JORGE PIMENTEL CINTRA

CONTRIBUIÇÕES AO ESTUDO DE REPRESENTAÇÃO DE SUPERFÍCIES COM O AUXÍLIO DO COMPUTADOR

Tese apresentada à Escola Politécnica da USP para a obtenção do título de Doutor em Engenharia

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
BIBLIOTECA

SÃO PAULO

- 1985 -

DEDALUS - Acervo - EPEC

31400024196

JORGE PIMENTEL CINTRA

Engenheiro Civil, Escola Politécnica da USP, 1974

Mestre em Engenharia, Escola Politécnica da USP, 1981

CONTRIBUIÇÕES AO ESTUDO DE REPRESENTAÇÃO DE SUPERFÍCIES COM O AUXÍLIO DO COMPUTADOR

Tese apresentada à Escola Politécnica da USP para a obtenção do título de Doutor em Engenharia

ORIENTADOR: Professor Doutor MILTON VARGAS

Professor Catedrático do Departamento de Engenharia de Estruturas e Fundações da Escola Politécnica da USP.

SÃO PAULO

- 1984 -

Biblioteca da Escola Politécnica
São Paulo

FT-79
e.2

AGRADECIMENTOS

À FAPESP, pela bolsa concedida para a realização deste trabalho.

Ao Professor Milton Vargas, pela amizade e pela orientação segura e oportuna.

Ao Nelson Covas, pelas horas e noites que passamos junto ao microcomputador.

Ao Professor Dante Guelpa, pelo estímulo e pelas observações.

À INTERTEC, pelo apoio e pela utilização do equipamento gráfico.

À MÉTODO INFORMÁTICA, pela utilização do microcomputador e do "plotter" de mesa.

Ao sr. João Baptista Esteves de Oliveira, special typist, pelos serviços de datilografia.

Ao Adri, pelos desenhos e pela ajuda que só nós sabemos.

Os Mapas

"Não acredito nos mapas, porque, quando se chega na área, ela nunca se parece com o que está neles".

- De um anúncio publicado pela BREWER'S SOCIETY.

Os mapas constituem simplesmente modelos do mundo real, e não existe o modelo perfeitamente fiel; somente sendo infiel em algum aspecto é que poderá representar o seu original".

- CHORLEY & HAGGET, Modelos físicos e de informação em geografia.

ÍNDICE GERAL

Lista de Figuras VII

Lista de Tabelas X

Lista de símbolos XI

ABSTRACT. XIII

RESUMO. XIV

I PARTE: INTRODUÇÃO GERAL 1

CAP. 1 - INTRODUÇÃO 1

1.1. Aplicações das Curvas de Isovalor 1
1.2. Os problemas da Engenharia e sua Solução através de Mapas Topográficos. 4
1.3. Representação do Relevo Topográfico 5
1.4. A Construção de Mapas pelo Processo Topográfico Clássico. 10

1.4.1. Levantamento de Dados. 11
1.4.2. Cálculos 12
1.4.3. Traçado das Curvas de Nível. 12
1.4.4. Inovações Recentes 15

CAP. 2 - A CONSTRUÇÃO DE MAPAS COM O AUXÍLIO DO COMPUTADOR. 17

2.1. O Modelo Digital do Terreno e os Critérios de Classificação 17
2.2. Modelos que utilizam Equações Analíticas. 19

2.2.1. Superfícies Globais. 22
2.2.2. Várias Superfícies Locais. 27

2.3. Modelos que utilizam uma Rede de Pontos 29

2.3.1. Malha Quadrada 29
2.3.2. Rede Triangular. 37

2.4. Outros Métodos. 42
2.5. Avaliação 43
2.6. Sistemas e Programas Existentes 44

II PARTE - ESTUDO DO MÉTODO DE REPRESENTAÇÃO DE SUPERFÍCIES DESENVOLVIDO POR HIROSHI AKIMA. 48

CAP. 3 - O MÉTODO DE AKIMA: ESTRUTURA E VISÃO GERAL 50

3.1. Triangulação. 50
3.2. Interpolação Sobre o Triângulo. 51
3.3. Suavidade dos Valores Interpolados. 53
3.4. Extrapolação. 54
3.5. Cálculo das Derivadas Parciais. 56

3.6. Considerações Gerais de Programação 58

3.6.1. Considerações Sobre a Triangulação 60
3.6.2. Considerações Sobre a Busca de Vizinhos. . . . 61
3.6.3. Considerações Sobre a Localização de Pontos. . 61
3.6.4. Considerações Sobre o Ajuste de Superfícies. . 62
3.6.5. Considerações Sobre o Cálculo de Derivadas . . 62

3.7. Exemplos de Aplicação 63
3.8. Propriedades do Método. 67

CAP. 4 - O ALGORITMO PROPOSTO POR AKIMA 69

4.1. Descrição Geral do Programa 69
4.2. Memória, Tempos e Modos de Computação 71
4.3. Descrição das Sub-rotinas 73

4.3.1. Função IDXCHG. 74
4.3.2. Sub-rotina IDTANG. 80
4.3.3. Sub-rotina IDCLDP. 90
4.3.4. Sub-rotina IDGRID. 94
4.3.5. Sub-rotina IDLCTN. 102
4.3.6. Sub-rotina IDPDRV. 109
4.3.7. Sub-rotina IDPTIP. 115

4.4. Conclusão 124

III PARTE: CONTRIBUIÇÕES AO MÉTODO DE AKIMA E ESTUDOS ESPECIAIS . . 126

5.1. Introdução. 126
5.2. Estudo Especial Sobre a Triangulação. 127

5.2.1. Triangulações. 127
5.2.2. O Algoritmo IBM. 130
5.2.3. O Algoritmo de Düppe e Gottschalk. 131
5.2.4. O Primeiro Critério de Lawson. 132
5.2.5. O Algoritmo e Critério de Cavendish. 133
5.2.6. O Segundo Critério de Lawson 136
5.2.7. O Critério de Nelson 139
5.2.8. O Critério de Sibson 142
5.2.9. O Critério de Lewis e Robinson 142
5.2.10. Conclusão 143

5.3. A Triangulação de Delaunay. 143

5.3.1. Definições 143
5.3.2. Exemplo Gráfico. 145
5.3.3. Propriedades 148
5.3.4. Otimização Local e Otimização Global 150
5.3.5. Breve Histórico e Aplicações 154
5.3.6. Algoritmos para a Construção 155

5.4. Implementação do Critério do Círculo. Comparações . . 158

5.4.1. Esquema. 158
5.4.2. Comparação de Operações e Memória. 159
5.4.3. Comparação de Tempos 160
5.4.4. Conclusão. 163

5.5. Aplicações da Técnica de Divisão e Conquista ao Método de Akima 163

5.5.1. Vantagens. 164
5.5.2. Notação e Estrutura de Dados 165
5.5.3. Construção da Triangulação de Delaunay 167
5.5.4. Análise de Tempos e Conclusões 173

5.6. Contribuições para a Determinação de Vizinhos 175
5.7. Implementação da Determinação de Vizinhos de Voronoi — Comparações 178

5.7.1. Esquema. 179
5.7.2. Comparação de Memórias 180
5.7.3. Comparação de Tempos 181
5.7.4. Conclusão. 181

5.8. Contribuições para o Cálculo das Derivadas. 183
5.9. Implementação do Novo Cálculo das Derivadas — Comparações. 186

5.9.1. Comparação de Memórias e de Tempo. 186
5.9.2. Comparação de Desenhos 188
5.9.3. Conclusão. 191

5.10. Estudo da Interpolação no Triângulo 191

5.10.1. Introdução. 191
5.10.2. Alguns Esquemas Gerais de Interpolação. . . . 192
5.10.3. Esquemas que Utilizam uma Analogia com a Teoria dos Elementos Finitos 193
5.10.4. Outros Esquemas de Interpolação no Triângulo. 197
5.10.5. Conclusão 199

5.11. Comparação de Tempos Totais 200

IV PARTE: COMPLEMENTOS AO PROGRAMA DE AKIMA E OUTRAS CONTRIBUIÇÕES. 203

CAP. 6 - DESENHO DAS CURVAS DE ISOVALOR 203

6.1. Introdução. 203
6.2. Opções nos Métodos que Utilizam Malhas Regulares. . . 204
6.3. Casos de Ambigüidade em Malha Regular 205
6.4. Breve Comparação de Três Métodos. 207
6.5. O Algoritmo de Snyder 210

6.5.1. Descrição do Programa Computacional. 211
6.5.2. Detalhes e Modificações para Implantação em Micros 215

6.6. Contribuições para o Traçado das Curvas de Nível. . . 217

6.6.1. Adaptação de uma Curva Suave 217
6.6.2. O Método de Cline. 219
6.6.3. Malha Triangular Refinada. 222
6.6.4. Desenho Baseado no Polinômio Interpolador. . . 223

CAP. 7 - ADAPTAÇÃO DO SISTEMA PARA UTILIZAÇÃO EM MICROCOMPUTADORES. 233

7.1. Introdução. 233
7.2. Capacidade de Memória 233

7.3. Fracionamento do Programa e Capacidade Máxima 234
7.4. Fracionamento do Programa e Gravação em Disco 238
7.5. Capacidade Máxima 240
7.6. Implantação do Sistema. 244
7.7. Esquema de Funcionamento. 246
7.8. A Opção por Microcomputadores 249

V PARTE: CONCLUSÃO FINAL 251

CAP. 8 - CONCLUSÃO E DIRETRIZES PARA TRABALHOS FUTUROS 251

8.1 Trabalhos futuros 251
8.2 Conclusão final 253

BIBLIOGRAFIA 258

LISTAS DE FIGURAS

1.1. A cota do terreno, como função de posição. 2
1.2. Mapa de precipitação atmosférica 3
1.3. Mapa de um campo magnético 3
1.4. Diagrama do Potencial energético de uma reação química 4
1.5. Pontos cotados 6
1.6. Perfil 6
1.7. Curvas de nível. 7
1.8. Esquema da poligonal e cobertura do terreno. 11
1.9. Construção das curvas de nível 13

2.1. Mapa utilizando um polinômio do 3º grau. 22
2.2. Onda de senos com um harmônico na direção x. 25
2.3. Onda de senos com dois harmônicos na direção x 25
2.4. Onda de senos com um harmônico em x e outro em y. 25
2.5. Onda de senos com dois harmônicos em cada direção. 25
2.6. Rede quadrada representativa do terreno. 30
2.7. Bloco perspectivo de uma malha quadrada. 31
2.8. "Distância média" entre pontos amostrados. 32
2.9. Rede triangular para estimar cotas 33
2.10. Definição de vizinhos através de um círculo 34
2.11. Esquema de atribuições de pesos 36
2.12. Pontos fornecidos para uma triangulação 37
2.13. Triangulação aceitável. 38
2.14. Interpolação das curvas de nível. 38
2.15. Duas opções de triangulação 39

3.1. Esquema de regiões externas de extrapolação. 55
3.2. Mapas comparativos de curvas de nível (exemplo de 30 pontos). 65
3.3. Mapas comparativos de curvas de nível (50 pontos). 66

4.1. Esquema dos dois programas originais de Akima. 70
4.2. Quadrilátero e produtos vetoriais. 75
4.3. Esquemas de configuração de quadriláteros a), b) 76

4.3. Esquemas de configurações de quadriláteros c), d) 77
4.4. Quadrilátero e nomenclatura. 78
4.5. Geometria para interpretação de DSQF e SIDE. 82
4.6. Matriz das distâncias calculadas 83
4.7. Determinação de arestas visíveis 85
4.8. Caso limite de alinhamento 87
4.9. Estudo de troca de triângulos. 88
4.10. Escolha de vizinhos mais próximos 92
4.11. Interpretação geométrica das funções SIDE e SPDT. 96
4.12. Retângulo de busca associado a um triângulo 97
4.13. Tipos de áreas externas 98
4.14. Restrição de regiões retangulares de busca. 99
4.15. Restrição de regiões triangulares de busca. 101
4.16. Divisão da área em 9 sub-regiões. 104
4.17. Localização de um ponto externo ao polígono 108
4.18. Esquema do produto vetorial 110
4.19. Posição relativa de vetores 114
4.20. Mudança de sistema: de x-y para u-v 118
4.21. Transformação de coordenadas (área retangular). 122

5.1. Geração da malha triangular (Cavendish). 133
5.2. Teste de configurações 135
5.3. Critério do círculo (Lawson) 136
5.4. Equivalência dos critérios do círculo e ângulo max-min 137
5.5. Início da contrução da rede (Nelson) 139
5.6. Determinação da melhor solução (Nelson). 140
5.7. Triangulação de Delaunay 146
5.8. Contra-exemplo na escolha da melhor configuração 150
5.9. Otimização local × otimização global 151
5.10. Esquema da função IDTROQ. 159
5.11. Comparação de tempos IDTANG (c/IDXCHG e IDTROQ) 162
5.12. Definição das funções SUCC e PRED 166
5.13. Definição da função FIRST 166
5.14. União de duas fronteiras convexas 168
5.15. Ilustração do processo de fusão 172
5.16. Esquema da rotina IDVZVR. 179
5.17. Comparação de tempos (IDCLDP × IDVZVR) 182

5.18. Anomalias na escolha de vizinhos. 184
5.19. Distorções para o cálculo das derivadas 185
5.20. Comparação de tempos (IDPDRV × IDPDER) 187
5.21. Curvas de nível, método de Akima. 189
5.22. Curvas de nível, método modificado. 190
5.23. Classes de continuidade de superfícies. 191
5.24. Esquema de distâncias (método de Mc Lain) 198
5.25. Comparação de tempos totais 201

6.1. Curvas de nível através de uma malha regular 204
6.2. Possibilidades de esquemas dentro de uma célula 206
6.3. Caminhamento de uma curva dentro da célula 209
6.4. Esquema da sub-rotina CNPLOT 212
6.5. Esquema da sub-rotina CNI. 212
6.6. Suavização de curvas (esquema IBM) 218
6.7. Esquema de subdivisão de triângulos. 222
6.8. Esquema de determinação de uma curva 225
6.9. Coordenadas globais do triângulo 226
6.10. Coordenadas isoparamétricas do triângulo. 226
6.11. Relações entre diferenciais 227
6.12. Curvas de nível num triângulo (1ª versão) 229
6.13. Curvas de nível num triângulo (versão melhorada). 232

7.1. Esquema de memórias do programa original 235
7.2. Esquema do programa de Akima 239
7.3. Esquema adaptado para microcomputadores. 239
7.4. Esquema do sistema para micros 245
7.5. Esquema de funcionamento 247

8.1. Bloco perspectivo do terreno 253

LISTA DE TABELAS

1.1. Valor usual do espaçamento das curvas de nível 8

3.1. Quadro dos termos do polinômio interpolar. 52
3.2. Tempos de computação (exemplos de Akima) 61
3.3. Pontos amostrados: coordenadas x, y, z 64

4.1. Tempo de computação de IDBVIP. 72
4.2. Tempo de computação de IDSFFT. 73

5.1. Tempo de computação (Green e Sibson) 156
5.2. Comparação de operações (IDXCHG × IDTROQ) 160
5.3. Comparação de memória (IDXCHG × IDTROQ) 160
5.4. Comparação de tempos (IDXCHG × IDTROQ). 161
5.5. Comparação de tempos em IDTANG 163
5.6. Comparação de memórias (IDCLDP × IDVZVR). 180
5.7. Comparação de tempos (IDCLDP × IDVZVR). 181
5.8. Comparação de tempos (IDPDRV × IDPDER). 188
5.9. Quadro de tempos totais. 200

7.1. Comparação de áreas de memória (Akima × modificado) 240

LISTA DE SÍMBOLOS E NOTAÇÕES

* Em geral, a notação vem descrita no lugar correspondente: tabela, figura, início da sub-rotina. Por comodidade, apresentamos a seguir uma relação, dividida por capítulos.

CAPÍTULO 1

E_H - escala horizontal

E_V - escala vertical

e - espaçamento vertical entre duas curvas de nível consecutiv

d - idem, espaçamento horizontal

dz - diferencial

CAPÍTULOS 4 e 7

U_1, U_2, ... - produtos vetoriais

NDP - número de pontos

NXI - número de pontos da malha na direção x

NYI - número de pontos da malha na direção y

XD, YD, ZD - listas contendo as coordenadas x,y,z respectivamente, dos pontos fornecidos

NT - número de triângulos

IPT - lista de vértices dos triângulos

NL - número de segmentos de borda

IPL - lista de segmentos de borda

IWL, IWP, WK - diferentes áreas de trabalho

NCP - número de pontos mais próximos de um vértice

IPC - lista de vizinhos de um vértice

XI, YI - listas contendo, respectivamente, as coordenadas das colunas e linhas de uma malha

NGP - lista da quantidade de nós da malha, por triângulos

IGP - lista dos nós que caem em cada triângulo

CAPÍTULO 5

V, $\mathbf{v}_i$, v_j, v_k - vértices de uma triangulação

NT - número de triângulos

NA - número de arestas

NL - número de arestas de borda

PRED, SUCC, FIRST, DELETE, INSERT, QTEST - funções

CH - fronteira convexa

V_L - conjunto de vértices da esquerda, numa partição vertical

V_R - conjunto de vértices da direita, numa partição vertical

DT (V_i) - triangulação de Delaunay para os vértices V_i

O(n) - tempo de computação da ordem de n, proporcional a n

$\Omega(n)$ - "piso inferior"; tempo mínimo teórico de computação proporcional a n

LM(S) - ponto mais à esquerda do conjunto S

RM(S) - ponto mais à direita do conjunto S

K_i - circuncírculo de um triângulo

NVZ - número de vizinhos de um vértice

NCP - número de pontos mais próximos de um vértice

IVZ - lista de vizinhos de um vértice

ABSTRACT

Contour maps are extremely useful in the solution of engineering problems and some areas of applications are:survey, minerals and petroleum exploration, study of magnetic fields, metereology and others.

The computer has been used for drawing automatically such maps and there are already several programs developed with this purpose.

We have selected as starting point of our study, a method with proved qualities, developed by Hiroshi Akima, entitled: "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points".

Our contribution to the work of Akima consists in reducing the computing time, improving the visual aspect of the map and in adapting the system to the use in microcomputers, with the consequent reduction of the cost of map's production.

Since Akima's program does not provide any routine for drawing maps, we further implement one for such purpose and this is another contribution of this thesis.

RESUMO

As curvas de nível são extremamente úteis na solução dos problemas da engenharia e algumas áreas de aplicação são: topografia, exploração de minérios e petróleo, estudo de campos magnéticos, meteorologia e outros.

Tem-se recorrido ao auxílio do computador para desenhar automaticamente esse tipo de mapas e são muitos os programas desenvolvidos com essa finalidade.

Escolhemos como ponto de partida de nosso estudo um método, de comprovadas qualidades, desenvolvido por Hiroshi Akima: "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points".

A nossa contribuição a esse trabalho consiste na redução do tempo de computação, na melhoria da qualidade do desenho e na adaptação do sistema para uso em microcomputadores, com a consequente redução do custo da produção de mapas.

Como o programa de Akima não fornece rotinas de desenho, nós implementamos uma com esse propósito e esta é uma outra contribuição desta tese.

I PARTE: INTRODUÇÃO GERAL

1. INTRODUÇÃO

1.1. APLICAÇÕES DAS CURVAS DE ISOVALOR

Neste trabalho fazemos diversas referências às curvas de nível aplicadas ao caso da topografia. No entanto, o método não se prende a um tipo concreto de aplicação, podendo ser utilizado no mapeamento de qualquer função de duas variáveis, ou seja, para qualquer função z caracterizada por valores associados a pontos definidos por duas outras variáveis (coordenadas x e y). As duas únicas restrições inerentes ao processo são, que essas duas variáveis devem ser medidas sobre eixos ortogonais, e que a cada ponto deve corresponder um único valor de z.

Como exemplos de aplicação podemos citar, entre outros:

- curvas de igual distribuição de minérios (ouro, prata, etc.)
- curvas de igual índice pluviométrico
- curvas de igual campo magnético
- curvas de igual potencial energético em reações químicas
- curvas de nível subterrâneas indicando a profundidade de jazidas de carvão, minerais, petróleo, etc.

- curvas de isomomentos ou isotensões no cálculo e estudo de estruturas de concreto.
- curvas de igual potencial do coração humano
- curvas de igual pressão barométrica
- curvas de igual distribuição de luz
- curvas de nível do fundo do mar e de rios
- etc.

Como exemplo ilustrativo apresentamos as figuras 1.2, 1.3 e 1.4, todas elas desenhadas por computador conectado com equipamento gráfico.

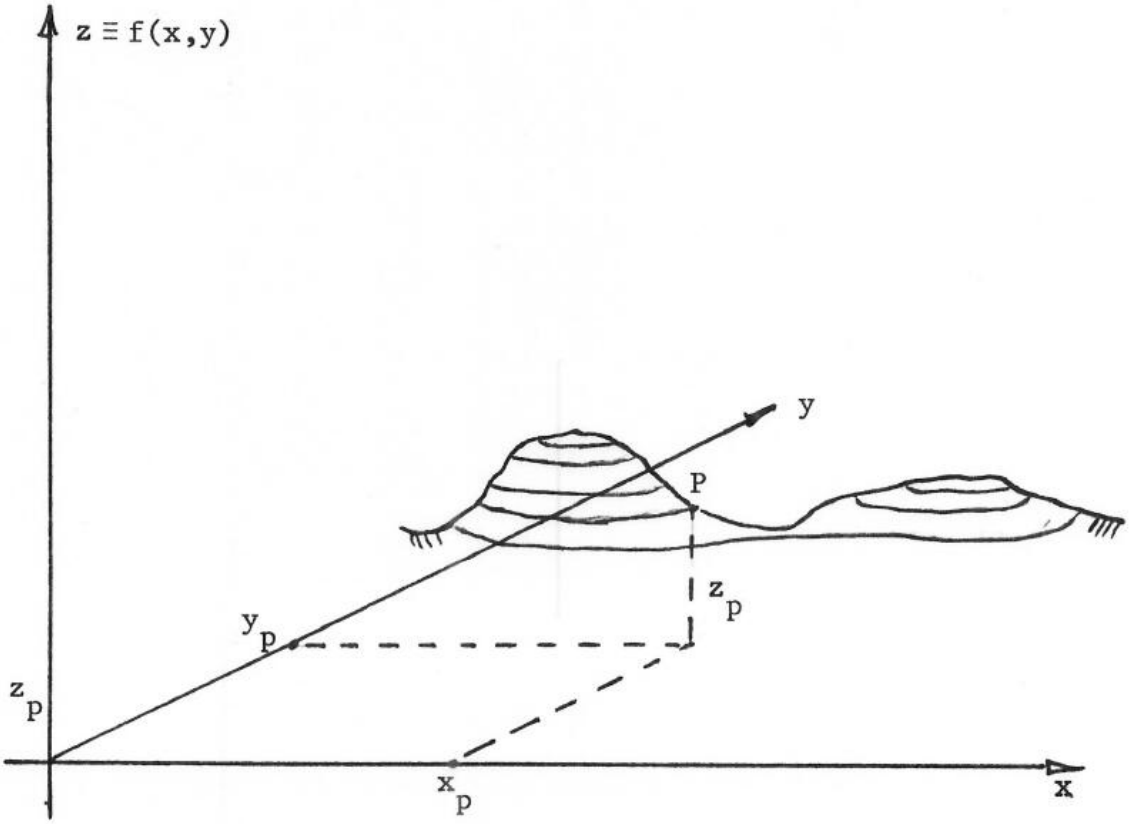


Figura 1.1 - A cota z do terreno, como uma função da posição geográfica x,y.

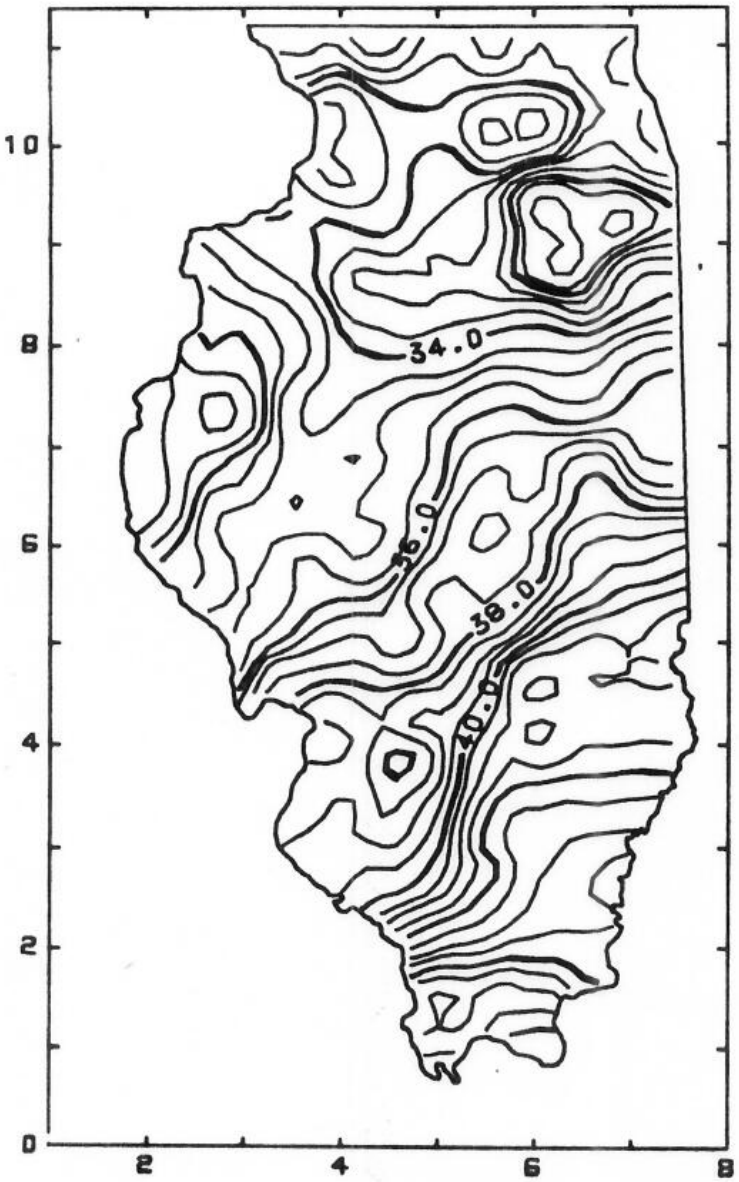


Figura 1.2 - Média anual de precipitação atmosférica. Desenho feito por computador, através do programa Surface II.

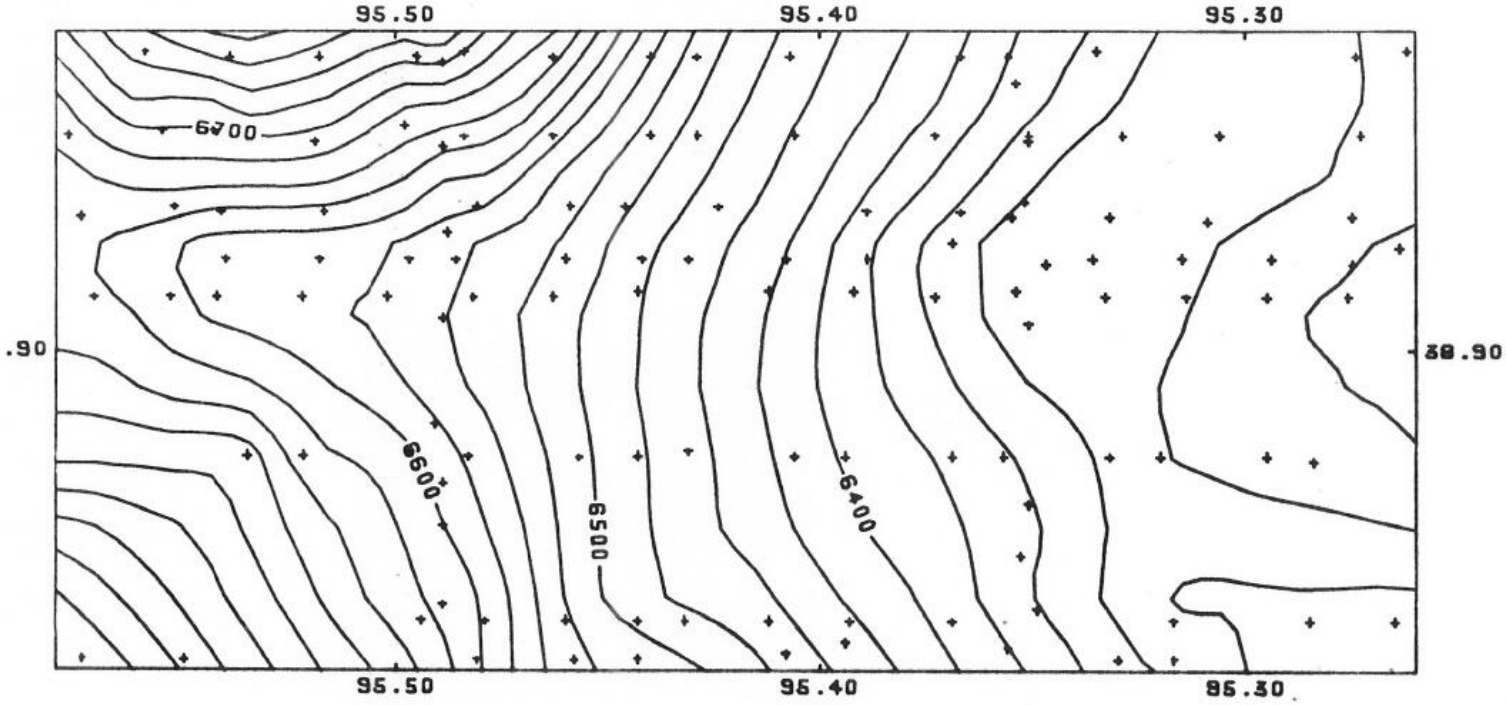


Figura 1.3 - Intensidade do campo magnético. Desenho feito por computador, também através do Surface II

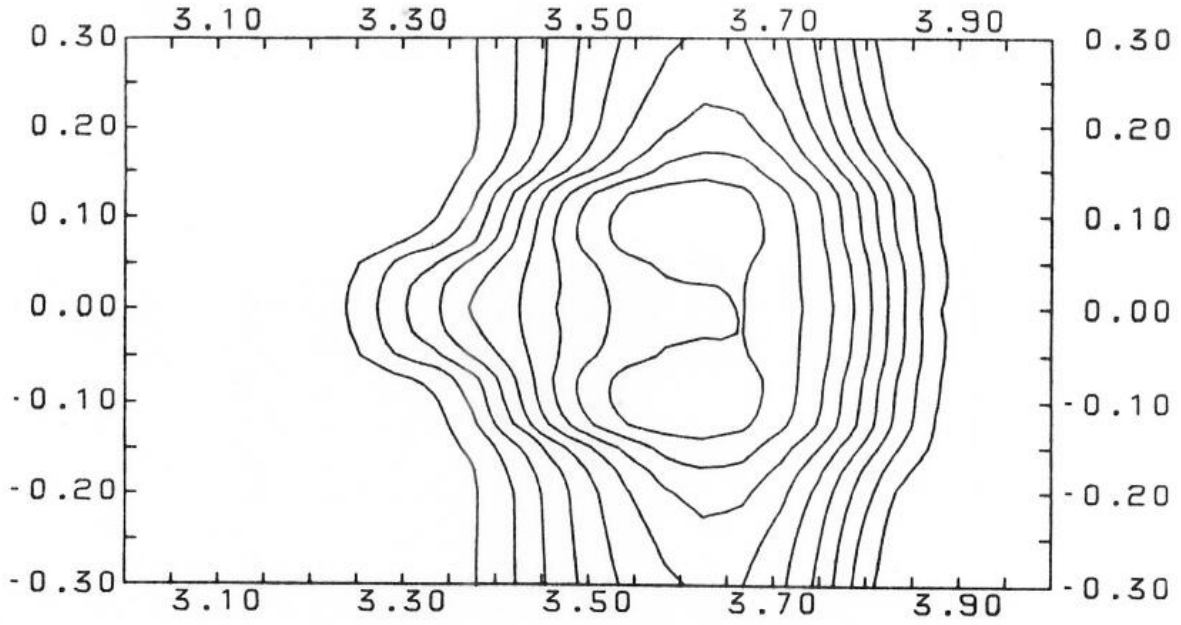


Figura 1.4 - Diagrama de Newton do potencial energético da reação entre Kr^+ e CH_4 para formar $KrCH_3^+$.

1.2. OS PROBLEMAS DA ENGENHARIA E SUA SOLUÇÃO ATRAVÉS DE MAPAS TOPOGRÁFICOS

A grande maioria dos problemas da engenharia requer a construção prévia de um mapa topográfico que represente adequadamente a superfície do terreno, e que se constitui assim num ponto de apôio fundamental para o projeto e execução das obras.

Dentro da engenharia civil, por exemplo, temos os casos da construção de edifícios, estradas, túneis, pontes, aeroportos, hidrovias, barragens, bacias de inundação, traçados de redes de comunicações, dutos de água, esgoto, óleo, loteamentos, projetos de urbanismo, etc. Qualquer obra assenta-se sobre um terreno, cujas características devem ser conhecidas.

Cada problema específico exige um tipo de mapa e cálcu-

los também específicos. Assim, podemos relacionar os seguintes problemas:

- criação de uma planta topográfica
- determinação da diferença de cota entre pontos
- análise de inclinações de taludes
- cálculo de áreas e volumes
- construção de perfis e secções transversais
- criação de uma rede de pontos cotados
- problemas de intervisibilidade e intersecção de linhas com a superfície do terreno
- construção de perspectivas tridimensionais.

A construção de uma planta topográfica pode solucionar, direta ou indiretamente a maior parte desses problemas; mas nem sempre ela é necessária ou fornece a precisão desejada. Em diversos casos pode ser preferível construir diretamente outro tipo de mapa (perfis, secções ou pontos cotados) mais adequados e mais práticos para a função específica que se deseja.

1.3. REPRESENTAÇÃO DO RELEVO TOPOGRÁFICO

O objetivo da representação do relevo topográfico é registrar e permitir a visualização da forma do terreno, e ao mesmo tempo fornecer os dados numéricos necessários para a solução dos problemas da engenharia: distâncias,e principalmente cotas de pontos desejados, com a precisão suficiente.

Descrevemos a seguir três maneiras principais de fornecer esses dados: pontos cotados, perfis e curvas de nível. Outros

modos, não descritos aqui são, por exemplo, hachuras, sombreados, e perspectivas.

Os pontos cotados (Figura 1.5) são assinalados apenas em pontos típicos, criteriosamente escolhidos. Convencionalmente

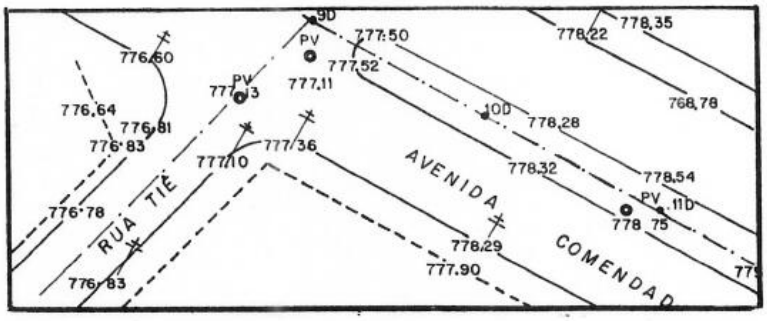


Figura 1.5 - Pontos cotados.

faz-se a separação da parte decimal do valor da cota com a própria representação do ponto. A precisão do processo é boa, mas a visualização do relevo é mínima. A representação por pontos cotados é utilizada isoladamente ou como complemento de outros mapas. São indispensáveis quando se requer precisão no valor das cotas (centímetro ou milímetro), coisa que não se consegue representar com outros tipos de mapa. São muito utilizados nos projetos de canalizações de água, esgoto, etc.

Os perfis representam cortes verticais do terreno ao longo de uma linha (Figura 1.6).

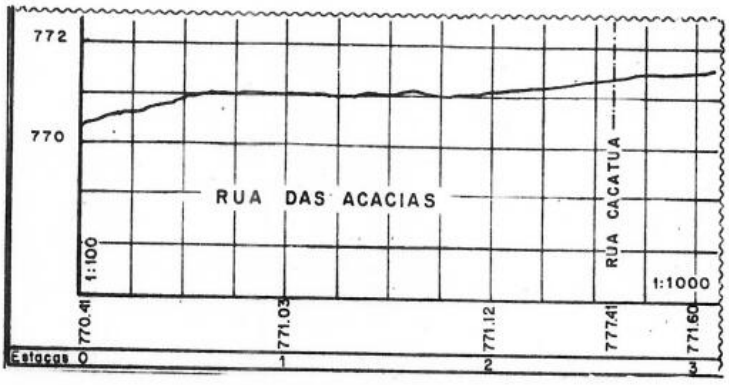


Figura 1.6 - Perfil do eixo de uma rua.

A escala vertical é escolhida maior que a horizontal para acentuar o revelo: $E_V = 2$ a $10\ E_H$.

O desenho é feito em papel milimetrado para facilidade, e é convencional o uso de cores: para linhas, o preto representa o terreno e o vermelho os perfis do projeto; para as aguadas, o amarelo representa áreas de corte e o vermelho áreas em aterro.

As <u>curvas de nível</u> ou linhas de cota constante constituem a representação mais comum do relevo topográfico.

Correspondem à intersecção do terreno com planos horizontais convenientemente espaçados. Para ser mais rigorosos devemos falar na intersecção com superfícies equipotenciais. Nessa formulação uma imagem clássica, cfr. Raiz(1969), é representada pelas diferentes linhas costeiras que se formariam se o mar subisse com intervalos de cota constante.

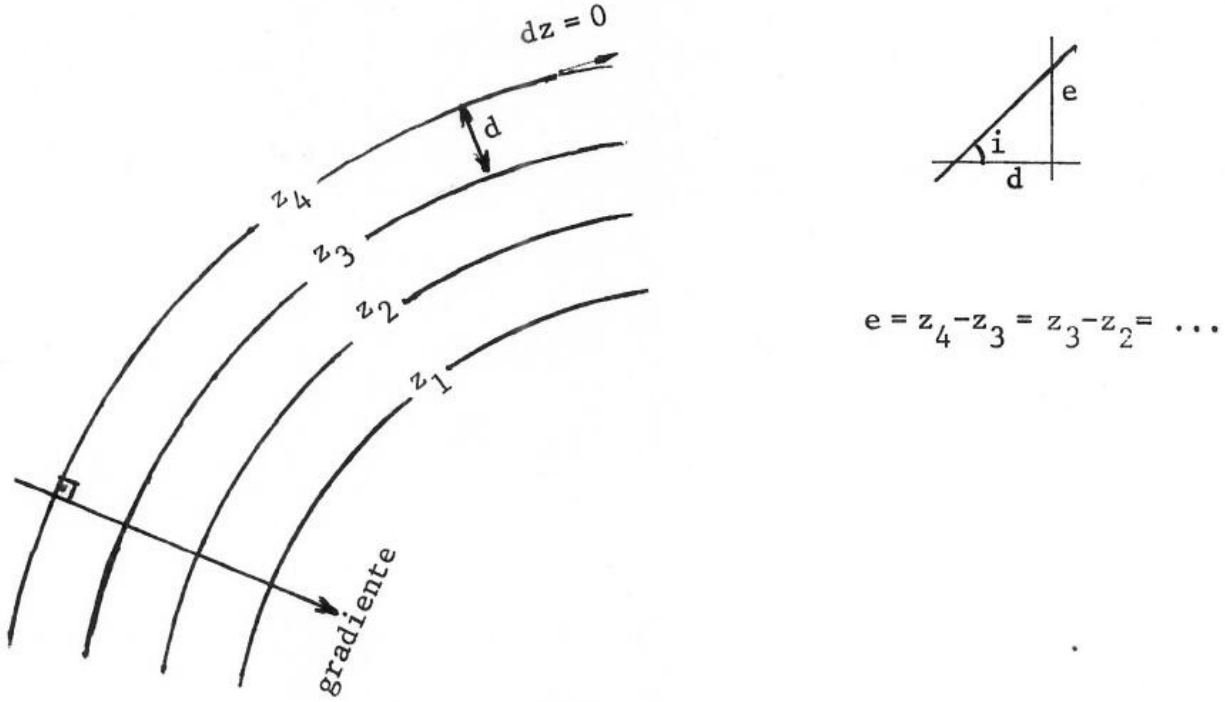


Figura 1.7 - Curva de nível, gradiente, espaçamento.

Nos desenhos utilizam-se cotas inteiras com espaçamento conveniente e constante. O espaçamento vertical (e) é a diferença de cota entre duas curvas consecutivas. O espaçamento horizontal (d) diminui com o crescimento da declividade do terreno: nas montanhas as curvas estão mais próximas, e nos vales mais afastadas.

Matematicamente a curva de nível corresponde à linha de z constante ou derivada dz/ds nula. A declividade máxima, correspondente ao gradiente, é normal à curva no ponto, e é dada por $tg i = e/d$ (conforme Figura 1.7). As águas (rios, chuvas) escoam perpendicularmente às curvas de nível.

A escolha da eqüidistância depende da escala utilizada para o mapa, da importância do desenho e da precisão do levantamento.

Há plantas de irrigação com espaçamento de 25 cm, e mapas com curvas de 100 em 100 metros. Os valores usuais (Tabela 1.1) costumam ser definidos em função da escala.

TABELA 1.1 - Valores usuais do espaçamento em função da escala

Esc. 1:	e	Esc. 1:	e
500	0,5 m	50.000	25 m
1.000	1,0	100.000	50
2.000	2,0	200.000	100
2.500	2,5	250.000	100
5.000	5,0	500.000	200
10.000	10,0	1.000.000	200
25.000	10,0	10.000.000	500

DADOS HISTÓRICOS - Em tempos hitóricos, em que os mapas eram

simbólicos, o relevo era representado por ilustrações: desenhos esquemáticos sem preocupação quantitativa.

Com os tempos modernos – navegações marítimas, mapas portulanos, projeção sobre a esfera, trabalhos de Mercator – os mapas passaram a ter preocupação pela precisão numérica e gráfica. Em planta os trabalhos desenvolveram-se rapidamente chegando-se a boas soluções. O desenvolvimento quantitativo do relevo demorou mais tempo.

As notícias mais remotas que se têm sobre as curvas de nível atingem o engenheiro holandes N. Cruquius (1728 ou 1730) que as empregou pela primeira vez para representar o fundo do rio Merwede, para o estudo de sua navegabilidade. Em 1737 Filipe Buache (1700-1773), geógrafo francês, empregou as curvas de nível para indicar as diferentes profundidades do canal da Mancha. Mas passaram-se ainda muitos anos antes de que fossem empregadas nos mapas terrestres. O primeiro mapa importante com curvas de nível é o da França, por Dupain-Triel (1791). Nos mapas militares de precisão (século XIX), empregaram-se curvas de nível, mas só nos originais: nas reproduções eram omitidas ou parcialmente suprimidas para evitar confusões pois só se utilizavam os tons branco e preto. Com a invenção da litografia, o Serviço Geográfico Militar da Inglaterra confeccionou mapas com curvas sombreadas em tons de cinza e, antes de completar a tiragem, utilizou-se a impressão a cores, recém aperfeiçoada. Em 1878 adotou-se, pelo Serviço Geológico dos Estados Unidos, o sistema em uso atualmente de representar as curvas de nível em sépia (terra-siena), os rios em azul, e os

detalhes relativos a culturas e edificações em preto.

1.4. A CONSTRUÇÃO DE MAPAS PELO PROCESSO TOPOGRÁFICO CLÁSSICO

Essa elaboração pode ser dividida em três etapas. A primeira é constituída pelo levantamento: o trabalho de campo em que se obtém os dados. A segunda consiste nos cálculos: tratamento e transformação adequada dos dados para que forneçam os elementos desejados. A terceira é o desenho ou representação gráfica do terreno.

Os mapas constituem um modelo iconográfico (do grego, icone = imagem), uma abstração e simplificação da realidade. Na verdade, a superfície da terra é um "continuum" tridimensional, e o que se pretende é representar a elevação z do terreno acima de uma cota-base arbitrária ou, de preferência, referida ao nível do mar. Essa altitude z pode ser assimilada a uma função cujo valor depende da posição geográfica definida por coordenadas x, y com origem x_0, y_0 arbitrária.

Nessa concepção, a planta topográfica é constituída pela projeção ortogonal em x, y das linhas de z constante. Trata-se pois de uma representação discreta de uma superfície contínua. Os detalhes e informações contidas entre duas curvas de nível são perdidas, podendo ser aproximadamente recuperadas através de interpolações que serão mais ou menos confiáveis em função da qualidade do mapa. Essa qualidade do mapa e sua fidelidade ao terreno estão diretamente relacionadas com a escala, o espaçamento vertical, a prática do desenhista, etc.

1.4.1. LEVANTAMENTO DE DADOS

Como a superfície do terreno é constituída por infinitos pontos, impõem-se uma escolha de valores discretos, que sejam representativos e permitam uma representação confiável.

Os critérios que norteiam essa escolha devem ser técnicos e econômicos, procurando conciliar o pouco dispêndio (reduzido número de pontos) com a qualidade do mapa (mais pontos, fidelidade ao terreno).

Dessa forma, os pontos levantados na topografia não são amostras escolhidas ao acaso. Há uma nítida preferência por pontos notáveis: pontos mais altos (picos), mudanças de declive, linha de cumeada ou de vale , linha d'água, etc.O profissional de campo costuma escolher os pontos de tal forma que uma interpolação linear na direção do gradiente seja uma aproximação razoável.

A distribuição espacial dos pontos também não é aleatória pois os métodos de levantamento procuram cobrir toda a região com uma distribuição relativamente uniforme dos pontos sobre a superfície a ser mapeada. As estações da poligonal são

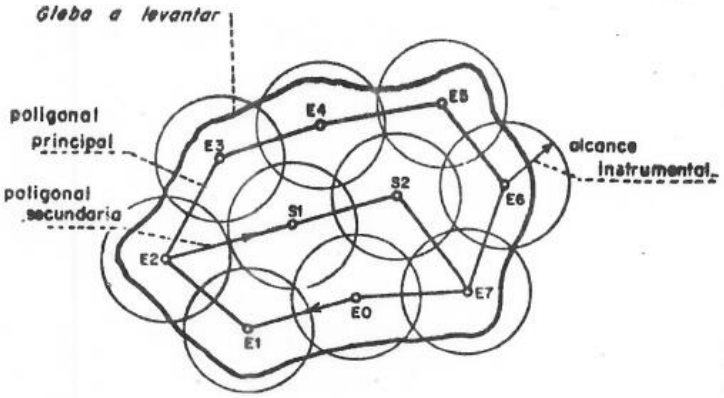


Figura 1.8 - Esquema da poligonal e a cobertura do terreno.

cuidadosamente escolhidas, bem como os pontos detalhes, de forma a não deixar regiões com baixa densidade de pontos.

1.4.2. CÁLCULOS

Os valores obtidos em campo constituem os dados brutos que geralmente não são utilizados diretamente. Os ângulos devem ser convertidos em azimutes verdadeiros e as distâncias devem ser reduzidas a sua projeção horizontal. As coordenadas polares são transformadas em retangulares.

Os dados são levantados de forma a permitir um controle de sua qualidade: a poligonal é fechada e portanto pode-se calcular o erro angular, que deve ser inferior a um valor pré-fixado; as distâncias permitem o cálculo das coordenadas das estações e a determinação de um erro de fechamento linear que também deve estar dentro de padrões admissíveis.

O volume de cálculos, tendo em conta também o número de pontos, é considerável e as calculadoras, de bolso, de mesa, ou microcomputadores, prestam auxílio inestimável.

Terminados os cálculos dispõe-se das coordenadas (x, y e z) dos pontos.

1.4.3. TRAÇADO DAS CURVAS DE NÍVEL

Dispondo-se das coordenadas dos pontos pode-se passar à fase de desenho, com o auxílio do coordenatógrafo, escalas, compasso, transferidor, etc.

Inicialmente, escolhida a escala de acordo com a finali-

dade, determina-se o tamanho do papel (A_0,A_1,...) e desenha-se uma quadrícula, com espaçamento uniforme, geralmente de 5 ou 10 cm. Demarcados os pontos com suas cotas, procede-se à interpolação dos pontos de cota inteira. Para isso tomam-se pares de pontos próximos e definem-se as posições de cota inteira ao longo da reta que os une (conforme esquema da Figura 1.9). A interpolação, quase sempre linear, pode ser feita através de cálculos, ábacos ou expedientes empíricos que dão resultados aceitáveis.

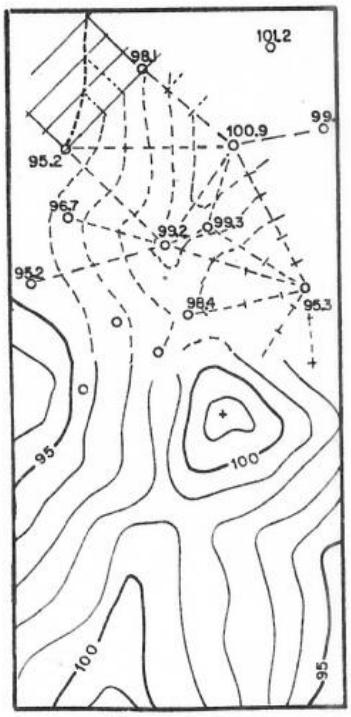


Figura 1.9 - Construção das curvas de nível.

Procura-se então visualizar as curvas mestras (95,100, etc.) e delineá-las no desenho, para depois completar com as demais

A união estrita dos pontos interpolados produz um desenho muito anguloso, mecânico. Impõe-se uma suavização e por isso retocam-se as curvas de modo a torná-las coerentes com a conformação geral do terreno.

Essa suavização permite diminuir as angulosidades. De certa forma é arbitrária e depende da perícia e da prática do desenhista.

O desenho assim realizado, em que não se apagam as construções auxiliares, constitui o original cartão.

A partir dele são tiradas as cópias em vegetal ou base plástica (cronaflex). O traçado a tinta é feito com um tira-linhas móvel. Deve-se ressaltar as curvas mestras, que são reforçadas em sua espessura e levam (e só elas) a indicação da cota. Na Figura 1.9 são as de cota 95 e 100.

Pode-se tirar a seguir cópias heliográficas, que normalmente são as utilizadas para projetos e medições sobre planta.

As curvas de nível, quando bem traçadas, apresentam harmonia de conjunto e permitem por si mesmas visualizar a forma do terreno.

Convém notar também que as curvas de nível não se cruzam nem se interrompem; não se tangenciam a si mesmas; não formam espirais; cortam as linhas d'água ortogonalmente, formam curvas fechadas em torno de elevações e tendem ao paralelismo com as linhas de vales.

Para o estudo teórico das curvas de nível pode ser útil a consulta a algum manual clássico: Jordan (1944), Carvalho Xerez (1947), Kissam (1947), Davis & Foot & Kely (1942) ou Breed (1966).

1.4.4. INOVAÇÕES RECENTES

Como observação final, conviria falar de um dos progressos recentes da topografia, a digitalização de dados e a automação dos cálculos. Tendo em vista a enorme quantidade de dados envolvida em problemas de mapeamento, torna-se bastante compensador estudar as interfaces do sistema a fim de otimizar as operações de transferência de dados e conseqüentemente minimizar também os possíveis erros na transmissão de valores de um subsistema a outro.

Dessa forma, foram desenvolvidos aparelhos registradores para facilitar a tarefa de digitalização de pontos, durante o próprio levantamento no campo. Dispensa-se assim a caderneta de campo, com suas trabalhosas anotações por colunas.

Como exemplo desses aparelhos temos o Reg-Elta 14, o EOT 2.000, o Aga-Geodat 120, etc. Esses aparelhos permitem o registro automático e/ou manual de todos os dados necessários: ângulos horizontais e verticais, distâncias inclinadas ou hozontais, diferença de cota, código dos pontos, etc. Trabalham tanto com fita perfurada (plastificada para resistir às interpéries) como com pequenas fitas magnéticas. O tamanho é bastante cômodo (dimensões de um deles: 9 × 17 × 3 cm; 0,6 Kg) e a capacidade de memória atende plenamente as necessidades (32 K bytes podendo ser ampliada indefinidamente através da transferência dos dados para fitas magnéticas, com grande capacidade, no próprio campo.

A digitalização dos pontos sob essa forma permite a trans-

ferência direta dos valores registrados em campo para os computadores de cálculo, poupando a fase intermediária de transferência manual, escrita ou digitada, dos dados no escritório.

Um dos inconvenientes das fitas atualmente em uso é que só servem para o tipo de calculadora para o qual foram planejadas e a transferência das informações para outro meio,disquete por exemplo, ainda é muito dispendiosa. Com o tempo se prevê uma maior flexibilidade de uso, e versatilidade de intercomunicações.

A automação dos cálculos topográficos caminha paralelamente ao desenvolvimento dos instrumentos de cálculo. As calculadoras mecânicas foram substituídas pelas eletrônicas: com 4 operações; funções trigonométricas; programáveis. O porte destas últimas, tanto em memória como em rapidez,também cresceu muito. Atualmente muitas firmas dispõe de microcomputadores, de diversos tipos, para esse serviço.

Os programas-prontos,disponíveis ou desenvolvidos pelas firmas, são inúmeros e alguns são oferecidos na compra de um micro.

2. A CONSTRUÇÃO DE MAPAS COM O AUXÍLIO DO COMPUTADOR

Mas além de realizar mais rapidamente os cálculos numéricos tradicionais, o computador pode prestar outro tipo de auxílio à topografia. Atendendo a esta etapa inicial, permite a criação de pontos adicionais, a geração de curvas de nível (seqüência de pontos fornecidos por suas coordenadas), a representação tridimensional do terreno em vídeos (telas), desenhos perspectivos, comando de mesas automáticas de desenho e, principalmente, a criação de um modelo digital do terreno, que é a base para a solução numérica de uma série de problemas da engenharia.

2.1. O MODELO DIGITAL DO TERRENO E OS CRITÉRIOS DE CLASSIFICAÇÃO

Para a representação em computadores, é útil encarar o problema sob o ângulo de um modelo digital do terreno: este é constituído, por exemplo, por um conjunto de pontos significativos do terreno, armazenados através do registro das três coordenadas x, y e z. Pode ser representado também por superfícies matemáticas ou ainda de outras maneiras, como veremos.

Um dos modos de classificar os modelos baseia-se em suas procedências. Por exemplo: pontos fornecidos pela topografia clássica, pela aerofotogrametria, por uma mesa digitalizadora,

etc.

Outra forma de classificar os modelos baseia-se na distribuição espacial dos pontos fornecidos. Podemos ter uma distribuição regular (malhas retangulares, quadradas, triangulares, hexagonais) ou uma distribuição irregular (aleatória,uniformemente distribuída, ou ainda concentrada em regiões).

Quanto ao modo de representar os valores fornecidos podemos ter uma interpolação, quando o modelo interpola os valores intermediários respeitando exatamente os valores fornecidos; ou então uma aproximação, quando os valores fornecidos só são respeitados aproximadamente, como é o caso dos métodos de mínimos quadrados.

Com relação à abrangência e extensão,os métodos podem ser de caráter local, quando o cálculo do valor da cota de um ponto só depende dos dados situados numa certa vizinhança; ou global, quando todos os dados influem no cálculo dessa cota.

No entanto, parece-nos preferível dividir os modelos digitais de acordo com o modo específico de representar o terreno, isto é, de acordo com o modelo matemático empregado. Dessa forma sugerimos a divisão em dois grandes grupos, que secundariamente englobariam as divisões já apresentadas.

O primeiro desses grupos é aquele que trabalha fundamentalmente com equações matemáticas, definindo uma superfície contínua para representar o terreno. As equações matemáticas podem ser globais, aplicáveis a toda a área a ser mapeada; ou locais, quando se subdivide o terreno em várias parcelas. As funções escolhidas podem ser polinômios, séries de Fourier,

funções splines, etc.

O segundo grupo é o que trabalha com uma rede de pontos, representando o terreno através de valores discretos em pontos que podem estar distribuídos de forma ordenada ou de forma aleatória. Utilizam-se com maior freqüência as redes quadradas, criadas artificialmente, e as redes triangulares apoiadas nos pontos fornecidos. Alguns desses modelos utilizam, num segundo estágio, uma função interpoladora, válida em cada quadrado ou triângulo.

Existem diversos trabalhos que estudam os diferentes modelos digitais. Podem ser consultados com proveito, três deles que dão uma visão geral de todos os métodos: Schumaker (1976), Barnhill (1977) e Sabin (1980).

A seguir apresentamos uma breve descrição de alguns desses métodos.

2.2. MODELOS QUE UTILIZAM SUPERFÍCIES (EQUAÇÕES ANALÍTICAS)

A essência destes métodos consiste em aproximar a superfície do terreno por uma função contínua $z = z(x,y)$, que procura representar o melhor possível a superfície em questão.

Teoricamente, seria possível adaptar uma equação analítica que passasse por todos os pontos fornecidos, por exemplo, um polinômio de grau adequado. No entanto, na maioria dos casos essa equação teria um grau muito elevado e seria de difícil manuseio, mesmo com a utilização de computadores. Além dis-

so, a prática mostra que nesses casos a superfície, nas áreas entre os dados, fica comprimida e apresenta uma configuração completamente insatisfatória quando comparada com o aspecto real.

Computacionalmente é necessário diminuir o grau do polinômio para reduzir o tempo de processamento, sem que no entanto isso prejudique a representação da superfície. Escolhem-se, portanto, formas matemáticas que, sem serem demasiadamente complexas, se adaptem relativamente bem aos pontos fornecidos.

Os valores da função nos pontos conhecidos não serão exatamente respeitados: substituindo-se as coordenadas de cada ponto nas equações matemáticas encontra-se um valor de z ligeiramente diferente. Comparando-se esse valor com o real observa-se um desvio. Através de técnicas de mínimos quadrados consegue-se determinar os coeficientes da equação que melhor representa o terreno, isto é, que torna mínima a soma dos desvios quadráticos.

As superfícies matemáticas, principalmente se forem compactas, possibilitam o cálculo da cota de qualquer ponto,bastando para isso introduzir as coordenadas desse ponto na expressão matemática representativa da superfície: dispensa-se qualquer tipo de interpolação. A determinação das curvas de nível pode ser feita facilmente empregando um método de dois estágios. Em primeiro lugar determina-se a cota nos nós de uma malha quadrada e a seguir utiliza-se uma rotina para a interpolação das curvas de nível.

O cálculo de perfis e secções transversais também se faz

de modo semelhante, bastando calcular o valor da função em pontos discretos sobre as linhas desejadas.

Pensando matematicamente, seria fácil obter curvas de nível através de um processo de intersecção de superfícies: a linha de encontro da superfície do terreno (equação matemática) com planos horizontais sucessivos dá origem às curvas de nível; a intersecção com planos verticais dá origem a secções transversais ou a perfis.

Na prática esse processo torna-se pouco viável, devido a complexidade matemática da superfície global. Normalmente ela é de um grau relativamente alto e bivariada em x e y e, portanto, não se consegue uma representação explícita em função de uma das variáveis. Isso implica em um processo iterativo para determinar as linhas de cota inteira, e portanto, grande tempo de computação.

Há sempre um comprometimento no grau do polinômio utilizado: quando se quer uma alta fidelidade o grau é muito elevado, de difícil computação e com tendência a formar "enrugamentos" irreais nas regiões entre os pontos amostrados; quando o grau do polinômio é pequeno, ele não representa com suficiente detalhe a superfície em questão. Os próprios valores fornecidos não são totalmente respeitados, resultando um desvio algumas vezes inaceitável. Além disso há a tendência ao arredondamento e suavização das formas, distorcendo o aspecto real. Esse efeito pode ser apreciado na Figura 2.1, que corresponde ao seguinte polinômio:

$$z = c_1 + c_2x + c_3y + c_4x^2 + c_5y^2 + c_6xy + c_7x^3 + c_8y^3 + c_9x^2y + c_{10}xy^2$$

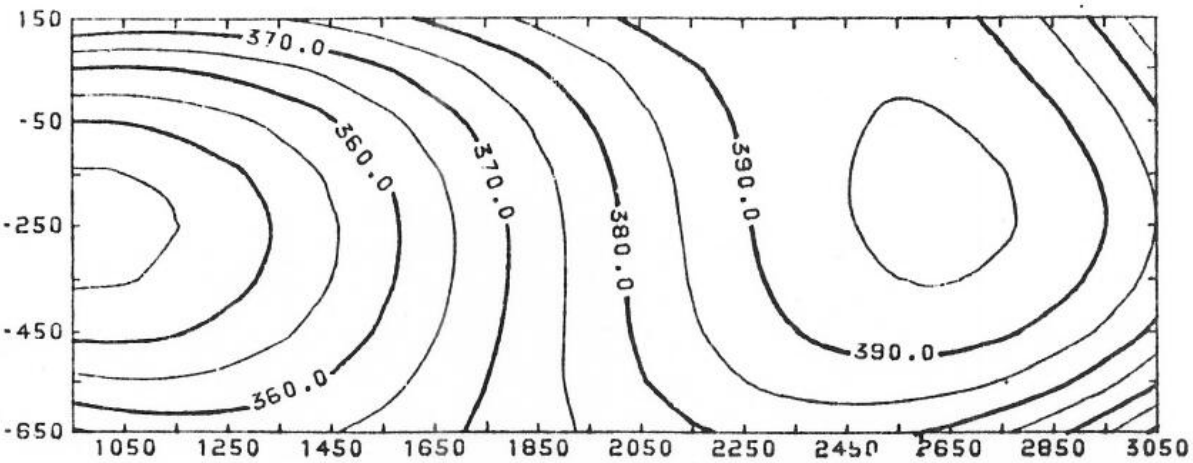


Figura 2.1 - Efeito de arredondamento e suavização das formas em um mapa topográfico, utilizando um polinômio de terceiro grau como superfície de tendência.

Saliente-se mais uma vez que quando o grau do polinômio é muito grande, o volume de cálculos(e o tempo de computação) é considerável, já que costuma crescer proporcionalmente ao quadrado do número de pontos amostrados. Por outro lado, a precisão do processo tende a decair, já que a estimativa através dos mínimos quadrados exige a solução de um sistema de equações de ordem elevada, com inversão de matrizes e outros cálculos que acarretam perda de precisão.

Esse tipo de método costuma ser encontrado sob o título "Trend surface analysis" ou "Least square surface". Em português utilizam-se expressões como "superfícies de tendência" ou "superfícies de mínimo quadrado". Pode-se consultar por exemplo: Davis (1973), Diniz (1982), IBM (1965) e Davis e Mc Culla [(1978

2.2.1. SUPERFÍCIES GLOBAIS

Quando possível, a representação de superfícies através

de uma expressão matemática é claramente desejável. A preferência tem recaído sobre a utilização de um polinômio, uma série de Fourier ou uma função spline; superfícies que passamos a analisar.

a) Polinômios

O teorema da aproximação de Wierstrass estabelece que qualquer superfície contínua pode ser aproximada com um pequeno erro arbitrário ε, através de um polinômio de grau suficientemente alto. Esse polinômio, somado com o erro ε, representa exatamente a cota em cada ponto. Poderia assumir, por exemplo, uma expressão do seguinte tipo:

$$z = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^3 + \ldots + \varepsilon.$$

Os coeficientes do polinômio em questão devem ser determinados através das técnicas de mínimos quadrados. O grau do polinômio pode ser escolhido de forma a manter o valor de ε abaixo de um máximo pré-fixado para cada caso.

Tendo em conta que qualquer levantamento trabalha com um número finito de pontos, seria possível escolher um polinômio de grau tal que a superfície passasse através de todos os pontos, tornando nulos os desvios. Como já foi apontado, esse procedimento torna-se impraticável computacionalmente, e impõe-se necessariamente a escolha de um polinômio de grau inferior.

b) Séries de Fourier

Outro modo de representar superfícies nasceu por inspi-

ração da própria ondulação do terreno: séries de senos e cossenos, desenvolvidas por Fourier e que podem assumir uma formulação como a que apresentamos a seguir para representar a altitude z do terreno.

$$z = \sum_{i=0}^{KC} \sum_{j=0}^{LC} cc_{ij} \cos\left(\frac{2\pi iy}{M}\right) \cos\left(\frac{2\pi jx}{N}\right) +$$

$$+ \sum_{i=0}^{KC} \sum_{j=0}^{LS} cs_{ij} \cos\left(\frac{2\pi iy}{M}\right) \operatorname{sen}\left(\frac{2\pi jx}{N}\right) +$$

$$+ \sum_{i=0}^{KS} \sum_{j=0}^{LC} sc_{ij} \operatorname{sen}\left(\frac{2\pi iy}{M}\right) \cos\left(\frac{2\pi jx}{N}\right) +$$

$$+ \sum_{i=0}^{KS} \sum_{j=0}^{LS} ss_{ij} \operatorname{sen}\left(\frac{2\pi iy}{M}\right) \operatorname{sen}\left(\frac{2\pi jx}{N}\right)$$

onde:

M = comprimento de onda fundamental na direção y;

N = comprimento de onda fundamental na direção x;

KC = máximo cosseno harmônico na direção y;

LC = máximo cosseno harmônico na direção x;

KS = mínimo seno harmônico na direção y;

LS = mínimo seno harmônico na direção x;

$$\left.\begin{array}{l} cc_{ij} \\ cs_{ij} \\ sc_{ij} \\ ss_{ij} \end{array}\right\} = \text{coeficientes da série.}$$

A acuidade de representação está em função do número de termos escolhidos que, como no caso dos polinômios, não pode ser muito elevado por limitações computacionais.

O emprego de uma série pode levar a resultados mais precisos que um polinômio; no entanto, a série de Fourier mostra-se bastante sensível com relação à escolha da direção dos eixos x e y. Uma inversão parcial ou rotação de eixos leva a resultados sensivelmente diferentes.

Como ilustração, apresenta-se nas Figuras 2.2, 2.3, 2.4 e 2.5 o aspecto dessas equações utilizadas para representar o terreno.

