AMBIENTE PARA GERAÇÃO DE MODELOS DE ELEMENTOS FINITOS EM MICROCOMPUTADORES

GERSON GOMES CUNHA

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS–GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊCIAS E ENGENHARIA CIVIL.

Aprovada por:

Prof. Nelson Favilla Ebecken, Dr.
(presidente)

Prof. Agustin Juan Ferrante, Phd.

Prof. Breno Pinheiro Jacob, Dr.

Eng. Alvaro Maia da Costa, Dr.

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 1991

CUNHA, GERSON GOMES
Ambiente para Geração de Modelos de Elementos Finitos em Microcomputadores, Rio de Janeiro, 1991.
XII,603 p. 29.7 cm (COPPE/UFRJ, M.Sc., Engenharia Civil, 1991)
Tese - Universidade Federal do Rio de Janeiro, COPPE.
1. Pré-processamento de Elementos Finitos
2. Computação Gráfica 3. CAD I. COPPE/UFRJ
II. Título (série)

A meus pais,
e a todos aqueles
que acreditaram
em mim.

AGRADECIMENTOS

Aos meus pais pelo apoio, incentivo e suporte financeiro durante toda a formação escolar, sem os quais esta etapa possivelmente nem teria se iniciado.

Aos meus tios Wandirley e Erny Marques pela acolhida familiar e incentivo no início do curso.

Aos colegas Paulo Cesar D'elia e Luis Fernando Melo pelo auxílio na instalação, configuração e apoio nos primeiros passos na utilização do editor de textos.

A Maria Tereza pelo valioso trabalho de datilografia e a Suzana Satamini pela minusiosa revisão da primeira versão do texto.

A Lino Rodrigues Moreira pelo apoio e incentivo recebidos, a troca de experiências nos momentos de discussão filosófica e as sugestões no desenvolvimento.

A Maria Célia Lopes pelo grande auxílio na digitação, revisão e correção do texto final, plotagem das figuras, diagramação e impreses preliminares e final da tese.

Ao CNPq pelo suporte financeiro no início do curso e a PETROBRÁS pelas facilidades computacionais na confecção do texto.

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

AMBIENTE PARA GERAÇÃO DE MODELOS DE ELEMENTOS FINITOS EM MICROCOMPUTADORES

Gerson Gomes Cunha

Abril de 1991

Orientador: Prof. Nelson Favilla Ebecken

Programa: Engenharia Civil

O presente trabalho consiste de um ambiente de pré e pós–processamento gráfico interativo para discretização de modelos estruturais, planos ou espaciais, em malhas de elementos finitos em microcomputadores.

Este trabalho vem contribuir na popularização do método dos elementos finitos, tornando acessíveis os sofisticados recursos de processamento numérico e gráfico para análise de estruturas ao pequeno usuário ou empresa.

O ambiente é integrado pois possibilita que projetistas e desenhistas utilizem a mesma plataforma gráfica. Recursos de computação gráfica são utilizados de modo a permitir a visualização do modelo e a interface interativa faz com que o sistema apresente respostas imediatas aos comandos do usuário. Todas estas características reunidas de uma maneira particularmente atrativa, possibilitam não só a discretização (geração da malha), mas também a edição, reordenação nodal e geração do arquivo de dados do modelo.

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

FINITE ELEMENT MODELS GENERATION
ENVIRONMENT IN MICROCOMPUTERS

Gerson Gomes Cunha

April, 1991

Thesis Supervisor: Prof. Nelson Favilla Ebecken

Department: Civil Engineering

The present work consists of an interactive graphics pre and post–processing environment for structural models discretization, plane and espatial, into finite element meshes in microcomputers.

This work comes to make a contribution in the popularization of finite elements method, allowing sofisticated resources of numeric and graphics processing for structural analysis to the small port user or company.

The environment is integrated as allows designers and draftmen to use the same graphic platform. Computer graphics resources are employed to make possible models vizualization and the interactive interface causes the system to give back immediate answers to users requests. All of these characteristics, gathered of a particularly attractive way, allows not only discretization (mesh generation), but also edition, nodal reordering and model database generation.

INDÍCE

Capítulo I 1

I.1. Introdução 1

Capítulo II 8

II.1. A Computação Gráfica 8

II.2. CAD – Projeto Assistido por Computador 12

II.3. O Método dos Elementos Finitos 17

II.4. Os Microcomputadores 25

Capítulo III 32

III.1. Geração de Malhas 33

III.1.1. Funções de Interpolação 34

III.1.2. Mapeamento 38

III.1.3. Mapeamento Laplaciano 40

III.1.4. Mapeamento Isoparamétrico 43

III.1.5. Mapeamento Transfinito 49

III.1.6. Mapeamento Transfinito Discreto 56

III.1.7. Triangularização 59

III.1.8. Modelagem Geométrica 66

III.2. Suavização 67

III.3. Diagonalização de Elementos Quadriláteros 74

Capítulo IV 78

IV.1. Histórico 78

IV.2. Generalidades 80

IV.3. Criação de Entidades 83

IV.4. Visualização do Desenho 86

IV.5. Edição do Desenho 88

IV.6. Operação do Programa 89

IV.7. Arquitetura 90

IV.8. Equipamento 92

IV.8.1 Equipamento Opcional 93

IV.9. Banco de Dados 95

IV.10. Customização 97

Capítulo V 102

V.1. O Sistema AutoMESH 102

V.2. Definição dos Elementos Estruturais 106

V.3. O Ambiente AutoMESH 109

V.4. Os Comandos do AutoMESH 121

V.4.1. O Comando Startup 123

V.4.2. O Comando Material 127

V.4.3. O Comando Node 149

V.4.4. O Comando Element 152

V.4.5. Comandos para Geração Automática 158

V.4.5.1. Geração Automática em Regiões de Formato Genérico 159

V.4.5.1.1. O Comando Isomapp 159

V.4.5.1.2. O Comando Double 173

V.4.5.1.3. O Comando Trilinear 179

V.4.5.1.4. O Comando Transit 186

V.4.5.2. Geração Automática em Regiões de Formato Específico 193

V.4.5.2.1. O Comando Gemat 194

V.4.5.2.2. O Comando Sphere 200

V.4.5.2.3. O Comando Domes 204

V.4.5.2.4. O Comando Tube 208

V.4.5.2.5. O Comando Torus 211

V.4.6. O Comando Restric 215

V.4.7. O Comando Load 220

V.4.8. Comandos de Edição 235

V.4.8.1. O Comando Merge 237

V.4.8.2. O Comando Breakel 246

V.4.8.3. O Comando Desloc 248

V.4.8.4. O Comando Delete 257

V.4.8.5. O Comando Copy 262

V.4.8.6. O Comando Scale 265

V.4.8.7. O Comando Rotate 271

V.4.8.8. O Comando Mirror 274

V.4.8.9. O Comando Array 277

V.4.9. Comandos Utilitários 282

V.4.9.1. O Comando Find 283

V.4.9.2. O Comando Numbering 285

V.4.9.3. O Comando Shrink 288

V.4.9.4. O Comando Face 289

V.4.9.5. O Comando Project 292
V.4.9.6. O Comando Sections 296
V.4.9.7. O Comando Display 302
V.4.9.8. O Comando Intert 304
V.4.9.9. O Comando Convert 306
V.4.9.10. O Comando Clear 308
Capítulo VI 311
VI.1. Programas Externos 311
VI.2. Install.com 311
VI.3. Gemat.exe 316
VI.4. Intert.exe 318
VI.5. Reords.exe 319
VI.6. Reordw.exe 320
VI.7. AutoMESH.exe 320
VI.8. AutoLOAD.exe 322
VI.9. Dbcreat.exe 324
Capítulo VII 336
VII.1. Exemplos Acadêmicos 337
VII.1.1. Exemplo I 338
VII.1.2. Exemplo II 339
VII.1.3. Exemplo III 339
VII.1.4. Exemplo IV 340
VII.1.5. Exemplo V 341
VII.1.6. Exemplo VI 342
VII.1.7. Exemplo VII 342
VII.1.8. Exemplo VIII 343
VII.1.9. Exemplo IX 343

VII.2. Aplicações Práticas 2D 372
VII.2.1. Exemplo X 372
VII.2.2. Exemplo XI 373
VII.2.3. Exemplo XII 373
VII.2.4. Exemplo XIII 374
VII.2.5. Exemplo XIV 375
VII.3. Modelos Reticulados 383
VII.3.1. Exemplo XV 383
VII.3.2. Exemplo XVI 384
VII.3.3. Exemplo XVII 385
VII.3.4. Exemplo XVIII 385
VII.3.5. Exemplo XIX 386
VII.3.6. Exemplo XX 387
VII.4. Elementos de Contorno 440
VII.4.1. Exemplo XXI 440
VII.4.2. Exemplo XXII 441
VII.4.3. Exemplo XXIII 441
VII.4.4. Exemplo XXIV 442
VII.4.5. Exemplo XXV 442
VII.4.6. Exemplo XXVI 443
VII.5. Casos Especiais 469
VII.5.1. Exemplo XXVII 469
VII.5.2. Exemplo XXVIII 470
VII.5.3. Exemplo XXIX 471
VII.5.4. Exemplo XXX 472
VII.5.5. Exemplo XXXI 473
VII.5.6. Exemplo XXXII 474

Capítulo VIII 546

VIII.1. Conclusões 546

Apêndice I 548

AI.1. Funções de Forma 548

Apêndice II 562

AII.1. AutoLISP 562

AII.1.1. Conceitos Básicos 563

AII.1.2. Funções 566

AII.1.3. Tipos de Dados Fundamentais 567

AII.1.4. As Funções do AutoLISP 571

Bibliografia 582

CAPÍTULO I

I.1 INTRODUÇÃO

A tecnologia em informática tem realizado, desde sua entrada em nosso mundo, uma autêntica revolução, permitindo–nos a execução de tarefas até pouco tempo inimagináveis. Todos os nossos processos inteligentes foram de alguma maneira afetados por serem em sua essência, mecanismos de transformação, gerência de transmissão de informações.

Vista no tempo, a mudança foi ainda mais espetacular – quase diríamos que explosiva. Em apenas trinta anos, todas as nossas atividades foram "alcançadas". A palavra "cérebro eletrônico" ou "computador", outrora pronunciada com um temor reverencial e um certo caráter mágico, passou a ser de uso comum, uma vez que uma nova linguagem, toda ela em siglas, começou a ser coloquial. Simultaneamente, apareceram os elementos físicos, os aparelhos da nova tecnologia, que denominamos em nossa nova gíria de *hardware*. Essas "geringonças", impulsionadas por um *marketing* agressivo e o encanto indiscutível de suas possibilidades, começaram a passar do nosso local de trabalho aos nossos lares, convertendo–se em mais um símbolo de nossa civilização de consumo.

A análise de tensões foi historicamente a primeira função da engenharia estrutural a ser automatizada.

A aplicação de computadores na solução de tais problemas, possibilitou a inclusão de muito mais detalhes e precisão numérica num modelo matemático que por métodos manuais, aumentando a eficiência do engenheiro.

Hoje, a maioria dos engenheiros responsáveis pela determinação da integridade estrutural de complexos sistemas, tem disponíveis sofisticados programas de elementos finitos para auxiliar nessa tarefa.

A análise numérica em engenharia de complexos sistemas é tradicionalmente considerada como consistindo de três estágios:

(1) **pré–processamento**, ou a preparação da descrição do sistema em uma forma suficientemente completa e detalhada, tal que possa ser analisada por um computador digital.
(2) **processamento**, ou a realização dos algoritmos da análise central.
(3) **pós–processamento**, ou redução e interpretação de análise para obter as específicas informações necessárias para avaliação e decisões.

Embora os usuários do método dos elementos finitos sejam atraídos pela grande liberdade dada para atacar os mais fundamentais problemas, são simultaneamente aterrorizados pela quantidade de dados necessários para aplicações realísticas.

Na maioria dos casos, a tediosa preparação e verificação da malha explica–se por uma grande proporção de dados. A geração automática de malhas é,portanto, uma valiosa prática na redução do peso deste trabalho. Assim sendo, como o usuário necessitará concentrar–se em

apenas uns poucos parâmetros de entrada a ocorrência de erros humanos na preparação de dados diminui grandemente.

Duas filosofias básicas podem ser seguidas para obter uma automação do processo (ZIENCKIEWICZ et alia [89]):

(1) a malha é construída pelo computador a partir de uma mínima quantidade de informações suprida de forma digital.
(2) o posicionamento da malha é estabelecido por computação gráfica interativa usando mesas digitalizadoras ou *mouses*.

No projeto de qualquer esquema geral de geração de malhas, o objetivo é minimizar a entrada de dados consistentes com:

(1) uma adequada descrição do contorno sendo este singular ou multiplamente conectado;
(2) uma facilidade para descrição de zonas de diferentes materiais, onde estes ocorram;
(3) uma facilidade de gradação da malha (concentração variável de elementos) para obter a necessária precisão de idealização;
(4) uma obtenção de elementos de razoável forma para impedir mal condicionamento;
(5) um sistema de numeração o qual resulte em melhor eficiência computacional.

Geralmente se observa que a quantidade de dados supridos dependerá largamente dos três primeiros ítens.

Atualmente, o custo da análise usando técnicas de elementos finitos para uma complexa estrutura é significativamente menor que o custo envolvido na extensa discretização da estrutura e no tempo consumido na interpretação dos resultados.

Houve muitos avanços recentes no aumento da velocidade do *hardware*, técnicas mais eficientes para solução de grandes problemas e elementos finitos mais versáteis e sofisticados.

Implementações de sistemas de modelagem interativa, contudo, têm se conduzido numa proporção muito menor.

Sendo assim, é desejável desenvolver um sistema interativo para auxiliar o projetista a ver seu projeto corrente, modificá–lo se desejado for e facilmente obter gráficos ou tabular resultados da análise baseada em elementos finitos.

A ênfase do presente trabalho está na efetiva **interface homem–máquina** pela qual um engenheiro pode manipular sistemas de análise numérica na pesquisa e projeto. Em particular, o foco está em **Computação Gráfica Interativa**. É apropriado que uma grande quantidade de dados característicos de análise numérica sejam criados e mantidos graficamente, já que um engenheiro pode obtê–los e interpretá–los muito mais rápido visualmente. Isto é particularmente importante, desde que, quase toda a informação em engenharia é por si mesma geométrica ou associada a entidades geométricas.

O termo **interativamente** quando associado a computação gráfica, tem dois significados distintos (ABEL et alii [83]). Primeiro, a resposta do computador deve ser rápida, isto é, preferivelmente em fração de segundos. Isso implica em uma grande quantidade de esforço computacional para controlar o equipamento de apresentação gráfica. Segundo, em adição a apresentação visual dos resultados, os dados do usuário devem ser, o máximo possível, gráficos ou análogos, já que este é o mais natural e realístico modo de comunicação. Este segundo implica em equipamentos específicos para entrada de dados, tais como mesas digitalizadoras (*Digitalizing Tablets*), *Mouse* ou *Lightpen.*

O texto foi concebido pressupondo que o leitor esteja familiarizado com o método dos elementos finitos e suas técnicas de utilização em projeto. Tendo isto em mente, a apresentação de fundamentos teóricos e procedimentos ou técnicas de projeto serão dados apenas quando estes forem necessários para elucidar um problema de programação.

Nos capítulos que se seguem são apresentados os resultados e os meios pelos quais estes foram obtidos, além das justificativas da adoção de tais métodos.

No Capítulo II, apresenta–se de uma forma geral a relação entre computação gráfica, microcomputadores e o método dos elementos finitos, de modo a demonstrar as vantagens nesta união. São apresentadas também as caracteristicas básicas necessárias a sistemas CAD (Projeto Assistido por Computador), geradores de malhas e pré–processadores de elementos finitos.

O capítulo III é dedicado a formulação teórica dos métodos de geração de malhas de elementos finitos. É apresentado um apanhado geral dos métodos mais bem sucedidos encontrados na bibliografia e atenção especial é dada aos métodos implementados na tese.

No capítulo IV é apresentado o ambiente gráfico **AutoCAD** adotado como base no desenvolvimento do pré–processador de elementos finitos. São apresentadas as características básicas de utilização e programação, a estrutura de dados interna e listadas algumas das aplicações deste sistema.

No Capitulo V é apresentado o sistema **AutoMESH** de pré–processamento de elementos finitos proposto e implementado. Esse capítulo compõe o núcleo da tese, onde são apresentados o ambiente de trabalho com suas considerações básicas de operação, entidades estruturais e suas representações gráficas. Os comandos do sistema são descritos e exemplos de operação apresentados.

No capítulo VI são apresentados os programas externos ao ambiente **AutoCAD**. Esses programas fazem parte do sistema **AutoMESH**, mas executam suas tarefas independentemente do **AutoCAD**. Os programas são descritos em funcionamento e aplicações.

O capítulo VII é dedicado aos exemplos de utilização. Os exemplos apresentados visam demonstrar a potencialidade, interatividade e facilidade de utilização do sistema **AutoMESH**. Dentre os exemplos, estão aplicações de todos os tipos, gerados no sistema **AutoMESH** ou carregados neste para visualização ou pós–processamento.

O presente trabalho tem por finalidade, facilitar o trabalho de desenho e projeto de estruturas e também estimular o usuário a extender seus conhecimentos e aumentar a capacidade e versatilidade de seus sistemas CAD. O trabalho também serve como referência para usuários de CAD, gerentes e consultores, os quais necessitam saber como CAD pode ser extendido mum amplo campo de aplicações de produção em projeto ou escritório de engenharia.

CAPÍTULO II

Neste capítulo será apresentada de uma forma geral a relação entre computação gráfica, microcomputadores e o método dos elementos finitos, de modo a demonstrar as vantagens nesta união. São apresentadas as características básicas necessárias a sistemas CAD (Projeto Assistido por Computador), geradores de malhas e pré–processadores de elementos finitos. Estas características são adotadas como critério de seleção e desenvolvimento do trabalho.

Este capítulo visa fornecer as bases de apresentação dos demais capítulos como justificativa de todas as técnicas utilizadas e desenvolvidas e da tese como um todo.

II.1. A COMPUTAÇÃO GRÁFICA

Em 1950, apareceu o primeiro dispositivo de visualização (*display*) controlado por computador. Tal dispositivo foi desenvolvido no **MIT** (*Massachussets Institute of Tecnology*) e era manejado por um dos primeiros computadores, o **Whirlinind–I**. Este primeiro display controlado era constituído por um tubo de raios catódicos ou **CRT** (*Cathode Ray Tube*) similar ao utilizado nos aparelhos comerciais de televisão.

Durante os anos cinquenta, não foram feitos progressos dignos de menção no terreno dos gráficos interativos. A razão desta aparente falta de interesse, era determinada pela inexistência de computadores capazes de suportar este tipo de processo. Por isso, os computadores funcionavam em modo *batch* (também denominado "processo por lotes"), o que impedia que o operador pudesse estabelecer um diálogo operativo.

Ao final dos anos cinquenta, começaram a ser desenvolvidos computadores especializados em processos interativos. Os computadores em questão eram os **TX–0** e **TX–2**, criados no **MIT**. Com a aparição destes computadores, renasceu o interesse pelos gráficos interativos. Contudo, o alvo fundamental neste campo foi a apresentação da tese de doutorado de E. Sutherland, no ano de 1962. Seu trabalho entitulado *Sketchpad: A Machine Graphics Communication System*, estabeleceu as bases dos gráficos interativos por computador, tal como os conhecemos na atualidade, e atuou como catapulta ao convencer importantes companhias, da potência e rentabilidade econômica da nova tecnologia. Como consequência, durante os anos sessenta, tiveram início um conjunto de projetos que começaram a se tornar públicos nos anos setenta.

Do ponto de vista de sua aplicação, os gráficos interativos começaram a existir nos anos setenta, se bem que para isso requeriam a existência de complexos e caros computadores. Esta dificuldade inicial haveria de ser resolvida com o advento dos micros e minicomputadores durante o final da década, circunstância que possibilitou o aproveitamento industrial da tecnologia.

Na atualidade, a redução dos preços do *hardware* e o aumento da potência dos processadores permitiram que o **CAD** entrasse com toda a força no mundo industrial.

Esta situação de mudança está, é lógico, relacionada aos concomitantes avanços nas potencialidades e necessidades de métodos numéricos e no ambiente computacional disponível.

Não apenas a capacidade de processamento tem se tornado mais rápida, como também acessível a um número cada vez maior de usuários.

Com o crescente aumento da capacidade computacional, é claro que uma grande quantidade de informações será processada muito mais rapidamente do que ela possa ser entendida sem efetivo uso, automatizado ou interativo da computação gráfica, que permite tirar vantagem do grande campo abordado pela visão humana.

No contexto dessas crescentes capacidades, o uso da computação gráfica associada aos métodos numéricos tem se tornado comum. Tanto passiva como interativa, a computação gráfica está sendo usada rotineiramente em pré e pós–processamento para análises numéricas em engenharia.

O uso da computação gráfica é vital na obtenção de dois objetivos fundamentais do **CAE** (*Computer Aided Engineering*): produtividade e controle (SHEPHARD et alii [88]). No que cabe a produtividade isto é evidente. Por exemplo, uma grande parte do tempo de um engenheiro é frequentemente gasta na tediosa (e altamente sujeita a erros) tarefa de preparação detalhada dos dados para o computador e interpretação de volumosos resultados. Estas são significativamente mais eficientes por interação gráfica, e assim o engenheiro pode realizar análises mais

rapidamente ou muito mais análises em um dado tempo.

O segundo objetivo é o suficiente controle sobre o processo computacional, tal que permita ao engenheiro intervir durante a execução, não só para obtenção dos resultados desejados, mas também para uma mais eficiente utilização do computador.

A **Computação Gráfica Interativa** foi primeiramente aplicada ao pré e pós–processamento apenas, e estas são realmente as fases onde a produtividade é drasticamente melhorada pelo seu uso.

Nos últimos anos, o uso de **Computação Gráfica Interativa** tem crescido acentuadamente. Com o advento da computação gráfica, a descrição de um problema para análise e projeto pode ser concebida em pouco tempo e com poucos erros.

Pesquisas básicas em computação gráfica continuam sendo feitas e estas inovações serão efetivamente usadas para melhorar a utilização dos métodos numéricos. Por estes trabalhos serem algoritmos, o círculo é fechado no sentido que métodos numéricos comumente usados em engenharia são levados para o desenvolvimento de novos sistemas gráficos para **CAE**. (ABEL et alii [83])

II.2. CAD – PROJETO ASSISTIDO POR COMPUTADOR

Em todas as tecnologias em informática até o momento existentes, é o *software* o elemento mais criativo e atraente. No caso do CAD, a espetacularidade que se consegue alcança cotas dificilmente superáveis.

O tipo de informação a ser tratada e suas implicações no processamento de produção fazem com que o *software* do **CAD** se adiante definitivamente ao *hardware*. Muitos usuários referem–se ao CAD, fazendo menção exclusiva a pacotes particulares de aplicação.

Talvez a qualidade mais significativa da informação utilizada nos sistemas do CAD seja seu elevado número de graus de liberdade. Isto pode parecer uma frase vaga mas ocorreriam, por exemplo, diversas formas de representar um cubo. Essa riqueza de representações possíveis da informação gráfica, objetivo do **CAD**, evidencia a preponderância que o *software* possui sobre o *hardware* nesta tecnologia.

O **CAD** não é um produto que pretende gerar gráficos, mas sim elementos geométricos. Não se deve confundir a visualização dos gráficos no terminal, com o conceito geométrico que a mencionada distribuição possui. Assim, pode ocorrer que um conjunto de pontos de um terminal gráfico, em disposição mais ou menos alinhada, seja a representação de um elemento geométrico (uma reta) ou simplesmente se tratem de *pixels* sem informação geométrica associada.

O **CAD** trata, portanto, de construir geometria e não simplesmente gráficos; a visualização desta geometria é um subproduto do **CAD**, mas de forma alguma é seu objetivo.

O **CAD** está diretamente associado a idéia de **gráficos interativos**, expressão cujo significado vem determinado pelo termo interativo, com o qual queremos indicar que o operador, mediante sua ação sobre determinados dispositivos, produz modificações dos gráficos que o computador mostra. Efetivamente, se estabelece um diálogo entre o dispositivo denominado **estação de trabalho** e o **computador**.

Se é possível projetar manualmente, então é igualmente possível criar um sistema **CAD** o qual produza o mesmo resultado em uma fração de tempo. Mas enquanto isto é uma meta altamente desejável para liberar o projetista da tediosa aritmética, esta atitude não faz mais do que apenas iniciar a exploração do oferecido pelos modernos computadores.

Mesmo embora o computador não substitua uma bem fundamentada intuição de projeto, um vantajoso subproduto do seu uso, é este em que um projetista pode construir uma profunda experiência muito mais rapidamente. Liberto do tempo de calcular, um número de diferentes esquemas pode ser processado em um pequeno tempo, auxiliando assim o projetista a devotar–se ao planejamento, avaliando resultados e tomando decisões.

É inteiramente possível escrever um programa que é capaz de produzir uma solução automática sem intervenções do usuário, senão sua especificação inicial dos parâmetros de projeto. Mas estas, são soluções únicas, ótimas apenas no campo envolvido nos critérios do programador.

Na prática, otimização é em grande parte, um exercício intuitivo o qual frequentemente é melhor servido por programas que dão aos projetistas a oportunidade de definir a rota para a solução.

A visualização de objetos é uma parte inerente de desenho e projeto técnico. Técnicos de desenho e convenções e muitas disciplinas de engenharia ou projeto, têm envolvido de uma forma universal o aspecto visual que comunica ambos, o conteúdo e a execução de uma idéia.

Sistemas de projeto/desenho por computador tornam-se crescentemente mais acessíveis a projetistas, engenheiros, arquitetos e desenhistas na proporção em que o *hardware* e *software* decrescem de custo.

Sistemas **CAD** são uma aplicação exigente para tecnologia baseada em microcomputadores, devido a natureza intensiva dos cálculos. O projetista de programas **CAD** enfrenta um interessante espectro de opções na estruturação de como o programa armazena e manipula os elementos gráficos que compõem suas imagens.

Há duas considerações no projeto e função de sistemas CAD:

(1) o conceito de projeto ou modelagem de sólidos e,
(2) o conceito de desenho manual. (PFORTMILLER [53]).

O primeiro, cria um modelo análogo dos objetos ou processo e os armazena como dados geométricos. Este procedimento pode ser usado para ambos, visualização real do objeto e geração de tradicionais desenhos de produção. O segundo cria e manipula **primitivas gráficas** básicas. O desenho manual e desenho de linhas e arcos no papel é repassado ao computador por operações similares.

Um programa escrito com a finalidade de produzir uma representação realística de um modelo tridimensional em um monitor gráfico, precisa possuir certas características básicas: (HOFMEISTER [30])

(1) **Projeção Ortográfica** – Esta é a projeção de um objeto tridimensional em um plano bidimensional, onde todas as linhas de projeção são paralelas (linhas do objeto tridimensional para os correspondentes pontos na sua imagem bidimensional projetada). Este tipo de imagem é necessária para a produção de desenhos de trabalho, devido a distância do objeto tridimensional para o usuário.

(2) **Projeção em Perspectiva** – Neste tipo de projeção, os raios visuais partem do observador em vários ângulos, ao invés de todos serem paralelos como na projeção ortográfica. Assim a imagem bidimensional projetada é distorcida, na proporção da distância do observador ao objeto tridimensional, e a distância do observador ao plano de projeto.

(3) **Escala** – Isto permite ao usuário do programa gráfico apresentar a imagem bidimensional em qualquer tamanho desejado, o que possibilita o uso de uma variedade de monitores gráficos tendo diferentes áreas para apresentar a mesma imagem.

(4) **Posicionamento da Imagem** – Esta característica dá ao usuário, a habilidade de posicionar a imagem em qualquer parte do espaço disponível na tela do monitor, e rodar a imagem.

(5) **Construção de Janelas (*Windowing*)** – Permite ao usuário apresentar apenas específicas porções do objeto tridimensional.

(6) ***Clipping*** – Remove do objeto tridimensional aquelas linhas que ficam fora da janela definida pelo usuário.

(7) **Remoção de Linhas Ocultas** – Remoção ou pontilhamento das linhas não vistas de determinado ponto de observação. Esta característica essencialmente transforma a imagem transparente projetada em opaca, e constitui a maior porção da lógica de um programa.

(8) **Apresentação da Imagem** – Apresentação de uma imagem bidimensional projetada (por janela, *clipping* e variação da escala) para propósitos de verificação da precisão do modelo de elementos finitos, e auxilio ao usuário para conferência de dados usados na preparação do modelo.

(9) **Desenhar elementos específicos** – O usuário de um programa gráfico deverá ter a opção de especificar quais elementos do modelo deverão aparecer no desenho final. O usuário poderá construir desenhos apenas com nós ou elementos por exemplo.

(10)**Especificação de cotas** – O sistema deverá permitir o desenho e a anotação das dimensões, diretamente dos dados do modelo de elementos finitos.

Os sistemas **CAD** têm provado ser extremamente úteis para a automação de projetos e procedimentos de desenho bidimensionais. Notável sucesso tem sido obtido por sistemas **CAD** tridimensionais para o projeto de superfícies curvas ou esculturadas na indústria automotiva, aeroespacial e naval. Embora tais sistemas **CAD** também chamados *Wire–frame* (estrutura de arames) sejam úteis para o projeto das superfícies externas, (por exemplo painéis metálicos de automóveis) são impróprios e difíceis de usar para o projeto de componentes sólidos funcionais tais como *pistons* de motores, e outras partes que são usualmente moldadas, prensadas ou forjadas. Com o desenvolvimento de sistemas de modelagem de sólidos, tem–se caminhado para a construção de sistemas CAD, os quais integram as distintas funções de projeto da parte mecânica e análise estrutural.

II.3. O MÉTODO DOS ELEMENTOS FINITOS

O termo **método dos elementos finitos** geralmente indica a análise, mas seu programa consiste das seguintes partes: **pré–processador**, **processador** e **pós–processador**. O **pré–processador** trabalha para diminuir a tediosa tarefa do usuário na preparação de todos os dados da análise. O **processador** trabalha de acordo com os dados de entrada, os quais são gerados no **pré–processador** e os resultados são governados por estes mesmos dados. O fato indica que um melhor **pré–processador** pode não apenas salvar o custo para a preparação dos dados de entrada, mas também dar melhores resultados e prover a veracidade destes.

Antigamente, programas de elementos finitos eram primitivos em projeto, ineficientes e enfocavam na implementação o uso do método, ao invés de otimização e utilização de pré e pós–processadores. Nestes programas, tipicamente os elementos e dados nodais eram introduzidos sequencialmente como matrizes ou vetores individualmente definidos pelo analista. Este procedimento manual é, não apenas cansativo, mas também facilmente admite erros na descrição da discretização e dados. Como o método é aplicado sucessivamente a grandes problemas, o tempo para a preparação e edição também se torna proibitivo. Algumas dessas dificuldades têm sido aliviadas pelo desenvolvimento e implementação de algoritmos de geração de malhas e, mais recentemente, algoritmos de refinamento adaptativo.

Até recentemente, **pré–processadores** para elementos finitos tinham se concentrado na geração automática da malha sem dar muita importância ao provimento da comunicação entre analista e máquina. De um amplo ponto de vista, um **pré–processador** é agora considerado como uma combinação de código e técnicas de entrada de dados, os quais melhor aliviam o analista da árdua tarefa de geração do banco de dados e verificação, enquanto enfatizam os aspectos criativos de projeto e modelagem.

Para prover uma eficiente interface entre analista e máquina, um **pré–processador** de elementos finitos necessita realizar as seguintes tarefas (PERUCCHIO [159]):

(1) reduzir ao mínimo o tempo gasto na preparação e verificação dos dados estruturais;
(2) prover a precisão e qualidade dos dados que a estrutura necessita.

O mais eficiente meio para gerar dados geométricos, imediatamente apresentados em um monitor, com a mínima quantidade de entrada digitada via teclado, é através de mesas digitalizadoras ou *mouses*. Esta resposta visual dá ao usuário a capacidade de monitorar a estrutura de dados e assim reduzir grandemente a tarefa de detecção de erros e modificações de dados.

O termo *User Friendly* tem sido usado para descrever um **pré–processador**, o qual pode operar com relativa facilidade, pressupondo–se que o usuário é um engenheiro que tem alguma familiaridade com projeto assistido por computador (**CAD**).

As informações que são produzidas por pré–processamento podem ser subdivididas em três grandes grupos (PESQUERA [156]):

(1) geometria;
(2) atributos e cargas;
(3) dados suplementares.

A **geometria** é definida pela coleção de pontos, linhas, superfícies e volumes que definem a estrutura. A geração da geometria é considerada a maior tarefa de qualquer pré–processador. Atributos incluem as informações relativas aos elementos da estrutura, tais como condições de

contorno, lados dos membros e propriedades dos membros.

Em geral, **cargas** podem incluir carregamentos estáticos e dinâmicos, bem como influências tais como efeitos térmicos.

Dados suplementares consistem de informações particulares à estrutura, mas não especificamente associadas a quaisquer de seus componentes. Exemplos deste tipo de informação são o título do problema, e os parâmetros usados em especificações e controle da análise.

Em geral, a especificação do terceiro tipo de informação não requer recursos gráficos interativos, mas um completo pré–processador gráfico acomodará também este aspecto ou definição do problema.

Várias técnicas de geração automática tem sido usadas largamente, porém, sua aplicabilidade ao pré–processamento interativo precisa ser julgada pelo grau de controle que o analista retém sobre a malha final e sobre a maioria das informações geométricas requeridas para cada técnica. Infelizmente, os dois requisitos competem no sentido que um refinamento no controle da malha é usualmente pago em termos de um grande banco de dados inicial.

Um gerador de malhas necessita ter as seguintes características (HABER [82]):

(1) **Uma precisa modelagem do contorno.** Nenhum erro na discretização inerente à escolha do modelo de elementos finitos deve ser introduzido pelo processo de geração. Não deverá ser limitado na forma das curvas do contorno que podem ser modelados.

(2) **Boa correlação entre o interior da malha e a informação em seu contorno.** As curvaturas e espaçamento entre nós no contorno da região deverão ser representados no interior da malha. Isto permite ao analista, redefinir o espaçamento da malha onde uma mais precisa discretização é necessária. Refinamentos desnecessários são também evitados.

(3) **Um mínimo esforço de entrada de dados**, para reduzir o tempo requerido pelo analista para definir um modelo de elementos finitos. A maioria dos dados de entrada requeridos deverão ser também reduzidos tanto quanto possível. Isto diminuirá a chance de introdução de erro humano na análise.

(4) **Um vasto campo de aplicabilidade** para minimizar o tempo de aprendizagem do usuário, tempo de desenvolvimento e tamanho do programa. É desejável que um pequeno conjunto de técnicas de geração de malhas possa ser aplicado a um vasto campo de topologias estruturais, ao invés de um grande conjunto de geradores de malhas específicos.

(5) **Topologia geral.** O método de geração da malha não deverá impor nenhuma restrição na topologia da malha em uma região.

(6) **Geração automática da topologia.** No sentido de gerar uma malha de elementos finitos, deverá criar a conectividade dos elementos sem a interferência do usuário. Isso reduz a quantidade de dados do usuário, sendo que, esta característica pode estar em conflito com a necessidade de topologias gerais.

(7) **Formatos favoráveis de elementos.** Os elementos produzidos por métodos automáticos deverão possuir formas que não produzam mau condicionamento no modelo de elementos finitos.

(8) **Numeração otimizada.** A numeração de nós e elementos da estrutura deverá ser tal que, condições favoráveis sejam obtidas para a solução do sistema de equações. O que constitui uma condição favorável, depende do método de solução de sistemas de equações usado no programa de análise. Mínima largura de banda, *Wavefront* e *Profile* são algumas características comumente desejadas. Já que estas técnicas referem–se à estrutura inteira, não é suficiente considerar uma simples região isoladamente. Para estruturas com muitas regiões, os nós comuns entre duas ou mais regiões adjacentes, deverão aparecer apenas uma vez no banco de dados.

(9) **Eficiência computacional.** O método de geração de malhas deverá fazer eficiente uso dos recursos do computador de forma a minimizar os custos e prover boa resposta quando aplicado em um ambiente interativo.

Para contornos arbitrários onde nenhuma descrição contínua pode ser usada, técnicas de digitalização podem ser utilizadas para auxiliar a redução do problema de entrada de dados.

Um pré–processador ideal deverá ter as seguintes qualidades e características básicas:

(1) **Flexibilidade.** O usuário deverá estar habilitado a seguir qualquer caminho que deseje na definição da estrutura. Por exemplo, uma vez que a geometria tenha sido definida, poderá escolher a ordem na qual prefere especificar os atributos dos nós e membros.

(2) **Retorno a um estado anterior.** Uma ação incorreta do usuário pode ter efeitos indesejados, talvez drásticos no banco de dados. Uma previsão deverá ser feita para recuperar o estado do banco de dados anterior a última ação do usuário.

(3) **Geração** de ***backup***. Se um erro fatal é cometido durante a operação do programa, as informações geradas anteriores a um certo estágio, deverão estar disponíveis para um novo início, usando as informações recuperadas.

(4) **Entrada de dados flexível.** A entrada de informações deverá ser tão fácil quanto possível. Modos alternativos de entrada de dados deverão ser previstos. Exemplos: mesas digitalizadoras, *mouses*, terminais alfanuméricos, arquivos.

(5) **Resposta visual.** A resposta gráfica para toda ação executada pelo usuário deverá estar imediatamente disponível assim que a ação tenha sido executada.

(6) **Checagem.** Deverá ter a habilidade de verificar e realizar resposta visual, de acordo com a ação prévia, atribuições e definições.

(7) **Simplificações na apresentação.** A imagem de uma estrutura sendo apresentada pode ser complicada. Operações tais como a habilidade de isolar regiões de uma estrutura devem ser providas para auxiliar na compreensão da estrutura.

(8) **Comando para duplicação.** A habilidade de assumir os mesmos atributos ou quantidade para múltiplas entidades ou locais com um simples comando é uma característica frequentemente usada e deve ser incorporada.

(9) **Documentação conectada.** Deverá ter descrição das opções disponíveis ao usuário durante toda operação, arquivos tipo *help*.

(10) **Expansibilidade.** Previsão deverá ser feita para facilitar a adição de novas opções e operações.

(11)**Transportabilidade**: Para qualquer pré–processador ter total sucesso este necessita ser capaz de ser usado em uma variedade de diferente sistemas. Este é de todos o mais difícil requisito a satisfazer.

Desde que, a maioria das análises por elementos finitos apresentem saídas de dados longas e relativamente complexas, um engenheiro precisa gastar uma quantidade considerável de tempo para total compreensão dos resultados e dedução de conclusões. Ele pode, então, tentar expor e suportar suas conclusões em um formato que é facilmente entendido por clientes, perícia ou gerência. A computação gráfica avançada provê o recurso para resolver este problema.

II.4. OS MICROCOMPUTADORES

O advento dos microcomputadores tem dado ao analista de elementos finitos uma conveniente ferramenta para o pré e pós–processamento de grandes bancos de dados comumente encontrados em tais análises.

A atual tecnologia de apresentação gráfica de computadores pessoais é adequada a muitas tarefas que requeriam sofisticadas estações de trabalho a poucos anos atrás.

Aumentando a potência do equipamento e diminuindo preços, chegaremos a muito mais sofisticados sistemas **CAD**, capazes de prover os mais complexos projetos de engenharia, análise, e técnicas de modelagem de sólidos agora disponíveis apenas em sistemas muito grandes.

Uma revisão dos computadores existentes indica que é prático o uso de microcomputadores não dispendiosos para a análise de sistemas estruturais com várias centenas de incógnitas. Isso não é possível, contudo, se os existentes programas estruturais são diretamente convertidos de grandes computadores para microcomputadores sem extensivas modificações na organização do programa.

Os primeiros *chips* (processadores) baratos para microcomputadores foram desenvolvidos a mais de 10 anos. Até este ponto, contudo, menos que um por cento das análises de elementos finitos está sendo conduzido neste tipo de equipamento.

Há muitas razões válidas pelas quais houve esta significante espera no uso de microcomputadores para análises práticas de elementos finitos. Primeiro, houve vários anos de espera no desenvolvimento de um estável sistema operacional, após o desenvolvimento da unidade de processamento central (**CPU**). Também, antes que programas de elementos finitos possam ser desenvolvidos ou transferidos para microcomputadores um seguro FORTRAN, ou linguagem de programação, precisa existir sendo compatível tanto com a **CPU** quanto com o sistema operacional. Apenas depois destas duas etapas estarem completas, o computador (*hardware*) pode ser usado em um prático ambiente de desenvolvimento.

Até o presente momento, o uso de programas de elementos finitos de propósito geral não está restrito a grandes empresas. Dentro de poucos anos, estarão disponíveis para pequenas firmas de engenharia, baratos sistemas computacionais de modo a terem a mesma capacidade computacional dos atuais sistemas existentes de grande porte.

Sistemas de multi–usuários requerem um especialista, ou super usuário para organizar ou manter separadas contas e senhas. Todos os arquivos são usualmente mantidos em um *hard disk* (disco de armazenamento) do sistema. Um único usuário tornar–se–á um usuário de segunda classe, que não é responsável pela seleção ou mudanças no sistema operacional. Isto leva a estabelecer um centro computacional dentro da empresa e altos investimentos relativos a salário de gerenciamento aos recursos computacionais da companhia.

Muitas organizações, as quais adotaram este procedimento, tem achado que custos internos com seu próprio computador foram significativamente crescendo relativamente ao uso de serviços de *bureau* de computador externos.

A segurança de sistemas multi–usuário é consideravelmente menor que vários sistemas de único usuário. Quando o sistema multi–usuário é "derrubado", ou estragado, uma grande quantidade de usuários é afetada e importantes arquivos podem ser perdidos.

Sistemas de um único usuário são simples de usar. Neles, o engenheiro é responsável pela seleção e alterações no sistema operacional, linguagem e armazenamento de arquivos. A velocidade dos cálculos é sempre a mesma e não é função do número de usuários no sistema. A capacidade de "ver" o tempo requerido para as várias fases da solução de um problema de engenharia, tem significante valor educacional no projeto de novos métodos computacionais para a solução de problemas.

Sistemas operacionais, os quais acessam dados fora de sua real memória, são chamados **sistema operacionais virtuais**, e a transferência de blocos de dados de disco com baixa velocidade de acesso, para a memória com alta velocidade é chamada **paginamento**.

Uma das significantes desvantagens de tais sistemas é que um número muito grande de usuários causará um extensivo paginamento e a performance do computador pode deteriorar–se significantemente O *overhead* no tempo de execução, o qual é inerente aos sistemas de paginamento multiusuário, tornará este tipo de sistema obsoleto no futuro por causa da redução dos custos dos microcomputadores e memória real.

É evidente a necessidade de programas que sintetizem os méritos de duas categorias, isto é, realizem uma refinada e conceitualmente correta análise e tomem total vantagem das características interativas e gráficas de microcomputadores.

A efetiva exploração dos recursos gráficos e interativos de microcomputadores pode ser julgada com base nas seguintes características:

(1) **Legibilidade**
(2) **Diagnósticos**
(3) **Corregibilidade**

em particular,

(1) a entrada e saída de dados deverá ser apresentada de uma forma tão sintética e exaustiva quanto possível, pela construção, sempre que viável, de representação gráfica, ao invés de impressão de valores numéricos e pela automática realização de todas as operações necessárias a minimizar a elaboração imediata de gráficos usualmente feita pelo engenheiro.

(2) particular cuidado deverá ser devotado a diagnósticos, pela apresentação automática de todas as inconsistências entre os dados e/ou resultados e princípios gerais, e pela apresentação de representação gráfica, a qual faz identificação de erros bem cedo estimulando o engenheiro a sistematicamente verificar dados e resultados.

(3) Correções deverão ser feitas tão fáceis quanto possíveis, pela interação durante qualquer fase da elaboração do problema e por realização automática dos ajustes consequentes de uma simples correção.

O principal propósito de um pré–processador é reduzir o tempo do usuário, necessário para preparação de dados. Ao lado desta função, há a exigência de reduzir erros pela realização de operações de checagem mais fáceis.

Vantagens de computadores pessoais:

(1) **Preço**

Um tradicional programa de elementos finitos necessita de um computador de grande porte que, geralmente, é um investimento nada modesto e causa decorrentes custos para operação. Um computador pessoal e os devidos programas de elementos finitos são pagos por um valor comparativamente modesto. Assim, a alta qualidade dos cálculos técnicos torna–se acessível para pequenas firmas e consultores.

(2) **Conveniência**

Na era da régua de cálculo, o engenheiro tinha todo o equipamento necessário na sua mesa. Com os computadores de grande porte, contudo, ele gasta horas e horas em salas de terminais e busca de listagem no centro de computação, frequentemente em vão. Ao contrário, com computadores pessoais a seu serviço a situação ideal é refeita, ficando o engenheiro com todo o equipamento necessário ao seu alcance.

(3) **Tempo**

Pode–se dizer que, uma grande quantidade de processos de análise por elementos finitos consome um tempo total de processamento menor em computadores de grande porte. Isto depende primeiramente da interatividade do *hardware* oferecido e um programa conversacional bem escrito. Sem dúvidas, na construção das matrizes de rigidez e solução do sistema de equações, os computadores grandes são altamente superiores. Sempre há, contudo, o problema de filas e prioridades, e os processos acabam sendo executados à noite e os erros só serão percebidos na manhã seguinte. Uma semana é assim, rapidamente gasta. Em computadores pessoais, o usuário inicia o processo uma vez. O computador se situa em sua mesa e dá um sinal acústico se algum erro ocorre e indica qual o tipo de erro, podendo ser este imediatamente corrigido. Frequentemente, o problema é resolvido no mesmo dia.

(4) **Interatividade**

Um programa bem escrito para computador pessoal sempre indica via monitor o que está sendo feito, isto é, "construindo matriz de rigidez do elemento 27", por exemplo. Deste modo, o usuário pode avaliar como o processamento está indo e quando ele precisará vê–lo novamente.

(5) **Gráficos**

Geralmente computadores pessoais possuem possibilidades gráficas de alta qualidade. Resposta gráfica rápida é uma necessidade na fase de geração de malhas e apresentação dos resultados.

(6) **Transportabilidade**

Por causa de sua forma compacta e pequeno peso um computador pessoal é fácil de transportar.

CAPÍTULO III

Este capítulo é dedicado a formulação teórica dos métodos de geração de malhas de elementos finitos. É apresentado um apanhado geral dos métodos mais bem sucedidos encontrados na bibliografia.

Em todos os métodos descritos a seguir, uma partição é imposta na estrutura a qual cria uma ou mais regiões de forma simples. Uma partição em segundo nível é imposta em cada região, a qual leva diretamente a desejada malha de elementos finitos.

As técnicas descritas abaixo são concebidas com uma partição em segundo nível e pressupõem que o primeiro nível tenha já sido realizado.

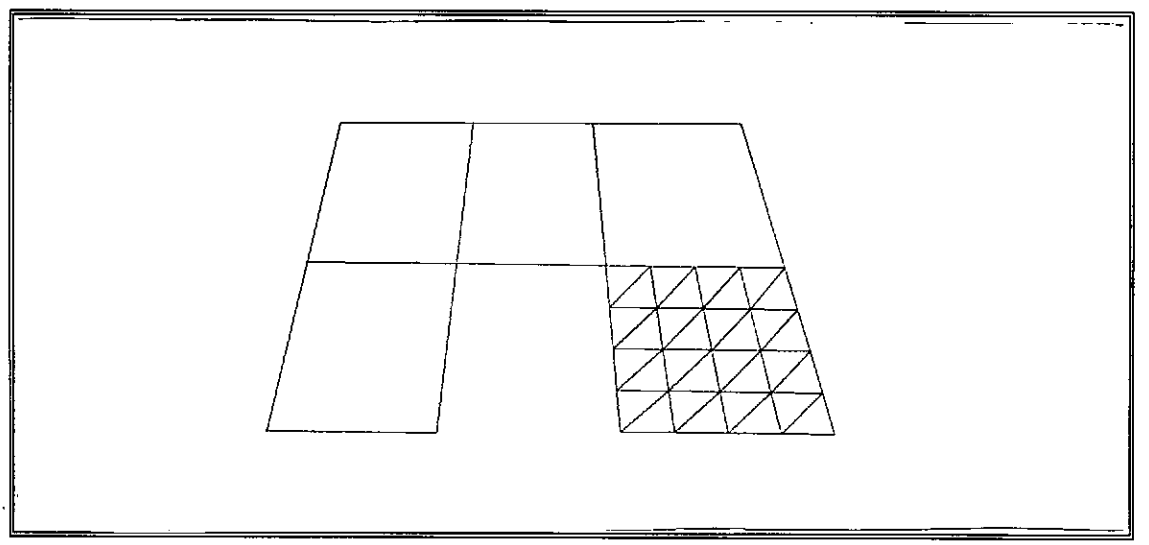


Fig III.1

Para avaliar os méritos relativos a estes diferentes métodos, é usual notar as seguintes características desejadas em técnicas de geração automática de malhas:

1. precisa modelagem do contorno;
2. boa correlação entre a malha interior e a informação no contorno da malha;
3. mínimo esforço de entrada de dados;
4. amplo campo de aplicações;
5. topologia geral;
6. geração automática da topologia;
7. formato favorável dos elementos;
8. numeração ótima;
9. eficiência computacional.

III.1. GERAÇÃO DE MALHAS

Nesta seção, daremos uma breve revisão em algumas das mais bem sucedidas tentativas de "automatizar" a geração de malhas de elementos finitos (o termo "automatizar" é, é lógico, um exagero na maioria dos casos desde que o analista usualmente interage ou contribui significantemente no processo de geração).

A maioria dos geradores desenvolvidos, são referenciados aqui como interpoladores ou geradores de malha por interpolação. Estes geradores requerem alguma forma de particionamento da estrutura em sub–blocos mais simples. Um sub–bloco é, então, modelado primeiro por

aproximação do seu contorno com curvas paramétricas (geralmente polinômios de baixa ordem ou *piecewise polynomial interpolants*). O processo inteiro é equivalente à construção de um mapeamento de um vetor a partir de um domínio canônico (por exemplo, um cubo) em um sub–bloco. O mapeamento pode interpolar o sub–bloco em um número de pontos finitos ou infinitos.

O mapeamento produz uma descrição matemática explícita do sub–bloco e pode ser usada para induzir–lhe coordenada natural (geralmente retangular).

O segundo tipo de gerador de malhas, provêm uma geração de malhas de elementos finitos muito útil em duas dimensões: são os chamados trianguladores automáticos. Em aplicações bi–dimensionais, esses algoritmos produzem construções irregulares de triângulos, os quais facilmente acomodam gradações locais de malha. Em aplicações tri–dimensionais, agregados de tetraedros são gerados, os quais podem ser ditos como triangulações tridimensionais. Com exceção do gerador descrito por FREDERICK [112], nenhum dos geradores bi–dimensionais estende–se naturalmente para três dimensões.

III.1.1. FUNÇÕES DE INTERPOLAÇÃO

Considerável esforço tem sido devotado a problemas planos e três diferentes procedimentos básicos têm sido aplicados. Primeiro, pode–se gerar a locação de um número específico de pontos no interior e contorno e então constrói–se a malha pela união dos pontos. A segunda técnica é

geração de malhas isoparamétricas, na qual coordenadas curvilíneas são construídas em regiões constituídas de polinômios de ordem relativamente baixa. Uma extensão da técnica de coordenadas curvilíneas isoparamétricas é o esquema de interpolação transfinita, a terceira técnica, na qual uma generalizada interpolação multivariada é usada para mapear um domínio quadrado unitário em um domínio bidimensional a ser discretizado.

A idéia de representação de superfícies ou curvas é de interpolar em um meio homogêneo, através de alguns dados discretos, por aplicação de "funções de peso" ou interpolantes. Uma superfície a qual é uma função bivariada, isto é, uma função de duas variáveis, pode ser sintetizada a partir de pontos ou curvas, enquanto uma curva (ou função univariada) pode apenas ser interpolada através dos pontos. Com base nos dados discretos aplicados, a representação de superfícies pode ser classificada em três categorias. A primeira categoria é o produto cartesiano a qual interpola apenas através de pontos dados. As outras categorias são as representações *lofting* e transfinita – ambas interpolam através de curvas. O último é uma interpolação através de duas famílias de curvas paramétricas, enquanto a definição de *lofting* utiliza apenas uma pré–selecionada família de curvas paramétricas, isto é, curvas as quais têm diferentes valores fixos da mesma coordenada paramétrica.

Estes três tipos de representação de superfície correspondem a uma hierarquia de sofisticação e complexidade matemática. Destes, o produto cartesiano de representação de superfícies é o mais simples, mas o que apresenta o menor controle sobre a qualidade de representação; enquanto a definição de superfície transfinita é a mais complexa mas a que tem maior controle sobre a geometria resultante.

Os mais comuns interpoladores usados na aproximação de curvas são polinômios de várias formas e ordens. É preferível selecionar uma ordem de polinômio, apenas alto o suficiente para garantir a mínima continuidade requerida na malha resultante, já que polinômios de alta ordem tendem a ondular entre pontos selecionados e requerem um esforço computacional extra.

A flexibilidade do método para modificar as curvas aproximadas é, também, essencial em um processo de aproximação interativo, no qual uma detalhada digitalização da seção é mantida para comparação com a curva aproximada, até uma marca satisfatória ser obtida.

Métodos clássicos de aproximação de curvas incluem esquemas de interpolação do tipo lagrangiano ou hermitiano. Interpolação lagrangiana marca um número de pontos na curva original, e a ordem dos polinômios interpolantes cresce indesejavelmente quando o número de pontos dados cresce. O método hermitiano é usualmente implementado como um caso particular no qual cada segmento da curva é interpolado em termos de posições e derivadas de alguma ordem, apenas nos dois pontos extremos de um segmento. A curva composta é, então, uma combinação de pequenos segmentos, unidos com continuidade de posição e de específicas derivadas nos intersegmentos. O caso particular do método hermitiano evita assim, o problema de polinômios de ordem excessivamente alta e é menos restritivo que o procedimento lagrangiano. Embora o caso particular de interpolação do tipo hermitiano seja altamente usado em indústrias automotivas e aeroespaciais, quando o procedimento é aplicado em um modo interativo é difícil ao usuário visualizar o efeito na curva, devido às mudanças nos valores das derivadas nas uniões entre segmentos.

Outro tipo de esquema de interpolação é um, no qual a aproximação da curva é definida por um polígono, ao invés de pontos na curva original. Aqui, a aproximação da curva é o resultado da suavização da forma do polígono de alguma maneira, embora a curva não passe necessariamente através dos vértices do polígono. Desde que a curva tenda a copiar a forma do polígono de controle, é relativamente fácil perceber como mudanças no polígono afetarão a curva resultante. Este é um recurso desejável para um processo de aproximação iterativa.

Tanto curvas de *Bezier* como *B–Spline* pertencem a este tipo de método de interpolação. O último, porém, é melhor para aproximação de curvas que o anterior pelas seguintes razões. Primeiro, a ordem de uma curva de *Bezier* cresce quando o número de lados do polígono cresce, enquanto a ordem de uma *B–spline* é determinada para o propósito da aplicação e não a complexidade da curva. Segundo, desde que a forma resultante de uma curva *B–spline* é mais fechada com o seu polígono de controle que a curva de *Bezier*, é fácil modificar uma *B–spline* interativamente para alcançar a curva original. Finalmente, quando a posição de um vértice do polígono de controle é mudado, o efeito na curva é localizado para a vizinhança do vértice mudado para uma curva *B–spline* mas não localizado para uma curva de *Bezier*; esta característica do anterior é a significante vantagem para manipulações efetivamente interativas.

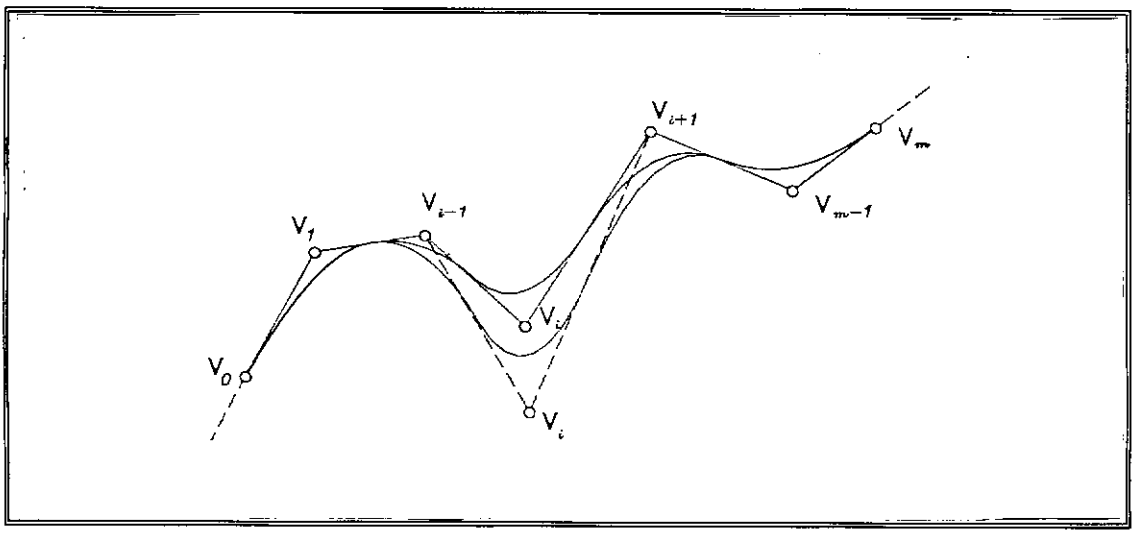


Fig III.2

III.1.2. **MAPEAMENTO**

O problema de gerar as coordenadas dos pontos de uma superfície no espaço pode ser colocada da seguinte forma: dados os pontos (r,s) de uma região plana definida em um domínio Ω de referência, determinar as coordenadas x, y, z dos pontos da superfície S.

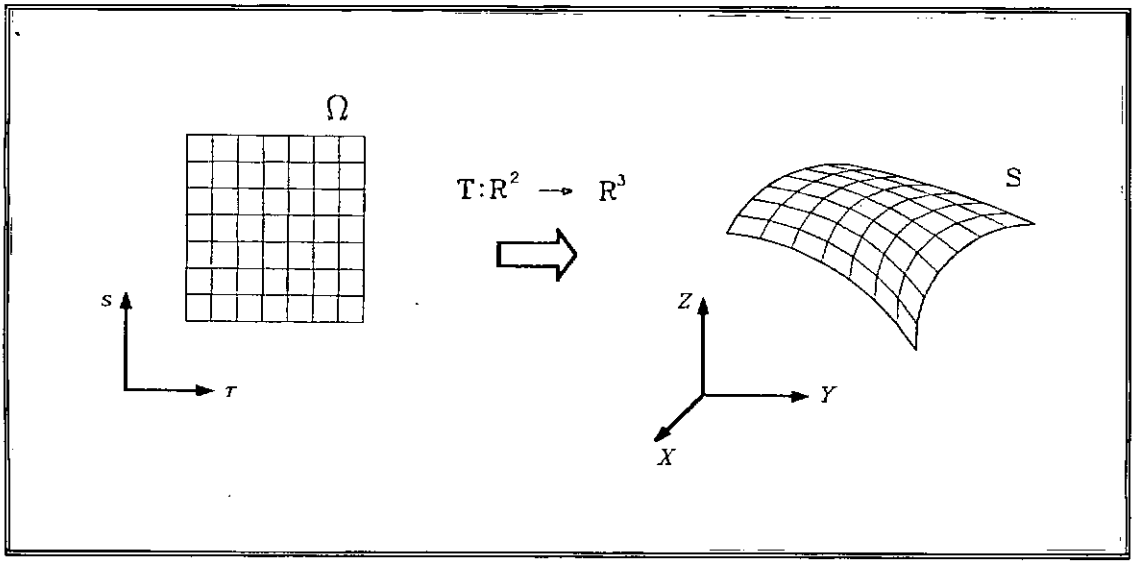


Fig III.3

Trata–se portanto, de determinar a transformação T:$\mathbb{R}^2 \rightarrow \mathbb{R}^3$ que mapeia a região plana na superfície *S*. Matricialmente pode–se escrever:

$$\begin{vmatrix} x \\ y \\ z \end{vmatrix} = \underset{\sim}{T} \begin{vmatrix} r \\ s \end{vmatrix} \qquad \text{(III.1)}$$

A determinação simples e direta de tal relação é praticamente impossível na maioria dos casos. Essa dificuldade pode ser contornada, porém, utilizando–se uma estratégia semelhante a do método dos elementos finitos, ou seja, define–se inicialmente a transformação e então a superfície é aproximada por um número de pontos pré–definidos no domínio de referência.

III.1.3. MAPEAMENTO LAPLACIANO

Os esquemas laplacianos de geração de malhas são uma família de variações em um método básico, o qual é atribuído a WILSON [62]. No método básico, os nós interiores são posicionados de acordo com o vetor Pn e cada nó i interior satisfaz a seguinte equação:

$$P_i = \frac{1}{2N_i} \sum_{n=1}^{N_i} (P_{nj} + P_{nl}) \qquad \text{(III.2)}$$

onde N_i é o número de elementos conectados ao nó i, P_{nj} e P_{nl} são vetores de posição dos nós adjacentes no vizinho elemento n, como mostrado na figura III.4;

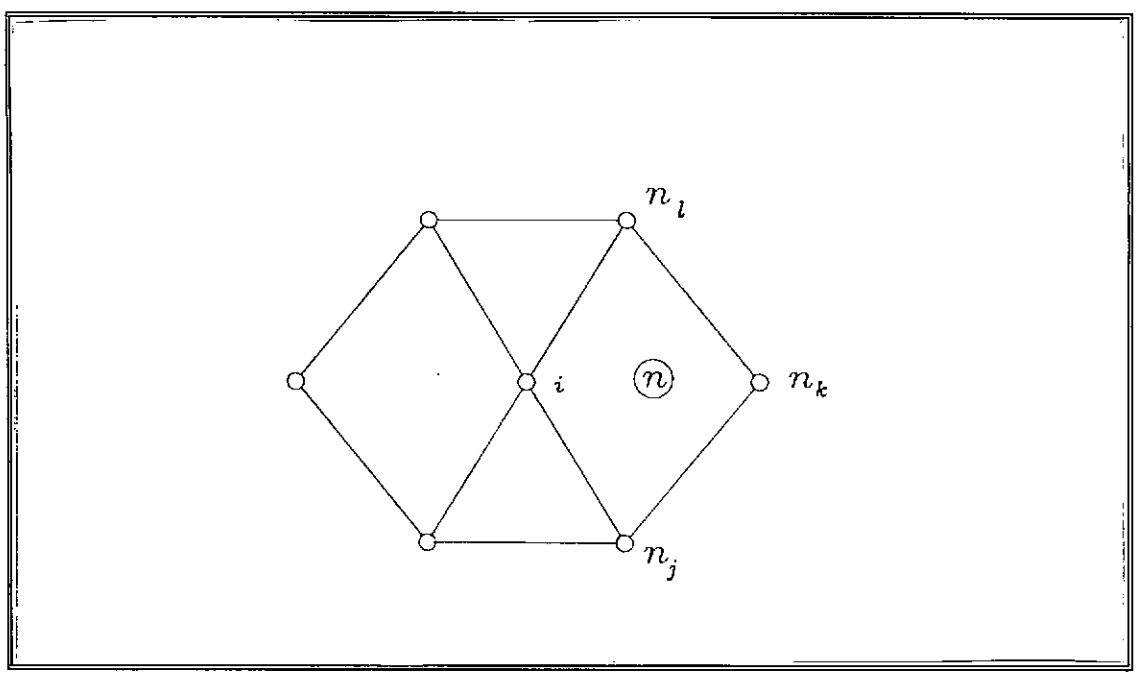


Fig III.4

O nome do método deriva do fato desta equação (III.2) poder ser interpretada como o operador laplaciano de diferenças finitas para as incógnitas P_i, quando uma malha retangular é usada. Malhas produzidas por esta técnica tenderão a ter elementos de forma moderadamente uniforme. obtida via uma técnica iterativa, tal como *Gauss–Seidel* ou *Jacob.* Isto representa um esforço computacional relativamente grande para geração de malhas. Uma desvantagem deste método é que a malha resultante não representa adequadamente informações do contorno no que diz respeito ao espaçamento e curvatura. Uma modificação do esquema básico foi desenvolvido por HERMANN [107] para aliviar este problema. Neste método, a equação iterativa deriva da equação para a origem do sistema de coordenadas paramétricas em um elemento finito isoparamétrico de oito nós:

$$P_i = \frac{1}{N_i \ (2-W)} \sum \ (\ P_{nj} + P_{nl} - W \ P_{nk} \ ) \ ;$$

$$0 \leq W \leq 1 \qquad \text{(III.3)}$$

Diferentes valores de W produzem uma família de esquemas chamados esquemas laplacianos–isoparamétricos. Quando W é igual a zero, a primeira equação é obtida. Quando W é posto igual a unidade, o esquema laplaciano isoparamétrico puro é produzido. (Este método não deve ser confundido com o mapeamento isoparamétrico discutido a seguir). O método isoparamétrico puro produz malhas bem condicionadas com apropriada reflexão da geometria do contorno no interior da malha.

Infelizmente, *Hermann* reporta que o número de iterações *Gauss–Seidel* requeridas para convergência cresce fortemente quando w aproxima seu valor ótimo da unidade.

Há assim, um balanço negativo entre qualidade da malha e eficiência computacional com este método.

O esquema laplaciano como apresentado acima, não impõe um método inerente para geração da conetividade dos elementos em uma região. Isto remove restrições na topologia da malha, mas também requer informações adicionais supridas pelo analista. Em um recente artigo, DENAYER [87] apresenta uma técnica automática para a geração da conectividade dos elementos. O método envolve um mapeamento entre uma região imaginária, definida por uma malha idealizada composta inteiramente de polígonos regulares e uma região a ser discretizada. A malha idealizada é construída a partir de informações do contorno, incluindo o número de elementos conectados a cada nó do contorno. Uma vez que a topologia da malha é criada, as localizações dos nós interiores podem ser selecionadas tais que a equação (primeira) é satisfeita.

DENAYER [87] também apresentou um método para satisfazer a equação (III.2), sem recorrer a solução de métodos iterativos. O mapeamento entre a malha idealizada e a verdadeira malha é primeiro assumido para ser da forma:

$$P = \sum_{m=1}^{M} P_m \, \Phi_m \, (u,v) \qquad \text{(III.4)}$$

onde P_m são, agora, as incógnitas vetores de posição de M nós de contorno, Φ_m são funções de forma aplicadas sobre a região a ser discretizada, e u e v são as coordenadas paramétricas da malha realizada. Se o mapeamento de uma malha idealizada para a malha real e harmônica, a equação (III.2) será satisfeita. Isto é obtido se o funcional

$$W = \int_{\Omega} \left[\left(\frac{\partial P}{\partial u} \right)^2 + \left(\frac{\partial P}{\partial v} \right)^2 \right] d\Omega \qquad \text{(III.5)}$$

é minimizado, onde Ω é a região a ser discretizada. Um procedimento variacional de elementos finitos baseado na representação da equação (III.4), é usado para extremizar a equação (III.5). Isto leva direto a matriz de equações de incógnitas nodais (coordenadas).

Os esquemas *laplacianos* têm sido usados em conjunção com outros métodos de geração de malhas, como a etapa de *Smoothing* ou suavização, na tentativa de prover a uniformidade da malha final.

III.1.4. MAPEAMENTO ISOPARAMÉTRICO

Um natural desenvolvimento do uso de mapeamento isoparamétrico para representar elementos finitos curvados é o uso do mesmo mapeamento, para malhas de todas as regiões da estrutura. O uso de mapeamento isoparamétrico para geração de malhas foi descrito por

ZIENKIEWICZ e PHILLIPS [89]. Neste método, funções de interpolação polinomial (funções de forma) são dadas como Ω_m em equação (III.4) para prover um único mapeamento entre coordenadas curvilíneas e coordenadas cartesianas. Isto representa o mapeamento entre um simples polígono (geralmente uma unidade quadrada ou triângulo) e a região real.

Os contornos da região resultantes são modelados por simples polinômios de *Lagrange*. O sistema de coordenadas curvilíneas produzido por este método provê um natural método de produzir a topologia dos elementos automaticamente. Pontos nodais podem ser localizados na interseção de coordenadas curvas constantes nas direções u e v. Conectando os nós ao longo de curvas, produz–se uma malha de elementos quadriláteros. Esta malha pode ser diagonalizada para formar elementos triangulares. Este método impõe uma restrição na topologia da malha: precisa haver um igual número de elementos ao longo de lados opostos da região. Contudo, a criação da conectividade da malha não requer qualquer esforço de parte do analista. Topologias mais gerais podem ser produzidas aumentando os dados de entrada pela seleção de uma malha geral e coordenadas curvilíneas e usando o mapeamento para gerar as coordenadas cartesianas dos pontos nodais.

Na referência [89] uma região é definida por oito pontos nodais, reproduzindo um mapeamento bilinear entre dois pares de curvas polinomiais quadráticas de curvas de contorno. Isto significa que, pontos de inflexão e inclinação descontínuos não podem ser incorporados ao contorno de uma simples região. Assim, múltiplas regiões serão requeridas para modelar um modelo complexo. Funções de forma de alta ordem podem ser usadas também para produzir contornos mais complexos. Contudo, descontinuidades de inclinação ainda não são possíveis dentro de uma única região.

Um problema adicional com mapeamento isoparamétrico é que, erros na adequada definição de curvas são introduzidos na descrição de estruturas, as quais contornos não podem ser exatamente descritos por polinômios de mesma ordem que os adotados nas funções de forma. Em contrário a estes problemas, o mapeamento isoparamétrico ainda descreve malhas bem condicionadas através de expressões de "forma fechada".

A descrição dos contornos é obtida em uma eficiente maneira para localização de um pequeno número de pontos para interpolação de *Lagrange*. Esta forma de descrição geométrica é bem satisfeita por técnicas gráficas de entrada de dados.

A essência do esquema é o uso de mapeamento curvilíneo isoparamétrico de quadriláteros. Isto, introduzido inicialmente por derivação de elementos de forma especial, permite um único mapeamento de coordenadas curvilíneas e coordenadas cartesianas. Considerando o caso particular de um quadrilátero parabólico da figura III.5, no qual as coordenadas x, y (e z) de oito nós são conhecidas, podemos escrever:

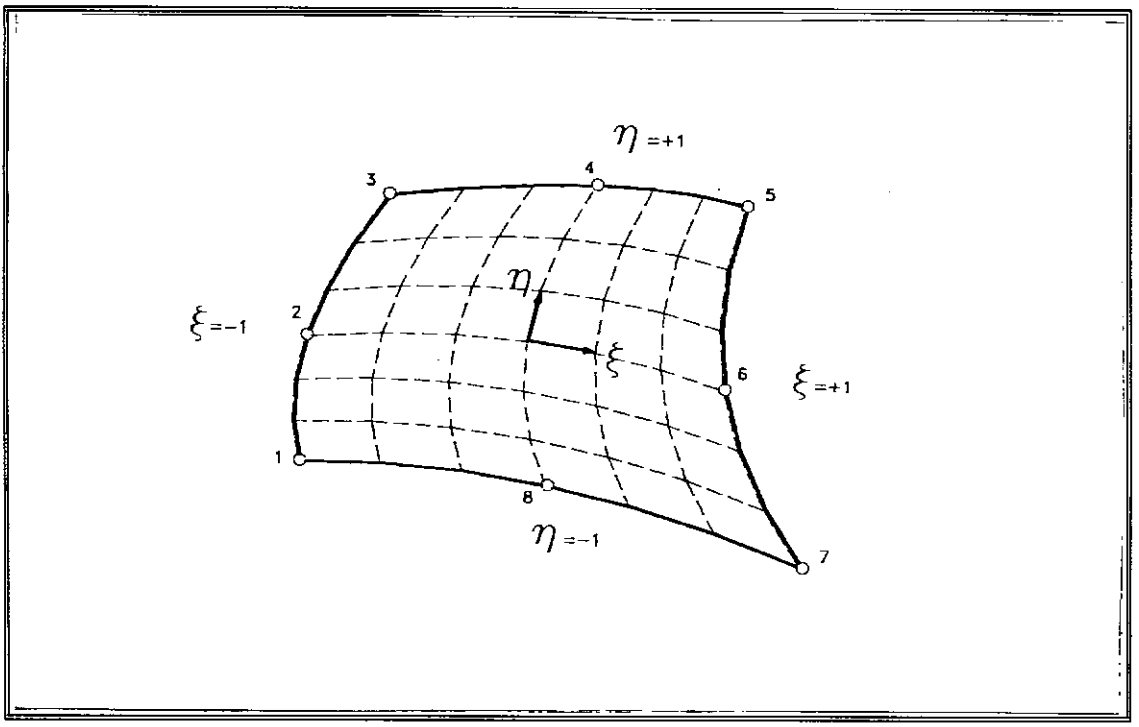


Fig III.5

$$X = \sum_{i=1}^{8} N_i \; x_i \qquad \text{(III.6.a)}$$

$$Y = \sum_{i=1}^{8} N_i \; y_i \qquad \text{(III.6.b)}$$

$$Z = \sum_{i=1}^{8} N_i \; z_i \qquad \text{(III.6.c)}$$

Onde N_i é a função de forma associada com cada nó e definida em termos de um sistema de coordenadas curvilíneo ξ e η, o qual tem valores variando entre *1* e *−1* nos dados opostos.

Para o elemento em particular, as funções de forma são dadas por:

$$N_1 = \frac{-1}{4}(1-\xi)(1-\eta)(\xi+\eta+1) \qquad \text{(III.7.a)}$$

$$N_2 = \frac{1}{2}(1-\xi)(1-\eta^2) \qquad \text{(III.7.b)}$$

etc.

Se as coordenadas dos pontos nodais são conhecidas, então as coordenadas cartesianas de algum específico ponto (ξ,η) pode ser simplesmente obtido pelas equações acima.

Se a região como um todo, na qual a malha será gerada, não pode ser descrita adequadamente por um quadrilátero na forma da figura anterior,a malha de algum refinamento poderá ser automaticamente gerada especificando:

1. Coordenadas dos oito pontos nodais;
2. O número de subdivisões nas direções ξ e η.

Os pontos nodais de uma malha precisam ser criados e numerados sem incrementos iguais de ξ e η; e as conexões dos elementos, estabelecidas por uma regra apropriada.

Assim, uma malha de qualquer complexidade pode ser gerada para uma região elementar descrita por este mínimo número de dados.

Dois adicionais pontos devem ser notados. Se igual subdivisão ao longo de um lado reto é suficiente, as coordenadas do ponto médio podem ser omitidas e estabelecidas por interpolação interna direta.

Em geral, contudo, este ponto não irá especificar a forma parabólica, mas irá também determinar a gradação da malha como mostra a figura III.6.

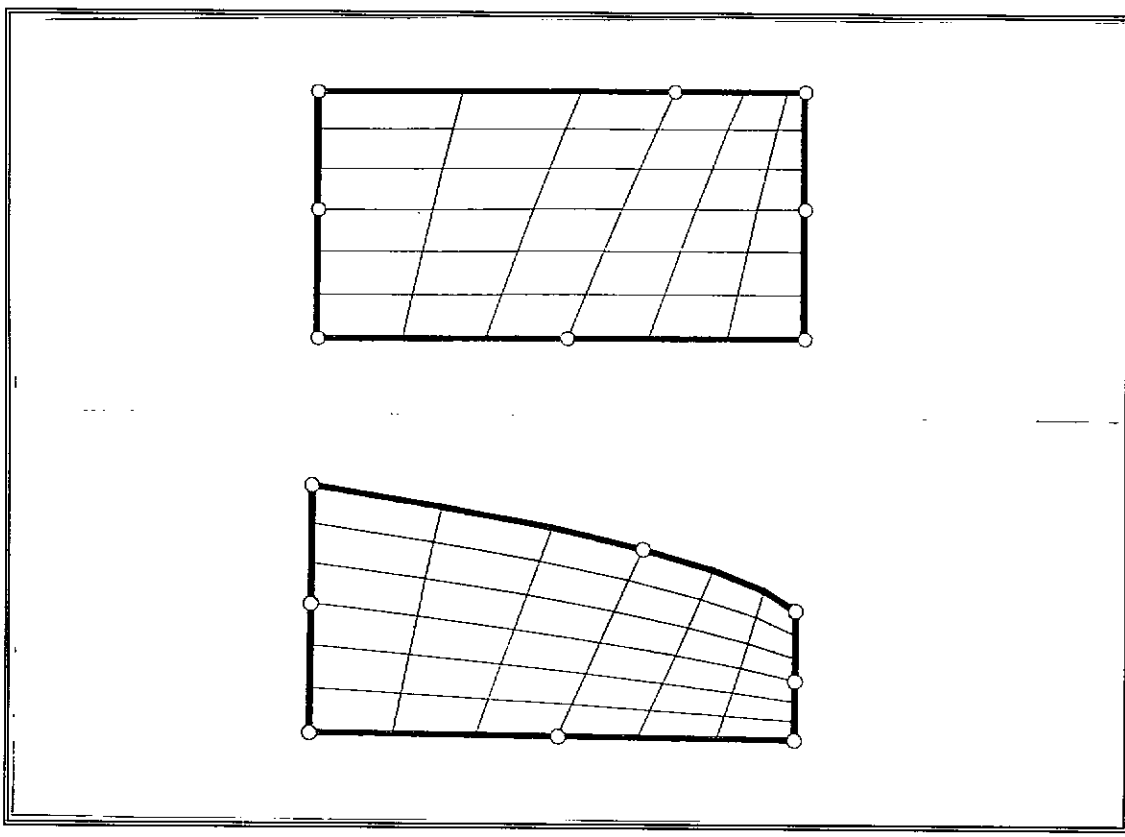


Fig III.6

A gradação pode também ser obtida por aplicação de fatores de peso a subdivisões diferentes, embora o posicionamento dos pontos nodais seja normalmente suficiente.

A forma quadrilátera elementar pode ser drasticamente distorcida, ao ponto de duas linhas de lado estarem em uma mesma reta fazendo um triângulo.

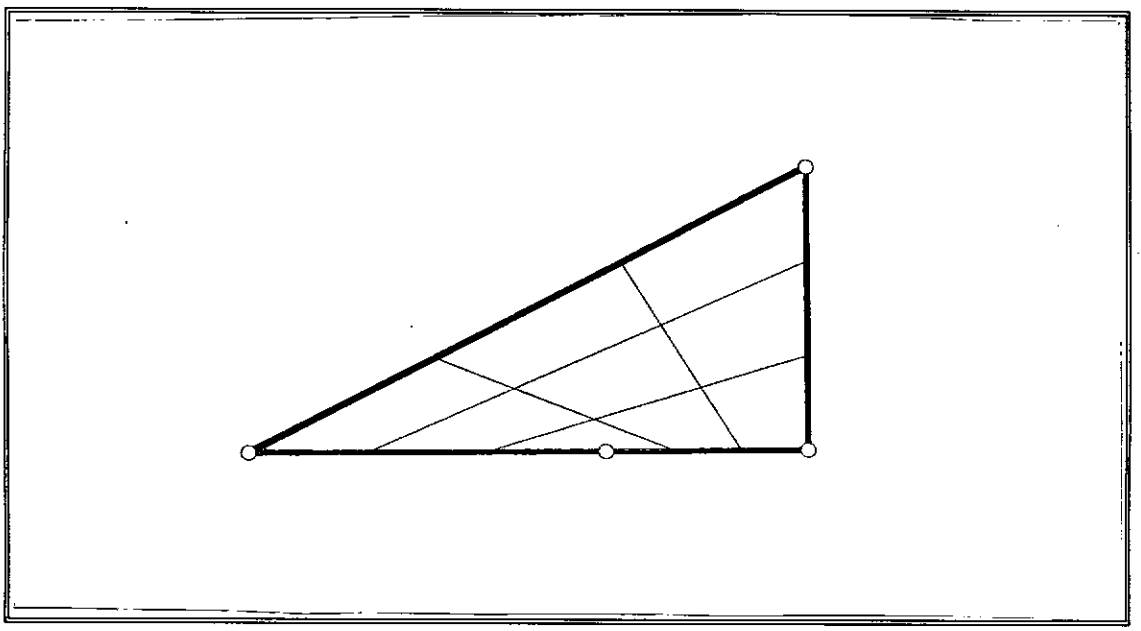


Fig III.7

Cuidado deve ser tomado para não fazer nenhum ângulo de vértice maior que 180 graus, pois uma singularidade no mapeamento pode ocorrer.

III.1.5. MAPEAMENTO TRANSFINITO

O nome transfinito deriva do fato que este método descreve uma superfície ou volume por um conjunto não–enumerável de pontos, em contraste com o método isoparamétrico, no qual a superfície mapeada coincide com a verdadeira somente em um número finito de pontos.

As técnicas de mapeamento transfinito são uma classe de métodos para estabelecer sistemas de coordenadas em domínios arbitrários. O método foi primeiramente desenvolvido por GORDON e HALL [108] para a aproximação de superfícies complexas e volumes. O método transfinito geral descreve uma superfície ou volume, o qual irá marcar uma desejada ou verdadeira, superfície ou volume em um não–enumerável número de pontos. Isto é propriamente o que dá o real sentido do termo mapeamento transfinito. Esta propriedade contrasta com o mapeamento isoparamétrico, que marca a verdadeira superfície em apenas um número de pontos finitos, isto é, os pontos usados para a interpolação. Em um caso limite de duas dimensões, o mapeamento transfinito pode ser feito para exatamente modelar todos os contornos da região, e assim nenhum erro geométrico é introduzido pelo mapeamento.

Para descrever este mapeamento é necessário introduzir o conceito de um projetor $\mathscr{P}$. Um projetor é qualquer operador idempotente, o qual mapeia uma superfície verdadeira a uma superfície aproximada, sujeita a certas restrições interpolatórias. Assim, há uma ampla variedade de possíveis projetores. Atenção será limitada a uma família de projetores baseados em um projetor muito simples, o projetor linear. O projetor linear realiza uma interpolação linear, entre duas curvas de contorno, $\Psi_1(u)$ e $\Psi_2(v)$.

$$\mathscr{P}\,[\mathrm{F}] \equiv \mathrm{P}(u,v) = (1-v)\ \Psi_1(u)\ \Psi_2(u) + v\ \Psi_2(u)\ ;$$
$$0 \leq u \leq 1\ ,\ 0 \leq v \leq 1 \qquad \text{(III.8)}$$

onde u é uma coordenada paramétrica normalizada ao longo de Ψ_1 e Ψ_2, e v é uma coordenada normalizada a qual vale zero em Ψ_1 e unidade para Ψ_2 como mostra a figura III.8.

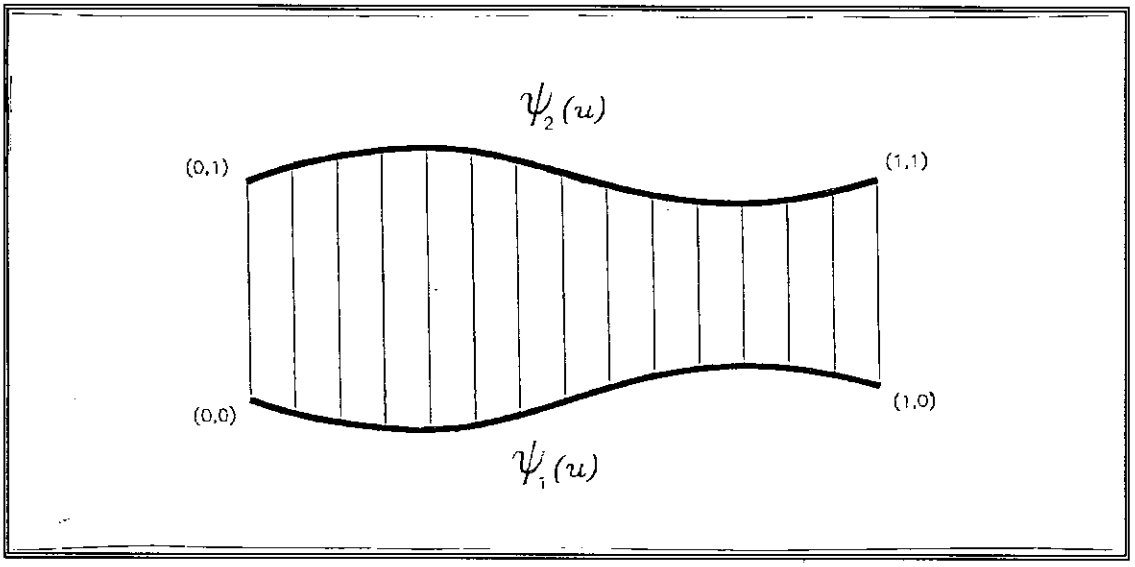


Fig III.8

Note que o contorno de $\mathcal{P}$ [F] marca F exatamente ao longo de Ψ_1 e Ψ_2 e é uma aproximação linear nos lados remanescentes. Note também que $\mathcal{P}$ é idempotente, isto é:

$$\mathcal{P}\{\ \mathcal{P}[\mathrm{F}]\ \} = \mathcal{P}\{\mathrm{F}\} \qquad \text{(III.9)}$$

Conjuntos destes projetores podem ser reunidos para formar projetores mais complexos, os quais podem marcar os contornos de F em todos os pontos. Um projetor o qual opere em uma região F definida nos contornos pelas quatro curvas $\Psi_1(u)$, $\Psi_2(u)$, $\xi_1(u)$ e $\xi_2(v)$, é mostrado na figura III.9.

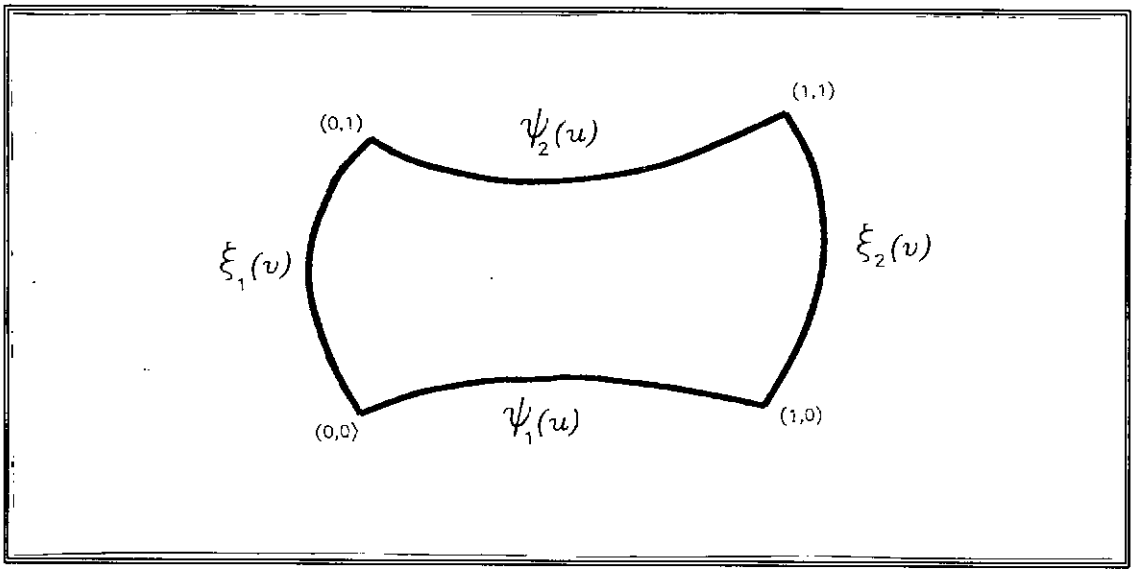


Fig III.9

Dois projetores básicos podem ser formados, um interpolado na direção u, e outro na direção v:

$$\mathcal{P}_1[F] = P_1(u,v) = (1-v)\ \Psi_1(u) + v\ \Psi_2(u) \qquad \text{(III.10.a)}$$

$$\mathcal{P}_2[F] = P_2(u,v) = (1-u)\ \xi_1(v) + u\ \xi_2(v) \qquad \text{(III.10.b)}$$

$$0 \leq u \leq 1\ ,\ 0 \leq v \leq 1$$

Estes projetores são mostrados na figura III.10 junto com o projetor produto $\mathcal{P}_1\mathcal{P}_2[F] = \mathcal{P}_1\mathcal{P}_2[F]$.

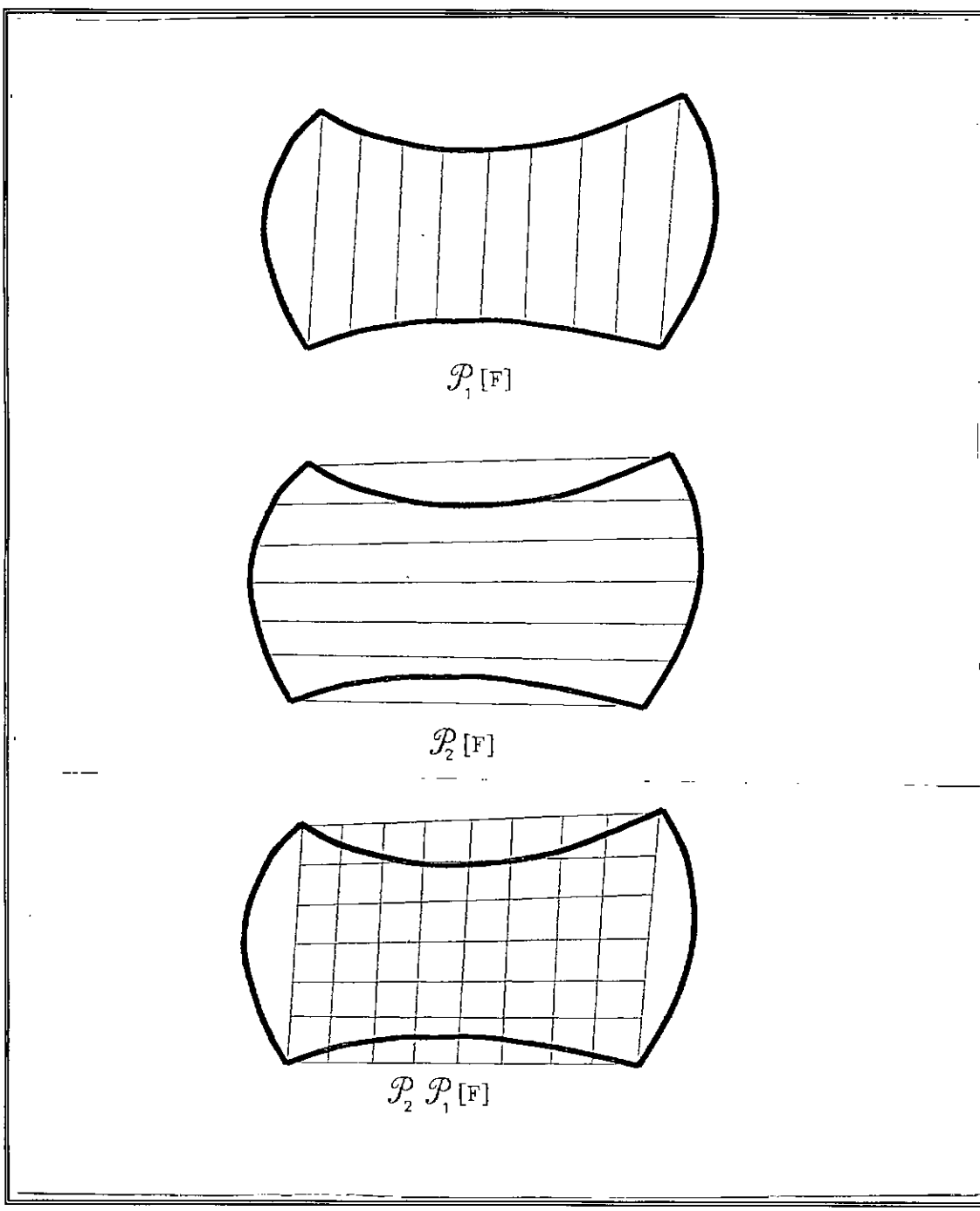


Fig III.10

Estes projetores marcam F exatamente nos quatro vértices com aproximações lineares ao longo dos quatro lados. Finalmente, uma soma bolfana é definida tal que F e marcado exatamente em todo o seu contorno.

$$(\mathscr{P}_1 \oplus \mathscr{P}_2)[F] \equiv \mathscr{P}_1[F] + \mathscr{P}_2[F] - \mathscr{P}_1\mathscr{P}_2[F] \tag{III.11.a}$$

$$(\mathscr{P}_1 \oplus \mathscr{P}_2)[F] \equiv P_b(u,v) \tag{III.11.b}$$

$$\begin{aligned}(\mathscr{P}_1 \oplus \mathscr{P}_2)[F] \equiv\ & (1-v)\ \Psi_1(u) + v\ \Psi_2(u) + (1-u)\ \xi_1(v) + \\ & u\ \xi_2(v) - (1-u)\ (1-v)\ F(0,0) - \\ & (1-u)\ v\ F(0,1) - u\ v\ F(1,1) - u\ (1-v)\ F(1,0)\end{aligned} \tag{III.11.c}$$

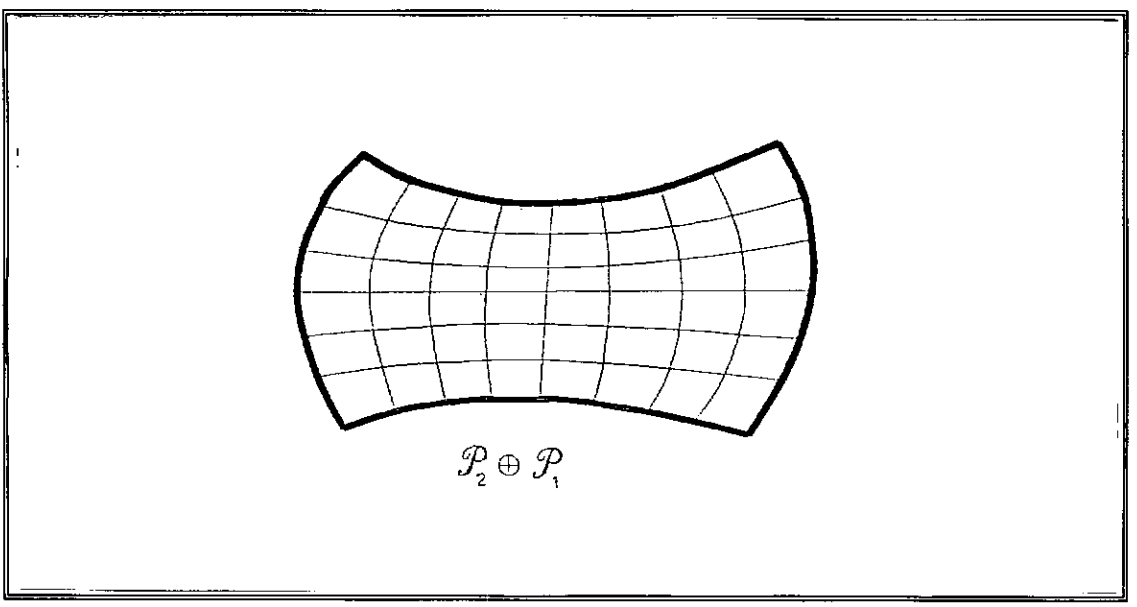


Fig III.11

Esse projetor representa um sistema de coordenadas curvilíneas criadas pelo mapeamento de um quadrado unitário em F. Este projetor pode ser chamado de interpolador transfinito bilinear de *Lagrange* em F. Se as curvas de contorno são definidas por polinômios de *Lagrange*, o mapeamento isoparamétrico é obtido como um caso especial da família mapeamento transfinito. Interpoladores de alta ordem podem ser usados para forçar as coordenadas curvas a passar através de específicas curvas no interior de F. GORDON e HALL [108] apresentam o desenvolvimento destas técnicas.

Outro projetor, é o baseado no mapeamento de um triângulo unitário em uma região definida por três curvas de contorno. Isto pode ser obtido usando um interpolador trilinearmente reunido como descrito por BARNHILL et alii [109] usando um sistema cartesiano em um triângulo unitário. Esta notação preserva a cíclica natureza do mapeamento em equações resultantes.

Quando curvas de contorno muito distorcidas são usadas para definir uma região, um problema de sobreposição pode ocorrer. Este problema acontece quando as linhas da malha de elementos finitos cortam a região dos contornos da região. Uma opção é quebrar a região altamente distorcida em duas ou mais regiões com geometrias mais regulares. Este método não requer o uso de mapeamento de alta ordem.

III.1.6. MAPEAMENTO TRANSFINITO DISCRETO

Duas grandes classes podem ser identificadas para a representação apropriada dos contornos de uma região: forma contínua e discreta. A forma contínua representa o vetor de posição para um contorno curvo como uma função de algumas coordenadas paramétricas. A parametrização pode ser em qualquer número de formas, tais como: comprimento de arco, radianos, coordenadas cartesianas, etc... A representação contínua permite a posição curva ser avaliada em todos os pontos ao longo da curva.

A representação discreta consiste de lista finita de pontos localizados na curva, com uma única coordenada associada com cada ponto da lista. A posição da curva pode ser avaliada apenas nos pontos contidos na lista e é indefinida fora.

A forma de representação discreta é inteiramente geral e pode ser usada em qualquer forma de curva. A representação contínua, dá uma mais completa definição da curva, mas não é geral. Por exemplo, diferentes formas de representação são requeridas para descrever, exatamente, arcos circulares e curvas polinomiais, enquanto não existe exata representação matemática para algumas curvas. Contudo, há vários métodos, notavelmente modelos *spline*, os quais provêem formas gerais que são apropriadas para a aproximada descrição de qualquer curva.

Nessa discussão será assumido que a interseção das curvas de coordenadas constantes no mapeamento transfinito será usada para estabelecer a localização de pontos nodais para a malha de elementos finitos. Uma família finita de curvas em cada direção coordenada é identificada no mapeamento bilinear, curvas de direção u são definidas por coordenadas

constantes v e vice–versa. Assim, uma lista de n valores de constante v_i, $i=1,..,n$, define n curvas correndo na direção u. Uma lista similar de m valores de u_j, $j=1,..,m$, define um conjunto de m curvas correndo na direção v. Examinando a equação (7), verificamos que para avaliar a localização de todos nm pontos de interseção dessas curvas, as curvas de contorno apenas necessitam ser avaliadas nos $2(n+m)$ pontos:

$$\{ \xi_1(v_i),\ \xi_2(v_i) \} \quad i=1,n \tag{III.12.a}$$

$$\{ \Psi_1(u_j),\ \Psi_2(u_j) \} \quad j=1,m \tag{III.12.b}$$

Para a eficiente avaliação de todos os pontos de interseção nm, é claro que, $2(n+m)$ curvas de contorno precisam ser avaliadas primeiro e armazenadas para evitar cálculos redundantes. Esta primeira avaliação é, em efeito, uma conversão da original curva de representação do contorno para forma discreta. Similares resultados são obtidos para mapeamento trilinear, onde uma malha triangular de $n(n+1)/2$ pontos de interseção podem ser avaliados baseados em $3n$ ponto da curva de contorno.

$$\{ \Psi(u_i),\ \xi(v_i),\ \eta(w_i)\ ;\quad i=1,n \} \tag{III.13}$$

Se as curvas de controno são arbitradas para serem definidas em forma discreta, é possível escrever subrotinas altamente eficientes e totalmente gerais para realizar mapeamento transfinito. Complexas curvas, criadas pela concatenação de uma série de curvas simples, podem ser usadas

sem adicionar complexidade às rotinas de mapeamento. Contornos arbitrários que não têm representação matemática exata, também não apresentam dificuldades na forma discreta.

As representações curvas discretas requerem que uma coordenada normalizada seja assumida para cada ponto. Este processo pode ser automatizado assumindo a coordenada $i/n-1$ para o i–ésimo ponto na curva contendo n pontos. Note que não há necessidade de nenhuma relação entre este sistema de coordenadas e qualquer quantidade geométrica, tal como comprimento de arco. Todavia, algum relacionamento pode ser estabelecido se os pontos usados na discretização são escolhidos apropriadamente.

Um aparente inconveniente da forma discreta de representação de curvas é a grande quantidade de dados requeridos para descrever um contorno curvo. Esta inconveniência pode ser eliminada do usuário pela construção de um conjunto de geradores de curva que, automaticamente, criam descrição discreta de curvas baseados em vários modelos contínuos.

Bases polinomiais de alta ordem ou *splines* são necessárias para modelar superfícies curvas adequadamente. Descrições *lagrangianas* têm sido usadas por serem mais fáceis de manipular em um ambiente gráfico interativo que descrições *hermitianas*. Contudo, descrições *hermitianas* podem ser usadas onde continuidade é requerida em regiões adjacentes.

III.1.7. TRIANGULARIZAÇÂO

Descreve–se, a seguir, dois tipos de triangularização automática.

No primeiro, o contorno do domínio é discretizado por um número de pontos compatível com a densidade da malha que se deseja, obtendo–se, desta forma, um contorno poligonal. A partir deste ponto, inicia–se a fase de triangularização. Chama–se esta técnica de *Cutside–in*, uma vez que o procedimento parte do contorno em direção ao interior.

Uma destas técnicas consiste em remover, primeiramente, todos os vértices do polígono do contorno que formam ângulos menores que 90 graus, resultando em um triângulo para cada vértice removido. Em seguida, selecionam–se os vértices com ângulos menores que 180 graus, sendo estes substituídos por dois novos nós no interior da região. As coordenadas destes nós são calculadas com base nos nós vizinhos do vértice removido. A área removida juntamente com este vértice é, então, dividida em dois triângulos. Novamente, se algum dos três novos vértices formar um ângulo menor que 90 graus, deve ser removido, dando origem a outro triângulo. Este processo é repetido até que o polígono contenha somente três vértices, o que define o último triângulo.

No outro tipo de triangularização, os nós do interior do domínio são fornecidos juntamente com os nós do contorno. Em seguida, estes nós são interligados segundo algum critério de triangularização. Ficam, então, definidas duas fases distintas: a fase de geração dos nós e outra de triangularização. Dentre as técnicas de geração de nós, podemos citar o trabalho de CAVENDISH [119]. Das técnicas de triangularização, a de Delaunay é talvez a mais importante, pois maximiza a soma dos menores ângulos de cada triângulo da malha.

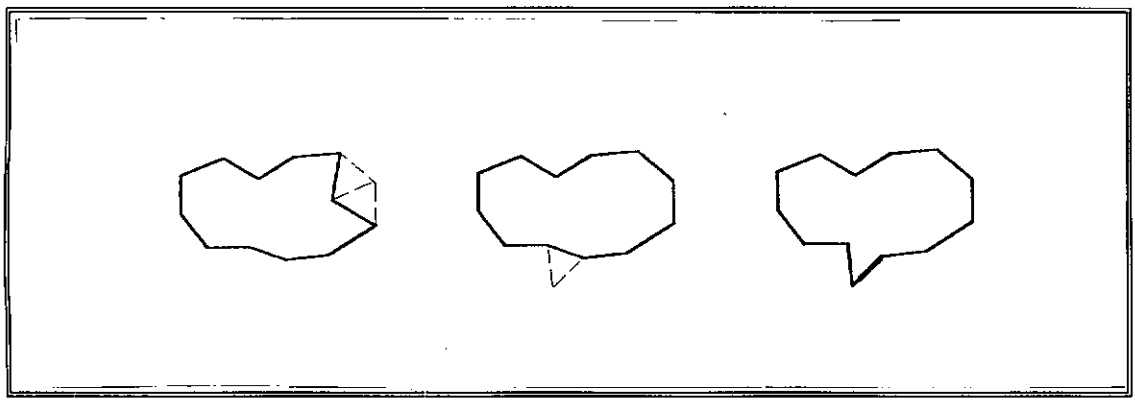


Fig III.12

Dado um domínio Ω, podemos proceder o problema de geração da malha a partir de um ponto qualquer. Alguns pesquisadores, no desenvolvimento de algoritmos nos últimos anos, têm se dirigido recursivamente à construção de discretizações gerais, objetivando elementos bem condicionados. Serão descritas brevemente algumas das idéias essenciais e resultados.

Vamos começar ignorando a questão de gradação apropriada da malha e, ao invés disso, dirigir a atenção para o problema puramente geométrico de definição da triangulação a partir de um específico contorno $\partial\Omega$. O objetivo é preencher o interior Ω_0 com triângulos próximos da forma equilátera. Por conveniência e sem perder a generalidade, consideramos elementos com lados retos e um contorno poligonal $\partial\Omega$.

Dado o polígono de contorno, o primeiro triângulo da discretização pode ser construído por introdução de um novo nó ou pela união de dois nós do contorno.

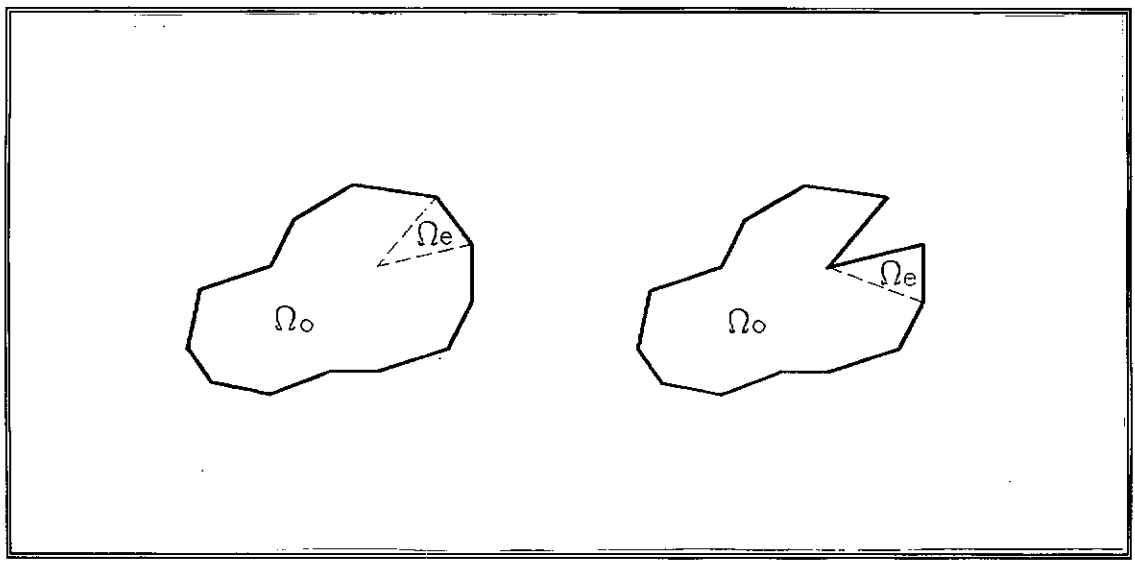


Fig III.13

Tendo construído o dito triângulo Ω_e, consideramos o novo, menor domínio $\Omega_1=\Omega_o-\Omega_e$ e construímos o próximo triângulo. Este procedimento simples pode ser repetido até o domínio reduzir–se ao triângulo final e a discretização se completa (SADEK [85]).

Uma das desvantagens deste esquema é que pode produzir malhas, nas quais a gradação pode não ser a desejada ou bem provida para o problema a ser solucionado. De fato, a função de força F, coeficientes e condições de contorno do problema de valor de contorno, são importantes na determinação da natureza e gradação de uma malha apropriada para um dado problema, mas são ignorados no presente esquema. Este ponto é de menor significância se uma malha inicial estiver sendo gerada por um algoritmo de refinamento adaptativo, desde que os posteriores refinamentos locais da malha levarão a uma gradação apropriada. Outra maneira de obter–se tal gradação é a decomposição do domínio em sub–regiões e imposição a cada sub–região, de triangulações com gradações diferentes. Isto adapta–se muito bem a processos de geração de malha interativos.

O problema relatado acima desfaz–se quando é dado um conjunto de pontos nodais e quando uma triangularização estiver por ser construída. O conjunto de pontos nodais pode ser gerado, na prática, usando–se um digitalizador e o algoritmo é, então, projetado para construir uniões na triangularização.

Consideraremos um conjunto de pontos nodais (x_i,y_i) $i=1,2,...,N$ no plano. A película convexa do conjunto de pontos é a menor região convexa que contém os pontos. Se o domínio não é convexo, especificamos o contorno.

Em uma maneira similar a este algoritmo, o triângulo inicial pode ser construído pela seleção de um ponto de conjunto e conectando–o por "arcos", a algum outro par de pontos para formar um triângulo. Sucessivamente, triângulos são gerados enquanto se assume que nenhum arco corta outro. Esta estratégia irá, contudo, geralmente levar a malhas com elementos de formato pobre. Melhoramentos simples deste esquema são baseados na união de nós próximos com arcos. Já neste caso, alguns elementos esbeltos serão gerados se os pontos próximos forem quase colineares. Podemos verificar os elementos e mínimo dos triângulos gerados. Uma modificação do algoritmo consistiria, então, em se considerar algum par melhor de triângulos. Assim sendo, uma sequência de trocas poderá ser feita para melhorar a malha. Sendo E, F e V o número de lados, faces e vértices, respectivamente, e sendo V_b o número de vértices no contorno, é fácil mostrar por indução que:

$$F = 2V - V_b - 2 \qquad \text{(III.14.a)}$$

$$E = F + V - 1 = 3V - V_b - 3 \qquad \text{(III.14.b)}$$

$$V-2 \leq F \leq 2V-5$$

$$2V-3 \leq E \leq 3V-6$$

É evidente que a troca das diagonais não troca os parâmetros V, E e F. Além disso, por indução pode–se mostrar que, todas as possíveis triangularizações do domínio podem ser geradas pela aplicação da mudança da diagonal de uma dada triangularização (RENKA [125]).

A base da escolha da diagonal é o critério de máximo e mínimo ângulo. Mais formalmente, seleciona–se o par de triângulos que maximiza o mínimo dos seis ângulos interiores quando o quadrilátero é estritamente convexo. Se o quadrilátero não for convexo, a troca de diagonal não será feita se o teste de ângulo máximo e mínimo for positivo. Um segundo critério usado é baseado em um teste de circum–círculo, onde se seleciona um par de triângulos, os quais um circum–círculo não contém os vértices remanescentes. Os critérios do círculo e os critérios de máximos e mínimos ângulos são equivalentes.

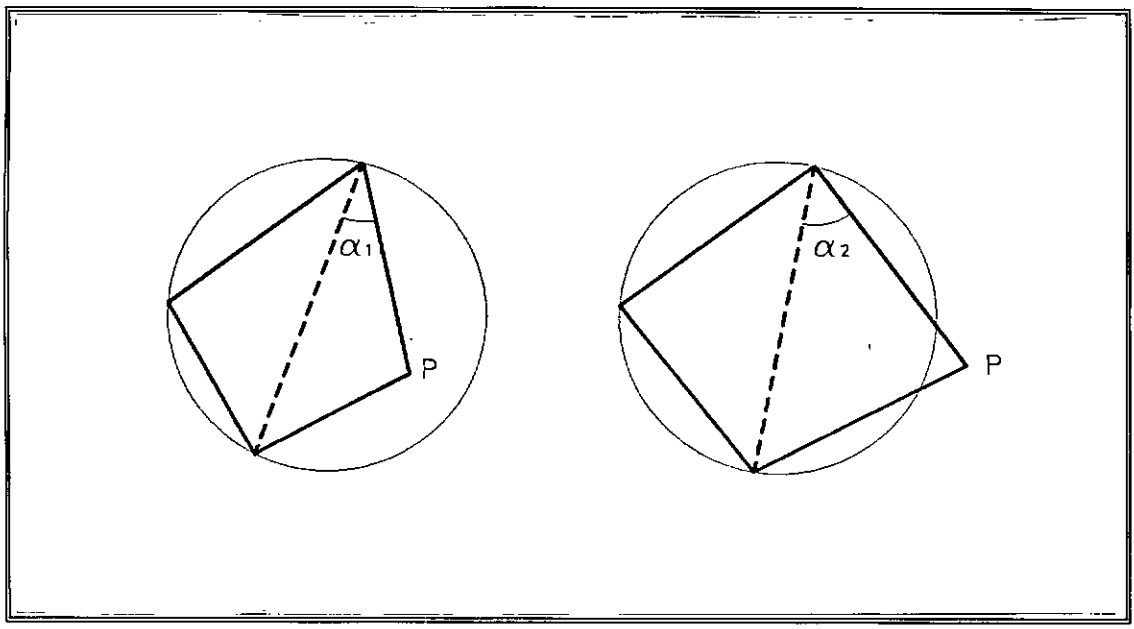


Fig III.14

Os testes provêem um procedimento para a geração de uma boa triangulação. Uma triangulação ótima (Thiessen ou Delaunay) tem todos os arcos localmente ótimos.

Um algoritmo, para iterativamente construir a triangulação, pode ser como o seguinte:

1. primeiro, constrói–se uma triangularização inicial de Thiessen T_j dos primeiros j nós não colineares:

(a) selecionar um subconjunto de j nós $3 \leq j \leq V$, não todos colineares. Nós $1,2,...,(j-1)$ definem um segmento da linha S com dois nós como extremos.

(b) ordenar os $j-1$ nós pela distância do primeiro extremo de S.

(c) conectar os primeiros $j-1$ nós, fazendo um par de nós N_1 e N_2 adjacentes se, e somente se, N_1 imediatamente precede ou segue N_2 na ordenação da segunda etapa.

(d) conectar j para cada um dos primeiros $j-1$ nós.

2. Iterativamente, para $k=j+1,..,N$ construir uma triangularização Thiessen T_k dos nós $1,2,..,k$:

(a) achar um triângulo ou arco do contorno de T_{k-1} que contenha k nós ou um par de nós do contorno, adjacentes a k que possam ser conectados a k por um segmento de reta que nem intercepte, nem ultrapasse arcos de T_{k-1}, se k é exterior a T_{k-1}.

(b) se k está em um contorno de arco, conecte-o aos extremos. Se k está contido em um triângulo, conecte-o aos vértices. De outra maneira, conecte k a todos os nós do contorno adjacentes a k.

(c) usar a troca da diagonal para melhorar a malha.

Maiores detalhes deste algoritmo é um eficiente esquema de armazenamento são descritos por RENKA [125]. Representações de malhas de triângulos necessitam de $7N$ locações de armazenamento em adição às $2N$ locações para as coordenadas nodais. O total de operações está estimado em aproximadamente $O(N^{1.6})$.

III.1.8. MODELAGEM GEOMÉTRICA

Modelagem geométrica é o processo de criação, edição e representação de objetos sólidos em um computador. Um sistema de modelagem geométrica é um programa que incorpora estas funções e provê uma base para diferentes programas de aplicação que operem nestes objetos modelados. Geração de malhas de elementos finitos é uma destas aplicações.

A emergência de modelagem geométrica, como uma tecnologia viável, é um evento recente. REQUICHA e VOELCKER [42] apresentam um apanhado histórico neste campo, no qual também definiram seu potencial. REQUICHA [43] apresenta uma discussão na teoria de modelagem geométrica, incluindo esquemas de representações de superfícies e suas propriedades.

Os dois mais populares esquemas são B–rep (Boundary Representation) ou representação do contorno e CSG (Construtive Solid Geometry) ou Geometria Sólida Construtiva.

Na CSG, os objetos são definidos como uma combinação de primitivas, tais como blocos retangulares e cilindros, afetados pelo uso de um conjunto de operadores regularizados. A combinação é expressa como uma árvore binária, comumente conhecida como uma árvore CSG, na qual os nós não terminais são operadores booleanos e os demais primitivas.

Dois procedimentos para geração automática de malhas baseados em modelagem geométrica, foram reportados recentemente. WORDENWEBER [110] usa o esquema de B–REP com operadores de Euler.

YERRY e SHEPHARD [111] usam um procedimento bastante diferente, baseado em *Quadtree* ou árvore de quatro ramos para objetos em duas dimensões e *Octree*, árvore de oito ramos para três dimensões.

O objeto é fechado em uma célula quadrada, a qual é recursivamente subdividida em subcélulas. As subcélulas que estão dentro do objeto são retidas e contribuem para os elementos da malha; as de fora são descartadas e as subcélulas que interceptam o contorno são subdivididas. A subdivisão continua até uma certa resolução ser obtida.

Em ambos os procedimentos, a geração de um elemento é afetada pelo objeto como um todo.

No trabalho de WORDENWEBER [110], a topologia do objeto é o fator de controle, enquanto que no Quadtree/Octree a forma dos elementos na superfície depende da orietanção da superfície com respeito aos eixos do esquema de subdivisão .

A perda de controle na geometria de um elemento finito pode ser compensado, em ambos os casos, pela introdução em um estágio subsequente de procedimentos interativos para relocação dos nós de modo a otimizar a forma dos elementos.

III.2. SUAVIZAÇÃO

Em análises com elementos finitos, a geometria de um elemento exerce um efeito significante na precisão dos resultados. As regras na geometria de um elemento são largamente baseados na experiência prática com sistemas de análises, embora possam ser formulados a partir de princípios teóricos, na maioria dos casos. Especificamente, restrições são

postas no comprimento e proporção entre lados e ângulos de um elemento. Não nos deteremos em específicos valores destes atributos, uma vez que eles variam de sistema para sistema e também, em diferentes aplicações para o mesmo sistema.

Os comprimentos dos lados de um elemento dependem da densidade da malha.

Um "bom elemento" é definido como um elemento com lados e ângulos dentro dos respectivos "bons" campos, isto é, máximo e mínimo comprimento e máximo e mínimo ângulos. A boa malha é uma que leve ao mínimo, os erros nos resultados da análise.

Todavia, desde que os resultados da análise só podem ser obtidos após a geração de uma malha, não é prático o uso desta definição na determinação da qualidade da malha. Ao invés disto, uma boa malha é definida por ser uma que contenha apenas bons elementos.

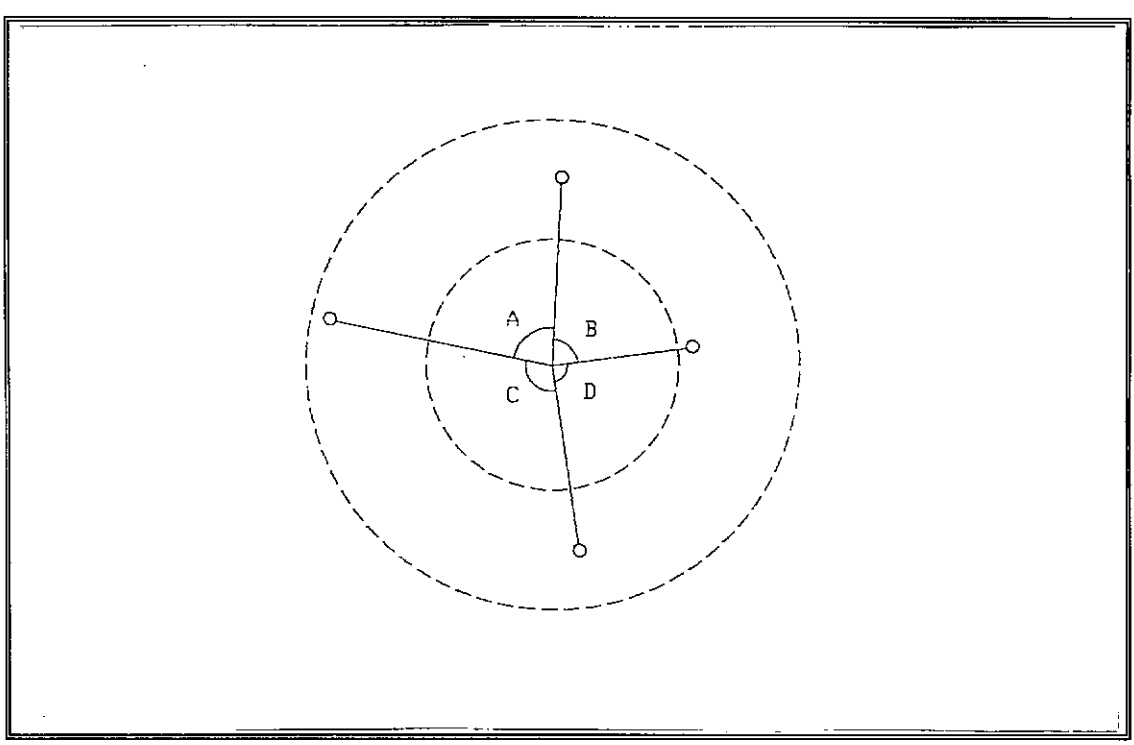


Fig III.15

Para todo ponto P em um conjunto bem distribuído:

(1) Nenhum outro ponto no conjunto pode estar dentro de uma distância de P menor que a mínima distância;

(2) A união de P a todos dentro de uma distância máxima de P, o ângulo entre lados adjacentes precisa ser menor que o ângulo máximo (figura acima).

Estas regras auxiliam a explícita computação para uma boa distribuição. Um algoritmo necessita apenas examinar cada ponto ao redor e verificar que estes satisfazem as condições. Contudo, precisa ser notado que:

(A) A regra 1 pode ser violada se um objeto contiver contornos menores que o comprimento mínimo. Isto não pode ser retificado pelo algoritmo, uma vez que este não tem controle sobre a geometria do objeto.

(B) Nem todos os conjuntos de pontos que satisfazem estas regras levam a boas malhas. Bons elementos não dependem somente da definição de atributos a pontos isolados, mas a todos os pontos nodais.

As técnicas de suavização ou *smoothing* são assim denominadas em virtude de serem frequentemente utilizadas em conjunto com outros métodos de geração, com a finalidade de melhorar a uniformidade dos elementos.

O esquema *laplaciano* consiste em posicionar os I nós de uma malha segundo as equações

$$X_i = \frac{1}{4} \left(x_{i1} + x_{i2} + x_{i3} + x_{i4} \right) , \qquad j=1,..,I \tag{III.15.a}$$

$$Y_i = \frac{1}{4} \left(y_{i1} + y_{i2} + y_{i3} + y_{i4} \right) , \qquad j=1,..,I \tag{III.15.b}$$

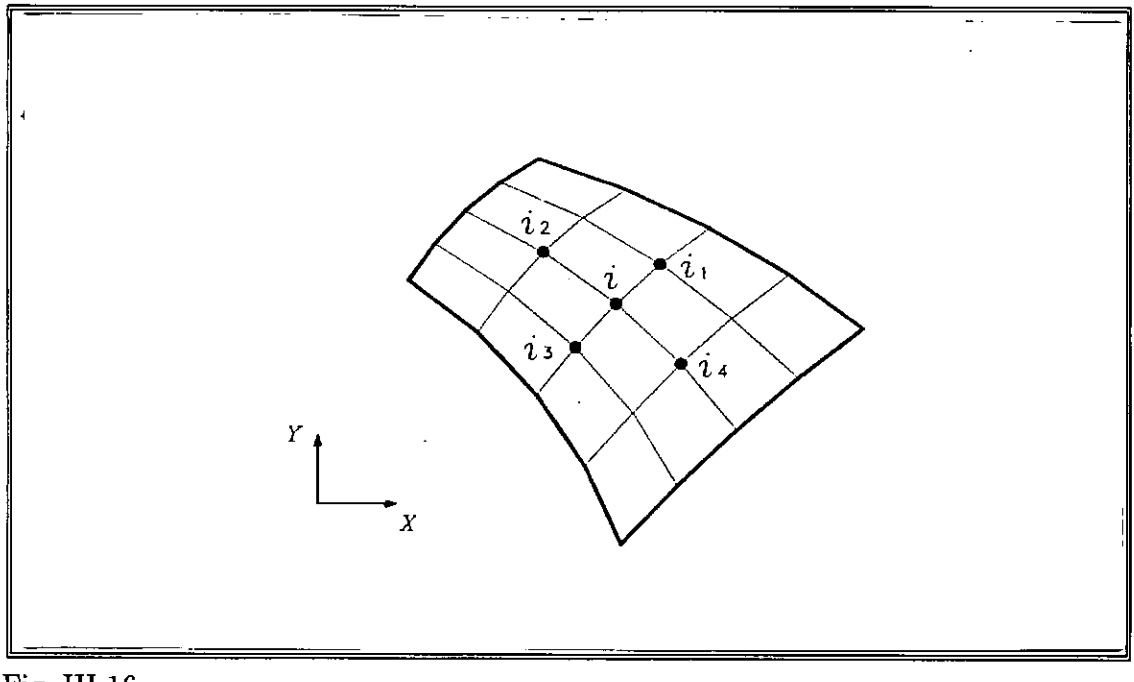


Fig III.16

A seguinte modificação permite que sejam considerados N_i elementos incidentes no nó i.

$$X_i = \frac{1}{2N_i} \sum_{n=1}^{N_i} (x_{nj} + x_{nl}) , \qquad i=1,..,I \qquad \text{(III.16.a)}$$

$$Y_i = \frac{1}{2N_i} \sum_{n=1}^{N_i} (y_{nj} + y_{nl}) , \qquad i=1,..,I \qquad \text{(III.16.b)}$$

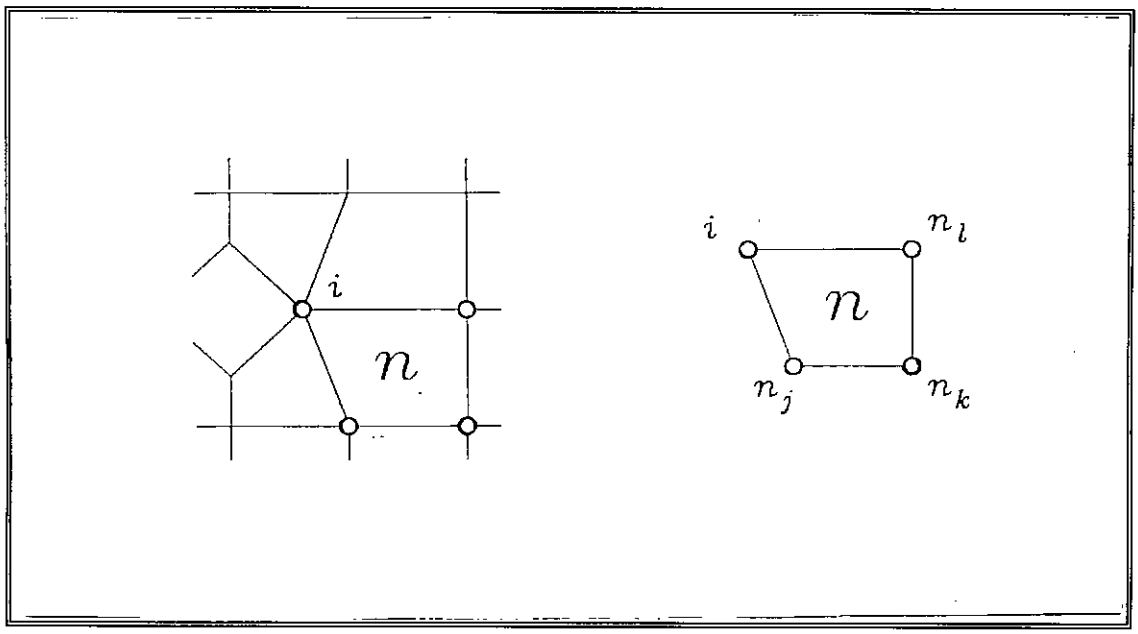


Fig III.17

Para malha, retangulares as equações (III.15) são idênticas às (III.16).

As equações (III.15) e (III.16) constituem um sistema de equações lineares e são normalmente resolvidas através de métodos iterativos, tais como *Gauss–Seidel* e *Jacob*.

Este esquema de geração conduz a malhas de razoável uniformidade, mas não é suficientemente sensível a informações contidas nos nós situados no contorno, no que diz respeito ao espaçamento da malha e curvatura do contorno.

A figura ilustra um caso típico deste problema.

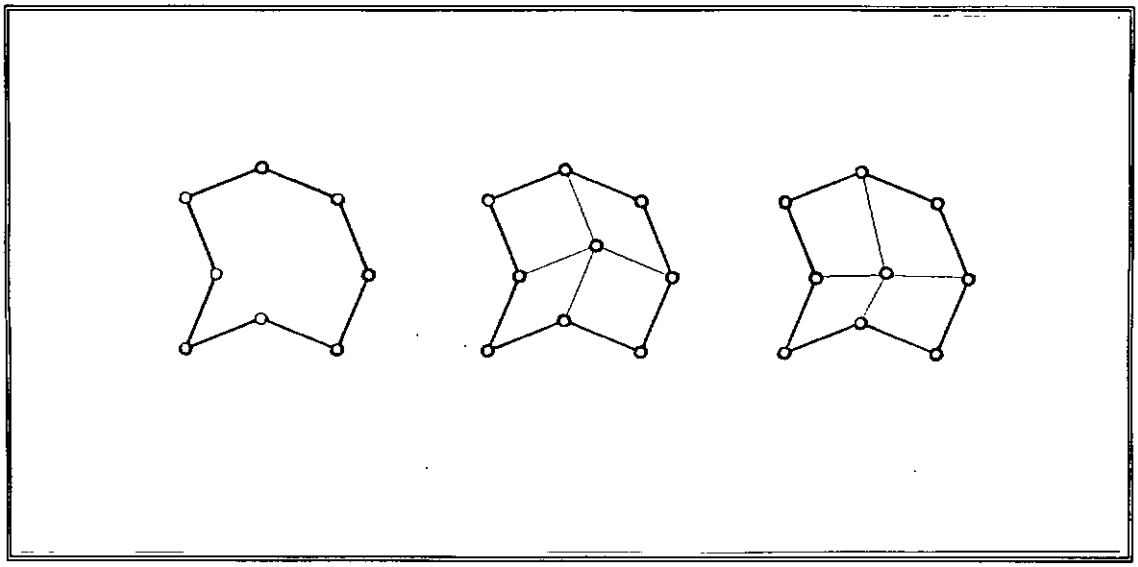


Fig III.18

Para melhorar o efeito da propagação das curvas do controno, HERMANN [107] apresentou uma alternativa para este esquema. Observando que as coordenadas do nó central de um elemento isoparamétrico de 8 nós são dadas por

$$X_i = \frac{1}{4} \left[2(x_{i1}+x_{i2}+x_{i3}+x_{i4}) - (x_{i5}+x_{i6}+x_{i7}+x_{i8}) \right] \qquad \text{(III.17.a)}$$

$$Y_i = \frac{1}{4} \left[2(y_{i1}+y_{i2}+y_{i3}+y_{i4}) - (y_{i5}+y_{i6}+y_{i7}+y_{i8}) \right] \qquad \text{(III.17.b)}$$

e generalizado para o caso de malhas não retangulares, por

$$X_i = \frac{1}{N_i \ (2-W)} \sum_{n=1}^{N_i} (x_{nj} + x_{nl} - W\, x_{nk}) \qquad \text{(III.18.a)}$$

$$Y_i = \frac{1}{N_i \ (2-W)} \sum_{n=1}^{N_i} (y_{nj} + y_{nl} - W\, y_{nk}) \qquad \text{(III.18.b)}$$

Para $W=1$, as equações acima correspondem ao esquema isoparamétrico de suavização, e se $W=0$ tem-se o esquema laplaciano.

Estas equações também são resolvidas via métodos iterativos.

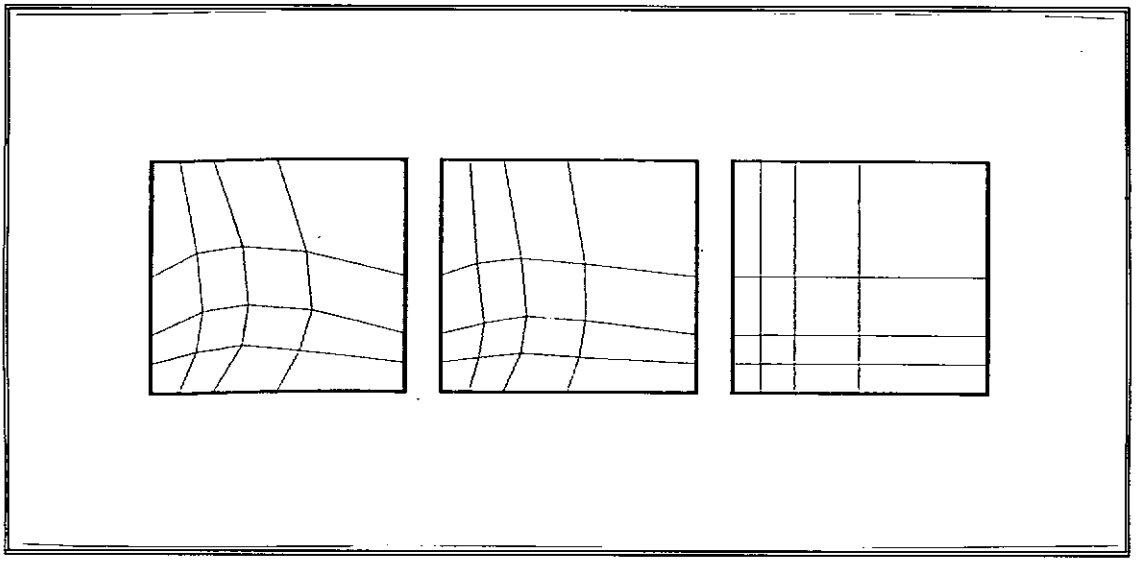


Fig III.19

III.3. DIAGONALIZAÇÃO DE ELEMENTOS QUADRILÁTEROS

Em muitos dos métodos apresentados neste capítulo, a geração de malhas é realizada para elementos quadriláteros, para o obtenção de malhas com elementos triangulares, podendo–se utilizar técnicas de diagonalização ou divisão dos elementos quadiláteros.

Para dividir um quadrilátero em duas áreas triangulares, dois requisitos devem ser tomados em consideração (COHEN [84]):

1. os elementos resultantes são estes de melhor condicionamento. Isto é obtido pela escolha da menor diagonal.
2. as áreas dos triângulos são próximas uma da outra.

Estes requisitos são algumas vezes contraditórios.

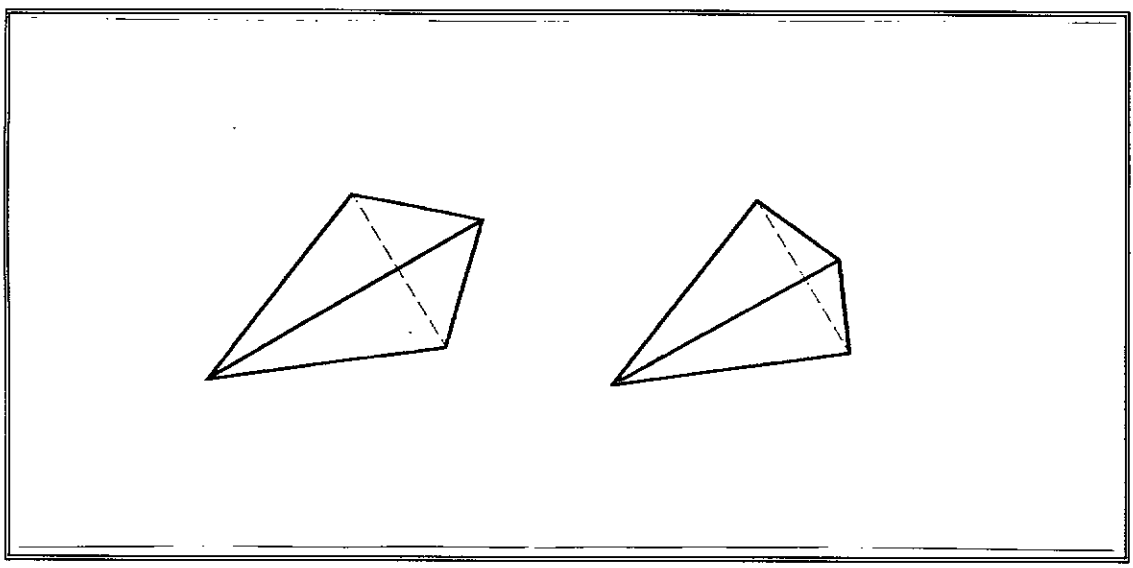


Fig III.20

Para explicar como a subdivisão é selecionada, considere a área quadrilátera da figura:

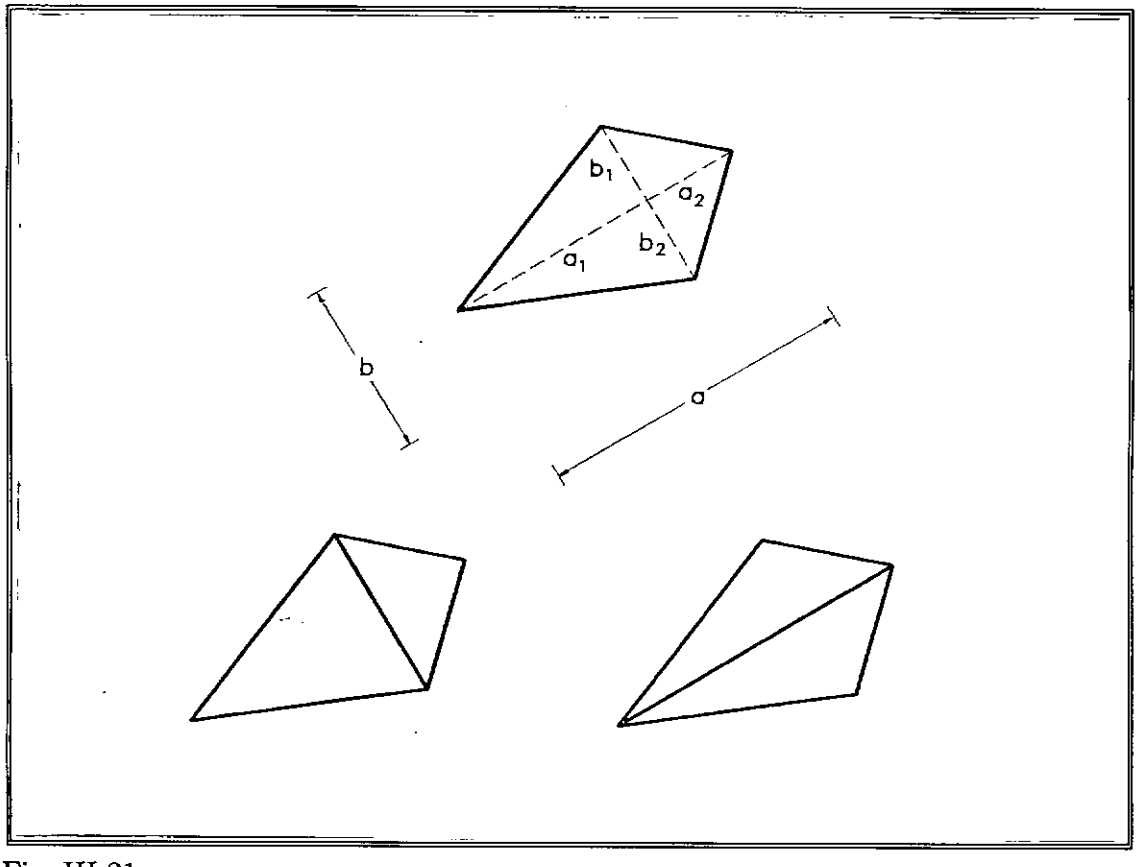


Fig III.21

Para a subdivisão A um fator R é dado por

$$R = \frac{a}{b} + \frac{b_1/b_2}{a_1/a_2} \qquad \text{(III.19)}$$

Similarmente, o fator R é determinado para a subdivisão B por

$$R = \frac{b}{a} + \frac{a_1/a_2}{b_1/b_2} \qquad \text{(III.20)}$$

O primeiro termo de R dá preferência à subdivisão, a qual corresponde a menor diagonal. Contudo, o segundo termo dá preferência à subdivisão que produz duas áreas próximas. Por isto, a subdivisão com menor R é considerada a que satisfaz um dos requisitos e desvia menos do outro.

CAPÍTULO IV

Neste capítulo, são apresentadas as características básicas de utilização e programação, a estrutura de dados interna e listadas algumas das aplicações do **AutoCAD**, com a intenção de demonstrar a potencialidade e justificar a adoção de tal sistema gráfico como base do desenvolvimento do pré–processador gráfico **AutoMESH**.

IV.1. HISTÓRICO

O primeira versão do **AutoCAD** foi desenvolvida em 1982. Nesta época existiam algumas soluções a nível de CAD, porém, disponíveis a um número extremamente limitado de usuários. Empresas como **Intergraph**, **ComputerVision**, **IBM**, etc. e programas como **CADAM** estavam disponíveis no mercado. Estes sistemas forneciam elementos gráficos que eram utilizados na criação e edição de desenhos com o auxílio do computador. Para a grande massa de usuários que necessitavam de um sistema CAD, entretanto, esta era uma solução parcial devido ao seu elevado custo, somente acessível a empresas de grande porte.

Embora as soluções existentes, baseadas em *mainframes* ou minicomputadores apresentassem velocidade, versatilidade e precisão, o investimento necessário era da ordem de centenas de milhares de dólares, não sendo raro atingirem a casa do milhão de dólares. Além disso, a complexidade dos sistemas dificultavam o aprendizado por parte dos usuários convencionais (desenhistas e projetistas). A disponibilidade de *hardware* opcional era inexistente, fazendo com que o usuário adquirisse uma "caixa preta" com pouquíssima capacidade de adaptabilidade. Finalmente, existiam conflitos quanto a prioridade de uso, devido a problemas de planejamento e, durante o uso intensivo do sistema, degradação da velocidade de resposta.

Em 1982, o aparecimento do IBM/PC faz surgir um *harware* em potencial para a solução das necessidades do usuário de CAD.

Um grupo de 13 especialistas em computação gráfica em computadores de grande porte, resolveu dedicar–se a tarefa de desenvolver um CAD baseado em PC e em outubro de 1982 apresentou no Las Vegas Trade Show a versão 1.0 do **AutoCAD**. Devido a grande aceitação no mercado, a **AutoDESK** concentrou–se em redefinir, testar e melhorar o produto. Após este passo, o primeiro usuário do **AutoCAD** pode utilizá–lo no início de 1983.

IV.2. GENERALIDADES

Aplicações utilizando CAD são ferramentas de trabalho extremamente poderosas. A velocidade e facilidade com as quais um desenho pode ser preparado e modificado utilizando–se o computador, oferecem uma vantagem fenomenal sobre o desenho manual.

O AutoCAD é o software de CAD mais comercializado no mundo todo. Até junho de 1990, cerca de 265.000 cópias oficiais foram comercializadas, dominando cerca de 60% do mercado norte–americano de CAD (o segundo colocado tem menos de 15% , e o terceiro menos de 6%). Estima–se mais de 1.500.000 cópias ilegais.

Virtualmente, não há limite sobre o tipo de desenho que podemos preparar utilizando o **AutoCAD**. Se um desenho pode ser feito manualmente, então ele pode ser feito pelo **AutoCAD**.

