

ANÁLISE DE PAVIMENTOS DE EDIFÍCIOS PELO MÉTODO  
DOS ELEMENTOS FINITOS EM MICROCOMPUTADOR

AUTOR : Eng. Marcelo Novaes de Rezende

ORIENTADOR : Prof. Dr. João Batista de Paiva

UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS  
DEPARTAMENTO DE ESTRUTURAS

ANÁLISE DE PAVIMENTOS DE EDIFÍCIOS PELO MÉTODO  
DOS ELEMENTOS FINITOS EM MICROCOMPUTADOR

Eng. Marcelo Novaes de Rezende

Orientador : Prof. Dr. João Batista de Paiva

Dissertação apresentada à Escola  
de Engenharia de São Carlos, da  
Universidade de São Paulo, como  
parte dos requisitos para obten-  
ção do título de "Mestre em  
Engenharia de Estruturas"

COMISSÃO EXAMINADORA :

Prof. Dr. João Batista de Paiva (EESC-USP) (ORIENTADOR)

Prof. Dr. Caetano Traina Jr. (ICMSC-USP)

Prof. Dr. Wilson Sérgio Venturini (EESC-USP)

SUPLENTE :

Prof. Dr. Mitsuo Tsutsumi (EESC-USP)

São Carlos  
Julho de 1990

Aos meus pais,  
Nicias e Judith.

## AGRADECIMENTOS

Agradeço :

ao Prof. Dr. João Batista de Paiva, pela valiosa orientação dada ao longo deste trabalho.

à FAPESP, pela bolsa de estudos concedida.

## RESUMO

Este trabalho aborda aspectos da análise de pavimentos de edifícios em microcomputador, pela aplicação do método dos elementos finitos

Inicialmente, apresentam-se os elementos finitos utilizados na discretização das lajes e vigas dos pavimentos. Destacam-se a formulação do elemento finito triangular DKT, destinado à análise de flexão de placas delgadas, e a formação, a partir da composição de quatro destes, de um elemento quadrangular.

No escopo da implementação computacional realizada, descreve-se a etapa de entrada de dados, a qual é facilitada pelo uso de um interpretador de comandos, adaptável às necessidades específicas dos usuários.

Propõe-se um algoritmo alternativo de montagem da matriz de rigidez da estrutura, caracterizado pela geração das linhas da matriz, correspondentes a cada ponto nodal da discretização efetuada, e pela conseqüente simplicidade na montagem e armazenamento da matriz particionada em blocos de linhas. É também descrito o processo de resolução do sistema de equações lineares, baseado no gerenciamento otimizado da área de memória central do microcomputador.

Finalmente, são exemplificadas aplicações práticas do sistema computacional desenvolvido.

## ABSTRACT

This work is about building floor analysis, using the finite element method in microcomputers.

Initially, the finite elements used in the floor slab and beam discretizations are presented. The DKT triangular plate bending element formulation and the composition of four DKT elements in order to produce a quadrangular plate bending element are detailed.

In the computational implementation scope, first it is described the data entry process, which is made with a command interpreter adaptable to particular users applications.

An alternative algorithm to the structure stiffness matrix assemblage is proposed. The main feature of this algorithm is the generation of the matrix lines, corresponding to each nodal point of the discretization, making easier the partitioned assemblage and storage of the matrix. It is also described the process used to solve the linear system of equations. The basic idea of this solving process is the main memory optimized management.

Finally, several problems are carried out showing practical applications of the computational system developed.

## ÍNDICE

I.	INTRODUÇÃO .....	1
II.	ELEMENTOS FINITOS UTILIZADOS NA DISCRETIZAÇÃO DOS PAVIMENTOS DE EDIFÍCIOS .....	4
	2.1 - Introdução .....	4
	2.2 - Formulação do elemento finito triangular DKT .....	4
	2.2.1 - Introdução .....	4
	2.2.2 - Teoria de placas com pequenos deslocamentos e consideração de deformação por efeito de força cortante .....	5
	2.2.3 - Matriz de rigidez do elemento DKT .....	10
	2.2.4 - Formação de elemento finito quadrangular pela composição de quatro elementos finitos trian- gulares DKT .....	16
	2.2.5 - Obtenção dos esforços inter- nos no elemento triangular DKT e no elemento quadrangular condensado .....	18
	2.2.6 - Elemento finito de barra .....	19
	2.2.7 - Análise de pavimentos de edifícios pela utilização conjunta de elementos finitos de placa e de barra .....	21
III.	SISTEMA COMPUTACIONAL PARA ANÁLISE DE PAVIMENTOS DE EDIFÍCIOS EM MICROCOMPUTADOR .....	23
	3.1 - Introdução .....	23
	3.2 - Descrição funcional .....	24
	3.3 - Estrutura de dados utilizada .....	26
	3.4 - Diagrama de fluxo de dados .....	27
	3.5 - Limitações do sistema computacional .....	28

	3.5.1 - Aplicabilidade.....	28
	3.5.2 - Limitações técnicas.....	29
IV.	A ENTRADA DE DADOS DO SISTEMA	
	4.1 - Introdução.....	31
	4.2 - Processos usuais de entrada de dados.....	31
	4.2.1 - A entrada de dados conversa-	
	cional.....	31
	4.2.2 - Entrada via arquivo de dados.....	32
	4.3 - O processo de entrada de dados proposto..	32
	4.3.1 - Introdução.....	32
	4.3.2 - Características gerais do proces-	
	so proposto.....	33
	4.3.3 - Aspectos da implementação do in-	
	terpretador.....	33
	4.3.3.1 - Algoritmo básico.....	33
	4.3.3.2 - Comandos implementados..	34
	4.3.3.3 - Erros verificáveis pelo	
	interpretador.....	45
	4.3.3.4 - Criação de novos coman-	
	dos.....	46
	4.4 - Exemplo de utilização do interpretador... 47	
	4.4.1 - Caracterização do problema.....	47
	4.4.2 - Utilização dos comandos do in-	
	terpretador na definição do	
	problema.....	50
	4.4.3 - Arquivo do usuário.....	52
	4.4.4 - Confirmação gráfica da entrada	
	de dados.....	55
V.	MONTAGEM ALTERNATIVA DA MATRIZ DE RIGIDEZ E	
	PROCESSO UTILIZADO NA RESOLUÇÃO DO SISTEMA DE	
	EQUAÇÕES LINEARES.....	57



5.1 - Introdução.....	57
5.2 Montagem alternativa da matriz de rigidez da estrutura.....	57
5.2.1 - O processo tradicional.....	57
5.2.2 - Processo alternativo de montagem da matriz de rigidez da estrutura..	59
5.2.3 - Algoritmos para a montagem alternativa da matriz de rigidez da estrutura.....	60
5.2.4 - Imposição de condições de contorno paralelamente à geração da matriz de rigidez da estrutura.....	63
5.3 - Resolução do sistema de equações lineares.....	64
5.3.1 - Introdução.....	64
5.3.2 - Algoritmos auxiliares para a resolução do sistema de equações lineares.....	65
5.3.3 - Algoritmo básico utilizado.....	68

VI. EXEMPLOS DE UTILIZAÇÃO DO SISTEMA COMPUTACIONAL.

6.1 - Introdução.....	70
6.2 - Exemplo 1 : placa quadrada simplesmente apoiada, com carregamento uniformemente distribuído.....	70
6.3 - Exemplo 2 : placa quadrada engastada no contorno, com carregamento uniformemente distribuído.....	71
6.4 - Exemplo 3 : placa esconsa.....	73
6.5 - Exemplo 4 : placa quadrada apoiada em vigas elásticas.....	74
6.6 - Exemplo 5 : Análise de pavimento de edifício.....	75

VII.	CONCLUSÕES DO TRABALHO E SUGESTÕES PARA PROSSEGUIMENTO DE PESQUISA.....	83
VIII.	REFERÊNCIAS BIBLIOGRÁFICAS.....	85
IX.	BIBLIOGRAFIA.....	86

## - CAPÍTULO I -

### INTRODUÇÃO

São numerosos os problemas na área de engenharia de estruturas, envolvendo o elemento estrutural placa. Dentre os mais conhecidos, aparecem os pavimentos de edifícios, os tabuleiros de pontes e os reservatórios.

A notória complexidade matemática do tratamento de tais problemas nos casos reais, quase sempre os fez representar por aproximações de problemas teóricos de solução matematicamente viável na mecânica do contínuo. As limitações evidentes desse tipo de procedimento, levaram à busca de processos operacionalmente factíveis, que conduzissem a respostas precisas aos problemas práticos.

O surgimento dos computadores digitais, proporcionou a solução automática de problemas "discretos", e assim, vislumbrou-se a possibilidade de utilização da máquina para tratamento de problemas do meio contínuo, via discretização do mesmo.

O método dos elementos finitos (assim nomeado no início da década de 60), pode ser definido como um procedimento geral de discretização de problemas contínuos, delineados por expressões definidas matematicamente. O processo de discretização é feito de acordo com o seguinte princípio : o meio contínuo é dividido em um número finito de partes (elementos), cujo comportamento se especifica em função de um número finito de parâmetros. A formulação detalhada do método dos elementos finitos apareceu inicialmente em ZIENKIEWICZ[1], sendo que sucessivas edições ampliadas, o tornam ainda base recomendável para os iniciantes no assunto.

Particularmente, a análise de flexão de placas e cascas pelo método dos elementos finitos tem sido objeto de muita pesquisa, e muito se tem publicado sobre o

assunto, desde os primeiros tempos da efetiva utilização do referido método.

Como pode ser observado em BATOZ[2], a análise de flexão de placas delgadas pode ser realizada genérica e eficientemente, utilizando-se elementos finitos triangulares com nove graus de liberdade (deslocamento e rotações nos pontos nodais localizados nos vértices do triângulo). Em sua análise comparativa dos elementos finitos (com nove graus de liberdade) disponíveis para a análise de flexão de placas delgadas, BATOZ, mostra em um pequeno histórico, os problemas apresentados pelos diversos elementos já desenvolvidos, quando se pensa em generalidade na aplicação do mesmo. Os empecilhos notados, vão desde a falta de convergência para alguns tipos de malha, até a profunda variação de resultados com a mudança da orientação da malha.

Concluindo, BATOZ[2] apresenta o elemento DKT (Discrete Kirchhof Theory), como sendo o mais eficiente numérica e computacionalmente, na análise de placas delgadas. A eficiência computacional com relação à pequena área de memória central requerida, torna o uso do elemento finito DKT extremamente vantajoso em se tratando de microcomputadores.

O objetivo deste trabalho é a viabilização da utilização prática do elemento DKT, através do desenvolvimento de um sistema computacional destinado à análise de pavimentos de edifícios pelo método dos elementos finitos, onde o DKT aparece como elemento discretizador das lajes.

O capítulo II apresenta a formulação do elemento DKT e, além disso, descreve a formação de um elemento quadrangular pela composição de quatro elementos DKT, com posterior condensação estática dos parâmetros internos. A matriz de rigidez de um elemento finito destinado à análise de grelhas também é mostrada. A

utilização conjunta desses três tipos de elementos foi implementada no sistema.

O terceiro capítulo descreve as características gerais do sistema computacional desenvolvido, e as funções específicas de cada módulo componente do mesmo.

O capítulo IV traz a descrição do processo de entrada de dados do sistema, o qual é realizado por um interpretador de comandos que visa descomplicá-lo, ou, se possível, automatizá-lo. A implementação realizada permite também ao usuário, a adaptação do processo de entrada de dados às suas necessidades específicas.

No quinto capítulo apresenta-se a maneira alternativa com que é montada a matriz de rigidez da estrutura, voltada à implementação em computadores de pequena área de memória central. O processo utilizado para a resolução do sistema de equações lineares, também é descrito nesse capítulo.

O capítulo VI traz diversos exemplos de utilização do sistema computacional desenvolvido.

Finalmente, o sétimo capítulo, apresenta as conclusões do trabalho e sugestões para o prosseguimento de pesquisa na área.

- CAPÍTULO II -

ELEMENTOS FINITOS UTILIZADOS NA DISCRETIZAÇÃO DOS  
PAVIMENTOS DE EDIFÍCIOS

2.1 INTRODUÇÃO :

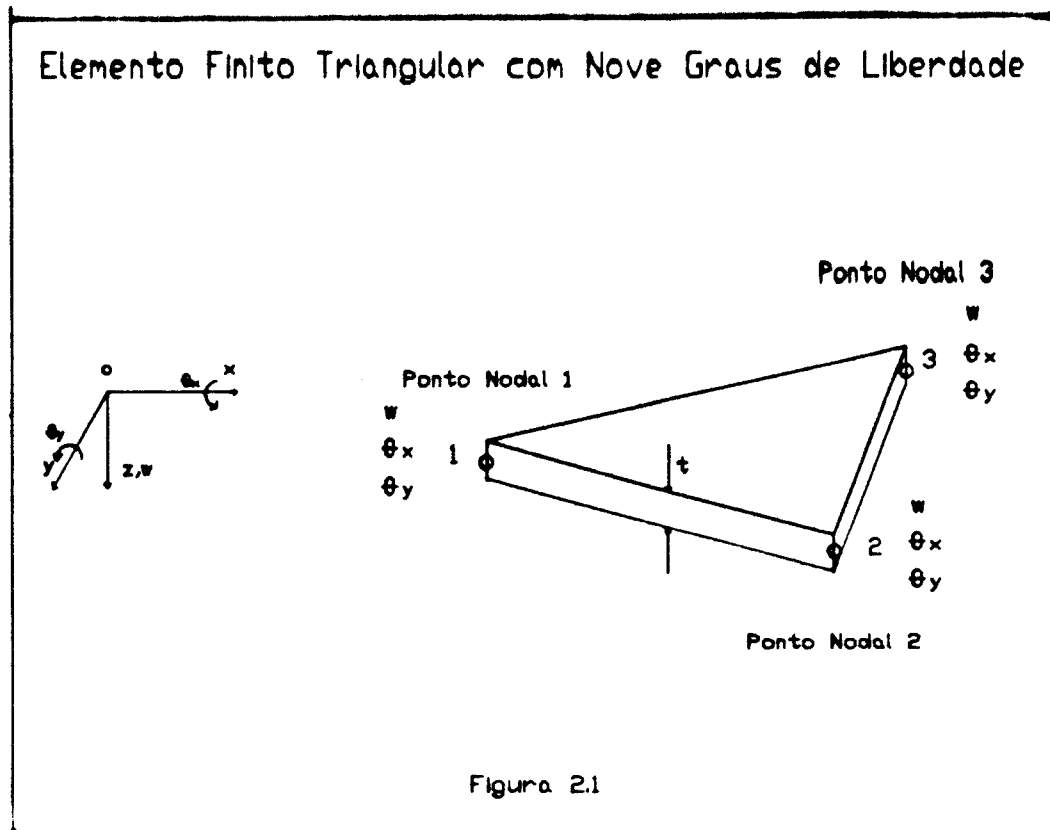
O incremento na utilização do método dos elementos finitos na análise estrutural, tem estimulado a pesquisa de elementos finitos eficientes e confiáveis na solução de problemas cada vez mais genéricos. Em particular, se pretender-se generalidade na análise de flexão de placas delgadas, notar-se-á que poucos elementos finitos apresentam desempenho satisfatório e, dentre esses, o elemento DKT (Discrete Kirchhoff Theory) tem se mostrado seguro e numericamente preciso em variadas análises estáticas e dinâmicas, como pode ser observado em BATOZ [2].

A formulação do elemento DKT, a ser detalhada neste capítulo, baseia-se em BATOZ[2] e C. JEYACHANDRABOSE[3]. A teoria de placas com consideração de deformação por esforço cortante, é o ponto de partida para a formulação. A hipótese clássica de Kirchhoff para placas finas ("pontos da placa originalmente normais à superfície média indeformada, permanecem normais à superfície média deformada"), é imposta discretamente ao longo dos lados do elemento. A parcela da energia de deformação relativa ao esforço cortante é finalmente desprezada, havendo a convergência para o modelo clássico de Kirchhoff para placas delgadas.

2.2 FORMULAÇÃO DO ELEMENTO FINITO TRIANGULAR DKT

2.2.1 INTRODUÇÃO

O elemento finito triangular DKT pertence à classe dos elementos triangulares com 9 graus de liberdade (deslocamento vertical  $w$  e as rotações  $\theta_x$  e  $\theta_y$  nos vértices, como pode ser visto na fig. 2.1), destinados à análise de flexão de placas.



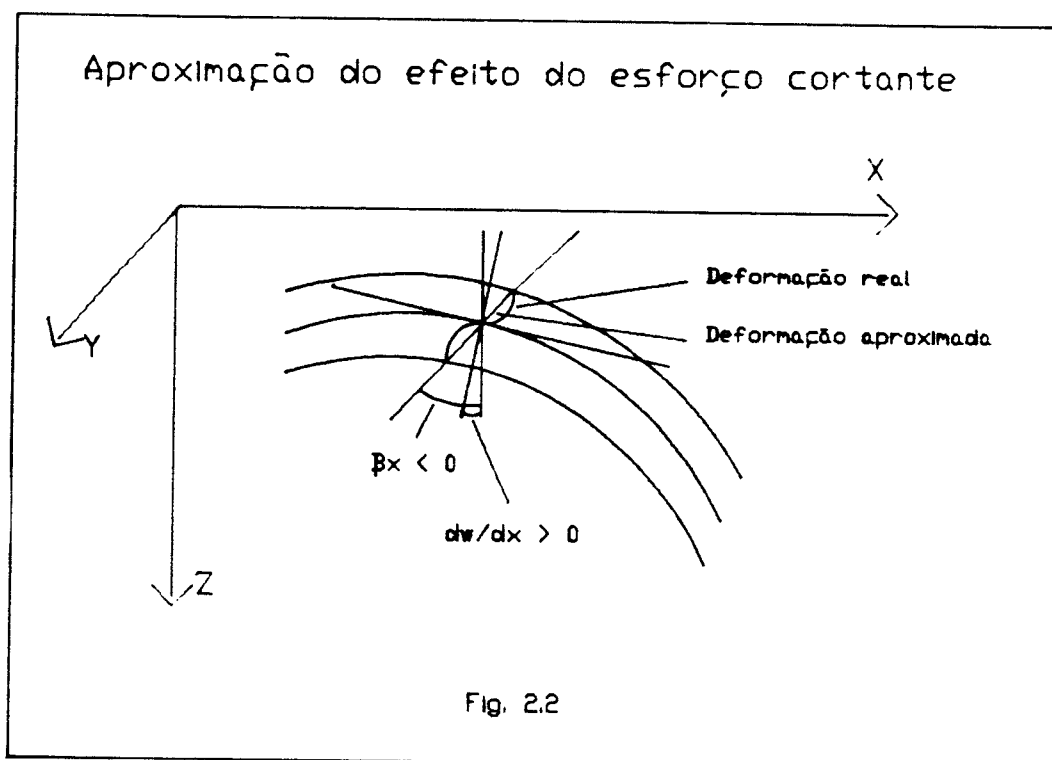
A formulação do elemento DKT envolve conceitos advindos da teoria de placas considerando-se a deformação por esforço cortante, sendo que este fato requererá um breve resumo da referida teoria.

### 2.2.2 TEORIA DE PLACAS COM PEQUENOS DESLOCAMENTOS E CONSIDERAÇÃO DE DEFORMAÇÃO POR EFEITO DE FORÇA CORTANTE

A teoria clássica de placas, devida a Kirchhoff, conduz a uma equação diferencial em derivadas parciais de quarta ordem para o deslocamento vertical  $w$  e, portanto, duas condições de contorno podem se cumprir em cada borda. Para uma placa com espessura finita, é natural exigirem-se três condições de contorno por borda. A razão formal que impede o cumprimento de mais de duas condições de

contorno por borda na teoria clássica é a ordem da equação; fisicamente, provém do fato de desprezar-se a distorção devida aos esforços cortantes ao se estabelecerem as relações tensão-deformação da placa. A não consideração do efeito do esforço cortante, equivale a supor-se infinito o módulo de elasticidade transversal  $G_2$ .

Eric Reissner[8] desenvolveu a teoria de placas considerando o efeito do esforço cortante utilizando-se de uma generalização da hipótese clássica de Kirchhoff: "pontos da placa originalmente normais à superfície média indeformada permanecem em uma linha reta, mas não necessariamente normal à superfície média deformada". Na hipótese de Reissner, o fato dos pontos normais à superfície média indeformada permanecerem em linha reta, implica em assumir-se uma rotação média devida ao esforço



cortante, mesmo sabendo-se que a distribuição de tensões responsável por tal rotação é parabólica, não conduzindo portanto, a uma linha reta (fig. 2.2). A rotação média

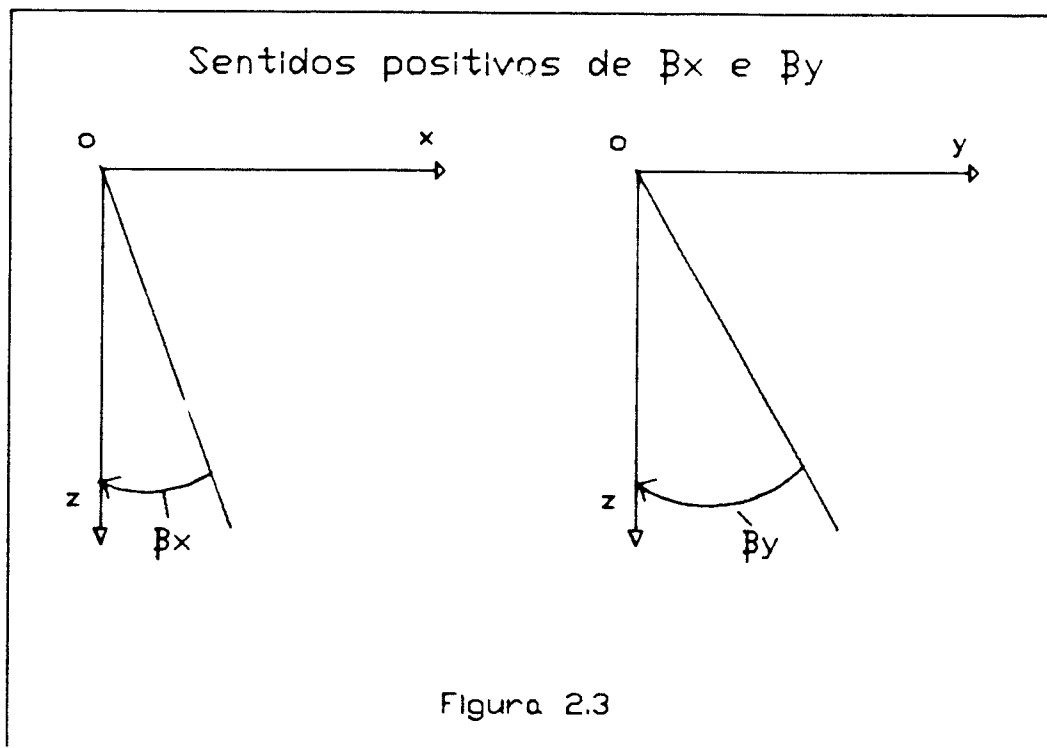


suposta, no entanto, produz trabalho igual às deformações causadas pela real distribuição de tensões.

Utilizando-se a hipótese de Reissner, as componentes de deslocamento de um ponto com coordenadas  $x$   $y$   $z$  são :

$$u = z\beta_x(x,y) \quad v = z\beta_y(x,y) \quad w = w(x,y) \quad (1)$$

onde  $w$  é o deslocamento transversal,  $\beta_x$  e  $\beta_y$  são as rotações da normal à superfície média indeformada nos planos  $x$ - $z$  e  $y$ - $z$  respectivamente ( fig. 2.3). Na teoria clássica, tem-se por hipótese que  $\beta_x = -w,x$  e  $\beta_y = -w,y$ .



As deformações por flexão ( lineares ao longo da espessura ) são dadas por

$$\{\epsilon\}_b = z(\kappa) \quad (2)$$

onde  $\{\kappa\}$  é o vetor de três componentes de curvatura

$$\{\kappa\} = \begin{Bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \beta_{x,y} + \beta_{y,x} \end{Bmatrix} \quad (3)$$

A deformação transversal por esforço cortante pode ser escrita como

$$\{\gamma\} = \begin{Bmatrix} w_{,x} + \beta_x \\ w_{,y} + \beta_y \end{Bmatrix} \quad (4)$$

As relações tensão-deformação referentes a uma placa de espessura constante "h", constituída de material homogêneo elástico e isotrópico, para uma ponto genérico são dadas por

$$\sigma_b = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\nu^2)} z \begin{Bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{Bmatrix} \{\kappa\} = z [D] \{\kappa\} \quad (5)$$

$$\sigma_s = \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \frac{Ek}{2(1+\nu)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \{\gamma\} = [E]\{\gamma\} \quad (6)$$

As variáveis  $E$  e  $\nu$  das equações 5 e 6 são o módulo de Young e o coeficiente de Poisson; "k" é o fator de correção para força cortante, usualmente tomado como sendo 5/6. Os índices "b" e "s" referem-se respectivamente à flexão e ao cisalhamento.

