

FD-12/18

Sao Paulo, 1991

Orientador: Prof. Dr. Oscar Brito Augusto
Professor Assistente Doutor do Departamento de
Engenharia Naval da Escola Politécnica da Universidade
de Sao Paulo

Dissertação apresentada a Escola Politécnica
da Universidade de Sao Paulo para obtenção
do Título de Mestre em Engenharia.

**PRÉ E PÓS PROCESSAMENTO GRÁFICO DE DADOS
PARA O MÉTODO DOS ELEMENTOS FINITOS PARA
SISTEMAS BASEADOS EM MICROCOMPUTADORES**

Alexandre Kawano
Engenheiro, Escola Politécnica da Universidade de Sao Paulo - 1988

Aos homens de bem

AGRADECIMENTOS

Apresento os meus agradecimentos

ao Prof. Oscar Brito Augusto, pelo seu constante estímulo a minha carreira universitária, por sua orientação segura, e por sua amizade;

ao Prof. Carlos Alberto Nunes Dias, que com suas poucas palavras sábias muito colaborou para a execução deste trabalho;

ao Prof. Sérgio Gontijo, que com seu equilíbrio, soube transmitir a segurança tão necessária à atividade docente;

ao Prof. Celso Taniguchi, pelo seu apoio e incentivo à minha carreira universitária;

ao Prof. Luiz de Mello Bandeira Laterza pela confiança depositada em meu trabalho;

ao Prof. Racine Tadeu Araújo Prado, que através de sua filosofia tem me auxiliado muito nas horas difíceis;

ao Prof. Eduardo Ioshimoto pela sua amizade, simpatia e pelos seus sempre úteis conselhos;

ao Departamento de Engenharia de Construção Civil e Urbana da EPUSP;

ao Departamento de Engenharia Naval da EPUSP;

ao Centro de Estruturas Navais e Oceânicas da USP (CENOC).

Há alguns anos atrás, a área de estruturas do Departamento de Engenharia Naval e Oceânica da Escola Politécnica da USP possuía apenas um programa para pré e pós processamento de dados para o método dos elementos finitos, não interativo e baseado em um computador de grande porte (BURROUGHS 6900). Percebendo que, em se tratando de recursos computacionais gráficos, haviam se estacionado no tempo [OSC], e que os microcomputadores passaram a ser acessíveis, os Profs. Carlos Alberto Nunes e Oscar Brito Augusto decidiram criar mais uma nova linha de pesquisa que pudesse eliminar as suas deficiências na área gráfica. Seu seu primeiro fruto foi um simples programa pré e pós processador resultante de uma pesquisa de iniciação científica [ALE]. A presente dissertação faz parte de um trabalho de consolidação, apresentando as bases matemáticas do pré e pós processamento gráfico de dados para o método dos elementos finitos destinados a sistemas baseados em microcomputadores.

O trabalho apresenta os fundamentos matemáticos envolvidos em rotinas gráficas associadas à programação de pré e pós processamento de dados para o método dos elementos finitos, incluindo traçado de curvas de isotensão, remoção de linhas escondidas e desenho da estrutura deformada. Dois novos algoritmos, que levam em conta características das malhas de elementos finitos, são apresentados. Um é uma adaptação do algoritmo de Roberts para a remoção de linhas escondidas, e o outro se aplica no problema do traçado de curvas de isovalor.

O NAVGRAF, um programa gráfico para o auxílio das fases de pré e pós processamento de dados para o método dos elementos finitos, é gerado a partir das ideias apresentadas nesta dissertação.

ABSTRACT

The mathematical basis involved in graphical routines associated to pre and post processing of data for the finite element method, including stress level contour plotting, hidden lines removal and deformed structure display are discussed. Two new algorithms which take into account finite element meshes characteristics are presented. The first is an adaptation of the Roberts algorithm for hidden lines removal and the second is applied to the stress contour plotting problem.

From the ideas presented in this work, NAVGRAF, a computer graphics program which can be used to aid the pre and post processing of data for the finite element method phases is developed.

~	: Produto vetorial.	
.	: Produto escalar.	
a, θ, ϕ	: ângulo de rotação.	
$c = a + b \cdot j$: número complexo.	
j	: número imaginário, $j^2 = -1$.	
[T]	: Matriz de transformação linear.	
[Trans]	: Matriz de translação.	
$[\text{Rot}_z^\theta], [\text{Rot}_y^\theta], [\text{Rot}_x^\theta]$: Matriz de rotação em torno do eixo Oz.	
	: Matriz de rotação em torno do eixo Oy.	
	: Matriz de rotação em torno do eixo Ox.	
$[\text{Rot } \vec{e}_i]$: Matriz de rotação em torno de um eixo \vec{e}_i .	
[RX]	: Matriz de reflexão.	
[H]	: Matriz de homotetia.	
[L]	: Matriz 4x4 tal que $([P][L])[P]^t = 0$, com [P] um ponto qualquer dado em coordenadas homogêneas.	
[]	: Ponto dado em coordenadas homogêneas.	
()	: Ponto dado em coordenadas ordinárias.	
P, Q	: Ponto genérico.	
P_0, Q_0	: Ponto de posição inicial.	
x, y, z, w	: Coordenadas cartesianas.	
Ox, Oy, Oz	: Eixos cartesianos.	
$\delta x, \delta y, \delta z$: Componentes de vetor deslocamento.	
$(a, b, c), \vec{v}$: Vetor qualquer do \mathbb{R}^3 .	
\vec{e}_1	: Vetor de norma unitária.	
n	: Vetor normal.	

λ	: Escalar
θ, ϕ	: Ângulo de rotação.
α	: Plano de projeção; Quadro de projeção.
O	: Centro de projeção; Observador.
P_0	: Reta projetante; Raio visual.
d	: Distância entre o observador e o quadro de projeção.
D	: Distância entre o observador e a origem do sistema coordenado associado ao objeto.
l	: Reta que liga o observador à origem do sistema coordenado associado ao objeto.
x, y, z, w	: Coordenadas cartesianas.
Ox, Oy, Oz	: Eixos cartesianos.
[]	: Ponto dado em coordenadas homogêneas.
()	: Ponto dado em coordenadas ordinárias.
(x^o, y^o, z^o)	: Coordenadas da projeção de um ponto em relação ao sistema associado ao plano de projeção.
(x^o, y^o, z^o)	: Coordenadas de um ponto em relação ao sistema associado ao observador.
e_1^o, e_2^o	: Vetores
$r^o = (r_1^o, r_2^o, r_3^o)$: Vetor que dá a direção dos raios visuais no caso da perspectiva cavaleira.
[T]	: Matriz de transformação linear.
[A]	: Matriz cujas linhas são os vetores de uma base que gera um plano de projeção.
[Trans]	: Matriz de translação.
$[Rot^z \theta], [Rot^y \theta], [Rot^x \theta]$: Matriz de rotação em torno do eixo Oz.
$[Rot^y \theta], [Rot^x \theta]$: Matriz de rotação em torno do eixo Oy.
$[Rot^x \theta], [Rot^y \theta]$: Matriz de rotação em torno do eixo Ox.
[T-ISO]	: Matriz de transformação para obtenção de perspectiva isométrica.
[M]	: Matriz de transformação isométrica valendo-se da mudança da posição do observador.
[CV]	: Matriz de transformação para obtenção de perspectiva cavaleira.

x, y, z, w	: Coordenadas cartesianas.
i, j, k	: Versores do sistema cartesiano.
$ax+by+cz+e$: Equação de um plano.
$Prof(x, y)$: Função que retorna a profundidade z de um ponto cujas coordenadas são x, y, z .
$Intens(x, y)$: Função que retorna a cor, ou a intensidade da cor de um ponto cujas coordenadas são (x, y, z) .
$major(v)$: Operador que retorna o maior componente, em valores absolutos, de um vetor v .
J	: Janela a ser subdividida no algoritmo de Warnock.
J_i	: Uma das quatro sub-janelas em que a janela J foi subdividida.
$LC(J)$: Lista contendo todos os polígonos contidos no interior da janela J que, potencialmente, são visíveis.
$[I]$: Ponto, dado em coordenadas homogêneas, que se localiza no interior de um volume convexo.
$[V]$: Matriz volumétrica. É a matriz que descreve um volume convexo.
λ	: Parâmetro da equação da reta que passa por um segmento sob teste de visibilidade de Roberts.
α	: Parâmetro da equação de um raio visual.
P_i, P_j	: Extremos de um segmento de reta.
P_i^j, P_j^i	: Projecção de P_i^j .
PQ	: Segmento PQ .
$[P(X)]$: Conjunto dos pontos que pertencem à reta que contém a aresta P_i^j .
$[R(\alpha, \lambda)]$: Conjunto de todos os pontos que pertencem ao raio visual que parte de $P(X)$.
$[g] = [g^1 g^2 g^3 0]$: Vetor que aponta para o observador impróprio.
V^1, V^2, \dots, V^n	: Vértices de um sólido.
F	: Face F .
F_i	: Projecção de F .

[f] : Vetor coluna cujas componentes são os
coeficientes da equação do plano.
: Normal à uma aresta da perspectiva de um
polígono.

Δ	: Gradiente
$\ \cdot \ $: Norma euclidiana.
x, y, z	: Coordenadas cartesianas do espaço físico
ξ, η	: Coordenadas cartesianas do espaço do elemento
(ξ^i, η^i)	: Ponto gerado sobre o "parent element" no passo i .
T	: Função escalar da tensão.
T^0	: Valor de tensão mapeado.
$\langle T \rangle$: Vetor das tensões nos nós que se ligam a um determinado elemento.
σ	: Tensão genérica.
$\langle X \rangle$: Coordenadas nodais dos vértices do elemento.
DL	: Distância entre pontos gerados na curva de isotensão.
$\delta \xi, \delta \eta$: Incrementos.
∇	: Vetor de correção no trágado de curvas isovalor.
[Transf]	: Matriz de transformação geométrica.
$[H(\xi, \eta)]$: Matriz das funções de forma.
[CO]	: Matriz das coordenadas nodais do modelo indeformado.
[COD]	: Matriz das coordenadas nodais do modelo deformado.
[DEF]	: Matriz dos vetores de deformação.

001	1.1. Sistema integrado de análise por elementos finitos.
002	1.2. O impacto da computação gráfica na utilização de um programa de elementos finitos.
004	1.3. Requisitos para pré e pós processadores.
005	1.4. O programa NAVGRAF.
006	
008	2. Transformações geométricas lineares.
008	2.0. Introdução.
008	2.1. Características gerais das matrizes de transformação geométricas.
009	
011	2.2. Translação.
014	2.3. Rotação.
014	2.3.1. Rotação em torno do eixo Oz
015	2.3.2. Rotação em torno do eixo Oy
017	2.3.3. Rotação em torno do eixo Ox
017	2.3.4. Rotação em torno de um eixo qualquer
019	2.3.4.1. Tratamento tradicional
019	2.3.4.2. Tratamento usando analogia com números complexos
022	
026	2.4. Reflexão.
031	2.5. Homotetia.
031	2.6. Fechamento.
032	3. Perspectivas analíticas.
032	3.0. Introdução.
032	3.1. Tipos fundamentais de perspectivas.
032	3.1.1. Perspectivas de centro de projeção impróprio
033	3.1.2. Perspectivas de centro de projeção próprio
034	3.1.3. Classificação das perspectivas
035	
038	3.2. Obtenção das perspectivas usando operações lineares.
038	3.2.1. Obtenção de perspectivas axonométricas usando-se matrizes de rotação (mudança da posição do objeto).
039	3.2.2. Obtenção de perspectivas axonométricas através da mudança da posição do quadro de projeção.
042	3.2.3. Obtenção de perspectivas axonométricas através da mudança da posição do observador.
045	3.2.4. Obtenção de perspectivas cônicas.
051	3.2.4.1. Coordenadas homogêneas.
051	3.2.5. Obtenção de perspectivas cavaleiras.
056	
060	3.3. Fechamento.

061	4. Remoção de linhas invisíveis.....
061	4.0. Introdução.....
063	4.1. Remoção de linhas invisíveis usando o Z-buffer.....
064	4.2. Algoritmo de Warnock.....
068	4.3. Algoritmo de Roberts.....
068	4.3.1. Descrição de objetos pertencentes a um domínio
069	4.3.1.1. Os volumes
071	4.3.1.2. Os segmentos de reta
071	4.3.1.3. Os raios visuais
072	4.3.2. Determinação da arestas escondidas
074	4.3.3. Técnicas para acelerar os algoritmos de remoção de linhas invisíveis
074	4.3.3.1. Teste da normal a um plano
075	4.3.3.2. Teste da superposição na direção de \vec{g}
076	4.3.3.3. Teste minímax
077	4.3.4. Adaptação do algoritmo de Roberts ao pós processamento gráfico de malhas de elementos finitos
078	4.3.4.1. Ideia básica da adaptação
080	4.3.4.2. Determinação da intersecção de um raio visual que parte de P com o plano π que contém a face F
081	4.3.4.3. Condições necessárias para que uma região de PIP esteja obscurada por F quando consideramos suas arestas
087	4.3.4.4. O algoritmo
089	4.4. Algoritmo de Newell, Newell and Sancha.....
090	4.4.1. Porque a ordenação em função da profundidade falha?
090	4.4.2. Um teste mais sofisticado
091	4.4.3. Complicações que podem surgir
093	4.4.4. O algoritmo
095	4.5. Aplicações e fechamento do capítulo.....
098	5. Representação de Tensões e Deformações.....
098	5.0. Introdução.....
112	5.1. Traçado de curvas isotensas.....
100	5.1.1. Tratamento usando mapeamento contínuo
108	5.1.1.1. Problemas com a implementação do algoritmo
109	5.1.2. Tratamento usando mapeamento discreto
110	5.1.2.1. Qual o número ideal de triângulos ?
112	5.1.3. Aplicações
114	5.1.4. Conclusões sobre o mapeamento de tensões
117	5.2. Representação da estrutura deformada.....
119	5.3. Fechamento do capítulo.....
120	6. Considerações finais.....

8. Apêndices.....	125
7. Referências.....	122

1. INTRODUÇÃO

As técnicas de pré e pós processamento de dados para um programa de elementos finitos¹, também chamados de "técnicas auxiliares para o método dos elementos finitos" têm tido importância cada vez mais acentuada nos últimos anos, especialmente após a introdução da computação gráfica em sistemas integrados para análise de sistemas físicos. A fase de pré processamento inclui a geração de malha de elementos, verificação de consistências do modelo e renumeração dos nós e elementos do modelo a fim de diminuir os requisitos de memória. A fase de pós processamento se liga à apresentação de resultados de uma maneira útil e significativa. Veremos logo a seguir como as técnicas da computação gráfica podem contribuir nas fases "auxiliares" no processamento de um modelo por elementos finitos.

Antes porém de prosseguirmos, vale lembrar que tanto as técnicas auxiliares como o próprio método se aplicam em várias disciplinas da engenharia e da ciência, tais como na análise de campos eletromagnéticos, na análise de escoamentos de fluidos, na transmissão de calor, entre outras. Entretanto, para simplificar, este trabalho se concentrará na maioria das vezes em problemas e tecnologias da análise estrutural.

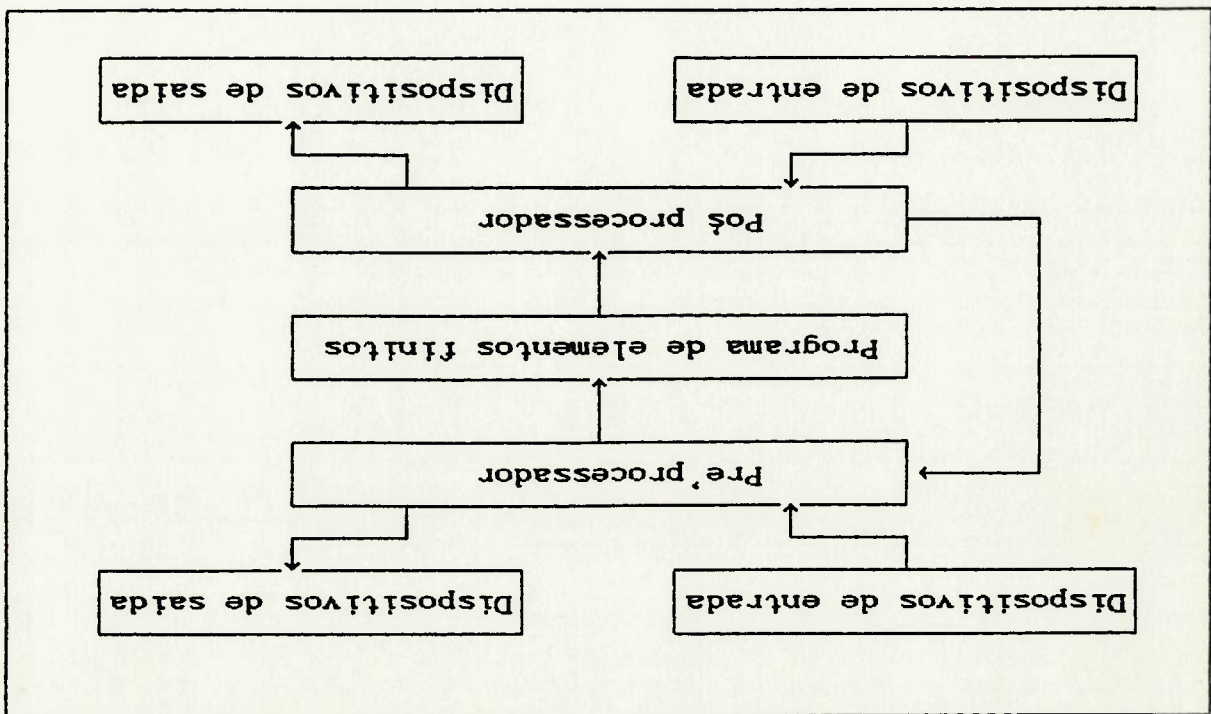
¹ Neste trabalho, estamos supondo que o leitor tem familiaridade com o método dos elementos finitos. Em todo caso, podemos dizer que o método dos elementos finitos é uma ferramenta computacional-numérica que visa obter soluções aproximadas para modelos discretos como representação de sistemas físicos.

A idealização de uma dada estrutura é o primeiro passo no processo de análise. A estrutura real é substituída por uma coleção de pequenos elementos não infinitesimais, mais sim "finitos". Portanto, ela é substituída por uma idealização, ou um modelo computacional. Após a idealização ter sido efetuada, dados relativos a ela devem ser preparados e inseridos no computador. Essa fase está sujeita a muitos erros além de ser extremamente demorada, chegando a ser o gargalo de toda a análise de uma estrutura. Os programas de elementos finitos modernos possuem sistemas de detecção de erros, mas eles atualmente são capazes apenas de diagnosticar erros de sintaxe da estrutura, tais como por exemplo, comprimentos nulos de vigas. Erros mais graves, como erros na especificação das coordenadas de um ponto, erros na especificação na incidência de elementos ou superposição de malhas não são detectados. Esses erros só podem ser diagnosticados com o auxílio do usuário, um ser humano que terá uma participação mais efetiva no processo de análise quanto melhor forem os recursos de interação com o sistema computacional. Exatamente aí entra a computação gráfica, fornecendo elementos para uma comunicação efetiva entre o homem e a máquina. O esquema de um sistema integrado para a análise por elementos finitos é mostrado abaixo:

-Existem diversos programas de elementos finitos, cada um com seus pontos fortes e fracos, provavelmente, nenhum podendo ser considerado "o melhor".

fatos:
No projeto de pré e pós processadores de um sistema integrado, devem ser levados em consideração os seguintes

sistema computacional.
que tenha por finalidade a comunicação entre o usuário e o
entim, qualquer dispositivo existente, ou que venha a existir,
monitores de vídeo, teclados, impressoras, plotters, mouses,
gráficas. Os dispositivos de entrada e de saída se referem a
usuário de maneira útil e relevante se valendo de técnicas
enviados ao pós processador que se encarrega de fornecê-los ao
enviados ao programa de elementos finitos. Os resultados são
Após os dados serem gerados pelo pré processador, eles são



A inclusão de técnicas de computação gráfica foi provavel-
 mente a maior contribuição externa ao método dos elementos
 finitos nos últimos anos. Com elas, o método vem se tornando
 prático e rápido o suficiente para ser incluído no ciclo de
 projeto desde seus primeiros passos, ao contrário do tratamento
 tradicional que aplicava o método apenas como uma verificação a
 posteriori de cálculos anteriores. [KNOW] indica que em
 escritórios de análise estrutural onde o projeto por elementos
 finitos é uma constante, a produtividade aumentou em cerca de
 80%, embora o próprio artigo reconheça que em ambientes onde o
 método é aplicado apenas esporadicamente, a produtividade pouco
 aumentou ou até mesmo diminuiu. As causas apontadas são o tempo
 requerido de treinamento, a quantidade de informações que deve
 ser guardada (e não esquecida) por um dado usuário, e o que
 todos nós experimentamos quando estamos à frente de um sistema
 gráfico interativo - ficamos experimentando várias funções do

1.2. O impacto da computação gráfica na utilização de um
 programa de elementos finitos.

Percebemos então que, do ponto de vista do projeto de
 sistemas integrados para análise por elementos finitos,
 pré e pós processadores devem ser o mais flexíveis
 possível, para que a longo prazo a manutenção de tais
 sistemas seja facilitada.

-O método dos elementos finitos é geral. Ele pode ser
 aplicado em uma variedade de disciplinas. O que mais muda
 de uma aplicação para outra não é o código do programa do
 método, mas sim dos programas de pré e pós processamento.

-Programas de elementos finitos podem ser encontrados em
 diversas configurações que rodam em microcomputadores e
 até em computadores de "grande porte".

- Gerar coordenadas nodais,
- Gerar elementos,
- Gerar condições de contorno,
- Gerar carregamentos nodais e distribuídos,
- Gerar dados relativos a propriedades dos materiais e geométricas,
- Verificar dados quanto à sintaxe,
- Mostrar graficamente a malha de elementos,
- Renumerar nós ou elementos a fim de reduzir a quantidade de memória computacional requerida pelo modelo.

Pré processamento compreende todas as atividades associadas com preparação, geração e verificação de dados antes que eles sejam enviados a um programa de elementos finitos. Especificamente, os pré processadores devem ser capazes de:

1.3. Requisitos para pré e pós processadores.

Com a inclusão de recursos gráficos, sistemas de análise por elementos finitos passaram a ser mais atraentes e portáteis, mais pessoas passaram a interessar-se por eles. O grande entusiasmo em torno desses sistemas parece indicar que os usuários se esquecem que o método se fundamenta em modelos aproximados, e que por isso, não são infalíveis do ponto de vista físico, apesar de que do ponto de vista matemático o programa estar correto. Em outras palavras, os usuários parecem ignorar, ou não se importar, com a diferença entre verificação e validação de um sistema. O primeiro conceito se refere à correção matemática e sintática do sistema, não se importando com a sua adequação ao mundo físico, enquanto que o segundo conceito se liga à capacidade do sistema fornecer resultados coerentes com nossa realidade física. A validação de um sistema passa necessariamente por um engenheiro experiente, já que somente ele é capaz de avaliar a aderência dos resultados de um modelo resolvido pelo sistema e a realidade.

sistema, mesmo que sejam irrelevantes para o trabalho em questão.

-E capaz de mostrar a malha tanto em projeção axonométrica como em projeção cônica a partir de qualquer posição no espaço.

-E capaz de mostrar condições de carregamento e especificações de condições de contorno.

-E capaz de mostrar numeração de nós e de elementos (hidden lines).

O programa NAVGRAF foi desenvolvido com as técnicas que iremos discutir neste trabalho. Basicamente, trata-se de um processador útil nas fases de pré e pós processamento de um modelo de elementos finitos. Entre as suas características podemos destacar:

1.4. O programa NAVGRAF.

Este trabalho se restringirá em discutir apenas os itens das listas acima que se ligam diretamente à computação gráfica, dando ênfase à obtenção de perspectivas por computador e remoção de linhas invisíveis, atividades essenciais em qualquer pré e pós processador.

-Mostrar graficamente os resultados (Campo de deslocamentos, forças de reação, etc),

-Gerar e mostrar curvas de isovalor,

-Trabalhar as tensões fornecidas pelo programa a fim de indicar tensões equivalentes, tensões máximas e mínimas, etc.

-Preparar gráficos e tabelas para a inclusão em relatórios.

Pós processamento compreende todas atividades relacionadas com o tratamento de resultados obtidos pelo programa de elementos finitos. Especificamente, os pós processadores devem ser capazes de:

Esse programa será utilizado muitas vezes durante o trabalho para ilustrar alguns resultados teóricos. O trabalho enfocará as técnicas de computação gráfica que permitem a visualização da malha de elementos finitos, a remoção de linhas ocultas, e a representação de resultados (tensões e deformações).

- E capaz de traçar curvas de isovalor.
- E capaz de mostrar a estrutura deformada.
- E altamente interativo com o usuário.

**2. TRANSFORMAÇÕES
GEOMÉTRICAS LINEARES**

2.0. Introdução

Podemos efetuar um grande número de transformações sobre entes geométricos (pontos, retas, planos, superfícies, etc), tais como rotações em torno de um eixo qualquer e translações, usando-se apenas operações matriciais simples (transformações lineares). Por exemplo, suponhamos o cubo de lado unitário mostrado na figura

2.1. Uma matriz $[T] = \begin{bmatrix} 0.707 & -0.408 & 0 \\ 0 & 0.816 & 0 \\ -0.707 & -0.408 & 0 \end{bmatrix}$, resultado de uma con-

catenação de operações simples, quando aplicada ao cubo, o transforma em sua representação isométrica (portanto plana). Neste capítulo estudaremos tais transformações, como elas podem ser construídas, e algumas de suas aplicações na computação gráfica.

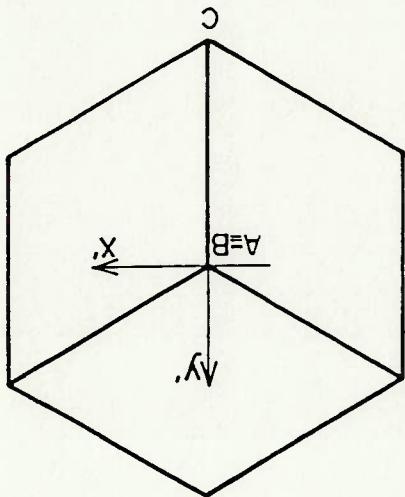
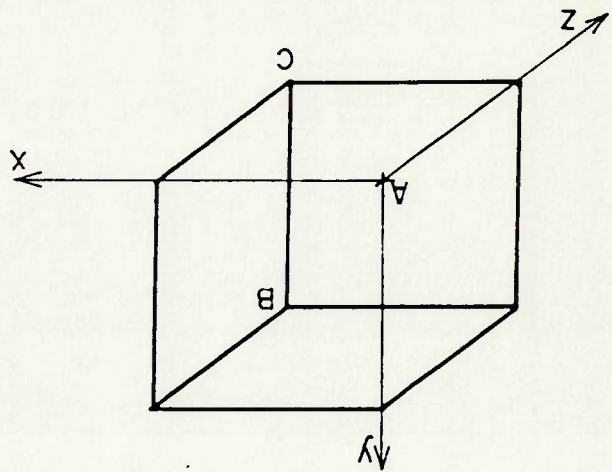


FIGURA 2.1

Obtenção de uma perspectiva isométrica usando-se uma matriz de transformação $[T]$.

Essas combinações são, na verdade, frutos de uma transformação linear F que foi colocada na forma matricial. No exemplo acima, F pode ser facilmente dada por:

$$[2 \ 3 \ 4][T] = [2x_1 + 3x_0 + 4x(-3) \ 2x_3 + 3x_2 + 4x_1 \ 2x_4 + 3x(-2) + 4x_5] = [-10 \ 16 \ 22] .$$

O produto $[2 \ 3 \ 4][T]$ resulta em um outro ponto cujas coordenadas são resultantes de uma combinação entre 2, 3 e 4:

$$[T] = \begin{bmatrix} 1 & 3 & 4 \\ 0 & 2 & -2 \\ -3 & 1 & 5 \end{bmatrix}$$

As matrizes podem ser encaradas como uma maneira fácil de se combinar linearmente as coordenadas de um dado ponto. Por exemplo, seja o ponto $[2 \ 3 \ 4]$ e a matriz $[T]$:

2.1. Características gerais das matrizes de transformação geométricas

Este capítulo usa vários conceitos da álgebra linear. Por esse motivo, foi incluído um apêndice ao final do trabalho sobre aquele assunto para os leitores que julguem necessária uma revisão.

Antes de iniciarmos nosso estudo, devemos notar que qualquer transformação geométrica pode ter pelo menos duas interpretações, dada a relatividade dos movimentos segundo diferentes sistemas de referência arbitrariamente escolhidos. Assim, a situação em que um ponto gira em torno de um eixo fixo de um sistema de coordenadas, também pode ser interpretada como a rotação do sistema de eixos sobre a origem, mantendo-se o ponto imóvel (fig. 2.6). Neste capítulo, adotaremos o primeiro enfoque, mas o leitor deve ficar atento às demais interpretações, visto que elas serão usadas no capítulo seguinte - PROJEÇÕES.

$$F(X, Y, Z) = (1X - 3XZ, 3XX + 2XY + 1XZ, 4XX - 2XY + 5XX).$$

Acontece que algumas transformações lineares se revelam úteis do ponto de vista geométrico, como veremos logo abaixo.

Como cada ponto do espaço tridimensional possui três coordenadas, seria de se supor que uma matriz de transformação fosse quadrada e de ordem 3. A primeira conclusão é correta em geral, mas a segunda é errada. Perceberemos que nem todas as transformações geométricas são lineares, e que para contornar essa dificuldade, deveremos usar um artifício para "linearizá-las", adotando uma quarta coordenada. Embora em nem todas as transformações ela seja requerida, ela será mantida por uma questão de compatibilidade, como mostraremos a seguir.

Dois transformações podem ser combinadas em uma única, através de uma operação conhecida na literatura por *concatenação*, que nada mais é do que produto entre duas matrizes. Sejam duas operações $[S]$ e $[T]$ a serem aplicadas, nessa ordem, sobre o ponto $[P] = [X \ Y \ Z]$. O resultado seria $[P][S][T]$. É fácil perceber que uma única matriz $[R] = [S][T]$ faz o mesmo trabalho que $[S]$ e $[T]$ juntas. Assim, o fato de todas as matrizes possuírem a mesma ordem facilitaria a construção de operação mais complexas.

Em geral não se pode afirmar propriedades comuns para todas as matrizes de transformação geométricas, a não ser que algumas de ordem 4. Porém, para conjuntos específicos, existem propriedades interessantes a respeito da ortogonalidade, sobre os autovalores, sobre os determinantes e assim por diante. Por exemplo, um auto vetor de uma matriz de rotação deve corresponder a um único auto valor real e esse auto vetor deve ser exatamente o eixo de rotação.

Passamos agora ao estudo das matrizes de transformações geométricas.

Suponhamos que desejamos transladar o ponto P_0 para uma nova posição P (fig. 2.2). Vetorialmente teríamos:

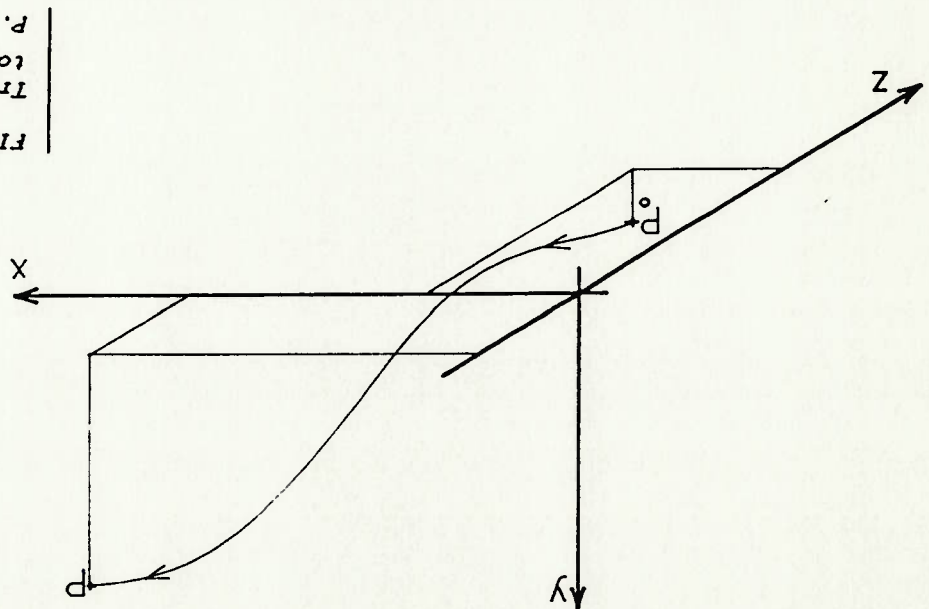


FIGURA 2.2

Translação de um ponto P_0 para a posição P .

$$P_0 = (x_0, y_0, z_0)$$

$$P = (x, y, z)$$

$$P - P_0 = (x - x_0, y - y_0, z - z_0) = (\delta x, \delta y, \delta z)$$

$$P = P_0 + (P - P_0)$$

Expressando a relação acima como uma lei de formação F , que relaciona todo ponto $Q=(x,y,z)$ a um outro ponto P do espaço, temos:

$$F(Q) = P, F: \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

$$F(x, y, z) = (x + \delta x, y + \delta y, z + \delta z), F: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \quad (2.1)$$

Embora a operação de translação seja simples, ela é provavel-mente, dentre todas as operações que estudaremos, a mais rica do ponto de vista conceitual. Devemos atentar ao fato que a trans- formação (2.1) não é linear!

As condições para que uma transformação $F: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ seja linear são:

a) $F(kv) = k \cdot F(v)$, com $v \in \mathbb{R}^3$ e $k \in \mathbb{R}$. Em particular, se

b) $F(u+v) = F(u) + F(v)$, com $u, v \in \mathbb{R}^3$.

$k=0, F(0)=0.$

nada w.

a) Fazer uma projeção ortogonal do sistema x, y, z, w sobre o espaço x, y, z . Para passar-
mos do sistema quadridimensional para o
tridimensional, bastaria ignorar a coorde-

maneira de se fazer essa correspondência :
Se pensarmos em termos geométricos, temos em princípio, duas

sentação.

em como fazer a correspondência entre os dois sistemas de repre-
mentação num sistema quadridimensional, a questão agora se torna
Percebendo que pode ser vantajoso representar um ponto tridi-

b) $F(k \cdot (x, y, z, w)) = k \cdot F(x, y, z, w)$

a) $F((x_0, y_0, z_0, w_0) + (x_1, y_1, z_1, w_1)) = F(x_0, y_0, z_0, w_0) + F(x_1, y_1, z_1, w_1)$

é linear pois:

$F(x, y, z, w) = (x+w, \delta x, y+w, \delta y, z+w, \delta z, w)$ (2.2)

A translação em quatro dimensões (Para fixar as idéias, $w=1$)

linear.

espaço tridimensional torna possível a operação de translação
(x, y, z, w). Veremos que o acréscimo de uma dimensão a mais em nosso
ar w , de modo que o ponto (x, y, z) passa a ser representado por
Para contornar essa dificuldade, criamos uma variável auxili-

das matrizes).

Esse fato implica, entre outras coisas, em que a
transformação acima, como foi definida, não pode ser colocada em
forma matricial (e portanto não pode gozar das boas propriedades

3 Ver apêndice sobre álgebra linear. A base usual, ou canônica é: $\langle (1,0,0,0) \rangle, \langle (0,1,0,0) \rangle, \langle (0,0,1,0) \rangle, \langle (0,0,0,1) \rangle$.

Seja um ponto $P(2,3,1)$. Sua representação em coordenadas homogêneas é dada por $[P] = (2 \ 3 \ 1 \ 1)$. Vamos aplicar uma matriz de transformação linear (Trans) correspondente à uma translação de $v = (-3, 2, 1)$. Uma explanação mais detalhada será efetuada no item 3.2.4.1, do capítulo referente à obtenção de projeções ou perspectivas analíticas.

Exemplo:

$$[x \ y \ z \ w] = [x_0 \ y_0 \ z_0 \ w_0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Matricialmente, a operação de translação linear (quadridimensional) fica na base usual:

Veremos mais adiante que o uso da projeção cônica oferece vantagens, e as coordenadas tetradimensionais recebem um nome especial, *coordenadas homogêneas*.

$$[xw \ yw \ zw] \equiv (xw/w, yw/w, zw/w) \quad (2.3)$$

b) Fazer uma projeção cônica, tendo como centro de projeção o ponto $(0,0,0,0)$ e como quadro o hiperplano $w = 1$. Um ponto (x,y,z,w) seria projetado em $(x/w, y/w, z/w, 1)$. Colocado de uma outra forma, o ponto ordinário (x,y,z) seria representado nesse sistema de coordenadas por $[xw \ yw \ zw]$, e a relação entre os modos de representação seria dada por:

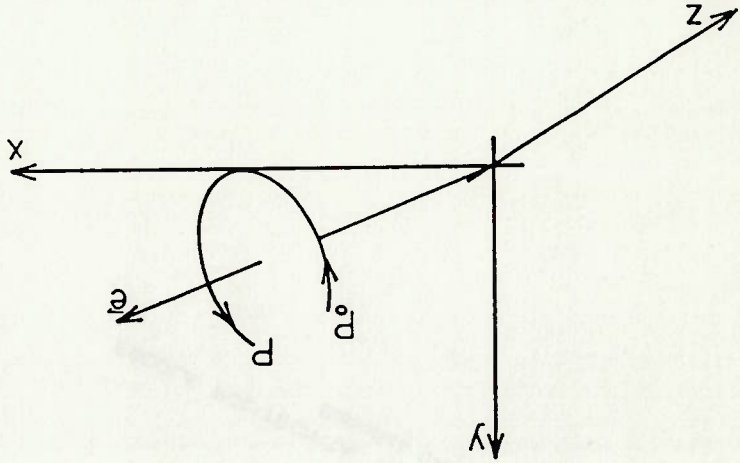


FIGURA 2.3
 Rotação de um ponto P em torno de um eixo e do espaço 3D.

Um ponto sofre rotação em torno de um eixo de espaço tridimensional quando ele descreve uma circunferência num plano perpendicular ao eixo de rotação (fig. 2.3). Neste trabalho, veremos as operações de rotação em torno dos eixos coordenados Ox, Oy, Oz e em torno de um eixo qualquer. Concionamos como sendo o sentido positivo de rotação aquele dado pela regra da mão direita. O dedo indicador aponta o sentido positivo, se o polegar apontar no sentido da normal ao plano onde ocorre a rotação (fig. 2.4).