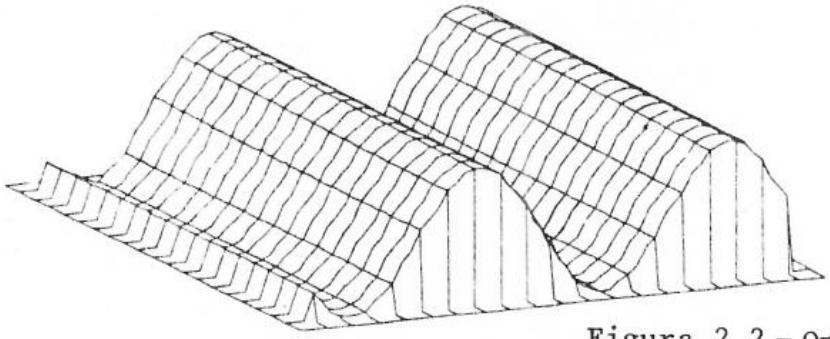


Figura 2.2 - Onda bidimensional de senos com um harmônico simples na direção x.

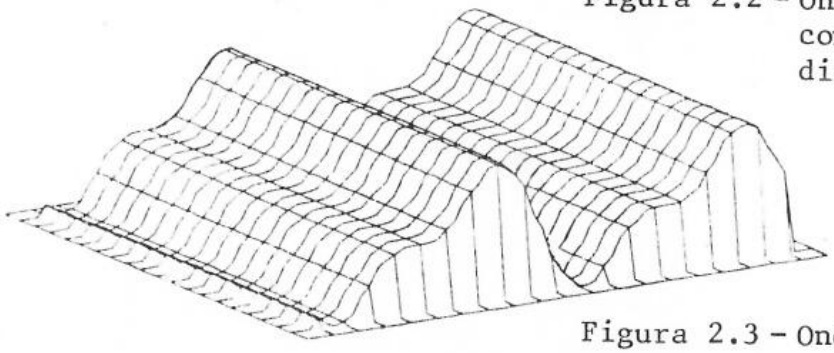


Figura 2.3 - Onda bidimensional de senos com dois harmônicos na direção x.

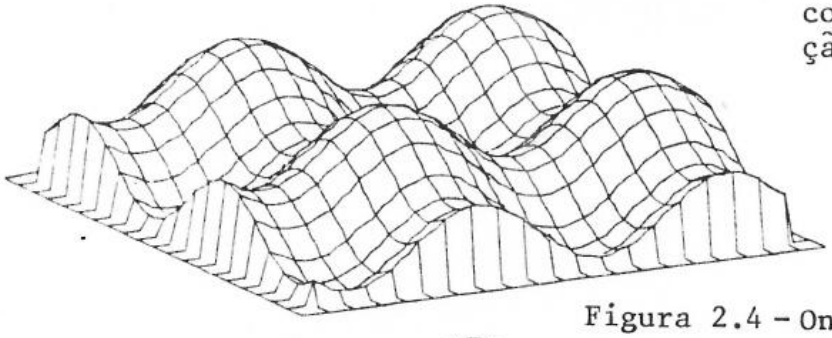


Figura 2.4 - Onda bidimensional de senos com um harmônico simples na direção x e outro na direção y.

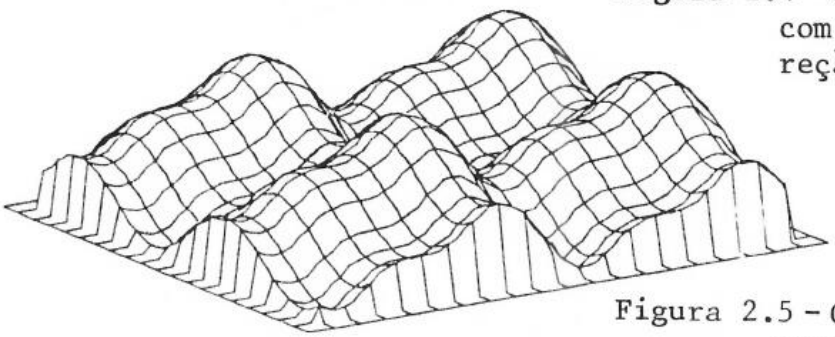


Figura 2.5 - Onda bidimensional de senos com dois harmônicos em cada uma das direções.

c) Funções Splines

As funções splines ainda que sejam um tipo especial de polinômio, merecem destaque por sua importância. Trata-se de um desenvolvimento recente da matemática (Schoenberg, 1946), com inúmeras aplicações: aproximação de funções, interpolação numérica, solução numérica de equações diferenciais, ajustamento de curvas e superfícies, etc.

A inspiração surgiu da necessidade de desenvolver a equação de uma curva suave passando por determinados pontos. Os desenhistas já utilizavam desde há muito, aparelhos de madeira ou plástico, como a Curva Francesa, para traçar uma linha sem angulosidades, entre pontos especificados. Também se utiliza a conhecida entre nós como Curva Japonesa, constituída por uma barra alongada de material elástico, flexível e maleável, que pode ser deformada para adquirir o contorno da curva a ser traçada. Por extensão, o nome dessa barra, em inglês "stripe" ou "spline", foi aplicado por Schoenberg às funções matemáticas por ele propostas como modelo analítico para a obtenção de de elementos suaves e uniformes.

Uma função spline de grau k, adaptada a n pontos, é constituída em cada intervalo entre dois pontos por arcos polinomiais de grau menor ou igual a k, e tendo as k-1 primeiras derivadas contínuas. Em função das imposições adicionais, em cada caso, a família de curvas splines ganha nomes particulares: spline natural, spline cúbica, spline bi-cúbica, B-spline, etc. Para cada família e em cada aplicação concreta, existe a curva que é a mais suave e são conhecidos os algoritmos para determiná-la.

Existe farta literatura sobre as funções splines e suas aplicações. Por exemplo, Ahlberg (1967), Birkhoff (1969), Boor (1962), Boor (1972), Cline (1974), Duris (1980), Hall (1973), Hall (1969), Haussmann (1973), Munteanu (1973), Munteanu e Schumaker (1974), Whitten e Koelling (1975).

Entre essas aplicações interessa-nos a representação de superfícies, em que se deve utilizar um spline bi-dimensional.

2.2.2. VÁRIAS SUPERFÍCIES LOCAIS

Uma maneira simples de reduzir os inconvenientes gerados pelos polinômios é limitar a região de sua validade e empregar diversos polinômios de grau limitado. Para isso divide-se o terreno em áreas menores, com menos pontos em cada uma.

Para uma região suficientemente pequena, até mesmo um polinômio linear, um plano, pode ser uma boa aproximação. Trata-se pois de encontar um ponto de equilíbrio entre a divisão excessiva da superfície e a complexidade da expressão matemática.

O principal problema que surge neste novo enfoque é resolver a descontinuidade na linha de fronteira entre dois domínios quaisquer: cada região deve ser compatibilizada com suas vizinhas.

Existem métodos matemáticos para forçar a continuidade das linhas, utilizando por exemplo superfícies intermediárias de concordância. No entanto, a maior parte deles requer um sistema complexo de equações diferenciais parciais, anulando

assim as vantagens de utilizar várias superfícies locais para diminuir o volume de cálculo.

Um método engenhoso e ilustrativo do processo foi desenvolvido por Jancaitis & Junkins (1973), professores da Universidade de Virgínia: cria-se uma região intermediária de superposição das duas funções na qual o valor adotado é uma média ponderada das duas funções iniciais. O coeficiente de ponderação de cada uma varia, na região de superposição, de 1 até 0 para uma das funções e de 0 a 1 para a outra, de modo a fazer que as influências sejam proporcionais às distâncias (a mais próxima influi mais). Na região das bordas a influência de uma função é total e a influência da outra é nula.

Essas funções podem ser polinomiais, de qualquer grau; as funções de peso também podem ser variáveis.

A função de peso linear garante a continuidade da função mas acarreta a criação de angulosidades nas interfaces.

Para eliminar esse efeito e obter a suavidade pode-se impor que a primeira derivada das funções de peso sejam nulas nessas interfaces, o que leva a adotar uma função do terceiro grau para representá-las.

No caso de funções bidimensionais, domínios quadrados, o método pode ser o seguinte. Cada região situa-se numa localização central e possui 4 regiões de fronteira; a função aproximativa será composta pela soma ponderada de quatro superfícies preliminares, que possuem em comum o quadrado em análise.

Para obter a função estimativa z multiplica-se cada superfície preliminar pela correspondente função de peso e somam-se as quatro parcelas. A superfície final é válida somente para a região central. Para cobrir toda a região o processo deve ser repetido adequadamente.

O U.S. Army desenvolveu o programa UNAMACE, de acordo com essa formulação: trabalha com superfícies parciais válidas localmente.

Em um exemplo representativo, para gerar as aproximações preliminares,utilizou-se uma superfície do primeiro grau (plano) ajustada pelo método dos mínimos quadrados em cada intervalo de validade. As funções de peso adotadas foram do terceiro grau, resultando uma superfície do quarto grau, que em cada intervalo assume a forma:

$$z = \sum_{i=0}^{4} \sum_{j=0}^{4} c_{ij} x^i y^j ,$$

onde estão envolvidos 25 coeficientes c_{ij}.

2.3. MODELOS QUE UTILIZAM UMA REDE DE PONTOS

Passamos a estudar outro grande grupo de modelos: os que trabalham com uma malha de pontos, que pode ser triangular, retangular, hexagonal ou quadrada. Estudaremos com mais atenção o caso das malhas quadradas e triangulares,que são as mais utilizadas.

2.3.1. MALHA QUADRADA

A malha quadrada por ser regular e ordenada apresenta u-

ma série de vantagens. A ordenação dos pontos melhora a sua operacionalidade, isto é, o armazenamento dos pontos se faz com menos memória pois utilizando-se uma matriz para gravar a cota z de cada ponto, dispensa-se o armazenamento das coordenadas x e y que ficam implícitas na posição i, j do elemento da matriz.

O cálculo de distâncias entre dois nós pode ser feito em função da diferença de linhas e de colunas entre os elementos, que multiplicada pela dimensão linear da célula da malha, fornece o resultado procurado.

A listagem (ou desenho) de perfis ou secções transversais também se torna mais simples, bastando para isso imprimir (ou desenhar) uma linha ou coluna da matriz z.

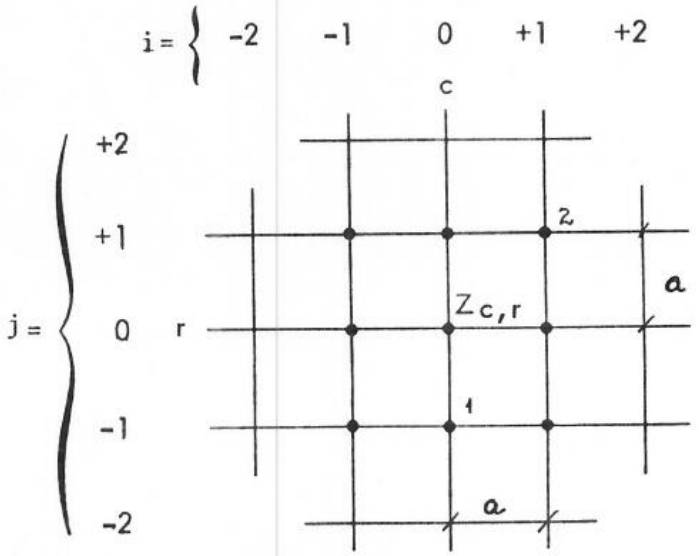


Figura 2.6 - Rede quadrada representativa do terreno.

$$\text{distância } 12 = a \cdot \sqrt{(\text{dif.de linhas})^2 + (\text{dif.de colunas})^2} =$$

$$= a \cdot \sqrt{(2)^2 + (1)^2} = a . \sqrt{5} = \ldots$$

Na Figura 2.6 mostra-se um esquema da matriz z de pontos e indica-se o modo de calcular distâncias a partir dela.

Na Figura 2.7 mostra-se a perspectiva de um diagrama de blocos, constituído pelo lançamento das cotas em cada nó da malha.

Este bloco perspectivo indica um modo fácil de calcular volumes no terreno: através da média ponderada das alturas de uma superfície prismática. Cada cota vertical terá pesos:1,3/4 1/2 ou 1/4, em função do número de arestas que concorrem no nó correspondente.

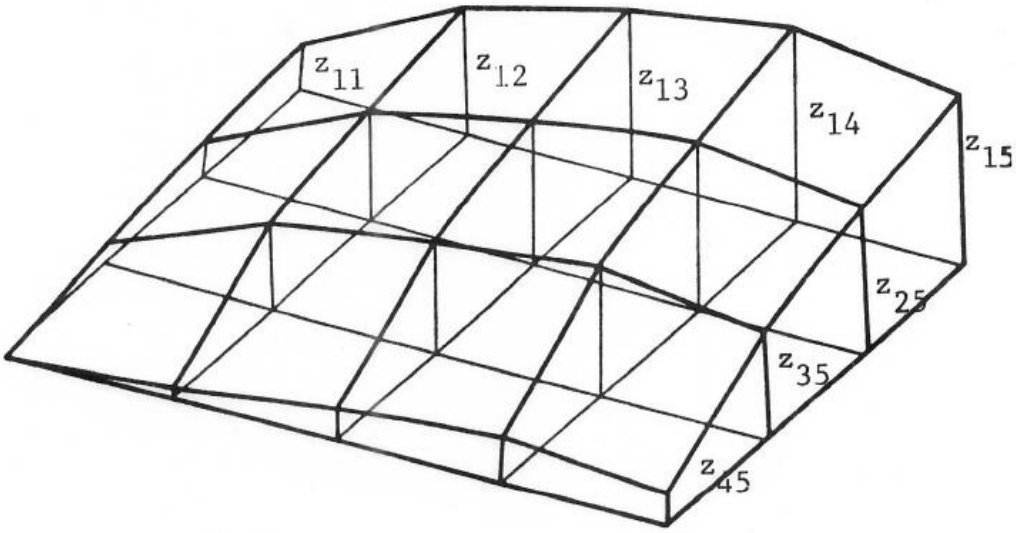


Figura 2.7 - Diagrama de blocos perspectivo construído pela alocação vertical da cota de cada ponto da malha regular.

Algumas vezes os pontos podem ser fornecidos de forma regular através de um levantamento coincidente com a malha, que fica assim automaticamente definida. No entanto, na maior parte dos casos a distribuição dos pontos é aleatória e necessitamos definir as dimensões da malha, o espaçamento das linhas Uma malha muito aberta não tem sentido já que não serviria pa-

ra a função para a que foi planejada: densificar a rede inicial ou substituí-la. Uma malha muito fina, pode provocar problemas de armazenamento ou, conforme o processo, gerar cotas "falsas", distorcendo o modelo representativo do terreno.

O Programa Surface II calcula uma distância média d_m, em função da qual a dimensão a da malha será correlacionada. Para isso determina-se um retângulo que envolve todos os pontos tendo como base b a diferença das coordenadas máximas e mínimas em x, e como altura h a diferença das coordenadas máximas e mínimas em y. Divide-se essa área pelo número n de pontos e extrai-se a raiz quadrada:

$$d_m = \sqrt{\frac{(x_{max}-x_{min})\times(y_{max}-y_{min})}{n}}$$

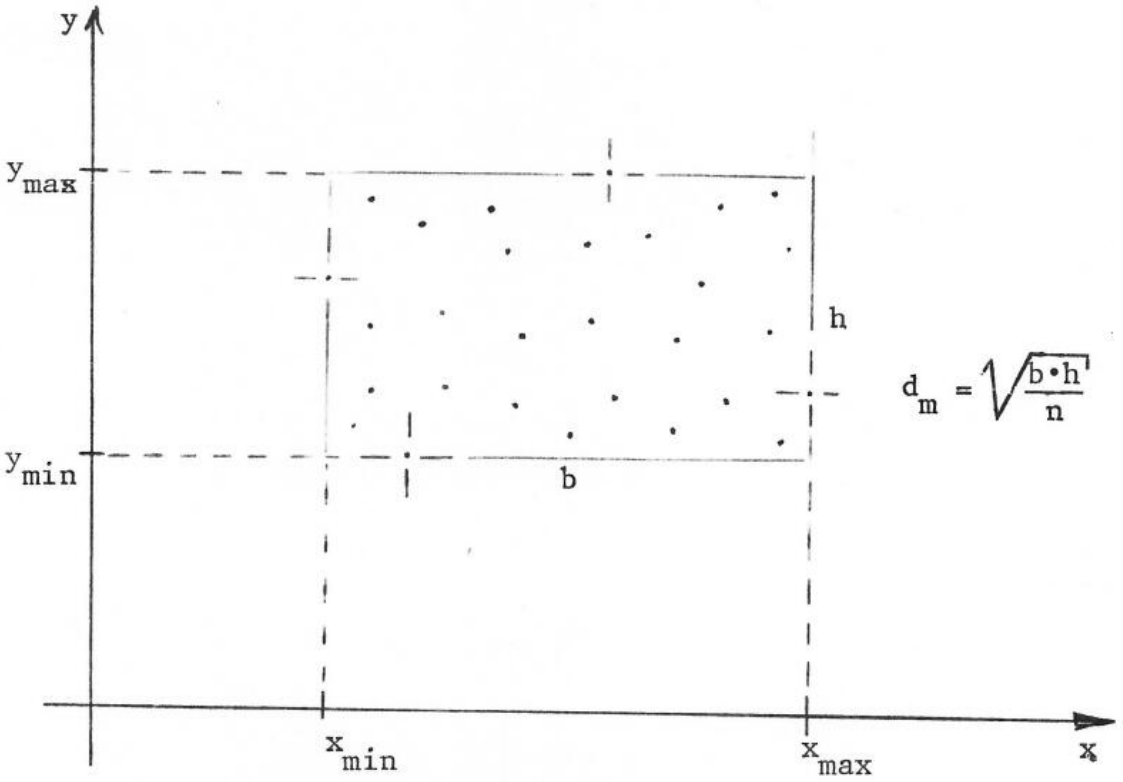


Figura 28 - Cálculo de uma "distância média" entre os pontos de uma amostra.

O significado "físico" dessa distância média é representado pelo lado de um quadrado que corresponde a cada ponto se dividirmos a área total pelo número de pontos. De alguma forma está relacionada com a densidade de pontos na região.

Os usuários do Surface II recomendam, para a dimensão a da malha, um valor que está entre $d_m/2$ e $d_m/4$; valores ditados pela prática; veja-se por exemplo Diniz (1982).

A seguir deve-se gerar as cotas dos nós da rede. Encontramos basicamente três modos de fazer isso: por ajuste de uma superfície analítica, por ajuste de um polinômio, ou por uma aproximação numérica.

O ajuste por superfície analítica utiliza um dos métodos já vistos, por exemplo, um polinômio ou uma série de Fourier, que ajustados aos pontos deixa um resíduo em cada um deles. A minimização desses resíduos (desvios) produz a superfície ótima ou de mínimo quadrado. Com essa equação calcula-se a cota dos nós da rede.

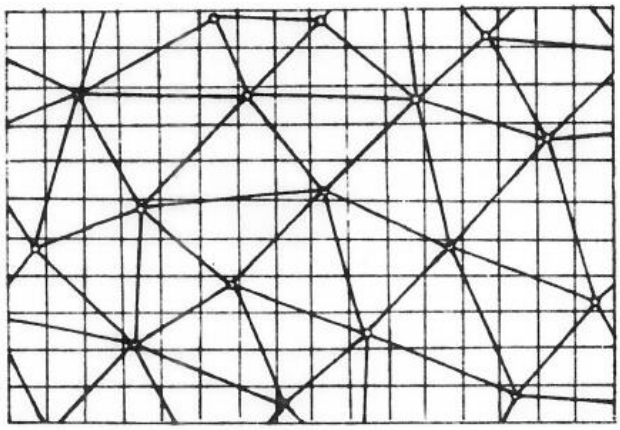


Figura 2.9 - Construção de uma rede triangular para estimar as cotas de uma malha quadrada, através de interpolações lineares.

Um segundo modo de gerar a rede consiste em aproximar a superfície do terreno por um poliedro de faces triangulares cujos vértices são representados pelos pontos levantados. Nenhum ponto levantado deve cair dentro de algum triângulo. Ilustra-se essa situação com a Figura 2.9.

A intersecção entre as faces triangulares planas e as verticais que passam por cada nó da malha, determina a cota de cada ponto. Isso supõe que a interpolação linear é uma boa aproximação, o que em certos casos não costumam estar longe da verdade.

O passo mais difícil, no entanto, é a formação da malha triangular através do computador. Torna-se necessário estabelecer alguns critérios que permitam decidir sobre a conveniência de unir ou não, 2 pontos quaisquer.

Os algoritmos para formar essa triangulação são muitos, mas nem todos fornecem eficiência computacional.Por outro lado, o método da superfície poliédrica pode ser utilizado em si

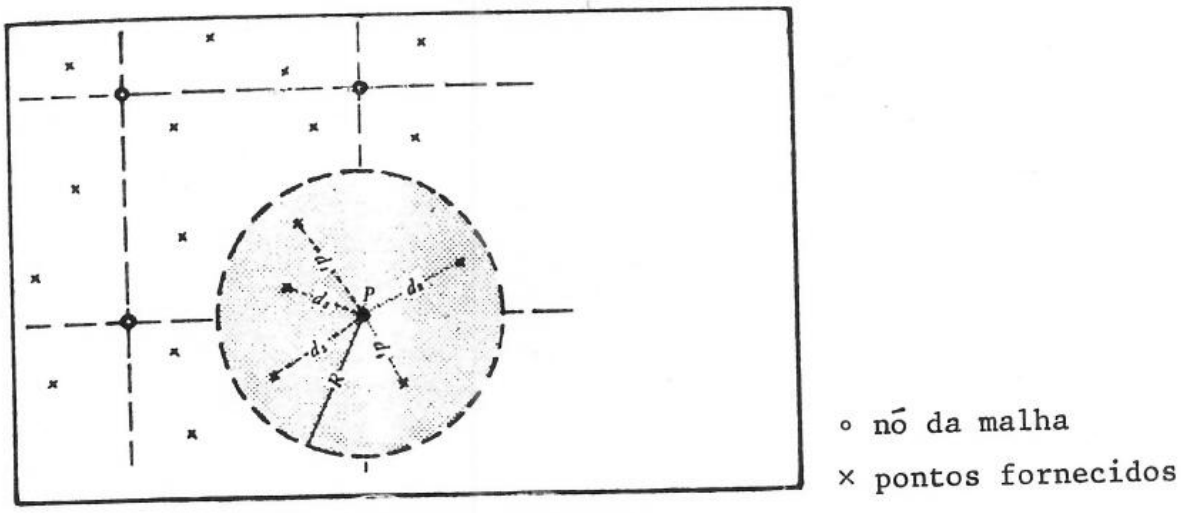


Figura 2.10 - Definição dos pontos vizinhos de P através de um círculo de raio R.

mesmo, como processo independente. Ao estudar o método de Akima, no corpo deste trabalho, desenvolvemos os estudos relativos à triangulação.

O terceiro processo consiste em definir uma vizinhança para o nó da malha e calcular a cota deste ponto através de uma média ponderada das cotas dos vizinhos.

Existem vários critérios para definir esta região de vizinhança. O primeiro deles estipula um determinado raio R em função da densidade de pontos e da conformação do terreno e considera todos os pontos que caem dentro do círculo assim definido (Figura 2.10).

Um segundo critério impõe a existência de um mínimo de pontos (5 ou 6) dentro do círculo, e amplia o raio R para que se obtenha esse número.

Um terceiro critério pesquisa a distribuição em quadrantes ou octantes e exige um ou dois pontos por setor, visando uma melhor configuração espacial para os vizinhos.

Outros critérios podem ser também definidos, baseados por exemplo em estudos estatísticos sobre a distribuição espacial dos pontos. É o que faz, por exemplo um método proposto por D. G. Krige (1966) e que ficou conhecido como "Krigagem".

Uma vez definidos os vizinhos pode-se então adotar uma função de ponderação que permita calcular a cota do nó como uma média ponderada.

$$z_p = \frac{\sum_{i=1}^{n} \mu_i \cdot z_i}{\sum_{i=1}^{n} z_i}$$

- pontos vizinhos: $1, 2, \ldots, n$
- cotas: $z_1, z_2, \ldots, z_n$
- funções de ponderação: $\mu_1, \mu_2, \ldots, \mu_n$.

A função de ponderação deve garantir que quanto mais perto do nó está um ponto, maior é o seu peso. Foram propostas várias fórmulas: o inverso da distância, o inverso da distância ao quadrado, ao cubo, ou outras formas mais complicadas.

Uma correta ponderação deve levar em conta não só as distâncias mas também a distribuição espacial dos pontos. Um exemplo de como as ponderações podem ser corretas ou incorretas vem mostrado na Figura 2.11.

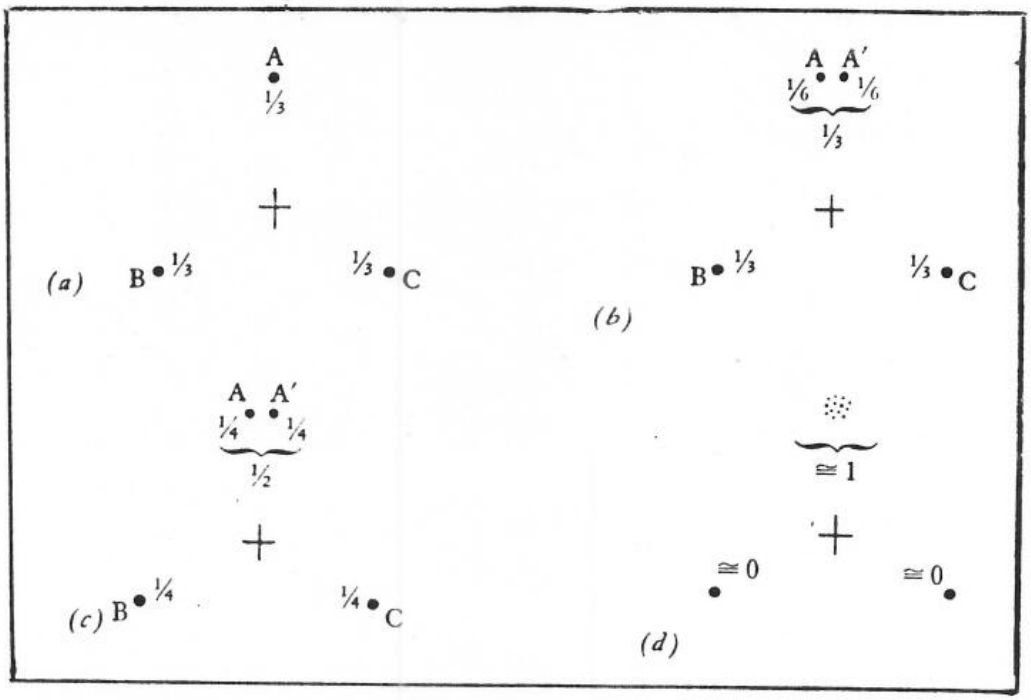


Figura 2.11 - Diferentes pesos atribuídos aos pontos amostrados, para o cálculo do valor da cota dos pontos assinalados com +. Os esquemas (a) e (b) definem pesos corretos, enquanto que (c) e (d) atribuem pesos insatisfatórios pois não consideram o arranjo espacial dos pontos.

A etapa final consiste em desenhar as curvas através da

rede de pontos. Isto será visto com detalhe ao analisarmos os algoritmos de desenho.

Para o estudo geral das malhas quadradas podem ser úteis os seguintes trabalhos: Akima (1974), Akin & Gray (1977), Akin & Gray (1979), Calcomp (1972), Cottafava & Le Moli (1969), Dayhoff (1963), Dynamic Graphics (1978), Sampson (sem data), Sibson & Thomson (1981), Snyder (1978).

2.3.2. REDE TRIANGULAR

Este método é o que corresponde mais fielmente ao esquema seguindo quando o desenho é feito manualmente. Em primeiro lugar unimos convenientemente os pontos de maneira a formar uma triangulação aceitável. A seguir realiza-se a interpolação em cada triângulo para, finalmente, desenhar as curvas de nível.

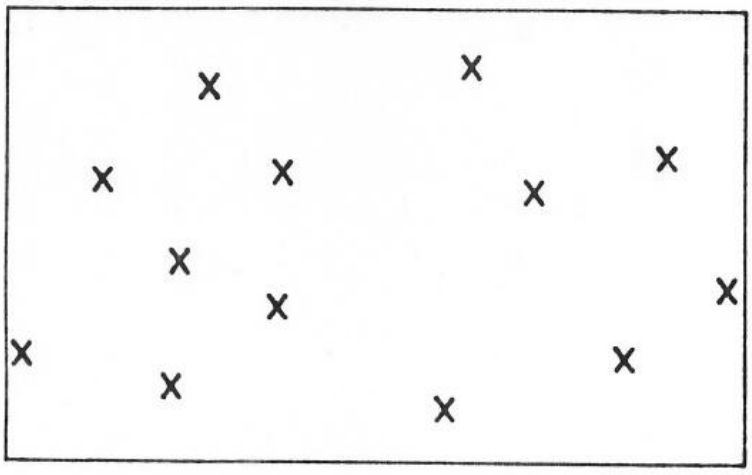


Figura 2.12 - Pontos fornecidos

Nas Figuras 2.12 a 2.14 encontra-se a representação dos passos necessários.

Na primeira etapa os pontos são fornecidos através de suas

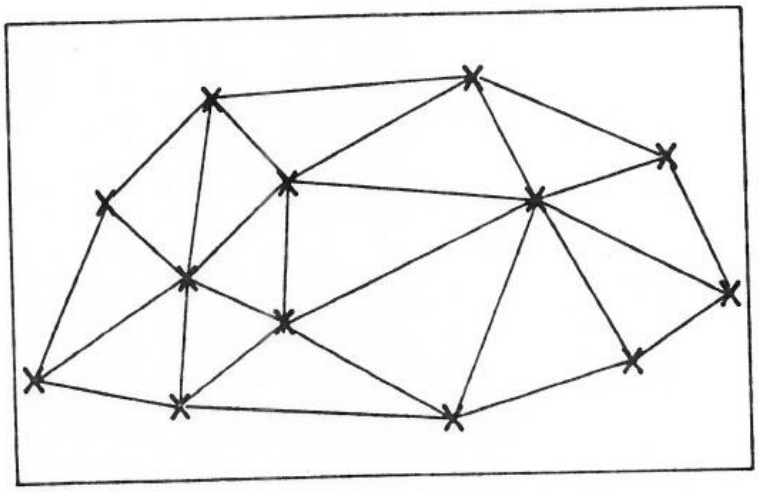


Figura 2.13 - Triangulação aceitável

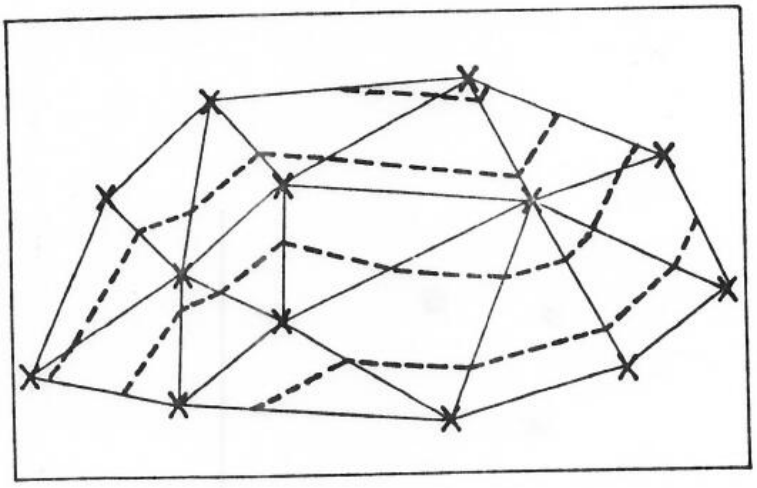


Figura 2.14 - Interpolação das curvas de nível

coordenadas e encontram-se distribuídos de forma aleatória sobre o plano x, y. A seguir unem-se adequadamente os pontos para formar a malha triangular. Impõe-se que não existam linhas que se cruzem nem pontos isolados que não pertençam a nenhum triângulo.

Essas condições, mesmo para idênticos pontos, não conduzem a uma triangulação única. Existem diversas opções e deve-se preferir aquelas que apresentem triângulos bem conformados, isto é, evitando lados muito compridos e ângulos mui-

to agudos. Mostra-se a seguir na Figura 2.15, duas opções de triangulação com os mesmos pontos.

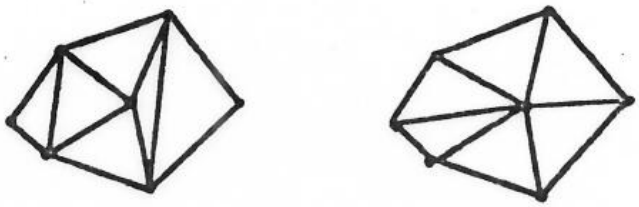


Figura 2.15 - Duas opções de triangulação para os mesmos pontos.

A elaboração de algoritmos para dividir uma região em triângulos aceitáveis com vértices nos pontos fornecidos não é tão fácil como parece à primeira vista. Os primeiros algoritmos, de que temos notícia, surgiram nos anos 70 e nem todos fornecem boas triangulações. Como exemplo ilustrativo, o proposto por Düppe e Gottschalk (1970), é o seguinte: todas as possíveis linhas unindo os pontos dados são calculadas e ordenadas de acordo com o seu comprimento. Toma-se o segmento mais curto como ponto de partida e eliminam-se todos os demais que cruzam com este. A seguir toma-se a linha mais curta dentre as que possuem um vértice em comum com o primeiro segmento: está formado o primeiro triângulo. O processo é repetido até que todos os triângulos estejam formados. Pode-se demonstrar que a soma dos lados encontrados por esse processo é mínima.

No entanto, tendo em conta o grande número de pontos encontrados na prática, este método torna-se dispendioso e in-

viável, por ter um tempo de processamento muito elevado.

Com a malha triangular pronta, devemos estabelecer um procedimento para realizar a interpolação em cada triângulo que possibilitará, entre outras coisas, o cálculo e desenho das curvas de nível.

O procedimento mais simples consiste em definir um plano passando pelos três vértices do triângulo. A cota z assume a forma $z = a+b \cdot x+c \cdot y$, onde os coeficientes podem ser facilmente determinados resolvendo um sistema de 3 equações a 3 incógnitas. Esse método produz uma superfície poliédrica de face triangulares para representar o terreno.

Fica garantida a continuidade global da representação em todas as regiões triangulares e na área toda, já que as interfaces são linhas retas, comuns a triângulos vizinhos. No entanto, haverá a formação de arestas vivas, precisamente nos lados de cada triângulo.

Para obter uma suavização de toda a superfície pode-se adaptar uma equação matemática para cada triângulo, impondo-se por exemplo a continuidade da primeira derivada ao longo do lado, na direção perpendicular. É necessário portanto abandonar a interpolação linear e adotar uma concepção mais ampla: uma função interpoladora que pode ser um polinômio bivariado em x e y, uma função lagrangeana, uma função racional, etc.

Tem sido sido muito explorada também a analogia com a definição de elementos de concordância suave na teoria dos elementos finitos. Demonstrou-se que para construir um elemento

com suporte nos triângulos adjacentes a um determinado vértice e com continuidade global da primeira derivada, é necessário um polinômio de 5º grau, no mínimo.

Nesses métodos é necessário avaliar as derivadas da superfície matemática nos vértices do triângulo e às vezes ao longo dos lados. Isso é feito através dos vizinhos, definidos de acordo com certos critérios, por exemplo, todos os vértices das arestas que concorrem no ponto em questão.

Quando se utilizam funções interpoladoras, o método da triangulação pode ser entendido como uma composição de superfícies válidas localmente em cada triângulo. A diferença com relação aos métodos de superfície localmente válidas, visto anteriormente, é que neste caso os domínios não são definidos arbitrariamente mas seguem a divisão natural dos triângulos. A continuidade não é conseguida por outras funções auxiliares mas pelas próprias condições impostas na construção de cada interpoladora.

O traçado das curvas de nível através de uma rede triangular mostra-se mais preciso e definido do que aquele que se faz através das redes quadradas ou retangulares. Não existe a condição de ambiguidade, que surge numa malha quadrada quando existem quatro intersecções de uma mesma curva com uma célula. Uma vez que determinada curva entra numa célula triangular, só existe uma única possibilidade de saída, ou seja, ou uma curva de nível não intercepta o triângulo ou cruza exatamente 2 lados.

Voltaremos com detalhe sobre o assunto ao tratarmos do

desenho das curvas de nível.

A bibliografia sobre os métodos de triangulação também é citada mais adiante, já que o método de Akima, em estudo, pertence a esta categoria.

2.4. OUTROS MÉTODOS

Existem muitas outras possibilidades e variantes para representar o terreno. Por exemplo, combinando dois modelos. Dessa forma, existem sistemas que realizam primeiramente uma aproximação sobre uma malha regular para depois ajustar uma superfície de mínimos quadrados através dos novos pontos.

Outras vezes os métodos ganham nomes especiais mas no fundo podem reduzir-se a uma variante dos modelos já apresentados. Por exemplo, a krigagem, processo idealizado por D. G. Krige (1966), é denominado como um método geoestatístico. Na realidade, em sua versão original, trabalha com uma malha quadrada, apresentando especiais cuidados na avaliação da cota dos pontos situados nos nós da malha. Para isso, através de processos estatísticos, determina a correlação espacial entre o valor da função, em um ponto e em seus vizinhos, construindo uma função de correlação (variograma).

Outro método é apresentado por Shepard (1968), que pode ser utilizado local ou globalmente. Estabelece uma média ponderada de funções analíticas (superfícies matemáticas) para efetuar o cálculo da coordenada de um ponto qualquer da região em estudo. Pode ser visto como um método de superfícies de mí-

nimos quadrados em que se atribuem diferentes pesos às funções, em proporção à distância ao ponto em que está sendo feita a avaliação.

O próprio Shepard admitiu algumas falhas e pontos fracos que pensava corrigir mais adiante: o crescimento exponencial do volume de cálculo com o número de pontos e, principalmente, o "achatamento" da superfície interpolada na vizinhança dos pontos fornecidos. Diversas variantes de seu método foram propostas por outros autores.

Outros métodos que inspiraram variantes são os de Mac Lain (1974) e Maude (1973). De qualquer forma, seria praticamente impossível fazer um estudo detalhado de todos os diversos métodos que foram surgindo nos últimos anos. Como exemplo da quantidade de idéias já exploradas computacionalmente, o trabalho de Franke (1982) divide os métodos em 6 grandes categorias e realiza uma comparação entre 32 programas.

2.5. AVALIAÇÃO

Embora já se tenham feito alguns comentários sobre cada método no lugar correspondente, parece conveniente apresentar algumas considerações que dirigiram este estudo para os métodos que utilizam redes triangulares.

Os métodos que utilizam superfícies matemáticas não respeitam exatamente os pontos fornecidos, apresentando um certo resíduo, o que, em muitos casos, é um incoveniente grande. Por outro lado, o tempo de processamento também costuma ser um pon-

to fraco deste tipo de método.

Os métodos que utilizam uma rede ordenada de pontos, embora sejam práticos para o armazenamento e trabalhabilidade dos dados, introduzem um passo a mais na solução do problema que, de acordo com o sistema empregado, pode ser crítico: a avaliação do valor da função nos nós da malha. Isso pode significar o desrespeito aos pontos fornecidos, a perda de precisão ou a criação de valores irreais. Analisando o Surface II, que utiliza esse processo, notamos que a mecânica interna leva à criação de valores fictícios em zonas de poucos dados e que as curvas de nível não são satisfatórias: apresentam angulosidades acentuadas e freqüentemente se cruzam, coisa inaceitável.

Os métodos que utilizam a triangulação são os que adotam um procedimento mais próximo ao método manual, definem estritamente a região a ser mapeada, não apresentam situações ambíguas na definição das curvas de nível e vários dos sistemas que utilizam essa solução apresentam curvas suaves e desenhos bons ou aceitáveis.

Tendo em conta essas vantagens preliminares, escolhemos um algoritmo dessa classe como ponto de partida desta tese.

No final deste trabalho apresentamos um esboço de metodologia prática para um estudo comparativo de diversos métodos.

2.6. SISTEMAS E PROGRAMAS EXISTENTES

Existem muitos e variados programas e sistemas utiliza-

dos na criação automática de mapas. Poderíamos citar alguns: Surface II, IBM, Gradis-Contraves, Topsys, SACM, CALCOMP , SPEC, DTM, todos eles disponíveis no mercado brasileiro ou utilizados por firmas nacionais.

Em geral, a palavra sistema (ou alternativamente programa) é utilizada para designar um instrumento computacional completamente desenvolvido e testado para representar o terreno e desenhar as curvas de nível, estando disponível no mercado para compra ou utilização dos usuários.

Todos eles contêm embutido um programa computacional para realizar essas operações e ao qual, na quase totalidade dos casos, os usuários não têm acesso, sendo por isso conhecidos como "pacotes fechados". Excetua-se o Surface II, um dos sistemas mais antigos de que se tem notícia e o método desenvolvido por Akima: "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data point", disponível na coleção de algoritmos e programas da ACM, Association for Computing Machinery. O fato de estar disponível foi também um fator decisivo na opção por este programa.

A palavra método ou modelo é utilizada para designar o modo (modelo matemático) que cada sistema emprega basicamente para representar o terreno: equação analítica, malha ordenada de pontos, rede triangular. Alguns dos grandes sistemas são bastante flexíveis, permitindo utilizar mais de um método, um de cada vez.

Os algoritmos são as unidades básicas de que se compõem tanto os métodos quanto os sistemas. Referem-se a modos e se-

qüências de passos a serem executados para resolver um problema concreto: realizar a busca dos vizinhos de um ponto, passar uma curva de nível através de uma malha quadrada, conseguir a continuidade da derivada numa fronteira de domínios, etc. As melhoras dentro dos métodos e sistemas devem ser pensadas em termos de melhora da eficiência e qualidade dos algoritmos.

Os esforços para criar esses sistemas nasceram e se desenvolveram graças principalmente às pesquisas relacionadas com a exploração de petróleo, gazes e minérios. As somas envolvidas em negócios como esses permitem e viabilizam a automação da construção de mapas, que são elaborados com maior rapidez, fornecendo em pouco tempo os dados sobre as reservas do interior da terra, necessários para a elaboração de projetos.

Talvez esses processos, por seu custo, não seriam viáveis economicamente se projetados unicamente para atender aos fins topográficos.

Encontramos referências de dezenas e talvez centenas de programas escritos com o propósito de construir mapas. Citando palavras de J. R. Sampson podemos dizer que "a gama de variação dos programas é muito grande, desde simples e ineficazes rotinas de domínio público, até sistemas gráficos altamente sofisticados e avaliados em dezenas de milhares de dolares (...). Os algoritmos internos variam grandemente, o mesmo acontecendo com o sucesso com que representam o mapa. Desafortunadamente a perfomance de um determinado processo não pode ser prevista a partir da magnitude do preço nem da complexidade dos programas" (in Davis (1978), p. 244).

Outras queixas, que se encontram com freqüência na boca de especialistas ao avaliar diversos métodos, refere-se à pouca base teórica e a ausência de linhas de rumo na construção dos sistemas. Também apontam a relutância de diversos autores em submeter seus programas a uma análise comparativa.

- ■ -

II PARTE

ESTUDO DO MÉTODO DE REPRESENTAÇÃO DE SUPERFÍCIES DESENVOLVIDO POR HIROSHI AKIMA

"A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points"

Publication: ACM Transactions on Mathematical Software, 4.2, pp. 148-159 (junho de 1978)

INTRODUÇÃO

Nesta segunda parte desenvolvemos um estudo pormenorizado sobre o método de Akima, dividido em 2 capítulos (3 e 4).

Procuramos apresentar o método em si, tal como foi forlulado acrescentando somente alguns esclarecimentos que nos pareceram necessários para que fique bem claro o método original na formulação de seu autor e assim seja mais fácil distinguir nossa contribuição, que se encontra nas partes III e IV (capítulos 6 a 8).

No capítulo 3, baseando-nos na bibliografia nº 4, citada como epígrafe desta parte, estudamos o método em sua estrutura e teoria geral, fornecendo um resumo, os critérios e fundamentos matemáticos e analisamos algumas qualidades tais como: suavidade dos valores interpolados, tempo de computação e eficiência de algoritmos.

No capítulo 4 detalhamos o programa computacional, dividindo-o por sub-rotinas e apresentando para cada uma, um breve resumo do que faz e os esclarecimentos referentes a cada passo do programa que os necessitem. Explicitou-se também, sempre que oportuno, os conceitos matemáticos e os artifícios geométricos implícitos nas fórmulas algébricas.

- ■ -

3. O MÉTODO DE AKIMA: ESTRUTURA TEÓRICA E VISÃO GERAL

* "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points"

por: Hiroshi Akima, do Institute for Telecommunication Sciences

O método divide o plano x-y agrupando os pontos dados em um certo número de triângulos que devem ser o mais equiláteros possíveis. A seguir calcula uma função interpoladora em z(x,y) para cada célula triangular: um polinômio do 5º grau em x e y. Para determinar cada polinômio utilizam-se os valores de z e os valores estimados das derivadas parciais, em cada vértice. A seguir lança uma malha regular calculando a cota em cada nó. As rotinas de desenho, a serem fornecidas pelo usuário, devem partir dessa malha.

3.1. TRIANGULAÇÃO

Para triangularizar (ou triangular) os pontos fornecidos no plano x-y, adotou-se o critério do ângulo max-min sugerido por Lawson.

Esse critério indica que, "existindo um quadrilátero com cada ângulo interno menor do que π (180º), deve-se escolher dentre as duas possíveis partições de triângulos, aquela que maximize o mínimo ângulo interno dos triângulos produzidos".

Na triangularização do plano x-y, primeiramente unem-se

os dois pontos mais próximos para formar a aresta mínima da rede. A seguir acrescenta-se um ponto de cada vez, em ordem crescente de distância ao ponto médio dessa aresta mais curta. Esta ordenação na adição de novos pontos assegura que o novo ponto a ser acrescentado sempre cai fora do polígono construído com os pontos antigos: o novo ponto cai fora do círculo que, centrado no ponto médio da aresta mais curta, passa pelo último ponto acrescentado; enquanto que o polígono cai dentro desse círculo. Cada vez que um novo ponto é acrescentado, deve-se construir triângulos pela conexão desse ponto com todos os antigos que são "visíveis" do novo ponto e, quando necessário, trocam-se os triângulos.

O estudo da rotina IDTANG (Capítulo 4) ajuda a explicitar o mecanismo deste método.

3.2. INTERPOLAÇÃO SOBRE O TRIÂNGULO

A interpolação dos valores de z num triângulo é baseada nas três hipóteses seguintes:

1º) o valor de z é interpolado por um polinômio bivariado do 5º grau em x e y:

$$z(x,y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} \cdot x^j \cdot y^k$$

Essa fórmula desenvolvida resulta no seguinte quadro de termos do polinômio:

Tabela 3.1 - Quadro dos termos do polinômio interpolador

k =	0	1	2	3	4	5
j = 0	q_{00}	$q_{01}\cdot y$	$q_{02}\cdot y^2$	$q_{03}\cdot y^3$	$q_{04}\cdot y^4$	$q_{05}\cdot y^5$
1	$q_{10}\cdot x$	$q_{11}\cdot yx$	$q_{12}\cdot y^2x$	$q_{13}\cdot y^3x$	$q_{14}\cdot y^4x$	
2	$q_{20}\cdot x^2$	$q_{21}\cdot yx^2$	$q_{22}\cdot y^2x^2$	$q_{23}\cdot y^3x^2$		
3	$q_{30}\cdot x^3$	$q_{31}\cdot yx^3$	$q_{32}\cdot y^2x^3$			
4	$q_{40}\cdot x^4$	$q_{41}\cdot yx^4$				
5	$q_{50}\cdot x^5$					

Convém notar que há 21 coeficientes a serem determinados.

2º) Os valores da função e as derivadas parciais, de 1ª e 2ª ordem, são dadas (calculadas) em cada vértice do triângulo. Essa hipótese fornece 18 condições independentes. Ou seja, são dados:

$z,\ z_x,\ z_y,\ z_{xx},\ z_{xy},\ z_{yy}$ = 6 condições em cada vértice

$3\cdot 6 = 18$ condições no total

3º) A derivada parcial da função diferenciada na direção perpendicular (normal) a cada lado do triângulo é um polinômio do 3º grau, no máximo, na variável medida na direção do lado do triângulo. Como são três lados, essa hipótese fornece 3 condições adicionais.

A admissão desta hipótese tem dois propósitos. Em primeiro lugar adiciona 3 condições às 18 anteriores e possibilita portanto a determinação dos 21 coeficientes do polinômio in-

terpolador. Em segundo lugar, assegura a suavidade dos valores interpolados.

3.3. SUAVIDADE DOS VALORES INTERPOLADOS

A suavidade dos valores interpolados e portanto a suavidade da superfície resultante ao longo do lado do triângulo pode ser provada da seguinte forma.

Em primeiro lugar introduzimos um outro sistema cartesiano de coordenadas, que chamaremos sistema s-t, de tal forma que o eixo s seja paralelo ou coincidente com o lado do triângulo.

Para trabalhar com o novo sistema, determinam-se através do cálculo, as equações de transformação de x-y para s-t, obtendo-se relações lineares, isto é, os valores de z_x, z_y, z_{xx}, z_{yy} e z_{xy} de cada vértice determinam de maneira unívoca os valores de z_s, z_t, z_{ss}, z_{tt} e z_{st} nos mesmos vértices, cada um destes últimos como uma combinação linear dos primeiros.

No novo sistema podemos considerar agora a equação de z(s,t) ao longo do eixo s. A variável t assume o valor zero, pela própria definição dos eixos, fazendo com que a equação de z (polinômio do 5º grau em s) fique univocamente determinada pelos valores calculados de z, z_s e z_{ss}, nos dois vértices (6 valores).

Analisemos a seguir dois triângulos que possuem uma aresta em comum. Os polinômios z(x,y) do 5º grau em x e y que correspondem à função interpoladora de cada triângulo podem ser

transformados em um polinômio do 5º grau em s, para representar o valor de z na aresta comum. Como o valor do polinômio em z (expressão algébrica) só depende da magnitude de grandezas medidas nos 2 vértices comuns, isso prova que a equação de z é a mesma para os 2 triângulos. Fica assim demonstrada a continuidade da função z, ao passarmos de um triângulo a outro, através de uma aresta comum.

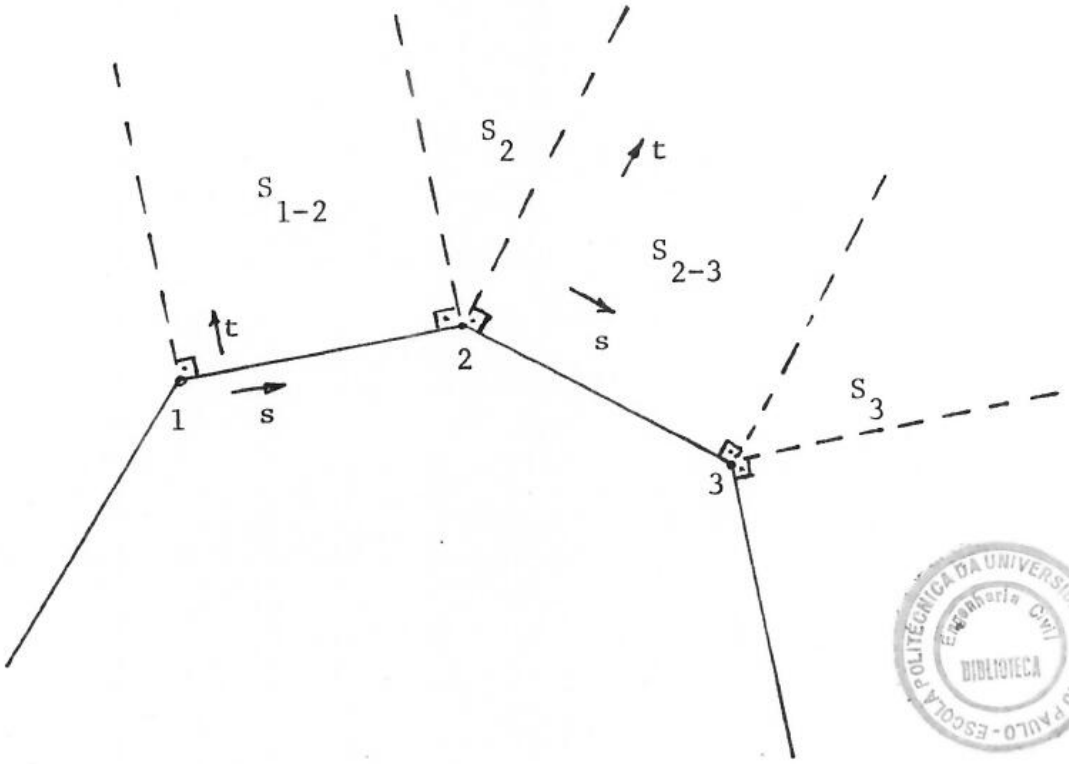
De forma análoga, podemos analisar a continuidade da derivada na direção perpendicular ao lado s. O polinômio (z_t) que representa essa derivada é assumido por hipótese como sendo do 3º grau em s, no máximo. Pode-se demonstrar facilmente que ele fica determinado univocamente pelos valores de z_t e $z_{st} = z_{ts}$ calculados nos dois vértices da aresta em questão. Com isto, os dois polinômios representando z_t em 2 triângulos que têm um lado em comum se reduzem a uma mesma equação ao longo desse lado. Isso prova a continuidade de z_t e portanto a suavidade de z ao longo do lado do triângulo.

3.4. EXTRAPOLAÇÃO

O método proposto visa fundamentalmente a interpolação de valores dentro do polígono convexo formado pelas projeções dos pontos dados. No entanto, fornece a possibilidade de extrapolação. Não é esperado um alto grau de precisão nessa extrapolação, mas em todo caso, é desejável que os valores extrapolados sejam suaves, e estejam conectados suavemente com os valores interpolados dentro do polígono.

Para estabelecer as funções de extrapolação, podemos di-

vidir a região externa ao polígono em 2 tipos de áreas, conforme a Figura 3.1.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Engenharia Civil
BIBLIOTECA

Figura 3.1 - Extrapolação na região externa ao polígono

a) Áreas semi-infinitas retangulares; apoiadas nos segmentos limites do polígono e delimitadas lateralmente por perpendiculares a esse lado (por exemplo, a área S_{1-2} na Figura 3.1); uma extrapolação de natureza desejável pode ser dada por um polinômio bivariado, do 5º grau na variável (s) medida na direção do segmento de reta (1-2) e do 2º grau na distância (t) medida na perpendicular a esse segmento de reta.

b) Áreas semi-infinitas triangulares; entre dois retângulos semi-infinitos (por exemplo S_2 na Figura 3.1),

com vértice em um ponto do polígono; a extrapolação pode ser dada por um polinômio bivariado em x e y que conecta suavemente com os dois polinômios dos retângulos vizinhos, garantindo a suavidade nas duas arestas semi-infinitas, que partem do vértice comum (no caso o 2), perpendicularmente aos lados do polígono.

3.5. CÁLCULO DAS DERIVADAS PARCIAIS

Os procedimentos para estimar as 5 derivadas parciais em cada ponto dado não são únicos. As derivadas podem ser determinadas como derivadas parciais de um polinômio do 2º grau em x e y, que passa por 6 pontos: o ponto considerado e seus 5 vizinhos mais próximos. Esse procedimento é uma extensão bivariada do método utilizado na interpolação osculatória univariada, elaborada por Ackland (1915). A determinação das derivadas é feita através da expressão algébrica do polinômio.

A adoção desse procedimento tem a vantagem de que, quando z é um polinômio do 2º grau em x e y, o método oferece resultados exatos. No entanto mostra-se mais adiante que esse procedimento acarreta, algumas vezes, resultados pouco razoáveis, gerando valores fictícios e anomalias locais (Figura 3.3, c).

No presente método, Akima adota um enfoque diferente para estimar as derivadas parciais. Utilizam-se dois estágios, no primeiro estimam-se as derivadas de primeira ordem e no segundo as de segunda ordem.

Para calcular as derivadas de primeira ordem num determinado ponto P_0, utilizam-se alguns pontos adicionais P_i ($i=1,2,\ldots,n_c$) cujas projeções são as n_c mais próximas da projeção P_0' de P_0, selecionando esses pontos P_i dentre todos os pontos dados (excluindo P_0, naturalmente).

Tomam-se dois pontos P_i e P_j, pertencentes ao conjunto dos n_c vizinhos de P_0 e constrõe-se o vetor produto de $\overline{P_0P_i}$ e $\overline{P_0P_j}$, isto é, um vetor $\vec{p}$ que é simultaneamente perpendicular a $\overline{P_0P_i}$ e $\overline{P_0P_j}$, com direção indicada pela *regra da mão direita*, e que tem magnitude igual à área do paralelogramo formado por $\overline{P_0P_i}$ e $\overline{P_0P_j}$. Deve-se tomar P_i e P_j de tal forma que o componente z do produto seja sempre positiva: $\vec{p} \wedge \vec{k} \geqslant 0$).

Constrõe-se o vetor produto de todas as possíveis combinações de $\overline{P_0P_i}$ e $\overline{P_0P_j}$ ($i \neq j$, i e $j \in a\ n_c$) e toma-se o vetor soma de todos os vetores produtos assim construídos. Assume-se então que as derivadas de primeira ordem z_x e z_y no ponto P_0 coincidem com as derivadas do plano que é normal a esse vetor soma. Convém notar que, quando $n_c = 2$, as estimativas de z_x e z_y coincidem com as derivadas parciais do plano que passa por P_0, P_1 e P_2. Note-se também que quando $n_c = 3$ e a projeção de P_0 cai dentro do triângulo formado pelas projeções de P_1, P_2 e P_3, os valores estimados de z_x e z_y são iguais às derivadas parciais do plano que passa por P_1, P_2 e P_3.

No segundo estágio, utiliza-se o mesmo procedimento de derivação parcial descrito acima (1º estágio), aplicado aos valores estimados de z_x nos pontos P_i ($i=0,1,2,\ldots,n$) e obtém-se as estimativas de $z_{xx} = (z_x)_x$ e $z_{xy} = (z_x)_y$. O mesmo proce-

dimento, efetuado com os valores de z_y, permite obter $z_{xy} = (z_y)_x$ e $z_{yy} = (z_y)_y$. Teoricamente $z_{xy} = z_{yx}$. Na prática, como não coincidem, adota-se a média para essa derivada mista.

A seleção de n_c também não é única. Obviamente, n_c não pode ser menor do que 2. Deve ser também menor que o número total de pontos dados. Afora essas exigências, "não se vê que exista um critério, uma teoria que indique um valor definido para n_c" afirma Akima. O melhor, segundo sua recomendação, baseada em diversos exemplos, é um valor entre 3 e 5 (inclusives) para n_c. Esse número corresponde a uma tentativa de encontrar o número ideal de vizinhos a serem levados em conta no cálculo das derivadas parciais.

Quando as projeções do ponto P_0 e dos n_c pontos selecionados P_i $(i=1,2,\ldots,n_c)$ são colineares no plano x-y, o procedimento descrito acima não funciona. Quando isto acontece, substitui-se o ponto mais afastado de P_0, dentre os n_c selecionados, pelo ponto seguinte que seja o mais próximo de P_0 e seja não colinear com este e com os outros selecionados anteriormente.

A mecânica prática do cálculo é desenvolvida nos comentários à rotina IDPDRV (Capítulo 4).

3.6. CONSIDERAÇÕES DE PROGRAMAÇÃO

Estão previstos dois casos típicos de configuração dos pontos nos quais a interpolação pode ser realizada. O usuário pode querer efetuar a interpolação num conjunto de pontos à

sua escolha (distribuição qualquer na saída de valores), ou então pode desejar ensaiar uma superfície suave interpolando sobre uma malha regular de pontos. É desejável dispor de uma sub-rotina separada para cada caso.

O método proposto consiste nos 5 procedimentos seguintes:

1. triangularização do plano x-y (partição num número determinado de triângulos);
2. determinação de alguns pontos (vizinhos) que são os mais próximos de cada ponto dado e que são utilizados para estimar as derivadas parciais;
3. localização dos pontos de saída nos quais a interpolação bivariada deve ser feita (isto é, determinação do triângulo em que cada ponto cai) ou organização de uma malha de saída para ajustar uma superfície suave; essa organização é feita através da relação dos nós da malha que caem dentro de cada triângulo;
4. estimativa das derivadas parciais em cada ponto;
5. interpolação puntiforme em cada ponto de saída.

Como cada um destes passos é computacionalmente indepente, foi desenvolvida uma sub-rotina para cada um.

Em algumas aplicações pode ser necessário repetir os cálculos com os mesmos valores das coordenadas x, y dos pontos dados. Em tais aplicações os dois primeiros procedimentos descritos acima não necessitam ser repetidos, se os resultados intermediários forem armazenados adequadamente. Em outras aplicações é necessário repetir os cálculos com os mesmos va-

lores x, y dos pontos de entrada e com alguns dos pontos de saída. Nesse caso, os 3 primeiros procedimentos não precisam ser repetidos. O algoritmo em questão foi feito de modo a poder poupar tempo de computação, evitando as repetições desnecessárias.

Dessa forma, habilita-se uma área de trabalho utilizada durante a computação para armazenar os resultados intermediários. Essa área de trabalho poderia ter sido reservada nos subprogramas ou no programa chamado pelo usuário. Por motivo de flexibilidade do algoritmo, preferiu-se a segunda opção.

3.6.1. CONSIDERAÇÕES SOBRE A TRIANGULAÇÃO

A triangularização (ou triangulação, simplesmente) é um procedimento consumidor de tempo. Num trabalho anterior, Akima (1975), esse algoritmo requeria um longo tempo de computação, que era proporcional ao cubo do número de pontos fornecidos ($t_c \simeq \alpha n^3$); e requeria uma área de trabalho proporcional ao quadrado do número de dados ($S_t \simeq \beta n^2$). Por causa dessas exigências, sua aplicabilidade estava limitada a conjuntos de 100 pontos no máximo, para a maioria dos computadores. Com a implementação do presente método, Akima melhorou substancialmente o algoritmo de triangulação.

Este novo algoritmo requer um tempo de computação que é aproximadamente proporcional ao quadrado do número de pontos. Por exemplo, consome 0,005; 0,48 e 32,0 segundos para 10, 100,

e 1.000 pontos dados, respectivamente, no computador CDC-6.600. Sua área de trabalho é proporcional (e cerca de 20 vezes) ao número de pontos dados. O quadro abaixo ajuda a visualisar esses dados.

Tabela 3.2 - Tempo de computação e área de trabalho utilizada pelo computador CDC-6.600 para diversos números de pontos

Número de pontos n	Tempo de computação $\alpha \cdot n^2$ ($\alpha \simeq 3$ a 5.10^{-5})	Área de trabalho $\beta \cdot n$ ($\beta \simeq 20$)
10	0,005	200
100	0,48	2.000
1.000	32,0	20.000

3.6.2. CONSIDERAÇÕES SOBRE A PROCURA DE VIZINHOS

A seleção dos pontos que estão mais próximos de um ponto especificado (vizinho) é outro procedimento consumidor de tempo. O algoritmo apresentado no trabalho anterior, (Akima 1975), era ineficiente e foi feita uma melhora. O atual requer um tempo de computação proporcional ao quadrado do número de pontos, e que é menor que a metade do tempo gasto na triangularização do mesmo número de pontos. Ou seja, a constante de proporcionalidade neste procedimento é metade da constante α da triangularização.

3.6.3. CONSIDERAÇÕES SOBRE A LOCALIZAÇÃO DE PONTOS

A localização de um ponto na malha triangular (isto é a determinação do triângulo no qual o ponto cai) é também um pro-

cedimento consumidor de tempo. A melhora deste processo foi feita pela divisão do plano x-y em nove áreas retangulares e pela listagem dos triângulos associados a cada retângulo. O algoritmo apresentado requer um tempo de computação aproximadamente proporcional à raiz quadrada do número de ponto dados, sendo bastante mais rápido que o apresentado anteriormente: Akima (1975).

3.6.4. CONSIDERAÇÕES SOBRE O AJUSTE DE UMA SUPERFÍCIE

Para ajustar uma superfície suave, todos os pontos da malha retangular são localizados e é feita a interpolação para cada um deles. Neste caso, tanto a localização como a interpolação são mais rápidas na base de triângulo por triângulo do que na base de ponto por ponto da malha. Akima programou uma nova sub-rotina IDGRID que organiza os pontos da malha para o ajuste da superfície através da classificação (ordenação) dos mesmos com relação aos triângulos; para cada triângulo, esta sub-rotina varre a malha retangular de pontos, pesquisando os que caem nesse triângulo, listando-os em uma tabela. Dessa forma poupa-se tempo pois a interpolação é realizada consecutivamente para os pontos que caem em cada triângulo, de maneira a evitar repetições no cálculo dos coeficientes do polinômio interpolador.

3.6.5. CONSIDERAÇÕES SOBRE O CÁLCULO DAS DERIVADAS

Outro problema de programação é a escolha entre:

1) a computação no começo e o armazenamento das estima-

tivas das derivadas parciais em todos os pontos ou

2) computar as derivadas em cada ponto no momento em que seja necessário.

Foi escolhida a primeira opção porque o primeiro passo do método para estimar as derivadas parciais num ponto dado envolve o cálculo em vários pontos adicionais e é mais simples realizar o primeiro passo inicial para todos os pontos. Obviamente, esta opção é ineficiente no caso de muitos pontos dados e poucos pontos de saída. No entanto, através da redução prévia dos pontos fornecidos, o usuário pode diminuir os tempos de computação não só para o cálculo das derivadas mas também para os outros procedimentos, incluindo a triangularização.

3.7. EXEMPLOS DE APLICAÇÃO

Tomando um exemplo de um estudo anterior, Akima (1974), foi ilustrada a aplicação deste novo método. Para isso foram tomados aleatoriamente 50 pontos de uma superfície (conforme coordenadas da Tabela 3.3).

A Figura 3.2 mostra o mapa de curvas de nível para a superfície resultante de 30 pontos com asterisco da Tabela 3.3, enquanto que a Figura 3.3 mostra o mesmo mapa para todos os 50 pontos da mesma tabela.