A energia de deformação se escreve como

$$U = U_b + U_s \quad (7)$$

onde

$$U_b = \frac{1}{2} \int_A \{\kappa\}^t [D]_b \{\kappa\} dx dy \quad (8)$$

$$U_s = \frac{1}{2} \int_A \{\gamma\}^t [D]_s \{\gamma\} dx dy \quad (9)$$

e

$$[D]_b = \int_{-h/2}^{h/2} [D] z^2 dz \quad (10)$$

$$[D]_s = k \int_{-h/2}^{h/2} [E] dz \quad (11)$$

As expressões explícitas de  $U_b$  e  $U_s$  são dadas por

$$U_b = K1 \int_A \{ \beta_x, x^2 + \beta_y, y^2 + 2\nu\beta_y, y\beta_{2x}, x + \frac{(1-\nu)}{2} (\beta_y, x + \beta_x, y) \}^2 dx dy$$

$$U_s = \frac{Ehk}{4(1+\nu)} \int_A \{ w, x + \beta_x \}^2 + \{ w, y + \beta_y \}^2 dx dy \quad (12, 13)$$

onde  $K1 = \frac{Eh^3}{24(1-\nu^2)}$

A expressão de  $U$ , dada nas equações 7 a 9 são usadas na formulação de elementos finitos para a análise

de placas espessas e moderadamente espessas. Sendo  $w$ ,  $\beta_x$  e  $\beta_y$  os parâmetros independentes submetidos à variação, somente a continuidade  $C_0$  é requerida, com vistas à conformidade do elemento.

### 2.2.3 MATRIZ DE RIGIDEZ DO ELEMENTO DKT

Na análise de placas delgadas, a deformação transversal por força cortante é desprezível, e, conseqüentemente, a energia de deformação a ela associada também o é. Assim sendo, ao se considerar somente a energia de deformação referente à flexão, teremos a expressão da energia de deformação para um elemento dada por

$$U = \frac{1}{2} \int_A \{x\}^t [D]_b \{x\} dx dy \quad (14)$$

onde  $A$  é a superfície média de um elemento,  $\{x\}$  e  $[D]_b$  são dados nas equações 3 e 10 respectivamente.

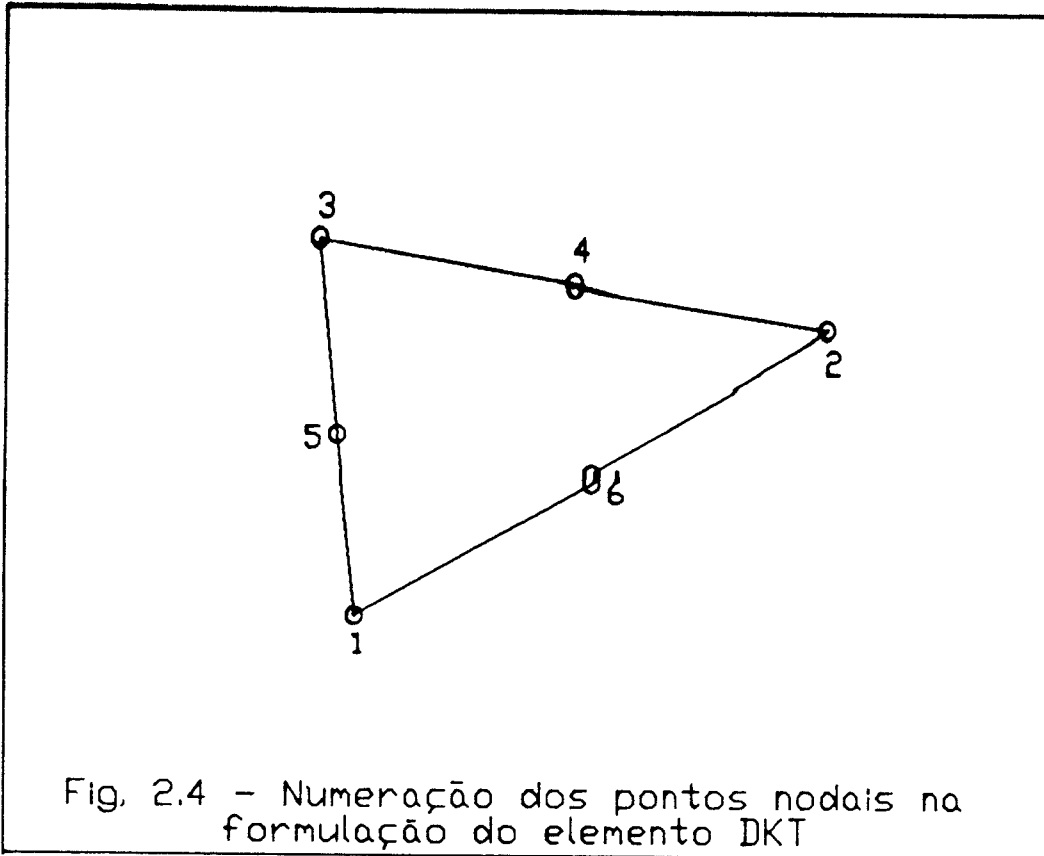
Para a obtenção da matriz de rigidez do elemento DKT, assumem-se, inicialmente, as seguintes hipóteses :

1-  $\beta_x$  e  $\beta_y$  variam quadraticamente no elemento, ou seja

$$\beta_x = \sum_{i=1}^6 N_i \beta_{xi} \quad \text{e} \quad \beta_y = \sum_{i=1}^6 N_i \beta_{yi} \quad (15,16)$$

onde  $\beta_{xi}$  e  $\beta_{yi}$  são os valores nodais nos vértices e nos pontos médios dos lados (fig. 2.4);  $N_i(\xi, \eta)$  são as funções de forma apresentadas a seguir;  $\xi$  e  $\eta$  são as coordenadas de área.

$$\begin{aligned} N_1 &= 2(1-\xi-\eta) \left( \frac{1}{2} - \xi - \eta \right) \\ N_2 &= \xi(2\xi-1) \\ N_3 &= \eta(2\eta-1) \\ N_4 &= 4\xi\eta \\ N_5 &= 4\eta(1-\xi-\eta) \\ N_6 &= 4\xi(1-\xi-\eta) \end{aligned} \quad (17)$$



2- A hipótese de Kirchhoff é imposta nos pontos nodais dos vértices e nos pontos nodais dos pontos médios dos lados :

$$\{\gamma\} = \begin{Bmatrix} \beta_x + w_{,x} \\ \beta_y + w_{,y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \text{ nos pontos 1, 2 e 3}$$

e

$$\beta_{sk} + w_{,sk} = 0 \quad \text{para } k=4, 5 \text{ e } 6 \quad (18, 19)$$

3- A variação de  $w$  ao longo dos lados é cúbica, portanto

$$w_{,sk} = -\frac{3}{2l_{ij}} w_i - \frac{1}{4} w_{,si} + \frac{3}{2l_{ij}} w_j - \frac{1}{4} w_{,sj} \quad (20)$$

onde  $k$  representa o ponto médio do lado  $ij$  e  $l_{ij}$  refere-se

ao comprimento do lado  $l_{ij}$ .

4-  $\beta_n$  varia linearmente ao longo dos lados, ou seja, o valor de  $\beta_n$  no ponto nodal médio dos lados, escreve-se como média aritmética dos  $\beta_n$  dos vértices do referido lado.

Baseando-se nas quatro hipóteses anteriormente citadas e nas particularidades geométricas do triângulo, pode-se escrever  $\beta_x$  e  $\beta_y$  em cada ponto do triângulo, como função dos parâmetros nodais  $\{\delta\}$ , ou seja :

$$\beta_x = [1 \quad \xi \quad \eta \quad \xi^2 \quad \xi\eta \quad \eta^2][G]\{\delta\} \quad (21)$$

$$\beta_y = [1 \quad \xi \quad \eta \quad \xi^2 \quad \xi\eta \quad \eta^2][H]\{\delta\} \quad (22)$$

onde

$$\{\delta\} = (w_1 \quad \theta_{x1} \quad \theta_{y1} \quad w_2 \quad \theta_{x2} \quad \theta_{y2} \quad w_3 \quad \theta_{x3} \quad \theta_{y3}) \quad (23)$$

e as matrizes [G] e [H] são ambas de dimensão 6X9. As matrizes [G] e [H] são dadas a seguir.

$$[G]^t = \begin{bmatrix} 0 & -6a_6 & -6a_5 & 6a_6 & -6(a_5 - a_6) & -6a_5 \\ 0 & 4b_6 & 4b_5 & -4b_6 & -4(b_5 + b_6) & -4b_5 \\ 1 & -3-4c_6 & -3-4c_5 & 2+4c_6 & 4(1+c_5+c_6) & 2+4c_5 \\ 0 & 6a_6 & 0 & -6a_6 & -6(a_4+a_6) & 0 \\ 0 & 4b_6 & 0 & -4b_6 & 3(b_6+b_4) & 0 \\ 0 & -1-4c_6 & 0 & 2+4c_6 & 4(c_6-c_4) & 0 \\ 0 & 0 & 6a_5 & 0 & 6(a_4+a_5) & -6a_5 \\ 0 & 0 & 4b_5 & 0 & 3(b_4-b_5) & -4b_5 \\ 0 & 0 & -1-4c_5 & 0 & 4(c_5-c_4) & 2+4c_5 \end{bmatrix}$$

$$[H]^t = \begin{bmatrix} 0 & 6d_6 & -6d_5 & -6d_6 & 6(d_5-d_6) & 6d_5 \\ -1 & 3+4e_6 & 3+4e_5 & -2-4e_6 & -4(1+e_5+e_6) & -2-4e_5 \\ 0 & -4b_6 & -4b_5 & 4b_6 & 4(b_5+b_6) & 4b_5 \\ 0 & -6d_6 & 0 & 6d_6 & 6(d_4+d_6) & 0 \\ 0 & 1+4e_6 & 0 & -2-4e_6 & 4(e_4-e_6) & 0 \\ 0 & -4b_6 & 0 & 4b_6 & 3(b_6-b_4) & 0 \\ 0 & 0 & 6d_5 & 0 & -6(d_5+d_4) & -6d_5 \\ 0 & 0 & 1+4e_5 & 0 & 4(e_4-e_5) & -2-4e_5 \\ 0 & -4b_6 & -4b_5 & 0 & 3(b_5-b_4) & 4b_5 \end{bmatrix}$$

onde

$$\begin{aligned} a_k &= -x_{ij} / l_{ij}^2 \\ b_k &= \frac{3}{4} x_{ij} y_{ij} / l_{ij}^2 \\ c_k &= \left( \frac{1}{4} x_{ij}^2 - \frac{1}{2} y_{ij}^2 \right) / l_{ij}^2 \\ d_k &= -y_{ij} / l_{ij}^2 \\ e_k &= \left( \frac{1}{4} y_{ij}^2 - \frac{1}{2} x_{ij}^2 \right) / l_{ij}^2 \\ l_{ij}^2 &= (x_{ij}^2 + y_{ij}^2) \end{aligned}$$

$k=4,5,6$  para os lados  $ij=23,31$  e  $12$  respectivamente.

Obtidos  $\beta_x$  e  $\beta_y$ , pode-se obter o vetor de curvaturas da seguinte maneira :

$$\beta_{x,x} = \frac{1}{2\Delta} [1 \quad \xi \quad \eta] \begin{bmatrix} b_2^{(G)}_2 + b_3^{(G)}_3 \\ 2b_2^{(G)}_4 + b_3^{(G)}_5 \\ b_2^{(G)}_5 + 2b_3^{(G)}_6 \end{bmatrix} (\delta) \quad (24)$$

$$\beta_{x,x} = \frac{1}{2\Delta} [1 \quad \xi \quad \eta] [X] (\delta) \quad (25)$$

onde  $b_i = (y_i - y_m)$  e  $c_i = (x_m - x_j)$  com  $i, j$  e  $m$  tomando os valores 1, 2 e 3, ciclicamente,  $\Delta$  é a área do triângulo e  $\langle G \rangle_i$ , representa a  $i$ -ésima linha da matriz  $[G]$ .  
 Similarmente, tem-se os outros elementos dados por :

$$\beta_{y,y} = \frac{1}{2\Delta} [1 \quad \xi \quad \eta] [Y] (\delta) \quad (26)$$

e

$$\beta_{x,y} + \beta_{y,x} = \frac{1}{2\Delta} [1 \quad \xi \quad \eta] [Z] (\delta) \quad (27)$$

onde

$$[Y] = \begin{bmatrix} c_2^{(H)}_2 + c_3^{(H)}_3 \\ 2c_2^{(H)}_4 + c_3^{(H)}_5 \\ c_2^{(H)}_5 + 2c_3^{(H)}_6 \end{bmatrix} \quad (28)$$

$$[Z] = \begin{bmatrix} c_2^{(H)}_2 + c_3^{(G)}_3 + b_2^{(H)}_2 + b_3^{(H)}_3 \\ 2c_2^{(G)}_4 + c_3^{(G)}_5 + 2b_2^{(H)}_4 + b_3^{(H)}_5 \\ c_2^{(G)}_5 + 2c_3^{(G)}_6 + b_2^{(H)}_5 + 2b_3^{(H)}_6 \end{bmatrix} \quad (29)$$

Substituindo-se o vetor de curvaturas na equação 14, tem-se que efetuar apenas as integrais referentes ao produto de  $\{\gamma\}$  e  $\{\gamma\}^t$ , resultando



$$\int_0^{\xi} \int_0^{1-\xi} \begin{bmatrix} 1 \\ \xi \\ \eta \end{bmatrix} [1 \ \xi \ \eta] d\xi d\eta = \frac{1}{24} \begin{bmatrix} 12 & 4 & 4 \\ 4 & 2 & 1 \\ 4 & 1 & 2 \end{bmatrix} \quad (30)$$

Já que

$$\int_0^{\xi} \int_0^{1-\xi} \xi^m \eta^n d\xi d\eta = \frac{m!n!}{(m+n+2)!}$$

Assim, a energia de deformação U pode ser escrita como:

$$U = \frac{1}{2} (\delta)^T [K_{DKT}] (\delta) \quad (31)$$

onde a matriz de rigidez  $[K_{DKT}]$  é dada por :

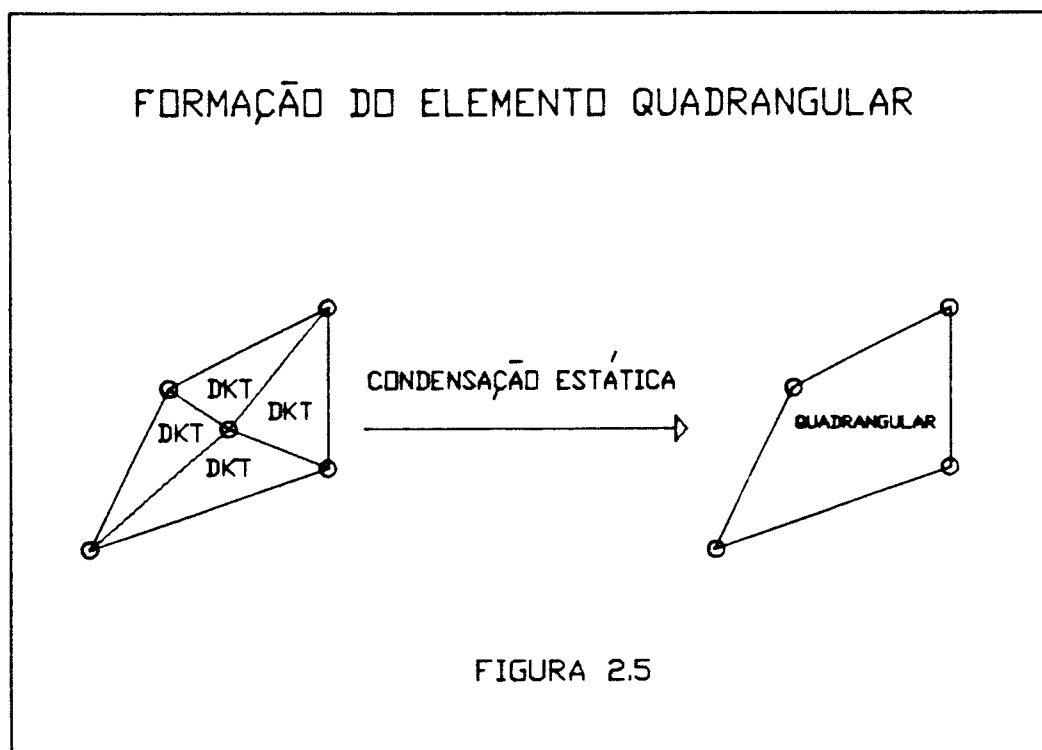
$$[K_{DKT}] = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^T \begin{bmatrix} d_{11}[R] & d_{12}[R] & d_{13}[R] \\ (\text{sim}) & d_{22}[R] & d_{23}[R] \\ & & d_{33}[R] \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (32)$$

onde  $d_{ij}$  são os termos da matriz de elasticidade [D] (apresentada na equação 5 para o caso de placas de material elástico, homogêneo e isotrópico). A matriz [R] é dada por :

$$[R] = \frac{1}{24} \begin{bmatrix} 12 & 4 & 4 \\ 4 & 2 & 1 \\ 4 & 1 & 2 \end{bmatrix} \quad (33)$$

#### 2.2.4 FORMAÇÃO DE ELEMENTO FINITO QUADRANGULAR PELA COMPOSIÇÃO DE QUATRO ELEMENTOS FINITOS TRIANGULARES DKT.

A utilização dos elementos finitos quadrangulares é geometricamente vantajosa na análise de placas, pois esse tipo de elemento pode ser, em particular, um retângulo e, assim, compor malhas facilmente automatizáveis, com base nessa figura. Por outro lado, o elemento quadrangular, pode também criar malhas para figuras grandemente irregulares e, nesse aspecto, ele traz vantagens semelhantes às do elemento triangular.



A geração de um elemento quadrangular, pode ser facilmente implementada, pela composição de quatro elementos triangulares e, posterior condensação estática dos parâmetros internos. Considere-se a figura 2.5, onde aparece um quadrilátero dividido em quatro elementos finitos triangulares. A matriz de rigidez do quadrilátero  $[K]$ , relaciona-se com os parâmetros nodais  $\{\delta\}$  e vetor de cargas  $\{f\}$ , da seguinte maneira :

$$[K]\{\delta\}=\{f\} \quad (34)$$

Essa expressão, pode ser reescrita, distinguindo-se os parâmetros localizados nos vértices do quadrilátero ( $\delta_e$ ), dos parâmetros internos ( $\delta_i$ ) e, fazendo-se o mesmo para o vetor de cargas. Desta forma, a expressão 34 fica expressa como

$$\begin{bmatrix} K_{11} & K_{1e} \\ K_{e1} & K_{ee} \end{bmatrix} \begin{Bmatrix} \delta_i \\ \delta_e \end{Bmatrix} = \begin{Bmatrix} f_i \\ f_e \end{Bmatrix} \quad (35)$$

Explicitando-se os produtos matriciais da expressão 35, tem-se

$$\{f_i\} = [K_{ii}]\{\delta_i\} + [K_{ie}]\{\delta_e\} \quad (36)$$

$$\{f_e\} = [K_{ei}]\{\delta_i\} + [K_{ee}]\{\delta_e\} \quad (37)$$

Da expressão (36), pode-se obter  $\{\delta_i\}$  da seguinte maneira

$$\{\delta_i\} = [K_{ii}]^{-1}(\{f_i\} - [K_{ie}]\{\delta_e\}) \quad (38)$$

Substituindo-se a expressão de  $\{\delta_i\}$  na equação 37, e, considerando-se o fato da matriz de rigidez do quadrilátero ser simétrica, o que resulta  $[K_{ei}] = [K_{ie}]$ , pode-se escrever

$$\{f_e\} - [K_{ei}][K_{ii}]^{-1}\{f_i\} = [[K_{ee}] - [K_{ei}][K_{ii}]^{-1}[K_{ie}]]\{\delta_e\} \quad (39)$$

Resumidamente, a equação (39) pode ser reescrita da forma

$$\{f_c\} = [K_c]\{\delta_e\} \quad (40)$$

onde, finalmente,  $[K_c]$ , seria a matriz de rigidez do quadrilátero, em sua forma condensada, ou seja, tendo-se eliminados os parâmetros internos. O lado esquerdo da equação 39, mostra a alteração sofrida pelo vetor de cargas nodais do quadrilátero.

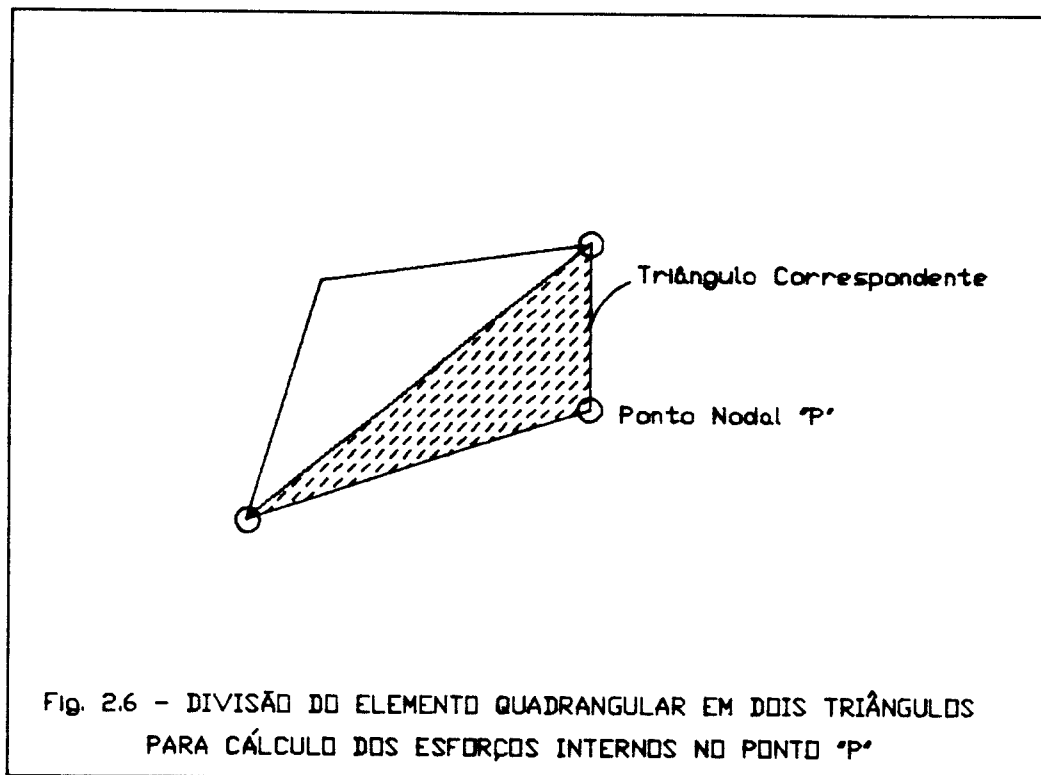
Utilizando-se quatro elementos finitos triangulares DKT, dispostos como na figura 2.5, e, valendo-se do processo de condensação estática descrito anteriormente, pode-se, facilmente, implementar um elemento finito quadrangular. Esse elemento quadrangular é o elemento básico para discretização de lajes, no sistema computacional desenvolvido. Para o usuário, não é

transparente a condensação estática, pois o mesmo apenas define os pontos nodais dos vértices do quadrilátero. O ponto nodal interno é gerado pelo sistema, tendo como coordenadas, a média aritmética das coordenadas dos pontos nodais dos vértices e sendo finalmente eliminado.

#### 2.2.5 OBTENÇÃO DOS ESFORÇOS INTERNOS NO ELEMENTO FINITO TRIANGULAR DKT E NO ELEMENTO QUADRANGULAR CONDENSADO

A obtenção dos esforços internos no elemento triangular DKT (momentos  $x$ ,  $y$  e  $xy$ ), é facilmente realizável, partindo-se das equações 25, 26 e 27, as quais fornecem o valor das curvaturas em todo o domínio do elemento e, em particular, nos seus pontos nodais.

Ao utilizar-se o elemento quadrangular apresentado no item 2.2.4, a determinação dos esforços internos no mesmo, sugere que se obtenham os parâmetros internos



eliminados, recuperando-se os quatro elementos

triangulares DKT originais , permitindo assim, a aplicação do procedimento descrito no parágrafo anterior em cada um dos quatro triângulos. A obtenção dos esforços internos realizada dessa maneira é, certamente ineficiente em termos de número de operações necessárias, tornando desejável a utilização de um processo alternativo.

