UMA ESTRATÉGIA SEMI-IMPLÍCITA PARA A SOLUÇÃO DE PROBLEMAS DE DINÂMICA ESTRUTURAL

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 MESTRE EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

Prof. Breno Pinheiro Jacob, D.Sc.

Prof. Álvaro L. G. A. Coutinho, D.Sc.

Eng. Isaías Quaresma Masetti, D.Sc.

Prof. Hélio J. C. Barbosa, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

FEVEREIRO DE 1998

RODRIGUES, MARCOS VINÍCIUS

Uma Estratégia Semi-Implícita para a Solução de Problemas de Dinâmica Estrutural [Rio de Janeiro] 1998

XI, 99 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Civil, 1998)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Dinâmica Estrutural
2. Elementos Finitos

I. COPPE/UFRJ II. Título (série)

Aos meus pais Fabio e Elizabete
Aos meus irmãos Diogo e Raquel

AGRADECIMENTOS

A Deus, por tudo.

Ao Prof. Breno, por sua orientação, pela motivação e pela paciência.

Ao Prof. Elson Magalhães Toledo, pelo incentivo inicial.

Ao Eng. Isaías Quaresma Masetti, pela chance e apoio concedidos.

Ao CNPq, pelo suporte financeiro a este trabalho.

À minha família, pelo apoio fundamental.

Aos colegas da COPPE e do CENPES: Elaine, Ana Maria, Cristiane, André, Luciana, Ana Cláudia, Luís Antônio, Sérgio, Eduardo, Iuri, Silvestre, Vardaro, Fábio, Mauro, Ronaldo, Felipe e a todos que, se fosse enumerá-los não haveriam páginas suficientes.

Aos amigos Breno Ferreira Grossi, Sérgio Moraes Cardoso, Adriano de Andrade Barbosa e Rommerson Fábio Soares, que, além de dividirem comigo as contas a pagar, tornaram estes anos de trabalho menos cansativos.

À Érika Maria Carvalho Silva, Stefânia Defilippo Rocha e Stael Ferreira Senra pela amizade.

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

UMA ESTRATÉGIA SEMI-IMPLÍCITA PARA A SOLUÇÃO DE PROBLEMAS DE DINÂMICA ESTRUTURAL

Marcos Vinícius Rodrigues

Fevereiro/1998

Orientador: Breno Pinheiro Jacob

Programa: Engenharia Civil

Esta tese apresenta um estudo onde o principal objetivo é a economia de tempo de CPU na análise dinâmica não-linear de estruturas. Este tipo de análise pode ser aplicado a estruturas que apresentam grandes deslocamentos e incidência de carregamentos variáveis com o tempo. As aplicações aqui apresentadas estão voltadas para estruturas off-shore.

A estratégia apresentada consiste em reunir características vantajosas de métodos de integração implícitos (maiores valores de intervalo de tempo) e explícitos (menor custo por instante de tempo), surgindo daí a idéia do método semi-implícito. Para tal utilizou-se um "solver" iterativo (Método dos Gradientes Conjugado Pré-Condicionado) com tolerância variável para resolver o sistema de equações proveniente do tratamento numérico dado ao problema em questão.

Além disso, propõe-se um esquema auto-adaptativo onde, de acordo com a aplicação em questão, o próprio algoritmo decide se é mais vantajoso utilizar o esquema "semi-implícito" proposto ou o esquema implícito tradicional baseado em "solver" direto.

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

A SEMI-IMPLICIT STRATEGY FOR SOLUTION OF STRUCTURAL DYNAMICS PROBLEMS

Marcos Vinícius Rodrigues

February/1998

Advisor: Breno Pinheiro Jacob

Department: Civil Engineering

This thesis presents a study where the principal objective is to reduce CPU costs in non-linear dynamics analysis of structures. This kind of analysis can be applied to structures presenting large displacements and time varying load. The applications here presented are oriented to off-shore structures.

The strategy presented consists in combining the advantages of implicit (larger time-step values) and explicit (smaller cost by time-step) integration methods. The resulting scheme is a semi-implicit method based on an iterative solver (Pre-Conditioned Conjugate Gradient Method) with variable tolerance to solve the effective systems of equations deriving from the numerical treatment of the dynamic equilibrium equation.

It is also proposed an auto-adptive scheme where, for any given application, the algorithm decides when the semi-implicit scheme is more advantageous than the traditional implicit scheme based on a direct solver.

ÍNDICE

1 - INTRODUÇÃO.....1

1.1 - Objetivos e Motivação.....1

1.2 - Metodologia.....3

1.3 - Organização do trabalho.....4

2 - FORMULAÇÃO BÁSICA.....6

2.1 - Solução numérica de problemas dinâmicos lineares.....6

2.2 - Seleção de um algoritmo de integração.....9

2.2.1 - Propriedades intrínsecas de um algoritmo.....10

2.2.2 - Fatores externos ao algoritmo.....14

2.3 - Problemas não-lineares..... 16

2.3.1 - Tratamento do problema n-l em algoritmos implícitos.....16

2.3.2 - Formulação incremental-iterativa.....18

3 - ESQUEMAS HÍBRIDOS DE INTEGRAÇÃO NO TEMPO..... 20

3.1 - Histórico.....20

3.1.1 - Técnicas de partição do domínio.....20

3.1.2 - Métodos de partição do operador (Operator Splitting)..... 21

3.2 - O Método dos Gradientes Conjugados Pré-Condicionados (MGCP)....23

3.3 - O esquema "semi-implícito".....24

3.4 - Um esquema auto-adaptativo para resolução do sistema de equações.. 35

4 - APLICAÇÕES NUMÉRICAS.....37

4.1 - Torre estaiada.....38

4.1.1 - Análise com esquema implícito tradicional..... 42

4.1.2 - Análise com esquema semi-implícito proposto..... 44

4.1.2.1 - Critério de Papadrakakis..... 44

4.1.2.2 - Critério de Dembo / Steinhaug..... 48

4.1.2.3 - Critério de Dembo / Steinhaug modificado..... 52

4.1.2.4 - Critério de Nour-Omid (t = 1.5)..... 55

4.1.2.5 - Critério de Nour-Omid modificado (t=1)..... 58

4.1.2.6 - Critério de Nour-Omid modificado (t=0.7)..... 61

4.1.3 - Comparação de desempenho..... 64

4.1.4 - Análise com esquema auto-adaptativo..... 66

4.2 - Conjunto de risers flexíveis..... 70

4.2.1- Análise com esquema implícito tradicional 73

4.2.2- Análise com esquema semi-implícito proposto 75

4.2.2.1- Critério de Papadrakakis..... 75

4.2.2.2 - Critério de Nour-Omid modificado (t = 0.7)..... 78

4.2.3 - Comparação de desempenho..... 81

4.2.4 - Análise com esquema auto-adaptativo..... 82

5 - CONCLUSÕES..... 83

REFERÊNCIAS BIBLIOGRÁFICAS..... 86

LISTA DE SÍMBOLOS

SÍMBOLOS LATINOS:

a	- vetor de acelerações discretizado no tempo
$\hat{A}$	- matriz efetiva
$\hat{b}$	- vetor de cargas efetivo
C	- matriz de amortecimento
c	- constante de erro
d	- vetor de deslocamentos discretizado no tempo
$F(t)$	- vetor de forças externas
F_I	- forças inerciais
F_{INT}	- forças internas
F_{EXT}	- forças externas
$\|g(x_k)\|$	- norma do vetor de resíduos efetivos
h	- espaçamento nodal da malha
$k(\hat{A})$	- razão espectral
k	- contador de iterações Newton-Raphson
K	- matriz de rigidez
K_T	- matriz de rigidez tangente
m	- ordem de precisão ou taxa de convergência
M	- matriz de massa
p_i	- parâmetro MGCP
Q	- funcional quadrático para resolução do sist. de equações

r_i - parâmetro MGCP

R - pré-condicionador no Método dos Gradientes Conjugados

R(u) - esforços internos

t - expoente utilizado nos critérios de variação de tolerância

T_n - menor período natural da malha de elementos finitos

u(t) - vetor de deslocamentos

$\dot{u}(t)$ - vetor de velocidades

$\ddot{u}(t)$ - vetor de acelerações

v - vetor de velocidades discretizado no tempo

x - variável do sistema de equações

SÍMBOLOS GREGOS:

α - parâmetro para controle da dissipação

α_i - parâmetro MGCP

β - parâmetro da família de algoritmos de Newmark

β_i - parâmetro MGCP

Δt - intervalo de tempo

Δt_{cr} - intervalo de tempo crítico

Δd_i - parâmetro MGCP

Δd - incremento de deslocamento

γ - parâmetro da família de algoritmos de Newmark

η_k - tolerância do ciclo iterativo MGCP

λ_1 - menor autovalor do espectro

λ_n - maior autovalor do espectro

$\rho(A)$ - raio espectral da matriz de amplificação

$\tau(t)$ - erro de truncamento local

CAPÍTULO 1

INTRODUÇÃO

1.1 - Objetivos e Motivação

O objetivo principal desta tese é apresentar um esquema para a resolução de problemas dinâmicos estruturais visando a obtenção de melhor eficiência computacional. Tal esquema será aplicado a problemas inerciais não-lineares, onde a resposta dinâmica é dominada pelos modos de vibração de frequências mais baixas, com consideração particular para estruturas off-shore tais como torres complacentes, unidades flutuantes ancoradas e suas linhas de ancoragem, risers flexíveis, etc.

Para ilustrar a motivação para o desenvolvimento deste esquema, vamos considerar o caso da indústria do petróleo off-shore no Brasil [1]. As atividades iniciais de exploração de petróleo em águas marinhas eram feitas próximas à costa, em profundidades não superiores a 100 m. Para tal exploração eram utilizadas estruturas metálicas fixas (as "jaquetas") sobre as quais as plataformas se apoiavam. Tais estruturas quase não possuíam efeitos dinâmicos e não-lineares, de modo que a análise estrutural destas estruturas era bastante simples.

Recentemente, com a verificação da existência de reservatórios em águas cada vez mais profundas, tornou-se necessário o desenvolvimento de novas concepções estruturais. Com isso surgiu o conceito de *estruturas complacentes*, com períodos naturais de vibração superiores aos períodos das cargas de projeto, e que podem apresentar grandes deslocamentos sob ação de cargas ambientais. Dentre as estruturas complacentes podemos citar: Torre estaiada, Plataformas semi-submersível, Plataforma

de pernas tensionadas (TLP). Além disso podemos também citar os risers flexíveis, usados para transportar petróleo do fundo do mar para a plataforma e as linhas usadas na ancoragem das plataformas flutuantes.

Obviamente estas estruturas, por sofrerem grandes deslocamentos e terem um comportamento dinâmico bastante acentuado, necessitam de um tratamento especial em sua análise estrutural. Várias simplificações feitas anteriormente, na análise de jaquetas não podem mais ser adotadas; devem ser consideradas adequadamente suas propriedades de inércia, um tratamento mais adequado aos efeitos não-lineares, e ainda devem ser empregados modelos mais refinados; por exemplo, no caso de torres devem ser empregados modelos completos que permitam a verificação de elementos individuais da estrutura, e no caso de sistemas flutuantes ancorados devem ser utilizados modelos acoplados que tratam em uma mesma análise dinâmica não-linear o casco, as linhas de ancoragem e os "risers".

Com isso, observamos a necessidade do desenvolvimento de ferramentas computacionais, como a proposta nesta tese, devido ao grau de complexidade atingido na análise de tais problemas. Estas ferramentas têm como principal objetivo tornar as análises cada vez mais econômicas em termos de esforço computacional.

Devemos lembrar que as aplicações citadas anteriormente são apenas uma parte de inúmeras outras que podem ser analisadas desta forma, utilizando esta filosofia. Para isto basta que o problema em questão tenha um comportamento dinâmico não-linear considerável e possa ser discretizado. Podemos citar: análise de edifícios altos, pontes com grandes vãos, estádios de futebol, etc.

1.2 - Metodologia

Na análise de problemas inerciais, sabe-se que os modos que devem ser integrados corretamente são de frequência mais baixa (período mais alto). Isto favorece o desempenho de um algoritmo implícito, já que o intervalo de tempo necessário para integrar com a precisão desejada estes modos de vibração é muito maior que o intervalo de tempo crítico necessário para se obter estabilidade condicional em algoritmos explícitos. Este fato compensa o maior custo por instante de tempo gasto pelos algoritmos implícitos em relação aos explícitos.

A equação diferencial do movimento deve ser integrada no tempo utilizando-se um método numérico que forneça um resultado satisfatório com o menor esforço computacional possível. Para isso apresenta-se um esquema "híbrido" ou "semi-implícito", que procura combinar as vantagens oferecidas pelos métodos de integração implícitos (maiores valores de intervalo de tempo) e explícitos (menor custo por intervalo de tempo).

Este esquema é montado a partir de:

- um algoritmo implícito (a "regra trapezoidal");

em conjunto com:

- o Método de Newton-Raphson (N-R) para tratamento dos efeitos não-lineares.

Para a solução dos sistemas efetivos de equações algébricas lineares, oriundos do tratamento numérico da equação do movimento discretizada no tempo e no espaço, utiliza-se o Método dos Gradientes Conjugados Pré-condicionado (MGCP). Observa-se que, se utilizarmos pequenos valores de tolerância para a convergência do MGCP,

teremos resultados com precisão semelhante aos obtidos pelos métodos implícitos baseados em "solvers" diretos, porém sem vantagens em termos de esforço computacional. Conceitualmente, a característica híbrida ou "semi-implícita" deste esquema surge à medida em que os valores para a tolerância do MGCP são aumentados. Para empregar valores maiores de tolerância de uma forma controlada, adota-se um critério de tolerância variável ao longo das iterações de N-R.

Além disso, surge a proposta de um esquema auto-adaptativo onde o próprio programa decide sobre qual esquema de solução é o mais indicado para o problema em questão: se o esquema tradicional para resolução de sistemas de equações, ou o esquema "semi-implícito" baseado no MGCP com tolerância variável.

1.3 - Organização do trabalho

O trabalho está organizado da seguinte forma: o capítulo 2 trata da formulação básica das equações do movimento, definição de algoritmos implícitos e explícitos, utilização de algoritmos para resolução de problemas não-lineares (Newton-Raphson) e critérios utilizados na seleção de um algoritmo de integração.

O capítulo 3 apresenta um histórico dos esquemas híbridos de integração no tempo, um resumo da formulação do Método dos Gradientes Conjugados Pré-condicionados, e descreve a formulação para a implementação do esquema híbrido em problemas não lineares, que inclui os critérios utilizados na implementação da tolerância variável para o MGCP. Este capítulo apresenta também a proposta de um esquema auto-adaptativo para a seleção do esquema de solução.

No capítulo 4 temos aplicações numéricas onde são testadas as implementações de tolerância variável em problemas de engenharia off-shore e o esquema auto-adaptativo. Foram utilizados para análise de resultados exemplos de torre complacente de exploração e produção de petróleo e um conjunto de risers.

Finalmente, o capítulo 5 traz conclusões a respeito do trabalho apresentado.

CAPÍTULO 2

FORMULAÇÃO BÁSICA

2.1 - Solução Numérica de Problemas Dinâmicos Lineares

A descrição matemática do comportamento dinâmico de estruturas é composta por um problema de valor inicial e de contorno, que incorpora um sistema de equações diferenciais parciais (as equações de movimento), e um conjunto de condições iniciais (no tempo) e de contorno (no espaço). Para a resolução deste problema e a obtenção da resposta dinâmica, utilizam-se métodos numéricos que discretizam as equações diferenciais no espaço e no tempo [2].

Assumindo que o Métodos dos Elementos Finitos é utilizado para a discretização espacial, obtém-se a seguinte forma semi-discreta para a equação de movimento, válida para problemas lineares:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{F}(t) \tag{1}$$

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. Consideramos que a introdução das condições de contorno está incorporada no processo de discretização espacial. As condições iniciais são dadas por:

$$\mathbf{u}(0) = \mathbf{u}_0 \; ; \qquad \dot{\mathbf{u}}(0) = \mathbf{v}_0 \tag{2}$$

onde $\mathbf{u}_0$ e $\mathbf{v}_0$ são, respectivamente, vetores contendo os valores conhecidos para os deslocamentos e velocidades nodais no instante t = 0.

Para a discretização das equações (1) no tempo utiliza-se um algoritmo de integração apropriado, que fornece aproximações $\mathbf{a}_{n+1}$, $\mathbf{v}_{n+1}$ e $\mathbf{d}_{n+1}$ para os valores exatos

$\ddot{\mathbf{u}}(t_{n+1})$, $\dot{\mathbf{u}}(t_{n+1})$, $\mathbf{u}(t_{n+1})$ a partir de aproximações obtidas em instantes anteriores. Substituindo estas aproximações na equação (1), obtemos a forma discretizada no tempo das equações de movimento:

$$\mathbf{M}\,\mathbf{a}_{n+1} + \mathbf{C}\,\mathbf{v}_{n+1} + \mathbf{K}\,\mathbf{d}_{n+1} = \mathbf{F}_{n+1} \tag{3}$$

Vamos considerar a família de algoritmos de Newmark [3], cujos operadores escrevem-se:

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\,\mathbf{v}_n + \frac{\Delta t^2}{2}[(1-2\beta)\mathbf{a}_n + 2\beta\mathbf{a}_{n+1}] \tag{4}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t[(1-\gamma)\mathbf{a}_n + \gamma\mathbf{a}_{n+1}] \tag{5}$$

Para cada valor dos parâmetros γ e β temos um algoritmo em particular. Neste trabalho iremos considerar a "Regra Trapezoidal", dada por $\gamma=1/2$ e $\beta=1/4$.

A implementação computacional de tal algoritmo pode ser feita de várias maneiras. Dentre elas, é interessante mencionar a implementação "por acelerações", onde os operadores de Newmark (4) e (5) são substituídos nas equações de movimento (3), resultando em:

$$\begin{aligned}[\mathbf{M} + \gamma\,\Delta t\,\mathbf{C} + \beta\,\Delta t^2\,\mathbf{K}]\,\mathbf{a}_{n+1} = \\ = \mathbf{F}_{n+1} - \mathbf{C}\,[\mathbf{v}_n + (1-\gamma)\,\Delta t\,\mathbf{a}_n] - \\ - \mathbf{K}\left[\mathbf{d}_n + \Delta t\,\mathbf{v}_n + \left(\frac{1}{2} - \beta\right)\Delta t^2\,\mathbf{a}_n\right]\end{aligned} \tag{6}$$

ou

$$\hat{\mathbf{A}}\,\mathbf{a}_{n+1} = \hat{\mathbf{b}} \tag{7}$$

que é um "sistema efetivo" de equações algébricas lineares que deve ser resolvido a cada instante de tempo. Neste caso as incógnitas são as acelerações; $\hat{\mathbf{A}}$ é a "matriz efetiva", e

$\hat{\mathbf{b}}$ é o "vetor de cargas efetivo" cujas expressões podem ser deduzidas observando a forma completa (6).

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* [2].

- $\beta \neq 0$: O sistema efetivo é acoplado. Neste caso exige-se uma técnica para resolução de sistemas de equações algébricas, e temos as características de um algoritmo *implícito*.

A implementação mais usual para a solução de problemas de dinâmica estrutural consiste em reescrever os operadores de Newmark em termos de acelerações e velocidades, e substituí-los na equação do movimento. Com isso, chegamos à seguinte expressão, em termos de deslocamentos, para o sistema efetivo:

$$\left[\frac{1}{\beta\,\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\,\Delta t}\mathbf{C} + \mathbf{K}\right]\mathbf{d}_{n+1} =$$

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

$$+ \mathbf{C}\left[\frac{\gamma}{\beta\,\Delta t}\mathbf{d}_n - \left(1 - \frac{\gamma}{\beta}\right)\mathbf{v}_n - \left(1 - \frac{\gamma}{2\beta}\right)\Delta t\,\mathbf{a}_n\right] \qquad (8)$$

ou

$$\hat{\mathbf{A}}\,\mathbf{d}_{n+1} = \hat{\mathbf{b}} \qquad (9)$$

onde, apesar da mesma denominação empregada na expressão (7), a "matriz efetiva" $\hat{\mathbf{A}}$ e o "vetor de cargas efetivo" $\hat{\mathbf{b}}$ têm expressões distintas que podem ser deduzidas observando-se a forma completa (8).

2.2 - Seleção de um algoritmo de integração

Para se decidir qual o algoritmo apropriado para a solução de um problema dinâmico, podem ser usados critérios baseados nas propriedades descritas a seguir, agrupadas em duas categorias:

- Propriedades intrínsecas ao algoritmo, e
- Fatores externos.

Dentre as propriedades intrínsecas, iremos descrever e comentar as seguintes:

- *Custo computacional por instante de tempo*
- *Precisão*
- *Convergência*
- *Consistência*
- *Estabilidade*
- *Amortecimento numérico*

Em seguida, iremos comentar sobre os seguintes fatores externos ao algoritmo:

- *Tipo de problema*
- *Topologia da malha de elementos finitos*
- *Tipo de computador*

2.2.1 - Propriedades intrínsecas de um algoritmo

• ***Custo computacional por instante de tempo***: É menor para algoritmos explícitos, pois neste caso a solução do sistema de equações é trivial.

• ***Precisão***: Propriedade de um algoritmo de se afastar pouco da solução exata, ou de introduzir poucos erros. O erro pode ser medido pelo alongamento do período (dispersão) e pelo decaimento da amplitude (dissipação), conforme mostrado na figura 2.1.

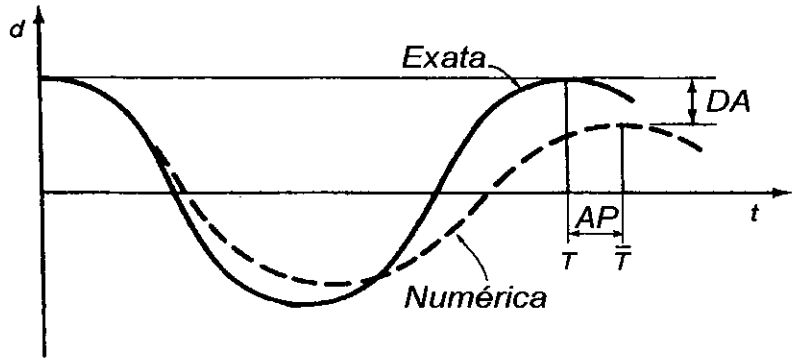


Fig. 2.1 - Alongamento de período e decaimento da amplitude.

As medidas de erros são representadas em gráficos, como os das figuras 2.2 e 2.3 [4], em função de valores de (Δt / T), onde T é o período natural de um dado modo de vibração da estrutura. Observe-se que a regra trapezoidal (Newmark com β=1/4 e γ=1/2) não apresenta nenhum decaimento de amplitude, e é o algoritmo com menor alongamento de período dentre os apresentados no gráfico da figura 2.3.

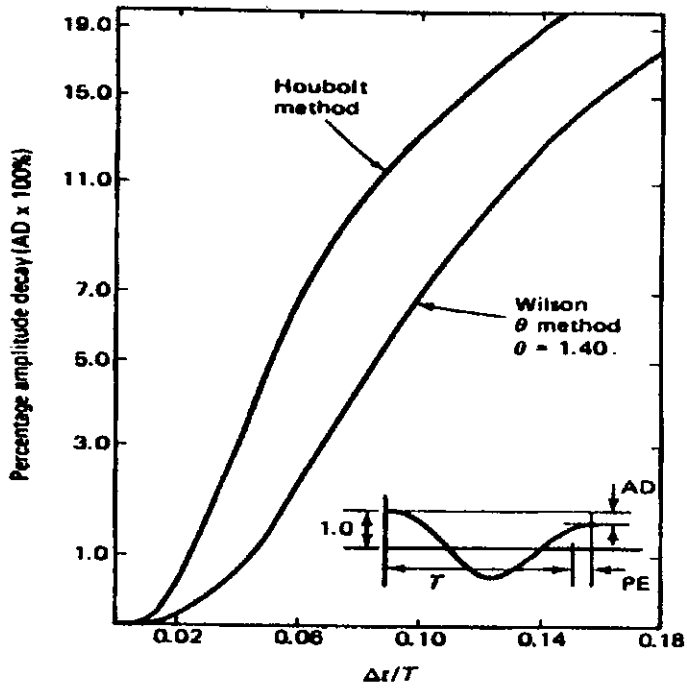


Fig. 2.2 - Percentagem de decaimento da amplitude

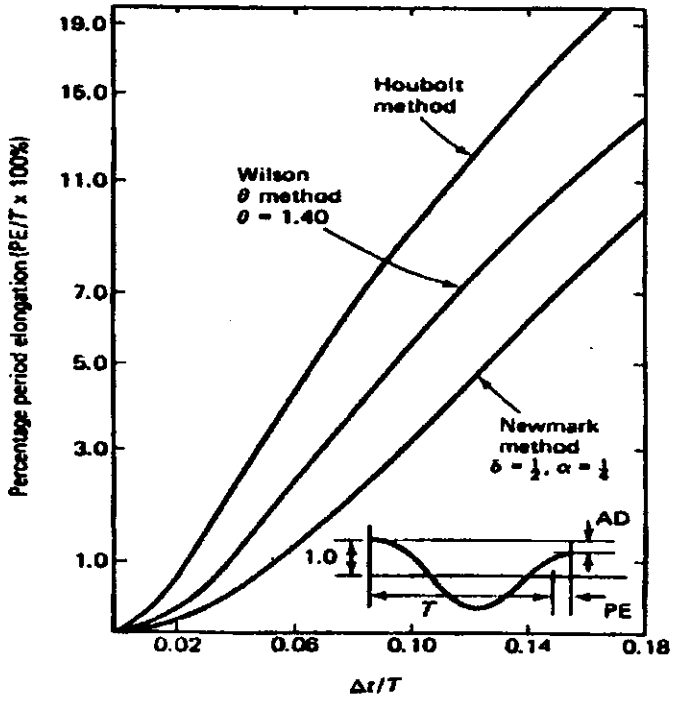


Fig. 2.3 - Percentagem de alongamento de período

• ***Convergência***: Um algoritmo é convergente quando a solução aproximada converge para a solução exata à medida que o intervalo de tempo tende a zero. Segundo

o Teorema de Lax as propriedades de consistência e estabilidade, sobre as quais iremos comentar a seguir, são necessárias e suficientes para a convergência do algoritmo:

• ***Consistência***: um algoritmo é consistente quando o erro de truncamento local, isto é, o resíduo das equações de recorrência do algoritmo quando se toma a solução exata no lugar da solução aproximada, tem um limite máximo, ou seja:

$$|\tau(t)| \leq c\, \Delta t^k \qquad (10)$$

onde:

$\tau(t)$ - erro de truncamento local;

c - constante de erro que não depende do intervalo de tempo Δt;

k - Ordem de Precisão ou Taxa de Convergência (k>0 de modo que o erro é diretamente proporcional ao Δt).

• ***Estabilidade***: Propriedade de um algoritmo de não amplificar erros resultantes de truncamento numérico e integração imprecisa dos modos de frequência alta. Quanto à estabilidade um algoritmo pode ser classificado em:

- *Algoritmo instável*: quando, devido à amplificação destes erros, a solução cresce sem limites.

- *Algoritmo com estabilidade condicional*: a estabilidade só é garantida com a utilização de intervalos de tempo menores que um determinado valor crítico. Para $\Delta t > \Delta t_{cr}$ a integração é instável. Δt_{cr} é função da maior frequência natural do modelo estrutural. Os métodos explícitos são condicionalmente estáveis. Para o Método da Diferenças Centrais, que é um método explícito (Newmark $\beta = 0$, $\gamma = 1/2$), temos $\Delta t_{cr} = T_n / \pi$, onde T_n é o menor período natural da malha utilizada na discretização.

- *Algoritmo incondicionalmente estável*: quando o algoritmo é estável para qualquer valor de intervalo de tempo. Na seleção de um valor de intervalo de tempo, basta atender um compromisso entre precisão e custo computacional.

Os membros da família de Newmark com $2\beta \geq \gamma \geq 0.5$ são implícitos e incondicionalmente estáveis, é o caso da regra trapezoidal. Existem algoritmos mais precisos do que a regra trapezoidal (Método de Fox-Goodwin, onde $\beta = 1/12$ e $\gamma = 1/2$, e o Método da Aceleração Linear, onde $\beta = 1/6$ e $\gamma = 1/2$), porém são menos utilizados devido à restrição de estabilidade condicional.

• ***Amortecimento Numérico*** : Na discretização espacial via MEF ocorre a introdução de erros no espectro de frequências, em comparação com o problema real. As equações semi-discretas aproximam melhor as frequência mais baixas. As frequências mais altas podem não ter significado físico algum, tornando-se desejável uma forma de eliminá-las da resposta dinâmica. Para isso introduz-se dissipação ou amortecimento numérico ao problema utilizando-se uma das alternativas abaixo:

- Amortecimento viscoso: Poderia-se usar a parcela de amortecimento proporcional à massa e/ou à rigidez não para representar um amortecimento real, mas como um artifício para dissipar os modos de alta frequência. Esta não é uma boa solução pois é possível demonstrar [1] que não há dissipação dos modos de frequência mais altos, somente as frequências intermediárias são afetadas.