2.3. Rotação

Pela relação (2.4) temos [Trans] dada por:

$$[Trans] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & 2 & 1 & 1 \end{bmatrix}$$

O ponto [P] será trasladado para

$$[2 \ 3 \ 1 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & 2 & 1 & 1 \end{bmatrix} = [-1 \ 5 \ 2 \ 1]$$

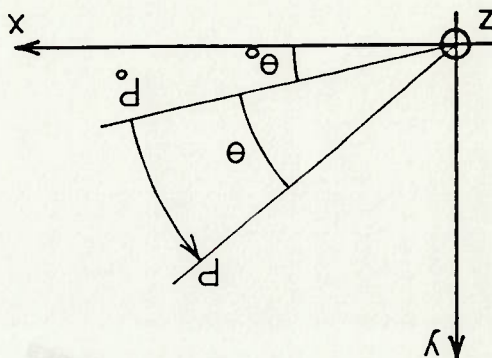


FIGURA 2.5
 Rotação de um ponto
 de θ graus em torno
 do eixo Oz.

$$\begin{aligned}
 P_0 &= [r \cdot \cos(\theta_0) \quad r \cdot \sin(\theta_0) \quad z] = [x_0 \quad y_0 \quad z_0] \\
 P &= [r \cdot \cos(\theta_0 + \theta) \quad r \cdot \sin(\theta_0 + \theta) \quad z] = [x \quad y \quad z]
 \end{aligned}$$

Como $r \cdot \cos(\theta_0 + \theta) = r \cdot \cos(\theta_0) \cdot \cos(\theta) - r \cdot \sin(\theta_0) \cdot \sin(\theta)$
 $r \cdot \sin(\theta_0 + \theta) = r \cdot \sin(\theta_0) \cdot \cos(\theta) + r \cdot \cos(\theta_0) \cdot \sin(\theta)$

Suponhamos que o ponto P seja a posição final do ponto P_0 depois que este sofre uma operação de rotação de θ graus em torno do eixo Oz (fig. 2.5). Temos diretamente da figura:

2.3.1. Rotação em torno do eixo Oz

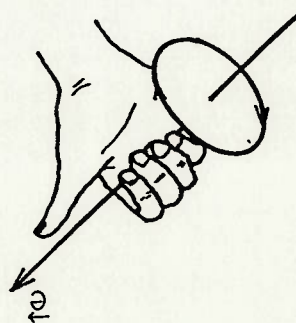


FIGURA 2.4
 Regra da mão direita.
 O indicador mostra o
 sentido positivo de
 rotação.

- a) O ponto P_0 gira em torno de Oz.
 b) O sistema de referência gira em torno de sua origem, mas o ponto P_0 continua imóvel (fig. 2.6)

operação acima:

Vale observar que duas interpretações são válidas para a

$$e \quad [x \ y \ z \ w] = [x_0 \ y_0 \ z_0 \ w_0] \cdot [Rot^z \ \theta]$$

$$[Rot^z \ \theta] = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

suas propriedades:

Apenas para manter a compatibilidade com as matrizes 4x4 da translação (e portanto permitindo o produto matricial para compor transformações), podemos expandir a matriz acima sem prejuízo de

$$[Rot^z \ \theta] = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$P = P_0 \cdot [Rot^z \ \theta] \text{ onde}$$

teremos:

já é linear, isto é, não é necessário o emprego da quarta dimensão. Assim, matricialmente (considerando-se a base usual)

$$F: \mathbb{R}^2 + \mathbb{R}^2 \text{ com } F(x, y, z) = (x \cdot \cos\theta - y \cdot \sin\theta, x \cdot \sin\theta + y \cdot \cos\theta, z)$$

nida como

Podemos verificar facilmente que a transformação acima defini-

$$\begin{aligned} x &= x_0 \cdot \cos(\theta) - y_0 \cdot \sin(\theta) \\ y &= x_0 \cdot \sin(\theta) + y_0 \cdot \cos(\theta) \end{aligned} \quad (2.6)$$

Temos facilmente:

$$[Rot^x \theta] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$P = P_o \cdot [Rot^x \theta]$ onde

De maneira também totalmente análoga, temos para a rotação em torno do eixo Ox, considerando a base usual:

2.3.3. Rotação em torno do eixo Ox

$$[Rot^y \theta] = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & -\sin(\theta) & 1 \end{bmatrix} \quad (2.9)$$

$P = P_o \cdot [Rot^y \theta]$ onde

De maneira totalmente análoga, chegamos à seguinte operação matricial (considerando-se a base usual) para a rotação de um ponto em torno do eixo Oy :

2.3.2. Rotação em torno do eixo Oy

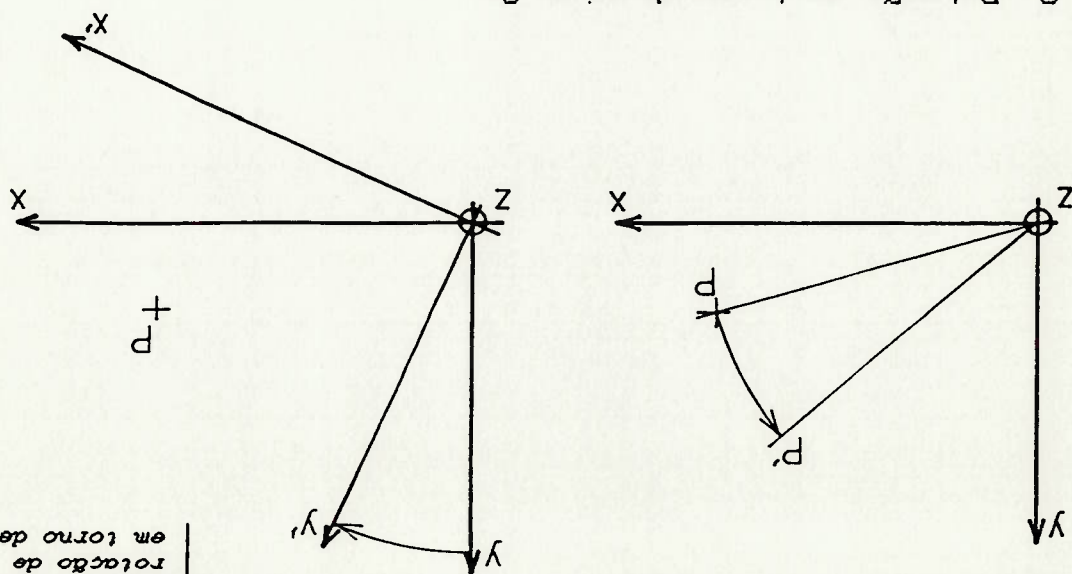


FIGURA 2.6
Duas interpretações para uma mesma rotação de um ponto em torno de um eixo.

Exemplo:

Vamos fazer uma rotação de +30 graus do ponto P(2,3,1) em torno de um eixo m paralelo à Oy e que passa pelo ponto Q(3,-2,-1).

Os pontos P e Q, representados em coordenadas homogêneas, são dados por P=[2 3 1 1] e Q=[3 -2 -1 1]. Para que a operação pedida seja efetuada, devemos em primeiro lugar, trasladar o eixo de rotação para sobre Oy (ou numa outra interpretação, trasladar o ponto "para perto de Oy").

A matriz de translação é dada por (2.4)

$$[T]= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 2 & 1 & 1 \end{bmatrix} \quad [2 \ 3 \ 1 \ 1] = [-1 \ 5 \ 2 \ 1]$$

Depois que a translação foi efetuada, a rotação em torno do eixo Oy pode ser feita. A matriz de rotação (Roly 30) é segundo (2.8):

$$[Roly \ 30] = \begin{bmatrix} \cos(30) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \sin(30) & 0 & \cos(30) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [-1 \ 5 \ 2 \ 1] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \cos(30) & 0 & -\sin(30) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(30) & 0 & \cos(30) & 0 \end{bmatrix} = [0,18 \ 5 \ 7,28 \ 1]$$

Depois que o ponto foi rotacionado, devemos retornar o eixo de rotação à sua posição original.

$$[0,18 \ 5 \ 7,28 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [3,18 \ 3 \ 6,28 \ 1]$$

Antes de finalizarmos, note que poderíamos ter concatenado as operações em uma única matriz (T) que realizaria todo de uma só vez:

$$[T]=[T][Roly \ 30][T]^{-1} \text{, de modo que } [2 \ 3 \ 1 \ 1XT] = [3,18 \ 3 \ 6,28 \ 1].$$

⁴ Oy por exemplo. Poderia ser um eixo coordenado qualquer.

1) Rotação do objeto O e do eixo \vec{e} em torno do eixo $Oy^{(4)}$, de maneira que \vec{e} se posicione sobre o plano xy .

operações seriam:

deveremos retornar \vec{e} para a sua posição inicial. Especificamente as da de α graus. Logicamente, depois que a rotação for efetuada, nova posição para o eixo \vec{e} seria fácil realizarmos a rotação pedida em torno de Oz de maneira que o eixo \vec{e} fique sobre o eixo Ox . Nesta rotação sobre o plano xz , e a segunda, uma nova rotação, agora em rotação em torno do eixo Oy de tal maneira que o eixo \vec{e} seja posição, digamos, Ox . Efetuá-la em duas etapas, a primeira, uma qualquer seria reposicionarmos \vec{e} de tal sorte que ele coincida com a estratégia para executarmos uma rotação em torno de um eixo por qualquer uma das matrizes vistas no item anterior. Assim, uma boa estratégia para executarmos uma rotação em torno de um eixo dos, a operação pedida seria extremamente simples, sendo executada. Devemos notar inicialmente que se \vec{e} fosse um dos eixos coordenados (e_1, e_2, e_3) , é possível norma unitária, isto é, $(e_1^2 + e_2^2 + e_3^2) = 1$. eixo que passa pela origem, e tem direção dada pelo vetor \vec{e} = matriz de transformação correspondente [Rot \vec{e}] em torno de um Suponhamos efetuar a rotação de α graus de um objeto O (com

2.3.4.1. Tratamento tradicional

Estudaremos agora a rotação de um ponto em relação a um eixo arbitrário no espaço tridimensional. Veremos dois tratamentos equivalentes. O primeiro é mais intuitivo, valendo-se das matrizes que acabamos de ver na seção precedente. O segundo foi incluído nesse trabalho, por julgarmos-lo extremamente criativo ao se valer de uma analogia com os números complexos.

2.3.4. Rotação em torno de um eixo qualquer.

$$[R_y] = \begin{bmatrix} A & 0 & B & 0 \\ 0 & 1 & 0 & 0 \\ -A & 0 & B & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

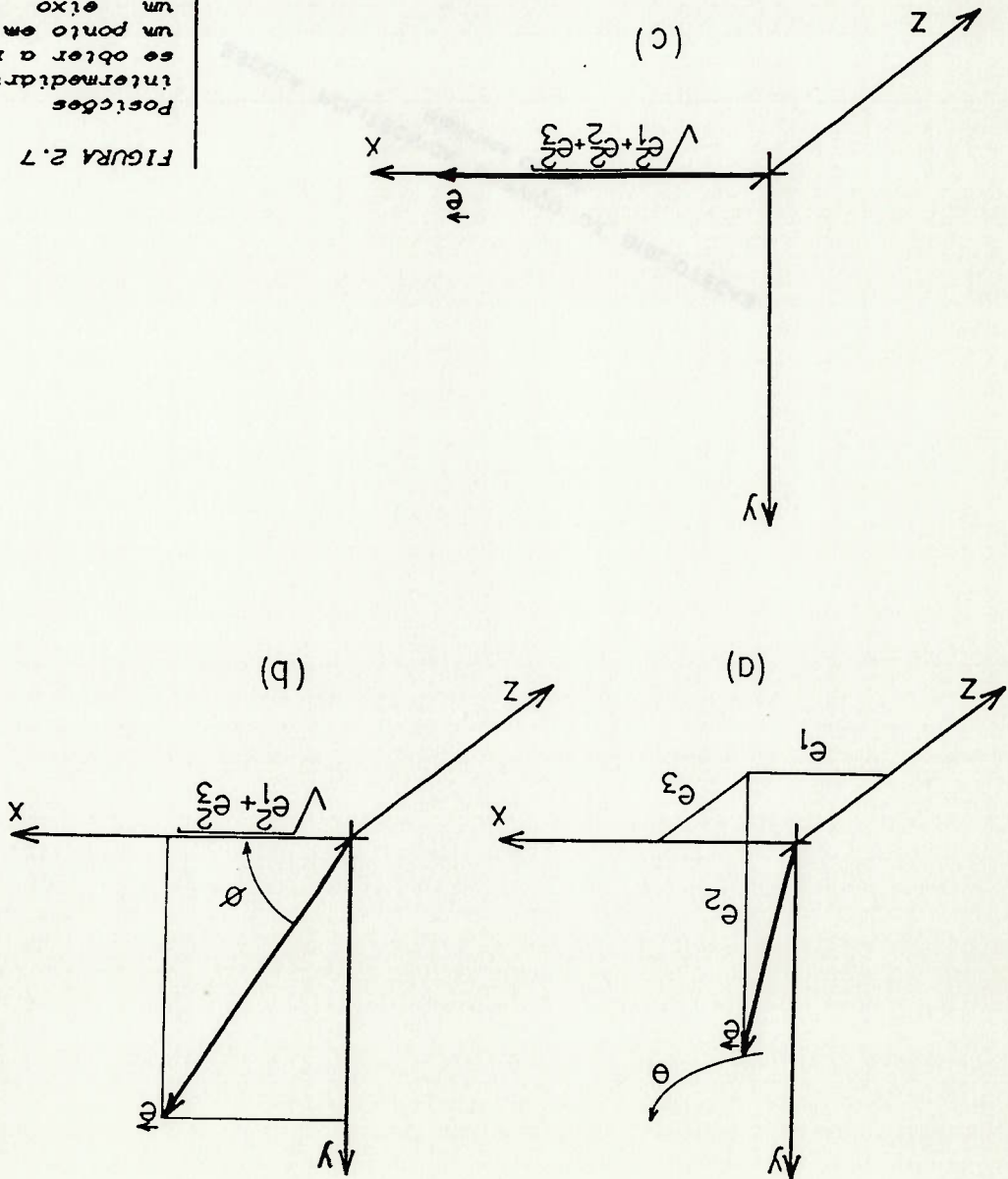
(2.11)

A matriz correspondente a essa operação é

$$\cos\theta = e^1 / (e^1 + e^2) \quad \text{e} \quad \sin\theta = e^2 / (e^1 + e^2)$$

Deveremos efetuar uma rotação de θ graus, com (Fig. 2.7):

FIGURA 2.7
 Posições intermediárias para se obter a rotação de um ponto em relação a um eixo qualquer: abordagem tradicional



Remova fazer a rotação de 30 graus do ponto $P=(3,5,2)$, cuja representação em coordenadas homogêneas é $[3 \ 5 \ 2 \ 1]$, em torno do eixo $e = (1,1,1)$. A representação homogênea para ele é $[1 \ 1 \ 1 \ 0]$ (note o zero na quarta coordenada. Podemos interpretar um vetor como sendo um ponto ideal).

Exemplo:

$$[Rot \ e] = [R_y][R_z][R_x][R_y]^{-1} \quad (2.13)$$

Assim, para efetuarmos rotação de α graus de um objeto O em torno de um eixo que passa pela origem e tem direção dada pelo vetor $\vec{e} = (e_1, e_2, e_3)$ (e achar a matriz correspondente $[Rot \ \vec{e}]$), basta concatenarmos as operações acima e aplicarmos as respectivas operações inversas de modo que o eixo \vec{e} seja recolocado em sua posição original:

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de rotação é:

sobre Ox):

iii) Rotação em torno do eixo Ox (lembre-se que agora \vec{e} está

$$[R_z] = \begin{bmatrix} C & -D & 0 & 0 \\ D & C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

A matriz correspondente a essa operação é

$$\begin{aligned} \cos\phi &= (e_2^2 + e_3^2)^{1/2} / (e_2^2 + e_3^2)^{1/2} = (e_2^2 + e_3^2)^{1/2} = C \\ \sin\phi &= -e_2 / (e_2^2 + e_3^2)^{1/2} = -e_2 = D \end{aligned}$$

Deveremos efetuar uma rotação de ϕ graus, com :

cione sobre o eixo x :

ii) Rotação do objeto O e do eixo \vec{e} , de maneira que \vec{e} se posi-

5 Note que $[Rot \alpha]^{-1} = [Rot -\alpha]$.

Perceba o leitor que, como já era de se esperar, $[1 \ 1 \ 0]$ é um autovetor de $[Rot \alpha]$, sendo $\lambda=1$ seu auto valor correspondente.

$$[Rot \alpha] = \begin{bmatrix} 0.911 & 0.333 & -0.244 & 0 \\ -0.244 & 0.911 & 0.333 & 0 \\ 0.333 & -0.244 & 0.9108 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[Rot \alpha] = [Rot y \ 45][Rot z \ -35.26][Rot x \ 30][Rot y \ -45] \quad (5)$$

onde $[P'] = [P][Rot \alpha]$,

$[P']$ é a posição final de $[P]$. Em resumo,

$$[P'] = [Rot z \ +35.26][Rot y \ (-45)][P]$$

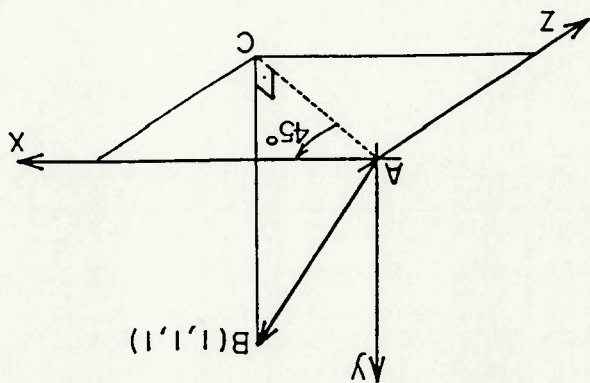
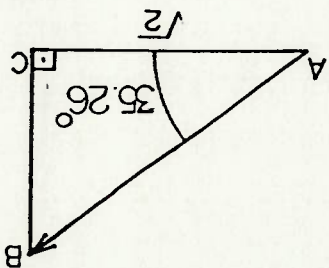
Para finalizar o problema, devemos fazer o eixo de rotação voltar para a sua posição original:

$$[P] = [Rot x \ 30][P']$$

Nesta posição o problema se resume em fazer uma rotação do ponto $[P]$ em torno do eixo Ox de $+30$ graus:

Ângulos necessários para que o eixo e se posicione sobre o eixo Oz .

FIGURA 2.8



$$[P] = [Rot y \ 45][Rot z \ (-35.26)][P']$$

Em primeiro lugar, devemos levar o eixo e o ponto para uma posição tal que o eixo fique alinhado com Ox . Pela figura 2.8 percebemos que isso é conseguido através de duas rotações consecutivas. A primeira em torno do eixo Oy de $+45$ graus e a segunda de -35.26 graus em torno do eixo Oz . Assim essa primeira parte da operação final seria dada por:

2.3.4.2. Tratamento usando analogia com números complexos.

Veremos agora uma maneira extremamente engenhosa para se obter a matriz de transformação de rotação em relação a um eixo qualquer, se valendo de uma analogia com números complexos.

Antes de iniciarmos nosso estudo é conveniente lembrarmos que um número complexo c tem a forma $a+bi$, onde a e b são reais e j , denominado imaginário, é tal que $j^2 = -1$. Lembramos também que o número c pode ser escrito de outras maneiras, entre elas destaca-se a forma exponencial

$$c = a + bi = r \cdot (\cos \theta + j \cdot \text{sen} \theta) \left\{ \begin{array}{l} a = r \cdot \cos \theta \\ b = r \cdot \text{sen} \theta \end{array} \right.$$

onde r é denominado módulo de c , e θ seu argumento. Essa forma permite uma interpretação geométrica para o número c como mostra a figura 2.9. Note que o espaço que contém c é simplesmente o plano xy nessa interpretação. Devemos ainda notar que o resultado da multiplicação entre dois números $c = re^{j\theta}$ e $d = se^{j\phi}$ é o número $g = (r \cdot s) \cdot e^{j(\theta + \phi)}$. Note com atenção que o argumento do produto é igual à soma dos argumentos dos fatores.

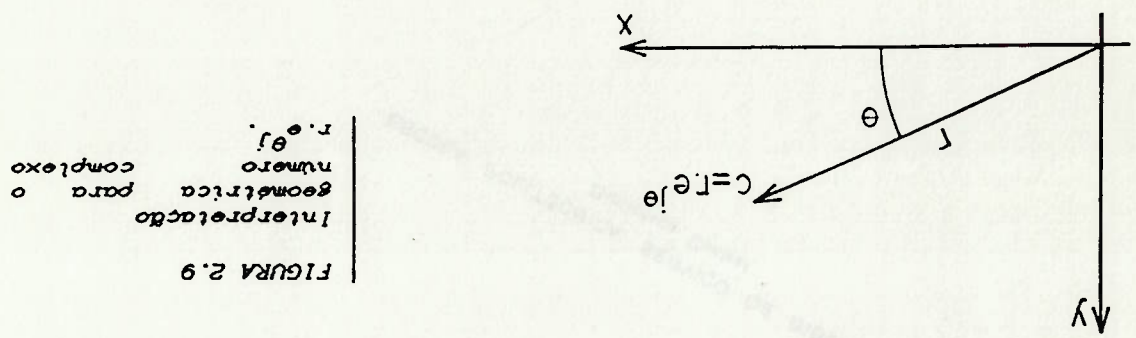


FIGURA 2.9
Interpretação geométrica para o número complexo $r \cdot e^{j\theta}$.

A analogia pode ser iniciada. Perceba que tanto a rotação de um ponto (vetor) no espaço quanto a rotação de um complexo podem ser obtidas através alguma multiplicação entre entidades. A ideia central na analogia consiste em perceber que existe uma matriz $[L]$ que se comporta da mesma maneira que o número imaginário $j = e^{j(\pi/2)}$.

Passamos a mostrar qual é essa matriz [L] de comportamento especial. Quando um número complexo c é multiplicado por $e^{j(\pi/2)}$, temos que c sofre uma rotação de 90° no sentido anti-horário em torno de um eixo passando pela origem e perpendicular ao plano xy . Uma boa sugestão para um operador análogo à (c, j) (vezes j) seria então o produto vetorial $\vee (0, 0, 1), (-1)$, que também ao ser aplicado a um vetor do plano xy , o rotaciona de 90° no sentido anti-horário. Ora, como não estamos mais presos ao plano xy (temos agora todo o espaço tridimensional), estamos livres para escolher um outro vetor (a, b, c) , ainda de norma unitária, diferente de $(0, 0, -1)$, de tal maneira que o operador $\vee (a, b, c), (-1)$ continue sendo o análogo de (c, j) (Note que $\vee a, b, c), (-1)$ é ortogonal à \vee).

O operador $\vee (a, b, c), (-1)$ leva um vetor \vee à:

$$(x, y, z) \vee (a, b, c), (-1) = (zb-yc) \hat{e}_x + (xc-za) \hat{e}_y + (ya-xb) \hat{e}_z$$

Expressando a ação do operador $\vee (a, b, c), (-1)$ em termos de uma

função linear:

$$L: \mathbb{R}^3 \rightarrow \mathbb{R}^3, L(x, y, z) = (zb-yc, xc-za, ya-xb)$$

No 4D:

$$L(x, y, z, w) = (zb-yc, xc-za, ya-xb, 0)$$

Perceba que $(zb-yc, xc-za, ya-xb, 0)$ é ortogonal a qualquer vetor (x, y, z, w) .

Matricialmente, a matriz de transformação relativa à base canônica associada à L fica:

$$[L] = \begin{bmatrix} 0 & c & -b & 0 \\ -c & 0 & a & 0 \\ b & -a & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.14)$$

Caso o leitor se interesse, é fácil verificar que curiosamente:

$$[L] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot [L]^2 = \frac{1}{2} \begin{bmatrix} -2 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Vamos fazer a mesma rotação do exemplo anterior, apenas mudando o tratamento para solução do problema. O eixo de rotação é dado pela direção (1 1 0). Note que a norma de (1 1 0) não é um. Por isso, devemos normalizar o vetor dividindo-o por $\sqrt{1+1+0} = \sqrt{2}$. Assim, o eixo de rotação normalizado seria $\frac{1}{\sqrt{2}} \cdot (1 \ 1 \ 0)$. Segundo a nomenclatura do item precedente, $a=(1/\sqrt{2})$, $b=(1/\sqrt{2})$, $c=(0/\sqrt{2})$. Assim, temos:

Exemplo:

A matriz de rotação é dada por $[I] + \text{sen}\theta \cdot [L] + (1 - \text{cos}\theta) \cdot [L]^2$;

matriz de rotação em torno do eixo \vec{e} .

$$[\text{Rot } \vec{e}] = e^{[L]\theta} = [I] + \text{sen}\theta \cdot [L] + (1 - \text{cos}\theta) \cdot [L]^2 \quad (2.15)$$

Agora que determinamos um operador análogo ao (j), basta determinarmos um análogo a um número complexo "genérico" $e^{j\theta}$ (até agora tratamos apenas de $e^{j(\pi/2)}$). Basta fazermos:

$$\left\{ \begin{array}{l} [L]^2 = [L] \\ [L]^3 = -[L] \\ [L]^4 = [L] \\ [L]^5 = -[L] \\ [L]^6 = [L] \\ [L]^7 = -[L] \\ [L]^8 = [L] \end{array} \right. \left\{ \begin{array}{l} j^2 = -j \\ j^3 = -j \\ j^4 = -j^2 \\ j^5 = j \\ j^6 = j^2 \\ j^7 = -j \\ j^8 = -j \end{array} \right. \text{ segue o mesmo padrão de}$$

2.4 Reflexão

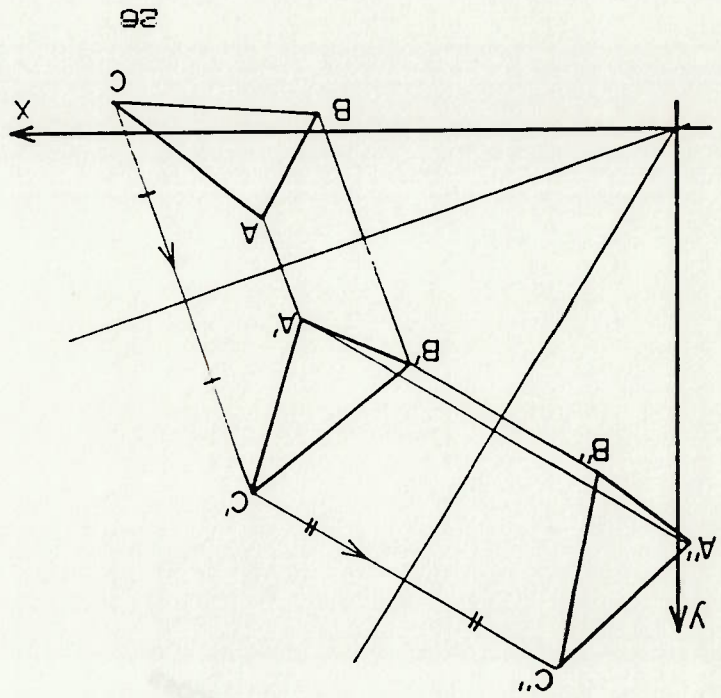
Doi, usando (2.15),

$$[Rot \theta] = [R] + \cos \theta [L] + (1 - \cos \theta) [L]^2$$

$$[Rot \theta] = \begin{bmatrix} 0.911 & 0.333 & -0.244 & 0 \\ -0.244 & 0.911 & 0.333 & 0 \\ 0.333 & 0.333 & 0 & 0 \\ 0 & 0.9108 & 0 & 1 \end{bmatrix}$$

Estudaremos agora a operação de reflexão de um objeto tridimensional em relação a um plano no espaço tridimensional (no 2D seria uma reta e no 4D em relação a um hiper-plano) qualquer. Esta transformação é importante por dois motivos. O primeiro é o seu extensivo uso em modelagem geométrica. Para construirmos um sólido simétrico, basta a descrição de apenas uma metade. O segundo é que conceitualmente, esta transformação é a unidade básica para a construção das matrizes de rotação e de translação. De fato, suponha o triângulo mostrado na fig. 2.10. Duas reflexões consecutivas, em relação aos eixos indicados, é equivalente a uma única rotação em torno do eixo Oz. A figura 2.11 mostra o caso da translação. Duas reflexões consecutivas em relação aos eixos indicados é equivalente a uma única translação na direção das normais aos eixos.

FIGURA 2.10
Duas reflexões é equivalente a uma rotação.



Suponhamos que seja feita a operação de reflexão do ponto $P(x_0, y_0, z_0)$ em relação a um plano α passando pela origem como mostra a fig. 2.12.

Veremos como deve ser a matriz [RX] para efetuar uma reflexão em relação a um plano qualquer do espaço tridimensional. Antes porém convém um pequeno lembrete. Como já vimos, a operação de translação requer uma matriz 4×4 . Para mantermos a compatibilidade, todas as matrizes de transformação devem ser formuladas de modo a possuir essa ordem. Inicialmente, iremos estudar um caso simples, onde o plano passa pela origem. em seguida iremos retirar essa restrição.

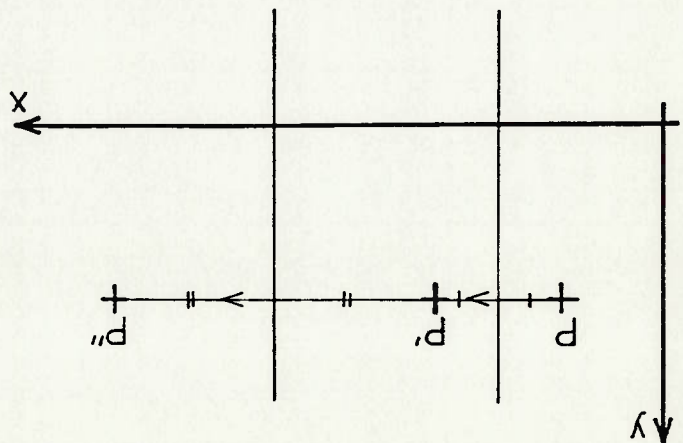


FIGURA 2.11
Duas reflexões consecutivas é igual a uma única translação.

$$d = \frac{\sqrt{(a^2 + b^2 + c^2)}}{a \cdot x + b \cdot y + c \cdot z + e}$$

6 a) distância de um ponto $P(x^0, y^0, z^0)$ a um plano α ($a \cdot x + b \cdot y + c \cdot z + e = 0$) é dada por:

01. (fig. 2.13).
 A normal ao plano α pode ser dada por (sendo o

b) Matriz de reflexão em relação ao plano α .

$$[R_X \ XZ] = [I] - 2 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

matriz de reflexão $[R_X \ XZ]$ pode ser construída da seguinte forma: A normal ao plano XZ é dada por $(0 \ 1 \ 0 \ 0)$. Nesse modo, a

a) Matriz de reflexão em relação ao plano XZ :

Verifiquemos agora com um pequeno exemplo, que com duas reflexões, uma em relação ao plano XZ e outra em relação a um plano α que passe pela origem, é perpendicular ao plano XY , e faz um ângulo de 30 graus com a horizontal, é possível se construir uma única operação de rotação em torno do eixo Z de 60 graus.

Exemplo:

$$\text{ou, } [P] = [P^0] \cdot [R_X] \text{ onde } [R_X] = [I] - 2 \cdot [N]^T [N]$$

$$[P] = [P^0] - 2 \cdot [P^0] [N]^T [N], \text{ onde } \begin{cases} [N] = [a \ b \ c \ 0] \\ [P^0] = [x^0 \ y^0 \ z^0 \ 1] \\ [P] = [x \ y \ z \ 0] \end{cases} \quad (2.16)$$

forma (6):

Matricialmente essa operação poderia ser efetuada da seguinte

$$\left\{ \begin{array}{l} P = P^0 - 2 \cdot d \cdot \hat{n}, \text{ onde} \\ \hat{n} = (a, b, c) \text{ é a normal ao plano } \alpha \\ \text{com } \|(a, b, c)\| = 1 \\ d = \text{é a distância do ponto } P \text{ à } \alpha \end{array} \right.$$

Vetorialmente, a nova posição do ponto deve ser dada por:

A relação matricial é análoga à feita acima, apenas devemos notar que agora o termo independente "e" de $\alpha (a \cdot x + b \cdot y + c \cdot z + e = 0)$ não é nulo. Assim a distância de $[P_0]$ à α não pode ser dada sim-

Continuamos usando a relação vetorial $\vec{P} = \vec{P}_0 + 2 \cdot d \cdot \vec{n}$.

Notemos que a transformação acima não permite a reflexão de um ponto em relação a um plano que não passa pela origem. Assim, uma operação de transformação não poderia ser obtida a partir da matriz acima. Veremos agora como podemos fazer a reflexão em relação a um plano qualquer. Para isso, devemos reconhecer que um sistema tridimensional é incapaz de realizar a transformação li- nearmente (pois transformação esta involvida).

Compare com a matriz obtida acima com (2.8).

$$[T] = [R_X] [O] [R_X] [XZ] = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [Rot \ z \ \theta] //$$

c) Finalmente, concatenando as duas operações chegamos a:

$$[R_X] [O] = [I] - 2 \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 - 2 \cdot \cos \theta & -2 \cdot \sin \theta & 0 & 0 \\ 2 \cdot \sin \theta & 1 - 2 \cdot \cos \theta & 0 & 0 \\ 0 & 0 & 1 - 2 \cdot e \cdot s & 0 \\ 0 & 0 & 0 & 1 - 2 \cdot e \cdot s \end{bmatrix} = \begin{bmatrix} 1 - 2 \cdot \cos \theta & 0 & 0 & 0 \\ 0 & 1 - 2 \cdot \cos \theta & 0 & 0 \\ 0 & 0 & 1 - 2 \cdot e \cdot s & 0 \\ 0 & 0 & 0 & 1 - 2 \cdot e \cdot s \end{bmatrix}$$

Desse modo, a matriz de reflexão $[R_X]$ pode ser construída da seguinte forma:

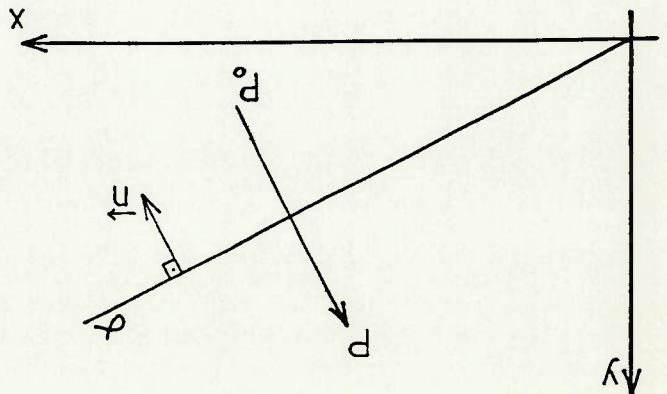


FIGURA 2.12
Reflexão de um ponto P_0 em relação a um plano α .

$$[RX1] = [11-2] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

A primeira reflexão é dada por

- i) Reflexão em relação a um plano que passa por (0,0,0) e tem normal (1,0,0).
- ii) Reflexão em relação a um plano que passa por (3,4,2) e tem normal (1,0,0).

Vamos verificar com um pequeno exemplo, que uma operação de translação pode ser construída a partir de duas operações de reflexão. Digamos que desejamos trasladar o ponto (-2,1,0) para uma posição (4,1,0) ou seja, um deslocamento dado pelo vetor (6,0,0). As operações de reflexão (arbitariamente escolhidas) são:

Exemplo

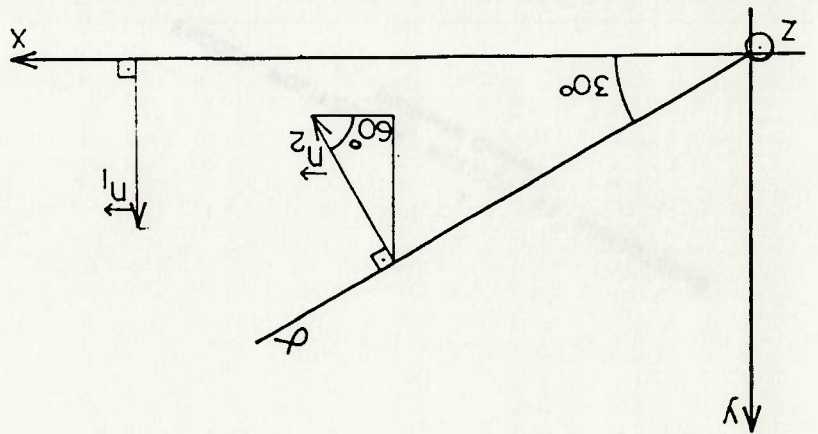


FIGURA 2.13
Determinação da normal ao plano α , que faz um ângulo de 30 graus com o plano horizontal.

com $[P] = [x \ y \ z \ 1]$ e $[P_0] = [x_0 \ y_0 \ z_0 \ 1]$

$$[RX] = [11-2] \cdot \begin{bmatrix} a \\ b \\ c \\ e \end{bmatrix} = [a \ b \ c \ 0]$$

(2.17)

$[P] = [P_0][RX]$ onde

$$[P] = [P_0] - 2 \cdot [P_0] \cdot \begin{bmatrix} a \\ b \\ c \\ e \end{bmatrix} \cdot [a \ b \ c \ 0]$$

zero pois a normal ao plano não deve ser componente na direção OY.

transformação é :
puesmente por $d = [P_0][a \ b \ c \ 0]^T$. Se agora incluímos uma quarta coordenada, d pode ser dada por $[P_0][a \ b \ c \ e]^T$. Dessa maneira, a

Vimos nesse capítulo as principais matrizes de transformação utilizadas na representação computacional gráfica. Muitas outras matrizes podem ser obtidas da mesma maneira que usamos nesse capítulo. De fato, qualquer transformação linear pode ser colocada na forma matricial. No capítulo seguinte veremos algumas aplicações para as matrizes aqui obtidas, em especial a obtenção de perspectivas, que consiste na principal atividade de um pré/pós processador gráfico de malhas de elementos finitos.

2.6 Fechamento do capítulo

$$[H] = \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

Suponhamos uma transformação que realize a operação de homotetia em relação à origem (0,0,0,0). Essa transformação é caracterizada por $F: \mathbb{R}^3 \rightarrow \mathbb{R}^3$, $F(x,y,z) = (k \cdot x, k \cdot y, k \cdot z)$. Por conveniência, $k=1$. Matricialmente a operação fica (considerando a base usual):

2.5. Homotetia.

De fato, [T] é a matriz de translação correspondente ao vetor (0,0,0) //

$$[T] = [RX][RXZ] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A combinação das duas reflexões é dado pelo produto:

$$[RXZ] = [I]-2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A segunda:

3. PERSPECTIVAS ANALITICAS

3.0. Introdução

Neste capítulo estudaremos quais as principais maneiras de obter imagens tridimensionais de um conjunto de pontos, em especial, de uma malha de elementos finitos. As ferramentas básicas já foram vistas no capítulo precedente. Começamos com a classificação das perspectivas, e em seguida, aplicaremos as transformações lineares para obtê-las. Deve-se salientar que, provavelmente, este é o principal assunto quando lidamos com pré e pós processamento gráfico de dados para o método dos elementos finitos, já que é através dele é que erros grosseiros na malha podem ser detectados.

3.1. Tipos fundamentais de perspectivas

Perspectivas nada mais são que projeções particulares, e ai dois elementos são fundamentais em seu estudo. O plano de projeção α e o centro de projeção O . O modelo geométrico será baseado na geometria euclidiana.