No sistema computacional desenvolvido, a obtenção dos esforços internos nos elementos quadrangulares se dá, inicialmente pela divisão hipotética do mesmo em dois triângulos, tendo os mesmos, vértices coincidentes com os vértices do quadrilátero (fig. 2.6). Feita a divisão, os esforços internos são calculados da maneira apresentada anteriormente, para cada um dos triângulos. Na implementação computacional realizada, só são obtidos os esforços internos correspondentes aos vértices dos elementos, e para isso há um critério de divisão do elemento quadrangular. De acordo com esse critério, o ponto nodal no qual se calculam os esforços fica sempre oposto ao lado do triângulo que divide o quadrilátero (fig. 2.6). A obtenção dos esforços internos nos vértices do elemento quadrangular pela aplicação desse processo, mostrou-se numericamente precisa nas variadas análises apresentadas no sexto capítulo.

#### 2.2.6 ELEMENTO FINITO DE BARRA

O elemento finito de barra, utilizado na discretização das vigas do pavimento do edifício, tem sua matriz de rigidez relativa ao sistema local de coordenadas (figura 2.7), apresentada a seguir.

$$\begin{bmatrix}
 \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} \\
 0 & \frac{GI_x}{L} & 0 & 0 & -\frac{GI_x}{L} & 0 \\
 -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} \\
 -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} \\
 0 & -\frac{GI_x}{L} & 0 & 0 & \frac{GI_x}{L} & 0 \\
 -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L}
 \end{bmatrix}$$

onde E e G, representam respectivamente o módulo de elasticidade e o módulo de elasticidade transversal do material da barra.  $I_y$  é o momento de inércia da barra, e  $I_x$ , seu momento de inércia à torção. L é o comprimento do elemento finito.

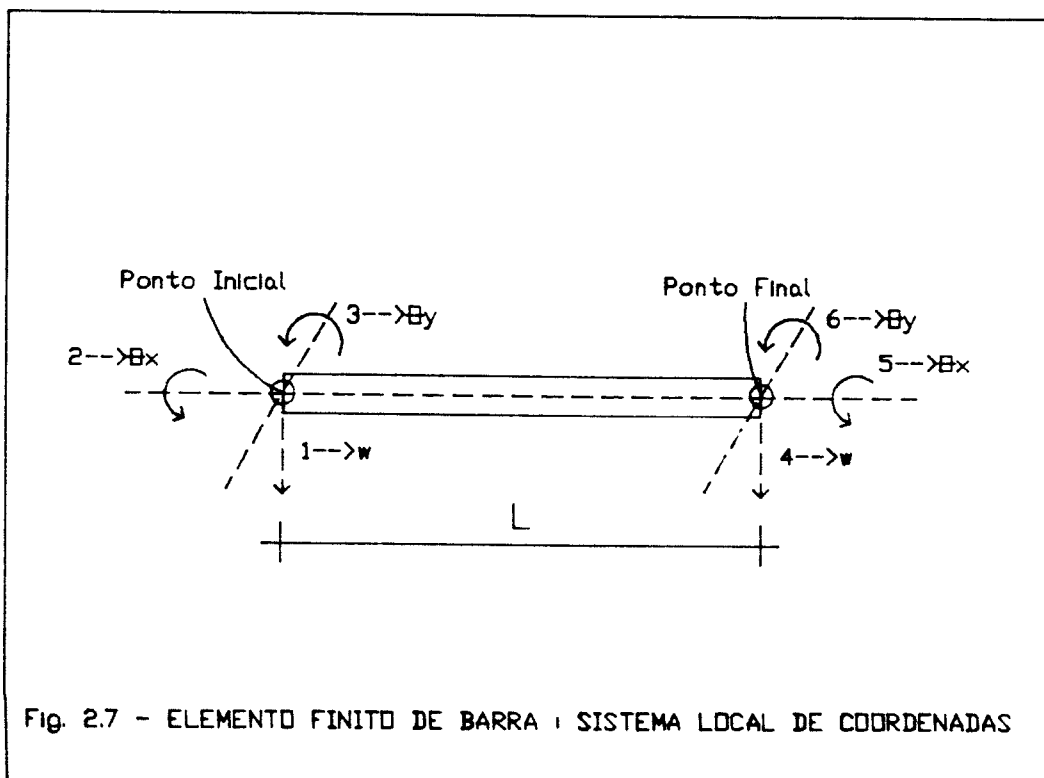


Fig. 2.7 - ELEMENTO FINITO DE BARRA | SISTEMA LOCAL DE COORDENADAS

## 2.2.7 ANÁLISE DE PAVIMENTOS DE EDIFÍCIOS PELA UTILIZAÇÃO CONJUNTA DE ELEMENTOS FINITOS DE PLACA E DE BARRA

Efetuando-se uma discretização de um pavimento em elementos finitos de placa (para as lajes) e elementos finitos de barra (para as vigas), pode-se escrever a energia de deformação correspondente a todos os elementos de placa ( $U_p$ ), da seguinte forma

$$U_p = \frac{1}{2} \{\delta_{pav}\}^t [K_{placas}] \{\delta_{pav}\} \quad (41)$$

onde  $[K_{placas}]$  é a matriz de rigidez referente aos elementos de placa e  $\{\delta_{pav}\}$  é o vetor de parâmetros nodais do pavimento. Analogamente, a energia de deformação das barras ( $U_b$ ) é escrita como

$$U_b = \frac{1}{2} \{\delta_{pav}\}^t [K_{barras}] \{\delta_{pav}\} \quad (42)$$

onde  $[K_{barras}]$  é a matriz de rigidez associada aos elementos de barra.

Como  $U_b$  e  $U_p$  escrevem-se ambos em função do vetor de parâmetros nodais do pavimento, a energia de deformação total do pavimento ( $U_{pav} = U_b + U_p$ ) fica, portanto escrita como

$$U_{pav} = \frac{1}{2} \{\delta_{pav}\}^t [K_{placas} + K_{barras}] \{\delta_{pav}\} \quad (43)$$

fazendo  $[K_{placas} + K_{barras}] = [K_{pavimento}]$ , a expressão (43) fica reescrita da seguinte forma

$$U_{pav} = \frac{1}{2} \{\delta_{pav}\}^t [K_{pavimento}] \{\delta_{pav}\} \quad (44)$$

onde  $[K_{pavimento}]$  é a matriz de rigidez do pavimento.

Utilizando-se o procedimento usual do método dos elementos finitos, pode-se obter o vetor de parâmetros nodais, pela resolução do seguinte sistema de equações lineares

$$[K_{\text{pavimento}}]\{\delta_{\text{pav}}\} = \{CN\} \quad (45)$$

onde  $\{CN\}$ , corresponde ao vetor de cargas nodais do pavimento.



- CAPÍTULO III -

SISTEMA COMPUTACIONAL PARA ANÁLISE DE PAVIMENTOS DE EDIFÍCIOS EM MICROCOMPUTADOR

3.1 INTRODUÇÃO

Conforme apresentado no capítulo anterior, a literatura técnica traz testes comprobatórios da eficiência numérica do elemento finito DKT na análise genérica de placas elásticas delgadas, fato que o torna apto a proceder análises de problemas reais onde a generalidade impera, como é o caso dos pavimentos de edifícios.

A utilização prática do elemento finito DKT na análise de pavimentos de edifícios, implica na existência de um sistema computacional que o utiliza na discretização das lajes do pavimento e, portanto, a eficiência desse instrumento de análise, depende não somente da eficiência numérica do elemento finito utilizado, mas sim, da qualidade global do sistema. Esse princípio norteou o projeto e a efetiva implementação de um sistema computacional para análise de pavimentos de edifícios, pelo método dos elementos finitos, tendo-se por base o elemento finito DKT.

O sistema computacional desenvolvido, é composto por cinco módulos, codificados em linguagem Pascal, sendo cada um deles representado por um programa independente. O interfaceamento intermodular se dá pelo acesso a arquivos comuns em disco.

O presente capítulo, inicialmente resume a estrutura geral do sistema computacional pela apresentação da descrição funcional dos módulos, da estrutura de dados e do diagrama de fluxo de dados e, finalmente, descreve as principais limitações do sistema.

### 3.2 DESCRIÇÃO FUNCIONAL

O primeiro módulo é um interpretador de comandos. A função básica desse módulo é transformar os comandos do arquivo de dados do usuário, o qual contém a descrição do problema a ser analisado, em estruturas de dados passíveis de processamento pelos módulos subsequentes. Após a interpretação do arquivo de dados do usuário, o interpretador gera também um arquivo de saída formatado, o qual traz todas as informações numéricas do problema, obtidas após a interpretação. Esse arquivo de saída presta-se a conferências minuciosas dos dados do problema, ou mesmo à utilização do interpretador em outros sistemas computacionais. O interpretador de comandos é detalhadamente apresentado no quarto capítulo.

Cabe ao segundo módulo montar a matriz de rigidez e o vetor de cargas nodais correspondentes ao pavimento, além de impor as eventuais condições de contorno ao problema. A matriz de rigidez é montada em grupos de linhas consecutivas que vão sendo armazenados em disco e, assim, liberando a área de memória central do computador. No desenvolvimento desse módulo, criou-se um processo alternativo de montagem da matriz de rigidez, o qual é tratado no quinto capítulo.

O terceiro módulo tem por função, obter os parâmetros nodais do pavimento, pela resolução do sistema de equações lineares em banda, formado pela matriz de rigidez do pavimento e seu vetor de cargas nodais. O fato da matriz de rigidez estar particionada em blocos e armazenada em disco, requer algoritmo especial para a resolução do sistema de equações. Desenvolveu-se um processo de solução do sistema de equações que utiliza ao máximo, a disponibilidade de memória central do computador. Esse processo, transfere do disco a maior quantidade possível de blocos da matriz de rigidez para a

memória central, quando determinado termo da matriz de rigidez é requerido, se o termo pertencer a algum dos blocos colocados na memória central, o acesso é imediato; caso contrário, o bloco da memória central que teve menor número de chamadas é gravado, e, em sua área de memória é posto o bloco do disco ao qual pertence o elemento solicitado. Dessa forma, o número de acessos a disco é minimizado e, conseqüentemente, o tempo total de processamento do módulo. A descrição detalhada desse processo aparece no quinto capítulo.

O quarto módulo efetua o cálculo dos esforços internos e reações em todos os pontos nodais do pavimento, para os elementos de placa e de barra. O valor do esforço interno nodal é avaliado, no caso dos elementos de placa, pela média aritmética do referido esforço entre todos os elementos de placa, convergentes ao ponto nodal. Os esforços e deslocamentos nodais dos elementos de placa e viga são armazenados em arquivos para a utilização de módulos de pós processamento, ou seja, módulos, operam os resultados obtidos pelo sistema, com o objetivo de proporcionar uma resposta adequada ao usuário. O quarto módulo gera também, para consulta, um arquivo de saída formatado, contendo todos os resultados do processamento.

O quinto módulo é um exemplo de pós processamento possível para o sistema. Sua função é obter diagramas dos esforços e deslocamentos das lajes, em direções especificadas pelo usuário. As etapas de processamento, fornecem esforços e deslocamentos somente nos pontos nodais e desta forma, o traçado dos diagramas requer aproximações desses valores no domínio dos elementos. Quando se trata de elemento triangular, a aproximação dos valores em seu domínio é feita por uma interpolação quadrática dos valores dos pontos nodais e dos pontos médios dos lados, os quais são obtidos como média aritmética dos valores dos pontos nodais do referido

lado. No caso de elemento quadrangular, o mesmo é dividido em dois triângulos, e então aplica-se o processo para o caso dos elementos triangulares

### 3.3 ESTRUTURA DE DADOS UTILIZADA

Os problemas são basicamente descritos, pelas informações contidas em duas listas lineares armazenadas em disco. A primeira delas traz dados referentes aos elementos finitos, e cada registro dela é formado pelos seguintes campos :

- 1- Pontos nodais : indica os pontos nodais dos vértices do elemento finito.
- 2- Espessura do elemento finito ( não é utilizado no caso de elementos de barra).
- 3- Geometria : quadrangular, triangular ou viga.
- 4- Tipo : se o tipo é diferente de zero, há outro elemento (marcado com o tipo em questão) na malha com as mesmas características, a menos dos pontos nodais.
- 5- Carga : se o elemento é quadrangular ou triangular, é o valor da carga por unidade de área; se o elemento é de barra (viga), refere-se à carga por unidade de comprimento.
- 6- Inércia à flexão e inércia à torção : só são utilizados nos elementos de barra.
- 7- Rotações dos nós : indica as rotações no plano xy, das coordenadas globais dos pontos do elemento. Essas rotações advém de imposição de contorno em bordas esconsas, por exemplo.
- 8- Rotação : variável Booleana (do tipo verdadeiro ou falso) indicadora de rotação em algum dos nós do elemento, se verdadeira.
- 9- Material : Contém o número do material do elemento finito.

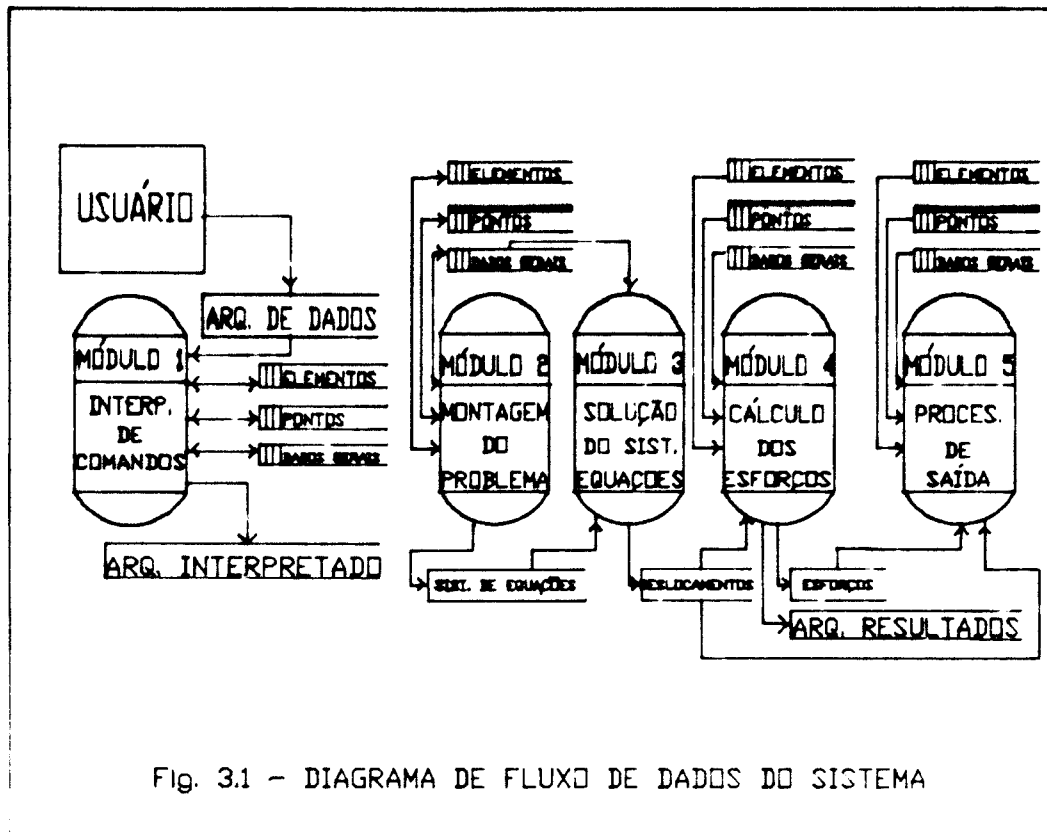
A segunda lista linear básica da estrutura de dados, apresenta informações relativas aos pontos nodais,

e cada registro dessa lista traz os seguintes campos .

- 1- Coordenadas cartesianas  $x$  e  $y$  do ponto nodal.
- 2- Carga concentrada vertical e momento concentrado nas direções  $\theta_x$  e  $\theta_y$ .
- 3- Borda : se o valor armazenado em "borda" é diferente de zero, indica que o ponto nodal pertence à borda correspondente ao referido valor.
- 4- Número de elementos concorrentes no ponto nodal
- 5- Elementos concorrentes no ponto nodal e correspondente ponto nodal : lista contendo o número dos elementos concorrentes no ponto nodal e o número correspondente do ponto, no âmbito do elemento finito.
- 6- Restrições de deslocamento : são três variáveis que informam sobre a existência ou não de restrições ao deslocamento vertical e rotações  $\theta_x$  e  $\theta_y$ , no ponto nodal.
- 7- Rigidezes adicionais : são os valores dos "coeficientes de mola" que podem ser acrescentados ao ponto nodal, na direção vertical e nas rotações  $\theta_x$  e  $\theta_y$ . São utilizados na modelagem dos pilares ligados ao pavimento

### 3.4 DIAGRAMA DE FLUXO DE DADOS

O diagrama de fluxo de dados, apresentado na figura 3.1, tem por objetivo, mostrar o relacionamento entre os módulos do sistema e os arquivos principais. A orientação da seta, indica a orientação da informação. Por exemplo, se a seta converge para o módulo, este recebe informações.



### 3.5 LIMITAÇÕES DO SISTEMA COMPUTACIONAL

#### 3.5.1 APLICABILIDADE

A primeira limitação a ser considerada, refere-se à aplicabilidade do sistema computacional como instrumento de análise, ou seja, quais elementos estruturais podem ser discretizados com os elementos finitos implementados.

Para a análise das lajes do pavimento, o sistema dispõe do elemento finito DKT nas formas triangular e quadrangular condensado. Ambos os elementos, destinam-se unicamente à análise de flexão de placas elásticas delgadas. Desta forma, por hipótese, o material das lajes tem comportamento elástico linear e, além disso, a espessura das mesmas não ultrapassa  $1/5$  da menor das outras dimensões.

Analogamente, as vigas, discretizadas por elementos finitos de barra, deverão ter dimensões tais que

justifiquem sua classificação como tal elemento estrutural, apresentando comprimento maior que três vezes a altura de sua seção transversal. O comportamento do material das vigas, também por hipótese é elástico linear.

### 3.5.2 LIMITAÇÕES TÉCNICAS

Na implementação realizada, as estruturas de dados são sempre armazenadas em disco, e, desta forma, os limites quanto a número máximo de pontos nodais ou elementos finitos, ficam basicamente vinculados à área disponível em disco. Existe, no entanto, limitação da máxima largura de semi banda do sistema de equações. Essa limitação advém do fato da matriz de rigidez estar armazenada em blocos de linhas consecutivas e, cada bloco poder conter no mínimo, três linhas da matriz de rigidez, o que corresponde à contribuição de um ponto nodal. Sendo o número máximo de elementos por bloco igual a dez mil ( $\cong$  64 Kbytes), necessariamente, a largura máxima de semi banda torna-se 3333.

É limitado também em quinze, o número máximo de elementos concorrentes em um ponto nodal. Esse limite existe como consequência do processo alternativo de montagem da matriz de rigidez da estrutura, o qual necessita de uma estrutura de dados associando a cada ponto nodal, os elementos finitos nele convergentes. Alterando-se o dimensionamento da referida estrutura de dados, é portanto possível aumentar-se o limite de elementos convergentes em um ponto nodal; entretanto, o limite estabelecido satisfaz plenamente as necessidades de discretização de pavimentos usuais.

Exemplificando as necessidades de área livre em disco para processamento de problemas, considere-se um pavimento discretizado em 2000 pontos nodais, 1800 elementos finitos e apresentando largura de semi banda igual a 200. A cada ponto nodal, desprezadas as condições

de contorno, correspondem três linhas da matriz de rigidez e assim, o sistema de equações terá 6000 linhas. Sendo 200 a largura de semi banda e, 10000 o número máximo de termos por bloco, cada bloco armazenará 50 linhas e, conseqüentemente, serão utilizados 120 blocos para o armazenamento das 6000 linhas. Cada bloco ocupa 64 Kbytes e, portanto, o armazenamento dos 120 blocos, implicará na utilização de 7,68 Mbytes de área em disco. Além disso, as estruturas de dados dos pontos nodais e elementos, requerem mais aproximadamente 100 bytes cada por unidade de ponto nodal ou elemento finito respectivamente, correspondendo a um acréscimo de 380 Kbytes de área livre em disco. Desta forma, a área livre em disco para o processamento de um problema com essas dimensões, é da ordem de 10 Mbytes.



- CAPÍTULO IV -  
A ENTRADA DE DADOS DO SISTEMA

#### 4.1 INTRODUÇÃO

Os sistemas computacionais científicos, de maneira genérica, requerem pequeno número de dados e efetuam processamento complexo. Em contraposição, os sistemas ditos comerciais, operam com elevado número de dados e processamento de menor complexidade.

Dentre os sistemas científicos, os que se utilizam do método dos elementos finitos, são exceção à regra anteriormente citada, pois além de processamento de considerável complexidade, também recebem grande quantidade de dados, principalmente, quando se almeja maior precisão de resultados. A etapa de entrada de dados nesses sistemas, é portanto, tarefa penosa ao usuário e, muitas vezes, fator desestimulante do uso sistemático do método dos elementos finitos.

No desenvolvimento do presente trabalho, procurou-se criar um processo de entrada de dados que viesse por mitigar, ou mesmo solucionar, os maiores empecilhos dos processos usualmente utilizados.

Inicialmente, como será visto adiante, analisam-se as vantagens e desvantagens dos processos usuais, e, ao final, apresenta-se, detalhadamente, o processo de entrada de dados implementado no sistema.

#### 4.2 PROCESSOS USUAIS DE ENTRADA DE DADOS

##### 4.2.1 ENTRADA DE DADOS CONVERSACIONAL

A entrada de dados conversacional caracteriza-se pela existência de um programa específico, que solicita ao usuário, via terminal de vídeo, todos os dados necessários.

A vantagem desse tipo de entrada é o seu caráter didático, ou seja, usuários com pouca experiência no uso

do sistema, têm uma orientação contínua, durante a etapa de entrada de dados.

Como foi visto anteriormente, nos sistemas que se utilizam do método dos elementos finitos, é elevado o número de dados de entrada, conseqüentemente, a probabilidade de ocorrência de equívocos por parte do usuário é grande. Percebe-se então, que os sistemas conversacionais devem prever várias possibilidades de alteração e verificação dos dados introduzidos. Para fornecer tais recursos no processo conversacional, muitas perguntas adicionais são dirigidas ao usuário, tornando ainda mais complexa a entrada de dados.

#### 4.2.2 ENTRADA VIA ARQUIVO DE DADOS

Nesse tipo de entrada, o usuário, utilizando-se de um editor, monta um arquivo contendo todos os dados necessários ao processamento.

Tipicamente, os arquivos de dados nesse processo, constituem-se de grande quantidade de números e, desta forma, não transmitem ao usuário, com facilidade, o significado dos mesmos na definição do problema. Além disso, editar um arquivo basicamente numérico e com grande quantidade de itens, é sempre uma atividade desgastante para o usuário.

### 4.3 O PROCESSO DE ENTRADA DE DADOS PROPOSTO

#### 4.3.1 INTRODUÇÃO

Conceitualmente, a entrada de dados proposta é feita via arquivo de dados, ou seja, o usuário necessita, *a priori*, editar um arquivo de dados. Entretanto, esse arquivo não se constitui apenas de números, e sim de comandos do tipo "palavras chaves", facilmente memorizáveis.

A idéia básica do processo proposto é a utilização de um interpretador de comandos. A função

básica do interpretador pode ser resumida, como sendo a análise de cada comando do arquivo do usuário, e produção posterior de arquivos compatíveis com a entrada dos módulos de processamento.

#### 4.3.2 CARACTERÍSTICAS GERAIS DO PROCESSO PROPOSTO

A existência de comandos automatizantes para a definição do problema a ser processado, é um dos principais recursos do processo proposto. Os comandos automatizantes tornam o arquivo de dados compacto, e além disso, devido ao caráter mnemônico dos mesmos, fazem-no auto explicativo.