- Algoritmos que apresentam $\rho(A) < 1$ (onde $\rho(A)$ é o raio espectral da matriz de amplificação do algoritmo, e, por este valor ser menor que 1, há um decaimento na resposta): são os métodos da família de Newmark com $\gamma > 0.5$ e $\beta = \frac{(\gamma + 0.5)^2}{4}$, os métodos de Houbolt, Park e Wilson-θ [3]. Todos estes métodos apresentam algumas

desvantagens, com a ordem de precisão caindo para 1 e/ou afetando em demasia os modos de frequência mais baixos.

- Algoritmo com dissipação numérica controlada: A grande vantagem destes algoritmos é a disponibilidade de um parâmetro que permite controlar o grau de dissipação, e o fato de não afetar significantemente os modos de frequência mais baixa. O algoritmo *αH-Newmark (HHT)* [5], por exemplo, utiliza os operadores da família de Newmark, porém com a introdução de um parâmetro ajustável α que permita controlar o grau de dissipação ($-1/3 \leq \alpha \leq 0$). Assim temos a seguinte equação de movimento discretizada no tempo:

$$\mathbf{M}\,\mathbf{a}_{n+1} + (1+\alpha)\,\mathbf{C}\,\mathbf{v}_{n+1} - \alpha\,\mathbf{C}\,\mathbf{v}_n + (1+\alpha)\,\mathbf{K}\,\mathbf{d}_{n+1} - \alpha\,\mathbf{K}\,\mathbf{d}_n = \mathbf{F}_{n+1+\alpha} \tag{11}$$

$$\mathbf{F}_{n+1+\alpha} = (1+\alpha)\,\mathbf{F}_{n+1} - \alpha\,\mathbf{F}_n \tag{12}$$

e os parâmetros γ e β são dados por:

$$\gamma = \frac{(1-2\alpha)}{2} \tag{13}$$

$$\beta = \frac{(1-\alpha)^2}{4} \tag{14}$$

O algoritmo resultante é incondicionalmente estável, com ordem de precisão 2. Observa-se que, para o caso particular $\alpha=0$, recai-se na regra trapezoidal, que não apresenta amortecimento numérico.

2.2.2 - Fatores externos ao algoritmo

- ***Tipo de problema***:

Em problemas inerciais ou de dinâmica estrutural, a resposta dinâmica é dominada por modos de vibração de frequências mais baixas. Neste caso deve-se integrar

com precisão apenas os modos de frequência 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 iria requerer 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 frequências intermediárias ou altas. Neste caso, os modos que devem ser integrados corretamente são os modos de frequência 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. Consequentemente 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 frequências mais altas do modelo reproduzam adequadamente as frequências correspondentes do problema físico real.

- ***Topologia da malha de elementos finitos***:

Problemas onde a matriz efetiva tem largura de banda pequena não são muito prejudicados pelo custo da solução do sistema de equações, o que favorece o uso de algoritmos implícitos.

- ***Tipo de computador***:

Computadores com limitação severa de memória podem forçar a utilização de esquemas explícitos, sem o armazenamento de matrizes globais.

Com base nos critérios apresentados ao longo da seção 2.2, é possível concluir que, para os problemas inerciais que iremos considerar neste trabalho, a regra trapezoidal ou suas variações com amortecimento numérico (como o Método αH-Newmark) são de fato os algoritmos mais adequados. Para chegar a esta conclusão podemos observar também os teoremas de Dahlquist:

Teoremas de Dahlquist:

a - Não existe um algoritmo explícito incondicionalmente estável.

b - Não existe um algoritmo incondicionalmente estável com ordem de precisão maior ou igual a 3.

c - O algoritmo incondicionalmente estável com ordem de precisão igual a 2 e com menor constante de erro é a regra trapezoidal.

2.3 - Problemas não-lineares

2.3.1 - Tratamentos do problema não-linear em algoritmos implícitos

As equações de movimento semi-discretas para problemas não-lineares podem ser expressas da forma:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{R}(u) = \mathbf{F}(u,t) \tag{15}$$

As não-linearidades associadas a (15) estão embutidas nas parcelas $\mathbf{R}(u)$ e $\mathbf{F}(u,t)$. A primeira parcela inclui efeitos geométricos e/ou de materiais com comportamento elástico não-linear. A segunda parcela considera não-lineaeridade devido à variação das cargas externas com a geometria, caracterizando carregamentos não-conservativos.

A solução do sistema de EDO não-linear (15) associado a algoritmos implícitos exige procedimentos específicos para o tratamento das não-linearidades.

A forma discretizada das equações do movimento é expressa por:

$$\mathbf{M}\,\mathbf{a}_{n+1} + \mathbf{R}(\mathbf{d}_{n+1}) = \mathbf{F}_{n+1}(\mathbf{d}_{n+1}) \tag{16}$$

No entorno de uma configuração deformada u, o problema pode ser assumido localmente linear. Esta linearização consiste em tomar a seguinte aproximação para as parcelas não-lineares, através de uma série de Taylor com termos de ordem superior truncados:

$$\mathbf{R}(\mathbf{d}_{n+1}) = \mathbf{R}(\mathbf{d}_n) + \left.\frac{\partial \mathbf{R}}{\partial \mathbf{d}}\right|_{\mathbf{d}_n} \Delta\mathbf{d} \tag{17}$$

$$\mathbf{F}_{n+1}(\mathbf{d}_{n+1}) = \mathbf{F}_{n+1}(\mathbf{d}_n) + \left.\frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{d}}\right|_{\mathbf{d}_n} \Delta\mathbf{d} \tag{18}$$

onde

$$\Delta\mathbf{d} = \mathbf{d}_{n+1} - \mathbf{d}_n \tag{19}$$

A última parcela de (18), 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:

$$\mathbf{K}_T = \frac{\partial \mathbf{R}}{\partial \mathbf{d}} \tag{20}$$

Substituindo (17), (18) e (20) em (16), as equações de equilíbrio dinâmico escrevem-se da seguinte forma:

$$\mathbf{M}\,\mathbf{a}_{n+1} + \mathbf{K}_T\,\Delta\mathbf{d} = \mathbf{F}_{n+1}(\mathbf{d}_n) - \mathbf{R}(\mathbf{d}_n) \tag{21}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta\mathbf{d} \tag{22}$$

onde $\mathbf{R}(\mathbf{d}_n)$ são os esforços elásticos resistentes calculados com os deslocamentos do intervalo anterior.

Deve-se observar que as equações (21) não mais garantem o equilíbrio dinâmico ao fim do intervalo de tempo t_{n+1}, devido às linearizações assumidas em (17).

2.3.2 - Formulação incremental-iterativa

Em problemas não-lineares os sistemas "efetivos" de equações algébricas, que resultam da aplicação do algoritmo de integração, passam a ser não-lineares, para cuja solução pode-se empregar técnicas iterativas semelhantes às utilizadas para problemas estáticos, através de uma formulação incremental-iterativa. Dentro da classificação genérica de formulações incremental-iterativas (i-i) são englobadas várias estratégias que buscam garantir o equilíbrio dinâmico. A obtenção da resposta dinâmica não-linear recai na montagem e solução de uma série de sistemas de equações algébricas não-lineares; as formulações i-i utilizam técnicas iterativas para resolver estes sistemas.

Como os intervalos de tempo que são requeridos para a integração com precisão adequada dos modos de vibração que dominam a resposta dinâmica geralmente representam incrementos de carga menores que os utilizados em casos estáticos, e a matriz efetiva é mais bem condicionada que a matriz de rigidez original, a solução iterativa de um problema dinâmico não-linear tende a convergir mais rapidamente do que um problema estático. Assim, uma análise dinâmica não necessita de estratégias iterativas mais caras ou elaboradas, e tradicionalmente emprega-se o método de Newton-Raphson Modificado (NRM) [1] no qual a matriz tangente é mantida constante ao longo do ciclo iterativo, e é reavaliada apenas ao início de instantes de tempo escolhidos arbitrariamente.

A formulação do Método de Newton-Raphson baseia-se em adotar a linearização (17) e iterar com matrizes tangentes como a dada por (20). No método Newton-Raphson padrão NRP, a matriz tangente é reavaliada em todas iterações. No entanto, 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 opção mais utilizada. Nesta técnica, a matriz de rigidez tangente $\mathbf{K}_T$ 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.

As equações de movimento não-lineares discretizadas no tempo (21) devem ser colocadas de uma forma incremental iterativa (i-i) para a implementação do algoritmo de Newmark com a formulação NRM. Considerando-se que $\Delta\Delta\mathbf{d}_{n+1}^{(k)}$ representa a variação dos deslocamentos incrementais obtida a cada iteração do ciclo de verificação do equilíbrio, a expressão i-i escreve-se na forma:

$$\mathbf{M}\,\mathbf{a}_{n+1}^{(k)} + \mathbf{K}_T\,\Delta\Delta\mathbf{d}_{n+1}^{(k)} = \mathbf{F}_{n+1}(\mathbf{d}_n) - \mathbf{R}(\mathbf{d}_{n+1}^{(k-1)},\ \Delta\Delta\mathbf{d}_{n+1}^{(k-1)}) \tag{23}$$

$$\Delta\mathbf{d}^{(k)} = \Delta\mathbf{d}^{(k-1)} + \Delta\Delta\mathbf{d}_{n+1}^{(k)} \tag{24}$$

$$\mathbf{d}_{n+1}^{(k)} = \mathbf{d}_{n+1}^{(k-1)} + \Delta\Delta\mathbf{d}_{n+1}^{(k)} \tag{25}$$

O vetor de cargas externas $\mathbf{F}_{n+1}(\mathbf{d}_n)$ é reavaliado ao início de cada intervalo de tempo, e é mantido constante ao longo do ciclo iterativo. Os esforços elásticos resistentes $\mathbf{R}(\mathbf{d}_{n+1}^{(k-1)},\ \Delta\Delta\mathbf{d}_{n+1}^{(k-1)})$ estão expressos também em função das variações dos deslocamentos incrementais $\Delta\Delta\mathbf{d}_{n+1}^{(k-1)}$ porque estes são utilizados na formulação do elemento de pórtico não-linear tridimensional empregado [6].

CAPÍTULO 3

ESQUEMAS HÍBRIDOS DE INTEGRAÇÃO NO TEMPO

3.1 - Histórico

Os esquemas híbridos de integração no tempo [7] têm como objetivo alcançar maior eficiência computacional, procurando combinar as vantagens oferecidas pelos métodos implícitos e explícitos, de modo a permitir o uso de maiores valores de intervalo de tempo, com menores custos por instante de tempo. Tais esquemas englobam diferentes técnicas (conceitualmente semelhantes), que serão revisadas a seguir.

3.1.1 - Técnicas de partição do domínio

Tais técnicas são empregadas 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 incluem-se:

- os *Métodos de partição implícito-explícito* [8] 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* [9, 10] que empregam o mesmo operador em todas as regiões (implícito ou, mais geralmente, explícito), porém diferentes intervalos de tempo para cada região. Tais métodos tem sido aplicados em problemas de propagação de ondas em solos com camadas de propriedades físicas diferentes, e cravação de estacas.

3.1.2 - Métodos de partição do operador (Operator splitting)

Estes métodos são desenvolvidos a partir de um operador implícito, empregando artifícios numéricos para reduzir os custos de solução do sistema efetivo. Pode-se demonstrar que métodos de partição do operador são conceitualmente semelhantes aos métodos de partição do domínio; no entanto, estes últimos são orientados a problemas particulares caracterizados por regiões com propriedades distintas. Já os métodos de partição do operador pretendem ser de aplicação geral, onde o próprio algoritmo de integração é modificado de modo a obter um novo algoritmo que idealmente combine as características de estabilidade incondicional de um operador implícito com o baixo custo por instante de tempo de um operador explícito.

Métodos de partição do operador têm forte tradição em conjunto com a discretização espacial por diferenças finitas. A aplicação de tais métodos associada ao método dos Elementos Finitos inicia-se a partir do trabalho de Trujillo [11], que sugeriu "partir" as matrizes de rigidez e amortecimento em matrizes triangulares inferiores e superiores, substituindo-se a decomposição da matriz global por uma operação de substituição e uma retro-substituição. Pode-se citar também o trabalho de Park [12], que identificou problemas de precisão no método de Trujillo e apresentou variações e extensões, chamando o método resultante de "semi-implícito".

Os métodos de partição do operador são também chamados de métodos "semi-implícitos", "semi-explícitos", "explícitos incondicionalmente estáveis" [13,14]. Demonstra-se que, de fato, tais métodos mantém a estabilidade de algoritmos implícitos, a um custo computacional menor, no entanto a precisão é inferior à dos métodos implícitos tradicionais. Além disso herdam a condição de consistência condicional, característica de métodos similares em diferenças finitas. A característica de consistência

condicional indica que o erro de truncamento local terá um limite máximo somente quando $\Delta t \rightarrow 0$ mais rapidamente que $h \rightarrow 0$, onde h é o espaçamento nodal da malha. Isto significa que, para um determinado valor de intervalo de tempo, o refinamento da discretização espacial poderá degradar a qualidade da resposta. Por um raciocínio inverso, para uma determinada discretização espacial a restrição ao valor do intervalo de tempo é mais severa do que a exigida por critérios de precisão para os métodos implícitos.

Os métodos semi-implícitos tradicionais podem ser vistos como a aplicação de técnicas iterativas "rudimentares" na solução dos sistemas efetivos de equações resultantes da implementação de um método implícito [7]. Na verdade, a fronteira entre "métodos semi-implícitos" e "métodos implícitos com solução iterativa dos sistemas de equações" é difusa. Pode-se julgar que um método deixa de ser semi-implícito quando a técnica iterativa é rigorosa o suficiente para fornecer uma solução adequada para o sistema de equações efetivo. Os problemas de precisão associados a técnicas semi-implícitas deixam de existir, mantidas as características de estabilidade e precisão do método implícito original.

A evolução natural da linha de "métodos semi-implícitos" ou "semi-explícitos" coincide com a motivação da realização do presente trabalho, que consiste em estudar a melhor forma de utilização e implementação de técnicas iterativas para a solução dos sistemas efetivos com o objetivo de minimizar os custos da solução. Um passo nessa direção foi dado por Nour-Omid [15] utilizando o esquema de Trujillo para montar um pré-condicionador para o Método dos Gradientes Conjugados Pré-condicionado (MGCP) [16]. Desde então esta linha de pesquisa tem se consolidado, desvinculada dos

métodos de integração no tempo e concentrada no estudo de técnicas iterativas (como o MGCP).

3.2 - O Método dos Gradientes Conjugados

O Método dos Gradientes Conjugados [17] explora o fato de que a solução do sistema de equações $\hat{\mathbf{A}}x = \hat{\mathbf{b}}$ equivale à minimização do funcional quadrático:

$$Q = \frac{1}{2} x^t \hat{\mathbf{A}} x - x^t \hat{\mathbf{b}} \tag{26}$$

Assim, processos iterativos utilizados para minimizar o funcional Q, conduzem à solução do sistema de equações algébricas associadas a este funcional. Nos algoritmos de solução que empregam o Método Iterativo dos Gradientes Conjugados a matriz dos coeficientes atua unicamente como operador linear, fornecendo uma maneira de explorar a esparsidade da matriz. A eficiência do método está intrinsicamente relacionada com as características próprias do operador $\hat{\mathbf{A}}$, isto é, seus autovalores. Neste caso a razão entre o maior e o menor autovalores do espectro, λ_n e λ_1 respectivamente,

$$k(\hat{\mathbf{A}}) = \frac{\lambda_n}{\lambda_1} \tag{27}$$

é denominada razão espectral, e quando a minimização do funcional é eficiente, este valor se aproxima da unidade. Caso contrário, torna-se necessário a utilização de uma técnica de pré-condicionamento.

Logo, com o objetivo de atingir uma convergência mais rápida, o Método dos Gradientes Conjugados pode ser aplicado a um sistema original, mas que tenha sido pré-condicionado. Pré multiplicando-se a equação original do sistema pela matriz não-singular R e aplicando-se o Método dos Gradientes Conjugados ao sistema pré-condicionado, obtém-se o algoritmo dos Gradientes Conjugados Pré-condicionado

(MGCP). Definindo-se como matriz R a diagonal da matriz dos coeficientes $\hat{\mathbf{A}}$, temos o pré-condicionamento diagonal, que devido à sua simplicidade de implementação, baixo número de operações envolvidas e efeito favorável de escalonamento de variáveis de grandezas dimensionais diferentes é muito utilizado.

- O ciclo iterativo do MGCP

→ Iteração MGCP

(1) Inicializa $p_0 = 0$, $r_0 = \hat{\mathbf{b}}$, $\Delta d_0 = - R^{-1} r_0$

(2) $\alpha_i = r_i^T R^{-1} r_i \,/\, \Delta d_i^T \hat{\mathbf{A}}\, \Delta d_i$

$p_{i+1} = p_i + \alpha_i \Delta d_i$

$r_{i+1} = r_i + \alpha_i \hat{\mathbf{A}}\, \Delta d_i$

se $\| r_{i+1} \| \,/\, \| r_0 \| \leq \eta$ sair do loop $p = p_{i+1}$

senão continuar passo 3

(3) $\beta_i = r_{i+1}^T R^{-1} r_{i+1} \,/\, r_i^T R^{-1} r_i$

$\Delta d_{i+1} = - R^{-1} r_{i+1} + \beta_i \Delta d_i$

retorna para passo (2)

R - pré-condicionador (p.ex. diagonal da matriz efetiva)

3.3 - O esquema "semi-implícito"

Os seguintes pontos, relacionados à implementação do MGCP em problemas dinâmicos, servem de motivação para os desenvolvimentos deste trabalho:

• Devido à contribuição da matriz de massa diagonal **M** sobre a matriz efetiva **A** (eq. (8)), esta fica mais diagonal-dominante ou mais bem condicionada do que a matriz de rigidez original, o que vem favorecer a convergência do MGCP. Mais ainda, quanto menores os intervalos de tempo mais este efeito positivo será realçado, pois contribuição da matriz de massa **M** é afetada por um coeficiente inversamente proporcional ao quadrado do valor do intervalo de tempo.

• Mesmo com a utilização de valores de tolerância pequenos, o MGCP tem se mostrado competitivo com técnicas diretas na solução de grandes sistemas de equações lineares.

O comentário citado anteriormente pode ser comprovado se observarmos a figura 3.1 [6,18]. Podemos notar que, quanto menor o intervalo de tempo utilizado em uma determinada análise, menor o número médio de iterações MGCP necessários para a convergência do método. O número médio de iterações MGCP é obtido dividindo-se o número total de iterações MGCP pelo número total de iterações NR, acumulados ao longo de toda a análise.

Com a finalidade de compararmos estes valores com o "solver" direto utilizamos um número equivalente de iterações MGCP. Para obter este número dividimos o tempo de CPU da solução direta, que é constante para esta aplicação, pelo custo de uma iteração MGCP, o qual, por sua vez, é obtido dividindo-se o tempo total de CPU correspondente às soluções MGCP pelo número total de iterações MGCP.

Para qualquer intervalo de tempo usado, o "solver" iterativo foi sempre mais vantajoso que o "solver" direto.

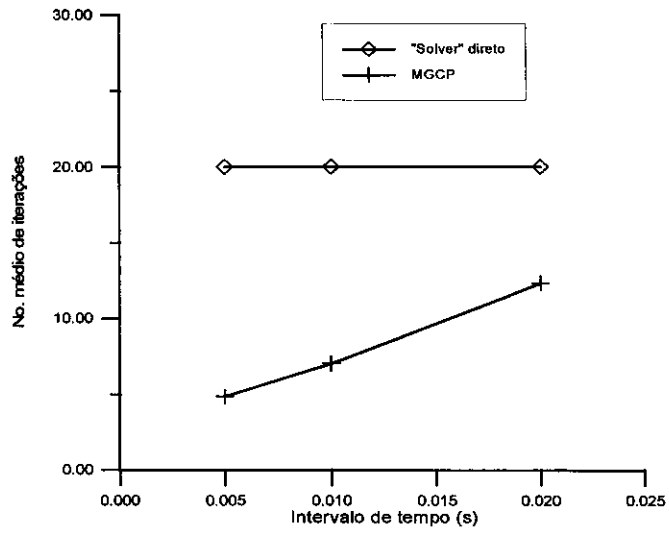


Figura 3.1 - Performance relativa "solver" direto x MGCP

Além disso, lembramos os seguintes pontos que dizem respeito à implementação do MGCP em problemas dinâmicos não-lineares:

- A matriz efetiva deve ser atualizada com frequência; com isso reduz-se a vantagem de técnicas diretas que permitem a solução do sistema efetivo apenas através de retrosubstituição de uma matriz decomposta constante;

- A utilização do MGCP resulta em um ciclo iterativo interno ao ciclo de N-R;

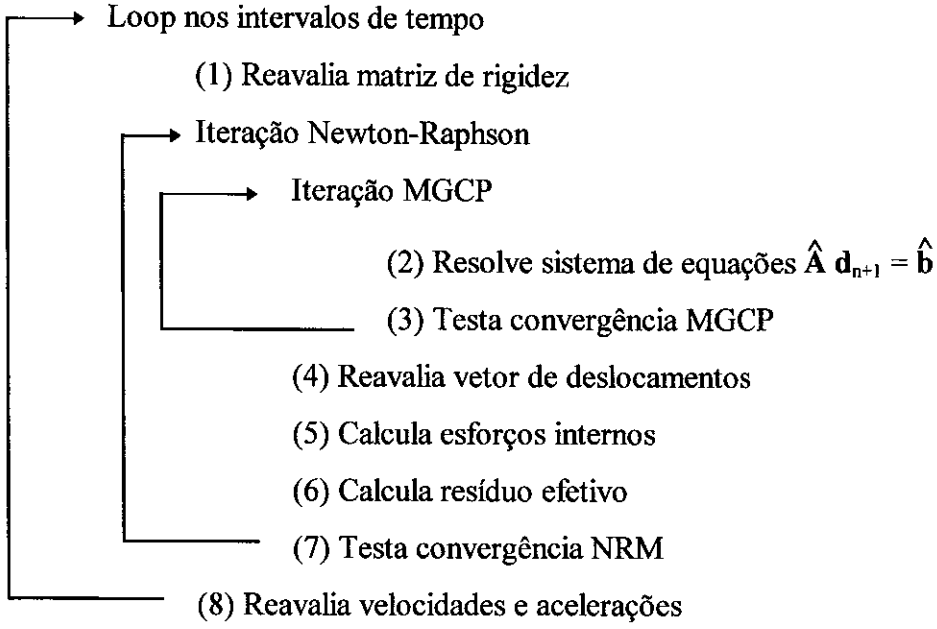
- Apesar de apresentar precisão de razoável a pobre, métodos "semi-implícitos" como de Trujillo ou Park foram capazes de reduzir custos computacionais.

Estes pontos nos levam às seguintes conclusões: Se utilizarmos um valor de tolerância muito pequeno para o MGCP, teremos um resultado idêntico ao obtido por uma técnica direta de solução em um método implícito tradicional. No entanto, é possível ***definir um esquema híbrido aumentando de forma controlada o valor da tolerância, de modo a obter eficiência computacional e manter as características de estabilidade e precisão de um método implícito***.

Pode-se além disso adotar um esquema de ***tolerância variável*** para o MGCP ao longo das iterações de N-R, lembrando que nas primeiras iterações de N-R pode-se obter uma solução menos precisa do que a requerida próximo à convergência do ciclo N-R.

Com isso temos o seguinte esquema "semi-implícito" de solução:

- O ciclo iterativo Newton-Raphson



Tal esquema pode ser semelhante ao que tem sido apresentado na literatura para problemas estáticos, usualmente referidos como "Truncated Newton Method". Por exemplo, Dembo e Steinhaug [19] propuseram a tolerância (η) variando da seguinte forma:

$$\eta_k = \min\left[\frac{1}{(k+1)}, \|g(x_k)\|^t\right] \qquad \text{com } 0 < t \leq 1 \qquad (28)$$

onde k é o contador de iterações Newton-Raphson; η é a tolerância do ciclo iterativo do MGCP, e $\|g(x_k)\|$ é a norma do vetor de resíduos efetivos de esforços internos dependente dos deslocamentos x.

Nour-Omid et alii. [20] propõem um outro critério de convergência especificado por:

$$\eta_k = \eta_0 \cdot \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^{1.5} \qquad (29)$$

Papadrakakis [21] propõe o seguinte critério:

$$\eta_k = \min\left[\eta_0, \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^t\right] \qquad \text{com } 0 < t < 1 \tag{30}$$

onde a tolerância inicial η_0 e o expoente t são parâmetros ajustáveis.

Com o objetivo de melhorar os critérios de tolerância variável serão propostas, neste momento, variações de alguns critérios acima apresentados.

A primeira variação é sobre o critério de Dembo / Steinhaug. A variação proposta foi a de se trocar o valor da norma do resíduo da k-ésima iteração pela razão da norma do resíduo da k-ésima iteração pela razão da norma da primeira iteração, e pode ser equacionada da seguinte maneira:

$$\eta_k = \min\left[\frac{1}{(k+1)}, \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^t\right] \qquad \text{com } 0 < t < 1 \tag{31}$$

Também foram propostas variações sobre o critério de Nour-Omid, onde procurou-se variar o expoente no cálculo da tolerância:

$$\eta_k = \eta_0 . \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^{1.0} \tag{32}$$

$$\eta_k = \eta_0 . \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^{0.7} \tag{33}$$

Estas variações foram propostas com o intuito de se fazer com que a tolerância do MGCP diminuísse mais lentamente à medida que as iterações N-R fossem evoluindo, isto é, à medida que a razão da norma do resíduo fosse diminuindo.

Para que se tenha uma melhor noção do comportamento de variação do MGCP em função da norma do vetor de resíduos efetivos, as figuras 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 e 3.8 a seguir apresentam gráficos de todos os critérios descritos anteriormente.

Para o traçado destes gráficos foram tomados valores típicos para a norma do resíduo, observados em análises não-lineares para os exemplos apresentados no capítulo 4. Assim, as retas verticais representam valores observados na segunda iteração NR de cada instante de tempo. Estes valores aparecem neste momento apenas para se ter uma noção prática de sua ordem de grandeza.

O gráfico da figura 3.2 mostra o critério proposto por Papadrakakis, que toma o mínimo entre um valor η_0 e a razão entre a norma do resíduo da atual iteração NR e a norma do resíduo da primeira iteração. Cada curva deste gráfico corresponde a um valor arbitrado para o expoente t. Com a variação destes valores obtivemos a variação da tolerância MGCP e pudemos observar qual seria o comportamento desta variação. A reta horizontal é o valor da tolerância MGCP, constante a cada iteração NR, utilizado na comparação do valor mínimo deste critério.

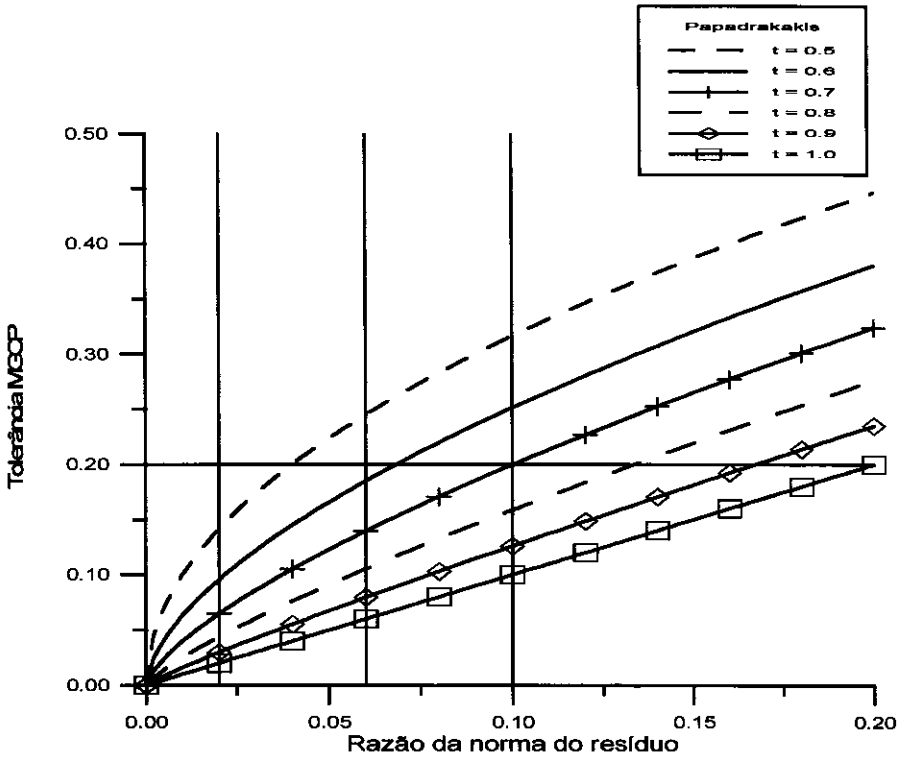


Fig. 3.2 - Variação da tolerância segundo o critério de Papadrakakis -

$$\eta_k = \min\left[\eta_0, \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^t\right]$$

O gráfico da figura 3.3 apresenta o critério proposto originalmente por Dembo / Steinhaug (eq. (28)) o qual faz uma escolha de mínimo entre o inverso do número da iteração NR e a norma do resíduo da iteração atual elevada a um expoente t. O gráfico foi montado variando-se o expoente t e obtendo-se valores de tolerância para convergência no MGCP. Um fato observado é o de que o valor da tolerância seria sempre limitado pelo valor 1/(k+1) devido aos altos valores obtidos com a norma do resíduo. Além disso estes limites para a tolerância MGCP continuaram elevados em relação ao esperado, o que pôde ser verificado na utilização deste critério nas aplicações numéricas, onde ocorreram problemas de precisão na resposta obtida, conforme poderemos observar no sub-item 4.1.2.2. Tentando solucionar este problema propusemos a variação na equação (31).