Em cada figura, a superfície original da qual os pontos foram amostrados é mostrada em (a). A superfície ensaiada com trechos de planos, cada um aplicável a um triângulo, está mostrada em (b). Naturalmente, tal superfície é contínua mas não

Tabela 3.3 - Um exemplo de pontos amostrados. São fornecidos 50 pontos através de suas coordenadas x, y e z

i	x_i	y_i	z_i	i	x_i	y_i	z_i
1 *	11,16	1,24	22,15	26	3,22	16,78	39,93
2 *	24,20	16,23	2,83	27 *	0,00	0,00	58,20
3	12,85	3,06	22,11	28 *	9,66	20,00	4,73
4 *	19,85	10,72	7,97	29	2,56	3,02	50,55
5 *	10,35	4,11	22,33	30 *	5,22	14,66	40,36
6	24,67	2,40	10,25	31 *	11,77	10,47	13,62
7 *	19,72	1,39	16,83	32	17,25	19,57	6,43
8	15,91	7,74	15,30	33 *	15,10	17,19	12,57
9 *	0,00	20,00	34,60	34 *	25,00	3,87	8,74
10 *	20,87	20,00	5,74	35	12,13	10,79	13,71
11	6,71	6,26	30,97	36 *	25,00	0,00	12,00
12	3,45	12,78	41,24	37	22,33	6,21	10,25
13 *	19,99	4,62	14,72	38	11,52	8,53	15,74
14	14,26	17,87	10,74	39 *	14,59	8,71	14,81
15 *	10,28	15,16	21,59	40 *	15,20	0,00	21,60
16 *	4,51	20,00	15,61	41	7,54	10,69	19,31
17	17,43	3,46	18,60	42 *	5,23	10,72	26,50
18	22,80	12,39	5,47	43	17,32	13,78	12,11
19 *	0,00	4,48	61,77	44 *	2,14	15,03	53,10
20	7,58	1,98	29,87	45 *	0,51	8,37	49,43
21 *	16,70	19,65	6,31	46	22,69	19,63	3,25
22 *	6,08	4,58	35,74	47 *	25,00	20,00	0,60
23	1,99	5,60	51,81	48	5,47	17,13	28,63
24 *	25,00	11,87	4,40	49 *	21,67	14,36	5,52
25 *	14,90	3,12	21,70	50 *	3,31	0,13	44,08

* Tabela fornecida por Akima, bibliografia [4]

suave. A superfície ajustada pelo método que estima as derivadas parciais com um polinômio do segundo grau é mostrada em (c). As superfícies ajustadas pelo método proposto usando 3,4 e 5 pontos adicionais para estimar as derivadas parciais em cada ponto estão mostradas em (d), (e) e (f), respectivamente. No desenho das curvas de nível os valores de z foram interpolados, pelos respectivos métodos, nos nós de uma malha consistente em 100 × 80 quadrados. Em cada quadrado os valores de

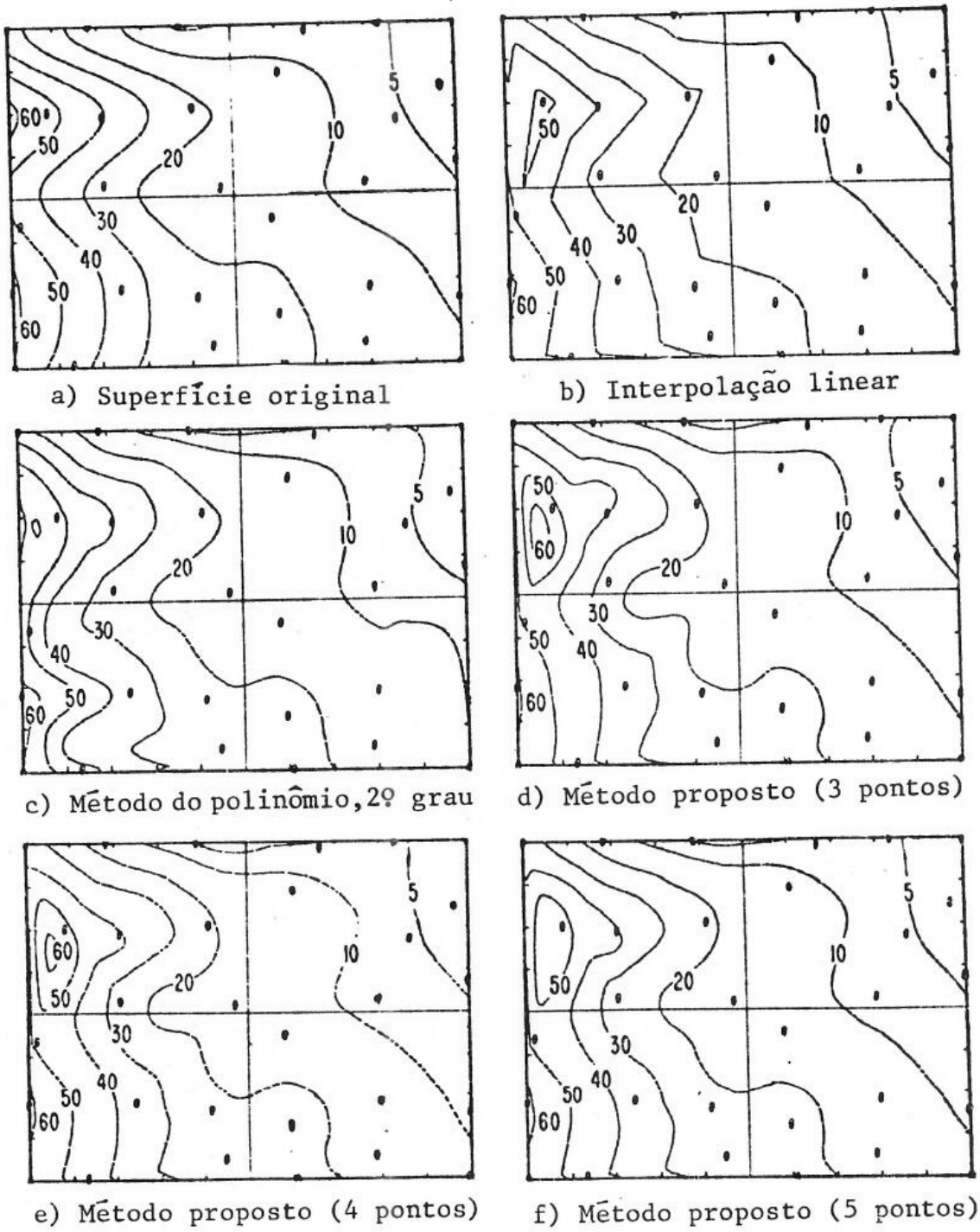


Figura 3.2 - Mapa de curvas de nível utilizando os 30 pontos da Tabela 3.3, marcados com asterísco. O número de pontos entre parenteses para (d), (e) e (f) refere-se ao número de vizinhos utilizados para o cálculo das derivadas parciais.

* Figuras fornecidas por Akima, bibliografia [4]

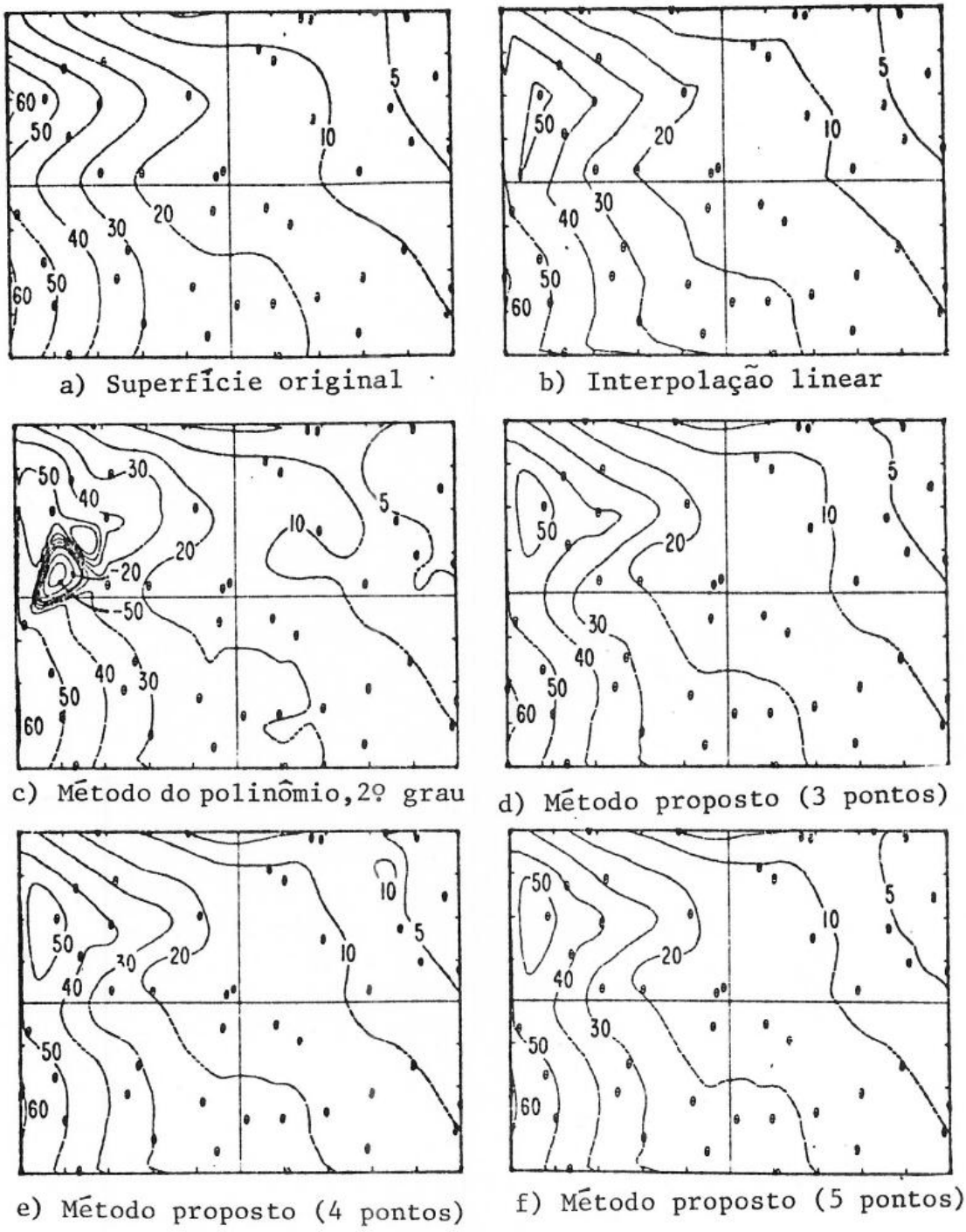


Figura 3.3 - Mapa de curvas de nível para os 50 pontos da Tabela 3.3.

* Figuras fornecidas por Akima, bibliografia [4]

z foram interpolados linearmente.

As Figuras 3.2 e 3.3 indicam que o método proposto produz resultados razoáveis. Nessas figuras nota-se que há uma pequena variação no desenho quando se empregam diferentes números de vizinhos (3, 4 ou 5) para estimar as derivadas parciais. As Figuras 3.2 (c) e 3.3 (c) demonstram a peculiar idiosincrasia do método que calcula as derivadas através de um polinômio do segundo grau: mais pontos levam a resultados muito piores, neste exemplo.

A decisão de se o método proposto é aplicável ou não a um problema particular compete a cada possível usuário do mesmo. Espera-se que o exemplo fornecido possa ajudar numa tomada de decisão. Uma comparação de (d), (e) ou (f) realizadas com o método proposto, com a superfície original (a) ou a superfície de trechos de plano (b), em cada figura, pode ser uma boa ajuda para tal decisão. A comparação das Figuras 3.2 e 3.3 dá uma idéia da variação da superfície resultante com relação ao número total de pontos fornecidos.

3.8. PROPRIEDADES DO MÉTODO

Akima apresenta em seu trabalho as seguintes propriedades:

1. O método não arredonda os dados, isto é, a superfície resultante passa através de todos os pontos fornecidos. Nesse sentido, o método só é aplicável quando os valores de z são conhecidos com exatidão, ou quando os erros são desprezíveis.

2. Da mesma forma que qualquer método de interpolação, a

precisão dos valores intermediários não pode ser garantida. Pode-se dizer no entanto que o método em questão foi testado com sucesso contra os valores precisos de uma forma funcional (função matemática conhecida).

3. O resultado do método é invariante quanto à rotação do sistema de coordenadas x-y.

4. O método é linear. Em outras palavras, se

$$z(x_i,y_i) = a \cdot z'(x_i,y_i) + b \cdot z''(x_i,y_i)$$

para todo i, os valores interpolados satisfazem a

$$z(x,y) = a \cdot z'(x,y) + b \cdot z''(x,y),$$

onde a e b são constantes reais arbitrárias.

5. O método fornece resultados exatos quando z(x,y) representa um plano, isto é,

$$z(x,y) = a_{00} + a_{10} \cdot x + a_{01} \cdot y,$$

onde a_{00}, a_{10}, a_{01} são constantes reais arbitrárias.

6. O método requer somente procedimentos diretos. Não há problemas de estabilidade computacional ou de convergência em sua aplicação.

Além disso, merecem destaque as propriedades do método de Akima apresentadas por Franke (1982), em que são comparados 32 métodos selecionados dentre os melhores em uso nos EUA. Essa comparação levou em conta os seguintes aspectos: precisão, as-

pecto visual, sensibilidade com relação a parâmetros internos, tempo de processamento e memória requerida. Passamos a analisar cada um desses fatores, numerando-os em sequência com relação às qualidades já apresentadas.

7. Precisão. É um importante aspecto de comparação, que se refere à fidelidade com que um método reproduz a superfície que está sendo modelada.

Nas aplicações usuais não existe uma função matemática que represente a superfície com toda exatidão. Na realidade, só conhecemos o valor da função em alguns pontos amostrados e fazemos hipóteses razoáveis para determinar o seu valor nas regiões intermediárias.

Para avaliar a precisão de um método costuma-se fazer o teste contra uma forma funcional, que consiste em aplicar o algoritmo a superfícies conhecidas, isto é, a superfpicies representadas por funções matemáticas pré-definidas. Dessa forma pode-se quantificar os desvios e medir a precisão.

Franke (1982) descreve os testes que fez, classificando a seguir os 32 métodos analisados, de acordo com notas que variam de A até F. O método de Akima recebeu a nota B, o que nos parece um resultado bastante satisfatório.

8. Aspecto visual. Ainda que conceitualmente não tenha tanto relevo quanto a precisão, o aspecto visual não pode, de forma alguma, ser desprezado.

Em primeiro lugar porque o usuário está acostumado com a boa qualidade das curvas de nível feitas por um desenhista e reluta em aceitar um produto de pior qualidade pois, para ele,

uma aparência pior está associada a um menor cuidado em todo o trabalho. Não só no desenho mas também na interpolação e precisão dos resultados.

A definição da qualidade estética de um mapa é à primeira vista um tanto subjetiva; mas as pessoas que trabalham com curvas de nível estão aptas a dar um parecer digno de confiança, e o mais provável é que a classificação de dois entendidos coincida.

Para avaliar o desempenho de diversos métodos com relação a este parâmetro Franke comparou os respectivos desenhos e perspectivas tridimensionais, separando-os por classes, de A até F. O método de Akima recebeu a qualificação C, o que significa que é aceitável; pode no entanto ser melhorado. E foi o que fizemos, entre outras coisas, neste trabalho.

9. Sensibilidade a parâmetros. Muitos métodos envolvem a fixação ou definição, por parte do usuário, de um ou mais parâmetros. Os resultados, tanto em valores numéricos como no desenho, podem variar grandemente em função dos valores adotados para esses parâmetros. A magnitude dessa variação constitui a sensibilidade do método aos parâmetros. Para verificar essa sensibilidade e, dentro do possível, quantificá-la, pode-se utilizar o mesmo conjunto de dados, processando-o cada vez com um valor diferente dos parâmetros.

No entanto, mais do que medir ou avaliar, é desejável que o método seja invariante a parâmetros e a mudanças ou rotações de eixos: a superfície real não depende de parâmetros.

No método original de Akima dispõe-se de um parâmetro

que é o número de vizinhos para o cálculo de derivada. Pode variar de 2 a 25, sendo que o autor recomenda os valores 3, 4 ou 5.

Com a introdução de algumas modificações fizemos com que o método trabalhe com os vizinhos naturais (ou de Voronoi), que é um invariante dado pela configuração geométrica. Eliminamos portanto a sensibilidade a parâmetros, conseguindo a estabilidade do interpolante.

10. Tempo de processamento. Para os grandes computadores, em nosso país, esse parâmetro tem grande importância pois é um dos principais determinantes do custo.

Franke afirma que esse recurso não tem grande interesse, a menos que seja muito alto. Essa afirmação não é válida entre nós já que se cobra o tempo de processamento por segundo de utilização, cuja taxa está longe de ser desprezível.

Por isso justificam-se entre nós os investimentos para minimizar o recurso tempo. O método de Petinatti (1983) mostra essa preocupação clarissimamente, desenvolvendo altas técnicas para baixar o tempo de computação e apontando esse fator como decisivo para definir o sucesso ou o fracasso de um método.

O método de Akima, de acordo com o estudo de Franke (1982) é de longe o mais rápido de todos os 32 métodos estudados, tendo recebido a nota A/A.

Examinando os outros 31 métodos, pode-se ver que há um compromisso entre as diversas qualidades. Os métodos que apresentam qualidade visual ótima (nota A) exigem muitíssimo tempo de computação, recebendo uma qualificação do tipo B/C ou C/D, o

que os inviabiliza do ponto de vista econômico para os grandes computadores, e do ponto de vista operacional (tempo) para os microcomputadores.

Com relação às qualificações, a primeira letra representa a soma de tempos para 3 casos com 100, 33 e 25 vértices, e as faixas de tempo A, B, C, D e E são respectivamente (0,7], (7,21], (21,31], (31,50] e (50,∞). A segunda letra representa o tempo total para 100 vértices e uma malha de 33 x 33 (1089) pontos. As faixas de tempo são, respectivamente, (0,4], (4,12], (12,20], (20,30], e (30,∞).

11. Memória alocada. No cômputo deste fator deve-se levar em conta a área de programa e a área para dados. Franke, em seu estudo, não levou em consideração a área destinada ao programa pois preocupa-se com grandes computadores. No entanto, para aplicações em micro, a memória é um dado de muitíssima relevância, que pode determinar sua adoção ou não. Deve-se ter em conta necessariamente a área de programa e também a dimensão da maior sub-rotina para efeitos de fracionamento de programa.

Pode-se ver também no trabalho de Franke que os métodos de melhor qualidade, além de grande tempo de processamento, exigem grande capacidade de memória, proporcional a n^2 (quadrado do número de pontos). Isso os inviabiliza para a aplicação em micros, que é um dos objetivos desta tese.

4 - O ALGORITMO PROPOSTO POR AKIMA

* Collected Algoritms from ACM

Algorithm 526: *Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points* (E_1).

Linguagem: ANSI Standard Fortran

Neste capítulo faremos uma descrição geral do algoritmo 526; estudaremos suas sub-rotinas explicitando a mecânica de cálculo e os fundamentos matemáticos envolvidos.

Como é natural, os comentários tornam-se mais fáceis de entender quando lidos paralelamente com a listagem do programa.

4.1. DESCRIÇÃO GERAL DO PROGRAMA

Este algoritmo está composto por 2 programas principais que realizam a interface com o usuário. São eles:

IDBVIP - (Sigla para BiVariate InterPolation) - realiza a interpolação em um ou alguns pontos especificados, isto é, calcula o valor de z em pontos cujas coordenadas x, y devem ser fornecidas.

IDSFFT - (Sigla para SurFace FiTting) - realiza a interpolação preparando o ajuste de uma superfície

suave. Para isto estima o valor de z, como no caso anterior mas tomando os pontos de maneira sistemática: nos nós de malha regular, tão densa quanto se queira. Gera uma matriz bidimensional contendo os valores de z estimados.

Restam 6 sub-rotinas básicas, que constituem o suporte para os 2 programas principais. Essas sub-rotinas são chamadas por IDBVIP ou por IDSFFT ou por ambos, conforme o esquema que apresentamos a seguir na Figura 4.1.

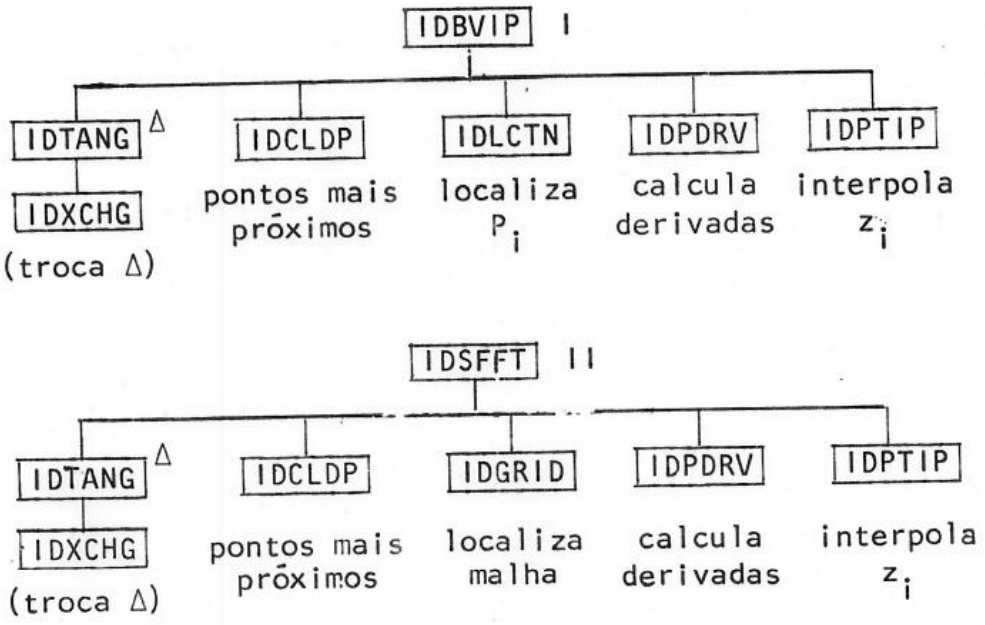


Figura 4.1 - Esquema dos 2 programas, indicando as sub-rotinas chamadas.

As seis sub-rotinas são as seguintes:

1. IDCLDP - (CLosest Data Points) - determina os n_c pontos mais próximos de cada vértice da triangulação

2. IDGRID - (GRID - malha) - organiza a saída dos pontos da malha para o programa IDSFFT, classificando-os por

ordem crescente do número de triângulos.

3. IDLCTN - (LoCaTe a poiNt) - localiza um ponto para o programa IDBVIP, isto é, determina o triângulo em que o ponto cai.

4. IDPDRV - (Partial DeRiVates) - calcula as derivadas parciais nos pontos fornecidos.

5. IDPTIP (PuncTual InterPolation) - realiza a interpolação em um ponto, isto é, calcula os coeficientes do polinômio e a seguir o valor de z(x,y).

6. IDTANG - (TriANGulates) - triangulariza os pontos dados.

Existe também uma função que é a seguinte:

7. IDXCHG - (eXCHanGe) - chamada por IDTANG, esta função verifica se é necessária ou não uma troca de triângulos.

Na Figura 4.1, indicamos o esquema de chamada das subrotinas pelos 2 programas IDBVIP e IDSFFT.

A ordem de chamada é da esquerda para a direita. Pode-se notar um bloco comum de rotinas para os 2 programas, variando somente IDLCTN (que só é chamada por IDBVIP) e IDGRID (que só é chamada por IDSFFT).

4.2. MEMÓRIA, TEMPOS E MODOS DE COMPUTAÇÃO

O pacote, que inclui dois blocos comuns (IDLC e IDPI), ocupa 3.500 posições de memória no computador CDC-6600.

Quando se utiliza um dos programas repetidamente com dados parcialmente idênticos como parte da próxima chamada, pode-se poupar tempo de computação especificando um apropriado modo (MD) de computação.

MD = 1 utilizado para a chamada inicial ou para os casos em que se altera o número de vizinhos (NCP) para estimar as derivadas parciais, ou ainda quando se fornece um novo conjunto de dados (x,y,z).

MD = 2 utilizado quando se mantém o número de vizinhos e o conjunto de pontos fornecidos (coordenadas x, y,z) e se quer calcular os valores de z em outra malha diferente da inicial.

MD = 3 utilizado quando se mantém o número de vizinhos, o conjunto de pontos fornecidos e se quer calcular os valores de z na mesma malha inicial acrescida de alguns pontos.

Tabela 4.1 - Tempo de computação aproximado de IDBVIP; computador CDC-6600

NDP	NPI	Tempos (segundos)		
		MD = 1	MD = 2	MD = 3
10	10	0,03	0,02	0,02
	100	0,13	0,12	0,08
	1.000	1,10	1,10	0,70
100	10	0,75	0,10	0,07
	100	0,86	0,22	0,14
	1.000	2,10	1,50	0,84
1.000	10	47,00	0,84	0,60
	100	47,00	1,10	0,68
	1.000	50,00	3,80	1,40

* cfr. Akima [4]

Tabela 4.2 - Tempo de computação aproximado de IDSFFT; computador CDC-6600

NDP	NXI*NYI	Tempo (segundos)		
		MD = 1	MD = 2	MD = 3
10	11*11	0,08	0,07	0,04
	33*33	0,41	0,40	0,22
	101*101	3,50	3,40	1,80
100	11*11	0,86	0,20	0,14
	33*33	1,30	0,66	0,37
	101*101	4,60	4,00	2,00
1.000	11*11	47,00	1,10	0,70
	33*33	49,00	2,60	1,30
	101*101	54,00	8,20	3,50

As Tabelas 4.1 e 4.2 mostram o tempo de computação aproximado, necessário para os programas IDBVIP e IDSFFT, respectivamente, no computador CDC-6600, quando são utilizados 4 pontos adicionais para estimar as derivadas parciais em cada ponto (NCP = 4).

Nessas tabelas NDP é o número de pontos dados, NPI é o número de pontos de saída nos quais a interpolação deve ser realizada e NXI e NYI são os números de pontos da malha (divisões) nas coordenadas x e y. MD é o modo de computação.

4.3. DESCRIÇÕES DAS SUB-ROTINAS

Descrevemos a seguir as seis sub-rotinas básicas e a função IDXCHG, seguindo, por motivos didáticos, a mesma ordem em que são chamadas pelos 2 programas principais.

Tenha-se em conta que TTIDBS é uma rotina de teste e que pode ser eliminada uma vez que o programa esteja acertado. IDBVIP e IDSFFT não necessitam maiores explicações pois sua

função é chamar as sub-rotinas menores (Figura 4.1). Por isso não damos maiores detalhes referentes a esses três componentes do programa de Akima.

3.1. FUNÇÃO IDXCHG (algoritmo CACM 526, p. 26)

Esta função determina se uma troca de 2 triângulos é necessária ou não, utilizando o critério do ângulo max-min, fornecido por C.L. Lawson.

Os parâmetros a serem fornecidos são:

* x,y = listas contendo as coordenadas dos vértices (pontos dados)
* I1, I2, I3 e I4 = número de quatro pontos P_1, P_2, P_3 e P_4, que formam um quadrilátero, com P_3 e P_4 conectados diagonalmente.

A função retorna à rotina IDTANG com o valor inteiro 1 (um) quando a troca é necessária, e com o valor 0 (zero) caso contrário.

Segundo a ordem de listagem do programa podemos fazer os seguintes comentários e esclarecimentos:

1) As fórmulas U3 = (Y2-Y3)*(X1-X3) - (X2-X3)*(Y1-Y3) e as análogas U4 = ... U1 = ... U2 = ... representam o módulo do produto vetorial de vetores que são lados do quadrilátero. De acordo com a Figura 4.2 podemos escrever as equações correspondentes.

$$\vec{p} = \overrightarrow{P_3P_1} = P_1 - P_3 = \Delta x_{13}\vec{i} + \Delta y_{13}\vec{j} = (x_1-x_3)\vec{i} + (y_1-y_3)\vec{j}$$

$$\vec{q} = \overrightarrow{P_3P_2} = P_2 - P_3 = \Delta x_{23}\vec{i} + \Delta y_{23}\vec{j} = (x_2-x_3)\vec{i} + (y_2-y_3)\vec{j}$$

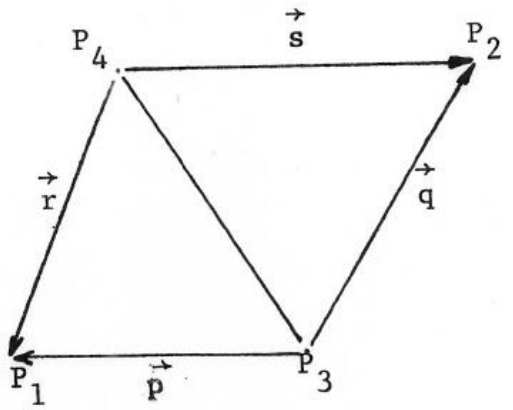


Figura 4.2 - Quadrilátero e produtos vetoriais

$$|\vec{t}| = |\vec{p}\wedge\vec{q}| = \Delta x_{12}\cdot\Delta y_{23} - \Delta x_{23}\cdot\Delta y_{13} \qquad \vec{t} = |\vec{t}|\cdot\vec{k}$$

formalmente:

$$u_3 = \begin{vmatrix} \Delta x_{13} & \Delta y_{13} \\ \Delta x_{23} & \Delta y_{23} \end{vmatrix} \qquad |\vec{p}\wedge\vec{q}| = 2\times S\Delta_{123}$$

$$p.q.\ \text{sen}\theta$$

$$u_4 = \begin{vmatrix} \Delta x_{24} & \Delta y_{24} \\ \Delta x_{14} & \Delta y_{14} \end{vmatrix} \qquad |\vec{s}\wedge\vec{r}| = 2\times S_{\Delta 124}$$

$$u_1 = \begin{vmatrix} \Delta x_{41} & \Delta y_{41} \\ \Delta x_{31} & \Delta y_{31} \end{vmatrix} \qquad |-\vec{r}\wedge-\vec{p}| = |\vec{r}\wedge\vec{p}| = 2\times S\Delta_{134}$$

$$u_2 = \begin{vmatrix} \Delta x_{32} & \Delta y_{32} \\ \Delta x_{42} & \Delta y_{42} \end{vmatrix} \qquad |-\vec{q}\wedge-\vec{s}| = |\vec{q}\wedge\vec{s}| = 2\times S\Delta_{234}$$

Nas fórmulas à direita explicitamos o fato de o produto vetorial ser igual ao dobro da área do triângulo formado pe-

los 2 vetores. Isso será utilizado logo a seguir.

2) A verificação de se U3*U4 é menor ou igual a zero, determina a configuração geométrica do quadrilátero. Caso haja um ângulo interno maior ou igual a 180º, não se deve modificar o triângulo 234 e então se faz IDX= = IDXCHG = 0 e se retorna à sub-rotina IDTANG. Na Figura 4.3 indicamos as possíveis configurações do quadrilátero em função do valor do produto U3*U4.

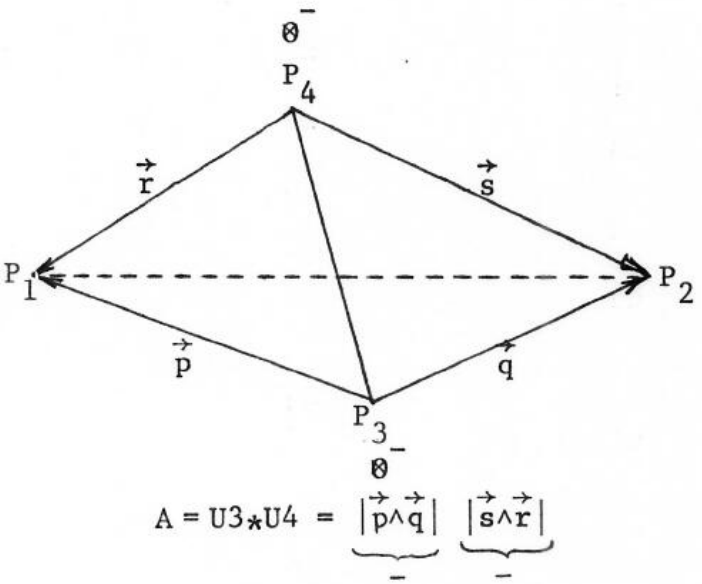


Figura 4.3,a) - A > 0, os produtos vetoriais são de mesmo sinal e o seu produto A = U3*U4 é positivo. Os cálculos prosseguem para verificar se a troca de triângulos é necessária ou não: isso dependerá dos ângulos.

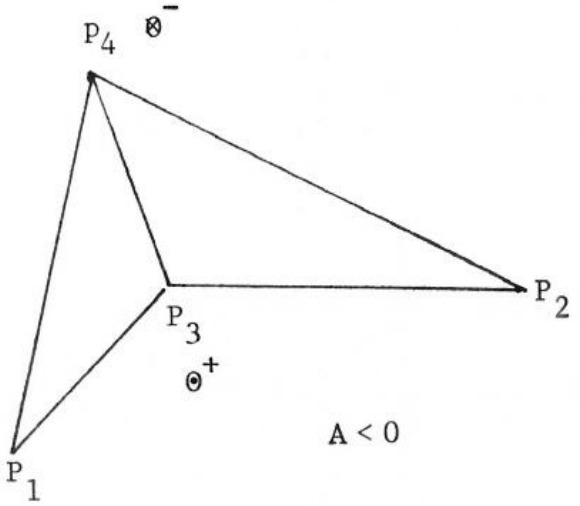


Figura 4.3,b) - A < 0, os produtos vetoriais são de sinais contrários e o seu produto A = U3*U4 é negativo. Isso significa que um dos ângulos internos é maior que 180º e o quadrilátero não é estritamente convexo. Não é necessária a troca e os cálculos terminam. Retorna-se à rotina principal com IDXCHG = 0.

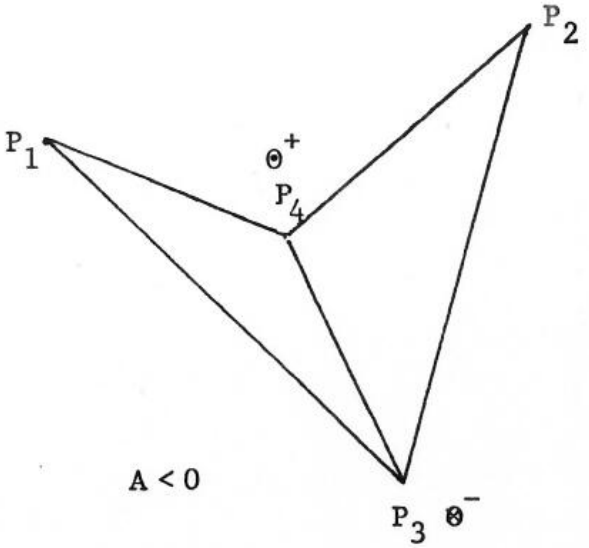


Figura 4.3,c) - Trata-se de situação completamente análoga à anterior. A única diferença reside em qual dos ângulos é maior que 180º. IDXCHG = 0.

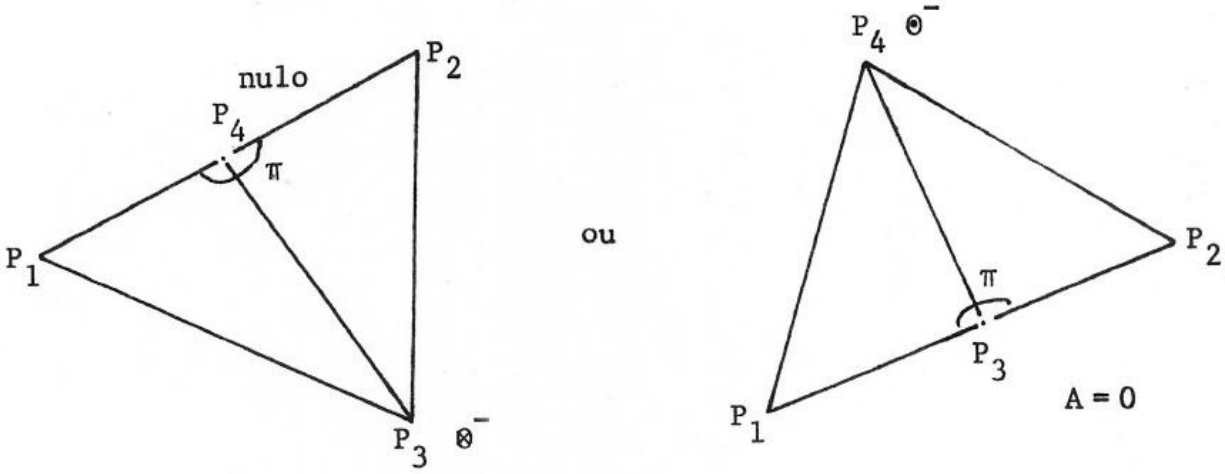


Figura 4.3,d) - A = 0, um dos produtos vetoriais (ou ambos, em caso extremo não representado) é nulo. Isso indica que três pontos estão alinhados e não é necessária a troca de triângulos (produziria um ângulo nulo, que portanto é o mínimo geral). Retorna-se à rotina principal com IDXCHG = 0.

3) Neste comentário, vamos interpretar algumas fórmulas apresentadas, de acordo com a nomenclatura da Figura 4.4.

$$a_1sq = a_1^2 = p^2 = (x_1-x_3)^2 + (y_1-y_3)^2$$

$$b_1sq = r^2$$

$$c_1sq = s^2$$

$$a_2sq = q^2$$

$c_3sq = \qquad f^2$

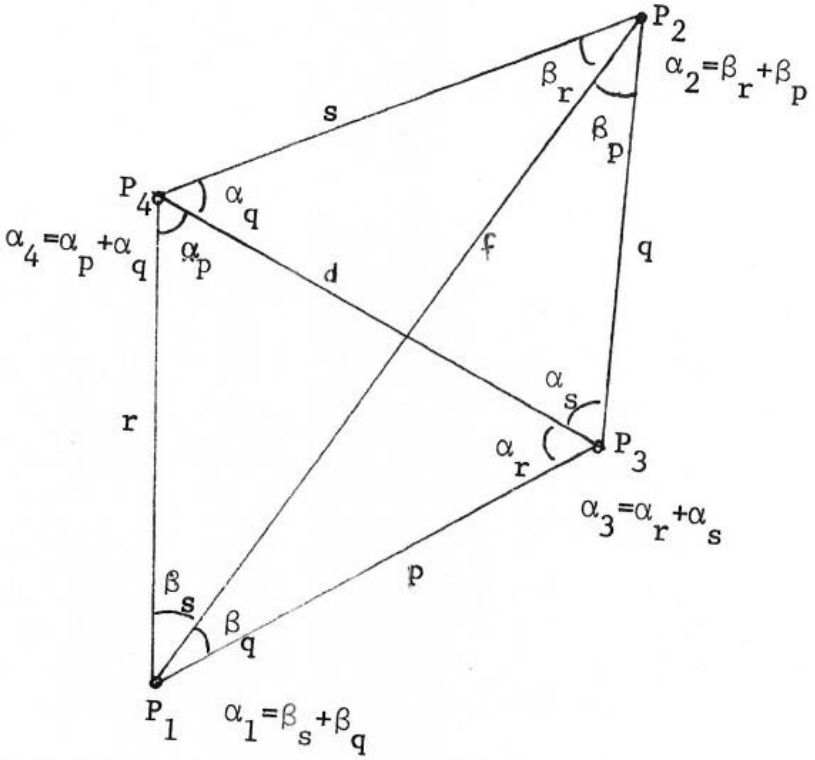


Figura 4.4 - Quadrilátero e nomenclatura

4) Lembrando que os produtos vetoriais representam o dobro da área do triângulo podemos escrever

$$U1 = 2 \cdot p \cdot r \cdot \text{sen}\alpha_1$$

$$U2 = 2 \cdot s \cdot q \cdot \text{sen}\alpha_2$$

$$U3 = 2 \cdot p \cdot q \cdot \text{sen}\alpha_3$$

$$U4 = 2 \cdot r \cdot s \cdot \text{sen}\alpha_4$$

Isso nos leva a compreender o significado de mais algumas expressões:

$$S1SQ = s_1^2 = U1^2 \ / \ c_1^2 * \max(a_1^2 \ / \ b_1^2) = \frac{4 \cdot p^2 \cdot r^2 \text{sen}^2\alpha_1}{d^2 \cdot \max(p^2 \ / \ r^2)}$$

a) se $p > r$ $\qquad (\alpha_p > \alpha_r)$

$$s_1^2 = 4\cdot(\text{sen}\alpha_r)^2 \qquad \left\{\frac{\text{sen}\alpha_1}{d} = \frac{\text{sen}\alpha_r}{r} = \frac{\text{sen}\alpha_p}{p}\right\}$$

b) se $r < p$ $\qquad (\alpha_r > \alpha_p)$

$$s_1^2 = 4\cdot(\text{sen}\alpha_p)^2$$

donde se conclui que $s_1^2 = 4\cdot(\text{sen}\alpha_{min})^2$; ou seja s_1^2 é uma medida do ângulo mínimo do triângulo 1-3-4: α_p ou α_r. Lembrar que quando maior um ângulo, até 90º, maior o seu seno.

Analogamente, S2SQ é uma medida do mínimo ângulo do triângulo 2-3-4: α_q ou α_s.

E da mesma forma S3SQ é uma medida do mínimo ângulo do triângulo 1-2-3: β_p ou β_q. E S4SQ é uma medida do ângulo mínimo do triângulo 1-2-4: : β_s ou β_r.

5) O passo seguinte do programa realiza uma comparação entre o mínimo dos ângulos α_i com o mínimo dos ângulos β_i. Pode-se então determinar o máximo dos ângulos mínimos e escolher a configuração correspondente já que a configuração atual corresponde aos α_i e a configuração opcional aos β_i. Se o mínimo dos β_i for maior que o mínimo dos α_i, então trocam-se os triângulos.

Como observação podemos dizer que não é necessário, para encontrar o ângulo mínimo absoluto, fazer a comparação com os ângulos nos vértices ($\alpha_1, \alpha_2, \alpha_3$ ou α_4) pois estes são fracionados pelas diagonais em teste, dando origem a 2 ângulos que serão forçosamente menores do que ele.

4.3.2. SUB-ROTINA IDTANG (pp. 22 a 26)

Esta sub-rotina realiza a triangulação, ou seja:

- interliga adequadamente os pontos do plano x-y, formando triângulos aceitáveis;
- determina os segmentos (arestas) que formam a borda da área;
- determina o número dos triângulos correspondentes aos segmentos de borda;
- lista o número dos vértices de cada triângulo no sentido anti-horário;
- lista o número dos dois extremos de cada segmento de borda, no sentido anti-horário; sendo que a ordem de listagem dos segmentos de borda também é anti-horária;

Esta sub-rotina chama a função IDXCHG para verificar se a troca de triângulos é necessária ou não.

Os parâmetros a serem fornecidos são:

* NDP = número de pontos dados (vértices)
* XD = lista de dimensão NDP contendo as coordenadas x dos pontos
* YD = idem, coordenadas y dos pontos fornecidos

Os parâmetros de saída são:

* NT = número de triângulos
* IPT = lista de inteiros de dimensão 6*NDP - 15, onde os números dos vértices do (IT)^{ésimo} triângulo são armazenados como sendo os (3*IT-2), (3*IT-1)

e $(3*IT)^{ésimos}$ elementos dessa lista. IT = 1,2,.. .,NT.

Como se verá mais adiante, 6*NDP - 15 é o número máximo de vértices a serem armazenados para uma triangulação com NDP pontos.

* NL = número de segmentos de borda;
* IPL = lista inteira de dimensão 6*NDP onde: o número dos 2 vértices extremos de cada $(IL)^{ésimo}$ segmento de borda e o número do respectivo triângulo a que IL pertence são armazenados como sendo os (3*IL-2), (3*IL-1) e $(3*IL)^{ésimo}$ elementos. IL = = 1,2,...,NL.

Os outros parâmetros são:

* IWL = lista de inteiros usada internamente como área de trabalho, com dimensão 18*NDP;
* IWP = idem, dimensão: NDP;
* WK = lista (valores reais e não inteiros) de dimensão NDP, usada internamente como área de trabalho.

Acompanhando a listagem fazemos as seguintes observações:

1) Significado das funções <u>DSQF</u> e <u>SIDE</u>:

De acordo com a Figura 4.5, DSQF(U1,V1,U2,V2) = d_{12}^2 (distância entre dois pontos ao quadrado). Quando se trata de fazer comparação entre duas distâncias não é necessário extrair a raiz quadrada, o que só implicaria em mais cálculos.

A função SIDE pode ser interpretada como sendo o produto vetorial:

$$|\vec{p}_{12} \wedge \vec{q}_{13}| = \begin{vmatrix} (u_2-u_1) & (v_2-v_1) \\ (u_3-u_1) & (v_3-v_1) \end{vmatrix}$$

No entanto, pode ser interpretada também como um modo de determinar a posição (side = lado) de um ponto (no caso o ponto 3), com relação a uma reta (no caso a determinada por 1-2). Se SIDE for positiva, 3 estará à esquerda de 1-2; se for negativa, à direita e se for nula teremos o alinhamento dos 3 pontos.

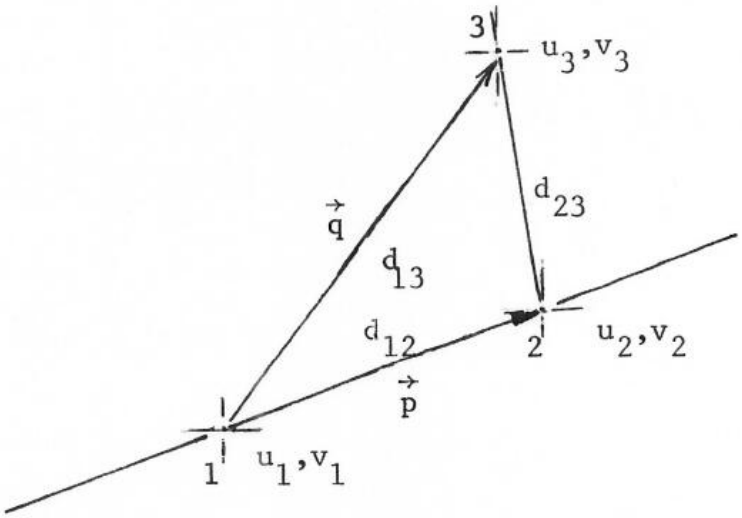


Figura 4.5 - Geometria para a interpretação das funções DSQF e SIDE.

2) <u>Determinação do par de pontos mais próximos e seu ponto médio</u>

São calculadas as distâncias (ao quadrado) entre todos os pontos dados, sem repetir nenhum cálculo. Numa representação matricial (i,j) vemos que só é necessário calcular os elementos estritamente acima da diagonal principal (Figura 4.6).

Adota-se inicialmente $d_{min} = d_{12}$ e, a seguir, para cada

$$
\begin{array}{c|ccccccc|}
i = \rightarrow & 1 & 2 & 3 & 4 & 5 & \dots & n \\
j = \downarrow & & & & & & & \\
1 & 0 & d_{12} & d_{13} & d_{14} & d_{15} & \dots & d_{1n} \\
2 & & 0 & d_{23} & d_{24} & d_{25} & \dots & d_{2n} \\
3 & & & 0 & d_{34} & d_{35} & \dots & d_{3n} \\
4 & & & & 0 & d_{45} & \dots & d_{4n} \\
5 & & & & & 0 & \dots & d_{5n} \\
 & & & & & & & \vdots \\
n-1 & & & & & & & d_{n-1n} \\
n & & & & & & & 0
\end{array}
$$

Propriedades:

$d_{ij} = 0$ (para $i = j$)

$d_{ij} = d_{ji}$ (simetria)

Figura 4.6 - Matriz das distâncias calculadas

distância calculada verifica se é menor que a d_{min} atual. Nesse caso faz a substituição. Quando estiverem varridos todos os pontos dispõe-se da aresta mínima.

Pela média das coordenadas x e y obtém-se as coordenadas do ponto médio da aresta mínima (PM).

3) Reordena os outros (NDP-2) vértices, por ordem crescente de distância ao ponto médio (PM). Armazena o número desses vértices na lista IWP.

   a) reagrupa os vértices na lista IWP, deixando as duas primeiras posições livres para a aresta mínima e mantendo as demais arestas na mesma posição relativa. Calcula também as distâncias de cada ponto a PM, armazenando em WK.

   b) reagrupa os vértices (de 3 a n-1) de acordo, agora, com a distância ao ponto PM. O conteúdo da área de

trabalho WK é perdido pelos sucessivos rearranjos; conserva-se somente a ordenação dos pontos.

4) Se necessário, modifica a ordem dos pontos em IWP de modo que os <u>três primeiros vértices</u> sejam <u>não-colineares</u>.

Realiza o produto vetorial, calcula o seu módulo (função ABS) e verifica se o módulo é maior que zero. Na realidade substitui o valor zero por um $\varepsilon = d_{min} \cdot 10^{-6}$. Isso corresponde a verificar se os pontos estão alinhados (ou quase alinhados).

Caso seja necessário — pontos alinhados — reorganiza a lista IWP para garantir que haja pelo menos um triângulo não degenerado. Caso todos os pontos estejam alinhados a rotina acusa o fato apontado uma mensagem de erro.

5) <u>Forma o primeiro triângulo</u>. Além disso, armazena o número dos vértices desse triângulo na lista IPT e armazena o número de vértices que pertencem a um segmento da fronteira atual e o número do respectivo triângulo na lista IPL.

Utiliza a função SIDE para verificar se a ordenação dos pontos no triângulo é anti-horária. Se não for troca os vértices 1 por 2, o que faz com que a seqüência de armazenamento seja a desejada.

A lista de fronteira, para esse primeiro triângulo, é simples.

6) <u>Acrescenta os (NDP-3) pontos remanescentes, um a um</u>.

É o enlace principal desta rotina, que vai até o endere-

ço 79, na p. 25.

a) <u>determina os segmentos de bordas visíveis</u> a partir do ponto em questão.

É fornecida a lista da fronteira IPL, com NL segmentos, sendo que o ponto em questão, a ser acrescentado, sempre cai fora da circunferência que envolve o polígono formado com os pontos anteriores (basta pensar na circunferência centrada em PM e com raio igual à distância de PM ao último ponto acrescentado).

A determinação dos segmentos visíveis é feita através dos produtos vetoriais, tendo como centro o ponto P_i (em questão). Une-se P_i com os pontos da lista de fronteira, formando-se os vetores $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ (conforme Figura 4.7).

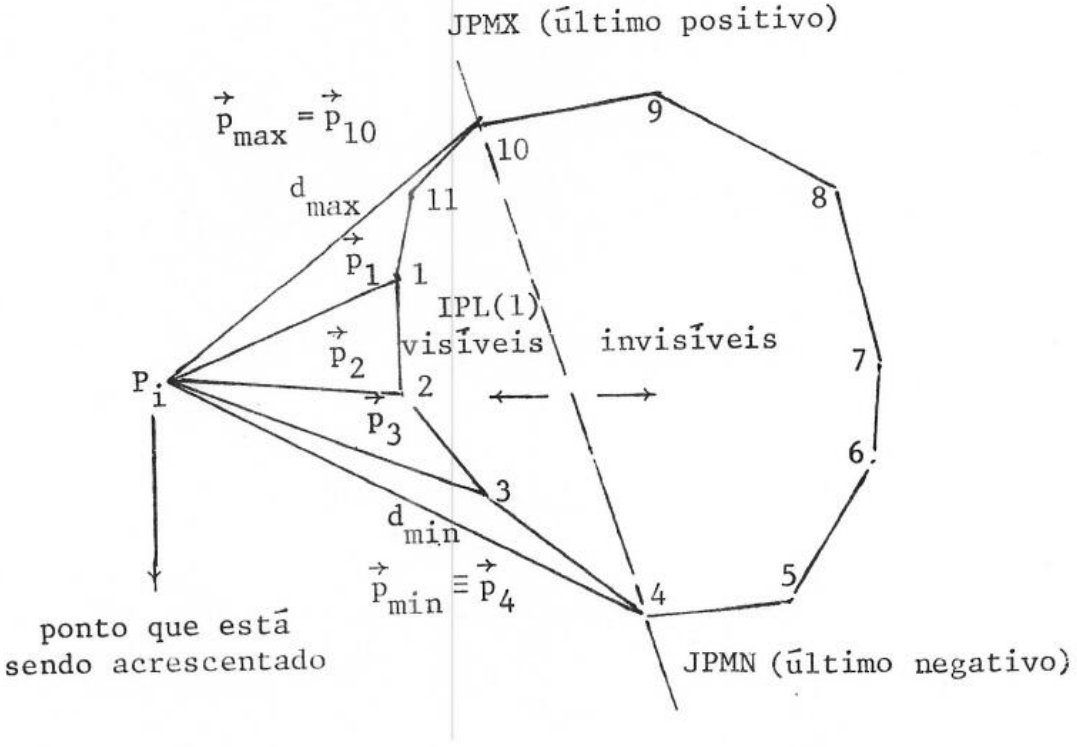


Figura 4.7 - Determinação das arestas visíveis a partir de P_i.

Trata-se pois de encontrar entre os $\vec{p}_i$, o vetor mais à direita de $\vec{p}_1$ (tangente inferior traçada de P_i ao polígono, representada na figura por $\vec{p}_{min} \equiv \vec{p}_4$) e o vetor mais a esquerda de $\vec{p}_1$ (tangente superior ao polígono, representada por $\vec{p}_{max}$).

Isso pode ser feito através do produto vetorial. Inicialmente faz-se $\vec{p}_{min} = \vec{p}_{max} = \vec{p}_1$. Efetua-se a seguir o produto vetorial $\vec{p}_1 \wedge \vec{p}_2$; se der negativo isso significa $\vec{p}_2$ a direita e adota-se então $\vec{p}_2$ como $\vec{p}_{min}$ e prossegue-se analogamente até que o produto vetorial se torne positivo (na figura isso acontece para $\vec{p}_5$). O último vetor que forneceu um produto vetorial negativo ($\vec{p}_4$) é a tangente inferior procurada.

Raciocínio semelhante, agora com os produtos vetoriais positivos, fornece a tangente superior.

Os números dos pontos que fornecem a tangência mínima e máxima são controlados pelos índices JPMN e JPMX. Os segmentos de borda entre esses dois pontos serão visíveis no sentido horário e invisíveis no sentido anti-horário.

Na Figura 4.7, JPMN = 4, JPMX = 10, portanto serão visíveis as seguintes arestas: 4-3, 3-2, 2-1, 1-11 e 11-10; serão invisíveis: 4-5, 5-6, 6-7, 7-8, 8-9 e 9-10.

Esse passo do programa prevê também casos limites, em que o vetor mínimo (ou máximo) esteja apenas a um ε à direita (esquerda) do vetor anterior. Nessa situação examina-se se o módulo desse possível vetor mínimo é maior ou menor que o módulo do vetor anterior, isto é, compara as distâncias P_0P_n e P_0P_{n-1}.

Se a distância for maior não se efetua a troca pois essa união acaba gerando um triângulo com um ângulo muito pequeno (quase nulo). Essa situação não acontece se a distância for menor: o ângulo formado é maior. Nos casos de alinhamento ou quase alinhamento essa medida faz com que se tome sempre o segmento mais curto.

A Figura 4.8 ajuda a ver porque num caso se efetua a união e no outro não.

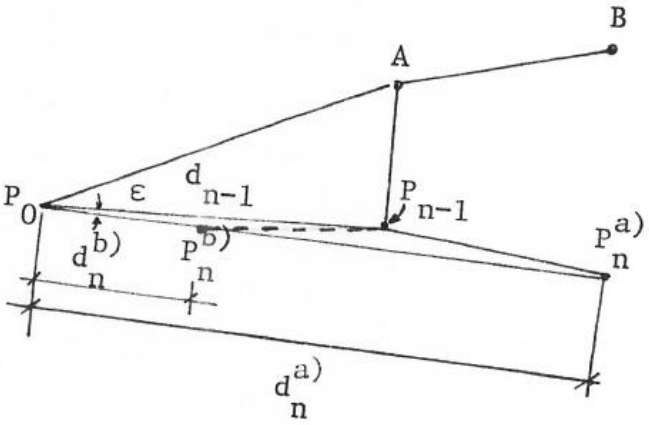


Figura 4.8 - Caso limite em que não se adota o ponto mais à direita: ângulo muito pequeno ε.

b) <u>Roda a lista IPL</u> para que as bordas invisíveis ocupem a primeira parte desta lista.

Para isto efetua-se um giro de pontos e coordenadas até que o ponto seguinte a JPMN ocupe a primeira posição. Computacionalmente é necessário copiar os valores das primeiras posições de IPL (correspondentes aos pontos de 1 a JPMN + 1) no fim da lista para que não sejam perdidos ao gravarmos outros valores nessa posição. A seguir deslocam-se todos os valores em bloco, até que a primeira posição seja ocupada por JPMN+1.

c) Acrescenta triângulos à lista correspondente (IPT), atualiza os segmentos de borda na lista IPL e envia para a lista IWL os segmentos de borda que necessitam ser reexaminados.

c-1) Acrescenta à lista IPT um dos triângulos formado pela união do ponto em questão com dois vértices visíveis (Figura 4.9).

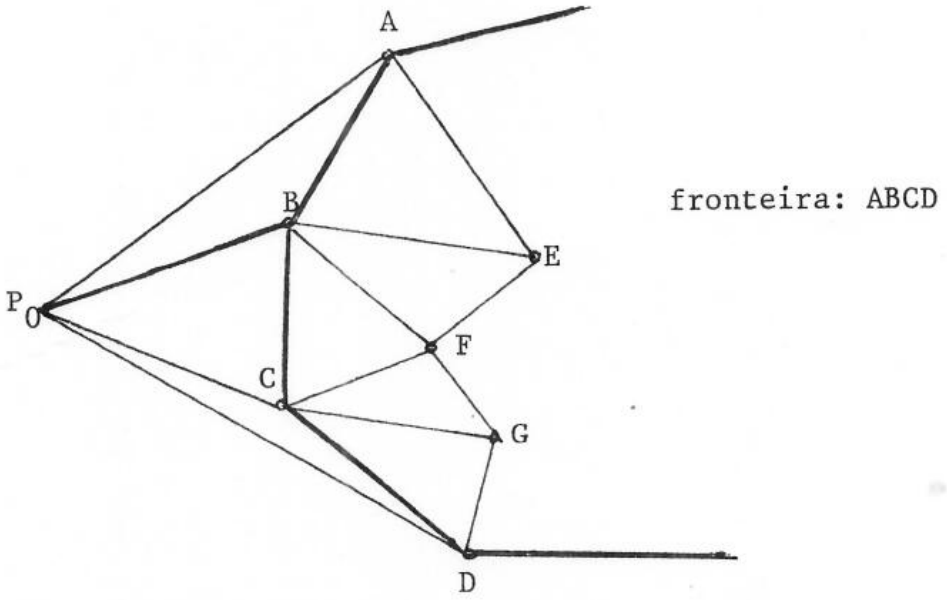


Figura 4.9 - Triângulos formados pela união de P_0 com os vértices visíveis; possíveis trocas, em torno das diagonais que são as arestas da borda, deverão ser testadas.

Na primeira passagem acrescenta-se o triângulo P_0AB, na segunda P_0BC e assim por diante.

c-2) Atualiza a lista IPL (borda) para introduzir as modificações devidas ao acréscimo de P_0. Na situação da Figura 4.9, seria necessário incluir AP_0 e P_0B na lista de fronteira (sentido anti-horário) e suprimir AB. Após a última passagem AP_0 e P_0D terão

substituído os segmentos AB, BC, CD.

c-3) Determina o terceiro vértice que não cai na fronteira, isto é, dada uma aresta de borda e o correspondente triângulo (interior à fronteira) determina o terceiro vértice desse triângulo. Ainda na Figura 4.9, dada a aresta AB, determina o ponto E, interno e que forma o triângulo ABE. Nas passagens seguintes determina os pontos F e G.

c-4) Verifica se uma troca de triângulos é necessária.

Isto é feito com os quadriláteros que se formam com a introdução de P_0. A formação dessa figura é sempre P_0, uma aresta visível (AB, por exemplo) e o terceiro ponto do triângulo (E). O teste de troca é feito pela função IDXCHG em torno da aresta atualmente unida que é o segmento de borda AB.

*) Se a mudança não for necessária, repete o ciclo com o triângulo seguinte (P_0BC). Depois do último triângulo acrescenta-se um novo ponto exterior ao polígono recém formado. O ciclo se repete até que o último ponto seja acrescentado.

*) Se a mudança for necessária modifica a lista IPT para gravar os novos triângulos e suprimir os antigos.

c-5) armazena aresta em IWL.

Ao ser introduzida uma nova aresta, esta deve ir para a lista IWL já que deve ser testada contra outras possíveis mudanças. Isto será feito no reexame da lista IWL, item d-1) abaixo.

d) Melhora a triangulação.

d-1) Localiza na lista IPT dois triângulos de ambos os lados de uma aresta armazenada em IWL, para reexame. Para isso percorre a lista IPT verificando os triângulos que possuem essa aresta como um dos seus lados. Serão encontrados 2 triângulos.

d-2) determina o terceiro vértice de cada um desses dois triângulos, isto é, o vértice que não pertence à aresta comum.

d-3) verifica se é necessária uma troca de triângulos em torno a essa aresta comum, utilizando a função IDXCHG.

Em caso afirmativo rearranja a lista IPT,assinalando novas arestas a serem testadas, ou seja, organiza IWL para uma nova rodada de testes. Quando a lista IWL estiver vazia,a triangulação termina.

7) Reagrupa a lista IPT

Nesta altura do programa, todos os triângulos já estão formados, restando somente ordená-los adequadamente:listá-los no sentido anti-horário.

Recorre-se então, novamente, à função SIDE que é aplicada a cada triângulo para verificar se seus vértices estão listados no sentido desejado.

4.3.3. SUB-ROTINA IDCLDP (pp. 6-7)

Esta sub-rotina seleciona alguns pontos que são os mais

próximos de cada vértice.

Os parâmetros de entrada são:

* NDP = número de vértices fornecidos.
* XD, YD = listas de dimensão NDP contendo as coordenadas x e y dos pontos dados.
* NCP = número de pontos mais próximos de cada vértice (vizinhos).

Os parâmetros de saída são:

* IPC = matriz de inteiros de dimensões NCP*NDP, onde o número dos NCP pontos mais próximos de cada vértice dado são armazenados.

Impõe-se a restrição de que NCP seja menor ou igual a 25. Parece mais do que suficiente.

Seguindo a listagem do programa fazemos as seguintes obs.

1) <u>Seleciona NCP pontos</u> para cada NDP, que são os mais próximos.

Tomando um vértice de cada vez, calcula os NCP vizinhos mais próximos de cada um.

1-a) Seleciona NCP pontos, sem a preocupação de distância.

Suponhamos, para exemplificação e por comodidade que NCP seja igual a 4: queremos os 4 vizinhos mais próximos.

Toma-se inicialmente o primeiro ponto que vem após o ponto em questão (P_0) na lista de coordenadas (P_0+1). A distância entre ambos é tomada como sendo dsqmax (distância máxima).

Está formado um círculo, com centro em P_0 e raio dsqmax.

Tomam-se a seguir os NCP-1 pontos seguintes da lista, no caso 3. Se esses pontos cairem dentro do círculo, este passo está terminado. Se, durante a seqüência de operações um ponto cair fora do círculo, amplia-se o seu raio para que esse ponto seja incluído exatamente, isto é, adota-se como raio dsqmax a distância desse ponto ao centro do círculo.

Este passo termina quando temos exatamente 4 pontos dentro de um círculo de raio igual à distância do ponto mais afastado ao ponto P_0 (Figura 4.10).

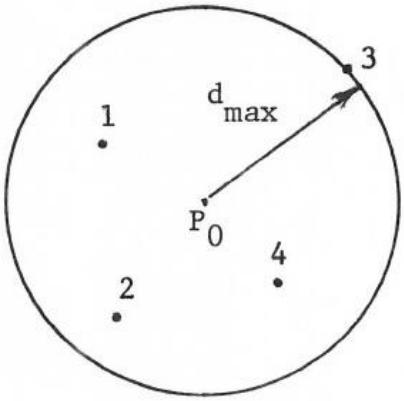


Figura 4.10 - Círculo contendo os 4 pontos seguintes a P_0 na lista de coordenadas. Seu raio é igual à distância do ponto mais afastado a P_0.

2) <u>Verifica se</u> esses NCP pontos <u>são os mais próximos</u>.

Neste passo estudam-se os pontos restantes (NDP-NCP), que ainda não foram examinados. Os pontos que cairem fora do círculo anteriormente construído são desconsiderados por, forçosamente, não pertencem aos NCP mais próximos. Quando um ponto cai dentro do círculo introduz-se esse ponto na relação dos NCP, eliminando-se o ponto mais afastado, que é aquele que deu

origem a dsqmax.

2-a) Determina-se, dentre os NCP-1 pontos e o novo ponto, qual o mais afastado de P_0. Sua distância a esse centro, já calculada no passo anterior (para ver se caía ou não dentro do círculo ou nos cálculos iniciais) passa a ser o novo dsqmax.

No final deste passo, depois que foram examinados todos os pontos, dispõe-se dos NCP pontos mais próximos do vértice em estudo.

3) Verifica se todos os NCP pontos e o ponto em questão são colineares.

Isto é feito através do produto vetorial. Deve haver pelo menos um desses produtos que seja não nulo. Isso acontecendo não é necessário prosseguir.

Caso todos os pontos sejam colineares passa-se à busca do próximo ponto não alinhado.

4) Procura o ponto mais próximo, não colinear.

Essa busca é feita pelo exame do valor de sucessivos produtos vetoriais, que devem ser diferentes de zero para indicar o não alinhamento. Dentre os produtos não nulos toma-se aquele cujo ponto originante for mais próximo de P_0.

Se todos os pontos forem colineares o programa emite essa mensagem e pára, já que essa configuração não permite o cálculo das derivadas.

5) Preparação dos valores de saída.

No fim de cada passo transferem-se esses NCP pontos,não alinhados para a lista definitiva de saída e prossegue-se com os cálculos para determinar os vizinhos do próximo vértice.