Obtém-se a projeção de um ponto P no quadro α a partir do centro de projeção O , determinando-se a intersecção entre a reta $P0$, denominada reta projetante [DES], e o plano α (Fig.3.1). O ponto obtido P' é a projeção de P no plano α sendo o observador O . Entendendo a noção para um objeto, podemos afirmar que sua projeção é o conjunto de todas as projeções de seus pontos. Sua obtenção prática fica facilitada se notarmos que um ponto se projeta em um ponto, uma reta se projeta em

Existem os casos particulares, mas não precisamos nos preocupar com eles, pois eles são tratados de maneira natural pelas técnicas que iremos apresentar.

Dizemos que o centro de projeção é impróprio quando ele está infinitamente distante do plano de projeção e do objeto. Assim, as retas projetantes correspondentes a diferentes pontos do objeto devem ser paralelas. Temos agora uma subdivisão na nossa classificação. Se os raios visuais forem ortogonais ao quadro, a projeção é dita *projeção paralela ortogonal* ou *projeção cilíndrica ortogonal*, caso contrário, ela é dita *pro-*

3.1.1. Projeções com centro de projeção impróprio - Perspectivas cilíndricas.

Dependendo da posição relativa entre O , α , e o objeto, poderemos ter diferentes classes de projeções. Aquelas projeções de objetos que permitirem a visualização tridimensional, chamaremos de perspectivas. Conceitualmente, quando tratamos de perspectivas, o centro de projeção O passa a ser denominado *observador*; o plano α , de *quadro*; e a reta projetante, de *raio visual*.

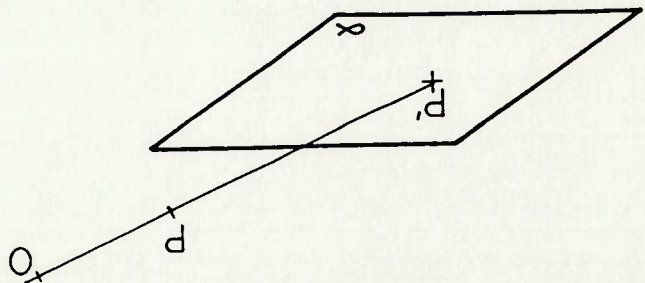
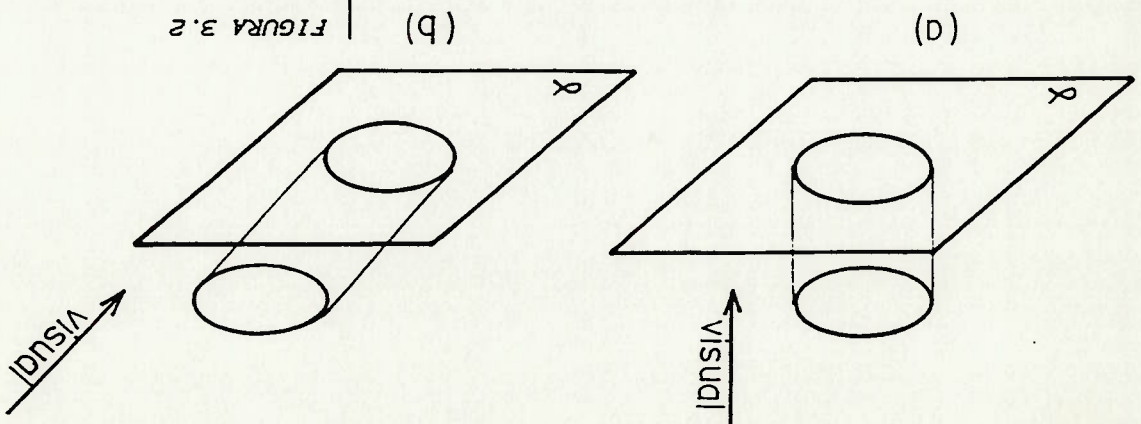


FIGURA 3.1
A projeção do ponto P no plano α segundo o observador O é o ponto P'.

uma reta, um plano se projeta em um plano, etc. Assim, por exemplo, basta projetarmos apenas dois pontos de uma reta para que sua projeção seja determinada.



Ocorre a projeção cilíndrica ortogonal quando os raios visuais chegam perpendicularmente ao quadro. Caso contrário, temos a projeção cilíndrica oblíqua.

FIGURA 3.2

Aquelas projeções cilíndricas ortogonais que permitem a sensação visual de tridimensionalidade, são chamadas perspectivas axonométricas. Da mesma maneira perspectivas cavalieiras associam-se às projeções cilíndricas oblíquas.

3.1.2 Projeções com centro próprio - Perspectivas cônicas.

Dizemos que o centro de projeção é próprio quando ele está a uma distância finita do plano de projeção e do objeto. Neste caso, as retas projetantes não são paralelas, elas convergem para o centro de projeção. A projeção obtida recebe o nome de projeção cônica, e a perspectiva associada, de perspectiva cônica.

3.1.3 Classificação das perspectivas.

Algumas perspectivas recebem nomes particulares. Para facilitar nossa classificação imaginemos um cubo unitário alinhado com os eixos coordenados (base usual⁽²⁾). As perspectivas são :

a) Perspectivas axonométricas

i) Isométrica

Na perspectiva isométrica, o quadro e o observador são posicionados de forma que a projeção do cubo possua todos os lados encurtados de um mesmo fator. Curiosamente, a perspectiva externa do cubo é um simples hexágono [DES]. Figura 3.3

ii) Dimétrica (ou monodimétrica)

Na dimétrica, as arestas correspondentes a duas direções, por exemplo Ox e Oy sofrem reduções de um mesmo fator, enquanto as arestas paralelas à Oz sofrem redução de um fator diferente. Figura 3.4.

iii) Trimétrica (ou anisométrica)

Agora, todas as arestas quando projetadas sofrem reduções de fatores diferentes. Praticamente todas as perspectivas obtidas por um programa usando computação gráfica são trimétricas. Figura 3.5.

²Veja apêndice sobre álgebra linear

As perspectivas cavaleiras não apresentam subdivisões marcantes. Somente deve-se salientar alguns itens da nomenclatura do desenho técnico. Tomemos novamente o cubo alinhado com os eixos coordenados como no item anterior. Note a posição relativa entre quadro, objeto e observador. Sua perspectiva cavaleira é mostrada na fig. 3.6. A projeção do eixo Oz é denominada eixo *das fugantes* e o ângulo α que ele faz com a horizontal, *ângulo das fugantes*. O fator de ampliação das arestas paralelas ao eixo Oz recebe o nome de *fator de deformação k*. Note que o ângulo α e o eixo das fugantes são independentes entre si. Isso significa que seria correto do ponto de vista geométrico termos por exemplo, $k=1000$ e $\alpha=30^\circ$ (projeção do cubo com imensa distorção).

c) Perspectivas cônicas

A distinção que se faz entre as perspectivas cônicas se refere ao número de pontos de fuga da figura projetada. Retomemos o nosso cubo alinhado com os eixos coordenados. Na perspectiva cônica com um único ponto de fuga, as arestas paralelas a uma direção somente, por exemplo paralelo à Oz, convergem para um único ponto. Na perspectiva com dois pontos de fuga, arestas correspondentes a duas direções, por exemplo Oz e Ox, convergem para dois pontos diferentes. No caso mais geral, temos três pontos de fuga correspondendo às três direções coordenadas [DES].

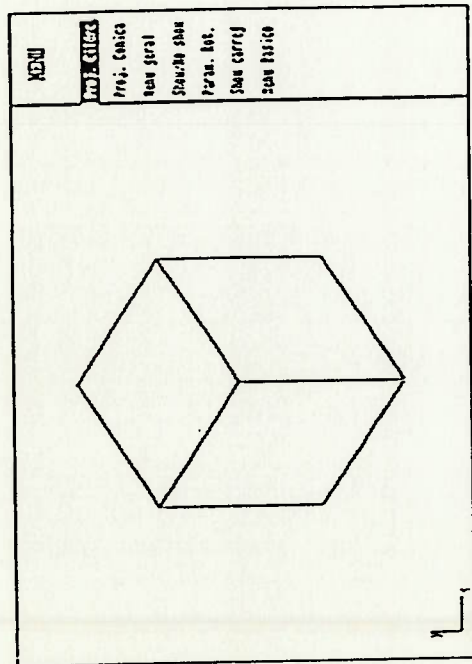


Figura 3.3
Perspectiva Isométrica

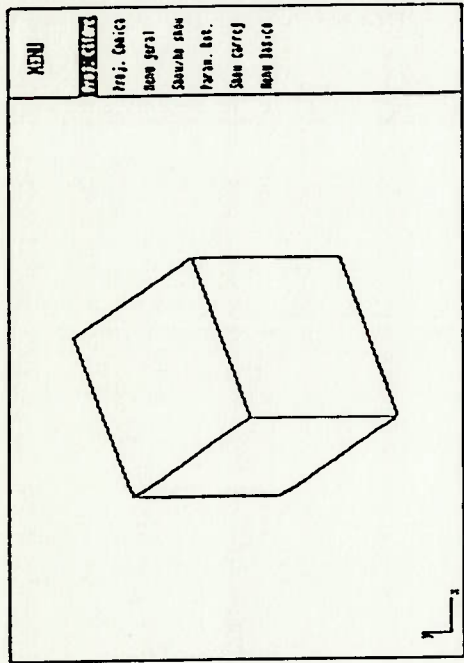


Figura 3.4
Perspectiva Dimétrica

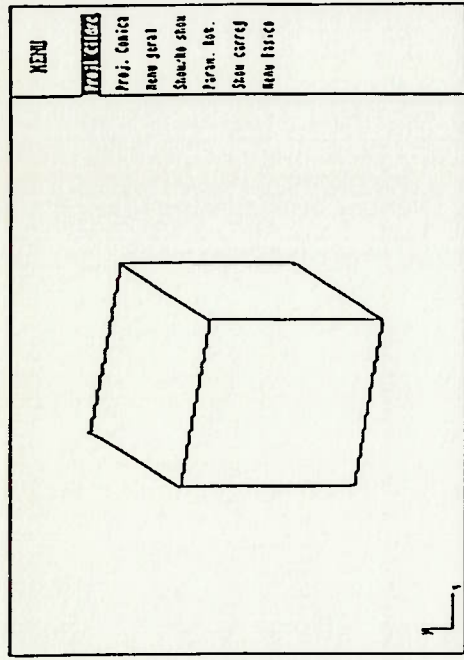
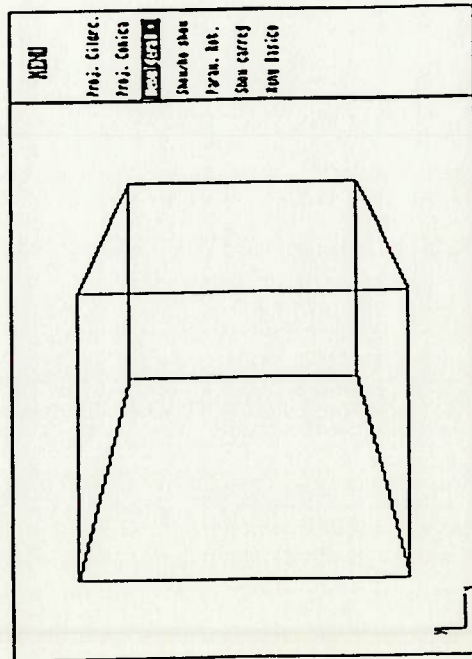
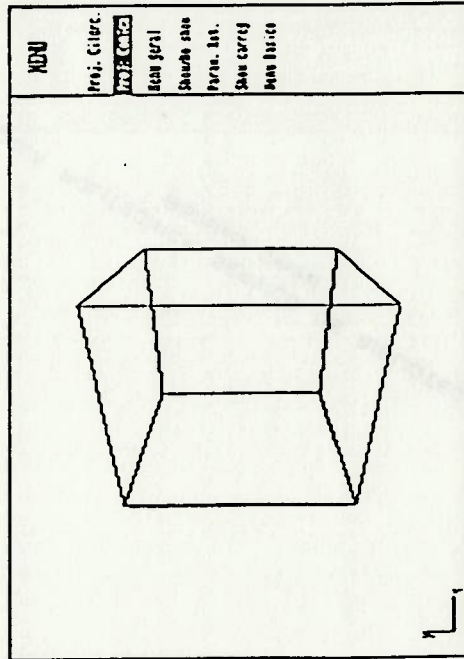


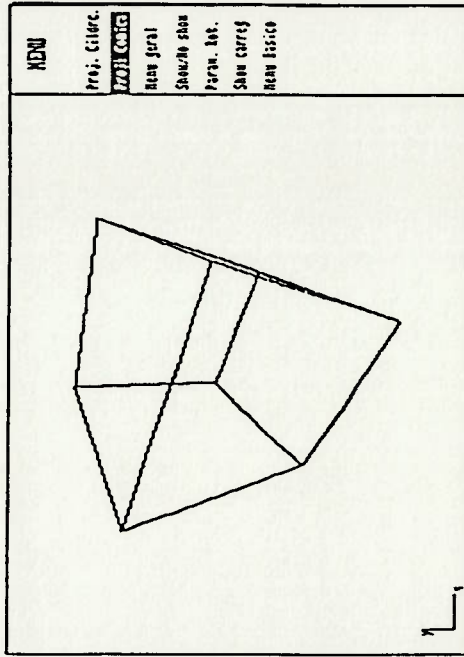
Figura 3.5
Perspectiva Trimétrica



Perspectiva cônica - um ponto de fuga



Perspectiva cônica - dois pontos de fuga



Perspectiva cônica - três pontos de fuga

É interessante notar que pessoas com diferentes formas tem mais afinidade com um tratamento do que com outros. Um engenheiro civil por exemplo, prefere a mudança do observador, enquanto uma outra pessoa com formação mais mecânica dá preferência à mudança da posição do objeto.

Veremos agora como obter as perspectivas com equivalências matriciais vistas no capítulo anterior. Três tratamentos diferentes serão utilizados: Mudança da posição do objeto, mudança da posição do quadro de projeção e mudança da posição do observador. Veremos que cada tratamento está associado um sistema de coordenadas próprio, e as respectivas coordenadas: Coordenadas do objeto, do quadro de projeção e do observador.

3.2 Obtenção das perspectivas usando operações lineares.

FIGURA 3.6
Posições relativas entre observador, quadro e objeto na perspectiva cavaleira.

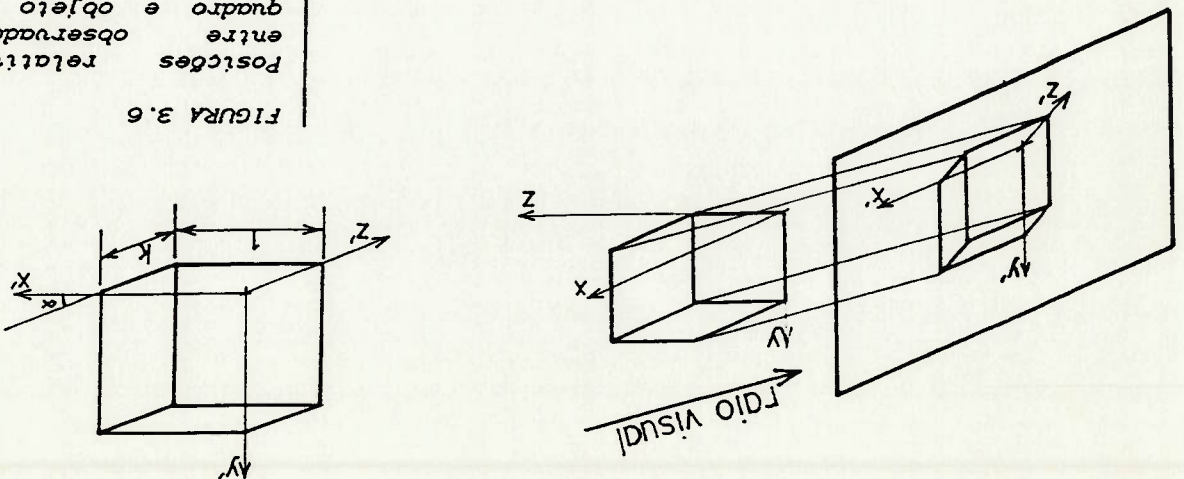
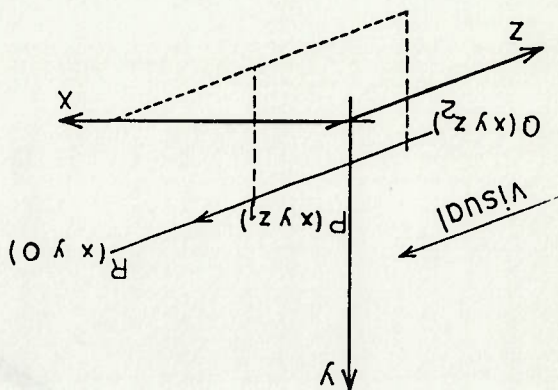


FIGURA 3.7
As projeções de P e Q
são coincidentes no
quadro.



Suponha que nessa situação, temos dois pontos $P(x, y, z^1)$ e $Q(x, y, z^2)$ com $z^1 \neq z^2$. Consideremos as projeções desses dois pontos. Naturalmente, as projeções serão coincidentes em um único ponto $R(x, y, 0)$ (Fig. 3.7). Isto é, para projetarmos um ponto sobre o quadro xy podemos simplesmente eliminar a coordenada z. Do ponto de vista físico, seria o mesmo que achatarmos um sólido contra o plano xy. Assim, para obtermos uma perspectiva axonométrica de um sólido, uma alternativa seria rotacioná-lo até uma posição conveniente e depois projetá-lo no quadro xy (eliminando a coordenada z dos pontos do objeto). O nosso problema passa a ser então determinar qual é essa rotação conveniente. O caso mais simples é o da trimétrica: Devido à sua sua generalidade, muito provavelmente, qualquer sequência de rotações seguida de projeção serviria. Veremos agora como resolver o caso mais difícil: obter uma perspectiva isométrica.

Como já vimos, as posições do quadro e do observador são extremamente importantes na obtenção de perspectivas. Neste item, iremos adotar o quadro de projeção sendo o plano xy, e o observador no infinito, na direção positiva de Oz. Assim, os raios visuais chegam perpendicularmente ao quadro.

3.2.1. Obtenção das perspectivas axonométricas usando-se matrizes de rotação (mudança da posição do objeto).

Seja um cubo de lado unitário com suas arestas alinhadas com os eixos coordenados (fig. 2.1). Nosso objetivo agora é determinar uma transformação linear (matricial) que leve esse cubo à sua perspectiva isométrica. Para isso, devemos notar que a diagonal AB deve ser visualizada como sendo apenas um ponto. Para que isso ocorra, a aresta AB do cubo rotacionado deve ficar perpendicular ao quadro xy. Vejamos agora quais são as rotações necessárias:

- 1) Rotação em torno do eixo Oy
- Inicialmente, faremos com que o segmento AB fique sobre o plano yz. Para isso, é necessária uma rotação de -45° em torno do eixo Oy.
- 11) Rotação em torno do eixo Ox

Após a rotação acima ter sido efetuada, devemos fazer com que AB se alinhe com o eixo Oz. O ângulo necessário é de $35,26^\circ$ como pode ser evidenciado pelo triângulo retângulo ABC da fig. 2.8.

Concatenando-se as operações acima, temos:

$$[T_{-150}] = \begin{bmatrix} \cos \theta & 0 & 0 & 0 \\ \sin \theta & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\sin \theta & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & 0 & 0 \\ 0 & -\sin \phi & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \cos \phi & 0 & 0 \\ 0 & \sin \phi & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde $\theta = -45^\circ$ e $\phi = 35,26^\circ$.

Realizando as operações chegamos à:

$$[T_{-150}] = \begin{bmatrix} 0,7071 & -0,4082 & 0 & 0 \\ 0 & 0,8166 & 0 & 0 \\ -0,7071 & -0,4082 & 0 & 0 \\ 0 & 0,5773 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Após a rotação tiver sido efetuada, cada ponto do sólido deverá ser projetado. Como já vimos, para isso basta eliminarmos a coordenada z de cada um deles. Como uma reta se transforma em uma reta, são necessários apenas dois pontos de cada aresta para que sua projeção seja determinada. Note que estamos usando

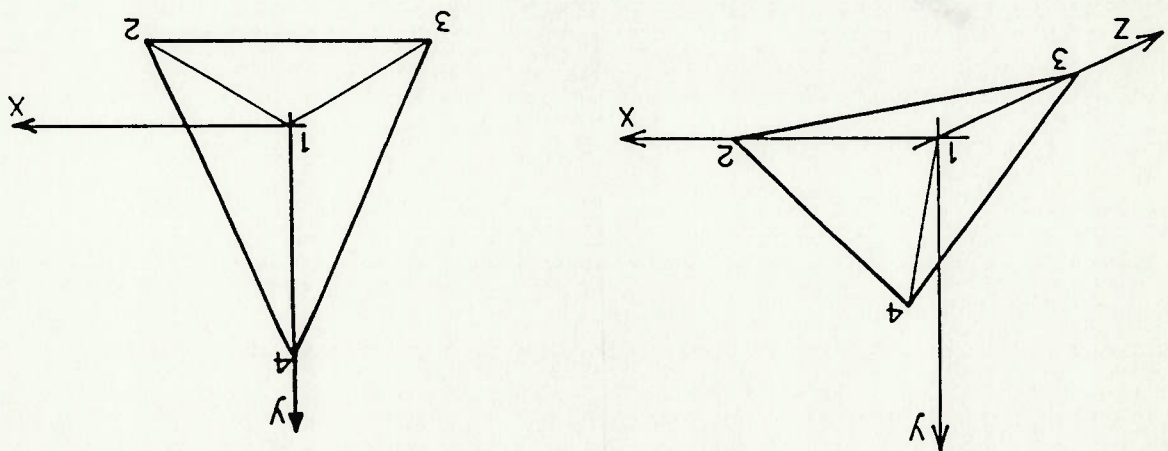
apenas um sistema de coordenadas, o associado ao objeto. Essas coordenadas são denominadas *coordenadas do objeto*.

Exemplo:

Vamos obter agora uma perspectiva isométrica de um tetraedro (envolvido por um paralelepípedo imaginário). As coordenadas homogêneas dos vértices do sólido são (fig. 3.8):

- 1 + (0 0 0 1)
- 2 + (1 0 0 0 1)
- 3 + (0 0 1 0 1)
- 4 + (1 1 0 0 1)

FIGURA 3.8
Perspectiva isométrica do tetraedro 1 2 3 4.



A matriz de transformação para a obtenção da perspectiva isométrica é segundo (3.1):

$$[T-isom] = \begin{bmatrix} 0.7071 & -0.4082 & 0 & 0 & 0 \\ 0 & 0.8166 & 0 & -0.7071 & -0.4082 \\ 0 & 0.5773 & 0 & 0 & 0 \\ 0 & 0.5773 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Apenas por uma questão operacional, vamos agrupar as coordenadas dos vértices em uma única matriz [V]:

$$[V] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Agora, determinamos as novas coordenadas dos vértices fazendo [V][T-isom]:

$$[V][T-isom] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.7071 & -0.4082 & 0 & -0.7071 \\ 0 & 0.8166 & 0 & -0.4082 \\ 0 & 0.5773 & 0 & -0.4082 \\ 0 & 0 & 1 & -0.4082 \end{bmatrix}$$

O quadro de projecção é um plano que pode ser gerado por uma base de dois vetores (e_1, e_2) não linearmente dependente. e_1 e e_2 estão contidos no plano, e não são necessariamente ortogonais. Os pontos da figura projetada serão escritos em função

não aparece explicitamente neste item).
 observador é repositionado (o conceito de centro de projecção devem chegar ortogonalmente ao quadro, fica subentendido que o do quadro de projecção. É bom lembrar que como os raios visuais podemos obter uma perspectiva axonométrica através da mudança [NUM]. Com uma ligeira adaptação, apresentaremos agora como numérico quando se estuda solução de equações superdeterminadas Este problema, em geral, é tratado em livros de cálculo

3.3.2. Obtenção de perspectivas axonométricas através da mudança do quadro de projecção.

Para obtermos outras perspectivas, o processo é análogo. Perceba por exemplo que se o ângulo ϕ acima fosse outro não igual a 35.26° , a perspectiva obtida seria uma dimétrica, e se ao invés de usarmos apenas duas rotações, usarmos três e permitirmos livre escolha de ângulos de rotação, em geral obteremos perspectivas trimétricas.

1 + x=0	y=0
2 + x=7.07	y=-4.08
3 + x=-7.07	y=-4.08
4 + x=0	y=11.55

Embora tenhamos conseguido três coordenadas (x, y, z) para cada ponto transformado, devemos ignorar a terceira coordenada, pois estamos fazendo uma projecção ortogonal sobre o plano xy. Desta maneira, as coordenadas finais da nossa perspectiva isométrica serão:

A primeira linha de [V] se transforma na primeira linha de [V1T-iso], a segunda na segunda, e assim por diante. Note que [V] não precisa ser quadrada.

dessa base de vetores, em outras palavras, serão escritos como combinações lineares deles', sendo que os coeficientes de e_1 e e_2 serão as coordenadas dos pontos projetados. Denominaremos essa coordenadas de coordenadas do quadro.

Seja agora uma matriz $[A]$ onde suas linhas são exatamente os vetores e_1 e e_2 . Notar que se e_1 for um vetor de ordem 4, $[A]$ será uma matriz 2×4 .

Imaginemos agora o vetor $[V] = [P][A] = [x \ y] \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$. Perceba

que claramente $[V]$ é uma combinação linear dos vetores da base $\langle e \rangle$. Vamos impor agora que $[V]$ seja a projeção ortogonal de um ponto $[Q]$ (pertencente ao espaço quadridimensional) sobre o espaço gerado por $\langle e \rangle$ (e daí $[P]$ fornece as coordenadas da projeção do ponto relativamente à base $\langle e \rangle$). Para que isso aconteça, o vetor $[V] - [Q]$ deve ser perpendicular a qualquer vetor $[U]$ do espaço gerado por $\langle e \rangle$ (Fig. 3.9):

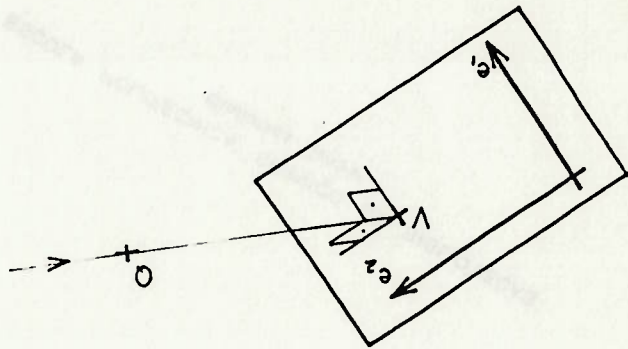


FIGURA 3.9
O vetor $[V] - [Q]$ é perpendicular ao espaço gerado pela base $\langle e \rangle$.

$$\begin{aligned} C [V] - [Q] \cdot C [U] [A]^T &= 0 \\ C [P] [A] - [Q] \cdot C [A]^T [U]^T &= 0 \\ C [P] [A] [A]^T - [Q] [A]^T \cdot C [U]^T &= 0 \end{aligned}$$

Assim,

$$[P] = [Q] [A]^T [A] [A]^T^{-1} \quad (3.2)$$

Ver apêndice sobre álgebra linear

5/3 apenas para normalizar o vetor.

$$\vec{e}_1 = (0, 1, 0) \quad \vec{n} = (0, 707, 0, -0, 707) \\ \vec{e}_2 = \vec{n} \wedge \vec{e}_1 = (-0, 408, 0, 816, -0, 408)$$

(fig. 3.10):

Veremos agora um exemplo um pouco mais sofisticado. Vamos obter uma matriz para perspectiva isométrica. Novamente em primeiro lugar, devemos determinar uma base ortonormal que gere o quadro de projeção. Nosso quadro de projeção será um plano que possui normal $\vec{n} = (1, 1, 1)/\sqrt{3}$. A base de vetores (\vec{e}_1, \vec{e}_2) será obtida da seguinte maneira

Verifique que um ponto $[P] = [a, b, c, 1]$ é transformado para $[a, b, c, 1/A] = [a, b, c]$, como já era de se esperar.

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

ortogonal e simplesmente:

1) Determinaremos agora a matriz de projeção ortogonal sobre o plano xy. Iremos usar esse exemplo como um teste de verificação, e apreciação, da simplicidade do método, já que o resultado é intuitivamente conhecido. A primeira providência a ser tomada é determinarmos uma base de vetores ortonormal que gere o plano xy. Essa base é $(1, 0, 0, 0)$ e $(0, 1, 0, 0)$. Desta maneira, a matriz de projeção ortogonal é simplesmente:

Veremos dois exemplos.

Exemplos:

Note a simplicidade! $[A]^T$ é uma matriz onde suas colunas são respectivamente \vec{e}_1 e \vec{e}_2 ;

$$[P] = [Q][A]^T$$

↙ ponto a ser projetado
↘ ponto projetado

(3.3)

Suponha agora que a base $\{e\}$ é ortonormal (o que é desejável). Neste caso, $[A][A]^T = [I]$. Daí, a operação acima fica simplificada para

Chegamos à terceira forma de se obter uma perspectiva axonométrica - através da mudança do quadro de projeção. Esta maneira de se obter perspectivas é a mais sofisticada, requerendo a aplicação de três sistemas de referência simultaneamente fig. 3.11:

3.2.3. Obtenção de perspectivas axonométricas através da mudança do centro de projeção

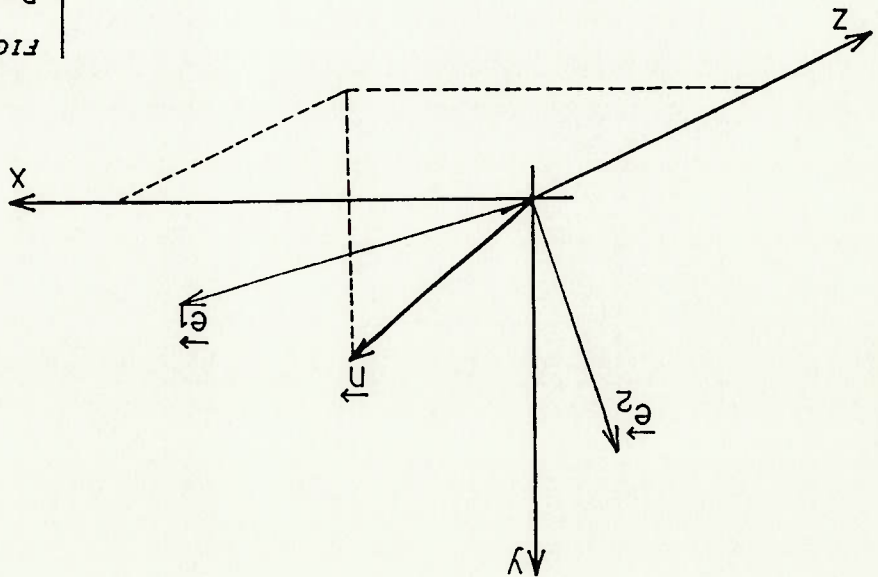
$$[T^{-1} \cdot \text{iso}] = [A]^{-1} = \begin{bmatrix} 0.207 & 0 & 0 \\ -0.207 & 0 & 0 \\ -0.408 & 0.816 & 0 \\ 0 & -0.408 & 0 \end{bmatrix} \cdot \text{Compare com (3.1) !!}$$

Assim a matriz de projeção ortogonal isométrica fica:

$$[e_1] = [0.207 \ 0 \ -0.207 \ 0] \\ [e_2] = [-0.408 \ 0.816 \ -0.408 \ 0]$$

Colocando os vetores em coordenadas homogêneas:

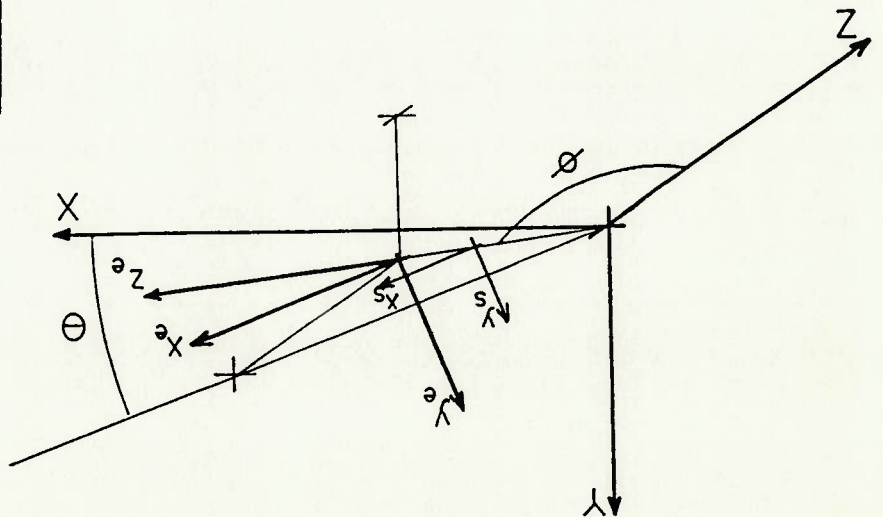
FIGURA 3.10
Base ortônoma de vetores (e_1, e_2, e_3)



O quadro de projeção será colocado a uma distância d do observador e perpendicularmente à linha l que liga o observador e a origem do sistema coordenado do objeto. O sistema do observador x^o, y^o, z^o estará sobre O e com seu eixo Oz^o alinhado com Oz . Para determinarmos as coordenadas da projeção de um ponto em relação ao sistema associado ao quadro de projeção, devemos seguir o seguinte roteiro:

Suponhamos o observador localizado sobre um ponto O (a, b, c) dado em coordenadas do objeto. Para facilitar, vamos localizar o ponto (a, b, c) através de um sistema de coordenadas esférico, como é indicado na figura 3.12.

- a) Um sistema associado ao objeto (x y z)
- b) Um sistema associado ao quadro de projeção (x^e y^e)
- c) Um sistema associado ao observador (x^o y^o z^o)



Três sistemas de referência: Do observador, do objeto e do quadro.

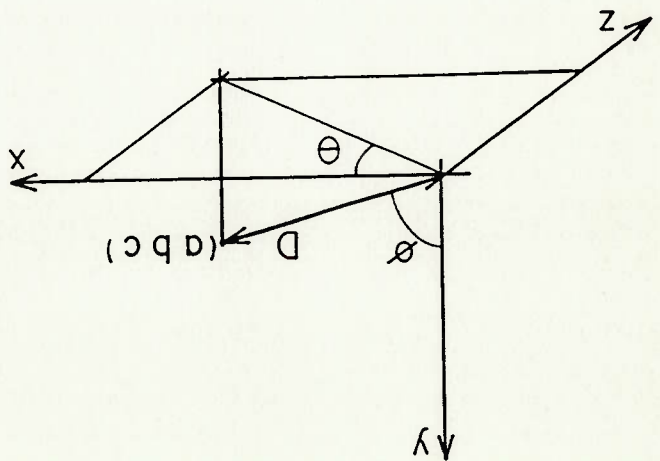
FIGURA 3.11

Essa é uma interpretação. Lembra-se do item 2.0, que afirma que temos duas interpretações diferentes, uma com referencial no objeto e outra com ele no próprio sistema de coordenadas.

Vamos escrever as coordenadas dos pontos de um sólido, que estão referenciadas inicialmente ao sistema do objeto, em função do sistema coordenado do observador. A ideia básica é levar o sistema do objeto para sobre o sistema do observador, conservando as posições dos pontos do objeto inalteradas. Dessa maneira, as coordenadas do objeto seriam escritas em função da nova posição do sistema de eixos, que como já dissemos, coincide com o sistema do observador. Para levarmos o sistema do objeto para a sua nova posição, iremos seguir o roteiro (acompanhe a figura 3.13):

- 1) Determinar as coordenadas dos pontos do objeto em relação ao sistema do observador;
- 1.1) Determinar as coordenadas dos pontos projetados em relação ao sistema coordenado do quadro.

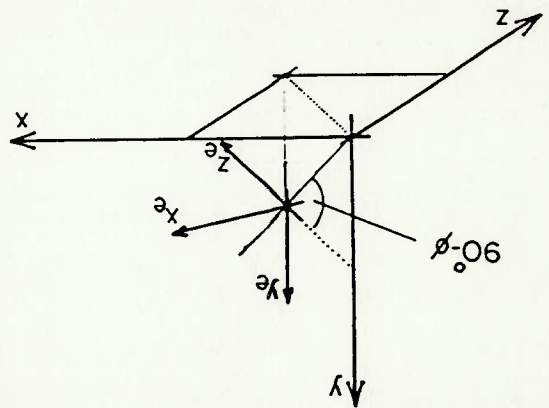
FIGURA 3.12
Posição do observador
dado em coordenadas
esféricas.



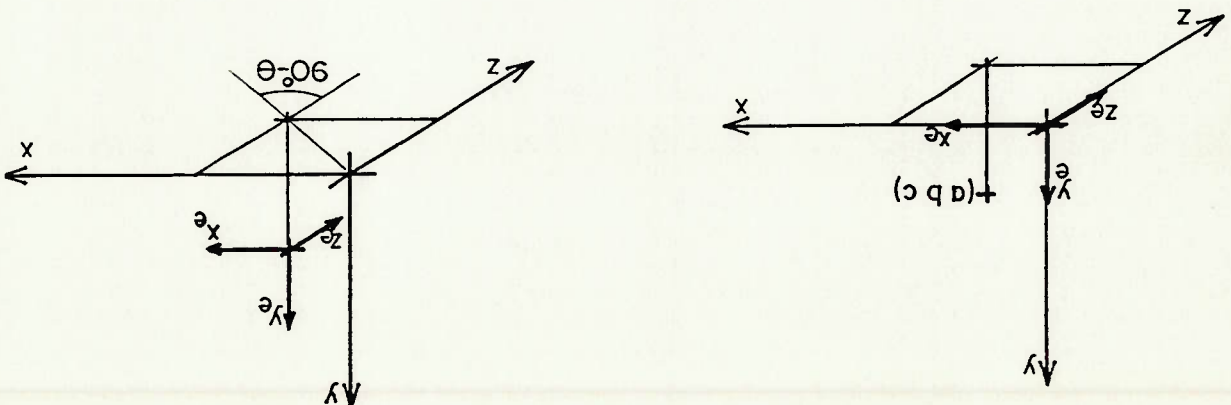
Relembre-se que temos sempre duas interpretações para as transformações geométricas (item 2.0). Então, afirmar que o sistema é trasladado de um vetor (a,b,c) e os pontos se mantem imóveis, é o mesmo que dizer que os pontos se trasladam de um vetor (-a,-b,-c) e o sistema se mantém imóvel. A matriz de transformação é:

$$[Trans] = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

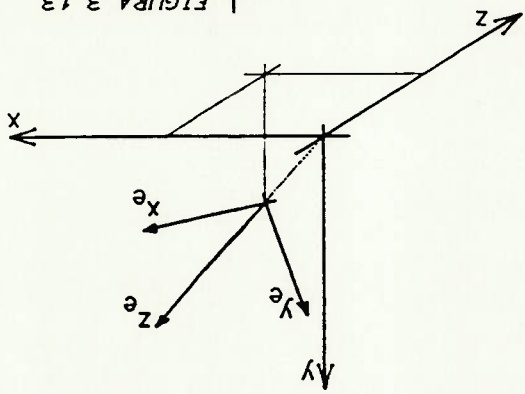
i) Translação da origem do sistema para (a,b,c).