Para o gráfico da figura 3.4 empregou-se a variação aqui proposta (eq. (31)). Com isto obtivemos menores valores de tolerância MGCP, resolvendo o problema de precisão e possibilitando a comparação com os demais critérios propostos. As retas horizontais correspondem aos valores limites 1/(k+1) tanto para a figura 3.3 quanto para a figura 3.4.

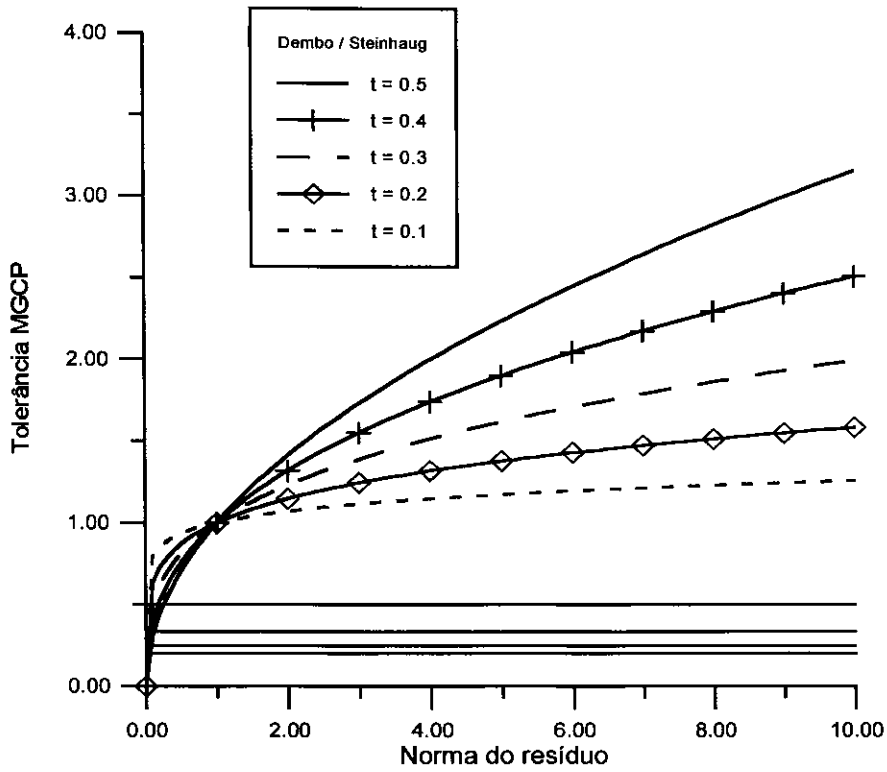


Fig 3.3 - Variação da tolerância segundo o critério de Dembo / Steinhaug -

$$\eta_k = \min\left[\frac{1}{(k+1)}, (\|g(x_k)\|)^t\right]$$

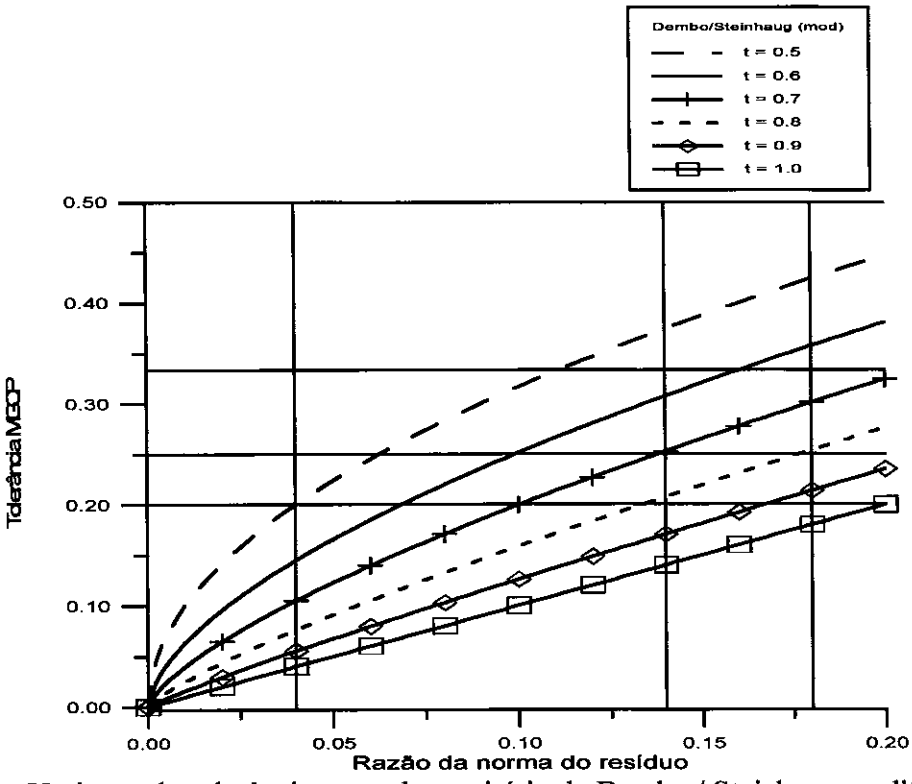


Fig 3.4 - Variação da tolerância segundo o critério de Dembo / Steinhaug modificado -

$$\eta_k = \min\left[\frac{1}{(k+1)}, \left(\frac{\|g(x_k)\|}{\|g(x_0)\|}\right)^t\right]$$

O gráfico da figura 3.5 foi obtido pela análise do critério proposto por Nour-Omid (eq. (29)) onde não se faz comparação de valores mínimos como nos critérios descritos anteriormente. Neste caso o valor da tolerância é calculado pelo produto de um coeficiente η_0 pela razão da norma do resíduo elevado a 1.5.

Podemos notar que o valor da tolerância decai à medida que a razão da norma do resíduo diminui. Com o objetivo de, ainda nas primeiras iterações NR termos maiores valores de tolerância MGCP, são propostas variações no critério de Nour-Omid.

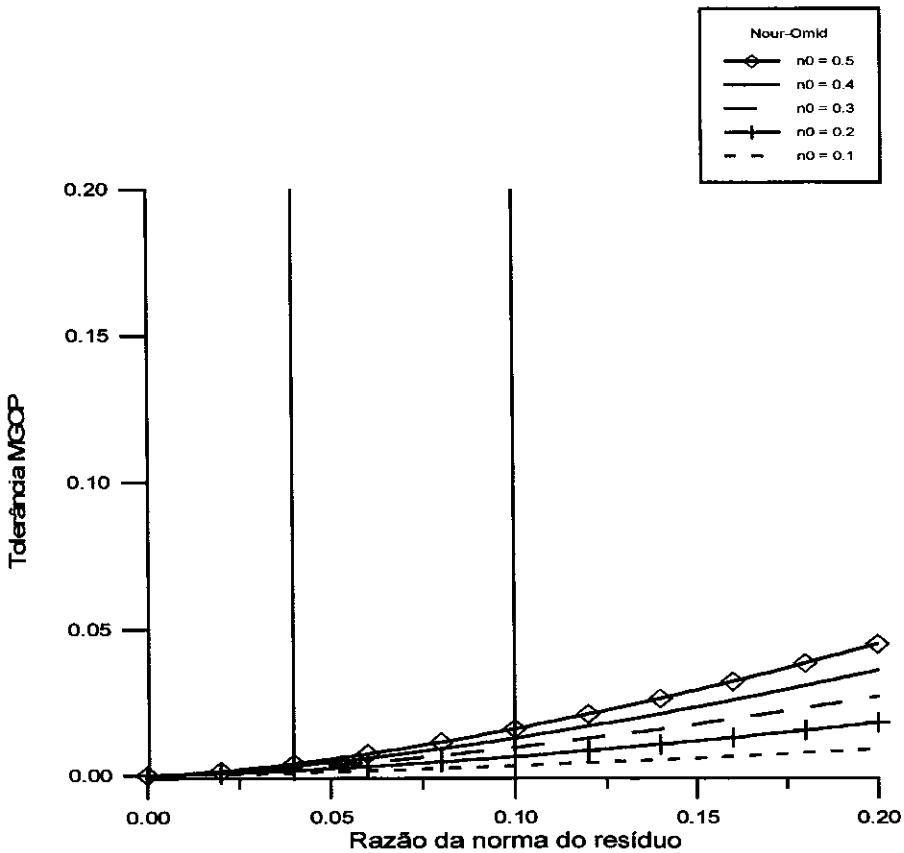


Fig. 3.5 - Variação da tolerância segundo o critério de Nour-Omid -

$$\eta_k = \eta_0 \cdot \left(\frac{\|g(x_k)\|}{\|g(x_0)\|} \right)^{1.5}$$

No gráfico da figura 3.6 propusemos um valor de expoente igual a 1 variando da mesma forma descrita anteriormente o valor de η_0. Feito isto espera-se obter algum ganho em termos de custo computacional.

Já no gráfico da figura 3.7 com o expoente assumindo o valor de 0.7 espera-se obter resultados onde o ganho se tornará mais evidente visto que os valores de tolerância MGCP são ainda maiores nas primeiras iterações NR decaindo próximo à convergência.

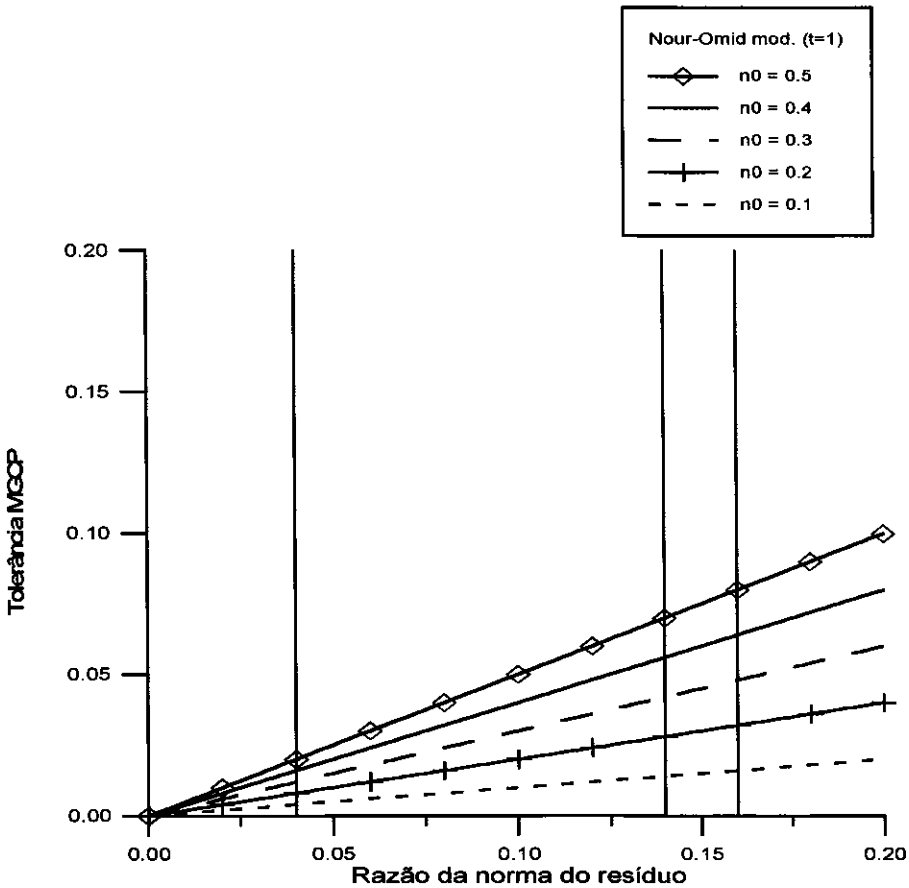


Fig. 3.6 - Variação da tolerância segundo o critério de Nour-Omid modificado - $\eta_k = \eta_0 \cdot \left(\frac{\|g(x_k)\|}{\|g(x_0)\|} \right)^1$

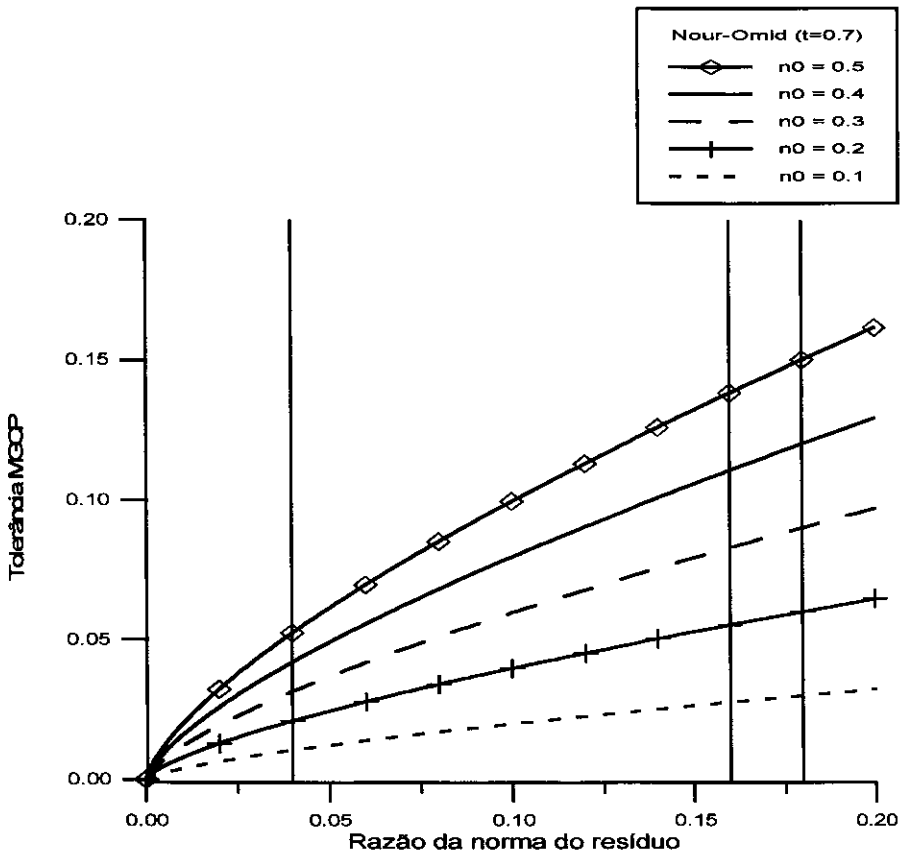


Fig. 3.7 - Variação da tolerância segundo o critério de Nour-Omid modificado - $\eta_k = \eta_0 \cdot \left(\frac{\|g(x_k)\|}{\|g(x_0)\|} \right)^{0.7}$

No gráfico da figura 3.8 temos a superposição de todos os critérios descritos anteriormente com os valores de coeficientes escolhidos de forma a tornar a análise mais eficiente e que serão utilizados nos exemplos descritos a seguir.

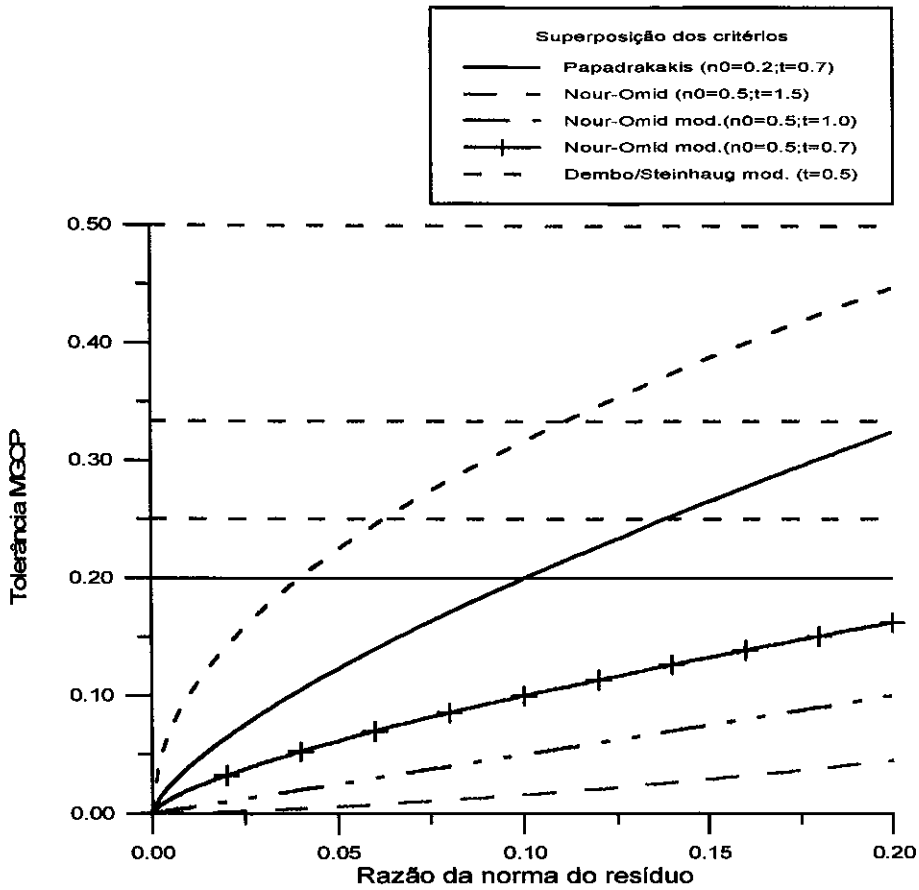


Fig. 3.8 - Superposição dos critérios c/ parâmetros usados nas aplicações numéricas

3.4 - Um esquema auto-adaptativo para resolução do sistema de equações

Neste momento, surge a proposta de utilização de um esquema auto-adaptativo para resolução do sistema de equações, com o objetivo de obter maior eficiência computacional procurando minimizar custos computacionais associados aos algoritmos de análise dinâmica não-linear.

O esquema é baseado em uma pesquisa comparativa, no início da simulação numérica de uma dada aplicação, do tempo de CPU dispendido utilizando-se um "solver" direto e o MGCP com tolerância variável. Feita esta pesquisa e, utilizando-se os resultados obtidos, o programa decide qual dos dois métodos será utilizado no restante da análise.

Neste ponto, notamos que, de acordo com o que já foi citado, quanto menor o intervalo de tempo necessário para correta integração da equação do movimento, mais vantajoso torna-se o uso do MGCP com tolerância variável e, à medida que o intervalo de tempo cresce, pode-se tornar vantajoso a utilização do "solver" direto.

Com isto, somos induzidos a concluir o seguinte: caso utilizemos um esquema de determinação automática de valores de intervalo de tempo, é de grande utilidade termos este esquema auto-adaptativo para resolução de sistemas de equações atuando em conjunto, visto que, acompanhando a variação do intervalo de tempo poderemos utilizar a mais vantajosa técnica de resolução de sistemas, em termos de medida de tempo de CPU.

Na atual implementação, o teste é feito nos primeiros 20 intervalos de tempo para o solver direto, do vigésimo ao quadragésimo intervalo de tempo para o MGCP com tolerância variável, e, a partir daí é feita a escolha do esquema de solução que será utilizado até o final da análise.

CAPÍTULO 4

APLICAÇÕES NUMÉRICAS

Neste capítulo vamos comparar o desempenho de um esquema implícito tradicional, que emprega um "solver" direto, ao esquema "semi-implícito" proposto, que incorpora o MGCP com tolerância variável segundo os critérios indicados para a solução dos sistemas "efetivos" de equações ao método de Newton-Raphson modificado para o tratamento dos efeitos não-lineares.

Para isso vamos considerar duas aplicações:

- A análise de uma torre estaiada para exploração e produção de petróleo offshore;

- Um conjunto de 8 risers flexíveis utilizados no transporte de fluidos que resultam do processo de produção (óleo, gás, lama, água), desde a cabeça do poço de petróleo, no fundo do mar, até a plataforma de exploração na superfície.

Os exemplos são detalhados nos sub-ítens a seguir.

Para análise dos resultados obtidos, foram plotados, para cada esquema de solução utilizado, gráficos que mostram a variação do número de iterações N-R, a norma do resíduo, a razão da norma do resíduo, a variação da tolerância MGCP e o número de iterações MGCP ao longo do tempo.

A comparação do desempenho computacional de cada esquema estudado será apresentada após o estudo de cada critério nos ítens 4.1.3 e 4.2.3.

4.1 - Torre estaiada.

Uma torre estaiada é uma estrutura complacente onde o elemento estrutural que produz as forças restauradoras é composto por linhas de amarração em catenária, ancoradas no fundo do mar e conectadas à estrutura em um nível próximo ao da resultante das cargas ambientais (fig. 4.1). Estas linhas são projetadas para apresentar rigidez variável de acordo com as condições ambientais. Com isto, a estrutura apresenta períodos naturais maiores que os períodos das ondas, e as forças de inércia contribuem para a estabilidade da estrutura, que pode suportar grandes deslocamentos provocados pelos carregamentos ambientais [22].

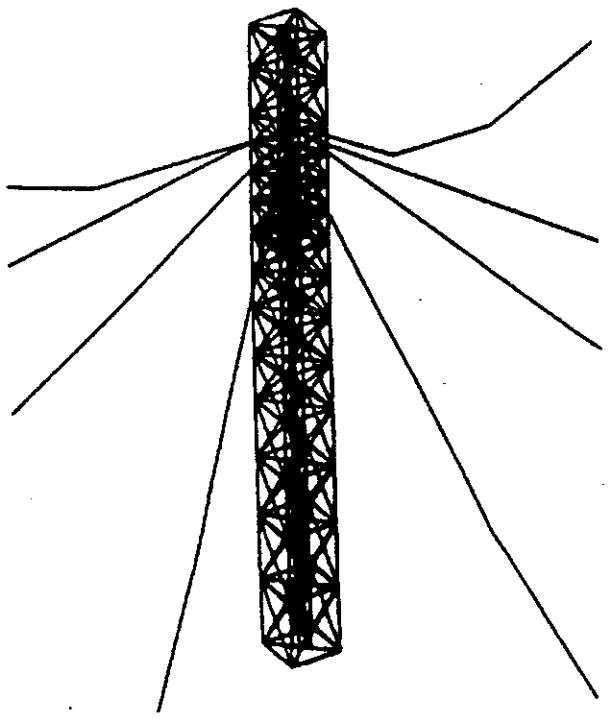


Fig. 4.1 - Torre estaiada (modelo completo)

A torre considerada tem 330 metros de altura, e é composta por uma torre externa de 27 metros de seção transversal quadrada conectada por uma torre interna de 7m de seção transversal quadrada. As barras constituintes têm 1.20m de diâmetro e espessura de 0.10m [22].

Na modelagem utilizada, conforme podemos ver na figura 4.2, todos os componentes da estrutura são discretizados através de elementos de pórtico (num total de 520 elementos e 139 nós). Além disso, a interação entre a torre e suas linhas de amarração é tratada através de um modelo desacoplado, que leva a uma simplificação para obter economia computacional. Neste modelo, inicialmente efetua-se a análise das linhas isoladamente; como resultado obtém-se curvas (deslocamento x força) no ponto de conexão com a torre. Estas curvas servirão então para definir molas não-lineares, que serão empregadas para representar as linhas na análise do modelo completo da estrutura.

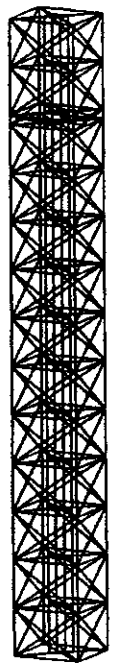


Fig. 4.2 - Torre estaiada (modelo desacoplado)

O carregamento ambiental considerado consiste em uma onda determinística de Airy com período de 14 segundos e altura de 18m, e correnteza descrita por um perfil de velocidades variando de 1.45 m/s na superfície a 0.25 m/s no fundo. Considera-se também a ação do vento, com velocidade de 55 m/s. Para representar as cargas do convés aplica-se uma carga vertical distribuída nos nós do topo da torre de 30000 kN. As figuras 4.3 e 4.4 apresentam, respectivamente, resultados típicos em termos de

deslocamentos horizontais e verticais no topo da torre para ambos os esquemas de solução considerados. Os resultados são, como seria de se esperar, praticamente coincidentes.

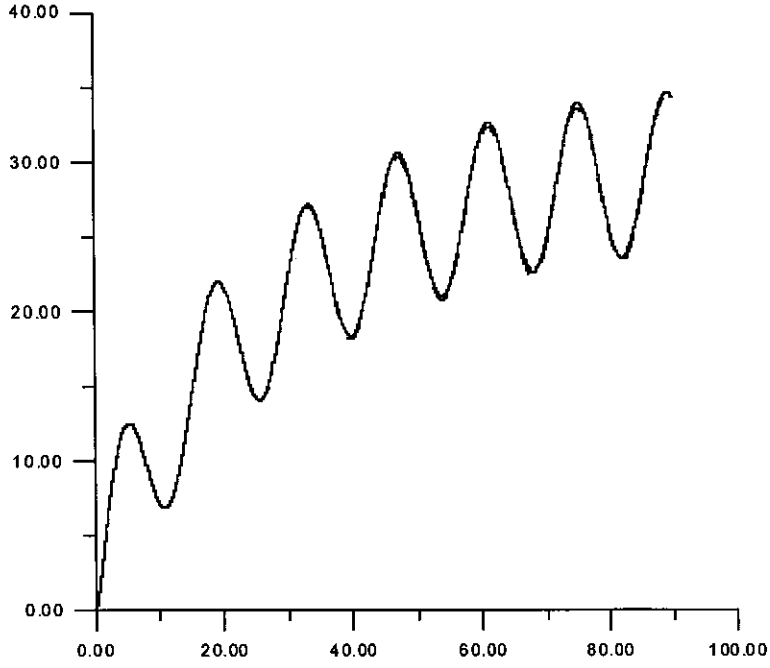


Fig. 4.3 - Deslocamento horizontal do topo

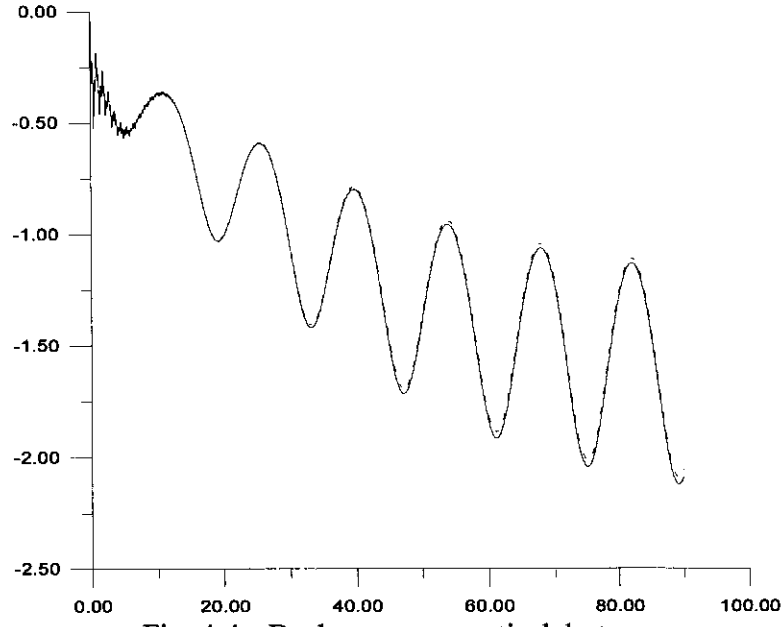


Fig. 4.4 - Deslocamento vertical do topo

Ao analisar-se este exemplo procedeu-se da seguinte forma: utilizou-se inicialmente o "solver" direto, e em seguida variações do MGCP com tolerância variável (Papadrakakis, Nour-Omid, Dembo / Steinhaug). Além destes critérios foram utilizadas as variações propostas nos critérios de Nour-Omid e Dembo / Steinhaug com a finalidade de se obter melhores resultados.