4.3.4. SUB-ROTINA IDGRID (pp. 7 a 10)

(Chamada somente por IDSFFT)

Esta sub-rotina organiza os pontos da malha para o ajuste de uma superfície, escolhendo-os por ordem crescente do número de triângulos e do número dos segmentos de borda.

Os parâmetros a serem fornecidos são:

* XD, YD = listas de dimensão NDP contendo as coordenadas x e y dos vértices.
* NT = número de triângulos.
* IPT = listas de inteiros contendo o número de vértices dos triângulos. Dimensão: 3*NT.
* NL = Número de arestas de borda.
* IPL = lista inteira de dimensão 3*NL contendo o número dos vértices iniciais e finais de cada aresta de borda e o número do respectivo triângulo.
* NXI = número de pontos da malha na direção x.
* NYI = número de pontos da malha na direção y.
* XI, YI = matriz de dimensão NXI e NYI contendo as coordenadas x e y dos nós da malha.

Os parâmetros de saída são:

* NGP = lista de inteiros de dimensão 2*(NT+2*NL) onde é armazenada a quantidade de nós da malha que pertencem a cada triângulo (ou aresta de borda). Por exemplo: o triângulo 1 tem 5 nós da malha em seu interior, o triângulo 2 tem 6, etc.

* IGP = lista inteira de dimensão NXI*NYI onde são armazenados os números dos pontos da rede seguindo uma ordem crescente dos triângulos (ou segmentos de borda). Ou seja, listam-se os nós pertencentes a cada triângulo (borda).

Seguindo a listagem temos:

1) Significado das funções SIDE e SPDT.

Conforme visto anteriormente, SIDE pode ser um produto vetorial ou pode indicar a posição de um ponto com relação a uma reta. Nesta sub-rotina utiliza-se a segunda interpretação, com uma variante: indica em que semi-plano um ponto cai.

Na Figura 4.11, SIDE positiva significa que o ponto x está no semi-plano da esquerda com relação a 1-2. SIDE negativa significa o semi-plano da direita. SIDE nula significa alinhamento.

A função SPDT (Scalar ProDucT) pode ser entendida como sendo o produto escalar, que também divide o plano em duas regiões, só que neste caso perpendicularmente à reta 1-2 (Figura 4.11).

Combinando as duas funções pode-se separar o plano em 4 quadrantes, o que, na realidade já seria previsível tendo em

conta que correspondem às funções trigonométricas de sen e cos, a menos de uma constante (p.q.) que é o produto dos módulos dos vetores.

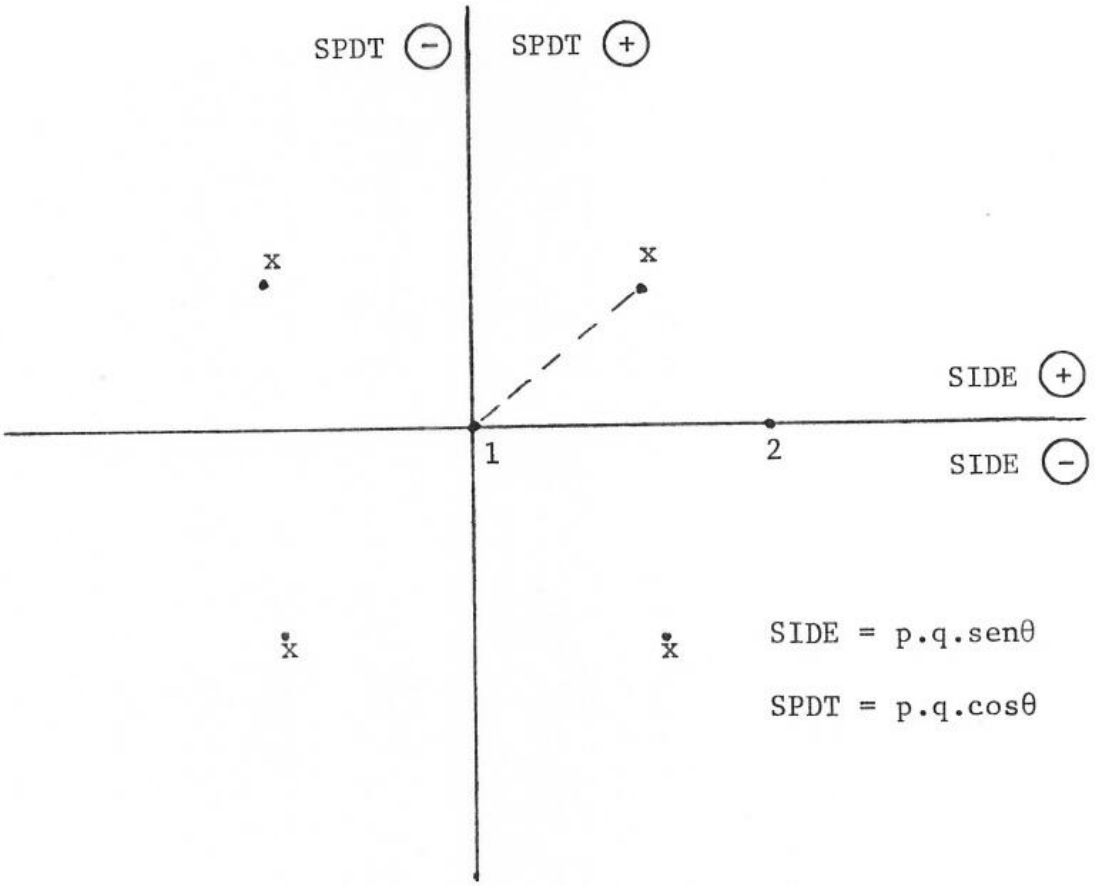


Figura 4.11 - Interpretação geométrica das funções SIDE e SPDT, aplicadas aos pontos 1, 2, x

2) Determinação dos pontos da malha que caem dentro da área , isto é, internos à fronteira.

2,a) Dado um triângulo, determina as coordenadas máximas e mínimas de seus vértices, o que corresponde a associar um retângulo a esse triângulo (Figura 4.12).

2,b) Faz uma exploração da malha caminhando na direção x determinando os índices (ix_{min} e ix_{max}) dos possíveis pontos da malha que cairão dentro do triângulo. Essa primeira comparação será feita com o retângulo associa-

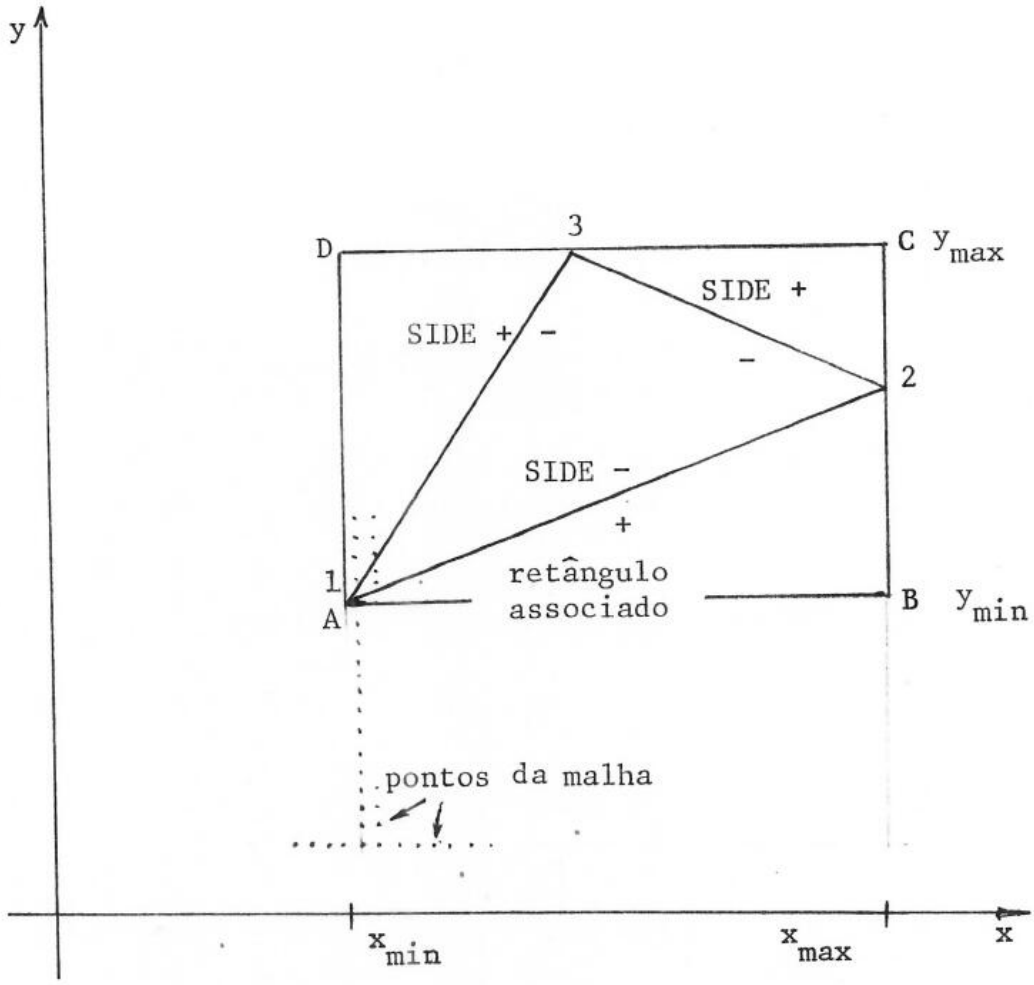


Figura 4.12 - Retângulo associado a um triângulo, para facitar a exploração dos pontos da malha.

do e por isso os pontos de malha devem ter coordenadas $\underline{x}$ que estejam entre x_{min} e x_{max}.

2,c) Explora a malha, agora na direção y, para determinar todos os pontos da malha que caem dentro do retângulo associado (ABCD).

2,d) Para cada ponto que cai dentro do retângulo verifica se cai dentro do triângulo 123. Esse exame realiza-se através da função SIDE, que deve dar negativa nos três testes com relação aos 3 lados do triângulo. Basta um deles ser positivo para que se sai-

ba que o ponto cai fora do triângulo.

Se o ponto cair sobre um dos lados do triângulo, também será armazenado na lista IGP, mas em uma posição especial: a partir do fim da lista e trabalhando com índices decrescentes. Esse cuidado evitará a duplicação de cálculos já que pontos desse tipo pertencem a dois triângulos.

2,e) Armazenamento.

A seguir, caso o ponto caia dentro do triângulo, armazena-o na lista IGP, e incrementa de 1 o NGP (número de nós da malha, por triângulo) correspondente a esse triângulo.

3) Determina os pontos da malha que caem fora da área.

A fronteira é constituída por um polígono convexo e as áreas externas serão associadas às arestas de acordo com o es-

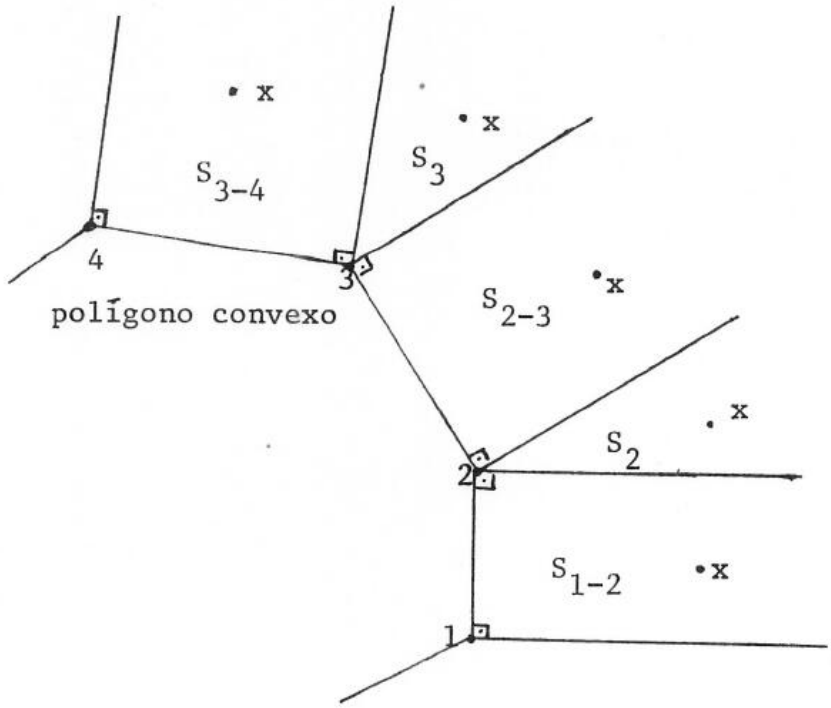


Figura 4.13 - Representação dos 2 tipos de áreas externas ao polígono de fronteira, às quais se associam os pontos externos.

quema teórico : existem dois tipos de áreas a serem consideradas, (1) – áreas semi-infinitas retangulares como S_{2-3} na Figura 4.13 ou (2) – áreas semi-infinitas triangulares, como as indicadas por S_2 ou S_3, nessa mesma figura.

3.1) Áreas semi-infinitas retangulares.

a) Determina regiões retangulares no plano para restringir a busca. Cada retângulo é determinado pelos limites absolutos da malha e por um x (max ou min) e um y (max ou min) em função da posição espacial da aresta em questão. Para um ponto da malha, cair dentro do retângulo restritivo é condição, necessária mas não su-

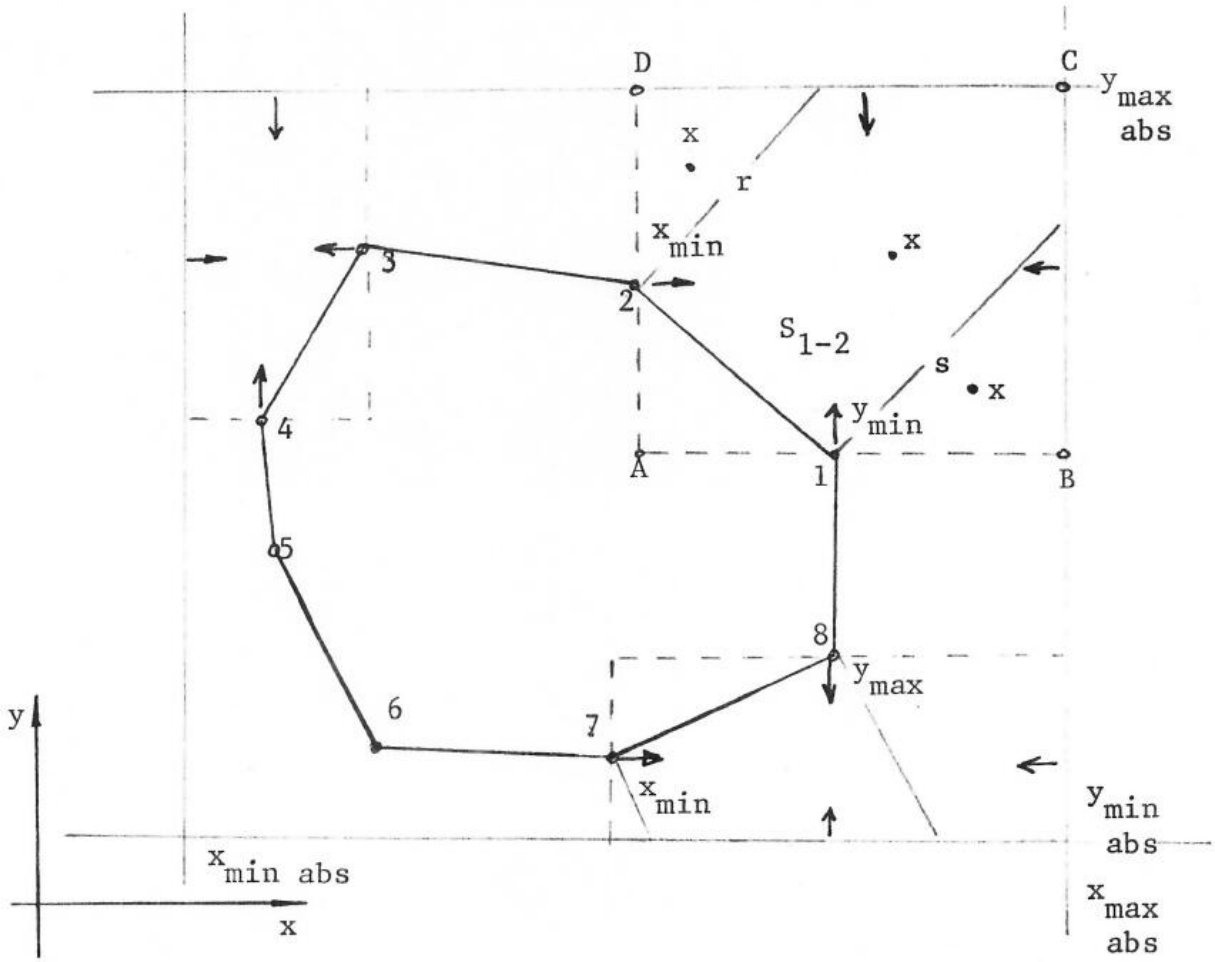


Figura 4.14 - Restrição das regiões de busca em função das coordenadas máxima e mínima da aresta em questão. Associa-se um retângulo a cada aresta, por exemplo, ABCD para a aresta 1-2.

ficiente, para que esteja associado a uma aresta.

A Figura 4.14 ajuda a esclarecer a construção acima descrita.

b) Varre o retângulo restrito de busca, segundo x, determinando os índices x_{max} e x_{min} dos pontos que caem dentro do retângulo de busca.

c) a seguir varre na direção y, determinando cada ponto da malha que cae dentro do quadrilátero de busca e examinando se o ponto cai dentro da área retangular semi-infinita, por exemplo, S_{1-2} na Figura 4.14, associada à aresta 1-2.

d) para esse teste utilizam-se as funções SIDE (uma vez) e SPDT (duas vezes). Na Figura 4.14, o ponto $\underline{x}$ deve estar à direita de r (teste SPDT com 1,2 e x); à esquerda de s (teste SPDT com 2,1 e x) e à direita de 1-2 (teste SIDE com 1, 2 e x).

Os pontos que caírem dentro são armazenados convenientemente na lista IGP após o último triângulo, ao mesmo tempo que se vai contabilizando a quantidade de pontos que cai dentro de cada área semi-infinita retangular.

3.2) <u>Áreas semi-infinitas triangulares</u>.

a) Como no caso anterior, delimita inicialmente regiões menores de busca, através de $x_{min/max}, y_{min/max}$ ou combinações. Esses valores correspondem às coordenadas do ponto em questão ao qual está associada a área triangular.

Biblioteca da Escola Politecnica
São Paulo

A região restrita corresponde a um retângulo cujos outros limitantes são os x e y máximos e mínimos absolutos (da malha).

Na Figura 4.15 indicam-se os retângulos limitantes, em função da posição do ponto em questão.

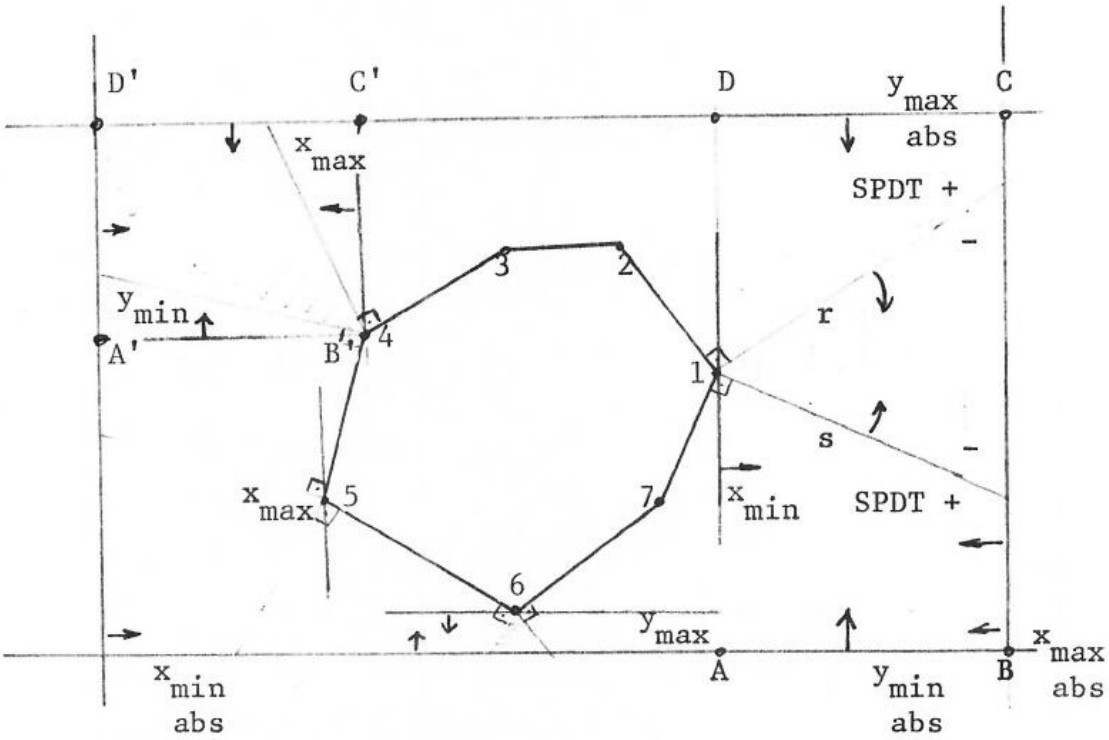


Figura 4.15 - Delimitação de áreas menores para a busca dos pontos da malha que caem em regiões triangulares semi-infinitas associadas a pontos (encontro de duas arestas).

b), c) e d) <u>Localização dos pontos de malha</u>

Analogamente ao caso anterior, varre a malha em x (passo b), em y (passo c) e verifica se o ponto está na área triangular semi-infinita (passo d).

Essa última operação se faz utilizando SPDT. Na Figura 4.15, um ponto x deve estar à direita de r (testando SPDT com 2,1 e x deve ser negativo) e à esquerda de s (testando SPDT com

BIBLIOTECA ESCOLA POLITÉCNICA - USP

7,1 e x deve dar negativo). Se uma das condições falhar o ponto x estará fora do triângulo.

Os pontos que estiverem dentro dessa área são armazenados em IGP e contabiliza-se o número de pontos associados ao vértice.

4.3.5. SUB-ROTINA IDLCTN (pp. 11 a 13)

(Chamada somente por IDBVIP)

Esta sub-rotina localiza um ponto, isto é, determina a que triângulo um dado ponto pertence. Quando o ponto fornecido não cai dentro da área, esta sub-rotina determina: um segmento de borda quando o ponto cai numa área retangular semi-infinita, ou dois segmentos de borda quando o ponto cai numa área triangular semi-infinita (conforme Figura 4.13).

Os parâmetros a serem fornecidos são:

* NDP = número de vértices da triangulação (pontos dados).
* XD, YD = listas de dimensão NDP, contendo as coordenadas x e y dos vértices.
* NT = número de triângulos.
* IPT = lista de inteiros, de dimensão 3*NT, contendo os números dos vértices dos triângulos.
* NL = número de arestas de borda.
* IPL = lista de inteiros de dimensão 3*NL contendo: o número dos pontos extremos das arestas e o número do respectivo triângulo.

* XII, YII = coordenadas do ponto a ser localizado.

Os parâmetros de saída são:

* ITI = número do triângulo, quando o ponto cai dentro da área ou dois segmentos de borda IL1 e IL2 codificados como sendo ITI = IL1*(NT+NL)+IL2 quando o ponto cai fora da área.

Os outros parâmetros são:

* IWK = lista de inteiros de dimensão 18*NDP usada internamente como área de trabalho.
* WK = lista de dimensão 8*NDP usada internamente como área de trabalho.

Acompanhando a listagem do programa fazemos os seguintes comentários:

1) Significado das funções SIDE e SPDT.

Já esclarecido anteriormente (item 4.3.4): possibilita dividir o plano em 4 quadrantes: 2 semi-planos, etc.

2) Divide o plano x, y em nove secções retangulares.

a) determina os 4 limitantes gerais que formam o retângulo continente, isto é, x_{min}, x_{max}, y_{min}, y_{max}. Para isso compara as coordenadas x e y de todos os pontos fornecidos.

b) Determina quatro linhas que fracionam o retângulo em 9 sub-regiões. Para isso calcula as distâncias referentes aos terços, em x e y. A Figura 4.16 representa essa situação.

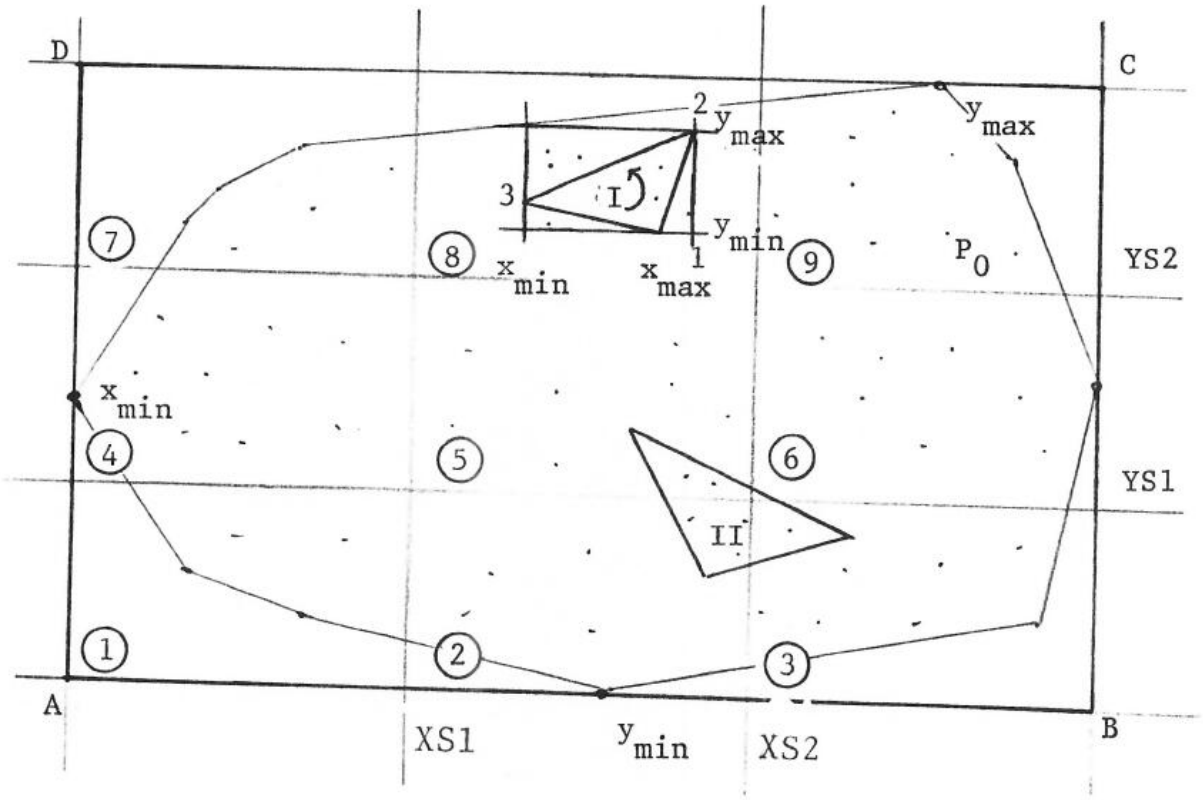


Figura 4.16 - Divisão do retângulo ABCD (continente) em 9 sub-regiões de busca (secções).

3) Determina e armazena na lista IWK o número dos triângulos associados a cada uma das 9 secções.

a) Determina os máximos e mínimos x e y para cada triângulo, associando a cada um, um retângulo envolvente (por exemplo o triângulo I, na figura 4.16, região 8).

b) Determina as regiões (de 1 a 9) que possuem superfície em comum com o triângulo examinado, isto é, regiões que são interceptadas por qualquer uma das arestas (lados) do triângulo. Por exemplo, o triângulo II da Figura 4.16 está associado às regiões 2, 3, 5 e 6.

Para cada região (i) interceptada (associada) a um triângulo faz-se IDSC(i) = 1; caso contrário não se altera o va-

lor zero definido anteriormente.

c) Examina as 9 regiões para ver a quais o triângulo está associado. Quando para uma região o resultado for positivo armazena-se na lista IWK o número do triângulo. Isso deve ser feito para todas as regiões e por isso monta-se uma adequada estrutura de dados: a cada triângulo corresponderão 9 posições de memória,na lista IWK. Para o primeiro triângulo as posições de 1 a 9; para o segundo as posições de 10 a 18; para o terceiro, de 19 a 27 e assim por diante. Se o primeiro triângulo estiver associado, por exemplo, às regiões 1, 2, 5, o seu número aparecerá nas seguintes posições de IWK: 1, 2 e 5. Se o triângulo 2 pertencesse a essas mesmas regiões apareceria nas posições 10 (1+9), 11 (2+9) e 14 (5+9). E assim por diante.

4) Armazena na lista WK os valores máximos e mínimos das coordenadas x e y, de cada triângulo.

Corresponde a associar um retângulo a cada triângulo.

OBSERVAÇÃO. Desse passo do programa deve-se pular para o endereço 110, devido à seqüência lógica imposta. Seguiremos essa seqüência.

5) <u>Localiza um ponto da malha</u> (x_0, y_0).

5.1) Determina a região (de 1 a 9) em que o ponto cai.

Para isso compara suas coordenadas com XS1 e XS2 (x) e com YS1 e YS2 (y). Através de engenhoso procedimento com a variável ISC, faz com que esta indique (seja igual) à região em

que o ponto cai.

Suponhamos que o ponto esteja na região 9 (P_0, na Figura 4.16). O processo inicia-se com ISC = 1. Comparando x_0 com XS1 vê-se que é maior e então adiciona-se uma unidade a ISC, que passa a ser ISC = 2. Compara-se a seguir x_0 com XS2 e como é maior soma-se uma unidade a ISC que passa a ser ISC = 3. A seguir compara-se y_0 com YS1; como é maior soma-se 3 a ISC, que passa a ser ISC = 6. Finalmente compara-se y_0 com YS2 e como também é maior soma-se 3 a ISC que passa a ser, definitivamente, ISC = 9 e portanto o ponto pertence à região de número 9.

5.2) Verifica em que triângulo o ponto cai examinando somente os triângulos associados àquela secção.

a) Se não houver nenhum triângulo associado à região em questão (NTSCI = 0) isto significa que o ponto cai fora da fronteira do polígono convexo formado pelos vértices e deve-se procurar fora da região. O programa faz um desvio para o endereço 130 (passo 7, abaixo).

b) A busca realiza-se em todos os triângulos associados, de 1 a NTSCI (número de triângulos da secção).

c) Verifica inicialmente se o ponto cai dentro do retângulo envolvente definido anteriormente (passo 3,a). Se cair fora vai para outro retângulo.

d) Caso esteja dentro do retângulo envolvente verifica se está dentro de algum dos triângulos associados a essa área retangular.

Para verificar se o ponto cai dentro do triângulo utili-

za-se a função SIDE 3 vezes. Se o ponto ocupar os semi-planos corretos nos três testes, estará dentro do triângulo: conforme Figura 4.12.

6) Armazena o número do triângulo em que o ponto cai, que é o único parâmetro de saída. A sub-rotina está terminada, se o ponto caiu dentro da área em questão. Se o ponto tiver caído fora do polígono convexo (conforme 5.2, a), acima) o procedimento é o descrito a seguir.

7) <u>Exame dos pontos que caem fora da fronteira convexa</u>.

Para isso deve-se percorrer as arestas da fronteira, de 1 a NL (todas), realizando alguns testes baseados nas funções SIDE e SPDT.

a) Se testando uma aresta, por exemplo a 1-2 na Figura 4.17, os testes derem: SPDT-1(+); SPDT-2 (+) e SIDE (-), o ponto situa-se na área retangular semi-infinita associada à aresta 1-2. Faz-se então IL2 = IL1, e constroi-se o número de código para essa situação: ITO = = IL1 * NTL + IL2, com NTL = NT + NL .

b) Caso o segundo teste (SPDT) seja negativo, após um primeiro teste (SPDT) positivo, devia-se para o comando número 140 que verificará a seguir se o ponto cai na área triangular semi-infinita com vértice em 2 (Figura 4.17).

Para isso toma-se a aresta seguinte, através da função MOD e verifica se SPDT dá um valor negativo. Nesse caso o pon-

to estará na área triangular e então passa-se ao endereço 160 para construir o número de código: ITO = IL1 * NTL + IL2, e o programa está encerrado.

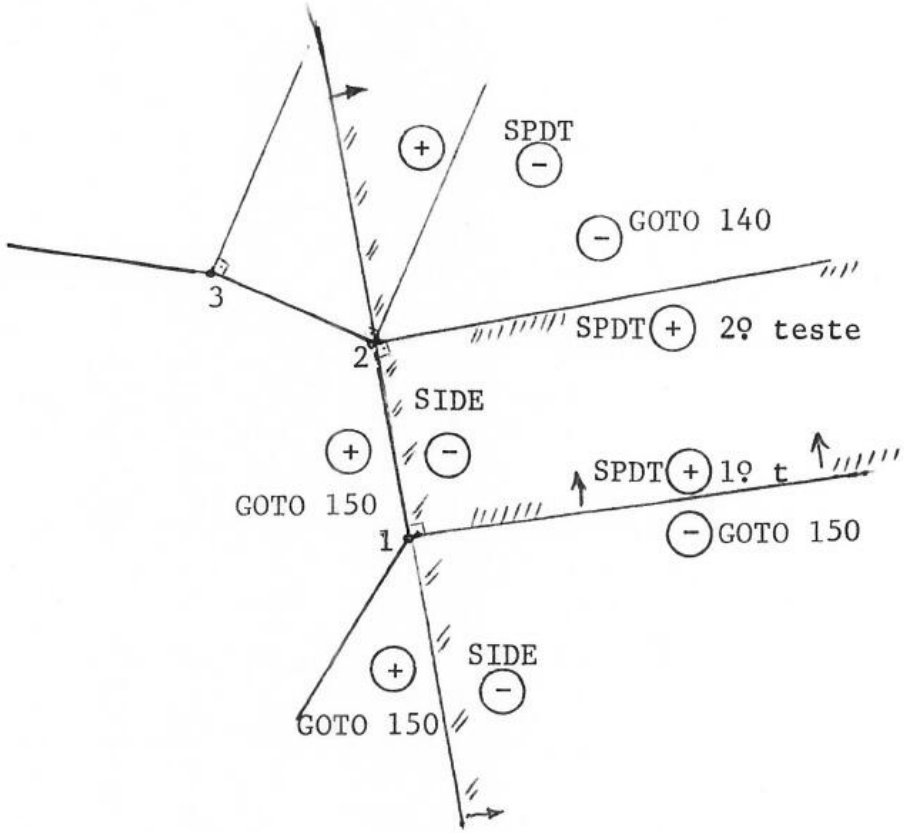


Figura 4.17 - Localização de um ponto externo ao polígono.

Se SPDT for positiva passa-se ao exame de outra aresta, e assim por diante.

OBSERVAÇÃO. As sub-rotinas seguintes que o necessitam encarregam-se de decodificar ITO. Se ITO for menor que NTO isso significa que o ponto cai no triângulo ITO, dentro da área. Se ITO for maior que NTO realiza-se mais um teste: faz IL1 = ITO/NTL e IL2 = ITO - IL1 * NTL. Se IL1 = IL2 significa que o ponto cai numa área retangular. Se for diferente, área triangular.

8) Depois da primeira passada verifica se o novo ponto

cai na mesma região que o anterior: triângulo ou área externa (retangular ou triangular).

Realiza-se, para essa operação, a decodificação de ITO conforme o procedimento descrito acima.

4.3.6. SUB-ROTINA IDPDRV (pp. 13 a 15)

Esta sub-rotina estima as derivadas parciais de primeira e segunda ordem nos vértices fornecidos.

Os parâmetros a serem fornecidos são:

* NDP = número de vértices.
* XD, YD, ZD = listas de dimensão NDP contendo as coordenadas x, y e z dos vértices.
* NCP = número de pontos adicionais (vizinhos) utilizados para estimar as derivadas parciais em cada ponto.
* IPC = lista de inteiros de dimensão NCP * NDP contendo o número dos NCP pontos mais próximos de cada um dos NDP vértices.

Os parâmetros de saída são:

* PD = lista de dimensão 5*NDP, onde são armazenados os valores estimados de z_x, z_y, z_{xx}, z_{xy} e z_{yy}, nessa ordem.

De acordo com a teoria (desenvolvida no item 3.5) o cálculo das derivadas é feito através da soma de produtos vetoriais. Para isso toma-se como centro o ponto P_0 e constrõem-se

os vetores que ligam esse ponto aos seus n_c vizinhos: $\vec{p}_1$, $\vec{p}_2$, $\vec{p}_3, \ldots, \vec{p}_{nc}$.

A seguir efetuam-se todos os possíveis produtos vetoriais, combinando dois a dois os $\vec{p}_i$ vetores. A ordem dos vetores em cada produto deve ser tal que a resultante seja sempre positiva, isto é, a componente em z deve ser não negativa.

Efetua-se a soma de todos os vetores e assume-se que as correspondentes inclinações do plano perpendicular ao vetor soma são as derivadas na direção x e y.

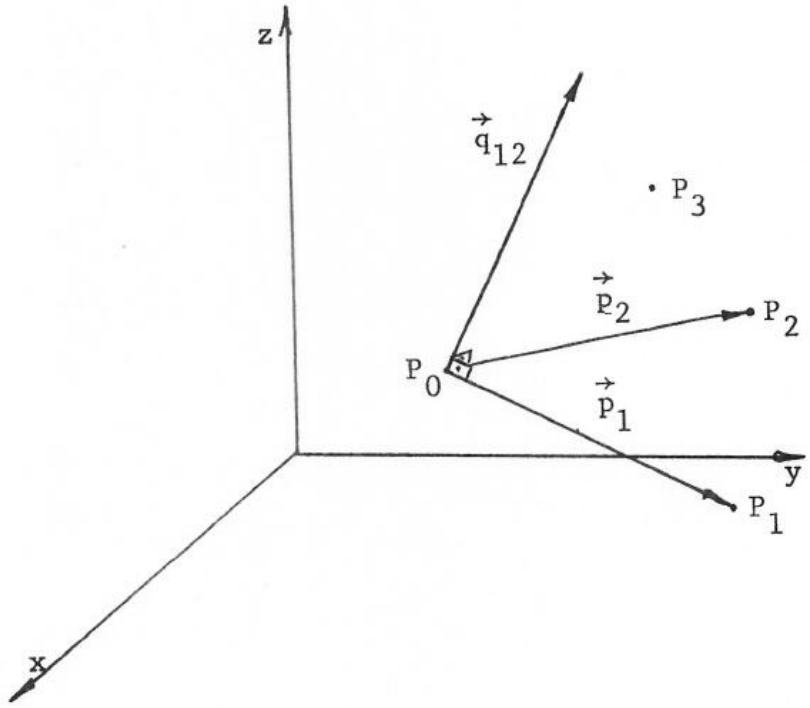


Figura 4.18 - Produto vetorial

A representação gráfica e as notações podem ser as seguintes:

$$p_1 = \Delta x_1 \cdot \vec{i} + \Delta y_1 \cdot \vec{j} + \Delta z_1 \cdot \vec{k}$$

$$\vec{p}_2 = \Delta x_2 \vec{i} + \Delta y_2 \vec{j} + \Delta z_2 \vec{k}$$

$$\Delta x_1 = x_1 - x_0 \qquad \Delta x_2 = x_2 - x_0$$

$$\Delta y_1 = y_1 - y_0$$

$$\Delta z_1 = z_1 - z_0$$

$$\vec{q}_{12} = \vec{q}_i = \vec{p}_1 \wedge \vec{p}_2$$

$$\vec{q}_i = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \Delta x_2 & \Delta y_2 & \Delta z_2 \end{vmatrix}$$

ou

$$\vec{q}_i = \underbrace{\begin{vmatrix} \Delta y_1 & \Delta z_1 \\ \Delta y_2 & \Delta z_2 \end{vmatrix}}_{\alpha_i} \vec{i} \underbrace{- \begin{vmatrix} \Delta x_1 & \Delta z_1 \\ \Delta x_2 & \Delta z_2 \end{vmatrix}}_{\beta_i} \vec{j} + \underbrace{\begin{vmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \end{vmatrix}}_{\gamma_i} \vec{k}$$

Assim sendo, o vetor $\vec{q}_i$ pode ser representado por:

$$\vec{q}_i = \alpha_i \vec{i} + \beta_i \vec{j} + \gamma_i \vec{k} \qquad \gamma_i \geq 0$$

E o vetor $\vec{q}$, somatória de $\vec{q}_i$ fica sendo:

$$\vec{q} = \underbrace{(\sum \alpha_i)}_{a} \vec{i} + \underbrace{(\sum \beta_i)}_{b} \vec{j} + \underbrace{(\sum \gamma_i)}_{c} \vec{k} \qquad (1)$$

$$\vec{q} = a\vec{i} + b\vec{j} + c\vec{k} \qquad (2)$$

Como observação, devemos ter $\gamma_i \geq 0$; caso contrário inver-

tem-se os sinais dos 3 coeficientes α_i, β_i e γ_i pois $(\vec{p}_1 \wedge \vec{p}_2) = -(\vec{p}_2 \wedge \vec{p}_1)$.

De (2) podemos determinar a equação do plano perpendicular a $\vec{q}$ e que é a seguinte:

$$a \cdot (x - x_0) + b \cdot (y - y_0) + c \cdot (z - z_0) = 0$$

ou

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \tag{3}$$

De (3) tiramos:

$$z = -\frac{a}{c} \cdot x - \frac{b}{c} \cdot y - \frac{d}{c},$$

donde

$$\underbrace{z_x}_{\frac{\partial z}{\partial x}} = -a/c \qquad \underbrace{z_y}_{\frac{\partial z}{\partial y}} = -b/c$$

que são as derivadas procuradas.

$$z_x = \frac{-\sum \alpha_i}{\sum \gamma_i} \tag{4}$$

$$z_y = \frac{-\sum \beta_i}{\sum \gamma_i} \tag{5}$$

Para o cálculo das derivadas segundas, os Δx_i e Δy_i mantém-se os mesmos e portanto os γ_i não variam.

Os Δz passam a ser Δz_x e Δz_y, originando, em 2 ciclos de cálculos: z_{xx}, z_{xy}, z_{yy} e z_{yx}.

Teoricamente $z_{xy} = z_{yx}$; na prática, por utilizarmos diferenças finitas são diferentes e então toma-se a média.

Tendo em conta esse desenvolvimento podemos entender facilmente os passos do programa.

1) Estimativa de z_x e de z_y

No enlace externo (DO 24) toma-se cada ponto e seus NCP vizinhos; no enlace intermediário (DO 23) realizam-se os produtos vetoriais e no enlace interno (DO 22) somam-se as componentes.

No programa temos as seguintes variáveis correspondentes à equação (1) acima:

$DNMX = \alpha_i$

$DNMY = \beta_i$ (são as componentes de um produto vetorial genérico)

$DNMZ = \gamma_i$

Temos também:

$NMX = \sum \alpha_i$

$NMY = \sum \beta_i$ (são as componentes do vetor soma).

$NMZ = \sum \gamma_i$

Das equações (4) e (5) resulta:

$z_x = -NMX/NMZ$ (armazenada em: PD (JPDO-4))

$z_y = -NMY/NMZ$ (armazenada em: PD (JPDO-3)).

Um exemplo prático pode ajudar a compreender melhor as operações a serem realizadas. Suponhamos um ponto com 4 vizi-

nhos. Surgem imediatamente 4 vetores: $\vec{p}_1$, $\vec{p}_2$, $\vec{p}_3$, $\vec{p}_4$. Devem ser realizados 6 produtos vetoriais:

$$\vec{q}_1 = \vec{p}_1 \wedge \vec{p}_2 \qquad \vec{q}_4 = \vec{p}_2 \wedge \vec{p}_3 \qquad \vec{q}_6 = \vec{p}_3 \wedge \vec{q}_4$$
$$\vec{q}_2 = \vec{p}_1 \wedge \vec{p}_3 \qquad \vec{q}_5 = \vec{p}_2 \wedge \vec{p}_4$$
$$\vec{q}_3 = \vec{p}_1 \wedge \vec{p}_4$$

Se a posição relativa dos vetores for a indicada na Figura 4.19 automaticamente todos os produtos vetoriais terão a componente em z (DNMZ) positiva. Caso contrário deve-se inverter o sinal das componentes (DNMX, DNMY e DNMZ).

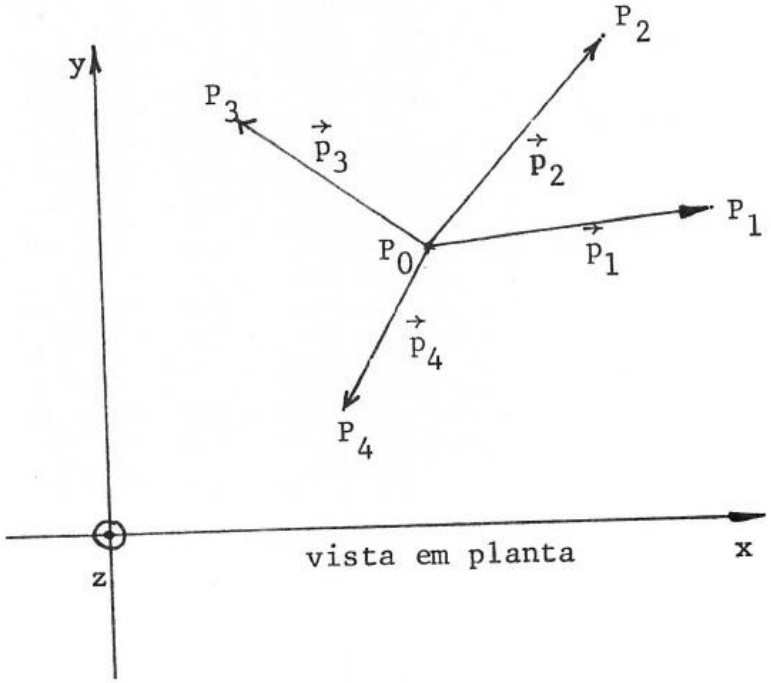


Figura 4.19 - Posição relativa de vetores que fornecem todos os produtos vetoriais positivos.

2) <u>Estimativa de</u> z_{xx}, z_{xy}, z_{yy}

O processo é análogo ao descrito no item 1, acima, havendo 3 enlaces que correspondem exatamente aos 3 enlaces anteriores:

DO 34 - DO 24 enlace mais externo

DO 33 - DO 23 enlace intermediário

DO 32 - DO 22 enlace interno.

A diferença reside em que o valor de z utilizado no cálculo das derivadas primeiras é substituído agora por z_x e por z_y. Computacionalmente as duas operações podem ser realizadas simultaneamente, poupando-se passos de programa.

As componentes em z (DNMZ e NMZ) são invariantes para o cálculo das derivadas já que só dependem dos valores de x e y, que não se modificam.

Quando se utiliza z_x em lugar de z obtém-se z_{xx} e z_{xy}. Quando se utiliza z_y no lugar de z obtém-se z_{yx} e z_{yy}. Toma-se a média de z_{xy} e z_{yx}.

O armazenamento é o seguinte:

z_{xx} = PD (JPDO-2) = NMXX/NMZ

z_{xy} = PD (JPDO-1) = (NMXY + NMYX)/(2*NMZ)

z_{yy} = PD (JPDO) = NMYY/NMZ.

Está terminada a sub-rotina.

4.3.7. SUB-ROTINA IDPTIP (pp. 15 a 19)

Esta sub-rotina realiza a interpolação ou extrapolação, isto é, determina o valor de z em um ponto, interno ou externo ao polígono convexo dos pontos.

Os parâmetros de entrada são:

* XD, YD, ZD = listas de inteiros de dimensão NDP contendo as coordenadas x, y, e z dos NDP vértices.

* NT = número de triângulos.

* IPT = lista de inteiros de dimensão 3*NT contendo a numeração dos vértices dos triângulos.

* NL = número de segmentos de borda.

* IPL = lista de inteiros de dimensão 3*NL contendo a numeração dos pontos extremos dos segmentos da fronteira e o número do respectivo triângulo.

* PDD = lista de dimensão 5*NDP contendo as derivadas parciais nos vértices.

* ITI = número do triângulo em que cai o ponto no qual a interpolação deve ser feita.

* XII, YII = coordenadas do ponto no qual a interpolação deve ser feita.

O parâmetro de saída é ZII, o valor interpolado de z.

Comentários que podem ser feitos seguindo a listagem:

*) <u>Processamento preliminar</u>

Decodifica o valor ITO de acordo com as regras assinaladas na sub-rotina IDLCTN, item 4.3.5, n. 7.

Se ITO for menor que NTL o ponto cai dentro da fronteira e o programa vai para o endereço 20 (item I desta descrição).

Se ITO for maior que NTL há duas possibilidades,ambas de

áreas externas. Se IL1 for igual a IL2 o ponto cai numa área retangular semi-infinita (passo II desta descrição). Se IL1 for diferente de IL2 o ponto cai numa área triangular semi-infinita e a descrição desse processo figura no item III desta descrição.

Estudemos cada um desses passos.

I. Cálculo de ZII por interpolação (área interna).

Inicialmente confere, pensando numa segunda passada, se os coeficientes do polinômio já foram calculados o que equivale a pensar que algum outro ponto (de uma malha, por exemplo) já caiu nesse triângulo.

a) carrega a sub-rotina com as coordenadas e com as derivadas parciais dos 3 vértices do triângulo em que o ponto cai.

b) determina os coeficientes das fórmulas de transformação de coordenadas do sistema x-y para o sistema u-v e vice-versa.

Através das fórmulas matemáticas pode-se deduzir que o sistema u-v é o seguinte:

– u coincide com o lado 1-2 do triângulo (nesse sentido)

– v coincide com o lado 1-3 do triângulo (nesse sentido)

– as coordenadas dos pontos são as seguintes:

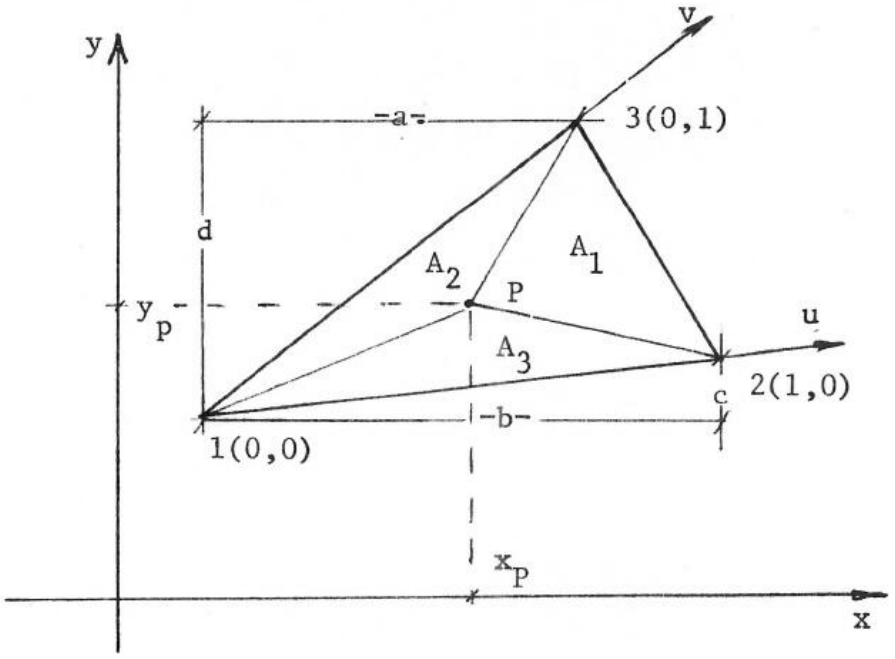
* ponto 1 (0,0) origem do sistema.

* ponto 2 (1,0) sobre o eixo u.

* ponto 3 (0,1) sobre o eixo v.

Trata-se portanto da conhecida transformação para o sistema de coordenadas baricêntricas (ou coordenadas de área) em que o triângulo é representado em sua forma paramétrica. A terceira coordenada pode ser omitida pois se sabe que a soma das três é igual à unidade. Para um maior desenvolvimento podem ser consultados, além dos livros de cálculo, os trabalhos de Bell (1969), Barnhill & Farin (1981) e Klucewicz (1978).

Trabalhar com coordenadas baricêntricas significa simplificar bastante os cálculos, principalmente os referentes à determinação dos coeficientes do polinômio interpolador. O primeiro passo do cálculo passa a ser a conversão das coordenadas do ponto, do sistema x-y para o u-v. O polinômio será calculado em u-v, sendo a sua equação um polinômio z(u,v) com as mesmas características de grau e potências que o polinômio



$u = A_2/A$

$v = A_3/A$

$w = A_1/A = 1-(u+v)$

$A = A_1+A_2+A_3$

Figura 4.20 - Mudança de coordenada do sistema x-y para o sistema u-v.

z(x,y); mudam os coeficientes.

Será necessário também calcular as derivadas parciais no sistema u-v e para isso aplica-se uma transformação linear, que se resolve sem maiores problemas.

Na Figura 4.20 apresenta-se uma representação dos dois sistemas de coordenadas.

c) converte as derivadas parciais do sistema x-y para o u-v.

Essa transformação é linear e sua dedução se faz com relativa facilidade, podendo-se consultar, por exemplo, Smirnov (1969). As fórmulas são as seguintes:

$$z_u = a \cdot z_x + c \cdot z_y$$

$$z_v = b \cdot z_x + d \cdot z_y$$

$$z_{uu} = a^2 \cdot z_{xx} + 2 \cdot a \cdot c \cdot z_{xy} + c^2 \cdot z_{yy}$$

$$z_{uv} = a \cdot b \cdot z_{xx} + (a \cdot d + b \cdot c) \cdot z_{xy} + c \cdot d \cdot z_{yy}$$

$$z_{vv} = b^2 \cdot z_{xx} + 2 \cdot b \cdot d \cdot z_{xy} + d^2 \cdot z_{yy}$$

As constantes a, b, c e d foram definidas previamente no programa e constam na Figura 4.20.

d) <u>Cálculo dos coeficientes do polinômio</u>.

Para acompanhar esse cálculo tenha-se em conta também que o polinômio z_{ii} assume uma forma análoga à do item 3.2, substituindo-se os coeficientes q por p e x,y por u,v.

O princípio geral é utilizar as 21 condições para determinar os 21 coeficientes do polinômio.

Para cada vértice temos 6 condições representadas pelos valores da função e das 5 derivadas. Substitui-se as coordenadas do ponto (u e v) no polinômio em z e iguala-se ao valor da função no ponto. Da mesma forma, realizam-se as 5 derivadas da função (x, y, xx, xy e yy) e igualam-se essas equações com o valor das respectivas derivadas no ponto.

Esse procedimento é repetido para todos os vértices e assim compreende-se a vantagem de utilizar um sistema que simplifique as coordenadas desses vértices para que assumam valores (0,0), (1,0) e (0,1): muitos termos desaparecem e as potências se simplificam já que a unidade elevada a qualquer expoente é sempre igual à unidade.

Os coeficientes estão listados com suficiente clareza nesse passo do programa e poupamos a sua repetição, bem como a sua dedução, que é um tanto maçante exigindo um grande cuidado na notação e a organização metódica dos cálculos. Facilita muito a utilização de variáveis auxiliares: H1, H2, H3, etc.

Uma descrição do cálculo dos coeficientes, passo a passo, encontra-se em Akima (1975) ou, de forma menos explícita, em Barnhil & Farin (1981).

e) <u>Converte as coordenadas x-y em u-v</u>.

Para isso vale-se dos coeficientes já calculados anteriormente: a, b, c e d e também de: ap = d/dlt; bp = -b/dlt; cp = = -c/dlt e dp = a/dlt onde dlt = a•d-b•c.

As fórmulas de conversão, facilmente dedutíveis, são:

$$u = ap\cdot(x_{ii}-x_0) + bp\cdot(y_{ii}-y_0)$$

$$v = cp\cdot(x_{ii}-x_0) + dp\cdot(y_{ii}-y_0)$$

f) Calcula o polinômio.

Através de fórmula engenhosa e compacta consegue evitar perda de precisão e ter que utilizar a potenciação, operação consumidora de tempo. Alguns manuais de FORTRAN oferecem maneiras semelhantes a essa para o cálculo de polinômios, mas só para funções univariadas (só x, ou só y).

A equação do polinômio está muito bem descrita na listagem do programa e por isso dispensamo-nos de copiá-la.

II Cálculo de ZII por extrapolação, em áreas retangulares semi-infinitas.

Inicialmente verifica se porventura já foram calculados, em uma passagem anterior, os coeficientes do polinômio extrapolador para essa área. Nesse caso passa direto aos itens e) e f): converte x_{ii} e y_{ii} para o sistema u-v e calcula o polinômio. Caso os coeficientes não estejam calculados o procedimento é:

a) Chama do arquivo os valores das coordenadas e das derivadas nos pontos extremos do segmento de fronteira em questão.

b) Determina os coeficientes de transformação do sistema x-y para o sistema u-v.

Os coeficientes são análogos aos determinados anterior-

mente. Neste caso as coisas se simplificam tendo em conta que o sistema u-v é ortogonal, com origem em 1, sendo o eixo u coincidente com o lado 1-2 e sendo orientado de 1 para 2 (Figura 4.21). Aplicada a transformação de escala o ponto 2 passa a ter coordenadas (0,1).

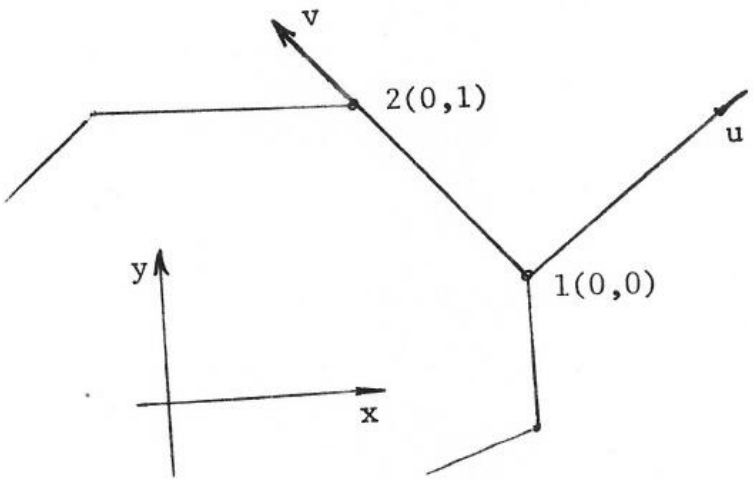


Figura 4.21 - Transformação de coordenadas, do sistema x-y para o sistema u-v.

c) Converte as derivadas parciais dos pontos extremos do segmento de borda, do sistema x-y para o sistema u-v.

As fórmulas, análogas às do triângulo, simplificam-se bastante.

d) <u>Calcula os coeficientes do polinômio</u>.

Para representar a superfície nesta região adota-se um polinômio bivariado do 5º grau na variável medida na direção da aresta (v) e do 2º grau na distância medida na perpendicular à aresta (u).

$$z = P_0 + u*(P_1 + u \cdot P_2)$$

$$P_0 = P_{00} + v*(P_{01}+v*(P_{02}+v*(P_{03}+v*(P_{04}+v*P_{05}))))$$

$$P_1 = P_{10} + v*(P_{11}+v*(P_{12}+v*P_{13}))$$

$$P_2 = P_{20} + v*(P_{21}+v*(P_{22}+v*P_{23}))$$

Ao todo são 14 coeficientes que são determinados através das equações fornecidas: pelo valor da função e das derivadas nos 2 pontos, e pelas equações de forma de variação imposta.

e) Converte as coordenadas do ponto, do sistema x-y para u-v. Sem maiores dificuldades.

f) Calcula o valor de z (polinômio). Também sem dificuldades, pela fórmula apresentada acima: d).

III <u>Cálculo de z_{ii} por extrapolação, para pontos que caem numa das zonas triangulares semi-infinitas</u>.

Verifica inicialmente se porventura os coeficientes correspondentes já não foram calculados.

a) Chama os valores das coordenadas e das derivadas no vértice da área triangular.

b) Calcula os coeficientes do polinômio.

Nessa região a função interpoladora é um polinômio bivariado em x e y, que conecta suavemente com os polinômios de cada uma das áreas retangulares vizinhas. Em termos de u, v é dado pela seguinte expressão:

$$z = P_0 + u*(P_1+u*P_{20})$$

$$P_0 = P_{00} + v*(P_{01}+v*P_{02})$$

$$P_1 = P_{10} + v*P_{11}$$

São 6 coeficientes a serem determinados por 6 condições: o valor da função e das cinco derivadas no ponto em questão.

c) Converte x_{ii} e y_{ii} para o sistema u-v.

Trata-se de uma simples translação de coordenadas.

d) Cálculo do polinômio.

Pela fórmula apresentada acima (item b).

4.4. CONCLUSÃO

Uma vez terminada a descrição do programa computacional podemos fazer uma breve avaliação do mesmo.

Trata-se de um programa muito bem elaborado, sendo esta a segunda versão de um algoritmo que já funcionava razoavelmente.

Serve-se de conceitos e fundamentos matemáticos densos para a adequada solução dos problemas, utilizando também engenhosos artifícios. Veja-se por exemplo as diversas aplicações do produto vetorial, que serviu para traçar a tangente a um polígono, para determinar o alinhamento de pontos, ordenar os vértices de um triângulo no sentido anti-horário, calcular derivadas parciais, dividir um plano em duas regiões, etc.Outro destaque pode ser a mudança de coordenadas para simplificar o cálculo dos 25 coeficientes do polinômio interpolador.

Preocupa-se por cobrir as situações limites e casos particulares que poderiam originar problemas de não funcionamen-

to do programa, por exemplo: regiões com menos de 4 pontos; todos os pontos alinhados; triângulos degenerados; etc.Em cada caso emite as correspondentes mensagens ao usuário.

A concepção como um todo parece razoável e coerente para obter bons resultados, e a prática o tem demonstrado. O ajuste de um polinômio do 5º grau, comprovadamente seguro, contínuo e suave, permite a criação de um modelo espacial que se ajusta bem às superfícies encontradas na prática.

A própria publicação como um algoritmo da ACM (Association Computing Machinery) e como artigo da divisão TOMS (Transations on Mathematical Software) avalizam de alguma forma o trabalho já que um corpo de consultores especializados revê os artigos, rejeita ou sugere melhorias. O de Akima foi revisto e corrigido. Os algoritmos são testados pelos usuários assinantes da revista que também fazem observações. Em um número posterior dessa publicação, Akima registra algumas modificações e agradece aos leitores que as enviaram.

Do ponto de vista computacional e de algoritmos utilizados, parece-nos muito bom. Condensa passos de programação, evita perda de precisão e procura o modo mais compacto e elegante de resolver os problemas. A própria disposição dos comandos, preocupação pelas margens, comentários, etc., indica um cuidado fora do comum.

- ■ -

III PARTE: CONTRIBUIÇÕES AO MÉTODO DE AKIMA E ESTUDOS ESPECIAIS

5.1. INTRODUÇÃO

Para a introdução de melhorias pareceu-nos necessário estudar a teoria matemático-geométrica do método, que dividimos em três áreas:

1) o estudo da triangulação incluindo também o critério de triangulação ótima;

2) o estudo da determinação e seleção de vizinhos de um vértice e o cálculo das derivadas e

3) o estudo da interpolação no triângulo.

Esses estudos permitiram visualizar contribuições em três pontos:

1. a substituição do critério do ângulo max-min pelo critério do círculo, que, sendo equivalente, é no entanto mais simples do ponto de vista geométrico. Computacionalmente economiza um pouco de memória e é mais rápido, de acordo com a implantação que fizemos através da função IDTROQ, substituindo a original IDXCHG;

2. o desenvolvimento detalhado da técnica de divisão e conquista para a diminuição do tempo de computação e

o fracionamento do problema em partes visando grandes conjuntos de pontos. Tendo em conta que os resultados obtidos com a rotina IDTANG original em microcomputadores foram satisfatórios (pode-se processar 400 pontos em cerca de 10 minutos) e que havia outras melhorias e complementos prioritários, não nos pareceu necessário desenvolver a parte computacional desse método. De qualquer forma, o estudo está suficientemente explícito para sua futura implantação.

3. com relação ao método de determinar vizinhos de um ponto, sugerimos um novo critério que nos pareceu mais adequado do ponto de vista físico-topológico: utilizar os vizinhos de Voronoi. Essa modificação propiciou uma melhor representação da superfície, como se comprovou em casos práticos. Implantamos uma nova rotina (IDVZVR) para determinar esses vizinhos, que se demonstrou sensivelmente mais rápida do que a sua correspondente (IDCLDP) no método original. Para o cálculo das derivadas foram suficientes algumas modificações na rotina existente (IDPDRV) para compatibilizá-la com um número de vizinhos variável e não fixo.

Outras contribuições ao método de Akima surgiram com os complementos e sua adaptação a microcomputadores.

5.2. ESTUDO ESPECIAL SOBRE A TRIANGULAÇÃO E OS CRITÉRIOS DE DEFINIÇÃO DA TRIANGULAÇÃO ÓTIMA

5.2.1. TRIANGULAÇÕES

No item 2.3.2 já tratamos da triangulação: em resumo, fornecidos alguns pontos distribuídos de maneira aleatória, trata-se de uni-los convenientemente para formar uma rede de triângulos. A malha triangular deve ter como vértices somente os pontos fornecidos, isto é, não deve haver cruzamentos de linhas que unem dois pontos, o que daria origem a um vértice fictício.

Não deve haver pontos isolados ou não totalmente conectados. Por exemplo, cada vértice de fronteira deve estar unido a pelo menos duas arestas e cada vértice interno a pelo menos três. Caso contrário seria necessário completar a triangulação.

Teoricamente, dado um conjunto de pontos, existe uma série de triangulações topologicamente distintas, que obedecem às propriedades acima descritas. Seu número fornecido pelo calculo combinatório pode ser calculado pela fórmula deduzida por Brown (1965):

$$D_{n,m} = \frac{2\cdot(2m+3)!(4n+2m+1)!}{(m+2)!m!n!(3n+2m+3)!} \qquad (1)$$

onde:

$D_{n,m}$ - número de triangulações distintas;

m - número de pontos na fronteira convexa menos 3;

n - número de pontos internos.

