

**PROCESSO ADAPTATIVO R-H GLOBAL CONJUGADO COM
MÉTODO MULTIMALHA PARA ELEMENTOS FINITOS**

Ana Maria Gontijo Figueiredo

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ESTRUTURAS

**"PROCESSO ADAPTATIVO R-H GLOBAL CONJUGADO COM
MÉTODO MULTIMALHA PARA ELEMENTOS FINITOS"**

Ana Maria Gontijo Figueiredo

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Estruturas da Escola de Engenharia da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do título de "Mestre em Engenharia de Estruturas".

Comissão Examinadora:

Prof. Estevam Barbosa de Las Casas
DEES/UFMG - (Orientador)

Prof. Jayme Pereira de Gouveia
IME

Prof. Philippe Bernard Remy Devloo
INPE

Prof. Márcio B. Carvalho
UFMG

Belo Horizonte, 25 de fevereiro de 1992

Aos meus pais, Delba e Eurico.

Aos meus filhos, Fernanda e Lucas.

"Tudo no mundo está dando respostas,
o que demora é o tempo das perguntas".

José Saramago
em "Memorial do Convento"

AGRADECIMENTOS

Ao meu orientador e amigo Estevam Barbosa de Las Casas, origem, possibilidade e concretização deste trabalho, cuja orientação dedicada e paciente foi decisiva para a sua realização.

Ao Departamento de Engenharia de Estruturas da Escola de Engenharia da UFMG pela oportunidade.

Aos colegas professores e aos funcionários do Departamento de Engenharia de Estruturas pela amizade e apoio cotidianos.

Ao amigo mestrando Ramon Pereira da Silva pela grande e paciente ajuda nos meandros da computação.

Aos amigos Professora Elizabeth Vieira Maia e Professor Estêvão Bicalho Pinto Rodrigues pelo grande apoio na conclusão deste trabalho.

Ao Professor Márcio Luiz Bunte de Carvalho pelas valiosas sugestões.

Ao Doutor Philippe Remy Bernard Devloo pelo interesse e incentivo.

Aos meus filhos pela compreensão e renúncia.

Aos meus pais, fortes raízes, por todos os motivos.

SUMÁRIO

I	INTRODUÇÃO.....	1
I-1	Considerações Iniciais.....	1
I-2	Processos Adaptativos.....	4
I-2-1	Principais estratégias dos processos adaptativos.....	6
I-3	Estimativas de Erro "A-Posteriori".....	11
I-4	Método Multimalha de Solução de Sistemas de Equações.....	19
I-4-1	Idéias básicas do Método Multimalha.....	19
I-4-2	O Método Multimalha.....	28
I-5	O Sistema para Desenvolvimento de Programas Baseados no Método dos Elementos Finitos - SDP....	37
I-6	Objetivo do Trabalho.....	42
II	PROCESSO ADAPTATIVO R-H GLOBAL. IMPLEMENTAÇÃO.....	44
II-1	Fase R. Processo R.....	44
II-2	Fase H. Implementação do Processo H.....	49
II-2-1	Geração das novas malhas M1, M2, M3.....	50
II-2-2	Estrutura do banco de dados de M1, M2 e M3.....	57
II-2-3	Estrutura do processador GERA.....	62
III	MÉTODO MULTIMALHA. IMPLEMENTAÇÃO.....	66
III-1	Preparação de Dados para Solução do Sistema de M3 por MM. O Processador PREMG.....	67
III-2	Solução do Sistema de Equações de M3 pelo Método Multimalha. O Processador MULGR.....	74
III-2-1	Descrição do algoritmo.....	74
III-2-2	Diagrama de estrutura do processador MULGR.....	79
III-3	Cálculo e Impressão de Tensões.....	82

IV-12	Problema BOUSSINESQ.....	98
IV-13	Problema BOUSSINESQ. Malha 0R-0H.....	100
IV-14	Problema BOUSSINESQ. Sequência de malhas da fase R.....	102
IV-15	Problema BOUSSINESQ. Detalhe da malha 3R-3H.....	103
IV-16	Problema BOUSSINESQ. Solução analítica.....	104
IV-17	Problema BOUSSINESQ. Malha (0R-0H)'.....	108
IV-18	Problema semelhante ao BOUSSINESQ. Parsons e Hall.....	110
IV-19	Problema BOUSSINESQ. Evolução do erro com os ciclos MM.....	111
IV-20	Problema BOUSSINESQ. Tensões σ_y em (x;3,0) obtidas com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H.....	112
IV-21	Problema BOUSSINESQ. Tensões σ_y em (x;3,0) obtidas com a malha 3R-3H e com a solução analítica.....	113
IV-22	Problema BOUSSINESQ. Tensões σ_y em (2,0;y) obtidas com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H.....	114
IV-23	Problema BOUSSINESQ. Tensões σ_y em (2,0;y) obtidas com a malha 3R-3H e com a solução analítica.....	115
IV-24	Problema MANCAL : chapa com furo.....	116
IV-25	Problema MANCAL - Malha 0R-0H.....	117
IV-26	Problema MANCAL. Sequência de malhas da fase R.....	118
IV-27	Problema MANCAL. Detalhe da malha 3R-3H.....	119
IV-28	Problema MANCAL. Tensões σ_y na linha 8-7 obtidas com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H.....	123
A1-1	Coordenadas nodais do elemento triangular de três nós.....	129
A1-2	Elementos triangulares semelhantes.....	130
A2-1	Diagrama de estrutura do processador ELIGA.....	132

IV	RESULTADOS NUMÉRICOS.....	83
IV-1	Problema TRINCA.....	85
	IV-1-1 Descrição do problema. Definição das malhas.....	85
	IV-1-2 Análise do desempenho do algoritmo MM....	91
	IV-1-3 Análise da eficácia do processo adaptativo R-H global.....	97
IV-2	Problema BOUSSINESQ.....	100
	IV-2-1 Descrição do problema. Definição das malhas.....	100
	IV-2-2 Análise do desempenho do algoritmo MM....	105
	IV-2-3 Análise da eficácia do processo adaptativo R-H global.....	112
IV-3	Problema MANCAL.....	116
	IV-3-1 Descrição do problema. Definição das malhas.....	116
	IV-3-2 Análise do desempenho do algoritmo MM....	120
	IV-3-3 Análise da eficácia do processo adaptativo R-H global.....	122
V	CONCLUSÕES E SUGESTÕES.....	124
APÊNDICES		
1	Matrizes de Rigidez de Elementos Triangulares Planos.....	128
2	Descrição do Processador ELIGA.....	132
3	Descrição do Processador GAUSDL.....	134
4	Descrição do Processador POSMG.....	137
REFERÊNCIAS BIBLIOGRÁFICAS.....		
BIBLIOGRAFIA.....		
		144

RESUMO

Este trabalho descreve a implementação e avaliação de um processo adaptativo R-H global para problemas de estado plano em elasticidade. O sistema linear gerado é resolvido por um algoritmo Multimalha, aproveitando a similaridade da estrutura de dados gerada no refinamento com aquela necessária para o algoritmo da solução.

O procedimento foi implantado utilizando o Sistema para Desenvolvimento de Programas Baseados no Método dos Elementos Finitos - SDP. Foram analisados três problemas, com os quais se fez uma avaliação do algoritmo Multimalha utilizado, comparativamente com os processos de Gauss e Gauss-Seidel, além de um estudo da influência dos parâmetros empregados e da irregularidade da malha no desempenho do algoritmo Multimalha.

Os resultados obtidos com o processo adaptativo R-H global foram comparados com resultados da literatura. Tanto este quanto o algoritmo de solução foram considerados satisfatórios.

Ao final são apresentadas sugestões para a continuação do trabalho.

SUMMARY

This work describes the implementation and evaluation of a global R-H adaptive process for plane problems in elasticity. The generated linear system is solved using a Multigrid algorithm, taking full advantage of the similarity of the data structure generated during refinement with the one required by the solver.

The method was implemented using the System for Finite Element Software Development (SDP). Three problems were analysed and used for validation of the proposed algorithm, comparison with Gauss and Gauss-Seidel, and assessing the sensibility to parameter change and mesh irregularity.

Obtained results using R-H adaptive process and performance of the solver were compared to data from the literature, and considered satisfactory.

Finally, suggestions are made to broaden the scope of this study.

LISTA DE TABELAS

IV-1	Problema TRINCA. Parametrização do MM - Malha 5R-3H.....	91
IV-2	Problema TRINCA. Aumento do número de ciclos MM com o aumento da irregularidade da malha.....	93
IV-3	Problema TRINCA - Malha 5R-3H. Tempo de CPU gasto na solução do sistema pelos processos EG, GS e MM.....	93
IV-4	Problema TRINCA - Malha 0R-3H. Tempo de CPU gasto na solução do sistema pelos processos EG, GS e MM.....	94
IV-5	Problema BOUSSINESQ. Parametrização do MM - Malha 3R-3H.....	106
IV-6	Problema BOUSSINESQ - Malha 3R-0H. Aumento do número de ciclos MM com o aumento do coeficiente de Poisson.....	106
IV-7	Problema BOUSSINESQ. Aumento do número de ciclos MM com o aumento da irregularidade da malha.....	107
IV-8	Problema BOUSSINESQ - Malha 3R-3H. Tempo de CPU gasto na solução do sistema de equações pelos processos EG, GS e MM.....	109
IV-9	Problema BOUSSINESQ - Malha 0R-3H. Tempo de CPU gasto na solução do sistema de equações pelos processos EG, GS e MM.....	109
IV-10	Problema MANCAL. Parametrização do MM - Malha 3R-3H.....	120
IV-11	Problema MANCAL. Fator irregularidade da malha.....	121
IV-12	Problema MANCAL - Malha 3R-3H. Tempo de CPU gasto na solução do sistema de equações pelos processos EG, GS e MM.....	121
IV-13	Problema MANCAL - Malha 0R-3R. Tempo de CPU gasto na solução do sistema de equações pelos processos EG, GS e MM.....	122

A3-1	Diagrama de estrutura do processador GAUSDL.....	135
A4-1	Diagrama de estrutura do processador POSMG.....	138

LISTA DE FIGURAS

I-1 a)	Refinamento H global	
	b) Refinamento H seletivo.....	7
I-2 a)	Refinamento P global	
	b) Refinamento P seletivo.....	8
I-3	Adaptação R.....	9
I-4	Tensões do MEF e tensões suavizadas.....	16
I-5	Modos v_j com números de onda k.....	24
I-6	Suavização de componentes oscilatórias do erro com processo de Jacobi.....	25
I-7	Componente do erro em Ω^h e Ω^{2h}	26
I-8	Interpolação de vetor de Ω^{2h} para Ω^h	27
I-9	Versão básica do Método Multimalha.....	30
I-10	Versões comuns do Método Multimalha.....	30
I-11	Esquema N do MM.....	36
I-12	Organização dos programas no SDP.....	38
I-13	Diagrama de estrutura dos processadores da versão básica do SDP.....	40
II-1	Diagrama de estrutura do processador da Fase R.....	45
II-2	Procedimento usado na rotina RLIVRE para mover um nó.....	47
II-3	Deslocamento do nó N (nó de contorno) na rotina ACHAY.....	48
II-4	Deslocamento do nó N (nó de contorno) na rotina AHCIR.....	49
II-5	Refinamento global H em três etapas.....	50
II-6	Elementos semelhantes com incidência correspondente.....	51
II-7	Ordem de numeração dos nós gerados a partir de um elemento genérico de M0.....	52
II-8	Sequência de numeração dos elementos de M1, M2 e M3.....	53
II-9	Coordenadas do novo nó 1.....	54

II-10	Definição das condições de contorno de um novo nó.....	55
II-11	Definição do carregamento de M3.....	56
II-12	Vetor IAPC.....	58
II-13	Exemplo de refinamento H.....	60
II-14	a) Exemplo de matriz armazenada em IDH1. b) Matriz associada armazenada em IDH2.....	62
II-15	Diagrama de estrutura do processador GERA.....	63
III-1	Sequência de renumeração dos GL dos elementos de M1, M2 e M3 gerados a partir de um elemento de M0.....	68
III-2	Parte do vetor IDH2 e vetor ID21 referentes a M1.....	70
III-3	Diagrama de estrutura do processador PREMG.....	72
III-4	Algoritmo esquemático do ciclo N implementado.....	74
III-5	Descrição do algoritmo implementado.....	75
III-6	Diagrama de estrutura do processador MULGR.....	80
IV-1	Problema TRINCA : chapa tracionada com trinca na região central.....	85
IV-2	Problema TRINCA. Malha 0R-0H.....	86
IV-3	Problema TRINCA. Sequência de malhas da fase R.....	87
IV-4	Problema TRINCA. Detalhe da malha 5R-3H.....	88
IV-5	a) Problema TRINCA. Malha CIMINI..... b) Detalhe da malha CIMINI.....	89 90
IV-6	Problema semelhante ao TRINCA. Parsons e Hall.....	95
IV-7	Malha de Parsons e Hall.....	95
IV-8	Evolução do erro com MM.....	96
IV-9	Problema TRINCA. Tensões σ_y em (x;0,0) obtidas com as malhas 0R-0H, 5R-0H e CIMINI.....	97
IV-10	Problema TRINCA. Tensões σ_y em (x;0,0) obtidas com as malhas 0R-3H e 5R-3H.....	98
IV-11	Problema TRINCA. Tensões σ_y em (x;0,0) obtidas com as malhas 5R-3H e CIMINI.....	99

CAPÍTULO I

INTRODUÇÃO

I-1 Considerações Iniciais

Os conceitos fundamentais do Método dos Elementos Finitos, introduzidos por Leibnitz no século XVII, podem ser resumidos na aproximação de uma função que descreve o comportamento de um meio contínuo por várias pequenas funções suaves, ou seja, substitui-se a solução exata de um problema, muitas vezes impossível de ser encontrada analiticamente, por uma solução aproximada. Atualmente, com o desenvolvimento acelerado dos computadores, o método se tornou uma das mais poderosas ferramentas na análise de problemas físicos, principalmente os governados por equações diferenciais parciais, presentes em várias áreas do conhecimento, particularmente nos diversos ramos da Engenharia.

A confiabilidade dos resultados de uma análise feita com elementos finitos é função de diversos fatores, sendo determinada pelos erros introduzidos nas várias etapas do processo. As principais fontes de erro são a formulação matemática do problema físico, a discretização do modelo matemático contínuo, a implementação de algoritmos numéricos em computadores de precisão finita e as incertezas e erros cometidos na manipulação dos dados de entrada.

À utilização de uma determinada formulação matemática do problema físico deve preceder a identificação do seu espectro de validade. Os cuidados tomados principalmente no que se refere à não violação das hipóteses simplificadoras básicas da teoria minimizam o erro cometido nessa etapa.

A discretização do meio contínuo, isto é, a

escolha de uma malha de elementos finitos que substitua o problema matemático contínuo, gera o erro de discretização, que se traduz na diferença entre a solução exata do problema e a solução aproximada obtida com o modelo discreto. A obtenção de uma malha que leve a uma boa solução não é uma operação simples, uma vez que envolve um processo de tentativa e erro baseado, na maioria das vezes, em critérios semi-empíricos e na experiência anterior do analista.

A grande maioria dos sistemas computacionais comerciais que utilizam o Método dos Elementos Finitos (MEF) como método de análise não fornecem informações sobre a precisão dos resultados obtidos, isto é, não fazem análise de erro. Este fato é gerador de insegurança no usuário quanto à confiabilidade do modelo por ele utilizado ou, o que é bem mais grave, pode induzir a uma falsa segurança com origem no desconhecimento da existência desses erros.

A partir da década de 70 vários trabalhos foram desenvolvidos buscando técnicas de geração de malhas ótimas de elementos finitos, isto é, malhas que produzem a solução mais próxima possível da solução exata. Neste sentido foi proposta inicialmente uma formulação do Método dos Elementos Finitos que minimiza o funcional da energia potencial em relação às coordenadas nodais, além da sua minimização em relação aos deslocamentos nodais (Carrol-1978). Então, além dos deslocamentos nodais, as coordenadas nodais são tratadas como incógnitas do problema. Assim, a expressão da energia potencial

$$\Pi_P = \sum_{e=1}^n \left[\underline{u}_e^T \left[\int_V \underline{B}_e^T \underline{D} \underline{B}_e dV \right] \underline{u}_e \right] - \underline{P} \underline{u} \quad (I.1)$$

onde \underline{B}_e = matriz de transformação deformação-deslocamento
 \underline{D} = matriz de propriedades elásticas
 \underline{P} = vetor de cargas equivalentes
 \underline{u}_e = vetor de deslocamentos do elemento
 \underline{u} = vetor de deslocamentos globais
 n = número de elementos

ao ser minimizada para coordenadas e deslocamentos nodais,

$$\frac{\partial \Pi_P}{\partial \tilde{u}_{ei}} = 0 \quad i=1,2,\dots,n_a \quad n_a = n \text{ de deslocamentos nodais não restritos} \quad (I.2.a)$$

$$\frac{\partial \Pi_P}{\partial \tilde{c}_j} = 0 \quad j=1,2,\dots,n_b \quad n_b = n \text{ de coordenadas nodais não restritas} \quad (I.2.b)$$

$\tilde{c}_j = \text{vetor de coordenadas nodais}$

produz dois sistemas de equações

$$\tilde{K} \tilde{u} - \tilde{P} = 0 \quad \text{com } K = \sum_1^n \left[\int_v \tilde{B}_e^T \tilde{D} \tilde{B}_e dV \right] \quad (I.3.a)$$

$$0.5 \tilde{u}^T \frac{\partial \tilde{K}}{\partial \tilde{c}_j} \tilde{u} - \frac{\partial \tilde{P}}{\partial \tilde{c}_j} \tilde{u} = 0 \quad \text{com } j=1,2,\dots,n_b \quad (I.3.b)$$

que ao serem resolvidos determinam a melhor solução possível em elementos finitos para um dado tipo de elemento, para um número fixo de graus de liberdade.

Como este processo de otimização das coordenadas nodais requer técnicas complexas de programação matemática não linear que tornam o custo da análise quase proibitivo, foram propostas outras formas de obtenção da malha ótima. Estas técnicas formaram a base para o desenvolvimento de processos adaptativos de melhora da qualidade das soluções através da modificação da discretização inicial, processos estes que ultimamente têm sido objeto de inúmeros trabalhos.

Os processos adaptativos de melhora de malhas de elementos finitos, em suas várias versões, utilizam, na sua maioria, estimativas de erro de discretização após o

processamento do problema - estimativas de erro "a posteriori" - que sinalizam as modificações a serem feitas na malha inicial de modo a melhorar a solução. A aplicação repetida do processo leva à obtenção de uma malha que se aproxima da malha ótima - a malha quase-ótima.

Os erros provenientes da implantação de algoritmos numéricos em computadores de precisão finita, os chamados erros de truncamento ou arredondamento, podem ser muito significativos. Apesar disso, os procedimentos propostos para análise adaptativa não tratam deste tipo de erro, ficando a cargo do analista tomar os cuidados necessários para seu controle.

Os sucessivos sistemas de equações gerados nos procedimentos adaptativos são preferencialmente resolvidos por processos iterativos de solução de sistemas de equações, como Gauss-Seidel, Jacobi, SOR, nos quais se utiliza a solução de uma etapa como aproximação inicial na etapa seguinte, acelerando o processo. Novos processos de solução de sistemas de equações têm sido propostos, destacando-se entre eles o Método Multimalha que por suas características, tais como a estrutura do banco de dados que requer, podem torna-lo adequado à aplicação nos processos adaptativos.

I-2 Processos Adaptativos

Entende-se por processo adaptativo em análise numérica aquele que utiliza resultados intermediários, gerados durante a própria solução, para modificar o modelo visando melhorar os resultados. Neste sentido, o procedimento usual para se obter uma boa malha de elementos finitos é adaptativo por natureza, pois um bom analista, de posse dos resultados de um primeiro processamento e baseado em sua experiência anterior, pode identificar os possíveis problemas na discretização inicial e então modificar a malha, num processo que é repetido tantas vezes quantas forem

necessárias para que, no seu julgamento, a solução resulte satisfatória. Os problemas de discretização geralmente se referem a regiões onde a malha não está suficientemente refinada, a orientação inadequada dos elementos, a irregularidades dos elementos (elementos muito distorcidos), a transições bruscas no tamanho dos elementos de uma região para outra (problemas de suavidade), a utilização inadequada de um determinado tipo de elemento ou, até mesmo, a uma descrição inadequada dos materiais empregados e do carregamento atuante. Portanto, são muitos os parâmetros a serem considerados pelo analista para que ele possa propor uma malha cujos resultados se aproximem satisfatoriamente da solução exata.

Por volta de 1980 começaram a ser propostos sistemas que executam procedimentos adaptativos automaticamente, isto é, sem interação com o analista - processos auto-adaptativos. Todas as menções que se fizer a seguir a processos adaptativos referem-se a processos auto-adaptativos.

A análise adaptativa de modelos em elementos finitos é importante na medida em que elimina grande parte do caráter subjetivo da adequação de malhas. O processo envolve estimativas de erro "a priori" ou "a posteriori" que tornam possível a obtenção da solução "melhor possível" a um custo aceitável.

As estimativas de erro "a priori" requerem o conhecimento prévio das características da solução exata e fornecem informações qualitativas sobre a convergência da solução quando o número de graus de liberdade tende ao infinito. Geralmente não fornecem informações sobre o erro real de um determinado modelo, fato que somado à grande dificuldade em sua formulação restringe sua aplicação a alguns casos simples.

As estimativas de erro "a posteriori" têm recebido ultimamente grande atenção dos pesquisadores e sua aplicação tem se mostrado muito eficaz e eficiente. Têm sido propostos vários estimadores e indicadores de erro baseados

em informações obtidas durante o processo de solução, tais como normas de energia de deformação ou tensões máximas. Estas estimativas de erro são feitas localmente no nível do elemento e algumas hipóteses prévias sobre a solução são consideradas.

Todos os algoritmos adaptativos utilizam, direta ou indiretamente, estimadores ou indicadores de erro e usualmente tentam igualar o valor desses erros nos diversos elementos (ver seção I-3).

I-2-1 Principais estratégias dos processos adaptativos

Dentre as técnicas propostas para a minimização do erro de discretização, três se destacam:

- aumento global ou seletivo do número de elementos da malha (processo H),
- aumento global ou seletivo do grau do polinômio de interpolação (processo P),
- redistribuição dos nós (graus de liberdade) no domínio (processo R).

Combinações dos processos acima dois a dois também têm sido utilizadas com sucesso.

A fundamentação teórica dos diversos processos adaptativos está bem desenvolvida para o caso unidimensional. A formulação para problemas multidimensionais é normalmente fundamentada em argumentos heurísticos e analogias com os problemas unidimensionais.

a) Processo H

A versão H dos processos adaptativos consiste no aumento do número de graus de liberdade da discretização através da subdivisão dos elementos originais - refinamento da malha - mantendo-se constante o grau do polinômio de interpolação. O refinamento pode ser feito globalmente em todo o domínio, mas em geral é feito localmente em regiões onde o erro é maior (refinamento seletivo).

Alguns cuidados devem ser tomados na

implementação do processo, como tentar manter a malha regular evitando a criação de elementos muito distorcidos, cuidar para que não sejam feitas transições muito bruscas no tamanho dos elementos no caso de refinamento seletivo e impor condições de contorno corretas bem como gerar cargas nodais equivalentes para os novos nós.

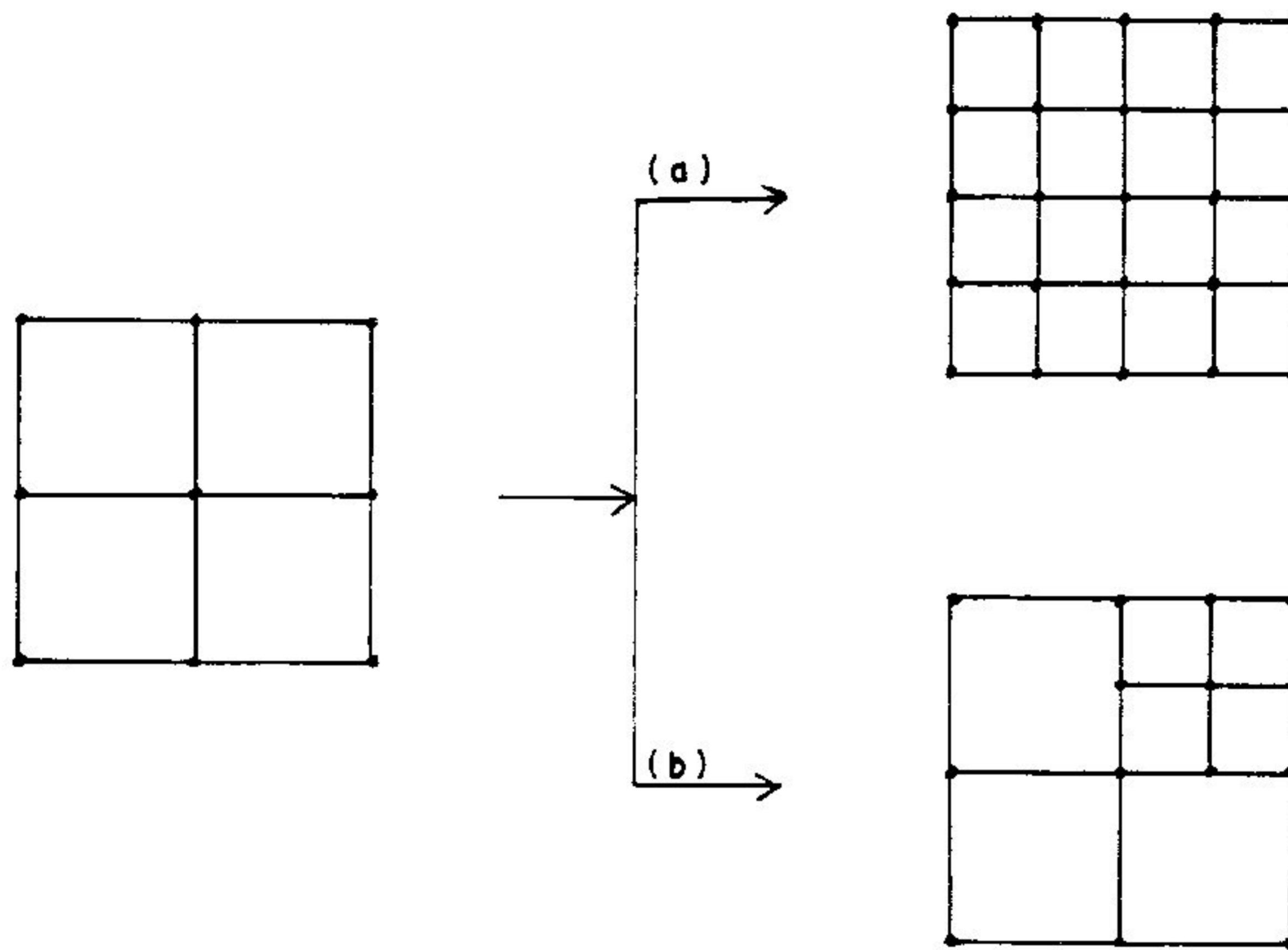


FIG. I.1 a) Refinamento H global
b) Refinamento H seletivo

b) Processo P

Outra maneira de aumentar o número de graus de liberdade da discretização é através do aumento do grau do polinômio de interpolação, mantendo-se constante o número de elementos da malha. Este procedimento é chamado versão P dos processos adaptativos. O processo é particularmente eficiente quando se utiliza funções de interpolação hierárquicas, isto é, funções que geram matrizes de rigidez dos elementos de ordem mais baixa contidas nas matrizes de rigidez dos elementos de ordem superior.

O refinamento de malhas utilizando o processo P pode ser feito, da mesma forma que no processo H, globalmente em todo o domínio, mas em geral é feito localmente nas regiões onde o erro for maior. O refinamento seletivo cria algumas dificuldades no tratamento do problema pois leva à criação de nós não conformes, além de exigir a consideração simultânea de polinômios de graus diferentes na mesma discretização.

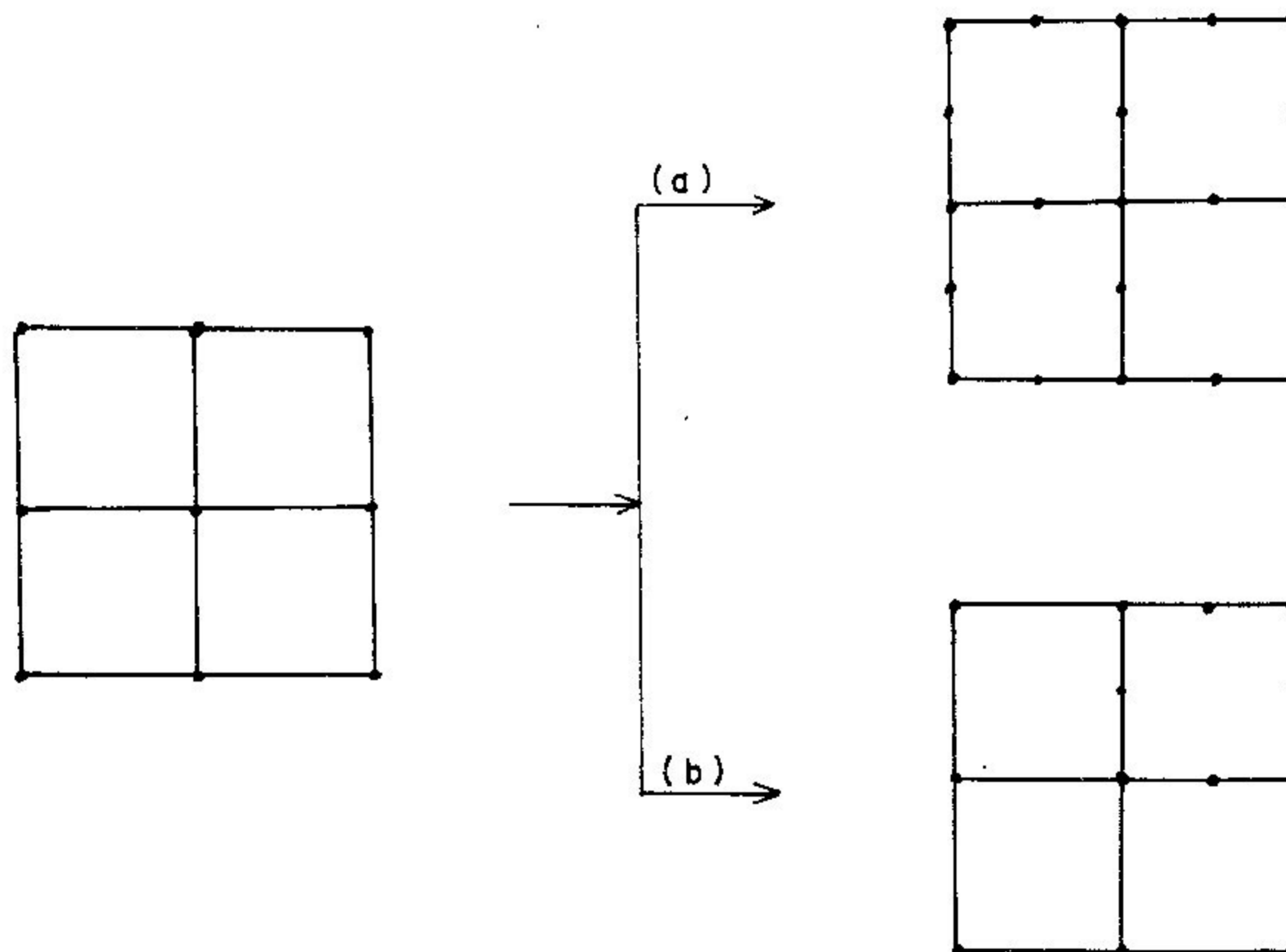


FIG. I.2 a) Refinamento P global
b) Refinamento P seletivo

c) Processo R

Um outro processo de melhora adaptativa de discretizações em elementos finitos, chamado processo R, consiste em se alterar as posições dos nós na malha mudando com isto a distribuição espacial dos elementos, sem alterar o número de graus de liberdade e o tipo de elemento utilizado.

Como esta estratégia redistribui os nós sobre o domínio até que o erro, medido através de um estimador (ou indicador) particular, seja minimizado, a utilização de

processos H ou P se torna bem mais simples pois não se faz mais necessária a identificação das regiões onde o erro é maior. Nesse caso o refinamento H ou P é feito geralmente na sua forma global, isto é, em todo o domínio do problema.

e) Comparação entre os processos

Em princípio, a discretização resultante do processo R seria a mais eficiente, isto é, a que levaria a um menor erro para um determinado número de graus de liberdade. Na prática isto não é necessariamente verdadeiro devido a alguns fatores como o efeito da distorção excessiva dos elementos durante o processo, a imprecisão das estimativas de erro e principalmente a restrições ao livre posicionamento dos nós no contorno (Las Casas-1988).

Nos problemas elípticos os processos H e P têm sido mais empregados que o processo R. A versão P apresenta algumas vantagens tais como a de produzir matrizes mais densas sendo portanto de solução mais simples, principalmente quando se utiliza elementos hierárquicos. Além disso, tem se mostrado menos sensível aos problemas de "locking" encontrados na presença de materiais muito incompressíveis. Em presença de singularidades a versão P apresenta, em problemas selecionados, uma taxa de convergência duas vezes maior que a versão H (Noor & Babuska-1987).

Os refinamentos H e P globais são de implementação bem mais simples que os seletivos, mas geram um volume muito grande de dados a serem manipulados. Em contrapartida, os refinamentos seletivos - apenas nas regiões onde o erro é maior - exigem formulações especiais para que não sejam gerados nós não conformes e podem ocasionar problemas devidos a grande variação no grau dos polinômios de elementos vizinhos ou a uma transição brusca no refinamento de uma região para outra (Las Casas-1990). No caso de malhas aproximadamente uniformes a taxa de convergência da versão P é maior que a taxa para a versão H baseada no mesmo número total de graus de liberdade (Noor & Babuska-1987).

A versão H-P apresenta uma taxa de convergência exponencial em muitos tipos de problemas de Engenharia, contra uma taxa polinomial das versões H ou P (ver item I-3). A versão P, entretanto, quando usada em malhas bem graduadas, pode apresentar uma taxa de convergência próxima à da versão H-P na faixa pré-assintótica (Noor & Babusca-1987 e Ewing-1990).

As versões R-H e R-P são interessantes pois partindo-se de uma malha já melhorada através da minimização do erro (fase R), o número de refinamentos H ou P que se terá que fazer para se atingir uma determinada precisão é, em princípio, bem menor que nas versões H ou P simples. Isto leva a uma boa solução com um número menor de graus de liberdade. Além disso, aqui os refinamentos H ou P podem ser feitos globalmente, o que diminui consideravelmente o grau de complexidade da implantação dos processos.

I-3 Estimativas de Erro "A-Posteriori"

O objetivo dos processos adaptativos é melhorar uma dada malha de elementos finitos de tal forma que sua solução apresente resultados o mais próximo possível da solução exata.

O erro E_h da discretização h de um determinado domínio Ω , também chamado erro global, é a norma da diferença entre a solução exata \underline{u} e a solução aproximada \underline{u}_h obtida com a malha de elementos finitos.

$$E_h = || \underline{u} - \underline{u}_h || \quad (I-4)$$

onde $||.\|$ é uma norma previamente selecionada.

Como a solução exata não é conhecida, é necessário estabelecer-se uma forma de estimar o erro de discretização das malhas.

Segundo Noor e Babuska-1987, nos problemas elípticos lineares, positivos definidos, o erro de discretização é constituído de duas partes: o erro de aproximação e o erro de perturbação. O primeiro, erro de aproximação, caracteriza a aproximação da solução em elementos finitos e depende da ordem da derivada de ordem maior usada na formulação. O segundo, erro de perturbação, resulta de aproximações geométricas do contorno, de aproximações das condições de contorno essenciais e de continuidade e das aproximações numéricas das integrais.

As estimativas de erro tradicionais para os métodos de elementos finitos e diferenças finitas são limites pré-estabelecidos - estimativas "a priori" - que determinam a taxa de convergência assintótica quando a dimensão da malha tende a zero. Este tipo de estimativa de erro não tem se mostrado muito eficaz pois fornece pouca informação sobre o erro real de uma determinada malha, além de ser de difícil formulação, fato que restringe sua aplicação a alguns casos simples (Ewing-1990).

As estimativas de erro "a priori" têm sido substituídas por estimativas "a posteriori" feitas localmente no nível do elemento. Estas estimativas são baseadas em informações obtidas durante o processo de solução e utilizam estimadores de erro (erros em valores absolutos) ou indicadores de erro (erros relativos).

A escolha de uma medida de erro (estimadores ou indicadores) depende do objeto da análise, isto é, de quais resultados devem ser calculados com precisão. Diferentes estimadores devem ser usados para controlar variáveis diferentes, uma vez que um erro pequeno em uma variável, deslocamentos por exemplo, não garante um erro pequeno em resultados derivados do primeiro, tais como tensões ou deformações.

As três normas mais utilizadas na estimativa de erros de discretização em elementos finitos são:

a) Norma de energia

$$\|\underline{e}\|_E = \left(\int_{\Omega} U_{(e)} d\Omega \right)^{1/2} \quad (I-5)$$

ou seja, é a raiz quadrada da energia do erro, onde Ω é o domínio da solução, U é a energia de deformação e \underline{e} é a função erro de discretização.

Esta norma também pode ser escrita como

$$\|\underline{e}\|_E = \left(\int_{\Omega} (\underline{e}^T \underline{L} \underline{e}) d\Omega \right)^{1/2} \quad (I-6)$$

onde \underline{e} é o erro e \underline{L} é um operador. No caso específico de elasticidade,

$$\|\underline{e}\|_E = \left(\int_{\Omega} (\underline{S} \underline{e})^T \underline{D} (\underline{S} \underline{e}) d\Omega \right)^{1/2} \quad (I-7 a)$$

$$= \left(\int_{\Omega} (\underline{e}_{\varepsilon})^T \underline{D} (\underline{e}_{\varepsilon}) d\Omega \right)^{1/2} \quad (I-7 b)$$

$$= \left(\int_{\Omega} (\underline{e}_{\sigma})^T \underline{D}^{-1} (\underline{e}_{\sigma}) d\Omega \right)^{1/2} \quad (I-7 c)$$

onde \underline{D} é a matriz de elasticidade,

\underline{S} é a matriz de transformação deslocamento-deformação,

$\underline{e}_{\varepsilon} = \underline{\varepsilon} - \underline{\varepsilon}_h$ é o erro na deformação,

$\underline{e}_{\sigma} = \underline{\sigma} - \underline{\sigma}_h$ é o erro na tensão,

$\underline{e} = \underline{u} - \underline{u}_h$ é o erro no deslocamento.

b) Norma dos quadrados principais (L^2)

$$\|\underline{e}\|^0 = \left(\int_{\Omega} \underline{e}^2 d\Omega \right)^{1/2} \quad (I-8)$$

ou seja, é a raiz quadrada dos quadrados principais do erro no domínio.

c) Norma máxima

$$\|\underline{e}\|_{\infty} = \max_{x \in \Omega} | \underline{e}(x) | \quad (I-9)$$

ou seja, é o valor absoluto máximo do erro no domínio.

Em (b) e (c) ξ se refere também, por exemplo, a deslocamentos ou tensões.

A norma de energia tem sido mais empregada e é equivalente à norma dos quadrados principais em tensões (Ewing-1990, Zienkiewicz e Zhu-1987). Como se estará fazendo referência, daqui por diante, sempre à norma de energia, o índice E será omitido.

Babuska e Rheinboldt formularam em 1978 um estimador de erro local que se baseia na norma da diferença, para cada nó, da solução da malha original (u_0) e da solução de uma malha mais refinada ($\bar{u}_h(\Omega_i)$) nas vizinhanças do nó em questão.

$$\bar{\eta}_i = \| u_0 - \bar{u}_h(\Omega_i) \| \quad (I-10)$$

A convergência deste estimador foi provada matematicamente, mas o seu uso foi reduzido devido a dificuldades de programação introduzidas por tantos problemas locais. A relação entre o tamanho dos elementos e os resíduos locais originados do desequilíbrio de forças quando se aplica o operador original à solução de elementos finitos levou-os à importante conclusão de que uma malha próxima à ótima pode ser obtida quando os estimadores de erro local são iguais em todos os elementos. Esta propriedade tem sido largamente utilizada em pesquisas desde então.

Bank e Weiser, em 1985, propuseram um estimador de erro baseado na solução de problemas locais introduzidos pelo resíduo da equação diferencial e pela descontinuidade das derivadas no contorno dos elementos.

Gago, Kelly e Zienkiewicz, em 1981, formularam e Zienkiewicz e Craig, em 1986, desenvolveram um estimador baseado no cálculo de aproximações de ordem mais alta da solução original, sem mudar os valores da aproximação original.

Rank e Zienkiewicz em 1987 e Lyra em 1988 desenvolveram um estimador baseado no cálculo da diferença

entre o gradiente da aproximação em elementos finitos e o gradiente "suavizado" pós-processado.

Las Casas em 1988 utilizou com sucesso um estimador de erro baseado na semi-norma do espaço de Sobolev (Ciarlet, 1988), a partir do qual prova-se que uma solução próxima da ótima é obtida quando o erro medido pela expressão I-11 é constante no domínio.

$$e_i = \frac{h^{k+1}}{\rho^m} |u_h|_{k+1} \quad (I-11)$$

onde e_i é o erro no elemento i , h é o diâmetro máximo do elemento i , ρ é o diâmetro mínimo do elemento i , k é a ordem do polinômio e $|u_h|_{k+1}$ é a semi-norma de Sobolev da solução obtida em deslocamentos.

Zienkiewicz e Zhu, em 1987, propuseram um estimador que consiste em aproximar o campo de tensões exatas por um campo de tensões contínuas (suavizadas), obtido segundo uma interpolação semelhante à dos deslocamentos,

$$\tilde{\sigma}^* = \tilde{N} \tilde{\sigma}^* \quad (I-12)$$

onde \tilde{N} é a matriz que contém funções de interpolação do mesmo grau que as usadas para os deslocamentos e $\tilde{\sigma}^*$ são os valores nodais das tensões obtidas ponderando-se a diferença entre os campos de tensões $\tilde{\sigma}^*$ e σ_h em relação às funções de interpolação, com o que se obtém

$$\tilde{\sigma}^* = \tilde{A}^{-1} \int_{\Omega} \tilde{N}^T \sigma_h d\Omega \quad (I-13)$$

sendo \tilde{A} a matriz quadrada simétrica

$$\tilde{A} = \int_{\Omega} \tilde{N}^T \tilde{N} d\Omega \quad (I-14)$$

Como a equação (I-7 c) pode ser escrita na forma

$$\|e\|^2 = \int_{\Omega} (\underline{\sigma} - \underline{\sigma}_h)^T \underline{D}^{-1} (\underline{\sigma} - \underline{\sigma}_h) d\Omega \quad (I-15)$$

substituindo-se $\underline{\sigma}$ (tensões exatas) por $\underline{\sigma}^*$ tem-se

$$\|e\|^2 = \int_{\Omega} (\underline{\sigma}^* - \underline{\sigma}_h)^T \underline{D}^{-1} (\underline{\sigma}^* - \underline{\sigma}_h) d\Omega \quad (I-16)$$

A diferença entre o campo de tensões suavizadas e o campo de tensões obtido pelo MEF, base desta estimativa de erro, em uma malha de três elementos reticulados lineares é mostrada na FIG. I-4.

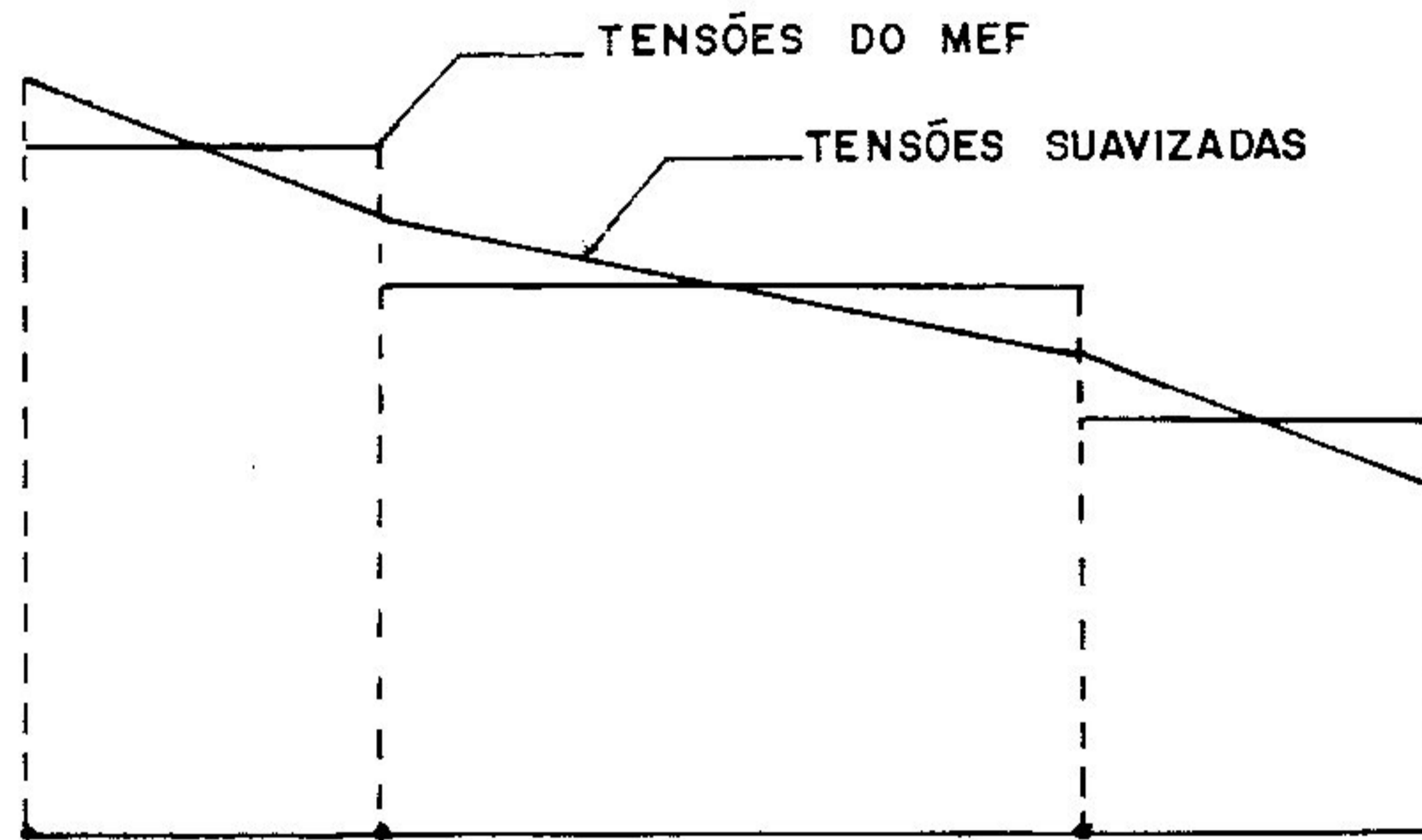


FIG. I-4 Tensões do MEF e tensões suavizadas em malha de três elementos reticulados lineares.

Este estimador, além de avaliar com bastante precisão os erros locais, fornece uma boa estimativa do erro

global no domínio Ω , que pode ser obtido pela raiz quadrada da soma das contribuições de todos os elementos.

$$\|e\|^2 = \sum_{i=1}^m \|e\|_i^2 \quad (I-17)$$

Neste trabalho é utilizado o conceito de percentagem relativa de erro

$$\eta = \frac{\|e\|}{\|u\|} \times 100\% \quad (I-18)$$

que possui uma interpretação física mais direta que o valor absoluto da norma de energia, e tende para zero quando o erro também tende para zero.

No sentido de obter uma medida da melhora da malha em um processo adaptativo, Las Casas-1988 utilizou um índice de performance definido como

$$IP = \frac{\max e_i}{\min e_i} \quad (I-19)$$

que tende para a unidade quando a malha converge para a configuração ótima. Este índice foi por ele utilizado como critério de parada do processo adaptativo, após a definição do limite de aceitação da solução em elementos finitos.

Uma questão importante é quando um determinado estimador de erro "a posteriori" ϵ pode ser considerado assintoticamente correto. Segundo Ewing-1990, sendo $\|e\|$ uma determinada norma do erro real e θ a razão entre ϵ e $\|e\|$,

$$\theta = \frac{\epsilon}{\|e\|} \quad , \quad (I-20)$$

o estimador do erro ϵ é considerado assintoticamente correto se $\theta \rightarrow 1$ quando $\|e\| \rightarrow 0$, ou seja, quando a dimensão da malha tende a zero ou o grau do polinômio de interpolação tende ao infinito. Em aplicações práticas é suficiente que $|\theta-1|$ seja pequeno, isto é, menor que 0,2 quando o erro $\|e\|$ for da ordem de 10% da norma da solução e que decresça quando

o erro decrescer. É preferível, de qualquer maneira, que θ seja maior que a unidade porque, neste caso, o erro verdadeiro estará superestimado.

No que se refere à aplicação dos processos H, P e H-P, é importante observar a grande diferença que existe entre eles com relação à convergência. No processo H, $h < 1$ é o parâmetro e p é a constante. Se θ da equação (I-20) satisfaz

$$\|\theta\| \leq C(p) h^s \quad (I-21)$$

para algum $s > 0$, C dependerá fortemente de p : $p \rightarrow \infty$ quando $C \rightarrow \infty$. No processo P, $p > 1$ é o parâmetro e h é a constante. Portanto

$$\|\theta\| \leq C(h) p^s \quad (I-22)$$

No processo H-P, tanto h como p são parâmetros. Sendo N o número de graus de liberdade,

$$\|\theta\| \leq C \exp(-N)^s \quad (I-23)$$

onde $s > 0$ e C é independente tanto de p como de h . Portanto, teoricamente, pode-se neste caso obter uma convergência exponencial contra uma convergência polinomial nos dois outros casos (Ewing-1990).

O cálculo do erro na norma de energia, no caso de problemas elípticos, corresponde a uma percentagem pequena do tempo total de solução. À utilização de diferentes normas no cálculo do erro podem corresponder custos computacionais muito diferentes e, quanto maior a ordem do elemento, mais tempo será gasto na estimativa do erro (Noor e Babuska-1987).

A estimativa do erro em uma variável específica, tal como tensões ou deslocamentos, algumas vezes se torna necessária. Estas estimativas são consideravelmente mais difíceis de serem feitas do que a avaliação do erro na norma de energia. No caso de não se dispor de estimativas desse tipo, pode-se resolver o problema calculando-se o erro na norma de energia e submetendo a solução a testes

posteriores tais como a verificação do equilíbrio de elemento por elemento e do equilíbrio de ações e reações (Noor e Babuska-1987).

I-4 Método Multimalha de Solução de Sistemas de Equações

Um aspecto importante a ser considerado na implementação de processos adaptativos para malhas de elementos finitos é a escolha do algoritmo a ser utilizado na solução dos sistemas de equações. As versões H, P e R geralmente se mostram mais eficientes quando são empregados processos iterativos de solução de sistemas, nos quais a solução de uma malha é utilizada como aproximação inicial na solução da malha mais refinada.

I-4-1 Idéias básicas do Método Multimalha

a) Erro algébrico, resíduo e equação residual. Correção pelo resíduo.

A solução de um sistema de equações obtida através de um processo iterativo qualquer é uma solução aproximada. A diferença entre a solução obtida por um processo direto, tratada neste item por solução exata, e a obtida por um processo iterativo é chamada erro algébrico (e_h). O problema-modelo aqui considerado é a equação $\underline{K} \underline{u}_h = \underline{f}$, sendo \underline{u}_h a solução exata. Ao se resolver esta equação por um processo iterativo, a solução aproximada obtida será denominada \underline{v} e o erro algébrico desta solução será então

$$\underline{e}_h = \underline{u}_h - \underline{v} \quad (I-24)$$

cuja magnitude pode ser avaliada por alguma norma padrão, tal como

$$\underline{u}_h = \underline{v} + \underline{e}_h \quad (\text{I-29})$$

b) Características da convergência nos processos iterativos básicos. Correção pela malha grossa.

Os processos iterativos comuns podem ser descritos na forma geral

$$\underline{v}^{(1)} = \underline{R} \underline{v}^{(0)} + \underline{g} \quad (\text{I-30})$$

onde \underline{R} é a matriz de iteração, específica do método utilizado. No caso do Método de Jacobi, sendo \underline{D} , $-\underline{L}$ e $-\underline{U}$ as partes diagonal, triangular inferior e triangular superior, respectivamente de \underline{K} ,

$$\underline{K} = \underline{D} - \underline{L} - \underline{U} \quad (\text{I-31})$$

$$\underline{R} = \underline{D}^{-1} (\underline{L} + \underline{U}) \quad (\text{I-32})$$

$$\underline{g} = \underline{D}^{-1} \underline{R} \quad (\text{I-33})$$

Portanto,

$$\underline{v}^{(1)} \leftarrow \underline{D}^{-1} (\underline{L} + \underline{U}) \underline{v}^{(0)} + \underline{D}^{-1} \underline{R} \quad (\text{I-34})$$

No método de Gauss-Seidel, que resulta do Jacobi com a utilização das novas aproximações tão logo elas sejam obtidas, tem-se

$$\underline{R} = (\underline{D} - \underline{L})^{-1} \underline{U} \quad (\text{I-35})$$

$$\underline{g} = (\underline{D} - \underline{L})^{-1} \underline{R} \quad (\text{I-36})$$

Portanto,

$$\underline{v}^{(1)} \leftarrow (\underline{D} - \underline{L})^{-1} \underline{U} \underline{v}^{(0)} + (\underline{D} - \underline{L})^{-1} \underline{R} \quad (\text{I-37})$$

A convergência ocorre somente se a solução do problema original $\tilde{K} \tilde{u}_h = \tilde{f}$ não for mudada pela iteração,

$$\tilde{u}_h = \tilde{R} \tilde{u}_h + \tilde{g} \quad (I-38)$$

Com $\tilde{e}_h = \tilde{u}_h - \tilde{v}$, subtraindo (I-30) de (I-38) tem-se

$$\tilde{e}_h \leftarrow \tilde{R} \tilde{e}_h \quad (I-39)$$

o que mostra que a evolução do erro é determinada somente pela matriz de iteração e pelo erro inicial.

No método de Jacobi, se h é o espaçamento da malha e $\rho(\tilde{R})$ é o raio espectral ($= \max_{1 < i < n} |\mu_i(\tilde{R})|$, μ_i são os autovalores de \tilde{R}), considerando a simetria de $\tilde{R} = \tilde{I} - (h^2/2) \tilde{K}$, pode-se limitar a norma euclidiana do novo erro, de maneira que

$$\|\tilde{e}_{(\text{nov})}\| \leq \|\tilde{R}\| \|\tilde{e}\| = \rho(\tilde{R}) \|\tilde{e}\| \quad (I-40)$$

Definindo como fator de convergência o valor

$$\|\tilde{e}_{(\text{nov})}\| / \|\tilde{e}\|, \quad (I-41)$$

a equação (I-40) mostra que o raio espectral é o limite para este fator e que o erro decresce a cada passo da iteração desde que $\rho(\tilde{R}) < 1$.

Sendo \tilde{R} um polinômio em \tilde{K} , o espectro $\lambda(\tilde{R})$ consiste nos auto-valores μ_i que podem ser escritos como

$$\mu_1 \cong 1 - \frac{\Pi^2 h^2}{2}, \quad \mu_2 \cong 1 - 2\Pi^2 h^2, \dots, \quad \mu_n \cong -1 + \frac{\Pi^2 h^2}{2} \quad (I-42)$$

onde a inequação $\rho(\tilde{R}) < 1$ é satisfeita.

Analisando-se as equações (I-42) pode-se observar que os auto-valores associados aos modos de baixa frequência (suaves) ($i \ll n$) têm valores próximos de 1, enquanto os associados aos modos de alta frequência (oscilatórios) ($i \cong n$) têm valores próximos a -1. Um raio

espectral muito próximo de 1 resulta em uma convergência pobre, o que justifica o fato do método de Jacobi apresentar uma boa redução do erro nas primeiras iterações quando estará reduzindo as componentes oscilatórias do erro e, à medida que se prosseguem as iterações, o desempenho piora sensivelmente. Pode-se concluir, portanto, que o método é eficiente em reduzir as componentes oscilatórias, mas não as componentes suaves do erro.