I



II



III

Rotatio para posicio-
namento do sistema
coordenado do obser-
vador.

FIGURA 3.13

$$\begin{aligned} \cos\theta &= \frac{\sqrt{a^2+z^2}}{a} & \cos\phi &= \frac{\sqrt{a^2+b^2+z^2}}{b} \\ \text{sen}\theta &= \frac{\sqrt{a^2+z^2}}{c} & \text{sen}\phi &= \frac{\sqrt{a^2+b^2+z^2}}{\sqrt{a^2+z^2}} \end{aligned}$$

As funções trigonométricas são dadas por:

$$[M] = \begin{bmatrix} \text{sen}\theta & 0 & 0 & 0 \\ -\text{cos}\theta & 0 & 0 & 0 \\ -\text{cos}\theta \cdot \text{cos}\phi & \text{sen}\phi & \text{cos}\theta \cdot \text{sen}\phi & 0 \\ -\text{sen}\theta \cdot \text{cos}\phi & \text{sen}\theta \cdot \text{sen}\phi & -\sqrt{a^2+b^2+z^2} & 1 \end{bmatrix}$$

Efetuada as contas temos:

$$[M] = [\text{Trans}][\text{Rot } y(-90+\theta)][\text{Rot } x(90-\phi)]$$

do objeto para sua nova posição:

O produto dessas três transformações deve levar o sistema

$$[\text{Rot } x(90-\phi)] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -\text{sen}(90-\phi) & \text{cos}(90-\phi) & 0 \\ 0 & \text{cos}(90-\phi) & \text{sen}(90-\phi) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

interpretação, temos:

Com essa rotação, o eixo z na sua nova posição, passa pela origem em sua posição original. Novamente recorrendo à segunda

111) Rotação do sistema em torno de seu eixo x de $(-90+\phi)$ graus

$$[\text{Rot } y(-90+\theta)] = \begin{bmatrix} \text{cos}(\theta-90) & 0 & 0 & 0 \\ \text{sen}(\theta-90) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen}(\theta-90) & 0 & \text{cos}(90-\theta) & 0 \end{bmatrix}$$

matriz de transformação é:

Fazer uma rotação de $(-90+\theta)$ em torno do eixo y (novo). A equívales. Fazer uma rotação do eixo de $(90-\theta)$ é o mesmo que eixo y original. Novamente, temos duas interpretações O objetivo é fazer com que o novo eixo z intercepte o

11) Rotação do sistema em torno de seu eixo y de $(90-\theta)$ graus

8 Na verdade, ele mapeia um paralelepípedo com arestas alinhadas com os eixos coordenados do sistema do objeto em sua representação isométrica.

7 A reta AB é dada por $\lambda(1,1,1)$, onde λ é qualquer real.

Substituindo os valores encontrados para θ e ϕ em (3.4) temos a matriz de transformação que mapeia um objeto em sua representação isométrica (8) :

A distância D entre o observador e a origem do sistema do objeto é irrelevante já que iremos fazer uma projecção cilíndrica ortogonal. Não nos interessa também a distância d pelo mesmo motivo.

$(1,1,1) \cdot (0,0,1) = 1 = \sqrt{3} \cdot \cos \theta \Rightarrow \theta = 54.73$ graus (primeiro quadrante).

Vamos obter a perspectiva isométrica de uma outra maneira, através da matriz do observador. Nesse caso, devemos determinar qual é a sua posição em coordenadas cilíndricas. Como a diagonal AB de um cubo imaginário (fig. 2.1) deve ser mapeada para um ponto na isométrica, o observador deve estar sobre a reta AB (lembre-se que o eixo Oz aponta para a origem do sistema do objeto). Em termos vetoriais, podemos dizer que ele deve estar numa direcção dada por $(1,1,1)$. Dado esse vector, podemos determinar agora os ângulos θ e ϕ . Naturalmente, $\theta = 45$ graus. ϕ é o ângulo formado entre $(1,1,1)$ e o eixo Oz. Ele pode ser facilmente calculado usando-se o produto escalar usual:

Exemplo:

Com a relação acima, é possível, então, determinarmos as coordenadas dos pontos de um sólido em função do sistema relativo ao do observador. Para determinarmos agora as coordenadas dos pontos da projecção do sólido, basta, como já vimos, eliminar a coordenada z.

$$[P]^{obv} = [P]^{obj} \cdot [M] \quad (3.5)$$

ou seja, sendo um ponto $[P]^{obj}$ referenciado ao sistema do objeto, este se relaciona com sua representação no sistema do observador através de:

A representação homogênea de um ponto pertencente a um espaço n -dimensional possui $(n+1)$ coordenadas. As coordenadas do espaço n -dimensional são chamadas de coordenadas ordinárias, enquanto aquelas do espaço $(n+1)$ -dimensional, recebem o nome de coordenadas homogêneas. Para fixarmos nossas idéias, vamos imaginar o espaço ordinário bidimensional. A correspondência entre um ponto $P(x,y)$ e sua representação homogênea se faz de modo não único. De fato, a representação homogênea de $P(x,y)$ tem a forma $\lambda(x,y,1)$, $\lambda \in \mathbb{R}$. Caso exista o mapeamento entre um

Faremos uma exposição extremamente breve do que são coordenadas homogêneas. Para o leitor interessado, recomendamos a leitura de [RIE].

3.2.4.1. Coordenadas homogêneas.

Este, provavelmente é o item mais difícil deste capítulo. Aqui trabalharemos explicitamente com o espaço tetradimensional, fazendo uso intensivo da representação de um ponto em coordenadas homogêneas.

3.2.4. Obtenção de perspectivas cônicas

Note a diferença entre as interpretações entre as abordagens da mudança da posição do observador. Enquanto na primeira abordagem os pontos transformados são referenciados ao sistema do objeto, na segunda eles são referenciados a um novo sistema, o sistema do observador.

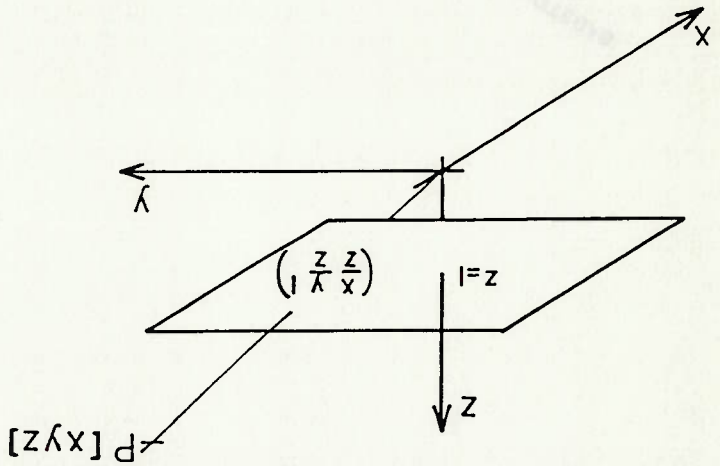
$$[T^{-1}S^{-1}] = \begin{bmatrix} 0.707 & 0 & 0 & 0 \\ -0.707 & 0 & 0 & 0 \\ -0.707 & -0.408 & 0 & -1.732 \\ 0 & 0.577 & 0 & 0 \end{bmatrix}$$

Pode ser que não exista esse mapeamento. Por exemplo, $(0,0,0)$ não corresponde a nenhum ponto no $\mathbb{R}D$.

Para um espaço ordinário tridimensional, valem as mesmas conclusões, apenas que agora, não se tratam mais de planos e sim de hiper-planos.

Interpretado geométrico para a correspondência entre coordenadas homogêneas e ordinárias.

FIGURA 3.14



O mapeamento entre um ponto e a sua representação homogênea tem uma interpretação geométrica interessante. Podemos imaginar que o espaço bidimensional ocorre em uma pequena porção do espaço tridimensional homogêneo, mais precisamente, sobre o plano $z=1$. A correspondência entre espaços seria feita através de uma perspectiva cônica onde o quadro é o plano $z=1$ e o centro de projeção, a origem do espaço tridimensional (Fig. 3.14).

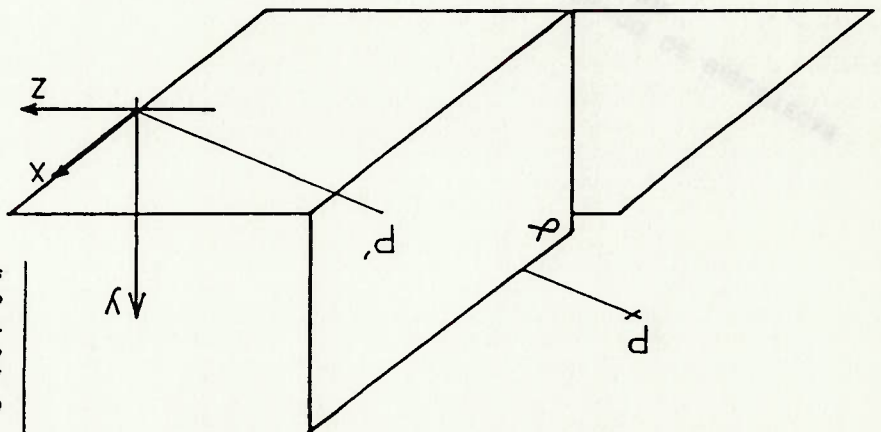
Um ponto $(x, y, 1)$ e um (x, y) , esse mapeamento recebe o nome de projeção.

3.2.4.2. Obtenção de uma transformação para a perspectiva cônica

A figura 3.15 mostra a situação encontrada para a obtenção de uma perspectiva cônica. Em geral o observador é colocado sobre a origem $(0,0,0,1)$ do sistema tridimensional ordinário (o que é consistente com o que acabamos de ver no item anterior), e o quadro de projeção é colocado paralelamente ao plano xy , distando d da origem. Para obtermos a perspectiva cônica do objeto, devemos determinar a intersecção dos raios visuais provenientes de cada ponto do objeto com o quadro de projeção.

FIGURA 3.15

Obtenção de uma perspectiva cônica no quadro a partir de um observador colocado sobre a origem do sistema ordinário.



Determinemos agora quais são as coordenadas da projeção no quadro (pensemos num plano $\alpha // xy$) de um ponto $P(x, y, z, 1)$. Naturalmente, o raio visual que parte da origem e passa pelo ponto em questão tem equação $P(t) = (0,0,0,1) + \lambda \cdot (x, y, z, 0)$. Sua intersecção com o quadro (plano $z=d, w=1$) é dada por :

$$(3.6) \quad \begin{cases} \lambda = d/z \\ x = x \cdot \lambda \\ y = y \cdot \lambda \end{cases}$$

As relações acima mostram um comportamento compatível com nossa experiência cotidiana. Perceba que quanto maior a

distância z^o do ponto a ser projetado ao observador, menores são as dimensões da projeção do objeto, e quanto maior a distância do quadro ao observador, maior o tamanho da perspectiva. O segundo efeito é utilizado em lunetas; veja que quanto maior o seu tamanho, maior a distância entre quadro e centro de projeção, e portanto maior o seu grau de aumento.

Um fato importante deve ser notado: a coordenada z^o está no numerador. Essa é a característica das perspectivas cônicas: Divisão é inevitável. Isso nos permite concluir que a transformação que leva um objeto à sua perspectiva cônica não pode ser linear, e daí, não seria possível encontrar uma matriz de transformação. Entretanto, podemos contornar essa dificuldade se usarmos a relação de correspondência entre coordenadas homogêneas e coordenadas ordinárias. Engenhosamente, podemos deixar a operação de divisão para ser efetuada fora da transformação linear, quando a relação de equivalência entre sistemas de representação for estabelecida.

Geometricamente, o que faremos é determinar uma matriz de transformação [T] que leva o sólido tridimensional (ele existe no hiper-plano $w=1$) a um outro tetradimensional⁽¹⁰⁾, que quando projetado no hiper-plano $w=1$, segundo o observador $(0,0,0,0)^{(11)}$, resulte numa perspectiva cônica onde as relações da (Eq. 3.6) sejam satisfeitas. Esquemáticamente:

¹⁰ Existem infinitos sólidos tetradimensionais candidatos. Tetradimensional no sentido que precisamos além das três coordenadas homogêneas, uma quarta w que varia de ponto para ponto (w não é constante igual a um).

¹¹ Origem do sistema tetradimensional. Ele não possui correspondente no espaço ordinário.

Lembre-se que como tanto uma transformação linear quanto uma projeção cônica mapeiam retas em retas e planos em planos, temos que uma face plana de \mathcal{S} continua plana em \mathcal{S}' ;

$$(3.8) \quad [T] = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & M & 1 \\ 0 & 0 & N & 0 \end{bmatrix} \quad \begin{matrix} M = \frac{(z_{\max} - z_{\min})}{z_{\max}} \\ N = \frac{(z_{\max} - z_{\min})}{-z_{\max} \cdot z_{\min}} \end{matrix}$$

A transformação [T] completa que faz essa transformação é:

Dessa maneira, qualquer ponto [P]=[x y z 1] após ter sido levado à [P][T]=[Q], e após a correspondência entre [Q] e sua representação tridimensional for restabelecida (dividindo as três primeiras coordenadas do ponto pela quarta), as relações de (3.6) serão naturalmente satisfeitas. Note que sobre a terceira coluna de [T] não incide nenhum requisito. Talvez o mais fácil seria fazê-la nula, mas temos uma alternativa melhor. Vamos fazer com que a maior coordenada z seja mapeada para um (1) e a menor para zero (0). Fazendo isso, teremos obtido uma propriedade importante, e interessante, que liga o sólido \mathcal{S} ao sólido \mathcal{S}' : A figura resultante da perspectiva cônica do sólido \mathcal{S} é igual à figura resultante da projeção cilíndrica ortogonal de \mathcal{S} sobre o plano $z=d, w=1$ (fig 3.16).

$$(3.7) \quad [T] = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & ? & 1 \\ 0 & 0 & ? & 0 \end{bmatrix}$$

intuitivamente na forma:

A transformação [T] é linear e pode ser escrita

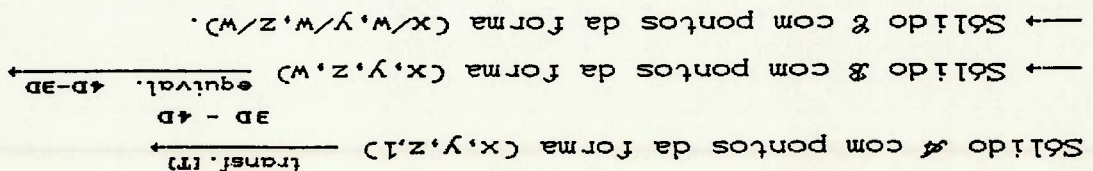
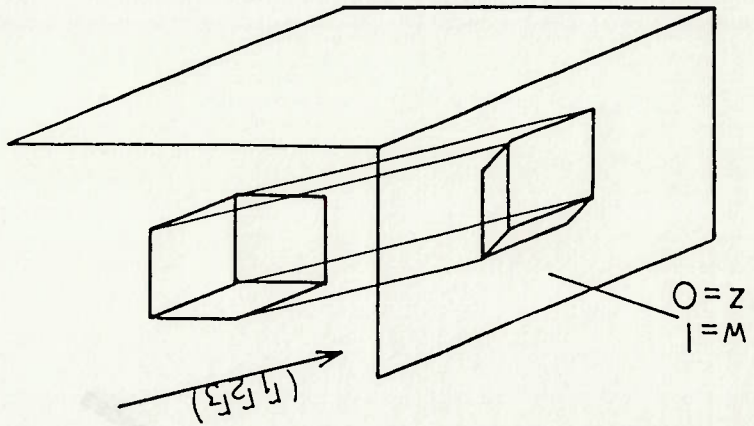


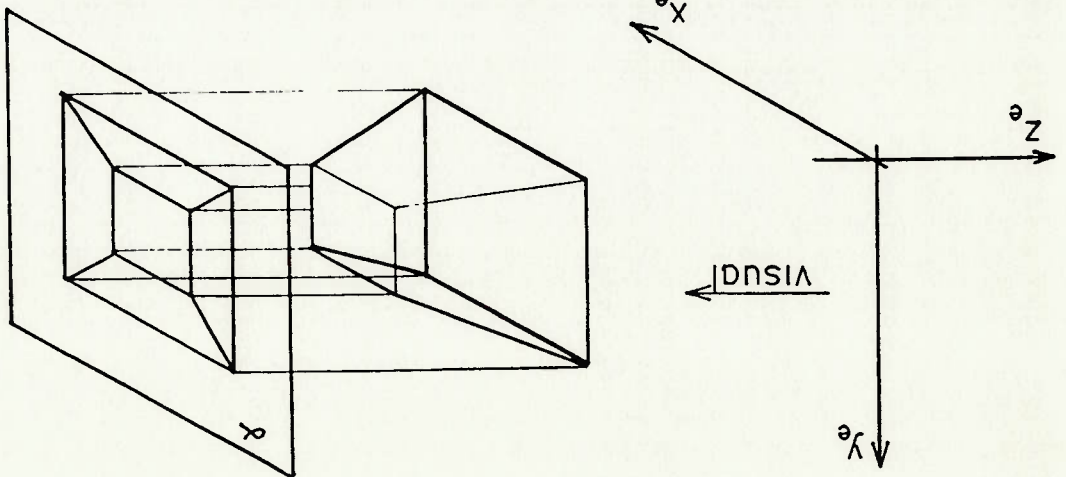
FIGURA 3.17
A direção dos raios
visuais é dada por
 (r_1, r_2, r_3)



Na perspectiva cavaleira, os raios visuais são paralelos entre si, mas chegam ao quadro de modo oblíquo. O quadro será adotado como sendo o plano $(z=0, w=1)$. Os raios visuais terão direção dada pelo vetor $\vec{r}=(r_1, r_2, r_3, 0)$ (fig. 3.17), e o único sistema será o do objeto. Para a obtenção da perspectiva cavaleira de um ponto $P(x, y, z, 1)$, devemos calcular a intersecção entre o raio visual que passa por P e tem direção \vec{r} com o quadro $(z=0, w=1)$.

3.2.5. Obtenção de perspectivas cavaleiras.

FIGURA 3.16
A projeção cilíndrica
ortogonal do sólido
é igual à projeção
cônica de um cubo.



Notemos em primeiro lugar que r^E não deve ser nulo. Caso isso ocorra, os raios visuais ficarão paralelos ao quadro, e não haverá intersecção. Adotemos arbitrariamente $r^E = 1$ (isso não restringe em nada a obtenção de qualquer perspectiva cava-

apesar de correta, a expressão acima usa termos estranhos ao desenho técnico. Lá, uma perspectiva cavaleira não é especificada pela direção do observador r^E , mas sim pelo ângulo das fugantes α e pelo fator de deformação k . Por isso, passamos agora a reescrever (3.12) em função de α e de k .

$$[CV] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -r^E/r^E & 0 & 0 & 0 \\ -r^E/r^E & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Em termos matriciais, ela assumiria a forma:

$$CV(x, y, z, w) = \left[x - (r^E/r^E) \cdot z, y - (r^E/r^E) \cdot z, 0, w \right] \quad (3.11)$$

Essa transformação poderia ser colocada facilmente em termos de uma função $CV: \mathbb{R}^4 \rightarrow \mathbb{R}^4$ da seguinte maneira:

$$\begin{aligned} x^E &= x - (z/r^E) \cdot r^E \\ y^E &= y - (z/r^E) \cdot r^E \\ z &= 0 \end{aligned} \quad (3.10)$$

Daí temos imediatamente as coordenadas do ponto de intersecção (x^E, y^E, z^E) :

$$\lambda = -z/r^E \quad (3.9)$$

obtida fazendo-se $z + \lambda \cdot r^E = 0$. Isso implica: da reta raso visual. A intersecção com o plano ($z=0, w=1$) é

Seja $R(\lambda) = (x^E, y^E, z^E, 1) + \lambda(r^E, r^E, r^E, 0)$ a equação paramétrica

Matematicamente:

mais natural.

Escolhemos a segunda linha de (3.14) pelo fato de ser

$$\begin{matrix} 1 \rightarrow (1,0,1) \\ 2 \rightarrow (1,1,1) \\ 3 \rightarrow (0,1,0) \\ 4 \rightarrow (0,0,0) \\ 5 \rightarrow (1,1,0) \\ 6 \rightarrow (1,0,0) \end{matrix} \xrightarrow{\text{matriz dos pontos } [P]} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Vejam quais são as coordenadas da perspectiva cavaleira de um prisma de coordenadas (fig. 3.15) com $\alpha=30$ graus e $k=1$:

Exemplo.

$$[CV] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -k \cdot \cos \alpha & 1 & 0 & 0 \\ 0 & -k \cdot \text{sen} \alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

técnico :

Temos finalmente a matriz [CV]⁽¹²⁾ compatibilizada com desenho

$$\begin{aligned} r_1 &= +k \cdot \cos \alpha \quad \text{e} \quad r_2 = +k \cdot \text{sen} \alpha \\ \text{ou} \\ r_1 &= -k \cdot \cos \alpha \quad \text{e} \quad r_2 = -k \cdot \text{sen} \alpha \end{aligned} \quad (3.14)$$

Notando agora que essas equações são satisfeitas se

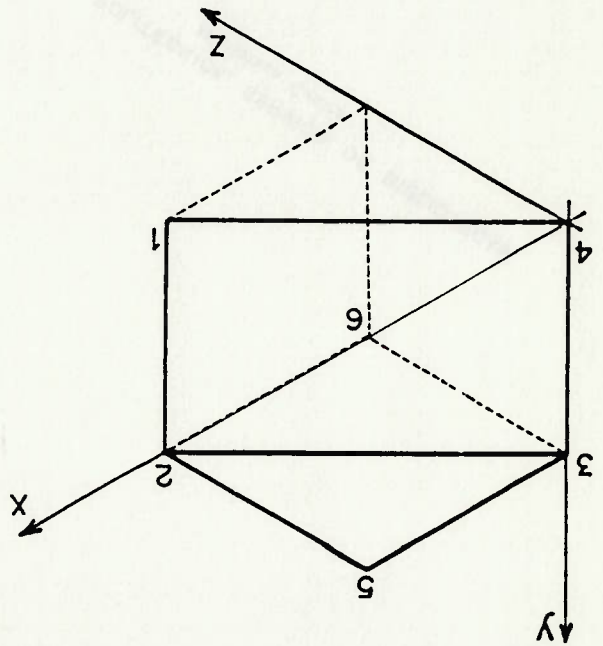
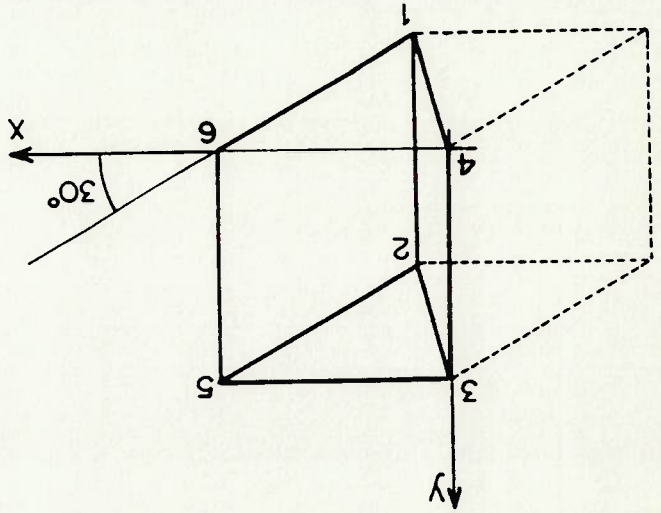
Assim, deveremos ter $\text{tg} \alpha = (r_2/r_1)$ e $(r_1^2 + r_2^2) = k^2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -r_1 & -r_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [-r_1 \quad -r_2 \quad 0 \quad 0] \quad (3.13)$$

letra). Vamos, usando (3.12), projetar o vetor (0,0,1), que deverá ser transformado em um outro vetor de comprimento k e que faz um ângulo α com a horizontal na perspectiva :

Perspectiva Cavalreira.
Posição inicial e final.

Figura 3.18



Acima são mostradas as coordenadas dos vértices da figura formada na perspectiva cavalreira.

$$[P][CV] = [P] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -0.866 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0.134 & 0.5 & 0 & 0 & 1 & 0 \\ 0.134 & -0.5 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Usando agora a matriz [CV] com $\alpha=30^\circ$ e $k=1$ sobre [P] temos:

Do ponto de vista do pré e pós processamento de dados para o método dos elementos finitos, esta é a principal atividade, já que é através dela que a maior parte dos erros na malha de elementos finitos é detectada. Contudo, ela não é suficiente, porque numa malha em geral existem linhas em demasia, causando um emaranhado de linhas que impossibilita mostrar, de maneira clara, a numeração de nós, elementos, ou o mapeamento de curvas de iso-tensão. No próximo capítulo, veremos como podemos aliviar o desenho da malha do excesso de linhas, eliminando as linhas ocultas que, num objeto, são invisíveis ao observador.

Vimos nesse capítulo como podemos obter uma série de perspectivas usando matrizes de transformação. Neste trabalho, operamos apenas com transformações de pontos, retas e planos.

4. REMOÇÃO DE LINHAS INVISÍVEIS

4.0. Introdução

Remover linhas invisíveis significa retirar da imagem final do computador aquelas arestas, ou contornos, de um objeto que não são visíveis por estarem por trás de algum material opaco. Essas linhas são correspondentes às linhas tracejadas do desenho técnico.

Do ponto de vista das fases de pré e pós processamento de dados para o método dos elementos finitos, esse assunto é importante, uma vez que é grande o número de linhas, arestas de elementos ou barras, se cruzando no emaranhado da malha. Poderíamos afirmar que sem os algoritmos para remoção de linhas ocultas (em muitos casos diríamos "graficamente excedentes"), seria impossível a visualização do mapeamento de tensões, da numeração de nós e elementos, enfim, de qualquer tratamento gráfico posterior que dependesse de um substrato não polido. Tudo isso sem mencionarmos que do ponto de vista da feitura de relatórios finais, um desenho cujas linhas ocultas foram removidas, apresenta um acabamento superior.

Dois características do método dos elementos finitos devem ser levadas em conta na problemática de remoção de linhas invisíveis. A primeira é o fato de que a malha, em geral, é extremamente discretizada, isto é, apresenta um grande número de pequenas superfícies interconectadas. A segunda característica é o fato da malha apresentar diferentes tipos de elementos, e portanto, de diferentes topologias, unindo num mesmo domínio, segmentos de reta isolados (vigas e treliças) com faces planas

Pixel (picture element) é o menor elemento gráfico acessível de um monitor de vídeo. Quanto maior sua quantidade num monitor, maior é a sua resolução.

Basicamente, podemos classificar os algoritmos em três categorias conforme o espaço sobre o qual operam: Naquelas que trabalham no espaço do objeto, no espaço da imagem e mistos (Tila de prioridades)[SHU]. A primeira categoria se refere aos algoritmos que, através de relações geométricas, determinam quais entes (objetos, arestas, pontos, etc) devem ser visíveis segundo um dado observador. A segunda se refere aos algoritmos que determinam quais deverão ser os pixels⁽¹⁾ acessos ou apagados, ou até as cores ou intensidades dos pixels da imagem acabada. Note que o esforço computacional aumenta com a complexidade da forma dos objetos no primeiro caso, enquanto no segundo caso ele aumenta com a complexidade da cena final. Os mistos empregam técnicas derivadas das duas categorias.

Durante muito tempo o problema foi considerado como sendo um dos mais desafiantes na área da computação gráfica [NEW], devido à complexidade de se resolver um problema que envolve muitas operações numa máquina com poucos recursos como as de hoje. Hoje temos à disposição uma série de algoritmos específicos para cada tipo de aplicação, existindo inclusive alguns já implementados no nível de hardware.

Essas duas características implicam em certas adaptações em algoritmos clássicos, simplificando-os e tornando-os mais robustos, no sentido de se tornarem capazes de lidar com topologias diversas.

Trata-se de um algoritmo que trabalha no espaço da imagem e é extremamente simples [SHU,NEW,ROG], correspondendo ao que poderia ser chamado de uma solução intuitiva do problema. Para cada pixel da imagem, são armazenadas a profundidade e a intensidade do ponto mais próximo da tela que se enquadra no pixel e é pertencente à algum objeto. Em outras palavras, dentre todos os pontos dos objetos cujas projeções caem sobre o pixel P, guarda-se na memória a profundidade e a intensidade daquele que se situar mais próximo do observador.

Em termos algorítmicos temos:

1. Para cada pixel P de posição (x,y) da imagem, inicie $prof(x,y)=infinito$ e $intens(x,y)=cor$ do fundo.
2. Para cada polígono da cena, determine os pontos (x,y,z). Para cada um desses pontos faça:
 - 2.1 Determine a profundidade z de (x,y,z)
 - 2.2 Se $z < prof(x,y)$ então faça:
 - 2.2.1 $intens(x,y)=intensidade$ do polígono
 - 2.2.2 $prof(x,y) = z$
3. Desenhe o pixel de posição (x,y) com a intensidade $intens(x,y)$.
4. Fim

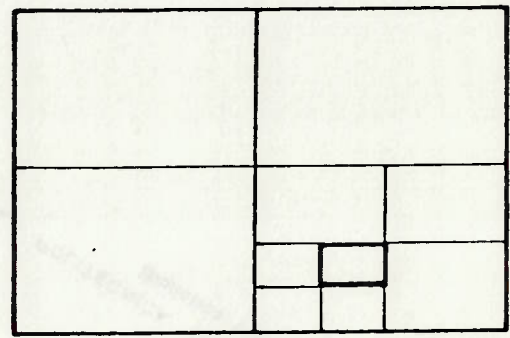
Fica bem claro que qualquer cena pode ser resolvida com extrema simplicidade, mesmo aquelas envolvendo superfícies curvas e intersecções. Entretanto, a aplicabilidade do algoritmo fica limitada à capacidade do hardware à disposição, tanto no que se refere à quantidade de memória, quanto à velocidade de processamento. Como ilustração, pode-se tomar uma cena contida num vídeo com resolução de 640×200 pixels. A memória requerida para armazenar as profundidades e intensidades é de:

intensidade (acesso e apagado) = 750 kbytes 1111
 profundidade (real requer 6 bytes)

Poderíamos afirmar que o algoritmo que acabamos de ver pertence à classe dos algoritmos de força bruta. Veremos agora, alguns outros que aplicam técnicas mais inteligentes.

4.2. Algoritmo de Warnock

A idéia básica do algoritmo de Warnock [SHU,NEW,ROG] é muito elegante, aproximando-se do modo como o cérebro humano processa uma imagem complexa. Quando observamos uma cena, inicialmente somos capazes de apenas captar as formas e cores gerais. Somente quando nossa atenção se volta para uma pequena parte do todo é que os detalhes são resolvidos.



Divisão em janelas

O algoritmo de Warnock usa esse mesmo princípio. Quando ele se julga incapaz de resolver uma imagem contida numa determinada janela, ele a divide em janelas menores, na suposição de que aí encontrará cenas mais simples. Se mesmo assim ele

Inicia-se o algoritmo com a lista ativa contendo todos os polígonos do modelo geométrico. Seja J uma janela subdividida em quatro sub-janelas J_i , com $i=1,2,3,4$. Da lista ativa $L(J)$ relativa à janela J , construímos as listas ativas $L(J_i)$ associadas aos seus quatro filhos, como se segue:

4.2.2. Construção da lista ativa

- i) Só existe um polígono na lista ativa.
- ii) Só existe uma aresta na lista ativa. Essa aresta poderia pertencer a vários polígonos simultaneamente.
- iii) O tamanho da janela chegou a um valor mínimo pré-especificado.

ocorre:

A capacidade de resolução de uma cena está intimamente associada à uma lista de polígonos ativos. A lista de polígonos ativos é formada por todos os polígonos que potencialmente podem ter alguma aresta a ser desenhada numa determinada janela. O algoritmo de Warnock para o processo de subdivisão, e desenha o conteúdo da lista ativa quando um dos seguintes fatos

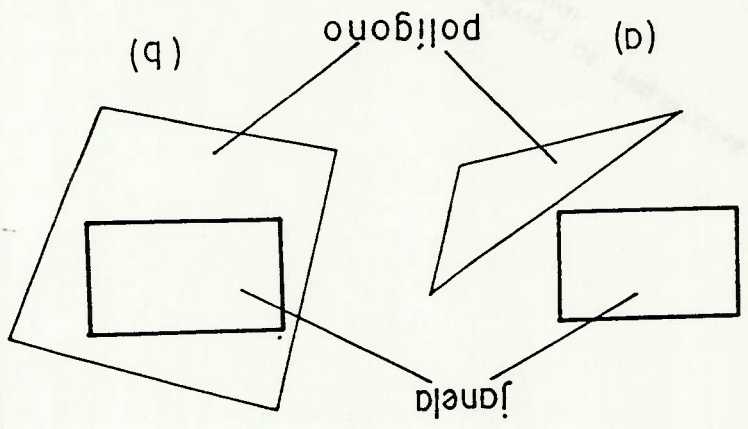
4.2.1. Capacidade de solucionar uma cena.

Claramente, o ponto crucial do algoritmo é o teste de verificação da capacidade de se resolver uma cena contida em uma janela, e caso ele seja capaz, qual a atitude a ser tomada.

ainda se julgar incapaz de resolver os sub-problemas, ele aplica recursivamente o algoritmo até que uma condição de parada seja satisfeita. No algoritmo originalmente proposto por Warnock, as janelas são retangulares, e a cada nível de resolução, ela é subdividida em novas quatro sub-janelas iguais.

- 1) Faça inicialmente $L(J) = L(J)$.
- 2) Remova de $L(J)$ todos os polígonos disjuntos da janela J (fig. 4.1)
- 3) Remova da lista todos os polígonos P que encobrem totalmente a janela J (fig. 4.1). Eles não devem estar na lista porque suas arestas não são visíveis na janela.
- 4) Remova da lista todos os polígonos que estão atrás de um polígono que encobre totalmente a janela.

FIGURA 4.1
 Polígonos que estão disjuntos da janela (a) e que a encobrem totalmente (b) devem ser retirados da lista ativa.



O conceito de "estar atrás" deve ser melhor explicado. A verificação de que a face Q está atrás de uma outra face P se faz através da comparação de sinais. Suponha que façamos a substituição das coordenadas do observador na equação do plano que passa por P ; digamos que o sinal do valor obtido seja positivo. Se ao substituirmos as coordenadas de todos os vértices de Q na equação de P , obtivermos somente valores negativos,

2) Polígonos disjuntos são aqueles que não possuem nenhum ponto em comum, levando-se em conta suas respectivas regiões internas (fig. 4.1 a)

3) Equação do plano que passa por P

EXERCÍCIOS DE LÓGICA DE PROGRAMAÇÃO

Os passos do algoritmo acima devem ser seguidos toda vez que uma janela for subdividida. A condição de parada já foi mostrada no item anterior.

então a face Q está atrás de P.

Roberts desenvolveu o primeiro algoritmo para a eliminação de linhas invisíveis no domínio do objeto conhecido [ROG, NEW, SCHU].

Embora a ideia básica seja simples, a maneira como ela é implementada é muito elegante, envolvendo inclusive programação linear (No caso, método simplex). Nela, cada aresta relevante é testada contra cada sólido para se verificar a ocorrência de trechos encobertos. Este teste é executado traçando-se um raio visual entre o observador e um ponto G genérico da aresta. Caso exista intersecção entre esse raio visual e o sólido, tira-se a conclusão que G não pode ser visível, já que a informação luminosa enviada ao observador é bloqueada. A partir dos pontos de intersecção entre o raio e o sólido, determina-se as regiões invisíveis da aresta.

O algoritmo, originalmente proposto por Roberts, restringe-se ao tratamento de domínios contendo apenas objetos convexos. Neste trabalho derrubaremos esta restrição e mostraremos uma versão do algoritmo útil no processamento de imagens para elementos finitos.

4.3.1 Descrição dos objetos pertencentes a um domínio

Passamos a descrever os entes envolvidos na remoção de linhas invisíveis de um sólido usando-se o algoritmo de Roberts, como ele apresentou originalmente.

Todos os objetos considerados devem ser convexos e formados por faces planas. Objetos que não obedecem tal regra são quebrados em outros mais simples. Cada volume convexo é então descrito por uma coleção de planos que passam pelas faces. Desta maneira, um cubo seria descrito por seis planos de fronteira ("bounding planes").

Quando substituímos as coordenadas de um ponto qualquer do espaço na equação de um plano, o sinal resultante pode ser nulo (se ele for pertencente ao plano), positivo ou negativo. Ajusta-se os coeficientes dos planos de fronteira de modo que o sinal correspondente a um ponto no interior do sólido seja positivo.

Mais especificamente, lembremos que um plano pode ser representado por:

$a \cdot x + b \cdot y + c \cdot z + d = 0$. Como um ponto em coordenadas homogêneas pode ser escrito como $[x \ y \ z \ 1]$, temos, a equação do plano em forma matricial:

$$a \cdot x + b \cdot y + c \cdot z + d = [x \ y \ z \ 1] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = 0. \quad (4.11)$$

O interior de um volume convexo de n faces poderia ser descrito como o conjunto de todos os pontos $[1] = [x \ y \ z \ 1]$, tal que :

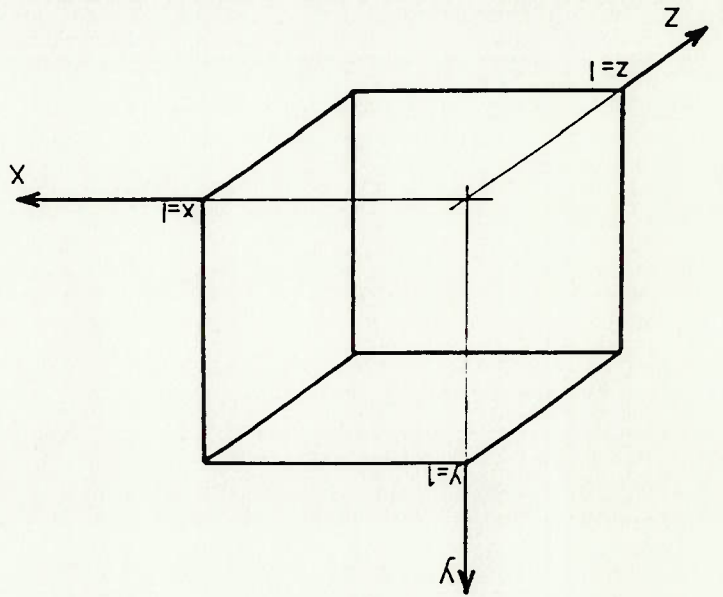


FIGURA 4.2
Cubo de lado unitário.

