ALGORITMOS OTIMIZADOS PARA A ANÁLISE ACOPLADA
DE SISTEMAS FLUTUANTES NA EXPLORAÇÃO DE PETRÓLEO OFFSHORE

Marcos Vinícius Rodrigues

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 A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

_____

Prof. Breno Pinheiro Jacob, D.Sc.

_____

Prof. Webe João Mansur, Ph.D.

_____

Prof. Antonio Carlos Fernandes, Ph.D.

_____

Prof. Elson Magalhães Toledo, D.Sc.

_____

Eng. Márcio Martins Mourelle, D.Sc.

_____

Prof. Paulo Batista Gonçalves, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
DEZEMBRO DE 2004

RODRIGUES, MARCOS VINÍCIUS

Algoritmos Otimizados para a Análise Acoplada de Sistemas Flutuantes na Exploração de Petróleo Offshore

[Rio de Janeiro] 2004

IX, 197 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Civil, 2004)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Sistemas Offshore

2. Análise Acoplada

3. Computação Paralela

I. COPPE/UFRJ II. Título (série)

Dedico à minha esposa Fátima
e aos meus pais Fábio e Elizabete.

AGRADECIMENTOS

A Deus por tudo.

À minha esposa Fátima pelo carinho, amizade e companheirismo.

Aos meus pais Fábio e Elizabete e meus irmãos Diogo e Raquel, por constituírem a fundamental base familiar.

Ao Professor Breno Pinheiro Jacob pelo apoio e incentivo durante todo o doutorado, mas acima de tudo pela amizade e pela convivência.

Aos companheiros de "batalha" Eduardo Vardaro, Fabrício Nogueira, Luciano Franco, Luciano Tardelli, Glauco Rodrigues, Carl Albrecht e Fábio Simões.

Aos meus amigos e colegas de trabalho do LAMCSO (Laboratório de Métodos Computacionais em Sistemas Offshore) do Programa de Engenharia Civil COPPE/UFRJ e do LAMVI (Laboratório de Métodos Visuais) do Departamento de Expressão Gráfica da EE/UFRJ.

Aos amigos e colegas de trabalho do Programa de Engenharia Civil da COPPE – UFRJ e do CENPES - PETROBRAS pelo apoio e incentivo ao longo destes anos.

Ao CNPq pelo apoio financeiro.

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

ALGORITMOS OTIMIZADOS PARA A ANÁLISE ACOPLADA DE SISTEMAS FLUTUANTES NA EXPLORAÇÃO DE PETRÓLEO OFFSHORE

Marcos Vinícius Rodrigues

Dezembro/2004

Orientador: Breno Pinheiro Jacob

Programa: Engenharia Civil

O principal objetivo deste trabalho é a busca de redução de tempo computacional na análise dinâmica não-linear acoplada de unidades flutuantes, linhas de ancoragem e risers. Buscando este objetivo a tese é dividida em duas principais linhas de estudo:

- Implementação de Subciclagem associada aos Métodos Explícitos de Integração no Tempo;
- Implementação dos Algoritmos de Partição de Domínio Implícitos Iterativos (PDII) em Métodos Implícitos de Integração no Tempo.

A subciclagem consiste na utilização de diferentes intervalos de tempo em diferentes trechos da malha de uma linha de ancoragem ou riser, discretizados por elementos finitos e integrados ao longo de uma análise dinâmica de maneira acoplada à plataforma.

Os Algoritmos de Partição Implícitos Iterativos consistem em um método de partição de domínio onde uma linha de ancoragem ou riser é dividida em segmentos e cada segmento é integrado independentemente, também de forma acoplada à plataforma, com o auxílio da computação paralela.

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

OPTIMIZED ALGORITHMS FOR A COUPLED ANALYSIS
OF FLOATING SYSTEMS IN OFFSHORE OIL EXPLORATION

Marcos Vinícius Rodrigues

December/2004

Advisor: Breno Pinheiro Jacob

Department: Civil Engineering

The aim of this work is the search for reduction of computational time in coupled non-linear dynamic analysis of floating units, mooring lines and risers. This thesis has two main lines of study:

- Implementation of Subcycling Algorithm with Explicit Time Integration Methods;
- Implementation of Iterative Group Implicit (IGI) Algorithm with Implicit Time Integration Methods.

The main characteristic of subcycling algorithm is the use of different time-step along the mesh of a mooring line or riser, discretized by Finite Element Method and integrated in a coupled dynamic analysis.

The Iterative Group Implicit Method is a Domain Partition Method where a mooring line or riser is partitioned in subdomains and each subdomain is solved independently in a context of a coupled dynamic analysis, with a parallel computation tool.

ÍNDICE

1. INTRODUÇÃO 1
1.1. Contexto 1
1.2. Motivação 2
1.3. Objetivo 3
1.4. Descrição dos Capítulos 6

2. SISTEMAS OFFSHORE 7
2.1. Plataformas para Exploração de Petróleo Offshore 7
2.2. Risers 11
2.3. Linhas de Ancoragem 13

3. EQUAÇÕES DE MOVIMENTO DA UNIDADE FLUTUANTE 16
3.1. Introdução 16
3.2. Sistemas de Referência 16
3.3. Formulação das Equações de Movimento 19
3.4. Solução das Equações de Movimento 26

4. FORMULAÇÃO ESTRUTURAL DAS LINHAS 31
4.1. Introdução 31
4.2. Modelo Matemático; Solução Numérica 31
4.3. Discretização Espacial 32
4.3.1. Elemento de Treliça 33
4.3.2. Elemento de Pórtico 33
4.4. Discretização no Tempo - Solução Numérica de Problemas Dinâmicos Lineares 34
4.4.1. Formulação do Problema Dinâmico 34
4.4.2. Procedimento de Solução do Problema Dinâmico 35

5. CARREGAMENTOS AMBIENTAIS 39
5.1. Introdução 39
5.2. Ondas 39
5.2.1. Modelo Matemático 39
5.2.2. Teoria Linear de Airy 41
5.2.3. Representação Espectral 44
5.2.4. Cálculo das Forças 46
5.2.5. Modelo Híbrido 51
5.3. Correnteza 54
5.4. Vento 55

6. MÉTODOS EXPLÍCITOS DE INTEGRAÇÃO NO TEMPO 58
6.1. Introdução 58
6.2. Método das Diferenças Centrais 59
6.3. Algoritmo de Chung e Lee 61
6.4. Método Explícito Generalizado - Algoritmo de Hulbert e Chung 62

6.5. Comparação entre os algoritmos 64

7. MÉTODOS IMPLÍCITOS DE INTEGRAÇÃO NO TEMPO 67
7.1. Introdução 67
7.2. Implementação por Deslocamentos 67
7.3. Problemas não-lineares em algoritmos implícitos 69
7.4. Solução do Problema Dinâmico: O algoritmo αB-Newmark 72
7.5. Tratamento de Problemas Não-lineares: Implementação Otimizada αB-Newmark/Newton-Raphson 74
7.6. Aspectos da Implementação 78

8. OTIMIZAÇÕES PARA MÉTODOS EXPLÍCITOS DE INTEGRAÇÃO 81
8.1. Introdução 81
8.2. Subciclagem 82
8.2.1. Subciclagem Casco-Linhas 82
8.2.2. Subciclagem Interna das Linhas 83
8.2.3. Aspectos da Implementação 84
8.2.4. Subciclagem Interna na Análise Acoplada 86
8.2.5. Comentários 87
8.3. Exemplos 88
8.3.1. Viga Biengastada sob Carga Transversal 88
8.3.2. Viga Monoengastada sob Carga Axial 90

9. OTIMIZAÇÕES PARA MÉTODOS IMPLÍCITOS DE INTEGRAÇÃO 92
9.1. Partição do Domínio 92
9.2. Algoritmo "Partição de Domínio Implícito" (PDI) 93
9.3. Algoritmo "Partição de Domínio Implícito Iterativo" (PDII) 96
9.4. Algoritmo "Partição de Domínio Implícito Iterativo" (PDII) para Problemas Não-Lineares 98
9.5. Aspectos da Implementação 99
9.6. Exemplos 103
9.6.1. Viga Biengastada sob Carga Transversal 103
9.6.2. Viga Monoengastada sob Carga Axial 104

10. IMPLEMENTAÇÃO DO MÉTODO PDII EM COMPUTADORES COM ARQUITETURA PARALELA 106
10.1. Ambiente de Computação Paralela 106
10.1.1. Introdução 106
10.1.2. Paralelismo 106
10.1.3. Biblioteca de comunicação MPI 108
10.1.4. Medidas de desempenho 111
10.2. Algoritmo PDII em Computadores com Arquitetura Paralela 114
10.2.1. Introdução 114
10.2.2. Análise do código sequencial 114
10.2.3. Estratégia para Implementação em Paralelo 116

11. APLICAÇÕES NUMÉRICAS 121
11.1. Introdução 121

11.2. Riser em Catenária sob Movimento Imposto no Topo 123
11.2.1. Descrição do Modelo 123
11.2.2. Dados Geométricos e do Material 124
11.2.3. Carregamento 124
11.2.4. Subciclagem - Resultados 125
11.2.5. PDII - Resultados 127
11.2.6. PDII – Variação do número de processadores 129
11.3. Riser em Catenária sob Ação de Correnteza 136
11.3.1. Descrição do Modelo 136
11.3.2. Carregamento 136
11.3.3. Subciclagem - Resultados 137
11.3.4. PDII - Resultados 138
11.4. Plataforma P10 141
11.4.1. Características do Casco 141
11.4.2. Características das Linhas de Ancoragem 143
11.4.3. Características do Riser de Perfuração 143
11.4.4. Dados Ambientais 144
11.4.5. Modelo Numérico 144
11.4.6. Subciclagem - Resultados 149
11.4.7. PDII - Resultados 153
11.5. Plataforma P18 160
11.5.1. Características do Casco 160
11.5.2. Características das Linhas de Ancoragem 162
11.5.3. Características dos Risers 163
11.5.4. Dados Ambientais 166
11.5.5. Modelo Numérico 167
11.5.6. Subciclagem - Resultados 173
11.5.7. PDII - Resultados 177

12. CONCLUSÕES 186
12.1. Considerações Finais 186
12.1.1. Algoritmos Explícitos e Subciclagem 186
12.1.2. Algoritmos PDII 187
12.2. Propostas para Desenvolvimentos Futuros 190
12.2.1. Subciclagem 190
12.2.2. Algoritmos PDII 190
12.2.3. Combinação Subciclagem / Algoritmos PDII 191

13. REFERÊNCIAS 193

1. INTRODUÇÃO

1.1. CONTEXTO

A exploração de petróleo em alto-mar (offshore) avança em grande velocidade rumo a novas fronteiras até então inimagináveis. O Brasil, através da Petrobras, é pioneiro nesta exploração em águas profundas e impulsiona a pesquisa neste sentido, pois, a cada novo desafio, novas ferramentas computacionais são necessárias de forma a prever-se, da melhor forma possível, o comportamento de estruturas responsáveis por esta exploração.

Os métodos numéricos contribuem neste sentido fazendo com que, a partir de simulações em computadores, seja possível prever o comportamento de sistemas estruturais para suporte a plataformas de exploração de petróleo, analisando possíveis problemas e soluções antes mesmo de se ir a campo. Para se obter uma boa avaliação numérica, torna-se necessária uma representação de todos os componentes de um sistema offshore, incluindo plataforma, linhas de ancoragem e dutos de transporte de óleo, os risers.

1.2. MOTIVAÇÃO

As ferramentas numéricas tradicionalmente usadas na análise de unidades flutuantes ancoradas adotam um procedimento de análise desacoplada, que trata os movimentos do casco da unidade flutuante separadamente do comportamento estrutural dinâmico não-linear das linhas de ancoragem e risers.

A análise desacoplada, de um modo geral, ignora o fato de que o casco, as linhas de ancoragem e os risers compõem um sistema integrado, introduzindo simplificações que fazem com que a interação do comportamento dinâmico não linear destes componentes não seja considerada de forma rigorosa, o que pode penalizar seriamente a qualidade dos resultados.

Sabe-se que as simplificações relacionadas a este procedimento de análise desacoplada se tornam mais graves para sistemas com grande número de risers, e/ou instalados em lâminas d'água profundas.

Por outro lado, a utilização de uma formulação acoplada permitirá avançar além do estado-da-arte atual de projeto, contribuindo para a integração entre o projeto de ancoragem e o projeto dos risers. Além disso, os resultados serão mais precisos do que os obtidos através de uma seqüência de análises desacopladas, já que a formulação acoplada considera implicitamente e automaticamente os efeitos não-lineares e dinâmicos decorrentes da interação entre os cascos e as linhas.

O grande problema encontrado é a viabilidade de um projeto de um sistema a partir de análises acopladas. Geralmente, em projetos de sistemas de unidade flutuante, linhas de ancoragem e risers, o tempo de processamento é um fator importantíssimo e que deve ser ponderado durante a execução do projeto. Atualmente as análises acopladas ainda não têm se mostrado competitivas com as formas tradicionais de desenvolvimento de projetos devido ao elevado custo de CPU despendido neste tipo de análise.

Portanto, a grande motivação desta tese é caminhar no sentido de fazer com que as análises acopladas se tornem competitivas e atraentes como ferramenta de projeto de unidades flutuantes ancoradas. Esta contribuição se dará através de algumas, entre muitas outras possíveis, formas de otimização desta análise acoplada.

1.3. OBJETIVO

Devido à grande quantidade de graus de liberdade envolvidos em uma análise dinâmica acoplada, onde as linhas conectadas à plataforma são discretizadas em elementos finitos, mostram-se pertinentes otimizações nos métodos de integração no tempo. As otimizações estudadas neste trabalho concentram-se nas linhas, pois são as linhas discretizadas as responsáveis pelo maior consumo de CPU em uma análise acoplada.

Os algoritmos otimizados para a solução de problemas não-lineares dinâmicos no domínio do tempo têm como principal objetivo reduzir os requisitos de tempo de processamento de uma análise acoplada. Os métodos a serem investigados se apresentam de uma forma resumida a seguir e serão detalhados nos capítulos adiante:

- Técnicas de Partição do Domínio, incluindo subciclagem;
- Baseando-se na partição do domínio, a implementação em computadores com arquitetura paralela de modo a adaptar os algoritmos para a execução em máquinas com processamento paralelo e assim reduzir o tempo de processamento de programas acoplados.

De acordo com o tipo de algoritmo empregado, as otimizações implementadas seriam agrupadas da seguinte forma:

- Otimizações para Métodos Explícitos de Integração no Tempo;
- Otimizações para Métodos Implícitos de Integração no Tempo.

As otimizações para Métodos Explícitos se concentram no requisito de utilização, em uma análise dinâmica, de um intervalo de tempo menor que um intervalo de tempo "crítico". Conforme será apresentado no capítulo 6 o intervalo de tempo crítico depende do menor período natural da discretização da malha de elementos finitos da estrutura. E este é, sem dúvida, o ponto crítico numa análise explícita de integração no tempo: o intervalo de tempo muito reduzido necessário para análise.

De um modo geral, a discretização de um riser ou linha de ancoragem varia ao longo do comprimento. Isto é, refina-se a malha em trechos críticos como o topo ou a região de toque no fundo (Touch Down Point) enquanto que os trechos retos apoiados no fundo ou suspensos têm uma malha menos discretizada. Esta característica nos induz

à observação de que seria possível a utilização de diferentes intervalos de tempo para cada trecho da estrutura de acordo com a sua discretização. O que acontece corriqueiramente é a utilização do menor intervalo de tempo, requerido pela malha mais discretizada, ao longo de toda a estrutura. Portanto, uma das implementações aqui efetuadas, é justamente esta utilização, de diferentes intervalos de tempo para cada trecho da estrutura, denominada subciclagem. Este estudo é detalhado no capítulo 8.

As otimizações para Métodos Implícitos de integração no tempo têm aqui um outro enfoque. Estes algoritmos, por serem incondicionalmente estáveis, permitem a utilização de intervalos de tempo muito superiores aos utilizados nos Métodos Explícitos. Isto é, não existe mais a limitação do intervalo de tempo devido à discretização da malha. Esta limitação aparece apenas na precisão dos resultados. O intervalo de tempo pode ser aumentado o quanto for desejado desde que seja respeitada a precisão dos resultados.

O ponto crítico nestes algoritmos é justamente a necessidade da resolução do sistema de equações a cada intervalo de tempo, ou, em caso de problemas não-lineares, que é o problema em questão, a cada iteração.

Portanto a otimização mais eficiente neste caso se daria não mais em termos de intervalo de tempo mas sim na busca de eficiência na resolução do sistema de equações.

Neste caso, a busca da eficiência foi feita pela partição do domínio em subdomínios e a resolução concorrente de cada subdomínio independentemente, em paralelo, aumentando em número, mas reduzindo o tamanho do sistema de equações a ser resolvido em um cada intervalo de tempo, ou em cada iteração. Este aumento de número de sistemas de equações deixa de ser um problema desde que exista um número de processadores suficientes para se distribuir estes sistemas de equações reduzidos.

Portanto esta é a implementação efetuada visando a otimização dos Métodos Implícitos de integração no tempo e descrita nos capítulos 9 e 10.

Os desenvolvimentos apresentados neste trabalho são implementados no programa de análise acoplada PROSIM [1]. Este programa é baseado em uma formulação acoplada que incorpora, em uma única estrutura de código e de dados, um modelo hidrodinâmico para a representação do casco da unidade flutuante, e modelos de elementos finitos para a representação rigorosa das linhas.

O programa PROSIM é baseado em uma formulação acoplada onde, a cada instante do processo de integração no tempo das equações de movimento do casco efetua-se uma análise não-linear dinâmica de um modelo de elementos finitos de cada uma das linhas, sob ação da onda, correnteza, peso próprio, e das componentes de movimento transmitidas pelo casco. As forças no topo de cada linha, obtidas como resultado destas análises, são então aplicadas no lado direito das equações de movimento do casco.

Na versão atual do programa PROSIM, o modelo hidrodinâmico do casco é baseado em uma formulação híbrida que combina a fórmula de Morison com a teoria da Difração. A formulação de Morison original é adequada para membros que podem ser representados por elementos unifilares com diâmetros pequenos em relação ao comprimento das ondas, de modo que as ondas incidentes não são perturbadas. A formulação híbrida adotada no programa PROSIM permite representar a difração e radiação das ondas que ocorrem em membros cilíndricos de maior diâmetro; nesta formulação, as forças de deriva lenta, bem como o amortecimento dependente da freqüência das ondas ("radiation damping"), são incorporados através da leitura de coeficientes gerados por um programa de difração como o WAMIT.

O uso desta formulação faz com que o programa PROSIM seja adequado para a análise de unidades flutuantes compostas por membros cilíndricos de pequenos ou grandes diâmetros.

1.4. DESCRIÇÃO DOS CAPÍTULOS

Os demais capítulos estão organizados da seguinte forma:

O capítulo 2 apresenta um resumo com os principais componentes de um sistema offshore: plataforma, risers e linhas de ancoragem.

O capítulo 3 apresenta uma revisão da formulação teórica das equações de movimento da unidade flutuante.

O capítulo 4 apresenta a formulação estrutural para solução das linhas de ancoragem e risers.

O capítulo 5 apresenta a contribuição dos carregamentos ambientais como forças externas aplicadas à unidade flutuante e às linhas.

O capítulo 6 apresenta descrição dos algoritmos explícitos de integração no tempo.

O capítulo 7 apresenta descrição dos algoritmos implícitos de integração no tempo.

O capítulo 8 apresenta as otimizações desenvolvidas e implementadas para os métodos explícitos de integração no tempo, baseados no algoritmo de subciclagem.

O capítulo 9 apresenta as otimizações desenvolvidas e implementadas para os métodos implícitos de integração no tempo, baseados no algoritmo de "Partição de Domínio Implícito".

O capítulo 10 apresenta um resumo sobre o ambiente de computação paralela e a implementação do algoritmo de "Partição de Domínio Implícito Iterativo" em computadores com arquitetura paralela.

O capítulo 11 apresenta uma série de aplicações numéricas comparando resultados e desempenho dos algoritmos implementados.

O capítulo 12 apresenta as conclusões obtidas no presente trabalho e propostas para desenvolvimentos futuros.

2. SISTEMAS OFFSHORE

Para melhor entendimento dos conceitos básicos envolvendo a engenharia offshore, algumas considerações serão apresentadas nos próximos itens:

2.1. PLATAFORMAS PARA EXPLORAÇÃO DE PETRÓLEO OFFSHORE

- **Plataforma fixa**

Inicialmente a extração de petróleo offshore no Brasil era efetuada em lâmina d'águas denominadas rasas, com profundidades variando de 100m a 500m. Para tal eram utilizadas plataformas fixas (Fig 1) apoiadas no leito marinho. Como estas plataformas são fixadas no fundo e são estruturas relativamente rígidas, os efeitos dinâmicos e os efeitos não lineares devido aos carregamentos de onda, vento e correnteza não se apresentam de forma muito significativa.

Figura 1 – Plataforma Fixa

À medida que foram sendo descobertos novos reservatórios de petróleo em lâminas d'água mais profundas (500m a 1000m), observou-se que a freqüência natural deste tipo de plataforma se aproximava perigosamente da freqüência de excitação causada pelas ondas. Isto poderia fazer com que o sistema entrasse em ressonância

ocasionando um desastre de grandes proporções. Para evitar-se este problema seria necessário construir uma estrutura muito rígida, o que mostrou-se economicamente inviável.

Para compor novas alternativas na exploração de petróleo em águas profundas, foram introduzidos os sistemas flutuantes ancorados no fundo do mar por meio de cabos. Estes sistemas são descritos a seguir.

- **Plataforma semi-submersível**

As semi-submersíveis (Figura 2) são plataformas com estruturas flutuantes largamente empregadas para produção, completação e perfuração. Consistem de dois flutuadores compartimentados em tanques com finalidades de oferecer lastro e flutuação à plataforma. Estes flutuadores são denominados de "pontoons", os quais apóiam as colunas, também chamadas de pernas, e que por sua vez sustentam os conveses. Sua profundidade pode ser alterada através do bombeio de água para o tanque de lastro.

Figura 2 – Plataforma Semi-submersível

As Semi-submersíveis podem ser empregadas tanto em produção quando perfuração. As plataformas Semi-submersíveis de perfuração (figura 3) são geralmente denominadas de MODU (*Mobile Offshore Drilling Unit*).

Figura 3 – Plataforma de Perfuração

- **Navios FPSO**

Navios do tipo FPSO (Floating Production, Storage and Offloading Vessel ou Unidade de Produção, Armazenamento e Alívio de Petróleo) (Fig 4), que são navios adaptados a extrair, armazenar e exportar petróleo, estando estes também ancorados ao fundo do mar por meio de cabos.

Figura 4 – Unidade FPSO

- **TLP**

A TLP (*tension leg platform*) consiste numa estrutura similar à semi-submersível, sendo mantida na locação através de tirantes (pernas) que são ancorados

no fundo através de estacas e tracionadas no topo pela força resultante entre peso e empuxo (*restauração hidrostática*). Esta tração deve ser mantida ao longo de todo seu comprimento a fim de evitar a desconexão no fundo do mar. Seu casco é semelhante ao casco da plataforma Semi-submersível.

A TLP (Fig 5) permite que o uso da completação dos poços seja do tipo 'seca', isto é, o controle e intervenção nos poços é feito na plataforma e não no fundo do mar. Desta forma torna-se desnecessária a utilização de embarcações com posicionamento dinâmico para a intervenção nos poços, o que ocorre quando é utilizada a completação 'molhada' em que as árvores de natal ficam no fundo do mar.

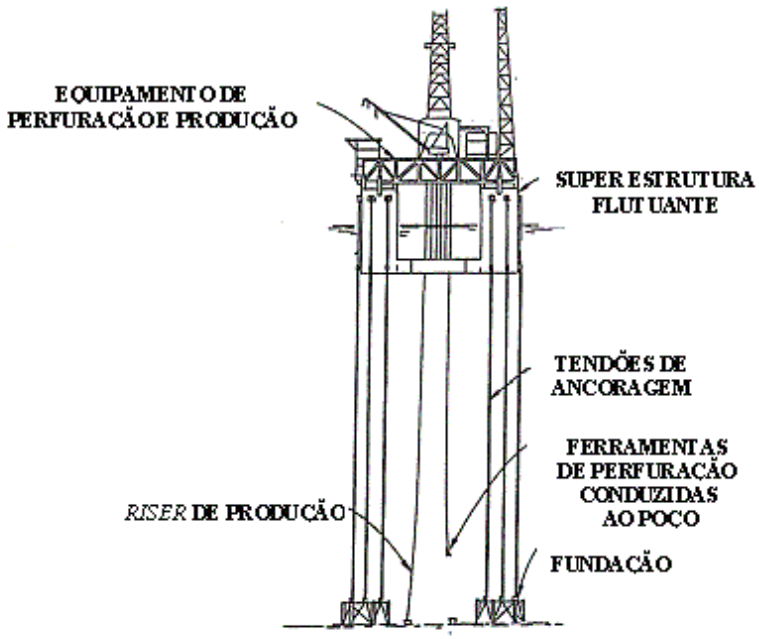


Figura 5 – TLP

- **Spar-buoy**

O sistema Spar consiste de um único cilindro vertical de aço de grande diâmetro, ancorado, operando com um calado de profundidade constante de cerca de 200 metros, o que gera apenas pequenos movimentos verticais e, conseqüentemente, possibilita a adoção de risers rígidos de produção. Neste tipo de plataforma há utilização de supressores de vórtices em torno do cilindro com o objetivo de inibir vibrações induzidas pelo fenômeno de *vortex shedding* decorrente principalmente pelas correntes marinhas.

Conforme pode ser observado nos exemplos citados, os sistemas flutuantes estão mais susceptíveis às ações dinâmicas, originadas das ações ambientais, do que as plataformas fixas. Também devido à elevada complacência destes sistemas, as ferramentas numéricas tiveram que se desenvolver de forma a considerar não só os efeitos dinâmicos como também os efeitos não lineares devido a grandes deslocamentos a que a unidade está sujeita.

2.2. RISERS

Para transportar o óleo do fundo do mar até às unidades flutuantes ou fixas são necessárias tubulações que devem ser analisadas cuidadosamente. Estes tubos suspensos, geralmente dispostos em configurações em catenária, recebem a denominação de risers (fig 6) e podem ser flexíveis, formados por camadas alternadas de plástico e aço, ou rígidos, constituídos de aço.

Figura 6 – Riser Flexível / Riser Rígido

Para o dimensionamento e verificações estruturais dos risers conectados à unidades flutuantes também são necessárias ferramentas específicas que considerem o efeito dinâmico e não linear dos movimentos impostos por estas unidades ao topo dos risers, além do efeito das ondas e correnteza agindo diretamente sobre estas linhas.

Os risers flexíveis podem assumir diferentes configurações em catenária como "Steep Wave" e "Lazy Wave" (Figura 7). Estas configurações possuem seções intermediária com flutuadores, cujo empuxo alivia o peso suportado pelo sistema flutuante e, quando sob solicitação lateral, contribui com momentos restauradores.

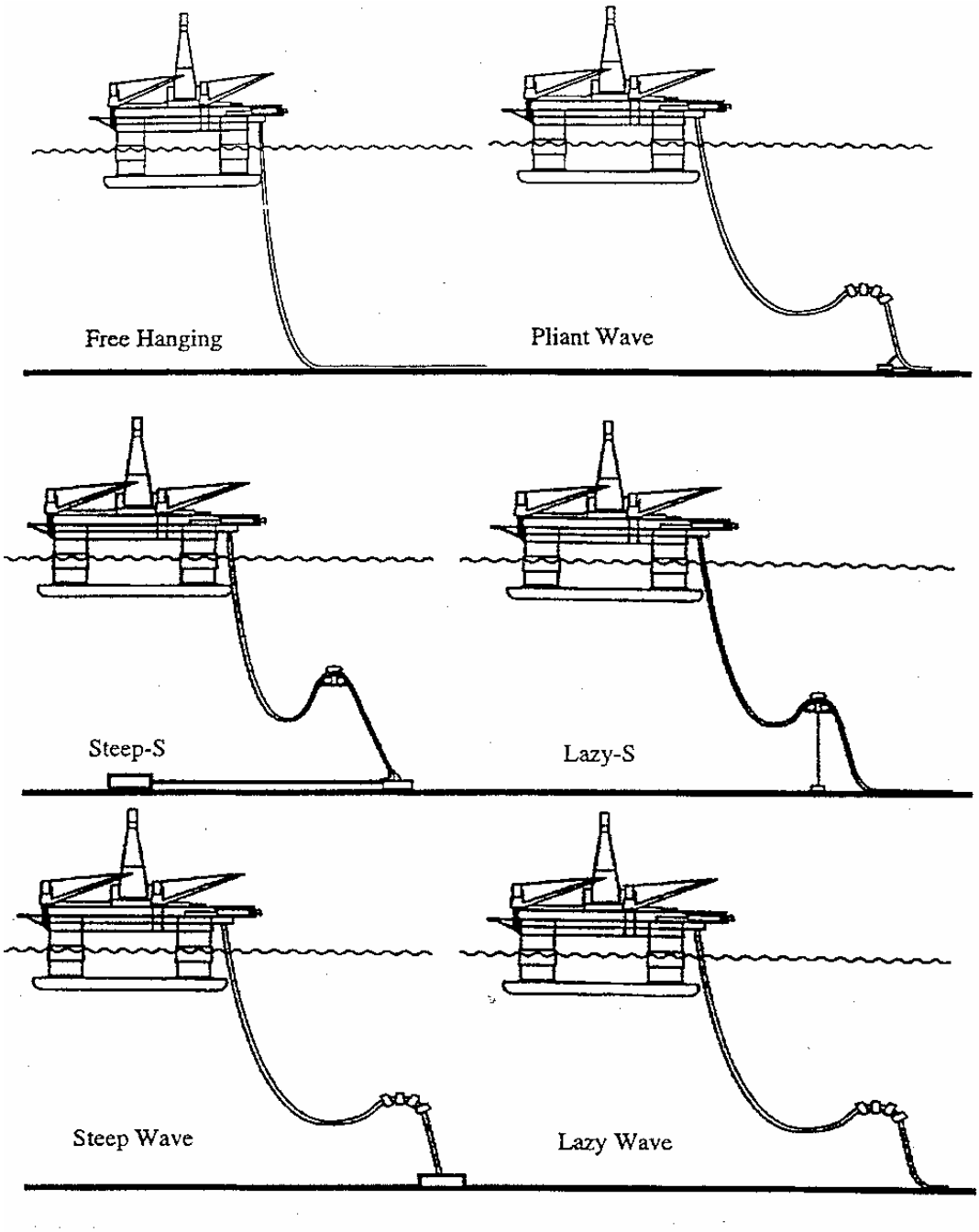


Figura 7 – Configuração de Riser Flexível

2.3. LINHAS DE ANCORAGEM

As linhas de ancoragem têm a função estrutural de fornecer forças de restauração para manter em posição os sistemas flutuantes tais como plataformas semi-submersíveis ou navios. Para oferecer a força de restauração necessária são dispostas em catenária (ancoragem convencional) ou utilizadas como linhas retesadas (*taut-leg*) ou tendões.

- **Ancoragem Convencional**

Denomina-se ancoragem convencional a ancoragem em catenária (Figura 8). Esta técnica de ancoragem é utilizada em operações de produção ou perfuração. A ancoragem em catenária mantém a unidade flutuante em uma locação através da força de restauração das linhas. As linhas ancoradas são presas ao fundo do mar por âncoras de resistência horizontal. A força de restauração está relacionada com vários parâmetros, um deles é o raio de ancoragem, um dos principais problemas deste tipo de ancoragem.

Para atender os critérios de projeto para passeio das unidades flutuantes ancoradas (10% da lâmina d'água) tem-se a necessidade de ter um raio de ancoragem razoavelmente grande. Conseqüentemente em um campo de exploração de petróleo isto gera um congestionamento de linhas de unidades próximas, interferindo diretamente no posicionamento das mesmas, juntamente com equipamentos submarinos.

Semi-submersíveis e Navios FPSO's podem ser ancorados com sistema convencional. Os navios FPSO's com ancoragem convencional utilizam o sistema SPM (*Single Point Mooring*). Este sistema de ancoragem é composto por um ponto simples de ancoragem do tipo *Turret* interno ou externo. O *Turret* permite que o navio gire livremente ao redor das linhas de ancoragem e risers e fique orientado na direção das cargas ambientais, reduzindo a atuação destas na estruturas.

- **Taut-leg**

A ancoragem Taut-Leg (Figura 8) é constituída por linhas retesadas com um ângulo de topo de aproximadamente 45° com a vertical. Conseqüentemente, tem-se uma projeção horizontal menor do que a ancoragem convencional, com relação a mesma ordem de grandeza da lâmina d'água.

As linhas da ancoragem Taut-Leg são constituídas nas suas extremidades por cabos de aço ou amarras e no seu trecho intermediário por cabo de poliéster. Esta configuração de linha pode ser a mesma adotada para ancoragem convencional.

As linhas da ancoragem Taut-Leg são fixas na suas extremidades inferiores por meio de estacas de sucção, VLA (âncoras com resistência vertical) ou estacas de fundeio. A ancoragem Taut-Leg é geralmente empregada em plataformas Semi-submersíveis e navios FPSO's.

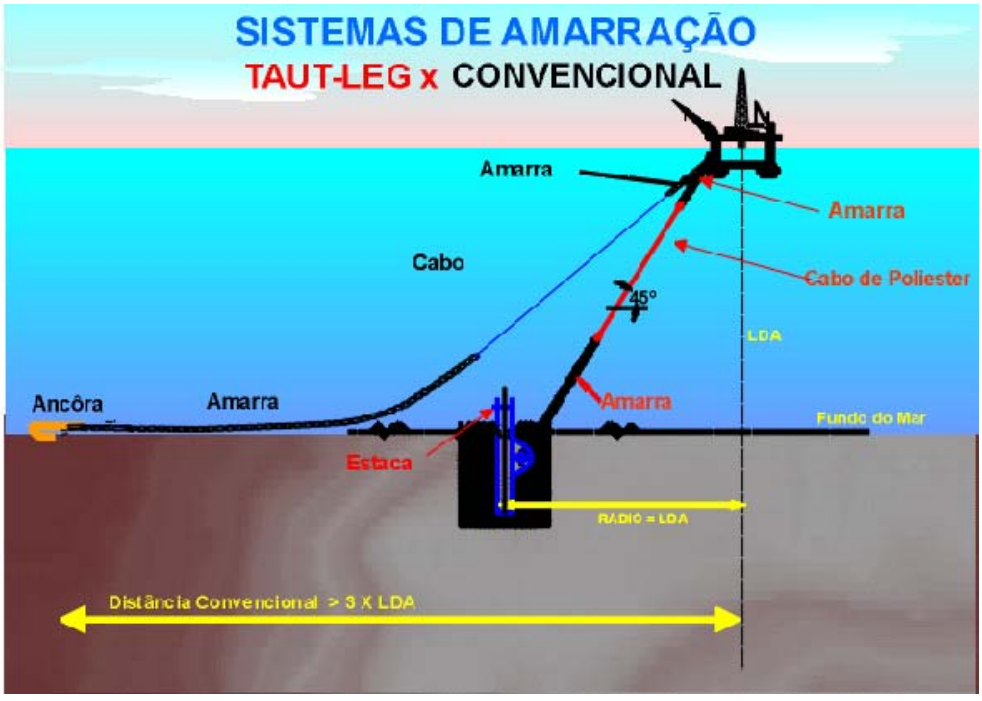


Figura 8 – Sistema de ancoragem Taut-Leg x Convencional

- **Tendões**

Os tendões podem ser de cabo de aço ou material sintético, proporcionando alta rigidez no plano vertical e baixa rigidez no plano horizontal. A força de restauração no plano horizontal é fornecida pela componente horizontal da força de tração nos tendões.

Para tendões de pequenos diâmetros (d ≅ 0.25 m), os efeitos de flexão podem ser desprezados enquanto que para grandes diâmetros (d ≅ 1.00 m) os efeitos de flexão devem ser considerados.

Este tipo de ancoragem baseia-se na utilização de tendões verticais, que precisam estar sempre tracionados devido ao excesso de empuxo proveniente da parte submersa da embarcação. Este tipo de ancoragem é usado principalmente em plataformas tipo TLP (*Tension Leg Plataform*), mas também pode ser adotada por bóias, monobóias, entre outras.

- **Dicas**

Atualmente a PETROBRAS vem empregando o sistema DICAS [2] de ancoragem, o qual consiste basicamente de um sistema de amarração disperso com diferentes resistências na proa e na popa do navio. A diferença básica entre o sistema DICAS e um SPM é com relação ao alinhamento. No sistema SPM o navio se alinha completamente com a direção das cargas ambientais enquanto que no sistema DICAS isto ocorre parcialmente. O sistema DICAS por dispensar o “turret” é um sistema mais simples sob o ponto de vista de construção.

3. EQUAÇÕES DE MOVIMENTO DA UNIDADE FLUTUANTE

3.1. INTRODUÇÃO

O procedimento de análise acoplada é baseado na integração numérica no domínio do tempo das equações de movimento de corpo rígido da plataforma.

Nesta seção, apresenta-se de forma resumida a formulação das equações de movimento de grande amplitude que representam os movimentos de corpo rígido da plataforma; maiores detalhes podem ser encontrados em Meirovich [3]. Apresenta-se também o procedimento de integração no tempo para a solução destas equações.

A formulação apresentada nesta seção tem origem na análise de movimento no domínio do tempo seguindo o enfoque desacoplado tradicional, mas será estendida também para a incorporação no modelo acoplado, como será comentado em seções posteriores.

As equações de movimento consideram efeitos não lineares geométricos decorrentes de grandes deslocamentos do corpo; além disso, outros efeitos não-lineares são considerados na formulação do modelo acoplado, relacionados ao comportamento hidrodinâmico na interação fluido-estrutura (incluindo a força de arrasto viscosa, função quadrática da velocidade relativa entre o fluido e o corpo), e à interação não-linear com as linhas de ancoragem e *risers* modelados por elementos finitos, configurando o modelo acoplado.

3.2. SISTEMAS DE REFERÊNCIA

***Sistema Global Geral* (Constante, "inercial")**

Inicialmente, define-se o sistema de coordenadas **global geral** (*x,y,z*) primário, único, ao qual estarão referenciados todos os demais sistemas (figura 9). Os eixos *x* e *y* deste sistema global estão contidos em um plano horizontal, e o eixo *z* corresponde à direção vertical, orientado de baixo para cima. Em princípio, a profundidade da origem deste sistema de referência global geral a partir do nível de águas tranqüilas, pode ser

definida através de uma variável h, mas usualmente é mais conveniente fazer com que a origem esteja contida no plano da superfície média da água, ou seja, $h = 0$.

As coordenadas dos nós da malha de elementos finitos que representam as linhas no modelo acoplado são expressas neste sistema global geral (eventualmente também referido como o sistema "inercial").

Sistema Local das Ondas

Em seguida define-se um sistema de coordenadas (ξ,η,ζ) para descrever os movimentos da onda. Neste sistema são calculadas as velocidades, acelerações e pressões do fluido induzidas pela onda. O plano $\xi\zeta$ está na superfície média do mar, e o eixo η é vertical. O eixo-ξ é paralelo à direção de propagação da onda, e faz um ângulo β com o eixo x-global (positivo no sentido horário de x para ξ).

Desta forma, as seguintes expressões podem ser empregadas para transformar as coordenadas de um ponto do sistema global (x,y,z) para o sistema da onda:

$$\begin{aligned} \xi &= x\cos\beta - y\,\mathrm{sen}\,\beta \\ \eta &= z + \mathrm{h} \\ \zeta &= x\,\mathrm{sen}\,\beta + \mathrm{y}\cos\beta \end{aligned} \qquad (1)$$

Sistema Estrutural da Plataforma (Móvel, "Fixo no Corpo")

Define-se também um sistema de coordenadas **estrutural** (X,Y,Z), específico da unidade flutuante. Trata-se de um sistema **móvel** que acompanha os movimentos do corpo. As equações de movimento do corpo, apresentadas mais adiante, são escritas neste sistema e expressam a posição e os movimentos do sistema móvel (X,Y,Z) em relação ao sistema global (x,y,z).

A origem deste sistema de referência estrutural do corpo está localizada no seu centro de gravidade (CG). Inicialmente, os eixos têm orientação semelhante à do sistema global geral (x,y,z), ou seja, inicialmente o plano XY está contido em um plano horizontal e o eixo Z é vertical, orientado de baixo para cima.

A definição da posição inicial da origem do sistema estrutural (X,Y,Z), em relação ao sistema global geral (x,y,z), é feita através de três valores que definem a

distância de sua origem (o CG) até a origem do sistema de global geral, e de um quarto valor que representa o ângulo, em graus, que define o aproamento da unidade flutuante. Este ângulo é medido no plano horizontal, entre o eixo *X*-global geral e o eixo *x*-estrutural da unidade.

Sistema Local dos Membros da Plataforma e dos Elementos Finitos das Linhas e Risers

Finalmente, cada elemento da malha de elementos finitos para as linhas, e cada membro reticulado da plataforma tem seu próprio sistema de referência **local** ($\bar{x},\bar{y},\bar{z}$). As propriedades dos elementos ou membros devem ser fornecidas neste sistema local. A origem deste sistema está localizada no nó 1 do membro. A direção local $\bar{x}$ coincide com o eixo do membro, e é orientada do nó 1 para o nó 2; as direções locais $\bar{y}$ e $\bar{z}$, ortogonais a $\bar{x}$, estão contidas na seção transversal do membro. Para um membro com orientação geral no espaço, a direção local $\bar{y}$ é horizontal, e a direção local $\bar{z}$ é perpendicular às direções $\bar{x}$ e $\bar{y}$. Para um membro horizontal, a direção local $\bar{z}$ é vertical, paralela ao eixo global z. Para um membro vertical, a direção local horizontal $\bar{y}$ é paralela ao eixo global y. Por sua vez, a direção local $\bar{z}$, também horizontal, é paralela ao eixo global x (mas em sentido contrário).

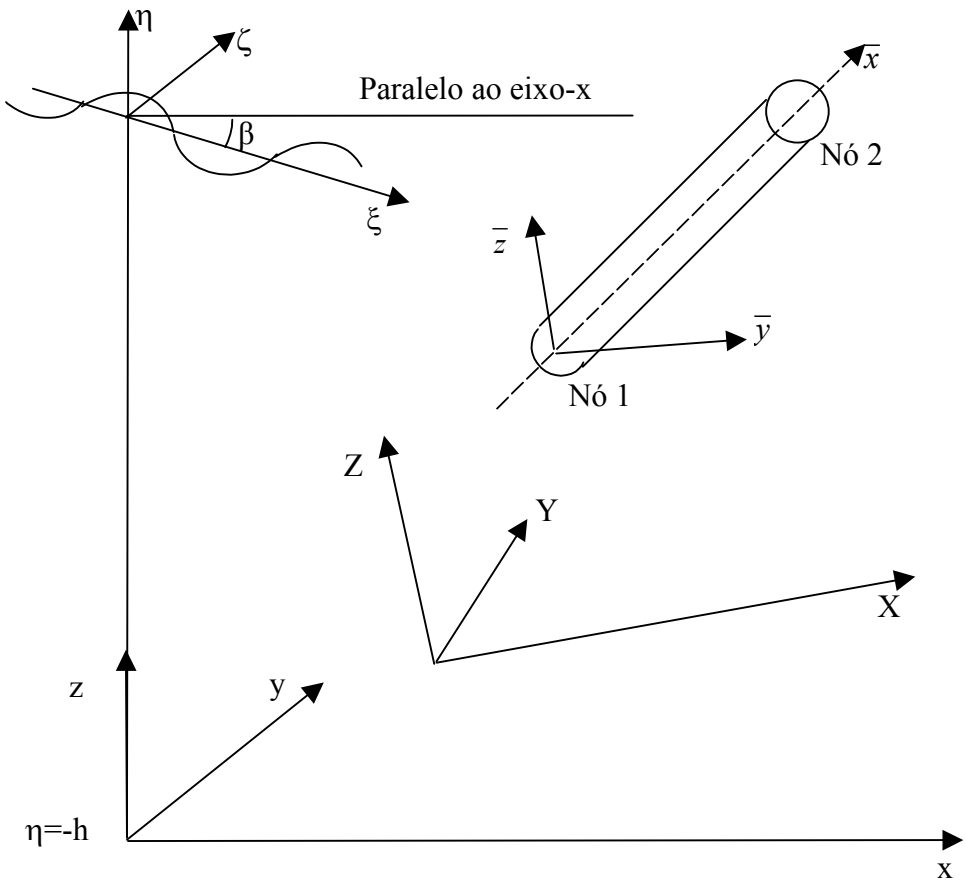


Figura 9 – Sistemas de Referência

3.3. FORMULAÇÃO DAS EQUAÇÕES DE MOVIMENTO

No raciocínio que se segue, vamos supor que o aproamento da plataforma em relação ao sistema global (*x,y,z*) é zero, ou seja, que o sistema estrutural da plataforma (*X,Y,Z*) e o sistema global (*x,y,z*) são originalmente paralelos. A extensão para casos mais gerais com aproamento diferente de zero é trivial.

O deslocamento do corpo pode ser expresso como o somatório de uma translação da origem do sistema de coordenadas estrutural da plataforma, e uma rotação em torno de um eixo passando pela origem do sistema estrutural:

Deslocamento de Translação

A translação $\mathbf{x}_l(t)$ é expressa pela variação da origem do sistema estrutural da plataforma (*X,Y,Z*), ou seja, a variação da posição do centro de gravidade (CG), medida em relação ao sistema global (*x,y,z*). As componentes de $\mathbf{x}_l$ são $x_{l1}(t)$, $x_{l2}(t)$, $x_{l3}(t)$.

Deslocamento de Rotação

De forma similar, o movimento de rotação é a variação angular dos eixos do sistema estrutural em relação ao sistema global. Para expressar a posição relativa rotacional desses dois sistemas de referência, empregam-se os *ângulos de Euler*, denominados γ, β, α. A sequência de rotações que define estes ângulos é descrita a seguir e ilustrada na Figura 10.

- Assume-se que originalmente o sistema da plataforma (X,Y,Z) e o sistema global (x,y,z) são coincidentes.
- Inicialmente, a plataforma gira em torno do seu eixo-Z através do ângulo de yaw γ.
- Em seguida, a partir da posição resultante, gira em torno do eixo-Y através do ângulo de pitch β;
- Finalmente, a partir desta última posição, gira em torno do eixo-X através do ângulo de roll α.

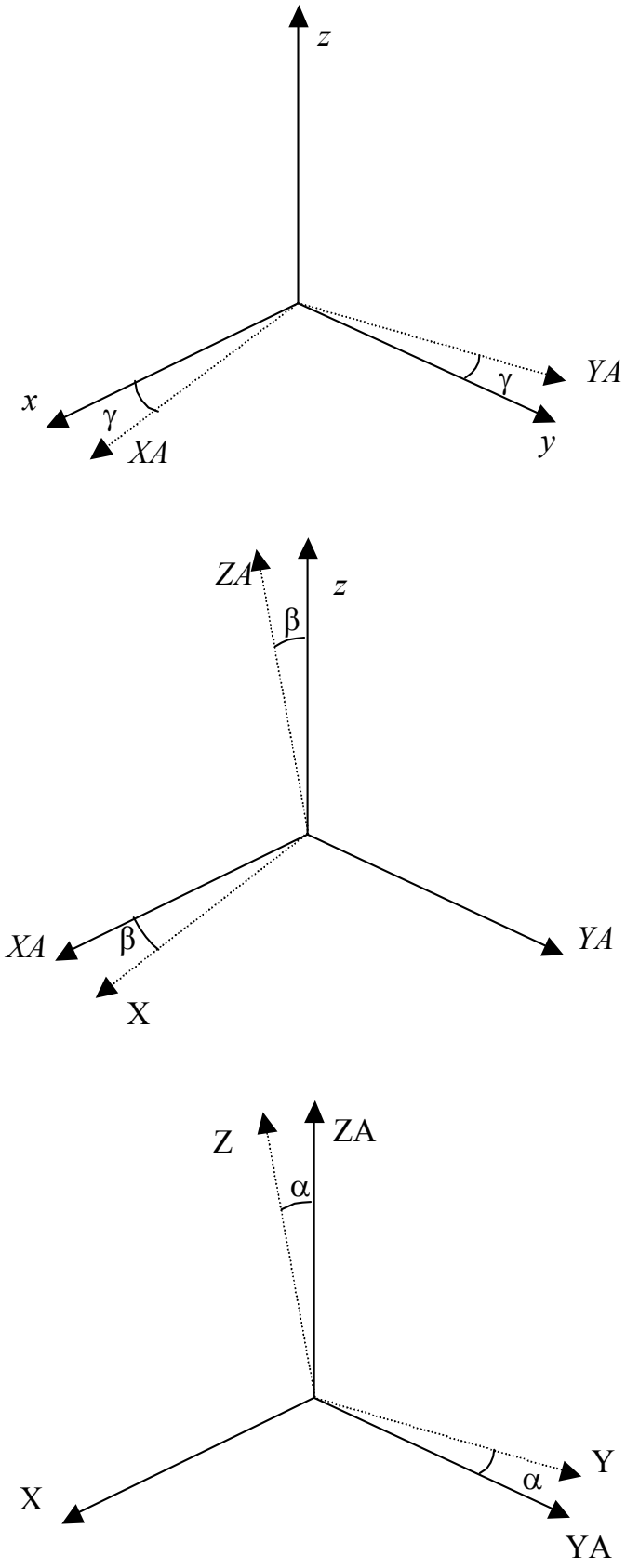


Figura 10 – Transformação do Sistema de Coordenadas

Transformação de Coordenadas

Após o movimento do corpo, a seguinte expressão relaciona as coordenadas de um ponto, expressas no sistema estrutural da plataforma $\mathbf{X} = (X,Y,Z)$, com as coordenadas do mesmo ponto expressas no sistema global $\mathbf{x} = (x,y,z)$, em função do movimento de translação $\mathbf{x}_l = (x_{l1}, x_{l2}, x_{l3})$ e do movimento de rotação definido pelos ângulos de Euler γ, α, β:

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} cos\beta cos\alpha & sen\beta cos\alpha & -sen\alpha \\ -sen\beta cos\gamma + cos\beta sen\alpha sen\gamma & cos\beta cos\gamma + sen\beta sen\alpha sen\gamma & cos\alpha sen\gamma \\ sen\beta sen\gamma + cos\beta sen\alpha cos\gamma & -cos\beta sen\gamma + sen\beta sen\alpha cos\gamma & cos\alpha cos\gamma \end{bmatrix} \begin{Bmatrix} x - x_1 \\ y - x_2 \\ z - x_3 \end{Bmatrix} \quad (2)$$

A equação (2) define a transformação geral rotacional de coordenadas, relacionando o sistema de coordenadas globais fixo no espaço com o sistema de coordenadas fixo na plataforma. Esta relação pode ser reescrita na forma seguinte, onde, com a exceção de **X**, as letras maiúsculas e em negrito representam matrizes e as letras minúsculas e em negrito representam vetores:

$$\mathbf{X} = \mathbf{A}\,(\mathbf{x} - \mathbf{x_1}) \quad (3)$$

Na equação (3), X é o vetor cujas componentes são (X, Y, Z), x e x_1 são similarmente definidas em termos das coordenadas globais do ponto em consideração e as translações das coordenadas de origem, e A é a matriz de rotação de coordenadas 3x3. Observando-se que A é uma transformação ortogonal, sua inversa é igual à sua transposta, e a transformação inversa será dada por:

$$\mathbf{x} = \mathbf{x_1} + \mathbf{A}^T\mathbf{X} \quad (4)$$

Transformação de Velocidades

Considerando que ω seja o vetor de velocidades angulares do corpo expressa em termos de ω_1, ω_2, ω_3 em torno dos eixos xyz, a relação entre ω e a derivada no tempo dos ângulos de Euler é dada pela equação (5):

$$\omega = B\,\dot{\theta} \quad (5)$$

onde $\dot{\theta} = \{\dot{\gamma}, \dot{\alpha}, \dot{\beta}\}$ e B é definido:

$$B = \begin{bmatrix} 1 & 0 & -sen\ \alpha \\ 0 & cos\ \gamma & sen\ \gamma.cos\ \alpha \\ 0 & -sen\ \gamma & cos\ \gamma.cos\ \alpha \end{bmatrix} \tag{6}$$

B, em geral, é uma matriz quadrada e não singular, portanto a sua inversa existe e assim a transformação inversa de (6) pode ser escrita como:

$$\dot{\theta} = B^{-1}\omega \tag{7}$$

Segunda Lei de Newton

A segunda lei de Newton para movimentos translacionais e rotacionais é dada pela equação (8):

$$f = \frac{d}{dt}(Mv) \tag{8}$$

$$m = \frac{d}{dt}(I\omega)$$

onde f e m são os vetores de força e momento externos, M e I são matrizes 3x3 compostas da massa do corpo, e seus momentos e produtos de inércia de acordo com as equações (9) e (10):

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \tag{9}$$

$$I = \begin{bmatrix} I_{11} & -J_{12} & -J_{13} \\ -J_{21} & I_{22} & -J_{23} \\ -J_{31} & -J_{32} & I_{33} \end{bmatrix} \tag{10}$$

Onde: m = massa da plataforma

I_{ii} = momento de inércia em torno dos eixos i;

$$I_{ii} = \int (x_j^2 + x_k^2)\ dm \qquad j,k \neq i \tag{11}$$

J_{ij} = ij-ésimo produto de inércia

$$J_{ij} = \int x_i x_j \; dm \qquad\qquad i \neq j \tag{12}$$

$J_{ij} = J_{ji}$.