A mesma análise pode ser feita para outros processos iterativos comuns, chegando-se a conclusões em tudo semelhantes às obtidas para o método de Jacobi (Briggs-1987 e Briggs e McCormick-1987).

Considerando a equação $\underline{K} \underline{u}_h = \underline{f}$, o termo \underline{f} influi na equação (I-39) apenas pela via do erro inicial \underline{e}_h . Como este erro é arbitrário, a análise de métodos iterativos estacionários lineares gerais pode se restringir à equação homogênea $\underline{K} \underline{u}_h = \underline{0}$ com uma aproximação inicial arbitrária \underline{v} . Assim, como $\underline{u}_h = \underline{0}$, de (I-29) $\underline{e} = -\underline{v}$, o que significa que a aproximação corrente é também o erro real.

As considerações feitas até agora permitem observar que \underline{K} e \underline{R} possuem os mesmos auto-vetores, os quais podem ser representados pelos modos de Fourier (análise espectral)

$$v_j = \text{sen} \left(\frac{jk\pi}{N} \right), \quad 1 \leq j \leq N \quad \text{e} \quad 1 \leq k \leq N-1 \quad (\text{I-43})$$

onde N = número de intervalos na malha de um problema em 1-D

j = componente do vetor \underline{v} (nó da malha)

k = número de onda (número de meias-ondas de seno que constituem \underline{v} no domínio do problema)

(Briggs-1987).

A FIG. I-5 ilustra os modos v_1 , v_2 e v_3 em um problema uni-dimensional com espaçamento de malha constante.

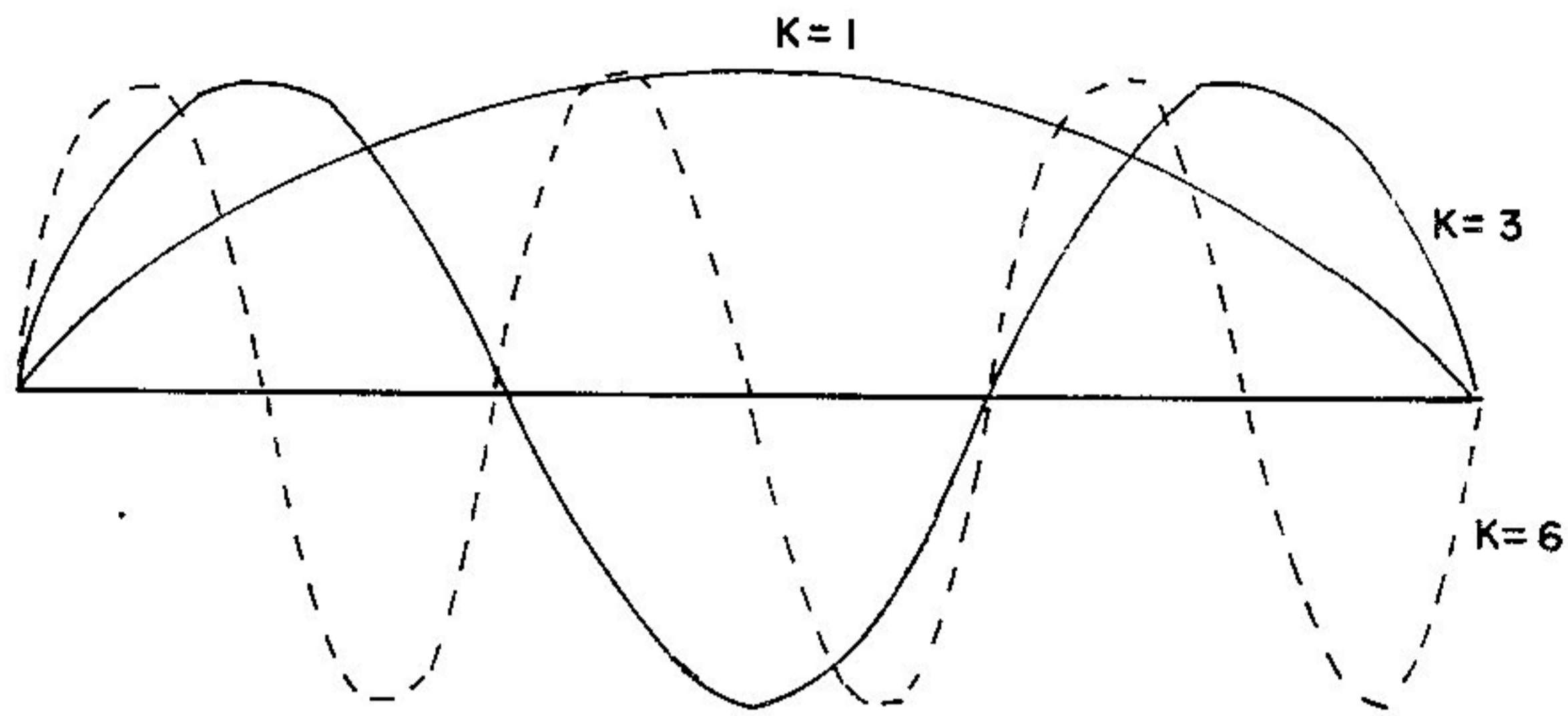


FIG. I-5 Modos $v_j = \text{sen}\left(\frac{jk\pi}{N}\right)$, $0 \leq j \leq N$, com números de onda $k=1,3,6$. O modo k consiste em $k/2$ ondas de seno completas no intervalo (Briggs-1987).

A FIG. I-6 mostra como o método de Jacobi ponderado aplicado à equação homogênea se comporta com relação a certas componentes do erro.

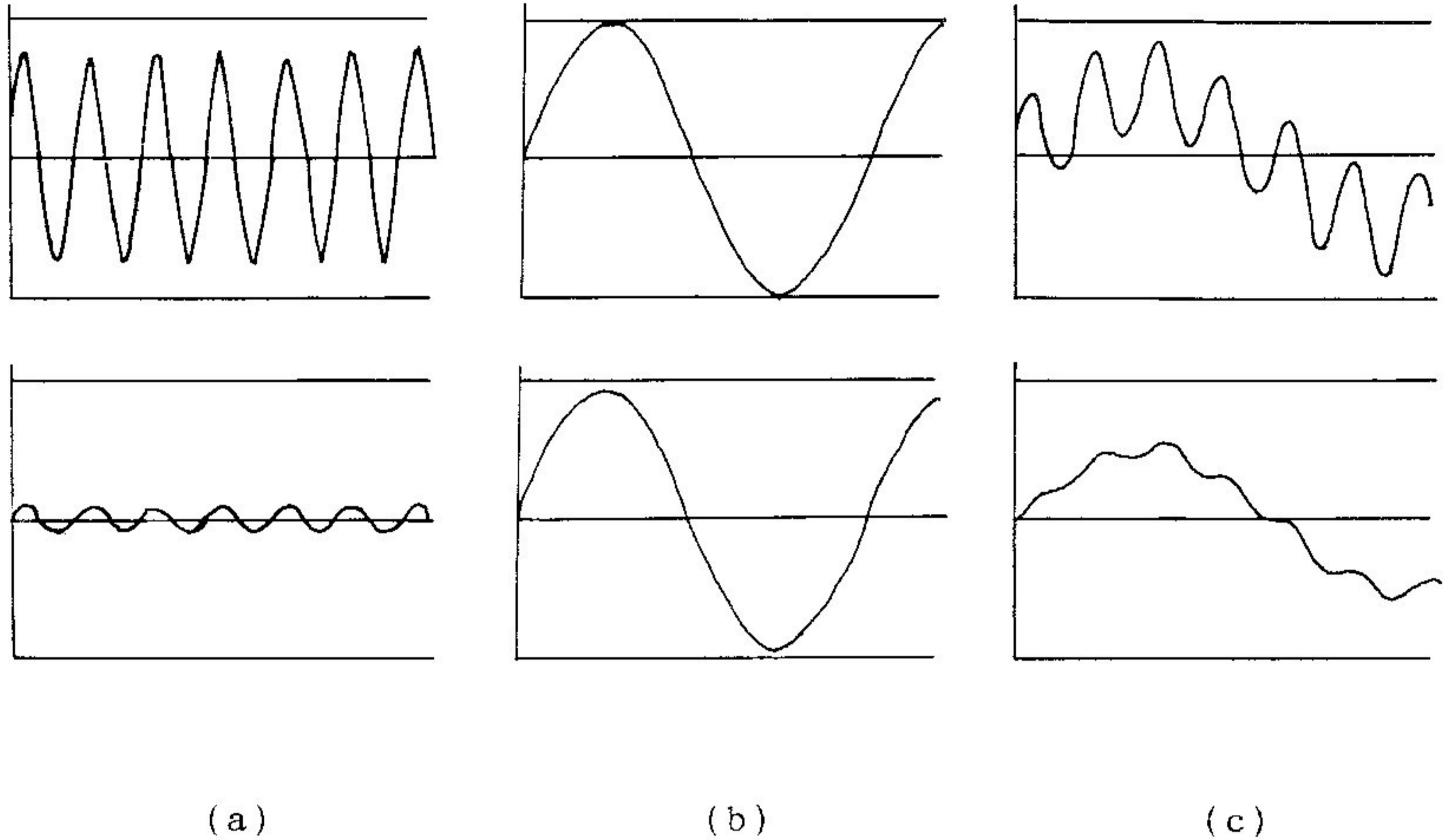


FIG. I-6

- a) Em componentes de frequência oscilatória do erro, 10 iterações produziram um decréscimo significativo na solução.
- b) Em componentes de frequência suave do erro, 10 iterações produziram um decréscimo insignificante na solução.
- c) Em uma combinação de componentes suave e oscilatória, 10 iterações produziram uma redução significativa na componente oscilatória enquanto a componente suave permaneceu praticamente inalterada (Briggs e McCormick-1987).

Trabalhando em 1-D numa malha Ω^h , de forma mais geral uma função arbitrária $u \in H^h$ pode ser representada por uma série de Fourier na forma

$$v_j = \sum_k c_k e^{i\theta_j} \quad (\text{I-44})$$

onde $i = \sqrt{-1}$. Apenas poucos modos n são necessários para a sua representação. Os números de onda θ_k têm a forma $2\pi k/(n+1)$ para $-n/2 \leq k \leq n/2$ e correspondem aos modos de Fourier com comprimentos de onda $\lambda_k = \infty, 1, 1/2, \dots, 4h, 2h$. Observa-se que variações na escala menores que $2h$ não podem ser resolvidas em Ω^h .

É interessante notar que uma componente suave em Ω^h é oscilatória em Ω^{2h} (FIG. I-7).

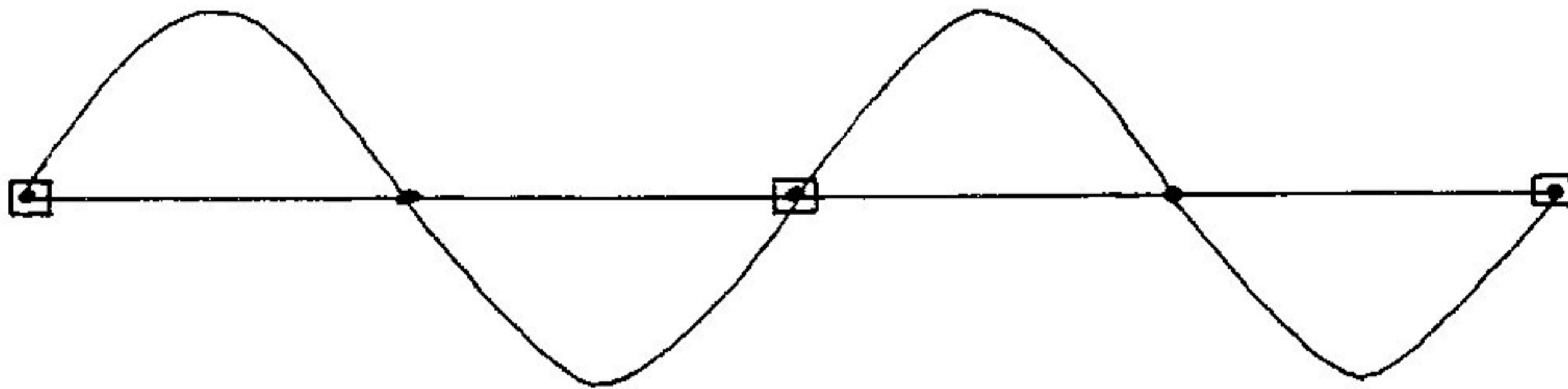


FIG. I-7 Uma componente suave em Ω^h , representada pelos pontos \square , é oscilatória em Ω^{2h} , representada pelos pontos \cdot e $+$.

Portanto, um método de relaxação como Jacobi, ao eliminar os modos oscilatórios do erro em Ω^h , geralmente deixa modos suaves que são ainda oscilatórios em Ω^{2h} . Este fato sugere que a aplicação repetida da relaxação em malhas sucessivamente mais grossas deve eliminar progressivamente todas as componentes do erro, procedimento este que é conhecido como "correção pela malha grossa".

c) Iteração encaixada (nested iteration)

A busca de uma forma de melhorar o desempenho de métodos iterativos básicos leva naturalmente à idéia de melhorar o valor inicial de \tilde{v} a um custo baixo. Uma estratégia há muito conhecida é a de, estando em questão uma malha Ω^h , inicialmente relaxar numa malha mais grossa Ω^{2h} e interpolar os resultados para Ω^h que seriam, aí, utilizados como valores iniciais.

Denominando I_{2h}^h o operador de transferência dos resultados da malha grossa para a fina, tem-se

$$\tilde{v}^h = I_{2h}^h \tilde{v}^{2h} \quad (I-45)$$

Existem várias formas de se fazer esta transferência, com operadores I_{2h}^h diferentes. Uma forma comumente utilizada é construir a malha Ω^{2h} contida na malha Ω^h mantendo-se nós alternados de maneira que os valores $\tilde{v}^{(0)}$ de Ω^h podem ser obtidos por interpolação linear dos valores \tilde{v} de Ω^{2h} (FIG. I-8).

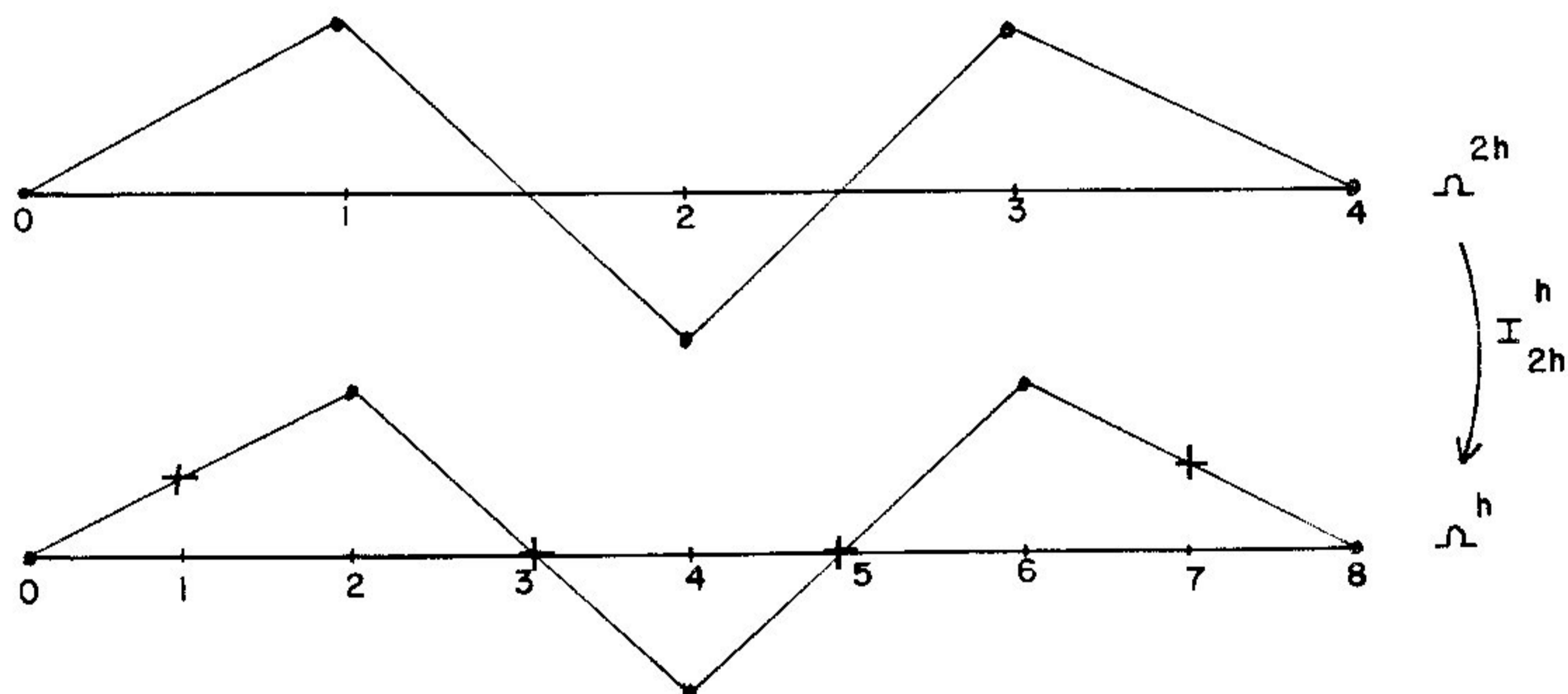


FIG. I-8 Um vetor \tilde{v}^{2h} (.) interpolado para um vetor \tilde{v}^h em uma malha mais fina (. e +) através de interpolação linear $\tilde{v}^h = I_{2h}^h \tilde{v}^{2h}$ em 1-D.

A utilização repetida de resultados de relaxações em malhas grossas como aproximações iniciais em malhas cada vez mais finas, umas nas outras, é o procedimento conhecido como iteração encaixada.

O processo é interessante mas possui algumas limitações importantes. Uma delas se refere às características da solução exata u_h que, caso seja oscilatória, os valores interpolados de resultados obtidos em uma malha mais grossa (que levarão a uma solução inicial necessariamente suave na malha fina) se constituirão em uma aproximação que não melhorará significativamente o processo. Outra limitação é dada pelo fato de que as taxas de convergência assintótica são geralmente independentes da aproximação inicial, o que faz com que taxas lentas apareçam rapidamente mesmo com a utilização da iteração encaixada. Este fato é significativo principalmente quando é necessário um alto grau de precisão, caso em que é imprescindível uma computação eficiente (Briggs e McCormick-1987).

I-4-2 O Método Multimalha

As três idéias apresentadas na seção anterior - correção pelo resíduo, correção pela malha grossa e aceleração por iteração encaixada - apresentam aspectos que limitam a sua eficiência como técnicas de melhora do desempenho de processos iterativos comuns, limitações estas que são, como visto, de natureza diferente umas das outras. Por outro lado, os aspectos positivos destas técnicas atuam no desempenho do método por três vias distintas e justificaram o emprego destas técnicas separadamente por longo tempo.

A correção pela malha grossa, usada recursivamente para a eliminação das componentes suaves do erro, foi introduzida por Fedorenko em 1962. Bakhvalov, em 1966, indicou a possibilidade da utilização da técnica da iteração encaixada conjuntamente com a correção pela malha grossa.

A combinação adequada das três técnicas resultou no Método Multimalha de solução de sistemas de equações, cujos elementos fundamentais foram desenvolvidos mais sistematicamente em 1976 por Hackbush que também apresentou uma teoria geral de convergência para o método.

A partir de 1977, o Método Multimalha vem ganhando progressiva aceitação e sua utilização foi estendida a problemas de diversas naturezas tais como dinâmica dos fluidos, fluxos transônicos, choques, equações de Euler e Navier-Stokes, problemas elípticos, parabólicos e não-elípticos dinâmicos, problemas de valor de contorno lineares e não-lineares. Além de equações diferenciais e algébricas, equações integrais também podem ser eficientemente resolvidas pelo método.

No caso de problemas-padrão (problemas elípticos simples de segunda ordem em duas dimensões), o método tem se mostrado comprovadamente mais eficiente que os métodos iterativos comuns e, no mínimo, competitivo com os métodos diretos (Stüben e Trottenberg-1981). A sua aplicação ao Método dos Elementos Finitos vem recebendo cada vez mais atenção de pesquisadores como A. Brandt, O. C. Zienkiewicz, R. Bank, J. Z. Zhu e outros, principalmente quando da utilização de processos adaptativos de melhora da malha.

A essência do Método Multimalha (MM) é a utilização repetida da iteração encaixada intercalada com a relaxação em Ω^h , isto é, iteração encaixada na equação residual (I-28) para reduzir as componentes suaves de \underline{e}^h e relaxação na equação original $\underline{K} \underline{v} = \underline{f}$ para reduzir as componentes oscilatórias.

a) Principais versões

O MM possui várias versões que diferem entre si basicamente no número de níveis de malha utilizados e na sequência empregada.

A versão básica é constituída de dois níveis de malha e é apresentada esquematicamente a seguir.

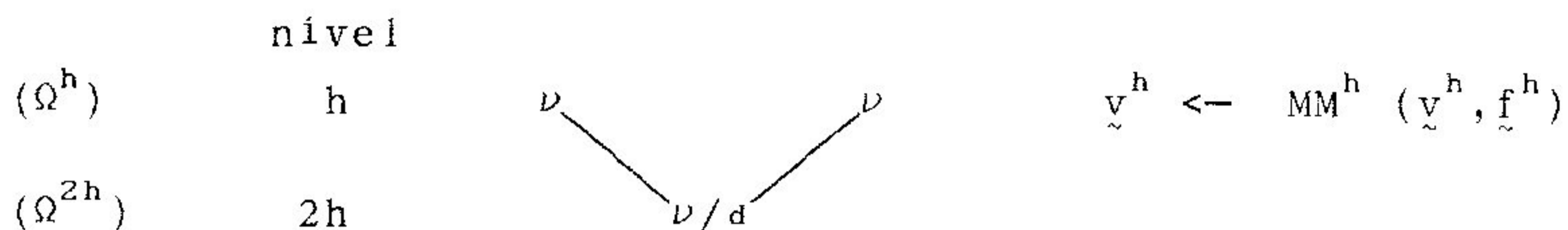
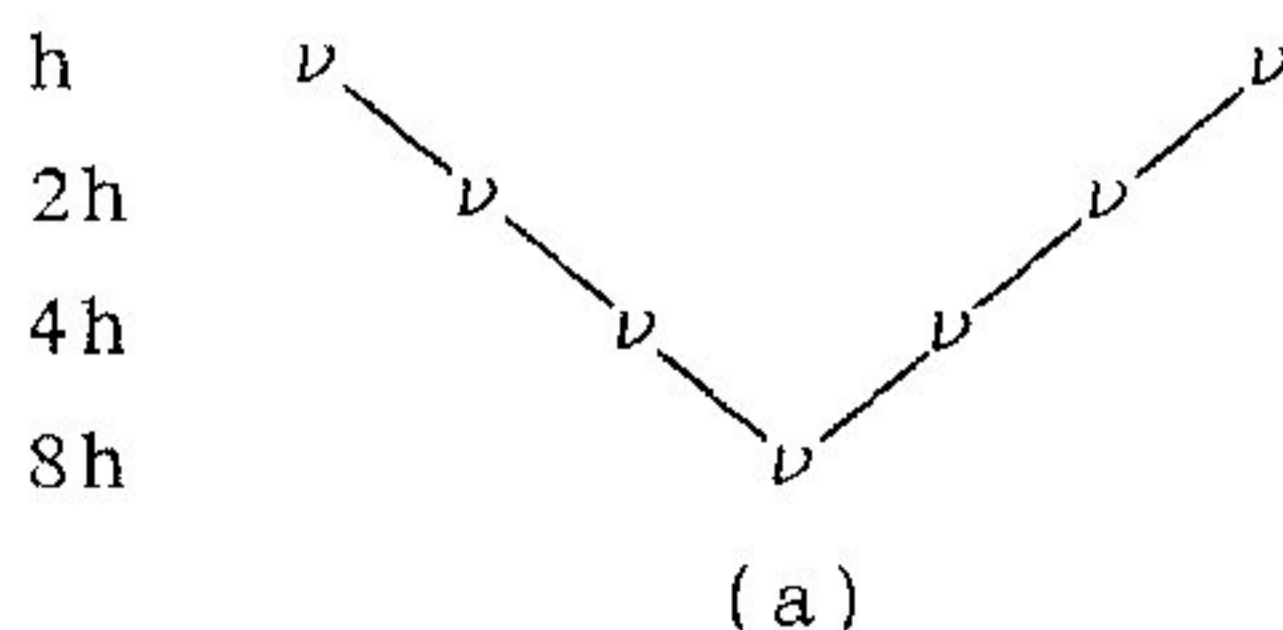


FIG. I-9 Versão básica do Método Multimalha

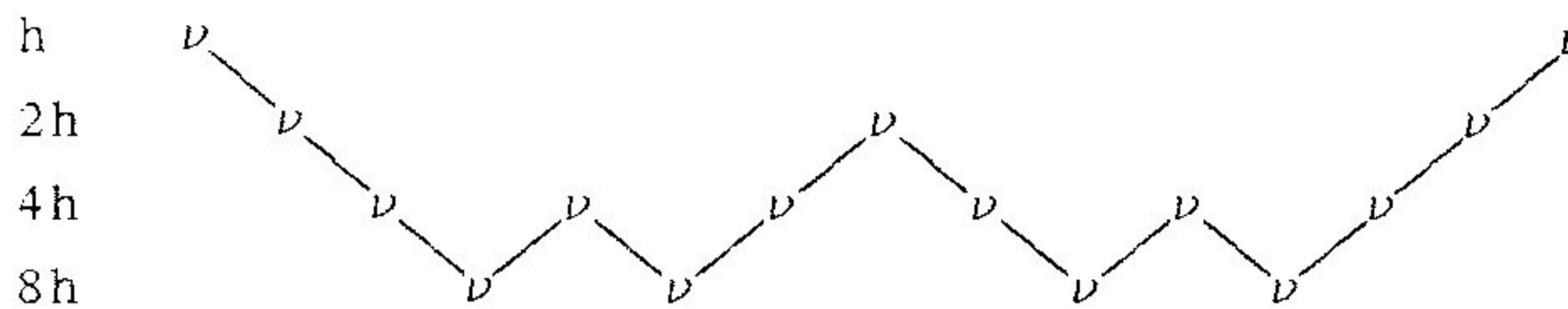
- PASSO 1: (pré-relaxação): Relaxação em $\tilde{K}^h \tilde{v}^h = \tilde{f}^h$, em Ω^h , com ν ciclos e aproximação inicial $\tilde{v}^{\wedge h}$.
- PASSO 2: (solução da malha grossa): Cálculo do resíduo $\tilde{r}^h = \tilde{f}^h - \tilde{K}^h \tilde{v}^h$; transferência do resíduo para a malha grossa (Ω^{2h}), $\tilde{r}^{2h} = \tilde{I}_h^{2h} \tilde{r}^h$; solução de $\tilde{K}^{2h} \tilde{e}^{2h} = \tilde{r}^{2h}$ por processo direto (d) ou iterativo, neste caso com $\tilde{e}^{\wedge 2h} = 0$ e ν ciclos.
- PASSO 3: (correção pela malha grossa): Transferência do erro para a malha fina $\tilde{e}^h = \tilde{I}_{2h}^h \tilde{e}^{2h}$; correção do valor anterior de \tilde{v}^h , $\tilde{v}^{\wedge h} = \tilde{v}^h + \tilde{e}^h$.
- PASSO 4: (pós-relaxação): Relaxação em $\tilde{K} \tilde{v}^h = \tilde{f}^h$ com aproximação inicial $\tilde{v}^{\wedge h}$ e ν ciclos.

O ciclo MM apresentado na FIG. I-9 pode ser repetido tantas vezes quantas forem necessárias para se atingir a tolerância desejada.

As versões (ou algoritmos) mais comumente utilizadas são os chamados ciclo V, ciclo W e ciclo FMV, os quais são apresentados sumariamente abaixo.

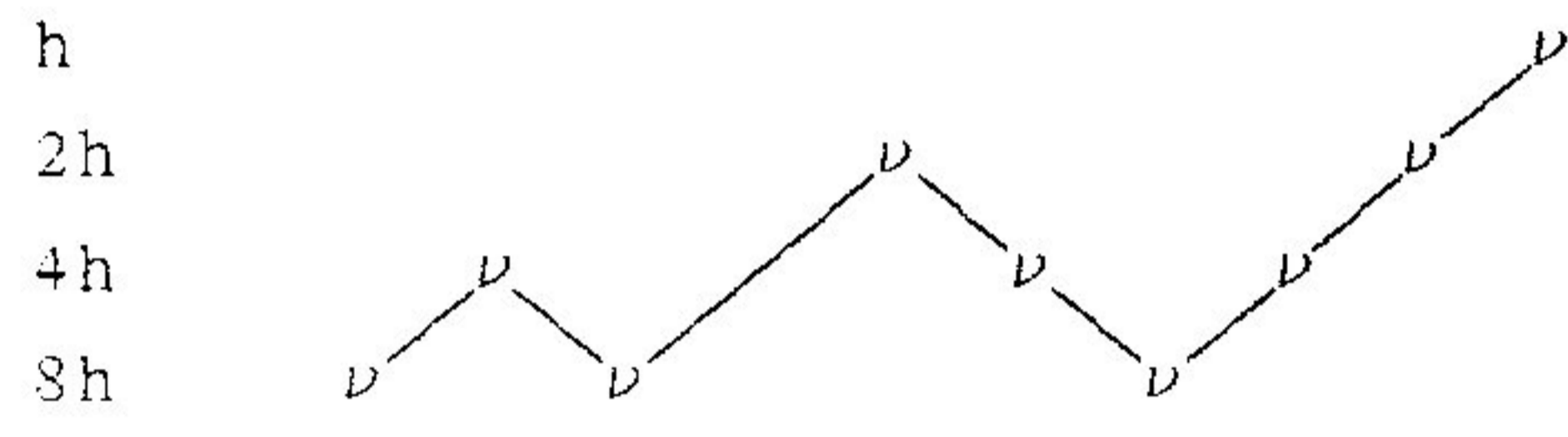


Ciclo V com 4 níveis



(b)

Ciclo W com 4 níveis



(c)

Ciclo FMV com 4 níveis

FIG. I-10 Versões comuns do Método Multimalha

A FIG. I-10 mostra algoritmos recursivos nos quais são usadas malhas cada vez mais grossas. A versão W pode ser interessante para aplicações mais gerais pois demanda menor esforço nas malhas mais grossas, produzindo melhores correções das malhas finas. A versão FMV visa a obtenção de uma aproximação inicial para o nível h antes que qualquer processamento tenha sido executado neste nível, o que, em princípio, melhora a eficiência do método (Briggs e McCormick-1987)

Para cada uma das três versões apresentadas, existem varios parâmetros a serem considerados, tais como o número de níveis da malha, o número de ciclos MM e o número de ciclos de relaxação em cada nível, além do critério empregado para a estimativa do erro algébrico e o erro permitido (tolerância).

Como operadores de transferência de valores dos resíduos de malhas finas para malhas grossas (operadores de restrição) os dois mais comumente utilizados são o de injeção e o de ponderação. Na injeção, os valores de τ nos

nós da malha grossa são exatamente os valores nos nós correspondentes da malha fina. A ponderação utiliza uma média ponderada dos valores, na malha fina, nos nós correspondentes e nos nós vizinhos mais próximos. O operador de restrição que transfere valores de Ω^h para Ω^{2h} é representado por \tilde{I}_h^{2h} .

A transferência dos valores dos erros de malhas grossas para malhas finas é feita através de operadores de interpolação (ou prolongamento), geralmente interpolação linear, e são representados por \tilde{I}_{2h}^h .

A matriz \tilde{K} de cada nível de malha pode ser obtida por operadores de transferência ou, principalmente no caso de malhas de elementos finitos, pela montagem da matriz em cada nível a partir das matrizes de rigidez elementares. Este procedimento tem a possível desvantagem de levar à falta de consistência entre operadores de interpolação e restrição, mas a operação de transferência de \tilde{K} pode, em grande número de casos, ser muito trabalhosa e não justificar-se quando comparada com a alternativa (Briggs-1987 e Parsons e Hall-1990).

b) Desempenho

A avaliação do desempenho do MM demanda a análise do custo de armazenamento de dados, do custo da aritmética computacional (ou custo computacional) e da convergência.

Quanto ao custo de armazenamento, de acordo com Briggs-1987, para problemas uni-dimensionais, o espaço de armazenamento necessário é menor que o dobro do necessário para a solução da malha fina por processos comuns e, para problemas de duas ou mais dimensões, o custo cai para menos de 4/3 do custo da solução da malha fina. Pode-se concluir que o custo de armazenamento de algoritmos MM, em relação aos processos tradicionais, decresce com o aumento da dimensão do problema.

O custo computacional é medido em unidades de trabalho UT que é o custo de um ciclo de relaxação na malha fina. Desprezando-se o custo das operações de transferência

interníveis, que é da ordem de 15% a 20% do custo total do ciclo MM, o custo computacional de um ciclo V com $\nu=1$ será

$$\frac{2}{1-2^{-d}} UT \quad (I-46)$$

Portanto, para problemas uni-dimensionais ($d=1$), o ciclo V simples custa $4UT$, para $d=2$ custa $(8/3)UT$ e para $d=3$ custa $(16/7) UT$.

O ciclo FMV, com $\nu=1$, tem o custo computacional igual a

$$\frac{2}{(1-2^{-d})^2} UT \quad (I-47)$$

Portanto, para $d=1$ o ciclo FMV custa $8UT$, para $d=2$ custa aproximadamente $(7/2)UT$ e para $d=3$ custa aproximadamente $(5/2)UT$.

A análise de convergência do MM não é uma tarefa simples e tem sido muito estudada nos últimos anos. Em linhas gerais, sendo E^h o erro global dado pela equação (I-4), uma medida de quanto a solução exata do problema discreto se aproxima da solução exata do problema original contínuo,

$$E_{h_i} = u(x_i) - u_{h_i} \quad 1 \leq i \leq N \quad (I-48)$$

este deve ser limitado em norma por

$$\|E_h\| \leq Kh^p \quad (I-49)$$

onde K é uma constante e $(p/2)+1$ é o número de malhas usado na solução.

A tolerância ϵ é definida como

$$\epsilon > \|u - v\| \quad (I-50)$$

onde o lado direito da equação é a norma do erro algébrico

dado pela equação (I-24).

A equação (I-50) pode ser satisfeita se

$$\|E_h\| + \|e_h\| < \varepsilon \quad (I-51)$$

onde e_h é dado pela equação (I-24).

Uma maneira de satisfazer a equação (I-51) é tomar

$$\begin{aligned} & \|E_h\| < \varepsilon/2 & (a) & \quad (I-52) \\ e & \|e_h\| < \varepsilon/2 & (b) & \end{aligned}$$

A primeira condição (a) mostra que se deve definir o espaçamento da malha (h) de tal forma que

$$h < h^* = \left(\frac{\varepsilon}{2K} \right)^{1/p} \quad (I-53)$$

onde h^* é denominado espaçamento crítico. Portanto, o erro global determina o espaçamento crítico da malha.

A segunda condição (b) determina o quanto a aproximação y deve se aproximar da solução exata do problema discreto u_h .

Portanto, se os ciclos de relaxação ou MM tiverem sido processados até que $\|e_h\| < \varepsilon/2$ numa malha Ω^h onde $h < h^*$, e tomando-se ε como a precisão da máquina utilizada, então pode-se afirmar que a convergência se deu para o nível de truncamento, isto é, para valores que têm erros da ordem dos erros introduzidos pelas limitações da máquina.

Segundo Briggs-1987, como o custo da convergência de um ciclo V simples é $O(N^d)$, isto é, da ordem de N^d , onde N^d é o número de pontos da malha; o custo da convergência para o nível de truncamento é $O(N^d \log N)$, que é comparável ao custo computacional dos melhores processos diretos de solução, quando da análise de problemas-modelo. O custo computacional do ciclo FMV é $O(N^d)$, que também é considerado um bom resultado.

Pode-se concluir que o número total de operações necessárias para se obter uma solução convergente é proporcional à dimensão do problema (número de graus de liberdade da malha mais fina), o que significa que o número de ciclos MM necessários para se alcançar a convergência é independente da dimensão do problema. Esta conclusão foi condicionada, por Parsons e Hall em 1990, à fixação de parâmetros apropriados na definição da variante do método utilizada, bem como ao problema analisado.

As considerações feitas neste item referem-se ao esquema fixo, no qual os parâmetros cíclicos, tais como ν , são fixados "a priori" e permanecem inalterados durante todo o processamento. Um outro esquema, muitas vezes utilizado, é o chamado esquema acomodativo, no qual os parâmetros são determinados durante o processamento e se adaptam aos parâmetros de convergência (Brandt-1981).

c) Aplicação do MM em processos adaptativos de malhas de elementos finitos

Uma limitação que tem sido alegada para a utilização do MM na solução de malhas de elementos finitos diz respeito ao tamanho do banco de dados gerado e à dificuldade na manipulação de tantos dados.

No caso dos processos adaptativos H e P, são geradas malhas mais refinadas que são comumente resolvidas por algum processo iterativo, onde se utiliza com frequência, já aqui, os conceitos de iteração encaixada (ver item I-2).

Alguns trabalhos têm sido desenvolvidos usando o MM na solução dos sistemas gerados nos processos adaptativos H e P com bons resultados. Craig e Zienkiewicz-1985, apresentaram um algoritmo MM para a versão P com base hierárquica. Zhu e Craig-1985 aplicaram à versão P com base hierárquica o método MM - ciclo N esquematizado a seguir.



FIG. I-11 Esquema N do método MM (Zhu e Craig-1985)

Os resultados obtidos foram bons e o algoritmo foi considerado eficiente. Stein, Bischoff, Brand e Plank-1985 apresentaram uma técnica de refinamento H global usando o método MM, para a solução de problemas de estado plano. Bank, Dupont e Yserentant-1988 utilizaram o MM para a versão P de base hierárquica em problemas elípticos de valor de contorno usando elementos finitos triangulares lineares, que se mostrou um método robusto, com uma taxa de convergência quase ótima. Devloo-1987 analisou um método adaptativo H-P usando MM para a simulação numérica de equações de Navier-Stokes compressíveis.

Parsons e Hall-1990 apresentaram um estudo detalhado da aplicação do método MM em problemas bi e tri-dimensionais resolvidos por elementos finitos. O trabalho não menciona processos adaptativos, mas a subdivisão feita nas malhas para a solução do sistema por MM é a mesma usada em um processo H global. São analisadas malhas uniformes e não uniformes. São estudados os ciclos V e W, com relaxação feita por Gauss-Seidel (GS) e gradiente conjugado pré-condicionado de Jacobi (JCG), transferência inter-níveis feita através de operadores de interpolação linear e injeção, matrizes K dos vários níveis obtidas por montagem e por operadores de transferência e condições de contorno simuladas através de molas nos apoios. A tolerância para convergência foi tomada como $\epsilon = 10^{-6}$; a convergência foi considerada atingida quando $R = \frac{\|\tilde{r}_f\|}{\|\tilde{f}_f\|} \leq \epsilon$ e foi utilizada a aproximação inicial $\hat{v} = 0$ na solução da malha mais fina.

Foram, neste trabalho, analisadas, dentre outras coisas, a sensibilidade do desempenho do MM a algumas características do problema em questão, tais como a quantidade de flexão presente e o grau de não uniformidade da malha, aos processos de relaxação utilizados e à forma de obtenção da matriz K . A eficiência do ciclo V foi comparada à do ciclo W e foi investigada a independência do número total de operações requerida em relação ao número de graus de liberdade do problema.

I-5 O Sistema para Desenvolvimento de Programas Baseados no Método dos Elementos Finitos - SDP

O SDP é um sistema computacional desenvolvido a partir da tese de doutoramento de Jayme Gouveia, que se destina a auxiliar a elaboração de programas aplicativos técnico-científicos que utilizam o MEF. Foi elaborado segundo conceitos de programação estruturada e de rede integrada de programas de modo a criar um ambiente computacional que dispõe de um sistema de gerenciamento de dados, de facilidades para depuração e testes de rotinas e de bibliotecas de procedimentos de cálculo com matrizes, solução de sistemas, cálculo de funções de interpolação, etc..Fornece também uma padronização para documentação de rotinas visando sua portabilidade (Gouveia-1987).

Os programas do SDP são organizados de acordo com níveis hierárquicos e cada um pode ser dividido em três etapas:

- iniciação dos procedimentos de gerenciamento e dos utiliários;
- gerenciamento e manipulação da memória de trabalho e do banco de dados;
- rotinas de cálculo propriamente ditas.

Seguindo esta divisão, um programa é organizado de acordo com os níveis mostrados na FIG I-12.

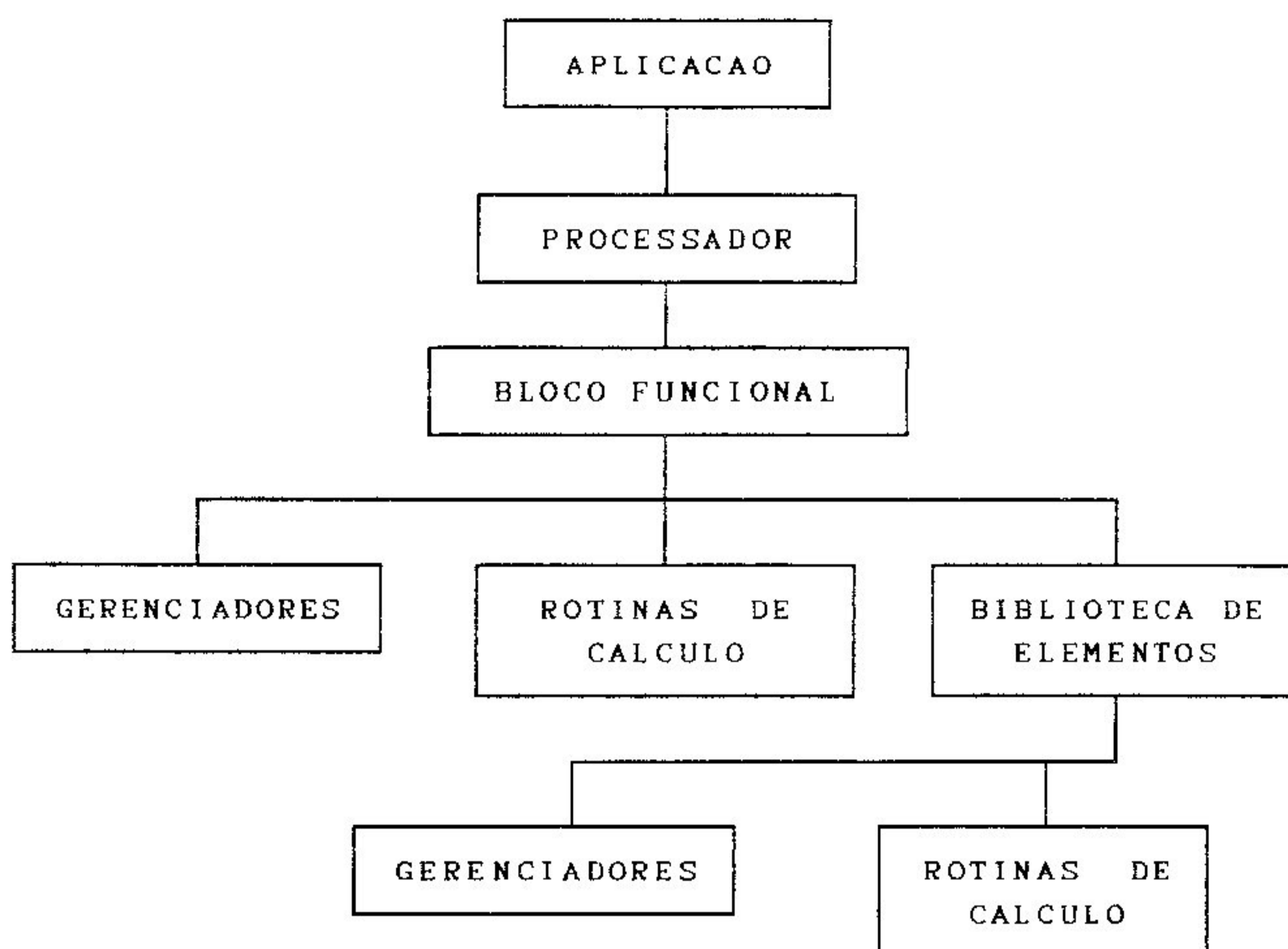


FIG.I-12 Organização dos programas no SDP

A aplicação é um conjunto de programas formando uma rede integrada, que permite a solução de um determinado problema (somente existe para efeito de organização).

O processador gerencia a execução de tarefas inerentes a uma dada aplicação e geralmente é dividido em três etapas: pré-processadores onde são gerados bancos de dados, processadores que trabalham os bancos de dados gerados na etapa anterior e geram novos bancos de dados e pós-processadores que calculam resultados derivados dos obtidos pelos processadores.

O bloco funcional é uma sub-rotina encarregada de gerenciar a execução de uma determinada tarefa através da chamada das rotinas de cálculo, biblioteca de elementos e gerenciadores. Gerencia a área de trabalho e o banco de dados e tem como função apenas a transferência das informações necessárias para a área de trabalho e dos resultados obtidos

nas rotinas de cálculo para o banco de dados.

Na sua versão básica para análise de elementos triangulares nos estados planos, o SDP é constituído de um programa aplicativo com quatro processadores, cada um com vários blocos funcionais e suas muitas rotinas de cálculo e gerenciamento. A FIG. I-13 apresenta o diagrama de estrutura do programa até o nível dos blocos funcionais.

PROCESSADOR

BLOCOS FUNCIONAIS

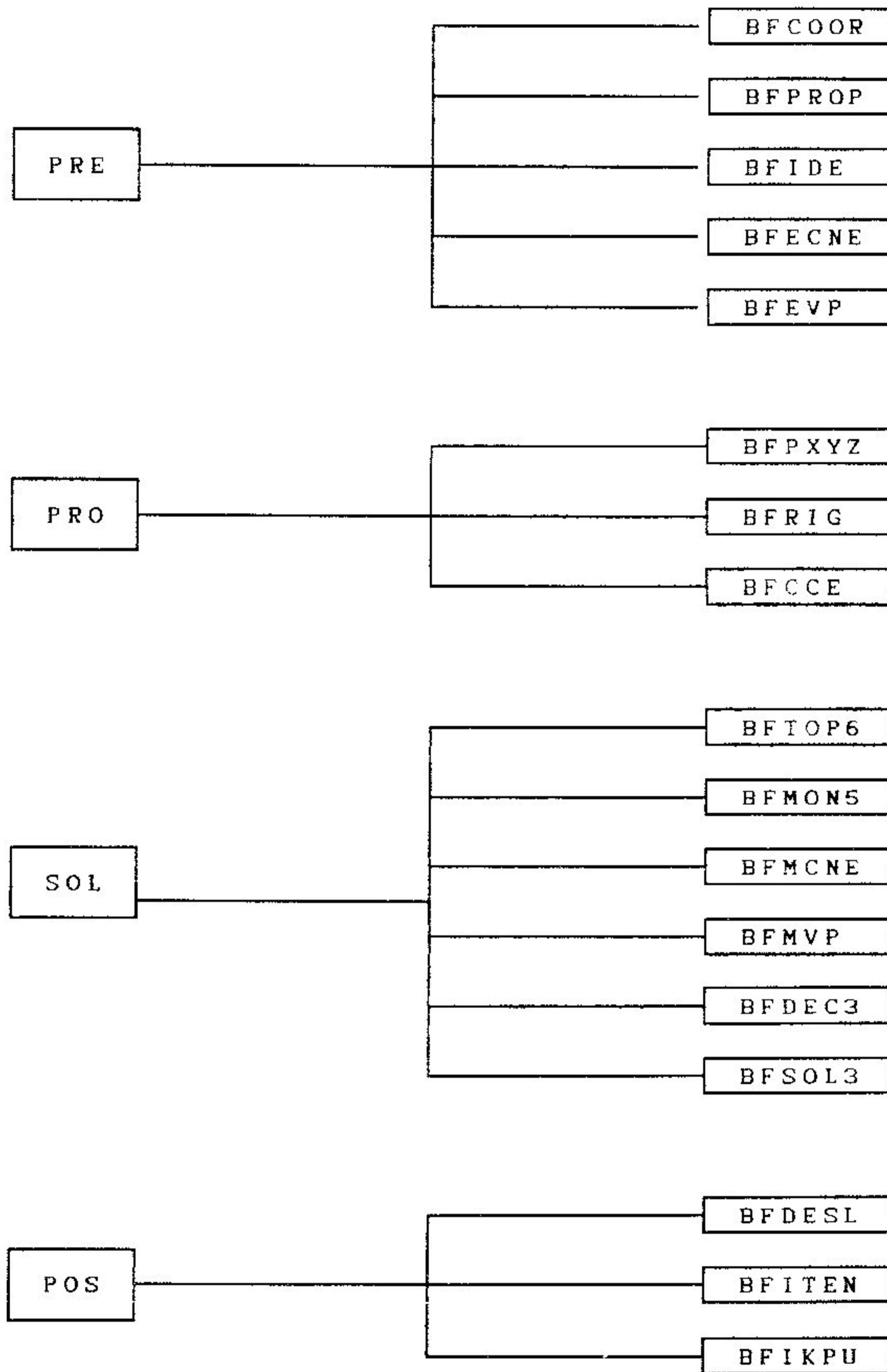


FIG. I-13 Diagrama de estrutura dos processadores da versão básica do SDP para análise elasto-linear em estado plano com elementos triangulares.

A seguir apresenta-se, sumariamente, a função dos processadores e blocos funcionais da FIG. I-10:

- * PRE: prepara um banco de dados para comunicação entre um arquivo de dados de entrada e um processador elasto-linear;
 - .BFCCOOR: lê e modifica a tabela de coordenadas;
 - .BFPPROP: lê e modifica tabelas de propriedades;
 - .BFIDE : lê e modifica matriz das condições de contorno;
 - .BFECNE: lê e modifica tabela de carregamento;
 - .BF EVP : lê e modifica tabela de valores prescritos;
- * PRO: calcula as matrizes de rigidez e solicitações elementares;
 - .BFXYZ: prepara dados para os grupos de elementos;
 - .BFRIG : calcula as matrizes de rigidez elementares;
 - .BFCCE : calcula os esforços elementares;
- * SOL: monta a matriz de rigidez global, o vetor de cargas equivalentes e resolve o sistema por eliminação de Gauss;
 - .BFTOP6: calcula a topologia da matriz de rigidez global;
 - .BFMON5: monta a matriz de rigidez global (triangular superior em sky-line);
 - .BFMCNE: monta vetor de cargas nodais equivalentes;
 - .BF MVP : introduz prescrições;
 - .BFDEC3: triangulariza o sistema;
 - .BFSOL3: resolve por retro-substituição o sistema triangularizado;
- * POS: calcula e imprime reações de apoio, tensões e o produto $\underline{K}\underline{v}$;
 - .BFDESL: imprime deslocamentos;
 - .BFITEN: calcula e imprime tensões;
 - .BFIKPU: imprime o produto $\underline{K}\underline{v}$.

As tabelas, na área de trabalho, são armazenadas em um único vetor pré-dimensionado, mantido sempre compactado através de gerenciadores.

O SDP tem sido utilizado em várias instituições e têm sido desenvolvidas, através dele, aplicações em

análise linear de estruturas, condução de calor, elasto-plasticidade incremental, análise dinâmica, análise limite e problemas de contato. Existem versões destes programas para computadores de grande porte, estações de trabalho e micro-computadores.

I-6 Objetivo do Trabalho

Pretendeu-se, neste trabalho, implantar no SDP um processo adaptativo R-H global para malhas de elementos finitos triangulares nos estados planos de tensão e deformação, utilizando o Método Multimalha de solução de sistemas de equações.

No processo R utilizou-se a estimativa de erro proposta por Zienkiewicz e Zhu-1987, o processo H foi realizado de forma global e uniforme e, para a solução dos sistemas de equações, foi utilizada uma versão do algoritmo N multimalha proposto por Zhu e Craig-1985 tendo como parâmetros os utilizados por Parsons e Hall-1990.

Foram obtidos resultados numéricos para alguns problemas simples, resultados estes que foram comparados com os obtidos no processamento das malhas originais e das malhas sujeitas ao processo R simples. O desempenho do MM foi comparado ao desempenho do processo iterativo Gauss-Seidel e da solução por eliminação de Gauss.

O processo adaptativo H global é, como visto, menos utilizado que aquele que executa refinamentos localizados e, neste, o MM apresenta maiores dificuldades de aplicação devido ao caráter não uniforme da subdivisão da malha. No processo misto R-H, onde se procede a uma adaptação R e, sobre a malha já com o erro equidistribuído, se procede a um refinamento H global, o banco de dados gerado na segunda etapa é em tudo semelhante ao que é necessário para a solução do sistema gerado pela malha mais fina quando se utiliza o MM. Pode-se prever, portanto, que a utilização simultânea dos

dois procedimentos - processo R-H global e MM - resultará em um processo eficiente e de custo computacional reduzido.

CAPÍTULO II

PROCESSO ADAPTATIVO R-H GLOBAL IMPLEMENTAÇÃO

Descreve-se neste capítulo a implementação de um processo adaptativo R-H global no sistema SDP, versão básica apresentada no item I-5, para elementos triangulares de três nós nos estados planos de tensão e deformação.

O banco de dados da fase H foi gerado de tal forma que pudesse ser utilizado diretamente na solução do sistema de equações da malha mais refinada pelo Método Multimalha.

II-1 Fase R - Processo R

Na fase R do processo adaptativo R-H foi utilizado o procedimento implantado no SDP por Las Casas em 1990. Este procedimento modificou a versão básica do SDP apresentada no item I-5, substituindo o pré-processador PRE por outro chamado PREAD e criando um novo processador, ADA, após o processador POS, de acordo com a FIG.II-1.

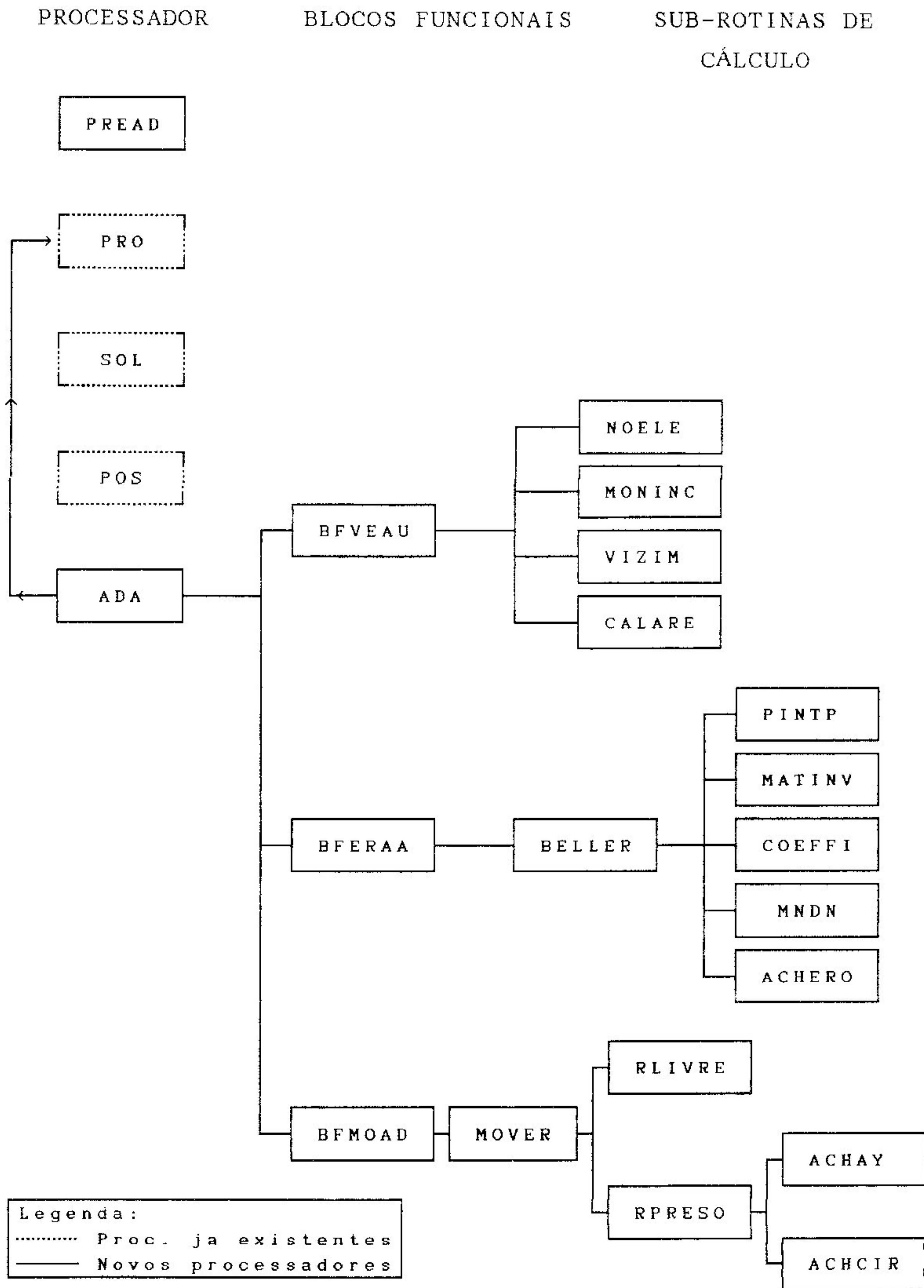


FIG. II-1 Diagrama de estrutura dos processadores criados para a Fase R por Las Casas em 1990.