Vejamos agora como um cubo pode ser modelado, em termos dos planos fronteira, resultando na matriz volumétrica. Seja o cubo de lado unitário da figura 4.2. As equações dos planos que contêm as faces e a matriz volumétrica são mostradas abaixo:

Exemplo:

$$matriz \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} > 0.$$

como:

Assim o interior de um sólido convexo poderia ser descrito

em valor algébrico. Por exemplo, $matriz \begin{bmatrix} -5 & -3 & -1 & -2 \end{bmatrix} = -1$. $matriz[V]$ retorna o maior componente de uma matriz linha

operador $matriz$:

Apenas para facilitar nossa nomenclatura, vamos definir o

$$[V] = \begin{bmatrix} a_1 & b_1 & a_1 \\ a_2 & b_2 & a_2 \\ \dots & \dots & \dots \\ a_n & b_n & a_n \end{bmatrix}$$

é chamada de matriz volumétrica

e onde

(4.2)

$$[I], [V] = [p \ q \ r \ s] \text{ com } \begin{cases} p > 0 \\ q > 0 \\ r > 0 \\ s > 0 \end{cases}$$

4.3.2 Determinação das arestas escondidas

Agora que temos as descrições dos elementos que participam do algoritmo, podemos afirmar que o trecho de um segmento $P_i P_j$ que é obscurecido por um volume modelado pela matriz volumétrica $[V]$ é o conjunto dos pontos $[P(\lambda)]$ tal que o raio visual associado $[R(\alpha, \lambda)]$ intercepta o volume. Matematicamente falando, o conjunto dos pontos de $P_i P_j$ obscurecidos pelo volume $[V]$ constitui-se dos pontos $P(\lambda)$ tal que

$$matrix [R(\alpha, \lambda)] [V] > 0 \quad (4.5)$$

Assim, num sólido convexo que possui intersecção com um raio visual $[R(\alpha, \lambda)]$ (fig.4.3), os pontos a partir de onde o segmento começa a ficar invisível, e visível novamente, são pontos de mínimo e de máximo de $\lambda^{(4)}$ que obedecem a restrição: $matrix [R(\alpha, \lambda)] [V] > 0$. O problema, desta maneira estabelecido, pode ser resolvido pelo método simplex [EHR], como veremos a seguir.

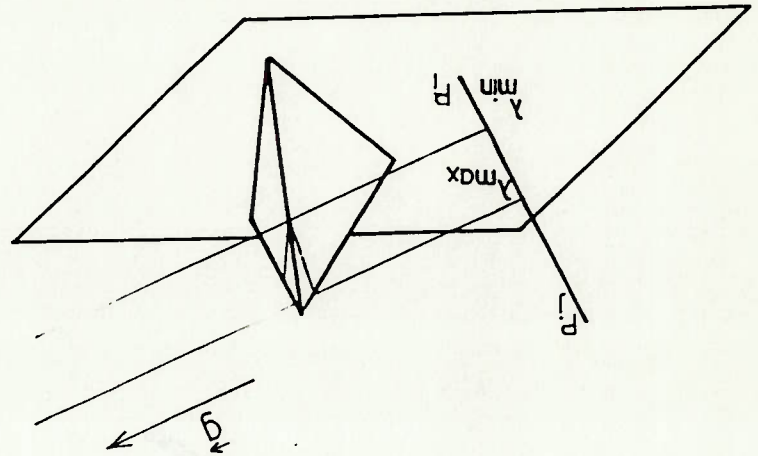


FIGURA 4.3
Condições limites, de máximo e de mínimo para a visibilidade de um segmento.

⁴ Para algum α .

Exemplo:

Vamos resolver o problema de se determinar a porção de um segmento P1P2 que é encoberta por um cubo C. Dados:

- Posição do observador impróprio dado por $(g) = (1 \ 1 \ 0)$
- $(P1) = (-2 \ 0 \ 1)$ $(P2) = (2 \ -1 \ 1)$
- Coordenadas dos vértices do cubo (fig. 4.2):

1	(0	0	1)
5	(0	1	0)
2	(0	0	1)
6	(0	1	1)
3	(1	0	1)
7	(1	1	1)
4	(1	0	0)
8	(1	1	0)

começamos a resolver o exemplo:

a) Construção da matriz volumétrica:
 A matriz volumétrica desse cubo já foi calculada no exemplo anterior e é dada por:

$$[V] = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

b) Equação dos raios visuais que partem de um ponto de P1P2 e partem na direção do observador.

$$[R(\alpha, \lambda)] = [(P1) + \alpha \cdot (P2) - (P1)] \cdot \lambda + \alpha \cdot (g)$$

c) Montagem do problema simplex:

Os pontos do segmento P1P2 que são obscurecidos pelo volume são aqueles que para algum $\alpha \in [0, 1]$ tem-se:
 $max_{\alpha, \lambda} [R(\alpha, \lambda)](V) > 0$.
 Explicitamente:

$$\begin{aligned} \lambda + \alpha &> 0 \\ 3 - 4\lambda - \alpha &> 0 \\ 1 - 2\lambda + \alpha &> 0 \\ 1 - \lambda - \alpha &> 0 \\ -2 + 4\lambda + \alpha &> 0 \\ 2\lambda - \alpha &> 0 \end{aligned}$$

Além de:

$$\begin{aligned} \alpha &> 0 \\ 1 - \lambda &> 0 \\ \lambda &> 0 \end{aligned}$$

Como o eólido é convexo, basta calcularmos os pontos a partir de onde o segmento começa a ficar invisível, e outro, a partir de onde o segmento se torna novamente visível. Isto é, devemos resolver dois problemas simplex. Um maximizando λ e

Suponha uma face de um sólido convexo. Calcule a normal a essa superfície de modo que ela aponte para fora do volume (Essa normal terá, então, sentido inverso ao do que é usado no algoritmo de Roberts). Se o ângulo formado entre essa normal e o vetor do raio visual \vec{g} for obtuso (Fig. 4.4), então claramente, a face deve ser oculta do observador pelo próprio volume do sólido. Conseqüentemente, essa face pode ser eliminada do conjunto dos elementos a serem verificados pelo algoritmo.

4.3.3.1. Teste da normal a um plano.

Vamos mostrar algumas dessas técnicas.

Existem técnicas que podem ser utilizadas para diminuir o número de cálculos no algoritmo que acabamos de ver. Deve-se ressaltar, entretanto, que essas técnicas, que são baseadas em critérios geométricos, estão presentes também em quase todos os outros algoritmos utilizados para a remoção de linhas invisíveis [SHU].

4.3.3. Técnicas para acelerar o algoritmo de remoção de linhas invisíveis.

Caso não exista solução viável no método simplex, podemos concluir que todo o segmento está visível.

$$P(0, 0000) = (-2 \ 0 \ 1 \ 1) + 0,3333 \cdot (4 \ 1 \ -2 \ 0) = (10,67 \ 0,67 \ -0,33 \ 1)$$

$$P(0, 3333) = (-2 \ 0 \ 1 \ 1) + 0,3333 \cdot (4 \ 1 \ -2 \ 0) = (10,67 \ 0,67 \ 0,33 \ 1)$$

Assim, o segmento P1P2 está invisível entre

$$\lambda \text{ mínimo} = 0,3333 \quad (\alpha = 0,6666)$$

$$\lambda \text{ máximo} = 0,6666 \quad (\alpha = 0,3333).$$

Resolvendo, chegamos a λ mínimo = 0,3333 ($\alpha = 0,6666$) e λ máximo = 0,6666 ($\alpha = 0,3333$). Resolvendo, chegamos a λ mínimo = 0,3333 ($\alpha = 0,6666$) e λ máximo = 0,6666 ($\alpha = 0,3333$).

Resolvendo, chegamos a λ mínimo = 0,3333 ($\alpha = 0,6666$) e λ máximo = 0,6666 ($\alpha = 0,3333$).

Resolvendo, chegamos a λ mínimo = 0,3333 ($\alpha = 0,6666$) e λ máximo = 0,6666 ($\alpha = 0,3333$).

Resolvendo, chegamos a λ mínimo = 0,3333 ($\alpha = 0,6666$) e λ máximo = 0,6666 ($\alpha = 0,3333$).

- 1) Sejam o segmento $P_i P_j$ e o sólido S de vértices V_1, V_2, \dots, V_n .
 2) Determine qual dos dois pontos P_i ou P_j possui menor coordenada $z = z_{\text{menor}}$.
 3) Dentre todos os vértices V_1, V_2, \dots, V_n determine qual possui

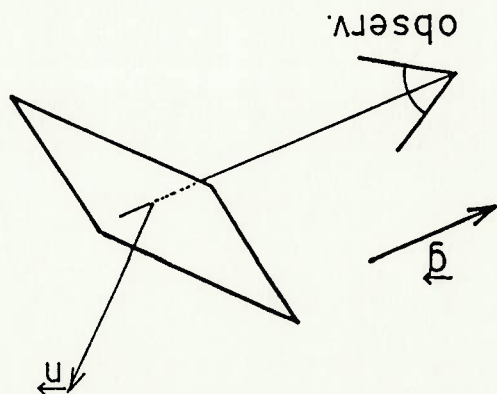
Esse teste fica simplificado nos algoritmos de remoção de linhas invisíveis uma vez que em geral a direção \vec{g} é dada pelo eixo Oz. Nesse caso, bastaria o seguinte algoritmo (Fig. 4.5) :

Sejam um segmento $P_i P_j$ e um sólido S . Projete tanto o segmento como o sólido sobre o vetor \vec{g} . Se conseguirmos provar que não existe superposição entre as projeções, e que um ponto da projeção do segmento se encontra mais próximo do observador do que um ponto da projeção do sólido, então para esse par face-sólido não é necessário aplicar o método simplex para determinar qual a porção do segmento que está escondida. Todo o segmento estará visível.

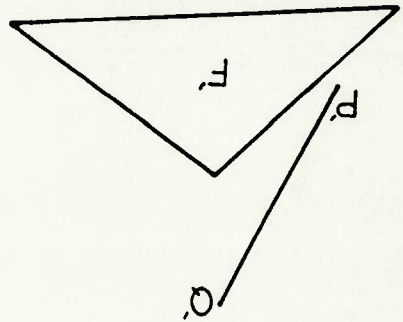
4.3.3.2. Teste da superposição na direção de \vec{g} .

Teste da normal à uma face. Se o produto escalar entre o vetor \vec{g} e a normal que aponta para fora do volume for negativo (como é o caso), então, a face pode ser excluída do algoritmo.

FIGURA 4.4.



A maneira como o teste pode ser efetuado é simples. Para determinarmos se o ângulo é obtuso, basta calcularmos o produto escalar entre \vec{g} e a normal. Caso o resultado seja negativo, o ângulo é obtuso, senão, o ângulo é agudo.



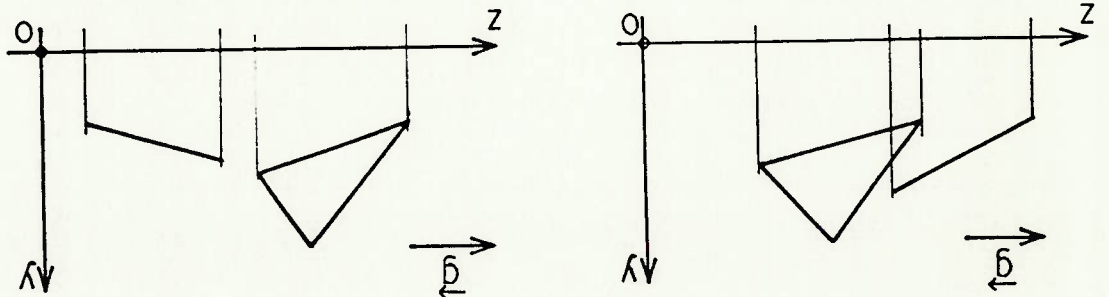
As projecções da sua
perfeite e do segmen-
to não se sobrepõe.

FIGURA 4.6

Este teste visa detectar aqueles casos em que não importa se um sólido está na frente de um segmento ou se o segmento está na frente do sólido. Essa situação de liberdade ocorre quando a perspectiva do sólido não se sobrepõe à perspectiva do segmento (fig. 4.6). Devemos nos lembrar que o teste deve ser necessariamente rápido para ter utilidade (se for lento, seria melhor rodar o simplex diretamente). Numa relação de compromisso entre tempo e precisão, é possível fazermos uma simplificação. Não testamos mais a projeção do segmento contra a projeção exata do sólido, mas contra uma caixa mínima que a envolve (fig. 4.7). Desta maneira, o teste é acelerado, embora nem todas as vezes em que a caixa intercepta a projeção do segmento, necessariamente o simplex deva ser ativado. Note que o teste é "conservativo".

4.3.2.3. Teste mínimo.

FIGURA 4.5
Teste de superposto
na direção Oz.



4) Se $z_{menor} > z_{maior}$ então o segmento está na frente do sólido e o simplex não precisa ser ativado.

maior coordenada z maior.

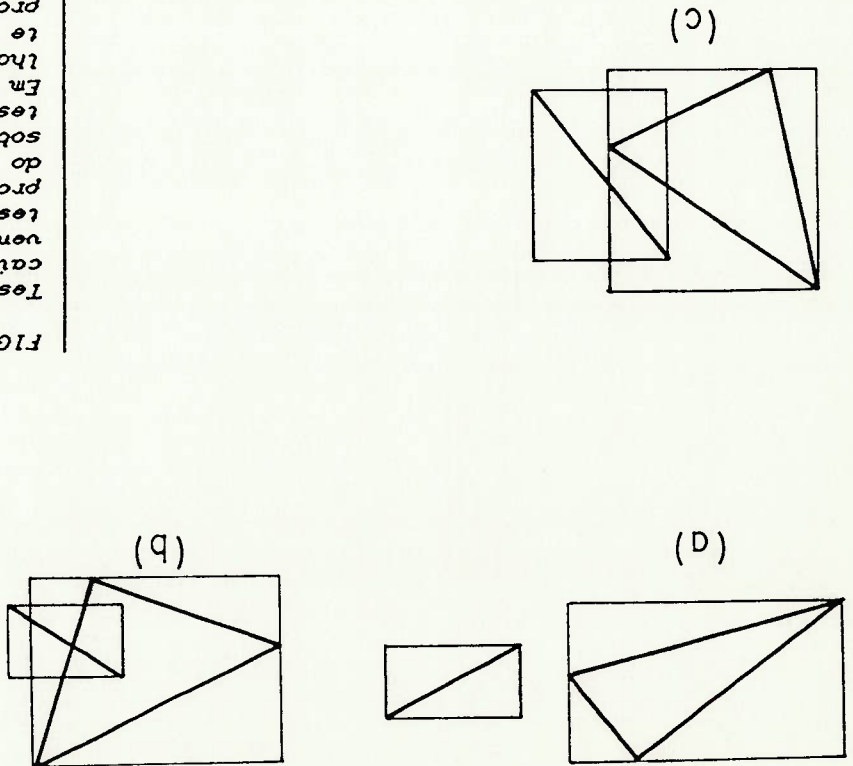
4.3.4. Adaptação do algoritmo de Roberts ao pós processamento gráfico de malhas de elementos finitos.

Num modelo usando o método dos elementos finitos, é muito comum a ocorrência de segmentos e faces soltos (elementos de viga, de treliça, de membrana, de placa, etc), sem conexão com nenhum sólido convexo. Deve ser ressaltado o fato de que as superfícies que ocorrem no método dos elementos finitos sempre são convexas.

Apresentamos agora uma adaptação do algoritmo de Roberts, útil no pós processamento gráfico de malhas de elementos finitos.

Teste minímax usando caixas mínimas envolventes. Em (a), o teste revela que as projeções da face e do segmento não se sobrepõe. Em (b) o teste indica que sim. Em (c) o teste "falha" conservativamente e indica que as projeções se interseccionam.

FIGURA 4.7



Nesta adaptação, consideraremos apenas quadriláteros e triângulos planos como superfícies permitidas. Caso um sólido participe da malha de elementos, este deverá ser modelado geométricamente através de suas faces. Segmentos soltos relativos à elementos de viga e de treliça (no caso da análise estrutural) são permitidos.

A idéia básica é reduzir a dimensão do domínio considerado, passando de 3D para 2D. Para percebermos como isso pode ser feito, vamos considerar uma face F e um segmento PQ (fig. 4.8).

Vamos supor que PQ se localize atrás de F . Neste caso, para determinarmos a região escondida de PQ , basta analisarmos o que ocorre com as projeções sobre o quadro de PQ e F (fig. 4.9).

Assim desta maneira estamos resolvendo um problema originalmente proposto em três dimensões num domínio bi-dimensional. Agora existe apenas um pequeno problema: Como saberemos se PQ realmente está na frente de F , ou pior, se PQ não intersecta F ? Devemos nos lembrar que nenhum dos métodos rápidos do item 4.3.2 é capaz de responder essa questão. Ela só pode ser resolvida através do cálculo de α . A região de PQ que se situa na frente de F é aquela que apresenta $\alpha < 0$ para um determinado λ entre 0 e 1 (fig. 4.10). Começamos então estabelecendo uma relação entre α e λ na intersecção de um raio visual que parte de um ponto de um segmento genérico P_j^1 com direção \vec{g} com um plano π que contém uma face F .

$$[R(\alpha, \lambda)] = [P(\lambda)] + \alpha \cdot [g] = [P_j^1] + \lambda \cdot ([P_j^1] - [P_j^2]) + \alpha \cdot [g]$$

$$0 \leq \lambda \leq 1, \quad 0 \leq \alpha$$

5 Lembre-se que o conjunto dos raios visuais é descrito por (item 4.3.1.3):

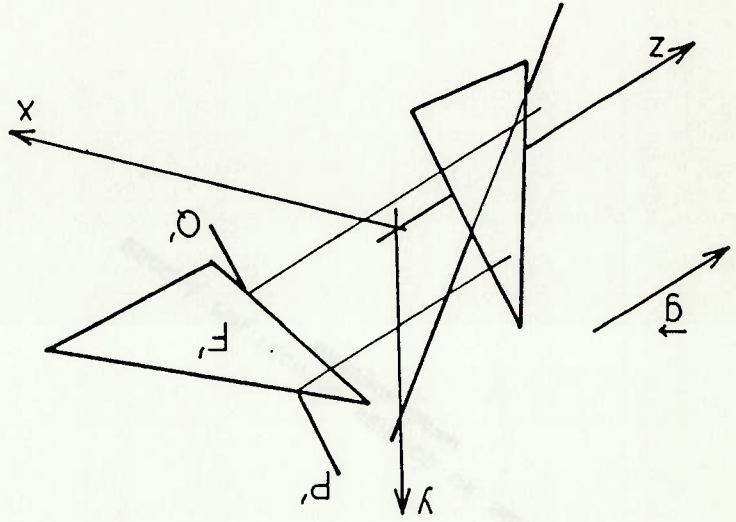


FIGURA 4.9
 A redução a um plano ZD se faz considerando-se apenas as projeções dos elementos no quadro.

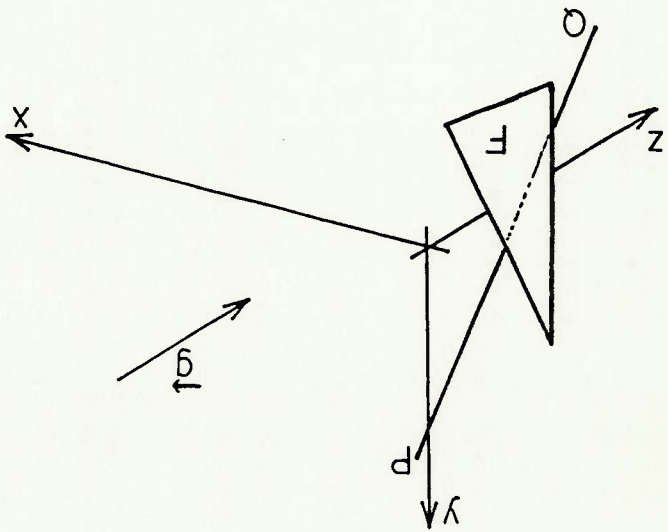


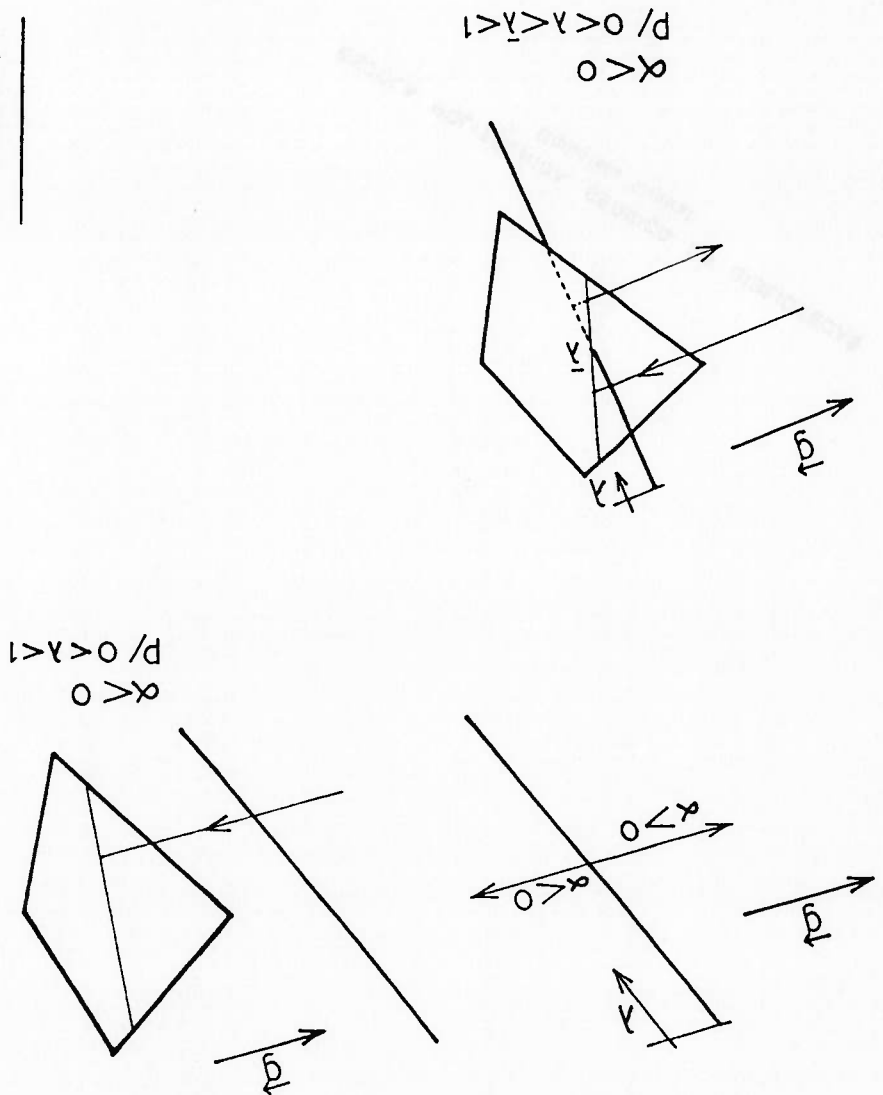
FIGURA 4.8
 A face F e o segmento PQ serão projetados no plano xy.

b) Um ponto do raio visual : $[R(\alpha, \lambda)] = [P_i] + \lambda([P_j] - [P_i]) + \alpha.[g]$

Sejam: a) Um ponto do segmento $P_i P_j$: $[Q(\lambda)] = [P_i] + \lambda([P_j] - [P_i])$

4.3.4.2. Determinação da intersecção de um raio visual que parte de P_i com o plano π que contém a face F_i .

FIGURA 4.10
O trecho encoberto pela superfície α quele em que apresenta em sua extensão $\lambda < 0$ para algum λ entre 0 e 1.



Essas condições serão tiradas a partir das projeções tanto do segmento como da face f sobre o quadro, usando como no algoritmo de Roberts, o método simplex. Note que como passamos a

4.3.4.3. Condições necessárias para que uma região de P_i esteja obscurecida por f quando consideramos suas arestas.

Passamos agora a estabelecer as condições necessárias para que uma região esteja escondida por uma face f considerando-se agora suas arestas.

Vale aqui uma observação. Caso o denominador de (4.6) for nulo, isso significa que a normal à face F é perpendicular à direção de visada $[g]$, ou seja, a face F não aparecerá para o observador (Pensando fisicamente, se a face possuir espessura, o observador verá apenas um "segmento"). Desta maneira, nenhum segmento pode ser obscurecido por ele.

Uma condição necessária para que uma região de P_i seja obscurecida por f é $\alpha > 0$, com $0 \leq \lambda \leq 1$.

$$\alpha = \frac{[g][f]}{(\lambda([P_i] - [P_j]) - [P_i]) [f]} \quad (4.6)$$

Resolvendo para α , temos:

$$[R(\alpha, \lambda)][f] = 0$$

A intersecção entre o raio visual $[R(\alpha, \lambda)]$ e π é dada por:

$$[x \ y \ z \ 1][f] = 0$$

$$[x \ y \ z \ 1] = 0. \text{ Denominamos } [f] = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

c) Equação do plano π que contém f :

trabalhar com projeções sobre o quadro, o problema passa a ser bidimensional! Seguiremos a seguinte notação:

F' é a projeção de F sobre o quadro

(Fig. 4.11)

P_i, P_j' é a projeção de P_i, P_j .

Em geral, o apóstrofe indicará projeção.

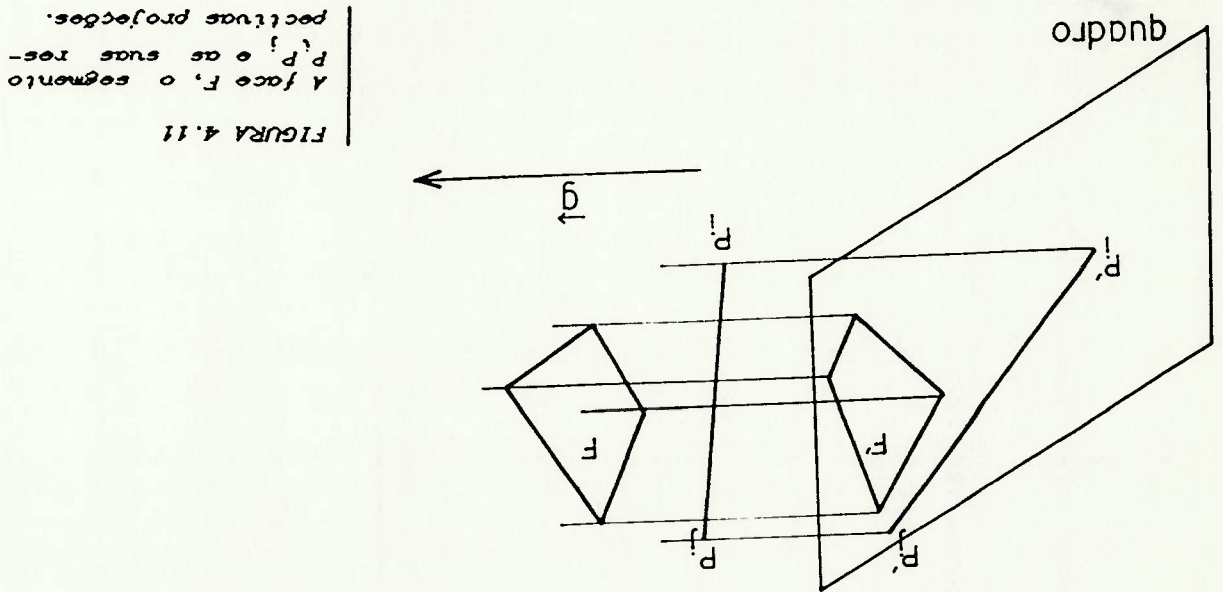


FIGURA 4.11
A face F , o segmento P_i, P_j e as suas respectivas projeções.

Para que o método simplex seja formulado, é necessário que o interior de F seja descrito. Procederemos analogamente ao que foi feito no algoritmo de Roberts.

Seja a figura 4.12 que mostra uma aresta F_i, F_j' da face F' e a respectiva normal \vec{n}_i que aponta para o interior de F' (note que \vec{n}_i é um ente que só existe na projeção). Um ponto A_i , projeção de A , localizado no interior de F' deve satisfazer as relações vectoriais:

$$\begin{aligned} (A_i - F_i) \cdot \vec{n}_1 &> 0 \\ &\vdots \\ (A_i - F_j) \cdot \vec{n}_2 &> 0 \\ &\vdots \\ (A_i - F_n) \cdot \vec{n}_n &> 0 \end{aligned}$$

(4.7)

Usando o que acabamos de ver, vamos determinar a visibilidade do segmento P_1P_2 em relação à uma face F de vértices F_1, F_2, F_3, F_4 . Dados (já em coordenadas homogêneas):

Exemplo:

- a) $\alpha > 0$, com α dado por (4.6)
- b) $0 \leq \lambda \leq 1$ (4.9)

Agora estamos em condições de afirmar que um segmento P_1P_2 estará obscurecido por uma face F se forem respeitadas a equação 4.8 e mais as condições:

$$\begin{aligned}
 ([Q(\lambda)] - [F_1]) [n_1] &> 0 \\
 ([Q(\lambda)] - [F_2]) [n_2] &> 0 \\
 \vdots & \\
 ([Q(\lambda)] - [F_n]) [n_n] &> 0
 \end{aligned}
 \tag{4.8}$$

Para que uma região de P_1P_2 esteja obscurecida, devemos ter necessariamente (em coordenadas homogêneas):

Lembrando que se a equação do segmento P_1P_2 é $[Q(\lambda)] = [P_1] + \lambda([P_2] - [P_1])$, com λ entre 0 e 1, então a equação da projeção P_1P_2 deve ser $[Q(\lambda)] = [P_1] + \lambda([P_1'] - [P_1])$. Atenção!! note que λ é o mesmo!!!

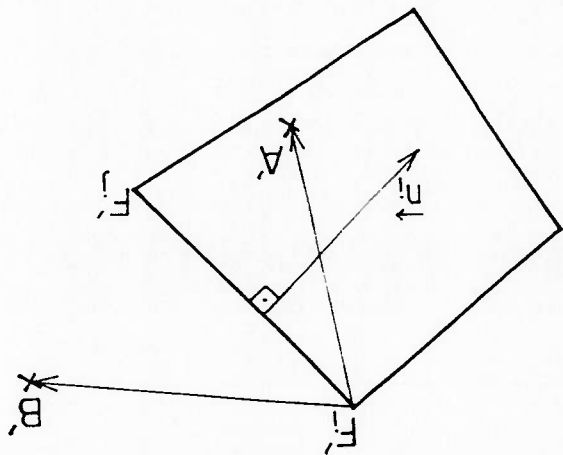


FIGURA 4.12
 n_i é a normal à aresta F_iF_{i+1} contida no próprio quadro de projeção.

b) Determinação da região interna à projeção da face F (fig. 4.14)

$$(x \ y \ z) \cdot \begin{bmatrix} -5 \\ -2 \\ 1 \\ -1 \end{bmatrix} = 0$$

Dai, matricialmente temos a equação do plano:

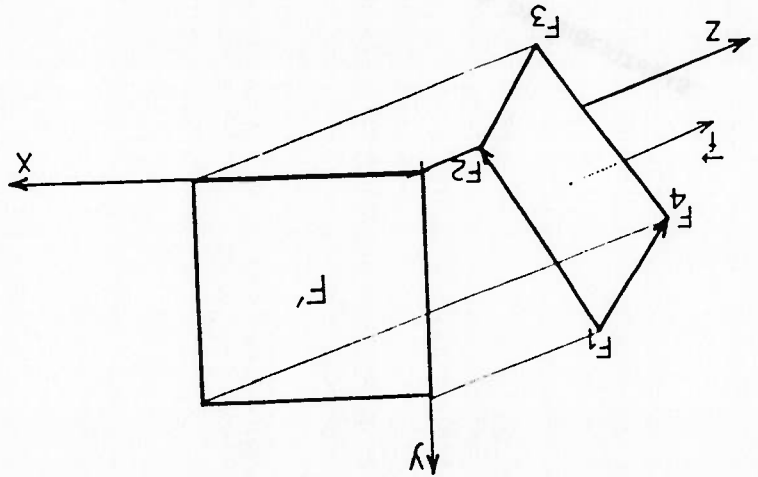
$$d = -(a, b, c) \cdot F_1 = -(a, b, c) \cdot (0, 1, 3) = -1$$

Para determinarmos o termo independente d fazemos:

$$\vec{r} = (F_2 - F_1) \wedge (F_3 - F_1) = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & -1 & -2 \\ 1 & 0 & 5 \end{vmatrix} = -5\vec{i} - 2\vec{j} + \vec{k}$$

$$\left. \begin{matrix} a = -5 \\ b = -2 \\ c = 1 \end{matrix} \right\}$$

FIGURA 4.13 $\vec{r} = (F_2 - F_1) \wedge (F_3 - F_1)$



a) Cálculo da equação do plano $ax+by+cz+d=0$ que contém a face F:
 A normal \vec{r} ao plano, que é o vetor $\vec{r}=(a,b,c)$, pode ser encontrado fazendo-se o produto vetorial entre os vetores F_1F_2 e F_1F_3 (fig. 4.13). Assim, temos:

$$\begin{aligned} [F_1] &= [0 \ 1 \ 0 \ 3 \ 0 \ 1] & [F_1] &= [-0.5 \ 0.5 \ 0.5 \ 0.5 \ 1] \\ [F_2] &= [0 \ 1 \ 0 \ 1 \ 0 \ 1] & [F_2] &= [3 \ 1 \ 1 \ 0 \ 0 \ 1] \\ [F_3] &= [1 \ 0 \ 0 \ 6 \ 0 \ 1] & & \\ [F_4] &= [1 \ 0 \ 1 \ 0 \ 8 \ 0 \ 1] & [g] &= [0 \ 1 \ 0 \ 1 \ 0] \end{aligned}$$

[g] é a direção que aponta para o observador impróprio.

Comecemos estabelecendo as relações (restrições) que devem ser obedecidas pelo conjunto dos pontos que são obscurecidos pela face F. Temos dois tipos de restrições: o primeiro tipo procura verificar quais pontos a possuem projectos Q' no interior da projecto de F (ver eq. 4.7). Nem todos esses pontos a são obscurecidos por F, pois eles podem estar na frente de F. Assim, segundo tipo de restrições visa verificar quais pontos estão atrás de F (ver eq. 4.8).

d) Determinação, através de um algoritmo de programação linear, da região de P1P2 obscurecida pela face F.

$$Q(\lambda) = P1 + \lambda(P2 - P1)$$

$$Q(\lambda) = [-0.5 \ 0.5 \ 1] + \lambda([3.1 \ 1 \ 1] - [-0.5 \ 0.5 \ 1])$$

$$Q(\lambda) = [-0.5 \ 0.5 \ 1] + \lambda(3.6 \ 0.5 \ 0)$$

ou em coordenadas homogêneas:

$$Q(\lambda) = P1 + \lambda(P2 - P1)$$

$$Q(\lambda) = [-0.5, 0.5, 1, -0.5, 0.5] + \lambda(3.1, 1, -0.5, 0.5, 0)$$

$$Q(\lambda) = [-0.5, 0.5, 1, -0.5, 0.5] + \lambda(3.6, 0.5)$$

A equação paramétrica de P1P2 é:

e) Equação da projecto do segmento P1P2

$$Q - F1' \geq n1 > 0$$

$$Q - F2' \geq n2 > 0$$

$$Q - F3' \geq n3 > 0$$

$$Q - F4' \geq n4 > 0$$

A região interna é formada pelos pontos Q tal que:

F1' F2'	+	n1	=	(1 0)
F2' F3'	+	n2	=	(0 1)
F3' F4'	+	n3	=	(-1 0)
F4' F1'	+	n4	=	(0 -1)

NORMAL (BIDIMENSIONAL)

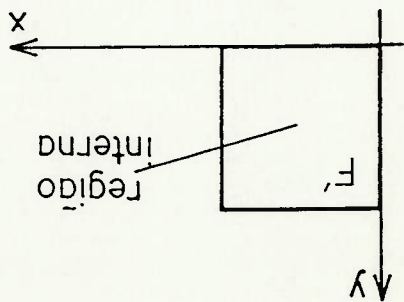


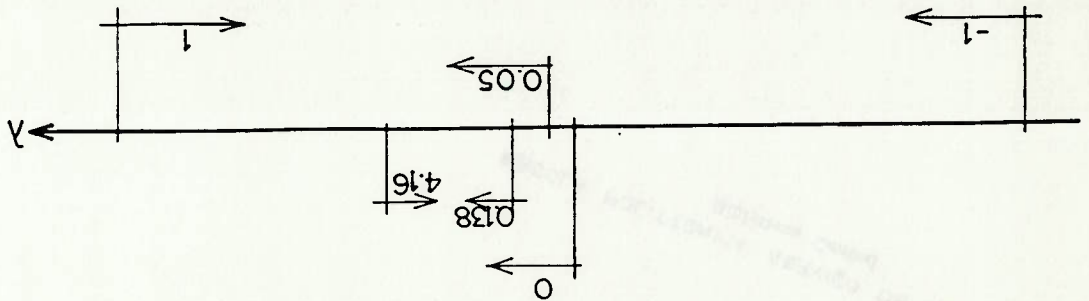
FIGURA 4.14
Região interna da face F

Para descrevermos a região interna da face F, devemos conhecer as normais às arestas F1F2, F2F3, F3F4 e F4F1. Atenção: Como estamos trabalhando com projectos sobre o quadro, os vetores são bidimensionais.

E daí, chegamos a dois trechos de P1P2 que não são obscuros, um que se inicia em P1 e vai à P3, e outro que vai de P4 à P2 (fig. 4.15).

Busca dos valores extremos (λ_{max} e λ_{min}) que atendem as restrições (A), (B) e (C).

FIGURA 4.15



Cada um deles é correspondente à:

$$\left\{ \begin{array}{l} \alpha(\lambda_{min}) = (0 \ 0.569 \ 0.431) = P3 \\ \alpha(\lambda_{max}) = (1 \ 0.708 \ 0.291) = P4 \end{array} \right.$$

$$f_{max} = 0.416 \quad f_{min} = 0.138$$

Fazendo isso, chegamos à (fig. 4.15):

f que atende as restrições e outra buscando o maior valor. esse conjunto, analogamente ao algoritmo de Roberts, devemos usar o método simplex duas vezes. Uma buscando o menor valor de pontos de P1P2 que são escondidos por f. Para determinarmos As restrições (A), (B) e (C) delimitam o conjunto dos

Além das restrições A e B existe, naturalmente,

$$0 < \lambda < 1 \quad (C)$$

$$\alpha = \frac{(g)(f)}{(P1P2) - P1 \cdot (f)} > 0 \quad (B)$$

$$\alpha = \frac{(g)(f)}{(P1P2) - P1 \cdot (f)} > 0$$

$$\alpha = \frac{(g)(f)}{(P1P2) - P1 \cdot (f)} > 0$$

O segundo tipo requer (ver eq. 4.6):

$$\begin{array}{l} (A) \\ \alpha(\lambda) - F1 \cdot \alpha > 0 + -0.5 + 3.6\lambda > 0 \\ \alpha(\lambda) - F2 \cdot \alpha > 0 + 0.5 - 0.5\lambda > 0 \\ \alpha(\lambda) - F3 \cdot \alpha > 0 + 1.5 - 3.6\lambda > 0 \\ \alpha(\lambda) - F4 \cdot \alpha > 0 + 0.5 + 0.5\lambda > 0 \end{array}$$

O primeiro tipo requer (ver eq. 4.7):

* Face_atual ← Primeiro elemento da lista Q;
 * Enquanto Face_atual não é o último elemento de Q faça:
 Início
 * Calcule os coeficientes a,b,c,d da equação do plano $ax+by+cz+d=0$ que contém Face_atual;
 * Se $c > 0$ então faça:
 Início
 * Calcule as normais n_k ($k=1..n$ número de arestas de Face_atual) às projeções das arestas de Seg_atual + Primeiro elemento da lista S;
 * Enquanto Seg_atual não é o último elemento de S faça:
 Início
 * Se não se pode provar rapidamente que Seg_atual está na frente de Face_atual usando-se o teste da superposição na direção de g então faça:

INICIO

A entrada da rotina Roberts consiste em duas listas Q e S contendo respectivamente faces e segmentos. O algoritmo processa as listas e devolve em S uma lista alterada, contendo apenas os fragmentos de faces e arestas realmente visíveis a partir de um determinado observador, ou seja, desenhando-se o conteúdo de S, já temos a figura final exata.
 A direção g é Oz.