A seguir, temos algumas das aplicações nas quais o **AutoCAD** está sendo utilizado atualmente:

. Desenhos de arquitetura de todos os tipos
. Desenhos de interiores
. Fluxogramas e diagramas de blocos organizacionais
. Gráficos de todos os tipos
. Desenhos para aplicações eletrônicas, químicas, mecânicas,

automotivas e aeroespaciais

. Mapas topográficos e cartas náuticas

. Desenhos de navios e embarcações de recreio

. Plotagem e outras representações de funções matemáticas

. Desenhos de iluminação de cenários

. Pautas musicais

. Ilustração técnica e diagramas de montagem

. Logotipos de companhias

. Cartões de visita

. Etc.

As aplicações não terminam com imagens estáticas, entretanto, quando em conjunto com o programa **AutoSHADE** de visualização de sólidos e com o **AutoFLIX** para animação, o **AutoCAD** se torna a base de um sistema de apresentações dinâmicas e interativas.

À medida que trabalhamos com o **AutoCAD**, estamos criando mais do que simples desenhos. Podemos colocar objetos associados em uma única camada, de modo a agrupá–los, formando objetos complexos que podem ser manipulados como um todo. O **AutoCAD** mantém uma base de dados com tamanho, localização, cor e etc. de tudo que é desenhado, para posterior análise e manipulação.

O **AutoCAD** versão 10, é a última versão deste sistema e está disponível em uma grande variedade de microcomputadores e estações de trabalho de engenharia, podendo ser executado sob os sistemas operacionais PC–DOS, MS–DOS, OS/2, UNIX, AEGIS, VMS, e Apple Macintosh II.

O **AutoCAD** versão 10 introduz o conceito de sistemas de coordenadas definidas pelo usuário (UCS – *user coordinates system*). Desta forma, o desenho poderá ter o sistema de coordenadas global (WCS – *world coordinate system*) e vários sistemas locais, cada um com uma origem e uma orientação em relação ao WCS, simultaneamente.

Os sistemas de coordenadas UCS permitem deslocar o plano de construção, simplificando a localização de pontos tridimensionais. Por exemplo, é mais fácil desenhar o telhado de uma casa ou a lateral de um equipamento se definirmos um UCS posicionando e orientando em relação à superfície sobre a qual queremos trabalhar.

Diversos sistemas de unidades podem ser utilizados simultaneamente: métrico, científico, polegadas decimais, polegadas fracionárias e etc. para as unidades lineares, e graus–minutos–segundos, grados ou radianos para as unidades angulares, permitindo assim ao usuário trabalhar no sistema que lhe convir.

No topo do vídeo é apresentada, continuamente, a coordenada atual do cursor. O número de casas decimais apresentadas pode ser definida pelo usuário entre 0 e 14 (embora o **AutoCAD** trabalhe com 64 dígitos internamente).

A entrada de coordenadas pode ser feita através do teclado, pela digitação das coordenadas absolutas ou relativas (cartesianas ou polares), ou pelo cursor. Neste caso, normalmente são utilizados os filtros (que permitem definir um ponto que tenha a mesma abcissa X que uma entidade e mesma ordenada Y de outra entidade), ou através do "trancamento" de objetos quando, por exemplo, pode–se precisar o ponto médio ou extremidade de uma reta, o centro de um arco, etc., garantindo assim a exatidão matemática do nosso desenho.

IV.3. CRIAÇÃO DE ENTIDADES

A maior parte do trabalho com o **AutoCAD** consiste em posicionar entidades em coordenadas específicas. Uma entidade é um elemento pré–definido, que é colocado no desenho através de um simples comando. Linhas, arcos e círculos são exemplos de entidades utilizadas frequentemente.

A listagem a seguir descreve as entidades oferecidas pelo **AutoCAD**.

. ***Line*** :

Linhas podem ser desenhadas com vários tipos de padrões (pontilhado, traço, ponto, etc.). Para desenhar um segmento de linha, pode–se utilizar coordenadas bi ou tridimensionais.

. ***Arc/Circle*** :

Arcos e círculos também podem ser desenhados com vários padrões de linhas. Vários métodos estão disponíveis para o desenho dos mesmos.

. ***Point*** :

Pontos podem aparecer como quadrados, círculos, cruzes, etc., ou qualquer combinação destes. Podem ser posicionados através de coordenadas bi ou tridimensionais.

. ***Text*** :

Textos podem aparecer numa grande variedade de fontes, qualquer tamanho e orientação que se deseje. Em adição, podem ser criados estilos de texto, aplicando compressão ou inclinação aos fontes de caracteres.

. ***Trace*** :

Trace são linhas bidimensionais sólidas, com qualquer espessura que se deseje.

. ***Solid*** :

Estas entidades são formadas por objetos triangulares ou quadrilateros sólidos (completamente preenchidos).

. ***Shape*** :

São pequenos objetos que podemos definir fora do **AutoCAD**, e os quais são inseridos em nossos desenhos em qualquer escala e orientação.

. ***Block*** :

Blocos são entidades compostas, formadas de grupos de outras entidades.

. ***Attribute*** :

Atributos anexam uma informação constante ou variável a cada ocorrência de um bloco. Podemos indicar se eles serão visíveis ou não. São extremamente úteis para a geração de listas de materiais.

. ***Dimension*** :

Dimensões são entidades compostas similares a blocos, contendo todas as linhas, setas e textos que compreendem uma anotação de dimensão.

. ***Polyline*** :

Polilinhas são linhas e segmentos de arcos bidimensionais conectados, opcionalmente com algum padrão não contínuo de linha e espessura. São fornecidos comandos internos no **AutoCAD** para que, através das polilinhas, sejam construídas elipses, polígonos regulares, anéis, etc.

. ***3DPolyline*** :

São polilinhas tridimensionais compostas de segmentos de reta espaciais. Estas polilinhas podem ser ajustadas (suavizadas) posteriormente, através de *splines* quadráticas, cúbicas ou de *Bezier*.

. ***3DFace*** :

São entidades tridimensionais planas triangulares ou quadrangulares. São fornecidos comandos internos ao **AutoCAD** para construção de toróides, domos, esferas, cones, etc. utilizando estas entidades.

. ***3DMesh*** :

São malhas tridimensionais poligonais. Podemos especificar o tamanho da malha e a localização para a construção de superfícies desenvolvíveis, superfícies de revolução, ou superfícies tabuladas através destas linhas.

Cada entidade plana pode ser desenhada a uma elevação especificada (cota Z), acima ou abaixo do plano XY do sistema de coordenadas UCS corrente. Em adição, muitas das entidades podem ter uma espessura associada, isto é, elas podem ser extrudadas de um determinado valor.

IV.4. VISUALIZAÇÃO DO DESENHO

O AutoCAD oferece–nos várias maneiras de vermos o desenho, enquanto este estiver em progresso ou após terminado.

. ***Zooming*** e ***Paning*** :

Podemos ampliar ou reduzir a imagem do desenho no vídeo (***zoom***). Podemos ainda deslocar nosso vídeo sobre o desenho (***pan***) em qualquer direção, sem alterar o seu tamanho relativo na tela. Ambos os comandos (***zoom*** e ***pan***) são dinâmicos, isto é, podem ser executados enquanto estivermos no meio de outro comando do **AutoCAD**.

. **Vista em Planta e em 3D** :

O termo "Vista em Planta" (*Plan View*) é derivado da vista quando olhamos uma planta baixa de arquitetura. No **AutoCAD**, ela é uma vista no UCS corrente, a partir de uma posição diretamente acima do plano de construção, o qual é paralelo ao monitor.

Podemos também ver nosso desenho de qualquer ponto no espaço (até mesmo de dentro de um objeto). Através da mudança do ponto de vista, podemos examinar todas as partes de um objeto, como se estivéssemos caminhando ao redor, ou como se uma seção do mesmo tivesse sido removida.

De qualquer ângulo de observação, pode-se escolher entre projeções paralelas e perspectivas cônicas, bem como seccionar o objeto entre dois planos, esconder as linhas invisíveis, etc..

IV.5. EDIÇÃO DO DESENHO

As ferramentas de edição do **AutoCAD** facilitam a correção ou revisão de um desenho. Por exemplo, podemos mover um objeto para uma nova coordenada, deslocá–lo para uma camada diferente, ou apagá–lo. Se o quisermos de volta, um simples comando irá restaurá–lo imediatamente. Cópias múltiplas de objetos, organizadas de forma retangular ou circular, podem ser feitas com um simples comando.

O **AutoCAD** também nos permite executar as seguintes funções de edição:

Rotação, ampliação ou redução, compressão, extensão ou corte até encontrar outro objeto precisamente, divisão de um objeto em partes iguais, espelhamento, criação de filetes e chanfros, etc.

Algumas funções de edição são específicas de algumas entidades:

Podemos deletar partes de um círculo, ajustar curvas a polilinhas, explodir polilinhas (transformando–as em entidades simples).

Podemos hachurar objetos fechados, definindo uma escala e ângulo de rotação para a hachura.

Dimensionamentos podem ser feitos das mais diversas maneiras (vertical, horizontal, alinhado, angular, raio, diâmetro, etc.). Basta apontar o objeto a ser dimensionado e a posição na qual queremos colocar o texto do dimensionamento. O **AutoCAD** calcula automaticamente a cota, posicionando–a no desenho. Se o objeto for deformado mais tarde, esta cota será automaticamente alterada.

IV.6. OPERAÇÃO DO PROGRAMA

A operação do **AutoCAD** é totalmente orientada por *menus*. Uma vez estando no editor gráfico, vários menus nos auxiliam na confecção/alteração do desenho:

. ***Screen Menu*** :

Este *menu* é apresentado do lado direito do monitor. Ele permite que forneçamos um comando, simplesmente apontando–o com o auxílio de um dispositivo apontador (*mouse*, mesa digitalizadora, *ligth pen* ou teclas de direção do teclado).

. ***Tablet Menu*** :

Até quatro *menus* de comandos podem ser colocados em uma mesa digitalizadora. Bastará apontar o comando na mesa para que a função seja executada.

. ***Button Menu*** :

Se o cursor da mesa digitalizadora ou o *mouse* possue vários botões, podemos associar estes a alguns comandos do **AutoCAD** mais frequentemente utilizados.

. ***Pull–Down Menu*** :

Se o monitor for capaz de trabalhar com a AUI (*Advanced User Interface*), podemos selecionar comandos de uma barra de *menus* no topo da tela, como no WINDOWS ou OS/2.

. ***Icon Menu*** :

Qualquer item de *menu* selecionado pode apresentar um *menu* de *icons*, que apresenta as opções possíveis em uma forma pictórica. Isto é feito, por exemplo, na seleção do tipo de hachura, ou do estilo do texto. Este tipo de *menu* exige que o monitor seja capaz de operar com AUI.

. ***Dialog Box*** :

São quadros de diálogos que facilitam a interface com o usuário. São utilizados para a operação de camadas, solicitação de atributos, etc., exigindo o AUI.

A interface AUI é disponível para todos os padrões de vídeo estabelecidos pela IBM, como CGA, EGA, VGA, PGC, bem como para algumas placas não padrões (Hercules por exemplo).

IV.7. ARQUITETURA

O **AutoCAD** foi projetado como um sistema de desenho de uso geral. Muitas companhias, entretanto, tem os seus próprios padrões, assim como cada desenhista possue o seu próprio estilo. Para satisfazer as necessidades de cada operador do **AutoCAD**, o sistema foi elaborado utilizando o conceito de arquitetura aberta. Desta forma, é possível:

- Definir *menus* de tela, *pull–down*, *icon*, *tablet* e *buttons* personalizados, automatizando as operações necessárias.
- Criar arquivos *script* para automatizar sequências longas de comandos.
- Definir novos fontes de texto.
- Definir novos tipos de linhas.
- Definir novos estilos de hachuras.
- Criar desenhos de prótotipos com parâmetros customizados.
- Alterar o arquivo *help*, retratando novos comandos.
- Utilizar arquivos DXF ou IGES para transferir a geometria do desenho para outros programas para análise, ou para criar desenhos a partir de dados gerados por outros programas.
- Gerar arquivos *slides* dos seus desenhos, para incorporação em documentos produzidos com *software* de *desktop publishing.*
- Executar programas externos durante a edição de um desenho.
- Utilizar a linguagem **AutoLISP** (embutida no **AutoCAD**) para executar cálculos, automatizar tarefas repetitivas, criar novos comandos para o **AutoCAD** ou redefinir comandos já existentes.

IV.8. EQUIPAMENTO

Recomenda–se um computador do tipo AT para uso industrial. Esperar pelo computador para processar o desenho, significa uma perda de concentração e produtividade. O mais próximo que se estiver da resposta instantânea, maior a produtividade.

AutoCAD é tão poderoso que necessita de toda a memória que se possa dispor. Versão 2.5 em diante requer 640K RAM (*Randomic Access Memory*) e suporta os novos padrões para *Extended and Expanded Memory.* Uma unidade de disco rígido é necessária e para versão 9.0 exige–se o uso de co–processador matemático.

Em alguns computadores, o **AutoCAD** usa dois monitores, um para comandos e textos e outro para gráficos. Nestes sistemas, o monitor gráfico pode também apresentar um menu de tela ao longo de seu lado direito e uma linha de comandos através da base.

Em outros sistemas, um simples monitor pode ser usado para ambos, gráficos e textos. Aqui, há três linhas na base da tela reservadas para comandos, e o lado direito pode conter um menu de tela. Quando **AutoCAD** está executando em um sistema com um único monitor, ele guarda um total de 24 linhas (ou mais) de texto, apenas como a regular tela de textos. Se a informação "correr" para fora das três linhas da tela, pode–se usar uma tecla para trocar para a tela de textos para ver essa informação. O **AutoCAD** automaticamente troca para a tela de textos quando emite uma grande quantidade de informações.Retorna automaticamente a tela gráfica quando desenha algo.

IV.8.1. EQUIPAMENTO OPCIONAL

O equipamento listado acima é suficiente para uma instalação básica. Com esta instalação, pode–se aprender como usar e fazer desenhos.

Equipamentos adicionais são necessários se se deseja fazer cópias em papel. Para realizar total uso dos benefícios da produtividade do sistema CAD, em alguns casos, uma porta serial RS–232C adicional será necessária para conectar este equipamento ao computador.

Uma *plotter* ou uma impressora gráfica podem ser conectadas ao sistema por uma porta serial RS–232C. Impressoras normalmente produzem desenhos mais grosseiros que as *plotters*, mas são geralmente muito mais rápidas. Assim, pode–se usar impressoras para frequentes verificações e então enviar o desenho final para a *plotter*.

Sugere–se que sejam adicionados equipamentos de digitalização tais como *mouses*, mesas digitalizadoras ou canetas óticas. A entrada através do teclado é relativamente fácil, mas apontar na tela e apertar um botão é sempre mais fácil. Em adição à locação de pontos e entrada de comandos, pode–se usar a mesa digitalizadora para copiar desenhos existentes.

Mesas digitalizadoras têm prós e contras. Algumas opiniões de usuários são baseadas em economia, já que mesas digitalizadoras custam três a sete vezes mais que um *mouse*.

A maioria dos grandes sistemas CAD tem digitalizadoras e *menus* definíveis pelo usuário. Há uma razão para isto: com *menus* de tela, a organização dos comandos é fragmentada em unidades discretas de 20 opções ou menos, quando consideramos que um, dois ou três espaços são necessários na base de cada sub–*menu*, apenas para retornar ao *menu* principal ou ao *menu* anterior.

Por ser cada sub–*menu* de tela uma unidade discreta, temos que colocar opções comuns em cada um, como *object snap* (pode colocar este em um sub–*menu* próprio, mas isto necessitará duas adicionais seleções para todo uso).

Menus de mesa são mais simples de criar e organizar que os *menus* de tela.

Por poder cancelar cada prévio comando de seleção de outro, o *menu* de mesa, auxilia a trocar de idéia instantaneamente e sem penalidades. *Menus* de tela podem necessitar três seleções adicionais se trocar de idéia.

Pelo uso de um menu de mesa e uma boa disposição, usuários podem realizar um crescimento de 50 por cento na curva de aprendizado, bem como 25 por cento de aumento na produtividade sobre outros equipamentos de entrada.

IV.9. BANCOS DE DADOS

Além de uma ferramenta de desenho que permite a construção de figuras geométricas, tais como linhas, arcos e círculos, permite a criação e revisão de atributos , extração de arquivos de dados e o uso de blocos e símbolos no gerenciamento de bancos de ados. Permite também, a conversão e transferência de dados entre sistemas CAD.

O AutoCAD também permite a extração de informações visíveis e invisíveis de um desenho, colocando estas informações em formato legível, traduzindo então para um diferente ambiente (tal como um processador de textos ou um gerenciador de banco de dados) para a geração de listas de materias, relatórios, tabelas ou outro tipo qualquer de informação.

De fato, os usuários do **AutoCAD** podem manipular dados de três maneiras:

A primeira é com informações gráficas manipuladas com o editor do **AutoCAD** por si mesmo. Esta é a função de desenho.

A segunda é com **AutoLISP**, uma linguagem de programação embutida no **AutoCAD** que auxilia o usuário a escrever seus próprios programas, os quais gerenciam e manipulam os dados gráficos e não gráficos de um modo impossível de ser feito com o **AutoCAD** unicamente.

A terceira maneira é extrair as informações desejadas do arquivo gráfico e organizá–las fora do **AutoCAD** com programas gerenciadores de dados, externos, tais como DBase ou LOTUS 1–2–3 ou um outro qualquer.

Um banco de dados é uma coleção de informações a respeito de um determinado assunto, organizado de modo a torna–las, legíveis e acessíveis ao usuário. Um gerenciador de banco de dados, em outras palavras, é um programa de computador que nos auxilia a criar, organizar e acessar todas as informações contidas no banco de dados. DBase, por exemplo, pode ser usado para criar, organizar e acessar um banco de dados tal como uma lista de endereços.

O **AutoCAD** é também um gerenciador de banco de dados. Ele pode ser usado para criar, organizar e acessar dados gráficos, que são desenhos.

O que é mais importante e parece pouco óbvio, é o modo que o **AutoCAD** pode gerenciar informações não gráficas sobre um desenho. Não apenas pode o **AutoCAD** traçar as figuras de um projeto, como também traçar os dados alfanuméricos associados a essas figuras. Qualquer coisa desenhada com **AutoCAD**, mesmo sendo peça mecânica, uma fornalha ou um prédio, podem ser tratados ambos, como especificações dentro do banco de dados do **AutoCAD**.

Uma lista de endereços, por exemplo, pode ser usada para enviar anúncios para todas as pessoas da lista. Para fazer isto, o banco de dados precisa ser estruturado para que contenha uma suficiente descrição sobre cada indivíduo, tal que o anúncio possa chegar até ele. Um banco de dados gráfico é estruturado exatamente da mesma maneira que outros bancos de dados. Seu propósito é descrever um desenho de um modo não gráfico.

Um arquivo de banco de dados gráfico consiste de campos de descrições alfanuméricas de desenhos primitivos (o **AutoCAD** chama estes desenhos primitivos de entidades).

Um desenho primitivo é uma forma básica usada em um desenho, tal como uma linha, arco, círculo, ponto ou texto. Quando desenhamos uma linha no **AutoCAD**, estamos acionando uma primitiva chamada ***line*** e definindo seus pontos inicial e final. O **AutoCAD**, então, coloca uma descrição desta primitiva no banco de dados, e uma representação gráfica deste campo do banco de dados é apresentada na tela do monitor.

Quando se trabalha com o **AutoCAD**, automaticamente adiciona–se, altera–se ou retira–se informações do banco de dados. Este é o primeiro tipo de gerenciamento de dados do CAD.

IV.10. CUSTOMIZAÇÃO

O **AutoCAD** é um poderoso e de fácil uso, programa de projeto e desenho assisitido por computador. Com um entendimento elementar, usuários podem criar desenhos precisos muito mais rápido que manualmente. Muitos de nós usam **AutoCAD** neste nível, apenas como este veio de fábrica: com um *mouse* ou mesa digitalizadora e o *menu standard* do AutoCAD. O real poder do **AutoCAD** caminha junto com a sua habilidade de ser facilmente configurado para nossas necessidades.

O trabalho apresentado aqui, abrange um campo de recursos do **AutoCAD** que vai desde o sistema operacional (DOS), melhor utilização de discos rígidos até macros em *menus*, rotinas externas e rotinas em **AutoLISP**.

Iniciantes, o maior grupo de usuários, frequentemente utilizam menos da metade dos benefícios e produtividade oferecidos por seus sistemas **AutoCAD**. Eles tendem a alcançar um platô após terem podido fazer uma linha no papel um pouco mais rápido com **AutoCAD** que manualmente. Mesmo não tendo aprendido muitas técnicas simples, podem salvar horas de trabalho em cada semana.

Usuários intermediários geralmente conhecem alguns comandos e truques do **AutoCAD**. Estão familiarizados com seu sistema operacional e com um processador de textos já fazem algumas configurações pessoais.

Usuários avançados gastam longas horas de trabalho em *menus* e macro para o **AutoCAD**. Eles podem ler todas as referências, instigados pelo "*Mainframe*" CAD no micro.

De fato, um típico usuário de CAD, usa apenas 20 por cento dos comandos e opções do **AutoCAD** para produzir 80 por cento do trabalho – a familiar regra dos 20/80. Parte disso, deve–se ao fato de que o **AutoCAD** foi projetado para satisfazer as necessidades de milhares de usuários e foram incluídos mais comandos que qualquer indivíduo pode regularmente usar. Contudo, o completo número de comandos disponíveis e opções pode frequentemente fazer o sistema incômodo e lento.

Isto demonstra a necessidade de repensar os menus e modificá–los para fazer estes importantes 20 por cento trabalharem melhor para o usuário.

O **AutoCAD** apresenta uma arquitetura aberta, o que permite ao usuário facilmente reescrever *menus* e criar poderosas ferramentas para suprir uma determinada disciplina. Não é preciso esperar por alguém de uma distante cidade desenvolver uma nova função ou criar uma aplicação especializada; não é preciso aceitar a interface homem/máquina idealizada por um programador não familiarizado com suas específicas necessidades.

O **AutoCAD** criou um mecanismo gráfico universal, dotado de rapidez, facilidade de uso e ferramentas de configuração pessoal. Em cada revisão do programa, a Autodesk aumenta a capacidade do **AutoCAD** de adaptar–se às necessidades individuais. Com a versão 2.5 foi criada um dos mais podersos e excitantes recursos internos de programação de qualquer sistema CAD, o **AutoLISP**. O potencial do **AutoCAD** agora é limitado apenas a imaginação do usuário.

Falou–se de produtividade, a qual simplesmente indicamos como a habilidade de alcançar e manter maior rapidez, performance e flexibilidade com o sistema. Qualquer desenvolvimento do **AutoCAD**, que regularmente salve tempo, é bem recompensado no esforço gasto na sua criação.
Alguns dos beneficios obtidos:

1. **Baixo custo de suporte** :

O **AutoCAD** é um sistema CAD universal, mas pode ser facilmente personalizado para satisfazer as necessidades dos mais variados departamentos e aplicações.

2. **Tempo salvo** :

Pelo uso de algumas técnicas, o pressionamento de muitas teclas pode ser combinado em um comando de menu. Programas **AutoLISP** podem pedir os dados e então executar internamente, criando o desenho por si mesmos. Dados de engenharia podem ser passados diretamente ao **AutoCAD** e o desenho criado sem a intervenção do operador.

3. **Redução** de **erros** :

Expressões matemáticas podem ser construídas em programas **AutoLISP** de tal modo que, tudo que se tem a fazer é responder a poucas questões e nunca se preocupar com cálculos incorretos.

4. **Consistência** :

Macros e programas **AutoLISP** combinam grupos de repetitivos pressionar de teclas em umas poucas, aumentando a consistência e ganhando tempo.

5. **Necessidade de pouca prática** :

Isto deve–se ao fato de funções complexas poderem ser bastante simplificadas. Usuários pouco experientes podem realizar o mesmo trabalho que antes só usuários avançados podiam realizar.

6. **Melhores projetos** :

Situações alternativas podem ser rapidamente exploradas quando rotinas **AutoLISP** são usadas para criar desenhos a partir de parâmetros especificados pelo usuário.

7. **Motivação criativa** :

O CAD torna–se mentalmente fatigante quando seus operadores realizam o mesmo tipo de desenho e repetem o mesmo conjunto de pressionar de teclas dia após dia. O conhecimento de macros e rotinas **AutoLISP** encoraja a procurar caminhos para automatizar e expandir seu conhecimento de **AutoCAD**. Como todos sabemos, criatividade traz satisfação e aumenta a segurança no trabalho.

Como em todo o sistema de pré–processamento gráfico de elementos finitos, alguns cuidados devem ser tomados para a manutenção da consistência entre os dados gráficos e estruturais.

Devido ao esquema de armazenamento adotado, os elementos de função estrutural, ou seja, elementos finitos, nós, carregamentos e restrições nodais, deverão ser criados ou alterados pelos comandos do **AutoMESH. A** utilização de comandos do AutoCAD para a realização de tais tarefas é possível mas desaconselhada, pois a má utilização acarretaria inconsistências nos dados estruturais.

A grande inovação feita neste sistema está na forma de armazenamento de dados adotada.

Nos sistemas gráficos de uso geral conhecidos e utilizados no pré–processamento de elementos finitos, há uma distinção a nível de banco de dados entre elementos estruturais e entidades puramente gráficas.

As características e propriedades do elemento são armazenadas normalmente em uma estrutura de dados paralela aos dados, que define sua representação gráfica.

CAPÍTULO V

V.1. O SISTEMA AUTOMESH

Neste capítulo apresenta-se o sistema proposto, através dos diversos comandos que o compõem. Na primeira parte, apresenta-se o ambiente de trabalho com suas considerações básicas de operação, entidades estruturais e suas representações gráficas, bem como a lista dos comandos do sistema. Na segunda parte, os comandos são descritos e exemplos de operação são apresentados.

O conjunto de comandos que compõe o sistema em momento algum substitui os comandos básicos do **AutoCAD**. O trabalho de desenho permanece disponível: detalhamento, cotas com dimensões, estampas, notas (textos); e todas as demais entidades gráficas do **AutoCAD** podem ser incluídas juntamente com os elementos estruturais, sem prejudicar a consistência da malha de elementos finitos. Isto significa que, desenhos detalhados de estruturas podem ser discretizados em malhas de elementos finitos ou o inverso, malhas de elementos finitos podem ser detalhadas sem prejudicar os dados estruturais.

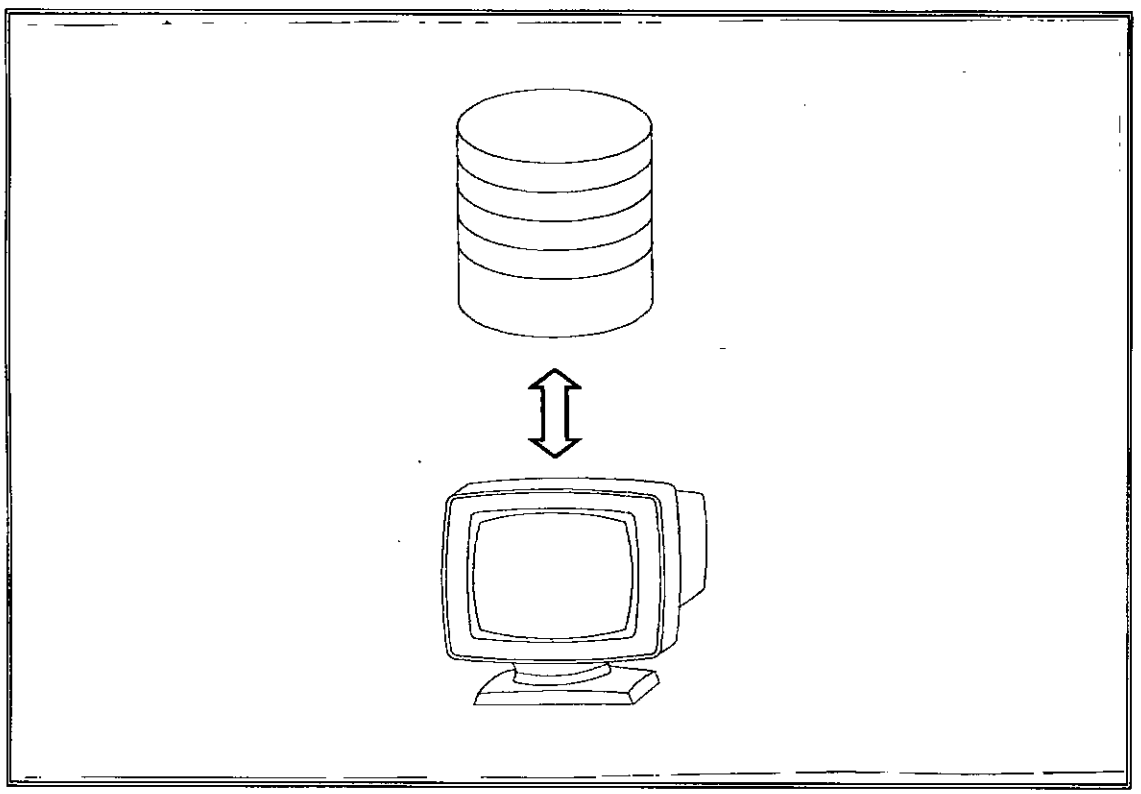


Fig V.2

O ambiente que cerca a entidade gráfica define a distinção entre elementos estruturais e não estruturais no momento da extração do banco de dados. Por exemplo, determinado símbolo (***shape***) em um específico ***layer*** representa um ponto nodal, em outro ***layer*** qualquer seria apenas um ***shape***.

Este conceito ficará mais claro após análise do esquema estrutural adotado.

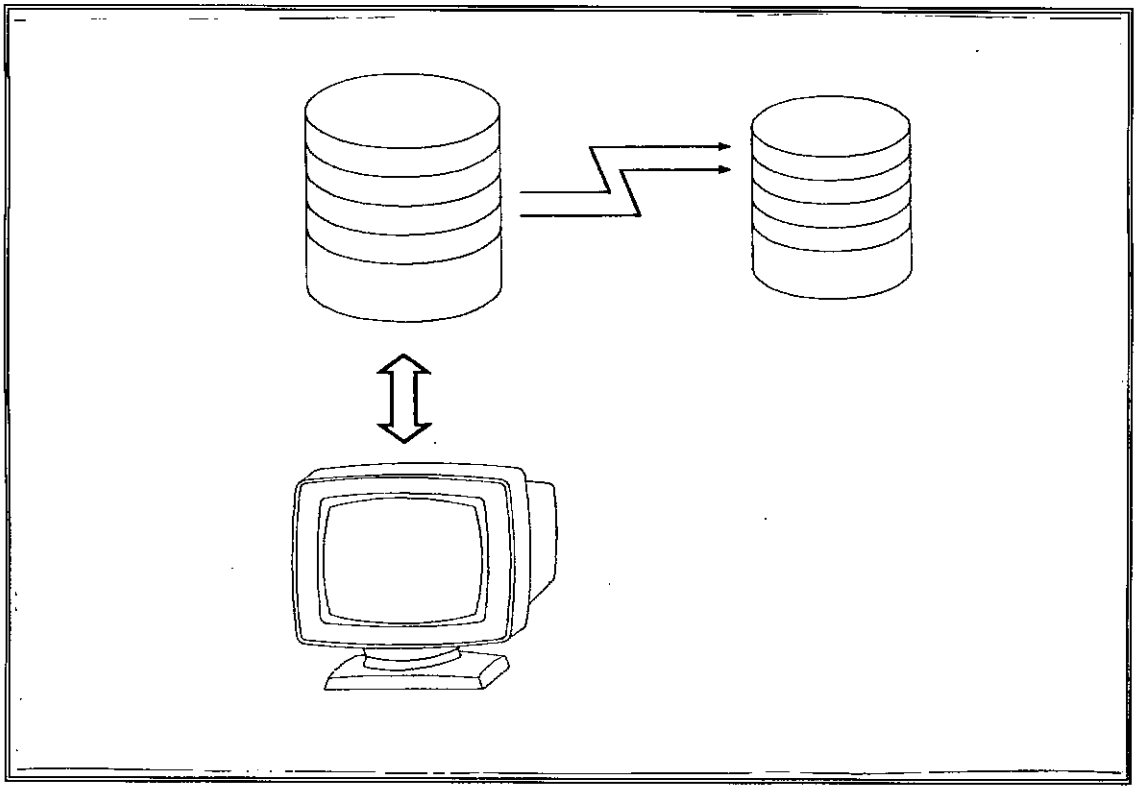


Fig V.1

No sistema proposto, tal distinção não é feita. Trabalha–se o tempo todo com entidades puramente gráficas.

A distinção é feita no momento da extração do banco de dados para a análise. Com isto, ganha–se tempo nos processos de criação e alteração da malha de elementos finitos, além de uma grande redução na quantidade de dados armazenados.

Para simbolizar os pontos nodais com alguma restrição nodal, foram criados 63 ***shapes*** que correspondem às possíveis combinações de restrições para os seis graus de liberdade.

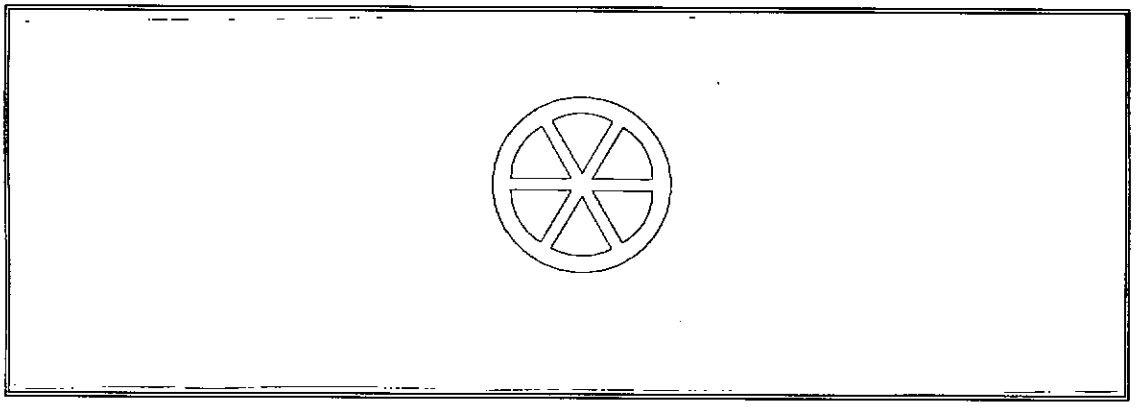


Fig V.4

A figura acima representa um nó com os seis graus de liberdade restritos, sendo que cada gomo corresponde a um grau de liberdade.

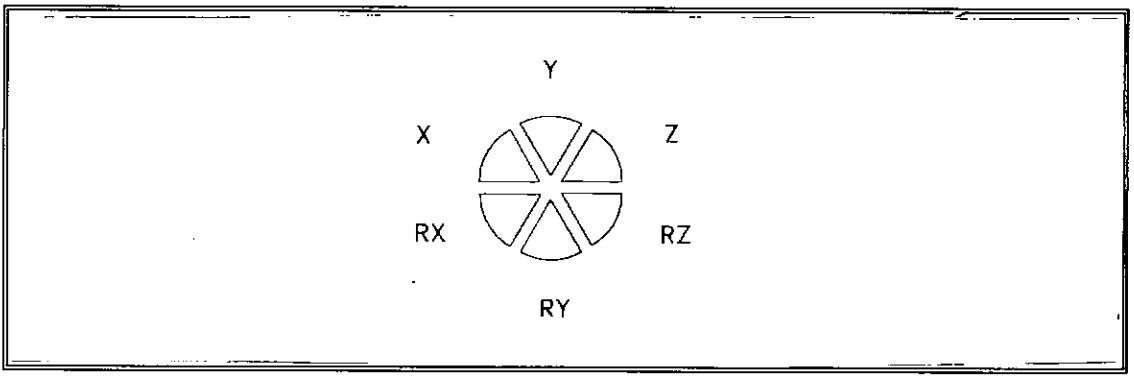


Fig V.5

V.2. DEFINIÇÃO DOS ELEMENTOS ESTRUTURAIS

Para efeito de representação gráfica, existem três entidades estruturais distintas:

. NÓS
. ELEMENTOS
. CARREGAMENTOS

Os nós são representados como ***shapes*** pré–definidos. O símbolo nodal tem função puramente estética, serve para facilitar a localização dos pontos nodais, nós e elementos, e ainda auxiliar na distinção entre elementos com funções de forma linear e parabólica, por exemplo, distinguir entre elementos de placa de quatro e oito nós.

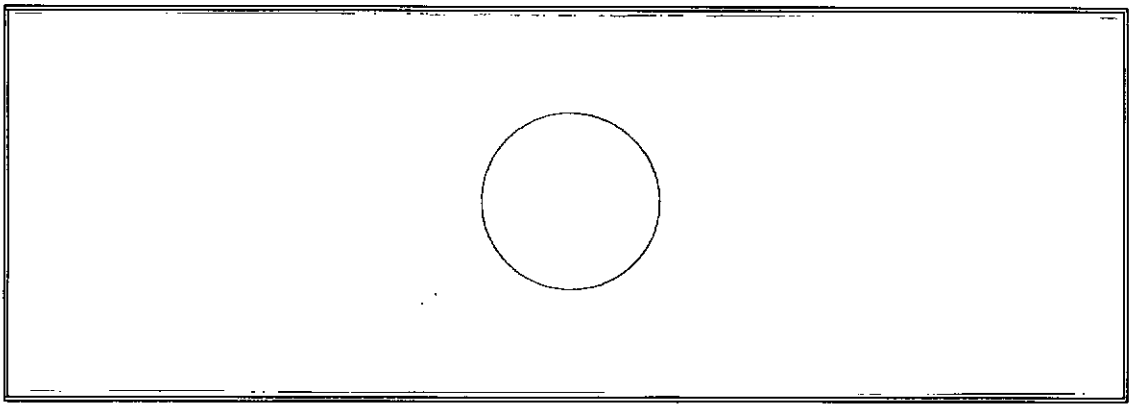


Fig V.3

Este símbolo refere–se a um ponto nodal sem restrições nodais.

Elementos planos para estruturas espaciais são representados por ***3dfaces***, os quais permitem a criação de superfícies e, portanto, a retirada de linhas ocultas pelo comando ***Hide*** do **AutoCAD**. Porém, só poderão ser criados elementos de três ou quatro nós, devido a própria definição da entidade gráfica.

Elementos sólidos não estão implementados para o **AutoCAD** por limitações próprias do produto, no entanto a **AutoDESK**, empresa que produz o **AutoCAD**, oferece o **AutoSOLID**, um produto mais adequado a este tipo de trabalho.

No **AutoSOLID**, a criação dos modelos é feita utilizando-se técnicas de construção geométrica de sólidos (*CSG Construtive Solid Geometry*).

A exportação do modelo do **AutoSOLID** para o **AutoCAD** é feita através do padrão **DXF**, permitindo assim que o desenho seja cotado, observações sejam adicionadas, etc.

V.3. O AMBIENTE AUTOMESH

O ambiente **AutoMESH** segue a mesma filosofia de interface com o usuário que o **AutoCAD**. Qualquer usuário do **AutoCAD** poderá, portanto, utilizar facilmente o sistema de pré-processamento de estruturas de elementos finitos, sem quaisquer problemas e um mínimo tempo de aprendizado.

Na figura V.6, apresentam–se alguns exemplos:

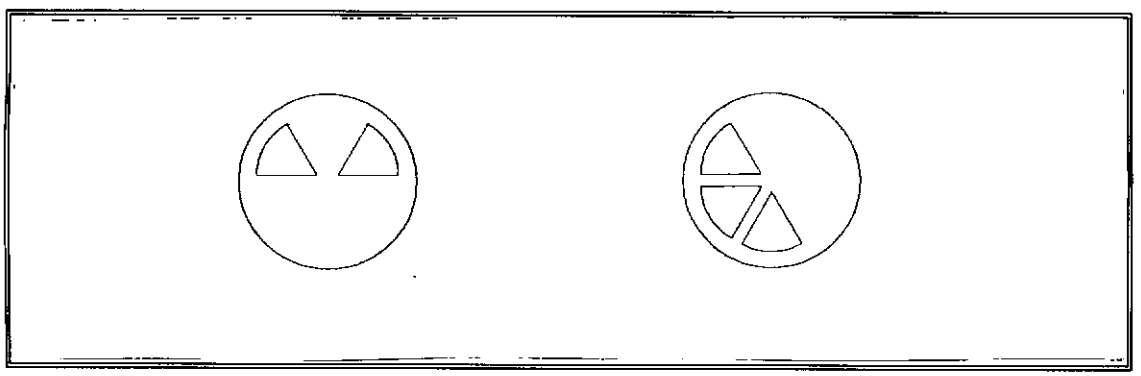


Fig V.6

Os elementos finitos possuem representações distintas para os diversos tipos possíveis e são constituídos basicamente de ***polylines***, ***3dfaces*** e ***3dlines***.

Essas entidades gráficas podem ser distintas das demais como elementos finitos, por estarem definidas em ***layers*** específicos, isto é, todo ***polyline***, ***3dface*** ou ***3dline*** encontrado nestes ***layers*** será tratado como elemento finito, os demais não.

Elementos reticulados são representados como ***3dline*** se possuírem apenas dois nós e como ***polyline*** se isoparamétricos de três nós.

Elementos planos são representados como ***polylines*** e estes podem ter um número qualquer de nós ou mesmo serem elementos infinitos. As ***polylines*** serão, então, abertas ou fechadas respectivamente.

São providos também ***menus*** de comandos adicionais para o monitor (*screen*) e mesa digitalizadora (*tablet*) como também ***help*** para cada comando adicionado. O usuário experiente sentirá um **AutoCAD** expandido e não um programa com apêndices.

Na figura V.7, apresenta–se o menu de mesa (*tablet*) proposto.

Este menu será adicionado aos outros três oferecidos pela **AutoDESK** junto com o **AutoCAD**.

Os quatro trabalharão em conjunto na construção da geometria e discretização da estrutura.

Poderão ter, ainda, os comandos redistribuídos de acordo com a aplicação a que se destinam de modo ergonométrico.

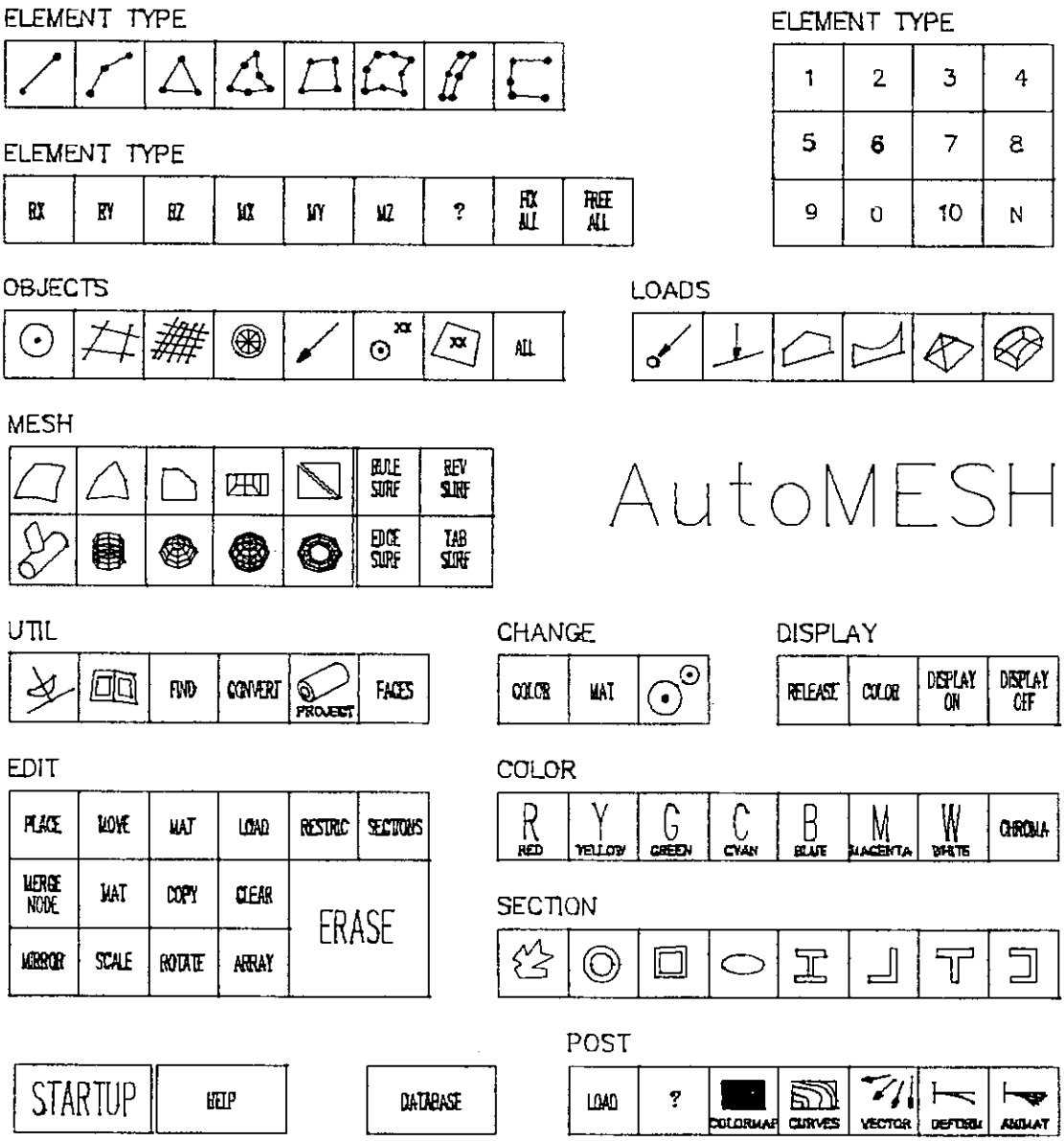

ELEMENT TYPE
ELEMENT TYPE
RX
RY
RZ
MX
MY
MZ
?
FIX ALL
FREE ALL
ELEMENT TYPE
1
2
3
4
5
6
7
8
9
0
10
N
OBJECTS
XX
XX
ALL
LOADS
MESH
RULE SURF
REV SURF
EDGE SURF
TAB SURF
AutoMESH
UTIL
FIND
CONVERT
PROJECT
FACES
CHANGE
COLOR
MAT
DISPLAY
RELEASE
COLOR
DISPLAY ON
DISPLAY OFF
EDIT
PLACE
MOVE
MAT
LOAD
RESTRIC
SECTIONS
MERGE NODE
MAT
COPY
CLEAR
ERASE
MIRROR
SCALE
ROTATE
ARRAY
COLOR
R
RED
Y
YELLOW
G
GREEN
C
CYAN
B
BLUE
M
MAGENTA
W
WHITE
CHROMA
SECTION
STARTUP
HELP
DATABASE
POST
LOAD
?
COLORMAP
CURVES
VECTOR
DEFORM
ABOMAT

FIG VII.7.

A distribuição adotada para o menu do **AutoMESH** reúne os comandos em três grupos:

. SET
. CREATE/EDIT
. DISPLAY

O primeiro grupo é responsável pela definição dos parâmetros que gerenciam a criação ou discretização da estrutura em elementos finitos. Basicamente, é responsável pela definição das propriedades que permanecem ativas e regem a execução de outros comandos.

O segundo grupo inclui os comandos responsáveis especificamente pela criação e alteração da malha de elementos finitos ou seus elementos, nós, materiais, restrições e carregamentos.

O terceiro grupo engloba os comandos auxiliares, comandos de visualização. São em geral, comandos primitivos do **AutoCAD** em formato terminal e dirigido, ou seja, os comandos são executados imediatamente para grupos de objetos pré–definidos, suprimindo as *prompts* (diretivas) intermediárias do processo interativo.

A seguir, apresenta–se o menu de tela (*screen menu*) proposto para o **AutoMESH**.

Este menu foi criado como um submenu do menu fornecido com o **AutoCAD** e mantém a mesma estrutura hierárquica.

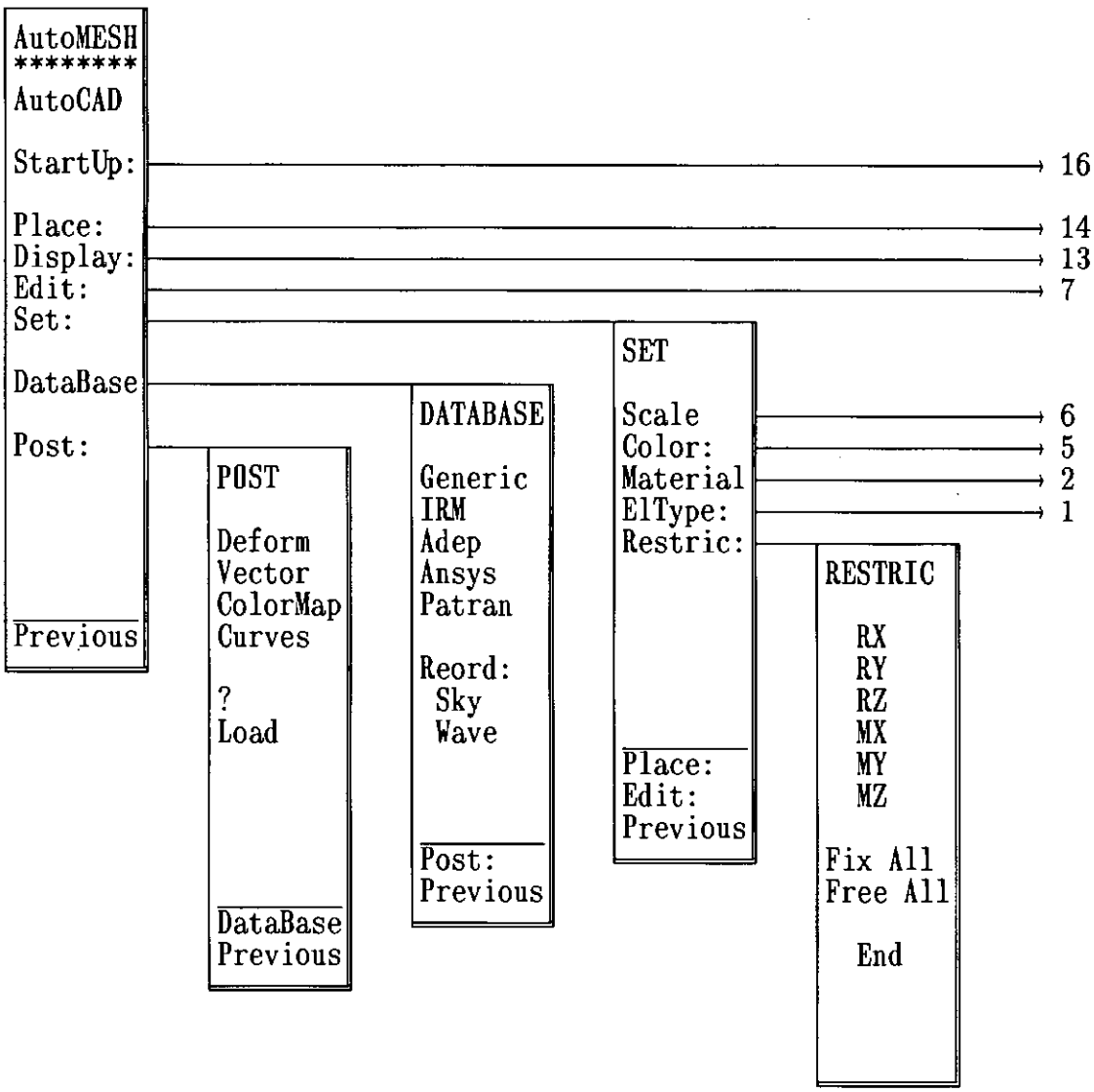

AutoMESH
****
AutoCAD
StartUp:
Place:
Display:
Edit:
Set:
DataBase
Post:
Previous
16
14
13
7
POST
Deform
Vector
ColorMap
Curves
?
Load
DataBase
Previous
DATABASE
Generic
IRM
Adep
Ansys
Patran
Reord:
Sky
Wave
Post:
Previous
SET
Scale
Color:
Material
ElType:
Restric:
Place:
Edit:
Previous
6
5
2
1
RESTRIC
RX
RY
RZ
MX
MY
MZ
Fix All
Free All
End

Fig V.8

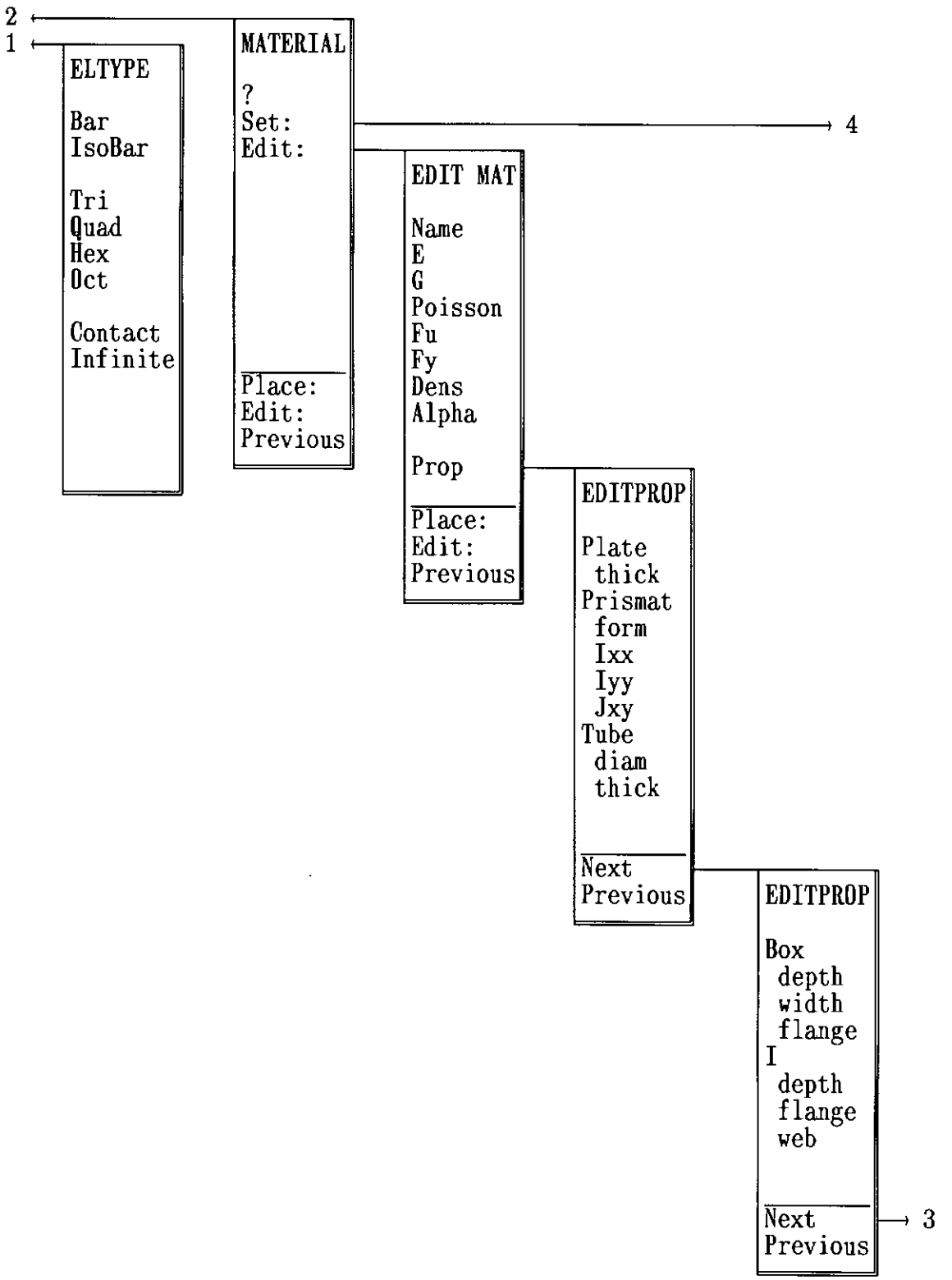


Fig V.8 (continuação)

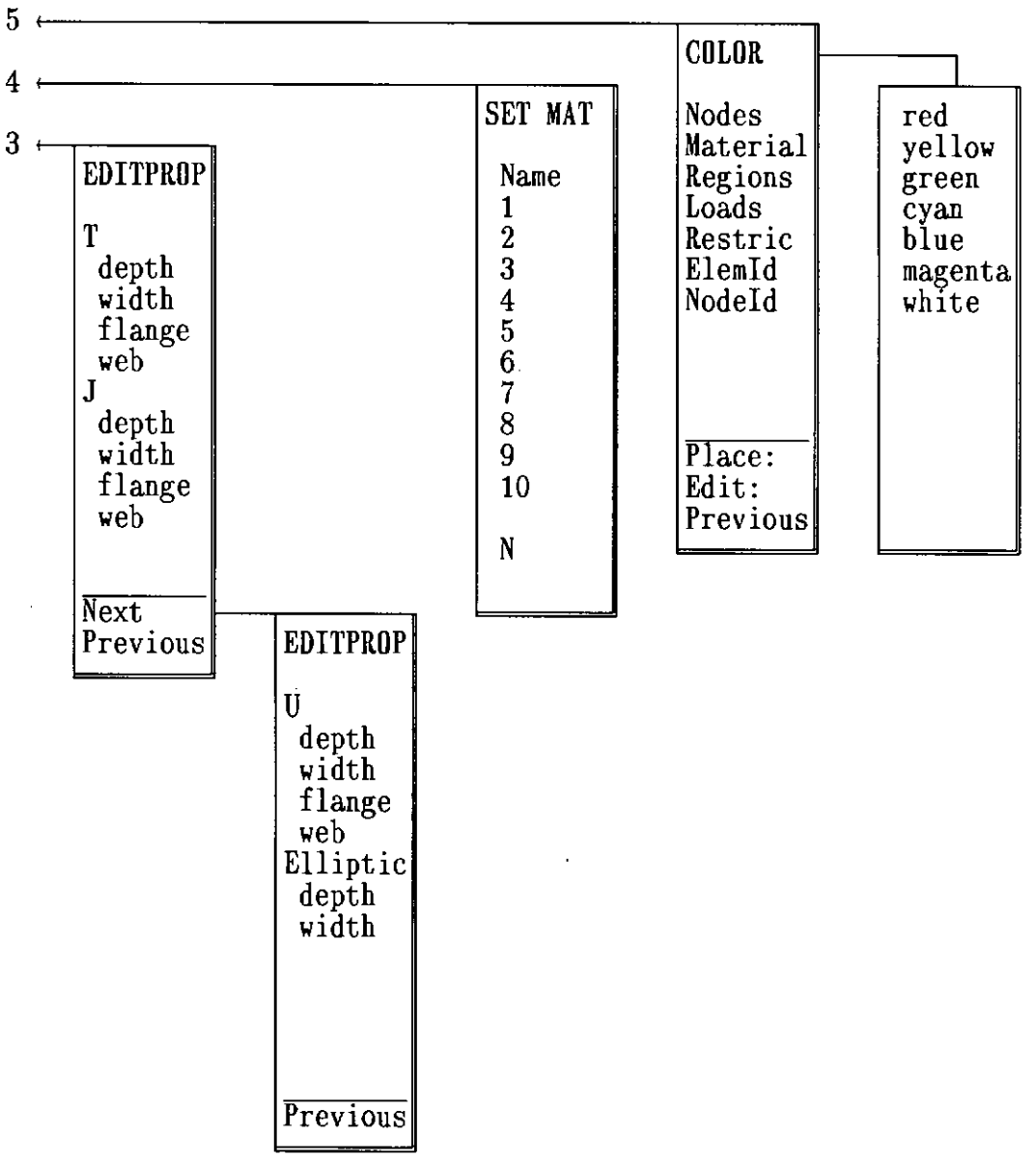


Fig V.8 (continuação)

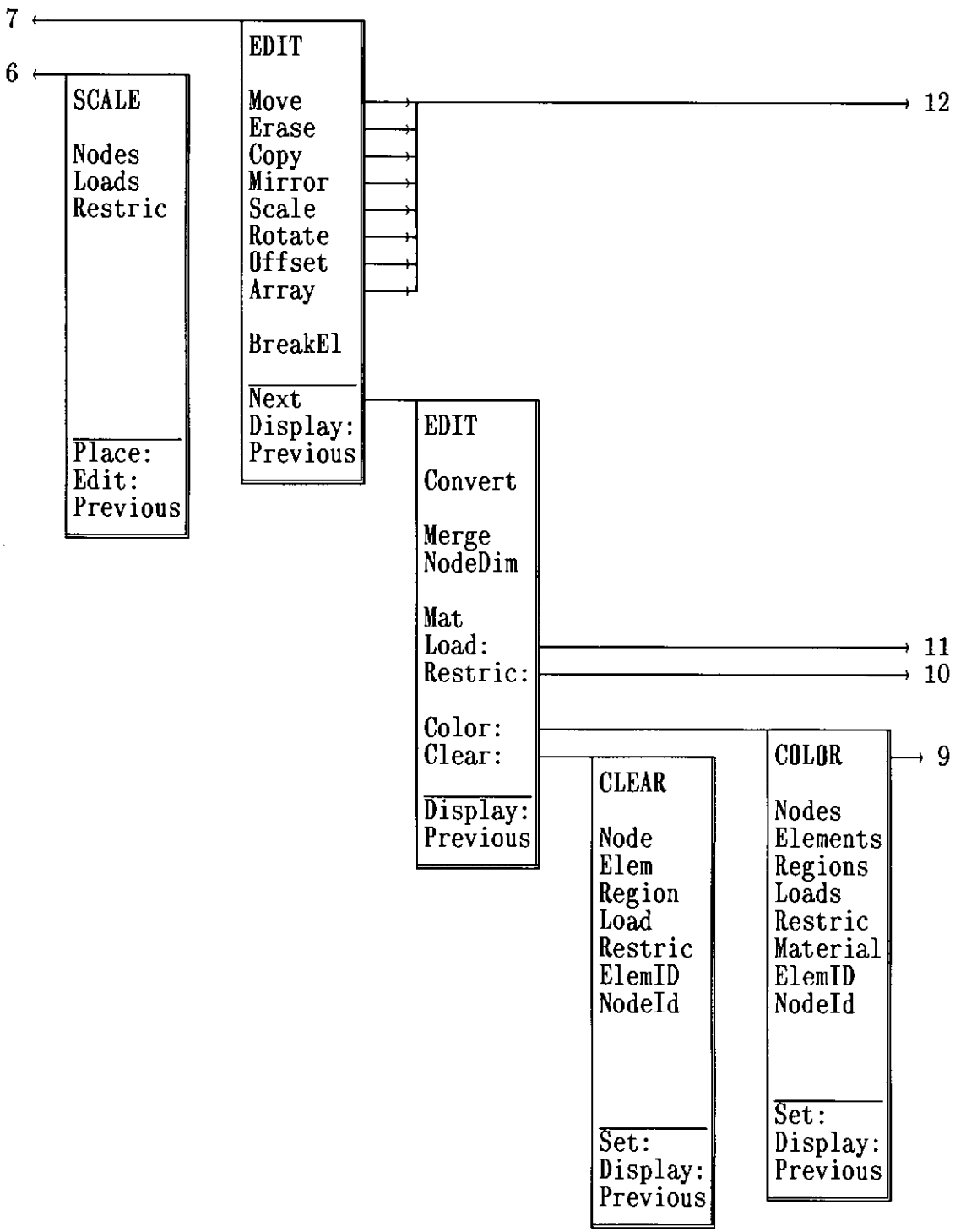


Fig V.8 (continuação)

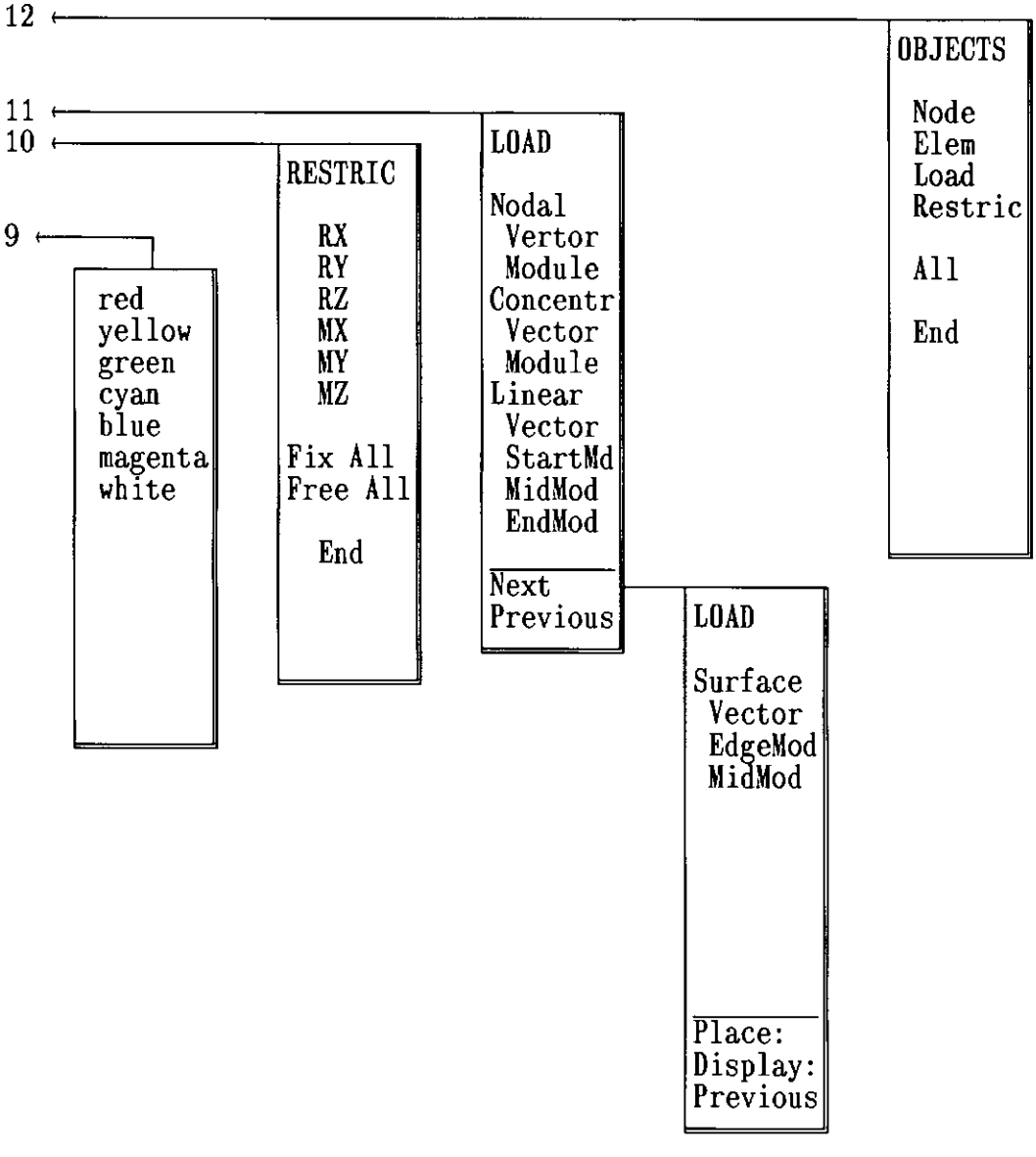


Fig V.8 (continuação)

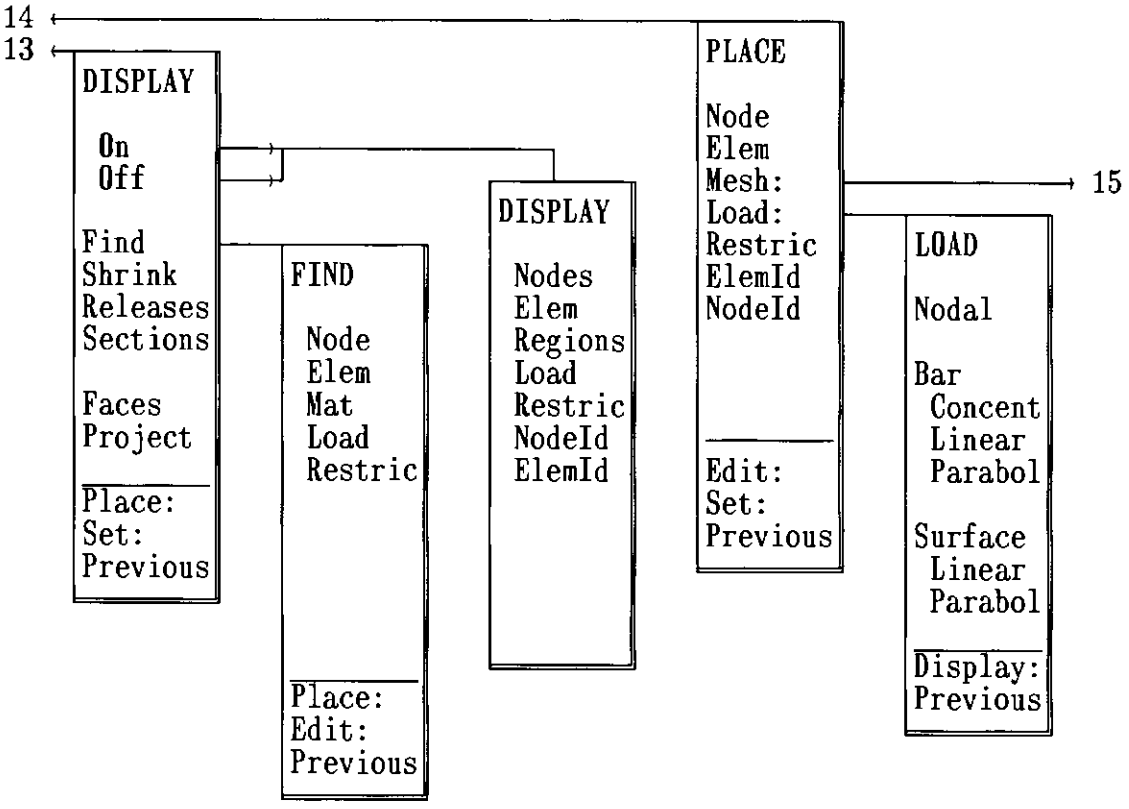


Fig V.8 (continuação)

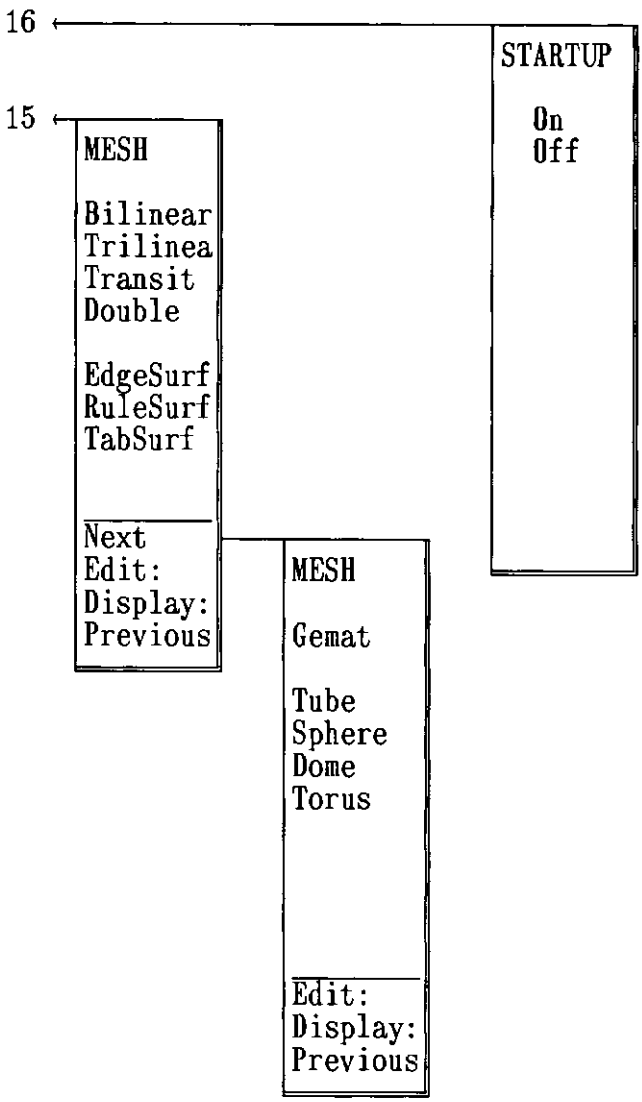


Fig V.8 (continuação)

Na figura deverão ser destacados os seguintes ítens:

. Cada faixa representa um submenu, isto é, o que será apresentado na tela de cada vez.

. Os ítens com o sinal de "dois pontos" representam título de sub–grupos.

. Os ítens com o sinal de "dois pontos" representam chamadas a submenus, ou seja, os comandos propriamente ditos estão localizados em submenus.

. Os ítens sem marcas de "igual" ou "dois pontos" constituem comandos e a sua seleção poderá estar ligada a submenus apenas para auxiliar a definição dos parâmetros de execução. Nestes casos, após feita a seleção dos parâmetros, o retorno a um submenu anterior e execução do comando serão imediatos.

. Os ítens com "ponto" antecedendo, representam os parâmetros que definem o tipo de execução do comando a eles associado.

. Os ítens da linha horizontal que aparecem nas faixas, correspondem a meios de acesso direto a sub–menus, independentemente da cadeia ou hierarquia.

O uso dos sub–menus apresentados não inviabiliza ou desativa os submenus do **AutoCAD** que permanecem ativos, podendo ser acessados a qualquer momento.

V.4. OS COMANDOS DO AUTOMESH

A seguir, apresenta–se o sistema proposto através dos diversos comandos que o compõem.

O conhecimento do funcionamento de cada comando, além da função, facilitará sensivelmente o processo de utilização do sistema **AutoMESH/AutoCAD** na criação e discretização de malhas de elementos finitos.

Os comandos estão reunidos de acordo com cada função, em grupos e sub–grupos. Esta sistemática foi adotada na confecção dos menus, hierárquico (*screen menu*) ou *tablet*, para efeito ergonométrico, ou seja, a busca de um determinado comando é facilitada sabendo–se a que tipo de tarefa se destina.

GRUPOS	COMANDOS
Init	StartUp
Creat	Material Node Element Isomapp Trilinear Transit Double Gemat Tube Sphere Dome Torus Restric Load

Tab V.1

Edit	Merge BreakEl Desloc Delete Copy Scale Rotate Mirror Array
Util	Find Numbering Shrink Face Project Setions Display Intert Convert Clear DataBase Post
Filter	Erase Stretch Trim Break Explode Pedit Extend Move Change Offset

Tab V.1 (continuação)

Na seção seguinte, apresentam–se os grupos de comandos com descrição de cada comando em termos de função e operação.

V.4.1. O COMANDO STARTUP

O comando **STARTUP** é responsável pela inicialização do sistema **AutoMESH**.

O comando possui as seguintes opções:

ON/ OFF: *<default>*

– A opção **ON** permite ao usuário inicializar o sistema **AutoMESH**, caso não tenha sido inicializado alguma vez, ou simplesmente ativá–lo.

O processo de inicialização ou ativação poderá ser realizado automaticamente por opção do usuário, no início da seção gráfica através do arquivo **ACAD.LSP**.

O processo de ativação difere do processo de inicialização, pois no primeiro, o ambiente já está construído, ou seja, as variáveis necessárias já possuem valores iniciais e as funções já se encontram carregadas.

Quando ativo, o sistema **AutoMESH** altera o funcionamento dos comandos básicos de edição do **AutoCAD**. Isto será melhor esclarecido nas seções seguintes.

– A opção **OFF** possibilita ao usuário retornar o **AutoCAD** ao modo de funcionamento padrão. Os comandos básicos de edição do **AutoCAD** não passam por filtros ou processos de checagem prévia antes da execução. Este estado é melhor recomendado para as etapas de detalhamento ou finalização do projeto.
Os comandos do sistema **AutoMESH** são desativados.

O comando **STARTUP** deverá ser o primeiro dos comandos do **AutoMESH** a ser usado. É responsável pela inicialização, preparação do arquivo gráfico para a discretização. Não necessariamente deverá ser o primeiro comando da seção gráfica, mas sim do processo de discretização da estrutura, normalmente já definida graficamente.

Este comando cria as variáveis, define os parâmetros iniciais e carrega funções e arquivos necessários a operação do **AutoMESH**.

Dentre as operações realizadas pelo comando **STARTUP** devemos frisar:

. Criação de ***layers*** básicos e definição das respectivas cores iniciais.

layer	COR INICIAL
NODES	Cyan
SUBREGIONS	Red
LOADS	Blue
RESTRIC	Yellow
NODEID	White
ELEMID	White

Tab V.2

. Inicializar as variáveis necessárias ao **AutoMESH**.

VARIAVEL	TIPO	VALOR INICIAL	DESCRIÇÃO
USERI1	Inteiro	4	Número de nós por elemento
USERR1	Real	1.0	Escala de inserção dos simbolos nodais
USERI2	Inteiro	0	Restrição nodal
USERI3	Inteiro	0	Tipo de material
USERR2	Real	1.0	Escala de inserção dos simbolos de carregamento

Tab V.3

. Inicializar as variáveis do sistema **AutoCAD**, necessáriais a operação desejada do AUTOLISP.

VARIAVEL	TIPO	VALOR INICIAL	DESCRIÇÃO
CMDECHO	Bit	0	Desativa a visualização dos comandos do AutoLISP durante sua execução

Tab V.4

Carregamento do arquivo de ***shapes*** com símbolos de nós e restrições nodais.

O **AutoCAD** oferece ao usuário as variáveis inteiras **USERI1**, **USERI2**, **USERR3**, **USERI4**, **USERI5** e as variáveis reais **USERR1**, **USERR2**, **USERR3**, **USERR4** e **USERR5**, que podem ter seus valores definidos a qualquer momento, sendo estes mantidos mesmo após o término da seção gráfica. Isto permite que trabalhos de discretização de estruturas em elementos finitos, possam ser feitos em várias seções gráficas, sem que para isto seja necessário redefinir todo o ambiente no início de cada seção subsequente.

Uma vez executado, este comando permanecerá válido para todo o processo de discretização da estrutura, mesmo que o trabalho seja interrompido e venha a ser concluído em outra seção gráfica.

V.4.2. O COMANDO MATERIAL

O comando **MATERIAL** é responsável pela criação, alteração e definição das propriedades de materiais e geométricas dos elementos finitos.

Os tipos distintos de materiais são definidos em função de diferentes ***layers***.
Os ***layers*** para materiais terão seus nomes no formato:

```
$MATxx
```

onde **xx** corresponde a um contador numérico referente a ordem de criação dos materiais.

O comando **MATERIAL** possui as seguintes opções de execução:

```
?/CHange/CReat/Make/<Set>/Edit:
```

– A opção ? tem como função apresentar os materiais já definidos bem como suas propriedades .

A seguir apresenta–se um exemplo de utilização do comando material com opção ? .

```
Command> Material
?/CHange/CReat/Make/<Set>/Edit: ?

$MAT01                    **active

Material name: STEELM
E: 210000.0
G: 80769.2266
Poisson: 0.3
Fy: 42000
Fu: 30000
Dens: 0.0
Alpha: 0.0002
Property type: PLATE
Thickness: 0.1

$MAT02

Material name: STEELE
E: 2.9E07
G: 1.1154E07
Poisson: 0.3
Fy: 0.0
Fu: 0.0
Dens: 0.283
Alpha: 0.0
Property type: TUBE
Diameter: 80.0
Thickness: 3.0

Command>
```

Nenhum teste de coerência é efetuado sobre os valores fornecidos para as propriedades. O usuário deverá verificar a consistência dos dados fornecidos com relação as dimensões adotadas para o desenho do modelo. Por exemplo, os desenhos que tiverem o metro como unidade adotada deverão ter suas propriedades expressas em metros, assim como, o Newton para unidade de força. As propriedades deverão ser expressas em unidades compatíveis com as unidades das dimensões.

– A opção **CHANGE** permite ao usuário mudar o tipo de material de um ou mais elementos finitos.
Esta opção funciona com o ***selection set*** padrão do **AutoCAD**, ou seja, possibilita que a seleção dos elementos seja feita por:

. Window
. Crossing
. Previous
. Last
. Single
. Multiple
. Remove
. Add

Abaixo apresenta–se um exemplo de execução do comando **MATERIAL** com opção **CHANGE**:

```
Command> Material
?/CHange/CReat/Make/<Set>/Edit: CHange
Select objects:
New Material:<default>
```

O comando interage com o usuário de modo a obter o conjunto de elementos finitos a terem seus materiais trocados, bem como o novo material. Como *default* para o novo material, o sistema utiliza o material do grupo de elementos selecionados se todos possuírem o mesmo tipo de material, caso contrário, a palavra VARIOUS aparecerá.

O comando realiza uma verificação quanto a validade do novo material para o conjunto de elementos finitos selecionado. O sistema não permite que um usuário desatento, atribua a um conjunto de elementos reticulados, propriedades de elementos de placa ou casca ou vice–versa. Para tais casos, o sistema emitirá uma mensagem de alerta e a operação será abandonada.

```
Invalid property for set of elements
Command>
```

– A opção **CREAT** é responsável pela criação de novas propriedades e/ou tipos de materiais.

Para a definição das propriedades do material, os dados requisitados ao usuário são:

Material Name	=	Nome do Material
E	=	Modulo de Elasticidade
G	=	Modulo Cortante
Poisson	=	Coeficiente de Poisson
Fy	=	Tensão de Escoamento
Fu	=	Tensão de Ruptura
Dens	=	Densidade
Alpha	=	Coeficiente de Dilatação Térmica

Para a definição das propriedades geométricas, a opção **CREAT** do comando **MATERIAL** possui duas sub–opções:

Bar/<Plate>:

– O sub–opção **PLATE** é adotada como *default* para este comando, e especifica que os elementos finitos associados a esta propriedade serão do tipo placa ou casca.

A propriedade geométrica a qual é requisitada a definição ao usuário é:

Thickness = Espessura do Elemento

– A opção **BAR** refere–se à definição das propriedades geométricas de elementos finitos do tipo reticulado. Para este tipo de elemento finito, os dados requisitados ao usuário dizem respeito à definição das dimensões da seção transversal.

Estão disponíveis algumas seções transversais básicas, as quais serão atribuídas os valores das dimensões e uma seção de formato genérico cujas propriedades deverão ser previamente calculadas e posteriormente fornecidas.

As opções de seção transversal para elementos do tipo reticulado são:

Prismat/<Tube>/Box/Elliptic/I/T/L/U:

– Para as seções transversais do tipo **PRISMAT**, o usuário será requisitado a selecionar um perfil transversal previamente desenhado. Para os demais tipos de seção transversal, o sistema automaticamente construirá o perfil da seção trasversal, segundo as dimensões fornecidas pelo usuário.

A opção **PRISMAT** permite ao usuário definir uma seção transversal de formato qualquer, diferente dos perfis comumente usados.

A sequência de operações para a definição de uma seção transversal de formato genérico é apresentada a seguir:

```
Command> Material
?/CHange/CReat/Make/<Set>/Edit: Creat
Material name: M1
E: 17000
G: 50000
Poisson: 0.25
Fy: 55000
Fu: 16000
Dens: 1500
Alpha: 0.03
Bar/<Plate>: Bar
Prismat/<Tube>/Box/Elliptic/I/T/L/U: Prismat
Select section:
Ixx: 1.52
Iyy: 1.52
Jxy: 0.90
Command>
```

Os dados requisitados ao usuário, especificamente para definição das propriedades da seção transversal para uma seção do tipo **PRISMAT** são, portanto:

Select Section	=	Formato da Seção Transversal
Ixx	=	Momento de Inércia na Direção X
Iyy	=	Momento de Inércia na Direção Y
Jxy	=	Momento de Inércia Transversal

A área e o perímetro da seção selecionada são automaticamente determinados pelo sistema.

A forma da seção transversal é definida pela seleção de um polígono fechado, não vazado e com número de vértices qualquer. Isto correponde a dizer que a seção transversal deverá ser representada por um único ***polyline*** fechado.

A figura V.10 apresenta alguns exemplos de perfis prismáticos que podem ser definidos:

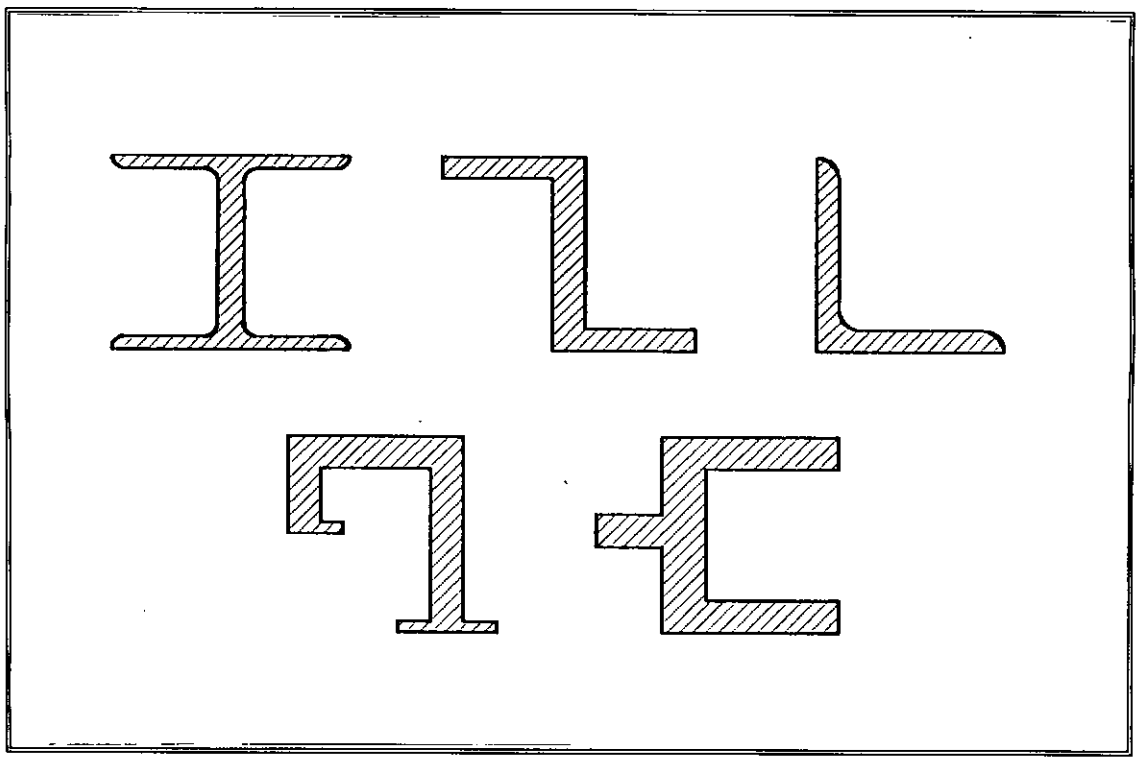


Fig V.10

. Para a definição das propriedades de seções transversais padronizadas, o usuário será requisitado a fornecer apenas as dimensões.

– A opção **TUBE** é adotada como *default* pelo sistema para a definição das propriedades geométricas para elementos finitos do tipo reticulado.

Os únicos dados necessários para a sua definição são:

Diam: = Diâmetro Externo do Tubo

Thickness: = Espessura da Parede do Tubo

A figura V.11 apresenta uma seção transversal do tipo **TUBE**:

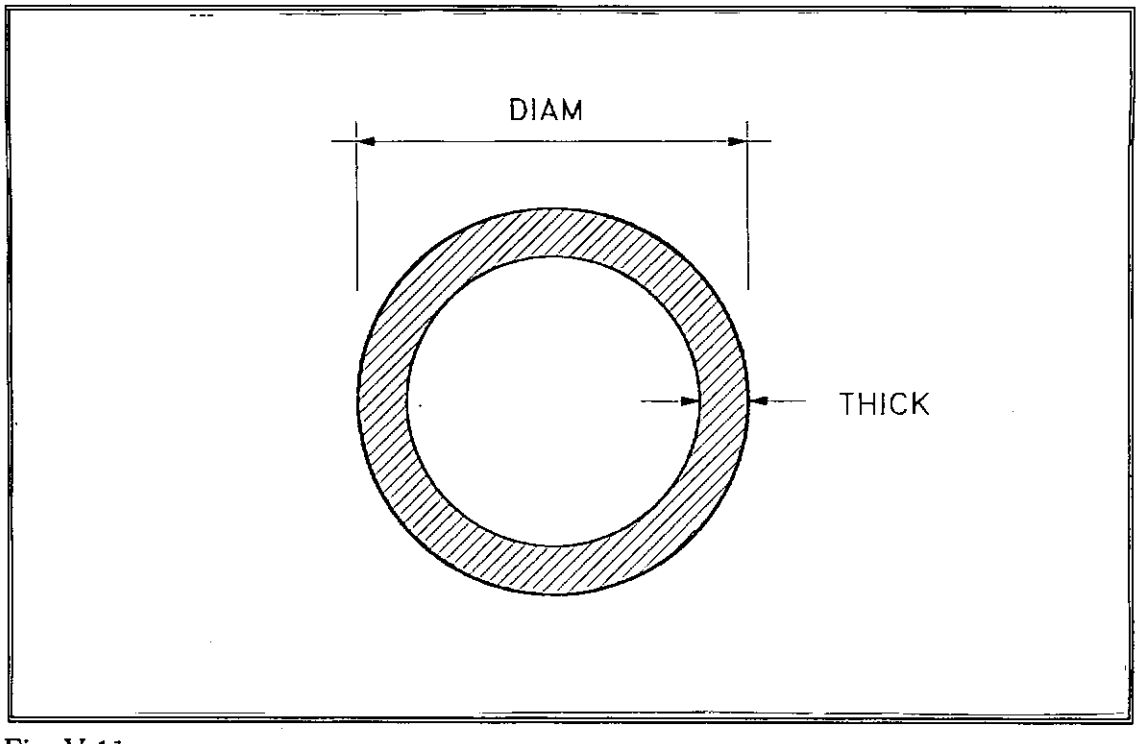


Fig V.11

– A opção **BOX** permite ao usuário definir propriedades de elementos reticulados com seção transversal do tipo "caixão".

Os dados requisitados ao usuário para definição das dimensões são:

```
Width:              = Largura da Seção Transversal
Depth:              = Altura da Seção Transversal
Wall Width Thickness: = Espessura da Parede na Largur
Wall Depth Thickness: = Espessura da Parede na Altura
```

A figura V.12 apresenta uma seção transversal do tipo BOX:

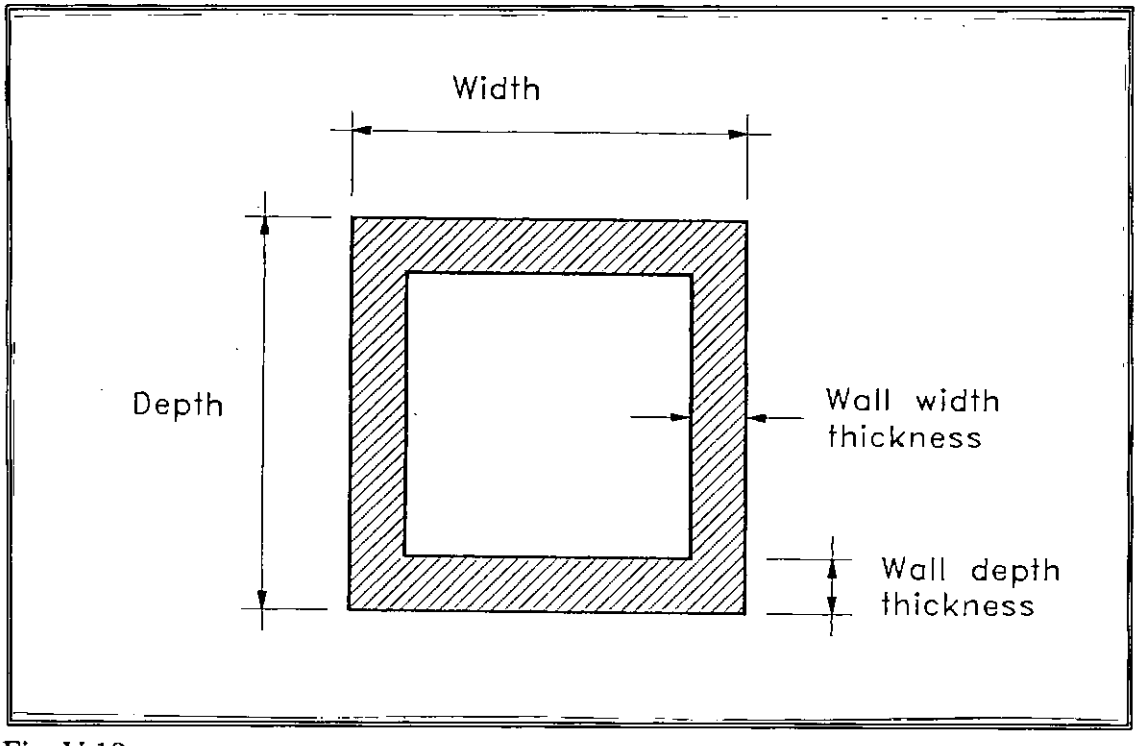


Fig V.12

– A opção **ELLIPTIC** permite ao usuário definir propriedades de elementos reticulados com seção transversal elíptica.

Os dados requisitados ao usuário para definição das dimensões da seção são:

```
Width:              = Largura da Seção Transversal
Depth:              = Altura da Seção Transversal
Wall Width Thickness: = Espessura da Parede na Largura
Wall Depth Thickness: = Espessura da Parede na Altura
```

A figura V.13 apresenta uma seção transversal do tipo **ELLIPTIC**:

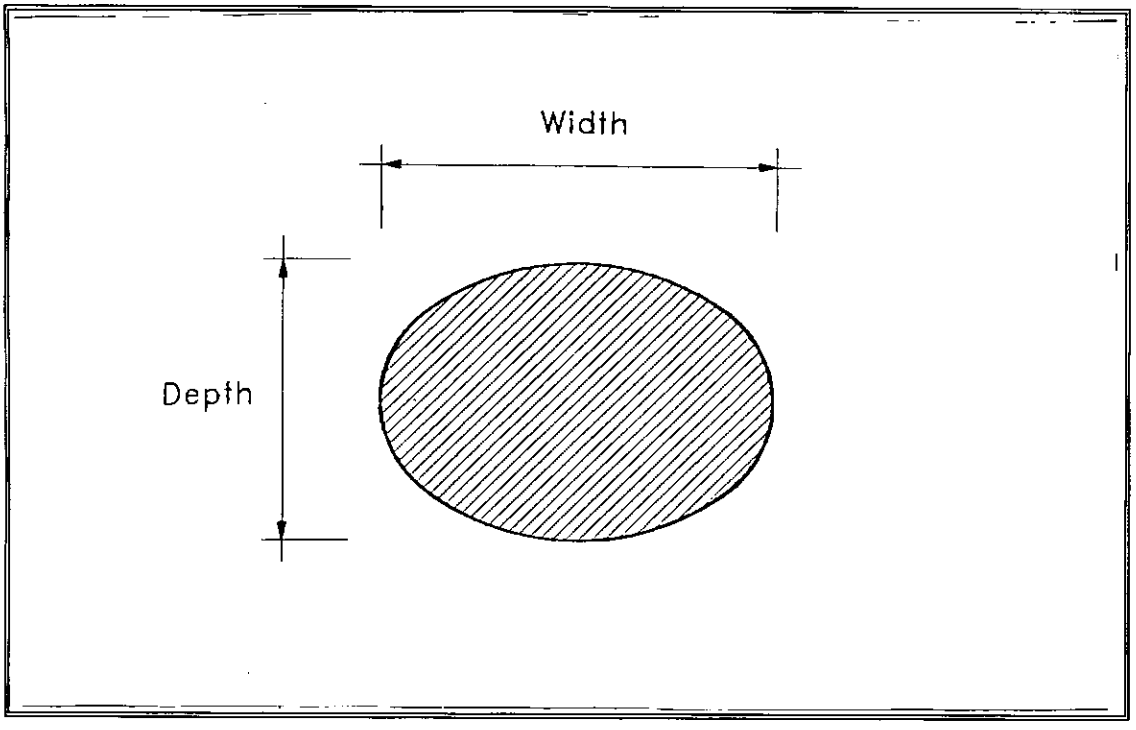


Fig V.13

– A opção **I** permite ao usuário definir propriedades de elementos do tipo reticulado com seção transversal do tipo I, H ou duplo T como também são chamados.

Os dados requisitados ao usuário para definição das dimensões são:

```
Width:              =  Largura da Seção Transversal
Depth:              =  Altura da Seção Transversal
Flange Thickness: =  Espessura da Mesa
Web Thickness       =  Espessura da Malha
```

A figura V.14 apresenta uma seção transversal do tipo I:

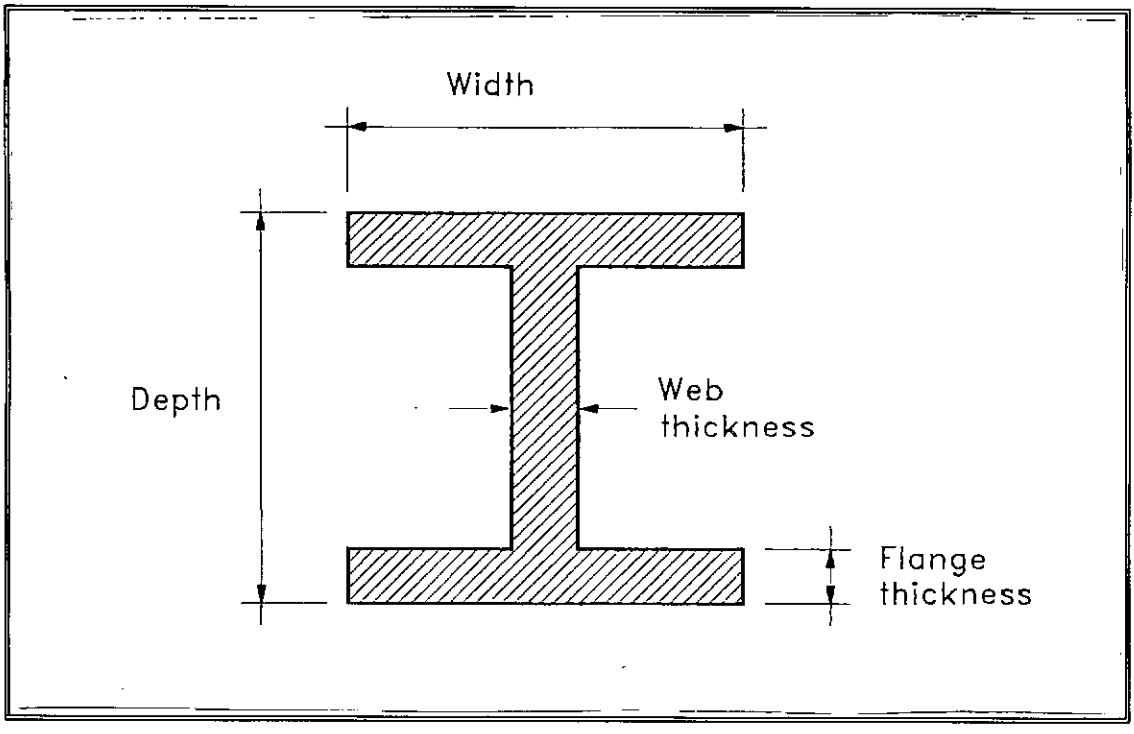


Fig V.14

– A opção T permite ao usuário definir propriedades de elementos reticulados com seção transversal do tipo T.

Os dados requisitados ao usuário para definição das dimensões da seção são:

Width:	=	Largura da Seção Transversal
Depth:	=	Altura da Seção Transversal
Flange Thickness:	=	Espessura da Mesa
Web Thickness	=	Espessura da Malha

A figura V.15 apresenta uma seção transversal do tipo T:

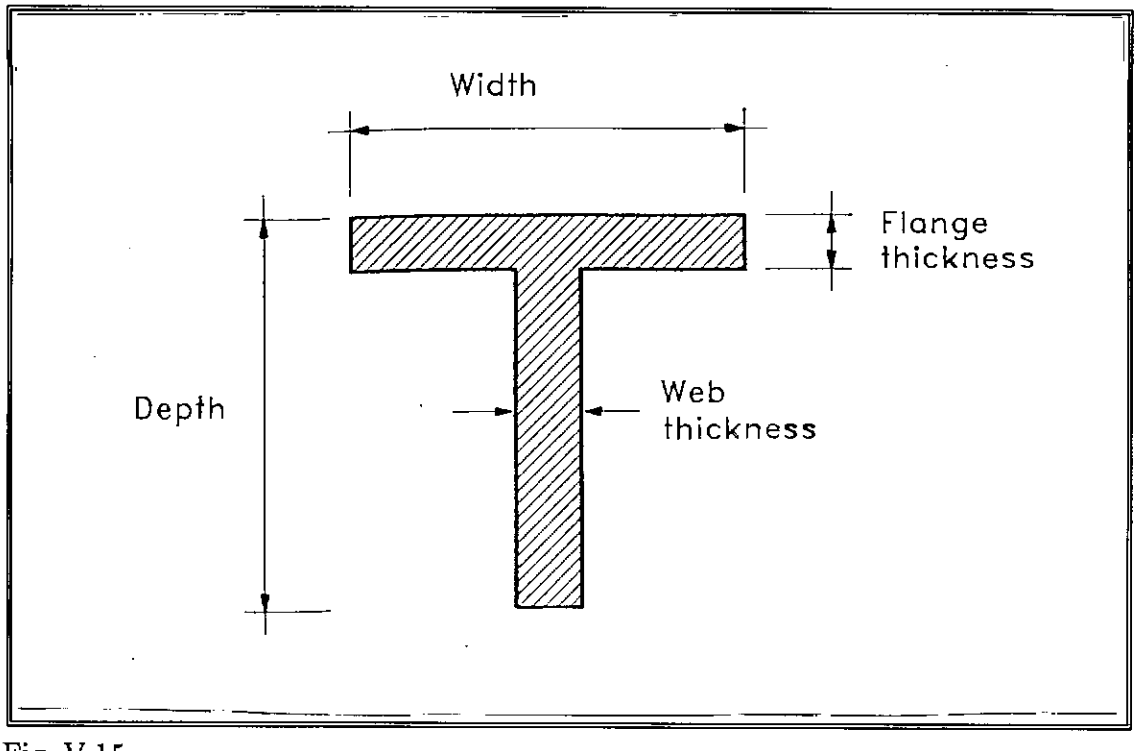


Fig V.15

– A opção L permite ao usuário definir propriedades de elementos reticulados com seção transversal do tipo L ou "cantoneira" como também são chamados.

Os dados requisitados ao usuário para definição das dimensões da seção são:

Width:	=	Largura da Seção Transversal
Depth:	=	Altura da Seção Transversal
Flange Thickness:	=	Espessura da Mesa
Web Thickness	=	Espessura da Malha

A figura V.16 apresenta uma seção transversal do tipo L:

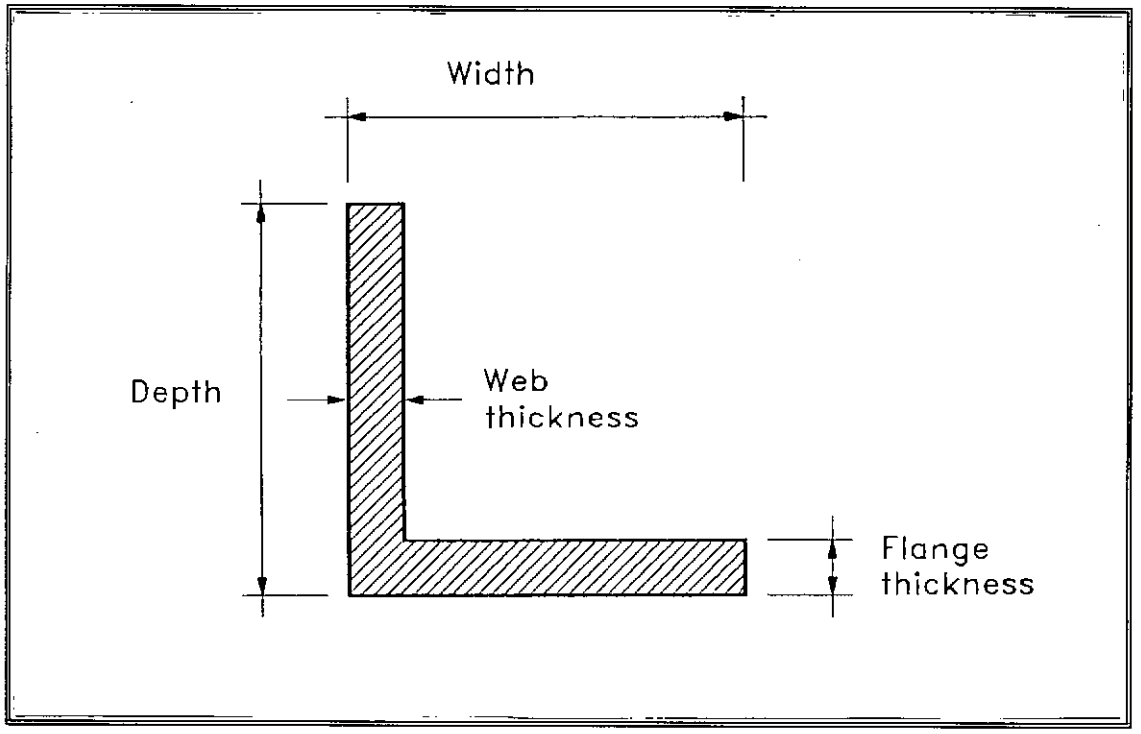


Fig V.16

– A opção U permite ao usuário definir propriedades de elementos reticulados com seção transversal do tipo U, "calha" ou duplo L como também são chamados.

Os dados requisitados ao usuário para definição das dimensões da seção são:

```
Width:              = Largura da Seção Transversal
Depth:              = Altura da Seção Transversal
Flange Thickness:   = Espessura da Mesa
Web Thickness       = Espessura da Malha
```

A figura V.17 apresenta uma seção transversal do tipo U:

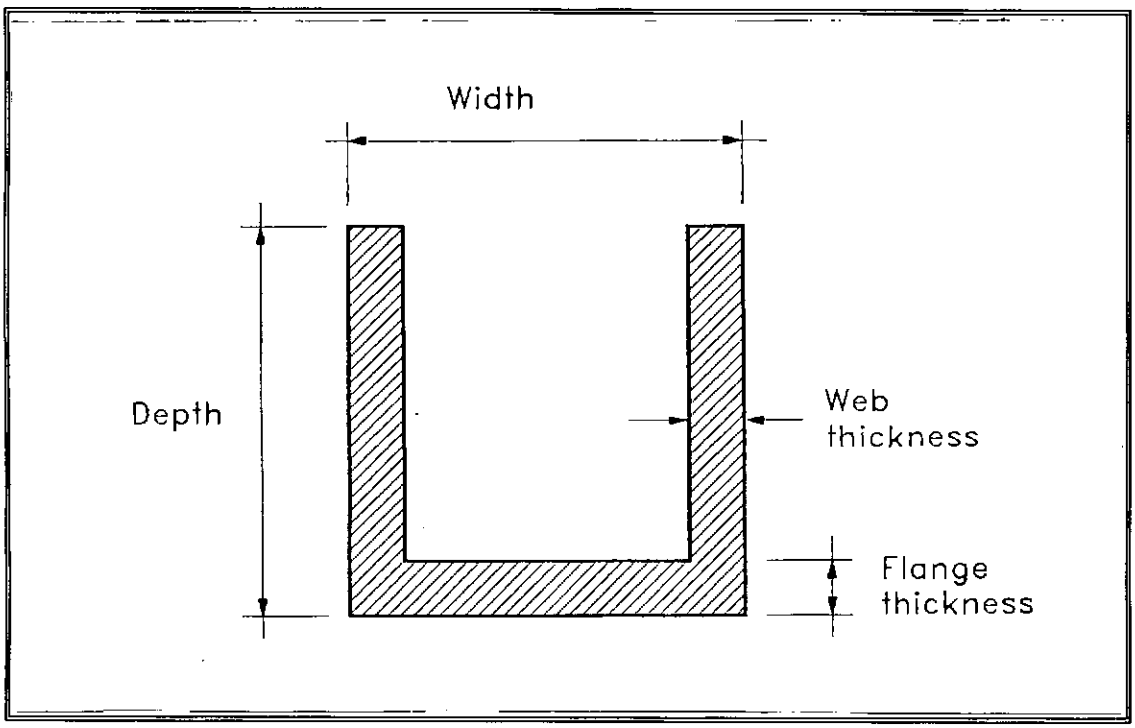


Fig V.17

– A opção **MAKE** do comando **MATERIAL** permite que, além de criado o material, seja definido como *default* para subsequentes operações de inserção de elementos finitos no modelo.

A sequência de operações para o comando material com opção **MAKE** é exatamente igual a sequência de operações para a opção **CREAT**.

O uso da opção **MAKE** do comando **MATERIAL** é equivalente ao uso da opção **CREAT** sequida da opção **SET** para o novo material.

– A opção **SET** do comando **MATERIAL** permite ao usuário definir um material previamente criado como *default*, para todas as subsequentes operações de inserção de elementos finitos no modelo.
Todo elemento finito criado e inserido no modelo estará necessariamente associado a um material. O material adotado pelo sistema para associá–lo aos novos elementos finitos, será o especificado pela última execução do comando **MATERIAL** com opções **MAKE** ou **SET**.

– A opção **EDIT** do comando material permite ao usuário alterar qualquer dos valores atribuídos às propriedades dos materiais ou geométricas.

O usuário poderá alterar o valor de apenas uma especificação de propriedade do material.

A seguir apresenta–se uma sequência de operações parciais do comando **MATERIAL** opção **EDIT**:

```
Command> Material
?/CHange/CReat/Make/<Set>/Edit: Edit
Material name: M1
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>:
```

Após selecionada a opção **EDIT**, o usuário é requisitado a determinar qual o material que terá suas propriedades alteradas.
A seguir um conjunto de sub–opções é apresentado

```
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>:
```

Este conjunto de sub–opções corresponde às diferentes propriedades de material disponíveis ao usuário para atribuição de valores, com exceção das duas últimas (**GEometry** e **eXit**).

A seleção de qualquer das opções referentes às propriedades do material possibilita ao usuário a alteração de seu valor. O sistema requisitará ao usuário um novo valor para a propriedade em edição e assumirá como *default* o valor atual. Após alteração ou aceitação do atual valor, o sistema retornará ao conjunto de opções de edição, até que a opção **EXIT** seja selecionada. Isto possibilita que mais de uma propriedade seja alterada para cada material selecionado.

A seguir apresenta–se a sequência de operações para alteração dos valores referentes às propriedades do material:

```
Command> Material
?/CHange/CReat/Make/<Set>/Edit: Edit
MAterial name: M1
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: E
New E:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: G
New G:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: P
New Poisson:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: Fy
New Fy:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: Fu
New Fu:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: D
New Dens:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: A
New Alpha:<default>
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: X
Command>
```

– A opção **GEOMETRY** permite ao usuário realizar alterações nas propriedades geométricas.

Para propriedades de materias associadas a elementos de casca ou placa, o sistema requisitará ao usuário um novo valor para a espessura do elemento e assumirá como *default* o atual valor.

A sequência de operações para este tipo de material é apresentada a seguir:

```
Command> Material
?/CHange/CReat/Make/<Set>/Edit: Edit
Material name: M1
E/G/Poisson/Fy/Fu/Dens/Alpha/GEometry/<eXit>: GE
Prismat/Tube/Box/Elliptic/I/T/L/U:<default>
```

Dependendo do tipo de seção selecionada, o sistema requisitará ao usuário os dados necessários.

Para todos os tipos de seção são apresentadas as dimensões atuais como *default* e ao usuário é permitida a alteração destes valores.

Para seções transversais do tipo prismático, o usuário é requisitado a indicar um novo polígono para definir a forma da seção transversal.

Para as propriedades geométricas da seção transversal, o sistema apresenta os valores atuais como *default* e requisita ao usuário novos valores.

A seguir apresenta–se os *prompts* do sistema para cada tipo de seção transversal:

```
Prismat   = Select New Section:
            New Area:<default>
            New Perimeter:<default>
            New Ixx:<default>
            New Iyy:<default>
            New Jxy:<default>
Tube      = New Diameter:<default>
            New Wall Thickness:<default>
Box       = New Width:<default>
            New Depth:<default>
            New Wall Width Thickness:<default>
```

```
                    New Wall Depth Thickness:<default>
Elliptic        =   New Width:<default>
                    New Depth:<default>
I, T, L, U      =   New Width:<default>
                    New Depth:<default>
                    New Flange Thickness:<default>
                    New Web Thickness:<default>
```

Deve-se notar que, apesar do sistema automaticamente calcular a área e o perímetro de uma seção transversal do tipo **PRISMATIC**, é permitido ao usuário alterar estes valores.

Como nas opções **CREAT** e **MAKE** do comando **MATERIAL**, aqui também o usuário deverá verificar a coerência dimensional dos valores fornecidos às propriedades com as unidades adotadas para definição das dimensões do modelo.

V.4.3. O COMANDO NODE

O comando **NODE** é responsável pela definição da dimensão e inserção de pontos nodais no modelo.

O comando **NODE** possui duas opções de execução, a saber:

Scale/<Point>:

– A opção **SCALE** permite ao usuário definir a escala ativa, ou seja, o tamanho do símbolo nodal para as subsequentes operações de inserção de pontos nodais no modelo, quer isoladamente, quer por comando de edição ou geração automática.

O sistema requisitará ao usuário um novo fator de escala para o símbolo nodal, apresentando o atual como *default.*

Node Scale:<*default*>

Um valor unitário atribuído ao fator de escala, fará com que todos os símbolos nodais subsequentemente inseridos no modelo tenham diâmetro unitário. O valor nulo quando atribuído ao fator de escala para o símbolo nodal, informa ao sistema que o usuário não deseja a inserção de símbolos nodais no desenho.

Deve-se salientar que o símbolo nodal não tem função estrutural, apenas visual, possibilitando ao usuário distinguir com facildade elementos de primeira e segunda ordem do mesmo tipo.

A figura V.18 demonstra o caso de elementos finitos triangulares semelhantes com e sem símbolos nodais.

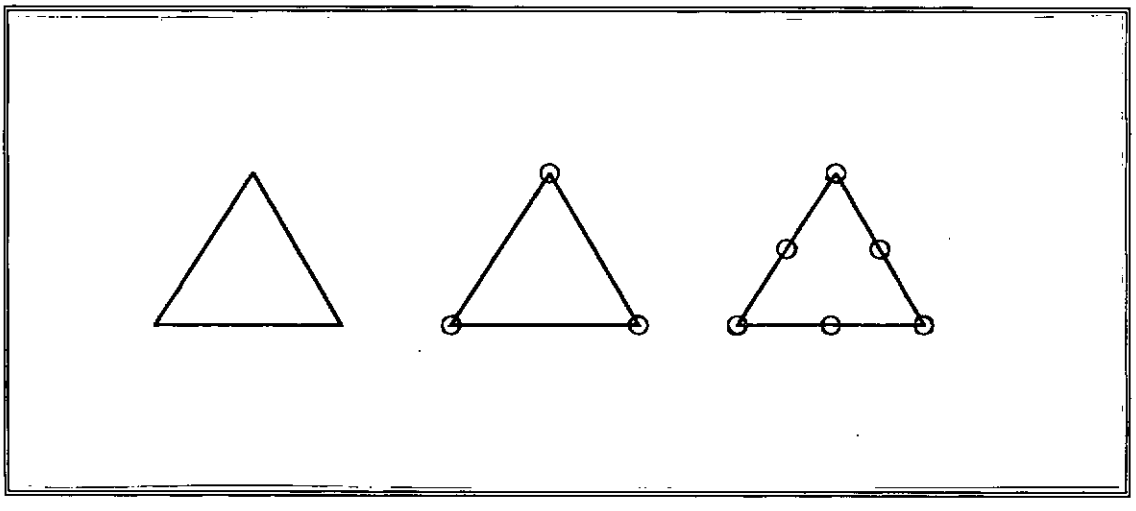


Fig V.18

– A opção **POINT** permite ao usuário inserir um símbolo nodal no ponto desejado, segundo os padrões do **AutoCAD** para inserção de entidades gráficas no desenho. Esta inclusão poderá portanto, ser definida por coordenadas polares, relativas, locais ou iterativamente, ou ainda auxiliada pelos comandos ***snap***, ***ortho*** ou ***osnap*** do **AutoCAD**.

A opção **POINT** é adotada como *default* para o comando **NODE.**

O símbolo nodal normalmente reside no ***layer*** **NODES** puramente por razões de organização. Devido ao fato de possuir função puramente visual, nada impede que possa ser transferido para outros ***layers***

V.4.4. O COMANDO ELEMENT

O comando **ELEMENT** detém como funções a definição do tipo de elemento ativo e a inserção isolada de elementos finitos no modelo.

O comando **ELEMENT** possui duas opções de execução:

Type/<Place>:

– A opção **TYPE** permite ao usuário definir o tipo de elemento ativo, dentre um conjunto de tipos de elementos finitos disponíveis no sistema **AutoMESH.**

Ao executar o comando **ELEMENT** com opção **TYPE,** o usuário será requisitado a selecionar um novo tipo de elemento finito ativo. O sistema apresenta como *default* o tipo de elemento ativo no momento da execução do comando.

Bar/ISobar/Tri/Quad/Hex/Oct/Contact/INfinite:<*default*>

A tabela V.19 descreve a nomenclatura adotada para especificar os tipos de elementos disponíveis.

DESCRIÇÃO	ELEMENTOS	
	primeira ordem	segunda ordem
Elementos Reticulados	BAR (2 nós)	ISOBAR (3 nós)
Elementos Triangulares	TRI (3 nós)	HEX (6 nós)
Elementos Quadrilateros	QUAD (4 nós)	OCT (8 nós)
Elemento de Contato		CONTACT (6 nós)
Elemento Infinito		INFINITE (5 nós)

Fig V.19

As figuras V.20 e V.21 apresentam a representação gráfica para cada um dos tipos de elementos finitos.

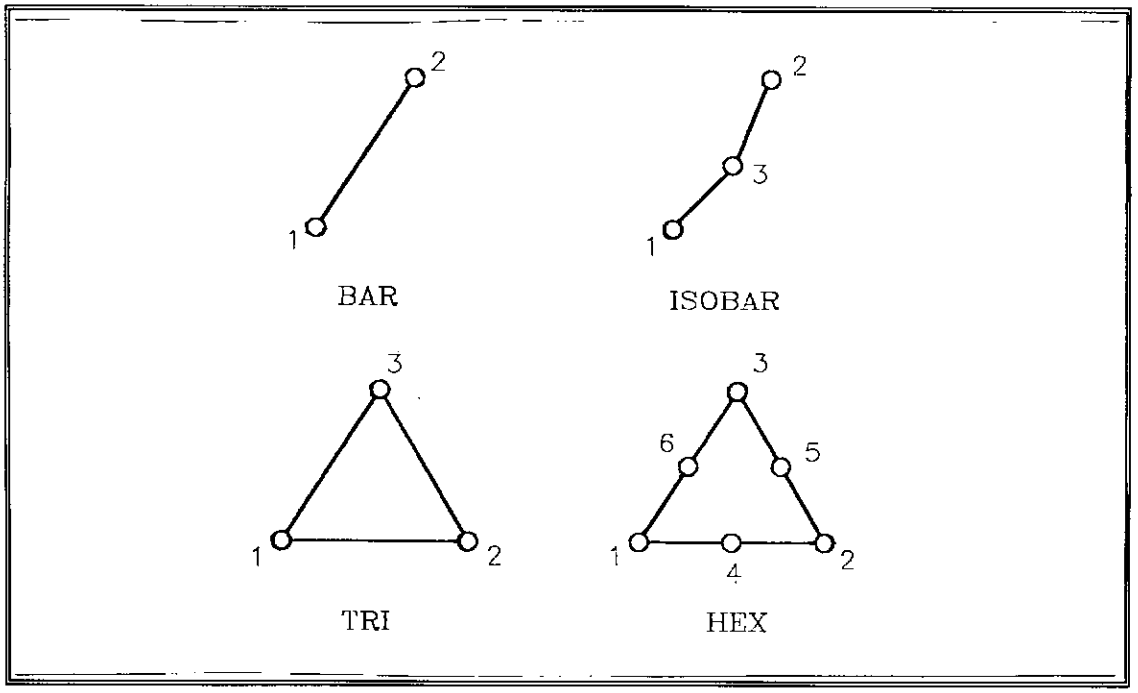


Fig V.20

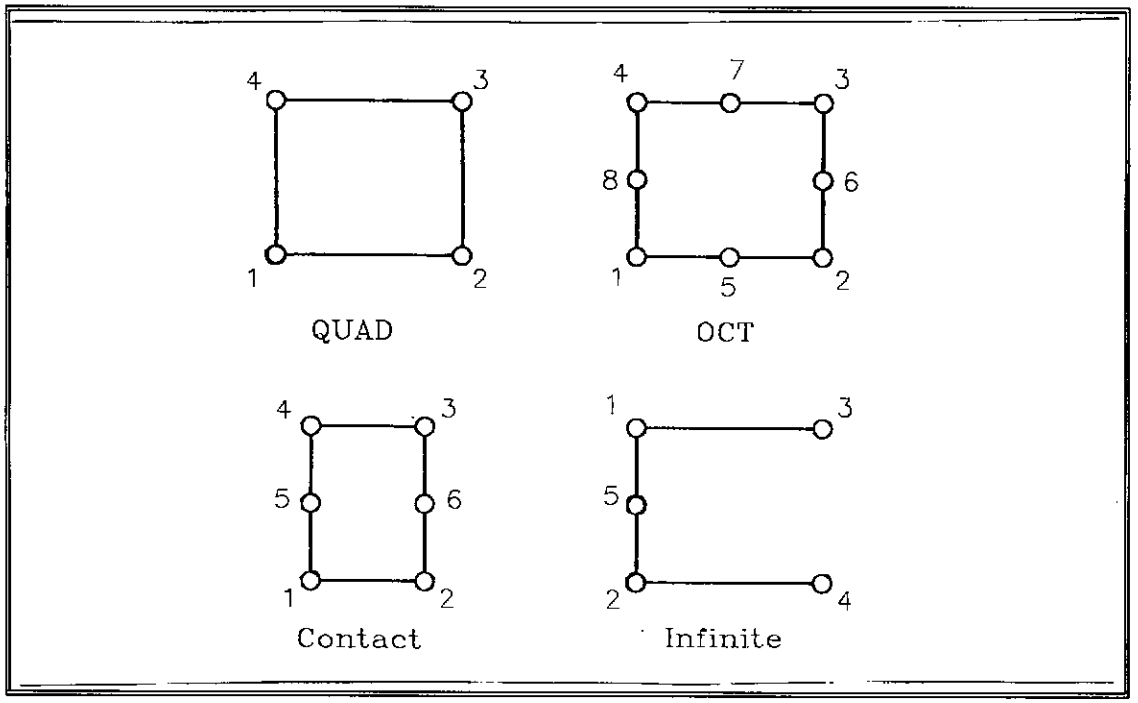


Fig V.21

Uma vez definido o tipo de elemento ativo, este permanecerá válido para para todas as operações de inserção individual, cópia ou geração automática de elementos finitos subsequentes.

– A opção **PLACE** é adotada como *default* para este comando e corresponde à função de inserção individual de elementos finitos no modelo.

O sistema requisitará ao usuário a identificação dos pontos nodais, segundo a ordem apresentada nas figuras V.20 e V.21, de acordo com o tipo de elemento ativo.

Por exemplo, a sequência de operações para a inserção de um elemento do tipo **HEX** no modelo, exigiria a definição de seis pontos como mostrado a seguir.

```
Command> Element
Type/<Place>: Place
First Node: 0,0,0
Second Node: 20,0,0
Third Node: 0,20,0
Fourth Node: 10,0,0
Fifth Node: 10,10,0
Sixth Node: 0,10,0
Command>
```

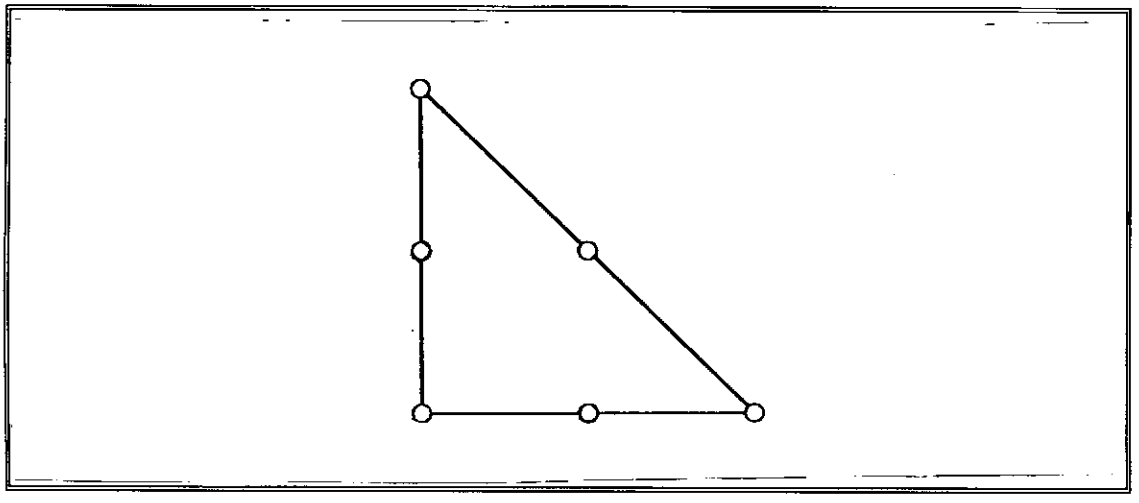


Fig V.22

Antes de realizar qualquer operação de inserção ou criação de elementos finitos no modelo, o sistema verificará a associação deste elemento com o material ativo. Verificada a coerência, o processo se dá automaticamente. A tentativa de associar um tipo de elemento a uma propriedade incompatível, faz com que o sistema emita uma mensagem de aviso e cancele a operação.

Durante a execução de qualquer dos comandos de inserção de elementos finitos no modelo, quer individualmente, quer por comando de edição ou geração automática, o usuário estará informado sobre a propriedade e o tipo de elemento ativos.

A figura V.23 apresenta o aspecto da tela gráfica do **AutoCAD** durante a operação do comando **ELEMENT**.

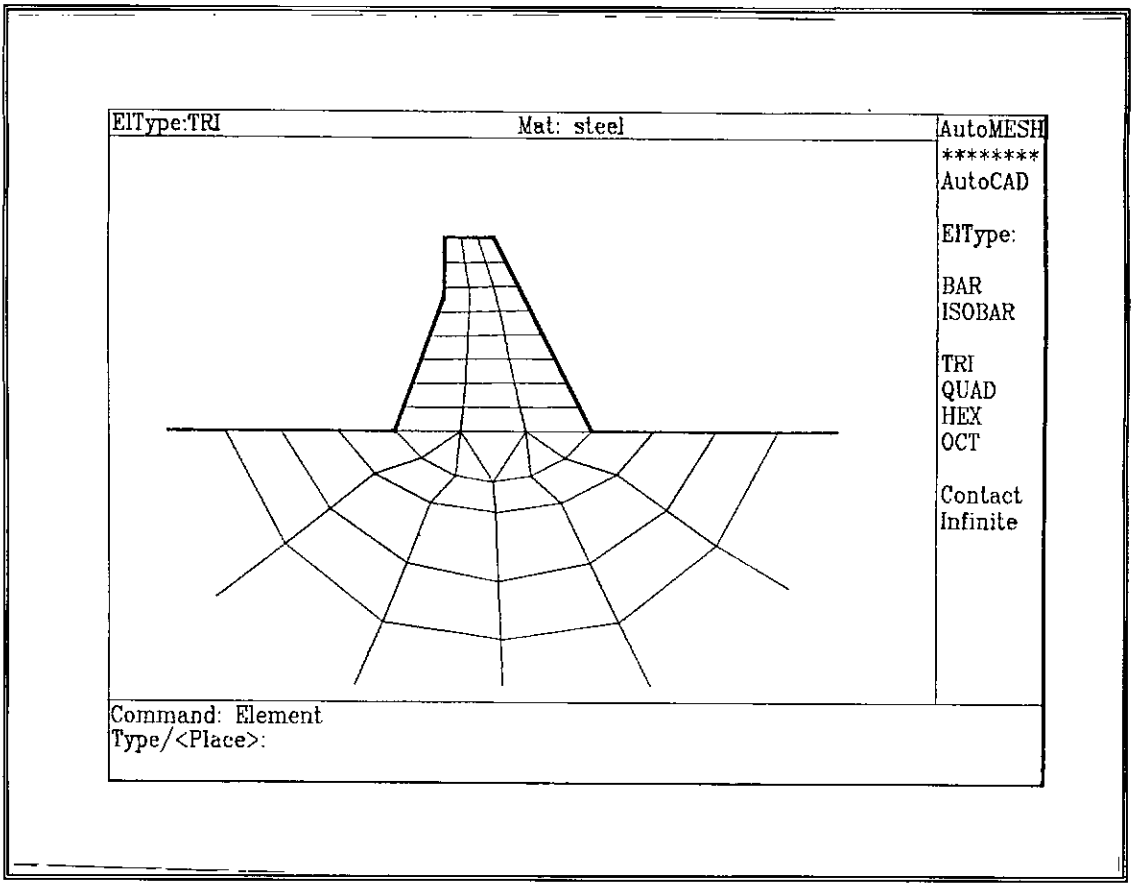


Fig V.23

Os elementos finitos inseridos no modelo ficarão obrigatoriamente residentes em ***layers*** do tipo **$MAT01**, **$MAT02** e etc., os quais correspondem às propriedades dos materias a que estão associados.

Os elementos finitos em sua representação só podem ser distinguidos das demais entidades gráficas pelos ***layers*** aos quais pertencem. Isto significa que ***polylines***, ***3dfaces*** ou ***3dlines*** construídos nestes ***layers***, poderão ser tratados como elementos finitos na fase de geração do arquivo de dados. O usuário deverá estar alerta para evitar este tipo de problema, o que poderia causar inconsistências no modelo e arquivo de dados.

V.4.5. COMANDOS PARA GERAÇÃO AUTOMÁTICA

Os comandos para geração automática compõem um sub–grupo do grupo **CREAT** de comandos do sistema **AutoMESH.**

Estes comandos têm como função a discretização automática de regiões do modelo definidas pelo usuário em malhas de elementos finitos.

Os comandos para geração automática podem ser divididos em dois grupos básicos, de acordo com os tipos de região que são capazes de discretizar, a saber:

. Regiões de Formato Genérico

. Regiões de Formato Específico

O primeiro grupo corresponde aos comandos onde a discretização é realizada sobre uma região de formato qualquer, definida pelo usuário juntamente com os demais dados.

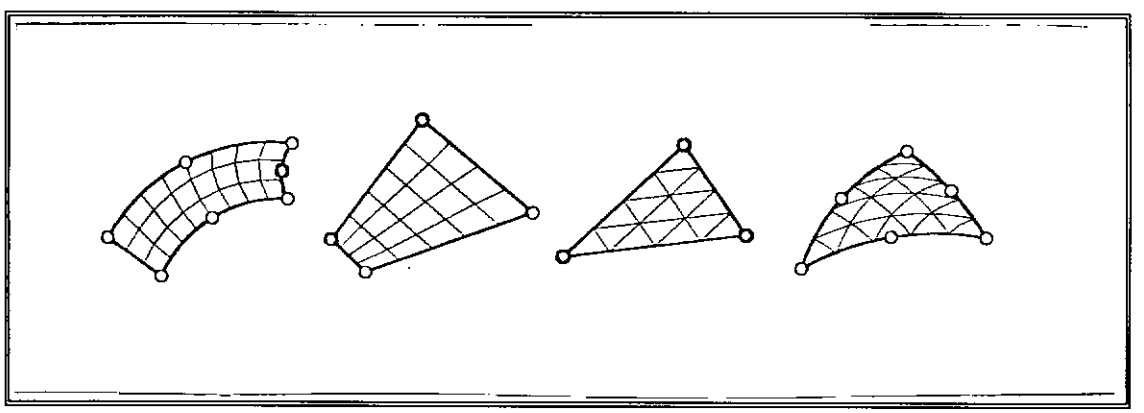


Fig V.24

O segundo grupo é composto dos comandos de discretização de regiões de formato definido, onde o usuário tem controle apenas sobre as dimensões e grau de refinamento da malha.

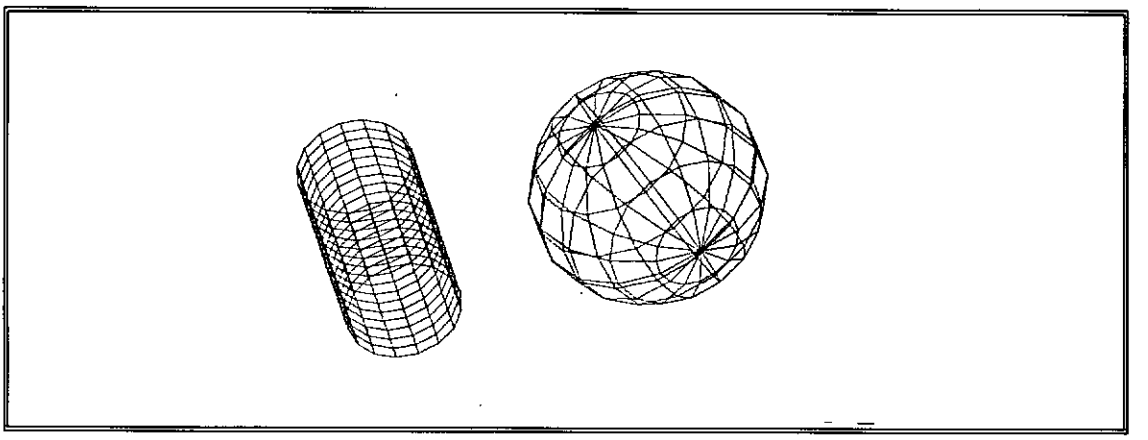


Fig V.25

V.4.5.1. GERAÇÂO AUTOMÁTICA EM REGIÕES DE FORMATO GENÉRICO

V.4.5.1.1. O COMANDO ISOMAPP

O comando **ISOMAPP** executa a discretização de uma região quadrilátera de arestas retas ou parabólicas em uma malha de elementos finitos, pelo método de mapeamento isoparamétrico.

A região a ser discretizada é definida por quatro vértices e quatro pontos que definem os pontos médios das arestas.

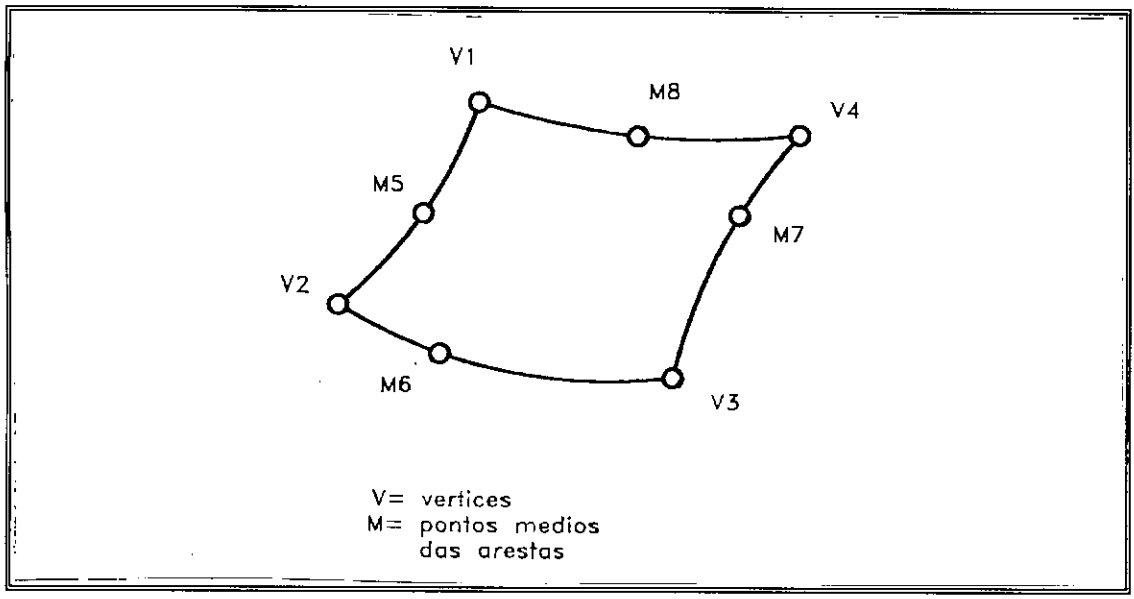


Fig V.26

Os vértices deverão ser obrigatoriamente definidos. Já os pontos médios podem ser omitidos, fazendo assim com que a aresta seja linear e a distribuição dos nós uniforme.

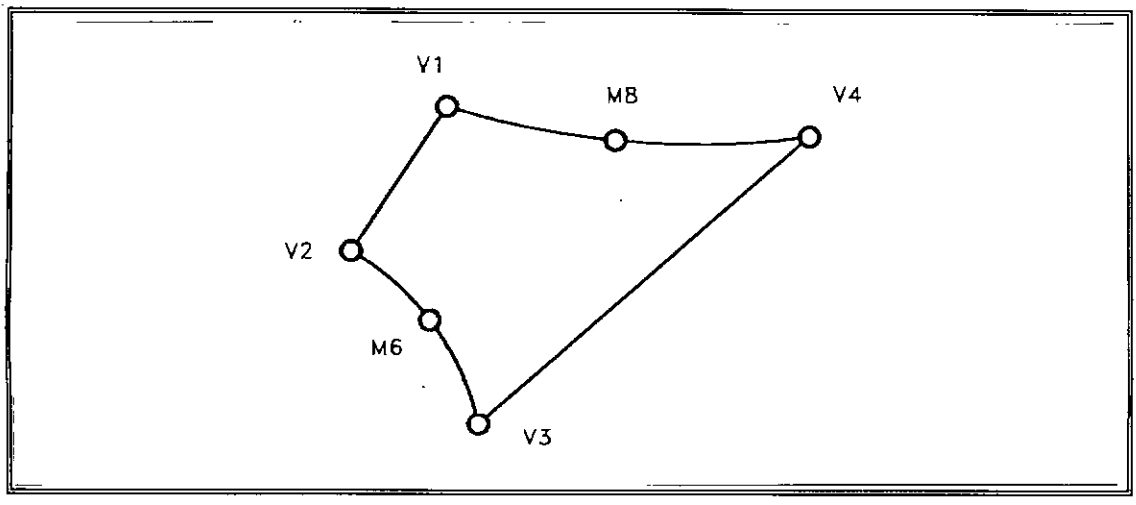


Fig V.27

Os pontos médios também controlam a distribuição dos nós na aresta.

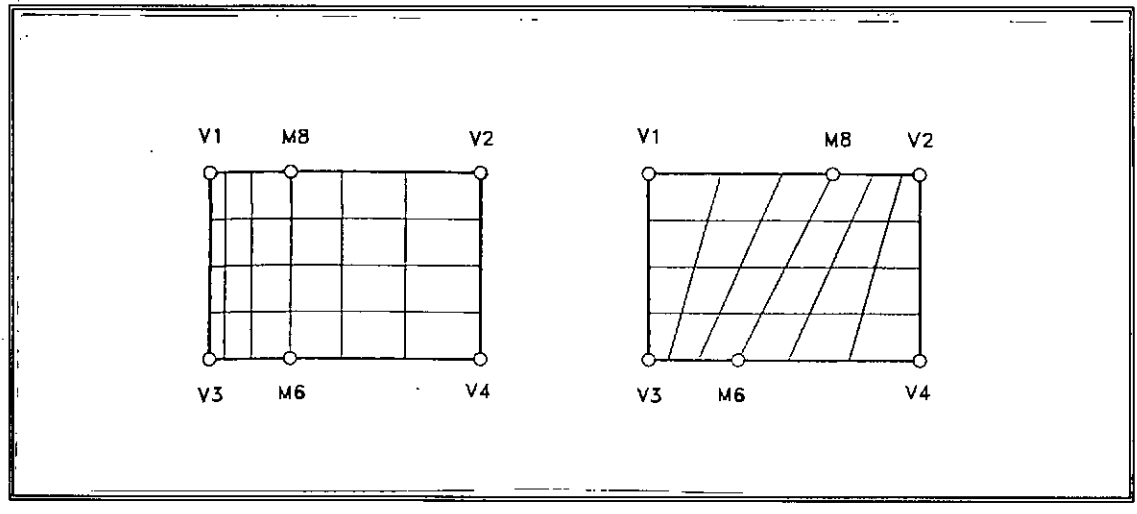


Fig V.28

O nível de refinamento da malha é definido em número de colunas e linhas de elementos.

O número de colunas corresponde ao número de partições entre o primeiro e o quarto vértices, enquanto que o número de linhas corresponde ao número de partições entre os vértices um e dois.

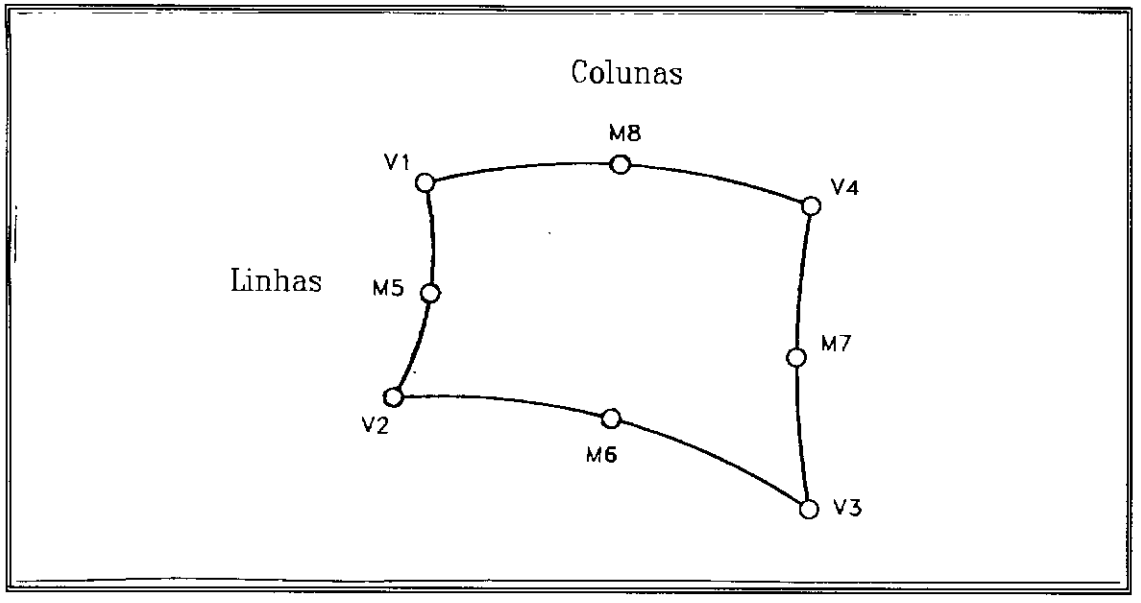


Fig V.29

A versão implementada deste comando opera para elementos quadriláteros de quatro ou oito nós e elementos triangulares de três ou seis nós.