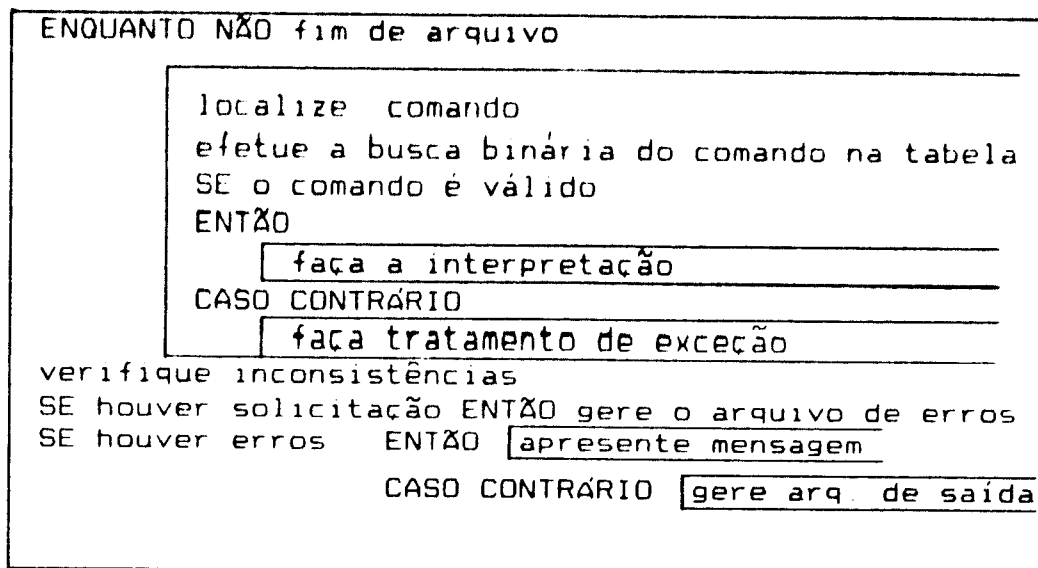
A localização de erros no arquivo de dados é bastante facilitada, pois, opcionalmente, é gerado um arquivo que apresenta todos os problemas verificados pelo interpretador.

A criação de novos comandos por parte do usuário, não tem implementação complexa, como será visto adiante. Essa possibilidade, viabiliza o constante desenvolvimento do processo, sem no entanto, requerer profundas alterações do sistema original.

#### 4.3.3 ASPECTOS DA IMPLEMENTAÇÃO DO INTERPRETADOR

##### 4.3.3.1 ALGORITMO BÁSICO

O algoritmo do interpretador de comandos possui dois grupos básicos de rotinas. As rotinas do primeiro grupo, têm por função localizar os comandos do interpretador no arquivo do usuário, ou seja, realizam uma análise léxica. As rotinas do segundo grupo executam as funções referentes à cada comando definido. O algoritmo básico do interpretador de comandos é apresentado a seguir, em pseudo código.



#### 4.3.3.2 COMANDOS IMPLEMENTADOS

Apresenta-se, a seguir, a lista dos comandos já implementados no interpretador:

01-Palavra chave: BORDA

-Parâmetros : Vinculação, número de tramos (número de pontos -1) e os pontos pertencentes à borda.

-Exemplo : borda engaste 5 1 2 3 4 5 6

-Análise : Define uma borda engastada entre os pontos 1 e 6 da placa.

02-Palavra chave: CARGA\_DIST\_CONSTANTE

-Parâmetros : Carga

-Exemplo : carga\_dist\_constante 2.0

-Análise : Atribui carga distribuída constante igual a 2, para todos os elementos de placa até então definidos.

03-Palavra chave: ELEMENTO\_CARREGADO

-Parâmetros : Número do elemento e carga.

-Exemplo : elemento\_carregado 4 1.5

-Análise : Atribui carga distribuída constante igual a 1.5 ao elemento número 4.

04-Palavra chave: ESPESSURA

-Parâmetros : Número do elemento e espessura

-Exemplo : espessura 6 1 0

-Análise : Atribui espessura 1 0 ao elemento 6. Só se aplica a elementos finitos de placa.

05-Palavra chave: ESPESSURA\_CONSTANTE

-Parâmetros : Espessura

-Exemplo : espessura\_constante 2.0

-Análise : Atribui espessura 2.0 a todos os elementos de placa então definidos.

06-Palavra chave: GERE

-Parâmetros : Saída a ser gerada

-Exemplo : gere listing

-Análise : Gera um arquivo que aponta os possíveis erros do arquivo de dados.

07-Palavra chave: INSIRA\_PONTO

-Parâmetros : Número do ponto, coordenadas cartesianas x e y.

-Exemplo : Insira\_ponto 45 12.0 20.0

-Análise : Define o ponto 45, com as coordenadas cartesianas 12.0 e 20.0 para x e y respectivamente. Se o ponto 45 já está definido, tem seu número alterado para 46, incrementando também, o número de todos os pontos nodais com numeração superior a 45.

A vantagem desse comando é a definição de novos pontos na malha, sem grande aumento da largura da semi banda do sistema de equações lineares. Isto ocorre, pois a largura da semi banda é proporcional à maior diferença entre os números dos pontos nodais de um mesmo elemento finito, e ao inserirmos um ponto com numeração próxima à dos vizinhos, não haverá alteração substancial da largura

da semi banda.

08-Palavra chave: MATERIAL

-Parâmetros : Número do material, Módulo de Young e Coeficiente de Poisson

-Exemplo : material 2 2100.0 0.2

-Análise : Define o material 2 do problema com módulo de Young igual a 2100 (unidade compatível com o restante da definição do problema) e coeficiente de poisson igual a 0.2. É comando obrigatório.

09-Palavra chave: NUMERO\_DE\_ELEMENTOS

-Parâmetros : Número total de elementos

-Exemplo : numero\_de\_elementos 500

-Análise : Define o número total de elementos como sendo 500. A função desse comando é de auxílio à verificação de consistência de dados por parte do interpretador. É um comando opcional.

10-Palavra chave: NUMERO\_DE\_PONTOS

-Parâmetros : Número total de pontos

-Exemplo : numero\_de\_pontos 340

-Análise : Define o número total de pontos nodais como sendo 340. Analogamente ao comando "NUMERO\_DE\_ELEMENTOS", tem função de auxílio à verificação de consistência dos dados. Também é um comando opcional

11-Palavra chave: PONTO

-Parâmetros: Número do ponto e coordenadas cartesianas x e y.

-Exemplo : ponto 67 20.0 30.0

-Análise : Define o ponto nodal 67, com as coordenadas cartesianas 20 e 30 para x e y respectivamente.

Diferentemente do comando "INSIRA\_PONTO", caso já exista o ponto 67, o mesmo será redefinido com as novas coordenadas

cartesianas lidas. Nada se altera nos outros pontos nodais já definidos.

12-Palavra chave: PONTOS

-Parâmetros: Número de pontos a serem lidos (mesma linha da palavra chave), seguindo-se a definição dos mesmos (linhas seguintes, um ponto por linha).

-Exemplo : pontos 3  
23 1.0 2.0  
34 90.0 89.0  
76 0.0 0.0

-Análise : Define os pontos 23 34 e 76 com as coordenadas cartesianas 1 e 2, 90 e 89, e 0 e 0, respectivamente. Analogamente ao comando "PONTO", efetua a redefinição de pontos (se os mesmos já estiverem definidos).

13-Palavra chave: PONTOS\_AUTOMATICOS

-Parâmetros: Número de pontos a gerar, número do primeiro ponto, incremento na numeração dos pontos gerados, coordenadas cartesianas do ponto inicial e coordenadas cartesianas do ponto final

-Exemplo : pontos\_automaticos 3 1 1 0.0 0.0 9.0 0.0

-Análise : Gera, automaticamente, 3 pontos. O primeiro ponto tem número 1 e coordenadas cartesianas  $x,y=0,0$ . O ponto final tem coordenadas  $x,y=9,0$ . O incremento na numeração é 1, assim, temos os pontos 1,2 e 3, definidos da seguinte maneira :

ponto	coordenada x	coordenada y
1	0.0	0.0
2	4.5	0.0
3	9.0	0.0

14-Palavra chave: PONTO\_CARREGADO

-Parâmetros: Número do ponto , carga vertical e momentos

**aplicados nas direções  $\theta_x$  e  $\theta_y$**

-Exemplo : ponto 1 20 0 0 0 0.0

-Análise : Atribui ao ponto 1 a carga vertical de valor 20 0 e momentos nulos nas direções  $\theta_x$  e  $\theta_y$  (unidades coerentes com o restante da definição do problema).

**15-Palavra chave: PONTO\_VINCULADO**

-Parâmetros : Número do ponto vinculado, restrição w, restrição  $\theta_x$  e restrição  $\theta_y$ .

-Exemplo : Ponto vinculado 3 1 0 0

-Análise : Define que o ponto 3 terá deslocamento vertical impedido e as rotações  $\theta_x$  e  $\theta_y$  liberadas (1=impedido, 0=liberado).

**16-Palavra chave: QUADRANGULO**

-Parâmetros: Número do elemento, número dos quatro vértices, definidos de maneira cíclica e horária e material do elemento.

-Exemplo : quadrangulo 2 3 4 5 1 2

-Análise : Define o elemento quadrangular com número 2 e tendo como vértices os pontos 3,4,5 e 1 e material 2. Se já existir o elemento 2, ele será redefinido com as novas características lidas.

**17-Palavra chave: QUADRANGULOS**

-Parâmetros: Número de elementos quadrangulares a serem lidos (mesma linha da palavra-chave), seguindo-se a definição dos elementos (nas próximas linhas, sendo um elemento por linha).

-Exemplo : quadrangulos 2  
7 3 2 5 6 1  
9 8 10 15 14 1

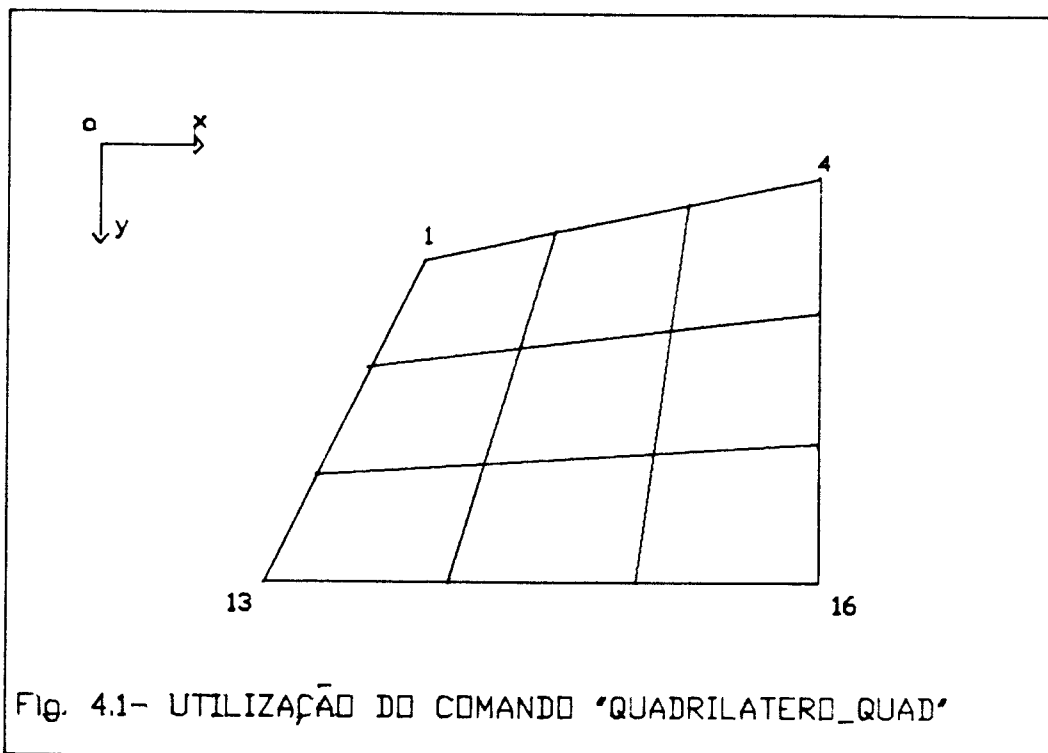
-Análise : Define dois elementos quadrangulares (7 e 9), respectivamente com os vértices 3-2-5-6 e 8-10-15-14, ambos com material 1. Como no caso do comando



"QUADRANGULO", se houver definição prévia dos elementos 7 e 9, haverá redefinição

18-Palavra chave QUADRILATERO\_QUAD

-Parâmetros Número do primeiro elemento finito a ser gerado, número de pontos do quadrilátero, número do vértice superior esquerdo, número do vértice superior direito, número do vértice inferior esquerdo, número do vértice inferior direito, número do primeiro ponto de numeração maior que o do vértice superior direito e número do material dos elementos gerados.



-Exemplo : quadrilatero\_quad 16 1 1 4 13 16 5 1

-Análise : Gera um quadrilátero de 16 pontos, contendo apenas elementos quadrangulares ( figura 4.1), entre os pontos 1 (superior esquerdo), 4 (superior direito), 13 (inferior esquerdo) e 16 (inferior direito). O número do primeiro elemento é 1, e os próximos têm numeração incrementada de 1, da esquerda para direita e de cima para

baixo. Na ordem crescente, o próximo ponto nodal após o 4 é o 5 (pode ser diferente, se a linha de pontos entre 1 e 4 contiver mais pontos nodais).

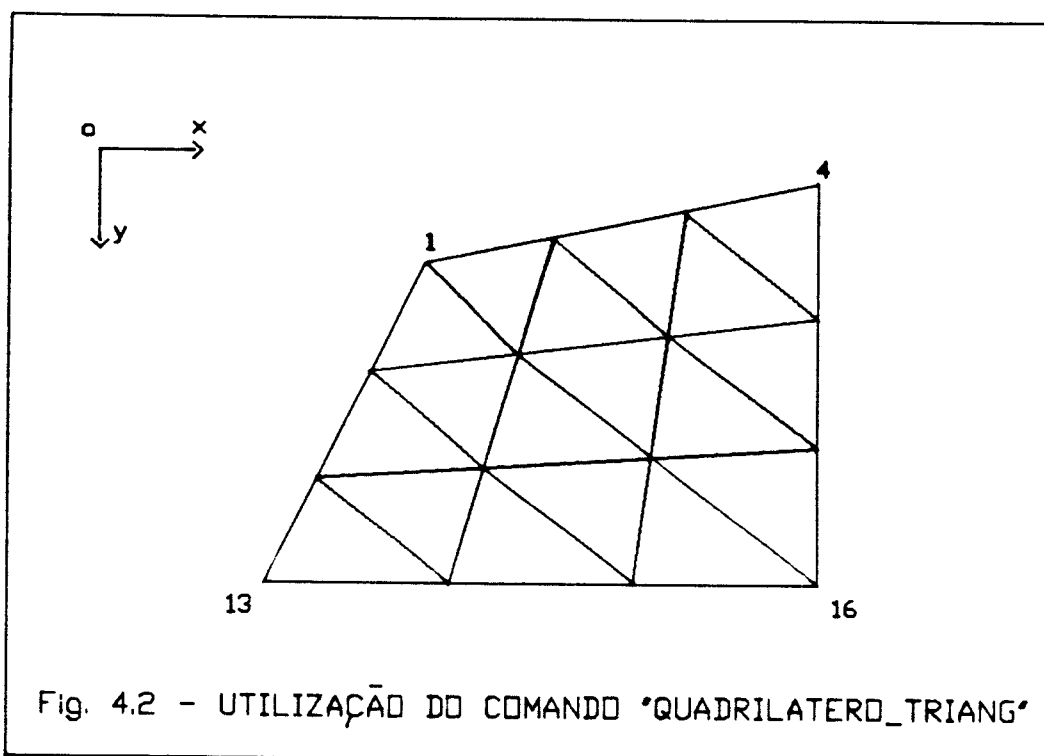
Os pontos dos vértices devem estar pré definidos, ou seja, comandos geradores de pontos devem ter sido utilizados previamente e, portanto, as coordenadas dos pontos 1,4,13 e 16, devem estar estabelecidas.

Os pontos nodais intermediários (entre os vértices), além dos elementos finitos, também são automaticamente definidos e têm numeração crescente da esquerda para a direita e de cima para baixo.

O comando "QUADRILATERO\_QUAD" tem grande poder de automatização de geração de malha, bastando ao usuário, dividi-la convenientemente em quadriláteros.

19-Palavra chave: QUADRILATERO\_TRIANG

-Parâmetros : Os mesmos de "QUADRILATERO\_QUAD".



-Exemplo : O mesmo de "QUADRILATERO\_QUAD".  
-Análise : O comando "QUADRILATERO\_TRIANG" difere do comando "QUADRILATERO\_QUAD", apenas no fato dos elementos gerados serem triangulares (figura 4.2). Assim, cada elemento quadrangular gerado pelo comando "QUADRILATERO\_QUAD", fica dividido em dois elementos triangulares. A orientação da divisão é tal que o vértice superior esquerdo e o inferior direito de cada quadrângulo, sempre pertencem aos dois triângulos gerados.

20-Palavra chave: QUADRO\_CARGA

-Parâmetros : Carga, ponto nodal extremo superior esquerdo e ponto nodal extremo inferior direito.

-Exemplo : quadro\_carga 10.0 23 67

-Análise : Os elementos finito de placa, cujos pontos médios localizarem-se no interior do retângulo com vértice superior esquerdo definido pelo ponto 23, e vértice inferior direito definido pelo ponto 67, terão carga distribuída igual a 10. Os pontos nodais 23 e 67 devem estar pré definidos.

Caso os elementos internos ao retângulo já tenham sua carga distribuída definida, haverá a redefinição.

21-Palavra chave: QUADRO\_ELIMINAR

-Parâmetros: Ponto nodal extremo superior esquerdo e ponto nodal extremo inferior direito.

-Exemplo : quadro\_eliminar 5 89

-Análise : Elimina todos os elementos finitos de placa, cujos pontos médios estejam internos ao retângulo de vértice superior esquerdo definido pelo ponto 5, e vértice inferior direito definido pelo ponto 89.

A definição de aberturas no pavimento, principalmente devidas à existência de elevadores e escadas, são facilmente realizadas, com a utilização deste comando.

Necessariamente, os pontos nodais 5 e 89 devem estar pré definidos.

22-Palavra chave: QUADRO\_ESPESSURA

-Parâmetros : Espessura, ponto nodal extremo superior esquerdo e ponto nodal extremo superior direito

-Exemplo : quadro\_espessura 2.0 34 57

-Análise : Todos os elementos cujos pontos médios estiverem internos ao retângulo de vértice superior esquerdo definido pelo ponto 34, e vértice inferior direito definido pelo ponto 57, terão espessura igual a 2.

Os vértices superior esquerdo e inferior direito do retângulo, devem estar pré definidos (no exemplo, 34 e 57).

Deve-se observar que os comandos "QUADRILATERO\_QUAD" e "QUADRILATERO TRIANG" geram blocos de elementos sem a definição de espessura dos mesmos. Pretende-se com isso, desvincular-se a geração da malha de elementos finitos da definição de espessura dos mesmos, a qual pode ser eficientemente realizada com o comando "QUADRO\_ESPESSURA".

23-Palavra chave: REM

-Parâmetros : Não há

-Exemplo : REM METODO DOS ELEMENTOS FINITOS

-Análise : O comando "REM", define linha de comentário, assim, ao encontrá-lo em uma linha, o restante da mesma é desprezado. Tem como objetivo, facilitar a documentação dos problemas.

24-Palavra chave: RETIRE\_ELEM

-Parâmetros : Número do elemento a ser retirado

-Exemplo : retire\_elem 12

-Análise : O elemento 12 será retirado da malha. Haverá renumeração de elementos. No exemplo, ao ser

eliminado o elemento 12, o elemento 13 (se houver) será agora o número 12, e todos os elementos subsequentes (se houver) terão numeração decrementada de 1.

Se, no mesmo problema, for utilizado por mais de uma vez, o comando "RETIRE\_ELEM", deverá se observar a variação da numeração de elementos a cada utilização do mesmo.

25-Palavra chave: RIGIDEZ

-Parâmetros : Ponto nodal e rigidezes a serem incluídas nas direções vertical,  $\theta_x$  e  $\theta_y$ .

-Exemplo : Rigidez 5 10.0 20.0 30.0

-Análise : Esse comando destina-se, principalmente, à modelagem da influência da rigidez dos pilares vinculados ao pavimento. Assim, no exemplo, modelar-se-ia a influência do pilar ligado ao ponto nodal 5 da malha, pela inserção das rigidezes supostas do mesmo ( em unidades coerentes ) nas direções vertical,  $\theta_x$  e  $\theta_y$ .

26-Palavra chave: TITULO

-Parâmetros : Título do problema em processamento

-Exemplo : titulo PAVIMENTO TIPO EDIFICIO AVARE

-Análise : Atribui ao problema em processamento, o título : " Pavimento Tipo Edificio Avaré". O título será apresentado nas saídas produzidas pelo sistema.

27-Palavra chave: TRIANGULO

-Parâmetros : Número do elemento triangular e os três pontos nodais dos vértices (ciclicamente, no sentido horário) e número do material do elemento.

-Exemplo : triangulo 45 2 9 8 1

-Análise : Define um elemento triangular de número 45, com vértices 2, 9 e 8, e material número 1.

28-Palavra chave: TRIANGULOS

-Parâmetros : Número de elementos triangulares a serem lidos (mesma linha da palavra-chave), seguindo-se a definição de cada elemento (um elemento por linha)

-Exemplo : TRIANGULOS 2  
29 2 4 5 1  
60 90 102 88 2

-Análise : Define dois elementos triangulares, o elemento 29 (com vértices 2, 4, 5 e material 1) e o elemento 60 (com vértices 90, 102, 88 e material 2).

29-Palavra chave: VIGA

-Parâmetros : Nome da viga, número do material, momento de inércia à flexão e momento de inércia à torção, carga distribuída na viga, número de elementos finitos de viga e definição do número dos pontos nodais ao longo da viga.

-Exemplo : viga V11 1 1000.0 0.25 0.10 4 10 11 12  
13

-Análise : Define uma viga com nome "V11", entre os pontos 10 e 13, tendo a mesma, material de número 1, momento de inércia igual a 1000, momento de inércia à torção 0.25 e carga por unidade de comprimento 0.10 (unidade coerente com o restante da definição do problema).

30-Palavra chave : VIGA\_AUT

-Parâmetros: Nome da viga, número do material, momento de inércia à flexão e momento de inércia à torção, carga distribuída na viga, número de elementos finitos de viga, número do primeiro ponto nodal e incremento na numeração dos pontos.

-Exemplo : Viga\_aut 1 1000 0.25 0.10 4 10 1

-Análise : O comando apresentado no exemplo, desempenha

a mesma função do exemplo do comando "VIGA", com a vantagem da automatização da definição dos pontos nodais, já que os mesmos apresentam incremento constante de numeração.

#### 4.3.3.3 ERROS VERIFICÁVEIS PELO INTERPRETADOR

Quando o comando "GERE LISTING" é utilizado, o interpretador produz um arquivo com o mesmo nome do arquivo de dados e a extensão ".lst", contendo os erros encontrados.

Apresentam-se, a seguir, os tipos de erros verificáveis pelo interpretador :

1-Mensagem : CÓDIGO INVÁLIDO

-Interpretação : O comando utilizado pelo usuário não pertence à tabela de comandos válidos (apresentados no item anterior).

2-Mensagem : PROBLEMA DE LEITURA DE PARÂMETRO

-Interpretação : Os parâmetros exigidos pelo comando não foram fornecidos, ou o foram de maneira equivocada.

3-Mensagem : NÚMERO DE ELEMENTOS DEFINIDOS ESTÁ ERRADO

-Interpretação : Utilizou-se o comando "NÚMERO DE ELEMENTOS", e ao final da interpretação, o número de elementos definidos não é aquele fornecido pelo usuário.

4-Mensagem : NÚMERO DE PONTOS DEFINIDOS ESTÁ ERRADO

-Interpretação : O comando "NÚMERO DE ELEMENTOS" foi utilizado, e o número fornecido pelo usuário não é o número total de pontos verificado pelo interpretador.

5-Mensagem : ÁREA NEGATIVA OU NULA EM ELEMENTO

-Interpretação : Os pontos nodais do elemento foram definidos erradamente (por exemplo, três pontos

alinhados).

6-Mensagem : ELEMENTO INDEFINIDO

-Interpretação : Falta definição dos pontos nodais dos vértices em elemento

7-Mensagem : PONTO INDEFINIDO

-Interpretação : Falta definição de coordenadas cartesianas de ponto nodal.

8-Mensagem : ESPESSURA INDEFINIDA

-Interpretação : Falta definição de espessura em elemento.

9-Mensagem : MATERIAL INDEFINIDO

-Interpretação : Falta definição dos materiais utilizados.

#### 4.3.3.4 CRIAÇÃO DE NOVOS COMANDOS

Devido à simplicidade da estrutura de dados e do algoritmo do interpretador de comandos, é possível, a usuários com algum conhecimento em informática (especificamente linguagem Pascal), a criação de novos comandos.

Inicialmente, o usuário deve alterar o valor da constante determinadora do tamanho da tabela de comandos, incrementando-a de 1.

O próximo passo é a inserção na tabela de comandos, da palavra-chave do novo comando criado. A inserção deve respeitar a ordem alfabética, pois a busca dos comandos na tabela é binária.