O intervalo de tempo utilizado para as análises efetuadas da torre foi de 0.005s. Este valor foi o escolhido porque, utilizando-se um intervalo de tempo maior (0.01s), ocorriam problemas de convergência antes do final da análise, devido à acentuada não-linearidade do exemplo; para um intervalo de tempo menor (0.0025s) o resultado obtido era idêntico ao obtido utilizando-se 0.005 segundos, porém era dispendido maior tempo de CPU. O valor adotado para a tolerância no Método de Newton-Raphson é de 0.01. Para o MGCP a tolerância inicial utilizada variou de acordo com o critério utilizado.

4.1.1 - Análise com esquema implícito tradicional

Os gráficos mostrados a seguir serão tomados como base para a comparação entre o "solver" direto e os critérios utilizados para a tolerância variável do MGCP. O gráfico da figura 4.5 apresenta a variação do número de iterações de Newton-Raphson ao longo do tempo de simulação.

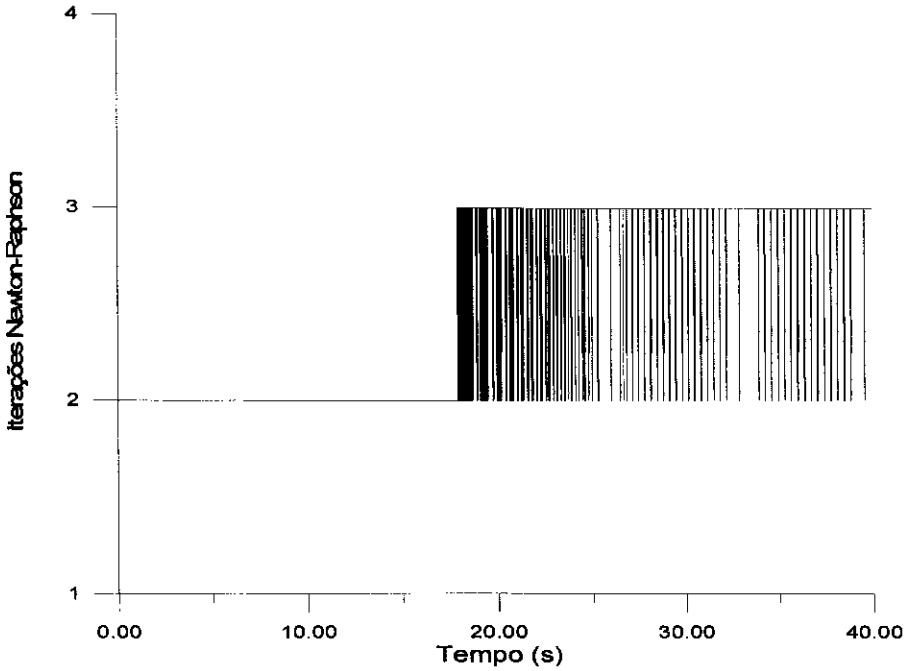


Fig. 4.5 - Iterações Newton Raphson X tempo

Já no gráfico da fig. 4.6 temos a variação da norma do resíduo efetivo também ao longo do tempo. Devemos notar que é importante a plotagem desta variável visto que a mesma influencia diretamente no cálculo da tolerância MGCP segundo os critérios apresentados anteriormente. O valor da norma é mostrado para cada iteração NR, onde na primeira iteração, por estarmos mais longe do equilíbrio e portanto da convergência temos maiores valores de resíduo; à medida em que a convergência se aproxima os valores do resíduo diminuem consideravelmente.

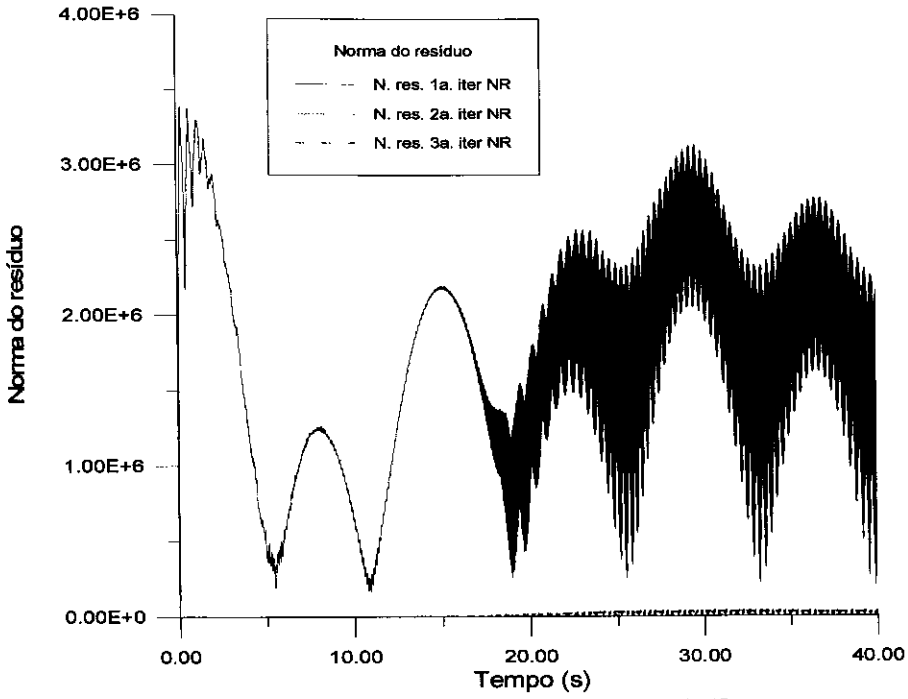


Fig. 4.6 - Norma do resíduo (em cada iter. NR) X tempo

A figura 4.7 apresenta o gráfico da razão entre a norma do resíduo da iteração NR correspondente e a norma do resíduo da primeira iteração NR.

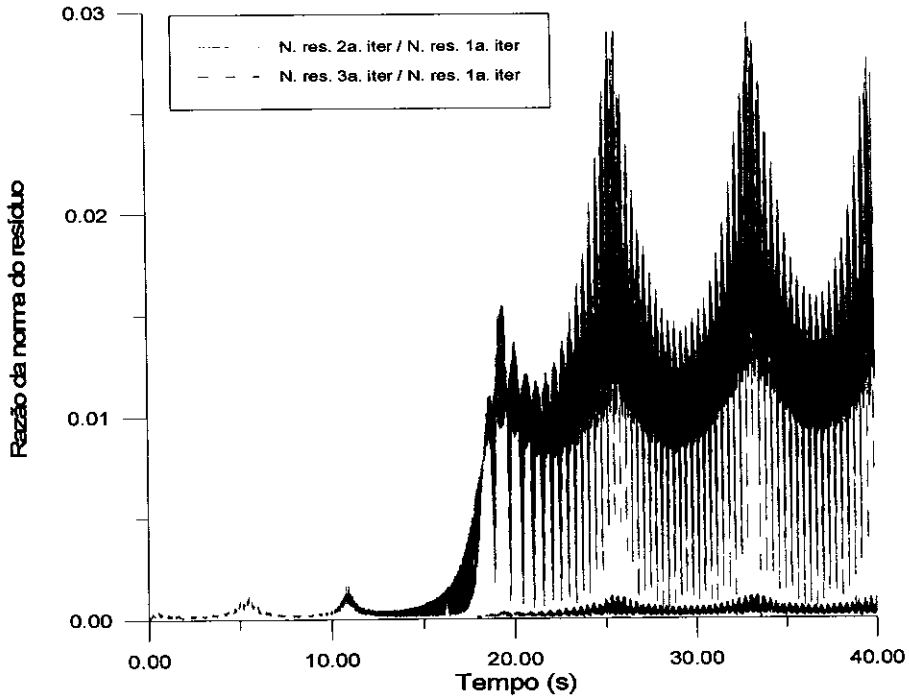


Fig. 4.7 - Razão da norma do resíduo (em cada iter NR) X tempo

4.1.2 - Análises com o esquema semi-implícito proposto

4.1.2.1 - Critério de Papadrakakis

No critério proposto por Papadrakakis, o expoente t (eq(30)) mais eficiente, para o exemplo em questão, obtido do estudo paramétrico no item 3.3 foi igual a 0.7.

Comparando o gráfico da fig. 4.8 com o gráfico da figura 4.5, observamos que com "solver" direto foram necessárias menos iterações NR para convergência do que o número de iterações no critério de "solver" iterativo com tolerância variável proposto por Papadrakakis. Este comportamento era esperado, já que no esquema com o MGCP a precisão na obtenção da resolução do sistema efetivo nas primeiras iterações NR é menor.

O importante é que este aumento no número de iterações de NR (e portanto de soluções do MGCP) será compensado pela redução do custo de cada solução.

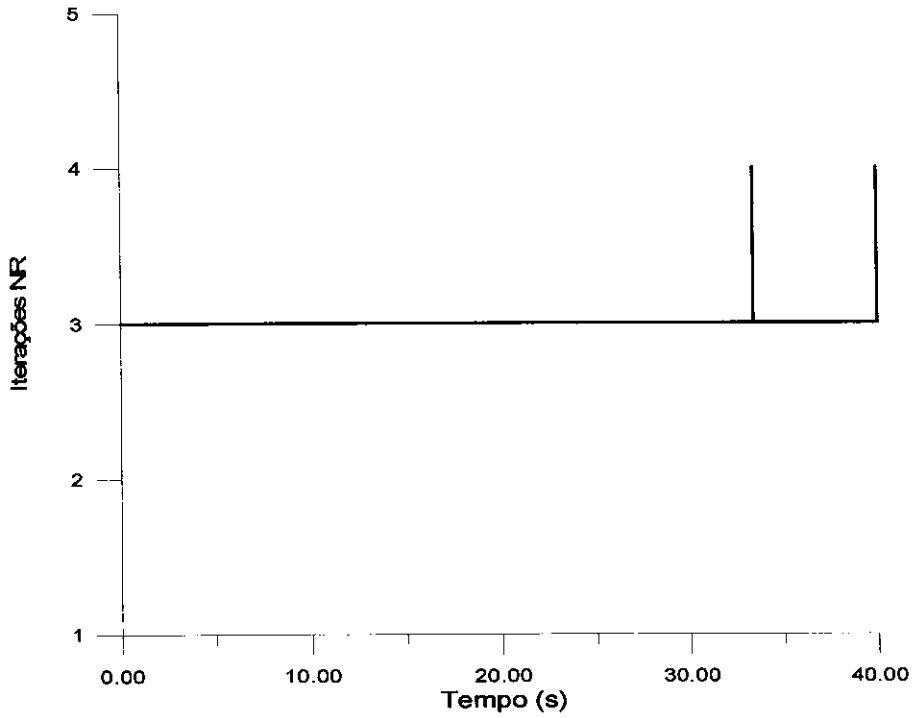


Fig. 4.8 - Iterações Newton-Raphson X tempo

A seguir temos, nas figuras 4.9 e 4.10 a norma do resíduo efetivo em cada iteração NR e a razão da norma do resíduo.

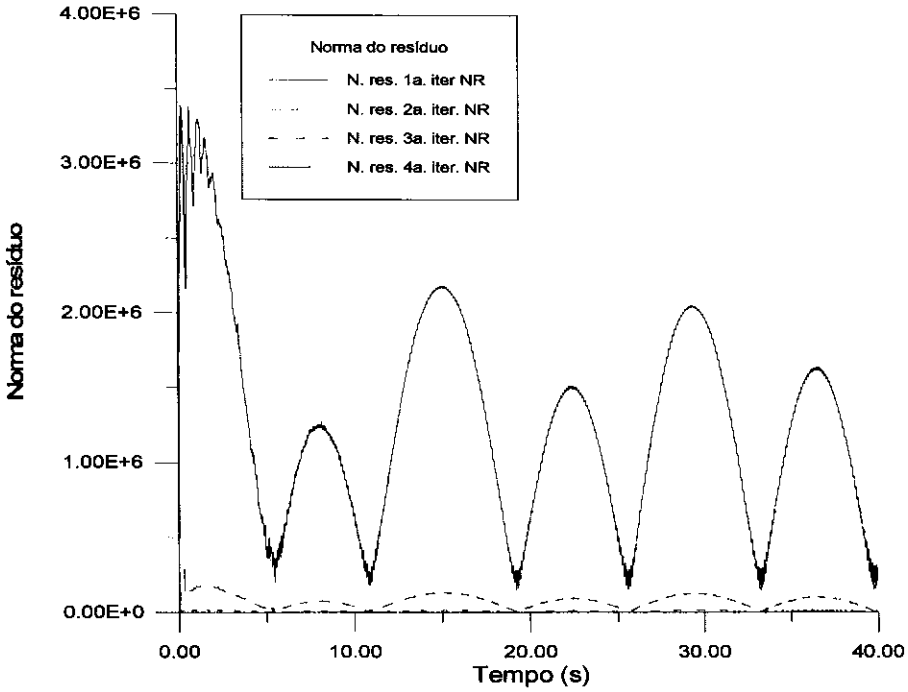


Fig. 4.9 - Norma do resíduo (em cada iter. NR) X tempo

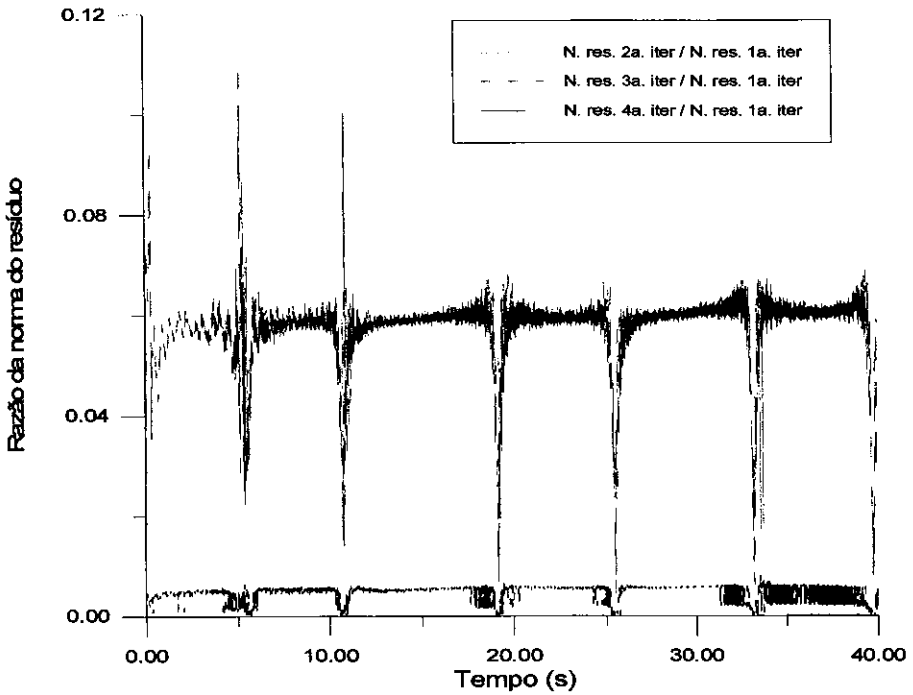


Fig. 4.10 - Razão da norma do resíduo (em cada iter NR) X tempo

Na fig. 4.11 observamos a variação da tolerância MGCP para cada iteração NR ao longo do tempo. Para a primeira iteração devemos notar que o valor assumido foi o limite máximo proposto de 0.20. Já a partir da segunda iteração NR a tolerância passa a assumir valores calculados a partir da razão da norma do resíduo, portanto com um comportamento semelhante ao da fig. 4.10.

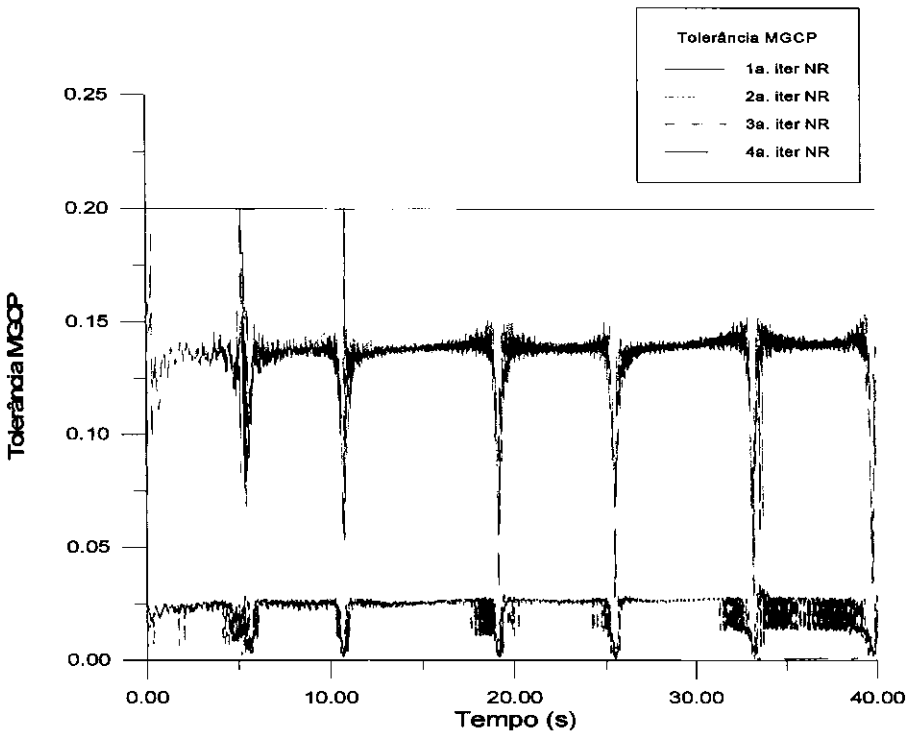


Fig. 4.11 - Variação da tolerância MGCP (em cada iter NR) X tempo

Na fig. 4.12 temos a variação das iterações MGCP também para cada iteração NR ao longo do tempo. Nas primeiras iterações NR notamos que, como temos valores maiores de tolerância para convergência do MGCP, temos menos iterações MGCP com economia de custo. À medida que as iterações NR avançam os valores de tolerância MGCP diminuem, levando a um maior número de iterações para convergência. Em consequência, espera-se alcançar a precisão desejada na resposta.

Em qualquer caso é notável o baixo número de iterações MGCP, o que confirma a suposição de que a matriz efetiva que resulta da aplicação do Método de Newmark é bem condicionada.

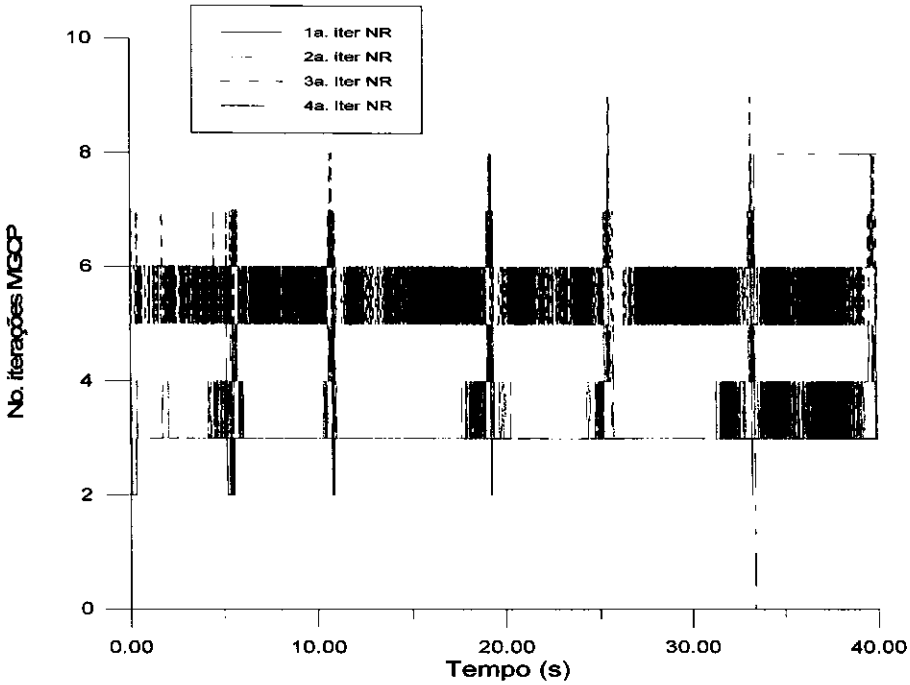


Fig. 4.12 - Número de iterações MGCP (em cada iter NR) X tempo

4.1.2.2 - Critério de Dembo / Steinhaug

Apresentamos os resultados obtidos utilizando-se o critério proposto por Dembo / Steinhaug (eq. (28)). Nos gráficos das figuras 4.13 à 4.18 podemos observar melhor o comportamento deste critério.

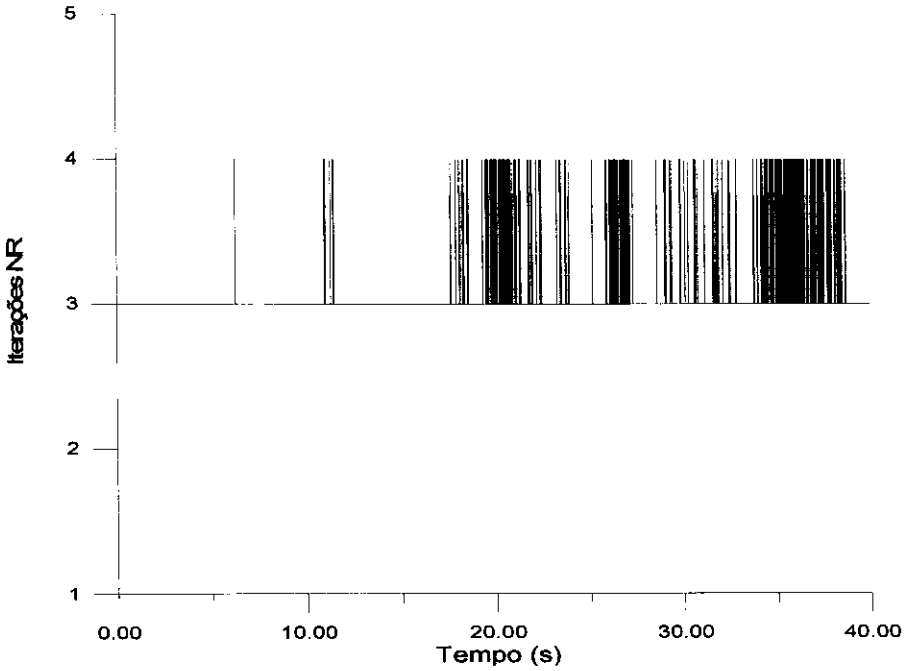


Fig. 4.13 - Iterações Newton-Raphson X tempo

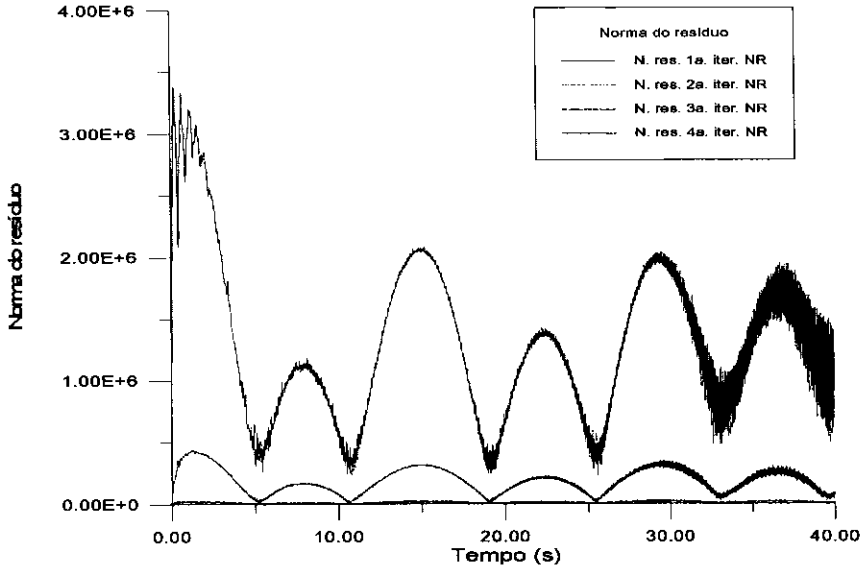


Fig. 4.14 - Norma do resíduo (em cada iter. NR) X tempo

Conforme comentado no item 3.3 verificamos, na fig. 4.15, que os valores de tolerância MGCP utilizados serão sempre obtidos do inverso do contador de iterações NR (1/(k+1)) devido aos altos valores da norma do resíduo.

Um fato importante observado foi o de que nos critérios que utilizam a razão da norma do resíduo no cálculo da tolerância MGCP, obteve-se um comportamento melhor do que no critério agora apresentado, que utiliza os valores da norma do resíduo. Isto ocorre devido ao melhor comportamento de uma razão entre dois números, que envolverá sempre valores relativos, o que não ocorre com valores de norma do resíduo, visto que estaremos lidando com valores absolutos que dependem do problema em questão.

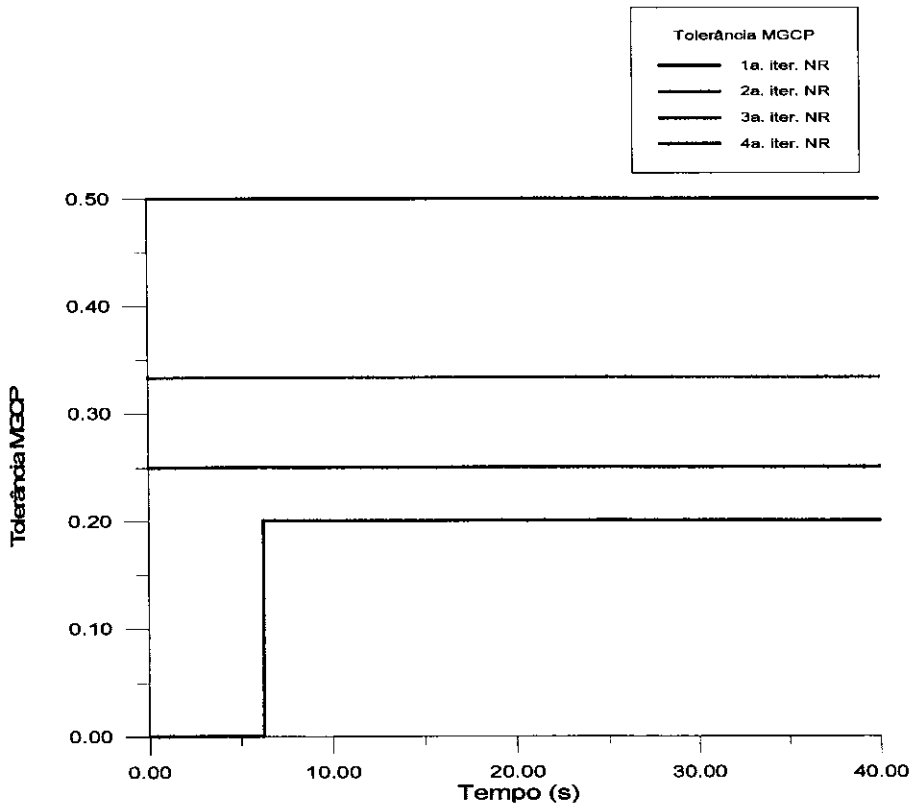


Fig. 4.15 - Variação da tolerância MGCP (em cada iter NR) X tempo

Devido aos maiores valores de tolerância utilizados por este critério, notamos na figura 4.16 os baixos valores do número de iterações MGCP.

