI.5 Pórtico plano com carga vertical

Os dados dos elementos tubulares, com diâmetro externo D e espessura t , do pórtico da figura (VI.5) são:

- Viga transversal horizontal:

$$D = 219 \text{ mm} ; t = 4.37 \text{ mm} ; \sigma_y = 361 \text{ MPa}$$

- Braços diagonais:

$$D = 120 \text{ mm} ; t = 2.0 \text{ mm} ; \sigma_y = 420 \text{ MPa}$$

O gráfico (VI.5) apresenta os resultados da análise não-linear elasto-plástica perfeita dos programas USFOS (Moan et alii, 1985) e NLFA, assim como a curva do ensaio experimental.

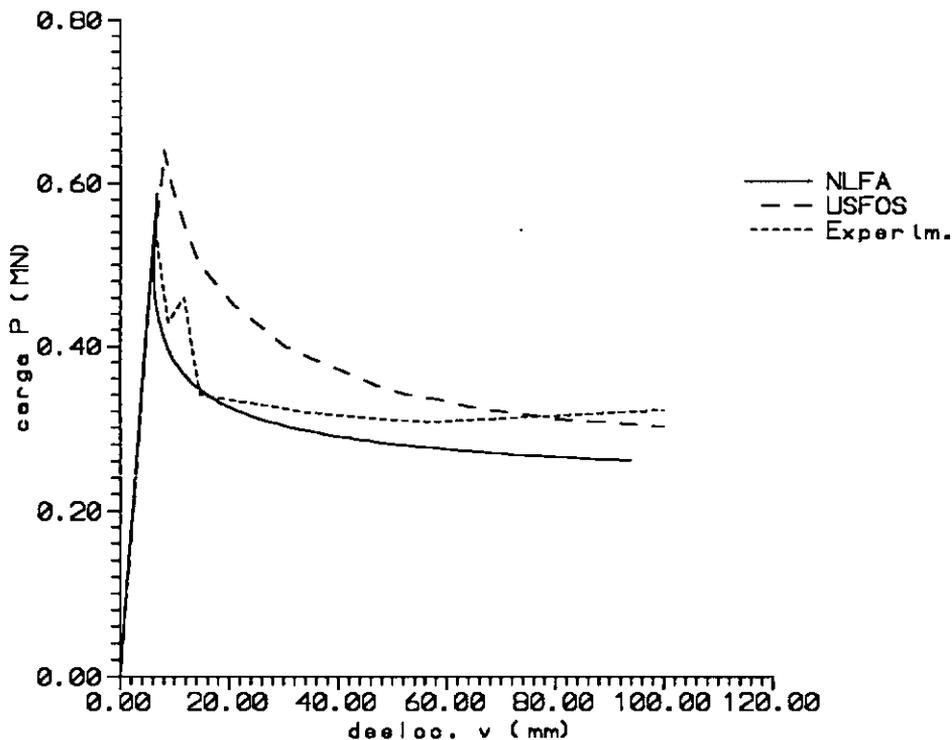


Gráfico VI.5 Curvas carga versus deslocamento transversal para o pórtico da figura (VI.5)

A carga limite do ensaio experimental é de 0.55 MN e a do programa NLFA é de aproximadamente 0.58 MN. Na solução do programa NLFA a curva apresenta uma concavidade dirigida para a esquerda, logo após a carga limite. A carga limite do programa NLFA, sem considerar a presença de imperfeições iniciais, é igual a 0.632 MN.

A carga de colapso do programa USFOS está acima da carga obtida pelo programa NLFA, por causa da não consideração da presença de imperfeições iniciais na análise.

A curva experimental perde resistência logo após a carga de colapso, devido ao efeito da flambagem local que ocorre nos elementos diagonais.

Para a região pós-crítica de grandes deslocamentos, a capacidade resistente da estrutura é subestimada pelos programas USFOS e NLFA, devido possivelmente a não consideração do efeito de endurecimento do material na análise.

VI.7 Pórtico espacial

O pórtico espacial da figura (VI.6) é utilizado para comparação de resultados com o programa FENRIS (ver Moan et alii, 1985).

Os resultados do programa FENRIS foram obtidos utilizando-se um modelo com 108 elementos de pórtico espacial e 93 pontos nodais. O comportamento do material é elasto-plástico perfeito.

O programa NLFA utiliza 180 elementos de pórtico espacial e 165 nós. A tensão de escoamento do material é igual a 330 MPa.

A presença de imperfeições iniciais no modelo não foram consideradas na presente análise.

Este pórtico espacial pode ser considerado com um subsistema de uma jaqueta offshore de quatro pernas.

As tolerâncias de convergência de deslocamentos e forças desequilibradas são respectivamente, $\varepsilon_d = 0.0001$ e $\varepsilon_f = 0.01$

A análise é feita considerando-se dois casos de cargas:

- O primeiro caso de carga é um carregamento vertical resultante de 120 MN, distribuído igualmente entre os nós E, F, G e H.
- O segundo caso de carga é constituído por cargas horizontais atuando na direção x global da estrutura. Estas cargas horizontais estão também aplicadas igualmente nos nós E, F, G e H. Este carregamento é aplicado após a atuação das cargas verticais na análise não-linear.

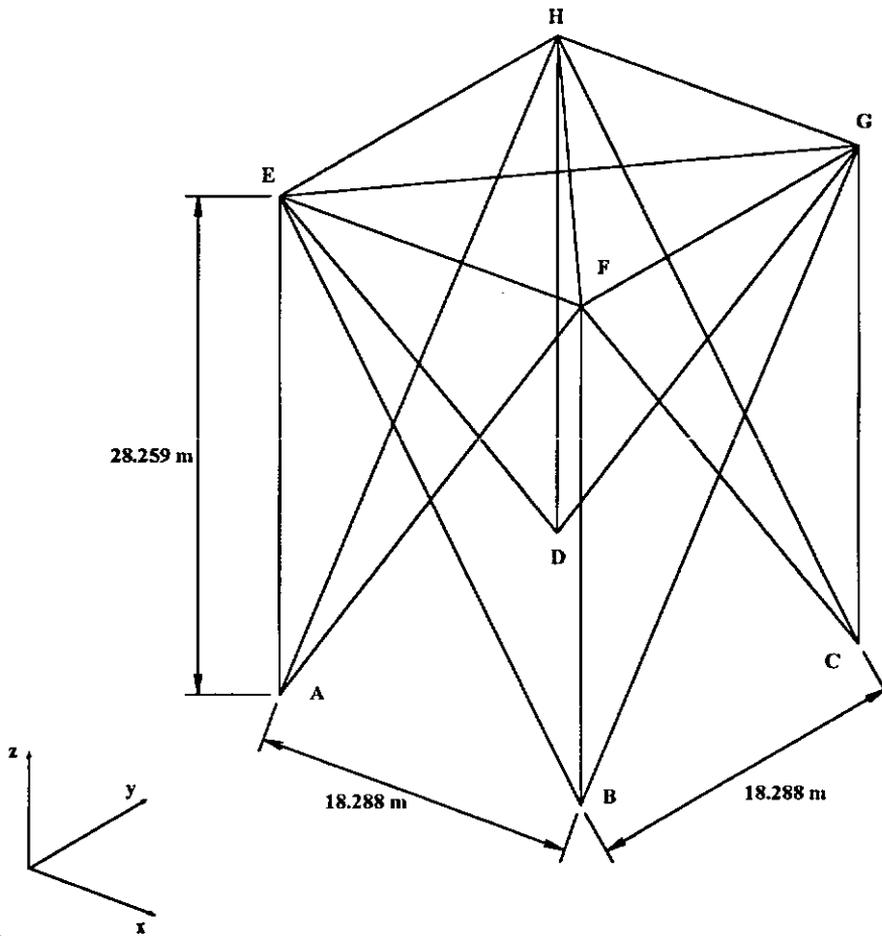


Figura VI.6 Pórtico espacial com cargas concentradas

Os dados gerais do pórtico espacial da figura (VI.6), constituído por elementos com seção transversal tubular, estão apresentados na tabela (VI.1).

Tabela VI.1 Propriedades do pórtico espacial da figura (VI.6)

Tipo de membro	D (m)	t (m)	σ_y (MPa)
Pernas verticais	2.000	0.016	330.0
Elem .horizontais	1.000	0.040	330.0
Braços-x horizontais	1.000	0.040	330.0
Braços-x verticais	1.016	0.038	330.0

No gráfico (VI.6) tem-se as respostas dos programas NLFA e FENRIS, para o deslocamento horizontal no topo do pórtico em função da carga horizontal resultante na estrutura.

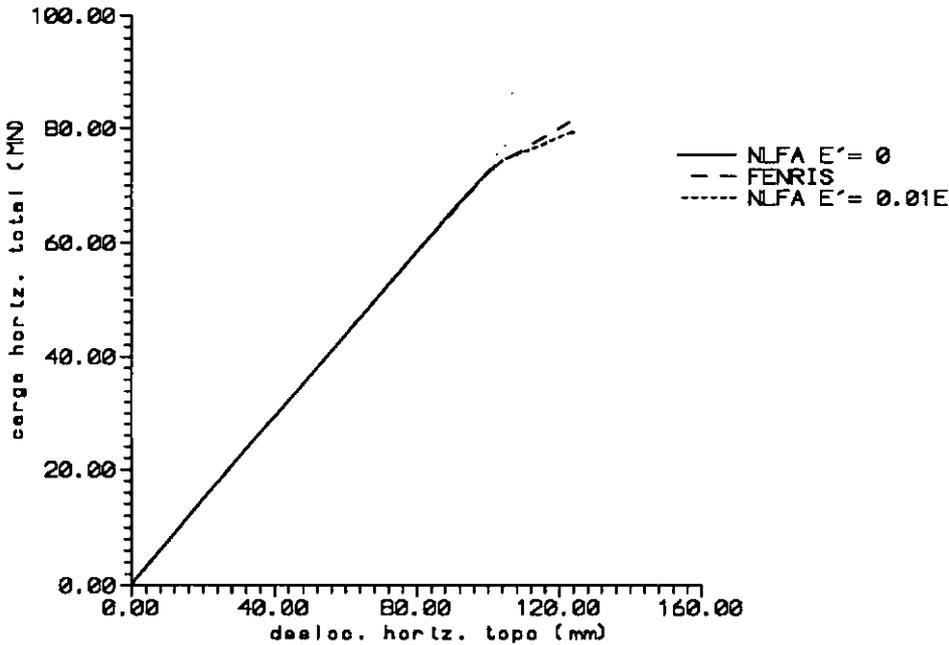


Gráfico VI.6 Curvas carga versus desloc. horiz. no topo do pórtico da figura (VI.6)

Na análise elasto-plástica perfeita, a carga de não convergência do programa NLFA foi obtida quando os elementos das diagonais B-E e C-H plastificaram.

Na região elástica as respostas dos programas NLFA e FENRIS são praticamente coincidentes.

A carga horizontal total de não convergência do programa NLFA, na análise elasto-plástica perfeita é igual a 74 MN e o deslocamento no topo da estrutura é igual a 102 mm.

A carga do colapso do programa FENRIS é igual a 82 MN.

A diferença de resultados pode ser explicada pelo fato de que a superfície de escoamento (V.16), utilizada pelo program NLFA, está à favor da segurança no caso de estruturas tubulares espaciais. Por exemplo, Ueda e Rashed (1991) e Moan et alii (1985) utilizam a seguinte função de escoamento para seções transversais tubulares:

$$F(S) = \left| \sqrt{M_y^2 + M_z^2} \right| / M_p - \cos \pi |N| / 2N_p \quad (\text{VI.1})$$

No entanto, a função de escoamento (V.16) foi mantida por estar à favor da segurança e também por poder ser utilizada para outros tipos de seções transversais.

Ainda no gráfico (VI.6), tem-se a resposta do programa NLFA, com consideração do efeito de "strain-hardening" do material igual a $E' = 0.01 E$. Neste caso não existe carga de instabilidade e a análise de controle de deslocamentos foi feita até para o deslocamento no topo da estrutura igual a 125 mm.

VI.8 Plataforma fixa hipotética

A plataforma fixa hipotética, proposta por Benjamin e Ebecken (1987), com 111 nós e 406 elementos de pórtico espacial, é reanalisada neste item.

A geometria do modelo da plataforma está apresentada na figura (VI.7). 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 atual da MEF, é maior que a modelagem original de Benjamin e Ebecken (1987), para permitir uma determinação mais precisa das cargas de colapso de estrutura, pelo método dos elementos finitos. Também as cargas ambientais de vento e corrente ficam melhor representadas, porque são consideradas na presente análise como cargas nodais equivalentes.

Stewart et alii (1988) apresentam alguns resultados interessantes sobre a necessidade de um refinamento adequado de modelos estruturais de pórtico espacial, pela formulação do método dos elementos finitos.

No entanto, os elementos baseados na teoria de viga-coluna (ver Timoshenko e Gere, 1963), como por exemplo o elemento ISUM de Ueda e Rashed (1991), podem utilizar um único elemento entre juntas da plataforma.