Continuando, o usuário deve criar um procedimento correspondente ao comando implementado, que operando a estrutura de dados do interpretador, desempenhe a função por ele desejada.

Finalmente, é necessário inserir-se o nome do



procedimento criado, na lista de procedimentos executáveis pelo interpretador, tendo o mesmo, o número correspondente à posição da palavra-chave na tabela de comandos válidos

#### 4.4 EXEMPLO DE UTILIZAÇÃO DO INTERPRETADOR

##### 4.4.1 CARACTERIZAÇÃO DO PROBLEMA

Considere-se a utilização do interpretador de comandos para a discretização do pavimento de edifício em concreto armado da figura 4.3, encontrado em ANDRADE[4].

As dimensões do pavimento relevantes no exemplo, aparecem na figura 4.3, sendo complementadas pelas tabelas 4.1 e 4.2.

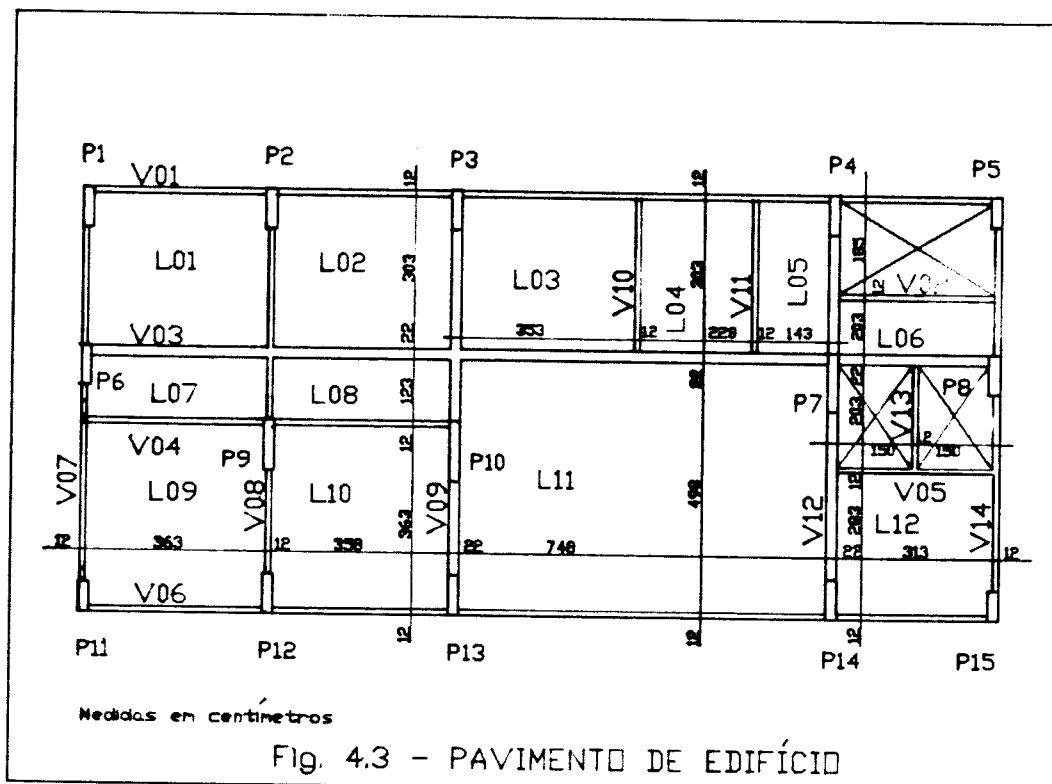


TABELA 4 1 - DADOS DAS LAJES

LAJE	ESPESSURA (m)	* CARGA TOTAL (KN/m <sup>2</sup> )
01	0.09	4.75
02	0.10	9.02
03	0.09	4.75
04	0.07	6.44
05	0.07	6.19
06	0.07	4.75
07	0.07	7.89
08	0.07	4.75
09	0.09	4.75
10	0.09	4.75
** 11	0.15	6.00
12	0.08	4.50

\* Inclui o peso próprio

\*\* Em ANDRADE[4], optou-se por laje nervurada. No presente exemplo, por simplificação, a laje será maciça

TABELA 4 2 - DADOS DAS VIGAS			
VIGA	SECÇÃO TRANSVERSAL (m x m)	* MOM. DE INÉRCIA (m <sup>4</sup> x 10 <sup>-5</sup> )	** CARGA (KN/m)
V01	0.12 x 0.60	2.20	9.00
V02	0.12 x 0.40	0.64	8.40
V03	0.22 x 0.60	4.00	10.50
V04	0.12 x 0.40	0.64	8.40
V05	0.12 x 0.30	0.27	8.10
V06	0.12 x 0.60	2.20	9.00
V07	0.12 x 0.60	2.20	9.00
V08	0.12 x 0.60	2.20	9.00
V09	0.22 x 0.60	4.00	10.50
V10	0.12 x 0.40	0.64	8.40
V11	0.12 x 0.40	0.64	8.40
V12	0.22 x 0.50	2.30	10.00
V13	0.12 x 0.30	0.27	8.10
V14	0.12 x 0.60	2.20	9.00

\* Só se considerou a inércia à flexão das vigas.  
\*\* Inclui o peso próprio.

#### 4.4.2 UTILIZAÇÃO DOS COMANDOS DO INTERPRETADOR NA DEFINIÇÃO DO PROBLEMA

Inicialmente, definem-se o título do problema e as características do material dos elementos estruturais, respectivamente pelo uso dos comandos "TITULO" e "MATERIAL".

Utilizando-se o comando "QUADRILATERO\_QUAD", gera-se uma malha com dimensões próximas às do pavimento, constituída de elementos quadrados de dimensão 0.5m x 0.5m, sem atribuição de espessura aos mesmos (figura 4.4).

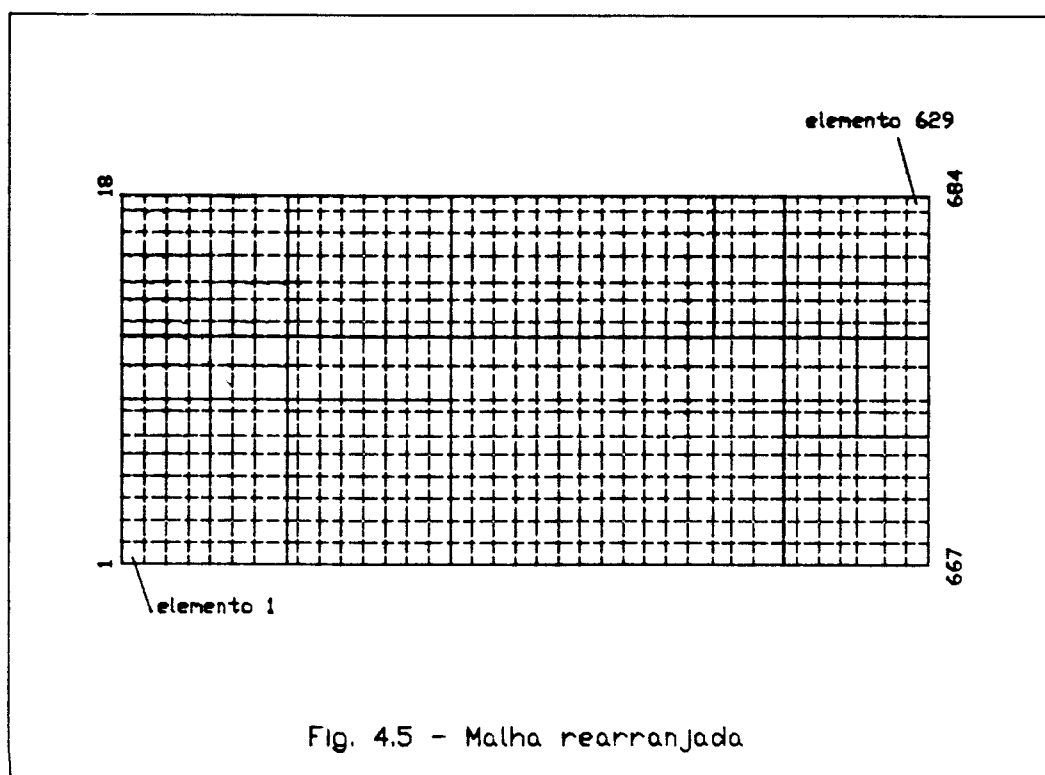
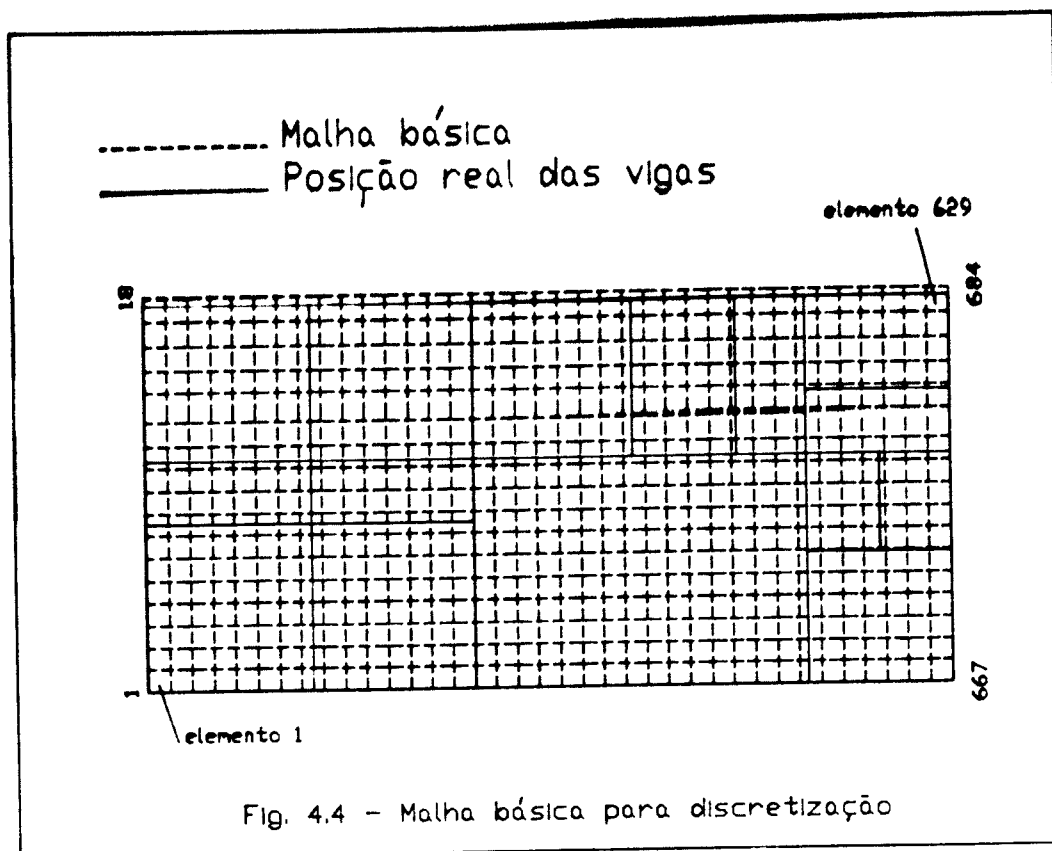
Como pode-se observar na figura 4.4, a malha básica gerada não coincide com as linhas dos eixos das vigas, assim, torna-se necessária a redefinição de algumas linhas através do comando "PONTOS AUTOMÁTICOS". O comando "PONTOS AUTOMÁTICOS" produz uma linha de pontos igualmente espaçados e, desta forma, se redefine-se uma linha paralela ao eixo x e então redefine-se uma linha paralela ao eixo y que intercepte a primeira, o ponto de interseção não mais necessariamente pertencerá à primeira linha, fato que requer a redefinição do referido ponto. A redefinição do ponto é feita pelo comando "PONTO". A figura 4.5 apresenta a malha básica rearranjada em função da posição das vigas.

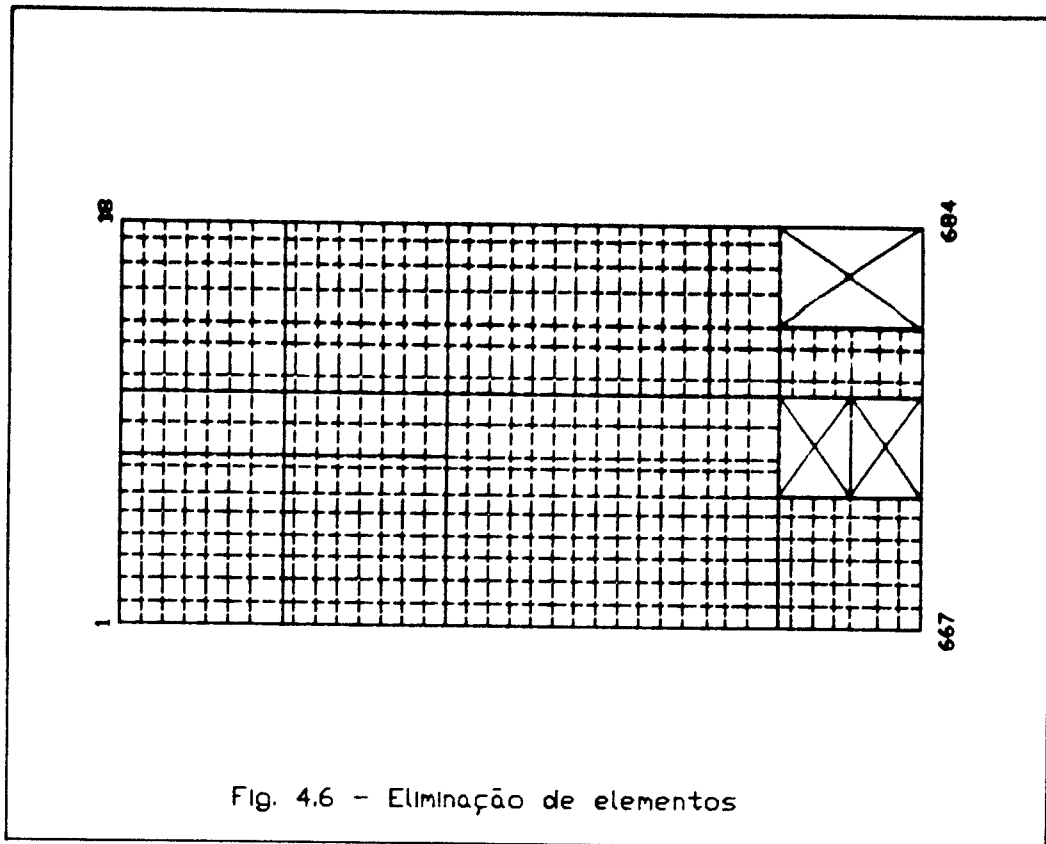
A inserção dos elementos de viga, efetivada pelo comando "VIGA AUT", é o passo seguinte da caracterização do problema.

Os pilares são modelados aproximadamente por pontos de deslocamento vertical nulo, definidos com o comando "PONTO VINCULADO".

A atribuição de espessura e carga às lajes se faz, respectivamente pelos comandos "QUADRO ESPESSURA" e "QUADRO CARGA".

Finalmente, o comando "QUADRO ELIMINAR" elimina os elementos de placa correspondentes às aberturas das lajes (figura 4.6).





#### 4.4.3 ARQUIVO DO USUÁRIO

O arquivo de dados completo, referente ao exemplo em questão, é apresentado a seguir.

```
rem ARQUIVO DE DADOS
rem
TITULO EXEMPLO
MATERIAL 1 2.7E7 0.2
rem
rem MALHA BASICA
rem
PONTOS 4
1 0.0 0.0
18 8.5 0.0
667 0.0 18.50
684 8.50 18.50
QUADRILATERO_QUAD 1 684 1 18 667 684 19 1
rem
rem LINHAS REDEFINIDAS
rem
PONTOS_AUTOMATICOS 18 127 1 0.0 3.75 8.5 3.75
PONTOS_AUTOMATICOS 18 397 1 0.0 11.20 8.5 11.20
PONTOS_AUTOMATICOS 18 487 1 0.0 13.60 8.5 13.60
```



VIGA\_AUI V8 1 2.2E-3 1.0E-15 17 9 0 127 1  
VIGA\_AUT V9 1 4.0E-3 1.0E-15 17 10 50 271 1  
VIGA\_AUT V10 1 6.4E-4 1.0E-15 7 8 4 407 1  
VIGA\_AUT V11 6.4E-4 1.0E-15 7 8.4 497 1  
VIGA\_AUT V12 2.3E-3 1.0E-15 17 10 0 541 1  
VIGA\_AUT V13 2.7E-4 1.0E-15 4 8 10 619 1  
VIGA\_AUT V14 1 2.2E-3 1.0E-15 17 9 0 667 1

rem

rem

PILARES

rem

PONTO\_VINCULADO 1 1 0 0  
PONTO\_VINCULADO 18 1 0 0  
PONTO\_VINCULADO 684 1 0 0  
PONTO\_VINCULADO 667 1 0 0  
PONTO\_VINCULADO 11 1 0 0  
PONTO\_VINCULADO 127 1 0 0  
PONTO\_VINCULADO 135 1 0 0  
PONTO\_VINCULADO 271 1 0 0  
PONTO\_VINCULADO 677 1 0 0  
PONTO\_VINCULADO 551 1 0 0  
PONTO\_VINCULADO 558 1 0 0  
PONTO\_VINCULADO 288 1 0 0  
PONTO\_VINCULADO 144 1 0 0  
PONTO\_VINCULADO 279 1 0 0  
PONTO\_VINCULADO 541 1 0 0

rem

rem

ESPESSURA E CARGA DE LAJES

rem

QUADRO\_ESPESSURA 0.09 1 279  
QUADRO\_CARGA 4.75 1 279  
QUADRO\_ESPESSURA 0.15 271 551  
QUADRO\_CARGA 6.00 271 551  
QUADRO\_ESPESSURA 0.07 9 281  
QUADRO\_CARGA 7.89 9 137  
QUADRO\_CARGA 4.75 135 281  
QUADRO\_ESPESSURA 0.09 11 144  
QUADRO\_CARGA 4.75 11 144  
QUADRO\_ESPESSURA 0.10 137 288  
QUADRO\_CARGA 9.02 137 288  
QUADRO\_ESPESSURA 0.09 281 414  
QUADRO\_CARGA 4.75 281 414  
QUADRO\_ESPESSURA 0.07 407 684  
QUADRO\_CARGA 6.44 407 504  
QUADRO\_CARGA 6.19 497 558  
QUADRO\_CARGA 4.75 551 680  
QUADRO\_ESPESSURA 0.08 541 673  
QUADRO\_CARGA 4.75 541 673

rem

rem

ABERTURAS NAS LAJES

rem

QUADRO\_ELIMINAR 554 684  
QUADRO\_ELIMINAR 547 677



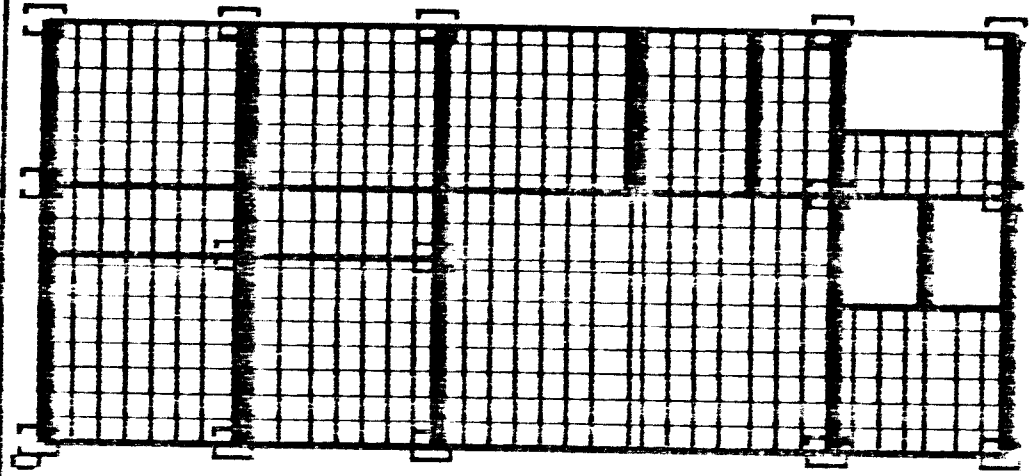
#### 4.4.4 CONFIRMAÇÃO GRÁFICA DA ENTRADA DE DADOS

O interpretador de comandos, ao final do processamento, apresenta em vídeo as principais características da malha de elementos finitos definida pelo usuário.

A confirmação gráfica, fornecida pelo interpretador no caso do exemplo em questão, é apresentada a seguir

INTERPRETADOR DE COMANDOS - ver 1.0  
Analyse de Placas Elasticas - M.E.Fin

### Exemplo



DADOS GEOM.

- △ 0
- 573
- 243
- 684
- + 816

LEGENDA :

- VINCULO
- VIGA
- B. LIVRE
- == APOIO.S
- ... ENGASTE

- CAPÍTULO V -

MONTAGEM ALTERNATIVA DA MATRIZ DE RIGIDEZ DA ESTRUTURA E  
PROCESSO UTILIZADO NA RESOLUÇÃO DO SISTEMA DE EQUAÇÕES  
LINEARES

5.1 INTRODUÇÃO

No desenvolvimento do sistema computacional para análise de pavimentos de edifícios, pelo método dos elementos finitos em microcomputador, as limitações de área de memória central desse tipo de equipamento, tornam-se mais evidentes nas etapas de montagem da matriz de rigidez da estrutura e resolução do sistema de equações lineares. Nessas etapas, ocorrem operações envolvendo matrizes de grandes dimensões e, portanto, o armazenamento das mesmas requer grande área de memória.

O tratamento de matrizes de grandes dimensões nos microcomputadores, pode ser realizado armazenando-se a referida matriz em dispositivos auxiliares de memória e, operando na memória central, somente a porção da mesma que é influenciada pela operação.

A limitação de área de memória central pode, portanto, ser solucionada pelo uso de dispositivos auxiliares de memória, os quais, normalmente caracterizados por grande disponibilidade de área de memória, porém, apresentando tempo de acesso grandemente superior ao da memória central do computador. Desta forma, este tipo de solução requer também, algoritmos cuidadosos quanto à minimização do tempo de processamento, sob pena de inviabilização do uso prático do sistema.

5.2 MONTAGEM ALTERNATIVA DA MATRIZ DE RIGIDEZ DA ESTRUTURA

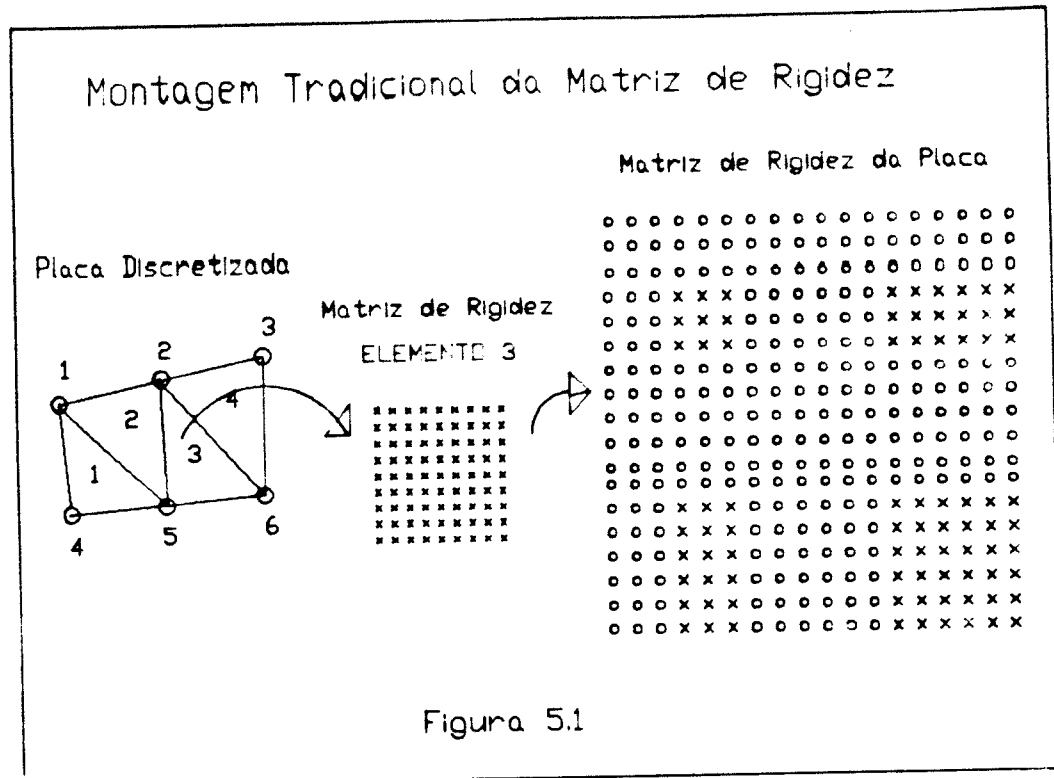
5.2.1 O PROCESSO TRADICIONAL

Refletindo a própria idéia do método dos

elementos finitos, o algoritmo usual de montagem da matriz de rigidez da estrutura, consiste na somatória de contribuições das matrizes de rigidez dos elementos finitos. Assim, as equações da matriz de rigidez de cada elemento finito são somadas às da matriz de rigidez da estrutura, correspondentes aos parâmetros nodais do referido elemento. Esse processo é denominado "usual" ou "tradicional" neste trabalho, por ser o algoritmo apresentado nas referências básicas do método dos elementos finitos, sendo particularmente encontrado em ZIENKIEWICZ[1] e SORIANO[5].