Alternativamente podemos utilizar a fórmula abaixo, que obedece a nomenclatura utilizada no programa de Akima:

$$D = \frac{2.(2*NL-3)!(4*NDP-2*NL-5)!}{(NL-1)!(NL-3)!(NDP-NL)!(3*NDP-NL-3)!} \qquad (2)$$

onde

D - número de triangulações distintas;

NDP - número total de pontos;

NL - número de pontos na fronteira.

A fórmula fornececida por Brown (1) tem sido repetida em alguns trabalhos, mas sem muita análise. Não se diz por exemplo que esse número se reduz consideravelmente de acordo com a disposição dos pontos. Assim, por exemplo, para um quadrilátero (4 pontos na fronteira) com um ponto interno, existem 5 configurações teóricas dadas pela fórmula, mas desde que a posição do ponto interno seja variável. Quando fixamos a posição desse ponto interno, só dispomos de 3 configurações distintas e, no caso particular de que o ponto interno esteja no cruzamento das diagonais, só existe uma possibilidade.

De qualquer forma, o número de triangulações possíveis continua sendo muito grande e é necessário fixar alguns critérios para obter uma configuração boa, e se possível ótima, para realizar com ela a interpolação de uma função.

A conformação dos triângulos devem ser "tão boa quanto possível". Essa expressão equivale a procurar que os triângulos sejam o mais equiláteros possíveis, evitando ângulos muito agudos e lados de tamanhos muito díspares num mesmo triângulo.

É necessário portanto definir um critério (e um algorit-

mo) para a união dos pontos a fim de conseguir triangulações aceitáveis e, se existir, a melhor delas.

A seguir fazemos a descrição de alguns algoritmos que contêm implícita ou explicitamente um critério de triangulação. Como se poderá ver, em geral, todos eles convergem para a triangulação de Delaunay.

5.2.2. O ALGORITMO DA IBM (1965):

(Contouring by triangulation - 1620 MP-21X)

Concebido para problemas que envolvem até 100 pontos distribuídos aleatoriamente, necessita como dados de entrada as coordenadas desses pontos.

São requeridos 3 passos para a realização da triangulação:

1. criar polígonos convexos concêntricos. Utilizando todos os pontos descobre-se a fronteira convexa e retiram-se esses pontos do conjunto. Com os remanescentes calcula-se a nova fronteira convexa, retirando também esses pontos. O processo prossegue até que não existam mais pontos internos;

2. quebram-se esses polígonos em triângulos unindo os vértices de um polígono ao seu interno mais próximo, tomando-se o cuidado de que os triângulos não se sobreponham;

3. testa-se a configuração desses triângulos para ver se é conveniente uma troca de triângulos na base de melhores ângulos.

A própria publicação da IBM desaconselha sua utilização, falando um uso em casos restritos. Isso também se deve à posterior recomendação de realizar uma interpolação linear em cada triângulo.

Como mostraremos mais adiante, esse algoritmo não garante que a triangulação seja boa pois não testa as arestas que pertencem aos sucessivos polígonos convexos-concêntricos para ver se uma troca de triângulos em torno delas não daria bons resultados.

5.2.3. O ALGORITMO DE DÜPPE E GOTTSCHALK (1970)

(critério da mínima soma das arestas)

"minimum spanning tree"

Calculam-se as distâncias entre cada ponto e todos os demais, destacando-se a aresta mínima. Eliminam-se a seguir todas as arestas que cruzam com esta.

Toma-se a seguir a aresta mais curta, dentre as que possuem um vértice em comum com a aresta mais curta. Estas duas arestas definem o primeiro triângulo.

O processo prossegue, sempre tomando as arestas de menor comprimento dentre as ainda não acrescentadas e eliminando-se as que cruzem triângulos já formados. Quando não restar mais nenhuma aresta o processo está terminado.

Pode-se demonstrar que esse algoritmo conduz à configuração que fornece a mínima soma das arestas, e este é um problema conhecido: o do caminho mínimo, árvore de mínimo cami-

nhamento, etc.

Pode-se demonstrar também, como veremos mais adiante,que esse critério não leva em conta os ângulos do triângulo, que podem ser muito agudos.

O mínimo comprimento global das arestas pode levar a triângulos que tenham lados díspares (um muito pequeno e outros médios) e portanto seria possível uma troca que produzisse melhores ângulos ao balancear o comprimento dos lados dentro do triângulo. São portanto preferíveis lados não-mínimos mas que, em cada triângulo, forneçam medidas mais proporcionais.

Por outro lado, computacionalmente este método não é dos mais eficazes já que dispende muito tempo de computação nos cálculos das distâncias e na verificação de cruzamentos de linhas.

5.2.4. O PRIMEIRO CRITÉRIO DE LAWSON (1972)

(o critério do ângulo max-min)

Já foi descrito anteriormente pois Akima o utiliza. Na teoria veja-se o item 3.1 e na prática veja-se a descrição da função IDXCHG que implementa o método, no item 4.3.1.

Pode-se resumir dizendo que "existindo um quadrilátero estritamente convexo, deve-se escolher dentre as duas partições de triângulo aquela que maximiza o mínimo ângulo interno dentre os 6 produzidos em cada configuração".

É um critério genérico, que não necessita começar com uma configuração particular nem exige o início do processo a-

través de uma aresta de fronteira. Na implementação de Akima detecta os possíveis casos de degeneração.

Quando se realizam todos os testes necessários, e Akima o faz, este critério conduz à triangulação de Delaunay, que é globalmente ótima, como explicaremos mais adiante.No entanto, nem Lawson nem Akima identificaram explicitamente a triangulação produzida com a de Delaunay.

5.2.5. O ALGORITMO E O CRITÉRIO DE CAVENDISH (1974)

(elementos finitos)

É um algoritmo utilizado na geração automática de uma rede plana de elementos triangulares para cálculos através do método de elementos finitos.

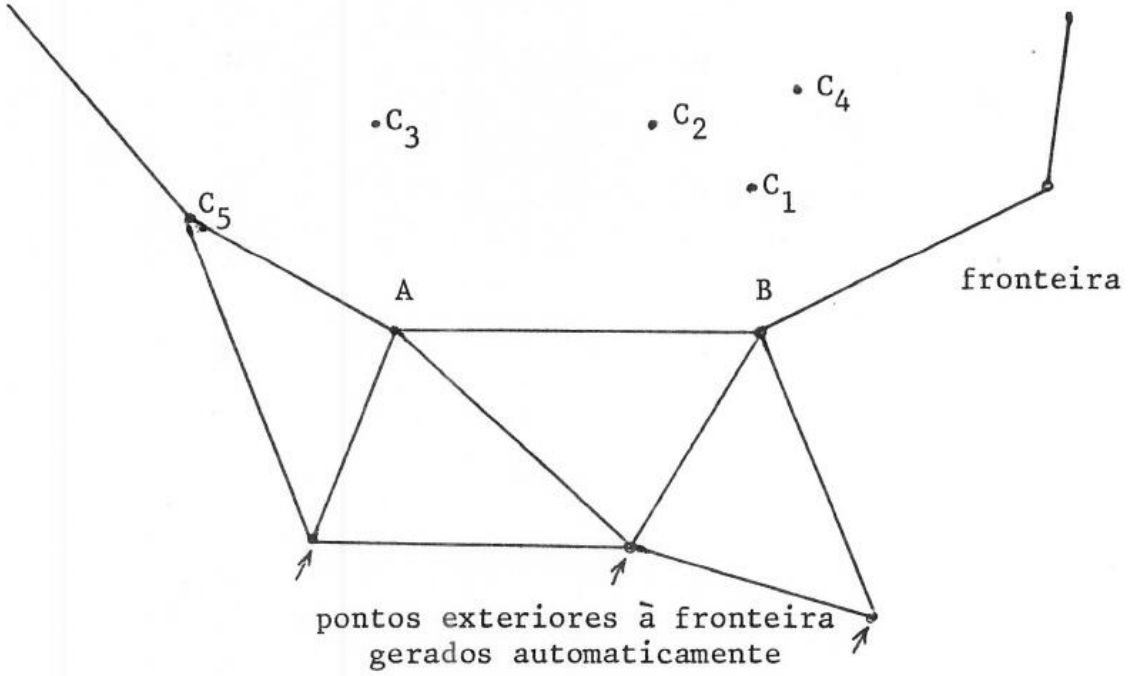


Figura 5.1 - Geração de malha triangular; início do processo: 5 vértices C_k candidatos à união para formar o triângulo ABC.

Trata-se de uma versão melhorada do algoritmo de Suhara e Fukuda (1972) e que trabalha da maneira descrita a seguir.

Deve ser fornecida uma lista de todos os pontos não gerados automaticamente pelo computador, destacando também os que pertencem à fronteira, cuidado esse que permitirá diferenciar os pontos fornecidos (internos) dos gerados (externos).

Conectam-se inicialmente dois nós A e B da fronteira. O objetivo é determinar um nó C, interno à fronteira (cai a esquerda do segmento orientado, conforme Figura 5.1) de maneira que o triângulo ABC seja, em certo sentido, ótimo.

Para isso selecionam-se 5 pontos C_k que possuem os mínimos comprimentos $d_k = AC + CB$; esses são os vértices candidatos à união.

Preliminarmente, supondo um passo genérico, realiza-se um teste para ver se o triângulo ABC_k $(k=1,\ldots,5)$ superpõe-se com algum triângulo ou borda de fronteira gerado anteriormente. Nesse caso elimina-se C_k.

O segundo teste verifica se ABC_k contém em seu interior algum outro vértice candidato, o que também leva à eliminação de C_k.

Para fazer a seleção dos restantes pontos viáveis leva-se em conta:

1. quão próximo de equilátero é o triângulo ABC_k;
2. quais são as possíveis conseqüências de usar AC_k e C_kB como bases para triângulos futuros.

Como se pode ver pela Figura 5.2, a escolha de C_2 torna ABC_2 próximo de equilátero mas BC_2C_1 está longe de sê-lo e por isso deve-se testar outra configuração, por exemplo ABC_1.

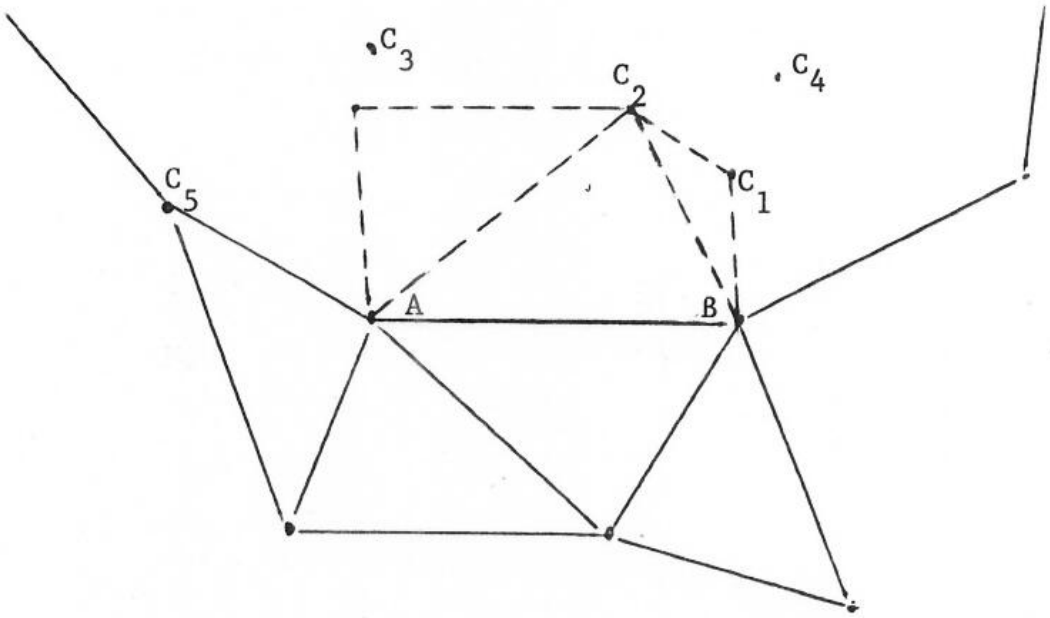


Figura 5.2.- Teste de possíveis configurações futuras que definirão o ponto a ser conectado em AB.

Resta operacionalizar o método através de um parâmetro p que seja, para um triângulo, a medida da sua "proximidade" do equilátero. Para isso γ_{ABC} pode ser definido como sendo a relação entre o maior lado e a altura correspondente de um triângulo; relação que para o equilátero vale $\sqrt{3}/2$. Por isso Cavendish utiliza como medida $p = |\gamma_{ABC} - \sqrt{3}/2|$.

Através da comparação de mínimos p que ocorreriam no próximo estágio consegue-se definir o vértice candidato que oferece melhores condições e deve ser o preferido para formar o triângulo. O processo continua até que não haja mais vértice a conectar.

Podemos fazer dois comentários. O primeiro refere-se às situações limites: pontos alinhados com AB ou os 5 pontos mais próximos pertencentes à fronteira. O programa não menciona como resolver esses casos.

Em segundo lugar, o critério proposto só garante a otimização num segundo nível, isto é, prevendo um passo futuro. Podem ocorrer, e de fato ocorrem, situações em que um ponto num terceiro estágio obriga a remodelação da configuração atual. Ou seja o método garante a otimização local mas não assegura a global.

5.2.6. O SEGUNDO CRITÉRIO DE LAWSON (1977)

(o critério do círculo)

Neste segundo trabalho Lawson propõe uma melhoria geomé-

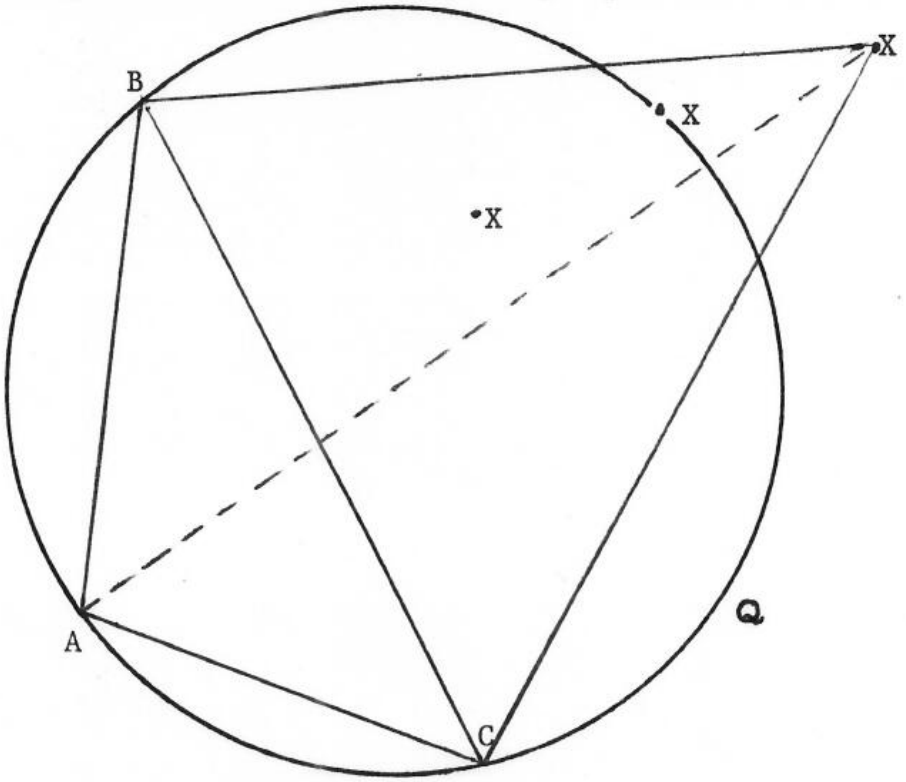


Figura 5.3 - Critério do círculo proposto por Lawson. Se X cair dentro trocam-se as diagonais; caso contrário, não.

trica para o seu trabalho inicial (1972), estabelecendo o critério do círculo, que pode ser formulado da seguinte maneira. Seja BC (Figura 5.3) uma aresta interna de uma triangulação e Q o quadrilátero formado pelos dois triângulos que compartilham a aresta BC. Considere-se agora o círculo que passa por três vértices de Q. Se o quarto vértice X estiver dentro do círculo deve-se trocar a diagonal BC por AX; caso contrário a configuração inicial deve permanecer.

Para mostrar que este novo critério é equivalente ao do ângulo max-min tomemos 4 pontos cocirculares, isto é, X caindo sobre o círculo de ABC (Figura 5.4).

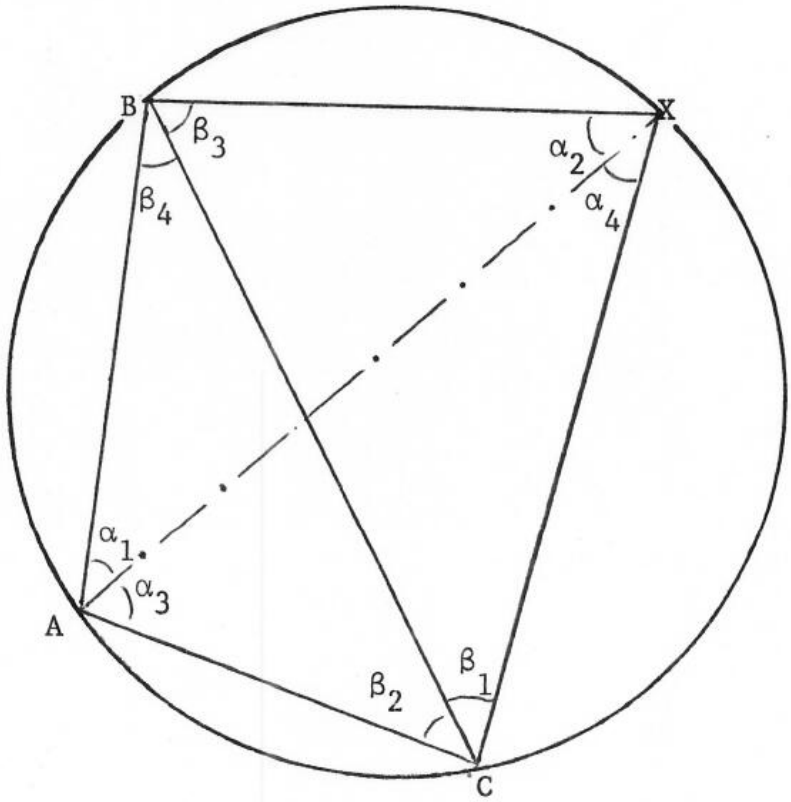


Figura 5.4 - Eqüivalência entre o critério max-min e o critério do círculo.

Pela geometria temos que:

$$\alpha_1 = \beta_1 \qquad \alpha_3 = \beta_3$$

$$\alpha_2 = \beta_2 \qquad \alpha_4 = \beta_4$$

Em resumo: $\alpha_i = \beta_i$,

pois cada par de ângulos α_i e β_i subentende o mesmo arco de circunferência.

Qualquer que seja a posição de X, sobre a circunferência, o mínimo ângulo α_i na configuração da diagonal AX conectada, será igual ao mínimo ângulo β_i (BC conectada). O critério do ângulo max-min permite escolher qualquer uma das diagonais, o mesmo acontecendo com o critério do círculo.

Quando movemos X para fora do círculo, a geometria indica que os ângulos α_i serão estritamente menores que os β_i correspondentes e portanto a configuração de máximo dos mínimos ângulos é a que fornece os ângulos β_i e portanto BC deve ser selecionada como diagonal.

Se movermos X para dentro do círculo, os ângulos α_i serão maiores que os β_i correspondentes e deve-se escolher a diagonal AX.

Portanto, os dois critérios são equivalentes, produzindo a mesma triangulação, que é localmente ótima.

Para obter uma triangulação globalmente ótima, nenhum ponto do conjunto deve cair dentro do círcuncírculo do triângulo em questão. Essa triangulação será única e pode-se demonstrar que coincide com a de Delaunay.

Este novo critério tem a vantagem de ser mais simples geometricamente e mais rápido do ponto de vista computacional já que determinar se um ponto cai dentro ou fora de um círculo é mais direto que calcular 8 ângulos e compará-lo para determi-

nar o mínimo.

Para melhorar o programa de Akima desenvolvemos a parte computacional referente a este novo critério para poder substituir o antigo. Houve uma economia de memória e de tempo, como se pode ver no item 5.4.

5.2.7. O CRITÉRIO DE NELSON (1978)

(elementos finitos)

É também uma modificação do algoritmo de Suhara-Fukuda, que tem em conta o algoritmo de Cavendish (5.2.5). Como este, parte de uma linha base AB e considera um dos semi-espaços definido por essa base que equivale, no de Cavendish, à região interna à fronteira convexa.

Considerando os pontos desse semi-espaço, define um cri-

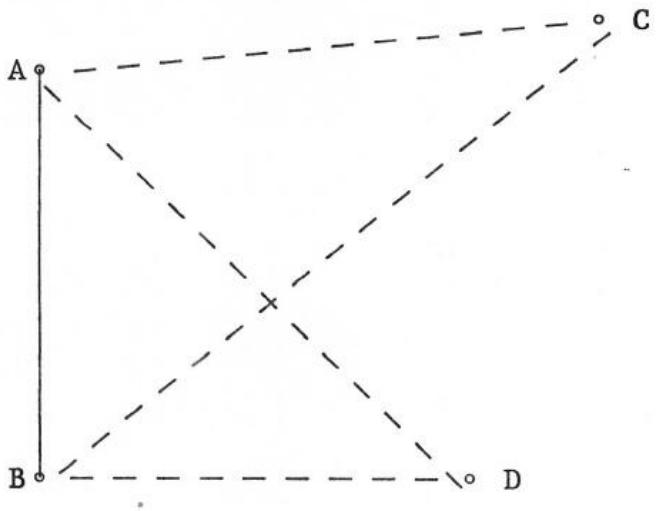


Figura 5.5 - Início da construção do melhor triângulo com base em AB.

tério para a formação dos triângulos que pode ser acompanhado

pela Figura 5.5 e que tem alguma semelhança com o critério do círculo.

Se AB é a base e D é mais aceitável que C, Nelson supõe que A é mais aceitável que C quando a base é BD. Se isso for verdade (pode-se demonstrar que realmente é, recorrendo à triangulação de Delaunay), evitaremos o teste de ver (na etapa seguinte) se o triângulo mais aceitável com base BD (está fora do círculo) cruza o elemento ABD. Isso poupa um tempo considerável nos testes de superposição que faz o algoritmo de Cavendish.

A seguir, determina o triângulo mais aceitável entre ABC e ABD: será aquele cuja circunferência circunscrevente possui o centro mais próximo da base AB (conforme Figura 5.6).

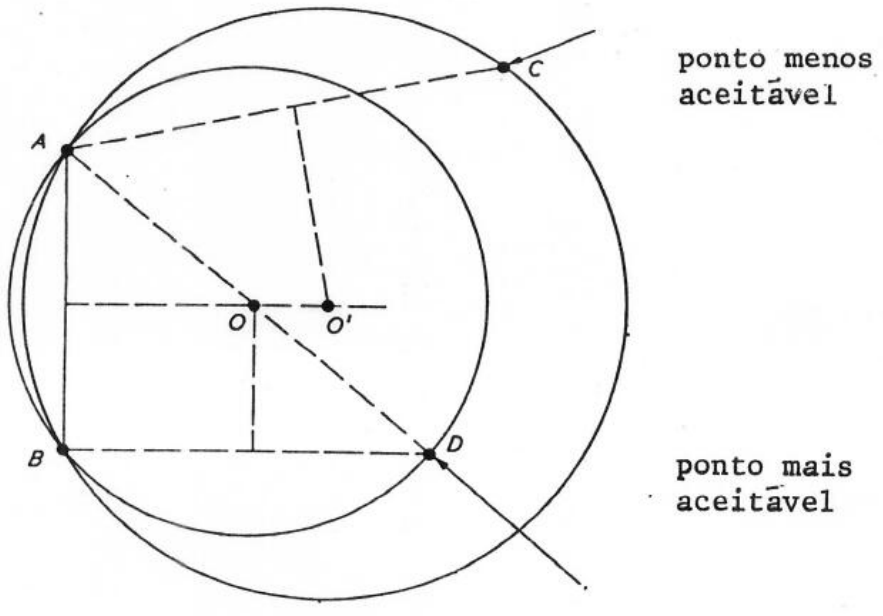


Figura 5.6 - Determinação do ponto mais aceitável (C ou D) para formar triângulo com AB.

Partindo de uma base de fronteira, escolhe o ponto mais aceitável dentre todos os que estão no interior do polígono, e forma-se o primeiro triângulo. Uma das duas novas arestas do

triângulo assim formado servirá como próxima base e a outra é armazenada. Sempre que a base em curso, numa etapa genérica, for um segmento de fronteira ou já tiver aparecido em dois triângulos existentes, busca-se uma outra base no armazém.

O algoritmo termina quando o armazém atinge o número de aresta previsto, que pode ser determinado previamente em função do número de vértices total e na fronteira, que devem ser fornecidos previamente.

Como avaliação podemos dizer que este processo produz uma triangulação localmente ótima e que, com pequenos cuidados adicionais poderia construir a triangulação de Delaunay e ser assim globalmente ótima. Nelson não mostra a preocupação por atingir um ótimo global mas uma das figuras constantes de seu trabalho fornece uma figura bastante razoável, bastando 2 ou 3 permutas para obter a triangulação de Delaunay.

Geometricamente é semelhante ao critério do círculo examinando no entanto somente um semi-plano e por isso pode não acontecer a triangulação ótima, em função da disposição dos pontos. Por outro lado, exige que o método se inicie na fronteira, o que pode ser uma limitação.

Necessita que a fronteira seja fornecida previamente;poderia ser implementado um método para isso já que existe muita literatura sobre a fronteira convexa.

Um problema não resolvido totalmente e que pode gerar muito dispêndio de tempo é o fato de ter que calcular para cada ponto o centro e o raio da circunferência gerados por ele e pelas bases que devem ser examinadas.

5.2.8. O CRITÉRIO DE SIBSON (1978)

Em um pequeno trabalho teórico, Sibson propôs um novo critério que demonstrou ser equivalente ao primeiro critério de Lawson (ângulo max-min) e "bem superior a este do ponto de vista computacional". Acontece que este novo critério é exatamente o critério do círculo, publicado um ano antes (1977) e que foi proposto pelo próprio Lawson.

O trabalho de Sibson foi enviado para publicação em março de 1977 e, depois das revisões correspondentes, foi publicado em 1978. Certamente não houve tempo hábil para tomar conhecimento da publicação do novo trabalho de Lawson.

Este novo critério (Sibson) foi implantado computacionalmente por Green e Sibson (1978) visando somente a triangulação, ficando as aplicações específicas por conta do usuário.

De acordo com informações desses autores, testado em diversos casos, o algoritmo apresentou resultados bastante satisfatórios.

5.2.9. O CRITÉRIO DE LEWIS E ROBISON (1978)

Esses autores fornecem um critério de triangulação idêntico à primeira versão do de Lawson, ainda que não o citam na bibliografia. São comparadas duas partições de triângulos, sendo escolhida aquela que produz o máximo dos ângulos mínimos. A única diferença é que, quando os ângulos mínimos são muito próximos prefere-se a partição ao longo da diagonal que apresenta o maior gradiente.

A diferença "muito pequena" entre os dois ângulos mínimos é arbitrada como sendo de 0,03 radianos.

Para este critério valem os mesmos comentários feitos para o de Lawson (1972). Com relação ao algoritmo como um todo, parece-nos que os mapas de curvas de nível apresentados deixam a desejar, principalmente por realizarem uma interpolação linear em cada triângulo.

5.2.10. CONCLUSÃO

Temos notícia de outros critérios, como os de Bykat (1976), Moscardini e outros (1981), Moscardini e outros (1983), mas os apontados bastam como indicativo de que os diversos métodos convergem para a produção da triangulação de Delaunay.

Dentre todos eles, o do círculo, na formulação de Lawson, foi o que nos pareceu mais satisfatório para ser implantado, por produzir com certeza e simplicidade essa triangulação.

5.3. A TRIANGULAÇÃO DE DELAUNAY

*Lee & Schachter (1980) apresentaram notável trabalho sobre o tema, que utilizamos com proveito para traçar as linhas básicas deste item.

5.3.1. DEFINIÇÕES

a) Polígonos de Voronoi

Dado um conjunto $V = \{v_1, v_2, \ldots, v_n\}$, $n \geq 3$, de vértices no

plano, chamemos de d_{ij} a distância entre v_i e v_j. A região

$$v(i) = \{x | d_{xi} \leq d_{xj}, j=1,\ldots,n\},$$

que é o lugar geométrico dos pontos (x) do plano mais próximos de v_i do que de qualquer outro vértice, é chamada de polígono de Voronoi (ou Dirichlet, Wigner-Seithz, Thiessen, etc.).

Pode ser pensado como um conjunto de cristais (ou células) em um processo de crescimento a partir do núcleo.

Suponhamos que cada vértice v seja um núcleo e que os cristais se expandem simultaneamente com velocidade uniforme até que encontrem outra frente de expansão, quando então se cria um ponto de contacto e também uma borda de fronteira. Nesse processo o plano ficará dividido em polígonos convexos, cada um em torno de um núcleo. Nas bordas da região os polígonos são abertos (ou semi-infinitos). Fica assim definida a partição (ou polígonos) de Voronoi.

Examinando com mais detalhe o processo, vemos que o ponto de contacto de duas células cristalinas ocorre inicialmente no ponto médio da aresta que une os dois núcleos e que continua a acontecer nos pontos eqüidistantes dos núcleos, ou seja, na mediatriz desse segmento que é chamada uma aresta de Voronoi. Tem a propriedade de ser eqüidistante dos núcleos e formar a fronteira entre as células. Essa aresta comum continua a expandir-se até que encontre uma outra aresta representando a expansão a partir de outros núcleos. O ponto de encontro é denominado ponto de Voronoi. Tem a propriedade de ser eqüidistante dos núcleos das células que concorrem nesse pon-

to, sendo portanto o circuncentro de um triângulo constituído pelos três núcleos (é o encontro de três alturas).

As células que possuem uma aresta em comum são chamadas vizinhas de Voronoi. Os núcleos dessas células são denominados vizinhos de Voronoi.

b) Triangulação de Delaunay

Unindo-se o centro de cada célula com o centro de todas as células vizinhas obtém-se a triangulação de Delaunay para os vértices v.

Uma partição é dual da outra, isto é, estão associadas segundo uma regra fixa de transformação, podendo-se obter uma a partir da outra.

Cada ponto de Voronoi está associado a um triângulo de Delaunay, sendo o seu circuncentro. Cada aresta de Voronoi está associada a uma aresta de Delaunay, sendo a sua mediatriz.

5.3.2. EXEMPLO GRÁFICO

Para vizualizar a partição Voronoi-Delaunay apresentamos a Figura 5.7, que construímos a partir do exemplo fornecido por Akima: Tabela 3.1 e Figuras 3.2 e 3.3.

Os trinta pontos desenhados nessa Figura são os assinalados com asterisco na Tabela 3.1. De acordo com o critério de Lawson e o procedimento de Akima construímos graficamente a triangulação, que está indicada com linhas cheias na figura. A poligonação de Voronoi, construída posteriormente está in-

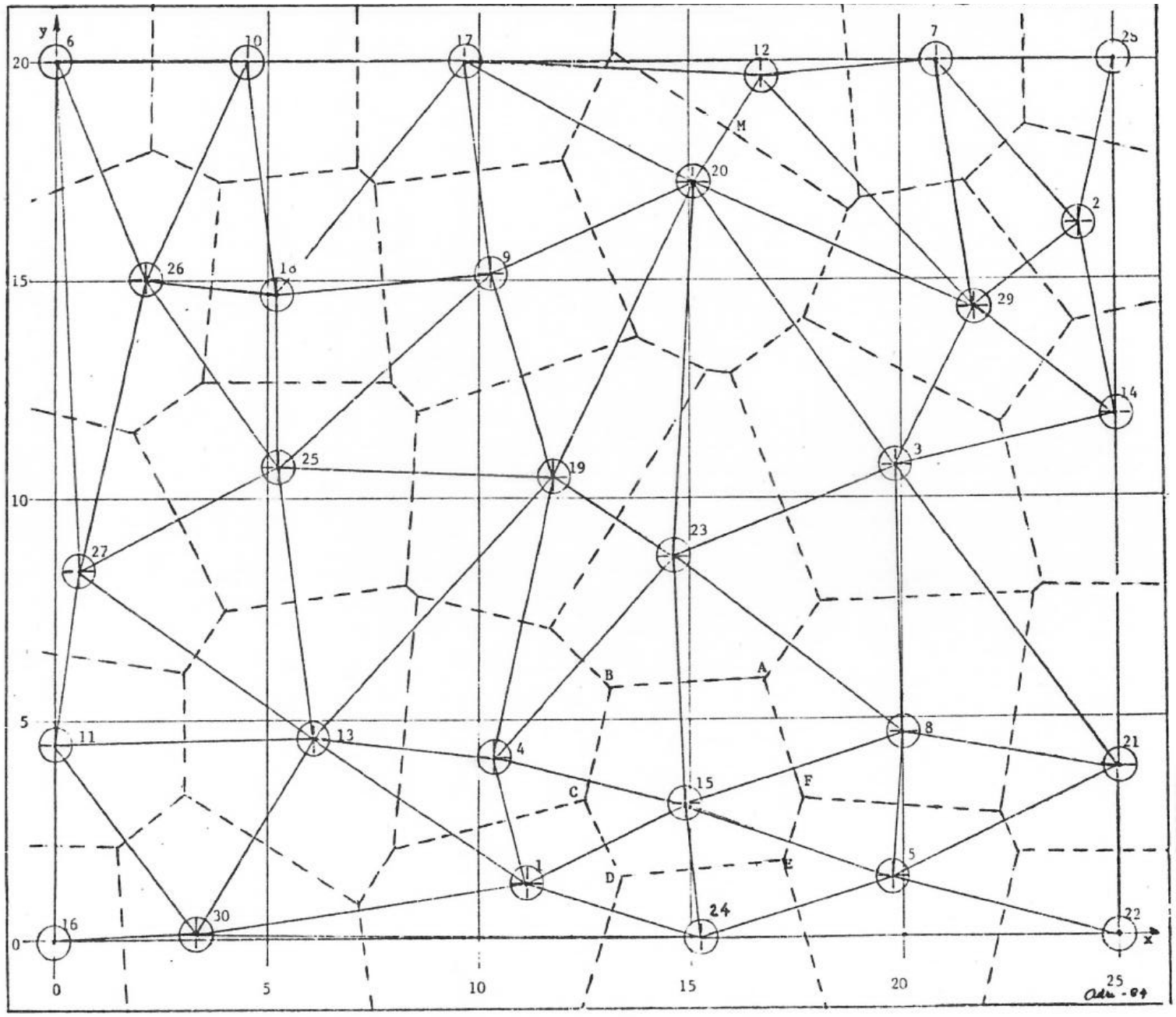


Figura 5.7 - Poligonação de Voronoi (linhas tracejadas) e Triangulação de Delaunay (linhas cheias).

dicada com linhas tracejadas.

Concentremos a atenção no ponto 15 para indicar a construção do polígono de Voronoi e a sua relação com a triangulação de Delaunay. Esse ponto é considerado o núcleo da célula e seus vizinhos imediatos são: 5, 8, 23, 4, 1 e 24. Traçando a mediatriz de 15-8 obtemos a linha AF; a mediatriz de 15-23 produz BA, e assim por diante.

O encontro dessas mediatrizes se dá em A, B, C, D e F que delimitam o polígono de Voronoi, que é a região do plano mais próxima de 15 do que de qualquer outro vértice.

As arestas AF, BA, CB, etc. são as arestas de Voronoi,perpendiculares às arestas de Delaunay: 15-8, 15-23, 15-4, etc. As células que contém 8, 23, 4, etc. são chamadas vizinhas de Voronoi, com relação à que contém o ponto 15.

Podemos notar que cada ponto de Voronoi (A, B, C, D,etc.) corresponde ao circuncentro de um triângulo de Delaunay. Por exemplo, A é o circuncentro do triângulo 15, 8, 23, bastando ver que as três alturas desse triângulo (mediatrizes dos lados) encontram-se precisamente em A.

Vale a pena a consideração de um caso particular, o daquele em que 4 (ou mais) vértices de Delaunay caem sobre uma mesma circunferência. Normalmente cada ponto de Voronoi corresponde ao encontro de 3 mediatrizes e portanto à junção de três células de Voronoi. No caso particular em consideração,o ponto de Voronoi corresponderia ao encontro de 4 (ou mais) células. Na Figura 5.7 isso quase ocorre na junção de 8, 21, 14 e 3 pois esses pontos estão quase sobre uma mesma circunferên-

cia.

5.3.3. PROPRIEDADES

LEMA 1. Dado um conjunto de NDP pontos (vértices), a triangulação de Delaunay contém um número de triângulos e de arestas que pode ser determinado em função de NDP e do número de arestas NL na fronteira convexa. Dessa forma temos:

a) Número de triângulos (NT)

$$NT = 2*(NDP-1) - NL$$

b) Número de arestas (NA)

$$NA = 3*(NDP-1) - NL$$

Isto pode ser demonstrado por indução, conforme por exemplo, Lee (1976). Essa quantificação, aparentemente sem maior transcendência é muito útil na determinação de posições de memória a serem reservadas numa determinada sub-rotina.

Como exemplo, sabemos que Akima reserva para a lista IPT de vértices de triângulos, 6*NDP – 15 posições de memória. Esse número pode ser obtido pensando que para cada triângulo devemos reservar 3 posições de memória. Então $IPT_{max} = 3*NT_{max}$. Da fórmula indicada em a) obtemos:

$$NT_{max} = 2*(NDP-1) - NL_{min} .$$

Mas $NL_{min} = 3$ pois a fronteira mínima seria um triângulo. Então:

$$NT_{max} = 2*NDP - 2 - 3 = 2*NDP - 5$$

e portanto $IPT_{max} = 6*NDP - 15$.

Para um estudo das regras que governam o número de elementos numa triangulação podem ser úteis: Führing (1975) e Ewing e outros (1970).

LEMA 2. Dado um conjunto de vértices V, uma aresta (v_i, v_j) é uma aresta da triangulação de Delaunay se e somente se existe um ponto x tal que o círculo com centro em x e passando por v_i e v_j não contém em seu interior nenhum outro ponto de V.

*COROLÁRIO: todas as arestas da fronteira convexa são arestas de Delaunay. Basta imaginar o centro x no infinito, do lado externo do polígono convexo; $v_i v_j$ fica sendo um trecho de uma circunferência que não contém em seu interior nenhum ponto de V .

Esta propriedade tem inspirado diversos métodos de construção da triangulação de Delaunay partindo precisamente da fronteira convexa que pertence toda ela à triangulação. Os algoritmos para a construção da fronteira convexa são abundantes, existindo programas computacionais à disposição. Nos últimos anos houve como que uma "corrida intelectual" em torno desse problema para encontrar a "solução ótima".

LEMA 3. Dado um conjunto de vértices $V \equiv \{v_1, v_2, \ldots, v_n\}$ o triângulo $v_i\ v_j\ v_k$ é um triângulo da triangulação de Delaunay se e somente se seu circuncírculo não contém nenhum outro ponto de V.

A demonstração dos lemas 2 e 3 pode ser encontrada em Lee

(1978) ou em Shamos e Hoey (1975). O Lema 3 corresponde ao critério do círculo, fornecido por Lawson (1977), já visto anteriormente no item 5.2.6.

O critério do círculo deriva portanto das propriedades naturais da triangulação de Delaunay e é apto para construi-la. Permite desenhá-la de maneira segura e é um instrumento para julgar os demais critérios. Por exemplo, Shamos e Hoey (1975) reinvindicam que a triangulação de Delaunay coincide com a de mínimo comprimento das arestas e, nesse caso poderia ser construída com um critério semelhante ao de Düppe-Gottschalk (1970). Lawson (1972) e Lloyd (1977) provam com um contra-exemplo, que isso não é verdade (conforme Figura 5.8).

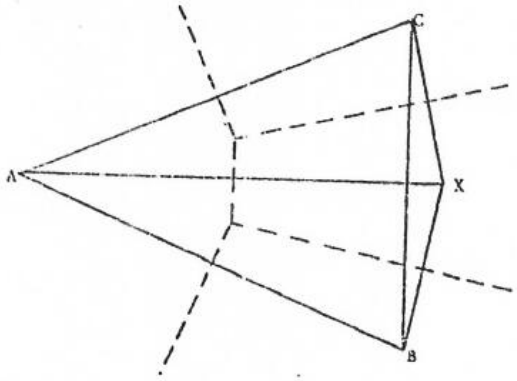


Figura 5.8 - Contra-exemplo para a afirmação de Shamos e Hoey: o mosaico de Voronoi indica o uso da diagonal mais comprida (AX). A outra partição produz ângulos muito agudos no triângulo CBX.

5.3.4. OTIMIZAÇÃO LOCAL E OTIMIZAÇÃO GLOBAL

Convém reparar que o critério de Lawson (1977) é de otimização local: "Dado um quadrilátero Q e uma diagonal d, tomemos o circuncírculo de um dos triângulos. Se o 4º ponto cair dentro devemos trocar a diagonal d pela outra; caso contrário mantemos d".

Ou seja, esse critério verifica a configuração de 4 pontos e nada nos diz sobre possíveis outros vértices que estejam nas proximidades. Para que os triângulos sejam realmente de Delaunay e portanto globalmente ótimos, é preciso que nenhum outro vértice caia dentro do circuncírculo considerado.

O fato de uma diagonal ter dado melhor resultado do que uma outra não significa necessariamente que ela seja uma aresta de Delaunay; pode ser que um outro teste venha a indicar sua troca por uma terceira.

Uma triangulação só será globalmente ótima quando não houver nenhuma troca possível de diagonais que melhore a configuração. Ou seja todas as arestas devem ter sido testadas, inclusive as novas arestas resultantes de cada troca feita. Esse procedimento é seguido pelo método de Akima.

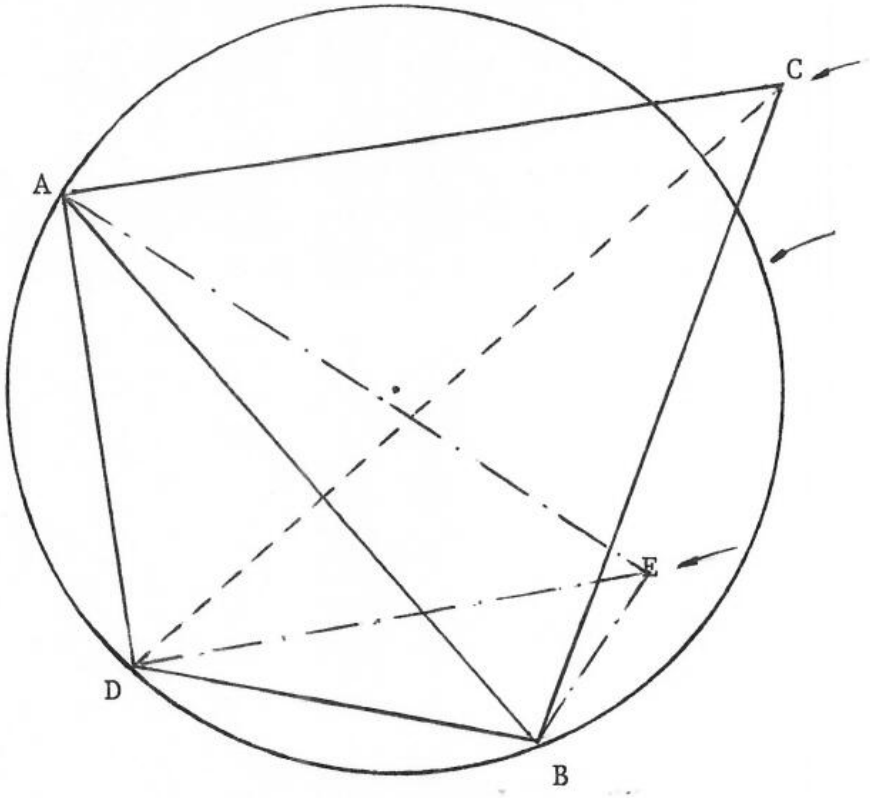


Figura 5.9 - Otimização local: AB é preferível a DC.
Otimização global: DE é preferível a AB.
O circuncírculo por ADB permite a verificação global.

Um exemplo ajuda a ilustrar esse fato. Dados 5 pontos, iniciemos o teste com o quadrilátero ABCD e então a diagonal AB será preferida a CD pelo critério de Lawson. Nem por isso AB seria definitiva já que quando testada contra DE, seria substituída por esta (conforme Figura 5.9).

5.3.5. BREVE HISTÓRIA E APLICAÇÕES DA TRIANGULAÇÃO DE DELAUNAY E DO MOSAICO DE VORONOI

A primeira referência ao tema deve-se a Pedro-Gustavo Lejeune Dirichlet (1805-Düren, 1859-Gotingen), matemático alemão, de ascendência francesa, catedrático de matemática da Universidade de Berlim (capital do então reino da Prússia) e sucessor de Gauss na cátedra da Universidade de Gotingen. Membro da Academia de Ciências de Berlim e associado estrangeiro da de Paris. Seu artigo foi publicado em 1850, pelo "Journal für die reine und angewandte Mathematik", 40, pp. 209-227.

A partição do plano recebe algumas vezes o nome de textura ou partição de Dirichlet. No entanto, associa-se normalmente o nome dessa partição a Voronoi, cujo trabalho foi publicado bem mais tarde (1908).

Georges Voronoi, matemático polonês, em contribuição para o mesmo "Journal für die reine und angewandte Mathematik" apresentou seu trabalho sobre a determinação de regiões paraleloédricas mais próximas de determinados pontos. Trata o problema de uma forma geral, para n dimensões. O caso mais simples (n=1) seria o de pontos sobre uma reta, associando dois segmentos, à direita e à esquerda, a cada ponto. O tamanho des-

se segmento é determinado pelo ponto médio com relação a seus vizinhos sendo que os pontos extremos estão associados também a uma semi-reta cada um. Para n = 2 teríamos a cobertura do plano, que é o nosso caso. Para n = 3 teríamos poliédros espaciais, e assim por diante.

B. Delaunay, matemático francês, descobriu a triangulação que leva o seu nome e que corresponde exatamente à união dos núcleos vizinhos de Voronoi. Publicou seu trabalho em 1934, no Boletim da Academia de Ciências da URSS.

Esses foram, ao que tudo indica, os primeiros três a tratar do tema. A partir daí multiplicam-se os autores que tratam do assunto.

A figura geométrica formada pelas células de Voronoi recebe diversos nomes: mosaico, polígono, textura, partição, disposição, diagrama, divisão, etc.

Da mesma forma, diversos autores emprestam seu nome à partição do plano em células de Voronoi. As diversas denominações devem-se aos pioneiros que, em cada área, propuseram aplicações concretas dessa partição.

Os geográfos falam em polígono de Thiessen (1911), que utilizou a partição para o cálculo de precipitações atmosféricas em grandes áreas. Rhynsburger (1973), também dessa área, descreve diversas aplicações. Seu método foi implantado computacionalmente: Tobler (1970).

Entre os que trabalham com a cristalografia, a partição foi utilizada como modelo para o crescimento de cristais, as-

sociando-se ao nome de Meijering. Gilbert (1962) é uma boa referência para as aplicações nesse campo.

Na área de estatística aplicada encontramos o seu uso no processo bidimensional de Poisson (distribuição aleatória de pontos no plano). Miles (1970) descreve diversas propriedades estatísticas da triangulação de Delaunay para esse processo de Poisson. Provavelmente a de Delaunay é a única triangulação não regular, tratável matematicamente.

Na área de bioestatística e ecologia das plantas, encontramos descrições de aplicações em Matern (1960),Switzer (1967), Mead (1971), Pielou (1977), e outros.

Na construção de modelos probabilísticos para fenômenos espaciais podem ser citados: Besag (1974) e Ripley (1977). Na astronomia, Kiang (1966); no urbanismo, Bogue (1949), e assim por diante.

As aplicações nas áreas de análise numérica e ciência da computação multiplicam-se a cada dia.

Como trabalho teórico de fundamentação vale a pena citar o de Rogers (1964), em que se desenvolve a teoria geral dos poliedros de Voronoi e da subdivisão de Delaunay.

Implementações de programas para n dimensões são descritas por exemplo em Bowyer (1981) e Watson (1981). Uma generalização do método de Akima, para a interpolação tridimensional pode ser encontrada em Navlakha (1984).

Na área de representação de terrenos e funções de duas variáveis, a triangulação de Delaunay é sem dúvida uma exce-

lente escolha pois permite uma boa distribuição dos pontos sobre o plano e evita triângulos com ângulos muito pequenos.Por isso tem sido recomendada por diversos autores: McLain (1976), Powell & Sabin (1977), Lawson (1977), Sibson (1978), Pettinati (1983).

5.3.6. ALGUNS ALGORITMOS PARA A CONSTRUÇÃO DA TRIANGULAÇÃO DE DELAUNAY

Dentre os diversos algoritmos que utilizam uma rede triangular de pontos, descrevemos brevemente a seguir alguns que utilizam o critério do círculo (ou equivalente) para produzir a triangulação de Delaunay.

a) Método de Akima (1978)

Utiliza o primeiro critério de Lawson (1972); descobre a aresta mais curta e seu ponto médio e vai acrescentando pontos por ordem crescente de distância.

Cada ponto acrescentado é ligado aos vértices visíveis, realizando-se a troca de triângulos quando necessário.

De acordo com informações do autor e testes que realizamos, o algoritmo roda em tempo inferior a $O(n^2)$.

b) Método de Green e Sibson (1978)

Apresenta a descrição geral de um algoritmo recursivo para a construção do polígono de Voronoi. A triangulação de Delaunay é obtida a partir daí.

Foi pensado para trabalhar com muitos pontos, tendo sido feitos diversos testes, incluindo problemas de até 12.000 pontos. Possui um programa de 700 linhas (contra 370 de Akima).

Dá bastante relevo à determinação dos pontos vizinhos pelo fato de consumir muito tempo. Os autores estimam um tempo $O(n^{3/2})$, que poderia ser reduzido, utilizando mais memória, para um tempo $O(n \log n)$.

No final apresenta alguns valores para o tempo gasto, em função também da memória utilizada e que podem ser resumidos no Quadro 5.1, abaixo.

Quadro 5.1 - Tempo de computação gasto pelo CDC-7600, em função do número de pontos e da memória máxima utilizada

Número de pontos	Memória máxima	Tempo (segundos)
1.000	10.000	0,8
9.000	90.000	9,7
10.000	90.000	11,6

c) Método de Lewis e Robinson (1977)

Utiliza o primeiro critério de Lawson (1972) sem referência explícita e introduzindo a modificação de escolher a diagonal de maior gradiente quando a diferença dos ângulos mínimos for muito pequena (menor que 0,03 radianos).

Realiza uma complexa partição dos pontos em subconjuntos menores; resolve estes e depois efetua a união: é a técnica da divisão e conquista.

Cada região pequena é triangularizada sem a preocupação

de otimização. Depois de haver triangularizado todos os pontos, isto é, feita a união das pequenas regiões, inicia o processo de troca de diagonais em um processo iterativo. As diagonais originadas em cada permuta são armazenadas para reexame na próxima iteração. Quando não existirem mais arestas por examinar (não existe troca que melhore a configuração) o processo está terminado.

Foi elaborado pensando principalmente em duas aplicações: elementos finitos e curvas de nível do terreno.

Em cada triângulo realiza uma interpolação linear, o que dá às curvas um aspecto um tanto pontiagudo, de difícil aceitação para os usuários e pouco compatível com os casos encontrados na realidade.

Os autores reivindicam, sem prova, que seu programa roda num tempo que é $O(n \log n)$. No entanto, Lee e Schachter (1980) demonstraram que na realidade é $O(n^2)$.

d) Método de Pettinati (1983)

Trata-se de autor brasileiro, mestre pela Escola Politécnica da Universidade de São Paulo (fevereiro de 1984),que desenvolveu seu algoritmo nos últimos anos, junto à empresa ITAUTEC. Representa um programa comercial com relativo sucesso, sendo o único desenvolvido em nosso meio(1983).

Utiliza o critério do círculo para obter a triangulação de Delaunay, e inicia o processo através das bordas.

O critério do círculo é testado, de forma racional, con-

tra todos os pontos do plano para garantir a otimização global. Para a racionalização utiliza-se de uma poderosa estrutura de dados para determinar a vizinhança de um triângulo; agrupa os pontos em árvore binária.

Preocupa-se também pelo aspecto estético, realizando a interpolação com o polinômio de Akima: mesma equação e mesmas condições impostas. Introduz algumas simplificação no cálculo das derivadas.

Como o método de Akima, este também converte a malha triangular em uma retangular para a posterior saída gráfica.

De acordo com as informações e testes do autor, o método é O(n log n), para um número grande de pontos. Para poucos pontos o comportamento é diferente e talvez isso se deva a que a estrutura de dados em árvore binária deve consumir um tempo comparativamente grande para problemas de pequena e média dimensão.

5.4. IMPLANTAÇÃO DO CRITÉRIO DO CÍRCULO.

COMPARAÇÕES

(Função IDTROQ)

5.4.1. ESQUEMA

De acordo com a formulação teórica (item 5.2.6), desenvolvemos o programa computacional correspondente ao critério do círculo e o seu esquema é o da Figura 5.10.

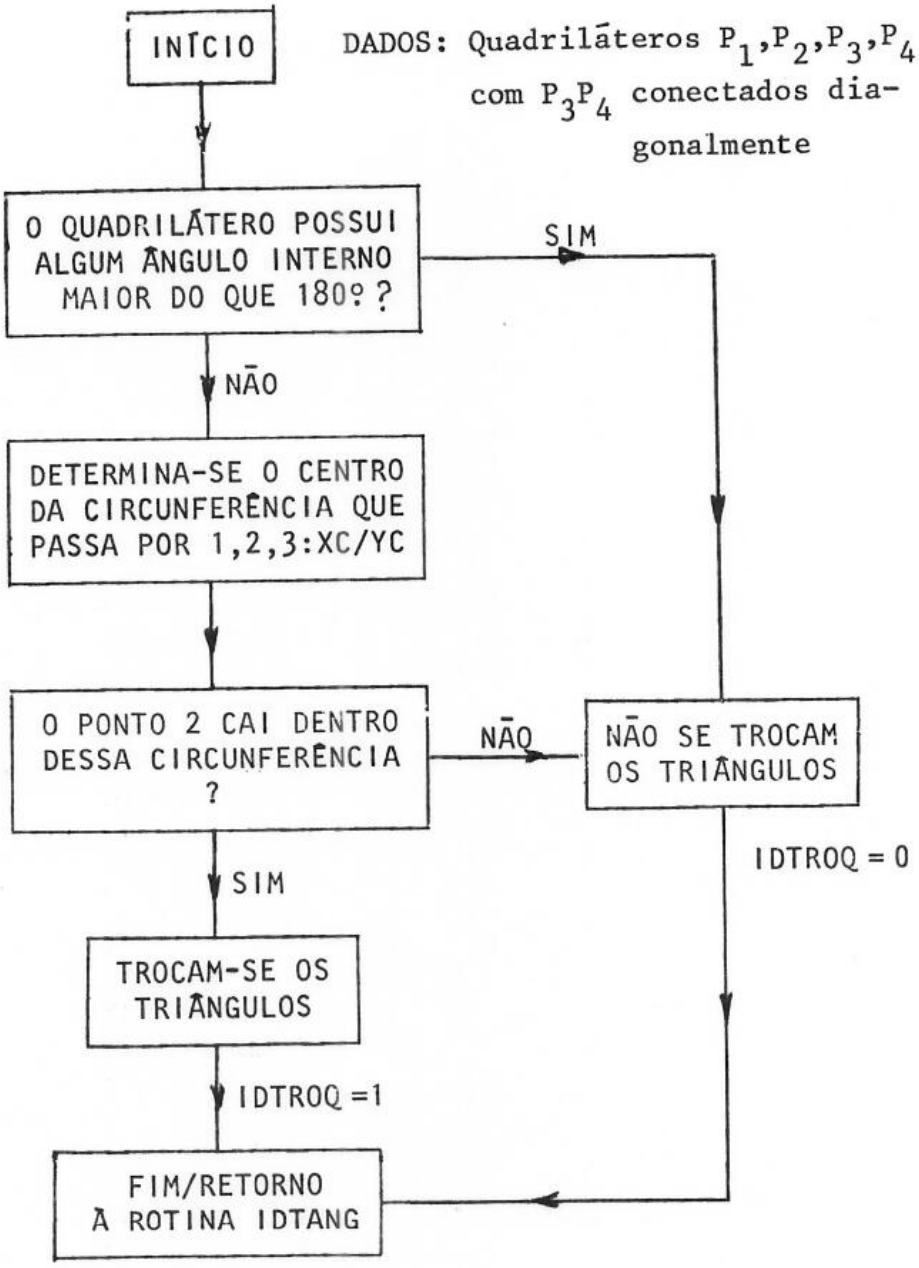


Figura 5.10 - Esquema da função IDTROQ

5.4.2. COMPARAÇÃO DE OPERAÇÕES E MEMÓRIA

Podemos resumir a comparação entre as funções IDXCHG (original) é IDTROQ nos Quadros 5.2 e 5.3, obtidos a partir das listagens do computador (.PRN).

Quadro 5.2 - Comparação de operações

	Função	
	IDXCHG	IDTROQ
Tipo de operação matemática	Número de vezes que é chamada	
+,-	38	29
*,/	21	20
**	12	-
IF	2	3
GO TO	1	2
AMAX 1	4	-
AMIN 1	2	-
EQUIVALENCE	6	-

Quadro 5.3 - Comparação de memória

Memória (bytes)	IDXCHG	IDTROQ
Programa	1.130	834
Dados	188	132
Total	1.318	966
Diferença		+ 352

Há portanto uma economia de memória de 352 bytes, que possibilitaria expandir um pouco mais a capacidade de pontos.

5.4.3. COMPARAÇÃO DE TEMPOS

Como IDXCHG e IDTROQ são chamadas pela rotina IDTANG, para comparar o tempo das duas utilizamos o seguinte procedimento.

Tomamos inicialmente um exemplo de 30 pontos e determi-

namos que chama 97 vezes a função de troca de triângulos (IDXCHG/IDTROQ), e medimos os respectivos tempos. A seguir, no mesmo exemplo, introduzimos os comandos necessários para repetir 100 vezes a função (IDXCHG/IDTROQ) cada vez que ela é chamada e todos os demais elementos permaneceram inalterados. Medimos então os novos tempos. A diferença dos tempos é devida unicamente à diferença de chamadas da função de troca de triângulos.

Com isso pudemos montar o quadro comparativo 5.4.

Quadro 5.4 - Comparação de tempos. Os tempos anotados representam a média de 6 leituras

Equipamento Polymax 201 DP	TRIANG		Número de chamadas da função de troca
	C/ IDXCHG	C/ IDTROQ	
Com repetição	52s	37s	97 × 100 = 9.700
Situação inicial	13s	11s	97
Diferença	39s	26s	9.603

Desse quadro pode-se concluir que a nova rotina é 30% mais rápida que a versão original ou que seu tempo é 0,67 do tempo de IDXCHG. O tempo de demora pode ser calculado da seguinte forma:

$$t_{IDXCHG} = \frac{52-13}{9700-97} = 4{,}1 \cdot 10^{-3} s$$

$$t_{IDTROQ} = \frac{37-11}{9700-97} = 2{,}7 \cdot 10^{-3} s.$$

No entanto, a economia de tempo, no cômputo geral vai depender do número de vezes que é chamada a função de troca de

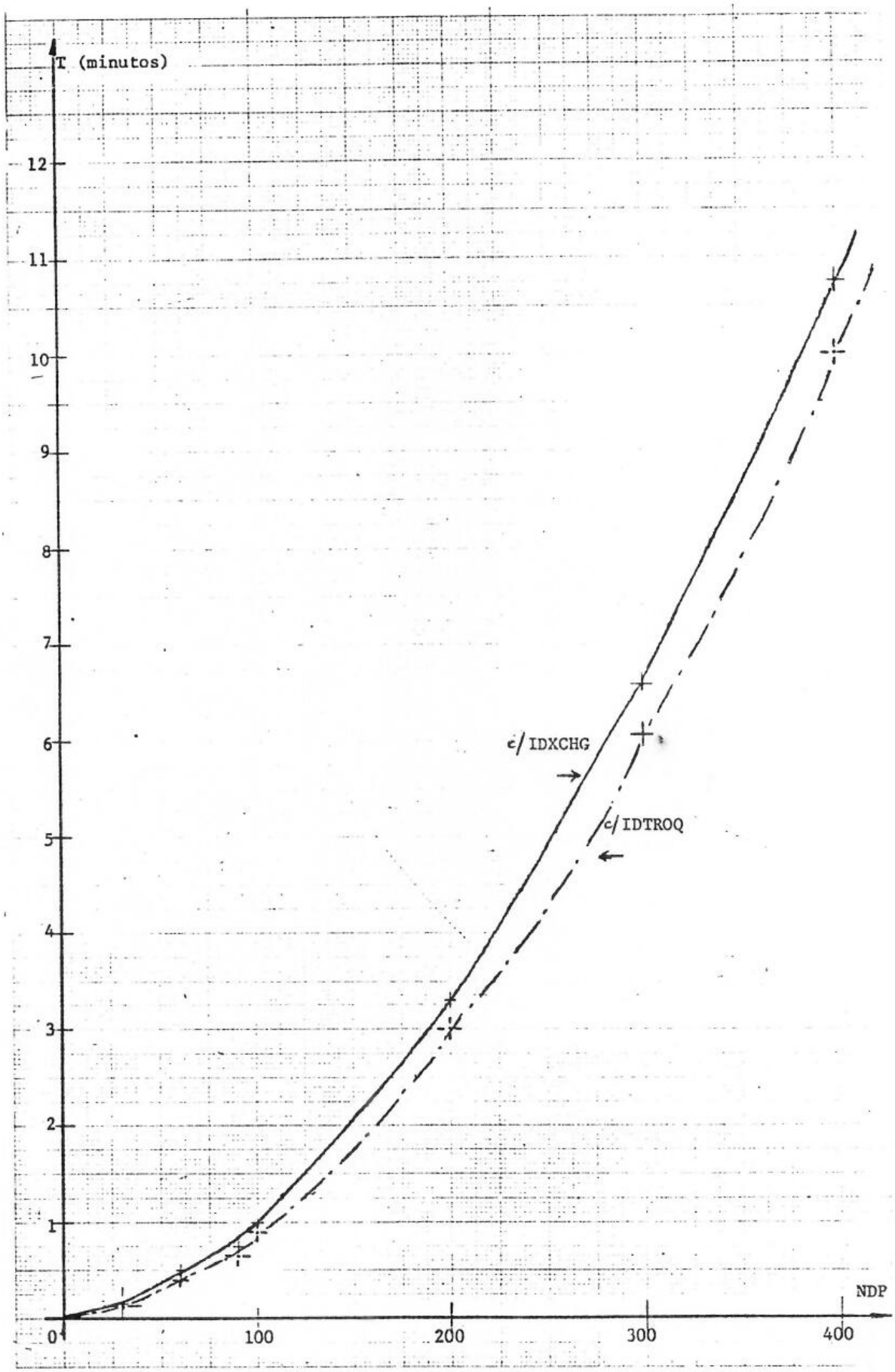


Figura 5.11 - Comparação de tempos da rotina TRIANG, com IDXCHG x com IDTROQ

triângulos. No exemplo acima, feito só para a medição de tempo, o número de chamada foi de 9.700. Para verificar o que acontece em casos concretos tomamos exemplos de 30, 60, 90, 100, 200, 300 e 400 pontos cujos resultados figuram no Quadro 5.5 e na Figura 5.11 (gráfico).

Quadro 5.5 - Comparação de tempos: IDXCHG × IDTROQ

	Tempos da função IDTANG						
	EX 30	EX 60	EX 90	EX 100	EX 200	EX 300	EX 400
c/ IDXCHG	9s	30s	46s	1m 02s	3m 19s	6m 34s	10m 46s
c/ IDTROQ	8s	25s	41s	54s	3m 01s	6m 05s	10m 03s
Diferença	1s	5s	5s	8s	18s	29s	43s

5.4.4. CONCLUSÃO

Pelos testes comparativos pode-se ver que há realmente uma melhora com a introdução do critério do círculo, tanto do ponto de vista de memória como do tempo de computação. Do ponto de vista geométrico é equivalente à função anterior já que as duas produzem a triangulação de Delaunay.

A maior ou menor relevância dessa melhora vai depender do problema concreto.

5.5. APLICAÇÃO DA TÉCNICA DE DIVISÃO E CONQUISTA AO MÉTODO DE AKIMA

Trata-se de um processo delineado por Lee & Schachter (1980), que procuramos detalhar e aplicar ao caso concreto.

De acordo com nossa proposta, aproveitaria o algoritmo de Akima, utilizando-o diversas vezes em conjuntos menores.

A filosofia é dividir o conjunto inicial em pequenos subconjuntos, realizando a seguir a união. É a chamada técnica da divisão e conquista (divide-and-conquer) aplicada com bastante sucesso a diversos outros campos.

5.5.1. VANTAGENS

As principais vantagens referem-se à diminuição do tempo de computação e à redução de memória necessária.

Lee e Schachter demonstram que esse enfoque leva a um tempo de computação que é O(n log n). Um exemplo didático pode ilustrar a diminuição dos tempos de computação.

De acordo com os testes realizados (Figura 5.11 e tabela correspondente), o tempo gasto na triangulação pelo método de Akima é de 8 segundos para 30 pontos, 10 m e 03 s para 400 pontos e aproximadamente 9 m e 30 s para 390 pontos.

Fracionando um problema de 390 pontos em 13 conjuntos de 30 pontos gastaríamos um tempo bem menor. Esse tempo poderia ser entendido como composto de duas parcelas. A primeira referente à triangulação dos conjuntos separados e a segunda à união dos mesmos para formar o todo.

O tempo para realizar a união das partes depende somente de alguns testes com o critério do círculo e varia de acordo com o número de pontos da fronteira. Por isso,podemos avaliar, por cima e com margem de segurança, uns 3 segundos por união,

em média.

O tempo total de computação pode então ser calculado:

8 s•13 conjuntos + 3 s•12 uniões = 104 + 36 = 2 m 20 s.