Os elementos representados por ***polylines*** são inseridos no ***layer*** referente ao tipo de material ativo. O tipo de elemento que construirá a malha é determinado pelo tipo de elemento ativo fixado com o comando **ELEMENT**.

A geração da malha para elementos triangulares é baseada em elementos quadráticos, os quais são divididos em dois triângulos.

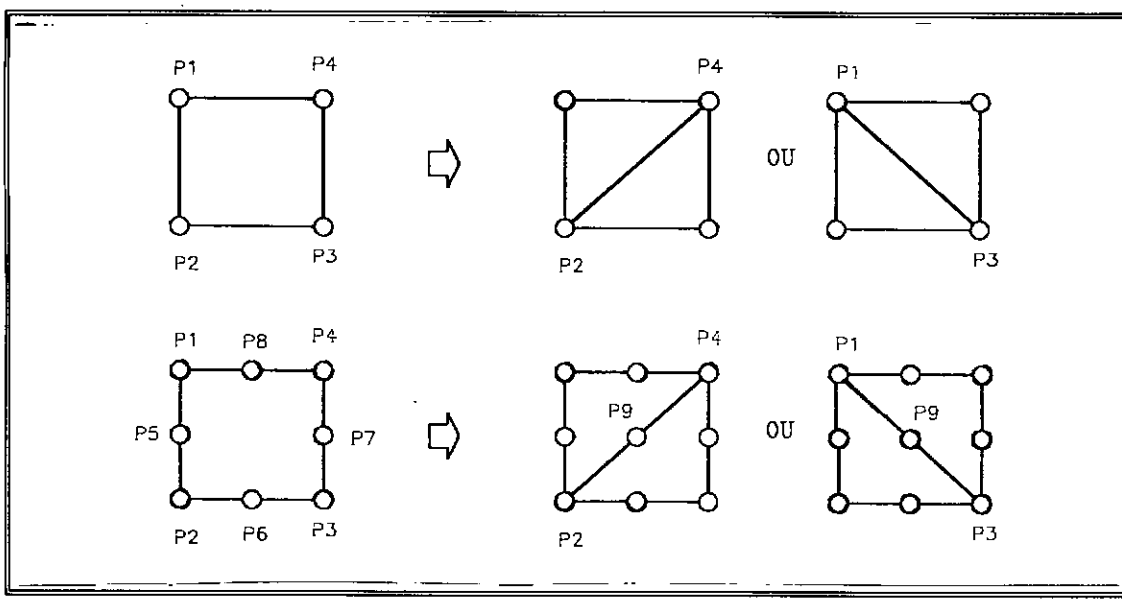


Fig V.30

A determinação do par de triângulos que melhor substitui o quadrado, é feita pela comparação das distâncias entre os vértices de diagonais opostas. Os triângulos serão criados com base na menor diagonal, obtendo-se assim elementos menos distorcidos, com ângulos internos nos vértices menos agudos e consequentemente melhor condicionados.

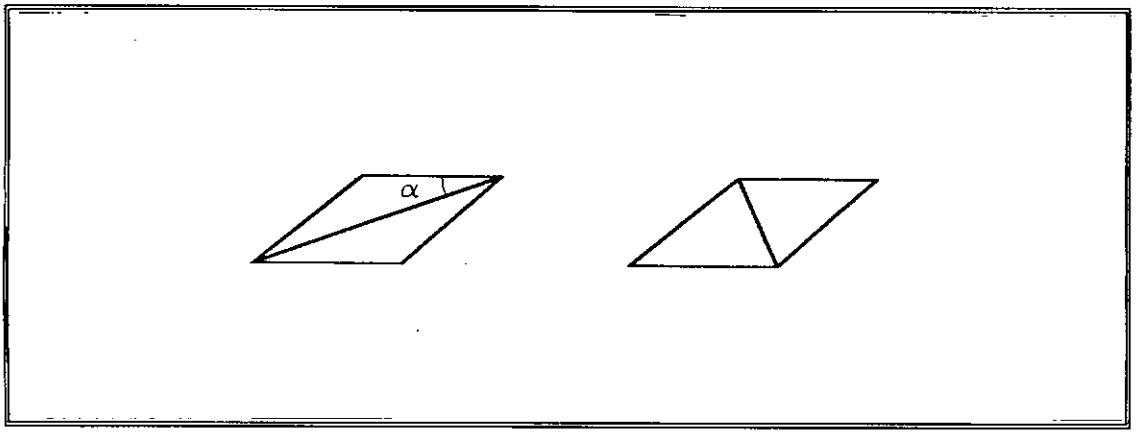


Fig V.31

Atenção também deverá ser tomada na definição da região a ser discretizada, pois devido as características do método utilizado (mapeamento isoparamétrico), é possível a introdução de mau condicionamento de elementos no processo de geração.

Um dos problemas mais comuns é a construção de elementos excessivamente distorcidos.

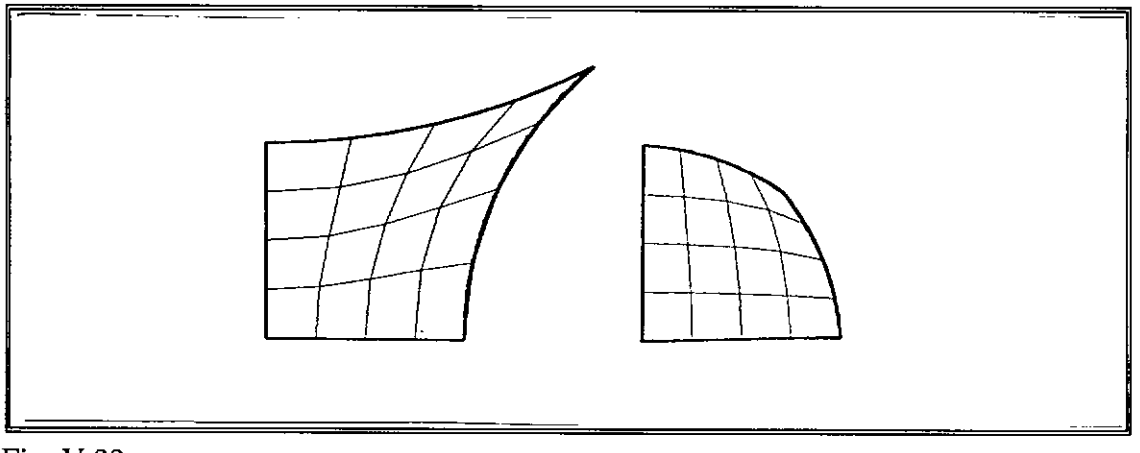


Fig V.32

Outro problema pode ocorrer com regiões excessivamente côncavas, onde haverá superposição de elementos ou mesmo ultrapassagem de fronteiras da região por estes.

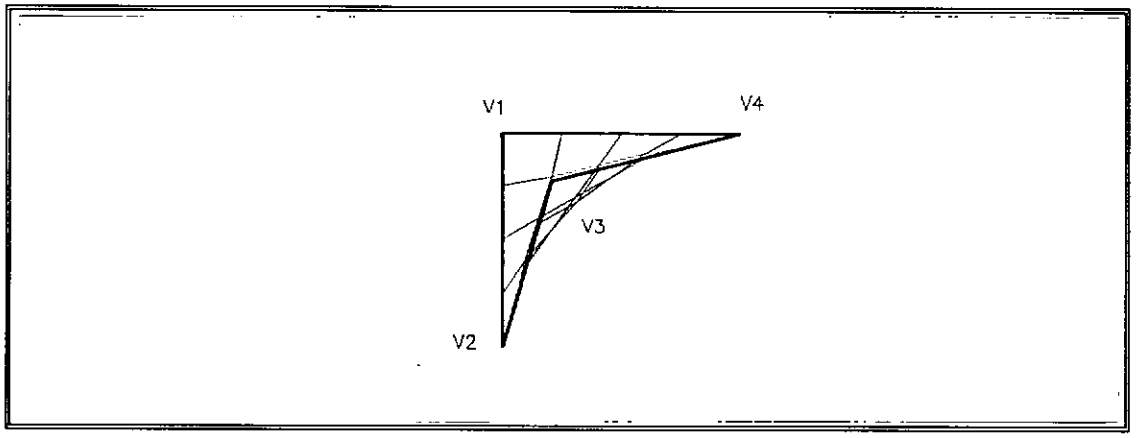


Fig V.33

Como dito anteriormente, a região a ser discretizada em elementos finitos é definida por oito pontos, quatro vértices e quatro pontos intermediários. Para o comando ISOMAPP, estes pontos possuem diretivas distintas.

Para requisitar a indicação dos vértices, o sistema apresentará a seguinte diretiva :

Nth node:

Onde **Nth** representa a numeração do ponto nodal da região, por exemplo, FIRST, SECOND e etc..

Para definição da posição do vértice, o usuário terá ao seu dispor todos os métodos padrões do **AutoCAD** (indicação da posição com o cursor, digitando as coordenadas em modo absoluto, relativo, polar ou ainda podendo utilizar os comandos de auxílio, **SNAP**, **GRID** ou **OSNAP**).

Para definição dos pontos médios, o sistema apresentará três opções ao usuário:

<Linear>/Neighbour/Point:

– A opção **LINEAR** indica ao sistema que a fronteira ou aresta correspondente ao ponto médio sendo definido, será retilínea e a distribuição dos pontos nodais uniforme. Nenhum símbolo nodal será inserido. Esta opção é *default* para a definição dos pontos médios da região.

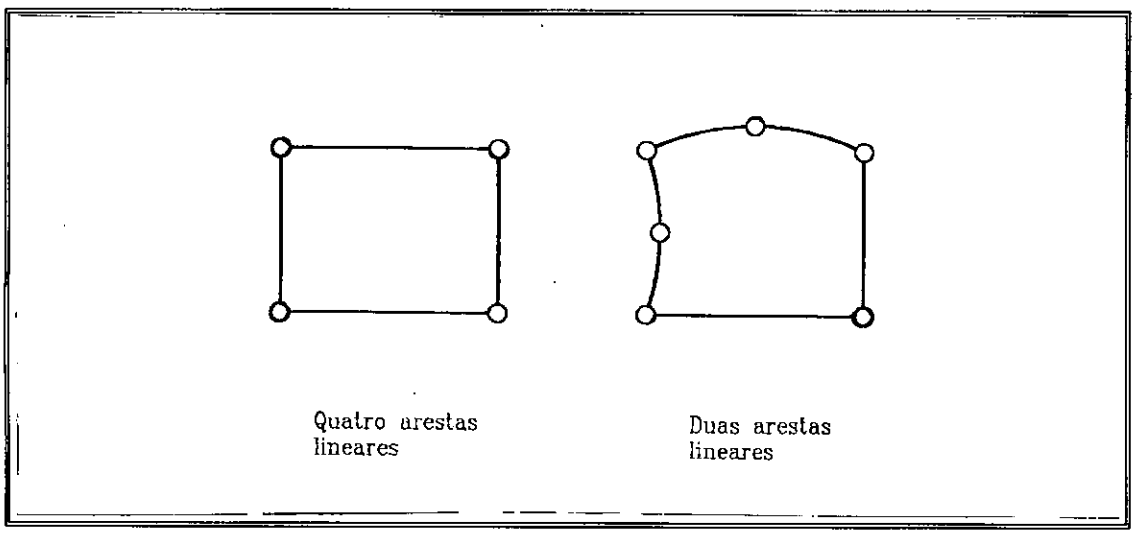


Fig V.34

– A opção **NEIGHBOUR** permite ao usuário informar ao sistema que a aresta correspondente ao ponto médio definido é vizinha a uma outra sub–regiao, ou seja, a aresta corresponde à fronteira entre duas regiões discretizadas em elementos finitos. Isto fará com que os nós dos novos elementos finitos da fronteira coincidam com os nós existentes.

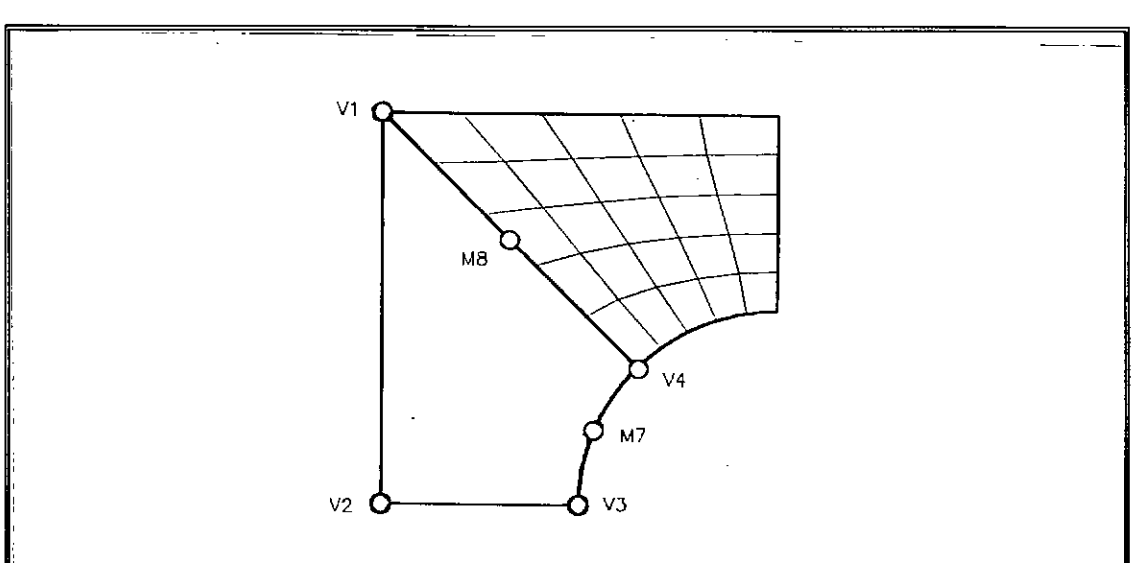


Fig V.35

A convergência dos nós é feita internamente pelo sistema a partir da opção **INS** do comando **OSNAP** do **AutoCAD**, que busca por um ponto de inserção nas proximidades de um ponto dado. Esta busca é regida pela dimensão do símbolo nodal ativa, que corresponde ao campo de busca para o **OSNAP**.

Caso nenhum objeto seja localizado no campo de busca para o ponto dado, a própria posição do ponto nodal calculada é usada.

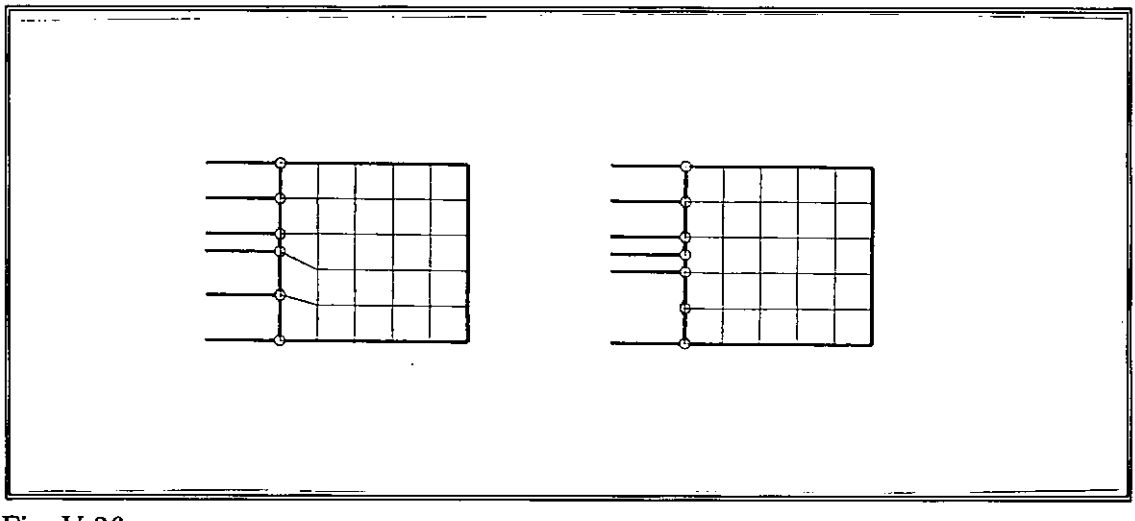


Fig V.36

O usuário deverá estar atento aos casos de geração automática de regiões vizinhas, de modo a evitar inconsistências ou descontinuidades no modelo como mostrado na figura V.37.

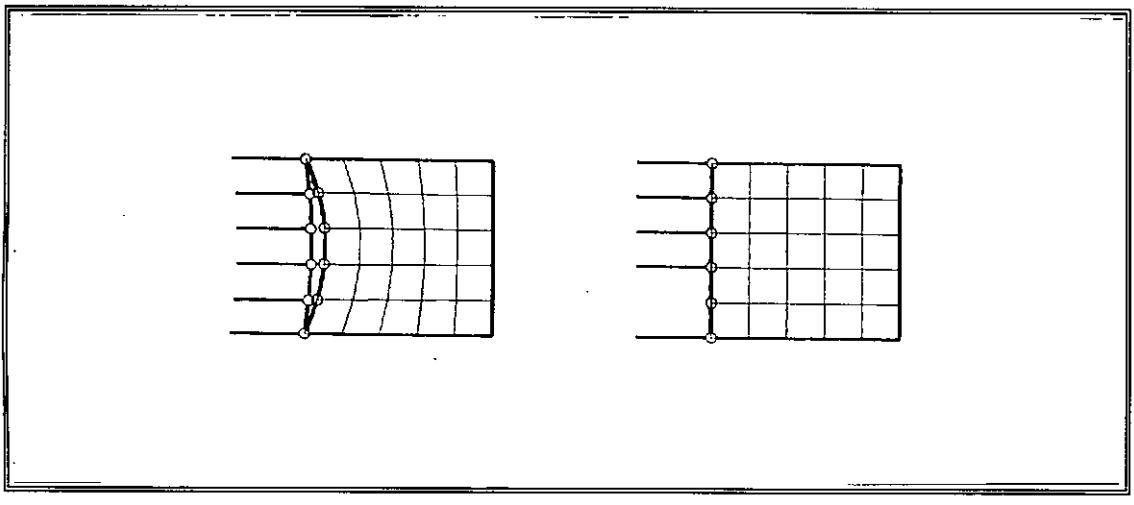


Fig V.37

Após selecionada a opção **NEIGHBOUR**, o usuário será requisitado a definir o tipo de aresta ou a posição do ponto médio através da seguinte diretiva:

```
<Linear>/Point:
```

A sub–opção **LINEAR** é semelhante a opção de mesmo nome descrita anteriormente.

A sub–opção **POINT** também é semelhante a opção de mesmo nome e será descrita a seguir.

– A opção **POINT** consiste especificamente na definição da posição do ponto médio. Esta definição poderá ser feita por qualquer dos métodos de definição de pontos do **AutoCAD**.

Não é necessário que o usuário selecione a opção **POINT** para definir a posição do ponto médio. As coordenadas via digitação ou posicionamento do cursor poderão ser fornecidas diretamente para esta diretiva. Caso selecione a opção **POINT**, o sistema instruirá o usuário a definir a posição do ponto médio da região no modelo.

A seguir apresenta–se um exemplo para a definição da posição do sexto nó de definição da região de discretização.

```
Sixth node:<Linear>/Neighbour/Point: 10,10,0
```

ou

```
Sixth node:<Linear>/Neighbour/Point: Point
Insertion Point: 10,10,0
```

Os demais dados requisitados pelo sistema são o número de colunas e linhas de elementos a serem construídos na discretização da região. O sistema apresentará como *default* os valores utilizados no último comando de geração automática.

Para finalizar, o usuário será requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda malha é automaticamente descartada do modelo. Isto inclui elementos, nós e o polígono que representa a sub–região.

A seguir apresenta–se um exemplo de sequência completa de geração automática através do comando ISOMAPP:

```
Command> Isomapp
First node: 0,0,0
Second node: 20,0,0
Third node: 20,20,0
Fourth node: 0,20,0
Fifth node: <Linear>/Neighbour/Point: L
Sixth node: <Linear>/Neighbour/Point: 20,5,0
Seventh node: <Linear>/Neighbour/Point: 10,5,0
Eighth node: <Linear>/Neighbour/Point: N
<Linear>/Point: L
Rows:<6> 4
Columns:<6> 6
Is it Ok ?<Y> Y
Command>
```

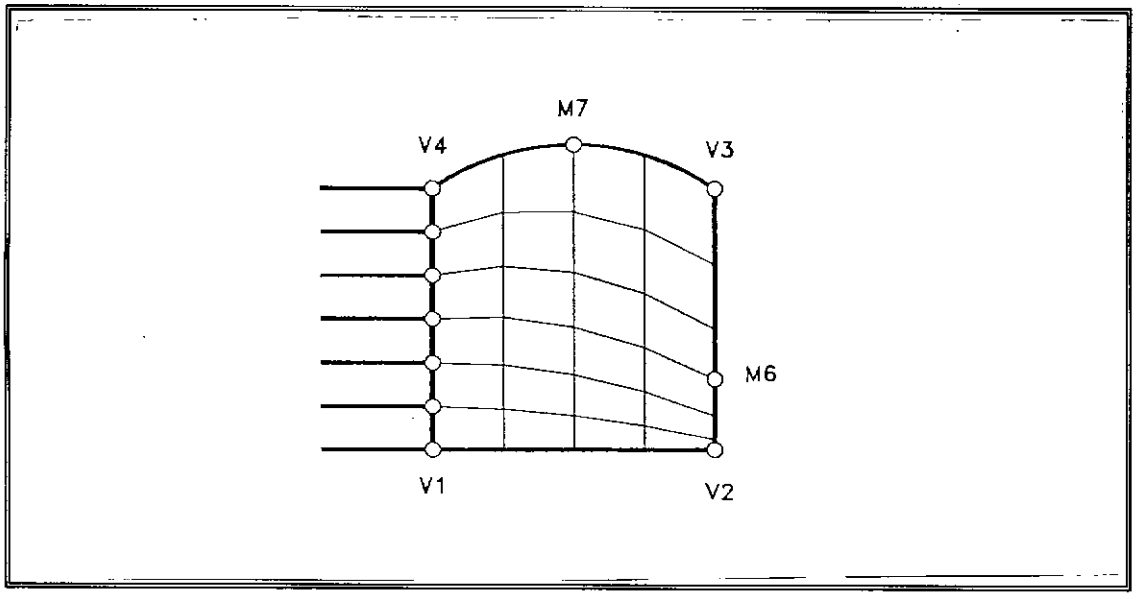


Fig V.38

O usuário deverá estar atento à definição do número de colunas e linhas. O número de linhas corresponde sempre ao número de partições da aresta que liga os vértices um e dois.

A representação da definição da região discretizada automaticamente, permanece registrada no ***layer*** SUBREGIONS. Isto permite a visualização da partição da estrutura em regiões de discretização automática. Esta partição poderá ser definida em função da geometria da estrutura ou necessidade de refinamento da malha em determinados locais.

V.4.5.1.2. O COMANDO DOUBLE

O comando **DOUBLE** é responsável pela geração automática de malhas, para regiões de transição entre regiões de diferentes graus de refinamento, usando elementos quadrangulares. Este comando utiliza a mesma técnica de mapeamento isoparamétrico do comando **ISOMAPP**. No entanto a malha de elementos finitos gerada difere por não ser regular e não ser definida por um número qualquer de linhas e colunas.

Na figura V.39 apresenta–se uma comparação do aspecto das malhas geradas pelos comandos **ISOMAPP** e **DOUBLE** em coordenadas naturais e após o mapeamento em coordenadas cartesianas.

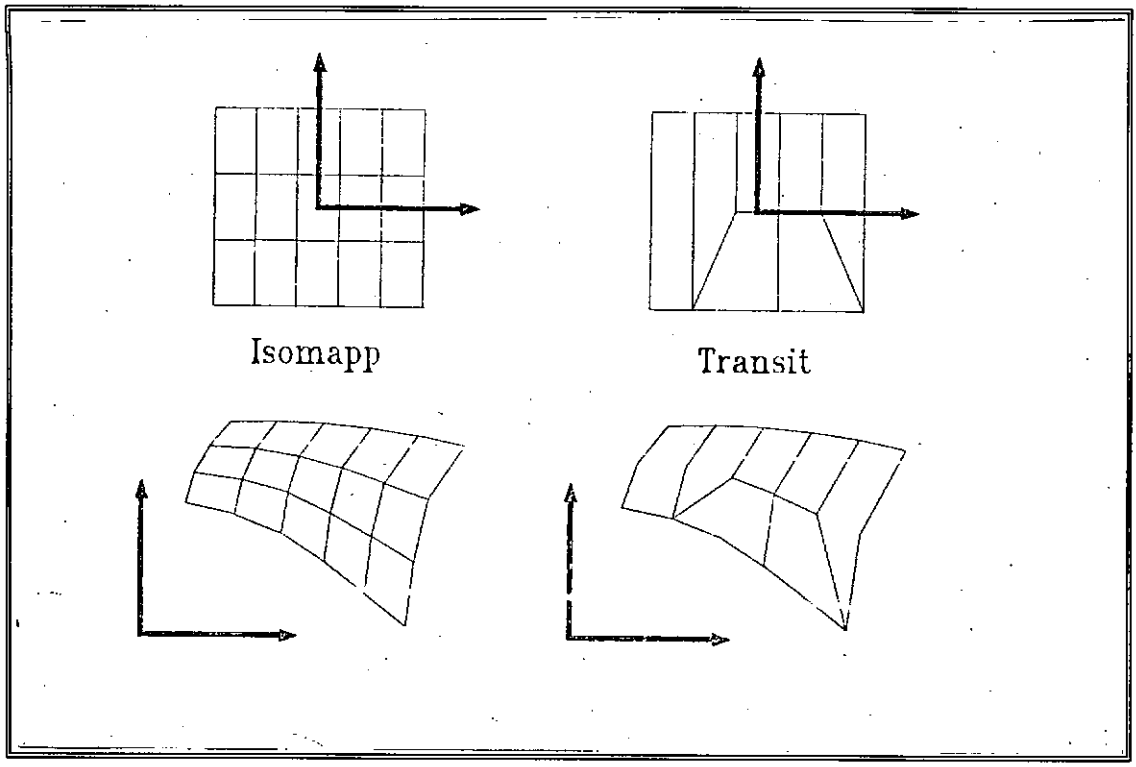


Fig V.39

Este comando duplica o número de elementos entre as arestas quatro, definida entre os vértices um e quatro, e a aresta dois, definida entre os vértices dois e três.

O número de elementos dado poderá ser o maior ou o menor, correspondendo às arestas quatro e dois respectivamente. O número de elementos não deverá ser necessariamente par, para isto está previsto um elemento inicial de fomato regular.

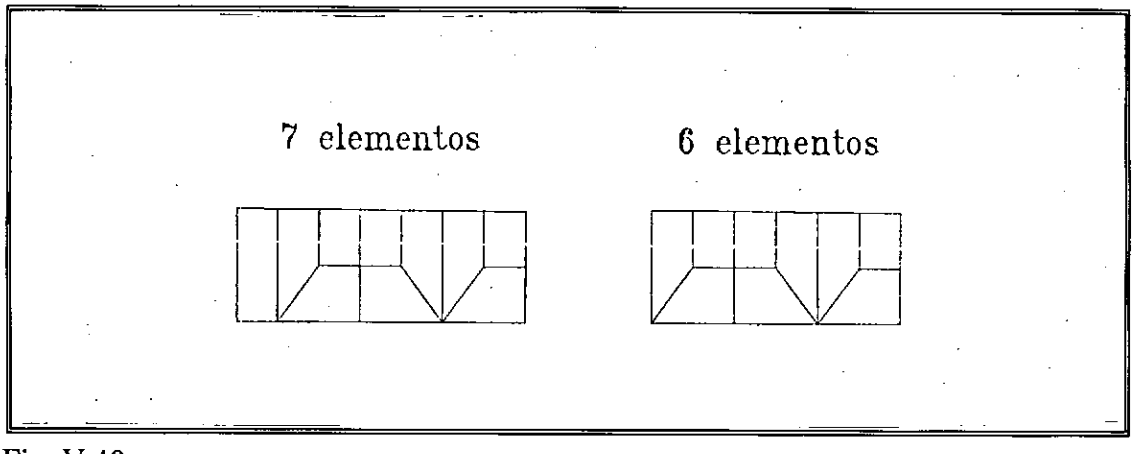


Fig V.40

O comando está disponível para elementos de quatro e oito nós apenas.

Como no comando **ISOMAPP**, a região a ser discretizada será definida por quatro vértices e até quatro pontos médios as arestas.

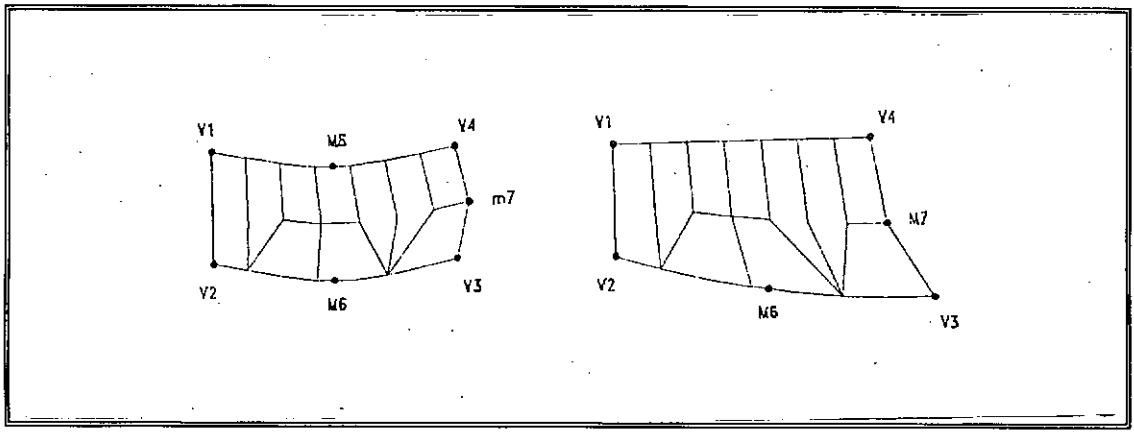


Fig V.41

A representação da definição da região a ser discretizada automaticamente, juntamente com os vértices e pontos médios das arestas, permanecem armazenados no ***layer*** SUBREGIONS para facilitar operações de verificação, ou mesmo como apoio na definição de outras regiões a serem discretizadas.

Para a definição da posição dos vértices e tipo de arestas ou posição dos pontos médios a estas, o sistema usará as mesmas diretivas e opções do comando **ISOMAPP**.

Para a definição do grau de refinamento ou o número de elementos que constituirão a malha, o sistema utiliza duas diretivas.

```
Columns:<default>
Increase/<Decrease>:
```

O sistema utiliza como *default* para o número de colunas, o valor usado no último comando de geração automática.

As opções **INCREASE** e **DECREASE** determinam se o número de elementos será duplicado ou reduzido à metade.

A figura V.42 apresenta os dois casos para o número de colunas igual a 6:

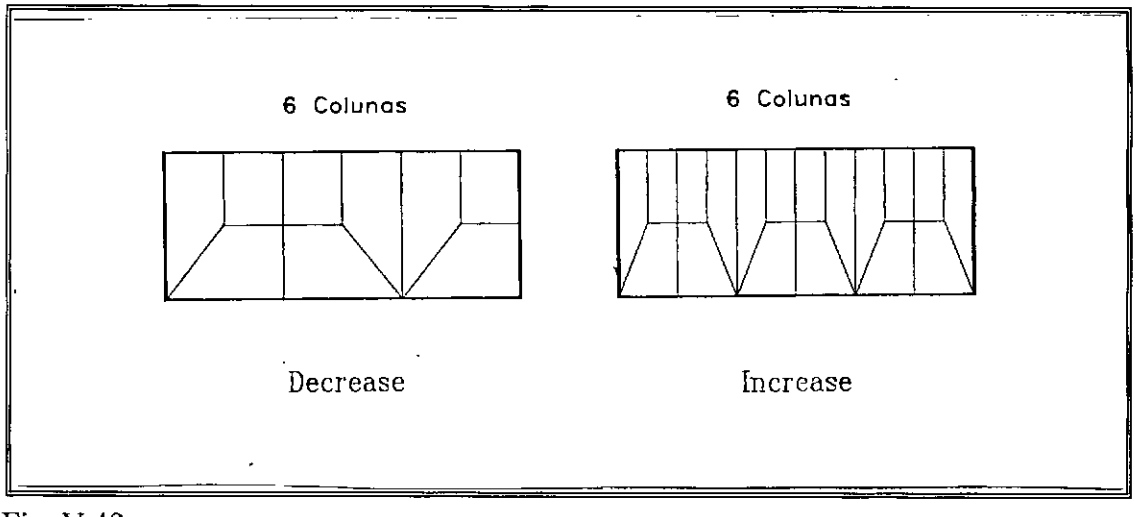
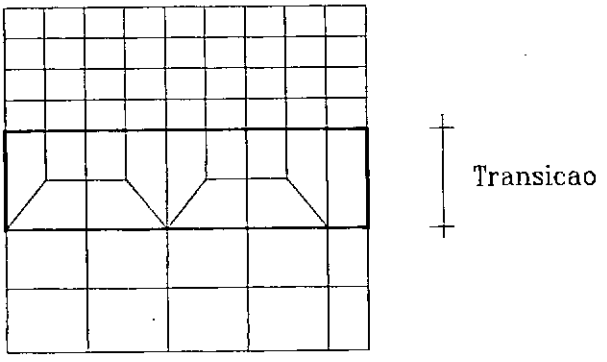


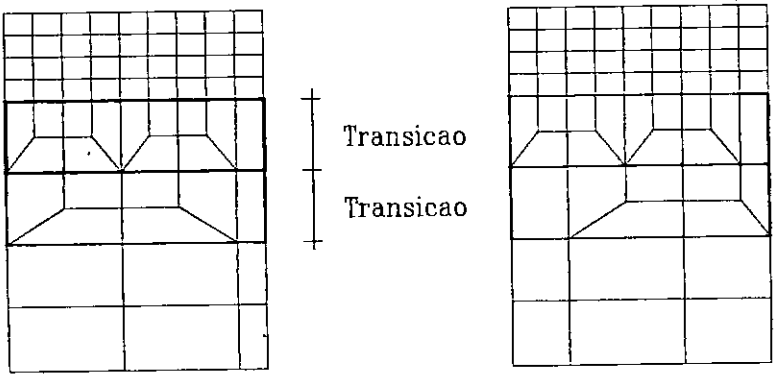
Fig V.42

Como no comando **ISOMAPP**, para finalizar o usuário será requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda a malha é automaticamente descartada do modelo. Isto inclui elementos, nós e o polígono que representa a subregião.

Este comando é especialmente recomendado para se realizar a transição entre regiões de grau de refinamento da malha diferentes, e onde se deseja utilizar apenas elementos quadriláteros.



De nove para cinco elementos



De nove para tres elementos

Fig V.43

O comando **DOUBLE** poderá ser usado repetidamente para regiões paralelas (vizinhas) de modo a obter–se a redução ou ampliação do grau de refinamento da malha.

O usuário deverá estar atento para os casos de discretizações sucessivas com o comando **DOUBLE** onde o número de colunas é ímpar. A figura V.43 apresenta um exemplo onde a utilização sucessiva do comando resultou em um elemento demasiadamente alongado no extremo da região de transição.

Este problema pode ser contornado alterando a sequência de definição dos pontos de definição da região, ou seja, definindo os vértices um e quatro ou dois e três em lados opostos para regiões subsequentes.

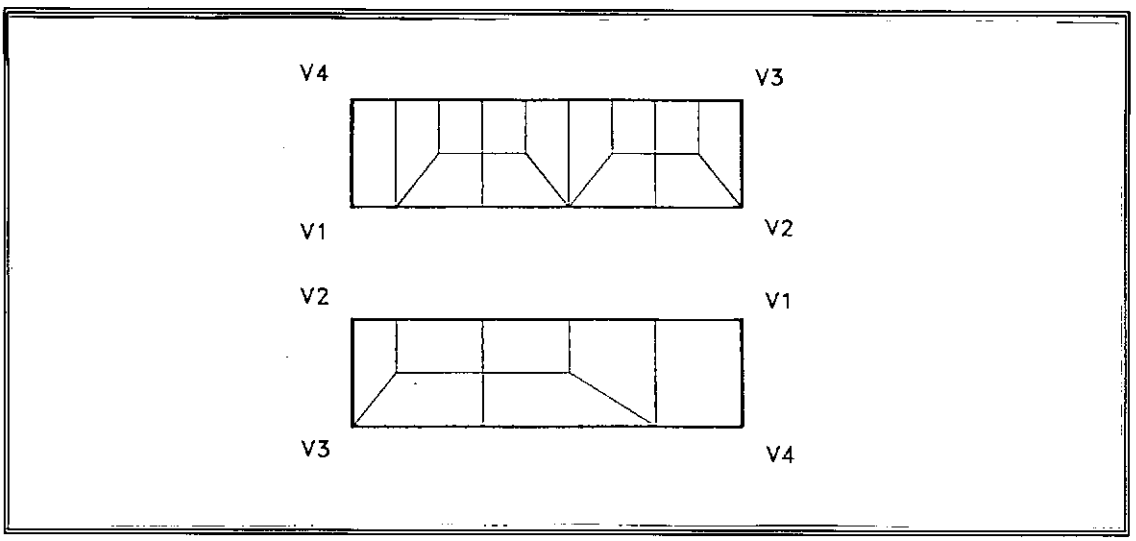


Fig V.44

Deve–se notar que o sentido na ordem de definição dos vértices foi mantido anti–horário para o exemplo, de modo a produzir elementos com vetores normais de mesmo sentido.

O procedimento apresentado produz nós mais uniformemente distribuídos nas arestas extremas das subregiões vizinhas.

V.4.5.1.3. O COMAMDO TRILINEAR

O comando **TRILINEAR** executa a discretização de uma região triangular, de arestas retas ou parabólicas, em uma malha de elementos finitos pelo método do mapeamento isoparamétrico.

A região a ser discretizada é definida por três vértices e três pontos que definem os pontos médios a estas arestas.

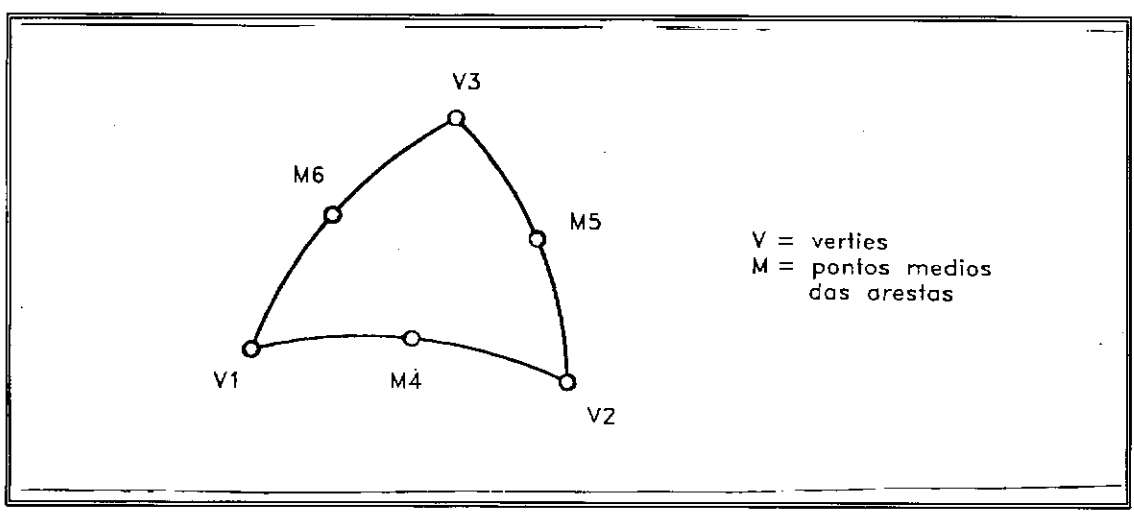


Fig V.45

Os vértices da região deverão ser obrigatoriamente definidos. Já os pontos médios podem ser omitidos, fazendo assim com que a aresta seja linear e a distribuição dos nós uniforme.

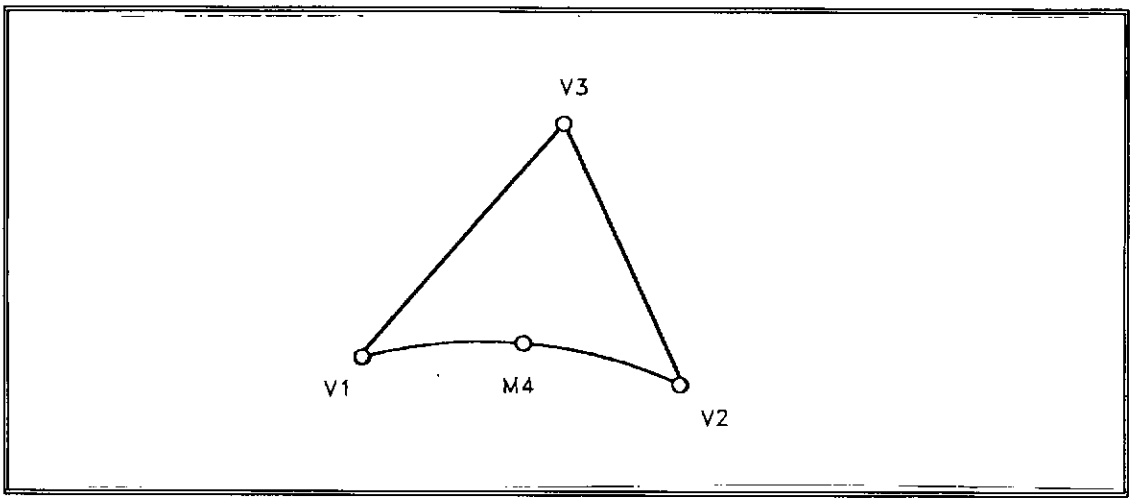


Fig V.46

Como nos demais comandos de discretização automática, o posicionamento dos pontos médios também controla a distribuição dos nós na aresta.

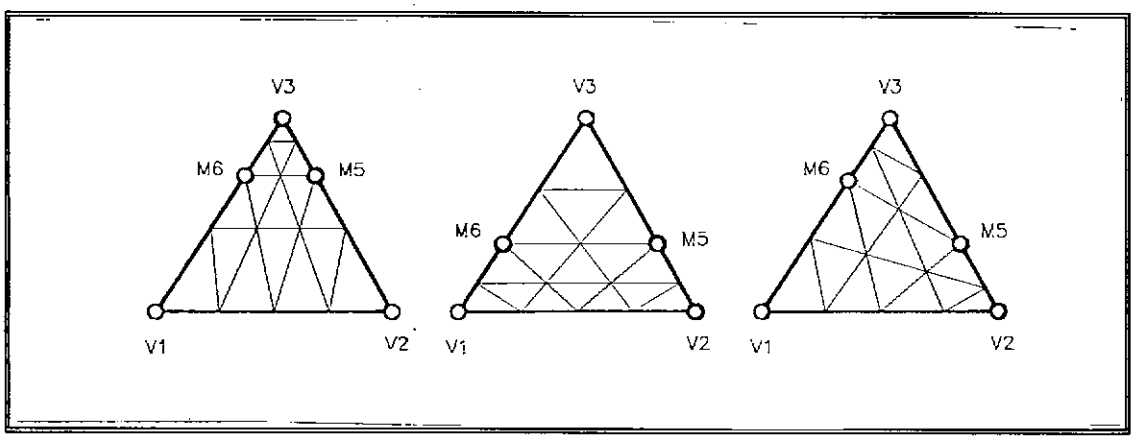


Fig V.47

O grau de refinamento da malha é definido pelo número de colunas de elementos que é igual para todas as arestas.

A versão implementada deste comando, efetua a discretização de uma região triangular com elementos triangulares de primeira ou segunda ordem, de três ou seis nós respectivamente.

Nenhum teste de seleção é realizado durante a geração para verificar qual a melhor forma para os elementos triangulares, como nos demais comandos de geração automática. A malha, portanto, terá forma regular em relação aos pontos que definem a subregião.

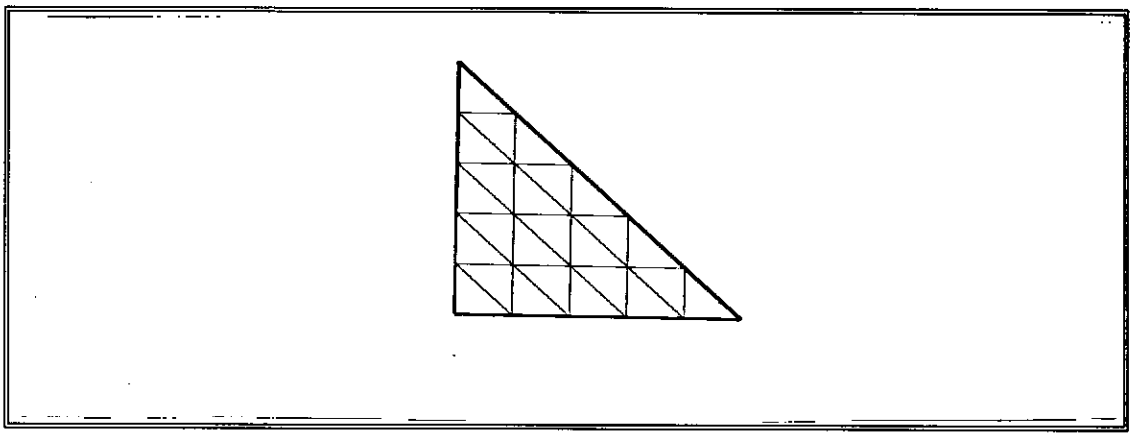


Fig V.48

O usuário deverá estar atento para o caso de regiões demasiadamente distorcidas, onde os elementos gerados poderão não ter um formato ótimo.

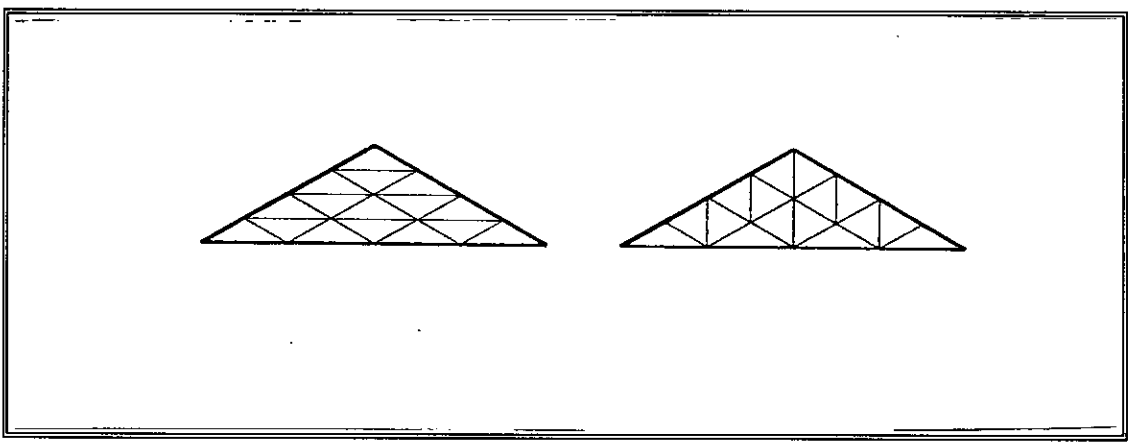


Fig V.49

Como dito anteriormente, a região a ser discretizada em elementos finitos é definida por seis pontos, três vértices e três pontos intermediários. Para o comando **TRILINEAR**, estes pontos possuem diretivas distintas.

Para requisitar a indicação dos vértices ao usuário, o sistema apresentará a seguinte diretiva:

`Nth Node:`

onde **N*th*** representa a numeração do ponto nodal da região, por exemplo, FIRST, SECOND, THIRD e etc..

Para definição da posição do vértice, o usuário terá ao seu dispor todos os modos padrão do **AutoCAD** (indicação da posição com o cursor, digitando as coordenadas em modo absoluto, relativo, polar ou ainda podendo utilizar os comandos de auxílio, **SNAP**, **GRID** ou **OSNAP**).

Para a definição dos pontos médios, o sistema apresentará três opções ao usuário :

`<Linear>/Neighbour/Point:`

Estas opções são exatamente iguais às opções apresentadas no comando **ISOMAPP**.

Todos os comentários feitos no comando **ISOMAPP** sobre a definição da região a ser discretizada permanecem válidos para o comando **TRILINEAR.**

O outro dado requisitado ao usuário pelo sistema, é o número de colunas de elementos na região de discretização. Este dado diz respeito ao grau de refinamento da malha e o valor corresponde ao número de elementos em uma das fronteiras da região. O sistema apresentará como *default* o valor utilizado para número de colunas no último comando de geração automática.

Para finalizar, o usuário será requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda a malha é automaticamente descartada do modelo. Isto inclui elementos, nós e o polígono que representa a região discretizada.

Os elementos representados por ***polylines*** são inseridos no ***layer*** referente ao tipo de material ativo. O tipo de elemento que constituirá a malha é determinado pelo tipo de elemento ativo fixado através do comando **ELEMENT.**

A figura a seguir apresenta um exemplo de sequência completa de geração automática através do comando **TRILINEAR**.

```
Command> Trilinear
First node: 0,0,0
Second node: 20,0,0
Third node: 0,20,0
Fourth node: <Linear>/Neighbour/Point: L
Fifth node: <Linear>/Neighbour/Point: 15,10,0
Sixth node: <Linear>/Neighbour/Point: N
<Linear>/Point: L
Columns:<4> 6
Is it Ok ?<Y> Y
Command>
```

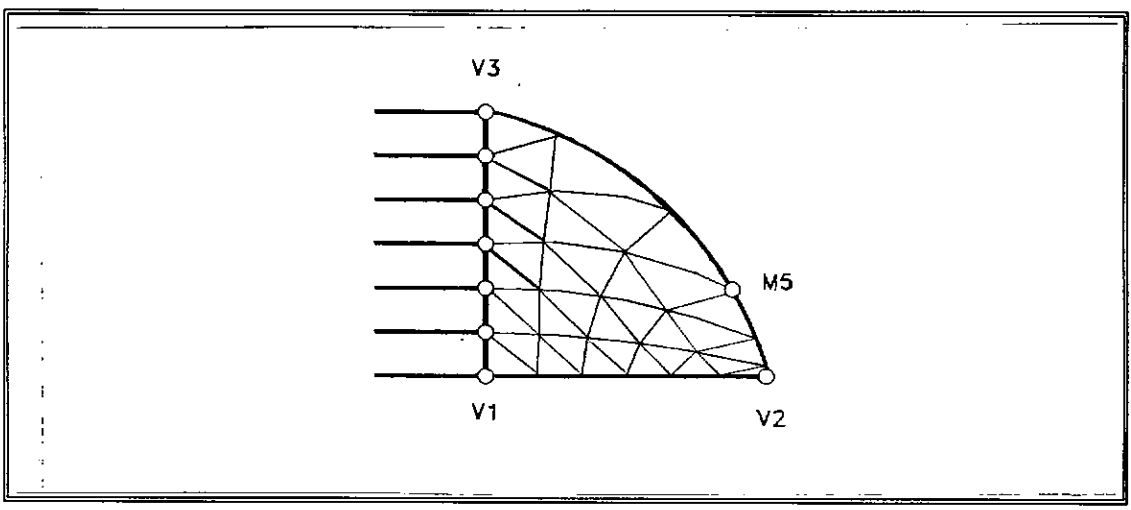


Fig V.50

V.4.5.1.4. O COMANDO TRANSIT

O comando **TRANSIT** realiza a discretização de uma região quadrilátera de arestas retas ou parabólicas, com diferente número de elementos em um malha de elementos finitos, pelo método do mapeamento isoparamétrico.

Como no comando **ISOMAPP**, a região a ser discretizada é definida por quatro vértices e quatro pontos que definem os pontos médios das arestas.

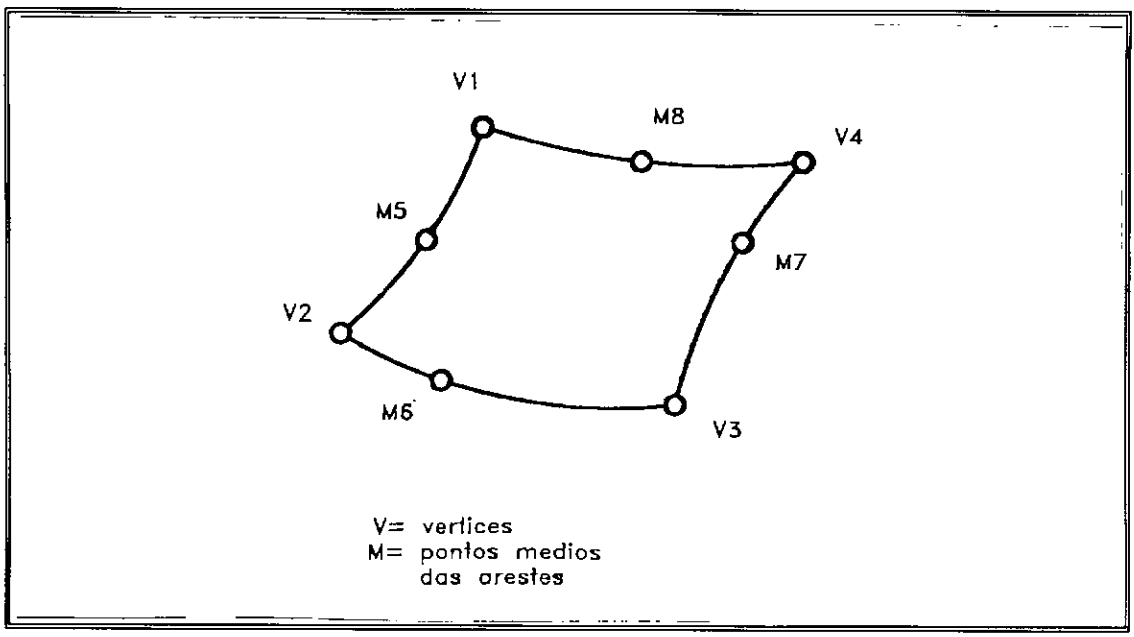


Fig V.51

Os vértices deverão ser obrigatoriamente definidos. Já os pontos médios podem ser omitidos, fazendo assim com que a aresta seja linear, e a distribuição dos nós uniforme.

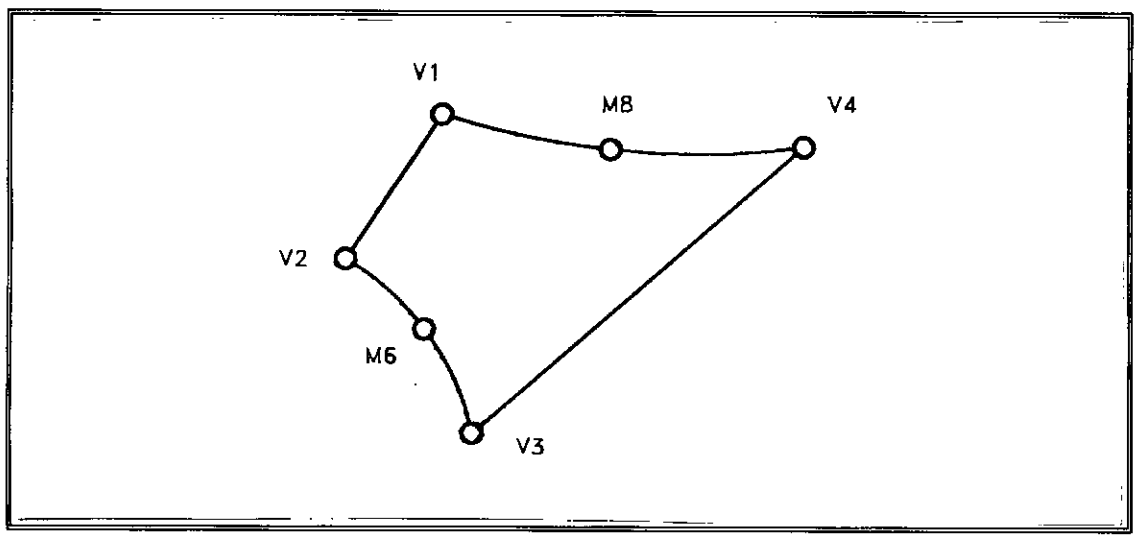


Fig V.52

Os pontos médios também controlam a distribuição dos nós na aresta.

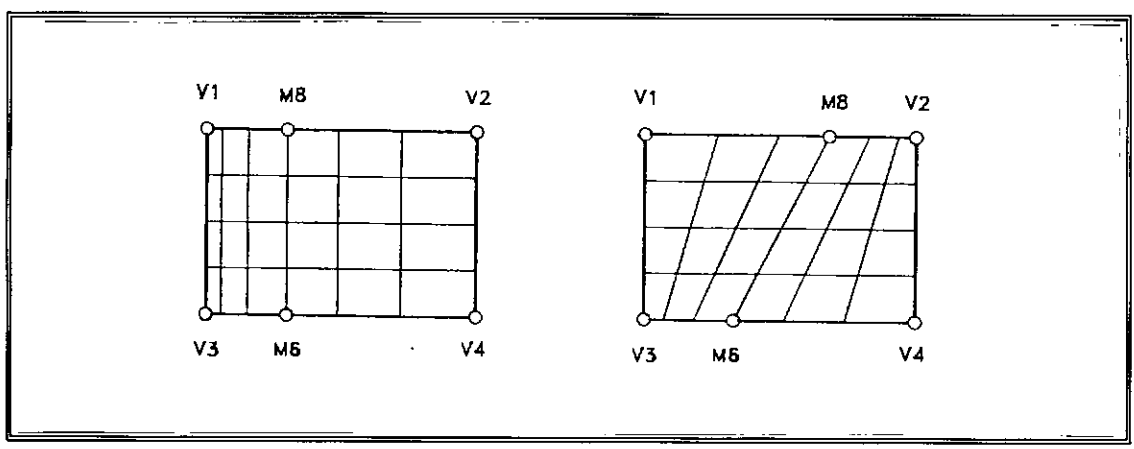


Fig V.53

O nível de refinamento da malha é determinado pelo número de elementos dado pelo usuário para cada aresta da região. As arestas laterais correspondem às arestas que ligam os vértices um e dois e vértices três e quatro. As arestas de topo e base correspondem às arestas entre os vértices um e quatro e vértices dois e três respectivamente.

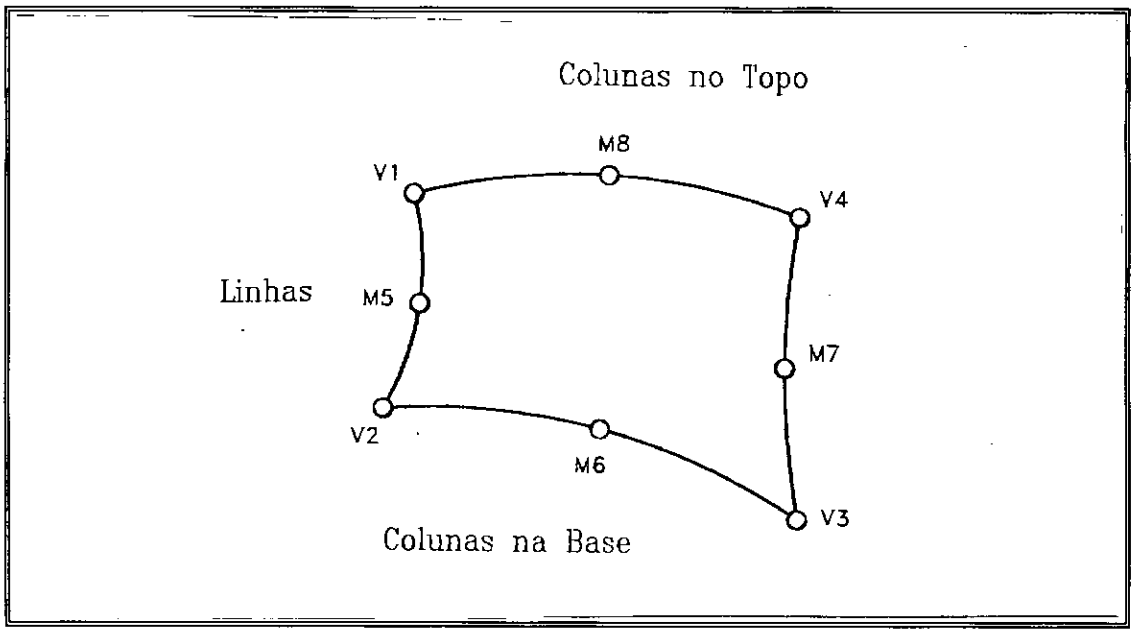


Fig V.54

O número de partições para as arestas laterais da região será igual.

O número de colunas no topo ou base da região poderá ser igual ou não, sendo que nunca a diferença entre eles deverá ser maior que o número de colunas nas laterais.

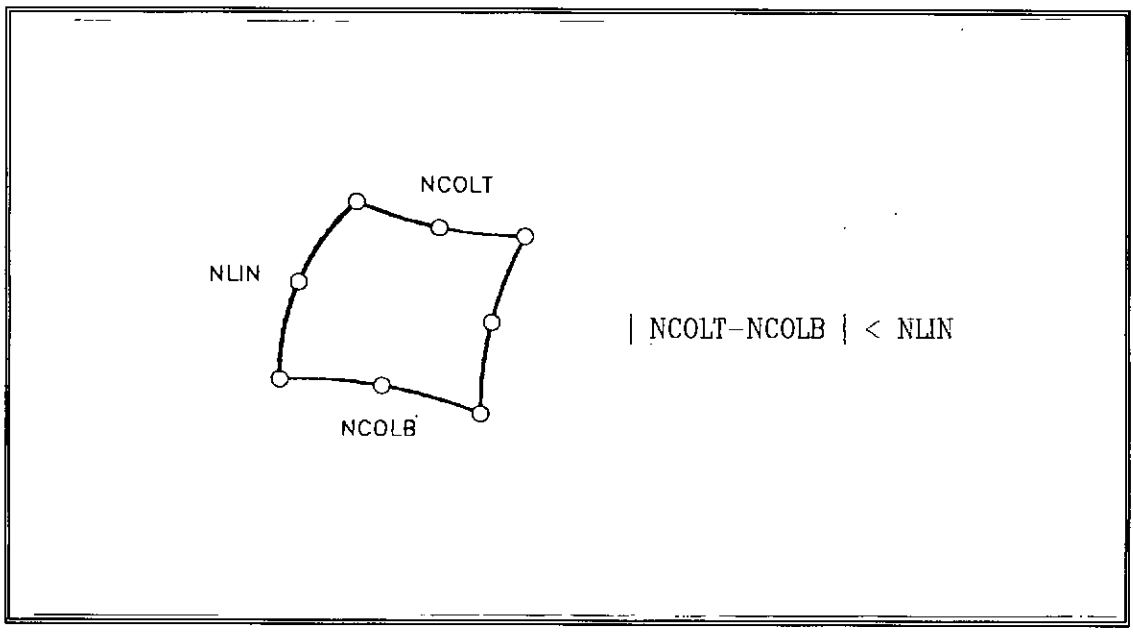


Fig V.55

Para requisitar a indicação dos vértices ao usuário, o sistema apresentará a seguinte diretiva:

Nth Node:

onde **N*th*** representa a numeração do ponto nodal da região, por exemplo, FIRST, SECOND, THIRD e etc..

Para definição da posição do vértice, o usuário terá ao seu dispor todos os modos padrões do **AutoCAD** (indicação da posição com o cursor, digitando as coordenadas em modo absoluto, relativo, polar ou ainda podendo utilizar os comandos de auxílio, **SNAP**, **GRID** ou **OSNAP**).

Para a definição dos pontos médios, o sistema apresentará três opções ao usuário :

```
<Linear>/Neighbour/Point:
```

Estas opções são exatamente iguais às opções apresentadas no comando **ISOMAPP**. Todos os comentários feitos no comando **ISOMAPP** sobre a definição da região a ser discretizada, permanecem válidos para o comando **TRANSIT**.

Os demais dados requisitados pelo sistema correspondem ao número de partições de elementos para as arestas. O sistema apresenta como *default* para o número de linhas e colunas no topo, os valores utilizados no último comando de geração automática, e para o número de colunas na base, o *default* será igual ao valor atribuído ao número de colunas no topo.

```
Rows:<default>
Columns on Top:<default>
Columns on Bottom:<default>
```

Deve–se observar que se o número de colunas na base for igual ao número de colunas no topo, chegamos a um caso particular onde os comandos **TRANSIT** e **ISOMAPP** produzem o mesmo resultado.

Para o caso geral, onde o número de partições no topo da região é diferente do número na base, a figura V.56 presenta um exemplo do aspecto da malha gerada.

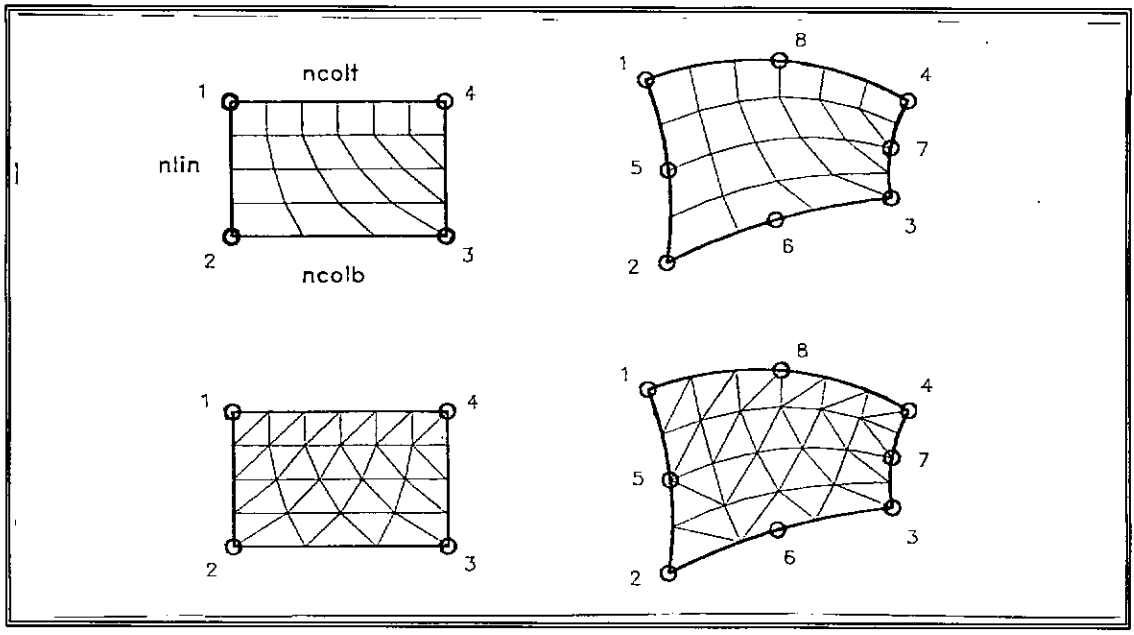


Fig V.56

O comando permite que a região seja discretizada com elementos triangulares ou quadriláteros de três e seis ou quatro e oito nós respectivamente.

Como podemos verificar na figura V.56 em ambos os casos, a malha gerada conterá elementos triangulares em uma das arestas laterais da região, isto deve–se ao algoritmo de geração adotado.

Para finalizar, o usuário será requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda a malha é automaticamente descartada do modelo. Isto inclui elementos, nós e o polígono que representa a região discretizada.

Os elementos representados por ***polylines*** são inseridos no ***layer*** referente ao tipo de material ativo. O tipo de elemento que constituirá a malha é determinado pelo tipo de elemento ativo fixado através do comando **ELEMENT**.

A figura a seguir apresenta um exemplo de sequência completa de geração automática através do comando **TRANSIT**.

```
Command> Transit
First node: 0,0,0
Second node: 20,0,0
Third node: 20,20,0
Fourth node: 0,20,0
Fifth node: <Linear>/Neighbour/Point: Linear
Sixth node: <Linear>/Neighbour/Point: 20,5,0
Seventh node: <Linear>/Neighbour/Point: 10,5,0
Eighth node: <Linear>/Neighbour/Point: N
<Linear>/Point: L
Rows:<6> 6
Columns on Top:<6> 4
Columns on Bottom:<4> 6
Is it Ok ?<Y> Y
Command>
```

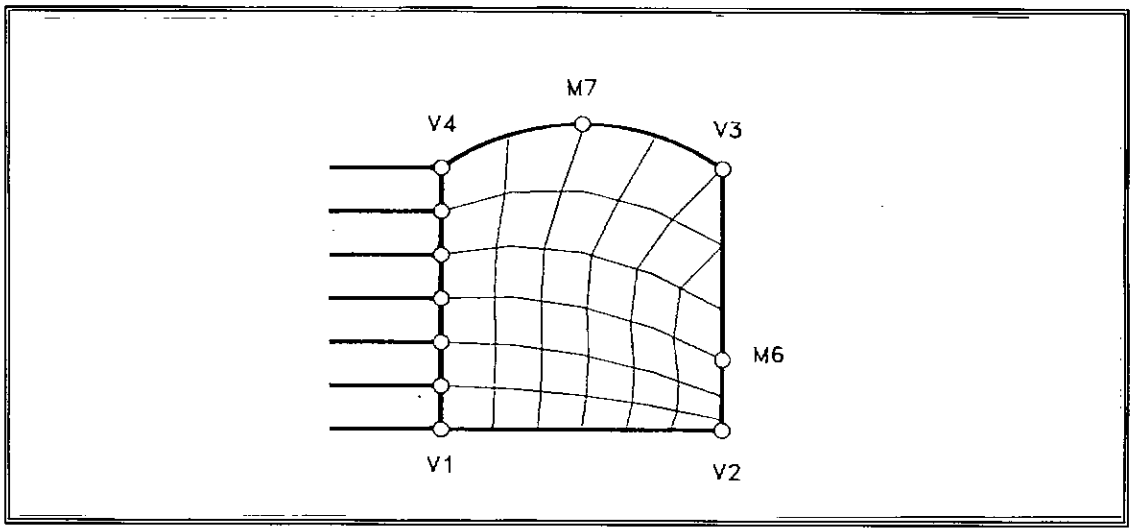


Fig V.57

O usuário deverá estar atento à definição do número de colunas e linhas, bem como à ordem de definição dos vértices. O número de linhas corresponde ao número de partições da aresta que liga os vértices um e dois e é igual ao número de partições da aresta que liga os vértices três e quatro.

V.4.5.2. GERAÇÂO AUTOMÁTICA EM REGIÕES DE FORMATO ESPECÍFICO

Este grupo de comandos é composto por programas ou rotinas de geração automática, voltados à discretização de alguns tipos de regiões de formato pré–determinado, adaptados para compor parte do sistema **AutoMESH**.

V.4.5.2.1. O COMANDO GEMAT

O comando **GEMAT** é voltado à discretização de juntas tubulares em malhas de elementos triangulares de três nós.

O programa GEMAT foi desenvolvido inicialmente por Márcio Alencar Ferreira como parte de sua tese de mestrado [191].

O comando **GEMAT** permite discretizar juntas com um número qualquer de tubos e configurações.

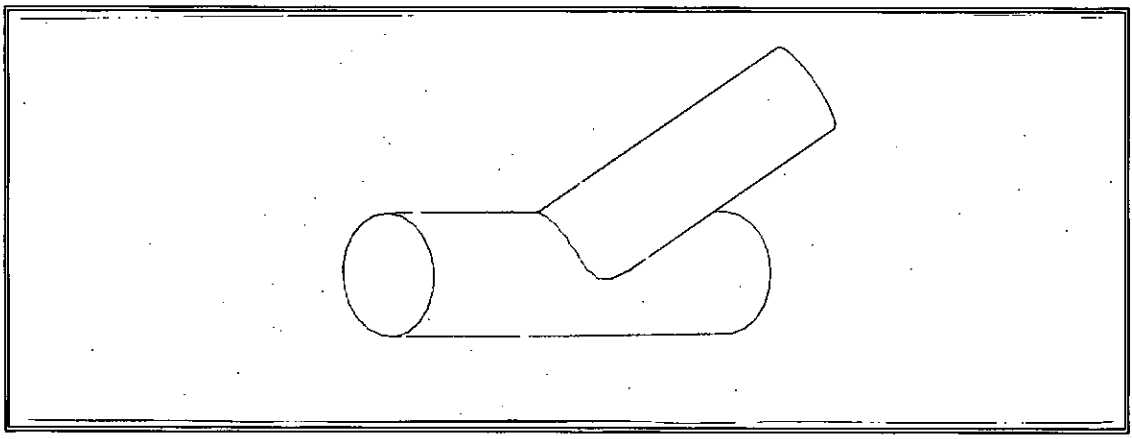


Fig V.58

Durante a sequência de operações, o usuário é requisitado a definir os elementos tubulares que fazem parte da junta, bem como o grau de refinamento desejado em cada tubo.

A definição dos elementos tubulares que compõem a junta, necessita que o usuário previamente construa sua configuração a partir das linhas de eixo dos tubos.

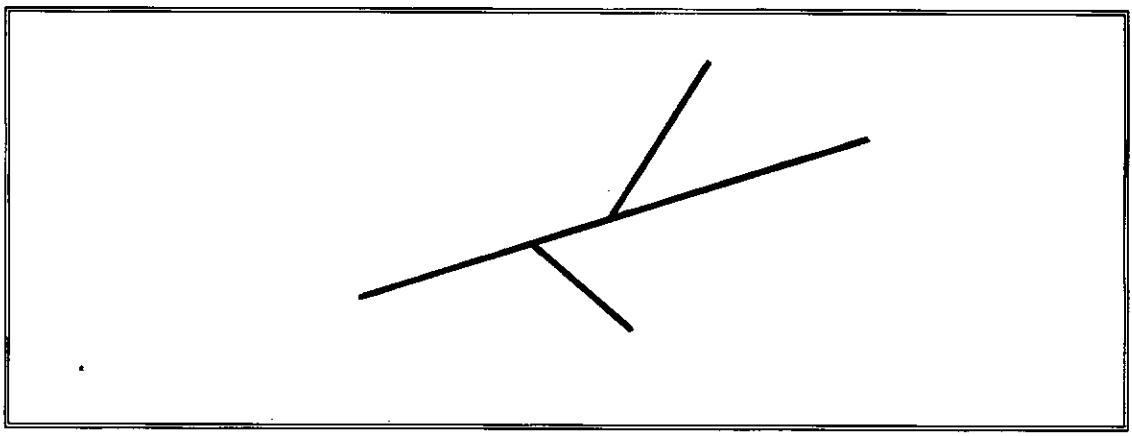


Fig V.59

O usuário é, então, requisitado a selecionar no desenho cada tubo e definir seus diâmetros.

Os tubos secundários deverão, necessariamente, ter um dos extremos de seu eixos interceptando o tubo principal. Caso contrário, serão descartados pelo sistema.

Outro dado necessário à discretização é o número de pontos nodais na interseção dos tubos. Este número determina o grau de refinamento da malha.

Apesar do programa **GEMAT** na sua concepção original apresentar opções de geração da malha com gradação controlada, optou–se por particularizar a geração da malha para uma discretização mais uniforme. Isto visa reduzir a quantidade de dados necessários à execução do comando e facilitar pela simplicidade sua utilização pelo usuário.

Para a definição dos dados do tubo principal, o sistema apresentará a seguinte sequência :

```
Select Main Tube:
Main <Diameter>/Radius:
```

Para selecionar o tubo, o usuário terá a disposição o *selection set* padrão do AutoCAD. O sistema permitirá, porém, a seleção de apenas uma entidade. Esta seleção poderá, então, ser coordenada através do cursor, opção **LAST** ou **PREVIOUS** e etc.. O sistema verifica se a entidade selecionada é realmente uma LINE ou 3DLINE. Caso contrário, uma mensagem é emitida ao usuário e uma nova seleção deverá ser feita.

A dimensão da seção transversal do tubo principal pode ser definida através do diâmetro ou raio. Em ambos os casos, estarão disponíveis para o usuário os métodos padrões do AutoCAD de definição de distâncias, ou seja, ou a indicação de um valor explícito ou de dois pontos em coordenadas absolutas ou relativas.

A opção *default* para a definição da dimensão da seção transversal do tubo principal é o diâmetro, o valor poderá ser dado diretamente neste ponto ou a opção **DIAMETER** poderá ser selecionada. O sistema requisitará, então, a definição explícita do diâmetro.

A opção **RADIUS** permite ao usuário definir a dimensão da seção transversal do tubo através do raio.

Para a definição dos dados dos tubos secundários, o sistema apresentará a seguinte sequência:

```
Select Secundary:
<Diameter>/Radius:
```

Para a seleção dos tubos secundários, não há restrições quanto ao uso do *selection set* padrão do **AutoCAD**. O usuário poderá selecionar um dos tubos secundários ou diversos com uma única operação, como no caso das opções de seleção **WINDOW** ou **CROSSING** por exemplo. O sistema então, requisitará para cada seleção realizada o diâmetro dos tubos secundários um a um. O tubo do qual está sendo requisitado o diâmetro, aparecerá na tela em coloração intensa (*highlight*).

Para cada tubo selecionado, o sistema verifica a interseção com o tubo principal. Uma mensagem é emitida ao usuário para casos de colinearidade entre os tubos e a operação é abortada.

Para a determinação das dimensões das seções transversais dos tubos secundários, as opções são as mesmas do tubo principal, porém a opção *default* corresponderá a opção selecionada na definição de diâmetro anterior.

O diâmetro para tubos secundários deverá ser sempre inferior ao diâmetro do tubo principal. O sistema emitirá uma mensagem de advertência e um novo valor será requisitado caso esta exigência não seja satisfeita.

Para finalizar, o usuário é requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda a malha é descartada automaticamente do modelo.

A seguir apresenta-se uma sequência completa de operação do comando **GEMAT**.

```
Command> Gemat
Select Main Tube:
Main <Diameter>/Radius: 40
Select Secundary:
<Diameter>/Radius: R
Radius: 30
Select Secundary: W
First point:
Second point:
<Diameter>/Radius: 15
<Diameter>/Radius: 10
Select Secundary: enter
Is it Ok ?<Y> Y
Command>
```

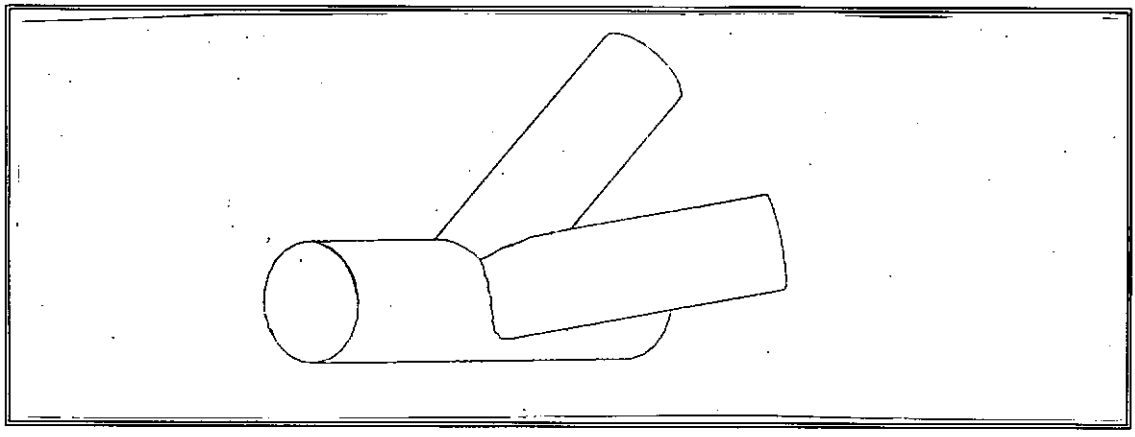


Fig V.60

O usuário deverá estar atento à definição das posições dos tubos secundários e suas dimensões pois, nenhum teste é realizado quanto à intersecção entre estes.

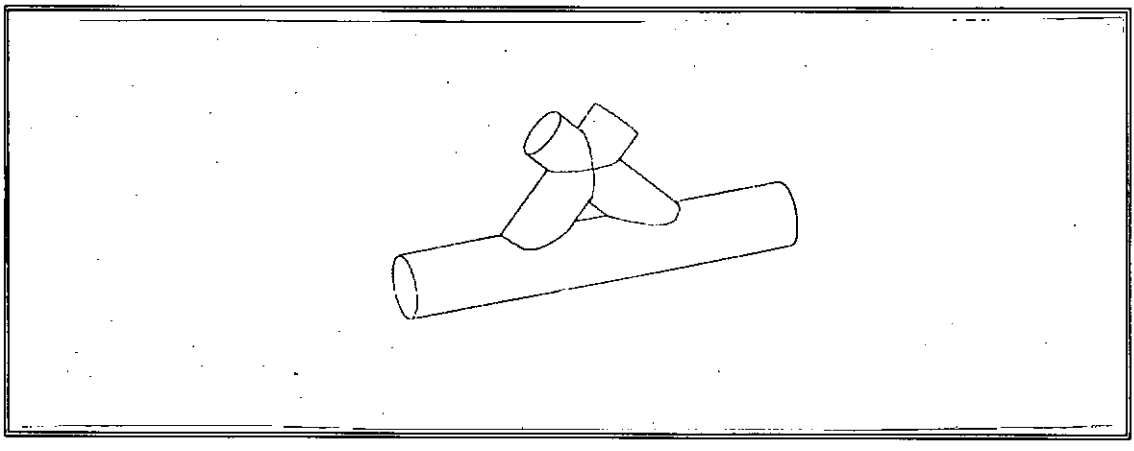


Fig V.61

V.4.5.2.2. O COMANDO SPHERE

O comando **SPHERE**, como o nome sugere, permite ao usuário gerar a malha de elementos finitos para uma esfera.

A técnica utilizada para a discretização da região, baseia–se na geração dos elementos para um dos gomos da esfera e consecutivas cópias com rotação deste gomo.

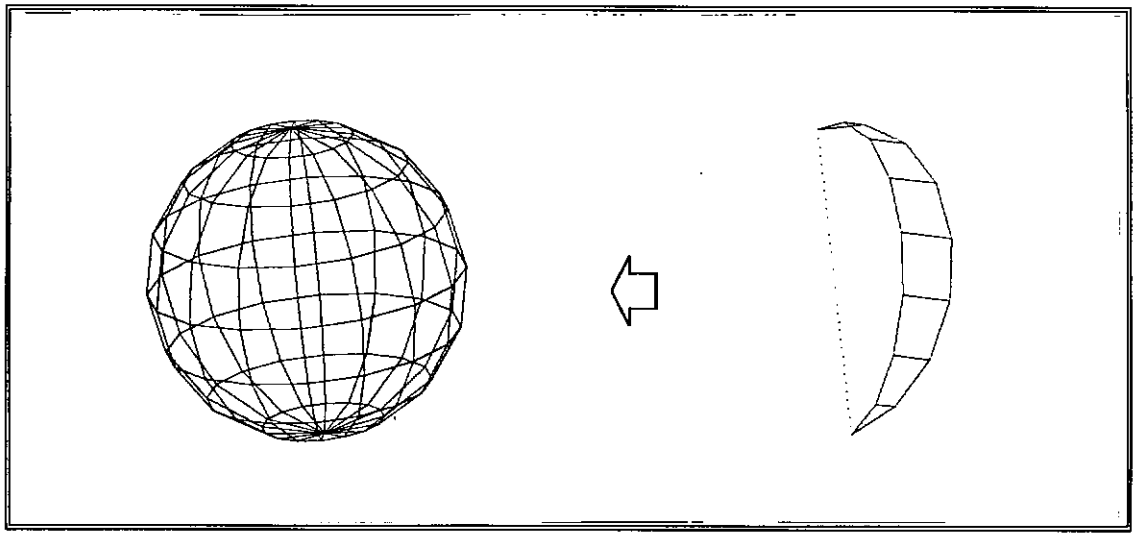


Fig V.62

O comando permite a discretização da região com elementos reticulados, triangulares, quadriláteros de primeira ou segunda ordem.

O elemento adotado na discretização corresponde ao elemento ativo, definido através do comando **ELEMENT** opção **SET**.
Os elementos representados por ***polylines*** são inseridos no ***layer*** referente ao tipo do material ativo, definido através do comando **MATERIAL** opção **SET**.

Para definição da região a ser discretizada, o usuário será requisitado a indicar o centro e as dimensões da esfera.

Para definição do centro, o usuário poderá utilizar qualquer dos modos padrões de indicação de pontos do **AutoCAD**, tais como coordenadas absolutas, relativas, através do cursor e ainda utilizar comandos auxiliares como **OSNAP**, **GRID** ou **SNAP**.

Para definir as dimensões da esfera, o usuário será requisitado a definir o diâmetro ou raio, o qual poderá ser definido por fornecimento direto do valor, através de dois pontos ou outro modo qualquer de definição de distâncias do **AutoCAD**.

Para definir o grau de refinamento da malha, o usuário será requisitado a definir o número de colunas e linhas da região que correspondem ao número de segmentos longitudinais e latitudinais respectivamente.

A seguir apresenta–se uma sequência de operações para o comando **SPHERE**.

```
Center of Sphere:
Diameter/<Radius>:
Columns:<default>
Rows:<default>
Is it Ok?<Y>
Command>
```

A malha gerada para a esfera tem eixo de geração passando pelo centro e com direção do eixo Z do sistema de coordenedas ativo. Para posicionar o eixo de geração da esfera em uma direção qualquer, o usuário poderá editar a malha gerada realizando rotações para a direção que deseja ou anteriormente a execução do comando **SPHERE**, posicionar um sistema de coordenadas locais com o eixo Z na direção em que deseja o eixo de geração da malha.

Os elementos da linha de topo da esfera serão sempre do tipo triangular, mesmo que o elemento ativo no momento da execução do comando seja do tipo quadrilátero. De modo a manter a continuidade, os elementos do topo terão sempre a mesma ordem dos demais elementos (3 ou 6 nós).

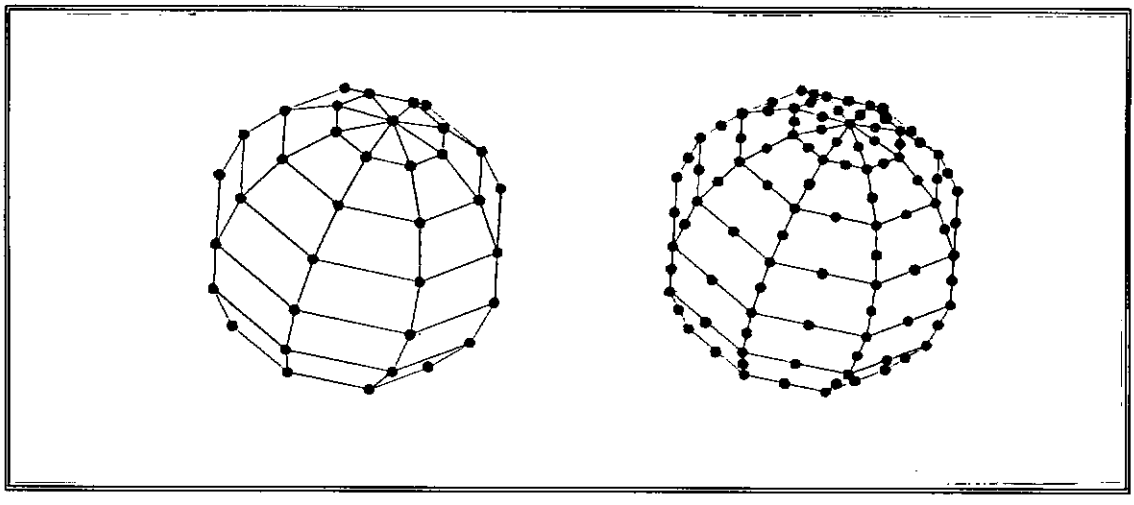


Fig V.63

Para o caso de malhas geradas com elementos triangulares, cada elemento quadrilátero dá origem a dois elementos triangulares.

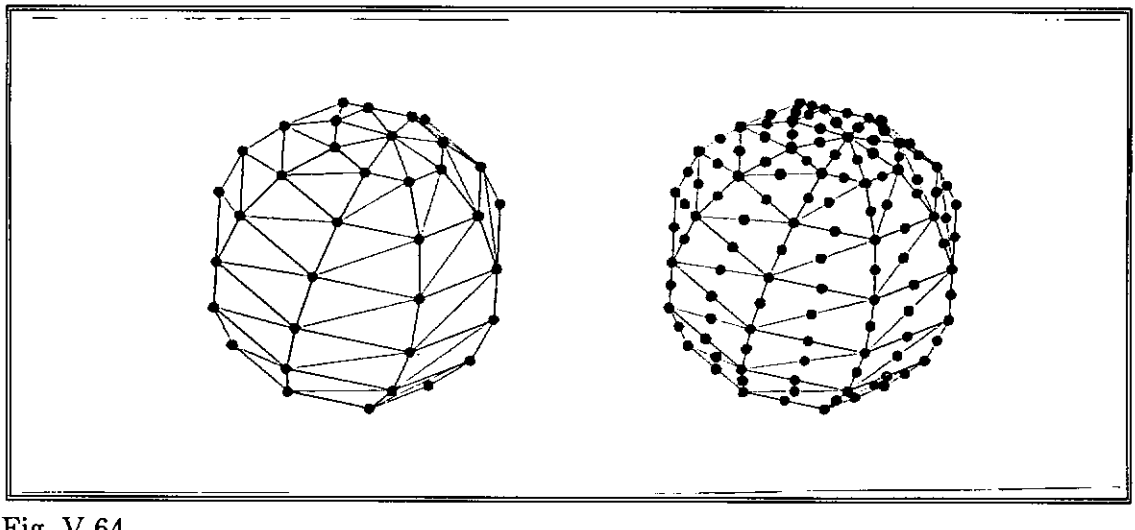


Fig V.64

Para finalizar, o usuário é requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda a malha é descartada automaticamente do modelo.

V.4.5.2.3. O COMANDO DOMES

O comando **DOMES** é semelhante ao comando **SPHERE** anteriormente apresentado. Neste comando, apenas um hemisfério da esfera é discretizado em elementos finitos.

A técnica utilizada para a discretização da região, baseia–se na geração dos elementos para um dos gomos do hemisfério e consecutivas cópias com rotação dos gomos.

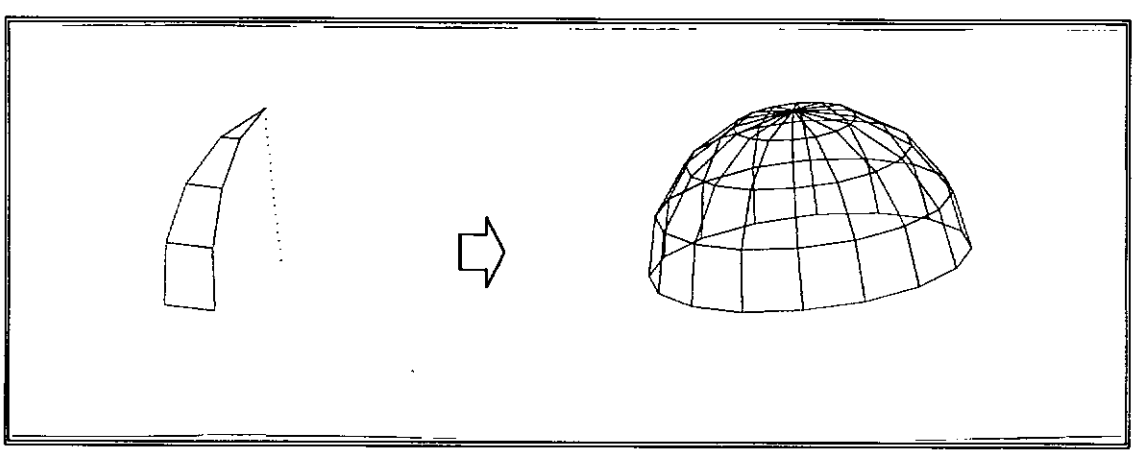


Fig V.65

Tudo o que foi descrito para o comando **SPHERE**, suas opções e entrada de dados aplica–se ao comando **DOMES**.

A seguir apresenta–se uma sequência de operações para o comando **DOMES**.

```
Command> Domes
Center of Domes:
Diameter/<Radius>:
Columns:<default>
Rows:<default>
Is it Ok ?<Y>
Command>
```

Os valores apresentados como *default* para o número de linhas e colunas são os utilizados no último comando de geração automática.

Deve–se observar que o número de linhas no hemisfério corresponde à metade do número de linhas na esfera. Isto significa que para o mesmo número de linhas e colunas obtém–se uma malha mais refinada no hemisfério.

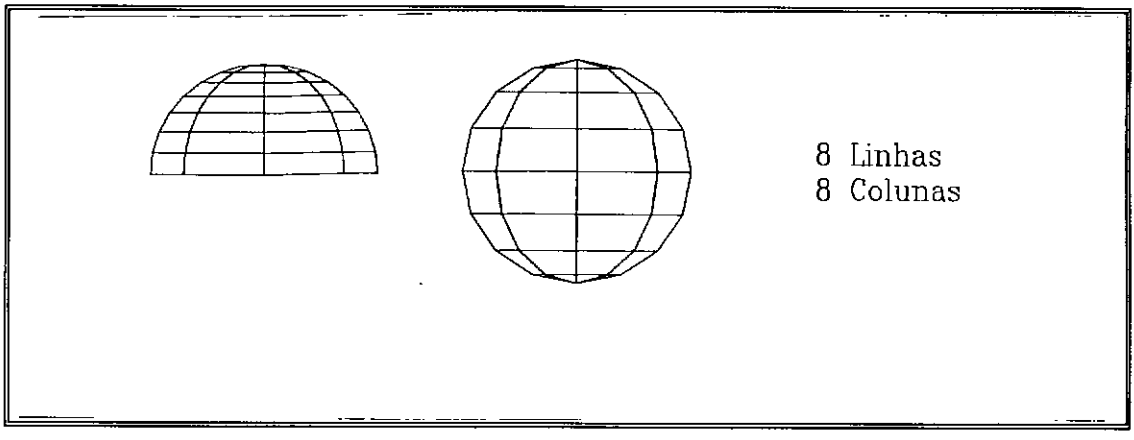


Fig V.66

Como no comando **SPHERE**, os elementos do topo serão sempre do tipo triangular, mesmo que o elemento ativo no momento da execução do comando seja do tipo triangular. De modo a manter a continuidade, os elementos do topo terão sempre a mesma ordem dos demais elementos (3 ou 6 nós).

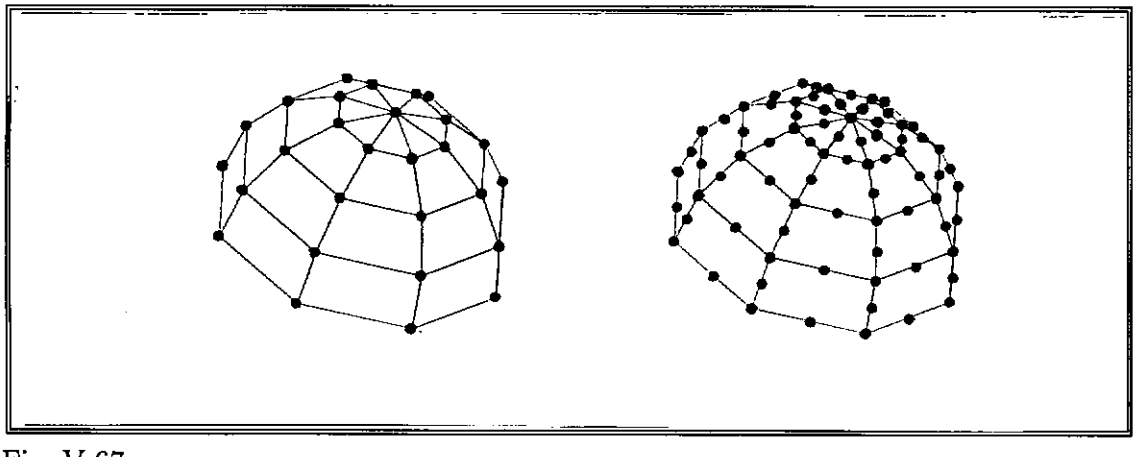


Fig V.67

Para o caso de malhas geradas com elementos triangulares, cada elemento quadrilátero dá origem a dois elementos triangulares.

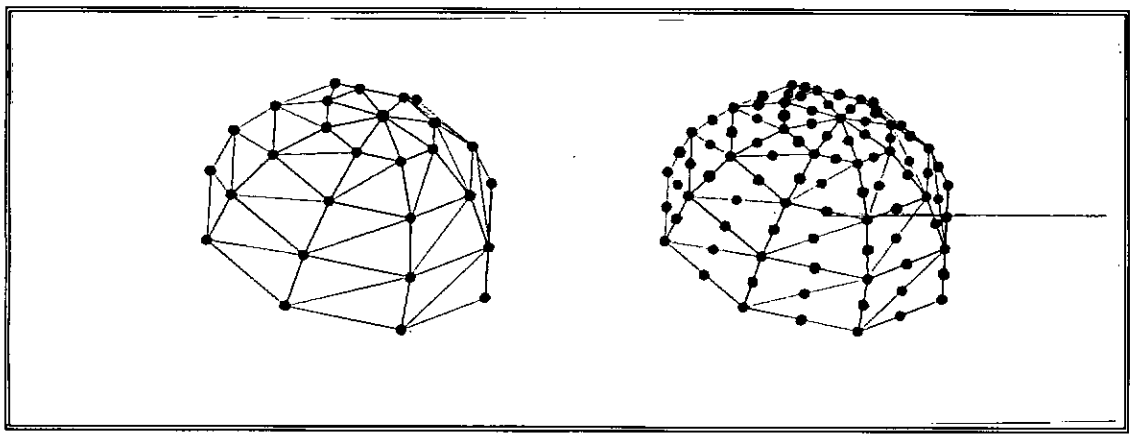


Fig V.68

A malha gerada para o hemisfério, tem eixo de geração passando pelo centro dado e com direção do eixo Z do sistema de coordenadas ativo. Para obter o eixo de geração em outra direção, o usuário poderá efetuar rotações no modelo através dos comandos de edição ou posicionar um sistema de coordenadas local com o eixo Z na direção desejada para o eixo de geração do hemisfério antes da execução do comando.

Após executada a discretização, o usuário tem a opção de rejeitar o resultado obtido, o que faz com que a malha gerada seja descartada.

V.4.5.2.4. O COMANDO TUBE

O comando **TUBE** permite ao usuário discretizar, em elementos finitos, a superfície de um tubo cônico.

A técnica utilizada para a discretização do tubo baseia–se na geração dos elementos para um segmento longitudinal e consecutivas cópias com rotação.

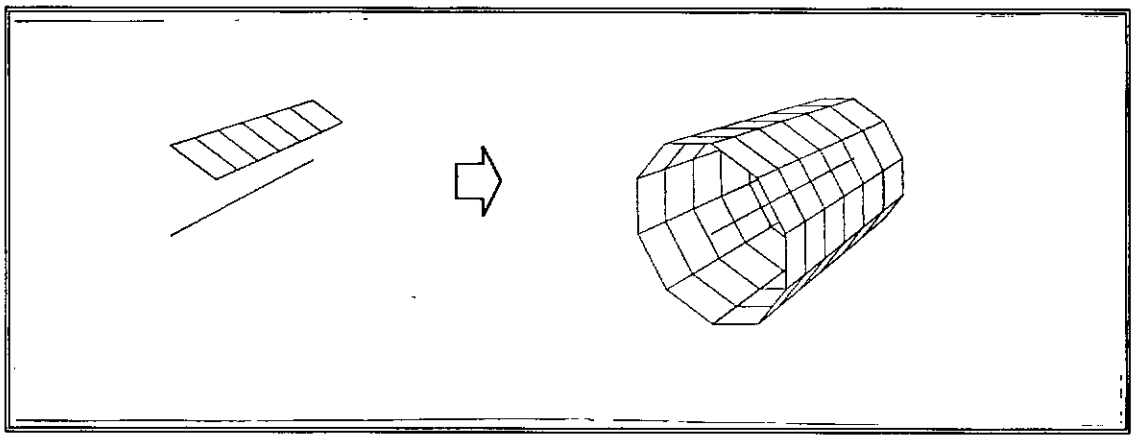


Fig V.69

O comando permite a discretização da região com elementos reticulados, triangulares, quadriláteros de primeira ou segunda ordem.

O elemento adotado na discretização corresponde ao elemento ativo, definido através do comando **ELEMENT** opção **SET**.

Os elementos representados por ***polylines*** são inseridos no ***layer*** referente ao tipo do material ativo, definido através do comando **MATERIAL** opção **SET**.

A região a ser discretizada é definida pelos pontos extremos do eixo de rotação e os respectivos diâmetros.

Para a definição dos pontos extremos do eixo de rotação do tubo, o usuário poderá utilizar qualquer dos modos padrões de definição de pontos do **AutoCAD**.

Para a definição dos diâmetros, permanecem disponíveis também os métodos padrões de definição de distâncias do **AutoCAD**.

Para definir o grau de refinamento da malha, o usuário será requisitado a definir o número de linhas e colunas de elementos da região, que correspondem ao número de segmentos transversais e longitudinais respectivamente.

A seguir apresenta–se uma sequência de operações para o comando **TUBE**.

```
Command> Tube
Base center point:
Diameter/<Radius> of base:
Top center point:
Diameter/<Radius> of top:<default>
Columns:<default>
Rows:<default>
Is it Ok ?<Y>
Command>
```

Para diâmetro ou raio do topo do tubo, o sistema apresenta como opção e valor *default*, os valores utilizados para diâmetro ou raio da base. O valor fornecido para base nunca deverá ser nulo, porém, para o topo isto é permitido. Neste caso, os elementos da última linha de elementos serão triangulares necessariamente, mesmo sendo quadrilátero o tipo de elemento ativo.

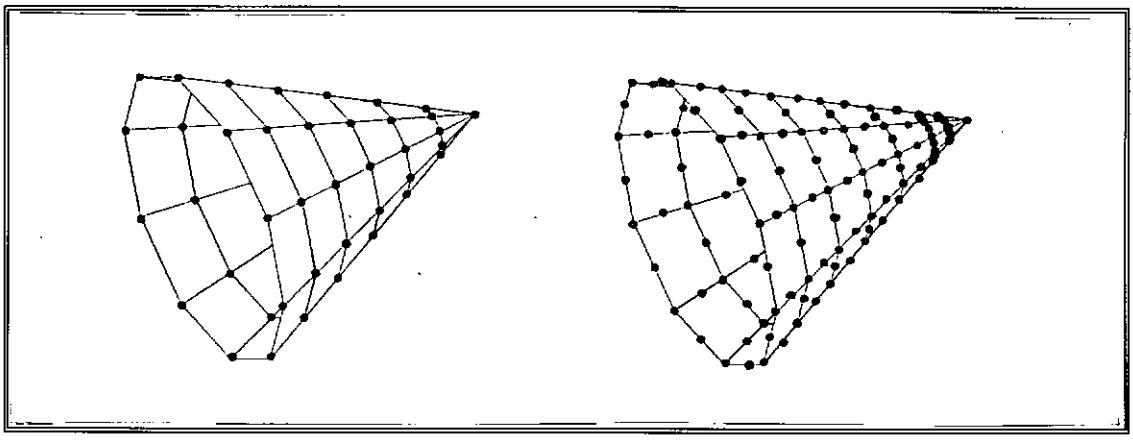


Fig V.70

Para o caso de malhas geradas com elementos triangulares, cada elemento quadrilátero dá origem a dois elementos triangulares.

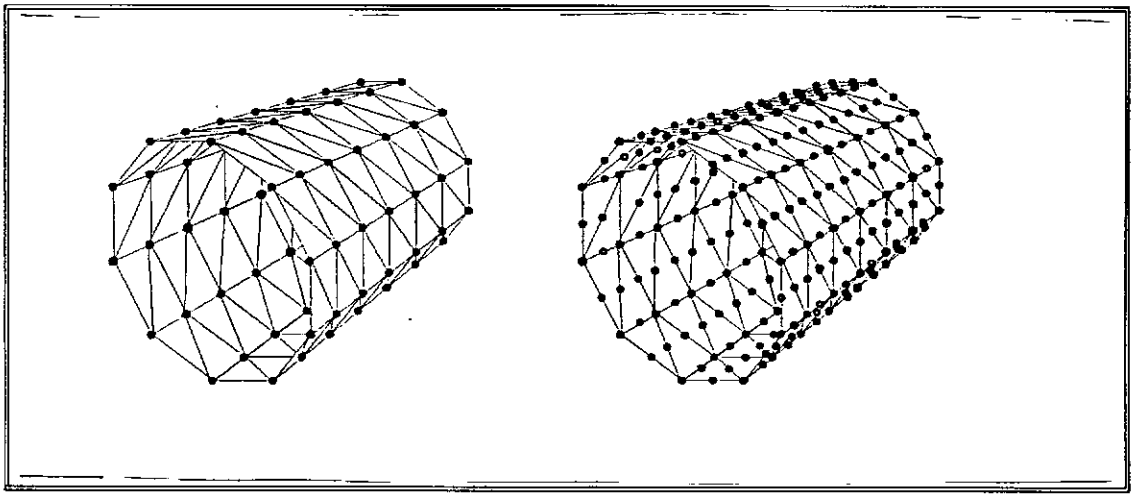


Fig V.71

Após executada a discretização, o usuário tem a opção de rejeitar o resultado obtido e toda a malha gerada é automaticamente descartada.

V.4.5.2.5. O COMANDO TORUS

O comando **TORUS** permite ao usuário gerar a malha de uma região em forma de toróide.

A técnica utilizada para a discretização da região baseia–se na geração dos elementos de um segmento transversal do toróide e consecutivas cópias com rotação do segmento.

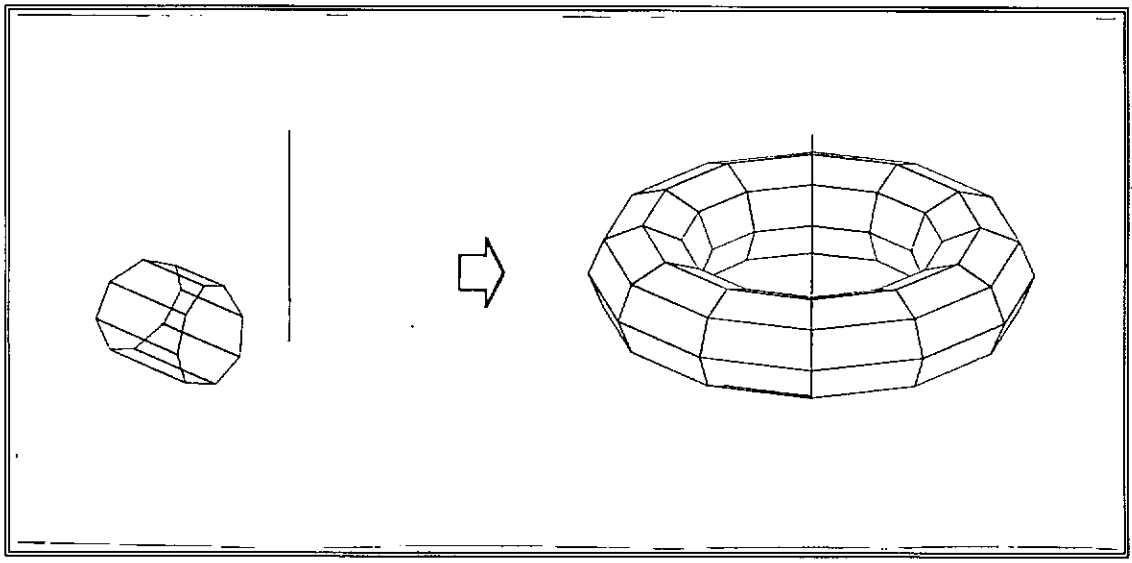


Fig V.72

O comando permite a discretização da região em elementos reticulados, triangulares ou quadriláteros de primeira ou segunda ordem.

O elemento adotado na discretização corresponde ao elemento ativo, definido através do comando **ELEMENT** opção **SET**. Os elementos representados por ***polylines*** são inseridos no ***layer*** referente ao tipo do material ativo, definido através do comando **MATERIAL** opção **SET**.

Para a definição das dimensões do toróide, o usuário será requisitado a definir o centro e o raio do toróide, bem como o raio do tubo que o forma.

Para definir o grau de refinamento da malha, o usuário será requisitado a definir o número de linhas e colunas de elementos da região, que correspondem ao número de segmentos transversais e longitudinais respectivamente.

A seguir apresenta–se uma sequência de operação do comando TORUS.

```
Command> Torus
Center point:
Diameter/<Radius> of torus:
Diameter/<Radius> of tube:
Columns:<default>
Rows:<default>
Is it Ok ?<Y>
Command>
```

Os valores *default*, apresentados pelo sistema para o número de linhas e colunas, correspondem aos valores adotados no último comando de geração automática.

Para o caso de malhas geradas com elementos do tipo triangular, cada elemento quadrilátero dá origem a dois elementos triangulares.

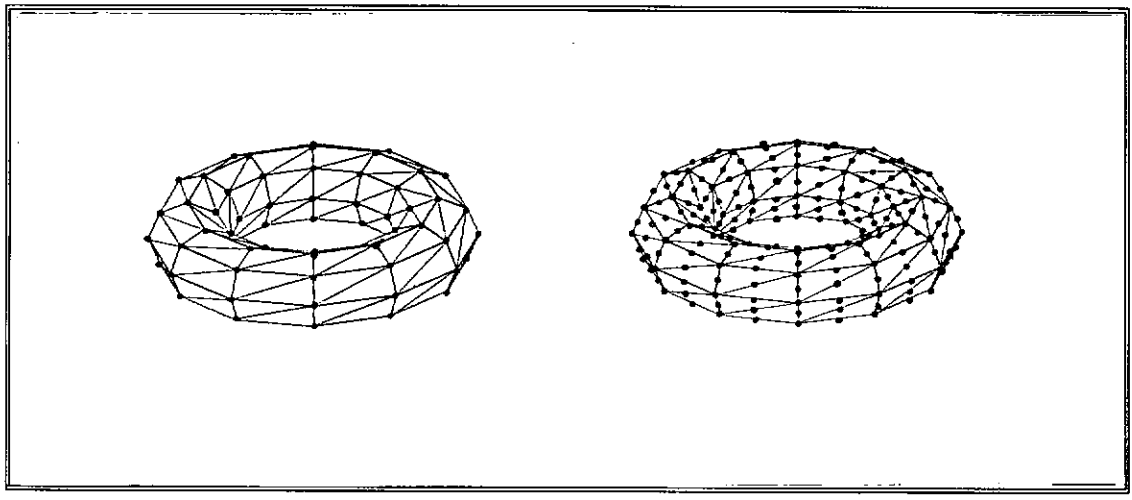


Fig V.73

A malha gerada para o toróide tem eixo de geração passando pelo centro dado e com direção do eixo Z do sistema de coordenadas ativo. Para obter o eixo de geração em outra direção, o usuário poderá efetuar rotações no modelo através dos comandos de edição ou posicionar um sistema de coordenadas local, com o eixo Z na direção desejada para o eixo de geração do hemisfério, antes da execução do comando.

Para finalizar, o usuário é requisitado a aceitar ou rejeitar a malha de elementos finitos gerada. Caso seja rejeitada, toda a malha é descartada automaticamente do modelo.

V.4.6. O COMANDO RESTRIC

O comando **RESTRIC** tem como função definir o tipo de restrição nodal ativa, bem como permitir ao usuário aplicar ou alterar os graus de liberdade de pontos nodais selecionados.

O comando RESTRIC possui quatro opções :

```
?/FRee/FIx/<Point>:
```

– A opção ? apresenta a situação dos graus de liberdade para a restrição nodal ativa.

A seguir apresenta–se uma sequência de operação do comando **RESTRIC** com opção ?.

```
Command> Restric
?/FRee/FIx/<Point>:?

  RX     free
  RY     free
  RZ     fix
  MX     free
  MY     fix
  MZ     free

?/FRee/FIx/<Point>: enter

Command>
```

No exemplo anterior, verifica–se que estão restritos os deslocamentos na direção Z e rotações na direção Y e liberados os demais graus de liberdade.

Executada a opção ?, o comando reapresenta as opções iniciais, permitindo ao usuário prosseguir a operação. Caso deseje encerrar o comando, basta pressionar *ENTER* sem nenhum dado suprido e o sistema concluirá a operação.

– As opções **FREE** e **FIX** têm como função alterar a restrição nodal ativa, liberando ou restringindo os graus de liberdade selecionados.

O usuário será, então, requisitado a definir qual grau de liberdade que deseja liberar ou restringir.

```
DOF to free:(RX/RY/RZ/MX/MY/MZ/ALL)
```

ou

```
DOF to fix:(RX/RY/RZ/MX/MY/MZ/ALL)
```

O usuário poderá realizar seleções sucessivas até a indicação de um valor nulo. Neste momento, o sistema retorna às opções básicas.

A seguir apresenta-se um exemplo de sequência de operações para alteração da restrição nodal ativa.

```
Command> Restric
?/FRee/FIx/<Point>: ?

  RX     free
  RY     free
  RZ     fix
  MX     free
  MY     fix
  MZ     free

?/FRee/FIx/<Point>: Free
DOF to free:(RX/RY/RZ/MX/MY/MZ/ALL) RZ
DOF to free:(RX/RY/RZ/MX/MY/MZ/ALL) enter
?/FRee/FIx/<Point>: Fix
DOF to fix:(RX/RY/RZ/MX/MY/MZ/ALL) MX
DOF to fix:(RX/RY/RZ/MX/MY/MZ/ALL) MZ
DOF to fix:(RX/RY/RZ/MX/MY/MZ/ALL) enter
?/FRee/FIx/<Point>: ?

  RX     free
  RY     free
  RZ     free
  MX     fix
  MY     fix
  MZ     fix

?/FRee/FIx/<Point>: enter
Command>
```

– A opção **POINT** permite ao usuário inserir restrições nodais no modelo.

O símbolo referente às restrições nodais será inserido no ***layer*** RESTRIC e estará relacionado a algum ponto nodal do elemento já existente no modelo. Diferentemente dos elementos, as restrições nodais poderão ser transferidas para outros ***layers*** de acordo com o desejo do usuário.

Para a definição do ponto de inserção da restrição nodal, o usuário será requisitado a selecionar um ponto nodal no modelo. Para indicação do ponto nodal, o usuário tem ao seu dispor todos os recursos de seleção de entidades (*selection set*) padrões do AutoCAD, tais como ***last***, ***previous***, ***window***, ***crossing*** e etc..

O sistema automaticamente inserirá a restrição nodal ativa em todos os pontos nodais selecionados.

A seguir apresenta-se um exemplo de sequência de operação do comando RESTRIC com opção **POINT**.

```
Command> Restric
?/FRee/FIx/<Point>: W
First point: 0,0
Second point: 2,10
5 nodes selected
?/FRee/FIx/<Point>: 10,4
?/FRee/FIx/<Point>: enter
Command>
```

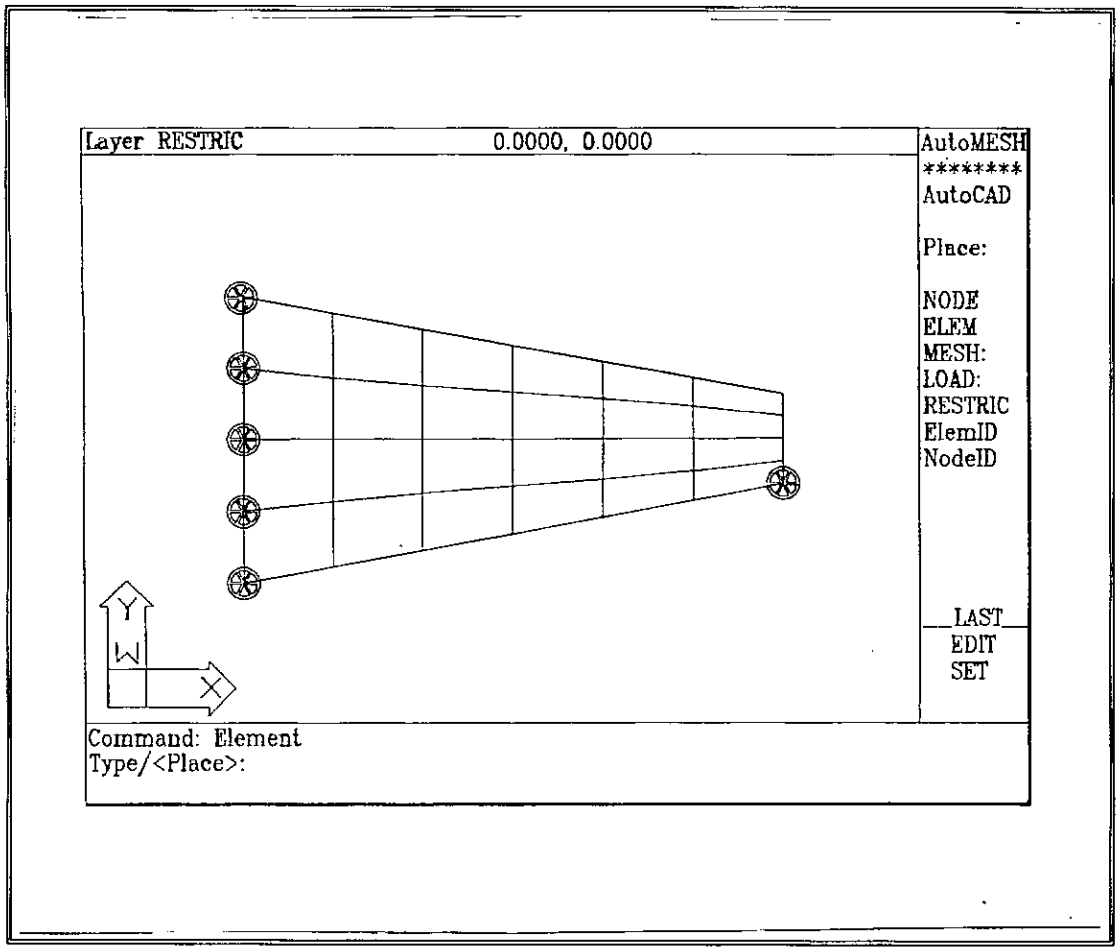


Fig V.74

As restrições nodais existentes são substituídas pelas ativas nos pontos nodais selecionados.

V.4.7. O COMANDO LOAD

O comando **LOAD** permite ao usuário definir os carregamentos aplicados ao modelo.

Os carregamentos foram divididos em três grupos básicos segundo o ponto de aplicação.

- **Nodal**
- **Elementos reticulados** :
  - Concentrado
  - Linear
- **Elementos finitos** :
  - Plano

O comando possui como opções exatamente estes tipos, além da opção que permite a edição de carregamentos já definidos.

```
<Nodal>/Concentrated/Linear/Surface/Edit:
```

Para todos os tipos de carga, o usuário será requisitado a definir o tipo entre força e momento.

```
<Force>/Moment:
```

O sistema assume como *default* a opção **FORCE**

– A opção **NODAL** permite ao usuário definir e aplicar carregamentos a pontos nodais do modelo.

Para esta opção, o usuário será requisitado a definir o tipo de carga, direção, módulo e ponto de aplicação.

```
<Force>/Moment:
Vector:<default>
Module:<default>
Select node:
```

O dado **VECTOR** corresponde ao *versor* (vetor unitário) que define a direção de aplicação da carga. O valor dado será normalizado pelo sistema. O usuário, portanto, não precisará preocupar–se em fornecer um vetor de comprimento unitário.

Ao usuário estão também disponíveis todos os recursos padrões de definição de direção do **AutoCAD**, tais como coordenadas polares, dois pontos e outros.

Os valores adotados como *default* para versor e módulo são os utilizados no último comando **LOAD** executado.

Para indicação do ponto nodal, o usuário tem ao seu dispor os recursos de seleção de entidades (*selection set*) padrões do **AutoCAD**, tais como ***last***, ***window***, ***crossing***, coordenadas e etc..

– A opção **CONCENTRATED** permite ao usuário definir e aplicar cargas concentradas em elementos finitos do tipo reticulado.

Para esta opção, o usuário é requisitado a definir o tipo de carregamento, direção, módulo, ponto de aplicação da carga, bem como o elemento a ser carregado.

```
<Force>/Moment:
Vetor:<default>
Module:<default>
Distance from first end:<Value>/Ratio:
Select line element:
```

Aqui os dados **VECTOR** e **MODULE** representam a direção e módulo do carregamento, como exposto anteriormente para a opção **NODAL**.

Para definir o ponto de aplicação da carga concentrada sobre o elemento reticulado, o usuário tem duas opções: **VALUE** e **RATIO**, que correspondem à distância em valor absoluto ou relativo ao comprimento total do elemento, em relação ao primeiro extremo respectivamente.

O sistema verifica a validade da distância absoluta fornecida pelo usuário e caso seja maior que o comprimento do elemento, o usuário é requisitado a fornecer um novo valor.

Entenda–se por primeiro extremo, a extremidade do elemento reticulado que corresponde à primeira incidência nodal deste.

Para a seleção do elemento a receber o carregamento, o usuário tem ao seu dispor os recursos padrões de seleção de entidades (*selection set*) do **AutoCAD**.

O sistema se encarregará de separar para processamento, apenas os elementos do tipo reticulado.

– A opção **LINEAR** permite ao usuário definir e aplicar cargas distribuídas sobre elementos finitos do tipo reticulado.

Para esta opção, o usuário é requisitado a definir o tipo de carregamento, direção, módulo, ponto de aplicação da carga, bem como o elemento a ser carregado.

Os carregamentos distribuídos estão divididos em dois grupos, segundo a variação do módulo entre os valores definidos pelo usuário; triangular e parabólico, para variação linear e parabólica do módulo do carregamento respectivamente.

```
<Force>/Moment:
Parabolic/<Triangular>:
Module on start:<default>
Module on mid:<default>
Module on end:<default>
Select line element:
```

As opções **FORCE** e **MOMENT** definem o tipo da carga, força ou momento, como exposto anteriormente.

As opções **PARABOLIC** e **TRIANGULAR** definem o tipo de distribuição do carregamento.

Estas opções, em conjunto com os valores dos módulos, definem o aspecto do carregamento.

A opção **TRIANGULAR** é adotada pelo sistema como *default.*

O dado **VECTOR** define a direção de aplicação da carga.

Apesar de permitido, o usuário deverá evitar casos de carregamento distribuído na direção do eixo da peça devido a singularidade na representação gráfica do carregamento. Para estes casos, o sistema emitirá uma mensagem de alerta ao usuário.

Os carregamentos distribuídos serão aplicados em toda a extensão do elemento reticulado. O usuário deverá, então, definir os valores de módulo nos extremos e ponto médio da peça.

O valor *default* adotado para o módulo no primeiro extremo do elemento corresponde ao valor utilizado no último comando **LOAD** executado. Para os valores do módulo no outro extremo e ponto médio, o sistema adota como *default* o valor dado no primeiro extremo, o que possibilita a definição facilitada de carregamentos uniformemente distribuídos.

Como na opção **CONCENTRATED**, para a seleção do elemento a receber o carregamento, o usuário tem ao seu dispor os recursos padrões de seleção de entidades (*selection set*) do **AutoCAD**.

O sistema se encarregará de separar para processamento, apenas os elementos do tipo reticulados.

– A opção **SURFACE** permite ao usuário definir e aplicar carregamentos uniformemente distribuídos em elementos finitos triangulares ou quadriláteros.

Por carregamento uniformemente distribuído, entenda–se o carregamento onde o valor do módulo da carga é igual em todos os pontos do elemento.

Foi assumido apenas este tipo de distribuição de carga para esta opção, devido à dificuldade do usuário em definir a ordem da incidência de cada elemento finito, quando se deseja aplicar o carregamento a vários elementos.

Para simular carregamentos não uniformes em regiões da malha, o usuário deverá aplicar o carregamento uniforme por grupos de elementos como exemplificado na figura V.75.

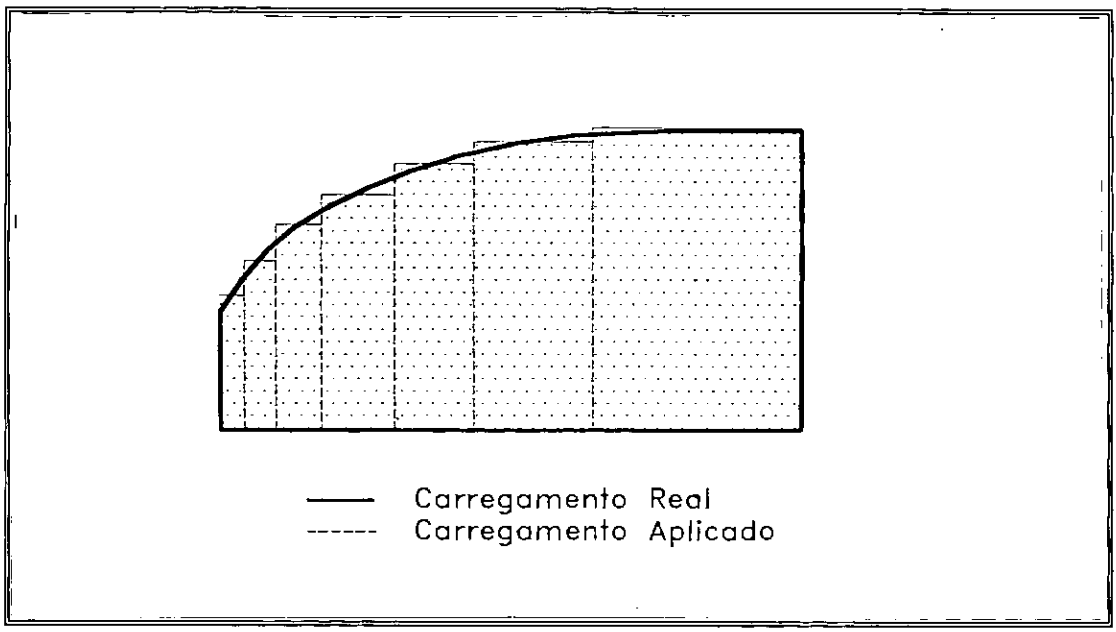


Fig V.75

Segundo o que foi apresentado acima, o usuário será requisitado a definir o tipo de carregamento, direção, módulo e o elemento a ser carregado.

```
<Force>/Moment:
Vector:<default>
Module:<default>
Select element:
```

Os valores adotados como *default* para direção e módulo correspondem aos valores utilizados no último comando LOAD executado.

Para seleção do elemento a receber o carregamento, o usuário tem ao seu dispor os recursos padrões de seleção de entidades (*selection set*) do **AutoCAD**.

O sistema se encarregará de separar para processamento, apenas os elementos do tipo triangular e quadrilátero.

As figuras V.76 e V.77 apresentam a representação gráfica típica para cada tipo de carga e ponto de aplicação.

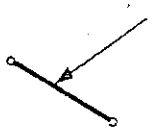
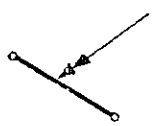
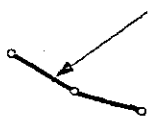
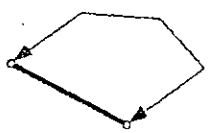
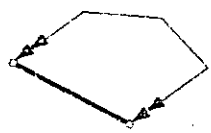
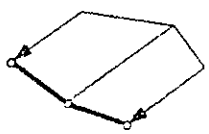
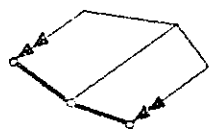
Tipo	Aplicacao	Forca	Momento
Nodal	NODE		
Concen-trated	BAR		
	ISOBAR		
Linear Triangular	BAR		
	ISC BAR		

Fig V.76

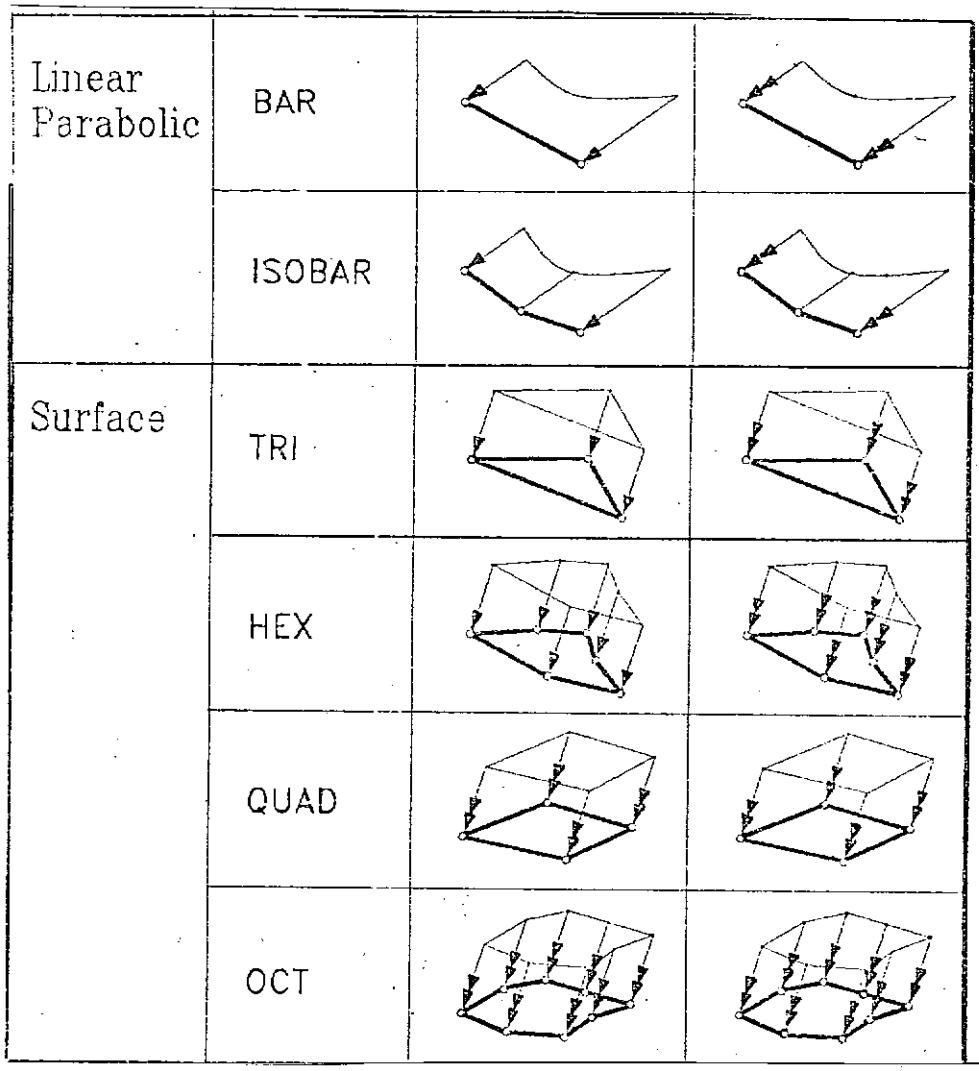


Fig V.77

Deve–se observar que elementos de interface ou contato e elementos infinitos não recebem carregamento.

Os carregamentos de um modo geral são representados por entidades do tipo ***block*** e sua existência é função do nó ou elemento a que foi aplicado, ou seja, a retirada (*delete*) de um nó ou elemento do modelo ocasiona a retirada dos carregamentos a ele aplicados.

Os carregamentos quando aplicados no modelo, ficarão naturalmente residentes no ***layer*** LOADS. Nenhuma restrição, porém, é imposta quanto à troca deste ***layer***.

– A última opção do comando **LOAD** é a EDIT e é responsável pela alteração dos valores que definem os carregamentos.

Para sua execução, o usuário será requisitado a selecionar o carregamento. Para esta seleção, o usuário terá ao seu dispor os recursos padrões de seleção de entidades (*selection set*) do **AutoCAD**, tais como ***window***, ***crossing***, ***last*** e outros.

De acordo com o tipo de carregamento selecionado, o sistema interagirá com o usuário de forma a permitir a sua alteração.

O comando permite que grupos de carregamentos diferentes sejam selecionados. Para estes casos, o sistema se encarregará de separar os grupos e processar por etapas e diferenciadamente.

Os grupos de elementos serão processados segundo a seguinte ordem:

. **Nodal**
. **Concentrated**
. **Linear**
. **Surface**

Para cada grupo, os valores fornecidos pelo usuário serão aplicados a todos os carregamentos.

Para o caso de carregamentos nodais, o sistema apresentará as seguintes opções de edição:

```
Type/Vector/Module:
```

Apenas os dados selecionados são processados, os demais permanecem com seus valores iniciais.

Se nenhuma opção é selecionada, a execução é encerrada e nenhuma alteração é processada.

O sistema apresentará como *default* para cada uma das opções acima o valor inicial, se em todos os carregamentos selecionados for igual, ou **VARIES** caso forem diferentes.

Abaixo apresenta–se um exemplo de sequência de operação do comando **LOAD** para edição de carregamentos nodais.

```
Command> Load
<Nodal>/Concentrated/Linear/Surface/Edit: E
Select load: W
First point: 0,0
Second point: 10,10
 5 nodal loads selected
Select load: enter
Nodal load data to edit:(Type/Vector/Module) T
Load Type:<force> Moment
Nodal load data to edit:(Type/Vector/Module) M
Module:<varies> 10.0
Nodal load data to edit:(Type/Vector/Module) enter
Command>
```

Para o caso de carregamentos do tipo **CONCENTRATED**, o usuário terá as seguintes opções de edição.

```
Type/Vector/Module/Position:
```

Para as opções **TYPE**, **VECTOR** e **MODULE**, o exposto para carregamentos nodais aplica–se para cargas concentradas. A opção **POSITION** permite reposicionar o carregamento relativamente aos extremos do elemento reticulado. O usuário poderá fornecer este valor como absoluto ou relativo ao comprimento do elemento.

Para a edição de grupos de carregamentos, o sistema verifica a validade da nova posição fornecida para cada elemento.

A seguir apresenta–se um exemplo de sequência de operações do comando **LOAD** para a edição de carregamentos em elementos reticulados.

```
Command> Load
<Nodal>/Concentrated/Linear/Surface/Edit: E
Select load: W
First point: 0,0
Second point: 10,10
 5 concentrated loads selected
Select load: enter
Concentrated load data to edit:(Type/Vector/Module
 /Position) V
Vector:<1,0,0> 0,1,0
Concentrated load data to edit:(Type/Vector/Module
 /Position) P
Distance from first end:<5.0>/Ratio: R
Ratio:<0.2> 0.5
Concentrated load data to edit:(Type/Vector/Module
 /Position) enter
Command>
```

Para o caso de carregamentos do tipo **LINEAR**, o usuário terá as seguintes opções de edição:

```
Type/VAriation/VEctor/Module:
```

A opção **VARIATION** permite ao usuário alterar o tipo de distribuição, parabólica ou triangular.

A execução da opção **MODULE** faz com que o sistema requisite ao usuário novos valores para os módulos nos extremos e ponto médio do elemento.

Abaixo apresenta–se um exemplo de sequência de operação do comando **LOAD** para a edição de carregamentos distribuídos em elementos finitos reticulados.

```
Command> Load
<Nodal>/Concentrated/Linear/Surface/Edit: E
Select load: W
First point: 0,0
Second point: 10,10
 5 linear loads selected
Select load: enter
Linear load data to edit: (Type/VAriation/VEctor/
 Module) VA
Distributed type:<linear> parabolic
Linear load data to edit:(Type/Variation/Vector/
 Module) M
Module on start:<varies> 5.0
module on mid:<varies> 20.0
Module on end:<varies> 15.0
Linear load data to edit:(Type/Variation/Vector/
 Module) enter
Command>
```

Para o caso de carregamentos do tipo **SURFACE**, o usuário terá como opções de edição :

```
Type/Vector/Module:
```

Estas opções são semelhantes às apresentadas para os outros tipos de carregamentos.

Abaixo apresenta-se um exemplo de um caso geral onde o usuário seleciona carregamentos de tipos diferentes para editar.

```
Command> Load
<Nodal>/Concentrated/Linear/Surface/Edit: E
Select load: W
First point: 0,0
Second point: 10,10
 5 nodal loads selected
 2 concentrated loads selected
 3 surface loads seleted
Select load: enter
Nodal load data to edit:(Type/Vector/Module) M
Module:<10.0> 20.0
Nodal load data to edit:(Type/Vector/Module) enter
Concentrated load data to edit:(Type/Vector/
 Module/Position) M
Module:<varies> 20.0
Concentrated load data to edit:(Type/Vector/
 Module/Position) enter
Surface load data to edit:(Type/Vector/Module) M
Module:<varies> 20.0
Surface load data to edit:(Type/Vector/Module)
 enter
Command>
```

V.4.8. COMANDOS DE EDIÇÂO

Os comandos de edição são todos aqueles que permitem ao usuário alterar o modelo.

Os comandos de edição, quando enumerados no início do capítulo, foram divididos em três grupos de acordo com o grau de diferenciação dos comandos básicos de edição do **AutoCAD**.

Ao executar–se o comando **STARTUP** descrito anteriormente, o sistema **AutoMESH** redefine todos os comandos de edição básicos do **AutoCAD**, quer por ampliação das capacidades ou restrição da aplicação.

O grupo EDIT constitui o grupo de comandos com funções mais poderosas que as oferecidas ou inexistentes dentre os comandos básicos do **AutoCAD**.

Dentre os comandos básicos de edição do **AutoCAD**, destacam–se dois grupos segundo o tipo de alteração provocada na execução pelo sistema **AutoMESH**.

. Execução Alterada

. Execução Restrita

Os comandos com execução alterada correspondem aqueles que sofrerão alteração no modo de operação, devido ao tratamento diferenciado que devem dar às diversas entidades de significado estrutural.

Os comandos com operação restrita são aqueles em que as operações de edição não se aplicam às entidades com significado estrutural. O sistema **AutoMESH**, através de uma técnica de filtro impede que certos comandos básicos do **AutoCAD** se apliquem sobre entidades gráficas de significado estrutural.

Dentre estes comandos podemos enumerar:

. Move
. Erase
. Break
. Pedit
. Explode
. Trim
. Extend
. Change
. Stretch
. Offset

Isto serve para evitar que o usuário acidentalmente, por exemplo, exploda um ***polyline*** que representa um elemento finito ou através do comando ***break***, parta–o em dois.

A técnica de filtro adotada pelo sistema, constitui–se basicamente em verificar e retirar do grupo de elementos selecionados, todos aqueles que tenham significado estrutural, antes da realização da operação de edição.

Em resumo, os comandos básicos de edição do **AutoCAD** continuam disponíveis e executam suas funções exatamente como antes, quando se trata de entidades gráficas, as quais não tenham nenhum significado estrutural para o sistema **AutoMESH**.

V.4.8.1. O COMANDO MERGE

O comando **MERGE** permite ao usuário eliminar do modelo, pontos nodais coincidentes, considerando uma dada tolerância.

O sistema faz com que dentre os pontos nodais selecionados, os que se distanciarem entre si de um valor menor que uma tolerância, convirjam para um único ponto.

Normalmente esta tarefa de convergência a um ponto, para nós considerados coincidentes, é feita automaticamente nos comandos de geração automática ou com o auxílio do comando ***osnap*** do **AutoCAD** para geração de elementos isolados pelo usuário.

Existirão casos, no entanto, em que interessa a existência de pontos nodais coincidentes como no caso de estudos de fraturas. Em outros casos, porém, o usuário deseja forçar uma convergência não obtida automaticamente.

Em geral, a tolerância adotada em convergências automáticas corresponde à dimensão ativa para símbolos nodais, definida através do comando **NODE** opção **SCALE**.

Ao acionar o comando **MERGE**, o usuário será requisitado a selecionar os pontos nodais a serem submetidos. Para seleção, o usuário terá ao seu dispor todos os métodos padrões de seleção de entidades do **AutoCAD**.

Terminada a seleção dos pontos nodais, o usuário será requisitado então, a especificar um valor para a tolerância entre nós coincidentes. O sistema apresentará como *default*, o valor correpondente à dimensão ativa para símbolos nodais.

A figura V.78 apresenta um exemplo de execução do comando **MERGE** com tolerância igual à dimensão ativa de símbolos nodais, aplicado sobre nós com símbolos nodais com dimensões iguais à ativa.

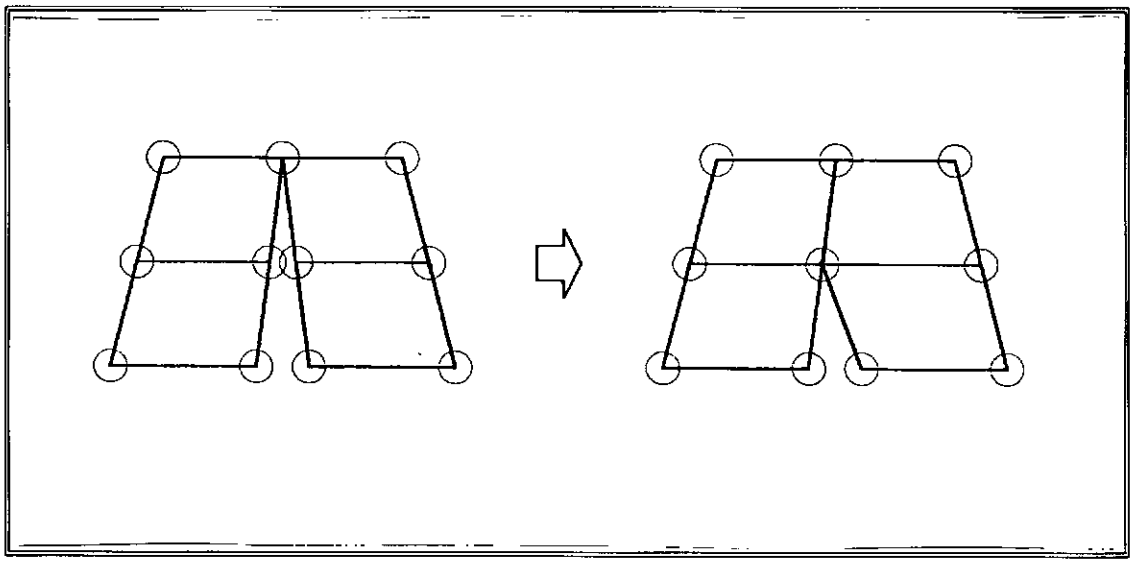


Fig V.78

Os pontos nodais coincidentes serão levados a convergir para o primeiro ponto nodal inserido no modelo para os nós em questão.

