ANÁLISE TRANSIENTE DE GRANDES DEFORMAÇÕES EM COMPUTADORES DE ARQUITETURA VETORIAL/PARALELA

José Luis Drummond Alves

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE 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:

Nelson Francisco Favilla Ebecken
(presidente)

Edison Castro Prates de Lima

Luiz Landau

Alvaro Maia da Costa

Abimael Fernando Dourado Loula

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 1991

ALVES, JOSÉ LUIS DRUMMOND

Análise Transiente de Grandes Deformações em Computadores de Arquitetura Vetorial/Paralela [Rio de Janeiro] 1991

xiv, 123 p., 29.7 cm, (COPPE/UFRJ, D.Sc., ENGENHARIA CIVIL, 1991)

TESE – Universidade Federal do Rio de Janeiro, COPPE

1 – Análise Transiente 2 – Algoritmos Explícitos 3 – Análise Não-linear

4 – Processamento Vetorial/Paralelo

I. COPPE/UFRJ II. Título(Série).

"No pressure, no diamonds"

Anônimo

À Helena

e ao nosso *pequeno-grande* Fábio

Agradecimentos

Ao amigo, Nelson Ebecken, pela orientação, parceria e perseverança nesses muitos anos de estrada. A Luiz Landau e Edison C.P.Lima pela amizade, idéias e incentivo ao longo desses mesmos longos e tortuosos anos de estrada.

Ao Alvaro, Moreira, Breno, Márcia, Padilha, Gilbertão, Gilberto Luziê ... a nós, os remanescentes daquele *anônimo* Grupo Offshore, pelo coleguismo.

A Simplício de Freitas, Paulo Chaves, Paulo Osório, Ronaldo Polak, Luiz Alberto, Cesar, Paulo Rubens, Ana Lucia, Mendonça, Yuan, Célio, Salek, Luiz Paulo e Marly, a equipe amiga da PETROBRAS/DEPEX/DISEP/SEAP, assim como ao Luiz Tamanini da IBM do Brasil, pela competência, acolhida e atenção dispensadas.A Álvaro Maia da Costa, da PETROBRAS/CENPES/DIPREX/SEDEM pelo decisivo apoio e incentivo para o início deste trabalho. Ao Salvatore, ao Paulo D'Elia e todo o *time* alocado no CENPES.

Ao Osni, Fernandão, Paraná, Geraldão, Zepa, André, Reynaldinho, Célio, P.T., Martita, Regina, Léa, Lena, Tereza, Eliane, Ronaldo, Michelle, Ney, Dudu, Magluta, ... enfim, à galera que dividiu os bons momentos e suportou (?) outros não tão bons.

Aos amigos na COPPE/UFRJ, em especial, no Programa de Engenharia Civil, ao corpo docente e funcionários técnico-administrativos.

Ao Dr.Fábio e à Da.Juraci, ao Sr.Teller e Da.Bertha, ao Bigo, Maurício, Vera, Jayme, Márcio, Geraldo, Maurício & Emma, ... pela constante e carinhosa torcida.

Ao "Vô" Belisário, e ao Velho Zezé do Gato que começaram tudo isso !

Last but not least, à Helena pelo amor, paciência, cumplicidade, e pela divina dádiva chamada **FÁBIO**.

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

Análise Transiente de Grandes Deformações em Computadores de Arquitetura Vetorial/Paralela

José Luis Drummond Alves

Dezembro de 1991

Orientador: Nelson Francisco Favilla Ebecken
Programa: Engenharia Civil

Neste trabalho são apresentadas as idéias básicas empregadas no desenvolvimento de uma implementação para a análise não-linear dinâmica em regime transiente, via Método dos Elementos Finitos (MEF), em ambiente computacional dotado de facilidades para o processamento vetorial/paralelo. A integração no tempo das equações resultantes do processo de semi-discretização emprega o operador de diferenças centrais, bem como sua variante: o operador particionado explícito-explícito. A análise estática é também disponível via relaxação dinâmica e viscosa. Ênfase especial é dada à implementação da estrutura de dados para a consideração do comportamento não-linear físico e geométrico. Para a avaliação do desempenho computacional da implementação foi efetuada uma série de testes sobre uma plataforma computacional de alto desempenho, o IBM 3090 600E/6VF, dotada das características requeridas de processamento vetorial e paralelo.

Palavras-chave: análise transiente, algoritmos explícitos, análise não-linear, processamento vetorial/paralelo.

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

Transient Analysis Considering Large Deformation in Vector/Parallel Computers

José Luis Drummond Alves

December, 1991

Thesis Supervisor: Nelson Francisco Favilla Ebecken
Department: Civil Engineering

This work addresses the basic underlying ideas employed in the development of a finite element code for nonlinear transient analysis in the framework of vector/parallel processing. The time integration of the semidiscrete equations of motion are carried out by the explicit central difference operator as well as its partitioned version. Nonlinear static problems can also be addressed through simultaneous relaxation approaches of the dynamic (hyperbolic) and viscous (parabolic) type. Due attention is given to the design of the data structure, regarding both material and geometric nonlinear behavior, in order to enable vector and parallel execution. For performance evaluation purposes, a computational platform endowed with the required capabilities was used, the IBM 3090 vector/multiprocessor mainframe.

Key Words: transient analysis, explicit algorithms, nonlinear analysis, vector/parallel processing.

Índice

I Introdução 1

I.1 Sistemas de Alto Desempenho e a Engenharia Assistida por Computador 3

I.2 Alguns Problemas Típicos 5

I.3 Motivação 7

I.4 Objetivos 8

II Características dos Novos Ambientes Computacionais 10

II.1 Classificação das Modernas Arquiteturas 10

II.2 Características dos Processadores 12

II.3 Características do Sistema IBM 3090 13

III Fundamentos da Mecânica dos Sólidos 17

III.1 Generalidades 17

III.2 Descrição do Movimento 18

III.3 Descrição Cinemática - Medidas de Deformação 20

III.3.1 Os Gradientes de Deformação e de Velocidade . . . 20

III.3.2 Medidas de Deformação Mais Empregadas 22

III.3.3 Aplicabilidade do Tensor Deformação Linear . . . 23

III.4 Descrição Cinética - Medidas de Tensão 24

III.4.1 Taxas Objetivas de Tensão 25

III.5 Semi-discretização MEF 28

IV Fundamentos Teóricos **31**

IV.1 Equações Básicas Envolvidas na Análise Transiente via MEF 31

IV.2 Estabilidade Numérica 33

IV.2.1 Condição de Courant 33

IV.2.2 Estimativas para a Freqüência Máxima 35

IV.3 Solução de Problemas Estáticos 36

IV.3.1 Algoritmos de Relaxação Simultânea 36

IV.3.2 Relaxação Dinâmica 37

IV.3.3 Relaxação Viscosa 39

V Operador Particionado Explícito-Explícito **42**

V.1 Generalidades 42

V.2 Descrição do Algoritmo 43

V.3 Aspectos da Implementação 45

VI Avaliação das Forças Internas **47**

VI.1 Expressão Geral 47

VI.2 Integração Numérica 47

VI.2.1 Quadratura de Gauss "Full-Rank" 47

VI.2.2 Integração Reduzida e Controle de Modos Espúrios 49

VI.3 Amortecimento Viscoso Artificial 53

VI.3.1 Aspectos da Implementação 55

VI.4 Comportamento Não-linear Físico 56

VI.4.1 Generalidades 56

VI.4.2 Definição do Problema 56

VI.4.3 Implementação do Algoritmo Preditor Tangente com Retorno Radial 58

VII Transferência de Dados Global-Local **65**

VII.1 Montagem de Vetores Locais e Globais 65

VII.2 Técnica de Agrupamento dos Elementos Disjuntos 69

VII.3 Mapeamento Inverso 70

VIII Paralelização do Código **73**

IX Aplicações Numéricas **78**

IX.1 Simulação Geofísica - Geração de Dados Sintéticos para VSP 78

X Conclusões **110**

A Medidas de Desempenho - Lei de Amdahl **122**

Lista de Figuras

IV.1 Algoritmo de Relaxação Dinâmica 38

IV.2 Algoritmo de Relaxação Viscosa Acelerada 41

V.1 Domínio Particionado 43

V.2 Algoritmo particionado Explícito-Explícito 44

V.3 Estrutura de Dados para as Partições 46

VI.1 Preditor Tangente com Retorno Radial 59

VI.2 Processamento sob Controle de Máscara 61

VI.3 Formação de uma Lista Compactada 62

VI.4 Relação Constitutiva Não-Linear – Algoritmo Vetorizável 64

VII.1 Transferência de dados local/global – relações entre os arranjos envolvidos nas etapas de assembling e disassembling (U, F arranjos globais; UE, FE arranjos locais) 68

VII.2 "Mesh Coloring" – aplicação da técnica a uma discretização bidimensional regular 69

VII.3 Mapeamento Inverso – tabela de Conetividade das Equações para o caso bidimensional 71

VIII.1 Organização de dados empregada para vetorização e paralelização do cálculo das forças internas 75

VIII.2 Implementação do paralelismo para a avaliação das forças internas 77

IX.1 Simulação Geofísica - Definição do problema 79

IX.2 Simulação Geofísica - Função de carga do tiro 79

IX.3 Simulação Geofísica – Instantâneo das velocidades resultantes para t=100 ms 82

IX.4 Simulação Geofísica – Instantâneo das velocidades resultantes para t=200 ms 83

IX.5 Calota Esférica - definição do problema 85

IX.6 Perfil de Utilização de CPU - cada setor representa o consumo percentual de tempo de processador para cada etapa básica da solução do problema da calota esférica 87

IX.7 Calota Esférica - Deformadas para malha com 1000 elementos . . 88

IX.8 Calota Esférica - Deformadas para malha com 240 elementos . . 89

IX.9 Calota Esférica - Deslocamentos do ápice (presente trabalho) . . 90

IX.10 Calota Esférica - Deslocamentos do ápice (segundo [Nag75]) . . . 90

IX.11 Calota Esférica – Deslocamentos do ápice para análise particionada não-linear 92

IX.12 Calota Esférica – Deformadas (Análise Particionada - 160 elementos) 93

IX.13 Barra de Taylor – Definição do problema 96

IX.14 Barra de Taylor – Deformada Final (t = 80. ms) 97

IX.15 Barra de Taylor – Deformada Final (detalhe) 97

Lista de Tabelas

III.1 Tensores conjugados para deformação e tensão 25

III.2 Transformações entre medidas de tensão 25

IX.1 Tempos de CPU para Simulação Geofísica 80

IX.2 Tempos de CPU para Calota Esférica. 84

IX.3 Calota Esférica com Partição Rígida – comparação do desempenho entre as análises convencional e particionada 91

IX.4 Tempos para análise com integração "full-rank" e cinco incrementos de carga 101

IX.5 Tempos para análise com integração reduzida e quatro incrementos de carga 103

IX.6 Medição do desempenho em paralelo - Formação das baterias de testes 104

IX.7 Resultado das medições para o problema da calota esférica com 240 elementos e 2000 intervalos de tempo 105

IX.8 Resultado das medições para o problema da calota esférica com 240 elementos e 7000 intervalos de tempo 106

IX.9 Resultado das medições para o problema da calota esférica com 1000 elementos 106

IX.10 Comparaç ão entre o ganho teórico máximo e o medido na execução concorrente 108

Capítulo I

Introdução

O interesse pela obtenção numérica da resposta transiente em problemas da mecânica do contínuo é bastante antigo.

No decorrer desse tempo foi desenvolvida uma infinidade de programas, baseados tanto no Método das Diferenças Finitas (MDF) quanto no Método dos Elementos Finitos (MEF), para a análise de problemas da Mecânica do Contínuo em geral.

Dentre os mais consagrados e que apresentam recursos para a análise dinâmica em regime transiente podemos citar:

STEALTH – (Solid and Thermal hydraulics code for EPRI Adapted from Lagrange TOODY and HEMP) – baseado no Método das Diferenças Finitas, faz uso do algoritmo explícito de diferenças centrais para a análise transiente não linear. Originalmente desenvolvido para a análise de problemas associados ao projeto de instalações nucleares. Emprega uma descrição Lagrangeana para a descrição dos deslocamentos e deformações;

ADINA – (Automatic Dynamic Incremental Nonlinear Analysis) – baseado no Método dos Elementos Finitos, o ADINA é um programa de emprego geral para problemas não lineares, estáticos e dinâmicos. Para a solução dos problemas dinâmicos permite o uso de métodos diretos e modais para a integração das equações do movimento. Deve-se ressaltar a sua implementação da família de algoritmos implícitos e das técnicas empregadas para solução de grandes sistemas

de equações além dos diversos tipos de modelos constitutivos disponíveis.

NASTRAN – também de emprego geral, apresenta porém limitada capacidade para análise não-linear. Talvez seja o mais conhecido devido ao seu emprego maciço no programa espacial norte-americano, em função do qual foi desenvolvido.

NIKE 2D/3D – família de programas para análise dinâmica não-linear via métodos implícitos do tipo Newton+BFGS. Permite a modelagem de fenômenos de contacto-impacto,e grandes deformações com o emprego das taxas objetivas de Green-Naghdi e integração das equações constitutivas via retorno radial. Desenvolvido no Lawrence Livermore Laboratory, o caráter altamente vetorial de sua implementação, desenvolvida e otimizada para a arquitetura CRAY, o torna bastante atual.

DYNA 2D/3D – variante da família NIKE, porém dedicado à análise dinâmica em regime transiente via algoritmos explícitos visando principalmente as análises de colisão (*crashworthiness*). Para isso a sua implementação apresenta também o compromisso com os ambientes computacionais de alto desempenho e particularmente otimizada para o ambiente CRAY. Foi um dos primeiros programas a tirar partido dessa característica.

PAM-CRASH – especializado para a análise não-linear de cascas delgadas (teoria de Mindlin), considerando fenômenos de contacto e impacto. Foi desenvolvido com vistas à simulação numérica dos ensaios experimentais empregados na indústria automobilística, é também derivado da família DYNA/NIKE, explora o multiprocessamento vetorial.

Vários programas de uso consagrado, foram desenvolvidos quando recursos computacionais tais como facilidades para processamento vetorial/paralelo não eram oferecidos pelos ambientes de processamento de dados existentes.

À medida em que tais recursos tornaram-se mais difundidos no meio técnico-científico, esforços foram empreendidos no sentido de adaptar os códigos

existentes às novas características do *hardware* – esse processo de adaptação é conhecido pelo termo *migração vetorial e/ou paralela.*

Vários trabalhos relatam a sua experiência nesse sentido, vide por exemplo [Mel89]. Como norma geral, procura-se atacar localmente os pontos críticos das aplicações [Hen87], na tentativa de identificar o seu potencial de vetorização e/ou paralelismo, e então promover uma reorganização local da estrutura de dados de forma a adequá-la às exigências da arquitetura.

Nem sempre a migração atinge todas as etapas do processo original. Na verdade o objetivo não é este, além do mais há o compromisso com a exequibilidade, dentro de prazos satisfatórios, das modificações globais que se fariam necessárias.

I.1 Sistemas de Alto Desempenho e a Engenharia Assistida por Computador

Do ponto de vista do desenvolvimento científico e tecnológico, os desdobramentos decorrentes do emprego de sistemas computacionais de alto desempenho em aplicações numéricas intensivas são ilimitados.

No transcurso das duas últimas décadas a evolução dos sistemas computacionais permitiu que, alguns problemas outrora intratáveis, pudessem ser abordados, com dispêndio de recursos mais ameno, e da sua análise inferir dados importantes para a segurança e bem-estar da sociedade como um todo.

O lançamento do CRAY-1, em 1976, efetivamente o primeiro computador vetorial disponível no mercado [Sch87], revolucionou o meio técnico e científico dedicado às aplicações numéricas intensivas. A disponibilidade dos sistemas de alto desempenho propiciou a revisão e novos desenvolvimentos no sentido de prover métodos de análise mais eficientes e poderosos.

No âmbito da engenharia, uma das conseqüências interessantes da disponibilidade de sistemas computacionais (*software & hardware*) de alto desem-

penho é a da rápida evolução que se pode observar, em termos de concepções inovadoras e mais ousadas, respaldadas por ferramentas de análise mais realistas.

O emprego maciço de poderosas ferramentas de análise em sintonia com as características dos modernos ambientes computacionais favorecem uma abordagem mais consistente e detalhada dos problemas. Essas análises devem evoluir também em compasso com o grau de sofisticação e ousadia das concepções estruturais atualmente empregadas.

Na tradicional área da pesquisa experimental em laboratório, simulações numéricas têm sido cada vez mais empregadas em substituição aos ensaios experimentais devido a vários fatores. O seu baixo custo, a exiguidade dos prazos em um projeto e as situações em que de fato se verifique a inviabilidade da realização de testes experimentais são alguns dos motivos de maior destaque. Um outro aspecto relevante é a oferta de um espectro mais amplo de aplicabilidade.

A tendência que se verifica é no sentido de se empregar os testes para a exploração dos aspectos básicos do comportamento dos sistemas estruturais e validação dos programas computacionais. Visa-se sobretudo a utilização em larga escala da simulação numérica no âmbito do processo de desenvolvimento do projeto de Engenharia.

No entanto, para tornar as simulações numéricas ferramentas úteis de fato nesse processo, as análises envolvidas devem ser exequíveis no espaço de alguns minutos de CPU e nesse ponto, o desempenho do sistema computacional alcança importância primordial.

Com relação à modelagem mais realista dos problemas cabe salientar que a análise não-linear está apenas emergindo no cenário da Engenharia Assistida por Computador (CAE). O aumento da capacidade e velocidade de processamento nos computadores modernos, assim como a redução dos custos envolvidos, permitem a utilização da análise não-linear em larga escala, podendo-se reduzir a quantidade de testes com protótipos e ainda, eventualmente, permitir a simulação de algumas fases do processo de fabricação. Isso constitui um importante dado para a agilização e integração das atividades de projeto e linha de produção sob

um ambiente computacional (CAD-CAM).

Nesse caso específico, Jarzab et al. [Jar87] ponderam que as simulações numéricas não substituem os ensaios em laboratório. No entanto, oferecem a possibilidade de uma análise mais detalhada, em relação às observações experimentais, dos fenômenos envolvidos, durante o impacto de veículos por exemplo. Podem ser descritas as histórias de todas as variáveis físicas relevantes, podendo também serem analisados efeitos tais como absorção de energia que, ao menos por enquanto, não podem ser medidos via análise experimental. Outra vantagem é a possibilidade de uma rápida avaliação do comportamento estrutural face às eventuais modificações introduzidas em um projeto.

I.2 Alguns Problemas Típicos

Notadamente um dos ramos de atividade mais intensa no que se refere aos desenvolvimentos para análise em regime transiente, a indústria automotiva tem demonstrado grande interesse pelas simulações numéricas para a análise do comportamento e da segurança de veículos em colisões - *crashworthiness*. Também nessa linha situa-se a indústria aereoespacial.

Na engenharia offshore podem ser identificadas diversas situações nas quais a análise transiente é especialmente indicada. A análise de segurança do heliponto ao impacto de aeronaves, a análise dos efeitos de explosões acidentais sobre os equipamentos e instalações, a queda de equipamentos e ferramentas sobre os elementos estruturais e suas conseqüências são alguns dos casos bastante corriqueiros. Fenômenos de impacto são também importantes na avaliação e dos danos provocados pelo abalroamento de embarcações de serviço sobre os componentes estruturais de plataformas marítimas. Dependendo da situação e de sua extensão, os danos podem ser tão graves que obriguem à interrupção da operação e evacuação da plataforma. Por exemplo, a colisão e danificação de uma coluna de estabilização em uma plataforma semi-submersível, com o comprometimento de sua flutuabilidade. De outra forma, uma eventual indentação localizada pode favorecer o colapso estrutural.

A análise de cravabilidade de estacas é basicamente um problema de impacto, e portanto convenientemente tratado pela análise transiente. A consideração do comportamento do solo adjacente, bem como da interface solo-estaca envolvem forte não-linearidade. Modelos simplistas, para a modelagem do problema em situações convencionais, são há muito conhecidos pelos projetistas. Entretanto, para concepções inovadoras, somente uma análise rica em detalhes, capaz de modelar realisticamente os fenômenos envolvidos pode de fato ser conclusiva [Cos88], [Cou88], [Ebe88]. O nível exigido de sofisticação do modelo torna a sua análise, em um ambiente computacional tradicional, proibitiva como ferramenta para um estudo paramétrico.

Uma área recentemente explorada com o emprego do MEF na análise transiente é a da prospecção geofísica [Ser89], especificamente na geração de dados sintéticos. A sua utilidade está na possibilidade de auxiliar as equipes de campo, encarregadas do levantamento real, no posicionamento dos equipamentos, bem como de fornecer dados para a aferição e validação de programas pós-processadores das informações obtidas no campo.

Alguns casos podem não configurar um problema dinâmico propriamente dito. No entanto o sua análise com o emprego de um método explícito de solução facilita em muito o tratamento das diversas fontes de não-linearidade. Apesar do maior número de operações requeridas, a robustez de métodos como a relaxação dinâmica aliada à sua simplicidade formal, podem torná-los bastante competitivos em ambientes de alto desempenho. Problemas de conformação mecânica (*metal forming*) e de indentação rígida (*rigid punch*) são cada vez mais estudados com o auxílio dos métodos numéricos simulando o processo de fabricação, em estamparias e afins, através de um enfoque pseudo-dinâmico.

Nos fenômenos de impacto a altas velocidades o desenvolvimento de grandes deformações dá-se em uma escala de tempo ínfima. Dadas as altas taxas de deformação verificadas, o comportamento do material pode em alguns casos afastar-se das hipóteses tradicionalmente empregadas na abordagem da mecânica dos sólidos. Na indústria bélica, por exemplo, um modelo muito empregado no projeto de determinados tipos de ogivas anti-blindagem baseia-se em um mecanis-

mo de *jato fluido* [Joh72]. Tal mecanismo deve-se à geometria especial da ogiva de material dútil e à carga explosiva que promovem o desenvolvimento de pressões elevadíssimas em comparação à tensão de escoamento. Dessa forma propõe-se um modelo simplificador, basicamente hidrodinâmico, ao se considerar o metal fluindo inviscidamente.

Sistemas de controle em tempo real são aplicações onde o tempo de resposta é fundamental. A demanda por uma alta capacidade de processamento, sobre um volume relativamente grande de dados de entrada simultaneamente fornecidos ao sistema, torna esse tipo de aplicação unicamente viável à luz das novas concepções de *hardware & software*. Como exemplos podem ser citados o controle de processos industriais, o controle de engenhos aeroespaciais automáticos (satélites de rastreamento, sensoriamento remoto, etc).

I.3 Motivação

A motivação para o presente trabalho foi a de se estudar estratégias de implementação adequadas ao ambiente computacional disponível, tirando máximo proveito de suas características de alto desempenho.

A idéia original era efetuar uma avaliação em termos do impacto sobre o desempenho computacional de uma implementação projetada para esse ambiente, desde a sua concepção, com uma outra implementação simplesmente migrada para o ambiente.

Partindo-se dessa noção, projetou-se a implementação de uma aplicação destinada à análise transiente em Mecânica do Contínuo, via Método dos Elementos Finitos (MEF), tirando máximo partido das características de alto desempenho do computador IBM 3090 600E/6VF.

Algumas das facilidades oferecidas pelos programas mais modernos foram consideradas de forma a proporcionar uma oportunidade para uma análise mais próxima das dificuldades que se apresentam a esse tipo de implementação.

Eventualmente, uma ou outra limitação foi deliberada, sem que implicasse em perda de generalidade para a abordagem do problema da implementaç ao vetorial/paralela. Uma dessas limitações foi a consideração única e exclusiva de problemas com geometria e carregamento axissimétricos, empregando elementos quadriláteros iso-paramétricos bi-lineares.

Foram considerados o comportamento não-linear físico (modelo elasto-plástico de Von Mises) e não-linear geométrico para grandes deslocamentos e pequenas deformações (Formulação Lagrangeana Total - TLF), grandes deformações (Formulação Lagrangeana Atualizada com Taxa de Jaumann - ULJ), além do caso linear (pequenos deslocamentos e pequenas deformações).

Os tipos de análises consideradas são:

- Análise transiente, empregando o operador explícito clássico;
- Análise transiente particionada, empregando o operador explícito-explícito;
- Análise estática, empregando o algoritmo de relaxação dinâmica ou ainda, o algoritmo de relaxação viscosa acelerada.

I.4 Objetivos

Como objetivos do presente trabalho podemos citar:

- o estudo de algumas das técnicas e formulações representativas do atual estado-da-arte para a análise estrutural em regime transiente;
- incorporar na medida do possível tais técnicas ao desenvolvimento de um programa original e estabelecer estratégias de implementação adequadas à sua vetorização e paralelização;
- validar a implementação através de uma série de análises para problemas extensivamente abordados na bibliografia, e

- observar o desempenho das estratégias implementadas em um ambiente computacional de alto desempenho, dotado de características que permitam o multiprocessamento vetorial.

Capítulo II

Características dos Novos Ambientes Computacionais

II.1 Classificação das Modernas Arquiteturas

Os novos sistemas computacionais oferecem uma gama ampla de opções em termos de arquitetura.

Sistemas dotados de múltiplos processadores podem ser genericamente descritos através de três elementos essenciais: granularidade, organização da memória, topologia e controle [Noo89]:

- GRANULARIDADE – refere-se ao número de processadores e envolve o potencial de cada processador individualmente. Sistemas ditos de baixa granularidade (*coarse grain*) apresentam um número relativamente baixo de processadores de alto poder computacional, normalmente empregando processadores vetoriais. É o caso dos CRAY's (4 a 8 processadores), dos IBM's 3090 (até 6 processadores) e ALLIANT's (até 8 processadores). Por outro lado, os sistemas de alta granularidade apresentam um número elevado de processadores relativamente simples. A CONNECTION MACHINE (64K processadores de 1 bit) é dito um sistema paralelo maciço. Os hipercubos INTEL iPSC 860 (até 128 nós) e o NCUBE (até 1024 nós) são sistemas de média granularidade. A granularidade afeta diretamente a estratégia de implementação das aplicações.

- ORGANIZAÇÃO DE MEMÓRIA – é uma das características mais importantes dos sistemas dotados de múltiplos processadores:

  - Memória Compartilhada ou global: compõe-se de uma memória central, monolítica, acessível a todos os processadores. As arquiteturas CRAY e IBM 3090 empregam tal organização;
  - Memória Distribuída ou Local: cada processador possui sua memória de acesso exclusivo.

- TOPOLOGIA – este ítem refere-se ao padrão adotado para a interconexão dos processadores, refletindo também a maneira pela qual os dados vão fluir no sistema. Dentre as topologias atualmente disponíveis, as mais empregadas são:

  - Barramento ou Anel: os vários processadores, as memórias e os dispositivos de entrada e saída são dispostos ao longo de um barramento de comunicação ou de conjuntos de barramentos interligados;
  - Hipercúbica: cada processador está ligado a um número n de processadores vizinhos, onde n é dito a dimensão do cubo. O número total de processadores é dado por 2^n. A organização de memória mais freqüente é a local. É o caso do iPSC, NCUBE e da Connection Machine

- CONTROLE – essa classificação também conhecida, pelo termo taxonomia de Flynn, designa o modo de operação dos processadores, mais especificamente, como são distribuídas e sincronizadas as tarefas. De especial relevância no reino dos supercomputadores são as siglas SIMD (Single Instruction Multiple Data) e MIMD (Multiple Instruction Multiple Data). Segundo essa taxonomia um IBM 3090 600 (6 processadores) é um sistema MIMD de memória compartilhada e um iPSC de dimensão 8 é um MIMD de memória distribuída. Ambos podem executar simultaneamente múltiplas instruções sobre múltiplos dados. Já uma CONNECTION MACHINE é um sistema SIMD de memória distribuída, onde uma única instrução é executada, operando simultâneamente sobre múltiplos dados. Processadores vetoriais podem ser encarados como unidades de processamento do tipo SIMD.

Na prática, sistemas de memória distribuída são mais difíceis de se programar do que os de memória compartilhada. Sistemas de memória compartilhada são geralmente de baixa granularidade pois é muito provável a saturação dos mecanismos de acesso à memória. Tal fato se reflete no problema da *contenção* de memória e *latência* do barramento.

II.2 Características dos Processadores

O superior desempenho dos modernos computadores em relação aos seus antecessores deve-se em grande parte à tecnologia empregada em seus processadores [Don89].

Internamente, são exploradas as possibilidades de paralelismo ao nível de microcódigo, com o eficiente emprego de múltiplas unidades funcionais de forma coordenada. Uma unidade funcional seria encarregada de operações de adição, uma outra especializada para multiplicações e assim por diante.

Em um nível ainda mais baixo, vislumbrou-se a possibilidade da segmentação de cada uma das unidades funcionais de forma que cada segmento fosse responsável pela decodificação , interpretação e execução de uma operação. Esse último enfoque dá lugar ao conceito de *pipelining* que assemelha-se a uma linha de montagem, onde cada tarefa é subdividida em micro-tarefas executadas por uma parte do *hardware*, de forma concorrente.

Uma forma de se alcançar desempenhos ainda mais elevados dá-se com o emprego de instruções vetoriais. Nesse caso, as instruções são decodificadas uma única vez, deixando a unidade funcional pronta para operar sobre dados continuamente alimentados no seu *pipeline*. As instruções vetoriais definem as operações a serem executadas e o endereço dos elementos iniciais dos operandos.