Esse valor é bem inferior ao tempo de 9 m 30 s gasto para a solução do problema como um todo indivisível. O tempo caiu para 1/4 do tempo previsto.

A técnica da divisão e conquista permite realizar a triangulação por partes, permitindo trabalhar com conjuntos maiores de pontos, processando-os separadamente e realizando a união.

A seguir descrevemos as linhas gerais para a implantação desse algoritmo.

5.5.2. NOTAÇÃO E ESTRUTURA DE DADOS

Para cada vértice v_i, armazenaremos uma lista ordenada de pontos vizinhos (adjacência) de v_i: $v_1,\ldots,v_k$, onde (v_i,v_j) $j = 1,\ldots k$, é uma aresta de Delaunay. A lista é duplamente unida e circular.

Definimos a seguir duas funções:

$PRED(v_i,v_j)$ - denota o ponto v_r que aparece imediatamente depois de v_j na lista de adjacência de v_i, no sentido horário (conforme Figura 5.12). Por exemplo,

$$PRED(v_1,v_3) = v_2$$
$$PRED(v_1,v_6) = v_5$$

$SUCC(v_i, v_j)$ - denota a função análoga, contando no sentido anti-horário. Por exemplo, na Figura 5.12 temos:

$$SUCC(v_1, v_2) = v_3$$

$$SUCC(v_1, v_3) = v_4$$

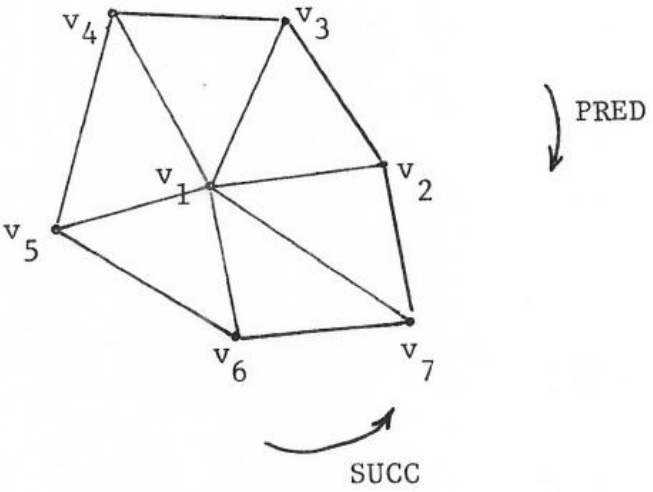


Figura 5.12 - Definição das funções SUCC e PRED, relativas à vizinhança de um vértice.

Se o ponto v_i pertence à fronteira convexa de V (convex hull), que denotaremos por CH(V), então a primeira entrada em sua lista de adjacência é o ponto denotado por FIRST (v_i), que

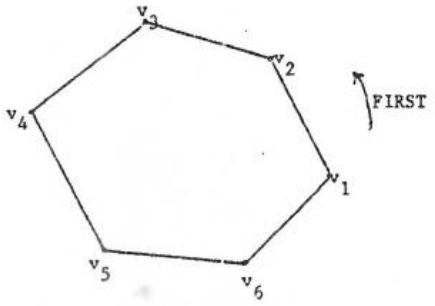


Figura 5.13 - Construção da função FIRST, relativa a pontos vizinhos sobre a fronteira convexa (v_1 a v_6).

aparece depois de v_i, se examinamos a fronteira CH(V) no sentido anti-horário. Na Figura 5.13 temos $FIRST(v_1) = v_2$. Caso se construa ordenadamente a função SUCC pode-se dispensar a função FIRST.

Seguindo essa notação e utilizando essas funções passemos à construção.

5.5.3. CONSTRUÇÃO DA TRIANGULAÇÃO DE DELAUNAY

Para dividir os pontos em subconjuntos ordenamos os N pontos em ordem lexicograficamente crescente e renumeremos os índices de forma que $v_1 < v_2 < v_3 < \ldots < v_n$ tal que

$$(x_i, y_i) = v_i < v_j$$

se e somente se $x_i < x_j$ ou $(x_i = x_j$ e $y_i < y_j)$.

A seguir dividimos V em dois subconjuntos V_L (Left = esquerda) e V_R (Right = direita), tal que

$$V_L = \left\{v_1, \ldots, v_{\lfloor N/2 \rfloor}\right\}$$

e

$$V_R = \left\{v_{\lfloor N/2+1 \rfloor}, \ldots, v_n\right\}.$$

Depois construímos recursivamente as triangulações de Delaunay $DT(V_L)$ e $DT(V_R)$. Para fundir as duas $DT(V_L)$ e $DT(V_R)$ de modo a formar $DT(V_L \cup V_R)$, fazemos uso da fronteira convexa $CH(V_L \cup V_R)$. Essa fronteira convexa pode ser obtida em tempo O(N), como demonstram Preparata & Hong (1977), a partir da união das fron-

teiras convexas $CH(V_L)$ e $CH(V_R)$. A fronteira convexa também pode ser calculada recursivamente. Formando a união de $CH(V_L)$ e $CH(V_R)$ resultará em duas novas arestas exteriores que são a tangente inferior (TI) e a tangente superior (TS) comuns a $CH(V_L)$ e $CH(V_R)$. Essas duas tangentes comuns pertencem à triangulação final de Delaunay.

A sub-rotina HULL, descrita a seguir encontra a tangente inferior de $CH(V_L)$ e $CH(V_R)$. Para cada fronteira convexa CH(S), deve-se conservar os pontos LM(S) e RM(S) que são os pontos mais a esquerda e mais à direita de S, repectivamente, e que podem ser determinados no momento da partição em subconjuntos.

Sub-rotina HULL (fronteira)

COMENTÁRIO: Os dados de entrada de HULL são duas fronteiras convexas. Ela encontra a tangente comum inferior (Figura 5.14). A tangente comum superior pode ser encontrada de maneira análoga.

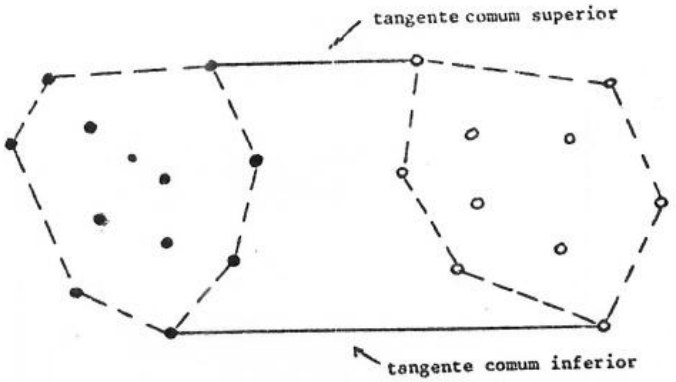


Figura 5.14 - União de duas fronteiras convexas através da determinação de 2 tangentes comuns.

HULL

```
X ← RM(VL); Y ← LM(VR)
Z ← FIRST(Y); Z' ← FIRST(X); Z* ← PRED(X,Z')
A:      IF(Z is-right-of l(X,Y))
        Z ← SUCC(Z,Y)
        Y ← Z
       ELSE
         IF (Z' is-right-of l(X,Y)
         Z* ← PRED(Z*,X)
         X ← Z*
         ELSE
         RETURN (X,Y)
         ENDIF
        ENDIF
        GO TO A
END HULL
```

$l(X,Y)$ = segmento (linha) orientado de X para Y.

A tangente comum inferior será utilizada como dado de entrada para a sub-rotina seguinte, que funde as duas triangulações parciais de Delaunay $DT(V_L)$ e $DT(V_R)$. A tangente superior servirá para indicar o fim do processo.

Sub-rotina MERGE (fusão)

COMENTÁRIOS:

* Os dados de entrada são duas triangulações e as tangentes comuns, inferior e superior, de suas fronteiras convexas. Ela funde as duas triangulações, partindo com a tangente comum inferior e ziguezagueando para cima até encontrar a tangente comum superior.

* Inicialmente os pontos adjacentes aos pontos extremos da tangente comum inferior são examinados. Utilizando o critério do círculo, conectam-se:

1. O extremo esquerdo (em V_L) da tangente comum inferior com o ponto adjacente ao extremo direito (em V_R) da tangente comum inferior; ou

2. O extremo direito (em V_R) da tangente comum inferior com o ponto adjacente ao extremo esquerdo (em V_L) da tangente comum inferior.

O processo acima é repetido, com a nova aresta encontrada tomando o lugar da tangente comum inferior, e cada nova aresta sucessiva tomando o lugar desta. A sub-rotina continua dessa forma até que a tangente comum superior seja encontrada.

* A lista de adjacência do extremo direito (esquerdo) em $V_R(V_L)$ da aresta atual em consideração é examinada no sentido horário (anti-horário), partindo com o extremo esquerdo (direito) da aresta.

* INSERT (A,B), insere o ponto A na lista de adjacência de B e o ponto B na lista de adjacência de A, na posição correspondente.

* DELETE (A,B), retira A da lista de adjacência de B, e B da lista de adjacência de A.

* QTEST (H,I,J,K) testa o quadrilátero que possui como vértices, no sentido anti-horário, H, I, J e K. Ele retorna TRUE (um) se a circunferência do triângulo H, I, J não contém K em seu interior, e retorna FALSE (zero), caso contrário. É o critério do círculo.

Tendo em conta esses comentários, pode-se entender mais facilmente o algoritmo da sub-rotina MERGE, que apresentamos

a seguir.

São 24 passos que devem ser examinados cuidadosamente para entender a lógica utilizada, servindo-se dos comentários acima.

```
                          MERGE
step 1:  INITIALIZATION
         BT ← lower common tangent
         UT ← upper common tangent
          L ← left endpoint of BT
          R ← right endpoint of BT
step 2:        DO UNITL (BT equals UT)
step 3:        A ← B ← FALSE
step 4:        INSERT(L,R)
step 5:        R1 ← PRED(R,L)
step 6:          IF(R1 is-left-of l(L,R))
step 7:          R2 ← PRED(R,R1)
step 8:            DO UNTIL (QTEST(R1,L,R,R2))
                   DELETE(R,R1)
                   R1 ← R2
                   R2 ← PRED(R,R1)
                   END DO UNTIL

step 9:          ELSE
                 A ← TRUE
                 ENDIF
step 10:       L1 ← SUCC(L,R)
step 11:         IF (L1 is-right-of l(R,L)
step 12:         L2 ← SUCC(L,L1)
step 13:           DO UNTIL (QTEST(L,R,L1,L2))
                   DELETE(L,L1)
                   L1 ← L2
                   L2 ← SUCC(L,L1)
                   END DO UNTIL
step 14:         ELSE
                 B ← TRUE
                 ENDIF
step 15:         IF(A)
                 L ← L1
step 16:         ELSE
step 17:          IF(B)
                  R ← R1
step 18:          ELSE
step 19:           IF(QTEST(L,R,R1,L1))
                   R ← R1
step 20:           ELSE
                   L ← L1
                   ENDIF
step 21:          ENDIF
step 22:         ENDIF
step 23:       BT ← L(L,R)
               END DO UNTIL
step 24:   END MERGE
```

Para mostrar que esse algoritmo funde as duas triangulações corretamente, é suficiente mostrar que cada aresta inserida na triangulação é uma aresta de Delaunay (passo nº 4). Inicialmente, a primeira aresta (L,R) é a tangente comum inferior, que é sabidamente uma aresta de Delaunay.

Os passos 5-8 eliminam as arestas em $DT(V_R)$ que não são arestas de Delaunay em DT(V), determinando se L está dentro do circuncírculo do triângulo R, R_1, R_2. Se isto acontece, a aresta (R,R_1) não é uma aresta de Delaunay e deve ser eliminada.

Os passos 10-13 realizam a mesma operação nas arestas de $DT(V_L)$.

Um exemplo desse processo é mostrado na Figura 5.15.

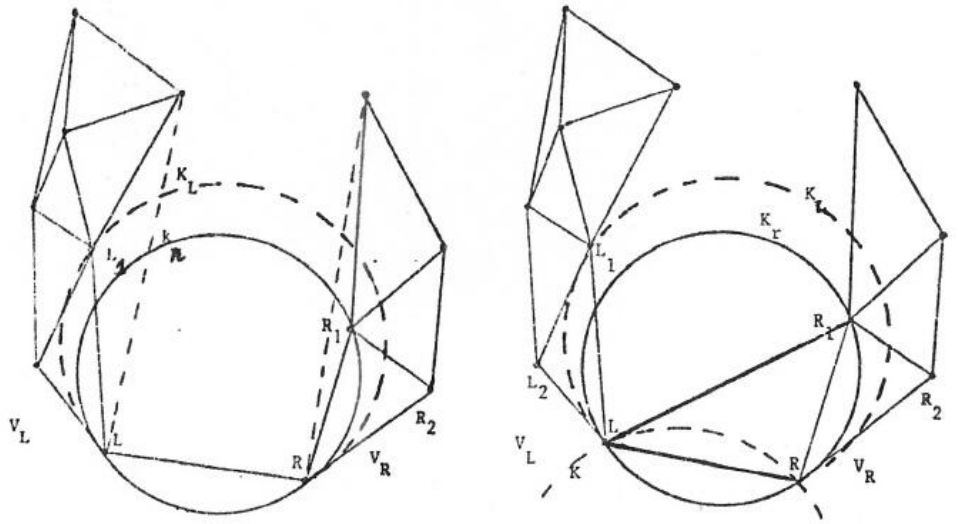


Figura 5.15 - Ilustração do processo de fusão. As linhas cheias são as aresta de Delaunay introduzidas pelo algoritmo MERGE.

O circuncírculo K_r do triângulo L, R, R_1 não contém nunhum ponto de V_R em seu interior e o circuncírculo K_L do tri-

ângulo R, L, L_1 não contém nenhum ponto de V_L em seu interior. Agora devemos escolher, como se vê na Figura 5.15, entre conectar L_1 com R ou R_1 com L.

O passo 19 escolhe a aresta correta aplicando o critério do círculo. Na figura em questão, K_L contém R_1 em seu interior, e então escolhe-se a aresta (L,R_1).

Tudo o que se fez foi mostrar que a aresta (L,R_1) é uma aresta de Delaunay, ou, o que é equivalente, demonstrar que o círculo K_R não contém nenhum ponto de V_L em seu interior.

Desde que a aresta (L,R) é uma aresta de Delaunay, existe um círculo K (Lema 2 do item 5.3.3) passando por L e R que não contém nenhum ponto de V no seu interior. É sabido também que o círculo K_L não contém nenhum ponto de V_L em seu interior. Como o círculo K_R cai dentro da união de K e K_L, segue-se que K_R não contém nenhum ponto de V_L em seu interior. Combinando esse resultado com o fato de que K_R não contém nenhum ponto de V_R em seu interior, conclui-se que K_R não contém nenhum ponto de $V(V_L \cup V_R)$ em seu interior . Então a aresta (L,R_1) é uma aresta de Delaunay.

A aresta (L,R_1) pode ser agora inserida para tomar o lugar da aresta (L,R). Na demonstração de que a próxima aresta a ser incluída depois de (L,R_1) é uma aresta de Delaunay, o círculo K_R representa o mesmo papel que o círculo K fez na demonstração primitiva.

5.5.4. ANÁLISE DE TEMPOS E CONCLUSÃO

Para a análise do tempo, notemos em primeiro lugar que du-

rante o processo de fusão, uma vez que uma aresta é eliminada, ela nunca mais é reexaminada. Como o total de arestas eliminadas é $O(n)$ e o número total de arestas acrescentadas também é $O(n)$, o tempo requerido para a fusão é $O(n)$. Suponhamos que $t(n)$ representa o tempo necessário para construir a triangulação de Delaunay para um conjunto de n pontos. Temos então a seguinte fórmula de recorrência:

$$t(n) = 2 \cdot t(n/2) + M(n/2, n/2)$$

$$t(1) \cong 0,$$

onde M representa o tempo necessário para o processo de fusão. Como $M(n/2,n/2) = 0\ (n)$; então, pelo cálculo temos:

$$t(n) = n \cdot \log n.$$

Shamos & Hoey (1975) mostraram que a construção de qualquer triangulação sobre n pontos requer um tempo cujo piso inferior é $\Omega(n \cdot \log n)$. Então, este algoritmo é assintoticamente ótimo.

* CONCLUSAO: Fizemos uma simulação deste algoritmo com o exemplo de Akima, realizando a divisão de 15 pontos para cada lado e fundindo as duas metades. Foi realizada a construção geométrica e todos os passos de HULL e MERGE funcionaram.

Algumas das rotinas utilizadas por este método já estão prontas, por exemplo, PRED, SUCC e FIRST decorrem imediatamente da lista de vizinhos de Voronoi (já desenvolvida por nós com IDVZVR); QTEST é o critério do círculo (desenvolvido por nós com IDTROQ); para encontrar a fronteira convexa pode-se utilizar o programa de Eddy (1977). No entanto,

tendo em conta que o tempo do atual algoritmo não é crítico e que a capacidade máxima para micros está razoável, pareceu-nos preferível concentrar esforços nos complementos do programa de Akima, que permitem chegar até o desenho final das curvas de nível.

Numa segunda etapa pensamos implantar este método.

5.6. CONTRIBUIÇÃO PARA A DETERMINAÇÃO DE VIZINHOS

Existe na literatura computacional dos últimos 10 ou 15 anos um número muito grande de artigos dedicados à busca de vizinhos; são conhecidos como "nearest neighbours" ou "closest point problem".

Dados n pontos trata-se de encontrar para cada um deles os n_c vizinhos mais próximos. Numa resolução direta, a complexidade e o tempo computacional aumentam exponencialmente com o número de pontos n. Por isso desenvolveram-se diversos algoritmos para minimizar esse tempo. Utilizam-se diversos enfoques: divisão e conquista; árvores binárias 2-d; divisão de Voronoi, etc.

O próprio Akima preocupou-se em melhorar a eficiência do algoritmo que utilizava inicialmente e implementou consideráveis melhoras, conforme descrição da sub-rotina IDCLDP (closest data point), item 4.3.3. De qualquer forma, ainda consome um bom tempo de computação. Na versão atual pode-se escolher o número de vizinhos a serem levados em consideração (NCP), que pode variar de 1 a 25, sendo que o autor aconselha a utilização de 3, 4 ou 5.

Outro autor, Pettinati (1983) também se preocupa por reduzir o tempo gasto na procura de vizinhos e para isso optou por montar uma complexa estrutura de dados, no intuito de ganhar tempo no cômputo geral.

Consultamos a bibliografia sobre o tema e incluímos algumas referências bibliográficas neste trabalho. No entanto, parece-nos que um enfoque diferente pode dar melhores resultados: escolher os vizinhos de Voronoi.

Desenhando e familiarizando-se com a triangulação de Delaunay, percebe-se a vantagem dessa solução.

Dessa forma, obedeceríamos melhor a configuração geométrica e cada ponto teria um número de vizinhos variável,o que é mais lógico pois, por exemplo, os pontos da fronteira tem características diferentes dos internos.

Esse critério define vizinhos mais "naturais" pois não seriam pré-fixados e teriam as vantagens ditadas pela teoria de Voronoi: esses são os vizinhos que, além de próximos, distribuem-se espacialmente de forma otimizada.

Por outro lado, sabemos que os segmentos que unem o vértice em estudo com os seus vizinhos de Voronoi definem as arestas de Delaunay, e a função interpoladora será aplicada exatamente sobre estas arestas e sobre os triângulos formados por estas arestas. Por isso, nada mais justo que adotar esses pontos como vizinhos, para o cálculo das derivadas em cada ponto.

Esse critério – vizinhança de Voronoi – pode ser uma

resposta a uma indagação implícita que Akima coloca em seu trabalho: "não se vê que exista um critério, uma teoria que indique um valor definido para n_c (número de vizinhos)". O valor de n_c passaria então a ser definido pela configuração topológica da triangulação de Voronoi e seria variável para cada ponto.

Realizamos a implementação computacional, tendo em conta que já dispomos da listagem completa dos triângulos, cada um deles com os vértices armazenados no sentido anti-horário. Para determinar os vizinhos de um ponto separamos todos os triângulos em que esse vértice figura. Os outros 2 vértices de cada triângulo são certamente seus vizinhos de Voronoi. A seguir eliminamos os vizinhos que aparecem duas vezes.

De acordo com a implementação feita (rotina IDVZVR), o tempo de computação diminuiu sensivelmente com relação à rotina original (IDCLDP).

Em termos de memória, o fato de o número de vizinhos ser variável não atrapalha muito. No caso de um número fixo de vizinhos pode-se montar uma matriz de NDP*NCP. Seguir esse mesmo esquema para um número de vizinhos variável nos obrigaria a definir um número máximo de vizinhos para cada ponto e perder algumas posições de memória. Akima estima o máximo em 25, o que daria, para 400 pontos, 10.000 posições de memória.

Para superar esse inconveniente organizamos uma lista seqüencial, armazenando em outra lista o número de vizinhos de cada ponto. O número máximo de posições de memória pode ser calculado facilmente: é igual ao dobro do número máximo de a-

restas. Assim, pela fórmula teórica (item 5.3.3) temos:

$$NVZ_{max} = 2*NA_{max} = 2*(3*(NDP-1)-NL_{min}) = 2*(3*NDP-3-3) =$$

$$= \underline{6*NDP - 12} \qquad (1)$$

No caso concreto, de 400 pontos $NVZ_{max} = 2.388$;somando as 400 posições para armazenar o número de vizinhos de cada vértice teremos um total de 2.788 posições contra as 10.000 previstas anteriormente.

O número de vizinhos num esquema como esse não difere muito dos valores empíricos propostos por Akima; entre 3 e 5.Para a configuração da Figura 5.7, a média calculada de vizinhos foi de 5,07. Se fizermos uma distinção entre pontos de borda e pontos internos, a média seria de 4,4 e 5,5 respectivamente.

Teoricamente poderíamos pensar numa comparação tomando uma triangulação totalmente equilátera que produziria uma configuração de hexágonos regulares. Cada ponto interno teria exatamente 6 vizinhos enquanto que os pontos de borda teriam 4 e a média geral seria um número entre 4 e 6, que dependeria da proporção entre pontos de borda e pontos internos. O limite máximo para a média de vizinhos é inferior a 6, como se pode ver dividindo a fórmula (1) acima por NDP.

5.7. IMPLANTAÇÃO DA DETERMINAÇÃO DOS VIZINHOS DE VORONOI. COMPARAÇÕES

(Rotina IDVZVR)

5.7.1. ESQUEMA

De acordo com a teoria vista no item 5.6, desenvolvemos a rotina IDVZVR (ViZinhos de VoRonoi) para determinar quais são esses vizinhos para cada vértice.

O esquema é o seguinte:

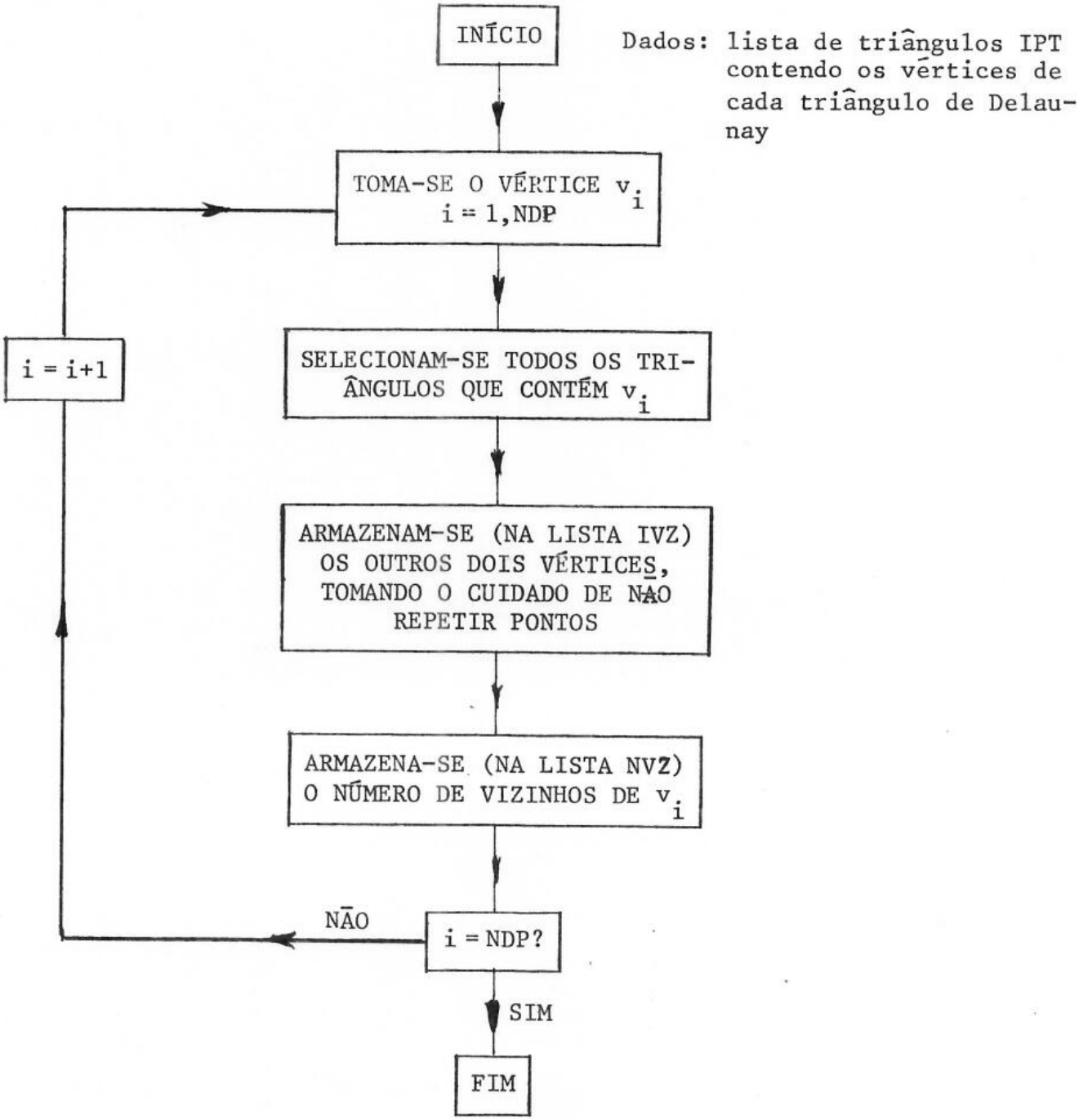


Figura 5.16 - Esquema da rotina IDVZVR

5.7.2 COMPARAÇÃO DE MEMÓRIAS

De acordo com as listagens (.PRN) montamos o quadro comparativo 5.6 entre a nova rotina (IDVZVR) e a rotina equivalente original (IDCLDP).

Quadro 5.6 - Comparação de memórias

Memória (bytes)	IDCLDP	IDVZVR
Programa	1.511	681
Dados	516	225
Total	2.027	906
Diferença		1.121

Há portanto uma economia de memória, com relação às rotinas consideradas em si mesmas.

Com relação às áreas não computadas acima, por estarem em regiões comuns (COMMON), a nova rotina utiliza as matrizes IVZ (com dimensão 6*NDP) e NVZ (com dimensão NDP), contra a matriz IPC (com dimensão 4*NDP) da rotina anterior.

Essa diferença (3*NDP) não deve ser computada pois utilizamos para IVZ e NVZ a mesma área IWL (das posições 14*NDP+1 a 21*NDP), que é uma área de trabalho, já computada em IDTANG.

Com isso, a área de dados não sofre acréscimo com a introdução de IDVZVR, e a área de programa diminui.

Com fracionamento do programa para adaptação a microcomputadores pode haver uma ligeira desvantagem para a nova rotina. De qualquer forma, esse problema não é crítico e as modificações são altamente vantajosas pela melhora propiciada na qualidade do desenho e no tempo de computação.

5.7.3. COMPARAÇÃO DOS TEMPOS (IDVZVR × IDCLDP)

Na comparação de tempos (pelo equipamento Polymax 201 DP), levamos em conta que ambas são rotinas básicas chamadas pelo programa principal e que os demais elementos permaneceram inalterados, podendo-se comparar diretamente os tempos de cada uma.

Para isso tomamos sete exemplos com 30, 60, 90, 100, 200, 300 e 400 pontos. Os resultados são os apresentados na Tabela do Quadro 5.7 e no gráfico da Figura 5.17.

Quadro 5.7 - Comparação de tempo: IDCLDP (4 e 6 vizinhos) × IDVZVR

	Tempos de diversos exemplos						
	EX 30	EX 60	EX 90	EX 100	Ex 200	EX 300	EX 400
1. IDCLDP (4 viz.)	7s	19s	33s	48s	3m02s	6m41s	11m52s
2. IDCLDP (6 viz.)	7s	21s	36s	50s	3m10s	6m58s	12m47s
3. IDVZVR	3s	5s	9s	10s	34s	1m11s	2m07s
Diferença 1-3	57%	74%	73%	79%	81%	82%	82%
Diferença 2-3	57%	76%	75%	80%	82%	83%	83%

5.7.4. CONCLUSÃO

A nova rotina é bem mais rápida que a anterior e possibilita uma significativa economia de tempo, principalmente para problemas maiores. Isso se deve à forma de variação exponencial da rotina IDCLDP, que se manifesta quer se empreguem 4 ou 6 vizinhos por ponto.

A introdução de novos vizinhos com IDVZVR melhora também

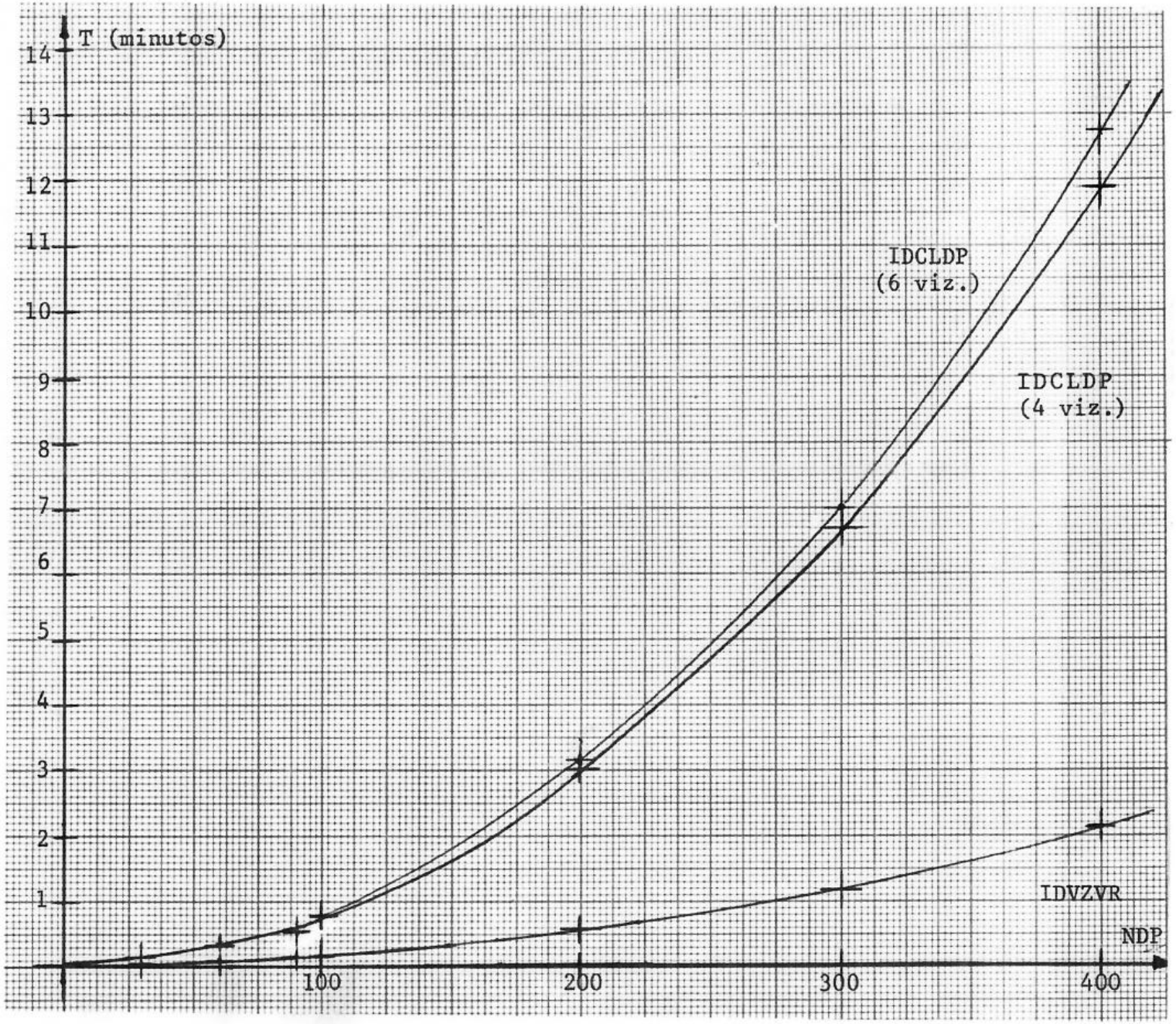


Figura 5.17 - Comparação de tempos: IDCLDP (4 e 6 vizinhos) × IDVZVR

a avaliação das derivadas e o desenho final, como veremos a seguir.

5.8. CONTRIBUIÇÕES PARA O CÁLCULO DAS DERIVADAS

A utilização dos vizinhos de Voronoi para o cálculo das derivadas parece mais razoável do que usar aqueles que resultaram de um número fixado a priori já que o novo critério tem em conta a distribuição espacial dos pontos.

O método de Akima para o cálculo das derivadas utiliza produtos vetoriais, o que acarreta a vantagem de realizar uma média ponderada que leva em conta não só as distâncias mas também a direção em que se encontram os pontos. Outros métodos só levam em conta as distâncias entre os vizinhos escolhidos e o vértice em questão.

No entanto, a escolha dos vizinhos em Akima é função somente da distância (pontos mais próximos), o que pode fazer com que os pontos escolhidos para o cálculo das derivadas tenham posições pouco convenientes, do ponto de vista da distribuição espacial. O polígono de Voronoi e a triangulação de Delaunay asseguram uma boa distribuição espacial dos vizinhos (otimizada), pois estes são escolhidos também em função dos ângulos. Evita ao máximo que sejam escolhidos pontos alinhados.

As distorções que podem ocorrer na escolha dos vizinhos tendo em conta somente a distância ficam mais claras ao pensarmos em alguns exemplos.

Suponhamos novamente uma rede de triângulos quase equiláteros proporcionando a formação de hexágonos. Cada ponto interno terá 6 vizinhos situados praticamente à mesma distância. Se por exemplo fixarmos em 4 o número de vizinhos estaremos eliminando 2 deles em função de sua maior distância com relação ao ponto central. Supondo que esses pontos são os de nn. 2 e 3 na Figura 5.18, vemos claramente que haverá uma distor-

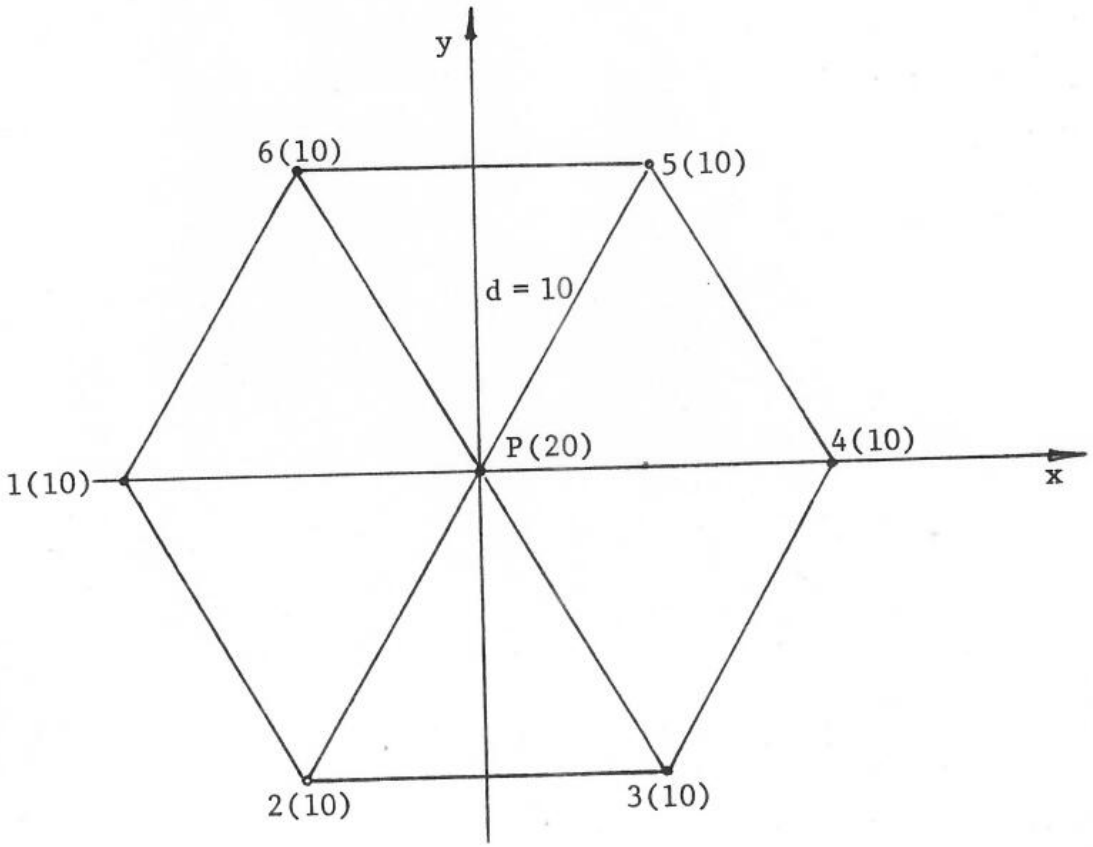


Figura 5.18 - Exemplo de situações anômalas que podem ocorrer quando se escolhem os vizinhos em função somente da distância: só se contam com pontos em um semi-espaço (1,4,5 e 6) para definir a derivada na direção y.

ção no cálculo da derivada na direção y, que passará a ser ditada somente pelos pontos 5 e 6. Se a cota (função z) do ponto P for 20 m e a dos demais pontos for 10 m, a derivada na

direção y será a inclinação do plano P-5-6 que é igual a 1,0 (supondo uma distância de 10 m em y), quando na realidade, levando em conta 2 e 3, essa derivada deveria ser aproximadamente nula.

Outras deformações podem ocorrer nas bordas ou nos cantos, por tomarmos um número de vizinhos inadequado. Pode ser que um canto só tenha 2 vizinhos de Voronoi e fixamos 5 (ou 6) vizinhos para todos os pontos. Pode acontecer que estejamos tomando pontos que pouco tem a ver com o ponto em questão (pouco correlacionados fisicamente), pode haver uma linha cumeada ou um vale em 1-2 ou 3-4 (conforme Figura 5.19) que tornam sem sentido uma interpolação baseada em pontos que estão além dessas linhas. A distribuição de Delaunay minimiza esses problemas.

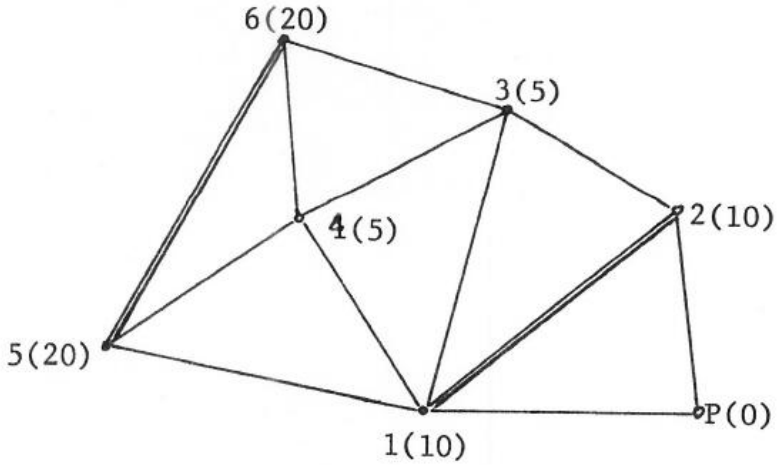


Figura 5.19 - Exemplo de como a fixação de um número arbitrário pode levar a distorções no cálculo das derivadas, para um ponto situado em uma borda.

Outras possíveis contribuições ao cálculo das derivadas

referem-se à forma de calculá-las. Existem outros métodos que não através de produtos vetoriais para avaliá-las, por exemplo, um polinômio completo do 2º grau ajustado a cada ponto e seus 5 vizinhos, ou ainda através da média simples ou ponderada das inclinação das arestas de Delaunay que partem de cada vértice. Esses procedimentos conduzem a estimativas piores que a de Akima. No entanto, uma pesquisa em torno deste ponto pode resultar em melhorias.

Tenha-se em conta a influência desse fator na representação da superfície: mesmo que a função interpoladora seja a mesma, os coeficientes variam em função da avaliação das derivadas. Conseqüentemente os desenhos serão diferentes, de melhor ou pior qualidade.

5.9. IMPLEMENTAÇÃO DO NOVO CÁLCULO DE DERIVADAS. COMPARAÇÕES

(IDPDRV x IDPDER)

5.9.1. COMPARAÇÃO DE MEMÓRIA E DE TEMPOS

Computacionalmente as modificações na rotina IDPDRV, que realiza o cálculo das derivadas, foram mínimas. Bastou fazer NCP variável, de acordo com os elemento fornecidos por IDVZVR (lista IVZ). Dessa forma, a memória ocupada é a mesma para as duas rotinas.

Com relação ao tempo de computação, a relação dos tempos vai depender do número de vizinhos (NCP) adotado por IDPDRV pa-

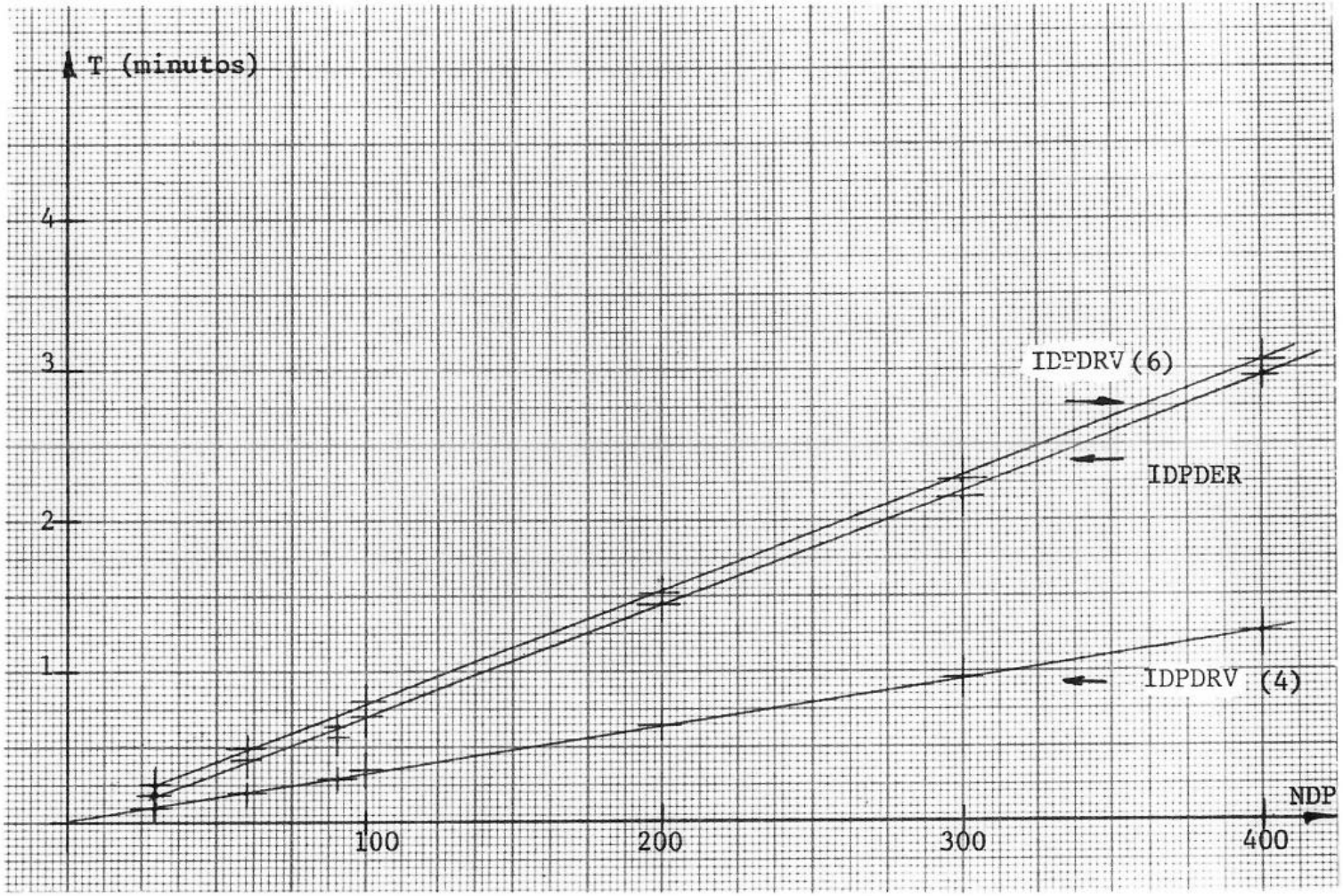


Figura 5.20 - Comparação de tempos: IDPDRV (4 e 6 vizinhos) × IDPDER

ra o cálculo das derivadas. Se o número de vizinhos for 6, o tempo das duas é praticamente o mesmo pois a média de vizinhos (de Voronoi) em IDPDER é no máximo 6 e em geral ultrapassa a média de 5. Se o número de vizinhos for 4, o tempo de IDPDRV será um pouco menor que o tempo da nova IDPDER. Essas previsões foram confirmadas pelos testes que realizamos e cujos resultados são os do quadro 5.8 e da Figura 5.20.

Quadro 5.8 - Comparação de tempos: IDPDRV × IDPDER

	Tempos de diversos exemplos						
	EX 30	EX 60	EX 90	EX 100	EX 200	EX 300	EX 400
IDPDRV (4 vz.)	7s	13s	17s	21s	38s	57s	1m 15s
IDPDRV (6 vz.)	14s	29s	38s	48s	1m 32s	2m 15s	3m 04s
IDPDER	10s	27s	35s	42s	1m 28s	2m 09s	2m 58s

5.9.2. COMPARAÇÃO DE DESENHOS

Melhorando a definição dos vizinhos, melhora a avaliação das derivadas e o ajuste da função interpoladora. Com isso as curvas de nível passam a representar melhor o terreno.

Em alguns casos testados essa diferença apareceu claramente. Com exemplo ilustrativo incluímos a seguir a comparação gráfica das diferenças que aparecem no próprio desenho-teste fornecido por Akima: os 30 pontos com asterisco na Tabela 3.2, representados na Figura 3.2.

Comparando-se os dois desenhos (Figuras 5.21 e 5.22) pode-se notar que a curva 15, não representada por Akima, apresenta um aspecto pouco razoável no trecho superior, no dese-

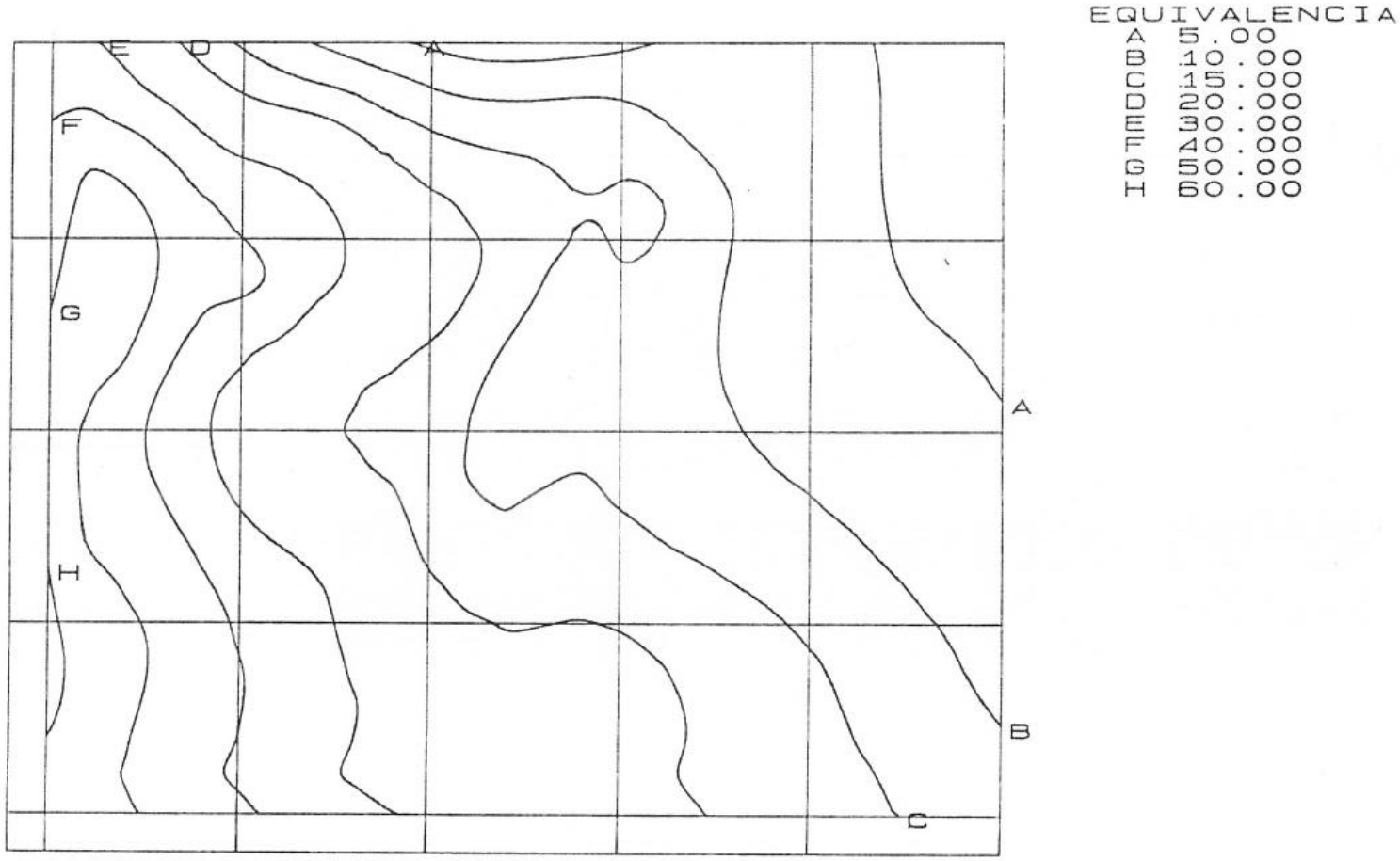


Figura 5.21 - Curvas de nível, método de Akima (4 vizinhos)

Configuração: 30 pontos com asterisco na tabela 3.3; 4 vizinhos para estimar as derivadas; malha regular com 26 x 21 (546) pontos; suavização do desenho através de splines sob tensão. Esse último dado técnico, junto com a malha mais aberta (a de Akima tem 80 x 100 pontos) faz com que o desenho tenha uma aparência, a nosso ver, melhor que o fornecido por Akima na figura 3.2 (e), p. 65.

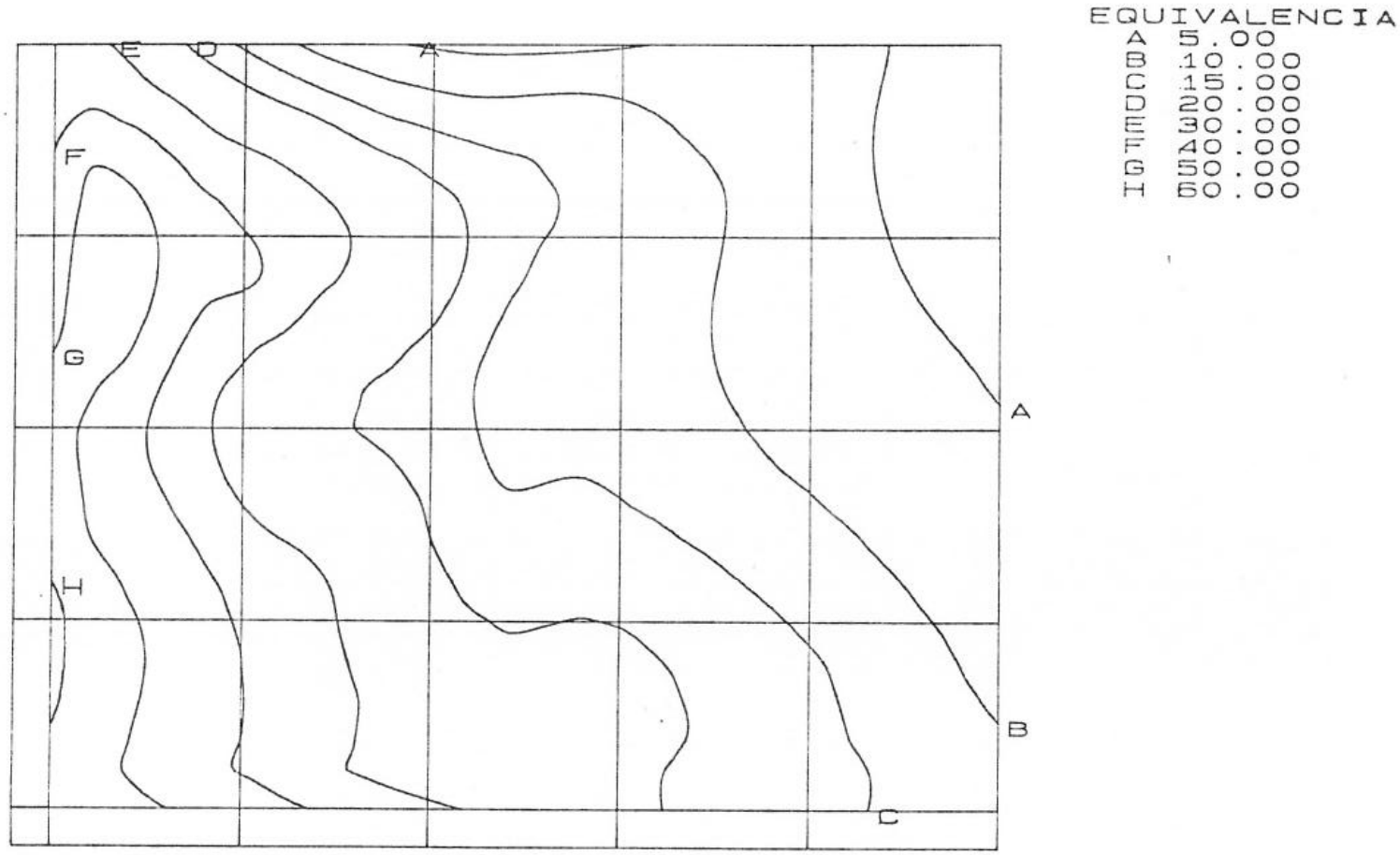


Figura 5.22 - Curvas de nível, método modificado (viz. de Voronoi)

Configuração: 30 pontos com asterisco na tabela 3.3; vizinhos de Voronoi para estimar as derivadas; malha regular com 26 x 21 (496) pontos; suavização do desenho através de splines sob tensão. A nova avaliação das derivadas evita o balão da curva (C), 15. Compare-se também com a superfície original e com o desenho de Akima, fornecidos na figura 3.2 (a) e (e), p. 65.

nho utilizando o método original. Isso não acontece na versão modificada, quando se empregam os vizinhos de Voronoi.

5.9.3. CONCLUSÃO

As memórias utilizadas são as mesmas e a relação dos tempos de computação depende do número de vizinhos adotado por IDPDRV. Em qualquer caso a diferença de tempos é pequena e a qualidade do desenho, com a nova rotina (IDPDER) é superior.

5.10. ESTUDO DA INTERPOLAÇÃO NO TRIÂNGULO

5.10.1. INTRODUÇÃO

As superfícies representativas do terreno podem ser classificadas segundo a classe de continuidade da função interpoladora (conforme Figura 5.23).

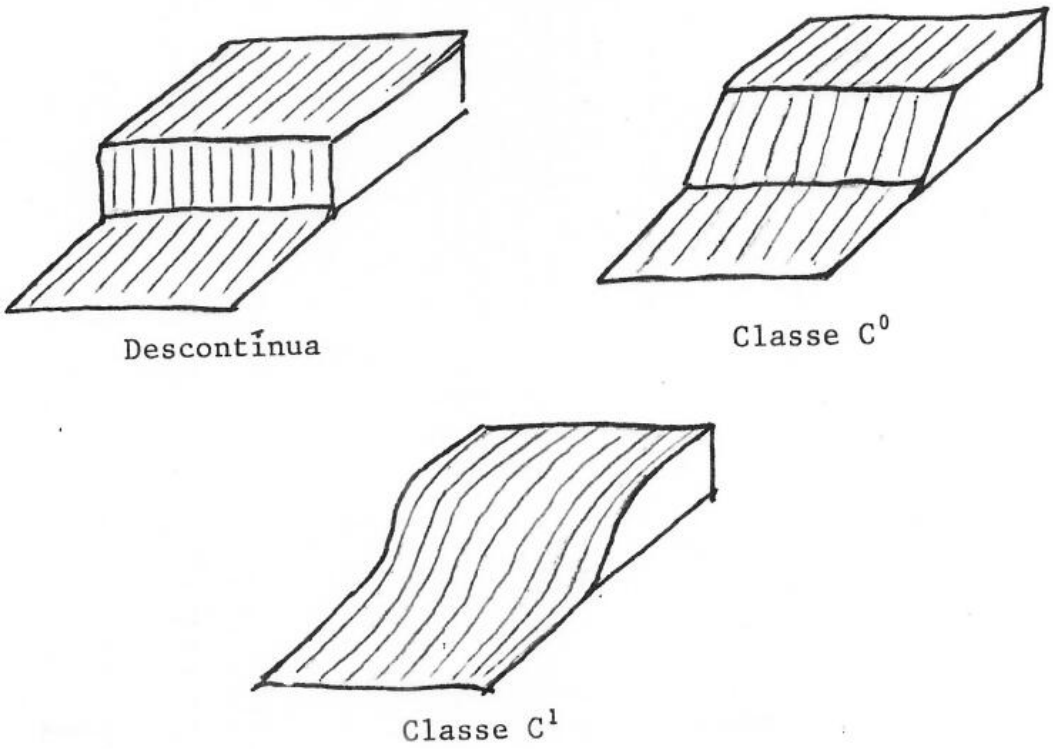


Figura 5.23 - Tipos de continuidade (classe) de superfícies.

Uma superfície é denominada de classe C^n se for continuamente derivável até a ordem n. Assim, uma superfície de classe C^0 é apenas contínua; uma superfície de classe C^1 é contínua e sua primeira derivada também o é; uma superfície de classe C^2 possui derivadas de 2ª ordem contínuas; e assim por diante.

As superfícies descontínuas, a menos de casos excepcionais, são completamente insatisfatórias para representar superfícies do terreno. Os primeiros métodos de representação de superfícies utilizaram equações de classe C^0, por exemplo, planos válidos em cada região triangular, como em Bengtsson & Nordbeck (1964). Mas por produzirem arestas vivas foram logo substituídos por superfícies de classe C^1, que são as mais utilizadas atualmente. Recentemente surgiram dois métodos que procuram trabalhar com superfícies de grandes variações de gradiente e que exigiram superfícies de classe superior a C^1. São eles: Franke e Nielson (1983) e MacCullagh (1983).

5.10.2. ALGUNS ESQUEMAS GERAIS DE INTERPOLAÇÃO

Existem diversos tipos de superfícies, aplicáveis a regiões diversas: retangulares, quadradas ou triangulares. Analisamos somente as que se referem especificamente a regiões triangulares pois são as que nos interessam mais de perto.

Do ponto de vista teórico, existem diversos estudos sobre esquemas gerais de interpolação, com aplicação para os triângulos, por exemplo, Boor (1962), Boor (1972), Hall (1973), Munteanu (1973), Munteanu & Schumaker (1974), Delvos (1975).

No entanto, a aplicação concreta exige um estudo e detalhamento complementar, e existem outros trabalhos mais específicos, cuja aplicação é quase imediata. Alguns deles estudam aspectos específicos, como detalhamos a seguir.

Barnhill e Mansfield (1974) estudam o problema da continuidade na fronteira, isto é, a suavidade dos valores na interface de duas regiões triangulares.

Barnhill, Birkhoff & Gordon (1973) estudam diversos esquemas gerais de interpolação no triângulo.

Barnhill & Gregory (1975) continuam o estudo anterior, dando especial ênfase à compatibilidade de superfícies vizinhas.

Hall (1969) apresenta fórmulas para a interpolação cúbica e bi-cúbica no triângulo, formulando três esquemas opcionais.

Powell & Sabin (1977) apresentam uma aproximação quadrática com continuidade C^1, utilizando os valores da função e da derivada primeira nos vértices para determinar os coeficientes da equação matemática. Apresentam três alternativas, que em sua metodologia fracionam o triângulo original em outros 4, 6 ou 12 sub-triângulos menores.

5.10.3. ESQUEMAS QUE UTILIZAM UMA ANALOGIA COM A TEORIA DOS ELEMENTOS FINITOS

Na teoria dos elementos finitos aplicada ao cálculo de estruturas, e particularmente para placas e cascas, empregam-se

diversos esquemas para definir a geometria das peças. Estas são subdivididas em pequenos elementos geométricos como por exemplo um retângulo ou um triângulo.

Impõe-se a seguir algumas condições de compatibilidade entre esses elementos componentes da peça, por exemplo, mesmos deslocamentos e rotações nos nós e interfaces, o que equivaleria a admitir uma função de classe C^1 para representar sua geometria.

Os elementos triangulares são muito atraentes pois são adaptáveis a diversos tipos de geometria, permitem trabalhar com fronteiras irregulares e pode-se variar o tamanho do elemento em locais de interesse, por exemplo, nas vizinhanças de áreas com concentração de tensões.

Muitos pesquisadores aprofundaram-se no estudo das características dos elementos triangulares, sugerindo diversos esquemas de representação que variam em função da forma matemática escolhida e dos graus de liberdade admitidos para a peça.

As funções matemáticas costumam ser polinômios bivariados (em x e y) e de um certo grau (3º, 5º, etc.), o que implica em um determinado número de coeficientes (parâmetros) a serem calculados. Para isso supõe-se conhecida a função e suas derivadas em alguns pontos, por exemplo, nos vértices do triângulo. Pode-se impor também alguma condição suplementar: grau do polinômio representativo das derivadas normais, valor da função no ponto médio dos lados, etc.

Em geral, procura-se que o número de parâmetros seja igual ao número de valores fornecidos, trabalhando-se dessa forma com um sistema determinado, de n equações a n incógnitas. Utilizam-se, por exemplo, famílias de curvas com 21, 18, 15 ou 12 parâmetros.

Dentre os trabalhos gerais sobre elementos finitos podem ser consultados entre outros os seguintes livros: Zienkiewicz (1967), Bell (1968), Bergan (1967). No campo de artigos específicos podem ser úteis os que indicamos a seguir.

Bramble e Zlamal (1970), estudam polinômios da classe C^n, com $n = 4 \cdot m+1$ $(m=0,1,2,...)$ aplicáveis a cada triângulo.

Birkhoff & Mansfield (1974) dão particular ênfase à compatibilidade dos elementos triangulares. Utilizam famílias de funções com 12 e 15 parâmetros. Foram obtidas a partir de outra família com 18 parâmetros, definida por Irons (1969), impondo restrições suplementares na fronteira e para os gradientes. Utiliza, num dos esquemas, um polinômio tricúbico (do 3º grau) e impõe valores nos vértices e nos pontos médios dos lados.

Zlamal (1968) discute um procedimento de aproximação utilizando um polinômio do 5º grau em x e y, sobre cada região triangular. Para determinar os coeficientes do polinômio utilizam-se os valores da função (z), das derivadas primeiras e segundas (z_x, z_y, z_{xx}, z_{xy}, z_{yy}) e também as três derivadas normais no ponto médio dos lados. Essa teoria foi generalizada por Zenisek (1970) para polinômios do grau $4 \cdot m+1$, para funções m vezes continuamente diferenciáveis num domínio triangular fe-

chado. Esses dois trabalhos sugeriram inúmeras idéias a Akima, como ele mesmo salienta.

Klucewicz (1978) desenvolveu um esquema de interpolação, com continuidade C^1, visando diretamente a aplicação para representar superfícies. O método é completo pois parte dos pontos fornecidos aleatoriamente e triangulariza-os. A seguir utiliza um interpolante que é a soma booleana proposta por Barnhill-Gregory e que garante a continuidade das derivadas na direção dos lados e também na direção perpendicular. Esse interpolante é determinado por 9 parâmetros (função e primeiras derivadas x e y; fornecidas nos vértices do triângulo) para 9 coeficientes. Fornece um exemplo prático, com os resultados e desenhos obtidos.

Gold, Charters & Ramsden (1977) também sugerem uma analogia com elementos finitos, apoiada no elemento conforme quártico proposto por Irons (1969).

Franke & Nielson (1980) analisam e comparam dois outros métodos que fornecem continuidade C^1. Utilizam, para os testes, os dados fornecidos por Akima.

Barnhill & Farin (1981) fornecem duas representações para uma interpolação quíntica (polinômio do 5º grau) em cada triângulo, garantindo a continuidade C^1. Como um polinômio completo do quinto grau possui 21 coeficientes, para determiná-los esses autores utilizam-se 18 dados (valor da função e as cinco derivadas primeiras e segundas, em cada vértice), juntamente com mais três condições: a derivada normal no ponto mé-

dio de cada lado. A condição de que a derivada normal ao longo de cada lado seja cúbica e não quártica também faria com que o polinômio se reduzisse ao P_{18} (18 coeficientes). Para obter as duas representações explícitas do P_{18} tomam-se como base os dois principais métodos empregados nos projetos geométricos assistidos por computador (CAGD): o esquema de Coon e a aproximação de Bernstein-Bézier, que são generalizados. Utilizam-se as coordenadas baricêntricas e indica-se, passo a passo, o modo de determinar os coeficientes. Esse esquema de interpolação (qualquer um dos métodos) é muito semelhante ao proposto por Akima (P_{18}; variação cúbica da derivada normal ao ao longo dos lados).