A seguir apresenta–se um exemplo de sequência de operações do comando **MERGE**.

```
Command> Merge
Select node: W
First point: 0,0
Second point: 10,10
 13 selected
Select node: enter
Tolerance:<1.0> 2.0
Command>
```

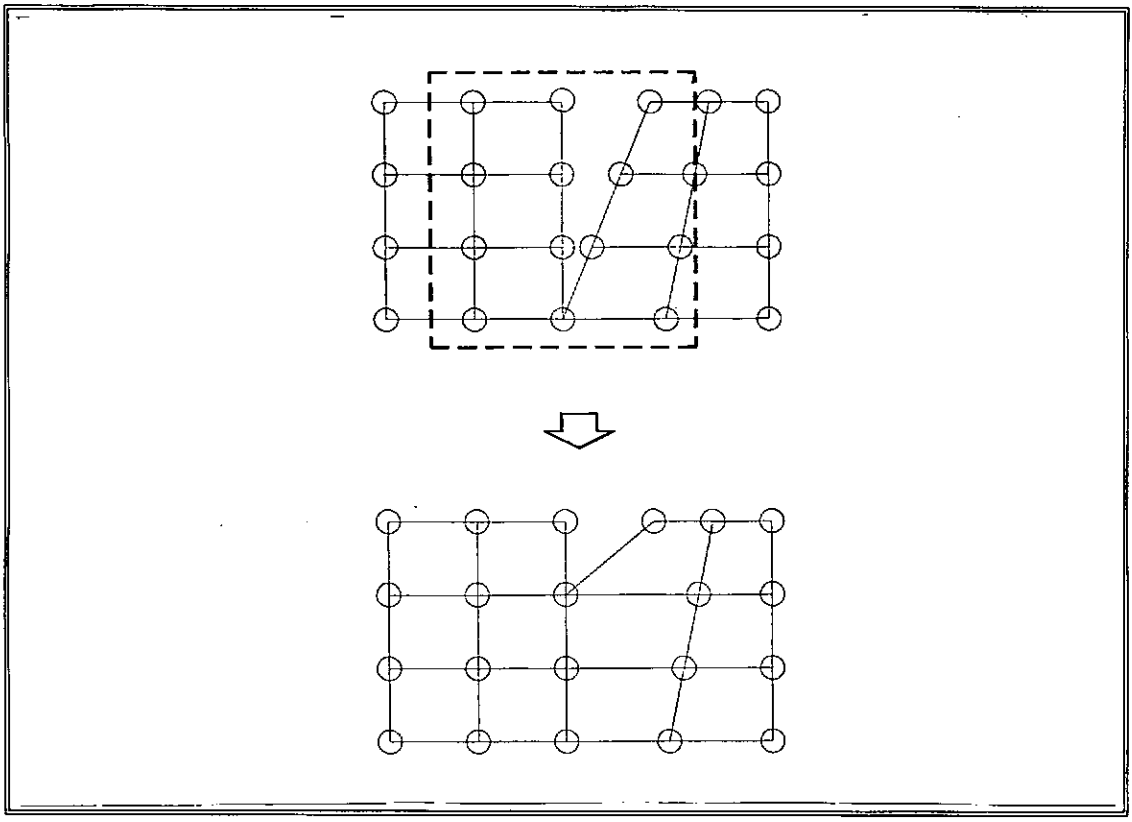


Fig V.79

Caso a posição obtida para os nós convergidos não seja a desejada pelo usuário, este poderá mover os pontos nodais desejados através do comando **DESLOC** que será apresentado nas próximas páginas.

Na figura V.80 apresenta–se um exemplo onde foi aplicada uma tolerância muito grande para um grupo de nós selecionados, fazendo–se com que todos convergissem para um único ponto.

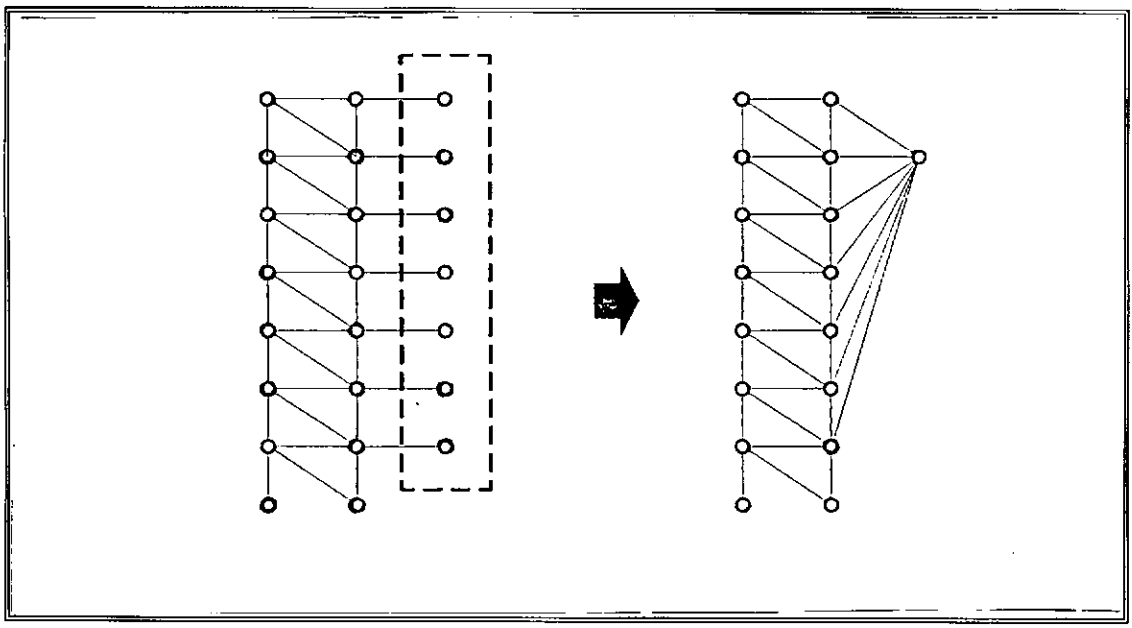


Fig V.80

O processo de convergência é realizado diretamente ao ser constatada uma distância inferior à tolerância. Isto pode fazer com que não aconteça a convergência para um único ponto nodal como ocorreria inicialmente.

A figura V.81 apresenta o mesmo exemplo da figura V.80, porém, com uma tolerância igual a duas vezes a distância entre os pontos nodais.

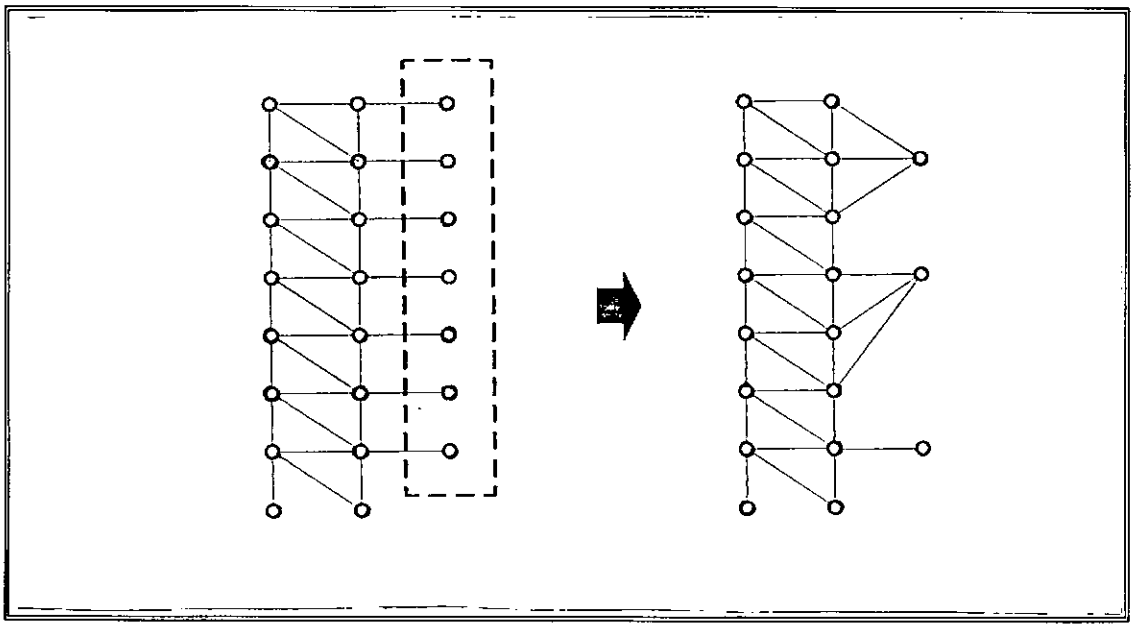


Fig V.81

O usuário deverá estar atento para a utilização deste comando devido a possibilidade de alterações nos tipos de elementos ou inclusão de inconsistências no modelo.

A figura V.82 apresenta algumas inconsistências do tipo elementos com dimensões nulas, que podem ser introduzidas no modelo com o uso do comando **MERGE**.

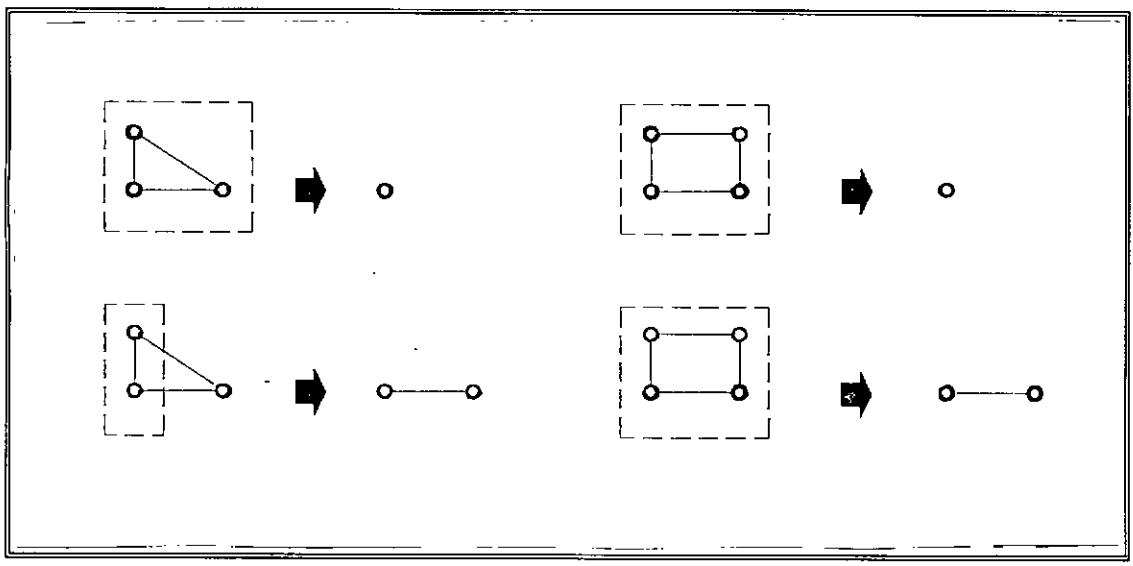


Fig V.82

Este tipo de inconsistência é detectada e os elementos eliminados do modelo, quando da geração do arquivo de dados através do comando **DATABASE**, que será apresentado no decorrer deste capítulo.

A figura V.83 apresenta alguns exemplos de alteração na forma dos elementos causados pelo uso do comando **MERGE**.

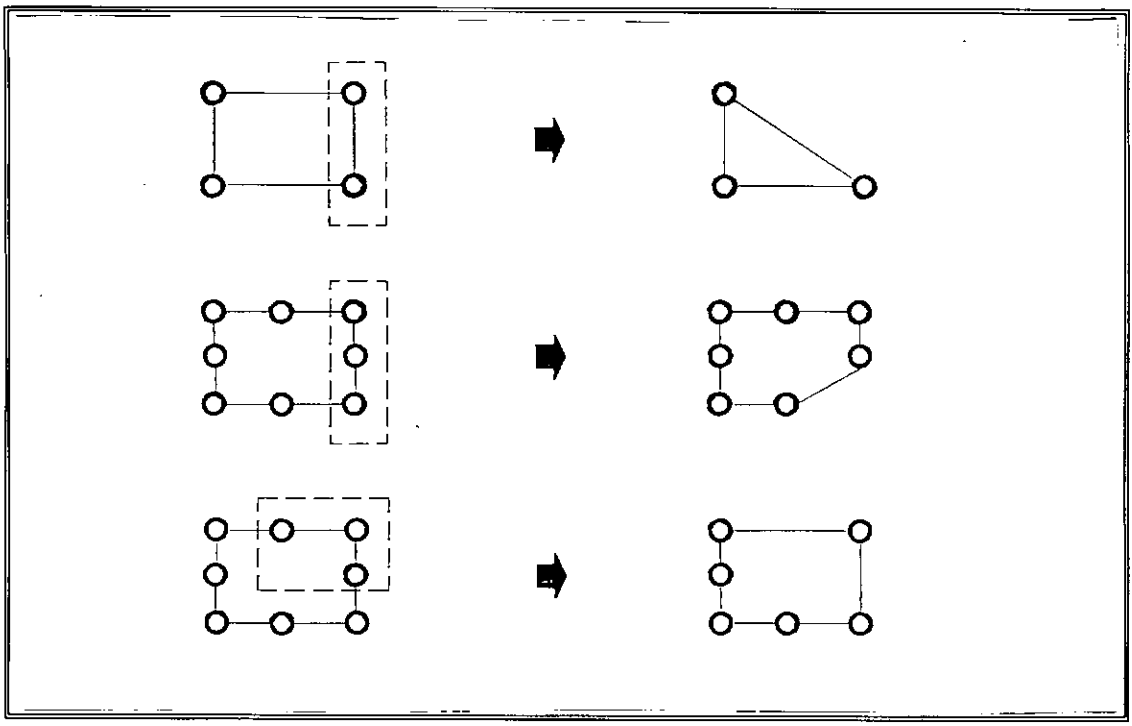


Fig V.83

Este tipo de alteração de forma é detectado pelo sistema, quando da geração do arquivo de dados do modelo, e a incidência correta é montada para o elemento considerando os nós coincidentes.

Deve–se esclarecer que os pontos nodais de um elemento continuam existindo apesar de coincidentes, ou seja, apesar de visualmente apresentarem, por exemplo, três pontos nodais, um elemento do tipo **QUAD** que tenha dois de seus vértices convergidos para um único ponto pelo comando **MERGE**, continua a ser representado graficamente por um ***polyline*** fechado de quatro vértices.

A alteração do formato de elementos pela utilização do comando **MERGE** pode ser especialmente útil quando se deseja utilizar elementos de primeira ordem em uma parte da malha e de segunda ordem em outra, como exemplificado na figura V.84.

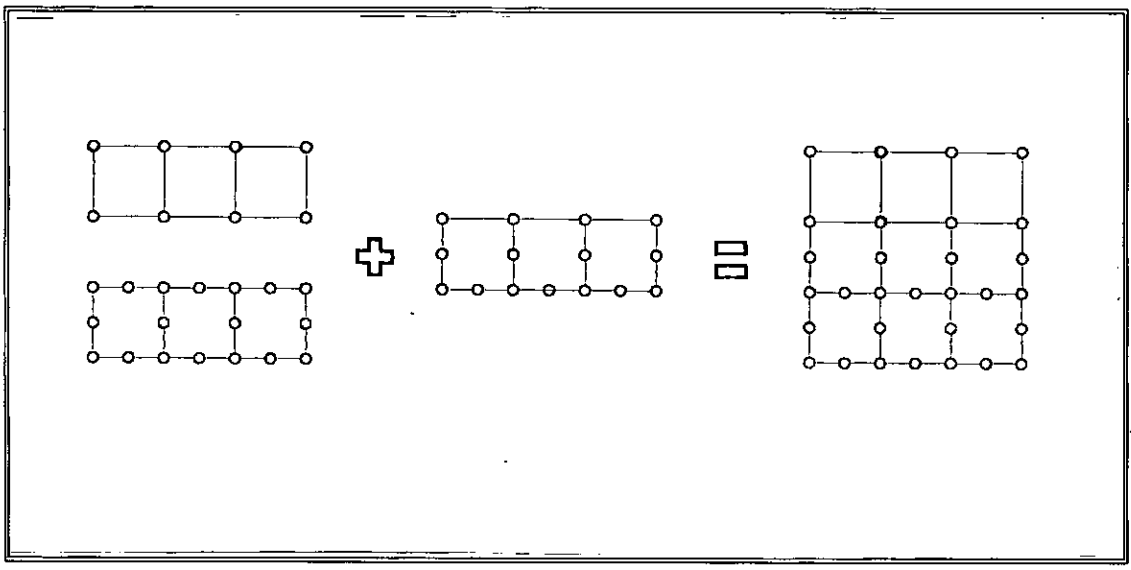


Fig V.84

Na figura V.84 os elementos da faixa superior são de primeira ordem, os da faixa inferior de segunda ordem, e os elementos da faixa intermediária são de segunda ordem, com os nós médios das arestas superiores coincidentes com os nós dos vértices.

O mesmo resultado pode ser obtido com o comando **DESLOC** a ser apresentado posteriormente.

Conforme são selecionados pelo usuário, os pontos nodais são evidenciados no desenho (*highlight*) para verificação visual, permitindo então através do *selection set* padrão do **AutoCAD** que mais elementos possam ser incluídos (***add***) ou retirados (***remove***) do grupo a ser editado.

V.4.8.2. O COMANDO BREAKEL

O comando **BREAKEL** permite ao usuário substituir elementos finitos quadriláteros por triangulares de mesma ordem.

Este comando é especialmente útil quando aplicado a elementos quadriláteros demasiadamente distorcidos, pois possibilita a obtenção de elementos triangulares melhor condicionados.

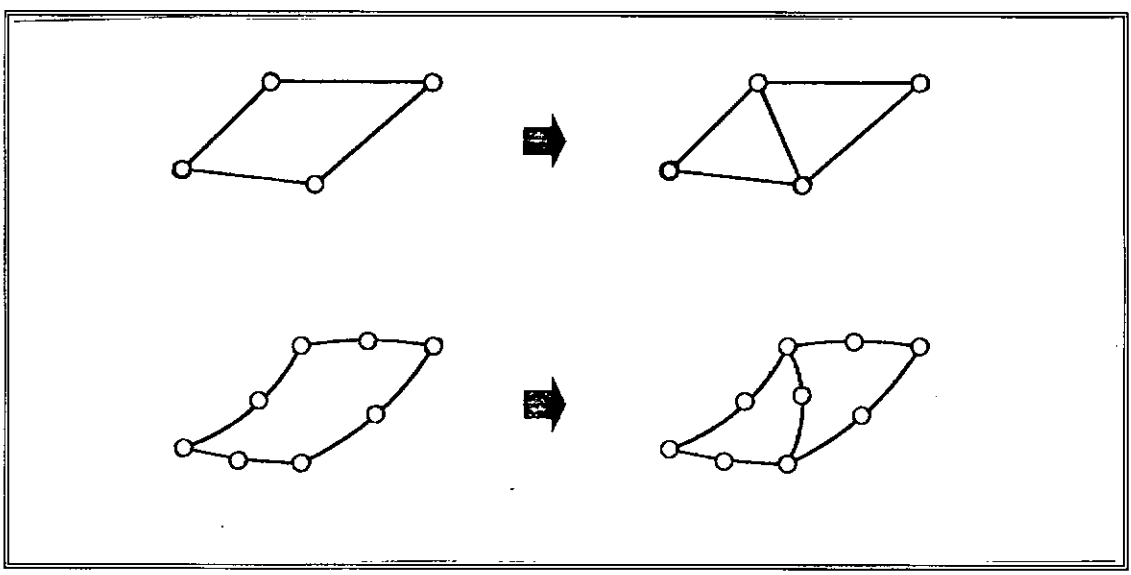


Fig V.85

A determinação do par de elementos triangulares que melhor substituem o elemento quadrilátero é feita utilizando a mesma técnica dos comandos de geração automática apresentados anteriormente.

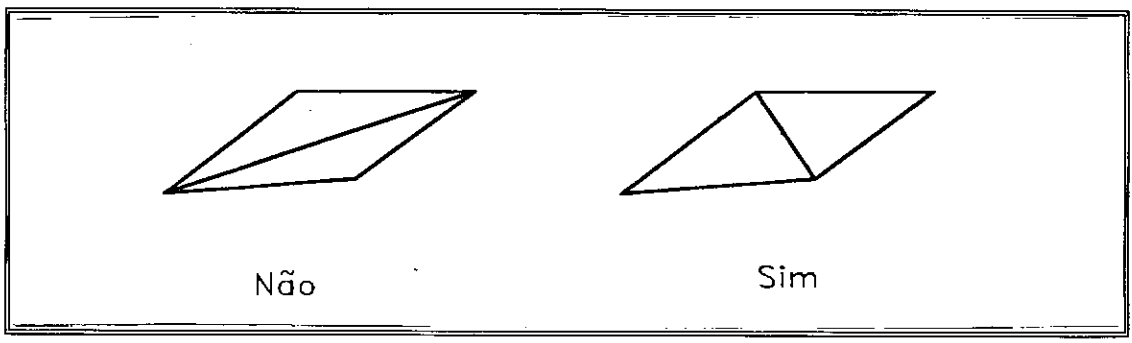


Fig V.86

Ao acionar o comando, o usuário é requisitado a selecionar os elementos a serem substituídos. Para esta seleção, o usuário tem ao seu dispor todos os métodos padrões de seleção de entidades do **AutoCAD**.

A seguir–se apresenta uma sequência de operação do comando **BREAKEL**.

```
Command> Breakel
Select element: 5,5
Select element: W
First point: 0,0
Second point: 10,2
 3 elements selected
Select element: enter
Command>
```

Os novos elementos triangulares terão a mesma propriedade do elemento quadrilátero que os originou, ficando assim residentes no *layer* correspondente a este material.

Para elementos de segunda ordem, um novo ponto nodal no centro do antigo elemento é gerado. O símbolo que representa este ponto nodal tem suas dimensões determinadas segundo o valor ativo definido através do comando **NODE** opção **SCALE**.

Conforme são selecionados pelo usuário, os elementos são evidenciados (*highlight*) para verificação visual, permitindo então através do *selection set* padrão do **AutoCAD**, que mais elementos possam ser incluídos ou retirados do grupo a ser editado.

V.4.8.3. O COMANDO DESLOC

O comando **DESLOC** permite ao usuário mover ou trocar de posição nós ou elementos.

O comando **DESLOC** possui duas opções básicas de acordo com o tipo de elemento estrutural ao qual será aplicado.

```
Element/<Node>:
```

– A opção **ELEMENT** permite ao usuário mover elementos finitos do modelo. Ao serem deslocados, os elementos carregam os carregamentos associados a eles. Nenhuma alteração no formato dos elementos é imposta com esta opção.

A seguir apresenta–se um exemplo de sequência de operações para o comando **DESLOC** opção **ELEMENT**.

```
Command> Desloc
Element/<Node>: Element
Select element: W
First point: 0,0
Second point: 6,10
 6 elements selected
Select element: Remove
Remove element: 4,7
Select element: enter
Base point: 0,0
New point: @-1,-2
Command>
```

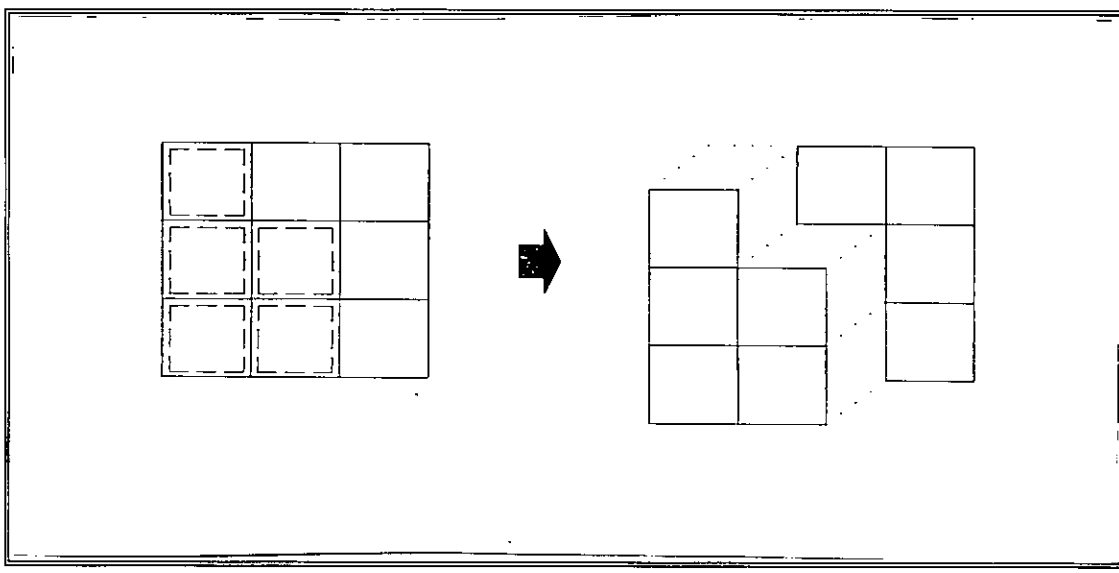


Fig V.87

Os dados **BASE POINT** e **NEW POINT** correspondem ao ponto tomado como base para o deslocamento e sua nova posição respectivamente. O deslocamento relativo aplicado a estes pontos será aplicado a todos os elementos selecionados. Estes pontos não necessitam estar necessariamente nos elementos e suas coordenadas podem ser definidas utilizando–se qualquer dos modos padrões de definição de pontos do **AutoCAD**.

Para este comando, o sistema trata diferenciadamente carregamentos em elementos de carregamentos e restrições nodais.

Os carregamentos nos elementos são deslocados conjuntamente com os elementos ao quais estão associados. Nenhuma alteração nos carregamentos é realizada.

Os carregamentos e restrições nodais em nós pertencentes a elementos selecionados serão deslocados conjuntamente com o elemento se nenhum outro elemento incidir sobre o nó em questão. Caso contrário, um novo ponto nodal é gerado e os carregamentos e restrições permanecerão imóveis.

A figura V.88 apresenta alguns exemplos da aplicação do comando **DESLOC** opção **ELEMENT** em modelos com restrições nodais e carregamentos.

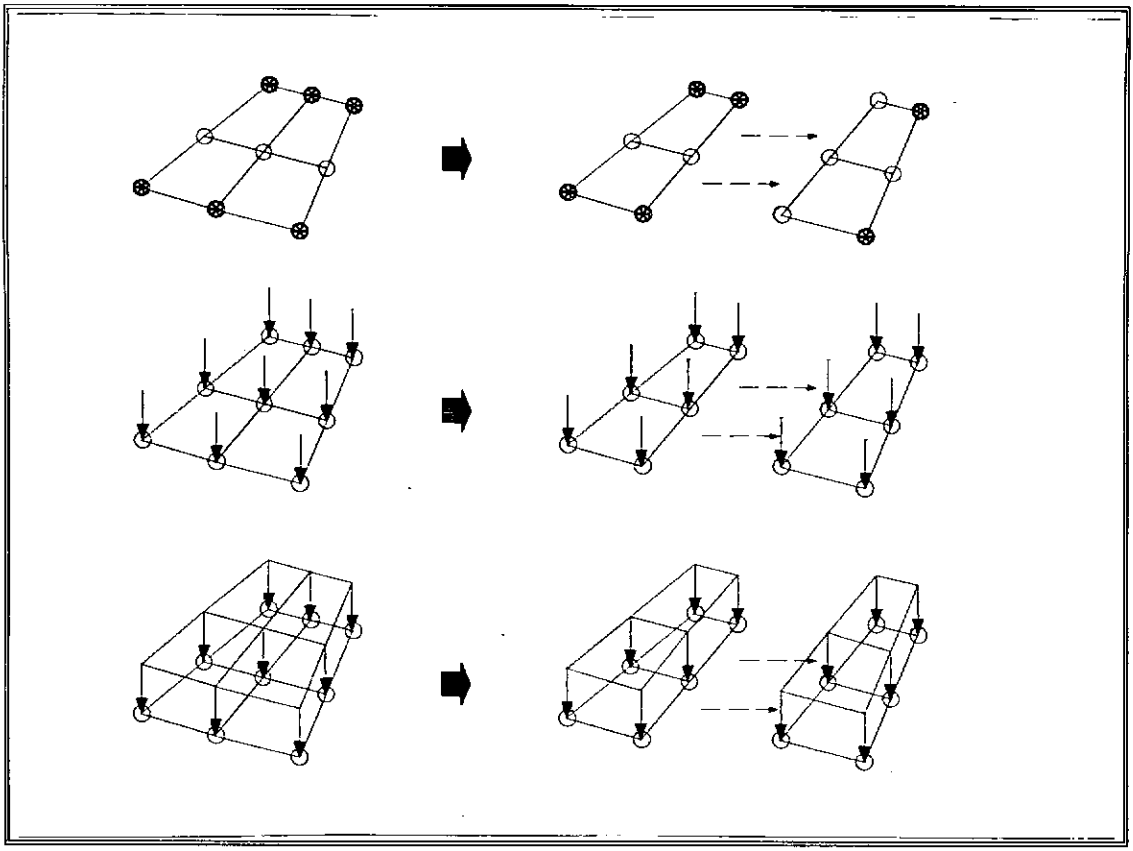


Fig V.88

– A opção **NODE** adotada como *default* para este comando, permite ao usuário deslocar pontos nodais, mantendo a conectividade dos elementos que incidem sobre este.

Com o deslocamento de parte dos pontos nodais que constituem um elemento, provoca–se uma alteração no formato do elemento.

A seguir apresenta-se um exemplo de sequência de operações para o comando **DESLOC** opção **NODE**.

```
Command> Desloc
Element/<Node>: Node
Select node: W
First point: 0,0
Second point: 6,10
 9 selected
Select node: Remove
Remove node: 6,7
Select node: enter
Base Point: 10,10
New point: @-1,-2
Command>
```

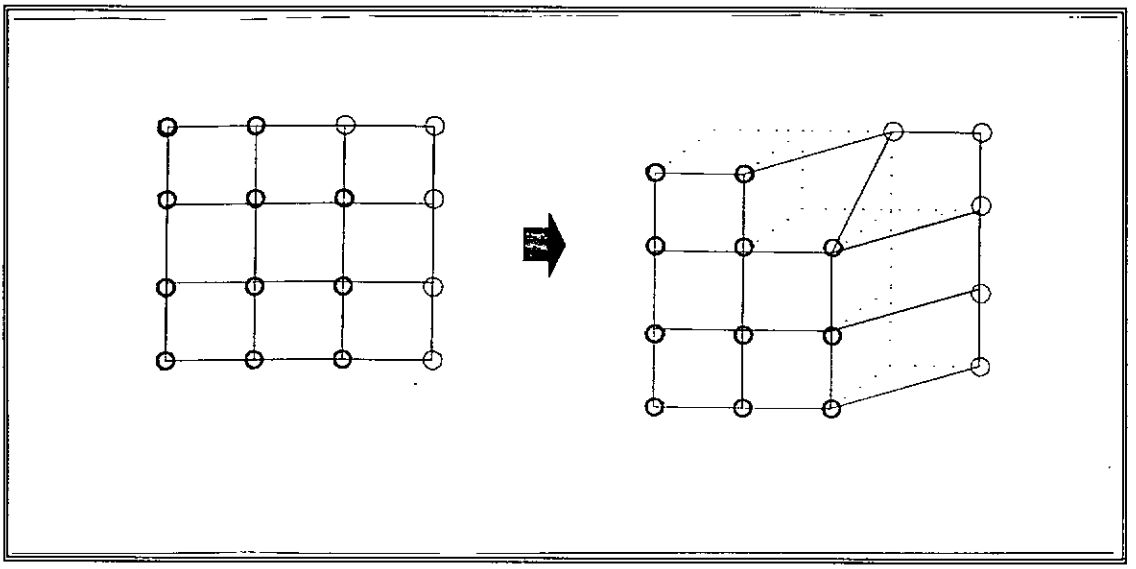


Fig V.99

Pode–se observar que os elementos que tiveram todos os seus nós deslocados, não tiveram alteração no formato.

Para a seleção dos pontos nodais a editar, o usuário tem ao seu dispor todos os recursos padrões de seleção de entidades do **AutoCAD**.

Para este comando, o sistema trata diferenciadamente carregamentos nos elementos de carregamentos e restrições nodais, para os nós selecionados.

Devido ao fato de em nenhum caso serem gerados pontos nodais, os carregamentos e restrições associados a pontos nodais a serem editados, são movidos juntamente com estes.

Para pontos nodais pertencentes a elementos com carregamento associado, tanto o elemento quanto os carregamentos serão distorcidos.

A figura V.90 apresenta alguns exemplos da aplicação do comando **DESLOC** opção **NODE** em modelos com restrições nodais e carregamentos.

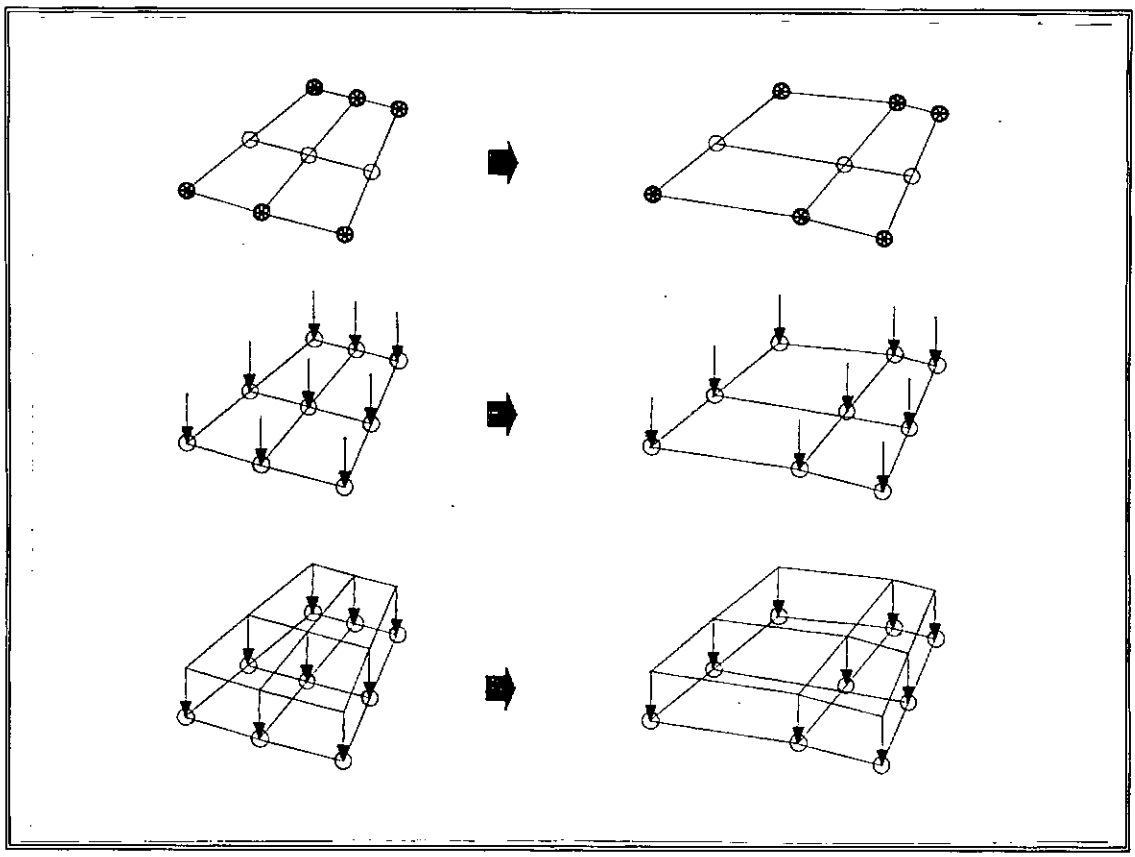


Fig V.90

Como no comando **MERGE**, o deslocamento de pontos nodais pode levar a inconsistências no modelo, tais como elementos de área nula, devido a possibilidade de alteração da forma dos elementos.

A figura V.91 apresenta alguns exemplos de inconsistência no modelo pelo uso do comando **DESLOC** opção **NODE**.

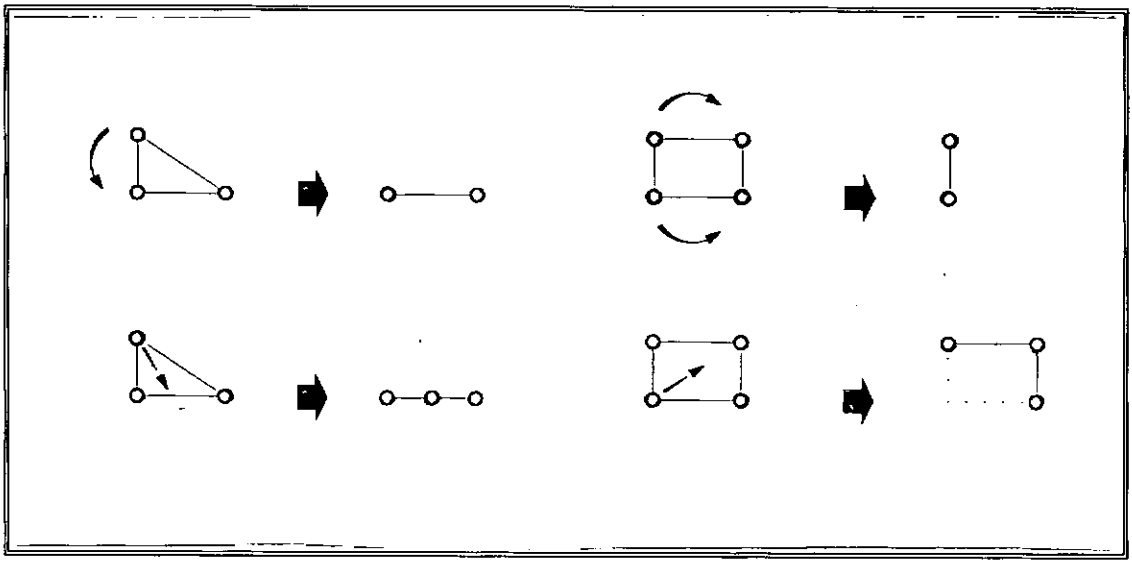


Fig V.91

Este tipo de inconsistência é detectado e os elementos são eliminados do modelo quando da geração do arquivo de dados através do comando **DATABASE** a ser apresentado posteriormente.

Deve–se salientar que, a alteração do tipo de elemento por deslocamento de pontos nodais, tornando–os coincidentes, não é causa de inconsistências. O sistema é capaz de detectar este tipo de alteração no formato e a incidência correta é montada no arquivo de dados do modelo.

A figura V.91 apresenta algumas alterações de tipo ou ordem de elementos pelo deslocamento de pontos nodais.

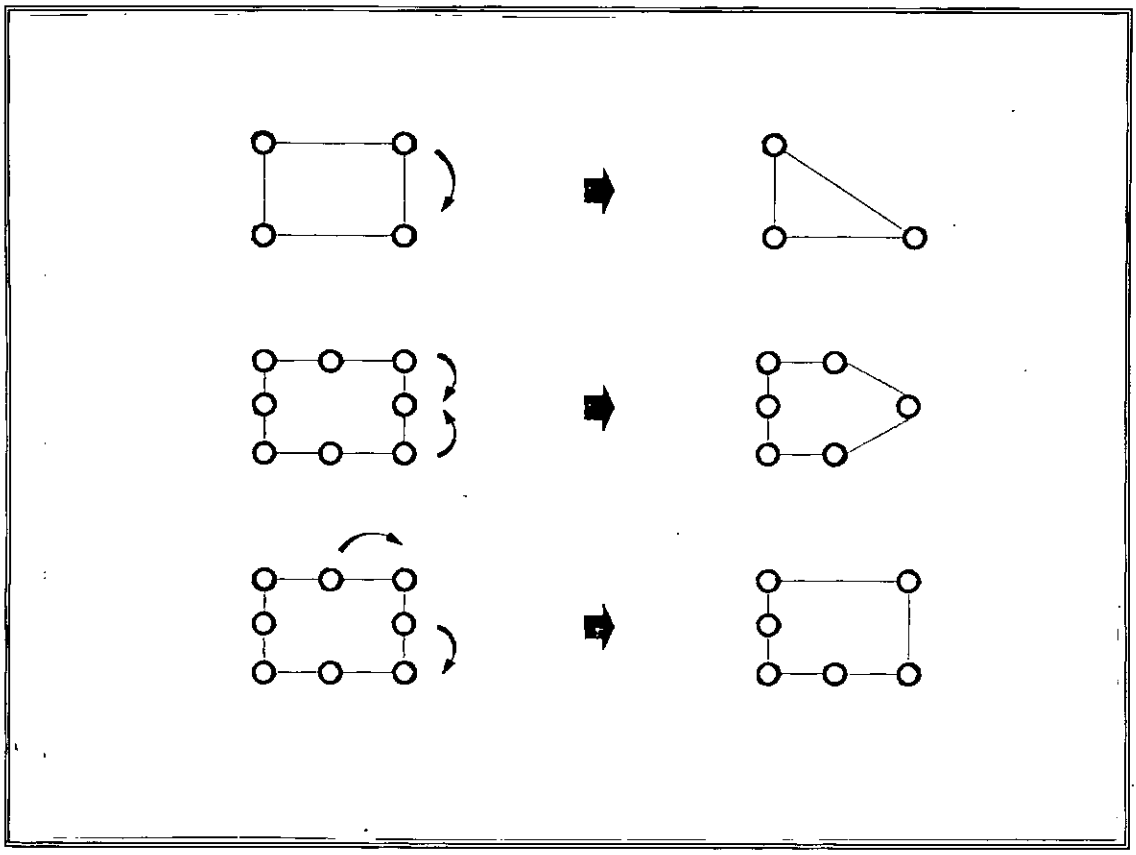


Fig V.92

Deve–se esclarecer que os pontos nodais do elemento continuam existindo apesar de coincidentes, ou seja, apesar de visualmente se apresentar do tipo **QUAD**, por exemplo, com nós coincidentes, este continua a ser representado graficamente por um ***polyline*** fechado de quatro vértices.

Além das inconsistências apresentadas anteriormente, o reposicionamento de pontos nodais pode causar mau condicionamento em alguns elementos da malha por deformação excessiva ou *overlapping.*

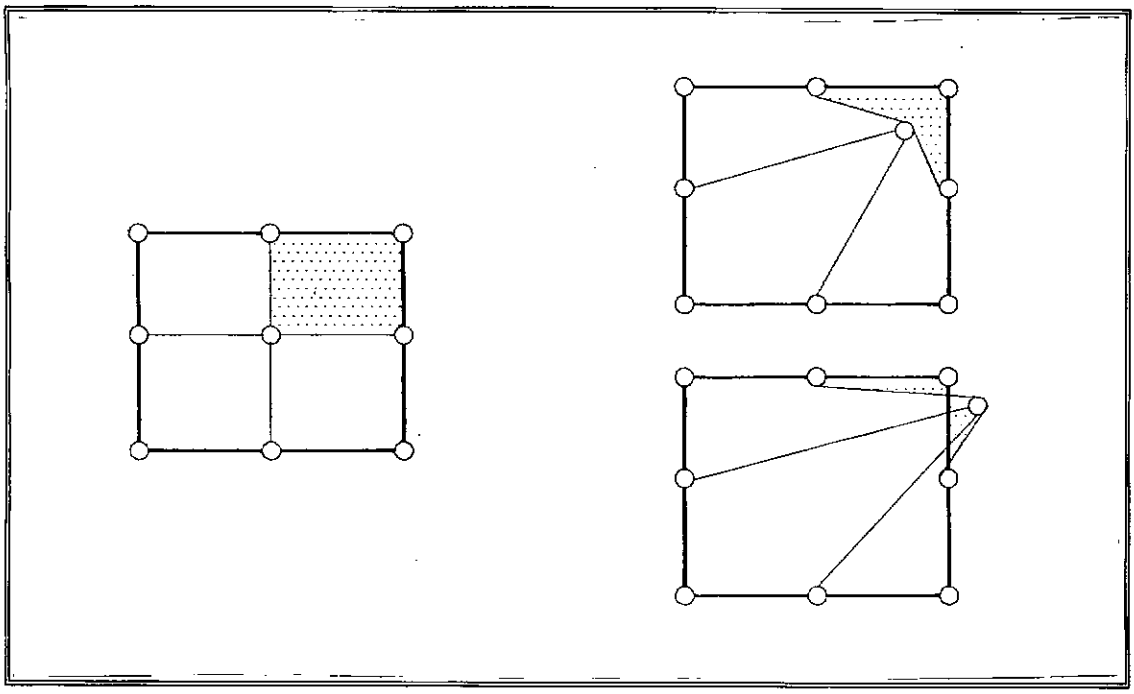


Fig V.93

Este tipo de problema é detectado pelo sistema quando da geração do arquivo de dados e o usuário é alertado.

V.4.8.4. O COMANDO DELETE

O comando **DELETE**, permite ao usuário retirar do modelo elementos estruturais de maneira consistente.

Este comando apresenta como opções de execução os diversos tipos de elementos estruturais.

```
Load/Restric/Element/<Node>:
```

Para cada tipo de elemento estrutural, o sistema realiza um conjunto adicional de operações internas, além da remoção da representação gráfica do desenho.

Para todas as opções, o usuário é requisitado a selecionar o grupo de elementos estruturais a ser retirado do modelo. Para esta seleção, o usuário tem ao seu dispor o *selection set* padrão do **AutoCAD**, ou seja, a seleção pode ser voltada não apenas a um único elemento, mas também a um grupo.

– A opção **LOAD**, permite ao usuário retirar do modelo carregamentos de qualquer tipo.

Para esta opção, o comando **DELETE** assemelha–se ao comando ***erase*** do **AutoCAD**, diferindo–se apenas na seleção interna que realiza entre as entidades gráficas, direcionando a operação apenas àquelas que representam carregamentos.

– A opção **RESTRIC**, permite ao usuário retirar do modelo restrições nodais.

Para esta opção, o sistema realiza uma substituição dos símbolos de restrições nodais por símbolos de pontos nodais, com todas as restrições liberadas se a escala ativa para inserção de simbolos nodais, definida através do comando, **NODE** opção **SCALE**, for diferente de zero, caso contrário nenhum símbolo é colocado.

– A opção **ELEMENT**, permite ao usuário a retirada de elementos finitos do modelo, bem como os carregamentos associados a eles.

Para cada elemento finito selecionado, o sistema verifica a existência de carregamentos associados a este e realiza a retirada de ambos do modelo, impedindo deste modo que venham a existir no modelo, carregamentos sem ponto de aplicação.

– A opção **NODE** adotada como *default*, é a mais complexa e permite a retirada de pontos nodais do modelo.

Para cada ponto nodal do modelo, o sistema verifica a existência de restrições, carregamentos e a incidência de elementos finitos e retira a todos do modelo. A operação é realizada em cadeia quando elementos incidem no ponto nodal retirado, ou seja, para cada ponto nodal retirado do modelo, são retirados também os elementos finitos que incidem neste, e para cada elemento retirado do modelo são também retirados os carregamentos associados a este.

A opção **NODE** do comando **DELETE** pode desencadear a execução interna da opção **ELEMENT**.

A figura V.94 apresenta alguns exemplos de execução do comando **DELETE** opção **NODE** nos diversos passos:

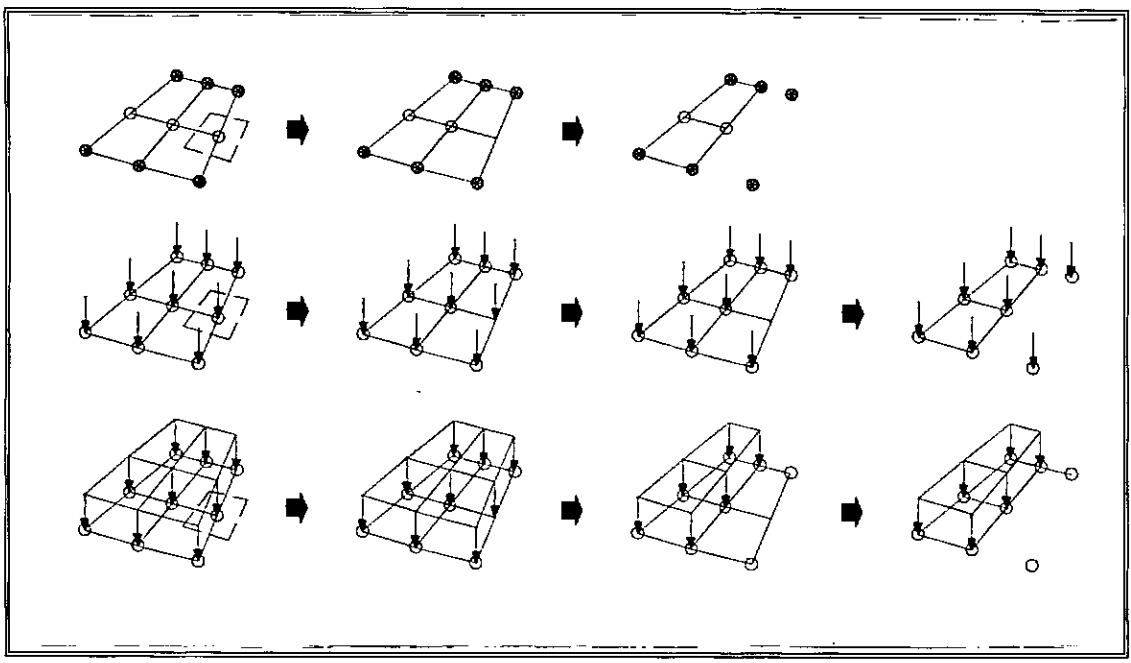


Fig V.94

A figura V.95 apresenta a sequência de retirada de elementos estruturais associados:

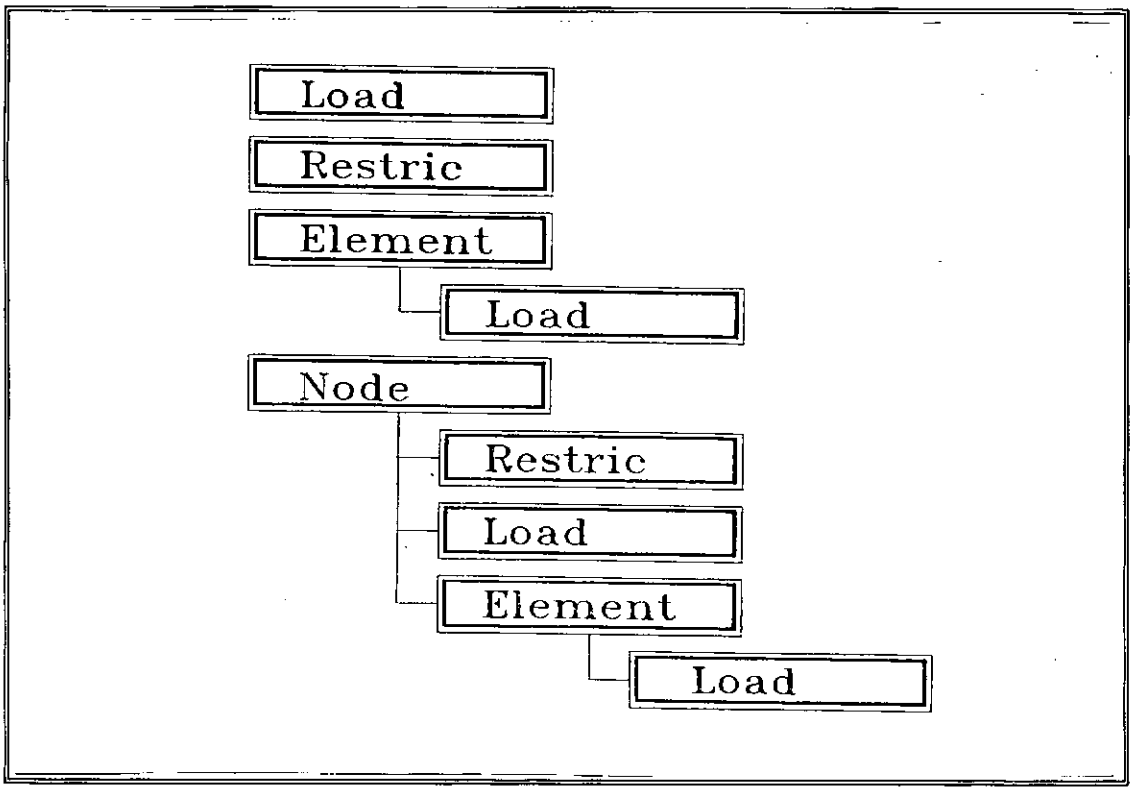


Fig V.95

O número de objetos selecionados não necessariamente corresponde à soma do número de nós, elementos, restrições e carregamentos, tendo em vista que o comando tem a capacidade de tratar também entidades puramente gráficas, sem significado estrutural.

V.4.8.5. O COMANDO COPY

O comando **COPY**, como o próprio nome já diz, permite ao usuário realizar operações de cópia sobre entidades gráficas.

Este comando é basicamente o mesmo comando ***copy*** padrão do **AutoCAD**, com algumas particularidades para o tratamento de entidades gráficas de significado estrutural.

O sistema se encarrega de apresentar em coloração intensa (*highlight*) as entidades gráficas selecionadas para a operação.

Para a seleção dos elementos, o usuário tem ao seu dispor todos os métodos padrões (*selection set*) do **AutoCAD**.

Para entidades gráficas sem significado estrutural, o comando realiza as operações, exatamente igual ao comando ***copy*** original do **AutoCAD**.

Para entidades gráficas com significado estrutural, o sistema apresenta um tratamento diferenciado. Dependendo do tipo de elemento estrutural, o processo de cópia pode desencadear a cópia de um outro tipo de elemento estrutural. A cópia de carregamentos desencadeia, necessariamente, a cópia dos pontos nodais ou elementos finitos aos quais estão associados. A cópia de elementos finitos faz com que o sistema automaticamente realize a cópia dos pontos nodais que formam a sua topologia.

A figura V.96 apresenta a sequência de cópia de elementos estruturais associados.

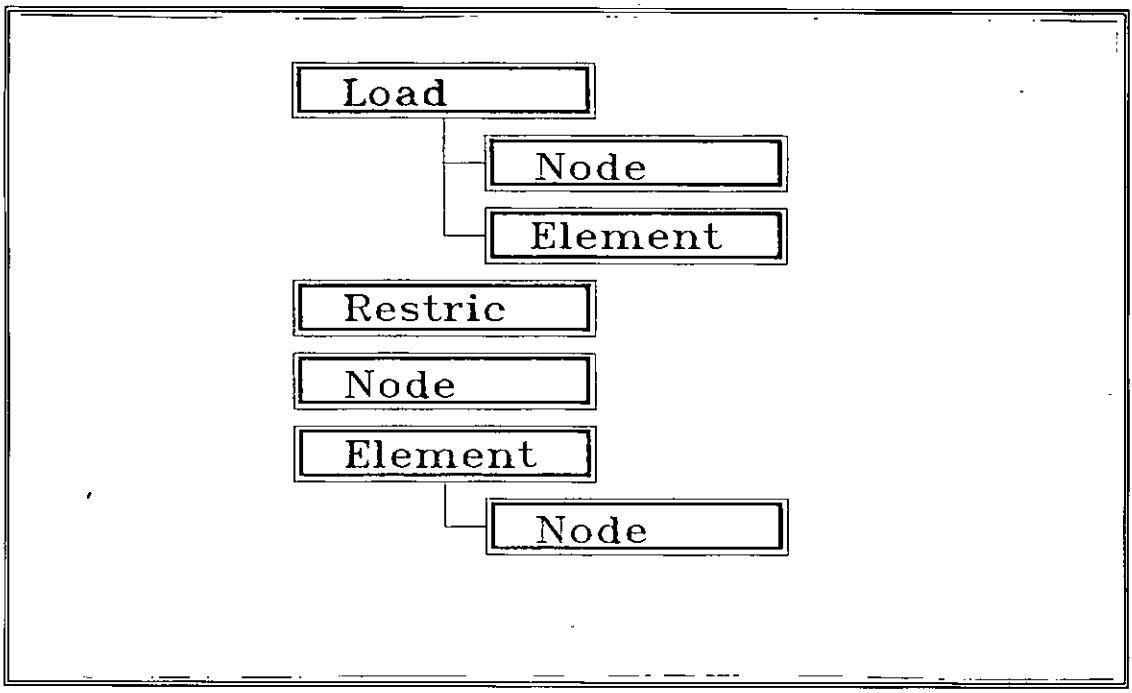


Fig V.96

Abaixo apresenta–se um exemplo de sequência de operações do comando **COPY**:

```
Command> Copy
Select object: Crossing
First point: 25.5
Second point: 35.30
 6 object selected
 3 Nodes
 2 Elements
 1 Restric
 1 Load
Select object: enter
<Base point or displacement>/Multiple:
Second point: P2
Command>
```

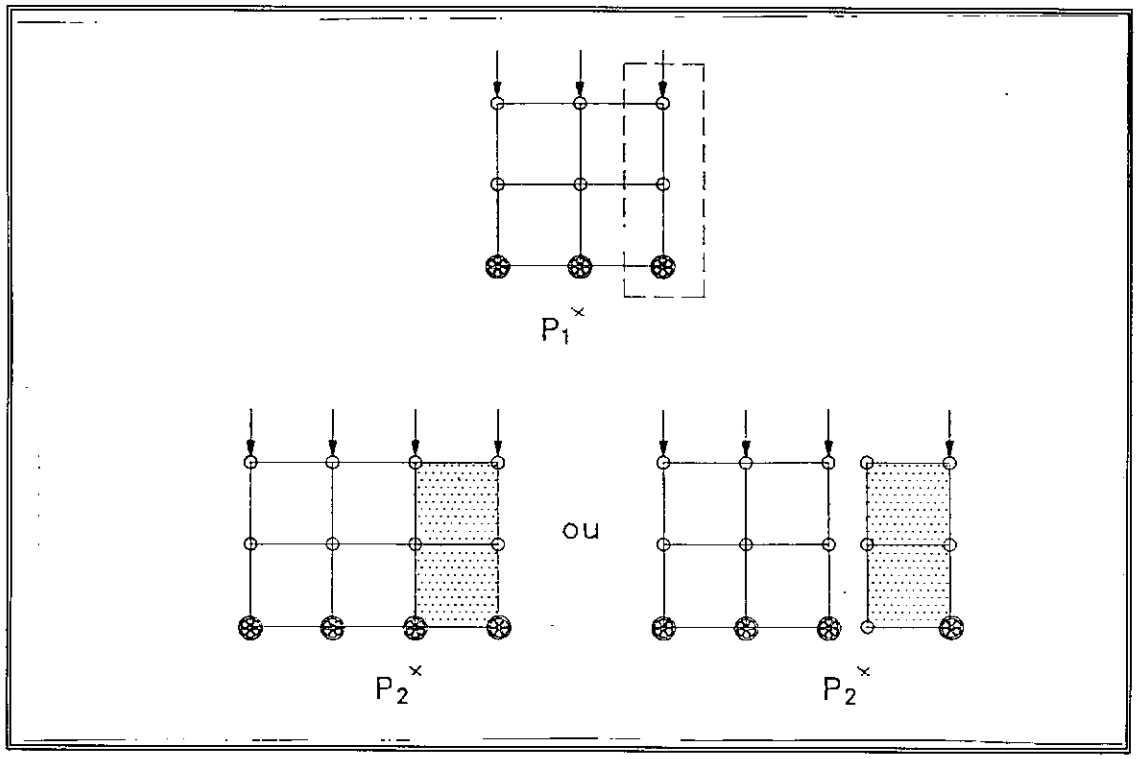


Fig V.97

Analisando a figura V.97, podemos verificar que para a segunda opção de posicionamento do ponto *P2*, são inseridos mais três pontos nodais no modelo.

A inserção ou não de novos pontos nodais no modelo é controlada pelo sistema, através da convergência automática de nós coincidentes para um único ponto nodal. A determinação da coincidência entre pontos nodais é feita, levando em consideração a escala ativa para inserção de pontos nodais como valor de tolerância. A escala ativa para inserção de pontos nodais é definida através do comando **NODE** opção **SCALE** apresentado anteriormente.

Os dados **BASE POINT** e **SECOND POINT**, requisitados ao usuário, correspondem ao ponto tomado como base para a cópia e sua nova posição respectivamente. As novas entidades gráficas são inseridas no modelo relativamente a estes pontos.

Se desejar, o usuário poderá realizar cópias sucessivas com o mesmo comando, atraves da opção **MULTIPLE**, bastando apenas indicar sucessivos **SECOND POINT**.

Estes dados são os mesmos utilizados no comando ***copy*** original do **AutoCAD**.

V.4.8.8. O COMANDO SCALE

O comando SCALE permite ao usuário aplicar operações de mudança de escala em entidades gráficas.

Este comando substitui e amplia o tipo de operação realizada pelo comando ***scale*** original do **AutoCAD**.

Além de permitir a aplicação de operações de mudança de escala diferentes para as direções X, Y e Z , algumas particularidades foram incluídas para o tratamento de entidades gráficas de significado estrutural.

O sistema se encarrega de apresentar em coloração intensa (*highlight*) as entidades gráficas selecionadas para a operação.

Para a seleção dos elementos, o usuário tem ao seu dispor todos os métodos padrões (*selection set*) do **AutoCAD**.

Para entidades gráficas sem significado estrutural, com fator de escala igual nas direções X, Y e Z, o comando realiza as operações exatamente igual ao comando ***scale*** original do **AutoCAD**. O comando permite, porém, a utilização de fatores de escala diferentes para cada direção. Neste caso, além da alteração do tamanho da entidade gráfica, esta terá seu formato distorcido.

Abaixo apresenta–se um exemplo de aplicação do comando **SCALE** para um círculo utilizando fatores de escala diferentes para X e Y:

```
Command> Scale
Select object: Last
Base point: 0.0
X scale factor<1>/Corner/XYZ: 2
Y scale factor:<2> 1
Command>
```

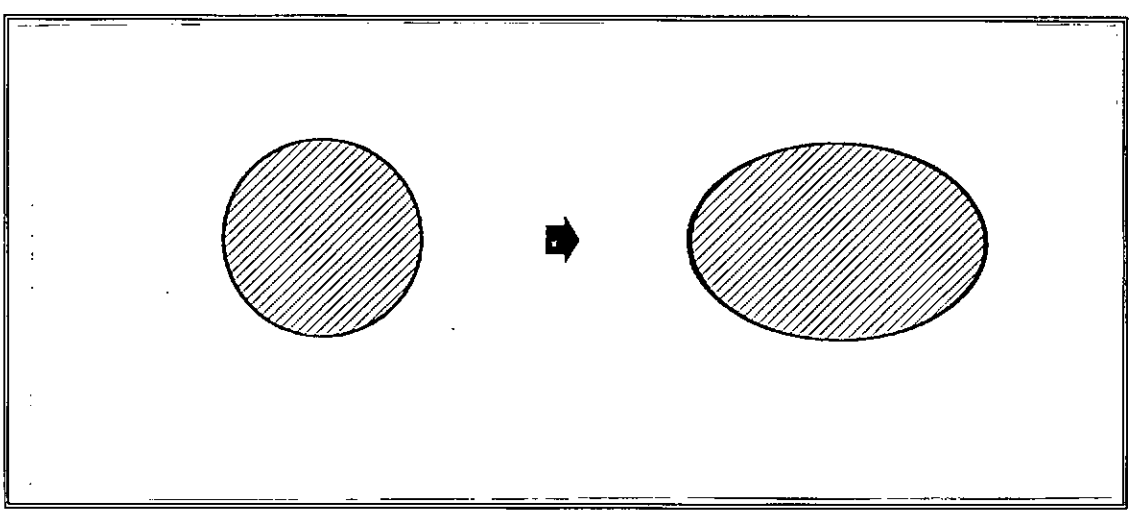


Fig V.98

O valor *default* adotado para a escala na direção X é sempre igual a 1 (um), e para a escala na direção Y, o mesmo indicado pelo usuário para a direção X.

– A opção **CORNER**, permite ao usuário definir um ponto, que em conjunto com o **BASE POINT** determinam as escalas em X e Y simultaneamente.

– A opção **XYZ**, permite ao usuário atribuir valores de escala para todas as direções X, Y e Z. Caso contrário, nenhuma alteração da escala na direção Z é aplicada ao grupo de entidades selecionadas.

Para entidades gráficas de significado estrutural, o sistema apresenta um tratamento diferenciado. Dependendo do tipo de elemento estrutural, o comando pode apresentar características do comando **MOVE** a nível de execução.

Para o caso de elementos finitos, a operação é exatamente igual a realizada para entidades gráficas sem significado estrutural.

Para os pontos nodais, restrições e carregamentos concentrados, aplicados tanto a elementos como a nós, o comando apresenta resultados semelhantes ao deslocamento destes elementos estruturais, correspondendo aos fatores de escala aplicados. Isto significa que, as dimensões dos símbolos que representarão pontos nodais, restrições e carregamentos, não serão alteradas com a aplicação do comando **SCALE**.

Para o caso de carregamentos distribuídos, do tipo **LINEAR** ou **SURFACE**, a alteração da forma segue a alteração da forma do elemento finito ao qual está associado. Nenhuma alteração na magnitude do carregamento é realizada com a aplicação do comando **SCALE**.

A seguir apresenta–se um exemplo de aplicação do comando **SCALE** a entidades gráficas com significado estrutural:

```
Command> Scale
Select object: W
First Point: 0.0
Second Point: 10.10
 23  object selected
 12  Nodes
 6  Elements
 4  Restric
 5  Loads
Select object: enter
Base point: 0,0,0
X scale factor <1>/Corner/XYZ: XYZ
X scale factor:<1> 2
Y scale factor:<2> 1
Z scale factor:<2> 1.5
Command>
```

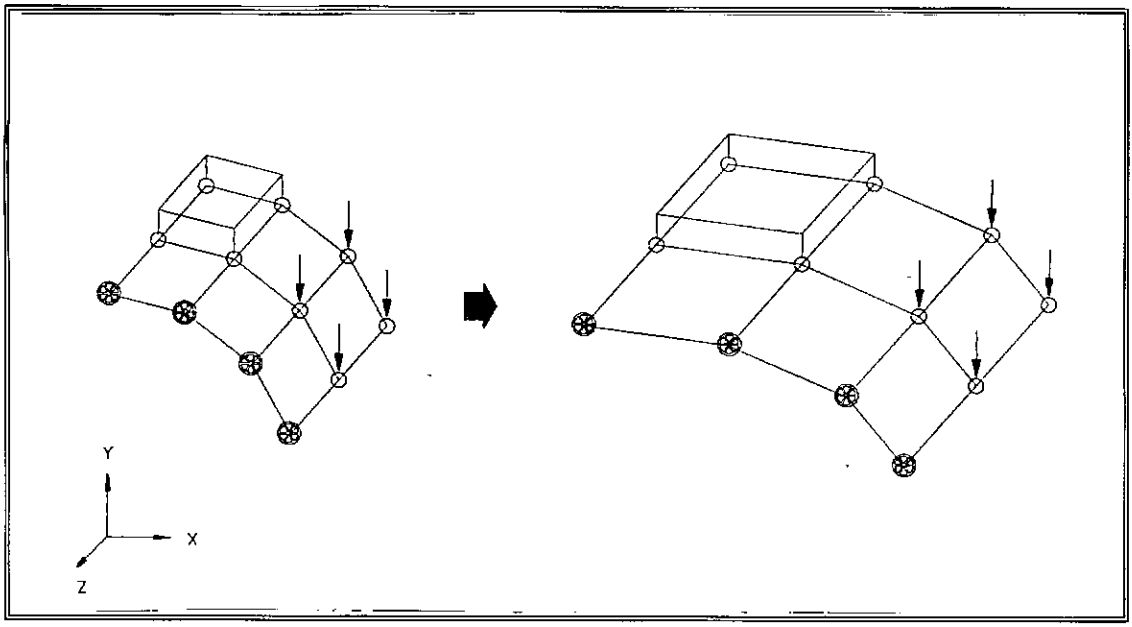


Fig V.99

Para o caso de operações de escala aplicadas a apenas alguns dos pontos nodais de um elemento, uma alteração no formato do elemento é realizada independentemente dos fatores de escala aplicados.

A figura V.100 corresponde ao exemplo apresentado na figura V.99, onde os elementos selecionados agora são os totalmente contidos na região tracejada.

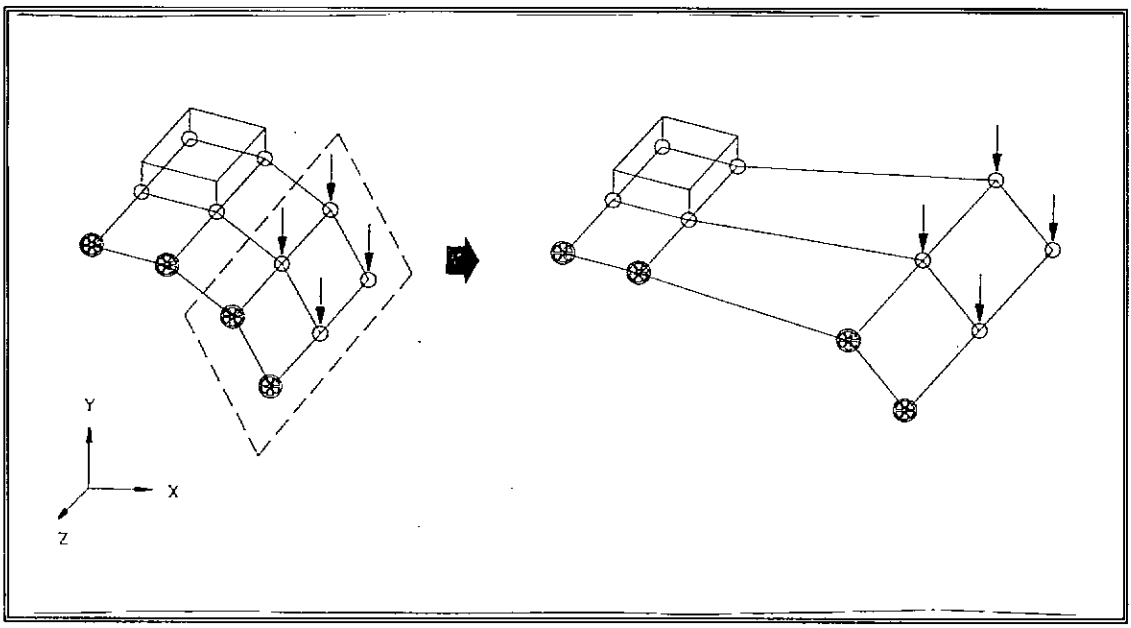


Fig V.100

Os carregamentos só sofrerão efeito do comando **SCALE**, se os pontos nodais ou elementos finitos aos quais estão associados, sofrerem operações de mudança de escala.

As operações do comando **SCALE**, devido a não utilização do comando original do **AutoCAD**, não permitem a execução com opção de visualização dinâmica (***drag***).

V.8.8.9. O COMANDO ROTATE

O comando **ROTATE** permite ao usuário aplicar operações de rotação a entidades gráficas.

Este comando é basicamente o mesmo comando ***rotate*** padrão do **AutoCAD**, com algumas particularidades para o tratamento de entidades gráficas com significado estrutural.

Para entidades gráficas sem significado estrutural, o comando realiza as operações exatamente igual ao comando ***rotate*** original do **AutoCAD**.

Para entidades gráficas com significado estrutural, o sistema apresenta um tratamento diferenciado. Dependendo do tipo de elemento estrutural, o processo de rotação pode resultar apenas em operações de deslocamento, ou alterar inclusive o formato das representações gráficas destes elementos.

Para pontos nodais, restrições e elementos finitos, o comando **ROTATE** apresenta os mesmos resultados do comando ***rotate*** original do **AutoCAD**.

Para carregamentos concentrados, aplicados tanto a elementos como a pontos nodais, o comando apresenta resultados semelhantes ao deslocamento destes elementos estruturais correspondentes à rotação aplicada. Isto significa que, os símbolos que representam a direção de aplicação de cargas concentradas, não sofrem rotação com a aplicação do comando **ROTATE**. O carregamento sofrerá apenas translação, acompanhando o ponto nodal ou elemento finito a que está associado.

Para carregamentos distribuídos (**LINEAR** ou **SURFACE**), uma alteração na representação gráfica pode ser verificada, de modo a manter a direção original de aplicação do carregamento.

A seguir apresenta-se um exemplo de aplicação do comando **ROTATE** a entidades gráficas com significado estrutural:

```
Command> Rotate
Select object: W
First Point: 0.0
Second Point: 10.10
 23  objects selected
 12  Nodes
 6  Elements
 4  Restric
 5  Loads
Select object: enter
Base point: 0,0
<Rotation angle>/Reference: -90
Command>
```

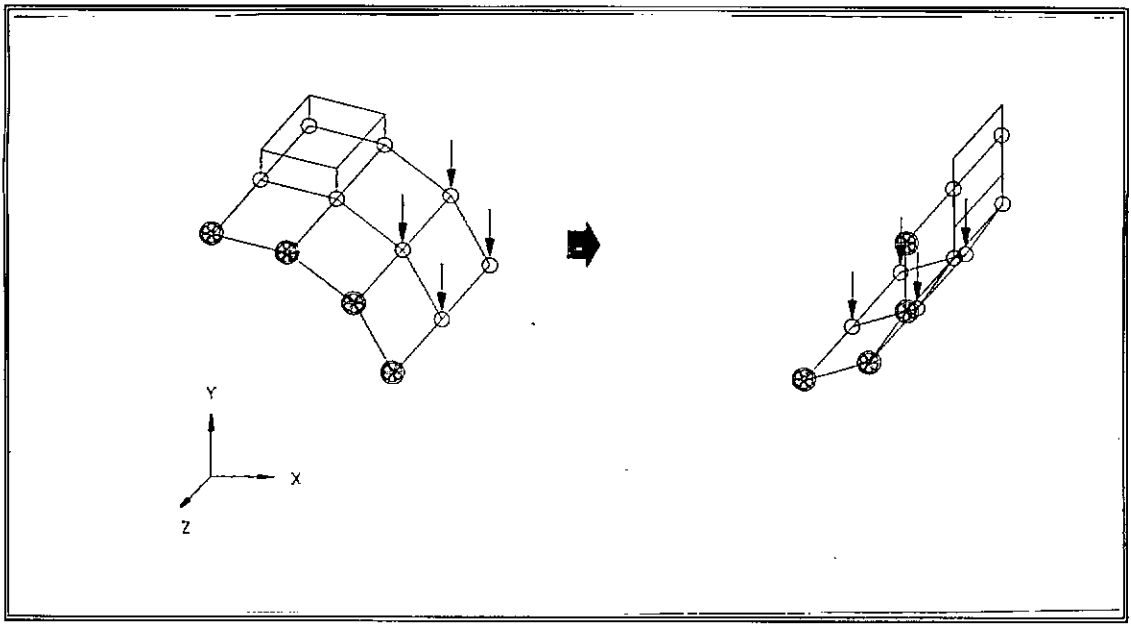


Fig V.101

Para o caso de operações de rotação aplicadas a apenas alguns dos pontos nodais de um elemento, uma alteração no formato do elemento é realizada.

A figura V.92 corresponde ao exemplo apresentado na figura V.101, onde os elementos selecionados agora, são os totalmente contidos na região tracejada.

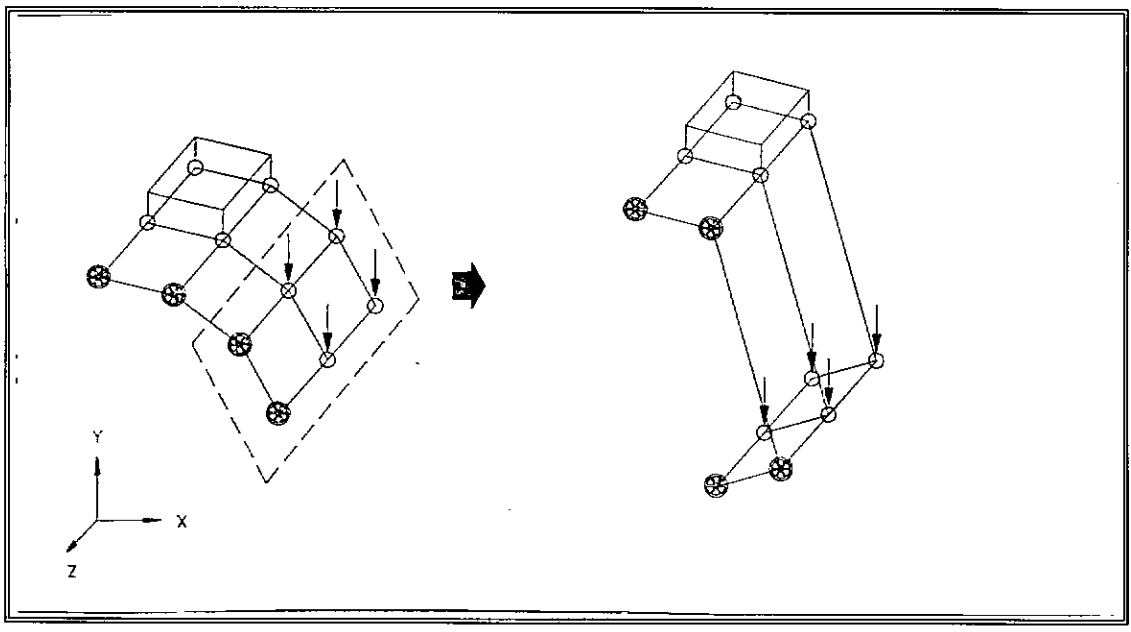


Fig V.92

V.8.8.10. O COMANDO MIRROR

O comando **MIRROR** permite ao usuário realizar operações de espelhamento de entidades gráficas.

Este comando é basicamente o mesmo comando ***mirror*** padrão do **AutoCAD**, com algumas particularidades para o tratamento de entidades com significado estrutural.

Para entidades gráficas sem significado estrutural, o comando realiza as operações exatamente igual ao comando ***mirror*** do **AutoCAD**.

Para entidades gráficas com significado estrutural, o sistema apresenta um tratamento diferenciado.

Tudo o que foi apresentado sobre o tratamento dos diversos tipos de elementos estruturais para o comando **ROTATE**, aplica–se ao comando **MIRROR**.

A seguir apresenta–se um exemplo de aplicação do comando **MIRROR** a entidades gráficas com significado estrutural.

```
Command> Mirror
Select object: W
First point: 0.0
Second point:10.10
 23  objects selected
 12  Nodes
 6  Elements
 4  Restric
 5  Loads
Select object: enter
First point or Mirror line:P1
Second point: P2
Delete old objects?<N> N
Command>
```

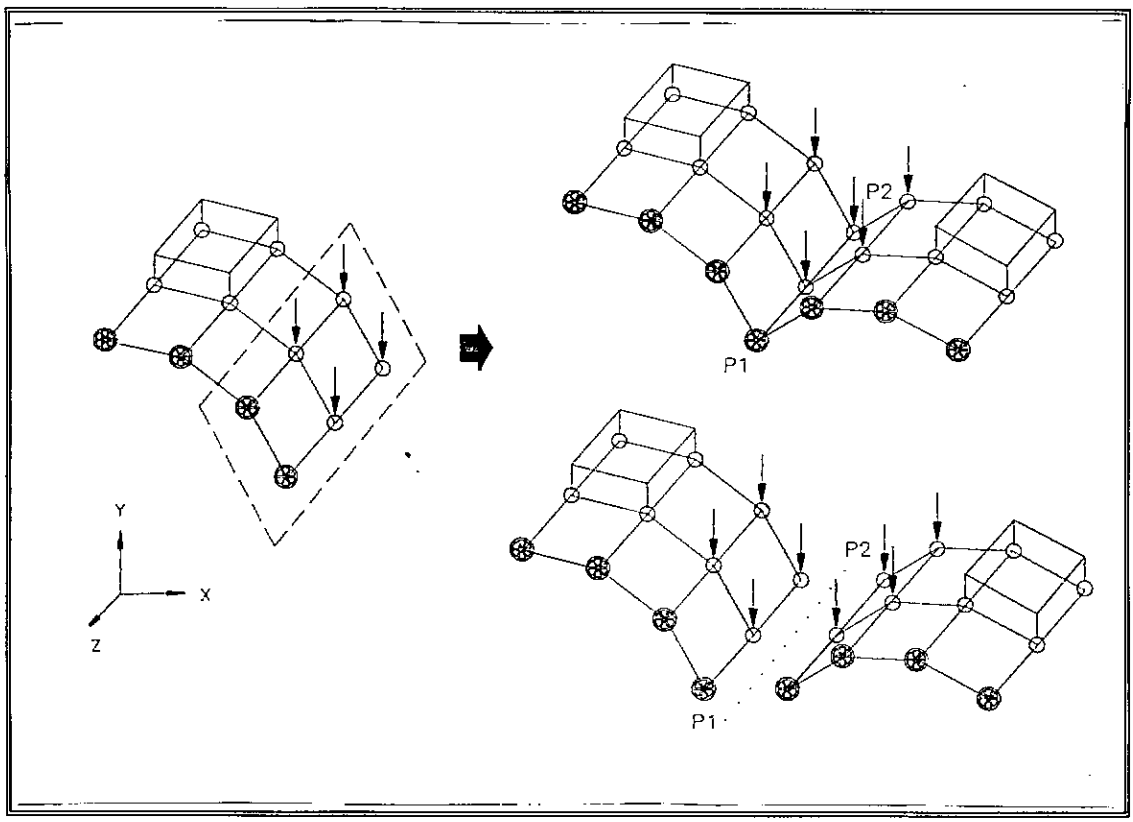


Fig V.103

A inserção ou não de novos pontos nodais no modelo é controlada pelo sistema, através da convergência automática de nós coincidentes para um único ponto nodal. Isto não é válido para o caso de carregamentos, o que significa que o carregamento aplicado nos nós na linha de espelhamento, para o primeiro caso da figura V.103, apresenta-se duplicado. Este tipo de problema é de dificil detecção visual e não incorre em inconsistência, devendo então o usuário estar atento para evitar que resultados indesejados venham a ser obtidos.

A determinação da coincidência entre pontos nodais é feita levando em consideração a escala ativa para a inserção de pontos nodais como valor de tolerância. A escala ativa para inserção de pontos nodais é definida através do comando **NODE** opção **SCALE** apresentado anteriormente.

Deve–se frisar que, o espelhamento de elementos finitos provoca a inversão no sentido da incidência nodal. Isto não chega a causar problemas, pois o sistema **AutoMESH** se encarrega de homogeneizar os sentidos das incidências nodais dos diversos elementos finitos, quando da geração do arquivo de dados do modelo.

V.8.8.11. O COMANDO ARRAY

O comando **ARRAY** permite ao usuário realizar múltiplas cópias de entidades gráficas segundo um campo retangular ou circular.
Este comando é basicamente o mesmo comando ***array*** padrão do **AutoCAD**, com algumas particularidades para o tratamento de entidades com significado estrutural.

Para entidades gráficas sem significado estrutural, o comando realiza as operações exatamente igual ao comando ARRAY original do **AutoCAD**.

Para entidades gráficas com significado estrutural, o sistema apresenta um tratamento diferenciado.

Tudo o que foi apresentado sobre tratamento dos diversos tipos de elementos estruturais para os comandos **ROTATE** e **COPY**, aplicam–se ao comando **ARRAY**.

A seguir apresenta–se um exemplo de aplicação do comando **ARRAY** com opção **RETANGULAR** a entidades gráficas com significado estrutural:

```
Command> Array
Select object: W
First point: 0.0
Second point:10.10
 26  Objects selected
 16  Nodes
 8  Elements
Select object: enter
Rectangular or Polar array(R/P): R
Number of columns(||): 3
Number of rows(——): 2
Distance between columns: 10
Distance between rows: 10
Command>
```

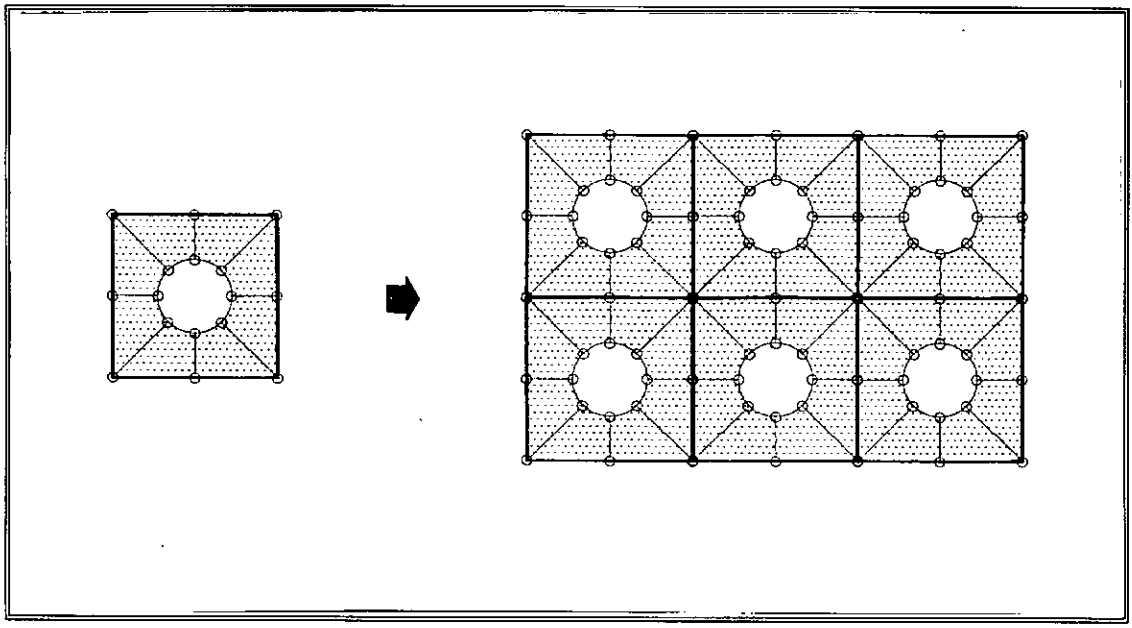


Fig V.104

O usuário poderá definir simultaneamente a distância entre colunas e linhas, indicando um ponto ao invés de um valor, quando requisitada a distância entre colunas. Para este caso, o usuário é requisitado a definir um segundo ponto que, em conjunto com o primeiro, possibilitam ao sistema determinar as distâncias entre linhas e colunas.

Abaixo apresenta–se um exemplo de aplicação do comando **ARRAY** com opção **POLAR**:

```
Command> Array
Select object: W
First point: 0.0
Second point: 10.10
 23  objects selected
 12  Nodes
 6  Elements
 4  Restric
 3  Loads
Select object: enter
Rectangular or Polar array(R/P): P
Center point: P1
Number of items: 4
Angle to fill:<360> 360
Rotate object?<N> Y
Command>
```

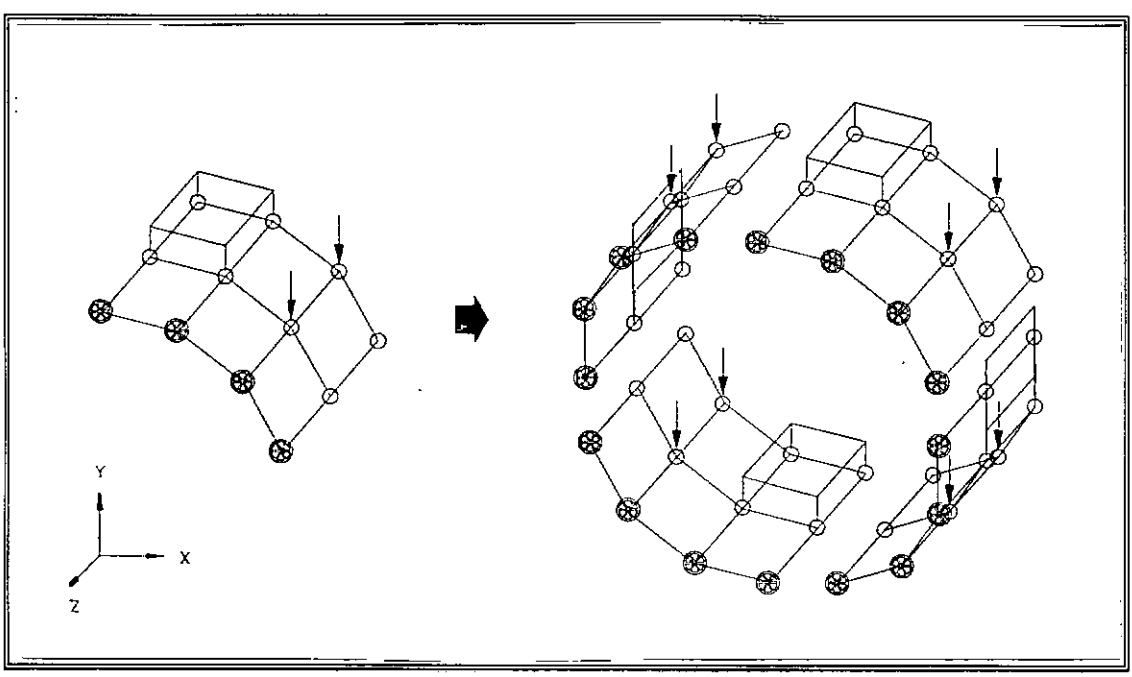


Fig V.105.a

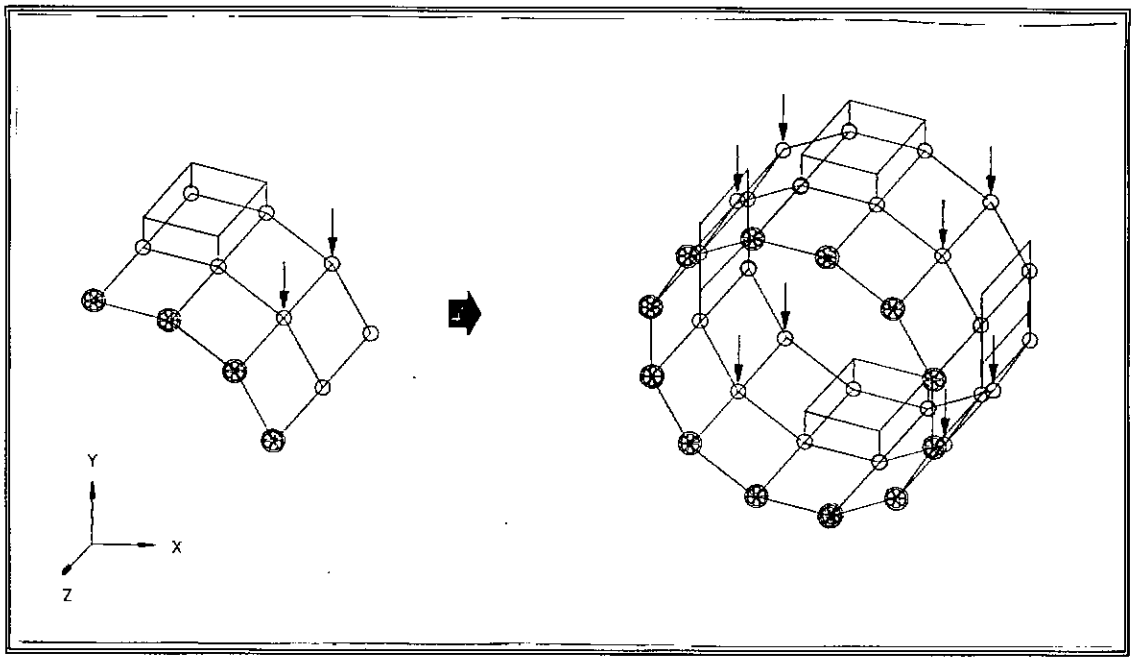


Fig V.105.b

Na figura V.105 são apresentados dois resultados de execução, um para o centro do array polar coincidindo com o centro da casca de cilindro discretizada e outro não. Para o primeiro caso, o sistema fará automaticamente com que os nós coincidentes convirjam para um único ponto. Em qualquer dos casos, o carregamento manterá a direção de aplicação de carga original.

V.4.9. COMANDOS UTILITÁRIOS

Os comandos utilitários são aqueles que possibilitam ao usuário um melhor entendimento ou verificação do modelo gerado.

Dentre estes comandos, podemos destacar três grupos distintos:

GRUPO	COMANDO
View	Find Numbering Shrink Face Project Releases Display
Tools	Intert Convert Clear DataBase
Post Processing	Post

Tab V.5

O grupo VIEW engloba os comandos responsáveis por facilitar a visualização ou verificação visual do modelo.

O grupo **TOOLS** engloba os comandos de auxílio à geração ou verificação do modelo.

O grupo **POST–PROCESSING** permite ao usuário realizar operações de pós–processamento dos resultados de análise do modelo gerado.

V.4.9.1. O COMANDO FIND

O comando **FIND** permite ao usuário localizar pontos nodais, elementos finitos ou materiais do modelo.

O comando **FIND** apresenta três opções de execução:

```
Material/Element/<Node:
```

– A opção **MATERIAL** permite ao usuário localizar todos os elementos finitos associados a um tipo de material definido pelo usuário.

O usuário é requisitado a especificar o nome do material. O sistema então, apresenta em coloração intensa (*highlight*) todos os elementos que possuem material igual ao especificado.

– A opção **ELEMENT** permite ao usuário localizar um determinado elemento finito no modelo, através da especificação de sua identificação numérica. O sistema apresenta o elemento finito em coloração intensa (*highlight*).

– A opção **NODE** é adotada como *default* para o comando **FIND** e permite ao usuário localizar um determinado ponto nodal no modelo.
O usuário é requisitado a especificar a identificação numérica do ponto nodal. O sistema localiza o dado ponto nodal e o apresenta ao usuário em coloração intensa (*highlight*).

Abaixo apresenta–se um exemplo da sequência de operações para o comando **FIND** na localização de um dado elemento finito:

```
Command> FIND
Material/Element/<Node>: E
Element ID: 10
Element ID: 3
Element ID: enter
Command>
```

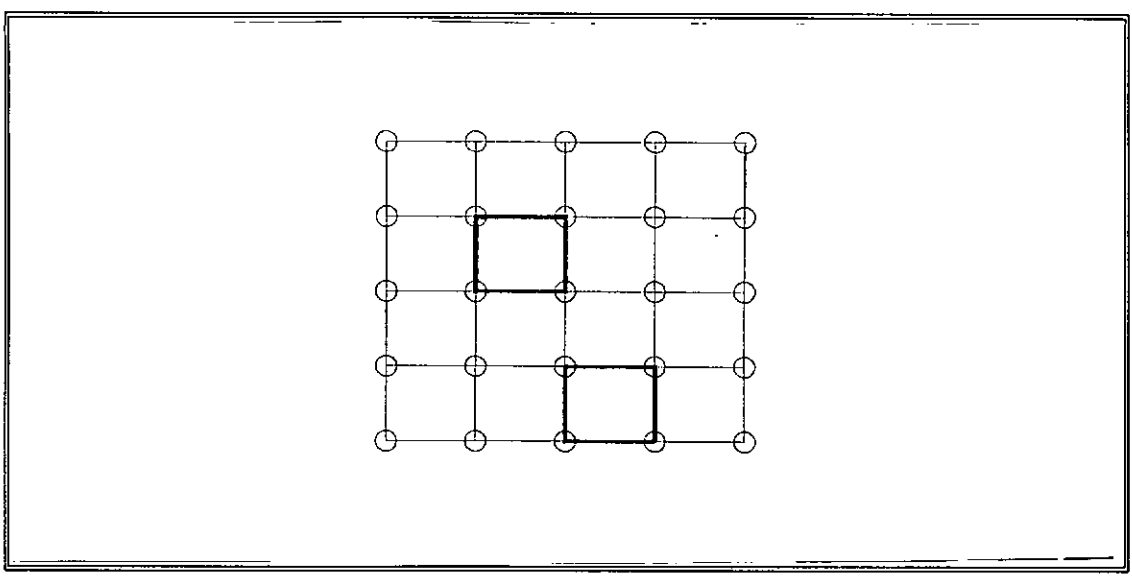


Fig V.106

Os elementos selecionados permanecem em coloração intensa até que o comando seja concluído.

V.4.9.2. O COMANDO NUMBERING

O comando **NUMBERING** permite ao usuário incluir no desenho a numeração nodal e/ou de elementos.

A identificação nodal e de elementos finitos é representada por textos sem nenhum significado estrutural, nem estão associados a nenhum elemento estrutural. Isto significa que, operações de edição sobre nós ou elementos finitos não acarretam qualquer efeito associado aos identificadores.

O comando **NUMBERING** possui três opções de execução:

```
Clear/Node/<Element>:
```

Os textos referentes à identificação de pontos nodais e elementos finitos ficam residentes nos ***layers***, NODEID e ELEMID respectivamente.

As dimensões dos textos são determinadas pelo valor referente à escala ativa de inserção de símbolos nodais.

Os pontos de inserção dos textos de identificação numérica correspondem ao ponto de inserção dos símbolos nodais para nós e ao ponto médio dos pontos nodais para elementos.

– A opção **CLEAR** permite ao usuário remover do modelo todos os textos que representam a numeração nodal ou de elementos.

O usuário é requisitado a determinar qual a numeração que deseja remover do modelo.

`NodeID/ElemID/<All>:`

Tendo em vista que os textos referentes à identificação numérica de nós e elementos finitos residem nos ***layers*** NODEID e ELEMID respectivamente, o sistema realiza através da opção **CLEAR** do comando **NUMBERING**, a retirada de todos os textos associados a estes ***layers***.

Recomenda–se que não sejam mantidos desnecessariamente os identificadores numéricos de nós ou elementos, devido ao fato de provocar um crescimento muito acentuado do arquivo de dados gráfico. Isto provoca uma queda na performance do sistema nas operações de atualização do desenho na tela, devido ao maior número de entidades a regenerar e um maior número de troca de dados (*overlay*) entre memória e disco.

Outro detalhe a considerar, é o fato do sistema não garantir a manutenção da numeração de nós e elementos após operações de edição. Isto significa que, a numeração dos pontos nodais e elementos finitos, pode variar durante o processo de geração da malha do modelo.

A figura V.107 apresenta um exemplo de malha de elementos finitos com numeração de nós e elementos, gerada através do comando **NUMBERING**.

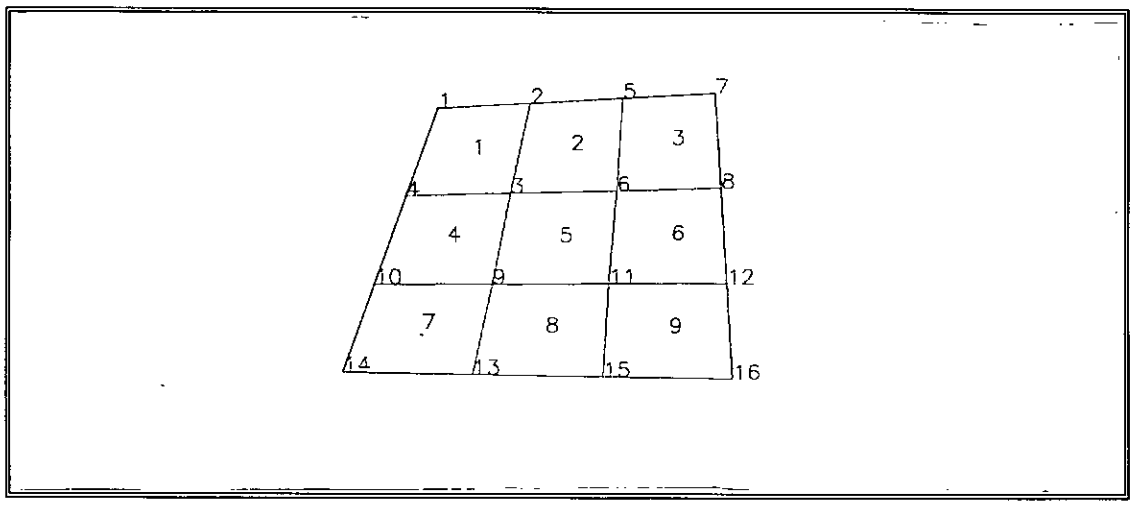


Fig V.107

A numeração segue a ordem de inserção no modelo e naturalmente a ordem de armazenamento no arquivo de dados gráfico.

V.4.9.3. O COMANDO SHRINK

O comando **SHRINK** permite ao usuário visualizar um elemento ou grupo de elementos com suas dimensões graficamente modificadas por um dado fator.

A seguir apresenta–se um exemplo de execução do comando **SHRINK** com um fator de encolhimento de 75 por cento.

```
Command> Shrink
Select elements: W
First point: 0.0
Second point: 10.10
 12 selected
Select element: enter
Shrink factor: 0.75
Command>
```

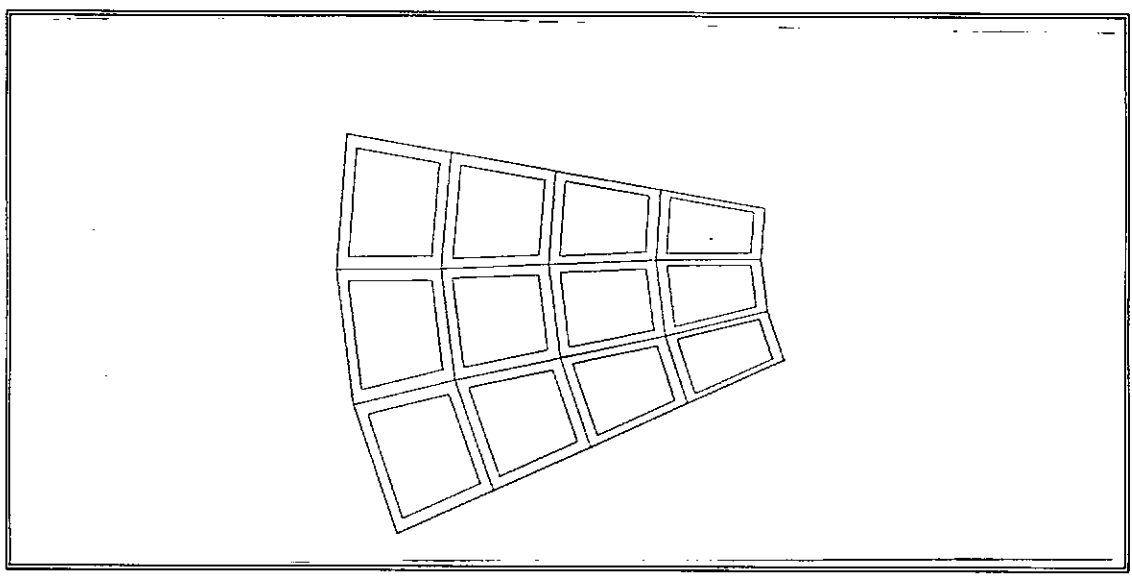


Fig V.108

O fator de encolhimento poderá ser qualquer valor não nulo e representa o fator de escala aplicado ao elemento em relação ao seu centróide.

Os elementos encolhidos são inseridos no desenho no ***layer*** ativo no momento da execução do comando **SHRINK** e são representados por ***polylines***.

Para seleção dos elementos finitos, o usuário tem ao seu dispor todos os métodos padrões (*selection set*) do **AutoCAD**.

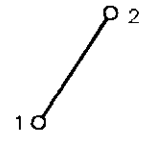
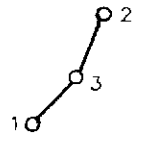
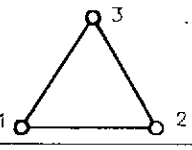
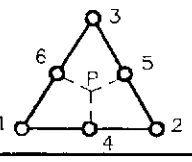
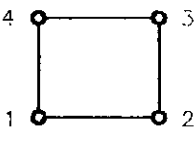
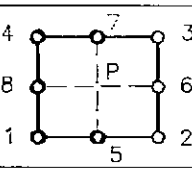
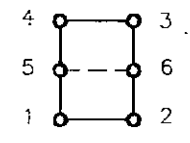
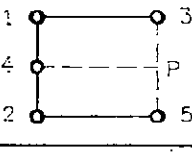
V.4.9.4. O COMANDO FACE

O comando **FACE** tem como função, construir elementos de superfície por sobre os elementos finitos do modelo, para possibilitar a utilização do comando ***hide*** do **AutoCAD**.

Devido ao fato da maioria dos elementos finitos adotados no sistema **AutoMESH** terem sua representação gráfica feita através de entidades gráficas do tipo ***polyline***, em modelos tridimensionais, como no caso de cascas ou placas, o uso do comando ***hide*** do **AutoCAD** para remoção de linhas ocultas torna–se ineficiente, pois os elementos não possuem características de superfície e, portanto, não são capazes de ocultar nenhuma linha.

O comando **FACE** realiza o preenchimento dos elementos finitos, representados por ***polylines*** com entidades gráficas do tipo ***3dface***. Para cada tipo de elemento finito, um número de ***3dfaces*** é necessário, tendo em vista que estas entidades possuem no máximo 4 vértices.

A tabela V.6 apresenta o número de ***3dfaces*** necessários para cada tipo de elemento, bem como sua configuração.

	Elemento Finito	3DFACE	
		Quantidade	Incidencia
BAR		NENHUM	–
ISOBAR		NENHUM	–
TRI		UM	1 2 3 3
HEX		TRES	1 4 P 5 2 6 P 4 3 5 P 6
QUAD		UM	1 2 3 4
OCT		QUATRO	1 5 P 8 1 6 P 5 3 7 P 6 4 8 P 7
Contact		DOIS	1 2 6 5 3 4 5 6
Infinite		DOIS	2 5 P 4 1 4 P 3

Tab V.6

As arestas das ***3dfaces*** interiores ao elemento, ou seja, que não coincidem com os segmentos do ***polyline***, são construídos como invisíveis. Isto possibilita que o modelo conserve o mesmo aspecto após a execução do comando **FACE**.

Os ***3dfaces*** gerados pelo comando **FACE** são inseridos no modelo associados ao ***layer*** ativo no momento da execução do comando.

Estas entidades não possuem nenhum significado estrutural, podendo ser retirados do desenho a qualquer momento, sem ocasionar nenhuma perda ao modelo.

O usuário é requisitado pelo sistema a fim de selecionar o grupo de elementos finitos aos quais o comando **FACE** será aplicado. Para a seleção, o usuário tem ao seu dispor todos os métodos padrões (*selection set*) do AutoCAD.

A figura V.109 apresenta os resultados da execução do comando ***hide*** do **AutoCAD** para o mesmo grupo de elementos finitos, antes e depois da execução do comando **FACE**.

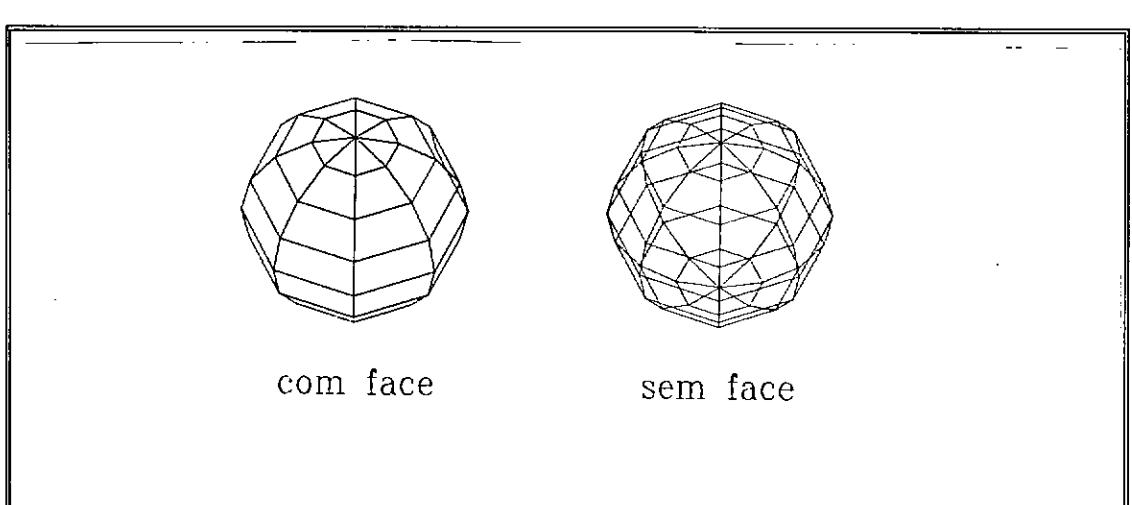


Fig V.109

Deve–se ressaltar que os elementos finitos utilizados na malha da figura V.109 são de segunda ordem.

A seguir apresenta–se a sequência de operações do comando **FACE** aplicado à malha de elementos finitos da figura V.109.

```
Command> Face
Select element: W
First point: 0.0
Second point: 10.10
 n elements selected
Select elements: enter
Command>
```

V.4.9.5. O COMANDO PROJECT

O comando **PROJECT** permite ao usuário visualizar o desenvolvimento dos perfis de elementos finitos do tipo reticulados.

Basicamente, este tipo de elemento finito é representado graficamente por meio de ***polylines*** abertos, de dois ou três vértices, que correspondem à linha de eixo do elemento.

O comando **PROJECT** constrói a representação gráfica para os elementos finitos do tipo BAR ou ISOBAR, considerando suas seções transversais.

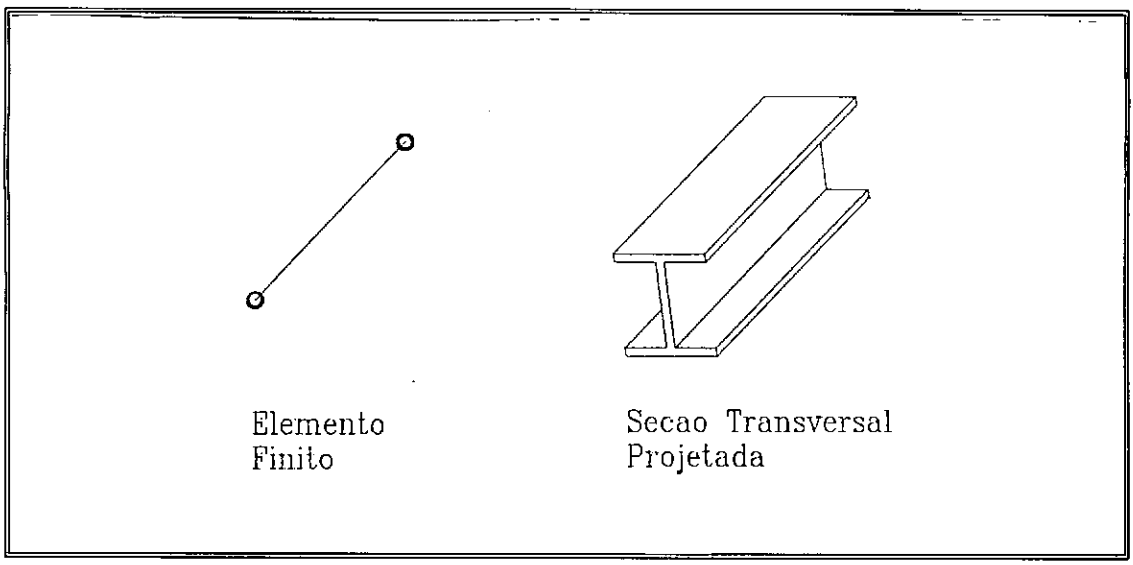


Fig V.110

Os dados referentes ao tipo de seção transversal, bem como suas dimensões, encontram–se registrados juntamente com o tipo de material associado ao elemento e são definidos através do comando **MATERIAL** apresentado anteriormente.

A seguir apresenta–se um exemplo de aplicação do comando **PROJECT** a um grupo de elementos finitos reticulados:

```
Command> Project
Select bars: W
First point: 0.0
Second point: 10.10
 5  Elements selected
Select elements: enter
Scale factor:<1> 1
Command>
```

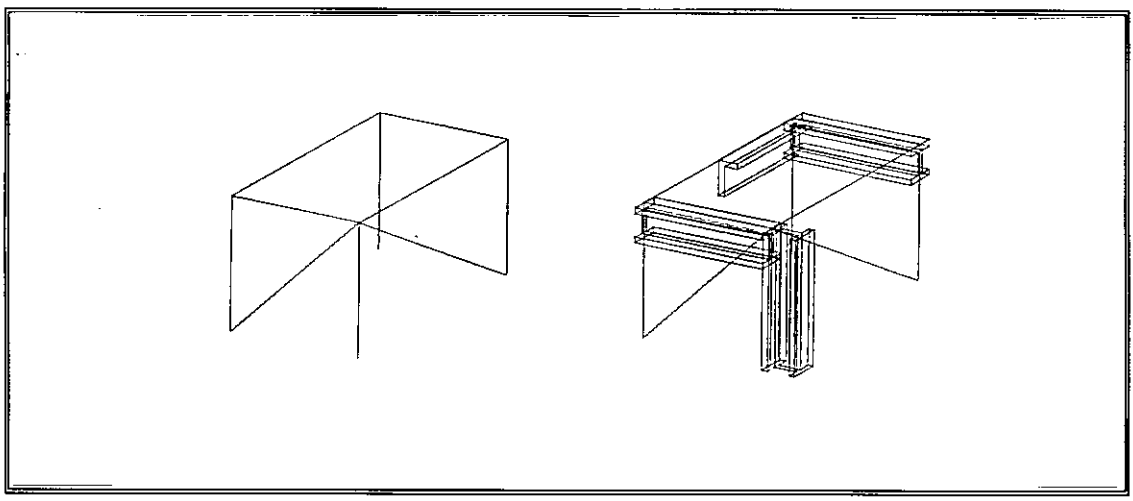


Fig V.111

A representação gráfica da seção transversal projetada, é inserida no desenho associada ao ***layer*** ativo no momento da execução do comando **PROJECT**.

Esta representação gráfica do elemento finito projetado é constituída de um ***polyline*** fechado, referente à seção transversal, extrudado na direção do eixo do elemento.

A figura V.112 apresenta o resultado da execução do comando ***hide*** do **AutoCAD** sobre um grupo de elementos com seção transversal projetada.

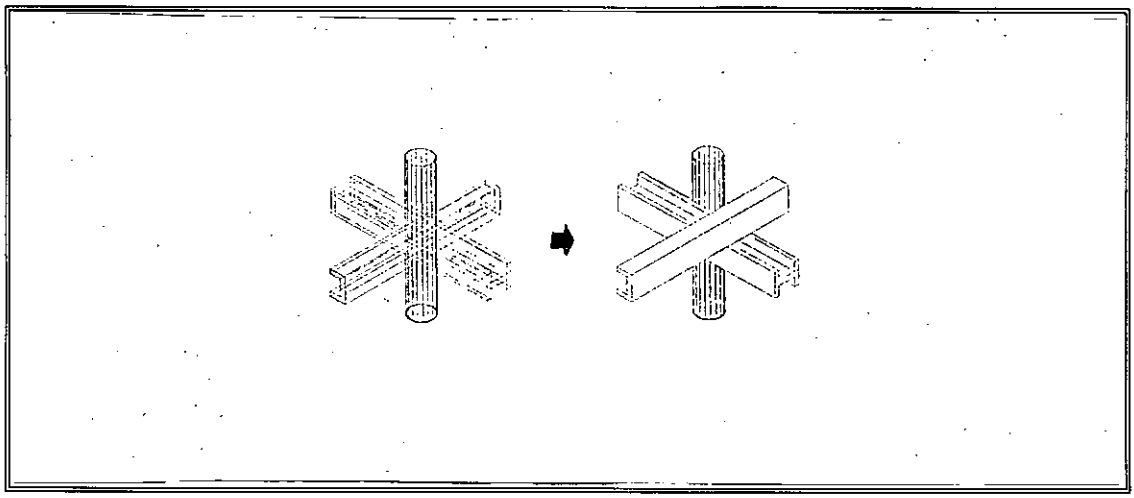


Fig V.112

Este comando é especificamente recomendado para verificação visual dos perfis utilizados, bem como a defasagem angular dos eixos locais da seção transversal, em relação aos eixos globais (ângulo beta).

Caso o usuário deseje, um fator de escala é aplicado à seção transversal do elemento com efeitos apenas visuais, nenhuma alteração é aplicada às dimensões reais do perfil.

O sistema adota como *default* um valor unitário para o fator de escala da seção transversal.

V.4.9.6. O COMANDO SECTIONS

O comando **SECTIONS** permite ao usuário visualizar o tipo de seção transversal ou alterar a posição relativa desta e o eixo do elemento finito do tipo reticulado.

Este comando possui três opções de execução:

```
<Display>/Rotate/Axi:
```

– A opção **DISPLAY** adotada como *default*, permite ao usuário visualizar a seção transversal dos elementos selecionados. O sistema construirá, em regime transitório, a representação da seção transversal em dimensões reais na extremidade do elemento finito reticulado correspondente a sua primeira incidência.

Por regime transitório, entenda–se a apresentação gráfica na tela, sem correspondente representação no arquivo de dados gráfico. Isto significa que, após o término da execução do comando **SECTIONS**, qualquer atualização da tela reestabelece o desenho ao estado de antes do comando.

Desenhos com apresentação gráfica em regime transitório, são obtidos pelo acesso direto ao dispositivo de tela através da função ***grdraw*** do AutoLISP.

Abaixo apresenta-se um exemplo de sequência de operações de comando **SECTIONS** com opção **DISPLAY**.

```
Command> Sections
<Display>/Rotate/Axi: Display
Select line elements: W
First point: 0.0
Second point: 10.10
Select line elements: enter
Command>
```

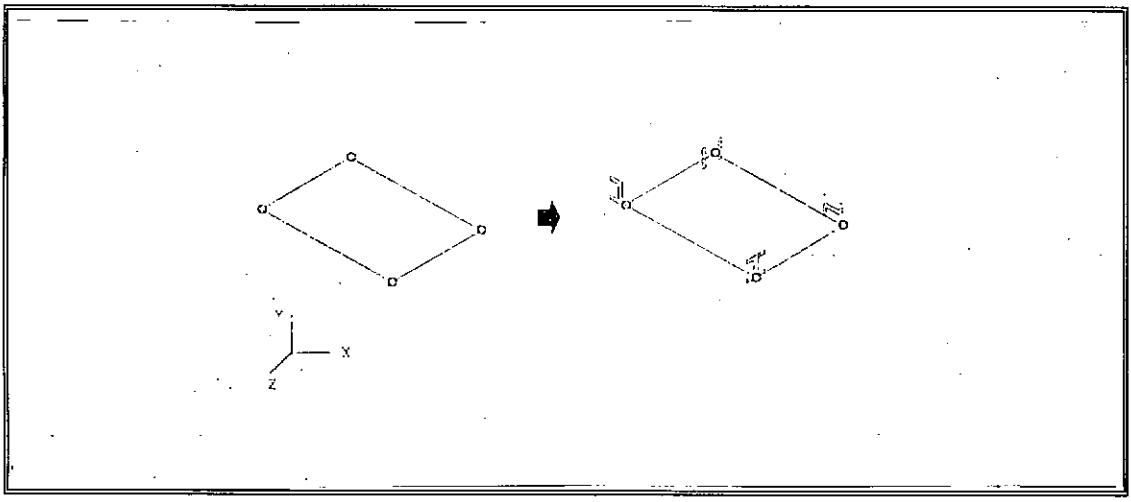


Fig V.113

– A opção **ROTATE** permite ao usuário alterar o ângulo beta de um elemento finito reticulado, aplicando uma rotação na seção transversal em relação ao eixo do elemento.

O usuário é requisitado a selecionar os elementos a serem operados. Para a seleção, o usuário tem ao seu dispor, todos os métodos padrões (*selection set*) do **AutoCAD**. O sistema apresenta então, a atual disposição das seções transversais dos elementos selecionados em regime transitório e requisita ao usuário a especificação de um ângulo de rotação.

A seguir apresenta–se um exemplo de sequência de operações para o comando **SECTIONS** opção **ROTATE**:

```
Command> Sections
<Display>/Rotate/Axi: Rotate
Select line elements: W
First point:0.0
Second point:10.10
Select line elements: enter
Roll angle:<0>
Command>
```

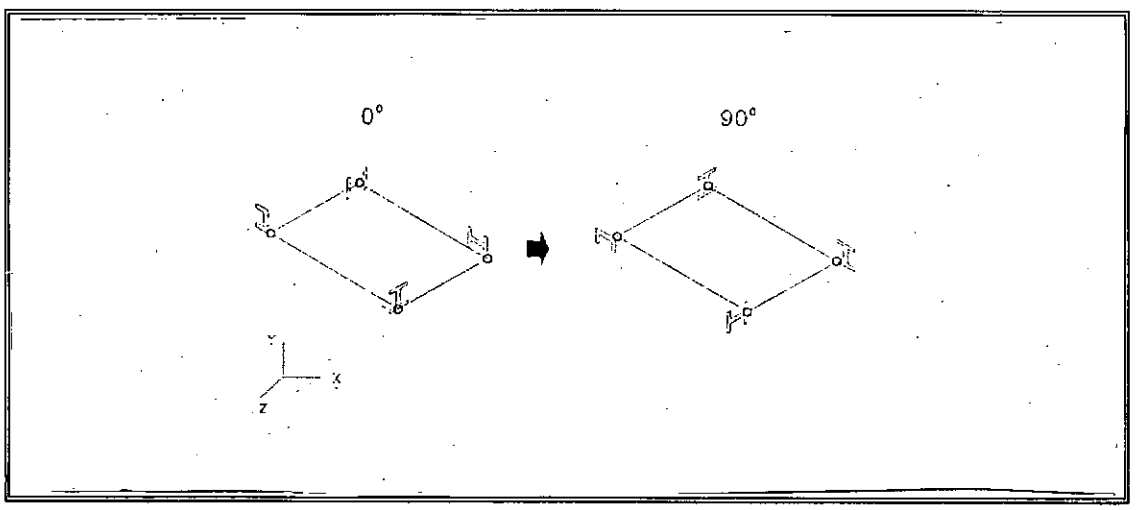


Fig V.114

– A opção **AXI** permite ao usuário alterar a posição da seção transversal dos elementos em relação ao seu eixo.

O sistema **AutoMESH** permite ao usuário selecionar entre nove posições relativas.

A figura V.115 apresenta as possíveis posições relativas do eixo para uma seção transversal do tipo I.

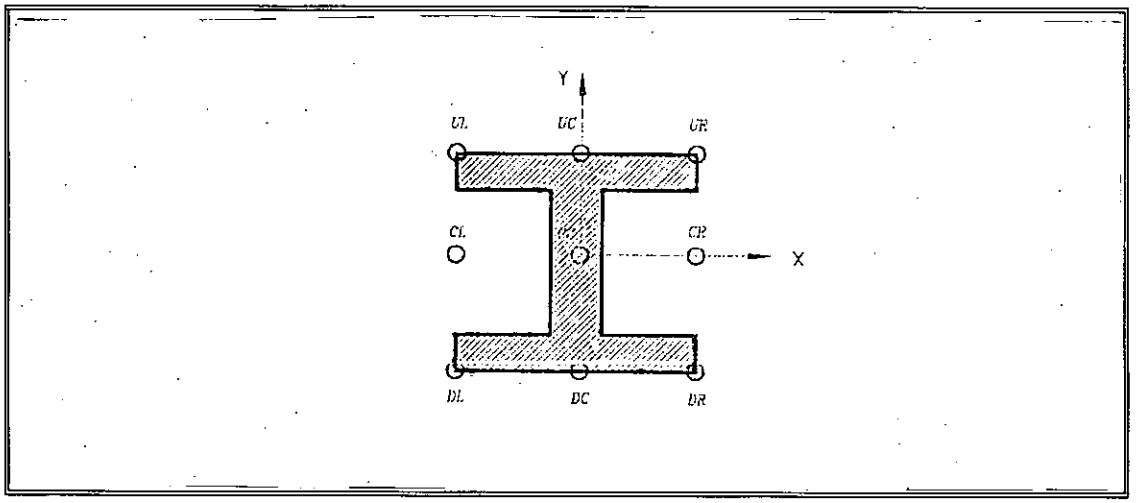


Fig V.115

Os pontos apresentados na figura V.115 são determinados pelo sistema, tendo em conta as dimensões máximas nas direções X e Y locais.

A tabela V.7 apresenta o significado das siglas adotadas para cada possível posição do eixo da seção transversal:

SIGLA	DESCRIÇÃO	
UL	Ponto Superior Esquerdo	(UPPER LEFT)
UC	Ponto Médio Superior	(UPPER CENTER)
UR	Ponto Superior Direito	(UPPER RIGHT)
CL	Ponto Médio Esquerdo	(CENTER LEFT)
CC	Ponto Central	(CENTER CENTER)
CR	Ponto Médio Direito	(CENTER RIGHT)
DL	Ponto Inferior Esquerdo	(DOWN LEFT)
DC	Ponto Médio Inferior	(DOWN CENTER)
DR	Ponto Inferior Direito	(DOWN RIGHT)

Tab V.7

A figura V.116 apresenta as possíveis posições dos eixos para os diversos tipos de seção transversal disponíveis no **AutoMESH**.

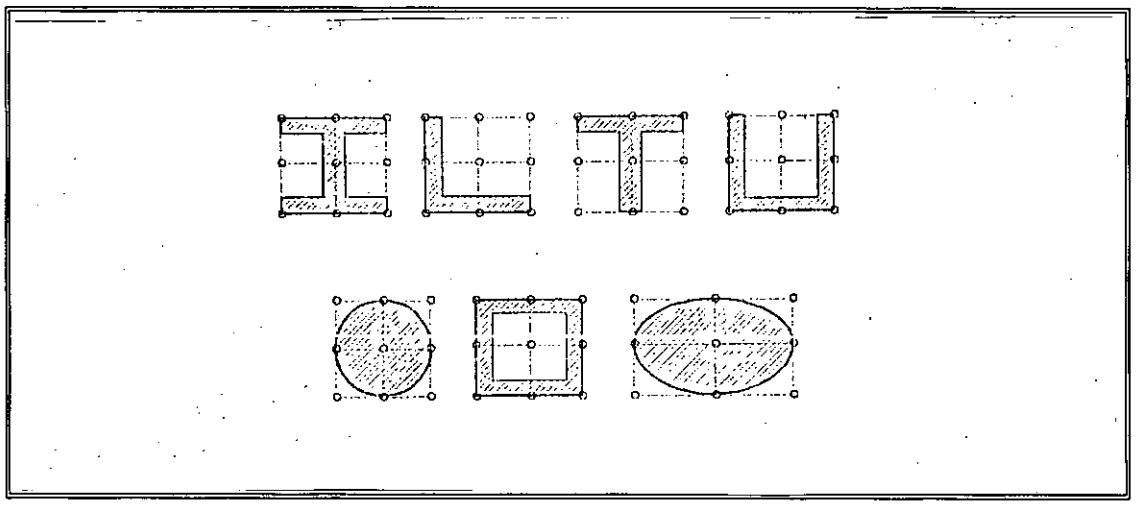


Fig V.116

Para a execução do comando, o usuário é requisitado a selecionar os elementos finitos do tipo reticulados a serem operados. O sistema apresenta então, a atual disposição das seções transversais dos elementos selecionados em regime transitório e requisita ao usuário a especificação da nova posição para a seção transversal relativamente ao eixo.

A seguir apresenta–se um exemplo de sequência de operações para o comando **SECTIONS** opção **AXI**:

```
Command> Sections
<Display>/Rotate/Axi: AXI
Select line elements: W
First point: 0.0
Second point: 10.10
Select line elements: enter
New axi point:UL/UC/UR/CL/<CC>/CR/DL/DC/DR: DC
Command>
```

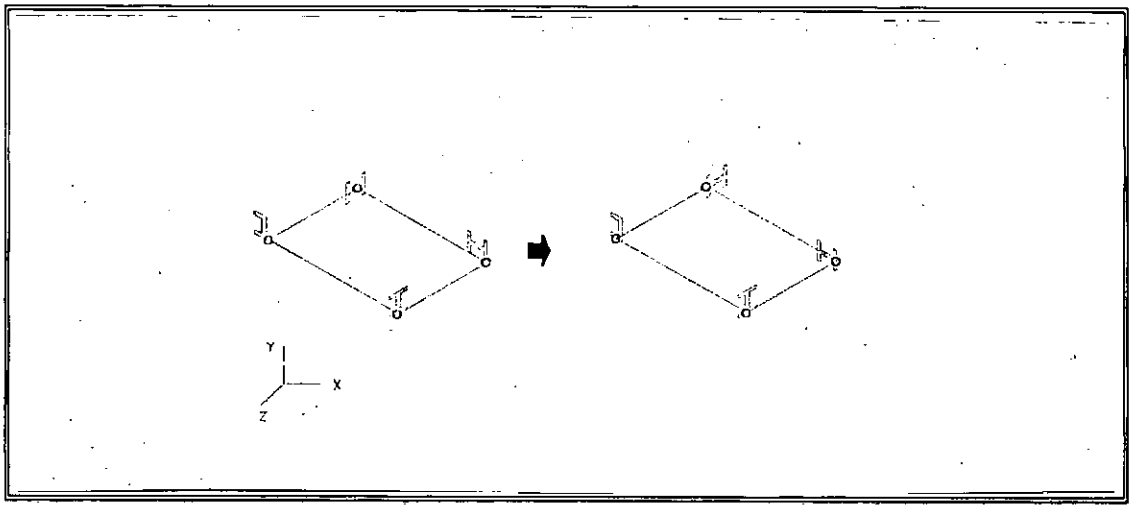


Fig V.117

V.4.9.7. O COMANDO DISPLAY

O comando **DISPLAY** tem como função definir a apresentação, visual ou não, de elementos estruturais.

Este comando corresponde a uma versão personalizada do comando ***layer*** do **AutoCAD** nas opções ***on*** e ***off*** com aplicação voltada a entidades gráficas com significado estrutural e serve como um utilitário de visualização ao usuário.

O comando **DISPLAY** tem como opções:

```
ON/OFF/Color:
```

As operações **ON**, **OFF** e **COLOR** correspondem às operações de ativação, desativação e troca da cor de determinado ***layer*** respectivamente.

Após selecionado o tipo de operação, o usuário é requisitado a especificar o tipo de elemento estrutural ao qual o comando será dirigido.

```
All/Node/Element/REGion/Loads/REStric/NodeID/ElemID:
```

As opções **NODE**, **REGION**, **LOADS**, **RESTRIC**, **NODEID** e **ELEMID** correspondem a aplicar o comando **DISPLAY** aos ***layers*** correspondentes a estas entidades gráficas.

Para a opção **ELEMENT**, o usuário é requisitado a especificar qual o tipo de material dos elementos finitos aos quais o comando será aplicado.

A opção **ALL** engloba todas as demais opções.

Para a execução do comando **DISPLAY** com opção **COLOR**, além de selecionar o tipo de elemento estrutural ao qual o comando será dirigido, o usuário é requisitado a especificar a nova cor a ser aplicada. O sistema apresenta como *default*, a cor atual para o selecionado tipo de elemento estrutural ou a mensagem VARIES se possuir mais de uma cor.

A seguir apresenta–se um exemplo de sequência de operações para o comando **DISPLAY**.

```
Command> Display
ON/OFF/Color: OFF
All/Node/Element/REGion/Loads/REStric/NodeID/
 ElemID: Region
ON/OFF/Color: ON
All/Node/Element/REGion/Loads/REStric/NodeID/
 ElemID: Element
Material name: *
ON/OFF/Color: Color
All/Node/Element/REGion/Loads/REStric/NodeID/
 ElemID: Node
New color:<varies> Yellow
ON/OFF/Color: Color
All/Node/Element/REGion/Loads/REStric/NodeID/
ElemID: Element
Material name: Steel
NEW Color:<Blue> Cyan
ON/OFF/Color: Color
All/Node/Element/REGion/Loads/REStric/NodeID/
 ElemID: Element
Material name: Aluminium
New color:<Blue> Magenta
ON/OFF/Color: enter
Command>
```

V.4.9.8. O COMANDO INTERT

O comando **INTERT** constitui–se em um comando utilitário para geração da linha de interseção entre elementos tubulares.

O usuário é requisitado a selecionar duas linhas quaisquer que correspondem aos eixos dos tubos e especificar os seus diâmetros. O sistema então, cria então, uma ***polyline*** fechada com quarenta vértices representando a *boca de lobo.*

A representação gráfica da *boca de lobo* é inserida no desenho associada ao ***layer*** ativo no momento da execução do comando **INTERT**.

Abaixo apresenta–se um exemplo de aplicação do comando **INTERT**:

```
Command> Intert
Select main tube:
Select secondary tube:
Main diameter:
Secondary diameter:
Command>
```

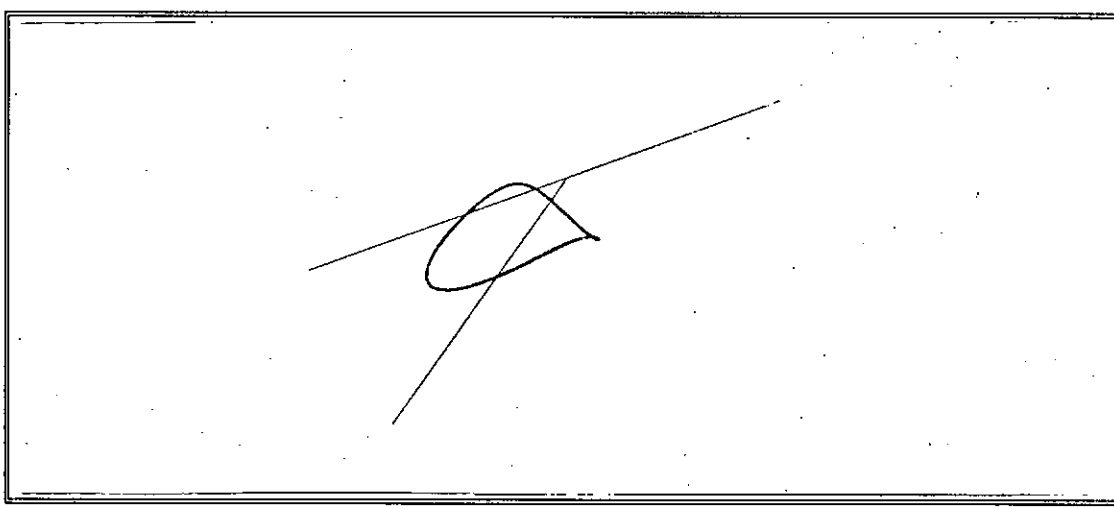


Fig V.118

Antes da execução, o sistema verifica se uma das extremidades do eixo do tubo secundário intercepta o eixo do tubo principal e se o diâmetro do tubo secundário especificado pelo usuário é menor que o diâmetro do tubo principal, caso contrário o sistema envia uma mensagem de alerta e o comando é abortado.

V.4.9.9. O COMANDO CONVERT

O comando **CONVERT** permite ao usuário converter entidades gráficas em elementos finitos. O comando se aplica a entidades do tipo ***solid***, ***3dface***, ***mesh*** e ***polylines***.

As entidades gráficas convertidas a elementos finitos estarão associadas ao material ativo no momento da execução do comando.

Para entidades gráficas do tipo ***solid***, ***3dface*** e ***line***, a conversão é direta.

Para entidades gráficas do tipo ***mesh***, geradas a partir dos comandos ***rulesurf***, ***tabsurf***, ***surftab***, ***edgesurf*** ou ***mesh*** do AutoCAD, o sistema inicialmente realiza uma divisão da entidade em ***3dfaces*** de três ou quatro vértices, para depois então, convertê–los em elementos finitos.

Para entidades do tipo ***polyline***, o sistema verifica inicialmente se esta corresponde a representação gráfica de algum dos elementos finitos disponíveis no sistema **AutoMESH**.

Caso a propriedade ativa não seja compatível com a entidade gráfica a ser convertida, o sistema requisitará ao usuário a definição da propriedade a qual os novos elementos finitos estarão associados.
Isto permite que, elementos planos e reticulados sejam processados em uma única operação do comando **CONVERT**.

Abaixo apresenta–se um exemplo de aplicação do comando **CONVERT**:

```
Command> Convert
Select object: W
First point: 0.0
Second point: 10.10
 6 selected
Select objectS: enter
Line element property: STEEL A
 2  Line elements
 9  Plate elements
 1  Infinite elements
command> enter
```

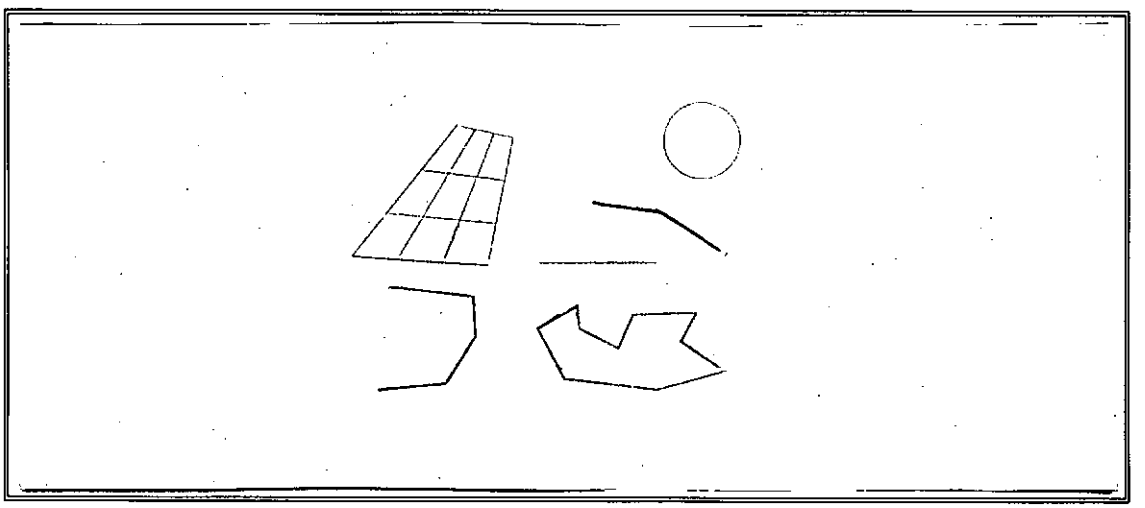


Fig V.119

No exemplo anterior, dentre o grupo de entidades selecionadas constam um ***circle***, um ***mesh***, um ***line*** e dois ***polyline***. Destes, o ***circle*** e um ***polyline*** foram desprezados pelo sistema.

Foi requisitado ao usuário, a especificação do material ao qual deveriam ser associados os elementos do tipo reticulado a serem convertidos. Isto deve–se ao fato da propriedade (material) ativa no momento da execução do comando **CONVERT** ser do tipo PLATE.

A especificação de um valor nulo para a propriedade requisitada, faz com que o sistema descarte este tipo de elemento da conversão.

V.4.9.10. O COMANDO CLEAR

O comando **CLEAR** funciona como um comando ***erase*** do **AutoCAD** de alto alcance. É capaz de remover todos os elementos estruturais de um determinado tipo em uma operação única e em um processo em cadeia.

O processo em cadeia citado aqui, é exatamente o mesmo do comando **DELETE** apresentado anteriormente.

Isto significa que, a remoção de determinado tipo de elemento estrutural provoca a remoção conjunta de todos os outros elementos estruturais associados a este.

As opções de operações do comando **CLEAR** são:

`All/Node/Element/REGion/Load/REStric/NodeID/ElemID:`

A figura V.120 apresenta a sequência de retirada de elementos estruturais associados.

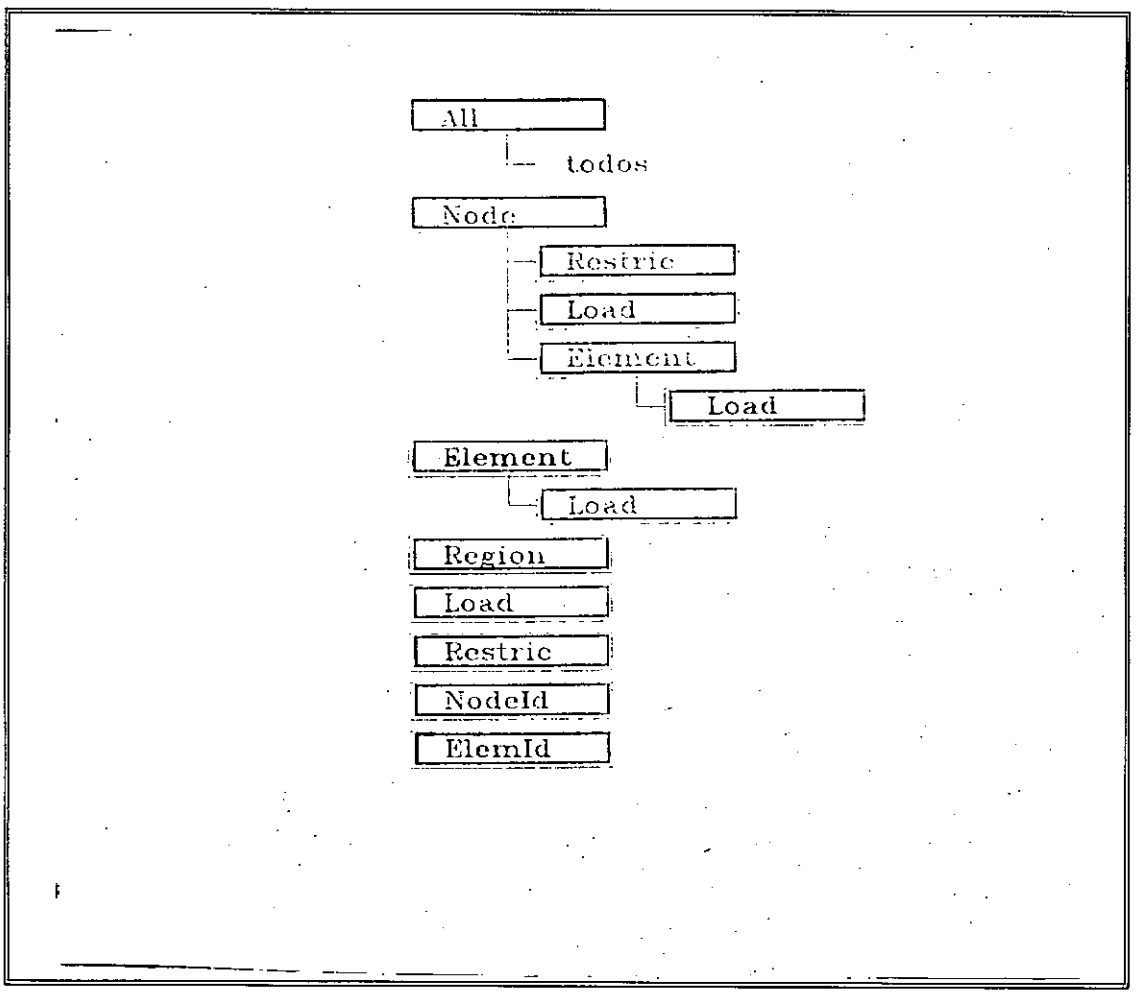


Fig V.120

– As opções **REGION**, **RESTRIC**, **NODEID**, **ELEMID** correspondem a remover todas as entidades gráficas contidas nos ***layers*** correspondentes a estes.