O lado direito das equações (8) representa as derivadas no tempo do momento e do momento angular. Se a velocidade translacional, v, do centro de gravidade do corpo e o vetor de forças, f, são expressos em relação ao sistema de referência inercial, $oxyz$, a primeira das equações (8) torna-se:

$$f = M \frac{dv}{dt} \tag{13}$$

$$v = \frac{dx}{dt} \tag{14}$$

onde $x = \{x_1, y_1, z_1\}$ são as coordenadas do centro de gravidade do corpo em relação ao sistema de referência $oxyz$.

É conveniente que a equação de momento angular seja avaliada em relação ao sistema de referência móvel, que permanece fixo no corpo, visto que a matriz de inércia é constante neste sistema. A derivada no tempo do momento angular é, portanto, avaliada num sistema de coordenadas que está girando, assim a segunda das equações (8) torna-se:

$$m = I \frac{d\omega}{dt} + \omega \times (I\omega) \tag{15}$$

<u>Equações de Movimento – Forma Inicial</u>

As equações (7), (13), (14) e (15) podem ser rearranjadas e reescritas como:

$$\frac{dv}{dt} = M^{-1} f$$

$$\frac{dx}{dt} = v \tag{16}$$

$$\frac{d\omega}{dt} = I^{-1} [m - \omega \times (I\omega)]$$

$$\frac{d\theta}{dt} = B^{-1} \omega$$

As equações (16) podem ser vistas como um sistema de doze equações de primeira ordem nas variáveis v, x, ω e θ. Estas variáveis, dependentes do tempo, expressam as velocidades e posição do corpo. É importante ressaltar dois tipos de não linearidade que ocorrem nestas equações:

- Os vetores de força e momento, f e m, são funções não lineares da posição do corpo e do estado de movimento;
- O produto vetorial ω x $(I\omega)$ e a matriz de transformação B^{-1} contêm termos não lineares envolvendo, respectivamente, produtos e potências das velocidades angulares, e funções trigonométricas dos ângulos de Euler.

Neste ponto, formulações simplificadas poderiam assumir pequenas amplitudes de movimento e desprezar termos de ordem superior contendo produtos ou potências de quantidades de menor ordem de grandeza. No entanto, como mencionado anteriormente, a presente formulação mantém todos os termos não-lineares e, portanto, é válida para grandes amplitudes de movimento; isto será possível já que a integração das equações será feita no domínio do tempo.

3.4. SOLUÇÃO DAS EQUAÇÕES DE MOVIMENTO

Equações de Movimento – Forma Final

Para a integração no tempo das equações de movimento, será empregado o método de Runge-Kutta de quarta ordem. Este método pode operar sobre um sistema de equações diferenciais acopladas da forma $\frac{dy}{dt} = f(y,t)$, que, como pode ser visto, é similar às equações (16). Trata-se de um método baseado em extrapolações polinomiais da variável principal no intervalo de tempo seguinte, e na determinação dos coeficientes do polinômio a partir de valores estimados das derivadas em instantes ao longo do intervalo de tempo.

Mais adiante, será demonstrado que os vetores de força e momento, **f** e **m,** têm componentes que são proporcionais às acelerações do corpo (as parcelas de inércia da fórmula de Morison). Estas componentes irão gerar termos de massa adicionada, que variam ao longo do tempo. Separando as parcelas de **f** e **m** que dependem das acelerações, e que são afetados por termos de massa adicionada, pode-se rearranjar as equações (16) da seguinte forma:

$$M\frac{dv}{dt} = -A\frac{dv}{dt} - B\frac{d\omega}{dt} + f_1 \qquad (17)$$

$$I\frac{d\omega}{dt} = -C\frac{dv}{dt} - D\frac{d\omega}{dt} + m_1 - \omega \times (I\omega)$$

Onde:

A e D são as matrizes da massa adicionada no presente tempo de integração,

B e C são os termos cruzados de massa adicionada,

f_1 e m_1 são as parcelas dos termos de força e momento que dependem da posição, velocidade e tempo, mas são independentes da aceleração.

As equações (17) são mais uma vez rearranjadas:

$$(M+A)\frac{dv}{dt} + B\frac{d\omega}{dt} = f_1 \qquad (18)$$

$$C\frac{dv}{dt} + (I+D)\frac{d\omega}{dt} = m_1 - \omega \text{ x } (I\omega),$$

que pode ser reescrita como,

$$\begin{bmatrix} M+A & B \\ C & I+D \end{bmatrix} \begin{Bmatrix} \frac{dv}{dt} \\ \frac{d\omega}{dt} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ m_1 - \omega \text{ x } (I\omega) \end{Bmatrix} \tag{19}$$

onde a matriz de massa global é dada por,

$$\overline{A} = \begin{bmatrix} M+A & B \\ C & I+D \end{bmatrix} \tag{20}$$

$\overline{A}$ é uma matriz simétrica e, em geral, não singular, assim a sua inversa pode ser expressa, numa forma particionada, como,

$$\overline{A}^{-1} = \begin{bmatrix} \overline{A}_{11} & \overline{A}_{12} \\ \overline{A}_{21} & \overline{A}_{22} \end{bmatrix} \tag{21}$$

Depois de pré multiplicar os dois lados de (19) por $\overline{A}^{-1}$, obtém-se,

$$\frac{dv}{dt} = \overline{A}_{11} f_1 + \overline{A}_{12} [\, m_1 - \omega \text{ x } (I\omega)] \tag{22}$$

$$\frac{d\omega}{dt} = \overline{A}_{21} f_1 + \overline{A}_{22} [\, m_1 - \omega \text{ x } (I\omega)]$$

As equações (22) encontram-se agora numa forma apropriada para a aplicação do algoritmo de Runge-Kutta, uma vez que o lado direito destas equações não possui mais termos com derivadas.

Método de Runge Kutta de Quarta Ordem

Este método é comumente utilizado na solução deste tipo de problema. Ele emprega uma função de extrapolação do tipo polinomial, expressando os coeficientes do polinômio em termos dos valores estimados das derivadas em pontos intermediários no interior e no contorno do intervalo de integração. A seguir uma breve descrição do algoritmo de Runge-Kutta:

Consideremos N equações diferenciais ordinárias de 1ª ordem:

$$\frac{dx_1}{dt} = f_i = (x_1, x_2, \ldots, x_N, t),\ i = 1, 2, \ldots, N \tag{23}$$

As soluções de (23) podem ser extrapoladas em um intervalo de tempo $t_1 \le t \le t_1 + h$ por uma aproximação polinomial de quarta ordem:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \tag{24}$$

As derivadas desta aproximação polinomial nos pontos t = 0, t = h e no ponto médio, t = h/2, do intervalo de extrapolação são:

$$\dot{x}(0) = a_1;$$

$$\dot{x}\left(\frac{h}{2}\right) = a_1 + a_2 h + \frac{3}{4} a_3 h^2 + \frac{1}{2} a_4 h^3; \tag{25}$$

$$\dot{x}(h) = a_1 + 2\, a_2 h + 3\, a_3 h^2 + 4\, a_4 h^3$$

O incremento em x de intervalo h é dado pela substituição em (24) como:

$$x(h) - x(0) = h\,(a_1 + a_2 h + a_3 h^2 + a_4 h^3) \tag{26}$$

Isto pode ser mostrado pela substituição:

$$\frac{1}{6}\left[\dot{x}(0) + 4\,\dot{x}\left(\frac{h}{2}\right) + \dot{x}(h)\right] = a_1 + a_2 h + a_3 h^2 + a_4 h^3 \tag{27}$$

Substituindo (27) em (26), a extrapolação apresenta-se:

$$x(h) = x(0) + \frac{h}{6}\left[\dot{x}(0) + 4\,\dot{x}\left(\frac{h}{2}\right) + \dot{x}(h)\right] \tag{28}$$

Ou, em termos mais gerais:

$$x(t+h) = x(t) + \frac{h}{6}\ [\ \dot{x}\ (t) + 4\ \dot{x}\left(t+\frac{h}{2}\right) + \dot{x}\ (t+h)\ ] \qquad (29)$$

Os termos que aparecem entre colchetes são derivadas de x(t) em 3 pontos do intervalo de extrapolação. Observando-se que a função é desconhecida neste intervalo torna-se necessário estimar-se valores para os termos entre colchetes. Uma estimativa é obtida pela modificação de (29):

$$x(t+h) = x(t) + \frac{h}{6}\ [\ \dot{x}_1 + 2\dot{x}_2 + 2\dot{x}_3 + \dot{x}_4] \qquad (30)$$

Nesta expressão, as derivadas são calculadas no início do tempo presente utilizando-se valores de variáveis do intervalo de tempo anterior. Assim, duas estimativas são obtidas para o ponto médio do intervalo de tempo e uma estimativa de derivada é obtida no final do intervalo de tempo. As expressões para estas estimativas são apresentadas a seguir:

$x_1 = x(t)$ no início do intervalo;

$\dot{x}_1 = f\ (x_1,t)$;

$x_2 = x_1 + \frac{h}{2}\ f(x_1,t) = x_1 + \frac{h}{2}\ \dot{x}_1$;

$\dot{x}_2 = f\ (x_2,t + \frac{h}{2})$; (Primeira estimativa de derivada em $\frac{h}{2}$); (31)

$x_3 = x_1 + \frac{h}{2}\ \dot{x}_2$;

$\dot{x}_3 = f\ (x_3,t + \frac{h}{2})$; (Segunda estimativa de derivada em $\frac{h}{2}$);

$x_4 = x_1 + h\ \dot{x}_3$;

$\dot{x}_4 = f\ (x_4,t + h)$.

O início de um método de integração deste tipo necessita apenas de valores de variáveis no tempo t = 0.

Algumas vezes ocorrem efeitos transientes severos associados ao conjunto de condições iniciais fornecidas. Estes efeitos podem persistir por um prolongado período durante a integração, dependendo do grau de amortecimento e outras características do sistema. O transiente pode ser reduzido por dois procedimentos:

1. Iniciando-se a análise dinâmica a partir dos resultados de uma análise estática anterior, na qual foram aplicadas as parcelas estáticas do carregamento;
2. Na análise dinâmica, introduzindo-se uma função rampa para as componentes dinâmicas dos carregamentos, de modo que estes sejam aplicados de forma integral apenas após um certo período de tempo t_0.

A função rampa empregada é da forma:

$$c(t) = \frac{1}{2}\left(1 - \cos\frac{\pi t}{t_0}\right); \text{ para } t \leq t_0 \qquad (32)$$

$$1.0; \text{ para } t > t_0$$

4. FORMULAÇÃO ESTRUTURAL DAS LINHAS

4.1. INTRODUÇÃO

Prosseguindo na descrição do procedimento de análise acoplada de plataformas flutuantes apresenta-se a seguir a formulação e o procedimento de solução do modelo matemático que representa o comportamento estrutural das linhas de ancoragem e risers.

Assim, nesta seção apresenta-se de forma resumida a formulação do modelo matemático que representa o comportamento dinâmico de sistemas estruturais, particularmente para o caso de estruturas esbeltas como as linhas de ancoragem e risers.

4.2. MODELO MATEMÁTICO; SOLUÇÃO NUMÉRICA

O comportamento dinâmico de uma estrutura pode ser descrito matematicamente por um problema de valor inicial e de contorno (**PVI/C**), constituído por um sistema de equações diferenciais parciais (**EDP**) hiperbólicas - as *Equações de Movimento* ou *equações de equilíbrio dinâmico*. Na montagem deste sistema estão incorporadas as *Equações constitutivas* relacionando tensões às deformações, e as *Equações deformação-deslocamento*.

A essas equações diferenciais parciais, está associado um conjunto de condições de contorno no espaço, que estabelece que o contorno da estrutura está dividido em uma região com deslocamentos conhecidos e outra com forças conhecidas. Além disso, está associado também um conjunto de condições iniciais no tempo, que estabelece que os deslocamentos e velocidades em qualquer ponto do domínio espacial têm valores conhecidos no tempo $t = 0$.

A construção deste modelo matemático diferencial está baseada em conceitos da Mecânica do Contínuo e da Teoria da Elasticidade. Usualmente, no procedimento de solução do problema estrutural o modelo matemático é reescrito em uma formulação integral, baseada em princípios variacionais. Esta formulação integral pode ser obtida

de diversas maneiras: através de princípios de energia, como o Princípio dos Trabalhos Virtuais ou o Princípio da Energia Potencial Estacionária; ou através do método de Galerkin, baseado na técnica de resíduos ponderados.

Uma descrição detalhada do estabelecimento e formulação destes modelos matemáticos pode ser encontrada em diversos textos [4, 5, 6]. Uma descrição mais concisa e didática pode ser encontrada no trabalho de Kayser Junior [7].

Para a solução do problema descrito por estes modelos matemáticos contínuos, que acarreta na obtenção da resposta dinâmica desejada, são empregados métodos numéricos que efetuam discretizações no espaço e no tempo. O processo usual consiste em efetuar as discretizações de forma independente (*semi-discretização*), em duas etapas:

A) Em uma primeira etapa, utiliza-se uma técnica para a discretização *espacial* do domínio. Em formulações diferenciais, as **EDP** seriam então transformadas em um sistema de equações diferenciais ordinárias (**EDO**) *semi-discretas* (porém ainda funções contínuas do tempo).

B) Em uma segunda etapa, efetua-se a discretização das **EDO** *no tempo*, obtendo-se a resposta através de um algoritmo de integração.

4.3. DISCRETIZAÇÃO ESPACIAL

No contexto da análise de estruturas esbeltas, especificamente de risers e linhas de ancoragem de plataformas flutuantes, a técnica de discretização empregada é o Método dos Elementos Finitos – **MEF**. A formulação do **MEF**, que tem sido extensivamente estudada ao longo das três últimas décadas, não será descrita neste texto; recomenda-se a leitura de referencias clássicas da literatura, tais como [4,5,6]. Novamente, uma descrição concisa e didática pode ser encontrada em [7].

Para a discretização espacial dos risers e linhas de ancoragem, o programa Prosim emprega elementos reticulados de treliça e pórtico. A seguir, apresenta-se uma descrição sucinta das características destes elementos.

4.3.1. Elemento de Treliça

Os elementos de treliça possuem 3 graus de liberdade por nó. Os graus de liberdade (U,V,W) representam movimentos lineares nas direções $\bar{x}$, $\bar{y}$ e $\bar{z}$, como ilustra a figura 11. Como este tipo de elemento não possui graus de liberdade angulares, conseqüentemente não é possível fornecer rigidez flexional. Por este motivo, estes elementos representam bem linhas que possuem baixa rigidez à flexão tais como linhas de ancoragem e umbilicais.

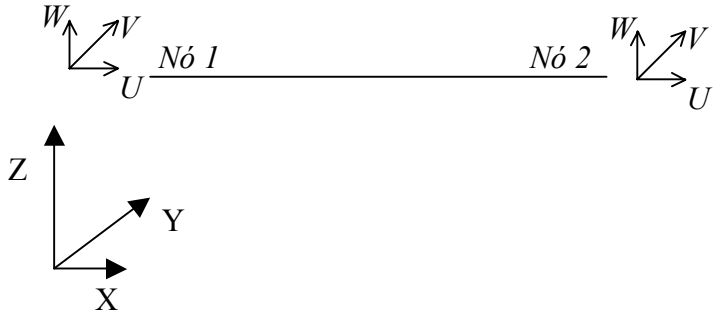


Figura 11 – Elemento de Treliça

4.3.2. Elemento de Pórtico

O programa Prosim emprega um elemento finito de pórtico espacial baseado em uma formulação corotacional [8, 9, 10]. O objetivo principal da formulação co-rotacional é separar os movimentos de corpo rígido dos movimentos que geram deformações. Com isso, obtém-se um elemento mais preciso, robusto e menos sensível à magnitude das rotações incrementais.

O elemento de pórtico espacial possui 6 graus de liberdade por nó. Os graus de liberdade (U,V,W,RU,RV,RW) representam movimentos lineares nas direções $\bar{x}$, $\bar{y}$ e $\bar{z}$ e movimentos angulares em torno destes mesmos eixos, como ilustra a figura 12. Com este tipo de elemento é possível considerar a rigidez à flexão das linhas, de modo a representar linhas cuja rigidez à flexão é representativa, tais como risers rígidos e risers flexíveis.

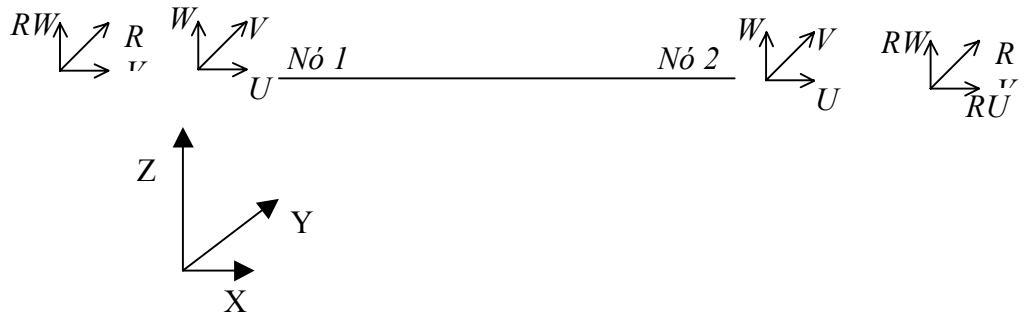


Figura 12 – Elemento de Pórtico Espacial

4.4. DISCRETIZAÇÃO NO TEMPO - SOLUÇÃO NUMÉRICA DE PROBLEMAS DINÂMICOS LINEARES

4.4.1. Formulação do Problema Dinâmico

Como resultado da aplicação do Método dos Elementos Finitos para a discretização espacial, o modelo matemático diferencial, originalmente um PVI/C composto por um sistema de equações diferenciais parciais (**EDP**) associado a um conjunto de condições de contorno no espaço e condições iniciais no tempo, se converteria em um problema de valor inicial composto por um sistema de equações diferenciais ordinárias (**EDO**) ***semi-discretas*** (discretizadas no *espaço*, mas ainda funções contínuas do *tempo*), e um conjunto de condições iniciais no tempo [11, 12].

Para problemas lineares, as EDO correspondem às *equações de movimento* escritas da seguinte forma:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{F}(t) \tag{33}$$

As incógnitas destas equações são os vetores $\mathbf{u}(t)$, $\dot{\mathbf{u}}(t)$ e $\ddot{\mathbf{u}}(t)$ contendo, respectivamente, componentes de deslocamentos, velocidades e acelerações para cada grau de liberdade da malha de elementos finitos empregada para efetuar a discretização espacial. O problema de valor inicial é composto por estas equações de movimento, associadas às seguintes condições iniciais:

$$\mathbf{u}(0) = \mathbf{u}_0 \quad ; \qquad \dot{\mathbf{u}}(0) = \mathbf{v}_0 \tag{34}$$

As três parcelas do lado esquerdo das equações de movimento representam, respectivamente, forças de inércia, amortecimento e forças elásticas. Estas forças

internas equilibram as forças externas no lado direito, que são representadas pelo vetor **F**(t) contendo as resultantes nodais das cargas.

Finalmente, **M**, **C** e **K** são as matrizes de massa, amortecimento e rigidez, simétricas e constantes no tempo. As matrizes de massa e rigidez podem ser deduzidas diretamente a partir da formulação de elementos finitos. A matriz de amortecimento **C**, por sua vez, é usualmente representada pela expressão de amortecimento de Rayleigh como uma combinação linear das matrizes de massa e rigidez [5]:

$$\mathbf{C} = \alpha_m \mathbf{M} + \alpha_k \mathbf{K}. \tag{35}$$

onde α_m e α_k são, respectivamente, coeficientes escalares proporcionais à massa e à rigidez, a ser determinados a partir de dois pares de valores (frequência *x* percentagem de amortecimento crítico).

4.4.2. Procedimento de Solução do Problema Dinâmico

Para a solução do problema de valor inicial composto pelas equações (33) e (34), utilizam-se algoritmos de integração no tempo. Para isso, inicialmente escreve-se uma forma discretizada no tempo das equações (33), onde os valores exatos $\ddot{\mathbf{u}}(t_{n+1})$, $\dot{\mathbf{u}}(t_{n+1})$ e $\mathbf{u}(t_{n+1})$ são substituídos por aproximações $\mathbf{a}_{n+1}$, $\mathbf{v}_{n+1}$ e $\mathbf{d}_{n+1}$

$$\mathbf{M}\,\mathbf{a}_{n+1} + \mathbf{C}\,\mathbf{v}_{n+1} + \mathbf{K}\,\mathbf{d}_{n+1} = \mathbf{F}_{n+1} \tag{36}$$

Assim, assume-se que o equilíbrio não será mais satisfeito a cada instante infinitesimal de tempo, mas apenas em um determinado número de instantes, separados por intervalos discretos de tempo.

Em seguida, utilizam-se operadores ou funções que, em um dado instante de tempo t_{n+1}, fornece aproximações $\mathbf{a}_{n+1}$, $\mathbf{v}_{n+1}$ e $\mathbf{d}_{n+1}$ a partir de aproximações obtidas em instantes anteriores. Em problemas discretizados no espaço pelo Método dos Elementos Finitos, é usual empregar a família de algoritmos de *Newmark*, que é caracterizada pelos seguintes operadores para fornecer as aproximações desejadas [5,6]:

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\,\mathbf{v}_n + \frac{\Delta t^2}{2}\left[(1-2\beta)\,\mathbf{a}_n + 2\beta\,\mathbf{a}_{n+1}\right] \tag{37}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\left[(1-\gamma)\,\mathbf{a}_n + \gamma\,\mathbf{a}_{n+1}\right] \tag{38}$$

Nestas expressões, γ e β são os parâmetros que caracterizam a família de algoritmos de Newmark. Por exemplo, a *regra trapezoidal* é um membro desta família,

caracterizado por estes operadores com os valores $\gamma = ½$ e $\beta = ¼$, e pela expressão discretizada (36).

Alternativamente, os operadores de Newmark podem ser escritos em termos de acelerações e velocidades:

$$\mathbf{a}_{n+1} = \frac{1}{\beta \Delta t^2} (\mathbf{d}_{n+1} - \mathbf{d}_n) - \frac{1}{\beta \Delta t} \mathbf{v}_n - \left(\frac{1}{2\beta} - 1\right) \mathbf{a}_n \qquad (39)$$

$$\mathbf{v}_{n+1} = \frac{\gamma}{\beta \Delta t} (\mathbf{d}_{n+1} - \mathbf{d}_n) - \left(\frac{\gamma}{\beta} - 1\right) \mathbf{v}_n - \left(\frac{\gamma}{2\beta} - 1\right) \Delta t \, \mathbf{a}_n \qquad (40)$$

Observa-se que a aplicação do algoritmo de integração leva a um sistema de três equações para as três incógnitas $\mathbf{a}_{n+1}$, $\mathbf{v}_{n+1}$, $\mathbf{d}_{n+1}$: as equações de movimento discretizadas (36), e os operadores (37) e (38), ou (39) e (40). Temos portanto duas opções para a implementação computacional, de acordo com a ordem em que são eliminadas as incógnitas:

- Implementação por Acelerações;
- Implementação por Deslocamentos.

A seguir será apresentada a implementação por acelerações e mais adiante, no capítulo específico de algoritmos de integração implícitos, a implementação por deslocamentos.

Implementação por Acelerações

Substituindo os operadores de Newmark (37) e (38) nas equações de movimento (36) resulta em:

$$[\mathbf{M} + \gamma \Delta t \, \mathbf{C} + \beta \Delta t^2 \, \mathbf{K}] \, \mathbf{a}_{n+1} =$$
$$= \mathbf{F}_{n+1} - \mathbf{C} \left[\mathbf{v}_n + (1 - \gamma) \Delta t \, \mathbf{a}_n\right] -$$
$$- \mathbf{K} \left[\mathbf{d}_n + \Delta t \, \mathbf{v}_n + \left(\frac{1}{2} - \beta\right) \Delta t^2 \, \mathbf{a}_n\right] \qquad (41)$$

Neste caso, as incógnitas do sistema efetivo são as acelerações. Observamos que, de acordo com o valor do parâmetro β, podemos identificar os seguintes casos:

- $\beta = 0$ e matrizes de massa e amortecimento diagonais: o sistema efetivo é desacoplado, o que quer dizer que não há necessidade de empregar uma técnica para resolução do sistema. As incógnitas são obtidas diretamente pela divisão dos termos do vetor de cargas efetivo pelos termos da diagonal da matriz efetiva. Esta característica identifica um algoritmo explícito.
- $\beta \neq 0$: o sistema efetivo é acoplado. Neste caso exige-se uma técnica para a resolução de sistemas de equações algébricas, e temos as características de um algoritmo implícito.

Com base nos critérios apresentados em [5], é possível concluir que para os problemas inerciais, a regra trapezoidal ou suas variações com amortecimento numérico (como os métodos αH-Newmark ou αB-Newmark que serão descritos mais adiante) são de fato os algoritmos mais adequados. Para chegar a esta conclusão podemos observar também os teoremas de Dahlquist [6]:

- Não existe um algoritmo explícito incondicionalmente estável.
- Não existe um algoritmo incondicionalmente estável com ordem de precisão maior ou igual a 3.
- O algoritmo incondicionalmente estável com ordem de precisão igual a 2 e com menor constante de erro é a regra trapezoidal.

Em problemas inerciais ou de dinâmica estrutural, a resposta dinâmica é dominada por modos de vibração de freqüências mais baixas. Neste caso deve-se integrar com precisão apenas os modos de freqüência mais baixa, sendo possível utilizar um valor maior para o intervalo de tempo. Assim, algoritmos implícitos incondicionalmente estáveis são mais apropriados para este tipo de problema, já que a restrição de estabilidade condicional de um algoritmo explícito requeriria o uso de valores muito pequenos de intervalos de tempo.

Em problemas de propagação de ondas, a resposta dinâmica é dominada por modos de vibração de freqüências intermediárias ou altas. Neste caso, os modos que devem ser integrados corretamente são os modos de freqüência mais alta. Para isso é necessário utilizar pequenos valores de intervalo de tempo Δt que podem ser próximos ao valor do intervalo de tempo crítico requerido pela restrição de estabilidade condicional. Conseqüentemente algoritmos explícitos, tal como o Método das

Diferenças Centrais, são mais apropriados a este tipo de problema, de modo a tirar partido de seu menor custo por instante de tempo. A discretização espacial é importante pois a malha deve ser refinada o suficiente para que as freqüências mais altas do modelo reproduzam adequadamente as freqüências correspondentes do problema físico real.

O capítulo 6 irá considerar especificamente o uso de algoritmos explícitos. Mais adiante, no capítulo 7, será apresentada a extensão do procedimento de solução para problemas dinâmicos não-lineares associado ao uso de algoritmos implícitos.

5. CARREGAMENTOS AMBIENTAIS

5.1. INTRODUÇÃO

Neste capítulo será apresentada a consideração dos carregamentos ambientais de onda, correnteza e vento nas análises dinâmicas do Prosim.

Em relação às ondas será apresentado o modelo para obtenção de velocidade, aceleração, deslocamento e pressão das partículas do fluido e sua transformação em forças através de modelos híbridos implementado no programa Prosim.

Em relação à correnteza será apresentado o procedimento para obtenção de forças a partir de velocidades de correnteza e ângulos de incidência.

Finalmente serão apresentadas formulações para obtenção de forças de vento a partir velocidade de vento, área exposta e outros fatores.

5.2. ONDAS

5.2.1. Modelo Matemático

O modelo matemático que representa o comportamento de ondas no mar é composto por um Problema de Valor de Contorno (PVC), que consiste em uma equação diferencial (Equação de Laplace) e as condições de contorno associadas. Para a formulação do modelo matemático, considera-se o sistema de coordenadas (ξ,ζ,η) descrito no item 3.2. Assume-se que o fundo do oceano é plano e com profundidade d (medida a partir do nível de águas tranqüilas) e que as ondas são bidimensionais no plano $\xi\eta$, periódicas, uniformes e progredindo na direção ξ positiva, são definidas em termos de sua altura H e período T. A incógnita básica do PVC é o potencial de velocidade do fluido Φ, a partir do qual, por derivação, podem ser obtidas as velocidades, acelerações, deslocamentos e pressões das partículas do fluido.

A seguir são apresentadas a Equação de Laplace e as condições de contorno. Uma descrição detalhada deste modelo pode ser encontrada em [13].

Equação de Laplace

$$\frac{\partial^2 \Phi}{\partial \xi^2} + \frac{\partial^2 \Phi}{\partial \eta^2} = 0 \tag{42}$$

Condições de Contorno

As duas condições de contorno na superfície livre, expressas em termos do potencial Φ, são:

- A condição de contorno dinâmica, que pode ser deduzida a partir da equação de Bernoulli, partindo da premissa que a pressão atmosférica fora da região do fluido é constante (como demonstrado em [14]):

$$\frac{\partial \Phi}{\partial t} + g\eta_s + \frac{1}{2}\left[\left(\frac{\partial \Phi}{\partial \xi}\right)^2 + \left(\frac{\partial \Phi}{\partial \eta}\right)^2\right] = 0 \qquad \text{em } \eta = \eta_s \tag{43}$$

  onde g é a aceleração da gravidade, e $\eta_s(\xi, t)$ é uma função que exprime a elevação da onda na superfície livre.

- A condição de contorno cinemática, que estabelece que uma partícula na superfície livre em um dado instante de tempo irá permanecer na superfície livre [14]:

$$\frac{\partial \eta_s}{\partial t} + \frac{\partial \Phi}{\partial \xi}\frac{\partial \eta_s}{\partial \xi} - \frac{\partial \Phi}{\partial \eta} = 0 \quad \text{em } \eta = \eta_s \tag{44}$$

Lembrando que o fundo do mar é assumido como plano e horizontal, a condição de contorno no fundo implica que a componente vertical da velocidade da partícula de fluido deve ser igual a zero.

$$\frac{\partial \Phi}{\partial \eta} = 0 \quad \text{em } \eta = -d \tag{45}$$

O problema de valor de contorno completo é portanto descrito pela equação de Laplace (42) e as três condições de contorno (43) a (45).

Este problema de valor de contorno é altamente não-linear, especialmente devido às condições de contorno de superfície livre. Desta forma, de modo geral não é possível obter uma solução analítica rigorosa para a equação diferencial de Laplace, e a solução (em termos de velocidades e acelerações das partículas fluidas induzidas pela onda) deve ser obtida introduzindo aproximações e/ou utilizando métodos numéricos. O procedimento mais usual, e que atende à prática de projeto de sistemas offshore,

consiste em empregar a Teoria Linear de Airy como o método numérico de solução da equação de Laplace.

5.2.2. Teoria Linear de Airy

A Teoria Linear de Airy está baseada na premissa de que a altura de onda é pequena comparada com o comprimento de onda ou a profundidade da água. Esta premissa permite que as condições de contorno de superfície livre sejam satisfeitas no nível médio de águas tranqüilas e não no nível real da elevação da onda. Para tanto, as condições de contorno são linearizadas, desprezando os termos de segunda ordem e de ordens superiores. Detalhes do procedimento de solução baseado na Teoria de Airy são apresentados em [13]. Como resultado, obtém-se a seguinte expressão:

$$\Phi(\xi,\eta,t) = \frac{\pi H}{k\,T}\frac{\cosh k(\eta+d)}{\operatorname{senh} kd}\cos(k\,\xi - w\,t - \theta) \tag{46}$$

Velocidades, Acelerações e Deslocamentos das Partículas do Fluido

Uma vez obtido o potencial de velocidade, as velocidades da partícula do fluido nas direções horizontal e vertical são obtidas diferenciando-se a equação (46) em relação a ξ e η:

$$\dot{u} = \frac{\partial\Phi}{\partial\xi} = \frac{\pi H}{T}\frac{\cosh k(\eta+d)}{\operatorname{senh} kd}\cos(k\,\xi - w\,t - \theta) \tag{47}$$

$$\dot{w} = \frac{\partial\Phi}{\partial\eta} = \frac{\pi H}{T}\frac{\operatorname{senh} k(\eta+d)}{\operatorname{senh} kd}\operatorname{sen}(k\,\xi - w\,t - \theta) \tag{48}$$

As acelerações da partícula de água nas direções horizontal e vertical são dadas por:

$$\ddot{u} = \frac{2\pi^2 H}{T^2}\frac{\cosh k(\eta+d)}{\operatorname{senh} kd}\operatorname{sen}(k\,\xi - w\,t - \theta) \tag{49}$$

$$\ddot{w} = \frac{2\pi^2 H}{T^2}\frac{\operatorname{senh} k(\eta+d)}{\operatorname{senh} kd}\cos(k\,\xi - w\,t - \theta) \tag{50}$$

Observando-se as expressões de velocidades horizontal e vertical, verifica-se que a velocidade horizontal da partícula de fluido é máxima (ou mínima) quando a velocidade vertical for zero e vice-versa. Como as amplitudes dessas duas velocidades

são geralmente diferentes, a partícula de fluido descreve uma trajetória elíptica sobre sua posição média, em um ciclo de onda completo.

Os deslocamentos da partícula de fluido a partir de sua posição média são obtidos pela integração de u̇ e v̇ em relação ao tempo t, aplicando-se a condição de contorno adequada para a constante de integração. Os deslocamentos nas direções horizontal e vertical, respectivamente u e w, são dados por:

$$u = -\frac{H}{2}\frac{\cosh k(\eta + d)}{senhkd} sen(k\xi - wt - \theta) \tag{51}$$

$$w = \frac{H}{2}\frac{senhk(\eta + d)}{senhkd} \cos(k\xi - wt - \theta) \tag{52}$$

O deslocamento vertical máximo, medido no nível de águas tranqüilas, é igual à amplitude da onda a = H/2.

Pressões

Finalmente, outro resultado de interesse é o campo de pressões no fluido. Tal resultado pode ser obtido através da aplicação da equação de Bernoulli [14]:

$$p = \rho g \eta + \rho \frac{\partial \phi}{\partial t} + \frac{1}{2}\rho (\nabla \phi)^2 \tag{53}$$

A primeira parcela desta expressão corresponde ao termo de pressão hidrostática. As demais parcelas correspondem às parcelas de primeira e segunda ordem da pressão dinâmica. De forma consistente com a expansão de primeira ordem do potencial de velocidade assumida pela teoria linear de Airy, a expressão da pressão dinâmica fica

$$p_{d1} = p = \rho g \frac{H}{2}\frac{\cosh ks}{\cosh kd} \cos(k\xi - wt) \tag{54}$$

Extrapolação de Wheeler

Conforme já observado, na Teoria Linear de Airy as condições de contorno de superfície livre são satisfeitas no nível médio de águas tranqüilas e não no nível real da elevação da onda.

Desta forma, em aplicações onde a altura de onda é significativa, o efeito de alteração da superfície livre sobre a força total induzida pela onda torna-se muito importante e, portanto, faz-se necessário algum tipo de aproximação para os pontos situados na superfície livre. Dentre os tipos de aproximações mais conhecidos destacam-se: extrapolação hiperbólica, linear e o método de extrapolação ou '*stretching*' de Wheeler [15], o qual é considerado para a parcela da onda acima da superfície livre.

O potencial de velocidade modificado pela proposição de Wheeler é apresentado a seguir:

$$\Phi(\xi,\eta,t) = \frac{ga}{\omega}\frac{\cosh\left[kd\left(\dfrac{d+\eta_s-h'}{d+\eta_s}\right)\right]}{\cosh kd} \tag{55}$$

onde h' = profundidade do ponto medido da superfície da onda.

<u>Resultados no Sistema Global de Coordenadas</u>

Na implementação computacional do programa Prosim, as propriedades do fluxo induzido pelas ondas são calculadas pela teoria de Airy no sistema de coordenadas bidimensional da onda ($\xi\eta\zeta$), e em seguida convertidas para o sistema tridimensional global (*xyz*).

O potencial de velocidade para águas profundas para o i-ésimo componente de onda é dado neste ponto por:

$$\Phi(x,y,z,t) = \frac{ga_i}{w_i}e^{k_i(z-h)}sen\left[k_i\left(x\cos\beta_i + zsen\beta_i\right) - w_i t - \theta_i\right] \tag{56}$$

Para todos os demais resultados (velocidades, acelerações e pressões) pode-se obter uma expressão no sistema global (*xyz*), procedendo-se de modo similar.

Por exemplo, as expressões para a transformação de coordenadas para as velocidades em oxyz são as seguintes:

$$\dot{u}_{(xyz)} = \frac{\partial\Phi}{\partial\xi}\cos\beta \tag{57}$$

$$\dot{v}_{(xyz)} = \frac{\partial\Phi}{\partial\eta} \tag{58}$$

$$\dot{w}_{(xyz)} = \frac{\partial \Phi}{\partial \xi} \text{ sen } \beta \quad (59)$$

5.2.3. Representação Espectral

O modelo matemático para a representação das ondas do mar que foi formulado em termos de um PVC e resolvido pela teoria de Airy, trata de apenas um único trem de ondas, definido por sua altura *H* e período *T*. Este tipo de representação é usualmente conhecido como "mar regular" ou "onda determinística".

Uma representação mais realista consiste em empregar um modelo espectral para um estado de "mar irregular", às vezes também referido como "ondas aleatórias". Neste modelo, o estado de mar irregular geral é representado pela superposição linear de várias ondas regulares, com diferentes valores de período, amplitude e fase. Para uma dada locação, medições e estudos estatísticos ajustam um modelo de espectro adequado para a representação da distribuição de densidade de energia apropriada das ondas do mar.

O ajuste do modelo espectral é feito em termos de parâmetros estatísticos, tais como fatores de forma espectral, altura significativa de onda e período de pico. Na estatística de curto prazo, estes parâmetros são supostos constantes, cada conjunto deles caracterizando um "estado de mar". A escolha do espectro de mar e de seus parâmetros característicos é função do fenômeno a ser estudado e dos levantamentos em medições realizadas na posição geográfica a que se queira referir.

O espectro mais comum de um único parâmetro é o modelo de Pierson-Moskovitz baseado na altura significativa de onda ou velocidade de vento. Dos espectros de dois parâmetros, os mais comumente usados são Bretschneider, Scott, ISSC e ITTC. O espectro de Jonswap é de cinco parâmetros, mas usualmente três destes parâmetros são mantidos constantes. A seguir será apresentado o modelo de espectro de Jonswap, que vem sendo estabelecido pela Petrobras como o padrão para a representação de estados de mar na Bacia de Campos.

Para o cálculo dos valores que caracterizam o comportamento das partículas do fluido em um dado ponto no espaço e um instante no tempo (tais como velocidades, acelerações e pressões), primeiramente efetua-se um procedimento de discretização do espectro em termos de um somatório de um número arbitrado de componentes de onda

regular. Neste procedimento, determinam-se os valores que caracterizam cada componente: períodos (ou frequencias), amplitudes e fases. Para cada componente aplicam-se as expressões de Airy, obtendo-se por exemplo as velocidades e acelerações em um dado ponto. Finalmente, os valores desejados para o estado de mar irregular podem ser determinados pelo somatório dos valores calculados para cada componente de onda regular.

Com isso é possível determinar as características da movimentação do fluido sob a ação de ondas (incluindo campos de **velocidades**, **acelerações** e **pressões**) sem a consideração da presença de um corpo flutuante ou submerso.

Espectro de Jonswap

O espectro de Jonswap resultou originalmente de um projeto conjunto executado no Mar do Norte, de onde deriva seu nome (*JOint North Sea WAve Project*). A expressão para o espectro de Jonswap pode ser escrita da seguinte forma [14]:

$$S(w) = \alpha \frac{g^2}{2\pi^4 w^5} \exp\left[-1.25\left(\frac{w}{w_p}\right)^{-4}\right] \gamma^{\exp\left[-\frac{(w-w_p)^2}{2\sigma^2 w_p^2}\right]} \tag{60}$$

Esta expressão fornece, a partir de um valor de freqüência w (em Hz), a densidade de energia correspondente S(w). Os parâmetros variáveis do espectro são a freqüência de pico w_p (em Hz), e os parâmetros de forma α e γ (este último conhecido como o "parâmetro de pico").

O parâmetro de forma σ é fixo, sendo determinado em função da relação entre a freqüência w e a freqüência de pico w_p:

$$\sigma = \begin{cases} \sigma_a = 0.07, & para \quad w \le w_p \\ \sigma_b = 0.09, & para \quad w > w_p \end{cases} \tag{61}$$

A figura 13 apresenta o espectro de Jonswap.

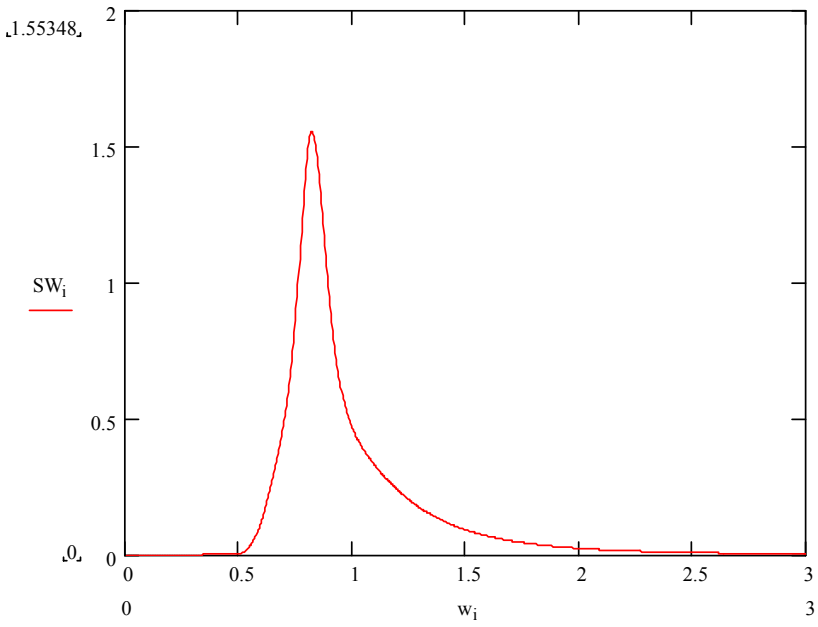


Figura 13 - Espectro de Jonswap

Recentemente a Petrobras propôs empregar uma expressão do espectro de Jonswap ajustada para as condições de onda da Bacia de Campos. Em particular, para projetos de fadiga estocástica, o espectro de onda de Jonswap pode ser usado na faixa de $4\text{s} \le T_p \le 17.7\text{s}$ e $0.47\text{m} \le H_s \le 6.51\text{m}$, estabelecendo as seguintes relações para determinar os parâmetros de forma α e γ a partir de H_s e T_p:

$$\gamma = \exp\left(1.0394 - 0.01966\frac{T_p}{\sqrt{H_s}}\right) \tag{62}$$

$$\alpha = 5.0609\frac{H_s^2}{T_p^4}\left[1 - 0.287 * \ln(\gamma)\right] \tag{63}$$

5.2.4. Cálculo das Forças

O presente item irá tratar dos procedimentos para o cálculo das **forças** no casco e nas linhas de ancoragem e risers exercidas pelo fluido. Este cálculo de forças é uma tarefa complexa, pois envolve diversas incertezas, que se somam às envolvidas na formulação do modelo de ondas, e na natureza randômica de um estado de mar real, como descrito no item anterior.

Atualmente existem formulações que, tendo sido verificadas e calibradas por ensaios experimentais e monitoração no mar, se mostram adequadas para representar com precisão as forças devidas à movimentação do fluido sobre sistemas offshore. Segundo Chakrabarti [14], estas formulações podem ser agrupadas em três classes principais, de acordo com sua adequação aos diferentes tipos de sistemas offshore:

- Formulação de Morison – casco, linhas de ancoragem e risers;
- Formulação de Froude-Krylov - casco;
- Modelo de Difração / Radiação - casco.

A seguir apresenta-se uma descrição resumida das principais características de cada uma destas formulações.

5.2.4.1. Formulação de Morison

A formulação de Morison é bastante difundida em aplicações práticas para o cálculo das forças de fluidos em corpos esbeltos, com dimensão transversal característica D pequena em comparação com o comprimento de onda λ. Um critério usualmente empregado para definir um "corpo esbelto" consiste em verificar se a seguinte relação é atendida:

$$\frac{D}{\lambda} < 5. \tag{64}$$

Nestes casos, a formulação de Morison assume que as forças podem ser computadas através de uma aproximação na qual os parâmetros importantes do fluxo na superfície do corpo, tais como pressão, velocidade e aceleração, podem ser aproximados pelo valor correspondente calculado no eixo da seção transversal do corpo esbelto.

A formulação de Morison considera que a força de onda é composta pela soma de duas parcelas:

1. Uma parcela de *arraste* associada a efeitos viscosos, proporcional às velocidades do fluido e do corpo;
2. Uma parcela de *inércia*, proporcional às acelerações do fluido e do corpo.

A equação de Morison pode ser expressa da seguinte forma:

$$\mathbf{F} = \frac{1}{2}\rho_w D\, C_d\, |\mathbf{u} - \dot{\mathbf{x}}|(\mathbf{u} - \dot{\mathbf{x}}) + \rho_w \frac{\pi D^2}{4} C_m \dot{\mathbf{u}} - \rho_w \frac{\pi D^2}{4} C_a \ddot{\mathbf{x}} \tag{65}$$

Nesta expressão, ρw é a massa específica do fluido; D é uma dimensão transversal característica do corpo (usualmente o diâmetro de um membro cilíndrico); e u, $\dot{x}$,$\dot{u}$ e $\ddot{x}$ são respectivamente as velocidades e acelerações do fluido e do corpo. O primeiro termo do lado direito desta equação (proporcional às velocidades) corresponde portanto à parcela de arraste; o segundo e terceiro termos (proporcionais às acelerações) correspondem à parcela de inércia. Geralmente considera-se que a formulação de Morison é mais aplicável quando a força de arraste é significativa, e os efeitos viscosos preponderam sobre os inerciais; este é usualmente o caso em corpos esbeltos [14].

A formulação de Morison é considerada semi-empírica, já que as parcelas de arraste e inércia são afetadas por coeficientes adimensionais C_d, C_m e C_a, que devem ser calibrados a partir da observação de resultados experimentais. Na análise de linhas de ancoragem e risers usualmente empregam-se valores de C_d variando entre 0,7 e 1,2, e valores de C_m em torno de 2,0 . O terceiro termo, afetado pelo coeficiente C_a (usualmente definido como $C_m - 1$) é proporcional às acelerações do corpo e está associado a efeitos de "massa adicionada".

A equação de Morison tem apresentado bons resultados em aplicações práticas tais como membros de plataformas fixas reticuladas (as *jaquetas*), e linhas de ancoragem e *risers* modelados por elementos finitos. Nestas aplicações, no entanto, deve-se ter em mente os seguintes aspectos:

- A Fórmula de Morison considera que a resposta do riser está alinhada com a direção do fluxo incidente. Portanto, omite forças de ***lift*** (sustentação) e forças de arrasto devido à vibração induzida por vórtices (***VIV***), que podem ser importantes em muitas situações.
- Não incorpora o efeito da esteira de interferência entre *risers* muito próximos (o que pode influenciar a parcela de arrasto). Um *riser* na esteira de outro pode receber menos carga, o que pode levar à colisão (***clashing***) entre os risers. Este efeito poderia ser modelado empiricamente, variando os valores do coeficiente C_d.

Como será comentado mais adiante, esta equação também pode ser empregada em plataformas flutuantes compostas por membros reticulados, tais como as plataformas semi-submersíveis, TLPs ou SPAR-buoys. Nestes casos, membros muito próximos podem "confinar" uma porção da massa de fluido, que pode agir como parte

da estrutura, levando ao aumento da força de massa adicionada. Assim, a utilização pura e simples da equação de Morison equivaleria a assumir que os membros, além de relativamente esbeltos, são razoavelmente espaçados entre si, de modo que o espaçamento médio entre dois membros é grande quando comparado com as dimensões transversais da seção. A força que o fluido exerce em cada membro não seria então afetada pela presença de outros membros, e a força total pode ser obtida somando-se as forças calculadas individualmente para cada membro. O efeito de "confinamento" do fluido poderia ser modelado empiricamente, aumentando o valor do coeficiente C_a (proporcional à aceleração do corpo), mas sem alterar o valor do coeficiente C_m que afeta apenas a força de inércia proporcional à aceleração do fluido.

5.2.4.2. Formulação de Froude-Krylov

Na formulação de Froude-Krylov, a força atuante no corpo é proveniente da pressão gerada pela passagem da onda incidente sobre a superfície do corpo, também considerando que a presença do corpo não afeta o fluxo. A partir de uma dada expressão para o campo de pressões no fluido gerado pela onda, podem ser obtidas as componentes de força resultante atuando em um corpo, em cada uma das direções de um sistema de eixos ortogonais. Para isto basta efetuar a integração da correspondente componente da pressão p, sobre a parte submersa do corpo, como indicado a seguir:

$$F_x = C_H \iint_s p \; n_x \; dS \tag{66}$$

$$F_y = C_V \iint_s p \; n_y \; dS \tag{67}$$

Estas expressões fornecem respectivamente as componentes horizontal e vertical da força resultante no corpo, n_x e n_y são as componentes horizontal e vertical do vetor normal à superfície do corpo, C_H e C_V são coeficientes de força horizontal e vertical, também determinados empiricamente, como será comentado a seguir (não devendo ser confundidos com os coeficientes de inércia e de arraste da fórmula de Morison).

A aplicação desta formulação torna-se mais conveniente quando associada a uma expressão do campo de pressões no fluido p derivada de uma teoria linear de onda, como por exemplo da teoria de Airy, que pode então ser empregada para fornecer a pressão dinâmica em um ponto na superfície de uma estrutura submersa, agindo normal

à superfície daquele ponto. Neste caso, a aplicação deste método é vantajosa já que, para algumas formas particulares de membros submersos (como cilindros ou esferas), podem ser obtidas expressões fechadas que fornecem as forças atuantes no corpo. Chakrabarti [14] demonstra que, em muitos casos, as expressões resultantes são semelhantes às obtidas pela parcela de inércia da fórmula de Morison (embora, como mencionado anteriormente, o coeficiente que deve ser determinado empíricamente não é o mesmo).

Desta forma, segundo Chakrabarti [14], a formulação de Froude-Krylov é mais aplicável quando a força de arraste é pequena, e os efeitos de inércia predominam sobre os viscosos, mas o corpo é ainda relativamente esbelto e portanto pode-se assumir que a sua presença não afeta significativamente o fluxo das partículas fluidas. Como, ainda segundo Chakrabarti, poucas aplicações práticas atendem a estas hipóteses, em casos onde os efeitos de difração são significativos, mas pequenos, é possível considerá-los na forma de um termo de correção nos coeficientes de força. Em casos mais gerais onde os efeitos de difração são mais importantes, isso não é possível. Além disso, a proximidade do corpo com o fundo ou a superfície livre pode gerar efeitos não facilmente quantificáveis nos coeficientes. Nestes casos, deveria então ser aplicada a formulação completa da teoria da difração.

Alternativamente, segundo Hooft [16], para corpos relativamente esbeltos tais como membros reticulados de plataformas flutuantes, a parcela de força de Froude-Krylov pode ser somada a termos de força de inércia e de arraste semelhantes às parcelas da fórmula de Morison.

5.2.4.3. Modelo de Difração / Radiação

Quando as dimensões do sistema offshore não são pequenas em relação ao comprimento de onda, as hipóteses consideradas nas seções anteriores não são válidas, e espera-se que a presença do corpo altere de forma significativa o campo de ondas na sua vizinhança, gerando efeitos de difração, interferência e radiação de ondas pelo corpo. Portanto, nestes casos de corpos de forma completamente geral, um método rigoroso para o cálculo das forças induzidas pela movimentação das partículas do fluido devida às ondas deve considerar um modelo de Difração/Radiação.

O modelo matemático tridimensional de Difração/Radiação é uma generalização do modelo bidimensional que representa a "*teoria de onda*". Enquanto o modelo da

"teoria de onda" tinha por objetivo apenas determinar velocidades e acelerações do fluido, sem considerar a presença do corpo, o modelo de Difração / Radiação considera a presença do corpo e tem por objetivo determinar as cargas que resultam da movimentação do fluido induzida pelas ondas.

Esse modelo está associado à Teoria Potencial, compondo um modelo matemático em termos de um PVC composto pela equação de Laplace tridimensional com as condições de contorno associadas, mas agora incluindo a consideração do corpo submetido à ação do fluido.

Em alguns casos particulares, como cilindros verticais fixos e semi-cilindros ou semi-esferas apoiadas no fundo, existem soluções analíticas fechadas disponíveis na literatura. Em casos mais gerais podem ser empregados métodos numéricos, baseados por exemplo na Teoria das Faixas e na formulação de Green [14].

O programa Wamit [17] é um código extensamente usado para computar cargas de fluido empregando um modelo de Difração/Radiação.

5.2.5. Modelo Híbrido

As formulações descritas no item anterior não necessitam ser consideradas como mutuamente exclusivas, e podem ser combinadas em modelos "híbridos" que combinam características positivas de mais de uma formulação, seguindo propostas como as apresentadas por [16] e [18]. Neste modelo híbrido, implementado no programa Prosim, combinam-se as seguintes forças:

- As forças de primeira ordem da fórmula de Morison, particularmente as forças viscosas de arraste;
- As forças de Froude-Krylov;
- As forças de segunda ordem oriundas da Teoria Potencial, incluindo efeitos de Difração e Radiação de ondas.

A utilização pura e simples da fórmula de Morison para o cálculo das cargas de fluido no casco de uma plataforma composta por membros reticulados tais como TLPs ou semisubmersíveis implicaria em assumir algumas simplificações, mencionadas a seguir.

Uma primeira simplificação está diretamente embutida na premissa em que se estabelece a formulação de Morison, pela qual um membro individual é esbelto o suficiente para que os efeitos de difração sejam insignificantes, e as forças possam ser computadas a partir de velocidades e acelerações calculadas pela teoria linear de Airy no eixo da seção transversal do membro. Com isso a perturbação da onda causada pela presença e movimento do corpo é ignorada.