Caramanlian (1983) estuda, para o problema da continuidade C^1 de placas, uma série de famílias de interpolantes de ordem 4 e 7 para serem aplicadas em triângulos. Utiliza adicionalmente um nó no meio de cada lado, paralelamente ao acréscimo de três variáveis de fixação (restrições).

Nessa mesma linha, com continuidade C^1, podem ser consultados ainda Lawson (1977), Nielson (1983), Poweel &Sabin (1977), Barnhill (1974), Farin (1983), Franke & Nielson (1983).

5.10.4. OUTROS ESQUEMAS DE INTERPOLAÇÃO NO TRIÂNGULO

Citamos apenas outros dois métodos, que se destacam por terem inspirado trabalhos posteriores.

McLain (1976) propos um esquema que obtém a triangulação de Delaunay e interpola, a seguir, em cada triângulo. O valor

da função em cada ponto interno P é obtido por uma média ponderada do valor de z de cada vértice:

$$z = w_1 \cdot z_1 + w_2 \cdot z_2 + w_3 \cdot z_3 .$$

Os pesos w_i são calculados levando em conta a distância perpendicular de P a cada lado e também as distâncias dos vértices ao pé dessas perpendiculares (são os valores a, b e c indicados na Figura 5.24, e seus análogos). Utiliza-se um esquema para o cálculo dos pesos, em função de uma variável n que é definida em função da continuidade desejada. Garante-se a continuidade da função e das (n-1) primeiras derivadas em todo o plano.

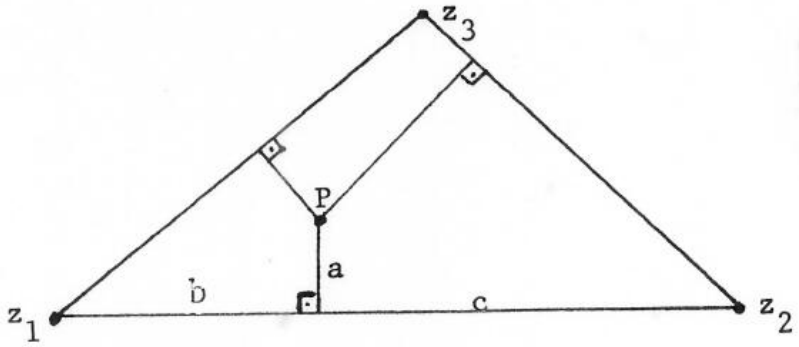


Figura 5.24 - Esquema geométrico das distâncias que são levadas em conta no cálculo dos pesos para o método de McLain.

Whitten & Koelling (1975) propõem um método para ajustar uma superfície em cada triângulo de forma a conseguir a conexão suave com suas vizinhas. São 15 coeficientes a serem determinados por 15 condições que asseguram essa conexão. O método nem sempre funciona bem pois há casos em que as condições não podem ser atendidas. E mesmo que atendidas, a função não é suave ao longo das mediatrizes do triângulo, como salien-

tou Akima.

5.10.5. CONCLUSÕES

Para a modelagem de superfícies, que é nosso caso, parece conveniente garantir a continuidade C^1, que ao mesmo tempo é suficiente para a maioria dos problemas. Sem essa continuidade o aspecto visual seria insuficiente; continuidades de ordem superior podem exigir muitos cálculos adicionais, na computação do polinômio e na determinação das derivadas.

Com relação à função interpolante a ser utilizada em cada triângulo, exige-se que não apresente descontinuidades internas nem externas (fronteira) ao triângulo. Podemos basear-nos no trabalho de Mansfield (1976) que demonstra que para obter um elemento com suporte nos triângulos vizinhos e com continuidade global C^1 é necessário utilizar um polinômio de grau 5, pelo menos.

Isso explica a utilização e/ou recomendação de um polinômio dessa ordem por diversos autores. Parece-nos portanto que a formulação de Akima é bastante satisfatória: um polinômio completo do quinto grau, bivariado em x e y e que utiliza somente os valores da função, das derivadas primeira e segunda e uma restrição quanto à variação da derivada normal ao lado. Assegura todo tipo de continuidade, isto é, em qualquer ponto e direção.

Aumentar o grau do polinômio (por exemplo 9 ou 13) também poderia levar a um polinômio que garante a continuidade, mas como já indicamos, parece-nos uma sofisticação desneces-

sária e que nem por isso conduz necessariamente a melhores resultados. Pelo contrário, aumenta as possibilidades de introduzir ondulações fictícias.

Por essas razões parece-nos bastante satisfatório o esquema de interpolação proposto por Akima e não vemos razões para alterá-lo.

5.11. COMPARAÇÃO DE TEMPOS TOTAIS

Uma vez terminadas as modificações que implicam tempo,pareceu conveniente incluir uma comparação de tempos totais.

Para isso somamos todos os tempos parciais referentes ao programa original e à versão modificada. Essa soma de parcelas (na realidade duas: TRIANG e INTERP) é necessária pois nosso sistema foi trabalhar com microcomputadores, nos quais o programa precisa ser subdividido.

O resultado dos tempos encontra-se na Tabela 5.9 e na Fi-

Tabela 5.9 - Tempos totais

	Tempos totais						
Pontos fornecidos	EX 30	EX 60	EX 90	EX 100	EX 200	EX 300	EX 400
Pontos na malha	(525)	(2.162)	(1.924)	(1.176)	(2.520)	(3.780)	(5.000)
1 original (4viz)	1m 30s	3m 40s	6m 12s	6m 17s	16m 07s	26m 25s	41m 06s
2 original (6viz)	1m 37s	3m 58s	6m 36s	6m 46s	17m 09s	28m 01s	43m 50s
3 modificado	1m 25s	3m 30s	5m 08s	5m 28s	13m 34s	21m 20s	32m 04s
diferença 1 - 3	5%	5%	17%	13%	16%	19%	22%
diferença 2 - 3	12%	12%	22%	19%	21%	24%	27%

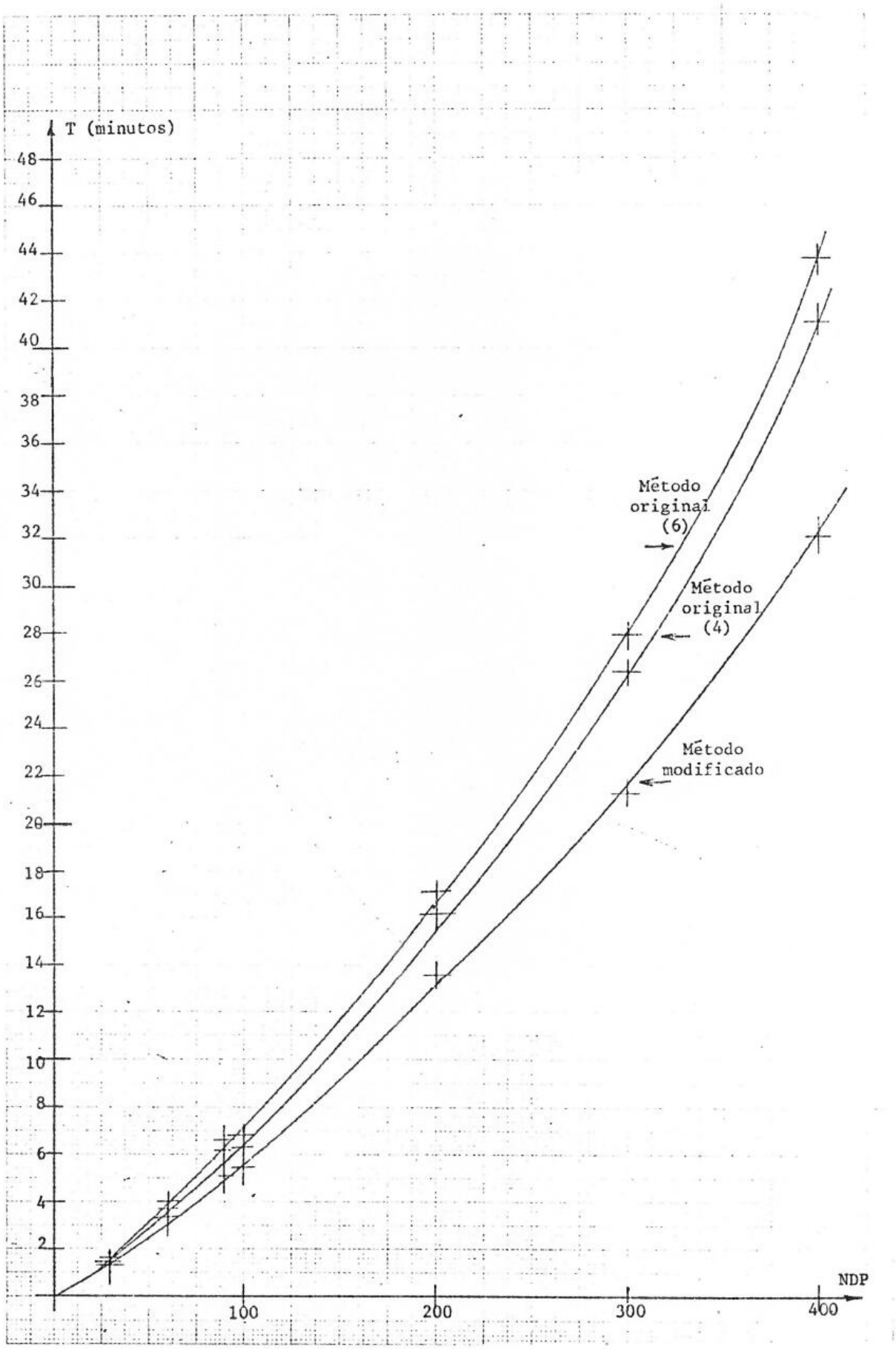


Figura 5.25 - Comparação de tempos totais

gura 5.25. Utilizando outros equipamentos os tempos serão certamente diferentes, no entanto pode-se supor que as relações de tempo permanecem.

Essa diminuição de tempo (cerca de 20% para conjuntos maiores) ganha um certo relevo quando se tem em conta que o método de Akima, de acordo com o estudo de Franke (1982), "é de longe o mais rápido de todos os métodos testados",(p.189).

Esse trabalho, Franke (1982), é a versão condensada de uma pesquisa que comparou 32 métodos tomados dentre os mais significativos dos EUA e desenvolvidos por renomados autores da área, como Akima, Nielson, Shepard, Lawson, Foley, McLain e o próprio Franke.

- ■ -

IV PARTE: COMPLEMENTOS AO PROGRAMA DE AKIMA E OUTRAS CONTRIBUIÇÕES

6. DESENHO DAS CURVAS DE ISOVALOR

6.1. INTRODUÇÃO

O método de Akima não está provido de uma rotina de desenho. Fornece uma malha regular, tão fina quanto se queira, com as coordenadas z dos nós. A partir daí o usuário deve adaptar a rotina de desenho de sua escolha.

Muitos sistemas utilizam a malha regular, por exemplo, o GPCP-II da Calcomp (1972) e o Surface Gridding Library da Dynamic Graphics (1978), mas não fornecem o programa fonte, nem maiores informações. Dentre os que facilitam informações técnicas e práticas, estudaremos comparativamente os de Cottafava & Le Moli (1969), o de Crane (1972) e o de Snyder (1978). Tendo em conta suas vantagens, implantamos em nosso systema este último método e por isso descrevemo-lo com maior detalhe.

Depois da comparação e da descrição do programa de Snyder apresentamos outras melhorias opcionais: a implantação de curvas que suavizam o resultado da malha quadrada; o refinamento do modelo através de uma malha triangular subdividida e o desenho das curvas de nível diretamente a partir do modelo triangular.

6.2. OPÇÕES NOS MÉTODOS QUE UTILIZAM MALHAS REGULARES

O problema consiste em desenhar linhas de igual nível através de uma malha, geralmente ortogonal, em que se conhece o valor da função em seus nós. A solução pode ser dividida em dois estágios. No primeiro calculam-se as coordenadas de todas as intersecções entre cada curva de nível e as arestas de cada célula. No segundo realiza-se a adequada união desses pontos.

A Figura 6.1 representa a passagem da curva de nível 50 através de uma rede onde estão indicadas as cotas de cada nó. O caminhamento através do pontos 1,2,...,8 é feito em seqüência.

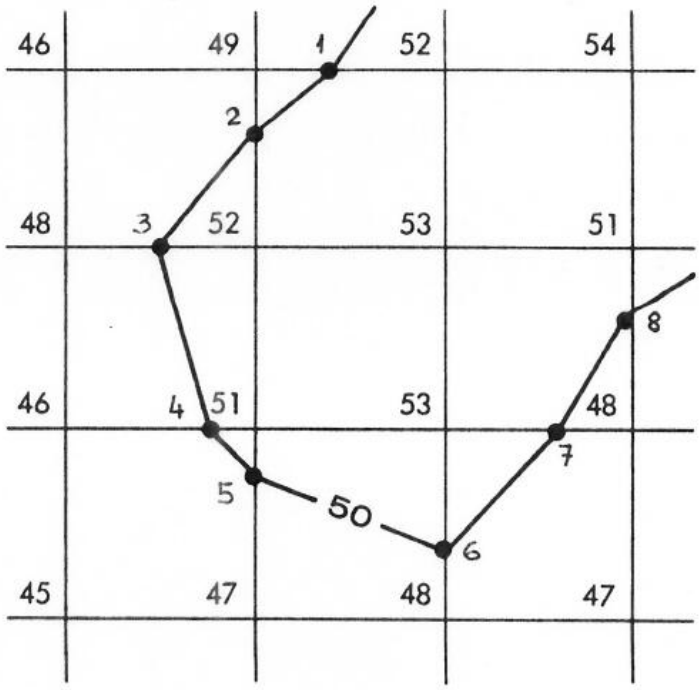


Figura 6.1 - Aspecto de uma curva de nível passando através de uma rede regular de pontos.

Com relação à solução do problema, existem duas classes

de métodos. A primeira opta por encontrar um ponto de uma curva de nível e segui-la até o fim, isto é, até que ela se feche sobre si mesma ou intercepte uma fronteira. O segundo método prefere examinar cada célula da malha e determinar todas as curvas de nível que interceptam essa célula.

As vantagens de seguir uma curva em seu caminhamento através da malha são duas: a rotulação das curvas (escrever o valor de sua cota) é relativamente fácil, e o movimento da pena (de um "ploter" incremental) é otimizado, isto é, ele não se move muito sem escrever nada. Tem a desvantagem de utilizar mais memória.

Por sua vez, o método que desenha todas as curvas encontradas dentro de uma célula tem a vantagem de necessitar menor capacidade de memória e de processar cada célula completamente antes de passar para a próxima, permitindo assim a geração de curvas sobre uma malha muito mais fina. Apresenta a desvantagem de tornar mais difíceis a união dos trechos de uma mesma curva e a rotulação das mesmas.

Para os microcomputadores, podemos evitar a dificuldade de memória do primeiro método através do processamento de uma curva de cada vez.

6.3. CASOS DE AMBIGÜIDADE EM UMA MALHA REGULAR

Na Figura 6.1, a curva de nível 50, ao entrar em uma célula só tem uma opção de saída. Mas isso nem sempre acontece; pode ocorrer um impasse quando não houver uma única saída possível. Essa configuração ocorre quando dois cantos opostos de

uma célula são mais altos do que a cota da curva de nível entrante; os outros dois cantos são mais baixos.

Teoricamente existem três possibilidades de solução e seus aspectos são os indicados na Figura 6.2.

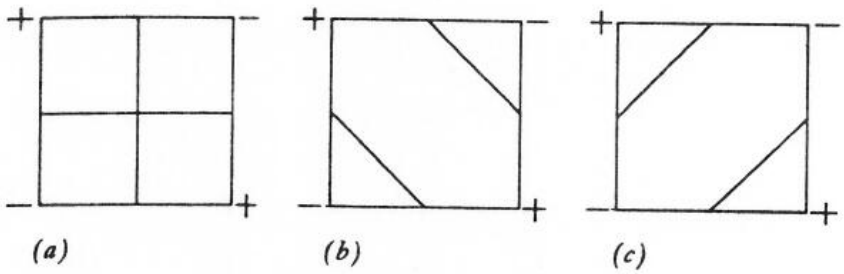


Figura 6.2 - Três possíveis aspectos de curvas de nível cruzando uma célula quadrada

Petinati (1983) estuda todos os casos possíveis, em função dos cantos concretos e dos valores de suas cotas. No entanto pode-se dizer que os três casos acima resumem a situação; as demais resolvem-se analogamente.

A possibilidade (a) é eliminada imediatamente tendo em conta que curvas de nível não se interceptam. No caso, salienta-se que se trata de dois tramos de uma mesma curva de nível.

A escolha deve então ser feita entre uma das duas alternativas remanescentes. O caso (b) corresponde a uma linha de cumiada e o caso (c) a uma linha de vale; coisas que podem ser verificadas através do exame de um perfil ao longo de uma diagonal, convenientemente escolhida.

As células triangulares não apresentam esse tipo de problema pois uma curva de nível ou cruza dois lados ou não cruza nenhum. Mas como em muitos casos é vantajoso utilizar uma

malha regular quadrada, é necessário encontrar uma solução para o problema da ambigüidade.

Por exemplo, o programa SURFACE II, que utiliza uma malha regular, recomenda e emprega o cálculo da cota do centro da malha, através da média simples das cotas dos quatro cantos. Se o valor no centro for maior do que o nível da curva a ser traçada, a solução é a configuração (b). Se o valor no centro for menor menor, opta-se pela configuração (c). Isso equivale na prática a dividir o quadrado em 4 triângulos com um vértice em comum (o centro da célula).

Existem outras maneiras de fugir da ambigüidade,como veremos ao comparar três métodos.

6.4. BREVE COMPARAÇÃO DE TRÊS MÉTODOS

Analisaremos brevemente os seguintes algoritmos, identificando-os pelos números:

1) - Cottafava & Le Moli (1969)
2) - Crane (1972)
3) - Snyder (1978).

Os três pertencem à classe dos que seguem uma curva até o fim em seu caminhamento através da malha. Utilizam como dado de entrada o valor da função nos nós dessa malha retangular.

A interpolação sobre a aresta para determinar o ponto de cruzamento da curva de nível é linear, isto é, todos os três utilizam uma fórmula do tipo

$$\Delta x = \frac{z - z_1}{z_2 - z_1} \cdot \ell$$

onde:

Δx - intersecção a partir do ponto 1;

z_2, z_1 - cota nos nós;

z - cota da curva de nível;

ℓ - espaçamento horizontal dos nós.

Os algoritmos (2) e (3) iniciam o processo logo que encontram uma borda de uma célula que é interceptada por uma curva de nível. As bordas interceptadas são identificadas por uma lista de inteiros em (2) e através de um mapa de bits em (3), o que poupa muita memória com relação a (2).

O algoritmo (1) realiza inicialmente a identificação da intersecção de todas as bordas com todas as curvas e só então passa à fase de união das intersecções. Os detalhes desse processo não são descritos.

Os algoritmos (1) e (2) iniciam a curva de nível na primeira borda encontrada em uma pesquisa que é feita ao longo de uma linha ou coluna da matriz de dados. O algoritmo (3) reduz o movimento da pena pesquisando ao longo de uma espiral que parte da posição atual da pena.

Nos três métodos desenham-se inicialmente as curvas que interceptam as bordas do desenho e só depois as curvas internas.

Outra comparação interessante refere-se ao modo como cada um resolve o caso de ambigüidade, isto é, quando existem 4

intersecções de uma mesma curva em uma célula (problema discutido no item anterior).

A solução dada pelo algoritmo (1) depende da ordem fixada para o exame das bordas. Se uma célula é examinada a partir de diferentes direções, diferentes curvas são traçadas.

O algoritmo (2) força as curvas a passarem pelo centro da célula, o que causa sérios problemas quando mais de uma curva intercepta as quatro arestas: as diferentes curvas se entrecruzam e o aspecto resultante parece um asterisco.

O algoritmo (3) examina o valor de Δx da interpolação nas arestas de cima e de baixo. Se o valor de Δx da borda superior for menor que o da inferior, então a curva de nível é desenhada do topo para a esquerda e do fundo para a direita (Figura 6.3,a). Caso contrário o desenho é feito do topo para a direita e do fundo para a esquerda (Figura 6.3,b).

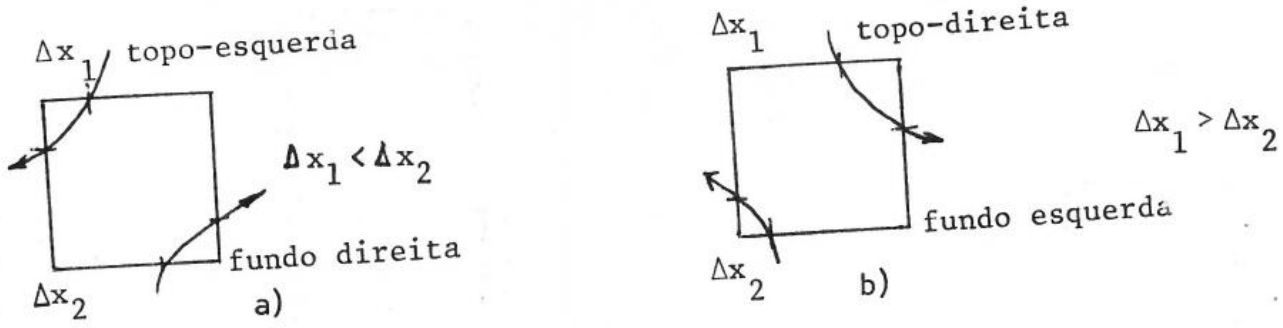


Figura 6.3 - Caminhamento das curvas de nível através da célula, em função do valor de Δx.

Este procedimento assegura que os mesmos trechos sejam escolhidos se os eixos x e y forem trocados, isto é, independe da ordem em que a célula é examinada.

Os algoritmos (1) e (2) consideram sempre todos os dados da tabela (matriz de cotas). O algoritmo (3) fornece um meio através do qual o usuário pode considerar separadamente só alguns elementos, correspondentes a uma região do terreno. Para isto basta definir adequadamente alguns elementos com cota superior ao valor de z_{max}. As arestas que unem qualquer desses elementos com outros não são examinadas. Essa característica é muito útil na prática pois geralmente os terrenos não possuem uma configuração retangular. Os algoritmos (1) e (2) apresentam curvas estranhas entre a região com dados e as que não contêm nenhum.

O método de saída de dados não é descrito em (1). O algoritmo (2) fornece como saída uma seqüência ordenada de pontos para cada curva a ser desenhada. O algoritmo (3) permite a opção entre uma seqüência ordenada de pontos, como (2), ou chamar uma rotina de desenho fornecida pelo usuário.

Saliente-se finalmente que nos algoritmos (2) e (3) as linhas da malha não precisam ser retas nem uniformemente espaçadas.

6.5. O ALGORITMO DE SNYDER (1978)

Recebe o nome de Contour Plotting (nº 531 da CACM) e foi elaborado por William V. Snyder, do 'Jet Propulsion Laboratory' (NASA).

É do tipo que segue uma curva de nível através da malha e fornece uma seqüência de pontos com os quais se pode desenhar as curvas através de uma rotina de desenho fornecida pe-

lo usuário.

Num problema representativo, gera 11.000 comandos para a mesa de desenho, contra aproximadamente 57.000 gerados por um programa utilizando o método celular (processar totalmente uma célula antes de passar a outra). Para esse problema,as linhas da malha, a identificação do usuário e a tabela de valores das curvas requerem cerca de 36.000 comandos adicionais.

O método justifica a interpolação linear, para determinar o ponto de intersecção de uma curva, ressaltando que o emprego de métodos não-lineares pode levar a uma intersecção múltipla de uma curva com uma aresta. Se a interpolação linear não produzir a suavidade desejada ou curvas precisas, o autor recomenda que se empregue uma malha mais refinada, calculando-se o valor da função nos novos nós ou empregando uma subtabulação interpoladora.

Se a memória necessária for maior que a disponível, pode-se dividir a área em parcelas a serem processadas separadamente.

Depois que as intersecções com as arestas das células foram encontradas deve-se dicidir como desenhar a curva dentro da célula, isto é, que curva utilizar para unir dois pontos. Snyder recomenda trechos de reta pois a utilização de outras curvas poderia causar balões ou cruzamentos indesejáveis.

6.5.1. DESCRIÇÃO DO PROGRAMA COMPUTACIONAL

De acordo com as sub-rotinas fornecidas montamos 2 esquemas de programa. O primeiro visando diretamente o desenho (de-

nominamos CNPLOT e seu esquema está representado na Figura 6.4

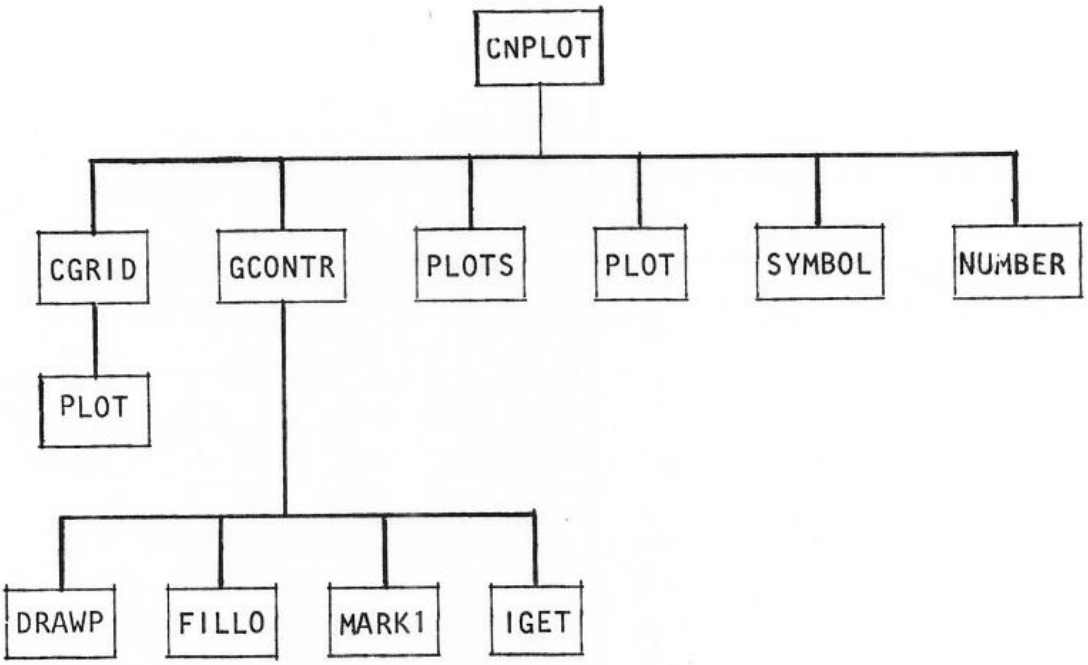


Figura 6.4 - Esquema do programa CNPLOT, para o desenho das curvas de nível e outros complementos.

e o segundo visando a impressão de seqüências de coordenadas de pontos (x,y), uma para cada curva (denominamos CNI, e seu esquema está representado na Figura 6.5).

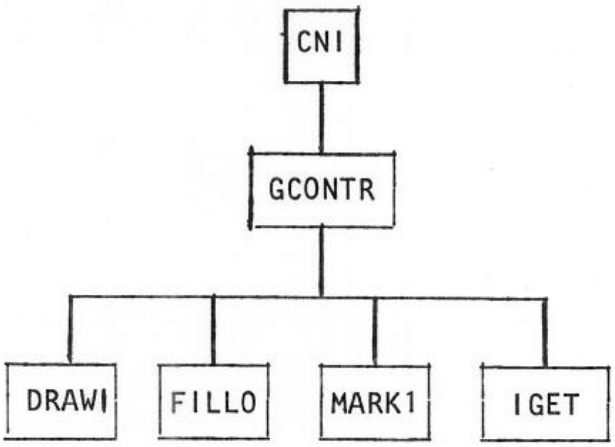


Figura 6.5 - Esquema do programa CNI, para a impressão de valores (x,y) das curvas de nível.

Uma breve descrição das sub-rotinas ajuda-nos a compreender o funcionamento dos dois esquemas.

As rotinas PLOT e PLOTS servem para desenhar retas,a rotina SYMBOL possibilita o desenho de símbolos especiais (asteriscos, triângulos, pequenos quadrados e outros para indicar convenções), a rotina NUMBER possibilita escrever números com a pena. Todas essas são rotinas que comandam os movimentos da pena de uma mesa de desenho.

A rotina CGRID também se refere a desenho: com o auxílio de PLOT serve para desenhar a quadrícula, isto é, uma malha de referência superposta ao desenho e cujo espaçamento padrão costuma ser de 5 ou 10 cm.

A rotina GCONTR é a peça fundamental do método, que descrevemos a seguir. Utiliza quatro outras sub-rotinas: FILLO, MARK1, IGET e DRAWP (ou DRAWI, no esquema para impressão somente).

A área de trabalho BITMAP (mapa de bits) serve para manter um registro de cada célula versus cada curva de nível. O estado 'zero' indica que aquela célula ainda não foi inspecionada para a curva de nível em questão. O estado 'um' indica que já foi examinada.

Como só existem duas opções (zero/um) isto possibilita trabalhar em nível de bits (e não em variáveis-palavra) economizando muita área de memória. O número de bits significativos por palavra depende do computador e deve ser levado em conta para passar para o nível de bits.

As três primeiras rotinas, FILLO, MARK1 e IGET trabalham em nível de bits. A primeira preenche o bitmap com zeros nas n primeiras posições, e é utilizada no início do processo. A segunda, MARK1 (BITMAP,N), coloca o número 'um' no enésimo bit do bitmap, para indicar que uma célula já foi examinada contra uma curva de nível. A terceira IGET (BITMAP,N) serve para verificar se uma aresta qualquer já foi examinada ou não. Para isso faz com que a variável I = IGET seja igual ao valor do enésimo bit do bitmap (zero/um).

A sub-rotina DRAWP (X,Y,IFLAG) a ser fornecida pelo usuário, serve para desenhar as curvas, por trechos de reta. Deve ser compatível com os equipamentos de desenho. X e Y correspondem às coordenadas dos pontos. IFLAG deve ter um dos seguintes valores convencionais:

1. continue o desenho de uma curva;
2. inicie uma curva, por um ponto da fronteira (limite externo do desenho);
3. inicie uma curva por um ponto interno ao desenho;
4. termine uma curva na fronteira;
5. termine uma curva fechada (não na fronteira).

Note-se que as solicitações 1, 4 e 5 são para movimentos com a pena abaixada (desenhando) e as 2 e 3 para movimentos com a pena livre.

6. apronte x e y para demarcar a posição da pena para a próxima curva. Caso seja omitida essa definição, a posição inicial fica sendo a correspondente a (1,1).

IFLAG/10- é o número da curva de nível.

A sub-rotina DRAWI é a versão de DRAWP para o segundo esquema (Figura 6.5). Em vez de desenhar, imprime uma seqüência de pontos, para cada curva de nível. Ao mesmo tempo escreve as mensagens equivalentes aos comandos 2 a 5 acima.

GCONTR, a rotina principal possui os seguintes argumentos:

-Z: é a matriz dos valores da variável, para as quais as curvas de nível devem ser desenhadas. Assume-se que os elementos de Z representam o valor da função nos nós de um sistema de coordenadas topologicamente retangular (por exemplo, cartesiano, polar com exceção da origem, etc.);

-NZ: é o número de linhas declarado para Z no programa de chamada;

-NX: é o limite para o primeiro subscrito de z (x_{max});

-NY: idem, para o segundo subscrito (y_{max});

-CV: são os valores das curvas de nível a serem desenhadas;

-NCV: é o número de curvas de nível a serem desenhadas;

-Z_{max}: é o máximo valor de z a ser considerado. Um valor de z(i,j) maior que z_{max} é sinal de que, esse ponto e os segmentos da malha que partem dele para seus vizinhos, devem ser excluídos do desenho.

6.5.2. DETALHES E MODIFICAÇÕES PARA IMPLANTAÇÕES EM MICROS

Implantamos o algoritmo de Snyder no micro 201 DP da Po-

lymax realizando os necessários ajustes e adaptações. Para expandir sua capacidade introduzimos modificações no sentido de processar uma curva de cada vez. Com isso o tamanho do bitmap ficou dividido pelo número de curvas de nível existentes, que num desenho pode ser da ordem de 10, 20 ou mais.

Para essa mesma finalidade, poupar memória, utilizamos o mesmo esquema da Figura 6.5, gravando em disco as seqüências de coordenadas, depois de processada cada curva.

Outra modificação para poupar memória, que implantamos, foi passar a cota z para inteiro (duplica a capacidade de memória). Supondo, por exemplo, que z seja fornecido com duas casas decimais, aplicamos inicialmente uma translação de coordenadas subtraindo $z_{ref} = z_{min} - 0,01$ de todos os valores da matriz e multiplicando a seguir os novos valores por um fator multiplicativo FM = 100. Com isso não se perde precisão, consegue-se uma boa faixa de amplitude para z e, tomando o cuidado de realizar as mesmas operações com o valor das curvas, em nada se altera a posição dos pontos de interpolação.

Com essas alterações para poupar memória foi possível implantar uma malha de 10.201 posições, o que equivale a uma malha quadrada de abertura 1 cm em papel A_0 (841 × 1.189 mm):

$$85 \times 120 = 10.200 \text{ posições.}$$

Caso o desenho não ocupe toda a área correspondente ao papel A_0 (isto é, seja menor), a malha pode ser mais refinada. Por outro lado, se se deseja uma malha ainda mais fina, pode-se executar o desenho em duas ou mais partes.

Para realizar a conexão do programa de Akima com este de

desenho foi necessário realizar previamente a ordenação de z, que no programa de Akima só é fornecido triângulo por triângulo e portanto, desordenadamente. Esse mesmo programa já converte z, de variável real para inteira.

6.6. CONTRIBUIÇÕES PARA TRAÇADO DAS CURVAS DE NÍVEL

6.6.1. ADAPTAÇÃO DE UMA CURVA SUAVE PASSANDO POR UMA SEQÜÊNCIA DE PONTOS

O desenvolvimento do algoritmo de Snyder e outros semelhantes conduz a uma seqüência de pontos para cada ramo de uma curva de nível. Esses pontos são unidos por trechos de reta.

Se a malha não for suficientemente fina ou as curvas de nível tiverem trechos com curvatura acentuada, podem aparecer bicos, de aparência visual indesejável.

O refinamento da malha seria uma solução mas até um certo limite pois pode tornar crítico o problema de memória ou o de tempo de computação.

A solução nesses casos pode ser adaptar uma curva suave passando através da seqüência de pontos fornecidos. Com uma equação algébrica calculam-se pontos adicionais a espaçamentos menores. A união desses pontos por trechos de reta já não dá a sensação de arestas vivas e a curva torna-se visualmente suave.

Para uma região com alta densidade de curvas,deve-se to-

mar os cuidados para evitar que as curvas interpoladoras se cruzem.

Na literatura existem diversos métodos e algoritmos para interpolar uma curva suave. Alguns deles, por exemplo, o 433 da ACM de Akima (1972); o de Ellis e McLain (1977); o 'Adapt' ou 525 da ACM de Rice (1978); o 547 da ACM de Duris (1980); o de Chung (1980); necessitariam adaptações, pois aplicam-se a curvas univariadas, isto é, que só admitem um valor de y para um mesmo x. Como as curvas de nível não são univariadas, seria necessário processá-las por trecho. Por isso é preferível recorrer a outros métodos.

Um processo simples é o utilizado pela IBM. Trata-se da composição de curvas circulares, que tentam imitar o procedimento do desenhista ao traçar curvas girando o lápis em torno de um ponto relativamente fixo: o ponto de contacto da mão com a mesa de desenho.

A Figura 6.6 descreve o processo. São tomados quatro pon-

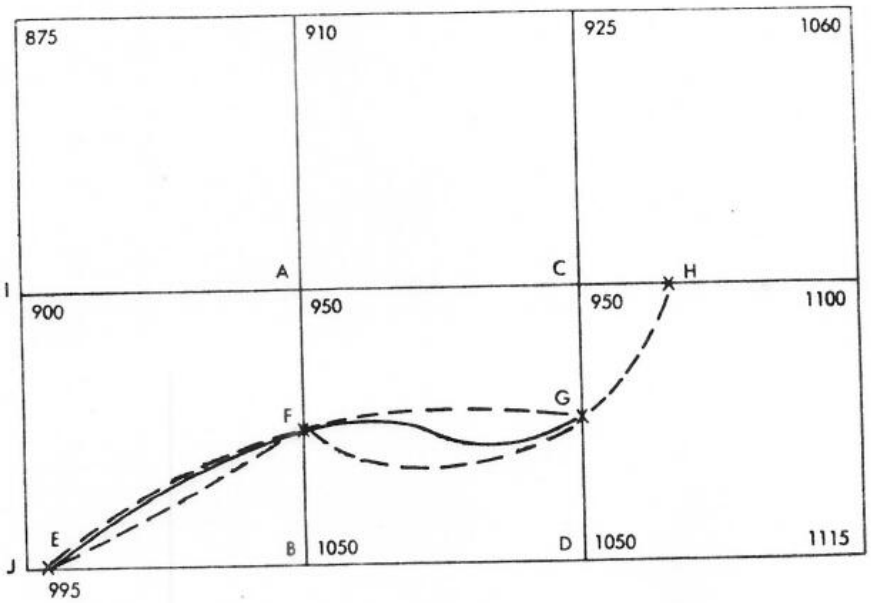


Figura 6.6 - Suavização de curvas através da composição ponderada de arcos circulares - técnica da IBM.

tos E, F, G e H de mesma cota inteira, determinados por interpolação nas arestas das células.

Inicialmente calculam-se as equações dos arcos de circunferência que passam pelo pontos EFG (um arco) e FGH (outro arco).

Trata-se de calcular a curva no trecho FG e isso é feito através da média ponderada entre um e outro arco. O arco da esquerda é ponderado com uma função linear que vale 1 no ponto F e zero no ponto G; o arco da direita é ponderado de maneira análoga e complementar; vale zero em F e um em G. A curva pode então ser desenhada calculando-se n pontos suplementares, intermediários entre F e G. O processo repete-se para outra seqüência de quatro pontos FGHI.

Outros métodos escolhem outros tipos de curvas,por exemplo uma parábola que passa por quatro pontos e que ajusta o intervalo para os dois pontos centrais; é o caso de McConalogue (1971).

Estudamos a seguir o método de Cline (1974), que oferece diversas opções, perfeitamente adaptáveis ao caso das curvas de nível e que implantamos efetivamente em nosso programa.

6.6.2. O MÉTODO DE CLINE (1974)

São seis subprogramas para o ajuste de curvas que utilizam splines sob tensão. Encontra-se disponível na coleção de algoritmos da ACM sob o número 476.

O primeiro par CURV1 e CURV2 resolve o problema padrão de

interpolação: determina uma função real que assume valores y_i, em pontos x_i (com $x_{i+1} > x_i$) ou seja, uma função univariada e que portanto não se aplica diretamente ao nosso problema. Pode ser útil, como todos os demais citados anteriormente, para o desenho suave de perfis.

O segundo par KURV1 e KURV2 resolve, em conjunto, o problema mais geral de passar uma curva suave através de uma seqüência de pontos no plano e serve perfeitamente para as curvas de nível.

O terceiro par KURVP1 e KURVP2, também em conjunto, resolve o mesmo problema, mas para curvas fechadas e portanto aplica-se notoriamente para curvas de nível interiores à região.

KURV1 requer o fornecimento da inclinação (tangente à curva) nos pontos inicial e final da curva. O usuário pode omitir essa informação (foi o que fizemos, eliminando também a opção) e então esses valores são produzidos internamente baseando-se nos pontos vizinhos. Se são fornecidos três ou mais pontos, como é o caso mais comum, esses valores, no ponto inicial e final, são calculados pela inclinação de um polinômio quadrático ajustado aos 3 primeiros (3 últimos) valores.Se só são fornecidos dois pontos, o resultado é uma linha reta.

KURVP1, sendo fechada, sempre tem vizinhos à direita e à esquerda e não necessita o fornecimento adicional de inclinações.

Em cada par de programas o primeiro só é chamado uma vez e determina os parâmetros da spline. O segundo realiza o ma-

peamento (adensamento) de pontos.

Durante o processo, cada curva é transformada em sua paramétrica, isto é, seu comprimento poligonal é definido como sendo 'um'. Para definir um ponto sobre a curva deve-se fornecer o parâmetro T, que será contado a partir da origem, com amplitude de zero a um.

Para calcular uma seqüência de M pontos deve-se chamar M vezes a função KURV2 (ou KUVP2). Em cada chamada o valor de T deve ser crescente e de preferência os incrementos devem ser constantes.

Na adaptação para microcomputadores introduzimos as simplificações necessárias para automatizar o processo: calcular diversos pontos em seqüência sobre uma mesma curva com uma única chamada; o número de pontos a serem calculados depende do número de pontos fornecidos, havendo entre eles uma relação constante que denominamos razão de compressão. Adotamos o valor 4 para esse parâmetro, que pode ser modificado pelo usuário.

É interessante e merece destaque nesse programa a variável SIGMA que define o fator de tensão da spline e serve como indicativo da curvatura. Se o valor de SIGMA é muito grande (por exemplo 50) a curva resultante é muito próxima da linha poligonal que passa pelos pontos dados. Se o valor de SIGMA for muito pequeno (por exemplo 0,1) a curva fica bastante aberta e 'bojuda'. O valor padrão para SIGMA recomendado pelo autor é 1,0 (um).

Podendo controlar o parâmetro SIGMA, conseguimos ajustar

bem as curvas. Se em algum caso concreto houver muitas curvas em uma região e existir o problema de superposição,pode-se utilizar um valor mais elevado para SIGMA.

Realizamos diversos testes para determinar o melhor valor de SIGMA. Tomamos para isso quatro curvas, duas abertas e duas fechadas, com significativas variações de curvatura.Testamos diversos valores para esse parâmetro, desde 0,1 a 10,0 e, como resultado, adotamos 1,5 para curvas abertas e 2,5 para curvas fechadas.

O valor 5,0 já produz uma curva muito próxima, quase coincidente, com a poligonal fornecida e portanto não introduz melhorias na suavidade. O valor 1,0 e os inferiores, produzem curvas com muitas ondulações e pontos de inflexão, de aparência visual pouco aceitável.

6.6.3. MALHA TRIANGULAR REFINADA

O programa de Akima fornece uma malha retangular cujas

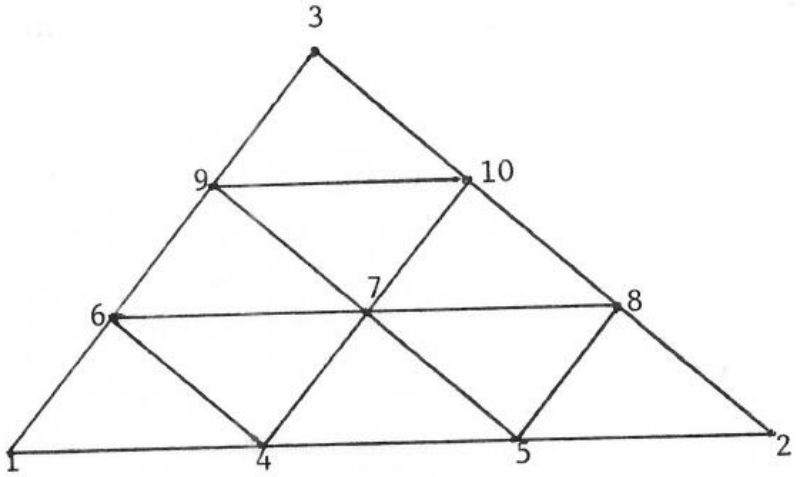


Figura 6.7 - Esquema de uma subdivisão de triângulos para a formação de uma rede mais densa.

cotas são calculadas nos nós, a partir de uma interpolação no triângulo.

Outra opção para adensar os pontos é montar uma malha triangular mais fina. Para isso subdividem-se os triângulos originais e calculam-se as cotas z nos novos nós formados (Figura 6.7).

Costuma-se trabalhar com a representação isoparamétrica do triângulo (coordenadas baricêntricas), o que facilita o cálculo das coordenadas nos novos nós. Uma solução desse tipo conectaria bem com o programa de Akima que calcula o polinômio e as interpolações baseado nessa representação. Uma sugestão de divisão de triângulos pode ser encontrada em Meek & Beer (1976), ou em Rivara (1984).

A interpolação ao longo de cada aresta de triângulo pode obedecer a um esquema linear. A união dos pontos de intersecção de cada curva de nível com as arestas pode ser feito por linhas retas, ou pela adoção de uma curva suave.

Um esquema desse tipo seria análogo ao de Snyder para a malha quadrada. A vantagem da malha triangular, como apontamos anteriormente, é que evita casos ambíguos de intersecção múltipla.

6.6.4. DESENHO DAS CURVAS DE NÍVEL BASEADO NO POLINÔMIO INTERPOLADOR

A utilização de uma malha de refinamento (retangular ou triangular) supõe um passo intermediário que pode ser evitado

considerando que dispomos em cada triângulo, do polinômio que representa a superfície. Uma curva de nível consiste na intersecção desse polinômio com o plano horizontal de cota igual ao valor da curva desejada.

Teoricamente seria possível desenhar as curvas de nível encontrando os zeros da função interpoladora,diminuída do valor de z da curva de nível. No entanto, a determinação de cada ponto (raiz do polinômio) necessita de uma série de cálculos iterativos. Sendo z(x,y) ou z(u,v) uma função bivariada do 5º grau, não é possível o cálculo direto. Arbitrando um valor para u, deve-se encontrar o valor de v que anula a função. O número de iterações dependerá da precisão desejada, que costuma ser algo em torno de 0,1 mm para ser compatível com o máximo erro permitido no desenho (erro de graficismo; distinção a olho nú), tornando imperceptíveis quaisquer descontinuidades.

Tendo em conta, além disso, que o espaçamento entre os pontos de uma curva de nível deve ser da ordem de uns 2 a 5 mm para se ter uma continuidade razoável, podemos ter uma quantidade de pontos da ordem do milhar em cada curva de nível. Desses dois fatores; número de iterações para cada ponto e número de pontos por curva de nível; podemos concluir que em muitos casos o processo é inviável do ponto de vista do tempo de computação.

Outra opção,sugerida por Schagen (1982) permite diminuir as iterações. Consiste em obter um ponto de partida e calcular a seqüência de pontos de forma que cada ponto se obtém a

partir do anterior.

De acordo com o esquema da Figura 6.8, calcula-se a tangente à curva no ponto atual, dá-se um incremento d_r sobre essa linha e obtém-se a perpendicular s que é a linha de procura do novo ponto.

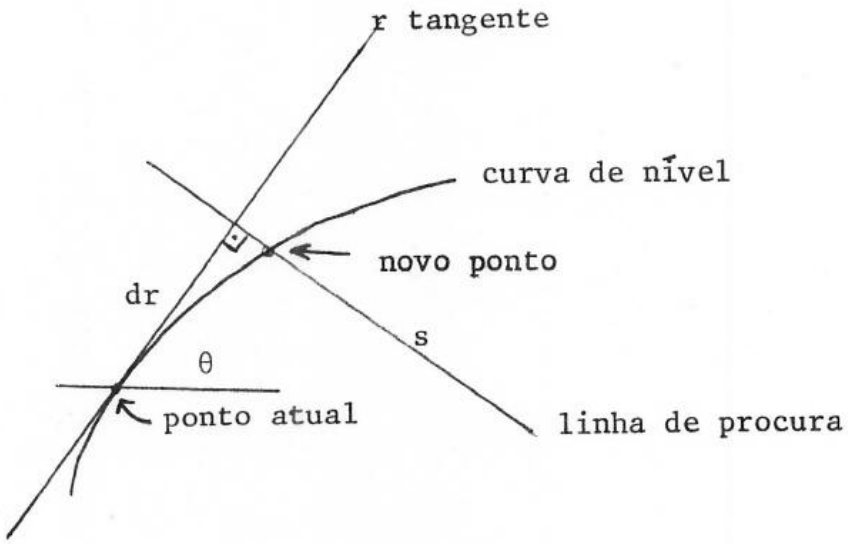


Figura 6.8 - Esquema de busca de um novo ponto, a partir de um ponto inicial: acréscimo sobre a tangente.

Em todo o caso, o número de iterações continua sendo grande: como os erros são cumulativos (cada ponto é calculado a partir do anterior) impõe-se uma boa precisão, o que leva a aumentar o número de iterações.

Outro método, que não necessita de processos iterativos e que fornece resultados comprovadamente bons, é o de Akin & Gray (1979). Realizamos a seguir uma descrição desse método, adaptando-o ao problema do triângulo, com um polinômio bivariado do 5º grau.

Consideremos o triângulo isoparamétrico formulado em ter-

mos de coordenadas isoparamétricas (u,v), contrapondo-se às globais (x,y).

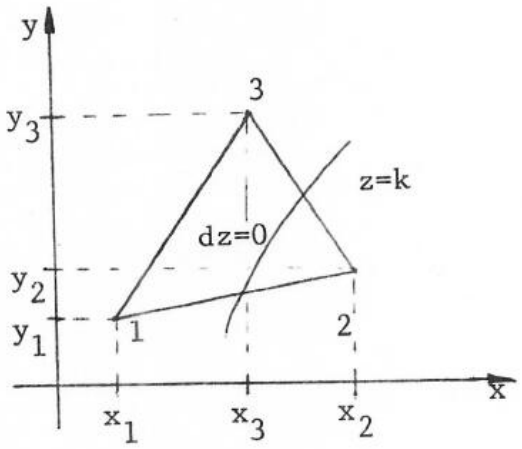


Figura 6.9 - Coordenadas globais do triângulo.

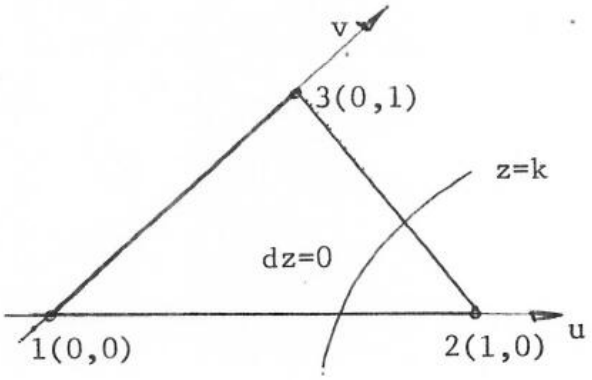


Figura 6.10 - Coordenadas isoparamétricas do triângulo.

Suponhamos que se quer desenhar uma curva de nível com valor k (constante) e que se conhece um ponto dessa curva, com coordenadas u_0 e v_0. Caso não conheçamos esse valor inicial podemos determiná-lo pela equação de z ao longo de um lado, por exemplo o lado u:

$$z(u,v)_{v=0} = P_{00} + P_{10}\cdot u + P_{20}\cdot u^2 + P_{30}\cdot u^3 + P_{40}\cdot u^4 + P_{50}\cdot u^5$$

Iterativamente pode-se determinar, com uma precisão arbitrária ε, o valor da coordenada u que anula $z(u,v) - k$.

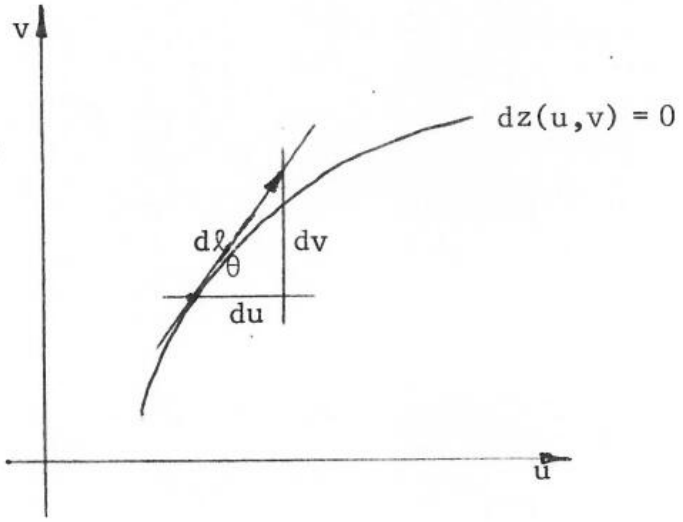
A equação da curva sobre a superfície é $dz = 0$. Em termos de coordenadas locais pode-se escrever:

$$0 = dz = \frac{\partial z}{\partial u}\, du + \frac{\partial z}{\partial v}\, dv,$$

ou

$$z_u \cdot du + z_v \cdot dv = 0, \qquad (1)$$

onde du e dv são componentes do deslocamento $d\ell$ que é tangente à curva. Podemos assumir que $d\ell$, o passo de caminhamento para desenho, seja um valor pré-fixado e constante. Assim sendo, valem as relações indicadas na Figura 6.11.



$dv = d\ell \cdot \operatorname{sen}\theta$

$du = d\ell \cdot \cos\theta$

$\operatorname{tg}\theta = \dfrac{dv}{du}$

$d\ell = \sqrt{du^2 + dv^2}$

Figura 6.11 - Relações entre as diferenciais dos deslocamentos.

O valor de $\operatorname{tg}\theta$ pode ser obtido pela equação (1) utilizando posteriormente os valores de z_u e z_v no ponto em que se de-

seja calcular essa tangente.

De (1) obtém-se

$$z_u + z_v \cdot \frac{d_v}{d_u} = 0$$

e então

$$\frac{d_v}{d_u} = \frac{-z_u}{z_v}$$

e finalmente $tg\theta = -z_u/z_v$. Os valores (equações) de z_u e z_v obtém-se derivando a equação z do polinômio interpolador.

Então, dado um ponto de partida de uma curva, pode-se desenhá-la utilizando a seguinte equação de recorrência:

$$u_n = u_{n-1} + \frac{d\ell}{\nabla} \cdot z_{v(n-1)}$$

$$v_n = v_{n-1} - \frac{d\ell}{\nabla} \cdot z_{u(n-1)}$$

onde:

u_n, v_n - coordenadas do novo ponto;

u_{n-1}, v_{n-1} - coordenadas do ponto anterior;

$d\ell$ - passo incremental constante (por exemplo, 1 mm);

$z_{v(n-1)}$ - derivada da função, com relação a v, no ponto anterior;

$z_{u(n-1)}$ - derivada da função, com relação a u, no ponto anterior;

$\nabla = \sqrt{z^2_{u(n-1)} + z^2_{v(n-1)}}$ = gradiente no ponto anterior

A precisão e o tempo de processamento (custo) dependem da

dimensão escolhida para o incremento $d\ell$.

Este processo pode desviar-se um pouco do caminhamento correto da curva de nível, por causa dos erros numéricos envolvidos, ou seja, depois de cada passo há um pequeno erro que se acumula aos anteriores. Na prática esse erro cumulativo não é perceptível visualmente, excetuando-se o caso, difícil de ocorrer, de uma curva fechada dentro de um triângulo. Por causa das imprecisões, tal curva pode não fechar e seguir uma espiral até que encontre um elemento de fronteira ou se atinja o número máximo de pontos fixado para uma curva. Esse último caso é o que vem desenhado na Figura 6.12.

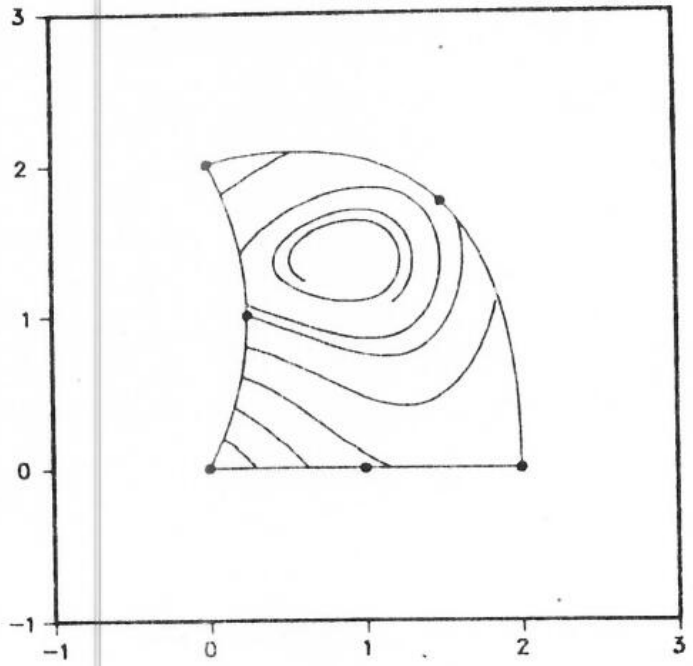


Figura 6.12 - Curvas de nível de um elemento triangular. Notar a curva espiral no centro que deveria ser fechada sobre si mesma.

No entanto, o procedimento acima foi modificado para tornar os erros cumulativos aceitavelmente pequenos, sem exces-

sivo acréscimo de tempo computacional.

As novas equações são as seguintes:

$$u_{cor} = u_n - \left(\frac{1}{\nabla}\right)^2 \cdot z_{u(n-1)} \cdot (z(u_n, v_n) - k)$$

$$v_{cor} = v_n - \left(\frac{1}{\nabla}\right)^2 \cdot z_{v(n-1)} \cdot (z(u_n, v_n) - k)$$

onde:

u_{cor}, v_{cor} - valores do novo ponto, corrigidos;

u_n, v_n - valores obtidos pela fórmula anterior;

k - valor da curva de nível;

os demais elementos obedecem à mesma simbologia definida anteriormente.

Estas correções produzem linhas que são satisfatórias e mais precisas, na maior parte dos casos, do que é possível obter com os sistemas gráficos.

Para terminar uma curva, deve-se verificar as coordenadas do ponto final de cada segmento, examinando se esse ponto ainda está dentro do triângulo.

Uma vez que uma curva foi completamente traçada deve-se converter as coordenadas locais (u,v) para o sistema global (x,y) a fim de realizar o desenho.

O esquema para o desenho das curvas através da malha triangular pode ser pensado em termos de 2 caminhos diferentes: rastreamento de uma curva ou mapeamento triângulo a triângulo

No primeiro caso segue-se uma curva através de toda a malha. Dado um ponto inicial calculamos a seqüência de pontos

no triângulo até chegar a uma outra aresta. Através dessa aresta passa-se ao triângulo contíguo, que pode ser determinado na lista IPT do programa de Akima. Processado esse novo triângulo passa-se ao seguinte, e assim por diante até chegar a uma aresta de borda (lista IPL no programa de Akima). Depois pesquisam-se outros possíveis ramos da mesma curva de nível examinando todos os outros triângulos. A seguir inicia-se novamente o processo com outra curva de nível.

Na segunda alternativa examina-se um triângulo de cada vez, processando todas as curvas existentes em seu interior. Com isto evita-se processar mais de uma vez um mesmo triângulo; evita-se o armazenamento de arestas de borda e a procura de triângulos adjacentes. Por outro lado, a rotulação das curvas torna-se mais difícil.

A comunicação de Akin & Gray (1979), só esquematiza o caso de um triângulo isolado, sendo necessário, para implantá-lo, resolver o problema da malha globalmente.

Essa implantação seria útil, poupando a rotina IDGRID e absorvendo IDPTIP. Dispensa-se também o programa anterior para o desenho de curvas de nível.

Esse método parece-nos satisfatório, tendo em conta a qualidade do desenho apresentado. Veja-se a seguir o mesmo desenho da figura anterior (6.12), com as correções (Figura 6.13).

No entanto, não se deve descartar a malha quadrada, pois representa um bom instrumento de trabalho ao permitir o armazenamento de um modelo digital do terreno, muito próprio para

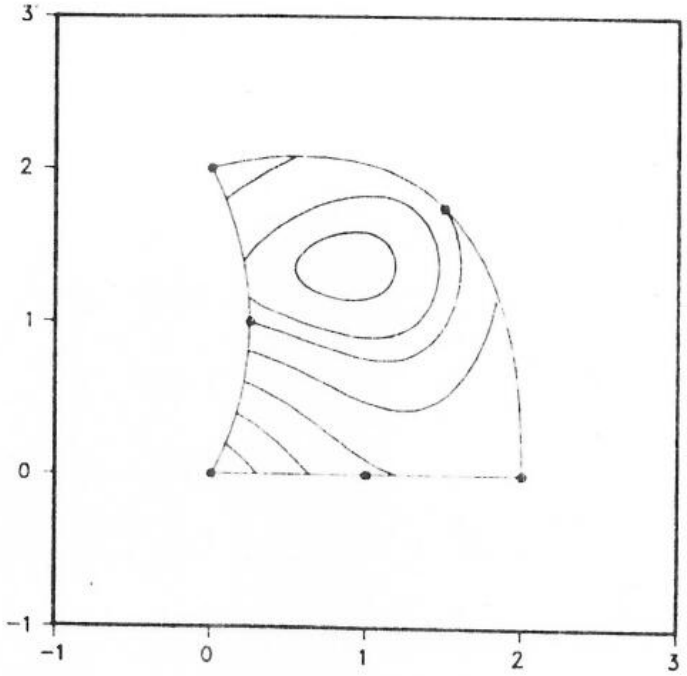


Figura 6.13 - Curvas de nível de um elemento triangular. Notar que, contrariamente ao exemplo anterior, a curva interior fecha exatamente sobre si mesma.

cálculos em computadores.

- ■ -

7 - ADAPTAÇÃO DO SISTEMA PARA UTILIZAÇÃO EM MICROCOMPUTADORES

7.1. INTRODUÇÃO

Tendo em conta as vantagens que essa solução oferece, procuramos adaptar o programa de Akima e os programas de desenho para sua utilização em microcomputadores.

Para isso, as principais modificações referem-se à otimização do espaço reservado à memória, recurso que é um fator limitante para os micros. No equipamento utilizado, um Polymax 201 DP, dispõe-se de 64 k bytes de memória.

7.2. CAPACIDADE DE MEMÓRIA

Pensando em poupar memória, introduzimos algumas modificações. Em primeiro lugar, como visamos fundamentalmente o desenho de curvas de isovalor, torna-se dispensável o cálculo de cotas em pontos isolados e por isso não é necessário o programa IDBVIP e a sub-rotina IDLCTN.

Também são perfeitamente dispensáveis as diversas opções geradas pelos diferentes valores de MD (1, 2 ou 3), pois para o desenho de um mapa só utilizamos um conjunto de pontos fornecidos e uma só malha de saída.

Da mesma forma, a diversidade de opções provocada pela

escolha do número de vizinhos fica automaticamente eliminada pela adoção dos vizinhos de Voronoi, que são determinados pela configuração geométrica do problema.

Eliminamos também os passos relativos à extrapolação de valores para pontos fora do polígono convexo. Essa política concorda com a visão teórica, já que não se pode prever o comportamento da função além do limite dos pontos amostrados, e também com a prática, já que se desejamos mapear determinada área devemos levantar pontos em toda a sua extensão, para não precisar recorrer a extrapolação.

Com isso conseguimos reduções de memória nas sub-rotinas IDGRID e IDPTIP.

7.3. FRACIONAMENTO DO PROGRAMA E CAPACIDADE MÁXIMA

Em sua versão original, o programa tem pouca capacidade quando utilizado em micros, devido à dimensão da área utilizada para o programa em si.

Para aumentar a capacidade, estudamos a alocação interna de memória das sub-rotinas, para verificar as possíveis partições em programas menores. Esse estudo está resumido na Figura 7.1, que procuramos explicar a seguir.

Para tornar o estudo mais concreto e aumentar a sensibilidade para as dimensões envolvidas, utilizamos um exemplo ilustrativo com 100 pontos fornecidos e uma malha de saída de 101 por 101 pontos.

Todas as diversas áreas de variável inteira estão grava-

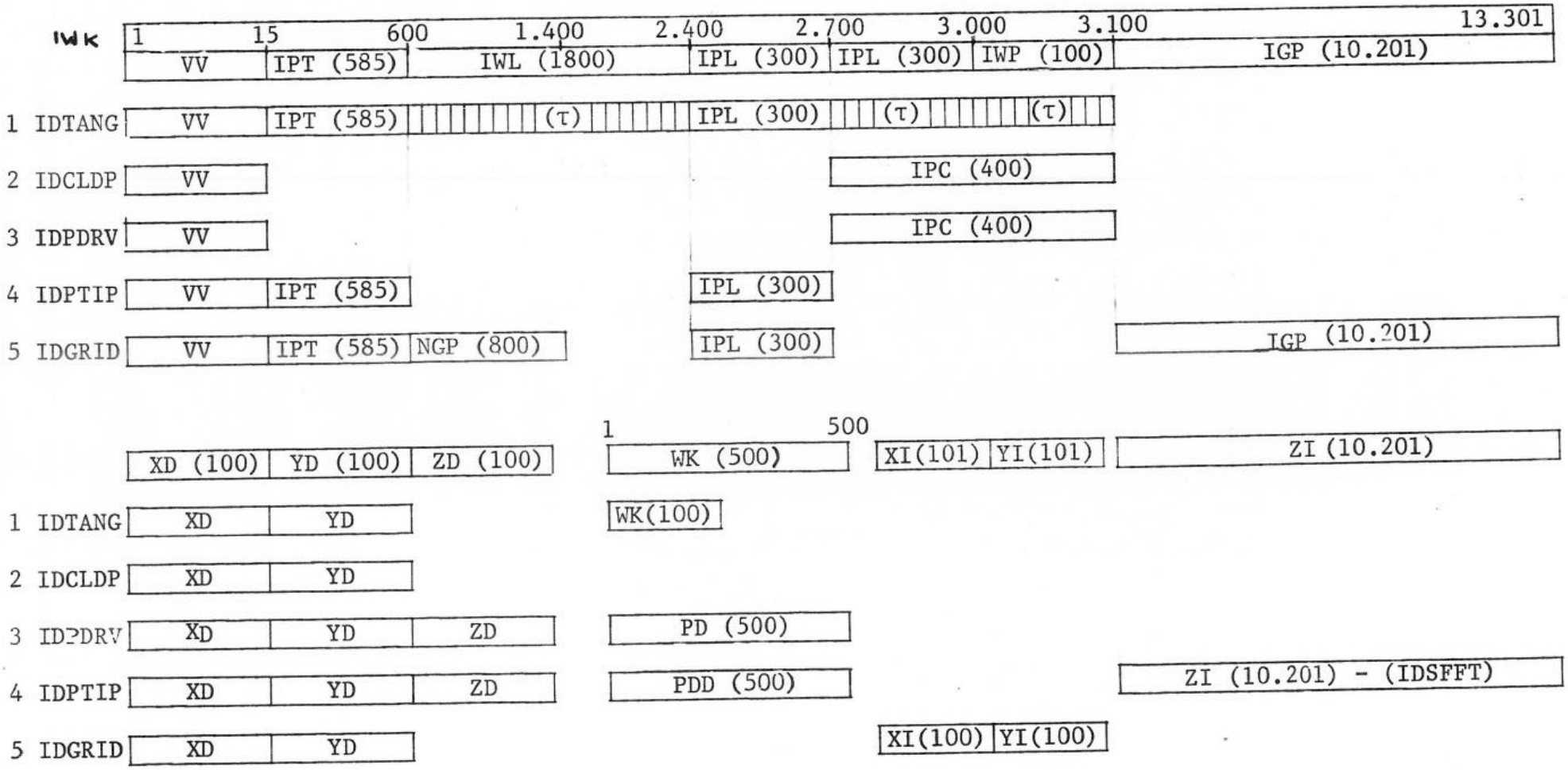


Figura 7.1 - Esquema de memórias do programa original - 100 pontos, 4 vizinhos, malha (101×101)

das no vetor IWK (primeira linha da Figura 7.1), que ocupa 13.301 posições, divididas em duas grandes áreas: as primeiras 3.100 posições (31*NDP), correspondente a diversos arquivos e 10.201 (101*101) pertencentes ao arquivo IGP, que corresponde aos nós da malha regular.