A função de cada nova rotina da FIG. II-1 é descrita sumariamente a seguir.

* PREAD: pré-processador que se distingue do original PRE apenas na leitura e gravação de parâmetros do processo adaptativo e MM, que são:

- número máximo de ciclos R
- número de níveis de malha no MM
- número de iterações Gauss-Seidel (ν) a ser executado, por nível, no MM
- tolerância na fase R
- tolerância no MM;

* ADA: pós-processador que calcula o erro em cada elemento e move os nós no domínio;

.BFVEAU: monta matrizes auxiliares para o cálculo do erro e do deslocamento dos nós;

NOELE: monta os vetores NOELE e JINC que contêm os elementos que possuem um determinado nó e o número de ordem do último elemento de cada grupo, respectivamente;

MONINC: monta o vetor INCT que contém a incidência nodal dos elementos de todos os grupos;

VIZIM: monta o vetor IVIZ que contém os elementos vizinhos (que têm um lado em comum) de cada elemento;

CALARE: monta os vetores ARE e CGR que contêm a área e as coordenadas do baricentro, respectivamente, de cada elemento;

.BFERAA: calcula e grava o erro em cada elemento;

PINTP: fornece os pontos de integração e os pesos correspondentes para os elementos triangulares planos;

MATINV: calcula a inversa da matriz de propriedades elásticas;

COEFFI: fornece as funções de interpolação do elemento;

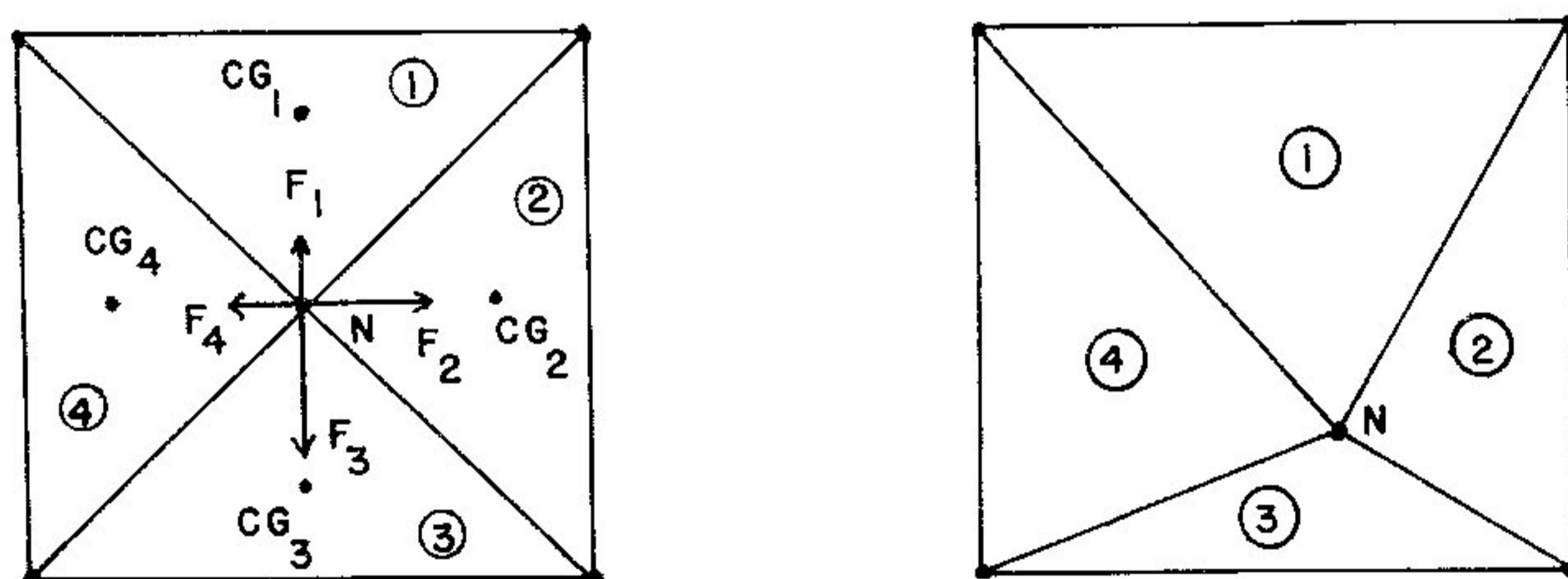
MNDN : calcula uma matriz auxiliar (MNDN);

ACHERO: calcula o erro em cada elemento utilizando o procedimento proposto por Zienkiewicz e Zhu-1987, descrito no item I-3 (equação I-16); a tensão em cada nó do elemento é tomada como a média das tensões neste nó referentes a todos os elementos que nele concorrem;

.BFMOAD: move os nós no domínio;

MOVER: gerencia a chamada das sub-rotinas de cálculo, de acordo com as restrições que o nó possa ter ao seu deslocamento;

RLIVRE: calcula as novas coordenadas dos nós que não têm qualquer restrição ao deslocamento R; este deslocamento é calculado considerando-se a atuação de "forças", em cada nó, proporcionais ao erro de cada elemento que o contém, na direção do baricentro dos elementos; o deslocamento resultante será correspondente à força resultante do sistema (FIG. II-2); com este procedimento se estará aumentando a densidade de elementos nas regiões onde o erro é maior, o que tende a equidistribuir o erro no domínio

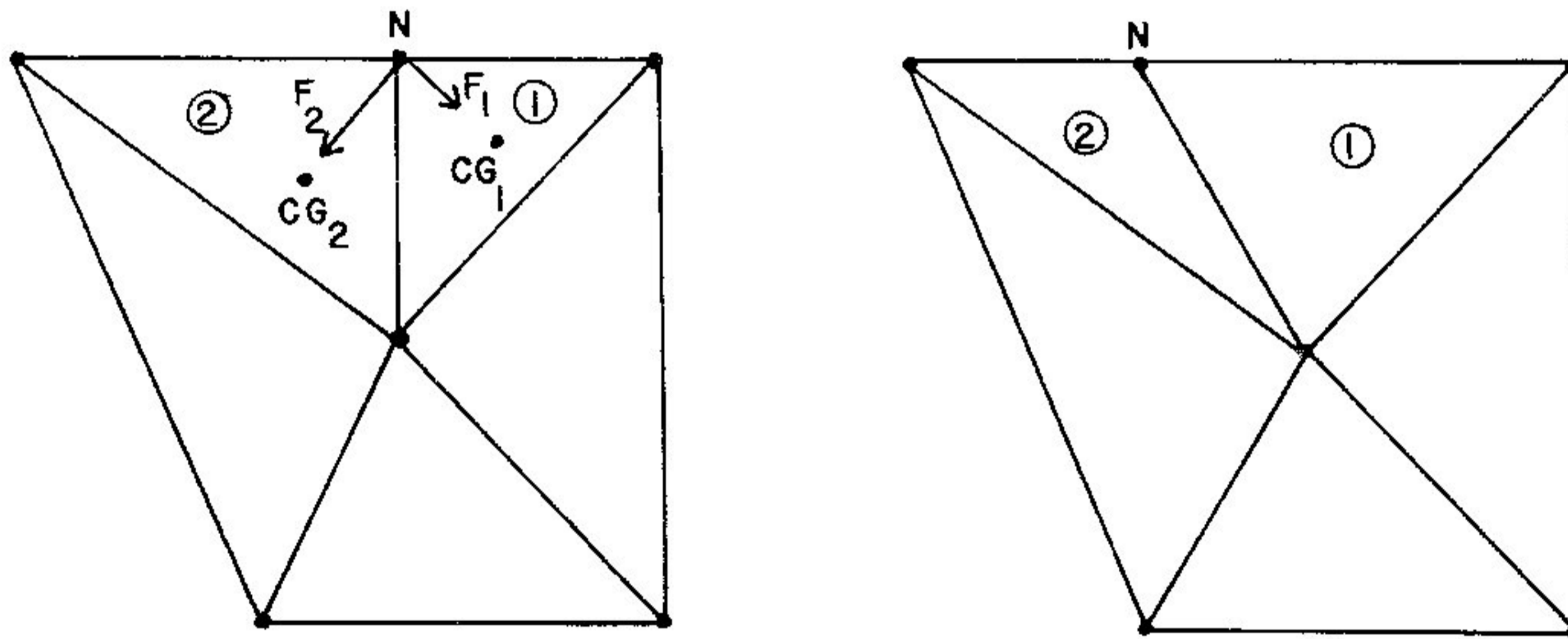


(a) posição inicial do nó N (b) posição final do nó N
 CG - posição do baricentro de cada elemento
 F - forças proporcionais ao erro de cada elemento

FIG. II-2 Procedimento usado na rotina RLIVRE para mover um nó.

RPRESO: calcula as novas coordenadas dos nós com restrições ao deslocamento (por exemplo, nós de contorno); o procedimento é o mesmo adotado na rotina RLIVRE, porém aqui o nó se moverá segundo a componente vetorial do deslocamento resultante na direção em que o nó é livre para se mover;

ACHAY: calcula o deslocamento quando ao nó é permitido mover-se apenas ao longo de uma reta ou de uma curva do segundo grau (FIG. II-3);



(a) posição inicial do nó N

(b) posição final do nó N

FIG. II-3 Deslocamento do nó N (nó de contorno), na rotina ACHAY.

ACHCIR: calcula o deslocamento quando ao nó é permitido mover-se apenas ao longo de uma circunferência (FIG. II-4);



(a) posição inicial do nó N (b) posição final do nó N

FIG. II-4 Deslocamento do nó N (nó de contorno)
na rotina ACHCIR.

Uma vez calculadas as novas coordenadas dos nós através do pós-processador ADA, os procedimentos PRO, SOL, POS e ADA são repetidos para a nova malha, de modo que esta é modificada novamente, assim sucessivamente, até que o erro seja considerado suficientemente equidistribuído no domínio. Neste momento, o procedimento é interrompido e tem-se como resultados finais os obtidos no último processamento POS.

II-2 Fase H - Implementação do Processo H

O refinamento H foi implementado através de um novo processador, GERA, que tem a função de gerar três níveis de malha (malha 1 (M1), malha 2 (M2), malha 3 (M3)) a partir da malha obtida após a fase R (malha 0 (M0)). O refinamento foi executado em três etapas, com o objetivo de gerar um banco de dados que pudesse ser diretamente utilizado na solução do sistema de equações da malha M3 pelo Método Multimalha.

II-2-1 Geração das novas malhas M1, M2, M3

a) Subdivisão das malhas

Os três novos níveis de malha foram gerados pela subdivisão, em cada nível, de cada elemento em quatro elementos semelhantes entre si e ao que os gerou, através da criação de um novo nó no ponto médio de cada lado do elemento original. Ao final do refinamento, cada elemento de M0 (elemento-pai) terá sido sub-dividido em 64 elementos-filhos (FIG. II-5).

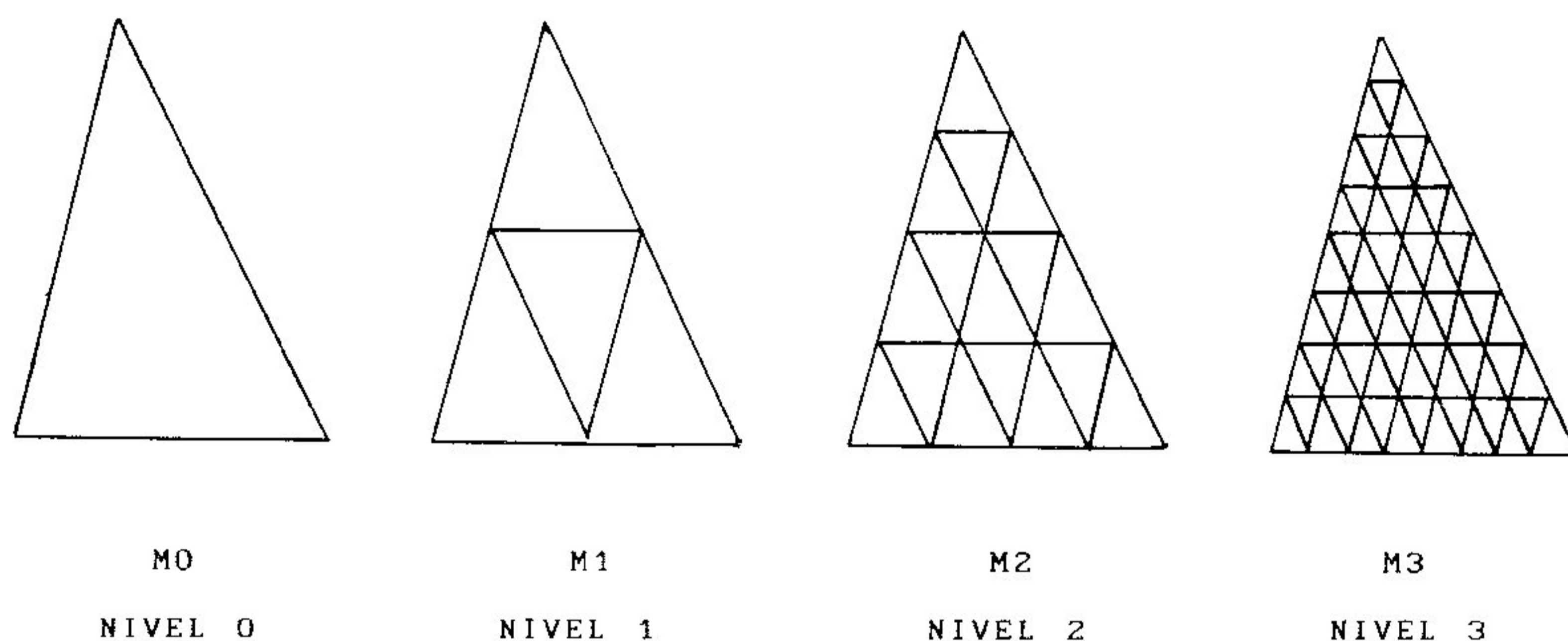


FIG. II-5 Refinamento global H em três etapas.

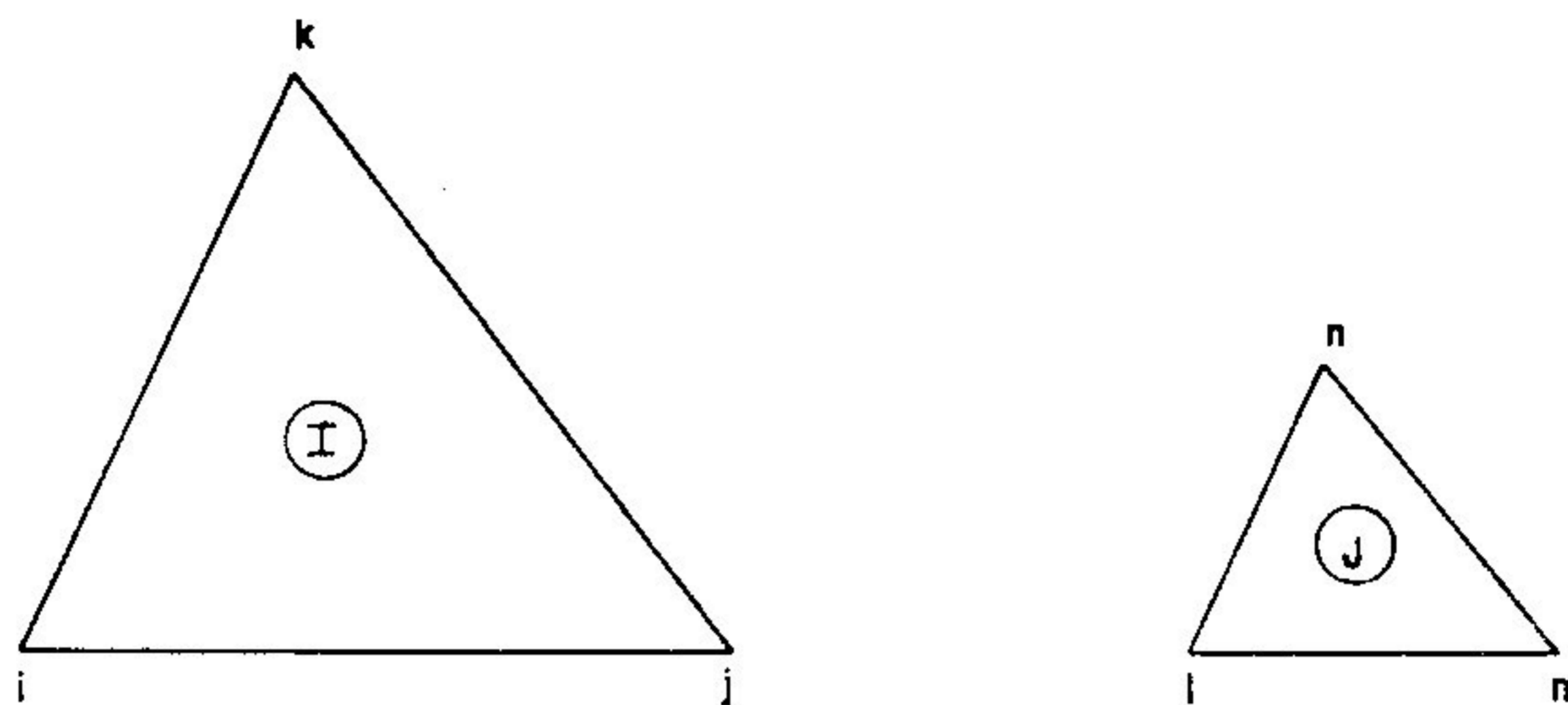
M0 : malha após fase r.

M1, M2, M3 : refinamentos globais H.

Todos os elementos são semelhantes entre si e ao elemento-pai, em todos os níveis.

Tal procedimento foi adotado a partir da constatação de que as matrizes de rigidez elementares de elementos semelhantes são idênticas (ver APÊNDICE 1), desde que suas incidências nodais (descrição dos nós de cada

elemento) mantenham correspondência entre si, FIG. II-6, e que o módulo de elasticidade e o coeficiente de Poisson sejam os mesmos.



ELEMENTO	INCIDENCIA NODAL		
I	i	j	k
J	l	m	n

FIG. II-6 Elementos semelhantes (I e J) com a incidência definida mantendo-se a correspondência entre os nós.

Esta propriedade tornou desnecessários o cálculo e o armazenamento de 84 matrizes de rigidez elementares para cada elemento de M0 (4 de M1, 16 de M2 e 64 de M3), o que representou grande economia de tempo de processamento e espaço de armazenamento dos dados.

A ordem de numeração dos novos nós, nos três novos níveis de malha, está indicada na FIG. II-7, que mostra um elemento genérico da malha M0, sub-dividido até o nível 3.

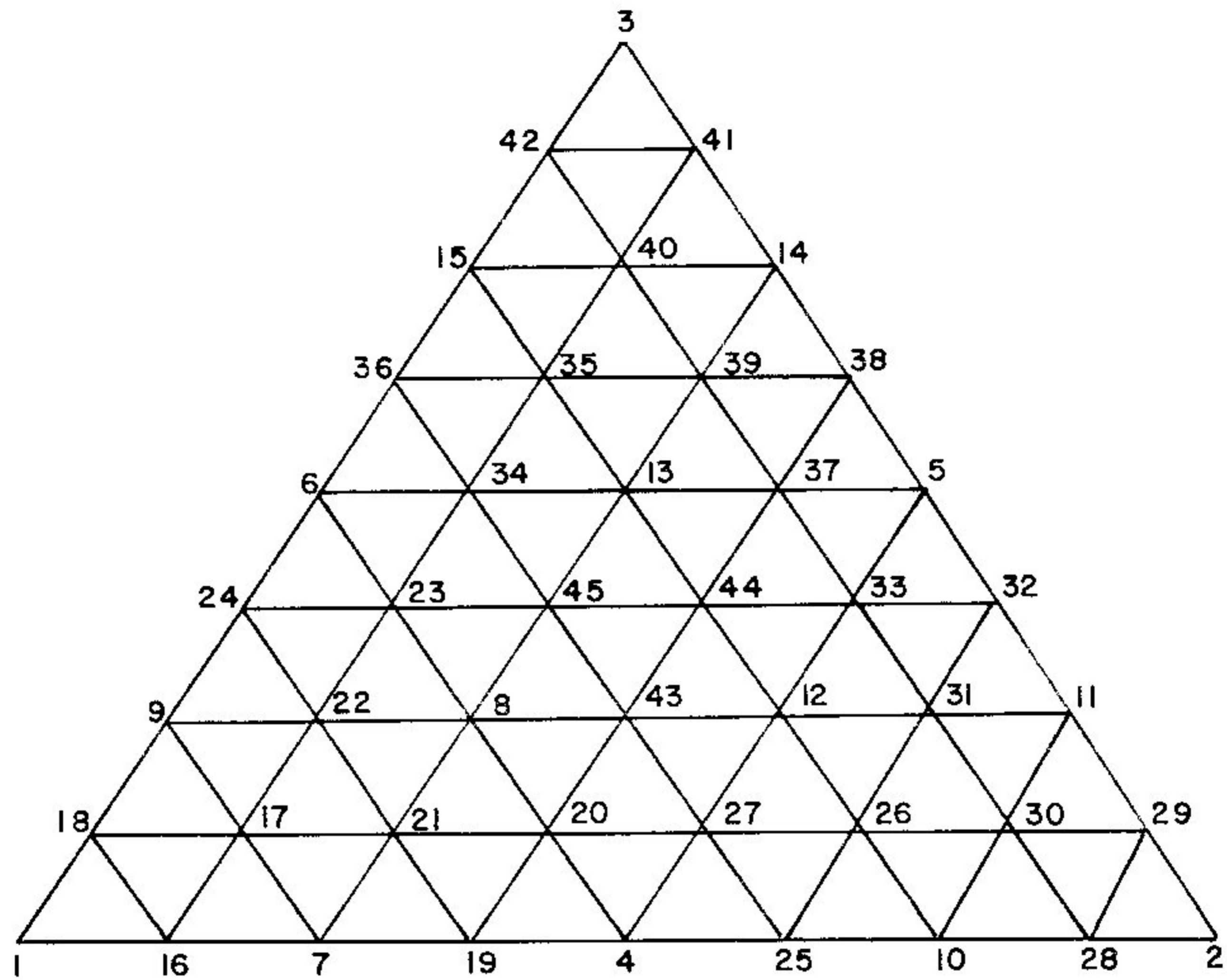


FIG. II-7 Ordem de numeração dos nós gerados a partir de um elemento genérico de M_0 , no processo de refinamento h em três níveis.

Os nós 1, 2 e 3 da Fig. II-7 correspondem à incidência nodal do elemento-pai genérico, nesta ordem. No nível 1 são criados os nós 4, 5 e 6 nesta ordem, isto é, o 4º nó no ponto médio do lado que une os dois primeiros nós (1 e 2), o 5º nó no que une o segundo e o terceiro nós (2 e 3) e o 6º nó no que une o terceiro e o primeiro nós (3 e 1).

Os quatro novos elementos malha M_1 são definidos pelas incidências

ELEMENTO	INCIDENCIA		
1	1	4	6
2	4	2	5
3	6	5	3
4	5	6	4

nesta ordem, onde a numeração dos elementos aqui representa a sequência de numeração.

A malha M2 é gerada pela criação dos nós 7 a 15, em cada elemento de M0, aplicando a mesma lei de formação da geração de M1 em cada elemento desta malha na ordem crescente de numeração. A definição dos 16 elementos de M2 é análoga à dos elementos de M1, considerando-se cada elemento de M1 da mesma forma que foram considerados os de M0.

O mesmo procedimento foi usado na geração da malha M3, com a criação dos nós 16 a 45 em cada elemento de M0, o que resulta em 64 novos elementos por elemento-pai.

A ordem de numeração dos elementos gerados a partir de cada elemento-pai, nos três níveis de malha, está mostrada na FIG. II-8.

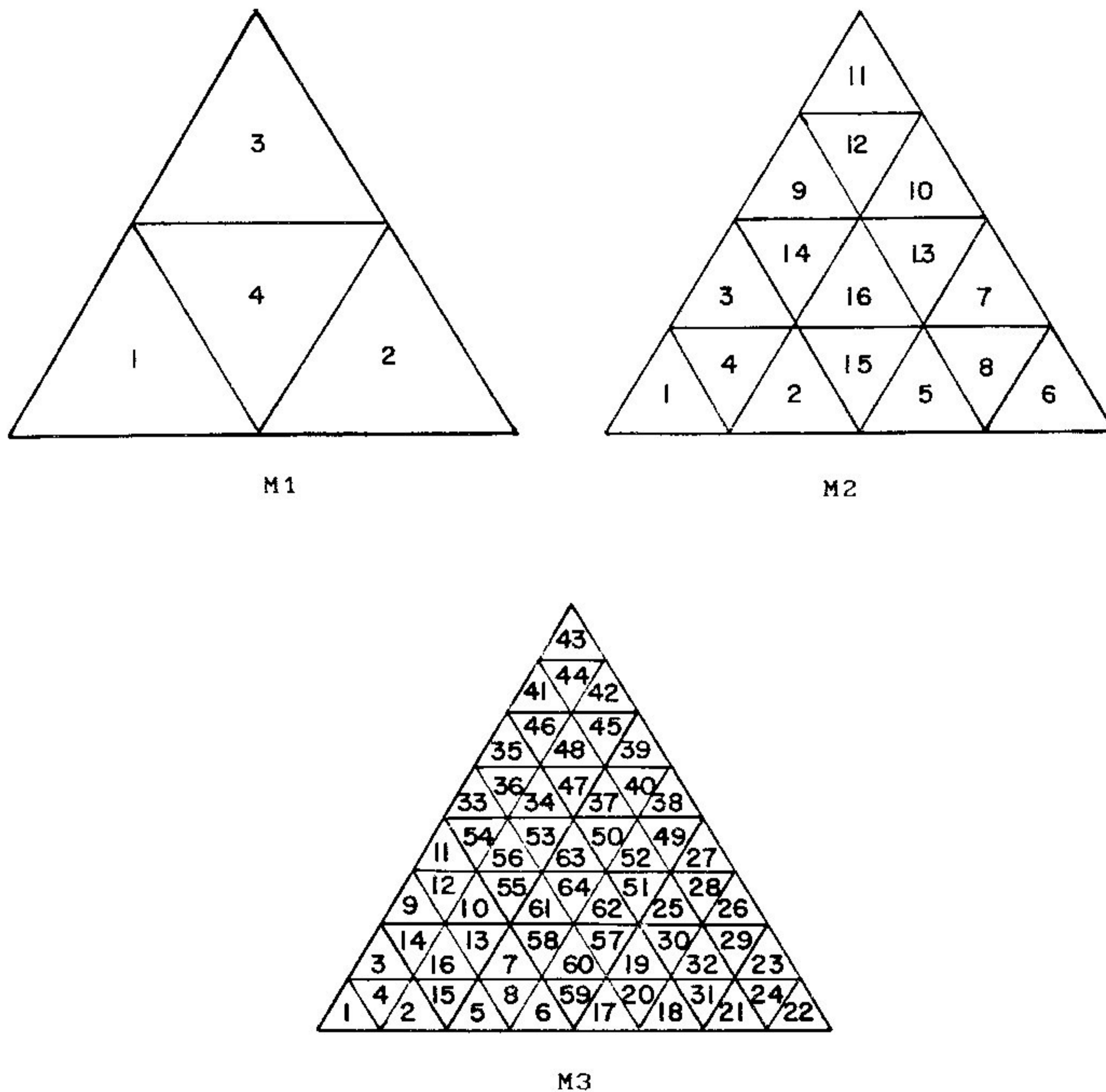


FIG. II-8 Sequência de numeração nos elementos de M1, M2 e M3.

A ordem de subdivisão dos elementos de uma malha segue a ordem de numeração destes elementos, por exemplo, na geração de M1 os elementos de M0 são subdivididos seguindo sua sequência de numeração, o mesmo sendo válido para a geração de M2 e M3 (FIG.II-7 e FIG. II-8).

É importante observar que na subdivisão de um elemento, ao se criar novos nós no seu contorno é necessário verificar se este elemento possui algum vizinho já subdividido. Neste caso, os novos nós que se situam no contorno já terão recebido uma numeração que deve ser mantida.

Pode-se notar que a numeração dos nós é contínua e fixa, sendo válida para todos níveis de malha, isto é, a partir do momento em que um nó recebeu um número, este será mantido em todos os níveis que contiverem o nó. Evidentemente, isto não é verdade para a numeração dos elementos, que é específica para cada um dos níveis (FIG.II-7 e FIG. II-8).

b) Coordenadas dos nós.

O cálculo das coordenadas x e y dos novos nós, em um determinado nível de malha, não apresenta dificuldades, uma vez que cada um deles se situa sempre no ponto médio do lado que une dois nós da malha imediatamente mais grossa, os quais possuem suas coordenadas conhecidas (FIG.II-9).

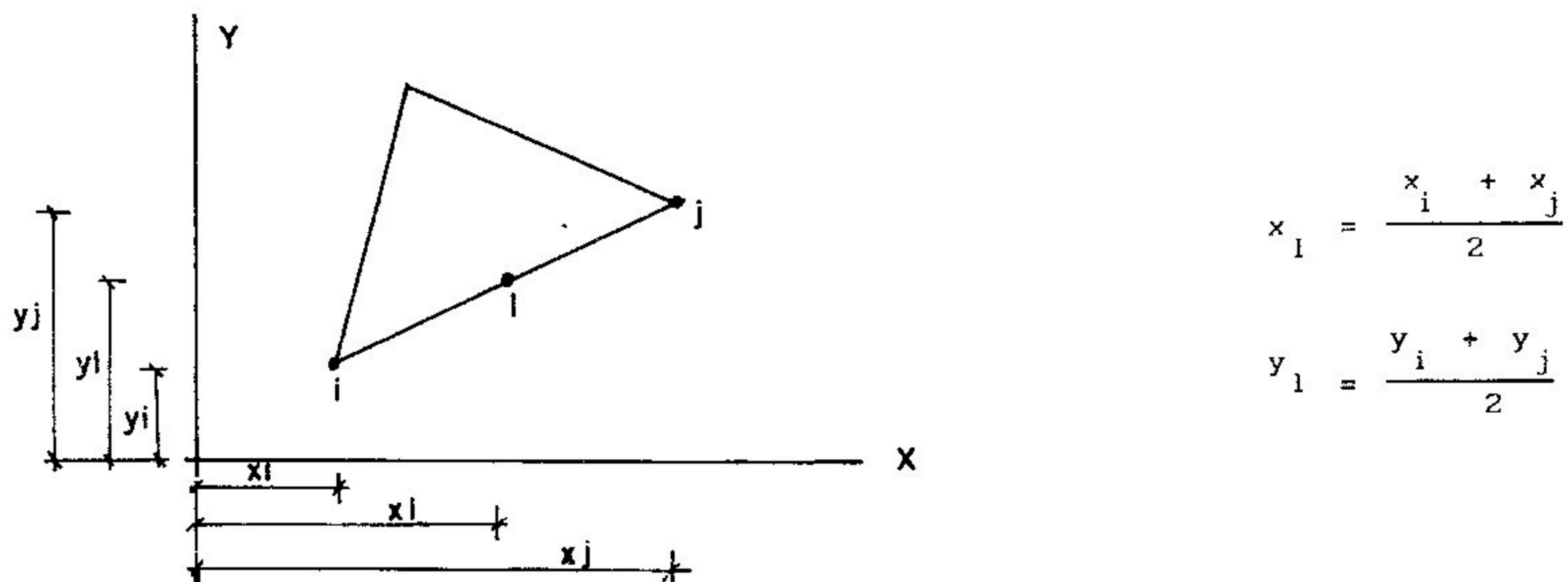
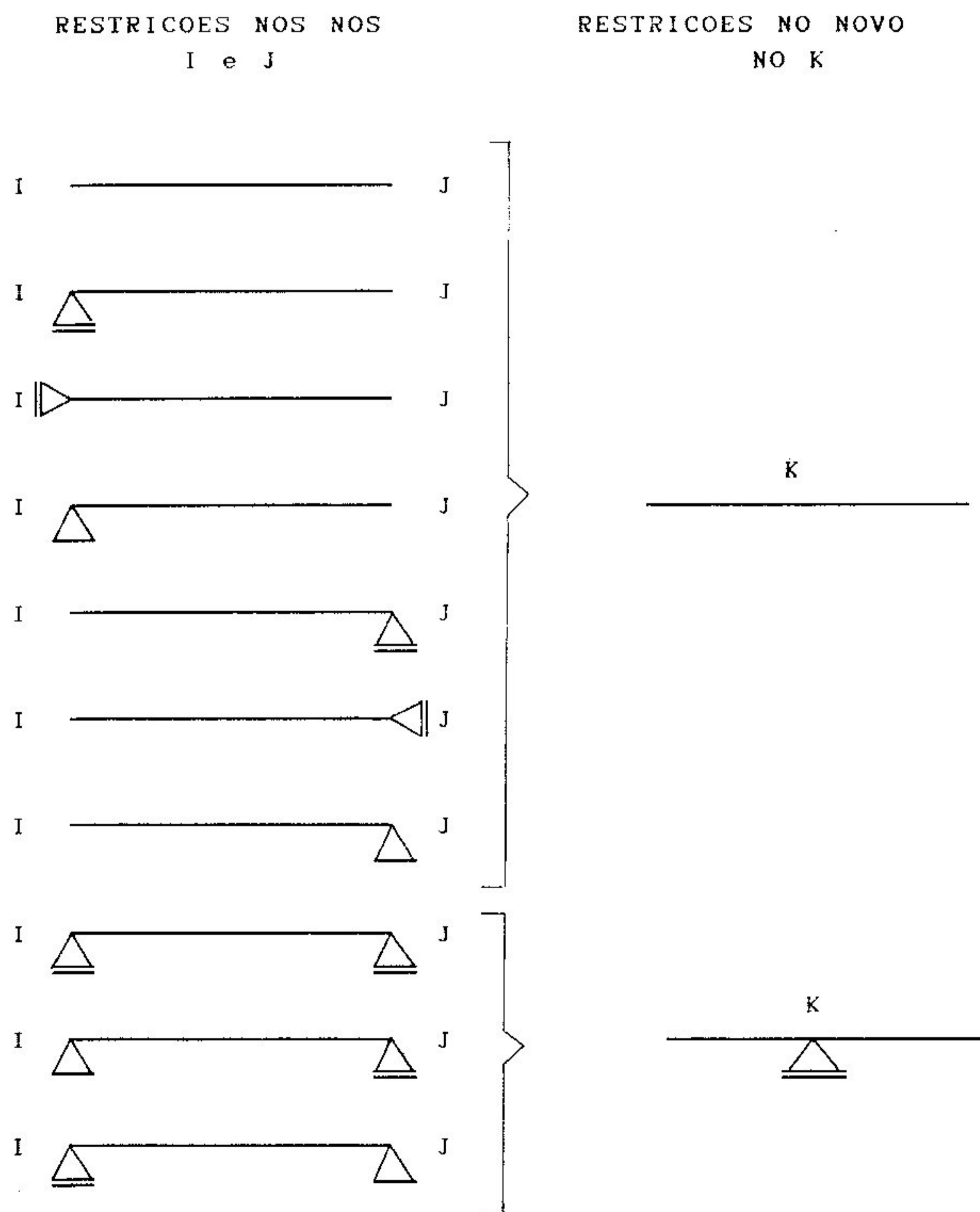


FIG. II-9 Coordenadas do novo nó 1.

c) Condições de contorno

As condições de contorno das novas malhas M1, M2 e M3 - restrições a deslocamentos nodais - foram definidas tendo como referência a malha imediatamente mais grossa de cada uma delas. Para um determinado novo nó de uma malha, é feita uma pesquisa em seus dois nós vizinhos, os quais vêm necessariamente da malha anterior, para saber-se se há alguma restrição ao deslocamentos destes nós. A definição das restrições do novo nó é feita da seguinte maneira: para cada direção x e y, o nó terá a característica do nó vizinho menos restrito. Podem ocorrer as situações mostradas na FIG.II-10, onde no lado IJ de um elemento é criado um novo nó K.



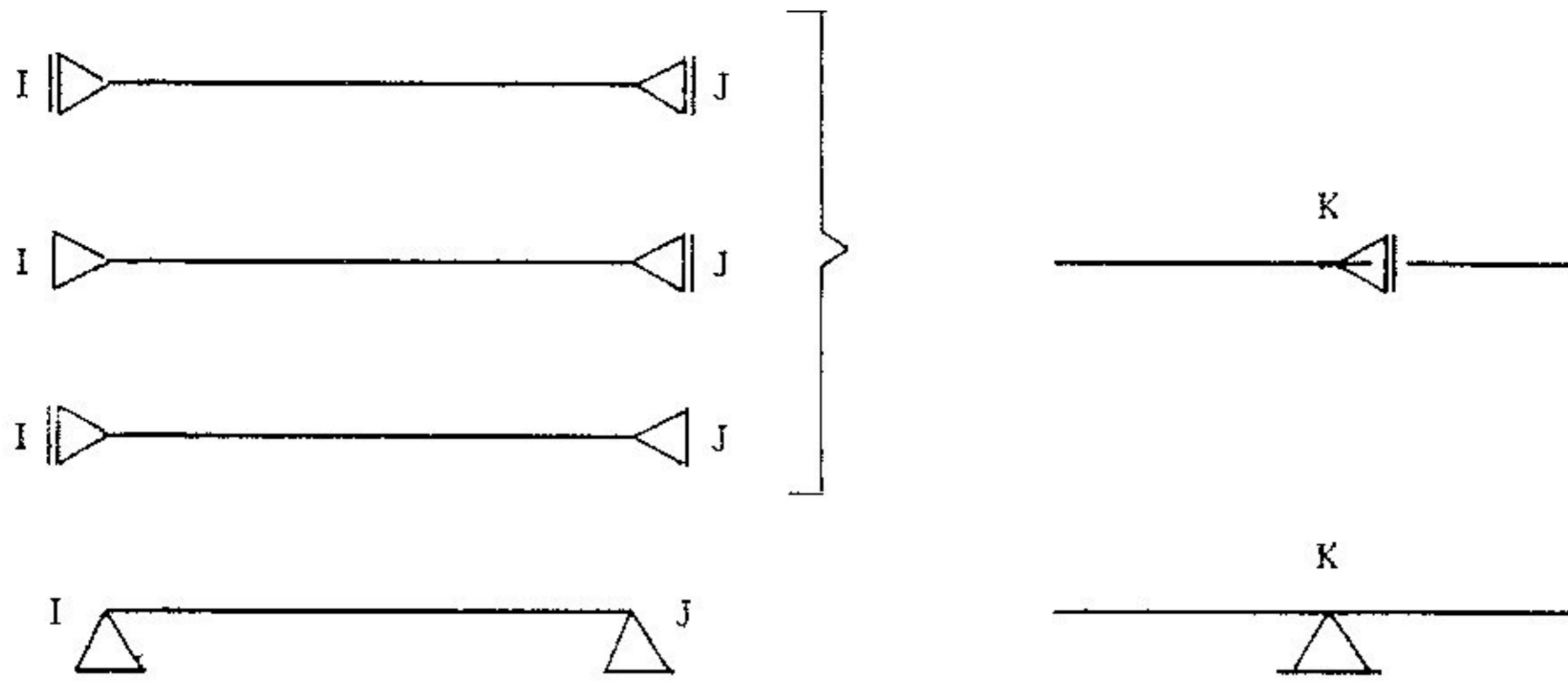


FIG. II-10 Definição das condições de contorno de um novo nó.

d) Carregamento.

As cargas concentradas nodais presentes em M0 permanecerão com os mesmos valores e aplicadas nos mesmos nós em que já se encontravam, quando da construção de M3.

As cargas distribuídas nos lados dos elementos da malha M0, são descritas pelos valores que assumem nos nós que limitam estes lados. Quando do refinamento da malha, para os nós criados no lado carregado de um elemento foi atribuído um valor de carga obtido pela interpolação linear dos valores nos nós do limite. Só há necessidade de montar o vetor de cargas para a malha M3, pois no MM os sistemas de M1 e M2 serão os correspondentes à equações residuais.

A FIG. II-11 ilustra o procedimento descrito.

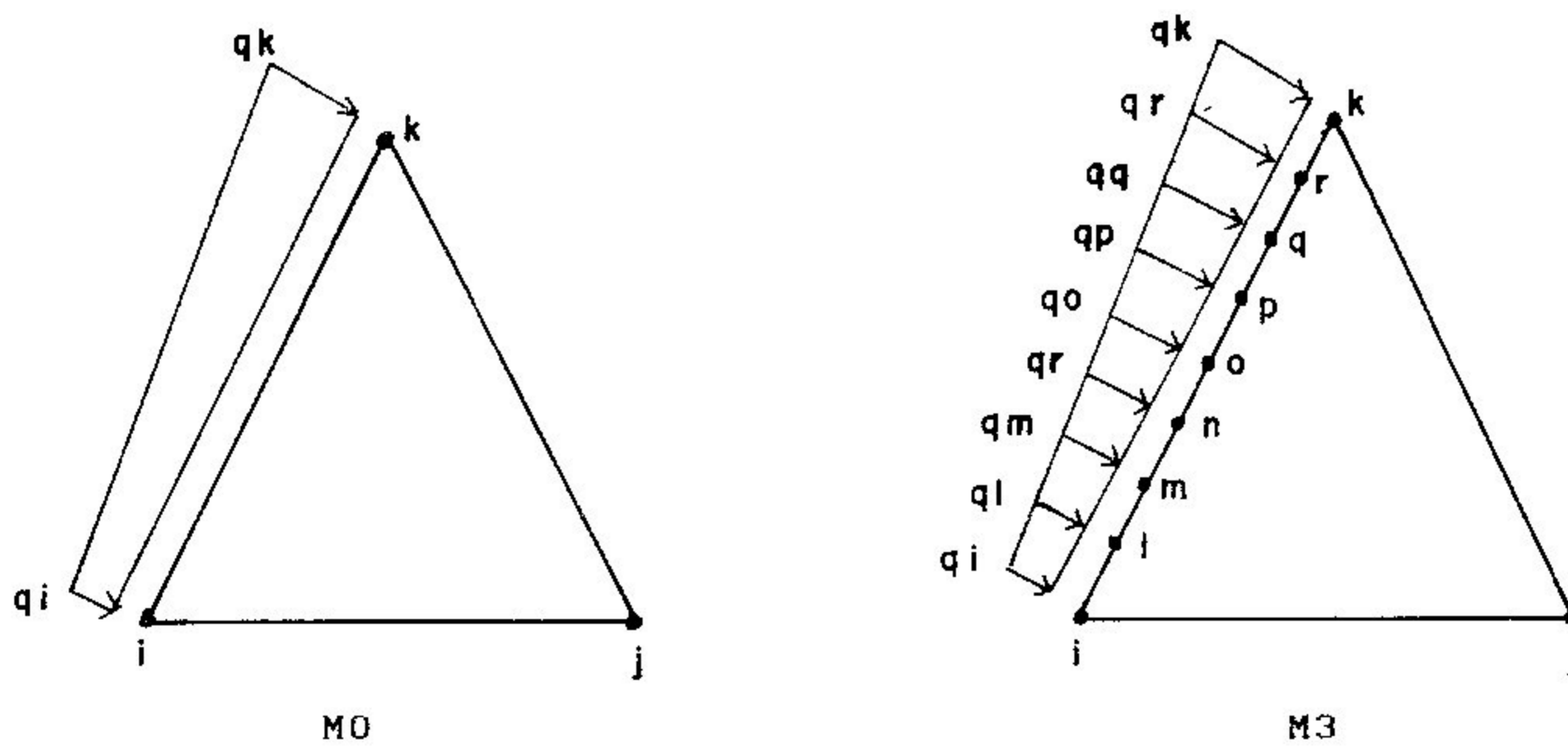


FIG. II-11 Definição do carregamento em M3.

II-2-2 Estrutura do banco de dados de M1, M2 e M3.

Como a geração dos elementos de todos os níveis é feita com referência aos elementos de M0, seguindo sempre a mesma lei de formação, os dados referentes à incidência nos três níveis foram armazenados de forma compacta em um vetor chamado INCH para cada elemento de M0. Um vetor INCH contém todos os nós gerados a partir de um elemento de M0 até M3 e haverá tantos vetores INCH quantos forem os elementos de M0.

As 45 posições do vetor indicam a sequência de geração dos nós no refinamento recursivo de um elemento. As três primeiras posições correspondem à incidência do elemento de M0, as três seguintes aos nós criados na geração de M1 seguindo a sequência descrita anteriormente, as nove seguintes aos nós criados na geração de M2 e as trinta últimas aos nós criados na geração de M3. Todos eles contidos no elemento-pai descrito pelas três primeiras posições.

Uma vez definido o vetor INCH de todos os elementos-pais, a incidência de cada elemento de cada malha é conseguida pela combinação de três dos 45 nós contidos no vetor. Como a lei de formação é sempre a mesma para todos os elementos-pais, a incidência, por exemplo, do 13^o elemento de M2 gerado a partir de qualquer um dos elementos de M0 será definida pelos nós que se encontram nas posições 5, 13 e 12, nesta ordem, do vetor INCH correspondente. Isto é válido para todos os elementos de M0.

Baseado neste fato, criou-se um vetor apontador de incidências - vetor IAPO - que fornece as incidências de todos os elementos de todos os níveis através da indicação das posições em que se encontram os nós no vetor INCH de cada elemento-pai. O vetor IAPO é único pois contém posições do vetor INCH e não números de nós e será utilizado tantas vezes quantos forem os elementos de M0. Possui 255 posições e define a incidência dos 85 elementos (1 de M0, 4 de M1, 16 de M2 e 64 de M3) que são gerados a partir do elemento de M0 definido pelas três primeiras posições (FIG. II-12).

1	2	3	1	4	6	4	2	5	6	5	3	5	6	4
1	7	9	7	4	8	9	8	6	8	9	7	4	10	12
10	2	11	12	11	5	11	12	10	6	13	15	13	5	14
15	14	3	14	15	13	6	13	12	13	6	8	12	8	4
8	12	13	1	16	18	16	7	17	18	17	9	17	18	16
7	19	21	19	4	20	21	20	8	20	21	19	9	22	24
22	8	23	24	23	6	23	24	22	8	22	21	22	9	17
21	17	7	17	21	22	4	25	27	25	10	26	27	26	12
26	27	25	10	28	30	28	2	29	30	29	11	29	30	28
12	31	33	31	11	32	33	32	5	32	33	31	11	31	30
31	12	26	30	26	10	26	30	31	6	34	36	34	13	35
36	35	15	35	36	34	13	37	39	37	5	38	39	38	14
38	39	37	15	40	42	40	14	41	42	41	3	41	42	40
14	40	39	40	15	35	39	35	13	35	39	40	5	37	33
37	13	44	33	44	12	44	33	37	13	34	45	34	6	23
45	23	8	23	45	34	12	43	27	43	8	20	27	20	4
20	27	43	8	43	45	43	12	44	45	44	13	44	45	43

Fig. II-12 Vetor IAPO , que aponta as posições, no vetor INCH, que contêm os números dos nós que constituem a incidência de todos os elementos gerados.

Os três primeiros valores do vetor IAPO indicam que a incidência de um elemento de M0 é composta pelos nós cujos números se encontram nas posições 1, 2 e 3, nesta ordem, do vetor INCH associado a ele. Os valores que se encontram nas três posições seguintes indicam que a incidência do primeiro elemento de M1 gerado pelo elemento de M0, ao qual se refere o vetor INCH em questão, é composta pelos nós que se encontram nas posições 1, 4 e 6, nesta ordem, deste vetor INCH, e assim sucessivamente.

Observa-se que os valores nas posições 4^a a 15^a do vetor IAPO, três a três consecutivamente, referem-se às incidências dos quatro elementos de M1, nesta ordem; os valores nas posições 16^a a 63^a, três a três consecutivamente, referem-se às incidências dos dezesseis elementos de M2, nesta ordem; os valores nas posições 64^a a 255^a referem-se, também três a três consecutivamente, às incidências dos sessenta e quatro elementos de M3, nesta ordem, todos gerados de um único elemento de M0.

É apresentado a seguir um exemplo do refinamento H resultante do procedimento descrito acima.

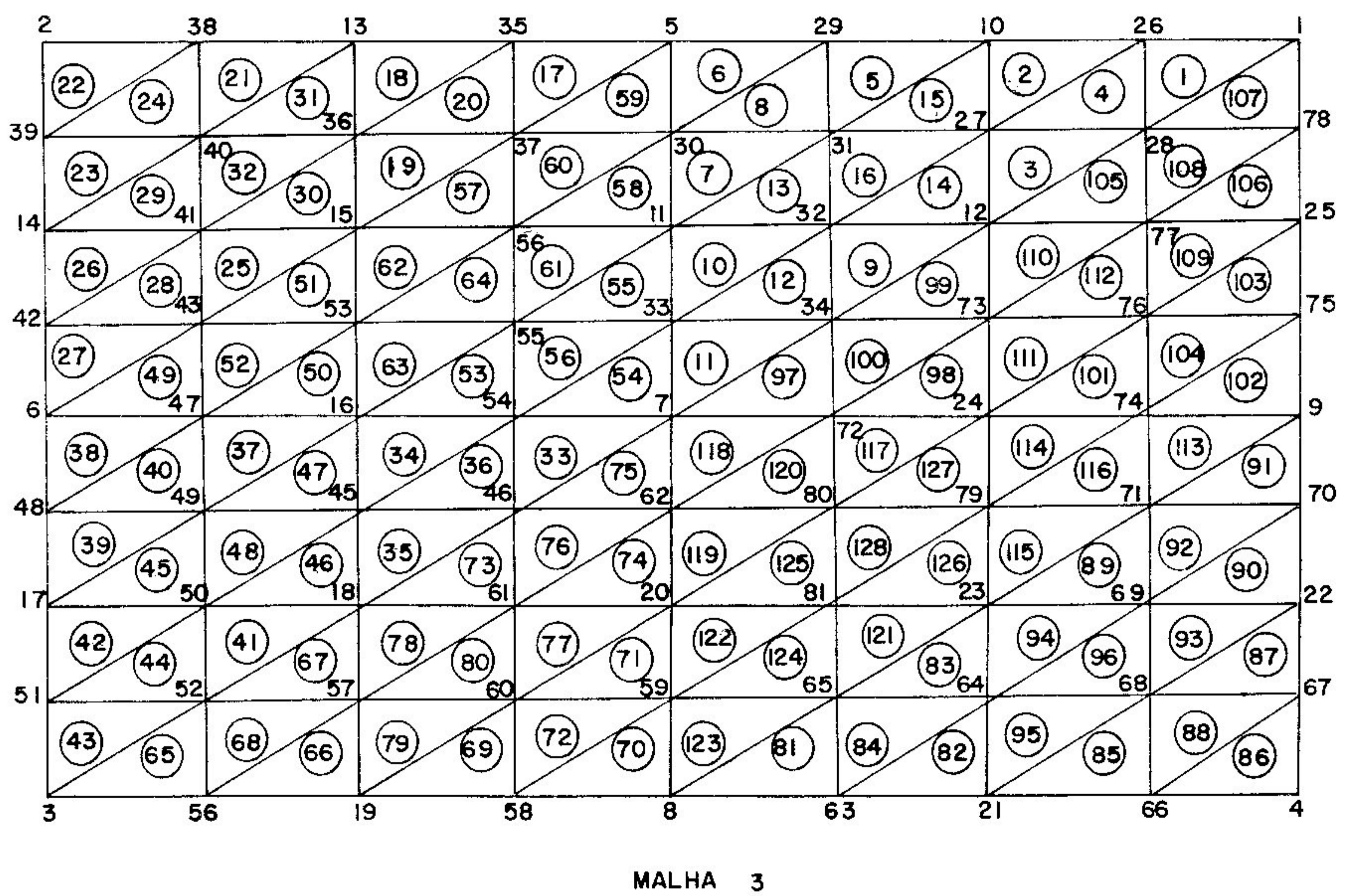
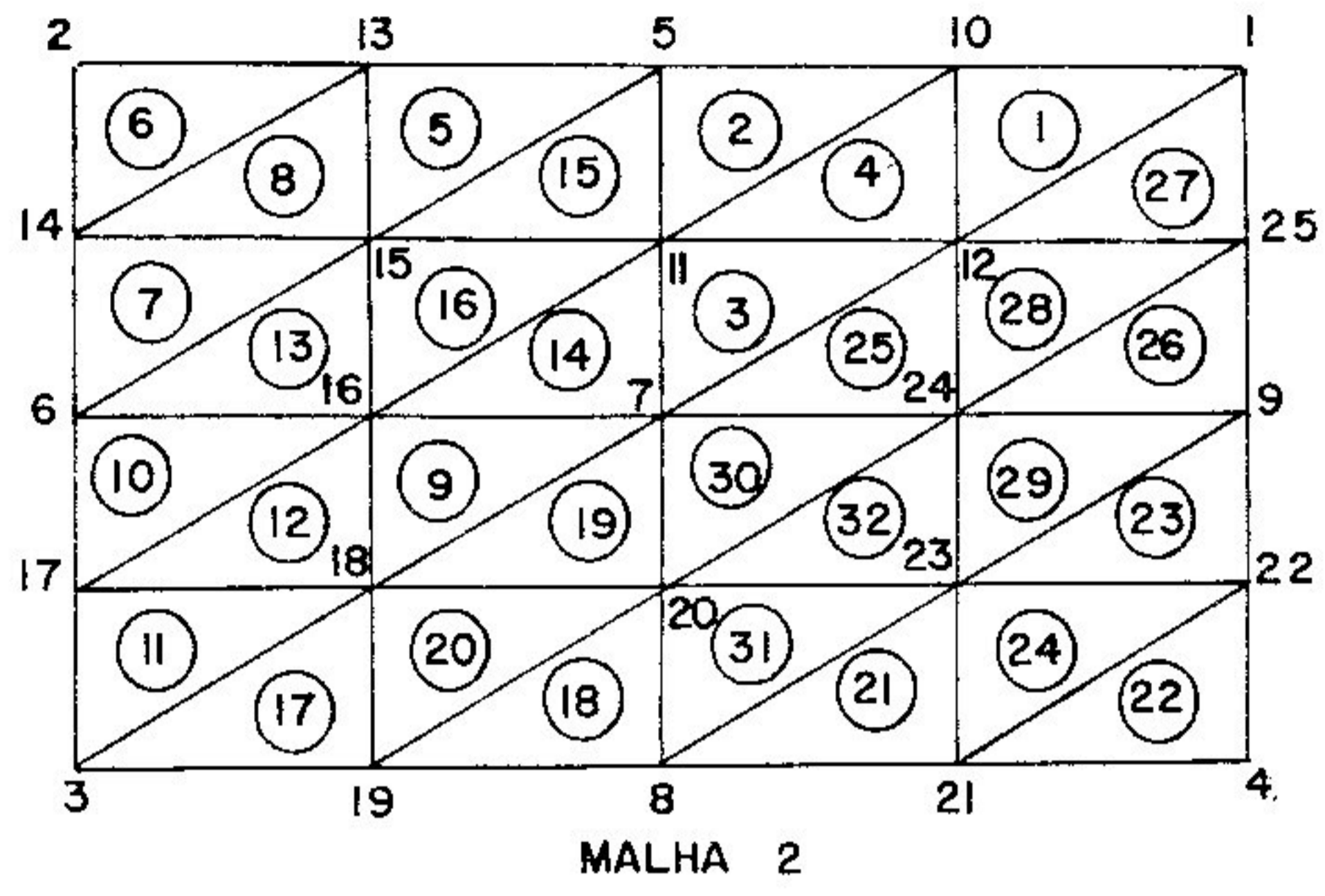
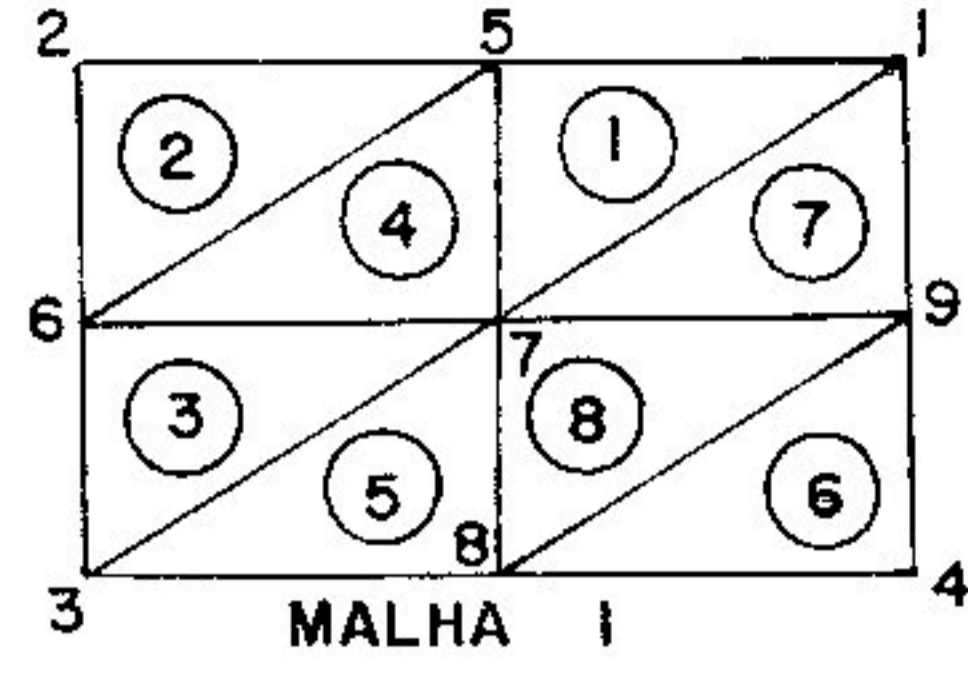
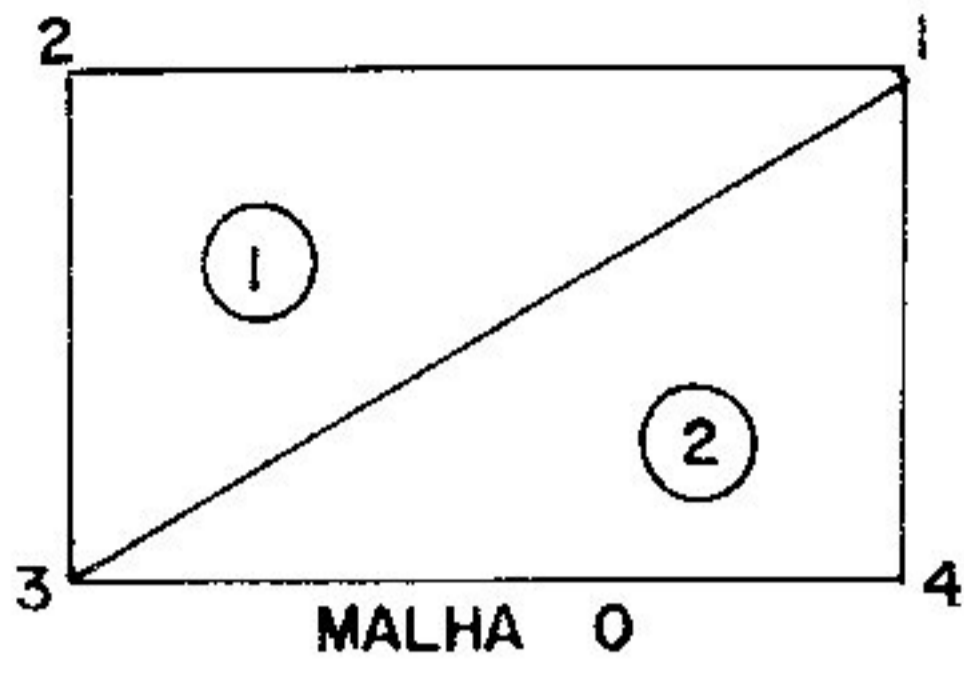


FIG. II-13 Exemplo de refinamento H.