– Para a opção **ELEMENT**, o usuário é requisitado a especificar qual o tipo de material dos elementos finitos aos quais o comando será aplicado.

A remoção de elementos desencadeia a remoção dos carregamentos associados a estes.

– A opção **NODE** é mais complexa, pois desencadeia a remoção de todos os carregamentos nodais, restrições e elementos finitos.

– A opção **ALL** engloba todas as demais opções e corresponde a descartar do desenho toda e qualquer entidade com significado estrutural.

A seguir apresenta–se um exemplo onde o usuário removeu do modelo todos os carregamentos e os elementos com material AB52:

```
Command>CLEAR
All/Node/Element/REGion/Load/REStric/NodeID/
 ElemID: Loads
Are you sure?<N> Y
All/Node/Element/REGion/Load/REStric/NodeID/
 ElemID: Elements
Material Name: AB52
All/Node/Element/REGion/Load/REStric/NodeID/
 ElemID: enter
Command>
```

CAPÍTULO VI

VI.1. PROGRAMAS EXTERNOS

Neste são apresentados os programas externos ao ambiente **AutoCAD**. São programas que fazem parte do sistema **AutoMESH**, mas que não executam unicamente em função do **AutoCAD**.

Muitos destes programas são acionados a partir do ambiente **AutoCAD**, através de comandos do **AutoMESH**. O **AutoMESH** neste caso é responsável pela interface com o usuário para obtenção dos dados necessários e processamento dos resultados – exerce uma função de gerenciador.

Outros programas são apenas utilitários ou versões compiladas de comandos do **AutoMESH**.

VI.2. INSTALL.COM

Este programa foi concebido com o intuito de facilitar o processo de instalação e adaptação do ambiente **AutoCAD** e a utilização do **AutoMESH**.

Sugere–se que, para execução de tal comando, o **AutoCAD** já tenha sido instalado no *hard disk* do microcomputador em questão.

O INSTALL.COM executará automaticamente a cópia dos arquivos necessários do **AutoMESH** para o *hard disk* em diretórios específicos, realizando a criação destes diretórios caso necessário.
Além da cópia dos arquivos específicos do **AutoMESH**, o INSTALL.COM também substitui arquivos do **AutoCAD**, os quais foram alterados para adaptarem–se às necessidades do **AutoMESH**.
Alguns deles são:

. **ACAD.DWG** Este é o arquivo semente de todos os novos arquivos gráficos do **AutoCAD**. As alterações neste arquivo consistem basicamente na alteração de variáveis do sistema que controlam a execução de comandos **AutoLISP** e nomes *default* para menus, arquivos de shapes e outros.

. **ACAD.LSP** Este arquivo contém código em **AutoLISP** e é automaticamente carregado no início de cada seção. Para o caso do **AutoMESH**, verifica a necessidade ou não de inicialização do sistema.

. **ACAD.PGP** Devido ao fato do **AutoCAD** tomar toda a memória do micro para as suas operações, torna–se necessário reservar uma parcela da memória para a execução de programas externos. Nestes arquivos são definidos os comandos externos, os quais poderão ser executados através do **AutoCAD**, bem como a porção de memória reservada a cada um.

. **ACAD.MNU** Esta é a versão extendida do arquivo de menus padrão do **AutoCAD**. Foram incluídos os *screen menus*, *tablet menus*, *icons* e *pull down menus*.

. **ACAD.HLP** Esta é a versão extendida do arquivo de *help* para os comandos do **AutoCAD**. Foram incluídos os comandos do **AutoMESH** em forma de descrição suscinta da operação e opções de execução.

. **ACAD.SLB** Esta é a versão extendida da biblioteca padrão de slides do **AutoCAD** utilizados nos *icon menus*. Foram incluídos todos os slides utilizados pelo **AutoMESH** como explanatórios ou em *icon menus*.

Dentre os arquivos específicos do **AutoMESH** a serem instalados estão:

. **AUTOMESH.SHX** Neste arquivo estão contidos os 63 shapes correspondentes às possíveis combinações de restrições nodais.

. **AUTOMESH.LSP** Este arquivo contém o programa de inicialização do **AutoMESH** em **AutoLISP**. Dentre as operações que realiza estão: checagem do ambiente para inicialização de variáveis, carregamento dos comandos em **AutoLISP**, *menus* e *shapes*, bem como redefinição dos comandos básicos do **AutoCAD**.

. _____.LSP Estes correspondem aos diversos comandos e funções do sistema **AutoMESH**, escritos em **AutoLISP**. Os arquivos com extensão .LSP estão divididos em dois grupos puramente por efeito de organização: os comandos e as funções. Para o LISP, todos os comandos são funções. Aqui, optou–se por chamar de comandos, as funções que são incorporadas à lista de comandos do **AutoCAD** e portanto, podem ser executados como tais; e chamar de funções, aquelas que funcionam como subrotinas dos comandos ou de outras funções. Os arquivos de funções podem ser reconhecidos por conterem um '_' (*underscore*) no nome do arquivo, por exemplo, FOUR_.LSP.

. **AUTOMESH.MNU** Arquivo de menus alternativo para o **AutoCAD** e **AutoMESH**. Apresenta o conjunto completo de *menus* do **AutoCAD** e **AutoMESH** (*tablet*, *screen*, *pull down menus*) redefinidos. Os comandos foram redistribuídos e para o *tablet* e *pull down menus*, o conjunto de comandos disponíveis também foi alterado. Estas alterações visam possibilitar o aumento do rendimento e utilização destes *menus*, colocando disponíveis os comandos mais utilizados para a modelagem de estruturas em elementos finitos de forma ergonométrica.

. **AUTOMESH.SLB** Biblioteca de slides específica do **AutoMESH**. Além dos slides utilizados nos *icon menus*, possui um conjunto de slides explanatórios que são utilizados em conjunto com o comando *help* do **AutoMESH**.

Além destes arquivos existe um grupo de arquivos de programas executáveis. Alguns destes arquivos são executados diretamente no **AutoCAD**; outros, a partir de comandos em **AutoLISP**, os quais constituem interface. Existem também os isolados, os quais são executados a partir do sistema operacional, ou seja, fora do **AutoCAD**, ou executados por outros programas.

Dentre os arquivos executáveis que constituem o sistema **AutoMESH**, temos:

. **GEMAT.EXE** É o programa que executa as tarefas de cálculo e geração do comando do mesmo nome.

. **INTERT.EXE** É o programa que executa as tarefas de cálculo e geração do comando do mesmo nome.

. **REORDS.EXE** Programa externo que realiza a reordenação nodal da malha de elementos finitos do modelo para solução por *skyline*.

. **REORDW.EXE** Programa externo que realiza a reordenação nodal da malha de elementos finitos do modelo para a solução por *wave–front*.

. **AUTOMESH.EXE** Programa de geração automática em subregiões.

. **DBCREAT.EXE** Programa que gera o arquivo de dados no formato final. Dentre as tarefas que realiza constam, os testes de consistência, reordenação e formatação dos dados.

. **AUTOLOAD.EXE** Programa que permite que arquivos de dados em diversos formatos, possam ter seu modelo construído no **AutoCAD** e manipulados através do **AutoMESH**.

. **AUTOPOST.EXE** Programa externo que prepara os resultados da análise de um modelo para serem pós–processados pelo **AutoMESH** através do comando **POST**. Dentre as tarefas que realiza estão:
geração de deformadas, isolinhas, mapeamento por cores e contrução de vetores.

VI.3. GEMAT.EXE

Este programa constitui uma ferramenta muito útil para a geração de malhas de elementos finitos em juntas tubulares.

Este programa foi elaborado por Marcio Ferreira Alencar como parte integrante da tese de mestrado [191].

Construído para executar inicialmente em um computador CDC (*Control Data Cyber*), possui versões em diversas máquinas e atualmente esta para microcomputadores.

Escrito em *fortran*, foram necessários apenas pequenos ajustes de modo a permitir sua compilação segundo o MICROSOFT FORTRAN 5.0.

O programa **GEMAT** é utilizado no sistema **AutoMESH** em conjunção com o comando **GEMAT**, escrito em **AutoLISP**, responsável pela obtenção e preparação dos dados e interface com o usuário. O comando **GEMAT** do **AutoMESH** também é responsável pelo gerenciamento e aproveitamento dos resultados e inclusão da malha gerada pelo programa no modelo.

Basicamente, o programa constitui–se de um gerador de elementos finitos triangulares em juntas tubulares com um número qualquer de tubos.

O refinamento da malha em determinadas regiões pode ser controlado pelo usuário, bem como o posicionamento de nós e elemento para definição de local de continuidade da malha ou fronteira entre subregiões.

Este programa é especialmente útil para o projeto de jaquetas de plataforma onde se deseja modelar uma junta para análise.

VI.4. INTERT.EXE

Este programa constitui um utilitário não só para a geração de modelos de elementos finitos, mas também para o detalhamento de estruturas onde ocorram interseções entre elementos tubulares.

Este programa foi originalmente elaborado pelo engenheiro Martius Rodrigues y Rodrigues e encontra–se implementado em VAX como parte do sistema GEDEP [191].

A versão utilizada no sistema **AutoMESH** sofreu pequenas alterações de modo a possibilitar a compilação e execução em microcomputadores. Escrito em *fortran*, apresenta–se compilado em MICROSOFT FORTRAN 5.0.

A esta versão também foi adicionada uma extensão segundo apresentado por PRASAD et alia [183] para determinação da linha de interseção entre elementos cônicos.

O programa **INTERT.EXE** é utilizado no sistema **AutoMESH** em conjunção com o comando **INTERT** escrito em **AutoLISP**. O comando **INTERT** é responsável pela obtenção dos dados necessários, interface com o usuário, gerenciamento e aproveitamento dos resultados da execução do programa e inclusão da figura gerada no desenho.

VI.5. REORDS.EXE

Este programa constitui–se de um reordenador nodal de uso geral para malhas de elementos finitos, onde no processo de análise utiliza–se um esquema de solução do sistema de equações por *skyline*.

O programa **REORDS** é utilizado no sistema **AutoMESH** em conjunção com o programa **DBCREAT**, como uma das etapas de montagem do arquivo de dados.

Este programa foi implementado originalmente pelo engenheiro Heitor Araújo Filho como parte integrante de sua tese de mestrado [193] e constitui–se em uma implementação do algoritmo de CURTHILL at alia [196] modificado como proposto por LIU et alia [197] para redução da matriz de rigidez.

A versão utilizada no sistema **AutoMESH** sofreu pequenas alterações de modo a possibilitar a compilação e utilização em microcomputadores.

Escrito em *fortran*, apresenta–se compilado em MICROSOFT FORTRAN *5.0*.

VI.6. REORDW.EXE

Este programa constitui–se de um reordenador nodal, de uso geral para malhas de elementos finitos, onde no processo de análise, utiliza–se um esquema de solução do sistema de equações por *wave–front.*

O programa **REORDW** é utilizado no sistema **AutoMESH** em conjunção com o programa **DBCREAT** como uma das etapas de montagem do arquivo de dados.

Este programa foi implementado originalmente pelo engenheiro Heitor Araújo Filho, como parte integrante de sua tese de mestrado [193] e constitui–se em uma implementação do argoritmo de KING [198] com variações sugeridas por PINA [199] para redução da matriz de rigidez para solução por *wave–front.*

A versão utilizada no sistema **AutoMESH**, sofreu pequenas alterações de modo a possibilitar a compilação e execução em microcomputadores.

Escrito em *fortran*, apresenta–se compilado em MICROSOFT FORTRAN 5.0.

VI.7. AUTOMESH.EXE

Este programa constitui o núcleo de geração automática de todo o sistema em versão compilada.

O programa **AUTOMESH** apresenta-se como uma opção mais sofisticada para os processos de geração automática implementados em **AutoLISP**.

Construído para possibilitar a geração de mallhas em processo alfanumérico, em conjunção com rotinas em **AutoLISP**, as quais constituem a interface com o usuário, permite a geração de malhas em subregiões graficamente.

Apesar de implementados totalmente em **AutoLISP** na sua versão principal, os comandos **ISOMAPP**, **TRANSIT**, **TRILINEAR** e **DOUBLE** fazem parte das tarefas realizadas pelo programa **AUTOMESH**.

Especialmente recomendado para ambientes onde a quantidade de memória do sistema seja reduzida ou a quantidade de elementos automaticamente gerados em uma subregião seja muito grande.

Escrito em *pascal* e compilado em TURBO PASCAL 5.0, apresenta as vantagens de velocidade de um programa compilado em relação ao **AutoLISP** interpretado e a possibilidade de utilização de *overlays*, necessitando portanto de pouca memória. Para a versão em **AutoLISP** dos comandos de geração automática, as funções necessitam estar na memória.

Para a geração automática de malhas de elementos finitos em regiões com poucos elementos, o ganho na velocidade de processamento do programa não chega a compensar quando somado ao tempo gasto na inclusão, inserção dos resultados do processamento no modelo.

Para a versão compilada da geração automática, as funções em **AutoLISP** para os comandos **ISOMAPP**, **TRILINEAR**, **DOUBLE** e **TRANSIT** executam apenas tarefas de interface com o usuário, captando os dados e gerenciando a execução e inserção dos resultados (malha gerada) no modelo.

No programa **AUTOMESH**, encontra–se implemetado o mapeamento transfinito, para a geração das malhas com possibilidade de particularização para o caso do mapeamento isoparamétrico por opção do usuário.

VI.8. AUTOLOAD.EXE

Este programa constitui um núcleo de tradução que permite ao usuário construir a representação gráfica de diversos modelos estruturais, a partir de um arquivo de dados.

O programa **AUTOLOAD** está preparado para converter arquivos de dados vindos de diversos programas tais como ADEP, IRM, PATRAN, GENESIS, ANSYS e GENÉRICO, em um arquivo do tipo **DXF** do **AutoCAD**.

Após montado o modelo, estarão disponíveis todos os recursos do **AutoCAD** e **AutoMESH**, para detalhamento do desenho, manipulação e edição da malha de elementos finitos, bem como propriedades, carregamentos ou restrições.

O programa **AUTOLOAD** permite a tradução de elementos sólidos, apesar do **AutoMESH** não tratar esse tipo de elemento finito. Este recurso visa permitir ao usuário visualizar, checar ou mesmo plotar modelos constituídos de elementos sólidos.

O programa **AUTOLOAD** está capacitado a traduzir arquivos de dados fornecidos pelo ADEP através do comando NEUTRAL FILE [186].

Modelos gerados no IRM [185] ou SMS [195] poderão ser traduzidos pelo **AutoMESH**, quando transferidos através do INTERGRAPH NEUTRAL FILE.

Arquivos de dados compatíveis com o ANSYS [187] poderão ser diretamente traduzidos.

Os arquivos de dados do programa GENESIS [194] poderão ser diretamente traduzidos e montados graficamente no **AutoCAD**.

O formato genérico diz respeito a um padrão de transferência de dados proposto na PETROBRÁS [190] e corresponde ao formato adotado pelo **AutoMESH** como *default*.

Deve–se salientar que diversos cuidados foram tomados na confecção dos tradutores, mas incompatibilidades continuam a existir devido a própria origem dos dados a serem traduzidos.

No caso de estruturas de dados complexas como o ADEP, IRM e ANSYS, O **AutoMESH** na versão atual não está capacitado a assimilar todos os dados fornecidos. Então, no caso da tradução inversa, para o caso de modelos editados ou extendidos, não será completa.

Para estruturas de dados simples tais como no caso do GENESIS, onde só são tratados os dados topológicos, o tradutor assume valores *default* para os dados não fornecidos, como por exemplo, as propriedades dos materiais.

VI.9. DBCREAT.EXE

Este constitui o programa mais importante do sistema **AutoMESH**. É responsável pela geração e checagem dos dados referentes à estrutura modelada em elementos finitos.

O programa **DBCREAT** é utilizado no sistema **AutoMESH** em conjunção com o comando **DATABASE** escrito em **AutoLISP**, o qual é responsável pela preparação dos dados necessários e interface com o usuário.

O processo de elaboração do arquivo de dados consiste de cinco fases distintas:

. Filtragem
. Checagem
. Geração de Topologia
. Reordenação
. Tradução

A etapa de filtragem consiste em selecionar apenas as entidades gráficas de interesse estrutural, (elementos, carregamentos, restrições) e seus atributos.

Dentre as entidades gráficas passíveis de seleção estão:

. *3dfaces*
. *Solids*
. *Polylines* fechadas com 3,4,6 ou 8 vértices
. *Polylines* abertas com 5 vértices
. *Blocks* de perfis

Desde que construídos em específicos ***layers*** com definição de propriedades e materiais.

Os dados referentes aos carregamentos e restrições são obtidos e checados através do comando **DATABASE**. Este comando extrai do modelo cada um dos ***blocks*** correspondentes aos carregamentos e restrições e verifica a existência dos nós ou elementos a que estão associados.

A etapa de checagem consiste da verificação das dimensões e incidência dos elementos.

Primeiro, verifica–se a proporção entre os lados do elemento que deverá ser inferior a 10 vezes, para evitar problemas de imprecisão dos resultados obtidos na análise.

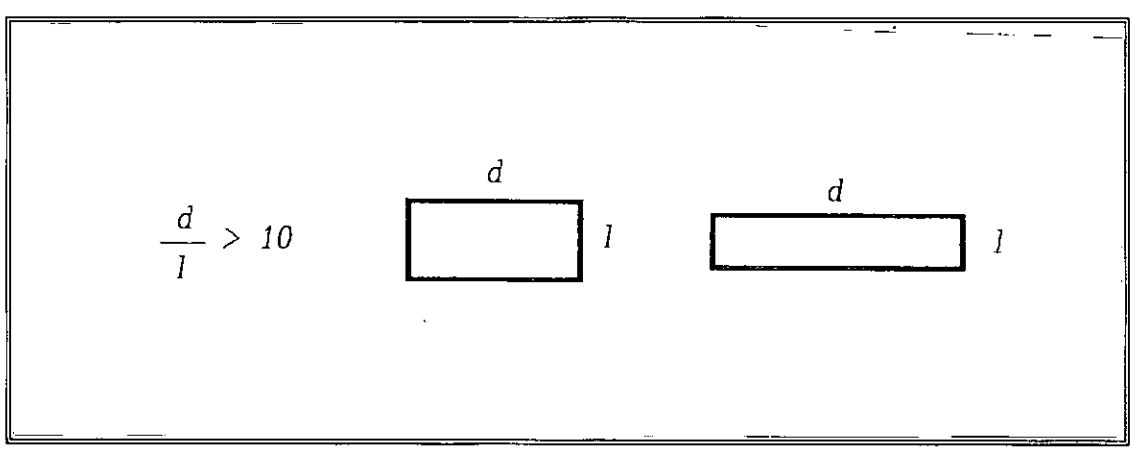


Fig VI.1

Para elementos de barra é verificado se possui comprimento nulo.

A seguir, verifica–se o grau de distorção do elemento no plano e fora do plano. No primeiro, verifica–se o ângulo formado nos vértices do elemento.

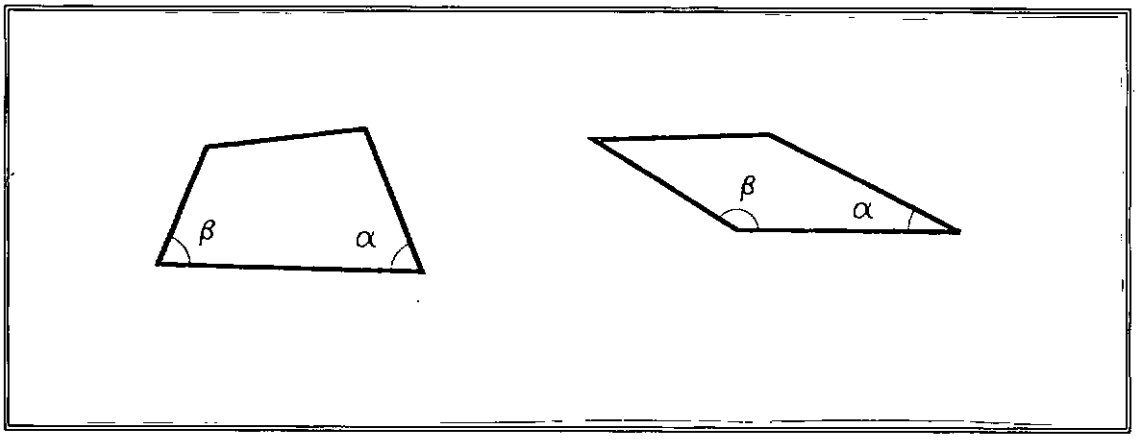


Fig VI.2

No segundo, verifica–se o ângulo formado pelas arestas que ligam dois nós e o plano formado pelos demais nós.

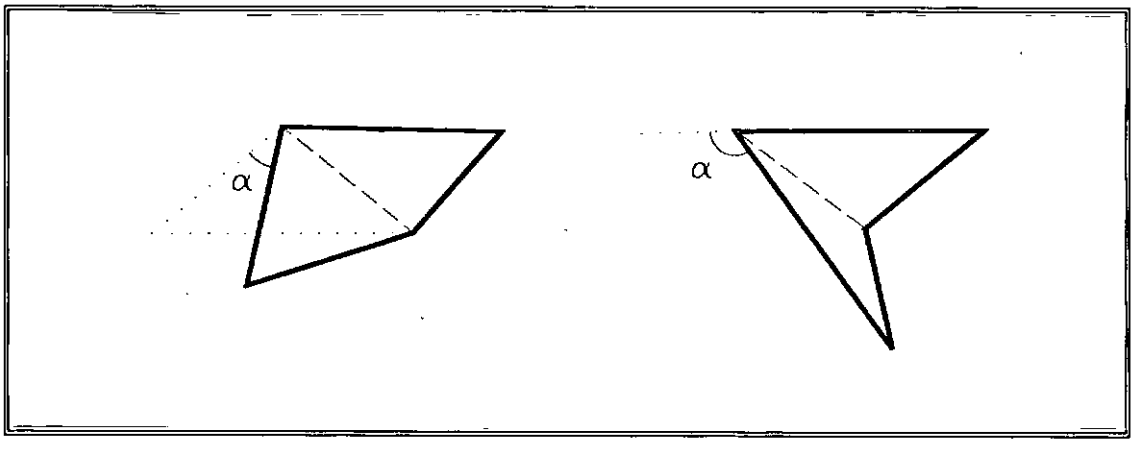


Fig VI.3

Outra checagem feita, é quanto ao sentido horário ou anti–horário da incidência, uma correção é efetuada se necessário.

Além das verificações de forma e incidência, são realizadas mais duas checagens de ligação entre elementos.

A primeira checagem diz respeito a duplicidade de elementos, ou seja, verifica–se se existem elementos coincidentes, não só do ponto de vista de incidência, mas localização.

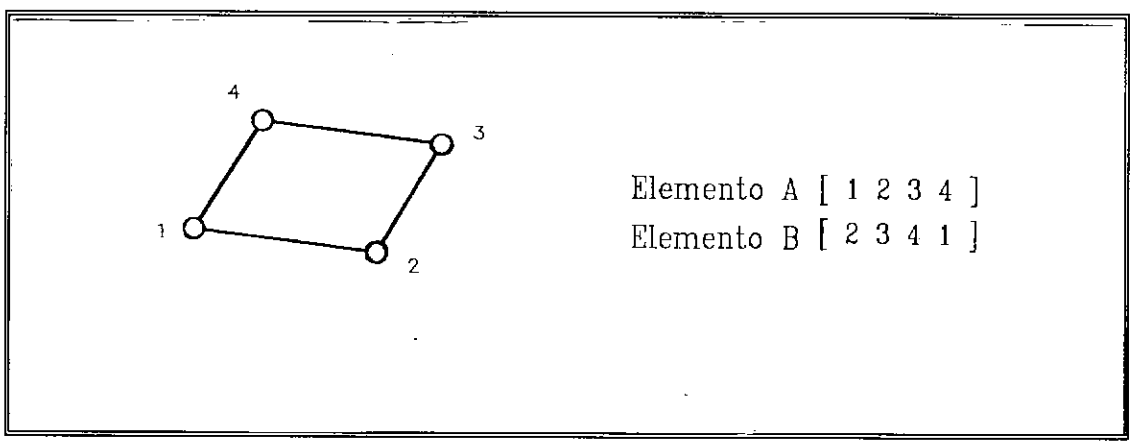


Fig VI.4

Como podemos verificar na figura acima, esta checagem não se trata apenas de uma comparação direta da incidência e sim uma comparação cíclica.

A segunda checagem tem como função, verificar a continuidade na ligação entre elementos.

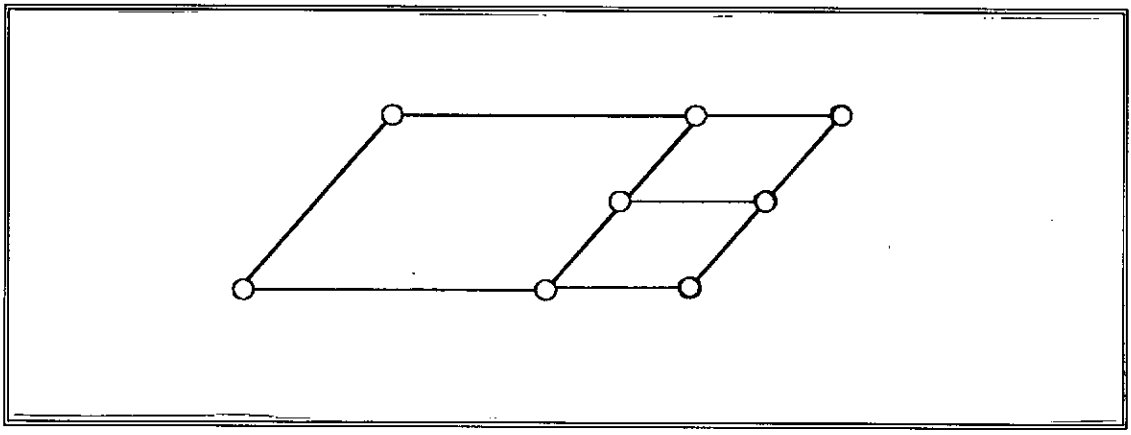


Fig VI.5

No **AutoMESH**, assume-se que os elementos utilizados, possuem funções de forma linear ou quadrática, e com nós apenas nos vértices ou também no meio dos lados respectivamente.

Para evitar a incompatibilidade de ligações entre elementos lineares e quadráticos, procede-se a verificação da incidência de todos os elementos em relação aos nós.

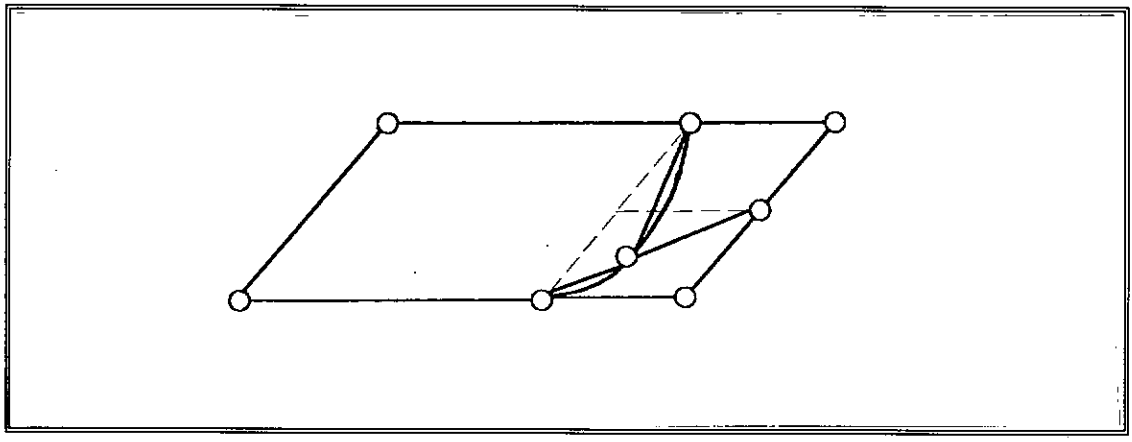


Fig VI.6

Para efeito desta checagem, os nós são divididos em duas categorias:

. nós de vértice,

. nós de aresta.

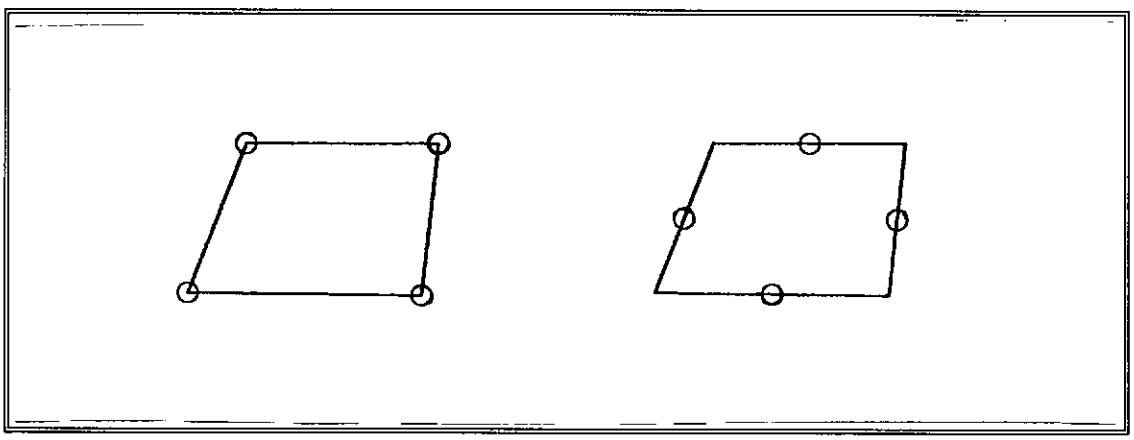


Fig VI.7

Cada um dos nós da malha de elementos finitos é classificado segundo uma destas categorias. Se um nó pertencente a dois elementos é classificado em uma categoria por um elemento e em outra categoria pelo outro, verificou–se então um caso de descontinuidade e o usuário é alertado do problema.

O programa **DBCREAT** fornece ao usuário um arquivo com todos os detalhes e descrição de todos os problemas encontrados durante a fase de checagem da malha, bem como os procedimentos adotados. As mensagens emitidas estão divididas em duas categorias:

. ALERTA

. ERRO

. INFORMATIVAS

As mensagens de ERRO informam ao usuário sobre condições críticas e sugerem a alteração da malha de elementos finitos.

As mensagens de ALERTA informam ao usuário sobre condições próximas às condições de erro.

As mensagens INFORMATIVAS referem–se a descrição dos procedimentos adotados pelo **DBCREAT** para a correção da malha.

Em todas as categorias, o DBCREAT fornece informações suficientes de modo a possibilitar ao usuário localizar os problemas ocorridos na malha para correção.

A etapa seguinte, diz respeito à geração da topologia propriamente dita. Nesta etapa, é montada a incidência dos elementos como serão apresentados externamente no arquivo de dados.

Os vértices de uma entidade gráfica, tal como ***polylines***, são armazenados sequencialmente no banco de dados gráfico do AutoCAD.

A tabela VI.8 apresenta as correspondências entre as entidades gráficas que representam os elementos finitos e suas incidências.

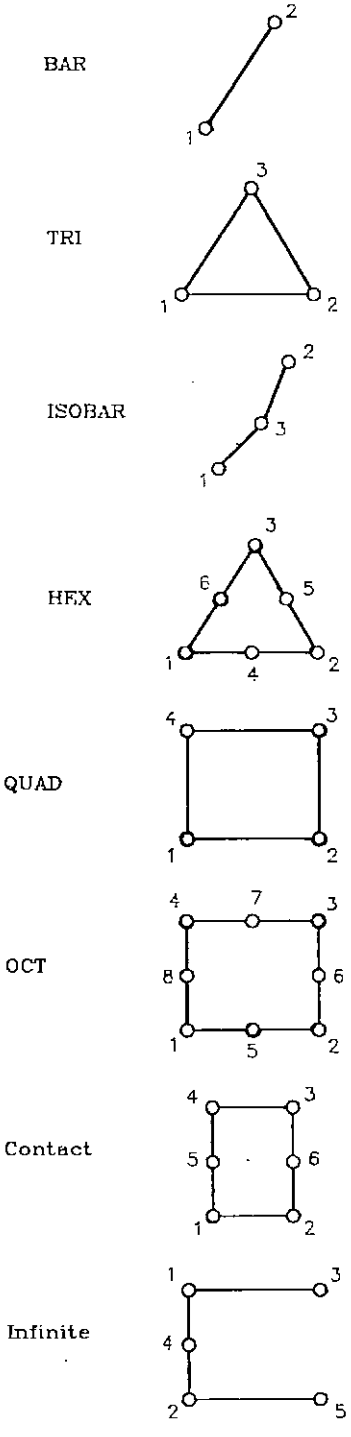

BAR
1 2 0 0 0 0 0 0
TRI
1 2 3 3 0 0 0 0
ISOBAR
1 2 3 0 0 0 0 0
HEX
1 3 5 5 2 4 0 6
QUAD
1 2 3 4 0 0 0 0
OCT
1 3 5 7 2 4 6 8
Contact
1 2 4 5 6 3 0 0
Infinite
2 4 1 3 5 0 0 0

Fig VI.8

A etapa seguinte corresponde a reordenação nodal da malha de elementos finitos.

Esta etapa é facultativa e ainda possui as opções de reordenação nodal para procedimentos de solução do sistema de equações por *wave–front* ou *skyline*. Estas tarefas são realizadas respectivamente pelos programas **REORDW** e **REORDS**, os quais são acionados pelo **DBCREAT**.

O **DBCREAT** gerencia o processo de reordenação nodal, suprimindo os dados necessários e captando os resultados dos programas **REORDS** e **REORDW**.

A etapa final do processo de geração do arquivo de dados do modelo estrutural é chamado de etapa de tradução.

Basicamente, o **DBCREAT** gera um arquivo de dados em formato genérico (*Generic File Format*) [190]. O usuário tem como opção a formatação do arquivo de dados de acordo com alguns programas de análise estrutural ou CAD (*Computer Aided Design*).

Os formatos disponíveis atualmente são:

. IRM/SMS

. ADEP

. ANSYS

. GENESIS

A tradução para o formato IRM [180] ou SMS [195] é feita através do INTERGRAPH NEUTRAL FILE.

O IRM constitui–se de um sistema completo de análise estrutural, ou seja, pré–processamento, processamento propriamente dito e pós–processamento, logo permitirá uma assimilação completa de todos os dados fornecidos pelo **AutoMESH**. Já o SMS, por tratar–se de um modelador de estruturas, tem a capacidade apenas de tratar elementos de barras e suas propriedades.

Para o ADEP, o formato adotado na tradução, é constituído da utilização de comandos de sua própria linguagem orientada.

Para o ANSYS, a tradução é feita através de arquivos de comandos, os quais serão executados através do próprio ANSYS, possibilitando a construção do modelo.

Para o GENESIS, o qual possui atualmente a estrutura de dados mais simples e menor, a tradução é imediata e o arquivo gerado pelo **DBCREAT** já corresponde ao modelo final.

Devemos salientar que incompatibilidades poderão ocorrer durante o processo de tradução. Estas incompatibilidades dizem respeito a dados fornecidos ou utilizados no **AutoMESH** sem correspondência no formato para o qual esta sendo traduzido. Estas incompatibilidades podem envolver propriedades dos elementos, cargas, restrições ou até mesmo o tipo de elemento utilizado no modelo.

O **DBCREAT**, durante o processo de tradução, busca a correspondência direta entre os dados. Caso esta correspondência não seja possível de realizar, tenta então transferir os respectivos dados de uma outra forma de modo a possibilitar que o dado seja efetivamente transferido. Caso não seja possível transferir, o dado é abandonado e mensagens de advertência são emitidas ao usuário.

Procedimento de tradução:

. Correspondência Direta

. Correspondência Indireta

. Abandono

A correspondência direta corresponde aos dados transferidos com sucesso.

No grupo dos dados traduzidos por correspondência indireta estão, por exemplo, dados relacionados a dimensões de perfis, onde o perfil adotado no modelo não possui correspondência no formato a ser traduzido. Para este caso, por exemplo, os dados poderiam ser transferidos através das propriedades da seção transversal, inércia, área, etc., calculados pelo **DBCREAT** a partir das dimensões e tipo da seção transversal.

No grupo dos dados abandonados estão todos aqueles onde não foi possível a tradução. Por exemplo, cargas, restrições e propriedades para o GENESIS.

O programa **DBCREAT** fornece ao usuário um arquivo com todos os detalhes e descrição das incompatibilidades encontradas e os procedimentos adotados.

Sugere–se que após obtido o arquivo de dados definitivo referente ao modelo, seja abandonada a malha original e uma nova seja construída a partir dos dados. Isto possibilita obter um desenho que representa fielmente o modelo a ser analisado, sem imperfeições ou incompatibilidades de ordem topológica, além de possibilitar a correta correspondência entre a numeração dos nós ou elementos no arquivo de dados e desenho. Esta opção é dada ao usuário através do próprio comando **DATABASE**, que se encarrega de realizar todas as tarefas automaticamente.

CAPÍTULO VII

Neste capítulo são apresentados diversos exemplos de utilização do sistema **AutoMESH**. Os exemplos visam demonstrar a potencialidade, interatividade e facilidade de utilização. Dentre os exemplos, estão aplicações básicas e simulações de condições particulares em discretizações de estruturas em elementos finitos, singulares e modelos especiais.

São apresentados também, exemplos de modelos reais efetivamente gerados a partir do **AutoMESH**, ou incluídos neste como demonstração da possibildade de utilização do **AutoMESH** para geração destes.

Este capítulo pretende demonstrar o desempenho do **AutoMESH** em relação à geração e edição das malhas de elementos finitos para diversos casos, desde os mais simples aos mais complexos.

Apesar de apresentar em alguns exemplos a sequência de operações, não pretende servir de guia de utilização.

Os exemplos apresentados aqui estão agrupados em cinco grandes conjuntos:

(1) **Exemplos acadêmicos** – Este grupo engloba os exemplos tirados da literatura, alguns dos quais utilizados inclusive para aferição de programas de análise estrutural.

(2) **Aplicações práticas 2D** – Neste grupo estão os exemplos referentes a modelos planos, bidimensionais. Dentre os exemplos, estão modelos gerados com o **AutoMESH** ou construídos neste através de arquivos de dados já existentes (gerados em outros sistemas de pré–processamento).

(3) **Modelos Reticulados** – Neste conjunto estão agrupados os exemplos de malhas com elementos finitos do tipo reticulado.

(4) **Elementos de Contorno** – Este grupo de exemplos apresenta um conjunto de modelos tridimensionais de malhas de elementos de contorno de aplicação prática.

(5) **Casos Especiais** – Neste conjunto estão agrupados os exemplos onde foram utilizadas técnicas "avançadas" para a geração da malha.

VII.1. EXEMPLOS ACADÊMICOS

Este grupo de elementos constitui um conjunto de aplicações básicas do sistema **AutoMESH**. São apresentadas algumas malhas e descritas as operações realizadas durante o processo de discretização.

Os exemplos I, II e III correspondem à malhas geradas com o auxílio do sistema **AutoMESH** e utilizadas para a aferição de programas de análise estrutural.

Os exemplos IV, V, VI e VII são apenas teóricos e visam demonstrar alguns recursos básicos do **AutoMESH**.

VII.1.1. EXEMPLO I

Este modelo foi discretizado baseado na malha apresentada por ZIENKIEWICZ [195], página 470 e representa uma placa retangular com um furo redondo no centro.

A figura VII.1 apresenta a malha gerada para um quarto da placa, segundo a simetria nas duas direções.

A figura VII.2 apresenta as sub–regiões utilizadas na discretização. Pode–se observar os pontos chaves (vértices e pontos médios às arestas) para cada sub–região. Deve–se lembrar que, como exposto anteriormente, as arestas sem ponto médio definido caracterizam uma distribuição retilínea e uniforme dos nós.

As figuras VII.3 e VII.4 apresentam os recursos de numeração nodal e de elementos respectivamente.

Foram utilizadas cinco sub–regiões neste exemplo, unicamente devido a distribuição específica de nós e elementos exigida.

VII.1.2. EXEMPLO II

Este modelo foi discretizado baseado na malha apresentada por ZIENKIEWICZ [200], página 472 e representa uma placa com um estrangulamento no meio.

A figura VII.5 apresenta a malha gerada para um quarto da placa. Os eixos de simetria correspondem às linhas horizontal superior e vertical esquerda.

Para a discretização do modelo foram utilizadas duas subregiões que dividem a malha horizontalmente. Ambas as sub–regiões foram definidas apenas através dos vértices, caracterizando assim uma distribuição retilínea e uniforme dos nós nas arestas.

As figuras VII.6 e VII.7 apresentam os recursos de numeração nodal e de elementos respectivamente.

VII.1.3. EXEMPLO III

Este modelo foi discretizado baseado na malha apresentada em ZIENKIEWICZ [200], página 474 e representa um vaso de pressão de aço.

A figura VII.8 apresenta a malha gerada para uma seção axissimétrica do modelo.

Este exemplo apresenta alguns detalhes interessantes a relatar a nível de discretização. Além dos comandos de geração automática para regiões de formato genérico (seção V.4.5.1), foram também utilizados comandos de edição (seção V.4.8) para a discretização do modelo.

O modelo foi dividido basicamente em três sub–regiões de discretização.

A região de topo foi discretizada manualmente, elemento por elemento. A região central, que corresponde à linha vertical dos elementos, foi gerada através do comando ISOMAPP com uma coluna e 44 linhas de elementos. No arco, os elementos foram gerados utlizando–se diversos recursos básicos do **AutoCAD**. Primeiro, desenhou–se os arcos internos e externos da cúpula. A seguir, através do comando DIVIDE do **AutoCAD**, os arcos foram divididos em setenta segmentos. Com o auxílio da opção ENDPOINT do comando OSNAP e dois segmentos paralelos, construiu–se um elemento finito. O passo seguinte foi utilizar o comando ARRAY do **AutoMESH** com opção POLAR para gerar a malha completa do modelo.

VII.1.4. EXEMPLO IV

Este exemplo visa mostrar a possibilidade de geração da malha com a utlização de sub–regiões.

A figura VII.9 apresenta a malha de elementos finitos para as cavernas de um navio. Foram utilizados em conjunto, elementos quadriláteros e triangulares.

A figura VII.10 apresenta as sub–regiões de discretização para o modelo. Todas as sub–regiões foram discretizadas com distribuição retilínea e uniforme dos nós nas arestas, ou seja, não foram utilizados os pontos médios às arestas.

VII.1.5. **EXEMPLO V**

Neste exemplo é apresentado o primeiro modelo tridimensional. O objetivo é demonstrar a aplicação das ferramentas de geração automática a modelos espaciais.

As figuras VII.11, VII.12, VII.13 e VII.14 apresentam diversas vistas da malha gerada.

A figura VII.15 apresenta as sub–regiões de discretização utilizadas para a geração automática. Para a definição dessas sub–regiões, primeiro desenhou–se o círculo e o retângulo em planos diferentes. A seguir, criou–se as linhas que ligam o círculo aos vértices do retângulo. O passo seguinte foi aplicar o comando ISOMAPP a cada sub–região, utilizando os extremos das linhas como vértices e o ponto médio, que define o quadrante do círculo, como ponto médio à aresta. A malha gerada é constituída de elementos triangulares e pode–se observar facilmente o processo de diagonalização aplicado aos elementos quadriláteros.

Obviamente, este modelo poderia ter sido discretizado de diversas outras formas, como por exemplo, discretizar um quarto do modelo e através dos eixos de simetria e do comando MIRROR. Outra possibilidade seria a geração da malha para cada sub–região através do comando EDGESURF do **AutoCAD**, utilizando como fronteiras um arco do círculo, as duas linhas e uma aresta do retângulo. A utilização subsequente dos comandos CONVERT e BREAKEL leva a mesma malha obtida através do comando ISOMAPP, com vantagens dos pontos nodais coincidirem perfeitamente com o arco.

A figura VII.16 apresenta o carregamento aplicado aos nós do topo.

VII.1.6. EXEMPLOS VI

Este exemplo representa uma cúpula de anfiteatro elíptica.

A figura VII.17 apresenta o modelo com as linhas ocultas removidas através do comando HIDE do AutoCAD.

As figuras VII.18, VII.19 e VII.20 apresentam diversas vistas para a malha gerada. Pode–se observar os carregamentos nodais aplicados.

A malha gerada foi obtida com a utilização do comando DOMES e SCALE que geraram os elementos e tornaram a calota elíptica respectivamente. A forma elíptica é obtida pela aplicação de fatores de escala diferentes para cada direção.

As aberturas na cúpula são obtidas pela remoção dos elementos e nós.

A figura VII.21 apresenta a utilização do comando SHIRINK para verificar a falta de elementos finitos no interior da malha ("vazios").

Na figura VII.22 retirou–se os elementos (ateavés do comando DISPLAY) e na figura VII.23, executou–se o comando HIDE do AutoCAD.

VII.1.7. EXEMPLO VII

Este exemplo representa uma cúpula de anfiteatro e difere do anterior pela forma de geração e elementos utilizados.

Neste modelo são utilizados exclusivamente elementos finitos triangulares.

A figura VII.24 apresenta a malha gerada. Neste modelo foi utilizado o comando TRILINEAR para a discretização de cada gomo da cúpula e o comando ARRAY com opção POLAR para a geração da malha como um todo.

A figura VII.25 apresenta o resultado da execução do comando SHIRINK e HIDE ao modelo discretizado.

VII.1.8. EXEMPLO VIII

As figuras VII.26 e VII.27 apresentam a malha de elementos finitos gerada a partir da aplicação do comando GEMAT a uma junta tubular, com dois tubos vista de dois pontos diferentes, após a aplicação do comando HIDE.

VII.1.9. EXEMPLO IX

A figura VII.28 apresenta outro exemplo de aplicação do comando GEMAT para a geração da malha de elementos finitos em uma junta tubular com três tubos.

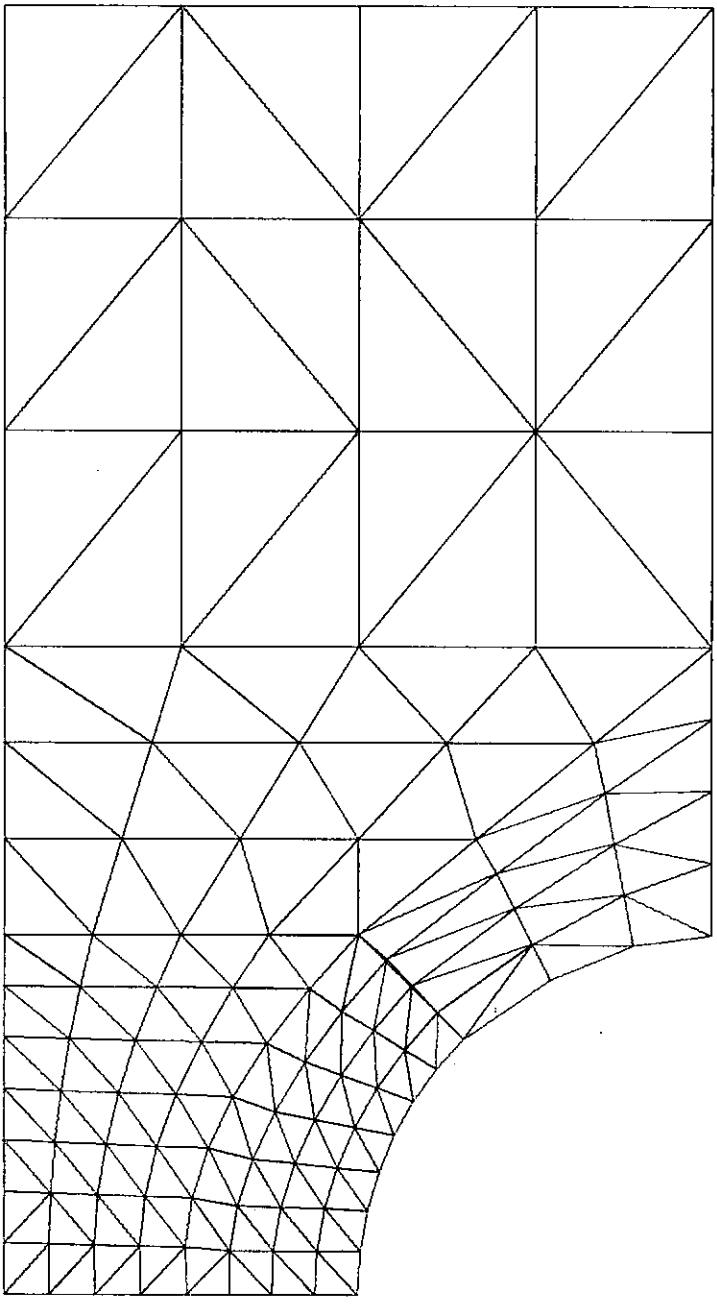


FIG VII.1.

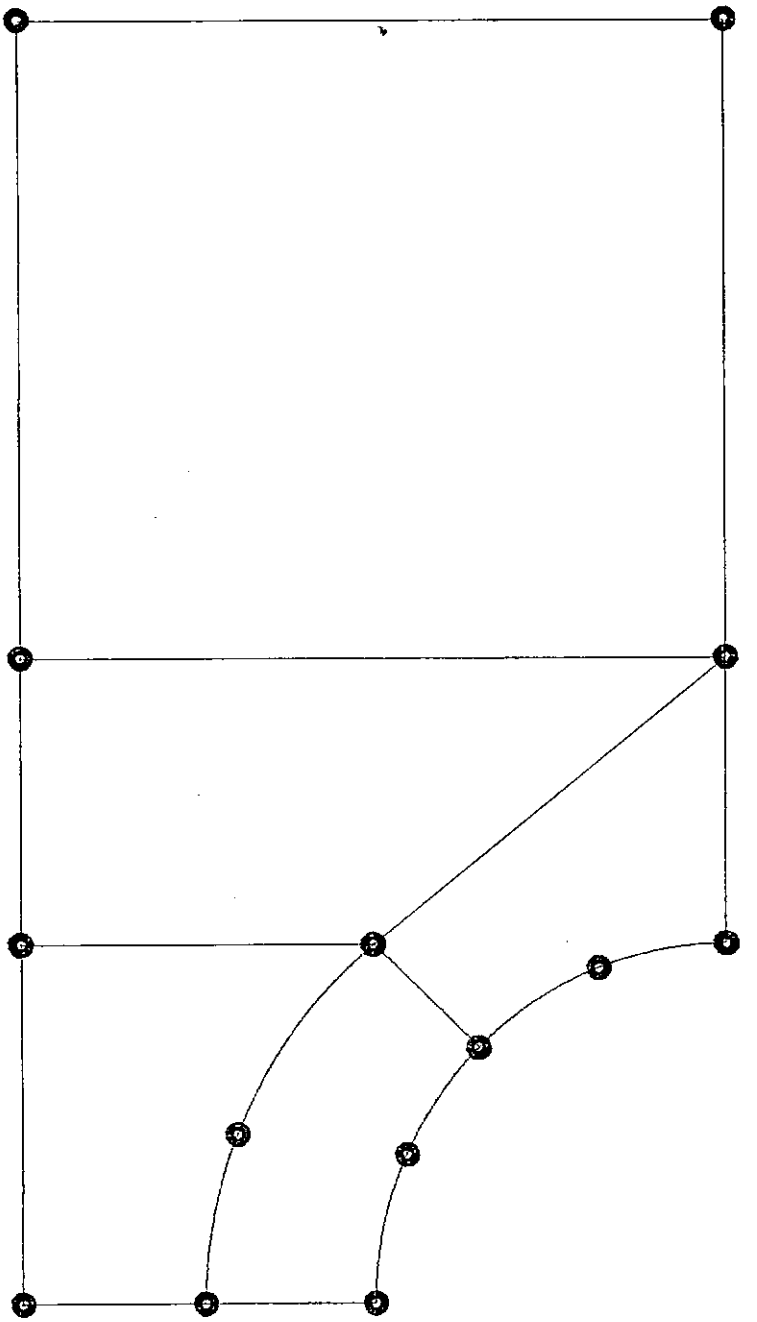


FIG VII.2.

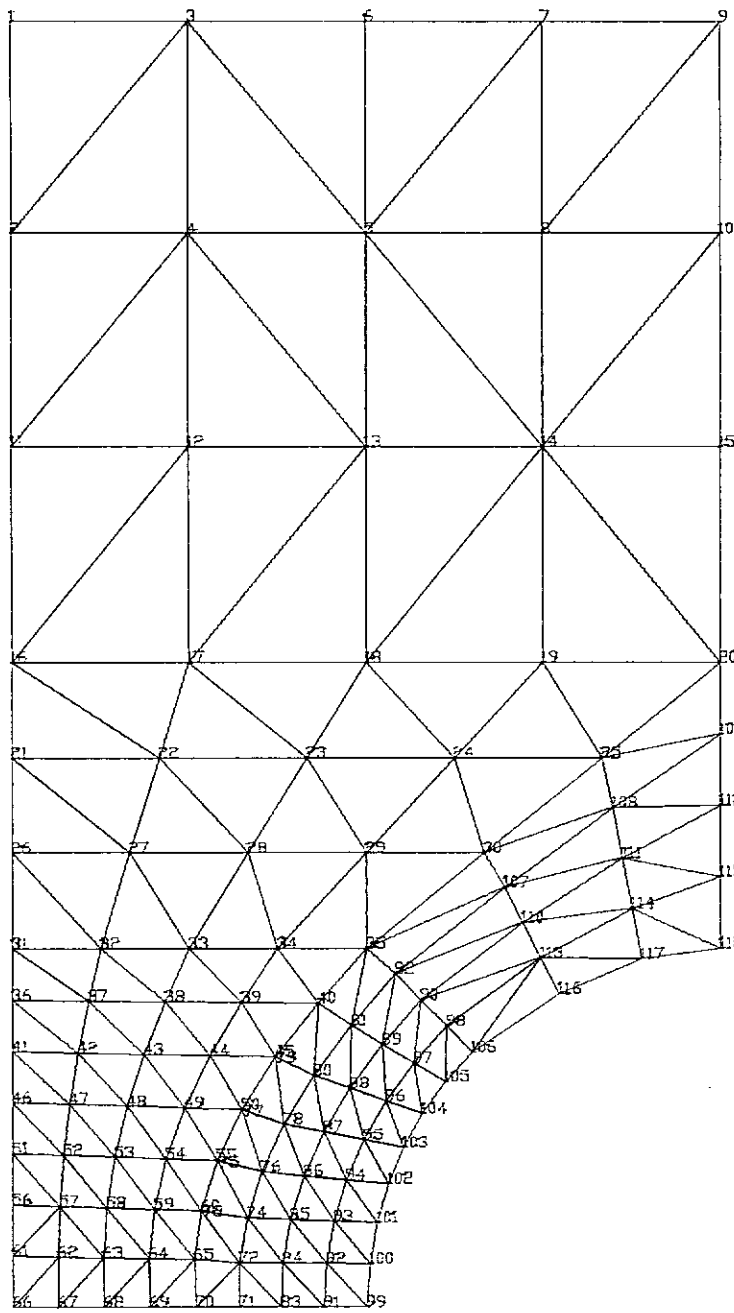


FIG VII.3.

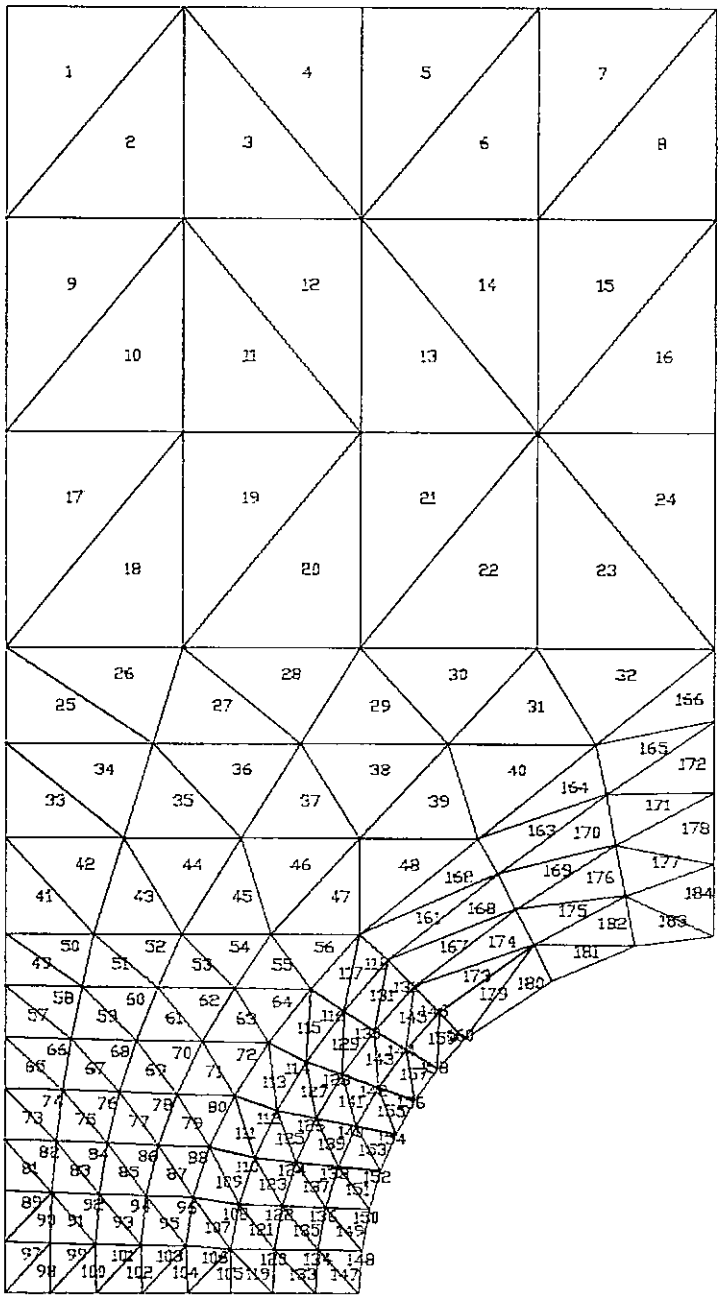


FIG VII.4.

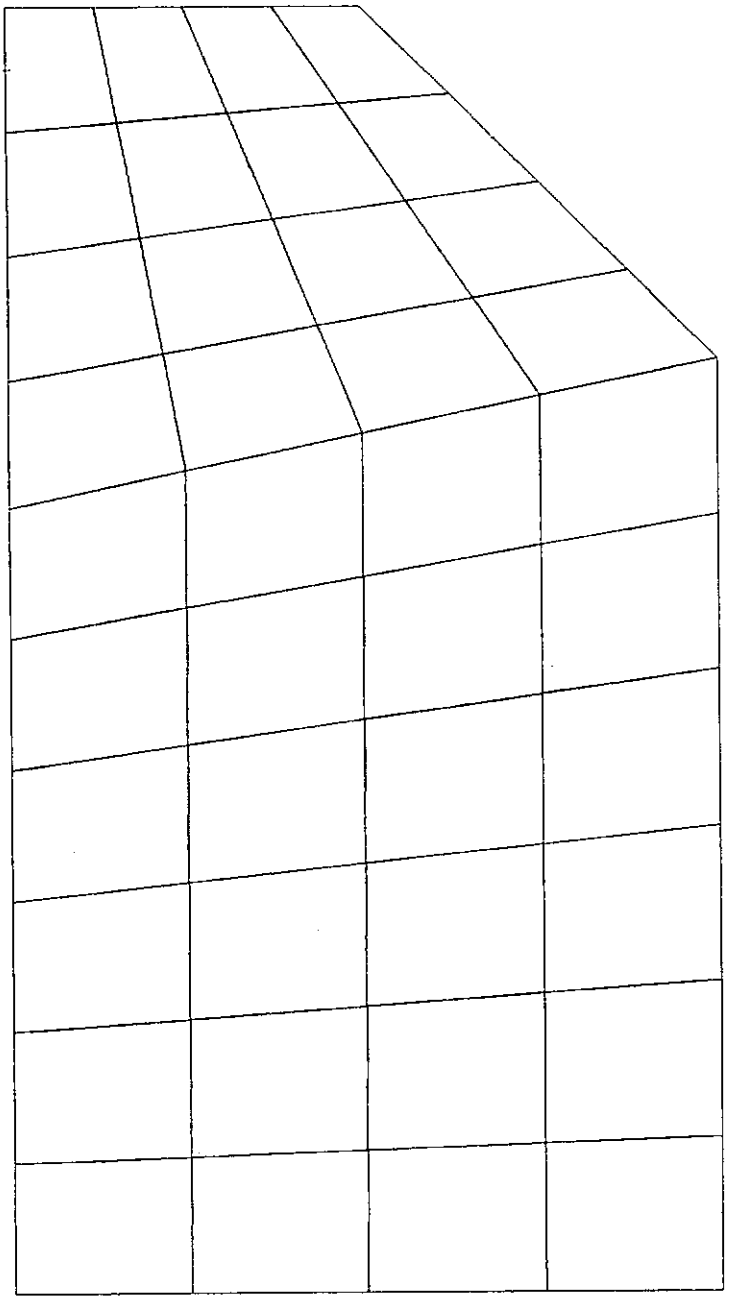


FIG VII.5.

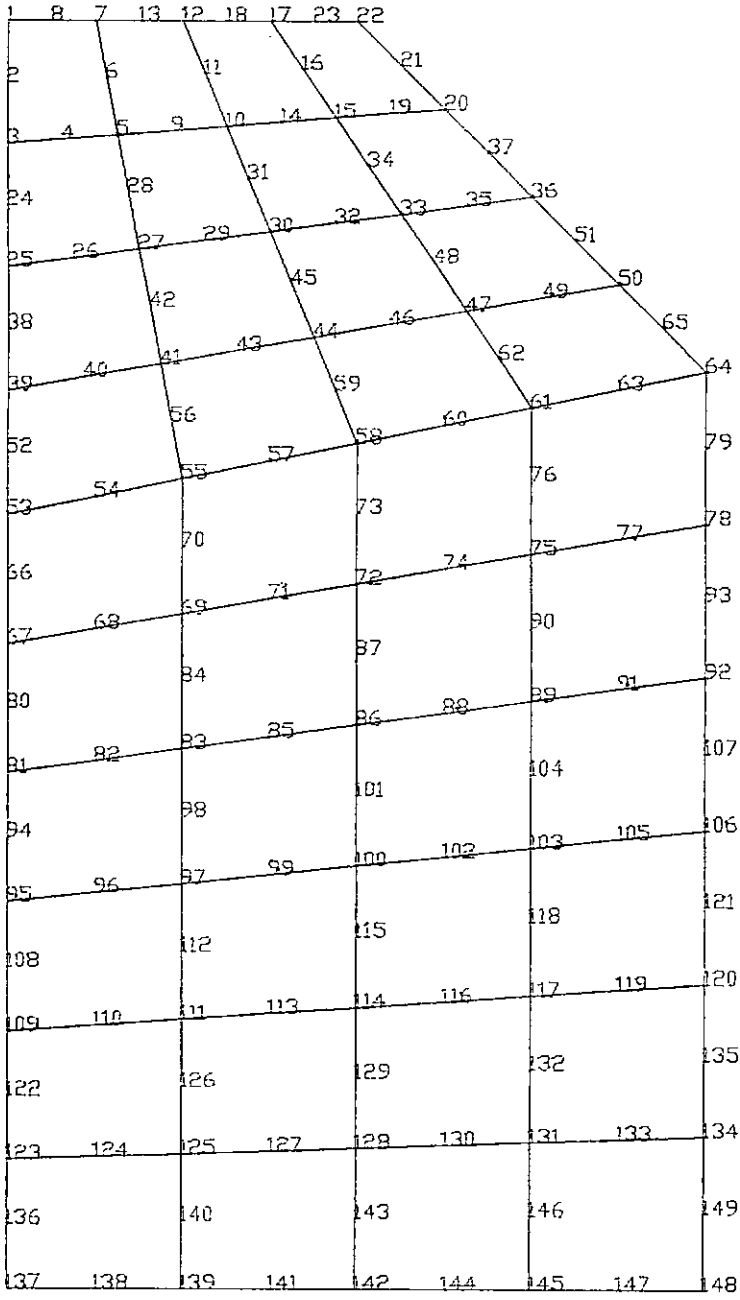

1 8 7 13 12 18 17 23 22
2 6 11 16 21
3 4 5 9 10 14 15 19 20
24 28 31 34 37
25 26 27 29 30 32 33 35 36
38 42 45 48 51
39 40 41 43 44 46 47 49 50
52 56 59 62 65
53 54 55 57 58 60 61 63 64
66 70 73 76 79
67 68 69 71 72 74 75 77 78
80 84 87 90 93
81 82 83 85 86 88 89 91 92
94 98 101 104 107
95 96 97 99 100 102 103 105 106
108 112 115 118 121
109 110 111 113 114 116 117 119 120
122 126 129 132 135
123 124 125 127 128 130 131 133 134
136 140 143 146 149
137 138 139 141 142 144 145 147 148

FIG VII.6.

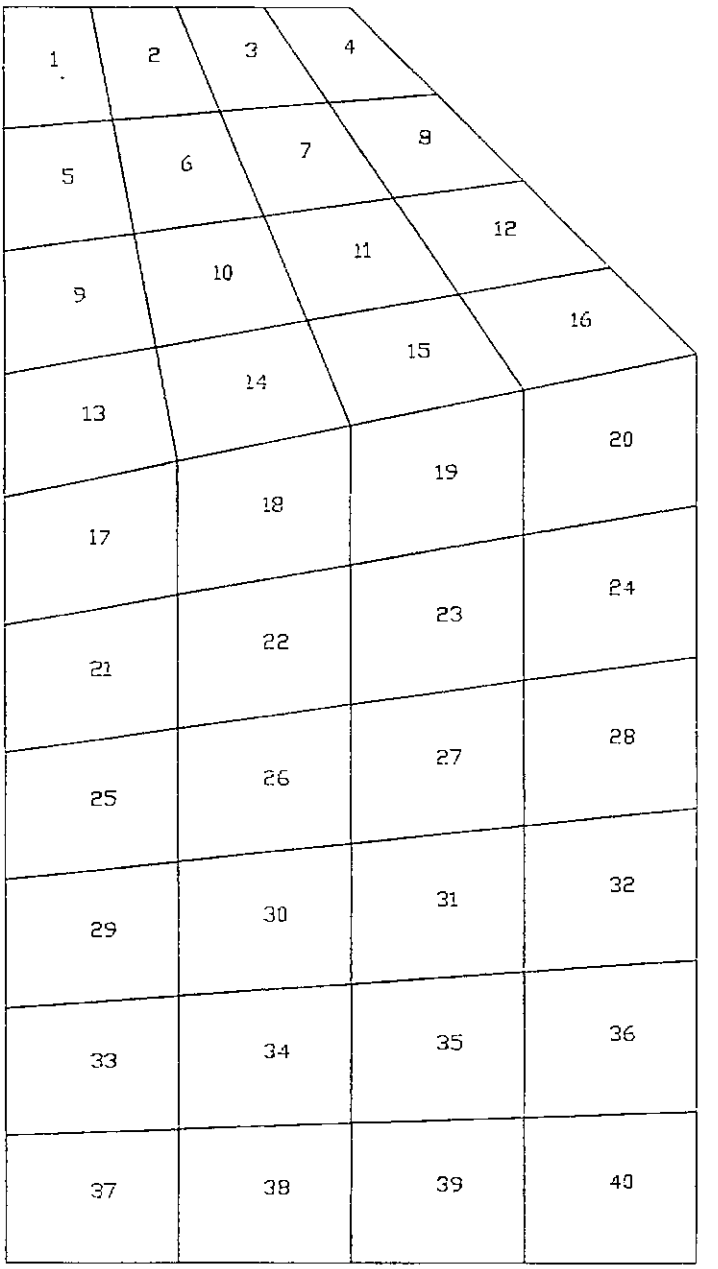

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

FIG VII.7.

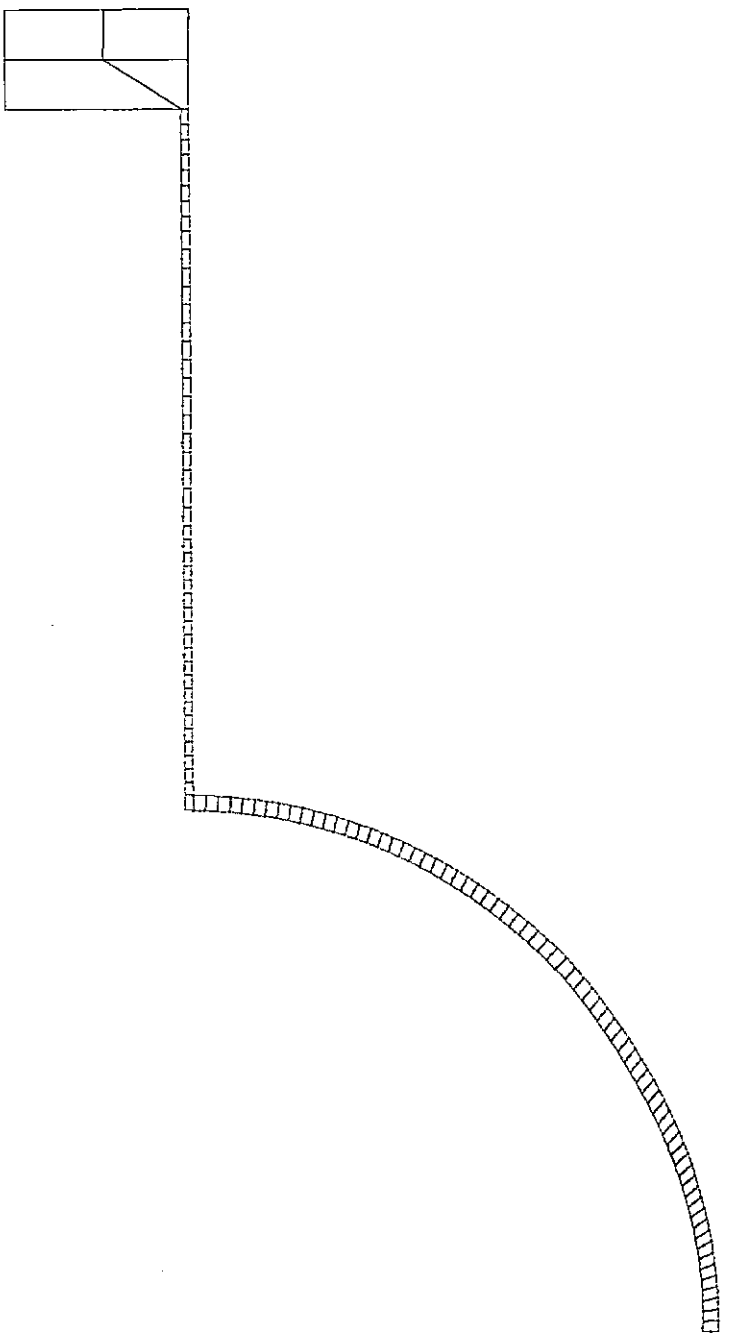


FIG VII.8.

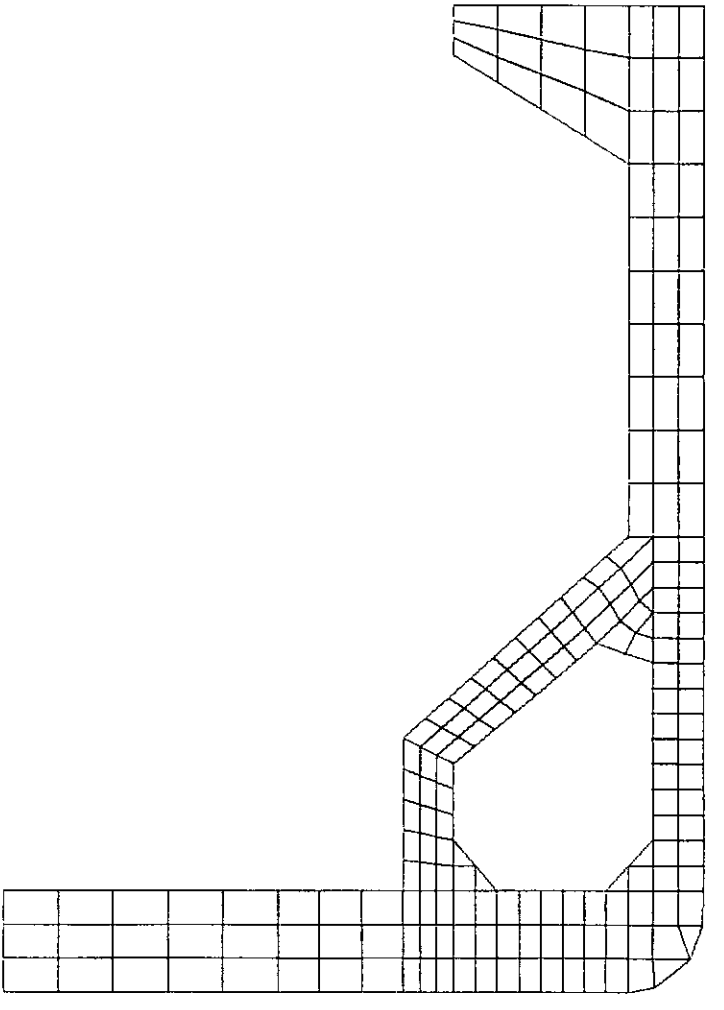


FIG VII.9.

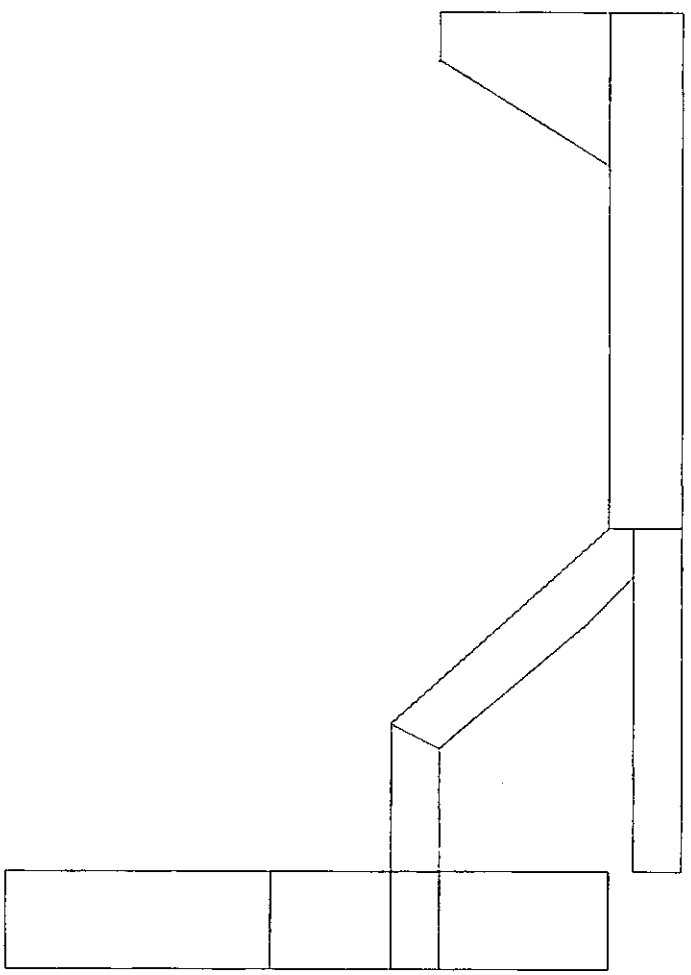


FIG VII.10.

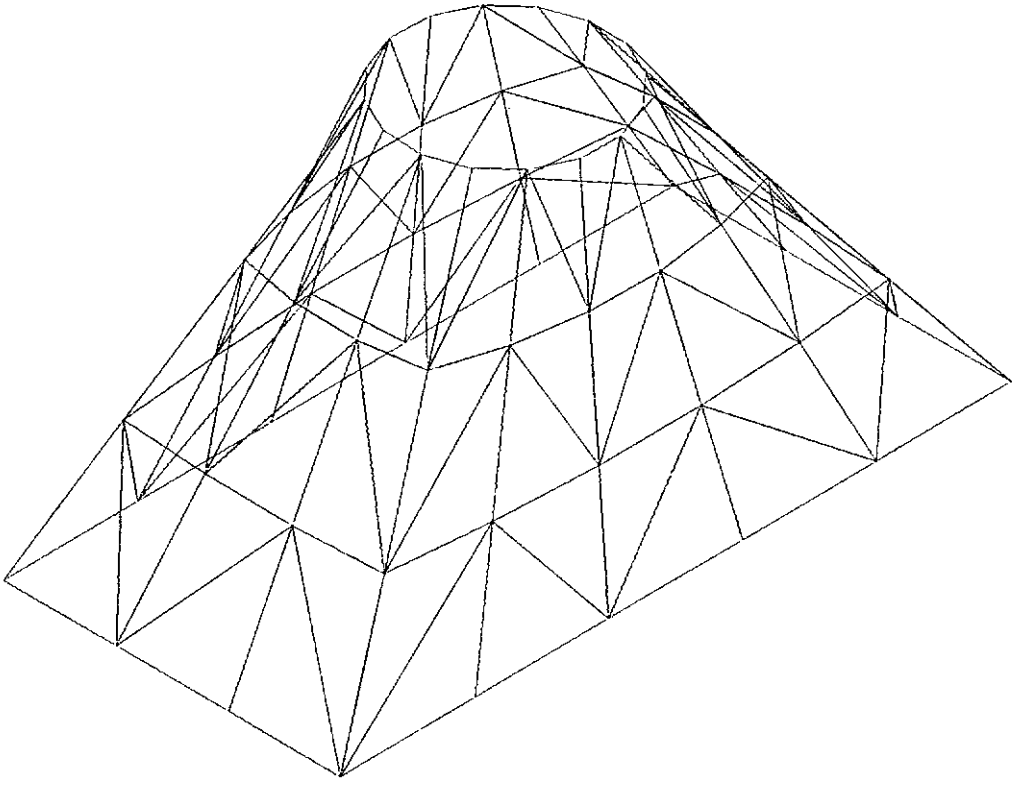


FIG VII.11.

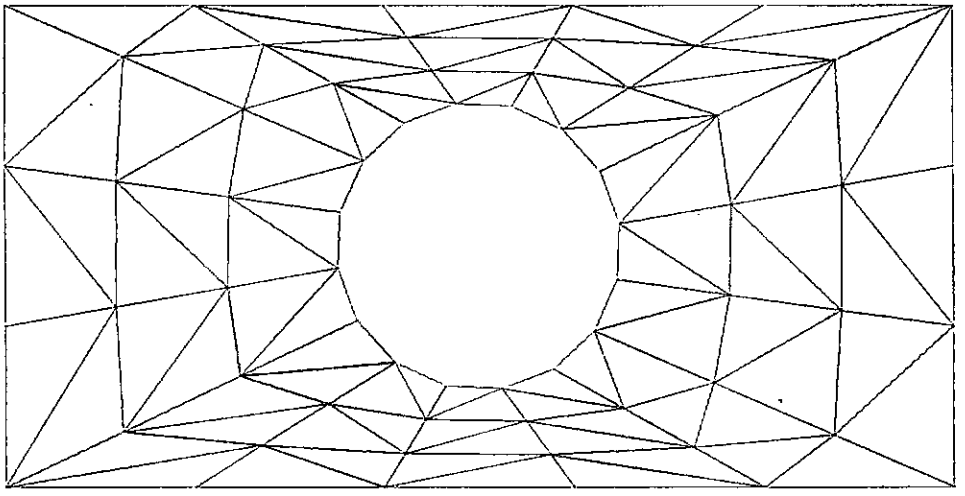


FIG VII.12.

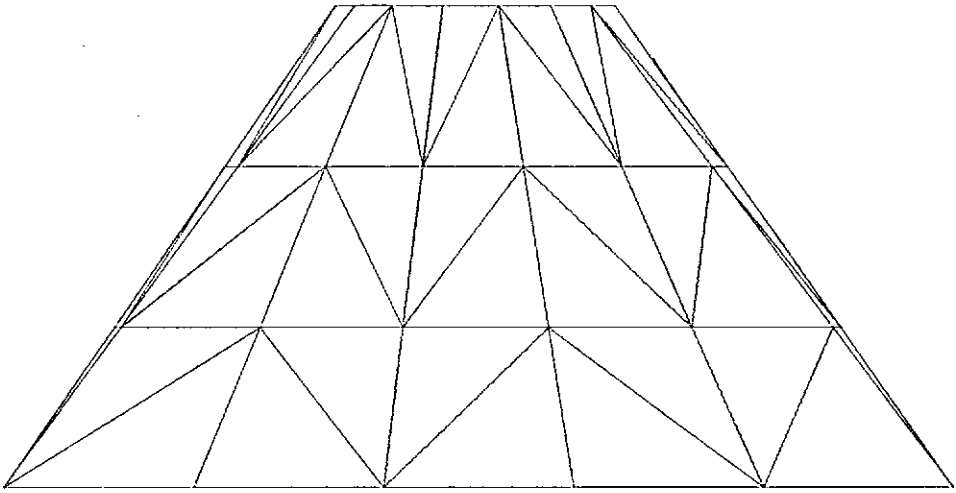


FIG VII.13.

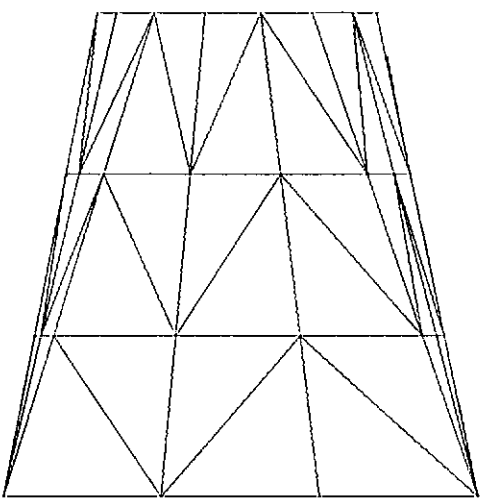


FIG VII.14.

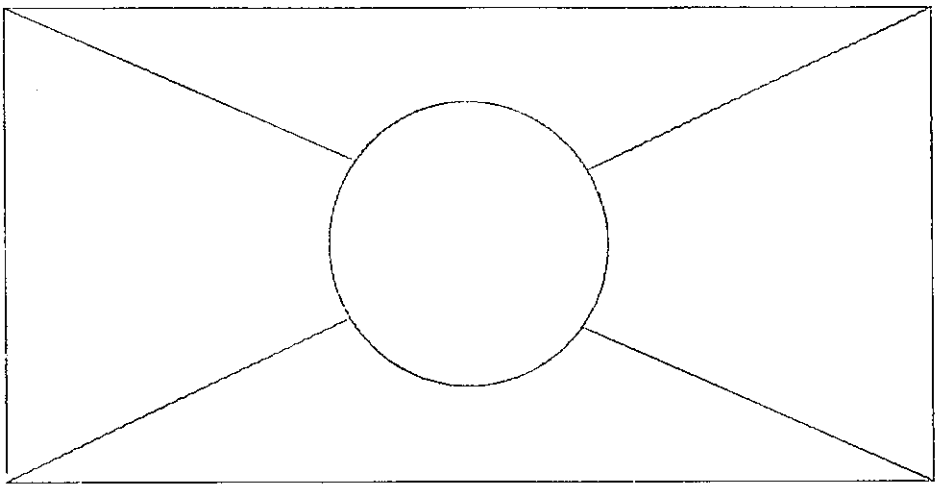


FIG VII.15.

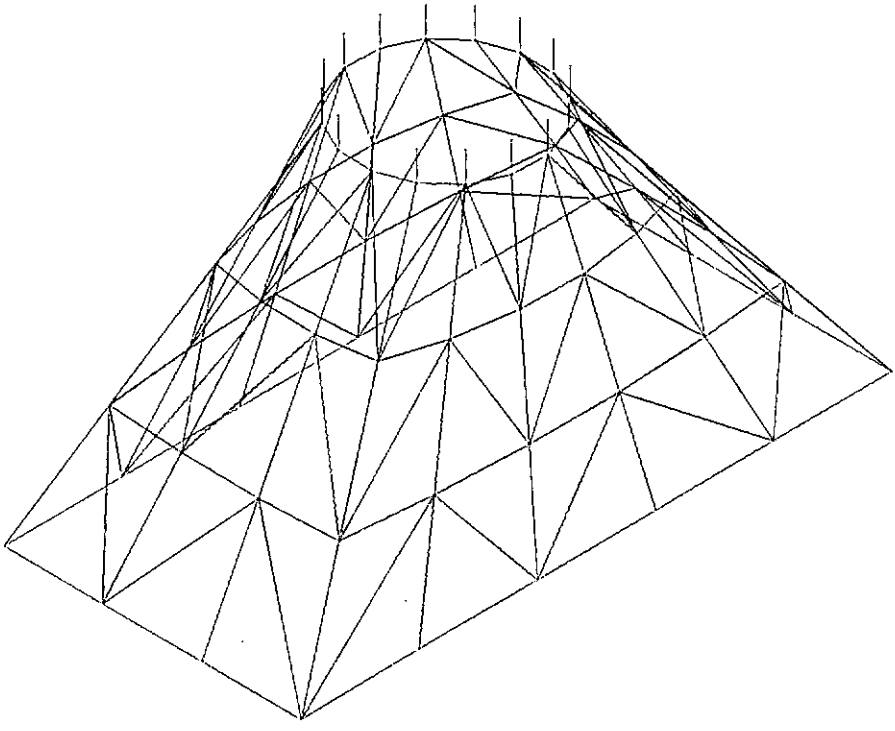


FIG VII.16.

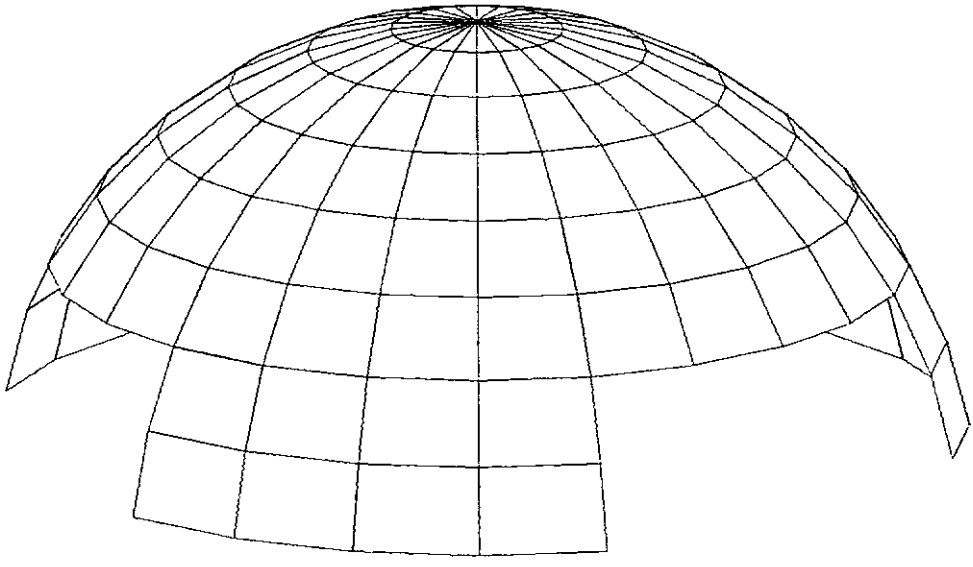


FIG VII.17.

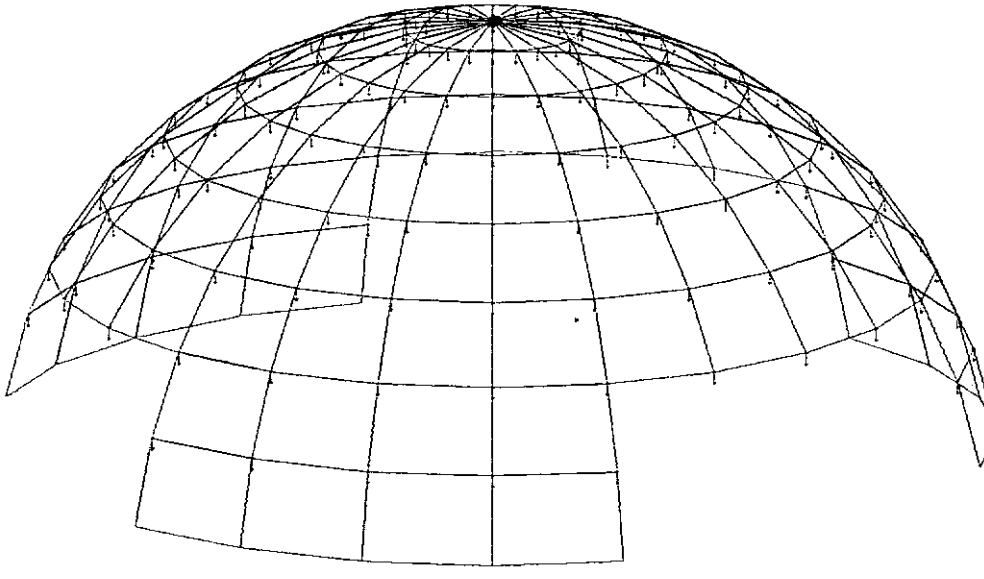


FIG VII.18.

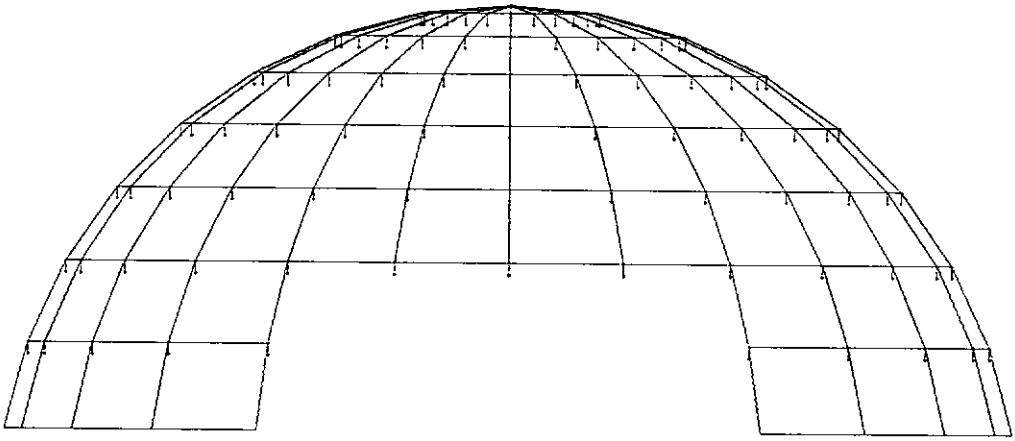


FIG VII.19.

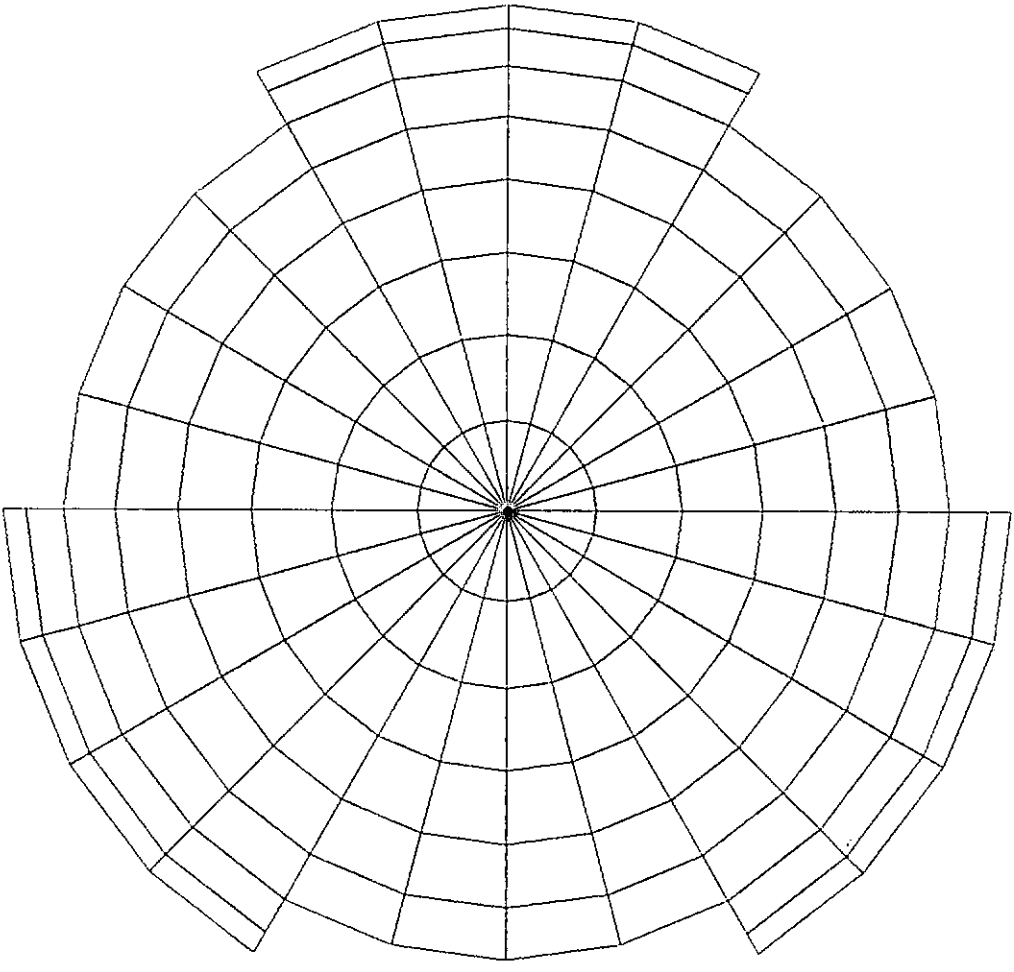


FIG VII.20.

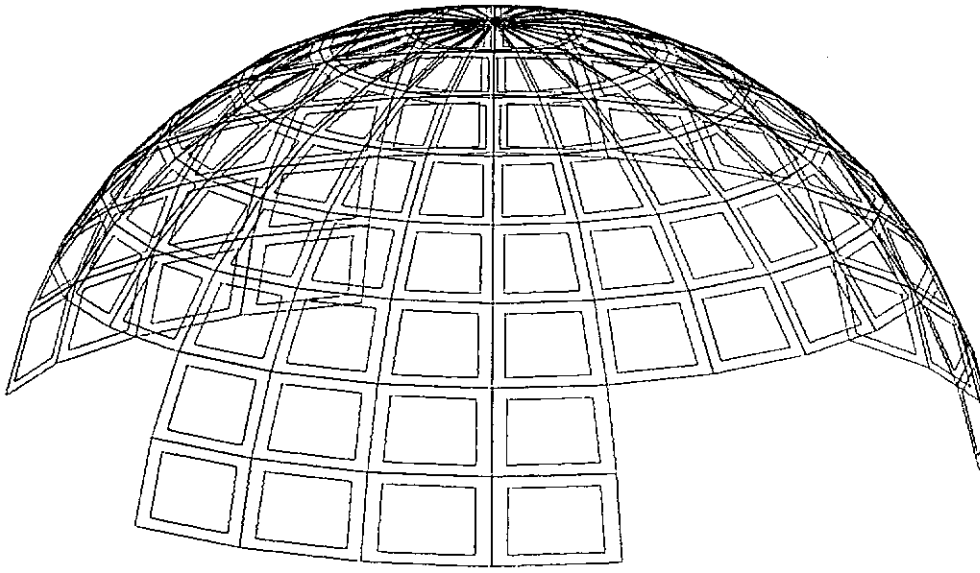


FIG VII.21.

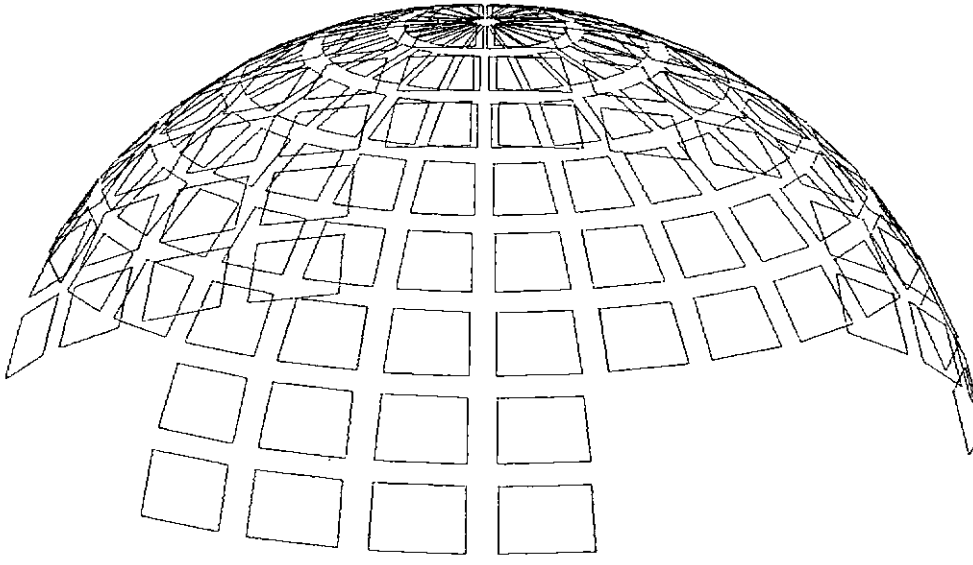


FIG VII.22.

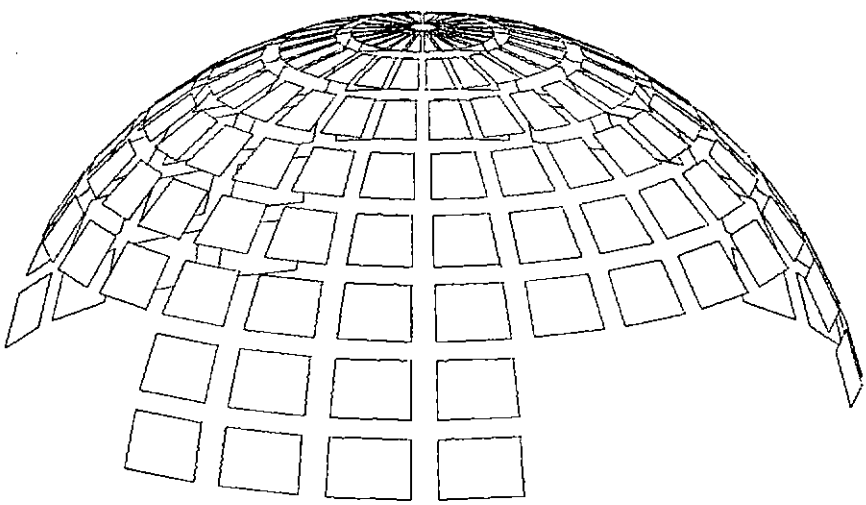


FIG VII.23

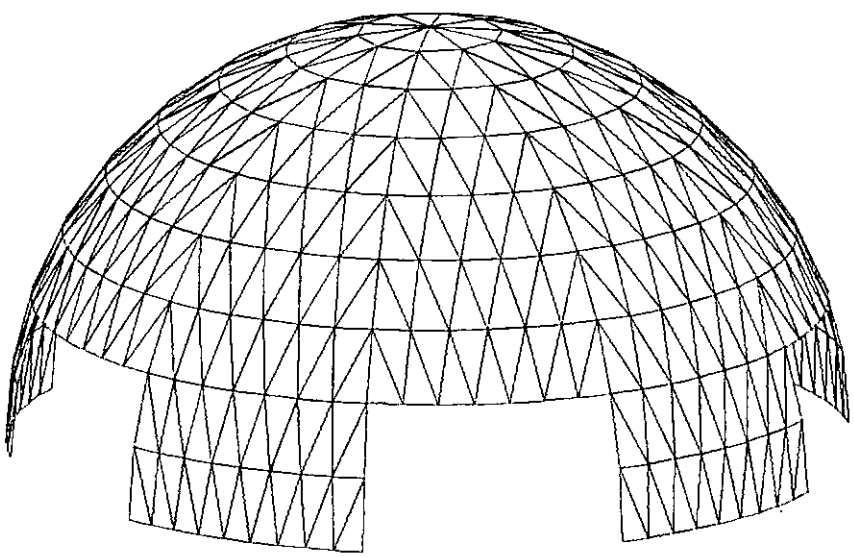


FIG VII.24.

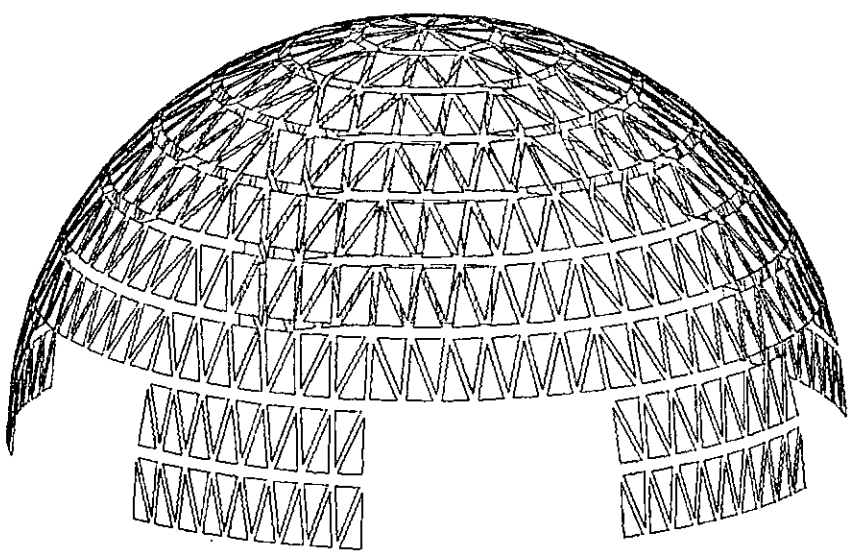


FIG VII.25.

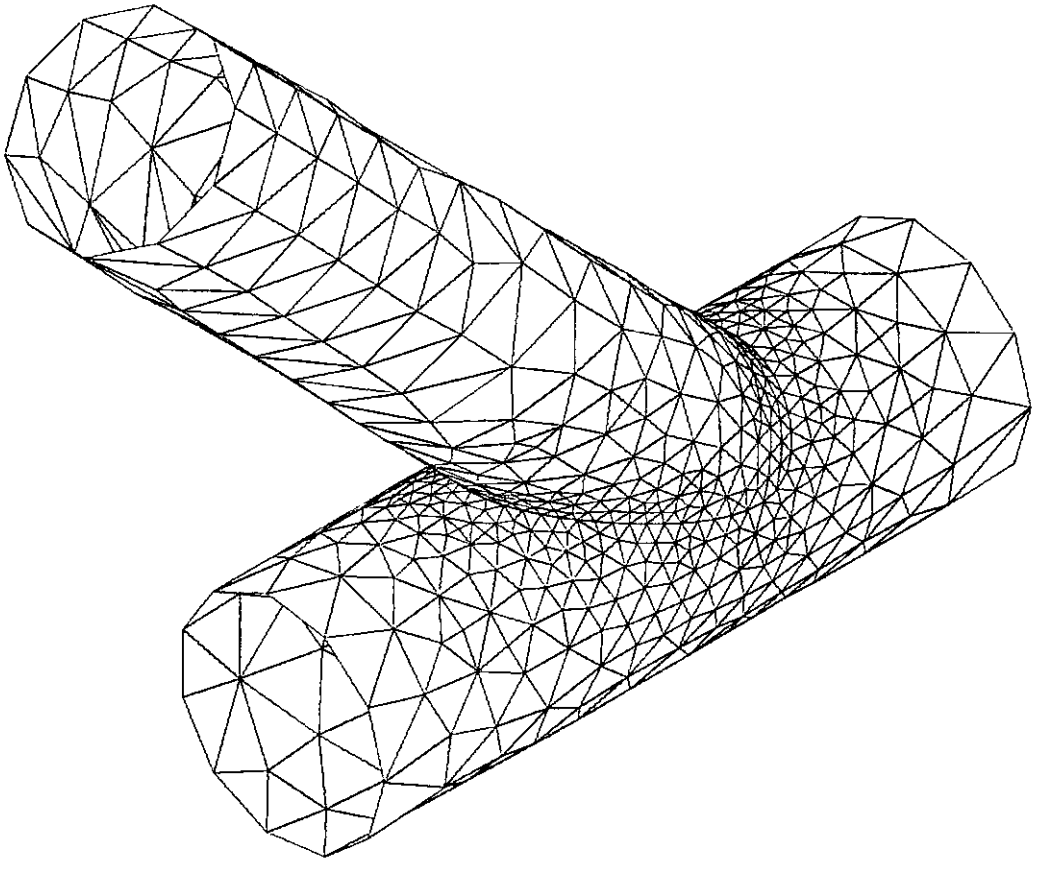


FIG VII.26.

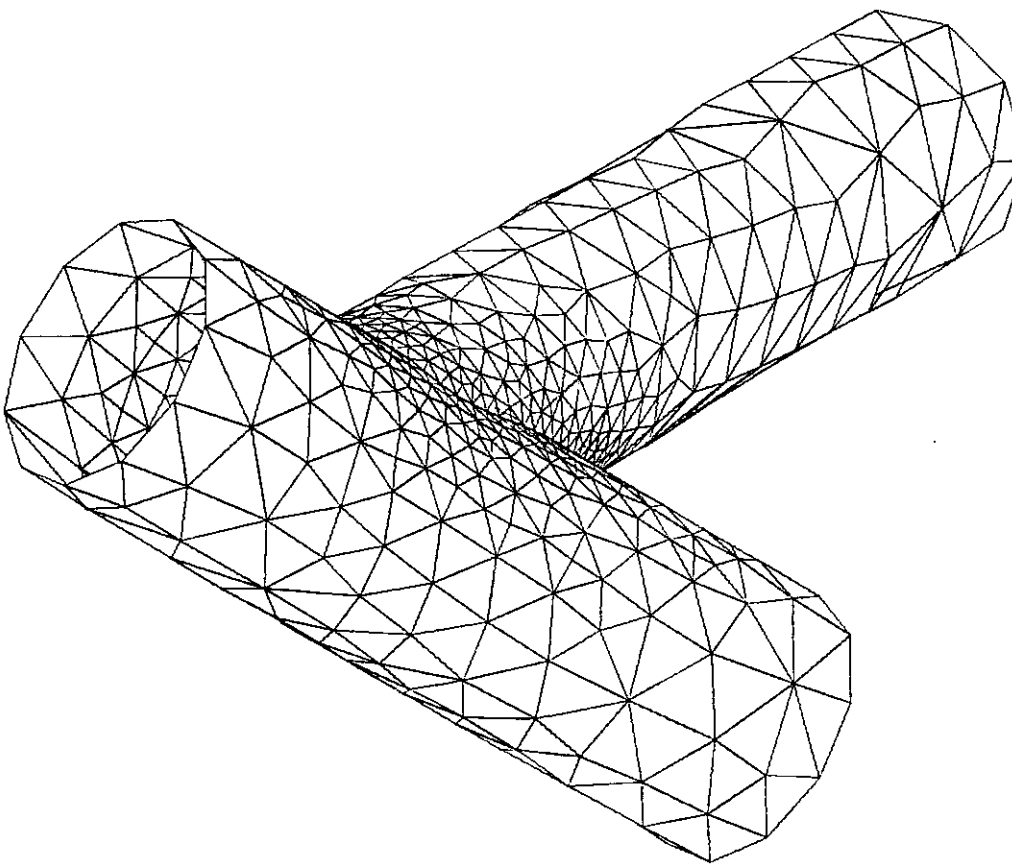


FIG VII.27.

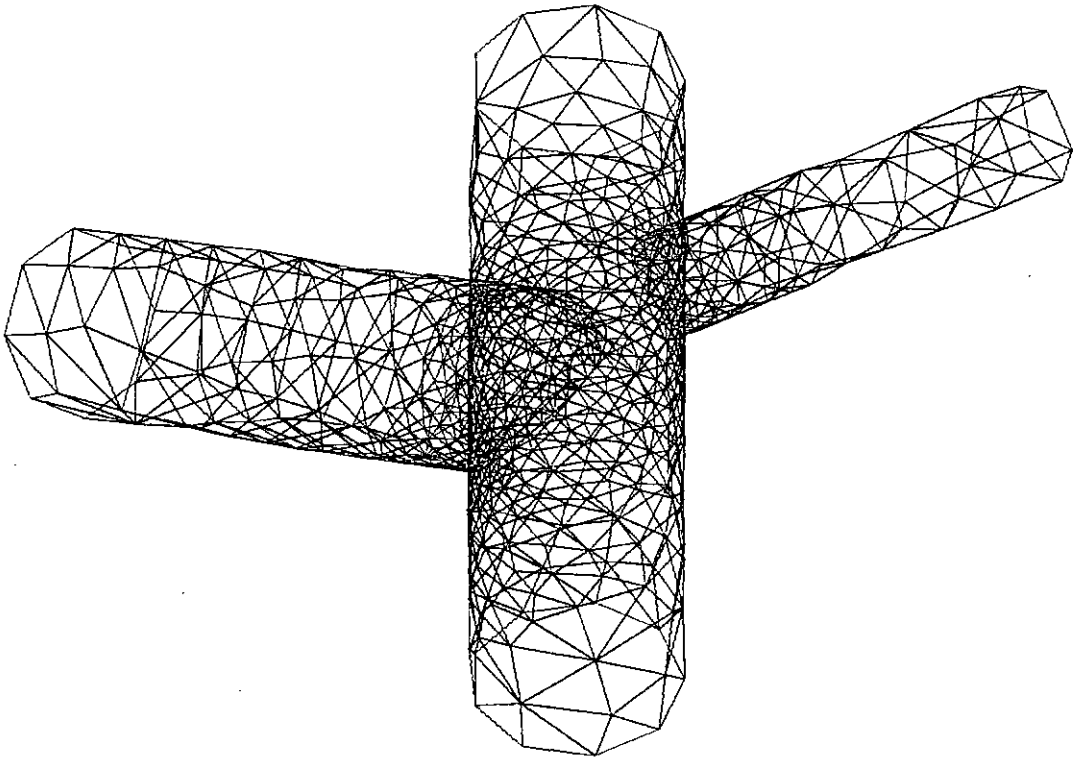


FIG VII.28.

VII.2. APLICAÇÕES PRÁTICAS 2D:

Neste conjunto de exemplos estão agrupados os modelos tridimensionais de aplicação prática. Alguns desses modelos foram totalmente gerados com o auxílio do sistema **AutoMESH**. Outros foram construídos a partir de arquivos de dados gerados por outros pré–processadores para visualização, plotagem, edição e pós–processamento.

A apresentação deste conjunto de exemplos tem por objetivo demonstrar a utilização do sistema **AutoMESH** na geração de malhas de elementos finitos para problemas reais. Poder–se–á observar a solução de problemas de cunho prático que surgem na discretização de grandes modelos ou modelos com diversas singularidades.

VII.2.1 EXEMPLO X

Este exemplo foi discretizado totalmente com a utilização do sistema **AutoMESH**. O modelo discretizado representa um maciço geológico com três minas de extração de sal. A malha obtida (figura VII.29) foi utilizada para análise de instabilidade.

A malha gerada possui algumas características específicas.

Além das singularidades que provocam concentração de tensões e exigem um refinamento da malha, o maciço é ainda constituído de diversas camadas de materiais. Deve–se notar também, que foram utilizados diversos tipos de elementos. No contorno da malha podem ser encontrados elementos infinitos e no interior, além dos quadriláteros, triangulares. Estes últimos, utilizados nas regiões de transição para gradação da malha.

A malha foi gerada pelo Eng. Lino Moreira utilizando uma versão protótipo do **AutoMESH**.

VII.2.2. **EXEMPLO XI**

Neste exemplo é apresentada a malha de elementos finitos construída no AutoMESH a partir de um arquivo de dados gerado através do sistema FEMS [179] pelo Eng. Lino Moreira para o Grupo de Geotecnia da PETROBRÁS/DIPREX (figura VII.30)

O modelo foi utilizado para a análise de instabilidade em minas de extração de sal para PETROMISA S.A. . O modelo é constituído de diversas camadas de materiais. Foram utilizados elementos quadriláteros, triangulares e infinitos. A região mais densamente discretizada corresponde ao local onde as escavações da mina avançam.

O modelo carregado no AutoMESH pode agora ser visualizado, plotado, editado ou até mesmo pós–processado.

VII.2.3. **EXEMPLO XII**

Neste exemplo é apresentada a malha de elementos finitos construída no **AutoMESH** a partir de um arquivo de dados gerado através do sistema FEMS [179] pelo Eng. Lino Moreira (figura VII.31)

O modelo representa a malha gerada para uma seção axissimétrica de uma estaca de ponta fechada e foi utilizado para análise de ruptura.

A figura VII.32 apresenta o pós–processamento da análise no sistema **AutoMESH** através do comando POST opção COLORMAP.

Infelizmente, os equipamentos de reprodução não permitem a visualização da distribuição das cores segundo as tensões no modelo.

VII.2.4. EXEMPLO XIII

Neste exemplo é apresentada uma malha de elementos finitos construída no **AutoMESH** a partir de um arquivo de dados gerado no sistema IRM [185] e transferida no FORMATO GENÉRICO [190].

A figura VII.33 apresenta a malha constituída de elementos quadriláteros, infinitos e de contato, utilizada para a análise de cravabilidade de uma estaca no solo. Cada camada horizontal de elementos corresponde a um tipo de material (solo).

A malha apresentada corresponde a uma seção axissimétrica do modelo.

A figura VII.34 apresenta um detalhe da ponta do modelo onde pode–se visualizar o solo, os elementos de contato e a estaca.

VII.2.5. **EXEMPLO XIV**

Neste exemplo é apresentada uma malha de elementos finitos construída no AutoMESH a partir de um arquivo de dados gerado no sistema IRM [185] e transferida no FORMATO GENÉRICO [190].

A figura VII.35 apresenta a malha para uma seção axissimétrica de conector de *riser* à cabeça de poço.

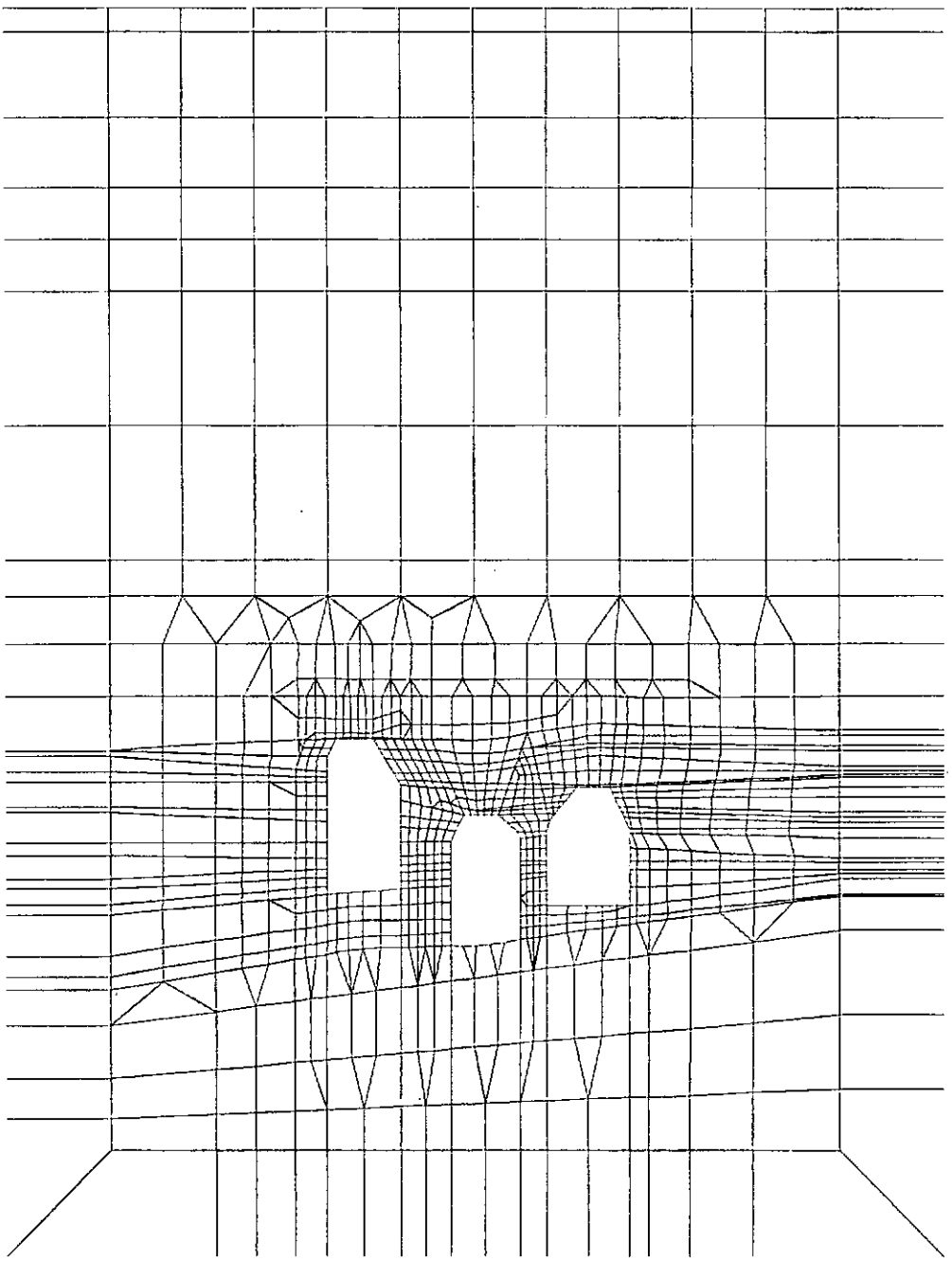


FIG VII.29.

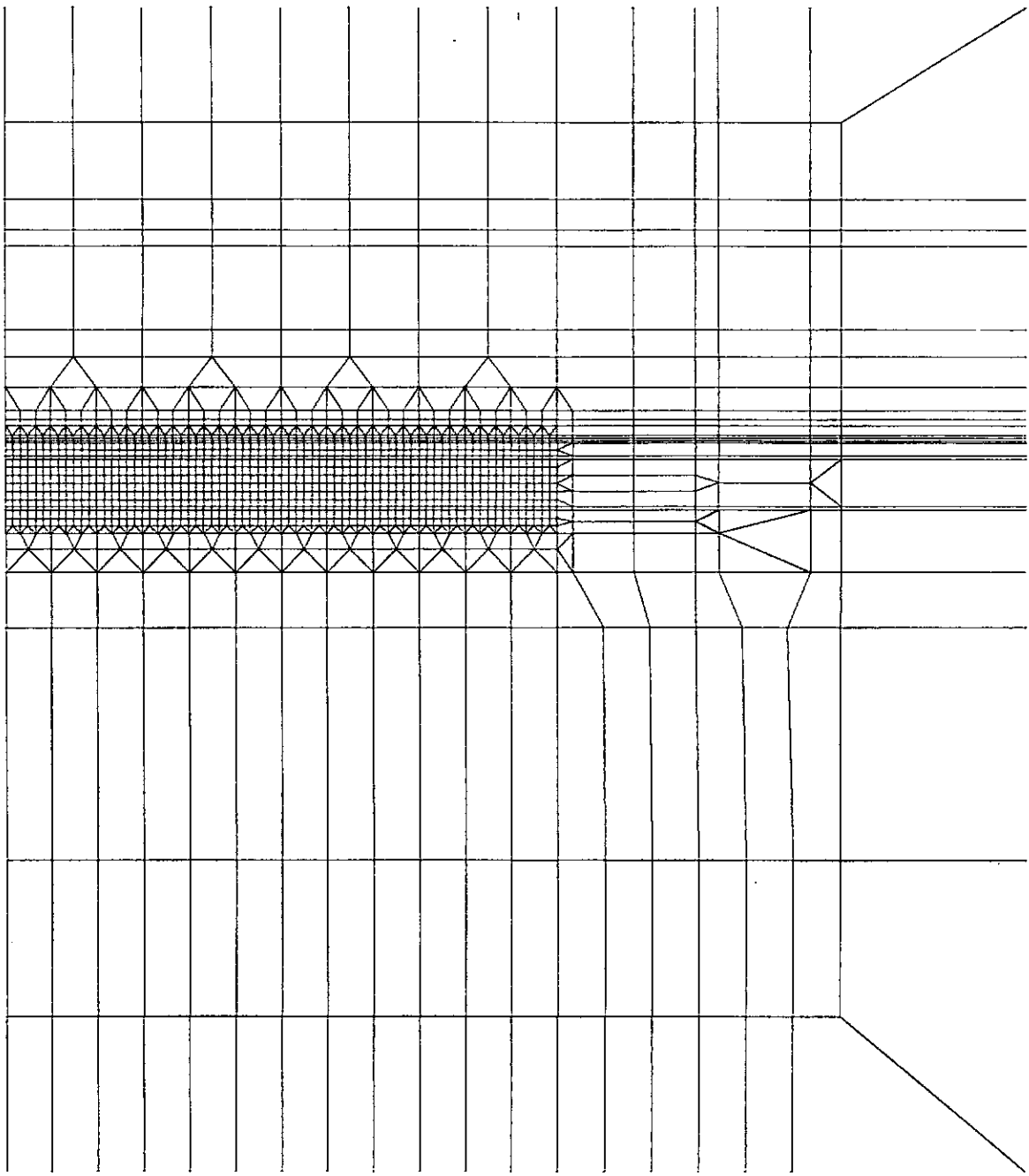


FIG VII.30.

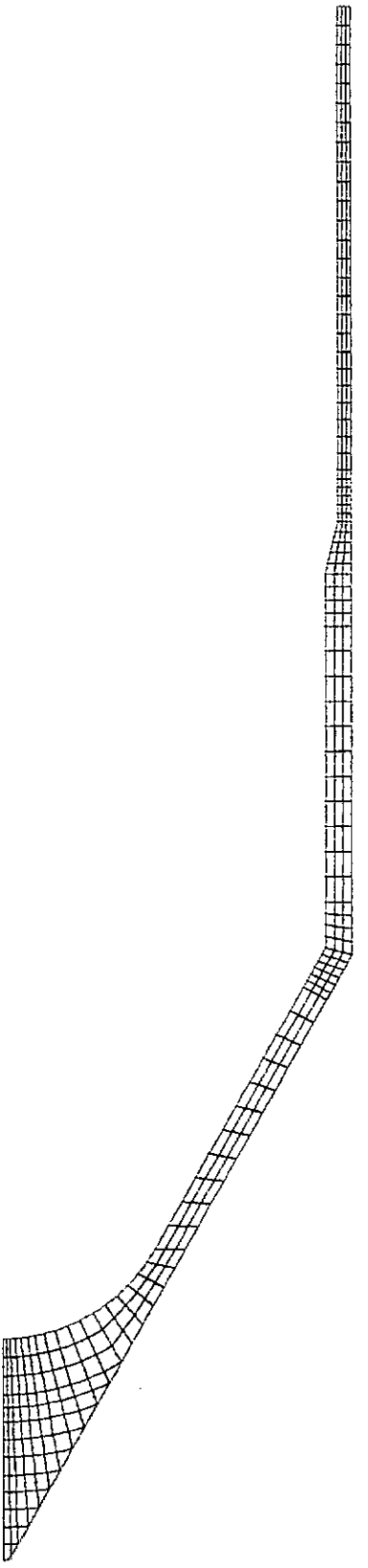


FIG VII.31.

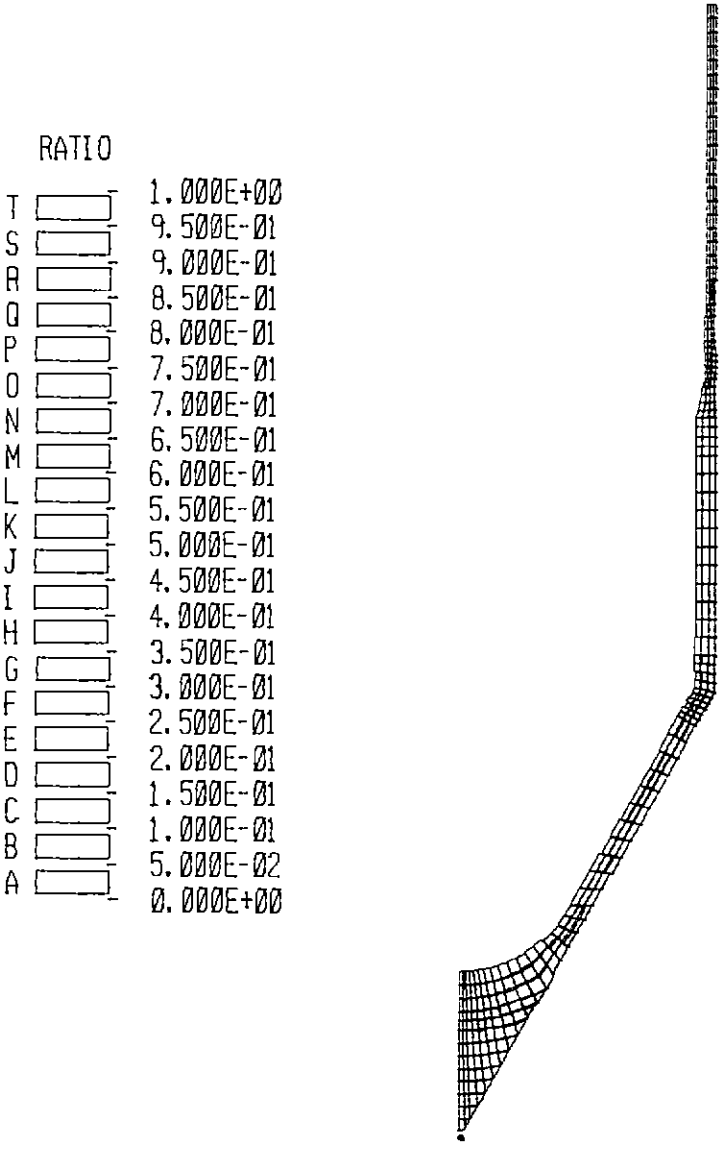

RATIO
T 1.000E+00
S 9.500E-01
R 9.000E-01
Q 8.500E-01
P 8.000E-01
O 7.500E-01
N 7.000E-01
M 6.500E-01
L 6.000E-01
K 5.500E-01
J 5.000E-01
I 4.500E-01
H 4.000E-01
G 3.500E-01
F 3.000E-01
E 2.500E-01
D 2.000E-01
C 1.500E-01
B 1.000E-01
A 5.000E-02
0.000E+00

FIG VII.32.

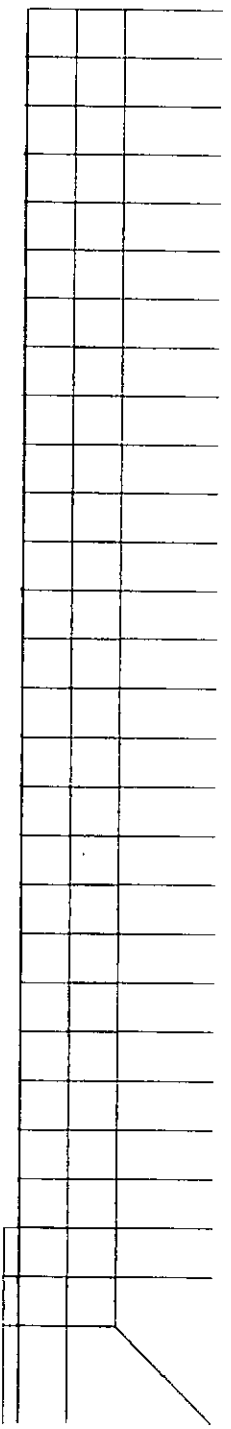


FIG VII.33.

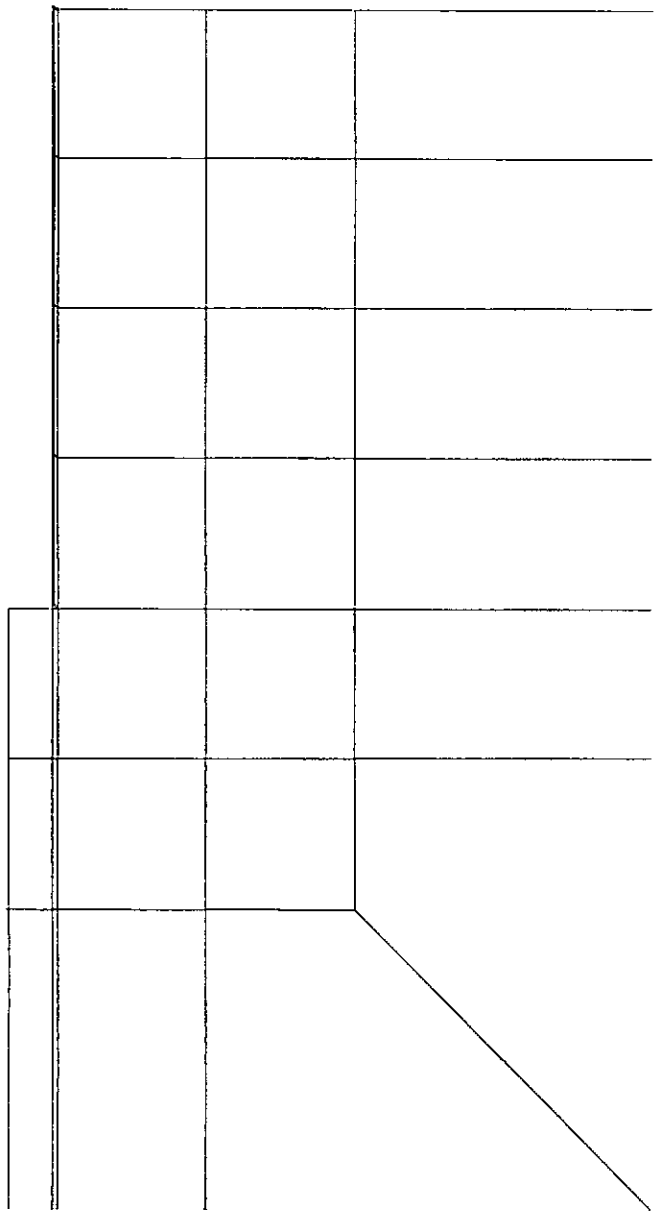


FIG VII.34.

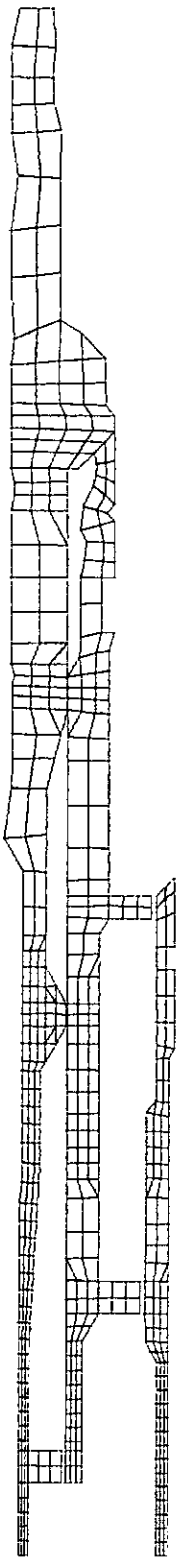


FIG VII.35.

VII.3. MODELOS RETICULADOS

Este grupo de exemplos está voltado à aplicações e modelos reticulados, ou que utilizem elementos finitos do tipo reticulado.

Estes exemplos foram agrupados em um conjunto isolado devido ao fato de existirem interesses específicos por parte dos projetistas para a definição de modelos que utilizem elementos finitos do tipo reticulado. Existe ainda, o fato dos elementos reticulados constituírem um sub–grupo dos elementos finitos e existirem comandos dentro do **AutoMESH** dirigido especificamente ao tratamento destes.

VII.3.1. EXEMPLO XV

Este exemplo visa demontrar a utilização do comando SECTIONS do **AutoMESH**, além da importância do comando HIDE do **AutoCAD**.

A figura VII.36 apresenta um nó de estrutura onde incidem sobre este, quatro elementos reticulados com propriedades de seção transversal do tipo I, H ou duplo T.

A visualização das seções transversais é possibilitada pela utilização do comando SECTIONS do **AutoMESH**. Este exemplo serve também, para mostrar que o uso indiscriminado desse comando, além de consumir muito espaço em disco e tornar o sistema mais lento, pode tornar o modelo difícil de visualizar ou entender.

As figuras VII.37 e VII.38 apresentam a aplicação do comando HIDE a duas vistas distintas da malha.

VII.3.2. EXEMPLO XVI

Na figura VII.39 apresenta–se a malha de elementos finitos para um silo metálico.

Na discretização foram utilizados elementos planos e reticulados. Isto serve como demonstração da não existência de restrições quanto a coexistência de tipos distintos de elementos finitos.

A figura VII.40 apresenta uma vista isométrica do modelo com remoção das linhas ocultas.

As figuras VII.41, VII.42 e VII.43 apresentam diversas vistas para a malha.

A figura VII.44 apresenta os carregamentos nodais aplicados ao modelo.

VII.3.3. EXEMPLO XVII

Este modelo representa uma estrutura reticulada do tipo pórtico espacial (figura VII.45)

Na figura VII.46 são apresentados os carregamentos aplicados aos membros. Pode–se observar os carregamentos uniformemente distribuídos aplicados à colunas, carregamentos linearmente distribuídos aplicados à vigas e carregamentos nodais.

As figuras VII.47, VII.48 e VII.49 apresentam diversas vistas do modelo.

VII.3.4. EXEMPLO XVIII

Neste exemplo apresenta–se o modelo de uma torre de transmissão.

A malha é gerada a partir do eixo de simetria vertical para metade da torre, como mostrado nas figuras VII.50 e VII.51. Os elementos foram inseridos no modelo um a um, através do comando ELEMENT. A malha como um todo é obtida pela aplicação do comando MIRROR do **AutoMESH** em relação ao eixo de simetria.

As figuras VII.52, VII.53, VII.54 e VII.55 apresentam diversas vistas da malha completa.

Neste exemplo são apresentados também alguns recursos de pós–processamento para estruturas reticuladas, em especial deformadas.

As figuras que se seguem apresentam os resultados (deformadas) para uma análise dinâmica onde determinou–se os modos de vibração. As figuras apresentam a malha original em contraste com a deformada.

As figuras VII.56, VII.57, VII.58 e VII.59 apresentam diversas vistas para o pós–processamento relativo ao primeiro modo de vibração.

As figuras VII.60, VII.61, VII.62 e VII.63 correspondem ao segundo modo de vibração.

As figuras VII.64, VII.65, VII.66 e VII.67 correspondem ao terceiro modo de vibração.

O quarto modo de vibração pode ser visualizado através das figuras VII.68, VII.69, VII.70 e VII.71.

O quinto modo de vibração pode ser visualizado através das figuras VII.72, VII.73, VII.74 e VII.75.

Por fim, o sexto modo de vibração está representado nas figuras VII.76, VII.77, VII.78, VII.79.

As deformadas foram construídas através do comando POST opção DEFORM, utilizando–se um fator de amplificação de 10.000.

VII.3.5. EXEMPLO XIX

As figuras VII.80, VII.81, VII.82 e VII.83 apresentam diversas vistas da plataforma de Enchova da PETROBRÁS. Este modelo foi construído no sistema **AutoMESH** de um modo não trivial.

O desenho da plataforma foi obtido primeiramente através do sistema GEDEP [192], a partir de dados fornecidos pelo sistema ADEP[186] com o comando PLOT DEVICE INTERGRAPH. O desenho foi, então, convertido para o formato DXF compatível com o AutoCAD e transferido para um microcomputador. Com o auxílio do comando DXFIN, um desenho unifilar foi criado no **AutoCAD** e através do comando CONVERT do **AutoMESH**, as linhas foram convertidas a elementos reticulados com propriedades associadas.

VII.3.6. EXEMPLO XX

O Modelo apresentado na figura VII.84 representa a plataforma de Enchova Oeste, alternativa 1B da PETROBRÁS.

O modelo foi criado no sistema ADEP [186] e transferido para o **AutoMESH** através do formato Neutral File.

O modelo pode agora ser editado, detalhado, plotado ou até mesmo pós–processado.

As figuras VII.85, VII.86 e VII.87 apresentam outras vistas do modelo.

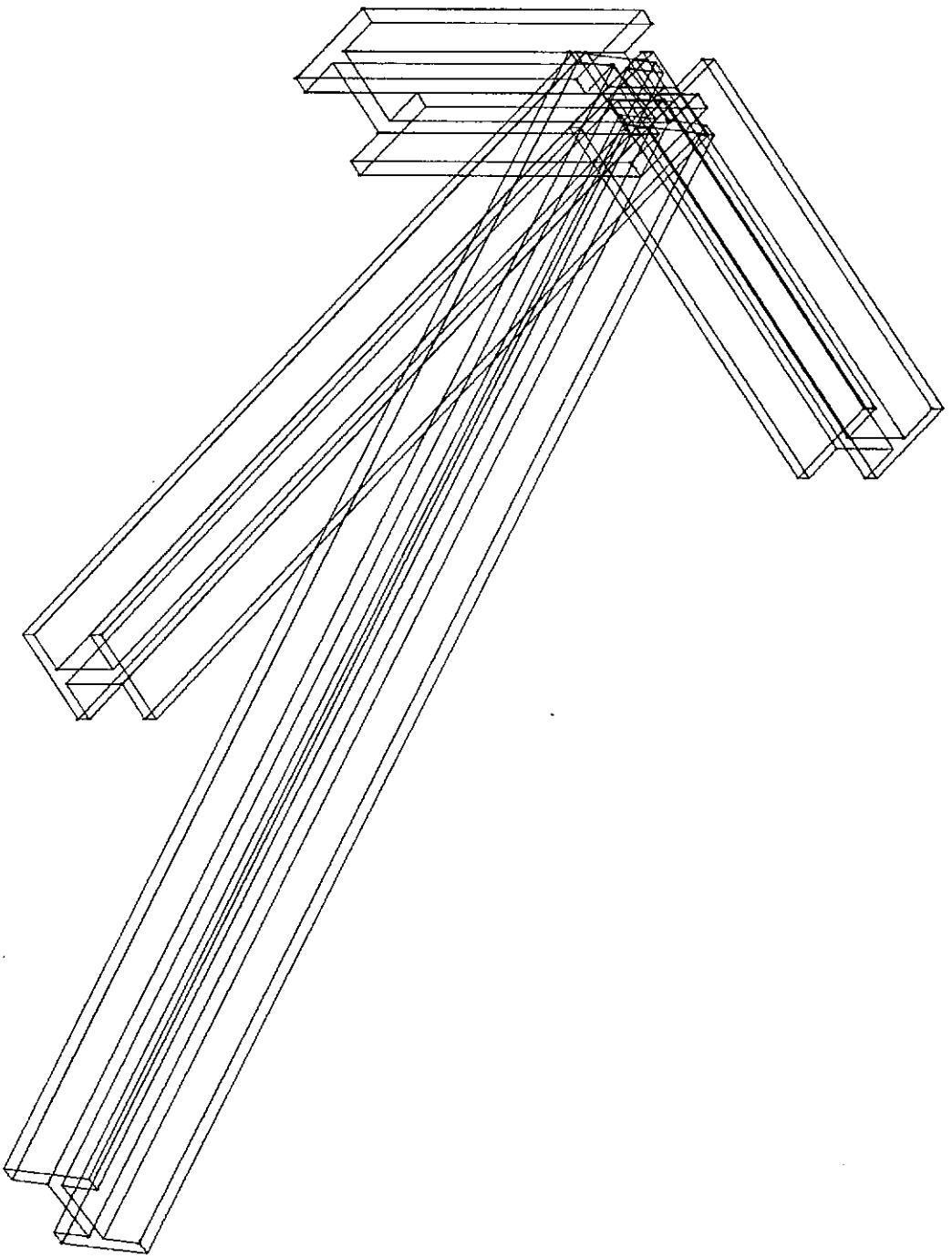


FIG VII.36.

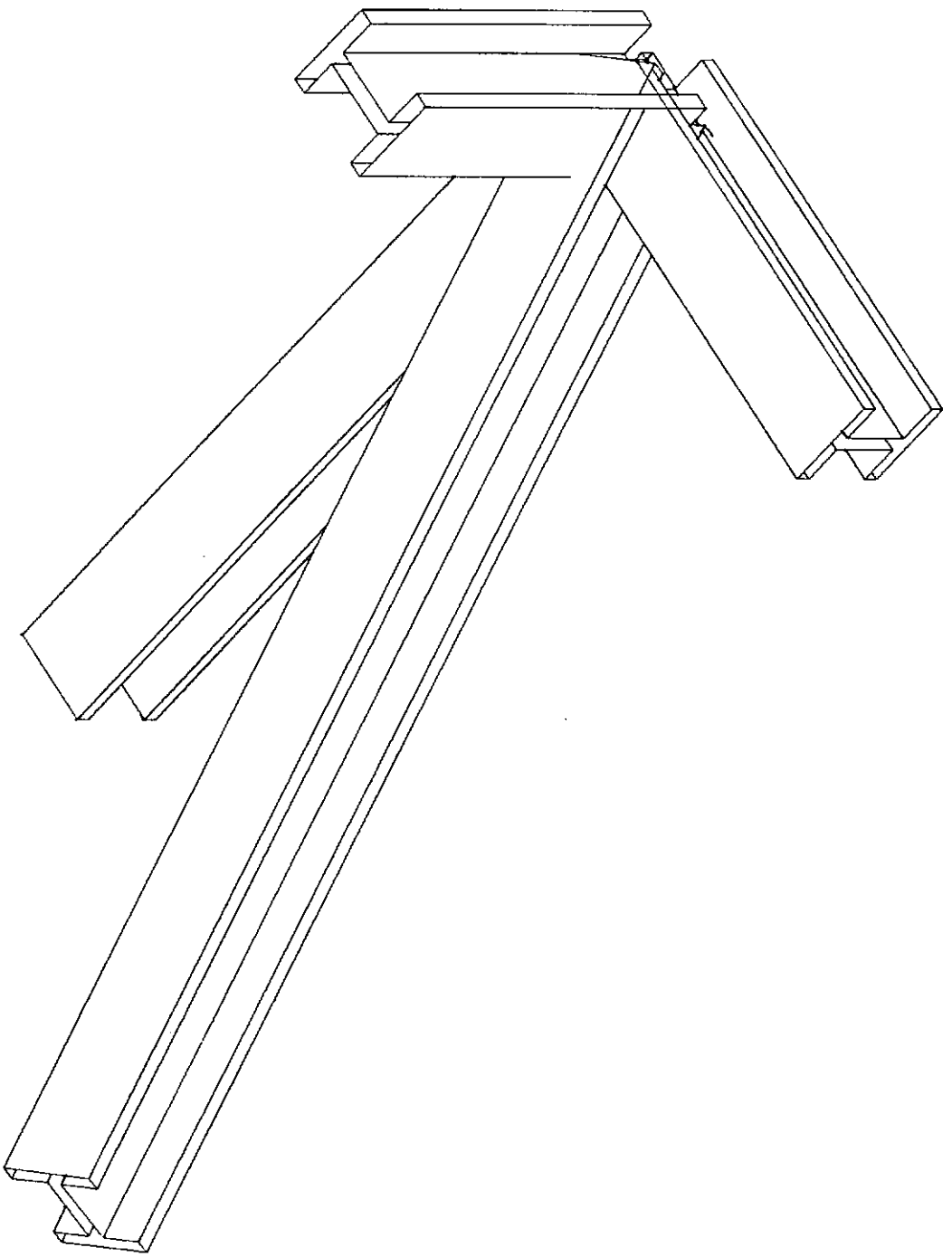


FIG VII.37.