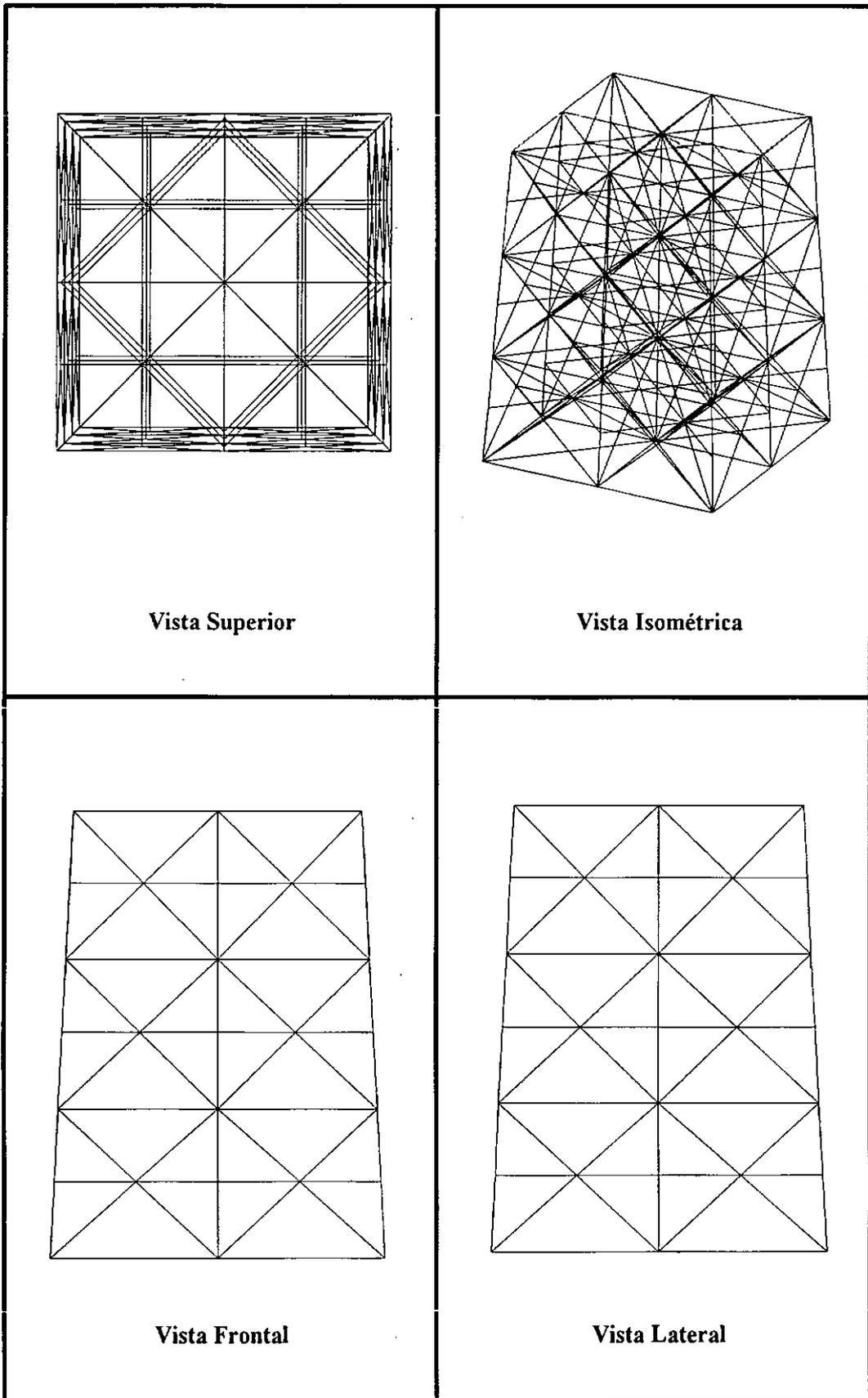


Figura VI.7 Plataforma fixa hipotética.

O elemento ISUM é considerado um superelemento, com capacidade de representar com suficiente precisão o comportamento não-linear físico e geométrico de uma barra entre juntas de uma plataforma.

As cargas atuantes na plataforma são agrupadas em três principais: cargas de peso próprio (carregamento 1), cargas atuantes no deck da plataforma (carregamento 2) e cargas ambientais (carregamento 3).

A resultante total de cargas verticais atuando no deck da plataforma é igual a 112 MN.

Os parâmetros para determinação das cargas de vento e onda na plataforma são:

- Altura do nível de lâmina de água igual a 65 m;
- Onda com 10 metros de altura e período de 11 segundos;
- Corrente com velocidade variando linearmente, de 1.5 m/s na superfície do mar a 0.09 m/s na base da plataforma;
- Velocidade do vento de 37.2 m/s na altura de referência;
- Cargas ambientais consideradas atuando na direção x global da estrutura;
- Coeficiente de arrasto igual a 1.2 e de inércia igual a 2.0, para determinação das cargas nodais equivalentes de corrente, pela fórmula de Morison para membros cilíndricos.

Benjamin e Ebecken (1987) utilizaram valores diferentes de coeficiente de arrasto e de inércia para determinação das cargas ambientais.

Portanto, os resultados desta análise são diferentes das apresentadas por Benjamin e Ebecken (1987), por causa da diferença de refinamento e também as cargas ambientais não são mais as mesmas.

Durante a análise não-linear incremental iterativa, as cargas de peso próprio e cargas do deck são aplicadas e mantidas constantes ao longo da análise (coef. λ_1 e λ_2 iguais a 1.0).

Somente as cargas ambientais (coef. λ_3) são aumentadas gradualmente, até se atingir a carga de colapso da estrutura.

A técnica de controle de deslocamentos é utilizada na análise não linear incremental iterativa. O deslocamento horizontal de um nó do topo da plataforma é escolhido como deslocamento de controle de análise. A atualização da matriz de rigidez da estrutura é feita em cada iteração da análise.

Na tabela (VI.2) tem-se o resultado do algoritmo de decomposição de domínio, utilizando-se quatro processadores.

Tabela VI.2 Decomposição de domínio da MEF da figura (VI.7) com 4 processadores

Subdomínio	1	2	3	4
Núm. de elementos	609	609	609	609
Núm. elem. interface	104	105	121	95
Núm. de nós	534	536	537	534
Núm. de nós interface	23	30	29	20

Na tabela (VI.3) tem-se o resultado da decomposição de domínio, utilizando-se oito processadores na análise.

Tabela VI.3 Decomposição de domínio da MEF da figura (VI.7) 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

Analisando-se os resultados apresentados nas tabelas (VI.2) e (VI.3), conclui-se que o algoritmo de decomposição de domínio utilizado produz uma distribuição balanceada 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 (VI.4) 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 VI.4 Tempos de execução (em segundos) da fase de pré-processamento sequencial da MEF da figura (VI.7)

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 (VI.4), 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.

A versão do programa NLFA, realizando a fase de pré-processamento da análise, utilizando um nó i860 do cubo, foi iniciada e não pode ser completada, pois ainda está faltando a fase de implementação da utilização do disco rápido do sistema concorrente de arquivos (SCA). O SCA do iPSC/860 está descrito no apêndice B.

Na tabela (VI.5) 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 VI.5 Tempos de execução (em segundos) de uma iteração típica de solução da análise da MEF da figura (VI.7)

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 (VI.5) 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 (IV.2) e (IV.3), respectivamente.

Na tabela (VI.6) 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.

Tabela VI.6 "Speed-up" e eficiência de uma iteração típica da análise da MEF da figura (VI.7)

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 (VI.6), que o "speed-up" e a eficiência da implementação paralela diminui com o aumento do número de processadores. Isto é explicado pelo alto custo de comunicação envolvido com o aumento do número de processadores utilizados.

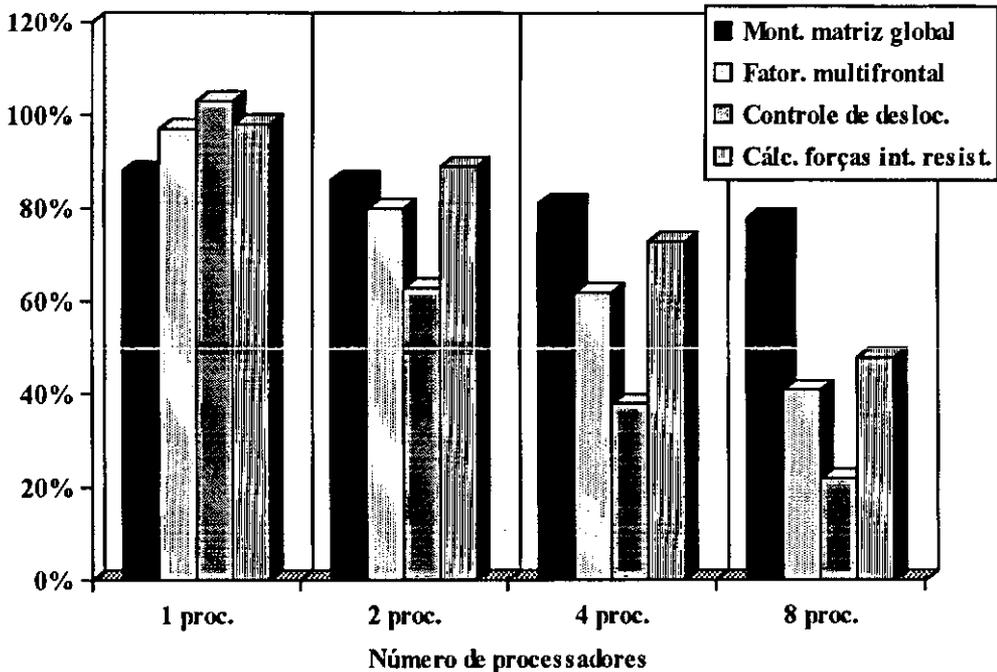


Gráfico VI.7 Eficiência de uma iteração típica da análise da MEF da fig. (VI.7)

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 (VI.7) tem 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.

A eficiência da presente implementação da fatoração multifrontal não difere muito dos resultados obtidos por Ascraft et alii (1990).

Os resultados obtidos nas tabelas (VI.5) e (VI.6) reforçam o potencial de paralelização do método dos elementos finitos em máquinas paralelas com memória distribuída.

Na tabela (VI.7) tem-se os tempos de solução total de CPU do processador supervisor da análise e o tempo total transcorrido para a execução da análise.

Tabela VI.7 Estatística de tempos de execução da MEF da figura (VI.7)

Núm. proc. (p)	Tempo de CPU proc 1	"Speed-up" proc. 1	Tempo exec. total	"Speed-up" tempo total
serial	7069.6	----	7086.0	----
1	7232.0	0.98	7782.0	0.91
2	4492.0	1.57	4763.0	1.48
4	2846.3	2.48	3120.0	2.66
8	2013	3.51	2285.0	3.10

Comparando-se na tabela (VI.7) os tempos de solução entre a versão serial e paralela, utilizando um processador, nota-se que existe na versão paralela a penalização da execução da fase de pré-processamento da análise no "host" i80386.

Para oito processadores tem-se uma eficiência no tempo de CPU total de 44% e no tempo total da rodada uma eficiência de 39%.

No gráfico (VI.8) tem-se a resposta do programa NLFA para a análise elasto-plástica perfeita e grandes deslocamentos para o modelo original de Benjamin e Ebecken (1987), porém com as cargas ambientais alteradas pelos novos coeficientes de inércia e arrasto deste trabalho.

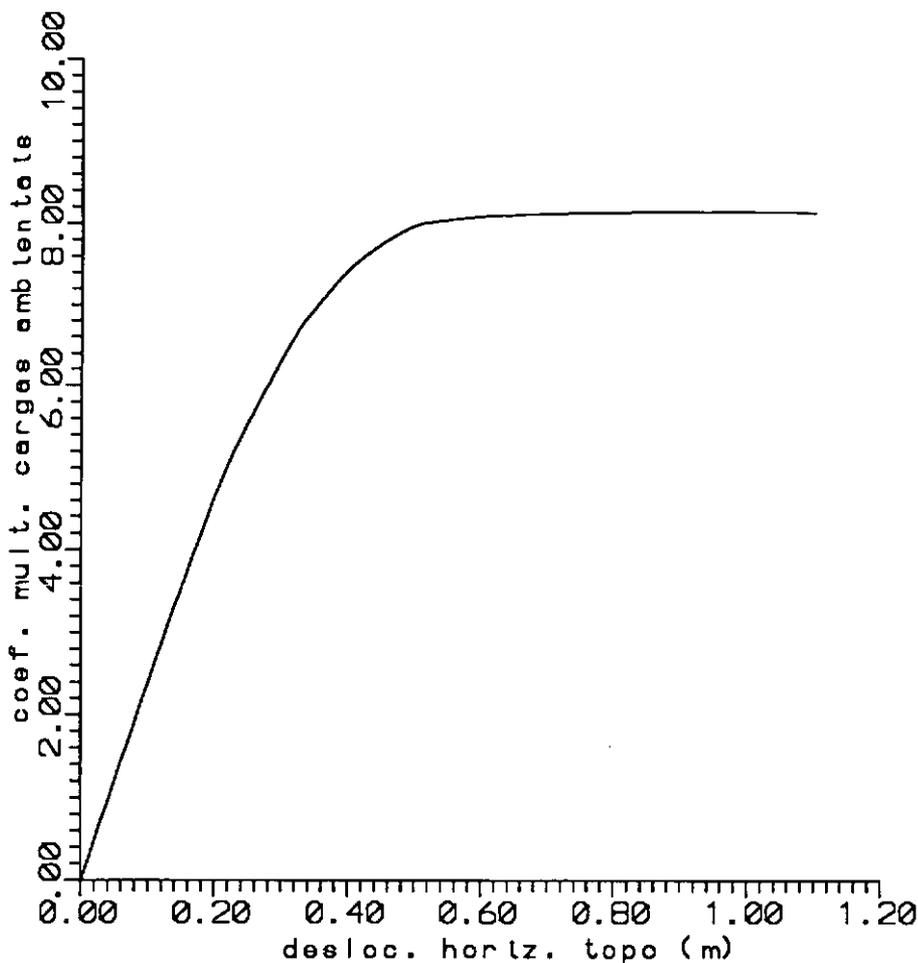


Gráfico VI.8 Resposta coef. carga versus desloc. horiz. no topo da plataforma original de Benjamin e Ebecken (1987)

Nesta análise o programa NLFA utilizou a técnica de controle de deslocamentos. As tolerâncias de convergência de deslocamentos e forças desequilibradas adotadas foram respectivamente, $\varepsilon_d = 0.0001$ e $\varepsilon_f = 0.01$.