As instruções podem valer-se ainda da concatenação de unidades funcionais distintas (*chaining*). Por exemplo as unidades de multiplicação e adição podem ter seus *pipelines* encadeados de forma a acomodar uma instrução composta (*compund instruction* capaz de atingir um desempenho próximo dos limites

teóricos do processador. Este é o caso, na maioria dos processadores vetoriais, do tratamento dado às operações do tipo SAXPY, segundo a terminologia BLAS [Don89].

Ainda no caso de processadores vetoriais emprega-se uma classificação segundo o tipo de comunicação processador/memória empregado. A comunicação entre as unidades funcionais e a memória pode ser dos tipos memória-memória ou registrador-registrador. No tipo memória-memória os operandos, resultados parciais e finais são armazenados diretamente na memória do sistema. No esquema registrador-registrador é necessário que os operandos sejam carregados da memória para registradores escalares ou vetoriais para que possam então serem utilizados pelas unidades funcionais. A grande maioria das máquinas vetoriais emprega este último esquema.

II.3 Características do Sistema IBM 3090

Para o desenvolvimento desse trabalho foi utilizado principalmente o sistema IBM 3090 600E/6VF instalado no centro de processamento de dados da PETROBRAS-EXPROPER/SEDE.

Detalhes de sua arquitetura e princípios operacionais podem ser encontrados em [IBM86a], [IBM86b] e [IBM87].

A arquitura do IBM 3090 600E/6VF é a de um sistema de memória compartilhada acessada por 6 CPU's fortemente acopladas, portanto um sistema de baixa granularidade, com capacidade de processamento vetorial.

Concebido como uma extensão do System/370, a característica marcante da arquitetura é a organização da memória segundo três níveis hierárquicos: cache, memória central e memória expandida. O endereçamento segundo uma estratégia de memória virtual proporciona a transparência desses níveis às aplicações. Apesar dessa *transparência*, essa hieraquia não pode ser negligenciada quando se busca desempenho.

No modelo 600E/6VF cada CPU incorpora uma memória cache associativa de 64 kbytes e um processador vetorial (*Vector Facility - VF*) do tipo registrador-registrador. O ciclo de máquina é de 17.2 ns e cada VF apresenta uma performance de pico da ordem de 116 Mflops [Web90]. A configuração instalada no DEPEX dispõe de memória central com 256 Mbytes e memória expandida com 512 Mbytes.

Os aplicativos são processados sob o controle do sistema operacional MVS/ESA, que proporciona um espaço enderaçável (*address space*) de até 2 Gbytes

O compilador IBM VS-FORTRAN oferece o suporte à vetorização de loops, com seccionamento automático dos vetores (*vector sectioning*), de forma a acomodar os operandos nos registradores vetoriais de acordo com as características da CPU. No modelo E, o *vector section size* é de 128 elementos (256, para o modelo S). A CPU vetorial dispõe de 16 registradores vetoriais, cada um acomodando 128 operandos de 32 bits, os quais podem ser reunidos aos pares formando 8 registradores vetoriais. Assim sendo, cada um destes pares é capaz de efetuar operações sobre 128 operandos de 64 bits.

O conjunto de instruções vetoriais consiste de 171 instruções, algumas expecíficas para operações compostas (*compound instructions*), favorecendo o desempenho através de operações com encadeamento de registradores (*register chaining*). O tempo necessário à execução de uma operação vetorial consiste basicamente na penalização (*overhead*) associado à inicialização e preenchimento do *pipeline*, mais um ciclo de máquina por resultado [Tuc86]. Cada instrução vetorial possui o seu tempo caracterśtico para inicialização do *pipeline*. O melhor desempenho é então obtido para operações sobre vetores bastante longos, de forma a minimizar este *overhead* de inicialização.

A estrutura de dados deve apresentar uma organização adequada à obtenção dos melhores benefícios decorrentes da vetorização. Como regra geral deve-se procurar uma organização dos dados em vetores tão longos quanto o possível na memória. Tal fato evidencia um claro antagonismo entre a velocidade

de processamento e a busca de uma economia na alocação de memória. Dependendo do volume de informações e da configuração disponível esse enfoque pode ser inviabilizado, tornando-se necessário o recurso a dispositivos de entrada e saída mais lentos (arquivos em discos, etc.).

A seqüência de acesso aos elementos do vetor é também um dado de grande impacto no desempenho da aplicação, considerando a lógica da arquitetura do 3090. Operações sobre elementos contíguos e subseqüentes na memória (*stride* unitário) são as mais eficientes pois proporcionam o melhor aproveitamento da cache [Sol86] [Shi89]. Dentre as operações aritméticas, a divisão é a menos eficiente. Operações vetoriais sob máscara e envolvendo enderaçamento indireto são também pouco eficientes.

A CPU do 3090 oferece um conjunto amplo de instruções, incluindo algumas específicas para o tratamento de endereçamento indireto. Por sua vez, o compilador IBM VS-FORTRAN [IBM90] é capaz de analisar e gerar código vetorial em loops envolvendo endereçamento indireto, desde que não ocorram dependências recursivas, as quais constituem o mais freqüente inibidor à vetorização nesse caso.

Para a implementação dos algoritmos utilizou-se o compilador vetorial IBM VS-FORTRAN versão 2.4. Uma série de funções intrínsecas vetorizadas é disponível em sua biblioteca. Para a paralelização foram empregadas as primitivas do MULTITASKING FACILITY (MTF) também disponíveis neste compilador. Com relação à implementação paralela, é oportuno ressaltar que a utilização do compilador IBM PARALLEL FORTRAN (PF) [IBM88] foi descartada uma vez que a sua versão disponível é baseada na versão 2.1 do VS-FORTRAN, a qual não prevê a vetorização de algumas construções presentes na codificação originalmente adotada para a implementação. Problemas semelhantes foram observados em compilações anteriores com o VS-FORTRAN versão 2.3.

Recentemente liberada, a versão 2.5 do VS-FORTRAN incorpora, como extensão da linguagem padrão ANSI-77, as potencialidades do PARALLEL FORTRAN, além do suporte às primitivas MTF. Entretanto, as potencialidades adicionais do IBM-PF presentes no VS-FORTRAN 2.5 não foram exploradas neste

trabalho.

Capítulo III

Fundamentos da Mecânica dos Sólidos

III.1 Generalidades

Neste capítulo abordaremos, resumidamente, os métodos e formulações usualmente empregadas para a análise não-linear via MEF de problemas da mecânica do contínuo.

As equações que descrevem o comportamento de um meio contínuo podem ser divididas segundo quatro grandes categorias [Ode72]:

a. cinemáticas;

b. cinéticas,

c. termodinâmicas e

d. constitutivas.

Relações cinemáticas dizem respeito ao estudo do movimento sem a preocupação com suas causas. A sua principal preocupação é a definição de medidas de deformação.

As relações cinéticas, ou dinâmicas, envolvem princípios fundamentais da mecânica (conservação da massa, conservação da quantidade de movimento) relacionando causa (forças) e efeito (movimento). Definem ainda uma

medida para a força de interação entre as partículas do contínuo, a medida de tensão.

As relações termodinâmicas, em última análise, ampliam os conceitos anteriores, unificando as variáveis cinemáticas e cinéticas a outras quantidades (quantidade de calor, temperatura, entropia, etc.) quem, juntas, caracterizam o estado termodinâmico de um sistema. Valem-se para isso do princípio da conservação da energia. No estudo dos fenômenos termo-mecânicos o interesse recai sobre as parcelas de energia mecânica (cinética+potencial) e sua conversibilidade em calor, e vice-versa.

As equações constitutivas estabelecem as relações entre as variáveis cinemáticas, cinéticas e termodinâmicas, caracterizando o material que compõe o contínuo.

A consideração de grandes deslocamentos e grandes deformações introduz uma grande variedade de opções para a descrição do movimento, considerando as referidas categorias das equações envolvidas.

Para maior comodidade na formulação, as descrições cinemática e cinética utilizadas devem ser energeticamente conjugadas, isto é, o produto interno da medida de tensão pela medida de deformação deve fornecer uma medida do trabalho realizado por unidade de volume.

III.2 Descrição do Movimento

O primeiro passo necessário à descrição do movimento e dedução das suas equações básicas é o da escolha de um sistema de referência.

Na mecânica do contínuo, são empregados as seguintes descrições de acordo com o referencial [Mal69]:

1. *descrição material* – nessa descrição as variáveis independentes são as partículas do material e o tempo t;

2. *descrição referencial*, na qual as variáveis independentes são a posição da partćula $\mathbf{X}$ em uma configuração de referência arbitrária, e o tempo t. Quando a configuração de referência é coincidente com a configuração no tempo $t = 0$, a descrição referencial é frequentemente chamada de *descrição Lagrangeana*. É também chamada por alguns de *descrição material*, ao se utilizar a posição referencial $\mathbf{X}$ para rotular a partícula material $\mathcal{X}$ propriamente dita. É uma descrição naturalmente adequada à formulação via MEF de problemas em mecânica dos sólidos pois as malhas se deformam *amarradas* à estrutura em questão;

3. *descrição espacial*, nesse caso a variável independente é posição $\mathbf{x}$ presentemente ocupada pela partícula no instante de tempo t e o tempo presente t. A descrição espacial fixa a atenção sobre uma determinada região do espaço ao invés de se fixar sobre um determinado volume de material. É a descrição mais empregada em mecânica dos fluidos, também conhecida como *descrição Euleriana*.

Empregando-se uma descrição Lagrangeana o movimento pode ser descrito pelo mapeamento:

$$\mathbf{x} = \phi(\mathbf{X}, t) \tag{III.1}$$

onde $\mathbf{x}$ são as coordenadas atualizadas, ou ainda, *espaciais* da partícula $\mathcal{X}$ que originalmente, no instante $t = 0$ ocupava a posição $\mathbf{X}$ – dita suas *coordenadas materiais*. O mapeamento $\phi(\mathbf{X}, t)$ descreve a trajetória da partćula. Os deslocamentos são definidos pela relação:

$$\begin{gathered} u = \phi(\mathbf{X}, t) - \mathbf{X} \\ \text{ou} \\ u = \mathbf{x} - \mathbf{X} \end{gathered} \tag{III.2}$$

Assume-se que a função ϕ é contínua e inversível, com derivadas conínuas em cada instante de tempo.

III.3 Descrição Cinemática - Medidas de Deformação

III.3.1 Os Gradientes de Deformação e de Velocidade

Medidas de deformação empregadas no caso geral são mais facilmente compreendidas abordando-se inicialmente o conceito de deformação em uma dimensão e, então, generalizando-o para um campo de deslocamentos arbitrários com o emprego do teorema da decomposição polar.

O Gradiente de deformação é dado pela expressão:

$$\mathbf{F}_{ij} = \frac{\partial x_i}{\partial X_j} \tag{III.3}$$

que possibilita mapear (*projetar*) um segmento $d\mathbf{X}$, na configuração indeformada, no seu correspondente, na configuração deformada $d\mathbf{x}$:

$$d\mathbf{x} = \mathbf{F} d\mathbf{X} \tag{III.4}$$

As condições de impenetrabilidade e continuidade exigem que $\mathbf{F}$ seja não-singular, isto é que o seu determinante J seja estritamente positivo e finito:

$$J = \det \mathbf{F} > 0 \tag{III.5}$$

Como o comportamento da maioria dos materiais de interesse no presente trabalho não depende da história das deformações, a formulação incremental das relações constitutivas torna necessária a definição de uma taxa de deformação.

Para isso, define-se o gradiente de velocidades, que mede a diferença de velocidade entre duas partículas infinitesimalmente próximas em uma determinada configuração no tempo:

$$d\dot{\mathbf{u}} = \frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{x}} d\mathbf{x} \tag{III.6}$$

$$L_{ij} = \frac{\partial \dot{u}_i}{\partial x_j} \tag{III.7}$$

que pode ser decomposto em duas parcelas, segundo:

$$\mathbf{L} = \mathbf{D} + \mathbf{W} \tag{III.8}$$

onde $\mathbf{D}$ é o tensor simétrico taxa de deformação, cujos termos são dados por:

$$D_{ij} = \left(\frac{\partial \dot{u}_i}{\partial x_j} + \frac{\partial \dot{u}_j}{\partial x_i} \right) \tag{III.9}$$

e $\mathbf{W}$ é o tensor antimétrico de vorticidade (*spin*), dado por:

$$W_{ij} = \left(\frac{\partial \dot{u}_i}{\partial x_j} - \frac{\partial \dot{u}_j}{\partial x_i} \right) \tag{III.10}$$

A taxa de variação de $\mathbf{F}$ é dada por:

$$\mathbf{F} = \mathbf{L}\dot{\mathbf{F}} \tag{III.11}$$

Uma vez que $\mathbf{F}$ é não-singular, pode ser decomposto segundo o produto de um tensor simétrico positivo-definido por um ortogonal O teorema da decomposição polar estabelece que toda *mudança de forma* pode ser representada por uma rotação de corpo rígido, seguida de extensões (*alongamento - stretch*) — ou vice-e-versa — segundo três direções ortonormais:

$$\mathbf{F} = \mathbf{RU} = \mathbf{VR} \tag{III.12}$$

sendo $\mathbf{U}$ e $\mathbf{V}$, respectivamente, os tensores *direito* e *esquerdo* de extensão, ambos positivos definidos. E $\mathbf{R}$ é ortogonal, $\mathbf{R}^T\mathbf{R} = \mathbf{I}$.

Da ortogonalidade de $\mathbf{R}$ e das relações III.12 conclui-se que $\mathbf{F}$, $\mathbf{U}$ e $\mathbf{V}$ possuem os mesmos autovalores e autovetores. Tais grandezas são identificadas, respectivamente, como os valores e direções das deformações principais.

Alternativamente, formas simetrizadas, definidas a partir do gradiente de deformação, podem também ser empregadas para a medida de deformação.

O tensor de deformação de Green, $\mathbf{C}$, relativo à configuração indeformada é dado por:

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} \tag{III.13}$$

e o tensor de deformação de Cauchy:

$$\mathbf{B}^{-1} = (\mathbf{F}^{-1})^T\mathbf{F}^{-1} \tag{III.14}$$

Os tensores de Green e Cauchy relacionam-se com os tensores de extensão, segundo:

$$\mathbf{C} = \mathbf{U}^2 \qquad\qquad \mathbf{B} = \mathbf{V}^2 \tag{III.15}$$

Como se pode notar, os tensores de Green e Cauchy para deformações totais são simétricos. Empregar estes tensores como medidas de deformação apresenta um pequeno inconveniente, pois para a configuração indeformada ambos resultam no tensor identidade (**I**), não nulo.

III.3.2 Medidas de Deformação Mais Empregadas

Como visto na seção III.3.1 há várias medidas possíveis para a deformação, no entanto é comum em mecânica do contínuo empregar-se uma das três medidas a seguir:

1. velocidade de deformação (taxa de deslocamento, *stretching*)

$$D_{ij} = \frac{1}{2}\left(\frac{\partial \dot{u}_i}{\partial x_j} + \frac{\partial \dot{u}_j}{\partial x_i}\right)$$

2. tensor deformações específicas de Green (também chamado Lagrangeano ou ainda de Green-Lagrange)

$$2\mathbf{E} = \mathbf{C} - \mathbf{I}$$

$$E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}\right)$$

3. tensor de defromações específicas de Almansi (Euleriano)

$$2\mathbf{E}^A = \mathbf{I} - \mathbf{B}^{-1}$$

$$E^A_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right)$$

Qualquer uma dessas medidas de deformação pode ser empregada com qualquer tipo de descrição (Lagrangeana, Euleriana, etc.). Um aspecto comum a todas essas medidas é que apresentam a confortável característica de serem identicamente nulas para movimentos de corpo rígido, sendo essa a sua principal conveniência.

Essas medidas têm distinções importantes. A velocidade de deformação mede a taxa instantânea de deformação mas não dá informação alguma sobre o estado total de deformação do contínuo. Além disso, a sua integração no tempo não possui significado físico tampouco define um tensor, independente da trajetória de forma que informações tais como deformações máximas, principais, direções principais, etc. não são disponíveis em um programa que empregue a velocidade de deformação apenas. Para que se torne representativa, deve ser transformada para alguma outra taxa de deformação ($\dot{E}$) que por sua vez integrada forneça uma medida do estado total de deformação.

As duas outras medidas, Green-Lagrange e Almansi, medem a deformação total. Um *incoveniente* está na sua relação não-linear com os deslocamentos.

III.3.3 Aplicabilidade do Tensor Deformação Linear

A medida de deformação mais conhecida em engenharia é dada pelo tensor deformação linear:

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{III.16}$$

O tensor de deformação linear só pode ser empregado em problemas para os quais as rotações de corpo rígido sejam pequenas. Quando as rotações são grandes, o tensor linear de deformação é não nulo, ainda que efetivamente não haja deformação. Obviamente tal fato o torna inútil como medida de deformação para o caso geral.

Considere-se por exemplo o caso a seguir. Os deslocamentos em qualquer rotação de corpo rígido podem ser escritos na forma:

$$\mathbf{u} = (\mathbf{R} - \mathbf{I})\mathbf{x}$$

$$u_i = (R_{ij} - \delta_{i,j})x_j$$

$$\mathbf{R}\mathbf{R}^T = \mathbf{I}$$

no caso bi-dimensional:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

onde θ é o ângulo de rotação,

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_i}{\partial x_j}\right)$$

$$\epsilon_{ij} = \frac{1}{2}\left(R_{ij} + R_{ji} - 2\delta_{ij}\right) = \frac{1}{2}(\mathbf{R}\mathbf{R}^T - 2\mathbf{I})$$

$$\epsilon = \begin{bmatrix} \cos\theta - 1 & 0 \\ 0 & \cos\theta - 1 \end{bmatrix}$$

cujos termos são sempre não-nulos para uma rotação finita. Isto significa que empregando-se o tensor de deformação linear em um problema que envolva grandes rotações, deformações *espúrias* serão obtidas, consequentemente induzindo tensões, ainda que o corpo tenha sofrido apenas uma rotação de corpo rígido.

A aplicabilidade do tensor de deformação linear pode ser estabelecida da seguinte maneira. Expandindo-se $\cos\theta$ em série de Taylor:

$$\cos\theta \approx \frac{1}{2}\theta^2 + O(\theta^4)$$

pode-se observar que as deformações espúrias decorrentes das rotações de corpo rígido são de ordem θ^2. Portanto, se as rotações são da mesma ordem de grandeza que as deformações, elas poderão ser desprezadas quando as deformações forem de ordem igual ou inferior a 10^{-2}, já que nesses casos o erro é da ordem de 1% devido à rotação de corpo rígido. Vale salientar que o tensor linear pode também ser empregado para deformações moderadamente grandes (≈ 0.1) desde que as rotações sejam pequenas.

Em geral, para estruturas esbeltas e que sejam pouco extensíveis, por exemplo cascas metálicas finas, as relações cinemáticas empregadas deverão ser de fato não-lineares, ainda que as rotações sejam da ordem de 10^{-2}, pois nesses casos o interesse recai sobre deformações da ordem de 10^{-4} a 10^{-3} de forma que o erro devido às rotações seria muito superior.

III.4 Descrição Cinética - Medidas de Tensão

Medidas de tensão e deformação são ditas conjugadas se o seu produto interno fornecer uma medida do trabalho realizado. A taxa das deformações de Green-

Lagrange não é conjugada das tensões de Cauchy. Medidas de tensão e suas medidas de deformação conjugadas são relacionadas na Tab III.1.

As relações entre as medidas de tensão mais empregadas são apresentadas na Tab III.2 (segundo [Bel83]).

Tensor Defomação	Tensor Tensão
Green-Lagrange	2^o Piola-Kirchoff
Almansi	Cauchy
Taxa de Deformação	Cauchy

Tabela III.1: Tensores conjugados para deformação e tensão

	Cauchy σ	Cauchy (co-rotacional) $\check{\sigma}$	2^o Piola-Kirchoff $\mathbf{S}$
σ		$\mathbf{R}\sigma\mathbf{R}^{\mathbf{T}}$	$J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^{\mathbf{T}}$
$\check{\sigma}$	$\mathbf{R}^{\mathbf{T}}\sigma\mathbf{R}$		$J^{-1}\mathbf{U}\mathbf{S}\mathbf{U}^{\mathbf{T}}$
$\mathbf{S}$	$J\mathbf{F}^{-1}\sigma(\mathbf{F}^{-1})^{\mathbf{T}}$	$J\mathbf{U}^{-1}\sigma(\mathbf{U}^{-1})^{\mathbf{T}}$	

Tabela III.2: Transformações entre medidas de tensão

III.4.1 Taxas Objetivas de Tensão

Um aspecto importante da medida de tensão adotada é o da invariância de sua taxa no tempo com relação à orientação do referencial adodato. Tal propriedade é conhecida pelo termo *indiferença referencial* ou ainda *objetividade*. Uma taxa objetiva deve transformar-se como um tensor de segunda ordem quando o sistema de referência sofre uma rotação:

$$\mathring{\sigma}^{*} = \mathbf{R}^{T}\,\mathring{\sigma}\,\mathbf{R}$$

Tal propriedade é fundamental já que as relações constitutivas são frequentemente expressas na forma de taxas, e se as taxas não forem objetivas, o comportamento do material variará segundo o sistema de coordenadas adotado. Obviamente, tal comportamento é fisicamente inadimissível.

A derivada no tempo do tensor de Cauchy ($\dot{\sigma}$) não é objetiva – ainda que a primitiva, o tensor de Cauchy o seja – e portanto não é adequada ao estabelecimento de uma relação constitutiva incremental, considerando-se deslocamentos e deformações arbitrárias.

Para ilustrar esse fato tome-se um sólido sob um estado de tensões σ, simplesmente girando em torno de um eixo qualquer, sem deformação adicional (rotação de corpo rígido). O tensor tensão transforma-se segundo:

$$\sigma^* = \mathbf{R}^T \sigma \mathbf{R} \tag{III.17}$$

e

$$\mathbf{x}^* = \mathbf{R}\mathbf{x} \tag{III.18}$$

sendo que $\mathbf{x}^*$ e $\mathbf{x}$ são instantâneamente coincidentes, porém girando um em relação ao outro, portanto:

$$\mathbf{R} = \mathbf{I} \tag{III.19}$$

$$\dot{\mathbf{R}} = \mathbf{W} = \frac{1}{2}\left(\frac{\partial \dot{u}_i}{\partial x_j} - \frac{\partial \dot{u}_j}{\partial x_i}\right) \tag{III.20}$$

Derivando-se III.17 em relação ao tempo tem-se:

$$\dot{\sigma}^* = \frac{\partial}{\partial t}(\mathbf{R}^T \sigma \mathbf{R}) = \dot{\mathbf{R}}\sigma\mathbf{R} + \mathbf{R}\dot{\sigma}\mathbf{R} + \mathbf{R}\sigma\dot{\mathbf{R}} \tag{III.21}$$

já que não há deformação adicicional, o estado de tensão é constante e portanto $\dot{\sigma} = 0$. Além disso, valendo-se de III.20 e de III.20, tem-se finalmente que:

$$\dot{\sigma}^* = \mathbf{W}\sigma + \mathbf{W}^T\sigma$$

que é sempre não nulo na presença de um estado inicial de tensões.

Taxas objetivas são essencialmente variantes das derivadas no tempo do tensor de Cauchy de forma a se preservar a objetividade. Uma taxa objetiva pode ser obtida pela subtração das quantidades dadas na equação anterior à derivada no tempo do tensor de Cauchy. É a chamada taxa de Jaumann:

$$\check{\sigma}_{ij} = \dot{\sigma}_{ij} - W_{ik}\sigma_{kj} - W_{jk}\sigma_{ki} \tag{III.22}$$

Taxas dessa natureza são ditas *co-rotacionais* por serem interpretadas como sendo a descrição do estado de tensões, tal como observado de um referencial afixado a

um volume elementar do material, girando com velocidade angular igual ao valor instantâneo da velocidade angular ω do material. Muitas outras taxas objetivas podem ser construídas de maneira similar. Esta diversidade na escolha é análoga à multiplicidade de opções encontradas para a medida de deformação e é, ainda hoje, um ponto bastante controverso.

Quaisquer taxas e medidas de deformação podem ser empregadas na formulação de uma equação constitutiva. A expressão mais popular relaciona a taxa de Jaumann com a velocidade de deformação:

$$\check{\sigma} = \mathbf{CD} \tag{III.23}$$

onde **C** nesse caso, é o tensor elástico de quarta ordem (não confundí-lo com o tensor de deformação de Green).

Uma crítica freqüente à taxa de Jaumann diz respeito ao seu comportamento oscilatório em ciclos de cisalhamento puro [Atl84]. No caso do endurecimento cinemático, por exemplo, dá-se preferência a uma outra taxa similar, a taxa de Green-Naghdi dada por:

$$\mathring{\sigma}_{ij} = \dot{\sigma}_{ij} - \Omega_{ik}\sigma_{kj} - \Omega_{jk}\sigma_{ki} \tag{III.24}$$

onde:

$$\Omega = \dot{\mathbf{R}}\mathbf{R}^{T} \tag{III.25}$$

é o tensor antimétrico que fornece a taxa de variação da rotação. Deve-se observar que esse tensor só coincide com o de vorticidade quando se trata instantâneamente de uma rotação de corpo rígido. Note ainda a semelhança entre as expressões III.24 e III.22.

O inconveniente no emprego da taxa de Green-Naghdi reside na necessidade de se efetuar a decomposição polar para a determinação do tensor rotação **R**, o que a torna computacionalmente mais onerosa [Ben86].

Além disso, mostra-se que os erros inerentes ao tratamento discreto

dado à formulação incremental podem gerar tensões residuais espúrias, acumulando energia de deformação, ainda que em ciclos elásticos completos [Koj87].

III.5 Semi-discretização MEF

Em mecânica dos sólidos as equações diferenciais que governam o problema são baseadas no princípio da conservação da quantidade de movimento. Referidas à configuração corrente, as equações de Cauchy são dadas por [Bel83] [Mal69]:

$$\sigma_{ij,j} + b_i = \rho \dot{u}_i \tag{III.26}$$

Na semi-discretização em um primeiro passo obtem-se uma forma fraca para as equações diferenciais do movimento. Em mecânica dos sólidos o princípio dos trabalhos virtuais incluindo a parcela inercial do princípio de d'Alembert é o enfoque usualmente aplicado [Bat82].

Em seguida uma aproximação de elementos finitos para o campo de deslocamentos é aplicada sobre o domínio do problema, conforme

$$u(x,t) = \sum u(t) N(x) \tag{III.27}$$

onde $N(x)$ são as funções de interpolação, dependentes exclusivamente da posição no espaço, e os deslocamentos dos pontos nodais da discretização $u(t)$ incorporam a dependência temporal.

Como resultado final do processo obtem-se um sistema de equações diferenciais ordinárias governando o modelo discreto, tendo o tempo como variável independente.

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}_{int} = \mathbf{f}_{ext} \tag{III.28}$$

onde $\mathbf{M}$ representa a massa do sistema, $\mathbf{f}_{int}$ as forças internas no volume e $\mathbf{f}_{ext}$ as forças externas.

A integração no tempo dessas equações pode ser efetuada através dos chamados métodos diretos ou dos métodos de redução de base.

Os métodos de redução de base ou de superposição modal visam principalmente reduzir a ordem do sistema de equações representadado por III.28. O sistema é representado por um número reduzido de modos considerados dominantes para a resposta.

Uma primeira dificuldade nessa abordagem reside em uma das características dos problemas transientes de curta duração. Nesse tipo de problema praticamente todos os modos contribuem para a resposta, o que impossibilita uma *redução de base*. Além disso a consideração dos efeitos não-lineares obrigariam freqüentes re-avalições da base modal, o que nessa circunstância seria inviável.

Os métodos de integração direta dividem-se em duas grande categorias: os impícitos e os explícitos. Para problemas da mecânica do contínuo os mais empregados podem ser englobados em uma única e extensa classe de operadores de integração: os métodos do tipo Newmark [Bat82].

Do ponto de vista algoritmico os métodos diretos do tipo implícito exigem a solução de um sistema de equações algébricas a cada instante de tempo. Já nos métodos explícitos, quando empregados com uma forma diagonalizada da matriz de massa, a solução de sistemas de equações fica dispensada.

Os métodos explícitos empregam o operador de diferenças centrais para a aproximação de acelerações e velocidades a partir do estabelecimento de um passo de tempo Δt e dos deslocamentos u.