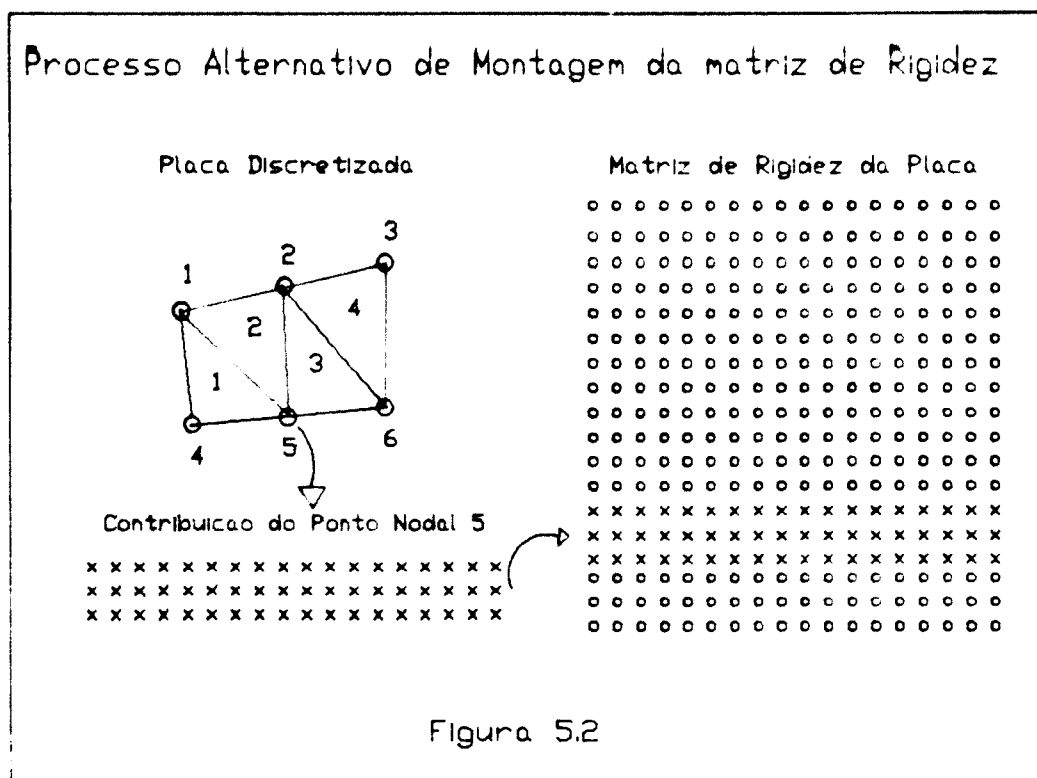
Exemplificando o processo usual de montagem da matriz de rigidez da estrutura, considere-se a figura 5.1, onde aparece uma placa discretizada em elementos finitos triangulares de três nós, localizados nos vértices e, três parâmetros por ponto nodal. Sendo o processo usual de montagem, baseado na somatória de contribuições dos elementos finitos, observe-se, por exemplo, a contribuição da matriz de rigidez do elemento finito 3, em nove linhas da matriz de rigidez da estrutura, sendo as nove linhas, correspondentes aos parâmetros dos pontos nodais 2, 5 e 6 da discretização efetuada. Desta forma, seriam necessárias quinze linhas consecutivas da matriz de rigidez da estrutura para conter a influência do elemento 3, ou, mais precisamente, seria necessário um número de linhas consecutivas, correspondente ao produto do módulo da maior diferença de numeração de pontos nodais no elemento pelo número de parâmetros nodais (no caso,  $5 \times 3$ ).

A montagem da matriz de rigidez em blocos pela contribuição de cada elemento finito, é, portanto, ineficiente, devido à influência dos elementos finitos na matriz de rigidez da estrutura poder abranger numerosas linhas e, assim, dificultar a geração de pequenos blocos, compatíveis com as limitações dos microcomputadores.



### 5.2.2 PROCESSO ALTERNATIVO DE MONTAGEM DA MATRIZ DE RIGIDEZ DA ESTRUTURA

A característica básica do processo alternativo de montagem da matriz de rigidez da estrutura é a geração de grupos de "n" linhas da matriz, sendo "n", o número de parâmetros nodais do elemento finito utilizado. Cada grupo de "n" linhas geradas, corresponde à contribuição de um ponto nodal, pela influência dos elementos finitos nele concorrentes. A figura 5.2, exemplifica o processo alternativo de montagem, apresentando a contribuição do ponto nodal 5, para o mesmo exemplo da figura 5.1.



### 5.2.3 ALGORITMOS PARA A MONTAGEM ALTERNATIVA DA MATRIZ DE RIGIDEZ DA ESTRUTURA

Antes de iniciar-se a montagem da matriz de rigidez pelo processo alternativo, é necessária a existência de uma estrutura de dados que informe quais elementos finitos concorrem em cada ponto nodal e, ainda, qual ponto nodal do elemento finito corresponde ao referido ponto. Nessa estrutura de dados, cada ponto nodal fica caracterizado pelo número de elementos finitos nele concorrentes e, por duas listas lineares contendo, respectivamente, o número de cada elemento finito concorrente e o número de cada ponto nodal do elemento, correspondente ao ponto nodal caracterizado. No exemplo da figura 5.2, considere-se, por exemplo, a seguinte definição dos elementos finitos 1,2 e 3 :

ELEMENTO FINITO	PONTO 1	PONTO 2	PONTO 3
1	4	1	5
2	5	1	2
3	2	6	5

Nesse caso, o ponto nodal 5, ficaria caracterizado da seguinte forma

Número de elementos concorrentes : 3

Elementos Concorrentes : 1, 2 e 3

Pontos nodais correspondentes : 3, 1 e 3

O algoritmo para a formação da estrutura de dados caracterizadora dos pontos nodais, é dado a seguir, em "pseudo código".

```
PARA i <— 1 até Número de pontos nodais
FAÇA
  Número de elementos concorrentes(i) <— 0
  PARA j <— 1 até Número de elementos finitos
  FAÇA
    k <— 1
    CONCORRENTE <— FALSO
    REPITA
      SE ponto k do elemento j = i
      ENTÃO
        Número de elementos concorrentes(i) <— Número de elementos concorrentes(i)+1
        Elemento concorrente(i, Número de elementos concorrentes(i)) <— j
        Ponto correspondente(i, Número de elementos concorrentes(i)) <— k
        CONCORRENTE <— VERDADEIRO
      k <— k+1
    ATÉ (k=Número de pontos nodais do elemento+1) OU
    (CONCORRENTE=VERDADEIRO)
```

Finalmente, o algoritmo responsável pela formação das linhas da matriz de rigidez da estrutura,

correspondentes a um ponto nodal "p" genérico, é dado a seguir, em "pseudo código". No algoritmo, NPAR é o número de parâmetros nodais dos elementos finitos utilizados. Deve-se observar que as linhas já são geradas no arranjo em banda, ou seja, devido à simetria da matriz de rigidez, só são armazenados os elementos de cada linha, a partir da diagonal principal.

```
PARA i<— 1 até Número de elementos concorrentes(p,i)
FAÇA
  ELEM<— elementos concorrentes(p,i)
  j<— Ponto correpondente(p,i)
  LINHA BASE<— NPAR*p-(NPAR-1)
  PARA k<— 1 até Número de pontos do elemento
  FAÇA
    k1<— pontos(elem,k)
    COLUNA<— NPAR* k1-(NPAR -1)
    PARA l<— 1 até NPAR
    FAÇA
      LINHA LOCAL<— NPAR*(j-1)+1
      LINHA GLOBAL<— (LINHA BASE-1)+1
      PARA m<— 1 até NPAR
      FAÇA
        COLUNA LOCAL<— NPAR*(j-1)+m
        COLUNA GLOBAL<— (COLUNA-1)+m
        DIFERENÇA<—COLUNA GLOBAL-LINHA GLOBAL
        +1
        SE DIFERENÇA > 0
        ENTÃO
          MATRIZ DA ESTRUTURA( LINHA GLOBAL,
          DIFERENÇA )= MATRIZ DA ESTRUTURA(
          LINHA GLOBAL,DIFERENÇA)+MATRIZ DO
          ELEMENTO(LINHA LOCAL,COLUNA LOCAL)
```

A montagem da matriz de rigidez da estrutura em blocos de linhas consecutivas, torna-se extremamente simplificada, ao se utilizar o processo alternativo. Por exemplo, no caso da figura 5.2, para obterem-se as "3n" primeiras linhas da matriz, basta aplicar-se o processo alternativo aos "n" primeiros pontos nodais.



#### 5.2.4 IMPOSIÇÃO DE CONDIÇÕES DE CONTORNO PARALELAMENTE À GERAÇÃO DA MATRIZ DE RIGIDEZ DA ESTRUTURA

No sistema computacional desenvolvido, a imposição de parâmetros nodais nulos ao problema é feita, eliminando-se a linha e a coluna correspondentes ao referido parâmetro nodal; além disso, é também eliminada a linha correspondente do vetor de cargas nodais.

Considere-se a matriz de rigidez da placa da figura 5.2. Para eliminarem-se, por exemplo, a linha e a coluna número 5 da mesma, será necessário o deslocamento de todas as linhas de numeração superior a 5, de uma posição para a esquerda e, analogamente, efetuar-se o deslocamento de todas as linhas de numeração superior a 5, de uma posição para cima. Concluída a eliminação da linha e da coluna, a matriz de rigidez, inicialmente de dimensão  $N \times N$ , passa a ter dimensão  $(N-1) \times (N-1)$ , facilitando o armazenamento da mesma e minimizando o tempo de processamento na resolução do sistema de equações lineares.

O processo de eliminação de linhas e colunas não deve, no entanto, partir da matriz já montada, o que geraria, inicialmente, a necessidade de armazenamento de toda a matriz de rigidez. A eliminação pode ser feita, paralelamente à montagem da matriz de rigidez da estrutura, através da inserção dos elementos na posição final que os mesmos ocupariam, após a eliminação das linhas e colunas desejadas.

Na implementação realizada, a cada índice da matriz, corresponde um outro, denominado "posição efetiva", o qual traz a posição final do mesmo, após a imposição das condições de contorno. Quando um índice refere-se a uma linha ou coluna eliminados, sua "posição efetiva" é zero. Assim, durante a montagem da matriz de rigidez da estrutura, se um dos índices do termo a ser

inserido na mesma, tiver posição efetiva igual a zero, a inserção não é feita, pois o termo corresponderia a uma linha ou coluna eliminados da matriz. Se nenhum dos dois índices do termo a ser inserido tiver posição efetiva igual a zero, o mesmo será inserido nas correspondentes posições efetivas.

O algoritmo para a formação da lista de posições efetivas é extremamente simples, sendo apresentado a seguir.

```
DESCONTO ← 0
PARA i ← 1 até Número de parâmetros da estrutura
FAÇA
  SE PARAMETRO(i) = 0
  ENTÃO
    POSIÇÃO EFETIVA (i) ← 0
    DESCONTO ← DESCONTO + 1
  CASO CONTRÁRIO
    POSIÇÃO EFETIVA(i) ← 1 - DESCONTO
```

### 5.3 RESOLUÇÃO DO SISTEMA DE EQUAÇÕES LINEARES

#### 5.3.1 INTRODUÇÃO

Após a etapa de montagem, a matriz de rigidez da estrutura fica armazenada em disco, particionada em blocos de linhas consecutivas. Basicamente, o processo desenvolvido para a etapa de resolução do sistema de equações lineares, parte de um algoritmo de resolução que supõe a matriz estar toda armazenada na memória central do computador. Cada referência, no algoritmo, a termos da matriz de rigidez, é substituída por uma chamada a um algoritmo de busca ao bloco no qual está o termo referenciado. Esse algoritmo, inicialmente, localiza o bloco que contém o termo, carregando-o, em seguida, na memória central do computador, e, permitindo assim, a operação dos termos do bloco e, em particular, a operação do termo referenciado. Finalmente, o bloco carregado, é novamente armazenado em disco, liberando a memória

central. Por exemplo, uma atribuição, no algoritmo de resolução, a determinado termo da matriz, é substituída por uma chamada a um algoritmo que realiza essa atribuição ao referido termo, porém tendo antecipadamente, a tarefa de localizar o bloco no qual está o termo e, além disso, carregá-lo na memória central.

Da maneira como foi apresentada, a execução do algoritmo básico do processo, descrito no parágrafo anterior, implica em duplo acesso ao disco (leitura e gravação do bloco), para cada referência a um termo da matriz. Evidentemente, no caso da resolução de sistemas de equações lineares, onde é grande o número de operações envolvendo termos da matriz, o processo descrito é altamente ineficiente quanto ao tempo de processamento. Assim sendo, a idéia básica apresentada, requer otimização, com vistas à minimização de acessos ao disco e, conseqüentemente, minimização do tempo de processamento total.

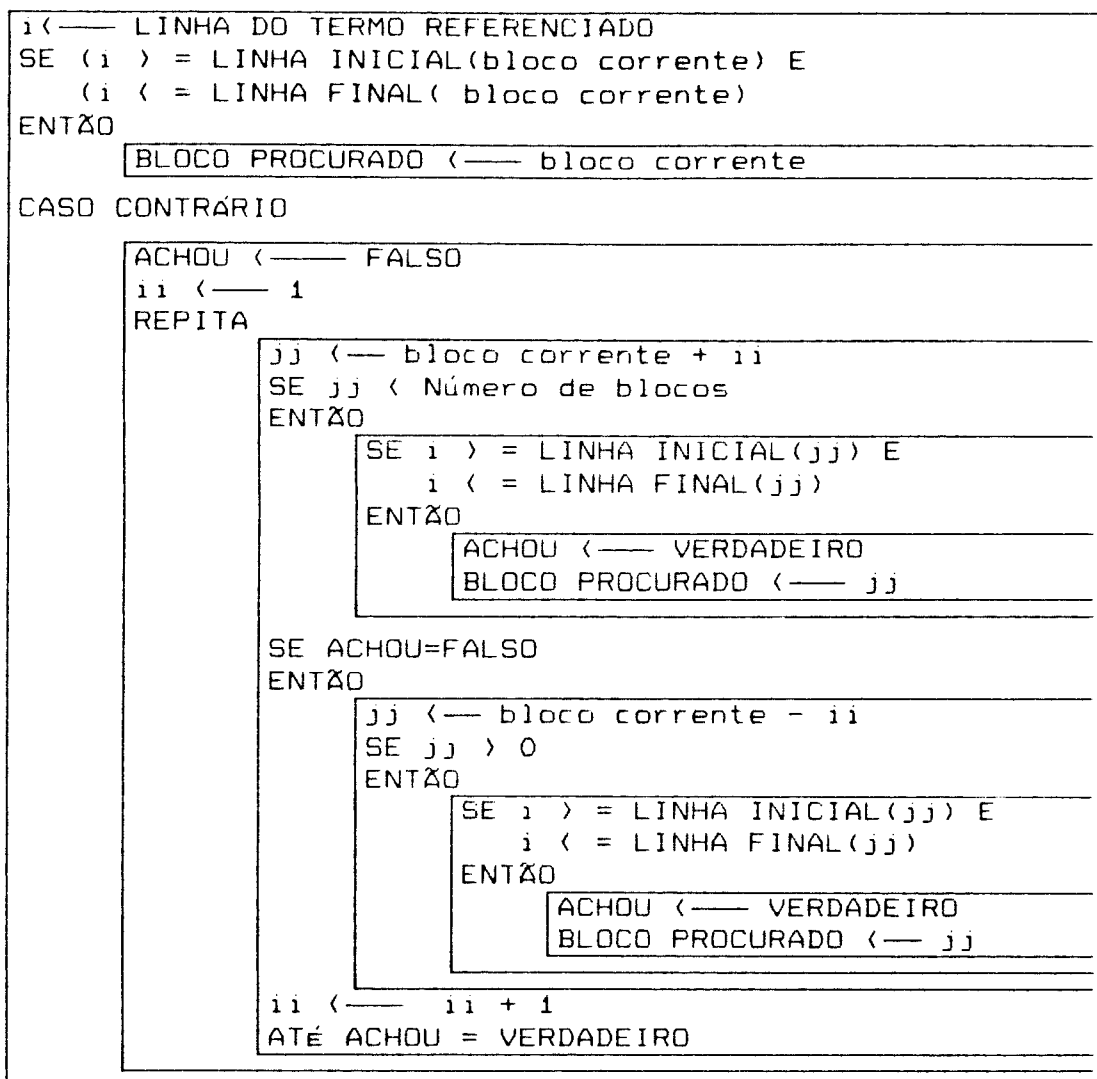
A necessidade de otimização do algoritmo básico de solução quanto ao tempo de processamento, levou ao desenvolvimento de um algoritmo que utiliza intensamente, a área de memória central, carregando nela, o número máximo possível de blocos. Desta forma, quando se faz necessário o acesso a determinado termo da matriz de rigidez, pode ocorrer do bloco que contém o termo referenciado estar carregado na memória, dispensando a necessidade do acesso ao disco. No caso do termo referenciado não pertencer a nenhum dos blocos da memória, o bloco carregado, referenciado há mais tempo é gravado e, em sua área, é carregado o bloco que contém o termo solicitado.

### 5.3.2 ALGORITMOS AUXILIARES PARA A RESOLUÇÃO DO SISTEMA DE EQUAÇÕES LINEARES

Conforme apresentado no item 5.3.1, o processo de

resolução do sistema de equações lineares, parte de um algoritmo que supõe a matriz de rigidez estar toda carregada na memória central e, efetua buscas aos termos localizados nos blocos armazenados em disco, a cada referência aos termos da matriz, no algoritmo básico

O primeiro algoritmo da busca é o que localiza em qual bloco está localizado o termo referenciado. Esse algoritmo, utiliza uma estrutura de dados que indica a primeira e a última linha de cada bloco. O último bloco operado na memória, é chamado "bloco corrente", e o bloco que contém o termo referenciado é o "bloco procurado". O primeiro algoritmo de busca é apresentado a seguir.



O segundo algoritmo tem por finalidade estabelecer qual bloco será retirado da memória, caso a mesma esteja carregada com o número máximo de blocos, e seja referenciado um termo contido em um dos blocos do disco. Quando um bloco é carregado na memória, o seu número é inserido numa fila de tamanho igual ao número máximo de blocos que podem ocupar a memória central. O último elemento da fila, refere-se ao bloco carregado há mais tempo, dentre os elementos da mesma, e, como critério, será o bloco retirado da memória. O algoritmo de inserção na fila, é apresentado a seguir, em "pseudo código".

```
PARA k ← 1 até Tamanho da fila
FAÇA
  m ← Tamanho da fila - k
  FILA(m+1) ← FILA(m)
FILA(1) ← NOVO BLOCO NA MEMÓRIA
```

O terceiro algoritmo mostra as etapas para acesso a um item genérico de índices "i" e "j", da matriz de rigidez. Esse algoritmo, utiliza uma estrutura de dados que indica a presença ou não, de determinado bloco na memória; além disso, usa os dois algoritmos anteriores. O terceiro algoritmo é apresentado a seguir.

```
BUSCA DO BLOCO QUE CONTEM A LINHA "i" (algoritmo 1)
SE CARREGADO (BLOCO PROCURADO)= FALSO
ENTÃO
  SE MEMÓRIA LIVRE ( ÁREA DE UM BLOCO
  ENTÃO
    MENOS UTILIZADO (— FILA (tamanho da fila)
    GRAVE EM DISCO (MENOS UTILIZADO)
    CARREGADO (MENOS UTILIZADO) (— FALSO)
  INSIRA NA FILA (BLOCO PROCURADO) (algoritmo 2)
  CARREGUE NA MEMÓRIA (BLOCO PROCURADO)
  CARREGADO (BLOCO PROCURADO) (— VERDADEIRO
BLOCO CORRENTE (— BLOCO PROCURADO
OPERE O TERMO NA ÁREA DE BLOCO PROCURADO
```

Com a utilização do algoritmo 3, o bloco que contém o termo desejado da matriz fica carregado na memória central. Apesar de conter determinado número de linhas consecutivas da matriz de rigidez, cada bloco é uma lista linear. O arranjo linear do bloco é feito para que se tenha melhor aproveitamento de sua área, para diferentes larguras de semi banda do sistema de equações lineares. Desta forma, para ter-se acesso ao termo de índices "i" e "j" da matriz de rigidez, depois de processado o algoritmo 3, dever-se-á acessar o termo "k" da lista linear (bloco), sendo k, dado por:

$$k = (i - \text{LINHA INICIAL}(\text{BLOCO PROCURADO})) * \text{BANDA} + j.$$

onde "BANDA" é a largura de semi banda do sistema de equações lineares.

### 5.3.3 ALGORITMO BÁSICO UTILIZADO

Conforme apresentado nos itens 5.3.1 e 5.3.2, o processo implementado para a etapa de resolução do sistema de equações lineares do sistema, requer um algoritmo

básico, além da utilização dos algoritmos de busca desenvolvidos. Na implementação realizada, utilizou-se como algoritmo básico, o "Método de Eliminação de Gauss", sendo porém, plenamente possível a utilização de outro algoritmo, complementado pelos algoritmos do item 5.3.2.

- CAPÍTULO VI -

EXEMPLOS DE UTILIZAÇÃO DO SISTEMA COMPUTACIONAL

6.1 INTRODUÇÃO

Neste capítulo, inicialmente são tratados exemplos referentes à análise de placas e da associação de placas e vigas, nos quais conhecem-se resultados teóricos provenientes da solução de equações diferenciais, ou mesmo resultados advindos da aplicação de métodos numéricos.

Ao final do capítulo, apresentam-se alguns resultados obtidos na análise do pavimento de edifício caracterizado no quarto capítulo (item 4.4.1).