ROTINA ROBERTS (Q: LISTA DE FACES ; S: LISTA DE SEGMENTOS)

O algoritmo testa todas as combinações relevantes de faces e arestas, armazenadas em duas listas separadas, que designaremos de Q e S respectivamente (Q+"quadriláteros" / S+"segmentos").

4.3.4.4. O algoritmo

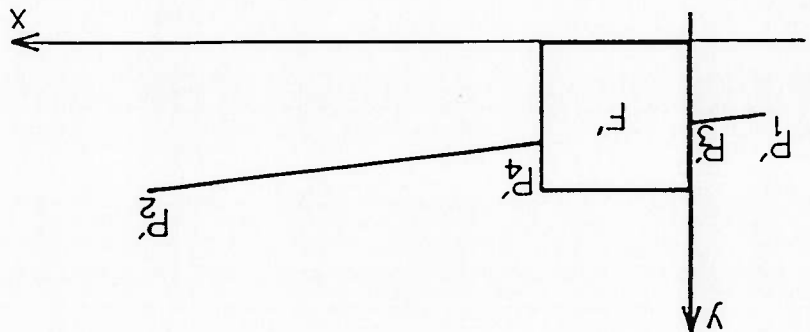


FIGURA 4.16

Regiões não obscurecidas: P1P3 e P4P2.

```

Início
* Se não se pode provar que as projeções de
Face_atual e Seg_atual não se sobrepe-
 usando-se o teste minmax então
 faça:
Início
* Se Seg_atual não pertence ao plano
  $ax+by+cz+d=0$  então faça:
Início
* Determine  $\lambda_{min}$  e  $\lambda_{max}$  que satisfa-
 zem 4.6 e 4.8.
* Se  $\lambda_{min} < \lambda_{max}$  faça:
Início
* p1 + Primeiro vértice de
Seg_atual;
* p2 + Segundo vértice de
Seg_atual;
* Caso:
a)  $\lambda_{max} < 1$  e  $\lambda_{min} > 0$  faça:
Início
* Cria um ponto p_novo1
correspondente a  $\lambda_{min}$ ;
* Cria um ponto p_novo2
correspondente a  $\lambda_{max}$ ;
* Seg_aux + Antecessor de
Seg_atual na lista S;
* Remove da lista S o
segmento Seg_atual;
* Instra na lista S o seg-
mento que liga p1 a
p_novo1;
* Instra na lista S o
segmento que liga p_novo2
a p2;
* Seg_atual + Seg_aux;
Fim
b)  $\lambda_{min} > 0$  e  $\lambda_{max} = 1$  faça:
Início
* Cria um ponto p_novo1
correspondente a  $\lambda_{min}$ ;
* Seg_aux + antecessor de
Seg_atual na lista S;
* Remove da lista S o
segmento Seg_atual;
* Instra na lista S o
segmento que liga p1 a
p_novo1;
* Seg_atual + Seg_aux;
Fim
c)  $\lambda_{min} = 0$  e  $\lambda_{max} < 0$  faça:
Início
* Cria um ponto p_novo2
correspondente a  $\lambda_{max}$ ;
* Seg_aux + antecessor de
Seg_atual na lista S;
* Remove da lista S o
segmento que liga p1 a
p_novo2;
* Seg_atual + Seg_aux;
Fim

```


Trata-se de um algoritmo misto, aplicando operações tanto no domínio do objeto como no domínio da imagem. O algoritmo inicialmente cria uma lista de faces ordenada por uma ordem baseada em relações como "está atrás" e "está na frente". Numa segunda fase, as faces da lista são mostradas em ordem na tela, mas com seus interiores pintados com as cores correspondentes. Desta maneira, as regiões mais distantes do observador são pintadas primeiro, e sobre elas são pintadas superfícies mais próximas. Assim, a cena resultante num vídeo raster, estará correta quanto à eliminação de linhas invisíveis. Intuitivamente, poderíamos pensar em ordenar as faces com base na profundidade apenas, mas veremos que isso leva a resultados errados.

4.4. O ALGORITMO DE NEWELL, NEWELL E SANCHIA

FIM DO ALGORITMO.

```

segmento Seg_atual;
o
*insira na lista S
segmento que liga
p_novos à p2;
*Seg_atual + seg_aux;
Fim
Fim caso
Fim
Senão
Início
* Seg_aux + antecessor de
Seg_atual na lista S;
* Remove Seg_atual da lista
S;
* Seg_atual + Seg_aux;
Fim
Fim (copiar)
Fim (minimax)
Fim (superposição em g)
Seg_atual + Sucessor de Seg_atual na lista S;
Fim (Não é o último segmento)
Fim (c ≠ 0)
Face_atual + Sucessor de Face_atual na lista Q;
Fim (Não é a última face de Q)

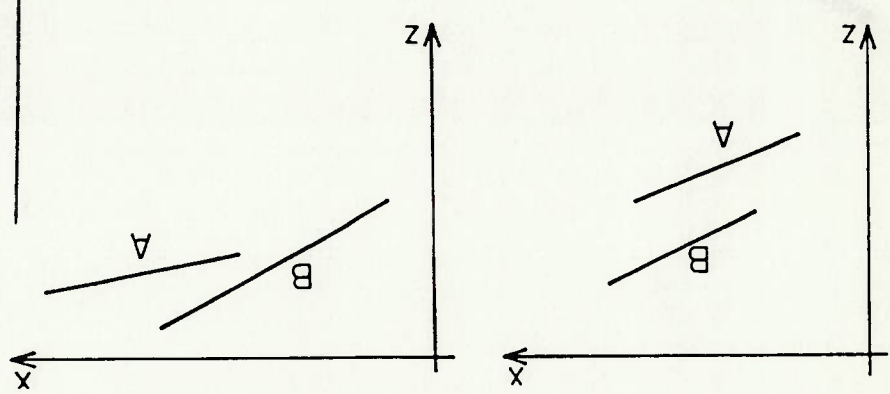
```

Uma maneira de se fazer a ordenação das superfícies seria ordená-las por ordem decrescente de distância, em relação ao observador, tomando por base o ponto mais próximo de cada superfície (fig.4.17). Certamente existe um conjunto de superfícies que se aplicado tal critério, não resultará na ordem correta (fig.4.17). Na verdade, qualquer que seja o critério baseado em ordenação por profundidade, sempre será possível montar um contra-exemplo. O seu princípio de funcionamento é ingênuo demais para dar certo!

4.4.1 Por que apenas a ordenação em função da profundidade falha?

Uma maneira de se fazer a ordenação das superfícies seria ordená-las por ordem decrescente de distância, em relação ao observador, tomando por base o ponto mais próximo de cada superfície (fig.4.17). Certamente existe um conjunto de superfícies que se aplicado tal critério, não resultará na ordem correta (fig.4.17). Na verdade, qualquer que seja o critério baseado em ordenação por profundidade, sempre será possível montar um contra-exemplo. O seu princípio de funcionamento é ingênuo demais para dar certo!

FIGURA 4.17
 Em (a), A está na frente de B segundo o critério de proximidade do ponto mais próximo da superfície ao observador. Em (b) a situação se inverte, mesmo usando o mesmo critério.

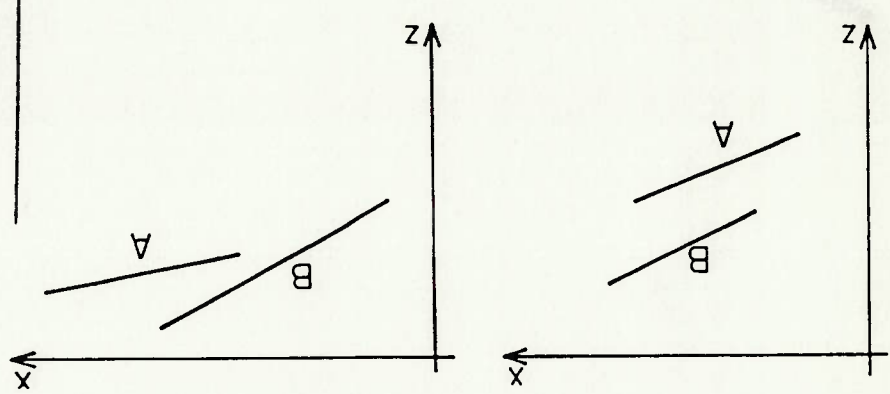


4.4.2 Teste mais sofisticado

4.4.1 Por que apenas a ordenação em função da profundidade falha?

Uma maneira de se fazer a ordenação das superfícies seria ordená-las por ordem decrescente de distância, em relação ao observador, tomando por base o ponto mais próximo de cada superfície (fig.4.17). Certamente existe um conjunto de superfícies que se aplicado tal critério, não resultará na ordem correta (fig.4.17). Na verdade, qualquer que seja o critério baseado em ordenação por profundidade, sempre será possível montar um contra-exemplo. O seu princípio de funcionamento é ingênuo demais para dar certo!

FIGURA 4.17
 Em (a), A está na frente de B segundo o critério de proximidade do ponto mais próximo da superfície ao observador. Em (b) a situação se inverte, mesmo usando o mesmo critério.



4.4.2 Teste mais sofisticado

Uma maneira de realizarmos o ordenamento seria utilizar os conceitos de "estar na frente" e "estar atrás".

Construamos as equações dos planos que contêm as diversas superfícies do nosso domínio de modo que quando substituirmos as coordenadas da posição do observador nestes planos, o sinal resultante é positivo.

Uma superfície P está na frente de uma Q se ao substituirmos as coordenadas dos vértices de P na equação do plano que contém Q, o sinal da expressão resultante para todos os vértices for positivo. De modo alternativo, P estará atrás de Q se todos os sinais forem negativos.

4.4.3 Complicações que podem surgir

Devemos notar que segundo o critério apontado acima, P estar na frente de Q não significa que Q está atrás de P !! Este fato curioso pode ser apreciado na fig.4.18. Outros casos patológicos podem surgir:

- a) Na fig.4.19, P está na frente de Q e, Q ao mesmo tempo está na frente de P !
- b) Na fig.4.20, P está na frente de Q, Q está na frente de R, mas R está na frente de P !

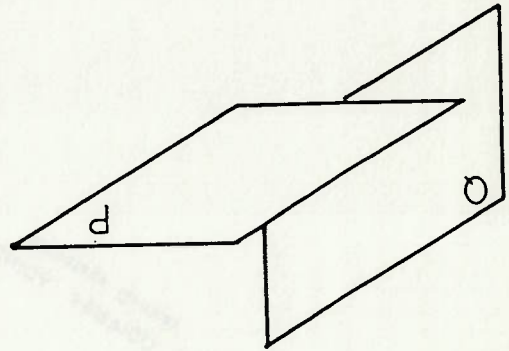


FIGURA 4.18
O fato de P estar na frente de Q não implica que Q esteja atrás de P.

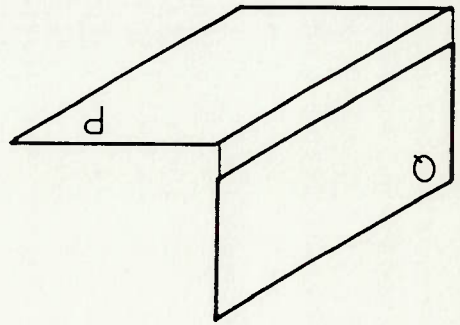


FIGURA 4.19
P está na frente de Q, e ao mesmo tempo, Q está na frente de P !!

Estes são fatos que o algoritmo deve estar preparado para resolver. Eles basicamente mostram que a propriedade transitiva não vale. O modo encontrado para contornar essa dificuldade é simplesmente dividir um dos polígonos em duas partes, usando como plano de corte o outro (ou um dos outros). Desse modo, o conflito desaparece.

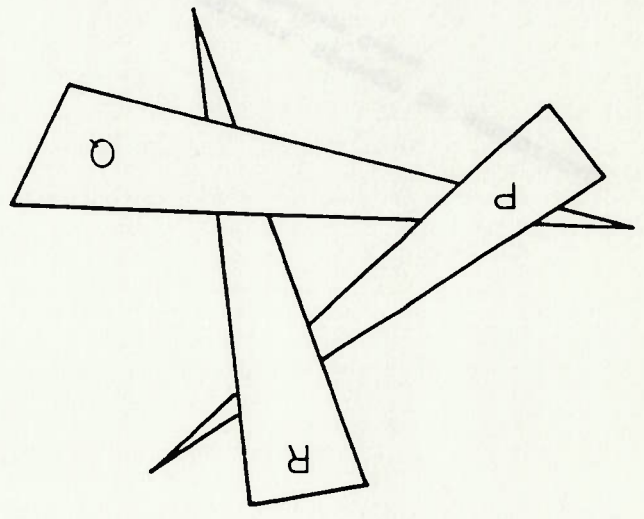


FIGURA 4.20
 Superposição cíclica: P
 sobrepõe Q, Q sobrepõe R,
 mas R sobrepõe P!

O algoritmo inicialmente faz uma ordenação por ordem decrescente de profundidade dos polígonos da cena, ou seja, o primeiro polígono da lista, que é o primeiro a ser desenhado, deve ser o polígono mais distante do observador. Essa ordenação é tomada como correta, a menos que dados dois polígonos P e Q (P ocorrendo antes de Q na lista), se comprove que Q está na frente de Q ou que Q está atrás de P. Devemos notar que estar atrás não é a negação lógica de estar na frente !!

ROTINA NEWEL-NEWEL-SANCHHA (Q : Lista de Faces)

< Esta rotina faz uma ordenação das faces usando o algoritmo de Newel-Newel e Sancha para remoção de linhas invisíveis. A entrada do algoritmo consta de uma lista de faces. Após o algoritmo ter sido rodado, a lista resultante poderá desenhada sequencialmente.>

Início
 * Se o número de superfícies na lista Q for maior do que 2 então faça :
 Início

* Faz ordenação por ordem decrescente de profundidade em relação ao observador. Basta verificar a coordenada Z !
 * Face_P ← Primeira face da lista Q.
 * Face_Q ← Sucessor de Face_P na lista Q.
 * Repita

* Se a faces Face_P e Face_Q não são coplanares, então faça:
 Início

* Se não se pode se provar que as projeções das faces Face_P e Face_Q não se sobrepõe na direção OZ, então faça:
 Início

* Se não se pode provar que as projeções das faces Face_P e Face_Q não se sobrepõe sobre o quadro de projeção usando-se o teste minímax, então faça :
 Início

* Se Face_Q não está na frente de Face_P e nem Face_P está atrás de Face_Q então faça :
 Início

* Se Face_Q está atrás de Face_P então faça:
 Início

O programa NAVGRAF também resolve o problema da remoção de linhas escondidas pelos métodos de Roberts e de Newell/Sancha. Eles são úteis na fase de pós processamento e na geração de figuras para relatórios. O algoritmo de Newell/Sancha é mais rápido do que o algoritmo de Roberts como vimos anteriormente, mas ele não é capaz de criar um desenho a ser enviado para o plotter, uma vez que ele se apóia na característica do monitor de vídeo raster de permitir o "desligamento" de pontos iluminados da tela. Desta maneira, o algoritmo de Roberts se torna essencial no pós processamento de malhas.

A figura 4.21 mostra um cubo numa perspectiva, onde as linhas ocultas foram removidas aplicando-se apenas o critério de ordenação por profundidade. Perceba que existe um erro no desenho final. Embora este método não forneça resultados sempre corretos, ele é muito rápido. Além disso, temos dois fatos que justificam sua inclusão em um programa pré processador de dados para o método dos elementos finitos. O primeiro é que a medida que a discretização da malha vai se tornando mais fina, com os elementos se tornando cada vez menores, esses erros vão tendendo a desaparecer. O segundo é que, em geral, salvo raras ocasiões, o analista necessita apenas de um desenho aproximado da malha para que ele verifique a correção de dados na fase de pré processamento. Por esses motivos é que este simples método é implementado no programa NAVGRAF (ver ref. [MAN]).

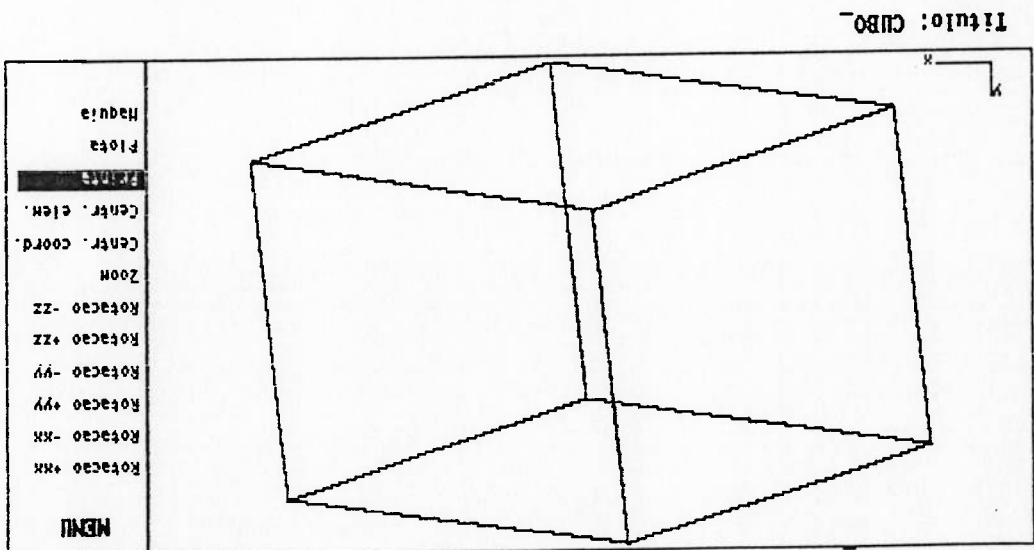
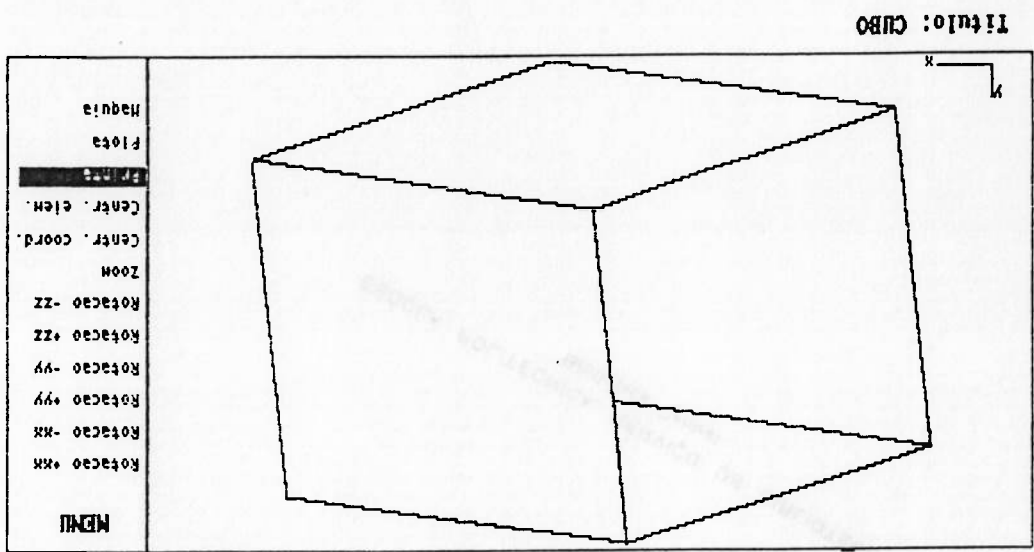
4.5. Aplicações e fechamento do capítulo

```

Fim
* Face_Q + Seu próprio sucessor.
* Se Face_Q é a última face da Lista Q então faça:
  Início
  * Face_P + Seu próprio sucessor.
  * Face_Q + Sucessor de Face_P.
  Fim;
Até que Face_P seja a última face da Lista Q.
Fim;
Fim;

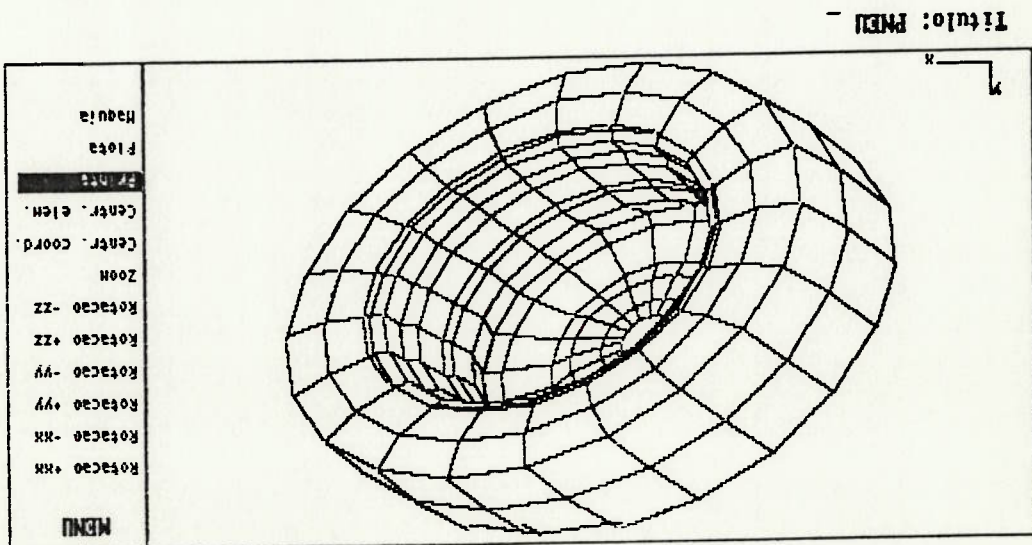
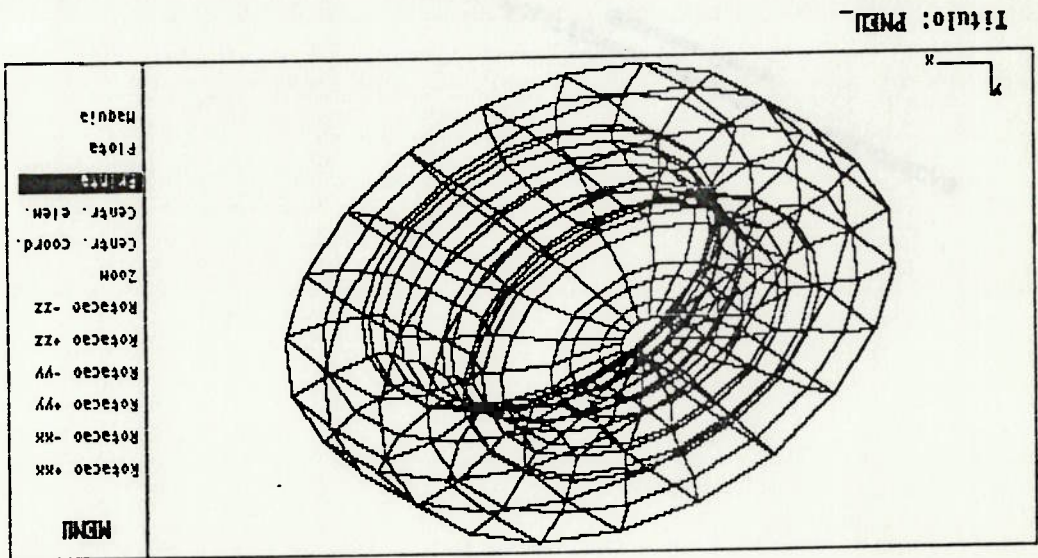
```

FIGURA 4.21
 Remoção de linhas invisíveis usando-se apenas a
 ordenação por profundidade.



A figura 4.22 mostra uma malha correspondente a um pneu, com linhas ocultas removidas usando-se o algoritmo de Newell e Sancha (O resultado é o mesmo se for aplicado o algoritmo de Roberts).

FIGURA 4.22
 Remoção de linhas invisíveis usando-se o algoritmo
 de Newell/Sancha.



**5. REPRESENTAÇÃO DE
TENSÕES E DEFORMAÇÕES**

5.0. Introdução

Em essência, a representação de tensões e deformações é a principal atividade da fase de pós processamento de dados no método de elementos finitos. Inicialmente, trataremos do mapeamento de níveis de tensões através do traçado de curvas de isovalor, e em seguida iremos discutir a representação de estruturas deformadas.

Nos restringiremos apenas aos aspectos gráficos da fase de pós processamento. Consideramos que todas as atividades envolvidas no cálculo de valores nos nós de uma malha, sejam tensões ou deformações, não fazem parte do escopo do trabalho.

É essencial para os algoritmos relativos ao mapeamento de tensões, que veremos mais adiante, que os valores sejam únicos em cada nó. Entretanto, não é isso que ocorre no método dos elementos finitos. No método, após a montagem da matriz de rigidez global, e a subsequente resolução do sistema de equações formado, somente os valores para os deslocamentos nodais são fornecidos. Apenas após um posterior processamento é que as tensões no interior de cada elemento podem ser conhecidas. Como a continuidade das funções relativas às tensões não são impostas no método, um resultado comum é termos tantos valores nodais quantos foram os elementos que compõem o elemento. Lham aquele nó.

Apesar da precisão para a distribuição de tensões não ser a mesma que para o campo de deslocamentos (devido à existência de derivadas [BUR]), usaremos, para os algoritmos nesta seção, as mesmas funções de forma do campo de deslocamentos.

Mostraremos dois tratamentos. O primeiro se baseia no mapeamento de funções contínuas, e o segundo em funções discretas (divisão em pequenas regiões triangulares).

5.1. Traçado de curvas isotensão.

Existem diversas maneiras para a obtenção de apenas um valor de tensão por nó do modelo, como pode ser apreciado nas refs [BUR], [HIN], [BAR], ou no apêndice ao final do trabalho. Resumidamente, podemos afirmar que, no interior dos elementos, existem pontos onde os valores das tensões apresentam um caráter "superconvergente", isto é, a medida que a malha vai sendo refinada, os valores nestas localidades convergem para a estabilidade mais rapidamente que qualquer outro valor obtido num outro ponto qualquer. Em particular, as tensões calculadas diretamente nos nós do elemento em geral são ruins. Os diferentes métodos de obtenção de um valor único para cada nó procuram extrapolar os valores nos pontos superconvergentes. Para os leitores interessados, recomendamos as referências citadas acima. Passamos agora ao estudo do traçado de curvas.

5.1.1. Tratamento usando mapeamento contínuo.

Suponhamos que temos um elemento finito isoparamétrico no espaço x, y, z , que se mapeia num elemento mãe¹ num domínio descrito pelas coordenadas ξ, η . Sobre ele desejamos traçar curvas de isovalor correspondentes a uma distribuição de tensões $T(x, y, z)$ (fig. 5.1).

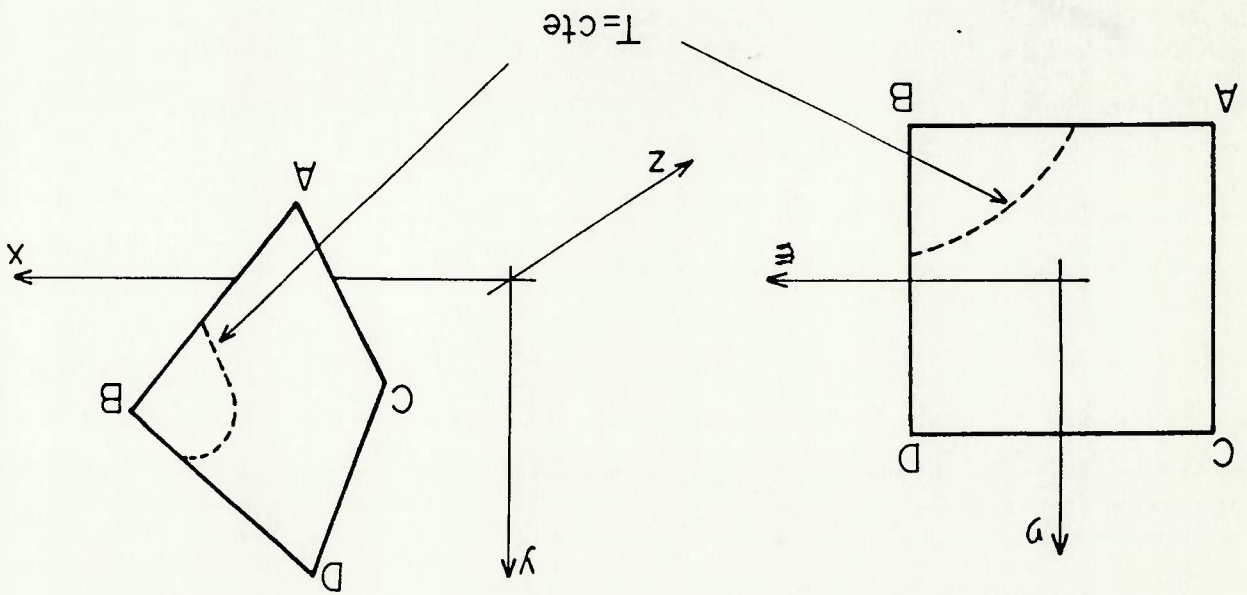


FIGURA 5.1

Curva de isovalor traçada no elemento real e no elemento mãe (domínio ξ, η)

Mapear a função T diretamente sobre a superfície do elemento no espaço real x, y, z , seria computacionalmente muito difícil, pelos seguintes motivos:

a) A distorção geométrica do elemento no espaço pode ser excessiva.

1 "Parent Element"

Essas lentes nodais podem ser obtidas por qualquer método e tratamento.

Onde $\frac{\partial \xi}{\partial T} = \left[\frac{\partial \xi}{\partial H} \right] \cdot \langle T^0 \rangle$ e $\frac{\partial \eta}{\partial T} = \left[\frac{\partial \eta}{\partial H} \right] \cdot \langle T^0 \rangle$ (5.4)

$\frac{\partial \xi}{\partial T}(\xi, \eta) + \frac{\partial \eta}{\partial T}(\xi, \eta) = 0$ (5.3)

Claramente, o diferencial $DT(\xi, \eta) = 0$. Em outros termos:

$\xi^{i+1} = \xi^i + dL \cdot \delta \xi^i$
 $\eta^{i+1} = \eta^i + dL \cdot \delta \eta^i$ (5.2)

Desajamos traçar uma curva $T(\xi, \eta) = cte$ usando um algoritmo iterativo. A estratégia a ser empregada será a seguinte: A partir de um ponto conhecido (ξ^i, η^i) , determinaremos (ξ^{i+1}, η^{i+1}) , distante dL do ponto anterior, através da relação:

Onde $\langle T^0 \rangle$ são os valores das tensões para os nós ligados ao elemento e e $[H(\xi, \eta)]$, a matriz das funções de forma.

$T(\xi, \eta) = [H(\xi, \eta)] \langle T^0 \rangle$ (5.1)

Vamos escrever $T(\xi, \eta)$ da seguinte maneira:

Desta maneira, o natural seria recorrermos ao elemento mãe no espaço bidimensional ξ, η . O traçado de curvas isovalor se daria aí, e posteriormente as curvas seriam mapeadas para o elemento no espaço real x, y, z . A questão naturalmente levantada agora é: Como vamos mapear $T(\xi, \eta)$ no espaço ξ, η ? É o que passamos a apreciar agora.

- b) Tanto $T(x, y, z)$ como a superfície do elemento são funções de três variáveis (3D).
- c) $T(x, y, z)$, provavelmente, não é um polinômio.

Vamos mapear sobre a superfície mostrada na figura 5.3 a

Exemplo:

$T(x, y, z) = T(x(\xi, \eta), y(\xi, \eta), z(\xi, \eta))$,
 sendo $[x(\xi, \eta), y(\xi, \eta), z(\xi, \eta)] = [H(\xi, \eta)]$ e (X^e) ,
 e (X^e) as coordenadas nodais dos vértices do elemento.

Após a curva isotensão ter sido traçada no domínio ξ, η , devemos mapear seus pontos para o domínio do elemento real, x, y, z . Isso, sendo o elemento isoparamétrico, é obtido fazendo-se:

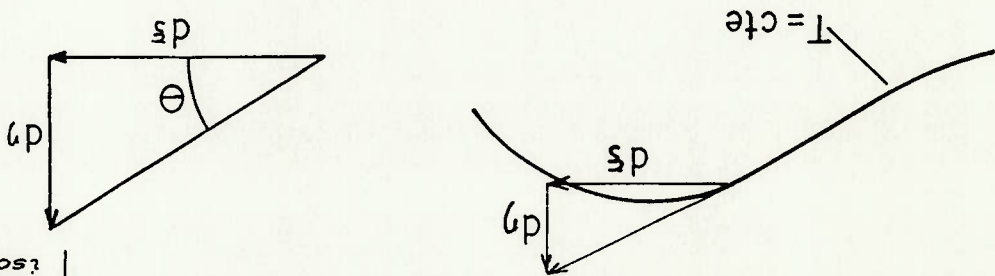
$$\delta \eta_i = \frac{\left[\frac{\partial T}{\partial \xi}(\xi, \eta, z) \right]_z + \left[\frac{\partial T}{\partial \eta}(\xi, \eta, z) \right]_z}{1} \cdot \frac{\partial \xi}{\partial \xi}(\xi, \eta, z) \quad (5.6)$$

$$\delta \xi_i = \frac{\left[\frac{\partial T}{\partial \xi}(\xi, \eta, z) \right]_z + \left[\frac{\partial T}{\partial \eta}(\xi, \eta, z) \right]_z}{1} \cdot \frac{\partial \eta}{\partial \xi}(\xi, \eta, z) \quad (5.5)$$

facilmente a:

Deste fato, e das relações (5.3) e (5.4) chegamos

E importante agora a interpretação geométrica de $\delta \xi$ e $\delta \eta$. Eles são componentes da tangente à curva $T(\xi, \eta) = cte$ (fig. 5.2).



Interpretação geométrica de $(d\xi, d\eta)$: vetor tangente à curva de iso-valor.

FIGURA 5.2

Para que a tensão T=21 seja mapeada, é necessário que um ponto inicial (r_0, n_0) seja determinado. Para determinarmos tal ponto inicial, vamos pesquisar os lados do elemento. Percebendo que a distribuição de tensões ao longo de cada lado é linear

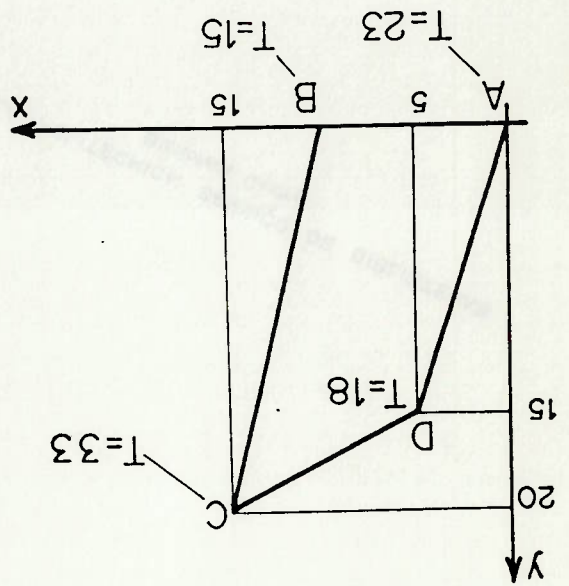


Figura 5.3
Elemento sobre o qual a curva T=21 será traçada

$$r_{i+1} = r_i + dL \cdot \frac{\Delta i}{(1.25 + 5.75 r_i)}$$

$$n_{i+1} = n_i + dL \cdot \frac{\Delta i}{(3.25 + 5.75 r_i)}$$

onde $\Delta i = \sqrt{(3.25 + 5.75 r_i)^2 + (1.25 + 5.75 n_i)^2}$
 $dL = 0.1$ (constante arbitrária)

(5.5) e (5.6) chegamos a:

$$[H] = \begin{bmatrix} n_1 & n_2 & n_3 & n_4 \\ 0.25(1-r)(1-n) & 0.25(1+r)(1-n) & 0.25(1+r)(1+n) & 0.25(1-r)(1+n) \end{bmatrix}$$

As funções de forma utilizadas são:
 tensão T=21 (unidades não importam).

Fazendo as operações das relações

3 Discussão sobre a importância da precisão das curvas logo a seguir.

No caso de elementos cujas funções de forma associadas são de grau baixo, por exemplo o elemento isoparamétrico de quatro nós classe C^0 , a precisão das curvas traçadas é muito boa. Experimentos realizados indicam que o valor da tensão ao longo da curva traçada não se distancia de mais de um por cento do valor a ser mapeado. Por outro lado, para elementos cujas funções de forma são de grau elevado (quadrático completo, por exemplo), a precisão atingida pode ser baixa, como mostra a figura 5.6.

Partindo de A até chegamos à B e de C chegamos à D. A figura 5.4 mostra o resultado obtido. As curvas têm aspecto suave, porém na verdade são constituídas de pequenos segmentos ligando cada par de pontos $(t^1, n^1) - (t^2, n^2)$ como podemos apreciar na figura 5.5.

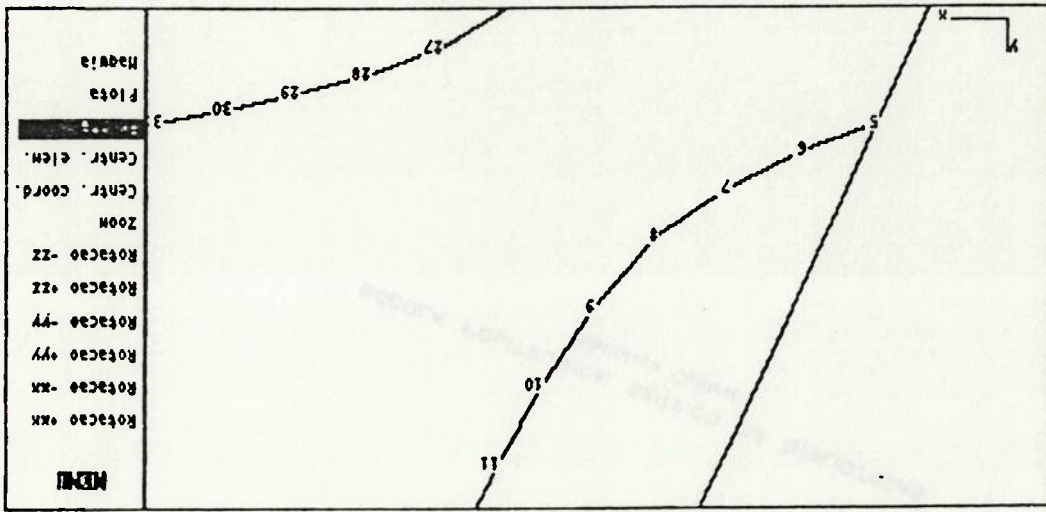
São eles (Em coordenadas (r, θ))
 Lado 1-2 : Ponto A(-0,5, -1)
 Lado 2-3 : Ponto B(+1, -0,33)
 Lado 3-4 : Ponto C(-0,6, +1)
 Lado 4-1 : Ponto D(-1, -0,2)

devido às características das funções de forma dos elementos isoparamétricos do exemplo, determinamos, através de interpolação linear, quatro pontos onde a tensão é igual à 21.