A velocidade entre dois instantes de tempo é considerada constante e aproximada segundo

$$\dot{u}^{n+1/2} = \frac{u^{n+1} - u^n}{\Delta t} \tag{III.29}$$

A aceleração para um instante de tempo n é dada por

$$\ddot{u}^n = \frac{\dot{u}^{n+1/2} - \dot{u}^{n-1/2}}{\Delta t} \tag{III.30}$$

Capítulo IV

Fundamentos Teóricos

IV.1 Equações Básicas Envolvidas na Análise Transiente via MEF

As principais relações envolvidas na análise transiente em mecânica do contínuo, que empregam a técnica de semi-discretização MEF-Diferenças Centrais, podem ser resumidamente dadas segundo:

$$\ddot{U}^n = M^{-1}\left(F^n - P^n\right) \tag{IV.1}$$

$$\Delta U^n = \varphi\left(\ddot{U}^n, \dot{U}^{n-\frac{1}{2}\Delta t}, \Delta t\right) \tag{IV.2}$$

$$U^n = U^{n-1} + \Delta U^n \tag{IV.3}$$

$$\sigma^n = \Psi\left(\Delta\epsilon^n, \sigma^{n-1}\right) \tag{IV.4}$$

$$p^n = \quad \sigma^n \tag{IV.5}$$

onde:

$U^n, \dot{U}^n, \ddot{U}^n$	-	vetores de deslocamentos, velocidades e acelerações nodais para o instante de tempo n ;
$\Delta U^n, \Delta u^n$	-	deslocamentos incrementais;
M	-	matriz de massa diagonalizada;
F^n	-	cargas externas aplicadas;
P^n, p^n	-	vetor de forças internas;
Δt	-	incremento de tempo;
σ^n	-	tensões;
φ	-	operador (explícito) de diferenças centrais para integração no tempo;
Ψ	-	relação constitutiva do material;
∇	-	operador *divergente.*

Caracteres maiúsculos representam grandezas medidas no sistema global de referência e os minúsculos as grandezas correspondentes, referidas ao sistema local dos elementos.

As Eqs. IV.1-IV.3 são expressas no sistema global, ao passo que as Eqs. IV.4 e IV.5 no sistema local dos elementos.

Com vistas à implementação vetorial/paralela da aplicação, cabe aqui ressaltar algumas características fundamentais do algoritmo apresentado. A sua natureza por si só permite, a priori, a identificação de vários focos potenciais de vetorização e paralelismo no processo.

Considerando-se a matriz de massa diagonalizada na avaliação dos efeitos inerciais, a avaliação de M^{-1} na Eq. IV.1 é trivial, resultando em uma simples operação de divisão. As Eqs. IV.2-IV.3 definem a evolução da solução no tempo e, dado o caráter da aproximação por diferenças centrais no tempo, a sua implementação envolve operações simples sobre vetores (adição e produto por escalar).

A natureza intrínseca do MEF, ou seja, a característica de suporte local da sua aproximação, conduz a relações a nível de elemento que são dependentes única e exclusivamente dos valores das grandezas associadas aos pontos nodais que definem cada elemento. Do ponto de vista inter-elemento, tal fato propicia uma total independência do fluxo de dados nos processos representados pelas Eqs. IV.4 e IV.5. Tal independência dá fortes indícios a uma oportunidade natural e atraente para a implementação vetorial/paralela destes processos.

Essa etapa, em um programa convencional destinado à análise transiente e empregando o algoritmo explícito, é reconhecidamente responsável pelo maior consumo de tempo de processador. É a etapa crucial do algoritmo e, portanto, digna de uma análise cuidadosa com vistas a uma eficiente implementação.

IV.2 Estabilidade Numérica

IV.2.1 Condição de Courant

Do ponto de vista prático, uma das mais sérias limitações ao uso do operador explícito de diferenças centrais é a sua estabilidade numérica. Em problemas lineares, a instabilidade fica flagrante pelo crescimento ilimitado da solução. No caso não-linear, a identificação pode não ser imediata. Em problemas elasto-plásticos, por exemplo, uma instabilidade temporária pode permanecer confinada pela dissipação plástica, conduzindo a resultados completamente errôneos.

Na análise linear a condição suficiente para a estabilidade numérica do integração no tempo das equações do movimento com o operador de diferenças centrais é dada pelo limite de Courant (condição de Courant-Friedrich-Lewy – CFL), segundo:

$$\Delta t \leq \frac{2}{\omega_{max}} \left(\sqrt{1+\xi^2} - \xi \right) \qquad \text{(IV.6)}$$

onde ω_{max} é a maior freqüência presente no modelo discreto e $\xi > 0$ é o coeficiente de amortecimento aplicado nessa freqüência [Bel78].

O valor da freqüência máxima presente na discretização é definido no problema de autovalor linearizado associado à equação III.28, dado por:

$$M\ddot{u} + Ku = 0 \tag{IV.7}$$

$$u = \Phi \sin \omega t \tag{IV.8}$$

$$K\Phi = \omega^2 M\Phi \leftrightarrow [\omega_i, \phi_i]\, i = 1, \ldots, m \tag{IV.9}$$

onde $[\omega_i, \phi_i]$ são as frequências e modos de vibração associados e m é a dimensão do problema. A frequência máxima é dada por :

$$\omega_{max} = \underset{i}{\mathrm{MAX}}\, \omega_i \tag{IV.10}$$

A investigação da estabilidade do operador de diferenças centrais para problemas não-lineares não conduz a fórmulas simples como no caso linear. Na prática, adota-se um intervalo de tempo tal que a condição CFL esteja *assegurada* em cada instante de tempo, conduzindo a:

$$\Delta t \leq \alpha \frac{2}{\omega_{max}} \tag{IV.11}$$

onde ω_{max} é calculada a partir dos valores correntes (instantâneos) do módulo tangente e $\alpha \leq 1$.

A validade da solução pode também ser verificada *a posteriori* através da monitoração do balanço da energia do sistema [Bel78]:

$$\left| W^{int} + W^{kin} - W^{ext} \right| < \epsilon \sum |W| \tag{IV.12}$$

onde W^{int}, W^{kin} e W^{ext} são respectivamente as parcelas da energia interna, cinética e externa do sistema em um dado instante, $\epsilon < 1$ é uma tolerância pré-fixada (freqüentemente no intervalo 0.01 – 0.05).

Esse procedimento é indiferente ao tipo de análise e, particularmente adequado à análise não-linear. No caso da instabilidade confinada, a sua manifestação pode ser detectada pelo aumento da parcela W^{int}. Caso não se verifique a relação IV.12 em um determinado instante, procede-se a re-avaliação do intervalo de tempo crítico e o caminho estável da solução é retomado.

IV.2.2 Estimativas para a Freqüência Máxima

Dada a dimensão de uma discretização típica, a solução do problema de autovalor associado é bastante dispendiosa. Por sua vez, a determinação de freqüências na parte superior do espectro apresenta consideráveis problemas numéricos. Como condição necessária para manter a estabilidade em um processo não-linear, o limite de Courant deve ser satisfeito durante toda a integração no tempo (no caso linear, o limite de Courant é uma condição suficiente à estabilidade do algoritmo). Considerando-se a variação do espectro em problemas não-lineares, a verificação *rigorosa* da condição CFL não é viável. Uma abordagem via solução de um problema de autovalor em torno de uma determinada configuração, a cada instante de tempo, seria absurda.

Assim sendo, estimativas para o valor da freqüência máxima presente em uma discretização são usualmente adotadas para a determinação do intervalo de tempo crítico.

Uma alternativa imediata para o estabelecimento do intervalo de tempo crítico considera o intervalo de tempo mínimo necessário ao trânsito de uma onda dilatacional entre dois nós adjacentes da discretização. Dessa forma, tem-se:

$$\Delta t \leq \frac{L_{min}}{C_d} \tag{IV.13}$$

onde L_{min} é a menor distância entre dois nós da discretização e C_d é a velocidade de propagação da onda dilatacional no meio. Dessa forma, o valor da frequência máxima, e por conseguinte o intervalo de tempo crítico, depende, além das características físicas do material modelado, do grau de refinamento empregado na discretização, mais propriamente do *tamanho* dos elementos utilizados.

Uma outra abordagem, bastante atraente, considera a freqüência máxima de cada elemento finito isolado para a definição de um limite superior ao valor da frequência máxima da discretização total [Bel78], [Fla81], [Fla84] e [Lin89]:

$$\omega_{max} \leq \underset{nel}{\mathrm{MAX}} [\omega_{e,max}] \tag{IV.14}$$

isto é, o limite superior para a freqüência máxima da discretização é o supremo do conjunto das freqüências máximas dos elementos.

A freqüência máxima de cada elemento isolado depende basicamente das seguintes características:

- tipo de elemento empregado
- dimensão do elemento
- características físicas do material modelado
- velocidade de propagação no meio

IV.3 Solução de Problemas Estáticos

IV.3.1 Algoritmos de Relaxação Simultânea

Soluções para problemas estáticos podem ser encaradas como as soluções em regime permanente de problemas transientes na presença de amortecimento supercrítico.

A robustez quanto ao tratamento de fenômenos fortemente não-lineares e a simplicidade do algoritmo em si constituem-se em alguns dos atrativos dos algoritmos de relaxação fundamentados no operador de diferenças centrais. Operando exclusivamente sobre os resíduos da equação de equilíbrio estático

$$\mathbf{P(u) = f} \qquad \text{(IV.15)}$$

Os métodos de relaxação do tipo hiperbólica e parabólica lidam essencialmente com arranjos unidimensionais, sem haver a necessidade de montagem ou fatoração de uma matriz de coeficientes associada à equação IV.15.

Essas características colocam os algoritmos de relaxação dinâmica e viscosa como alternativas atraentes para a implementação em processadores vetoriais. Nesse caso o desempenho do processador é a contra-partida para o volume adicional de operações, naturalmente envolvido em um método iterativo.

IV.3.2 Relaxação Dinâmica

Na relaxação dinâmica o processo iterativo busca a solução em regime permanente para o problema de segunda ordem no tempo, associado à equação IV.15 [Ott65], [Und83], [Fel84].

O núcleo do processo iterativo é o operador de diferenças centrais e padece dos habituais problemas de estabilidade.

Selecionam-se uma matriz diagonalizada de pseudo-massas e um fator de amortecimento do tipo Rayleigh, de forma a se obter a convergência para a solução do problema em um número mínimo de iterações. O papel da pseudo-massa é similar ao de um pré-condicionador nos métodos iterativos clássicos. Por outro lado, o papel do amortecimento é o de atenuar as oscilações de mais baixa freqüência para que solução *entre em regime* mais rapidamente [Fel86].

Em [Rer86] propõe-se uma estratégia para a o estabelecimento adaptativo de uma seqüência ótima de incrementos de carga para que sejam minimizados os efeitos inerciais e dissipativos.

A avaliação adaptativa dos parâmetros de relaxação, a massa e o amortecimento, é abordada em [Und83], [Fel86], [Far87] e [Qia88]. De acordo com esses trabalhos o parâmetro de amortecimento crítico é avaliado a cada iteração através de uma aproximação da freqüência mais baixa. Para isso é empregada uma aproximação tangente, diagonal, para a rigidez do sistema em conjunto com a expressão do quociente de Rayleigh para o menor autovalor.

A pseudo-massa do sistema deve ser calculada de forma a garantir também a estabilidade do método. Para isso [Und83] propõe a aplicação do teorema de Gershgorin na estimativa do maior autovalor.

Uma alternativa em [Kul88] e [Alv91] adota $\Delta t = 1$, (*pseudo*) tempo unitário, e avalia a pseudo-massa para fornecer um intervalo de tempo crítico 10% maior, preservando a estabilidade do processo. A avaliação da massa dos elementos pode ser feita revertendo-se as expressões de [Fla84] para as estimativas da freqüencia máxima dos elementos.

As linhas gerais do algoritmo de relaxação dinâmica com controle adaptativo do amortecimento é apresentado na figura IV.1.

1. para j = 1, 2, ... até o número de incrementos de carga
2. inicialize $\mathbf{v}^{-1/2}$, $\mathbf{r}^0 = \mathbf{f}^j$
3. itere para i = 1, 2, ... até que $\left|\mathbf{r}^i\right| \leq tol$
4. $\quad \mathbf{r}^i = \mathbf{f}^j - \mathbf{P}(\mathbf{u}^i)$
5. $\quad \mathbf{v}^{i+\frac{1}{2}} = \frac{(1-\alpha)}{(1+\alpha)}\mathbf{v}^{i-\frac{1}{2}} + \frac{1}{(1+\alpha)}\mathbf{M}^{-1}\mathbf{r}^i$
6. $\quad \mathbf{u}^{i+1} = \mathbf{u}^{i-1} + \mathbf{v}^{i+\frac{1}{2}}$
7. $\quad \mathbf{K} = \mathrm{diag}(K_{ii}) = \max(0, \Delta P_i/\Delta u_i)$
8. $\quad \alpha = \sqrt{\frac{\mathbf{u}^T\mathbf{K}\mathbf{u}}{\mathbf{u}^T\mathbf{M}\mathbf{u}}}$
9. fim loop i
10. fim loop j

Figura IV.1: Algoritmo de Relaxação Dinâmica

IV.3.3 Relaxação Viscosa

Uma curiosidade a respeito desse método é que, segundo [Ode73] sua aplicação para a solução de sistemas de equações algébricas lineares e não lineares remonta à época dos computadores analógicos. Ainda nesse trabalho é proposta a sua aplicação como alternativa aos métodos incrementais implícitos, com ou sem correção do tipo Newton-Raphson.

Analogamente à relaxação dinâmica, a relaxação viscosa (VR) parte do problema de primeira ordem no tempo associado à equação IV.15.

$$\mathbf{C}\dot{\mathbf{u}} + \mathbf{P}(\mathbf{u}) = \mathbf{f} \tag{IV.16}$$

Empregando o operador de diferenças centrais para aproximar as velocidades:

$$\dot{\mathbf{u}} = \frac{\mathbf{u}^{t+\Delta t} - \mathbf{u}^{t}}{\Delta t} \tag{IV.17}$$

pode-se chegar à expressão,

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^{t} - \Delta t \mathbf{C}^{-1}(\mathbf{P}(\mathbf{u}) - \mathbf{f}) \tag{IV.18}$$

Considerando-se $\mathbf{C} = \mathrm{diag}(c_{ii})$ Seu desempenho para problemas lineares é equivalente ao dos métodos iterativos básicos que não empregam pré-condicionadores, como o método de Jacobi [Fel86].

Reescrevendo-se IV.18 tem-se,

$$\Delta \mathbf{u} = -\Delta t \mathbf{C}^{-1} \Phi \tag{IV.19}$$

onde $\Phi = \mathbf{P}(\mathbf{u})$ f é o vetor de resíduos.

O intervalo de tempo Δt e a matriz diagonal C devem ser escolhidos de forma a assegurar a estabilidade numérica do processo. Aceleradores de convergência do tipo *line search* podem ser empregados conforme [Zie85] [Alv87] tornando o processo mais atraente.

$$\mathbf{u}_i = \mathbf{u}_{i-1} + s\Delta\mathbf{u} \tag{IV.20}$$

O parametro de *line search* s é determidado de forma que Δu seja ortogonal ao resíduo na k-ésima iteração ou ainda, com base na convergência do método para a solução assintótica de IV.16 e a predominância do seu menor autovalor, de forma que:

$$\mathbf{u} - \mathbf{u}_\infty = \mathbf{a}e^{-\lambda_{min}t} \tag{IV.21}$$

sendo a um certo vetor de amplitudes. Diferenciando-se IV.21 em relação ao tempo e empregando-se a IV.17, tem-se:

$$\mathbf{u}_\infty = \mathbf{u}^t + \frac{1}{\Delta t\lambda_{min}}\Delta\mathbf{u} = \mathbf{u} + s\Delta\mathbf{u} \tag{IV.22}$$

derivando-se novamente e notando que:

$$\Delta(\Delta\mathbf{u}) = -\Delta t\lambda_{min}\Delta\mathbf{u} \tag{IV.23}$$

chega-se finalmente à expressão:

$$s = -\frac{\Delta\mathbf{u}^T\Delta\mathbf{u}}{\Delta\mathbf{u}^T\Delta(\Delta\mathbf{u})} \tag{IV.24}$$

O parâmetro de aceleração s está portanto associado ao menor autovalor do problema pseudo-parabólico. A monitoração de seus valores subseqüentes fornece uma indicação da dominância de λ_{min} na solução, e portanto a aceleração só deve ser aplicada quando se verificar a seguinte relação:

$$\frac{\Delta s}{s} \leq s_{tol} \qquad \text{(IV.25)}$$

onde s_{tol} é uma tolerância pré-fixada.

As principais etapas do algoritmo estão apresentadas na figura IV.2 para a sua versão acelerada.

1. para j = 1, 2, ... até num. incrementos de carga
2. para i = 1, 2, 3, ... até $|\mathbf{r}^i| \leq 0$
3. $\mathbf{r}^i = \mathbf{f}^j - \mathbf{P}(\mathbf{u}^i)$
4. $\Delta\mathbf{u}^i = \Delta t \mathbf{C}^{-1}\mathbf{r}^i$
5. $s = -(\Delta\mathbf{u}^T\Delta\mathbf{u})/(\mathbf{u}^T\Delta\Delta\mathbf{u})$
6. se $|\Delta s/s| > s_{tol} \Rightarrow s = 1$
7. $\mathbf{u}^{i+1} = \mathbf{u}^i + s\Delta\mathbf{u}$
8. fim do loop i
9. fim do loop j

Figura IV.2: Algoritmo de Relaxação Viscosa Acelerada

Capítulo V

Operador Particionado Explícito-Explícito

V.1 Generalidades

Uma forma de aliviar as consequências das restrições impostas pelo limite de Courant, pode ser obtida com o emprego dos operadores particionados [Bel87].

Nesse enfoque, o domínio é subdividido em regiões ou partições sobre as quais são aplicados operadores de integração distintos (implícitos e/ou explícitos) e diferentes intervalos de tempo, de acordo com as exigências de precisão e estabilidade inerentes ao operador adotado.

Reside aí uma das principais vantagens dessa estratégia quando, por exemplo, considera-se a modelagem de problemas de interação entre meios que apresentam uma forte distinção entre suas características elasto-dinâmicas, ou então, diferenciação acentuada quanto ao grau de refinamento (discretização fina e pobre).

Na Figura V.1 apresenta-se um exemplo esquemático onde foram consideradas três partições (1, 2 e 3) de acordo com o grau de refinamento empregado na discretização. Para o caso em questão, observe a necessidade da inclusão, por exemplo, na partição 1 de pelo menos uma camada de elementos maiores na interface com a partição 2. O mesmo princípio deve ser observado na formação de partições, quando a distinção se deve às características dos materiais adjacentes.

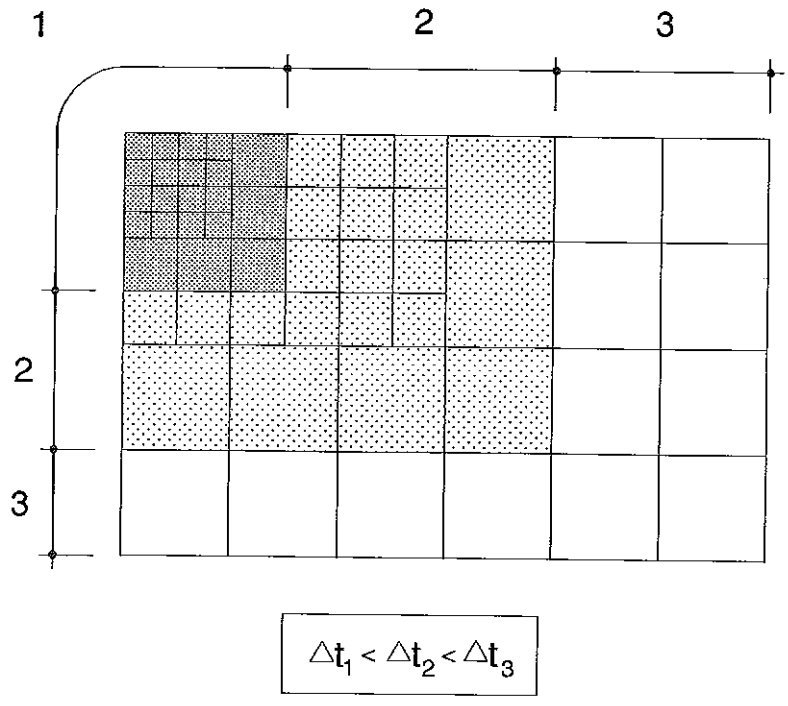


Figura V.1: Domínio Particionado

No presente estudo foi implementado o operador particionado do tipo explícito-explícito. Como uma decorrência natural da estratégia de implementação, o operador explícito convencional também é disponível.

V.2 Descrição do Algoritmo

Em linhas gerais, o processo de integração no tempo é controlado pelo *relógio-global* (*master-clock*) e pelos *relógios-das-partições* (*group-clock*). O relógio de cada partição é atualizado segundo o intervalo de tempo específico da própria, sendo o relógio-global atualizado pelo menor intervalo de tempo dentre aqueles atribuídos às partições (*subcycling*).

À medida em que a integração no tempo se desenvolve, sempre que uma partição estiver atrasada em relação ao relógio-global, os elementos que a definem são atualizados. A atualização dos elementos envolve o cálculo de deformações, tensões, e das forças nodais equivalentes. A atualização das grandezas nodais dá-se através de um procedimento análogo, baseado em um *relógio-nodal*. Na Figura V.2 é apresentado, em linhas gerais, o algoritmo particionado explícito-explícito.

```
inicialize p/ t = 0
defina Δt_no a partir de Δt_el
incremente RELÓGIO_NODAL por Δt_no
LOOP steps = 1,2 ... NSTEPS
incremente RELÓGIO_GLOBAL por Δt_min
calcule f(t)
   LOOP nas partições
      SE RELÓGIO_PARTIÇÃO ≤ RELÓGIO_GLOBAL
         calcule forças internas para o GRUPO (*)
         incremente RELÓGIO_PARTIÇÃO por Δt_g
   ENDLOOP partições
   LOOP nos = 1,2 ... NUMNP
      SE RELÓGIO_NODAL < RELÓGIO_GLOBAL
         atualize grandezas nodais p/ o nó
         incremente RELÓGIO_NODAL por Δt_no
   ENDLOOP nos
ENDLOOP steps
```

(*) precisando de valores nodais não atualizados, INTERPOLE com $\dot{u}$ constante

Figura V.2: Algoritmo particionado Explícito-Explícito

Consequentemente, em uma análise particionada não é necessário atualizar todos os elementos e pontos nodais a cada intervalo de tempo, mas apenas aqueles defasados em relação ao relógio-global.

V.3 Aspectos da Implementação

No presente estudo foi adotado um esquema de partições fixas, ou seja, uma vez definidas as listas de elementos e os intervalos de tempo de cada partição, esses dados permanecem inalterados ao longo da análise. Isso evita as penalizações inerentes ao gerenciamento e manipulação de uma estrutura de dados dinâmica durante a integração no tempo do problema.

Uma vetorização eficiente desse processo pode ser obtida através do agrupamento contíguo em memória dos elementos pertencentes a uma partição. Para facilitar esse aspecto da vetorização o agrupamento pode ser efetuado por ocasião da entrada de dados do problema. Melhoramentos adicionais podem ser alcançados pela ordenação das equações, iniciando pelas equações interiores a cada partição, e por fim as equações correspondentes às interfaces entre partições.

As partições são empilhadas de tal forma que as listas de elementos são contíguas na memória. Ponteiros indicam o início e fim de cada partição. Além disso, as particcões são ordenadas segundo a seqüência crescente dos seus intervalos de tempo. Assim, à medida em que o processo de integração no tempo evolui, o relógio-mestre é incrementado e é avaliada a extensão das pilhas de elementos (e equações) que deverão ser percorridas nos loops de atualização (Figura V.3). Com essa organização dos dados os loops de atualização são inicializados uma única vez. Esta operação é feita com stride unitário sobre os vetores mais longos possíveis, a cada intervalo de tempo.

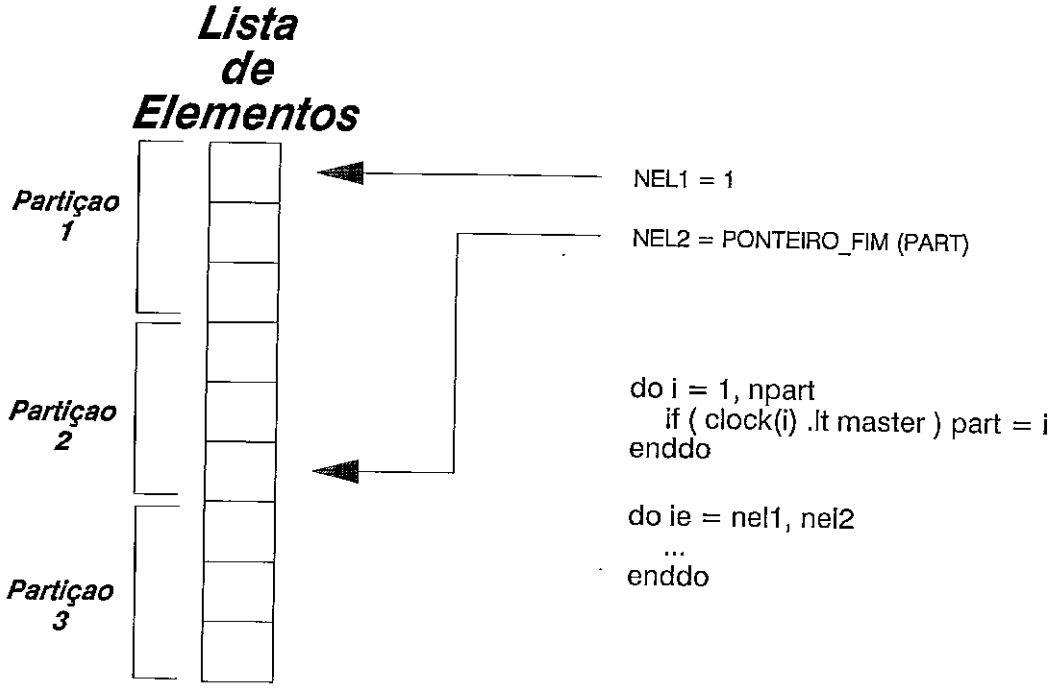


Figura V.3: Estrutura de Dados para as Partições

Um tratamento especial é dado às equações das interfaces entre partições, sempre que os relógios das respectivas partições não coincidirem. Nesse caso, os deslocamentos nodais são extrapolados, assumindo-se velocidade constante, a partir dos últimos valores disponíveis. Posteriormente, a montagem das forças nodais equivalentes para as interfaces é efetuada segundo a técnica de mapeamento inverso (que será tratado mais adiante), porém com a utilização de uma máscara para considerar apenas os elementos que tenham sido atualizados sob o mesmo incremento do relógio mestre. Essa é a única ocorrência de instruções sob o controle de máscara, ao longo do processo de atualização nodal, no algoritmo para análise particionada.

Capítulo VI

Avaliação das Forças Internas

VI.1 Expressão Geral

No âmbito da aproximação por elementos finitos, a avaliação das forças nodais equivalentes ao estado de tensões no interior de cada elemento é dada pela integral:

$$\mathbf{p}_e = \int_{V_e} \mathbf{B}^T \sigma dV_e \tag{VI.1}$$

onde V_e é o volume do elemento.

No caso geral, os termos sob a integral são não-lineares, e no processo incremental devem ser constantemente atualizados, tanto $\mathbf{B}_e$, o operador diferencial discreto, quanto σ, o estado de tensões.

VI.2 Integração Numérica

VI.2.1 Quadratura de Gauss "Full-Rank"

Dada a arbitrariedade da geometria dos elementos, a integral é avaliada numericamente pelo processo de quadratura de Gauss. No caso bi-dimensional o processo de quadratura de Gauss aplicado à expressão VI.1, é dado por :

$$p_e = \sum_{i}^{N} \sum_{j}^{N} \alpha_i \alpha_j \mathbf{B} \mid_{r_i,s_j} \sigma \mid_{r_i,s_j} \qquad \text{(VI.2)}$$

onde:

N	-	ordem da integração numérica
(r_i, s_j)	-	coordenadas naturais do ponto de integração
α_i , α_j	-	coeficientes de ponderação de Gauss
$\mathbf{B} \mid_{r_i,s_j}$	-	o operador diferencial discreto
$\sigma \mid_{r_i,s_j}$	-	o estado de tensões

A ordem de integração a ser empregada depende do grau da função polinomial encontrada no segundo membro da Eq. VI.2. Por sua vez, estas dependem das funções de interpolação empregadas no elemento e do aspecto (grau de distorção) desse elemento.

Em um caso bi-dimensional, para elementos quadriláteros bilineares regulares, a integração *exata* (*full rank*) pelo processo de quadratura de Gauss requer $N = 2$, ou seja, quatro pontos de integração no interior de cada elemento. O termo *regular* aqui empregado indica um elemento cujo aspecto geométrico desvie-se *ligeiramente* da forma de um quadrado.

Quanto maior o número de pontos necessários à correta integração de VI.2, maior o será o tempo de processamento requerido. Leve-se em conta a quantidade de elementos em uma discretização típica e o número de intervalos de tempo empregados para a análise transiente. Não resta outra alternativa para a agilização dos cálculos senão o recurso a uma redução no número de pontos por elemento.

VI.2.2 Integração Reduzida e Controle de Modos Espúrios

É sabido que uma vez asseguradas as condições de convergência, o modelo de deslocamentos do MEF proporciona um limite inferior para a energia de deformação do sistema considerado. Fisicamente, isso equivale a afirmar que a rigidez do sistema estrutural fica superestimada [Bat82].

Tal fato estimulou a investigação das técnicas de integração seletiva e de integração reduzida. Essas técnicas procuram uma melhor aproximação para a rigidez do sistema, através de integração numérica segundo uma ordem deliberadamente inferior à *exata*.