Observando as figuras 4.17 e 4.18 notamos que devido aos maiores valores de tolerância MGCP e, consequentemente, ao menor número de iterações MGCP necessários para convergência, obtivemos problemas de precisão na resposta, o que tornou este critério inviável e serviu como motivação para propormos a variação descrita no item 4.1.2.3

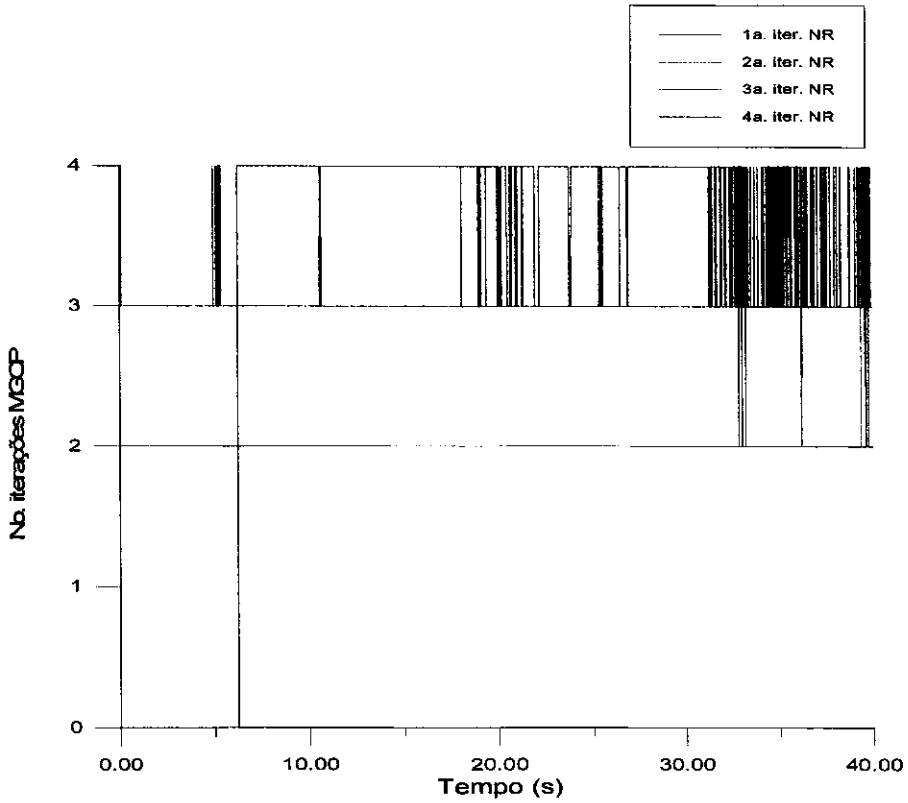


Fig. 4.16 - Número de iterações MGCP (em cada iter NR) X tempo

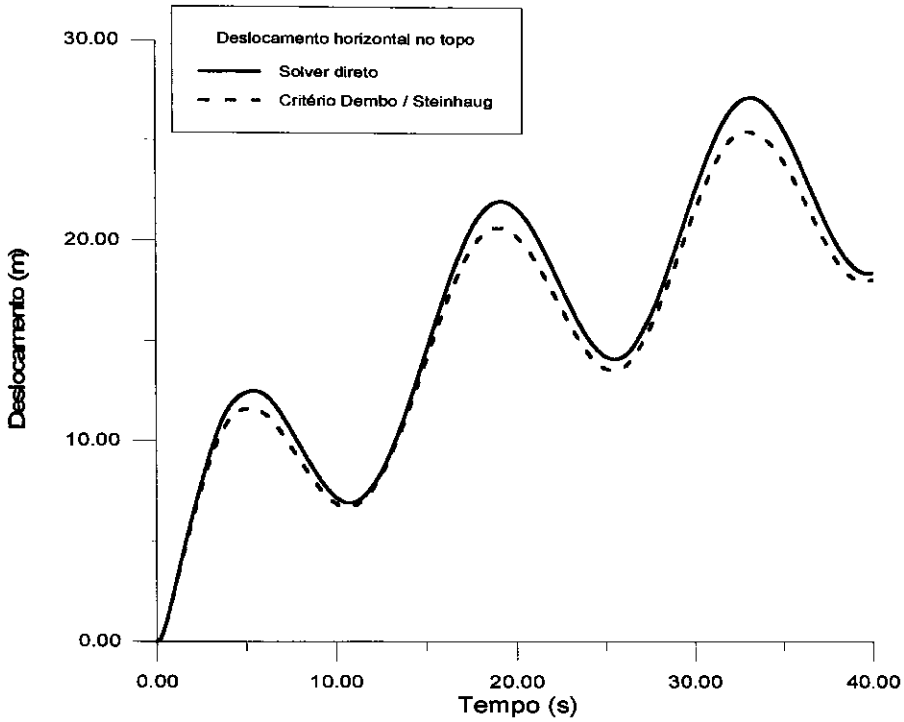


Fig. 4.17 - Deslocamento horizontal no topo da estrutura

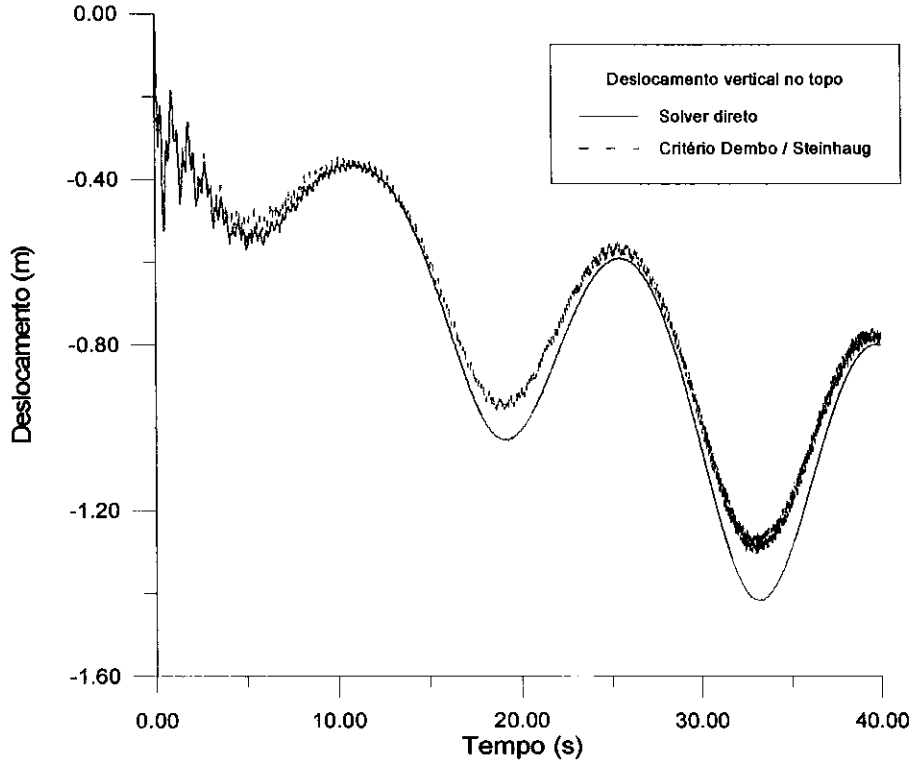


Fig. 4.18 - Deslocamento vertical no topo da estrutura

4.1.2.3 - Critério de Dembo / Steinhaug (modificado)

Neste sub-item apresentam-se os resultados obtidos com a utilização do critério Dembo / Steinhaug modificado. Para este critério o valor mais indicado para o coeficiente η_0 (eq. (31)) foi de 0.5 (com este valor obtivemos precisão e o máximo de economia computacional possível). Os resultados são apresentados em gráficos nas figuras 4.19 à 4.23, que seguem a mesma sequência do sub-item anterior.

Podemos observar que os resultados são muito próximos aos obtidos com o critério de Papadrakakis, visto que a única diferença entre ambos está na seleção do limite máximo para a tolerância do MGCP. Segundo Papadrakakis, este limite é dado pelo valor de η_0 fornecido pelo usuário e, constante durante toda a análise (eq. (30)). Já no critério de Dembo / Steinhaug modificado, o limite máximo é obtido tomando-se o inverso do contador de iterações NR (eq. (31)).

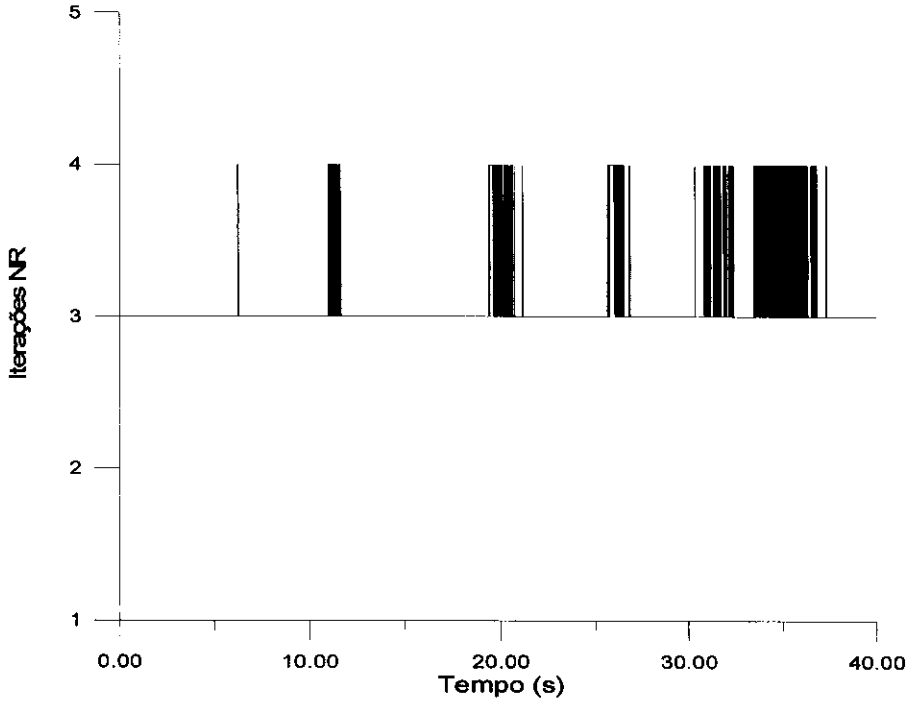


Fig. 4.19 - Iterações Newton-Raphson X tempo

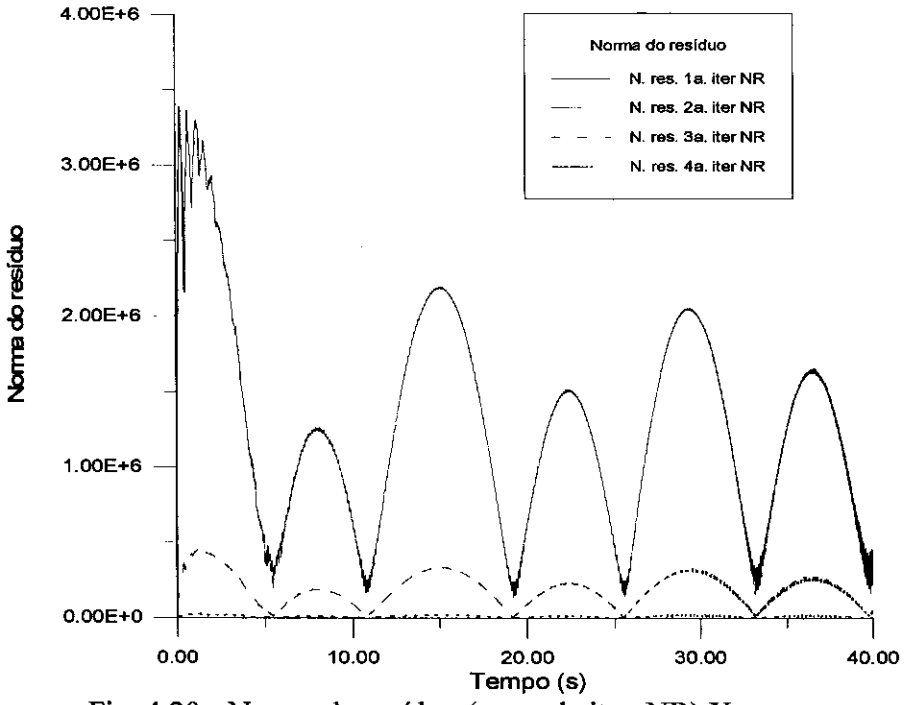


Fig. 4.20 - Norma do resíduo (em cada iter. NR) X tempo

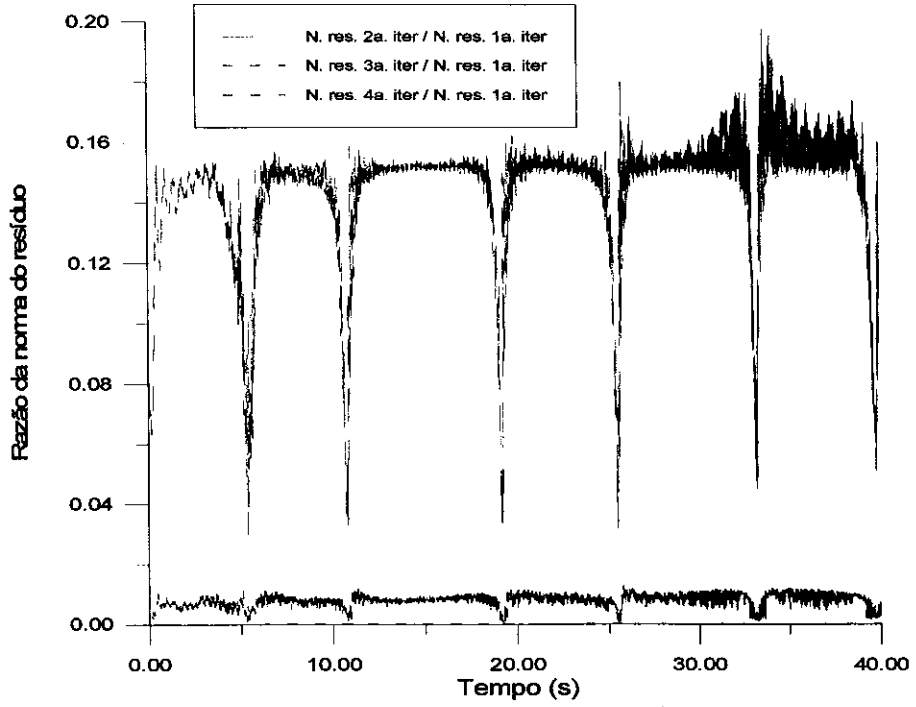


Fig. 4.21 - Razão da norma do resíduo (em cada iter NR) X tempo

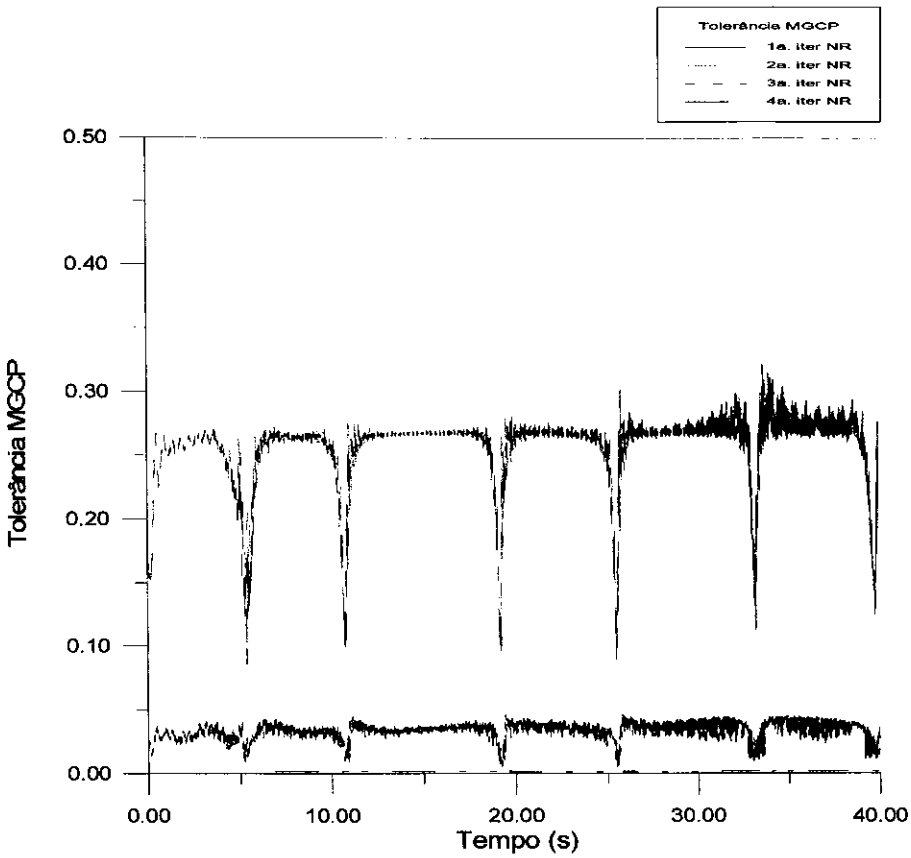


Fig. 4.22 - Variação da tolerância MGCP (em cada iter NR) X tempo

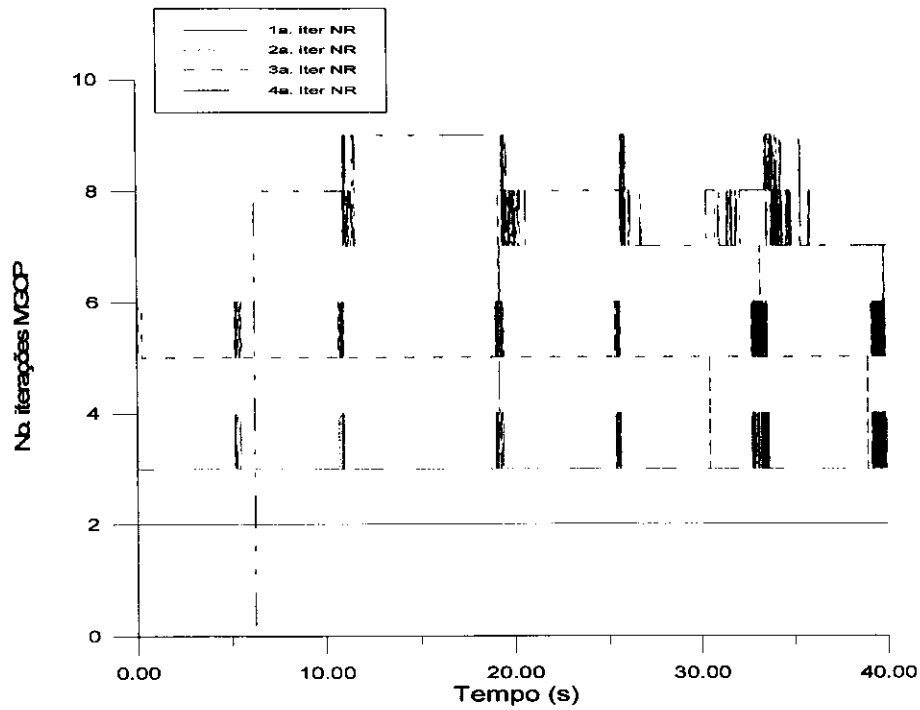


Fig. 4.23 - Número de iterações MGCP (em cada iter NR) X tempo

O desempenho deste e dos demais critérios comparados entre si, está na tabela 4.1 no item 4.1.3.

4.1.2.4 - Critério de Nour-Omid

No critério proposto por Nour-Omid observou-se que, devido ao expoente utilizado ser igual a 1.5, a tolerância MGCP diminui muito já na 2ª iteração NR, retirando assim a vantagem semi-implícita do método, sem ganho em termos de esforço computacional, surgindo assim a motivação proposta para este critério.

Para o critério proposto por Nour-Omid temos, a seguir, os gráficos suas características nas figuras 4.24 à 4.28.

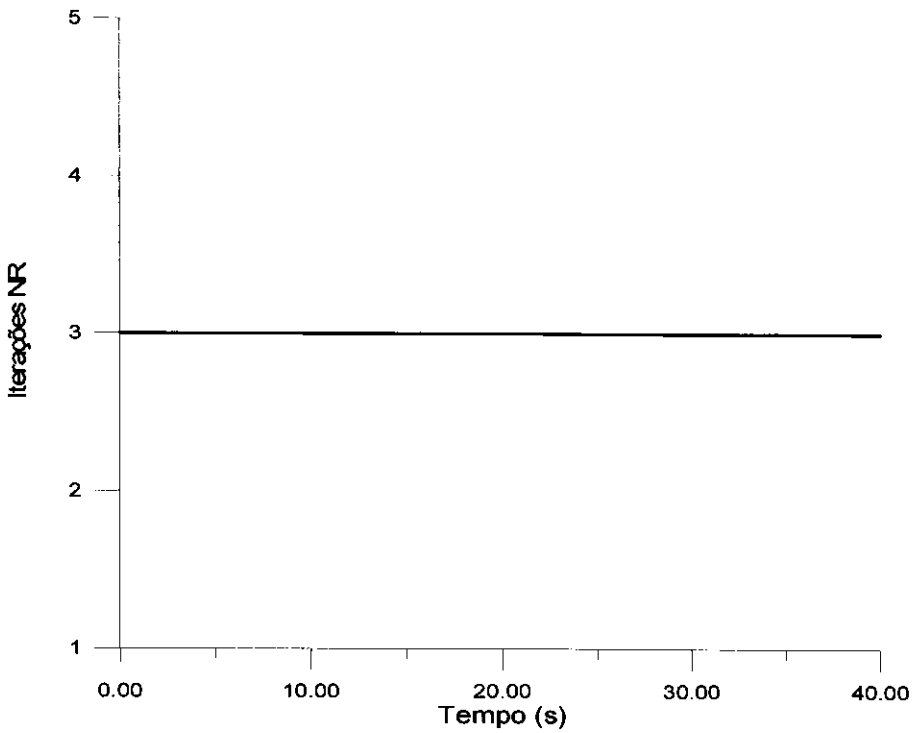


Fig. 4.24 - Iterações Newton-Raphson X tempo

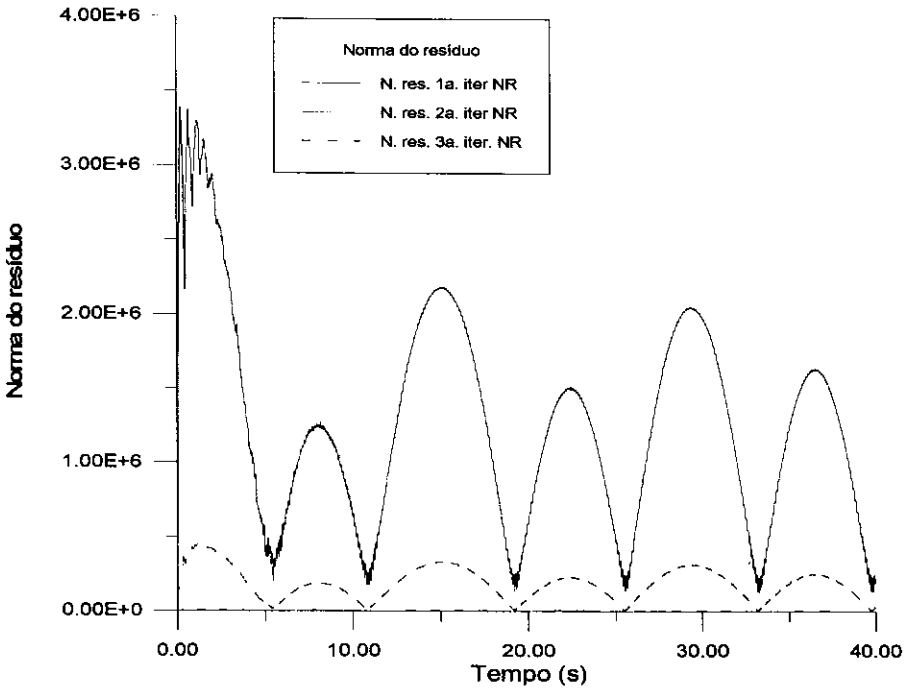


Fig. 4.25 - Norma do resíduo (em cada iter. NR) X tempo

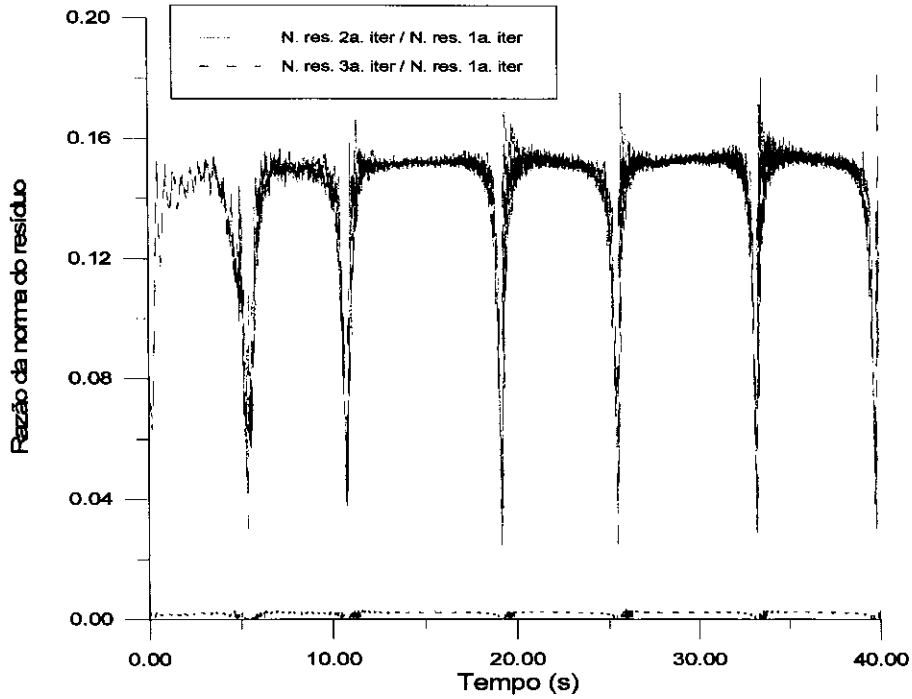


Fig. 4.26 - Razão da norma do resíduo (em cada iter NR) X tempo

Notamos que no gráfico da figura 4.27 a tolerância MGCP já na segunda iteração NR está próxima de zero fazendo com que o número de iterações seja maior (fig. 4.28) e

a solução do sistema de equações de forma iterativa apresente um maior esforço computacional sem ganhos significativos de precisão, como poderá ser observado na tabela 4.1.

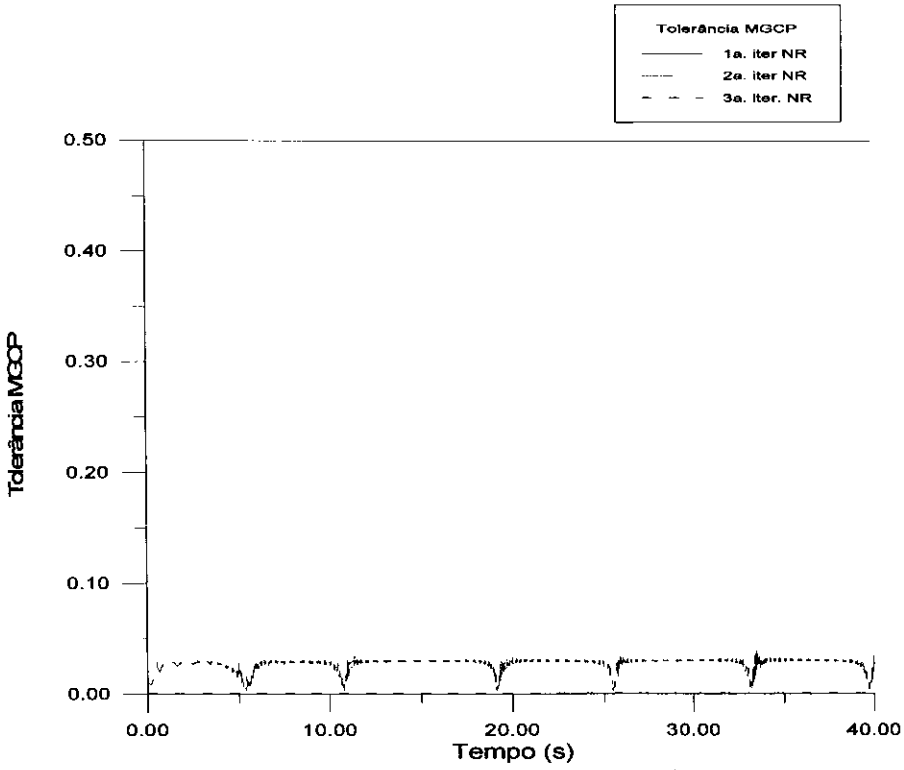


Fig. 4.27 - Variação da tolerância MGCP (em cada iter NR) X tempo

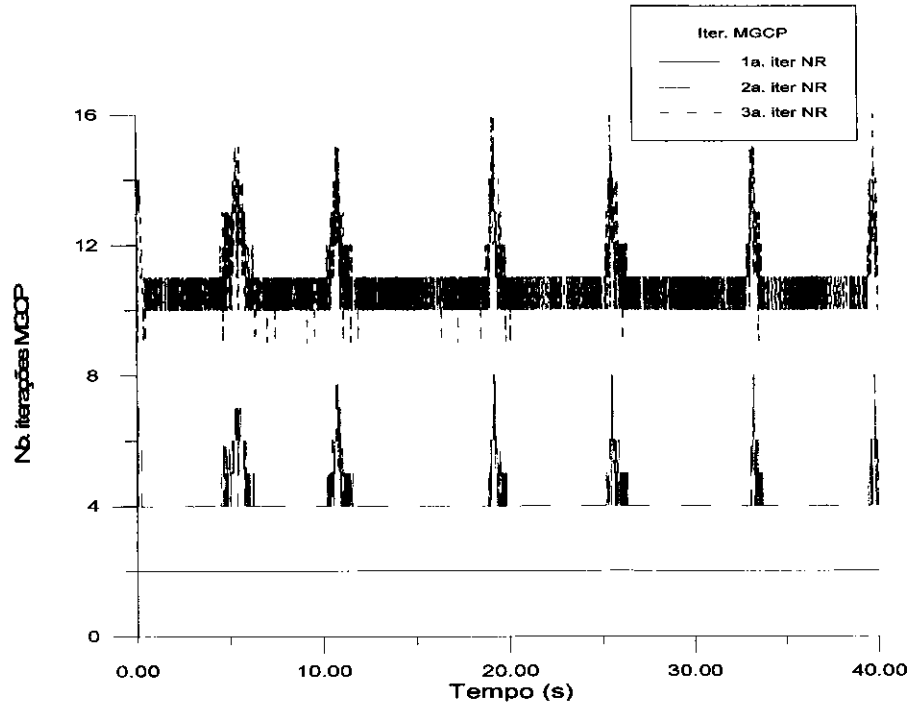


Fig. 4.28 - Número de iterações MGCP (em cada iter NR) X tempo

4.1.2.5 - Critério de Nour-Omid modificado (expoente t = 1)

Com o objetivo de aumentar o ganho em tempo de CPU, que não foi possível com o critério de Nour-Omid em sua formulação original, é proposto a variação do expoente de 1.5 para 1. Com isto a tolerância do MGCP mantém-se com valores mais "relaxados" na 2ª iteração NR.