6.2 EXEMPLO 1 : PLACA QUADRADA SIMPLEMENTE APOIADA, COM CARREGAMENTO UNIFORMEMENTE DISTRIBUÍDO.

Na resolução desse problema, devido à dupla simetria existente, discretizou-se apenas um quarto da área da placa, utilizando-se quatro malhas distintas de elementos finitos quadrangulares (figura 6.1)

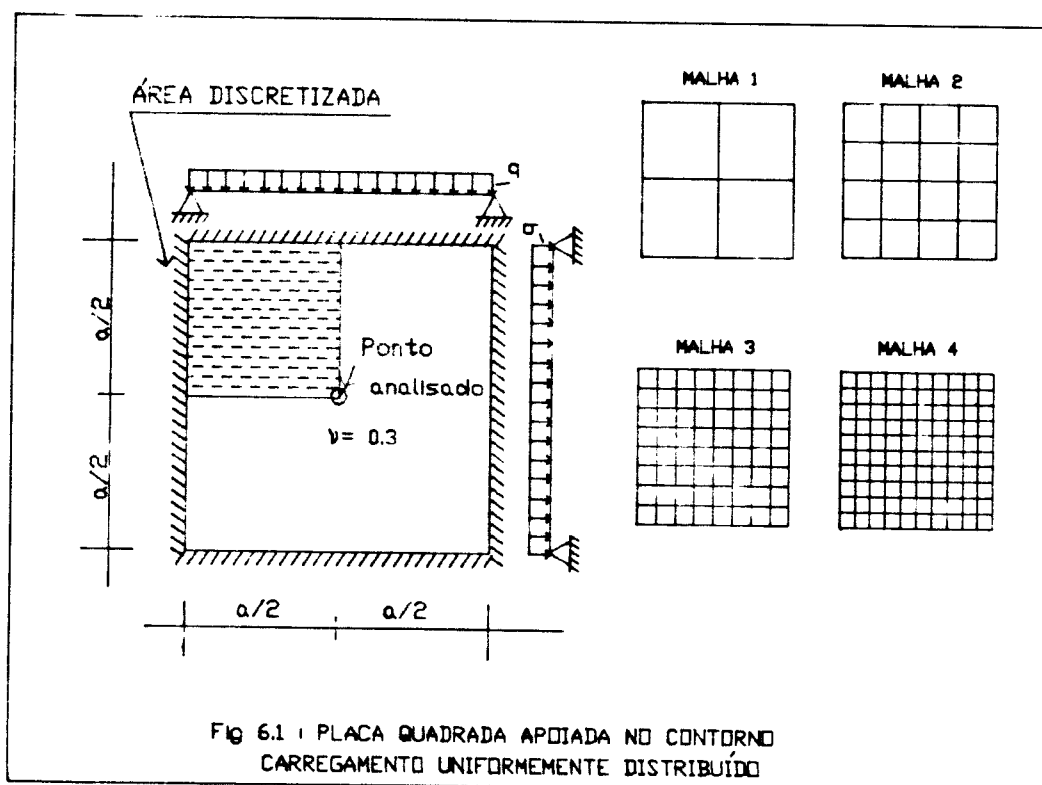
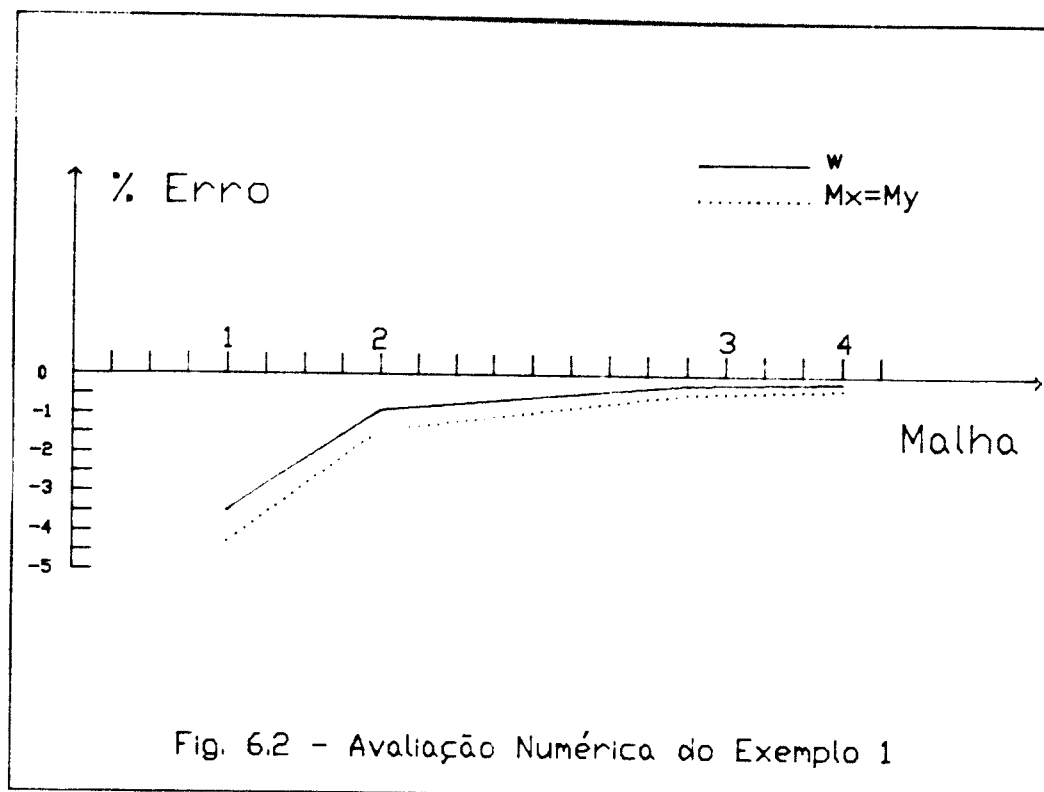


Fig 6.1 | PLACA QUADRADA APOIADA NO CONTORO  
CARREGAMENTO UNIFORMEMENTE DISTRIBUÍDO



A comparação dos resultados obtidos para deslocamento vertical e momentos fletores no centro da placa, em relação a resultados teóricos extraídos de TIMOSHENKO[6], é apresentada na figura 6.2



### 6.3 EXEMPLO 2 : PLACA QUADRADA ENGASTADA NO CONTORNO, COM CARREGAMENTO UNIFORMEMENTE DISTRIBUÍDO.

Analogamente ao exemplo 1, no tratamento desse problema, são feitas quatro discretizações em um quarto da área da placa (figura 6.3). São analisados o deslocamento vertical, momentos fletores no centro da placa (A) e os momentos fletores no ponto médio de um dos lados (B).

A comparação entre os resultados obtidos e os teóricos [6], aparece na figura 6.4.

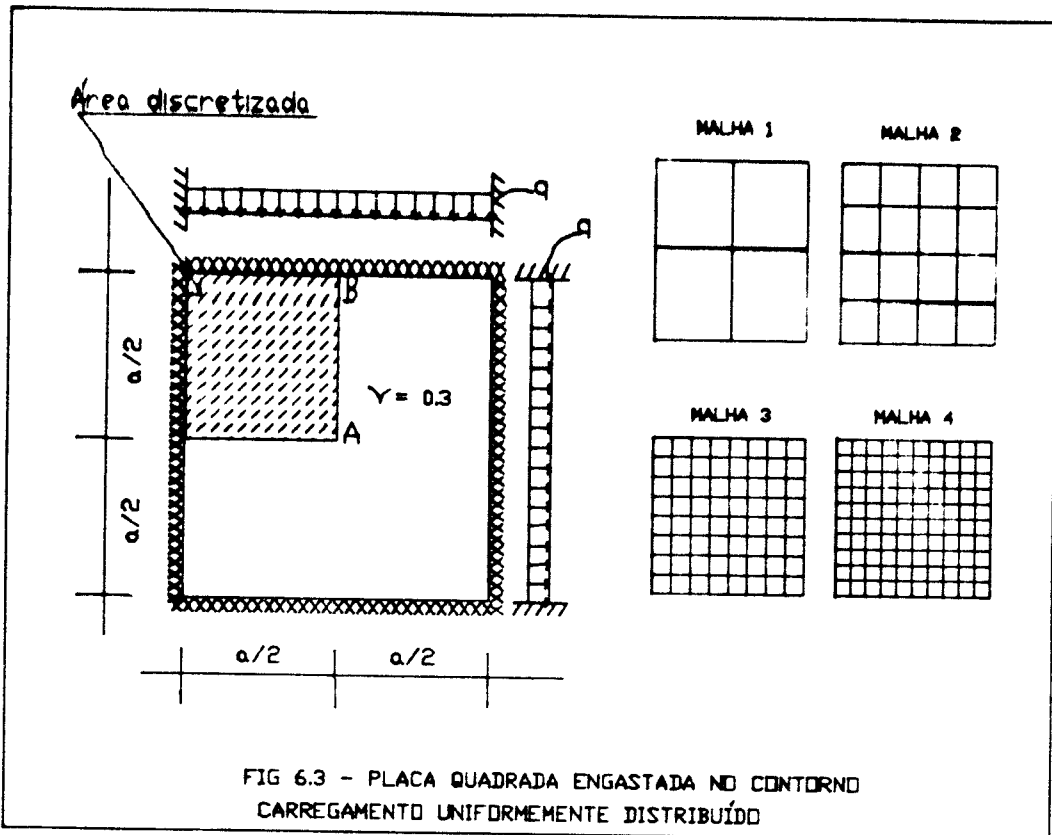


FIG 6.3 - PLACA QUADRADA ENGASTADA NO CONTOURO CARREGAMENTO UNIFORMEMENTE DISTRIBUÍDO

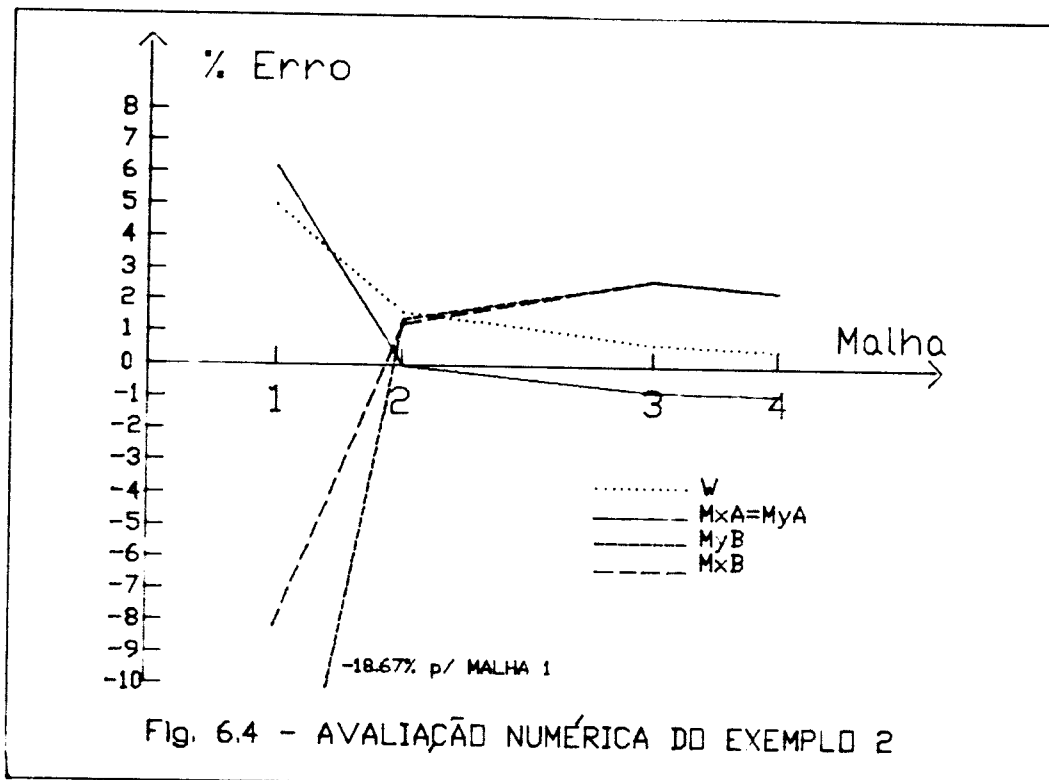
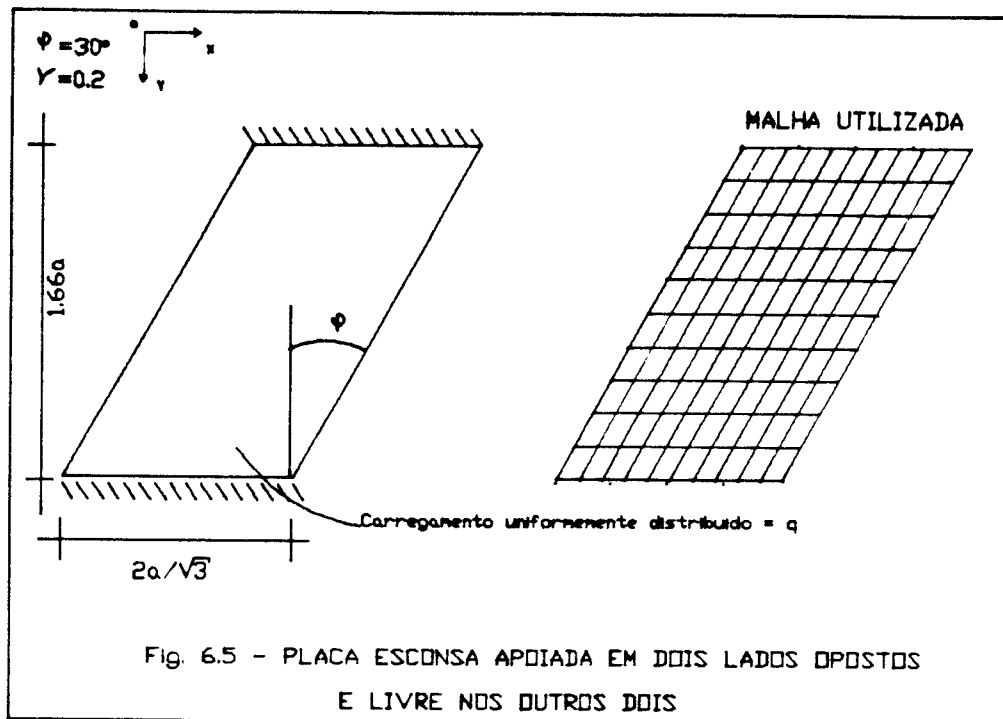


Fig. 6.4 - AVALIAÇÃO NUMÉRICA DO EXEMPLO 2

#### 6.4 EXEMPLO 3 : PLACA ESCONSA

Considere-se a placa esconsa de rigidez à flexão "D", apoiada em dois lados opostos e livre nos outros dois, submetida a carregamento uniformemente distribuído, conforme mostrado na figura 6.5. Nessa figura, indica-se também a malha utilizada na discretização do problema.



São analisados o deslocamento vertical e o momento fletor  $m_y$  no ponto médio da placa. Comparativamente, apresentam-se na tabela 6.1, resultados encontrados em PAIVA[7] para o mesmo problema em três diferentes abordagens numéricas: método dos elementos finitos via elemento T18, método dos elementos de contorno e diferenças finitas.

TABELA 6.1 - * ANÁLISE DE PLACA ESCONSA		
MÉTODO	DESL VERTICAL	my
RESPOSTA OBTIDA	0.1141	0.3606
** M. ELEM. CONTORNO	0.1129	0.3420
*** M. ELEM. FINITOS (T18)	0.1149	0.3593
DIFERENÇAS FINITAS	0.1183	0.3680
fator	$q a^4 / D$	$q a^2$

\* Valores obtidos para o centro da placa  
 \*\* Contorno dividido em 44 elementos  
 \*\*\* Domínio dividido em 184 elementos triangulares

#### 6.5 EXEMPLO 4 : PLACA QUADRADA APOIADA EM VIGAS ELÁSTICAS

Considere-se a placa de rigidez à flexão "D", apoiada em quatro vigas elásticas, conforme apresentado na figura 6.6. Desprezou-se a inércia à torção das vigas e o momento de inércia à flexão das mesmas "Ii" é dado por

$$I_1 = I_2 = I_3 = I_4 = \frac{5 a D}{E v}$$

A tabela 6.2 apresenta os resultados obtidos na análise do deslocamento vertical e dos momentos fletores  $m_x$  e  $m_y$  no ponto médio da placa. Apresentam-se também na mesma tabela, os valores teóricos para o problema, extraídos de TIMOSHENKO[6], e a variação percentual dos resultados obtidos em relação a estes.

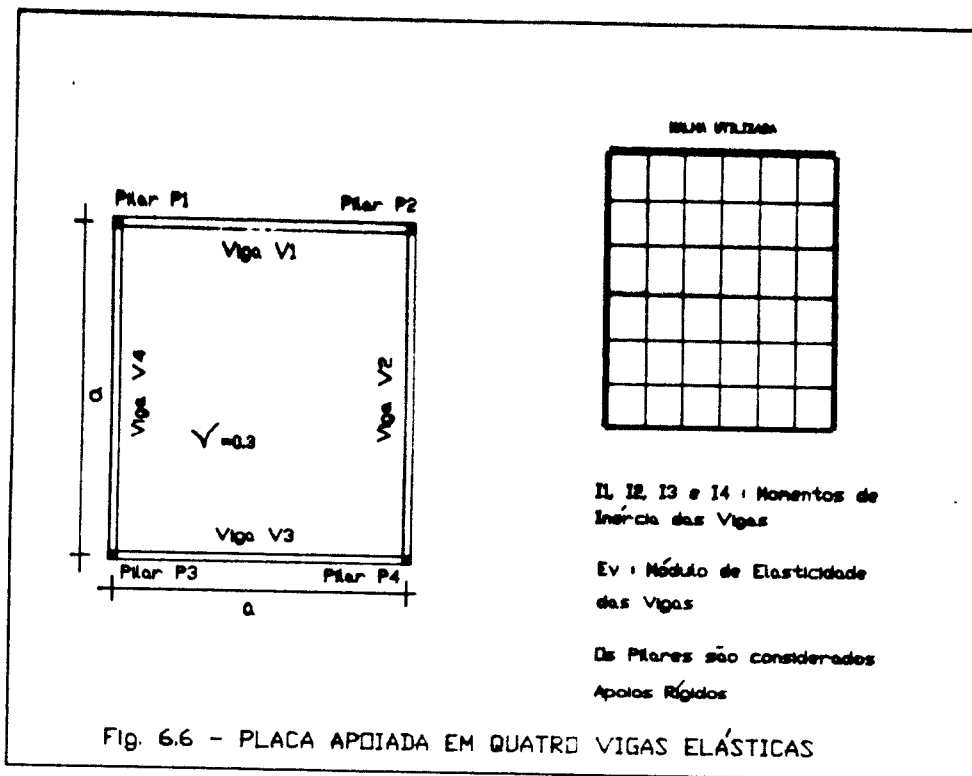


TABELA 6.2 - \* PLACA APOIADA EM VIGAS ELÁSTICAS

	Valor Obtido	Valor Teórico	Fator	Erro (%)
Desloc. vertical	$5.11 \cdot 10^{-3}$	$5.19 \cdot 10^{-3}$	$qa^4/D$	-1.54
Momento $m_x$	$5.07 \cdot 10^{-2}$	$4.94 \cdot 10^{-2}$	$qa^2$	2.63
Momento $m_y$	$5.07 \cdot 10^{-2}$	$4.94 \cdot 10^{-2}$	$qa^2$	2.63

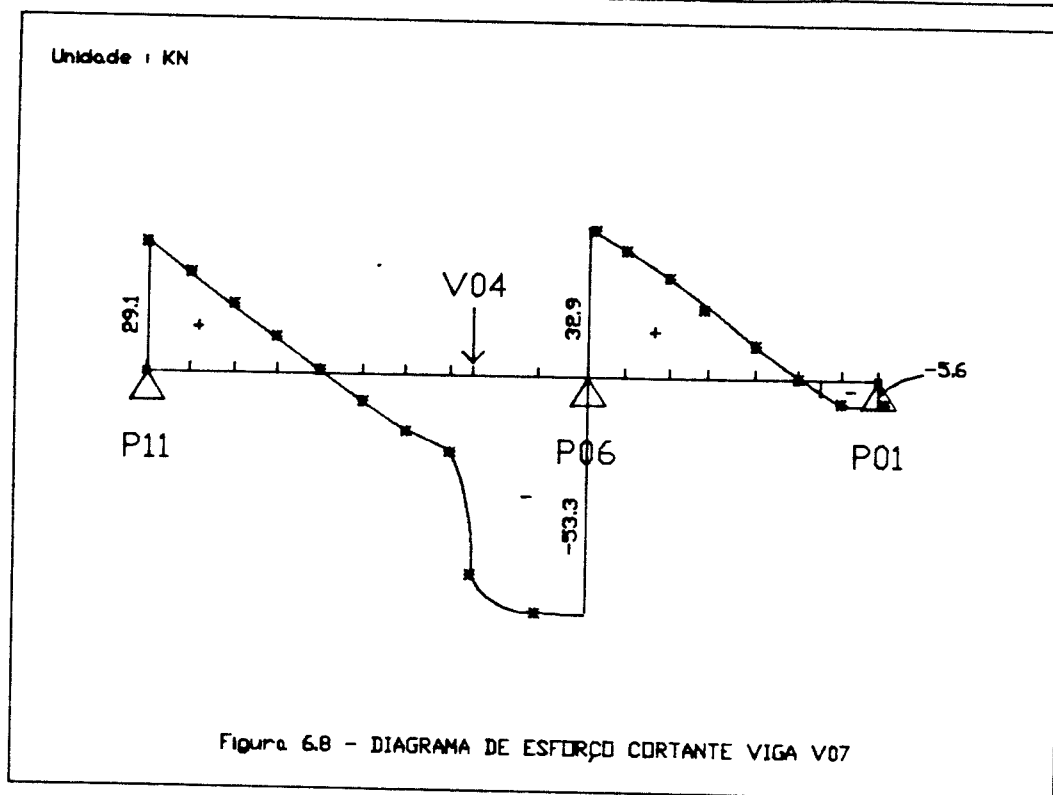
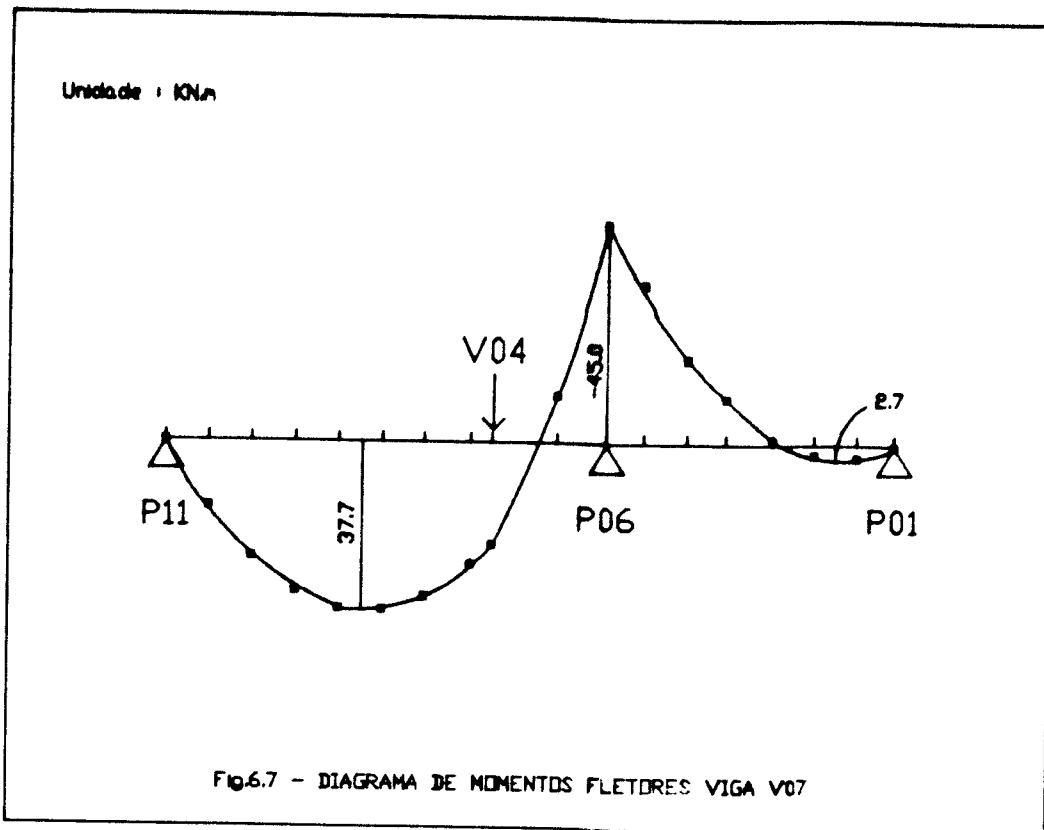
\* Valores referentes ao centro da placa

#### 6.6 - EXEMPLO 5 : ANÁLISE DE PAVIMENTO DE EDIFÍCIO

No quarto capítulo exemplifica-se a aplicação do interpretador de comandos, utilizando-o na discretização de um pavimento de edifício (item 4.4). Neste exemplo, apresentam-se alguns dos resultados obtidos pelo sistema, na análise do referido pavimento.