Em Mecânica do Contínuo, a integração seletiva adota ordens de integração diferenciadas para os termos correspondentes à energia de distorção (cisalhante) e expansão (dilatacional) [Hug87]. O termo pode ainda ser aplicado para os casos onde se empreguem ordens de integração distintas segundo alguma orientação espacial em particular – em um caso de anisotropia, por exemplo [Koh87].

Já o termo integração reduzida, refere-se a uma ordem de integração numérica única para todas as parcelas, em cada elemento.

O emprego de técnicas de integração reduzida e/ou seletiva exige, segundo [Bat82], que:

- o elemento não contenha nenhum modo espúrio de energia nula
- o elemento represente os estados de deformação constante

Atualmente, a técnica de integração reduzida tem sido muito utilizada no âmbito do MEF. As vantagens se apresentam tanto do ponto de vista da qualidade dos resultados como da sua eficiência computacional [Hug89].

Para elementos quadriláteros bi-lineares, por exemplo, a integração reduzida vale-se de um único ponto de Gauss localizado no centróide do elemento. Os elementos assim tratados reproduzem um campo de deformações constante em seu interior.

No entanto, os operadores diferenciais discretos assim obtidos apresentam uma deficiência de posto (*rank*) e, no contexto da Mecânica do Contínuo, modos de energia nula podem se manifestar. Essa manifestação pode ser detectada pela alternância do sinal das velocidades ao longo de uma linha de pontos nodais, em um fenômeno conhecido no jargão como *hourglassing*. As condições de contorno do problema têm também influência, podendo em alguns casos inibir o *hourglass*.

Os valores de tensões e deformações não são afetados pela manifestação moderada (pequenas amplitudes, em uma faixa *linear*) do *hourglass* pois estes são ortogonais aos modos de deformação próprios do elemento (os de corpo rígido e os demais de energia de deformação não nula). Porém a sua manifestação exagerada, factível uma vez que a malha não tem como lhes opor *resistência*, degrada completamente a solução.

O controle de *hourglassing* pode ser efetuado através da inclusão de forças viscosas e/ou elásticas para opor resistência a esses modos de deformação. e assim estabilizar a solução [Bel84] e [Fla81].

No caso dos quadriláteros planos, apresentam-se dois modos impróprios (*espúrios*) ortogonais. Para a sua estabilização definem-se duas taxas generalizadas de hourglass, $\dot{q}_\alpha$ e as forças de resistência correspondentes, Q_α. As taxas ou intensidades de *hourglass* são determinadas pela projeção do campo de velocidades nodais do elemento sobre os seus vetores de hourglass:

$$\dot{\mathbf{q}}_1 = \sum_{I=1}^{4} \mathbf{B}_I^H \mathbf{v}_{xI} \tag{VI.3}$$

$$\dot{\mathbf{q}}_2 = \sum_{I=1}^{4} \mathbf{B}_I^H \mathbf{v}_{yI} \tag{VI.4}$$

onde os vetores de hourglass são dados por:

$$\mathbf{B}_I^H = \mathbf{h}_I - c_x^H \frac{\partial \mathrm{N}_I}{\partial \xi} - c_y^H \frac{\partial \mathrm{N}_I}{\partial \eta} \tag{VI.5}$$

e

$$\mathbf{h}_I = [+1, -1, +1, -1] \tag{VI.6}$$

é a forma básica dos modos de hourglass nas coordenadas naturais do elemento (ξ, η), e ainda:

$$c_x^H = \frac{1}{J}(x_1 - x_2 + x_3 - x_4) \tag{VI.7}$$

$$c_y^H = \frac{1}{J}(y_1 - y_2 + y_3 - y_4) \tag{VI.8}$$

$$\begin{aligned} \frac{\partial \mathrm{N}_I}{\partial \xi} &= J \frac{\partial \mathrm{N}_I}{\partial x}(0,0) \\ \frac{\partial \mathrm{N}_I}{\partial \eta} &= J \frac{\partial \mathrm{N}_I}{\partial y}(0,0) \end{aligned} \tag{VI.9}$$

onde x_I, y_I são as coordenadas dos pontos nodais e J é o jacobiano no centróide do elemento.

Os modos de *hourglass* e as resistências são relacionadas através de uma lei visco-elástica:

$$\mathbf{Q}_\alpha = \alpha_H(\rho E A)^{1/2}\left(\frac{1}{4}\dot{\mathbf{q}}_\alpha + \omega_{max}\mathbf{q}_\alpha\right) \qquad \alpha = 1,2 \tag{VI.10}$$

onde:

A	=	área do elemento;
α_H	=	parâmetro de controle de *hourglass*;
ρ	=	densidade do material;
E	=	módulo de elasticidade, e
ω_{max}	=	freqüência máxima do elemento

A equação VI.10 é implementada definindo-se as quantidades intermediárias $\bar{\mathbf{Q}}_\alpha$ e $\dot{\bar{\mathbf{Q}}}_\alpha$, respectivamente a parcela *elástica* e a parcela *viscosa* das *tensões* de estabilização, segundo:

$$\dot{\bar{\mathbf{Q}}}_\alpha = \alpha_H(\rho E A)^{1/2} \qquad \alpha = 1,2 \tag{VI.11}$$

$$\bar{\mathbf{Q}}^{n+1} = \bar{\mathbf{Q}}^n + \Delta t \omega_{max}\dot{\bar{\mathbf{Q}}}_\alpha \tag{VI.12}$$

A quantidade $\bar{\mathbf{Q}}^{n+1}$ é armazenada e a quantidade $\mathbf{Q}_\alpha$, as tensões totais de estabilização, é avaliada segundo:

$$\mathbf{Q}_\alpha = \bar{\mathbf{Q}}_\alpha^{n+1} + \frac{1}{4}\dot{\bar{\mathbf{Q}}}_\alpha \tag{VI.13}$$

Finalmente, as forças nodais equivalentes a $\mathbf{Q}_\alpha$ são dadas por:

$$\begin{aligned} \mathbf{f}_{xI}^H &= V\mathbf{B}_I^H\mathbf{Q}_1 \\ \mathbf{f}_{yI}^H &= V\mathbf{B}_I^H\mathbf{Q}_2 \end{aligned} \tag{VI.14}$$

onde V é o volume do elemento.

O controle viscoso tende a amortecer as amplitudes dos modos espúrios sem no entanto impedir a sua manifestação. Já o controle elástico procura restringir os efeitos dos modos de *hourglassing* através de forças restauradoras, recuperando a estabilidade da malha. A combinação das duas técnicas é uma estratégia bastante empregada, ambas possuindo implementação simples e de custo computacional baixíssimo, tornando a integração reduzida viável e muito atraente.

VI.3 Amortecimento Viscoso Artificial

O amortecimento viscoso do tipo Rayleigh é freqüentemente empregado em dinâmica. Segundo essa hipótese a matriz de amortecimento é dada por:

$$C = \alpha M + \beta K \tag{VI.15}$$

sendo M e K, respectivamente, as matrizes de massa e de rigidez do sistema.

O coficiente α tem influência sobre as oscilações de baixa freqüência, enquanto que o coeficiente β exerce influência mais acentuada sobre aquelas de frequência mais alta.

Uma vez que, nos métodos explícitos, a matriz de rigidez nunca é de fato formada, a inclusão do amortecimento viscoso para as altas frequências é efetuada a nível da avaliação de forças internas.

Em linhas gerais, termos viscosos são acrescentados aos vetores de forças internas, com base na taxa de deformação e em uma rigidez fictícia para os elementos, simulando assim a parcela do amortecimento proporcional à rigidez.

A relação empregada para definir esse amortecimento pode ser deduzida a partir da análise das equações homogêneas modais do movimento. Para um sistema com amortecimento de Rayleigh, sendo $\alpha = 0$:

$$\ddot{z} + \beta\omega^2\dot{z} + \omega^2 z = 0 \tag{VI.16}$$

Se for considerada a equação correspondente à mais alta freqüência, então $\omega = \omega_{max}$. Tomando-se a mesma equação para o sistema sujeito ao amortecimento crítico:

$$\ddot{z} + 2\xi\omega\dot{z} + \omega^2 z = 0 \tag{VI.17}$$

Comparando-se VI.16 e VI.17 tem-se que:

$$\beta = \frac{2\xi}{\omega_{max}} \tag{VI.18}$$

Se considerarmos forças de volume do tipo viscoso, de forma a gerar amortecimento proporcional à rigidez do sistema, podemos escrever:

$$\beta\mathbf{K}\dot{u} = \int_V \mathbf{B}^T\hat{\sigma}^{vis}dV \tag{VI.19}$$

reescrevendo o lado esquerdo de VI.19 em termos de operações a nível de elemento, obtem-se:

$$\beta\int_V \mathbf{B}^T\mathbf{C}\mathbf{B}dV\dot{u} = \int_V \mathbf{B}^T\hat{\sigma}^{vis}dV \tag{VI.20}$$

Introduzindo o vetor gradiente de velocidades $\hat{\eta} = \mathbf{B}\dot{u}$ segue-se de VI.20 que:

$$\int_V \mathbf{B}^T\left(\beta\mathbf{C}\hat{\eta} - \hat{\sigma}^{vis}\right)dV = 0 \tag{VI.21}$$

que para ser satisfeita requer que o termo entre parêntesis seja identicamente nulo. Chega-se então à expressão que define as forças de volume a serem consideradas para se introduzir o amortecimento desejado:

$$\hat{\sigma}^{vis} = \beta \mathbf{C}\hat{\eta} = \frac{2\beta}{\omega_{max}}\mathbf{C}\hat{\eta} \qquad \text{(VI.22)}$$

A matriz constitutiva C não precisa ser rigorosamente a expressão da relação constitutiva do material, basta que seja uma aproximação. Nesse caso pode ser também interessante decompor as forças viscosas segundo parcelas hidrostática e desviatória.

O valor de ω_{max} a ser empregado em (VI.22) pode também ser aproximado pelas estimativas para as freqüências máximas dos elementos.

VI.3.1 Aspectos da Implementação

Uma diretriz importante para a implementação vetorial eficaz é a de que a formulaç ao do elementos escolhidos seja a mais simples possível. Elementos de mais baixa ordem são particularmente ideais pois o número reduzido de graus de liberdade por elemento, a geometria simples e o grau de suas funções permitem que as relações cinemáticas sejam facilmente desenvolvidas *in-line* e embutidas diretamente nas rotinas de avaliação das forças internas.

Dessa forma, não é comprometida a vetorização, no sentido dos elementos, das rotinas envolvidas nesse processo.

As possibilidades de otimização do código a tempo de compilação podem ser aumentadadas através da verificação dos tipos de instruções geradas, da alocação dos registradores e da movimentação dos dados (*LOAD's e STORE's*). Essas verificações podem ser realizadas através da análise das listagens do pseudo-assembler produzidas pelo compilador.

VI.4 Comportamento Não-linear Físico

VI.4.1 Generalidades

Para os propósitos do presente trabalho, o modelo de material considerado para implementação foi o de Prandtl-Reuss, obedecendo ao critério de plastificação de Von Mises, com lei associativa. Este modelo é adequado à modelagem de metais em processos isotérmicos.

O criterio é verificado e o estado de tensões é atualizado em cada ponto de integração de Gauss, para todos os elementos utilizados na discretização. Nos pontos em que se verifique a violação do critério, caracterizando o regime plástico, o estado de tensões deve ser corrigido.

VI.4.2 Definição do Problema

A partir de uma história de deformações e das condições iniciais, deve-se encontrar a história de tensões correspondente através da integração das equações constitutivas do material.

No caso geral, assumindo-se uma combinação linear dos comportamentos isotrópico e cinemático da superfície de escoamento. Dados os valores iniciais do estado de tensões (σ), da tensão de escoamento (k), da posição do centro da superfície (α) no espaço das tensões desviatórias, determine a evolução no tempo de σ, k, e α tal que as seguintes relações sejam satisfeitas [Hug89] [Sim88]:

a relação incremental para o estado de tensões em função das taxas das deformações total e plástica, por hipótese separáveis:

$$\dot{\sigma} = c(\dot{\epsilon} - \dot{\epsilon}^{pl}) \tag{VI.23}$$

a taxa de expansão da superfície de escoamento:

$$\dot{R} = \sqrt{\frac{2}{3}}\beta H' \dot{\epsilon}^{pl} \tag{VI.24}$$

a taxa de deslocamento do centro da superfície no plano das tensões desviatórias:

$$\dot{\alpha} = \frac{2}{3}(1-\beta)H' \dot{\epsilon}^{pl} \tag{VI.25}$$

a função de escoamento, que determina a admissibilidade do estado de tensões:

$$f(\sigma) = \frac{1}{2}\xi'_{ij}\xi'_{ij} - k^2 \leq 0 \tag{VI.26}$$

onde

$$\xi = \sigma - \alpha \tag{VI.27}$$

e

$$\xi'_{ij} = \xi_{ij} - \frac{1}{3}\xi_{kk}\delta_{ij} \tag{VI.28}$$

são as componentes desviatórias do estado de tensões *deslocado*.

De uma forma geral a integração das relações constitutivas para materiais elasto-plásticos emprega dois passos fundamentais, de forma a estabelecer a admissibilidade do estado de tensões em cada instante de tempo (eq VI.26). Em uma primeira *tentativa* considera-se um incremento de tensões no regime elástico. Caso o estado total resultante viole o critério de escoamento emprega-se alguma técnica para restabelecer um estado de tensões tal que $f(\sigma) = 0$, ou seja retornar à superfície definida pela equação VI.26.

Os chamados métodos de retorno possuem diversas alternativas: retorno radial, preditor tangente com retorno radial e o algoritmo do ponto intermediário [Kri77].

VI.4.3 Implementação do Algoritmo Preditor Tangente com Retorno Radial

Trata-se de um dos algoritmos mais tradicionais para a integração das equações constitutivas incrementais em elasto-plasticidade. Nele o método para retornar o estado de tenões à superfície de escoamento emprega uma série de subincrementos de deformação [Owe80], [Bat82], [Che88], [Kri77].

Uma interpretação geométrica do algoritmo justifica o seu nome. Em cada subincremento toma-se uma direção tangente à superfície no plano das tensões desviatórias para a previsão de um novo estado de tensões. Esta previsão tangente leva à um estado subsequente fora da superfície. Por fim, este estado é projetado sobre a superfície de escoamento segundo a direção normal (radial) à superfície. A figura VI.1 ilustra esquematicamente o processo empregado.

O número de subincrementos empregados é estabelecido de forma a assegurar uma boa precisão da aproximação.

Em relação à estratégia de implementação com vistas à vetorização desse processo devem ser ressaltados alguns pontos importantes. Os elementos (pontos de Gauss) que estão em regime plástico não ocupam necessáriamente posições contíguas na estrutura de dados, pelo contrário, podem estar aleatoriamente dispersos em uma análise típica.

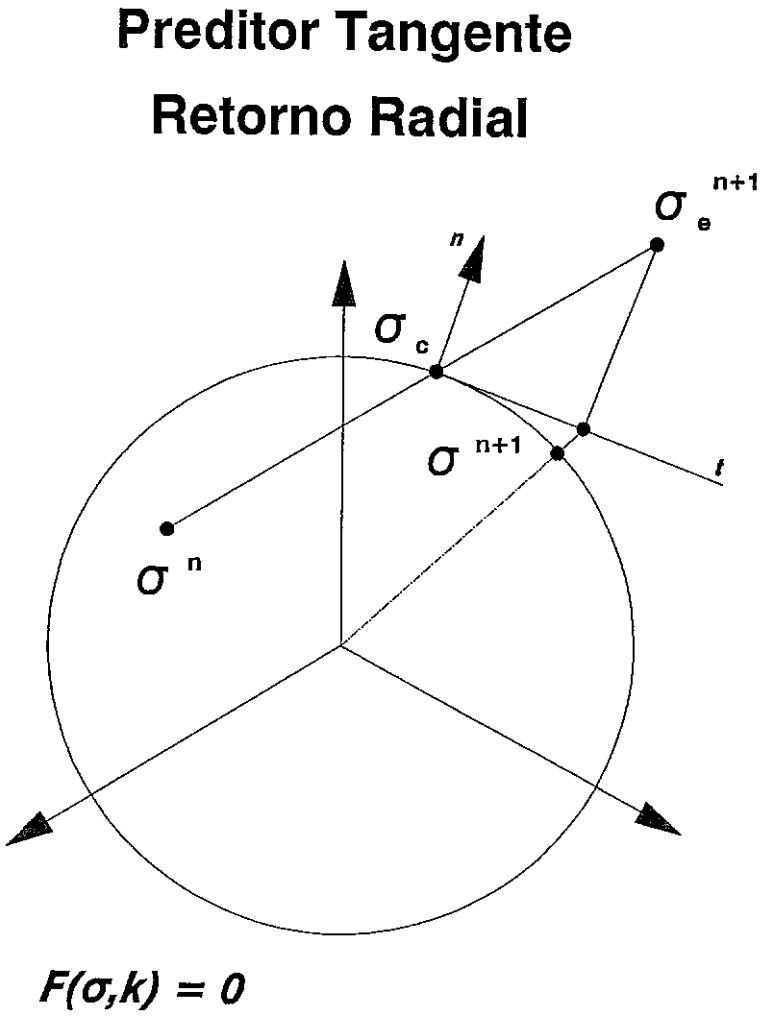


Figura VI.1: Preditor Tangente com Retorno Radial

Em virtude disto, a implementação do processo de correção do estado de tensões poderia valer-se de instruções sob desvios condicionais do tipo IF-THEN-ELSE, que podem ser vetorizadas. Contudo, esse é o caso onde o compilador VS-FORTRAN gera instruções vetoriais sob o controle de máscara (*masked vector instructions*). Tal situação, apesar de resolver o problema de vetorização, apresenta uma performance bastante pobre devida à penalização inerente a esse modo de operação.

O diagrama da (Figura VI.2) ilustra a idéia fundamental do modo de processamento vetorial sob o controle de máscara. Vamos imaginar que em um determinado trecho do programa vetorizado se dá um desvio condicional no fluxo de operações para o fluxo A ou, alternativamente para B. O registrador vetorial (REG), alocado para a operação, conterá os resultados da operação com A para todas as posições de uma seção do vetor.

A máscara (MASK) determinará quais posições serão aproveitadas (armazenadas em STO) ou descartadas, dependendo da verificação ou não da condição de teste para cada posição da *vector section*. Instruções especiais (*VECTOR LOAD MATCHED e VECTOR STORE MATCHED*) valem-se do conteúdo lógico da máscara para carregar/armazenar ou descartar o resultado.

Assim, não há efetivamente *economia* de operações, além disso há o *overhead* associado à geração e manipulação da máscara, sendo talvez esse modo de operação vetorial o mais penalizado.

A contrapartida está na relação entre a velocidade da operação em modo vetorial e em modo escalar, o que torna a construção vantajosa.

Construções IF-THEN-ELSE(-ELSEIF)-ENDIF são mais onerosas pois possuem múltiplos fluxos de dados. Sempre que possível deve-se procurar uma construção IF-ENDIF, mais simples.

Do que foi exposto, se apresenta como uma boa alternativa, agrupar os elementos que violam o critério em uma lista compacta de apontadores e, na seqüência, processá-los através de endereçamento indireto (Figura VI.3).

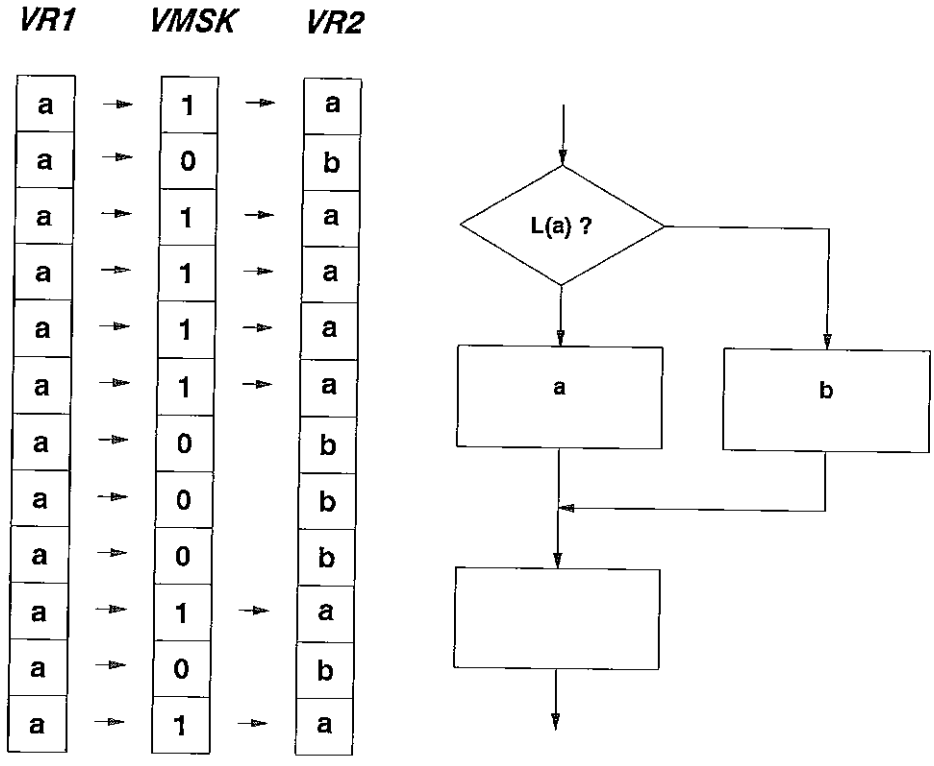


Figura VI.2: Processamento sob Controle de Máscara

Apesar de o conjunto de instruções da CPU 3090 incluir aquelas necessárias ao tratamento de listas compactadas (*VECTOR LOAD COMPRESSED* e *VECTOR STORE EXPANDED*), versões anteriores do compilador (versão 2.3 inclusive) não geravam o código necessário. Felizmente, o problema foi superado na versão 2.4, e a vetorização desse trecho do código é realizada a contento.

A correção do estado de tensões é realizada através de uma técnica puramente incremental (Euler-forward) tomando-se um determinado número de subincrementos de deformações e de tensões elasto-plásticas. Em implementações convencionais, o número de subincrementos é calculado e aplicado individualmente a cada elemento.

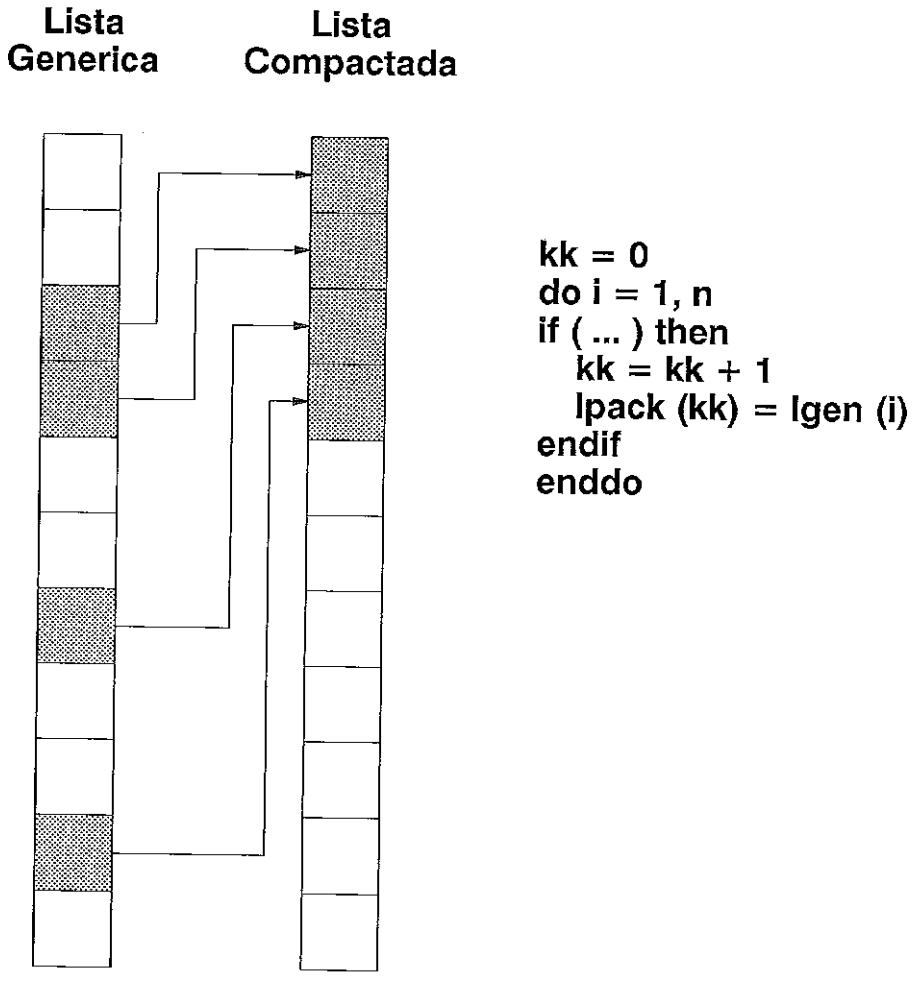


Figura VI.3: Formação de uma Lista Compactada

Para a vetorização do processo (no sentido dos elementos), um número de subincrementos a ser aplicado uniformemente a todos os elementos em regime elasto-plástico é definido com base no estado de tensões mais inconsistente dentre todos os elementos da lista.

A programação da matriz elasto-plástica associada ao critério de Von Mises é relativamente simples e as expressões envolvidas no processo subincremental podem ser totalmente desenvolvidas *in line*.

As linhas gerais da implementação adotada estão apresentadas na figura (Figura VI.4). As expressões VECTOR, MASKED, SCALAR e INDIRECT ADDRESSING, colocadas à esquerda das principais etapas do algoritmo, refletem a natureza das instruções geradas pelo compilador.

VECTOR	1. loop sobre os elementos passíveis de atualização calcule as deformações incrementais calcule as tensões totais (hipótese linear) verifique o critério de plastificação
MASKED	if elemento plastificou marque o elemento PLÁSTICO inclua o elemento na lista PLASTIFICADOS endif
	end loop 1.
INDIRECT ADDRESSING	2. loop sobre a lista de PLASTIFICADOS determine o número de subincrementos calcule a parcela elástica de deformação calcule tensões para o início da plastific. end loop 2.
VECTOR	3. recupere o no. máximo de subincrementos de deformação para a correção das tensões
INDIRECT ADDRESSING	4. loop sobre a lista de PLASTIFICADOS calcule o subincremento de deformação end loop 4.
OUTTER LOOP VECT.	5. loop sobre a lista de elementos PLASTIFICADOS 6. loop $i = 1$ até no. máximo de subincrementos
INDIRECT ADDRESSING	avalie a matriz elasto-plástica calcule o subincremento de tensão correspondente ao subincremento de deformação acumule o subincremento de tensão ao total atualize a superfície de escoamento (hardening) end loop 6. end loop 5.
VECTOR	7. loop sobre todos os elementos avalie a contribuição referente às cargas nodais equivalentes ao estado de tensões end loop 7.

Figura VI.4: Relação Constitutiva Não-Linear – Algoritmo Vetorizável

Capítulo VII

Transferência de Dados Global-Local

VII.1 Montagem de Vetores Locais e Globais

Um fato marcante em implementações convencionais do MEF é a freqüente utilização de instruções de endereçamento indireto. Principalmente nas etapas de transição entre os processos representados pelas equações IV.1 - IV.3 e IV.4 - IV.5, onde dá-se a transferência de informações das estruturas de dados globais do problema para as locais aos elementos e vice-versa.

Em implementações vetoriais/paralelas torna-se mais conveniente *replicar* as informações em estruturas de dados locais. A reprodução – replicagem – de informações contidas em uma estrutura de dados global em estruturas de dados locais atende a dois objetivos fundamentais.

Em primeiro lugar, o tratamento do problema localmente, ao nível dos elementos, é facilitado à luz dos fundamentos do MEF.

Por outro lado, a reprodução de informações em estruturas de dados locais viabiliza operações ditas *elemento-por-elemento* que, por sua natureza, apresentam um grande potencial para a sua implementação vetorial/paralela.

As etapas de transição local-global das informações constituem dois segmentos importantes da implementação e devem ser cuidadosamente tratadas através da eliminação dos potenciais focos de inibição à vetorização e paralelização. Tais etapas do processo são mais conhecidas no jargão, respectivamente, por localização (*disassembling*) e globalização (*assembling*).

O *disassembling* é a transferência de informações efetuada do sistema global de referência para o sistema local dos elementos. Tal processo pode ser realizado através de um mapeamento simples entre variáveis globais e locais, cuja implementação computacional pode ser dada pelo trecho de programa a seguir:

```
DO J = 1, ND
  DO N = 1, NEL
     VE (N,J) = V ( LM(N,J) )
  ENDDO
ENDDO
```

onde ND é o número de graus de liberdade por elemento finito, NEL é o número total de elementos, VE é o arranjo de grandezas locais ao elemento, V é o correspondente arranjo de grandezas globais e LM (*Localization Matrix*) é o arranjo que efetua o mapeamento entre grandezas locais e globais.

Do ponto de vista da vetorização desse processo, tal construção não apresenta nenhum empecilho (inibidor), sendo adequadamente tratada pelos compiladores que geram código vetorial.