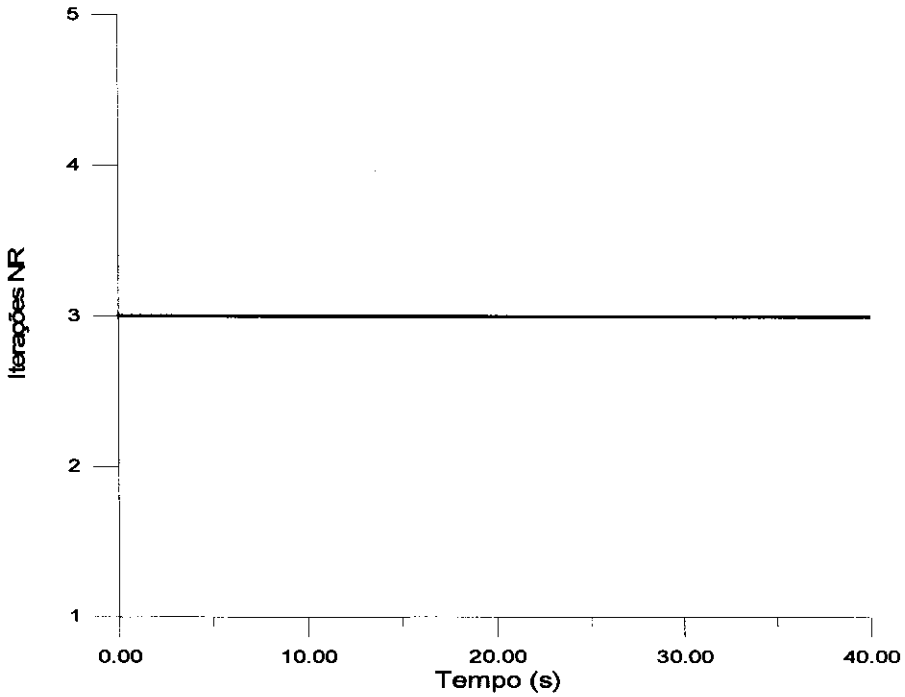


Fig. 4.29 - Iterações Newton-Raphson X tempo

Apesar das variações propostas no método devemos mencionar que os deslocamentos obtidos foram sempre tão precisos quanto os obtidos pelos métodos tradicionais de análise ("solver" direto).

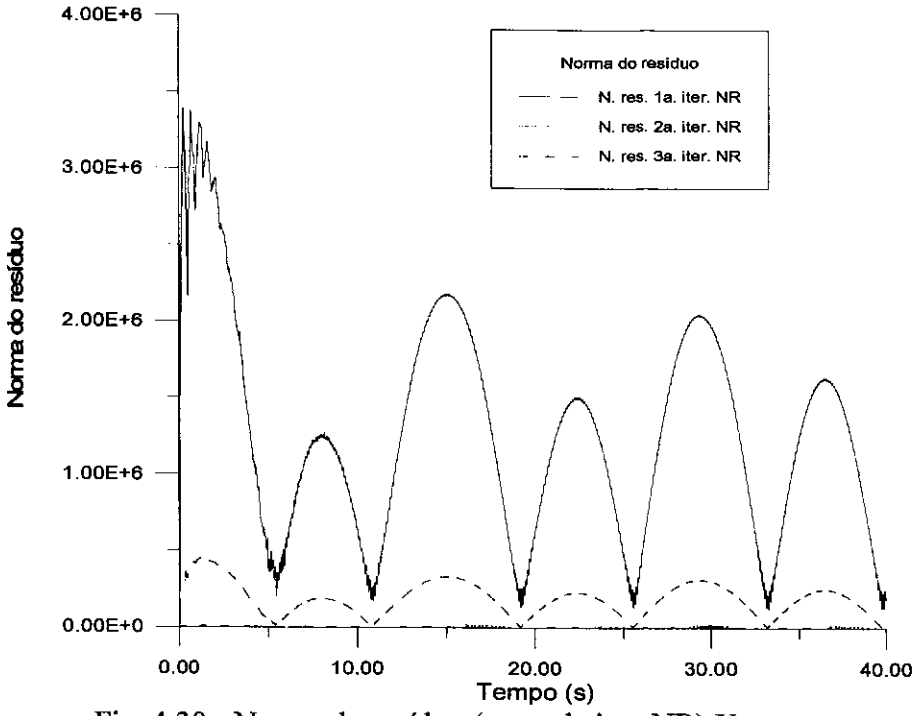


Fig. 4.30 - Norma do resíduo (em cada iter. NR) X tempo

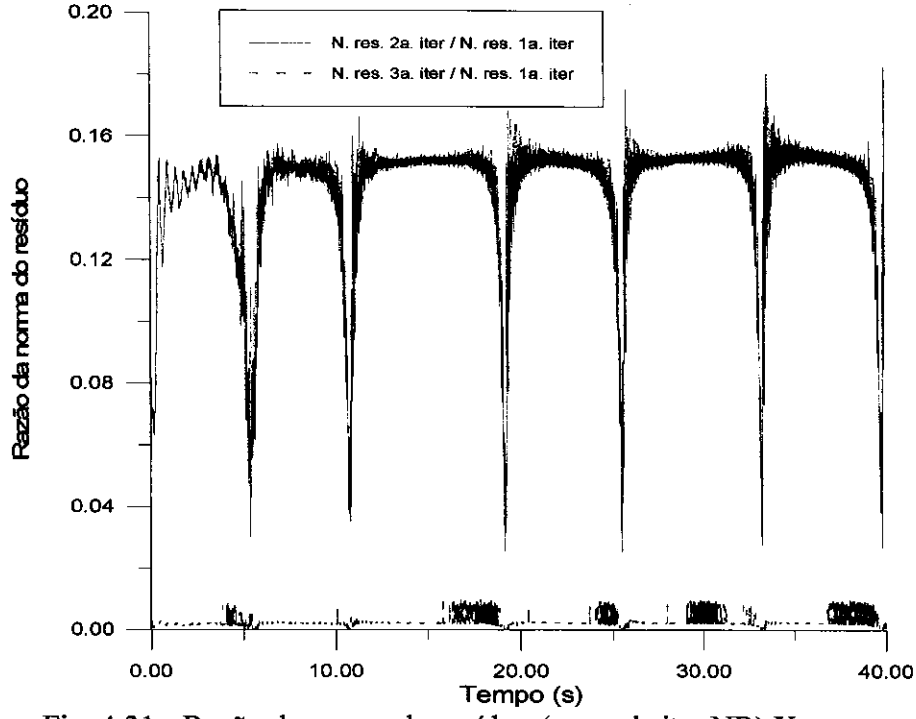


Fig. 4.31 - Razão da norma do resíduo (em cada iter NR) X tempo

Podemos notar agora que os valores de tolerância MGCP estão maiores que os obtidos anteriormente retomando assim o problema características semi-implícitas.

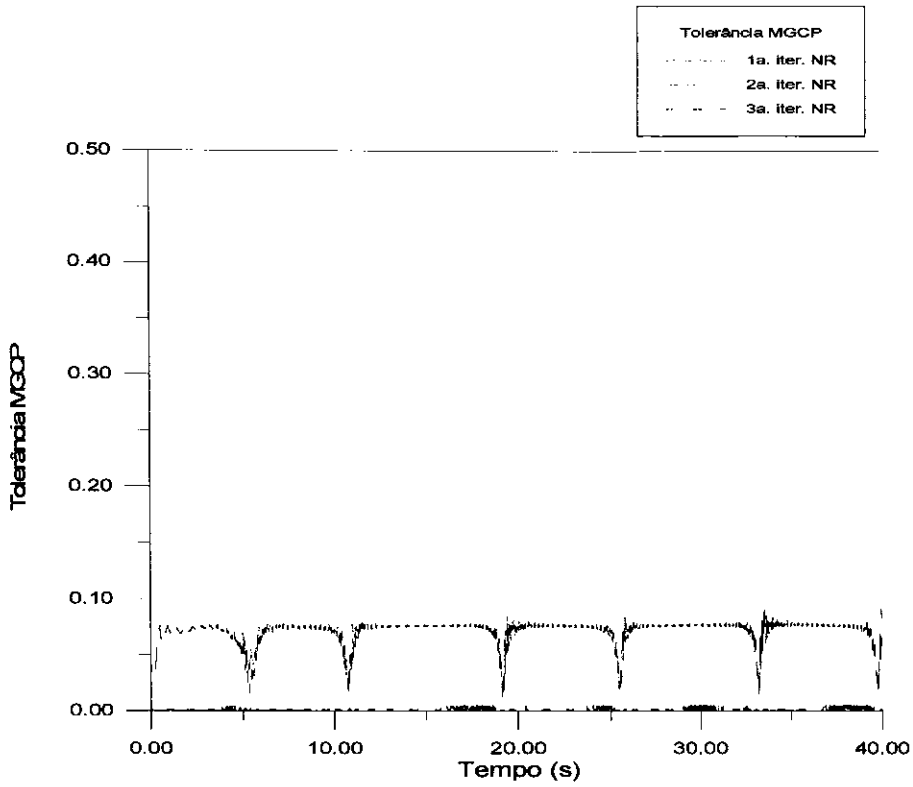


Fig. 4.32 - Variação da tolerância MGCP (em cada iter NR) X tempo

Conforme esperado, com maiores valores de tolerância MGCP obtivemos menos iterações para convergência do "solver" iterativo (fig. 4.33).

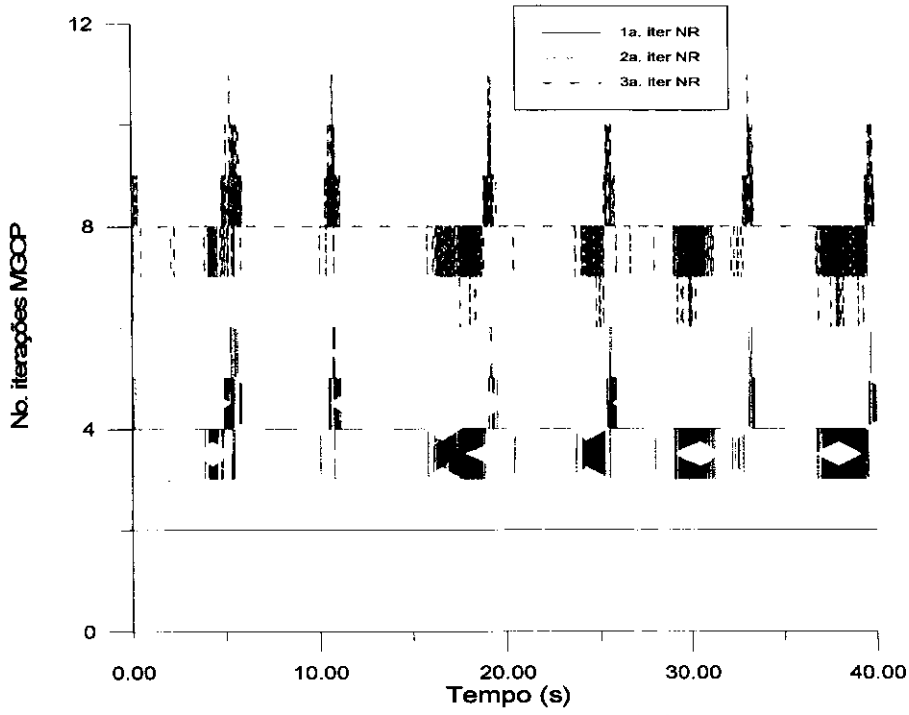


Fig. 4.33 - Número de iterações MGCP (em cada iter NR) X tempo

4.1.2.6 - Critério de Nour-Omid modificado (expoente t = 0.7)

Com o objetivo de se obter mais vantagens em termos de esforço computacional é proposto ainda a variação do expoente para 0.7. Com isto temos a tolerância MGCP decaindo mais lentamente à medida que a norma do resíduo diminui. Podemos notar com isto, na fig. 4.34 o aumento do número de iterações NR em relação aos dois sub-ítens anteriores.

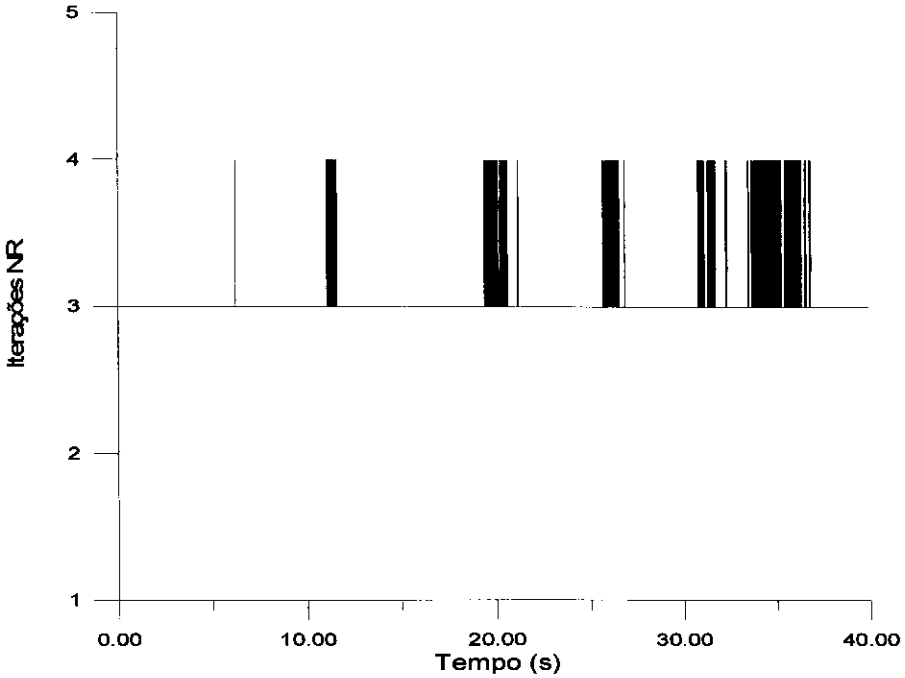


Fig. 4.34 - Iterações Newton-Raphson X tempo

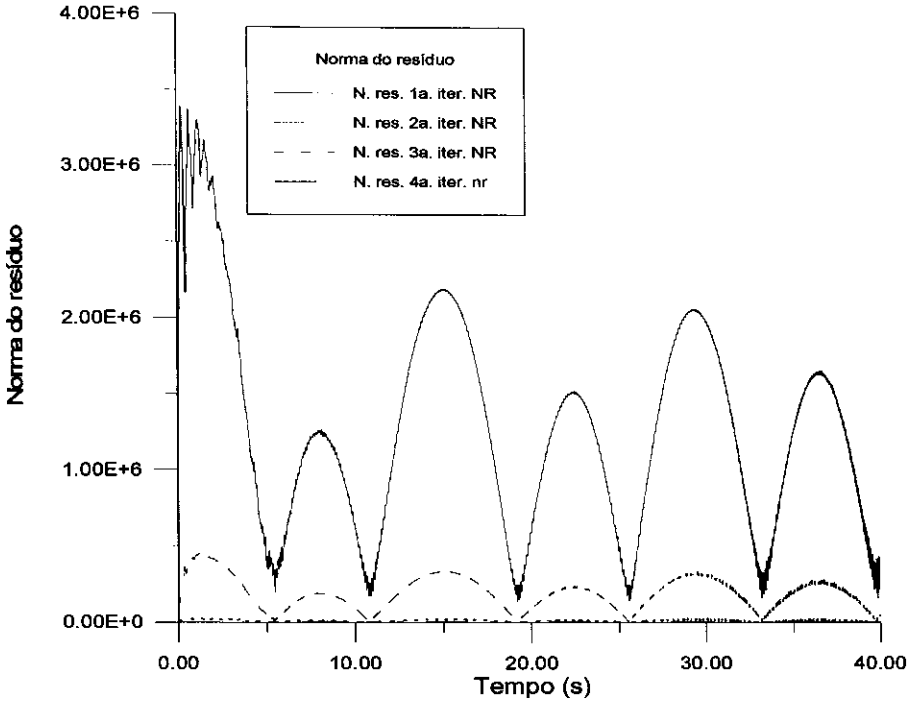


Fig. 4.35 - Norma do resíduo (em cada iter. NR) X tempo

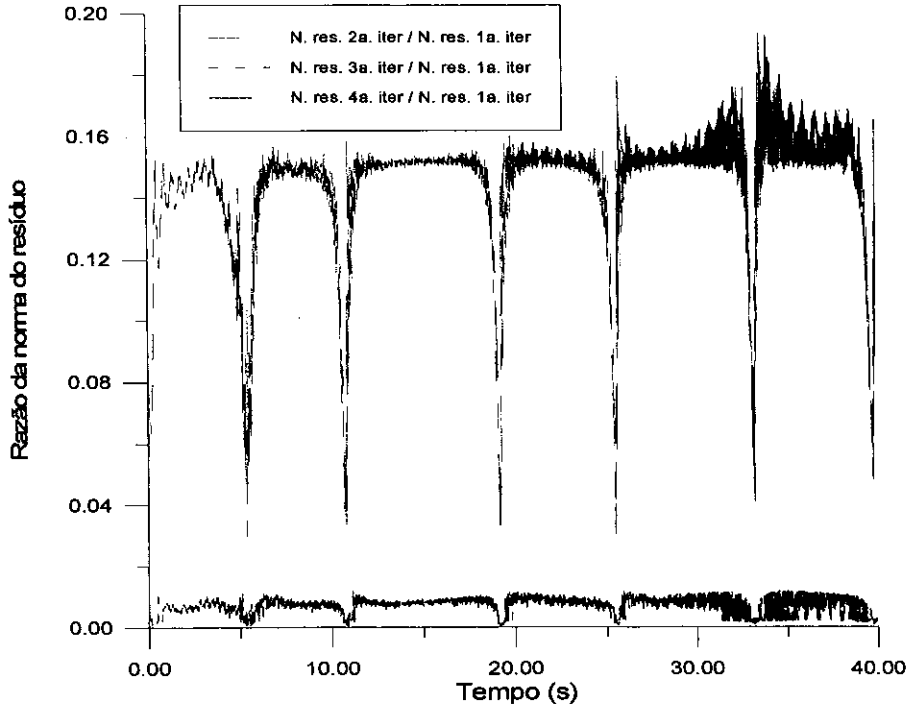


Fig. 4.36 - Razão da norma do resíduo (em cada iter NR) X tempo

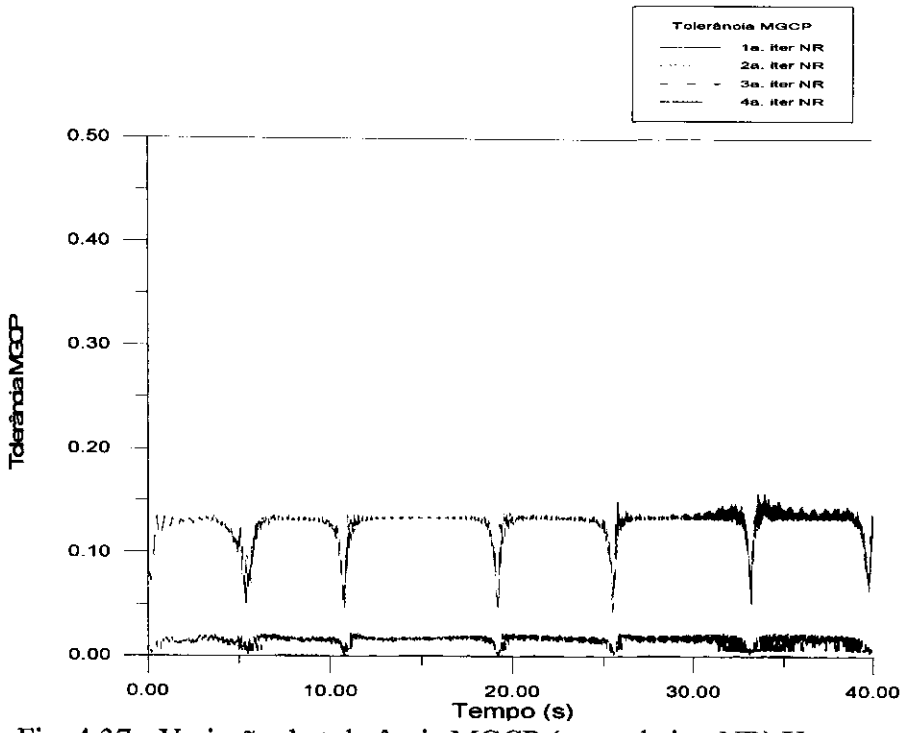


Fig. 4.37 - Variação da tolerância MGCP (em cada iter NR) X tempo

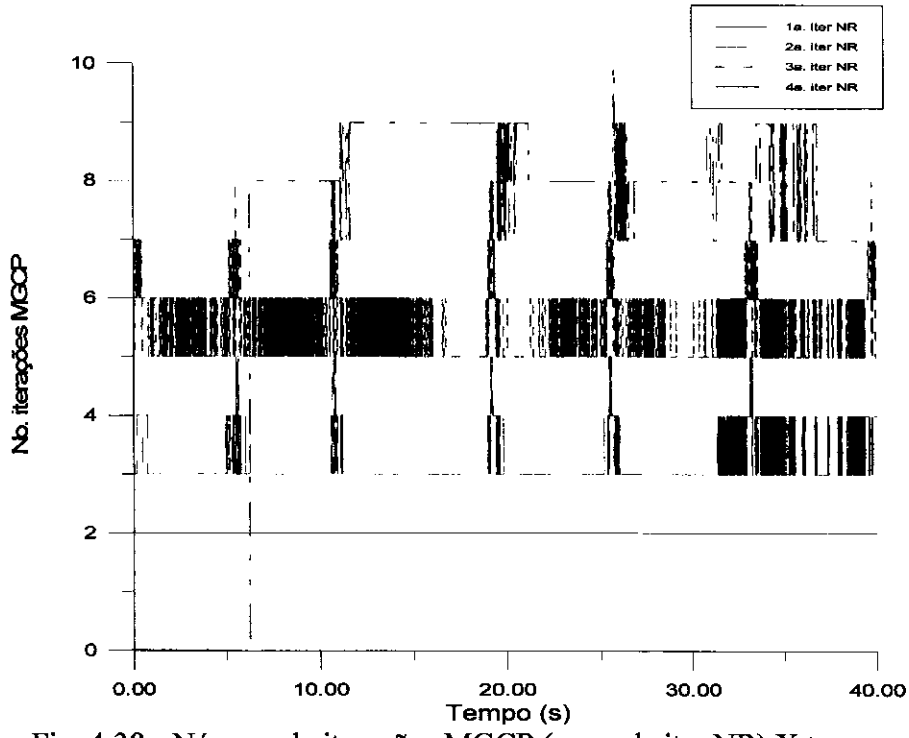


Fig. 4.38 - Número de iterações MGCP (em cada iter NR) X tempo

4.1.3 - Comparação do desempenho computacional

Na tabela 4.1 temos a comparação entre o tempo total de CPU gasto com a utilização de cada esquema de solução apresentado. Deve-se chamar a atenção ao fato de que em todos os critérios analisados a qualidade dos resultados se manteve.

Devemos notar que o tempo total de CPU dispendido na análise quando se utilizou o MGCP foi sempre menor que o tempo gasto utilizando-se "solver" direto, com exceção do critério proposto por Nour-Omid sem variações. Outra observação importante é que o expoente mínimo utilizado que nos forneceu resultados com precisão satisfatória foi $t = 0.7$.

A tabela apresenta também o custo de uma solução MGCP expresso em termos de um "número médio de iterações MGCP". Este valor é obtido dividindo-se o número total de iterações MGCP pelo número total de iterações NR, acumulados ao longo de toda a análise, conforme citado no item 3.3.

Estes valores podem ser comparados com o custo de uma solução direta expresso em termos de um "número equivalente de iterações MGCP". Para obter este número dividimos o tempo de CPU da solução direta, que é constante para esta aplicação, pelo custo de uma iteração MGCP, o qual, por sua vez, é obtido dividindo-se o tempo total de CPU correspondente às soluções MGCP pelo número total de iterações MGCP.

Com isso observamos que, em qualquer circunstância, a solução direta é mais cara; o esquema com o critério de Nour-Omid perde para o esquema implícito tradicional por não ser capaz de reduzir o número de iterações do MGCP em cada solução de modo a compensar o aumento do número de iterações de NR (e portanto o número de soluções de sistemas de equações pelo MGCP).

Tabela 4.1 - Comparação entre os Métodos

	Tempo(s)	Iterativo / Direto	PCG / NR*
Direto	5650	1.00	8.94+
Papadrakakis	5180	0.92	3.93
Nour (t = 1.5)	6220	1.10	5.65
Nour (t = 1)	5450	0.96	4.64
Nour (t = 0.7)	5250	0.93	3.57
Dembo (mod)	5310	0.94	3.47

* - Número médio de iterações MGCP.
+ - Número equivalente de iterações MGCP.

4.1.4 - Análise com esquema auto-adaptativo

Vamos agora demonstrar a aplicação do esquema auto-adaptativo para a seleção do método de resolução do sistema de equações mais vantajoso. Analisando-se a torre complacente descrita no item 4.1 faremos, a título de ilustração, gráficos onde a abscissa com o tempo de simulação vai apenas até 1 segundo para que possam ser notados claramente a troca de "solver" e a decisão de utilização daquele que foi, para o problema estudado, mais vantajoso.

São apresentados resultados para duas análises; na primeira utilizou-se intervalo de tempo igual a 0.005s, e na segunda análise um intervalo de tempo igual a 0.01s.

Para a primeira análise, a partir do gráfico da fig. 4.39, pode-se notar que até o instante 0.10 segundos a análise é feita utilizando-se o "solver" direto; de 0.10 seg. até 0.20 seg., quando o programa começa a reportar valores de tolerância MGCP a análise é feita utilizando-se o MGCP com tolerância variável; e a partir do instante 0.20 seg., quando o esquema adaptativo faz a análise de eficiência, podemos observar que o esquema semi-implícito com o MGCP foi o escolhido para ser utilizado até o final da análise, e a tolerância MGCP continua sendo reportada pelo programa.

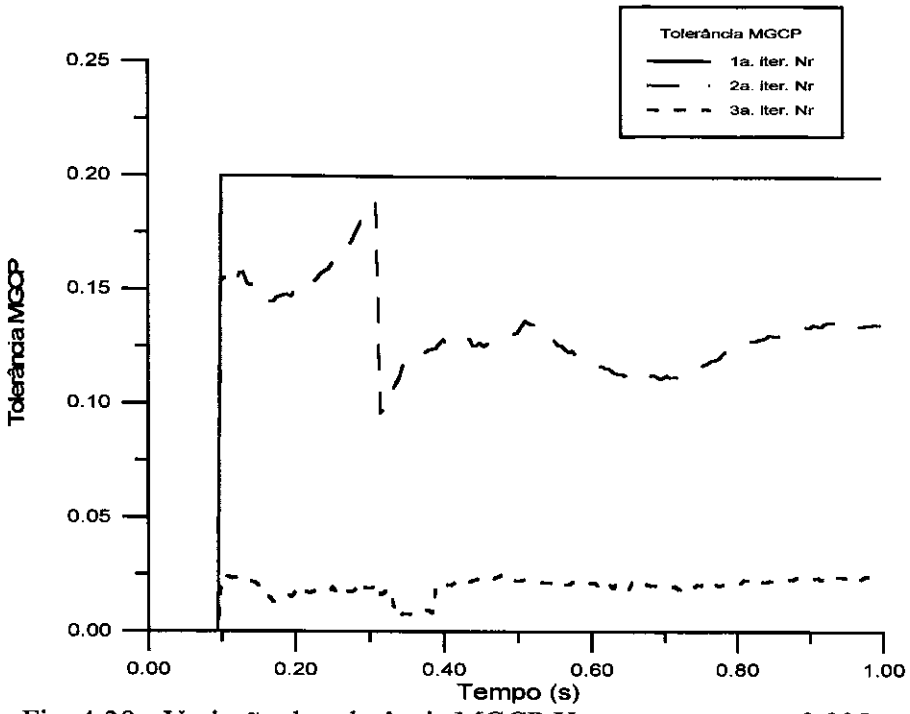


Fig. 4.39 - Variação da tolerância MGCP X tempo - t. step = 0.005 s

O gráfico da figura 4.40 apresenta a variação do número de iterações necessárias para convergência do MGCP.