Os vetores INCH associados aos dois elementos de M0 estão mostrados a seguir.

INCH do elemento 1 de M0

1	2	3	5	6	7	10	11	12	13	14	15	16	17	18
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55

INCH do elemento 2 de M0

3	4	1	8	9	7	19	20	18	21	22	23	24	25	12
56	57	52	58	59	60	61	62	46	63	64	65	66	67	68
69	70	71	72	73	34	74	75	76	77	78	28	79	80	81

Na verdade, os vetores INCH de todos os elementos de M0 compõem apenas um vetor INCH que armazena a série de colunas de uma matriz de dimensão $45 \times \text{NELG}$ onde NELG é o número de elementos de M0.

As coordenadas dos nós de M0, M1, M2 e M3, que estão todos presentes em M3, são armazenadas, da mesma forma que no programa original, em um vetor chamado COOH que contém as colunas de uma matriz de dimensão $2 \times \text{NN}$ onde NN é o número total de nós de M3. Portanto, as coordenadas x e y do nó 8, por exemplo, estarão armazenadas nas posições 15 e 16 de COOH.

As condições de contorno de todos os nós (M3), são armazenadas em dois vetores IDH1 e IDH2, da mesma forma que no programa original. O vetor IDH1 armazena por colunas uma matriz de dimensão $2 \times \text{NNO3}$ onde NNO3 é o número total de nós de M3. As linhas 1 e 2 armazenam as restrições nas direções x e y, respectivamente, de todos os NNO3 nós representados nas colunas. Neste vetor o número 1 representa uma restrição e o número 0 representa a ausência de restrição. O vetor IDH2 é semelhante ao primeiro, os números

0 são aqui substituídos por uma numeração crescente partindo de 1, seguindo as colunas. Os números 1 são aqui substituídos por 0. Este vetor, portanto, numera os graus de liberdade e os associa aos nós. A FIG. II-14 mostra um exemplo das matrizes armazenadas em IDH1 e IDH2.

1	0	0	1	0	0	0	1
1	0	0	0	0	0	1	1

(a)

0	1	3	0	6	8	10	0
0	2	4	5	7	9	0	0

(b)

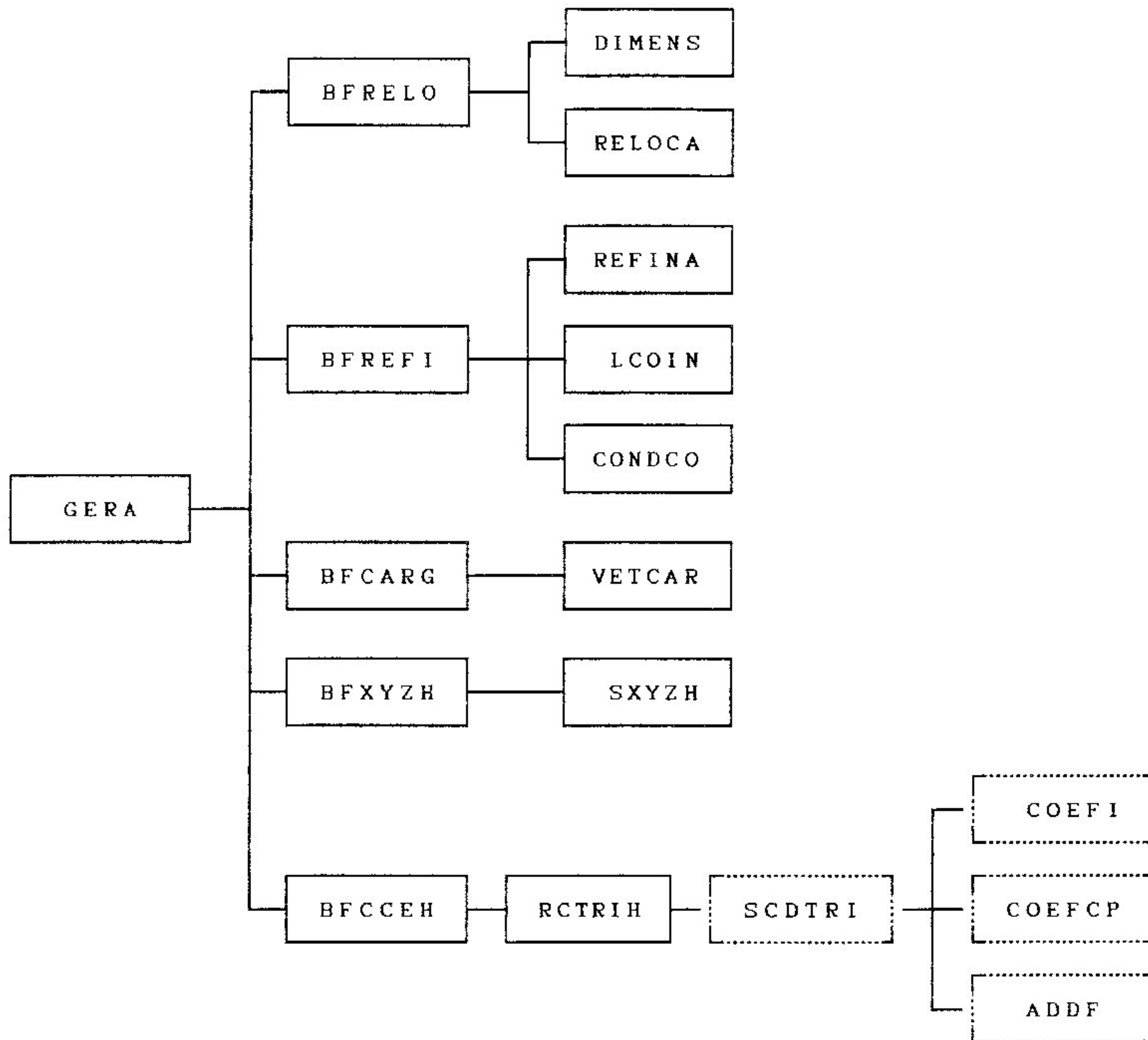
FIG. II-14 (a) Exemplo de matriz armazenada em IDH1.
 (b) Matriz associada armazenada em IDH2.

O carregamento da malha M3 também é armazenado da mesma forma que no programa original, através de um vetor que contém as colunas de uma matriz de dimensão $4 \times \text{NECD}$, onde NECD é o número de elementos com carga distribuída. As quatro linhas de cada elemento referem-se aos valores que a carga distribuída assume nas direções x e y do 1º e do 2º nós do lado carregado. Observa-se que, como no programa original, cada elemento pode ser carregado apenas em um dos lados.

II-2-3 Estrutura do processador GERA.

O diagrama de estrutura do processador GERA, criado para executar o refinamento H global a partir da malha modificada pelo processador ADA (fase R) é mostrado na FIG. II-15.

PROCESSADOR BLOCOS FUNCIONAIS SUB-ROTINAS DE CÁLCULO



Legenda:
 rotina ja existente
 — nova rotina

FIG. II-15 Diagrama de estrutura do processador GERA

A função de cada rotina da FIG. II-15 é descrita sumariamente a seguir.

- * GERA: processador que executa o refinamento H global, gerando três malhas sucessivamente mais refinadas, a partir dos bancos de dados gerados pelos processadores PREAD e ADA;
- .BFRELO: faz a transferência de valores armazenados em tabelas geradas em PREAD para algumas posições de tabelas do GERA;
- DIMENS: calcula a dimensão da tabela COOH;
- RELOCA: transfere valores das tabelas COOR e INCI, de PREAD para as tabelas COOH e INCH, respectivamente;
- .BFREFI: executa o refinamento de M0 em três etapas, criando as malhas M1, M2 e M3 sucessivamente mais refinadas; define as condições de contorno de M1, M2 e M3;
- REFINA: monta os vetores de incidência (INCH), apontador de incidências (IAPO) e de coordenadas nodais (COOH);
- LCOIN: imprime, em forma de matriz, as tabelas COOH e INCH;
- CONDCO: monta as tabelas de condições de contorno (IDH1) e de graus de liberdade (IDH2);
- .BFCARG: monta os vetores referentes ao carregamento para a malha M3;
- VETCAR: monta vetores de parâmetros de carregamentos elementares (NPA3), de lista de elementos carregados (LED3) e de valores do carregamento nos nós do lado carregado dos elementos (VCD3);
- .BFXYZH: monta os vetores de coordenadas nodais por elemento, para as malhas M1, M2 e M3;
- SXYZH: constroi as tabelas XYZ1, XYZ2 e XYZ3 que contêm, para cada nível de malha, as coordenadas nodais por elemento;

.BFCCEH: monta o vetor de cargas nodais equivalentes às cargas distribuídas; observa-se que é a mesma presente no programa original, mudando-se apenas as tabelas utilizadas (tabelas de M3);

RCTRIH: gerencia chamada das sub-rotinas de cálculo; vale a observação feita em BFCCEH;

SCDTRI: rotina já existente no programa original, que calcula as cargas nodais equivalentes nos pontos de integração, para o carregamento distribuído;

COEFI: rotina já existente no programa original, que fornece os pontos de integração e os pesos correspondentes;

COEFCP: rotina já existente no programa original, que calcula o vetor das funções de interpolação e o jacobiano;

ADDF: rotina já existente no programa original, que transfere o carregamento de um elemento para o vetor de cargas equivalentes global.

CAPÍTULO III

MÉTODO MULTIMALHA IMPLEMENTAÇÃO

A implementação da solução do sistema de equações da malha M3 pelo Método Multimalha (MM), que é correspondente à etapa executada pelo processador SOL do programa original, foi dividida em duas fases: fase de preparação dos dados dos sistemas (montagem das matrizes de rigidez globais de M1, M2 e M3 e montagem do vetor de cargas nodais equivalentes de M3) e fase de solução do sistema de M3 por MM. Estas duas fases foram implementadas em dois processadores consecutivos: PREMG e MULGR, respectivamente. Esta divisão foi motivada pelo fato de que, além da solução por MM, foram também implementadas as soluções do sistema de M3 usando processo direto de eliminação de Gauss (processador ELIGA) e processo iterativo de Gauss-Seidel (processador GAUSDL), para efeito de avaliação do MM comparativamente a estes dois outros processos. O banco de dados gerado pelo processador PREMG pode ser usado em qualquer um dos três processadores de solução do sistema (MULGR, ELIGA e GAUSDL) fato que, além de ter simplificado a formulação de ELIGA e GAUSDL, permitiu que fosse feita uma avaliação do desempenho dos três algoritmos de solução independentemente da montagem do banco de dados. Outras duas considerações importantes na definição da forma de implementação foram o equipamento utilizado e a estrutura básica do SDP, mantendo-se, para isto, o tipo de armazenamento da matriz de rigidez global. O programa foi implantado em micro-computador tipo AT, usando FORTRAN 77 como linguagem de programação. Daí a necessidade de se evitar módulos muito grandes para cada processador. Os dois objetivos descritos acima são às vezes conflitantes, pois uma otimização da forma de armazenamento do sistema poderia se dar às custas de modificações adicionais na

estrutura do ambiente computacional. Optou-se por manter a forma de armazenamento da matriz global como no sistema original.

A descrição dos processadores ELIGA e GAUSDL encontram-se nos APÊNDICES 2 e 3 respectivamente.

III-1 Preparação de Dados para Solução do Sistema de M3 por MM. O Processador PREMG.

O processador PREMG tem uma estrutura semelhante à da parte inicial do processador SOL, na qual é definida a topologia e montada a matriz de rigidez global, é construído o vetor de cargas nodais equivalentes e é introduzido o efeito das prescrições em ambos.

Este processador utiliza os bancos de dados gerados nos processadores PREAD, PRO, SOL e GERA e é dividido em quatro etapas: cálculo das topologias das matrizes de rigidez globais de M1, M2 e M3 ($\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$ e $\tilde{K}^{(3)}$), montagem do vetor de cargas nodais equivalentes de M3 ($\tilde{f}^{(3)}$) e introdução das prescrições em $\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$, $\tilde{K}^{(3)}$ e $\tilde{f}^{(3)}$.

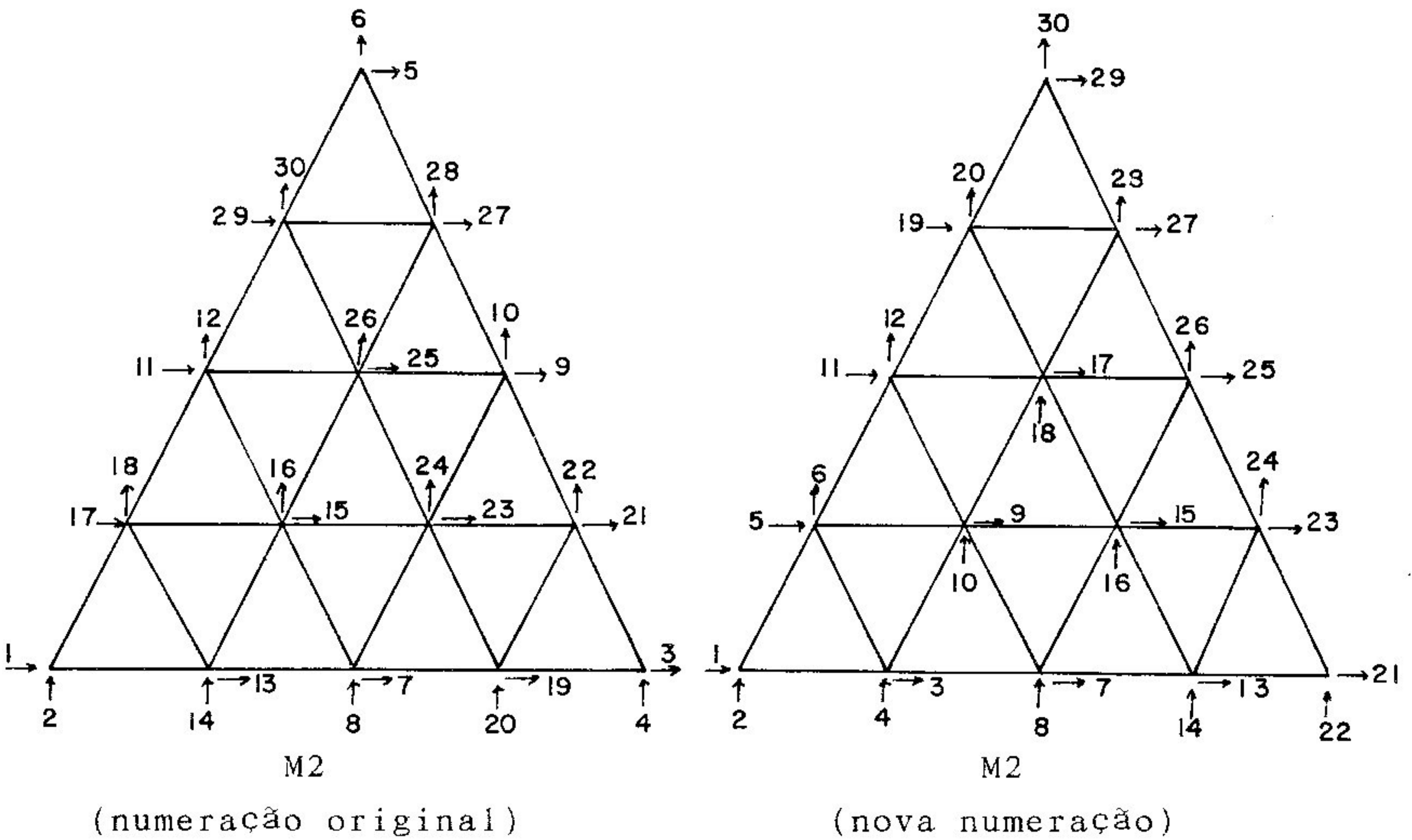
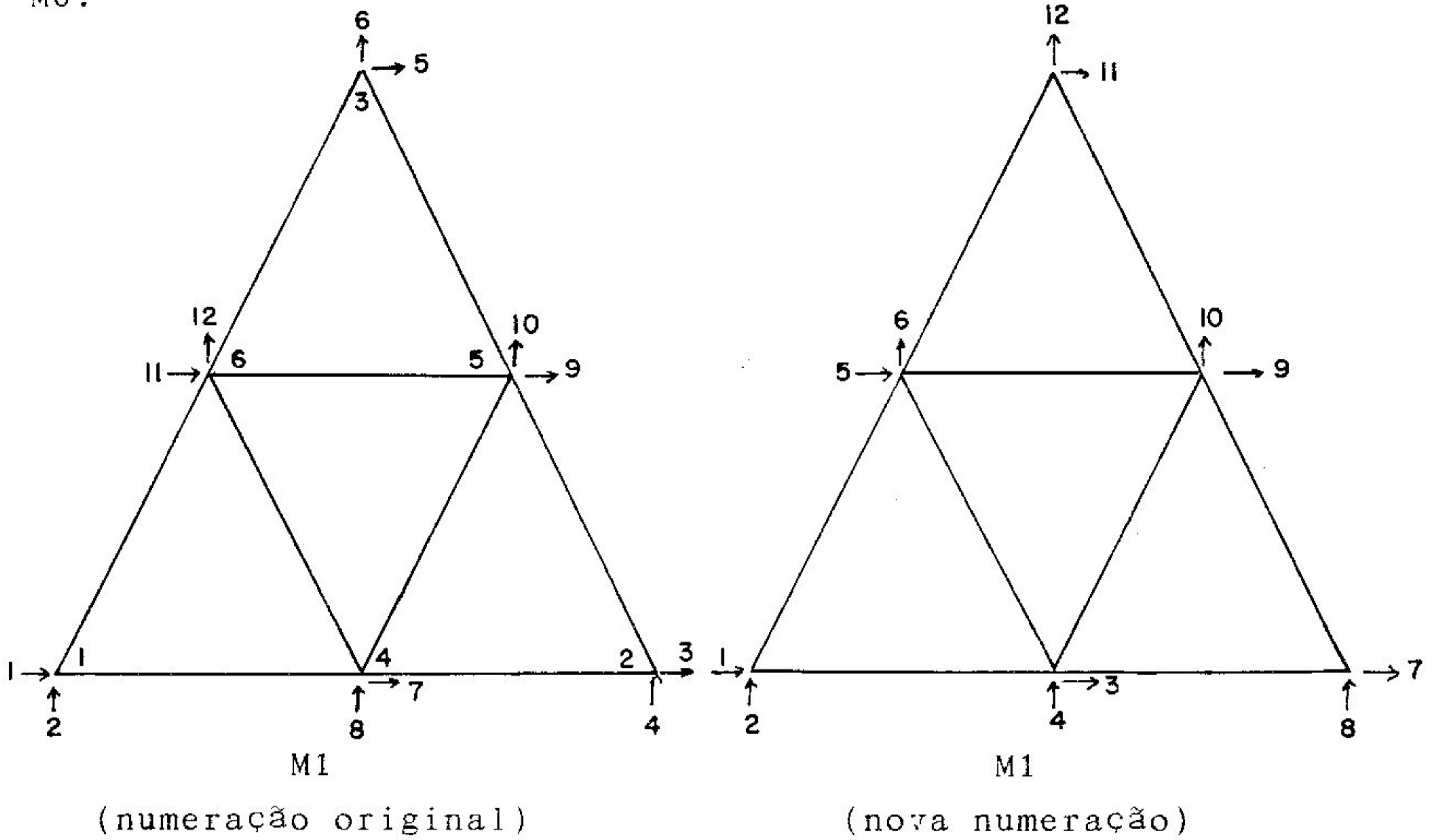
A determinação das topologias de $\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$ e $\tilde{K}^{(3)}$ é feita no bloco funcional BFTOPS, que inicialmente gerencia a renumeração dos graus de liberdade (GL) das malhas M1, M2 e M3 para que as matrizes de rigidez sejam menos esparsas, o que diminui o espaço de armazenamento necessário.

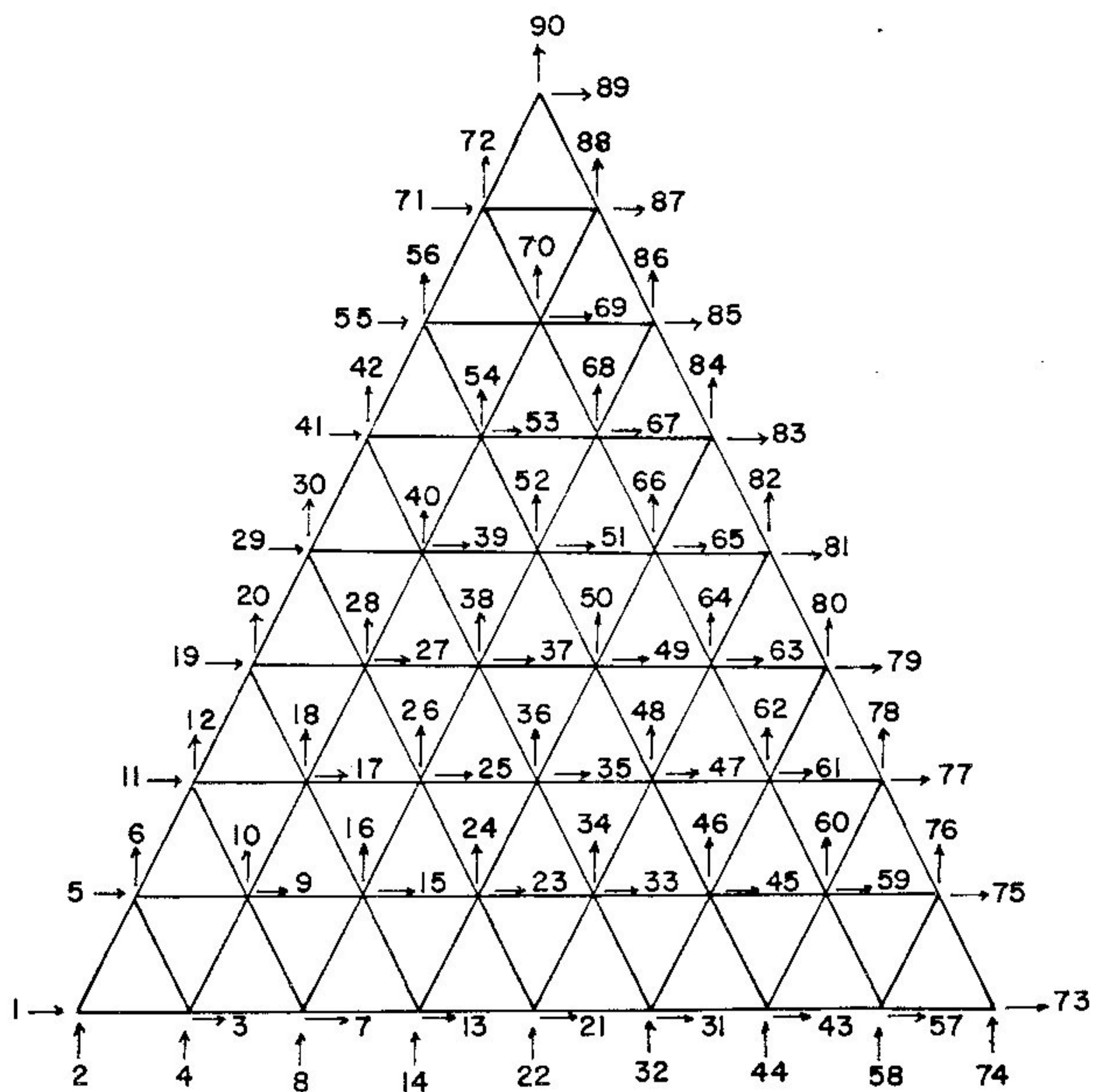
Foi feita a opção de renumerar os graus de liberdade e não os nós porque dessa forma foi necessário refazer apenas o vetor de graus de liberdade IDH2, criando os novos vetores ID21, ID22 e ID23 que fornecem os graus de liberdade associados aos nós das malhas M1, M2 e M3, respectivamente.

O procedimento adotado, bastante simples, consiste em renumerar, em cada elemento pai, os graus de liberdade de forma a diminuir o perfil da matriz de rigidez proveniente de cada um. Desta forma, a numeração dos

elementos é um parâmetro importante na definição dos requisitos de espaço de armazenamento.

A renumeração dos GL dos nós gerados por um elemento de M0 é feita na sequência mostrada na FIG. III-1 e é observada também a sequência de numeração dos elementos de M0.



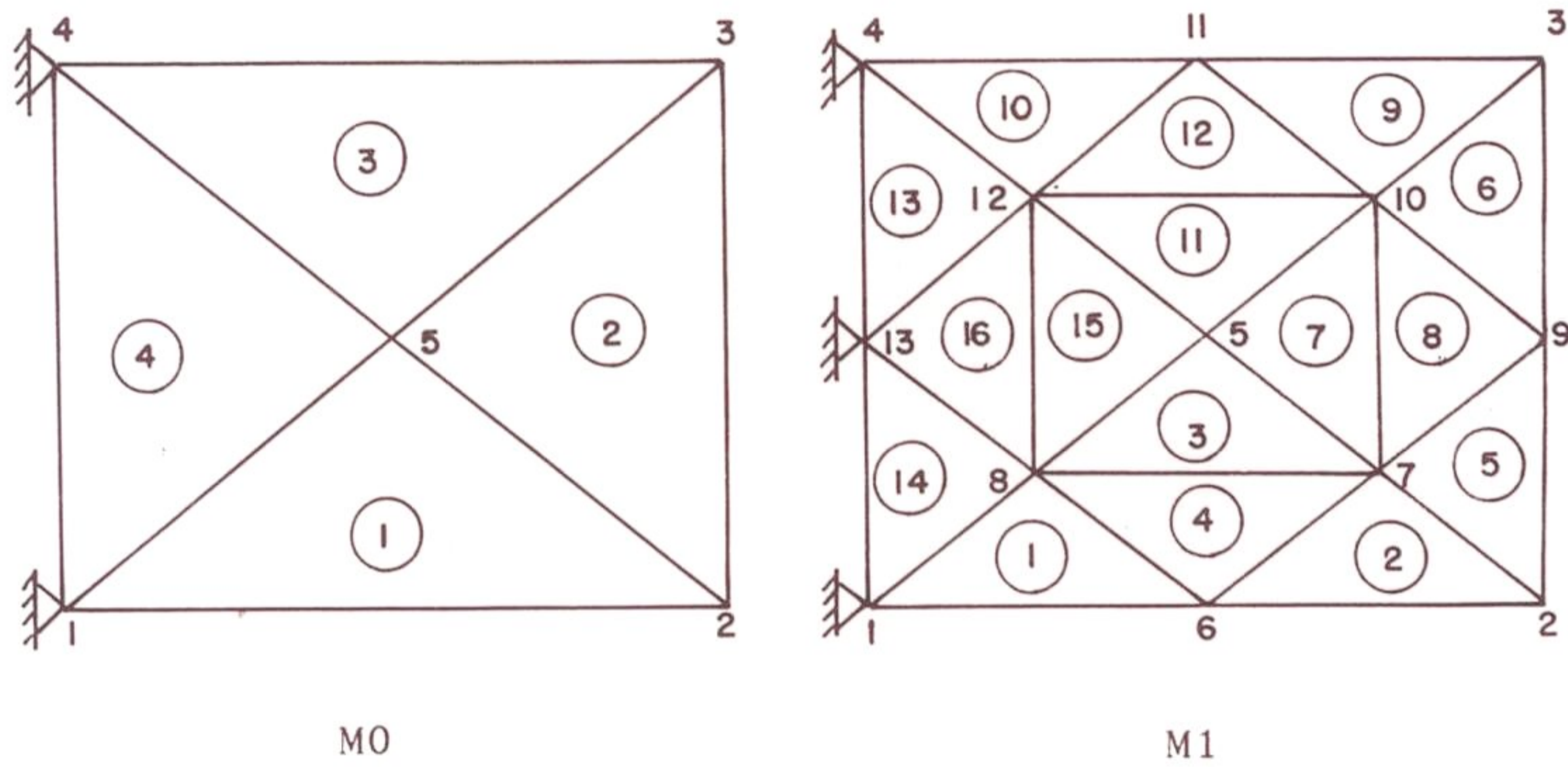


M3

(nova numeração)

FIG III-1 Sequência de renumeração dos GL dos elementos de M1, M2 e M3 gerados de um elemento de M0.

Na FIG. III-2 é mostrada a renumeração dos graus de liberdade da malha M1 de um exemplo simples.



	1	2	3	4	5	6	7	8	9	10	11	12	13
IDH2	0	1	3	0	5	7	9	11	13	15	17	19	0
	0	2	4	0	6	8	10	12	14	16	18	20	0

	1	2	3	4	5	6	7	8	9	10	11	12	13
ID21	0	5	13	0	9	1	7	3	11	15	17	19	0
	0	6	14	0	10	2	8	4	12	16	18	20	0

obs.: a numeração dos nós foi colocada acima das colunas para facilitar o entendimento.

FIG. III-2 Parte do vetor IDH2 (graus de liberdade) referente a M1 de um exemplo simples. Vetor ID21 (graus de liberdade renumerados) da mesma malha M1.

Após a renumeração dos GL de M1, M2 e M3, são montados os outros vetores referentes às topologias das matrizes de rigidez globais, para o que são utilizadas as mesmas rotinas de cálculo do programa original, três vezes consecutivas conforme diagrama da FIG. III-3. Uma nova rotina de cálculo (COLUNA) foi incluída para calcular as alturas das colunas de $\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$ e $\tilde{K}^{(3)}$ para posterior utilização na solução do sistema nos processadores MULGR e GAUSDL.

A montagem das matrizes de rigidez globais das malhas M1, M2 e M3 é feita em um bloco funcional chamado BFMONS, que tem a estrutura e utiliza as mesmas rotinas de cálculo do programa original, três vezes consecutivas (FIG. III-3). As matrizes $\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$ e $\tilde{K}^{(3)}$ são armazenadas em vetores através da técnica de "skyline". Foi introduzida aqui a rotina LINHA que calcula os comprimentos das linhas de $\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$ e $\tilde{K}^{(3)}$, os quais serão utilizados na solução do sistema de equações.

O vetor de cargas nodais equivalentes de M3 é montado no bloco funcional BFMCNS que difere do original apenas no banco de dados utilizado, sendo aqui o da malha M3.

A introdução de prescrições nas matrizes de rigidez e no vetor de cargas nodais equivalentes é feita no bloco funcional BFMVPS que difere do original no banco de dados utilizado. As rotinas de cálculo são as mesmas, sendo que a que modifica \tilde{K} é chamada três vezes consecutivas.

A FIG. III-3 mostra o diagrama de estrutura do processador PREMG.

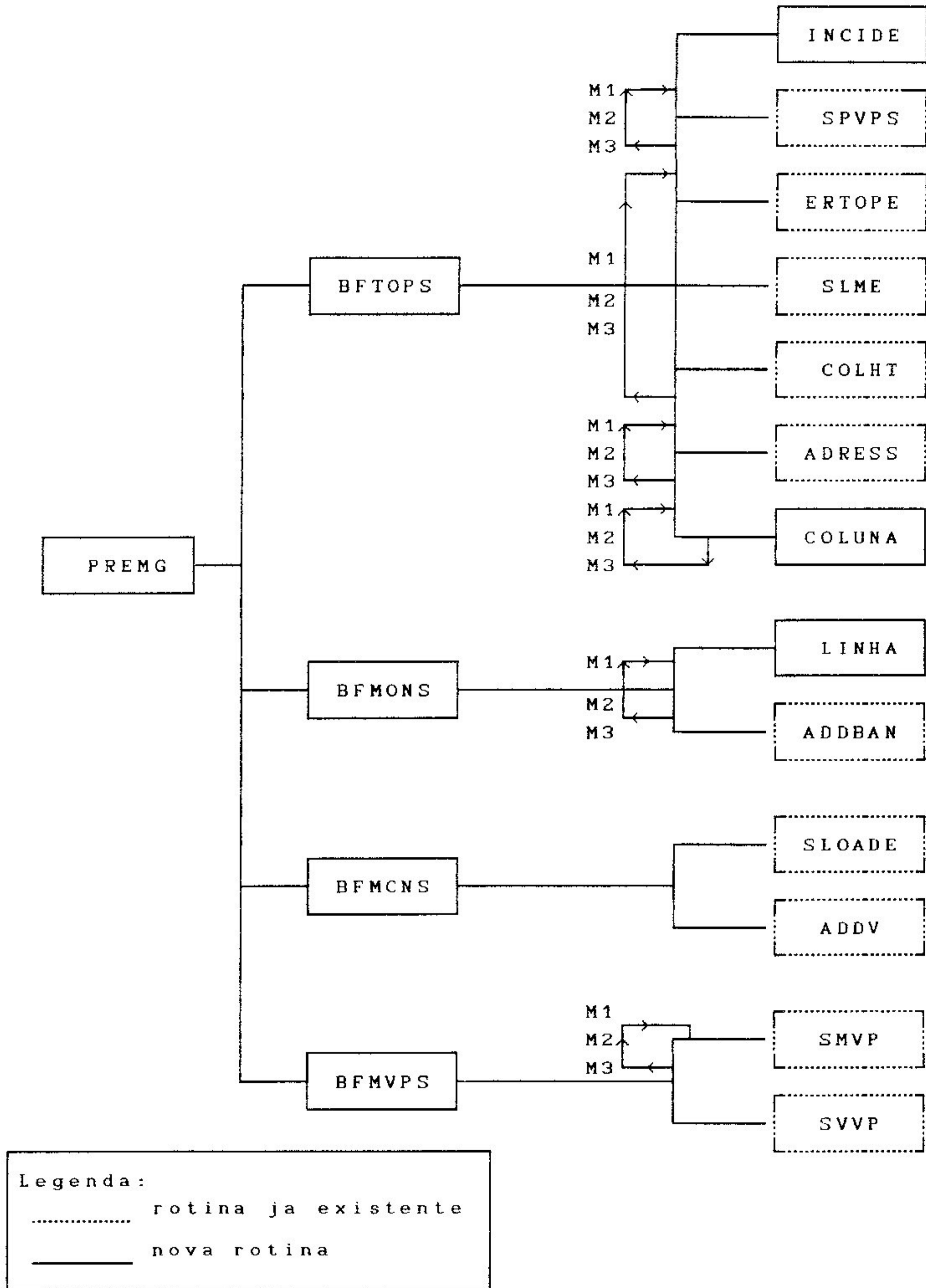


FIG. III-3 Diagrama de estrutura do processador PREMG.

A função de cada rotina da FIG. III-3 é descrita sumariamente a seguir.

- * PREMG: processador que prepara o banco de dados para a solução do sistema de equações de M3;
- .BFTOPS: gerencia o cálculo da topologia das matrizes de rigidez globais $\underline{K}^{(1)}$, $\underline{K}^{(2)}$ e $\underline{K}^{(3)}$;
- INCIDE: monta as tabelas de incidência das três malhas (INC1, INC2 e INC3), de uso restrito neste bloco funcional, e as tabelas de GL renumerados ID21, ID22 e ID23;
- SPVPS: rotina já existente de preparação de dados;
- ERTOPE: rotina já existente para verificação de consistência dos GL com o tipo do elemento;
- SLME: rotina já existente que associa cada elemento aos GL de seus nós;
- COLHT: rotina já existente que calcula as alturas das colunas de \underline{K} ;
- ADRESS: rotina já existente que monta os vetores MXA1, MXA2 e MXA3 que fornecem as posições iniciais das colunas de \underline{K} (apontador para \underline{K} armazenada em vetor por "skyline");
- COLUNA: monta vetor com altura das colunas de \underline{K} ;
- .BFMON5: gerencia a montagem das matrizes de rigidez globais $\underline{K}^{(1)}$, $\underline{K}^{(2)}$ e $\underline{K}^{(3)}$ em vetores utilizando a técnica de "skyline";
- LINHA: monta vetor com comprimento das linhas de \underline{K} ;
- ADDBAN: rotina já existente que transfere valores de uma matriz de rigidez elementar para a global;
- .BFMCNS: gerencia a montagem do vetor de cargas nodais equivalentes de M3;
- SLOADE: rotina já existente que monta vetor de cargas nodais equivalentes (concentradas e distribuídas) de um elemento;
- ADDV: rotina já existente que monta cargas nodais equivalentes do elemento no vetor de cargas nodais equivalentes global;

- .BFMVPS: gerencia a introdução de prescrições em $\underline{K}^{(1)}$, $\underline{K}^{(2)}$, $\underline{K}^{(3)}$ e no vetor de cargas nodais equivalentes de M3 ($\underline{f}^{(3)}$);
- SMVP: rotina já existente que modifica uma matriz \underline{K} de acordo com os valores prescritos;
- SVVP: rotina já existente que modifica o vetor $\underline{f}^{(3)}$ de acordo com os valores prescritos.

III-2 Solução do Sistema de Equações de M3 pelo Método Multimalha. O Processador MULGR.

Para a solução do sistema de equações da malha M3 foi implementada uma versão do algoritmo MM - ciclo N apresentado por Zhu e Craig-1985 esquematizada na FIG. III-4, através do processador MULGR, o qual utiliza os bancos de dados dos processadores PREAD, SOL, GERA e PREMG.

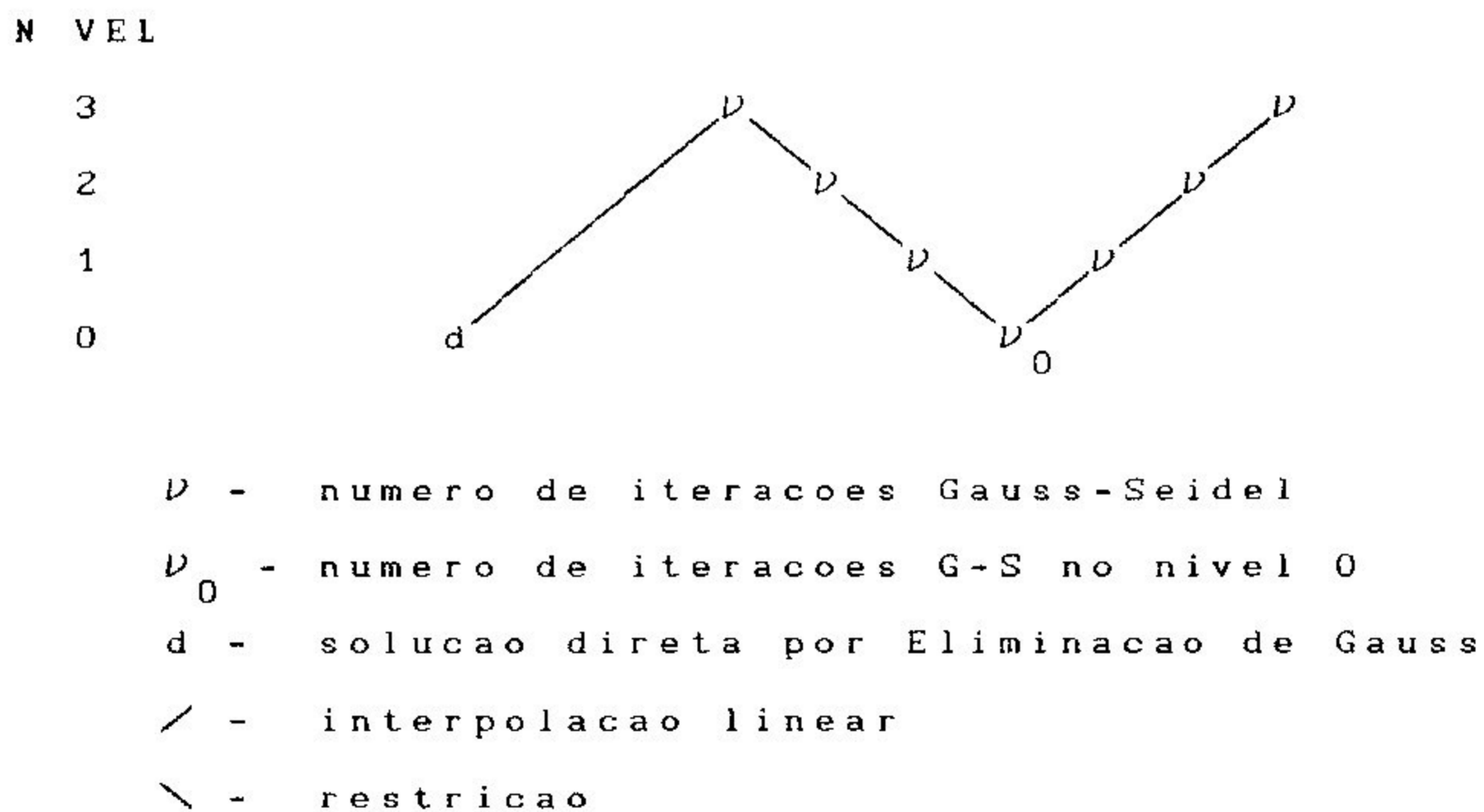


Fig. III-4 Algoritmo esquemático do ciclo N implementado.

III-2-1 Descrição do algoritmo.

Para ν iterações Gauss-Seidel (G-S) por nível

de malha (o número ν é fornecido como dado de entrada no pré-processador PREAD, com exceção do nível 0 onde é utilizado um critério de parada baseado na verificação da convergência), com $\hat{\cdot}$ indicando os valores iniciais para a relaxação em um nível, ε indicando a tolerância, e o índice entre parênteses indicando o nível de malha, o algoritmo pode ser descrito como:

$$\text{(sistema a ser resolvido)} \quad \rightarrow \quad \tilde{K}^{(3)} \tilde{v}^{(3)} = \tilde{f}^{(3)}$$

$$\tilde{v}^{(1)} = \tilde{I}_{(0)}^{(1)} u_h^{(0)}$$

$$\tilde{v}^{(2)} = \tilde{I}_{(1)}^{(2)} \tilde{v}^{(1)}$$

$$\tilde{v}^{(3)} = \tilde{I}_{(2)}^{(3)} \tilde{v}^{(2)}$$

$$\tilde{K}^{(3)} \tilde{v}^{(3)} = \tilde{f}^{(3)} \quad (\nu \text{ vezes})$$

$$\tilde{r}^{(3)} = \tilde{f}^{(3)} - \tilde{K}^{(3)} \tilde{v}^{(3)}$$

$$(1) \quad \tilde{r}^{(2)} = \tilde{I}_{(3)}^{(2)} \tilde{r}^{(3)}$$

$$\text{para } j = 2, 1 \quad \left[\begin{array}{l} \tilde{K}^{(j)} \tilde{e}^{(j)} = \tilde{r}^{(j)} \quad (\hat{\tilde{e}}^{(j)} = 0) \quad (\nu \text{ vezes}) \\ \tilde{r}^{(j)} = \tilde{r}^{(j)} - \tilde{K}^{(j)} \tilde{e}^{(j)} \\ \tilde{r}^{(j-1)} = \tilde{I}_{(j)}^{(j-1)} \tilde{r}^{(j)} \end{array} \right.$$

$$\text{para } j = 0, 1 \quad \left[\begin{array}{l} \tilde{K}^{(j)} \tilde{e}^{(j)} = \tilde{r}^{(j)} \quad \left[\begin{array}{l} j = 0 \Rightarrow \left[\begin{array}{l} \hat{\tilde{e}}^{(j)} = 0 \\ \nu_0 \text{ vezes} \end{array} \right. \\ j = 1 \Rightarrow \nu \text{ vezes} \end{array} \right. \\ \tilde{e}^{(j+1)} = \tilde{I}_{(j)}^{(j+1)} \tilde{e}^{(j)} \\ \hat{\tilde{e}}^{(j+1)} = \tilde{e}^{(j+1)} + \tilde{e}^{(j+1)} \end{array} \right.$$

$$\tilde{K}^{(2)} \tilde{e}^{(2)} = \tilde{r}^{(2)} \quad (\nu \text{ vezes})$$

$$\tilde{e}^{(3)} = \tilde{I}_{(2)}^{(3)} \tilde{e}^{(2)}$$

$$\tilde{v}^{(3)} = \tilde{v}^{(3)} + \tilde{e}^{(3)}$$

$$\tilde{K}^{(3)} \tilde{v}^{(3)} = \tilde{f}^{(3)} \quad (\nu \text{ vezes})$$

$$(2) \quad \tilde{r}^{(3)} = \tilde{f}^{(3)} - \tilde{K}^{(3)} \tilde{v}^{(3)}$$

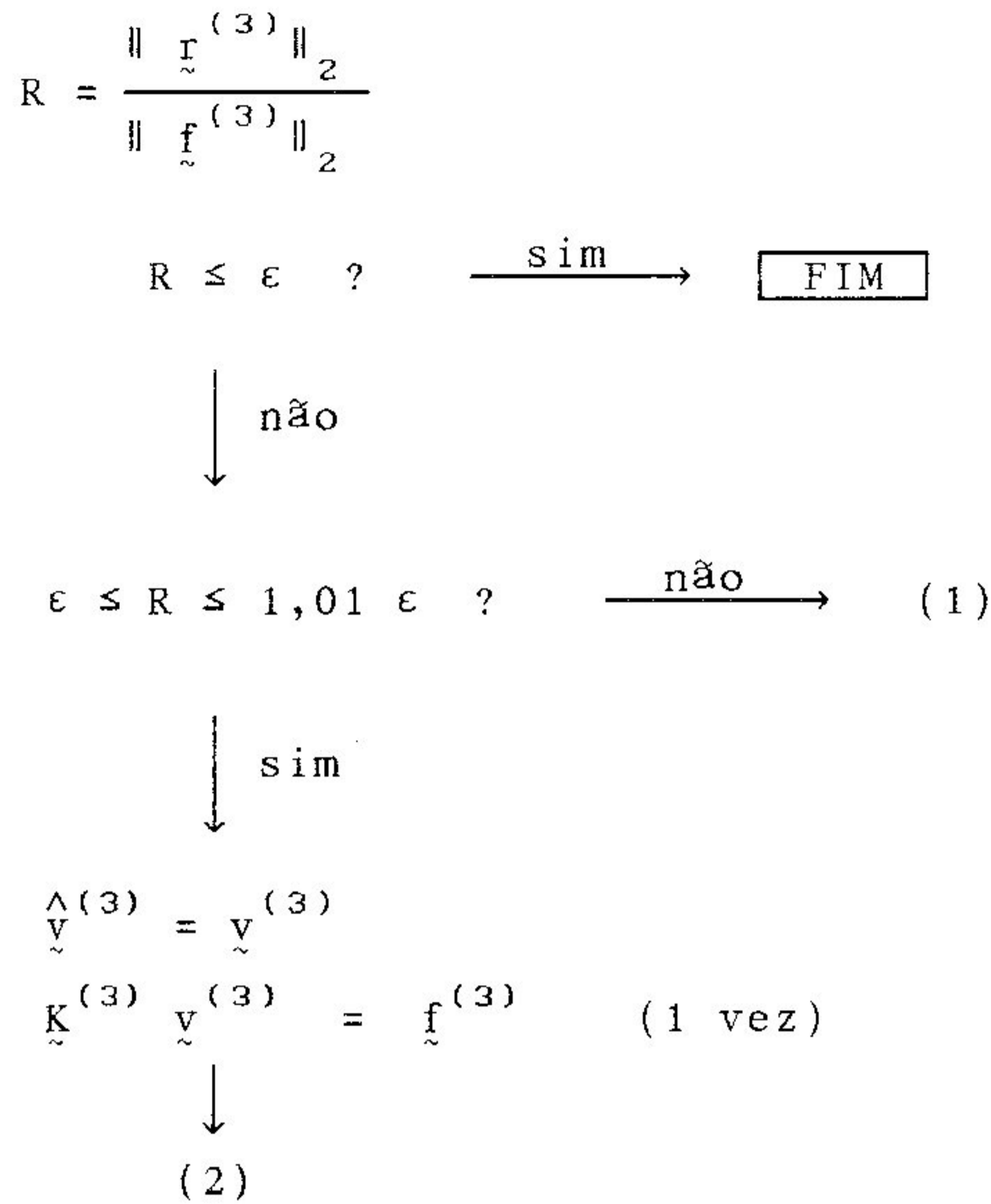


FIG. III-5 Descrição do algoritmo implementado

A FIG. III-5 mostra todos os passos do procedimento, quais sejam:

1- Os deslocamentos obtidos na fase R (nível 0) pelo método de eliminação de Gauss ($\tilde{u}_h^{(0)}$) são interpolados linearmente para os níveis 1, 2 e 3 ($\tilde{v}^{(1)}$, $\tilde{v}^{(2)}$ e $\tilde{v}^{\wedge(3)}$) sucessivamente, através dos operadores de interpolação linear $\tilde{I}_{(0)}^{(1)}$, $\tilde{I}_{(1)}^{(2)}$ e $\tilde{I}_{(2)}^{(3)}$ respectivamente.

2- O sistema de M3 ($\tilde{K}^{(3)} \tilde{v}^{(3)} = \tilde{f}^{(3)}$) é submetido a ν iterações (relaxações) G-S, com $\tilde{v}^{\wedge(3)}$ como aproximação inicial. O número de iterações ν é fornecido nos dados de entrada do processador PREAD.

3- É calculado o resíduo $\tilde{r}^{(3)}$ do nível 3, $\tilde{r}^{(3)} = \tilde{f}^{(3)} - \tilde{K}^{(3)} \tilde{v}^{(3)}$.

4- O resíduo $\tilde{r}^{(3)}$ é transferido para o nível 2 através do operador de restrição $\tilde{I}_{(3)}^{(2)}$, $\tilde{r}^{(2)} = \tilde{I}_{(3)}^{(2)} \tilde{r}^{(3)}$.

5- A equação residual do nível 2 ($\tilde{K}^{(2)} \tilde{e}^{(2)} = \tilde{r}^{(2)}$) é submetida a ν iterações G-S, com aproximação inicial $\tilde{e}^{(2)} = 0$.

6- É calculado o resíduo ($\tilde{r}^{(2)}$) do nível 2, $\tilde{r}^{(2)} = \tilde{r}^{(2)} - \tilde{K}^{(2)} \tilde{e}^{(2)}$.

7- O resíduo $\tilde{r}^{(2)}$ é transferido para o nível 1 através do operador de restrição $\tilde{I}_{(2)}^{(3)}$, $\tilde{r}^{(1)} = \tilde{I}_{(2)}^{(1)} \tilde{r}^{(2)}$.

8- A equação residual do nível 1 ($\tilde{K}^{(1)} \tilde{e}^{(1)} = \tilde{r}^{(1)}$) é submetida a ν iterações G-S com aproximação inicial $\tilde{e}^{(1)} = 0$.

9- É calculado o resíduo ($\tilde{r}^{(1)}$) do nível 1, $\tilde{r}^{(1)} = \tilde{r}^{(1)} - \tilde{K}^{(1)} \tilde{e}^{(1)}$.

10- O resíduo $\tilde{r}^{(1)}$ é transferido para o nível 0 através do operador de restrição $\tilde{I}_{(1)}^{(0)}$, $\tilde{r}^{(0)} = \tilde{I}_{(1)}^{(0)} \tilde{r}^{(1)}$.