Inicialmente, é mostrado o diagrama de momentos fletores e esforços cortantes para uma das vigas (V07), traçado em função dos valores nodais fornecidos pelo

sistema (figuras 6.7 e 6.8). No exemplo em questão, não se



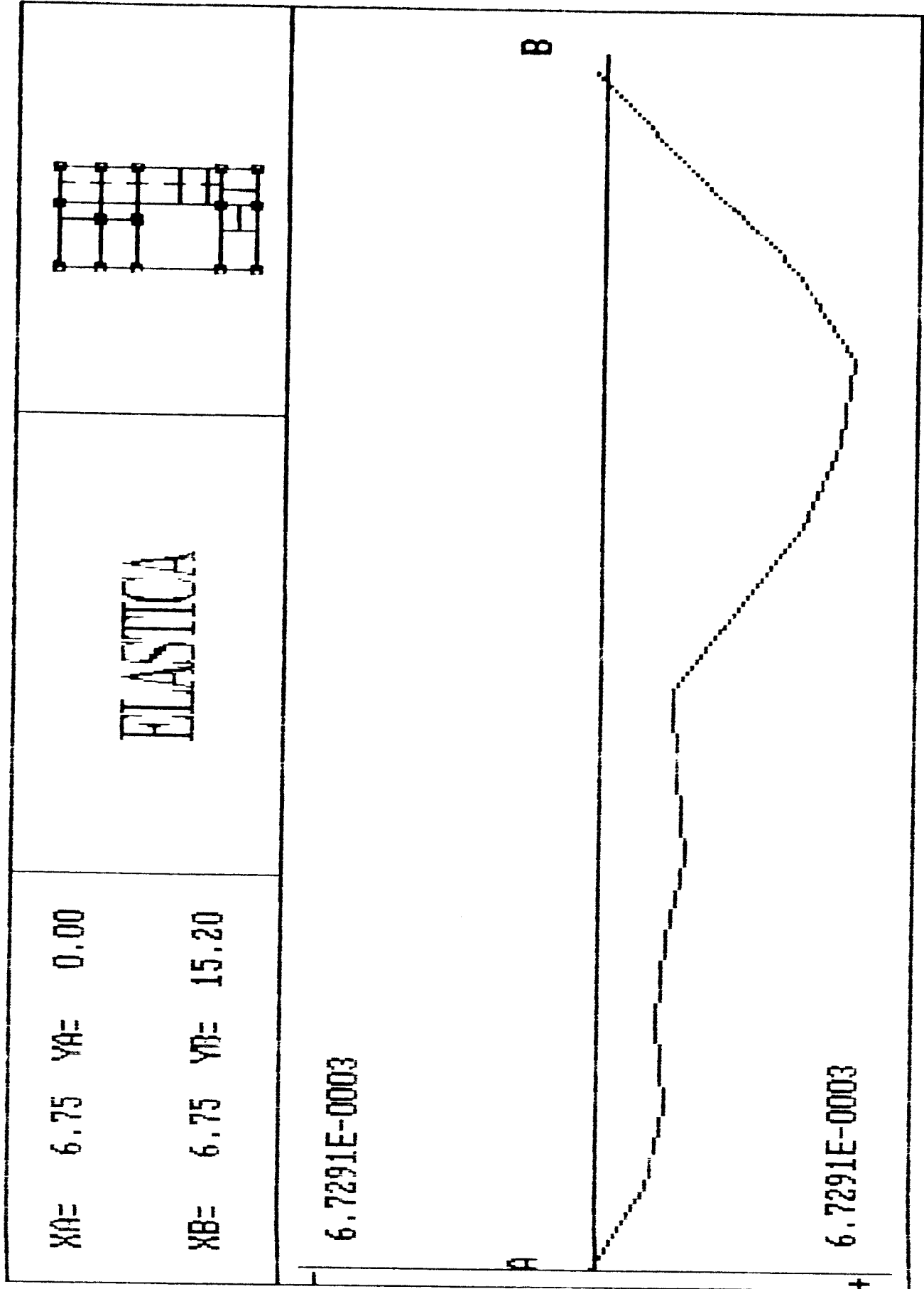
considerou a inércia das vigas à torção, e por esse

motivo, não é apresentado o diagrama de momentos torçores correspondente.

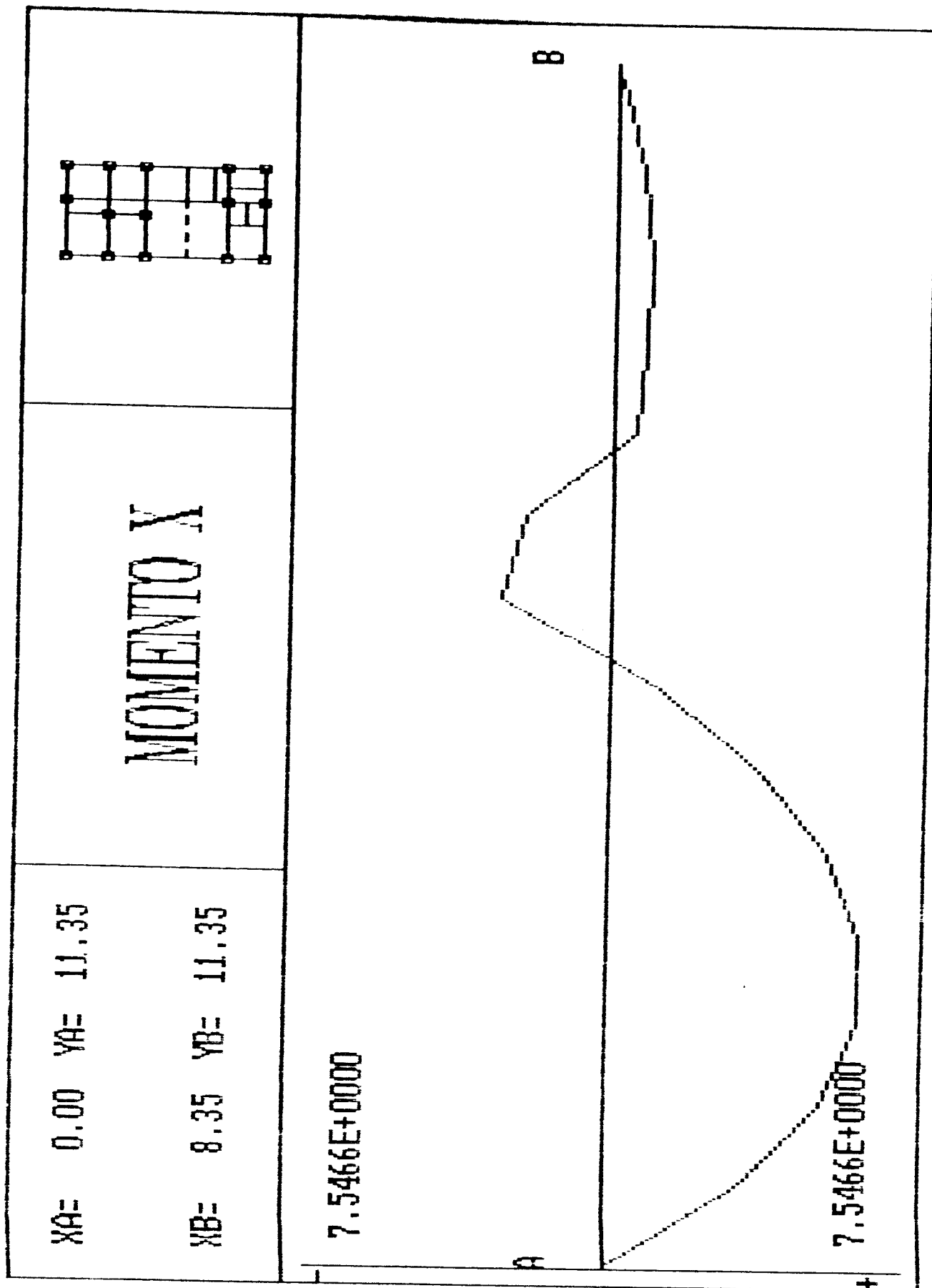
As cargas verticais nos pilares aparecem na tabela 6.3. Na modelagem do problema, os pilares foram considerados apoios rígidos com relação aos deslocamentos verticais. Se pretender-se uma análise mais próxima da realidade, pode-se usar o comando "RIGIDEZ" do interpretador de comandos, e assim, modelarem-se todos os pilares, substituindo-os por rigidezes ao deslocamento vertical e às rotações nas direções  $\theta_x$  e  $\theta_y$ , atuando no pontos nodais correspondentes aos pilares.

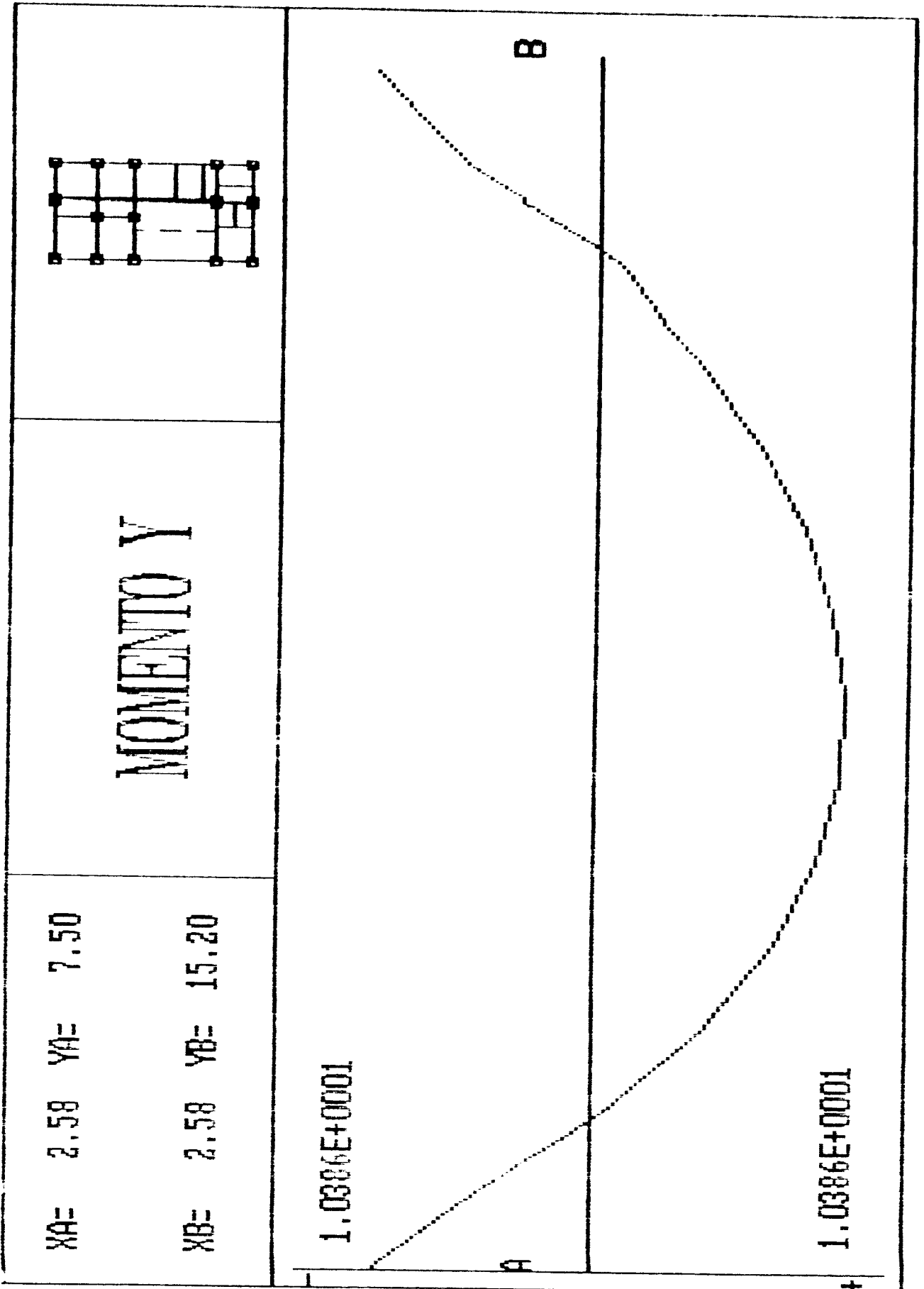
PILAR	CARGA VERTICAL (kN)
P01	26.6
P02	84.3
P03	186.0
P04	119.0
P05	-1.2
P06	131.0
P07	360.0
P08	60.0
P09	205.0
P10	340.0
P11	53.8
P12	48.8
P13	111.0
P14	165.0
P15	26.9

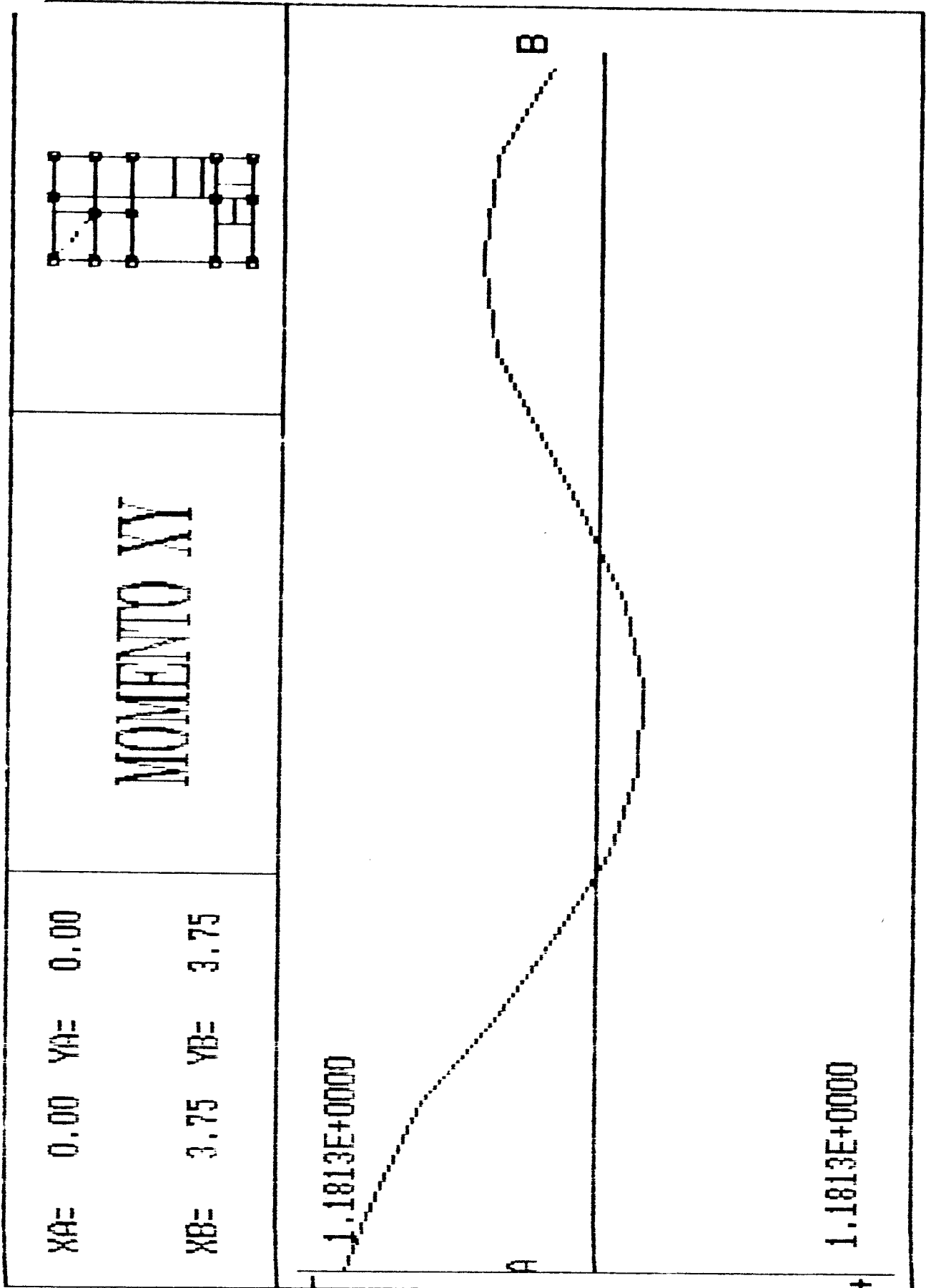
Finalmente, são apresentados os diagramas produzidos pelo módulo 5. Nesses diagramas, aparece em escala reduzida, uma representação do pavimento, contendo em linha tracejada, o segmento de reta correspondente ao diagrama.

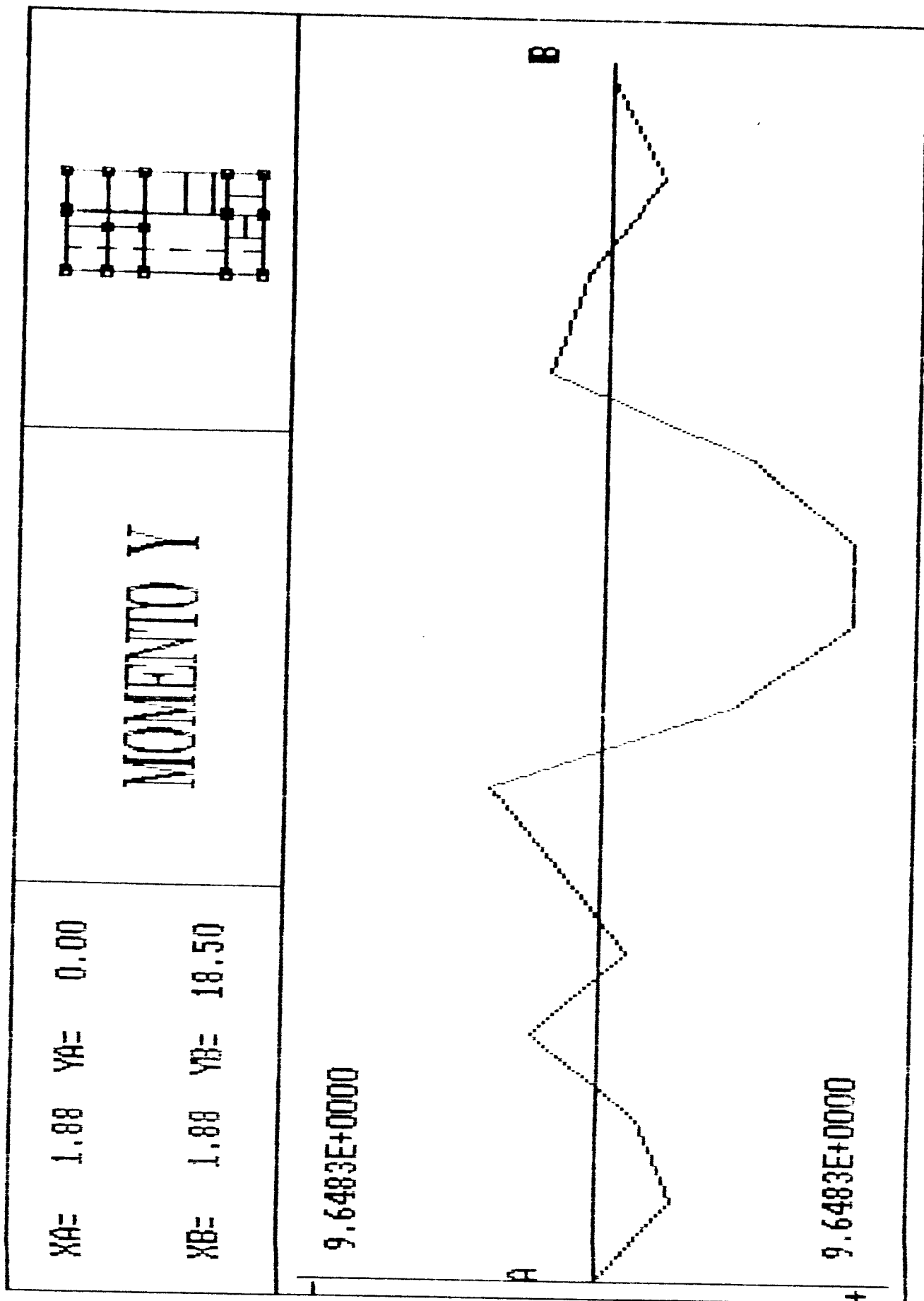












- CAPÍTULO VII -

CONCLUSÕES DO TRABALHO E SUGESTÕES PARA PROSSEGUIMENTO DE PESQUISA

Neste trabalho, abordaram-se basicamente três aspectos da análise de pavimentos de edifícios pelo método dos elementos finitos : a entrada de dados, a formulação dos elementos finitos utilizados e as peculiaridades da implementação voltada aos microcomputadores.

A busca da eficiência global do sistema, evidenciou a necessidade de um processo de entrada de dados, que viesse por tornar menos complexa a comunicação entre o usuário e o sistema. Para isso, desenvolveu-se um interpretador de comandos que transforma informações compactas do usuário, em dados numéricos processáveis pelo sistema. O estudo das características dos dados em tais sistemas, com a posterior identificação das etapas automatizáveis do processo de entrada de dados, e ainda a escolha das estruturas de dados mais adequadas, parece ser área interessante para o prosseguimento da pesquisa.

O sistema computacional não deve, em princípio, requerer conhecimentos profundos do método dos elementos finitos por parte do usuário e, desta forma, elementos finitos com eficiência numérica em situações genéricas são desejáveis, pois garantem respostas precisas, mesmo em discretizações menos adequadas ao problema. Dessa consideração, extraiu-se o critério para a escolha do elemento finito triangular DKT, na implementação do sistema.

A limitação da área de memória central dos microcomputadores, mostrou-se empecilho vultoso à implementação do sistema computacional, em decorrência do tratamento de matrizes de grandes dimensões ser inerente à utilização do método dos elementos finitos. Desenvolveram-se algoritmos voltados à implementação em microcomputadores, sendo os quais caracterizados pelo uso

combinado de dispositivos auxiliares de memória e da memória central do computador. A solução do problema da limitação de área de memória central pelo uso de dispositivos auxiliares de memória, gera como problema paralelo, o incremento no tempo de processamento, tornando imperativa a otimização dos algoritmos com vistas à minimização do número de acessos aos dispositivos auxiliares de memória, e desta forma, pesquisas nessa área têm notável aplicabilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ZIENKIEWICZ, O.C. The finite element method. Mc Graw Hill, 1967.
- [2] BATOZ, J.L; BATHE, K.J; HO, L.W. A study of three node triangular plate-bending element. International Journal for Numerical Methods in Engineering, (15), 1771, 1812, (1980).
- [3] JEYACHANDRABOSE, C; KIRKHOPE, J; RAMESH BABU, C. An alternative explicit formulation for the DKT plate-bending element. International Journal for Numerical Methods in Engineering, (21), 1289, 1293, (1985).
- [4] ANDRADE, J.R.L. Estruturas correntes de concreto armado - parte 1. São Carlos EESC-USP (apostila), 1982.
- [5] SORIANO, H.L. Sistemas de equações lineares em problemas estruturais. Lab. nacional de engenharia civil de Lisboa (apostila), Lisboa 1981.
- [6] TIMOSHENKO, S; WOINOWSKY-KRIEGER, S. Teoria de placas y laminas (Theory of plates and shells). Trad. F. J. Medina Salanova. Bilbao, España, Ediciones Urmo, 1970.
- [7] PAIVA, J. B. Formulação do método dos elementos de contorno para flexão de placas e suas aplicações em engenharia de estruturas. São Carlos, EESC-USP (Tese de doutorado, Orientador : Wilson Sérgio Venturini), 1987.
- [8] REISSNER, E. The Effect of transverse shear deformation on the bending of elastic plates. J. Appl. Mech, 12 : A-69, 1945.

BIBLIOGRAFIA

1. ANDRADE, J.R.L. Estruturas correntes de concreto armado - parte 1. São Carlos, EESC-USP (apostila), 1982.
2. BATOZ, J.L. An explicit formulation for an efficient triangular plate-bending element. International Journal for Numerical Methods in Engineering, (18), 1077, 1089 (1982).
3. BATOZ, J.L; BATHE, K.J; HO, L.W. A study of three node triangular plate-bending element. International Journal for Numerical Methods in Engineering, (15), 1771, 1812, (1980).
4. BREBBIA, C.A; FERRANTE, A.J. The finite element technique. Editora UFGS, Universidade Federal do Rio Grande do Sul, Porto Alegre, 1975.
5. DEGASPARE, J.C. Aplicação do elemento finito T-18 para flexão de placas delgadas elástico-isótropas. São Carlos, EESC-USP, 1975 (Dissertação de mestrado, Orientador : Walter Savassi).
6. GERE, J.M; WEAVER JR, W. Análise de estruturas reticulares. (Analysis of framed structures). Trad. Sérgio Vargas Romero. México, Companhia Editorial Continental, 1972.
7. JEYACHANDRABOSE, C; KIRKHOPE, J; RAMESH BABU, C. An alternative explicit formulation for the DKT plate-bending element. International Journal for Numerical Methods in Engineering, (21), 1289, 1293, (1985).



- 8 MONTANARI, I; MARTINELLI, D.A.O. Placas elásticas : equações gerais e placas retangulares. São Carlos EESC-USP (apostila), 1976.
- 9 PAIVA, J.B. Formulação do método dos elementos de contorno para flexão de placas e suas aplicações em engenharia de estruturas. São Carlos, EESC-USP (tese de doutorado, orientador : Wilson Sérgio Venturini), 1987.
10. PAIVA, J.B. Instabilidade de folhas planas com vinculações elásticas. São Carlos, EESC-USP (tese de mestrado, orientador : Munir Rachid), 1980.
- 11 PRESSMAN, R.S. Software engineering, a practitioner's approach. Mc Graw Hill, 1987.
12. REISSNER, E. The effect of transverse shear deformation on the bending of elastic plates. J. Appl. Mech, 12 : A-69, 1945.
- 13 SORIANO, H.L. Sistemas de equações lineares em problemas estruturais. Lab. nacional de engenharia civil de Lisboa (apostila), Lisboa 1981.
14. TIMOSHENKO, S; WOINOWSKY-KRIEGER, S. Teoria de placas y laminas (Theory of plates and shells). Trad. F. J. Medina Salanova. Bilbao, España, Ediciones Urmo, 1970.
15. ZIENKIEWICZ, O.C. The finite element method. Mc Graw Hill, 1967.