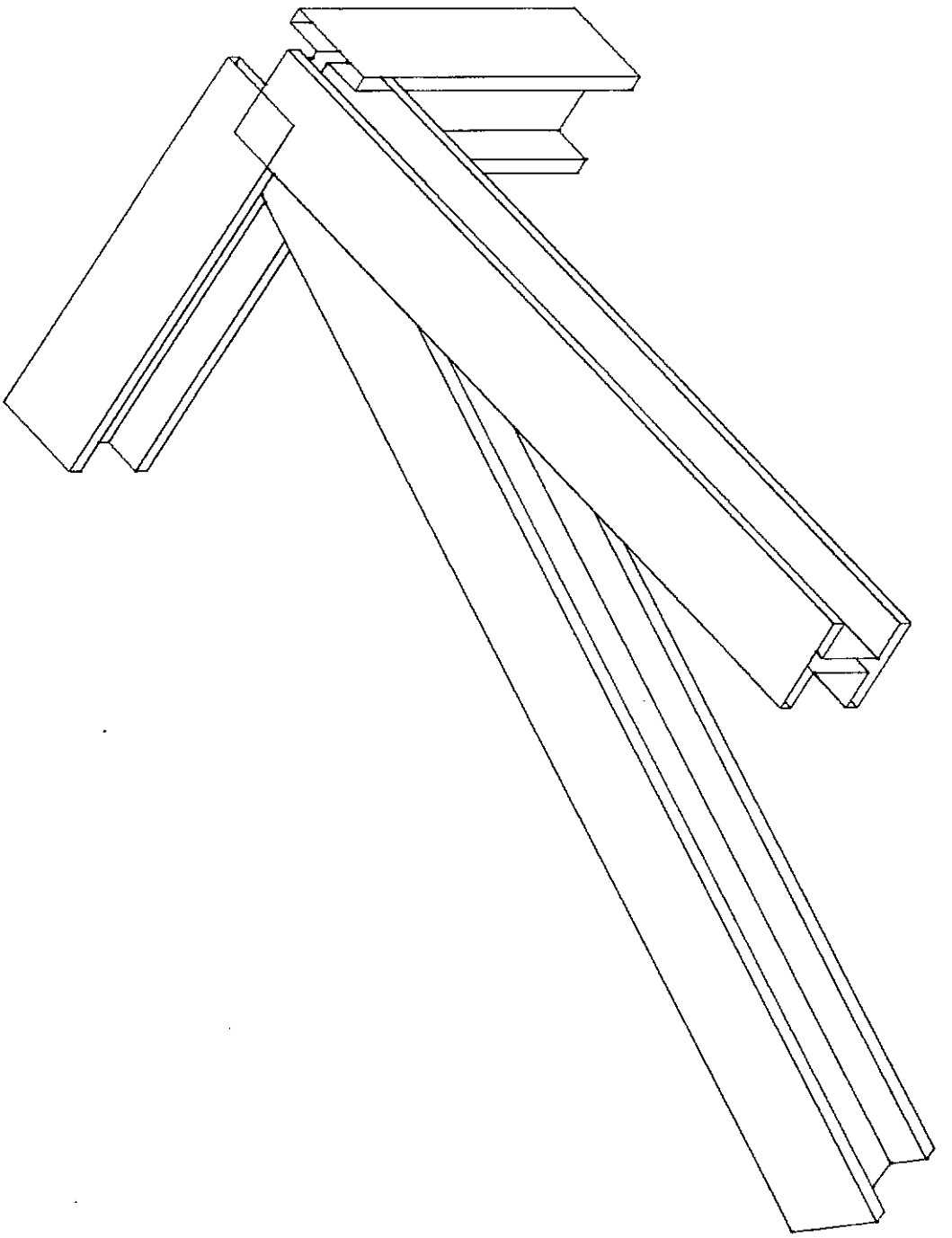


FIG VII.38.

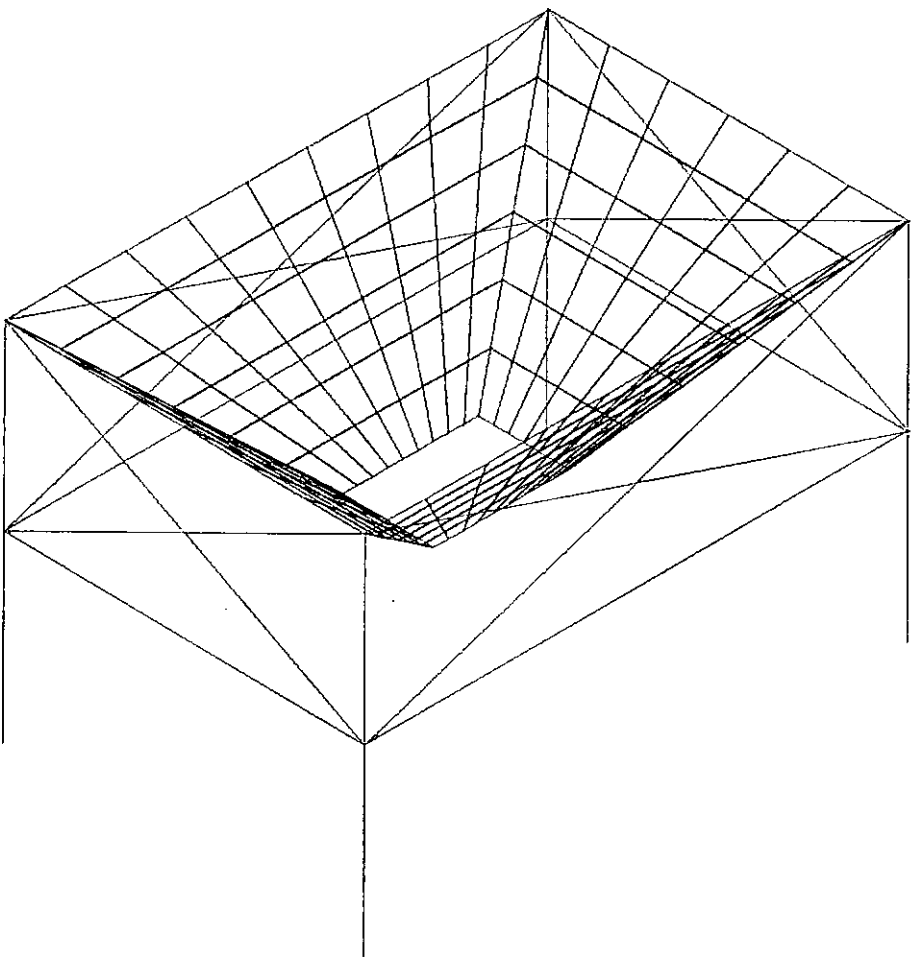


FIG VII.39.

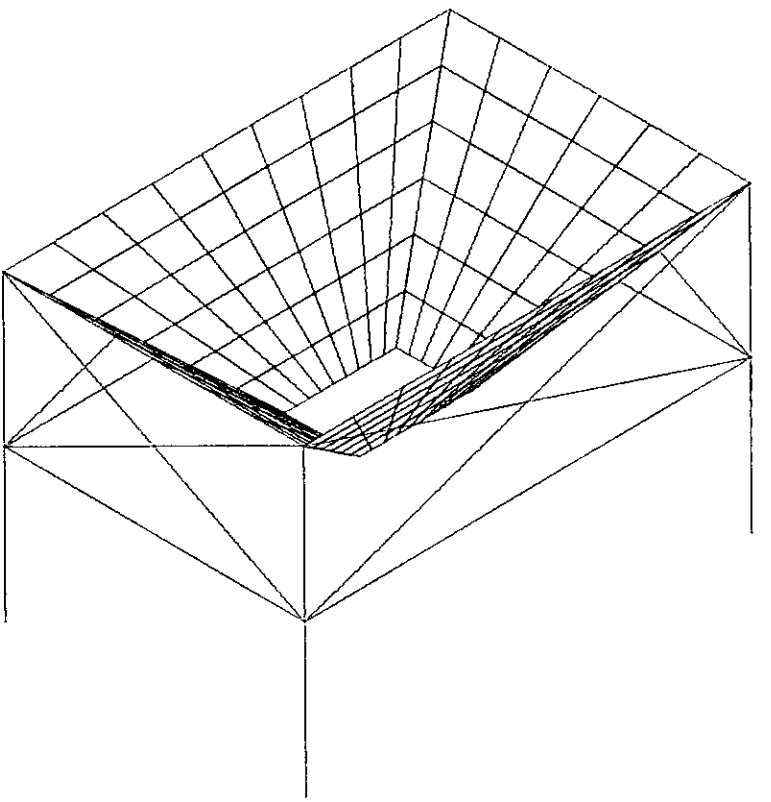


FIG VII.40.

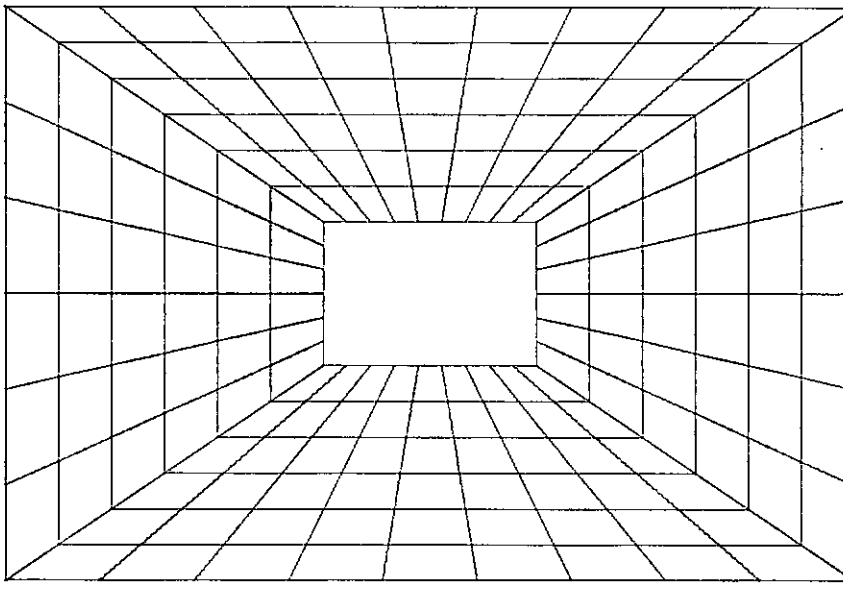


FIG VII.41.

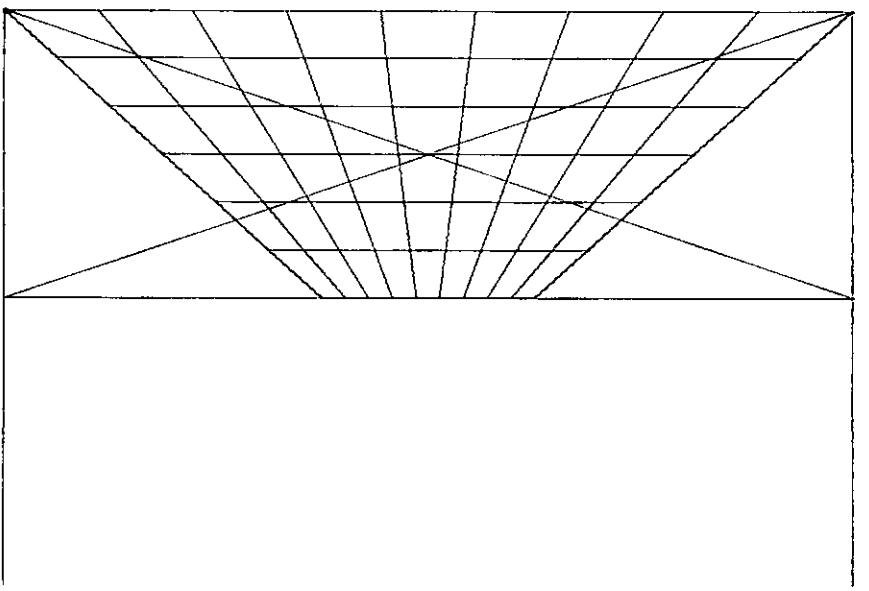


FIG VII.42.

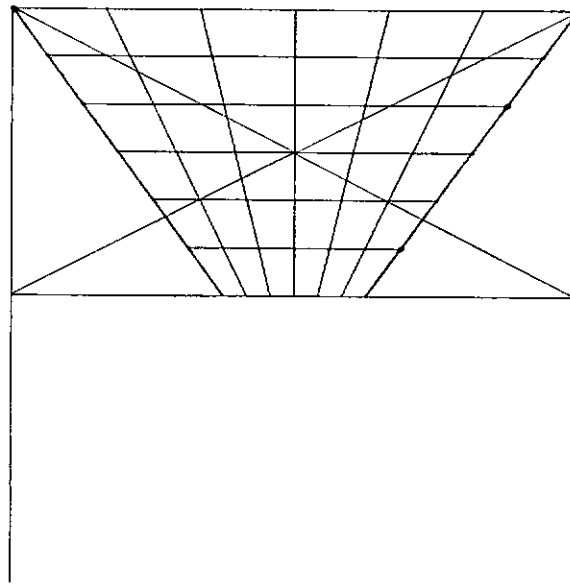


FIG VII.43.

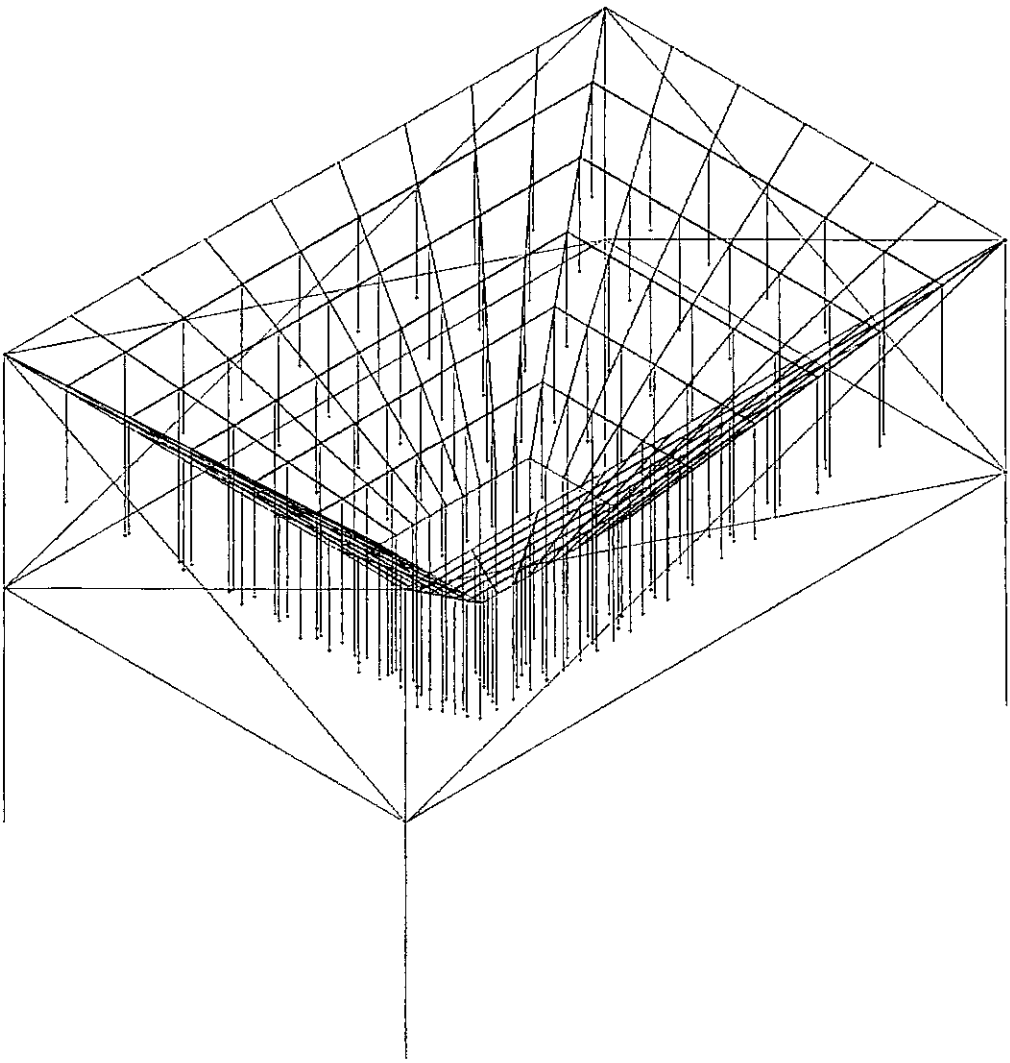


FIG VII.44.

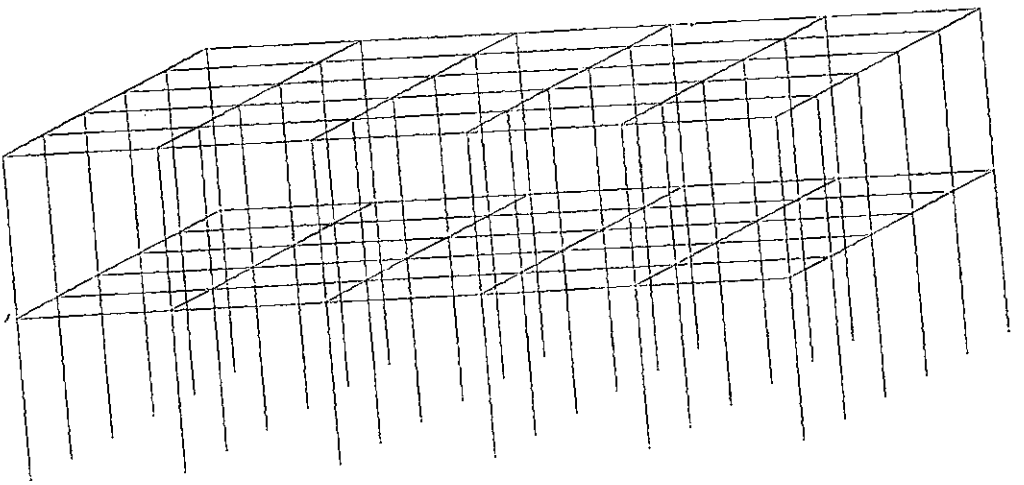


FIG VII.45.

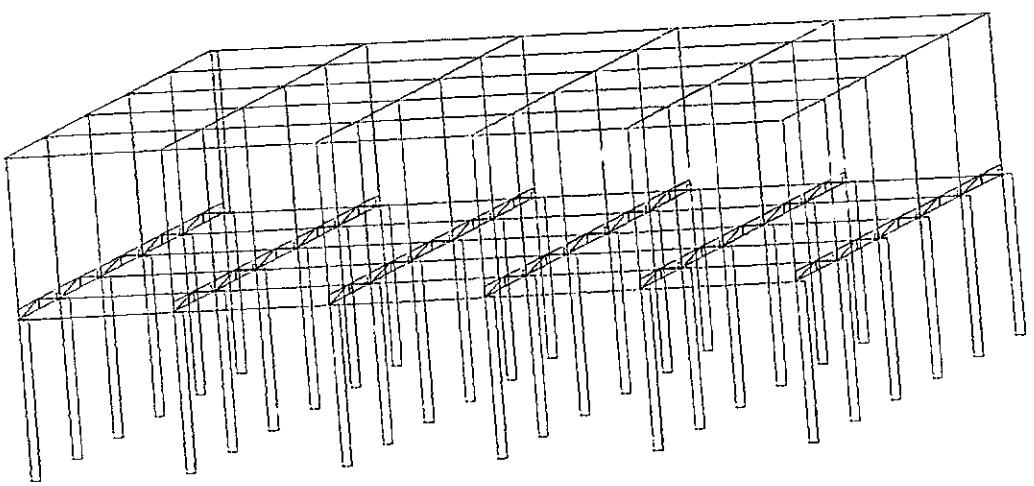


FIG VII.46.

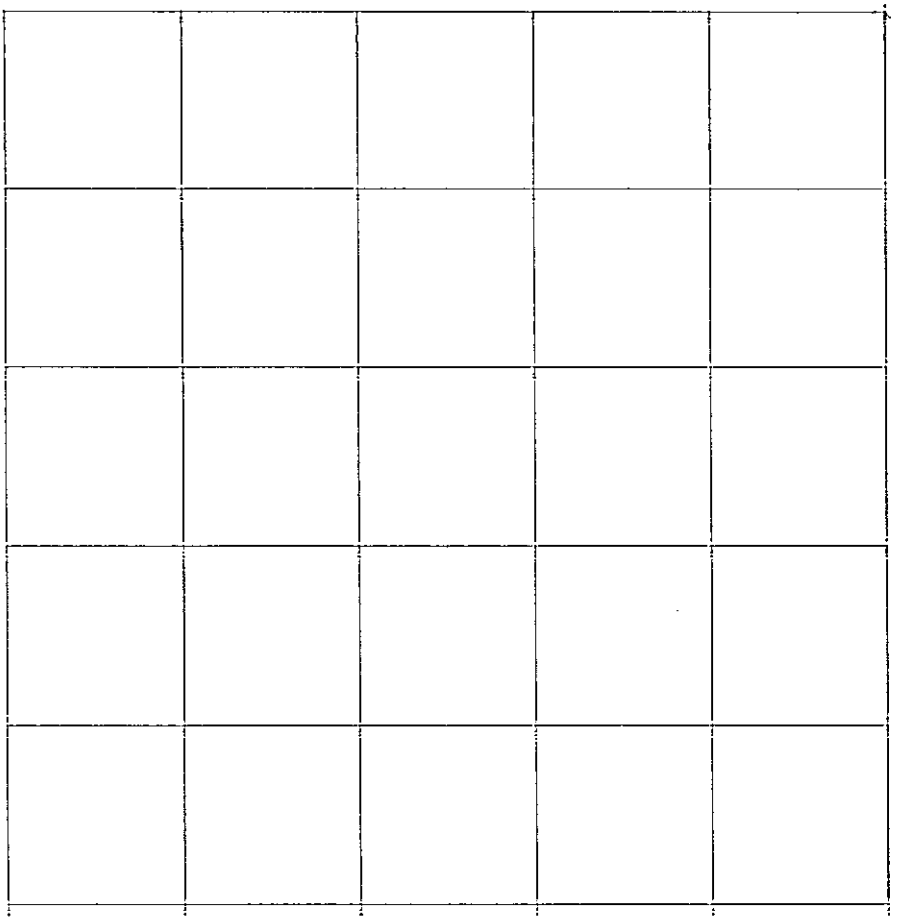


FIG VII.47.

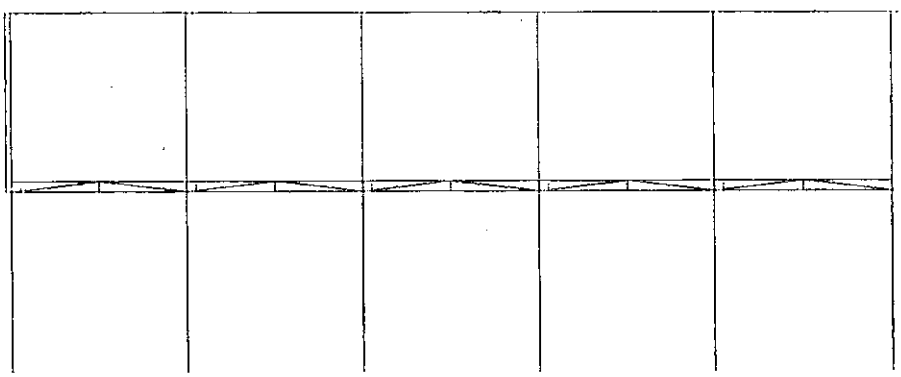


FIG VII.48.

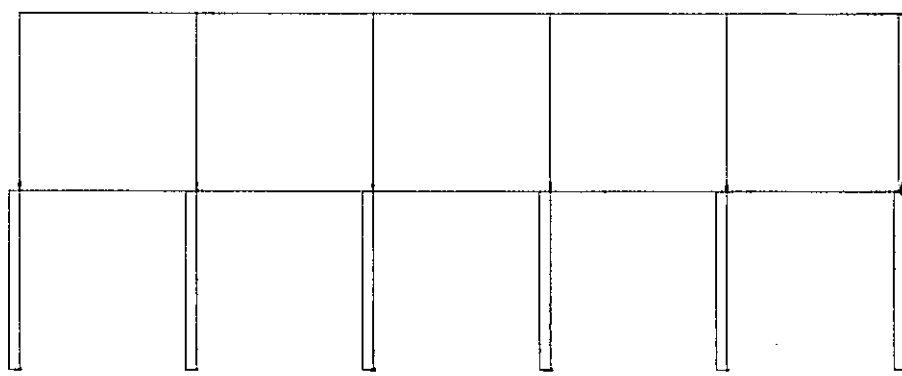


FIG VII.49.

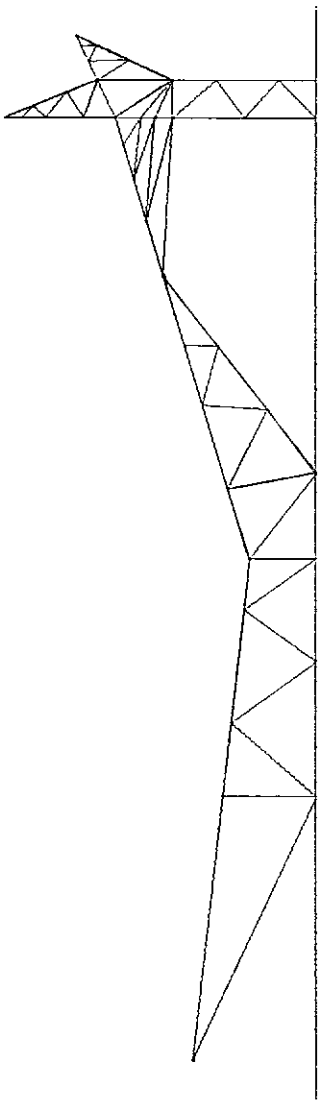


FIG VII.50.

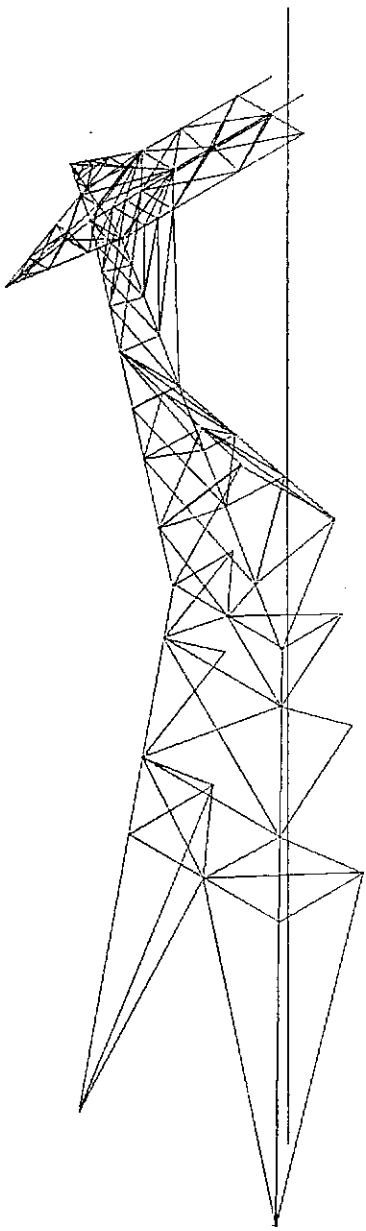


FIG VII.51.

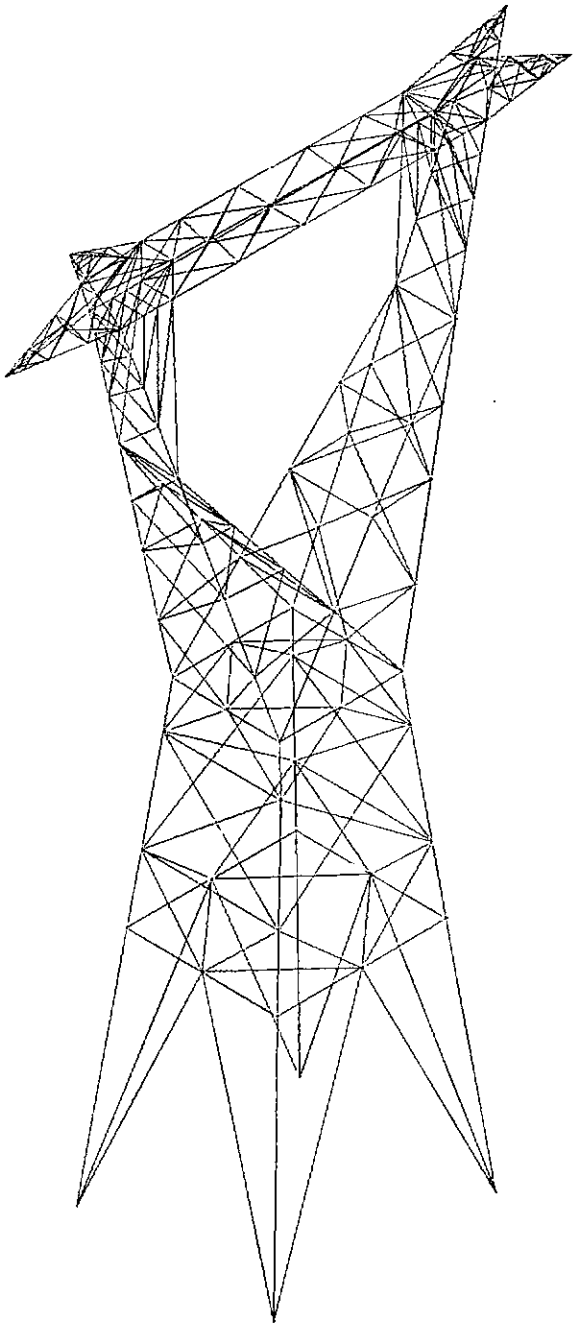


FIG VII.52.

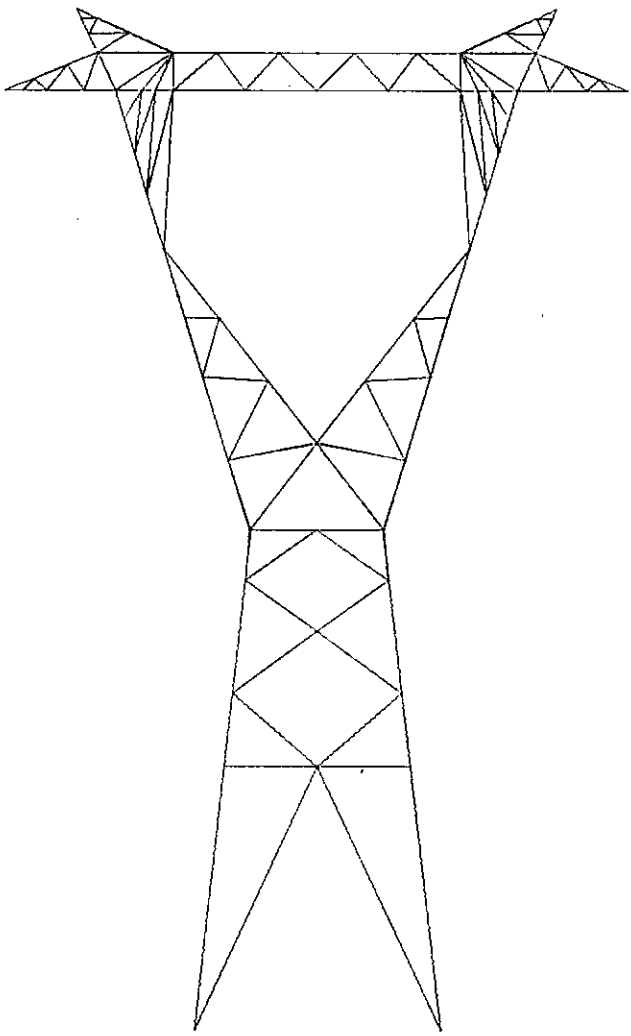


FIG VII.53.

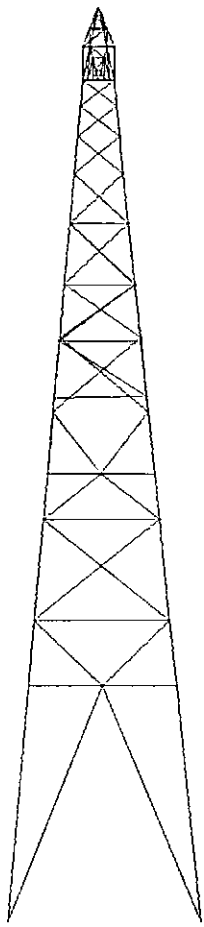


FIG VII.54.

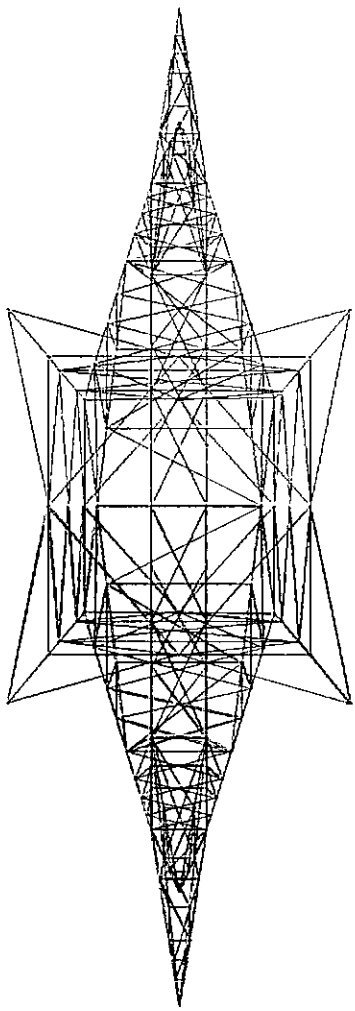


FIG VII.55.

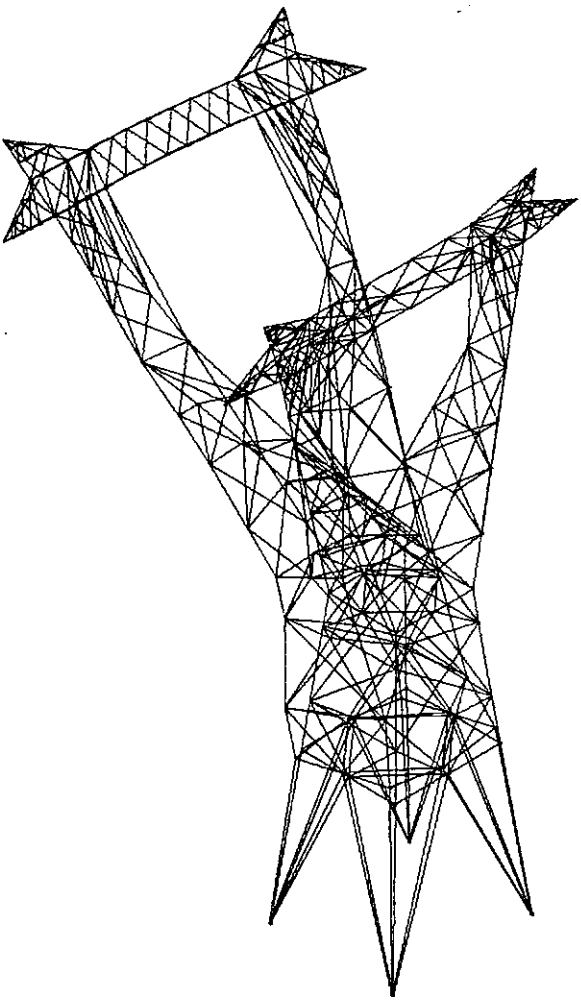


FIG VII.56.

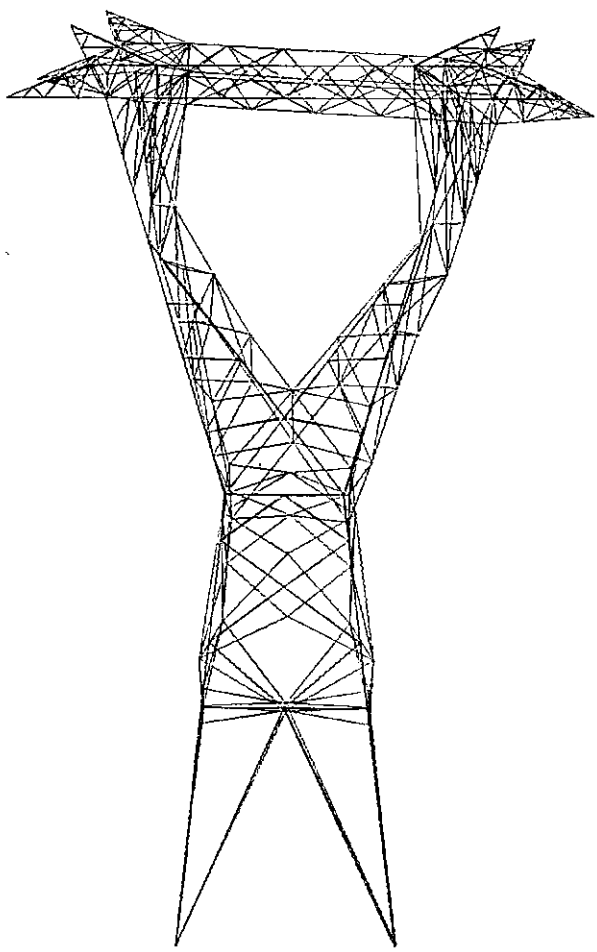


FIG VII.57.

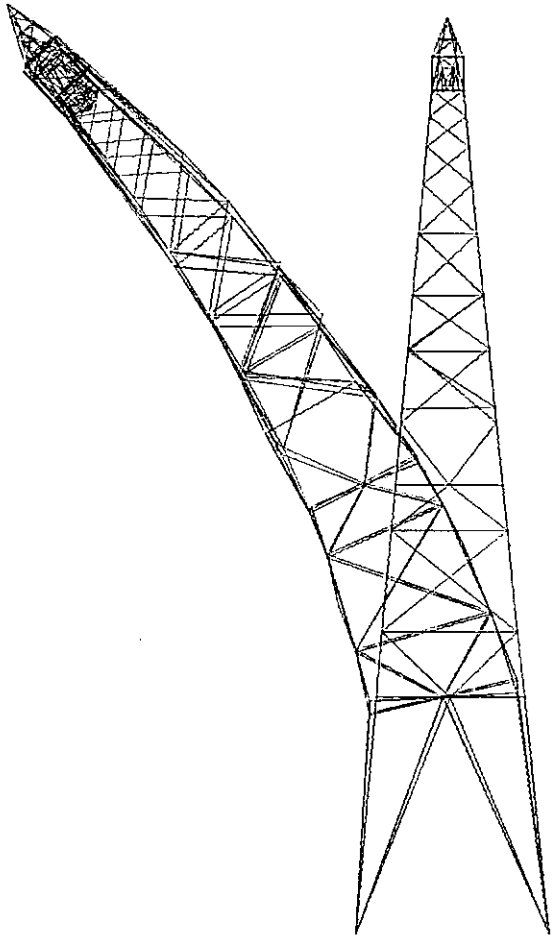


FIG VII.58.

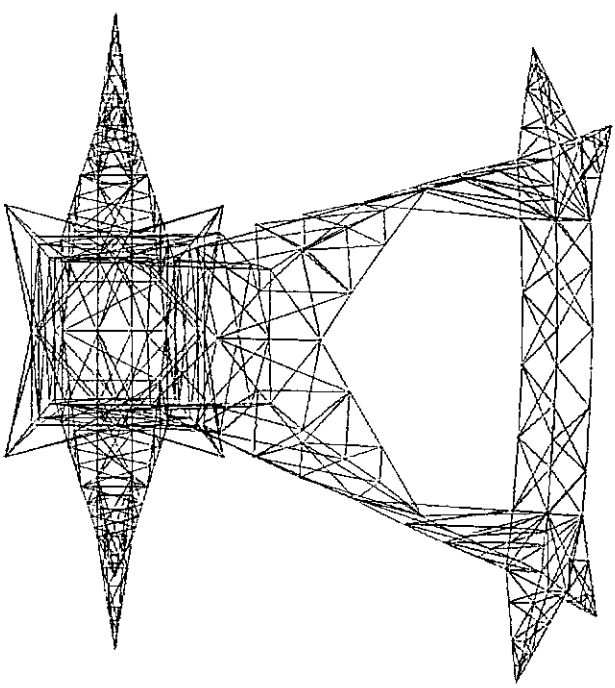


FIG VII.59

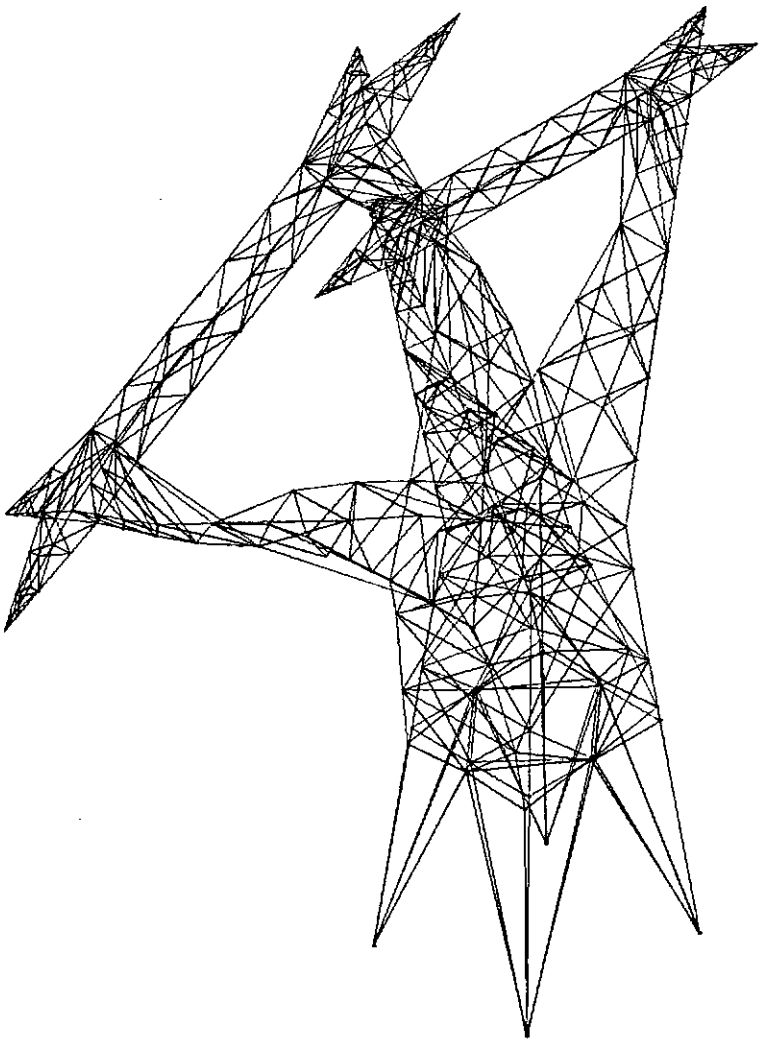


FIG VII.60.

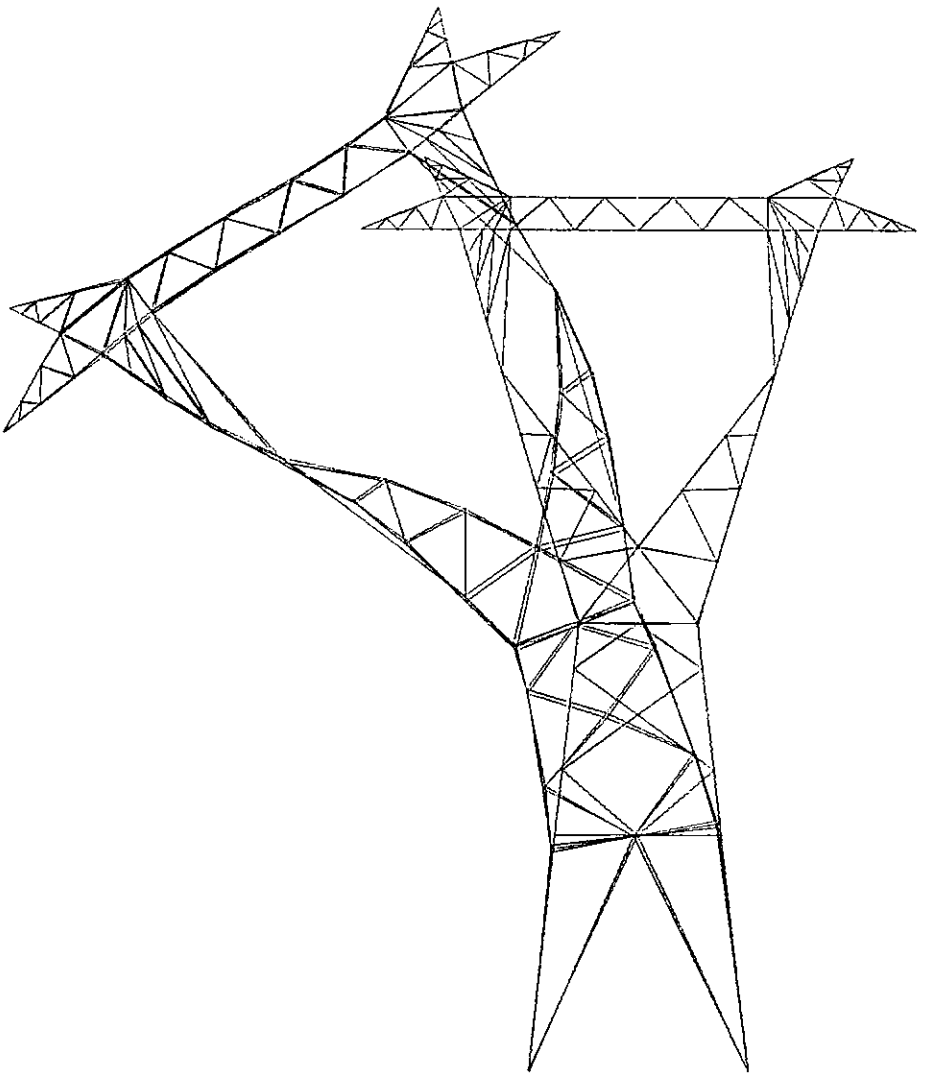


FIG VII.61.

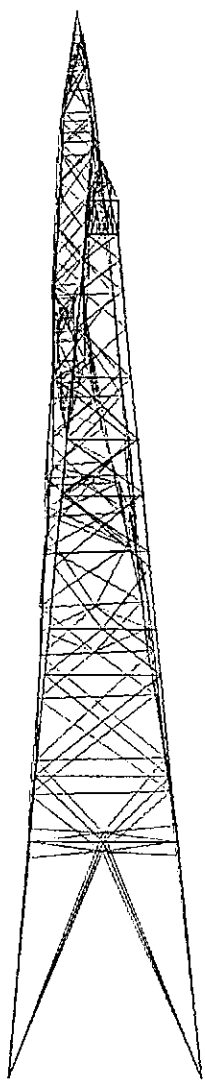


FIG VII.62.

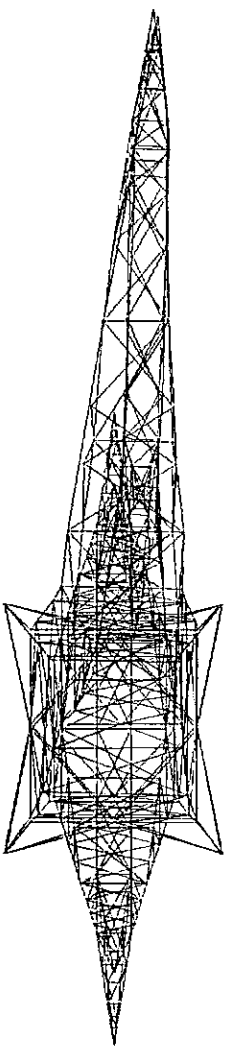


FIG VII.63.

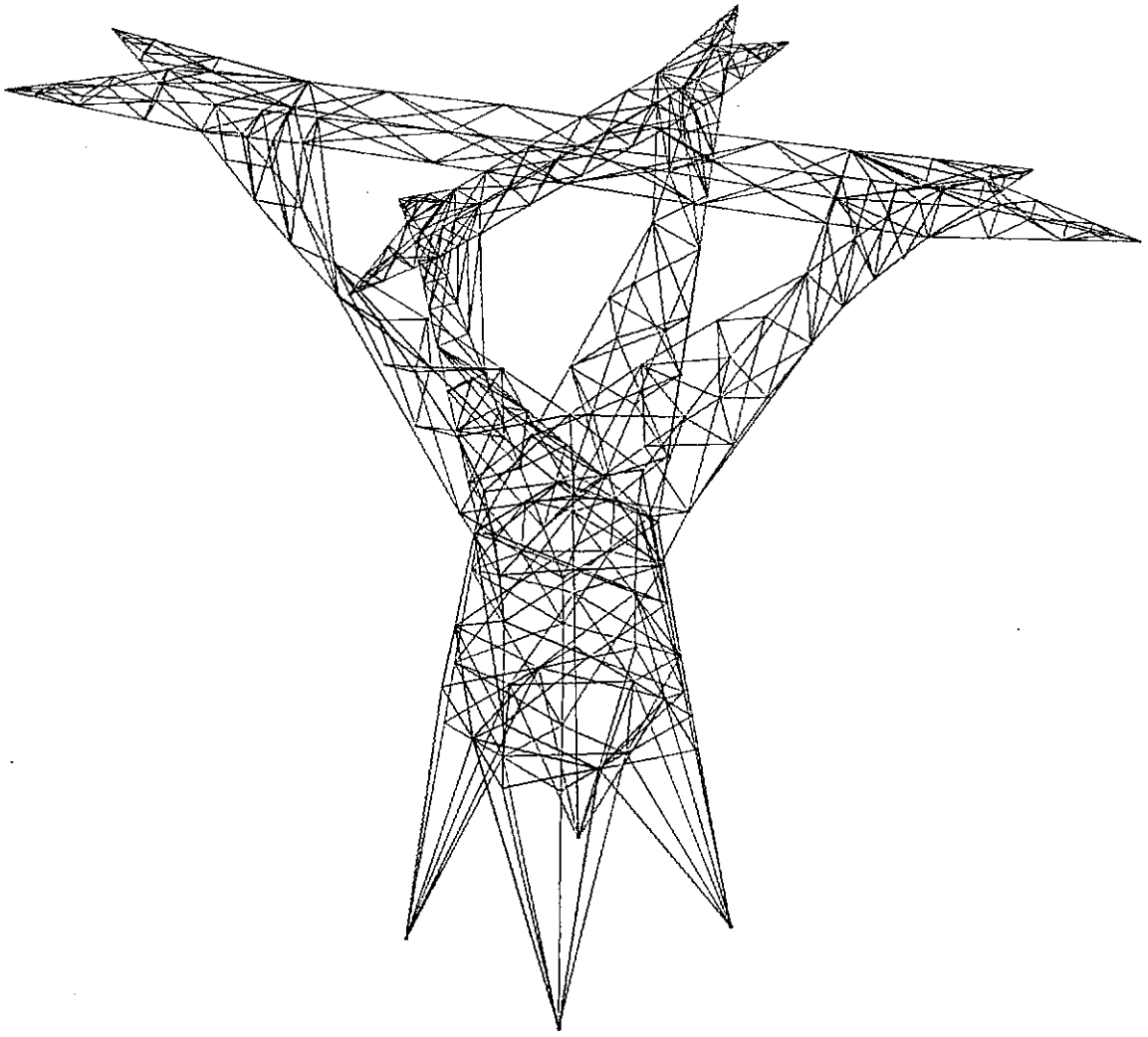


FIG VII.64.

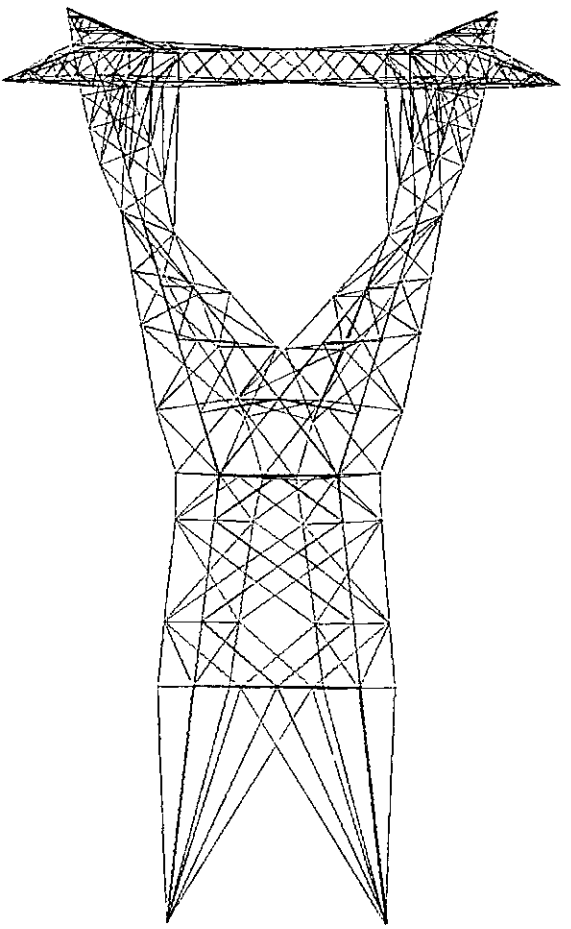


FIG VII.65.

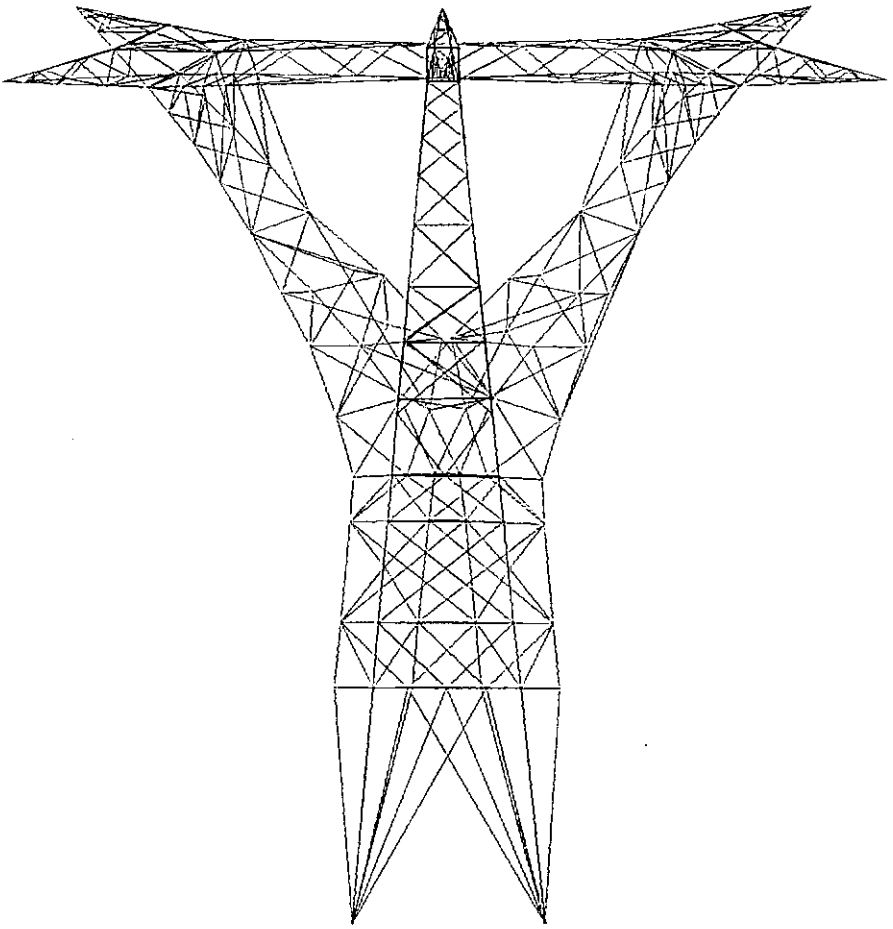


FIG VII.66.

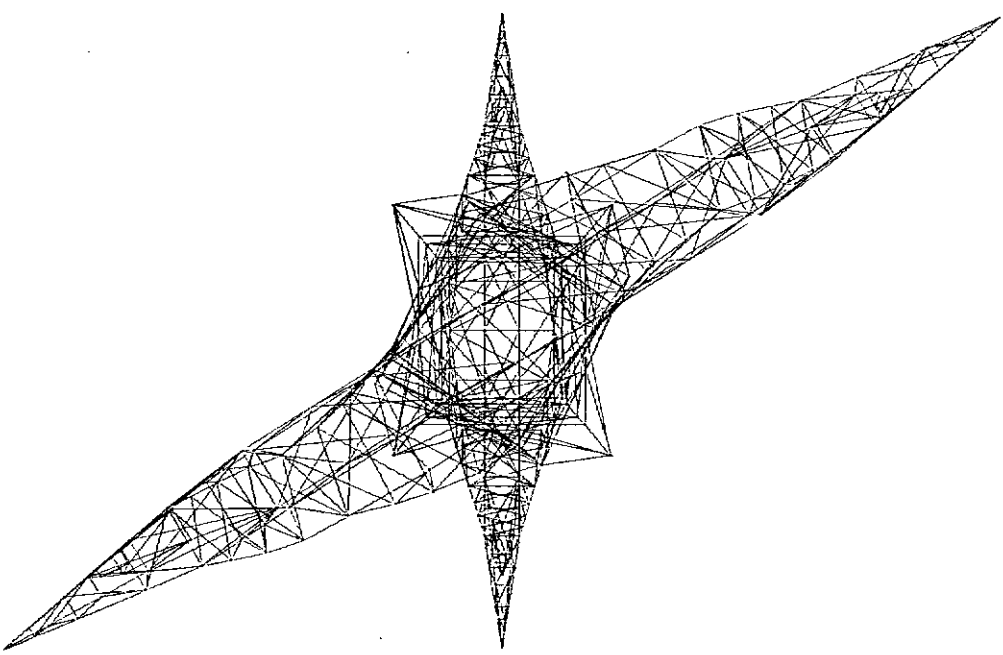


FIG VII.67.

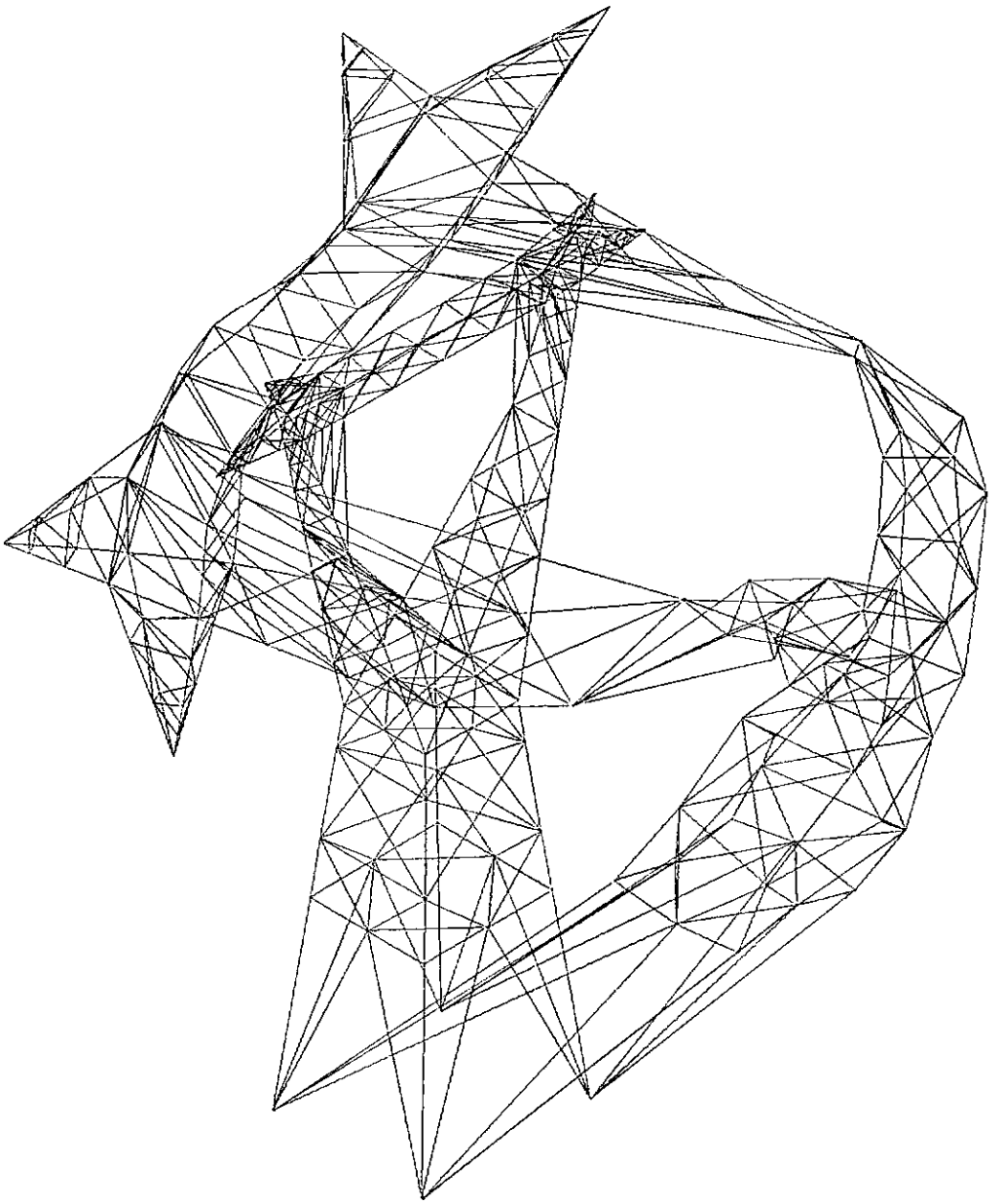


FIG VII.68.

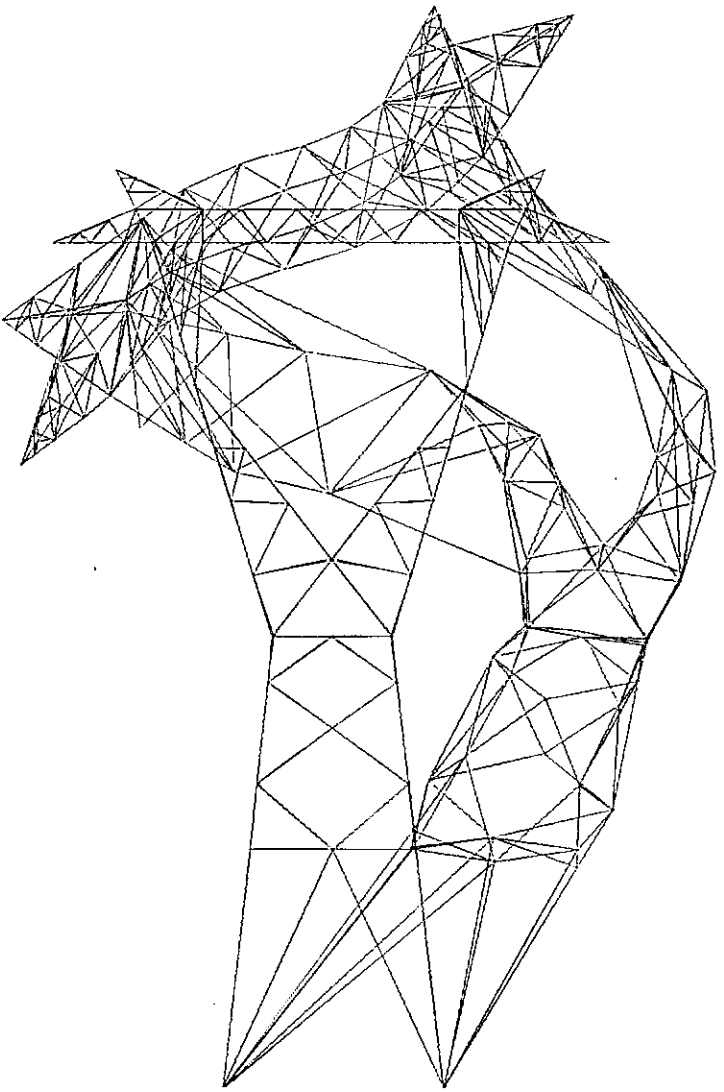


FIG VII.69.

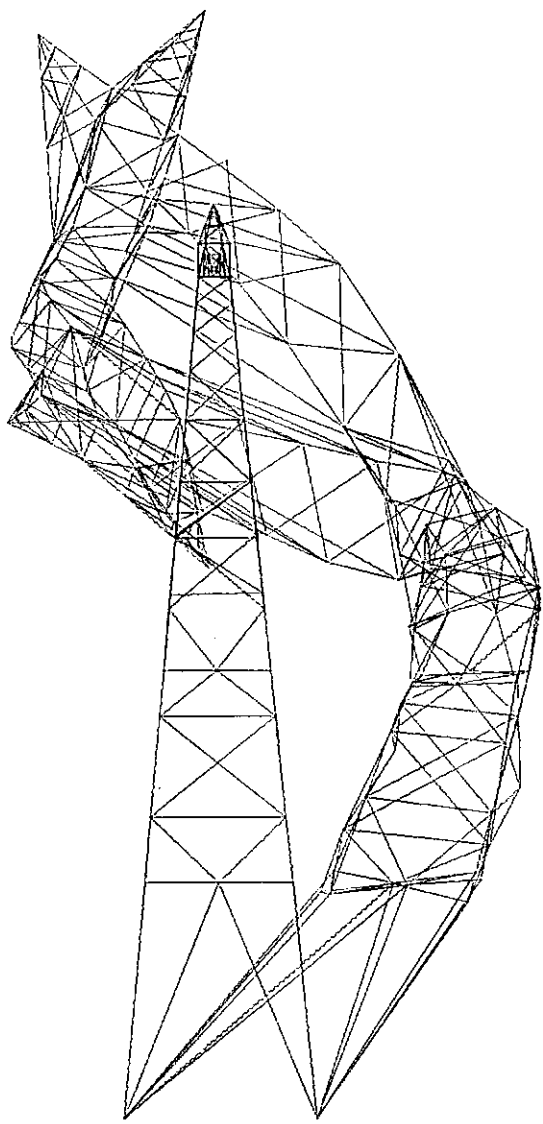


FIG VII.70.

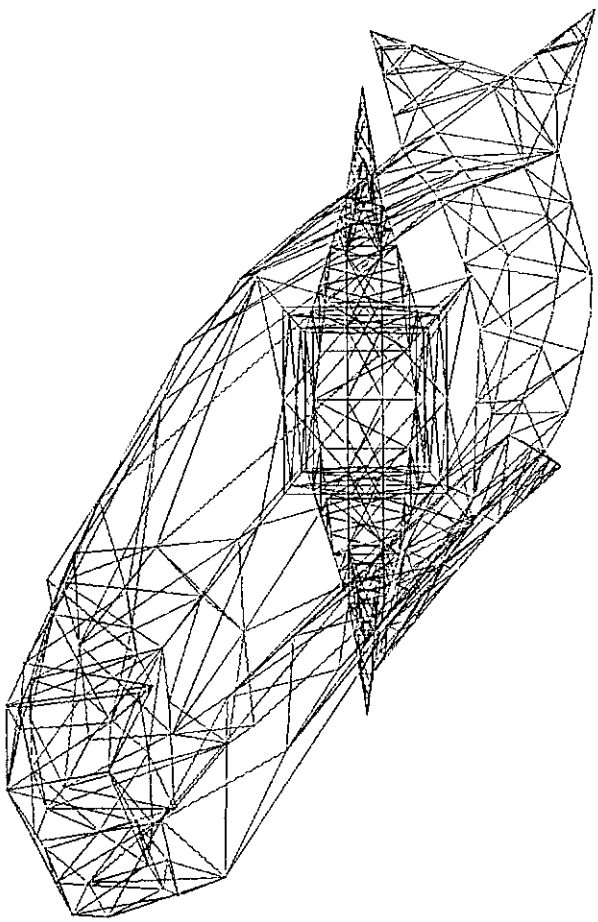


FIG VII.71.

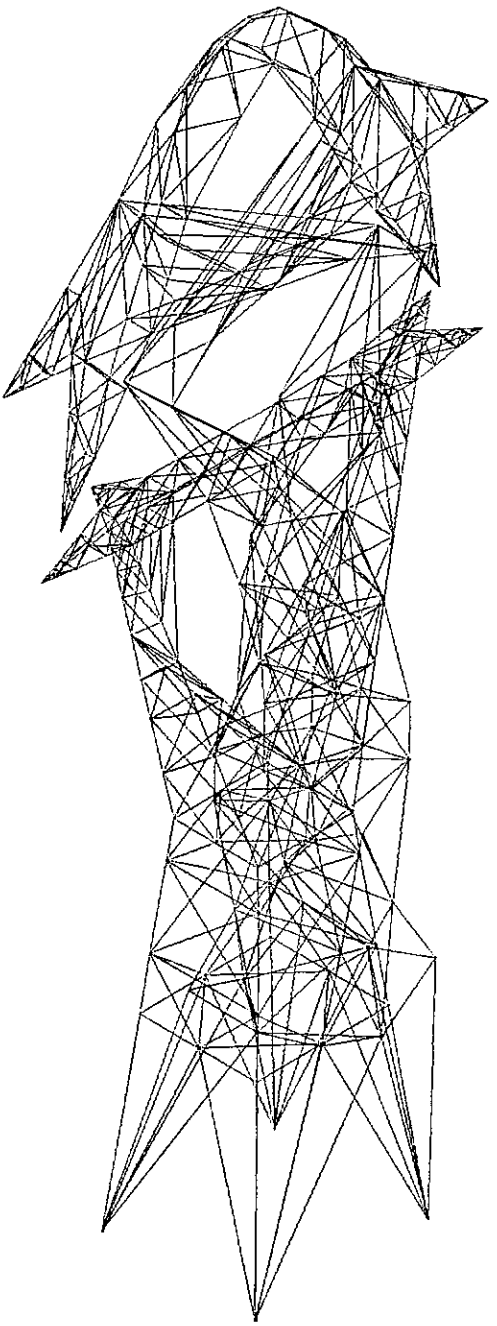


FIG VII.72.

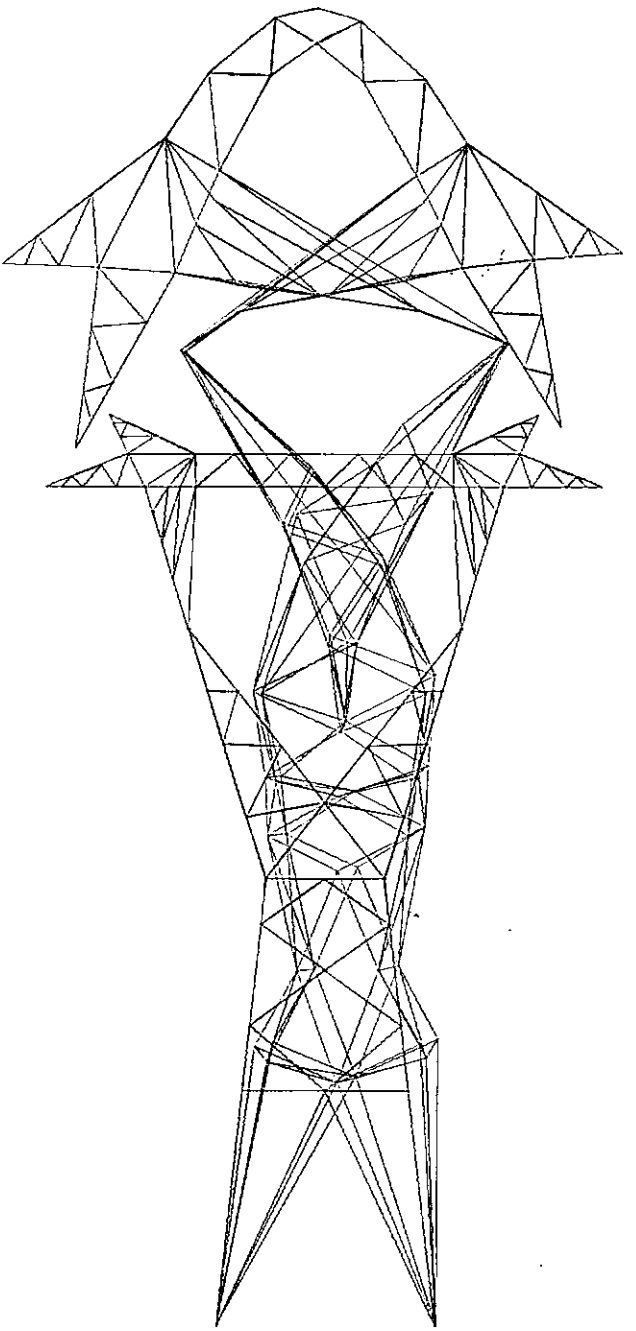


FIG VII.73.

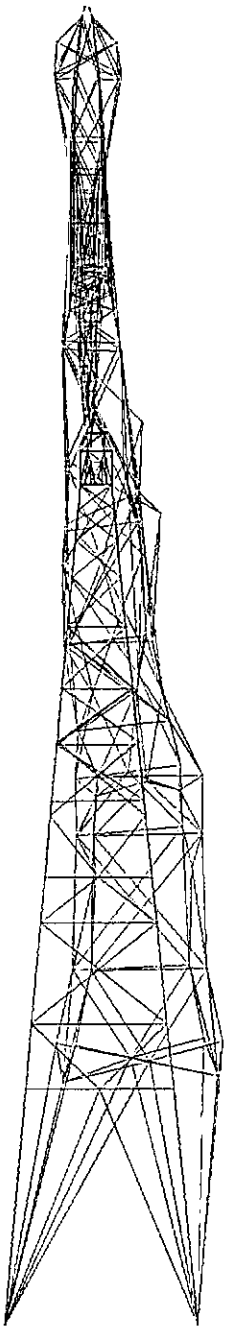


FIG VII.74.

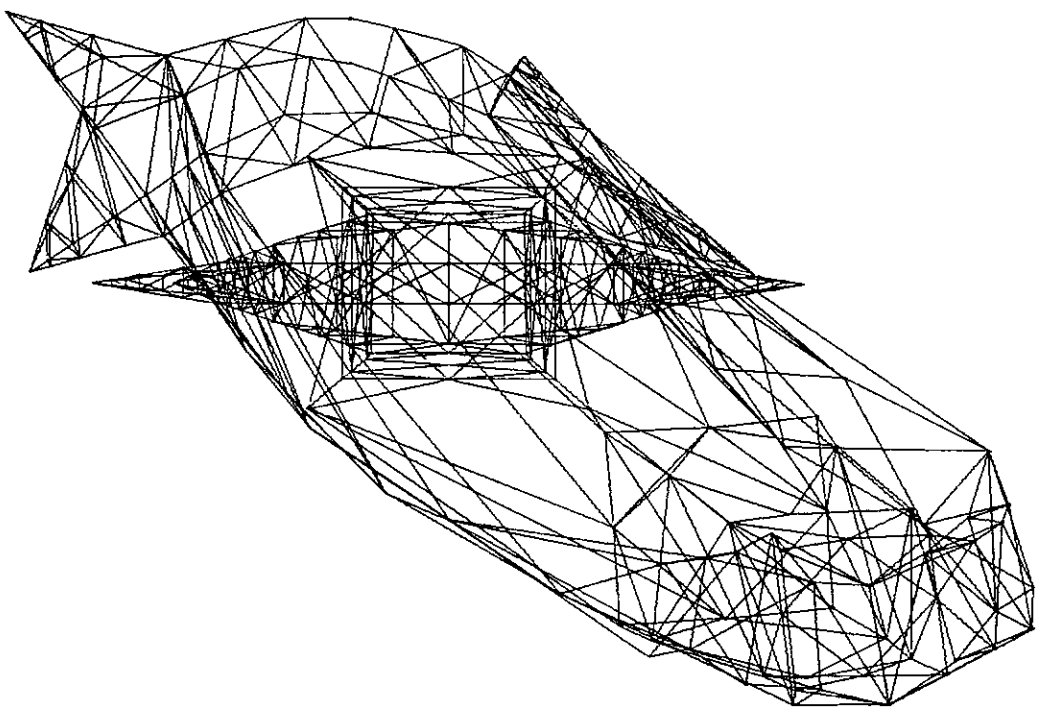


FIG. VII.75.

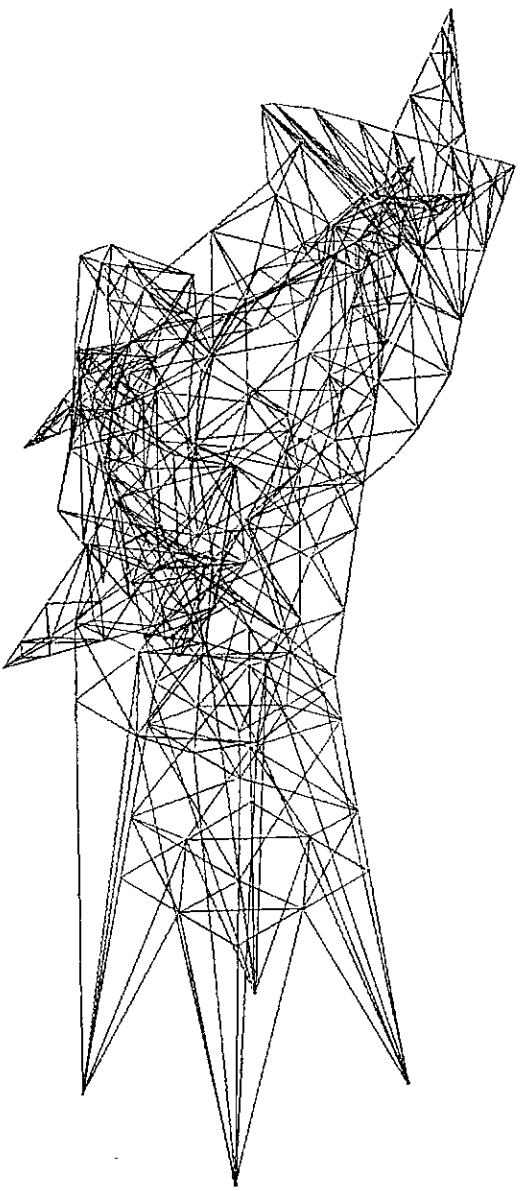


FIG VII.76.

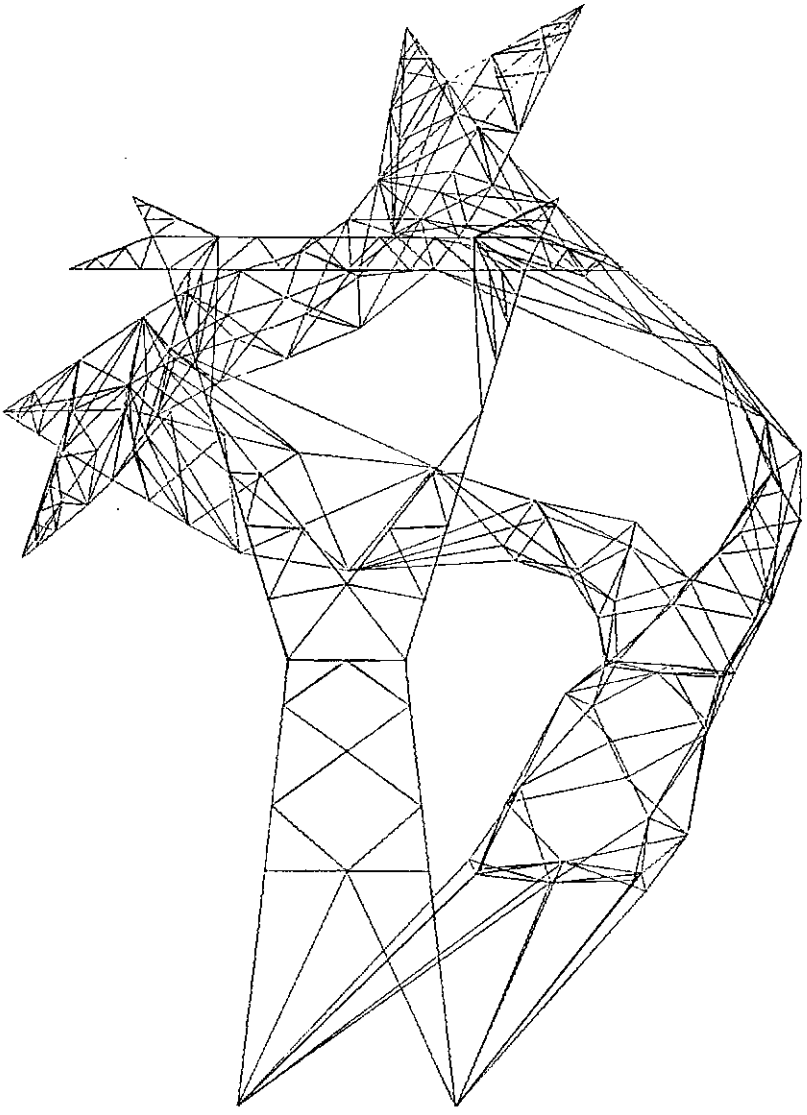


FIG VII.77.

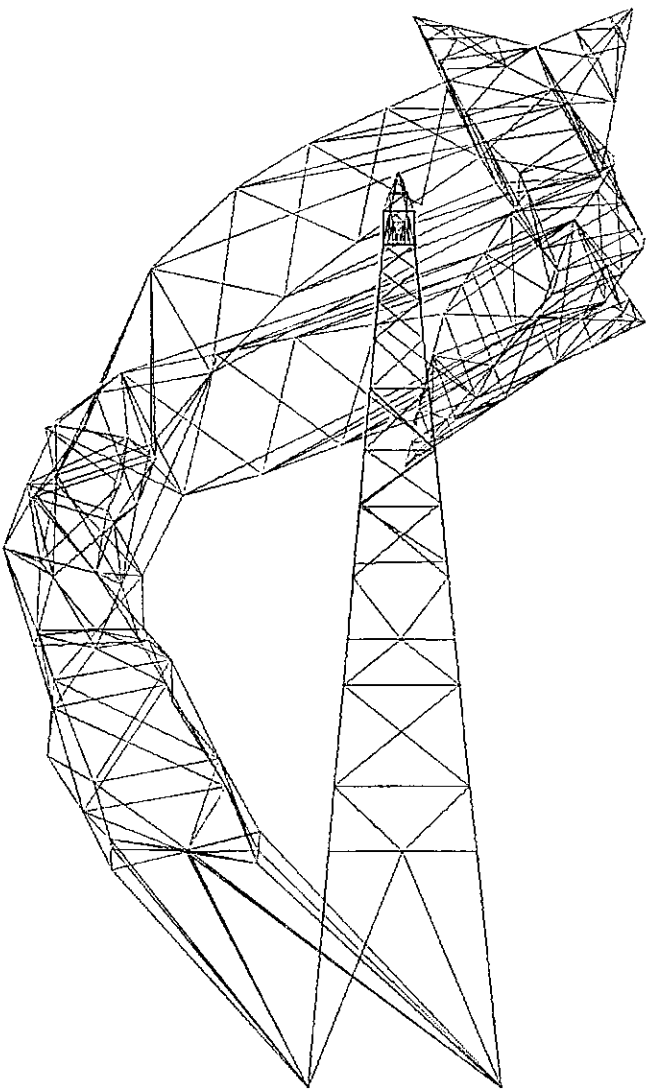


FIG VII.78.

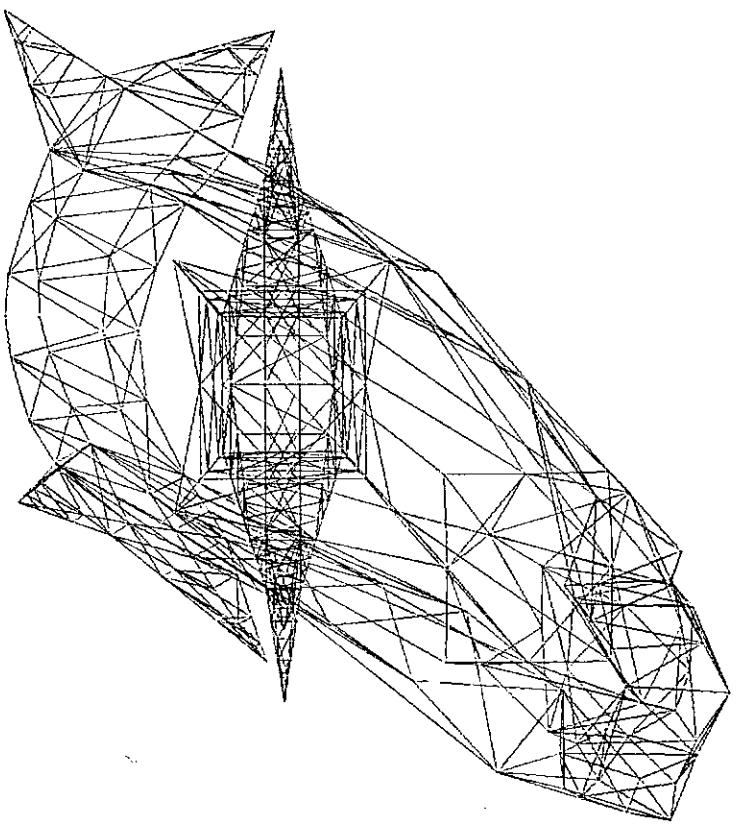


FIG VII.79.

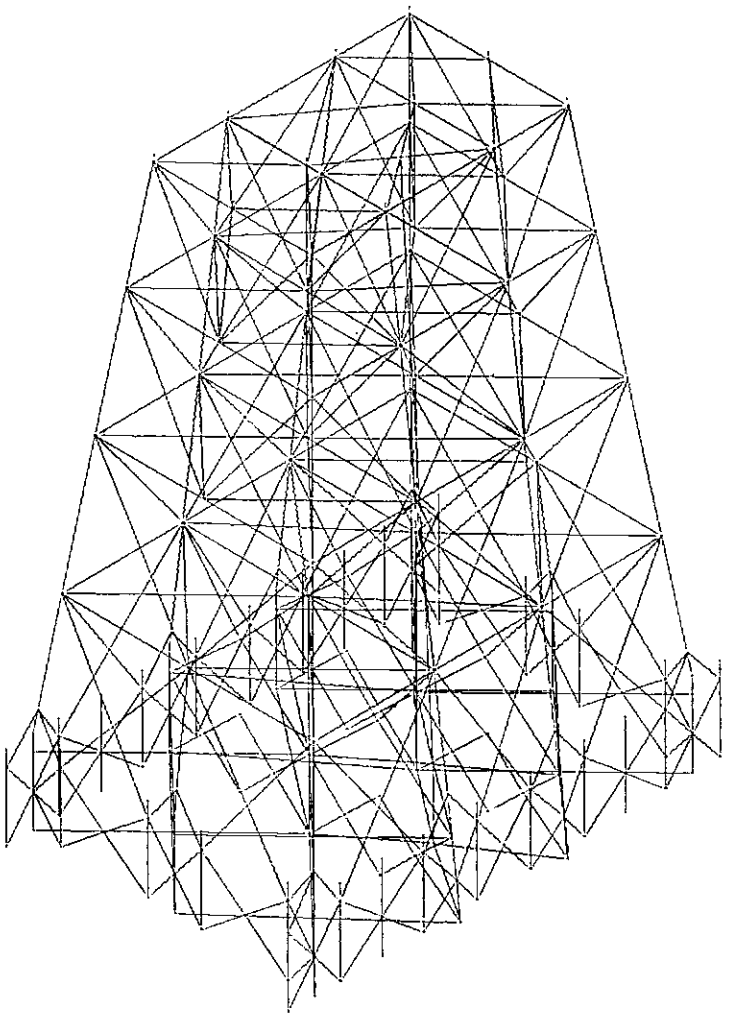


FIG VII.80.

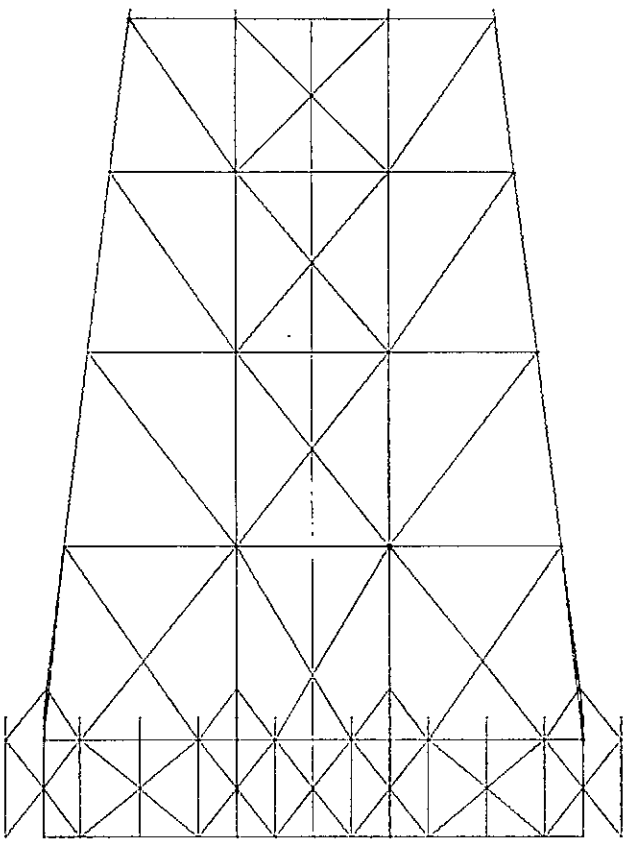


FIG VII.81.

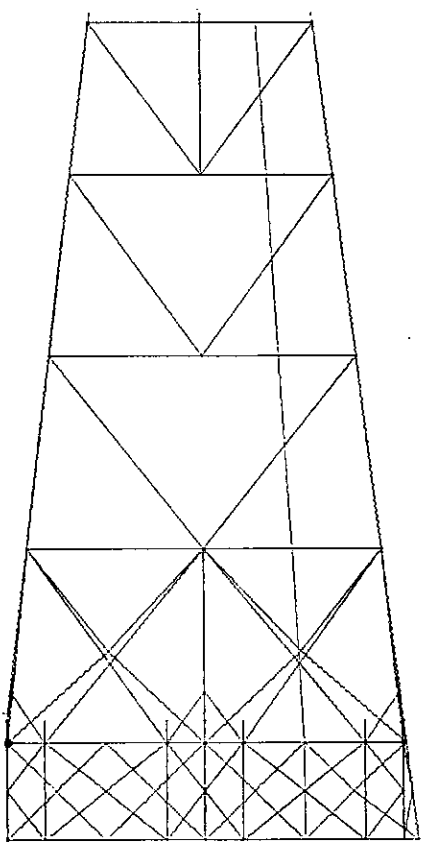


FIG VII.82.

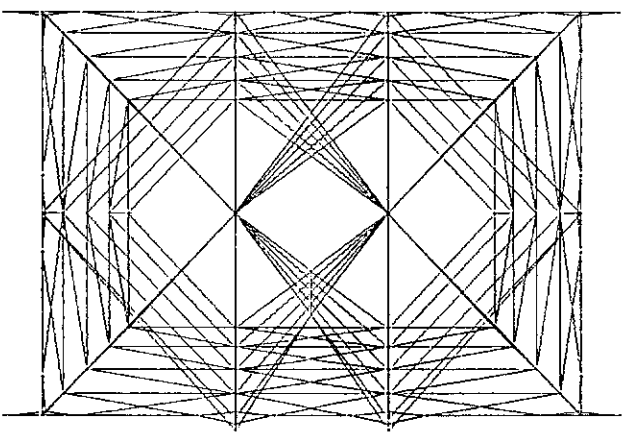


FIG VII.83.

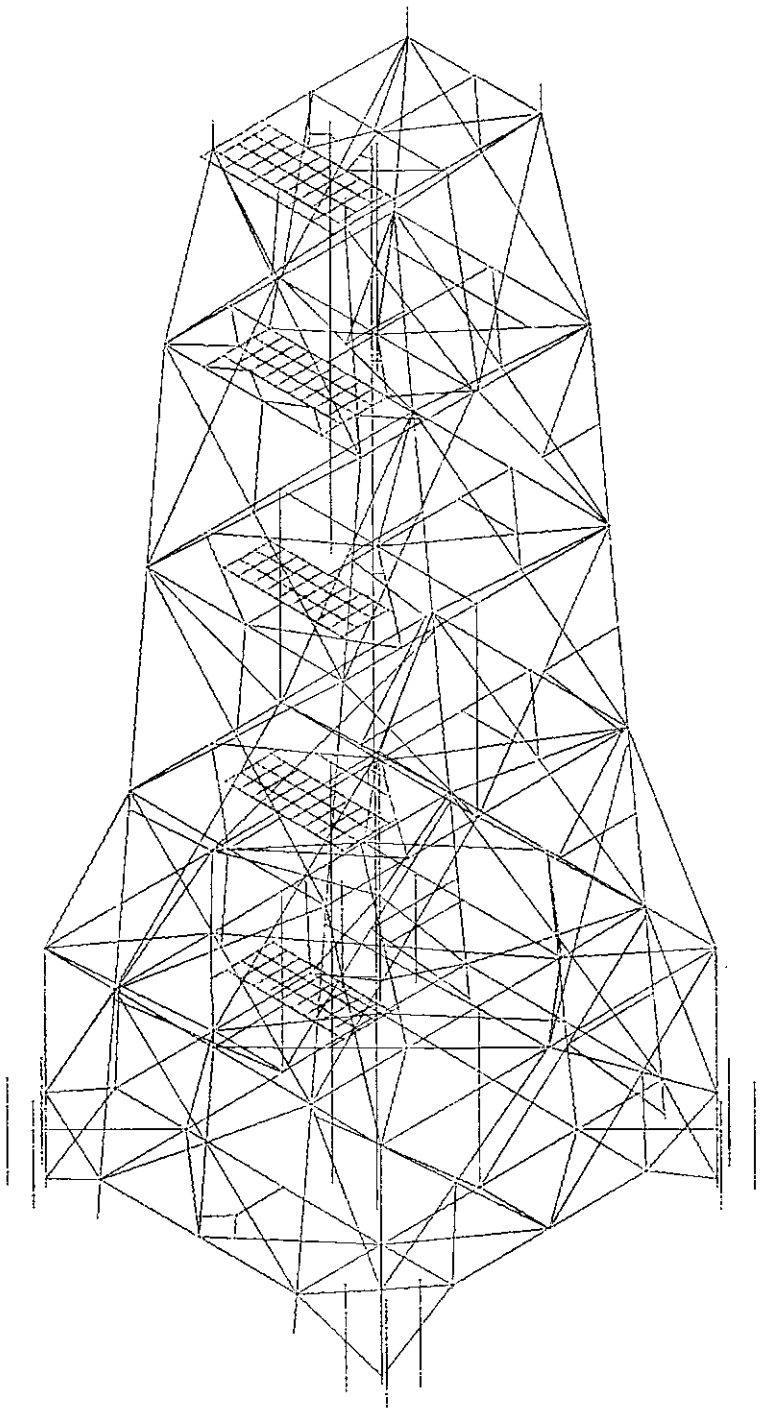


FIG VII.84.

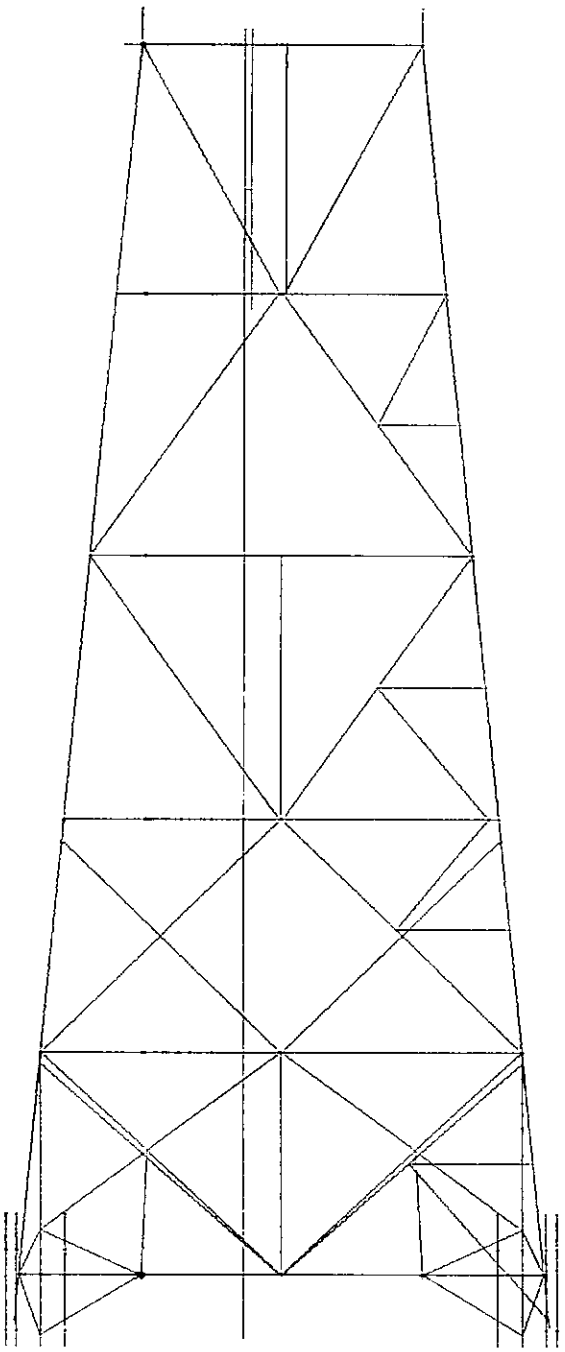


FIG VII.85.

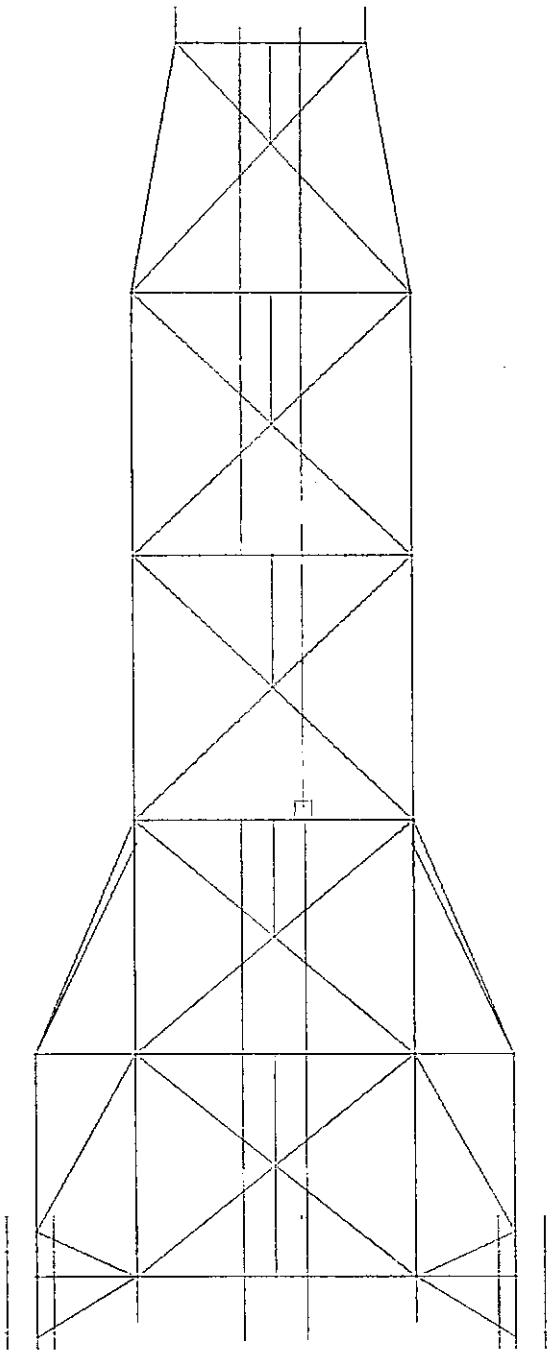


FIG VII.86.

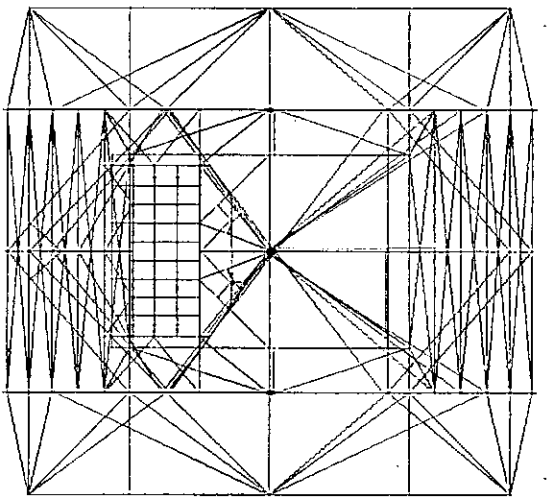


FIG VII.87.

VII.4. ELEMENTOS DE CONTORNO

Este grupo de elememtos é dedicado ao método dos elementos de contorno. São apresentados exemplos práticos onde o sistema **AutoMESH** foi utilizado para a visualização dos dados ou pós–processamento dos resultados.

Os exemplos apresentados aqui visam demonstrar a aplicabilidade do sistema **AutoMESH** à geração de malhas de elementos de contorno.

VII.4.1. EXEMPLO XXI

A figura VII.88 apresenta a malha de dois tanques, um cilíndrico e um retangular, utilizados no estudo da influência hidrodinâmica.

A figura VII.89 apresenta o mesmo modelo com remoção de linhas ocultas.

Deve–se observar o detalhe do fundo do tanque cilíndrico, onde nem todos os elementos têm seus vértices coincidindo. Esta condição deverá ser evitada para elementos finitos.

VII.4.2. EXEMPLO XXII

O modelo apresentado na figura VII.90 foi utilizado para estudo de influência hidrodinâmica entre uma plataforma semi–submersível e um rebocador.

A figura VII.91 apresenta a malha referente ao modelo e as figuras VII.92, VII.93 e VII.94 apresentam diversas vistas do modelo.

A malha como um todo foi obtida através da aplicação do comando MIRROR, levando–se em consideração a dupla simetria para a plataforma e simetria simples para o rebocador.

A figura VII.95 apresenta o detalhe de um dos flutuadores da plataforma e a figura VII.96, a malha correspondente.

VII.4.3. EXEMPLO XXIII

O modelo apresentado na figura VII.97 foi utilizado para o estudo de estabilidade e representa uma plataforma do tipo TLP com pernas e ligações retangulares.

A figura VII.98 apresenta diversas vistas do modelo.

A figura VII.99 apresenta o detalhe de um quarto do modelo, o qual foi utilizado para gerar o restante da malha, levando–se em conta dupla simetria.

A figura VII.100 apresenta o detalhe de uma das pernas onde pode ser observado o interior e o fundo da malha.

VII.4.4. EXEMPLO XXIV

O modelo apresentado na figura VII.101 foi utilizado para o estudo da estabilidade e representa uma plataforma do tipo TLP com pernas e ligações cilíndricas. Os resultados obtidos com a análise foram comparados aos obtidos com o modelo do exemplo anterior.

A figura VII.102 apresenta a malha correspondente ao modelo.

A figura VII.103 apresenta a malha correspondente a um quarto do modelo usada para gerar a malha como um todo, considerando a dupla simetria.

Na figura VII.104 pode–se observar o mesmo detalhe com as linhas ocultas removidas.

A figura VII.105 apresenta o detalhe da perna da plataforma onde pode–se observar o interior e a conexão vazada com os cilindros de ligação das pernas.

A figura VII.106 apresenta uma outra vista para o detalhe, onde pode–se observar o fundo da perna.

VII.4.5. EXEMPLO XXV

Neste exemplo é apresentada a malha de elementos de contorno para a parte molhada de um rebocador.

Afigura VII.107 apresenta uma vista isométrica do modelo com remoção das linhas ocultas.

A figura VII.108 apresenta diversas vistas da malha.

Este modelo foi utilizado entre outros para estudos de influência hidrodinâmica, estabilidade e ancoragem.

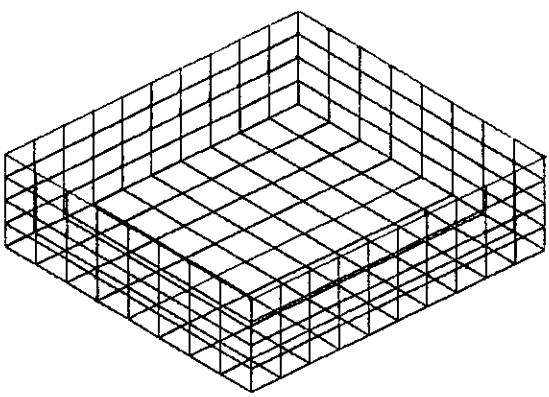
VII.4.6. EXEMPLO XXVI

Neste exemplo é apresentada a malha de elementos de contorno para a parte molhada de um navio.

A figura VII.109 apresenta uma vista isométrica do modelo com remoção das linhas ocultas.

A figura VII.110 apresenta diversas vistas da malha.

A malha como um todo foi obtida pela geração da malha pela metade do modelo como mostrado na figura VII.111. A utilização do comando MIRROR em relação ao eixo de simetria, permite a obtenção da malha total (figura VII.112).



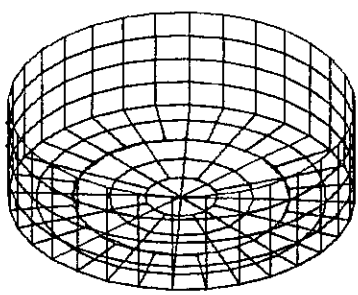
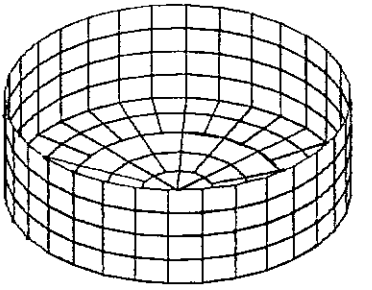


FIG VII.88.



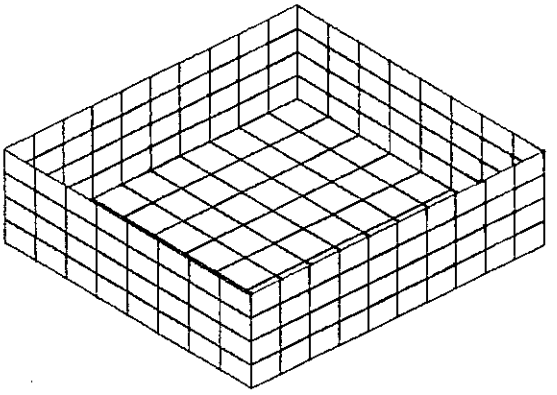


FIG VII.89.

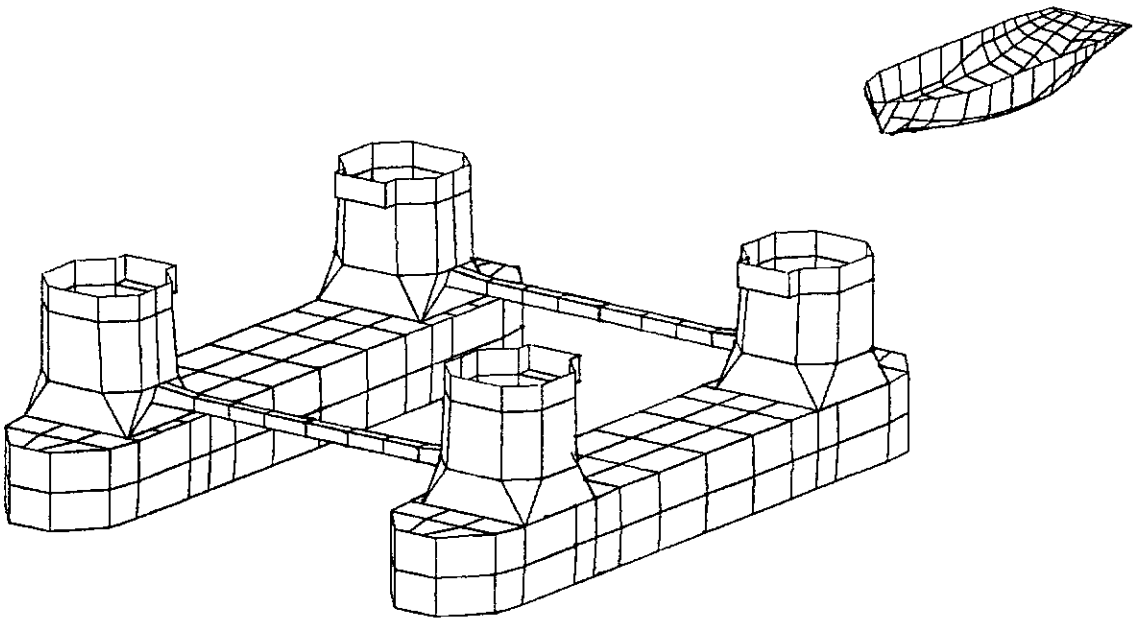


FIG VII.90.

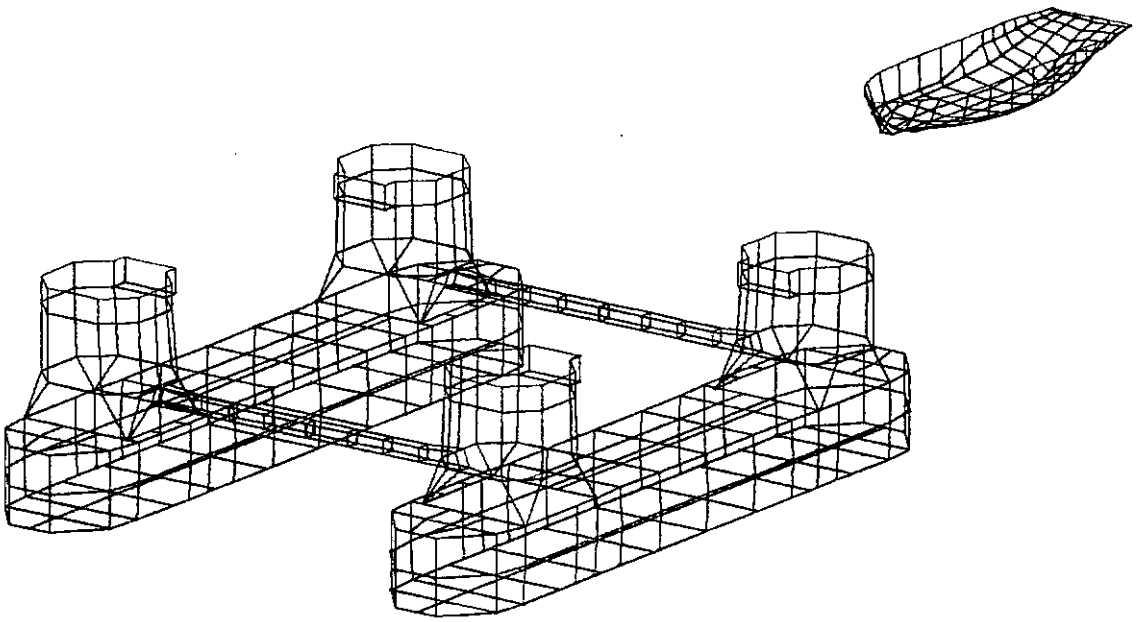


FIG VII.91.

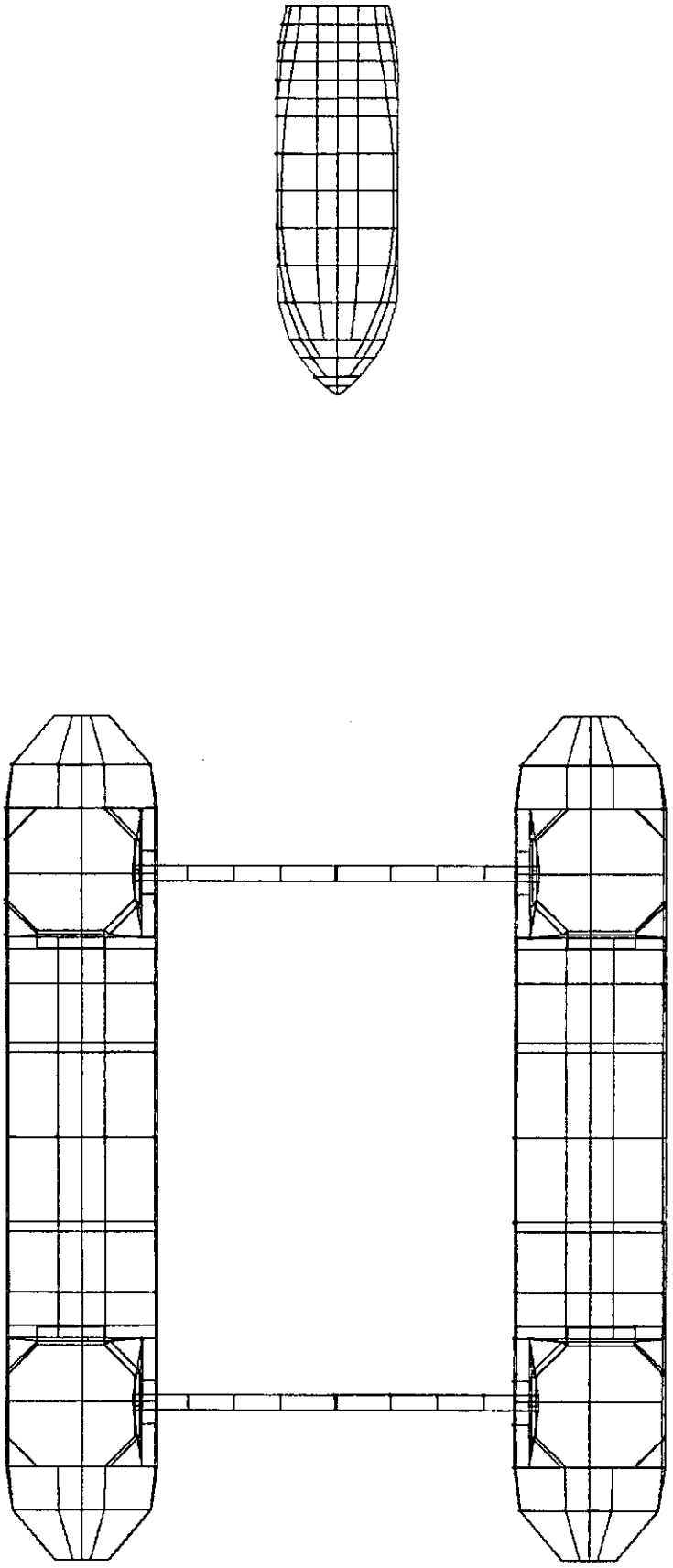


FIG VII.92.

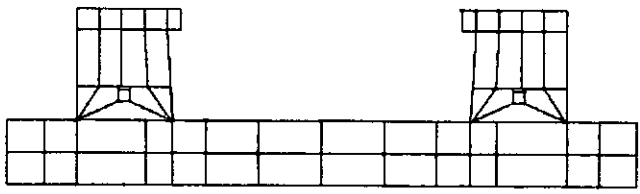


FIG VII.93.

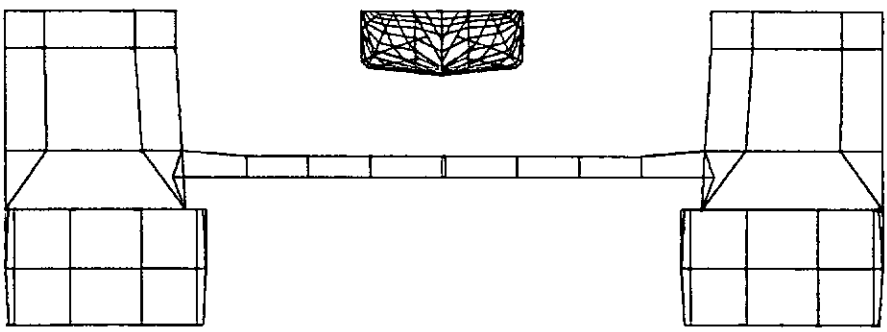


FIG VII.94.

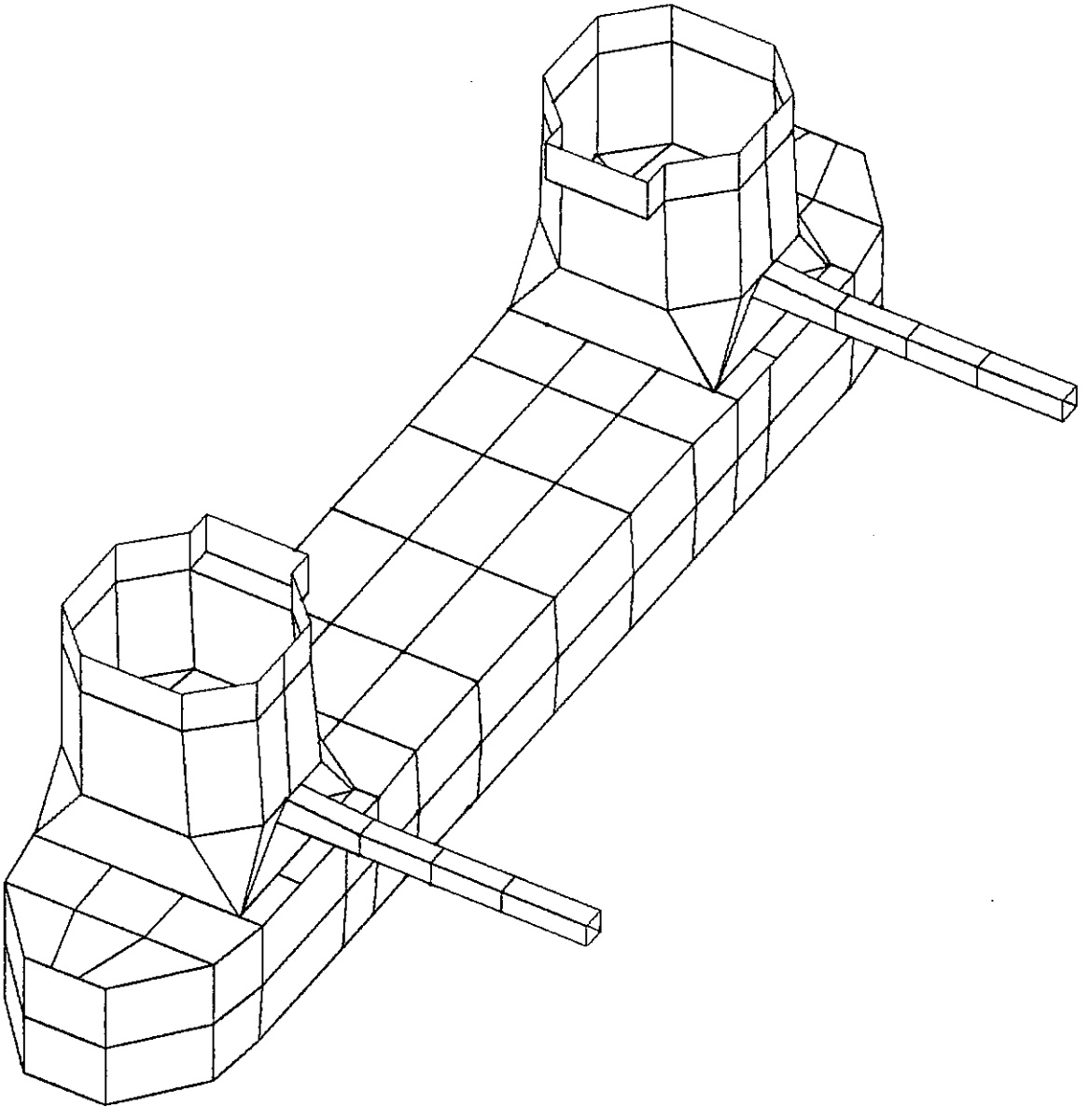


FIG VII.95.

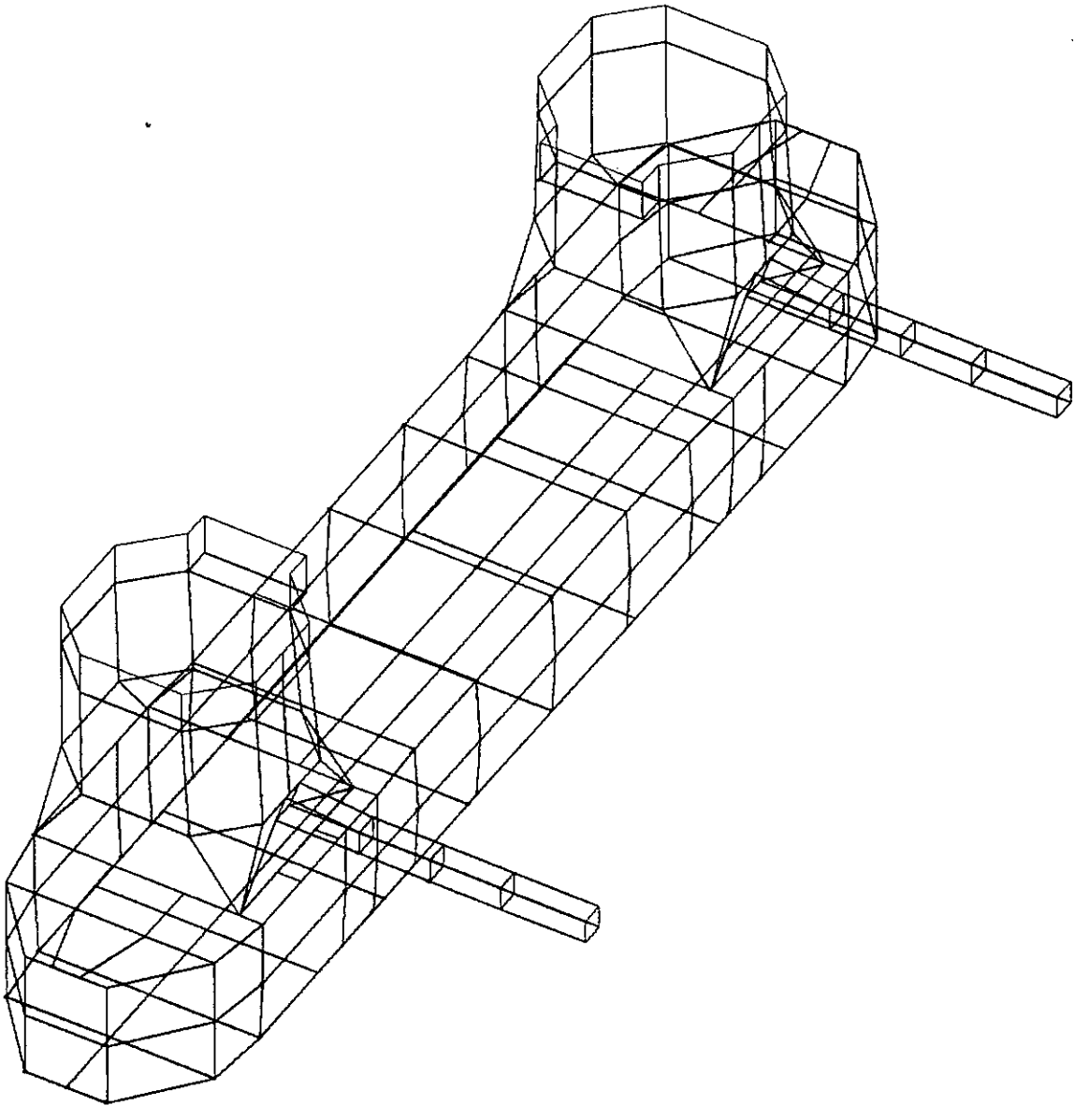


FIG VII.96.

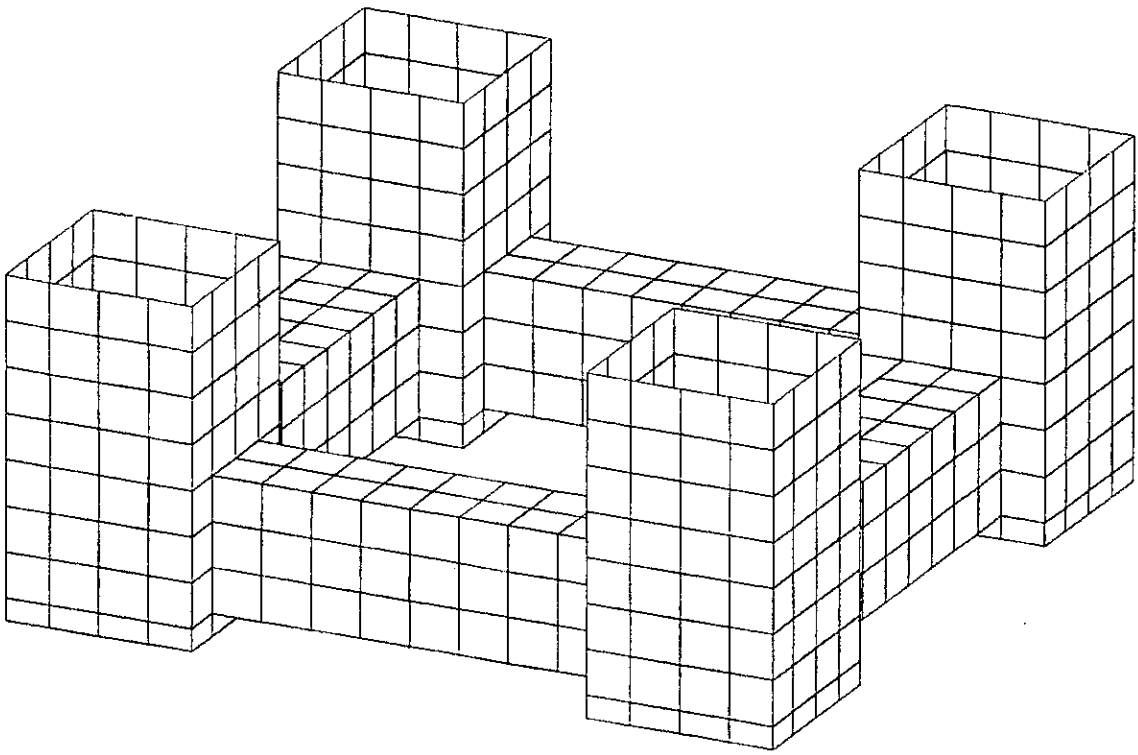
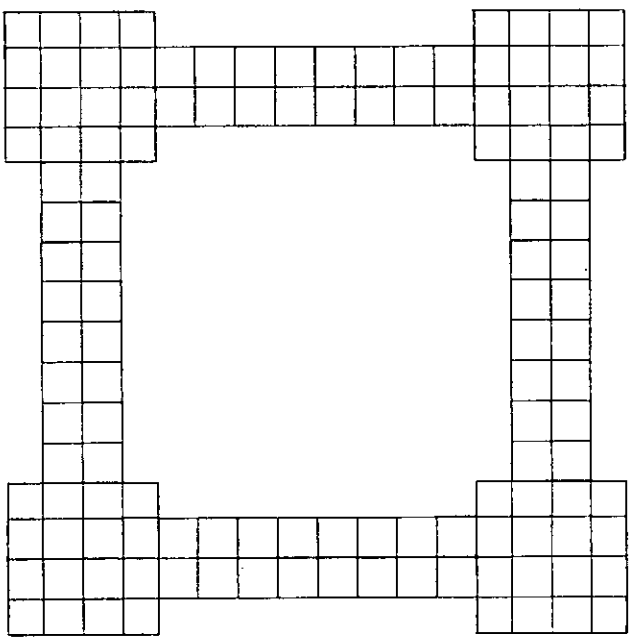
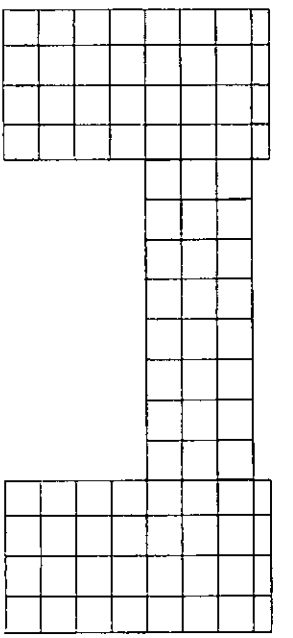


FIG VII.97.





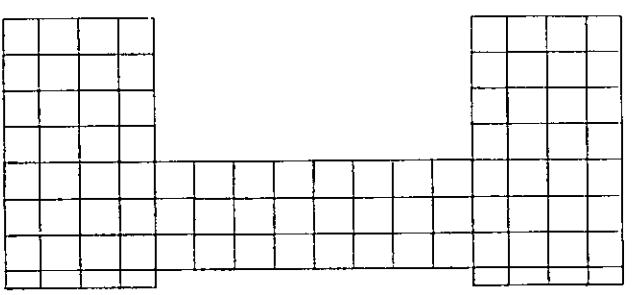


FIG VII.98.

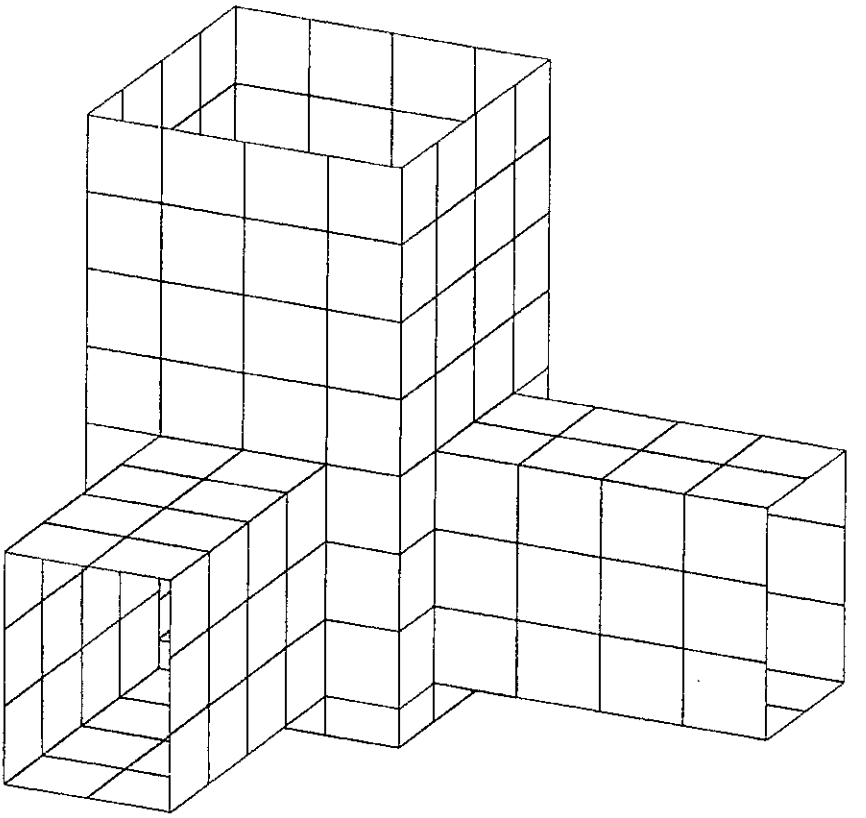


FIG VII.99.

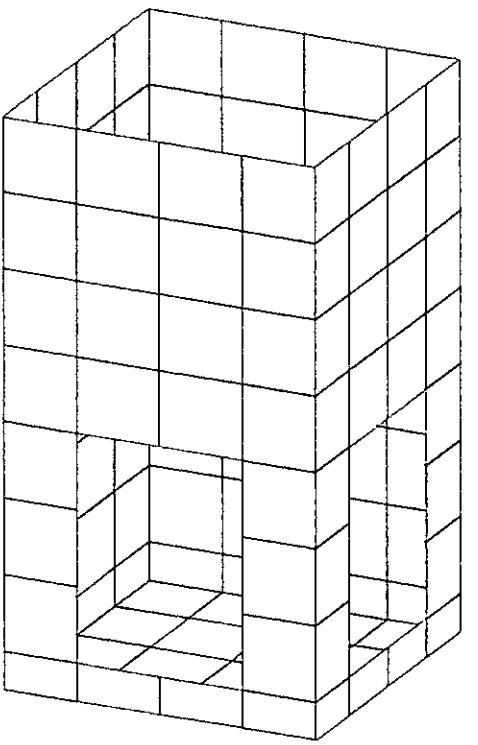


FIG VII.100.

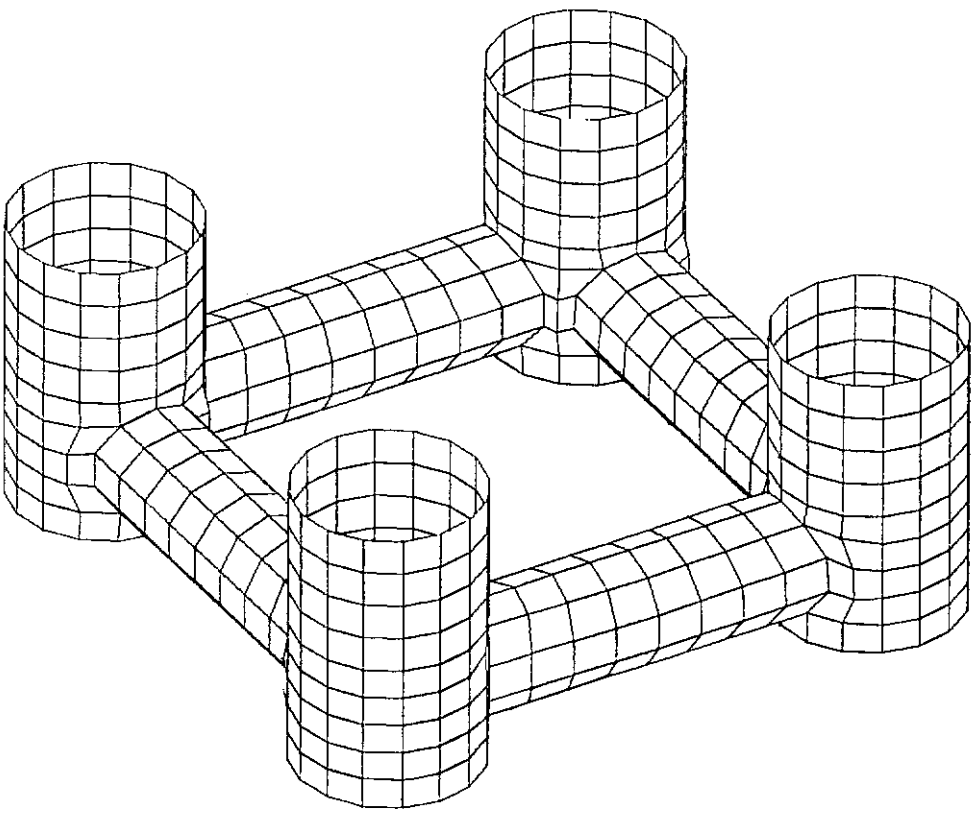


FIG VII.101.

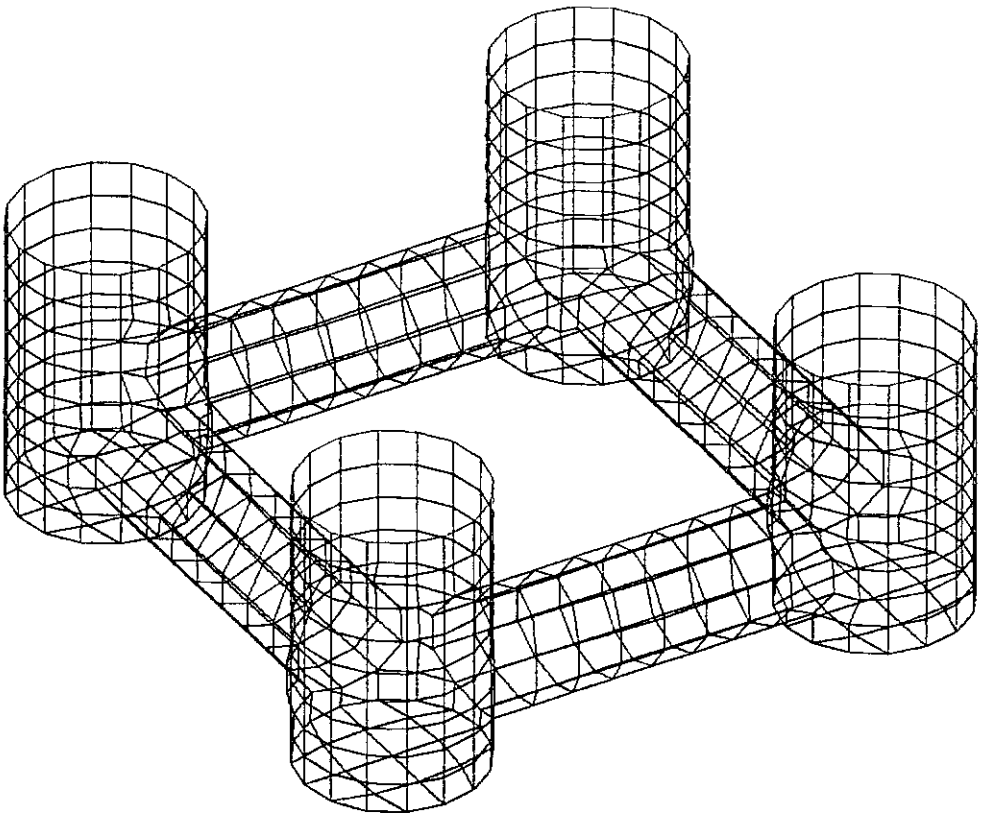


FIG VII.102.

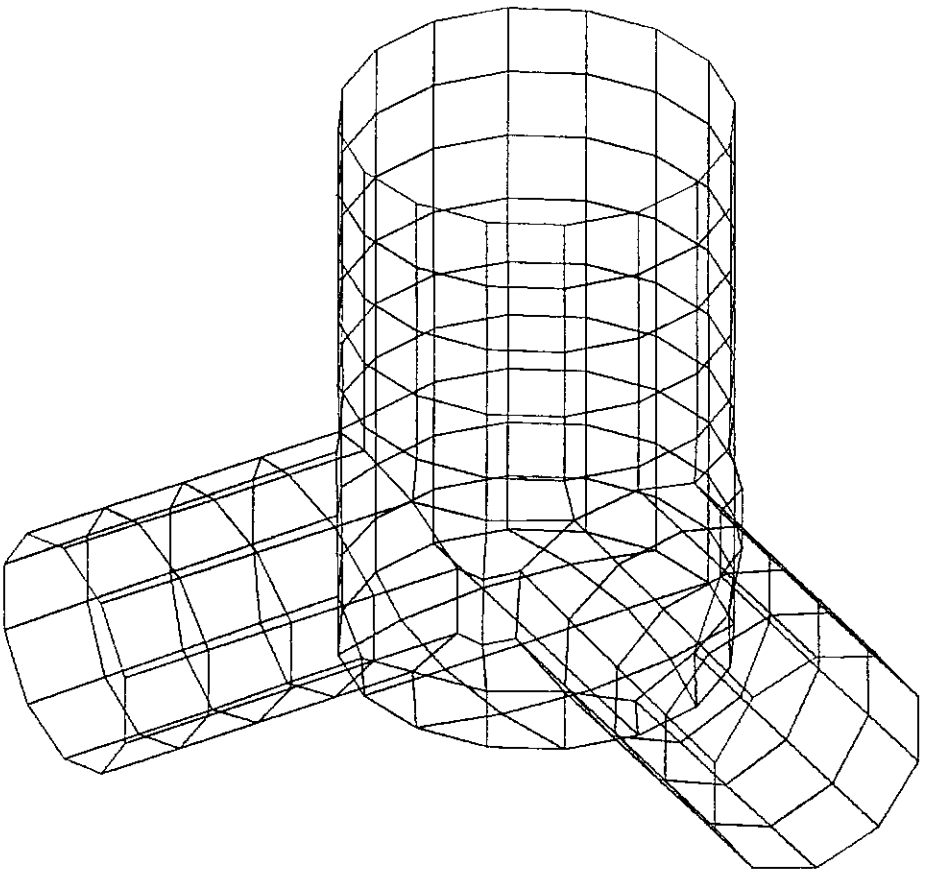


FIG VII.103.