11- A equação residual do nível 0 ($\tilde{K}^{(0)} \tilde{e}^{(0)} = \tilde{r}^{(0)}$) é submetida a ν_0 iterações G-S com aproximação inicial $\tilde{e}^{(0)} = 0$. O número de iterações ν_0 é definido pela análise da convergência, em que, para cada ciclo G-S é verificado se

$$\frac{|e_\nu - e_{\nu-1}|}{|e_M|} \leq \epsilon_0$$

onde $|e_\nu - e_{\nu-1}|$ = maior valor absoluto da diferença entre um e do ciclo e o e correspondente do ciclo imediatamente anterior,

$|e_M|$ = maior valor absoluto de e no ciclo,

ϵ_0 = a tolerância definida internamente no processador MULGR de forma a se obter uma boa solução para a malha grossa

12- O erro $\tilde{e}^{(0)}$ obtido é interpolado para o nível 1 através do operador linear $\tilde{I}_{(0)}^{(1)}$, $\tilde{e}^{(1)} = \tilde{I}_{(0)}^{(1)} \tilde{e}^{(0)}$.

13- O erro $\tilde{e}^{(1)}$ obtido pela última relaxação no nível 1 é corrigido pelo erro interpolado, $\tilde{e}^{(1)} = \tilde{e}^{(1)} + \tilde{e}^{(1)}$.

14- A equação residual do nível 1 ($\tilde{K}^{(1)} \tilde{e}^{(1)} = \tilde{r}^{(1)}$) é submetida a ν iterações G-S com aproximação inicial $\tilde{e}^{(1)\wedge}$ e o mesmo $\tilde{r}^{(1)}$ utilizado na última relaxação no nível 1.

15- O erro $\tilde{e}^{(1)}$ obtido é interpolado para o nível 2 através do operador linear $\tilde{I}_{(1)}^{(2)}$, $\tilde{e}^{(2)\wedge} = \tilde{I}_{(1)}^{(2)} \tilde{e}^{(1)}$.

16- O erro $\tilde{e}^{(2)}$ obtido pela última relaxação no nível 2 é corrigido pelo erro interpolado, $\tilde{e}^{(2)\wedge} = \tilde{e}^{(2)} + \tilde{e}^{(2)\wedge}$.

17- A equação residual do nível 2 ($\tilde{K}^{(2)} \tilde{e}^{(2)} = \tilde{r}^{(2)}$) é submetida a ν iterações G-S com aproximação inicial $\tilde{e}^{(2)\wedge}$ e o mesmo $\tilde{r}^{(2)}$ utilizado na última relaxação no nível 2.

18- O erro $\tilde{e}^{(2)}$ obtido é interpolado para o nível 3 através do operador linear $\tilde{I}_{(2)}^{(3)}$, $\tilde{e}^{(3)\wedge} = \tilde{I}_{(2)}^{(3)} \tilde{e}^{(2)}$.

19- Os deslocamentos $\tilde{v}^{(3)}$ obtidos pela última relaxação no nível 3 são corrigidos pelo erro interpolado, $\tilde{v}^{(3)\wedge} = \tilde{v}^{(3)} + \tilde{e}^{(3)\wedge}$.

20- A equação do nível 3, $\tilde{K}^{(3)} \tilde{v}^{(3)} = \tilde{f}^{(3)}$, é submetida a ν iterações G-S com aproximação inicial $\tilde{v}^{(3)\wedge}$.

21- É calculado o resíduo $\tilde{r}^{(3)} = \tilde{f}^{(3)} - \tilde{K}^{(3)} \tilde{v}^{(3)}$ do nível 3.

22- É feita a verificação de convergência utilizando o mesmo critério empregado por Parsons e Hall-1990, isto é, calcula-se

$$R = \frac{\|\tilde{r}^{(3)}\|_2}{\|\tilde{f}^{(3)}\|_2} \quad \text{onde } \|\cdot\|_2 \text{ é a norma euclideana do vetor e } \epsilon$$

é a tolerância definida nos dados de entrada do processador PREAD, e :

a) se $R > 1,01 \epsilon$ retorna-se ao passo (4) iniciando-se outro ciclo V;

b) se $\epsilon < R \leq 1,01 \epsilon$ retorna-se ao passo (20) procedendo-se a mais uma iteração G-S seguindo-se novamente os passos (21) e

(22); esta verificação destina-se a impedir que um novo ciclo MM seja iniciado quando se está muito próximo de satisfazer o critério de convergência, terminando a solução apenas com o algoritmo GS;

c) se $R \leq \epsilon$ o processo é interrompido e os resultados finais são os obtidos no passo (20).

II-2-2 Diagrama de estrutura do processador MULGR.

O diagrama de estrutura do processador MULGR é mostrado na FIG. III-6.

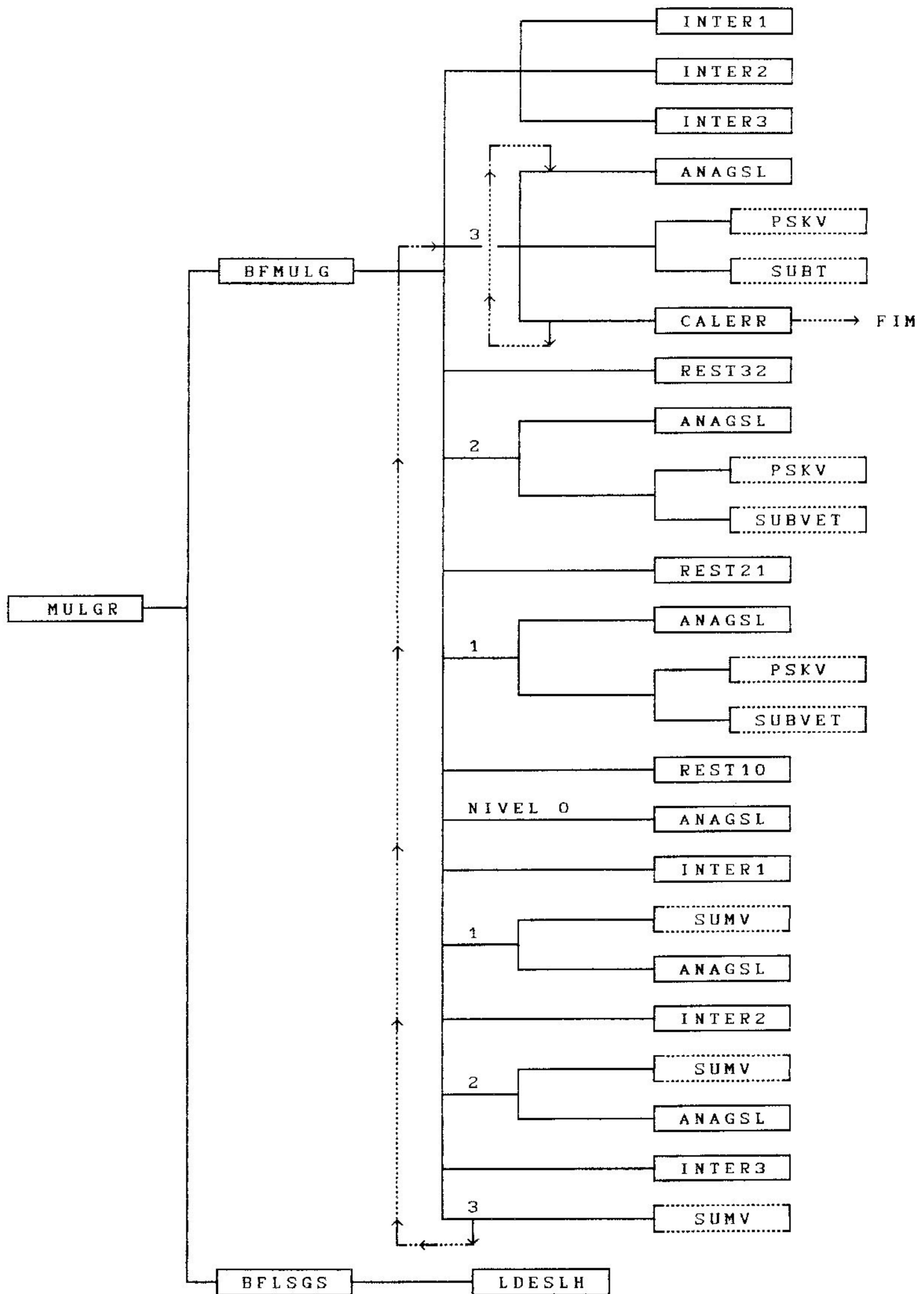


FIG. III-6 Diagrama de estrutura do processador MULGR.

A função de cada rotina da FIG. III-6 é descrita sumariamente a seguir.

* MULGR: processador que resolve o sistema de equações de M3 pelo Método Multimalha;

.BFMULG: gerencia a solução do sistema por MM;

INTER1: interpola linearmente valores de M0 para M1;

INTER2: interpola linearmente valores de M1 para M2;

INTER3: interpola linearmente valores de M2 para M3;

REST32: transfere valores de M3 para M2;

REST21: transfere valores de M2 para M1;

REST10: transfere valores de M1 para M0;

ANAGSL: resolve um sistema de equações lineares por G-S; o número de iterações para os níveis 3, 2 e 1 é fixado pelos dados de entrada de PREAD e para o nível 0 é determinado pela verificação da convergência;

CALERR: verifica a convergência do ciclo V multimalha cujo resultado define se é iniciado outro ciclo V ou interrompido o procedimento;

PSKV: rotina já existente que multiplica uma matriz por um vetor;

SUBT: rotina já existente que subtrai um vetor de outro;

SUBVET: rotina já existente que subtrai um vetor de outro;

SUMV: rotina já existente que soma dois vetores;

.BFLSGS: gerencia a impressão da solução do sistema de M3.

LDESLH: imprime a solução do sistema de equações de M3 (número dos nós e deslocamentos correspondentes).

É importante observar que o operador de interpolação atribui a um novo nó, criado no ponto médio do lado de um elemento, o valor correspondente à média aritmética dos valores dos nós dos vértices vizinhos. O operador de restrição faz a operação inversa, distribuindo metade do valor de um nó que desaparece para cada um dos dois

vizinhos remanescentes (Rivara-1984).

III-3 Cálculo e Impressão de Tensões.

O cálculo e a impressão de tensões foram implementados em um processador chamado POSMG que tem a mesma estrutura do processador POS do programa original, a menos do banco de dados utilizado, o qual é referente à malha M3.

O diagrama de estrutura do processador POSMG encontra-se no APÊNDICE 4.

CAPÍTULO IV

RESULTADOS NUMÉRICOS

Neste capítulo são apresentados os resultados obtidos do processamento de três problemas numéricos - problemas TRINCA, BOUSSINESQ e MANCAL, usando os procedimentos descritos nos Capítulos II e III.

Devido a limitações do equipamento utilizado - micro-computador com processador 80.286 - os problemas foram discretizados através de malhas iniciais muito pobres, fato que introduziu algumas restrições no desempenho do processo R-H utilizado.

Cada problema foi resolvido utilizando-se quatro modelos distintos, gerados a partir da malha inicial segundo os procedimentos descritos no Capítulo II, sendo elas:

- a) malha inicial - MALHA OR-OH;
- b) malha resultante da aplicação de uma série de adaptações R na malha inicial até que o erro tenha sido suficientemente equidistribuído no domínio - MALHA nR-OH, onde n é o número de adaptações R;
- c) malha resultante do refinamento H da malha inicial, feito em três etapas conforme descrito no Capítulo II - MALHA OR-3H;
- d) malha resultante do refinamento H da malha nR-OH, em três etapas - MALHA nR-3H.

O problema BOUSSINESQ foi resolvido também analiticamente, através da Teoria da Elasticidade e o problema TRINCA foi também resolvido através de uma malha bem refinada, utilizada para este problema por Cimini-1991 - MALHA CIMINI.

Os resultados em tensões obtidos para as diversas malhas de cada problema foram comparados entre si, com o objetivo de analisar a eficácia do processo adaptativo

R-H global.

O sistema de equações da malha nR-3H de cada problema foi resolvido utilizando-se três processos independentemente:

- a) método direto de Eliminação de Gauss (EG);
- b) processo iterativo de Gauss-Seidel por linhas (GS);
- c) método Multimalha conforme algoritmo descrito no Capítulo III (MM).

Os resultados em tensões obtidos através dos três processos foram comparados entre si com o objetivo de:

- a) verificar a validade e a precisão dos processos MM e GS implementados;
- b) comparar os tempos de CPU demandados pelos processamentos - análise de eficiência; os tempos de CPU apresentados referem-se a um micro-computador 486.

A solução dos sistemas de equações por MM foi processada várias vezes mudando-se os parâmetros do algoritmo - número de iterações GS em cada nível de malha (ν), erro tolerado no critério de parada (ϵ) e erro tolerado no critério de parada das iterações GS na malha do nível 0 (ϵ_0). Teve-se, com isto, o objetivo de encontrar a combinação de parâmetros mais eficiente para cada problema - parametrização do MM.

As malhas resultantes dos três refinamentos H juntamente com a malha inicial - nR-0H, nR-1H, nR-2H, nR-3H - correspondem aos quatros níveis de malha M0, M1, M2 e M3 do algoritmo MM implementado, conforme descrito no Capítulo III.

Procedeu-se, também, a uma análise do desempenho do MM em relação à distorção das malhas, comparando-se o número de ciclos MM demandado pela solução dos sistemas das malhas nR-3H dos três problemas.

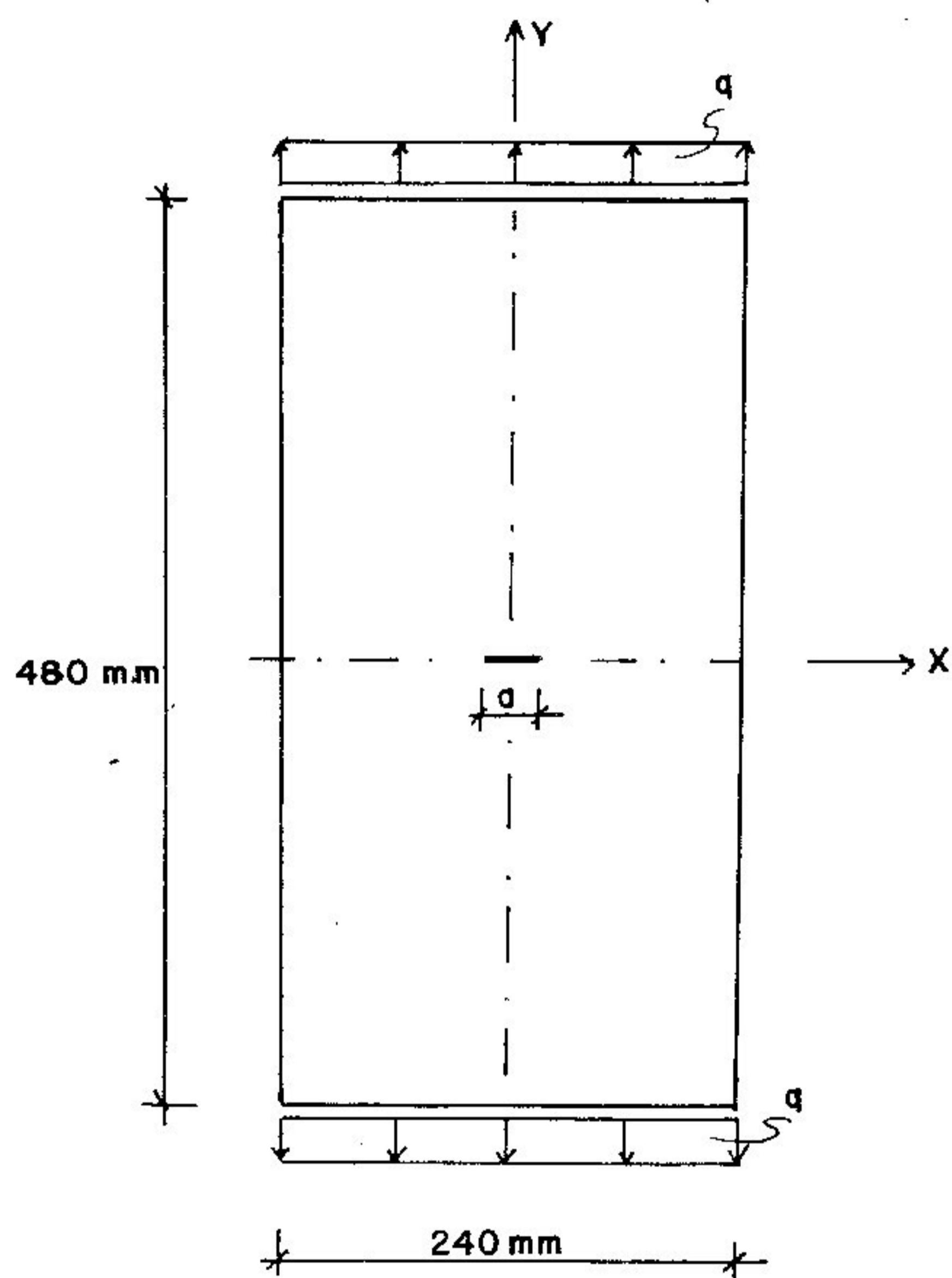
Verificou-se, também, o conhecido aumento do número de ciclos MM com o aumento da incompressibilidade do material (coeficiente de Poisson).

Os problemas TRINCA e BOUSSINESQ encontram similares em Parsons e Hall-1990, cujos parâmetros e resultados foram tomados como referências relativas.

IV-1 Problema TRINCA

IV-1-1 Descrição do problema - Definição das malhas

O problema TRINCA refere-se à chapa tracionada com uma trinca na sua região central, mostrada na FIG. IV-1.



$a = 20 \text{ mm}$ $q = 100 \text{ kgf/mm}$ $E = 500 \text{ kgf/mm}^2$ $\nu = 0.30$
 $e = \text{espessura} = 1.0 \text{ mm}$ Estado Plano de Tensão

FIG. IV-1 Problema TRINCA : chapa tracionada com trinca na região central.

A discretização inicial (MALHA OR-0H) foi construída para o quarto superior da chapa, FIG. IV-2, estando a trinca representada pelo lado do elemento 1 que liga os nós 1 e 2.

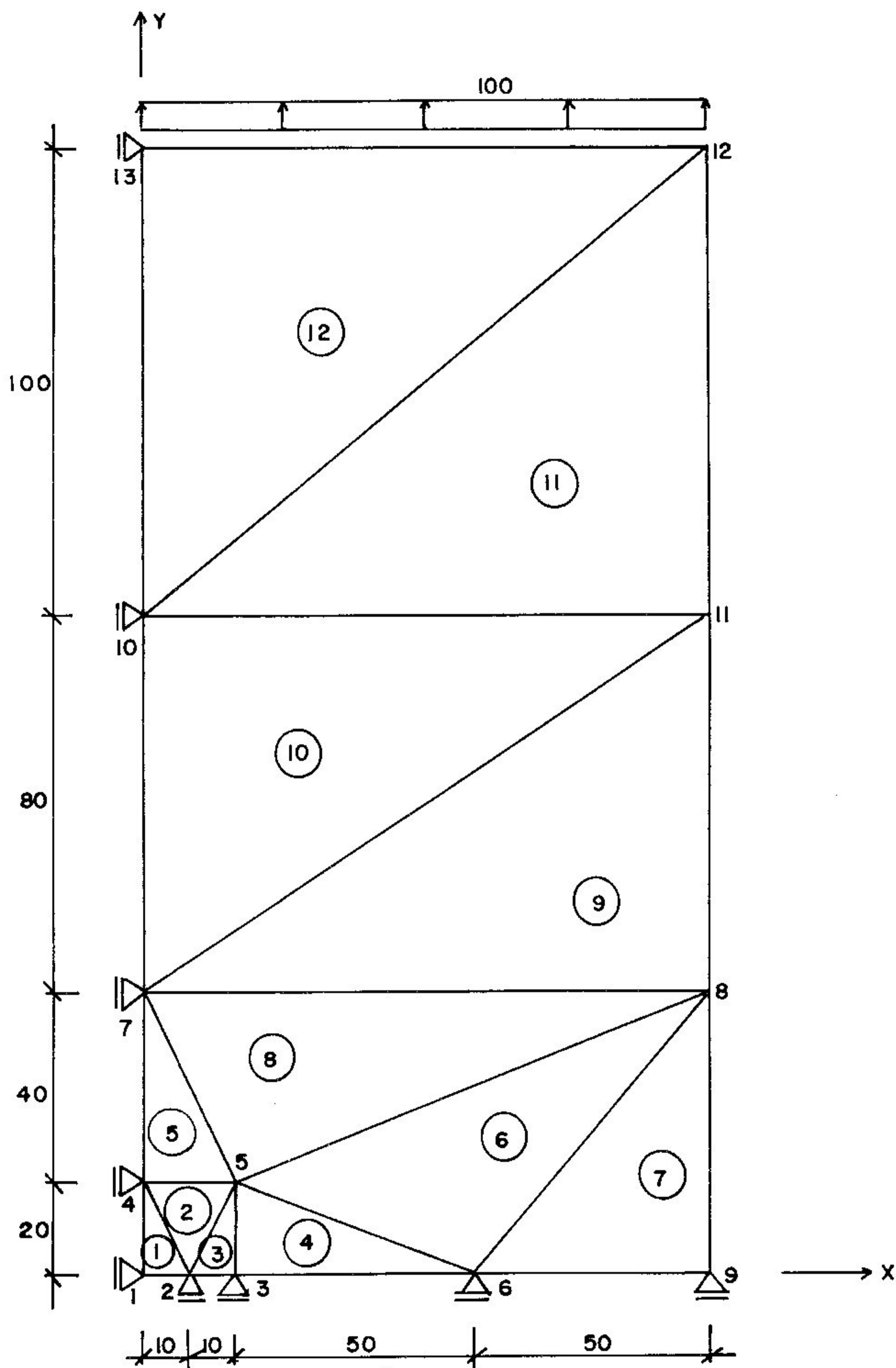


FIG. IV-2 Problema TRINCA
Malha OR-OH.

Para efeito do procedimento R, os nós 1,2,9,12 e 13 são impedidos de sofrerem qualquer deslocamento (relocação), os nós 3 e 6 são livres para o deslocamento segundo a direção x, os nós 4, 7, 8, 10 e 11 são livres para o deslocamento segundo a direção y e o nó 5 é livre para se deslocar em qualquer direção.

Foram executados cinco ciclos R na malha 0R-0H, ao fim dos quais o erro foi considerado suficientemente equidistribuído no domínio. A sequência de malhas obtidas com as adaptações R é mostrada na FIG. IV-3.

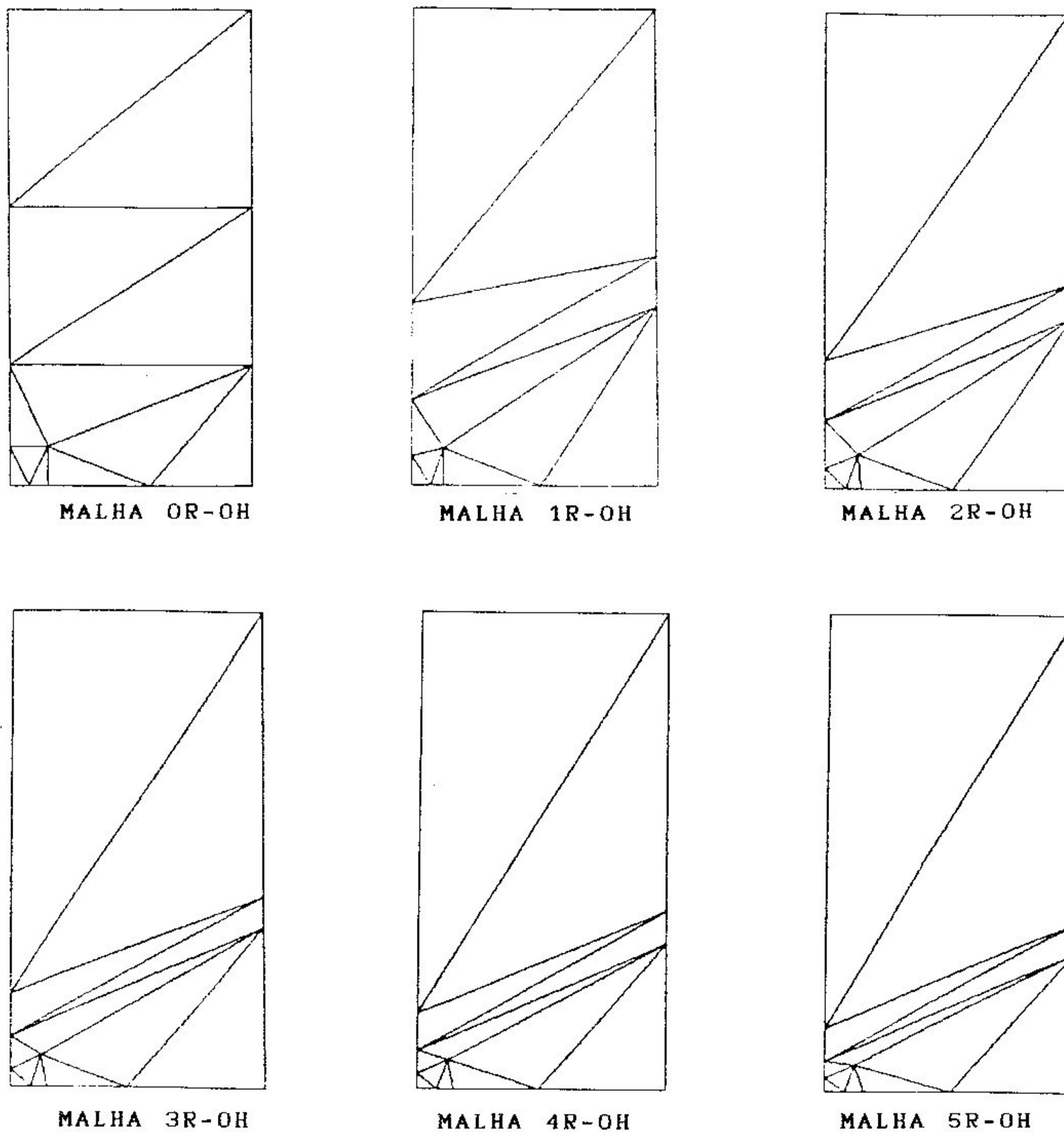


FIG. IV-3 Problema TRINCA

Sequência de malhas obtidas com 5 adaptações R sucessivas a partir da malha 0R-0H.

Pode-se notar, apenas por inspeção, a convergência da sequência de malhas da FIG. IV-3.

Um detalhe da malha 5R-3H, obtida pelo refinamento H global da malha 5R-0H três vezes sucessivas, é mostrado na FIG. IV-4, onde está feito o refinamento apenas do elemento 1 da malha 3R-0H.

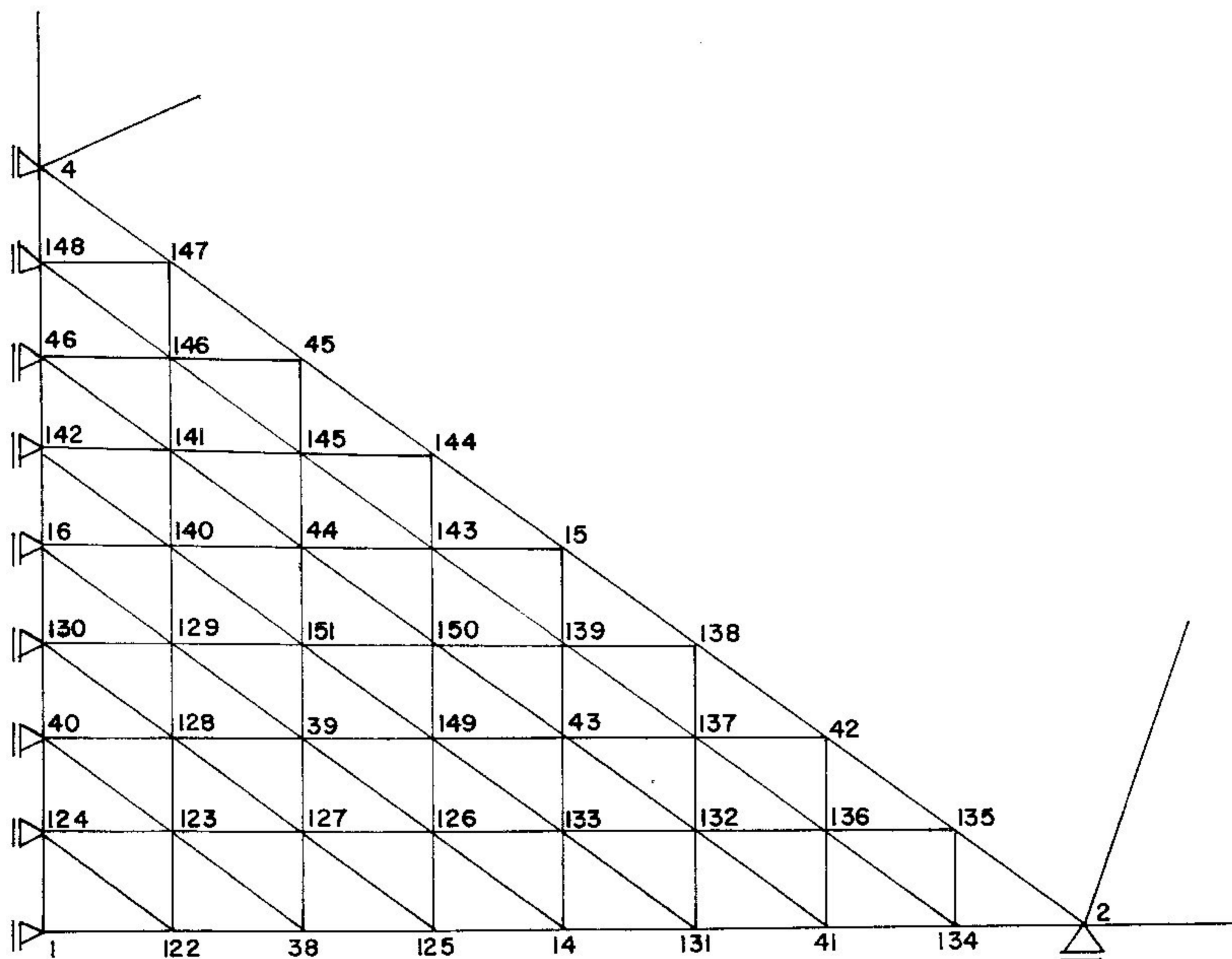


FIG. IV-4 Problema TRINCA
 Detalhe da malha 5R-3H.

O detalhe correspondente da malha OR-3H tem aspecto semelhante ao da FIG. IV-4.

As malhas nR-3H constituem-se em problemas de 808 graus de liberdade (433 nós, 768 elementos).

A malha CIMINI é mostrada na FIG. IV-5.

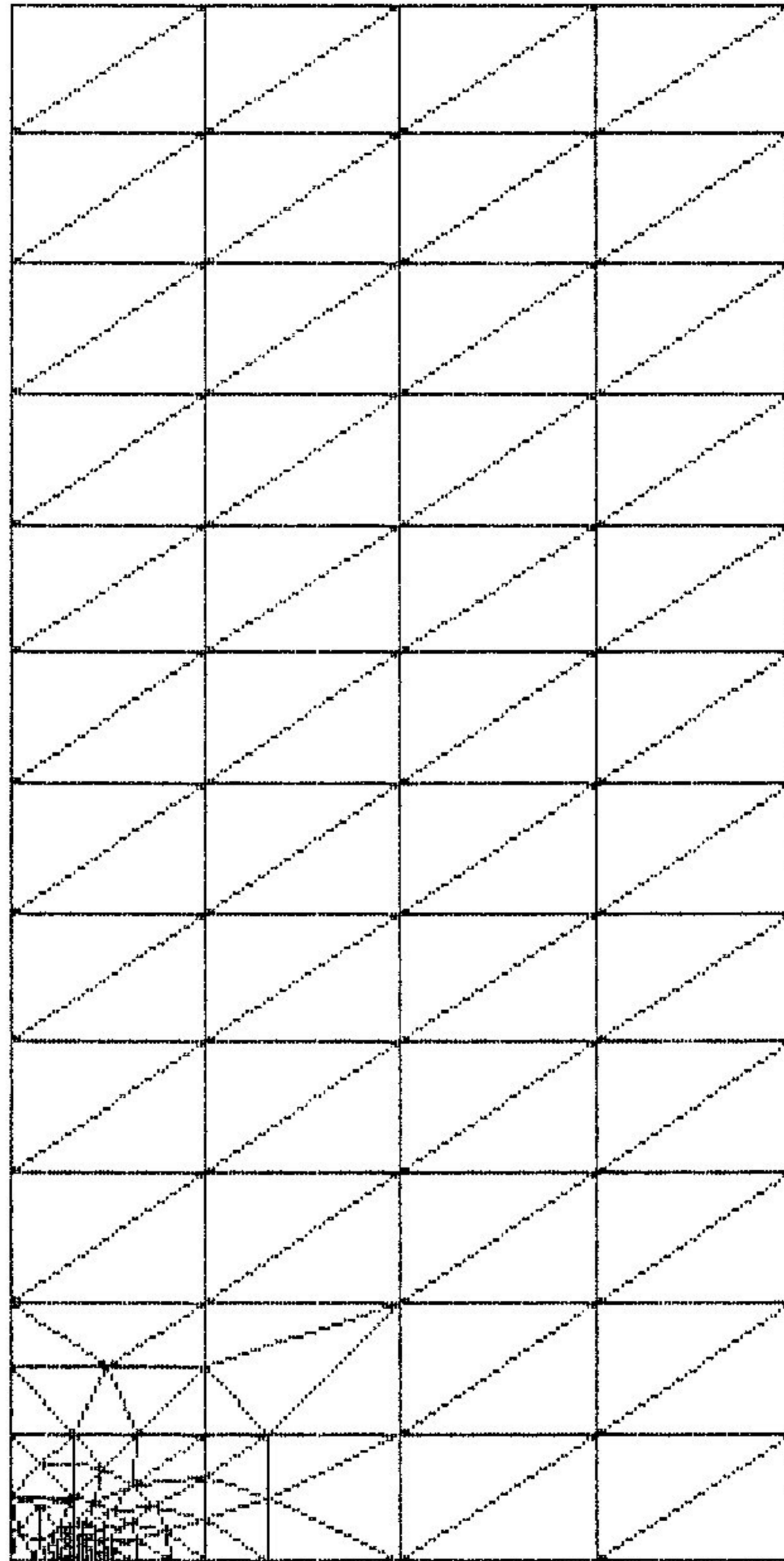


FIG. IV-5.a Problema TRINCA
Malha CIMINI

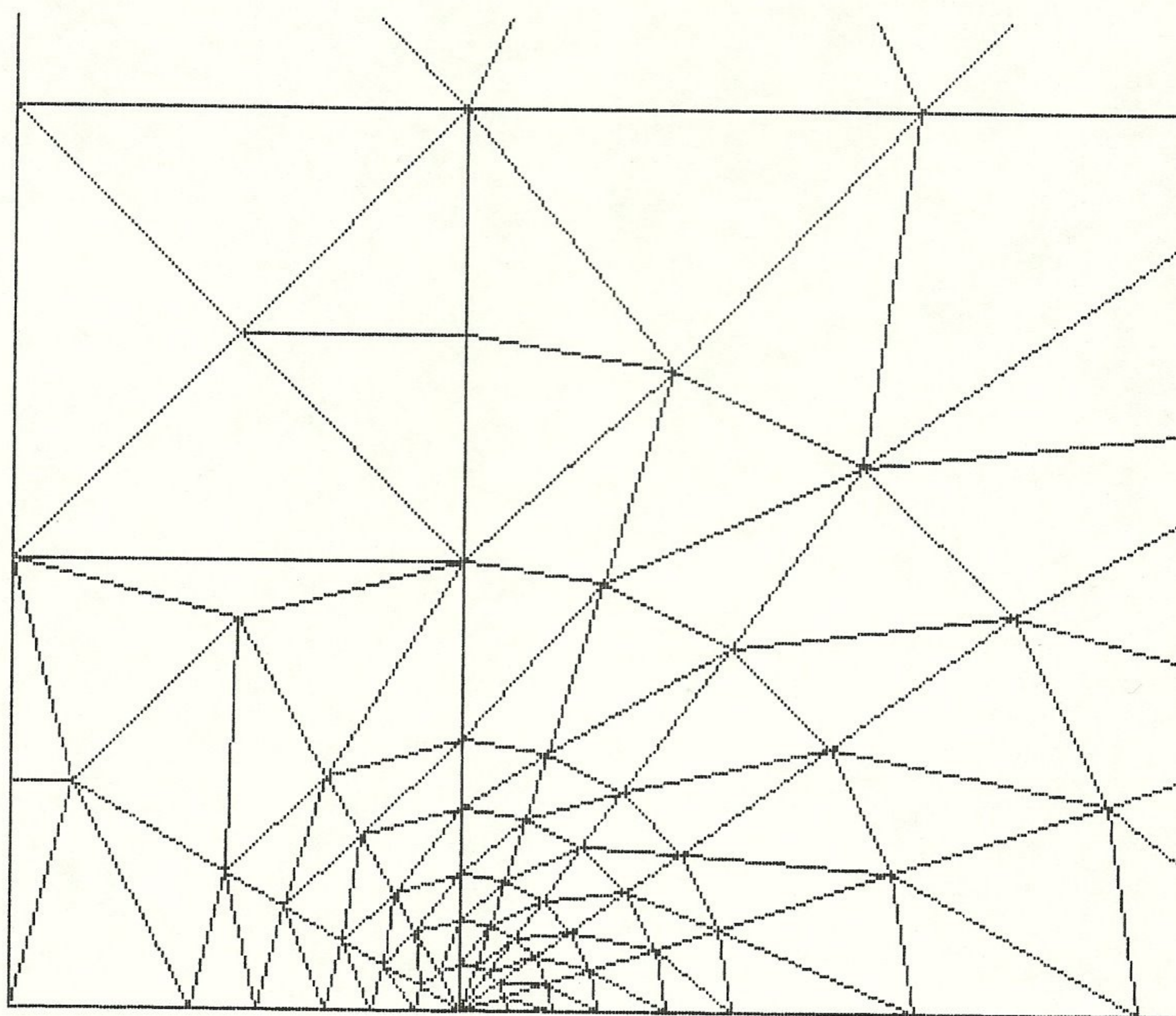


FIG. IV-5.b Problema TRINCA

Detalhe da malha CIMINI na região da trinca.

IV-1-2 Análise do desempenho do algoritmo MM

a) Parametrização do algoritmo MM

Na solução dos sistemas de equações das malhas 5R-3H e 0R-3H por MM e por GS, foi empregada a tolerância para o erro algébrico, utilizada no critério de parada dos algoritmos MM e GS, de $\epsilon = 1*10^{-6}$, a qual produz resultados em tensões coincidentes com os obtidos por EG. Este valor foi o empregado por Parsons e Hall. Uma redução deste valor para $1*10^{-5}$ produz resultados em tensões, para este problema, com diferenças da ordem de 5% em relação aos do EG.

A resposta do algoritmo MM à variação de dois parâmetros, número de iterações GS por nível de malha (ν) e tolerância para o erro algébrico da solução do sistema equações do nível 0 (ϵ_0), é mostrada na TAB. IV-1, a qual refere-se à malha 5R-3H.

ϵ_0	ν	N. CICL. MM	N. ITER. NÍVEL 0	N. ITER. FIN. M3	TEMPO CPU (SEG.)
10D-6	5	49	66	5	311,420
10D-3	5	49	28	5	306,040
10D-3	8	31	30	10	290,450
10D-3	10	26	30	10	296,200

TAB. IV-1 Problema TRINCA

Parametrização do MM - Malha 5R-3H

$$\epsilon = 1*10^{-6}.$$

Na TAB. IV-1 tem-se:

coluna 1 - tolerância ϵ_0 para critério de parada das iterações GS na malha do nível 0;

coluna 2 - número ν de iterações GS nas malhas dos níveis 1, 2 e 3 do MM;

coluna 3 - número de ciclos MM requerido para a convergência;

coluna 4 - número de iterações GS na última vez que é feita a relaxação na malha do nível 3;

coluna 5 - tempo de CPU requerido para o processamento da

solução do sistema por MM.

Pode-se notar que o valor de $\epsilon_0 = 10^{-3}$ é suficiente para a convergência do processo GS no nível 0; uma redução desse valor para 10^{-6} não diminuiu o número de ciclos MM total, apenas aumentou desnecessariamente o número de iterações no nível 0, aumentando o tempo de CPU.

Fixado ϵ_0 em 10^{-3} , procedeu-se à variação de ν . Constatou-se que, para este problema, o valor $\nu = 8$ é o mais eficiente, uma vez que sua diminuição para $\nu = 5$, ao aumentar o número de ciclos MM, mesmo com um número menor de iterações GS por nível, aumenta o tempo de CPU. Já o seu aumento para $\nu = 10$, apesar de diminuir bastante o número de ciclos MM, é excessivo, fazendo também aumentar o tempo de CPU. As alterações no desempenho observadas, neste exemplo, com a variação de ν e ϵ_0 são pequenas.

b) Fator irregularidade da malha

Uma característica importante constatada na solução dos sistemas por MM é a influência da irregularidade da malha no número de ciclos MM necessários para a convergência.

Pela FIG. IV-3 pode-se ver que, neste caso, as adaptações R aumentam sucessivamente a irregularidade da malha inicial OR-0H, que por sua vez já se mostrava bastante irregular. O processamento da solução dos sistemas das malhas OR-3H, 1R-3H, 2R-3H, 3R-3H, 4R-3H e 5R-3H, isto é, malhas que resultam do refinamento 3H das malhas mostradas na FIG. IV-3, mostra claramente o aumento sensível do número de ciclos MM necessários para a convergência com o aumento da irregularidade da malha analisada - TAB. IV-2.

MALHA	N. CICL. MM	TEMPO CPU (SEG.)
0R-3H	10	65,910
1R-3H	16	104,120
2R-3H	27	170,380
3R-3H	35	219,580
4R-3H	42	264,070
5R-3H	49	306,040

TAB. IV-2 Problema TRINCA

Aumento do número de ciclos MM com o aumento da irregularidade da malha.

$$\nu = 5 \quad \epsilon = 10^{-6} \quad \epsilon_0 = 10^{-3}$$

c) Análise da eficiência do algoritmo MM

A solução do sistema de equações da malha 5R-3H foi também obtida utilizando-se EG e GS. Os tempos de CPU gastos pelos três processos (EG, GS e MM) são apresentados na TAB. IV-3. Os resultados do MM referem-se a $\nu = 8$, $\epsilon = 10^{-6}$ e $\epsilon_0 = 10^{-3}$, os quais produziram os melhores resultados na parametrização. No processo GS foi utilizada a mesma tolerância $\epsilon = 10^{-6}$ para o critério de parada.

PROCESSO	N ITER GS* N CIC MM**	TEMPO CPU (SEG.)
EG	—	6,680
GS *	3812	5083,620
MM **	31	290,450

TAB. IV-3 Problema TRINCA - Malha 5R-3H

Tempo de CPU (seg.) gasto na solução do sistema pelos processos EG, GS, MM.

Pode-se constatar que o método direto EG é o mais eficiente para este problema. O método MM gastou 43,48 vezes mais tempo de CPU que o EG e o método GS gastou 17,5 vezes mais tempo de CPU que o MM e 761,02 vezes mais que o

EG. O número total de iterações GS para o MM no nível 3 foi de 248. Neste caso, portanto, o MM mostrou-se muito vantajoso em relação ao outro processo iterativo (GS), mas muito desvantajoso em relação ao método direto (EG).

Análise semelhante feita na malha menos irregular (0R-3H) produziu os resultados mostrados na TAB. IV-4.

PROCESSO	N ITER GS* N CIC MM**	TEMPO CPU (SEG.)
EG	—	6,680
GS *	2296	3060,010
MM **	10	65,910

TAB. IV-4 Problema TRINCA - Malha 0R-3H.

Tempo de CPU gasto na solução do sistema pelos processos EG, GS e MM ($\nu = 5$).

Na malha menos irregular o MM gastou 9,87 vezes mais tempo de CPU que o EG. O GS gastou 46,43 vezes mais tempo de CPU que o MM e 458,09 vezes mais que o EG. O número total de iterações GS para o MM foi de 80 no nível 3.

Pode-se concluir que, neste problema, o aumento da eficiência do MM com o aumento da regularidade da malha é maior que o aumento da eficiência do EG e GS nas mesmas condições.

Considerando que os processos iterativos comuns são tão mais eficientes em relação aos métodos diretos quanto maior a dimensão do problema e que o MM mostrou-se bastante vantajoso em relação ao GS, pode-se supor que ao aumentar-se a dimensão do problema em questão, o MM acabe por superar o EG em velocidade de processamento. Esta suposição não pôde ser comprovada neste trabalho devido às limitações do equipamento utilizado. Pôde-se constatar aqui, apenas a grande eficiência do MM quando comparado a um dos processos iterativos de uso corrente (GS por linhas).

d) Comparação com resultados de Parsons e Hall

Os resultados, em termos de número de ciclos MM, obtidos para o problema TRINCA foram comparados com os obtidos por Parsons e Hall para um problema semelhante (FIG. IV-6), discretizado através da malha mostrada na (FIG. IV-7)

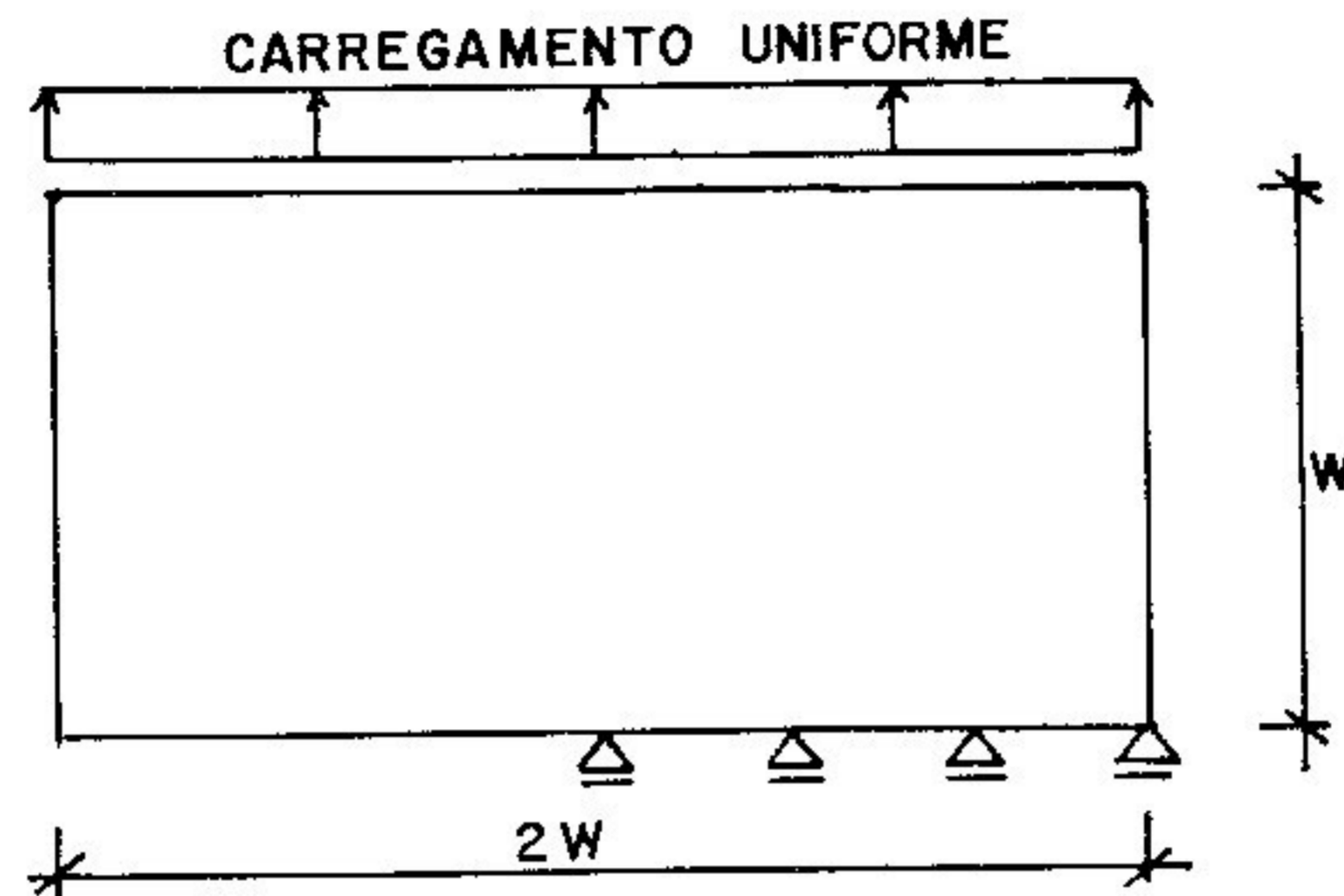


FIG. IV-6 Problema semelhante ao TRINCA
Parsons e Hall-1990.

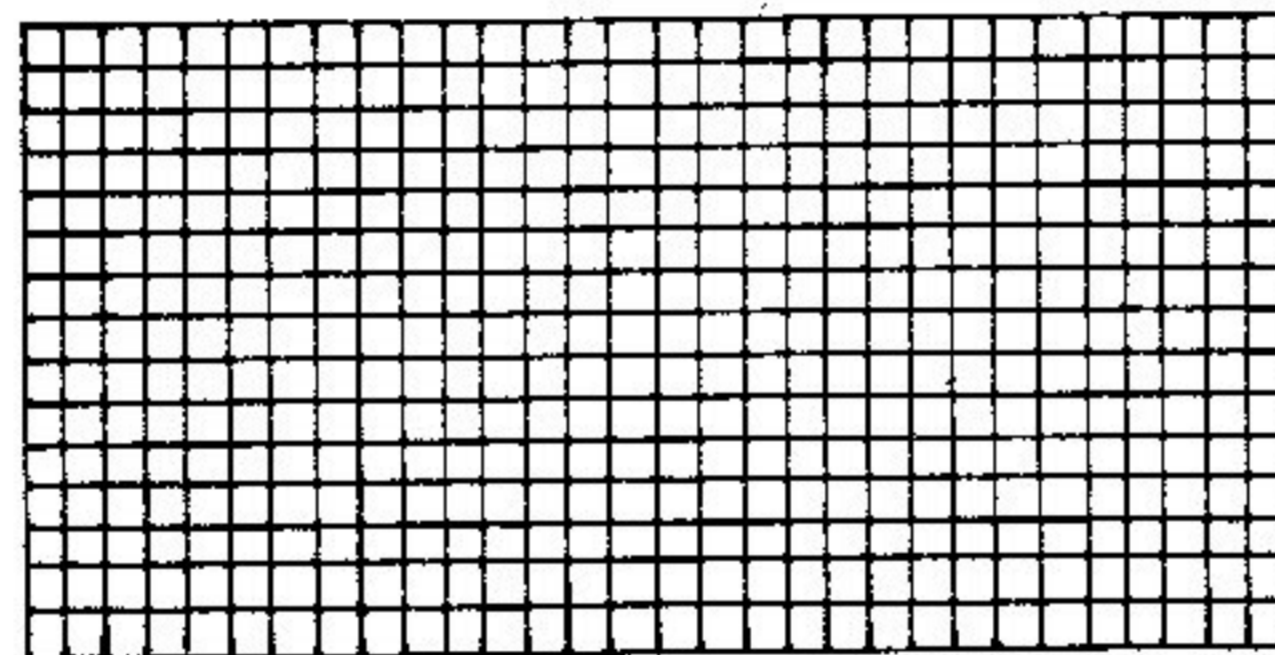


FIG. IV-7 Malha usada por Parsons e Hall para
o problema da FIG. IV-6

Considerando que o elemento utilizado por Parsons e Hall, retangular de quatro nós, é superior ao aqui empregado, e que a malha lá empregada é regular contra uma aqui muito irregular (malha OR-3H), pode-se considerar os resultados aqui obtidos como satisfatórios. Para $\nu = 5$ e $\epsilon = 10^{-6}$,

Problema \approx TRINCA (Parsons e Hall) \rightarrow N. de ciclos MM = 11

Problema TRINCA (malha OR-3H) \rightarrow N. de ciclos MM = 10.

e) Evolução do erro com os ciclos MM

A FIG. IV-8 mostra como o erro diminui após cada ciclo MM até a convergência. O erro em questão é o utilizado para a verificação da convergência descrito no Capítulo III, isto é.

$$R = \frac{\| \tilde{r}^{(3)} \|_2}{\| \tilde{f}^{(3)} \|_2}$$

onde $\tilde{r}^{(3)}$ é o resíduo e $\tilde{f}^{(3)}$ é o vetor de forças, ambos no nível 3.

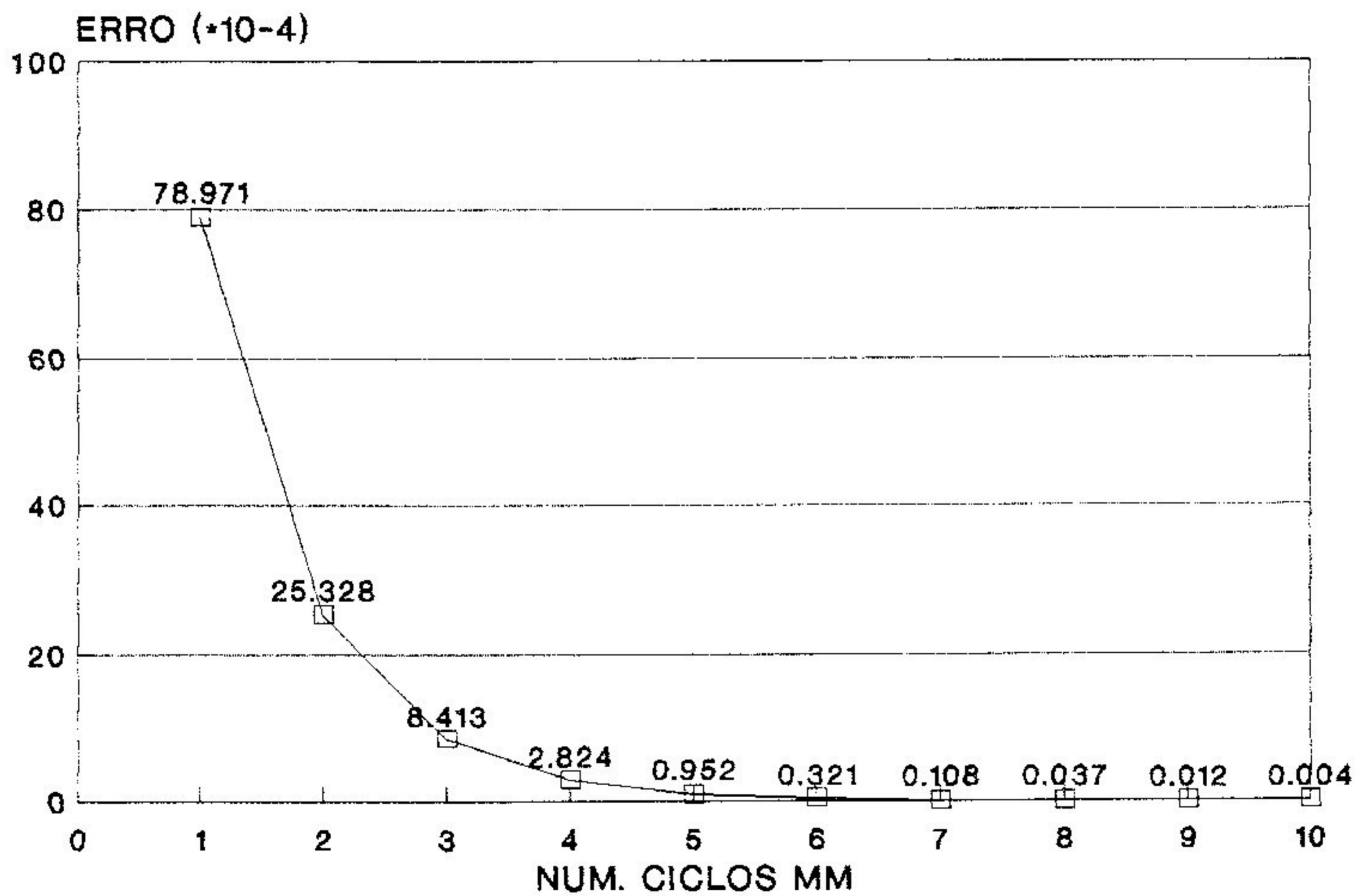


FIG. IV-8 Problema TRINCA. Malha OR-3H. $\nu=5$, $\epsilon=10^{-6}$.
Evolução do erro após cada ciclo MM.
(Número de ciclos MM = 10)

IV-1-3 Análise da eficácia do processo adaptativo R-H global

Os resultados em tensões σ_y obtidos com as malhas 0R-0H, 5R-0H, 0R-3H, 5R-3H e CIMINI, na linha da trinca (coordenadas $(x;0,0)$), são mostradas nas figuras seguintes.