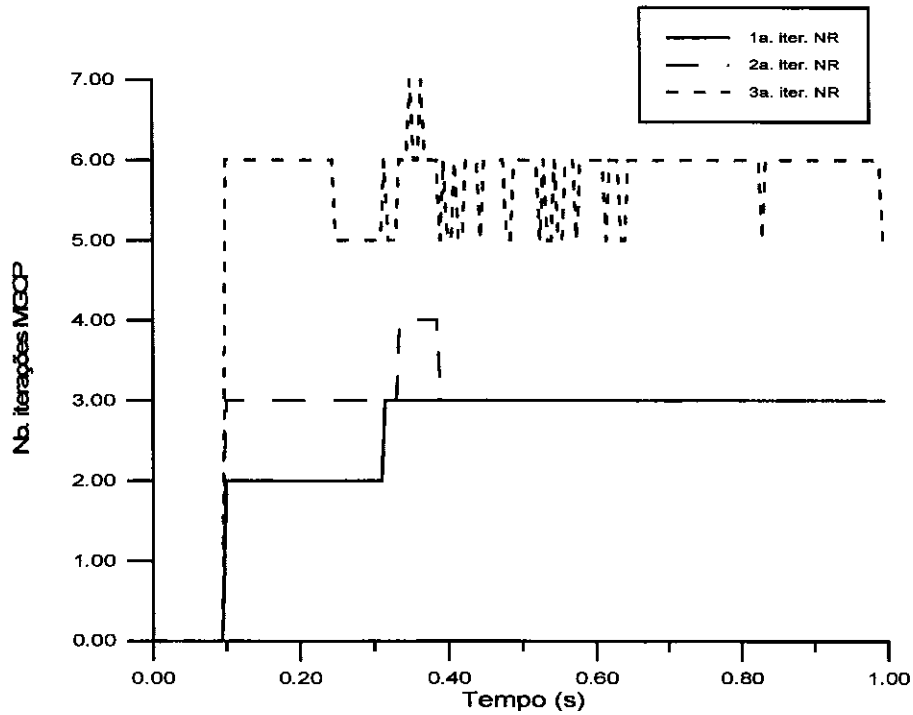


Fig. 4.40 - Número de iterações MGCP X tempo - t. step = 0.005 s

Aumentando o valor do intervalo de tempo de 0.005 seg. para 0.01 seg, obtemos os gráficos apresentados nas figuras 4.41 e 4.42 a seguir.

Neste caso, após a pesquisa efetuada até o instante 0.40 seg podemos observar no gráfico da fig. 4.41 que o algoritmo decidiu pela utilização do solver direto, que neste caso foi o o mais eficiente, visto que a tolerância MGCP pára de ser reportada.

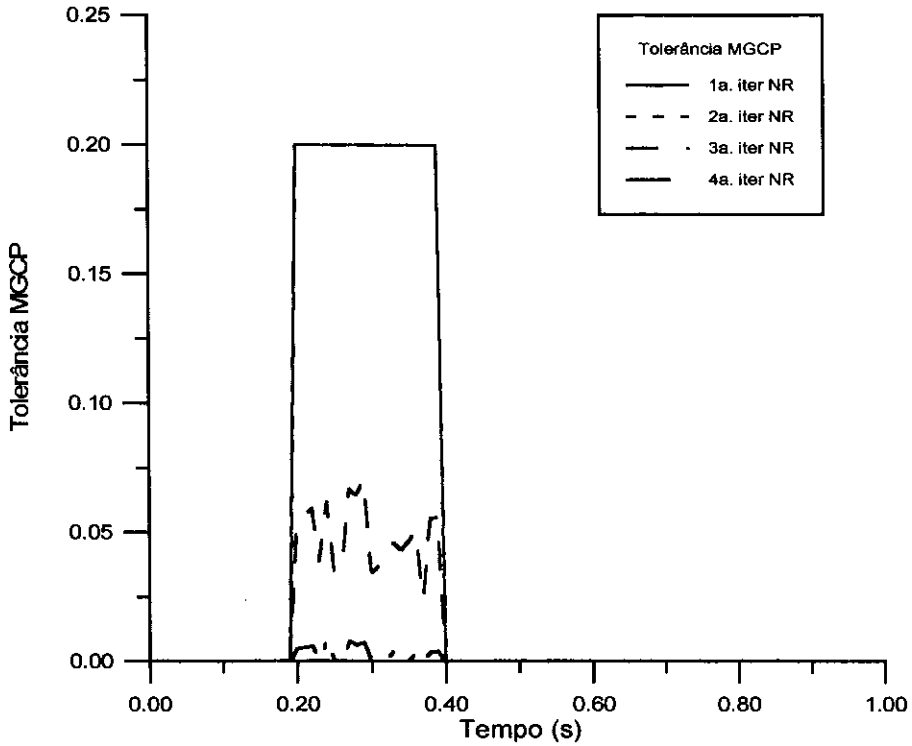


Fig. 4.41 - Variação da tolerância MGCP X tempo - t. step = 0.01 s

Finalmente, o gráfico da fig. 4.42 nos mostra o número de iterações MGCP ao longo do tempo onde o método iterativo foi acionado apenas para a pesquisa que decidiu o tipo de "solver" a ser utilizado.

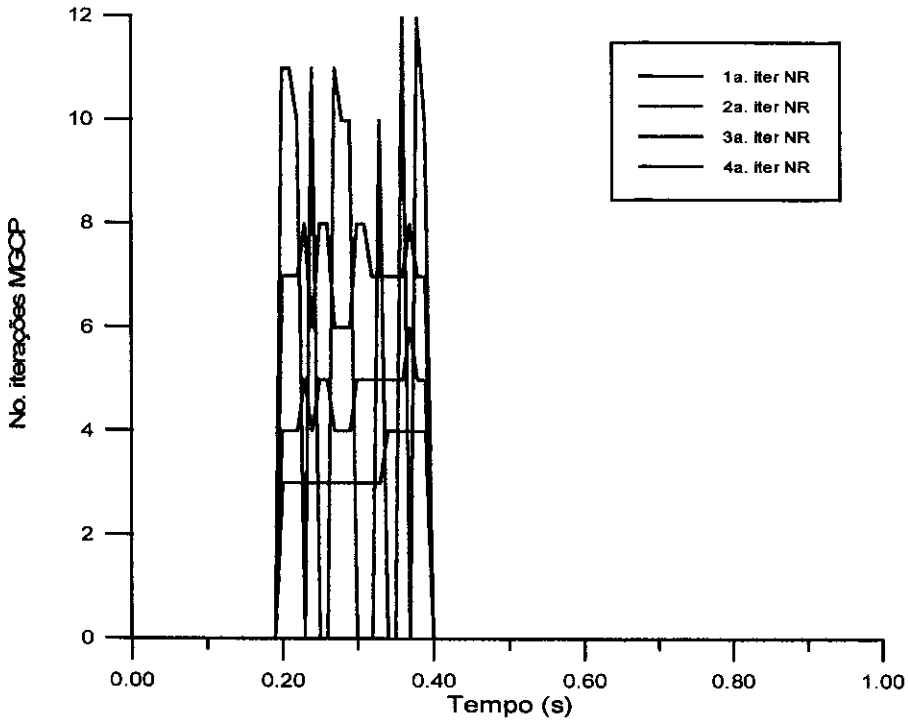


Fig. 4.42 - Número de iterações MGCP X tempo - t. step = 0.01 s

Pode-se notar que conforme já mencionado, para o menor intervalo de tempo, o MGCP com tolerância variável foi o mais econômico em termos de custo computacional, sendo este o "escolhido" e utilizado até o final da análise. Já para o maior intervalo de tempo o mais eficiente e consequentemente o utilizado foi o "solver" direto.

4.2 - Conjunto de risers flexíveis

Risers flexíveis são mangotes especiais empregados em atividades de produção, em plataformas baseadas em sistemas flutuantes tais como plataformas semi-submersíveis ou navios. As principais características de risers flexíveis residem em sua característica mais acentuadamente "complacente", já que podem acompanhar sem problemas todos os movimentos do sistema flutuante, e em permitir a conexão de plataformas a poços mais afastados.

A aplicação considerada neste item consiste em um conjunto de 8 risers flexíveis em catenária livre dispostos simetricamente formando um ângulo de 45° entre elas e concorrendo todas a um mesmo ponto na superfície. A profundidade é de 300m, os risers têm todos as mesmas propriedades: diâmetro externo de 0.155m, interno de 0.102m e comprimento de 430m. Os risers foram modelados com elementos de treliça espacial (num total de 234 elementos e 235 nós), visto que a rigidez à flexão pode ser usualmente desprezada. Para simulação do contato variável com o fundo foram utilizadas molas não-lineares atuando na direção vertical conectadas nos nós próximos ao fundo, com um "gap" igual à distância original ao fundo.

Na figura 4.43 podemos observar, de forma esquemática, o arranjo dos risers flexíveis concorrendo todas a um único ponto situado na superfície.

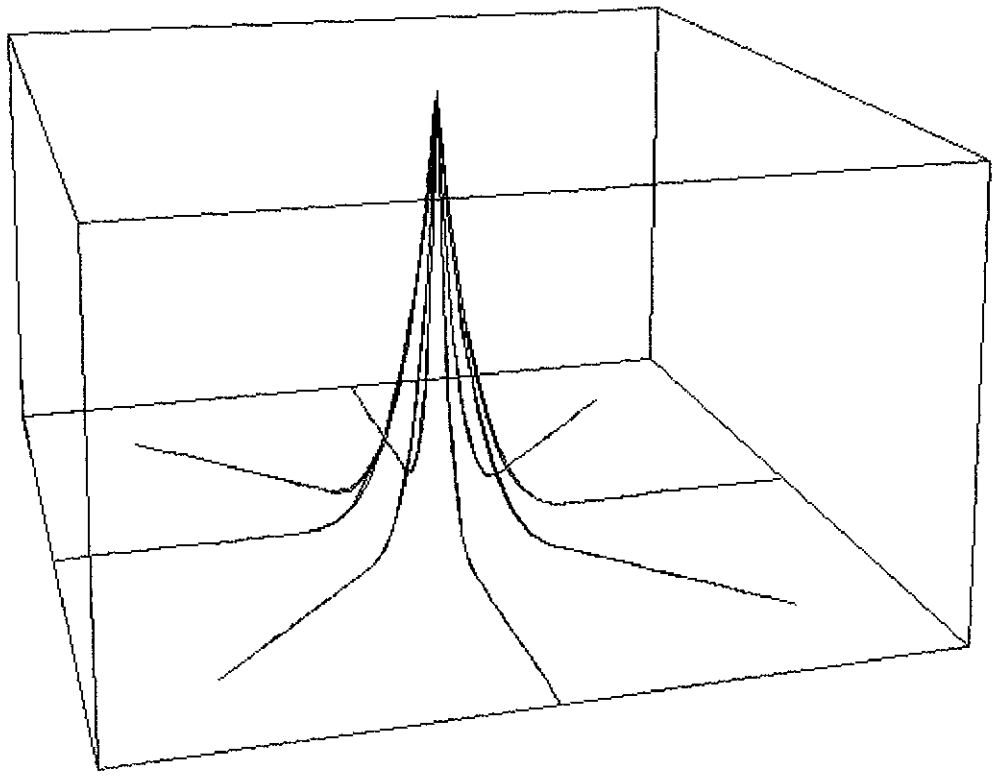


Fig. 4.43 - Disposição geral dos risers flexíveis

O carregamento ambiental aplicado consiste em uma onda determinística com período de 11.1 seg. e altura de onda de 14.17 m e correnteza com velocidade de 2 m/s na superfície.

Para este exemplo utilizamos os seguintes esquemas de solução:

- esquema direto;
- MGCP com tolerância variável seguindo os critérios de variação de tolerância propostos por:
  - Papadrakakis;
  - Nour-Omid com expoente t = 0.7;

visto que estes critérios foram os que forneceram resultados mais vantajosos no exemplo anterior.

Além disso utilizaremos o esquema adaptativo onde, conforme descrito anteriormente, o próprio algoritmo escolhe, através de parâmetros obtidos no início da análise, que tipo de solução de sistemas é a mais apropriada, se a solução direta ou a solução iterativa com tolerância variável (neste caso, utilizou-se apenas o critério proposto por Papadrakakis).

Na análise dinâmica do conjunto de risers utilizou-se:

- intervalo de tempo de 0.02 seg.;
- tolerância de convergência para o Método de Newton-Raphson de 0.01;
- tolerância inicial de convergência do MGCP de 0.3;
- expoente t utilizado no critério de Papadrakakis de 0.7.

Na figura 4.44 temos a história no tempo do esforço axial no elemento comum a todas os risers no topo da estrutura que foi coincidente para todos os esquemas de solução utilizados.

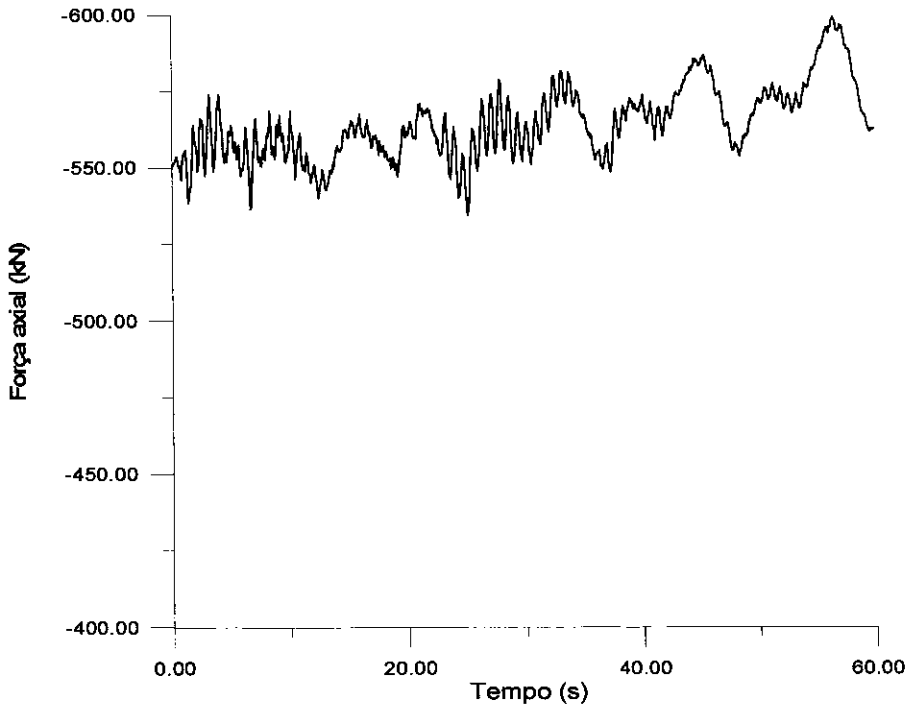


Fig. 4.44 - Força axial no topo da estrutura

4.2.1 - Análise com esquema implícito tradicional

No gráfico da figura 4.45 temos o número de iterações NR ao longo da análise, que, conforme podemos observar, manteve-se constante e igual a 2 durante toda a simulação.

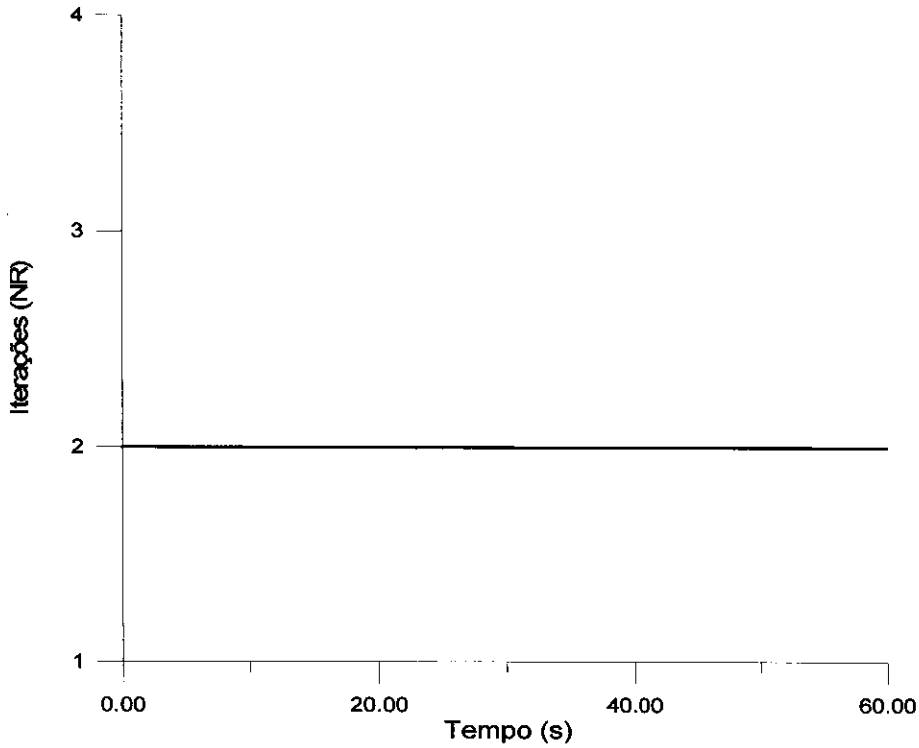


Fig. 4.45 - Iterações Newton Raphson X tempo

A seguir, na figura 4.46, temos o gráfico de norma do resíduo efetivo para cada iteração NR.

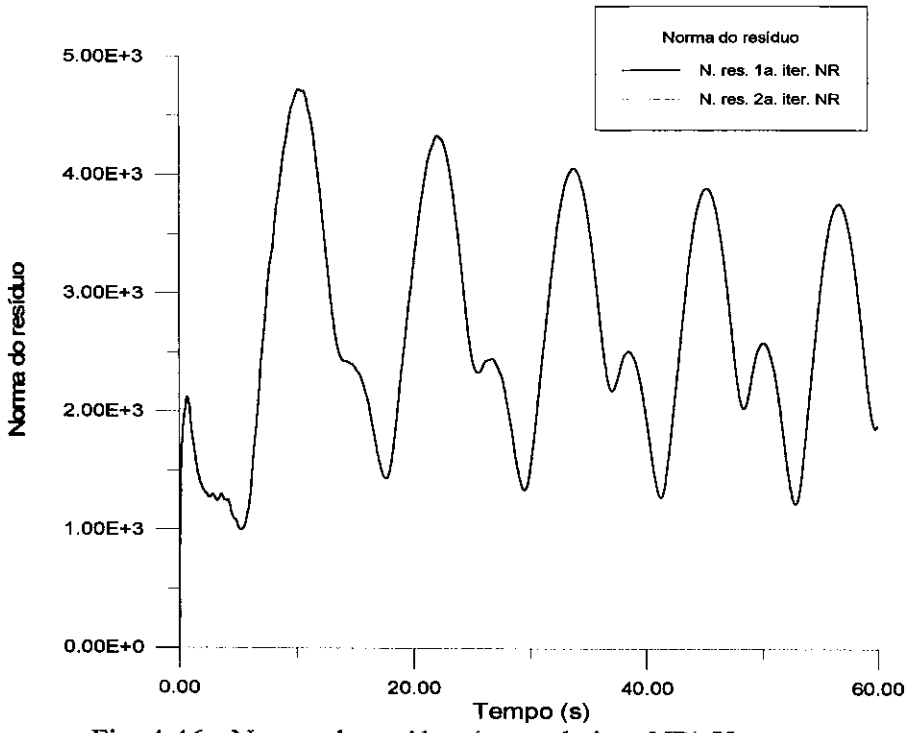


Fig. 4.46 - Norma do resíduo (em cada iter. NR) X tempo

A figura 4.47 nos mostra a relação entre a norma do resíduo da n-ésima iteração NR e a norma do resíduo da primeira iteração.

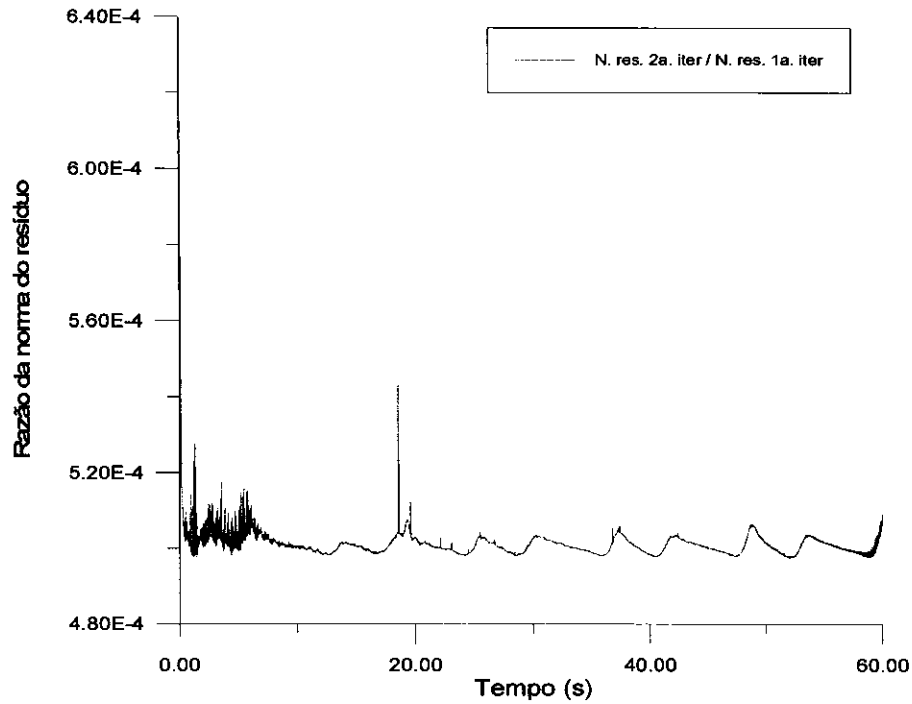


Fig. 4.47 - Razão da norma do resíduo (em cada iter NR) X tempo

4.2.2 - Análise com esquema semi-implícito proposto

4.2.2.1 - Critério de Papadrakakis

A partir deste sub-item serão plotados gráficos semelhantes aos apresentados para as aplicações do esquema semi-implícito no exemplo anterior.

Podemos notar na figura 4.48 o número de iterações NR variando entre duas, e três iterações. Isso demonstra novamente que a técnica de variação de tolerância no "solver" iterativo leva a um maior número de iterações NR.

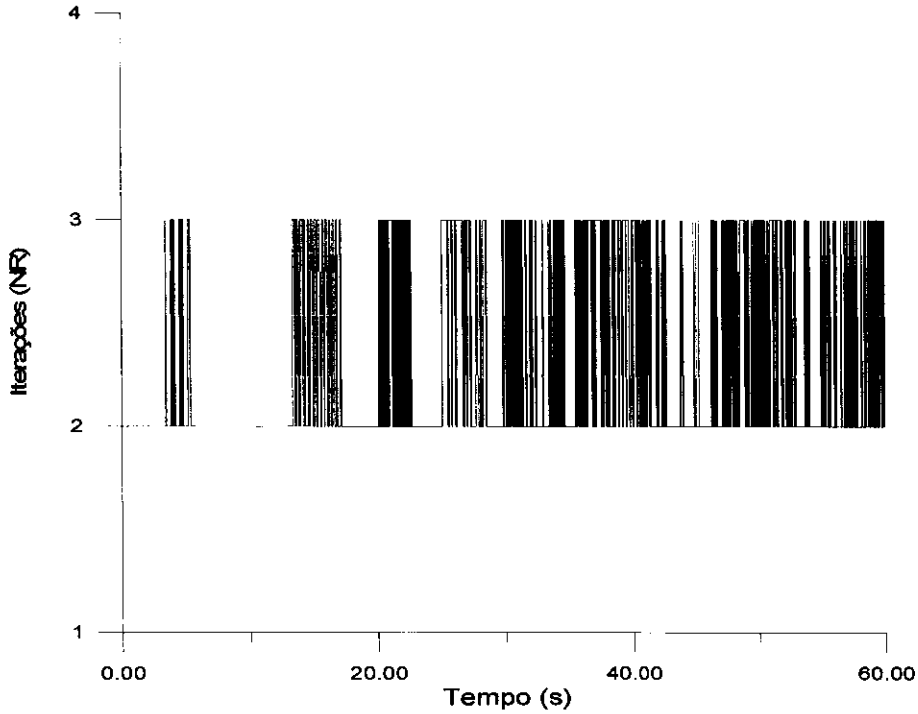


Fig. 4.48 - Iterações Newton Raphson X tempo

Nas figuras 4.49 e 4.50 os valores da norma do resíduo efetivo e a razão da norma. Notamos a semelhança entre estes gráficos e os relativos a estes obtidos ao utilizarmos o "solver" direto. Como era de se esperar, a norma do resíduo na 2ª iteração NR é maior do que a observada com a utilização do "solver" direto.

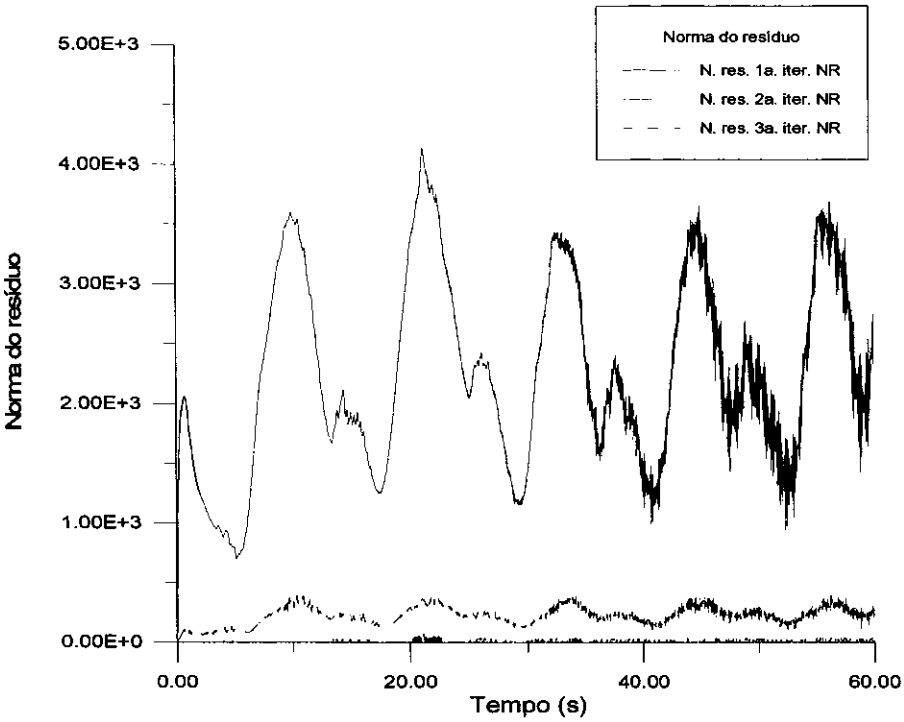


Fig. 4.49 - Norma do resíduo (em cada iter. NR) X tempo

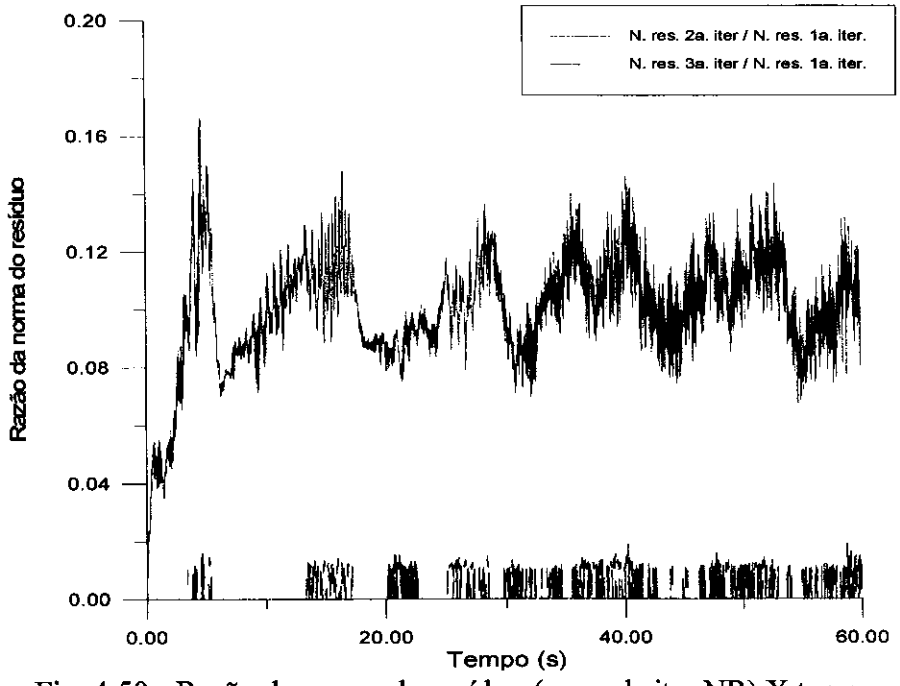


Fig. 4.50 - Razão da norma do resíduo (em cada iter NR) X tempo

Na figura 4.51 temos a variação da tolerância para convergância do MGCP. Devemos notar que na primeira iteração NR a tolerância estava constante com o valor máximo estipulado $\eta_0 = 0.3$.