Nas linhas subsequentes, à esquerda, indicam-se as diversas sub-rotinas indicando por barras os correspondentes arquivos de IWK que cada uma utiliza.

Assim, as primeiras 15 posições são utilizadas para o armazenamento de diversas variáveis isoladas: NDP (número de pontos), NCP (número de vizinhos), NT (número de triângulos), NL (número de arestas de borda), etc. Todas as sub-rotinas utilizam essa região.

O arquivo IPT; com 585 posições (6*NDP-15) é gerado por IDTANG e utilizado por IDPTIP e IDGRID. Armazena os vértices dos triângulos.

O arquivo IWL, com 1.800 posições (18*NDP) é utilizado por IDTANG como área interna de trabalho. As primeiras 800 posições (8*NDP) são utilizadas posteriormente para o arquivo NGP, onde são gravados o número de nós associados a cada triângulo, borda ou vértice.

O arquivo IPL, com 600 posições (6*NDP) é utilizado por IDTANG para armazenar os vértices das arestas de borda e o triângulo a que cada uma pertence. No fim do processo, todos os valores que interessam são deslocados para as primeiras 300 posições (3*NDP), que são chamadas pelas sub-rotinas IDPTIP e

IDGRID. As 300 posições finais serão aproveitadas por IDCLDP e IDPDRV.

A área IWP, com 100 posições (NDP), é utilizada por IDTANG como área interna de trabalho. Junto com a parte final de IPL, 300 posições (3*NDP) serve para o arquivo IPC, com 400 posições (4*NDP), onde se armazenam os vizinhos de cada ponto. Isso supondo 4 vizinhos por vértice. Esse arquivo, gerado por IDCLDP é utilizado por IDPDRV.

Finalmente temos a maior área de todas, IGP, com 10.201 posições (101×101) e que depende somente das dimensões da malha de saída. É utilizada por IDGRID para armazenar o número dos nós da malha por ordem de pertinência aos triângulos, arestas de borda ou vértice de fronteira.

Existem outras variáveis. Vejamos: XD, YD e ZD com 100 posições (NDP) cada uma; armazenam as coordenadas dos pontos e são utilizadas por várias sub-rotinas. XI e YI, com 101 posições cada uma representam as coordenadas das linhas horizontais e verticais da malha regular. Devem ser fornecidas pelo usuário e são chamadas pela rotina IDGRID.

A região WK, com 500 posições (5*NDP) é comum a 3 sub-rotinas. Utilizada como área de trabalho em IDTANG (100 primeiras posições) é usada para armazenar as 5 derivadas de cada vértice em PD (5*NDP), que serão chamadas por IDPTIP, logo após o cálculo por IDPDRV.

Finalmente existe a área ZI, onde se armazenam os valores interpolados da variável z para cada um dos nós da malha. A dimensão de ZI é portanto igual à de IGP: 10.201 posições.

7.4. FRACIONAMENTO DO PROGRAMA DE GRAVAÇÃO EM DISCO

Constata-se facilmente pelo estudo acima que as grandes áreas de memória são IGP e ZI, consumindo o seguinte número de bytes:

IGP: 10.201 × 2 = 20.402 bytes (variável inteira)

ZI : 10.201 × 4 = 40.408 bytes (variável real)

Esses valores referem-se a uma malha de saída de 101×101 pontos e que consumiriam 61 k bytes, quase toda a capacidade do micro, não sendo possível nem gravar o programa. Temos portanto a primeira providência ser tomada: gravar em disco os vetores IGP e ZI.

Examinando as sub-rotinas verificamos que a maior de todas é a IDTANG, em termos de programa, e de memória, em que utiliza 3.100 posições (31*NDP) do vetor IWK, além de XD, YD e WK, com 100 posições (NDP) cada um.

Portanto, a segunda conclusão: essa é a sub-rotina limitante, que determinará o número máximo de pontos que podem ser processados de uma só vez.

As outras sub-rotinas, somadas, ocupam uma área de memória equivalente a IDTANG e portanto sugerem uma partição em 2 programas equivalentes em tamanho.

Denominamos o primeiro de TRIANG; é composto por IDTANG e pela função IDXCHG, chamada por ela. Denominamos o segundo de INTERP; inclui: IDCLDP, IDGRID, IDPDRV e IDPTIP.

De acordo com a teoria desenvolvida, substituímos a fun-

ção IDXCHG por IDTROQ; IDCLDP por IDVZVR e IDPDRV por IDPDER. Reduzimos e simplificamos as rotinas IDGRID e IDPTIP suprimindo a extrapolação e gravando IGP e ZI em disco.

Essas alterações foram testadas e os dois programas funcionaram bem. Nas Figuras 7.2 e 7.3 encontram-se os esquemas dos programas, original e modificado.

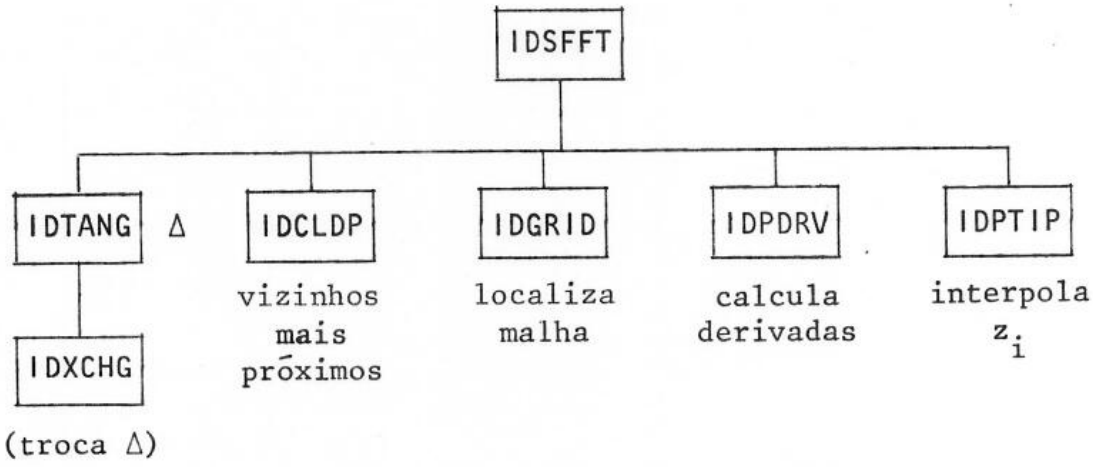


Figura 7.2 - Esquema original do programa de Akima, versão para grandes computadores.

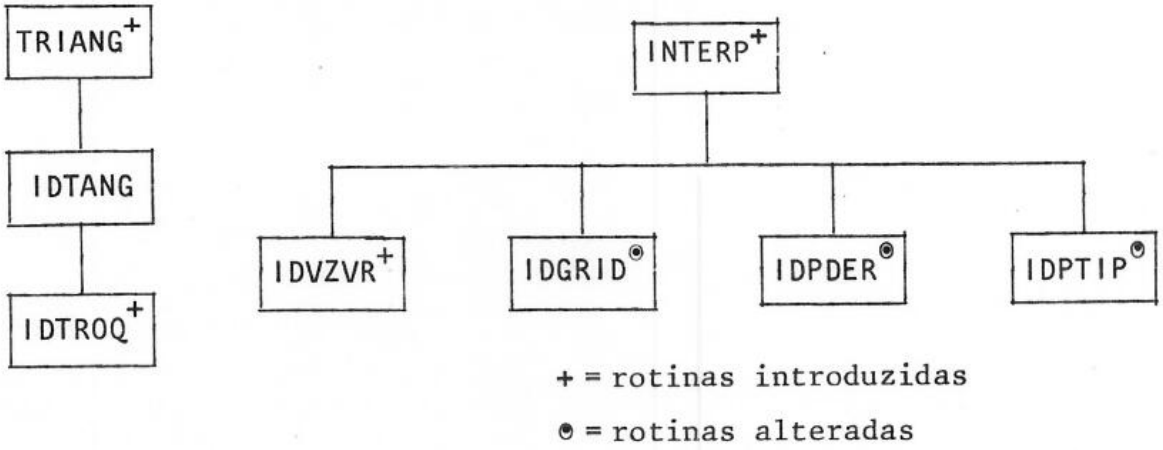


Figura 7.3 - Esquema adaptado para funcionamento em micros, tendo sidos introduzidas algumas melhorias.

7.5. CAPACIDADE MÁXIMA

O micro de 64 k possui na realidade a capacidade de

$$64 \times 1.024 = 65.536 \text{ bytes},$$

dos quais deve-se reservar uma área de aproximadamente 18.500 bytes que corresponde ao programa ligador de módulos (10.300) e ao sistema operacional (8.200).

Resulta portando uma capacidade realmente útil de aproximadamente 47 k, para ser repartida entre programa (incluindo as funções de biblioteca) e os dados.

Para uma comparação, apresentamos na Tabela 7.1 as áreas

Tabela 7.1 - Áreas ocupadas pelos programas: original e versão modificada

Nome do programa	Área para programa (+)	Área para dados (*)
IDSFFT (original)	38.500	8.500
TRIANG (triangulação)	16.200	30.800
INTERP (interpolação)	17.100	29.900

(+) = a área para programa refere-se à memória ocupada pelo programa, incluindo as funções biblioteca.
(*) = a área para dados obtém-se pela diferença entre a área para programa e a capacidade máxima (47k)
\- os valores da tabela são aproximados.

ocupadas pelo programa original de Akima e pelos dois módulos de nossa versão modificada.

A partir dessa tabela faremos alguns cálculos para deter-

minar a capacidade máxima de pontos processados.

a) TRIANG

Para o cálculo da capacidade máxima levamos em conta que as variáveis inteiras ocupam 2 bytes e as reais 4. As contas, em função do número (n) de pontos são as seguintes:

<u>variáveis em função de n:</u>

- IWK: $31n \cdot 2 = 62n$
- XD,YD: $2 \cdot n \cdot 4 = 8n$
- WK: $n \cdot 4 = 4n$ total parcial = 74n bytes

<u>variáveis fixas:</u> 300 bytes

* Total geral: 74n + 300 bytes.

Igualando esse total geral com a área máxima para dados da Tabela 7.1 obtemos:

$$74n_{max} + 300 = 30.800$$

donde

$$n_{max} = \frac{30.800 - 300}{74} = 412 \text{ pontos.}$$

Na prática utilizamos 400 pontos, o que deixou uma área livre de 900 bytes.

Se utilizássemos um programa ligador de módulos que consome menos memória poderíamos montar um programa com 48,5 k, expandindo a capacidade para 430 pontos já que os 1.500 bytes adicionais proporcionam mais uns 20 pontos (1.500/74).

Em um micro de 256 k bytes, a capacidade passaria a ser superior a 2.500 pontos, como se pode deduzir da fórmula acima.

b) INTERP

Para o cálculo da capacidade máxima de INTERP as contas foram as seguintes:

<u>variáveis em função de n:</u>

- IPC: $4 \cdot n \cdot 2 = 8 \cdot n$
- IPT: $6 \cdot n \cdot 2 = 12 \cdot n$
- IPL: $3 \cdot n \cdot 2 = 6 \cdot n$
- XD,YX,ZD: $3 \cdot n \cdot 4 = 12 \cdot n$
- WK: $5 \cdot n \cdot 4 = 20 \cdot n$ Total parcial = 58n bytes

<u>váriáveis fixas:</u>

As três primeiras tornam-se fixas devido à gravação em disco:

- NGP: $800 \cdot 2 = 1.600$
- IGP: $400 \cdot 2 = 800$
- ZI : $400 \cdot 4 = 1.600$

outras: 800 Total parcial = 4.800 bytes

* Total geral 58n + 4.800 bytes.

Igualando o total geral com a capacidade máxima para dados da Tabela 7.1 obtemos n_{max}.

$$58 \cdot n_{max} + 4.800 = 29.900;$$

donde

$$n_{máx} = \frac{29.900 - 4.800}{58} = 432 \text{ pontos.}$$

Na prática utilizamos 400 pontos já que esse programa é uma seqüência de TRIANG. As duas capacidades estão bastante balanceadas (principalmente se expandirmos TRIANG para 430), o que indica que a partição realizada é satisfatória.

Devido à gravação em disco dos valores de ZI interpolados, não há um tamanho limitante para a malha de saída.

c) IDSFFT

Para o cálculo da capacidade máxima do programa original sem fracionamento realizamos um cálculo análogo aos anteriores, chegando à equação:

$$130 \cdot n + 2.000 + 6\ m^2 = 8.500;$$

onde m^2 = número de pontos da malha. Daí tiramos:

$n_{máx}$ = 50 pontos (para o caso de não haver malha de saída);

$n_{máx}$ = 30 pontos (com uma malha de saída de 400 pontos: 20×20).

Como se pode ver, a capacidade do programa original é bastante limitada nos micros devido a área ocupada pelo programa, que deixa pouco espaço para os dados. Deve-se também à dimensão das áreas reservadas para a malha: deve-se gravar IGP e ZI

para cada nó da malha; se a malha for relativamente grande esses vetores influem muitíssimo na memória de dados.

A capacidade máxima do programa modificado pode ser aumentada adotando um algoritmo que realize a triangulação de Delaunay utilizando menos espaço de memória, coisa que é factível já que Lawson (1977) utiliza uma memória inferior a 18•n e Little (1983; no prelo) uma memória de 15•n. Esses métodos, no entanto, não fornecem a listagem de seus programas. Outra solução seria implantar o algoritmo de Lee & Schachter (1980) combinado com o de Akima (1978); conforme item 5.5.

7.6. IMPLANTAÇÃO DO SISTEMA

Resumindo as adaptações introduzidas e indicando outras, apresentamos na Figura 7.4 o esquema de implantação do sistema.

O usuário deve escolher um (nome) para o seu problema e fornecer o arquivo (nome). DAT, contendo o número de pontos, as coordenadas (x,y,z) dos vértices, a definição da malha de interpolação, o número e a definição das curvas a serem traçadas e a escala de desenho.

O programa TRIANG lê esses dados e realiza a triangulação de Delaunay, gravando o arquivo (nome). TRI.

O programa INTERP lê a triangulação e o arquivo .DAT e grava o arquivo (nome) .IGP, contendo o número de pontos da malha que pertencem a cada triângulo e quais são eles. Grava também o arquivo (nome) .ZTR que armazena as cotas dos pontos

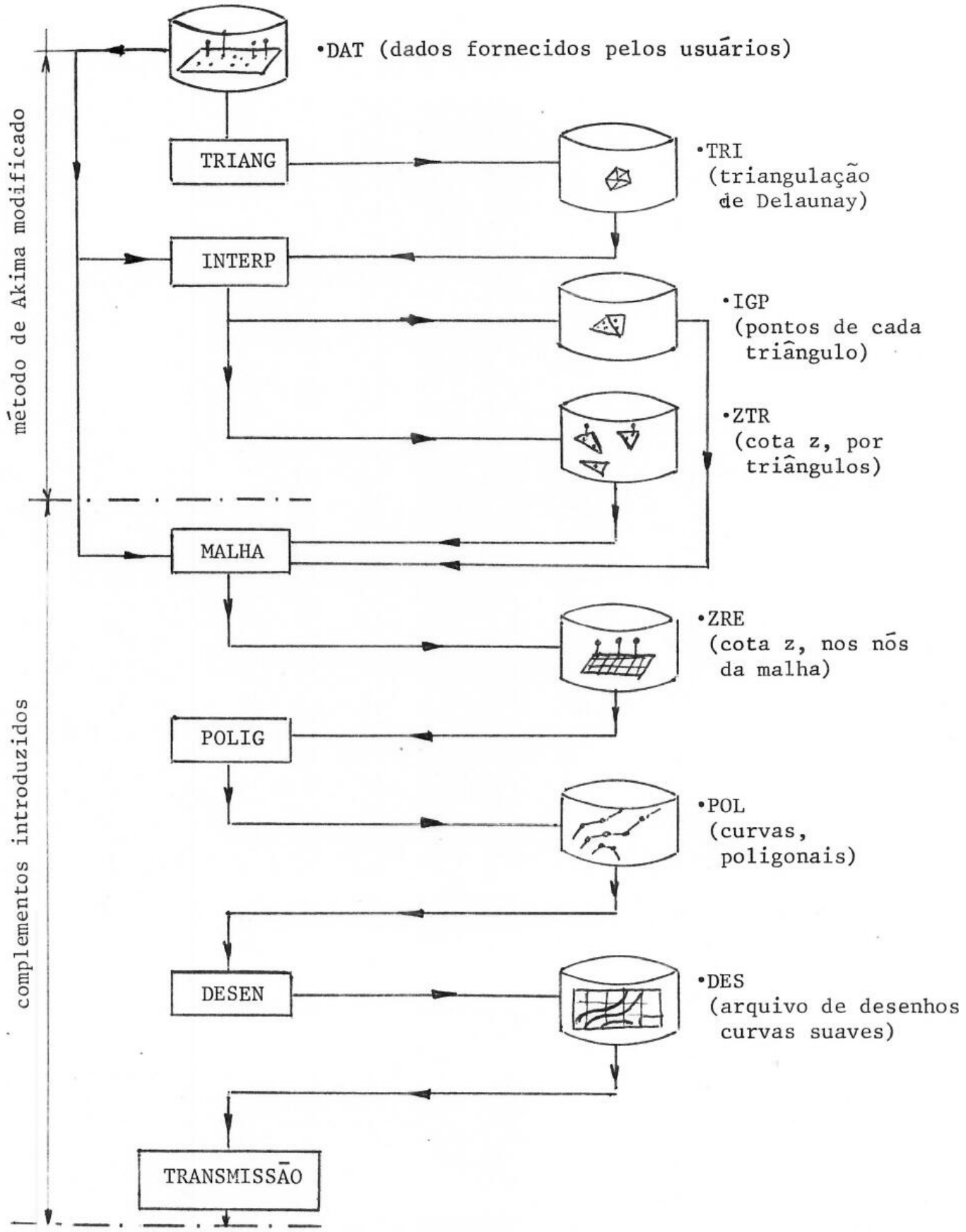


Figura 7.4 - Esquema de implantação do sistema

do arquivo .IGP, obedecendo a ordem dos triângulos.

Até este ponto segue-se o método de Akima, de acordo com as modificações introduzidas (conforme Figura 7.3). A seguir desenvolvemos o programa MALHA, que lê os arquivos .DAT e .ZTR, passa as cotas z para valores inteiros e grava o arquivo (nome) .ZRE, com as cotas ordenadas de acordo com a malha regular.

A seguir executa-se o programa POLIG, que lê .ZRE e fornece o arquivo (nome) .POL contendo seqüências de coordenadas x,y que representam as curvas de nível. Corresponde ao programa CNI modificado e adaptado (conforme Figura 6.5 e itens 6.5.1 e 6.5.2).

O próximo passo é o programa DESEN, que lê o arquivo .POL e gera o arquivo de desenho, suavizando as poligonais fornecidas através da adaptação do algoritmo de Cline (item 6.6.2) e completando o arquivo (nome) .DES com o desenho da quadrícula, do rótulo (cota) das curvas de nível, margens do desenho, etc.

Nesse momento são possíveis duas opções de saída (transmissão); pode-se comandar um 'plotter' de mesa (como por exemplo a HP 7475 que utilizamos) para desenhar um primeiro rascunho, ou pode-se gerar um arquivo em disquete para comandar mesas de desenho (em nosso trabalho utilizamos o 'plotter' eletrostático VERSATEC-8222A e o CALCOMP-960, de pena).

7.7. ESQUEMA DE FUNCIONAMENTO

Do ponto de vista prático é útil apresentar um possível

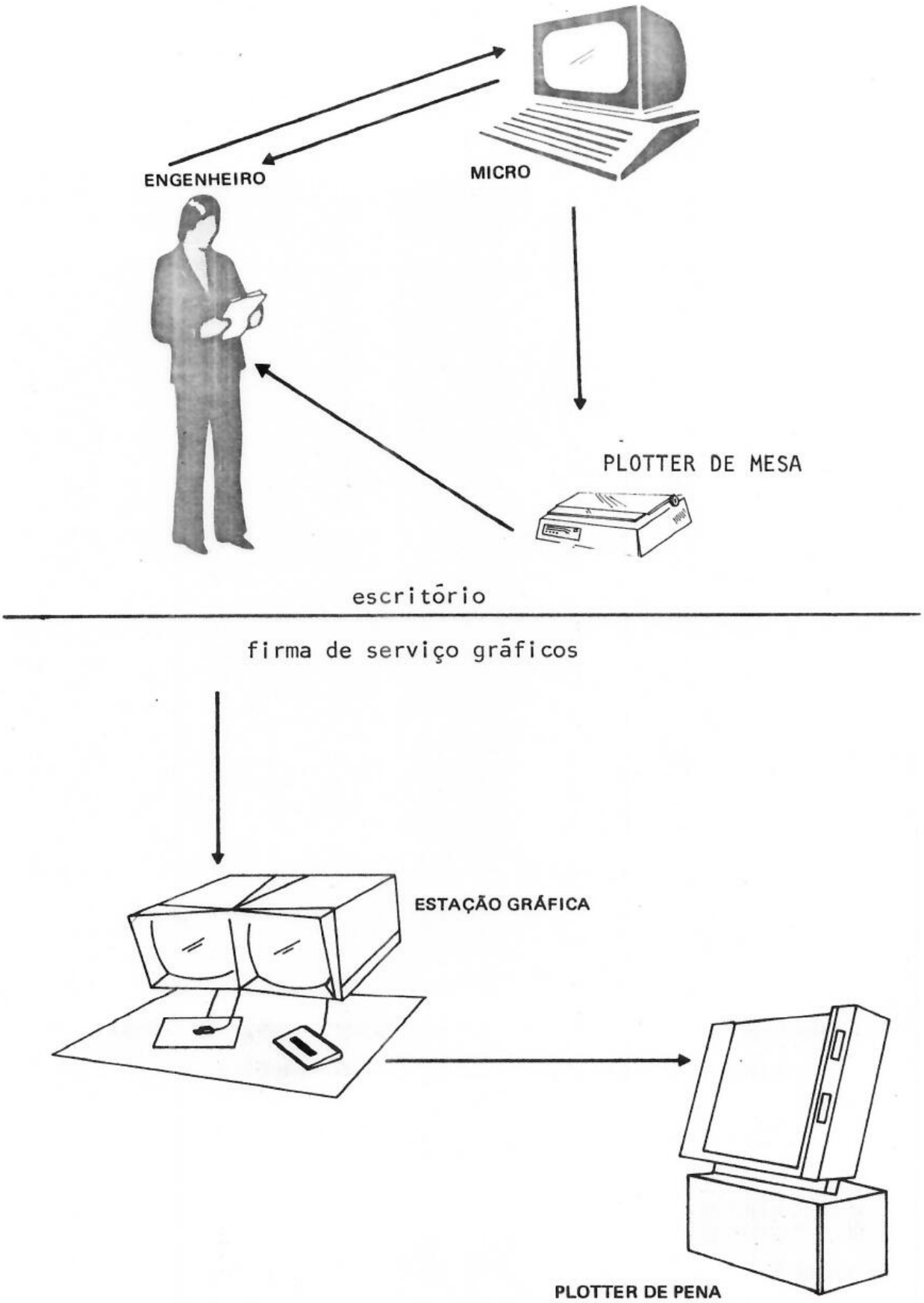


Figura 7.5 - Esquema de funcionamento para o desenho de curvas de nível com o auxílio do microcomputador.

esquema de funcionamento para escritórios de engenharia e pequenas firmas, que já está operando em larga escala, e com bons resultados, para o cálculo e desenho de vigas.

De acordo com o esquema da Figura 7.5, o usuário fornece os dados, efetua o processamento através do microcomputador e realiza um primeiro desenho em um 'ploter' de mesa, cujo custo é acessível.

Esse primeiro desenho pode ter diversas utilidades. Em alguns casos pode servir diretamente para projetos ou ante-projetos. Em outros pode fazer as vezes do original cartão, a partir do qual se tirem cópias em vegetal, introduzindo as modificações que se vejam oportunas. Pode servir como primeira aproximação para que o usuário analise e introduza correções diretamente no micro: suprimindo ou acrescentando pontos, melhorando a suavidade das curvas (parâmetro SIGMA), modificando a escala, etc.

Existe a limitação do tamanho do desenho (máximo A_3) e então será preciso recorrer a uma firma prestadora de serviços gráficos. O arquivo .DES é transferido (via disquete) para o equipamento de desenho dessa firma e as alterações pedidas (indicadas em um primeiro rascunho do 'ploter' de mesa, e em escala menor) são executadas com rapidez numa estação gráfica, e o desenho final é feito em um 'ploter' de pena, no tamanho desejado.

Um passo adiante nesse esquema pode ser dado caso de disponha de um micro com tela gráfica e possibilidade de intera-

ção (tipo IBM PC; Nexus, etc.). As alterações podem ser introduzidas diretamente através da tela, dispensando-se a estação gráfica.

7.8. A OPÇÃO POR MICROCOMPUTADORES

Do que já foi dito pode-se deduzir uma série de vantagens na utilização de micros.

Em primeiro lugar a praticidade e comodidade, já que o serviço pode ser executado no próprio escritório e evitam-se deslocamentos e filas de espera nas firmas de serviços de computação (processamento e cálculos).

Depois, a facilidade de fornecimento, introdução e correção de dados, bem como a possibilidade de interação e controle direto do processo.

Além disso, a possibilidade de utilizar a malha regular como um modelo digital do terreno para desenvolver interessantes problemas aplicativos para serem processados no próprio escritório: cálculos de volumes (cortes, aterros,plataformas, estradas), áreas (inundação de barragens, divisão de terras) e outros.

Finalmente, os custos, que são bem menores neste caso. Determinados problemas que só eram resolvidos em grandes computadores estão passando a ser processados por micros, que se demonstram altamente vantajosos para muitos problemas da engenharia.

A passagem de um sistema para outro depende da elaboração

de programas que se demonstrem realmente práticos para uso em micro, quanto à capacidade, tempo de processamento e qualidade do desenho.

Parece-nos que os resultados obtidos no presente trabalho são satisfatórios: capacidade para processar 400 pontos (450), que parece suficiente para a maioria dos problemas, em um tempo de pouco mais de 30 minutos. A qualidade do desenho é um pouco melhor que a do programa original de Akima, que é francamente aceitável.

Até o presente momento não se tem notícia de outras adaptações desse tipo, que são perfeitamente viáveis, contrariamente à idéia de muitos, de que esse tipo de programa não cabe nem nos minicomputadores.

- ■ -

V PARTE: CONCLUSÃO FINAL

8. CONCLUSÃO E DIRETRIZES PARA TRABALHOS FUTUROS

8.1 TRABALHOS FUTUROS

Em trabalhos futuros pensamos desenvolver a parte computacional referente a duas sugestões fornecidas ao longo deste trabalho: a construção da triangulação de Delaunay baseada no método da divisão e conquista (item 5.5) e o desenho das curvas de nível baseado no polinômio interpolador (item 6.6.4). Os fundamentos teóricos estão suficientemente detalhados para que isso seja feito com facilidade.

Pensamos desenvolver, também para micros, alguns programas aplicativos para o traçado de perfis, cálculos de áreas e volumes e o desenho de perspectivas para aplicação em arquitetura e urbanismo. Todas essas tarefas tornam-se bastante simples quando se dispõe de um modelo digital do terreno que nada mais é do que o armazenamento computacional de um modelo matemático que permita representar esse terreno e obter as informações e dados que se desejem.

O desenho de perfis pode iniciar-se com o levantamento de cotas ao longo da linha desejada. Caso essa linha coincida com alguma linha notável da malha quadrada, o trabalho se simplifica. Caso contrário deve-se recorrer à adequada interpolação de

valores que pode ser feita com o método já desenvolvido. A seguir pode-se unir suavemente esses pontos através de um dos métodos fornecidos nos itens 6.6.1 e 6.6.2 deste trabalho.

O cálculo de volumes (corte/aterro) pode ser feito quase que imediatamente através da malha quadrada e da definição dos limites da área que se vai escavar/aterrar. Os manuais de topografia fornecem as fórmulas para esse cálculo, que é bastante simples. Quando se quer ou se necessita calcular volumes através da área de secções transversais, o problema recai no item que veremos a seguir.

À primeira vista, a avaliação de áreas delimitadas por contornos irregulares não pode ser feito por métodos computacionais, e deveríamos recorrer a processos gráficos como a contagem de quadrículas, a divisão e substituição por figuras geométricas equivalentes, a divisão em faixas de largura constante e soma de alturas; ou a processos mecânicos, como a utilização de planímetros.

No entanto, o cálculo de áreas, para muitas aplicações como por exemplo o cálculo de volumes de reservatórios, pode ser feito com certa facilidade se tivermos em conta que dispomos de uma representação matemática das curvas de nível. Cada uma delas é definida por uma sequência de pontos cujas coordenadas (x,y) são conhecidas. Dispondo dos demais limites da área a ser avaliada, podemos fazer rapidamente o cálculo, através de conhecidas fórmulas para o cálculo da área interna de uma poligonal.

O desenho de perspectivas também é uma aplicação interessante pois permite uma maior visualização do terreno e dos pro

jetos implantados. Para gerar desenhos perspectivos como o da figura 8.1 pode-se utilizar programas computacionais já desenvolvidos, como por exemplo Watkins (1973), Williamson (1972) ou Wrigth (1974).

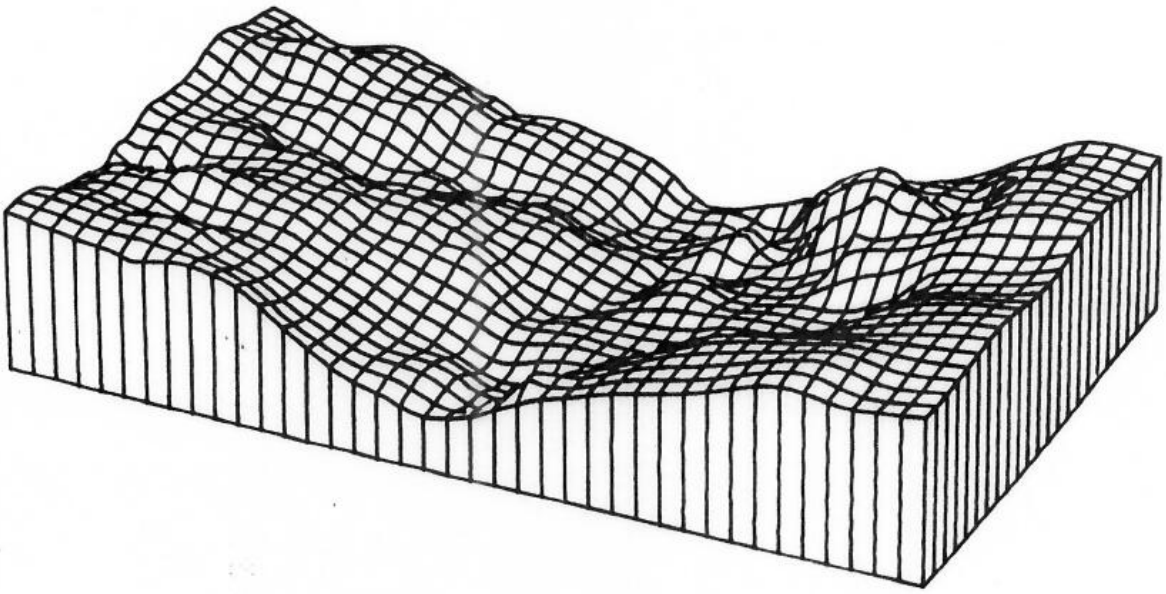


Figura 8.1 - Bloco perspectivo de um terreno.

Esses programas permitem variar a posição do observador, fazendo com que o terreno seja visto a partir de todos os ângulos desejados.

Permite portanto representar o impacto visual de obras arquitetônicas e de urbanização (estradas, plataformas, taludes, construções, etc.) através da comparação de perspectivas da região antes e depois (simulação) da implantação de determinados projetos.

8.2 CONCLUSÃO FINAL

Neste encerramento e conclusão final gostaríamos de expor também um sumário da metodologia de pesquisa utilizada neste

trabalho, que justifica sua sequência e estrutura.

A primeira parte corresponde a uma introdução geral em que salientamos o amplo espectro de aplicações das curvas de isovalor, nos diversos campos da ciência. Descrevemos também o processo clássico da construção de curvas de nível, que foi nossa área de atuação nesta Escola, durante muitos anos. A razão desta inclusão (capítulo 1) reside no contraste e na possibilidade de compará-lo com os processos computacionais (capítulo 2)

O primeiro problema com que se defronta quem se aproxima deste tema sem conhecimento prévio é entender os mecanismos através dos quais se pode desenhar curvas de nível com o auxílio do computador acoplado com equipamentos gráficos. Por isso se justifica o capítulo 2, ainda dentro da introdução geral. Ele é um apanhado das diversas soluções que foram dadas ao problema recorrendo a diversos modelos matemáticos. Desse estudo comparativo resultou também a conclusão de que existem soluções mais e menos satisfatórias e que todas podem e devem ser melhoradas.

Nossa contribuição consistiu em estudar um desses métodos que já apresenta boas qualidades, o de Akima, e sugerir alterações e complementos que conduzem a soluções ainda melhores.

Dessa forma, a parte II, é um estudo pormenorizado do método de Akima. O capítulo 3 estuda sua fundamentação teórica, procurando seguir fielmente a exposição original para facilita ao leitor a distinção entre o que é o método de Akima e o que contribuição nossa. O capítulo 4 é um detalhado estudo do programa computacional em que, partindo somente da listagem, desetranhamos os procedimentos implícitos e analisamos os fundamen

tos matemáticos envolvidos. Constitui-se em uma boa ajuda para quem quiser estudar o programa de Akima e, a partir dele, sugerir outras melhorias.

A partir desse ponto, todo desenvolvimento é nosso. As contribuições encontram-se portanto nas partes III e IV.

A parte III (capítulo 5), desenvolve estudos teóricos em torno dos pontos básicos de Akima: triangulação, determinação de vizinhos, cálculo das derivadas parciais e função interpoladora. Desse estudo teórico surgiram as seguintes contribuições:

* 1. Um novo critério para a definição da triangulação ótima: o critério do círculo. Os resultados práticos demonstraram que essa alteração poupa memória e torna o processo mais rápido

* 2. Um novo critério teórico para a definição dos vizinhos de cada vértice, que mostrou-se mais adequado do ponto de vista topológico e conduziu a melhores desenhos das curvas de nível, por estimar melhor as derivadas em cada ponto. Também é mais rápido do ponto de vista computacional.

* 3. A possibilidade de aplicar a técnica da divisão e conquista para poupar tempo de computação. Essa contribuição está em nível descritivo-detalhado.

A diminuição do tempo de computação, melhorando ao mesmo tempo a qualidade do desenho, é uma boa contribuição, por dois motivos. Em primeiro lugar porque o método de Akima é fruto de uma longa pesquisa e de um sério trabalho de otimização, apresentando algoritmos compactos e eficientes. Ele foi objeto de diversos estudos e modificações. No entanto, estudos recentes como o de Franke (1982) mostraram que até essa época ele ainda

era, de longe, o método mais rápido entre os trabalhos estudados por esse autor. Foram objetos dessa comparação 32 métodos selecionados dentre os melhores em uso nos EUA, e que são devidos a profissionais de gabarito como Shepard, Foley, Nielson, Lawson, Mc Lain, Little e o próprio Franke.

Sendo nosso método mais rápido que o de Akima, consome forçosamente menos tempo que todos os demais. E tendo em conta a categoria desses profissionais, isso é uma contribuição que tem o seu mérito, ainda que possam surgir mais adiante outros métodos mais rápidos.

Em segundo lugar, a economia de tempo não é uma simples brincadeira matemática em torno de algoritmos; ela significa uma redução no custo, coisa que viabiliza ainda mais a aplicação de computadores e, principalmente, a de microcomputadores. Com as modificações introduzidas conseguimos uma redução de cerca de 20% para problemas com mais de 200 pontos.

A parte IV (capítulos 6 e 7) é constituída por complementos ao programa de Akima. Como esse método não fornece rotinas para o cálculo e desenho das curvas de nível, esse é precisamente o objeto do capítulo 6, onde estudamos teoricamente o assunto, comparamos 3 métodos e desenvolvemos e adaptamos um deles ao método de Akima, que fornece somente as cotas, desordenadas, dos nós de uma malha quadrada. Nossa solução para o problema acrescenta também o fato de termos adaptado uma spline sob tensão para suavizar as curvas de nível obtidas (trechos de poligonal). Além disso, nesse mesmo capítulo, sugerimos detalhadamente um processo para desenhar as curvas de nível baseado di-

retamente no polinômio interpolador e não na malha regular.

No capítulo 7 descrevemos a adaptação feita de todo o sistema (método de Akima modificado e rotinas de desenho) para ser utilizado em microcomputadores. Essa solução tem evidentes vantagens, tanto de facilidade de utilização, desenvolvimento de aplicativos, como econômicas.

Os resultados obtidos demosntraram que a utilização em microcomputadores é perfeitamente viável e atende satisfatoriamente as necessidades da prática: é possível processar 400 pontos em cerca de 30 minutos.

Por tudo o que foi conseguido e apresentado, parece-nos que este trabalho apresenta as inovações e contribuições necessárias para torná-lo uma tese de doutoramento.

- ∎ -

BIBLIOGRAFIA

[1] - AHLBERG, J.H. e outros, *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.

[2] - AKIMA, HIROSHI, Bivariate interpolation and smooth surface fitting based on local procedures, *Comm. ACM 17* (1974), pp. 26-31.

[3] - AKIMA, HIROSHI, A method of bivariate interpolation and smooth surface fitting based on local procedures, *Comm. ACM 17* (1974), pp. 18-20.

[4] - AKIMA, HIROSHI, A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points, *ACM TOMS, 4,2* (1978), pp. 148-159.

[5] - AKIMA, HIROSHI, Collected algorithms from ACM, *Algorithms*, 433-526.

[6] - AKIMA, HIROSHI, Interpolation and smooth curve fitting based on local procedures, *Comm. ACM 15,10* (1972), pp. 914-918.

[7] - AKIN, J.E. & GRAY, W.H., Contouring on isoparametric surfaces, *International Journal for Numerical Methods in Engineering, 11*, (1977), pp. 1893-1897.

[8] - AKIN, J.E. & GRAY, W.H., An improved method for contouring on isoparametric surfaces, *International Journal for Numerical Methods in Engineering, 14* (1979), pp. 451-472.

[9] - BARNHILL, R.E., Smooth interpolation over triangles, in *Computer Aided Geometric Design*, Academic Press, 1974.

[10] - Barnhill, R.E.; BIRKHOFF, G. & GORDON, W.J., Smooth interpolation in triangles, *Journal of Approximation Theory 8* (1973), pp.114-128.

[11] - BARNHILL, R.E. & FARIN, G., C^1 quintic interpolation over triangles: two explicit representations, *International Journal for Numerical Methods in Engineering, 17* (1981), pp. 1763-1778.

[12] - BARNHILL, R.E. & GREGORY, J., Compatible smooth interpolation in triangles, *Journal of Approximation Theory 15* (1975) pp. 214-225.

[13] - BARNHILL, R.E. & MANSFIELD, LOUIS, Error bounds for smooth interpolation in triangles, *Journal of Approximation Theory, 11,* (1974), pp. 306-318.

[14] - BATEZ, JEAN-LOUIS, An explicit formulation for an efficient triangular plate-bending element, *International Journal for Numerical Methods in Engineering, 18*, (1982), pp. 1077-1089.

[15] - BELL, KOLBEIN, *Analysis of thin plates in bending using triangular finite elements,* Division of Structural Mechanics, The technical University of Norway, 1968.

[16] - BELL, KOLBEIN, A refined triangular plate bending finite elements, *International Journal for Numerical Numerical Methods in Engineering, 1* (1969), pp. 101-122.

[17] - BENGTSSON, B.E. & NORDBECK, S., Construction of isaritms and isarithmic maps by computers, *BIT, 4* (1964), pp. 87-105.

[18] - BENTLEY, J.L. & FRIEDMAN, J.H., Data structures for range searching, *ACM Computing Surveys 11,4* (1979), pp. 397-409.

[19] - BENTLEY, J.L. & SHAMOS, M.I., Divide and conquer for linear expected time, *Information Processing Letters, 7,2* (1978), pp. 87-91.

[20] - BENTLEY, J.L.; FAUST, M.G. & PREPARATA, F.P., Approximation algorithms for convex hulls, *Comm. ACM 25,1* (1982), pp. 64-68.

[21] - BENTLEY, J.L.; WEIDE, B.W. & YAO, A.C., Optimal expected-time algorithms for closest point problems, *ACM TOMS, 6,4* (1980), pp. 563-580.

[22] - BERGAN, P.G., *Plane stress analysis using finite element method. Triangular element with 6 parameter in each node,* Division of Structural Mechanics, The Tecnical University of Norway, 1967.

[23] - BESAG, J., Spatial interaction and the statistical analysis of lattice systems, *J. Royal Stat. Soc. B 36,2* (1974), pp. 192-236.

[24] - BIRKHOFF, GARRET, Piecewise bicubic interpolation and approximation in polygons, in *Approximation with Special Emphasis on Spline Functions*, J. Schoenberg (ed.), Academic Press,N.York, 1969, pp. 185-222.

[25] - BIRKHOFF, G. & DE BOOR, C., Piecewise polynomial interpolation and approximation, in *Approximation of Functions*,Garabendian (ed.) Elsevier, Amsterdam, 1965, pp. 164-190.

[26] - BIRKHOFF, G. & MANSFIELD, LOIS, Compatible triangular finite elements, *Journal of Mathematical Analysis and Applications 47*, (1974), pp. 531-553.

[27] - BOGUE, D.J., *The structure of the metropolitan community*, University of Michigan contributions of the Institute of Human Adjustment Social Science, Univ. of Michigan, Ann Arbor, 1949.

[28] - BOOR, CARL DE, Bicubic spline interpolation, *J. Math. and Phys. 41* (1962), pp. 212-218.

[29] - BOOR, CARL DE, On calculating with B-spline, *Journal of Approximation Theory 6* (1972), pp. 50-62.

[30] - BOWYER, A., Computing Dirichlet tessellations, *The Compt. Journal 24,2* (1981), pp. 162-166.

[31] - BRAMBLE, J. & ZLÁMAL, M., Triangular elements in the finite element method, *Mathematics of Computation, 24*, Oct. 1970, pp. 809-820.

[32] - BREED, CHARLES B., *The principles and pratice of surveying*, John Wiley & Sons, New York, (10ª ed.), 1966.

[33] - BROWN, WILLIAM G., Historical note on a recurrent combinatorial problem, *American Mathem. Monthly 72,9* (1965), pp. 973-977.

[34] - CADWELL, J.H. & WILLIAMS, D.E., Some orthogonal methods of curve and surface fitting, *Comput. Journal 4* (1961), p. 260 e ss.

[35] - CALCOMP, *GPCP - II USERS MANUAL*, Anaheim, USA, 1972

[36] - CARAMANLIAN, C., A solution to the C^1 continuity problem in plate bending, *International Journal for Numerical Methods in Engineering, 19* (1983), pp. 1291-1317.

[37] - CARAMANLIAN, C.; SELBY, K.A. & WILL, G. T., A quintic conforming plate bending triangle, *International Journal for Numerical Methods in Engineering, 12* (1978), pp. 1109-1130.

[38] - CARVALHO XEREZ, A. DE, *Topografia Geral,* Ed. Técnica,Lisboa,1947.

[39] - CAVENDISH, JAMES C., Automatic triangulation of arbitrary planar domains for the finite element method, *International Journal for Numerical Methods in Engineering, 8* (1974), pp.679-696.

[40] - CHUNG, WON L., Automatic curve fitting using an adaptive local algorithm, *ACM, TOMS, 6* (1980), pp. 45-57.

[41] - CHORLEY, R.J. & HAGGETT, P., *Modelos físicos e de informação em geografia,* EDUSP, São Paulo, 1975.

[42] - CLEARY, JOHN GERALD, Analysis of an algorithm for finding nearest neighbors in euclidean space, *ACM TOMS 5,2* (1979),pp.183-192.

[43] - CLINE, A.K., Scalar-and Planar-valued curve fitting using splines under tension, *Comm. ACM 17,4* (1974), pp. 218-220.

[44] - CLINE, A.K., *Algorithm 476 (CACM): Six subprograms for curve fitting using splines under tension,* 1974.

[45] - COTTAFAVA, G. & LE MOLI, G., Automatic contour map, *Comm.ACM 12,7* (1969), pp. 386-391.

[46] - CRANE, C.M.,Contour plotting for functions specified at nodal points of an irregular mesh based on an arbitrary two parameter coordinate system, *Comptr. Journal* (1972), pp. 382-384.

[47] - DAVIS, J.C., *Statistics and Data Analysis in Geology,* John Wiley & Sons, New York, 1973.

[48] - DAVIS, J.C. & McCULLAGH, M.J.,*Display and analysis of spatial data,* John Wiley & Sons, New York, 1978.

[49] - DAVIS, J.P., *Interpolation and Approximation,* Blaisdell,New York, 1963.

[50] - DAVIS, R.; FOOT, F. & KELLY, J.W., *Surveying - Theory and practice,* McGraw-Hill, 1966, (5ª edição).

[51] - DAYHOFF, M.O.,A contour map program of X-ray crystallography,*Comm. ACM 6* (1963), pp. 620-622.

[52] - DELVOS, Franz-Jürgen, On surface interpolation, *Journal of Approximation Theory, 15* (1975), pp. 209-213.

[53] - DINIZ DA GAMA, C. & PEREIRA, G., Métodos otimizados de mapeamento, *Sucessu, II° Simpósio sobre Aplicações Gráficas por Computador,* vol. II, 1982, pp. 11-29.

[54] - DIRICHLET, P.G.L., Über die reduction der positiven quadratischen ganzen zahlen, *Journal für die Reine und Angewandte Mathematik, 40* (1850), pp. 209-227.

[55] - DOMINGUES, FELIPPE A.A., *Topografia e astronomia de campo*,McGraw-Hill, São Paulo, 1980.

[56] - DÜPPE, R.D. & GOTTSCHALK, H.J., Automatische interpolation von isolinien bei willkürlich verteitten, *Allgemeine Vermessungsnachrichten, 77,* pp. 423-426, 1970

[57] - DURIS, CHARLES S., Fortran routines for discrete cubic spline interpolation and smoothing, *ACM TOMS, 6,1* (1980),pp. 92-103.

[58] - DYNAMICS GRAPHICS, *User Manual for 'Surface Gridding Library'*,Berkeley, USA, 1978.

[59] - EDDY, WILLIAM F., A new convex hull algorithm for planar sets, *ACM TOMS, 3,4* (1977), pp. 398-403.

[60] - EDDY, WILLIAM F., *Collected algorithms of ACM, algorithm 523:Convex.*

[61] - ELLIS, T.M. & McLAIN, D., A new method of cubic curve fitting using local data, *ACM TOMS, 3,2* (1977), pp. 175-178, (algoritmo 514 ACM).

[62] - ERH-RONG, WU, A cubic triangular element with local continuity, *International Journal for Numerical Methods in Engineering, 17* (1981), pp. 1147-1159.

[63] - EWING, D.J.F.; FAWKES, A.J. & GRIFFITHS, J.R., Rules governing the numbers of nodes and elements in a finite element mesh, *International Journal for Numerical Methods in Engineering, 2,* (1970), pp. 597-601.

[64] - FARIN, G., Smooth interpolation to scattered 3d data surfaces, in *CAGD*, North Holland, Amsterdam, pp. 43-63, 1983

[65] - FERGUSON, JAMES, Multivariable curve interpolation, *Journal of ACM*, *11,2* (1964), pp. 221-228.

[66] - FRANKE, RICHARD, Scattered data interpolation: test of some methods, *Mathematics of Computation* , *38, nº 157* (1982), pp.181-200.

[67] - FRANKE, RICHARD, Locally determined smooth interpolation at irregularly spaced points in several variables, *J. Inst. Maths. Applics.*, *19* (1977), pp. 471-482.

[68] - FRANKE, RICHARD & NIELSON, GREG, Smooth interpolation of large sets of scaretted data, *International Journal for Numerical Methods in Engineering*, *15* (1980), pp. 1691-1704.

[69] - FRANKE, RICHARD & NIELSON, GREG, Surface approximation with imposed conditions, *Surfaces in CAGD*, North-Holland, pp. 135-146, 1983

[70] - FRIEDMAN, J.H.; BENTLEY, J.L. & FINKEL, R.A., An algorithm for finding best matches in logarithmic expected time, *ACM TOMS* *3,3* (1977), pp. 209-226.

[71] - FÜHRING, H., The application of node-element rules for forecasting problems in the generation of finite elements meshes, *International Journal for Numerical Methods in Engineering* *9*, (1975), pp. 617-629.

[72] - GAREY, M.R.; JOHNSON, D.S.; PREPARATA, F.P. & TARJAN, R.E., Triangulating a simple polygon, *Information Processing Letters*, *7* (1978), pp. 175-179.

[73] - GILBERT, E.N., Random subdivision of space into crystals, *Ann. Math. Stat.*, *33* (1962), pp. 958-972.

[74] - GOLD, C.M.; CHARTERS, T. & RAMSDEN, Automated contour mapping using triangular data structures and an interpolant over each irregular triangular domain, *Comput. Graphics*, *11* (1977), pp. 170-175.

[75] - GODIM, CARLOS A.A., Sigma-um sistema econômico e eficiente de mapeamento automático, *Sucessu, IIº Simpósio sobre Aplicações Gráficas por Computador*, Vol. II, pp. 31-45.

[76] - GREEN, P.J. & SIBSON, R., Computing Dirichlet tesselations in the plane, *Computer J. 21,2* (1978), pp. 168-173.

[77] - HALL, C.A., Bicubic interpolation over triangles, *Journal of Mathematics and Mechanics, 19,*(1969), pp. 1-11.

[78] - HALL, C.A., Natural cubic and bicubic spline interpolation, *SIAM J. Numer. Anal., 10,6* (1973), pp. 1055-1060.

[79] - HAUSSMANN, W. & MUNCH, H.J., On the construction of multivariate spline systems, in *Approximation Theory,* G. G. Lorentz (ed.) Academic Press, New York, 1973, pp. 373-378.

[80] - HOSAKA, M., Theory of curve and surface synthesis and their smooth fitting, *Information Processing in Japan, 9* (1969), pp. 60-68.

[81] - IBM - Data Processing Appl., *Numerical surface techniques and contour map plotting,* White Plains, New York, 1965.

[82] - IRONS, B.M., A conforming quartic triangular element for plate bending, *International Journal Numerical Methods in Engineering, 1* (1969), pp. 29-65.

[83] - JANCAITIS, J.R. & JUNKINS, J.L., Modeling n-dimensional surfaces using a weighted function approach, *54th. Anual Meeting of American Geophysical Union,* Washington, D.C., 1973.

[84] - JORDAN, W., Tratado general de topografía, Gili, Barcelona, 1944.

[85] - KIANG, T., Random Fragmentation in two and three dimensions, *Z. Astrophysik, 64* (1966), pp. 433-439.

[86] - KISSAM, PHILIP, *Surveying,* McGraw-Hill, New York, 1947.

[87] - KLUCEWICZ, I.M., A piecewise C^1 interpolant to arbitrarily spaced data, *Computer Graphics and Image Processing, 8* (1978),pp.92-112.

[88] - KUNZ, K.S., *Numerical Analysis,* Mc-Graw-Hill, New York, 1957.

[89] - KONTOPIDIS, G. & LIMBERT, D.E., A predictor-corrector countouring algorithm for isoparametric 3-d element, *International Journal Numerical Methods in Engineering, 19* (1983), pp.995-1004.

[90] - KRIGE, D.G., Two-dimensional weighted moving average trend surfaces for ore valuation, *Journal South African Inst. Mining Metallurgy(Proc. Symposium Mathematical Statistics Computer Applications Ore Valuation),* (1966), pp. 13-79.

[91] - LADEIRA, CARLOS RODRIGUES, *Apostilas e Notas de aula - Topografia I*, EPUSP, (publicação interna).

[92] - LAURENT, P.J., *Approximation et Optimisation*, Hermann, Paris,1972.

[93] - LAWSON, C.L., Generation of a triangular grid with applications to contour plotting, *Technical Memorandum 299*,Jet Propulsion Laboratory, Pasadena, California, 1972.

[94] - LAWSON, C.L., Software for C^1 surface interpolation, in *Mathematical Software III*, J. Rice (ed.), Academic Press, New York, 1977.

[95] - LEE, D.T., On finding k-nearest neighbors in the plane, *Technical Report Eng. 76-2216, R. 728*. Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, 1976.

[96] - LEE, D.T., Two-dimensional Voronoi diagrams in the L_p-metric,*Journal ACM, 27,4* (1980), pp. 604-618.

[97] - LEE, D.T. & SCHACHTER, B.J., Two algorithms for constructing a Delaunay triangulation, *Int. Journal of Computer and Information Sciences 9,3* (1980), pp. 219-241.

[98] - LEE, D.T. & WONG, C.K., Voronoi diagrams in L_1 (L_∞) metrics with two-dimensional storage applications, *SIAM J. Computing 9,1*, (1980), pp. 200-211.

[99] - LEWIS, B.A. & ROBISON, J.S., Triangulation of planar regions with applications, *Computer Journal 21,4* (1978), pp. 324-332.

[100] - LUNG-AN,YING, Some 'special' interpolation formulae for triangular and quadrilateral elements, *International Journal for Numerical Methods in Engineering, 18,* (1982), pp. 959-966.

[101] - McCALLUM, D. & AVIS, D., A linear algorithm for finding the convex hull of a simple polygon, *Information Processing Letters, 9,5* (1979), pp. 201-206.

[102] - McCONALOGUE, D.J.,An automatic french-curve procedure for use with an incremental plotter, *Comptr. J. 14,2* (1971),pp.207-209.

[103] - McCULLAGH, M.J., Automatic contouring of faulted seismic data sets, *Eurographics UK chapter conference,* University of York ,(1983) pp. 57-66.

[104] - McLAIN, D.H., Two dimensional interpolation from random data, *The Computer Journal, 19-2,* (1976), pp. 178-181.

[105] - McLAIN, D.H., Errata, *The Computer Journal, 19,* (1976), p. 384.

[106] - MANSFIELD, L.E., Interpolation to boundary data in triangles with applications to compatible finite elements, in *Approximation Theory II,* por Lorentz (ed.), Academic Press, 1976.

[107] - MATERN, B., Spatial variation, *Medd. Statens Skogsforskninstit,36,* 5 (1960), pp. 1-144.

[108] - MEAD, R. Models for interplant competitions in irregularly spaced populations, in *Statistical Ecology,* Patil (ed.), vol. 2,Penn St. Univ. Press, pp. 13-30, 1971

[109] - MEEK, J.L. & BEER, G., Contour plotting of data using isoparametric element representation, *International Journal for Numerical Methods in Engineering, 10* (1976), pp. 954-957.

[110] - MILES, Roger E., Solution to problem 67-15: Probability distribution of a network of triangles, *SIAM Rev., 11* (1969), pp. 399-402. (solução)

[111] - MILES, Roger E., On the homogeneous planar Poisson point-process, *Math. Biosciences, 6* (1970), pp. 85-127.

[112] - MIRANTE, A. & WEINGARTEN, N., The radial sweep algorithm for constructing triangulated irregular networks, *IEEE Computer Graphics and Applications,* maio de 1982, pp. 11-21.

[113] - MORSE, STEPHEN P., Concepts of use in contour map processing,*Comm. of ACM, 12,3* (1969), pp. 147-152.

[114] - MOSCARDINI, A.O.; LEWIS, B.A. & CROSS, M., AGTHOM – Automatic generation of triangular and high order meshes, *International Journal for Numerical Methods in Engineering, 19* (1983); pp. 1331-1353.

[115] - MUNTEANU, MARIE-JEANNE, Generalized smoothing spline functions for operators, *SIAM J. Numer. Anal., 10,1* (1973), pp. 28-34.

[116] - MUNTEANU, M.J. & SCHUMAKER, L.L., Some multidimensional spline approximation methods, *Journal of Approximation Theory 10,*(1974) pp. 23-40.

[117] - NAVLAKHA, JAINENDRA K., An analytical technique for 3-dimensional interpolation, *BIT 24*, (1984), pp. 123-127.

[118] - NELSON, J.M., A triangulation algorithm for arbitrary planar domains, *Appl. Math. Modelling 2* (1978), pp. 151-159.

[119] - NIELSON, GREGORY M., A method for interpolating scattered data based upon a minimum norm network, *Mathematics of Computation, 40, 161*, (1983), pp. 253-271.

[120] - NOVEMBRE, ROBERTO LEITÃO DE, Aerofotogrametria como suporte na formação de um banco de dados, *Sucessu, II° Simpósio sobre Aplicações Gráficas por Computador*, Vol. II, 1982, pp.65-79.

[121] - OLEA, R.A., Optimal contour mapping using a universal kriging, *J. Geophysic. Res., 79,5*, pp. 695-702.

[122] - PELTO, CH. R.; ELKINS, T.A. & BOYD, H.A., Automatic contouring of irregularly spaced data, *Geophysics, 33,3* (1968), pp. 424-430.

[123] - PETTINATI, FÁBIO, Digital terrain modelling – a brazilian approach, *Proc. of the 1st int. conf. on computer appl.*, Amsterdan, 1983.

[124] - PETTINATI, FÁBIO, Terrain modelling and presentation, Artigo não publicado.

[125] - PIELOU, E.C., *Mathematical Ecology*, John Wiley and Sons, New York, 1977.

[126] - POWELL, M.J.D. & SABIN, M.A., Piecewise quadratic approximation on triangles, *ACM TOMS 3,4* (1977), pp. 316-325.

[127] - PREPARATA, F.P. & HONG S.J., Convex hull of finite sets of points in two and three dimensions, *Comm. ACM 20,2* (1977), pp. 87-93.

[128] - RAISZ, ERWIN, *Cartografia Geral*, Científica, Rio de Janeiro, 1969.

[129] - RHYNSBURGER, DIERK, Analytic delineation of Thiessen polygons, *Geographical Analysis, 5,2* (1973), pp. 133-144.

[130] - RICE, John R., ADAPT, Adaptative smooth curve fitting, *ACM TOMS, 4,1* (1978), pp. 82-94.

[131] - RIPLEY, B.D., Modelling spatial patterns, *Journal of the Royal Statistical Society*, serie B, Vol. 39, pp. 172-212, 1977

[132] - RIVARA M. CECILIA, Algorithms for refining triangular grids suitable for adaptive and multigrid tecnhniques, *International Journal for Numerical Methods in Engineering*, 20 (1984), 745-756.

[133] - ROGERS, C.A., *Packing and Covering*, Cambridge, University Press, 1964.

[134] - SAMPSON, ROBERT, J., *Surface II graphics system*, manual do usuário.

[135] - SCHACHTER, B.; ROSENFELD, A. & DAVIS, L.S., Random mosaic for textures, *IEEE Trans. Systems, Man and Cybernetics 8,9* (1978), pp. 694-702.

[136] - SCHACHTER, B.J., Decomposition of polygons into convex sets, *IEEE Trans. Comp. C-27,11* (1978), pp. 1078-1082.

[137] - SCHAGEN, I.P., Automatic contouring from scattered data points, *Comptr. Journal, 25,1* (1982), pp. 7-11.

[138] - SCHUMAKER, LARRY L., Fitting surfaces to scattered data, in *Appromation Theory II*, Lorentz, G.G. (ed.), Academic Press, London, (1976), pp. 203-268.

[139] - SHAMOS, M., *Computational Geometry*, Springer-Verlag, N. York, 1977

[140] - SHAMOS, M.I. & HOEY, D., Closest-point problems, *Proceedings of the 16th. Annual Symposium on the Foundations of Computer Science* (1975), pp. 151-162.

[141] - SIBSON, R., Locally equiangular triangulations, *Computer Journal, 21,3* (1978), pp. 243-245.

[142] - SIBSON, ROBIN & THOMSON, G.D., A seamed quadratic element for contouring, *The Compt. Journal* 24-4, (1981).

[143] - SMIRNOV, V., *Cours de mathématiques supérieurs*, Ed. MIR, Moscou 1969, Tomo I, pp. 385 e ss.

[144] - SNYDER, W.V., Contour plotting, *ACM TOMS, 4,3* (1978), pp. 290-294.

[145] - SNYDER, W.V., *Collected algorithms from ACM*, Algorithm 531.

[146] - SPEC, *Manual do Usuário*.

[147] - STEARNS, M.B., Probability distributions of a network of triangles (Problem 67-15), *SIAM Rev. 11* (1969), p. 399.

[148] - SUHARA, J. & FUKUDA, J., Automatic mesh generation for finite element analysis, Advances in computational methods in strutural mechanics and designs, UAH Press, Alabama, 1972.

[149] - SWITZER, P., Reconstructing patterns from sampled data, *Ann. Math. Stat., 38* (1967), pp. 138-154.

[150] - THIESSEN, A.H., Precipitation averages for large areas, *Monthly weather rev., 39* (1911), pp. 1082-1084.

[151] - TOBLER, WALDO R. (ed.), *Selected computer programs*, Michigan Geography Pub., Dept. of Geography, Univ. of Michigan, Ann Arbor, 1970.

[152] - VORONOI, G., Nouvelles applications des parametres continus à la théorie des formes quadratiques, Deuxième mémoire: rechercees sur les parallélloèdres primitifs, *Journal für die reine und angewandte mathematik, 134* (1908), pp. 198-287.

[153] - WATKINS, S.L., Masked three-dimensional plot program with rotations, *CACM nº 483*, 1973.

[154] - WATSON, D.F., Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes, *The Compt. J., 24,2* (1981), pp. 167-172.

[155] - WHITTEN, E.H.T. & KOELLING, M.E.V., *Computation of bicubic-spline surfaces for irregularly-spaced data*, T.R.3, Dept. of Geol. Sci., Northwestern U. Evanston, Ill, 1975.

[156] - WILLIAMSON, HUGH, Hidden-Line plotting program, *CACM nº 420*, 1972.

[157] - WRIGHT, THOMAS, Visible surface plotting program, *CACM nº 475*, 1974.

[158] - WU, SHENG-CHUAN e outros, An interactive computer graphics approach to surface representation, *Comm. ACM, 20,10* (1977), pp. 703-712.

[159] - ZENISEK, ALEXANDER, Interpolation polynomials on the triangles, *Numer. Math. 15*, (1970), pp. 283-296.

[160] - ZIENKIEWICZ, O.C., *The finite element method in structural and continuum mechanics*, McGraw-Hill, Londres, 3ª ed., 1977.

[161] - ZLAMAL, MILOS, On the finite element method, *Numerical Mathematical, 12*, (1968), pp. 394-409.

ADENDO BIBLIOGRÁFICO

[162] - AKIMA, HIROSHI, A method of bivariate interpolation and smooth surface fitting for values given at irregularly distributed points *U.S. Govt. Printing office*, O.T. Rep. 75-70, Washington, 1975.

[163] - BARNHILL, R.E., Representation and approximation of surfaces, in *Mathematical Software III* (J.R.Rice, ed.), Academic Press, N.Y. 1977, pp. 69-120.

[164] - BYKAT, A., Automatic generation of triangular grids, *Int. J. num. Math. Engng, 10, (1976)*, pp. 1329 e ss.

[165] - LLOYD, E.L., On triangulation of a set of points in the plane, T.M nº 88, MIT, Cambridge, Massachussets, 1977

[166] - LITTLE, Frank, CAGD report, University of Utah, no prelo (1983).

[167] - McLAIN, Dermot H., Drawing contours from arbitrary data poits, *Compt. J., v.* 17, 1974, pp. 318-324.

[168] - MAUDE, A.D., Interpolation-mainly for graph plotters, *Compt. J.*, v. 16, 1973, pp. 64-65

[169] - MOSCARDINI,A.O.;CROSS,M. e LEWIS,B.A., Assesment of three automatic triangular mesh generators for planar regions, *Adv. Num. Software 3 (3)*, 1981.

[170] - PETTINATI, Fábio, Modelamento digital e representação gráfica de superfícies, dissertação de mestrado apresentada à EPUSP, 1983

[171] - SABIN,M.A., Countouring - a review of methods for scattered data, in *Mathematical methods in computer graphics and design* (K.W. Brodlie, ed.), Academic Press, N.Y., 1980, pp. 63-85.

[172] - SHEPARD, Donald, A two-dimensional interpolation function for irregularly spaced data, Proc. 23rd Nat. Conf. ACM, 1968, pp. 517-523.

3 cópias

3,80 a 29 0,10
1,60 a 29 0,20

11/06/86		
~~11/09/89~~		
17/01/91		
~~29.01.91~~		
~~14/02/91~~		
28/12/06		
19/10/07		
19/11/07		

presentação de superfícies com o
auxílio do computador.