Dessa forma, processos ditos *elemento-por-elemento* (*EBE - element-by-element*) dispensarão a utilização de instruções de endereçamento indireto, pouco eficientes. Valendo-se de uma maior disponibilidade de memória no sistema, a implementação de estruturas de dados locais constitui um passo importante no sentido de assegurar a independência de dados, viabilizando assim a vetorização e, principalmente, a paralelização dos processos.

Vale notar que na construção apresentada anteriormente assume-se que a discretização emprega um único tipo de elemento sobre todo o domínio do problema. De outra forma, com relação à viabilidade da vetorização, a implementação dessa etapa exigiria uma ligeira modificação na sua estrutura. Por exemplo, os elementos finitos poderiam ser agrupados continuamente segundo o seu tipo.

Por outro lado, a operação inversa, o *assembling*, exige um tratamento especial. Uma vez calculadas as contribuições locais dos elementos, estas devem ser montadas em um vetor global, como é o caso do vetor P^n no segundo membro da equação (1). Tal processo de montagem de um vetor global, a partir das contribuições locais dos elementos, pode ser dada simbolicamente por:

$$P^n = \sum_{n=1}^{NEL} p^n \qquad \text{(VII.1)}$$

onde o somatório deve ser interpretado como uma operação *booleana* sobre os NEL elementos empregados na discretização. Em implementações escalares convencionais tal operação pode ser corretamente tratada, valendo-se dos mesmos arranjos definidos anteriormente, pela seguinte construção:

```
      DO ID = 1, ND
         DO IEL = 1, NEL
            V ( LM(IEL,ID) ) = V  ( LM(IEL,ID) )
     &                         + VE (NEL,ND)
         ENDDO
      ENDDO
```

Para uma implementação vetorial, observa-se uma potencial dependência recursiva, já que vários elementos podem contribuir para uma mesma posição dentro do arranjo V de grandezas globais. Isso constitui um obstáculo à vetorização automática do loop mais interno. Nesse caso, a utilização de uma diretiva de compilação, no sentido de ignorar a recorrência, sem um tratamento adequado à estrutura do arranjo VE, produziria um código vetorial errático.

A Figura VII.1 apresenta o diagrama de acesso à memoria correspondente às duas operações envolvidas no processo.

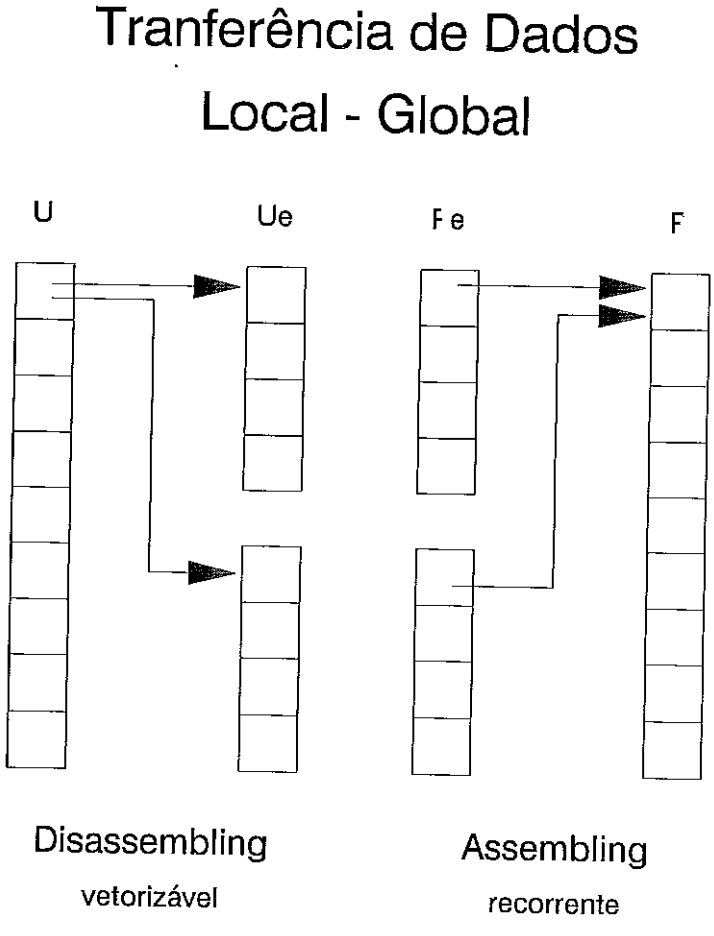


Figura VII.1: Transferência de dados local/global – relações entre os arranjos envolvidos nas etapas de assembling e disassembling (U, F arranjos globais; UE, FE arranjos locais)

VII.2 Técnica de Agrupamento dos Elementos Disjuntos

Para a vetorização do processo da montagem dos vetores globais, uma alternativa conduz a uma ligeira modificação na estrutura de dados de um programa convencional. O foco de inibição à vetorização pode ser contornado através de uma estratégia de reordenação dos elementos segundo blocos de elementos disjuntos [Far87]. Esse processo de agrupamento dos elementos é também conhecido como *colorir* a malha (*mesh coloring*).

O referido agrupamento implica em subdividir a discretização em tantos blocos (ou grupos) quantos forem necessários, de forma que, no interior de cada um, nenhum elemento compartilhe um mesmo ponto nodal. A cada grupo é atribuído um identificador (uma *cor*). A visualização desse processo é apresentada na Fig VII.2 para uma malha plana regular.

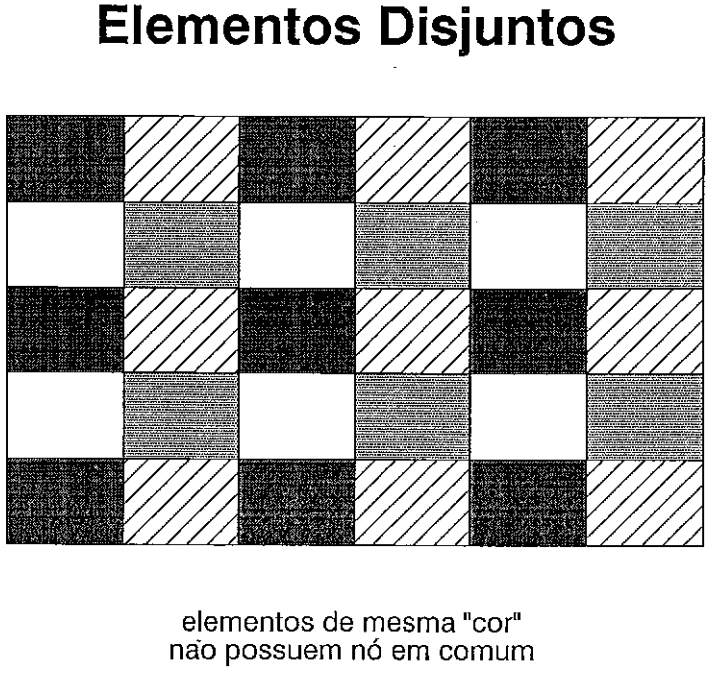


Figura VII.2: "Mesh Coloring" – aplicação da técnica a uma discretização bidimensional regular

Pode-se observar que os elementos de um mesmo grupo não compartilham um mesmo ponto nodal, são disjuntos. Conseqüentemente, não compartilham uma mesma equação. Assim sendo não há recorrência ao se processar a montagem das contribuições dos elementos de um mesmo grupo.

Tal estratégia pode ainda apresentar algumas variantes tais como a imposição de limitações à dimensão individual dos blocos de forma a melhor acomodar a estratégia à alguma característica do *hardware*.

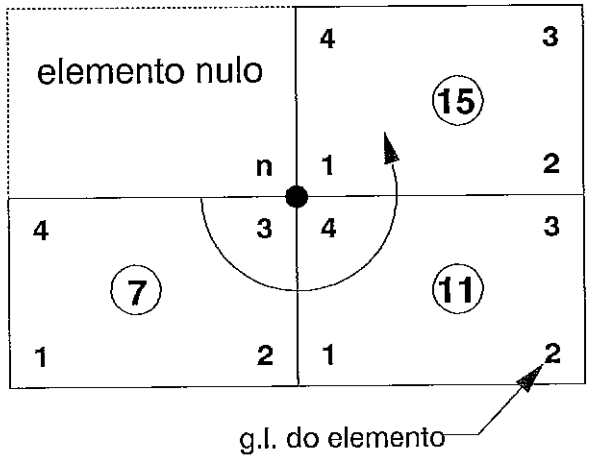
Assim, o processo da montagem dos vetores globais poderia ser dado conforme a seguinte construção:

```
      DO IBLOCK = 1, NUMERO_DE_BLOCOS
           NEL1 = PRIMEIRO_ELEMENTO_DO_BLOCO ( IBLOCK )
           NEL2 =   ULTIMO_ELEMENTO_DO_BLOCO ( IBLOCK )
           DO ID = 1, ND
C$DIR IGNORE RECRDEPS V
           DO IEL = NEL1, NEL2
              V ( LM(IEL,ID) ) = V ( LM(IEL,ID) )
     &                          + VE (NEL,ND)
           ENDDO
           ENDDO
      ENDDO
```

Deve-se observar-se o emprego de uma diretiva de compilação, pois a eliminação da recorrência nesse caso não é *visível* para o compilador.

VII.3 Mapeamento Inverso

Uma outra forma de contornar o problema emprega a técnica de mapeamento inverso, aparentemente introduzida por [Tgk89]. Para isso define-se uma tabela de conetividade das equações (*equation connectivity array*). Uma limitação desse enfoque é a de que ele assume o domínio discretizado através de uma malha regular de elementos finitos. São empregados dois arranjos, KLI e LIJ, que apontam respectivamente para o elemento topologicamente conectado a uma determinada equação global e para a equação local correspondente. Na Figura VII.3 pode ser visualisada a definição dos referidos arranjos, para o caso de uma discretização regular empregando elementos quadriláteros bi-lineares. Nesse caso, o número



kli (n,1) = 7 lij (n,1) = 3

kli (n,2) = 11 lij (n,2) = 4

kli (n,3) = 15 lij (n,3) = 1

kli (n,4) = 0 lij (n,4) = 1

Figura VII.3: Mapeamento Inverso – tabela de Conetividade das Equações para o caso bidimensional

máximo de elementos conectados a um mesmo nó é quatro, e a operação de *assembling* é dada por:

```
DO I = 1,  NEQ
   V(I) =  VE ( KLI(I,1) , LIJ(I,1) )
.        +  VE ( KLI(I,2) , LIJ(I,2) )
.        +  VE ( KLI(I,3) , LIJ(I,3) )
.        +  VE ( KLI(I,4) , LIJ(I,4) )
ENDDO
```

onde NEQ é o número de equações presentes na discretização. A construção acima é plenamente vetorizável.

Na implementação aqui referida optou-se pela técnica de mapeamento inverso, que apesar da penalização em termos de memória, é insensível às dimensões da discretização. Um fator também decisivo nessa opção foi a performance observada empregando-se a técnica de mapeamento inverso em comparação à estratégia de blocagem dos elementos em blocos disjuntos. Em um contexto similar [Cou89], a montagem dos vetores globais via mapeamento inverso mostrou-se mais eficiente.

Capítulo VIII

Paralelização do Código

A implementação de esquemas de paralelização para programas no IBM 3090 devem levar em consideração as características apresentadas por esta máquina. A arquitetura do seu hardware, os detalhes de gerenciamento de tarefas oferecidos pelo sistema operacional MVS e as primitivas de paralelismo do MTF no VS-FORTRAN e do PARALLEL FORTRAN recomendam implementações de um paralelismo de baixa granularidade (*coarse grain*).

Nele, as tarefas do programa que apresentam independência de dados entre si, portanto candidatas à execução concorrente nas várias CPU's disponíveis, devem representar, individualmente, uma cota razoável de esforço computacional (tempo de CPU). Isto é necessário de forma a compensar o tempo adicional de processamento exigido pelo sistema operacional para as operações de criação, disparo, sincronização e término das tarefas concorrentes (*task management*).

Quando estas tarefas consomem um tempo comparável ao tempo extra exigido do MVS para controlá-las (*overhead*), pode-se não ter ganhos em tempo de execução com o paralelismo. Podem até mesmo serem observadas perdas, em relação à execução serial do código, devidas à penalização inerente ao gerenciamento destas tarefas.

De acordo com a intenção original da implementação, foi elaborado o paralelismo a nível do processo de avaliação das forças internas, por ponto de integração (ponto de Gauss).

Considerando o emprego de elementos isoparamétricos bilineares, a integração numérica pelo processo de Quadratura de Gauss requer 4 pontos de integração por elemento (integração *full-rank*). Como visto anteriormente, a avaliação das grandezas relativas a cada ponto de Gauss é independente dos demais pontos de integração, caracterizando-se assim uma fonte natural de paralelismo facilmente identificável e, do ponto de vista da estrutura de dados, muito bem *localizada* .

Dessa forma, podem ser criadas 4 tarefas concorrentes, cada uma delas responsável pela determinação da contribuição de um ponto de Gauss para o processo de integração, ao longo de todos elementos da discretização. Os dados referentes aos deslocamentos nodais de cada elemento são compartilhados globalmente entre as tarefas. Áreas de memória de acesso restrito a cada tarefa, são definidas de forma a acomodar as variáveis envolvidas, correspondentes a cada ponto de Gauss. Nessas áreas estão alocadas:

- os operadores diferenciais discretos;
- as componentes de deformação,
- as componentes de tensão e
- as parcelas da contribuição às forças internas

Essas áreas *locais* de dados estão organizadas em arranjos no sentido dos elementos, permitindo assim a vetorização da tarefa. A Figura VIII.1 ilustra a esquematicamente a organização adotada para os principais arranjos envolvidos no processo.

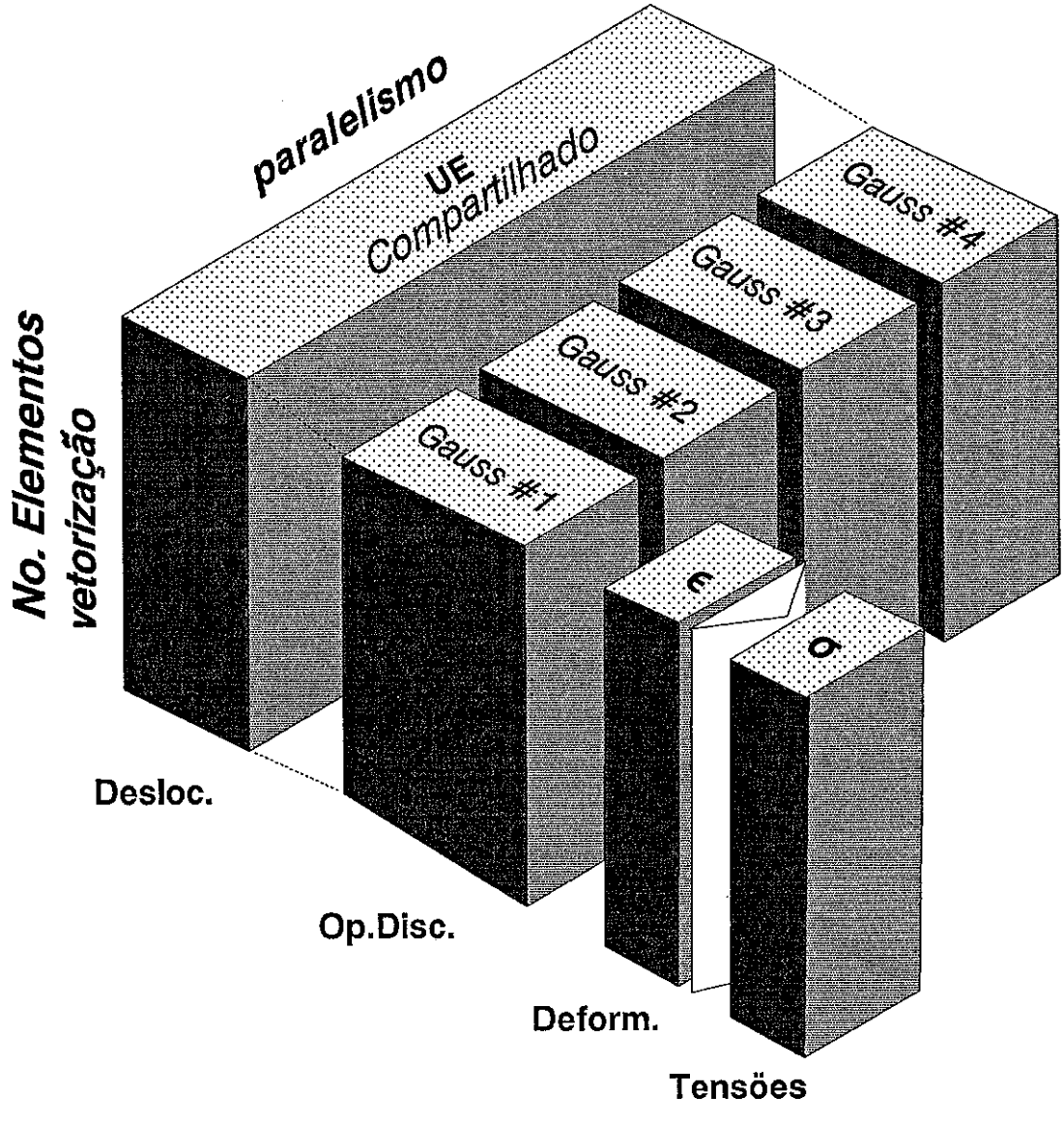


Figura VIII.1: Organização de dados empregada para vetorização e paralelização do cálculo das forças internas

A estratégia empregada possui ainda uma outra vantagem no tocante à distribuição da carga computacional (*load balancing*) entre as tarefas. Todas as tarefas processam operações praticamente idênticas sobre um mesmo número de elementos. A distinção ocorre nos casos não-lineares, quando a correção do estado elasto-plástico nem sempre se processa de maneira uniforme, causando alguma retenção para o perfeito sincronismo das tarefas. Ainda assim, o esquema proposto tende a minimizar esse problema.

O paralelismo foi implementado valendo-se das primitivas para o *schedulle* e sincronização de tarefas (*CALL DSPTCH e CALL SYNCRO*) disponíveis na extensão MTF do VS-FORTRAN. A Figura VIII.2 ilustra em diagrama de blocos o modo de processamento tradicionalmente empregado (SERIAL) em comparação à estratégia de processamento concorrente adotada (PARALELA).

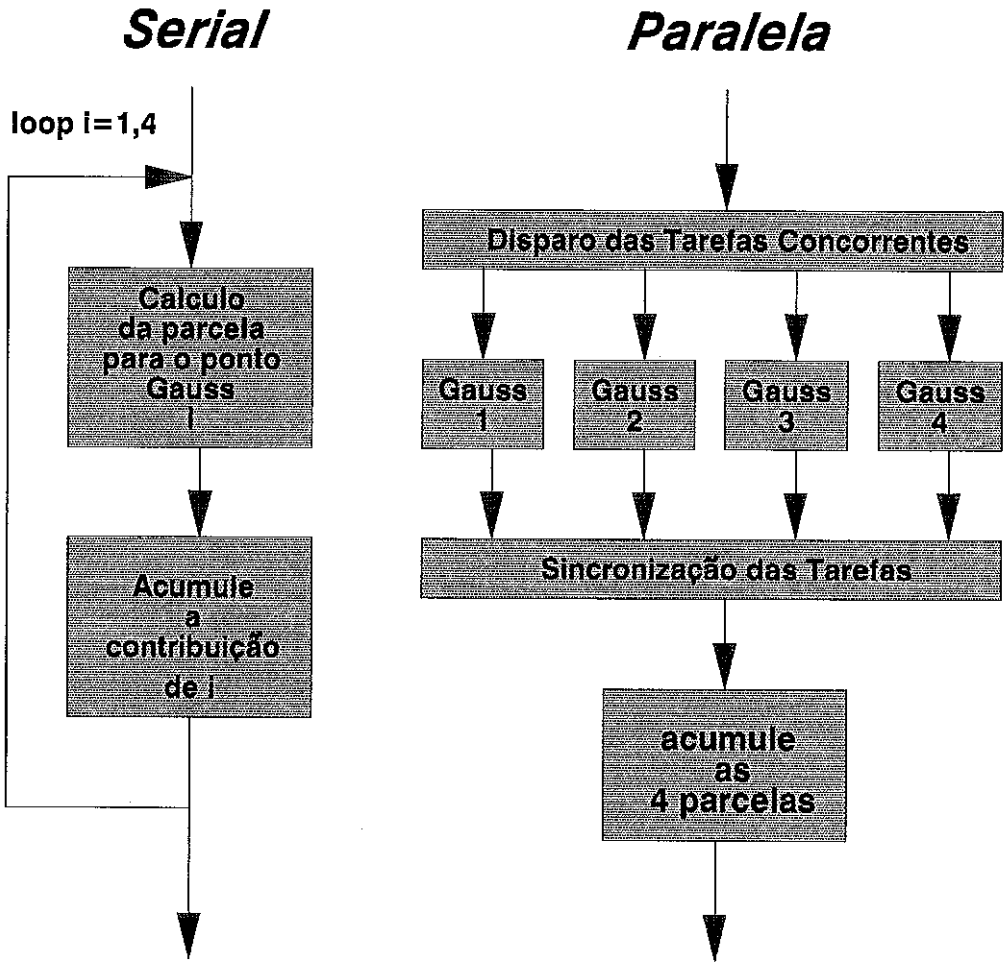


Figura VIII.2: Implementação do paralelismo para a avaliação das forças internas

Capítulo IX

Aplicações Numéricas

IX.1 Simulação Geofísica - Geração de Dados Sintéticos para VSP

O primeiro caso analisado aborda um exemplo simples de propagação de ondas acústicas em domínio bi-dimensional. Um maciço de solo estratificado é modelado para a síntese numérica de dados sísmicos para VSP (*vertical seismic profiling*), uma aplicação típica no âmbito da Geofísica. Neste tipo de análise, a propagação, através das diversas camadas do sub-solo das ondas de choque geradas pela detonação de cargas explosivas (*tiros*) na superfície, é medida e interpretada visando a obtenção de dados geológicos. O modelo é acústico e portanto a análise de propagação de ondas elásticas lineares é adequada ao problema.

O modelo analisado considera um maciço formado por três camadas horizontais, cada uma apresentando diferentes velocidades de propagação para a onda dilatacional, constituindo-se num caso favorável ao emprego do operador particionado. A descrição do problema é dada na Figura IX.1 e na Figura IX.2 pode ser observada a função de carga fornecida para modelar a ação da carga explosiva.

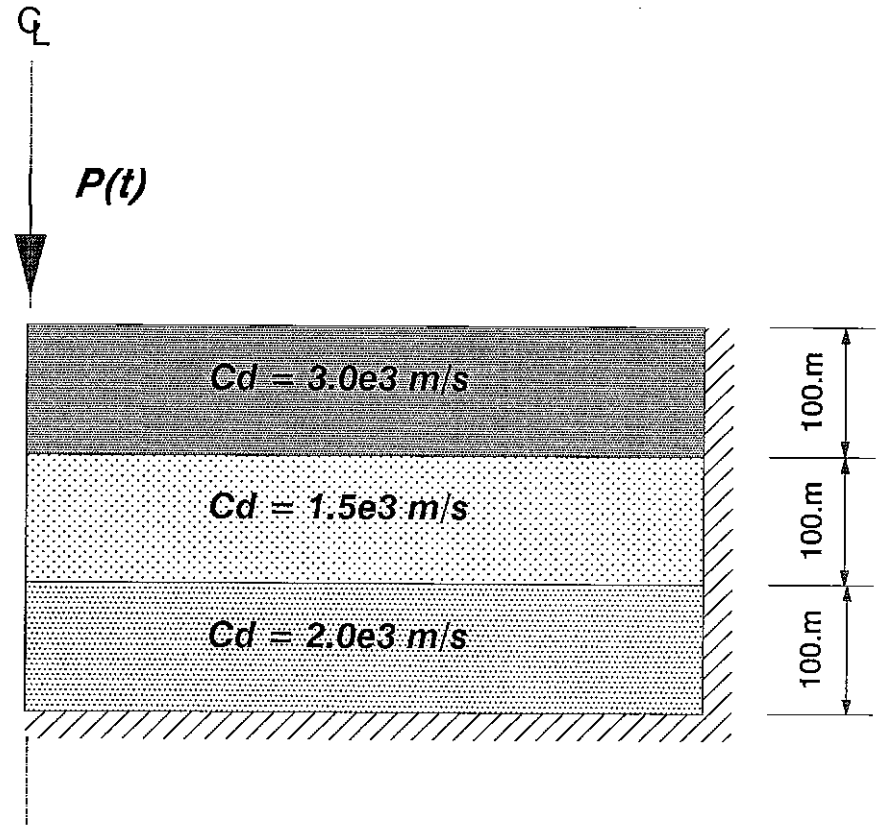


Figura IX.1: Simulação Geofísica - Definição do problema

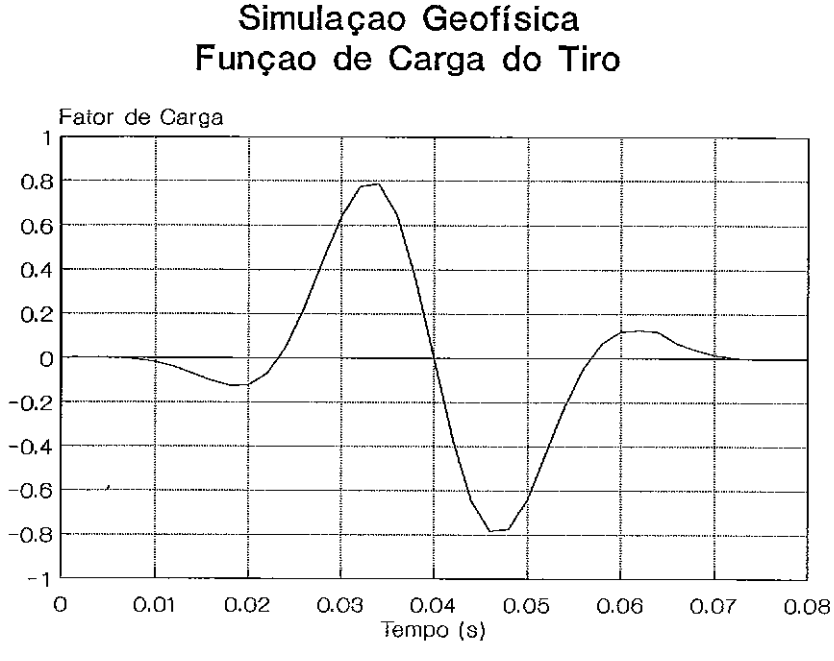


Figura IX.2: Simulação Geofísica - Função de carga do tiro

Foram empregadas malhas uniformes, compostas por elementos isoparamétricos bilineares para sólido axissimétrico. A dimensão dos elementos foi mantida fixa, cada elemento representando um quadrado de 5 m de lado. O número de elementos empregado na direção da profundidade foi mantido fixo, variando-se a extensão horizontal da discretização, de forma que o número de elementos por partição variasse linearmente. Cada camada é convenientemente associada a uma partição com intervalo de tempo ditado pela equação IV.6. Na análise convencional o intervalo de tempo para a solução estável foi de $1. \times 10^{-4} s$. Para a análise particionada foram empregados os seguintes intervalos de tempo $1. \times 10^{-4} s$, $2. \times 10^{-4} s$ e $4. \times 10^{-4} s$. Em todos os casos foram considerados 700 intervalos de tempo.

A Tabela IX.1 apresenta os tempos de processador (CPU) obtidos para as várias discretizações empregadas, tanto para o operador convencional quanto para o particionado. Na coluna *Speed-Up* apresentam-se os ganhos obtidos através da aplicação do algoritmo particionado, em termos do tempo total de processamento (CPU).

Como pode-se observar, para esse problema, a performance da análise particionada é em média 1.40 vêzes superior, em tempo de CPU, à análise convencional. É interessante observar o conteúdo vetorial em ambos os casos. O algoritmo convencional alcança 85% de utilização de CPU vetorial, ao passo que na análise particionada este índice é ligeiramente inferior devido ao processo de determinação da *profundidade das pilhas*, o qual é eminentemente escalar.

Simulação Geofísica						
		s/ Partição		Partição 1:2:4		Speed
Malha	Neq	total	vetorial	total	vetorial	up
80 x 60	9600	32	27	22	18	1.45
160 x 60	19200	63	54	46	39	1.36
320 x 60	38400	127	109	92	78	1.38
650 x 60	78000	257	222	187	157	1.37

Tabela IX.1: Tempos de CPU para Simulação Geofísica

Tomando a discretização 80×60 e extendendo a análise para 2000 intervalos de tempo, os instantâneos (*snapshots*) de velocidade resultante ($\sqrt{v_x^2 + v_y^2}$) para $t = 100.ms$ e $t = 200.ms$ são apresentados, respectivamente, nas Figura IX.3 e Figura IX.4. Vale notar que os resultados para as análises convencional e particionada apresentaram excelente concordância.