O programa convergiu sem problemas até o passo incremental 210, correspondendo ao deslocamento de controle igual a 1.1 m. O número total de iterações na análise foi igual a 2731. Os coeficientes multiplicadores de carga no final do passo incremental 210 são $\lambda_3 = 8.14573$, $\lambda_1 = 1.0$ e $\lambda_2 = 1.0$.

No gráfico (VI.9) tem-se o esforço axial de compressão do elemento diagonal 116, localizado próximo à base da plataforma, em função do deslocamento horizontal no topo da plataforma. Nota-se neste gráfico a plastificação do elemento ao longo da análise.

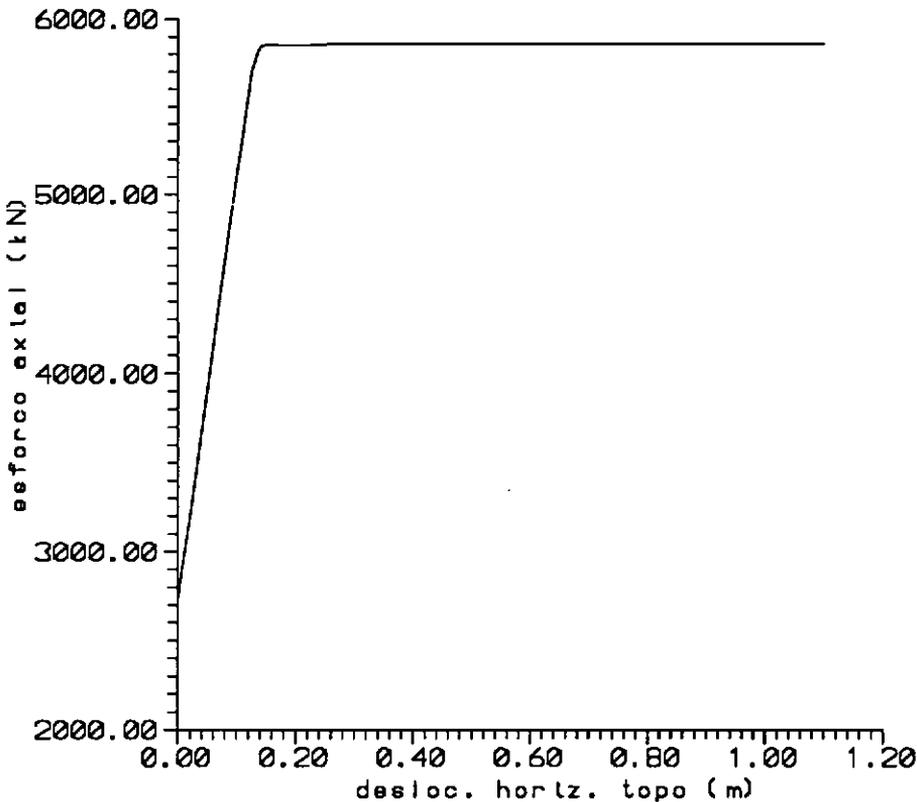


Gráfico VI.9 Esforço axial de um elemento versus desloc. horiz. no topo da plataforma original de Benjamin e Ebecken (1987)

No gráfico (VI.10) tem-se a resposta do programa NLFA para a análise elasto-plástica perfeita e grandes deslocamentos.

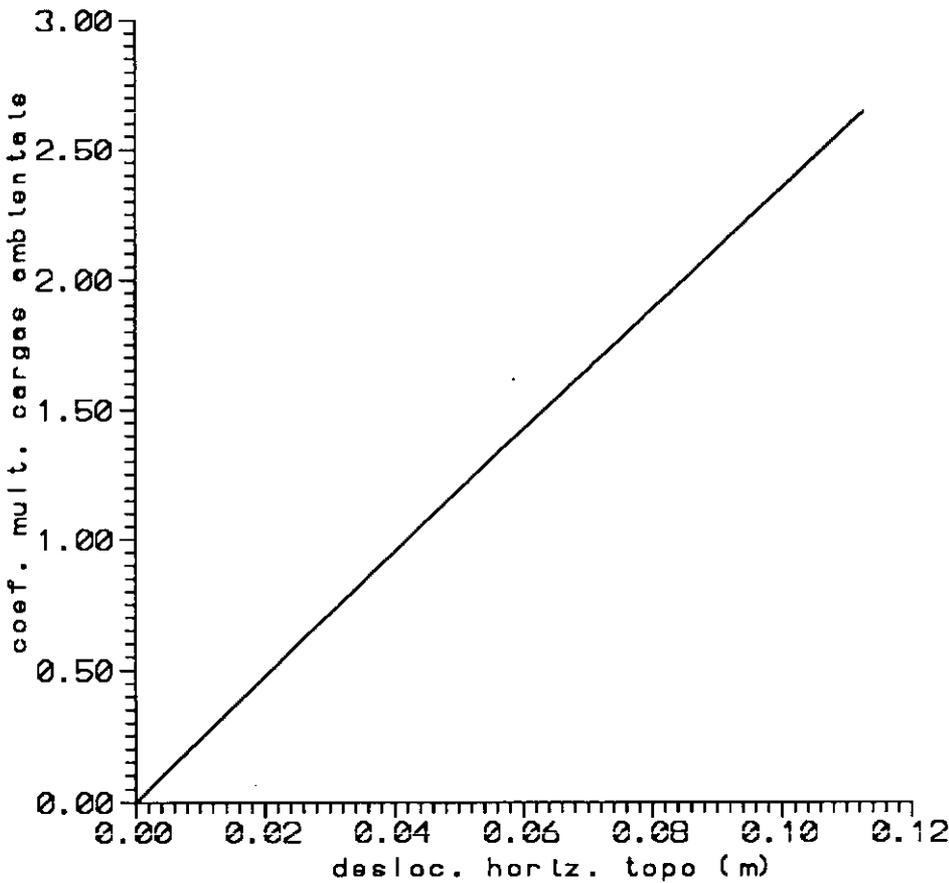


Gráfico VI.10 Resposta coef. carga versus desloc. horiz. no topo da plataforma da figura (VI.7)

No eixo das ordenadas tem-se o coeficiente multiplicador λ_3 de cargas ambientais e no eixo das abscissas o deslocamento x global no topo da plataforma.

As tolerâncias de convergência de deslocamentos e forças desequilibradas adotadas foram respectivamente, $\varepsilon_d = 0.0005$ e $\varepsilon_f = 0.05$.

O programa não convergiu no passo incremental 154, após exceder o limite de 50 iterações. O número total de iterações na análise foi igual a 691. A condição de não convergência ocorreu quando $\lambda_3 = 2.648$, $\lambda_1 = 1.0$ e $\lambda_2 = 1.0$.

Mesmo utilizando o método de controle de deslocamentos, torna-se difícil obter a resposta pós-crítica da estrutura, pois ocorrem grandes deformações nos elementos e os critérios de tolerância de convergência adotados não são atendidos.

O modelo refinado apresenta uma redistribuição maior de esforços que o modelo original de Benjamin e Ebecken (1987) e portanto a convergência do processo incremental iterativo é bem mais difícil. No modelo refinado o comportamento não linear físico e geométrico do modelo de elementos finitos fica melhor representado. Na plataforma analisada, os elementos apresentam grandes deformações e a formulação não-linear geométrica adotada neste trabalho admite grandes deslocamentos, porém as deformações devem ser pequenas. Isto prejudica a convergência do algoritmo na análise elasto-plástica de um modelo refinado.

VI.9 Plataforma fixa da bacia de Campos

A plataforma fixa do tipo jaqueta da figura (VI.8) é bastante similar às plataformas existentes na bacia de Campos no estado do Rio de Janeiro.

Esta plataforma possui cerca de 177.5 m de altura. As dimensões transversais no topo são 51 m na direção x e 38 m na direção y . Na base da plataforma as dimensões transversais são 80.6 m na direção x e 61.7 m na direção y .

O modelo da MEF encontra-se suficientemente refinado, para poder ser analisado pela formulação do método dos elementos finitos, do presente trabalho.

O modelo estrutural possui 5485 elementos de pórtico espacial e 4825 nós. O número total de equações do modelo é igual a 28791 e a matriz de rigidez global da estrutura é constituída por 515745 termos não nulos, após a reordenação nodal pelo método do grau mínimo.

As cargas atuantes na plataforma são: peso próprio, cargas de deck e cargas ambientais. A resultante total de cargas verticais atuantes no deck da plataforma é igual a 112 MN.

Os parâmetros gerais para determinação das cargas ambientais de vento e corrente são iguais aos definidos para a plataforma do item (VI.8), à exceção da mudança da altura da lâmina de água da plataforma.

Na análise não linear as cargas de peso próprio e cargas de deck são aplicadas gradualmente no início e após são mantidas constantes ao longo da análise (coeficientes λ_1 e λ_2 iguais a 1.0). Somente as cargas ambientais (coeficiente λ_3) são aumentadas gradualmente até atingir a configuração de colapso da estrutura.

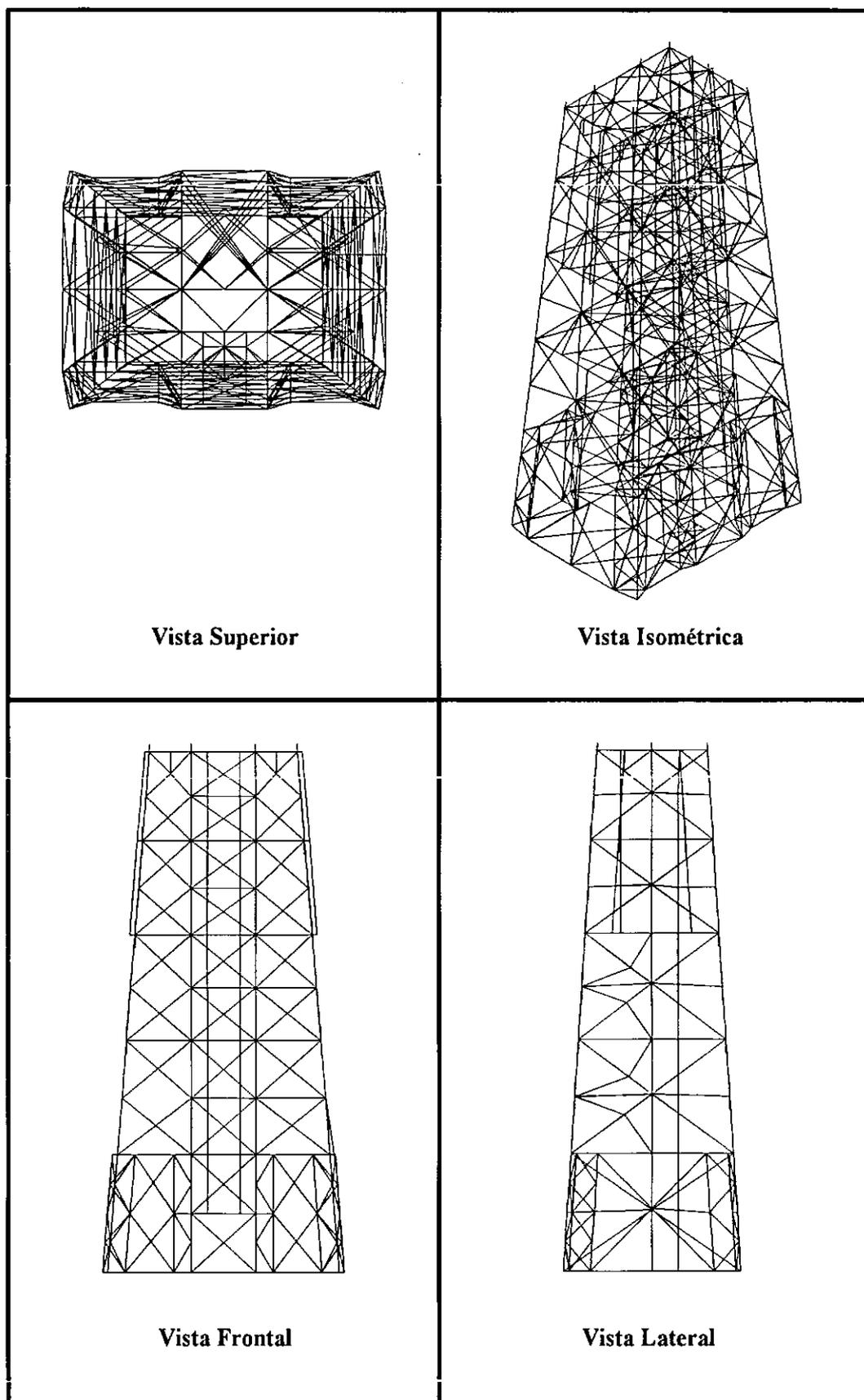


Figura VI.8 Plataforma fixa da bacia de Campos

A técnica de controle de deslocamentos é utilizada na análise não linear incremental iterativa. O deslocamento horizontal de um nó do topo da plataforma é utilizado para o controle da análise não linear pelo método de Newton-Raphson. A atualização da matriz de rigidez da estrutura é feita em cada iteração da análise.

Em função do porte do presente modelo e da limitação de memória disponível em cada nó do cubo, este modelo só pode ser executado utilizando no mínimo quatro processadores do cubo.

Na tabela (VI.8) tem-se os resultados do algoritmo de decomposição de domínio, utilizando-se quatro processadores.

Tabela VI.8 Decomposição de domínio da MEF da figura (VI.8) com 4 processadores

Subdomínio	1	2	3	4
Núm de elem.	1372	1372	1372	1369
Núm. elem. interface	174	176	103	202
Núm. de nós	1207	1201	1209	1208
Núm. de nós interface	44	64	33	63

Na tabela (VI.9) apresentam-se os resultados da decomposição de domínio, utilizando-se oito processadores na análise.