A FIG. IV-9 mostra os resultados obtidos com as malhas 0R-0H, 5R-0H e CIMINI, com o objetivo de constatar a pobreza da malha inicial e a relativa melhora dos resultados quando se procede às adaptações R, quando estes tendem a se aproximar do resultado da malha CIMINI, considerado um resultado satisfatório.

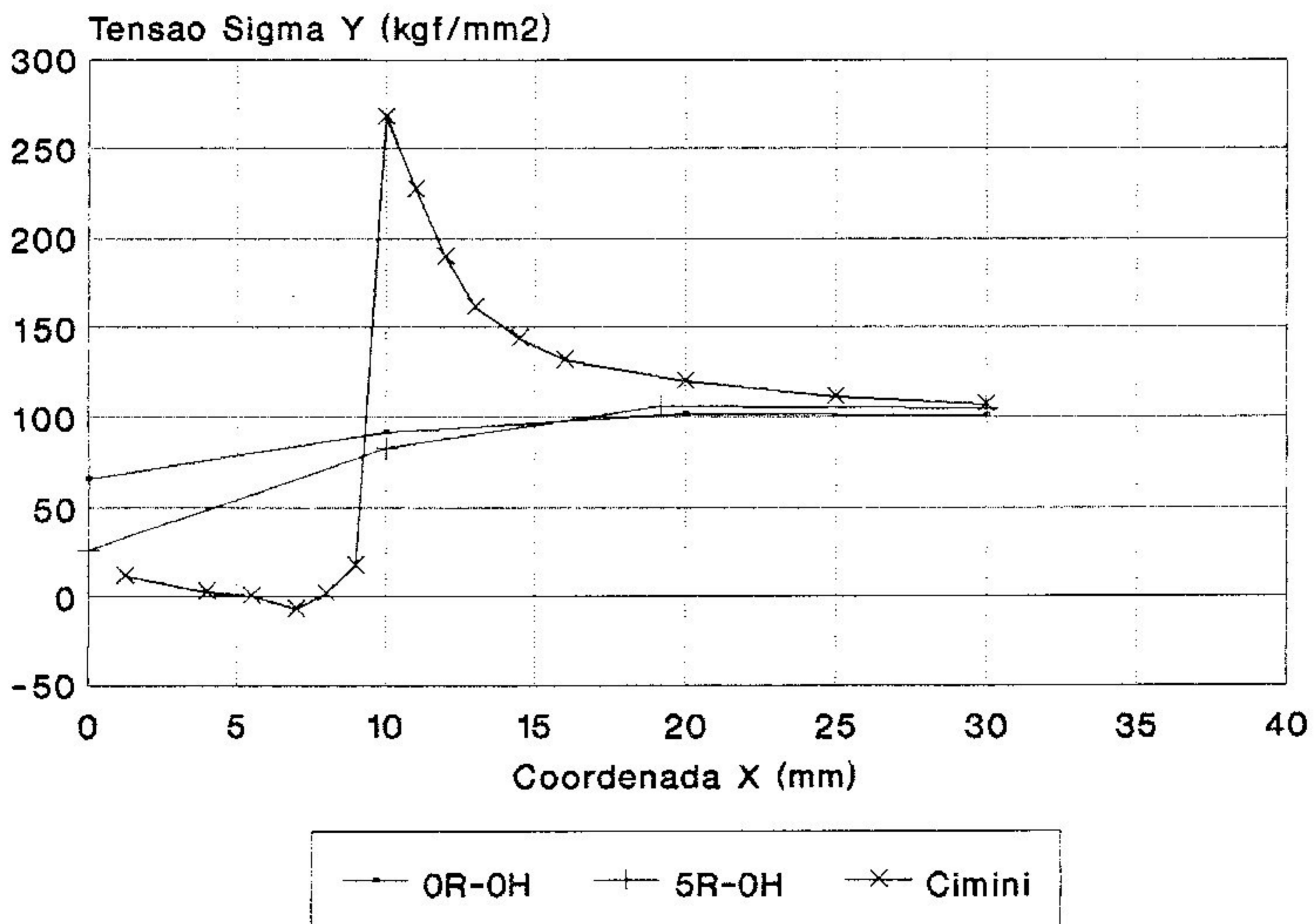


FIG. IV-9 Problema TRINCA
Tensões σ_y em $(x;0,0)$, obtidos com as malhas 0R-0H, 5R-0H e CIMINI.

A FIG. IV-10 mostra as tensões σ_y na linha da trinca, obtidas com as malhas 0R-3H e 5R-3H, onde se constata que as adaptações R feitas previamente na malha 0R-0H melhoram significativamente os resultados quando comparados com os resultados da malha submetida ao refinamento H sem a prévia adaptação R (0R-3H).

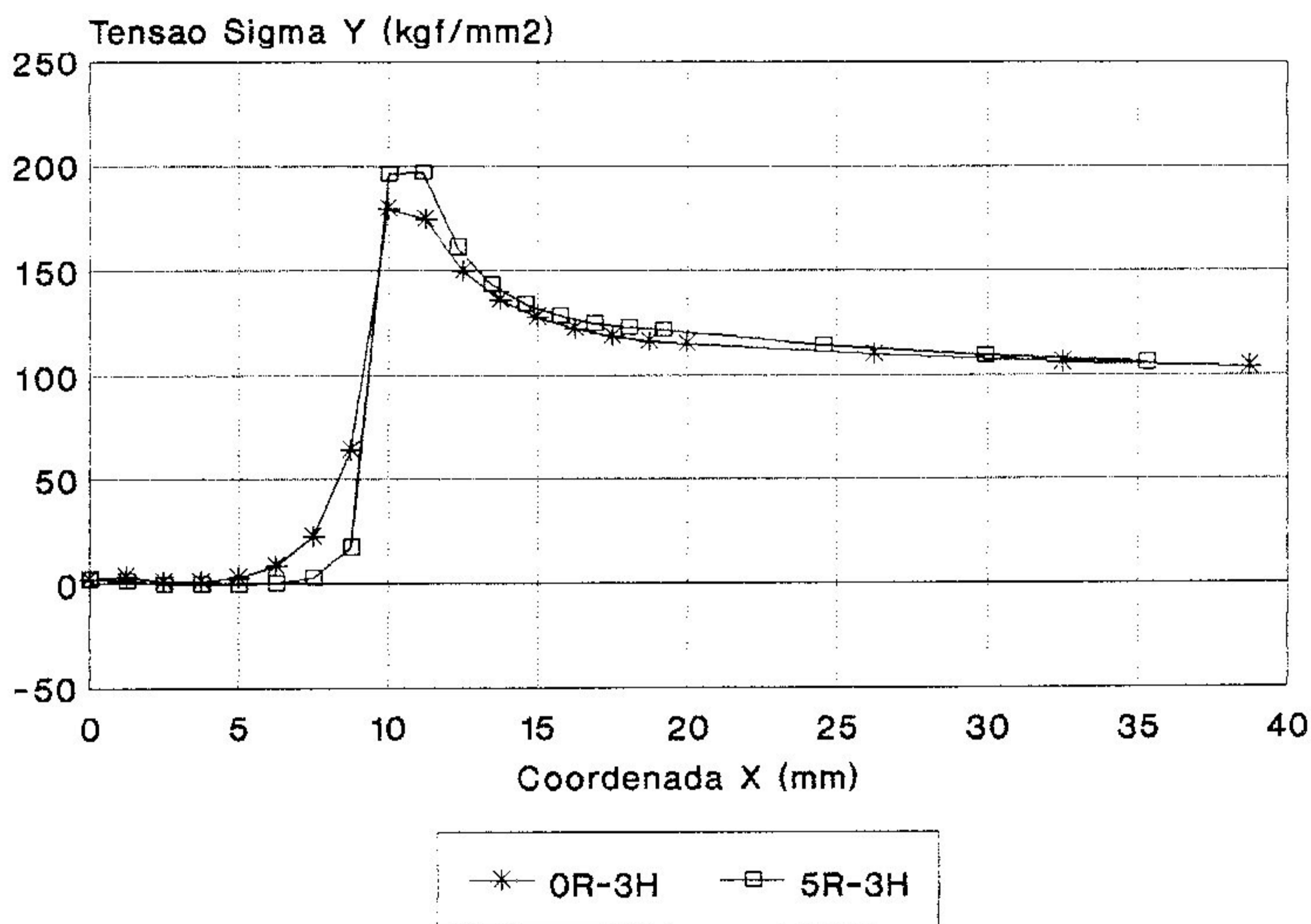


FIG. IV-10 Problema TRINCA
Tensões σ_y em $(x;0,0)$ obtidas com as malhas 0R-3H e 5R-3H.

Na FIG. IV-11 as tensões σ_y em $(x;0,0)$ na malha 5R-3H estão mostradas juntamente com as obtidas com a malha CIMINI, onde se constata que o processo adaptativo realizado na malha inicial levou a resultados bastante bons, ainda que a malha inicial (0R-0H) tenha sido muito pobre, fato que limita a eficácia das adaptações R.

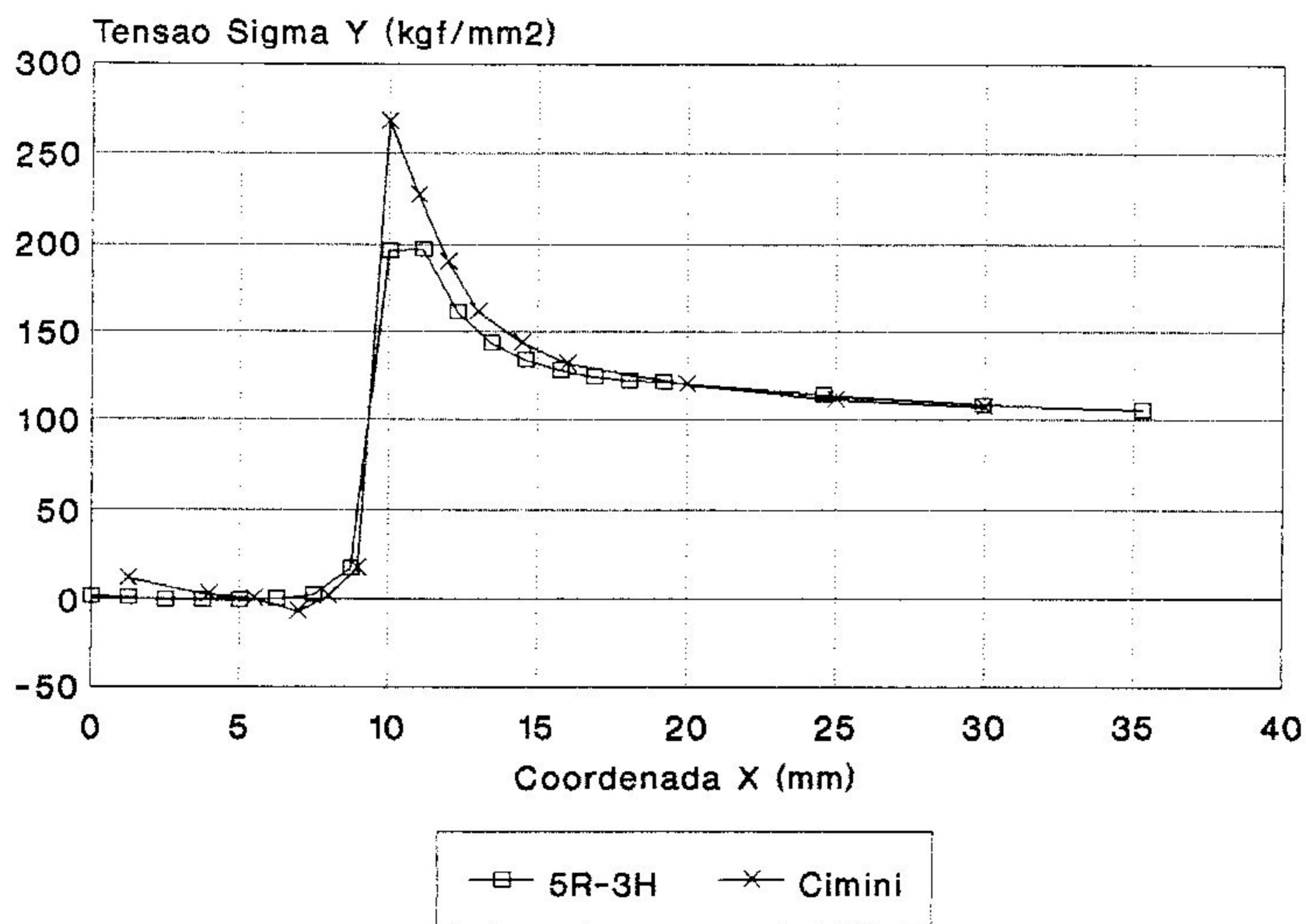


FIG. IV-11 Problema TRINCA

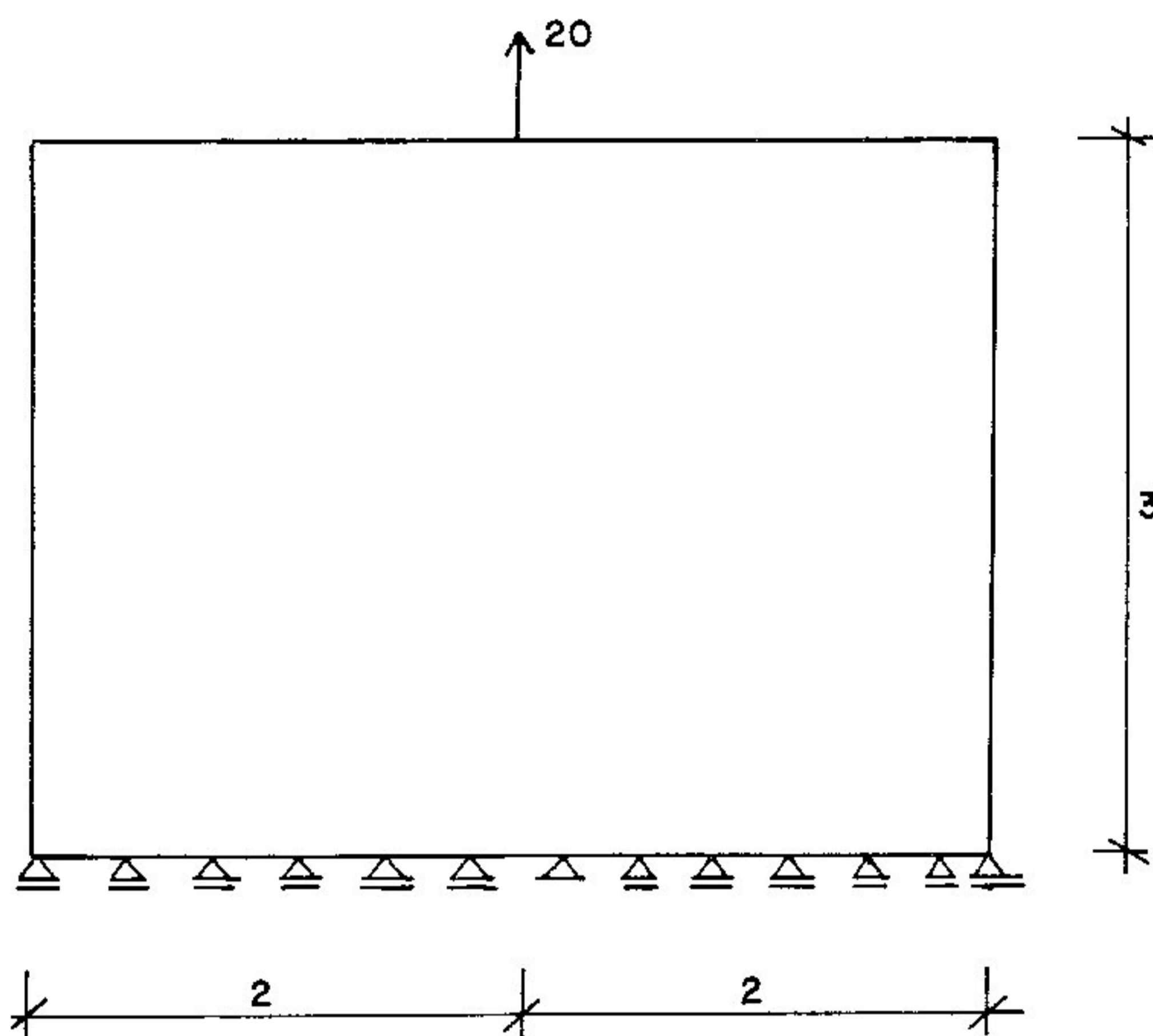
Tensões σ_y em $(x;0,0)$ obtidas com as malhas 5R-3H e CIMINI.

Pode-se, portanto, esperar que quando o equipamento utilizado comportar uma malha final maior que a apresentada, o que por consequência permitirá uma malha inicial menos pobre, tanto a eficácia do processo R-H global quanto a eficiência do MM em relação ao EG aumentem significativamente.

IV-2 Problema BOUSSINESQ

IV-2-1 Descrição do problema - Definição das malhas

O problema BOUSSINESQ é constituído de um semi-plano composto de material elástico e isotrópico, sujeito a uma carga concentrada de tração aplicada no ponto médio de um dos lados, conforme mostrado na FIG. IV-12.



$$E = 2100$$

$$\nu = 0,3$$

Estado Plano de Deformação

FIG. IV-12 Problema BOUSSINESQ.

A malha inicial OR-OH construída para este problema é mostrada na FIG. IV-13. Foi discretizada a metade esquerda da chapa. Construiu-se uma malha muito regular, apesar da carga concentrada solicitar um maior refinamento na região de sua aplicação, com o objetivo de analisar com mais

propriedade a influência da regularidade da malha na eficiência da solução do sistema de equações por MM e a eficácia do processo R na melhora dos resultados de malhas muito pobres.

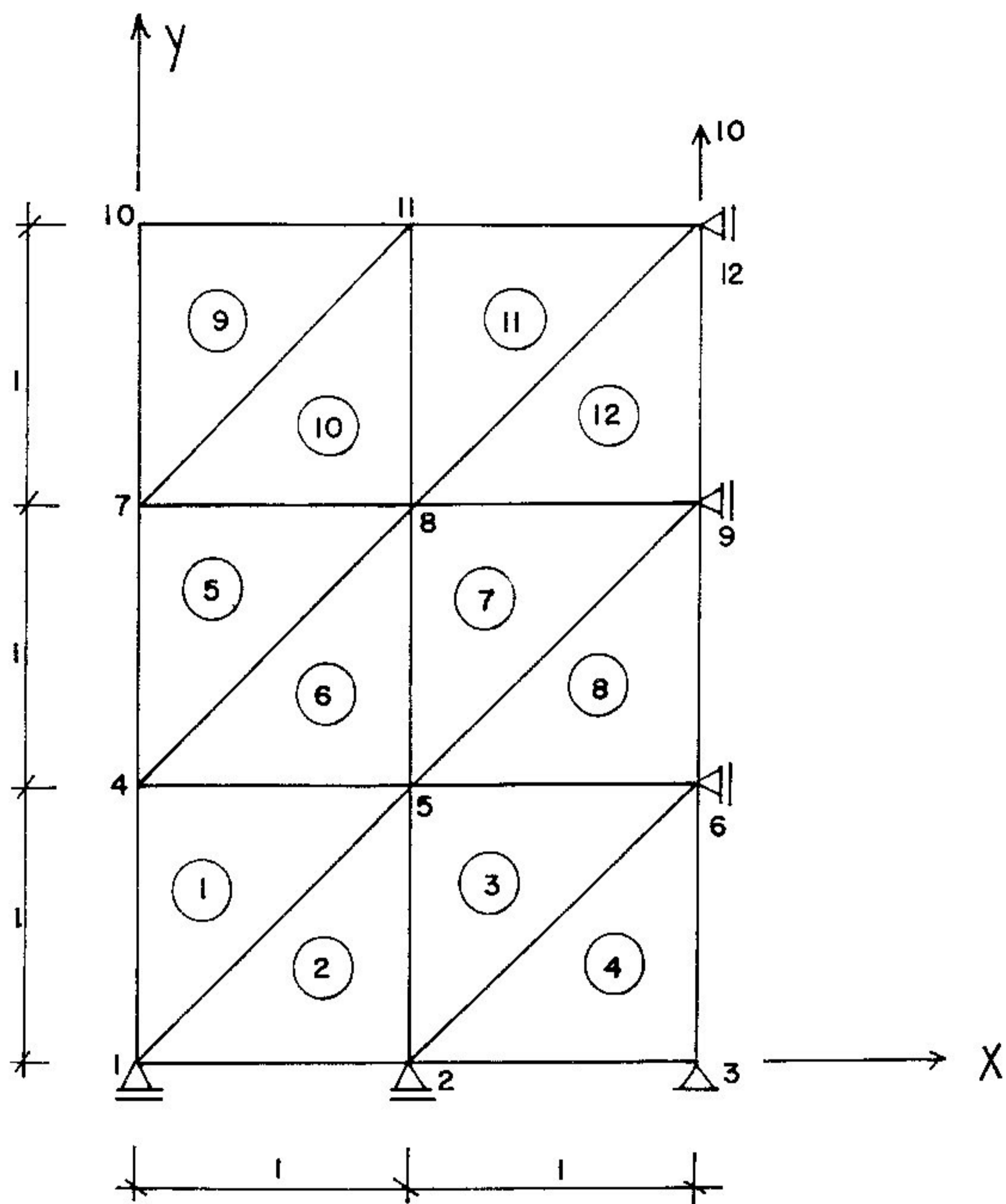
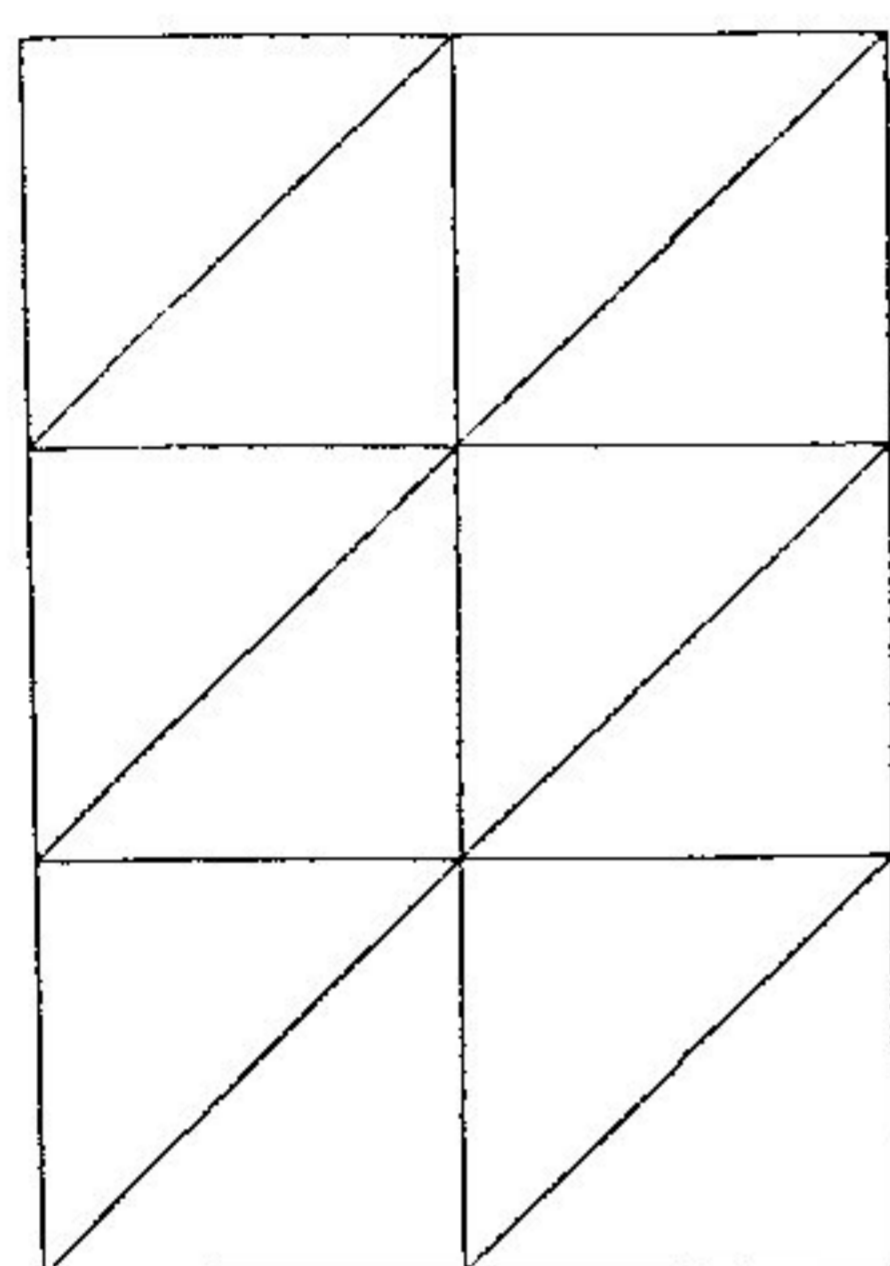


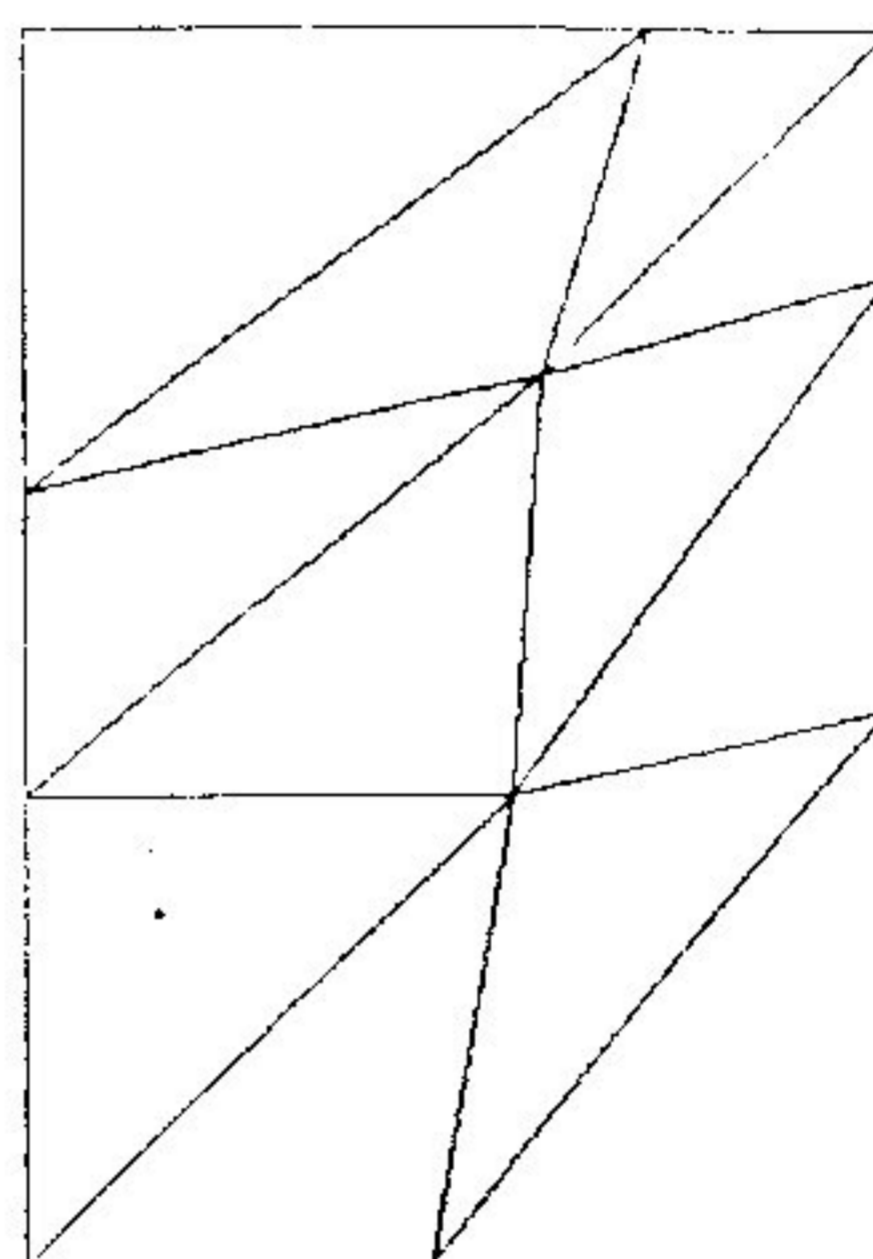
FIG. IV-13 Problema BOUSSINESQ
Malha OR-OH.

Para efeito do procedimento R os nós 1, 3, 10 e 12 foram impedidos de sofrer qualquer deslocamento, aos nós 2 e 11 foi permitido apenas o deslocamento na direção x, aos nós 4, 6, 7 e 9 foi permitido apenas o deslocamento na direção y e os nós 5 e 8 não sofrem qualquer restrição ao seu deslocamento.

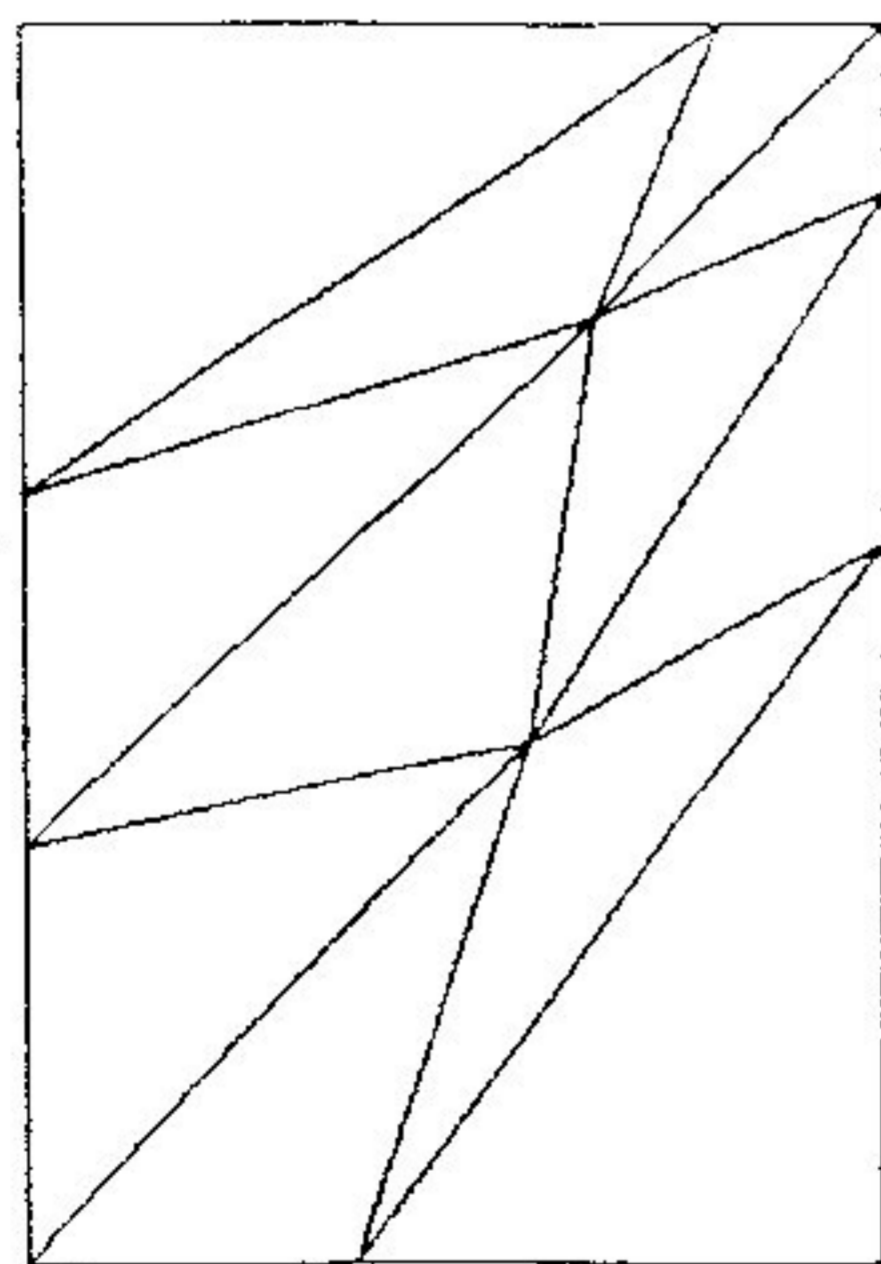
Neste problema, apenas três adaptações R foram suficientes para equidistribuir satisfatoriamente o erro no domínio. As malhas obtidas neste procedimento estão mostradas na FIG. IV-14.



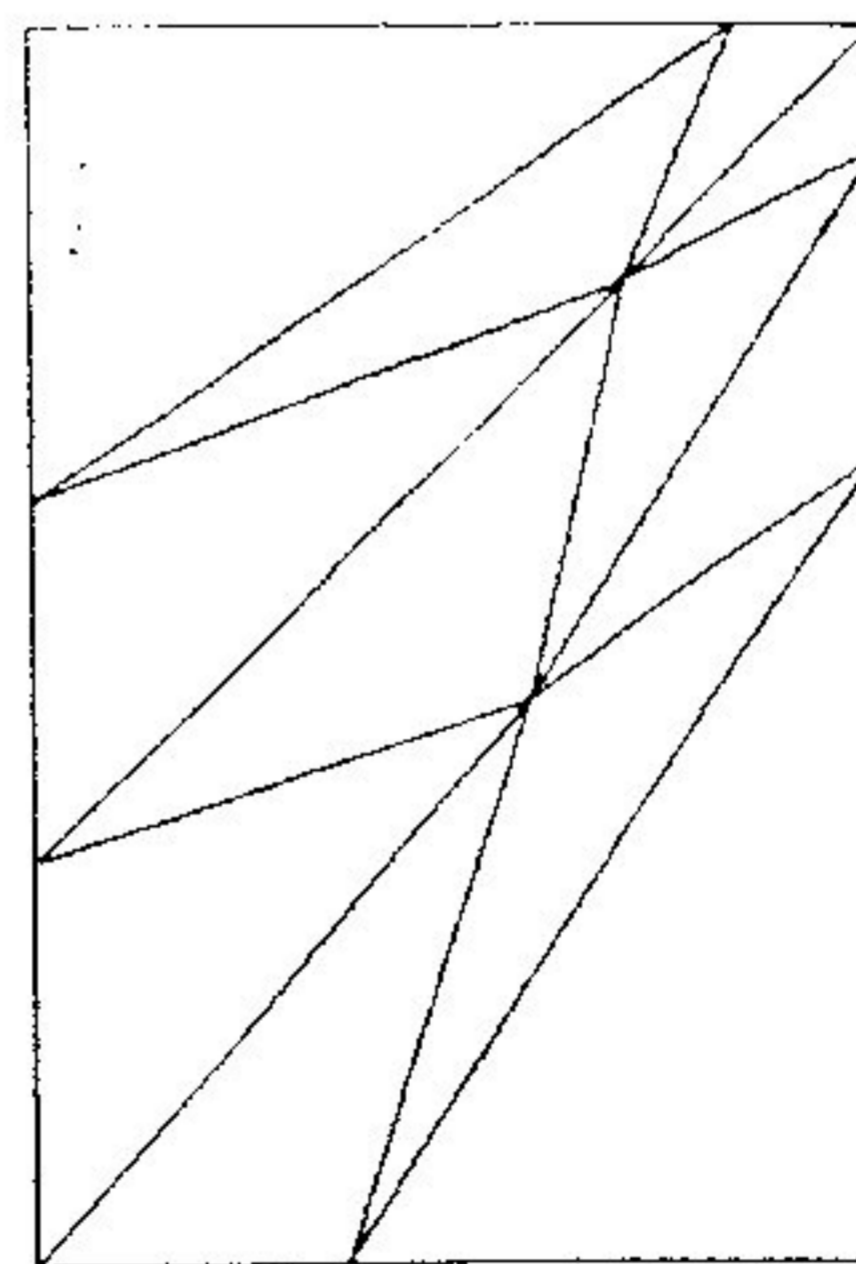
MALHA 0R-OH



MALHA 1R-OH



MALHA 2R-OH



MALHA 3R-OH

FIG. IV-14 Problema BOUSSINESQ
Sequência de malhas obtidas com três adaptações R sucessivas a partir da malha 0R-OH.

A malha obtida pelo refinamento H três vezes sucessivas a partir da malha 3R-0H tem um detalhe mostrado na FIG. IV-15, onde está refinado apenas o elemento 12 da malha 3R-0H.

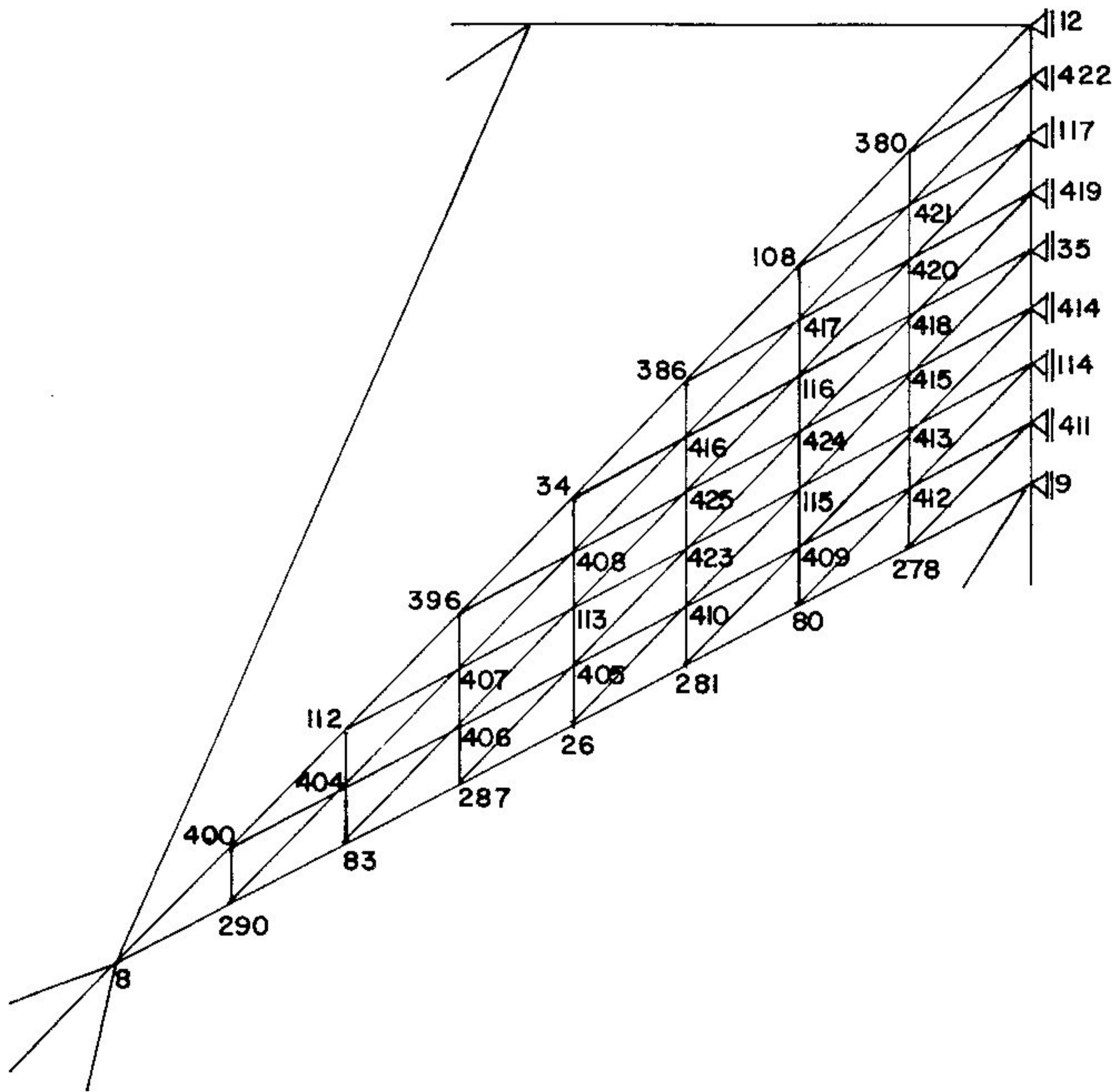


FIG. IV-15 Problema BOUSSINESQ
 Detalhe da malha 3R-3H.

O detalhe correspondente da malha OR-3H tem aspecto semelhante.

As malhas nR-3H constituem-se em problemas de 808 graus de liberdade (425 nós e 768 elementos).

O problema BOUSSINESQ foi resolvido também analiticamente, através dos resultados da Teoria da Elasticidade (Timoshenko e Goodier-1980) com objetivo de aferir os resultados obtidos com as diversas discretizações (FIG. IV-16).

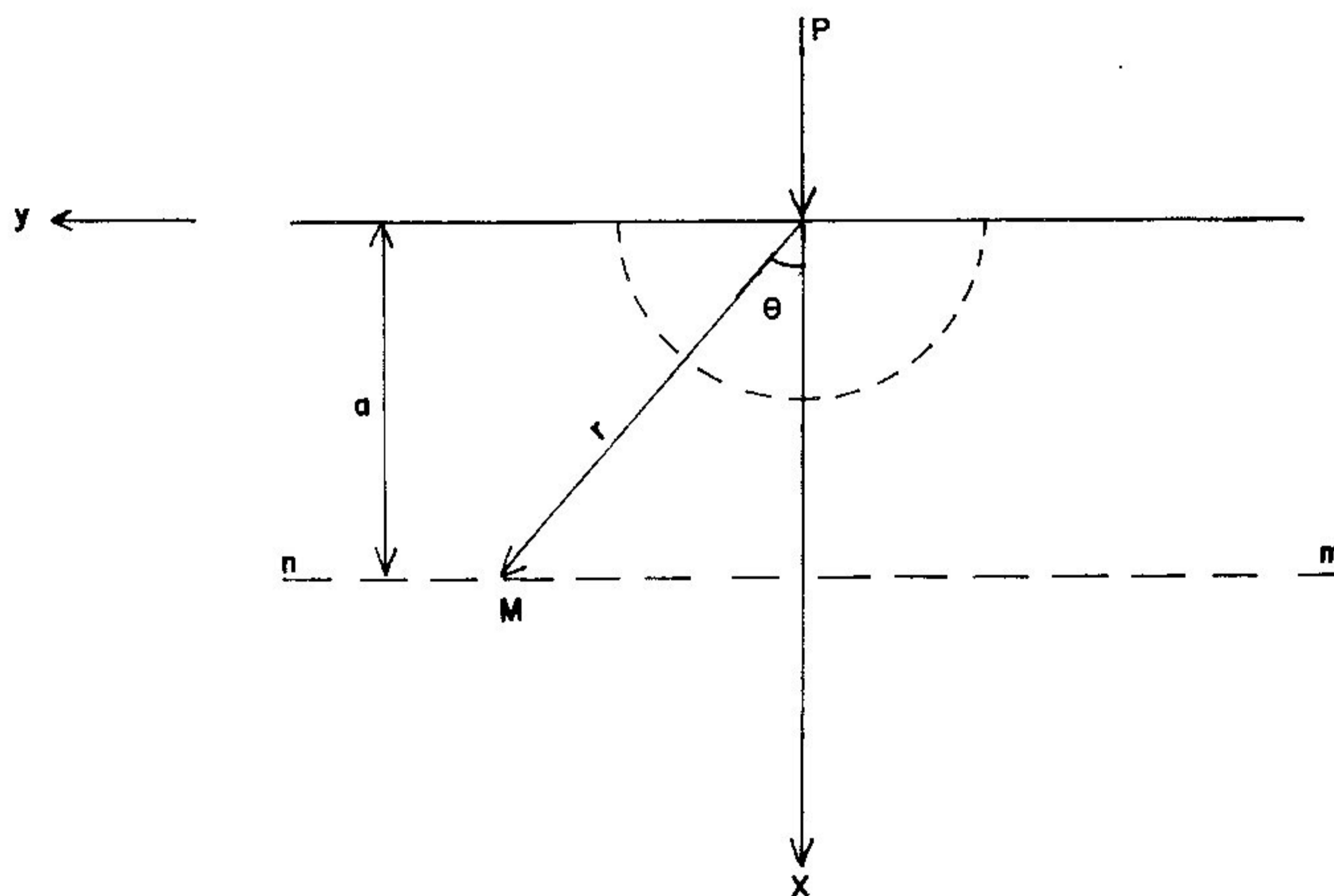


FIG. IV-16 Problema BOUSSINESQ
Solução analítica.

Referindo-se à FIG. IV-16,

$$\sigma_r = \frac{-2P}{\Pi} \frac{\cos\theta}{r} \quad (IV-1)$$

$$\sigma_\theta = \tau_{r\theta} = 0 \quad (IV-2)$$

no ponto M, situado em uma reta nm distante a da borda,

$$\sigma_x = \sigma_r \cos^2\theta = \frac{-2P}{\Pi} \frac{\cos^3\theta}{r} = \frac{-2P}{\Pi a} \cos^4\theta \quad (IV-3)$$

$$\sigma_y = \sigma_r \sin^2\theta = \frac{-2P}{\Pi a} \sin^2\theta \cos^2\theta \quad (IV-4)$$

$$\begin{aligned} \tau_{xy} &= \sigma_r \sin\theta \cos\theta = \frac{-2P}{\Pi} \frac{\sin\theta \cos^2\theta}{r} = \quad (IV-5) \\ &= \frac{-2P}{\Pi a} \sin\theta \cos^3\theta \end{aligned}$$

IV-2-2 Análise do desempenho do algoritmo MM

a) Parametrização do algoritmo MM

Foi também empregada, neste problema, a tolerância $\epsilon = 10^{-6}$ para o critério de parada do algoritmo MM, que, da mesma forma que no problema anterior, foi o maior valor que produziu resultados, em tensões, coincidentes com os do processo direto EG.

Na malha do nível 0 do MM, que corresponde à malha nR-0H, foi empregada a tolerância $\epsilon = 10^{-3}$ para o critério de parada das iterações GS, a qual apresentou, como no problema anterior, os melhores resultados em tempo de CPU.

A TAB. IV-5 apresenta a resposta do algoritmo MM à variação do número de iterações GS nos níveis 1, 2 e 3 do MM, na solução do sistema de equações da malha 3R-3H por MM.

ν	N. CICLOS MM	N. ITER. NIVEL 0	N. ITER. FINAIS M3	TEMPO CPU (SEG.)
2	38	31	2	195,530
3	27	47	3	183,880
4	22	59	4	186,910
5	19	64	5	191,750

TAB. IV-5 Problema BOUSSINESQ

Parametrização do MM - Malha 3R-3H

$$\epsilon = 10^{-6} \quad \epsilon_0 = 10^{-3}$$

Neste problema, o número de iterações GS por nível $\nu = 3$ mostrou-se mais eficiente que os outros, sendo o que gastou menos tempo de CPU.

Foi também feita uma verificação do conhecido aumento significativo do número de ciclos MM com o aumento da incompressibilidade do material (coeficiente de Poisson).

A malha 0R-3H, além de processada com coeficiente de Poisson igual a 0,3, também foi processada com coeficientes iguais a 0,45 e 0,49. Os resultados obtidos são apresentados na TAB. IV-6.

COEF. DE POISSON	N. CICL. MM	TEMPO CPU (SEG.)
0,30	8	83,870
0,45	16	161,91
0,49	50	510,54

TAB. IV-6 Problema BOUSSINESQ - Malha 0R-3H

$$\epsilon = 10^{-6} \quad \epsilon_0 = 10^{-3} \quad \nu = 5$$

Aumento do número de ciclos MM com o aumento do coeficiente de Poisson.

b) Fator irregularidade da malha

A TAB. IV-7 apresenta a resposta do algoritmo MM ao aumento da irregularidade da malha no problema BOUSSINESQ. Foram processadas as malhas 0R-3H, 1R-3H, 2R-3H e

3R-3H, obtidas pelo refinamento H das malhas apresentadas na FIG. IV-14, as quais apresentam um aumento sequencial de irregularidade.

MALHA	N. CICL. MM	TEMPO CPU (SEG.)
0R-3H	11	77,010
1R-3H	14	97,270
2R-3H	21	144,390
3R-3H	27	183,880

TAB. IV-7 Problema BOUSSINESQ

$$\varepsilon = 10^{-6} \quad \varepsilon_0 = 10^{-3} \quad \nu = 3$$

Aumento do número de ciclos MM com o aumento da irregularidade da malha.

Pode-se constatar, também neste problema, um aumento significativo do número de ciclos MM, e o conseqüente aumento do tempo de CPU gasto no processamento, com o aumento da irregularidade da malha.

Como as adaptações R tornam a malha, nestes casos, mais irregular como conseqüência da tentativa de aumentar a densidade de nós na região que apresenta concentração de tensões, foi levantada a hipótese de que o aumento do número de ciclos MM fosse devido, não à irregularidade da malha, mas a uma maior sensibilidade da malha ao carregamento. Tal hipótese foi analisada com o processamento do problema BOUSSINESQ através de uma malha (0R-3H)', proveniente de uma malha inicial (0R-0H)' muito irregular e mais refinada nas regiões de baixa concentração de tensões, portanto, com pouca sensibilidade ao carregamento (FIG.IV-17). Caso o número de ciclos MM se apresentasse com a ordem dos da TAB. IV-5, a hipótese acima estaria confirmada.

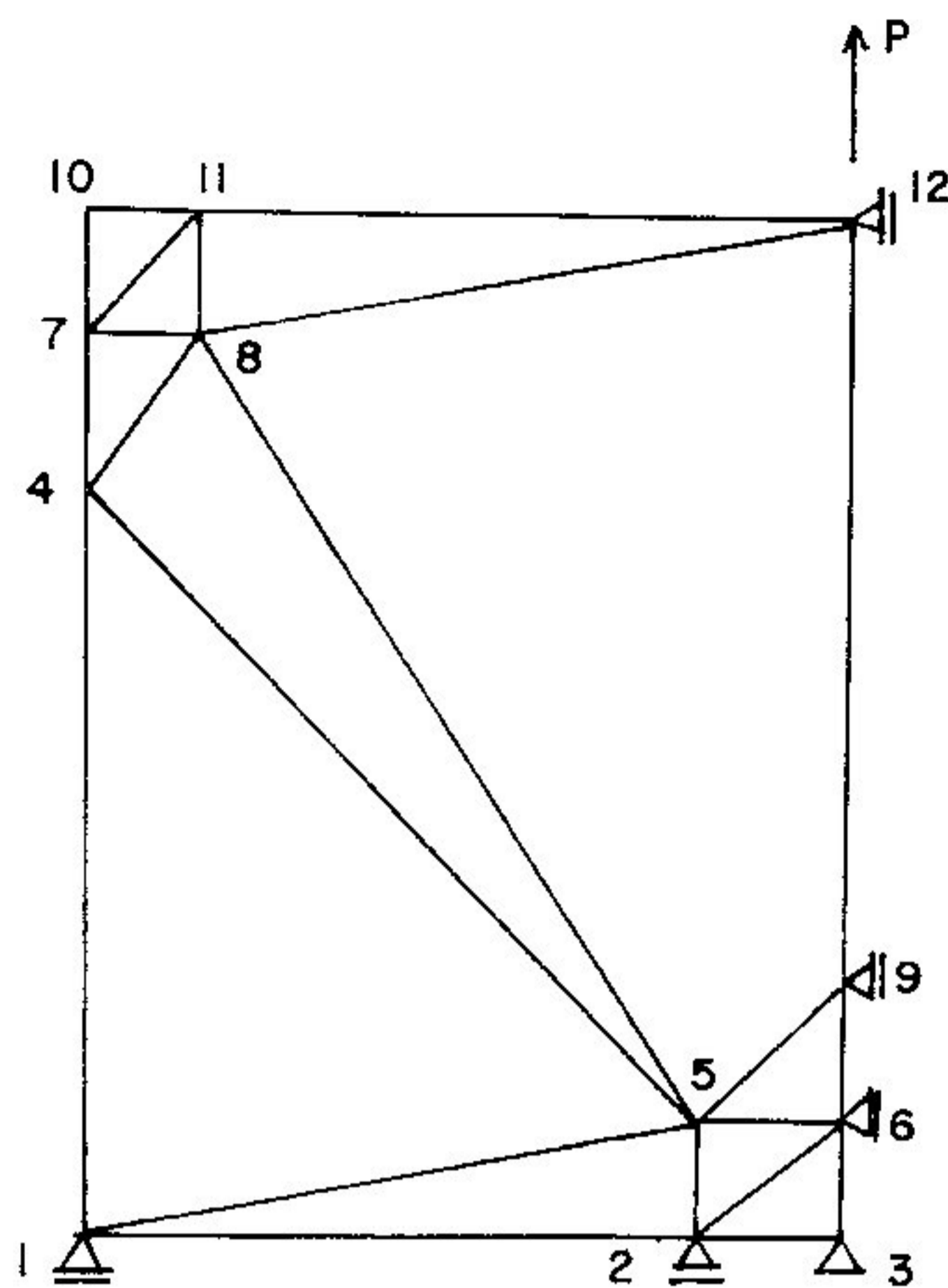


FIG. IV-17 Problema BOUSSINESQ

Malha inicial (0R-0H)' que originou a malha (0R-3H)' usada na verificação do fator irregularidade da malha.

Obteve-se, com esta malha, para $\varepsilon = 10^{-6}$
 $\varepsilon_0 = 10^{-3}$ e $\nu = 5$, os seguintes resultados:

número total de ciclos MM = 43

tempo de CPU = 424,020 seg

os quais se mostram muito piores que os obtidos com a malha mais irregular anterior, o que parece não deixar dúvidas quanto ao fato de que o aumento de irregularidade da malha, e não o aumento da sensibilidade ao carregamento, aumenta o número de ciclos MM.

c) Análise da eficiência do algoritmo MM

Os tempos de CPU gastos na solução do sistema de equações da malha 3R-3H utilizando os processos EG, GS e MM estão apresentados na TAB. IV-8.

PROCESSO	N ITER GS*	TEMPO CPU
	N CIC MM**	(SEG.)
EG	—	17,460
GS *	4357	9417,690
MM **	27	183,880

TAB. IV-8 Problema BOUSSINESQ - Malha 3R-3H
Tempo de CPU (seg.) gastos na solução do sistema de equações pelos processos EG, GS e MM ($\nu = 3$).

Verificou-se, neste problema, um aumento de 10,53 vezes no tempo de CPU gasto pelo MM em relação ao EG e de 51,22 vezes no tempo gasto pelo GS em relação ao MM. O tempo gasto pelo GS é 539,39 vezes maior que o tempo gasto pelo EG.

Confirmou-se aqui o observado no problema anterior: o MM é um processo iterativo muito mais eficiente que o GS por linhas, mas, para um problema com a dimensão do analisado, é menos eficiente que o método direto (EG).

A análise semelhante ($\nu = 3$) feita na malha mais regular (0R-3H) tem seus resultados mostrados na TAB. IV-9.

PROCESSO	N ITER GS*	TEMPO CPU
	N CIC MM**	(SEG.)
EG	—	17,460
GS *	3435	7411,750
MM **	11	77,010

TAB. IV-9 Problema BOUSSINESQ - Malha 0R-3H
Tempo de CPU gasto na solução do sistema de equações pelos processos EG, GS e MM.

Na malha 0R-3H o tempo de CPU gasto pelo MM é 4,41 vezes maior que o gasto pelo EG e o tempo gasto pelo GS é 96,24 vezes maior que o gasto pelo MM e 424,50 vezes maior

que o gasto pelo EG. Foram feitas pelo MM apenas 33 iterações GS na malha fina.