Além disso, a formulação de Morison assume também que o espaçamento médio entre dois membros da plataforma é grande em comparação com as dimensões da seção transversal. Deste modo, a força em um membro individual não é afetada pela presença de outros membros e, em conseqüência, a força total no casco pode ser calculada como a soma das forças calculadas para cada um dos membros individuais. No entanto, um tratamento mais rigoroso deveria considerar que existe interação entre os membros, o que leva a efeitos de cancelamento ou sobreposição de ondas, o que depende da frequencia de cada componente de onda.

Já a Teoria Potencial é capaz de tratar adequadamente os efeitos devidos à perturbação (difração e radiação) da onda causada pela presença e movimento do corpo, bem como os efeitos devidos à interação entre as ondas que refletem nos membros da plataforma. Estes efeitos incluem [19]:

- O **amortecimento potencial** por irradiação de ondas pelo corpo, e
- As **forças de deriva média e lenta**, devidas à difração e reflexão das ondas em torno do corpo.

Por outro lado, enquanto as forças de primeira e segunda ordem são avaliadas pela Teoria Potencial no domínio da freqüência e válidas para pequenas amplitudes de onda e movimento, a formulação de Morison avalia as forças de primeira ordem no domínio do tempo, a cada passo do processo de integração, e são válidas para grandes amplitudes de onda e movimento. A formulação de Morison associada a uma análise no domínio do tempo é, portanto, capaz de tratar adequadamente os efeitos não-lineares, levando em conta a superfície livre instantânea e determinando as cargas exatamente no trecho imerso de cada membro em cada instante de tempo.

Além disso, modelos de Difração/Radiação baseados na teoria potencial não incorporam efeitos devidos à viscosidade do fluido; por exemplo, no programa Wamit o

amortecimento viscoso deve ser introduzido externamente pelo usuário, através de uma matriz de amortecimento calibrada. Argumenta-se em favor da teoria potencial que este fato não constitui problema em sua aplicação a corpos de grandes dimensões como navios, onde os efeitos de inércia são preponderantes e os efeitos viscosos seriam importantes apenas como amortecimento para excitações de ressonância, com períodos próximos a períodos naturais do corpo (por exemplo o *roll* em navios). A questão que se coloca é se esse raciocínio permanece válido para membros cilíndricos de plataformas, onde o diâmetro pode não ser tão grande quando comparado como comprimento da onda. Nesses casos, os efeitos viscosos, além de serem importantes como amortecimento na ressonância, podem ser importantes também como força de arraste para outras faixas de freqüência.

Quanto às forças de deriva, é interessante mencionar que existe um tipo de força de deriva que é fornecido pela equação de Morison. Esta parcela é devida à diferença na força de arrasto da onda em membros cortados pela superfície da água, que resulta da diferença do comprimento molhado do membro, da crista para o cavado, ao longo da passagem da onda pelo membro [18].

Em resumo, no modelo híbrido empregado no programa Prosim, as forças atuando na plataforma devidas à movimentação do fluido são compostas por várias parcelas, definidas na seguinte expressão:

$$F_{wc} = \mathbf{f}_{FK} + \mathbf{f}_{Mmn} + \mathbf{f}_{Mdn} + \mathbf{f}_{Mdn} + \mathbf{f}_{Ma} + \mathbf{f}_{D} + \mathbf{f}_{PD} \quad (68)$$

- O primeiro termo, $\mathbf{f}_{FK}$, é a força de Froude-Krylov, função da pressão do fluido p;
- O segundo e terceiro termos, $\mathbf{f}_{Mmn}$ e $\mathbf{f}_{Mdn}$, correspondem aos termos de inércia e arraste da fórmula de Morison, sendo funções de $\mathbf{a}_{rn}$ e $\mathbf{v}_{rn}$ (respectivamente as componentes normais das acelerações e velocidades relativas fluido-estrutura) e dos coeficientes de massa adicionada e de arrasto quadrático $C_{\boldsymbol{a}}$ e $C_{\boldsymbol{d}}$;
- O quarto termo, $\mathbf{f}_{Ma}$, corresponde à componente axial das forças de inércia e arraste, calculadas para membros com extremidades expostas à ação do fluido;
- O quinto termo, $\mathbf{f}_{D}$, corresponde às forças de Deriva Média e Lenta;
- O sexto termo $\mathbf{f}_{PD}$ corresponde às Forças de Amortecimento Potencial.

Detalhes de cada uma destas parcelas são apresentadas em Senra [13]. Observa-se em especial que os dois últimos termos são forças que resultam da aplicação do modelo de difração-radiação da Teoria Potencial. São incluídos na formulação híbrida a partir de resultados previamente calculados no domínio da freqüência por programas como o WAMIT [17].

Observa-se que estas parcelas dizem respeito apenas às cargas geradas pela movimentação do fluido devido às ondas e correnteza; outras parcelas de carga atuando na plataforma, como, por exemplo, cargas de vento, são descritas no item a seguir.

5.3. CORRENTEZA

Prosseguindo na descrição da formulação das equações de movimento, esse item apresenta a formulação para o cálculo das cargas **correnteza** (atuando tanto no casco quanto nas linhas de ancoragem e risers).

A correnteza é definida através de um perfil poligonal, em que são fornecidos valores de velocidade e ângulos de incidência. Este tipo de carregamento geralmente é aplicado incrementalmente à estrutura e fornecido através de uma função tempo, que pode ser associada ao carregamento de onda e correnteza.

A correnteza pode ser considerada como carregamento estático, embora existam alguns efeitos dinâmicos associados à correnteza. Pode-se mencionar por exemplo as vibrações induzidas por vórtices (VIV); e a flutuação no valor da velocidade da correnteza medida no tempo, que é usualmente ignorada.

No caso de corpos submersos para os quais a fórmula de Morison pode ser aplicada, tais como membros reticulados de plataformas ou linhas de ancoragem e risers, as cargas de correnteza podem ser consideradas diretamente no cálculo da parcela de arraste que leva em conta as velocidades relativas fluido-estrutura, simplesmente efetuando uma soma vetorial das velocidades de correnteza com as velocidades do fluido devidas à onda e as velocidades da estrutura.

Em projetos recentes de plataformas em águas profundas, tem sido observado que a parcela de carga de correnteza atuando sobre as linhas pode ser da mesma ordem de grandeza da parcela que atua sobre o casco da plataforma.

5.4. VENTO

Esse item apresenta a formulação para o cálculo das cargas de **vento** (atuando nas áreas expostas do casco e do convés da plataforma).

As cargas de vento atuam sobre a área exposta do casco e do convés das plataformas flutuantes. As condições de vento usadas em projeto devem ser apropriadamente determinadas a partir de dados coletados, consistentes com os outros parâmetros ambientais que ocorrem simultaneamente.

Existem duas maneiras de se considerar os efeitos de vento no projeto, que dependem de parâmetros do sistema e objetivos da análise:

- Força de vento constante no tempo, calculada com base na velocidade média de 1 minuto;

- Força de vento variável, calculada em função de um componente permanente, baseado na velocidade média de 1 hora, mais uma componente variando com o tempo, calculada a partir de um espectro de vento adequado.

Para o cálculo da parcela estática da carga de vento é assumido que a área exposta à ação do vento na embarcação possa ser caracterizada por um único número, o qual representa o produto da área exposta ao vento pelo fator de forma. O programa PROSIM também fornece a opção de se interpolar ao longo do tempo a área exposta à ação do vento devido à variação do aproamento (yaw) da plataforma.

Considerando-se que o centro de pressão de vento seja conhecido, a força de vento é considerada atuando neste ponto, em cada intervalo de integração, pela seguinte equação:

$$F_{vento} = \frac{\rho}{2} A_{vento} V_{vento}^2 \tag{69}$$

onde:

ρ - densidade do ar;

A_{vento} – produto da área exposta ao vento pelo coeficiente de forma;

V_{vento} – velocidade média do vento.

Resultados de teste de túnel de vento podem ser usados para estabelecer coeficientes de força (força/velocidade^2) em determinadas direções de incidência do vento. Assim, basta multiplicar o valor da velocidade de vento ao quadrado pelo coeficiente de força obtido do ensaio, para que seja determinada a força de vento sobre a embarcação. Nos ensaios, os coeficientes de força de vento são normalmente determinados para uma altura de 10 metros acima da lâmina d'água. Assim, para se obter as forças, as velocidades de vento medidas precisam ser transportadas para esta mesma altura de 10 metros, de acordo com a fórmula abaixo [20]:

$$V_Z = V_{10m} \times \left(\frac{Z}{10} \right)^{0,13} \tag{70}$$

De modo similar às ondas, os ventos também geram forças variáveis no tempo. Embora métodos para determinar a parcela de força de vento variável no tempo (também referida como força de vento de baixa freqüência [21]) tenham sido extensivamente estudados, há ainda um substancial grau de incerteza nesta estimativa, particularmente na definição de um espectro de energia a partir de dados medidos de vento. Na falta de dados mais precisos, a parcela variável no tempo pode ser obtida a partir da simulação do espectro proposto pela API RP 2A [22], que é apresentado a seguir:

$$S(f) = \frac{\sigma(z)^2}{f_p \left[1 + 1.5 \frac{f}{f_p} \right]^{5/3}} \tag{71}$$

$$\sigma(z) = \mathrm{V}(1\mathrm{hr},z) \times \begin{cases} 0.15 \left(z / z_s \right)^{-0.125} & para \quad z \le z_s \\ 0.15 \left(z / z_s \right)^{-0.275} & para \quad z > z_s \end{cases} \tag{72}$$

$$V(1hr, z) = V(1hr, z_R) \left(\frac{z}{z_R} \right)^{0.125} \tag{73}$$

$$0.01 \le \frac{f_p \cdot z}{V(1hr, z)} \le 0.10 \tag{74}$$

onde:

S(f) – densidade de energia espectral, na elevação z;

f – freqüência em Hz;

f_p – freqüência de pico característica do espectro;

$\sigma(z)$ – desvio-padrão da velocidade de vento;

$V(1hr,z)$ –Velocidade média de vento em 1 hora, medida na elevação z;

z_S – 20m (espessura da "camada superficial");

z_r - 10 m (altura da referência).

A figura 14 mostra o espectro de vento para uma velocidade média horária de 35 m/s e $\frac{f_p \cdot z}{V(1hr,z)} = 0.05$.

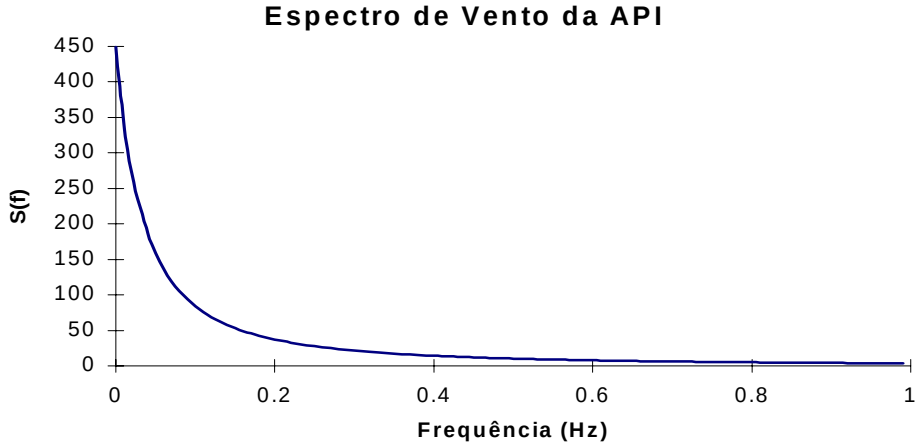


Figura 14 – Espectro de Vento

A simulação do vento consiste em uma série de componentes discretas, senoidais e unidirecionais, as quais são superpostas para se obter a velocidade instantânea do vento. Estas componentes são geradas em intervalos de igual energia do espectro, com fases distribuídas randomicamente no intervalo [0, 2π].

6. MÉTODOS EXPLÍCITOS DE INTEGRAÇÃO NO TEMPO

6.1. INTRODUÇÃO

Conforme mencionado no item 4.4.2, neste capítulo serão apresentados os algoritmos explícitos de integração no tempo. Inicialmente será descrito o Método das Diferenças Centrais, em seguida será apresentada a consideração de amortecimento numérico em algoritmos explícitos e uma comparação entre os algoritmos implementados.

Reapresentando os operadores de Newmark (37) e (38) nas equações de movimento (36), temos:

$$[\mathbf{M} + \gamma\,\Delta t\,\mathbf{C} + \beta\,\Delta t^2\,\mathbf{K}]\,\mathbf{a}_{n+1} = \\ = \mathbf{F}_{n+1} - \mathbf{C}\left[\mathbf{v}_n + (1 - \gamma)\,\Delta t\,\mathbf{a}_n\right] - \\ - \mathbf{K}\left[\mathbf{d}_n + \Delta t\,\mathbf{v}_n + \left(\frac{1}{2} - \beta\right)\Delta t^2\,\mathbf{a}_n\right] \qquad (75)$$

Adotando-se $\beta = 0$ e $\gamma = 1/2$, obtemos o método explícito mais comumente utilizado, que é o Método das Diferenças Centrais. Neste caso o sistema efetivo não é acoplado por coeficientes da matriz de rigidez e, caso a matriz de massa e a matriz de amortecimento sejam diagonais, tem-se a matriz efetiva diagonal, o que quer dizer que não há necessidade de se empregar uma técnica para resolução do sistema de equações. As incógnitas são obtidas diretamente pela divisão dos termos do vetor de cargas efetivo pelos termos da diagonal da matriz efetiva.

Em problemas não-lineares não há necessidade de se efetuar um tratamento específico, já que as não-linearidades são levadas em conta automaticamente durante a montagem do vetor de resíduos efetivos que estão do lado direito do sistema de equações e já são conhecidos no tempo n.

A seguir serão apresentados o Método das Diferenças Centrais [5] e mais dois métodos com dissipação numérica: Método de Chung-Lee [23] e Método de Hulbert-Chung [24].

6.2. MÉTODO DAS DIFERENÇAS CENTRAIS

Fazendo $\beta = 0$ e assumindo-se que a matriz de amortecimento pode ser negligenciada temos:

$$\mathbf{a}_{n+1} = \mathbf{M}^{-1} \left(\mathbf{F}_{n+1} - \mathbf{K} \left[\mathbf{d}_n + \Delta t\, \mathbf{v}_n + \left(\frac{1}{2}\right) \Delta t^2\, \mathbf{a}_n \right] \right) \tag{76}$$

Considerando também $\gamma = ½$ nas expressões (37) e (38), temos os seguintes operadores para deslocamentos e velocidades:

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\, \mathbf{v}_n + \frac{\Delta t^2}{2} \mathbf{a}_n \tag{77}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \left[\left(\frac{\mathbf{a}_n}{2} + \frac{\mathbf{a}_{n+1}}{2} \right) \right] \tag{78}$$

O MDC (Método das Diferenças Centrais) é o método explícito de integração mais comumente utilizado. Este método é indicado quando a matriz de massa é diagonal e pode-se desprezar a matriz de amortecimento. Uma desvantagem do MDC é a necessidade de utilização de um intervalo de tempo muito pequeno para integração. Isto torna-se necessário devido à característica de estabilidade condicional do algoritmo, que estabelece que o intervalo de tempo deve ser menor que um determinado valor crítico:

$$\Delta t \leq \Delta t_{cr} = \frac{T_n}{\pi} \tag{79}$$

onde T_n é o menor período natural da malha de elementos finitos com n graus de liberdade.

Em problemas de propagação de onda são necessários intervalos de tempo muito reduzidos para correta integração dos modos de vibração de freqüências mais elevadas. Estes intervalos de tempo necessários são geralmente menores que Δt_{cr}, daí a vantagem de utilização dos métodos explícitos visto que, para um mesmo intervalo de tempo, estes são mais eficientes que os métodos implícitos devido à não necessidade de resolução de sistemas de equações para obtenção da resposta.

Já para problemas inerciais, o menor intervalo de tempo necessário para integração correta do problema é geralmente muito maior que Δt_{cr}, daí a vantagem de

utilização dos métodos implícitos. Isto é, se utilizarmos o menor intervalo de tempo suficiente para um determinado problema inercial em um algoritmo explícito surgirão problemas de instabilidade na integração. Ou, por outro lado, se utilizarmos Δt_{cr} para este problema, estaremos utilizando um intervalo de tempo muito menor que o necessário, o que tornaria o método explícito menos eficiente que o implícito.

O MDC especificamente apresenta uma outra desvantagem, este método não apresenta dissipação numérica. Em alguns casos a discretização espacial do MEF introduz erros no espectro de freqüências em comparação com o problema real. Neste caso as equações de movimento semi-discretas aproximam melhor as freqüências mais baixas. As freqüências mais altas podem ser espúrias. Com isto torna-se desejável uma forma de eliminá-las da resposta dinâmica através da introdução de uma dissipação numérica controlada.

Os próximos 2 métodos explícitos de integração apresentados introduzem este amortecimento numérico.

6.3. ALGORITMO DE CHUNG E LEE

Este método, apresentado por Chung e Lee em 1995 [23], é uma variação do MDC onde é introduzido amortecimento numérico controlado de forma a eliminar as freqüências espúrias do problema.

Este amortecimento é introduzido no algoritmo através de parâmetros específicos apresentado no procedimento descrito a seguir [25].

A – Cálculos Iniciais:

1 – Montagem da matriz de massa **M** e vetor de esforços internos não-lineares $\mathbf{F}_{Int}(\mathbf{d}_n, \mathbf{v}_n)$;

2 – Inicialização de $\mathbf{d}_0$, $\mathbf{v}_0$, $\mathbf{F}_{Int}(\mathbf{d}_n, \mathbf{v}_n)$, $\mathbf{f}_0$ e $\mathbf{a}_0$, onde $\mathbf{a}_0 = \mathbf{M}^{-1}(\mathbf{f}_0 - \mathbf{F}_{Int}(\mathbf{d}_0, \mathbf{v}_0))$;

3 – Determinação do parâmetro β e do incremento de tempo (Δt) apropriados, onde $\Delta t < \Delta t_{critico}$, e cálculo das contantes de integração listadas abaixo:

$$\beta_1 = \Delta t^2 (1/2 - \beta) \qquad \beta_2 = \Delta t^2 \beta \tag{80}$$

$$\gamma_1 = -1/2\ \Delta t \qquad \gamma_2 = 3/2\ \Delta t$$

B – Para cada passo de tempo (n = 0, 1, 2,..., N-1):

1 – Cálculo da aceleração no tempo: $t_{n+1} = t_n + \Delta t$:

$$\mathbf{a}_{n+1} = \mathbf{M}^{-1} (\mathbf{f}_n - \mathbf{F}_{Int}(\mathbf{d}_n, \mathbf{v}_n)) \tag{81}$$

2 – Cálculo do deslocamento no tempo: $t_{n+1} = t_n + \Delta t$:

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\ \mathbf{v}_n + \beta_1\ \mathbf{a}_n + \beta_2\ \mathbf{a}_{n+1} \tag{82}$$

3 – Cálculo da velocidade no tempo: $t_{n+1} = t_n + \Delta t$:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \gamma_1\ \mathbf{a}_n + \gamma_2\ \mathbf{a}_{n+1} \tag{83}$$

4 – n ← n + 1, vai para o passo B-1.

O parâmetro β varia na seguinte faixa: $1 \leq \beta \leq 28 / 27$

Para $\beta = 1$ o algoritmo recai no MDC e para $\beta = 28 / 27$ o algoritmo apresenta a maior dissipação numérica possível dos modos de alta freqüência.

6.4. MÉTODO EXPLÍCITO GENERALIZADO - ALGORITMO DE HULBERT E CHUNG

Hulbert e Chung [24] também desenvolveram um algoritmo com a introdução da dissipação numérica controlada. O método aqui apresentado utiliza uma parcela preditora e uma parcela corretora para integração. Este método é denominado Método Explícito Generalizado α cujo esquema é apresentado a seguir [25].

1 – Predição para os valores de deslocamento e velocidade no tempo t_{n+1}:

$$\mathbf{dp}_{n+1} = \mathbf{d}_n + \Delta t_{n+1}\, \mathbf{v}_n + \Delta t^2_{n+1}\, (1/2 - \beta)\, \mathbf{a}_n \tag{84}$$

$$\mathbf{vp}_{n+1} = \mathbf{v}_n + \Delta t_{n+1}\, (1 - \lambda)\mathbf{a}_n$$

2 – Estima-se os valores de deslocamentos e velocidades em $t_{n+1-\alpha f}$:

$$\mathbf{d}_{n+1-\alpha f} = (1 - \alpha_f)\, \mathbf{dp}_{n+1} + \alpha_f\, \mathbf{d}_n \tag{85}$$

$$\mathbf{v}_{n+1-\alpha f} = (1 - \alpha_f)\, \mathbf{vp}_{n+1} + \alpha_f\, \mathbf{v}_n$$

3 – Aplica-se a equação de balanço para determinar $a_{n+1-\alpha m}$:

$$\mathbf{M}\, \mathbf{a}_{n+1-\alpha m} + \mathbf{C}\, \mathbf{v}_{n+1-\alpha f} + \mathbf{K}\, \mathbf{d}_{n+1-\alpha f} = \mathbf{F}\, (t_{n+1-\alpha f}) \tag{86}$$

4 – A partir da aceleração $a_{n+1-\alpha m}$ determina-se a aceleração a_{n+1}:

$$\mathbf{a}_{n+1} = \frac{\mathbf{a}_{n+1-\alpha m} - \alpha_m\, \mathbf{a}_n}{(1 - \alpha_m)} \tag{87}$$

5 – Obtida a aceleração a_{n+1}, faz-se a correção para os valores previstos para as velocidades e deslocamentos em t_{n+1}:

$$\mathbf{d}_{n+1} = \mathbf{dp}_{n+1} + \beta^2_{\Delta t\, n+1}\, \mathbf{a}_{n+1} \tag{88}$$

$$\mathbf{v}_{n+1} = \mathbf{vp}_{n+1} + \lambda_{\Delta t\, n+1}\, \mathbf{a}_{n+1}$$

Os parâmetros α_m, α_f, β e λ são dados por:

$$\alpha_m = \frac{2\,\rho_\infty - 1}{(\rho_\infty + 1)}$$

$$\alpha_f = \frac{\rho_\infty}{(\rho_\infty + 1)} \tag{89}$$

$$\lambda = ½ - \alpha_m + \alpha_f$$

$$\beta = ¼\,(\,½ + \lambda\,)^2$$

colocando α_m em função de ρ_p e ρ_s tem-se:

$$\alpha_m = \frac{2\,\rho_p\,\rho_s + \rho_p - 1}{(\rho_p + 1)\,(\rho_s + 1)} \tag{90}$$

Os parâmetros $\alpha_f = 1$, $\rho_p = \rho_s = 0.4$, foram utilizados de acordo com [24].

6.5. COMPARAÇÃO ENTRE OS ALGORITMOS

Os algoritmos descritos nos sub-itens anteriores foram implementados durante o desenvolvimento desta tese no sistema Prosim e, com o objetivo de verificação da implementação, é apresentado a seguir um estudo comparativo. O problema considerado nesta seção, ilustrado na Figura 15, consiste em uma viga monoengastada, submetida a uma carga aplicada axialmente [26]. Trata-se de um teste clássico, que excita frequências altas, e é representativo de problemas de propagação de ondas.

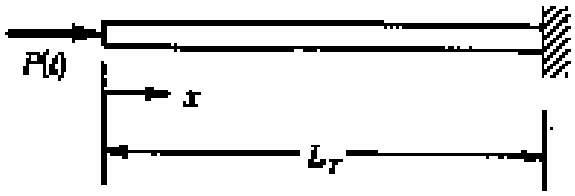


Figura 15 – Viga Monoengastada sob Carga Axial

Dados Geométricos e do Material

A Tabela 1 reproduz os dados geométricos e do material.

Tabela 1 – Propriedades

Área da Seção Transversal	A (pol^2)	1.0
Densidade do Material	ρ ($lb.s^2/pol^4$)	7.4 e–04
Módulo de Elasticidade	E (psi)	30 e+06
Comprimento	Lt (pol)	20

Basicamente, o modelo numérico consiste em uma malha uniforme de 40 elementos de pórtico; cada elemento tem portanto um comprimento igual a L = 0.5pol.

Carregamento

O carregamento consiste em uma carga axial de compressão com valor de 100 lb aplicada na extremidade livre da viga. O valor total da carga é aplicado de forma instantânea, atuando de forma constante até o final da análise, como indicado na Figura 16.

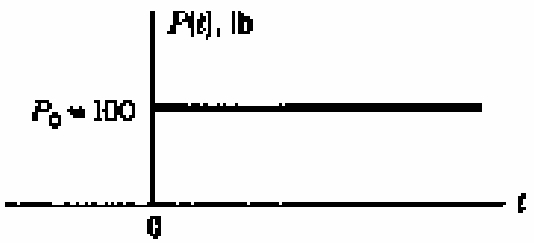


Figura 16 – Carregamento aplicado

Intervalo de Tempo

Algoritmos explícitos como o Método das Diferenças Centrais requerem o uso de um intervalo de tempo Δt *menor* do que um determinado valor crítico Δt_{cr}. Por sua vez, o valor do intervalo de tempo crítico Δt_{cr} é função de T_n, o menor período natural do modelo estrutural, discretizado pela malha de elementos finitos. No caso do Método das Diferenças Centrais, o intervalo de tempo crítico é dado por

$$\Delta t_{cr} = \frac{T_n}{\pi} = \left(\frac{2\pi}{\omega_n}\right)\frac{1}{\pi} = \frac{2}{\omega_n} \tag{91}$$

onde ω_n é a maior frequência natural da malha em que a estrutura foi discretizada, em rad/s.

Para problemas com amortecimento estrutural, o intervalo de tempo crítico também é função da percentagem de amortecimento crítico ξ para a maior frequência:

$$\Delta t_{cr} = \frac{2}{\omega_n}\sqrt{1+\xi^2} - \xi \tag{92}$$

No exemplo aqui apresentado não se utilizou amortecimento estrutural.

Em geral não se dispõe facilmente do valor exato de ω_n, que deveria ser determinado montando e resolvendo o problema de autovalor associado ao modelo de elementos finitos. Além disso, em problemas não-lineares o valor de ω_n pode variar ao longo do tempo. Por isso, é usual empregar uma estimativa para a maior frequência de um *elemento* da malha, em função da *velocidade de propagação da onda* (c) [26]:

$$(\omega_n)^e = \frac{2c}{L} = \frac{2}{L}\sqrt{\frac{E}{\rho}} \tag{93}$$

Para o problema da viga engastada, modelada por elementos com L=0.5pol, as expressões (93) e (91) fornecem as seguintes estimativas para a frequência e o intervalo de tempo crítico, respectivamente:

$(\omega_n)^e$ = 8.0539 e+05 rad/seg

$\Delta t_{cr}\ (\xi=0)$ = 2.4833 e–06 seg

A seguir, na figura 17, é apresentado a resposta de força no ponto central da viga utilizando-se os algoritmos apresentados e comparados com a resposta analítica.

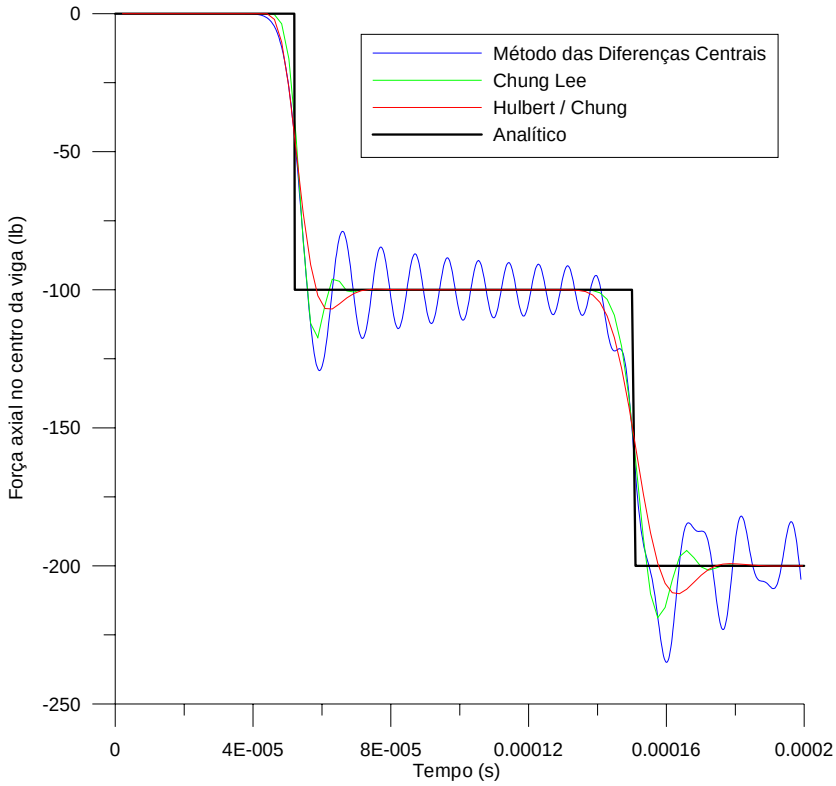


Figura 17 – Comparação entre os métodos

Conforme pode-se observar, o efeito do amortecimento numérico aproxima a resposta obtida nas análises da resposta analítica.

7. MÉTODOS IMPLÍCITOS DE INTEGRAÇÃO NO TEMPO

7.1. INTRODUÇÃO

Conforme mencionado no item 4.4.2, neste capítulo serão apresentados os algoritmos implícitos de integração no tempo. Inicialmente será descrita a implementação por deslocamentos do algoritmo implícito para problemas lineares, em seguida será apresentada a consideração de problemas não-lineares. Também será apresentado a consideração de amortecimento numérico em algoritmos implícitos e detalhes da implementação no programa Prosim.

7.2. IMPLEMENTAÇÃO POR DESLOCAMENTOS

Uma implementação usual para algoritmos implícitos consiste em empregar os operadores (39) e (40) para eliminar as acelerações e velocidades das equações de movimento (36), resultando em uma expressão onde as incógnitas são os deslocamentos. Antes de apresentar esta implementação, vamos observar as equações (39) e (40) e definir as seguintes constantes:

$$a_0 = \frac{1}{\beta \Delta t^2}\ ; \quad a_1 = \frac{\gamma}{\beta \Delta t}\ ; \quad a_2 = \frac{1}{\beta \Delta t}\ ; \quad a_3 = \left(\frac{1}{2\beta} - 1\right); \quad a_4 = \left(\frac{\gamma}{\beta} - 1\right); \quad a_5 = \left(\frac{\gamma}{2\beta} - 1\right) \Delta t \tag{94}$$

Empregando estas constantes, os operadores (39) e (40) podem ser escritos da seguinte forma:

$$\mathbf{a}_{n+1} = a_0 (\mathbf{d}_{n+1} - \mathbf{d}_n) - a_2 \mathbf{v}_n - a_3 \mathbf{a}_n \tag{95}$$

$$\mathbf{v}_{n+1} = a_1 (\mathbf{d}_{n+1} - \mathbf{d}_n) - a_4 \mathbf{v}_n - a_5 \mathbf{a}_n \tag{96}$$

Finalmente, vamos aplicar os operadores (95) e (96) para eliminar as acelerações e velocidades da expressão (36), e em seguida vamos passar os termos já conhecidos no instante t_n para o lado direito. Com isso, obtemos a seguinte expressão:

$$\left[a_0 \mathbf{M} + a_1 \mathbf{C} + \mathbf{K} \right] \mathbf{d}_{n+1} = \mathbf{F}_{n+1} + \mathbf{M} \left[a_0 \mathbf{d}_n + a_2 \mathbf{v}_n + a_3 \mathbf{a}_n \right] +$$

$$+ \mathbf{C} \left[a_1 \mathbf{d}_n + a_4 \mathbf{v}_n + a_5 \mathbf{a}_n \right] \tag{97}$$

Esta expressão define um ***Sistema Efetivo*** de equações algébricas lineares, que pode ser escrito da forma

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \tag{98}$$

onde a matriz de coeficientes **A** é a *matriz efetiva*, definida como uma combinação das matrizes de massa, rigidez e amortecimento, afetadas por coeficientes escalares; e o vetor de termos independentes **b** é o vetor de *cargas efetivas*, calculados em termos das cargas externas, e de forças elásticas, de inércia e de amortecimento do passo anterior.

Verificamos portanto que o processo de integração no tempo recai na ***solução de um sistema de equações algébricas lineares*** para cada instante de tempo. Considerando o uso de uma técnica direta para a solução dos sistemas de equações, o processo de integração no tempo em problemas lineares é o detalhado na Tabela 2:

Tabela 2 – Implementação Computacional do Procedimento de Integração no Tempo para Problemas Lineares

⇒ *Ao início da análise*, montar a matriz efetiva **A** / triangularizar.

⇒ *Loop de instantes de tempo*: conhecidas as aproximações $\mathbf{a}_n$, $\mathbf{v}_n$ e $\mathbf{d}_n$:

- Calcular o vetor de cargas efetivo **b** ;
- Efetuar uma retrosubstituição para resolver o sist. efetivo obtendo-se $\mathbf{d}_{n+1}$;
- Calcular $\mathbf{a}_{n+1}$ e $\mathbf{v}_{n+1}$ através dos operadores de Newmark;
- Incrementar *n* e passar para o próximo instante de tempo.

Nos próximos subitens serão apresentados os procedimentos para solução de problemas não-lineares para algoritmos implícitos e o algoritmo com dissipação numérica αB-Newmark.

7.3. PROBLEMAS NÃO-LINEARES EM ALGORITMOS IMPLÍCITOS

Para problemas não-lineares, as equações de movimento semi-discretas podem ser expressas da forma:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{R}(\mathbf{u}) = \mathbf{F}(\mathbf{u},t) \tag{99}$$

As não-linearidades estão embutidas nas parcelas **R**(u) e **F**(u,t). A parcela de esforços elásticos **R**(u) inclui efeitos geométricos e/ou de materiais com comportamento elástico não-linear. A parcela de cargas externas **F**(u,t) considera não-linearidade devido à variação das cargas externas com a geometria, caracterizando carregamentos não-conservativos.

A solução do sistema de EDO não-linear (99) associado a algoritmos implícitos exige procedimentos específicos para o tratamento das não-linearidades, como apresentado a seguir. Inicialmente, escreve-se a forma discretizada correspondente:

$$\mathbf{M}\,\mathbf{a}_{n+1} + \mathbf{C}\,\mathbf{v}_{n+1} + \mathbf{R}(\mathbf{d}_{n+1}) = \mathbf{F}_{n+1}(\mathbf{d}_{n+1}) \tag{100}$$

O tratamento do problema não-linear baseia-se em assumir que, no entorno de uma configuração deformada **u**, o problema pode ser considerado localmente linear. Esta linearização consiste em tomar a seguinte aproximação para as parcelas não-lineares $\mathbf{R}(\mathbf{d}_{n+1})$ e $\mathbf{F}_{n+1}(\mathbf{d}_{n+1})$, através de uma série de Taylor com termos de ordem superior truncados:

$$R(d_{n+1}) = R(d_n) + \left.\frac{\partial R}{\partial d}\right|_{d_n} \Delta d \tag{101}$$

$$F_{n+1}(d_{n+1}) = F_{n+1}(d_n) + \left.\frac{\partial F_{n+1}}{\partial d}\right|_{d_n} \Delta d \tag{102}$$

onde

$$\Delta\mathbf{d} = \mathbf{d}_{n+1} - \mathbf{d}_n \tag{103}$$

A última parcela de (102), que define a variação das cargas externas com a geometria, não será considerada nos desenvolvimentos posteriores. Esta parcela

usualmente só é levada em conta quando se exige um tratamento muito rigoroso de carregamento não-conservativo, já que compõe uma matriz não-simétrica.

A matriz de rigidez tangente é definida como:

$$\mathbf{K}_T = \frac{\partial \mathbf{R}}{\partial \mathbf{d}} \tag{104}$$

Formulação Incremental

Substituindo (101), (102) e (104) em (100), as equações de equilíbrio dinâmico discretizadas no espaço e no tempo escrevem-se da seguinte forma *incremental*:

$$\mathbf{M}\,\mathbf{a}_{n+1} + \mathbf{C}\,\mathbf{v}_{n+1} + \mathbf{K}_T\,\Delta\mathbf{d} = \mathbf{F}_{n+1}(\mathbf{d}_n) - \mathbf{R}(\mathbf{d}_n) \tag{105}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta\mathbf{d} \tag{106}$$

onde $\mathbf{R}(\mathbf{d}_n)$ são os esforços elásticos resistentes calculados com os deslocamentos do intervalo anterior, e a matriz de amortecimento de Rayleigh agora é dada por:

$$\mathbf{C} = \alpha_m \mathbf{M} + \alpha_k \mathbf{K}_T \tag{107}$$

Formulação Incremental-Iterativa

Deve-se observar que estas equações não mais garantem o equilíbrio dinâmico ao fim do intervalo de tempo t_{n+1}, devido às linearizações assumidas em (101) e (102). Por isto, é necessário empregar uma técnica iterativa para resolver o problema não linear. Usualmente emprega-se o Método de Newton-Raphson e suas variações, que consistem em escrever as equações de movimento na seguinte forma *incremental-iterativa*:

$$\mathbf{M}\,\mathbf{a}_{n+1}^{(k)} + \mathbf{C}\,\mathbf{v}_{n+1}^{(k)} + \mathbf{K}_T\,\Delta\Delta\mathbf{d}^{(k)} = \mathbf{F}_{n+1} - \mathbf{R}(\mathbf{d}_{n+1}^{(k-1)}, \Delta\Delta\mathbf{d}^{(k-1)}) \tag{108}$$

$$\Delta\mathbf{d}^{(k)} = \Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)} \tag{109}$$

$$\mathbf{d}_{n+1}^{(k)} = \mathbf{d}_{n+1}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)} \tag{110}$$

Nestas expressões, os superscritos *k* e *k-1* indicam um contador de iterações, e $\Delta\Delta\mathbf{d}^{(k-1)}$ representa a variação dos deslocamentos incrementais obtida a cada iteração do ciclo de verificação do equilíbrio.

A formulação do Método de Newton-Raphson baseia-se portanto em adotar a linearização (101) e iterar com matrizes tangentes como a dada por (104). No Método de Newton-Raphson Padrão **NRP**, a matriz tangente é reavaliada em todas iterações. No entanto, em alguns casos os custos de montagem e decomposição associados não compensam os ganhos com a convergência do processo, e o método de Newton-Raphson modificado **NRM** é uma alternativa interessante. Nesta técnica, a matriz de rigidez tangente K_T é calculada ao início de cada intervalo de tempo e mantida constante ao longo do ciclo iterativo, podendo ainda ser mantida constante ao longo de um certo número de intervalos de tempo.

O vetor de cargas externas $F_{n+1}(d_n)$ é reavaliado ao início de cada intervalo de tempo, e é mantido constante ao longo do ciclo iterativo. Os esforços elásticos resistentes **R** estão expressos também em função das variações dos deslocamentos incrementais $\Delta\Delta\mathbf{d}^{(k-1)}$ porque estes são utilizados na formulação do elemento de pórtico não-linear tridimensional empregado.

Aplicação do Operador de Integração no Tempo

Observando os operadores de Newmark em termos de acelerações e velocidades (95) e (96), verifica-se que os deslocamentos incrementais $(\mathbf{d}_{n+1} - \mathbf{d}_n)$ correspondem a $\Delta\mathbf{d}^{(k)}$, o qual por sua vez pode ser substituído pelo lado direito da expressão (109). Com isso, obtemos a seguinte forma "incremental-iterativa" para os operadores de Newmark:

$$\mathbf{a}_{n+1}^{(k)} = -a_2\,\mathbf{v}_n - a_3\,\mathbf{a}_n + a_0\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)}) \tag{111}$$

$$\mathbf{v}_{n+1}^{(k)} = -a_4\,\mathbf{v}_n - a_5\,\mathbf{a}_n + a_1\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)}) \tag{112}$$

De forma semelhante ao apresentado para o caso linear, a resposta dinâmica não-linear pode então ser obtida através da aplicação destes operadores (111) e (112) sobre a expressão incremental-iterativa das equações de movimento (108). Com isto, obtém-se um sistema "efetivo" de equações algébricas lineares, que devem ser resolvidas a cada iteração do procedimento de Newton-Raphson.

Uma forma geral para este sistema "efetivo" é dada por

$$\mathbf{A}\,\Delta\Delta\mathbf{d}^{(k)} = \mathbf{b}^{(k-1)} \tag{113}$$

onde a matriz de coeficientes **A** é a *matriz efetiva*, definida como uma combinação das matrizes de massa, rigidez e amortecimento, afetadas por coeficientes escalares; e o vetor de termos independentes $\mathbf{b}^{(k-1)}$ é o vetor de *resíduos efetivos*, calculados em termos das cargas externas, e de forças elásticas, de inércia e de amortecimento da iteração anterior.

A forma particular para **A** e $\mathbf{b}^{(k-1)}$ é definida de acordo com a implementação adotada para o operador de integração no tempo, como será demonstrado adiante.

7.4. SOLUÇÃO DO PROBLEMA DINÂMICO: O ALGORITMO αB-NEWMARK

O algoritmo empregado no Prosim, conhecido como αB-Newmark, resultou da proposta de Bossak e Zienkiewicz [27,28] para uma modificação no algoritmo original de Newmark, com objetivos e metodologia semelhantes à que levou ao desenvolvimento do algoritmo HHT ou αH-Newmark [29,30].

O algoritmo αH-Newmark é um algoritmo implícito, com propriedades de dissipação numérica capaz de reduzir a participação dos modos de vibração com frequências mais altas, que poderiam introduzir ruídos espúrios na resposta dinâmica. Emprega os mesmos operadores que caracterizam a família de algoritmos de Newmark, equações (37) e (38). A particularidade do algoritmo αH-Newmark consiste na expressão das equações de movimento discretizadas no tempo, que passam a ser escritas da seguinte forma:

$$\mathbf{M}\,\mathbf{a}_{n+1} + (1+\alpha)\mathbf{C}\,\mathbf{v}_{n+1} - \alpha\mathbf{C}\,\mathbf{v}_n + (1+\alpha)\mathbf{K}\,\mathbf{d}_{n+1} - \alpha\,\mathbf{K}\,\mathbf{d}_n =$$

$$(1+\alpha)\mathbf{F}_{n+1} - \alpha\mathbf{F}_n \qquad (114)$$

Observa-se que estas expressões introduziram um parâmetro α. Trata-se de um parâmetro ajustável que permite controlar o grau de dissipação, e que deve ser fornecido pelo usuário no intervalo [-1/3,0]. Além disso, os parâmetros γ e β também passam a ser definidos em função de α, da seguinte forma:

$$\gamma = (1 - 2\alpha) / 2 \qquad (115)$$

$$\beta = (1 - \alpha)^2 / 4 \qquad (116)$$

Com isso o algoritmo αH-Newmark também incorpora, como caso particular, a regra trapezoidal (já que, fornecendo-se $\alpha = 0$, as equações discretizadas recaem na forma (36), e os valores para os parâmetros γ e β recaem em $\gamma = ½$ e $\beta = ¼$). O algoritmo αH-Newmark é incondicionalmente estável, com ordem de precisão 2, como demonstrado nos estudos das propriedades de convergência, estabilidade, consistência e precisão apresentados em [29, 30].

O algoritmo αB-Newmark também emprega os mesmos operadores que caracterizam a família de algoritmos de Newmark, equações (37) e (38). Emprega também um parâmetro ajustável α, com o mesmo objetivo de controlar o grau de dissipação numérica para reduzir ruídos espúrios de alta frequência, e que deve ser fornecido pelo usuário no intervalo [-1/3,0]. Demonstra-se [6,28] que os métodos αH-Newmark e αB-Newmark fornecem resultados muito semelhantes, principalmente para os valores mais usualmente fornecidos para α, não muito próximos do limite -1/3.

A particularidade do algoritmo αB-Newmark consiste na expressão das equações de movimento discretizadas no tempo, que passam a ser escritas da seguinte forma:

$$(1-\alpha)\mathbf{M}\,\mathbf{a}_{n+1} + \alpha\,\mathbf{M}\,\mathbf{a}_{n} + \mathbf{C}\,\mathbf{v}_{n+1} + \mathbf{K}\,\mathbf{d}_{n+1} = \mathbf{F}_{n+1} \qquad (117)$$

Comparando esta expressão com a que caracteriza o algoritmo αH-Newmark, observa-se que os multiplicadores α não afetam os termos de amortecimento e de forças elásticas (que dependem das matrizes **C** e **K**), mas sim o termo de forças de inércia, que depende da matriz de massa **M**. Este fato acarreta em diversas vantagens na implementação computacional, que se torna mais simples, particularmente em problemas não-lineares.

7.5. TRATAMENTO DE PROBLEMAS NÃO-LINEARES: IMPLEMENTAÇÃO OTIMIZADA αB-NEWMARK/NEWTON-RAPHSON

Vamos inicialmente escrever a seguinte forma incremental-iterativa das equações de movimento discretizadas no tempo que corresponde ao algoritmo αB-Newmark associado ao Método de Newton-Raphson para o tratamento de problemas não-lineares:

$$(1-\alpha)\mathbf{M}\,\mathbf{a}_{n+1}^{(k)} + \alpha\,\mathbf{M}\,\mathbf{a}_n + \mathbf{C}\,\mathbf{v}_{n+1}^{(k)} + \mathbf{K}_T\,\Delta\Delta\mathbf{d}^{(k)} = \mathbf{F}_{n+1} - \mathbf{R}(\mathbf{d}_{n+1}^{(k-1)}) \qquad (118)$$

Vamos agora reescrever estas equações levando em conta a expressão para a matriz de amortecimento de Rayleigh (107):

$$\mathbf{M}\,[(1-\alpha)\mathbf{a}_{n+1}^{(k)} + \alpha_m\,\mathbf{v}_{n+1}^{(k)}] + \mathbf{K}_T\,(\alpha_k\,\mathbf{v}_{n+1}^{(k)} + \Delta\Delta\mathbf{d}^{(k)}) = \mathbf{F}_{n+1} - \mathbf{R}(\mathbf{d}_{n+1}^{(k-1)}) - \alpha\,\mathbf{M}\,\mathbf{a}_n \qquad (119)$$

Em seguida, vamos trabalhar sobre a forma incremental-iterativa dos operadores de Newmark, equações (111) e (112). Pode ser observado que os dois primeiros termos do lado direito daquelas equações não são incógnitas, já que dependem somente de valores já obtidos no passo anterior. Estes termos, portanto, definem aproximações iniciais para as acelerações e velocidades (também conhecidas como parcelas "preditoras" $\mathbf{a}^*$ e $\mathbf{v}^*$):

$$\mathbf{a}^* = -a_2\,\mathbf{v}_n - a_3\,\mathbf{a}_n \qquad (120)$$

$$\mathbf{v}^* = -a_4\,\mathbf{v}_n - a_5\,\mathbf{a}_n \qquad (121)$$

Assim, as equações (111) e (112) podem ser reescritas em termos destas parcelas preditoras, da seguinte forma:

$$\mathbf{a}_{n+1}^{(k)} = \mathbf{a}^* + a_0\,\Delta\mathbf{d}^{(k)} \qquad (122)$$

$$\mathbf{v}_{n+1}^{(k)} = \mathbf{v}^* + a_1\,\Delta\mathbf{d}^{(k)} \qquad (123)$$

ou, considerando (109),

$$\mathbf{a}_{n+1}^{(k)} = \mathbf{a}^* + a_0\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)}) \qquad (124)$$

$$\mathbf{v}_{n+1}^{(k)} = \mathbf{v}^* + a_1\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)}) \qquad (125)$$

Podemos agora empregar estas expressões reescritas dos operadores de Newmark (124) e (125) na forma das equações de movimento incremental-iterativa (119). Com isso o lado esquerdo destas equações passa a se escrever como:

$$\mathbf{M}\left[(1-\alpha)\left(\mathbf{a}^{*} + a_0\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)})\right) + \alpha_m\left(\mathbf{v}^{*} + a_1\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)})\right)\right] +$$

$$+ \mathbf{K}_T\left[\alpha_k\left(\mathbf{v}^{*} + a_1\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)})\right) + \Delta\Delta\mathbf{d}^{(k)}\right] = \ldots\ldots \qquad (126)$$

Neste ponto os termos já conhecidos na iteração *(k)* poderiam ser transferidos para o lado direito das equações de movimento, para serem incorporados no vetor de resíduos efetivo $\mathbf{b}^{(k-1)}$ da eq (113).

No entanto, para evitar as multiplicações com as matrizes globais e obter a desejada redução de custos computacionais, podemos introduzir um artifício [31] que consiste em definir um vetor $\Delta\Delta\hat{\mathbf{d}}^{(k)}$ como sendo os termos entre colchetes que multiplicam $\mathbf{K}_T$ em (126), ou seja

$$\Delta\Delta\hat{\mathbf{d}}^{(k)} = \alpha_k\left(\mathbf{v}^{*} + a_1\,(\Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}^{(k)})\right) + \Delta\Delta\mathbf{d}^{(k)} \qquad (127)$$

e, reciprocamente,

$$\Delta\Delta\mathbf{d}^{(k)} = \frac{\Delta\Delta\hat{\mathbf{d}}^{(k)} - \alpha_k\left(\mathbf{v}^{*} + a_1\,\Delta\mathbf{d}^{(k-1)}\right)}{\alpha_k\,a_1 + 1} \qquad (128)$$

Pode ser observado que, em casos particulares onde o amortecimento proporcional à rigidez não é considerado (ou seja, $\alpha_k = 0$ na eq (107)), as equações (127) e (128) se reduzem a:

$$\Delta\Delta\hat{\mathbf{d}}^{(k)} = \Delta\Delta\mathbf{d}^{(k)} \qquad (129)$$

Reescrevendo a equação (126) empregando a definição de $\Delta\Delta\hat{\mathbf{d}}^{(k)}$ expressa pela eq (127), e em seguida transferindo os termos já conhecidos na iteração *(k)* para o lado direito das equações de movimento, obtemos

$$\mathbf{M}\,(1-\alpha)\,(a_0 + \alpha_m\,a_1)\,\Delta\Delta\mathbf{d}^{(k)} + \mathbf{K}_T\,\Delta\Delta\hat{\mathbf{d}}^{(k)} =$$

$$\mathbf{F}_{n+1} - \mathbf{R}(\mathbf{d}_{n+1}^{(k-1)}) - \alpha\mathbf{M}\,\mathbf{a}_n - \mathbf{M}\left[(1-\alpha)\left(\mathbf{a}^{*} + a_0\,\Delta\mathbf{d}^{(k-1)}\right) + \alpha_m\left(\mathbf{v}^{*} + a_1\,\Delta\mathbf{d}^{(k-1)}\right)\right] \qquad (130)$$

Substituindo (128) em (130), o lado esquerdo das equações de movimento é expresso como:

$$\mathbf{M}\,(1-\alpha)\,(a_0 + \alpha_m\, a_1)\left(\frac{\Delta\Delta\hat{\mathbf{d}}^{(k)} - \alpha_k\,(\mathbf{v}^* + a_1\,\Delta\mathbf{d}^{(k-1)})}{\alpha_k\, a_1 + 1}\right) + \mathbf{K}_T\,\Delta\Delta\hat{\mathbf{d}}^{(k)} = \mathbf{..} \quad (131)$$

Definindo

$$\hat{\alpha}_0 = (1-\alpha)\,(a_0 + \alpha_m\, a_1)\left(\frac{1}{\alpha_k\, a_1 + 1}\right) \quad (132)$$

e novamente transferindo termos já conhecidos na iteração *(k)* para o lado direito das equações de movimento, obtemos

$$\hat{\alpha}_0\,\mathbf{M}\,\Delta\Delta\hat{\mathbf{d}}^{(k)} + \mathbf{K}_T\,\Delta\Delta\hat{\mathbf{d}}^{(k)} = \mathbf{F}_{n+1} - \mathbf{R}(\mathbf{d}_{n+1}^{(k-1)}) - \alpha\,\mathbf{M}\,\mathbf{a}_n - \\ - \mathbf{M}\left[(1-\alpha)\left(\mathbf{a}^* + a_0\,\Delta\mathbf{d}^{(k-1)}\right) + (\alpha_m - \hat{\alpha}_0\alpha_k)\left(\mathbf{v}^* + a_1\,\Delta\mathbf{d}^{(k-1)}\right)\right] \quad (133)$$

Finalmente, esta equação pode ser escrita em uma forma similar à (113), como um sistema efetivo de equações algébricas lineares a ser resolvido a cada iteração de Newton-Raphson:

$$\mathbf{A}\,\Delta\Delta\hat{\mathbf{d}}^{(k)} = \mathbf{b}^{(k-1)} \quad (134)$$

onde a matriz efetiva **A** é dada por

$$\mathbf{A} = \hat{\alpha}_0\,\mathbf{M} + \mathbf{K}_T \quad (135)$$

e o vetor de resíduos efetivos $\mathbf{b}^{(k-1)}$ é definido como sendo o lado direito da eq (133). Observa-se que o objetivo foi alcançado, ou seja, no lado direito não há nenhuma operação de multiplicação com matrizes globais de rigidez ou amortecimento.

A Tabela 3 resume a implementação otimizada do procedimento de solução do problema dinâmico não-linear.