Tabela VI.9 Decomposição de domínio da MEF da figura (VI.8) com 8 processadores

Subdomínio	1	2	3	4	5	6	7	8
Núm. de elementos	686	686	686	686	686	686	686	683
Núm. elem. interface	107	97	119	127	55	95	111	152
Núm. de nós	602	604	604	598	604	606	605	602
Núm. de nós interface	38	34	51	51	22	37	43	54

A partir dos resultados apresentados nas tabelas (VI.7) e (VI.8), conclui-se que o algoritmo de decomposição de domínio utilizado produziu uma distribuição balanceada de nós e elementos em cada subdomínio. O número de nós de interface em cada subdomínio é pequeno, em relação ao número de nós existentes em cada subdomínio.

Na tabela (VI.10) tem-se os tempos de execução da fase de pré-processamento da análise, rodada no modo sequencial.

O tempo de solução da fase de decomposição de domínio foi obtido para oito subdomínios.

Tabela VI.10 Tempos de execução (em segundos) da fase de pré-processamento sequencial da MEF da figura (VI.8)

Fase	i80386	i860
Leit. e impr. geometria	193.62	45.95
Reord. nodal grau mínimo	33.40	2.06
Preench. da matriz global	1.27	0.07
Leit. e impr. de cargas	171.86	43.74
Decomp. de domínio	547.8	30.49
Alocação. e carga do cubo	3.05	----

Analisando-se os resultados da tabela (VI.10), observa-se que a fase de decomposição de domínio é a que mais consome tempo de execução no processador "host" i80386. No i860 este tempo de solução é bem menor.

Portanto, para grandes modelos, a fase de pré-processamento é mais eficiente, quando executada no modo sequencial em um nó i860 do cubo.

Na tabela (VI.11) tem-se os tempos de execução de uma iteração típica da etapa de solução da análise, utilizando quatro e oito processadores. A rodada da versão serial não pode ser executada num nó do cubo. Para este problema a solução serial foi executada na estação Sun Sparc 1+, com 12 Mbytes de memória RAM e 2 Mflops de velocidade de processamento.

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

Tabela VI.11 Tempos de execução (em segundos) de uma iteração de solução da análise da MEF da figura (VI.8)

Fase	Sun Sparc1+	4 proc. i860	8 proc. i860
Mont. matriz global	46.2	2.2	1.1
Fator. multifrontal	38.7	4.2	3.6
Controle de desloc.	15.3	2.2	2.0
Cálc. forças int. resist.	14.8	1.5	1.1

Nota-se na tabela (VI.11) que houve "speed-up" na execução com oito processadores, em relação à solução com quatro processadores.

Observa-se ainda, na tabela (VI.11), que na Sun Sparc 1+, o tempo da fase de cálculo e montagem da matriz global é maior que o tempo da fase de fatoração. Isto pode ser explicado pelo fato de que, na fase de montagem da matriz global na Sun Sparc 1+, existem atividades de paginação em disco durante a utilização da memória virtual.

Este comportamento não acontece nas rodadas do cubo, porque todo o processamento é executado em memória RAM.

Na tabela (VI.12) tem-se os tempos totais de solução para o processador supervisor e o tempo total transcorrido para a execução das análises. O tempo transcorrido foi obtido através da rotina "time" do UNIX.

Tabela VI.12 Estatística de tempos de execução (em segundos) da MEF da figura (VI.8)

Máquina	Tempo CPU proc. 1	Tempo total execução
Sun Sparc 1+	-----	172360.0
4 proc. i860	15832.0	16830.0
8 proc. i860	12337.0	13353.0

Nota-se na tabela (VI.12) que na Sun Sparc 1+, o tempo total de análise é de 47h:52m:40s e que no cubo, utilizando 8 processadores, é de 3h:42m:33s.

No gráfico (VI.11) tem-se a resposta do programa NLFA para a análise elasto-plástica perfeita, com inclusão da não linearidade geométrica.

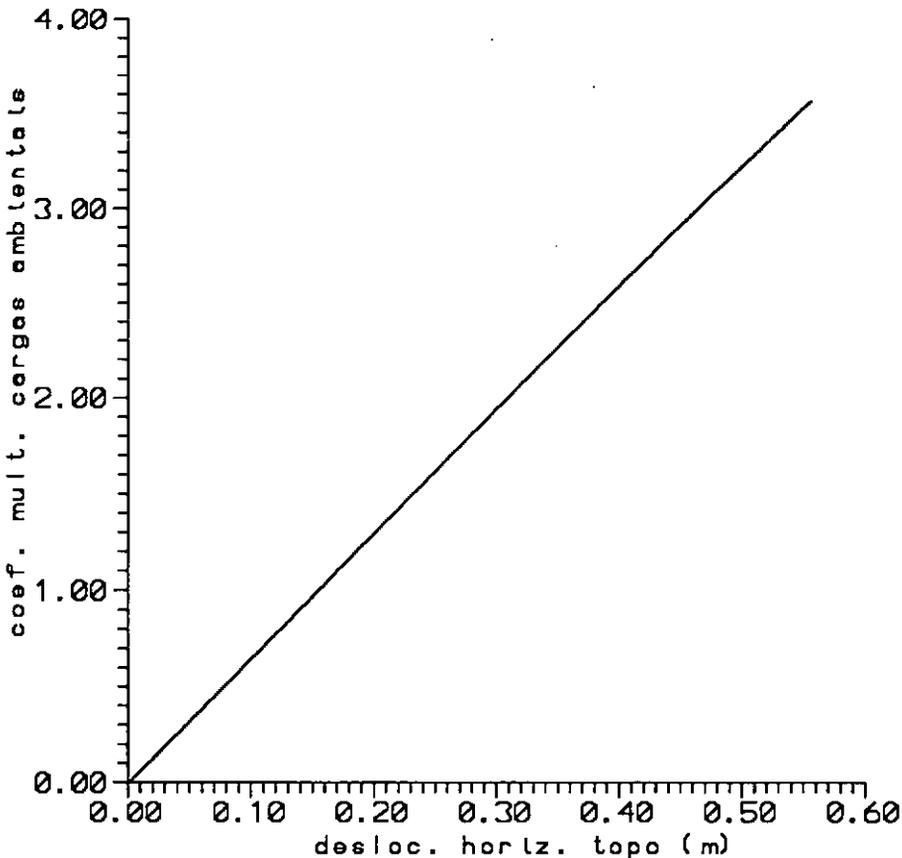


Gráfico VI.11 Resposta coef. carga versus desloc. horiz. da plataforma da figura (VI.8)

No eixo das ordenadas do gráfico (VI.11) tem-se o coeficiente multiplicador λ_3 das cargas ambientais e no eixo das abscissas o deslocamento x global no topo da plataforma.

As tolerâncias de convergência de deslocamentos e forças desequilibradas adotadas foram respectivamente, $\varepsilon_d = 0.0005$ e $\varepsilon_f = 0.05$.

O programa não convergiu no passo incremental 246 da análise, após exceder o limite de 50 iterações estabelecido para a análise.

A condição de não convergência ocorreu quando $\lambda_1 = 1.0$, $\lambda_2 = 1.0$ e $\lambda_3 = 3.569$ e o deslocamento x global no topo da plataforma foi igual a 0.5563 m.

No passo incremental de não convergência, o determinante da matriz de rigidez global da estrutura torna-se negativo, dificultando a convergência do processo incremental iterativo.

VI.10 Plataforma fixa do golfo do México

A plataforma fixa do tipo jaqueta da figura (VI.9) possui dimensões similares à plataforma de Bullwinckler, construída no golfo do México. Esta plataforma possui cerca de 429 m de altura. As dimensões transversais da plataforma no topo são 50.4 m na direção x global e 41.3 m na direção y global. Na base da plataforma as dimensões transversais são 137.1 m na direção x global e 121.9 m na direção y global.

Em função das dimensões deste modelo, o refinamento entre as juntas foi menor que o feito para os modelos anteriores.

Este modelo também precisa de uma análise dinâmica, pois as forças de inércia, são importantes num modelo deste porte. Portanto, o efeito de cargas dinâmicas deverá ser considerado nesta plataforma, em futuras análises dentro desta linha de pesquisa.

O objetivo desta análise estática elasto-plástica perfeita, com não linearidade geométrica, é o de avaliar a ordem de grandeza da carga de colapso da plataforma para as solicitações de cargas ambientais.

O efeito da interação solo-estrutura não foi considerado na análise. Os nós da base da plataforma estão apoiados na fundação.

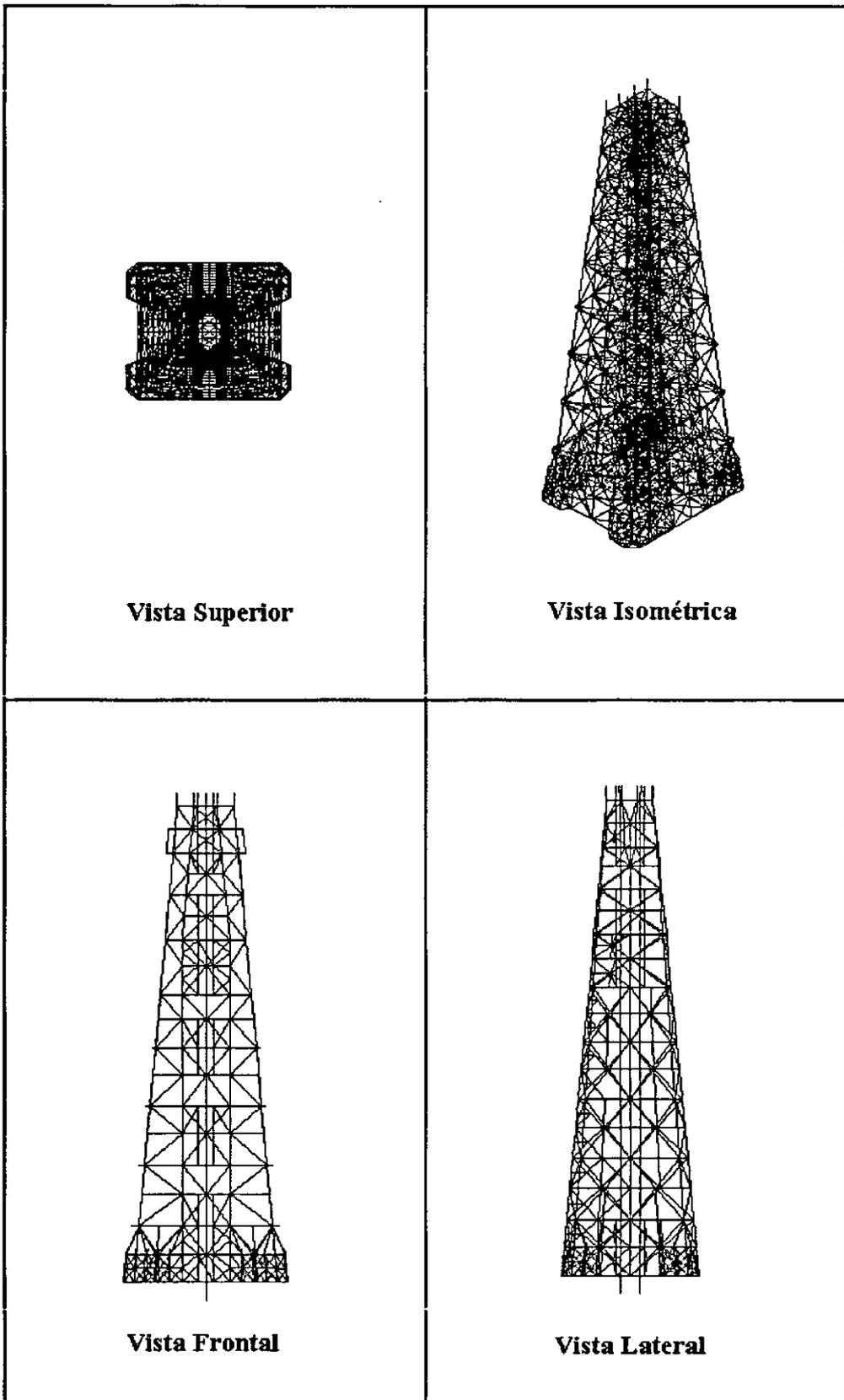


Figura VI.9 Plataforma fixa do golfo do México

O modelo estrutural possui 7034 elementos de pórtico espacial e 4685 nós. O número de equações do modelo é igual a 28022 e a matriz de rigidez global da estrutura é constituída por 806075 termos não-nulos, após a reordenação nodal pelo método do grau mínimo.

As cargas atuantes na plataforma são: peso próprio, cargas de deck e cargas ambientais.

Para os propósitos desta pesquisa acadêmica, os parâmetros gerais para determinação das cargas ambientais foram assumidos iguais aos definidos para a plataforma do item (VI.8), à exceção da altura da lâmina de água da plataforma.

Na análise não-linear as cargas de peso próprio e as cargas de deck são aplicadas gradualmente no início da análise e após são mantidas constantes (coef. λ_1 e λ_2 iguais a 1.0). Somente as cargas ambientais (coef. λ_3) são aumentadas gradualmente, até atingir a condição de não convergência da análise.

A técnica de controle de deslocamento é utilizada na análise não-linear incremental iterativa. O deslocamento de controle da análise é o deslocamento x global de um nó do topo da plataforma. A atualização da matriz de rigidez da estrutura é feita em cada iteração da análise incremental pelo método de Newton-Raphson.

Em função do tamanho do presente modelo e da limitação de memória RAM disponível em cada nó do cubo, este modelo só pode executado utilizando-se oito processadores.

Na tabela (VI.13) tem-se os resultados da decomposição de domínio para oito processadores.