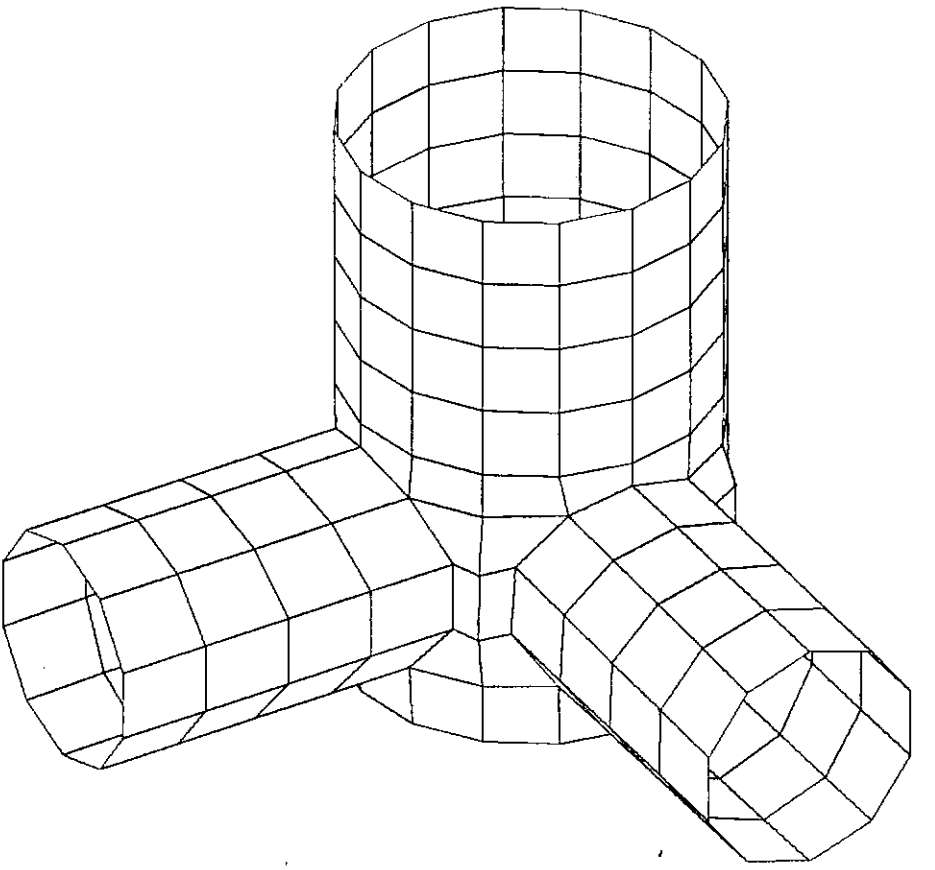


FIG VII.104.

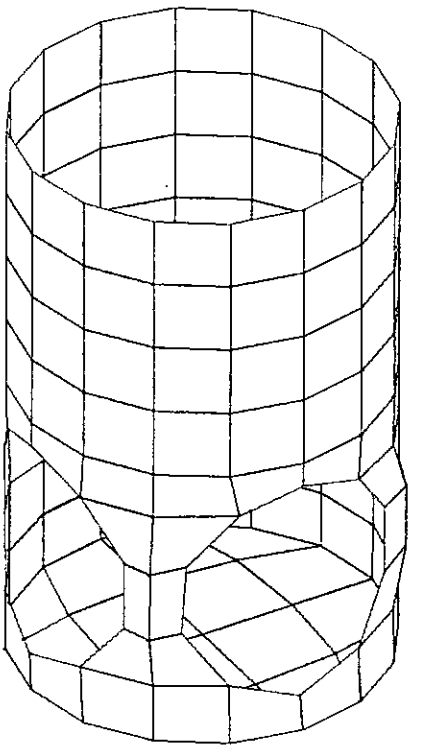


FIG VII.105.

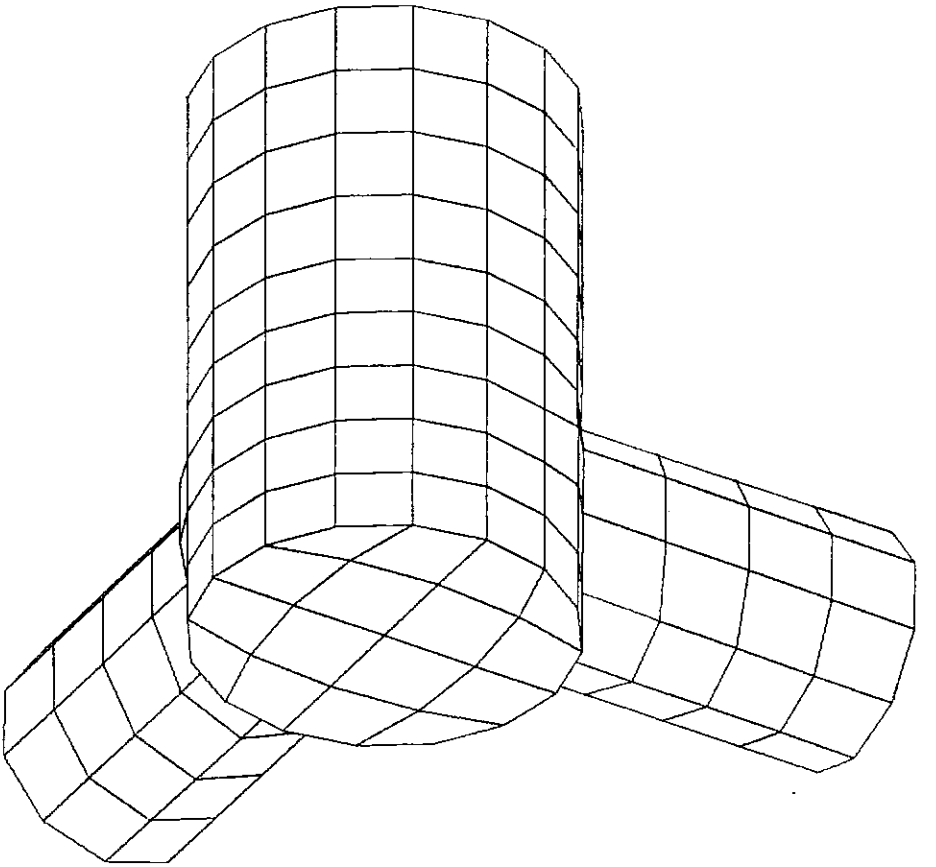


FIG VII.106.

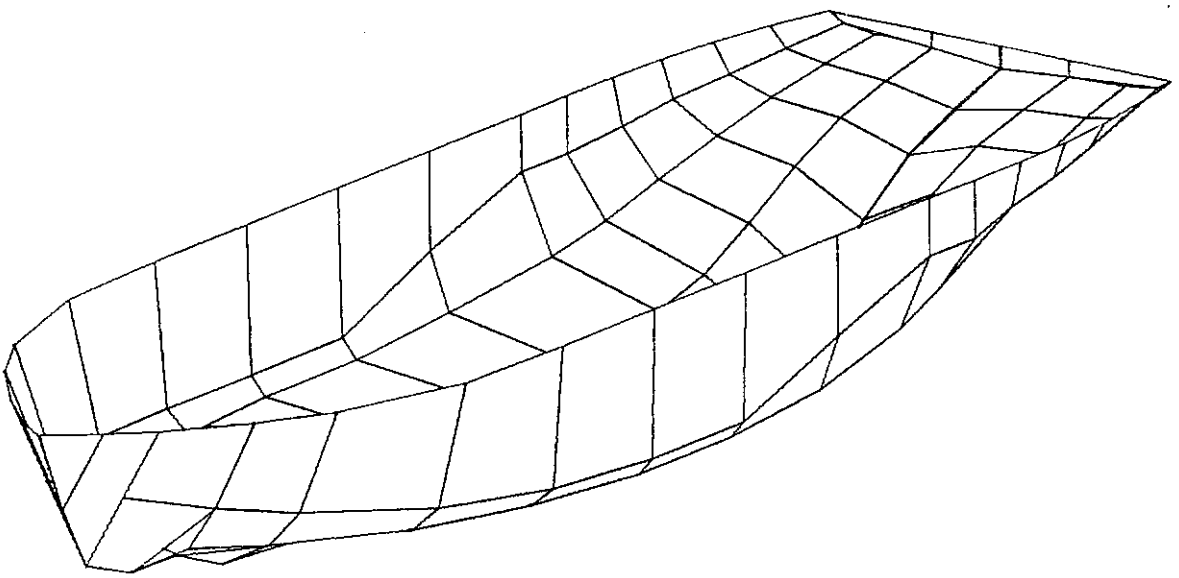


FIG VII.107.

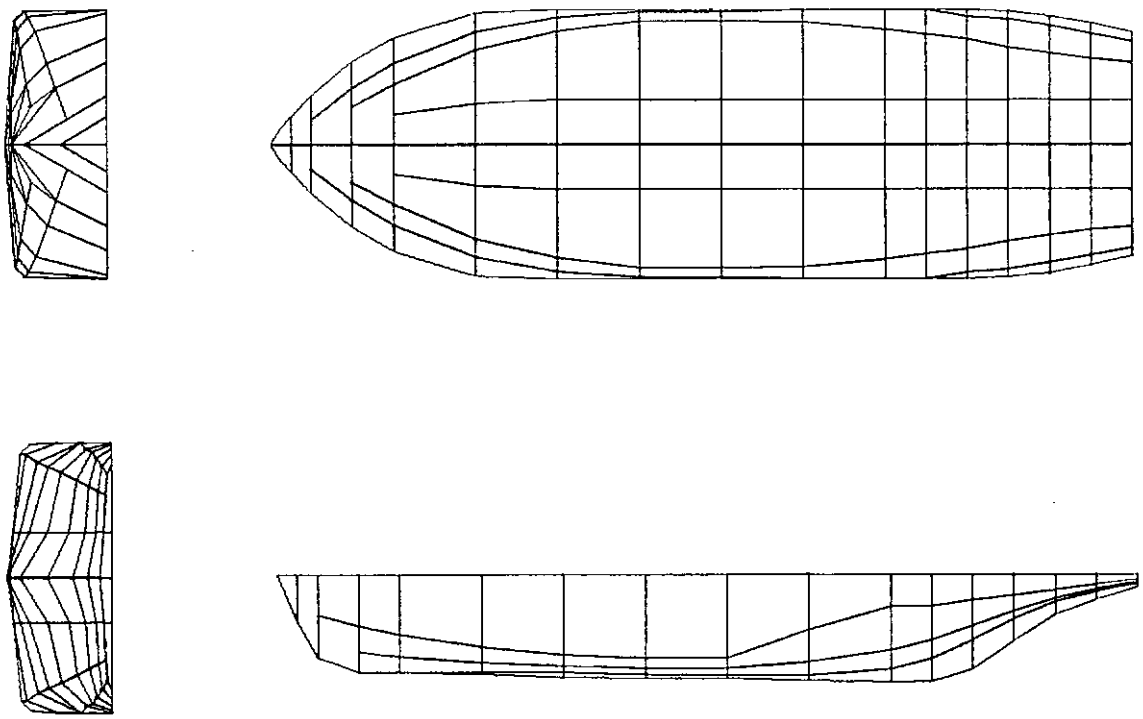


FIG VII.108.

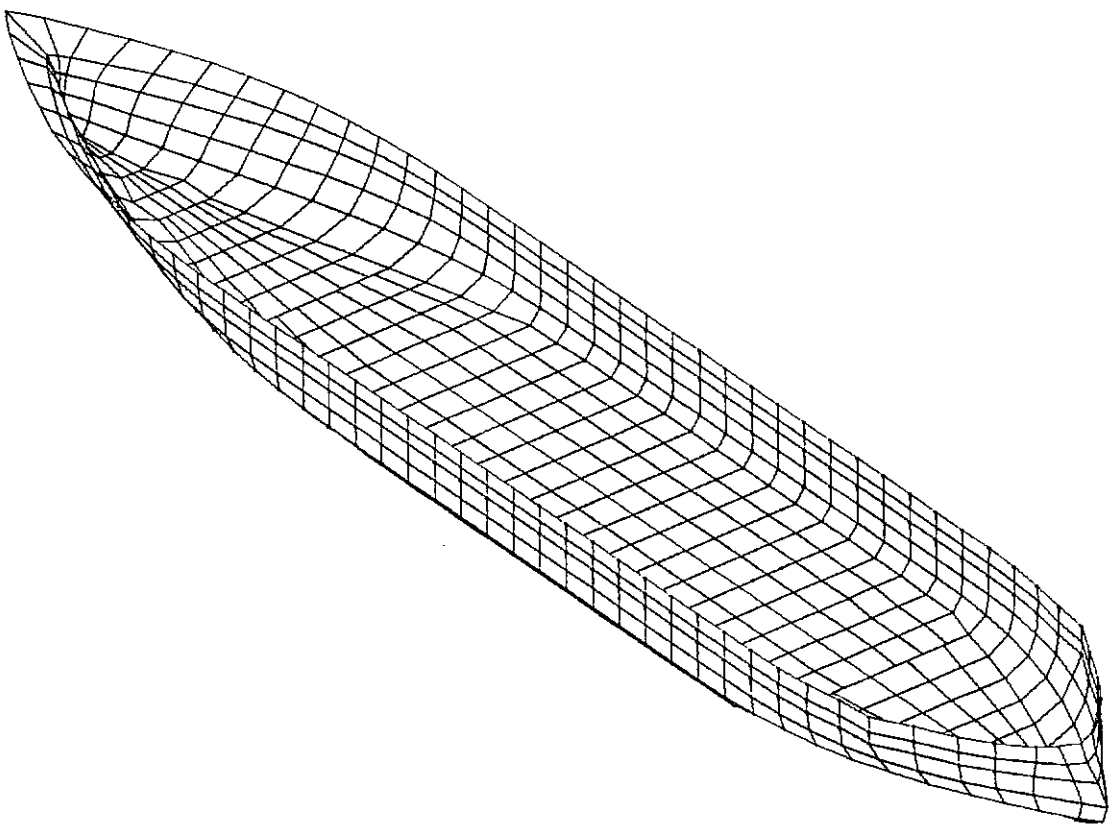


FIG VII.109.

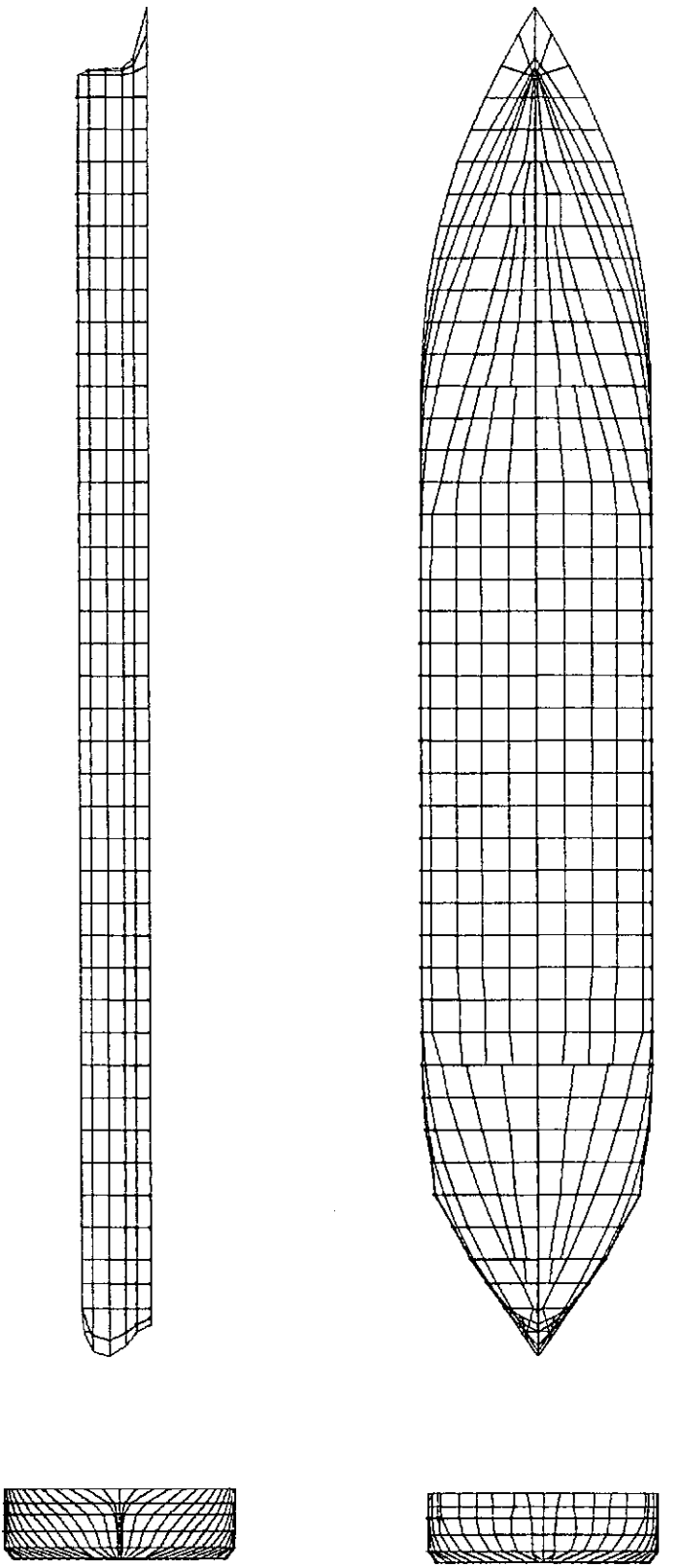


FIG VII.110.

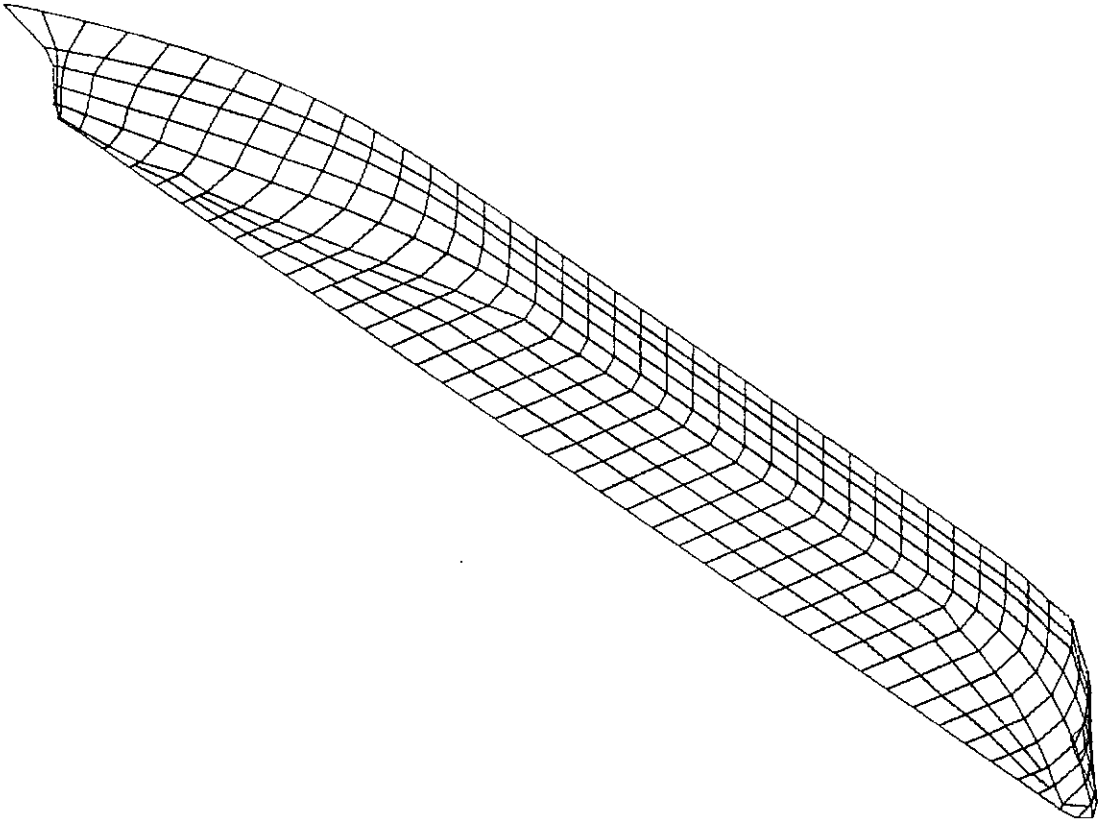


FIG VII.111.

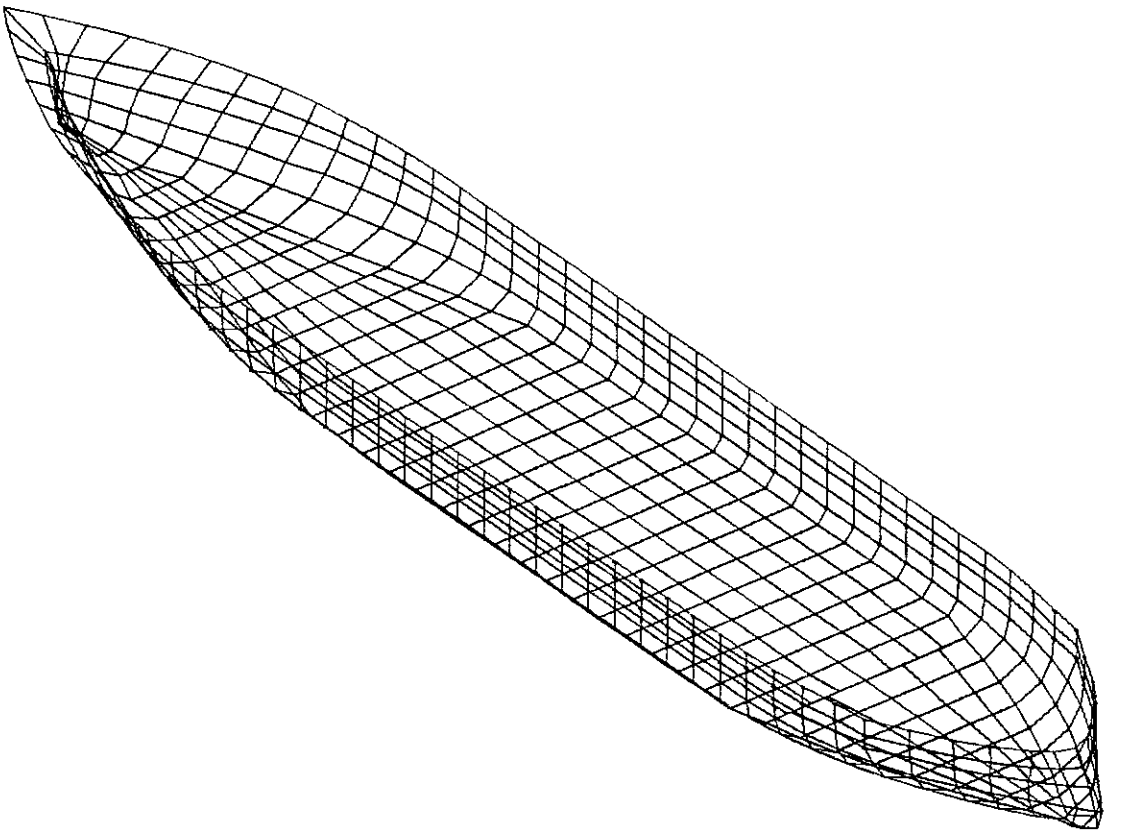


FIG VII.112.

VII.5. CASOS ESPECIAIS

Neste grupo são apresentados os exemplos onde técnicas "avançadas" foram utilizadas na geração da malha.

Os exemplos apresentados aqui não têm cunho prático, visando apenas demonstrar a potencialidade do sistema **AutoMESH**.

VII.5.1. EXEMPLO XXVII

A figura VII.113 apresenta a malha de elementos finitos para uma roda raiada de automóvel.

A malha como se pode observar é constituída de elementos quadriláteros e reticulados.

A figura VII.114 apresenta uma vista frontal da malha.

Na figura VII.115, pode–se observar a ligação dos raios com a coroa e com o aro.

No processo de geração da malha, definiu–se primeiramente os perfis para o aro e coroa e o eixo de giração da peça como mostrado na figura VII.116. A malha é, então, gerada com a utilização dos comandos REVSURF do **AutoCAD** e CONVERT do **AutoMESH**. A figura VII.117 apresenta uma vista isométrica da malha obtida.

Os elementos reticulados, referentes aos raios no modelo, são gerados com o auxílio do comando ARRAY opção POLAR, a partir de um grupo básico (quatro elementos) construído manualmente elemento a elemento.

VII.5.2. EXEMPLO XXVIII

Este exemplo corresponde à discretização de um suporte de eixo em elementos finitos.

A figura VII.118 apresenta uma vista isométrica do modelo com remoção das linhas ocultas.

A figura VII.119 apresenta a malha de elementos finitos para o modelo.

O modelo é formado por duas chapas paralelas em formato de "v", ligadas por uma chapa transversal.

A figura VII.120 apresenta a discretização de uma das chapas paralelas. Nesta figura podem ser observados diversos detalhes. Um dos detalhes é a discretização realizada na região do furo que resultou em um refinamento em torno desta singularidade. Outro detalhe é o grau de refinamento, diferente em cada extremo da peça, que exigiu uma gradação da malha em um dos braços. O último e mais crítico detalhe é a linha que corta a peça de ponta a ponta e que corresponde à ligação com a chapa transversal.

Em uma primeira etapa, foram geradas as malhas nas diversas sub–regiões desconsiderando a linha de ligação. No passo seguinte, alguns elementos quadriláteros foram divididos em dois triângulos e outros tiveram alguns vértices deslocados de modo a coincidirem com a linha de ligação.

A malha da chapa paralela foi obtida por cópia.

A figura VII.121 apresenta a malha para a chapa transversal.

VII.5.3. EXEMPLO XXIX

O modelo discretizado neste exemplo correponde a uma turbina de avião do tipo Boing. O modelo é constituído de duas partes, fuzelagem e turbina. A figura VII.122 apresenta uma vista geral do conjunto.

A malha foi gerada baseada na axissimetria do modelo. Definiu–se primeiramente o perfil transversal das peças, o eixo de giração e os círculos que serviram de guia das superfícies de rotação. A figura VII.123 apresenta uma vista global dos perfis e guias. Na figura VII.124 pode–se observar melhor os perfis transversais da fuzelagem e turbina. A figura VII.125 apresenta uma vista frontal.

As figuras VII.126, VII.127 e VII.128 apresentam vistas laterais para as malhas da turbina, fuzelagem e o conjunto respectiveamente, juntamente com os perfis.

A figura VII.129 apresenta uma vista frontal do conjunto completo. Pode–se observar o suporte da turbina que faz parte do modelo, mas não foi discretizado.

As figuras VII.130 e VII.131 apresentam vistas isométricas para a malha da turbina e fuzelagem respectivamente, em conjunto com as linhas dos perfis e guias.

As malhas da turbina e fuzelagem obtidas são apresentadas nas figuras VII.132 e VII.133 respectivamente. Para a malha da turbina além de elementos quadriláteros, também foram utilizados elementos triangulares como mostrado na figura VII.134.

As figuras VII.135, VII.136, VII.137 e VII.138 apresentam diversas vistas do conjunto com remoção das linhas ocultas.

VII.5.5. **EXEMPLO XXX**

A figura VII.139 apresenta a malha de elementos finitos para uma hélice de três pás. A malha é constituída basicamente de elementos quadriláteros. Alguns elementos triangulares foram usados na linha de ligação do eixo com as pás para compatibilização na ligação das malhas.

As figuras VII.140 e VII.141 apresentam duas vistas diferentes da malha.

Para a geração da malha, o modelo foi dividido em três partes. A discretização foi realizada em uma dessas partes. O restante da malha foi obtido por cópia e rotação do terço discretizado.

Para a geração da malha da pá da hélice, utilizou–se diversos artifícios:

Figura VII.142 – Primeiro foram marcados com círculos, os diâmetros internos e externos para eixo e pás nos planos superior e inferior do modelo. Foram, então, divididos os círculos nas partes correspondentes às pás e construídos os perfis inicial e final para uma das pás.

Figura VII.143 – Para a construção das linhas que definem a borda externa da pá e a ligação com o eixo, construiu–se duas malhas. Uma malha interna, coincidente com a face externa do eixo; e uma malha externa, coincidente com a borda externa da pá.

Figura VII.144 – Utilizando as malhas como guia, as linhas de definição da pá foram construídas ligando diagonalmente os nós da malha. Deve–se observar que para que a ligação seja correta, o número de linhas da malha

deverá ser igual ao número de colunas mais um. Estas malhas não são malhas de elementos finitos e sim entidades gráficas do tipo ***mesh*** sem nenhum significado estrutural.

Figura VII.145 – Retiradas as malhas auxiliares, obtém–se as linhas que definem a pá.

A figura VII.146 apresenta a malha de elementos finitos para um terço do modelo.

VII.5.5. **EXEMPLO XXXI**

A figura VII.148 apresenta a malha de elementos finitos para um parafuso sextavado. A malha é constituída de elementos quadriláteros e triangulares.

Os artifícios utilizados para a geração da malha na rosca do parafuso, assemelham–se aos utilizados no exemplo anterior. As figuras VII.148, VII.149, VII.150 e VII.151 apresentam as diversas etapas para a construção de uma das roscas.

As figuras VII.152 e VII.153 apresentam as duas vistas da malha.

As figuras VII.154 e VII.155 apresentam detalhes da cabeça do parafuso. Deve–se notar (figura VII.155) o detalhe da divisão do quadrilátero, demasiadamente distorcido, em dois elementos triangulares.

A figura VII.156 apresenta o detalhe da malha na base do parafuso.

VII.5.6. EXEMPLO XXXII

A figura VII.157 apresenta a amlha de elementos finitos de um componente articulado.

Este modelo agrega diversas características típicas de peças mecânicas no que diz respeito à geração de malha; tais como, geração de malhas em regiões com furos, refinamento variável, linhas de ligação entre chapas e discretização com vários tipos de elementos em diferentes regiões do modelo.

As figuras VII.158, VII.159, VII.160 e VII.161 apresentam diversas vistasda malha de elementos finitos para o modelo.

A figura VII.162 apresenta a definição das sub–regiões e linhas de ligação entre as chapas. As sub–regiões foram definidas de acordo com o grau de refinamento de modo a possibilitar a aplicação de comandos de geração automática. A malha foi gerada apenas ao redor de um dos furos, para oa outro utilizou–se o recurso de cópia com espelhamento em relação à linha de simetria entre os furos. A malha da outra chapa paralela foi obtida por cópia direta.

A figura VII.163 e VII.164 apresentam a definição das sub–regiões para a chapa transversal. Deve–se notar o detalhe dos furos.

A figura VII.165 apresenta a definição das sub–regiões para a peça toda.

As figuras VII.166, VII.167, VII.168 e VII.169 apresentam o resultado do pós–processamento de uma análise estática onde pode–se visualizar a peça deformada. Para a análise, a peça foi submetida a um carregamento vertical para cima aplicado aos nós dos furos superiores,

mantendo–se engastados os nós dos furos inferiores. A figura VII.170 apresenta uma vista isométrica da estrutura deformada, com remoção das linhas ocultas.

As figuras VII.171, VII.172, VII.173 e VII.174 apresentam as diversas vistas para a estrutura deformada em uma análise estática onde os pontos nodais dos furos supreriores foram submetidos a carregamentos concentrados verticais para cima e os pontos nodais dos furos inferiores a carregamentos concentrados horizontais para a direita. Além disso, foram engaastados os nós de uma das chapas transversais. A figura VII.175 apresenta uma vista isométrica da estrutura deformada, com remoção das linhas ocultas.

As figuras VII.176 e VII.177 apresentam a distribuição de tensões na estrutura através do traçado das linhas de isotensão para as faces esquerda e direita respectivamente e na figura VII.178 é apresentada uma vista de topo. A figura VII.179 apresenta uma vista isométrica da figura com remoção das linhas ocultas.

Infelizmente, os processos de reprodução disponíveis no momento impedem que possam ser visualizadas as diversas cores utilizadas no desenho referentes à escala de tensões.

Na figura VII.180 foi adicionada a escala de tensões com a respectiva tabela de cores.

A figura VII.181 apresenta o detalhe da distribuição de tensões nos furos inferiores.

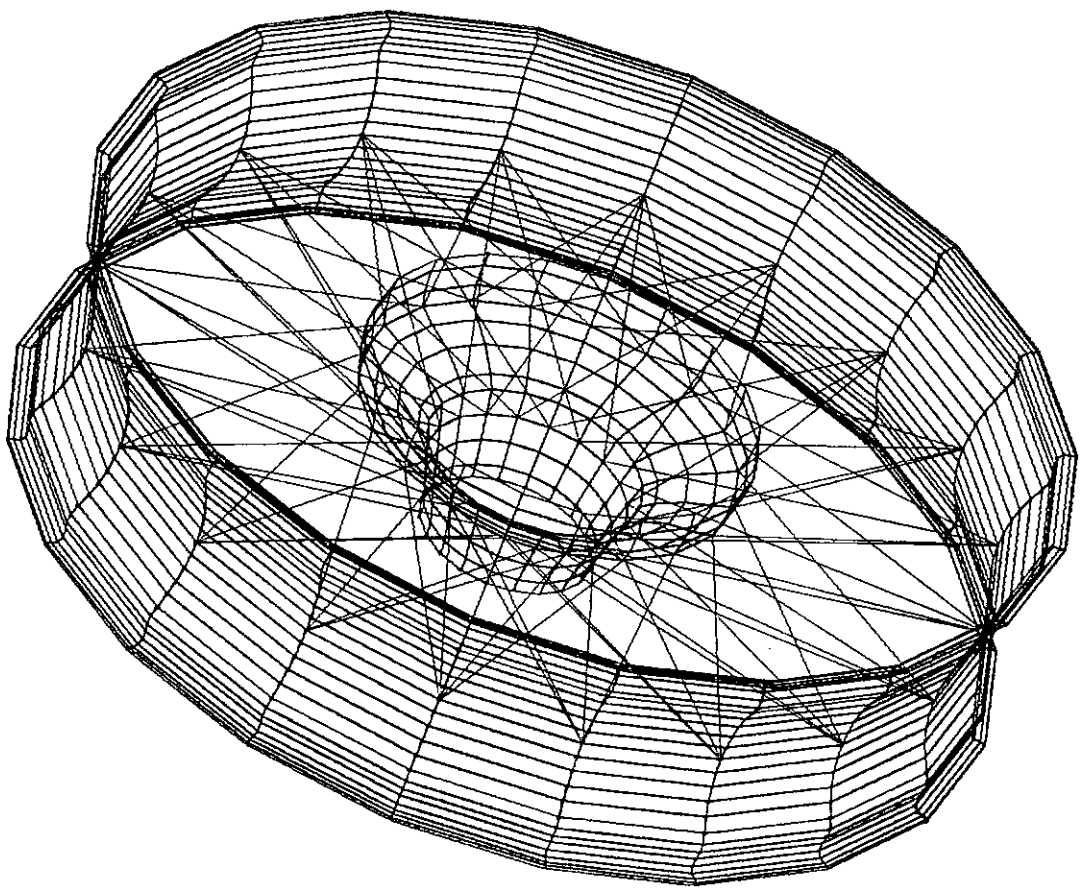


FIG VII.113.

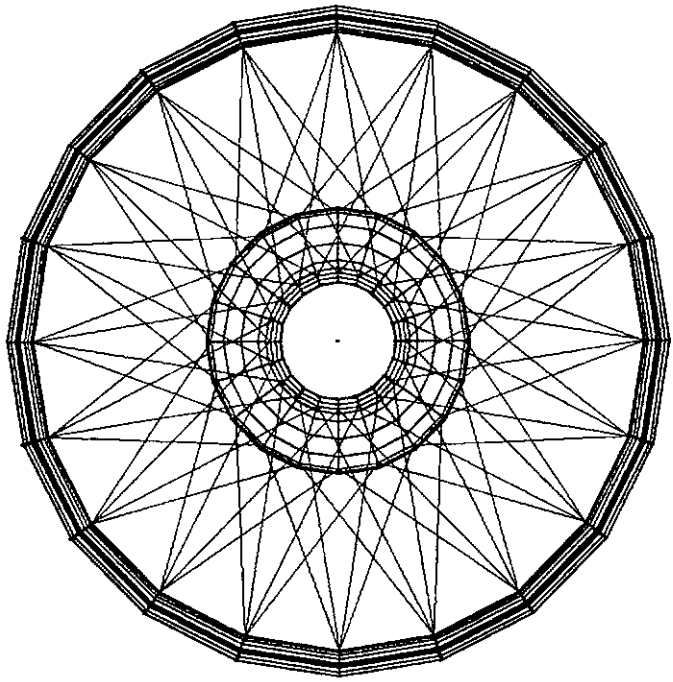


FIG VII.114.

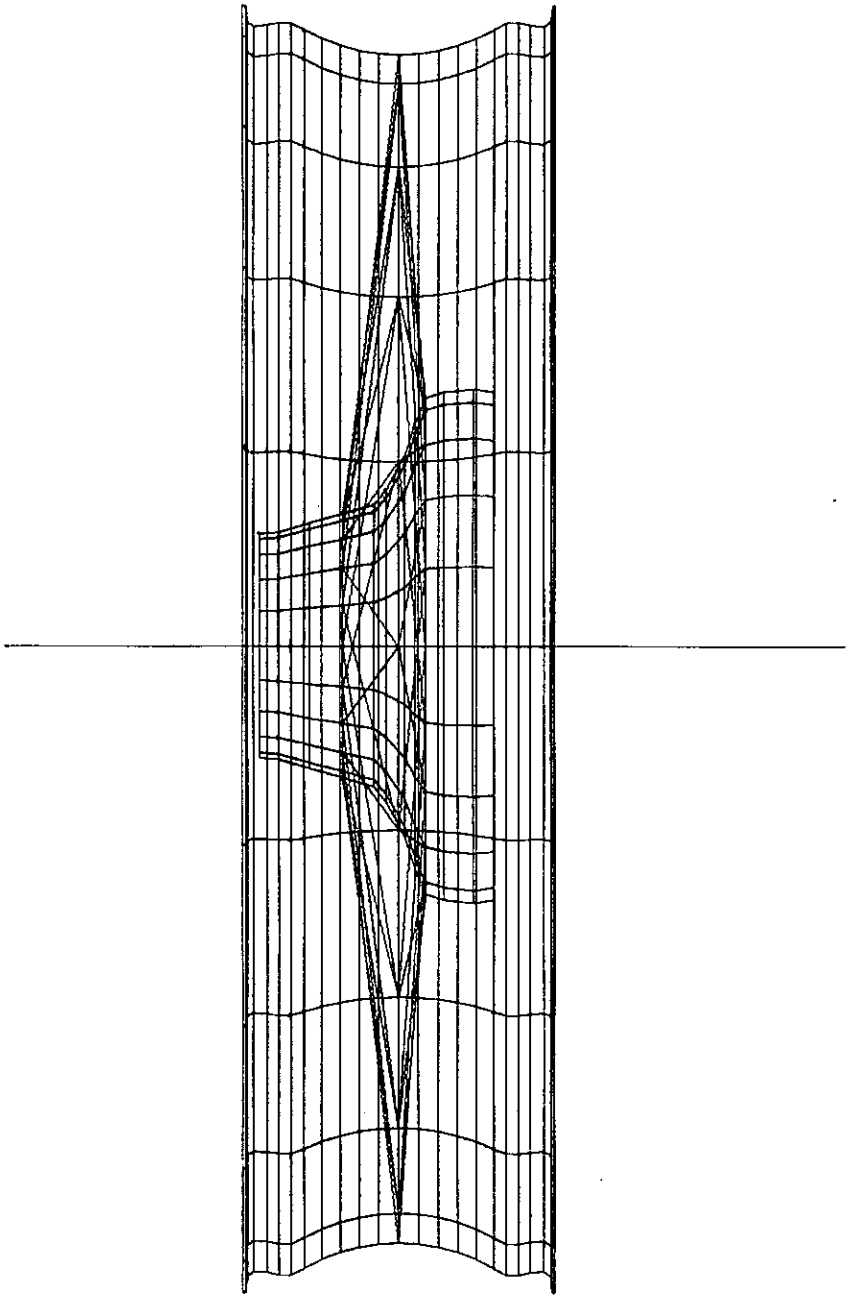


FIG VII.115.

FIG VII.116.

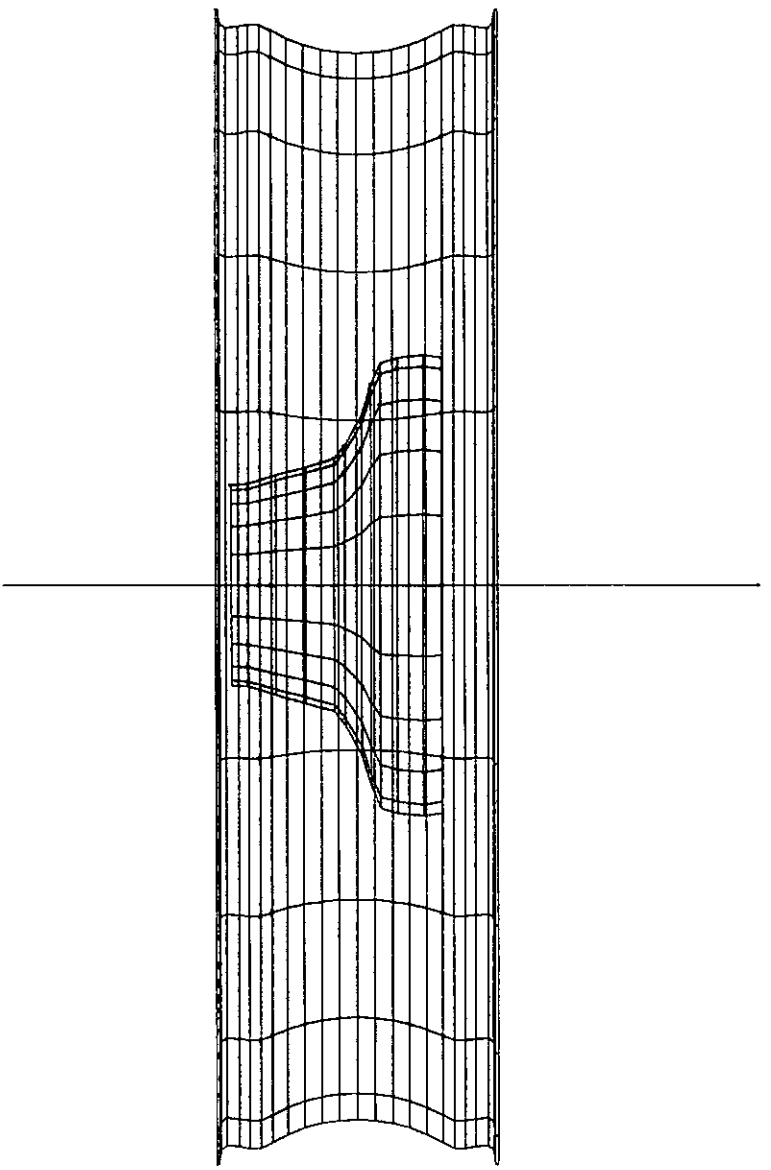


FIG VII.116.

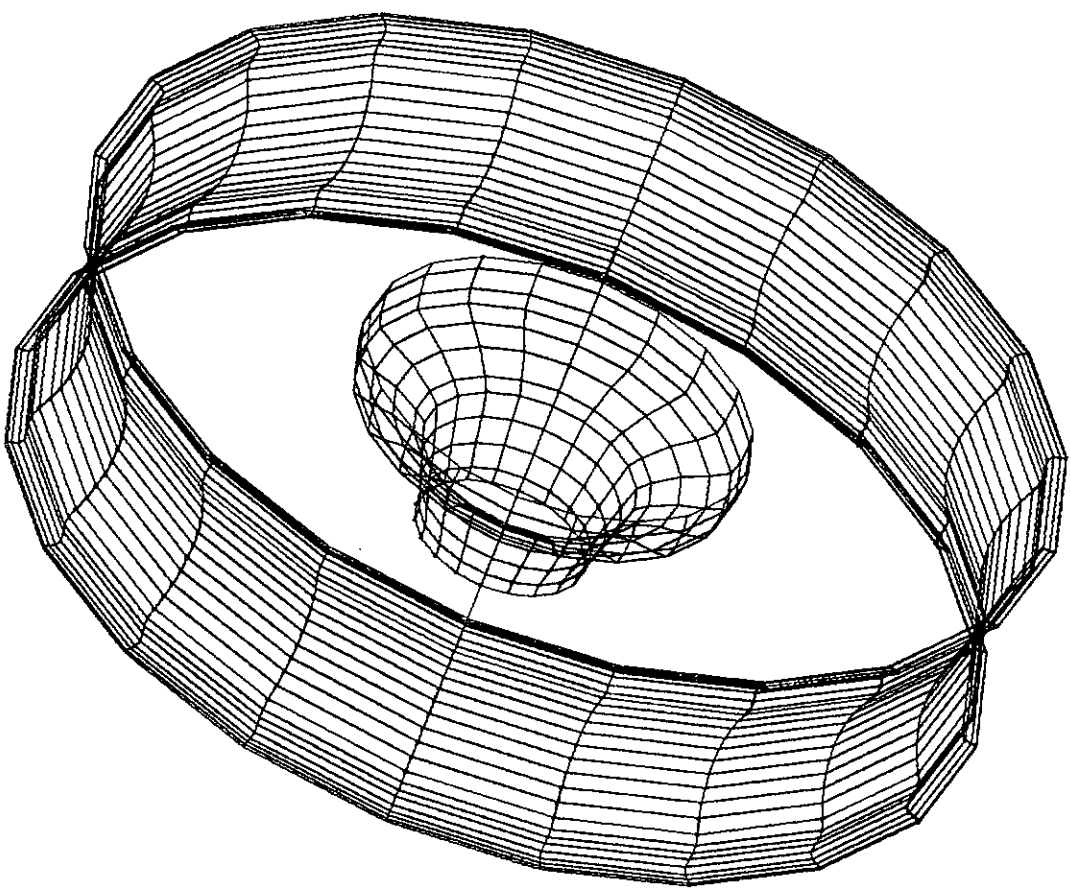


FIG VII.117.

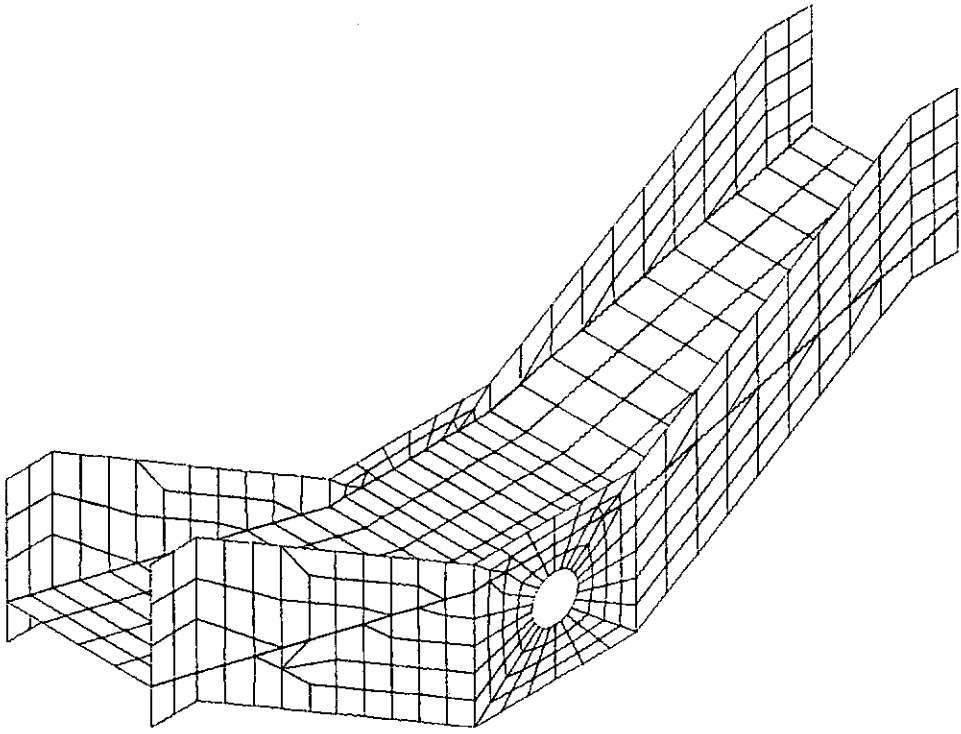


FIG VII.118.

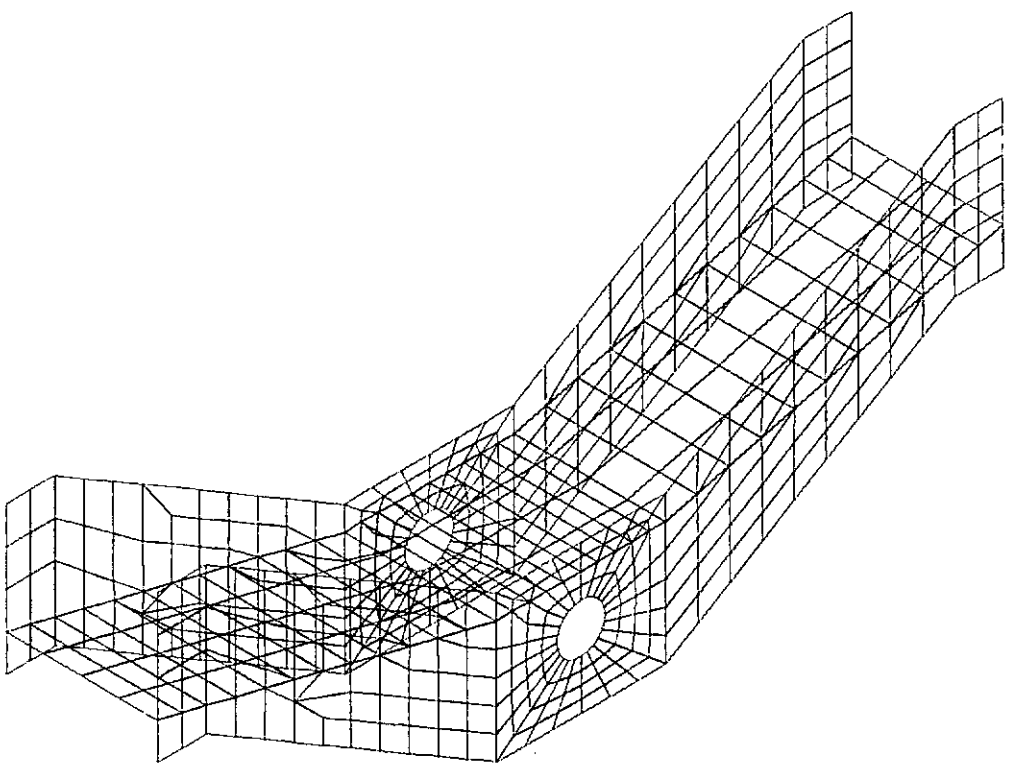


FIG VII.119.

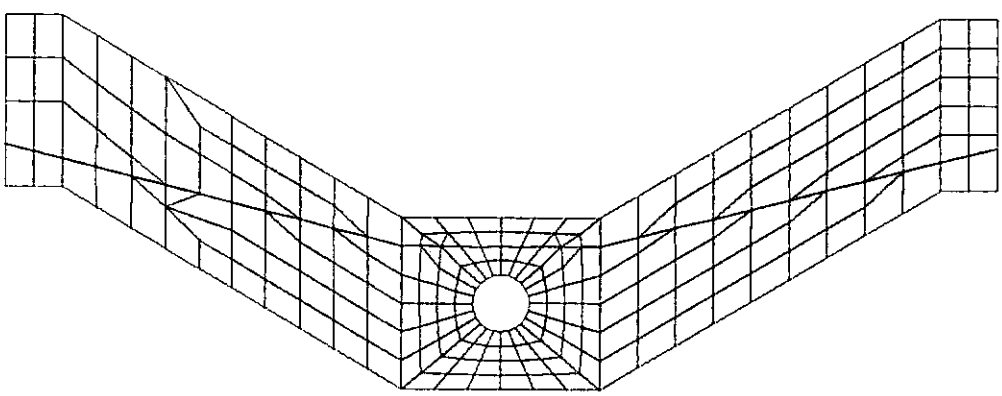


FIG VII.120.

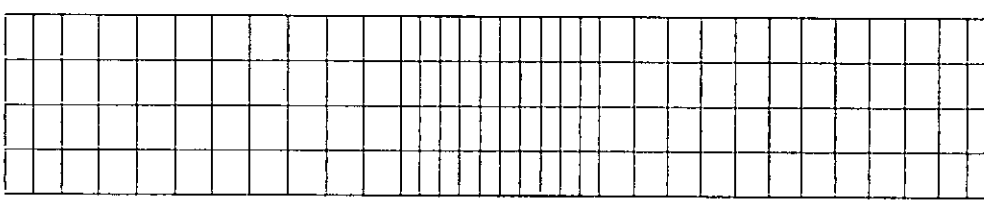


FIG VII.121.

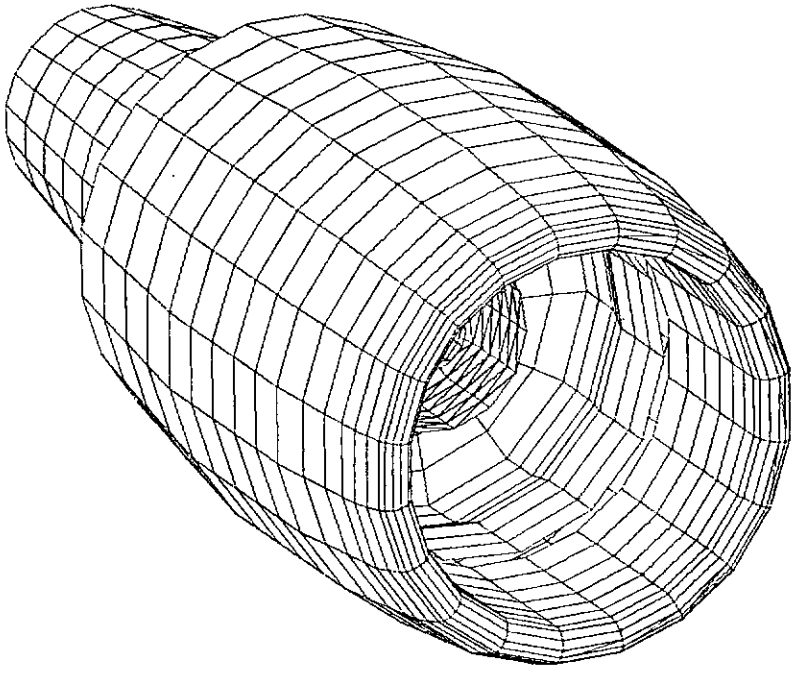


FIG VII.122.

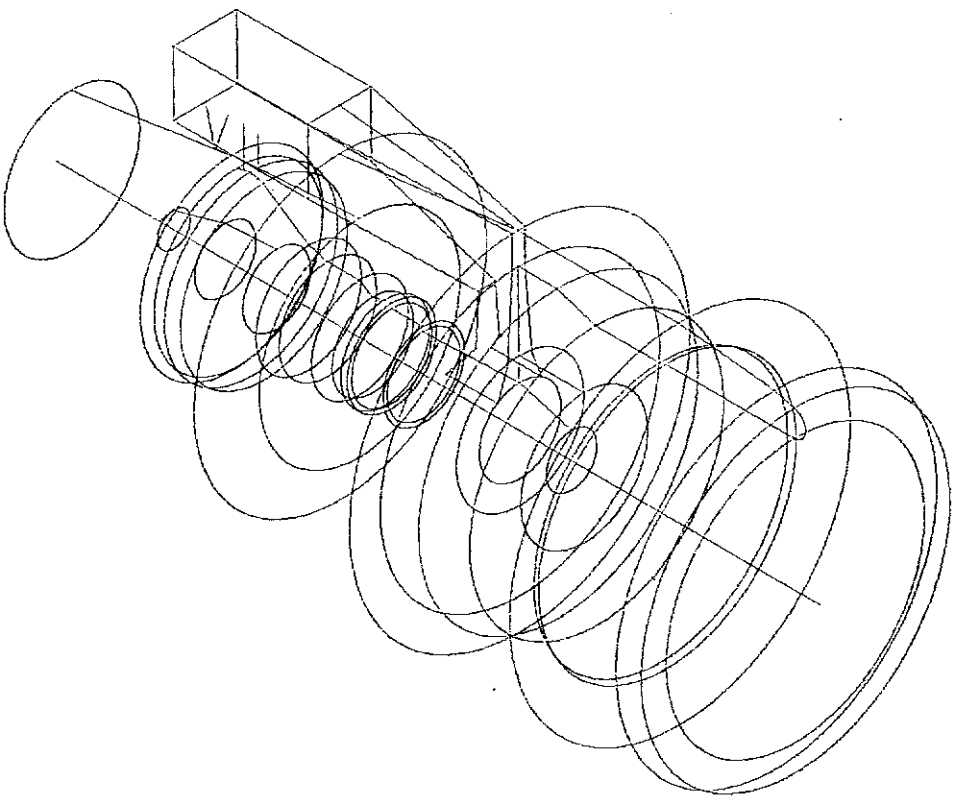


FIG VII.123.

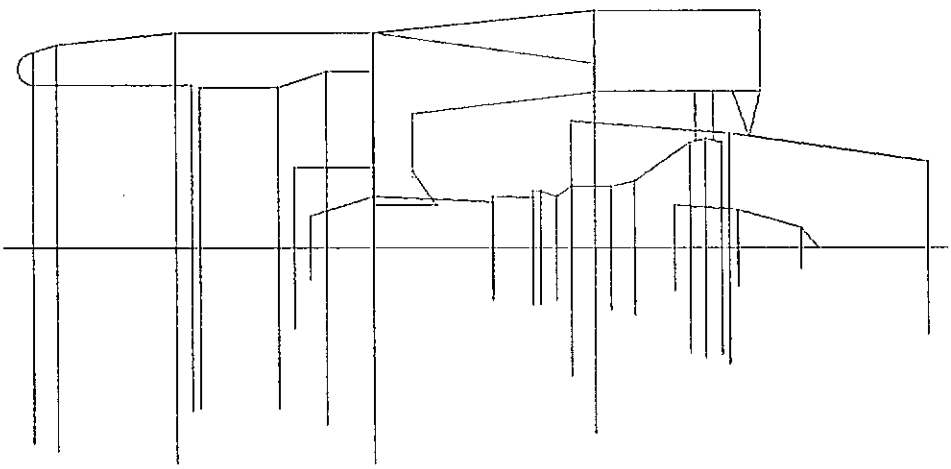


FIG VII.124.

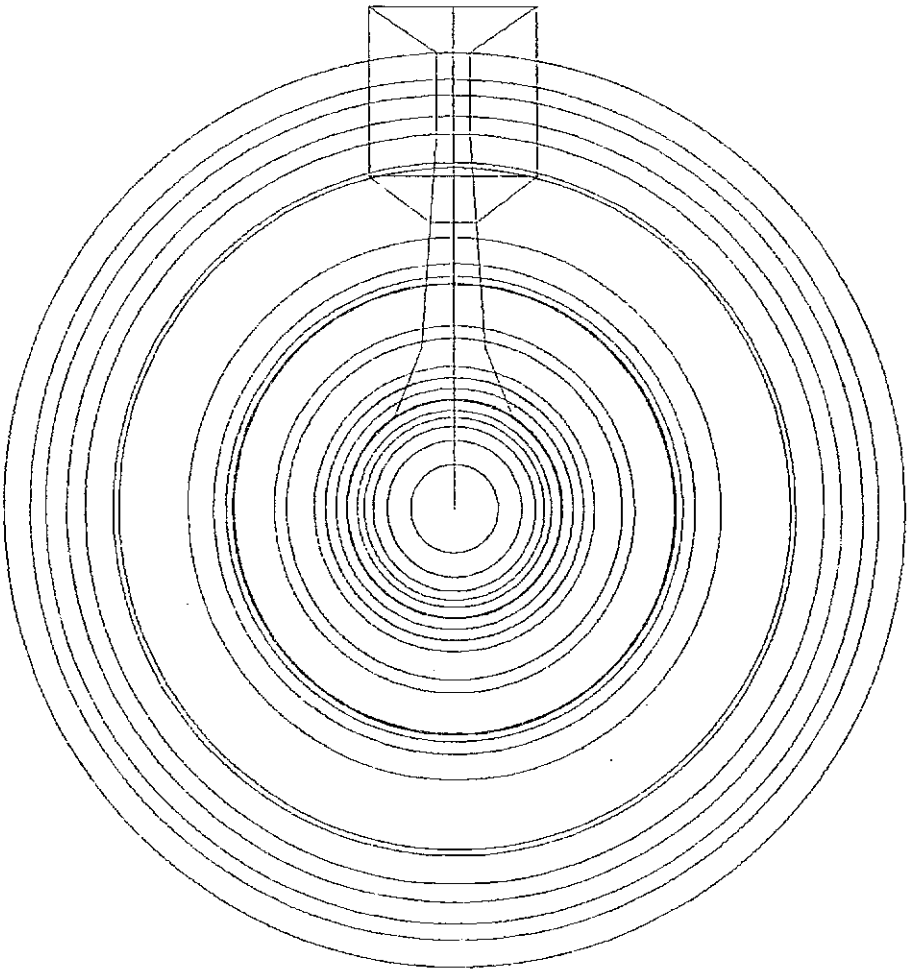


FIG VII.125.

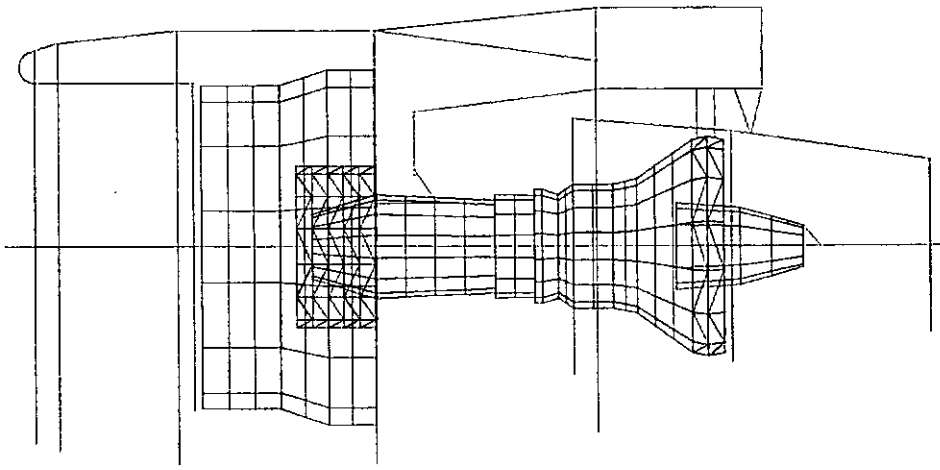


FIG VII.126.

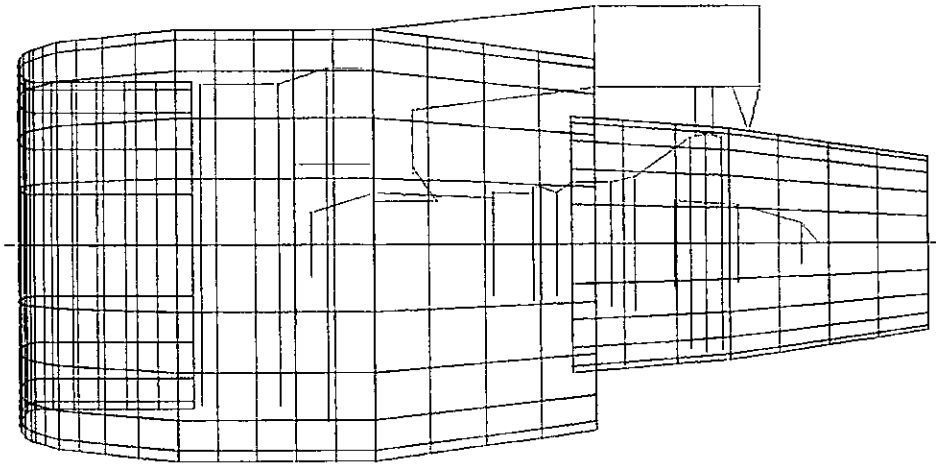


FIG VII.127.

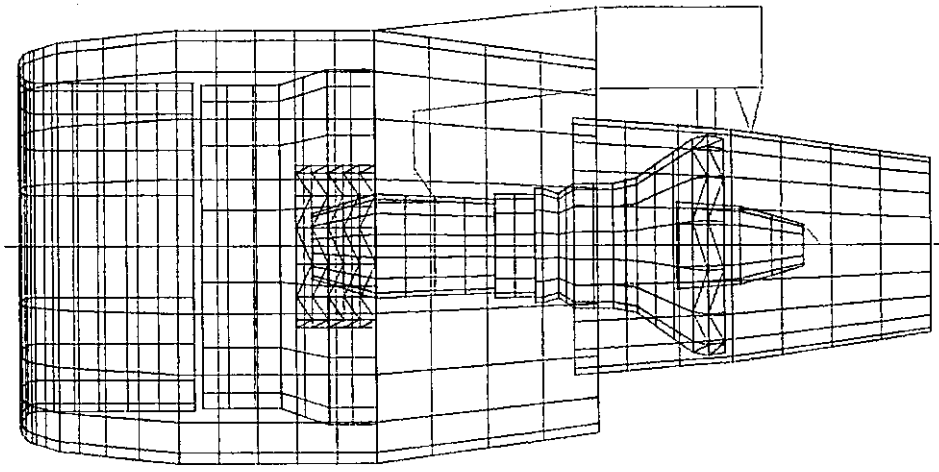


FIG VII.128.

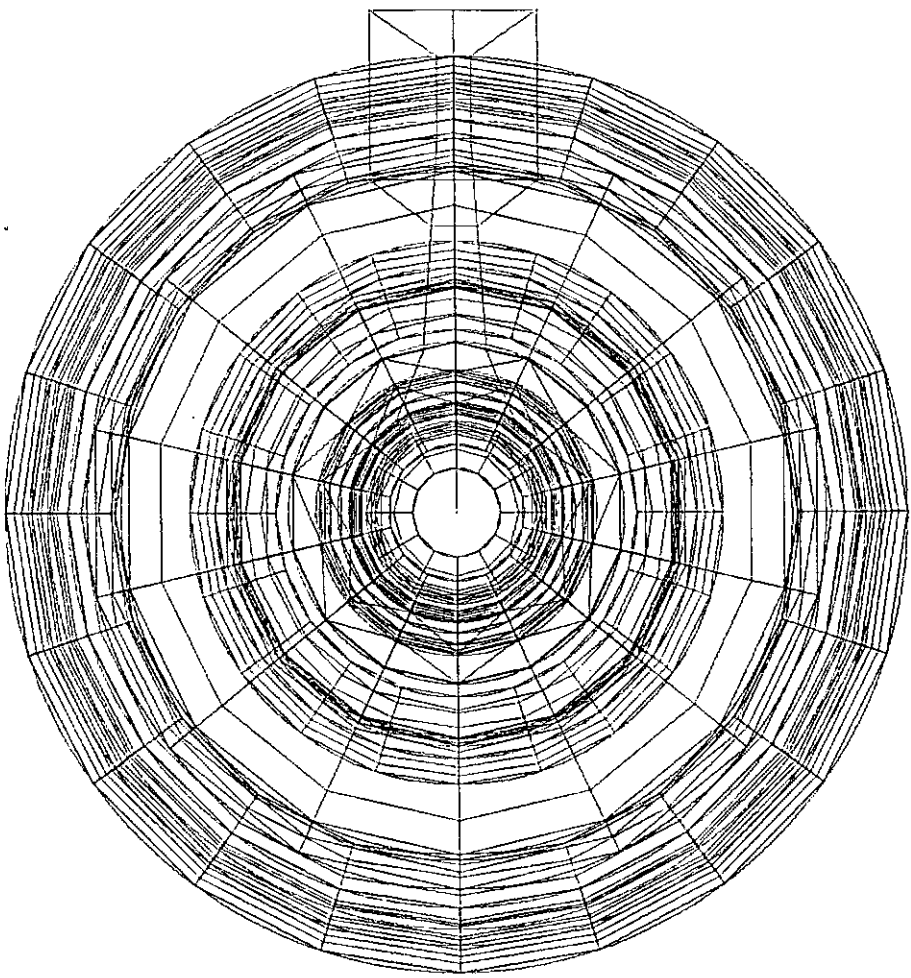


FIG VII.129.

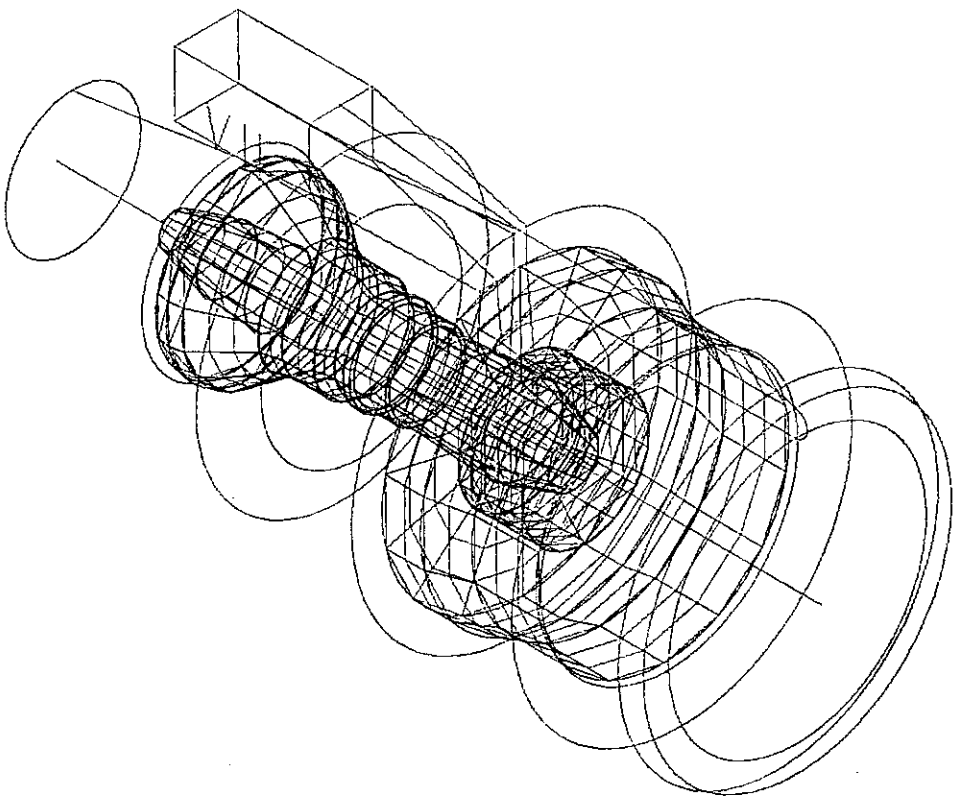


FIG VII.130.

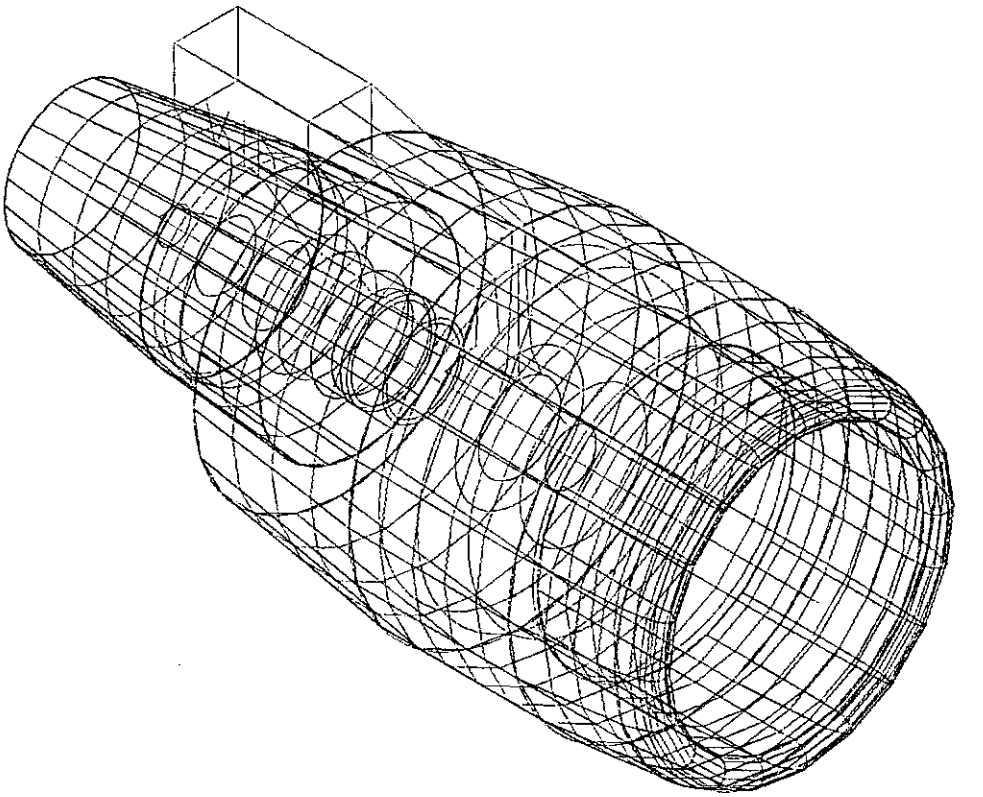


FIG VII.131.

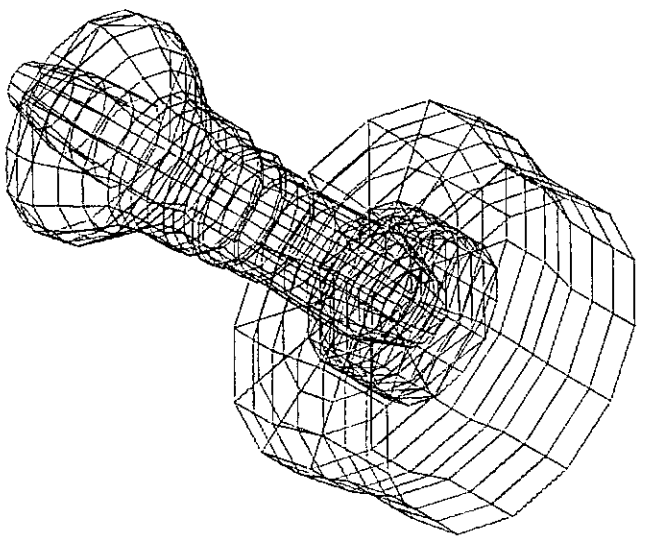


FIG VII.132.

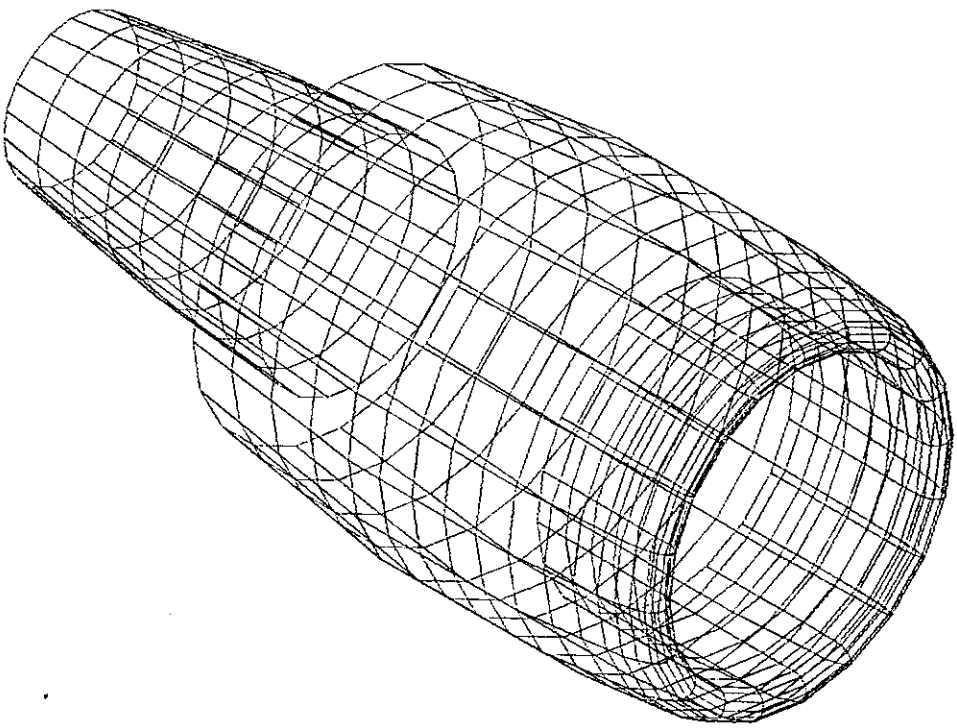


FIG VII.133.

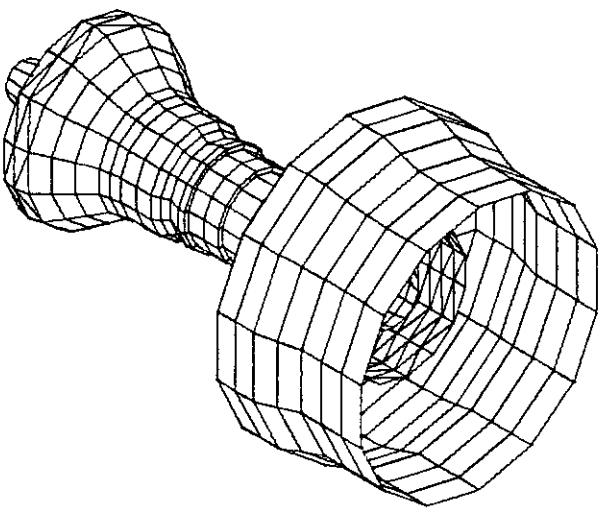


FIG VII.134.

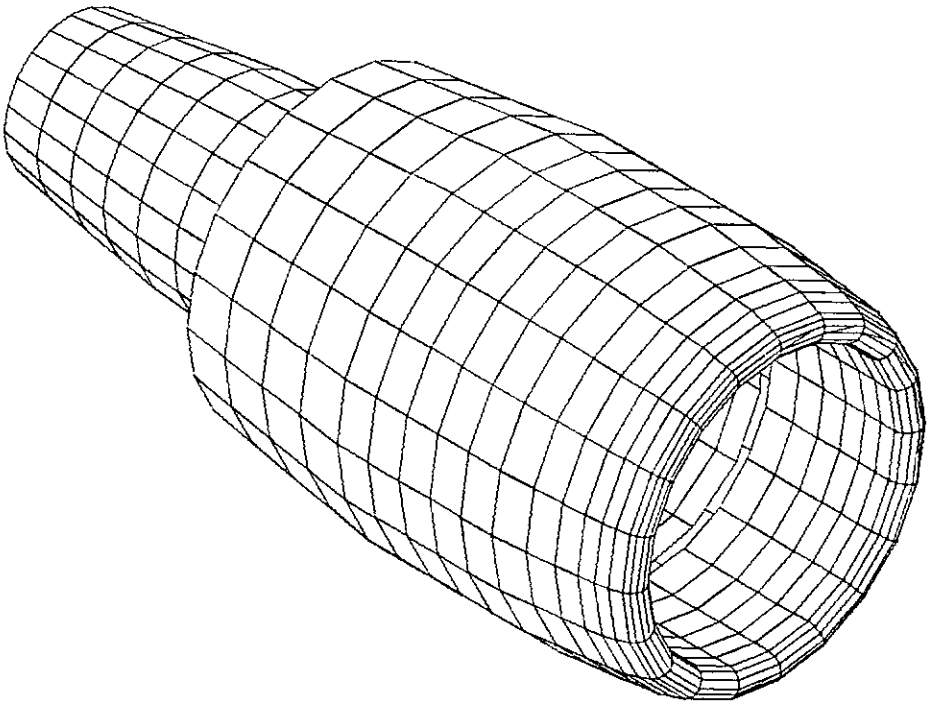


FIG VII.135.

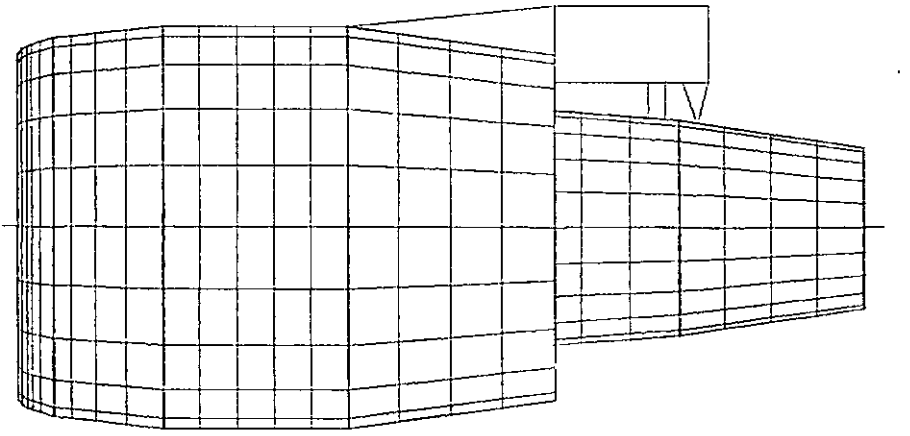


FIG VII.136.

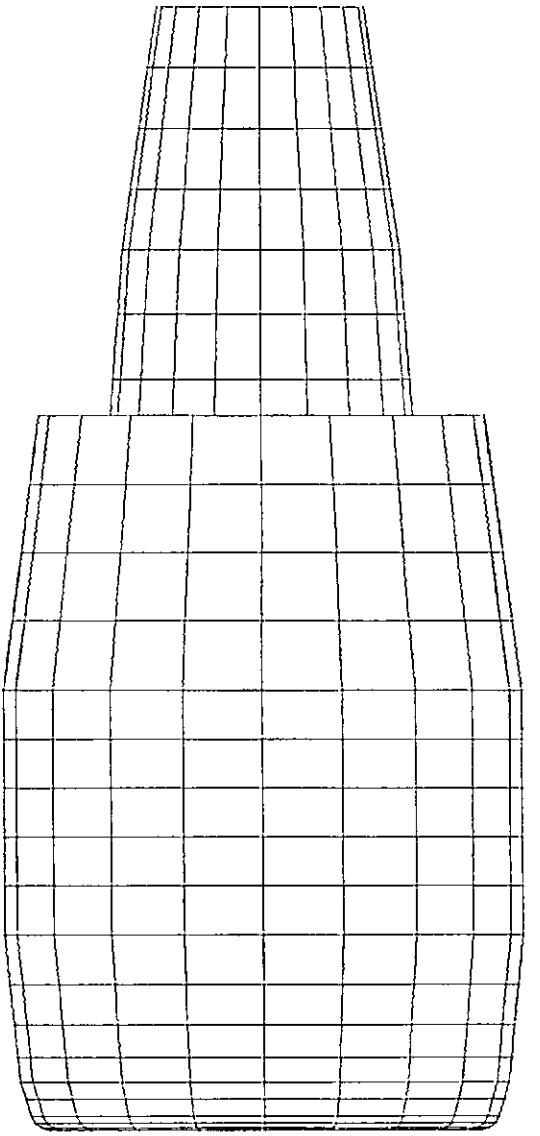


FIG VII.137.

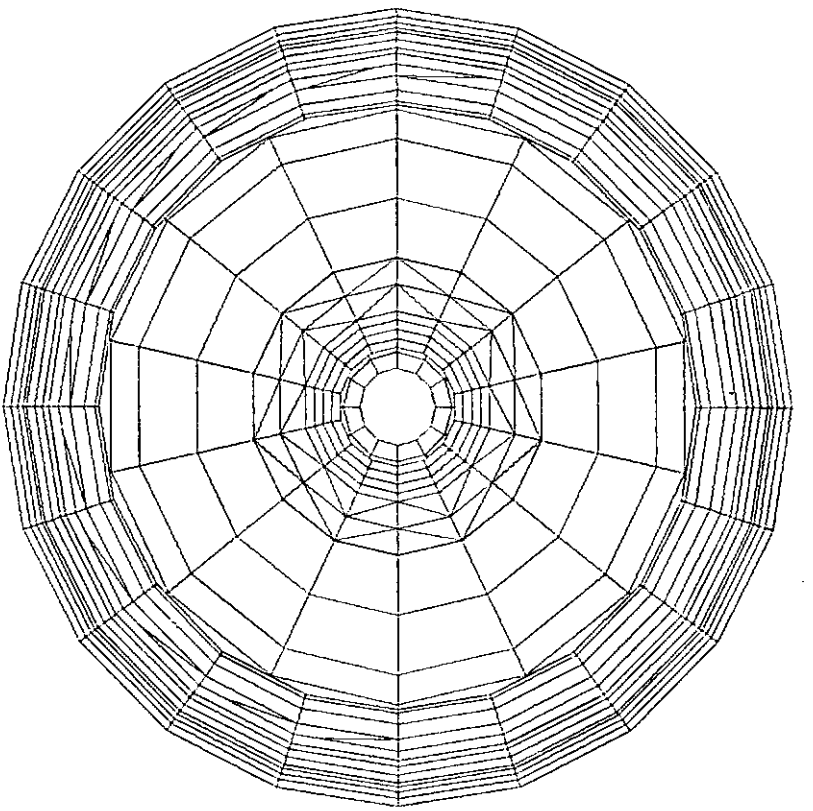


FIG. VII.138.

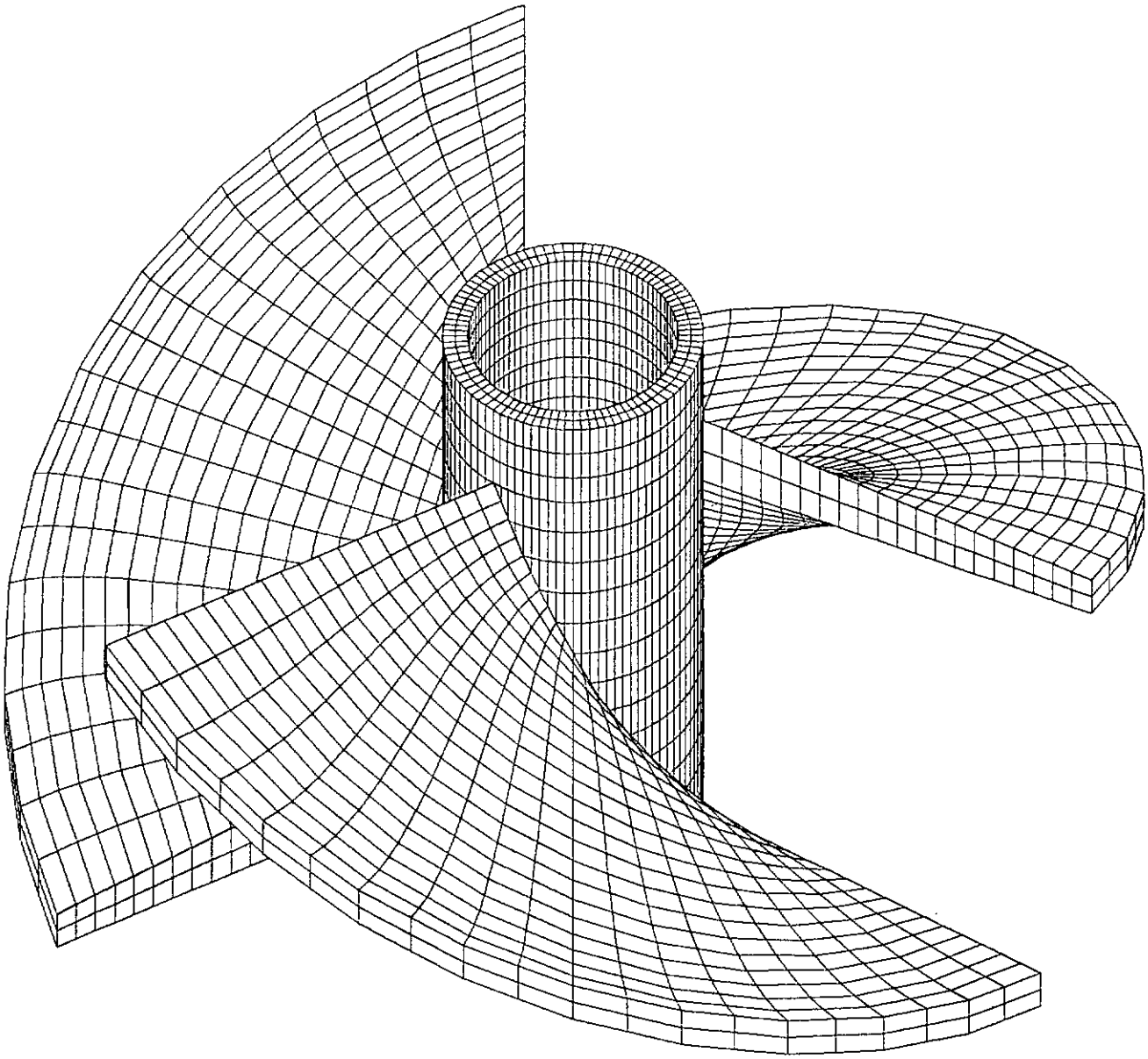


FIG VII.139.

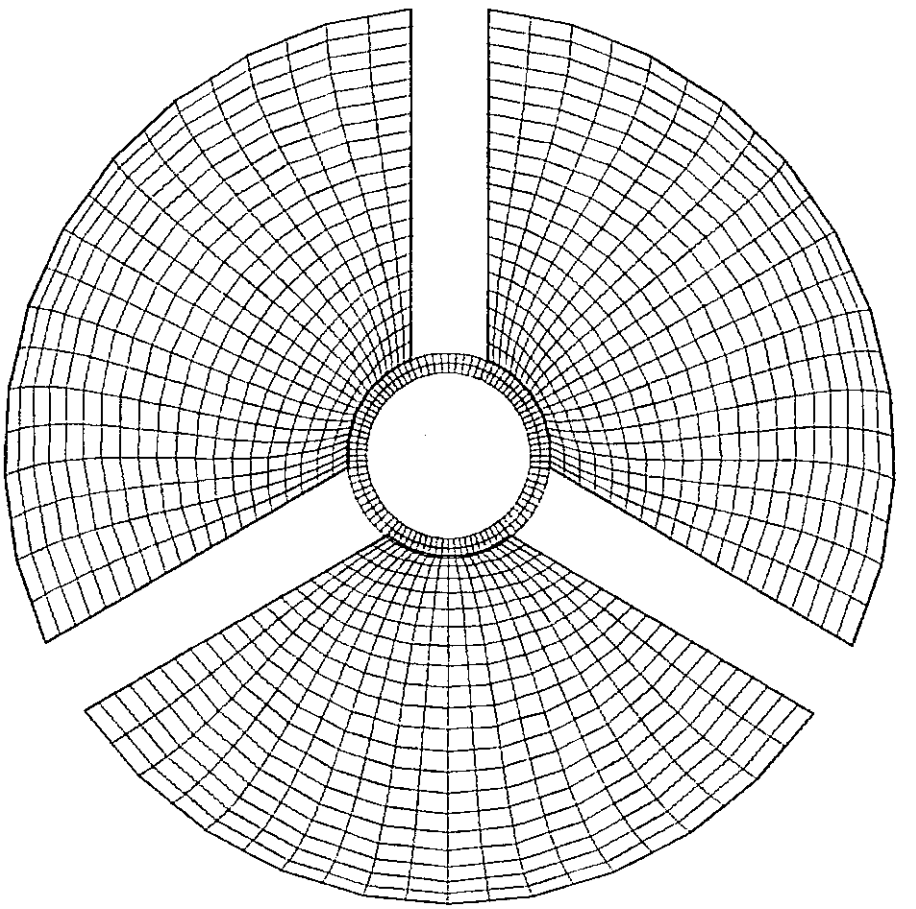


FIG VII.140.

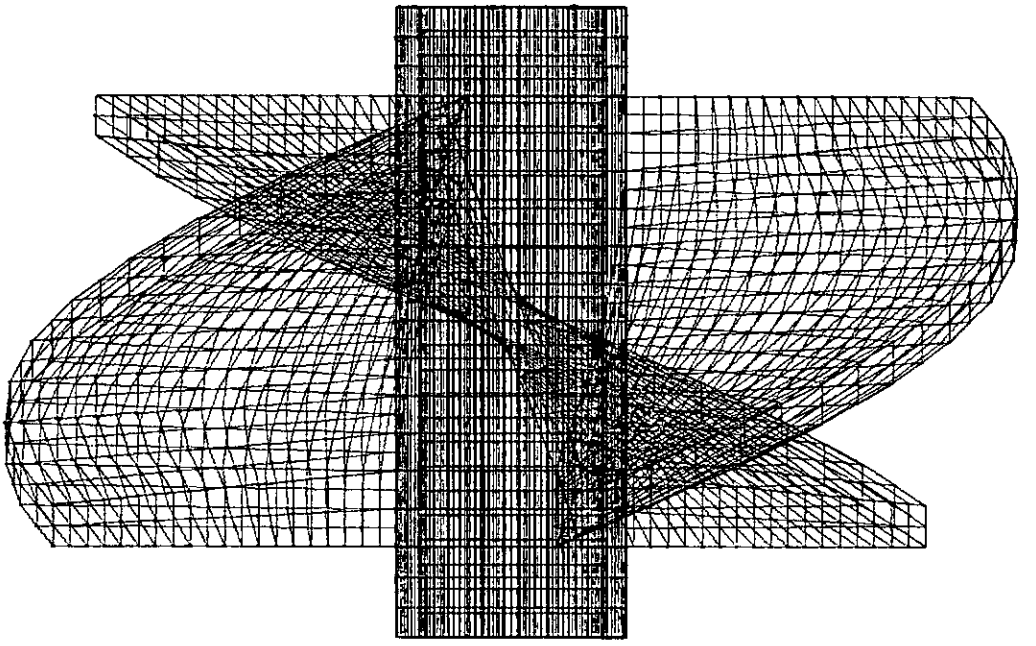


FIG VII.141.

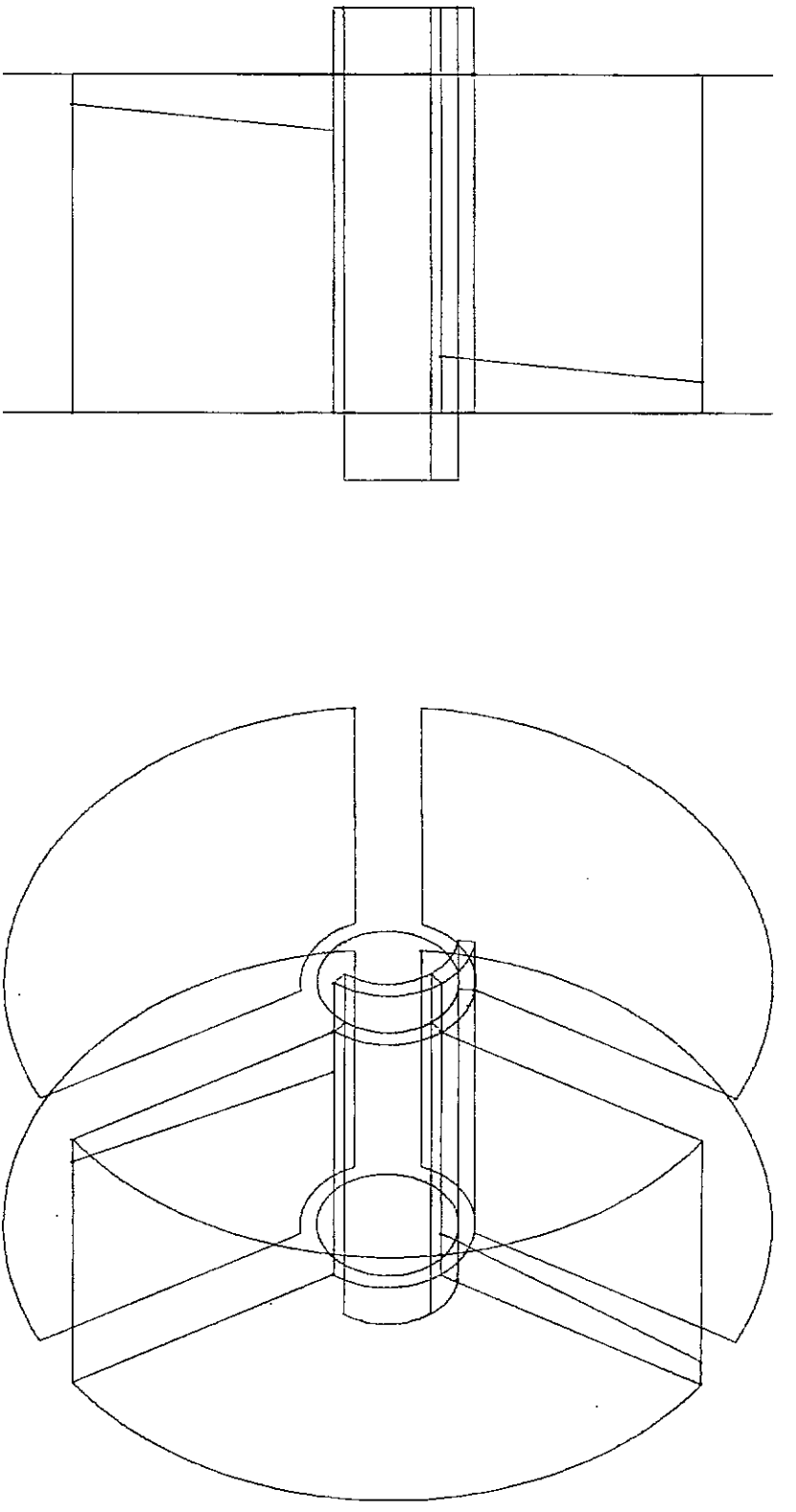
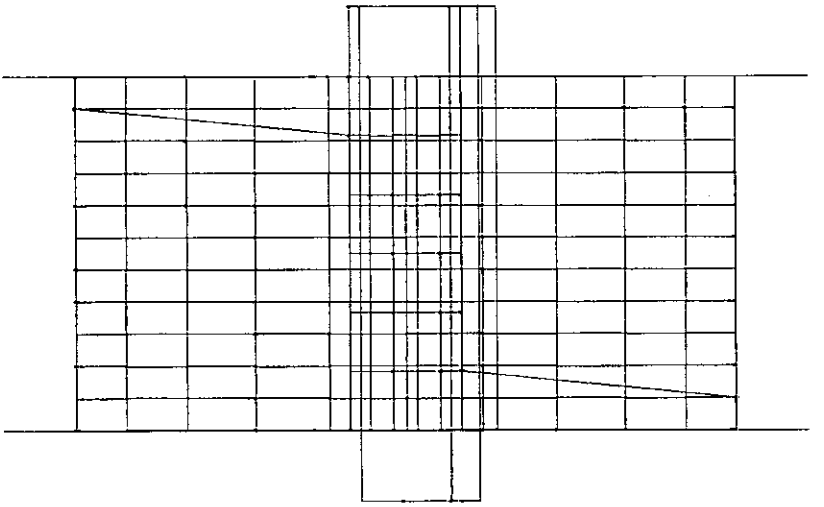


FIG VII.142.



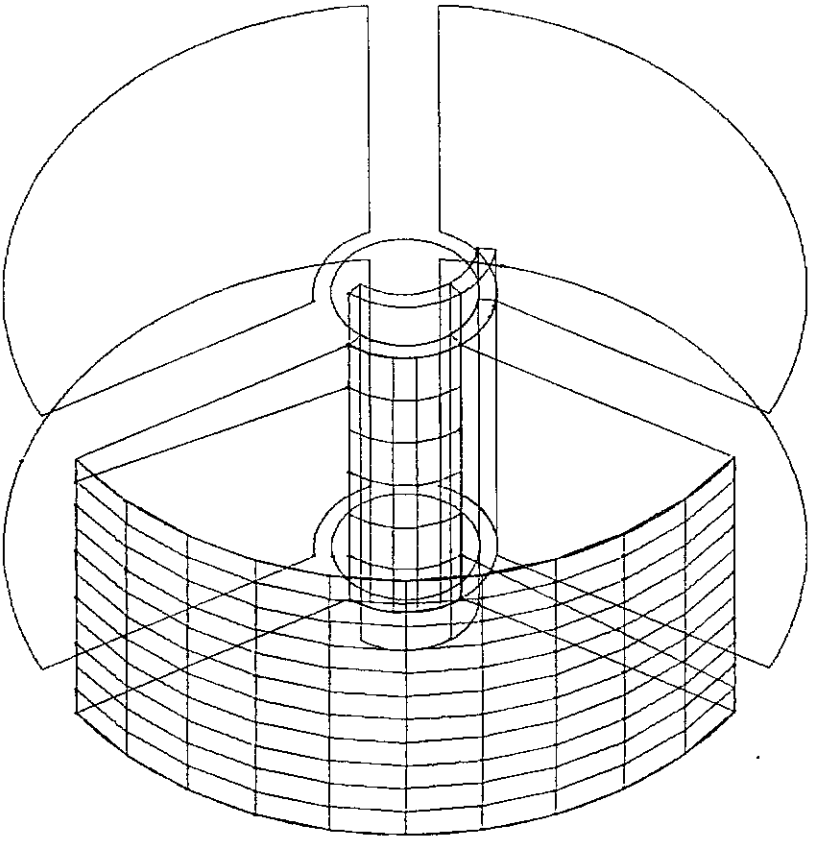
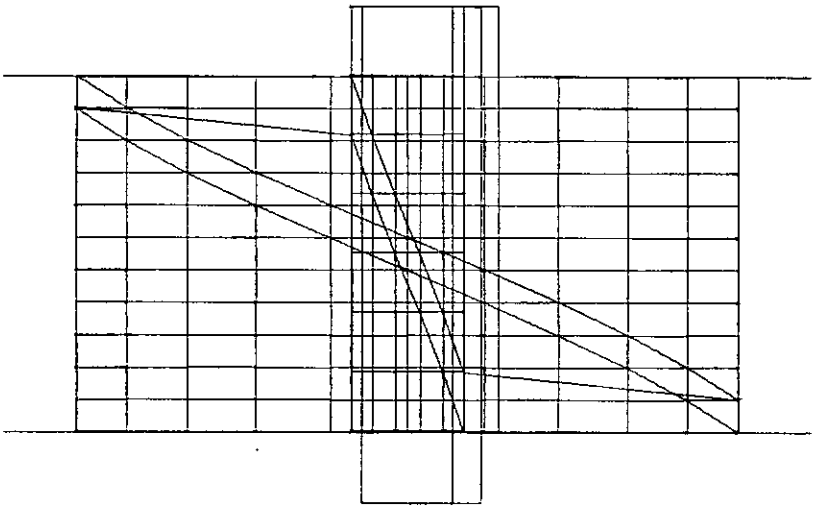


FIG VII.143.



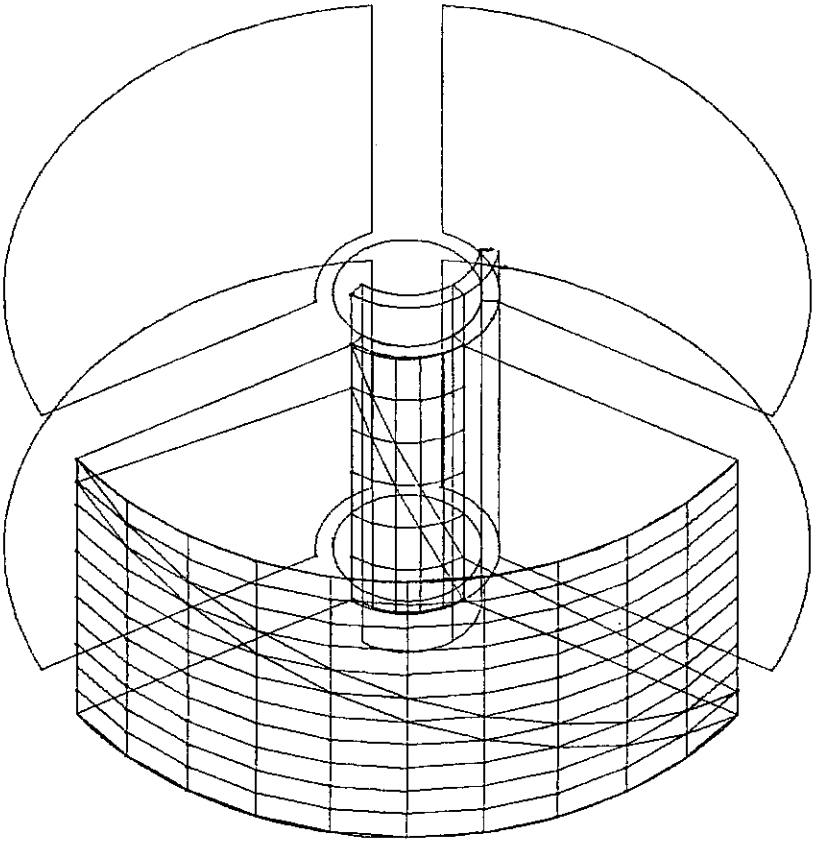


FIG VII.144.

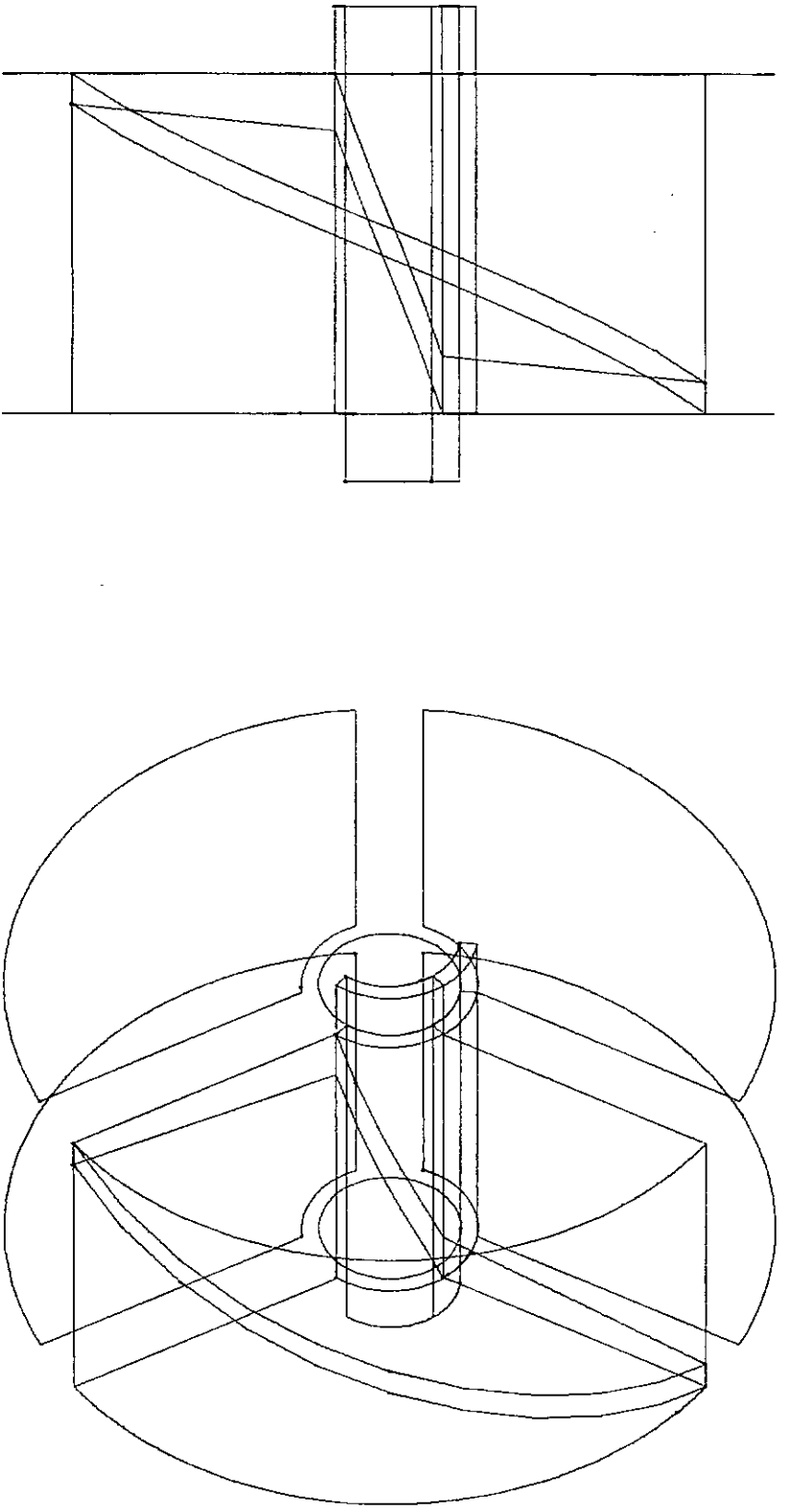


FIG VII.145.

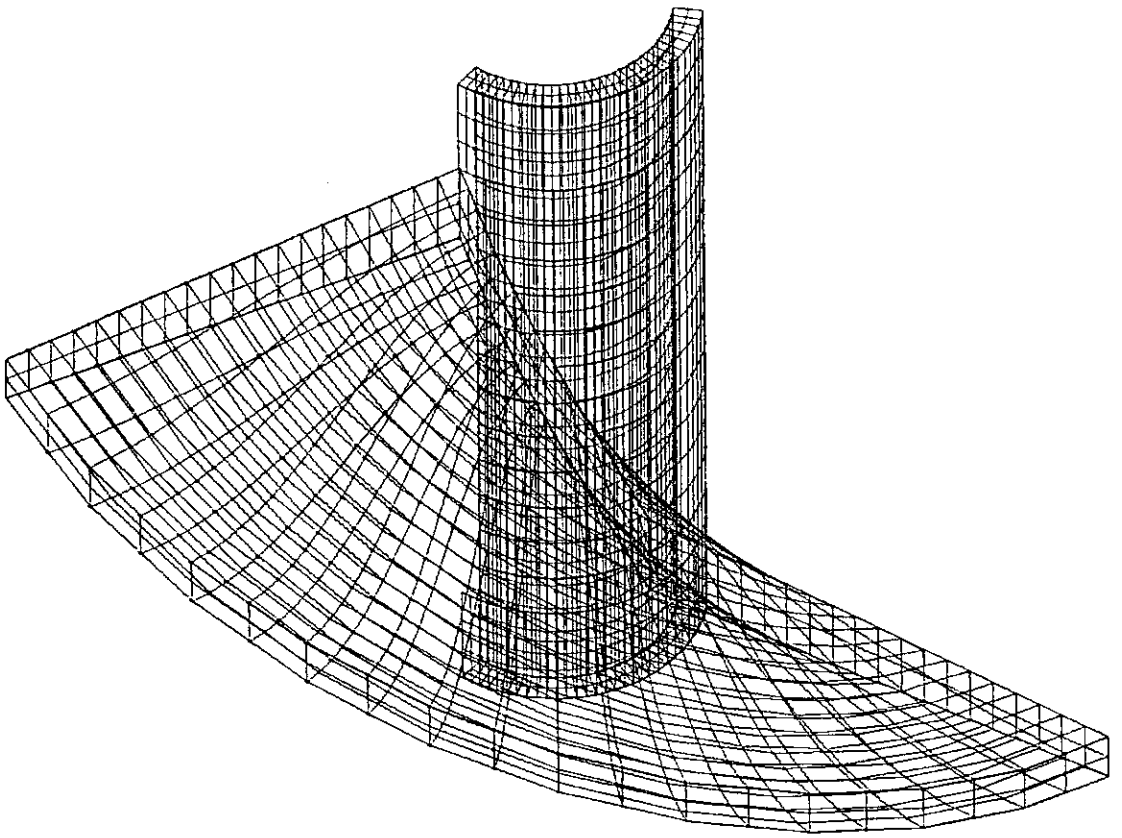


FIG VII.146.

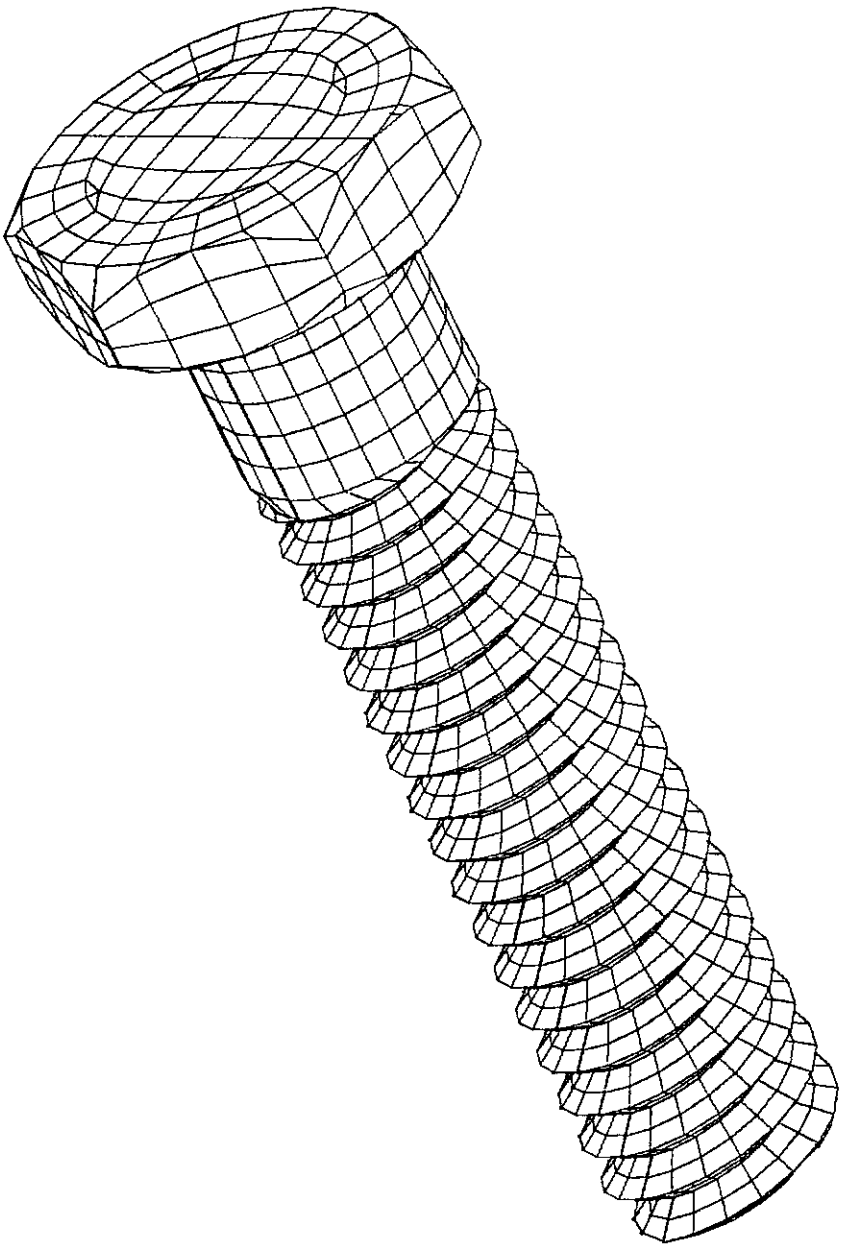
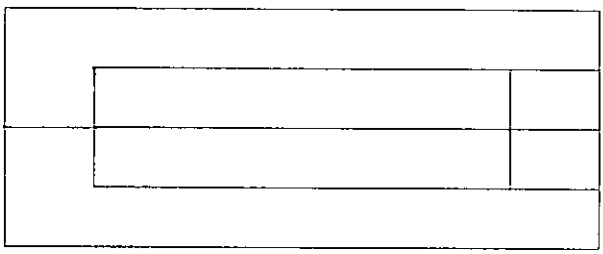


FIG VII.147.



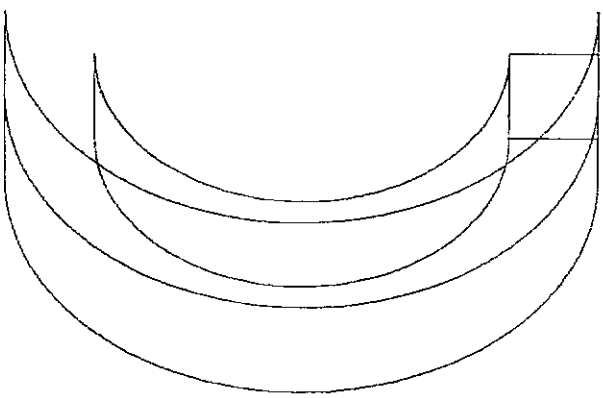
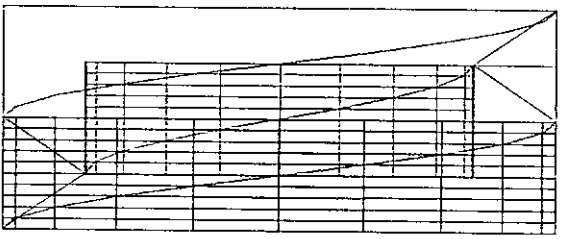


FIG VII.148.



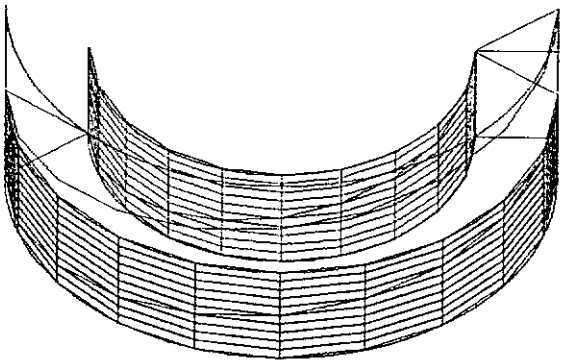
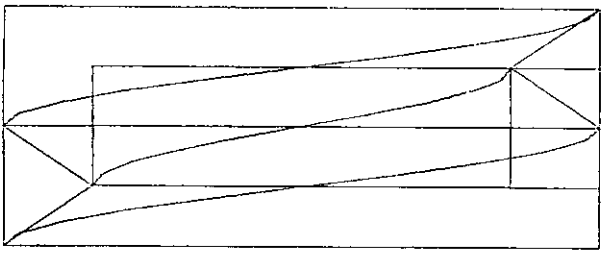


FIG VII.149.



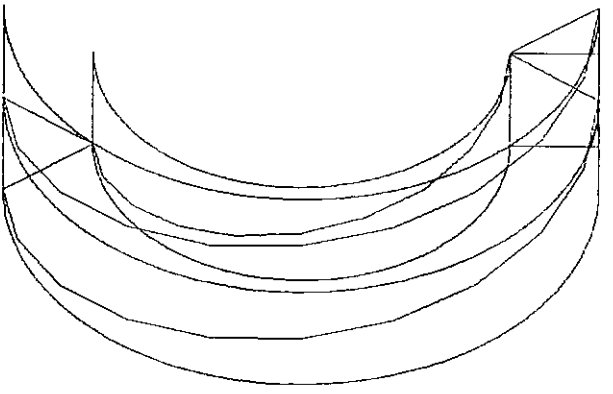


FIG VII.150.

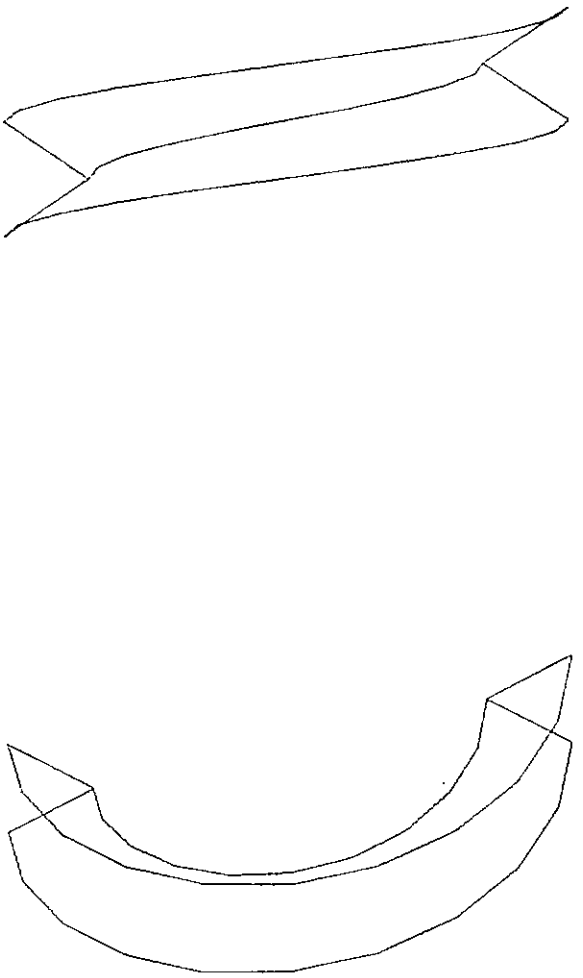


FIG VII.151.

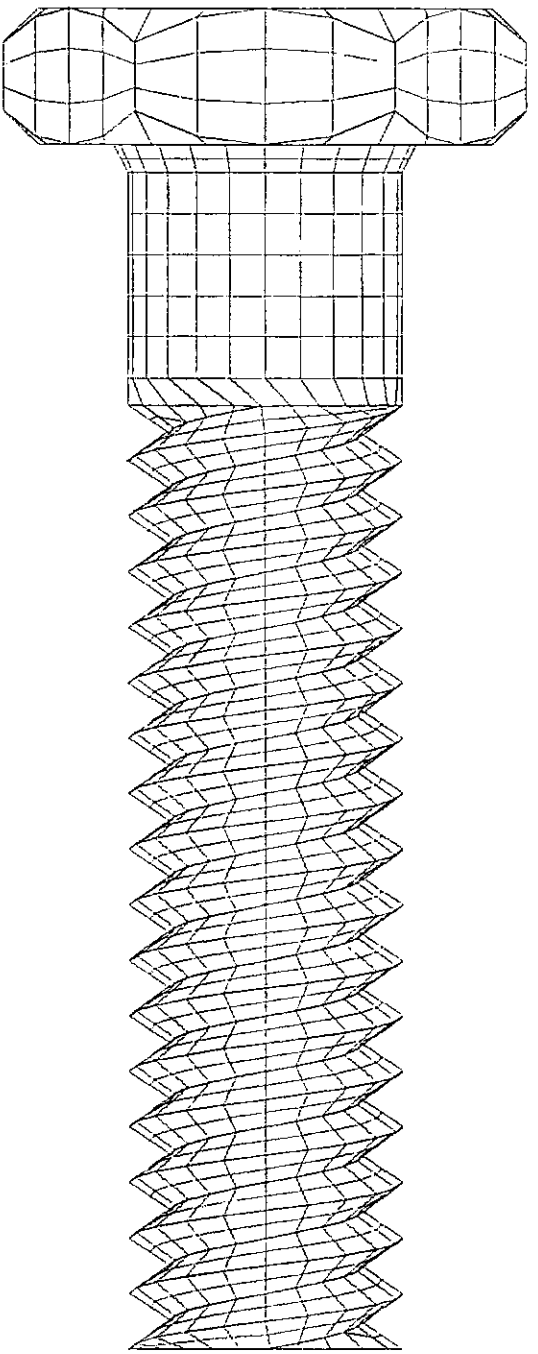


FIG VII.152.

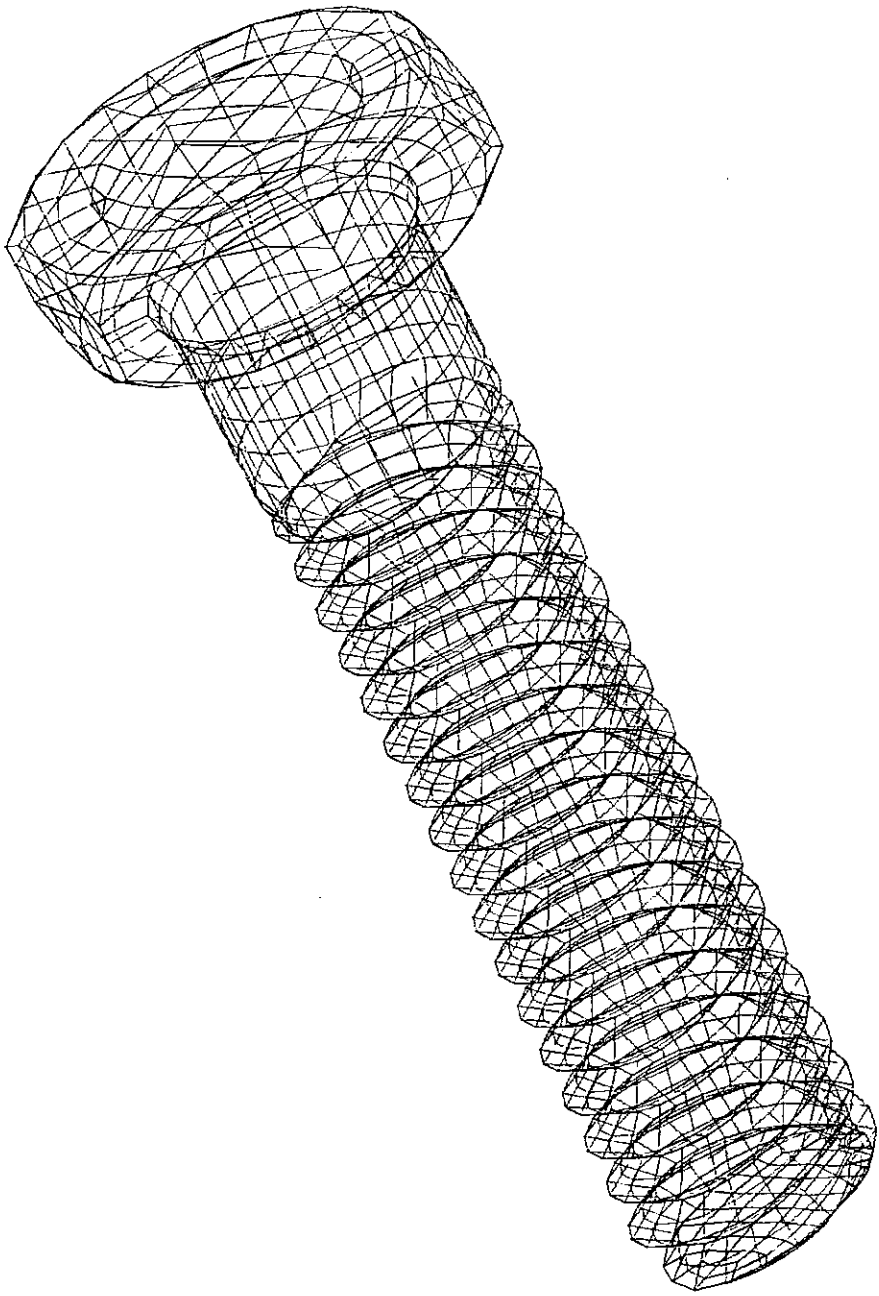


FIG VII.153.

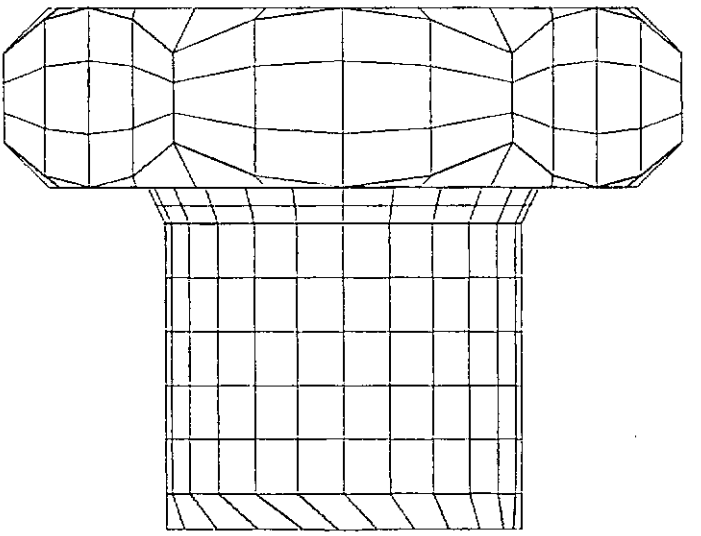


FIG VII.154.

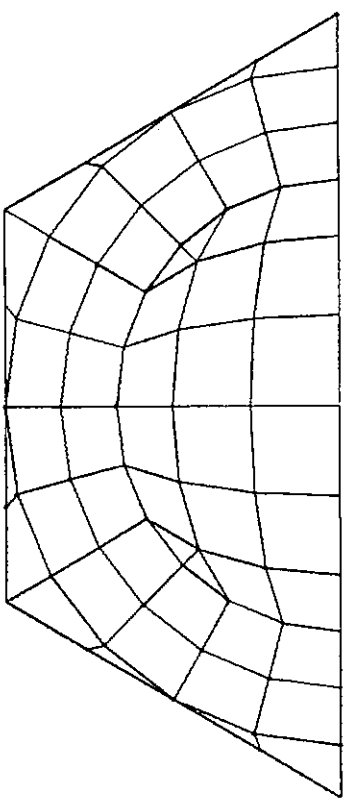


FIG VII.155.

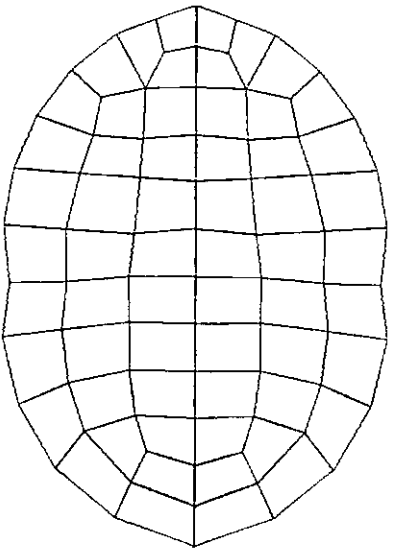


FIG VII.156.

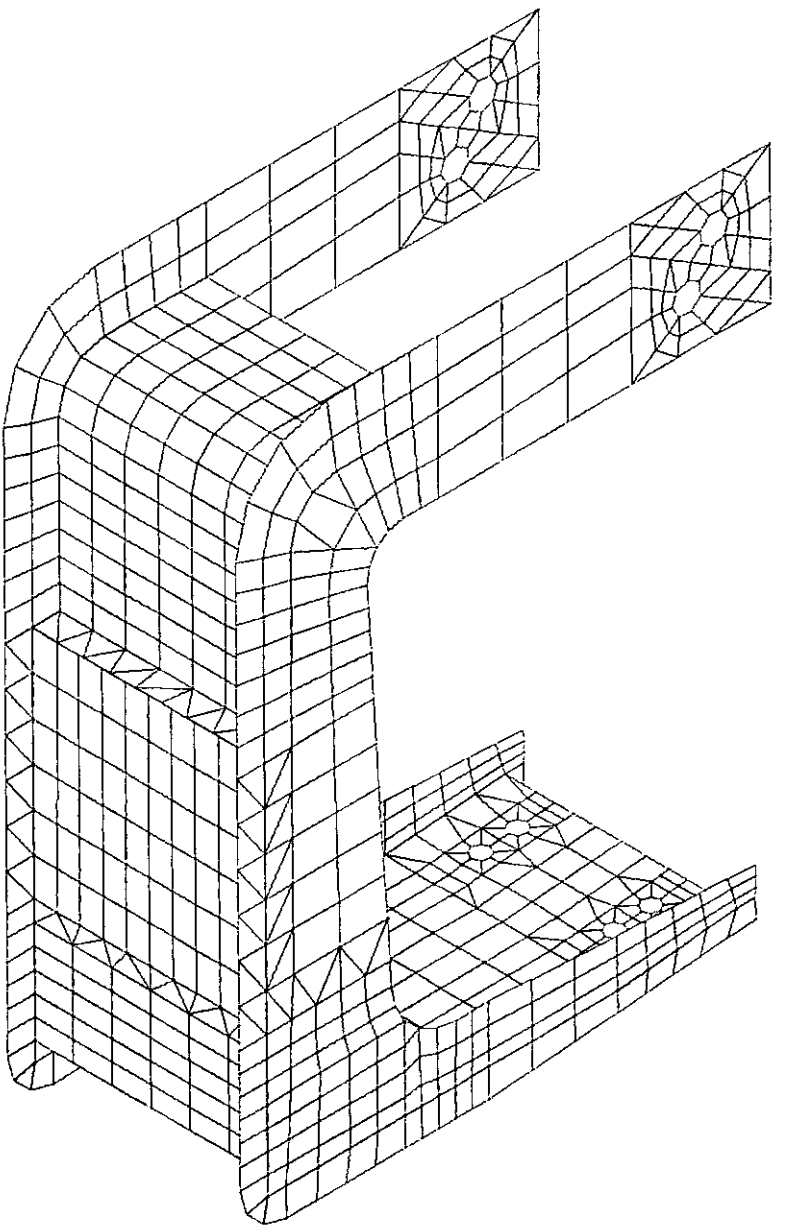


FIG VII.157.

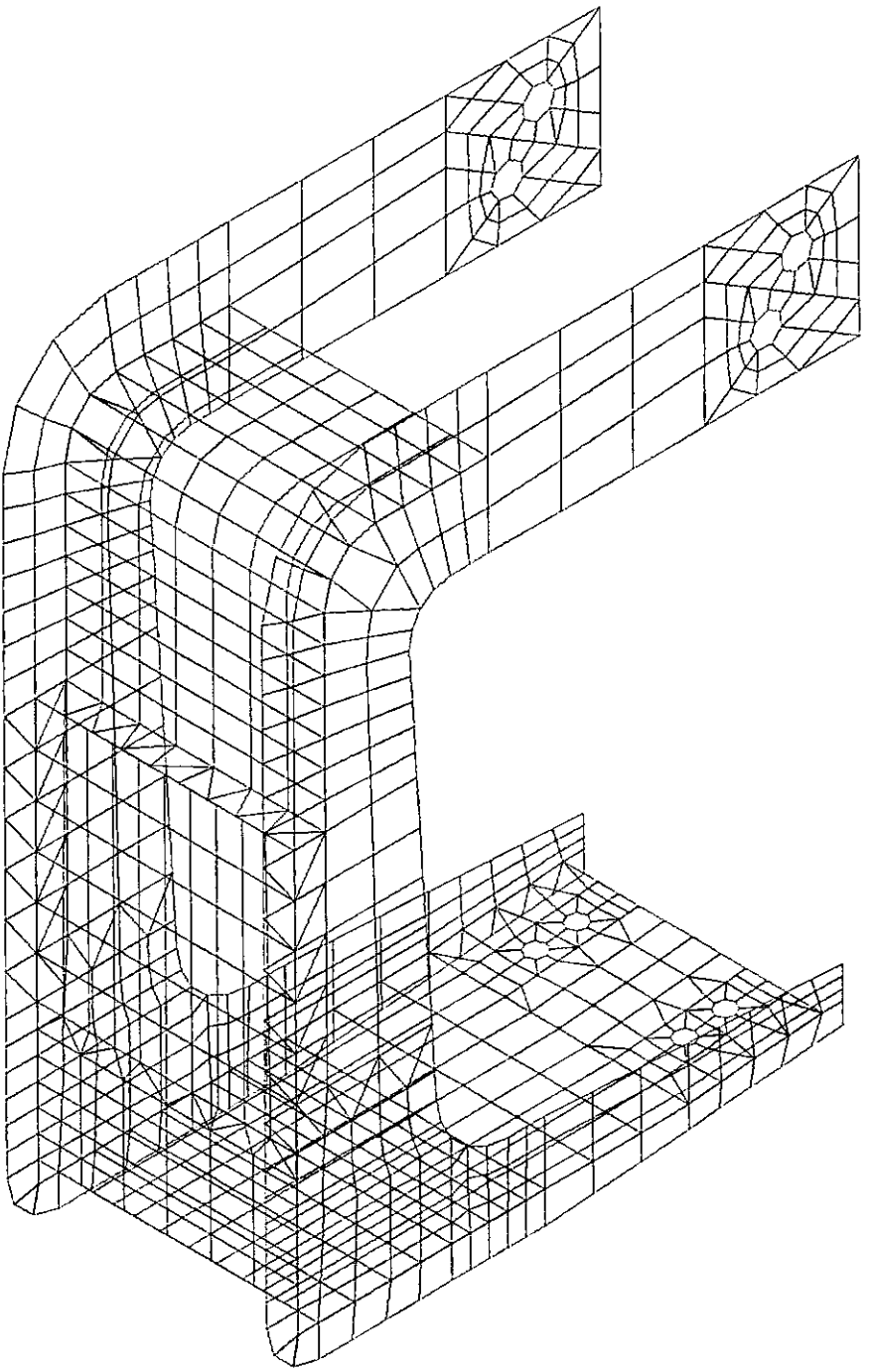


FIG VII.158.

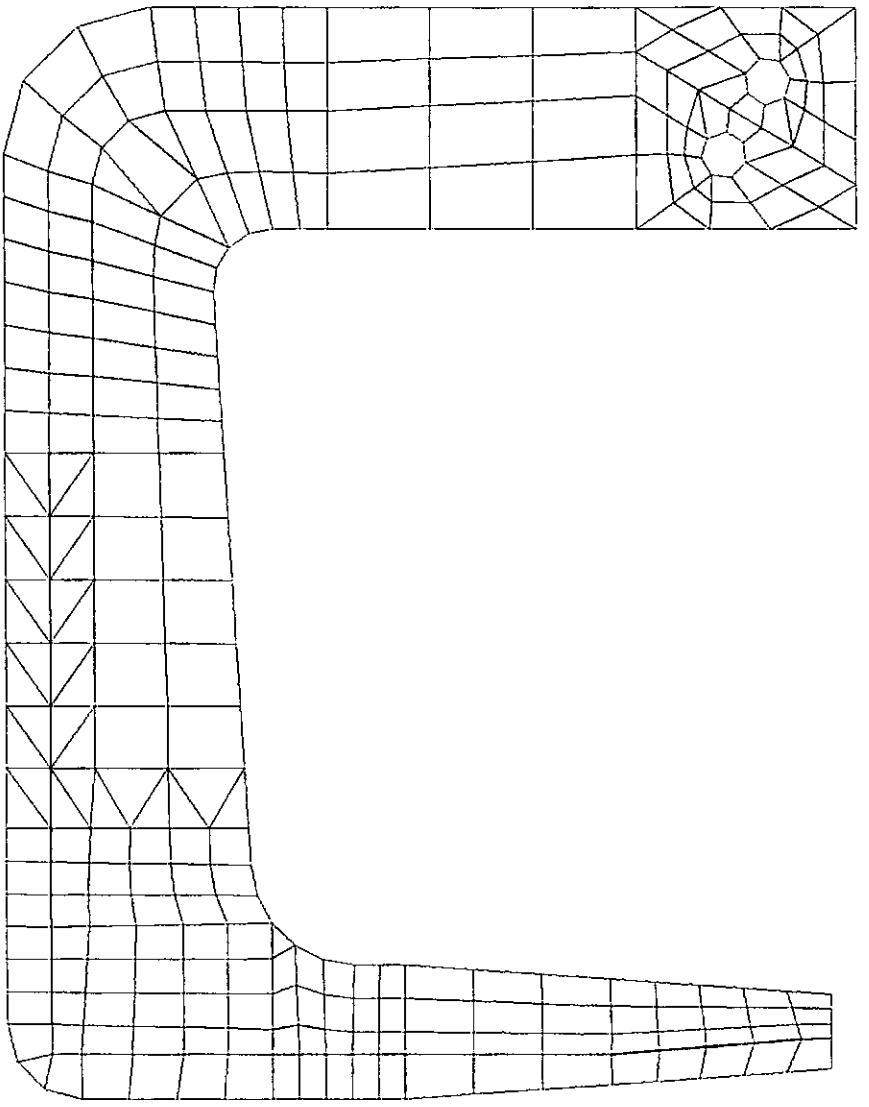


FIG VII.159.

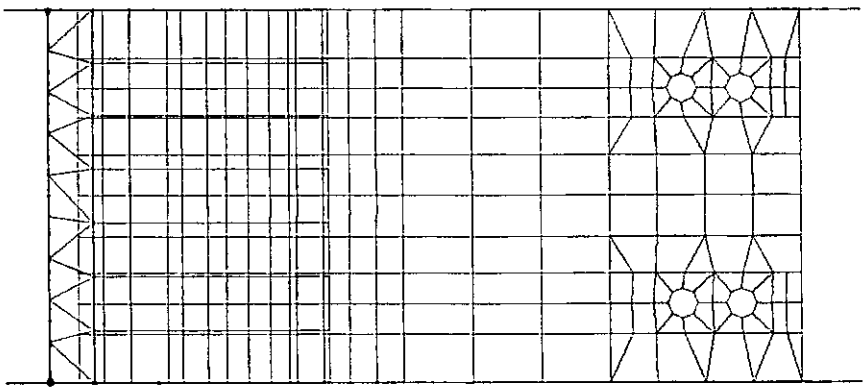


FIG VII.160.

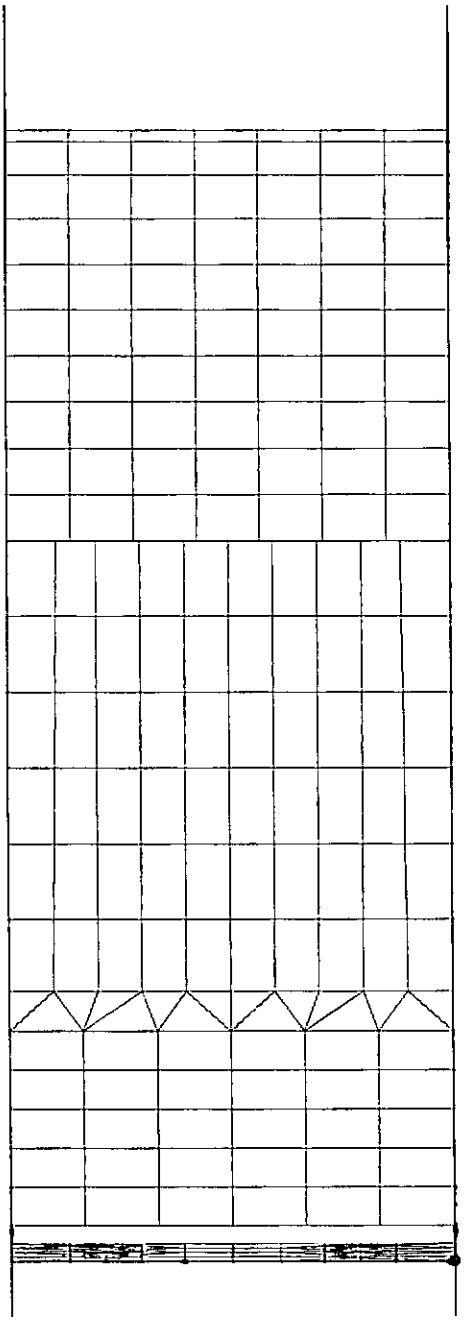


FIG VII.161.