Também, neste problema, pode-se verificar um aumento maior da eficiência do MM com o aumento da regularidade da malha do que o aumento de eficiência dos outros processos nas mesmas circunstâncias. Valem aqui as mesmas suposições feitas a esse respeito na análise do problema TRINCA.

d) Comparação com resultados de Parsons e Hall

Os resultados, em termos de número de ciclos MM, obtidos para o problema BOUSSINESQ (malha OR-3H), foram comparados com os obtidos por Parsons e Hall para um problema semelhante, mostrado na FIG. IV-18.

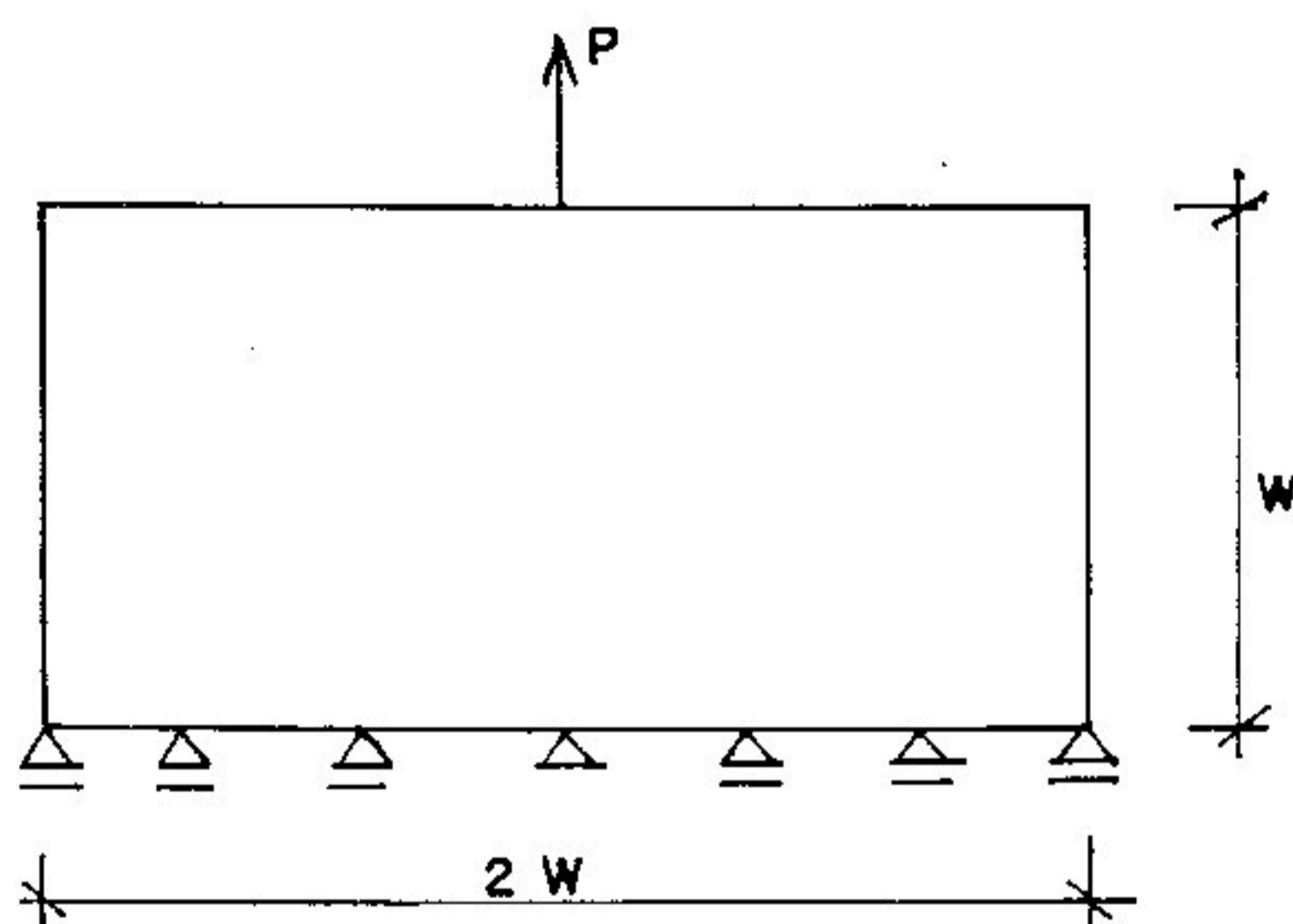


FIG. IV-18 Problema semelhante ao BOUSSINESQ
Parsons e Hall-1990.

A malha utilizada foi a mesma do problema semelhante ao TRINCA, mostrada na (FIG. IV-7), com 1122 graus de liberdade.

Tendo em vista que, apesar das duas malhas comparadas serem uniformes, o elemento utilizado por Parsons e Hall é superior ao empregado aqui, e os resultados obtidos neste trabalho podem ser considerados satisfatórios. Para $\nu = 5$ e $\epsilon = 10^{-6}$,

Problema \approx BOUSSINESQ (Parsons e Hall) \rightarrow N. de ciclos MM \approx 4
Problema BOUSSINESQ (malha OR-3H) \rightarrow N. de ciclos MM = 8

e) Evolução do erro com os ciclos MM

A FIG. IV-19 apresenta a evolução do erro, calculado após cada ciclo MM, até a convergência.

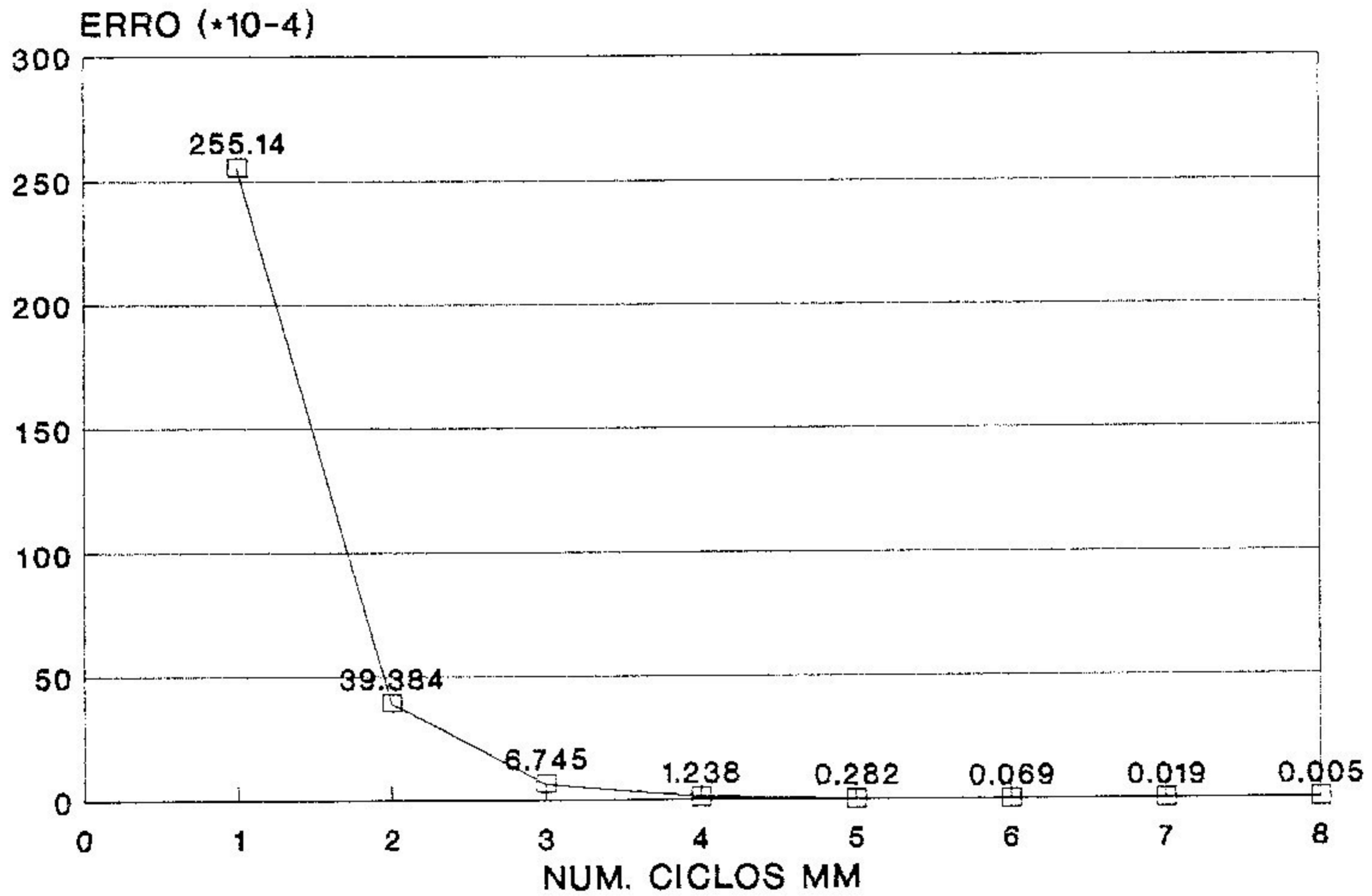


FIG. IV-19 Problema BOUSSINESQ. Malha OR-3H. $\nu=5$, $\epsilon=10^{-6}$.
Evolução do erro após cada ciclo MM.
(Número de ciclos MM = 8)

IV-2-3) Análise da eficácia do processo adaptativo R-H global

Os resultados, em tensões, obtidos com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H nas linhas de coordenadas $(x;3,0)$ e $(2,0;y)$ são mostrados nas figuras seguintes. São apresentados também os resultados obtidos pela solução analítica.

A FIG. IV-20 mostra as tensões na linha de coordenadas $(x;3,0)$ obtidas através das quatro malhas.

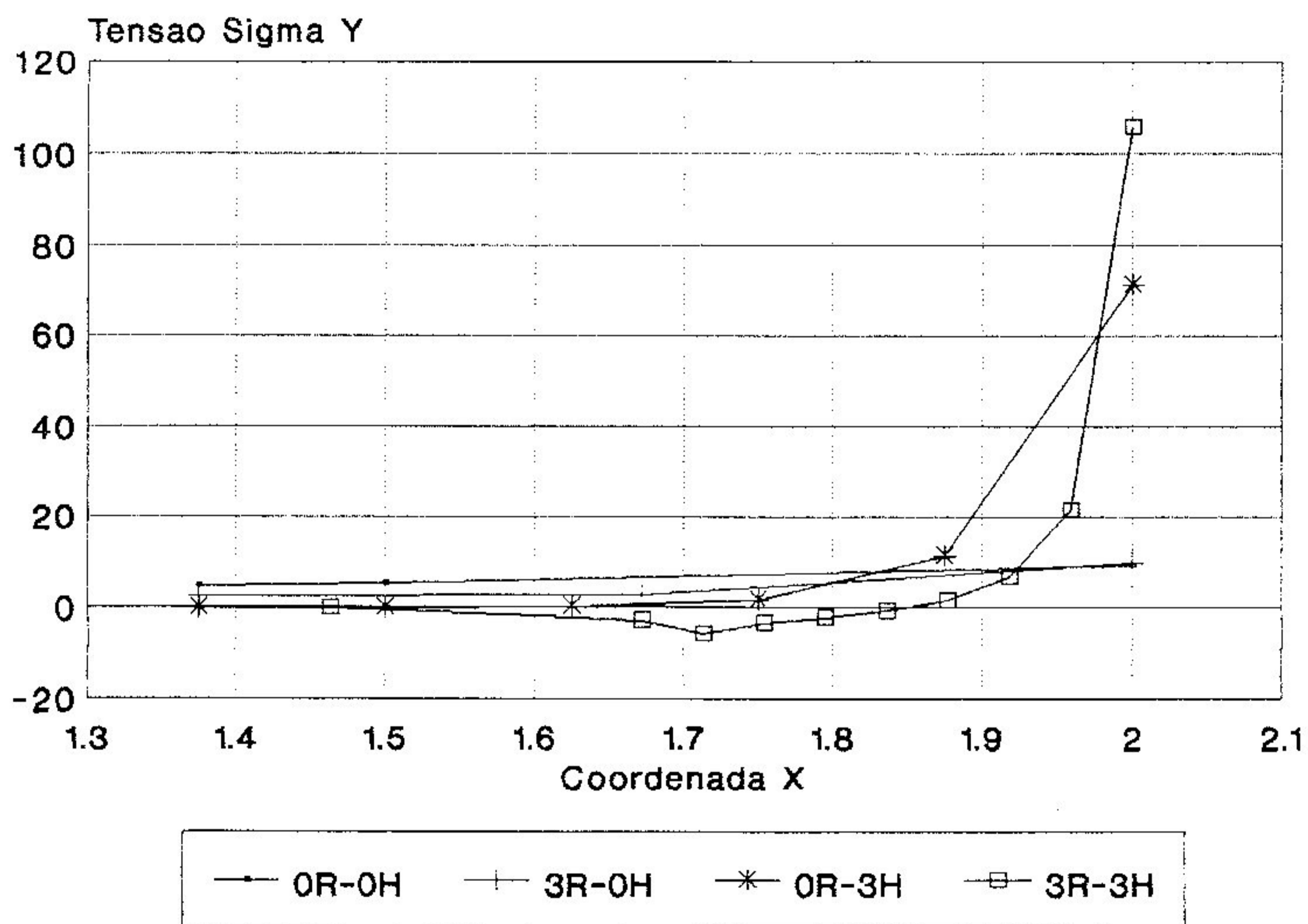


FIG. IV-20 Problema BOUSSINESQ
Tensões σ_y em $(x;3,0)$ obtidas com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H.

Pode-se perceber a pequena melhora dos resultados quando se procede às adaptações R (malha 3R-0H) na malha inicial (0R-0H) que é limitada pelo número muito pequeno de elementos da malha inicial, mas que apesar disso, quando submetida ao refinamento H (malha 3R-3H) produz uma sensível melhora nos resultados quando comparados com os da malha refinada sem a prévia adaptação R (malha 0R-3H).

A FIG. IV-21 apresenta as tensões na mesma linha ($x:3,0$) obtidas com a malha 3R-3H e pela solução analítica.

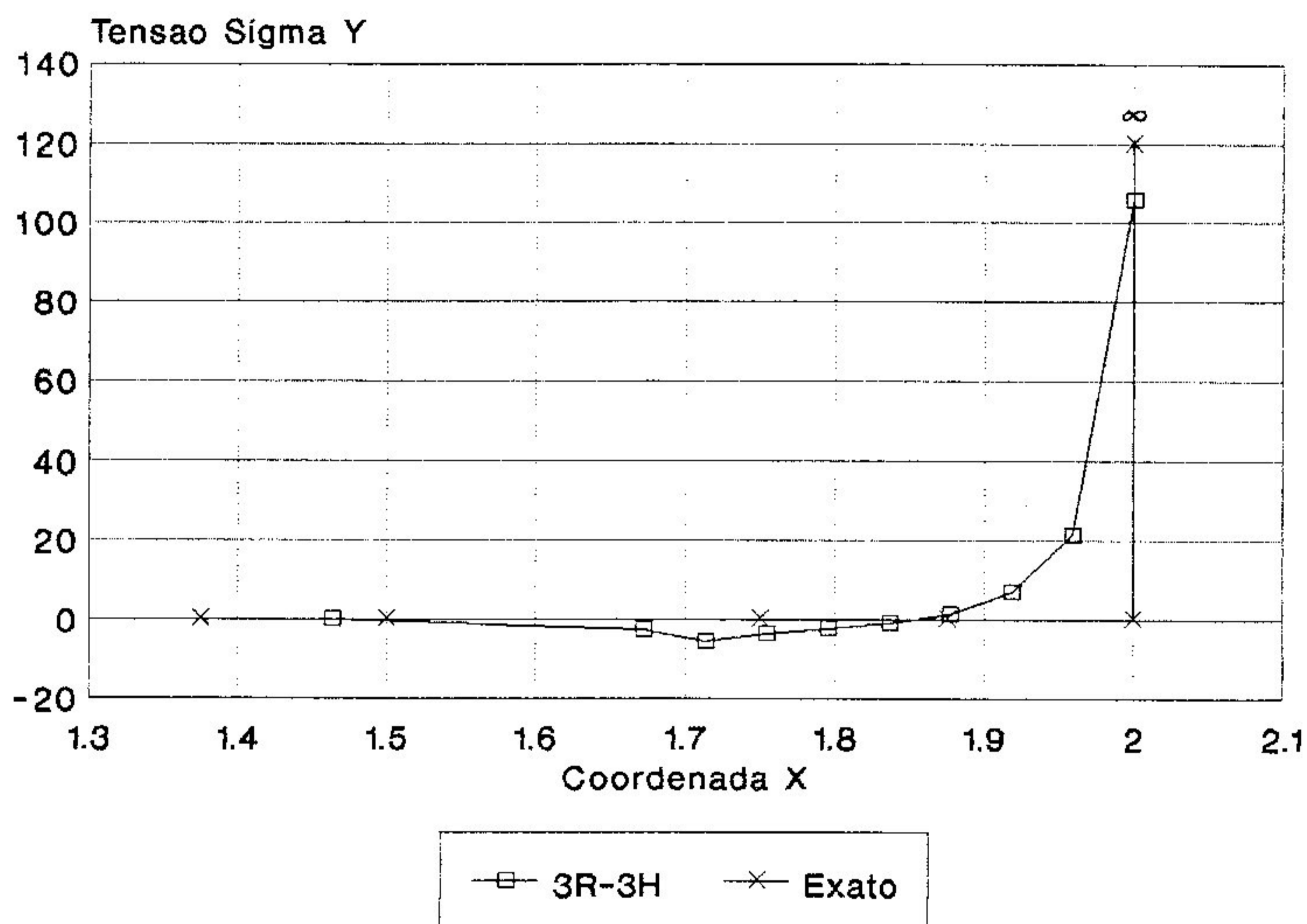


FIG. IV-21 Problema BOUSSINESQ
Tensões σ_y em ($x:3,0$) obtidas com a malha 3R-3H e com a solução analítica.

Nota-se que os resultados obtidos com a discretização aproximam-se satisfatoriamente dos exatos.

Na FIG. IV-22 são apresentadas as tensões σ_y na linha de coordenadas $(2,0;y)$ obtidas através das quatro malhas.

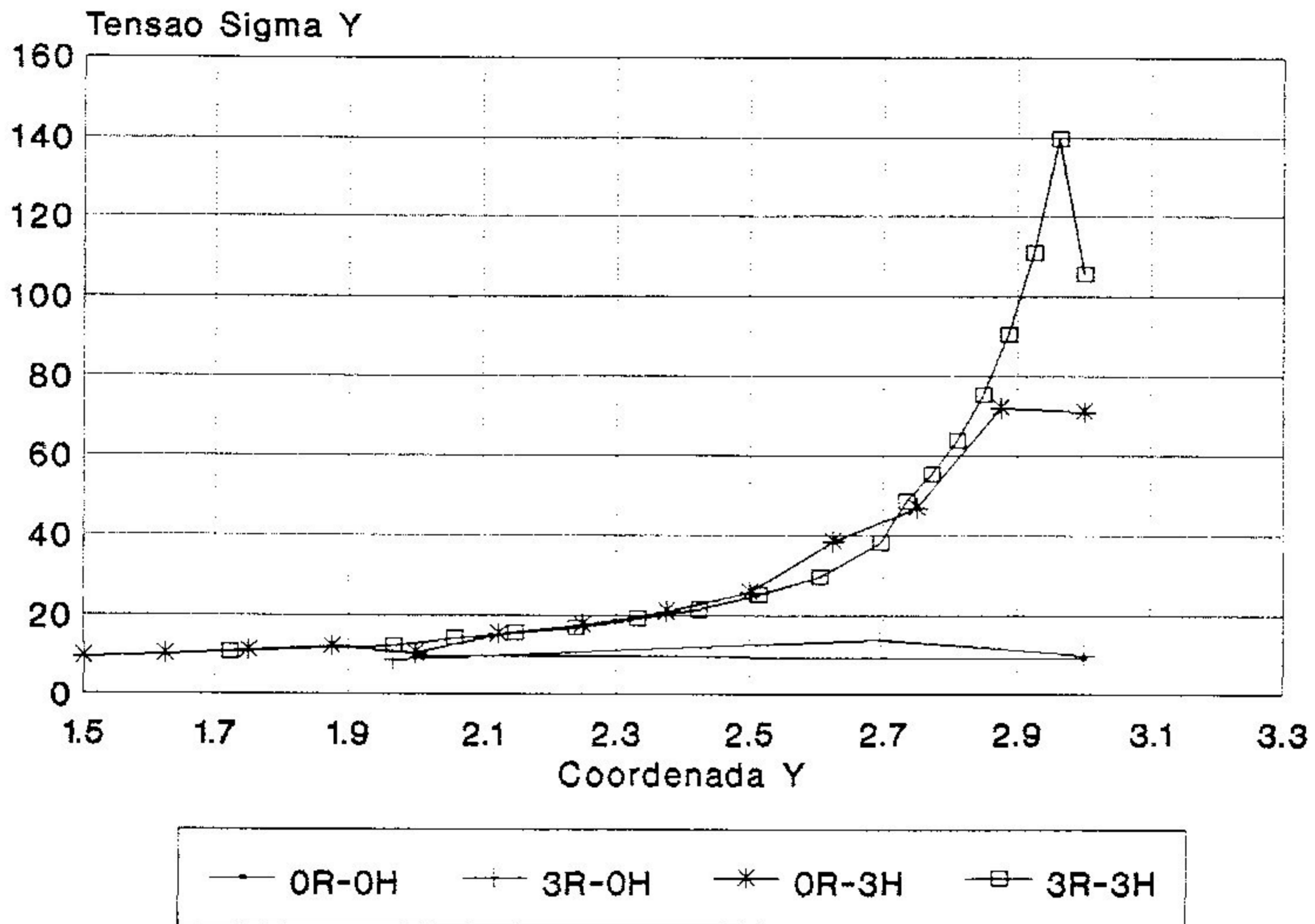


FIG. IV-22 Problema BOUSSINESQ
Tensões σ_y em $(2,0;y)$ obtidas com as malhas OR-0H, 3R-0H, OR-3H e 3R-3H.

Valem para estes resultados as observações feitas para a FIG. IV-20.

A FIG. IV-23 apresenta as tensões na mesma linha $(2,0;y)$ obtidas com a malha 3R-3H e com a solução analítica.

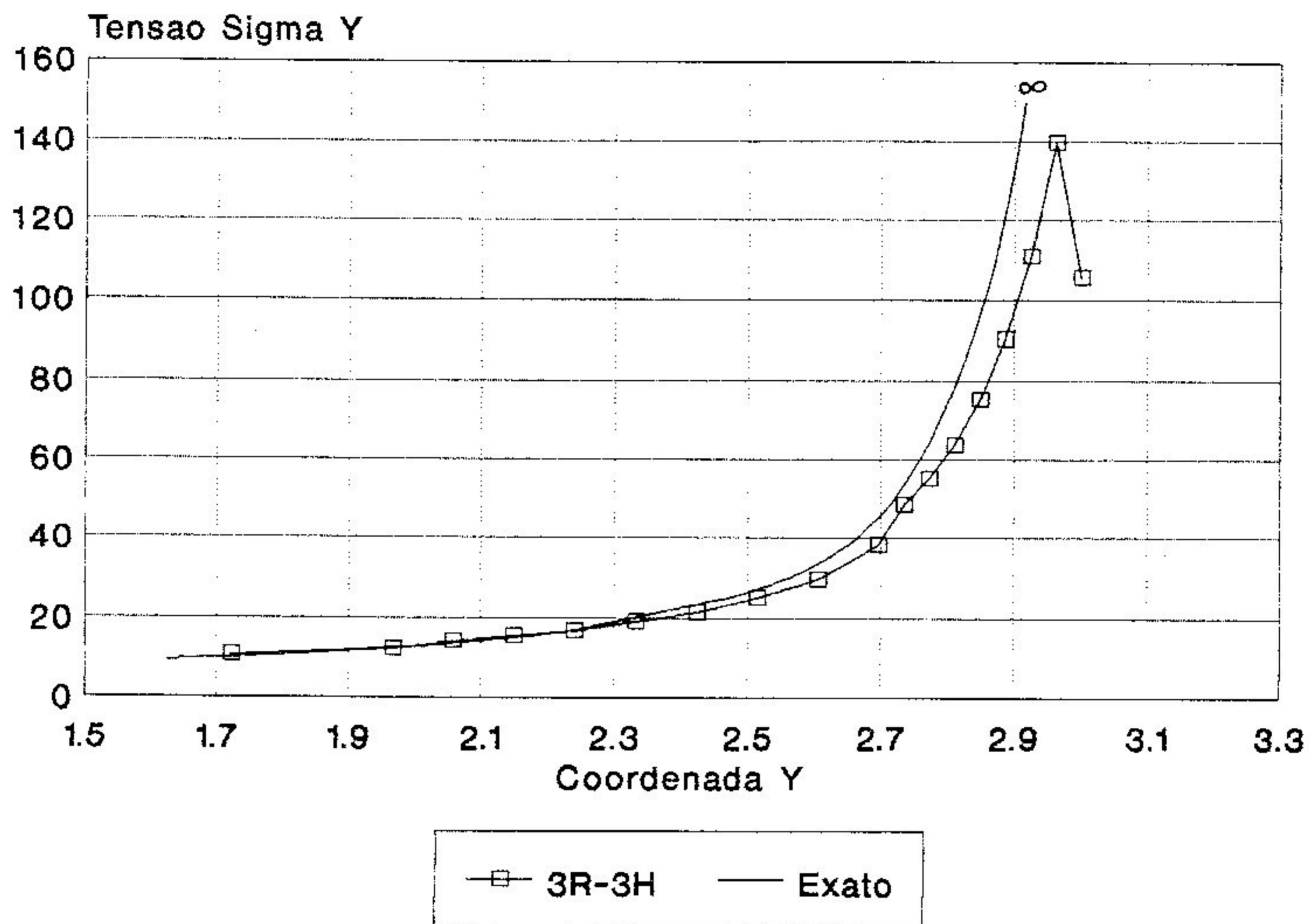


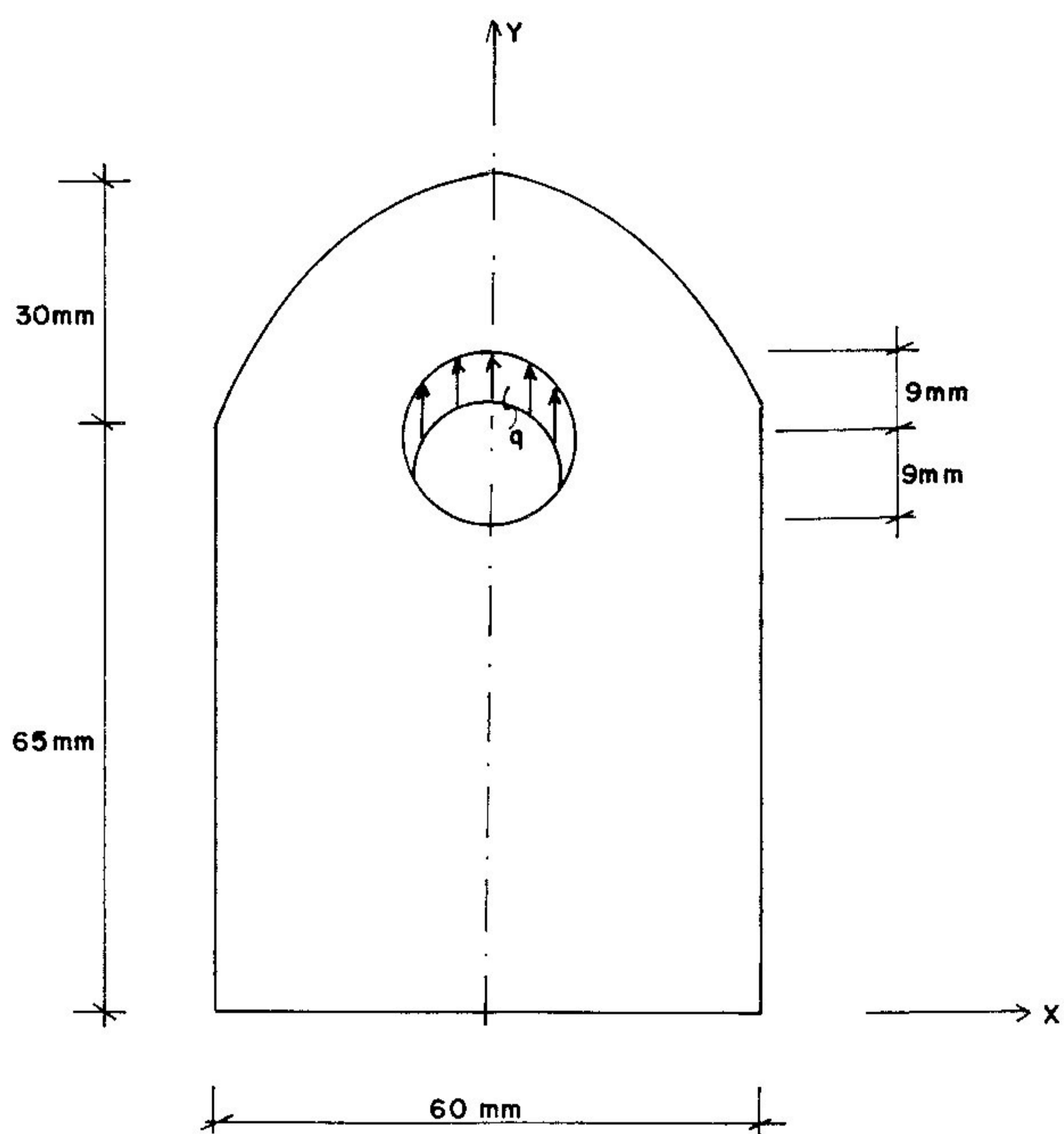
FIG. IV-23 Problema BOUSSINESQ
 Tensões σ_y em $(2.0;y)$ obtidas com a malha 3R-3H e com a solução analítica

Observa-se, também aqui, uma aproximação muito boa da solução analítica pela discretização 3R-3H. O decréscimo da tensão que aparece na coordenada correspondente ao nó 12, onde se encontra o carregamento, deve-se ao pequeno número de elementos que concorrem neste nó (dois elementos), fato que prejudicou a precisão do resultado neste ponto de grande concentração de tensões.

IV-3 Problema MANCAL

IV-3-1 Descrição do problema - Definição das malhas

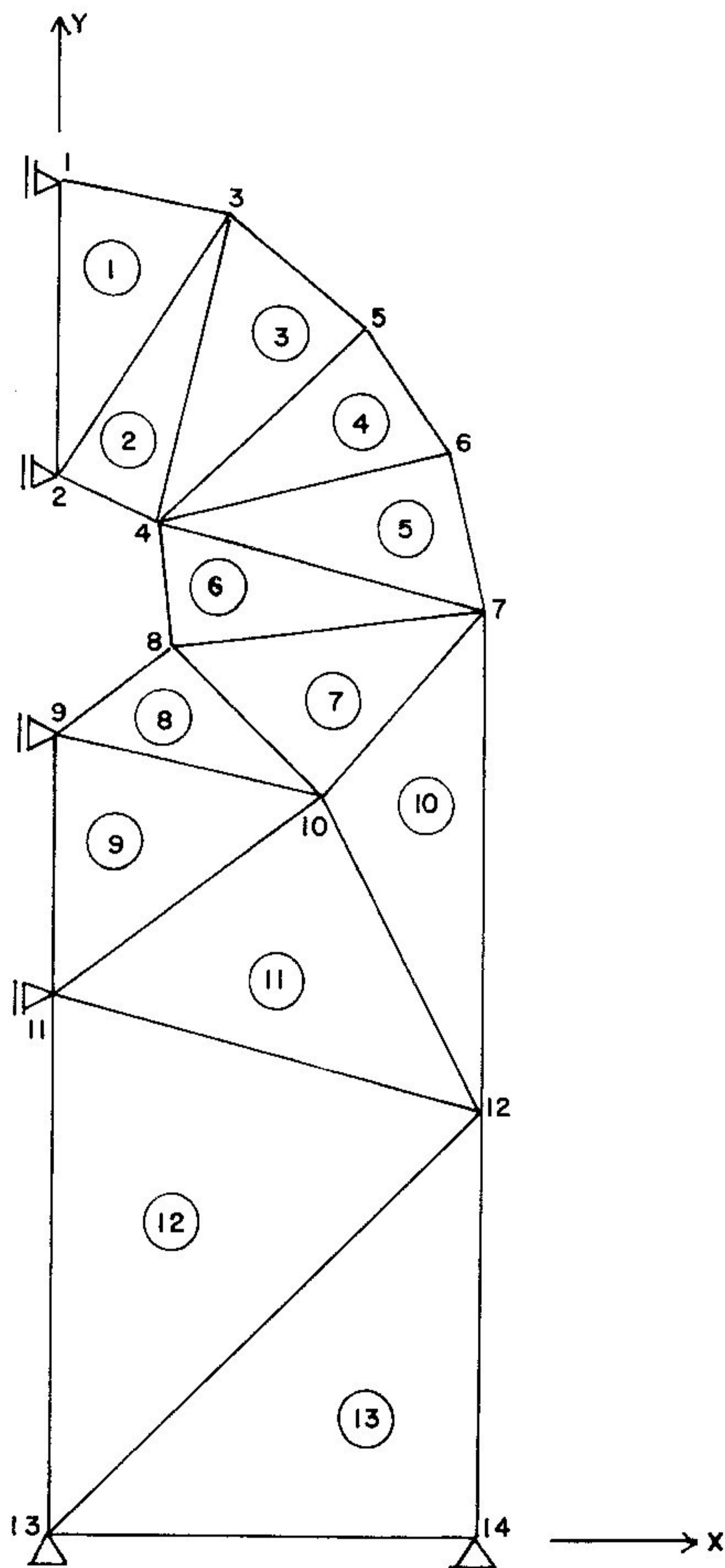
O problema MANCAL é um problema de estado plano de tensão, constituído de um mancal, ou orelha, com carregamento no furo central, mostrado na FIG. IV-24.



$$E = 2100 \text{ kgf/mm}^2 \quad \nu = 0,30 \quad e = 9 \text{ mm} \quad q \times e = 1000 \text{ kgf}$$

FIG. IV-24 Problema MANCAL
Chapa com furo

A FIG. IV-25 apresenta a malha inicial OR-OH construída para este problema. Foi discretizada a metade direita da peça.



NO	COORDENADAS	
	X	Y
1	0,0	95,0
2	0,0	74,0
3	12,0	92,5
4	7,0	71,0
5	21,5	85,0
6	27,5	76,0
7	30,0	65,0
8	8,0	62,5
9	0,0	56,0
10	19,0	52,0
11	0,0	38,0
12	30,0	30,0
13	0,0	0,0
14	30,0	0,0

CARREGAMENTO

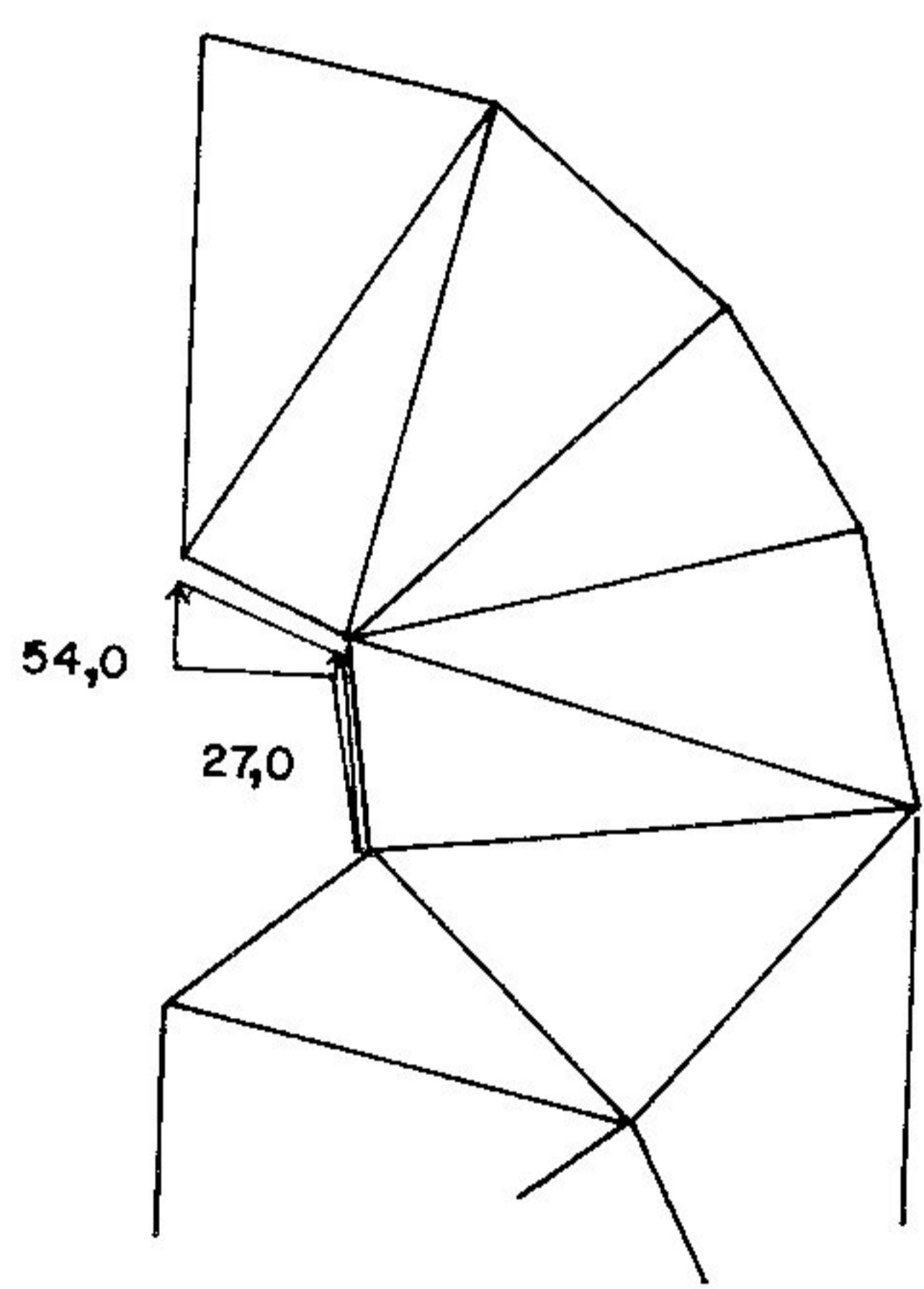


FIG. IV-25 Problema MANCAL
Malha OR-OH.

Para efeito das adaptações R, os nós 1, 2, 3, 4, 5, 6, 7, 8, 9, 13 e 14 foram impedidos de sofrer qualquer deslocamento. Aos nós 11 e 12 foi permitido o deslocamento na direção y e ao nó 10 foi permitido o deslocamento em qualquer direção. Observa-se que a pobreza da malha exigiu muitas restrições aos deslocamentos dos nós, sob pena de se descaracterizar o contorno da peça caso os nós 1 a 9 sofressem deslocamentos.

Apenas três adaptações R foram feitas na malha inicial e observa-se que o procedimento mudou muito pouco a discretização devido a tantas restrições aos deslocamentos dos nós impostas pela pobreza da malha inicial (FIG. IV-26).

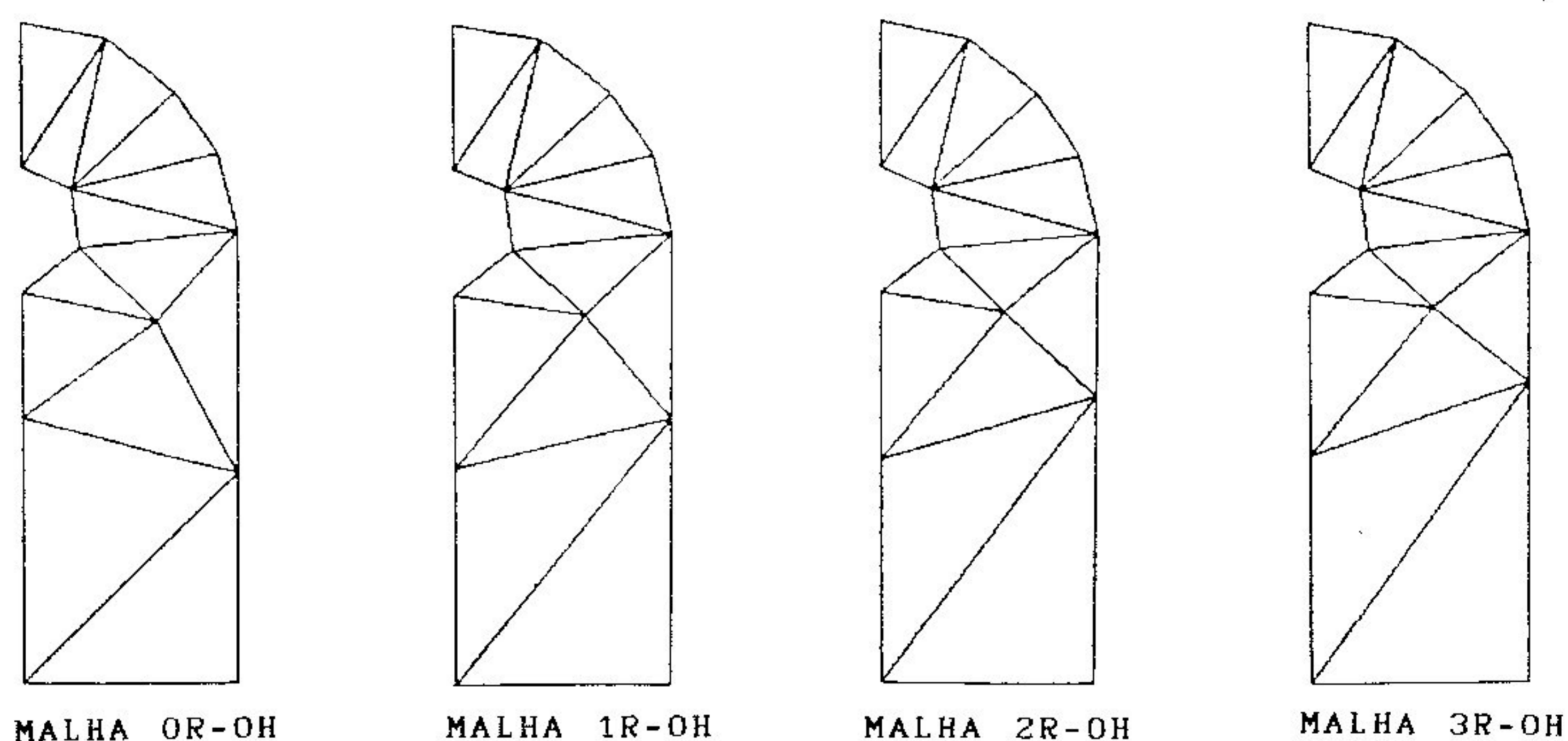


FIG. IV-26 Problema MANCAL
Sequência de malhas obtidas com três adaptações R sucessivas a partir da malha 0R-OH.

O refinamento H aplicado três vezes sucessivas na malha 3R-0H produziu a malha 3R-3H que tem um detalhe mostrado na FIG. IV-27, onde aparecem apenas os elementos originados do refinamento do elemento 1 da malha 3R-0H.

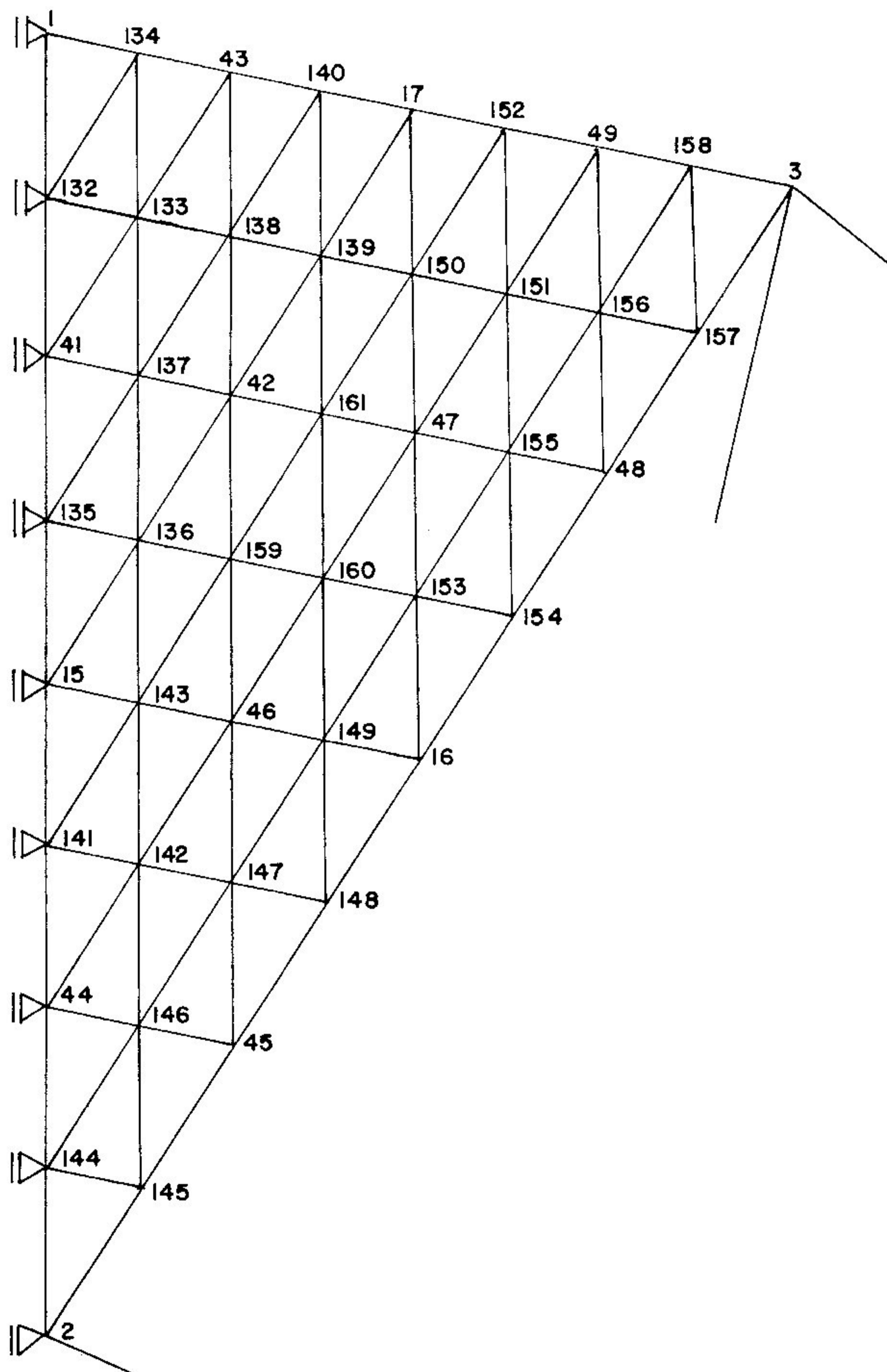


FIG. IV-27 Problema MANCAL
 Detalhe da malha 3R-3H.

O detalhe correspondente da malha 0R-3H é igual ao da malha 3R-3H, neste problema, pois os três nós 1,2 e 3 foram impedidos de sofrerem deslocamentos R.

As malhas nR-3H constituem-se em problemas de 895 graus de liberdade (469 nós e 832 elementos).

Não foi encontrada solução exata para este problema e as comparações se reduzem às feitas entre as quatro malhas, 0R-0H, 3R-0H, 0R-3H e 3R-3H, entre si.

IV-3-2 Análise do desempenho do algoritmo MM

a) Parametrização do algoritmo MM

Os resultados das análises análogas às feitas para os dois problemas anteriores são apresentados a seguir.

Empregados $\epsilon = 10^{-6}$ e $\epsilon_0 = 10^{-3}$, a TAB. IV-10 apresenta a resposta do algoritmo MM à variação do número de iterações GS nos níveis 1, 2 e 3, na solução do sistema de equações da malha 3R-3H por MM.

ν	N. CICLOS MM	N. ITER NIVEL 0	N. ITER FINAIS M3	TEMPO CPU (SEG.)
4	17	66	4	94,190
5	14	67	5	91,830
8	11	71	8	107,260

TAB. IV-10 Problema MANCAL

Parametrização do MM - Malha 3R-3H

$$\epsilon = 10^{-6} \quad \epsilon_0 = 10^{-3}$$

Aqui o valor de $\nu = 5$ apresentou melhores resultados que os obtidos com $\nu = 4$ e $\nu = 8$, levando a um menor tempo de CPU para o processamento.

b) Fator irregularidade da malha

Como as adaptações R alteraram muito pouco a malha inicial 0R-0H, as malhas resultantes são quase nada mais irregulares que a malha inicial, fato que originou a

quase coincidência das respostas obtidas para as malhas 0R-3H, 1R-3H, 2R-3H e 3R-3H, mostrados na TAB. IV-11.

MALHA	N. CICLOS MM	TEMPO CPU (SEG.)
0R-3H	14	91,770
1R-3H	14	91,770
2R-3H	14	91,830
3R-3H	14	91,830

TAB. IV-11 Problema MANCAL

Fator irregularidade da malha
 $\epsilon = 10^{-6}$ $\epsilon_0 = 10^{-3}$ $\nu = 5$

Este, portanto, não é um problema adequado para se analisar o fator irregularidade da malha.

c) Análise da eficiência do algoritmo MM

Os tempos de CPU gastos na solução do sistema de equações da malha 3R-3H, utilizando EG, GS e MM estão apresentados na TAB. IV-12.

PROCESSO	N ITER GS* N CIC MM**	TEMPO CPU (SEG.)
EG	—	5,330
GS *	14187	17949,780
MM **	14	91,830

TAB. IV-12 Problema MANCAL - Malha 3R-3H

Tempo de CPU (seg.) gasto na solução do sistema de equações pelos processos EG, GS e MM ($\nu = 5$).

Pela TAB. IV-12 pode-se observar que o tempo de CPU gasto pelo MM é 17,23 vezes maior que o tempo gasto pelo EG e que o tempo gasto pelo GS é 195,47 vezes maior que o gasto pelo MM e 3367,69 vezes maior que o gasto pelo EG.

O processamento da malha 0R-3H utilizando-se os três processos EG, GS e MM apresenta, como era de se esperar, resultados muito pouco diferentes dos da malha 3R-3H. A TAB. IV-13 apresenta estes resultados.

PROCESSO	N ITER GS* N CIC MM**	TEMPO CPU (SEG.)
EG	—	5,360
GS *	13751	17398,140
MM **	14	91,770

TAB. IV-13 Problema MANCAL - Malha 0R-3H
Tempo de CPU (seg.) gasto na solução do sistema de equações pelos processos EG, GS e MM.

IV-3-3 Análise da eficácia do processo adaptativo R-H global

Os resultados, em tensões, obtidos com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H, na linha que une os nós 8 e 7 estão mostrados na FIG. IV-28.

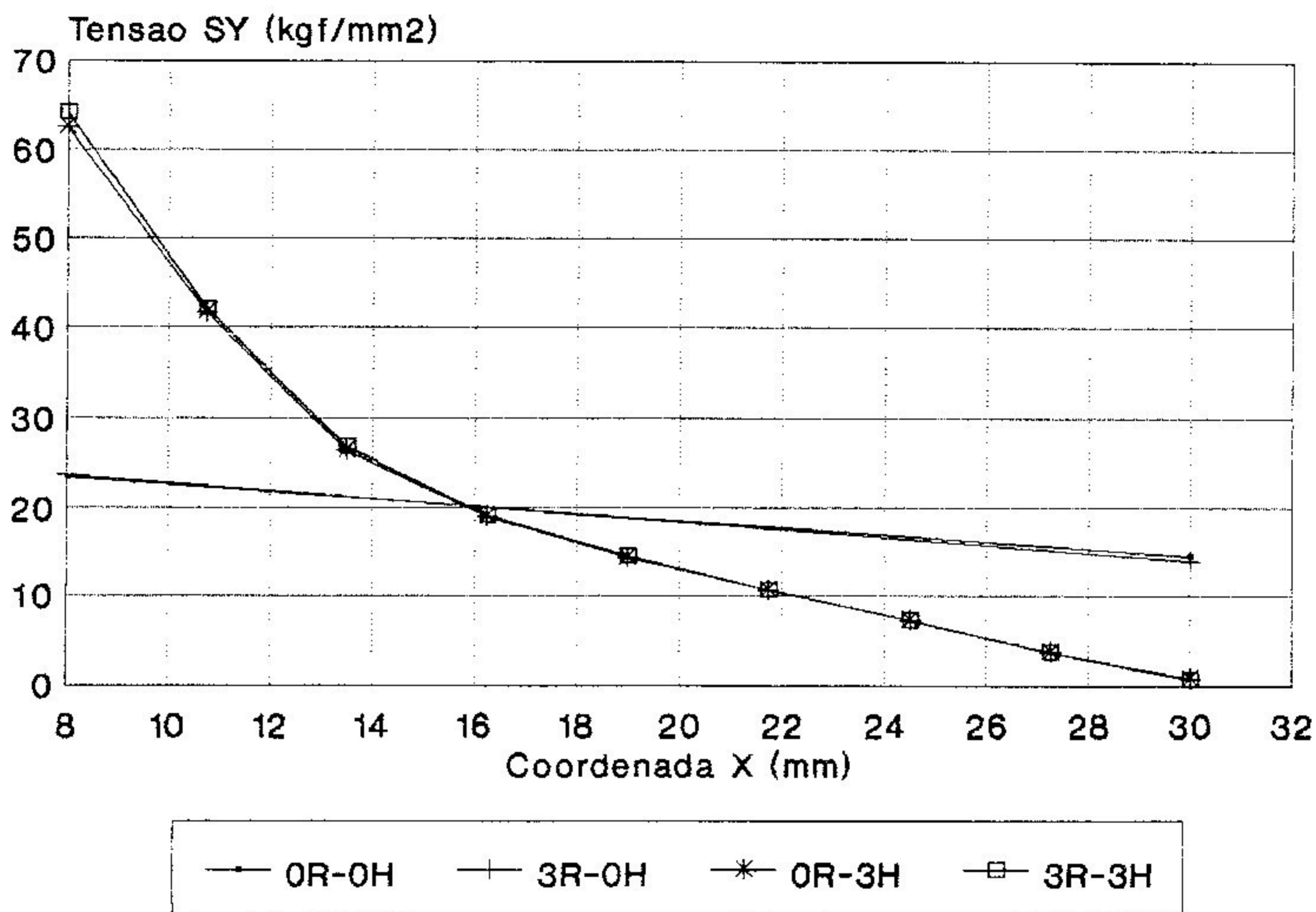


FIG. IV-28 Problema MANCAL

Tensões σ_y na linha que une os nós 8 e 7 obtidas com as malhas 0R-0H, 3R-0H, 0R-3H e 3R-3H.

Pelos motivos já expostos, nota-se a grande coincidência das curvas obtidas com as malhas 0R-0H e 0R-3H com as obtidas com as malhas 3R-0H e 3R-3H respectivamente. Reafirma-se novamente aqui o fato deste problema não ter se mostrado adequado à análise da eficácia do procedimento R aplicado na malha previamente ao procedimento H; justifica-se sua inclusão neste capítulo pela análise do MM. Nota-se, portanto, na FIG. IV-27, uma grande melhora dos resultados, que pode ser atribuída praticamente apenas ao refinamento H.

CAPÍTULO V

CONCLUSÕES E SUGESTÕES

Este trabalho teve o propósito de analisar a validade e a eficiência de uma implementação de um processo adaptativo R-H global conjugado com método multimalha de solução de sistema de equações.

Quanto à validade, esta ficou claramente comprovada, conforme mostrou a análise dos problemas resolvidos (Cap. IV). Os resultados em tensões mostraram-se bastante satisfatórios. Mostrou-se, com eles, a eficácia tanto do processo adaptativo R-H global quanto do método multimalha e da conjugação dos dois.

Sobre a eficiência tanto da conjugação do processo R-H global e método multimalha quanto do algoritmo multimalha isoladamente, pode-se fazer várias observações, sempre restritas aos problemas analisados, e algumas questões permaneceram sem resposta.