Tabela VI.13 Decomposição de domínio da MEF da figura (VI.9) com 8 processadores

Subdomínio	1	2	3	4	5	6	7	8
Núm. de elem.	880	880	880	880	880	880	880	874
Núm. elem. interface	207	233	372	332	350	230	152	202
Núm. de nós	586	586	586	589	589	587	586	576
Núm de nós interface	61	87	104	76	106	58	57	50

A partir dos resultados apresentados na tabela (VI.13), observa-se que existe uma distribuição equilibrada de nós e elementos em cada subdomínio. O número de nós de interface em cada subdomínio é pequeno, em relação ao número de nós existentes em cada subdomínio.

Na tabela (VI.14) tem-se os tempos de execução da fase de pré-processamento da análise, rodada no modo sequencial.

Tabela VI.14 Tempos de execução (em segundos) da fase de pré-processamento sequencial da MEF da figura (VI.9)

Fase	i80386	i860
Leit. e imp. geometria	184.16	48.06
Reord. nodal grau mín.	53.96	2.85
Preench. da mat. global	1.81	0.08
Leit. e impr. de cargas	170.85	42.49
Decomp. de domínio	702.91	39.64
Aloc. e carga do cubo	2.90	----

Nota-se na tabela (VI.14) que a fase de decomposição de domínio é a que mais consome tempo de execução no "host" i80386.

Portanto, conforme também foi observado para os modelos das plataformas anteriores, a fase de pré-processamento roda bem mais rápido no processador i860 do que no i80386.

Na tabela (VI.15) tem-se os tempos de execução de uma iteração típica de uma etapa de solução da análise.

A rodada da versão serial do programa foi feita na Sun Sparc 1+, para comparação de resultados com o iPSC/860.

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

Tabela VI.15 Tempos de execução(em segundos) de uma iteração de solução da análise da MEF da figura (VI.9)

Fase	Sun Sparc 1+	8 proc. i860
Mont. matriz global	138.6	1.6
Fator. multifrontal	120.5	14.0
Controle de desloc.	56.4	3.3
Cálc. forças int.. resist.	17.4	1.3

Observa-se nos resultados da tabela (VI.15) que, na Sun Sparc +1, a fase de cálculo e de montagem da matriz global é a que consome mais tempo de execução, devido a atividades de paginação em disco na utilização da memória virtual.

No iPSC/860 a fase de fatoração é a que consome mais tempo de execução, como também aconteceu no modelo de plataforma analisado no item anterior.

Na tabela (VI.16) tem-se os tempos totais de solução para o processador supervisor do cubo e tempo transcorrido total para as análises.

Tabela VI.16 Estatística de tempos de execução (em segundos) da MEF da figura (VI.9)

Máquina	Tempo CPU proc. 1	Tempo total execução
Sun Sparc 1+	-----	222011.0
8 proc. i860	13589.5	14790.0

Nota-se na tabela (VI.16) que na Sun Sparc 1+, o tempo transcorrido total de solução da análise é de 61h:40m:11s e que no iPSC/860, utilizando oito processadores, este tempo cai para 4h:06m:30s.

No gráfico (VI.12) tem-se a resposta do programa NLFA para análise elasto-plástica perfeita, incluindo a não-linearidade geométrica.

No eixo das ordenadas do gráfico (VI.12) tem-se o coeficiente multiplicador λ_3 das cargas ambientais e no eixo das abscissas o deslocamento x global no topo da plataforma.

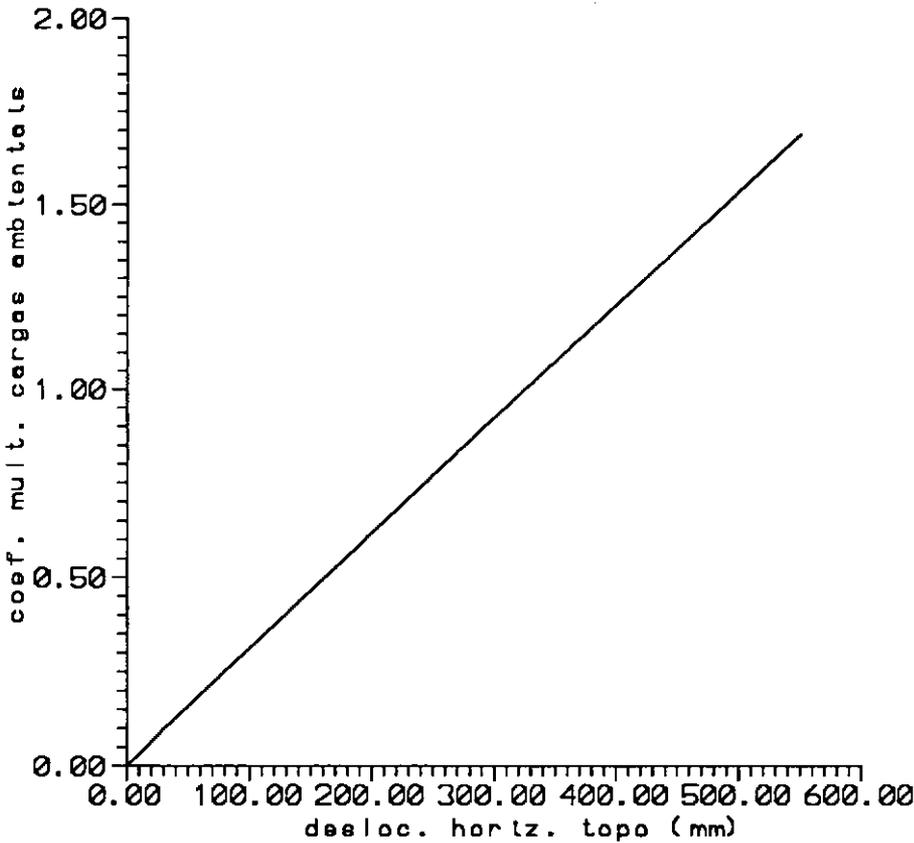


Gráfico VI.12 Resposta carga versus desloc. horiz. da plataforma da figura (VI.9)

As tolerâncias de convergência de deslocamentos e forças desequilibradas adotadas foram respectivamente, $\varepsilon_d = 0.0005$ e $\varepsilon_f = 0.05$.

O programa não convergiu no passo incremental 184, após exceder o limite de 50 iterações.

O número total de iterações realizadas na análise incremental foi igual a 950.

A condição de não convergência ocorreu quando $\lambda_1 = 1.0$, $\lambda_2 = 1.0$ e $\lambda_3 = 1.688$ e o deslocamento x global no topo da plataforma foi igual a 550.1 mm.

Mesmo utilizando-se técnica de controle de deslocamentos, a convergência do algoritmo é difícil, devido a instabilidades que ocorrem na análise, em função da redistribuição de esforços na análise elasto-plástica com grandes deslocamentos.

CAPÍTULO VII

CONCLUSÃO

VII.1 Considerações gerais

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. Destacam-se atualmente as aplicações na área de otimização e confiabilidade estrutural.

Neste trabalho foi realizada uma implementação paralela, no supercomputador iPSC/860, de um programa de elementos finitos para análise não-linear de estruturas offshore reticuladas.

VII.2 Contribuições

Esta seção apresenta uma revisão das contribuições desta tese, destacando-se:

- Utilização dos métodos diretos esparsos na solução de sistemas de equações de modelos de elementos finitos, como uma alternativa viável com relação aos métodos diretos de solução de banda variável ou perfil usualmente utilizados;
- Decomposição de domínio de malhas de elementos finitos irregulares, voltada para métodos esparsos;
- Implementação eficiente do método multifrontal esparsos no supercomputador iPSC/860;
- Análise do colapso elasto-plástico com endurecimento isotrópico e grandes deslocamentos de estruturas offshore reticuladas;
- Implementação do algoritmo de controle de deslocamento com mais de um vetor de referência de carga na análise.

A técnica de decomposição de domínio implementada permite a sub-divisão de uma malha irregular de elementos finitos entre os processadores alocados, garantindo carga computacional equilibrada e pouca comunicação entre os processadores.

O método multifrontal esparsa implementado neste trabalho é eficiente na solução de sistemas de equações de modelos de elementos finitos, tanto na implementação sequencial quanto na paralela no iPSC/860, devido à eliminação em massa de variáveis.

Na análise incremental iterativa para a obtenção de cargas de colapso, o comportamento não-linear dos membros da plataforma considera o efeito elasto-plástico nodal com endurecimento e grandes deslocamentos.

O algoritmo de controle de deslocamento apresentado no apêndice B permite a realização de análises não-lineares com a presença de mais de um vetor de carga, como acontece nas situações práticas de projeto.

A eficiência da implementação paralela, mostrada nos exemplos analisados viabiliza a utilização do supercomputador hipercubo iPSC/860 em aplicações de análise não-linear pelo método dos elementos finitos.

VII.3 Sugestões para trabalhos futuros

As seguintes sugestões para futuros trabalhos são apresentadas:

- Implementação, na formulação não-linear dos membros da plataforma, de modelos de juntas e consideração do efeito da flambagem local de cascas em seções tubulares finas (ver Ueda e Rashed, 1991). Também o efeito de "punching shear", presente nas juntas de plataformas, deverá ser analisado;
- Comparação de eficiência da solução multifrontal esparsa com outros métodos diretos de solução, como os métodos de banda variável vetorizados, e também com os métodos de solução iterativos, tais como os métodos dos gradientes conjugados e "multigrids";
- Alteração dos algoritmos apresentados para transferência de mensagens no modo assíncrono, para reduzir o custo de comunicação existente nas mensagens síncronas entre processadores;
- Estudo de novos métodos de decomposição de domínio e comparação dos algoritmos heurísticos diretos existentes com o algoritmo de otimização de análise combinatória "simulated annealing";

- Generalização da análise elasto-plástica com endurecimento do material, incluindo também o efeito do endurecimento cinemático;
- Substituição da formulação não linear geométrica do elemento de pórtico de Bathe e Bolourchi (1979), para grandes deslocamentos e pequenas deformações, por outro elemento finito que possibilite a análise com deformações moderadas ou grandes deformações. Isto deve reduzir os problemas de convergência encontrados para os modelos de plataformas analisados nesta tese.
- Utilização da presente estratégia de solução não-linear com métodos modernos de confiabilidade estrutural, para análise de plataformas offshore para águas profundas.

APÊNDICE A

REVISÃO DA TEORIA DE GRAFOS E SEU USO EM MATRIZES SIMÉTRICAS ESPARSAS.

A.1 Notações e definições básicas

George e Liu (1981) apresentam uma revisão geral da teoria de grafos para matrizes simétricas esparsas. O grafo $G = (X, E)$ consiste de um conjunto de membros designados nós (X) e de um conjunto de ligações (E), que são pares não ordenados dos nós.

Este grafo é dito indireto, pois as setas definindo o sentido de ligação entre dois nós podem ser eliminadas.

O grau de um nó é o número de ligações do grafo que incidem no mesmo.

Os nós que estão ligados entre si no grafo são ditos adjacentes.

A distância entre um par de nós é definida como sendo o número de ligações presentes no caminho mais curto conectando estes nós.

Um subgrafo $G' = (X', E')$ de G é um grafo tal que $X' \subset X$ e $E' \subset E$.

Para $Y \subset X$ o gráfico de seção $G(Y)$ é o subgrafo $(Y, E(Y))$ tal que:

$$E(Y) = \{(x, y) \in E \mid x, y \in Y\} \quad (\text{A.1})$$

Em termos matriciais, o subgrafo de seção $G(Y)$ é o grafo da matriz eliminando-se as linhas e colunas de G , exceto os nós pertencentes a Y .

Um grafo de seção é designado "clique" se os nós do subgrafo são pares adjacentes.

Em termos matriciais, o "clique" corresponde a uma submatriz cheia. Esta propriedade é importante e é utilizada na definição de variáveis supernodais do método multifrontal.

Na figura (A.2) tem-se a representação do grafo de uma malha de elementos finitos da figura (A.1). Por exemplo, o subgrafo constituído pelos nós $\{4, 5, 6 \text{ e } 7\}$ é um "clique" do grafo $G_F(A)$, que é o grafo da matriz A incluindo o preenchimento.

Na figura (A.3) tem-se a matriz de coeficientes A e a matriz de fatores de Cholesky L .