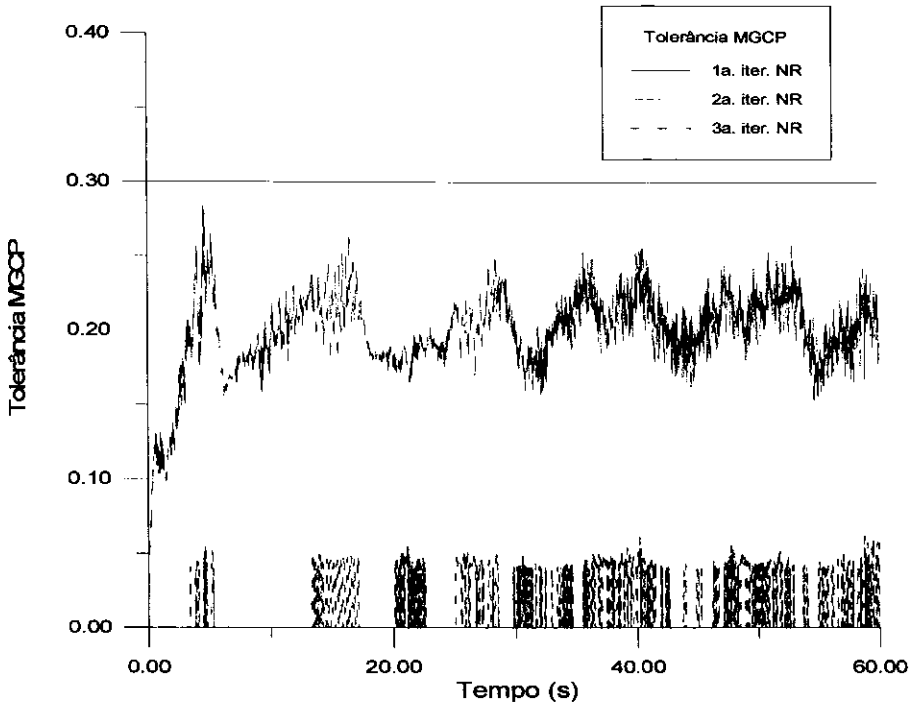


Fig. 4.51 - Variação da tolerância MGCP (em cada iter NR) X tempo

No gráfico da figura 4.52 temos a variação do número de iterações MGCP para cada iteração NR ao longo do tempo de análise. Como era esperado, à medida que as iterações NR avançam e o valor da tolerância MGCP diminui, o número de iterações MGCP aumenta.

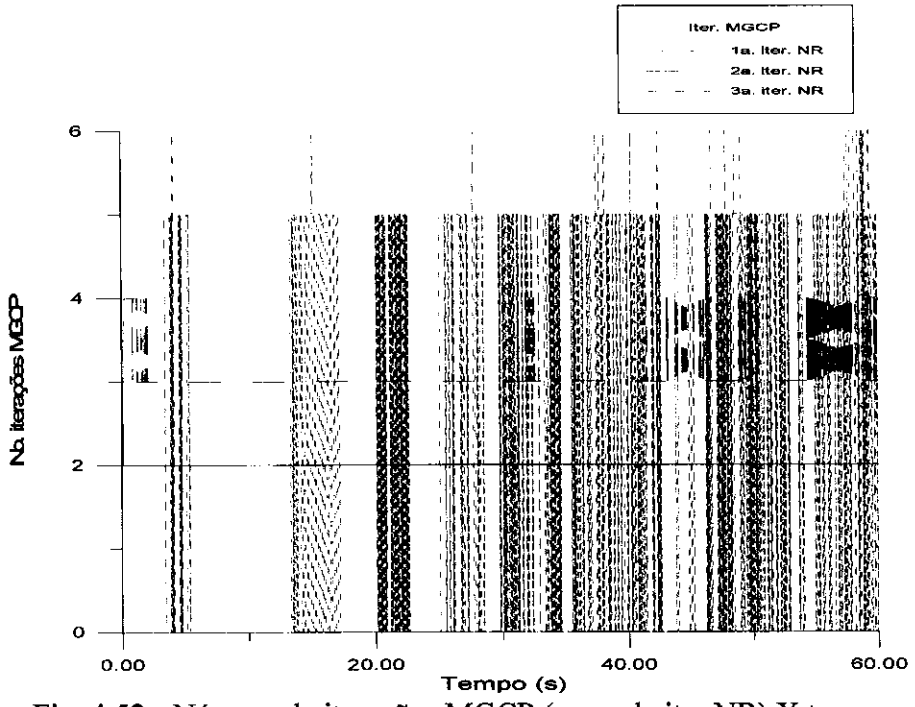


Fig. 4.52 - Número de iterações MGCP (em cada iter NR) X tempo

4.2.2.2 - Critério de Nour-Omid modificado (expoente t = 0.7)

Na análise deste exemplo utilizou-se também a variação proposta para o critério de Nour-Omid (onde o expoente t utilizado no cálculo da variação da tolerância é igual a 0.7) pois no exemplo anterior, este critério, juntamente com o proposto por Papadrakakis, foram os mais vantajosos em termos de economia de esforço computacional.

Temos, nas figuras 4.53 à 4.57 os gráficos semelhantes aos anteriores porém com os resultados obtidos do critério em questão. Primeiramente o gráfico do número de iterações NR.

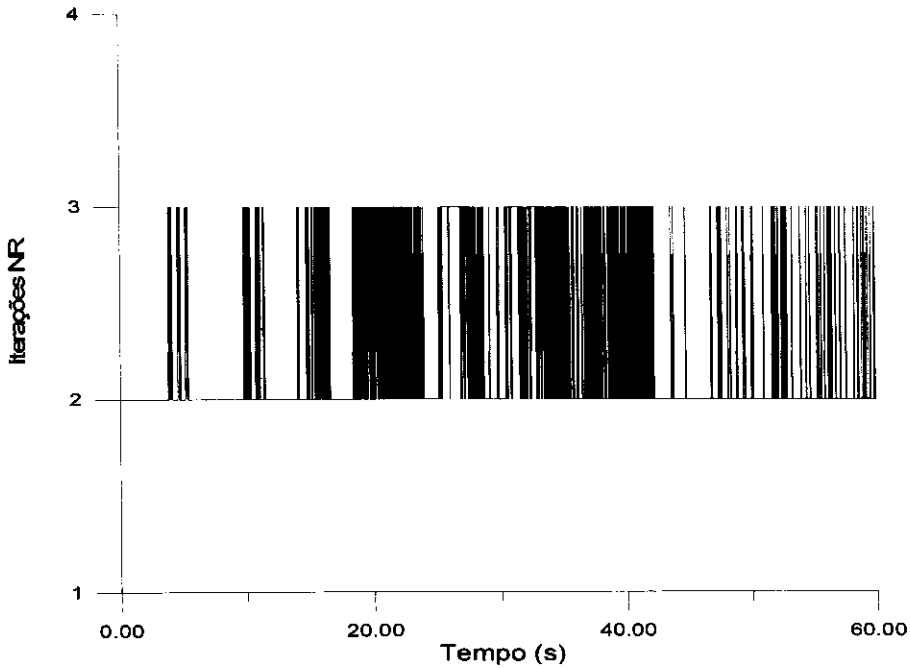


Fig. 4.53 - Iterações Newton Raphson X tempo

Nas figuras 4.54 e 4.55 os gráficos de norma e razão da norma do resíduo efetivo ao longo do tempo.

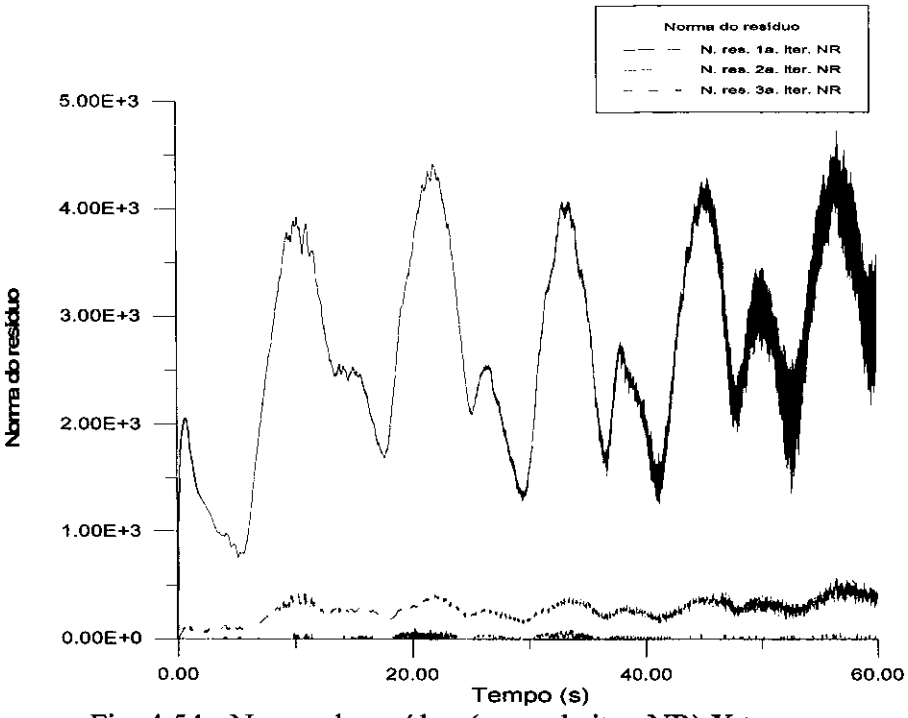


Fig. 4.54 - Norma do resíduo (em cada iter. NR) X tempo

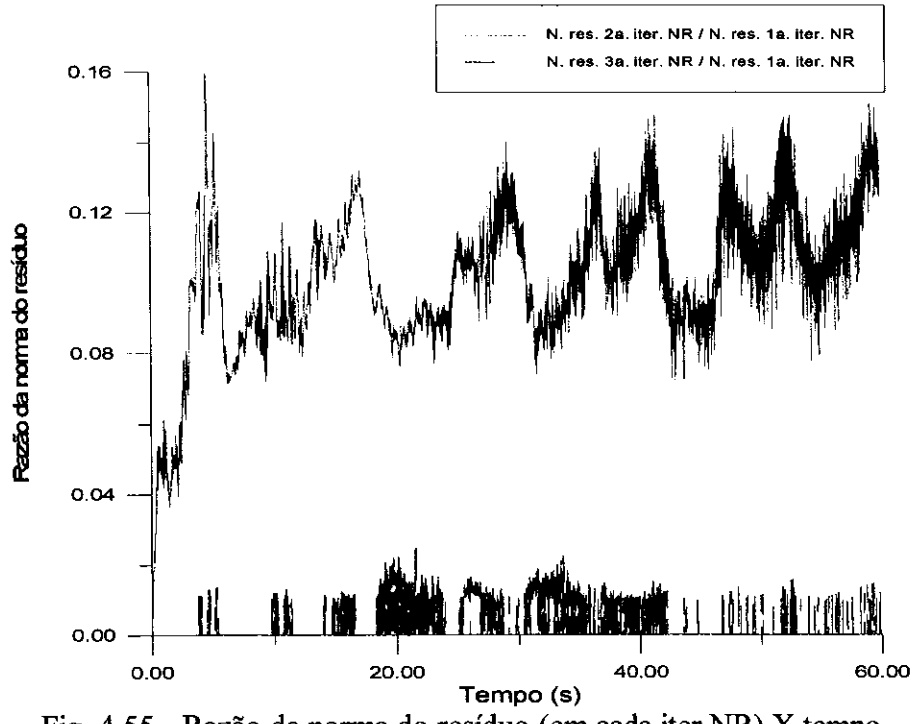


Fig. 4.55 - Razão da norma do resíduo (em cada iter NR) X tempo

Na figura 4.56 aparece a variação da tolerância MGCP ao longo do tempo.

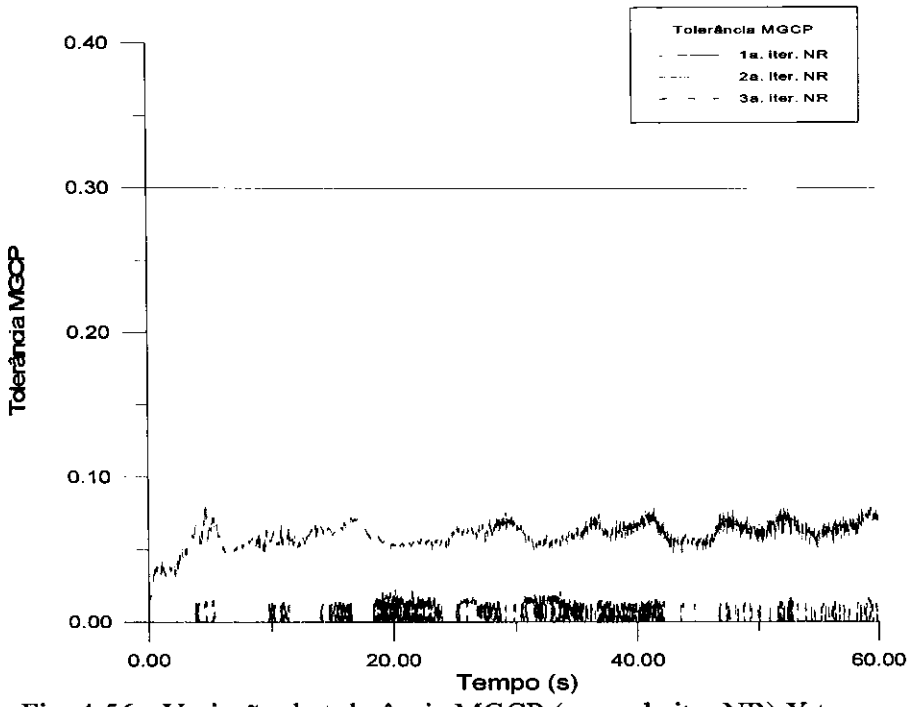


Fig. 4.56 - Variação da tolerância MGCP (em cada iter NR) X tempo

Finalmente, no gráfico da figura 4.57 observa-se a variação do número de iterações MGCP ao longo do tempo.

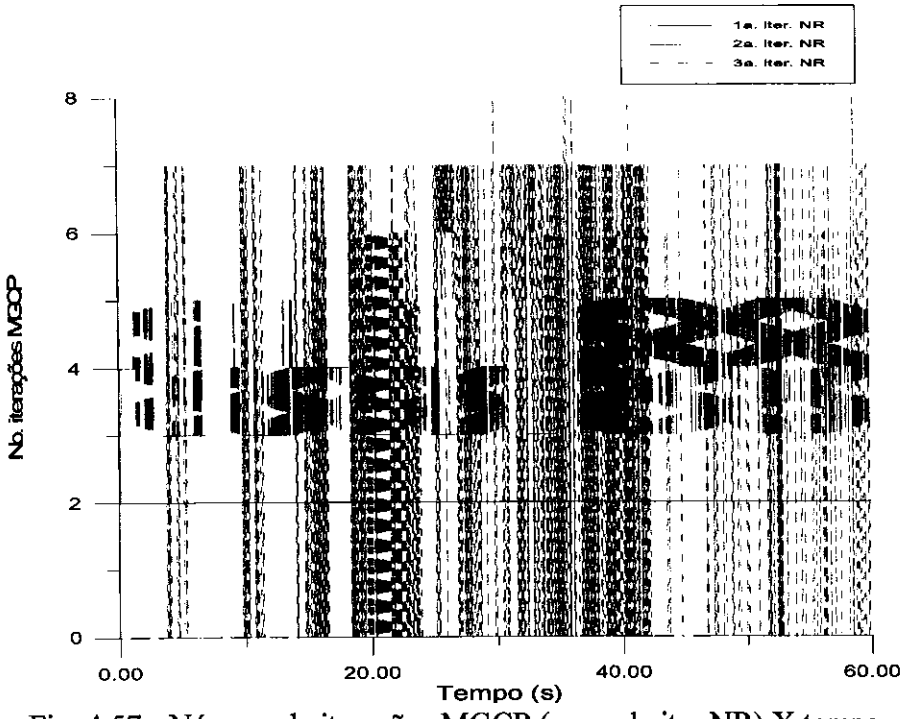


Fig. 4.57 - Número de iterações MGCP (em cada iter NR) X tempo

4.2.3 - Comparação de desempenho

Na tabela 4.2, a comparação entre o tempo total de CPU gasto em cada critério do MGCP utilizado. Além disso temos o custo de uma solução MGCP expresso em termos de um "número médio de iterações MGCP" já definida no item 3.3.

Tabela 4.2 - Comparação entre os métodos

	Tempo(s)	Iterativo / Direto	PCG / NR*
Direto	364	1.00	7.07+
Papadrakakis	334	0.92	2.83
Nour (t = 0.7)	344	0.95	3.23

* - Número médio de iterações MGCP.
+ - Número equivalente de iterações MGCP.

Nesta aplicação repetiu-se o ocorrido no exemplo anterior. Os métodos semi-implícitos foram mais vantajosos que o método tradicional implícito e o critério proposto por Papadrakakis mostrou-se de novo o mais vantajoso entre os métodos testados com uma economia de 8% em relação ao tempo gasto pelo "solver" direto.

4.2.4 - Análise com esquema auto-adaptativo

Seguindo a proposta do início de capítulo, utilizou-se na análise deste exemplo um critério auto-adaptativo em que o próprio algoritmo, no início da análise, faz uma comparação baseado em parâmetros de medidas de tempo e a partir daí decide qual o tipo de resolução de sistemas de equações, "solver" direto ou MGCP com tolerância variável (critério proposto por Papadrakakis), é o mais apropriado, em termos de menor esforço computacional para o exemplo em questão.

Podemos observar na fig. 4.58 que, a partir da análise comparativa de parâmetros o "solver" iterativo obteve o melhor desempenho, o que pode ser comprovado se observarmos o tempo total de análise utilizando-se cada tipo de solução proposta separadamente ("solver" direto e MGCP com tolerância variável).

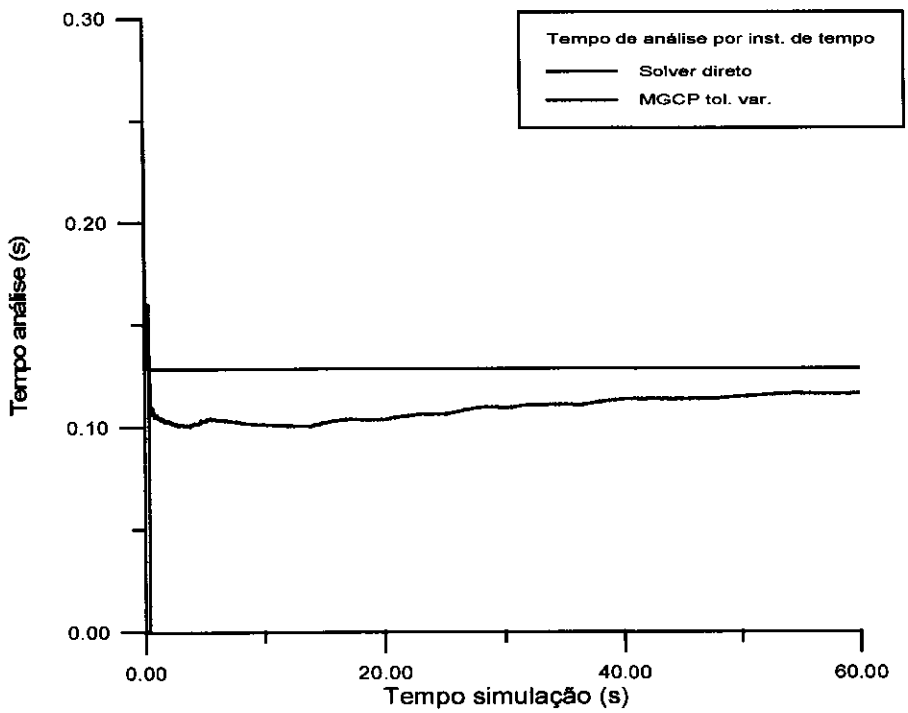


Fig. 4.58 - Tempo médio gasto na resolução de sistema de equações

CAPÍTULO 5

CONCLUSÕES

Verificou-se que, utilizando a técnica de tolerância variável em problemas dinâmicos não-lineares, é possível empregar valores de tolerância para o MGCP muito mais elevados que os tradicionalmente empregados em problemas estáticos. Com isso fomos capazes de obter economia considerável no esforço computacional, sem perda significativa de precisão, o que demonstra a eficiência do esquema semi-implícito considerado.

Podemos concluir também que o esquema "semi-implícito", nos exemplos estudados, foi sempre mais vantajoso que o esquema implícito tradicional. Além disso, observou-se que, possivelmente, com um estudo paramétrico mais detalhado, o que não foi possível devido à falta de tempo disponível, conseguiríamos mais economia em termos de tempo de CPU. Ficando assim uma proposta para continuação do trabalho aqui apresentado bastante promissora.

Deve-se ressaltar também que o esquema de armazenamento compacto da matriz efetiva, implementado para a solução via MGCP, permite uma redução considerável nos requisitos de memória central, o que pode viabilizar a análise de sistemas estruturais por modelos com centenas de milhares de graus de liberdade.

Outro fato para o qual deve-se chamar a atenção é a possibilidade de novas propostas para a variação da tolerância do MGCP ao longo das iterações NR. As propostas aqui apresentadas já foram significativas e trouxeram resultados satisfatórios. É claro que outras formas de variação da tolerância seriam motivos de novos estudos, desde que, respeitados os limites de precisão necessários para que se tenham resultados satisfatórios.

Dentre todos os critérios testados notamos que os mais vantajoso foram os baseados em variações do critério proposto por Nour-Omid e o critério proposto por Papadrakakis. Portanto, este seria o caminho inicial a ser seguido para novos estudos paramétricos e novas propostas de variação de tolerância MGCP.

A utilização do esquema auto-adaptativo é muito interessante no sentido de que, dependendo do problema em questão e do intervalo de tempo utilizado, o algoritmo pode, numa simples comparação, verificar qual o tipo de "solver" a ser utilizado e que proporcionará maior eficiência computacional sem maiores preocupações do usuário em relação a este tipo de problema.

Pode-se considerar que o esquema auto-adaptativo, que decide entre o MGCP com tolerância variável e "solver" direto, é um passo inicial na direção de um esquema "semi-implícito" adaptativo. Sabe-se que o desempenho do "solver" (direto ou iterativo) é dependente das características do comportamento dinâmico e não-linear de cada problema em particular, o qual irá se refletir no intervalo de tempo utilizado (que pode ser determinado automaticamente pelo programa, de acordo com uma estratégia adaptativa como a apresentada em [23]).

Portanto, o próximo passo seria a implementação de um esquema auto-adaptativo que incorpore a determinação automática de intervalos de tempo variando ao longo da análise e que permita a seleção do algoritmo mais apropriado, em termos de eficiência computacional, para resolução do sistema de equações ("solver" direto ou MGCP com tolerância variável) não só nos primeiros instantes de tempo, mas ao longo de toda a análise (em tempo real). Além disso, o esquema auto-adaptativo pode ser utilizado em conjunto com uma estratégia para a determinação automática dos instantes de tempo em que a rigidez deve ser reavaliada.

O resultado final seria além de economia em termos de esforço computacional, a minimização de custos de "mão-de-obra" do usuário, permitindo efetuar menos análises e gastar menos tempo para obter os resultados; e, tornar os resultados mais confiáveis e menos dependentes do conhecimento técnico do usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

[1]. B. P. JACOB, *Estratégias computacionais para a análise não-linear dinâmica de estruturas complacentes para águas profundas.* Tese de D.Sc., COPPE / UFRJ, Rio de Janeiro, RJ, Brasil (1990).

[2]. T. J. R. HUGHES, *The Finite Elements Method - Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, NJ, USA (1987).

[3]. N. M. NEWMARK, "A method of computation for structural dynamics". *J. Engng Mech Div.*, ASCE pp. 67-94 (1959).

[4]. K. J. B. BATHE, *Finite Element Procedures in Engineering Analysis.* Prentice-Hall, Englewood Cliffs, NJ, USA (1982).

[5]. A. C. BENJAMIN, *Análise Não-linear Geométrica de Pórticos Tridimensionais pelo Método dos Elementos Finitos.* Tese de M.Sc., COPPE / UFRJ, Rio de Janeiro, RJ, Brasil (1982).

[6]. H. M. HILBER, T. J. R. HUGHES and R. L. TAYLOR, "Improved numerical dissipation for time integration algorithms in structural dynamics", *Earthquake Engineering and Structural Dynamics,* vol. 5 pp. 283-292 (1977).

[7]. B. P. JACOB and N. F. F. EBECKEN, "Towards an adaptive 'semi-implicit' solution scheme for nonlinear structural dynamic problems". *Computer & Structures*, 52, no. 3, pp. 495-504 (1994).

[8]. T. J. R. HUGHES and T. BELYTSCHKO, "A précis of developments in computational methods for transient analysis". *J. appl. Mech.*, ASME 50, pp. 1033-1041 (1983).

[9]. T. BELYTSCHKO, H. J. YEN and R. MULLEN, "Mixed methods for time integrations". *Comput. Meth. Appl. Mech. Engng* 17/18, pp. 259-275 (1979).

[10]. W. K. LIU and T. BELYTSCHKO, "Mixed time implicit-explicit finite element for transient analysis". *Comput. Struct.* 15, pp. 445-450 (1982).

[11]. D. M. TRUJILLO, "An unconditionally stable explicit algorithm for structural dynamics". *Int. J. Numer. Meth. Engng* 11, pp. 1579-1592 (1977).

[12]. K. C. PARK, "An improved semi-implicit method for structural dynamics analysis". *J. appl. Mech.*, ASME 9, pp. 589-593 (1982).

[13]. R. MULLEN and T. BELYTSCHKO, "An analysis of an unconditionally stable explicit method". *Comput. Struct.* 16, pp. 691-696 (1983).

[14]. I. FRIED, "On a deficiency in unconditionally stable explicit time-integration methods in elastodynamics and heat transfer". *Comput. Meth. Appl. Mech. Engng* 46, pp. 195-200 (1984).

[15]. NOUR-OMID, "A preconditioned conjugate gradient method for solution of finite element equations". *In Innovative Methods for Nonlinear Problems.* Pineridge Press, Swansea (1984).

[16]. M. R. HESTENES and E. STIEFEL, "Methods of Conjugate Gradients for Solving Linear Systems", *J. Res. Natl. Bur. Stand.*, 45, pp. 409-436 (1952).

[17]. A. L. G. A. COUTINHO, J. L. D. ALVES, L. LANDAU e N. F. F. EBECKEN, "Solução iterativa de equações do Método dos Elementos Finitos no Computador IBM 3090", *In 2º Simpósio Brasileiro em Arquitetura de Computadores e Processamento Paralelo,* Ed. W. Bergamini, vol. 2, Águas de Lindóia, SP (1988).

[18]. M. V. RODRIGUES e B. P. JACOB, "Um esquema híbrido para a solução de problemas de dinâmica estrutural", *In XVIII CILAMCE - Congresso Íbero Latino-Americano de Métodos Computacionais em Engenharia,* vol. 2, pp 727-734, Brasília, DF (1997).

[19]. R. S. DEMBO and T. STEINHAUG, "Truncated-Newton algorithms for large scale unconstrained optimization". *Math. Programming* 26, pp. 190-212 (1983).

[20]. B. NOUR-OMID, B. N. PARLETT and R. L. TAYLOR, "A Newton-Lanczos method for solution of nonlinear finite element equations". *Comput. Struct.* 16, pp. 241-252 (1983).

[21]. M. PAPADRAKAKIS, *Solving large-scale non-linear problems in solid and structural methods. Solving Large Scale Problems in Mechanics*, Papadrakakis ed., John Wiley & Sons Ltd., England (1993).

[22]. B. P. JACOB, N. F. F. EBECKEN, A. C. BENJAMIN and M. M. MOURELLE, "Coupled and uncoupled solutions for the nonlinear dynamic behaviour of guyed deep water platforms". *Advances in Engineering Software*, Vol. 10 No. 1 pp. 2-7, Computational Mechanics Publications (1988).

[23]. B. P. JACOB and N. F. F. EBECKEN, Adaptive Time Integration of Nonlinear Structural Dynamic Problems. *European Journal of Mechanics*, A/Solids, Vol. 12 No. 2 pp. 277-298, Gauthier-Villars (1993).