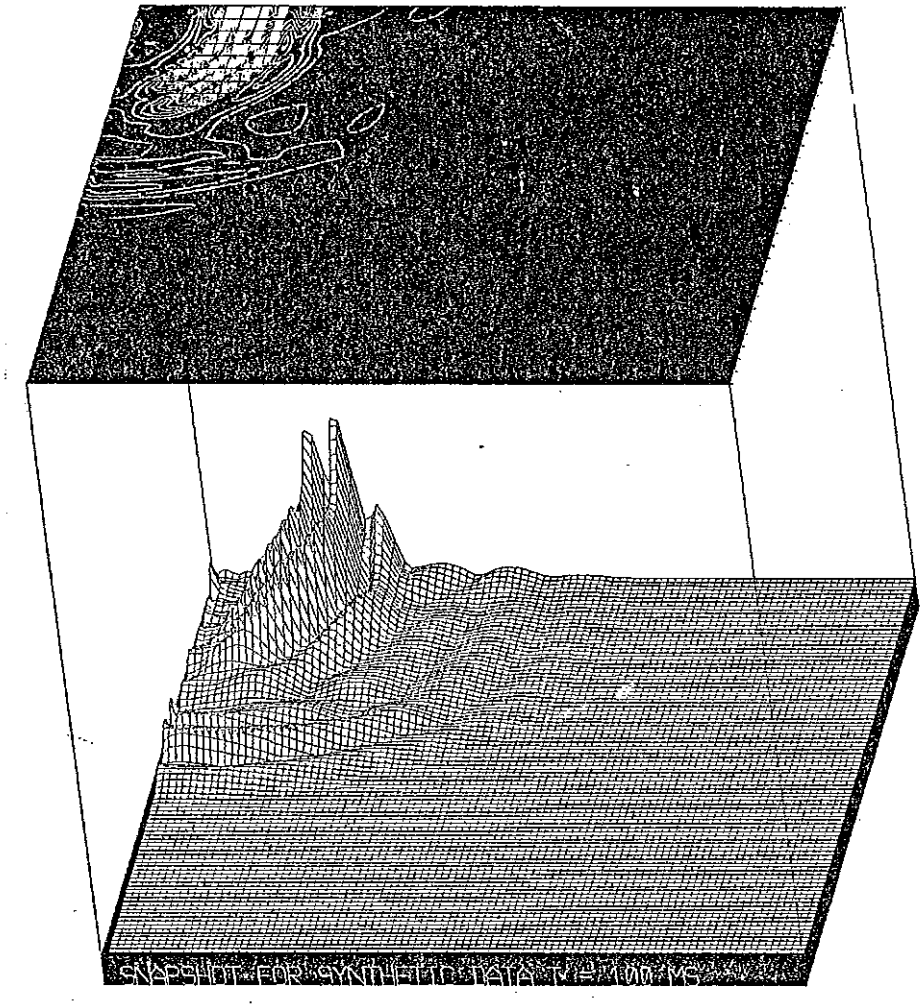


Figura IX.3: Simulação Geofísica – Instantâneo das velocidades resultantes para t=100 ms

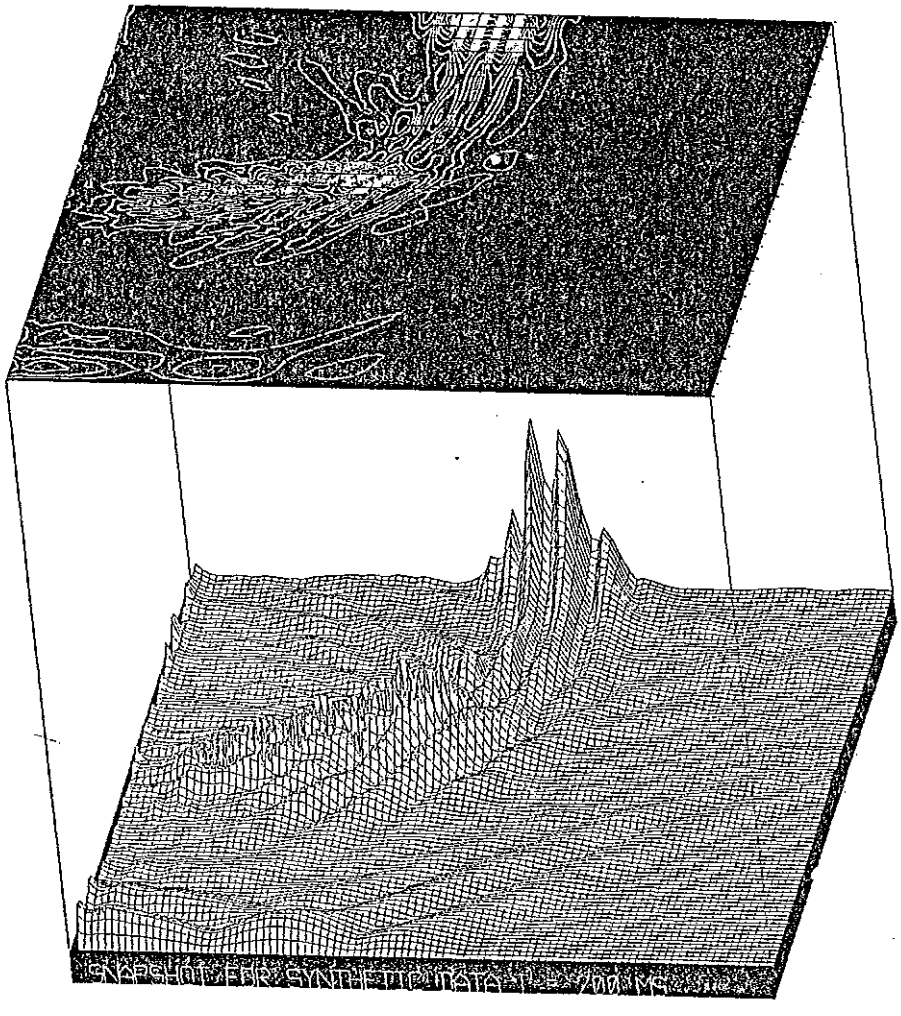


Figura IX.4: Simulação Geofísica – Instantâneo das velocidades resultantes para t=200 ms

IX.2 Calota Esférica Submetida à Pressão Externa

O segundo problema analisado refere-se à resposta estrutural de uma calota esférica submetida a uma pressão externa uniforme subitamente aplicada [Nag75] e [Sha76]. A definição do problema é dada na Figura IX.5 . São considerados grandes deslocamentos e o comportamento elasto-plástico do material.

Para a verificação do desempenho da versão paralelizada do algoritmo de integração das forças internas foi empregada uma discretização composta por 1000 elementos (10 ao longo da espessura, 100 na outra direção), compreendendo 1111 pontos nodais e totalizando 2189 equações.

Na Tabela IX.2 estão sumarizados os tempos obtidos nesta análise, bem como o ganho observado através do índice de ganho em desempenho (*Speed-Up*) medido pela relação entre os tempos de execução (*elapsed*) em modo vetorial/paralelo (*Vector-MTF*) e vetorial/seqüencial (*Vector*).

CALOTA ESFÉRICA				
Análise	CPU (s)		Elapsed (s)	Speed up
	total	vetorial		
Vector	565	303	591	—
Vector-MTF	606	331	236	2.50

Tabela IX.2: Tempos de CPU para Calota Esférica.

A análise VECTOR/MTF empregou quatro processadores, cada um associado a um ponto de Gauss.

A distribuição dos tempos de CPU entre as várias fases envolvidas na solução deste problema é apresentada na Figura IX.6, onde pode-se constatar a responsabilidade do processo de avaliação das forças internas (atualização dos elementos) no custo total da análise.

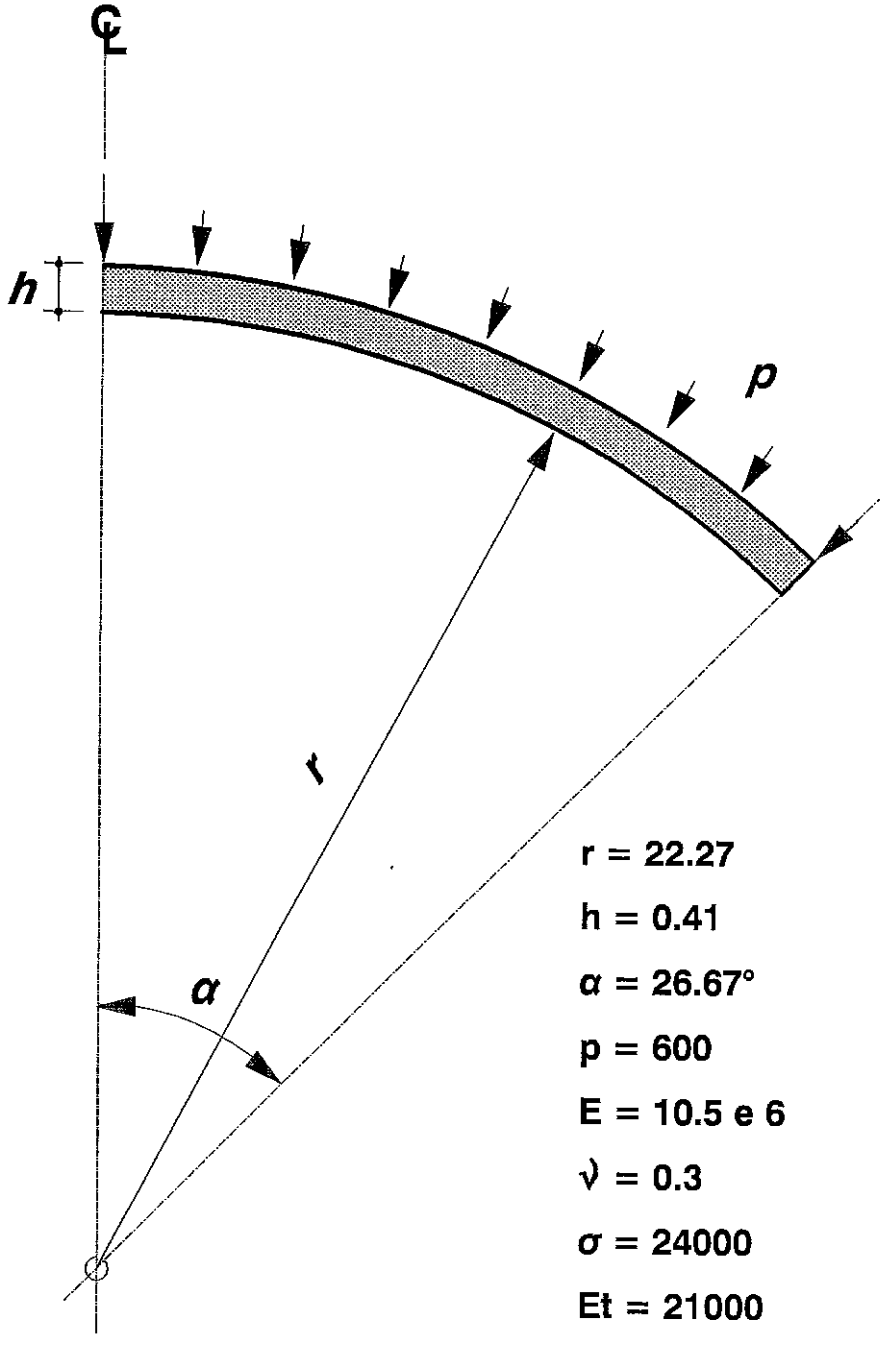


Figura IX.5: Calota Esférica - definição do problema

A Figura IX.7 apresenta uma seqüência de configurações deformadas para a discretização empregada, e na Figura IX.9 pode-se observar a comparação entre as histórias do deslocamento vertical do ápice, para os cinco modelos implementados, a saber:

- Linear físico e geométrico
- Elasto-plástico perfeito, pequenos deslocamentos
- Elasto-plástico com endurecimento isotrópico, peq. deslocamentos
- Elasto-plástico, grandes deslocamentos (TLF)
- Elasto-plástico, grandes deslocamentos e grandes deformações (ULJ)

Os resultados podem ainda ser comparados às curvas análogas, dadas na referência [Nag75], e reproduzidas na Figura IX.10 .

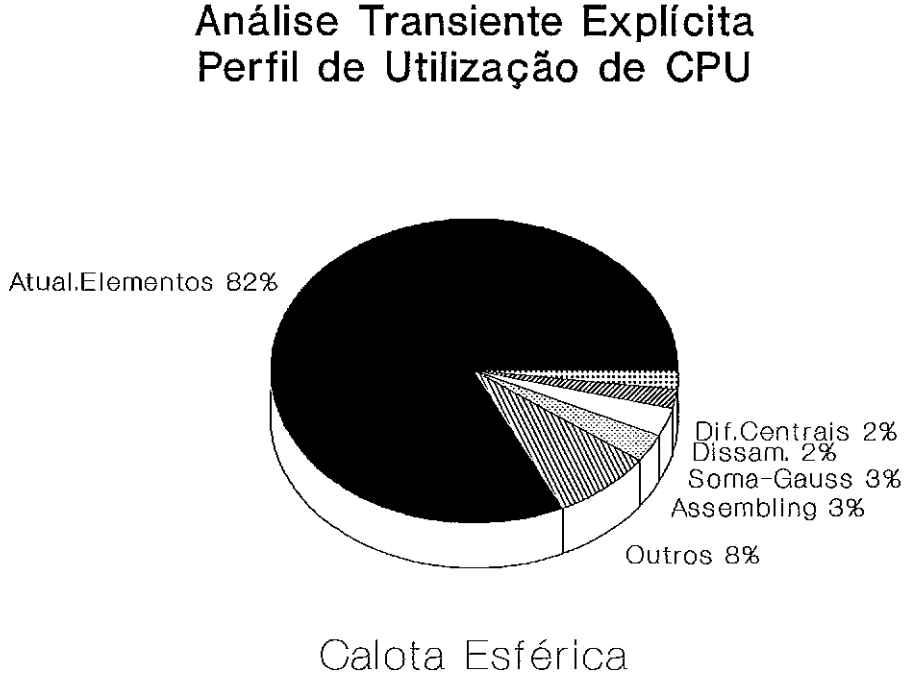


Figura IX.6: Perfil de Utilização de CPU - cada setor representa o consumo percentual de tempo de processador para cada etapa básica da solução do problema da calota esférica

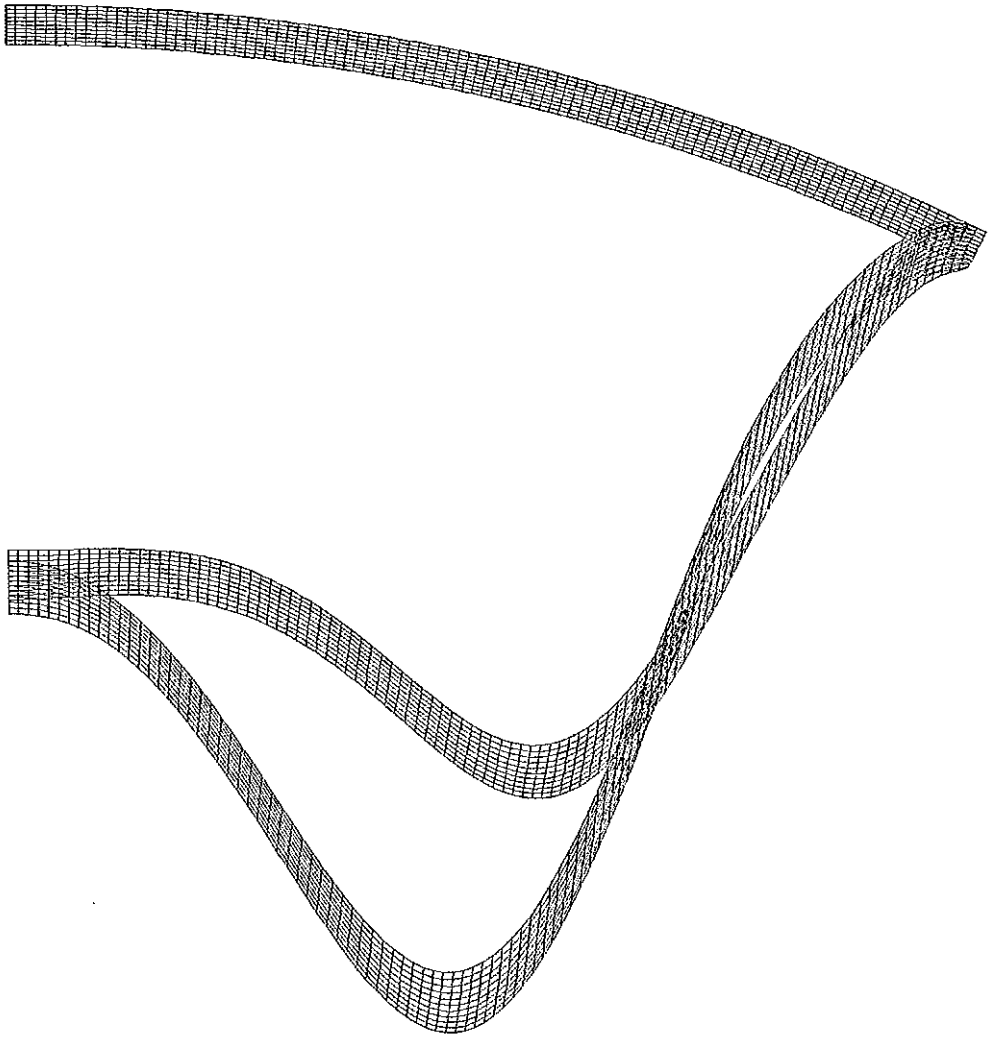


Figura IX.7: Calota Esférica - Deformadas para malha com 1000 elementos

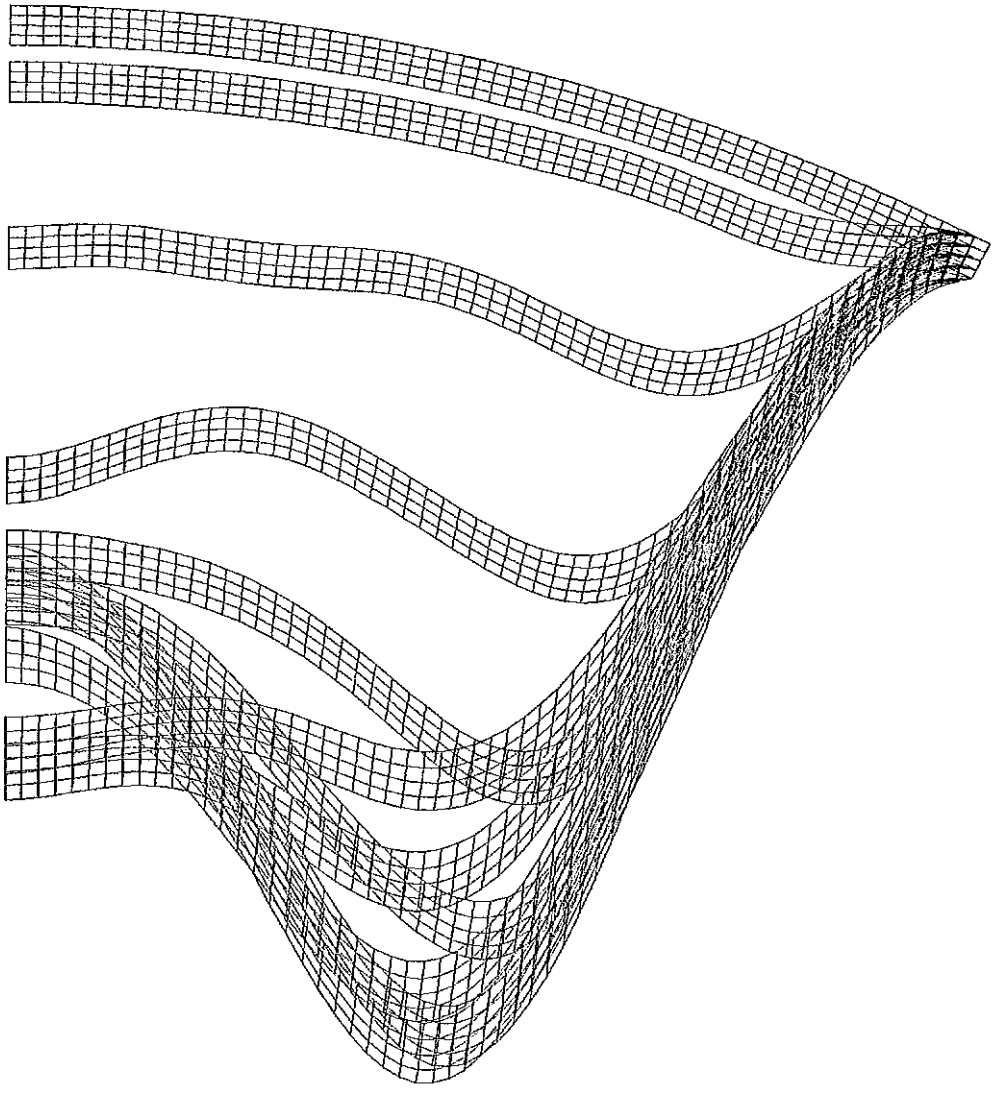


Figura IX.8: Calota Esférica - Deformadas para malha com 240 elementos

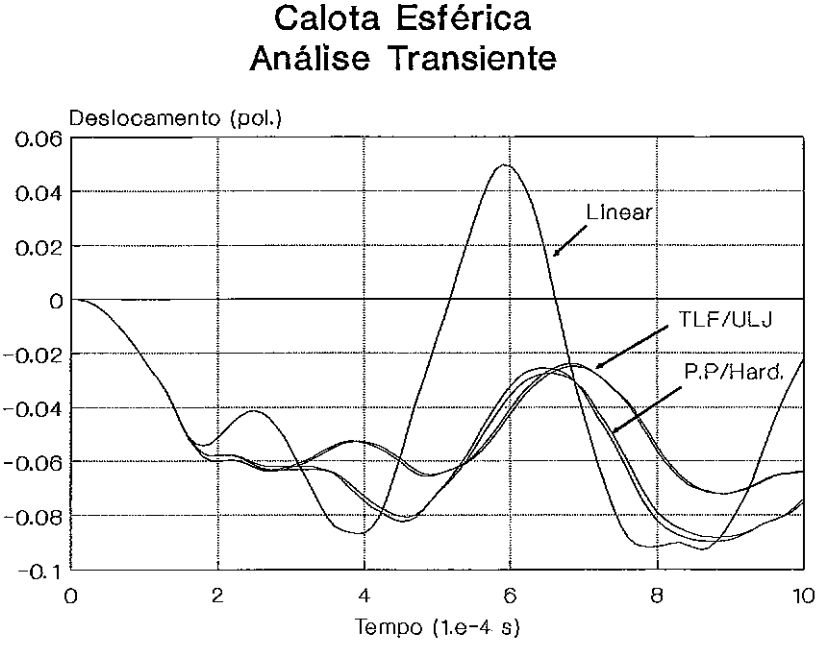


Figura IX.9: Calota Esférica - Deslocamentos do ápice (presente trabalho)

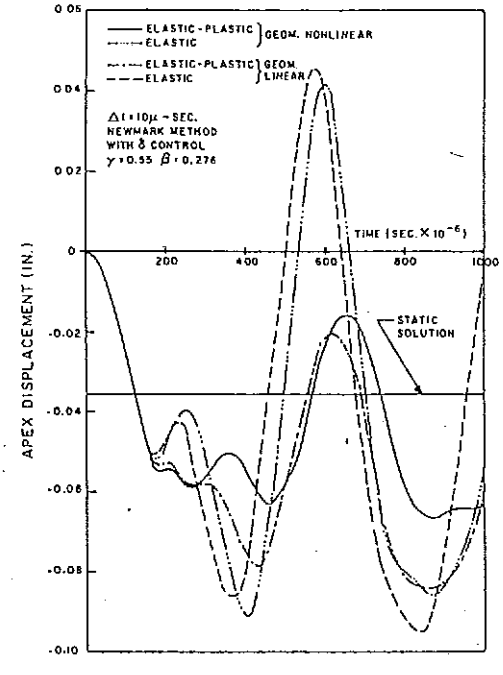


Figura IX.10: Calota Esférica - Deslocamentos do ápice (segundo [Nag75])

IX.3 Calota Esférica com Partição Rígida

Sobre uma variante do problema anterior foi efetuada uma análise particionada. Nessa análise foi considerada uma partição rígido-elástica mais densa, através de toda a espessura da casca, abrangendo um quinto de sua extensão total a partir do ápice.

As propriedades físicas desse coroamento são dadas a seguir:

módulo de elasticidade	$E = 0.105 \times 10^{9}$
coeficiente de Poisson	$\nu = 0.3$
densidade	$\rho = 0.780 \times 10^{-3}$
tensão de escoamento	$\sigma_y = 0.5 \times 10^{5}$

Empregou-se uma discretização menor — 160 elementos (40×4), 205 nós e 395 equações — para a observação do comportamento do algoritmo particionado em um problema não-linear. Para efeito de comparação com [Owe80], considerou-se apenas o comportamento elasto-plástico perfeito sob pequenas deformações e pequenos deslocamentos.

A Tabela IX.3 apresenta a comparação entre o desempenho da análise convencional e particionada, podendo observar-se, como esperado, um desempenho melhor da análise particionada para esse caso.

Calota Esférica com Partição Rígida				
Análise	CPU (s)		Porcentagem vetorial	Speed up
	total	vetorial		
Convencional	84.23	68.41	81.22	1.00
Particionada	57.35	44.18	77.04	1.47

Tabela IX.3: Calota Esférica com Partição Rígida – comparação do desempenho entre as análises convencional e particionada

Na Figura IX.11 é apresentada a história dos deslocamentos verticais do ápice, tanto para a análise particionada quanto para a análise convencional, ambas em perfeita concordância. Na Figura IX.12 pode-se observar a seqüência de configurações deformadas onde pode também ser notada a presença da partição rígida e seu comportamento ao longo do tempo.

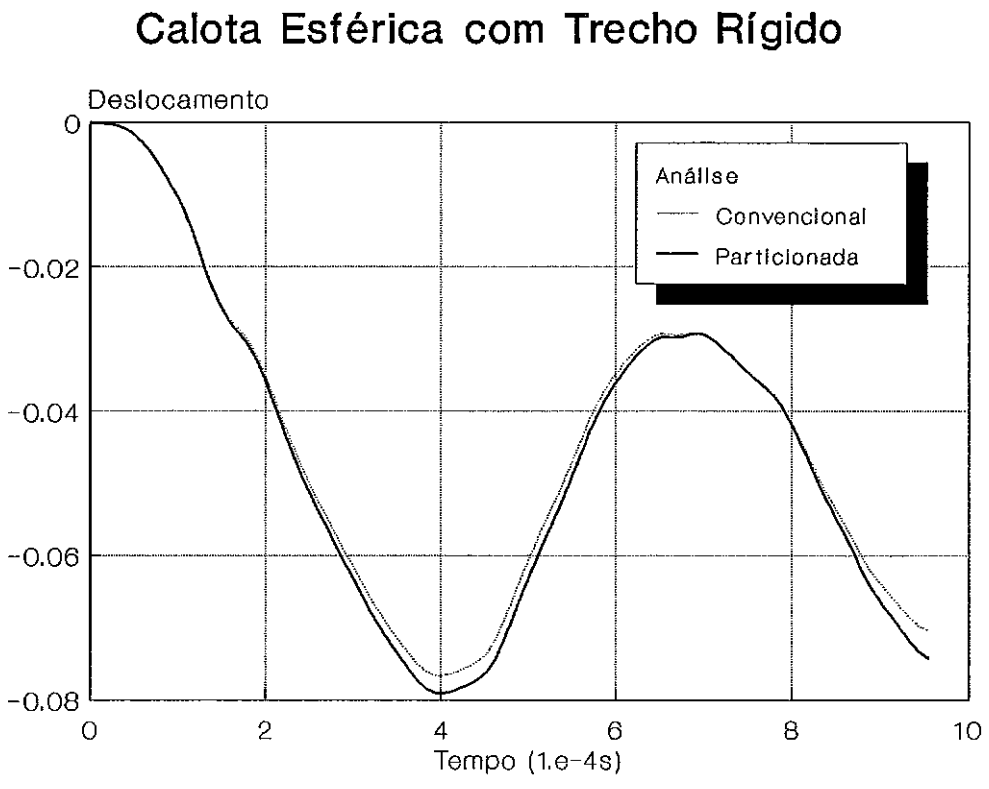


Figura IX.11: Calota Esférica – Deslocamentos do ápice para análise particionada não-linear

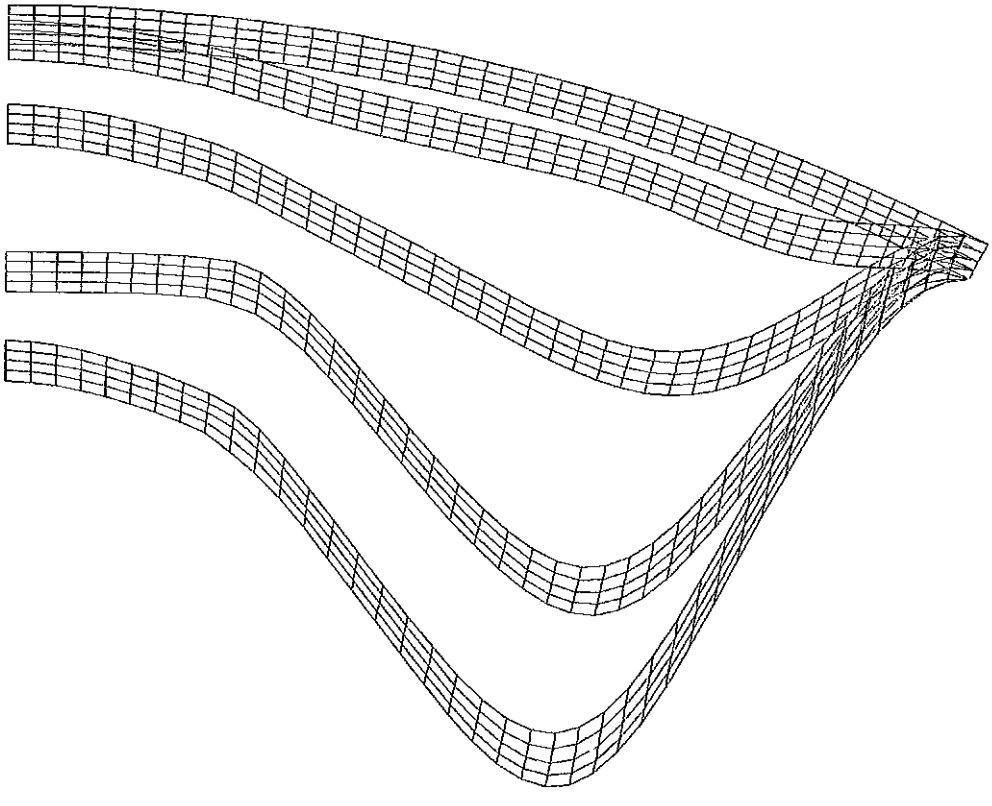


Figura IX.12: Calota Esférica – Deformadas (Análise Particionada - 160 elementos)

IX.4 Análise de Impacto - Barra de Taylor

Essa análise modela o impacto de uma barra de cobre contra um anteparo rígido, a 227 m/s. Experimentos assim conduzidos em laboratório são uma forma usualmente empregada para a determinação das equações de estado a altas pressões. Uma explanação mais detalhada dos fenômenos observáveis nesse problema pode ser encontrada em [Joh72].

Dadas as elevadas taxas de deformação desenvolvidas na região do impacto, é de se esperar a manifestação de grandes deformações e rotações. Por essas razões o problema é freqüentemente abordado para a aferição da formulação e verificação do desempenho de implementações computacionais envolvendo deformações finitas [Kam90], [Fer89], [Tgk89].

A barra modelada tem 3.2mm de diâmetro e 32.4mm de comprimento. O modelo empregado nessa análise compreende 5 elementos na direção radial por 50 elementos na direção longitudinal, totalizando 250 elementos, 306 pontos nodais e 555 equações. A velocidade da barra no instante do impacto é de 277 m/s. As condições de contorno impostas impedem os deslocamentos da base inferior da barra, simulando a situação de contato e impenetrabilidade no anteparo.

O comportamento não-linear é tratado via formulação Lagrangeana atualizada com taxa de Jaumann (formulação ULJ). As características do material foram: módulo de elasticidade de 117.GPa, tensão de escoamento de 400.MPa, módulo tangente 100.MPa e densidade 3970.kg/m^3. A lei constitutiva considerada obedece ao critério de von Mises com endurecimento isotrópico.

A análise compreendeu 16000 intervalos de tempo de 5.0×10^{-9}s até o instante de tempo t = 80.0 ms, não sendo consideradas partições. A avaliação das forças internas foi realizada com integração reduzida e contrôle dos modos de energia nula, empregando um ponto de integração no centróide dos elementos. O parâmetro de contrôle viscoso/elástico dos modos de energia nula foi de 5% e o de amortecimento viscoso artificial foi 10%.

A Figura IX.13 apresenta o diagrama com a definição do problema.

A Figura IX.14 apresenta a deformada para o instante de tempo t = 80.0ms. No detalhe da Figura IX.15 pode-se observar o aspecto final da malha na região do impacto.

Vale notar que a mesma análise, empregando integração *full rank* na avaliação das forças internas, manifesta o *locking* dos elementos, conduzindo a uma solução final em completo desacordo com a realidade do problema.

Valores insuficientes tanto para o parâmetro de controle de *hourglass* quanto para o amortecimento artificial, conduzem a soluções não satisfatórias. Observa-se nesses casos a violação da condição de impenetrabilidade na superfície de impacto, causado pela distorção excessiva das camadas de elementos adjacentes.

A influência do amortecimento pode ser observada na Figura IX.16 onde são apresentados os sinais da velocidade no topo da barra, para o ponto sobre o eixo de simetria do problema, variando-se o controle de hourglass e o amortecimento artificial.

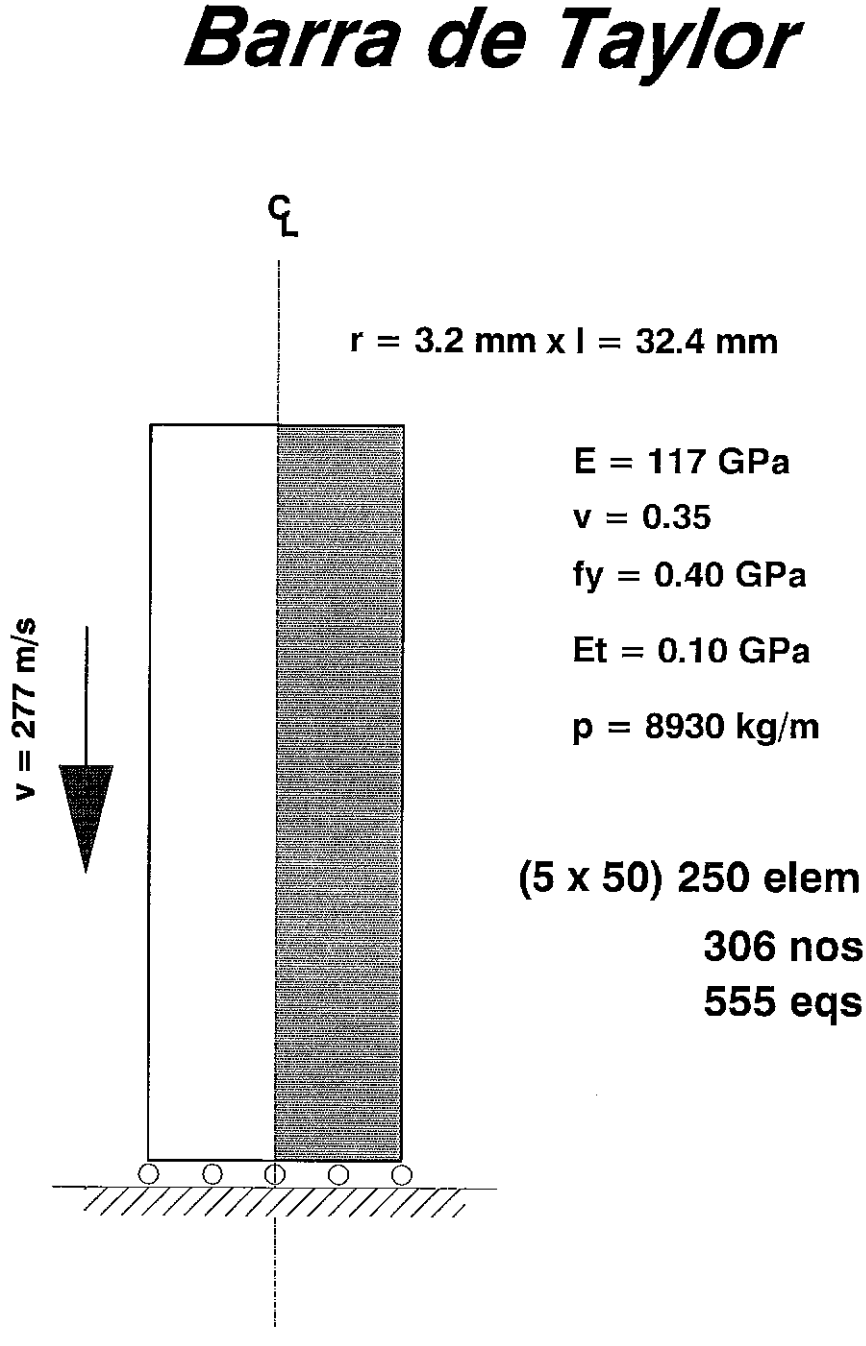


Figura IX.13: Barra de Taylor – Definição do problema

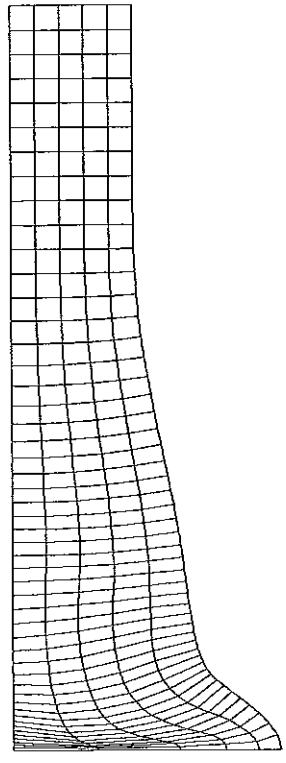


Figura IX.14: Barra de Taylor – Deformada Final (t = 80. ms)

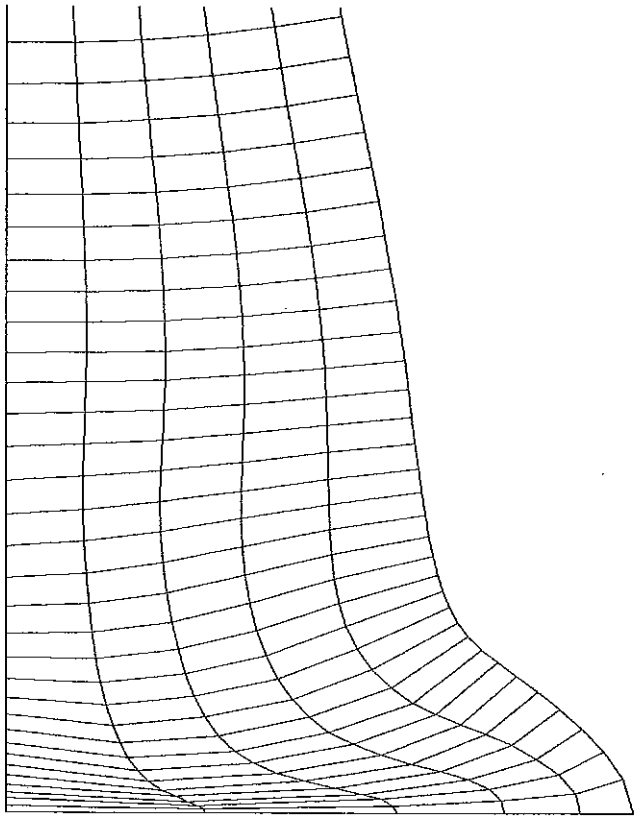


Figura IX.15: Barra de Taylor – Deformada Final (detalhe)

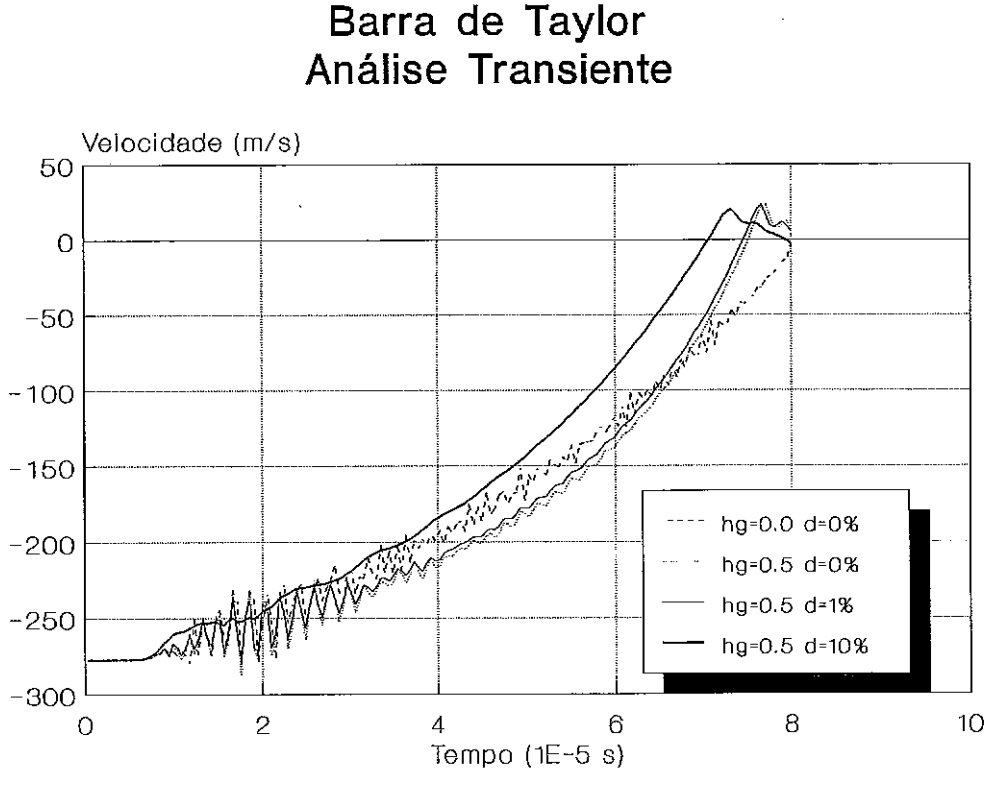


Figura IX.16: Barra de Taylor – Sinal de velocidade no topo

IX.5 Análise Estática - Capacidade de Carga de Fundação Rasa

Nesta seção, ilustraremos o comportamento da aplicação do algoritmo explícito como um método de relaxação simultânea para a solução iterativa de problemas estáticos não-lineares. Foram empregados os algoritmos de relaxação dinâmica e de relaxação viscosa acelerada.

O problema modelo adotado trata da determinação da capacidade de carga (carga limite) de uma fundação rasa, assentada sobre argila não-drenada (solo puramente coesivo).

Para a discretização do maciço, adotou-se um modelo axissimétrico cujas dimensões estão apresentadas na Figura IX.17. O modelo empregado compreeende 1200 elementos quadriláteros e 1271 pontos nodais perfazendo um total de 2355 equações. As caracterísicas físicas do solo são: módulo de elasticidade $E = 1.0 \times 10^5$ e coesão $c = 1.0 \times 10^3$, convenientemente convertida para a tensão de escoamento equivalente, para aplicação direta do critério de von Mises.

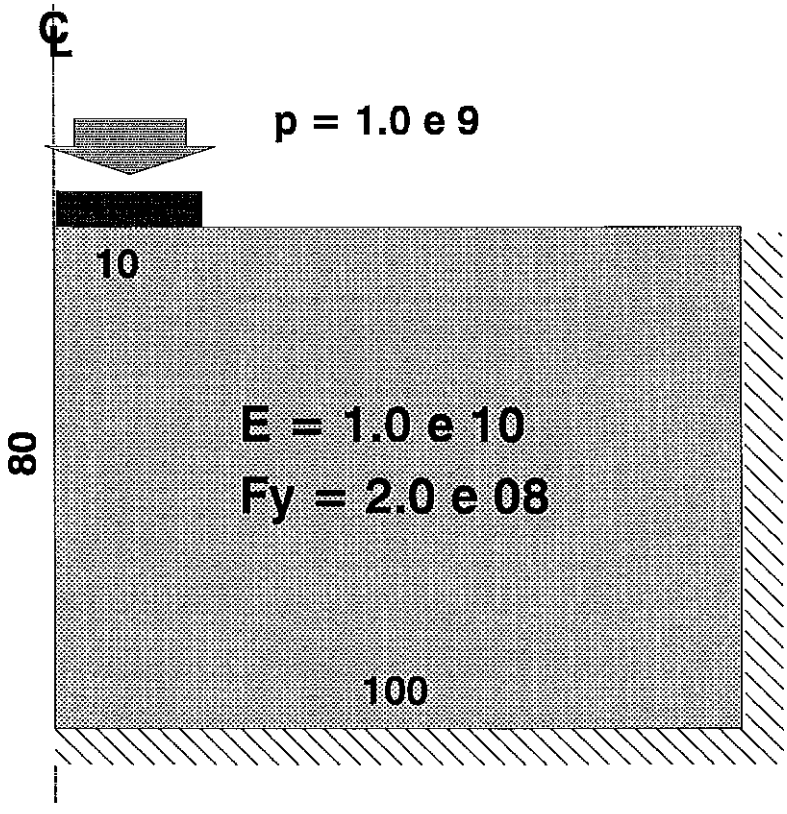


Figura IX.17: Análise limite de fundação rasa – definição do problema

A estrutura da fundação em si (a sapata), considerada como corpo rígido, não foi incluída no modelo. Ao invés, a sua interação com a camada de solo subjacente foi modelada através de restrições generalizadas para os deslocamentos verticais. Equações que descrevem restrições generalizadas, por sua própria natureza, não são vetorizáveis. Entretanto a influência desse fato no desempenho observado pode ser desprezada, dado o número reduzido de equações envolvidas (16) face ao número total de equações presente no modelo analisado (2355). Além disso, a inclusão de elementos com rigidez *infinita* para modelar a sapata, afetaria desfavoravelmente o condicionamento espectral do modelo, o que forçosamente se refletiria em um número de iterações inaceitável.

Para a integração das forças de volume nos elementos foram empregados os esquemas de integração *full-rank*, com quatro pontos de Gauss por elemento e de integração reduzida com controle de modos espúrios, para um único ponto por elemento. O parâmetro de controle viscoso/elástico de modos espúrios adotado foi de 0.08. A análise se desenvolveu para cinco incrementos de carga da ordem de 25, 50, 83.3, 91.7 e 100% da carga limite teórica ($p_u = 6c$). O critério de parada do processo iterativo foi estabelecido para uma tolerância da ordem de 1.0×10^{-3} na norma relativa dos resíduos.

Na Figura IX.18 podem ser observadas as curvas carga por deslocamento vertical para ambas as ordens de integração das forças internas. Vale notar que para a análise com integração reduzida a convergência só foi possível até o quarto incremento, conduzindo a uma estimativa inferior para a carga limite, mas ainda assim aceitável. A Figura IX.19 apresenta o aspecto da deformada final da análise a Figura IX.20 apresenta um detalhe da configuração deformada para o nível de carga final.

As informações relativas aos tempos de processamento para as análises efetuadas estão apresentados nas tabelas IX.4 e IX.5.

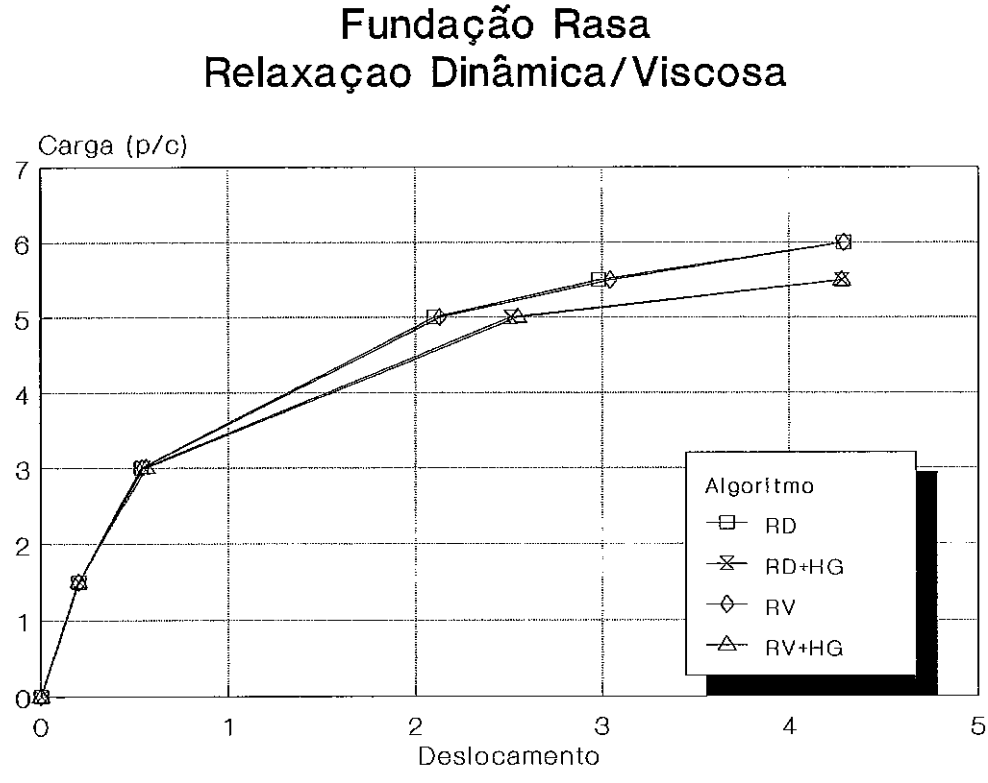


Figura IX.18: Análise limite de fundação rasa – curvas carga x deslocamento vertical

Análise Limite		
	Relax.Dinâmica	Relax.Viscosa
Iterações	2820	13243
CPU total (s)	168.0	491.8
CPU vetorial (s)	147.6	435.4
% vetorial	87.9	88.5

Tabela IX.4: Tempos para análise com integração "full-rank" e cinco incrementos de carga

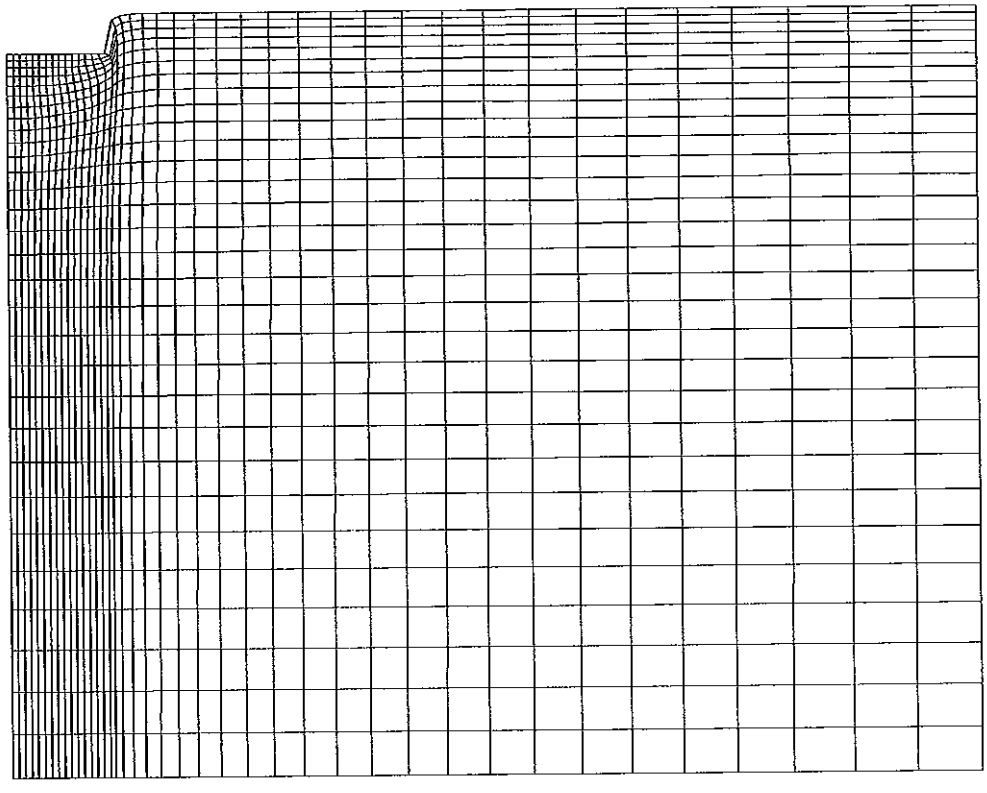


Figura IX.19: Análise limite de fundação rasa – deformada final

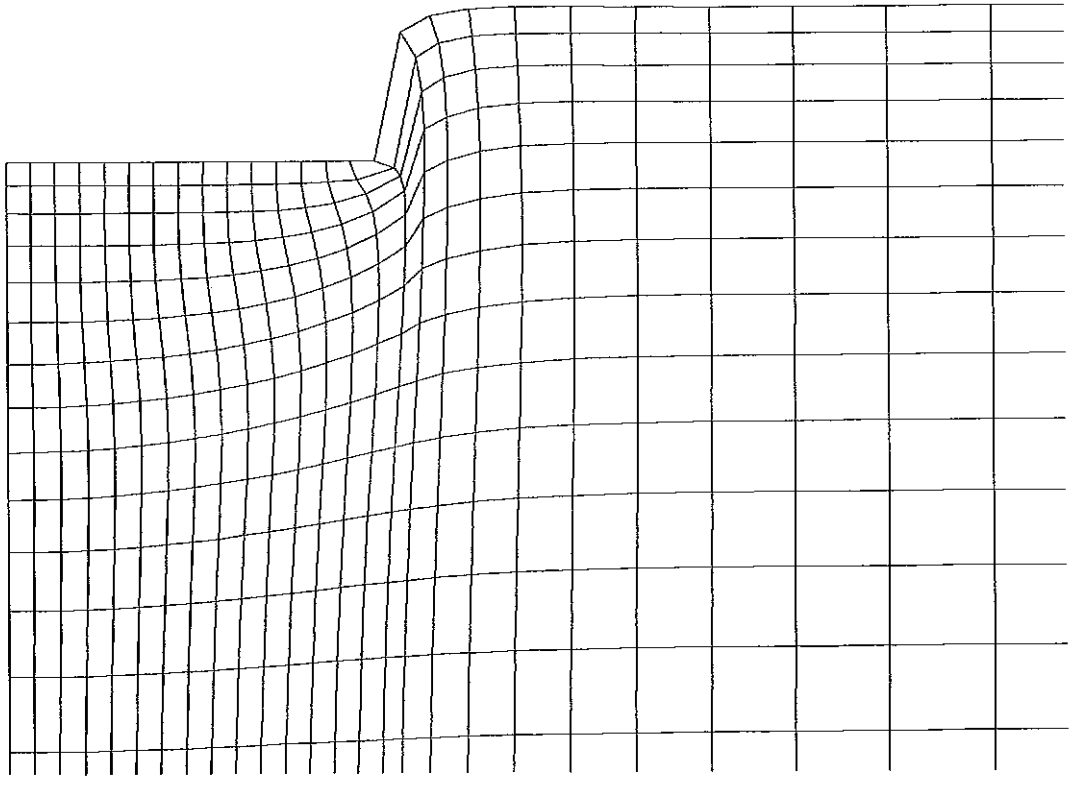


Figura IX.20: Fundação rasa – detalhe da deformada para carga limite

Análise Limite		
	Relax.Dinâmica	Relax.Viscosa
Iterações	2613	9535
CPU total (s)	75.0	204.3
CPU vetorial (s)	64.5	179.6
% vetorial	86.0	87.9

Tabela IX.5: Tempos para análise com integração reduzida e quatro incrementos de carga

IX.6 Desempenho da Implementação Paralela

Nesta seção serão apresentados resultados obtidos explorando-se uma das possibilidades de paralelização na análise transiente não-linear via elementos finitos.

Conforme exposto anteriormente, decidiu-se por explorar o paralelismo de baixa granularidade (*coarse grain*) tendo em vista a arquitetura do ambiente computacional. Nesse sentido, o principal foco de concorrência, facilmente identificado, é o da avaliação das forças internas em cada ponto de integração dos elementos.

Todas as análises foram efetuadas no IBM 3090 600E/6VF, cada uma empregando quatro tarefas concorrentes, sendo cada tarefa dedicada a um dos quatro pontos de Gauss e vetorizada ao longo dos elementos.

Para efeito das medições apresentadas a seguir, foram selecionadas duas discretizações, de ordem diferente, para o problema da calota esférica tratado na seção 2 deste capítulo, para a formulação Lagrangeana Total (TLF). A partir das duas discretizações formaram-se 3 problemas, conforme a Tabela IX.6.

Problema	Análises	Malha	Elementos	Equações	Intervalos de Tempo
1	S1-P1	60×4	240	595	2000
2	S2-P2	60×4	240	595	7000
3	S3-P3	100×10	1000	2189	7000
S# - análise sequencial-vetorial P# - análise concorrente-vetorial					

Tabela IX.6: Medição do desempenho em paralelo - Formação das baterias de testes

Todas as medições foram efetuadas em ambiente de produção, isto é, as análises foram processadas sem colocar o sistema em estado *dedicado*. Por tratar-se de um ambiente multi-usuário, as medições do desempenho de aplicações concorrentes sofrem, inapelavelmente, a influência da carga computacional global do sistema.

Para se ter uma noção dessa influência e observar o desempenho da implementação foram executadas 12 baterias de testes, distribuídas ao longo do dia segundo intervalos regulares de 2 horas entre cada bateria. Cada uma das 12 baterias de testes foi composta de uma execução seqüencial/vetorial e sua correspondente paralela/vetorial, para cada um dos 3 problemas, perfazendo um total de 6 análises por bateria.

Nas Tabelas IX.7-IX.9 são apresentados os resultados das medições efetuadas.