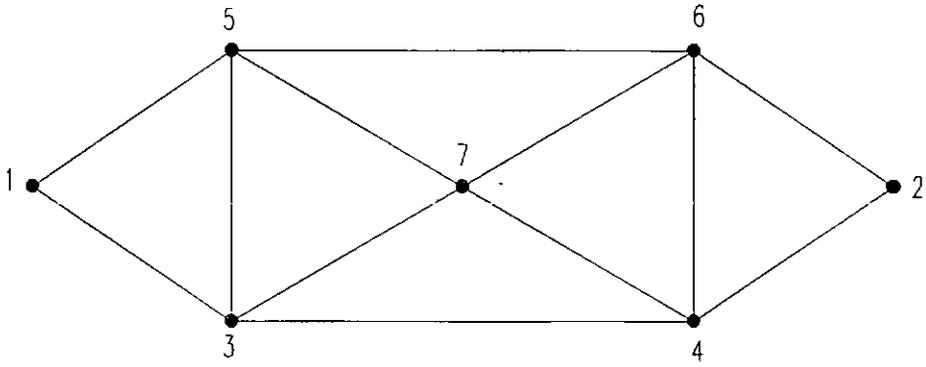
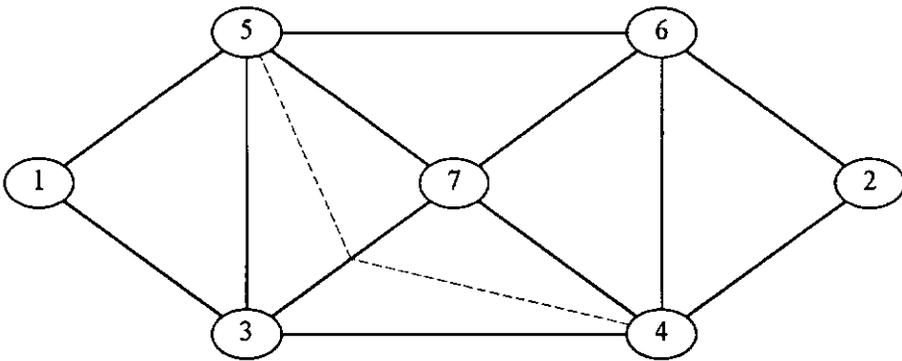


Figura A.1 Malha triangular de elementos finitos com uma equação por nó



----- ligação de preenchimento

Figura A.2 Grafo $G_F(A)$ da malha de elementos finitos da figura(A.1)

$$A = \begin{pmatrix}
 1 & \bullet & & & & & \\
 2 & & \bullet & & & & \\
 3 & \bullet & & \bullet & & & \\
 4 & & \bullet & \bullet & \bullet & & \\
 5 & \bullet & & \bullet & & \bullet & \\
 6 & & \bullet & & \bullet & \bullet & \bullet \\
 7 & & & \bullet & \bullet & \bullet & \bullet
 \end{pmatrix} \quad
 L = \begin{pmatrix}
 1 & \bullet & & & & & \\
 2 & & \bullet & & & & \\
 3 & \bullet & & \bullet & & & \\
 4 & & \bullet & \bullet & \bullet & & \\
 5 & \bullet & & \bullet & \circ & \bullet & \\
 6 & & \bullet & & \bullet & \bullet & \bullet \\
 7 & & & \bullet & \bullet & \bullet & \bullet
 \end{pmatrix}$$

○ – preenchimento
 ● – termos não-nulos

Figura A.3 Matrizes de coeficientes A e de fatores de Cholesky L do modelo de elementos finitos

A.2 Estrutura de dados para armazenar grafos

A estrutura de um grafo pode ser representada através de uma lista de adjacências. A lista de adjacências pode ser armazenada sequencialmente num vetor unidimensional adj e um vetor apontador $xadj$ de dimensão $n+1$. Para o exemplo da figura (A.1), tem-se o grafo $G(A)$ representado pela estrutura de adjacências a seguir :

$$adj() = \{\{1,3\}, \{4,6\}, \{1,4,5,7\}, \{2,3,6,7\}, \{1,3,6,7\}, \{2,4,5,7\}, \{3,4,5,6\}\} \quad (A.2)$$

$$xadj(n+1) = \{1,3,5,9,13,17,21,25,29\} \quad (A.3)$$

O número de elementos existentes no vetor adj é $|X|+2|E|+1$.

Sloan (1989) apresenta uma implementação eficiente de uma rotina em Fortran, para obtenção da estrutura de adjacências de um grafo nodal de uma malha qualquer de elementos finitos, utilizando a incidência dos elementos.

APENDICE B

DESCRIÇÃO DO HARDWARE DO iPSC/860

B.1 Nomenclatura hipercubo

Um multiprocessador hipercubo de dimensão n é uma arquitetura de $N = 2^n$ processadores idênticos. Cada processador possui memória local e desenvolve tarefas independentes dos demais processadores.

A comunicação entre processadores é feita através de transferência de mensagens.

Os nós (processadores) são numerados de 0 a $2^n - 1$. Cada nó encontra-se conectado a n nós adjacentes. Dois nós estão conectados se a representação binária de seus números (endereços) diferir de um e somente um bit.

Os canais que conectam os nós vizinhos do hipercubo são denominados de acordo com a dimensão correspondente. A dimensão de um canal entre dois nós é determinada a partir da operação binária de ou-exclusivo entre os endereços destes dois nós. A posição do bit que permanecer como um é a dimensão daquele canal.

Na figura (B.1) tem-se a representação de um hipercubo de dimensão 3.

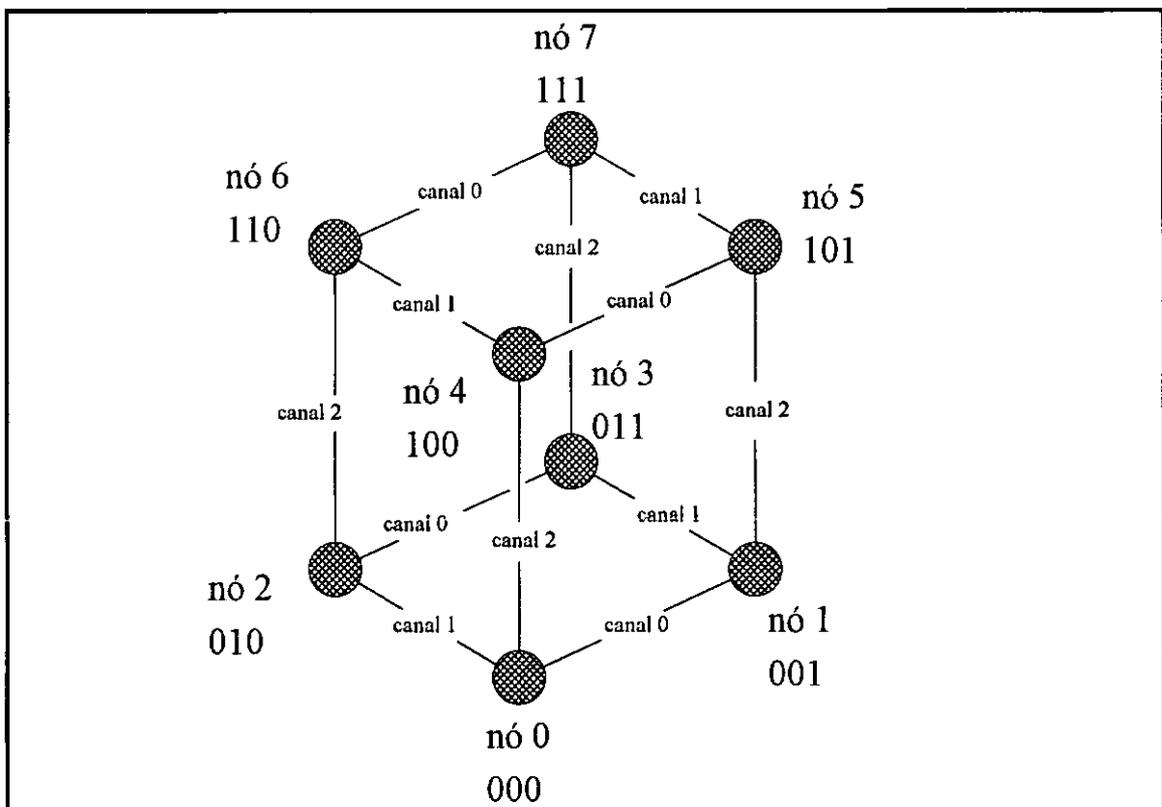


Figura B.1 - Hipercubo de dimensão 3

Para uma utilização eficiente do multiprocessador hipercubo, é preciso minimizar a penalização da comunicação e também o tempo de retenção de sincronização, para transferência de mensagens entre os processadores.

B.2 Microprocessador Intel i860

O microprocessador Intel i860 é um "chip RISC" de 64 bits com um milhão de transistores, 8 kbytes de "cache" de dados, 4 kbytes de "cache" de instruções, 32 registradores inteiros e 32 registradores de ponto flutuante. Em cada ciclo do microprocessador é possível:

- realizar uma instrução de inteiros;
- inicializar uma adição;
- inicializar uma multiplicação de 32 bits.

A uma frequência de 40 Mhz, o microprocessador atinge uma velocidade de pico de 80 Mflops em operações de 32 bits de precisão. Uma multiplicação de 64 bits pode ser inicializada a cada segundo ciclo, então o pico de velocidade do microprocessador, em operações de 64 bits, é de 60 Mflops, quando o número de adições é o dobro do número de multiplicações ou de 40 Mflops, quando o número de adições e multiplicações são iguais, como acontece em operações com matrizes.

Para obter maiores informações do processador Intel i860, ver Intel (1989).

B.3 O supercomputador iPSC/860

O supercomputador iPSC/860 é um multiprocessador hipercubo com memória distribuída e a comunicação entre os processadores é feita por meio de transferência de mensagens.

O iPSC/860 possui até 128 nós processadores. Cada nó possui até 64 Mbytes de memória, um chip i860, uma interface de rede e um hardware roteador direto de mensagens.

Cada nó está localizado em um vértice do cubo de dimensão n . Os canais de comunicação formam as arestas do cubo. Cada roteador pode manipular até 8 canais bidirecionais. Sete destes canais são vértices do hipercubo (o que leva ao limite de 128 nós) e o oitavo é utilizado para I/O externo. Os roteadores formam um circuito chaveado de comunicação na rede, que possibilita qualquer nó estabelecer a ligação com qualquer outro nó do hipercubo e transmitir diretamente a mensagem.

A interface da rede contém duas filas de 4 Kbytes. Para manter consistência de memória e de "cache" de dados do microprocessador, o i860 é responsável por todas as transferências de dados entre a memória e as filas.

Cada mensagem iniciada em um nó consome cerca de 65 microsegundos, de tempo de CPU, para estabelecer o caminho através do cubo, e então a mensagem é transferida com pico de largura de banda do hardware de 2.8 Mbytes por segundo.

Cada nó do cubo possui um hardware, com um processador i80386, para transferências de mensagens, separado do microprocessador i860. Assim, as mensagens que passam através de nós intermediários não interrompem estes processadores, e portanto virtualmente não existe penalização de comunicação entre nós não adjacentes do hipercubo.

Cada processador do hipercubo executa programas compilados em Fortran ou C, com a possibilidade de receber e enviar dados entre si através de rotinas utilitárias do sistema.

As transferências de mensagens podem ser síncronas ou assíncronas. Nas mensagens síncronas o processador fica bloqueado até que a mensagem tenha sido processada. Nas mensagens assíncronas o processador não fica bloqueado e isto é importante quando o processador tem cálculos a realizar, enquanto a mensagem está sendo enviada ou recebida. Em ambos os casos, o iPSC/860 será interrompido para transferir dados do "buffer" da fila de mensagens para a memória. O tamanho das mensagens entre dois nós do cubo pode variar de zero bytes até o limite da memória física do nó. O tamanho máximo de uma mensagem entre um nó do cubo e o "host" é de 256 kbytes.

O supervisor ("host") do iPSC/860 é uma estação de trabalho Intel 80386 com 16 Mbytes de memória, rodando o sistema operacional UNIX V.3. Este sistema é interligado a redes externas e é utilizado para gerenciar os recursos do cubo. O sistema supervisor está ligado a rede de transferência de mensagem do sistema, da mesma maneira que os nós do cubo.

O sistema operacional que roda nos nós do cubo é o NX/860, sendo utilizado para gerência dos processos e transferência de mensagens. Este sistema operacional é bastante simples, pode executar somente um processo no processador do tipo (RX) e não possui memória virtual.

Os programas feitos em Fortran ou C, para rodar no cubo, precisam utilizar compiladores para o i860. Para rodar um programa executável no cubo, o usuário precisa alocar um cubo, que consiste de todos ou parte dos nós disponíveis no sistema.

Com os nós alocados, o usuário executa o programa com um comando de carga no cubo.

Para maiores detalhes do funcionamento do iPSC/860, ver Withers (1991) e Intel Corporation (1991b).

Os nós do cubo podem acessar discos de acesso rápido e alta capacidade de armazenamento do Sistema Concorrente de Arquivos (SCA). Os discos do SCA são vistos como um único sistema virtual de disco, que pode ser acessado por vários nós simultaneamente.

A taxa de I/O para transferência de dados de um nó para o disco do SCA é de 1 Mbyte/s.

A capacidade de armazenamento do SCA varia de 760 Mbytes (1 disco) até 40 Gbytes. Para maiores informações sobre o SCA, ver Intel Corporation (1991b).

No gráfico (B.1) tem-se a eficiência do iPSC/860 e do Intel Delta na resolução de um sistema de equações, de ordem 1000×1000 , com o pacote Linpack. Os dados deste gráfico foram extraídos do trabalho de Dongarra (1993a). O tempo de execução para um processador do iPSC/860 e do Intel Delta é igual a 22 segundos.

Nota-se no gráfico (B.1) a perda de eficiência com aumento do número de processadores, devido ao alto custo de comunicação destas máquinas.

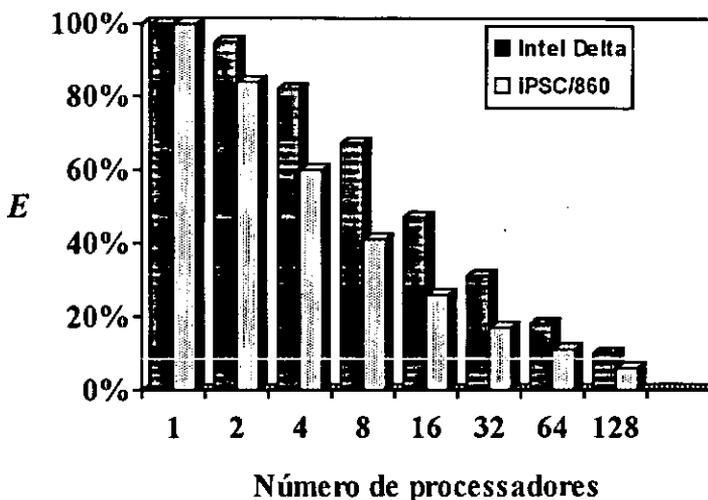


Gráfico B.1 Eficiência do iPSC/860 e do Intel Delta na solução de um problema 1000×1000 com o Linpack

No gráfico (B.2) tem-se a comparação da performance máxima obtida com a performance de pico teórica do iPSC/860, utilizando-se a biblioteca do Linpack, de acordo com Dongarra(1993b).