Considerando o algoritmo multimalha (MM) isoladamente pôde-se observar que:

a) Este mostrou-se muito mais eficiente que o outro processo iterativo utilizado - Gauss-Seidel por linhas (GS) - , ao mesmo tempo que muito menos eficiente que o processo direto de Eliminação de Gauss (EG). A ineficiência em relação ao EG pode ter sido causada pela limitação que o equipamento utilizado impôs à dimensão dos problemas analisados (é fato conhecido que os processos iterativos são tão mais competitivos com os diretos quanto maior for a dimensão do sistema).

O tempo de CPU fornecido para o MM inclui o tempo de operações de recuperação, gravação, remoção e rearranjo de tabelas de dados entre a memória principal e o disco. Estas operações podem ter tido bastante influência no

tempo total, uma vez que são feitas em grande número durante o processamento (bloco funcional BFMULG do processador MULGR). Estas operações foram necessárias devido à pequena capacidade de memória principal do equipamento empregado (640 kbytes), e ao fato de se ter aproveitado a estrutura do SDP, com armazenamento da matriz global. Observa-se aqui que no processador GAUSDL não foram necessárias estas operações. O espaço de armazenamento da matriz de rigidez pode ser otimizado, melhorando-se o algoritmo de renumeração dos graus de liberdade.

A escolha do método GS para a relaxação nos quatro níveis de malha do MM pode ter tido influência na sua eficiência. Sabe-se que o processo GS por linhas é pouco eficiente quando comparado com outros processos iterativos como SOR, gradiente conjugado, ou mesmo outras formas de algoritmo Gauss-Seidel.

Sugere-se a implementação de alguns destes outros processos iterativos no algoritmo MM, o que poderia, em princípio, melhorar a eficiência global do MM. Sugere-se, também, a implementação do procedimento em um computador com maior capacidade de memória principal. Além disto, sugere-se a implementação de algoritmos MM com diferentes números de níveis de malha e com um processo direto de solução de sistemas no nível 0.

b) O problema TRINCA apresenta uma deformação de flexão devida à presença da trinca, que é de difícil representação pelos elementos triangulares de três nós (fenômeno de "locking"). A este fato foi atribuído o número maior de ciclos MM do que os necessários para os outros dois problemas, principalmente o problema BOUSSINESQ.

c) Os resultados em termos de número de ciclos MM obtidos para dois dos problemas analisados foram comparados com resultados obtidos por Parsons e Hall para dois problemas semelhantes aos problemas TRINCA e BOUSSINESQ.

Considerando que o elemento utilizado por

Parsons e Hall-1990, retangular de quatro nós, é superior ao triangular de três nós aqui empregado, e que a malha irregular do problema similar ao TRINCA ali é bem menos irregular que a mais regular empregada aqui, pode-se considerar os resultados obtidos aqui como satisfatórios.

Sugere-se a implementação de elementos de ordem mais alta.

d) O algoritmo MM implantado mostrou-se muito sensível à irregularidade da malha, o que pode ser comprovado nos problemas TRINCA e BOUSSINESQ.

e) Não foi analisada, para o algoritmo implementado, a independência do número de ciclos MM em relação à variação da dimensão (número de graus de liberdade) da discretização de um problema. Sugere-se analisar esta questão.

f) O aumento da irregularidade da malha parece provocar também um aumento no número de iterações GS (ν) que produz maior eficiência do algoritmo. Sugere-se fazer esta análise.

g) A transferência dos resíduos de malhas finas para grossas foi, em uma fase inicial, feita através do operador de injeção. Observou-se que a solução convergia com um número muito grande de ciclos MM (acima de 100 em média). Com a mudança do operador para o de ponderação, conseguiu-se resultados muito bons (Cap. III e Cap. IV).

h) A variação do número de iterações GS por nível MM (ν), nos problemas analisados, não influencia significativamente o desempenho do algoritmo.

Considerando-se a conjugação do processo R-H global com método multimalha, não se pode fazer observações

conclusivas sobre a eficiência do procedimento, quando comparado com outros procedimentos adaptativos, a partir somente das análises feitas neste trabalho.

Seria necessário, para isto, a comparação de resultados do procedimento implementado e outros, tais como o processo H seletivo com sistema de equações resolvido por processos iterativos comuns e por MM.

A adaptação R tende, na maioria das vezes, a tornar a malha mais irregular, o que, como visto, piora o desempenho do algoritmo MM. O mesmo se dá com a solução por GS. Portanto, a utilização de um outro processo de solução de sistemas, ou uma outra variante do MM, pode mostrar-se mais eficiente.

Estas questões permaneceram sem resposta e demandam uma continuação do trabalho. Apesar disto, pode-se afirmar, tendo em vista os resultados obtidos, que o procedimento aqui estudado constitui-se em uma ferramenta válida para a adequação e a solução de malhas de elementos finitos.

APÊNDICE 1

MATRIZES DE RIGIDEZ DE ELEMENTOS TRIANGULARES SEMELHANTES

A matriz de rigidez de um elemento finito, calculada pela equação

$$\tilde{K}^e = \int_v \tilde{B}_e^T \tilde{D} \tilde{B}_e \, dV \quad (A1.1)$$

onde $dV = dx \, dy \, dz$ em coordenadas cartesianas e $dV = \det J \, dr \, ds \, dt$ em coordenadas naturais (elementos isoparamétricos), no caso de elementos triangulares de três nós para estado plano tem a forma explícita mostrada abaixo (Rockey, Evans, Griffiths e Nethercot-1975)

$$[K^e] = \frac{1}{4\Delta}$$

$d_{11}(y_2 - y_3)^2 + d_{33}(x_3 - x_2)^2$	$d_{12}(x_3 - x_2)(y_2 - y_3) + d_{33}(x_3 - x_2)(y_2 - y_3)$	$d_{11}(y_2 - y_3)(y_3 - y_1) + d_{33}(x_3 - x_2)(x_1 - x_3)$	$d_{12}(x_1 - x_3)(y_2 - y_3) + d_{33}(x_3 - x_2)(y_3 - y_1)$	$d_{11}(y_1 - y_2)(y_2 - y_3) + d_{33}(x_2 - x_1)(x_3 - x_2)$	$d_{12}(x_2 - x_1)(y_2 - y_3) + d_{33}(x_3 - x_2)(y_1 - y_2)$
$d_{21}(x_3 - x_2)(y_2 - y_3) + d_{33}(x_3 - x_2)(y_2 - y_3)$	$d_{22}(x_3 - x_2)^2 + d_{33}(y_2 - y_3)^2$	$d_{12}(y_3 - y_1)(x_3 - x_2) + d_{33}(x_1 - x_3)(y_2 - y_3)$	$d_{22}(x_3 - x_2)(x_1 - x_3) + d_{33}(y_2 - y_3)(y_3 - y_1)$	$d_{21}(x_3 - x_2)(y_1 - y_2) + d_{33}(x_2 - x_1)(y_2 - y_3)$	$d_{22}(x_2 - x_1)(x_3 - x_2) + d_{33}(y_1 - y_2)(y_2 - y_3)$
$d_{11}(y_2 - y_3)(y_3 - y_1) + d_{33}(x_1 - x_3)(x_3 - x_2)$	$d_{12}(x_3 - x_2)(y_3 - y_1) + d_{33}(x_1 - x_3)(y_2 - y_3)$	$d_{11}(y_3 - y_1)^2 + d_{33}(x_1 - x_3)^2$	$d_{12}(x_1 - x_3)(y_3 - y_1) + d_{33}(x_1 - x_3)(y_3 - y_1)$	$d_{11}(y_1 - y_2)(y_3 - y_1) + d_{33}(x_1 - x_3)(x_2 - x_1)$	$d_{12}(x_2 - x_1)(y_3 - y_1) + d_{33}(x_1 - x_3)(y_1 - y_2)$
$d_{21}(x_1 - x_3)(y_2 - y_3) + d_{33}(x_3 - x_2)(y_3 - y_1)$	$d_{22}(x_1 - x_3)(x_3 - x_2) + d_{33}(y_2 - y_3)(y_3 - y_1)$	$d_{12}(x_1 - x_3)(y_3 - y_1) + d_{33}(x_1 - x_3)(y_3 - y_1)$	$d_{22}(x_1 - x_3)^2 + d_{33}(y_3 - y_1)^2$	$d_{21}(x_1 - x_3)(y_1 - y_2) + d_{33}(x_2 - x_1)(y_3 - y_1)$	$d_{22}(x_1 - x_3)(x_2 - x_1) + d_{33}(y_1 - y_2)(y_3 - y_1)$
$d_{11}(y_1 - y_2)(y_2 - y_3) + d_{33}(x_2 - x_1)(x_3 - x_2)$	$d_{12}(x_3 - x_2)(y_1 - y_2) + d_{33}(x_2 - x_1)(y_2 - y_3)$	$d_{11}(y_1 - y_2)(y_3 - y_1) + d_{33}(x_1 - x_3)(x_2 - x_1)$	$d_{12}(x_1 - x_3)(y_1 - y_2) + d_{33}(x_2 - x_1)(y_3 - y_1)$	$d_{11}(y_1 - y_2)^2 + d_{33}(x_2 - x_1)^2$	$d_{12}(x_2 - x_1)(y_1 - y_2) + d_{33}(x_2 - x_1)(y_1 - y_2)$
$d_{21}(x_2 - x_1)(y_2 - y_3) + d_{33}(x_3 - x_2)(y_1 - y_2)$	$d_{22}(x_2 - x_1)(x_3 - x_2) + d_{33}(y_1 - y_2)(y_2 - y_3)$	$d_{12}(x_2 - x_1)(y_3 - y_1) + d_{33}(x_1 - x_3)(y_1 - y_2)$	$d_{22}(x_1 - x_3)(x_2 - x_1) + d_{33}(y_1 - y_2)(y_3 - y_1)$	$d_{21}(x_2 - x_1)(y_1 - y_2) + d_{33}(x_2 - x_1)(y_1 - y_2)$	$d_{22}(x_2 - x_1)^2 + d_{33}(y_1 - y_2)^2$

(A1.2)

onde \underline{D} é a matriz de propriedades do material, 2Δ é a área do elemento, t é a sua espessura e x_i e y_i ($i=1,2,3$) são as coordenadas cartesianas dos nós 1,2 e 3 (FIG. A1.1).

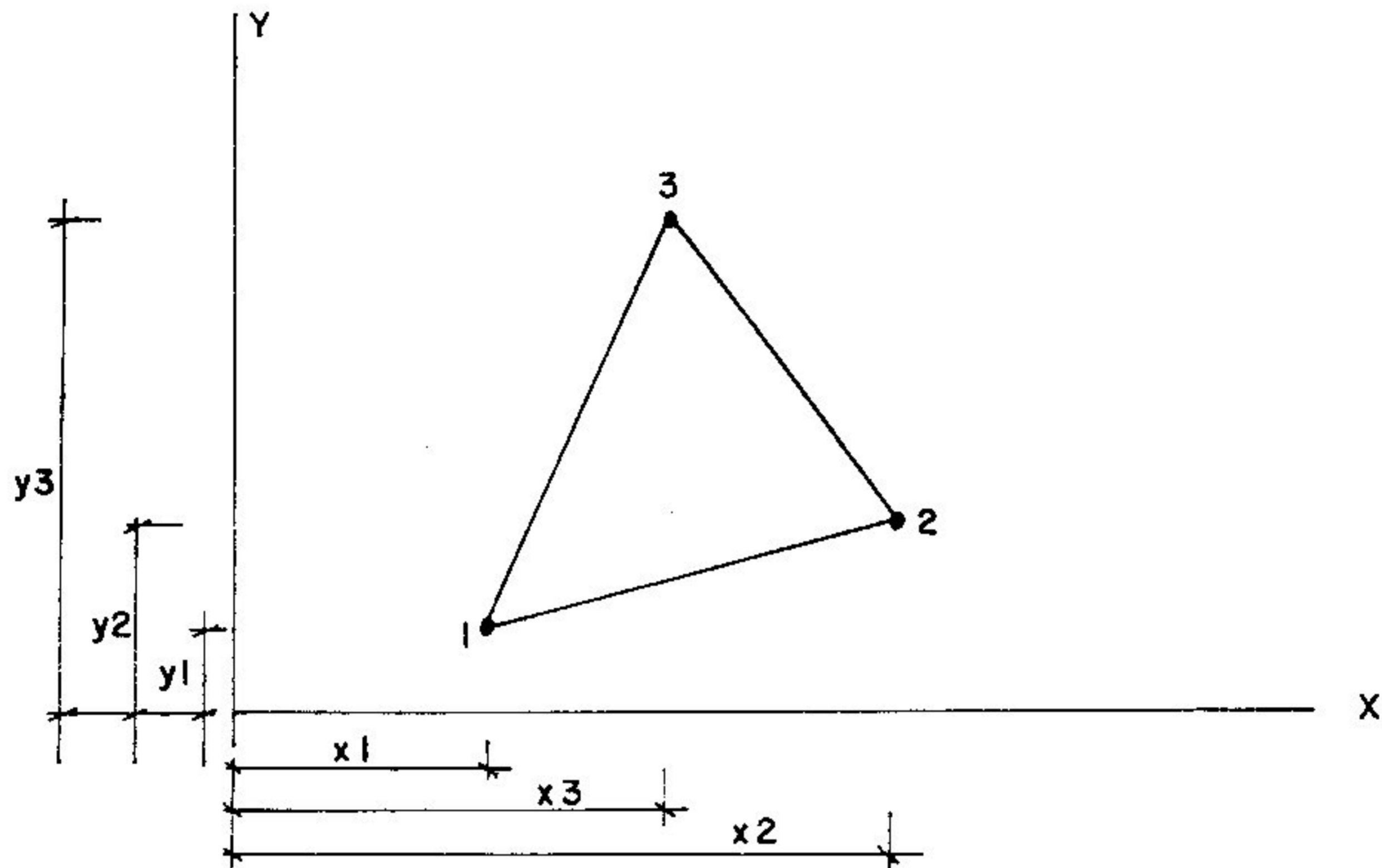


FIG. A1.1 Coordenadas cartesianas dos nós do elemento triangular de três nós.

Cada termo da matriz de rigidez é o resultado da soma de dois produtos de uma grandeza referente às características mecânicas do material por duas projeções dos lados nos eixos cartesianos. Portanto, todos os elementos da matriz (A1.2) têm como forma geral a expressão

$$(d \delta_i \omega_j + d \delta_k \omega_l) \frac{t}{4\Delta} \quad ; \quad i, j, k, l = 1, 2, 3 \quad (A1.3)$$

onde d é um dos elementos da matriz de propriedades do material, δ_i é o comprimento da projeção de um dos lados em x ou y e ω_i é o comprimento da projeção de um dos lados em x ou y .

Considerando-se dois elementos triangulares semelhantes, (FIG. A1.2),

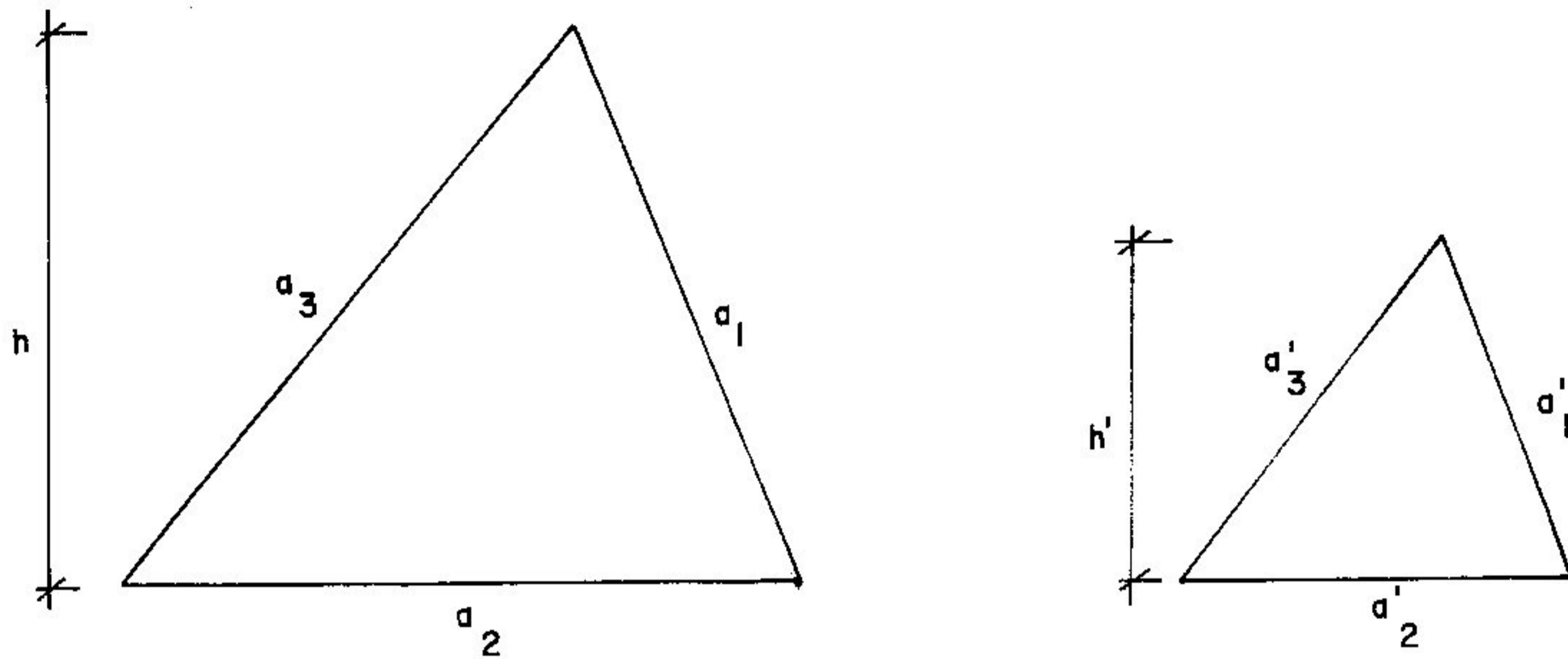


FIG. A1.2 Dois elementos triangulares semelhantes abc e a'b'c' de alturas h e h' e lados a_i e a'_i ($i = 1, 2, 3$) respectivamente.

sabe-se da geometria que

$$\frac{a_1}{a'_1} = \frac{a_2}{a'_2} = \frac{a_3}{a'_3} = \frac{h}{h'} = k \quad (\text{A1.4})$$

onde k é uma constante. Se 2Δ e $2\Delta'$ são as áreas dos dois elementos, tem-se

$$\begin{aligned} 2\Delta &= \frac{1}{2} a_2 h = \frac{1}{2} (k a'_2) (k h') \\ 2\Delta' &= \frac{1}{2} a'_2 h' \end{aligned} \quad (\text{A1.5})$$

e

$$\frac{2\Delta}{2\Delta'} = k^2 \quad . \quad (A1.6)$$

Tem-se, da equação (A1.4),

$$\frac{a_i}{a_i'} = \frac{a_i}{a_i'} = \frac{2\Delta}{2\Delta'} = k^2 \quad i=1,2,3 \quad . \quad (A1.7)$$

Sabendo-se que a relação entre comprimentos de projeções dos lados é igual à relação entre os comprimentos dos lados respectivos,

$$\frac{a_i}{a_i'} = \frac{\omega_i}{\omega_i'} = \frac{\delta_j}{\delta_j'} = k \quad . \quad (A1.8)$$

Então,

$$\frac{\omega_i}{\omega_i'} = \frac{\delta_j}{\delta_j'} = k^2 \quad (A1.9)$$

e, se os dois elementos têm a mesma espessura e são compostos do mesmo material, $d = d'$ e $t = t'$ e

$$\frac{t}{4\Delta} (d \delta_i \omega_j + d \delta_k \omega_l) = \frac{t'}{4\Delta'} (d' \delta_i' \omega_j' + d' \delta_k' \omega_l') \quad . \quad (A1.10)$$

Portanto, as matrizes de rigidez de elementos triangulares semelhantes de três nós para estado plano são iguais, independentemente dos sistemas de coordenadas locais e globais, desde que o material e a espessura sejam os mesmos e que a ordem da numeração dos nós mantenha correspondência entre os ângulos iguais.

APÊNDICE 2

DESCRIÇÃO DO PROCESSADOR ELIGA

O processador ELIGA destina-se a resolver o sistema de equações da malha M3 pelo método de eliminação de Gauss, para comparação do desempenho e resultados com os obtidos pelo Método Multimalha.

Sua estrutura é a mesma da segunda parte do processador SOL do programa original, onde o sistema é triangularizado e resolvido por retro-substituição. O processador difere do original apenas nos bancos de dados utilizados, sendo, aqui, os gerados pelos processadores PREAD, GERA e PREMG.

O diagrama de estrutura do processador ELIGA é mostrado na FIG. A2-1.

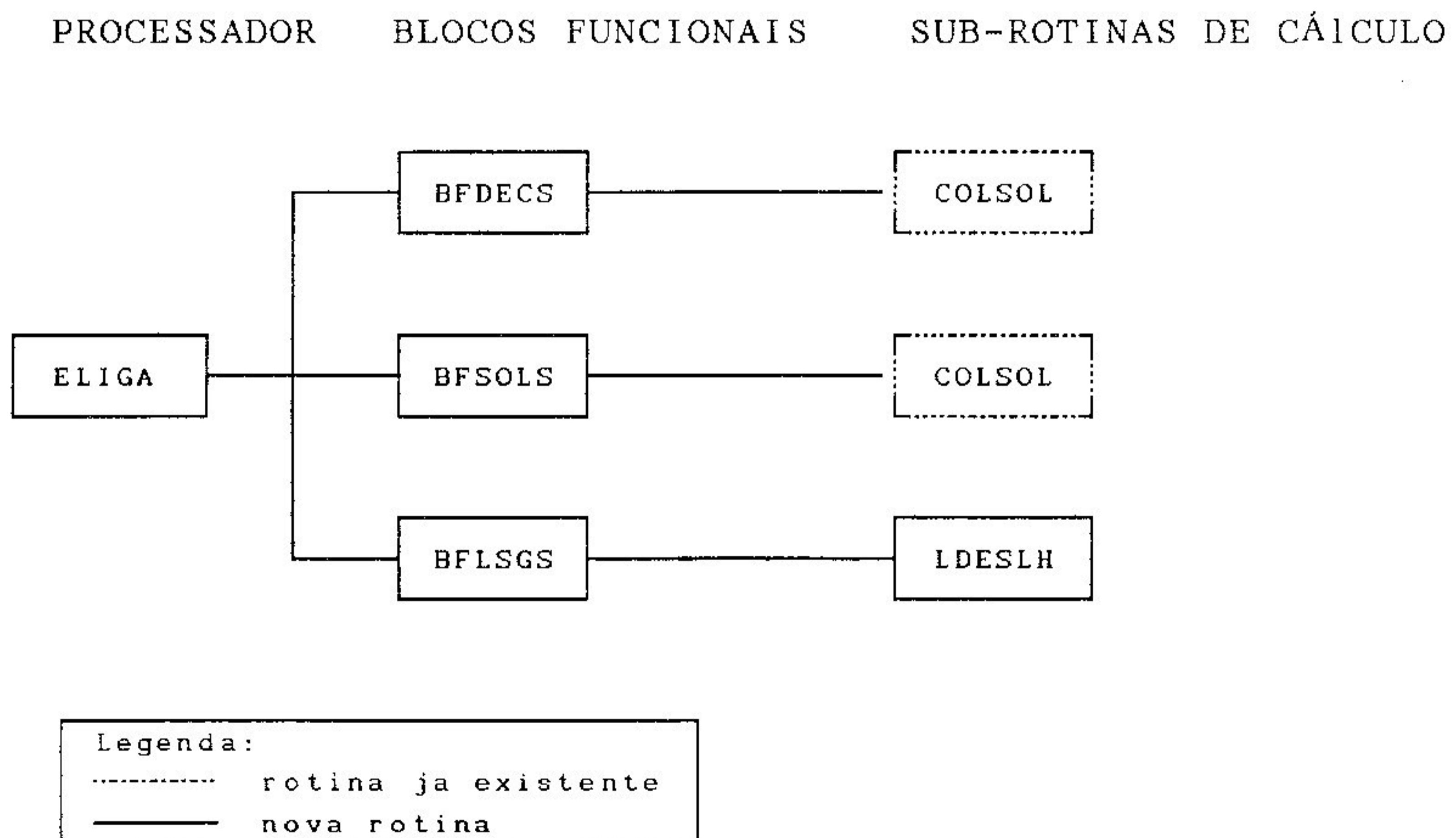


FIG. A2-1 Diagrama de estrutura do processador ELIGA

A função de cada rotina da FIG. A2-1 é descrita sumariamente a seguir.

- * ELIGA: processador que resolve o sistema de equações da malha M3 pelo método de eliminação de Gauss;
- .BFDECS: gerencia a decomposição do sistema de equações de M3, armazenado em forma escalonada, por redução de colunas;
- COLSOL: decompõe o sistema de M3 (triangularização) e resolve-o por retro-substituição;
- .BFSOLS: gerencia a solução do sistema de equações de M3 por retro-substituição;
- .BFLSGS: gerencia a listagem da solução do sistema de M3, associada aos números dos nós e dos graus de liberdade;
- LDESLH: lista a solução do sistema de M3, associada aos números dos nós e dos graus de liberdade.

APÊNDICE 3

DESCRIÇÃO DO PROCESSADOR GAUSDL

O processador GAUSDL destina-se a resolver o sistema de equações da malha M3 pelo método iterativo de Gauss-Seidel, para comparação do desempenho e resultados com os obtidos pelo Método Multimalha.

O sistema foi resolvido por linhas e foram utilizados os mesmos critérios para verificação da convergência e de parada implementados no processador MULGR (Parsons e Hall-1990). Assim, a cada ciclo de iteração é calculado o resíduo

$$\tilde{r}^{(3)} = \tilde{f}^{(3)} - \tilde{K}^{(3)} \tilde{v}^{(3)}$$

e em seguida é calculada a relação

$$R = \frac{\| \tilde{r}^{(3)} \|_2}{\| \tilde{f}^{(3)} \|_2}$$

onde $\| \cdot \|_2$ é a norma euclideana do vetor. Se R for menor ou igual à tolerância ϵ , o processo é interrompido; se não, é iniciado um novo ciclo de iteração.

Este processador utiliza os bancos de dados gerados pelos processadores PREAD, SOL, GERA e PREMG e, como aproximação inicial, foram empregados os valores obtidos pela interpolação dos deslocamentos da malha M0 (após fase R) até a malha M3.

O diagrama de estrutura do processador GAUSDL é mostrado na FIG. A3-1.

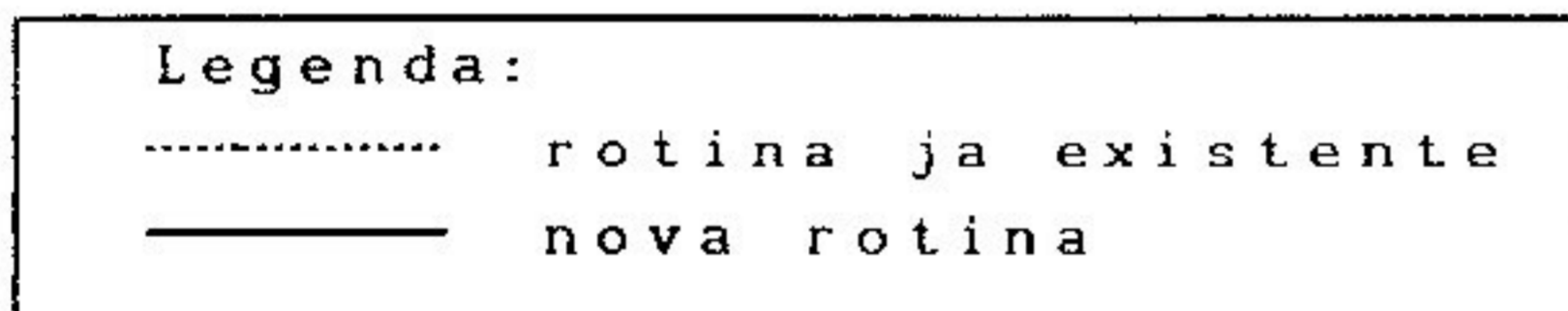
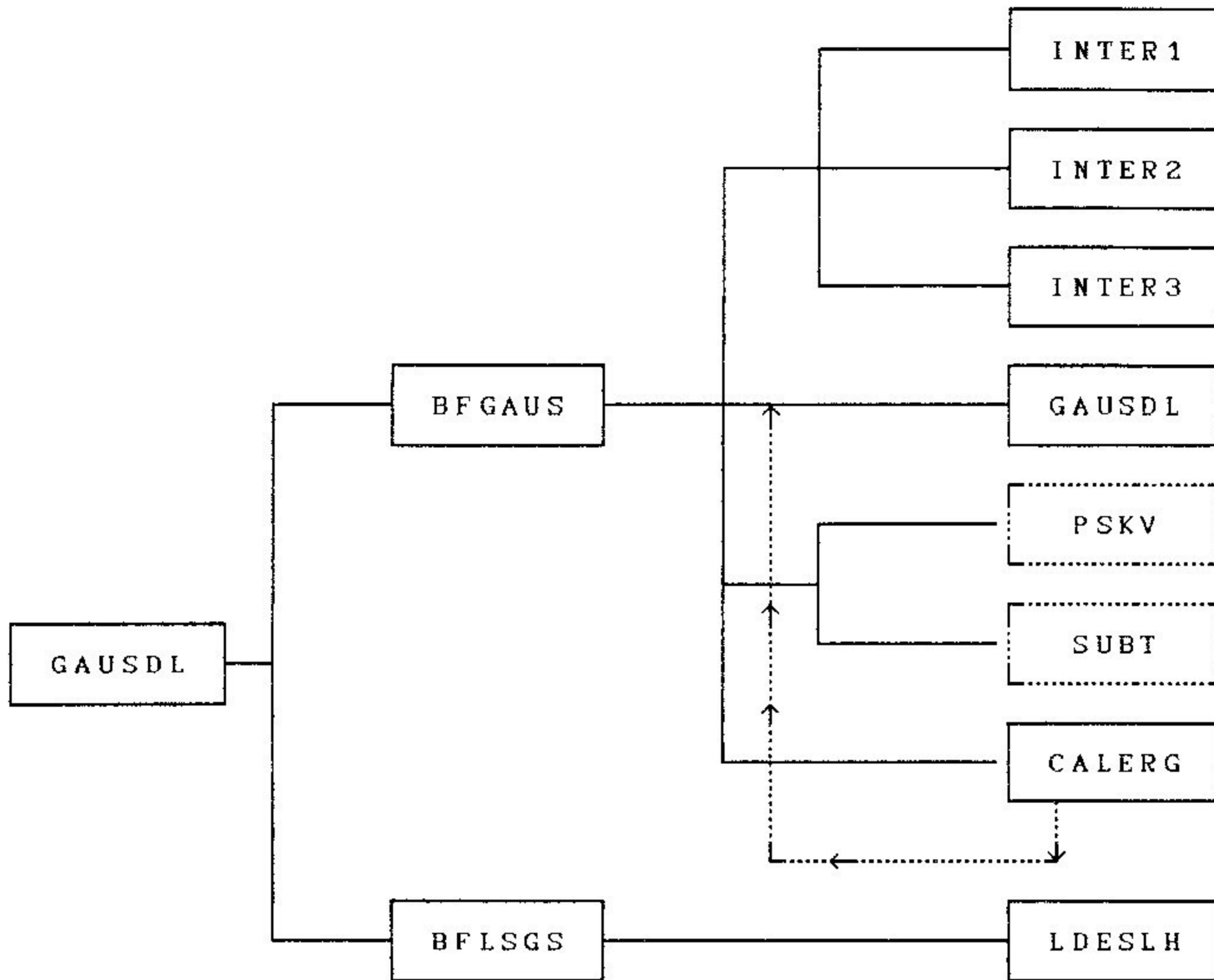


FIG. A3-1 Diagrama de estrutura do processador GAUSDL.

A função de cada rotina da FIG. A3-1 é descrita sumariamente a seguir.

*GAUSDL: processador que resolve o sistema de equações de M3 pelo processo iterativo de Gauss-Seidel;

.BFGAUS: gerencia a solução do sistema de equações de M3 por Gauss-Seidel;

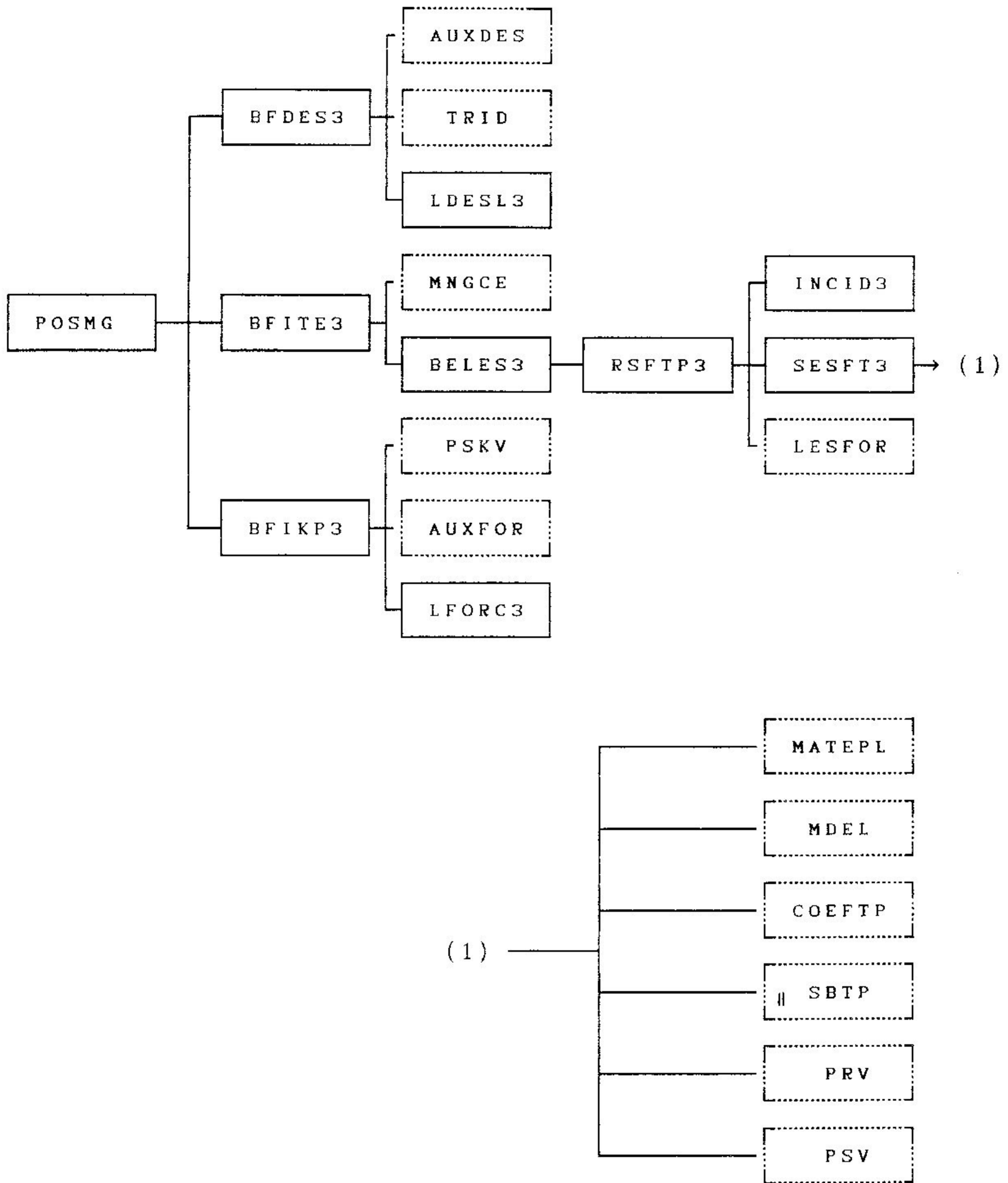
- INTER1: mesma rotina usada no processador MULGR, que interpola valores (deslocamentos) de M0 para M1;
- INTER2: mesma rotina usada no processador MULGR, que interpola valores (deslocamentos) de M1 para M2;
- INTER3: mesma rotina usada no processador MULGR, que interpola valores (deslocamentos) de M2 para M3;
- GAUSDL: resolve o sistema de equações de M3 pelo processo iterativo de Gauss-Seidel (por linhas);
- PSKV: rotina já existente que multiplica uma matriz por um vetor;
- SUBT: rotina já existente que subtrai um vetor de outro;
- CALERG: calcula $R = \frac{\| \tilde{r}^{(3)} \|_2}{\| \tilde{f}^{(3)} \|_2}$ e compara este valor com a tolerância ϵ ; se $R > \epsilon$ o programa retorna à rotina GAUSDL, se $R \leq \epsilon$ o processo é interrompido;
- .BFLSGS: gerencia a listagem dos deslocamentos calculados;
- LDESLH: lista os deslocamentos calculados, associados aos números dos nós e dos graus de liberdade.

APÊNDICE 4

DESCRIÇÃO DO PROCESSADOR POSMG

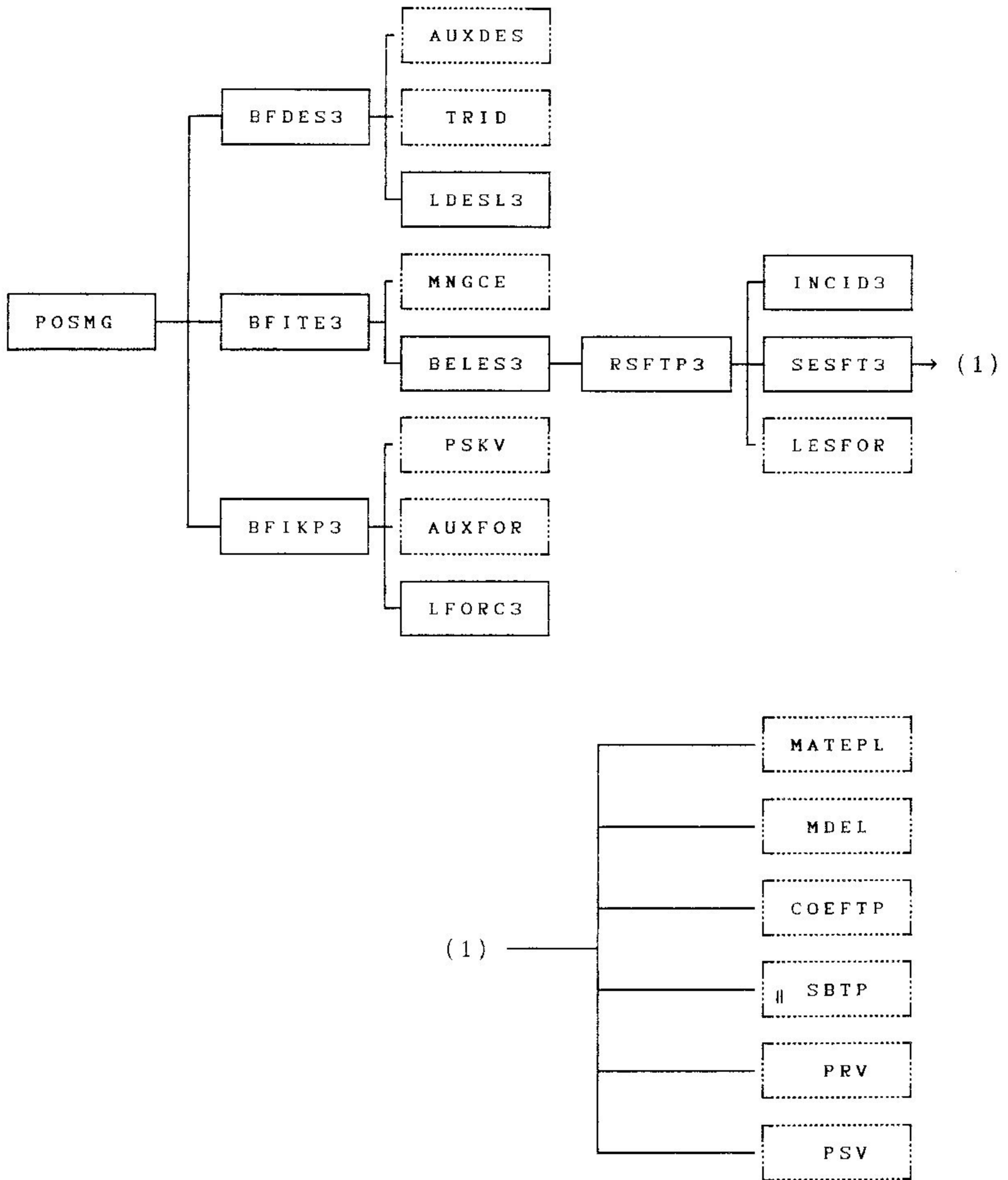
o pós-processador POSMG destina-se a calcular as tensões nos elementos da malha M3 e a imprimir os valores destas tensões, dos deslocamentos, bem como do produto $\tilde{K}^{(3)} \tilde{v}^{(3)}$ de GAUSDL e MULGR e $\tilde{K}^{(3)} \tilde{u}_h^{(3)}$ de ELIGA. Sua estrutura é a mesma do processador POS do programa original e utiliza os bancos de dados gerados pelos processadores PREAD, PRO, GERA, PREMG e MULGR (ou ELIGA ou GAUSDL).

O diagrama de estrutura do pós-processador POSMG é mostrado na FIG. A4-1.



Legenda:
 rotina ja existente
 — nova rotina

FIG. A4-1 Diagrama de estrutura do pós-processador POSMG.



Legenda:
 rotina ja existente
 — nova rotina

FIG. A4-1 Diagrama de estrutura do pós-processador POSMG.

A função de cada rotina da FIG. A4-1 é descrita sumariamente a seguir.

*POSMG: pós-processador que calcula as tensões nos elementos e imprime tensões, deslocamentos e produto \underline{K}_u da malha M3;

.BFDES3: gerencia a impressão dos deslocamentos;

AUXDES: rotina já existente que retorna com nomes dos graus de liberdade a serem impressos;

TRID: rotina já existente que transforma a matriz ID, incluindo informações sobre os graus de liberdade prescritos;

LDESL3: imprime os deslocamentos nodais, salientando os graus de liberdade prescritos;

.BFITE3: gerencia o cálculo e a impressão de tensões;

BELES3: chama rotina compatível com o tipo de elemento; difere da original apenas no banco de dados utilizado;

RSFTP3: gerencia o cálculo e impressão das tensões para elementos triangulares de 3-6 nós, nos estados planos de tensão e deformação; difere da original apenas no banco de dados utilizado;

INCID3: monta tabela de incidências de M3;

SESFT3: calcula tensões para elementos triangulares planos de 3-6 nós; difere da original apenas no banco de dados utilizado;

MATEPL: rotina já existente que fornece a matriz elástica para elementos nos estados planos de tensão e deformação;

MDEL: rotina já existente que monta o vetor de deslocamentos elementares;

COEFTP: rotina já existente que calcula as funções de interpolação e suas derivadas para elementos triangulares planos;

SBTP: rotina já existente que calcula a matriz \underline{B} para elementos triangulares planos;

REFERÊNCIAS BIBLIOGRÁFICAS

- 1) BABUSKA, I. e RHEINBOLDT, W.C. "Error Estimates for Adaptive Finite Element Computations", SIAM Journal of Numerical Analysis, 15, 1978, pp. 736-754.
- 2) BANK, R. E.; DUPONT, T. F. e YSERENTANT, H. "The Hierarchical Basis Multigrid Method", Numer. Math. vol.52, 1988, pp. 427-548.
- 3) BANK, R. e WEISER, A. "Some A Posteriori Error Estimators for Elliptic Partial Differential Equations, Mathematics of Computation, vol. 44, abril 1985, pp. 283-301.
- 4) BRANDT, A. "Guide to Multigrid Development", Proceedings (3), Köln-Poz, 1981, Eds. Lecture Notes in Mathematics, 960, pp. 220-312.
- 5) BRIGGS, W. L. "A Multigrid Tutorial", SIAM, 1987.
- 6) BRIGGS, W. L. e MCCORMICK, S. Cap. 1 - Introduction, "Multigrid Methods". Ed. Stephen F. McCormick, SIAM, 1987.
- 7) CARROL, W. E. "On the Reformulation of the Finite Element Method", Computer and Structures, vol. 8, 1978, pp. 547-552.
- 8) CIARLET, P. G. "The Finite Element Method for Elliptic Problems", North-Holland Publishing Company, 1978.
- 9) CIMINI Jr., C. A.; LAS CASAS, E. B. e TAROCO, A. "Determinação de Fatores de Intensificação de Tensões Mediante o Cálculo da Integral J Usando um Processo Adaptativo R", MECOM-91, Paraná-Santa Fé, Argentina, 1991, pp.427-436.
- 10) CRAIG, A. W. e ZIENKIEWICZ, O. C. "A Multigrid Algorithm using a Hierarchical Finite Element Basis", Multigrid Methods for Integral and Differential Equations, Eds. D. J. Paddon and Holstein, Clarendon Press, Oxford, 1985.

- 11) CYRINO, J. C. R. "Revisão dos Métodos para Redução do Erro de Discretização", Revista Brasileira de Engenharia, vol. 6/N.(2), 1990, pp.53-69.
- 12) DEVLOO, P. R. B. "An H-P Adaptive Finite Element Method for Steady Compressible Flow" Tese de Doutorado, University of Texas at Austin, 1987.
- 13) EWING, R. E. "A Posteriori Error Estimation" , Computer Methods in Applied Mechanics and Engineering 82, 1990, North-Holland, pp. 59-72.
- 14) GAGO, J.; KELLY, D. W.; ZIENKIEWICZ, O. C. "Adaptive Finite Element Schemes and A-Posteriori Error Analysis, and Evaluation of Alternatives", Proceedings 3rd World Conference on FEM, Los Angeles, California, 1981, Eds. Robinson, J..
- 15) GOUVEIA, J. P. "SDP: Sistema para Desenvolvimento de Programas Baseados no Método dos Elementos Finitos", IX COBEM, 1987, pp. 561-564.
- 16) LAS CASAS, E. B. "Um Processo Adaptativo Misto Global para o Método dos Elementos Finitos", Revista Brasileira de Engenharia, vol.6, (2), 1990, pp.41-52.
- 17) LAS CASAS, E. B. "R-H Mesh Improvement Algorithms for the Finite Element Method", Tese de Doutorado, Purdue University, 1988.
- 18) NOOR, A. K. e BABUSKA, I. "Quality Assessment and Control of Finite Element Solutions", Finite Element in Analysis and Design, vol. 3, 1987, pp. 1-26.
- 19) PARSONS, I. D. e HALL, J. F. "The Multigrid Method in Solid Mechanics: Part I - Algorithm Description and Behaviour", Int. J. Num. Meth. Eng., vol. 29, 1990, pp.719-737.
- 20) PARSONS, I. D. e HALL, J. F. "The Multigrid Method in Solid Mechanics: Part II - Practical Applications", Int. J. Num. Meth. Eng., vol. 29, 1990, pp.739-753.
- 21) RANK, E. e ZIENKIEWICZ, O. C. "A Simple Error Estimator in Finite Element Method", Communications in Applied Numerical Methods, vol. 3, 1987, pp. 243-249.

- 22) RIVARA, M. C.; "Adaptive Multigrid Software for the Finite Element Method", Tese de Doutorado, Katholieke Universiteit Leuven, 1984.
- 23) ROCKEY, K. C.; EVANS, H. R.; GRIFFITHS, D. W. e NETHERCOT, D. A. "The Finite Element Method - A Basic Introduction for Engineers", Granada Publishing Limited, 1975.
- 24) STEIN, E.; BISCHOFF, D.; BRAND, G. e PLANK, L. "Adaptive Multi-grid Methods for Finite Element Systems with Bi- and Unilateral Constraints", Comp. Meth. Appl. Mech. Eng., vol. 52, 1985, pp. 873-884.
- 25) STRÜBEN, K. e TROTTEBERG, U. "Multigrid Methods: Fundamental Algorithms, Model Problems Analysis and Applications", Proceedings Köln-Poz 1981, Lecture Notes in Mathematics, vol. 960, Springer-Verlag, pp.1-176.
- 26) TIMOSHENKO, S. P. e GOODIER, J. N. "Teoria da Elasticidade", 3^a ed., 1980, Ed. Guanabara Dois.
- 27) ZHU, J. Z. e CRAIG, A. W. "Finite Element Multigrid Algorithms and their Application to Engineering Problems", Proceedings of the NUMETA '85 Conference Swansea, 1985, pp. 927-932.
- 28) ZIENKIEWICZ, O. C. e ZHU, J. Z. "A Simple Estimator and Adaptive Procedure for Practical Engineering Analysis", Int. J. Num. Meth. Eng., vol. 24, 1987, pp. 337-357.

BIBLIOGRAFIA

- 1) BABUSKA, I. e RHEINBOLDT, W.C. "Error Estimates for Adaptive Finite Element Computations", SIAM Journal of Numerical Analysis, 15, 1978, pp. 736-754.
- 2) BANK, R. E.; DUPONT, T. F. e YSERENTANT, H. "The Hierarchical Basis Multigrid Method", Numer. Math. vol.52, 1988, pp. 427-548.
- 3) BANK, R. e WEISER, A. "Some A Posteriori Error Estimators for Elliptic Partial Differential Equations, Mathematics of Computation, vol. 44, abril 1985, pp. 283-301.
- 4) BRANDT, A. "Guide to Multigrid Development", Proceedings, (3), Köln-Poz, 1981, Eds. Lecture Notes in Mathematics, 960, pp. 220-312.
- 5) BRANDT, A. "Multi-Level Adaptive Solutions to Boundary-Value Problems", Math. Comp., vol. 31, n.138, 1977, pp. 333-390.
- 6) BRANDT, A. "The Weizmann Institute Research in Multi-level Computation: 1988 Report", Proc. 4th Copper Mountain Conf. on Multigrid Methods.
- 7) BRIGGS, W. L. "A Multigrid Tutorial". SIAM, 1987.
- 8) BRIGGS, W. L. e MCCORMICK, S. Cap. 1 - Introduction, "Multigrid Methods". Ed. Stephen F. McCormick, SIAM, 1987.
- 9) CARROL, W. E. "On the Reformulation of the Finite Element Method", Computer and Structures, vol. 8, 1978, pp. 547-552.
- 10) CIARLET, P. G. "The Finite Element Method for Elliptic Problems", North-Holland Publishing Company, 1978.
- 11) CIMINI Jr., C. A. ; LAS CASAS, E. B. e TAROCO, A. "Determinação de Fatores de Intensificação de Tensões Mediante o Cálculo da Integral J Usando um Processo Adaptativo R", MECOM-91, Paraná-Santa Fé, Argentina, 1991, pp.427-436.

- 12) CRAIG, A. W. e ZIENKIEWICZ, O. C. "A Multigrid Algorithm using a Hierarchical Finite Element Basis", Multigrid Methods for Integral and Differential Equations, Eds. D. J. Paddon and Holstein, Clarendon Press, Oxford, 1985.
- 13) CRAIG, A. W. e ZIENKIEWICZ, O. C. "A-Posteriori Error Estimation, Adaptive Mesh Refinement and Multigrid Methods Using Hierarchical Finite Element Bases", MAFELAP 5, Brunel University, maio 1984.
- 14) CYRINO, J. C. R. "Revisão dos Métodos para Redução do Erro de Discretização", Revista Brasileira de Engenharia, vol. 6/N.2, 1990, pp.53-69.
- 15) DEVLOO, P. R. B. "An H-P Adaptive Finite Element Method for Steady Compressible Flow" Tese de Doutorado, University of Texas at Austin, 1987.
- 16) EWING, R. E. "A Posteriori Error Estimation" , Computer Methods in Applied Mechanics and Engineering 82, 1990, North-Holland, pp. 59-72.
- 17) FIGUEIREDO, A. M. G. e LAS CASAS, E. B. "Conjugação de Método Multimalha e Processo Adaptativo para Solução de Problemas de Elasticidade Plana", MECOM-91, Argentina, pp. 651-659.
- 18) GAGO, J.; KELLY, D. W.; ZIENKIEWICZ, O. C. "Adaptive Finite Element Schemes and A-Posteriori Error Analysis, and Evaluation of Alternatives", Proceedings 3rd World Conference on FEM, Los Angeles, California, 1981, Eds. Robinson, J..
- 19) GOUVEIA, J. P. "SDP: Sistema para Desenvolvimento de Programas Baseados no Método dos Elementos Finitos", IX COBEM, 1987, pp. 561-564.
- 20) HACKBUSCH, W. "Multi-Grid Methods and Applications" Springer Series in Comp. Math., vol 4, Springer-Verlag, 1985.
- 21) LAS CASAS, E. B. "Um Processo Adaptativo Misto Global para o Método dos Elementos Finitos", Revista Brasileira de Engenharia, vol.6, (2), 1990, pp.41-52.

- 22) LAS CASAS, E. B. "R-H Mesh Improvement Algorithms for the Finite Element Method", Tese de Doutorado, Purdue University, 1988.
- 23) LAS CASAS, E. B. e FIGUEIREDO, A. M. G. "A Global Multi-Grid Adaptive Process for Finite Elements", ICES-91, Austrália, pp. 578-581.
- 24) NICOLAIDES, R. A. "On Finite Element Multigrid Algorithms and Their Use", MAFELAP, 1978, Academic Press, Londres, pp. 459-466.
- 25) NOOR, A. K. e BABUSKA, I. "Quality Assessment and Control of Finite Element Solutions", Finite Element in Analysis and Design, vol. 3, 1987, pp. 1-26.
- 26) PARSONS, I. D. e HALL, J. F. "The Multigrid Method in Solid Mechanics: Part I - Algorithm Description and Behaviour", Int. J. Num. Meth. Eng., vol. 29, 1990, pp.719-737.
- 27) PARSONS, I. D. e HALL, J. F. "The Multigrid Method in Solid Mechanics: Part II - Practical Applications", Int. J. Num. Meth. Eng., vol. 29, 1990, pp.739-753.
- 28) RANK, E. e ZIENKIEWICZ, O. C. "A Simple Error Estimator in Finite Element Method", Communications in Applied Numerical Methods, vol. 3, 1987, pp. 243-249.
- 29) RIVARA, M. C. "Adaptive Multigrid Software for the Finite Element Method", Tese de Doutorado, Katholieke Universiteit Leuven, 1984.
- 30) RIVARA, M. C. "A 3D Refinement Algorithm Suitable for Adaptive and Multigrid Techniques", Comm. Num. Meth., 1991.
- 31) RIVARA, M. C. "Local Modification of Meshes for Adaptive and/or Multigrid Finite Element Methods", J. Comp. App. Math., vol.36, 1991, pp. 79-89.
- 32) ROCKEY, K. C.; EVANS, H. R.; GRIFFITHS, D. W. e NETHERCOT, D. A. "The Finite Element Method - A Basic Introduction for Engineers", Granada Publishing Limited, 1975.

- 33) STEIN, E.; BISCHOFF, D.; BRAND, G. e PLANK, L. "Adaptive Multi-grid Methods for Finite Element Systems with Bi- and Unilateral Constraints", *Comp. Meth. Appl. Mech. Eng.*, vol. 52, 1985, pp. 873-884.
- 34) STRÜBEN, K. e TROTTEBERG, U. "Multigrid Methods: Fundamental Algorithms, Model Problems Analysis and Applications", *Proceedings Köln-Poz 1981, Lecture Notes in Mathematics*, vol. 960, Springer-Verlag, pp.1-176.
- 35) TIMOSHENKO, S. P. e GOODIER, J. N. "Teoria da Elasticidade", 3^a ed., 1980, Ed. Guanabara Dois.
- 36) ZHU, J. Z. e CRAIG, A. W. "Finite Element Multigrid Algorithms and their Application to Engineering Problems", *Proceedings of the NUMETA '85 Conference Swansea*, 1985, pp. 927-932.
- 37) ZIENKIEWICZ, O. C. e ZHU, J. Z. "A Simple Estimator and Adaptive Procedure for Practical Engineering Analysis", *Int. J. Num. Meth. Eng.*, vol. 24, 1987, pp. 337-357.