Ampliação de um trecho da curva mostrando que na verdade ela é composta por pequenos segmentos.

FIGURA 5.5

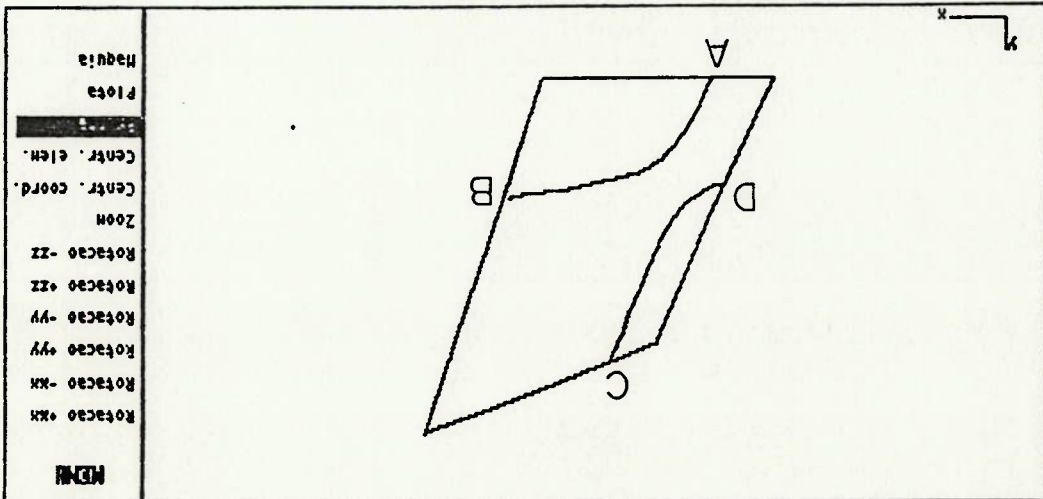
Título: IRACAO DE CURVA POR PEQUENOS SEGMENTOS.



Curvas correspondentes a $T=25$ traçadas. A termina em B e, C termina em D.

FIGURA 5.4

Título: CURVAS ISO-VALOR ISOPARAMETRICO 4 NOS.



4 Por mais discutível que seja a ânsia por boa precisão, o mínimo que se pode esperar de um algoritmo para traçado de curvas de iso-valor é que ele trace curvas que devem ser fechadas, fechadas.

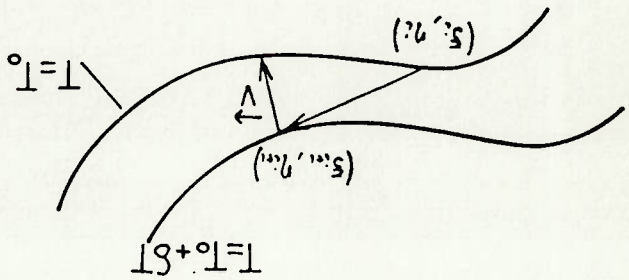


FIGURA 5.7
 (ξ', η') apresenta erro de ν em relação à sua posição correta, sobre a curva $T=I_0$.

Figura 5.7.

A fim de se melhorar a precisão do traçado pode-se criar um corretor para os valores da curva a ser traçada. Seja a

Trata-se do elemento isoparamétrico de oito nós classe C^1 . Note que existe um vale para a função $T(\xi, \eta)$ no centro do elemento e que o algoritmo para o mapeamento traga uma curva aberta, quando ela deveria ser fechada.

Título: TRACADO ISO VALOR, METODO CONTINUO, CURVA FECHADA ?

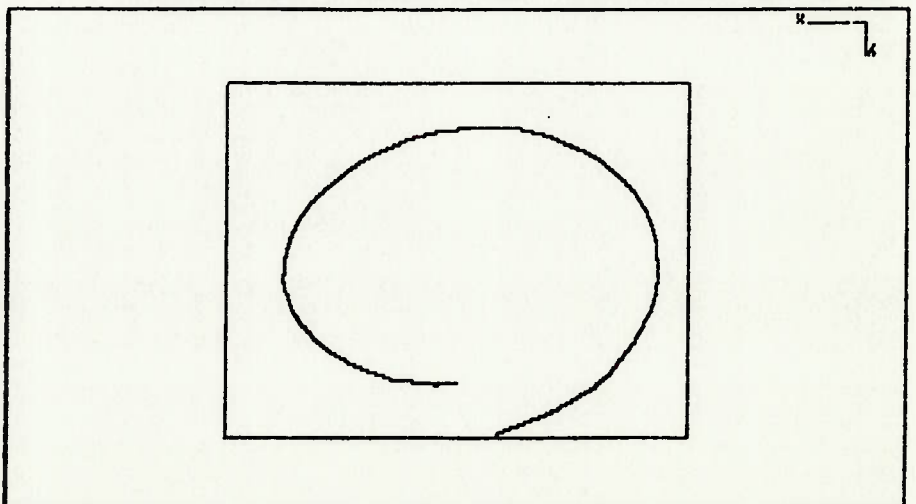


FIGURA 5.6
 Elementos cujas funções de forma têm grau elevado podem apresentar problemas: No caso, uma curva fechada acaba sendo descarterizada por acúmulo de erros de precisão numérica.

5^o A relação pode ser facilmente obtida lembrando-se que o gradiente de $T(\xi, \eta)$ é perpendicular às curvas de nível e que a

$$\frac{\Delta T}{(T_{correto} - T(\xi^i, \eta^i))}$$

o comprimento de \hat{v} deve ser $\frac{\Delta T}{(T_{correto} - T(\xi^i, \eta^i))}$. Usando-se essas novas relações é possível traçarmos uma curva muito melhor para o caso de elemento isoparamétrico de oito nós, como é mostrado na figura 5.8 (A título de teste de precisão, forçamos a curva ser traçada continuamente num laço

$$\delta \eta^i = \frac{\partial \eta}{\partial T(\xi^i, \eta^i)} \cdot (T_{correto} - T(\xi^i, \eta^i)) \quad (5.10)$$

$$\delta \xi^i = \frac{\partial \xi}{\partial T(\xi^i, \eta^i)} \cdot (T_{correto} - T(\xi^i, \eta^i)) \quad (5.9)$$

Onde:

$$\begin{aligned} \xi^{i+1} &= \xi^i + \delta \xi^i \\ \eta^{i+1} &= \eta^i + \delta \eta^i \end{aligned} \quad (5.8)$$

Usando este corretor, as relações de (5.2) passam a ser:

valor a ser mapeado.

Onde estamos denominando $T_{correto}$ como sendo o próprio

$$\hat{v} = \frac{\|\Delta T(\xi, \eta)\|}{\Delta T(\xi, \eta) \cdot (T_{correto} - T(\xi^i, \eta^i))} \quad (5.7)$$

$$\hat{v} = \frac{\|\Delta T(\xi, \eta)\|}{\Delta T(\xi, \eta)} \cdot \frac{\|\Delta T(\xi, \eta)\|}{(T_{correto} - T(\xi^i, \eta^i))}$$

Suponha que no passo i o ponto obtido seja exato, e que o ponto do passo $i+1$ seja gerado com uma pequena imprecisão. Para que essa imprecisão seja corrigida, devemos somar o vetor \hat{v} à posição (ξ^{i+1}, η^{i+1}) para que a posição exata seja recuperada. Podemos estimar \hat{v} pela relação:

a) É muito lento uma vez que deve calcular coordenadas de muitos pontos no traçado de uma curva.

b) Apresenta dificuldade na busca de um ponto inicial da curva. A tarefa de se determinar um ponto na fronteira é fácil. Os problemas começam quando devemos determinar se esse ponto também é ponto final de uma linha que já foi traçada. Além disso, pontos iniciais de curvas fechadas como a da figura 5.8 são difíceis de serem localizados.

O algoritmo não apresenta grandes dificuldades do ponto de vista matemático. Sua precisão aritmética é muito boa. Entretanto, o algoritmo que acabamos de apreciar oferece problemas em sua implementação. Entre eles, podemos destacar:

5.1.1.1. Problemas com a implementação do algoritmo.

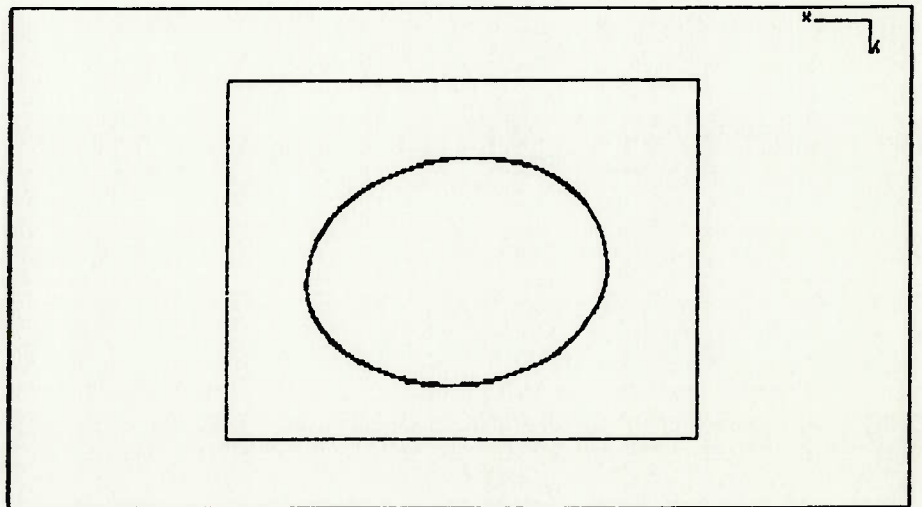


FIGURA 5.8
Com o corretor, a curva fechada é traçada fecha-
da ...

que se fecha três vezes). Agora, com corretor, as imprecisões se tornam desprezíveis em face à resolução gráfica dos dispositivos de saída.

6 Sendo a distribuição linear e a região triangular, temos apenas os casos a considerar (deixando de lado o caso em que todo o elemento apresenta $T=I_0$):

A ideia básica consiste em se dividir cada elemento da malha em pequenas regiões onde podemos supor uma distribuição de tensões fácil de ser mapeada. Sugerimos a divisão em pequenos triângulos, uma vez que com três vértices onde são conhecidas as tensões, podemos supor uma distribuição linear. Como a distribuição é linear, para traçarmos um trecho em seu interior que é correspondente à curva $T=I_0$, basta determinarmos dois pontos pertencentes a dois lados e ligá-los por um segmento de

ta" (curvas fechadas são traçadas como curvas fechadas). método dos elementos finitos, e mantém as formas da curva "exacta" (curvas fechadas são traçadas como curvas fechadas). Apresentamos agora uma outra abordagem para o traçado de curvas isovalor. O algoritmo resultante, que é implementado no programa NAVGRAF [ALE][MAN], é simples, rápido, tem precisão compatível com a confiabilidade alcançada para as tensões no método dos elementos finitos, e mantém as formas da curva "exacta" (curvas fechadas são traçadas como curvas fechadas).

5.1.2. Tratamento usando mapeamento discreto.

Essas dificuldades pesam muito mais do que a aparente vantagem de o algoritmo fornecer boa precisão numérica. Podemos afirmar "aparente vantagem" pois de nada adianta possuímos um algoritmo muito mais preciso para o traçado de curvas de isovalor que a própria função distribuição de tensões. Devido aos problemas apresentados acima é que passamos a discutir outra classe de algoritmos.

c) Apresenta dificuldade no estabelecimento de uma condição de parada. A cada ponto, o algoritmo deve verificar se ele pertence ainda ao interior do elemento. No caso de curvas internas fechadas o critério é mais complicado.

- a) Existem dois pontos nos lados do elemento que correspondem a $T=10$;
- b) Existe apenas um ponto que coincide com um dos vértices do elemento;
- c) Não existe pontos no triângulo que correspondem a $T=10$;
- d) Se os dois pontos do caso a) coincidem com dois vértices do triângulo, então a linha iso-valor em seu interior coincide com um dos seus lados.

Esta é uma questão difícil de ser respondida, mas em princípio, podemos dizer que ele deve ser compatível com a ordem de precisão atingida pela função distribuição de tensão. Assim, por exemplo, no caso do elemento isoparamétrico de quatro nós classe C^0 , como temos apenas aproximação linear para as tensões, é razoável termos apenas um ponto gerado em seu interior.

5.1.2.1. Qual o número ideal de triângulos ?

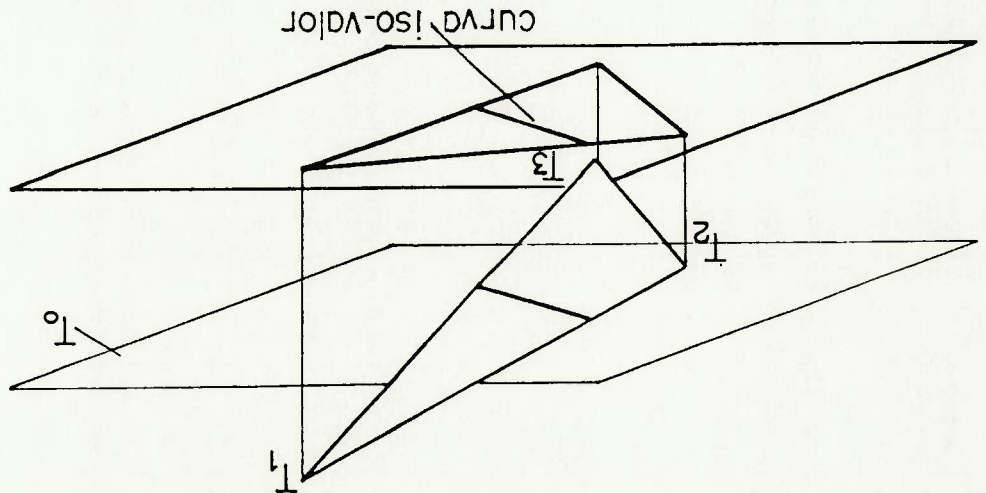
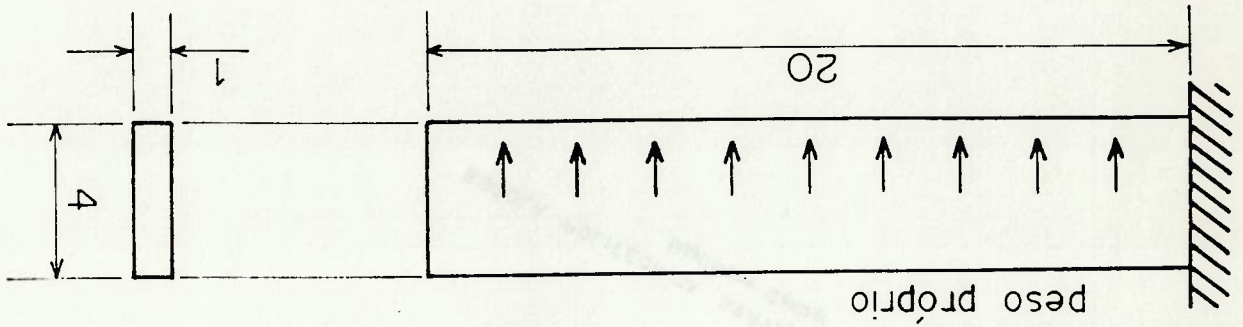


FIGURA 5.9

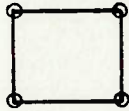
Para uma distribuição de tensões linear numa região triangular, basta determinarmos dois pontos pertencentes a dois lados e ligá-los por um segmento de reta.

Para uma distribuição de tensões linear numa região triangular, basta determinarmos dois pontos pertencentes a dois lados e ligá-los por um segmento de reta.

reta (fig.5.9). Percebe-se que por isso, o traçado de uma curva completa deve ser muito rápido. O número de triângulos gerados no interior do elemento é ditado pela precisão a ser atingida.

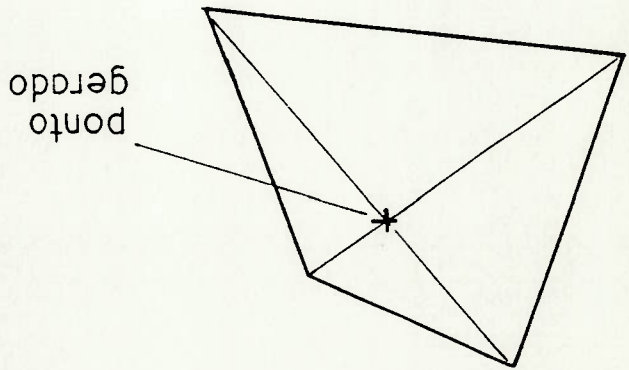


peso próprio



Modelo A - Viga em balanço modelada por elementos quadriláteros de 4 nós.

FIGURA 5.11



ponto gerado

Um ponto gerado no interior do elemento dá origem a quatro triângulos.

FIGURA 5.10

Onde α_i é a tensão no nó i .

$$\text{tensão no centro do elemento} = \alpha(\xi=0, \eta=0) = \frac{\alpha_1 + 0\alpha_2 + 0\alpha_3 + 0\alpha_4}{4}$$

rior (no centro), dando origem a quatro elementos triangulares (fig. 5.10). Neste caso, pode-se provar facilmente que (usando funções de forma de elemento quadrilátero de 4 nós C^0):

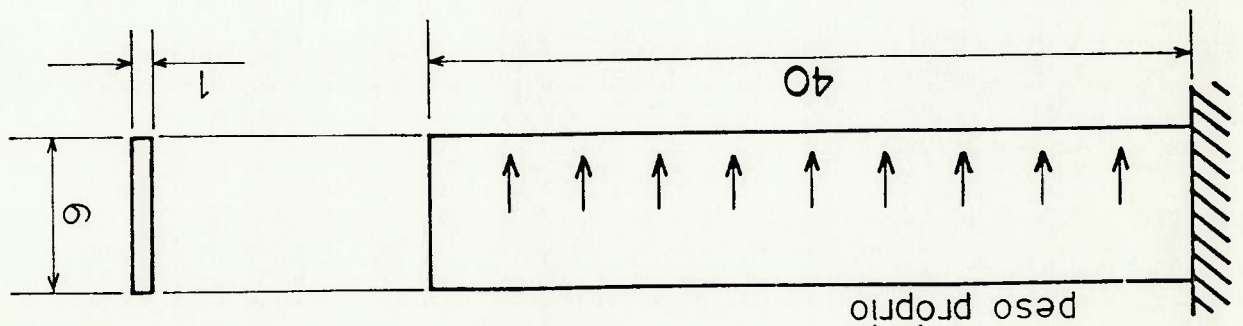
Serão mostrados agora alguns resultados do uso dos algoritmos vistos neste capítulo. Basicamente, temos dois modelos para os quais desejamos conhecer as curvas de

5.1.3. Aplicações.

Onde $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ são tensões correspondentes aos vértices do quadrilátero e $\sigma_5, \sigma_6, \sigma_7, \sigma_8$ são tensões correspondentes aos nós localizados no ponto médio dos lados.

$$\text{tensão no centro do elemento} = \sigma(\xi=0, \eta=0) = - \left(\frac{01+02+03+04}{4} \right) + \left(\frac{05+06+07+08}{2} \right)$$

No caso do elemento isoparamétrico de 8 nós C^0 , o algoritmo implementado no programa NAVGRAF o divide em doze pequenos triângulos. No caso do ponto no centro do elemento, pode-se verificar que:



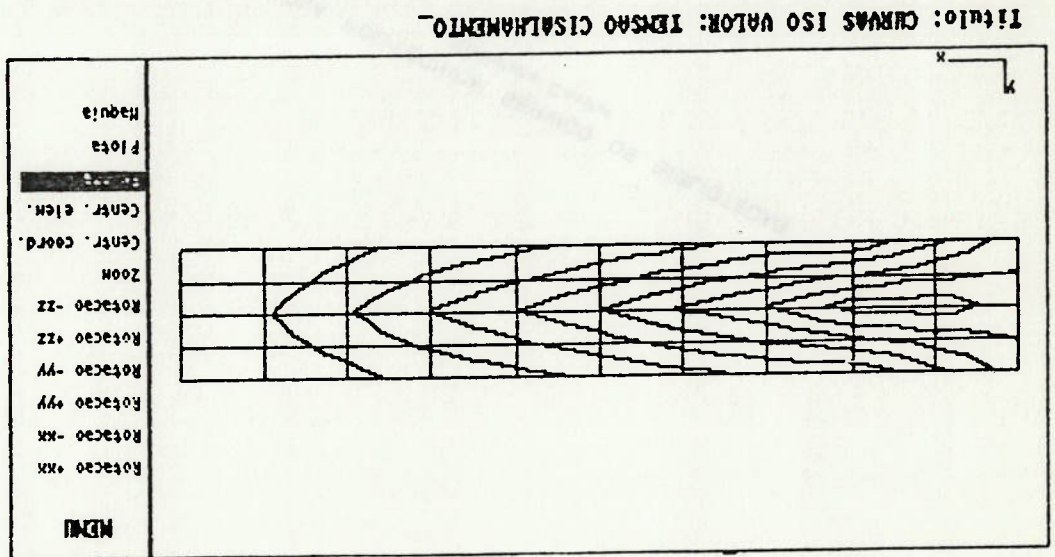
Modelo B - Viga em balanço modelada por elementos quadriláteros de 8 nós.

FIGURA 5.12

A figura 5.13 mostra as curvas isotensão geradas por esse algoritmo para o modelo A. As tensões nodais de cisalhamento foram obtidas de acordo com as técnicas mostradas no apêndice, e o traçado foi executado usando-se o algoritmo de divisão por

Curvas de isotensão geradas pelo algoritmo da divisão por triângulos. Modelo A.

FIGURA 5.13



isotensão. O primeiro é uma viga em balanço, de dimensões 20m x 4m x 1m sujeita, a um carregamento vertical distribuído de $w = 1.0ES$ N/m modelada através de elementos quadriláteros isoparamétricos de 4 nós de estado plano de tensões (classe C^0), como mostra a figura 5.11. O segundo modelo também é uma viga em balanço, mas de dimensões 40m x 6m x 1m, sujeita a um carregamento vertical de $w = 300$ N/m modelada através de elementos quadriláteros isoparamétricos de 8 nós de estado plano de tensões (classe C^0), como indica a figura 5.12. Vamos denotar o primeiro modelo como modelo A e o segundo como modelo B. Os elementos acima citados pertencem à biblioteca do programa NAVSUB, do Departamento de Engenharia Naval e Oceânica da Escola Politécnica da USP.

triângulos. Note que, embora a forma das curvas serem satisfatórias, nas bordas do modelo, existem inconsistências (as tensões de cisalhamento no topo e no fundo deveriam ser nulas).

A figura 5.14 (a) mostra as curvas isotensão geradas para o caso do modelo B. O traçado das curvas foi feito usando-se o algoritmo da divisão por triângulos. Cada elemento foi dividido em 12 triângulos cada. As tensões nodais não foram extrapoladas a partir das tensões nos pontos superconvergentes (como é mostrado no apêndice), mas sim obtidos diretamente pelo MEF⁷. Na figura 5.14 (b), é mostrado o mesmo modelo na mesma situação, mas tendo as tensões extrapoladas a partir daqueles pontos.

A figura 5.15 mostra uma comparação entre as curvas obtidas pelos dois tratamentos mostrados no trabalho.

5.1.4. Conclusões sobre o mapeamento de tensões.

O tratamento usando a "discretização" de um elemento finito em pequenos triângulos oferece um método mais simples e rápido do que o uso de uma função contínua, sem que a necessária precisão fique prejudicada.

O número de triângulos em que um dado elemento é dividido perde importância à medida em que a malha vai se tornando mais refinada, tendo elementos cada vez menores. Em outras palavras, as curvas de isotensão vão convergindo para o traçado "exato" à medida que a discretização da malha vai aumentando.

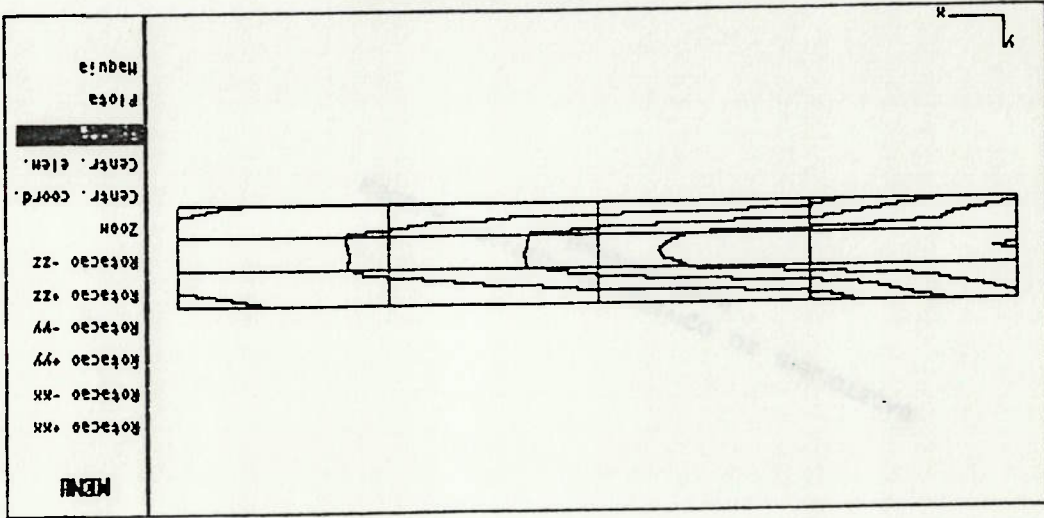
Pelos motivos apontados acima é que o algoritmo da divisão em triângulos foi escolhido para ser implementado no programa NAVGRAF [ALE][MAN].

⁷ Foi tirada a média aritmética das tensões atuantes em cada nó

Curvas de isovalor geradas pelo algoritmo da divisão por triângulos. Modelo B.

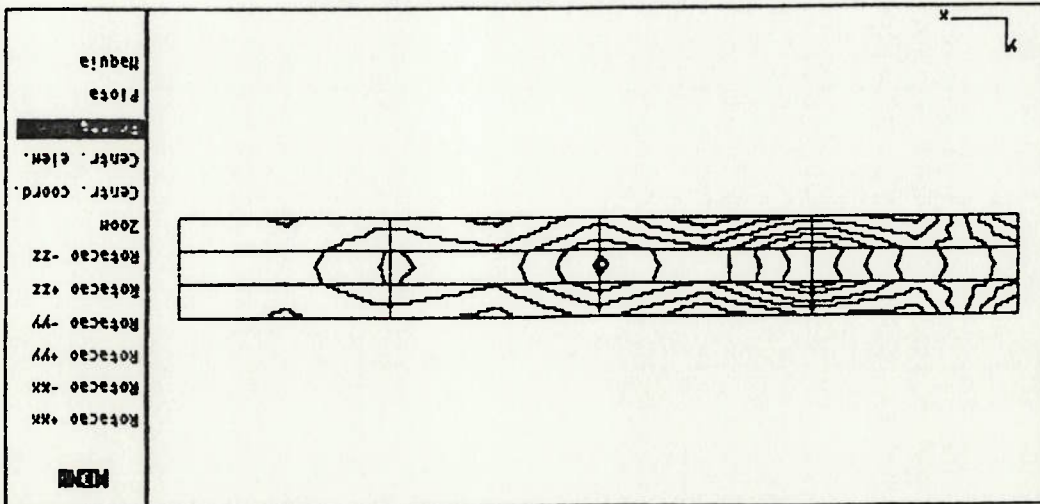
FIGURA 5.14

Título: CURVAS ISOVALOR: MOD B, TENSÕES EXTRAPOLADAS.



(b)

Título: CURVAS ISOVALOR CISAIS, MOD B TENSÕES PELO FEM.

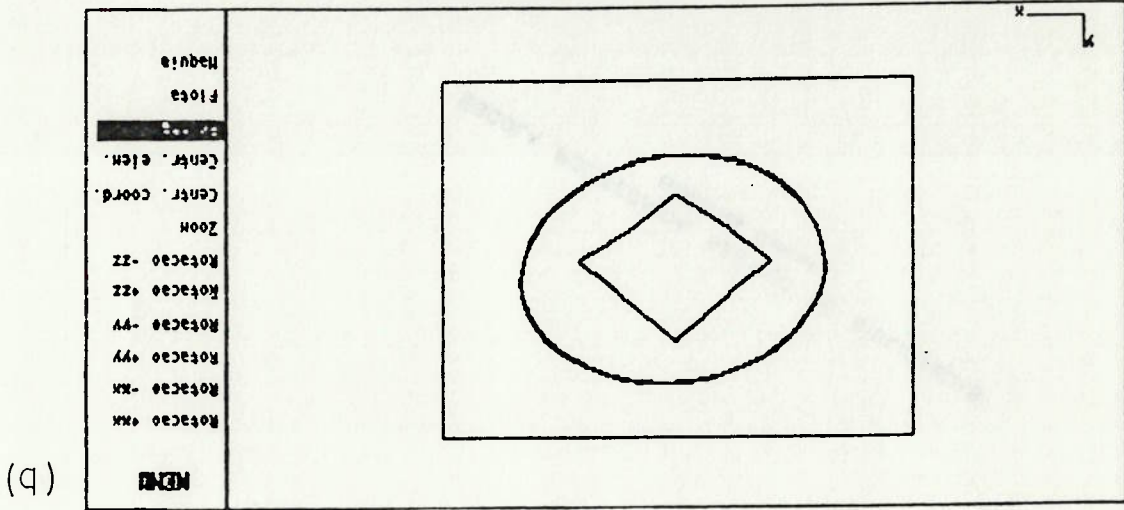


(d)

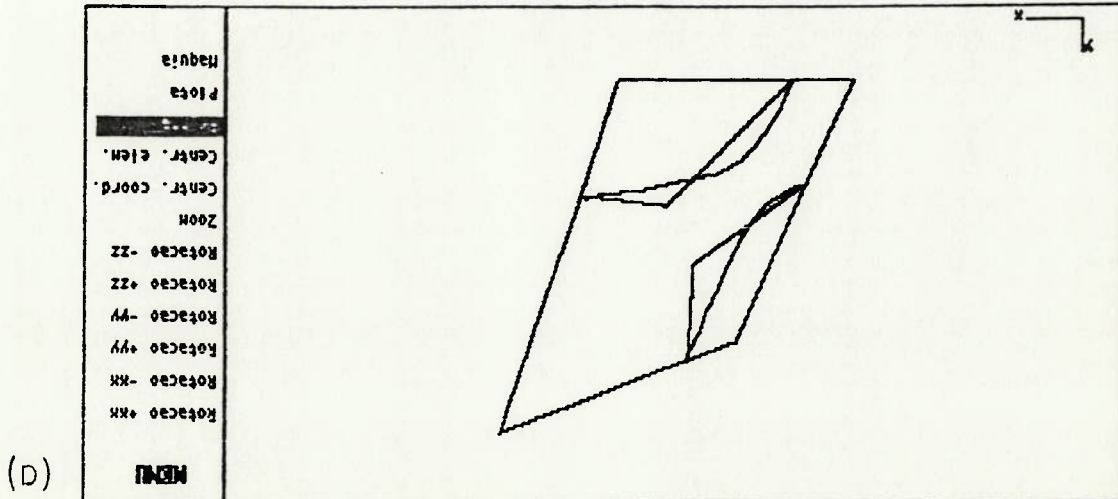
Comparação sobre um mesmo elemento, das curvas obtidas usando os dois algoritmos vistos no trabalho.

FIGURA 5.15

Título: COMPARAÇÃO : CONTÍNUO X DIVISÃO EM TRIÂNGULOS



Título: COMPARAÇÃO : CONTÍNUO X DIVISÃO EM TRIÂNGULOS



[COD] é a nova matriz de coordenadas nodais que deve ser utilizada para a construção de perspectivas analíticas. Os nós da estrutura neste caso podem ser ligados, obedecendo a tabela de incidência.

Durante a fase de pós processamento, [CO] passa por uma série de transformações tal que [CO] sempre permanece atualizada-

$$[CO]^{NX4} + [DEF]^{NX4} = [COD]^{NX4} \quad (5.12)$$

mento nodais:

mandando-se à matriz [CO] uma outra matriz de vetores de deslocamento nodais, a estrutura deformada precisa ser visualizada. Nestes instantes, a estrutura deformada pode ser construída sob a análise é que a estrutura deformada precisa ser visualizada. Apenas em capítulos anteriores). Apenas em determinadas etapas magões geométricas e remoção de linhas invisíveis (técnicas malha incluem construção de perspectivas analíticas, transformação geométrica e carregamentos. As técnicas para visualização da sua geometria e deformada, analisando a correção de trabalho com a estrutura deformada, analisando a correção de Durante a fase de pós processamento, em geral os analistas

$$[CO]^{NX4} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \dots & \dots & \dots & \dots \\ x_N & y_N & z_N & 1 \end{bmatrix} \quad (5.11)$$

indicado abaixo:

deformada possam ser colocadas em uma matriz [CO]^{Nx4}, como efetuada se compararmos com a representação dos níveis de tensão. Suponha que as coordenadas dos pontos da estrutura não A representação da estrutura é relativamente fácil de ser

Um vetor é um ente matemático que possui módulo, direção e sentido. Uma translação não afeta nenhuma dessas grandezas.

A figura 5.16 mostra uma roda deformada.

Resumindo: As transformações a que deve ser submetida [DEF] devem ser iguais à [Transf], excluindo-se as translações. Isto significa que [DEF] deve ser transformada por uma matriz correspondente à uma rotação pura, em torno da origem.

Embora a resposta esteja numericamente correta, desde que tenhamos o cuidado de certificarmos de que a quarta coluna da matriz [DEF] seja nula (trata-se de uma matriz de vetores), tanto do ponto de vista conceitual como do da eficiência computacional, ela está errada. Do ponto de vista conceitual, está errada porque embora possamos rotacionar um vetor, mudando sua direção, não podemos *transladá-lo*. Do ponto de vista da eficiência computacional está errado, porque qualquer translação efetuada sobre uma matriz onde a quarta coluna é nula, o efeito surtido é nenhum (ver capítulo 2). Ou seja, centenas ou milhares de produtos e somas são gastos inutilmente.

$$[T1][T2]..[TN] = [Transf]$$

A resposta intuitiva indica que elas devem ser tais que em seu conjunto se igualem à [Transf]:

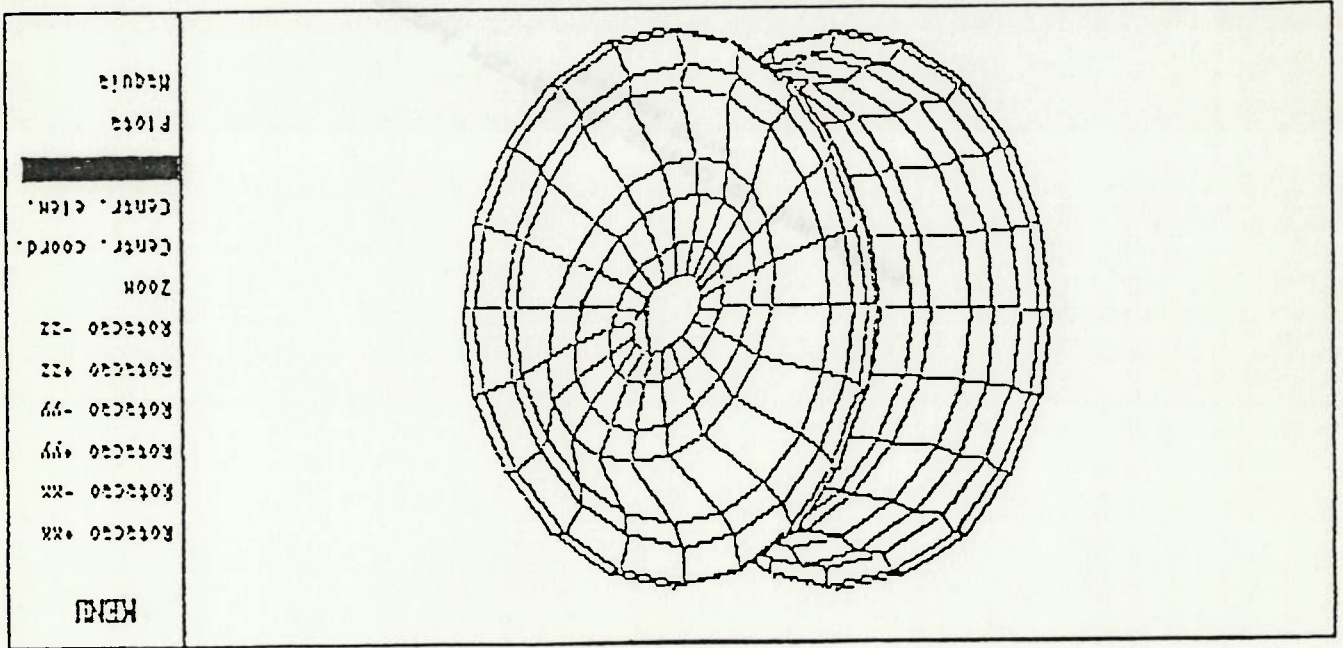
Pelo contrário, [DEF] só é atualizado nos momentos em que deseja-se a visualização da estrutura deformada. Assim surge naturalmente a questão: Quais as transformações a que ele deve ser submetido para acompanhar as transformações da matriz [CO] ?

Onde [Transf] é qualquer seqüência de operações geométricas

$$[CO] \leftarrow [CO].[Transf]$$

Figura 5.16 | Roda deformada.

Título: FADIGA DE CUBO - ESTRUTURA DEFORMADA



Vimos neste capítulo os principais procedimentos a serem utilizados no pós processamento de dados para o método dos elementos finitos. As conclusões tiradas a respeito do trágado de curvas isovalor (em especial sobre a aplicação do algoritmo da divisão por triângulos), e de quais são realmente as operações que devem ser aplicadas à matriz de deformações, forneceram subsídios para a implementação de rotinas no programa NAVGRAF.

Mostramos os principais pontos da ampla área de processamento gráfico para o método dos elementos finitos. Os resultados alcançados com a implementação das idéias aqui apresentadas no programa NAVGRAF foram satisfatórios. Eles mostram que um bom pré/pós processador gráfico, adequado às reais necessidades dos usuários reduz, em muito, o tempo gasto numa análise estrutural. Podemos até afirmar que sem um sistema pré/pós processador qualquer, a tarefa da análise de certas estruturas espaciais ficaria completamente inviabilizada.

Em termos do projeto de sistemas pré e pós processadores de dados, podemos concluir que a interatividade é importante, não podendo ser deixada de lado, de maneira alguma. Muitos sistemas comerciais, de certo renome, parecem não ter se importado com isso. Nestes sistemas, dados em demasia (e de difícil elaboração mental) devem ser fornecidos, para que se execute uma simples perspectiva. O enfoque adotado tem sido, na maioria dos casos, a mudança da posição do observador. No programa NAVGRAF, o tratamento é outro, sendo adotado a mudança da posição do objeto, exclusivamente por meio de rotações. O resultado é um sistema ágil e, principalmente, de fácil uso. Parece-nos que a visualização tridimensional de uma estrutura por meio da mudança da posição do quadro de projeção é promissora, pois é o tratamento que exige menor quantidade de cálculos e de dados de entrada.