Tabela 3 – Implementação Computacional do Procedimento Otimizado

A) *Processamento Inicial para o Instante n+1*

A.1) Inicializar deslocamentos totais, incrementais, e forças elásticas:

$$\mathbf{d}_{n+1}^{(0)} = \mathbf{d}_n; \qquad \Delta\mathbf{d}^{(0)} = \mathbf{0}; \qquad \mathbf{R}(\mathbf{d}_{n+1}^{(0)}) = \mathbf{R}(\mathbf{d}_n)$$

A.2) Avaliar o vetor de cargas externas $\mathbf{F}_{n+1}$;

Instantes de Tempo com Reavaliação de Rigidez:

A.I) Atualizar a matriz de rigidez tangente $\mathbf{K}_T$

A.II) Calcular matriz efetiva **A**, equações (135), (132)

A.III) Decompor matriz efetiva **A**

A.3) Calcular parcelas preditoras das acelerações $\mathbf{a}^*$ e velocidades $\mathbf{v}^*$, eqs (120), (121)

A.4) Calcular termo constante do vetor de resíduos efetivo:

$$\mathbf{b}^* = \overset{\text{cargex}}{\mathbf{F}_{n+1}} - \alpha \overset{\text{acel.ant.}}{\mathbf{M}\,\mathbf{a}_n} - \mathbf{M}\left[(1-\alpha)\,\mathbf{a}^* + (\alpha_m - \hat{\alpha}_0\alpha_k)\,\mathbf{v}^*\right]$$

A.5) Calcular vetor de resíduos efetivo para a primeira iteração:

$$\mathbf{b}^{(0)} = \mathbf{b}^* - \mathbf{R}(\mathbf{d}_{n+1}^{(0)})$$

B) *Ciclo Iterativo N-R:* $k = 1, N_{itmax}$

B.1) Resolver o sistema efetivo: $\Delta\Delta\hat{\mathbf{d}}^{(k)} = \mathbf{A}^{-1}\,\mathbf{b}^{(k-1)}$

B.2) Determinar $\Delta\Delta\mathbf{d}^{(k)}$ pelas eqs (128) ou (129).

B.3) Atualizar deslocamentos incrementais $\Delta\mathbf{d}^{(k)}$, eq. (109) / desl. totais $\mathbf{d}_{n+1}^{(k)}$, eq. (110)

B.4) Chamar rotinas de elementos para calcular $\mathbf{R}(\mathbf{d}_{n+1}^{(k)})$.

B.5) Calcular resíduos efetivos para a próxima iteração:

$$\mathbf{b}^{(k)} = \overset{\text{const}}{\mathbf{b}^*} - \overset{\text{fresi}}{\mathbf{R}(\mathbf{d}_{n+1}^{(k)})} - \overset{\text{inerc.}}{\mathbf{M}}\left((1-\alpha)\,a_0 + (\alpha_m - \hat{\alpha}_0\alpha_k)\,a_1\right)\Delta\mathbf{d}^{(k)}$$

B.6) Verificar os critérios de convergência, encerrar o ciclo iterativo se os critérios forem atendidos.

C) *Final do Ciclo Iterativo, Processamento Final para o Instante n+1*

C.1) Atualizar as acelerações e velocidades, eqs (122), (123).

C.2) *Para intervalos de tempo selecionados*: Efetuar a gravação de resultados de deslocamentos, e/ou velocidades e/ou acelerações para graus de liberdade selecionados; e de esforços para elementos selecionados.

7.6. ASPECTOS DA IMPLEMENTAÇÃO

Os aspectos relacionados à implementação otimizada podem ser ressaltados:

Parcela Constante do Vetor de Resíduos Efetivos

Observando o lado direito da eq. (133), que define o vetor de resíduos efetivo $\mathbf{b}^{(k-1)}$, pode-se identificar alguns termos que não variam ao longo do ciclo iterativo:

- As cargas externas $\mathbf{F}_{n+1}$,
- A matriz de massa $\mathbf{M}$,
- A parcela $\alpha\ \mathbf{M}\ \mathbf{a}_n$, e
- As parcelas preditoras das acelerações $\mathbf{a}^*$ e velocidades $\mathbf{v}^*$.

Com isto, é possível determinar um termo constante $\mathbf{b}^*$, que pode ser calculado apenas uma vez, antes do início do ciclo iterativo (passo A.4 da Tabela 3). O cálculo do resíduo efetivo total para cada iteração (passo B.5) é então realizado com o uso desse termo constante.

Cálculo das Velocidades e Acelerações

Nesta implementação otimizada, não há necessidade de operar com velocidades e acelerações dentro do ciclo iterativo de Newton-Raphson. As forças de amortecimento e inércia estão incorporadas no termo constante $\mathbf{b}^*$ e nos termos apresentados no passo B.5 da Tabela 3 que envolvem os deslocamentos incrementais atualizados $\Delta\mathbf{d}^{(k)}$.

Os valores finais das velocidades e acelerações são calculados somente após a convergência do ciclo iterativo, a partir das parcelas preditoras $\mathbf{a}^*$ e $\mathbf{v}^*$, e dos valores finais dos deslocamentos incrementais $\Delta\mathbf{d}^{(k)}$ (passso C.1 da Tabela 3).

Matriz de Amortecimento

Quando se fornecem valores globais para os coeficientes de amortecimento de Rayleigh α_m e α_k, a matriz de amortecimento $\mathbf{C}$ da eq (107) não precisa ser explicitamente calculada e armazenada. Isso evita requisitos adicionais de memória, já que uma matriz de amortecimento global montada a partir das matrizes de massa e rigidez pelo uso direto da eq (107) teria o mesmo padrão de esparsidade da matriz de rigidez $\mathbf{K}_T$. Além disso, evita-se também o custo computacional das operações de multiplicação que teriam que ser efetuadas com $\mathbf{C}$.

Nesta implementação otimizada, mesmo quando o amortecimento proporcional à rigidez é considerado, com a atribuição de valores não-nulos para o coeficiente α_k da eq (107), não é necessário efetuar as multiplicações matriz-vetor envolvendo $\mathbf{K}_T$ no cálculo dos resíduos efetivo.

Evidentemente, uma matriz de amortecimento global continuará sendo montada em casos particulares quando o usuário ativar o modelo de amortecimento do solo, e/ou optar por fornecer coeficientes de amortecimento a nível de elemento. Essa última opção é útil quando o usuário dispõe de valores locais para os parâmetros α_m e α_k que não são iguais para todos os elementos da malha, e/ou valores diferentes para graus de liberdade axiais e de flexão.

Minimização dos Requisitos de Memória

Nesta implementação otimizada, requer-se memória para armazenar apenas uma matriz global simultaneamente, já que a matriz efetiva **A** pode usar a mesma área de memória reservada para a matriz tangente $\mathbf{K}_T$, e, como comentado anteriormente, a matriz de amortecimento não precisa ser montada.

Desta forma, além de uma área de memória comum para armazenar $\mathbf{K}_T$ e **A**, e para a matriz de massa **M** (que usualmente é diagonal e portanto pode ser armazenada em um vetor), os requisitos de memória desta implementação otimizada consistem apenas nos seguintes vetores de ordem *n* (número de graus de liberdade livres da malha de elementos finitos):

- Os deslocamentos totais $\mathbf{d}_{n+1}^{(k)}$ e os deslocamentos incrementais $\Delta\mathbf{d}^{(k)}$;
- As forças elásticas **R** e as cargas externas **F**;
- As acelerações e velocidades do instante de tempo anterior, $\mathbf{a}_n$ e $\mathbf{v}_n$;
- As parcelas preditoras das acelerações e velocidades $\mathbf{a}^*$ e $\mathbf{v}^*$;
- O termo constante do resíduo efetivo $\mathbf{b}^*$ e o resíduo efetivo total $\mathbf{b}^{(k-1)}$;
- A variação efetiva e a variação final dos deslocamentos incrementais, $\Delta\Delta\hat{\mathbf{d}}^{(k)}$ e $\Delta\Delta\mathbf{d}^{(k)}$;
- Os valores finais para as acelerações e velocidades, $\mathbf{a}_{n+1}$ e $\mathbf{v}_{n+1}$.

No entanto, é fácil observar que o algoritmo não requer que todas estas quantidades estejam disponíveis simultaneamente em todos os passos. A otimização de

memória pode ser alcançada com a alocação sucessiva de diferentes quantidades para as mesmas posições de memória, resultando assim em um número total de 6 vetores para armazenar as 14 quantidades:

- Um único vetor para $\mathbf{F}$ e $\mathbf{b}^*$;
- Um único vetor para: $\mathbf{R}$, $\mathbf{b}^{(k-1)}$, $\Delta\Delta\hat{\mathbf{d}}^{(k)}$, e $\Delta\Delta\mathbf{d}^{(k)}$;
- Um único vetor para: $\mathbf{a}_n$, $\mathbf{a}^*$ e $\mathbf{a}_{n+1}$;
- Um único vetor para: $\mathbf{v}_n$, $\mathbf{v}^*$ e $\mathbf{v}_{n+1}$;
- Dois diferentes vetores para, respectivamente, $\mathbf{d}_{n+1}^{(k)}$ e $\Delta\mathbf{d}^{(k)}$.

8. OTIMIZAÇÕES PARA MÉTODOS EXPLÍCITOS DE INTEGRAÇÃO

8.1. INTRODUÇÃO

Conforme apresentado anteriormente, algoritmos explícitos como o Método das Diferenças Centrais requerem o uso de um intervalo de tempo Δt *menor* do que um determinado valor crítico Δt_{cr}. Por sua vez, o valor do intervalo de tempo crítico Δt_{cr} é função de T_n, o menor período natural do modelo estrutural, discretizado pela malha de elementos finitos. No caso do Método das Diferenças Centrais, o intervalo de tempo crítico é dado por:

$$\Delta t_{cr} = \frac{T_n}{\pi} = \left(\frac{2\,\pi}{\omega_n}\right)\frac{1}{\pi} = \frac{2}{\omega_n} \tag{136}$$

onde ω_n é a última (maior) frequência natural da malha, em rad/s.

Caso se esteja trabalhando com uma malha cuja discretização é não uniforme, isto é, com elementos de diferentes comprimentos ao longo da malha, ou composta por materiais com diferentes propriedades, seríamos forçados a adotar o intervalo de tempo ditado pelo menor elemento da malha ou pelo material com maior rigidez.

8.2. SUBCICLAGEM

A subciclagem consiste no uso de diferentes intervalos de tempo para diferentes grupos de elementos e nós podendo-se eliminar a necessidade de atualização de todos os graus de liberdade da malha com o intervalo de tempo do menor elemento desta malha ou do material com maior rigidez. A utilização da subciclagem pode introduzir uma economia computacional significativa à solução do problema.

No contexto de uma análise acoplada, podem ser identificados dois tipos principais de procedimentos de subciclagem:

- Subciclagem Casco-Linhas;
- Subciclagem Interna das Linhas.

Os itens a seguir descrevem cada um destes tipos.

8.2.1. Subciclagem Casco-Linhas

No procedimento de análise acoplada implementada em versões anteriores do Prosim, a cada instante do processo de integração no tempo das equações de movimento do casco efetuava-se uma análise não-linear dinâmica de cada linha sob a ação da onda, correnteza, peso próprio e das componentes de movimentos do casco.

No entanto, como em geral os intervalos de tempo requeridos para a análise dinâmica das linhas de ancoragem e risers são menores do que os intervalos de tempo para a análise do casco, foi introduzido o conceito de subciclagem casco-linha, o qual torna possível utilizar um passo de integração para as linhas NSUBDT vezes menor que o passo de integração para o casco (Figura 18), onde NSUBDT é um número inteiro. Desta forma, a cada avanço na solução do casco faz-se um ou mais avanços na solução das linhas.

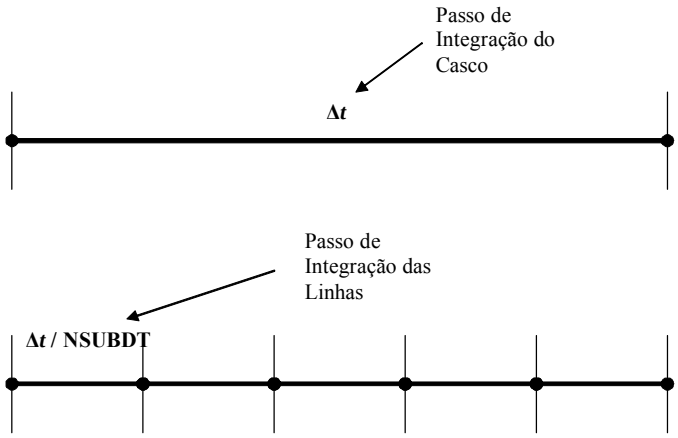


Figura 18 – Subciclagem casco-linha

Certamente esta modificação proporcionou ganho de desempenho do programa, possibilitando utilizar um passo de integração maior para o casco sem comprometer os resultados da análise dinâmica das linhas.

8.2.2. Subciclagem Interna das Linhas

A subciclagem interna permite a utilização de múltiplos intervalos de tempo na análise dinâmica de uma determinada linha. Neste caso, o domínio é dividido em diferentes subdomínios cujo intervalo de tempo requerido é definido pela malha ou material deste subdomínio.

Cada subdomínio é atualizado de acordo com o seu intervalo de tempo. Assim, caso tenhamos 2 subdomínios cuja razão entre os intervalos de tempo seja *r*, enquanto o subdomínio com maior intervalo de tempo é integrado uma vez o outro subdomínio será integrado *r* vezes, ou seja em *r* subciclos.

O maior problema observado na implementação da subciclagem [32] é a interface entre os subdomínios. Neste caso o nó de interface é atualizado com o maior intervalo de tempo dos subdomínios adjacentes. Para integração dos demais subdomínios adjacentes com menores intervalos de tempo assume-se que a velocidade ou aceleração neste nó de interface permanece constante durante os subciclos. Assim torna-se possível o cálculo dos esforços internos dos elementos adjacentes a este nó e pertencentes aos subdomínios com subciclos.

Smolinski [33] utiliza velocidade constante nos trechos não-atualizados e Daniel [32] utiliza aceleração constante nos trechos não-atualizados. Observou-se problemas de convergência na utilização de aceleração constante nos trechos não-atualizados [25]. Na

implementação efetuada utilizou-se velocidade constante nos nós de interface durante os subciclos.

Supondo 2 subdomínios cuja razão entre os intervalos de tempo seja r e que Δt seja o menor intervalo de tempo utilizado entre os subdomínios. Assim teremos um subdomínio com intervalo de tempo Δt e o outro subdomínio com intervalo de tempo $r\Delta t$.

A utilização da velocidade constante consiste em que, durante o cálculo das velocidades na interface, é utilizado o intervalo de tempo $r\Delta t$, e estes valores permanecem constantes durante os subciclos de integração do subdomínio com intervalo de tempo Δt.

Na implementação aqui efetuada, são utilizados 3 contadores de tempo e razões inteiras entre os intervalos de tempo dos subdomínios. Belytshko e Lu em [34] apresentam um algoritmo seguindo esta mesma filosofia de implementação. Devido a utilização de contadores de tempo, possivelmente não haveriam maiores problemas na utilização de intervalos de tempo com razões não-inteiras, porém este procedimento não foi avaliado no presente estudo.

8.2.3. Aspectos da Implementação

O esquema de subciclagem implementado consiste na utilização de 3 contadores de tempo:

- Contador de tempo global;

- Contador de tempo para os nós (graus de liberdade);

- Contador de tempo para os elementos.

No arquivo de entrada de dados é fornecido, para cada elemento, um intervalo de tempo, sendo que, na atual implementação, a razão entre os intervalos de tempo de cada subdomínio deve ser uma fração inteira dos demais. O intervalo de tempo utilizado nos graus de liberdade da interface é o maior intervalo de tempo entre os elementos adjacentes a este nó.

Conforme mencionado anteriormente, o intervalo de tempo de cada subdomínio deve ser menor que o intervalo de tempo crítico daquele subdomínio.

Definidos os intervalos de tempo de cada grupo de nós e elementos inicia-se o processo de integração de cada subdomínio. A atualização de um subdomínio só ocorrerá caso o contador de tempo de nós / elemento daquele grupo estiver com um tempo de simulação menor que o contador de tempo global.

O procedimento está representado a seguir:

- Passo 1: Inicializar variáveis;

- Passo 2: Inicializar contadores de tempo:

$t_{glob} = 0$;

$t_{ele} = 0$, para todos os elementos;

$t_{no} = 0$, para todos os graus de liberdade dos nós;

- Passo 3: Associar Δt_{ele} para todos os elementos e Δt_{no} para todos os graus de liberdade dos nós;

- Passo 4: Calcular forças nodais equivalentes:

(a) zerar vetor de forças;

(b) loop em todos os elementos;

(c) caso $t_{ele} < t_{glob}$ calcular forças para o elemento e fazer $t_{ele} = t_{ele} + \Delta t_{ele}$;

(d) montar vetor de forças nodais equivalentes;

(e) fim do loop.

- Passo 5: Atualizar nós:

(a) loop em todos os graus de liberdade;

(b) caso $t_{no} < t_{glob}$ calcular deslocamentos, velocidades e acelerações integrando-se as equações de movimento e fazer $t_{no} = t_{no} + \Delta t_{no}$;

(c) aplicar condições de contorno.

- Passo 6: se $t_{glob} < t_{final}$ voltar para Passo 3.

8.2.4. Subciclagem Interna na Análise Acoplada

A implementação da subciclagem das linhas em conjunto com análise da plataforma, isto é na análise acoplada, requer alguns cuidados devido à integração do casco.

Silveira em [25] implementa a subciclagem aplicada à análise dinâmica de linhas de ancoragem, porém analisadas individualmente.

Em análises acopladas, conforme pode ser observado no item 3.4, a integração do casco é feita utilizando-se o Método de Runge-Kutta de 4ª ordem. Neste método as derivadas das velocidades lineares e angulares são calculadas no início do tempo presente utilizando-se valores de variáveis do intervalo de tempo anterior, além de duas estimativas que são obtidas para o ponto médio do intervalo de tempo e uma estimativa de derivada que é obtida no final do intervalo de tempo.

A novidade da implementação da subciclagem em análises acopladas consiste na atualização dos contadores de tempo de elemento e nós apenas quando as derivadas forem calculadas no início do tempo presente e na segunda estimativa do ponto médio do intervalo de tempo.

Desta forma pode-se utilizar a subciclagem casco-linha com diferentes intervalos de tempo na integração do casco e das linhas em conjunto com a subciclagem interna das linhas onde também diferentes intervalos de tempo podem ser utilizados nos diferentes segmentos de uma linha.

8.2.5. Comentários

Alguns comentários relevantes são apresentados a seguir:

- Quanto maior a diferença entre o tamanho dos elementos entre subdomínios maior será a diferença de intervalo de tempo requerida para integração de cada subdomínio. Neste caso a utilização da subciclagem poderá trazer ganhos significativos em termos de custo computacional;
- Quanto maior a quantidade de elementos nos subdomínios não atualizados maiores também serão as vantagens em termos de custo computacional da utilização da subciclagem;
- Em malhas com número reduzidos de nós e elementos, onde a discretização é aproximadamente uniforme a utilização da subciclagem provavelmente não trará nenhum benefício.
- A subciclagem aplicada a análise acoplada pode trazer um ganho ainda maior que a subciclagem aplicada a uma linha individualmente. Isto porque, em uma análise acoplada, existe um conjunto de linhas com diferentes discretizações ou diferentes tipos de material (p.ex. linhas de ancoragem) às quais podem-se aplicar diferentes intervalos de tempo.

8.3. EXEMPLOS

8.3.1. Viga Biengastada sob Carga Transversal

Descrição do Modelo

O objetivo deste exemplo é somente a verificação da precisão das respostas sem preocupação com desempenho visto a pouca quantidade de elementos e o curto período de simulação envolvido nesta análise.

O problema considerado nesta seção, ilustrado na Figura 19 a seguir [35], consiste em uma viga biengastada submetida a uma carga vertical aplicada transversalmente. Com isto, são excitados modos naturais de frequência mais baixa, associados ao comportamento de flexão da viga.

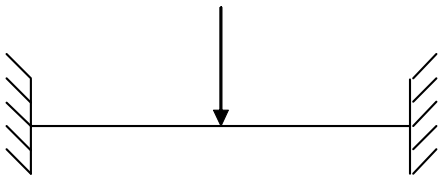


Figura 19 – Viga biengastada sob Carga Tranversal

Dados Geométricos e do Material

A Tabela 4 reproduz os dados geométricos e do material.

Tabela 4 – Propriedades

A (pol^2)	0.2
ρ ($lb.s^2/pol^4$)	0.000797
E (lb/pol^2)	10^7
I (pol^4)	0.000667
Comprimento Lt (pol)	20

Basicamente, o modelo numérico consiste em uma malha uniforme de 20 elementos de pórtico; cada elemento tem portanto um comprimento igual a L = 1 pol.

Carregamento

O carregamento consiste em uma carga aplicada transversalmente, com valor de 640 lb aplicada no meio do vão.

O valor total da carga é aplicado de forma instantânea, atuando de forma constante até o final da análise.

Resultados

Para utilização da subciclagem, neste exemplo, dividiu-se a viga em três segmentos com diferentes tamanhos de elementos conforme ilustrado na figura 20 a seguir:

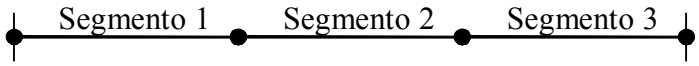


Figura 20 – Discretização da viga

Os segmentos 1 e 3 são constituídos de 5 elementos de 1 pol cada. O segmento 2 é constituído de 20 elementos de 0.5 pol cada.

Desta forma utilizou-se dois intervalos de tempo correspondentes a cada segmento:

- Segmentos 1 e 3: Δt = 2.5e-07 seg;

- Segmento 2: Δt = 1.25e-07 seg.

Na análise tradicional, sem subciclagem, utilizou-se o menor intervalo de tempo para todos os elementos, Δt = 1.25e-07 seg.

Não foi utilizado amortecimento numérico nesta análise.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim, sem o uso da subciclagem e a versão implementada com uso da subciclagem. Conforme pode-se observar na figura 21 os resultados são praticamente idênticos.

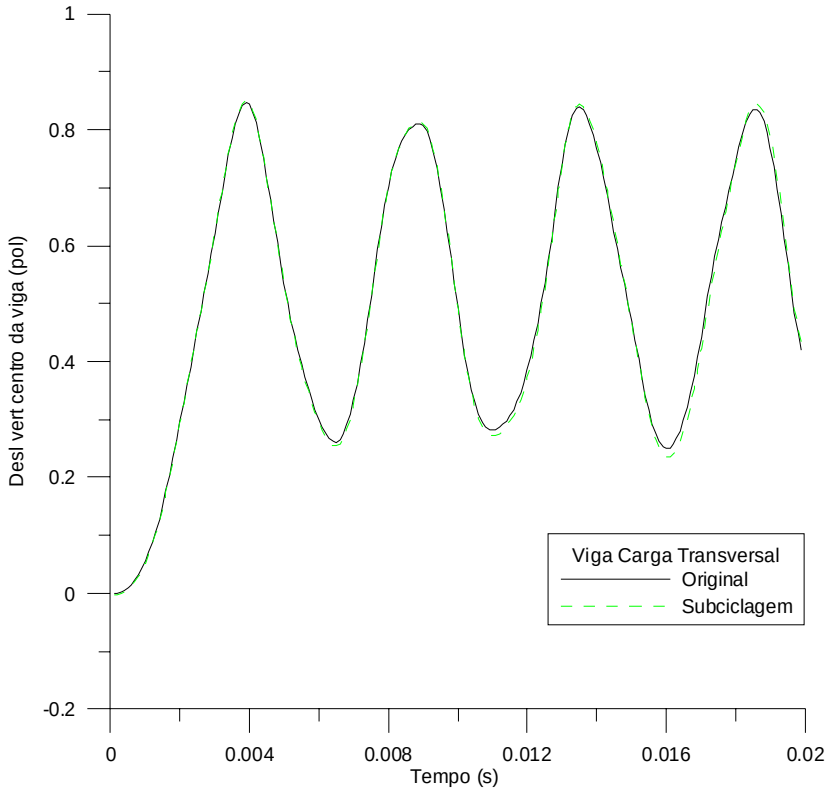


Figura 21 – Resultados comparativos

8.3.2. Viga Monoengastada sob Carga Axial

Este exemplo tem os mesmos dados, propriedades físicas, propriedades do material e carregamento do exemplo apresentado no item 6.5.

O objetivo deste exemplo é somente a verificação da precisão das respostas sem preocupação com desempenho visto a pouca quantidade de elementos e o curto período de simulação envolvido nesta análise.

Resultados

Para utilização da subciclagem, neste exemplo, dividiu-se a viga em dois segmentos com diferentes tamanhos de elementos conforme ilustrado na figura 22 a seguir:

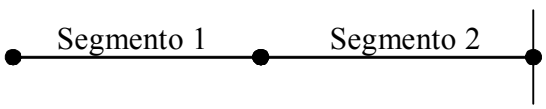


Figura 22 – Discretização da viga

O segmento 1 é constituído de 40 elementos de 0.25 pol cada. O segmento 2 é constituído de 20 elementos de 0.5 pol cada.

Desta forma utilizou-se dois intervalos de tempo correspondentes a cada segmento:

- Segmento 1: Δt = 1.2416 e–06 seg;

- Segmento 2: Δt = 2.4833 e–06 seg.

Na análise tradicional, sem subciclagem, utilizou-se o menor intervalo de tempo para todos os elementos, Δt = 1.2416 e–06 seg.

Não foi utilizado amortecimento numérico nesta análise.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim, sem o uso da subciclagem e a versão implementada com uso da subciclagem. Conforme pode-se observar na figura 23 os resultados são praticamente idênticos.

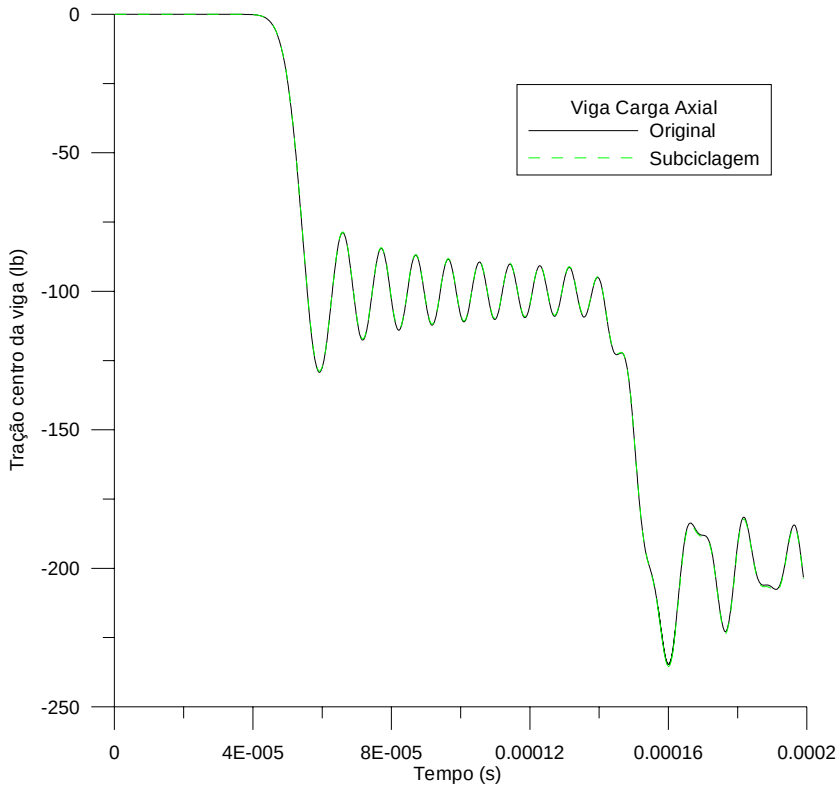


Figura 23 – Resultados comparativos

9. OTIMIZAÇÕES PARA MÉTODOS IMPLÍCITOS DE INTEGRAÇÃO

9.1. PARTIÇÃO DO DOMÍNIO

Nos algoritmos implícitos o estudo de otimização será efetuado através da partição de domínios. Tais técnicas eram empregadas, originalmente, na análise de sistemas rígido-flexíveis caracterizados por regiões (domínios) com propriedades físicas bem distintas, como em problemas de interação estrutura-fluido ou estrutura-solo. Nesta categoria de partição de domínio, incluem-se:

- os Métodos de partição implícito-explícito [36] que empregam um operador implícito na região rígida e um operador explícito na região flexível. A implementação mais eficiente para estes métodos envolve uma "partição por elementos", onde os elementos são divididos em implícitos e explícitos. O acoplamento entre as regiões ocorre como consequência do processo de montagem das matrizes globais a partir das contribuições dos elementos.

- os Métodos de subciclagem [37, 38] que empregam o mesmo operador em todas as regiões (implícito ou, mais geralmente, explícito, conforme apresentado no capítulo anterior), porém diferentes intervalos de tempo para cada região. Tais métodos têm sido aplicados preferencialmente em problemas de propagação de ondas.

- Métodos "Partição de Domínio Implícito – PDI" [39] e "Partição de Domínio Implícito Iterativo – PDII" [40]. Neste caso a partição de domínio será efetuada com o intuito de otimização na análise dinâmica de estruturas offshore, decorrente da utilização dos operadores implícitos em todas as regiões, aplicados a computadores com arquitetura paralela. Estes métodos serão descritos nos próximos subitens. Esta partição é efetuada em estruturas de risers e linhas de ancoragem. Assim, uma estrutura composta de uma linha de ancoragem ou riser é dividida em várias subestruturas. Cada trecho de linha, ou subestrutura é analisada separadamente em cada processo, para, em seguida, compatibilizar-se os nós de interface e obtenção da resposta.

9.2. ALGORITMO "PARTIÇÃO DE DOMÍNIO IMPLÍCITO" (PDI)

Apresentando novamente a equação de movimentos:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{F}(t) \tag{137}$$

Onde **M**, **C** e **K** representam, respectivamente, as matrizes de massa, amortecimento e rigidez; **F** é o vetor de forças externas, e as incógnitas são os vetores $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$ e **u** contendo, respectivamente, as acelerações, velocidades e deslocamentos nodais.

Esta equação pode ser reduzida a uma equação de primeira ordem através de uma mudança de variáveis. Apresentando:

$$\mathbf{z} = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} \tag{138}$$

A equação (137) toma a seguinte forma:

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{K} \\ -\mathbf{K} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}(t) \\ \mathbf{u}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(t) \\ 0 \end{bmatrix} \tag{139}$$

$$\begin{bmatrix} \dot{\mathbf{u}}(0) \\ \mathbf{u}(0) \end{bmatrix} = \begin{bmatrix} v_0 \\ d_0 \end{bmatrix}$$

Na notação matricial o sistema de equações reduzidas (139) pode ser escrito como:

$$\mathbf{A}\dot{\mathbf{z}}(t) + \mathbf{B}\mathbf{z}(t) = \mathbf{g}(t) \tag{140}$$

$$\mathbf{z}(0) = \mathbf{z}_0$$

O primeiro passo na construção do método é a partição da estrutura em grupos de elementos. A malha de elementos finitos pode ser vista como uma coleção de sub-domínios desconectados. Os campos de variáveis de um subdomínio genérico r são descritos em termos de vetores locais , como por exemplo z^r.

O vetor de variáveis estendidas $\bar{\mathbf{z}} = \{z^1, z^2, ..., z^r, ..., z^s\}$, onde s é o número de subdomínios, descreve completamente a estrutura. De um modo geral $\bar{\mathbf{z}}$ contém as mesmas informações que **z**, o vetor nodal global. As relações entre eles são dadas pelo seguinte mapeamento:

$$\mathbf{z} = \mathbf{L}^T \bar{\mathbf{z}} \tag{141}$$

Onde **L** é uma matriz booleana que localiza **z** no subdomínio r para obtenção de $\mathbf{z}^r$.

De forma análoga:

$$\mathbf{g} = \mathbf{L}^T \bar{\mathbf{g}} \tag{142}$$

Onde **g** e $\bar{\mathbf{g}}$ são os vetores de força global e estendidos $\{g^1,..., g^r, ..., g^s\}$. As matrizes estendidas correspondentes a $\bar{\mathbf{z}}$ são definidas como:

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}^1 & & & \\ & \mathbf{A}^2 & & \\ & & . & \\ & & & \mathbf{A}^s \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B}^1 & & & \\ & \mathbf{B}^2 & & \\ & & . & \\ & & & \mathbf{B}^s \end{bmatrix} \tag{143}$$

A operação de montagem dos vetores globais podem ser expressas da seguinte forma:

$$\mathbf{A} = \mathbf{L}^T \bar{\mathbf{A}}\, \mathbf{L} \tag{144}$$

$$\mathbf{B} = \mathbf{L}^T \bar{\mathbf{B}}\, \mathbf{L}$$

A idéia essencial deste método é permitir com que os vários sub-domínios da partição consigam evoluir independentemente, e, garantir-se a compatibilidade entre sub-domínios através de alguma projeção da solução estendida.

Por simplicidade, consideremos o caso sem carregamento externo, g = 0. Uma classe geral de algoritmos paralelos pode ser definida da seguinte forma:

- Localização das condições iniciais $\mathbf{z}_0$ nos sub-domínios para obtenção do vetor estendido $\bar{\mathbf{z}}_0$;
- Atualização do vetor estendido através da solução das equações de movimento desacopladas ao nível de sub-domínio;

$$\mathbf{A}^r \ddot{\mathbf{z}}^r + \mathbf{B}^r \mathbf{z}^r = \mathbf{0} \tag{145}$$

Para resolução do sistema de equações apresentado torna-se necessária a utilização de uma parcela preditora estendida denominada $\bar{\mathbf{z}}^{*\,r}_{n+1}$;

- As parcelas preditoras estendidos assumem múltiplos valores nos nós que pertencem a mais de um sub-domínio. O algoritmo é então completo pela ponderação destes valores nos nós de interface através de uma regra de ponderação, então a consistência é restaurada.

Portanto, a regra de ponderação pode ser escrita como:

$$\mathbf{z}_{n+1} = \mathbf{A}^{-1} \sum_{r=1}^{s} \mathbf{A}^r \bar{\mathbf{z}}^{*\,r}_{n+1} \tag{146}$$

Então $\bar{\mathbf{z}}^{*\,r}_{n+1}$ é primeiramente ponderado pelas matrizes $\mathbf{A}^r$, os vetores locais rearranjados em um vetor global é finalmente multiplicado por $\mathbf{A}^{-1}$. Portanto, a consistência deste método é garantida através desta operação.

No algoritmo PDI descrito até então, a compatibilidade entre sub-domínios utiliza a matriz **A** e é feita através de deslocamentos [41] nos graus de liberdade de interface. A regra de ponderação de massa utiliza a matriz de massa **M** na ponderação, conforme apresentada a seguir:

$$\mathbf{d}_{n+1} = \mathbf{M}^{-1} \sum_{r=1}^{s} \mathbf{M}^r \mathbf{d}^{*\,r}_{n+1} \tag{147}$$

Uma variação deste algoritmo proposta por Sotelino [39] propõe que, a compatibilidade seja efetuada em termos de acelerações. Portanto, para restaurar a compatibilidade entre subdomínios a regra de ponderação da massa é aplicada:

$$\mathbf{a}_{n+1} = \mathbf{M}^{-1} \sum_{r=1}^{s} \mathbf{M}^r \mathbf{a}^{*\,r}_{n+1} \tag{148}$$

A implementação efetuada no PROSIM utiliza a regra de ponderação de massa em termos de compatibilidade de deslocamento.

9.3. ALGORITMO "PARTIÇÃO DE DOMÍNIO IMPLÍCITO ITERATIVO" (PDII)

Problemas de precisão foram encontrados na aplicação do algoritmo PDI em determinados casos [42]. Estas imprecisões de resultados aparecem devido à utilização da regra de ponderação da massa, usada para garantir a compatibilidade de deslocamento ou aceleração dos graus de liberdade de interface, porém não mais garantindo o equilíbrio da estrutura.

Para resolver este problema, foi proposto um algoritmo PDI iterativo [40] (PDII) ainda voltado para o tratamento de problemas lineares. Neste algoritmo um procedimento iterativo é utilizado para garantir o equilíbrio nos graus de liberdade da interface quando da utilização da regra de ponderação da massa.

Neste método, da mesma forma como apresentado no Método PDI, a estrutura original é particionada em um determinado número de subdomínios. Cada subdomínio é resolvido independentemente e de forma concorrente, usando um método direto tradicional de resolução de sistemas de equações.

O algoritmo PDII é uma extensão do algoritmo PDI onde também se utiliza o método de compatibilidade de deslocamentos utilizado nos nós de interface. Entretanto, ao contrário do algoritmo PDI, no algoritmo PDII é utilizado um processo iterativo para restaurar o equilíbrio dos graus de liberdade da interface. Desta forma, espera-se, ao final das iterações do Método PDII, que o resultado obtido seja praticamente o mesmo da análise feita com a estrutura sem particionamento.

Uma comparação esquemática entre os algoritmos é apresentada a seguir. Note que até o momento os algoritmos são aplicados a problemas lineares:

- Algorimo PDI:

(a) Condições iniciais de deslocamento, velocidade e aceleração;

(b) Montagem da matriz efetiva e do vetor de resíduos efetivos considerando forças externas e internas;

(c) Resolução do sistemas de equações de cada subdomínio independentemente (compatibilidade da interface não é satisfeita);

(d) Obtenção de compatibilidade de deslocamento na interface através da utilização da regra de ponderação de massa (agora, o equilíbrio deixa de ser atendido, pois os esforços nos elementos de interface não são atualizados).

- Algorimo PDII:

(a) Condições iniciais de deslocamento, velocidade e aceleração;

(b) Montagem da matriz efetiva e do vetor de resíduos efetivos considerando forças externas e internas;

(c) Resolução do sistemas de equações de cada subdomínio independentemente (compatibilidade da interface não é satisfeita);

(d) Obtenção de compatibilidade de deslocamento na interface através da utilização da regra de ponderação de massa;

(e) Verificação de convergência do vetor de deslocamentos da interface;

(f) Reavaliação dos esforços internos dos elementos de interface, em cada subdomínio;

(g) Reavaliação do vetor de resíduos efetivos dos elementos da interface, em cada subdomínio;

(h) Montagem do vetor de resíduos efetivos global da interface;

(i) Verificação de convergência do vetor de resíduos efetivos global da interface;

(j) Se a convergência não foi atingida, deve-se redistribuir o resíduo efetivo entre os graus de liberdade da interface e voltar para o passo (c).

Desta forma, concluído o ciclo iterativo o equilíbrio está restaurado.

9.4. ALGORITMO "PARTIÇÃO DE DOMÍNIO IMPLÍCITO ITERATIVO" (PDII) PARA PROBLEMAS NÃO-LINEARES

Os algoritmos PDI e PDII descritos até o momento foram apresentados originalmente em [40] para problemas lineares. Para problemas com não-linearidade geométrica, que são os problemas aqui estudados, propõe-se neste trabalho tirar proveito do ciclo de iterações Newton-Raphson, utilizado para resolver a não-linearidade (ver item 7.3), para restaurar o equilíbrio dos graus de liberdade da interface. Com isso, em problemas não-lineares o algoritmo PDII não só será mais preciso do que o PDI, mas também apresentará uma maior eficiência, visto que o processo iterativo deve ser efetuado mesmo para problemas sem partição de domínio. A seguir uma descrição do algoritmo:

- Algorimo PDII para problemas não-lineares:

(a) Condições iniciais de deslocamento, velocidade e aceleração;

(b) Montagem da matriz efetiva e do vetor de resíduos efetivos considerando forças externas e internas;

(c) Resolução do sistemas de equações de cada subdomínio independentemente (compatibilidade da interface não é satisfeita);

(d) Obtenção de compatibilidade de deslocamento na interface através da utilização da regra de ponderação de massa;

(e) Verificação de convergência do vetor de deslocamentos;

(f) Reavaliação dos esforços internos de todos os elementos, inclusive os elementos de interface, em cada subdomínio;

(g) Reavaliação do vetor de resíduos efetivos de cada subdomínio;

(h) Montagem do vetor de resíduos efetivos global da interface;

(i) Verificação de convergência do vetor de resíduos efetivos global;

(j) Se a convergência não foi atingida, deve-se redistribuir o resíduo efetivo entre os graus de liberdade da interface e voltar para o passo (c).

Desta forma, concluído o ciclo iterativo o equilíbrio está restaurado.

9.5. ASPECTOS DA IMPLEMENTAÇÃO

Este item apresentará o procedimento seguido para implementação do algoritmo PDII em problemas não-lineares no programa Prosim.

9.5.1.1. Sistema de Equações

O programa Prosim tem, em relação à implementação do armazenamento da matriz efetiva, a filosofia de armazenamento dos elementos diferentes de zero da matriz descrito a seguir:

- inicialmente é montado o vetor BDIAG que contém os elementos da diagonal da matriz efetiva;

- em seguida é montado o vetor BP que contém os elementos de fora da diagonal da matriz efetiva;

- então é montado o vetor IPOS que define a posição dos elementos pertencentes ao vetor BP na matriz efetiva global;

- o vetor VRESEF contém inicialmente os elementos de força externa componentes do lado direito do sistema de equações e, após a resolução do sistema de equações, os deslocamentos resultantes são armazenados no próprio vetor VRESEF.

Portanto, para cada subestrutura serão montados os vetores IPOS, BDIAG, BP e VRESEF. Assim caso a estrutura esteja dividida em N subestruturas, haverão N sistemas de equações a serem resolvidas. Cada sistema de equações terá apenas os graus de liberdade referentes a cada subestrutura. Assim, caso a estrutura seja dividida em 2 ou mais subdomínios, cada sistema de equações a ser resolvido será sempre menor que o sistema de equações da estrutura original.

Os sistemas de equações são resolvidos pelo Método de Eliminação de Gauss utilizando-se técnicas de triangularização e retro-substituição da matriz efetiva.

Os sistemas de equações das subestruturas podem ser resolvidos independentemente sendo necessária a comunicação entre subdomínios apenas para compatibilização da interface.

Conforme pode-se notar, na utilização do PDI, o nó de interface terá dois valores calculados, um em cada subdomínio. Para compatibilizar-se estes valores é utilizada a regra de ponderação da massa, já descrito no subitem anterior.

No programa Prosim o Método Implícito é implementado por deslocamentos, assim a regra de ponderação pode ser escrita como:

$$\mathbf{d}_{n+1} = \mathbf{M}^{-1} \sum_{r=1}^{s} \mathbf{M}^{r} \mathbf{d}^{*\,r}_{n+1} \tag{149}$$

onde d^* é o deslocamento obtido em cada subdomínio, s é o número de subdomínio concorrentes, M é a massa correspondente ao grau de liberdade e d é o deslocamento já compatibilizado na interface.

9.5.1.2. Reavaliação de Elementos

Como as aplicações em questão se tratam de linhas de ancoragem ou risers, tem-se por conseqüência uma largura de banda pequena da matriz efetiva.

Em problemas onde a matriz efetiva tem largura de banda pequena, os elementos diferentes de zero estão localizados próximos à diagonal. Neste tipo de problema o custo da solução do sistema de equações não é tão superior aos demais procedimentos de cálculo.

A reavaliação dos elementos é feita através de um "loop", ou ciclo, em todos os elementos onde reavalia-se a matriz de rigidez / efetiva e são calculados os esforços internos da estrutura. Esta etapa de cálculo corresponde a um elevado percentual do tempo total de simulação, devido às características dinâmicas não-lineares do problema em questão. Assim a reavaliação dos elementos é também uma parcela importante no tempo de simulação do processo de análise dinâmica a ser particionado. Neste ciclo serão reavaliados apenas os elementos referentes a cada subdomínio.

Os dois trechos particionados até o momento no programa PROSIM correspondem a cerca de 80% do tempo de CPU gasto numa análise dinâmica, conforme poderá ser observado nas aplicações numéricas, divididos da seguinte forma:

- Resolução do sistema de equações: 30% do tempo total de CPU;

- Ciclo de elementos para reavaliação da rigidez e cálculo de esforços internos: 50% do tempo total de CPU.

9.5.1.3. Partição dos demais trechos do código

Como o sistema de equações, a reavaliação da rigidez e o cálculo dos esforços internos são particionados todos os demais procedimentos do programa Prosim relativos à análise dinâmica da linha podem também ser particionados sem prejuízo no resultado final da análise. Estes outros procedimentos são listados a seguir:

- Reavaliação das forças externas;
- Reavaliação dos deslocamentos incrementais;
- Reavaliação do vetor de deslocamentos totais;
- Reavaliação do vetor de velocidades;
- Reavaliação do vetor de acelerações.

Com este procedimento cerca de 90% do tempo total de CPU gasto numa análise dinâmica pode ser realizado concorrentemente e simultaneamente. Assim torna-se possível a partição de uma estrutura em subdomínios e a execução de uma análise dinâmica de forma independente e concorrente através da utilização de computadores com arquitetura em paralelo.

9.5.1.4. Verificação da convergência

A verificação da convergência é feita através da comparação de um valor de tolerância de deslocamento ou força, fornecido no arquivo de entrada, com a norma do vetor de deslocamentos ou força obtido na análise a cada iteração de Newton-Raphson.

A norma de um vetor *v* com *n* componentes é dada pela seguinte expressão:

$$||v(n)|| = \sqrt{\sum_{i=1}^{n} v(i)^2} \tag{150}$$

Porém, com a partição do domínio, teremos, em cada subdomínio, um vetor *v* com *p* componentes em *q* subdomínios. Para verificação da convergência da estrutura é necessária uma soma das normas calculadas em cada subdomínio, para que a norma do vetor a ser utilizada na verificação da convergência seja a norma de toda a estrutura e não apenas de cada subdomínio. Assim a norma é calculada da seguinte forma:

$$||v(n)|| = \sqrt{\sum_{j=1}^{q}\sum_{i=1}^{p} v(i)^2} \tag{151}$$

9.6. EXEMPLOS

9.6.1. Viga Biengastada sob Carga Transversal

Este exemplo tem os mesmos dados, propriedades físicas, propriedades do material e carregamento do exemplo apresentado no item 8.3.

O objetivo deste exemplo é somente a verificação da precisão das respostas sem preocupação com desempenho visto a pouca quantidade de elementos e o curto período de simulação envolvido nesta análise.

Resultados

Para utilização do algoritmo PDII, neste exemplo, dividiu-se a viga ao meio em dois segmentos conforme ilustrado na figura 24 a seguir:

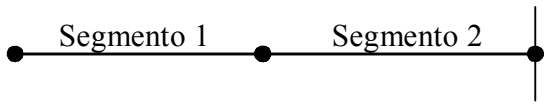


Figura 24 – Discretização da viga

Cada segmento é constituído de 10 elementos de 1 pol cada. Utilizou-se elementos de pórtico nesta modelagem.

A seguir são apresentados os principais parâmetros utilizados nesta análise:

- Intervalo de tempo: 5.e-06 seg;

- Tolerância de deslocamentos: 0.001

- Tolerância de forças: 0.01

- Amortecimento numérico: não foi considerado.

- Amortecimento proporcional à massa: não foi considerado.

- Amortecimento proporcional à rigidez: não foi considerado.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim e a versão implementada com uso do PDII. Conforme pode-se observar na figura 25 os resultados são praticamente idênticos.

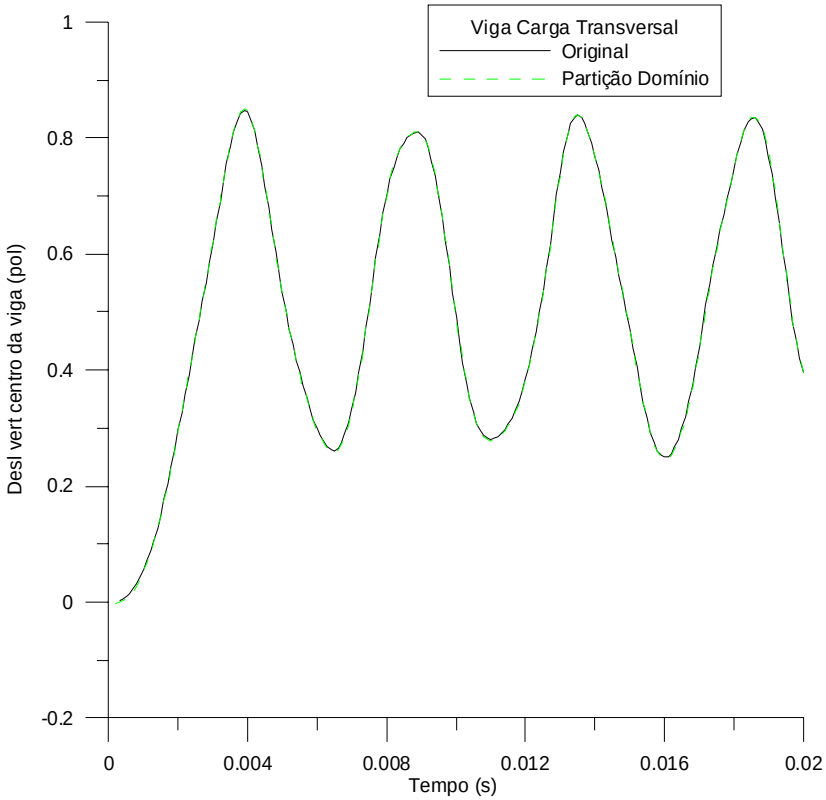


Figura 25 – Resultados comparativos

9.6.2. Viga Monoengastada sob Carga Axial

Este exemplo tem os mesmos dados, propriedades físicas, propriedades do material e carregamento do exemplo apresentado no item 6.5.

Da mesma forma objetivo deste exemplo é somente a verificação da precisão das respostas sem preocupação com desempenho visto a pouca quantidade de elementos e o curto período de simulação envolvido nesta análise.

Resultados

Para utilização do algoritmo PDII, neste exemplo, dividiu-se a viga ao meio em dois segmentos conforme ilustrado na figura 26 a seguir:

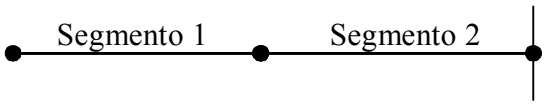


Figura 26 – Discretização da viga

Cada segmento é constituído de 20 elementos de 0.5 pol cada. Utilizou-se elementos de pórtico nesta modelagem.

A seguir são apresentados os principais parâmetros utilizados nesta análise:

- Intervalo de tempo: 2.4833 e−06 seg;

- Tolerância de deslocamentos: 0.001

- Tolerância de forças: 0.01

- Amortecimento numérico: não foi considerado.

- Amortecimento proporcional à massa: não foi considerado.

- Amortecimento proporcional à rigidez: não foi considerado.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim e a versão implementada com uso do PDII. Conforme pode-se observar na figura 27 os resultados são praticamente idênticos.

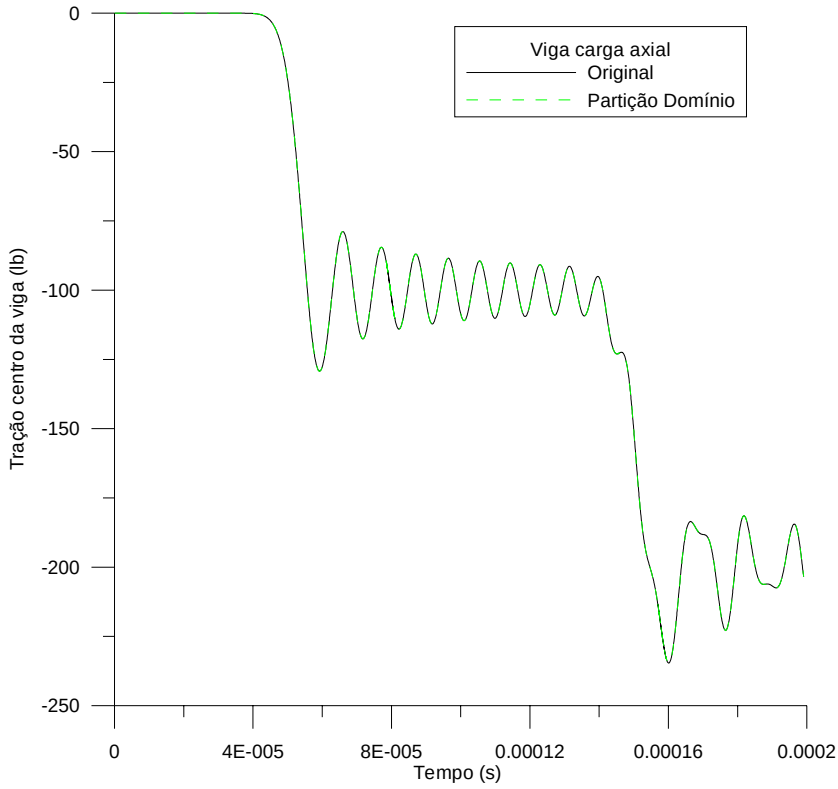


Figura 27 – Resultados comparativos

10. IMPLEMENTAÇÃO DO MÉTODO PDII EM COMPUTADORES COM ARQUITETURA PARALELA

10.1. AMBIENTE DE COMPUTAÇÃO PARALELA

10.1.1. Introdução

Este item tem o propósito de apresentar os conceitos básicos relacionados à programação em um ambiente paralelo. De uma forma sucinta, é apresentado o conceito de paralelismo, os tipos de paralelismo, as bibliotecas de comunicação MPI e medidas de desempenho.

10.1.2. Paralelismo

Entende-se por paralelismo como sendo uma técnica de dividir tarefas grandes em tarefas menores (figura 28), as quais serão distribuídas e executadas simultaneamente em vários processadores. Esses processadores se comunicam para que haja coordenação (sincronização) na execução das diversas tarefas executadas em paralelo. A paralelização é feita para aumentar o desempenho no processamento, resolver grandes desafios computacionais e diminuir o tempo gasto no processamento.

Figura 28 – Divisão de Tarefas

Dentre as várias formas de classificar o paralelismo, pode-se levar em consideração o objeto paralelizado classificando-o como paralelismo funcional e paralelismo de dados.

Paralelismo de Dados

No paralelismo de dados o processador executa as mesmas instruções sobre dados diferentes e é aplicado, por exemplo, em programas que utilizam matrizes imensas e para cálculos de elementos finitos. Neste tipo de paralelismo, enquadram-se os sistemas SMP (*Symmetric Multi Processing*) (Figura 29), que possuem mais de um processador em um mesmo computador. Todos eles operam independentemente, mas compartilham os recursos de memória e disco, segundo uma política de controle de concorrência adotada pelo sistema operacional. Esta arquitetura é bem transparente ao usuário, ficando a cargo do sistema operacional a maior parte da complexidade. O acesso pelos processadores à memória é feito diretamente sem a necessidade de passagem de mensagem (*Message Passing*). E somente um processador acessa um endereço da memória por vez. Como exemplo de aplicação deste tipo de paralelismo pode-se citar o Cray Y-MP.

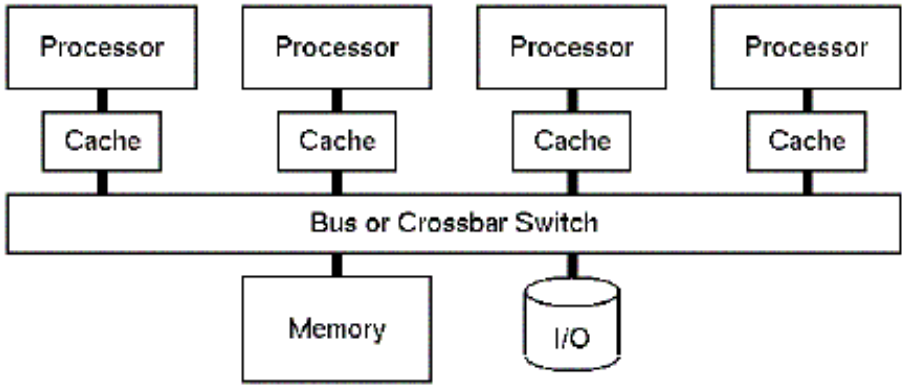


Figura 29 – Arquitetura SMP

Paralelismo Funcional

Já no paralelismo funcional o processador executa diferentes instruções que podem ou não operar sobre o mesmo conjunto de dados. É aplicado em programas dinâmicos e modulares onde cada tarefa será um processo diferente como por exemplo programas em MPI. Neste tipo de paralelismo, enquadram-se os sistemas MPP (*Massive Parallel Processing*) (Figura 30), onde há pouco ou nenhum compartilhamento de recursos entre processadores. Normalmente, cada nó de um sistema MPP é um computador independente, com memória e discos próprios. Nestes sistemas, o controle do paralelismo é realizado pelo programador, que deve coordenar as tarefas e a

coerência entre os diversos processos. Os processadores estão conectados em rede e o acesso as maquinas é feito por passagem de mensagem (*Message Passing*) [43].

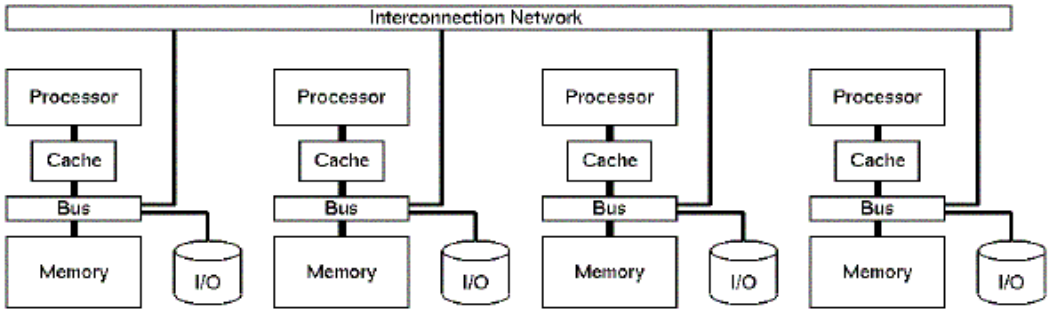


Figura 30 – Arquitetura MPP

O *Message Passing* é um método de comunicação baseada no envio e recebimento de mensagens através de uma rede de computadores seguindo regras de protocolo de comunicação entre vários processadores que possuam memória própria. As informações são enviadas da memória local do processo para memória local do processo remoto. Como exemplos de *Message Passing* pode-se citar:

- PVM - Parallel Virtual Machine;
- MPI - Message Passing Interface;

Um exemplo deste tipo de paralelismo pode ser encontrado em uma máquina IBM Risc/6000 SP2 ou em um cluster de PC's. A grande vantagem de uma máquina do tipo IBM Risc/6000 SP2 e cluster de PC's é o acesso à memória local onde não há contenção e, teoricamente, não existe um limite para o número de processadores. No entanto, existe uma dificuldade maior para mapear as informações, e o usuário é responsável pelo sincronismo e recebimento de dados.

10.1.3. Biblioteca de comunicação MPI

O MPI é um dos modelos de *Message Passing* mais empregado atualmente nas diversas áreas da computação em paralelo para ambiente de memória distribuída. Foi introduzido pelo MPI Fórum em maio de 1994 e atualizado em junho de 1995 [44]. É um produto resultante de um Fórum aberto constituído por 40 organizações de pesquisadores, empresas, usuários e vendedores que definiram a sintaxe, semântica e o conjunto de rotinas padronizadas para *Message Passing*. A documentação oficial se

chama "*MPI: A Message Passing Standard*", publicada pela *University of Tennesee*. MPI 2 fornece extensões para MPI e foi finalizado em julho de 1997.

Como característica, pode-se citar a eficiência, pois foi projetado para executar eficientemente em máquinas diferentes. É especificado somente o funcionamento lógico das operações. A implementação fica a cargo do próprio desenvolvedor que usa as características de cada máquina para gerar um código mais otimizado.

O MPI ou *Message Passing Interface* consiste em um conjunto de bibliotecas ou funções que auxiliam na comunicação entre processos. A Figura 31 apresenta 6 funções básicas do MPI, as quais são responsáveis pela inicialização e terminação do programa, identificação do rank do processo, e envio e recebimento da mensagem.

Figura 31 – Rotinas Básicas de MPI

O programa de MPI é um programa único e as tarefas são divididas em forma de desvios, onde cada tarefa executa uma cópia do programa. Este tipo de programa é classificado como SPMD (*Single Program Multiple Data*). O SPMD é uma taxonomia orientada a programação e é essencialmente igual a taxonomia de Flynn MIMD (*Multiple Instruction Multiple Data*) utilizada para classificação de máquinas.

O primeiro comando da Figura 4.4 (MPI_INIT) inicializa o MPI. Em seguida MPI_COMM_SIZE contabiliza o número de tarefas ou processos (*np*), definido na linha de comando de execução. Esses processos são associados a um comunicador e são capazes de comunicar apenas entre os processos pertencentes ao seu comunicador. Inicialmente, todos os processos são membros de um grupo com um comunicador já pré-estabelecido denominado MPI_COMM_WORLD. Os processos têm uma única identificação denominada de rank (0, 1, .., *np*-1, sendo np o número de processos),

atribuída pelo sistema quando o processo é inicializado e identificado pela chamada do comando MPI_COMM_RANK. Após estas etapas pode-se realizar a troca de mensagens entre os processos associados ao comunicador. Toda troca de mensagem em MPI possui o formato:

função (endereço, contador, tipo de dado, destino ou origem, etiqueta, comunicador, erro)

onde:

- *função*: pode corresponder a um comando de envio ou recebimento de mensagem;
- *endereço*: localização da memória (buffer) onde está armazenada a mensagem a ser enviada ou recebida;
- *contador*: especifica o tamanho da mensagem a ser enviada ou recebida;
- *tipo de dado*: : especifica o tipo de dado a ser enviado ou recebido, normalmente, devem ser iguais nas chamadas de envio e recebimento, exceto, como por exemplo, quando o dado é definido do tipo MPI_PACKED.
- *destino ou origem*: identificação do rank do processo receptor ou emissor;
- *etiqueta*: identificação da mensagem;
- *comunicador*: define um contexto e grupo de comunicação;
- *erro*: código de erro, retorna 0 em caso de sucesso ou código de erro em caso de falha na comunicação.

10.1.4. Medidas de desempenho

Desempenho em Aplicações Paralelas

Os parâmetros mais empregados para avaliar o desempenho em implementações paralelas são:

- Mflop/s: quantidades de operações flutuantes por segundo;
- "Speed-up": medida que avalia o ganho de desempenho do algoritmo;
- Eficiência: a medida de desempenho que exprime o comportamento do desempenho da aplicação com a adição de processadores, variando na faixa de $0 < E_p \leq 1$.

A medida de Mflop/s é calculada a partir da quantidade de instruções de ponto flutuante executadas pelo programa durante toda execução. O "speed-up" é baseado no tempo total de execução do melhor algoritmo serial e no tempo de execução paralela. A formulação para o cálculo do "speed-up" é dada por:

$$S_p = \frac{T_s}{T_p} \tag{152}$$

onde *Ts* é o tempo de execução do programa serial, *Tp* é o tempo de execução do programa paralelo e *p* é o número de processadores utilizados na execução.

A medida de "speed-up" pode variar no intervalo $0 < S_p \leq p$. Neste contexto, o "speed-up" ideal ocorreria quando fosse igual à p, o que é pouco provável de acontecer, pois ao ocorrer a comunicação entre os processadores sempre é adicionada uma carga extra ao tempo de processamento.

Por fim, a eficiência a qual pode variar no intervalo $0 < E_p \leq 1$ é dada pela fórmula:

$$E_p = \frac{S_p}{p} \tag{153}$$

A escalabilidade de um programa é avaliada através do cálculo de sua eficiência. O programa apresentará boa escalabilidade se a sua eficiência for mantida a medida que se aumenta o número de processadores e o tamanho do problema cresce.

Lei de Amdahl

Em alguns casos apenas parte do programa é paralelizável, isto é, algumas tarefas são eminentemente seqüenciais e não tiram proveito de um computador paralelo. Abordando esse tema, Amdahl propôs uma expressão para esse problema, que ficou conhecida como Lei de Amdahl e que está representada a seguir.

Sendo Ts o tempo de execução do programa serial, pode-se supor que o tempo de execução da fração passível de paralelização do programa seja $r.T_s$ e o tempo da fração serial seja $(1 - r).T_s$ [45]. Desta forma, o tempo de execução deste programa com p processadores será dada por:

$$T_p = \frac{r\,T_s}{p} + (1 - r)\,T_s \tag{154}$$

Resultando em:

$$S_p = \frac{1}{(1 - r) + \frac{r}{p}} \tag{155}$$

A eficiência também pode ser obtida pela lei de Amdahl pela fórmula:

$$E_p = \frac{1}{p\,(1 - r) + r} \tag{156}$$

Overhead

Já que a eficiência é a medida da utilização dos processadores durante a execução paralela de um programa, deve-se considerar também toda a carga extra adicionada ao programa devido à sua paralelização [45]. Seja a carga de trabalho de um programa serial igual ao seu tempo de execução, isto é, $W_s = T_s$, e a carga de trabalho de um programa paralelo igual à soma dos tempos de execução do programa em cada um dos processadores (T_{proc}), $W_p = T_p = p.T_{proc}$. Assim, pode-se reescrever a formulação de eficiência como:

$$E_p = \frac{T_s}{p\,T_{proc}} = \frac{W_s}{W_p} \tag{157}$$

Na prática o overhead é a carga extra de trabalho introduzida devido à paralelização do programa. Ou seja, é a diferença entre a quantidade de trabalho realizada por um programa serial e a quantidade de trabalho realizada por um programa paralelo, a qual é dada por:

$$T_0 = p\,T_{proc} - T_s \tag{158}$$

O overhead se origina de três fontes:

- Comunicação entre os processadores;
- Tempo ocioso em alguns processadores devido a uma possível diferença de desempenho em alguns dos processadores ou má distribuição de carga computacional entre processadores;
- Cálculos extras necessários na implementação paralela ou cálculos repetidos em vários processadores.

10.2. ALGORITMO PDII EM COMPUTADORES COM ARQUITETURA PARALELA

10.2.1. Introdução

Neste item será apresentada a estrutura geral do código seqüencial do programa Prosim. Em seguida, serão descritas as estratégias empregadas para a implementação em paralelo do algoritmo PDII.

10.2.2. Análise do código sequencial

Para a implementação do código seqüencial adotado neste trabalho realizou-se primeiramente uma decomposição do algoritmo em módulos. Esta decomposição levou em consideração o aspecto funcional da estrutura do código. De uma forma geral, o código seqüencial do Prosim se divide basicamente em seis módulos funcionais (Figura 32):

- leitura de dados de entrada;
- equilíbrio estático individual das linhas (de ancoragem e risers);
- análise estática acoplada (casco, linhas de ancoragem e risers) [46];
- análise dinâmica acoplada (casco, linhas de ancoragem e risers);
- geração de arquivos de saída;
- geração de arquivo para posterior reinício da análise.

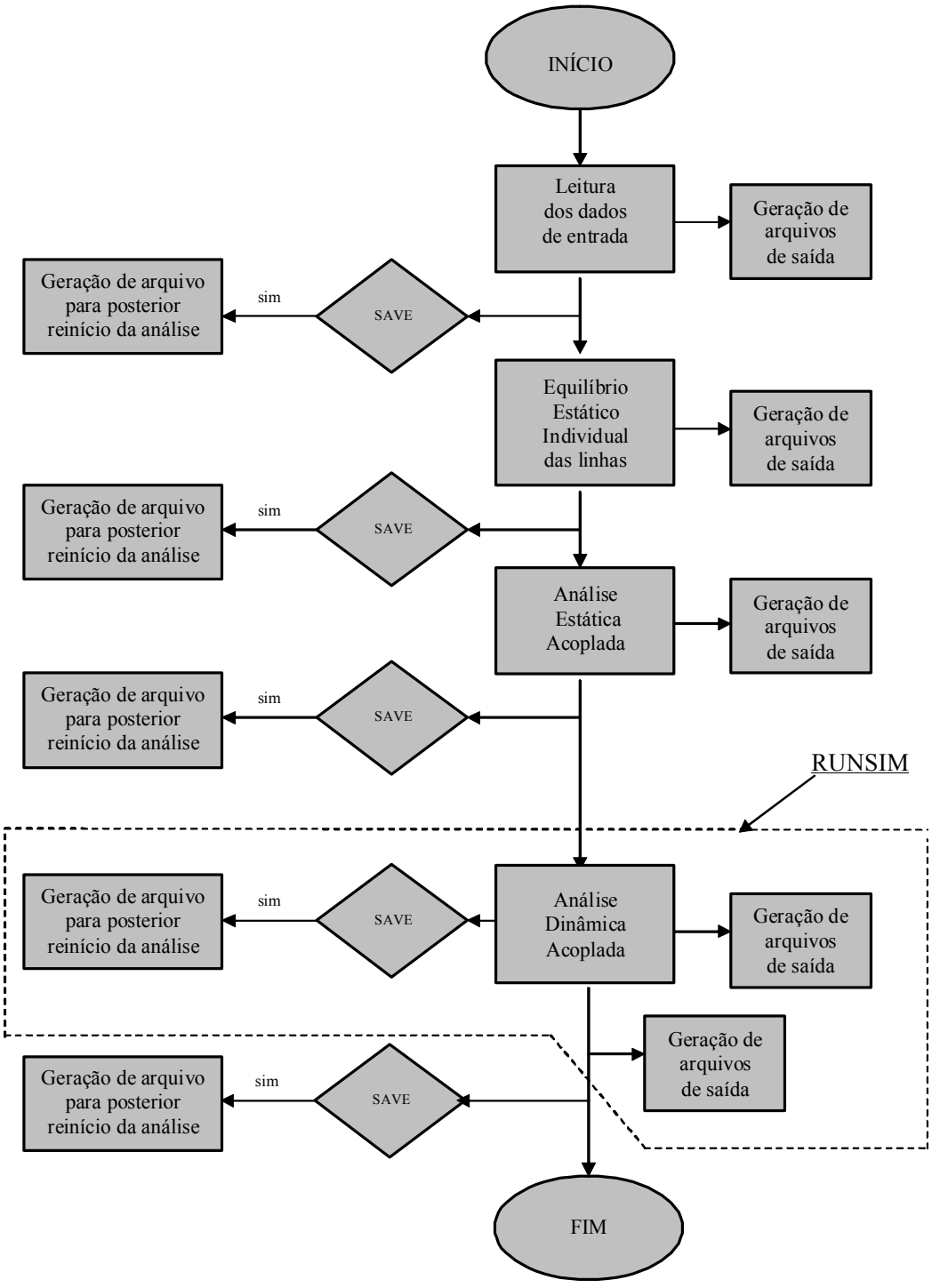


Figura 32 – Fluxograma Geral do Programa Prosim

10.2.3. Estratégia para Implementação em Paralelo

O módulo, "análise dinâmica acoplada", se destaca pelo consumo de CPU. Com base nos tempos de processamento obtidos do código seqüencial, observou-se que a maior parte do tempo de execução do programa seqüencial se concentra neste módulo. Para os tempos de simulação usualmente empregados em uma análise dinâmica, este módulo consome praticamente a totalidade do tempo de processamento. Desta forma, pode-se dizer que para obter uma paralelização eficiente deve-se concentrar os esforços no módulo de análise dinâmica.

O módulo de "análise dinâmica acoplada" se subdivide basicamente em dois procedimentos que ocorrem simultaneamente:

- Solução das equações de movimento do casco, resultando nos movimentos da unidade flutuante, sob ação das forças ambientais e da tração de topo das linhas de ancoragem e risers;
- Solução das equações de movimento de cada linha de ancoragem e risers, resultando nas da forças de topo das mesmas, sob ação dos movimentos do casco e das cargas ambientais.

Franco em [47] implementou a partição casco / linha, onde um processo gerencia e integra as equações de movimento do casco, enquanto que, as linhas, discretizadas em elementos finitos, eram distribuídas e integradas uma a uma pelos demais processos. Neste caso as linhas são integradas sem serem particionadas.

Com a implementação do algoritmo PDII as linhas passam também a ser particionadas. Com esta implementação torna-se possível um melhor balanceamento de carga entre processadores. Por exemplo: em um sistema composto por linhas e casco, caso tenhamos uma linha muito discretizada em relação às outras, isto é com muito mais graus de liberdade a serem integrados, esta linha terá um tempo de CPU muito mais elevado que as outras. Supondo que, cada linha esteja sendo analisada em um processador diferente, esta linha muito discretizada será o "gargalo" da análise. Com o algoritmo PDII, esta linha também poderá ser particionada e distribuída entre os processadores com a possibilidade de melhora no desempenho.

A implementação do PDII conduz ao trecho de código a ser paralelizado, conforme mencionado no capítulo anterior. A resolução do sistema de equações decorrente da integração das equações de movimento de cada linha de ancoragem e risers, proveniente da utilização de algoritmos implícitos, e o loop de elementos onde são reavaliados matriz de rigidez e esforços internos, proveniente dos efeitos dinâmicos não-lineares, são os trechos do programa onde consome-se cerca de 80% do tempo computacional. Os demais procedimentos paralelizados da análise dinâmica levam a um total de cerca de 90% do tempo computacional paralelizado.

Supondo então que 90% do tempo computacional esteja paralelizado a figura 33 apresenta o "speed-up" ideal e o máximo "speed-up" obtido pela Lei de Amdahl, que leva em conta a fração paralelizável do programa, variando-se o número de processadores.

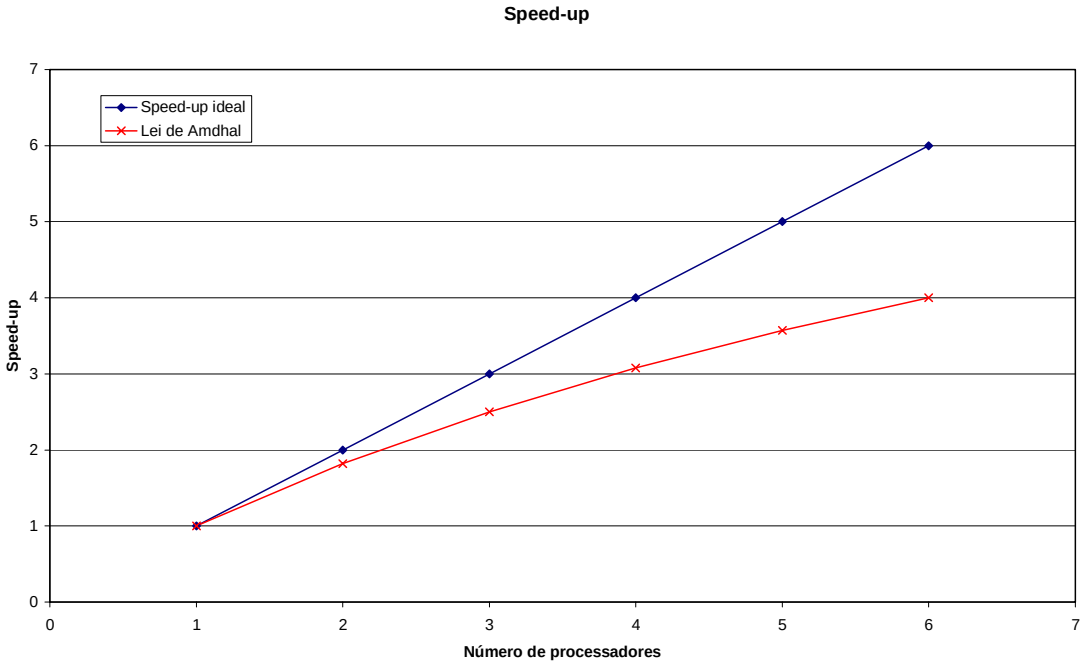


Figura 33 – Medidas de Speed-up

Observa-se que à medida que o número de processadores aumenta, há uma queda no "speed-up" seguindo-se a Lei de Amdahl, se comparada ao "speed-up" ideal. A seguir são apresentados os procedimentos de implementação em paralelo da resolução do sistema de equações e do ciclo para reavaliação dos elementos.

Resolução do sistema de Equações

A partição do domínio em subdomínios, resulta em sub sistemas de equações, provenientes do sistema de equações que seria montado caso a estrutura não fosse partida, a serem resolvidos pelo Método Direto de Gauss através da triangularização e retrosubstituição em cada processador. Nesta fase a distribuição de tarefas entre os processadores ocorre na resolução do sistema de equações.

Este procedimento é exemplificado a seguir, onde é apresentado o trecho da resolução do sistema de equações e a compatibilidade dos nós de interface.

1 – Triangularização da matriz efetiva em cada processador;

2 – Resolução do sistema de equações através da retro-substituição para obtenção dos deslocamentos;

3 – Envio dos deslocamentos das interfaces para o processador mestre;

4 – Compatibilidade dos deslocamentos nas interfaces através da Regra de Ponderação da Massa;

5 – Envio dos deslocamentos das interfaces compatibilizados para os demais processadores.

Ciclo para Reavaliação de Elementos

A seguir é apresentado um breve procedimento com a partição do ciclo de elementos.

1 – "Loop" no grupo de elementos pertencentes a cada subdomínio;

2 – Reavaliação da matriz efetiva do subdomínio avaliado;

3 – Cálculo dos esforços internos do subdomínio avaliado;

4 – Fim do "loop".

Troca de Mensagens

Como mencionado anteriormente, a troca de mensagens entre os processos foi dividida em duas fases:

As fase I corresponde à etapa inicial (de leitura de dados). A fase II corresponde à paralelização efetuada durante a análise dinâmica das linhas.

FASE I

A primeira fase trata-se de uma comunicação onde o processador mestre envia, aos demais processadores, informações necessárias para abertura dos arquivos de entrada. Efetivamente, a leitura dos arquivos de entrada de dados é feita por todos os processos aproveitando a fração de memória de disco compartilhada. Em cluster de PC's todos os processadores compartilham uma fração de memória do disco do PC administrador e desta forma todos processadores conseguem ler os arquivos contidos nesta fração de memória. Para distribuir os parâmetros foi empregada a operação de difusão realizada pelo comando MPI_BCAST.

FASE II

Na segunda fase de comunicação entre os processadores há uma troca de dados envolvendo todos os processos a cada iteração na solução das linhas. Para comunicação dos dados envolvidos nessa fase empregou-se a comunicação ponto a ponto adotando as funções MPI_SEND e MPI_RECV.

Balanceamento de Carga Computacional

O balanceamento de carga deve ser adotado a partir do momento em que se verifica a existência de um overhead muito grande, proveniente do tempo espera de alguns processadores. Este tempo de espera é conseqüência do desequilíbrio da carga computacional. Alguns processadores são sobrecarregados recebendo tarefas maiores do que os demais, devido por exemplo ao diferente número de elementos das malhas de cada linha e, conseqüentemente, tendem a consumir um tempo de processamento maior. Com o objetivo de diminuir a carga extra gerada pelo desequilíbrio da carga computacional deve-se buscar uma forma de distribuição simples e ao mesmo tempo eficaz capaz de realizar o balanceamento desta carga computacional.

Uma forma de distribuição eficaz é a busca do equilíbrio entre o número total de equações da malha de elementos finitos atribuído a cada processador. Neste procedimento tenta-se aproximar o número de equações de um processador da média (número total de equações / número de processadores).

Este procedimento pode ser feito já na montagem do arquivo de entrada do programa, onde a divisão em subdomínios pode ser efetuada já considerando-se este balanceamento, ou pode ser feito internamente ao programa de maneira automática.

Na implementação atual do algoritmo PDII, ao particionar uma linha, o balanceamento de carga deve ser efetuado anteriormente à análise e fornecido no arquivo de entrada de dados do Prosim de forma, preferencialmente, a uma distribuição uniforme dos graus de liberdade pelos processadores envolvidos na análise.

11. APLICAÇÕES NUMÉRICAS

11.1. INTRODUÇÃO

Neste capítulo apresentam-se resultados e estudos de performance obtida com as estratégias de subciclagem e implementação do PDII empregadas sobre o código do programa Prosim, na análise de exemplos clássicos, análise de riser e análise acoplada de unidades flutuantes com linhas de ancoragem e risers. Foram realizadas as seguintes aplicações:

- O primeiro exemplo consiste em um riser em catenária submetido a um movimento senoidal imposto ao topo do mesmo;

- No segundo exemplo este mesmo riser é submetido a um carregamento de correnteza e movimento prescrito no topo;

- O terceiro exemplo apresenta um modelo acoplado da plataforma P10 constituída de 8 linhas de ancoragem e um riser vertical de perfuração. Neste exemplo são aplicados carregamentos de onda e correnteza ao sistema;

- O quarto exemplo apresenta um modelo acoplado da plataforma P18 constituída de 8 linhas de ancoragem e 73 risers, sendo 72 risers flexíveis e um SCR (Steel Catenary Riser), todos com configurações em catenária livre. Neste exemplo são aplicados carregamentos de onda, correnteza e vento ao sistema.

Nos exemplos aqui apresentados o tempo total de simulação foi de apenas 50 segundos. Usualmente, em análises de sistemas offshore empregam-se tempos muito maiores (até 10800 segundos). Como aqui o objetivo é apenas comparar o desempenho computacional, o tempo total adotado é considerado suficiente.

Para aplicação numérica dos exemplos com subciclagem utilizou-se um computador com as seguintes características:

- Processador Athlon XP 2600 com 2.08 Ghz;
- 1.0 Gbyte de memória RAM;
- Sistema operacional: Windows XP.

Para aplicação numérica dos exemplos com método PDII utilizou-se o cluster instalado no LAMCSO (Laboratório de Métodos Computacionais e Sistemas Offshore), o qual apresenta as características apresentadas na tabela 5.

Tabela 5 – Especificação do Cluster

Processadores	6 processadores Pentium III 850 MHz
Memória RAM	256 Mbytes por unidade
Dispositivo de Comunicação	Rede Fast Ethernet 100 Mbps
Sistema Operacional	Linux Red Hat
Software de Comunicação	LAM/MPI versão 6.5.5

11.2. RISER EM CATENÁRIA SOB MOVIMENTO IMPOSTO NO TOPO

11.2.1. Descrição do Modelo

Este exemplo apresenta uma configuração típica de riser flexível de 10" utilizada na exploração de petróleo. É uma configuração em catenária simples com ângulo de topo de 7 graus na posição neutra situado numa lâmina d'água de 1000m. A seguir a figura 34 com a configuração.

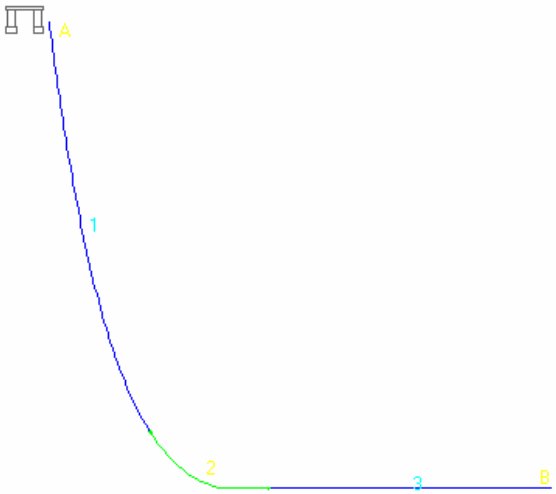


Figura 34 – Configuração do riser

Na análise do riser foram utilizados elementos de pórtico discretizados em 3 segmentos conforme apresentado na tabela 6.

Tabela 6 – Discretização do Riser

Segmento	Comprimento do segmento (m)	Comprimento dos elementos (m)	Número de elementos
1	905	15	60
2	300	5	60
3	600	10	60

A região do TDP, por ser uma região com maiores variações de curvatura, requer um maior refinamento na malha.

11.2.2. Dados Geométricos e do Material

A Tabela 7 reproduz os dados geométricos e do material.

Tabela 7 – Propriedades

Diâm. ext (m)	0.321
Diâm. int (m)	0.252
EA (kN)	83333
EI (kN.m^2)	0.7
Peso ar (kN/m)	1.521
Peso água (kN/m)	0.707

11.2.3. Carregamento

O carregamento utilizado neste exemplo consiste em um movimento senoidal prescrito no topo do riser e apresentado nas figuras a seguir para as direções horizontal (figura 35) e vertical (figura 36).

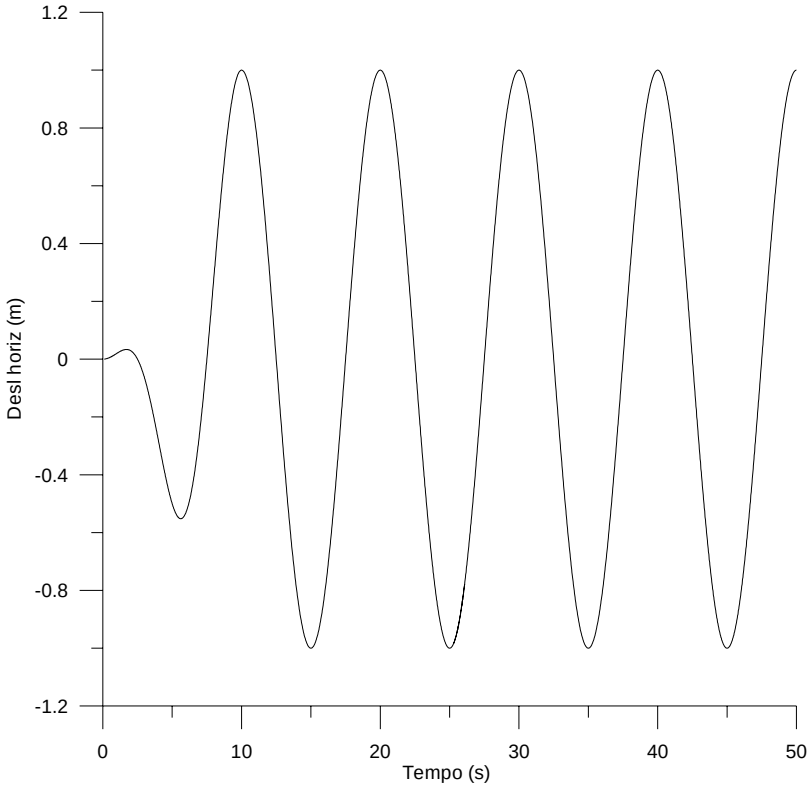


Figura 35 – Deslocamento horizontal

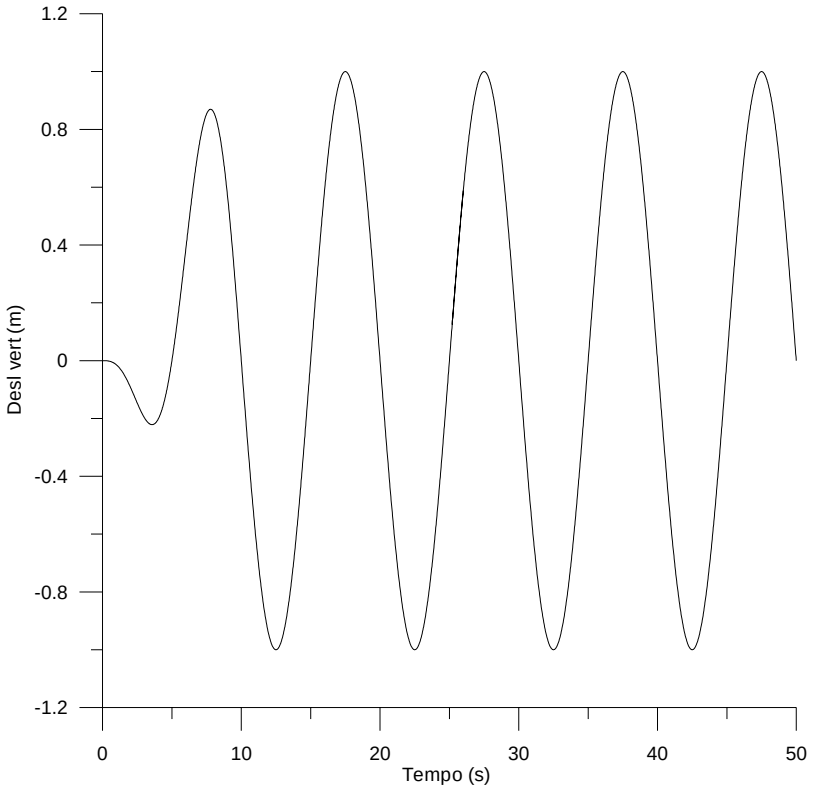


Figura 36 – Deslocamento vertical

O objetivo deste exemplo é observar o comportamento do riser sob movimento prescrito.

11.2.4. Subciclagem - Resultados

Para utilização da subciclagem, conforme o exemplo anterior, utilizando-se da divisão de segmentos do riser devido à discretização utilizada, associou-se intervalos de tempo a cada segmento.

Desta forma utilizou-se dois intervalos de tempo correspondentes a cada segmento:

- Segmentos 1 (topo) e 3 (âncora): $\Delta t = 0.001$ seg;

- Segmento 2 (TDP): $\Delta t = 0.0005$ seg.

Na análise tradicional, sem subciclagem, utilizou-se o menor intervalo de tempo para todos os elementos, $\Delta t = 0.0005$ seg.

Não foi utilizado amortecimento numérico nesta análise.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim, sem o uso da subciclagem e a versão implementada com uso da subciclagem. Conforme pode-se observar na figura 37 os resultados são praticamente idênticos.

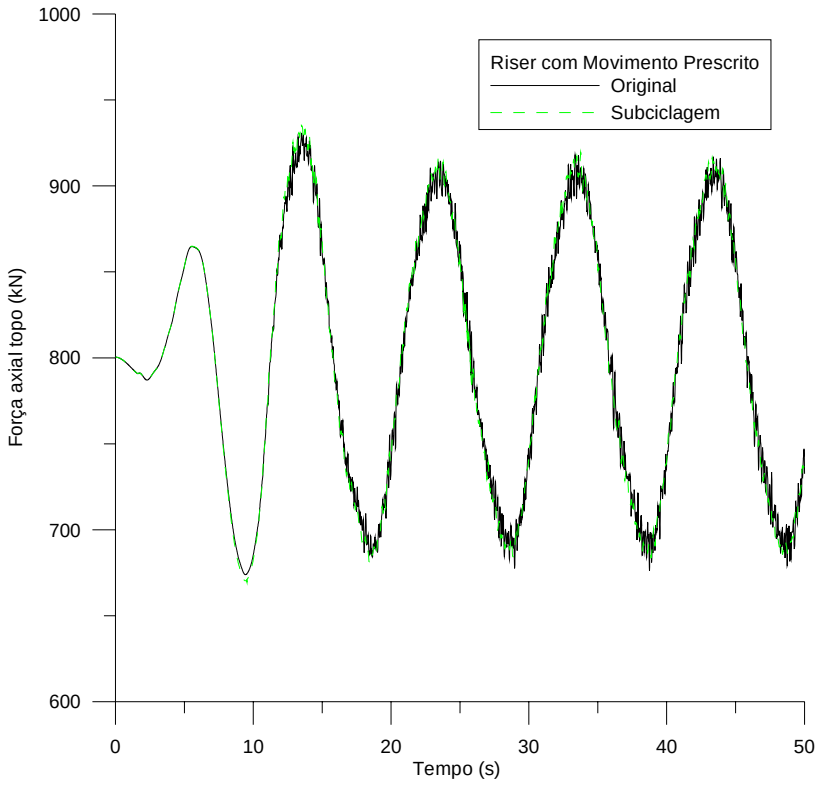


Figura 37 – Resultados comparativos

Visto a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido nesta análise. A tabela 8 apresenta o tempo de CPU gasto.

Tabela 8 – Desempenho Computacional

	Tradicional	**Subciclagem**	**Economia %**
Tempo CPU (seg)	1195.73	908.72	24.0

Com isto pode-se observar o ganho computacional significativo de 24.0% obtido neste caso.

11.2.5. PDII - Resultados

Para utilização do algoritmo PDII, neste exemplo, a partição do domínio foi efetuada utilizando-se da divisão de segmentos do riser devido à discretização utilizada. Portanto o riser foi particionado em 3 subdomínios.

A seguir são apresentados os principais parâmetros utilizados nesta análise:

- Intervalo de tempo: 0.0125 seg;

- Tolerância de deslocamentos: 0.001

- Tolerância de forças: 0.01

- Amortecimento numérico (coeficiente αB): -0.33.

- Amortecimento proporcional à massa: não foi considerado.

- Amortecimento proporcional à rigidez: não foi considerado.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim e a versão implementada com uso do PDII. Conforme pode-se observar na figura 38 os resultados são praticamente idênticos.

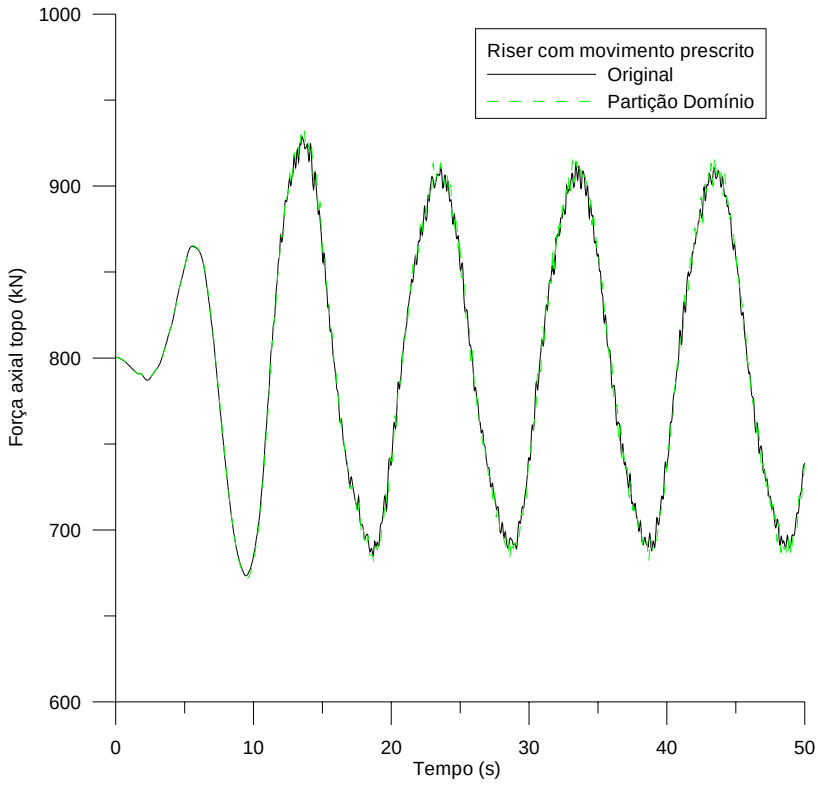


Figura 38 – Resultados comparativos

Visto a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido. A tabela 9 apresenta o tempo de CPU gasto.

Tabela 9 – Desempenho Computacional

	Tradicional	PDII	Economia %
Tempo CPU (seg)	148.1	78.6	46.9

O ganho computacional de 46.9% mostrou-se bastante significativo.

Uma análise mais detalhada dos tempos de processamento, comunicação e espera entre a análise seqüencial e a análise com 3 processadores é apresentado a seguir.

A figura 39 apresenta um gráfico comparativo dos tempos total paralelizado e total da análise. Além de frações do tempo total paralelizado divididos em tempo de espera, comunicação, processamento, tanto para a análise seqüencial como para a análise com o domínio dividido em 3 subdomínios.

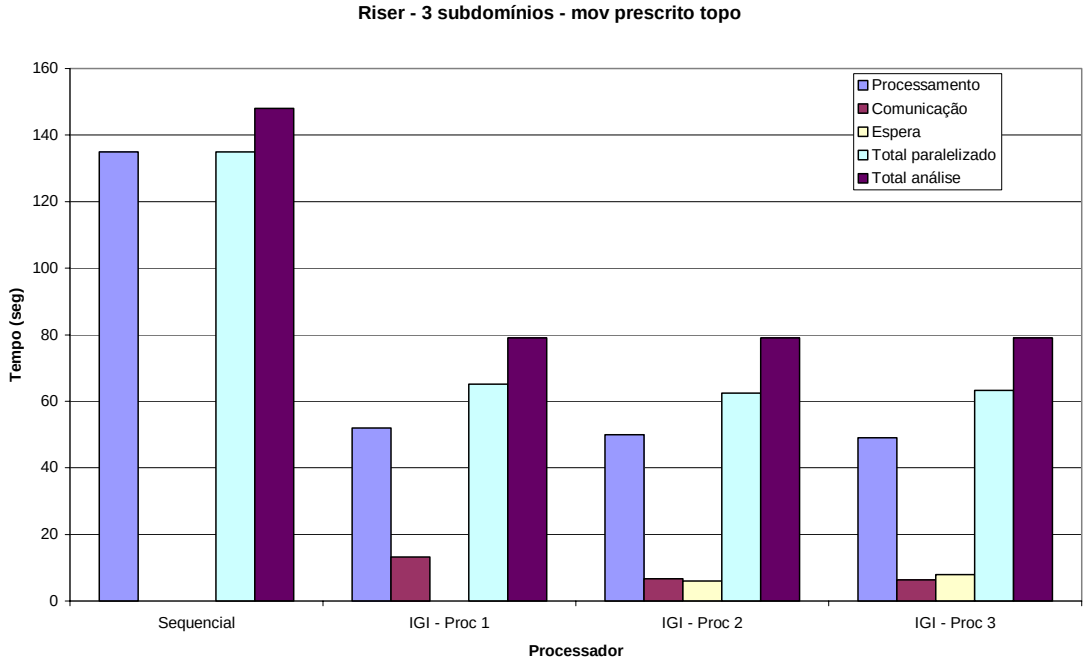


Figura 39 – Medidas de Desempenho

O tempo de processamento em cada processador onde o PDII é utilizado é praticamente 1/3 do tempo de processamento da análise seqüencial. O tempo de comunicação é maior no processador 1 pois é este processador que concentra os valores de deslocamento na interface de todos os processadores e faz a compatibilidade de deslocamentos. O tempo de espera nos processadores 2 e 3 aparecem devido ao envio de informações ao processador 1 pelos outros processadores. O tempo total da análise com PDII apresenta-se muito menor que o tempo total com análise seqüencial.

11.2.6. PDII – Variação do número de processadores

Com o objetivo de um melhor entendimento da partição do domínio, neste item é apresentado um estudo com a variação da partição da malha do riser de 1 a 6 subdomínios. A figura 40 mostra que o sinal de tração no topo do riser é praticamente coincidente para qualquer uma das divisões de domínio utilizadas.

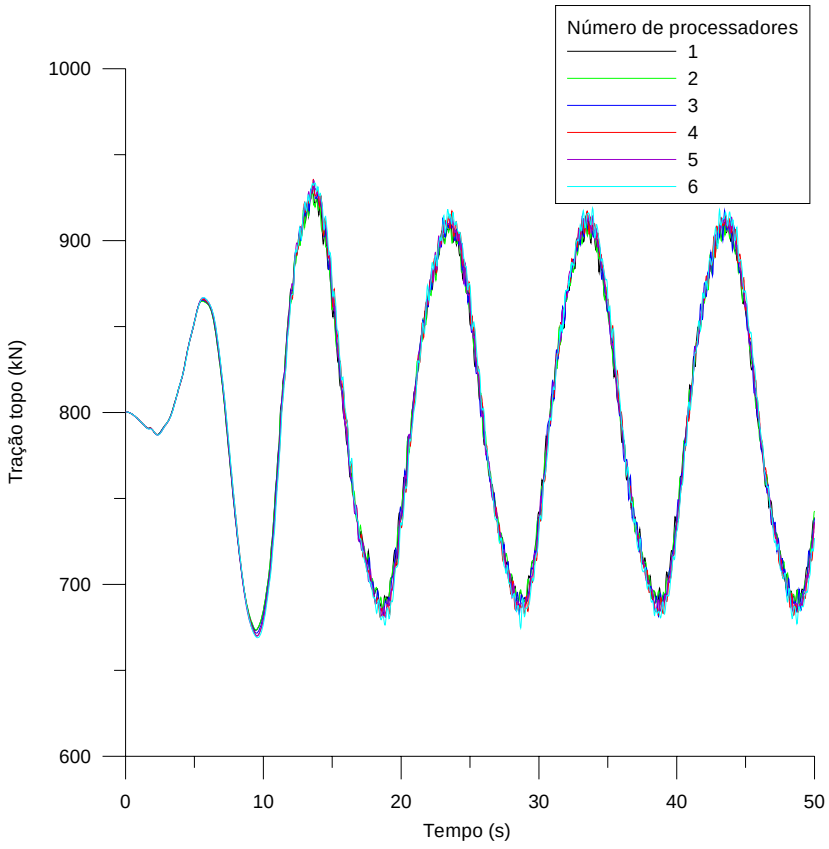


Figura 40 – Comparação de Resultados

Os gráficos das figuras 41, 42, 43, 44 e 45 a seguir mostram a variação dos tempos de processamento, espera, comunicação e total paralelizado destas análises sempre comparados com o tempo de processamento da análise seqüencial em 1 processador.

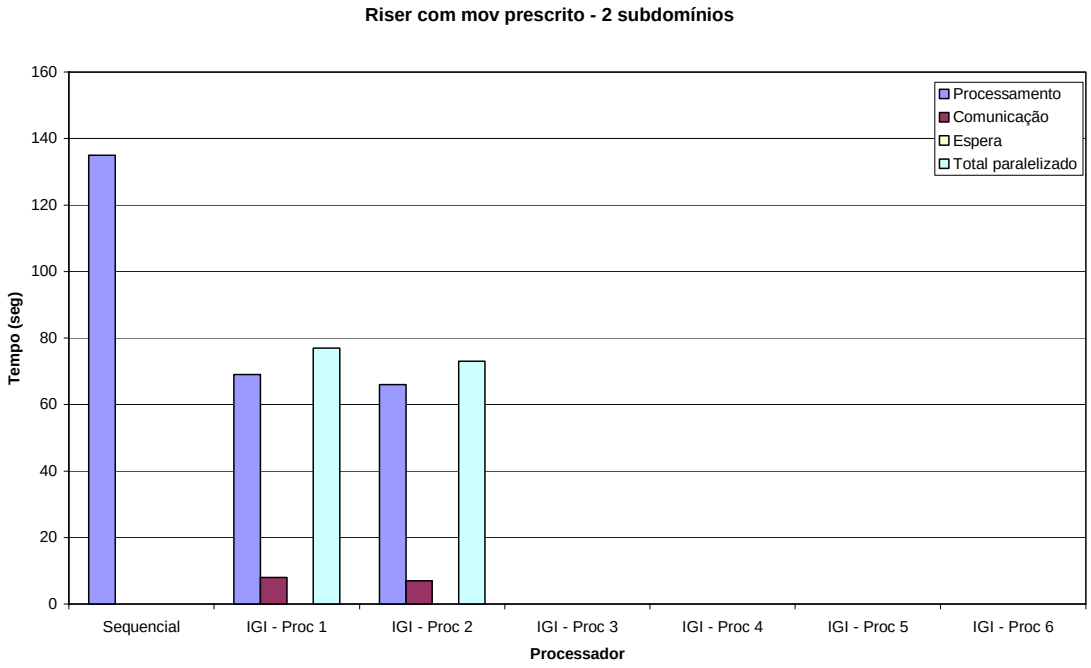


Figura 41 – Medidas de Desempenho – 2 processadores

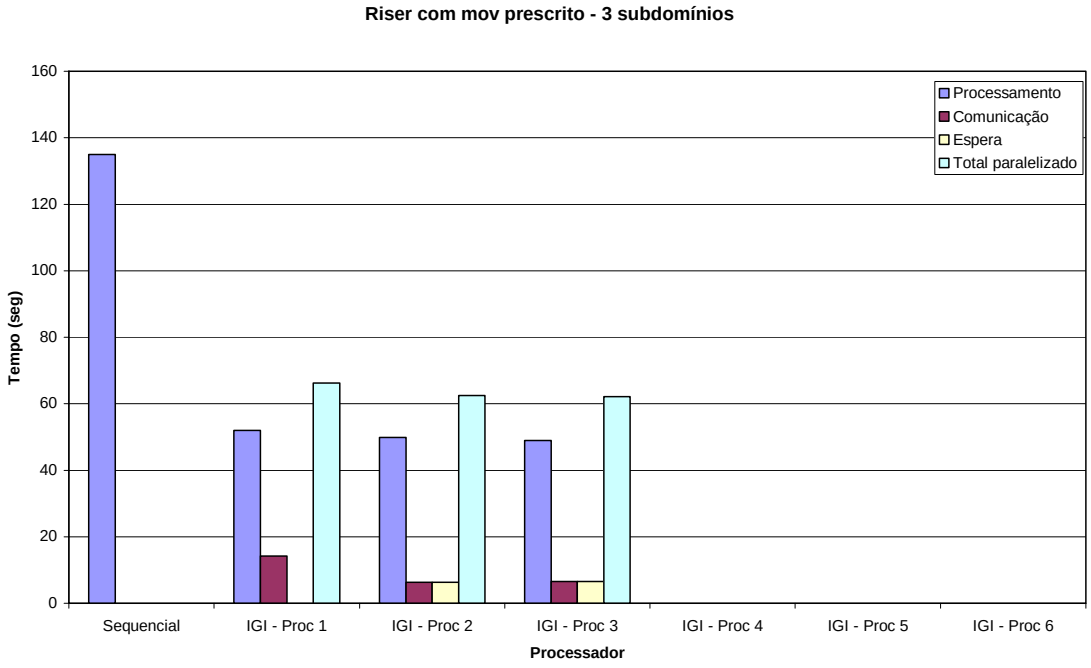


Figura 42 – Medidas de Desempenho – 3 processadores

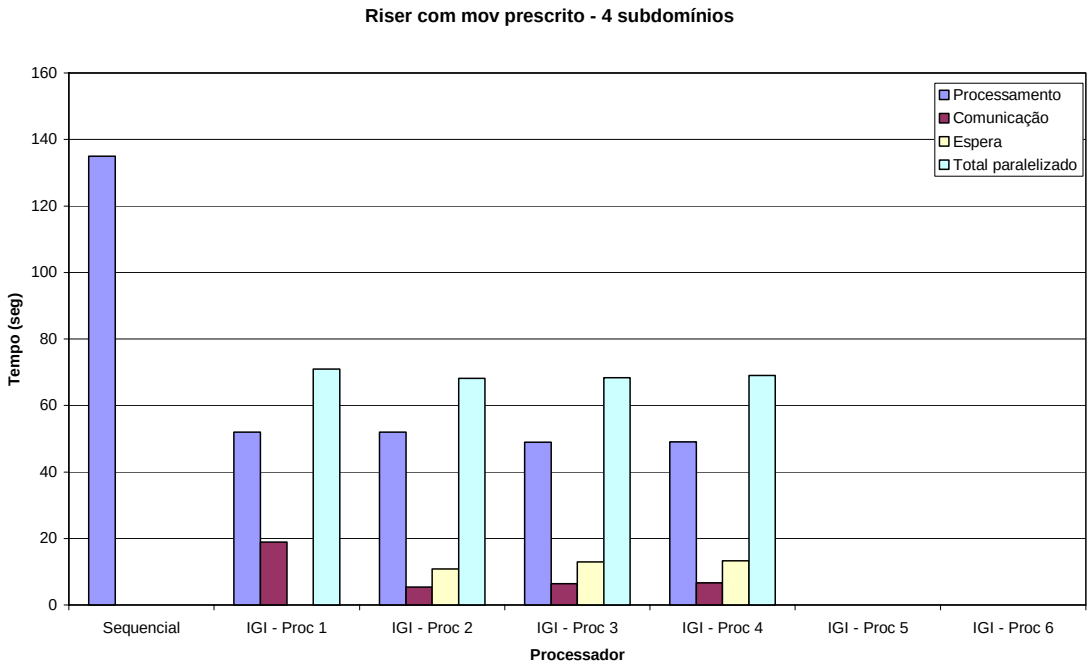


Figura 43 – Medidas de Desempenho – 4 processadores

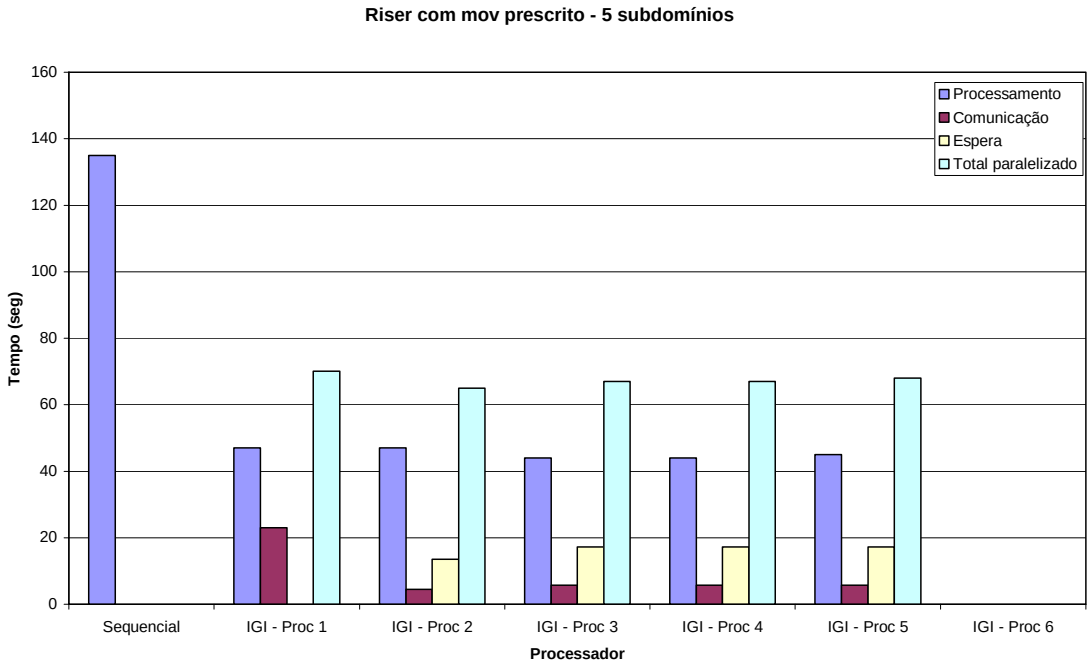


Figura 44 – Medidas de Desempenho – 5 processadores

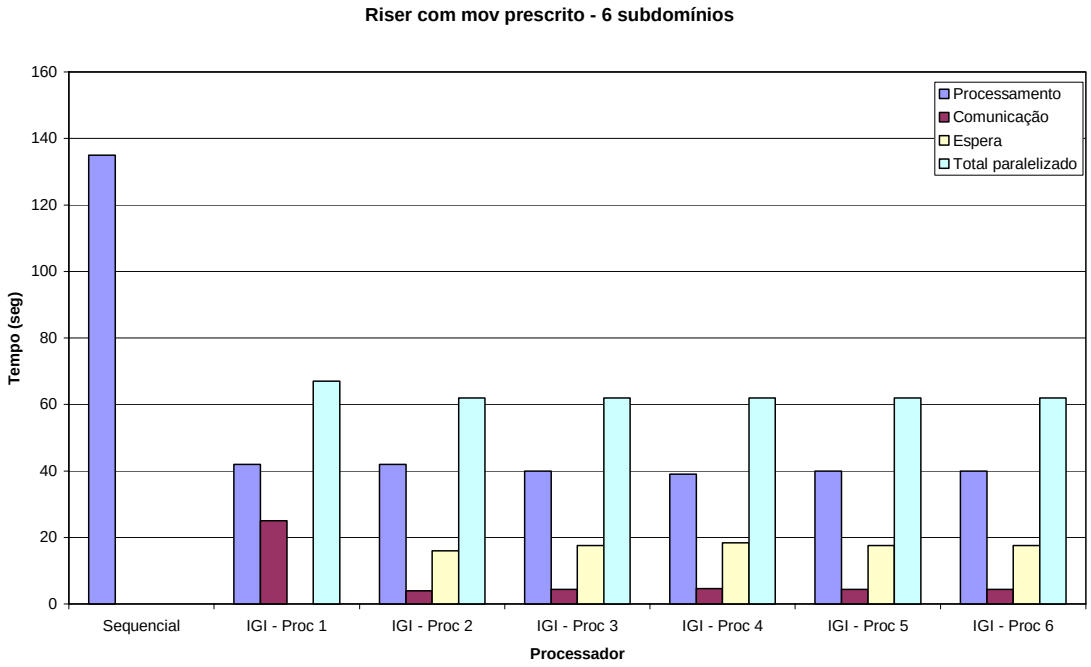


Figura 45 – Medidas de Desempenho – 6 processadores

Com os gráficos apresentados pode-se concluir que à medida que o número de processadores aumenta o tempo de comunicação do processador 1 também aumenta, isto ocorre porque, de acordo com a implementação efetuada os dados necessários para compatibilização da interface são enviados de todos os processadores para o processador 1 o que "congestiona" o tráfego de informações deste processador. Enquanto o processador 1 fica com excesso de tempo de comunicação os demais ficam com excesso de tempo de espera.