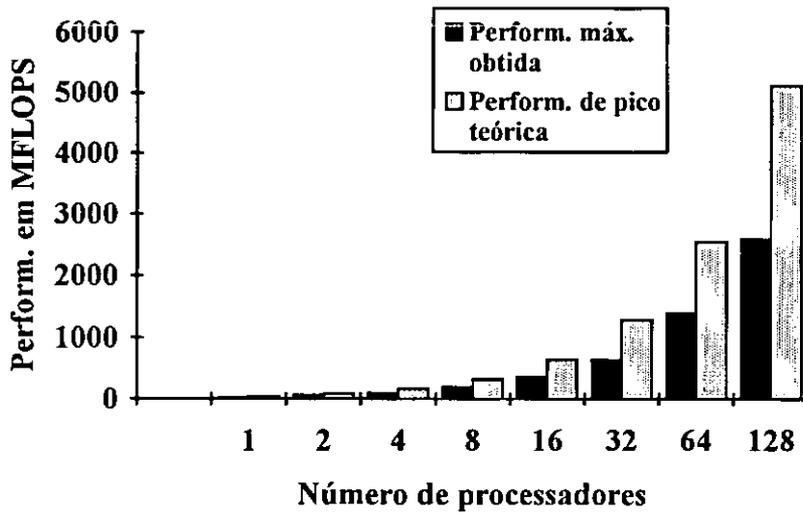


Gráfico B.2 Resultados de performance no iPSC/860, utilizando o Linpack

No gráfico (B.2), a performance teórica para um processador é igual a 40 Mflops e a máxima obtida pelo Linpack é igual a 24 Mflops.

APÊNDICE C

RECOMENDAÇÕES PRÁTICAS PARA PROGRAMAÇÃO NO iPSC/860

Em função da experiência adquirida com o uso do supercomputador iPSC/860, as seguintes recomendações práticas são sugeridas:

- Utilizar como linguagem de programação o Fortran ou o C. A linguagem Pascal não é muito utilizada em computação paralela. A nova versão do Fortran, designada Fortran 90, possui recursos de alocação dinâmica, ponteiros de memória, variáveis tipo registro, estruturas de controles do tipo "do while", "case", etc, e tem sido bastante utilizada nos programas de cálculo científico sequenciais e paralelos. Além disso, existe a portabilidade e facilidade de reutilização de códigos existentes e de bibliotecas matemáticas disponíveis para Fortran;
- As operações de ponto flutuante devem ser feitas em dupla precisão, em conformidade com a norma IEEE 754. A diretiva de compilação **-Knoieee**, do Fortran do i860, efetua as operações de ponto flutuante sem atender à norma anterior. Isto acelera o tempo de execução do processamento, à custa de perda de precisão nas operações, o que pode induzir a um acúmulo de erro durante as soluções incrementais iterativas, com muitos incrementos de solução;
- A diretiva de compilação de vetorização automática de código **-Mvect**, do Fortran do i860, deve ser testada em cada módulo do programa, pois pode causar erro de execução em alguns casos, na versão atual disponível do compilador Fortran do i860;
- A diretiva de compilação de otimização de código **-O4** deve ser usada, sempre que possível, pois utiliza os recursos de "pipeline" do processador "risc" i860;
- As funções do cubo que realizam operações globais como por exemplo, as funções `gcolx()`, `gdsum()`, `gsync()` e outras (ver Intel, 1991b), são muito mais eficientes que as funções primitivas de comunicação, como por exemplo, `csend()` e `crecv()`.
- As rotinas de alto nível Blas 1, 2 e 3, para operações de alto nível em vetores e matrizes (ver Dongarra et alii, 1979, 1985, 1990a, e 1990b), estão implantadas numa biblioteca disponível para linkedição de programas e devem ser utilizadas, sempre que possível, pois possibilitam portabilidade de código, além de utilizar os recursos de vetorização e otimização do processador i860;

- Os sistemas de transferência de mensagens assíncronas são, em geral, mais eficientes que as síncronas, principalmente quando a comunicação pode ser realizada assincronamente com a aritmética. No entanto, as dificuldades de implementação e depuração de erros são bem maiores em programas com mensagens assíncronas do que com síncronas, no iPSC/860;
- O computador "host", utilizado para acessar os processadores do cubo, possui um processador Intel 80386, o qual é muito mais lento que o processador i860 do cubo. Então, sempre que possível, o "host" deve ser utilizado somente para as funções de gerência como cubo, como por exemplo, compilações e linkedições, alocação do cubo, carga de programas no cubo, liberação do cubo, etc.
- O tempo de acesso do disco rígido do "host" é mais lento que o do disco do sistema concorrente de arquivos (SCA) do cubo. Assim, nas operações de I/O realizadas pelos processadores do cubo, o disco do SCA deve ser utilizado de preferência.

APÊNDICE D

ALGORITMO DE CONTROLE DE DESLOCAMENTOS

Os algoritmos de controle de deslocamentos propostos por Haisler, Stricklin e Key (1979) e Batoz e Dhatt (1979), possibilitam a análise de carregamentos contendo um único vetor de referência de cargas.

No algoritmo de Haisler, Stricklin e Key (1979) o sistema de equações precisa ser reorganizado, em função da componente de deslocamento prescrita escolhida na análise, que passa a ser a última equação do sistema.

No algoritmo de Batoz e Dhatt (1979) o sistema de equações não precisa ser modificado, portanto a componente prescrita de deslocamentos permanece em sua posição original.

Em termos computacionais, os dois algoritmos citados anteriormente são muito semelhantes.

Nas situações práticas de projeto, para avaliação da cargas últimas de colapso em estruturas, em geral é preciso considerar mais de um vetor de referência de carga, em termos de uma combinação linear de cargas, para formar o vetor de cargas externas da estrutura, como está apresentado na equação (D.2).

O algoritmo apresentado aqui é uma extensão do algoritmo de Batoz e Dhatt (1979), para considerar a presença de mais de um vetor de referência de carga na análise com controle de deslocamento.

A única restrição deste algoritmo proposto aqui é que somente um fator multiplicador de um dos vetores de referência de cargas fica variável na análise. Os demais fatores multiplicadores dos vetores de referências de carga devem permanecer invariantes dentro de um mesmo passo de carga, no entanto podendo variar entre passos de cargas sucessivos.

Considera-se o processo incremental iterativo de solução dado pela equação:

$$K^{m,i-1} \Delta U^{m,i} = P^m - F^{m,i-1} \quad (\text{D.1})$$

onde i é uma iteração do passo de carga m da análise.

O vetor de carga externas da estrutura na etapa m é definido pela combinação linear de n vetores de referência de cargas, de acordo com a equação (V.2):

$$P^m = \lambda_1^m P_1 + \lambda_2^m P_2 + \dots + \lambda_n^m P_n \quad (\text{D.2})$$

onde P_l é o vetor de referência de carga para a carga l aplicada na estrutura e λ_l^m é coeficiente multiplicador da carga l na etapa de incremento de carga m .

No método de controle de deslocamentos, a parcela de carga l será escolhida como tendo fator multiplicador variável no passo de carga m , o qual é obtido em função do deslocamento prescrito.

Assim, o vetor de cargas externas (D.2), na iteração i do passo de carga m , é dado por:

$$P^{m,i} = (\lambda_l^{m,i-1} + \Delta\lambda_l^{m,i})P_l + C^m \quad (\text{D.3})$$

onde C^m é a parcela do vetor de cargas externas $P^{m,i}$, que permanece invariante ao longo do processo incremental iterativo, dentro do passo de carga m .

Na equação (D.3) somente o coeficiente multiplicador de carga do vetor de referência P_l varia no processo iterativo.

Assim, a equação (D.1) pode ser reescrita na forma:

$$K^{m,i-1} \Delta U^{m,j} = P^{m,j} - F^{m,i-1} \quad (\text{D.4})$$

Tem-se as seguintes condições iniciais:

$$\lambda_l^{m,0} = \lambda_l^{m-1} \quad (\text{D.5})$$

$$F^{m,0} = F^{m-1} \quad (\text{D.6})$$

$$U^{m,0} = U^{m-1} \quad (\text{D.7})$$

No método de Newton-Raphson modificado, tem-se para a iteração i :

$$K^{m,i-1} = K^{m,0} \quad (\text{D.8})$$

Substituindo-se a equação (D.3) na equação (D.4) tem-se:

$$K^{m,i-1} \Delta U^{m,j} = (\lambda_l^{m,i-1} + \Delta\lambda_l^{m,i})P_l + C^m - F^{m,i-1} \quad (\text{D.9})$$

Na iteração $i=1$, tem-se:

$$K^{m,0} \Delta U^{m,1} = (\lambda_l^{m,0} + \Delta\lambda_l^{m,1})P_l + C^m - F^{m,0} \quad (\text{D.10})$$

Neste caso, resolvem-se os seguintes sistemas de equações:

$$K^{m,0} \Delta \bar{U}^{m,1} = \lambda_l^{m,0} P_l + C^m - F^{m,0} \quad (\text{D.11})$$

$$K^{m,0} \Delta \bar{\bar{U}}^{m,1} = P_l \quad (\text{D.12})$$

Considerando como componente de deslocamento prescrito a equação k , tem-se:

$$\Delta U_k^{m,1} = \Delta \lambda_l^{m,1} \Delta \bar{\bar{U}}_k^{m,1} + \Delta \bar{U}_k^{m,1} \quad (\text{D.13})$$

Assim,

$$\Delta \lambda_l^{m,1} = \frac{\Delta U_k^{m,1} - \Delta \bar{U}_k^{m,1}}{\Delta \bar{\bar{U}}_k^{m,1}} \quad (\text{D.14})$$

Para o caso particular em que o número de casos de carga n é igual a 1, então:

$$\Delta \bar{\bar{U}}_k^{m,1} = 0 \quad (\text{D.15})$$

Para as iterações $i=2,3,\dots$, tem-se também a solução dos seguintes sistemas de equações:

$$K^{m,i-1} \Delta \bar{U}^{m,i} = \lambda_l^{m,i-1} P_l + C^m - F^{m,i-1} \quad (\text{D.16})$$

$$K^{m,i-1} \Delta \bar{\bar{U}}^{m,i} = P_l \quad (\text{D.17})$$

No caso particular do método de Newton-Raphson modificado, a solução da equação (D.17) não precisa ser realizada, devido a condição dada pela equação (D.8):

$$\Delta \bar{\bar{U}}^{m,i} = \Delta \bar{\bar{U}}^{m,1} \quad (\text{D.18})$$

Para a componente de deslocamento prescrita na equação k , tem-se:

$$\Delta U_k^{m,i} = \Delta \bar{U}_k^{m,i} + \Delta \lambda_l^{m,i} \Delta \bar{\bar{U}}^{m,i} = 0 \quad (\text{D.19})$$

Assim,

$$\lambda_l^{m,i} = \lambda_l^{m,i-1} + \Delta \lambda_l^{m,i} \quad (\text{D.20})$$

$$\Delta U^{m,i} = \Delta \bar{U}^{m,i} + \Delta \lambda_l^{m,i} \Delta \bar{\bar{U}}^{m,i} \quad (\text{D.21})$$

$$U^{m,i} = U^{m,i-1} + \Delta U^{m,i} \quad (\text{D.22})$$

Portanto, na análise não linear com controle de deslocamentos, com mais de um vetor de referência de carga, tem-se:

- No método de Newton-Raphson tem-se, em cada iteração, a solução ("forward" e retro-substituição) de dois casos de carga;
- No método de Newton-Raphson modificado tem-se a solução de dois casos de carga na primeira iteração e, nas iterações subsequentes, de somente um caso de carga.

A análise com controle de deslocamentos é bastante eficiente com o método de Newton-Raphson modificado. Porém, o número de iterações necessárias para a convergência torna-se bastante alto nas regiões próximas aos pontos limites de colapso da estrutura.

BIBLIOGRAFIA

- Aho, A. V, Hopcroft, J. E. and Ullman, J. D. (1983), "Data structures and algorithms", Addison-Wesley Publishing Company.
- 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.
- Alves Filho, J. S. R. (1989), "The use of transputer based computers in finite element calculations", Ph.D. Thesis, Department of Civil Engineering, University College of Swansea.
- American Petroleum Institute (1986), "Recommended practice for planning, designing and constructing fixed offshore platforms", API RP2A.
- American Petroleum Institute (1989), "API recommended practice 2A-LRFD (RP 2A-LRFP)", First Edition, December.
- Amestoy, P. R. and Duff, I. S. (1989), "Vectorization of a multiprocessor multifrontal code", *The International Journal of Supercomputer Applications*, Vol. 3, Fall, pp. 41-59.
- Argyris, J. H. , Boni, B. , Hindenlang, U. and Kleiber, M. (1982), "Finite element analysis of two-and-three-dimensional elasto-plastic frames- the natural approach", *Computer Meth. in Applied Mechanics and Eng.*, 35, pp. 221-248.
- 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.
- Ashcraft, C. and Grimes, R. (1989), "The influence of the relaxed supernode partitions on the multifrontal method", *ACM Trans. Math. Software*, Vol. 15, No. 4, pp. 291-309.
- Ashcraft, C., Eisenstat, S. C. and Liu, J. W. H. (1990), "A fan-in algorithm for distributed sparse numerical factorization", *SIAM J. Sci Stat. Comput.*, Vol 11, No. 3, pp. 593-599.
- Bäcklund, J. (1976), "Large deflection analysis of elasto-plastic beams and frames", *Int. Journal Mechanical Science*, Vol. 18, pp. 269-277.
- Barton, M. L. and Scott, D. S. (1991), "Solving large finite element problems on the iPSC/860 parallel supercomputer", *Brasil Offshore's 91, VIII International Symposium on Offshore Engineering*, COPPE/UFRJ, Rio de Janeiro.
- Bathe, K. J. and Bolourchi, S. (1979), "Large displacement analysis of three-dimensional beam structures", *International Journal for Num. Meth. in Engineering*, Vol. 14, pp. 961-986.