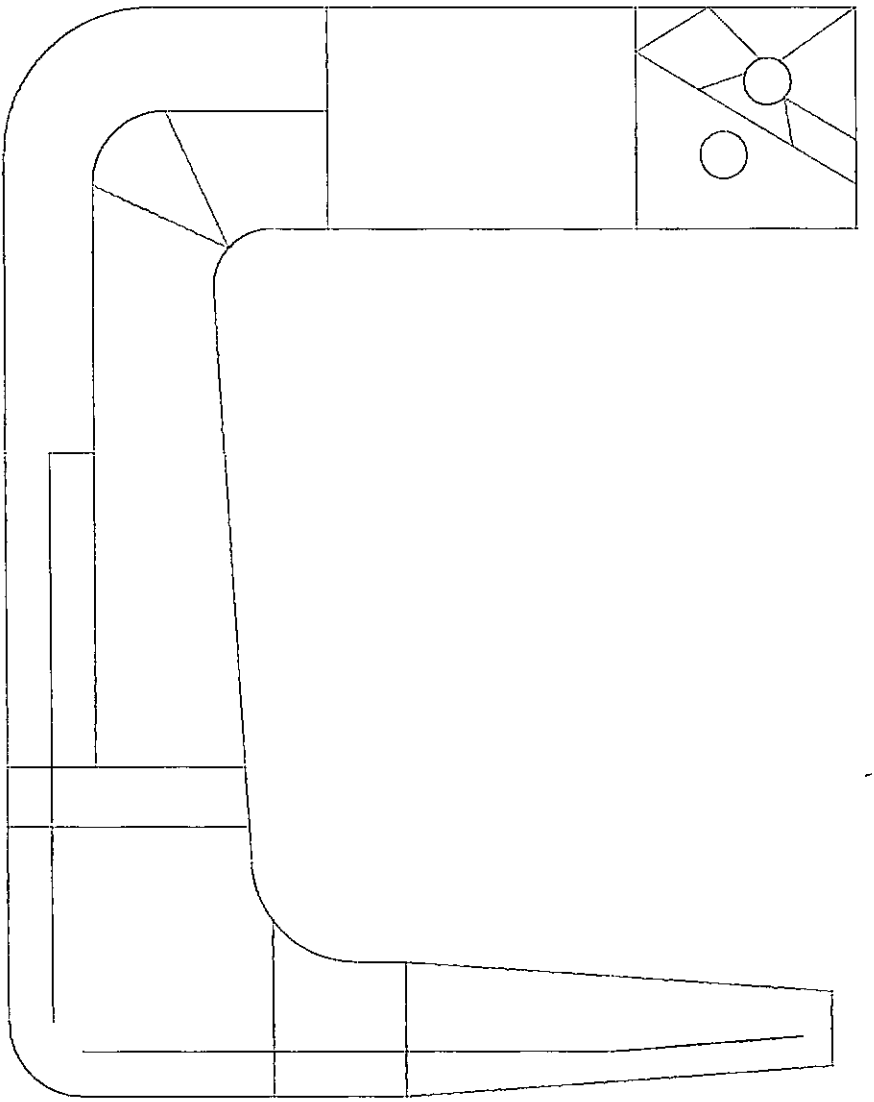


FIG VII.162.

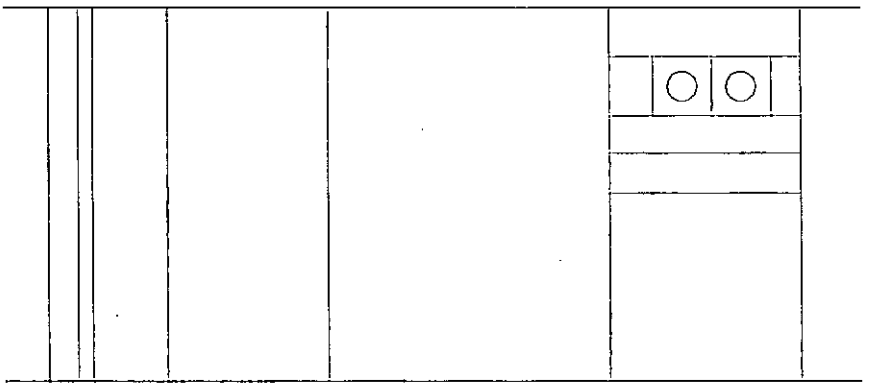


FIG. VII.163.

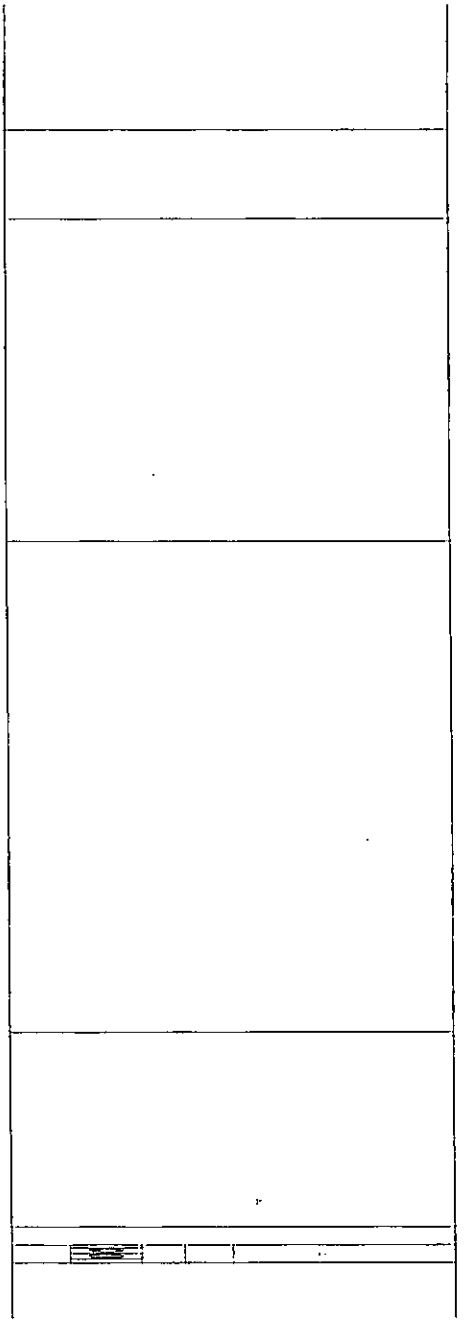


FIG. VII.164.

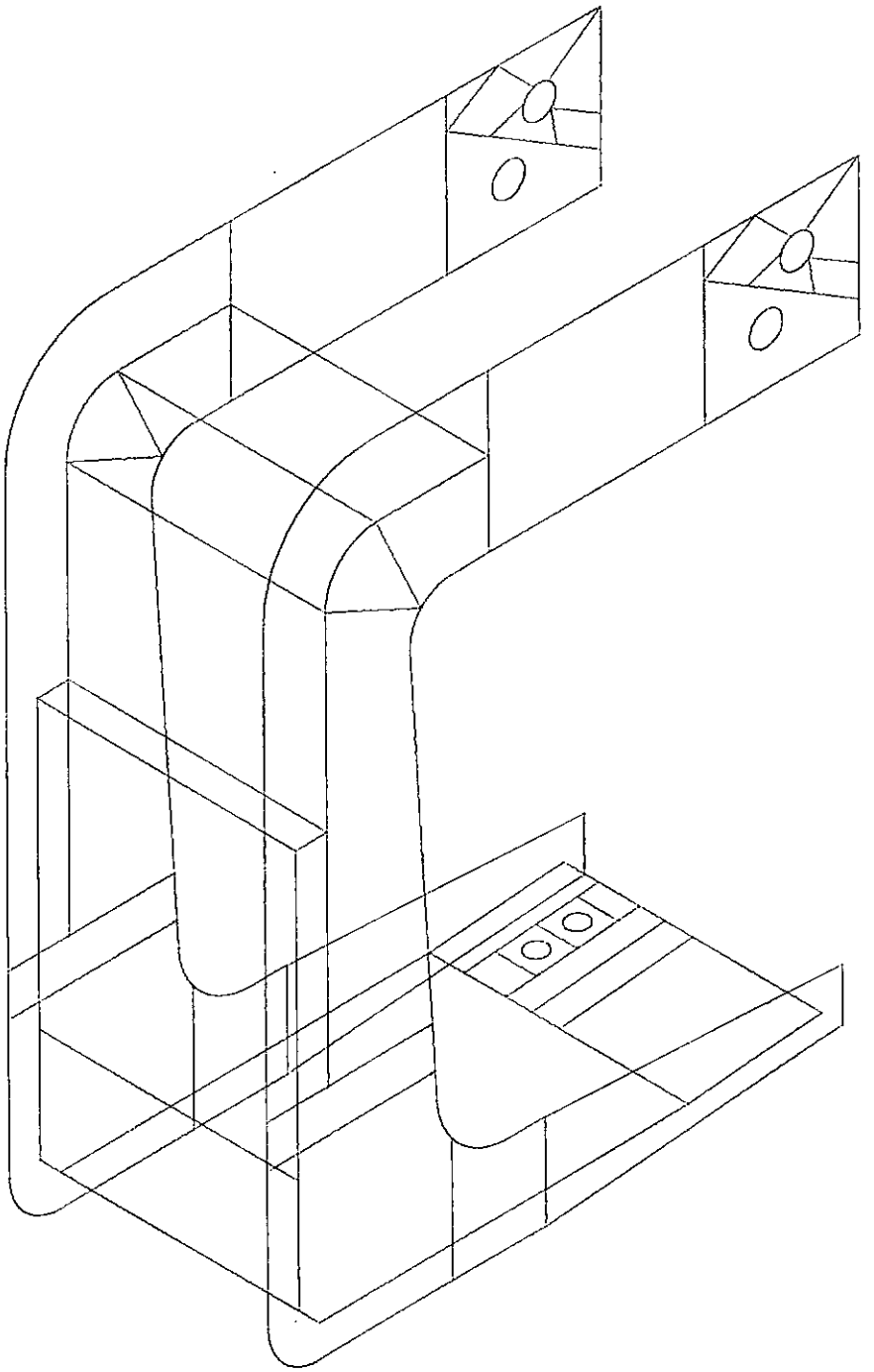


FIG VII.165.

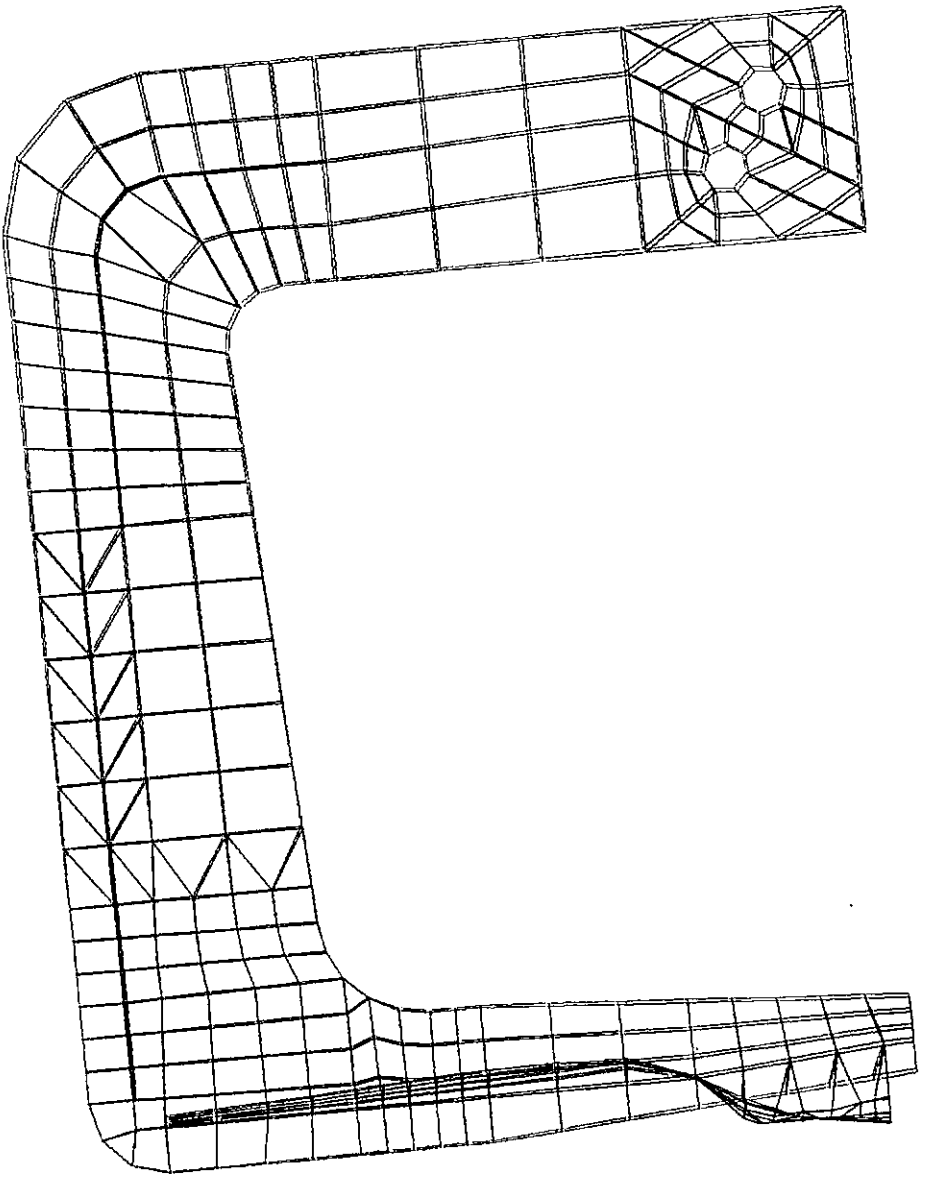


FIG VII.166.

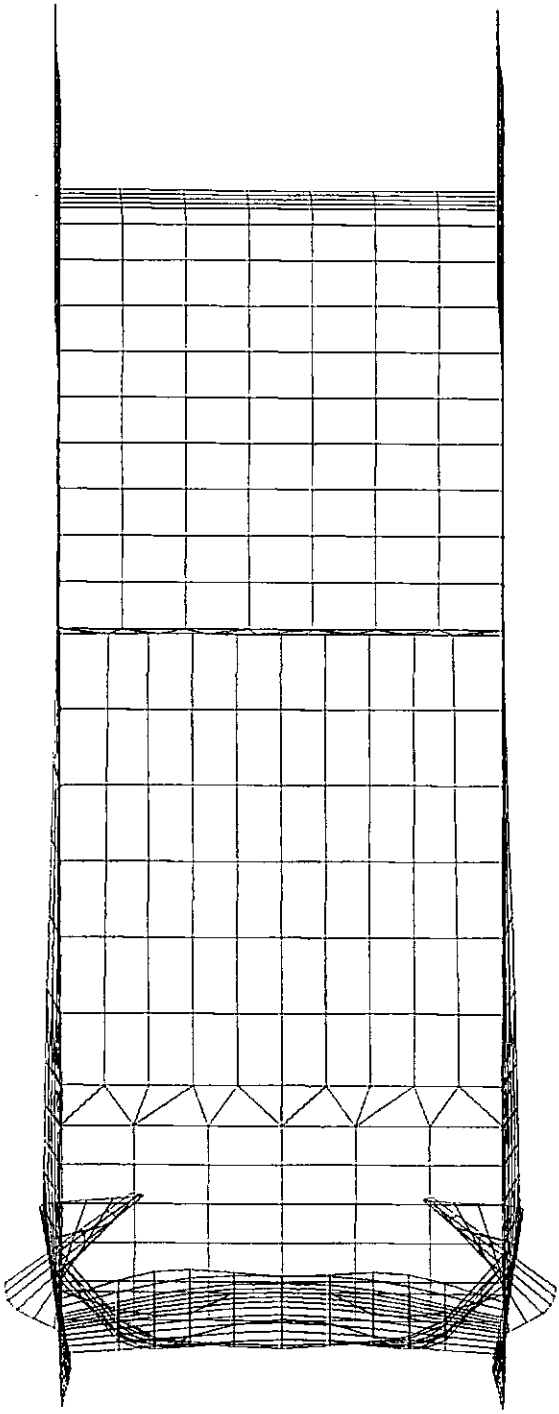


FIG VII.167.

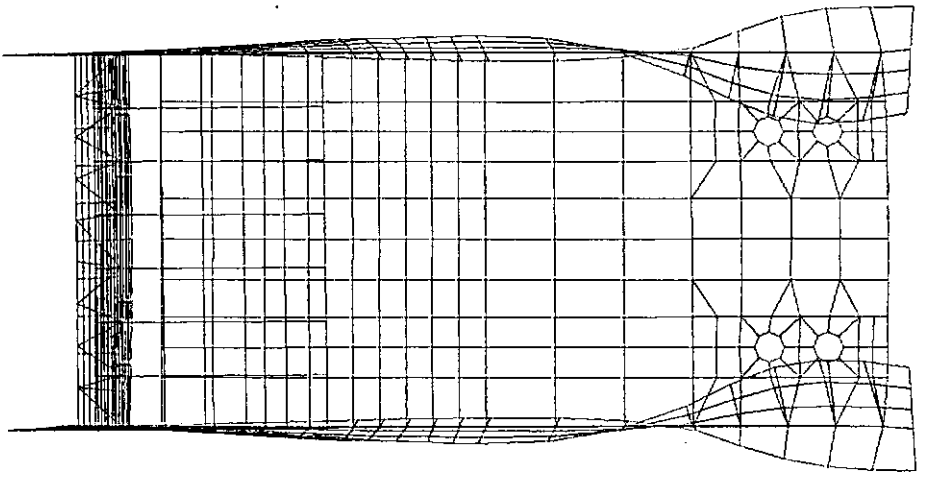


FIG VII.168.

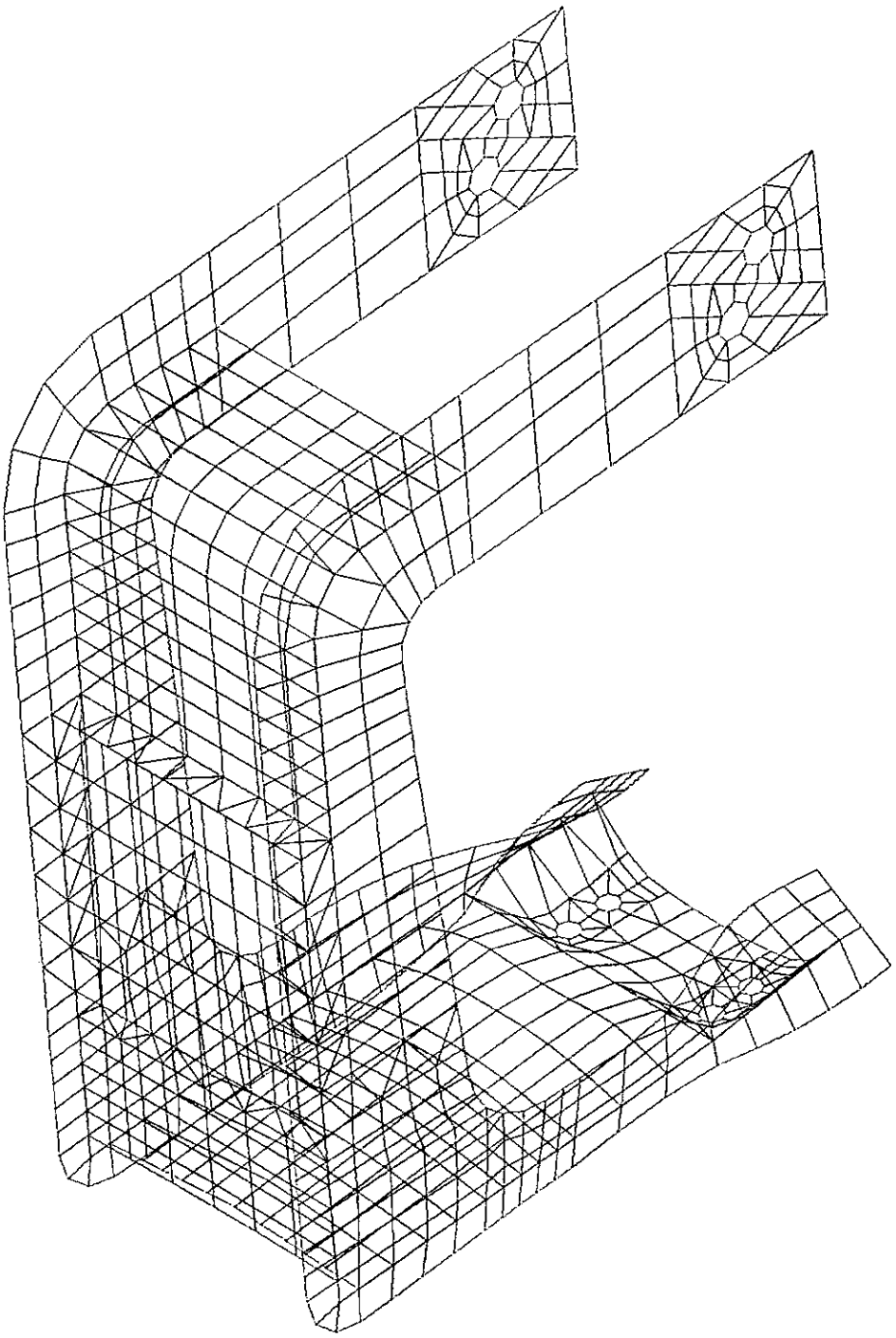


FIG VII.169.

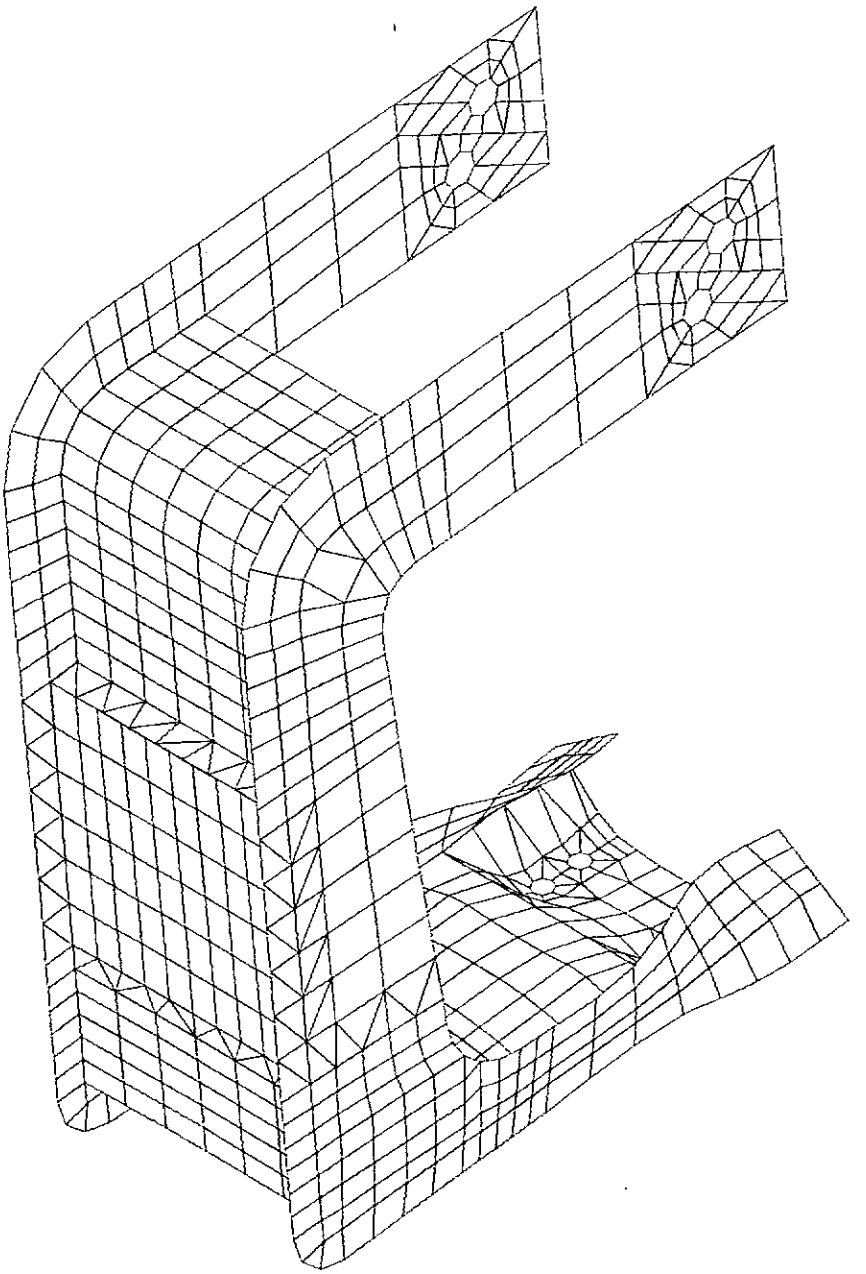


FIG VII.170.

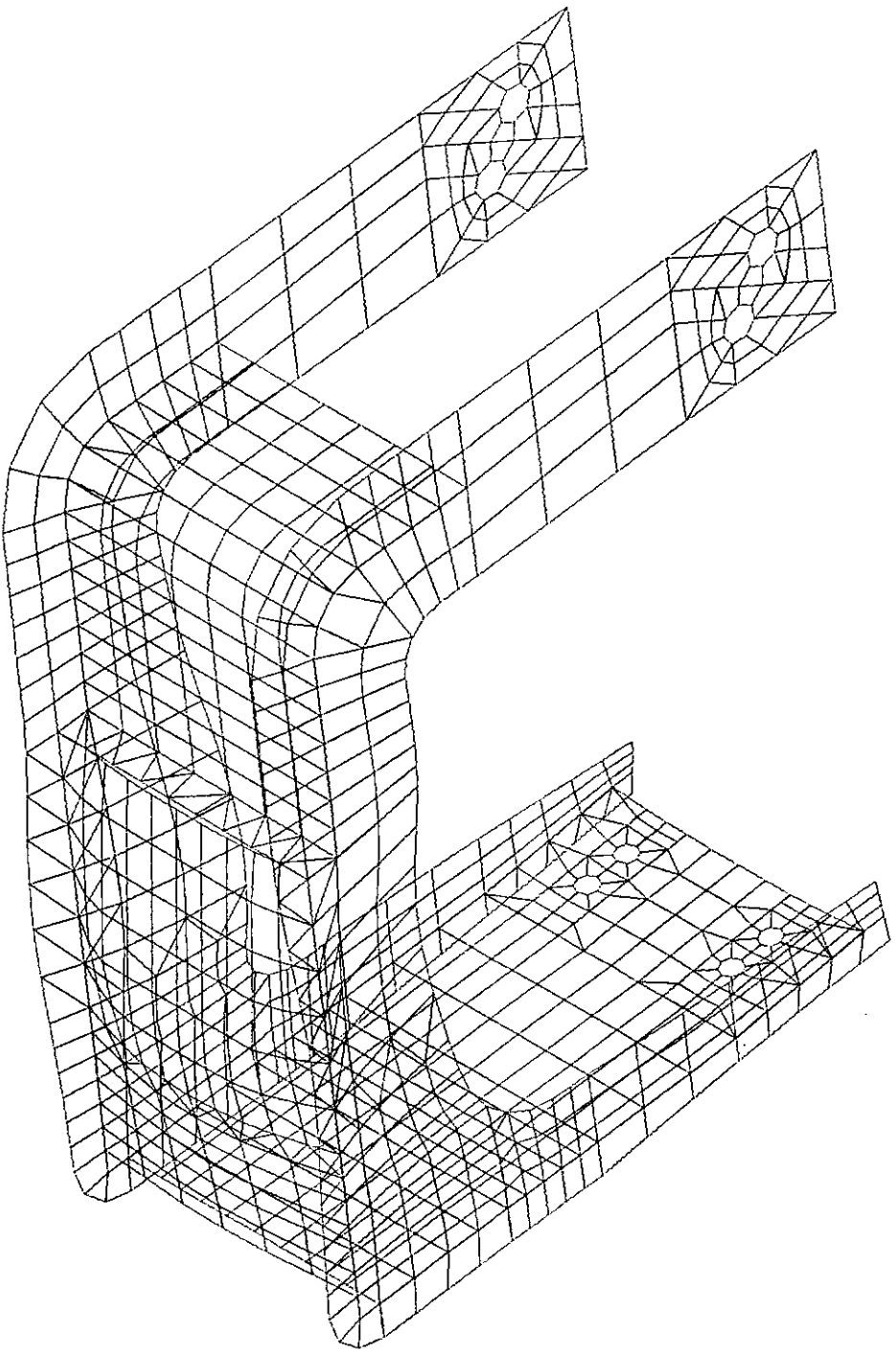


FIG VII.171.

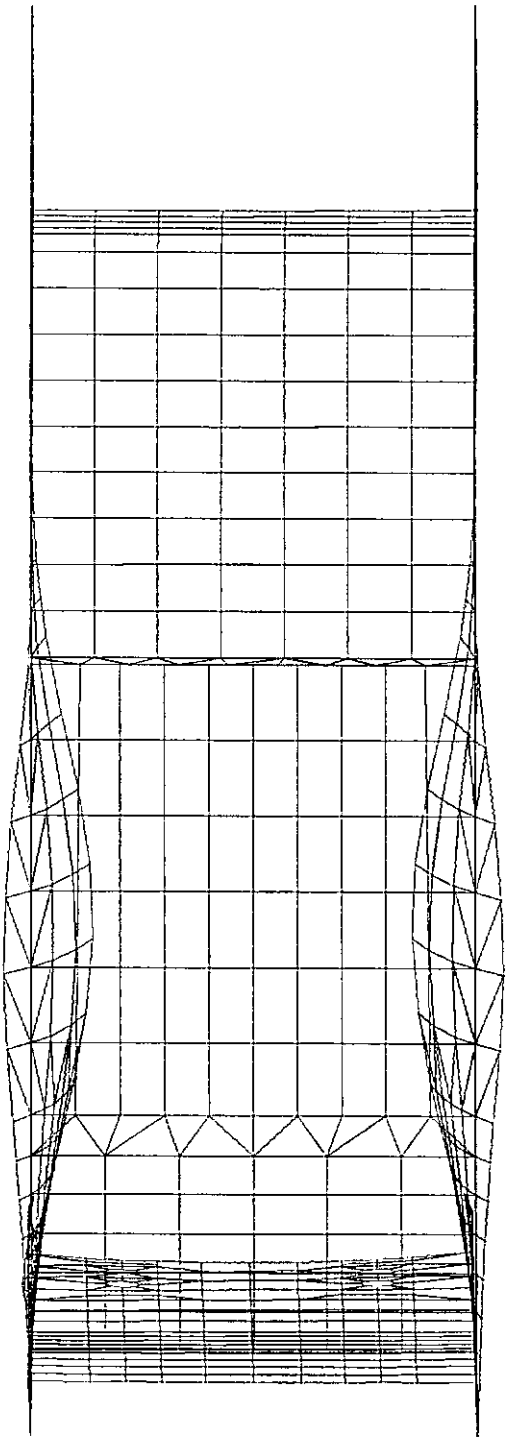


FIG VII.172.

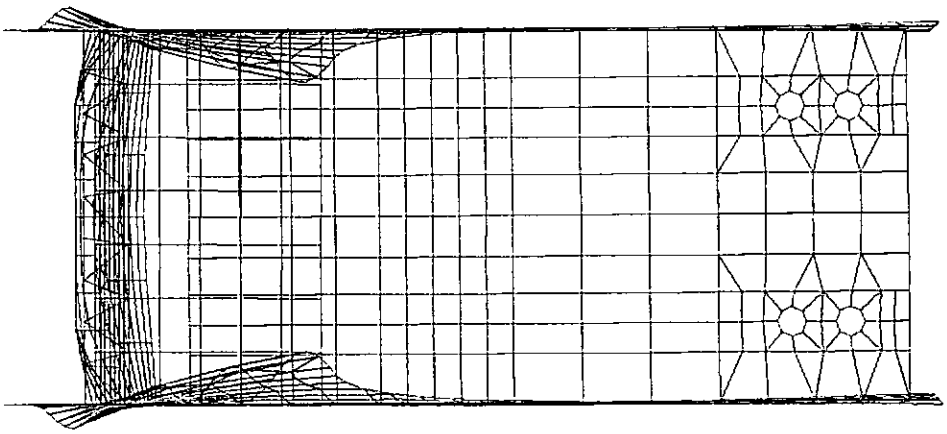


FIG VII.173.

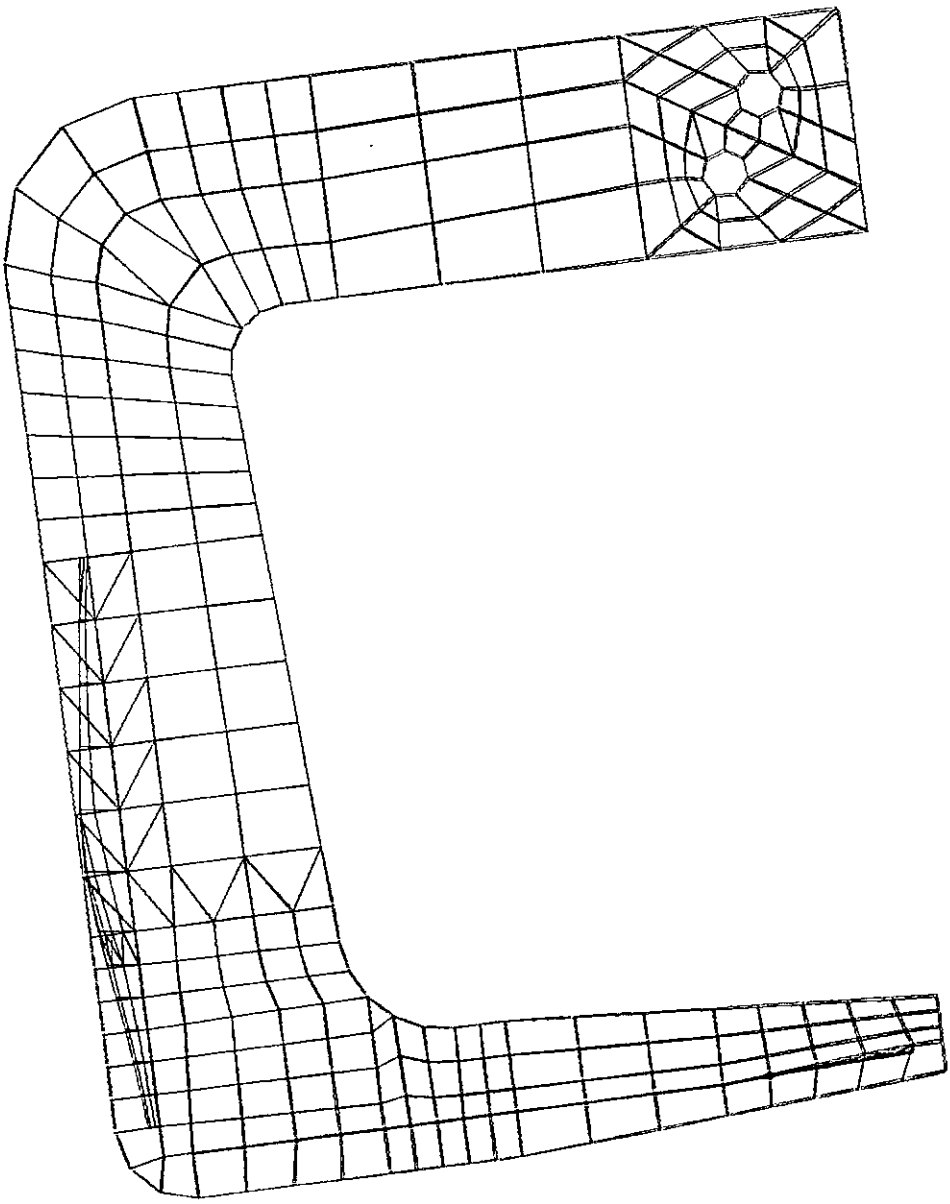


FIG VII.174.

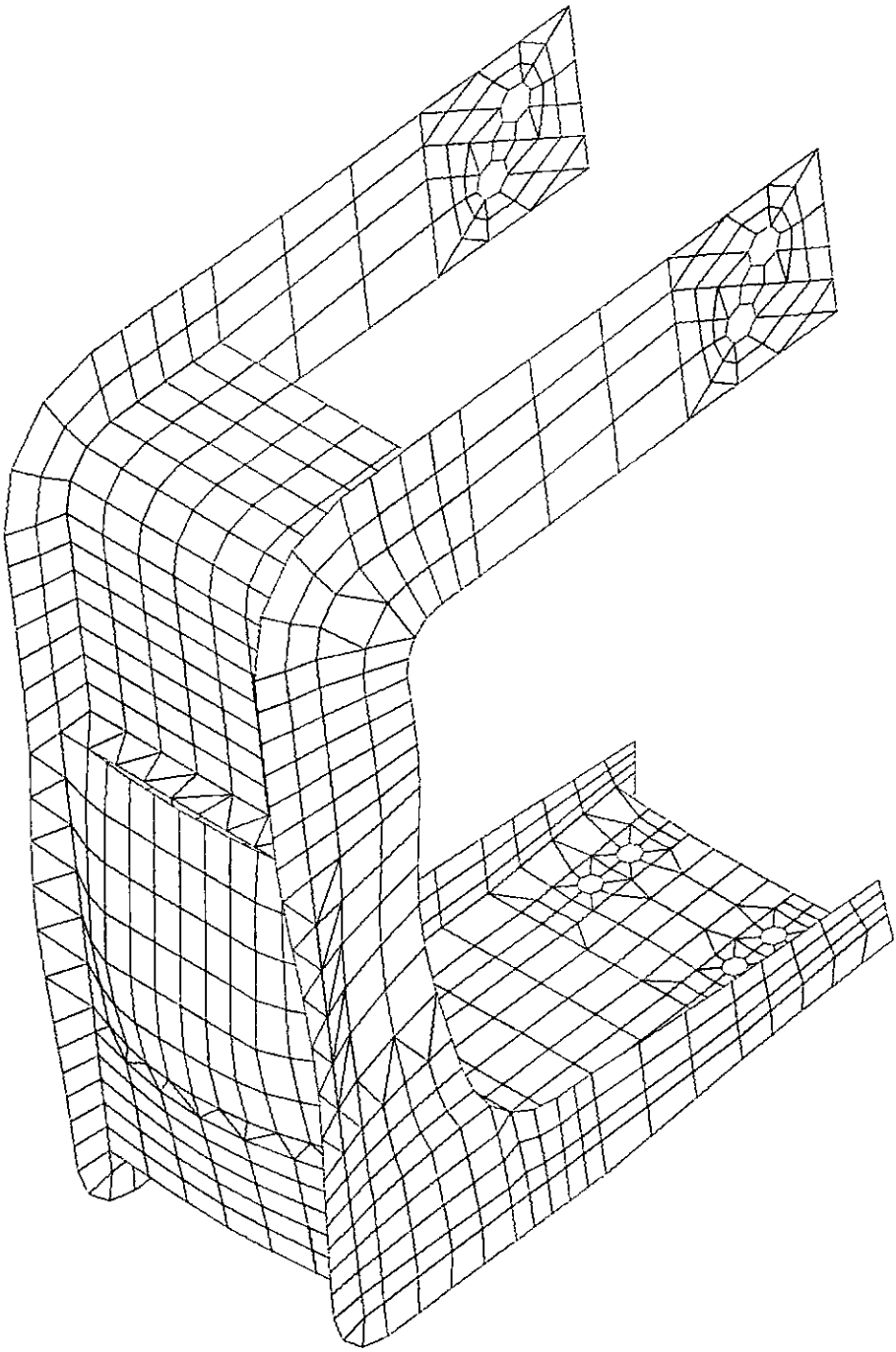


FIG VII.175.

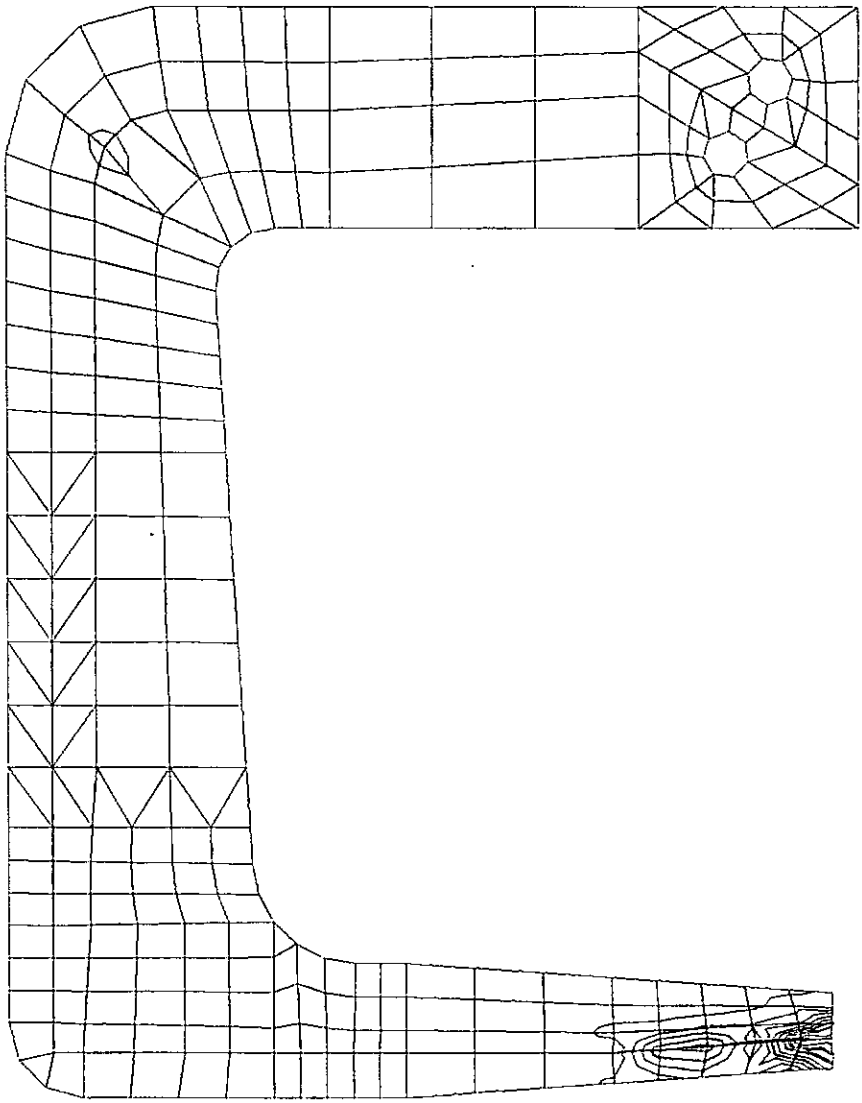


FIG VII.176.

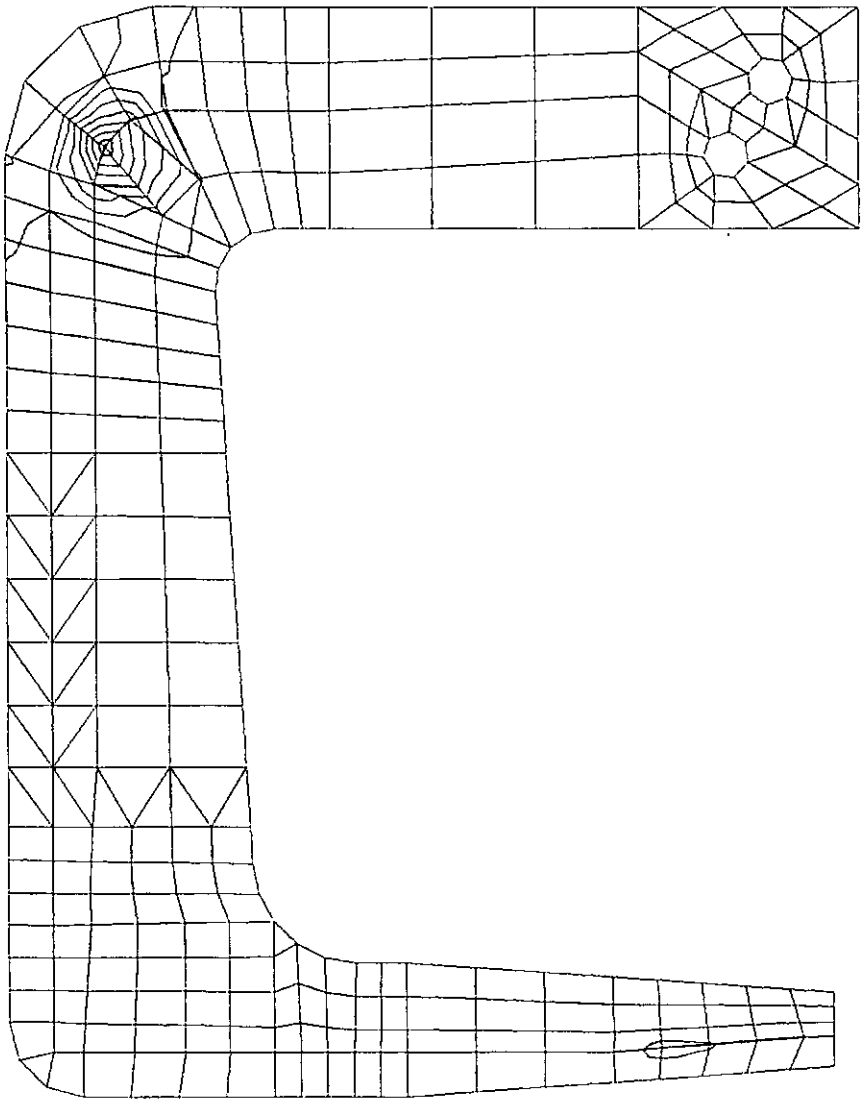


FIG VII.177.

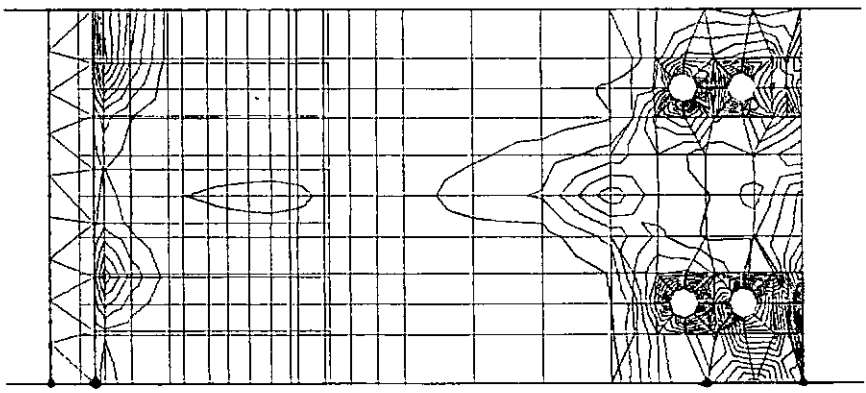


FIG VII.178.

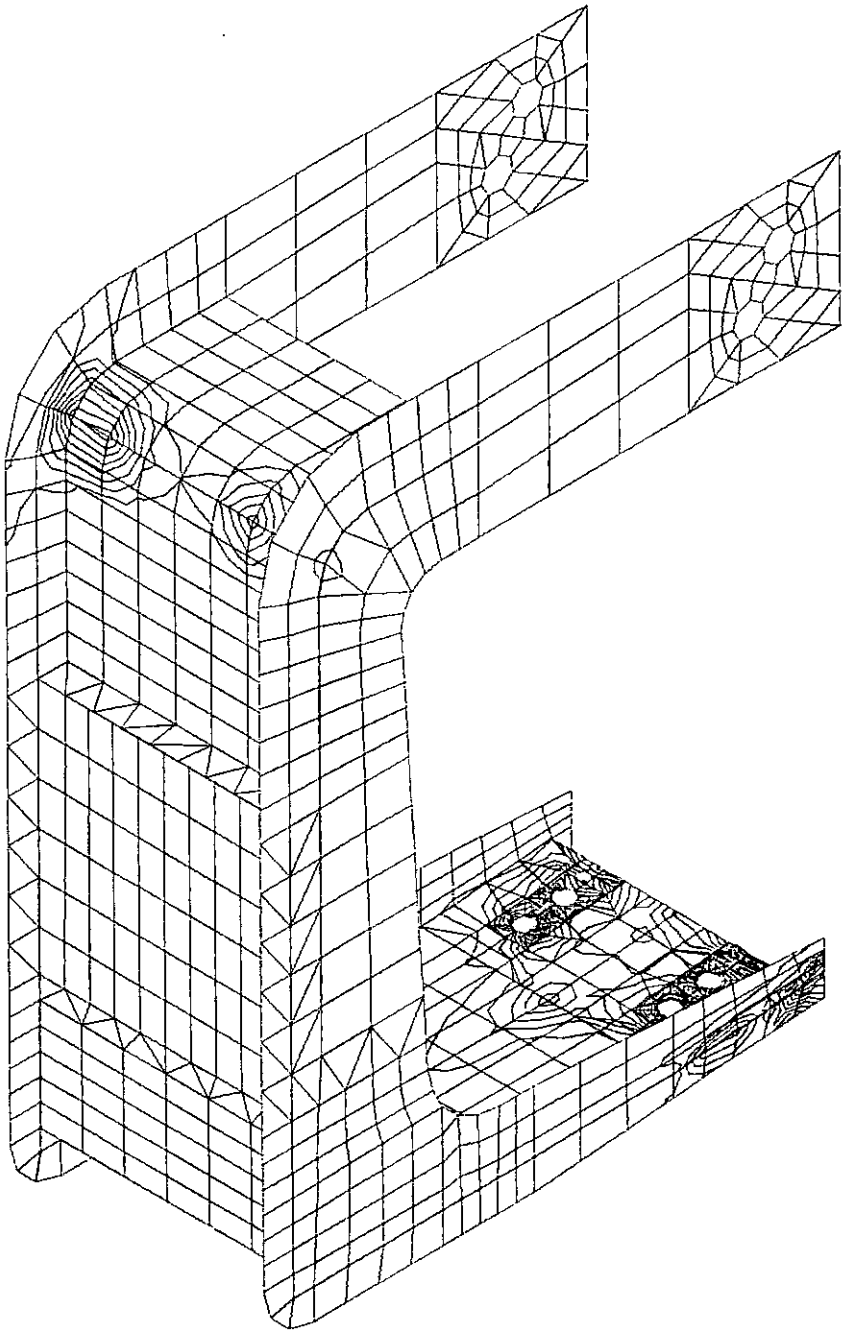


FIG VII.179.

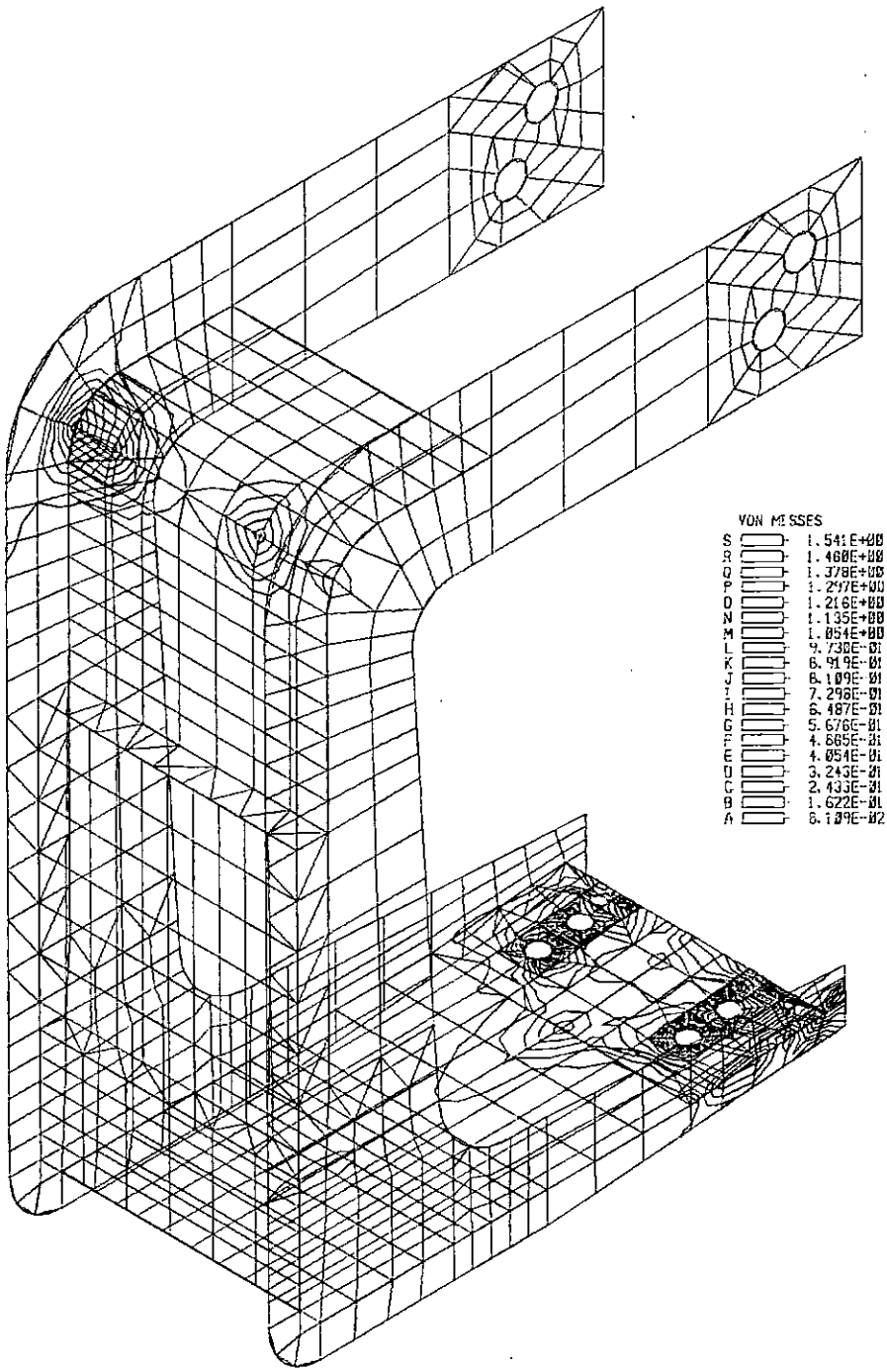

VON MISSES
S 1.541E+00
R 1.460E+00
Q 1.378E+00
P 1.297E+00
O 1.216E+00
N 1.135E+00
M 1.054E+00
L 9.730E-01
K 8.919E-01
J 8.109E-01
I 7.298E-01
H 6.487E-01
G 5.676E-01
F 4.865E-01
E 4.054E-01
D 3.243E-01
C 2.433E-01
B 1.622E-01
A 8.109E-02

FIG VII.180.

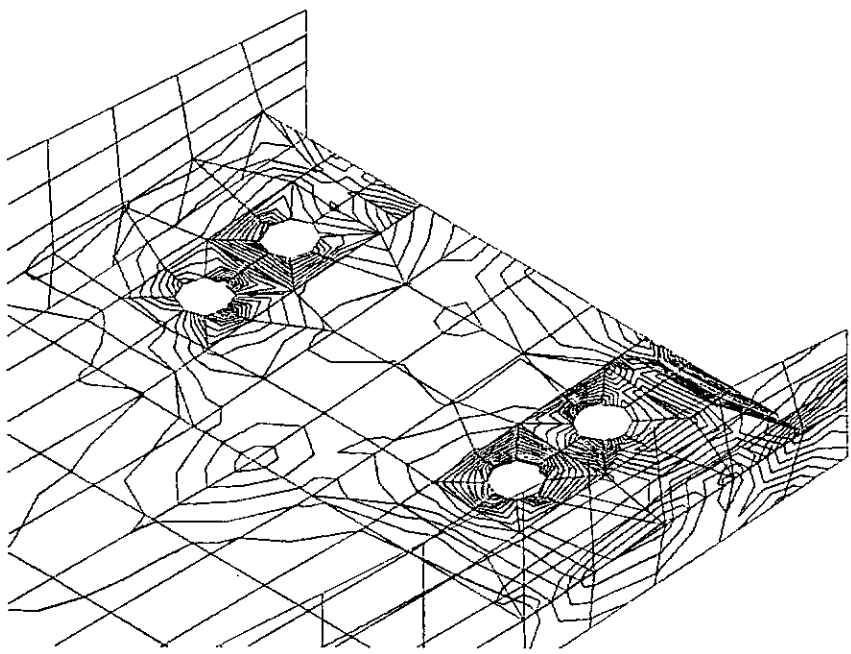

VON MISSES
S 1.541E+00
R 1.460E+00
Q 1.378E+00
P 1.297E+00
O 1.216E+00
N 1.135E+00
M 1.054E+00
L 9.730E-01
K 8.919E-01
J 8.109E-01
I 7.298E-01
H 6.487E-01
G 5.676E-01
F 4.865E-01
E 4.054E-01
D 3.243E-01
C 2.433E-01
B 1.622E-01
A 8.109E-02

FIG VII.181.

CAPÍTULO VIII

VIII.1. CONCLUSÕES

A conclusão imediata e definitiva é de que todos os objetivos foram alcançados e um sistema de pré e pós–processamento para modelos de elementos finitos para computadores pessoais e de aplicações gerais é viável e pode concorrer ou até vir a substituir os sistemas de grande porte.

O ambiente apresentado além de computacionalmente eficiente é de custo relativamente e baixo apresenta–se não só como uma solução para o pequeno usuário ou empresa, mas também uma alternativa inclusive para a grande empresa.

O trabalho também apresenta de uma forma detalhada e descritiva, todos os recursos desejados e indispensáveis em sistemas de pré e pós–processamento em modelos de elementos finitos e CAD, servindo como molde para o desenvolvimento ou avaliação de outros softwares de CAD.

É incontestável o ganho de tempo e trabalho de qualquer método automatizado sobre os métodos manuais. No presente trabalho não foram feitas menções aos tempos gastos em operações de edição e/ou geração das malhas. Isto deve–se ao fato do envolvimento de uma variedade muito grande de variáveis.

Sem considerar a variedade de equipamentos ou configurações destes, a simples configuração do sistema operacional ou do **AutoCAD** pode produzir diferenças superiores a 50% no tempo de uma simples regeneração na tela de modelos grandes.

O tempo total gasto no pré–processamento de modelos de elementos finitos está diretamente ligado ao número de operações de edição executadas. A velocidade de resposta para operações de edição em geral, varia de equipamento e depende da configuração do sistema e tamanho do modelo.

A diversidade de fatores impede que se determine qual a configuração ideal, isto é, um estudo deverá ser feito para cada caso em particular. Uma configuração básica é fornecida e não é necessario qualquer alteração para a utilização, porém o usuário com conhecimento no equipamento, sistema operacional e ambiente **AutoMESH/AutoCAD** poderá tirar mais vantagens dos seus recursos e obter taxas de eficiência computacional surpriendentes.

O sistema do modo como foi apresentado não constitui um trabalho estanque, isto é, a customização e adaptabilidade intrínseca do ambiente **AutoCAD** continua a existir. Os menus ou comandos do sistema **AutoMESH** são passíveis de alteração de modo a adaptarem–se às necessidades mais particulares. Foi vista a definição das propriedades associadas aos elementos finitos apenas para análise estrutural, sem considerar constantes térmicas ou eletromagnéticas por exemplo.

Uma extensão ao sistema **AutoMESH** sugerida e inclusive avaliada, é a inclusão de comandos para a geração automática de jaquetas de plataformas offshore.

APÊNDICE I

AI.1. FUNÇÕES DE FORMA

Neste apêndice serão apresentadas as funções de forma usadas para mapeamento das coordenadas naturais em coordenadas cartesianas.

Estas funções, também conhecidas como funções de interpolação, são as mesmas usadas da formulação dos elementos finitos isoparamétricos, daí o nome de método de mapeamento isoparamétrico.

O procedimento básico na formulação de elementos finitos isoparamétricos é expressar as coordenadas e deslocamentos em termos de interpolações, usando o sistema de coordenadas naturais do elemento. Este sistema de coordenadas é uni, bi, ou tri–dimensionais, dependendo da dimensionalidade do elemento.

A apresentação a seguir é dada para um elemento tridimensional. Contudo, elementos uni e bidimensionais são incluídos, simplesmente usando apenas os eixos coordenados relevantes e funções de interpolação apropriadas.

Considerando um elemento tridimensional genérico, as coordenadas interpoladas são:

$$\begin{cases} X = \sum_{i=1}^{q} h_i \, x_i & \text{(AI.1.a)} \\ Y = \sum_{i=1}^{q} h_i \, y_i & \text{(AI.1.b)} \\ Z = \sum_{i=1}^{q} h_i \, z_i & \text{(AI.1.c)} \end{cases}$$

onde x, y e z são as coordenadas em algum ponto do elemento e x_i, y_i, z_i, $i = 1...q$ são as coordenadas dos q nós do elemento. As funções de interpolação h_i são definidas em sistemas de coordenadas naturais do elemento, os quais têm variáveis r, s e, t que variam entre -1 e $+1$. Para elementos uni e bidimensionais, apenas as equações relevantes (das anteriores) devem ser aplicadas e as funções de interpolação deverão depender apenas das coordenadas naturais r, e r, s respectivamente.

A propriedade fundamental da função de interpolação h_i é que seu valor no sistema de coordenadas naturais é unitário no nó i e zero em todos os outros. Usando estas condições, as funções h_i correspondentes a um arranjo específico (distribuição) dos nós podem ser obtidas de uma maneira sistemática. Contudo, é conveniente construí–las por inspeção, como apresentado a seguir.

Partiremos da construção das funções de interpolação para um elemento de barra de três nós, generalizando para bi e tridimensionais finalmente.

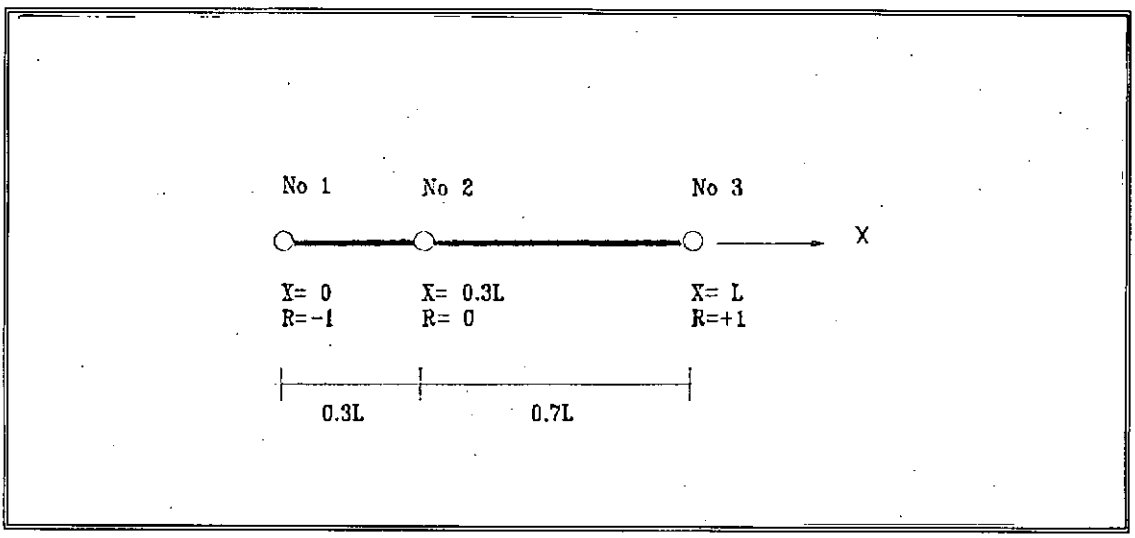


Fig AI.1

A primeira coisa a observar é que para um elemento de barra de três nós, são necessários polinômios que involvem $\mathbb{R}^2$ como a maior potência de $\mathbb{R}$ em outras palavras, as funções de interpolação deverão ser parábolas. A função h_2 pode, assim, ser construída sem muito esforço. Nominalmente, a parábola que satisfaz a condição de ser igual a *zero* em $\mathbb{R}=+1$ e igual a *1* em $\mathbb{R}=0$ é dada por $(1-\mathbb{R}^2)$.

As outras duas funções h_1 e h_3 são construídos por superposição de uma função linear e uma parábola. Considere a função de interpolação *h3*. Usando $\frac{1}{2}(1+\mathbb{R})$, as condições em que a função deverá ser *zero* em $\mathbb{R}=-1$ e *1* em $\mathbb{R}=+1$ são satisfeitas. A função de interpolação h_1 é obtida de modo semelhante.

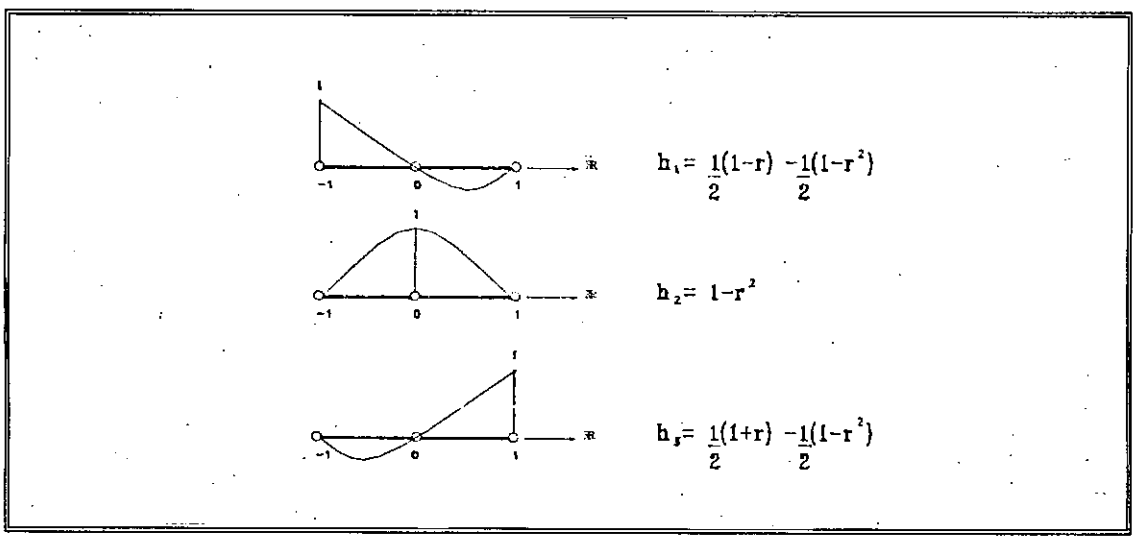


Fig AI.2

A adição de outro nó à barra resulta em uma função de interpolação adicional e uma correção a ser aplicada às funções de interpolação existentes.

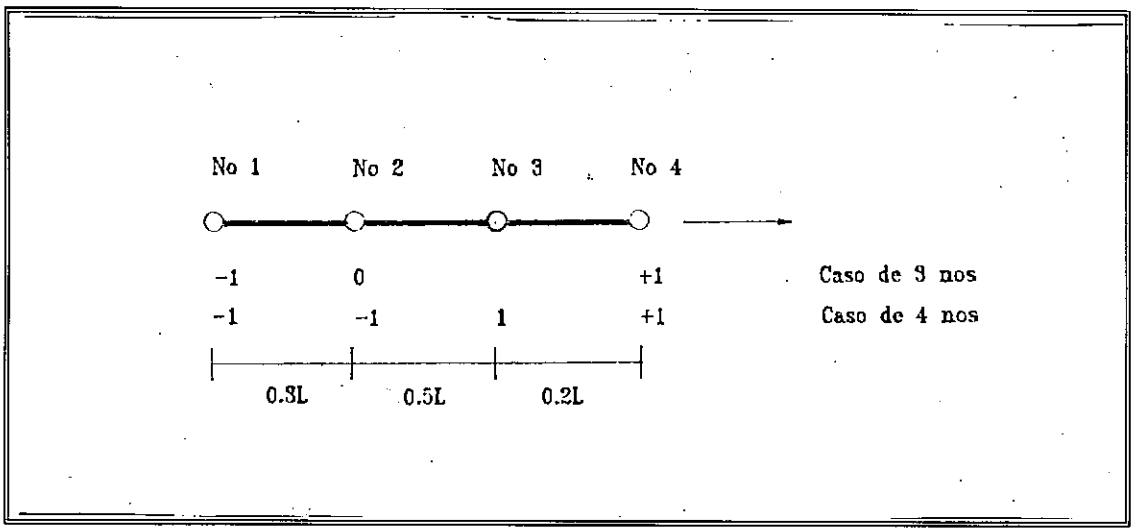


Fig AI.3

A tabela abaixo mostra as funções de interpolação de um elemento unidimensional considerado anteriormente com um quarto nó adicional possível. Como mostrado, o elemento pode ter de dois a quatro nós.

		incluído apenas se nó i esta presente	
		i=3	i=4
h_1	$(1-r)/2$	$-(1-r^2)/2$	$(-9r^3+r^2+9r-1)/16$
h_2	$(1+r)/2$	$-(1-r^2)/2$	$(9r^3+ r^2-9r-1)/16$
h_3	$(1-r^2)$		$(27r^3+7r^2-27r-7)/16$
h_4	$(-27r^3-9r^2+27r+9)/16$		

Tab AI.1

Este procedimento de construção das funções de interpolação de elementos unidimensionais pode ser diretamente generalizado para uso em duas e três dimensões.

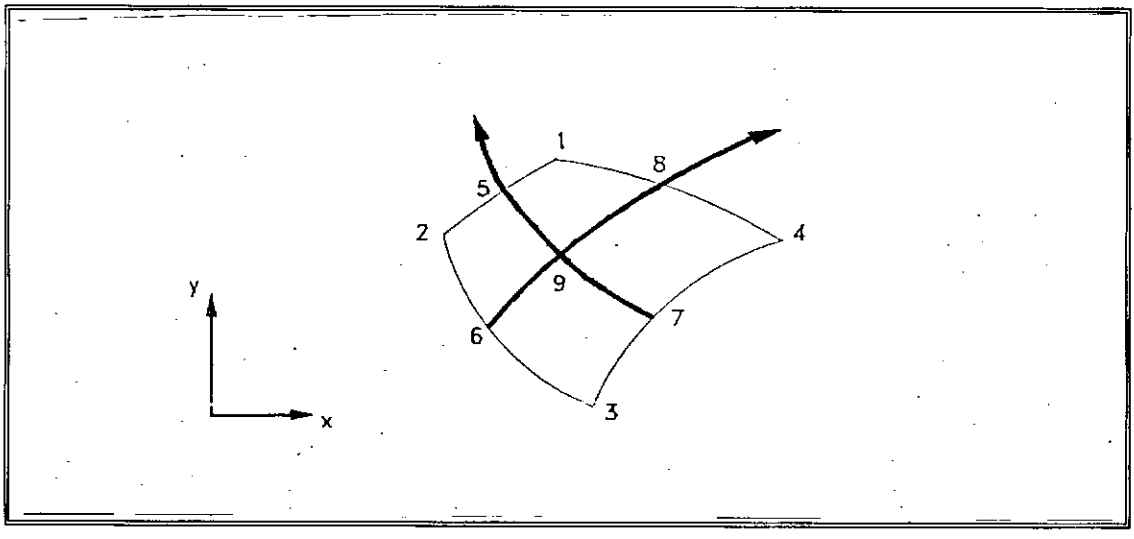


Fig AI.4

		incluídos somente se o nó i é definido				
		i=5	i=6	i=7	i=8	i=9
h_1	$(1+r)(1+s)/4$	$-h_5/2$			$-h_8/2$	$-h_9/4$
h_2	$(1-r)(1+s)/4$	$-h_5/2$	$-h_6/2$			$-h_9/4$
h_3	$(1-r)(1-s)/4$		$-h_6/2$	$-h_7/2$		$-h_9/4$
h_4	$(1+r)(1-s)/4$			$-h_7/2$	$-h_8/2$	$-h_9/4$
h_5	$(1-r^2)(1+s)/2$					$-h_9/4$
h_6	$(1-s^2)(1-r)/2$					$-h_9/4$
h_7	$(1-r^2)(1-s)/2$					$-h_9/4$
h_8	$(1-s^2)(1+r)/2$					$-h_9/4$
h_9	$(1-r^2)(1-s^2)$					

Tab AI.2

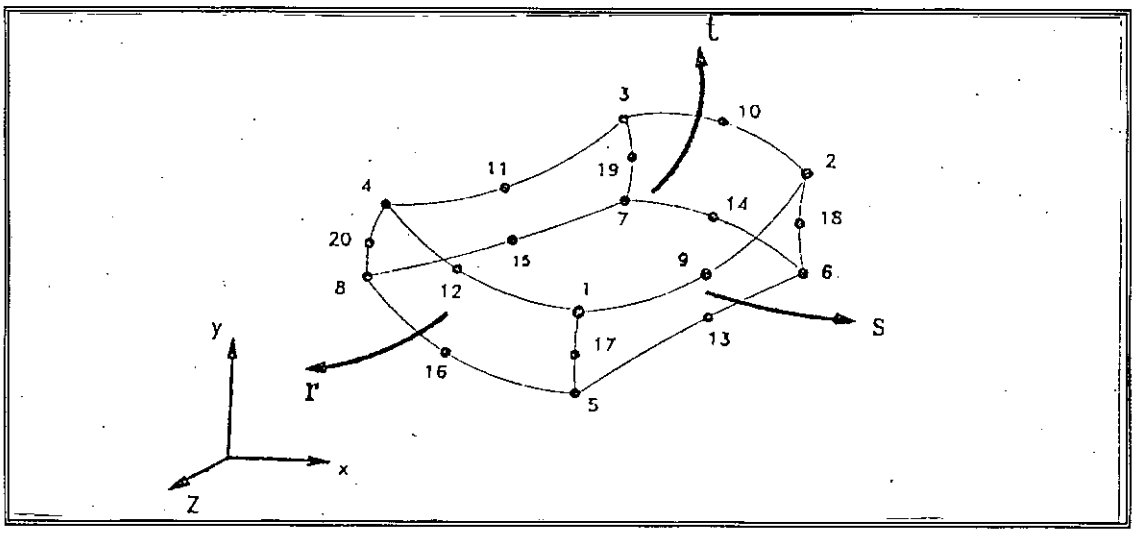


Fig AI.5

	i= 8
h_1	$G_1-(G_9+G_{12}+G_{17})/2$
h_2	$G_2-(G_9+G_{10}+G_{18})/2$
h_3	$G_3-(G_{10}+G_{11}+G_{19})/2$
h_4	$G_4-(G_{11}+G_{12}+G_{20})/2$
h_5	$G_5-(G_{13}+G_{16}+G_{17})/2$
h_6	$G_6-(G_{13}+G_{14}+G_{18})/2$
h_7	$G_7-(G_{14}+G_{15}+G_{19})/2$
h_8	$G_8-(G_{15}+G_{16}+G_{20})/2$
h_j	$G_j \quad \{j= 1..20\}$

Tab AI.3.a

	i= 1...20
G_i	$B(r,r_i)\ B(s,s_i)\ B(t,t_i)$
G_i	0 : nó não incluido

Tab AI.3.b

	$\beta_i = \pm 1$	$\beta_i = 0$
$B(\beta,\beta_i)$	$(1+\beta_i\beta)/2$	$(1-\beta^2)$

Tab AI.3.c

Para elementos triangulares as funções de forma podem ser obtidas pelo colapso de um dos lados de um elemento retangular.

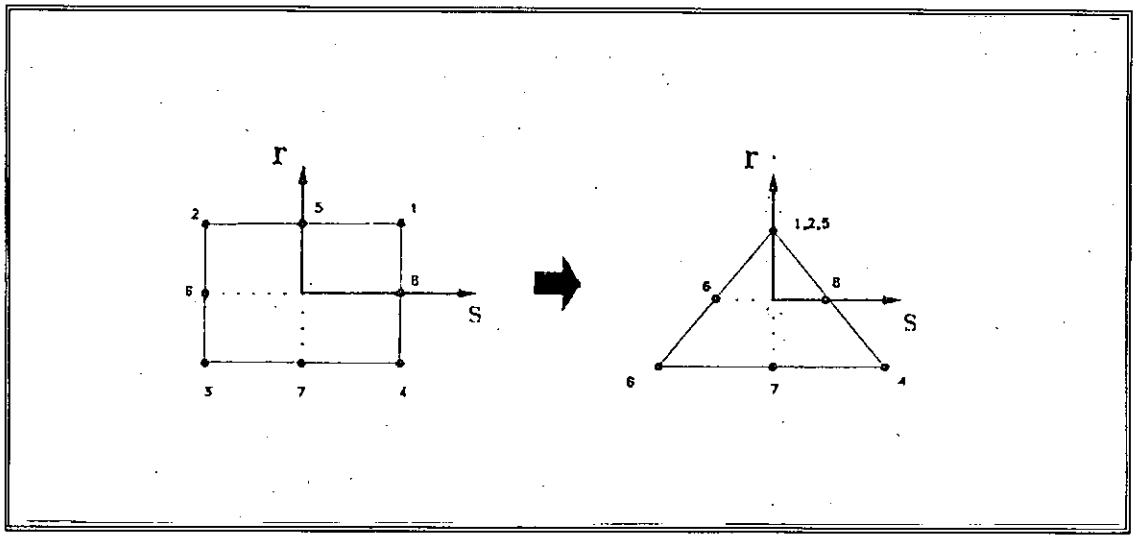


Fig AI.6

A figura anterior apresenta um elemento triangular de seis nós, obtido pelo colapso de um lado, 1–2–5, do elemento retangular. O elemento retangular colapsado não preenche os requisitos de que os deslocamentos internos U e V variem da mesma maneira para o deslocamento nodal de cada vértice e ponto médio à aresta.

De modo a preencher estes requisitos, correções precisam ser aplicadas as funções de interpolação dos nós 3, 4 e 7 para obter as funções de interpolação h_i do elemento triangular.

$$
\begin{cases}
h_1^* = \frac{1}{2}\,(1+s) - \frac{1}{2}\,(1-s^2) & \text{(AI.2.a)} \\
h_3^* = \frac{1}{4}\,(1-r)\,(1-s) - \frac{1}{4}\,(1-s^2)\,(1-r) - \\
\qquad \frac{1}{4}\,(1-r^2)\,(1-s) + \Delta h & \text{(AI.2.b)} \\
h_4^* = \frac{1}{4}\,(1+r)\,(1-s) - \frac{1}{4}\,(1-r^2)\,(1-s) - \\
\qquad \frac{1}{4}\,(1-s^2)\,(1-r) + \Delta h & \text{(AI.2.c)} \\
h_6^* = \frac{1}{2}\,(1-s^2)\,(1-r) & \text{(AI.2.d)} \\
h_7^* = \frac{1}{2}\,(1-r^2)\,(1-s) - 2\,\Delta h & \text{(AI.2.e)} \\
h_8^* = \frac{1}{2}\,(1-s^2)\,(1+r) & \text{(AI.2.f)}
\end{cases}
$$

$$
\Delta h = \frac{1}{8}\,(1-r^2)\,(1-s^2) \qquad \text{(AI.3)}
$$

Estas funções deverão ser adicionadas às funções de interpolação do elemento de oito nós.

Embora o procedimento de distorcer um elemento isoparamétrico para gerar um elemento triangular possa ser efetivo em alguns casos, como mostrado anteriormente, as funções de interpolação dos elementos triangulares podem ser obtidas diretamente usando coordenadas de área.

Considere o triângulo da figura a seguir, em coordenadas cartesianas:

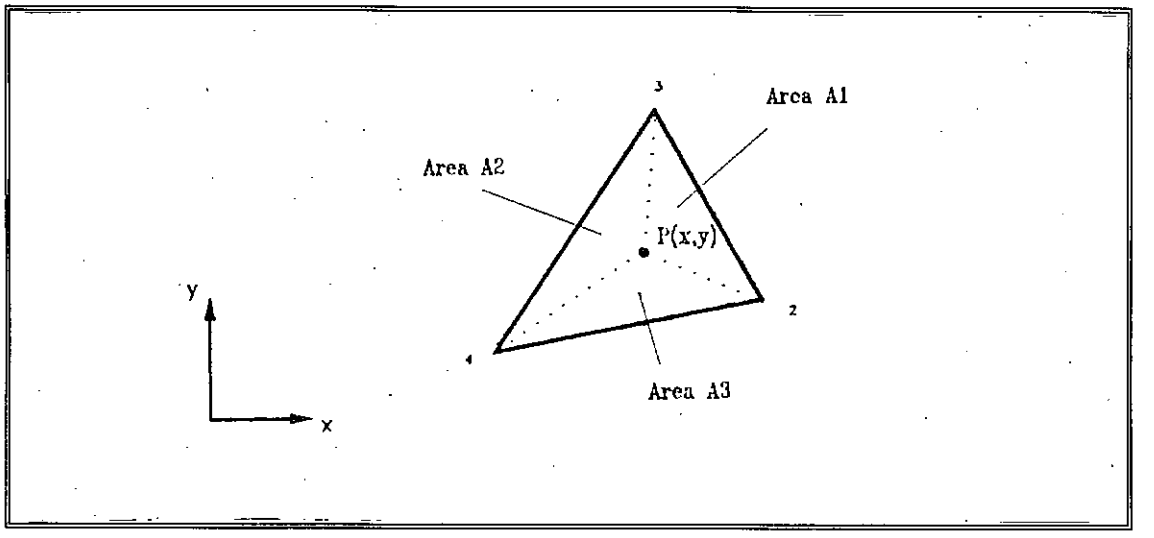


Fig AI.7

A posição de um ponto P típico, interior ao triângulo, de coordenadas X e Y é dado em coordenadas de área por:

$$L_1 = \frac{A_1}{A} \tag{AI.4.a}$$

$$L_2 = \frac{A_2}{A} \tag{AI.4.b}$$

$$L_3 = \frac{A_3}{A} \tag{AI.4.c}$$

onde A é a área total do triângulo.

Assim sendo, temos:

$$L_1 + L_2 + L_3 = 1 \tag{AI.5}$$

As funções de interpolação serão dadas como uma relação entre as coordenadas cartesianas e de área. Devido a variação de X e Y ser linear entre os pontos 1, 2 e 3, temos:

$$X = L_1\, x_1 + L_2\, x_2 + L_3\, x_3 \tag{AI.6.a}$$

$$Y = L_1\, y_1 + L_2\, y_2 + L_3\, y_3 \tag{AI.6.b}$$

onde x_1, x_2, x_3, y_1, y_2 e y_3 são respectivamente as coordenadas X e Y dos vértices 1, 2 e 3 do triângulo.

O sistema de equações obtido pode ser expresso como

$$\begin{bmatrix} 1 \\ X \\ Y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \tag{AI.7}$$

que correponde à

$$X = \sum_{i=1}^{3} h_i \; x_i \tag{AI.8.a}$$

$$Y = \sum_{i=1}^{3} h_i \; x_i \tag{AI.8.b}$$

Como esperado, L_i corresponde as funções de interpolação que relacionam as coordenadas cartesianas e coordenadas de área.

Aplicando o mesmo procedimento à figura abaixo obtemos as funções de forma que relacionam as coordenadas naturais e as coordenadas de área de um ponto,

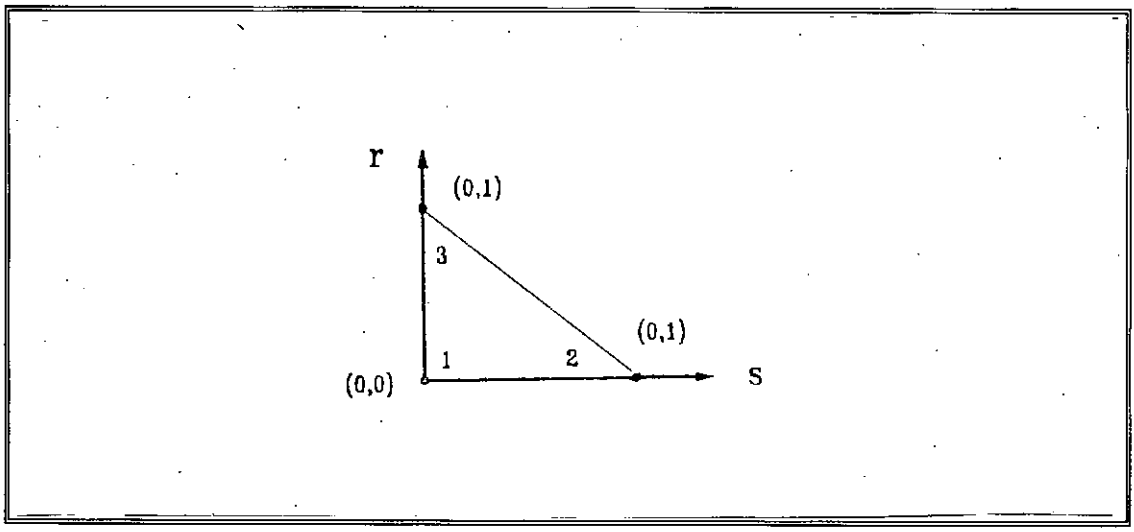


Fig AI.8

As funções de interpolação para este caso são então:

$$h_1 = 1 - r - s \qquad \text{(AI.9.a)}$$

$$h_2 = r \qquad \text{(AI.9.b)}$$

$$h_3 = s \qquad \text{(AI.9.c)}$$

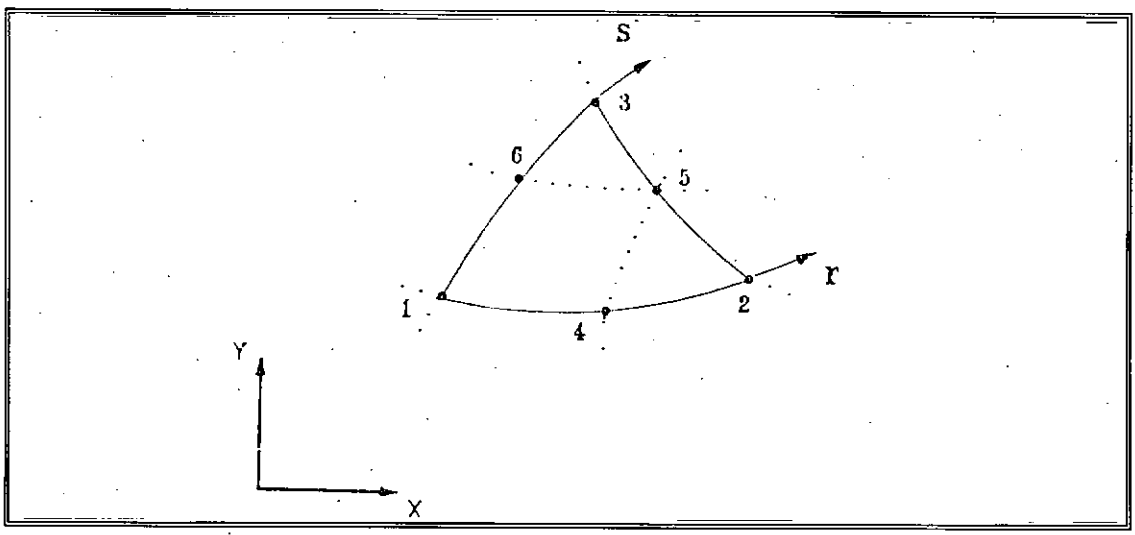


Fig AI.9

		incluídos somente se nó i é definido		
		i=4	i=5	i=6
h_1	1−r−s	$-h_4/2$		$-h_6/2$
h_2	r	$-h_4/2$	$-h_5/2$	
h_3	s		$-h_5/2$	$-h_6/2$
h_4	4r(1−r−s)			
h_5	4rs			
h_6	4s(1−r−s)			

Tab AI.4

APÊNDICE II:

AII.1. O AUTOLISP

Lisp provém de *List Processing*, uma simples porém altamente poderosa linguagem para manipulação de dados simbólicos (palavras em oposição a números). Por esta versatilidade e potencialidade da linguagem, o *lisp* é a linguagem escolhida pela maioria dos pesquisadores de Inteligência Artificial.

AutoLISP é um subconjunto do COMMON **LISP** com muitas funções de manipulação gráfica.

Existe uma idéia geral de que *lisp* é uma linguagem muito difícil de aprender, no entanto, o oposto é frequentemente verdadeiro. *Logo*, a linguagem de programação desenvolvida para ensinar crianças a programar, é um subconjunto do *lisp*.

AII.1.1. CONCEITOS BÁSICOS

Lisp é simples porque é uma linguagem que consiste de apenas três elementos: o átomo, a lista e a função.

– **Átomos**

A unidade básica de qualquer programa *lisp* é chamado de átomo. Um átomo *lisp* é simplesmente a menor unidade possível em um programa *lisp.*

Um átomo pode ser uma palavra, símbolo ou um número. O átomo pode ter um significado (neste caso pode ser chamado nome de variável ou de função) ou pode não ter nenhum significado atribuído.

– **Listas:**

Uma lista é simplesmente um grupo de átomos, ou mesmo um único átomo, cercado por parênteses. Listas precisam ser construídas de um modo particular para fazer sentido ao interpretador *lisp.*

Uma lista pode ser construída a partir de outras listas ou combinações de listas e átomos.

Listas podem ser aninhadas em diversos níveis.

Programas **AutoLISP** executarão muito mais rápido se complexas listas aninhadas forem usadas.

– **Funções:**

Uma função é definida como um programa *lisp*. Uma função pode ser chamada por outros programas *lisp* e executados, ou ser diretamente executado a partir da linha de comandos do **AutoCAD**. Funções são similares a programas, funções ou sub–rotinas em outras linguagens. Uma função sempre retorna um resultado.

AutoLISP é uma linguagem especial, usada para escrever sequências de instruções internas ao **AutoCAD**.

As instruções do **AutoCAD** que tenham sido escritas em **AutoLISP** são chamadas rotinas *LISP*.

As rotinas *LISP* estão contidas em arquivos ASCII que são chamados arquivos *LISP*.

Arquivos *LISP* podem ter qualquer nome válido como um nome de arquivo em DOS, e terão sempre extensão LSP.

Rotinas *LISP* são as mais poderosas ferramentas para otimizar a performance do **AutoCAD**. Elas têm muitas aplicações possíveis, inclusive a criação de novos e únicos comandos para o **AutoCAD**, a inserção de desenhos especiais e funções de cálculo em macros de *menus*, e o desenvolvimento de aplicações que automaticamente introduzem análises de gráficos e desenhos dentro do editor gráfico do **AutoCAD**.

Rotinas *LISP* executam rapidamente cálculos e análises de dados usados para gerar entidades gráficas na tela do monitor. Elas analisam entidades gráficas existentes ou criam novas entidades diretamente pela execução de comandos do **AutoCAD**. As entidades criadas pelas rotinas *LISP* são as mesmas criadas pelo usuário. Com o **AutoLISP** porém, o processo é mais rápido e fácil.

AutoLISP não é uma verdadeira linguagem de programação, tal como *pascal*, *fortran*, *cobol*, ou *C*, porque muitas das funções em **AutoLISP** podem ser usadas apenas para criar instruções para o AutoCAD.

O **AutoLISP** tem, contudo, muito em comum com as funções e a estrutura de sintaxe com o COMMON LISP. O COMMON LISP é por si mesmo apenas um dialeto de uma linguagem de programação que tem muitos dialetos diferentes, MacLISP, UCI LISP, InterLISP e Franz LISP são apenas alguns.

Embora o **AutoLISP** seja uma linguagem de programação limitada, usa muitas estruturas de controle convencionais que são comuns a todas as linguagens de programação.

A sintaxe do **AutoLISP** é construída em certos meios convencionais de organização das instruções computacionais em arquivos ASCII. Estas estruturas tornam possível ao **AutoCAD**, através do interpretador **AutoLISP**, ler os arquivos ASCII e executar as instruções contidas neles. Estas estruturas de sintaxe são relativamente pequenas em número, especialmente se comparadas com as ordinárias linguagens faladas.

Elas também tendem a ser mais rígidas, com poucas exceções e pequena tolerância para variações. As estruturas de sintaxe do **AutoLISP** são sumarizadas a seguir.

AII.1.2. FUNÇÕES

As funções em **AutoLISP** seguem as seguintes regras gerais:

– Estão contidas dentro de parênteses, todos os parênteses em **AutoLISP** são fechados, ou balanceados, isto é, um par de parênteses cerca uma função individual.

– As funções são lidas da esquerda para a direita.

– A primeira coisa dentro dos parênteses é o operador função, o qual é um comando para o interpretador AutoLISP fazer algo.

– O operador função é seguido pelos necessários argumentos. Argumentos são ítens individuais de informação que são requeridos pelo interpretador **AutoLISP** para alcançar o propósito da função.

– O operador função e seus argumentos são separados pelo menos por um espaço, de modo que o interpretador **AutoLISP** possa saber onde termina um e começa o próximo.

– Espaços extras e retornos de carro (CR e LF) dentro e entre funções não são requeridos e além disso são ignorados pelo interpretador **AutoLISP**. Isto significa que uma função pode ocupar muitas linhas em um arquivo ASCII.

– Funções usam caracteres ASCII padrão. Não fazem diferenciação entre maiúsculos e minúsculos.

– Os resultados retornados de funções podem ser usados como argumentos de outras funções em um processo chamado aninhamento.

AII.1.3. TIPOS DE DADOS FUNDAMENTAIS

Os dados usados pelo **AutoLISP** podem ser agrupados em tipos. Estas são categorias de dados que são distinguidos pelo que podem e não podem fazer quando referenciados e usados dentro de rotinas *lisp*.

Segue–se uma breve descrição dos tipos de dados do **AutoLISP** e suas funções.

(1) **Nomes de entidades**

Cada entidade gráfica criada no **AutoCAD** é listada em um banco de dados especial junto com todas as informações necessárias para reproduzir esta entidade.

Usando o **AutoLISP** é possível selecionar uma entidade a partir do banco de dados gráfico. Funções especiais retornam a entidade selecionada e a lista dos dados que a definem. O primeiro ítem tem uma lista deste tipo e o nome da entidade, um dado especial reservado para a manipulação da lista de dados de definição e distinção de outras entidades selecionadas.

(2) **Descrição de arquivos**

Funções especiais no **AutoLISP** podem abrir um arquivo em disco para armazenamento de informações ou leitura de informações contidas em arquivos de disco. Quando um arquivo é aberto no **AutoLISP**, é assumido um tipo de dado especial, chamado descritor de arquivos, o qual age como um apontador para este arquivo, pegando a posição da sua locação física no disco.

(3) **Inteiros e Reais**

O **AutoLISP** reconhece duas formas de dados numéricos, inteiros e reais.

Os inteiros podem ser processados rapidamente, mas são limitados em campo. Qualquer número entre –32768 e 32767, inteiro, é um número válido.

Reais são facilmente identificados porque sempre incluem um ponto decimal.

(4) **Listas**

Em **AutoLISP**, uma lista é qualquer grupo de ítens individuais de informações envolvidos por parênteses. Os ítens são separados uns dos outros por pelo menos um espaço. Ítens em uma lista podem ser números, caracteres, operadores funções, argumentos, ou mesmo outras listas. Elas derivam seus significados a partir da natureza de seus conteúdos e o contexto no qual aparecem. Assim, uma função é uma lista composta de um operador e seus argumentos. O tamanho de uma lista é o número de ítens individuais contidos nela. As listas podem ser de qualquer tamanho.

O **AutoLISP** trata as coordenadas X e Y como uma lista de dois números, assumindo que o primeiro número da lista é a coordenada X e o segundo a coordenada Y, nas versões 2.6 ou posteriores do **AutoCAD**, o terceiro número da lista será reconhecido como a coordenada Z de um ponto 3D.

(5) **STRINGS**

Strings são sequências de um ou mais caracteres (letras, números e marcas de pontuação) que não necessitam processamento matemático. Um *string* é sempre rodeado por aspas.

Strings poderão ser de qualquer tamanho, o computador alocará a memória necessária.

(6) **Símbolos**

O termo símbolo é um termo geral usado para descrever caracteres ASCII que esperam por algo. Nomes de variáveis são símbolos, operadores de funções também são. Um *string* é diferente de um símbolo porque é literal, se auto–representando. Da mesma maneira, um número é diferente de um símbolo porque este pode apenas representar seu próprio valor.

Os caracteres que montam um símbolo podem incluir qualquer combinação de letras ASCII, números e marcas de pontuação, exceto os seguintes:

( ) . ' " ;

Estas marcas de pontuação tem significado especial para o AutoLISP.

(7) SELECTION–SETS

O **AutoCAD** permite ao usuário selecionar grupos de entidades. A um grupo de entidades pode ser dado um nome. Os *selection–sets* podem ser tratados como grupos ou em partes, permitindo a manipulação do conjunto de entidades ou individualmente.

AII.1.4. AS FUNÇÕES DO AUTOLISP

A seguir apresenta–se um rápido sumário das funções do AutoLISP:

Funções Aritméticas:

/	executa divisão
*	executa multiplicação
+	executa adição
–	executa subtração
1+	adiciona um
1–	subtrai um
ABS	converte para valor positivo absoluto
EXP	retorna o antilogaritmo natural
EXPT	eleva um valor a uma potência específica
GCD	calcula o maior denominador comum de dois valores
LOG	retorna o logaritmo natural
MAX	retorna o maior de uma lista de números
MIN	retorna o menor de uma lista de números
REM	retorna o resto do resultado de uma divisão
SQRT	calcula a raiz quadrada

Funções do **AutoCAD**

COMMAND	executa um comando do **AutoCAD**
GETENV	retorna o valor de uma variável do ambiente ACAD
GETVAR	retorna o valor de uma variável do sistema ACAD
GRAPHSCR	muda a tela para o modo gráfico
MENUCMD	muda o menu na tela
OSNAP	aplica a opção OSNAP ao valor de um ponto
REDRAW	redesenha a tela
SETVAR	muda o valor de uma variável do sistema ACAD
TEXTSCR	muda a tela para o modo texto
VPORTS	lista os descritores de uma ***viewport*** (versão 10)

Funções de Contrôle e Programação:

~	retorna o complemento lógico de um inteiro
BOOLE	executa avaliações *booleanas* em inteiros
COND	avalia funções baseado em dadas listas de condições
FOREACH	repetidamente avalia funções para uma lista de valores

IF	avalia funções baseado em uma condição lógica
LOGAND	executa AND lógico de inteiros
LOGIOR	executa OR lógico de interios
LSH	retorna a inversão dos *bits* de um inteiro
PROGN	processa uma série de funções como uma única
TRACE	apresenta na tela a função e o resultado
UNTRACE	desliga a função TRACE
WHILE	avalia funções enquanto a condição lógica é verdadeira

Função de Conversão:

ANGTOS	converte ângulos em strings
ASCII	converte strings de caracteres para o código ASCII
ATOF	converte *strings* em reais
ATOI	converte *strings* em inteiros
CHR	converte códigos ASCII em *strings* de caracteres
FIX	converte reais em inteiros
FLOAT	converte inteiros em reais
ITOA	converte inteiros em *strings*
RTOS	converte reais em *strings*
TRANS	transfere pontos para outros sistemas de coordenadas

Funções de Controle de Dispositivos:

GRCLEAR	limpa temporariamente a tela gráfica
GRDRAW	desenha um vetor entre dois pontos
GRREAD	lê diretamente de um específico dispositivo
GRTEXT	apresenta dados na tela em modo texto ou restabelece o modo texto

Funções para Entidades Gráficas:

ASSOC	retorna a sub–lista associada a um elemento chave
ENTDEL	deleta entidades do banco de dados gráficos
ENTGET	retorna do banco de dados a sub–lista associada a uma entidade
ENTLAST	retorna do banco de dados a última entidade
ENTMOD	modifica a sub–lista associada a uma entidade
ENTNEXT	retorna do banco de dados a entidade seguinte a uma dada entidade gráfica
ENTSEL	seleciona uma entidade a partir de um ponto dado pelo usuário
ENTUPD	regenera entidades na tela

HANDENT	retorna a entidade associada ao manipulador (versão 10)
SSADD	adiciona entidades a *selection–sets*
SSDEL	remove entidades de *selection–sets*
SSGET	cria um *selection set* de entidades dadas
SSLENGTH	retorna o tamanho de um *selection–set*
SSMEMB	retorna verdadeiro se a entidade é membro do *selection–set*
SSNAME	retorna os nomes das entidades de um *selection–set*

Funções Primitivas e de Manipulação de Arquivos:

CLOSE	fecha arquivo em disco
FINDFILE	tenta localizar arquivos dentros dos diretórios de busca do AutoCAD
LOAD	carrega arquivos AutoLISP
OPEN	abre arquivos de disco para acesso
PRIN1	imprime expressões na tela e/ou arquivos
PRINC	imprime expressões com caracteres de controle
PRINT	imprime nova linha, expressão, espaço
READ–CHAR	lê um caracter do teclado ou arquivo
READ–LINE	lê um *string* do teclado ou arquivo
WRITE–CHAR	escreve um caracter em um arquivo
WRITE–LINE	escreve uma linha em um arquivo

Funções Geométricas:

ANGLE	retorna um ângulo absoluto (em radianos) entre dois pontos
DISTANCE	retorna a distância entre dois pontos
INTERS	retorna o ponto de interseção entre duas linhas
POLAR	referencia um ponto a partir de dados um ponto, ângulo e distância

Funções de Listas

APPEND	converte múltiplas listas em um a única
CAR	retorna o primeiro elemento da lista
CADR	retorna o segundo elemento da lista
CDR	retorna todos, exceto o primeiro da lista
CONS	cria uma lista, ou um par pontuado, dados dois elementos
LAST	retorna o último elemento da lista
LENGTH	retorna o número de elementos da lista
LIST	forma um lista a partir de elementos
MEMBER	retorna os elementos da lista a partir de um elemento dado
REVERSE	reverte a ordem dos elementos da lista
SUBST	substitui elementos em uma lista

Funções Relacionais:

>	Retorna verdadeiro se os valores da lista estão em ordem decrescente
<	retorna verdadeiro se os valores da lista estão em ordem crescente
>=	retorna verdadeiro se os valores da lista decrescem ou são iguais
<=	retorna verdadeiro se os valores da lista crescem ou são iguais
/=	retorna verdadeiro se os valores da lista são diferentes
=	retorna verdadeiro se os valores da lista são iguais
AND	retorna verdadeiro se todas as condições da lista são verdadeiras
ATOM	retorna verdadeiro se o item é um átomo
BOUNDP	retorna verdadeiro se o símbolo tem algum valor atribuído
EQ	retorna verdadeiro se dois símbolos tem atribuídos o mesmo valor
EQUAL	retorna verdadeiro se a lista de átomos avalia o mesmo resultado
LISTP	retorna verdadeiro se o símbolo é atribuído a uma lista
MINUSP	retorna verdadeiro se o símbolo é atribuído a um valor negativo

NULL	retorna verdadeiro se o símbolo não tem valor atribuído
NUMBERP	retorna verdadeiro se o símbolo é atribuído a um valor numérico
OR	retorna verdadeiro se o símbolo é atribuído ao valor numérico zero

Funções de Gerenciamento de Memória

ALLOC	especifica o tamanho dos nós de segmentos de memórias
EXPAND	reserva um número específico de segmentos
GC	torna todos os nós de memória não usados disponíveis para uso
MEM	apresenta o estado atual da memória AutoLISP
VMON	habilita função de paginamento virtual de memória

Funções Gerais

'	forma abreviada da função 'QUOTE'
ERROR	avalia funções se uma condição de erro existe
APPLY	aplica uma função a uma lista de argumentos

C:	prefixo que indica que a função trabalha como um comando ACAD
DEFUN	define novas funções
EVAL	avalia símbolos
LAMBDA	cria e avalia funções sem dar nomes
MAPCAR	aplica uma função sequencialmente para cada elemento da lista
NTH	retorna um elemento de uma lista, selecionado pelo número
SET	define o valor de um símbolo *quotado* ou referenciado
SETQ	cria um símbolo e/ou atribui a este um novo valor
PROMPT	apresenta mensagens na tela no campo de *prompt*
QUOTE	retorna o átomo sem avaliar
REPEAT	avalia funções um específico número de vezes
TERPRI	imprime uma nova linha na tela no campo de *prompt*
TYPE	retorna o tipo do dado
VER	retorna a versão do **AutoLISP** como *string*

Funções String:

READ	retorna a primeira lista ou átomo de uma *string*
STRCASE	converte caracteres para maiúsculo ou minúsculo
STRCAT	concatena *strings*
STRLEN	retorna o número de caracteres de um *string*
SUBSTR	retorna uma porção de um *string*

Símbolos:

PI	valor de π, aproximadamente 3.1415
PAUSE	executa uma pausa na função COMMAND e retorna a entrada de dados ao usuário

Funções de Tabela:

TBLNEXT	retorna a próxima entidade em uma específica tabela
TBLSEARCH	busca específicos símbolos na tabela

Funções Trigonométricas:

ATAN	retorna a tangente de um ângulo
COS	retorna o cosseno de um ângulo
LOG	retorna o logaritmo natural de um número
SIN	retorna o seno de um ângulo

Funções de Interface com o Usuário:

GETANGLE	aceita valores de ângulos no atual sistema de ângulos
GETCORNER	aceita os opostos vértices de um retângulo
GRTDIS	aceita a distância entre dois pontos
GETKWORD	aceita palavras chaves, baseado na prévia função INITGET
GETORIENT	aceita ângulos em orientação absoluta
GETPOINT	aceita coordenadas de um ponto
GETREAL	aceita números reais
INITGET	atribui opções e palavras chaves as funções GET*xxxx*

BIBLIOGRAFIA

CAPÍTULO II

[1] NEWMAN, W.M. e SPROULL, R.F., "Principles of Interactive Computer Graphics", McGraw–Hill Book Company, Segunda Edição, 1979.

[2] FOLEY, J.D. e DAM, A.V., "Fundamentals of Interactive Computer Graphics", Addison–Wesley Publishing Company, 1981.

[3] HARRIGTON, S., "Computer Graphics, A Programming Aproach", International Student Edition, McGraw–Hill, International Book Company, 1983.

[4] "CAD/CAM", Biblioteca Básica de Informática, no 29, Editora Século Ltda., 1986.

[5] BATHE, K.J., "Finite Element Procedures in Engineering Analysis", Prentice–Hall Inc., 1982.

[6] CLARK, D., "Computer Aided Structural Design", Intercience Publication, John Wiley and Sons, 1978.

[7] GIERTSEN, C. e UBERG, A.S., "Computer Graphics Hardware and Software Techniques", Computer Graphics Seminar, Rio de Janeiro, October 24–26, 1989.

[8] HOLTZ, B. e UDELL, J., "The Third Dimension – Three–Dimensional Modeling Brings New Excitement to Microcomputer–Based CAD", *Byte*, may, pp. 178–192, 1989.

[9] PROSISE, J., "3D CADD Workstation Tools for the PC Platform", *PC Magazine*, march, 173–244, 1990.

[10] TUTEN, P., "Claris CAD", *Byte*, june, pp. 209–210, 1989.

[11] CAVALCANTI, S.C., "Os Novos Recursos do CADTEC", *PC Mundo*, maio, pp. 34–35, 1989.

[12] PROSISE, J., "Proview – CADD – Minded Groupware Streamlines Drawing Review, Annotation", *PC Magazine*, march, pp. 56, 1989.

[13] ROBINSON, P., "The State of the Art in Workstation Graphics Today", *Byte*, february, pp. 255–264, 1989.

[14] "Produtos CAD sob Medida", *PC Mundo*, novembro, pp. 38–41, 1989.

[15] DVORAK, P., "Filling Out the FEA Family", *Machine Design*, february, pp. 74–80, 1990.

[16] FIGUEIREDO, A. "ICONUS – Sistema Gerenciador de Interfaces Gráficas", *III Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, Gramado , RS, pp. 43–49, 1990.

[17] MUNDIE, C., "Interacting with the Tiny and Immense", *Byte – Graphics Supplement*, vol. 14, no 4, pp. 279–288, april, 1989.

[18] ROBINSON, P., "Variations on a Screen", *Byte – Graphics Supplement*, vol. 14, no 4, pp. 251–158, april, 1989.

[19] KAPLAN, E., "Scientific Graphic Software – Not Just a Pretty Picture", *PC Magazine*, pp. 259–263, march, 1989.

[20] FLETCHER, Y., "A Tension–Compatible Patch for Shape–Preserving Surface Interpolation", *Computer Graphics and Applications*, may, pp. 70–75, 1989.

[21] ROSS,R., "The Map Geometric Transformation", *Computer Graphics and Applications*, may, pp. 45–55, 1989.

[22] BLINN, J.F., "How Many Different Cubic Curves are There?", *Computer Graphics and Applications*, may, pp. 78–83, 1989.

[23] WHELAN, T.G. e BRENNAM, M.J., "An Algorithm for Pictorial Projection on a Microcomputer", *Microsoftware for Engineering*, vol. 3, no 1, pp. 2–5, 1987.

[24] LIANG, Y.D. e BARSKY, B.A., "An Anlysis and Algorithm for Polygon Clipping", *Communications of the ACM*, vol 26, no 11, pp. 868–877, 1983.

[25] ROBERTSON, P.K., "Spatial Transformations for Rapid Scan–Line Surface Shadowing", *Computer Graphics and Applications*, march, pp.30–38, 1989.

[26] SAKURAI, H. e GOSSARD, D.C., "Solid Model Input Through Orthographic Views", *Computer Graphics*, vol. 17, no 3, pp. 243–252, 1983.

[27] CARVALHO, P.C.P., GATTASS, M. e MARTHA, L.F., "A Software Tool Which Allows Interactive Creation of Planar Subsection, and Applications to Educational Programs", PUC – Rio de Janeiro, 1989.

[28] SARRAGA, R.F., "Computer Modeling of Surfaces with Arbitrary Shapes", *Computer Graphics and Applications*, march, pp. 67–77, 1990.

[29] JANSSEN, T.L., "A Simple Efficient Hidden Lines Algorithm", *Computers and Structures*, vol. 17, no 4, pp. 563–571, 1983.

[30] HOFMEISTER, L.D., "A Drafting Program for Isoparametric Finite Elements", *International Journal for Numerical Methods in Engineering*, vol. 12, pp. 505–529, 1978.

[31] RECUERO, A., "Perspective Drawing of Three–Dimensional Objects", ,*Microsoftware for Engineers,* vol. 3, no 2, pp. 79–84, 1987.

[32] FONSECA, G.L. e GATTASS, M., "Algoritmos Simples de Linhas e Faces Escondidas", PUC Rio de Janeiro, pp. 1–26, 1989.

[33] LO, S.H., "Visualization of 3D Solid Finite Element Mesh by the Method of Sectioning", *Computers and Structures*, vol. 35, no 1, pp. 63–68, 1990.

[34] LO, S.H., "A Hidden–Line Algorithm Using Picture Subdivision Technique", *Computers and Structures*, vol. 28, no 1, pp. 37–45, 1988.

[35] LO, S.H., CHEUNG, Y.K. e LEUNG, Y.T., "An Algorithm to Display Three–Dimensional Objects", *Computers and Structures*, vol. 15, no 6, pp. 673–683, 1982.

[36] SMITH, R.E. e EVERTON, E.L., "Numerical Flow Field Visualization", *Computers and Structures*, vol. 30, no 1/2, pp. 411–419, 1988.

[37] PEGNA, J. e WILDE, D.J., "Spherical and Circular Blending of Functional Surfaces", *Transactions of the ASME*, vol. 112, may, pp. 134–142, 1990.

[38] WILHELMS, J. e SKDNNER, R., "A 'Notion' for Interactive Behavioral Animation Control", *Computer Graphics and Applications*, may, pp. 14–22, 1990.

[39] GOSS, M.E., "A Real Time Particle System for Display of Ship Wakes", *Computer Graphics and Applications*, may, pp. 30–35, 1990.

[40] TOENNIES, K.D., UDUPA, J.K., HERMAN, G.T., WORNOM III, I.L. e BUCHMAN, S.R., "Registration of 3D Objects and Surfaces", *Computer Graphics and Applications*, may, pp. 52–62, 1990.

[41] COELHO, L.C.G., GATTASS, M. e FILHO, W.C.F., "Um Algoritmo para Remoção de Linhas Ocultas", *III Simpósio Brasileriro de Computação Gráfica e Processamento de Imagens*, Gramado, RS, pp. 2–10, 1990.

[42] REQUICHA, A.A.G. e VOELCKER, H.B., "Solid Modeling: A Historical Sumary and Contemporary Assesment", *Computer Graphics and Applications*, vol. 2, march, pp. 9–24, 1982.

[43] REQUICHA, A.A.G., "Representations for Rigid Solid: Theory, Methods and Systems", *ACM Computer Survey*, vol. 12, no 4, pp. 437–464, 1980.

[44] FRANCIOSI, B.R.T. e CLAUDIO, D.M., "Projetos de Interfaces Gráficas", *XXII Congresso Nacional de Informática*, SUCESO, São Paulo, pp. 276–280, 1989.

[45] KOPP JUNIOR, R.V., "Computção Gráfica – Visualização Científica", *XXII Congresso Nacional de Informática*, SUCESU, São Paulo, pp. 792–797, 1989.

[46] SHOUMEI, W., "A Conception of Module Library and Data Base Management System for Finite Element Analysis", *Computers and Structures*, vol. 32, no 5, pp. 1073–1083, 1989.

[47] FIGUEIREDO, L.H. e GATTASS, M., "A Data Base Management System for Efficient Storage of Structural Loadings", *Computers and Structures*, vol. 32, no 5, pp. 1025–1034, 1989.

[48] GUDITZ, R., "An Alternative to IGES and PDES Direct Database Conversion", *Advances in Engineering Software*, vol. 11, no 2, pp. 107–108, 1989.

[49] ABEL, J.F. e COHEN, M.F., "Graphics for Numerics, Numerics for Graphics", 1986.

[50] THOMAS, G.A.N. e WISKIN, G.W., "Documentation of Software System", *Advances in Engineering Software*, vol. 2, no 1, pp. 29–33, 1980.

[51] JONES, M.V., "The Documentation and Checking of Computer Aided Engineering Computations", *Computers and Structures*, vol. 10, pp. 81–84, 1979.

[52] FAN, E.S., GINSBURG, S. e ABEL, J.F., "Some Aspects of Design and Programing of Computer Graphics for Structural Enginnering", *Advances in Engineering Software*, vol. 5, no 3,pp. 160–168, 1983.

[53] PFORTMILLER, L., "Data Structures in CAD Software", Byte, June, pp. 177–184, 1987.

[54] MAYER, R.J., "IGES – One Answer to the Problems of CAD Database Exchange", Byte, June, pp. 209–214, 1987.

[55] NEWSOME, S.L., SPILLERS, W.R. e VOSBURGH, A.M., "Quantitative Evaluations of Human Factors Savings in CAD Workstations", *Computers and Structures*", vol. 30, no 3, pp. 447–450, 1988.

[56] PEREIRA, J.C., "Metodologia para Desenvolvimento de Software para Engenharia", *VIII Congresso Latino–Americano e Ibérico sobre Métodos Computacionais para Engenharia*, vol. b, pp. 31–47, 1987.

[57] ROCHA, A.R., AGUIAR, T.C., NASCIMENTO, E.M. e FERREIRA, R.L., "Algoritmos Numéricos X Software Numérico : Uma Visão do Engenheiro de Software", *VIII Congresso Latino–Americano e Ibérico sobre Métodos Computacionais para Engenharia*, vol. b, pp. 31–47, 1987.

[58] VERROIJT, A., "Finite Element Calculations on a Micro–Computer", *International Journal for Numerical Methods in Engineering*, vol. 15, pp. 1570–1574, 1980.

[59] COGGINS, G.S. e TURCKE, D.J., "Interactive Graphics for Large Databases on Microcomputers", *Engineering Software for Microcomputers*, Proceedings of First International Conference, April, pp. 739–749, 1984.

[60] RECUERO, A. e GUTIERREZ, J.P., "The Solving of Large Linear Equations Systems with Microcomputers", *Engineering Software for Microcomputers, Procedings of First International Conference*, April, pp. 315–323, 1984.

[61] ALFORD, G.D., "A Low Cost CAD/CAM System with Simulation", *Enginnering Software for Microcomputers*, Procedings of First International Conference, april, pp. 210–214, 1984.

[62] WILSON, E.L., "Structural Analysis on Microcomputers", *Engineering Software for Microcomputers*, Proceedings of First International Conference, april, pp. 3–19, 1984.

[63] STELZER, J.F., "The Way to Microcomputer–Adequate Finite Element Software", *Enginnering Software for Microcomputers*, Porceedings of First International Conference, pp. 369–380, 1984.

[64] PEEKEN, H., LENSSEN, C. e WIDYANATA, J., "FE–Analysis on Microcomputers", *Engineering Software on Microcomputers*, Proceedings of First International Conference, pp. 425–434, 1984.

[65] RECUERO, A. e GUTIERREZ, J.P., "An Effective Bandwish reduction Algorithm for Microcomputers", *Enginnering Software for Microcomputers*, Proceedings of First International conference, pp. 519–530, 1984.

[66] HINTON, H. e ANDERSON, H.J., "A Finite Element Analysis System on a 16 BIT Microcomputer", *Engineering Software for Microcomputers*, Proceedings of First International Conference, pp. 908–922, 1984.

CAPÍTULO III

[67] ZIMMERLE,D., "Generation of Variable Order Elements by Generation", *Computers and Structures*, vol. 22, nº 3, pp. 291–297, 1986.

[68] BERRY, K.J., "Parametric 3D Finite–Element Mesh Generation", *Computers and Structures*, vol. 33, no 4, pp 966–976, 1989.

[69] GHASSEMI, F., "Automatic Mesh Generation Scheme for a Two or Three–Dimensional Triangular Curved Surface", *Computers and Structures*, vol. 15, no 6, pp. 613–626, 1982.

[70] VARDHANI, K.V.R. e PRASAD, N.S., "Mesh Generation for Spherical and Conical Surfaces Using Transfinite Interpolation", *Computers and Structures*, vol. 32, no. 6, pp. 1359–1362, 1989.

[71] LO, S.H., "Finite Element Mesh Generation Over Curved Surfaces", *Computer and Structures*, vol. 29, no 5, pp 731–742, 1988.

[72] ONWUBOLU, G.C., "Finite Element Mesh Generation of 3D–Surfaces in CAD", *Computers and Structures*, vol 32, no 1, pp. 31–36, 1986.

[73] SUSSMAN, T. e BATHE, K.J., "Studies of Finite Elements on Mesh Selection", *Computers and Structures*, vol. 21, no 1/2, pp. 257–264, 1985.

[74] SHEPHARD, M.S., GRICE, K.R., LO, J.A. e SCHOROEDER, W.J., "Trends in Automatic Three–Dimensional Mesh Generation", *Computer and Structures*, vol. 30, no 1/2, pp. 421–429, 1988.

[75] BLACKER, T.D., STEPHENSON, M.B., MITCHINER, J.L., PHILLIPS, L.R. e LIN, Y.T., "Knowledge System Approach to Automated Two–Dimensional Quadrilateral Mesh Generation", *Proceedings of Computers in Engineering Conference*, ASME, vol. 3, pp. 153–162, 1988.

[76] TSAMASPHYROS, G. e PAPAIOANNOU, M., "Mapping Techniques for the Automatic Mesh Generation Around Singularities", *Computers and Structures*, vol. 29, no 5, pp. 815–823,1988.

[77] CHAE, S.W. e BATHE, K.J., "On Automatic Mesh Construction and Mesh Refinement in Finite Element Analysis", *Computers and Structures*, vol. 32, no 3/4, pp. 911–936, 1989.

[78] REDDY, V.A., KUMAR, R. e SHARMA, S.P., "Mesh Generation and Graphic Check for 3D Building Frames", *Computers and Structures*, vol. 32, no 2, pp. 445–452, 1989.

[79] KAVEH, A., "Topological Properties of Skeletal Structures", *Computers and Structures*, vol. 29, no 3, pp.403–411, 1988.

[80] WU, S.C. e ABEL, J.F., "Representation and Discretization of Arhitrary Surfaces for Finite Element Shell Analysis", *International Journal for Numerical Methods in Engineering*, vol. 14, pp. 813–836, 1979.

[81] BOISSANMAT, J.D., "Geometric Structures for Three–Dimensional Shape Representation", *Transactions on Graphics*, ACM, vol. 3, no 4, pp. 266–286, 1984.

[82] HABER, R., SHEPHARD, M.S., ABEL, J.F., GALLAGHER, H. e GREENBERG, D.P., "A General Two–Dimensional Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mappings", *International Journal for Numerical Methods in Engineering*, vol. 17, pp. 1015–1044, 1981.

[83] HABER, R. e ABEL, J.F., "Discrete Transfinite Mapping for the Description and Meshing of Three–Dimensional Surfaces Using Interactive Computer Graphics", *International Journal for Numerical Methods in Engineering*, vol.18, pp. 41–66, 1982.

[84] COHEN, H.D., "A Method for the Automatic Generation of Triangular Elements on a Surface", *International Journal for Numerical Methods in Engineering*, vol. 15, pp. 470–476, 1980.

[85] SADEK, E.A., "A Scheme for the Automatic Generation of Triangular Finite Elements", *International Journal for Numerical Methods in Engineering*", vol. 15, pp. 1813–1822, 1980.

[86] LO, S.H.,"A New Mesh Generation Scheme for Arbirary Planar Domains", *International Journal for Numerical Methods in Engineering*, vol. 21, pp. 1403–1426, 1985.

[87] DENAYER, A., "Automatic Generation of Finite Elements Meshes", *Computers and Structures*, vol. 9, pp. 359–364, 1978.

[88] SHEPHARD, M.S., GALLAGHER, R.H. e ABEL, J.F., "The Synthesis of Near–Optimum Meshes With Computer Graphics", *International Journal for Numerical Methods in Engineering*, vol. 15, pp.1021–1039, 1980.

[89] ZIENKIEWICZ, O.C. e PHILLIPS, D.V., "An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by 'Isoparametric' Co–ordinates", *International Journal for Numerical Methods in Engineering*, vol. 3, pp. 519–528, 1971.

[90] LEE, Y.T., PENNINGTON, A. e SHAN, N.K., "Automatic Finite Element Mesh Generation From Geometric Models – A Point Based Approach", *Transactions on Graphics*, ACM, vol. 3, no 4, pp. 287–311, 1984.

[91] PHAI, N.V., "Automatic Mesh Generation With Tetrahedron Elements", *International Journal for Numerical Methods in Engineering*, vol. 18, pp. 273–289, 1982.

[92] CAVENDISH, J.C., FIELD, D.A. e WILLIAM, H. F., "An Approach to automatic Three–Dimensional Finite Element Mesh Generation", *International Journal for Numerical Methods in Engineering*, vol. 21, pp. 329–347, 1985.

[93] IMAFUKU, I., KODERA, Y., SAYAWAKI, M. e KONO, M., "A Generalized Automatic Mesh Generation Scheme for Finite Element Method", *International Journal for Numerical Methods in Engineering*, vol. 15, pp. 713–731, 1980.

[94] RECUERO, A. e GUTIERREZ, J.P., "Automatic Generation of a Triangular Mesh of Finite Elements", *Enginnering Software for Microcomputers, Proceedings of First International Conference*, pp. 393–401, 1984.

[95] RIBEIRO, F.L.B., "Técnicas de Geração Automática de Malhas", Primeiro Seminário de Doutorado, COPPE – Universidade Federal do Rio e Janeiro, 1987.

[96] TANIGUCHI, T., "Flexible Mesh Generator for Triangular and Quadrilateral Areas", *Advances in Engineering Software*, vol. 9, no 3, pp. 142–149, 1987.

[97] BAEHMANN, P.L., WITTCHEN, S.L., SHEPHARD, M.S., GRICE, K.R. e YERRY, M.A., "Robust, Geometrically Based, Automatic Two–Dimensional Dimensional Mesh Generation", *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 1043–1078, 1987.

[98] YERRY, M.A. e SHEPHARD, M.S., "Automatic Mesh Generation for Three–Dimensional Solids", *Computers and Structures*, vol. 20, no 1–3, pp. 31–39, 1985.

[99] TANIGUSHI, T., "An Interactive Automatic Mesh Generator for the Microcomputer", *Computers and Structures*, vol. 30, no 3, pp. 715–722, 1988.

[100] MOSCARDINI, A.O., LEWIS, B.A. e CROSS, M., "AGTHON – Automatic Generation of Triangular and Higher Order Meshes", *International Journal for Numerical Methods in Engineering*, vol. 19, pp. 1331–1353, 1983.

[101] JIN, H. e WIBERG, N.E., "Two–Dimensional Mesh Generation, Adaptative remeshing and Refinement", *International Journal for Numerical Methods in Engineering*, vol. 29, pp. 1501–1526, 1990.

[102] TALBERT, J.A. e PARKINSON, A.R., "Development of an Automatic Two–Dimensional Finite Element Mesh Generator Using Quadrilateral Elements and Bezier Curve Boundary Definition", *International Journal for Numerical Methods in Engineering*, vol. 29, pp. 1551–1567, 1990.

[103] GUIMARÃES, A.C.S. e FEIJÓ, R.A., "GAMAT2 – Gerador Automático de Elementos Finitos Lineares e Quadráticos", *III Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, pp. 70–79, 1990.

[104] AWRUCH, A.M., "Geração Automática da Malha num Ducto de Secção Transversal Variável e Eixo Curto", *VIII Congresso Latino–Americano e Ibérico sobre Métodos Computacionais para Engenharia*, vol. b, pp. 1–15, 1987.

[105] RIBEIRO, F.S., CARDOSO, J.R. e GAMBIRASIO, G., "Geração de Malha para Aplicação do Método dos Elementos Finitos em Estudos de Aterramento", *VIII Congresso Latino–Americano e Ibérico sobre Métodos Computacionais para Engenharia*, vol. b, pp. 163–179, 1987.

[106] BUELL, W.R. e BUSH, B.A.,"Mesh Generation – A Survey", *Journal of Engineering Industry*, ASME, ser. b, 95(1), 1973.

[107] HERMANN, L.R., "Laplacian – Isoparametric Grid Generation Scheme", *Journal of Engineering Mechanics Division*, ASME, vol. 102, pp. 749–756, 1976.

[108] GORDON, W.J. e HALL, C.A., "Construction of Curvilinear Coordinates System and Application of Mesh Generation", *International Journal for Numerical Methods in Engineering*, vol. 7, pp. 461–477, 1973.

[109] BARNHILL, E.E., BIRKHOFF, T. e GORDON,W.J., "Smooth Interpolation in Triangle", *Journal of Aproximation Theory*, vol. 8, pp. 114–128, 1973.

[110] WORDENWEBER, B., "Automatic Mesh Generation of 2 and 3 Dimensional Curvilinear Manifolds", Ph.D. Dissertation, Computer laboratory, University of Cambridge, England, 1981.

[111] YERRY, M.A. e SHEPHARD, M.S., "A Modified Quadtree Aproach to Finite Element Mesh Generation", *Computer Graphics Applications*, vol. 3, pp. 39–46, 1983.

[112] FREDERICK, C.O., WONG, Y.O. e EDGE, F.W., "Two–Dimensional Automatic Mesh Generation for Structural Analysis", *International Journal for Numerical Methods in Engineering*, vol. 2, pp. 133–144, 1970.

[113] AKYUS, F.A., "Natural Coordinate System, an Automatic Input Data Generation Scheme for Finite Element Method", *Nuclear Engineering Desing*, vol. 11, pp. 195–207, 1970.

[114] COOK, W. A., "Body Oriented (Natural) Coordinates for Generating Three Dimensional Meshes", *International Journal for Numerical Methods in Engineering*, vol. 8, pp. 27–43, 1974.

[115] ARAL, K., "Data Generation for 3D Finite Element System", *Computers end Engineering Conference*, ASME, vol. 77, pvp. 26, 1977.

[116] KAMEL, H.A. e EISENSTEIN, H.K., "Automatic Mesh Generation in Two and Three Dimensional Inter–Connected Domains", *Symposium on High Speed Computing of Elastic Structures*, Hege, Belgium, 1970.

[117] SUHARA, J. e FUKUDA, J., "Automatic Mesh Genration for Finite Element Analysis", *Advances in Computational Methods in Structural Mechanics Design*, pp. 607–624, 1974.

[118] JONES, R.E., "A Self–Orgaizing Mesh Generation Program", ASME, vol. 74, pvp. 13, 1974.

[119] CAVENDISH, J.C., "Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Methods", *International Journal for Numerical Methods in Engineering*, vol. 8, pp. 679–696, 1974.

[120] BATHIA, R.P. e LAWRENCE, K.L., "Two–Dimensional Finite Element Mesh Generation Based on Stripwise Automatic Triangulation". *Computers and Structures*, vol. 36, no 2, pp. 309–319, 1990.

[121] CAREY, G.F., SHARMA, M., WANG, K.C. e PARDHANANI, A., "Some Aspects of Adaptative Grid Computations", *Computers and Structures*, vol. 30, no 1/2, pp. 297–302, 1988.

[122] LUDWID, R.A., FLAHERTY, J.E., GUERINONI, F., BAEHMANN, P.L. e SHEPHARD, M.S., "Adaptative Solutions of Euler Equations Using Finite Quadtree and Octree Grids", *Computers and Structures*, vol. 30, no 1/2, pp.327–336, 1988.

[123] CRAWFORD, R.H., ANDERSON, D.C. e WAGGENSPACK, W.N., "Mesh Rezoning of 2D Isoparametric Elements by Inversion", *International Journal for Numerical Methods in Engineering*, vol. 28, pp. 523–531, 1989.

[124] PENMAN, J. e GRIEVE, M.D. "Self Adaptative Mesh Generation Technique for the Finite Element Method", *IEE Proceedings*, vol. 134, no 8, pp. 634–650, 1987.

[125] RENKA, R., "Triangulation and Bivariate Interpolation for Irregularly Distributed Data Points", Ph.D. dissertation, University of Texas at Austin, 1981.

[126] MANDEL, J., "Two–Level Domain Decomposition Preconditiong for P–version Finite Element Method in Three Dimensions", *International Journal for Numerical Methods in Engineering*, vol. 29, pp. 1095–1108, 1989.

[127] ANGELOV, T.A. e MANOACH, E.S., "A Point–in–Domain Identification Program", *Advances in Engineering Software*, vol. 11, no 2, pp. 99–106, 1989.

[128] MURTI, V., WANG, Y. e VALLIAPPAN, S., "Numerical Inverse Isoparametric Mapping in 3D FEM", *Computers and Structures*, vol. 29, no 4, pp. 611–622, 1988.

[129] FLORIANI, L., "A Pyramidal Data Structure for Triangle–Based Surface Description", *Computer Graphics and Applications*, march, pp. 67–78, 1989.

[130] CAREY, G.F. e ODEN, J.T., "Finite Elements, Computational Aspects", Prentice–Hall Inc., vol. 3, 1984.

CAPÍTULO IV

[131] JONES, F.H. e MARTIN, L., "The AutoCAD Database Book", Ventana Press, 1987.

[132] SCHAEFER, T. e BRITTAIN, J.L., "The AutoCAD Produtivity Book", Ventana Press, 1986.

[133] RAKER, D. e RICE, H., "Inside AutoCAD", New Riders Publishing, segunda edição, 1988.

[134] THOMAS, R.M., "Advanced Techniques in AutoCAD", Sibex Inc., segunda edição, 1988.

[135] "AutoCAD Release 10, Reference Manual", Autodesk Inc., february, 1989.

[136] "AutoLISP Release 10, Programmer's Reference", Autodesk Inc., december, 1988.

[137] "AutoCAD Applications Catalog", Autodesk Inc., 1985.

[138] Microequipo, "ACAD.LET", Microequipo Ldta., no 1, julho, 1990.

[139] Microequipo, "ACAD.LET", Microequipo Ldta., no 3, janeiro, 1991.

[140] MCGRATH, R.A., "AutoCAD – Based PCB Design", *Computer Graphics World*, january, pp. 77–80, 1987.

[141] BORRI, A., "Automatic Mapping for Seismic Risk on a Microcomputer", *Microsoftware for Engineers*, vol. 3, no 1, pp. 6–11, 1987.

[142] WOHLERS, T., "Bringing CADD to an Audience", *Computer Graphics World*, april, pp. 87–90, 1987.

[143] PROSISE, J., "New AutoCAD Embraces 3D Design", *PC Magazine*, january, pp. 33–34, 1989.

[144] PROSISE, J., "AutoCAD Smashes the 640K Barrier with Versions for OS/2 and 386 Machines", *PC Magazine*, march, pp. 33–34, 1989.

[145] ADELI, H. e PAEK, Y.J., "Computer–Aided Design of Structures Using LISP", *Computers and Structures*, vol. 22, no 6, pp. 939–956, 1986.

[146] ADELI, H. e PAEK, Y.J., "Computer–Aided Analysis of Structures in Interlisp Enviroment", *Computers and Structures*, vol. 23, no 3, 393–407, 1986.

CAPÍTULO V

[147] AFZALI, M., "Plane Intersections of a Three–Dimensional Mesh", *International Journal for Numerical Methods in Engineering*, vol. 18, pp. 1253–1259. 19xx.

[148] COLYER, B., "Animating Structure Plots: Experiments with a PERQ", *International Journal for Numerical Methods in Engineering*, vol. 20, pp. 1729–1743, 1984.

[149] AKIN, J.E. e STODDART,W.C.T., "Plane Intersection and Contours for General Isoparametric Solids", *Computers and Structures*, vol. 10, pp. 155–157, 19xx.

[150] BOURKE, P.D., "A Contouring Subroutine", Byte, June, pp. 143–150, 1987.

[151] DICKINSON, R.R., BARTELS, R.H. e VERMEULEN, A.H., "The Interactive Editing and Countoring of Empirical Fields", *Computer Graphics Applications,* may, pp. 34–43, 1989.

[152] SABIN, M.A., "Contouring – The State of the Art", *Fundamental Algorithms for Computer Graphics*, vol. 17, pp. 411–482, 1985.

[153] BERRY, K.J., "Parametric Finite Element Contour Mappinng", *Computers and Structures*, vol. 35, no 3, pp. 249–257, 1990.

[154] LUCENA, C.J.P., VEGA, I.S., SILVA, J.R., ARAKAKI, J. e FINGER, M., "AIPUSP: Ambiente para Projeto de Sistema de Produção", *XII Congresso Nacional de Informática*, SUCESU, São Paulo, pp. 665–670, 1989.

[155] CROSHECK, J. e FORD, M., "Simulation Takes Three–Wheeler for a Spin", *Mechanical Engineering*, November, pp. 48–51, 1988.

[156] PESQUERA, C.I., MCGUIRE, W. e ABEL, J.F., "Interactive Graphical Preprocessimg of Three Dimensional Framed Structures", *Computers and Structures*, vol. 17, no 1, pp. 1–12, 1983.

[157] KALDJIAN, M.J., "Three Dimensional Interactive Graphic Display Program for Michigan SAP (MSAP)", *Computers and Structures*, vol. 7, pp. 183–187.

[158] FREY, A.E., HALL, C.A. e PORSHING, T.A., "An Application of Computer Graphics to Three Dimensional Finite Element Analyses", *Computers and Structures*, vol. 10, pp. 149–154, 1979.

[159] PERUCCHIO, R., INGRAFFEA, A.R. e ABEL, J.F., "Interactive Computer Graphic Processing for Three–Dimensional Finite Element Analysis", *International Journal for Numerical Methods in Engineering*, vol. 18, pp. 909–926, 1982.

[160] PERUCCHIO, R. e INGRAFFEA, A.R., "Interactive Computer Graphic Preprocessing for Three–Dimensional Boundary–Integral Element Analysis", *Computers and Structures*, vol. 16, no 1–4, pp. 153–166, 1983.

[161] PERUCCHIO, R. e INGRAFFEA, A.R., "An Integrated Boundary Element Analysis System with Interactive Computer Graphics for Three–Dimensional Linear–Elastic Fracture Mechanics", *Computers and Structures*, vol. 20, no 1/3, pp. 157–171, 1985.

[162] LUCHI, M.L. e POGGIALINI, A., " Graphic Aids in a B.I.E. Program for 3D Elastostatic Analysis", *Advances in Engineering Software*, vol. 3, no 2, pp. 67–70, 1981.

[163] MARKATOS, N.C. e PERICLEOUS, C.A., "GRAFFIC: A Computer Package for the Interactive Graphical Representation of Fluid–Flow Phenomena", *Advances in Engineering Software*, vol. 5, no 2, pp. 86–91, 1983.

[164] CHRISTIANSEN, H.N. e BENZLEY, S.E., "Computer Graphics Display of Nonlinear Calculations", *Computer Methods in Applied Mechanics and Engineering*, vol. 34, pp. 1037–1050, 1982.

[165] GATTASS, M. e ABEL, J.F., "Interactive–Adaptative Large Displacement Analysis with Real–Time Computer Graphics", *Computers and Structures*, vol. 16, no 1/4, pp. 141–156, 1983.

[166] SAOUMA, V.E. e SIKIOTIS, E.S., "Interactive Graphics Nonlinear Constrained Optimization", *Computers and Structures*, vol. 21, no 4, pp. 759–769, 1985.

[167] SHORE, S., WILSON, J.L. e SEMSARZADEH, G.A., "Interactive Techniques with Graphical Output for Bridge Analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 5, pp. 197–209, 1975.

[168] BRAGA, F., DOLCE, M., FABRIZI, C. e LIBERATORE, D., "Interactive Building Analysis under Seismic Actions: The IBA Program", *Enginnering Software for Microcomputers*, Proceedings of First International Conference, april, pp. 210–214, 1984.

[169] ADELI, H. e FIEDOREK, J., "A Microcad System for Design of Steel Connections – I, Program Structure and Graphic Algorithms" , *Computers and Structures*, vol 24, no 2, pp. 281–294, 1986.

[170] ADELI, H. e FIEDOREK, J., "A Microcad System for Design of Steel Connections – II, Applications", *Computers and Structures*, vol 24, no 2, pp. 281–294, 1986.

[171] LEIMBACH, K.R., "CAL–GRAF – A Computer–Graphics Supplement for CAL", *Computers and Structures*, vol. 14, no 1/2, pp. 135–141, 1981.

[172] RAJAN, S.D., BELEGUNDU, A.D. e BUDIMAN, J., "An Integrated System for Shape Optimal Design", *Computers and Structures*, vol. 30, no 1/2, pp. 337–346, 1988.

[173] HEIZELMANN, E., BORTOLI, M.G.D., DRIESSEN, J.L., SANGOI, R. e BARBIERI, R., "Relato de um Desenvolvimento Integrado em CAE/CAD", *Maquinas e Metais,* junho, pp. 56–61, 1989.

[174] SANTOS, J.L.T., GODSE, M.M. e CHANG, K.H., "An Interactive Post–Processor for Structural Design Sensitivity Analysis and Optimization: Sensitivy Display and What–If Study", *Computers and Structures*, vol. 35, no 1, pp. 1–13, 1990.

[175] GROEHS, A. e SANTOS, M.I.G., "Interpretação Gráfica de resultados no Sistema GAELI", *III Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, Gramado , RS, pp. xxxxxxxx, 1990.

[176] JACOB, B.P. e EBECKEN, N.F.F., "Um Sistema Pré e Pós–processador Gráfico para Análise Estrutural", Anais do Colloquia'87 – *XXIV Jornadas Sul–Americanas de Engenharia Estrutural*, vol. 4, pp. 43–57, Porto Alegre, 1987.

[178] PAULINO, G.H. e GATTASS, M., "PréProcessamento de Estruturas Espaciais, com Reordenação Nodal, Usando Computação Gráfica Interativa", Relatório interno PUC–RJ – Departamento de Engenharia Civil, 1989.

[179] "The Finite Element Modeling System – FEMS, User Guide", Intergraph Corporation, versão 8.8, january, 1985

[180] "Intergraph–RandMicas (IRM) Analysis, Graphics Interface (Mechanical)", Intergraph Corporation, november, 1988.

[181] "Cadam GTSTRUDL Interface, Intallation and User's Guide", Gtices Systems Laboratory, Georgia Institute of Technology, version 21.0, 1988.

[182] "PATRAN Plus User Manual", PDA Engineering, July, 1987.

CAPÍTULO VI

[183] PRASAD, N.S. e VARDHANI, K.V.R., "Computer Aided Development of Surfaces for Cylinders and Cones", *Computers and Structures*, vol. 33, no 3, pp. 729–733, 1989.

[184] WU, E., "Techniques to Avoid Duplicate Nodes and Relax Restrictions on the Superelement Numberig in a Mesh Generator", *Computers and Structures*, vol. 15, no 4, pp. 419–422, 1982.

[185] "Intergraph–RandMicas (IRM) Analysis, Third–Party Interfaces, User's Guide", Intergraph Corporation, october, 1988.

[186] "ADEP – Analysis and Design of Offshore Structures, EDES – Specification od Structural Data, User's Manual", Cenpes – Petrobrás, september, 1987.

[187] "ANSYS – Engineering Analysis System, User's Manual", Swanson Analysis System Inc. june, 1987.

[188] FERRANTE. A.J., SANTOS, M.I.G., FILHO, E.C. e FRANCO, J.S.G., "Linguagem LORANE linear para Análise Estrutural por Computador", Curso de Pós–Graduação em Engenharia Civil, UFRJ, 1977.

[189] "Turbo Pascal, Reference Guide", Borland International, Versão 5.0, 1989.

[190] CUNHA, G.G., "Formato Genérico para Elementos Finitos", Publicação Interna, Petrobrás, 1990.

[191] ALENCAR, M.F.,"Geração Automática para Análise de Juntas Tubulares de Estruturas Offshore", Tese de Mestrado, COPPE–UFRJ, 1982.

[192] RODRIGUES, M.R. e CUNHA, G.G.,"Gedep, User's Guide", CENPES/COPPE, 1989.

[193] ARAÚJO, H.A., "Reordenação Nodal para Solução de Grandes Sistemas Considerando a Esparsidade", Tese de Mestrado, COPPE/UFRJ, 1986.

[194] CUNHA, G.G., LEVY, A., "GENESIS, User's Reference", CENPES/COPPE, 1990.

[195] "SMS – Structural Modeling System, User's Guide", Intergraph Corporation, march, 1988.

[196] CUTHILL, E. e MCKEE, J., "Reducing the Bandwith of Sparse Symmetric Matrix", ACM National Conference, 1969.

[197] LIU, W. e SHERMAM, A.H., "Comparative Analysis of the Cuthill–Mckee and Reverse Cuthill–Mckee Ordering Algorithms for Sparce Matrices", *Journal of Numerical Analysis*, vol. 13, no 2, abril, 1976.

[198] KING, J.P., "Automatic Reordering Scheme for Simultaneous Equations Derived From Network System", *International Journal for Numerical Methods in Engineering*, vol. 2, pp. 787–797, 1976.

[199] PINA, H.L.G., "An Algorithm for Frontwith Reduction", *International Journal for Numerical Methods in Engineering*, vol. 17, pp. 1539–1546, 1981.

CAPÍTULO VII

[200] ZIENKIEWICZ, O. C., "The Finit Element Method", McGraw–Hill Book Company, third edition, 1982.