O gráfico da figura 46 a seguir mostra os tempos de processamento, comunicação e total paralelizado do processador 1 (mestre).

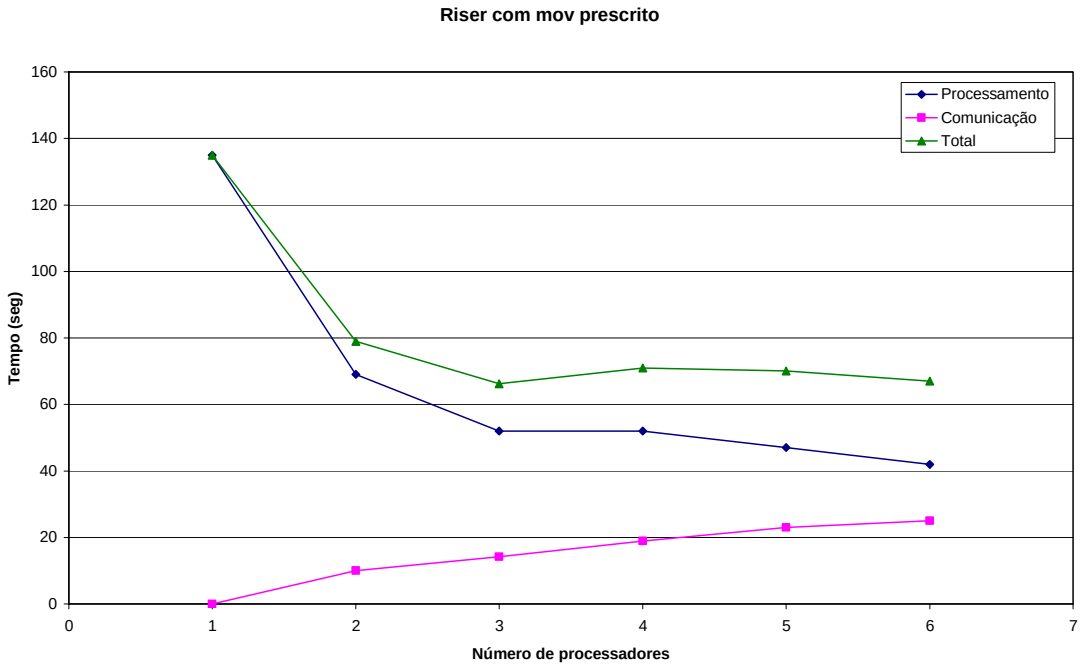


Figura 46 – Desempenho do Processador Mestre

Nota-se que o tempo de processamento diminui à medida que o número de processadores aumenta e o tempo de comunicação aumenta. O tempo total do trecho paralelizado do programa diminui até o uso de 3 processadores, com 4 processadores este tempo total aumenta e volta a diminuir com 5 ou 6 processadores.

A seguir mais um gráfico na figura 47 com medidas de speed-up comparados ao speed-up ideal e ao speed-up obtido pela Lei de Amdhal que considera a fração paralelizável do programa em relação ao tempo total da análise, no caso 92%.

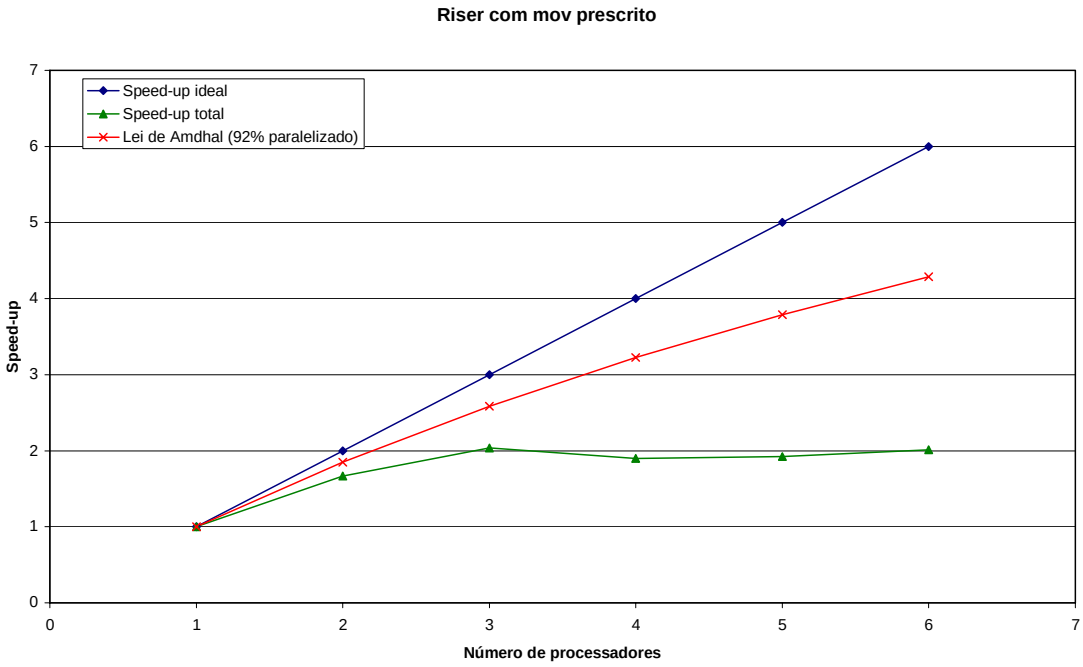


Figura 47 – Medidas de Speed-Up

Observa-se que com até 3 processadores o speed-up apresenta um ganho considerável tendo uma queda a partir do uso de 4 processadores.

A causa desta queda de rendimento do processamento com 4 ou mais processadores se deve ao número de iterações do ciclo de Newton-Raphson. À medida que o riser é mais particionado com um maior número de subdomínios, mostra-se necessário um maior número de iterações para convergência do método. Assim aumentam o tempo de processamento e o tempo de comunicação, visto que a troca de informações entre processadores é necessária a cada iteração.

A figura 48 mostra o número médio de iterações Newton-Raphson por intervalo de tempo durante a análise do riser.

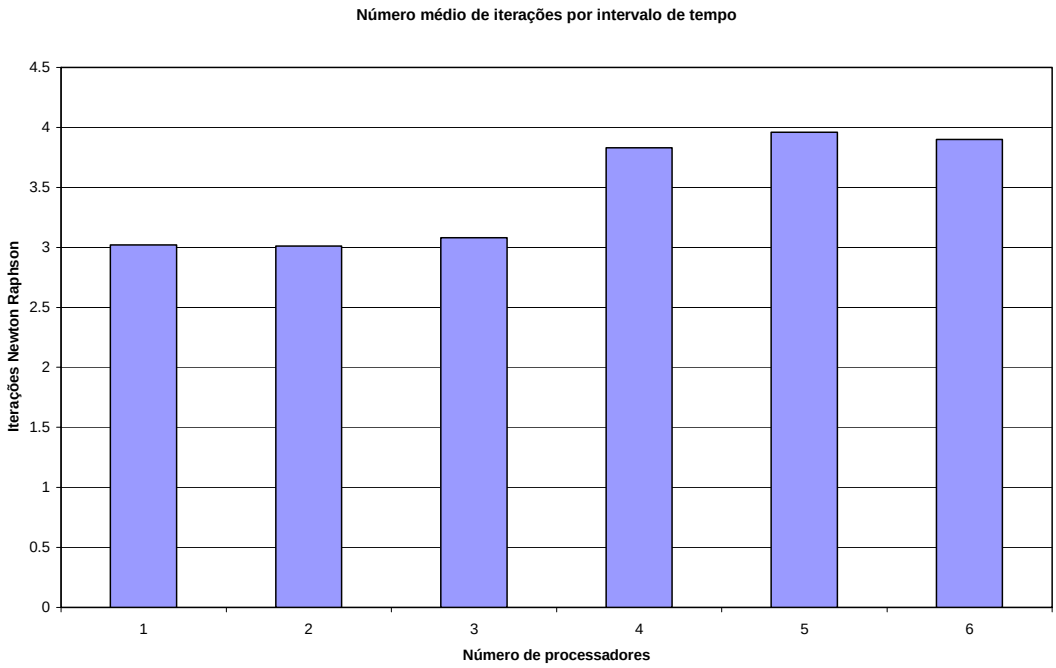


Figura 48 – Número de Iterações

Observa-se que com até 3 processadores são necessárias cerca de 3 iterações para convergência em cada intervalo de tempo enquanto que para 4 ou mais processadores este número aproxima de 4 iterações.

Portanto, nos problemas de análise de riser ou linhas de ancoragem, recomenda-se que esta partição seja feita em, no máximo, 3 subdomínios evitando-se assim um aumento do número de iterações que pode ocasionar uma queda de desempenho do algoritmo.

11.3. RISER EM CATENÁRIA SOB AÇÃO DE CORRENTEZA

11.3.1. Descrição do Modelo

O modelo utilizado nesta análise é igual ao modelo do exemplo anterior. O objetivo deste exemplo é a verificação da precisão da resposta sendo o modelo submetido a um carregamento distribuído de correnteza e movimento prescrito no topo.

11.3.2. Carregamento

Esta análise consiste na aplicação de um perfil de correnteza triangular e o movimento harmônico, idêntico ao do exemplo anterior, aplicado no topo do riser.

A tabela a seguir apresenta valores de correnteza utilizados.

Tabela 10 – Perfil de Correnteza

Profundidade (m)	Velocidade (m/s)
0	1,00
1000	0,00

A figura 49 apresenta a direção de aplicação do carregamento.

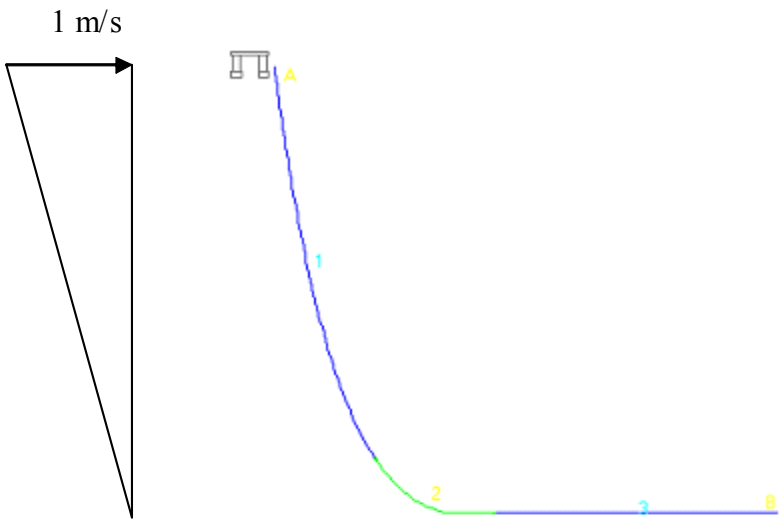


Figura 49 – Atuação do perfil de correnteza

11.3.3. Subciclagem - Resultados

Para utilização da subciclagem, neste exemplo, utilizando-se da divisão de segmentos do riser devido à discretização utilizada, associou-se intervalos de tempo a cada segmento.

Desta forma utilizou-se dois intervalos de tempo correspondentes a cada segmento:

- Segmentos 1 (topo) e 3 (âncora): $\Delta t = 0.001$ seg;

- Segmento 2 (TDP): $\Delta t = 0.0005$ seg.

Na análise tradicional, sem subciclagem, utilizou-se o menor intervalo de tempo para todos os elementos, $\Delta t = 0.0005$ seg.

Não foi utilizado amortecimento numérico nesta análise.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim, sem o uso da subciclagem e a versão implementada com uso da subciclagem. Conforme pode-se observar na figura 50 os resultados são praticamente idênticos.

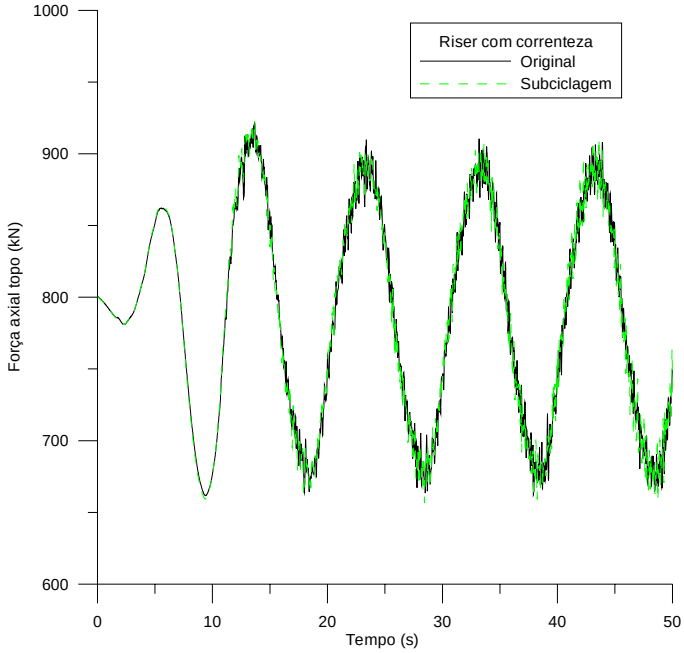


Figura 50 – Resultados comparativos

Visto a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido. A tabela 11 apresenta o tempo de CPU gasto.

Tabela 11 – Desempenho Computacional

	Tradicional	Subciclagem	Economia %
Tempo CPU (seg)	1252.13	949	24.2

Com isto pode-se observar o ganho computacional significativo de 24.2% obtido neste caso.

11.3.4. PDII - Resultados

Para utilização do algoritmo PDII, neste exemplo, a partição do domínio foi efetuada utilizando-se da divisão de segmentos do riser devido à discretização utilizada. Portanto o riser foi particionado em 3 subdomínios.

A seguir são apresentados os principais parâmetros utilizados nesta análise:

- Intervalo de tempo: 0.0125 seg;

- Tolerância de deslocamentos: 0.001

- Tolerância de forças: 0.01

- Amortecimento numérico (coeficiente αB): -0.33.

- Amortecimento proporcional à massa: não foi considerado.

- Amortecimento proporcional à rigidez: não foi considerado.

A seguir são apresentados resultados comparativos entre a versão original do programa Prosim e a versão implementada com uso do PDII. Conforme pode-se observar na figura 51 os resultados são praticamente idênticos.

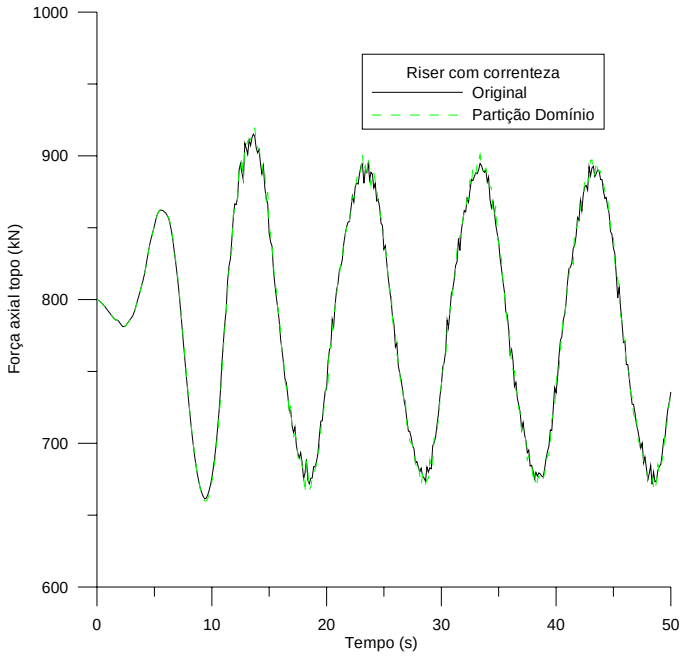


Figura 51 – Resultados comparativos

O principal objetivo deste exemplo é verificar a precisão dos resultados com o riser sob carregamento externo distribuído. O desempenho neste exemplo é semelhante ao exemplo anterior pois é utilizada a mesma malha e mesma partição.

Visto a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido nesta análise. A tabela 12 apresenta o tempo de CPU gasto.

Tabela 12 – Desempenho Computacional

	Tradicional	**PDII**	**Economia %**
Tempo CPU (seg)	152.0	82.1	46.0

Observa-se o ganho computacional significativo de 46% obtido nesta análise.

A figura 52 apresenta um gráfico comparativo dos tempos total paralelizado e total da análise. Além de frações do tempo total paralelizado divididos em tempo de espera, comunicação, processamento, tanto para a análise seqüencial como para a análise com para a análise com o domínio dividido em 3 subdomínios.

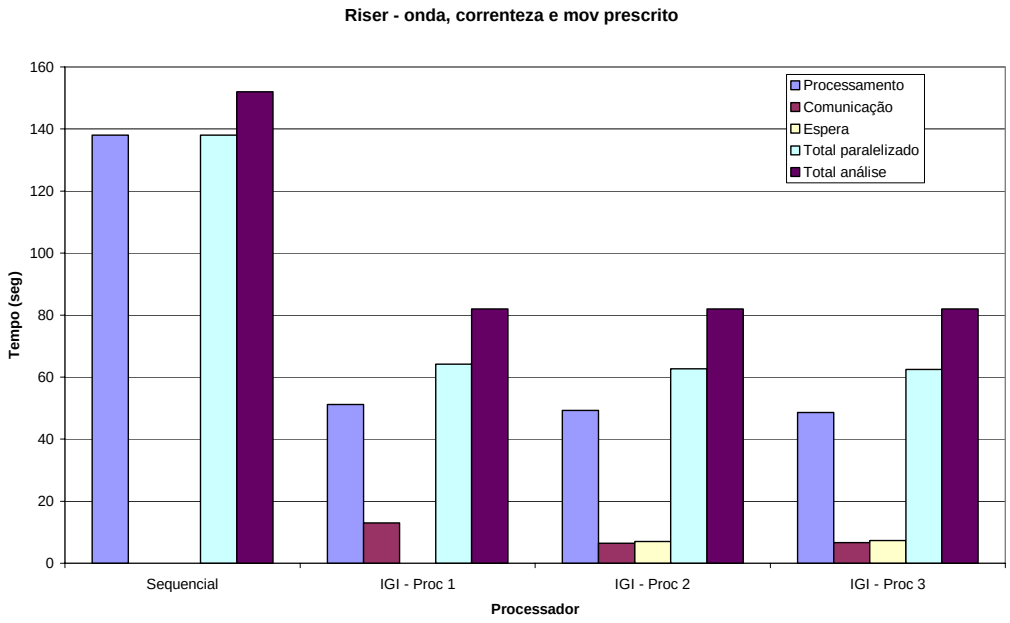


Figura 52 – Medidas de Desempenho

Da mesma forma como observado no exemplo anterior, o tempo de processamento em cada processador onde o PDII é utilizado é praticamente 1/3 do tempo de processamento da análise seqüencial. O tempo de comunicação é maior no processador 1 pois é este processador que concentra os valores de deslocamento na interface de todos os processadores e faz a compatibilidade de deslocamentos. O tempo de espera nos processadores 2 e 3 aparecem devido ao envio de informações ao processador 1 pelos outros processadores. O tempo total da análise com PDII apresenta-se muito menor que o tempo total com análise seqüencial.

11.4. PLATAFORMA P10

11.4.1. Características do Casco

A plataforma P10 é uma semi-submersível de perfuração com as seguintes características principais, extraídas de [48, 49]:

- LDA : 1200 m;
- Azimute de aproamento: 90^{o};
- Posição do centro de gravidade em relação ao eixo local: 16 m;
- Calado da plataforma: 20,0 m;
- Peso da plataforma: 2,3 E+05 kN.

Na Figura 53 é ilustrado o sistema de referência local (XYZ) e global (xyz) adotado. Verifica-se que o eixo x do sistema global possui o sentido positivo da proa a popa, os eixos x e y do sistema global formam um plano horizontal com nível do mar, os eixos X e Y do sistema local ou eixo estrutural formam um plano horizontal com a quilha.

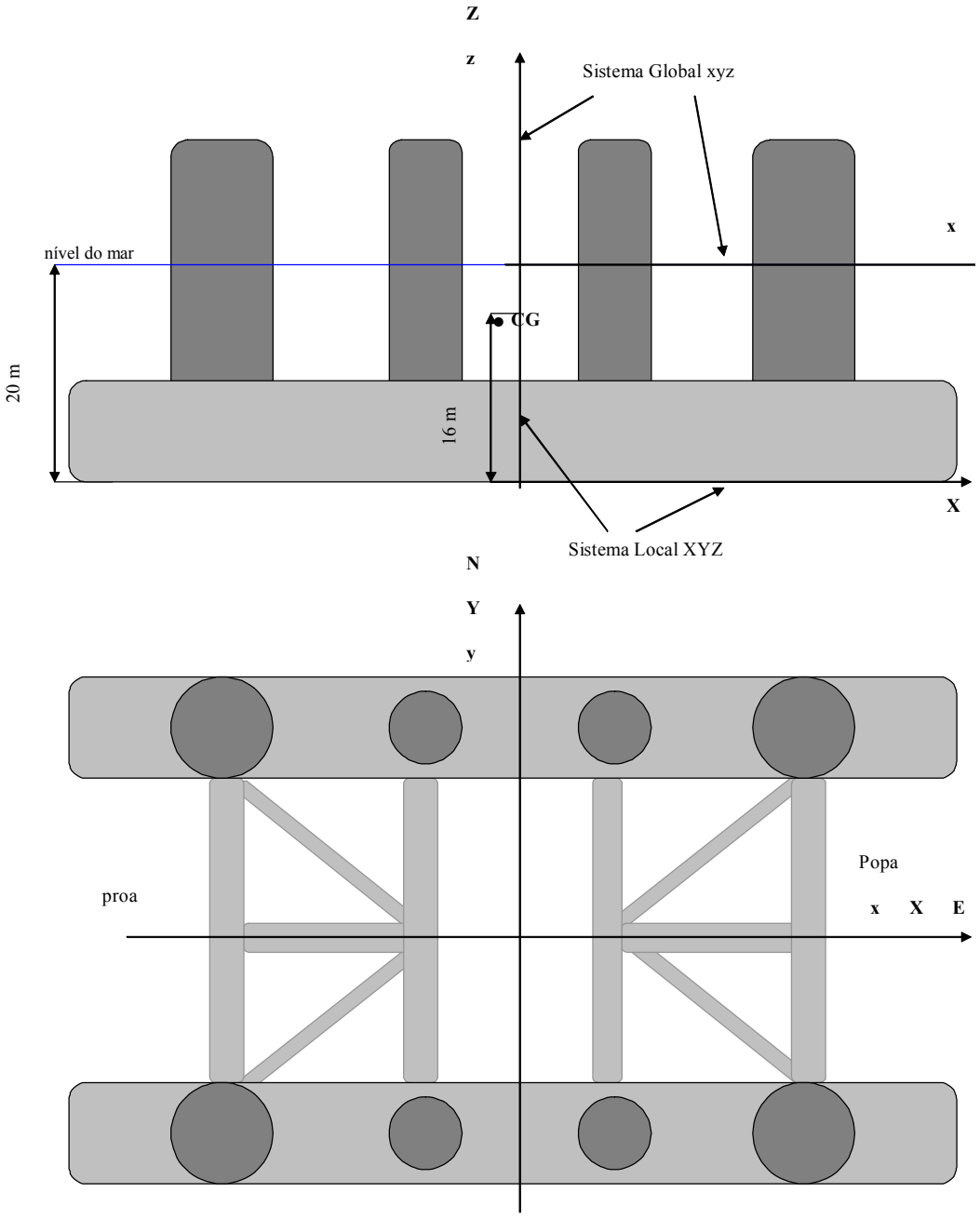


Figura 53 – Sistema de Referência da Plataforma P10

11.4.2.Características das Linhas de Ancoragem

A P10 é ancorada por um total de 8 linhas de ancoragem em configuração de catenária simples. As linhas são compostas por um trecho de amarra no fundo e no topo e cabo de aço intermediário. As características mais detalhadas das linhas de ancoragem são apresentadas na Tabela 13 e na Tabela 14.

Tabela 13 – Propriedades das Linhas de Ancoragem

Linha	Comprimentos (m)			Pré-tensão (kN)
	Amarra de Fundo	Cabo de Aço	Amarra de Topo	
Linha 1	926	1380	300	1292.2
Linha 2	926	1380	300	1298.9
Linha 3	926	1380	300	1355.6
Linha 4	926	1380	300	1340.4
Linha 5	926	1380	300	1340.4
Linha 6	926	1380	300	1355.6
Linha 7	926	1380	300	1298.9
Linha 8	926	1380	300	1292.2

Tabela 14 – Coordenadas de conexão das Linhas de Ancoragem

Linha	Coordenadas Globais			Azimute (^{o})
	X	Y	Z	
Linha 1	-35,00	32,50	-5,91	295.0
Linha 2	-31,00	32,50	-5,91	335.0
Linha 3	35,00	32,50	-5,91	25.0
Linha 4	35,00	32,50	-5,91	65.0
Linha 5	35,00	-32,50	-5,91	115.0
Linha 6	31,00	-32,50	-5,91	155.0
Linha 7	-31,00	-32,50	-5,91	205.0
Linha 8	-35,00	-32,50	-5,91	295.0

11.4.3.Características do Riser de Perfuração

O riser vertical é constituído de um riser rígido de 10.75” e está localizado no centro da plataforma e suas características principais em relação a dados geométricos e do material estão descritas na tabela 15.

Tabela 15 – Propriedades do riser

Diâm. ext (m)	0.2731
Diâm. int (m)	0.2318
E (kN/m^2)	2.07e8
Peso específico (kN/m^3)	77

11.4.4.Dados Ambientais

Para análise em questão foram considerados os seguintes carregamentos ambientais:

- Carregamentos atuantes no casco: correnteza e onda;
- Carregamentos atuantes nas linhas: correnteza e onda;

O perfil de corrente utilizado na simulação com a P10 é apresentado na tabela 16.

Tabela 16 – Perfil de Correnteza

Profundidade (m)	Velocidade (m/s)	Direção
0	0,70	NW
1200	0,00	NW

A onda utilizada nas análises é uma onda regular com Hmax = 5m e THmax = 10 seg também atuando na direção NW.

11.4.5.Modelo Numérico

CASCO

A embarcação é representada por um modelo reticulado de 36 elementos tubulares. Os dados necessários para compor o modelo são as coordenadas dos nós iniciais e finais de cada elemento, seus diâmetros equivalentes, e os respectivos coeficientes hidrodinâmicos.

A Figura 54 apresentam a visualização sólida da plataforma P10 levando em conta os diâmetros dos membros tubulares equivalentes.

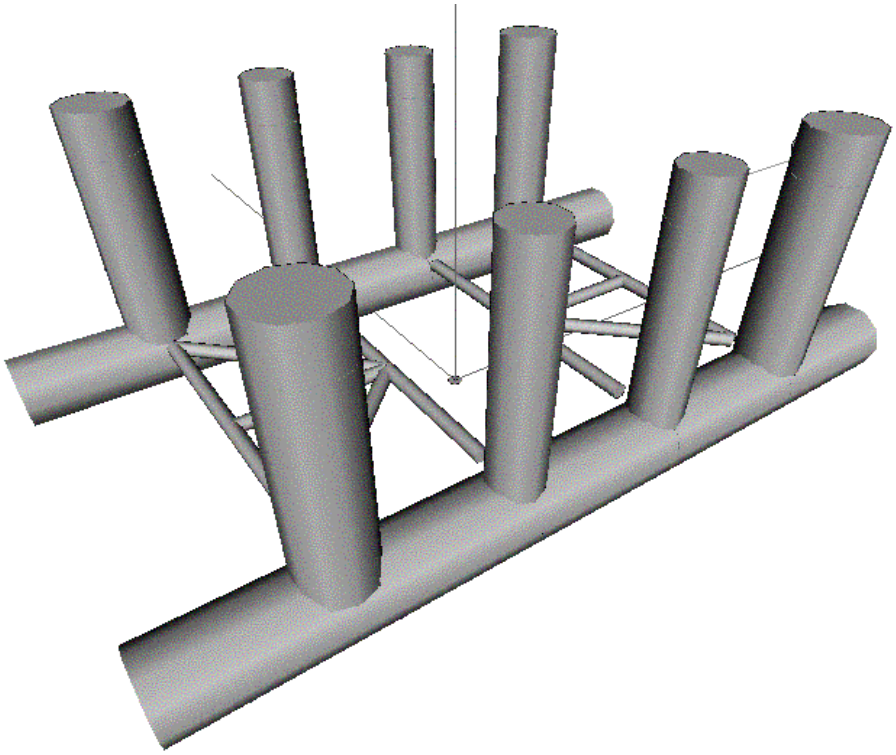


Figura 54 – Visualização Sólida do Modelo do Casco da P10

LINHAS DE ANCORAGEM

A discretização das linhas foi realizada com uma malha uniforme de elementos de treliça, resultando num total de 88 elementos de treliça para cada linha. Cada elemento tem um comprimento de 30 metros.

Na Figura 55 é ilustrado uma das linhas da plataforma P10 discretizada com elemento de treliça.

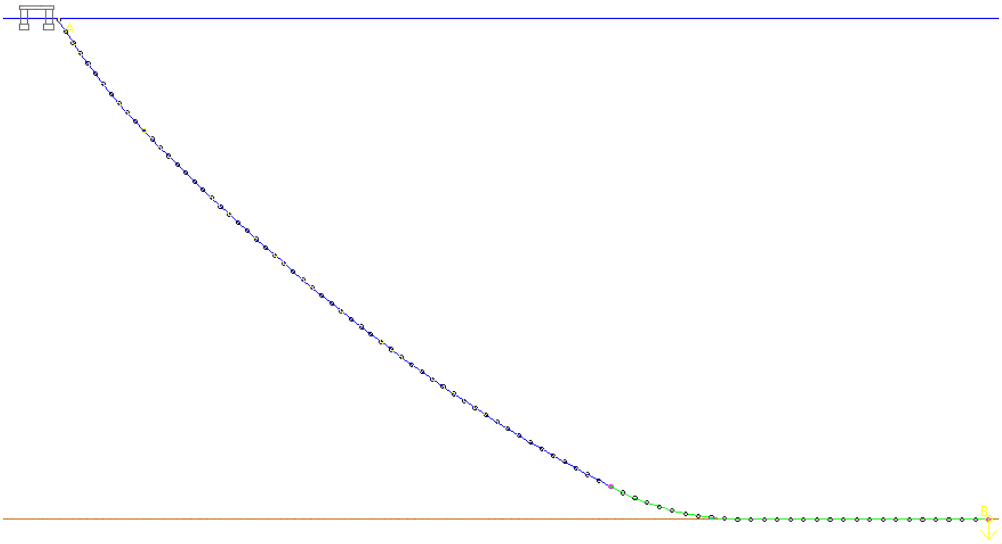


Figura 55 – Discretização de uma Linha de Ancoragem

RISER DE PERFURAÇÃO

Na análise do riser foram utilizados elementos de pórtico discretizados conforme apresentado na tabela 17.

Tabela 17 – Discretização do Riser

Segmento	Comprimento do segmento (m)	Comprimento dos elementos (m)	Número de elementos
1	1180	2.5	472

.

Na Figura 56 é ilustrado a configuração do riser vertical de perfuração.

Figura 56 – Configuração do riser

MODELO ACOPLADO

A seguir são apresentadas nas figuras 57, 58 e 59 o modelo acoplado da P10, incorporado as linhas de ancoragem.

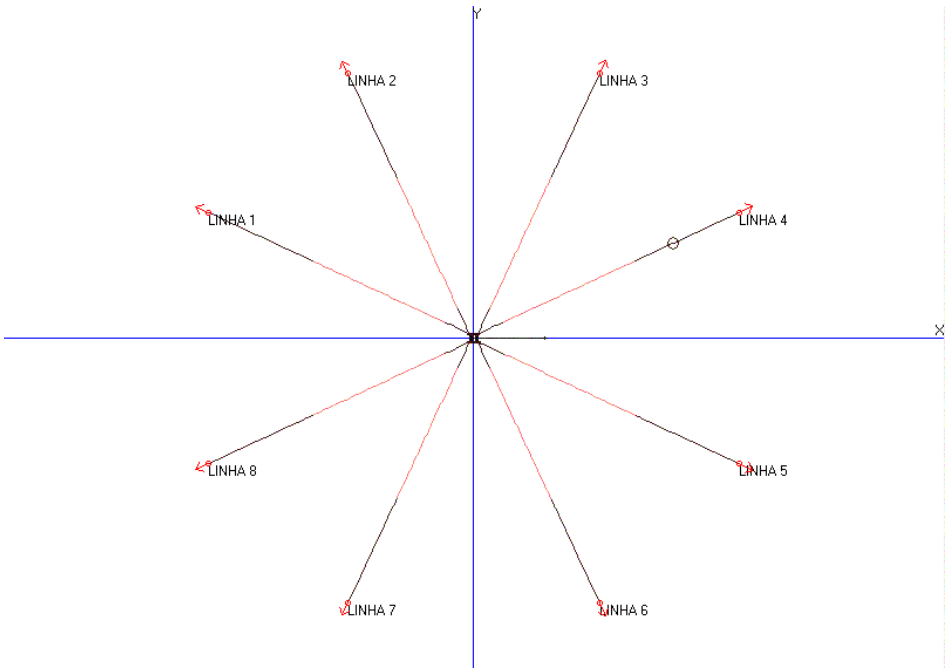


Figura 57 – Modelo Acoplado – Vista XY

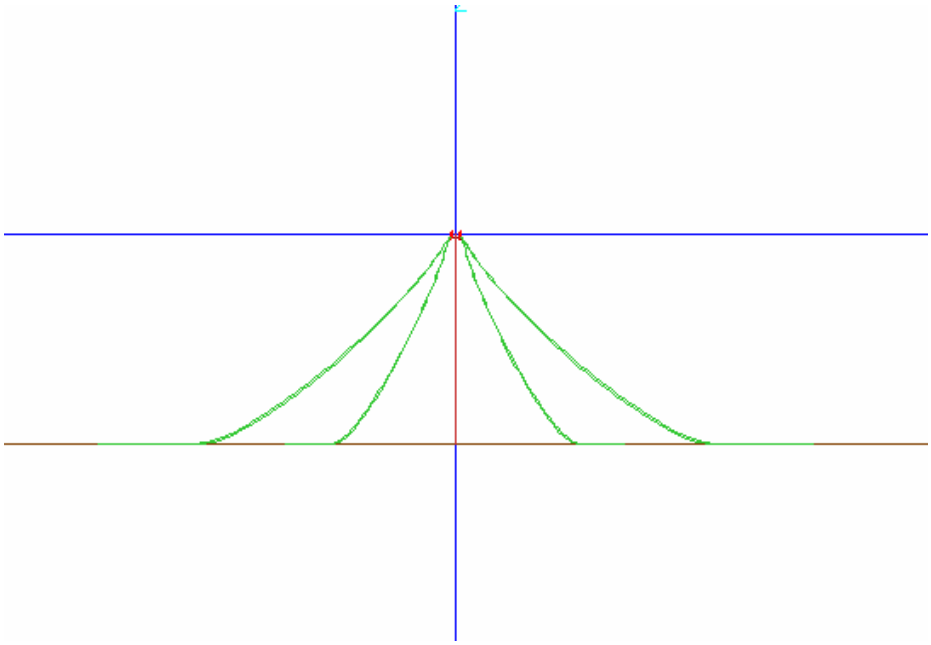


Figura 58 – Modelo Acoplado – Vista XZ

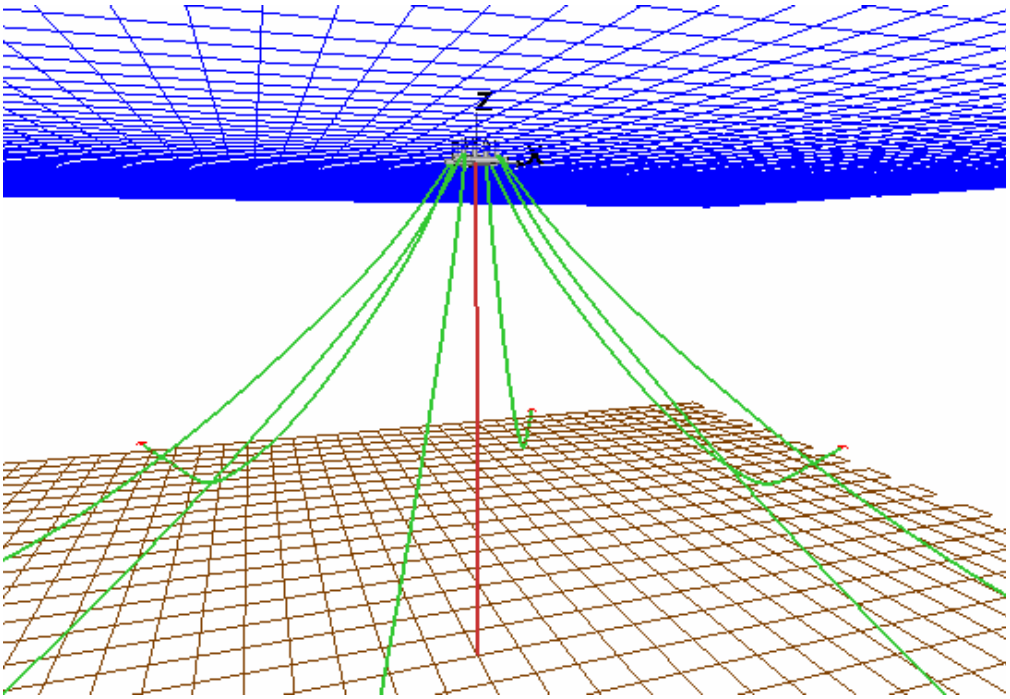


Figura 59 – Modelo Acoplado – Vista 3D

11.4.6. Subciclagem - Resultados

A subciclagem foi aplicada apenas às linhas de ancoragem. Desta forma, para melhor verificação desta implementação, o riser vertical não foi considerado na análise. A seguir a figura 60 com a plataforma e as linhas de ancoragem.

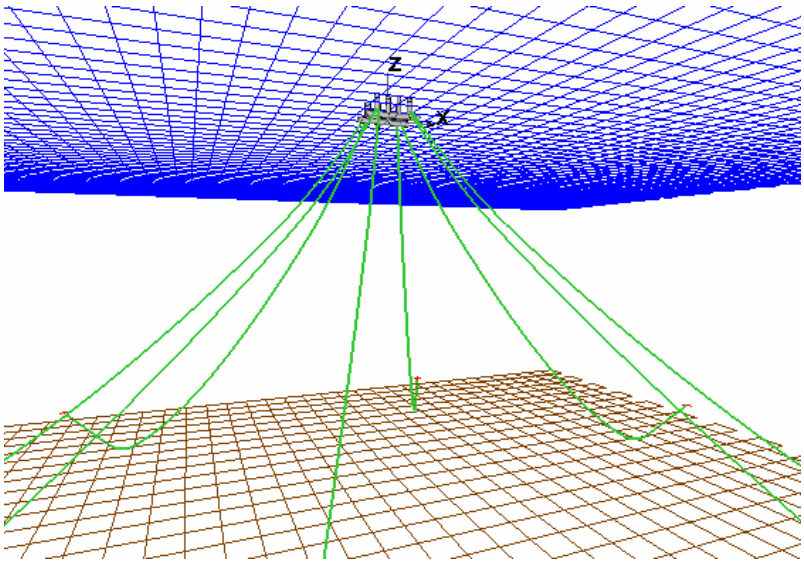


Figura 60 – Plataforma com Linhas de Ancoragem

Para utilização da subciclagem, conforme o exemplo anterior, utilizando-se da divisão de segmentos das linhas de ancoragem devido aos diferentes tipos de material utilizados (amarra e cabo), associou-se intervalos de tempo a cada segmento.

Desta forma utilizou-se dois intervalos de tempo correspondentes a cada segmento:

- Segmentos 1 (amarra topo) e 3 (amarra fundo): $\Delta t = 0.0005$ seg;

- Segmento 2 (cabo): $\Delta t = 0.00025$ seg.

Na análise tradicional, sem subciclagem, utilizou-se o menor intervalo de tempo para todos os elementos, $\Delta t = 0.00025$ seg.

Não foi utilizado amortecimento numérico nesta análise.

Nas figuras 61 e 62 são apresentados resultados comparativos de movimentos da embarcação e tração das linhas de ancoragem, entre a versão original do programa Prosim, sem o uso da subciclagem e a versão implementada com uso da subciclagem. Conforme pode-se observar os resultados são praticamente idênticos.

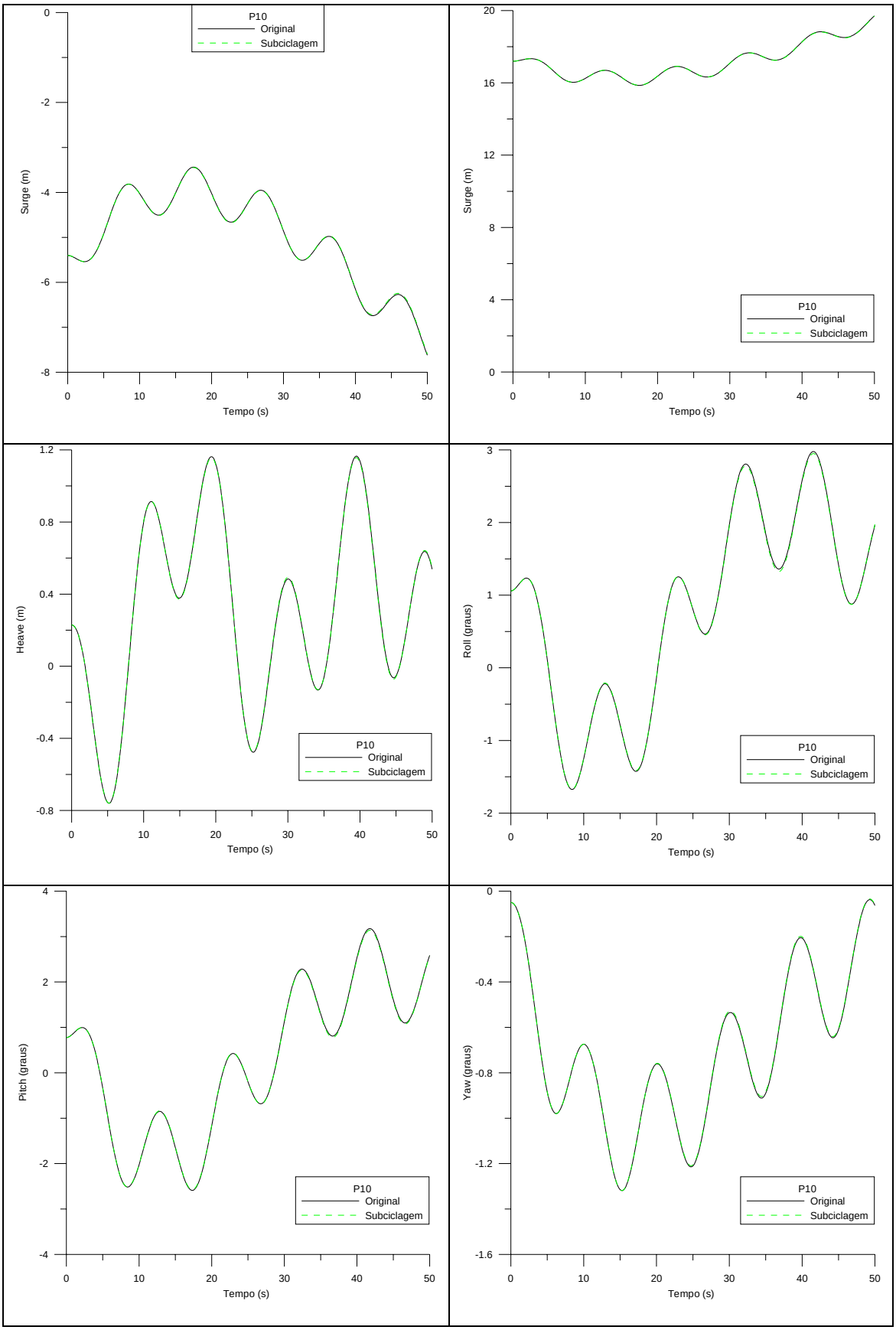
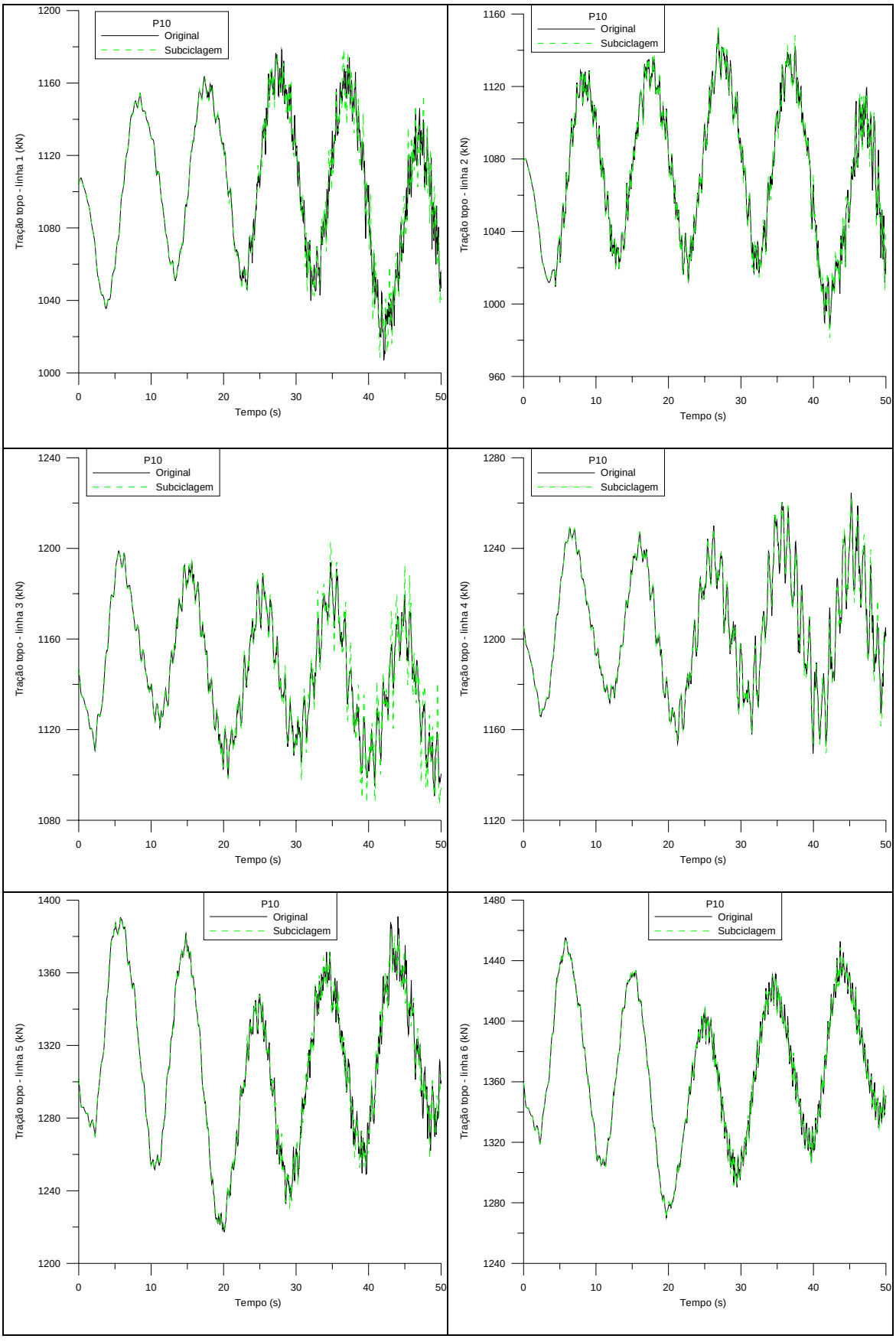


Figura 61 – Resultados comparativos – Movimentos P10


P10
Original
Subciclagem
Tração topo - linha 1 (kN)
Tração topo - linha 2 (kN)
Tração topo - linha 3 (kN)
Tração topo - linha 4 (kN)
Tração topo - linha 5 (kN)
Tração topo - linha 6 (kN)
Tempo (s)

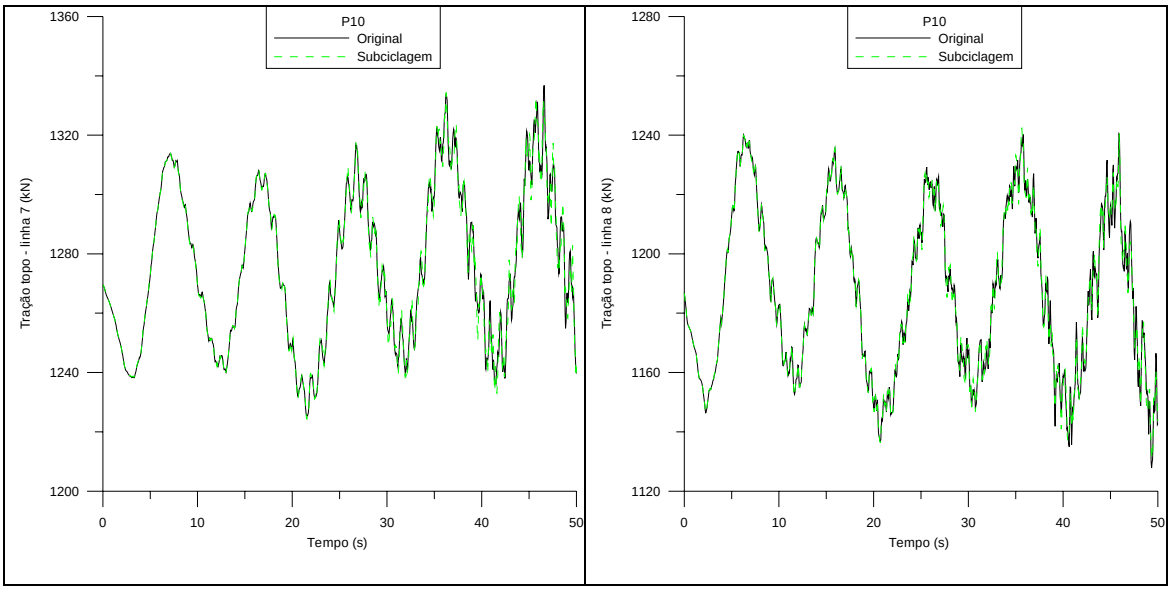


Figura 62 – Resultados comparativos – Tração topo das linhas

Com a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido nesta análise. A tabela 18 apresenta o tempo de CPU gasto.