- Bathe, K. J. (1982), "Finite elements procedures for engineering analysis", Prentice Hall, New Jersey, USA.
- Bathe, K. J. and Dvorkin, E. N. (1983), "On the automatic solution of nonlinear finite element equations", *Computer & Structures*, Vol. 17, No. 5-6, pp. 871-879.
- Batoz, J. L. and Dhatt, G. (1979), "Incremental displacement algorithms for nonlinear problems", *Int. J. Num. Meth. Engng.* 14, pp. 1262-1267.
- Benjamin, A. C. and Ebecken, N. F. F. (1987), "Nonlinear analysis of a fixed offshore platform with damaged members", *Offshore Engineering*, Vol. 6, Pentech Press, pp. 371-384.
- Benjamin, A. C. (1982), "Análise não-linear geométrica de pórticos tridimensionais pelo método dos elementos finitos", Tese de Mestrado, Programa de Engenharia Civil, COPPE/UFRJ
- Chen, W. F. and Han, D. J. (1985), "Tubular members in offshore structures", Pitman Publishing Inc.
- Chen, W. F. and Han, D. J. (1988), "Plasticity for Structural Engineers", Springer-Verlag New York Inc.
- Dongarra, J. J. (1993a), "LINPACK benchmark results: A look at parallel processing, Computer Science Department, University of Tennessee, March.
- Dongarra, J. J. (1993b), "LINPACK benchmark results: Highly parallel computing", Computer Science Department, University of Tennessee, March.
- Dongarra, J. J., Bunch, J. R., Bunch, C. B., Moler, C. B., and Stewart, G. W. (1979), "LINPACK User's Guide", SIAM Philadelphia.
- Dongarra, J. J., Croz, J. D., Hammarling, S. and Hanson, R. J. (1985), "A proposal for an extended set of fortran basic linear algebra subprograms", *ACM SIGNUM Newsletter*, Vol. 20, 1, pp. 2-18.
- Dongarra, J. J., Du Croz, J., Hammarling, S. and Duff, I. S. (1990a), "A set of level 3 basic linear algebra subprograms", *ACM transactions on Mathematical Software*, Vol. 16, No. 1, pp. 1-17.
- Dongarra, J. J., Du Croz, J., Hammarling, S. and Duff, I. S. (1990b), "Algorithm 679 - A set of level 3 basic linear algebra subprograms: model implementation and test programs", *ACM Transactions on Mathematical Software*, Vol. 16, No. 1, pp. 18-28.
- Dongarra, J. J., Duff, I. S., Sorensen, D. C. and Van der Vorst, H. A. (1991), "Solving linear systems on vector and shared memory computers", SIAM, Philadelphia.
- Duff, I. S. and Reid, J. K. (1983), "The multifrontal solution of indefinite sparse symmetric linear equations", *ACM Trans. Math. Software*, 9, pp. 302-325.

- Duff, I. S. (1986), "Parallel implementation of multifrontal schemes", *Parallel Computing*, 3, pp. 193-204, North-Holland.
- Duff, I. S., Grimes, R. G. and Lewis, John G. (1988), "User's guide for the Harwell-Boeing sparse matrix collection", Harwell and Boeing report.
- Duff, I. S., Erisman, A. M. and Reid, J. K. (1989), "Direct Methods for Sparse Matrices", Clarendon Press, Oxford.
- Duff, I. S., Reid, K. J. and Scott, A. (1989), "The use of profile reduction algorithms with a frontal code", *Int. J. Num. Meth. in Engineering*, Vol. 28, pp. 2555-2568.
- Duff, I. S. (1989), "Multiprocessing a sparse code on the Alliant FX/8", *Journal of Computational Mathematics*, 27, pp. 229-239, North-Holland.
- Duff, I. S., Grimes, R. G. and Lewis, J. G. (1989d), "Sparse matrix test problems", *ACM Trans. Math. Soft.*, 15, pp. 1-14.
- Farhat, C. (1987), "Multiprocessors in computational mechanics", Ph.D. Thesis in Engineering, University of California, Berkeley.
- Farhat, C. (1988), "A simple and efficient automatic FEM domain decomposer", *Computer & Structures*, Vol. 28, No. 5, pp. 579-602.
- Farhat, C. and Wilson, E. (1988), "A parallel active equation solver", *Computer & Structures*, 28(4), pp. 289-304.
- Farhat, C. (1989), "Redesigning the skyline solver for parallel/vector supercomputers", Report CU-CSSC-19, Department of Aerospace Engineering Sciences and Center for Spaces Structures and Controls, University of Colorado.
- 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.
- Fujikubo, M., Bai, Y. and Ueda, Y. (1991), "Dynamic elastic-plastic analysis of offshore framed structures by plastic node method considering strain-hardening effects", in *Proceedings of the First International Offshore and Polar Engineering Conference*, Edinburgh, U. K., pp. 26-33.
- George, A. and Liu, J. (1981), "Computer solution of large sparse positive definite systems", Prentice Hall, Inc. Englewood Cliffs, N.J.
- George, A., Heath, M. T., Ng, E. and Liu, J. (1987), "Symbolic Cholesky factorization on a local-memory multiprocessor", *Parallel Computing*, 5, pp. 85-95, North-Holland.
- George, A., Heath, M. T., Liu, J. and Ng, E. (1988), "Sparse Cholesky factorization on a local-memory multiprocessor", *SIAM J. Sci. Stat. Comput.*, Vol. 9, No. 2, pp. 327-340.

- Golumbic, M. C. (1980), "Algorithmic graph theory and perfect graphs", Academic Press.
- Haisler, W. E., Stricklin, J. A. and Key, J. E. (1977), "Displacement incrementation in non-linear structural analysis by the self-correction method", *International Journal for Numerical Methods in Engineering*, Vol. 2, pp. 3-10.
- Heath, M. T., Ng, E. and Peyton, B. W. (1990), "Parallel algorithms for sparse linear systems", in *Parallel Algorithms for Matrix Computations*, SIAM, pp. 83-124.
- Hilmy, S. I. and Abel, J. F. (1985), "Material and geometric nonlinear dynamic analysis of steel frames using computer graphics", *Computer & Structures*, Vol. 21, No. 4, pp. 825-840.
- 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.
- Irons, B.M. (1970), "A frontal solution program for finite element analysis", *Int. J. Num. Meth. in Engineering*, Vol. 2, No. 1, pp. 5-32.
- Jess, J. A. G. and Kees, H. G. M. (1982), "A data structure for parallel L/U decomposition", *IEEE Transactions on Computers*, Vol. C-31, No. 3, pp. 231-239.
- Justino Filho, M. R. (1991), "Solução multifrontal de sistemas de equações algébricas lineares esparsas", III Seminário de Doutorado no Programa de Engenharia Civil da COPPE/UF RJ.
- Justino Filho, M. R. and Ebecken, N. F. F. (1992a), "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. and Ebecken, N. F. F. (1992b), "Nonlinear analysis of fixed offshore platforms with damaged members on the Intel iPSC/860 hypercube", *International Conference on Computational Methods in Engineering (ICCME'92)*, Singapore, to appear.
- Kirkpatrick, S. , Gelatt Jr , C. D. and Vecchi, P. (1983), "Optimization by simulated annealing", *Science*, Vol. 220 , No. 4598, pp. 671-680.

- Lewis, J. G. and Simon, H. D. (1986), "The impact of hardware gather/scatter on Gaussian elimination", Mathematics & Modeling Technical report ETA-TR-33, Boeing Computer Services, Seattle, Washington.
- Lewis, J. G. , Peyton , B. W. and Pothen, A. (1989), "A fast algorithm for reordering sparse matrices for parallel factorization", SIAM J. Sci. Stat. Comput., Vol. 10, No. 6, pp. 1146-1173.
- Liu, J. W. H. (1985), "Modification of the minimum-degree algorithm by multiple elimination", ACM Transactions on Mathematical Software, Vol. 11, No. 2, pp. 141-153.
- Liu, J. W. H. (1986), "On the storage requirement in the out-of-core multifrontal method for sparse factorization", ACM Transactions on Mathematical Software, Vol. 12, No. 3, pp. 249-264.
- Liu, J. W. H. (1987), "A collection of routines for an implementation of the multifrontal method", Technical Report CS-87-10, Dept. of Computer Science, York University, North York, Ontario, Canada.
- Liu, J. W. H. (1989a), "The multifrontal method and paging in sparse Cholesky factorization", ACM Trans. Math. Software, Vol. 15, No. 4, pp. 310-325.
- Liu, J. W. H. (1989b), "Reordering sparse matrices for parallel elimination", Parallel Computing, 11, pp. 73-91, North Holland.
- Liu, J. W. H. (1989c), "The minimum degree ordering with constraints", SIAM J. Sci. Stat. Comput., Vol 10 , No. 6 , pp. 1136-1145.
- Liu, J. W. H. (1990), "The role of elimination trees in sparse factorization", SIAM J. Matrix Anal. Appl., Vol 11, pp. 134-172.
- Liu, J. W. H. (1991), "A generalized envelope method of sparse factorization by rows", ACM Trans. Math. Software, Vol 17, No. 1, pp. 112-129.
- Liu, J. W. H. (1992), "The multifrontal method for sparse matrix solution : theory and practice", SIAM Review, Vol. 34, No. 1 , pp. 82-109.
- Lucas, R. F. , Blank ,T. and Tiemann, J. J. (1987), "A parallel solution method for large sparse systems of equations", IEEE Transactions on Computer-Aided Design, Vol. CAD-6, No. 6.
- Lucas, R. F. (1987), "Solving planar systems of equations on distributed-memory multiprocessors", Technical Report Ph.D. Thesis, Integrated Circuits Laboratory, Stanford Electronics Laboratories, Stanford University, Stanford, California.
- 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.

- Malone, J. G. (1990), "Parallel nonlinear dynamic finite element analysis of three-dimensional shell structures", *Computer & Structures*, Vol. 35, No. 5, pp. 523-539.
- Meurant, G. (1992), "The evolution of scientific computing on parallel computers", I Escola de Computação Científica de Alto Desempenho, LNCC/CNPq, Rio de Janeiro, Março.
- Moan, T. A., Amdahl, J., Engseth, A. G. and Granli, T. (1985), "Collapse behaviour of trusswork steel platforms", 4th International Conference on Behaviour of Offshore Structures (BOSS'85), Delft, The Netherlands, pp. 255-267.
- Orbison, J. G., McGuire, W. and Abel, J. F. (1982), "Yield surface applications in nonlinear steel frame analysis", *Computer Methods in Applied Mechanics and Engineering*, 33 , pp. 557-573.
- Poole, E. L. , Knight , N. and Davis Jr., D. D. (1992), "High-performance equation solvers and their impact on finite element analysis", *International Journal for Num. Meth. in Eng.*, Vol. 33, pp. 855-868.
- Pothen, A. and Alvarado, F. L. (1992), "A fast reordering algorithm for parallel sparse triangular solution", *SIAM J. Sci. Stat. Comput.*, Vol 13, No. 2, pp. 645-653.
- Rorem, E. M. Q., Sollie, T. and Carlin, B. (1985), "Case histories of structural damages - lessons learned", 4th International Conference on Behaviour of Offshore Structures (BOSS'85), Delft, The Netherlands, pp. 1-6 .
- Scott, D. S., Castro-Leon, E. and Kushner, E. J. (1990), "Solving very large dense systems of linear equations on the iPSC/860", *The Fifth Distributed Memory Computing Conference*, IEEE Computer Society Press, Vol. 1, pp. 286-290.
- Sloan, S. W. (1989), "A fortran program for profile and wavefront reduction", *International Journal for Num. Meth. in Eng.*, Vol. 28, pp. 2651-2679.
- Smith, C. S., Somerville, W. L. and Swan, J. W. (1991), "Residual strength and stiffness of damage steel bracing member", *Offshore Technology Conference*, OTC 3981, Houston, Texas, pp. 273-282.
- Soreide, T. H., Amdahl, J., Granli, T. and Åstrud, O. C. (1986), "Collapse analysis of framed offshore structures", *offshore Tecnology Conference*, Houston, pp. 95-102.
- Stewart, G., Efthymiou, M. and Vugts, J. H. (1988), "Ultimate strength and integrity assessment of fixed offshore platforms", *BOSS*, pp. 1205-1221.
- Timoshenko, S. P. and Gere, M. (1963), "Theory of elastic stability", McGraw-Hill Book Company, New York, 2nd edition.
- Ueda, Y., Rashed , S. M. H. and Nakasho, K. (1985), "New efficient and accurate method of nonlinear analysis of offshore tubular frames (the idealized structural unit method)", *Journal of Energy Resources Technology*, *Trans. of the ASME*, Vol. 107, No. 2, pp. 204-211.

- Ueda, Y. and Rashed, S. M. H. (1991), "Modern method of ultimate strength analysis of offshore structures", *International Journal of Offshore and Polar Engineering*, Vol. 1, No. 1, pp. 27-41.
- Ueda, Y. and Fujikubo, M. (1991), "Generalization of the plastic node method", *Computer Methods in Applied Mechanics and Engineering*, Vol. 92, pp. 33-53.
- Ueda, Y. and Fujikubo, M. (1992), "Plastic node method considering strain-hardening effects", *Computer Methods in Applied Mechanics and Engineering*, Vol. 94, No. 3, pp. 317-337.
- Withers, G. (1991), "Parallel programming on the iPSC/860 systems", Intel Corporation.
- Yannakakis, M. (1981), "Computing the minimum fill-in is NP-complete", *SIAM Journal of Algorithms and Discrete Methods*, Vol. 2, No. 1, pp. 77-79.
- Zhang, W. and Liu, E. M. (1991), "Parallel frontal solver on the Alliant FX/80", *Computers & Structures*, Vol. 38, No. 2, pp. 203-215.
- Zienkiewicz, O. C. (1977), "The finite element method ", 3rd edition, McGraw Hill.