Problema 1				
Horário de Execução	Carga do Sistema (%)	Tempo decorrido (s)		Speed Up
		An.S1	An.P1	
03:00	30	20.17	14.93	1.35
05:00	27	20.32	15.00	1.36
07:00	35	25.05	18.58	1.35
09:00	99	53.65	73.22	0.74
11:27	82	24.65	28.98	0.85
13:00	95	23.11	44.61	0.52
15:00	80	22.82	29.54	0.77
17:00	75	23.39	36.32	0.64
20:48	50	20.44	12.27	1.67
21:30	85	21.81	18.95	1.15
23:00	80	21.14	27.02	0.78
02:18	30	20.34	12.49	1.63

Tabela IX.7: Resultado das medições para o problema da calota esférica com 240 elementos e 2000 intervalos de tempo

Caso as análises pudessem ser realizadas em ambiente dedicado, aplicando-se o modelo teórico de Amdahl seria possível a avaliação de um limite teórico para o ganho no desempenho (*speed-up*) dessa aplicação.

Problema 2				
Horário de Execução	Carga do Sistema (%)	Tempo decorrido (s)		Speed Up
		An.S2	An.P2	
03:00	30	73.12	40.78	1.79
05:00	27	70.13	41.60	1.69
07:00	35	70.71	49.13	1.44
09:01	99	203.0	243.9	0.83
11:26	82	115.6	89.10	1.30
13:01	95	86.20	148.2	0.58
15:00	80	78.65	93.99	0.84
17:01	75	94.68	79.81	1.19
20:48	50	71.23	39.31	1.81
21:31	85	73.48	52.77	1.39
23:00	80	72.44	63.12	1.15
02:19	30	70.30	62.69	1.12

Tabela IX.8: Resultado das medições para o problema da calota esférica com 240 elementos e 7000 intervalos de tempo

Problema 3				
Horário de Execução	Carga do Sistema (%)	Tempo decorrido (s)		Speed Up
		An.S3	An.P3	
03:01	30	432.6	190.8	2.27
05:01	27	423.1	198.2	2.13
07:01	35	431.8	198.2	2.18
09:05	99	987.8	1142.0	0.87
11:27	82	807.5	562.3	1.44
13:03	95	564.6	583.3	0.97
15:02	80	534.6	521.4	1.03
17:02	75	478.4	307.5	1.56
20:48	50	481.7	222.9	2.16
21:31	85	468.8	319.6	1.47
23:03	80	455.1	331.5	1.37
02:19	30	427.9	201.5	2.12

Tabela IX.9: Resultado das medições para o problema da calota esférica com 1000 elementos

Note-se que em vista dos horários de execução das baterias de testes, é muito provável que as análises tenham concorrido entre si, cada uma na disputa por quatro dos seis processadores reais disponíveis. Além disso, dada a filosofia do sistema operacional MVS/ESA e da implementação do MTF, nada impede que tarefas disparadas de um mesmo *job*, quando rodando em ambiente de produção, concorram pelo mesmo processador. A Figura IX.21 apresenta as curvas de desempenho observado, traçadas com base nos resultados das baterias de testes efetuadas.

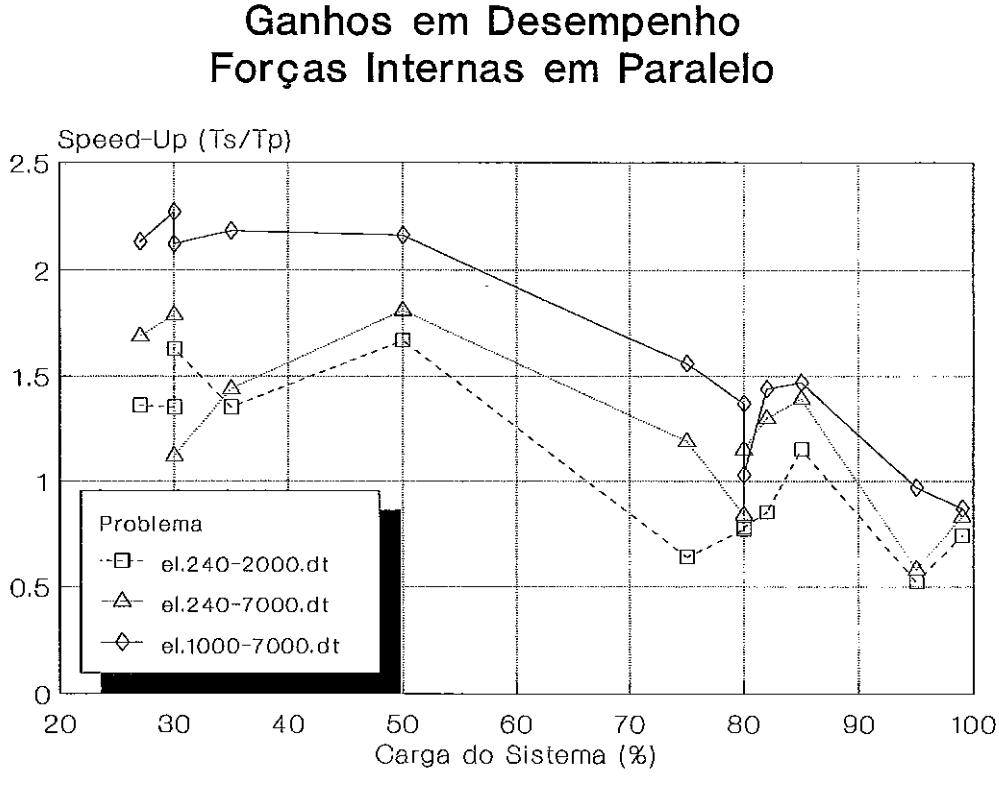


Figura IX.21: Curvas de desempenho em paralelo para os problemas 1, 2 e 3.

Observa-se claramente, como era de se esperar, que o melhor desempenho das análises concorrentes dá-se nos momentos de menor carga computacional do sistema.

De medições efetuadas com o auxílio do IBM FORTRAN-IAD sabe-se, por amostragem, o perfil de utilização de tempo de processador em cada rotina para cada problema analisado. Do modelo de paralelismo explorado na aplicação e da sua implementação, sabe-se que a rotina que efetua os cálculos nos pontos de Gauss é *perfeitamente* paralelizável. Imaginando-se uma situação ideal onde se verificasse uma relação 1:1 entre o tempo de processamento (CPU) e o tempo decorrido (*elapsed*) poderia ser aplicada a seguinte fórmula para a determinação do ganho máximo:

$$S_{max} = \frac{1}{(1 - p_{cpu}) + \frac{p_{cpu}}{n_p}} \qquad \text{(IX.1)}$$

onde p_{cpu} é o percentual de tempo total de processador dispendido na rotina paralelizável e n_p é o número de processadores empregados na execução em paralelo.

Na Tabela IX.10 estão apresentados os percentuais de processamento, o ganho teórico máximo e o ganho máximo observado nas medições, para cada um dos problemas abordados.

Problema	Elementos	CPU %	S_{max}	S_{med}
1	240	82	2.6	1.63
2	240	84	2.7	1.79
3	1000	86	2.85	2.27

Tabela IX.10: Comparação entre o ganho teórico máximo e o medido na execução concorrente

Faz-se notar na Tabela IX.10 o efeito do volume de operações envolvido nas tarefas disparadas sobre o desempenho real da aplicação. Os problemas 1 e 2, devido ao número reduzido de elementos presentes na discretização, provêm um volume de operações reduzido nas tarefas disparadas a cada intervalo de tempo da simulação. O desempenho fica então distante do teórico devido, entre outros fatores, principalmente à penalização inerente ao gerenciamento das tarefas no MTF.

Em contrapartida, no problema 3 tal efeito fica minimizado em decorrência de um peso maior das tarefas, dado a quantidade maior de elementos (cerca de 4 vêzes a dos problemas 1 e 2) sobre a qual cada tarefa concorrente deve operar. No caso do problema 3, vale notar que ainda que nos momentos de pico da carga computacional, a degradação no desempenho da implementação paralela ficou no entorno de 10%, um dado bastante animador quanto à adequação da estratégia empregada.

Capítulo X

Conclusões

Neste trabalho foram apresentados os aspectos relevantes da implementaçação do programa TRANSEP, destinado à análise em regime transiente de problemas da Mecânica do Contínuo via Método dos Elementos Finitos.

A principal preocupação ao longo do presente trabalho foi a de se obter uma implementação original, de grande eficiência computacional, em sintonia com as características de alto desempenho encontradas no sistema IBM 3090-600E/6VF. Ênfase especial foi dada aos aspectos relevantes da implementação vetorial e paralela do algoritmo, bem como do processo de otimização do código resultante.

A implementação adotada pautou-se em características bastante comuns a sistemas de arquitetura *shared memory*, incluindo facilidades para o processamento vetorial e suporte a multiprocessamento. Quanto à semântica do programa fonte, as construções adotadas são, via de regra, compatíveis com o atual estado da arte dos compiladores com potencial de geração de código vetorial.

Assim foi alcançado um outro objetivo importante no contexto desse trabalho: o de minimizar a dependência da implementação em relação a uma plataforma computacional específica, favorecendo a sua portabilidade entre sistemas de arquitetura equivalente.

São condições ideais para o sucesso de um empreendimento dessa natureza

- o domínio das técnicas numéricas envolvidas no processo;
- o conhecimento das características do ambiente computacional designado para a implementação;
- um perfeito entendimento quanto ao comportamento do sistema em relação aos algoritmos potencialmente aplicáveis ao problema em questão.

A conjunção desses requisitos fundamentou a decisão pela estrutura de dados e pela alternativa de implementação do algoritmo adotado.

A implementação obtida apresenta excelente conteúdo vetorial, situado na faixa de 90%, como observado ao longo das medições efetuadas. Este índice é bastante significativo, considerando-se aplicações científicas envolvendo intensa atividade numérica (*E/S-NIC Engineering/Scientific Numeric Intensive Computation*), como é o caso de aplicações do Método dos Elementos Finitos em problemas de Engenharia.

Foi também aplicada com sucesso a técnica de integração reduzida com a implementação vetorial do controle dos modos de energia nula (*hourglass*), proprocionando ganhos significativos às aplicações.

Com relação ao desempenho da versão paralelizada, os resultados foram animadores, ainda que pesem as penalizações inerentes ao ambiente oferecido pelo sistema operacional MVS/ESA. Considerando-se a influência da carga computacional do sistema sobre o desempenho de uma aplicação rodando em modo concorrente, em nenhuma medição efetuada observou-se uma degradação de performance superior a 13%, o que estimula o prosseguimento de pesquisas nessa linha. Cabe ressaltar que no contexto da versão 2.4 do compilador VS FORTRAN só é disponível um único modo de paralelismo, através das primitivas do MTF adequadas ao paralelismo *coarse grain*, não dispondo de capacidade de suporte a *microtasking*.

Outras estratégias de implementação de paralelismo no âmbito de aplicações do MEF poderiam ser exploradas dessa forma. Uma estratégia não abordada, por fugir ao escopo do presente trabalho, mas que apresenta um grande potencial com relação à eficiência do paralelismo e por essa razão digna de futuros estudos, é a do escalonamento seqüencial de tarefas (*sequential staging of tasks*) [Gaz89].

A alternativa de paralelismo explorada neste trabalho, paralelismo ao nível ds pontos de Gauss, não fica invalidada pela adoção de esquemas de integração reduzida. Essa estratégia pode ser facilmente adaptada para a análises considerando, por exemplo, os modelos de subcamadas. Nesse caso, cada tarefa e sua estrutura de dados seriam dedicadas a cada uma das subcamadas, considerando um ponto de integração por subcamada.

O alto desempenho da implementação, observado sobre um conjunto de aplicações representativas, demonstra a capacidade da estratégia empregada em explorar plenamente o potencial de sistemas de alto desempenho, que ofereçam facilidades de processamento vetorial e multiprocessamento em ambiente de memória compartilhada.

Face à a crescente demanda criada por projetos de alta tecnologia e das dimensões dos problemas encarados nesse contexto, a metodologia de desenvolvimento e implementação apresentada deve ser considerada sobretudo no aspecto estratégico da consolidação de tecnologia nacional.

A estratégia de implementação aqui empregada proprociona o esqueleto básico de uma potente ferramenta de análise, plenamente adequada ao ambiente oferecido pela classe de supercomputadores na qual se situa o IBM 3090.

Prevê-se a curto prazo uma difusão mais ampla dessa categoria de equipamentos no cenário técnico-científico nacional, justificando empreendimentos pautados nas mesmas premissas deste trabalho, quanto às estratégias de desenvolvimento e implementação, para a obtenção de ferramentas de análise modernas e mais potentes.

No tocante a futuras investigações e desenvolvimentos, à luz das observações realizadas nesse trabalho, pretende-se, no plano da análise transiente:

- extender as estratégias de implementação para problemas modelados com sólidos tri-dimensionais e cascas;
- a abordagem e tratamento de problemas de contacto e impacto estrutural em problemas planos e tri-dimensionais;
- o desenvolvimento de elementos de absorção de energia no contorno (fronteiras não-refletoras) para simulação de domínios infinitos em dinmica.

Finalmente, pretende-se estabelecer ainda uma base semelhante de desenvolvimentos, também voltados à análise transiente, em sistemas de arquitetura hipercúbica, tais como o iNTEL iPSC-860 recentemente instalado no Laboratório de Computação Paralela da COPPE/UFRJ.

Referências Bibliográficas

[Alv87] J.L.D.Alves, A.L.G.A.Coutinho, L.Landau e N.F.F.Ebecken - O Algoritmo de Relaxação Dinâmica Acelerada para a Análise Limite de Estruturas pelo Método dos Elementos Finitos, COLOQUIA'87.

[Alv90] J.L.D.Alves, A.L.G.A.Coutinho, L.Landau e N.F.F.Ebecken - A Vector/Parallel Multi-Step Algorithm for Transient Finite Element Analysis, anais Capri'90 - Parallel Computation: Achievements and Prospects, Capri, Itália, junho 1990.

[Alv91] J.L.D.Alves, L.A.Souza e N.F.F.Ebecken - Accelerated Relaxation Solution for Nonlinear Finite Element Analysis: A Vector-Parallel Implementation, anais NEC'91, Swansea, País de Gales, 1991.

[Atl84] S.N.Atluri - On Constitutive Relations at Finite Strain: Hypo-Elasticity and Elasto-Plasticity with Isotropic or Kinematic Hardening, C.M.A.M.E. v43 (1984) pp 137–171.

[Bat82] K.J.Bathe - Finite Element Procedures in Engineering Analysis. Prentice-Hall, Englewood Cliffs, 1982.

[Bel78] T.Belytschko - Explicit Time Integration of Structure-Mechanical Syatems, in Advanced Structural Dynamics, J.Donéa (ed), pp97-122, Applied Science Publishers, London, 1980.

[Bel87] T.Belytschko, N.Gilbertsen - Concurrent and Vectorized Mixed Time, Explicit Nonlinear Structural Dynamics Algorithm. Parallel Computations and their Impact on Mechanics, A.K. Noor (ed.), ASME 1987.

[Bel77] T.Belytschko, R.Mullen - Mesh Partitions of Explicit-Implicit Time Integration. Formulation and Computational Algorithms in Finite Element Analysis. K.J.Bathe, J.T.Oden, W.Wunderlich (eds.), MIT Press 1987.

[Bel89] T.Belytschko, T.J.R.Hughes, - Recent Advances in Nonlinear Finite Element Analysis, course notes, Palo Alto, 1989.

[Bel83] T.Belytschko - An Overview of Semidiscretization and Time Integration Procedures, pp 1-63 em Computational Methods for Transient Analysis, T.Belytschko e T.J.R.Hughes (eds.), North-Holland, Amsterdam, 1983.

[Bel84] T.Belytschko, J.S-J.Ong, W.K.Liu, J.M.Kennedy - Hourglass Control in Linear and Nonlinear Problems. C.M.A.M.E. 43 (1984) pp 251–276.

[Ben86] D.J.Benson e J.O.Hallquist - DYNA3D: A Finite Element Program for Supercomputers, Report UCRL-93113, Lawrence Livermore National Laboratory, 1986.

[Che88] W.F.Chen e D.J.Han - Plasticity for Structural Engineers, Springer-Verlag, London, 1988.

[Cla86] R.S.Clark, T.L.Wilson - Vector System Performance of the IBM 3090. IBM Systems Journal v25-1 (1986) pp 63–82.

[Cos88] A.M.Costa, L.F.R.Moreira, C.J.Medeiros Jr., L.H.Hassui, N.F.F.Ebecken, L.Landau, A.L.G.A.Coutinho, J.L.D.Alves, L.F.N.Mello - Pile Driving by the Finite Element Method, Offshore Engineering v.6, F.L.L.B.Carneiro, R.C.Batista, A.J.Ferrante (eds.) Pentech Press, London, 1988.

[Cou88] A.L.G.A.Coutinho, A.M.Costa, J.L.D.Alves, L.Landau, N.F.F.Ebecken - Pile Driving Simulation and Analysis by the Finite Element Method, Application of Stress-Wave Theory to Piles, 3rd International Conference, maio de 1988, Ottawa, Canadá, B.H.Fellenius (ed.) pp.197-207, 1988.

[Cou89] A.L.G.A.Coutinho, J.L.D.Alves, N.F.F.Ebecken - A Study of Implementations Schemes for Vectorized Matrix-Vector Multiplication. COPPE/UFRJ (1989) Research Report.

[Don89] J.J.Dongarra, I.S.Duff, D.C.Sorensen e H.A.Van der Vorst - Linear Algebra Computations on Vector and Parallel Computers, preprints 1989.

[Ebe88] N.F.F.Ebecken, L.Landau, A.L.G.A.Coutinho, J.L.D.Alves - Analysis of the Nonlinear Soil-Structure Interaction of Offshore Piles by an Explicit Finite Element Computer Program, Modelling Soil-Water-Structure Interactions SOWAS'88, P.A.Kolkman, J.Lindenberg, K.W.Pilarczyc (eds.), pp.335-342, A.A.Balkema, Rotterdam, 1988.

[Far87] C.Farhat, L.Crivelli - A general Approach to Nonlinear FE Computations on Shared Memory Multiprocessors. University of Colorado at Boulder (1987) Report no. CU-CSSC-87-09.

[Fel84] C.A.Felippa - Dynamic Relaxation under General Increment Control, anais da Int.Conf. on Innovative Methods for Nonlinear Problems, W.K.Liu, T.Belytschko e K.C.Park (eds.) , pp103, Pineridge Press, 1984.

[Fel86] C.A.Felippa - Solution of Nonlinear Static Equations, preprints de Computational Methods in Mechanics, volume em Large Deflection and Stability of Structures, K-J.Bathe.

[Fer87] R.M.Ferencz - Element-by-Element Preconditioning Techniques for Large-Scale, Vectorized Finite Element Analysis in Nonlinear Solid and Structural Mechanics, tese de doutorado (Ph.D.), Stanford University, 1989.

[Fer89] R.M.Ferencz - Element-by-Element Preconditioning Techniques for Large-Scale, Vectorized Finite Element Analysis in Nonlinear Solid and Structural Mechanics, tese Ph.D. Stanford University, 1989.

[Fla81] D.P.Flanagan, T.Belytschko - A Uniform Strain Hexahedron and Quadrilateral with Orthoginal Hourglass Control. I.J.N.M.E. v17 (1981) 679-706.

[Fla84] D.P.Flanagan, T.Belytschko - Eigenvalues and Stable Time Steps for the Uniform Strain Hexahedron and Qaudrilateral, Trans.ASME J.App.Mech., v51 (1984) pp35-40.

[Gaz89] J.Gazdag, H.H.Wang - Concurrent Computing by Sequential Staging of Tasks. IBM-Palo Alto Scientific Center (1989) Report G320-351.

[Gib86] D.H.Gibson, D.W.Rain, H.F.Walsh - Engineering and Scientific Processing on the IBM 3090. IBM Systems J. v25-1 (1986) pp 36-50.

[Hen87] R.A.Hendrickson, M.Metcalf - Coding for the Cray X-MP Using CFT77: An Introduction. CERN (1987) Report DD/87/11.

[Hug87] T.J.R.Hughes - The Finite Element Method, Linear Static And Dynamic Finite Element Analysis. Prentice-Hall, Englewood Cliffs, 1987.

[Hug89] T.J.R.Hughes e T.Belytschko - Recent Advances in Nonlinear Finite Element Analysis, notas de curso, Palo Alto, 1989.

[IBM87] IBM System/370 Extented Architecture - Principles of Operation, IBM publication SA22-7085-1, 1987.

[IBM86a] IBM System/370 - Vector Operations, IBM publication SA22-7125-1, 1986.

[IBM86b] IBM 3090 Vector Facility - Technical Reference, IBM International Technical Support Center, doc.no. GG24-3058, Poughkeepsie, NY, 1986.

[IBM88] Parallel Fortran - Language and Library Reference, manual da versão-beta, IBM publication SC23-0431-0, 1988.

[IBM90] VS FORTRAN Release 2.5 - Language and Library Reference e User Guide, IBM , 1990.

[Jar87] W.Jarzab, G.Bretz, R.Schwarz - Numerical Simulation and Experimental Validation in Crashworthiness Applications, in Structural Design and Crashworthiness of Automobiles, T.K.S.Murthy e C.A.Brebbia (eds.), Springer-Verlag, 1987.

[Joh72] W.Johnson - Impact Strength of Materials, Edward Arnold Ltd., London, 1972.

[Kam90] A.Kamoulakos - A Simple 'Benchmark' for Impact, NAFEMS Benchmark, july 1990, pp.31-35.

[Ken86] J.M.Kennedy e T.B.Belytschko - STRAW - A Nonlinear Transient Fluid-Structural Analysis Code, ANL Report, 1986.

[Koh87] B.C.Koh, N.Kikuchi - New Improved Hourglass Control for Bilinear and Trilinear Elements in Anisotropic Linear Elasticity. Comp. Meth. App. Mech. Engng., v65 (1987) pp 1-46.

[Koj87] M.Kojic e K-J.Bathe - Studies of Finite Element Procedures - Stress Solution of a Closed Elastic Strain Path with Stretching and Shearing Using the Updated Lagrangian Jaumann Formulation, Computers & Structures v26(1/2) (1987) pp 175--179.

[Kri77] R.D.Krieg e D.B.Krieg - Accuracies of Numerical Solution Methods for the Elastic-Perfectly Plastic Model, Trans.ASME J.Pres.Ves.Tech. v90-4 (1977) pp.510-515.

[Kul88] R.F.Kulak, C.Fiala - Neptune: A System of Finite Element Programs for Three-Dimensional Nonlinear Analysis . Nucl.Engng.Design, v106 (1988) pp 47-68.

[Lin89] J.I.Lin - An Element Eigenvalue Theorem and its Application for Stable Time Steps, C.M.A.M.E. v73 (1989) pp183-294.

[Mal69] L.E.Malvern - Introduction to the Mechanics of a Continuous Medium, Prentice-Hall, Englewood Cliffs, 1969.

[Mel89] P.Melli, P.Angeleri, D.F.Lozupone - Supercomputing in Engineering Structures: The ECSEC Experience on the IBM 3090/VF. Computer and Experiments in Stress Analysis. C.A.Brebbia, G.M.Carlomagno (eds.), Springer-Verlag, 1989.

[Nag75] S.Nagarajan, P.Popov - Nonlinear Dynamic Analysis of Axysimmetric Shells. I.J.N.M.E. v9 (1975) pp 535-550.

[Noo89] A.K.Noor - New Computing Systems and their Impact on Structural Analysis, Supercomputing in Engineerin Structures, P.Melli e C.A.Brebbia (eds.), Springer-Verlag, Berlin, 1987.

[Ode72] J.T.Oden - Finite Elements of Nonlinear Continua, McGraw-Hill, New York, 1972.

[Ode73] J.T.Oden e J.E.Key - Analysis of Static Non-Linear Response by Explicit Time Integration, I.J.N.M.E v7 (1973) pp225-240.

[Ott65] J.R.H.Otter - Computations for Prestressed Concrete Reactor Pressure Vessels Using Dynamic Relaxation, Nucl.Struc.Engng. v1 (1965) pp65-75.

[Owe80] D.R.J.Owen e E.Hinton - Finite Elements in Plasticity: Theory and Practice, Pineridge Press, Swansea (1980).

[Pap81] M.Papadrakis - A method for the Automatic Evaluation of the Dynamic Relaxation Parameters, C.M.A.M.E. v25 (1981) pp 35-48.

[Per89] L.C.T.Pereira, F.Yuan, A.V.Alves - Análise da Otimização / Vetorização do DINEXP. PETROBRA]S, relatório interno (1989).

[Qia88] S.Qiang - An Adaptive Dynamic Relaxation Method for Nonlinear Problems, Computers & Structures, v30 (1988) pp 855-859.

[Rer86] P.Řeřicha - Optimum Load History for Non-Linear Analysis Using Dynamic Relaxation, I.J.N.M.E. v23 (1986) pp 2313-2324.

[Sca87] R.G.Scarborough, H.G.Kolsky - A Vectorizing FORTRAN Compiler. IBM J.Res.Develop. v30-2 (1986) pp 163-171.

[Ser89] F.J.Serón, F.J.Sanz, M.Kindelan - Elastic Wave Propagation with the Finite Element Method, IBM ECSEC Technical Report ICE-0028, fevereiro de 1988.

[Sim88] J.C.Simo e T.J.R.Hughes - Elastoplasticity and Viscoplasticity: Computational Aspects, pre-prints, 1988.

[Sch87] W.Schönauer - Scientific Computing on Vector Processors, North-Holland, Amsterdam, 1987.

[Sch79] H.L.Schreyer, R.F.Kulak e J.M.Kramer - Accurate Numerical Solutions fo Elastic-Plastic Models, Trans.ASME J.Pres.Vess.Tech v101 (1979) pp.226-234.

[Sha76] D.Shantaram, D.R.J.Owen, O.C.Zienkiewicz - Dynamic Transient Behaviour of Two- and Three-Dimensional Structures Including Plasticity, Large Deformation Effects and Fluid Interaction. Earth.Engng.Str.Dyn. v4 (1976) 561-578.

[Shi89] D.Shie e B.Swope - Vector Facility Seminar, notas de curso, IBM do Brasil-Unicamp, Campinas, setembro de 1989.

[Sol86] D.B.Soll - Vectorization and Vector Migration Techniques, IBM Technical Bulletin, R.Stephens (ed.), IBM Technical Education Center, Chicago, IL, 1986.

[Tgk89] T.Torigaki - A Structure of an Explicit Time Integration Dynamic Finite Element Program Suitable for Vector and Parallel Computers. Proc. of the 4th Int. Conf. on Supercomputing and 3rd World Supercomputer Exhibition (ICS 89) - Santa Clara, CA, April/May 1989.

[Tuc86] S.G.Tucker - The IBM 3090 System : An Overview. IBM Systems Journal v25-1 (1986) pp 4-19.

[Und83] P.Underwood - Dynamic Relaxation - in: Computational Methods for Transient Analysis, T.Belytschko and T.J.R.Hughes (eds.). North Holland, Amsterdam,(1983) pp. 245-266.

[Web90] H.Weberpals - Architectural Approach to the IBM 3090 E Vector Performance. Parallel Comp. v13 (1990) pp 47-59.

[Zie85] O.C.Zienkiewicz e R.Lohner - Accelerated 'Relaxation' or Direct Solution? Future Prospects for FEM, I.J.N.M.E, v12 (1985) pp.1-11.

Apêndice A

Medidas de Desempenho - Lei de Amdahl

Para a avaliação e comparação do desempenho computacional da implementação foi empregada a medida de aceleração observada ou real (*speed-up real*) para processamento vetorial e paralelo, que por sua simplicidade mostram-se bastante adequadas no âmbito do presente trabalho. O modelo teórico para essas medidas é conhecido pela denominação *Lei de Amdahl* e fornecem o limite superior (teórico) do ganho em desempenho de uma implementação.

Segue-se a definição suscinta das medidas adotadas ao longo do trabalho.

Define-se o *speep-up real vetorial* de uma aplicação segundo :

$$S_v = \frac{T_{esc}}{T_v} \tag{A.1}$$

que mede o ganho relativo, em termos do tempo de CPU, entre a execução em modo escalar (T_{esc}) e a execução em modo vetorial (T_v).

O *speed-up* teórico é o limite superior dado pela relação

$$S_{v_{max}} = \frac{1}{(1-\alpha) + \frac{\alpha}{v_v}} \tag{A.2}$$

onde α representa a fração (porcentagem) do código executada em modo vetorial

e v_s é a taxa de performance nominal entre a CPU vetorial e a CPU escalar, ou seja, o quantas vezes é mais rápida a CPU vetorial em relação à CPU escalar para uma mesma operação.

Analogamente, define-se o *speed-up real paralelo*:

$$S_{par} = \frac{T_{ser}}{T_{par}} \tag{A.3}$$

Esse valor mede o ganho, em termos do tempo de execução, através da relação entre os tempos *elapsed* para uma execução serial (sem paralelismo - T_{ser}) e para a execução que explora o paralelismo (T_{par}).

O limite teórico, no caso de processamento paralelo, é dado por:

$$S_{p_{max}} = \frac{1}{(1-\alpha) + \frac{\alpha}{n}} \tag{A.4}$$

onde α é a fração do codigo designada para execução em paralelo empregando-se n processadores.

Outras medidas mais elaboradas podem ser empregadas. Uma exposição a respeito destas e de outras medidas de desempenho, bem como uma discussão sobre os limites de sua aplicabilidade e desvios verificados na prática, pode ser encontrada na referência [Sch87].