Tabela 18 – Desempenho Computacional

	Tradicional	**Subciclagem**	**Economia %**
Tempo CPU (seg)	8665.6	5430.0	37.3

Com isto pode-se observar o ganho computacional significativo de 37.3% obtido neste caso.

11.4.7. PDII - Resultados

Para utilização do algoritmo PDII, neste exemplo, utilizou-se o seguinte procedimento:

- Modelou-se o riser de perfuração vertical o qual foi particionado em 2 segmentos iguais. Sobre este riser utilizou-se o PDII em 2 processadores;
- As linhas de ancoragem foram agrupadas duas a duas e distribuídas em 4 processadores, cada processador com 2 linhas conforme o procedimento implementado por Franco em [47]. Esta divisão é adotada também na versão original do Prosim, utilizada para comparação de resultados e desempenho.

Para utilização do algoritmo PDII, neste exemplo, dividiu-se o riser ao meio em dois segmentos conforme ilustrado na figura 63 a seguir:

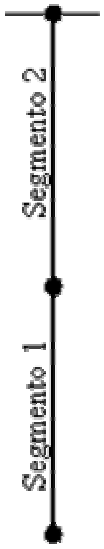


Figura 63 – Divisão do riser

A seguir são apresentados os principais parâmetros utilizados nesta análise:

- Intervalo de tempo: 0.0025 seg;

- Tolerância de deslocamentos: 0.001

- Tolerância de forças: 0.01

- Amortecimento numérico (coeficiente αB): -0.33.

- Amortecimento proporcional à massa: não foi considerado.

- Amortecimento proporcional à rigidez: não foi considerado.

Nas figuras 64 e 65 são apresentados resultados comparativos de movimentos da embarcação e tração no topo do riser vertical, entre a versão original do programa Prosim e a versão implementada com uso do algoritmo PDII. Conforme pode-se observar os resultados são praticamente idênticos.

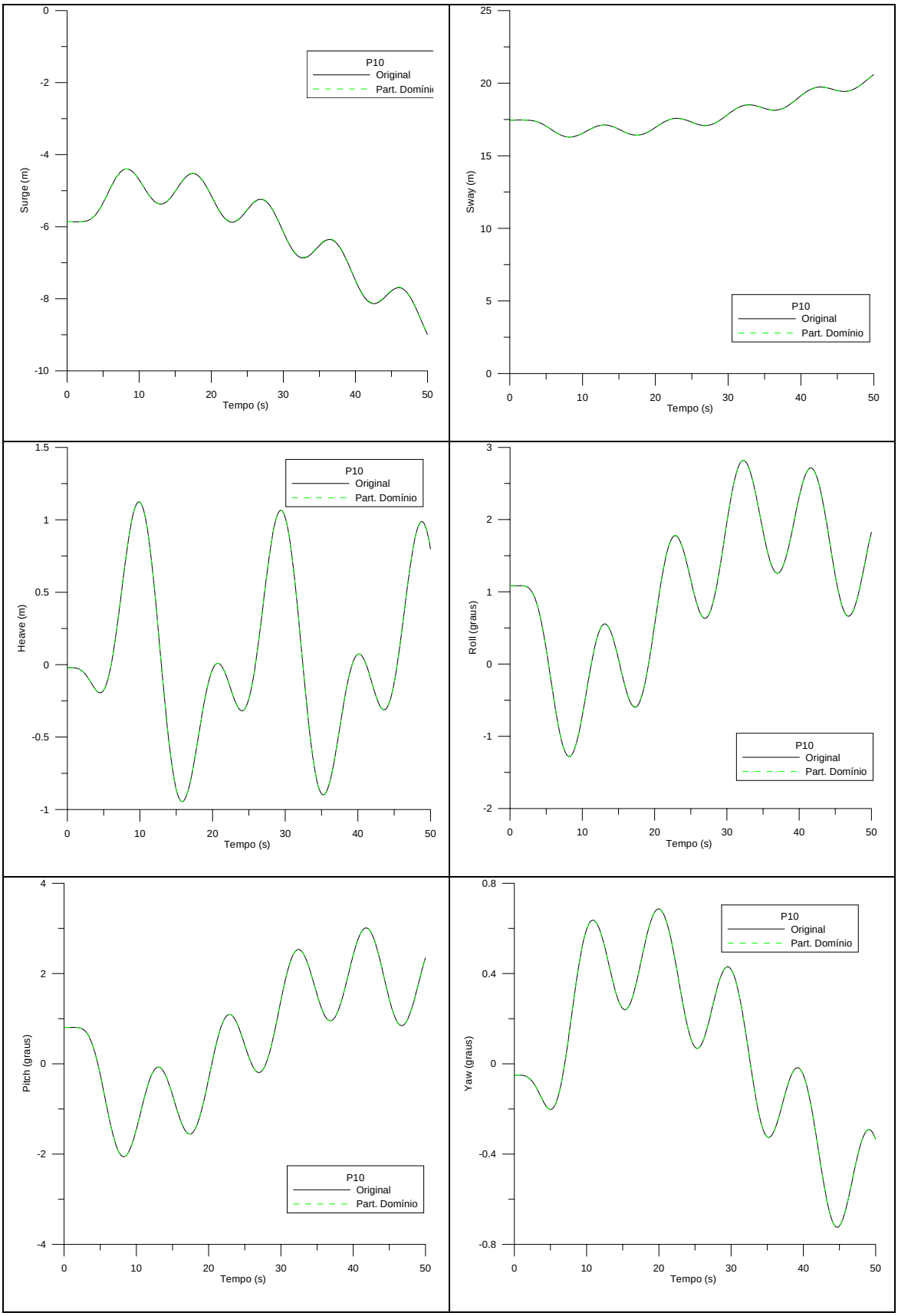


Figura 64 – Resultados comparativos – Movimentos P10

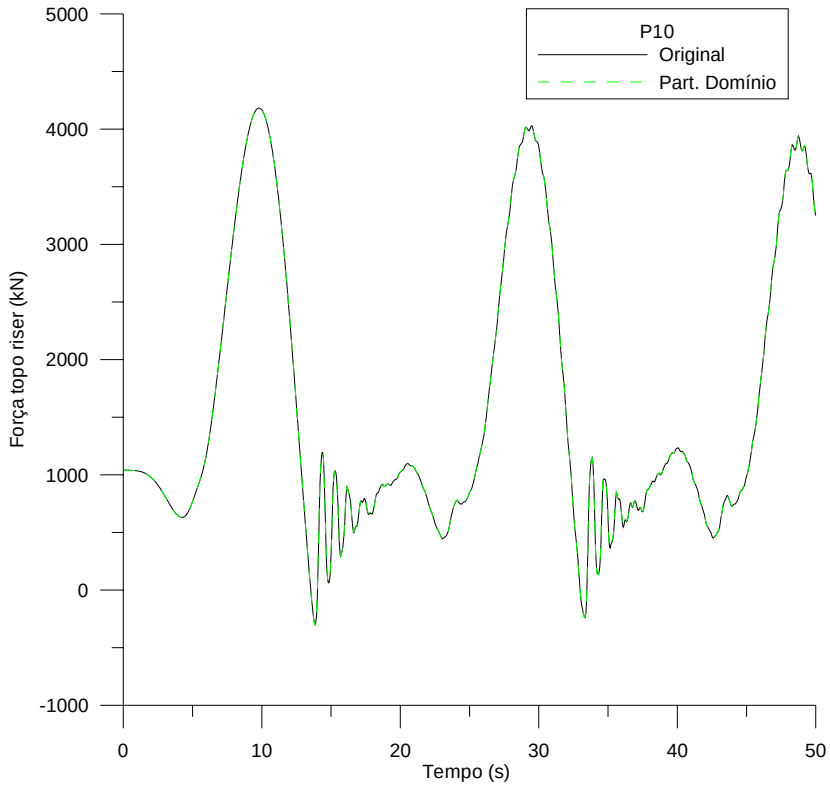


Figura 65 – Resultados comparativos – Tração topo riser perfuração

Visto a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido nesta análise. A análise tradicional apresenta o resultado da análise em que as linhas de ancoragem são distribuídas duas a duas entre os processadores 1 a 4 e o riser vertical é analisado seqüencialmente no processador 5. A análise com PDII apresenta o resultado da análise em que as linhas de ancoragem são distribuídas duas a duas entre os processadores 1 a 4 e o riser vertical é particionado em 2 segmentos e distribuído nos processadores 5 e 6. A tabela 19 apresenta o tempo de CPU gasto.

Tabela 19 – Desempenho Computacional

	Tradicional	**PDII**	**Economia %**
Tempo CPU (seg)	5390.9	3417.0	36.6

Esta análise apresentou um ganho significativo de 36.6% de economia computacional.

Para uma melhor observação e entendimento do desempenho obtido serão apresentados resultados mais detalhados em termos de tempo de espera, tempo de comunicação e tempo de processamento de cada processador.

As figuras 66 e 67 apresentam o tempo de espera, processamento, comunicação de cada processador, além do tempo da análise correspondente à fração paralelizável e o tempo total de análise. A figura 66 utiliza o procedimento tradicional com a divisão das linhas de ancoragem pelos processadores porém sem particionamento do riser de perfuração. Neste caso o riser de perfuração é analisado pelo processador 5. A figura 67 utiliza o método PDII onde o riser de perfuração é particionado e analisado pelos processadores 5 e 6. As linhas de ancoragem são analisadas pelos processadores 1 a 4 onde cada processador analisa 2 linhas de ancoragem.

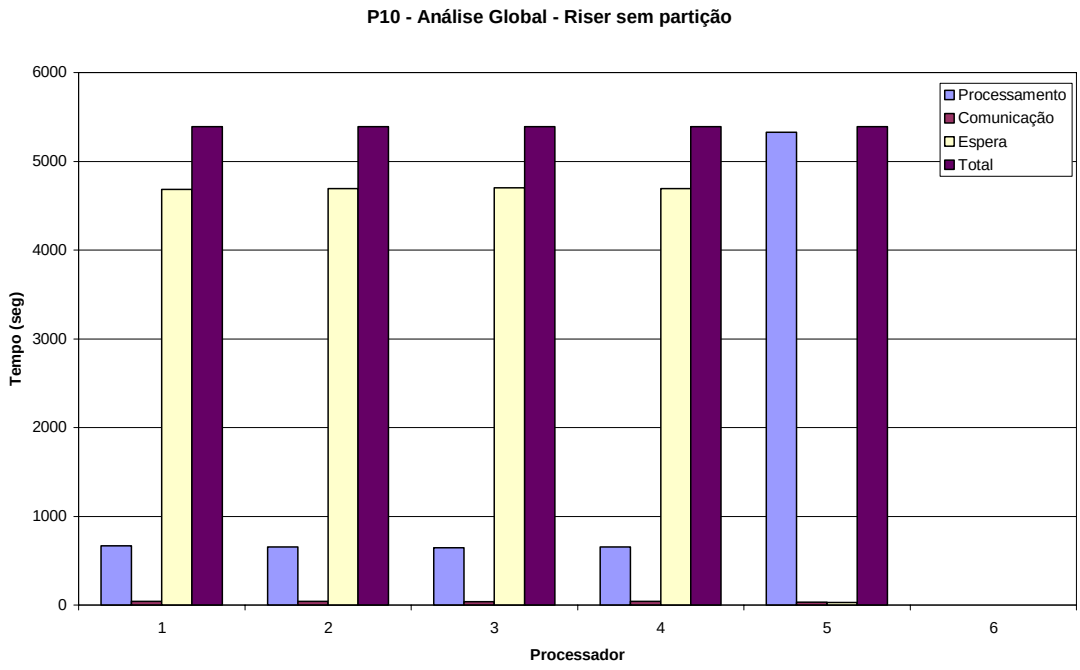


Figura 66 – Medida de Desempenho - Tradicional

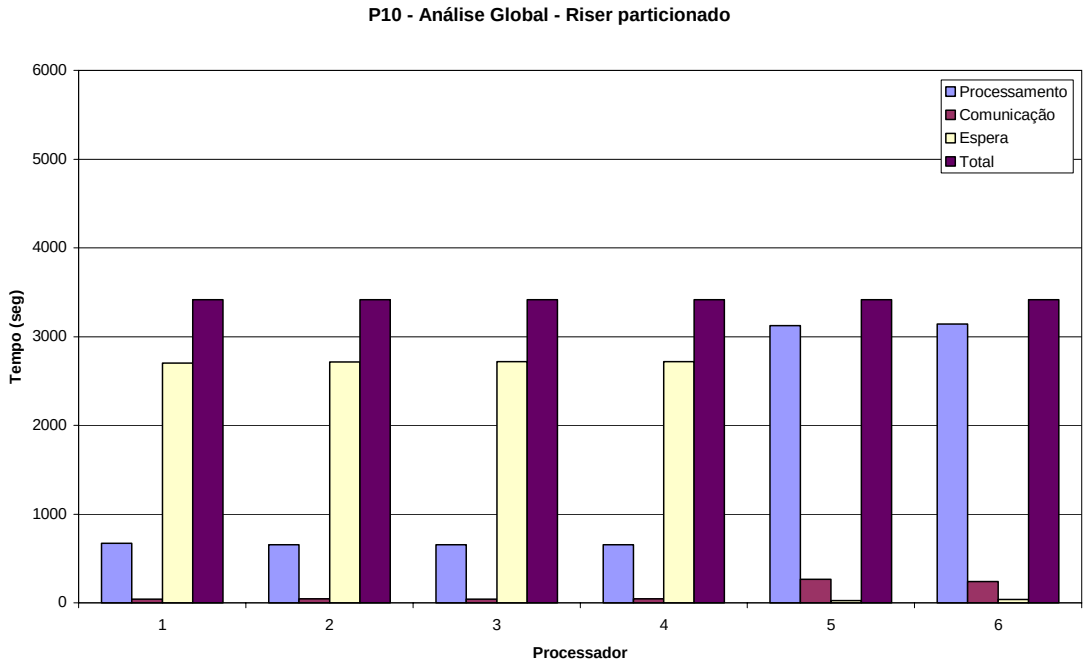


Figura 67 – Medida de Desempenho - PDII

Observa-se que devido ao elevado número de graus de liberdade do riser de perfuração em relação às linhas de ancoragem o tempo de espera dos processadores 1 a 4 é bem elevado. Com a utilização do PDII o tempo de espera deste processadores diminui assim como o tempo de processamento dos processadores 5 e 6 e o tempo total da análise em todos os processadores.

O tempo de comunicação entre os processadores 5 e 6 sofre um ligeiro aumento devido à comunicação necessária à compatibilidade dos nós de interface. O tempo de espera dos processadores 5 e 6 é quase nulo devido à igual divisão de graus de liberdade do riser de perfuração entre estes processadores e à comunicação total da análise se restringir a estes 2 processadores.

A figura 68 apresenta mais detalhes especificamente da análise do riser de perfuração seqüencialmente e com a utilização do PDII.

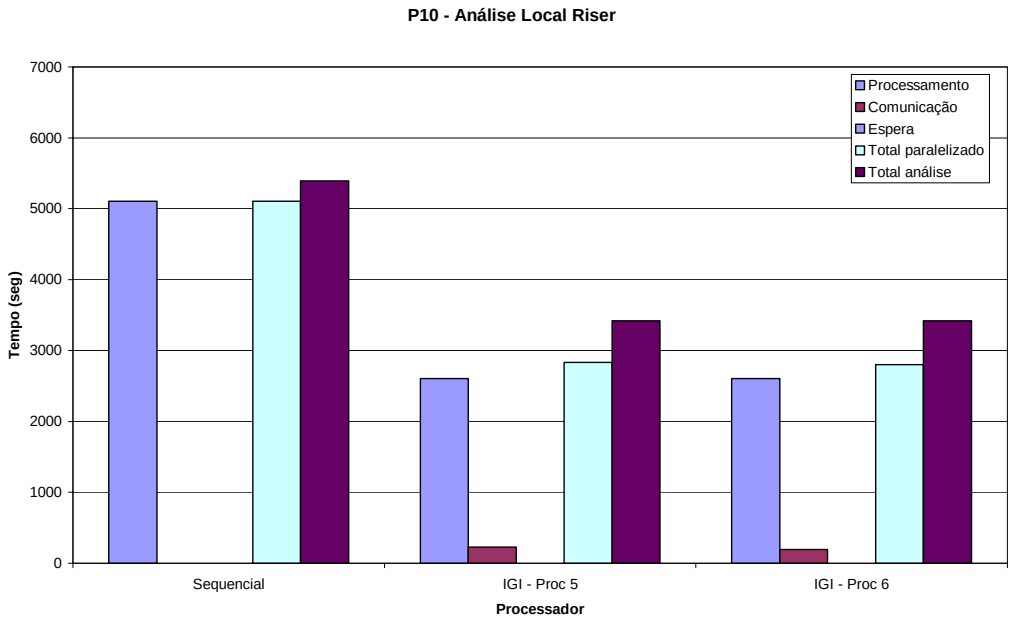


Figura 68 – Medida de Desempenho

Neste gráfico foram tomados tempo da fração paralelizada e do tempo total da análise. Pode-se observar o ganho considerável no tempo de processamento e no tempo total da análise comparando-se com a análise do riser vertical em 1 processador.

11.5. PLATAFORMA P18

A plataforma P18 é ancorada por 8 linhas. Possui um total de 73 risers, de produção, exportação, injeção e umbilicais, sendo 72 risers flexíveis e um SCR (Steel Catenary Riser), todos com configurações em catenária livre.

11.5.1. Características do Casco

A seguir são apresentadas algumas características da plataforma P18, extraídas de [13]:

- lâmina d'água: 910 m;
- azimute da plataforma: 203^{o};
- posição do centro de gravidade em relação ao eixo local: 19,30 m;
- Calado da plataforma: 23,10 m;
- peso da plataforma: 3.27E+05 kN.

A tabela 20 apresenta os raios de giração da P18.

Tabela 20 – Raio de Giração

	Valor
RXX	30.49
RYY	29.35
RZZ	32.64

Na Figura 69 é ilustrado o sistema de referência local (XYZ) e global (xyz) adotado. Empregaram-se coeficientes hidrodinâmicos gerados pelo programa WAMIT. O sistema de referência do WAMIT é o mesmo do Prosim. A Figura 70 mostra o modelo do casco da plataforma P18 utilizado na análise prévia do programa WAMIT para a obtenção das matrizes de amortecimento potencial e dos coeficientes de deriva.

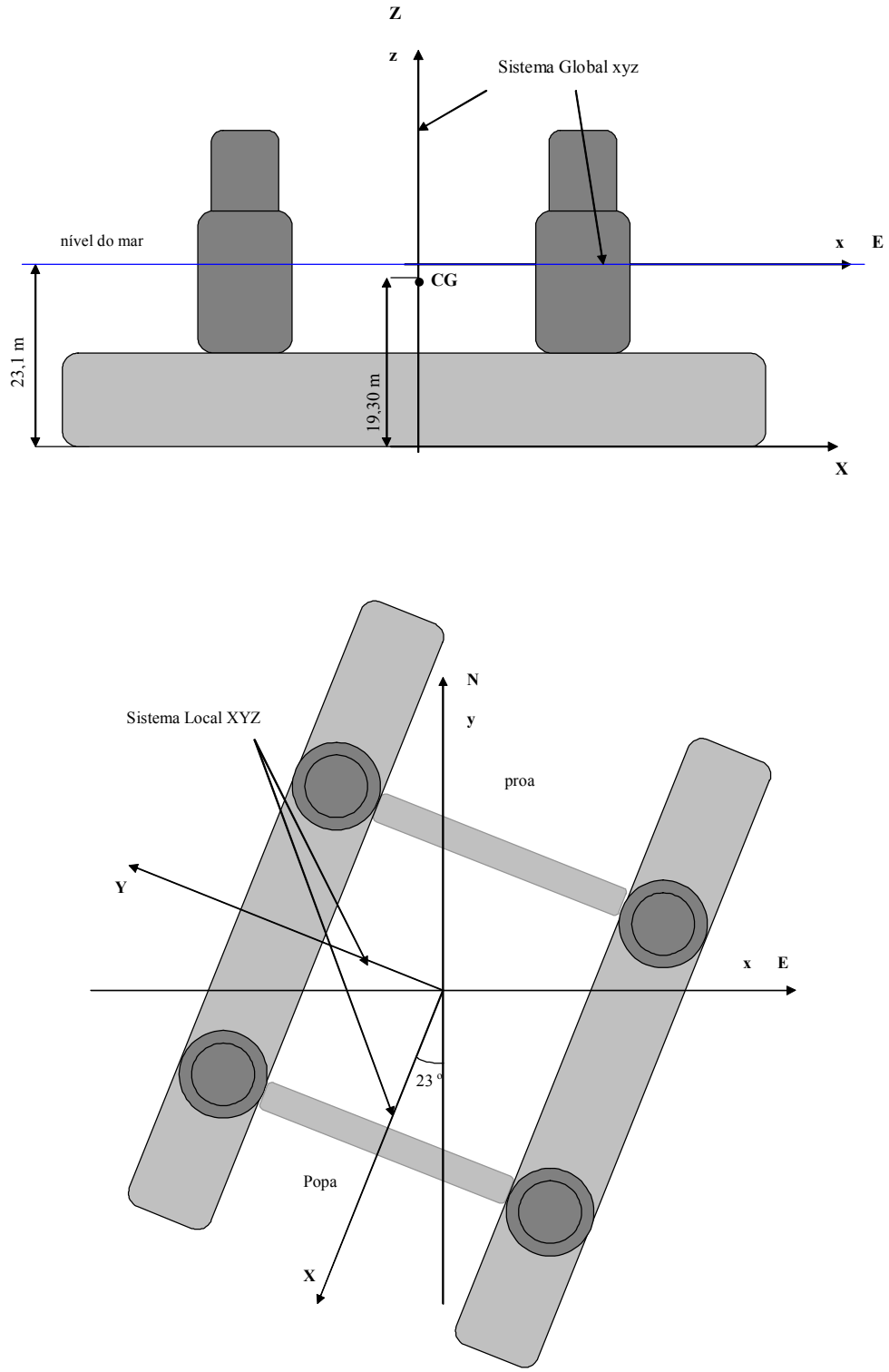


Figura 69 – Sistema de Referência da Plataforma P18

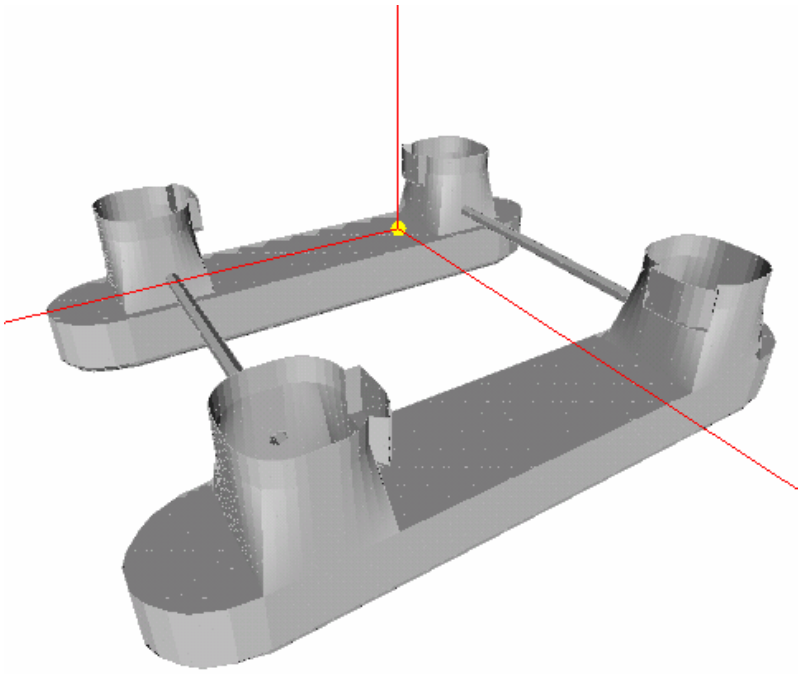


Figura 70 – Modelo do Casco da P18 usado pelo Wamit

11.5.2. Características das Linhas de Ancoragem

A P18 é ancorada por um total de 8 linhas de ancoragem compostas por um trecho de amarra no fundo e no topo e cabo de aço intermediário. As características das linhas de ancoragem são apresentadas na a seguir nas tabelas 13 e 14.

Tabela 21 – Propriedades das Linhas de Ancoragem

Linha	Comprimentos (m)			Projeção Horizontal (m)
	Amarra de Fundo	Cabo de Aço	Amarra de Topo	
Linha 1	1060	2377,735	368	2377,74
Linha 2	1290	2430,074	230	2430,07
Linha 3	1400	2597,106	354	2597,11
Linha 4	1230	2482,344	283.5	2482,34
Linha 5	1110	2513,828	439	2513,83
Linha 6	1480	2549,931	157	2549,93
Linha 7	1050	2418,237	610	2418,24
Linha 8	1735	2824,911	207	2824,91

Tabela 22 – Coordenadas de Conexão e Azimutes das Linhas de Ancoragem

Linha	Coordenadas Globais			Azimute (^{o})
	X	Y	Z	
Linha 1	-30,8	-37,4	-7,7	354.8
Linha 2	-25,5	-37,4	-7,7	325.5
Linha 3	25,5	-37,4	-7,7	268.1
Linha 4	30,8	-37,4	-7,7	227.3
Linha 5	30,8	37,4	-7,7	173.2
Linha 6	25,5	37,4	-7,7	131.8
Linha 7	-25,5	37,4	-7,7	74.8
Linha 8	-30,8	37,4	-7,7	39.8

11.5.3. Características dos Risers

A Tabela 23 apresenta as características dos risers, em termos de azimutes e os ângulos de topo. A P18 tem um total de 73 risers conectados, incluindo produtores, injetores e de exportação (dentre eles o SCR).

Tabela 23 – Risers Conectados à Plataforma

Número do Poço	Tipo de Riser		Azimute	Ângulo de Topo
1	Injetor	*4”*	7	3.81
	Umbilical	*UH*	7	3.81
2	Anular	*2.5”*	13	8.42
	Umbilical	*UH*	14	4.40
	Produtor	*4”*	12	8.75
3	Umbilical	*UH*	18	5.00
	Injetor	*4”*	17	4.40
4	Injetor	*4” ISU*	23	7.60
	Gasoduto	*8”*	28	8.00
5	Anular	*2.5”*	34	6.11
	Umbilical	*UH*	35	6.00
	Produtor	*4”*	36	4.44
6	Umbilical	*UH*	39	4.15
	Injetor	*4”*	41	4.15
7	Anular	*2.5”*	85	4.85
	Umbilical	*UH*	86	4.82
	Produtor	*4”*	84	4.76
8	Umbilical	*UH*	82	4.80
	Injetor	*4”*	88	4.80
9	Anular	*2.5”*	104	7.90
	Umbilical	*UH*	109	7.30
	Produtor	*4”*	103	7.90
10	Anular	*2.5”*	110	4.06
	Umbilical	*UH*	111	4.14
	Produtor	*4”*	106	3.88
11	Umbilical	*UH*	114	4.20
	Injetor	*4”*	116	4.20
12	Anular	*2.5”*	119	5.00
	Umbilical	*UH*	121	5.00
	Produtor	*4”*	124	4.52
13	Anular	*2.5”*	126	5.98
	Umbilical	*UH*	128	5.40
	Produtor	*4”*	127	5.10
14	Anular	*2.5”*	179	5.91
	Umbilical	*UH*	180	7.50
	Produtor	*4”*	182	6.38
15	Anular	2.5”	198	7.00
	Umbilical	UH	199	7.00
	Produtor	4”	197	7.00
16	Anular	2.5”	201	7.24
	Umbilical	UH	200	6.00
	Produtor	4”	202	6.81
	SCR – Gasoduto	10”	192.11	21.67

Número do Poço	Tipo de Riser		Azimute	Ângulo de Topo
17	Anular	2.5”	211	4.33
	Umbilical	UH	212	4.09
	Produtor	4”	213	3.98
18	Anular	2.5”	214	2.38
	Umbilical	UH	215	5.50
	Produtor	4”	218	5.23
19	Umbilical	UH	220	4.40
	Injetor	4”	221	3.89
20	Anular	2.5”	280	7.00
	Umbilical	UH	284	7.00
	Produtor	6”	285	7.00
21	Anular	2.5”	289	4.74
	Umbilical	UH	290	4.75
	Produtor	4”	288	4.76
	Gasoduto	10”	299	6.26
	Oleoduto	12”S	296	6.96
	Oleoduto	12”N	290	8.28
22	Anular	2.5”	319	3.87
	Umbilical	UH	320	4.00
	Produtor	4”	311	9.14
23	Anular	2.5”	318	5.00
	Umbilical	UH	317	5.00
	Produtor	4”	319	5.00
24	Umbilical	UH	317	4.50
	Injetor	4”	316	4.05
25	Anular	2.5”	305	8.82
	Umbilical	UH	309	7.93
	Produtor	4”	312	9.02
26	Umbilical	UH	323	5.70
	Injetor	4”	321	6.00

11.5.4. Dados Ambientais

Para a análise em questão foram considerados os seguintes carregamentos ambientais:

- Carregamentos atuantes no casco: onda, correnteza e vento;
- Carregamentos atuantes nas linhas: onda, correnteza, gravitacional e amortecimento hidrodinâmico.

Foi considerado para este exemplo onda regular unidirecional com Hmax = 6.0 m, THmax = 11.0 seg e direção vindo de NE.

Para o vento, considerou-se o espectro proposto pela API, com velocidade média de 13.709 nós, ângulo de ataque em relação ao eixo X Global de 29.6^{o} e direção vindo de N.

O perfil de corrente utilizado na simulação com a P18 é o apresentado na tabela 24.

Tabela 24 – Perfil de Correnteza

Profundidade (m)	Velocidade (m/s)	Direção
0	0.45	SW
50	0.46	SW
100	0.42	SWS
150	0.42	SWS
250	0.25	SWS
350	0.13	S
450	0.09	SE
550	0.12	E
650	0.18	NE
750	0.18	NE
900	0.23	NE
910	0	NE

11.5.5. Modelo Numérico

CASCO

O casco da plataforma é representado por um modelo reticulado de 36 elementos tubulares. As Figuras 71 e 72 apresentam, respectivamente, o esquema reticulado da plataforma P18 e a visualização sólida da plataforma P18 levando em conta os diâmetros dos membros tubulares equivalentes. Além disso, foram calculados os valores de coeficientes hidrodinâmicos CD e CM para cada trecho destes membros, baseando-se na Norma da DNV/POSMOOR , como descrito em [13].

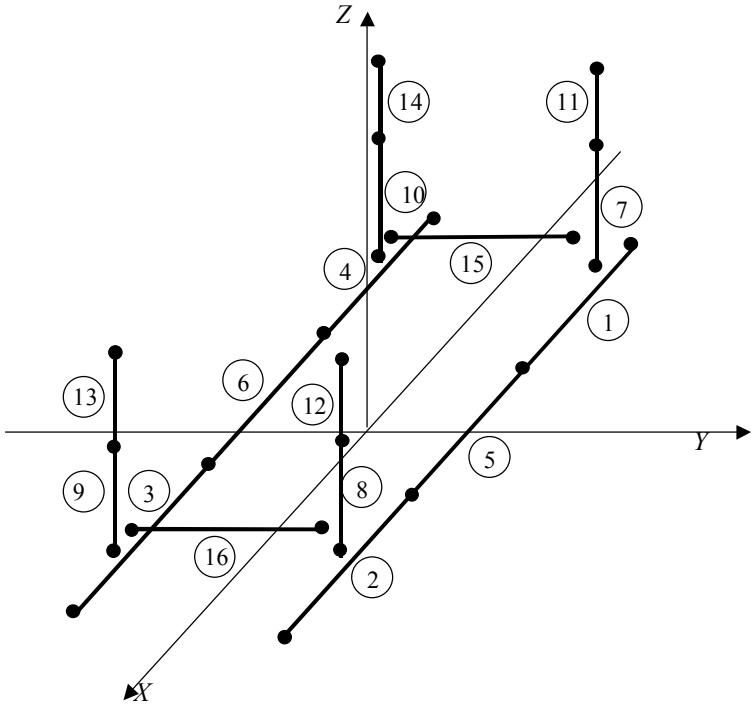


Figura 71 – Esquema Reticulado

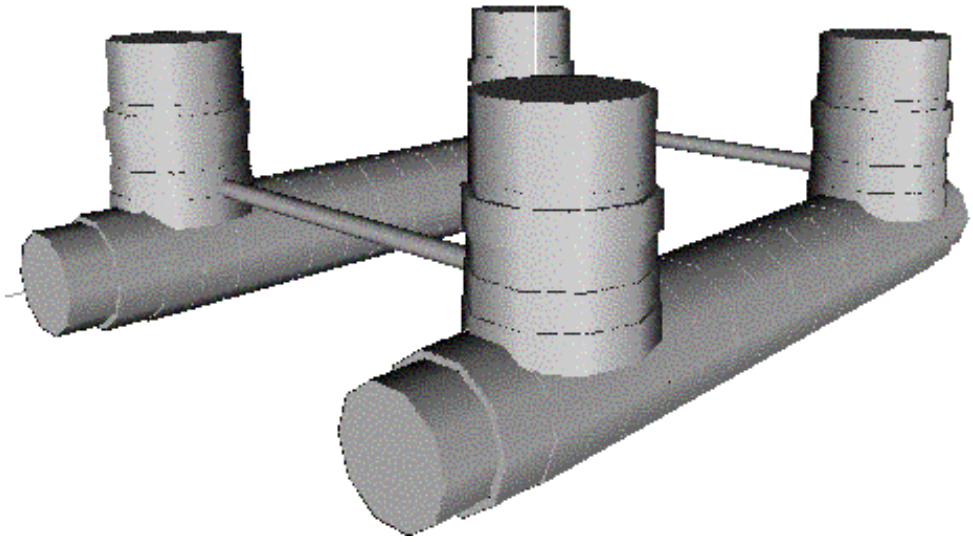


Figura 72 – Visualização Sólida do Modelo do Casco

CONJUNTO DE LINHAS DE ANCORAGEM E RISERS

As linhas e risers flexíveis foram discretizadas com elementos de treliça espacial e o SCR foi discretizado com elementos de pórticos espacial. As Tabelas 25, 26 e 27 apresentam, respectivamente, a malha típica de uma linha de ancoragem da P18, a malha típica de um riser flexível da P18 e a malha do SCR. A Tabela 28 apresenta o número total de elementos e número de equções para todas as linhas do modelo.

Tabela 25 – Malha Típica de uma Linha de Ancoragem

Segmento	Comprimento do Segmento (m)	Comprimento Inicial (m)	Comprimento Final (m)
1 – Topo	368	20	20
2	1250	20	20
3 – Âncora	1060	20	20

Tabela 26 – Malha Típica de um Riser Flexível

Segmento	Comprimento do Segmento (m)	Comprimento Inicial (m)	Comprimento Final (m)
1 – Topo	400	25	5
2	400	25	25
3	100	5	25
4	200	5	5
5 – Fundo	100	25	5

Tabela 27 – Malha do SCR

Segmento	Comprimento do Segmento (m)	Comprimento Inicial (m)	Comprimento Final (m)
1 – Topo	1,103	0,1575	0,1575
2	108,757	1	0,1575
3	100	20	1
4	926,304	20	20
5	100	5	20
6	130,786	5	5
7 – Fundo	700	20	5

Tabela 28 – Número de elementos e Graus de Liberdade de cada Linha

Linha	Tipo	Tipo de Elemento	Número de Elementos	Número de Equações
1 - 8	Riser Flexível	Treliça	60	180
9	Riser Flexível	Treliça	80	240
10-42	Riser Flexível	Treliça	60	180
43	Riser Rígido	Pórtico	264	1584
44-51	Riser Flexível	Treliça	60	180
52-54	Riser Flexível	Treliça	72	219
55-57	Riser Flexível	Treliça	60	180
58	Riser Flexível	Treliça	69	207
59-60	Riser Flexível	Treliça	66	198
61	Riser Flexível	Treliça	62	189
62-73	Riser Flexível	Treliça	60	180
74	Ancoragem	Treliça	133	402
75	Ancoragem	Treliça	138	420
76	Ancoragem	Treliça	150	453
77	Ancoragem	Treliça	138	417
78	Ancoragem	Treliça	140	423
79	Ancoragem	Treliça	144	435
80	Ancoragem	Treliça	144	441
81	Ancoragem	Treliça	159	480

Nas Figuras 73, 74 e 75 são ilustrados respectivamente a configuração típica de uma linha de ancoragem, do SCR e de um riser flexível da plataforma P18. Em todas as figuras citadas tem-se a configuração discretizada.

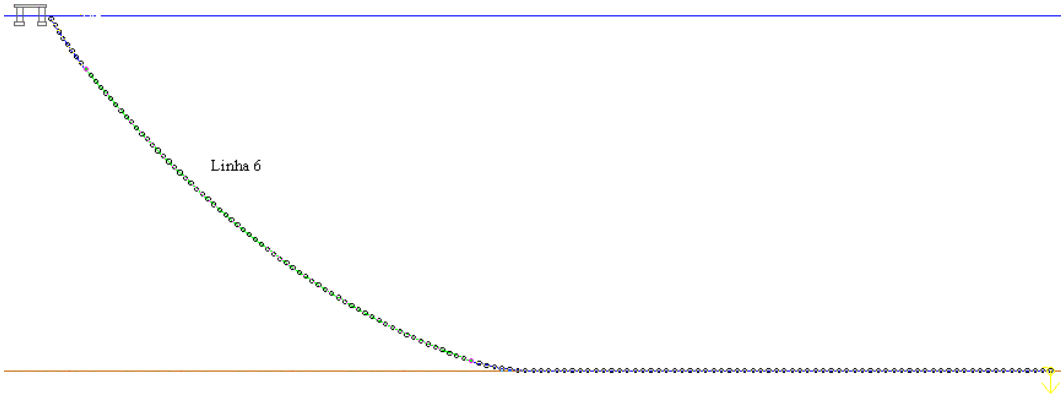


Figura 73 – Configuração Típica de uma Linha de Ancoragem

Figura 74 – Configuração do SCR

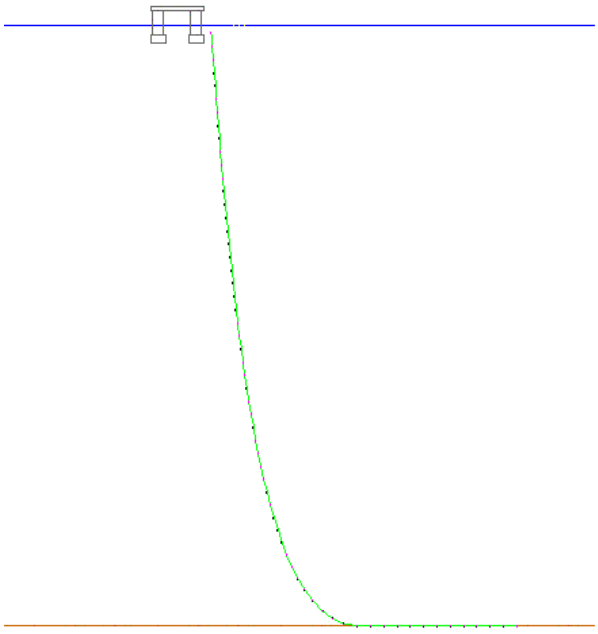


Figura 75 – Configuração de um Riser de Produção

MODELO ACOPLADO

A seguir, nas figuras 76, 77, 78 e 79, são apresentadas figuras ilustrando o modelo acoplado da P18, incorporado as linhas de ancoragem e aos risers.

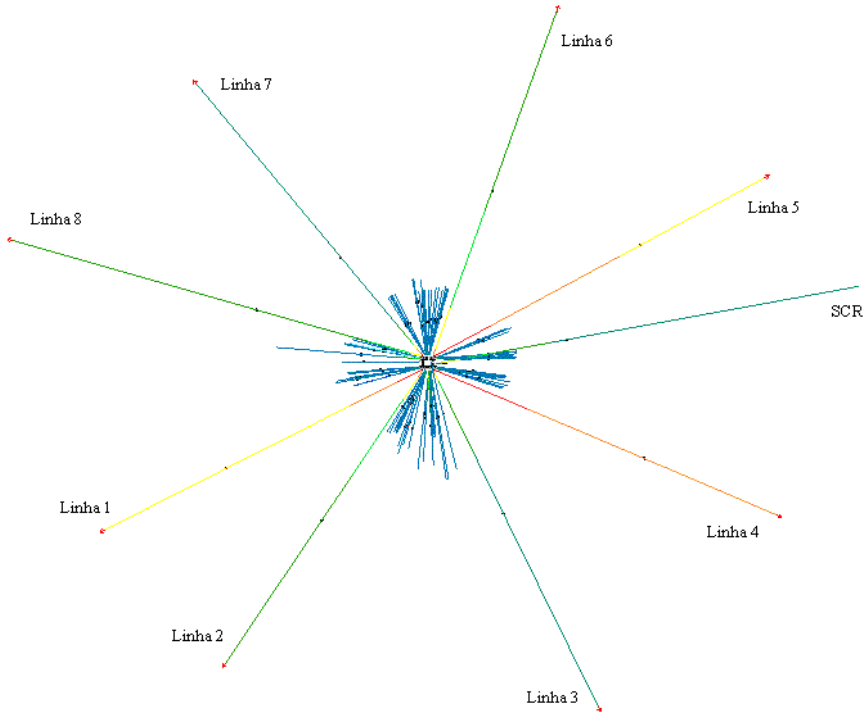


Figura 76 – Modelo Acoplado – Vista XY

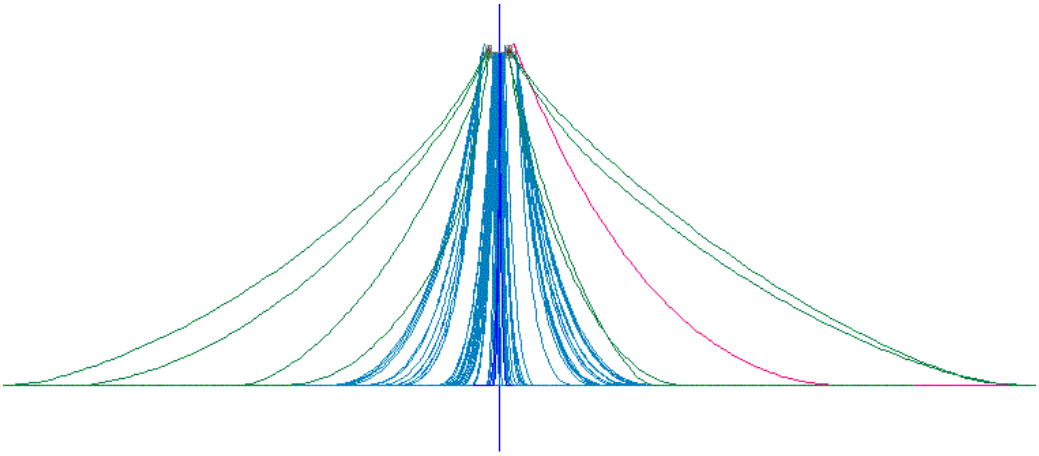


Figura 77 – Modelo Acoplado – Vista Lateral

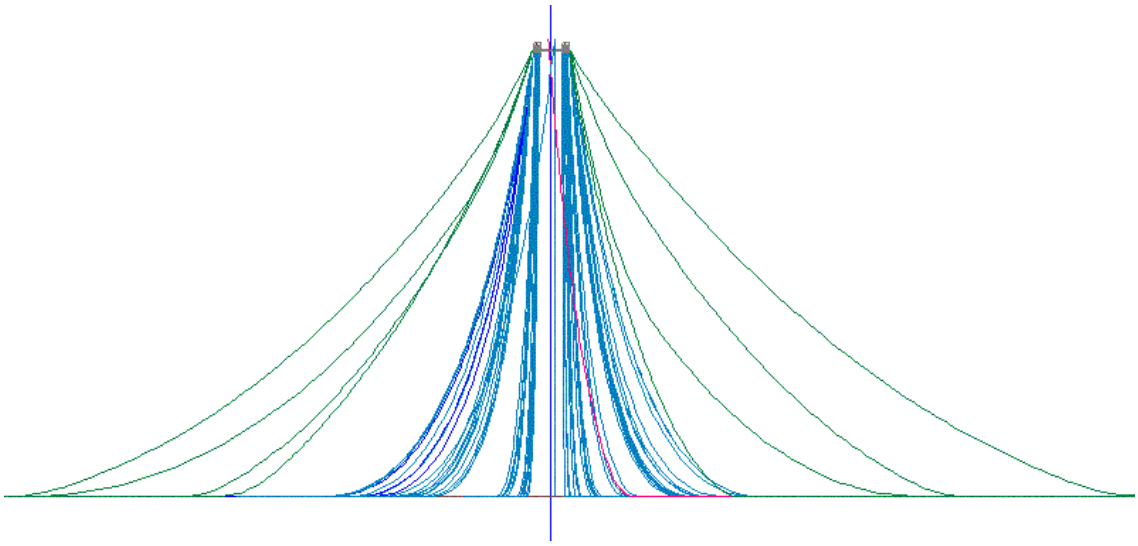


Figura 78 – Modelo Acoplado – Vista Frontal

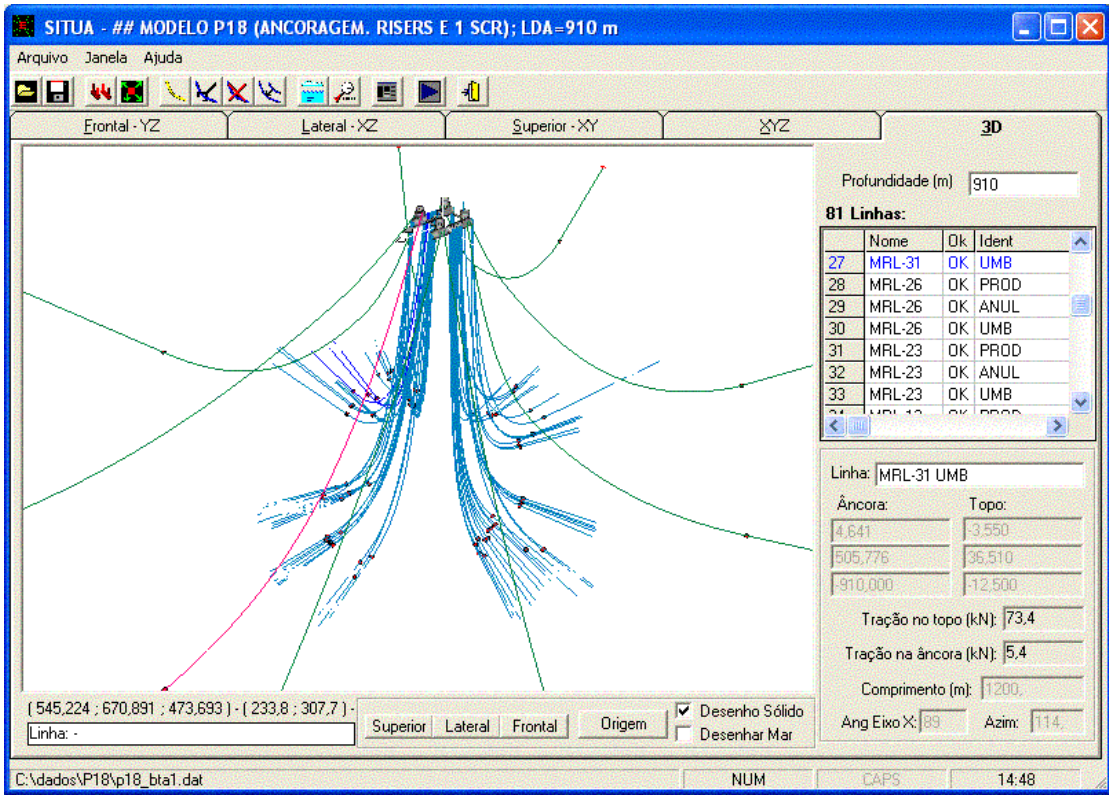


Figura 79 – Modelo Acoplado – Vista 3D

11.5.6. Subciclagem - Resultados

A subciclagem foi aplicada apenas às linhas de ancoragem, desta forma, para melhor verificação desta implementação os demais risers não foram considerados na análise. A figura 80 apresenta uma configuração deformada apenas com as linhas de ancoragem.

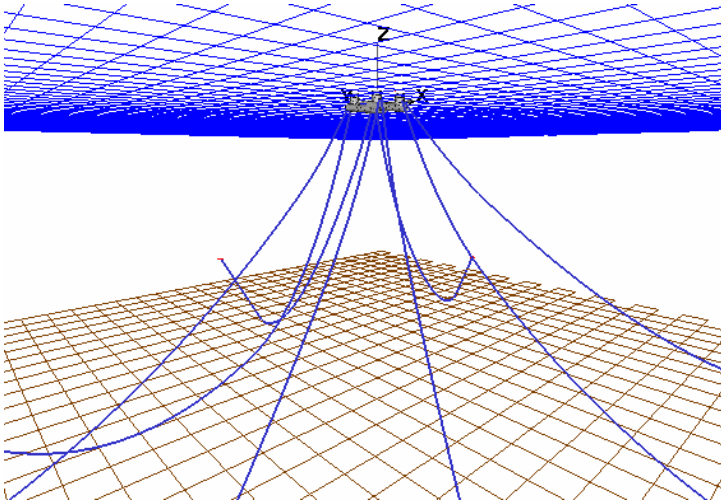


Figura 80 – Modelo da Plataforma com Linhas de Ancoragem

Para utilização da subciclagem, conforme o exemplo anterior, utilizando-se da divisão de segmentos das linhas de ancoragem devido aos diferentes tipos de material utilizados (amarra e cabo), associou-se intervalos de tempo a cada segmento.

Desta forma utilizou-se dois intervalos de tempo correspondentes a cada segmento:

- Segmentos 1 (amarra topo) e 3 (amarra fundo): $\Delta t = 0.0005$ seg;

- Segmento 2 (cabo): $\Delta t = 0.00025$ seg.

Na análise tradicional, sem subciclagem, utilizou-se o menor intervalo de tempo para todos os elementos, $\Delta t = 0.00025$ seg.

Não foi utilizado amortecimento numérico nesta análise.

A seguir, nas figuras 81 e 82, são apresentados resultados comparativos de movimentos da embarcação e tração das linhas de ancoragem, entre a versão original do programa Prosim, sem o uso da subciclagem e a versão implementada com uso da subciclagem. Conforme pode-se observar os resultados são praticamente idênticos.

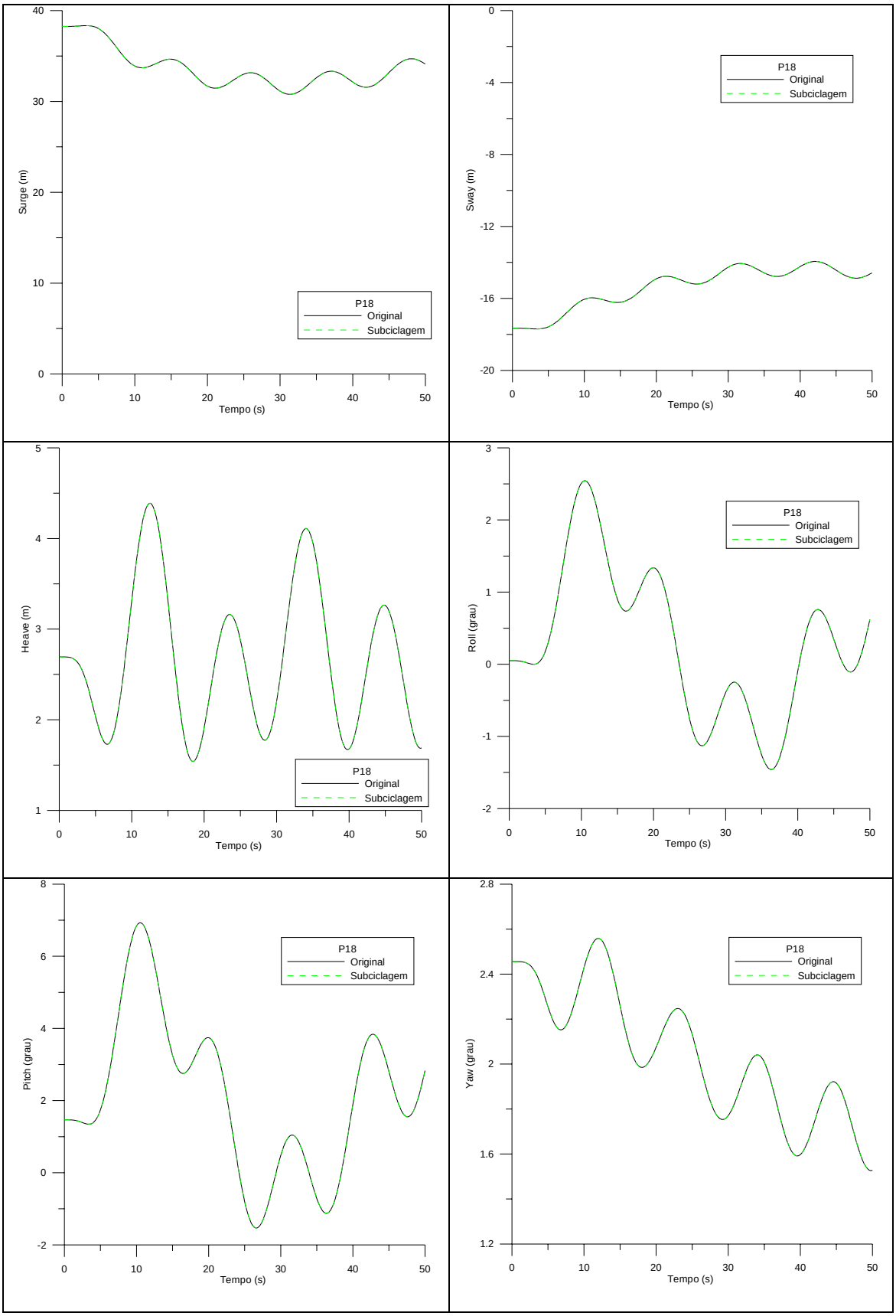
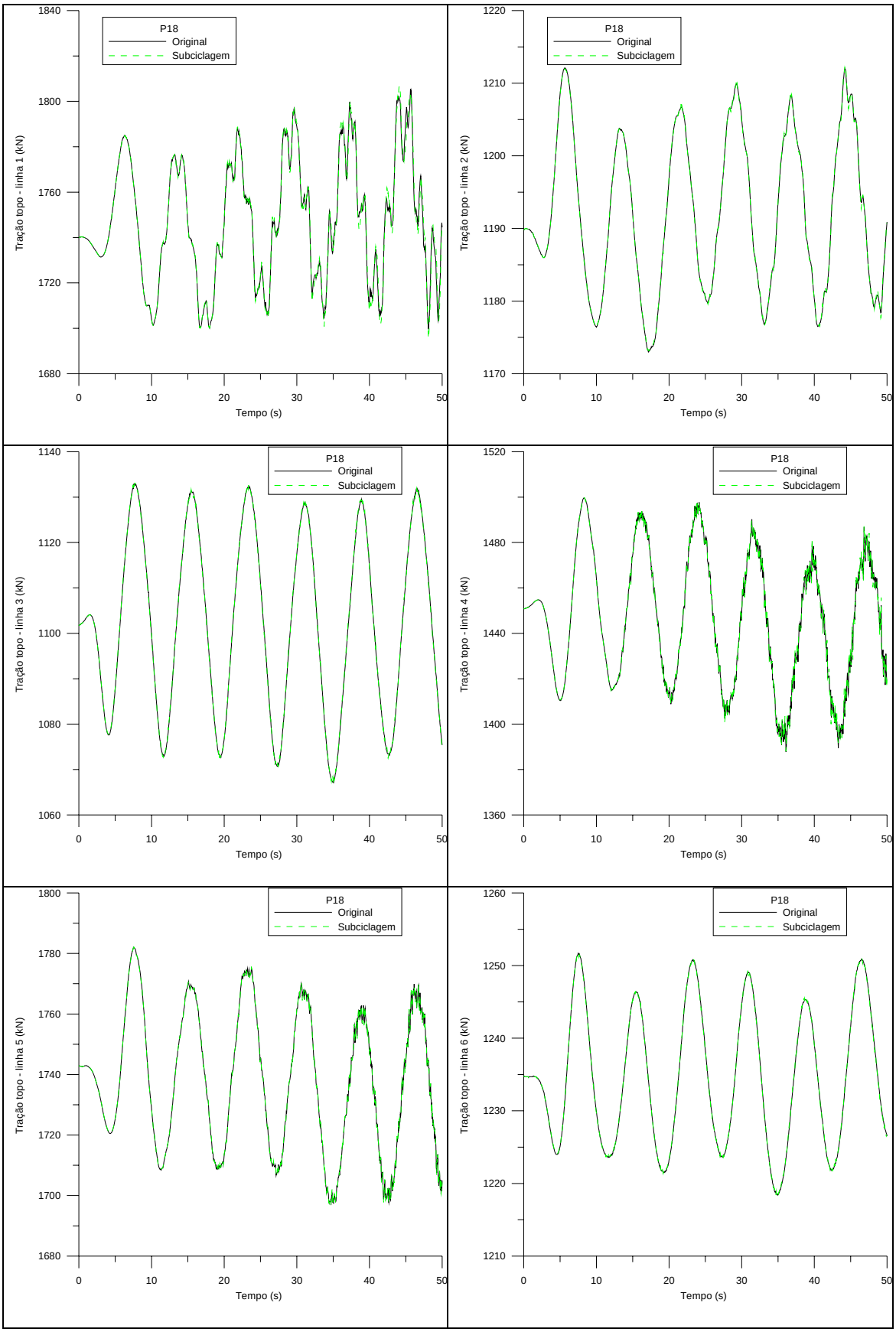


Figura 81 – Resultados comparativos – Movimentos P18


P18
Original
Subciclagem
Tração topo - linha 1 (kN)
Tração topo - linha 2 (kN)
Tração topo - linha 3 (kN)
Tração topo - linha 4 (kN)
Tração topo - linha 5 (kN)
Tração topo - linha 6 (kN)
Tempo (s)

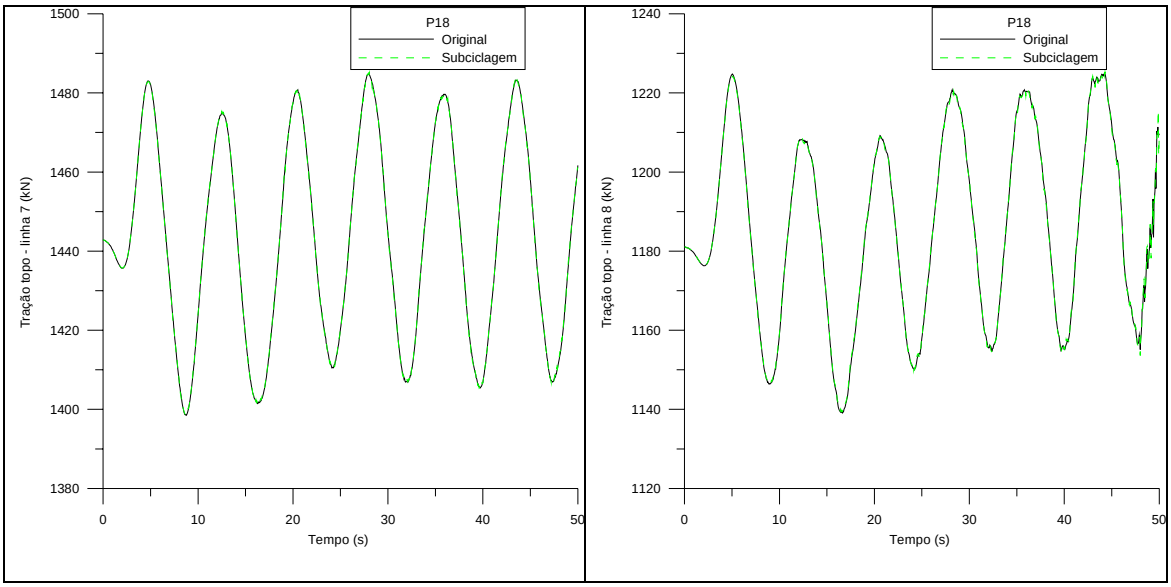


Figura 82 – Resultados comparativos – Tração topo das linhas

Com a boa precisão obtida nos resultados serão agora apresentados resultados comparativos do desempenho obtido nesta análise. A tabela 29 apresenta o tempo de CPU gasto.

Tabela 29 – Desempenho Computacional

	Tradicional	**Subciclagem**	**Economia %**
Tempo CPU (seg)	15006	9045	39.7

Com isto pode-se observar o ganho computacional significativo de 39.7% obtido neste caso.

11.5.7. PDII - Resultados

Devido à limitação de 6 processadores do cluster a aplicação do PDII no SCR com a utilização do modelo completo não traria nenhum benefício. Isto porque, com o modelo completo tem-se cerca de 3044 equações por processador, o SCR tem 1584 equações. Desta forma o SCR deveria ser analisado em conjunto com outras linhas em um determinado processador de modo que a soma das equações do SCR com estas outras linhas resultasse cerca de 3044 equações. Portanto, com o modelo completo dividido por 6 processadores, não faz sentido a utilização do PDII com o particionamento do SCR.

O PDII poderia ser utilizado no modelo completo caso o cluster tivesse, por exemplo, o número de processadores igual ao número de linhas, assim o processador que fosse analisar o SCR teria um número de equações muito superior aos demais processadores, o que justificaria a divisão do SCR em subdomínios visando a diminuição do número de equações por processador tornando assim o balanceamento de equações mais bem distribuído entre os processadores.

Com apenas 6 processadores a estratégia adotada, com o objetivo de mostrar efetivamente a utilização do PDII, foi a modelagem apenas das linhas de ancoragem e do SCR.

A seguir a figura 83 com a plataforma P18, linhas de ancoragem e SCR.

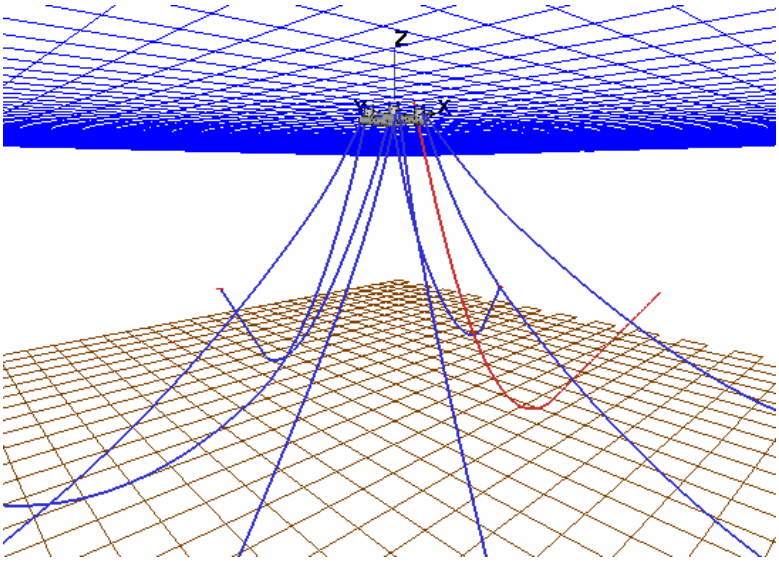


Figura 83 – Modelo Plataforma com Linhas de Ancoragem e SCR

A seguir são apresentados os principais parâmetros utilizados nesta análise:

- Intervalo de tempo do casco e linhas: 0.0025 seg;

- Tolerância de deslocamentos: 0.001

- Tolerância de forças: 0.01

- Amortecimento numérico (coeficiente αB): -0.33.

- Amortecimento proporcional à massa: não foi considerado.

- Amortecimento proporcional à rigidez: não foi considerado.

Na figura 84 são apresentados resultados comparativos de movimentos da embarcação da análise com todos os risers modelados e da análise com a modelagem das linhas de ancoragem e do SCR.

Esta comparação tem como objetivo mostrar a influência do conjunto de risers nos movimentos da plataforma e conseqüentemente a importância da análise acoplada na correta determinação dos movimentos da plataforma sob influência dos risers a ela conectados.

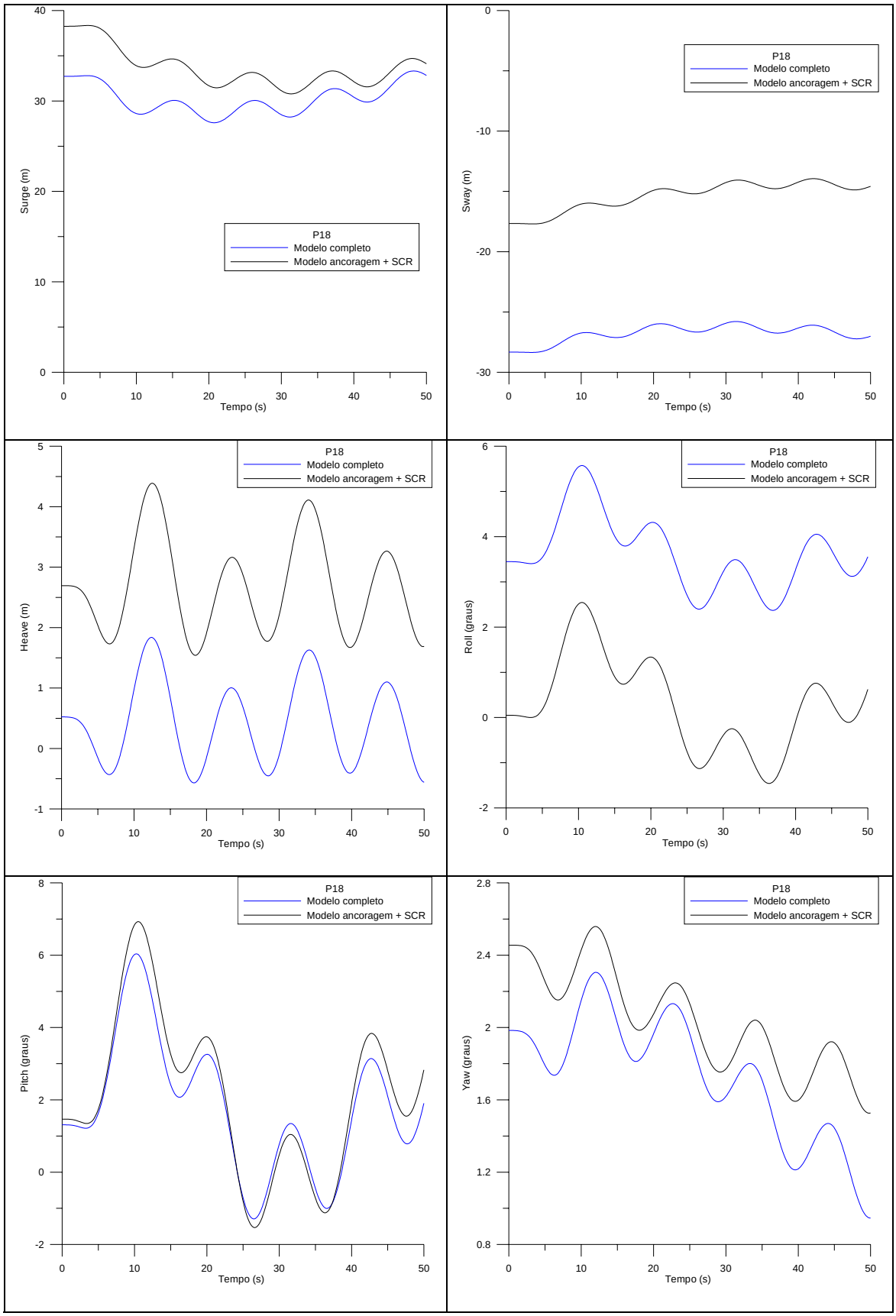


Figura 84 – Comparação de Movimentos

Para utilização do algoritmo PDII, neste exemplo, utilizou-se o seguinte procedimento, semelhante ao utilizado no exemplo da plataforma P10:

- O SCR foi particionado em 2 segmentos iguais. Sobre este riser utilizou-se o PDII em 2 processadores;
- As linhas de ancoragem foram agrupadas duas a duas e distribuídas em 4 processadores, cada processador com 2 linhas conforme o procedimento implementado por Franco em [47]. Esta divisão é adotada também na versão original do Prosim, utilizada para comparação de resultados e desempenho.

A seguir, na figura 85, o esquema de particionamento do SCR.

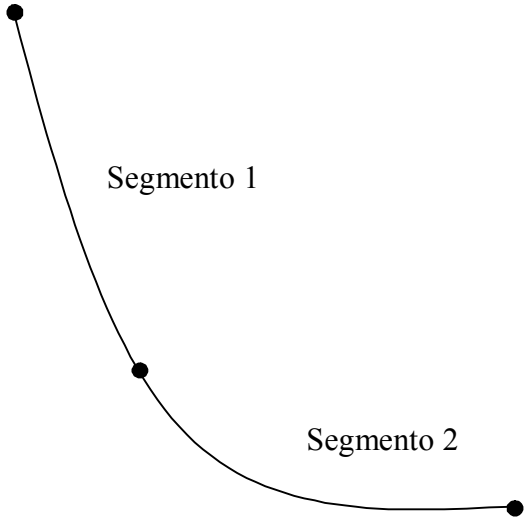


Figura 85 – Divisão do riser

Portanto as análises e comparações efetuadas adiante consideram apenas as linhas de ancoragem e SCR conectados à plataforma.

Nas figuras 86 e 87 são apresentados resultados comparativos de movimentos da embarcação e tração no topo do SCR, entre a versão original do programa Prosim e a versão implementada com uso do algoritmo PDII. Conforme pode-se observar os resultados são praticamente idênticos.

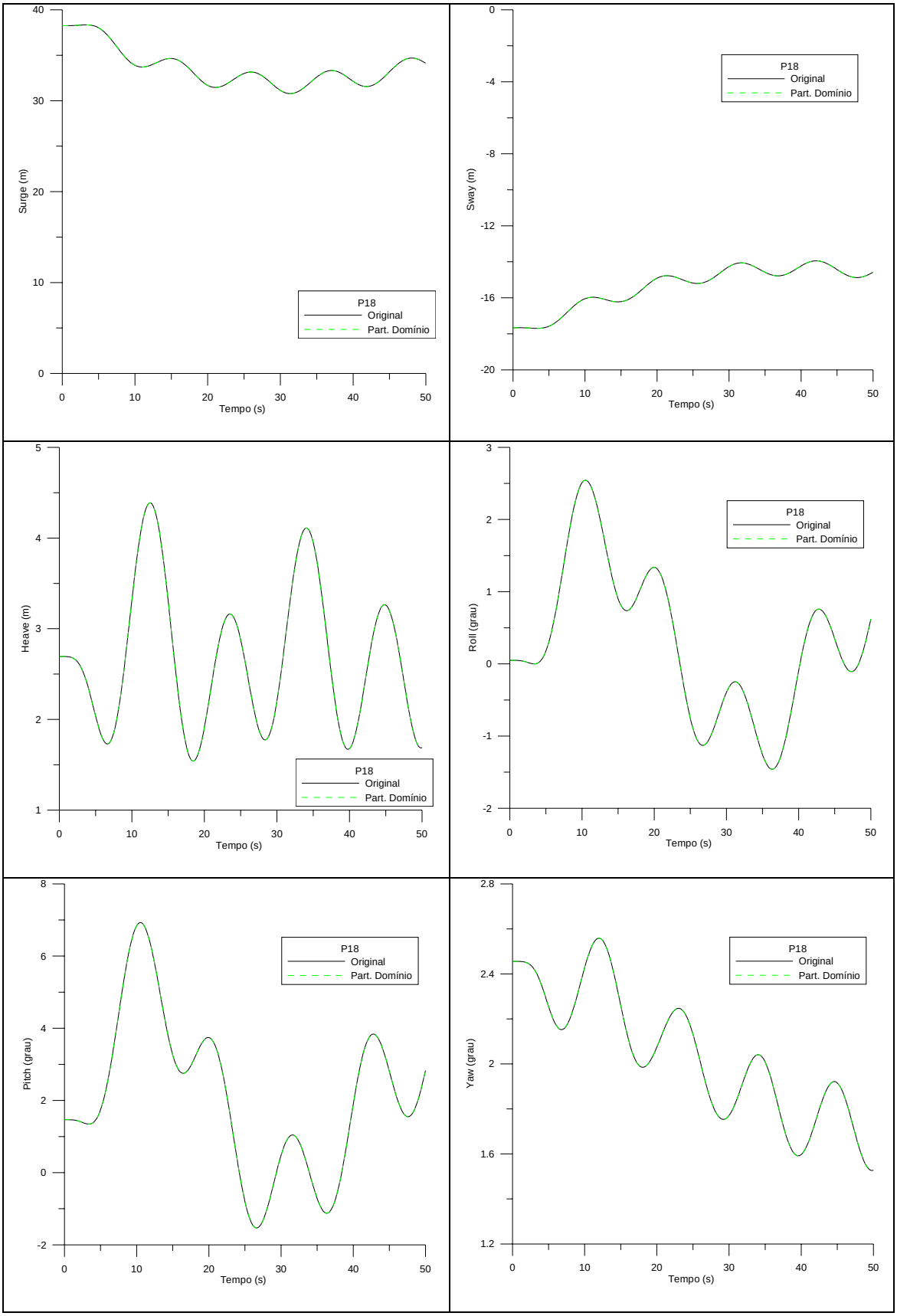


Figura 86 – Resultados comparativos – Movimentos P18

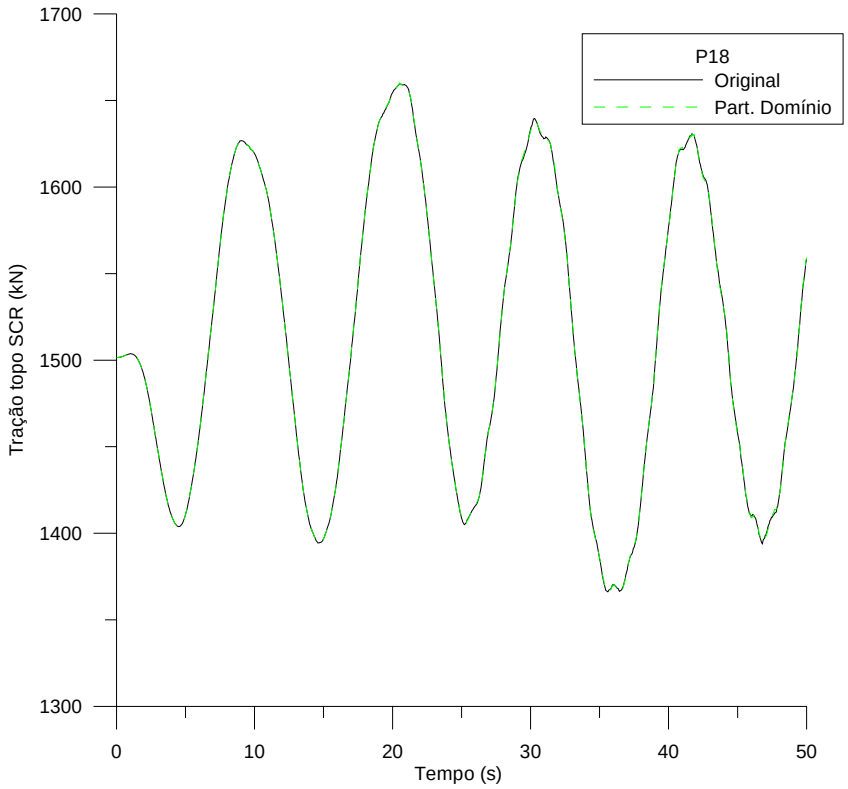


Figura 87 – Resultados comparativos – Tração topo SCR

Visto a boa precisão obtida nos resultados, serão agora apresentados resultados comparativos do desempenho obtido nesta análise. A análise tradicional apresenta o resultado da análise em que as linhas de ancoragem são distribuídas duas a duas entre os processadores 1 a 4 e o SCR é analisado seqüencialmente no processador 5. A análise com PDII apresenta o resultado da análise em que as linhas de ancoragem são distribuídas duas a duas entre os processadores 1 a 4 e o SCR é particionado em 2 segmentos e distribuído nos processadores 5 e 6. A tabela 30 apresenta o tempo de CPU gasto em cada um destes casos.

Tabela 30 – Desempenho Computacional

	Tradicional	**PDII**	**Economia %**
Tempo CPU (seg)	3886.8	2689.7	30.8

Observa-se assim o ganho significativo de 30.8% obtido na análise com o riser particionado.

Para uma melhor observação e entendimento do desempenho obtido serão apresentados resultados mais detalhados em termos de tempo de espera, tempo de comunicação e tempo de processamento de cada processador.

As figuras 88 e 89 apresentam o tempo de espera, processamento, comunicação de cada processador, além do tempo da análise correspondente à fração paralelizável e o tempo total de análise. A figura 88 utiliza o procedimento tradicional com a divisão das linhas de ancoragem pelos processadores porém sem particionamento do SCR. Neste caso o SCR é analisado pelo processador 5. A figura 89 utiliza o método PDII onde o SCR é particionado e analisado pelos processadores 5 e 6. As linhas de ancoragem são analisadas pelos processadores 1 a 4 onde cada processador analisa 2 linhas de ancoragem.

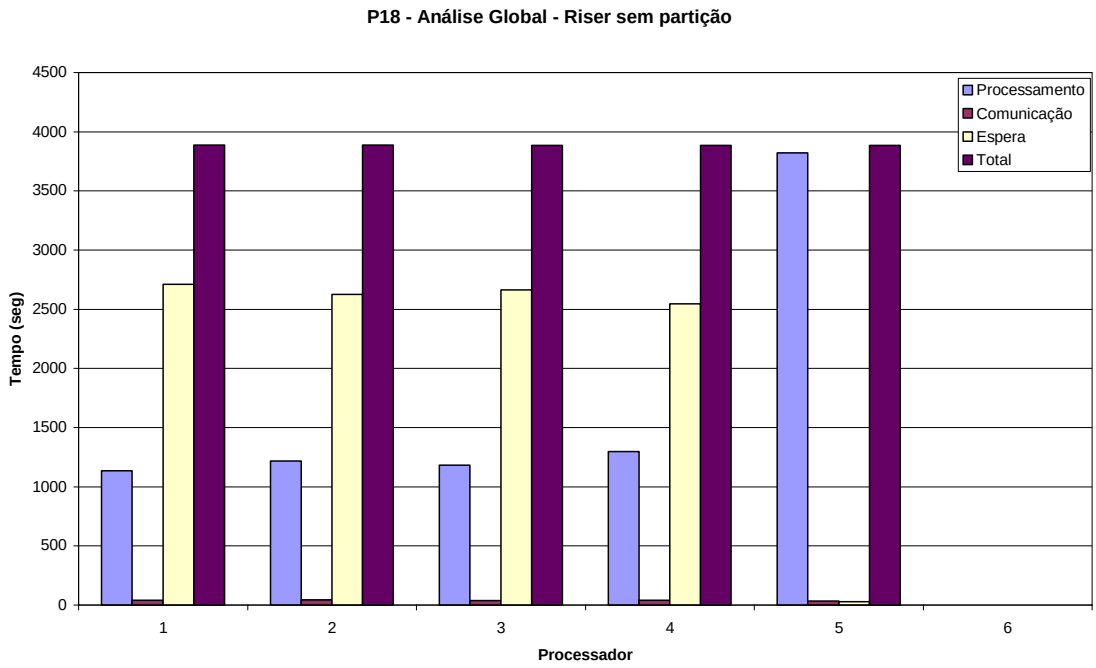


Figura 88 – Análise de Desempenho - Tradicional

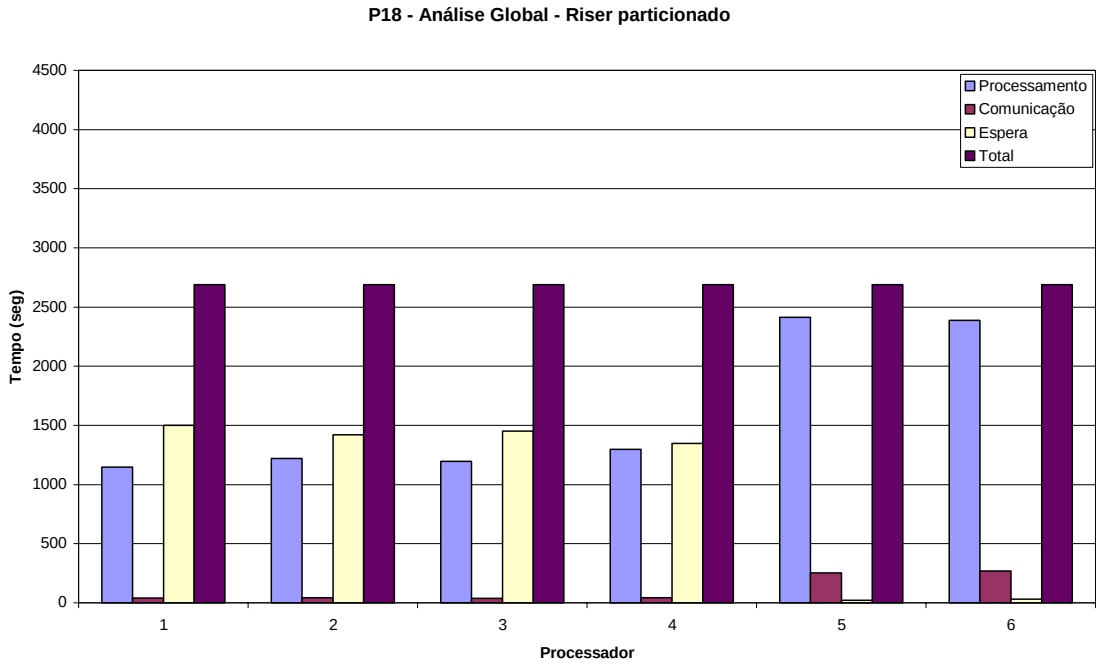


Figura 89 – Análise de Desempenho - PDII

Observa-se que, da mesma forma como ocorrido no exemplo da P10, devido ao elevado número de graus de liberdade do SCR em relação às linhas de ancoragem o tempo de espera dos processadores 1 a 4 é bem elevado. Com a utilização do PDII o tempo de espera deste processadores diminui assim como o tempo de processamento dos processadores 5 e 6 e o tempo total da análise em todos os processadores.

O tempo de comunicação entre os processadores 5 e 6 sofre um ligeiro aumento devido à comunicação necessária à compatibilidade dos nós de interface. O tempo de espera dos processadores 5 e 6 é quase nulo devido à igual divisão de graus de liberdade do riser de perfuração entre estes processadores e à comunicação total da análise se restringir a estes 2 processadores.

A figura 90 apresenta mais detalhes especificamente da análise do SCR seqüencialmente e com a utilização do PDII.

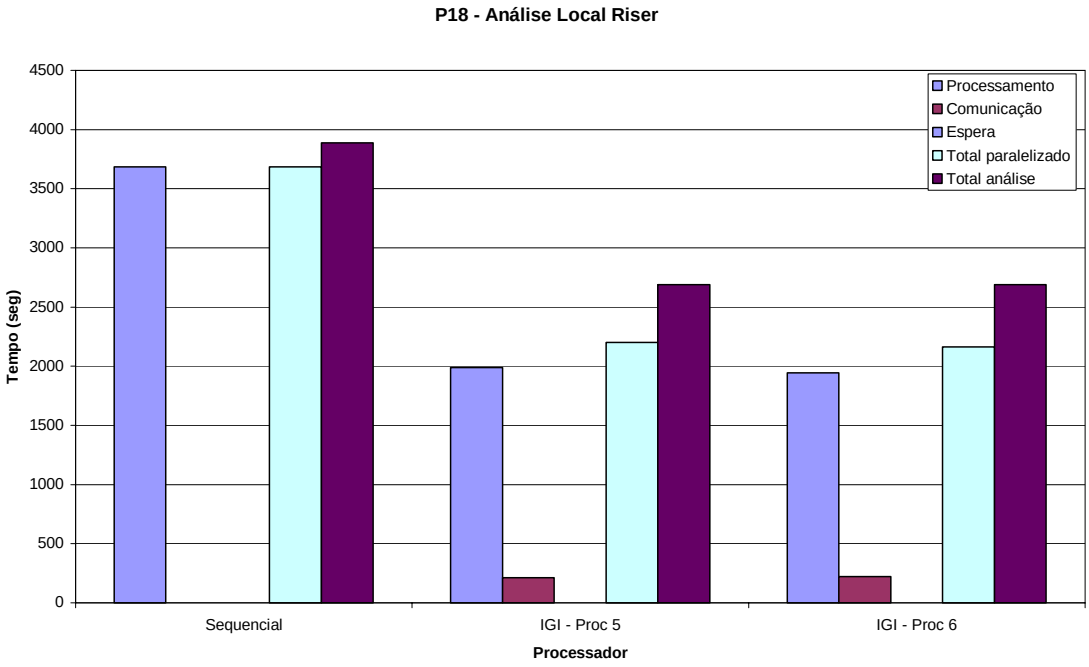


Figura 90 – Análise de Desempenho

Neste gráfico foram tomados tempo da fração paralelizada e do tempo total da análise. Pode-se observar o ganho considerável no tempo de processamento e no tempo total da análise comparando-se com a análise do riser vertical em 1 processador.

12. CONCLUSÕES

12.1. CONSIDERAÇÕES FINAIS

Os algoritmos implementados se mostraram plenamente capazes de tornar mais eficientes as análises acopladas de sistemas flutuantes. Como a análise acoplada, hoje em dia, apresenta um limitador importante na sua utilização efetiva, que é o tempo despendido durante a análise, o estudo aqui apresentado é um passo importante no sentido de tornar mais rápida e conseqüentemente mais rotineira a utilização da análise acoplada no dia-a-dia de projetos da indústria offshore. Neste contexto, de acordo com as metodologias estudadas em [13], a análise acoplada seria utilizada para determinar de forma mais precisa os movimentos da plataforma, considerando todos os efeitos de interação dinâmica não-linear entre o casco e o conjunto das linhas de ancoragem e risers.

12.1.1. Algoritmos Explícitos e Subciclagem

A primeira contribuição apresentada foi a implementação de algoritmos explícitos de integração no tempo. Implementou-se o método mais tradicionalmente utilizado, o Método das Diferenças Centrais e dois algoritmos com dissipação numérica, o algoritmo de Chung-Lee e o Método Explícito Generalizado α. Com estes algoritmos implementados abrem-se novas possibilidades de utilização da ferramenta computacional em problemas transientes onde mostre-se necessária a correta integração de modos de vibração com freqüências mais elevadas. Um exemplo deste tipo de aplicação seria na análise de movimentos de uma plataforma durante e após o transiente gerado pelo rompimento de uma linha de ancoragem, que é uma situação prevista nas normas que regem o projeto de sistemas de ancoragem.

Baseando-se na implementação dos métodos explícitos partiu-se para a implementação da subciclagem interna das linhas. Esta implementação apresentou ótimos resultados em termos de economia computacional. Nos exemplos específicos de análise acoplada o ganho apresentado foi bastante significativo se compararmos com o desempenho obtido pelas análises efetuadas tradicionalmente com método explícito de integração. Além disso o algoritmo implementado mostrou-se capaz de manter a precisão nas respostas dos casos analisados.

Quanto maior o número de linhas envolvidas numa análise acoplada espera-se que maior seja o ganho computacional ao utilizar-se a subciclagem interna das linhas – que não necessáriamente estará associada a algoritmos explícitos, mas pode ser utilizada também em conjunto com algoritmos implícitos, como será comentado mais adiante. Além disso, esta economia poderia ser ainda mais significativa com a utilização da subciclagem interna das linhas em conjunto com a subciclagem casco-linha, na qual diferentes intervalos de tempo seriam utilizados na integração do casco e dos subdomínios componentes de uma determinada linha.

A eficiência da subciclagem depende de vários fatores relativos especificamente ao caso em estudo. Um destes fatores é a diferença entre a discretização dos elementos em cada subdomínio, o que levará a uma maior ou menor relação entre os intervalos de tempo utilizados o que, conseqüentemente, terá influência direta no ganho computacional. Um outro fator importante é o número de elementos constituintes de cada subdomínio. O desempenho da subciclagem interna tende a ser melhor quanto maior o número de elementos pertencentes a subdomínios com maiores intervalos de tempo de integração associados, onde ocorrerão menos subciclos.

12.1.2. Algoritmos PDII

O principal objetivo da implementação do método PDII associado a análises acopladas foi alcançado. A precisão dos resultados obtida nas análises efetuadas foi mantida, e uma economia computacional considerável foi apresentada. Estudos detalhados do PDII, em termos de desempenho computacional, foram efetuados com o intuito de avaliação da possibilidade de maiores ganhos computacionais.

Devido à topologia particular da malha de elementos finitos empregada para discretizar as linhas de ancoragem e risers, e à natureza dinâmica não-linear do problema, o custo da resolução dos sistemas de equações algébricas não é tão preponderante, e os custos se mostram "espalhados" pelos diversas fases do procedimento de integração no tempo e do tratamento das não-linearidades. Em outras palavras, não existe um trecho de análise dinâmica, por exemplo a resolução do sistema de equações, cujo tempo de CPU mostre-se muito superior ao tempo dos demais procedimentos de cálculo.

A implementação efetuada conseguiu particionar e efetuar em paralelo a análise de praticamente todas as fases de cálculo pertencentes à análise dinâmica, resultando em uma análise de cada subdomínio praticamente de forma independente. O procedimento adotado tornou possível paralelizar mais de 90% da análise.

Estudos mais detalhados da comunicação entre processadores podem trazer ganhos ainda maiores nestas análises.. Na atual implementação os processadores se comunicam a cada iteração do ciclo de Newton-Raphson, ciclo este interno ao ciclo de intervalos de tempo. A comunicação é efetuada através de dados da interface dos subdomínios sendo enviados sempre para o processador mestre com o objetivo de compatibilização da interface. Esta comunicação pode resultar num congestionamento do fluxo de informações para o processador mestre, quanto maior for o número de subdomínios, ocasionando um possível prejuízo no desempenho final da análise.

Uma outra conclusão importante na implementação do algoritmo PDII é que, à medida que o número de partições de domínio aumenta, a tendência observada é de aumento do número de iterações Newton-Raphson, podendo também prejudicar o tempo total da análise. A recomendação é que, na prática usual de projetos, as linhas sejam particionadas em no máximo 3 subdomínios numa análise com o algoritmo PDII. Este fato foi observado em um exemplo apresentado, quando variou-se o número de processadores até o número máximo disponível no cluster.

Estudos mais detalhados podem e devem ser efetuados para a verificação do aumento do número de iterações ao aumentar-se o número de processadores. Em alguns casos, este aumento do número de iterações pode levar à não-convergência do algoritmo PDII forçando assim a utilização de menores valores de intervalo de tempo. No entanto, para a maioria dos problemas encontrados na análise acoplada de plataformas offshore, com um número elevado de linhas (podendo chegar a uma centena ou mais), e para a maioria das configurações de *clusters* disponíveis (onde não é usual encontrar mais do que poucas centenas de processadores), acredita-se que o procedimento normal será mesmo empregar não mais de que três subdomínios/processadores em cada linha.

Assim, demonstrou-se que a implementação do algoritmo PDII efetuada apresenta ganhos satisfatórios comparando-se com a análise seqüencial. O particionamento da estrutura e a implementação em paralelo do algoritmo PDII foi efetuado com sucesso em problemas dinâmicos não-lineares aplicado à análise acoplada de estruturas offshore, obtendo-se redução de custos computacionais e manutenção da precisão dos resultados.

12.2. PROPOSTAS PARA DESENVOLVIMENTOS FUTUROS

Os desenvolvimentos apresentados neste trabalho consistiram em um primeiro e importante passo dado na direção de otimização de análises acopladas. Outros desenvolvimentos devem ser estudados no intuito da busca de desempenho cada vez maior destas análises. A seguir são apresentadas algumas propostas visando este objetivo.

12.2.1. Subciclagem

A subciclagem interna das linhas mostra-se promissora também na utilização com algoritmos implícitos de integração, apesar de o intervalo de tempo utilizado em métodos implícitos não estar limitado ao intervalo de tempo crítico. Isto significa que, mesmo com a possibilidade de utilização de maiores intervalos de tempo em algoritmos implícitos a variação de discretização de uma linha permite que a subciclagem interna possa ser utilizada com a conseqüente possibilidade de obtenção de economia computacional.

A atual implementação utiliza razões inteiras entre os intervalos de tempo dos subdomínios. Devido a utilização de contadores de tempo na atualização do subdomínio, a utilização de razões não-inteiras entre os intervalos de tempo apresenta-se como uma possibilidade a ser avaliada.

No presente trabalho utilizou-se velocidade constante nos nós de interface durante os subciclos. Novos estudos podem ser efetuados com a utilização de aceleração constante nos nós de interface.

12.2.2. Algoritmos PDII

Em relação aos algoritmos PDII e à implementação em paralelo, a proposta para um próximo trabalho estaria concentrada inicialmente na comunicação de dados. Esta comunicação dos nós de interface poderia tornar-se mais eficiente caso os dados, ao invés de serem enviados todos para o processador mestre, conforme implementado, fossem enviados apenas aos processadores que partilhassem dos subdomínios com mesma interface, o que melhoraria ainda mais o desempenho do PDII.

Uma outra proposta a ser analisada estaria concentrada na compatibilidade dos nós de interface. A regra de ponderação da massa é utilizado neste trabalho. Outros métodos para compatibilidade dos nós de interface podem ser avaliados como, por exemplo, a utilização da matriz de rigidez nesta compatibilidade, ou alguma outra proposta de regra de ponderação.

Além disso, a compatibilidade foi efetuada através de deslocamentos na interface, poderiam ser utilizadas velocidades ou acelerações nesta compatibilidade. Estudos neste sentido também podem ser efetuados.

Como a solução evolui ao longo de um ciclo iterativo, no caso o ciclo relativo ao uso do algoritmo de Newton-Raphson, poderia-se estudar uma variação da tolerância utilizada para verificação da convergência ao longo dos ciclos. Um estudo neste sentido foi apresentado em [12]. Este estudo pode ser reavaliado para utilização em conjunto com o algoritmo PDII.

12.2.3. Combinação Subciclagem / Algoritmos PDII

As duas implementações efetuadas também poderiam ser utilizadas em conjunto, visto que a subciclagem interna foi implementada seqüencialmente como otimização para algoritmos explícitos e o algoritmo PDII foi implementado como otimização para algoritmos implícitos de integração no tempo. A subciclagem interna poderia ter sua aplicabilidade associada aos computadores com arquitetura em paralelo. Como a linha é dividida em segmentos em que, cada segmento, recebe um determinado valor de intervalo de tempo, estes segmentos poderiam ser integrados independente e concorrentemente com a utilização do algoritmo de partição de Domínio (PDII) implementado.

Uma outra proposta consiste na utilização de uma estratégia adaptativa para determinação automática do intervalo de tempo ao longo da análise dinâmica em conjunto com o algoritmo PDII e a subciclagem interna. Observa-se que os períodos naturais da estrutura podem variar ao longo da análise dinâmica não-linear devido à variação da configuração da estrutura ao longo da simulação, o que pode resultar na utilização de uma estratégia em que o intervalo de tempo varia automaticamente ao longo da análise. Assim, à medida que o intervalo de tempo varia alguns segmentos

podem ter mais ou menos subciclos ao longo da integração no tempo e os subdomínios, utilizados no método PDII, podem ser rearranjados ao longo da simulação.

Outra possibilidade de implementação é a utilização de uma estratégia adaptativa de refinamento da malha. Devido à dinâmica não-linear utilizada na análise de uma linha, trechos que antes poderiam ter um refinamento mais grosseiro passam a necessitar de um maior refinamento da malha de elementos finitos ou vice-versa. Assim o intervalo de tempo, calculado de acordo com o período natural da malha, também varia acarretando uma variação de subciclos utilizados na subciclagem interna e um possível rearranjo de subdomínios utilizados na partição do domínio ao longo da simulação.

Por fim, uma combinação da estratégia adaptativa de refinamento da malha em conjunto com a estratégia adaptativa de determinação automática de intervalo de tempo e a utilização da subciclagem interna das linhas e subciclagem casco-linha, além do algoritmo PDII, resultaria num algoritmo onde, se não todas, mas grande parte das possibilidades de otimização estariam implementadas e colaborando para otimização da análise acoplada.

De qualquer forma as opções de implementações citadas devem ser estudadas e analisadas com cuidado para avaliação das possibilidades onde espera-se a obtenção de ganho computacional.

13. REFERÊNCIAS

[1] JACOB, B.P, *Programa PROSIM: Simulação Numérica do Comportamento de Unidades Flutuantes Ancoradas, Versão 2.2ª – Manual de Entrada de Dados*, COPPE/UFRJ, Programa de Engenharia Civil, Rio de Janeiro, 2001.

[2] CORREA, F.N., *Aplicação de Metodologia Híbridas em Estudos Paramétricos sob o Comportamento de Sistemas Offshore*, Tese de M. Sc. , COPPE/UFRJ, Programa de Engenharia Civil, Março de 2003.

[3] MEIROVICH, L., *Methods of Analytical Dynamics*, McGraw-Hill, New York, 1970.

[4] ZIENKIEWICZ, O. C., TAYLOR, R. L., *The Finite Element Method*, 5 ed. Oxford, Butterworth-Heinemann, 2000.

[5] BATHE, K-J., *Finite Element Procedures,* New Jersey, Prentice-Hall, 1996.

[6] HUGHES, T. J. R. , *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[7] KAYSER JUNIOR, D. L., *Análise Dinâmica de Linhas Flexíveis com Elemento de Pórtico Não Linear Geométrico Híbrido*. Tese de M.Sc., COPPE/UFRJ – Programa de Engenharia Civil, Rio de Janeiro, 2003.

[8] CRISFIELD, M. A., "A Consistent Co-Rotational Formulation for Non-Linear, Three-Dimensional, Beam-Elements", *Computer Methods in Applied Mechanics and Engineering*, v. 81, pp. 131-150, 1990.

[9] MATHISEN, K.M., *Large Displacement Analysis of Flexible and Rigid Systems Considering Displacement-Dependent Loads and Nonlinear Constraints*. Division of Structural Engineering/The Norwegian Institute of Technology, Trondheim, Norway, 1990.

[10] MOURELLE, M. M., *Análise Dinâmica de Sistemas Estruturais Constituídos por Linhas Marítimas,* Tese de D.Sc., COPPE/UFRJ, Programa de Engenharia Civil, 1993.

[11] JACOB, B. P., *Notas de Aula da Disciplina "Dinâmica de Sistemas Discretos"*. Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, 1998.

[12] RODRIGUES, M. V., *Uma Estratégia Semi-Implícita para a Solução de problemas de Dinâmica Estrutural,* Tese de M.Sc., COPPE/UFRJ, Programa de Engenharia Civil, 1998.

[13] SENRA, S. F., *Metodologias de Análise e Projeto Integrado de Sistemas Flutuantes para exploração de Petróleo Offshore*. Tese de D.Sc., COPPE/UFRJ, Programa de Engenharia Civil, Rio de Janeiro, 2004.

[15] WHELLER, J. D., *Method for Calculating Forces Produced for Wave-Body Interactions*, Massachusetts Institute of Tecnology Department of Ocean Engeneering-MIT, 1991.

[14] CHAKRABARTI, S. K, *Hydrodynamics of Offshore Structures*, Computational Mechanics Publications Southampton Boston, 1987.

[16] HOOFT, J. P., *Advanced Dynamics of Marine Structures*, John Wiley & Sons, USA, 1982.

[17] WAMIT, *A Radiation-Difraction Panel Program for Wave-Body Interactions*. Version 5.3, User Manual, Department of Ocean Engineering Massachusettes Institute of Technology, 1995.

[18] PAULING, J. R., TDSIM6 – *Time Domain Platform Motion Simulation with Six Degrees of Freedom*: Theory and User Guide, 1992.

[19] FALTINSEN, O. M., *Sea Loads on Ships and Offshore Structures*, Cambridge University Press, Cambridge, UK, 1990.

[20] DNV/POSMOOR, *Rules for Classification of Mobile Offshore Units,* Part 6 chapter 2: Position Mooring *(POSMOOR).* Det Norske Veritas, 1989.

[21] CLARK, P.J., MALENICA, S., MOLIN, B., "An Heuristic Approach to Wave Drift Damping". *J. Applied Ocean Research*, vol.15, n^{o} 1, 1993.

[22] API RP 2A, *Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms.* Working Stress Design, RP 2A, Twentieth Edition, American Petroleum Institute, July 1, 1993.

[23] CHUNG, J., LEE. J. M., "A New Family of Explicit Time Integration Methods for Linear and Non-Linear Strucutural Dynamics", *International Journal for Numerical Methods in Engineering*, vol 37 pp. 3961-3976, 1994.

[24] HULBERT, G. M., CHUNG, J., "Explicit Time Integration Algorithms for Strucutural Dynamics with optimal Numerical Dissipation", *Computer Methods Applied Mechanics Engineering*, vol 137 pp. 175-188, 1996.

[25] SILVEIRA, E. S. S., *Análise dinâmica de Linhas de Ancoragem com Adaptação no Tempo e Subciclagem*. Tese de D.Sc., PUC, Departamento de Engenharia Civil, Rio de Janeiro, 2001.

[26] COOK, R.D., MALKUS, D.S. e PLESHA M.E., *Concepts and Applications of Finite Element Analysis*, Third edition, John Wiley & Sons, 1989.

[27] ADAMS, D.D. e WOOD, W.L., "Comparison of Hilber- Hughes-Taylor and Bossak α-methods for the Numerical Integration of Vibration Equations", *International Journal for Numerical Methods in Engineering*, vol 19 n. 5 pp. 765-771, 1983.

[28] WOOD, W.L., BOSSAK, M. e ZIENKIEWICZ, O.C., "An Alpha Modification of Newmark's Method", *International Journal for Numerical Methods in Engineering*, vol 15 pp. 1562-1566, 1980.

[29] HILBER, H. M., HUGHES, T. J. R. e TAYLOR, R. L., "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics", *Earthquake Engineering and Structural Dynamics*, vol. 5 pp. 283-292, 1977

[30] HILBER, H. M. e HUGHES, T. J. R., "Collocation, Dissipation and 'Overshoot' for Time Integration Schemes in Structural Dynamics", *Earthquake Engineering and Structural Dynamics*, vol. 6 pp. 99-117, 1978.

[31] JACOB, B.P. e EBECKEN, N.F.F., "An Optimized Implementation of the Newmark/Newton-Raphson Algorithm for the Time Integration of Nonlinear Problems", *Communications in Numerical Methods in Engineering*, vol. 10 pp. 983-992, John Wiley & Sons, UK/USA, 1994.

[32] DANIEL, W. J. T., "Analysis and Implementation of a New Constant Acceleration Subcycling algorithm"*, International Journal for Numerical Methods in Engineering,* vol. 40, 2841-2855, 1997.

[33] SMOLINSKI, P., "Subcycling Integration with Non-Integer Time Steps fos Structural Dynamics Problems"*, Computers & Structures* vol. 59, no. 2, 273-281, 1996.

[34] BELYTSCHKO, T. and LU, Y. Y., "Explicit Multi-Time Step Integration for first and Second Order Finite Element Semidiscretization", *Computer Methods in Applied Mechanics and Engineering* vol. 108, 353-383, 1993.

[35] JACOB, B.P. , *Estratégias Computacionais para Análise Não-Linear Dinâmica de Estruturas Complacentes para Águas Profundas*, Tese de D.Sc., COPPE/UFRJ, Programa de Engenharia Civil, 1990.

[36] HUGHES, T. J. e BELYTSCHKO, T., "A précis of developments in computational methods for transient analysis". *J. appl. Mech*., ASME 50, pp. 1033-1041, 1983.

[37] BELYTSCHKO, T., YEN H. J. e MULLEN, R., "Mixed methods for time integrations". *Comput. Meth. Appl. Mech. Engng* 17/18, pp. 259-275, 1979.

[38] LIU, W. K. e BELYTSCHKO, T., "Mixed time implicit-explicit finite element for transient analysis". *Comput. Struct.* 15, pp. 445-450, 1982.

[39] SOTELINO, E. D., "Stability of the GI concurrent algorithms"*, International Journal for Engineering Analysis and Design,* 1, 1-12, 1994.

[40] MODAK, S. e SOTELINO, E. D., "The Iterative Group Implicit algorithm for parallel transient finite element analysis"*, International Journal for Numerical Methods in Engineering,* 47, 869-885, 2000.

[41] ORTIZ, M. e NOUR-OMID, B., "Unconditionally Stable Concurrent Procedures for Transient Finite Element Analysis"*, Comp. Meth. Appl. Mech. Engr.,* Vol. 58, pp. 151-174, 1986.

[42] HAJJAR, J. F. e ABEL, J. F., "On the accuracy of some domain-by-domain algorithms for parallel processing of dynamics"*, International Journal for Numerical Methods in Engineering,* 28, 1855-1874, 1989.

[43] FREITAS, E. L. e GEYER, C., *Uma comparação entre os modelos de Message Passing MPI e PVM.* UFRGS. Março 2003. Disponível em: www.inf.ufrgs.br/procpar/disc/cmp134/trabs/T2/981/mpi.html. Acesso: set. 2003.

[44] MESSAGE PASSING INTERFACE FORUM. *MPI: A Message-Passing Interface Standard.* 1995. Disponível em: http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html. Acesso: jun. 2003.

[45] PACHECO, P. S., MING, W. C., *Introduction to Message Passing Programming: MPI User Guide in FORTRAN*. 1997. Disponível em: http://www.hku.hk/cc/sp2/training.html. Acesso em: jan. 2003.

[46] RODRIGUES, M. V., *Desenvolvimento de Ferramentas para Análise Acoplada de Linhas e Unidades Flutuantes*. Exame de Qualificação ao Doutorado, COPPE/UFRJ - Programa de Engenharia Civil, Rio de Janeiro, 2002.

[47] FRANCO, L. D., *Implementação Computacional em Ambiente Paralelo de memória Distribuída para Análise Acoplada de sistemas Offshore*. Tese de M.Sc., COPPE/UFRJ, Programa de Engenharia Civil, Rio de Janeiro, 2004.

[48] TECHNICAL REPORT – *Mobile Offshore Drilling Unit Petrobras X – Upgrading for 1200m Water Depth – Mooring Analysis* - RL-2311.00-1110-962-PPC-001.

[49] TECHNICAL REPORT – *Mobile Offshore Drilling Unit Petrobras X – Upgrading for 1200m Water Depth – Stability Analysis* - RL-2311.00-1110-960-PPC-001.