Quanto ao problema da remoção de linhas ocultas, podemos afirmar que no presente, um algoritmo suficientemente geral, capaz de resolver qualquer cena, é ainda muito lento para ser incluído num sistema interativo. No entanto, no NAVGRAF é também implementado um algoritmo veloz, baseado no

FECHAMENTO

conceito da ordenação por profundidade, que faz uso do fato de que as malhas de elementos são, em geral, constituídas por pequenas superfícies interconectadas.

Pensamos que um algoritmo adequado para o traçado de curvas de isovalor é aquele baseado na divisão em pequenos triângulos, uma vez que sua precisão é compatível com a atingida pelo próprio método dos elementos finitos. De nada adianta um algoritmo muito preciso para o traçado de curvas se os valores de base são pobres. A medida que a malha vai se refinando, as curvas obtidas vão convergindo para uma configuração aceitável. Tal algoritmo é usado pelo NAVGRAF com sucesso na análise de estruturas.

Alguns aspectos complementares à atividade de pré/pós processamento não foram tratados neste trabalho, mas sem dúvida mereceriam atenção. Assim, a numeração de nós e elementos, o encolhimento de elementos (shrink), a divisão em grupos, o aumento de determinadas regiões para uma melhor visualização (zoom), a criação e manipulação de menus interativos, etc, são atividades que o NAVGRAF executa, e que possivelmente serão tratados num artigo futuro.

Quanto à sugestão de prosseguimento da pesquisa, diríamos que a aplicação de várias cores é importante ser estudada, bem como a integração de sistemas gráficos com sistemas para geração automática de malhas para o método dos elementos finitos.

- [AHO] Aho, Alfred V., Hopcroft, John E., Ullman, Jeffrey D., Data Structures and Algorithms, Addison Wesley, Publishing Company, Reading Massachusetts, 1983
- [AKI] Akin, J.E., Gray, W.H., "Contouring on Isoparametric Surfaces", Int. J. Num. Meth. Engng, V.11, 1893-1897(1977).
- [ALE] Kawano, A., Augusto, O.B., "Desenvolvimento de recursos computacionais gráficos voltados à engenharia Naval e Oceânica", Relatório à FAPESP processo 88/0621-1, Fev 1989
- [BAR] Barlow, John, "Optimal Stress Locations in Finite Element Models", Int. J. Num. Meth. Engng, V.10, 243-251 (1976).
- [BRO] Bronson, Richard, Matrix Operations, Schaum's Outline Series, McGraw-Hill Company, New York, 1989
- [BUR] Burnett, David S., Finite Element Analysis, Addison-Wesley Publishing Company, Reading Massachusetts, 1987
- [GRA] Crandall, S.H., Engineering Analysis, McGraw-Hill, New York, 1956
- [DES] Landi, Apostila do curso de desenho, Escola Politécnica da Universidade de São Paulo.
- [EHR] Ehrlieh, Pierre Jacques, Pesquisa Operacional 5ª ed, São Paulo, Editora Atlas, 1985.

[FIL] Fillmore, Jay, A Note on Rotation Matrices, IEEE Computer Graphics and its applications, Vol 4, n.2, pp 30-33, Jan 1981.

[GRA] Gray, W.H., Akin, J.E., "An Improved Method for Contouring on Isoparametric Surfaces", Int. J. Num. Meth. Engng, V.14, 451-472 (1976).

[HIN] Hinton, E., Campbell, J.S., "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method", Int. J. Num. Meth. Engng, V.8, 461-480 (1976).

[KNOW] Knowles, N.C., "Finite Element Analysis", Computer Aided Design, Vol. 16, n.3, pp 134-140, May 1984.

[LIN] Dongarra, J.J. et alii, Linpack user's guide, Siam, Philadelphia, 1979

[LIP] Lipschitz, Seymour, Algebra linear, Coleção Schaum, McGraw-Hill do Brasil, São Paulo, 1972

[MAN] Manual de Utilização do programa NAVGRAF. Ceno-Departamento de Engenharia Naval da EPUSP (1991).

[MEE] Meek, J.L., Beer, G., "Contour Plotting of Data using Isoparametric Element Representation", Int. J. Num. Meth. Engng, V.10, 954-957 (1976).

[NEW] Newman, William M., Sproull, Robert F., Principles of computer graphics 2nd ed., McGraw-Hill, Auckland, 1981

[NUM] Scheid, Francis, Numerical Analysis 2º ed, Schaum's Outline Series, McGraw-Hill, 1988, New York.

[OSC] Plano de Pesquisa enviado à FAPESP relativo à bolsa de iniciação científica de Alexandre Kawano, Março 1988.

- [PAR] Park, Chan, Interactive Microcomputer Graphics, Addison-Wesley Publishing Company, Reading Massachusetts, 1985
- [RIS] Riesenfeld, Richard F., Homogeneous Coordinates and Projective Planes in Computer Graphics, IEEE Computer Graphics and its applications, Vol 1, n1, pp 50-56, Jan 1981.
- [ROG] Rogers, David F., Procedural elements for computer graphics, McGraw-Hill, New York, 1985
- [ROG2] Rogers, David F., Mathematical elements for computer graphics, McGraw-Hill, New York, 1976
- [RYA] Ryan, Patrick J., Euclidean and non-euclidean geometry, Cambridge University Press, Cambridge, 1986
- [SCH] Schulz, G. Biebig, Schulz, Ch, 3D Graphics in Pascal, John Wiley and sons, Chichester, 1990
- [SHU] Sutherland, Ivan E., Sproull, Robert F., Shumacker, R. A., " A characterization of ten hidden surface algorithms ", Computing Surveys, V.6, N.1, March 1974, ACM
- [STR] Strang, Gilbert, Linear Algebra and its applications, Academic Press, Inc, Orlando, 1980

APÊNDICES

1. ESPAÇO VETORIAL

Definição: Seja um conjunto não vazio com operações de adição e multiplicação por escalar que determinam para qualquer $u, v \in V$ uma soma $u+v \in V$ e para qualquer $u \in V, k \in \mathbb{R}$ um produto $k \cdot u \in V$.
 V é chamado espaço vetorial sobre \mathbb{R} (e os elementos de V são chamados vetores se os seguintes axiomas são verdadeiros:

A1) Para quaisquer vetores $u, v, w \in V, u+(v+w) = (u+v)+w$;

A2) Há um vetor em V denotado 0 , e chamado vetor nulo, para o qual $u+0=u$, para qualquer vetor $u \in V$.

A3) Para cada vetor $u \in V$ existe um outro em V denotado $-u$, para o qual $u+(-u)=0$.

A4) Para quaisquer vetores $u, v \in V, u+v = v+u$.

M1) Para qualquer escalar $k \in \mathbb{R}$ e quaisquer vetores $u, v \in V,$
 $k \cdot (u+v) = k \cdot u + k \cdot v$.

M2) Para quaisquer escalares $a, b \in \mathbb{R}$ e qualquer vetor $u \in V,$
 $(a+b) \cdot u = a \cdot u + b \cdot u$

M3) Para quaisquer escalares $a, b \in \mathbb{R}$ e qualquer vetor $u \in V,$
 $(a \cdot b) \cdot u = a \cdot (b \cdot u)$

M4) $1 \cdot u = u$ para qualquer $u \in V$.

- 1) $W = \{ (x, y, z) \in \mathbb{R}^3 \mid x+y=0 \}$ é subespaço de \mathbb{R}^3 pois:
- a) $(0, 0, 0) \in W$
- b) Sejam $u = (x_1, y_1, z_1), v = (x_2, y_2, z_2) \in W$
- $$\alpha \cdot u + \beta \cdot v = (\alpha \cdot x_1 + \beta \cdot x_2, \alpha \cdot y_1 + \beta \cdot y_2, \alpha \cdot z_1 + \beta \cdot z_2)$$

Exemplos:

- a) $0 \in W$
- b) $u, v \in W \rightarrow a \cdot v + b \cdot u \in W, \forall a, b \in \mathbb{R}$.
- Teorema: W é um subespaço de V se, e somente se,

Seja W um subconjunto de um espaço vetorial V sobre \mathbb{R} . W é chamado subespaço de V se W for também um espaço vetorial sobre \mathbb{R} em relação às operações de multiplicação de vetor por escalar e adição de vetores de V .

2. SUBESPAÇO

$k \cdot (a_1, a_2, \dots, a_n) = (k \cdot a_1, k \cdot a_2, \dots, k \cdot a_n) \quad \forall k \in \mathbb{R}$

e a de multiplicação por escalar:

$$(a_1, a_2, \dots, a_n) + (b_1, b_2, \dots, b_n) = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$$

for definida por:

1) O conjunto de todas as n-uplas de números reais, denominado \mathbb{R}^n , é um espaço vetorial sobre \mathbb{R} se a operação de adição de

desse conjunto é um vetor.

V é um espaço vetorial sobre \mathbb{R} em relação às operações usuais de multiplicação por uma constante e adição, e um elemento

$$a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \quad \text{com } a_i \in \mathbb{R}$$

1) Seja V o conjunto de todos os polinômios

Exemplos:

3. COMBINAÇÕES LINEARES, SUBESPAÇOS GERADOS

Como $\alpha \cdot x_1 + \beta \cdot x_2 + \alpha \cdot y_1 + \beta \cdot y_2 = \alpha \cdot (x_1 + y_1) + \beta \cdot (x_2 + y_2) = \alpha \cdot 0 + \beta \cdot 0 = 0$,
 $\alpha \cdot u + \beta \cdot v \in W$.
 Logo, W é um subespaço de \mathbb{R}^3

Seja V um espaço vetorial sobre \mathbb{R} e sejam $v_1, \dots, v_m \in V$.
 Qualquer vetor da forma

$$a_1 \cdot v_1 + a_2 \cdot v_2 + \dots + a_m \cdot v_m$$

onde $a_i \in \mathbb{R}$, é chamado uma combinação linear de v_1, \dots, v_m .

Seja S um subconjunto não vazio de V . O conjunto $L(S)$ de todas as combinações lineares dos vetores de S , é um subespaço de V contendo S . Além disso, se W qualquer outro subespaço de V contendo S , então $L(S) \subseteq W$. $L(S)$ é nessas condições o menor subespaço contendo S . Dizemos que $L(S)$ é o subespaço gerado por S .

Exemplos:

1) Sejam os vetores $u=1, v=t, w=t^2$ de um espaço vetorial V . O vetor $3 \cdot u - 4 \cdot v + 10 \cdot w = 3 - 4 \cdot t + 10 \cdot t^2$ é uma combinação linear de u, v, w .

2) Os vetores $u=(1,0,0), v=(0,1,0), w=(0,0,1)$ geram o espaço vetorial \mathbb{R}^3 pois qualquer vetor (a,b,c) pode ser escrito como combinação linear de u, v, w .

3) Para determinar se um dado vetor é ou não uma combinação linear de outros vetores basta calcularmos o determinante formado pelo conjunto dos vetores, onde cada vetor passa a ser uma linha do determinante:

$u=(1,2,3)$ é combinação linear de $(1,0,2)$ e $(0,2,-1)$ pois

$$\begin{vmatrix} 1 & 2 & 3 \\ 1 & 0 & 2 \\ 0 & 2 & -1 \end{vmatrix} = 0$$

4. DEPENDENCIA LINEAR

Definição: Seja V um espaço vetorial sobre \mathbb{R} . Diz-se que os vetores $v_1, \dots, v_m \in V$ são linearmente independentes sobre \mathbb{R} , ou simplesmente dependentes, se existem escalares $a_1, \dots, a_m \in \mathbb{R}$, nem todos simultaneamente nulos tais que

$$a_1 \cdot v_1 + a_2 \cdot v_2 + \dots + a_m \cdot v_m = 0$$

Caso contrário, diz-se que os vetores são linearmente dependentes.

Teorema: Os vetores não nulos v_1, \dots, v_m são linearmente dependentes se, e somente se, um deles for combinação linear dos demais.

5. Bases e Dimensão

Definição: Diz-se que um espaço vetorial V é de dimensão finita n ou é n -dimensional, e escreve-se $\dim V = n$, se existem vetores linearmente independentes e_1, e_2, \dots, e_n que geram V . A sequência (e_1, e_2, \dots, e_n) é então chamada uma base de V .

Teorema: Seja V um espaço vetorial de dimensão finita. Então todas as bases de V tem o mesmo número de elementos.

Seja $\{e_1, e_2, \dots, e_n\}$ uma base do espaço vetorial V sobre \mathbb{R} e seja u um vetor qualquer em V . Assim, como a base $\{e_i\}$ gera V , podemos escrever $v = a_1 \cdot e_1 + a_2 \cdot e_2 + \dots + a_n \cdot e_n$, $a_i \in \mathbb{R}$. Os escalares a_i são as coordenadas de v em relação à base $\{e_i\}$, e a n -upla (a_1, a_2, \dots, a_n) é chamada de vetor coordenada de v em relação à base $\{e_i\}$. Representamos v por suas

6. COORDENADAS

11) Sejam U e W os subespaços de \mathbb{R}^3 gerados por $\{(1, 0, 0), (0, 1, 1), (0, 1, 0)\}$ e $\{(1, 2, 0)\}$ respectivamente. Assim, $\dim U = 3$ e $\dim W = 1$. Logo, $\dim(U \cup W) = 3 + 1 - 3 = 1$.

1) Seja W o espaço vetorial dos polinômios de grau $\leq n$. O conjunto $\{1, t, t^2, \dots, t^n\}$ é linearmente independente de gera W . Assim, ele é uma base de W . Logo, $\dim W = n+1$.
 11) O conjunto $W = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ é linearmente independente e gera \mathbb{R}^3 .

Exemplos:

Teorema: Sejam U e W subespaços de dimensão finita de um espaço vetorial V . Então, $U+W$ tem dimensão finita e em particular, $\dim(U+W) = \dim U + \dim V - \dim(U \cap W)$.

Definição: Sejam U e W dois subespaços de um espaço vetorial V . A soma de U com W , escrita $U+W$, consiste de todas as somas $u+w$, onde $u \in U$ e $w \in W$:
 $U + W = \{u+w \mid u \in U, w \in W\}$
 $U+W$ é também um subespaço vetorial e é gerado pelos vetores de U e de W .

Definição: Uma transformação linear $F: V \rightarrow U$ é dita singular se existe um vetor $v \in V$, não nulo tal que $F(v) = 0$.

Definição: Uma transformação linear $F: V \rightarrow U$ é isomorfismo se ela é injetora. Diz-se que os espaços vetoriais V e U são isomorfos se existe um isomorfismo de V sobre U .

Definição: Sejam U e V espaços vetoriais sobre \mathbb{R} . Uma transformação $F: U \rightarrow V$ é chamada transformação linear se satisfaz as seguintes condições:
 a) Para qualquer $v, w \in V$, $F(v+w) = F(v) + F(w)$
 b) Para qualquer $v \in V$ e $k \in \mathbb{R}$, $F(k \cdot v) = k \cdot F(v)$

7. TRANSFORMAÇÕES LINEARES

$[v]_f = [3 \ -2 \ -5]_f$ (vetor coordenada de v em relação à base f)
 $\alpha = 3, \beta = -2, \gamma = -5$, ou seja,

Resolvendo-o temos:

$$\begin{aligned} 1 \cdot \alpha + 0 \cdot \beta + 0 \cdot \gamma &= 3 \\ 1 \cdot \alpha + 1 \cdot \beta + 0 \cdot \gamma &= 1 \\ 1 \cdot \alpha + 1 \cdot \beta + 1 \cdot \gamma &= -4 \end{aligned}$$

montar o sistema:

$$\begin{pmatrix} 3 \\ 1 \\ -4 \end{pmatrix} = \alpha \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \beta \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \gamma \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \text{ ou seja, podemos}$$

1) Consideremos os espaço vetorial \mathbb{R}^3 . Vamos encontrar as coordenadas de $v = (3, 1, -4)$ relativo à base $f^1 = (1, 1, 1), f^2 = (0, 1, 1), f^3 = (0, 0, 1)$:

Exemplos:

$$[v] = (a_1, a_2, \dots, a_n)$$

coordenadas em relação à uma base por:

Teorema: Uma transformação linear $F: V \rightarrow U$ é isomorfismo, se, e somente se, ela é não singular.

Exemplos:

1) Seja $F: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ a transformação $F(x, y, z) = (x+z, y+z, z)$. F é transformação linear porque:

$$\left\{ \begin{array}{l} F(u+v) = F(u) + F(v) \\ F(k \cdot u) = k \cdot F(u), \quad k \in \mathbb{R} \end{array} \right.$$

11) Seja V o espaço vetorial dos polinômios na variável t sobre \mathbb{R} . Então a transformação derivada que leva um polinômio à sua

derivada é linear:

$$D: V \rightarrow V, \quad D(P(t)) = \frac{dP}{dt} \text{ é linear pois:}$$

$$\left\{ \begin{array}{l} D(P_1 + P_2) = D(P_1) + D(P_2) \\ D(k \cdot P_1) = k \cdot D(P_1), \quad k \in \mathbb{R} \end{array} \right.$$

iii) Vejamos intuitivamente porque para que uma transformação linear $F: V \rightarrow V$ seja singular, deve existir um vetor $a \in V$ tal que $F(a) = 0$: Se a transformação é singular (ou não um isomorfismo), devemos ter para algum par $v, u \in V$ (u e v distintos entre si) tal que $F(u) = F(v)$, ou $F(u) - F(v) = 0$. Mas como a transformação F é linear, $F(u-v) = 0$. Como u e v são distintos, $u-v = a \neq 0$. Logo, existe de fato um $a \in V$ tal que $F(a) = 0$.

8. TRANSFORMAÇÕES LINEARES INVERSÍVEIS

Definição: Um operador linear $T: V \rightarrow V$ é inversível se tem inverso, isto é, se existe $T^{-1} \in \mathcal{L}(V)$ tal que $TT^{-1} = I$. $\mathcal{L}(V)$ é o espaço vetorial de todas as transformações lineares de V sobre V .

Teorema: Um operador linear $T: V \rightarrow V$ num espaço vetorial de dimensão finita é invertível se, e somente se, é não singular.

9. REPRESENTAÇÃO MATRICIAL DE UMA TRANSFORMAÇÃO LINEAR

Seja T um operador linear num espaço vetorial V sobre \mathbb{R} e suponhamos que $\{e_1, e_2, \dots, e_n\}$ seja uma base de V . Então, $T(e_1), T(e_2), \dots, T(e_n)$ são vetores de V . Logo, podemos escrever:

$$\begin{aligned}
 T(e_1) &= a_{11} \cdot e_1 + a_{12} \cdot e_2 + \dots + a_{1n} \cdot e_n \\
 T(e_2) &= a_{21} \cdot e_1 + a_{22} \cdot e_2 + \dots + a_{2n} \cdot e_n \\
 &\dots \\
 T(e_n) &= a_{n1} \cdot e_1 + a_{n2} \cdot e_2 + \dots + a_{nn} \cdot e_n
 \end{aligned}$$

Isso motiva:

Definição: A matriz acima dos coeficientes, denotada por $[T]$, ou simplesmente $[T]$, é chamada matriz de transformação e T em relação à base $\{e_i\}$ ou simplesmente, matriz de T na base $\{e_i\}$.

Teorema: Seja $\{e_1, e_2, \dots, e_n\}$ base de V e seja T qualquer operador em V . Então para qualquer vetor $v \in V$,

$$[T(v)] = [v] \cdot [T]$$

Ou seja, se multiplicarmos o vetor coordenada de v em relação à base $\{e_i\}$ pela representação matricial de T , obteremos o vetor coordenada de $T(v)$!

Exemplos:

1)

Seja uma transformação linear $F: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ com

$$F(x, y, z) = (x+2z, y+4z, z), \text{ e uma base } \{e_i\} = \{(1, 1, 0), (0, 1, 0), (1, 1, 1)\}$$

A matriz de transformação $[T]$ de T na base $\{e_i\}$ pode ser

obtida da seguinte maneira:

$$[M] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Então a matriz $[M]$ dos coeficientes acima, é chamada matriz de transição da base "velha" $\{e_i\}$ para a base "nova" $\{f_i\}$.

$$\begin{aligned} f_1 &= a_{11} \cdot e_1 + a_{12} \cdot e_2 + \dots + a_{1n} \cdot e_n \\ f_2 &= a_{21} \cdot e_1 + a_{22} \cdot e_2 + \dots + a_{2n} \cdot e_n \\ &\dots \\ f_n &= a_{n1} \cdot e_1 + a_{n2} \cdot e_2 + \dots + a_{nn} \cdot e_n \end{aligned}$$

Definição: Sejam $\{e_1, e_2, \dots, e_n\}$ e $\{f_1, f_2, \dots, f_n\}$ duas bases de V Suponhamos que

a base ?

Responderemos agora a pergunta natural: Como muda a representação matricial de uma transformação $T: V \rightarrow V$ quando mudamos

10. MUDANÇA DE BASE

$$[V] [T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [2 \ 8 \ 1] = [T(V)] ;$$

Agora a verificação:

$$v = 0 \cdot e_1 + 6 \cdot e_2 + e_3. \text{ Isto é, } [v] = (0, 6, 1).$$

$$[T(v)] = 2 \cdot e_1 + 8 \cdot e_2 + 1 \cdot e_3 = (2, 8, 1).$$

$$T(v) = (1+2 \times 1, 7+1 \times 6, 1) = (3, 11, 1).$$

$$\text{Seja } v = (1, 7, 1).$$

são dados no exemplo anterior.

11) Verifiquemos se realmente $[V] [T] = [T(V)]$, onde os dados

$$\left\{ \begin{aligned} f_1 &= (1, 1, 0) = e_1 \\ f_0 &= (0, 1, 0) = e_2 \\ f_1 &= (3, 5, 1) = 2 \cdot e_1 + 2 \cdot e_2 + e_3 \end{aligned} \right. \rightarrow [T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

Exemplo:

1) Vamos determinar a matriz de transição da base $\{e_i\}$ para a base $\{f_i\}$ onde $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$, $e_3 = (0, 0, 1)$; $f_1 = (\cos\theta, \sin\theta, 0)$, $f_2 = (-\sin\theta, \cos\theta, 0)$, $f_3 = (0, 0, 1)$. ver figura abaixo.

$$\begin{aligned} f_1 &= \cos\theta \cdot e_1 + \sin\theta \cdot e_2 + 0 \cdot e_3 \\ f_2 &= -\sin\theta \cdot e_1 + \cos\theta \cdot e_2 + 0 \cdot e_3 \\ f_3 &= 0 \cdot e_1 + 0 \cdot e_2 + 1 \cdot e_3 \end{aligned} \quad \rightarrow [M] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Teorema: Seja $[M]$ a matriz de transição da base $\{e_i\}$ para a base $\{f_i\}$ num espaço vetorial V . Então para qualquer vetor $v \in V$, $[v]_f = [v]_e \cdot [M]^{-1}$

Exemplo:

1) Sejam $\{e_i\}$ e $\{f_i\}$ como no exemplo anterior. Tomemos um vetor $[v] = (5, 0, 0)$. O vetor v escrito em função da base $\{f_i\}$ fica:

$$[v]_f = \begin{bmatrix} 5 & 0 & 0 \\ \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = [5 \cdot \cos\theta \quad -5 \cdot \sin\theta \quad 0]_f$$

Figura 2.

Teorema: Seja $[M]$ a matriz de transição da base $\{e_i\}$ para a base $\{f_i\}$ num espaço vetorial V . Então, para qualquer operador linear $[T]$ em V , $[T]_f = [T]_e \cdot [M]^{-1}$

Exemplo:

1) Seja a matriz de transformação $[T] =$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

De uma transformação linear $T: V \rightarrow V$ em relação à base $\{e_i\}$ do exemplo anterior. Vamos escrever essa mesma transformação, mas em relação à base $\{f_i\}$, também do exemplo anterior (vamos assumir $\theta = 45^\circ$):

$$[T]_f = [M][T][M]^{-1} = \begin{bmatrix} 1 & 0 & 2\sqrt{2} \\ 0 & 1 & \sqrt{2} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{[Esse resultado já era previsto. Ver capítulo sobre transformações]}$$

1. Introdução

Sabe-se que, em certos pontos no interior de um elemento, as tensões obtidas são melhores que em qualquer outro lugar do seu interior, no sentido da convergência para um valor estável. Esses pontos são denominados pontos superconvergentes [BUR].

Neste apêndice mostraremos uma maneira de como podemos extrapolar os valores das tensões obtidas nos pontos superconvergentes para os nós de um elemento. Recomendamos aos leitores mais interessados as referências [BUR], [HIN] e [BAR].

2. A precisão dos valores dos deslocamentos é maior do que a das tensões.

Para explicar porque os deslocamentos resultam mais precisos do que as tensões, suponhamos que um campo de deslocamentos U exato (solução da equação diferencial + condições de contorno regentes) seja aproximado por uma polinômio \tilde{U} de grau n completo, isto é, as funções de forma no método dos elementos finitos formam uma família de polinômio que é completa até o grau n . No caso bidimensional:

¹ Termos de grau superior a n de uma família incompleta são permitidos, mas a precisão não será melhorada por causa deles, a não ser ocasionalmente.

Existem certas formulações de certos elementos onde aparecem derivadas de ordem maior que um.

Existem algumas localidades (ξ, η) no interior do elemento tais que:

3. Pontos no interior do elemento que propiciam precisão superior se comparados com qualquer outro ponto do elemento.

Entretanto, existem pontos no interior do elemento que exibem precisão para as tensões igual a atingida para os deslocamentos. É o que veremos a seguir.

Vemos, portanto, que a precisão atingida para as tensões é grau pior que a aproximação do campo de deslocamentos.

$$T(\xi, \eta) - \tilde{T}(\xi, \eta) = O(n) \quad (B.4)$$

No caso das tensões, a ordem da aproximação deve ser pior, já que tensões envolvem derivadas do campo de deslocamentos. Assim, em geral, se o campo de deslocamentos aproximado \tilde{U} tem ordem n , o campo de tensões \tilde{T} deverá ter ordem $(n-1)^{(2)}$. Desta maneira, sendo $T(\xi, \eta)$ o campo exato,

$$U(\xi, \eta) - \tilde{U}(\xi, \eta) = O(n+1) \quad (B.3)$$

Podemos aceitar que, quando o tamanho dos elementos tende a zero, a aproximação de U por \tilde{U} é a melhor possível, e daí,

$$U(\xi, \eta) = d_0 + d_{11}\xi + d_{12}\eta + d_{21}\xi^2 + d_{22}\xi\eta + d_{23}\eta^2 + \dots \quad (B.2)$$

Vamos supor agora que o campo de tensão exato possa ser expandido em uma série de Taylor:

$$\tilde{U}(\xi, \eta) = c_0 + c_{11}\xi + c_{12}\eta + c_{21}\xi^2 + c_{22}\xi\eta + c_{23}\eta^2 + \dots + c_n^n \quad (B.1)$$

³ Pontos de Gauss

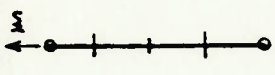
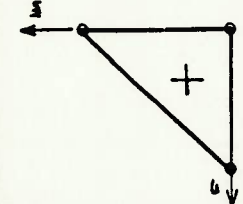
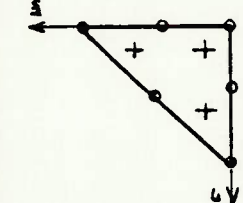
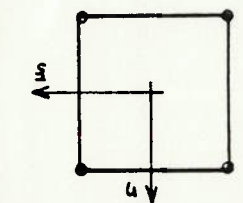
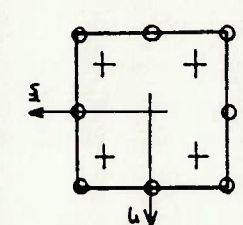
Isto é, existem pontos no interior do elemento para os quais a precisão alcançada é a mesma da do campo de deslocamentos. Este resultado vem da teoria dos elementos finitos, que afirma que o método, ao buscar o mínimo para a energia potencial total de um sistema, ele faz indiretamente um ajuste por mínimos quadrados de campo de tensões aproximado pelo exato (solução da equação diferencial + condições de contorno regulares do fenómeno) [BUR], [HIN]. Neste trabalho, apenas mostramos, na figura B.1, as localizações de tais pontos superconvergentes³. Para os leitores mais interessados, recomendamos a leitura da ref. [BAR] onde é mostrada uma técnica para se determinar tais pontos, não do ponto de vista matemático, mas sim do ponto de vista físico.

(Quando o maior lado do elemento tende a zero.

$$T(\xi', \eta') - T(\xi'', \eta'') = O(n+1)$$

(B.5)

FIGURA B.1
Localização de pontos superconvergentes para alguns tipos de elementos.

Função	Localização	Ilustração
C ³ cúbico	$\xi_1 = -\frac{\sqrt{3}}{3}$ $\xi_2 = \frac{\sqrt{3}}{3}$ $\eta_1 = \frac{\sqrt{3}}{3}$ $\eta_2 = \frac{\sqrt{3}}{3}$	
C ⁰ linear	$\xi_1 = \frac{1}{3}$ $\xi_2 = \frac{2}{3}$ $\eta_1 = \frac{1}{3}$ $\eta_2 = \frac{2}{3}$	
C ⁰ quadrático	$\xi_1 = \frac{1}{6}$ $\xi_2 = \frac{2}{3}$ $\xi_3 = \frac{5}{6}$ $\eta_1 = \frac{1}{6}$ $\eta_2 = \frac{1}{3}$ $\eta_3 = \frac{5}{6}$	
C ⁰ bilinear	$\xi_1 = 0$ $\xi_2 = 0$ $\eta_1 = 0$ $\eta_2 = 0$	
C ⁰ quadrático	$\xi_1 = -\frac{\sqrt{3}}{3}$ $\xi_2 = -\frac{\sqrt{3}}{3}$ $\xi_3 = +\frac{\sqrt{3}}{3}$ $\xi_4 = +\frac{\sqrt{3}}{3}$ $\eta_1 = -\frac{\sqrt{3}}{3}$ $\eta_2 = -\frac{\sqrt{3}}{3}$ $\eta_3 = +\frac{\sqrt{3}}{3}$ $\eta_4 = +\frac{\sqrt{3}}{3}$	

A existência de tais pontos pode ser utilizada de forma útil no mapeamento de tensões no modelo por elementos finitos, pelo simples fato de ser possível, através desses pontos, a recuperação da precisão perdida. Devemos buscar agora uma maneira de extrapolar tais valores para os nós de maneira segura.

4. Extrapolação dos valores de tensão obtidos nos pontos superconvergentes para os nós.

Existem basicamente duas maneiras de obtermos valores para as tensões nos nós do modelo, a partir dos valores calculados nos pontos superconvergentes. A primeira se aproveita de todo o domínio do modelo, criando uma única função que se ajusta, usando mínimos quadrados, aos valores das tensões nos pontos de extrapolação ótima de informações. A segunda usa apenas os valores no interior de cada elemento isoladamente, e somente numa fase posterior é que é tirada a média dos valores nodais. A referência [HIN] indica que ambos tratamentos conduzem a resultados próximos, embora o segundo seja mais rápido e eficiente. Mostraremos apenas o segundo tratamento.

Imaginemos que possuímos um elemento de N_n nós que possui em seu interior N_s pontos de extrapolação ótima de tensão. Em termos matriciais (para simplificar, vamos nos fixar em apenas uma componente de tensão, por exemplo, σ^x):

$$\left\{ \begin{matrix} \sigma^x(\xi_1, \eta_1) \\ \vdots \\ \sigma^x(\xi_{N_s}, \eta_{N_s}) \end{matrix} \right\} = \left\{ \sigma^x \right\}_{optm} = \left\{ \begin{matrix} \sigma^x(\xi_1, \eta_1) \\ \vdots \\ \sigma^x(\xi_{N_s}, \eta_{N_s}) \end{matrix} \right\} \quad (B.6)$$

O que procuraremos é determinar uma matriz de transformação [TE] que leve $\left\{ \sigma^x \right\}_{optm}$ a $\left\{ \sigma^x \right\}_{nós}$, ou seja, [TE] é tal que:

⁴ Pontos superconvergentes

A primeira etapa na obtenção de [TE] é a seleção da função $\phi(\xi, \eta)$ que deverá aproximar $\langle \sigma \rangle_{\text{olm}}^x$. Temos duas alternativas. A primeira seria escolher $\phi(\xi, \eta)$ tal que ele interpole os pontos de extragação ótima de tensão (neste caso, a aproximação por mínimos quadrados seria ótima). A segunda alternativa seria usar um polinômio de grau menor, de modo que $\phi(\xi, \eta)$ seja de fato uma aproximação. Ambos tratamentos fornecem resultados numéricos satisfatórios quando o número de pontos de Gauss ⁴ é baixo. Entretanto, quando seu número aumenta, a função que interpola os pontos de extragação ótima começa a apresentar um comportamento oscilatório indesejável, característico de funções interpoladoras de grau elevado. Veremos que mesmo o uso de uma função $\phi(\xi, \eta)$ de grau baixo fornece resultados muito bons (isto é, o uso de funções interpoladoras de grau elevado parecem ser desnecessárias). Neste trabalho, vamos escolher :

$$(B.8) \quad \langle \sigma \rangle_{\text{nós}}^x = \begin{Bmatrix} \sigma_{\text{nx}}^x(-1, -1) \\ \sigma_{\text{nx}}^x(+1, -1) \\ \vdots \\ \sigma_{\text{nx}}^x(-1, 0) \end{Bmatrix}_{8 \times 1} = \langle \sigma \rangle_{\text{olm}}^x = \begin{Bmatrix} \sigma_{\text{olmx}}^x(-\frac{1}{4}, -\frac{1}{4}) \\ \sigma_{\text{olmx}}^x(-\frac{1}{4}, +\frac{1}{4}) \\ \vdots \\ \sigma_{\text{olmx}}^x(+\frac{1}{4}, -\frac{1}{4}) \\ \sigma_{\text{olmx}}^x(+\frac{1}{4}, +\frac{1}{4}) \end{Bmatrix}_{4 \times 1}$$

$\langle \sigma \rangle_{\text{nós}}^x$:
de 8 nós (Classe C⁰). Neste caso, temos para $\langle \sigma \rangle_{\text{olm}}^x$ e vamos exemplificar com elementos quadriláteros isoparamétricos. Ao invés de continuarmos com essa abstração matemática,

as coordenadas (ξ, η) nodais são obviamente conhecidas. sido determinada, $\langle \sigma \rangle_{\text{nós}}^x$ pode ser facilmente calculado já que quadrados, aos valores discretos de $\langle \sigma \rangle_{\text{olm}}^x$. Após a função ter uma função $\phi(\xi, \eta)$ que deverá se ajustar, no sentido dos mínimos A ideia por trás da obtenção de [TE] está na escolha de

$$(B.7) \quad \langle \sigma \rangle_{\text{nós}}^x = [TE] \cdot \langle \sigma \rangle_{\text{olm}}^x$$

$$\langle c \rangle = 0.25 \begin{bmatrix} 1 & \sqrt{3} & \sqrt{3} & \sqrt{3} \\ 1 & \sqrt{3} & \sqrt{3} & \sqrt{3} \\ 1 & 1 & 1 & 1 \\ 1 & \sqrt{3} & \sqrt{3} & \sqrt{3} \end{bmatrix} \langle \sigma \rangle_{\text{olm}} \quad (\text{B.13})$$

Substituindo valores, chegamos a:

4	-	$\frac{\sqrt{3}}{1}$	+	$\frac{\sqrt{3}}{1}$	+	$\frac{\sqrt{3}}{1}$	
3	+	$\frac{\sqrt{3}}{1}$	+	$\frac{\sqrt{3}}{1}$	+	$\frac{\sqrt{3}}{1}$	
2	+	$\frac{\sqrt{3}}{1}$	-	$\frac{\sqrt{3}}{1}$	-	$\frac{\sqrt{3}}{1}$	
1	-	$\frac{\sqrt{3}}{1}$	-	$\frac{\sqrt{3}}{1}$	-	$\frac{\sqrt{3}}{1}$	
1		ξ		η			

$$[Q] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \xi^1 & \xi^2 & \xi^3 & \xi^4 \\ \eta^1 & \eta^2 & \eta^3 & \eta^4 \end{bmatrix} \quad \langle c \rangle = \begin{Bmatrix} c^1 \\ c^2 \\ c^3 \\ c^4 \end{Bmatrix}$$

$$[P] = \begin{bmatrix} \sum_{i=1}^4 1 & \sum_{i=1}^4 \xi^i & \sum_{i=1}^4 \eta^i \\ \sum_{i=1}^4 \xi^i & \sum_{i=1}^4 \xi^{2i} & \sum_{i=1}^4 \xi^i \eta^i \\ \sum_{i=1}^4 \eta^i & \sum_{i=1}^4 \xi^i \eta^i & \sum_{i=1}^4 \eta^{2i} \end{bmatrix}$$

Onde:

$$\langle c \rangle = [P]^{-1} [Q] \langle \sigma \rangle_{\text{olm}} \quad (\text{B.11})$$

Minimizando em relação a c^1, c^2 e c^3 chegamos ao sistema:

$$e(c^1, c^2, c^3) = \sum_{i=1}^4 (\sigma(\xi^i, \eta^i) - \sigma_{\text{olm}}(\xi^i, \eta^i))^2 \quad (\text{B.10})$$

O erro nos pontos de Gauss cometido pela aproximação é:

$$\sigma(\xi, \eta) = c^1 + c^2 \xi + c^3 \eta \quad (\text{B.9})$$

Antes de fecharmos este apêndice, devemos salientar um fato importante. Do ponto de vista teórico, quando o tamanho do elemento tende a zero, os valores para as tensões nos pontos de Gauss recuperam a precisão da aproximação perdida, como foi mencionado acima. Na prática, o tamanho dos elementos não podem ser reduzidos a "zero". Isto é, a confiabilidade para os níveis de tensão que são obtidos pelos programas de elementos finitos sempre são menores do que a confiabilidade para os valores dos deslocamentos.

Após as tensões terem sido obtidas no nível de elemento, podemos tirar a média dos valores nos nós. Isso pode ser feito aplicando-se uma simples média aritmética.

O procedimento acima pode ser aplicado para qualquer tipo de elemento para a obtenção de valores nodais. Em alguns casos, isso é muito simples, como no caso de elementos lineares que possuem apenas um ponto de extração ótima de tensões [BAR].

$$[TE] = 0.25 \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & +1 & -1 & 0 & -1 & +1 & -1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \end{bmatrix}$$

$$\left[\begin{array}{c} \sqrt{3} \\ \sqrt{3} \\ \sqrt{3} \\ \sqrt{3} \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right] \left[\begin{array}{c} \sqrt{3} \\ \sqrt{3} \\ \sqrt{3} \\ \sqrt{3} \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right] \left[\begin{array}{c} \sqrt{3} \\ \sqrt{3} \\ \sqrt{3} \\ \sqrt{3} \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right]$$

(5.14)

Dai, podemos tirar as tensões nodais do elemento, já que:
 $\sigma(\xi, \eta) = [1 \ \xi \ \eta] \langle c \rangle$
 Finalmente podemos determinar [TE]: