Interpolação Fractal Aplicada à Modelos Chuva-vazão Distribuídos.

Rosa Virginia Nunes Henriques dos Santos

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

Aprovada por:

Prof. Paulo Canedo de Magalhães, Ph.D.
(Presidente)

Prof. Luiz Carlos Guimarães, Ph.D.

Prof. José Paulo Soares de Azevedo, Ph.D.

Rio de Janeiro, RJ - BRASIL
ABRIL de 1994

SANTOS, ROSA VIRGINIA NUNES HENRIQUES DOS

Interpolação Fractal Aplicada a Modelos Chuva-vazão Distribuídos [RIO DE JANEIRO] 1994.
CXXXII, 132 p. 29,7 cm (COPPE/UFRJ), M. Sc., Engenharia Civil, 1994.
Tese - Universidade Federal do Rio de Janeiro, COPPE.
1 Modelo Chuva-vazão 2 Modelo Digital de Terreno
3 Interpolação Fractal
I. COPPE - UFRJ II. Título (Série)

Agradecimentos

Ao orientador Paulo Canedo de Magalhães pela confiança, oportunidade e acompanhamento do trabalho.

Ao co-orientador Luiz Carlos Guimarães e ao prof. Ronaldo César Marinho Persiano pela grande contribuição no desenvolvimento deste trabalho.

Aos amigos José Paulo Soares de Azevedo e Adilson Elias Xavier pelo apoio e incentivo que muito me foram necessários.

Aos amigos Newton, Lígia, Maria Clara, Fernanda, Helena, Fernando Menezes, Cezar e Mário por sua inestimável ajuda, dedico a minha eterna gratidão.

Aos estagiários do Laboratório de Hidrologia, em especial Leonardo, Eduardo e Simone, pela ajuda e paciência.

A todos os membros do programa de Engenharia Civil, em especial aos de Recursos Hídricos, pela oportunidade única que me foi dada pela qual terei uma eterna dívida de gratidão.

Ao amigo e mentor Nilo Koschek das Chagas pela preciosa orientação profissional.

A meus pais e irmãos pelo incentivo e carinho.

A meu marido, Luiz Carlos Carvalho dos Santos, pelo apoio e compreensão sem os quais eu não teria levado adiante meu trabalho.

À minha filha Camila pela tolerância com minhas ausências necessárias.

Resumo da tese apresentada à COPPE/UFRJ como proposta dos requisitos necessários para a obtenção do grau de mestre em ciências (M. Sc.).

Interpolação Fractal Aplicada à Modelos Chuva-vazão Distribuídos

Rosa Virginia Nunes Henriques dos Santos
Abril de 1994

Orientador: Prof. Paulo Canedo de Magalhães
Programa: Engenharia Civil

Nos últimos anos tem sido crescente o estudo de modelos distribuídos chuva-vazão que possibilitam levar em conta a variabilidade espacial das características físicas de uma bacia. Nestes modelos é essencial a utilização de modelos digitais de terreno que são responsáveis pelo armazenamento e manipulação dos dados referentes ao relevo e conseqüente obtenção de parâmetros fisiográficos necessários à modelagem do regime hidrográfico de uma bacia.

A modelagem eficiente do relevo com boa resolução torna estes modelos dependentes de uma grande base de dados.

Neste trabalho utiliza-se um algorítmo de interpolação fractal para a modelagem do relevo e derivação dos parâmetros a ele relacionados.

É implementado também um modelo chuva-vazão distribuído simplificado com o intuito de aplicar o modelo digital de terreno desenvolvido neste trabalho.

A principal característica desta modelagem, para um certo nível de detalhamento, está na redução do tamanho da base de dados, que possibilita o uso de uma resolução maior que a obtida por métodos convencionais para um mesmo esforço computacional.

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

Fractal Interpolation Applied to Distributed Rainfall-runoff Models

Rosa Virginia Nunes Henriques dos Santos
April of 1994

Thesis Supervisor: Prof. Paulo Canedo de Magalhães
Departement: Civil Engineering

In the last years, research on distributed rainfall-runoff models has been increasing. These models take into account the spatial variability of the catchment features. In these models it is essential the use of terrain digital models that are responsible for the storage and management of the data related to relief and which provide fisiographic atributes needed to distributed rainfall-runoff models.

These models depend on the large data bases to model efficiently the relief.

This work uses a fractal interpolation algorithm to model the relief and to compute the parameters related to the relief.

This work also implements a simple distributed rainfall-runoff model in order to apply the digital elevation model developed in this work.

The main feature of this modeling is the reduction of the size of the data base; this allows the use of a higher resolution than that used for conventional methods for the same computacional effort.

ÍNDICE

INTRODUÇÃO ---------- 1

CAPÍTULO I 3

REVISÃO BIBLIOGRÁFICA E FUNDAMENTOS TEÓRICOS 3

I.1 - INTRODUÇÃO ---------- 3

I.2 - MODELOS CHUVA-VAZÃO DISTRIBUÍDOS---------- 4
I.2.1 - Estruturas dos modelos distribuídos-- 4
I.2.2 - Modelos digitais de terreno (MDT)---- 6

I.3 - TEORIA DOS FRACTAIS---------- 8
I.3.1 - Introdução---------- 8
I.3.2 - Sistemas de funções iterativas------ 10
I.3.2.1 - Definição---------- 10
I.3.2.2 - Sistema dinâmico---------- 11
I.3.2.3 - Sistemas dinâmico de "shift" randômico---------- 12
I.3.3 - Interpolação fractal utilizando sistemas de funções iterativas------- 13
I.3.3.1 - Cálculo dos parâmetros do sistema de funções iterativas-- 15
I.3.3.2 - Propriedades da interpolação fractal---------- 18

CAPÍTULO II

MODELO DIGITAL DE TERRENO UTILIZANDO INTERPOLAÇÃO FRACTAL 19

II.1 - TRIÂNGULAÇÃO E CÁLCULO DOS PARÂMETROS------ 19

II.2 - COMO CALCULAR ALTITUDES A PARTIR DO SISTEMA DE FUNÇÕES ITERATIVAS---------- 21

II.3 - PRECISÃO E CONVERGÊNCIA---------- 23
II.3.1 - Número de iterações e estimativa de z_o inicial---------- 23
II.3.2 - Erros associados ao sistema dinâmico de shift randômico---------- 24
II.3.3 - Erros associados à localização de um ponto dentro da malha de triângulos---------- 25

II.4 - DECLIVIDADE EM UM PONTO QUALQUER DA SUPERFÍCIE---------- 26

II.5 - RUGOSIDADE DA SUPERFÍCIE---------- 30

II.6 - ESQUEMA DO ALGORÍTMO---------- 31

CAPÍTULO III

MODELO CHUVA-VAZÃO DISTRIBUÍDO 34

III.1 - INTRODUÇÃO---------- 34

III.2 - SIMULAÇÃO DOS PROCESSOS FÍSICOS---------- 37
III.2.1 - Distribuição da precipitação pela bacia---------- 37
III.2.2 - Perdas por evaporação---------- 37
III.2.3 - Perdas por infiltração---------- 38
III.2.4 - Escoamento superficial---------- 39

III.3 - CÁCULO DOS PARÂMETROS FISIOGRÁFICOS A PARTIR DE MODELO DIGITAL DE TERRENO--------- 41
III.3.1 - Introdução---------- 41
III.3.2 - Declividade das quadrículas

da malha---------- 42
III.3.3 - Direção de escoamento---------- 43
III.3.4 - Taxa de infiltração---------- 43
III.3.5 - Velocidade de escoamento superficial nas quadrículas---------- 44
III.3.6 - determinação automática de rede de rios ---------- 45

CAPÍTULO IV

APLICAÇÃO DO MODELO 47

IV.1 - REGIÃO ESCOLHIDA---------- 47

IV.2 - SIMULAÇÃO DO RELEVO---------- 47

IV.3 - CÁLCULO DOS PARÂMETROS PARA O MODELO HIDROLÓGICO---------- 51

IV.4 - SIMULAÇÃO DE UM EVENTO CHUVOSO---------- 55

CAPÍTULO V

CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS 58

V.1 - CONCLUSÕES---------- 58

V.2 - SUGESTÕES PARA TRABALHOS FUTUROS---------- 59

BIBILOGRAFIA ---------- 60

APÊNDICE I---------- 63

TABELAS DE DADOS E RESULTADOS

APÊNDICE II---------- 71

PROGRAMAS DESENVOLVIDOS

INTRODUÇÃO

Modelos chuva-vazão distribuídos utilizam-se de procedimentos para determinar a variação espacial das características fisiográficas que influenciam no regime hidrológico de uma bacia. Algumas dessas características são determinadas a partir do relevo. Este por sua vez , é representado por modelos digitais de terreno (MDT) ou modelos numéricos de terreno (MNT).

Modelos digitais de terreno se constituem de uma malha regular ou não onde são conhecidas altitudes e métodos de interpolação que permitam refinar essa malha e calcular por aproximação a altitude em pontos não medidos.

O grau de precisão de um modelo digital de terreno está diretamente ligado ao número da amostra de pontos coletados, sendo portanto muito dependente de uma grande base de dados.

No processo de refinamento da malha, a utilização de métodos de interpolação contribui para suavização irreal do relevo.

O objetivo deste trabalho é desenvolver um modelo digital de terreno que represente as variações do relevo independente do grau de refinamento da malha e que seja capaz de representar as variações do relevo em qualquer escala.

Para atingir este objetivo é implementado no presente trabalho um algorítmo que utiliza teoria dos fractais. A técnica fractal foi escolhida por ser capaz de gerar estruturas complexas a partir de pequena base de dados, e de reproduzir padrões de variação em qualquer escala.

É implementado também um modelo chuva-vazão distribuído no qual é utilizado o modelo digital de terreno desenvolvido neste trabalho.

A tese está dividida em cinco (5) capítulos.

No capítulo I é apresentada uma revisão bibilográfica objetivando fornecer algumas definições e os arcabouços teóricos das metodologias que serão utilizadas

no trabalho.

No capítulo II é apresentada a metodologia da interpolação fractal, de forma detalhada e com exemplo, uma vez que tal processo não é de uso comum entre os hidrólogos.

No capítulo III é apresentado o modelo chuva-vazão implementado no presente trabalho com a descrição dos procedimentos utilizados na modelagem dos processos físicos.

No capítulo IV são apresentados os resultados da aplicação do modelo a uma bacia hidrográfica.

No capítulo V são apresentadas as conclusões e sugestões desta dissertação.

CAPÍTULO I

REVISÃO BIBLIOGRÁFICA E FUNDAMENTOS TEÓRICOS

I.1 - INTRODUÇÃO.

Os modelos chuva-vazão são uma das ferramentas utilizadas pela Hidrologia na compreensão do comportamento do ciclo hidrológico. Baseiam-se em equações matemáticas, sejam conceituais ou empíricas, que representam os fenômenos físicos. Utilizam-se de dados coletados na natureza; sejam dados hidrometeorológicos, tais como chuva, evaporação, etc; sejam dados fisiográficos, tais como características do solo, distâncias, relevo, etc; para apresentar como resultado a vazão num determinado ponto de controle de uma bacia hidrográfica.

Foram desenvolvidos pela necessidade de conhecimento dos fenômenos hidrológicos, conhecimento que é utilizado no planejamento do uso dos recursos hídricos e em projetos hidráulicos.

Os modelos chuva-vazão atualmente mais utilizados são os chamados modelos concentrados. Estes tratam a bacia hidrográfica como se fosse uma região homogênea, ou seja, as características responsáveis pelo regime hidrológico são consideradas iguais para toda a área da bacia.

Essa simplificação acarreta incertezas quanto à representação física dos parâmetros e traz dificuldades na simulação de efeitos que decorram de mudanças do uso do solo, desmatamentos, urbanização e outros (CANEDO et al, 1989).

A necessidade de previsão de mudanças no regime hidrológico de uma região face aos múltiplos usos do solo e rápida urbanização tem instigado pesquisadores ao aprimoramento dos modelos chuva-vazão. Assim, o

desenvolvimento dos modelos chuva-vazão distribuídos ' conseqüência natural.

Basicamente, modelos chuva-vazão distribuídos procuram levar em conta a variabilidade das características e condições de uma bacia na simulação de seu regime hidrológico, tendo assim a vantagem de permitir uma melhor compreensão dos processos geofísicos. Entretanto, a aplicação desse modelos requer informações detalhadas sobre as características da bacia, que devem ser disponíveis com alta resolução no espaço, e dados de precipitação que também precisam ser disponíveis com alta resolução tanto no espaço como no tempo (NEUMANN, 1990).

O desenvolvimento da tecnologia de satélite, com conseqüente incremento das técnicas de sensoriamento remoto, e o aperfeiçoamento dos computadores muito tem contribuído para superar essas dificuldades (FETT, 1990), (BEVEN,1985).

I.2 - MODELOS CHUVA-VAZÃO DISTRIBUÍDOS

I.2.1 - Estrutura dos Modelos Distribuídos.

Modelos chuva-vazão distribuídos subdividem a área da bacia em diversos elementos de área. Estes podem ser elementos de uma malha regular ou não, sub-bacias ou regiões consideradas de regime hidrológico homogêneo.

Para cada elemento, as componentes hidrológicas tais como infiltração, evapotranspiração, escoamento superficial e subsuperficial, etc, são calculadas quantificando-se as leis físicas que governam os processos ou por parâmetros ajustáveis. As variações dos parâmetros e a variabilidade dos processos físicos não são considerados dentro de cada elemento, assim para uma aproximação mais precisa deve-se considerar as dimensões do elemento tal que ele possa ser considerado homogêneo (FETT, 1990). Neste sentido cada elemento pode ser visto como um modelo concentrado drenando para outro elemento e assim sucessivamente at' o canal ou até o ponto de controle como mostrado na figura 1.1.

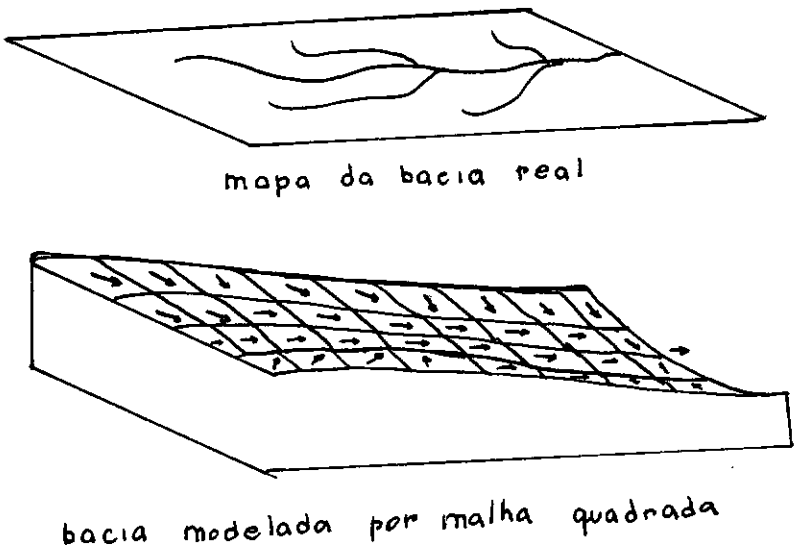


Figura 1.1 - Esquema de modelo distribuído chuva-vazão.

Os modelos chuva-vazão distribuídos são constituídos basicamente de:

a) Aquisição de dados:

Uma vez que os modelos distribuídos pretendem representar fisicamente os processos com toda sua variabilidade espacial, o volume de dados necessários é' muito superior à' quantidade de dados utilizados pelos modelos concentrados. Esses dados podem ser obtidos de mapas topográficos, aerofotos, sensoriamento remoto, radar meteorológico, pluviógrafos, pluviômetros, etc.

b) Transformação dos dados:

Alguns dados necessários ao modelo distribuído podem ser utilizados in natura necessitando apenas de procedimento para distribuí-los pela bacia. Outros dados necessitam ser manipulados de forma a se transformarem em

parâmetros geomorfológicos que serão utilizados na simulação dos processos físicos (FETT, 1990).

Os parâmetros que dependem da topografia são calculados a partir de dados de elevação conseguidos através de modelos digitais de terreno (MDT). Esses parâmetros definem as características de cada elemento da malha do modelo distribuído tais como declividade, radiação solar, direção de escoamento, etc. Podem também ser utilzados para definir a rede de drenagem e os divisores de sub-bacias (TRIBE, 1992).

c) Simulação dos processos:

Os modelos distribuídos fazem uso de equações que descrevem os processos hidrológicos de uma bacia. Essas equações podem ser conceituais como o caso das equações de quantidade de movimento, conservação de massa e energia, ou equações empíricas.

I.2.2 - Modelos Digitais de Terreno (MDT):

A topografia exerce considerável influência sobre os processos hidrológicos. Essa influência se faz sentir na infiltração, regiões planas estão mais sujeitas à infiltração que regiões íngremes; na evapotranspiração, quer seja através da vegetação quer seja através da ensolação; no regime de chuvas, através do efeito de barreira; e principalmente no escoamento, pela declividade e atrito causado pelas irregularidades do terreno. A topografia de uma região também determina os caminhos dos rios, as planícies de inundação, a velocidade da onda de cheia, etc.

Os modelos digitais de terreno são utilizados para o armazenamento dos dados de elevação que serão usados na determinação dos atributos fisiográficos que dão origem aos parâmetros a serem utilizados nas equações que simulam os processos físicos.

Modelos digitais de terreno consistem de um conjunto de dados que representam a variação espacial da

elevação para uma determinada escala. Esses dados são pontos discretos que podem ser a altitude em nós de uma malha regular ou irregular, ou altitude mêdia em cada elemento dessa malha.

As estruturas de malha mais utilizadas são:

a) Malha regular: pode ser usada estrutura triângular, retângular ou quadrada onde mais comumente as altitudes são correspondentes aos nós da malha (TORBOTON et al, 1991).

b) Malha irregular de triângulos: nessa estrutura cada elemento da malha é um plano definido pelas coordenadas de três vértices adjacentes. As arestas desses triângulos podem corresponder a vales, cumes ou mudanças bruscas na elevação (PALACIOS-VELES et al,1992).

c) Método baseado em isolinhas: essa estrutura é definida por polígonos irregulares contornados por curvas de nível adjacentes delimitadas. (MOORE et al, 1990).

A estrutura de malha regular é a mais utilizada pela facilidade de implementação e eficiência (COLLINS et al, 1981). Entretanto, curvaturas súbitas na superfície podem ser perdidas se localizadas dentro de uma unidade da malha. Sendo assim, a dimensão dos elementos da malha afeta os resultados obtidos (MOORE et al,1991).

Para um maior refinamento da malha ou para a distribuição regular de pontos de altitude irregularmente espaçados utiliza-se métodos de interpolação.

A precisão de um modelo digital de terreno é altamente dependente da quantidade de dados de altitude coletados (DONKER, 1992). Quanto maior a resolução mais dados serão necessários. As técnicas de interpolação também influem na precisão dos resultados. Técnicas de interpolação lineares, que são as mais utilizadas em MDT aplicadas à hidrologia, contribuem para a suavização irreal do relevo (DONKER, 1992).

A partir de MDT pode-se derivar parâmetros a serem utilizados nos modelos chuva-vazão. Estes parâmetros são declividade das quadrículas, azimute, área de drenagem e muitos outros (MOORE et al, 1991). A rede de drenagem de uma bacia hidrográfica também pode ser extraída automaticamente de MDT. Entretanto, os rios assim gerados

tendem a ser paralelos devido à estrutura dos modelos digitais de terreno e às simplificações necessárias para a determinação das direções de escoamento das quadrículas (TRIBE, 1992).

I.3 - TEORIA DOS FRACTAIS

I.3.1 - Introdução.

Até recentemente a geometria Euclidiana com suas formas simples era a principal ferramenta dos cientistas para descrever formas da natureza. Com o surgimento da geometria fractal, uma nova possibilidade de representar estas formas tem se apresentado. Esta geometria permite, e em alguns casos explica, a rugosidade e fragmentação de diversos aspectos da natureza.

MANDELBROT (1982) aponta a possibilidade de que sejam desenvolvidas novas descrições matemáticas para alguns dos mais complicados e irregulares fenômenos que nos cercam.

Uma das características que os pesquisadores tem encontrado em alguns fenômenos é a invariança através da escala, ou seja, se este fenômeno é observado em diferentes escalas pode-se encontrar os mesmos elementos fundamentais que caracterizam o fenômeno (JÜRGENS et al, 1990), (LOVEJOY, et al, 1985), (THEILER, 1990), (TORBOTON et al, 1988). Este padrão repetitivo ' definido como a dimensão fractal do fenômeno (MENDELBROT, 1977, 1982).

Alguns pesquisadores tem discutido se redes de drenagem possuem dimensão fractal associada. Entre estes pesquisadores estão TORBOTON et al (1988) que demostrou que a rede de rios de uma bacia é' um objeto fractal e calculou sua dimensão fractal, LA BARBERA et al (1989) que deduziu uma expressão para dimensão fractal da rede de drenagem em termos das proporções de Horton, NIKORA (1991) que apresentou relações entre sinuosidade, largura do canal, largura do vale e dimensão fractal.

MANDELBROT (1982) sugere que o relevo possui as características de invariança através da escala. A

rugosidade ou irregularidade do relevo esté expressa em suas formas montanhosas e até em seus grãos de areia. Modelos que se utilizam de algorítmos fractais seriam capazes de reproduzir esta característica do relevo.

LOVEJOY et al (1985) apresentou um modelo de modelagem de regime de precipitação que utiliza algorítmo fractal.

Fractais são expressos não em forma geométrica mas em algorítmos juntamente com procedimentos matemáticos. Esses algorítmos se transformam em formas geométricas com a ajuda do computador. De algorítmos simples pode-se produzir formas de geometria complexa (JÜRGENS et al, 1990);

Existem vários algorítmos que podem ser utilizados para simular geometria fractal. Estes algorítmos podem ser lineares ou não-lineares, randômicos ou determinísticos.

Não faz parte do escopo deste trabalho a análise comparativa dos diversos algorítmos existentes, desta forma será apresentado apenas o algorítmo escolhido para o desenvolvimento do mesmo.

Uma das maneiras mais utilizadas para se gerar fractais ' dada sob a forma de um sistema de funções iterativas (BARNSLEY, 1988) que é' um algorítmo determinístico.

Sistema de funções iterativas é um conjunto de transformações lineares ou não-lineares que, em conjunto com um sistema dinâmico, gera um objeto fractal. O termo objeto é utilizado neste contexto como qualquer forma que se queira reproduzir, seja um gráfico no $\mathbb{R}^2$, uma superfície no $\mathbb{R}^3$, ou a imagem de um objeto qualquer.

Cada transformação desse sistema representa uma parte ou um aspecto do objeto a ser simulado. O objeto fractal será simulado através da aplicação das transformações que o definem sobre um objeto inicial. A imagem de uma folha de samambaia, por exemplo, pode ser reproduzida por esse método a partir da imagem de um triângulo. O sistema para reproduzir esta folha é formado de cinco transformações com cinco parâmetros cada. As transformações são aplicadas ao triângulo, contraindo-o,

distorcendo-o e transladando-o, produzindo uma nova imagem à qual serão aplicadas novamente as transformações e assim sucessivamente até obter o nível de precisão desejado. A imagem da folha será a reunião de todas essas imagens.

Tal algorítmo parte do princípio de que se possa "dividir" o objeto em partes que serão representadas pelas transformações. A dificuldade reside na obtenção da menor família de transformações $\{f_1, f_2, f_3, \ldots . f_1\}$ necessárias para definir um objeto com desejável precisão (JÜRJENS et al, 1990).

BARNSLEY (1988) apresenta um método para obtenção dessas transformações que se aplica a objetos simples, mas essa dificuldade ainda não foi superada e tem sido motivo de continuado estudo.

I.3.2 - Sistemas de funções iterativas.

I.3.2.1 - Definição:

Um sistema de funções iterativas é um conjunto de transformações $\{ X; w_1, i = 1,2,3....N \}$ onde X é' espaço métrico (X,d) sendo d a métrica associada a esse espaço métrico e w_1 são contrações neste espaço métrico, cada transformação tem fator de contratividade s_1. O fator de contratividade de todo o sistema é dado por

$S = m_{SX} \{ s_1, i = 1,2,3...N \}$.

Se as transformações w_1 são contrações então esse sistema possui um atrator no caso linear. Para que as transformações sejam contrações, os autovalores da matriz formada por seus parâmetros devem pertencer ao intervalo (-1,1).

A métrica associada a esse espaço métrico é equivalente à métrica Hausdorff (BARNSLEY, 1988).

Aplicando-se a transformação w_1 a um ponto inicial x obtemos uma seqüência infinita de pontos :

$\{x, w_1(x), w_1(w_1(x)), w_1(w_1(w_1(x))), \ldots\ldots\}$

Essa seqüência convergirá, em um número infinito de pontos, para um ponto fixo se cada transformação w_1 é' uma contração.

Seja um sistema de três transformações { X; w_1, w_2, w_3} onde cada w_i é uma contração no espaço métrico (X,d). Cada w_i posui um ponto fixo. Podemos iterar dinamicamente esse sistema a partir de um ponto inicial *x* e obter uma seqüência infinita de pontos :

$\{x, w_2(x), w_1(w_2(x)), w_1(w_1(w_2(x))), w_3(w_1(w_1(w_2(x)))),...\}$

onde não importa a ordem em que as transformações são aplicadas. Essa seqüência convergirá,em um número infinito de pontos, para o atrator que é o ponto fixo desse sistema de transformações.

Se o ponto *x* inicial pertence ao atrator então toda a seqüência pertencerá ao atrator. Se *x* não pertence ao atrator então serão necessárias iterações até que a seqüência convirja para o atrator.

Sendo o atrator, por exemplo, uma função em $\mathbb{R}^2$, ela será descrita pelos pontos da seqüência que já tenham convergido para o atrator, para uma dada aproximação. Quanto maior o número de pontos da seqüência mais nitidamente será descrita a função.

Este atrator não pode ser expresso em forma de função analítica mas apenas como um conjunto de pontos gerados pela iteração sucessiva do sistema que o representa.

Qualquer ponto pertencente a essa seqüência será o resultado da aplicação das transformações a um ponto inicial qualquer um determinado número de vezes. Para calcular especificamente este ponto, não temos como determinar quantas iterações foram necessárias, nem qual a ordem em que foram aplicadas para produzí-lo. Para que seja possível determinar qualquer ponto pertencente ao atrator é utilizado um sistema dinâmico de "shift" randômico associado a esse sistema de funções iterativas.

I.3.2.2 - Sistema dinâmico:

Um sistema dinâmico é uma transformação invertível $f:X \to X$ sobre um espaço métrico (X,d) onde d é' a métrica associada a esse espaço métrico. A órbita do ponto $x \in X$ é a seqüência $\{ f^n(x) \}_{n=0}^{\infty}$ onde:

$$\{f^n(x)\}_{n=0}^{\infty}=\{x,f^1(x),f^2(x),f^3(x),.....f^n(x)\}$$

$$\begin{aligned} f^0(x) &= x \\ f^1(x) &= f(x) \\ f^2(x) &= f(f^1(x)) \\ f^3(x) &= f(f^2(x)) \\ &\vdots \\ f^n(x) &= f(f^{n-1}(x)) \end{aligned}$$

e a órbita inversa do ponto x é a seqüência

$$\{f^{-n}(x)\}_{n=0}^{\infty} = \{ x, f^{-1}(x), f^{-2}(x), f^{-3}(x),....f^{-n}(x) \}$$

onde f^{-1} ' a transformação inversa de f e

$$\begin{aligned} f^{-2}(x) &= f^{-1}(f^{-1}(x)) \\ f^{-3}(x) &= f^{-1}(f^{-2}(x)) \\ &\vdots \\ f^{-n}(x) &= f^{-1}(f^{-n+1}(x)) \end{aligned}$$

I.3.2.3 - Sistema dinâmico de "shift" randômico:

Dada uma seqüência $A = \{f^n(x)\}_{n=0}^{\infty}$, uma transformação "shift" sobre A é a transformação $S : A \to A$ definida por :

$S(a) = f^{-1}(a)$ para $a \in A$

O sistema { A; S } ' chamado sistema dinâmico de "shift". Para associarmos esse sistema a um sistema de funções iterativas consideramos que A é o atrator desse sistema e :

$S(a) = w_n^{-1}(a)$ para $a \in w_n(A)$

onde w_n é transformação invertível do sistema e $w_n(A)$ é a parte do atrator definida pela w_n.

A união dos $w_n(A)$ formam o atrator definido pelo sistema

{ X; w_n, n = 1,2,.....N }.

$A = w_1(A) \cup w_2(A) \cup w_3(A) \cup .....\cup w_n(A)$

Se $w_i(A) \cap w_j(A) = 0$ pelo menos para um $i \neq j$ $i = 1,2,....N$, e $j = 1,2,....N$, entro o atrator ' totalmente desconectado (não ' contínuo).

Seja o sistema de funções iterativas $\{ X; w_1, w_2\}$ onde w_1 e w_2 sro transformações invertíveis e A é' o seu atrator totalmente desconectado.

A seqüência de pontos $\{ x_i \}_{i=0}^{\infty}$ em A é chamada de órbita do sistema dinâmico de "shift" associado ao sistema de funções iterativas se :

$$x_{i-1} = \begin{cases} w_1^{-1}(x_i) & \text{quando } x_i \in w_1(A) \\ w_2^{-1}(x_i) & \text{quando } x_i \in w_2(A) \end{cases}$$

Se $w_i(A) \cap w_j(A) \neq 0$ para qualquer i, j, $i \neq j$; $i,j = 1,2..N$ então o atrator é contínuo.

Seja o sistema $\{ X; w_1, w_2\}$ onde w_1 e w_2 são transformações invertíveis e A seu atrator contínuo.

A seqüência de pontos $\{ x_i \}_{i=0}^{\infty}$' chamada órbita do sistema dinâmico de "shift" randômico associado ao sistema de funções iterativas se:

$$x_{i-1} = \begin{cases} w_1^{-1}(x_i) & \text{quando } x_i \in w_1(A) \text{ e } x_i \notin w_1(A) \cap w_2(A) \\ w_2^{-1}(x) & \text{quando } x_i \in w_2(A) \text{ e } x_i \notin w_1(A) \cap w_2(A) \\ w_1^{-1}(x_i) \text{ ou } w^{-1}{}_2(x_i) & \text{quando } x_i \in w_1(A) \cap w_2(A) \end{cases}$$

Com a órbita do sistema dinâmico de "shift" randômico podemos reconstituir a órbita de qualquer ponto do atrator .

I.3.3 - Interpolação fractal utilizando sistema de funções iterativas :

Seja um conjunto de dados coletados de um fenômeno qualquer na forma $\{ (x_j, F_j); j = 0,1,2,...N \}$. O objetivo dos métodos de interpolação é ajustar uma função ou funções a esses dados que possam descrever com uma determinada precisão o fenômeno.

Funções elementares tais como polinomiais, seno, cosseno, etc, são a base dos tradicionais métodos de

análise de dados experimentais.

Interpolação fractal apresenta um novo método de análise de dados.

Esta técnica de interpolação permite simular o comportamento de fenômenos cuja variação seja muito acentuada, para os quais os métodos de interpolação tradicionais não são satisfatórios. Podemos também fazer com que a dimensão fractal do fenômeno, presente nos dados seja reproduzida pela interpolação.

Seja o conjunto de dados $\{ (x_j, F_j), j=1.2,...n \}$. Podemos construir um sistema de funções iterativas em $\mathbb{R}^2$, tal que, seu atrator, denotado por A, é gráfico de uma função contínua de $\mathbb{R}$ em $\mathbb{R}$ que interpola os dados. Esse atrator passa por todos os pontos de interpolação. O sistema é da forma $\{ \mathbb{R}^2, w_i, i = 1,2,....N \}$ onde as w_i são transformações afim da forma :

$$w_i \begin{pmatrix} x \\ F(x) \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ F(x) \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}$$

onde a_i, c_i, d_i, e_i, f_i são parâmetros das transformações.

Do mesmo modo podemos interpolar um conjunto de dados em $\mathbb{R}^3$.

Seja um conjunto de dados $\{ (x_j, y_j, F(x_j, y_j)), j = 1,2,3....N \}$. Podemos construir um sistema de funções iterativas tal que seu atrator seja gráfico de uma função contínua de $\mathbb{R}^2$ em $\mathbb{R}$. Este atrator é uma superfície que interpola esses pontos. O sistema é da forma $\{ \mathbb{R}^3; w_i, i = 1,2,...N \}$ onde w_i são transformações afim da forma:

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ e_i & f_i & g_i \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} u_i \\ v_i \\ t_i \end{pmatrix}$$

onde $a_i, b_i, c_i, d_i, e_i, f_i, g_i, u_i, v_i, t_i$ são os parâmetros das transformações e $z = F(x,y)$.

Utilizando-se um sistema dinâmico de "shift" randômico (ver secão 3.2.3) podemos calcular qualquer ponto pertencente à essa superfície, ou seja, ao atrator A.

I.3.3.1 - Cálculo dos parâmetros do sistema de funções iterativas.

Para determinar os parâmetros a,b,c,d,e,f,g,u,v,t relativos a cada transformação utilizamos os pontos de interpolação da seguinte forma :

-Faz-se uma triângulaçfo usando os pontos $(x_j, y_j, F(x_j, y_j))$, figura 1.2a e 1.2b.

- Cada um destes triângulos é uma unidade de interpolação e tem uma transformação w_i relacionada a ela.

- Os triângulos devem estar totalmente inseridos num triângulo que chamaremos o triângulo domínio da função. Esse triângulo domínio é o limite do objeto fractal, ou seja, a superfície só será determinada dentro desse triângulo domínio. O triângulo domínio é determinado por três pontos de interpolação de maneira que todos os outros pontos de interpolação pertençam a ele.

- a superfície é contínua se todo triângulo pertencente ao triângulo domínio possuir uma transformação associada, (BARNSLEY,1988).

- Normaliza-se o domínio, ou seja, transporta-se todos os dados para o domínio [0,1]. Isto é necessário para que as transformações sejam contrações e possamos obter um atrator desse sistema.

Com essa triângulação dos pontos de interpolação obtemos uma superfície aproximada por planos.

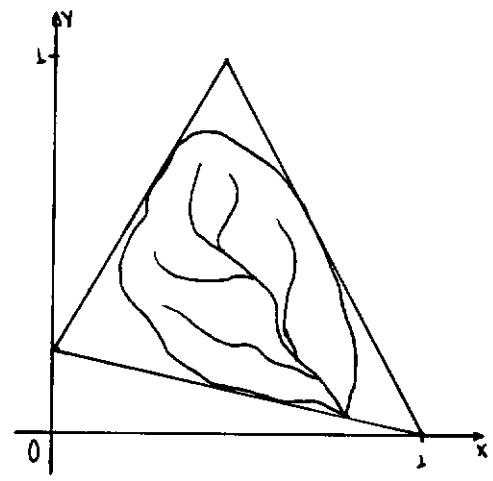


Figura 1.2a - Bacia inserida no triângulo domínio.

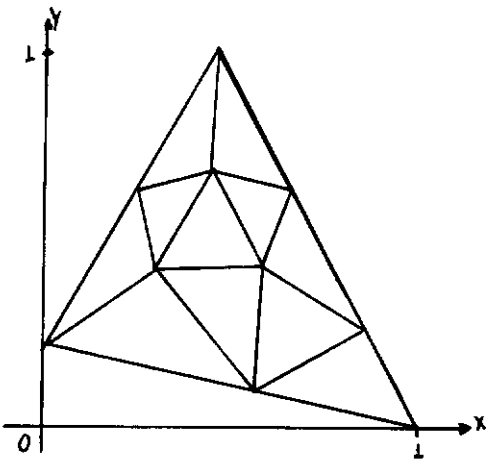


Figura 1.2b - Projeção da triângulação em z = 0.

Para o conjunto de 10 pontos de interpolação utilizados na figura 1.2a obtemos doze triângulos. O sistema de funções iterativas para essa superfície é $\{\mathbb{R}^3; w_1, w_2, w_3 \ldots . w_{11}\}$.

Cada w_i ' a transformação que contrai, distorce e translada o prisma do triângulo domínio no prisma da unidade de interpolação i. A altura do prisma representa a contração em F(x,y), ver figura 1.3a.

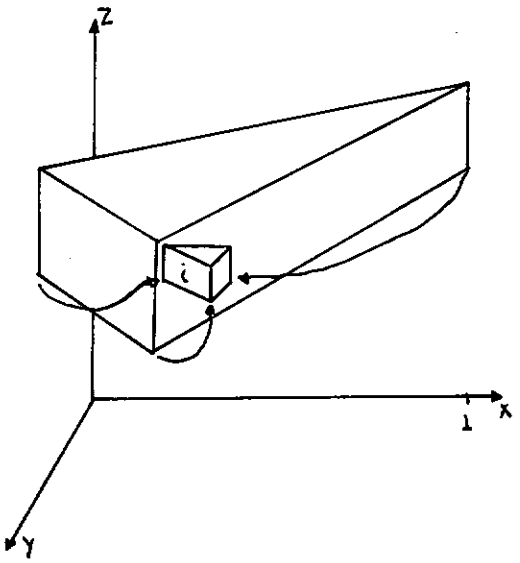


Figura 1.3a - Prisma do domínio transformado no prisma i

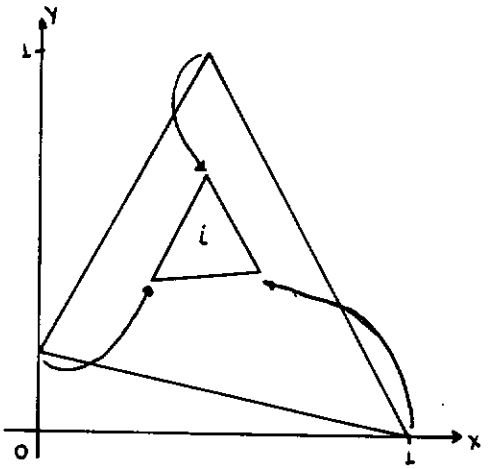


Figura 1.3b - Projeção em z = 0.

Assim cada vértice do triângulo i é o resultado da aplicação da w_i a um vértice do domínio. Designando os vértices do triângulo i por (x_k, y_k, z_k) e os vértices do triângulo domínio por $(\alpha_k, \beta_k, \lambda_k)$ k = 1,2,3, temos:

$$w_i \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \qquad w_i \begin{pmatrix} \alpha_2 \\ \beta_2 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \qquad w_i \begin{pmatrix} \alpha_3 \\ \beta_3 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$$

Temos então três sistemas da forma :

$$\begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ e_i & f_i & g_i \end{pmatrix} \cdot \begin{pmatrix} \alpha_k \\ \beta_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} u_i \\ v_i \\ t_i \end{pmatrix} \qquad k = 1,2,3$$

Pode-se então calcular os valores dos parâmetros dado que as coordenadas dos vértices são conhecidas.

Escrevendo-se as coordenadas x e y dos três vértices do triângulo 1 temos dois sistemas de três equações, onde as incógnitas do primeiro sistema são a_i, b_i, u_i; e as incógnitas do segundo sistema são c_i, d_i, v_i :

$$\begin{cases} a_i\alpha_1 + b_i\beta_1 + u_i = x_1 \\ a_i\alpha_2 + b_i\beta_2 + u_i = x_2 \\ a_i\alpha_3 + b_i\beta_3 + u_i = x_3 \end{cases}$$

$$\begin{cases} c_i\alpha_1 + d_i\beta_1 + v_i = y_1 \\ c_i\alpha_2 + d_i\beta_2 + v_i = y_2 \\ c_i\alpha_3 + d_i\beta_3 + v_i = y_3 \end{cases}$$

Escrevendo-se os valores de z dos três vértices do triângulo 1 temos um sistema de três equações onde as quatro incógnitas são e_i, f_i, g_i, t_i:

$$\begin{cases} e_i\alpha_1 + f_i\beta_1 + g_i\lambda_1 + t_i = z_1 \\ e_i\alpha_2 + f_i\beta_2 + g_i\lambda_2 + t_i = z_2 \\ e_i\alpha_3 + f_i\beta_3 + g_i\lambda_3 + t_i = z_3 \end{cases}$$

Os parâmetros g_i representam a contração em F(x,y) ou mais precisamente o fator de contração da altura do prisma do domínio na altura do prisma do triângulo i. Podemos considerá-los parâmetros livres e arbitrar valores para eles; com isso o sistema acima se reduz a um sistema de três incógnitas (e_i, f_i, t_i).

Os parâmetros $a_i, b_i, c_i, d_i, u_i, v_i$ dependem somente da triângulação e representam :

a_i - a contração em x do triângulo i.

b_i - a contração em y do triângulo i.

c_i, d_i cisalhamento do triângulo i.

u_i, v_i translação do triângulo i.

Os parâmetros livres g_i definem a rugosidade da

superfície gerada pelo sistema iterativo de funções. Quanto mais próximo de 1 (um), mais irregular será a superfície. Se g_i for igual a zero a superfície será exatamente igual à superfície inicial aproximada por planos.

Os parâmetros e_i, f_i, t_i dependem da triângulação e dos parâmetros g_i.

O fator de contratividade s_i da transformação w_i' dado por:

$$s_i = g_i \left(\det \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \right)$$

Repetindo-se este procedimento para todos os triângulos temos um conjunto de matrizes de parâmetros .

Os parâmetros g_i são livres no sentido que, para dois conjuntos de parâmetros g_i diferentes tem-se duas superfícies diferentes para uma mesma triângulação. Esses parâmetros podem ser manipulados com o objetivo de se conseguir o melhor ajuste dessa superfície a uma superfície real.

I.3.3.2 - Propriedades da interpolação fractal ;

O atrator definido pela interpolação através de sistema iterativo de funções possui as seguintes características:

- " gráfico de função contínua em todo domínio.
- A função passa por todos os pontos de interpolação.
- O atrator depende continuamente dos parâmetros.
- A função é diferenciável em todo domínio se $g_n < a_n$ e $g_n < d_n$ para todo n = 1,2,....N (ver seção 4, capítulo II) .
- Este método de interpolação permite variações não lineares dentro de uma unidade de interpolação, ou seja dentro dos triângulos.

CAPÍTULO II

MODELO DIGITAL DE TERRENO UTILIZANDO INTERPOLAÇÃO FRACTAL

II.1 - TRIÂNGULAÇÃO E CÁLCULO DOS PARÂMETROS.

A triângulação é feita utilizando-se os pontos de interpolação coletados de mapas topográficos que são da forma (x,y,z) onde x e y são coordenadas de um plano x,y que contém a planta da bacia e z ' é a altitude no ponto (x,y).

A bacia hidrográfica que será modelada tem que estar inserida no triângulo do domínio (figura 2.1a, 2.1b). Deve-se coletar pontos dentro de todo domínio, inclusive fora da bacia. Normaliza-se esses pontos trazendo-os para o domínio [0,1]. A triângulação é feita conforme explicado na seção 3.3.1 ,capítulo I .

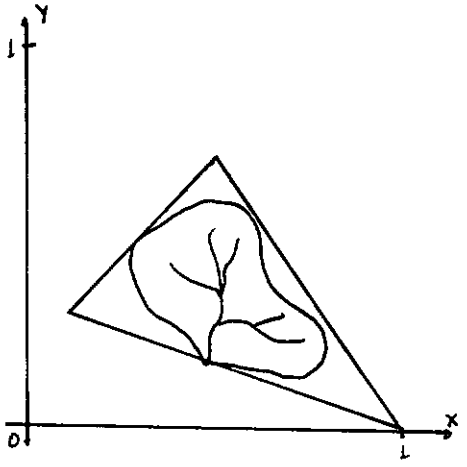


Figura 2.1a - Bacia inserida no triângulo e no domínio [0,1]

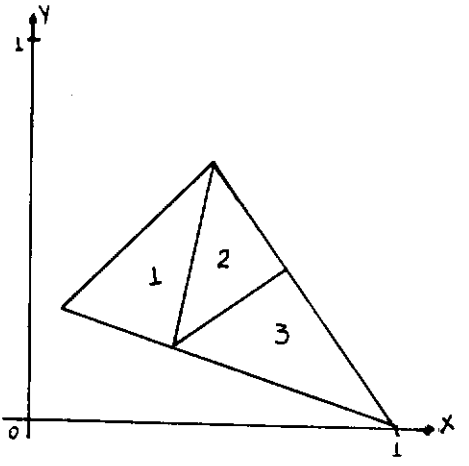


Figura 2.1b - Projeção da triângulação no plano z = 0 .

Para essa triângulação obtemos um sistema de funções iterativas $\{ \mathbb{R}^3; w_1, w_2, w_3 \}$ onde cada w_i é da forma:

$$w_i = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ e_i & f_i & g_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} u_i \\ v_i \\ t_i \end{pmatrix}$$

Os parâmetros de cada w_i são calculados utilizando-se os pontos de interpolação que são vértices do triângulo domínio e os pontos de interpolação que são vértices do triângulo i conforme procedimento descrito na seção 3.3.1 , capítulo I.

Neste exemplo usamos uma bacia hidrográfica fictícia, cujos pontos de interpolação (coletados k pontos nos vértices, arestas e interior do triângulo) são:

	pontos de interpolação		
k	x_k	y_k	z_k
1	0.1	0.3	0.08
2	0.4	0.2	0.03
3	0.5	0.7	0.06
4	0.7	0.42	0.1
5	1.0	0.0	0.07

Arbitrando os parâmetros g_i em (0.55,0.35,0.6), obtemos os seguintes valores para os demais parâmetros:

i	a_i	b_i	u_i	c_i	d_i	v_i
1	-0.104	-0.646	0.604	0.479	-0.229	0.221
2	0.042	0.458	0.358	-0.592	-0.108	0.792
3	0.063	-0.812	0.937	-0.487	-0.062	0.487

i	e_i	f_i	t_i
1	0.068	0.085	-0.046
2	0.007	0.110	-0.002
3	-0.056	-0.089	0.084

Os parâmetros g_i assim como os demais parâmetros devem pertencer ao intervalo (-1, 1) para que as transformações sejam contrações.

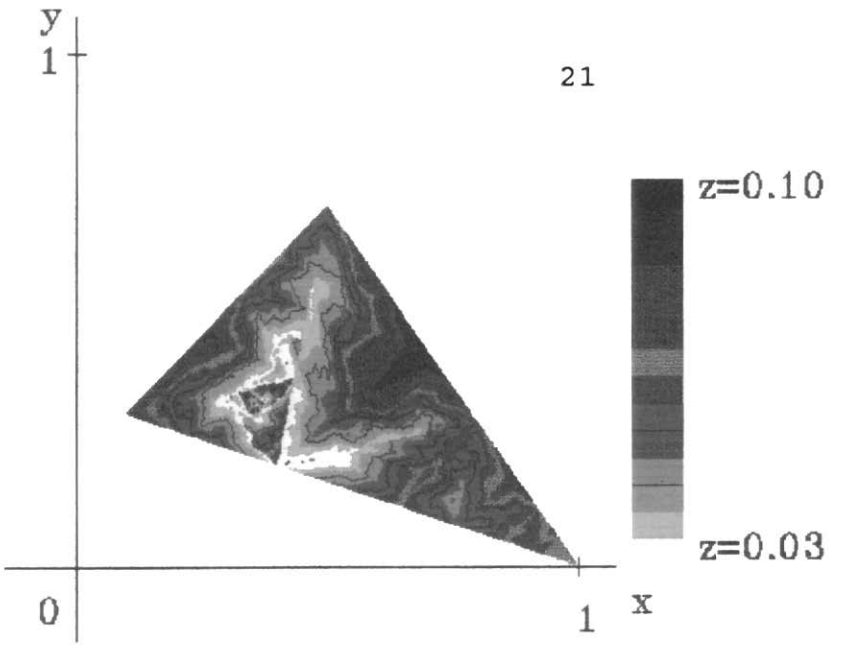


Figura 2.2 - Projeção da superfície no plano z = 0. As altitudes estão representadas em uma escala de cores.

A superfície associada a esse sistema iterativo de funções definido com estes parâmetros g_i é mostrada na figura 2.2.

II.2 - CÁLCULO DAS ALTITUDES A PARTIR DO SISTEMA DE FUNÇÕES ITERATIVAS

A partir do sistema de funções iterativas podemos obter qualquer ponto do atrator, utilizando sistema dinâmico de "shift" randômico.

O atrator neste caso, é uma superfície em $\mathbb{R}^3$ que passa por todos os pontos de interpolação e definida somente dentro do triângulo domínio.

Escolhendo-se qualquer ponto inicial $(x_0, y_0, z_0,)$ e aplicando qualquer das transformaçges w_i a esse ponto obtemos um outro ponto (x_1, y_1, z_1), aplicando essa ou outra transformação do sistema a esse ponto obtemos um novo ponto (x_2, y_2, z_2) e assim sucessivamente.

Essa seqüência convergirá para o atrator em um número infinito de pontos, mas a partir de um determinado

nimero de aplicações se obtem uma aproximação do atrator com grau de precisão suficiente para a presente aplicação.

O atrator é o objeto formado pela reunião dos pontos dessa seqüência a partir do ponto que corresponde a uma determinada precisão. A esse processo chamaremos iteração dinâmica. Essa seqüência é a órbita do ponto (x_0,y_0,z_0). Podemos calcular a órbita inversa desse ponto aplicando-se as inversas das transformações w_i.

Um ponto (x_k,y_k,z_k) pertencente ao atrator é' ponto da seqüência, entretanto não podemos precisar quantas iterações serão necessárias para se chegar até ele a partir de um determinado ponto inicial. Para calcular especificamente esse ponto utiliza-se sistema dinâmico de "shift" randâmico. O objetivo é calcular z_k (altitude) em determinado ponto (x_k,y_k) (latitude e longitude). Se (x_k,y_k) pertence ao triângulo domínio então o ponto (x_k,y_k,z_k) pertence ao atrator.

Calcula-se z_k reconstituindo-se a órbita inversa do ponto (x_k,y_k,z_k). Designaremos o ponto (x_k,y_k,z_k) por X_k.

O sistema dinâmico de "shift" randômico (definido na seção 3.2.2, capítulo I) garante que:

$$X_{k-1}=\begin{cases} w_i^{-1}(X_k) & \text{quando } X_k \in w_i(A) \text{ e } X_k \notin w_i(A) \cap w_j(A) \\ w_j^{-1}(X_k) & \text{quando } X_k \in w_j(A) \text{ e } X_k \notin w_i(A) \cap w_j(A) \\ w_i^{-1}(X_k) \text{ ou } w_j^{-1}(X_k) & \text{quando } X_k \in w_i(A) \cap w_j(A) \end{cases}$$

Isso significa que se o ponto X_k pertence ao triângulo j então o ponto X_{k-1} é' obtido aplicando-se a inversa da transformação w_j ao ponto X_k. Quando o ponto X_k pertence a uma das arestas do triângulo j então o ponto X_{k-1} é obtido aplicando-se a inversa da transformação w_j ou a inversa da transformação definida pelo triângulo adjacente ao triângulo j por esta aresta.

À órbita do atrator $\{ X_0,X_1,X_2,.....X_{k-1},X_k \}$ definida em $\mathbb{R}^3$ corresponde uma órbita no plano $F(x,y) = 0$ definida em $\mathbb{R}^2$ (BARNSLEY, 1988). Isso é possível devido à forma das transformações w_i:

$$\begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ e_i & f_i & g_i \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{pmatrix} + \begin{pmatrix} u_i \\ v_i \\ t_i \end{pmatrix}$$

Assim podemos reconstituir a órbita inversa do ponto (x_k, y_k, z_k) utilizando a órbita inversa do ponto (x,y) correspondente no $\mathbb{R}^2$.

$$\begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}^{-1} \cdot \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}^{-1} \cdot \begin{pmatrix} u_i \\ t_i \end{pmatrix}$$

onde:

$\begin{pmatrix} a_j & b_j \\ c_j & d_j \end{pmatrix}^{-1}$ matriz inversa da matriz $\begin{pmatrix} a_j & b_j \\ c_j & d_j \end{pmatrix}$

O índice j representa o número do triângulo ao qual o ponto (x_k, y_k) pertence. Ao obtermos o ponto (x_{k-1}, y_{k-1}) verifica-se a qual triângulo ele pertence.

Aplica-se a inversa dessa transformação ao ponto (x_{k-1}, y_{k-1}) e obtem-se o ponto (x_{k-2}, y_{k-2}) e assim sucessivamente, até um número m de iterações. Ao final de m iterações temos o valor de (x_{k-m}, y_{k-m}) e os índices das transformações utilizadas nestas iterações.

Pode-se então calcular o valor de z_k utilizando o ponto $(x_{k-m}, y_{k-m}, z_{k-m})$ como ponto inicial (x_0, y_0, z_0) onde z_{k-m} ' um valor estimado (ver seção 4.1, capítulo II). Com este ponto e os índices das transformações refaz-se a órbita do ponto $(x_{k-m}, y_{k-m}, z_{k-m})$ até o ponto (x_k, y_k, z_k)

O número de iterações m deve ser tal que possibilite z_k estar próximo do atrator com precisão suficiente para a utilização nesta aplicação, ou seja, um número tal que permita z_{k-m} convergir para o atrator (ver seção 3.1 , capítulo II).

II.3 - PRECISÃO E CONVERGÊNCIA.

II.3.1 - Número de iterações e estimativa de z_0 inicial:

Para testar a convergência utilizou-se a bacia fictícia do exemplo. Escolheu-se alguns pontos (x ,y) e calculou-se o valor de z para cada ponto conforme procedimento exposto na seção 6 deste capítulo. Foram utilizados conjuntos de parâmetros g_i diferentes, e diferentes estimativas de z_0 para o ponto inicial.

Para as estimativas de z_0 foram utilizadas a m'dia dos valores de z dos pontos de interpolação, o valor de z_0 igual a zero, e o valor de z_0 calculado através dos planos definidos pelos triângulos da interpolação do seguinte modo:

- calculado o valor de (x_{k-m}, y_{k-m}) através da órbita inversa verifica-se a que triângulo este ponto pertence.
- calcula-se z_{k-m} utilizando a equação do plano definido por este triângulo e utiliza-se este valor como z_0 .

Pelos resultados apresentados nas tabelas III, do APÊNDICE I vemos que os efeitos na convergência causados pelos valores dos parâmetros g_i são preponderantes. Quanto mais próximos de 1 (um) os valores dos parâmetros $|g_i|$ mais irregular é a superfície, necessitando portanto de um número maior de iterações para convergir.

A estimativa de z_0 não é relevante quando a variação entre as altitudes dos pontos de interpolação não é significativa.

O número de iterações necessário para a convergência ficou em torno de 10 a 15 iterações. Para a aplicação neste estudo o grau de precisão obtido com este número de iterações é suficiente.

II.3.2 - Erros associados ao sistema dinâmico de "shift" randômico:

No cálculo da órbita de um ponto os erros relativos são pequenos já que as transformações são

contrações (BARNSLEY,1988). Os erros diminuem com o processo iterativo.

No calculo da órbita inversa de um ponto é utilizado o inverso das transformações. O fator de contração dessas transformações torna-se um fator de expansão. A precisão na determinaçãã das inversas das transformações w vai definir a precisão destes fatores de expansão (BARNSLEY, 1988).

O erro associado à imprecisão do cálculo das inversas leva a órbita inversa a sair do atrator onde a função não está definida. Quanto maior o número de iterações, maior a possibilidade da órbita inversa sair do atrator, porque os erros se expandem com o processo iterativo.

A escolha do número de iterações a ser adotado deve levar em conta, além da convergência, a precisão do cálculo das inversas das transformações.

No presente trabalho foi adotado o método de Gauss com pivotiamento parcial para a determinação das inversas das matrizes das transformações.

II.3.3 - Erros associados à localização de um ponto dentro da malha de triângulos:

Para a localização de um ponto dentro da malha da triângulação utilizou-se as coordenadas baricêntricas $(\lambda_1,\lambda_2,\lambda_3)$ associadas a esse ponto. Essas coordenadas são determinadas por três equações :

$$\begin{cases} x = \sum v_x . \lambda_i \\ y = \sum v_y . \lambda_i \\ \sum \lambda_i = 1 \end{cases} \qquad i = 1,2,3$$

onde (x, y) ' o ponto a ser localizado na triângulação,

$v_x = (x_1,x_2,x_3)$ ' o vetor das abscissas dos vértices do triângulo testado

$v_y = (y_1,y_2,y_3)$ ' o vetor das ordenadas dos vértices do triângulo testado

Se $\lambda_i \in (0,1)$ para todo i = 1,2,3 então o ponto (x,y)

pertence ao triângulo testado.
Se algum $\lambda_i = 0$ $i = 1,2,3$ então o ponto (x,y) pertence a alguma aresta do triângulo.
Se dois dos $\lambda_i = 0$ $i = 1,2,3$ então o ponto (x,y) é vértice do triângulo testado.
Se $\lambda_i \notin [0,1]$ para algum $i = 1,2,3$ então o ponto (x,y) não pertence a esse triângulo.

Devido ‘ aritmética de ponto flutuante existe erro associado à determinação de λ_i.

II.4 - DECLIVIDADE EM UM PONTO QUALQUER DA SUPERFÍCIE

Inicialmente a superfície é aproximada por planos que ligam os pontos de interpolação numa dada triângulação.

A cada triângulo está associada um plano descrito por uma equação da forma :

$$z = A_i x + B_i y + C_i$$

onde A_i ,B_i ,C_i são coeficientes da equação do plano no qual está inserido o triângulo i (GILL ET AL, 1981). Sabemos que o vetor gradiente

$$grad_i = (A_i, B_i)$$

fornece a direção de maior declividade do plano como sendo $- grad_i$.

A superfície final (atrator) é obtida através da iteração, até um determinado limite m, de um sistema iterativo de funções da forma :

$\{ X ; w_i, i= 1,2,....N \}$

onde N é o número de triângulos formados pelos pontos de interpolação.

As transformações w_is são de forma :

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ e_i & f_i & g_i \end{pmatrix} . \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} u_i \\ v_i \\ t_i \end{pmatrix}$$

Iterar um sistema de funções até um limite m significa que na primeira iteração estaremos subdividindo cada triângulo conforme a triângulação inicial (aquela

definida pelos pontos de interpolação), respeitando a proporcionalidade dos n triângulos em relação ao todo. Na segunda iteração subdividiremos os triângulos resultantes novamente conforme a triângulação inicial usando o mesmo procedimento e assim sucessivamente até m vezes.

Assim cada w_i será responsável por uma das subdivisões do triângulo i num determinado estágio das iterações.

Utilizando a bacia fictícia temos a triângulação inicial (projeção da triângulação no plano x,y) mostrada na figura 2.3a.

Para esta triângulação temos três w_i's : w_1, w_2, w_3. Quando iterado uma vez o domínio terá a configuração mostrada na figura 2.3b.

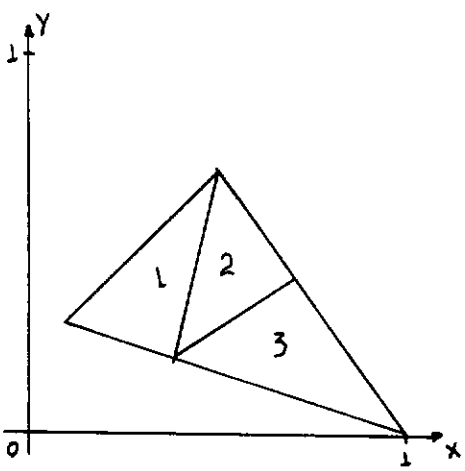


Figura 2.3a - Triângulação original do domínio.

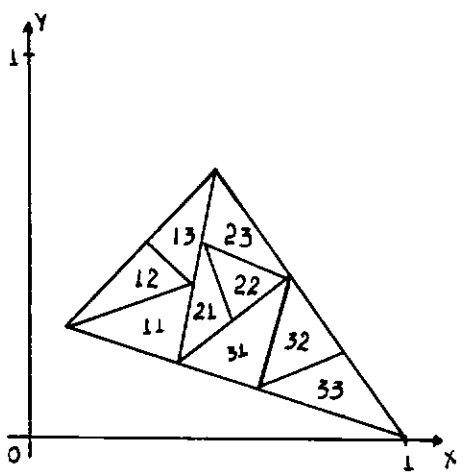


Figura 2.3b - Configuração da triângulção após uma iteração.

onde o triângulo 11 é resultado da w1 aplicada ao triângulo 1

triângulo 12 é resultado da w2 aplicada ao triângulo 1

triângulo 13 é resultado da w3 aplicada ao triângulo 1

triângulo 22 é resultado da w2 aplicada ao triângulo 2

triângulo 23 é resultado da w3 aplicada ao

triângulo 2

triângulo 31 é resultado da w_1 aplicada ao triângulo 3

triângulo 21 é resultado da w_1 aplicada ao triângulo 2

e assim por diante.

Assim as variações das coordenadas de um ponto qualquer pertencente ao triângulo 12 sofrem alterações conforme os parâmetros da w_2.

$$\begin{pmatrix} dx_{12} \\ dy_{12} \\ dz_{12} \end{pmatrix} = \begin{pmatrix} a_2 & b_2 & 0 \\ c_2 & d_2 & 0 \\ e_2 & f_2 & g_2 \end{pmatrix} \begin{pmatrix} dx_1 \\ dy_1 \\ dz_1 \end{pmatrix}$$

$$dx_{12} = a_2 dx_1 + b_2 dy_1$$

$$dy_{12} = c_2 dx_1 + d_2 dy_1$$

$$dz_{12} = e_2 dx_1 + f_2 dy_1 + g_2 dz_1$$

onde dx_{12}, dy_{12}, dz_{12} são as variações das coordenadas do triângulo 12;

dx_1, dy_1, dz_1 são as variações das coordenadas do triângulo 1.

então a derivada parcial de z em relação a x, para o triângulo 12, é :

$$\left(\frac{\partial z}{\partial x} \right)_{12} = \left. \frac{e_2 dx_1 + f_2 dy_1 + g_2 dz_1}{a_2 dx_1 + b_2 dy_1} \right|_{dy=0}$$

$$= \frac{e_2}{a_2} + \frac{g_2}{a_2} \left(\frac{\partial z}{\partial x} \right)_1$$

e a derivada de z em relação a y, para o triângulo 12, é:

$$\left(\frac{\partial z}{\partial y} \right)_{12} = \left. \frac{e_2 dx_1 + f_2 dy_1 + g_2 dz_1}{c_2 dx_1 + d_2 dy_1} \right|_{dx=0}$$

$$= \frac{f_2}{d_2} + \frac{g_2}{d_2} \left(\frac{\partial z}{\partial y} \right)_1$$

Como a cada triangulo i esta associado um gradiente (A_i, B_i) entao temos que:

$$\left(\frac{\partial z}{\partial x} \right)_1 = A_1 \qquad \left(\frac{\partial z}{\partial y} \right)_1 = B_1$$

Na segunda iteração temos o mesmo processo de subdivisão produzindo os triângulos (planos):

111 - resultado da w_1 aplicada ao triângulo 11 .

212 - resultado da w_2 aplicada ao triângulo 21 .

123 - resultado da w_3 aplicada ao triângulo 12 .

E assim por diante.

O gradiente de um plano , por exemplo , o triângulo 123 será :

$$\left(\frac{\partial z}{\partial x} \right)_{123} = \frac{e_3}{a_3} + \frac{g_3}{a_3} \left(\frac{e_2}{a_2} + \frac{g_2}{a_2} A_1 \right)$$

$$\left(\frac{\partial z}{\partial y} \right)_{123} = \frac{f_3}{d_3} + \frac{g_3}{d_3} \left(\frac{f_2}{d_2} + \frac{g_2}{d_2} B_1 \right)$$

Iterando o sistema de fundges até o infinito os planos formados pelas subdivisões sucessivas dos triângulos anteriores tenderão a pontos, ou seja sua área tenderá a zero. O gradiente deste ponto será então o gradiente da superfície neste ponto.

Seja $\sigma_1\sigma_2\sigma_3\sigma_4.....\sigma_m$ a órbita de um ponto p do atrator , então o gradiente neste ponto será :

$$\left(\frac{\partial z}{\partial x}\right)_{\sigma_1\sigma_2\ldots\sigma_m} = \frac{e_{\sigma 1}}{a_{\sigma 1}} + \frac{g_{\sigma 1}}{a_{\sigma 1}}\left(\frac{e_{\sigma 2}}{a_{\sigma 2}} + \frac{g_{\sigma 2}}{a_{\sigma 2}}\left(\ldots\ldots \frac{e_{\sigma m}}{a_{\sigma m}} + \frac{g_{\sigma m}}{a_{\sigma m}}\, A_1\right)\right)$$

$$\left(\frac{\partial z}{\partial y}\right)_{\sigma_1\sigma_2\ldots\sigma_m} = \frac{f_{\sigma 1}}{d_{\sigma 1}} + \frac{g_{\sigma 1}}{d_{\sigma 1}}\left(\frac{f_{\sigma 2}}{d_{\sigma 2}} + \frac{g_{\sigma 2}}{d_{\sigma 2}}\left(\ldots\ldots \frac{f_{\sigma m}}{d_{\sigma m}} + \frac{g_{\sigma m}}{d_{\sigma m}}\, B_1\right)\right)$$

onde $\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5\ldots\sigma_m$ significa qualquer combinação entre 1, 2 e 3, ou seja os índices das transformações W_i

Se os valores de g_i/a_i e g_i/d_i forem maiores que 1(um) os valores de $\partial z/\partial x$ e $\partial z/\partial y$ podem tender rapidamente para o infinito, porque estes são fatores de multiplicação para cada transformação. Os valores de $\partial z/\partial x$ e $\partial z/\partial y$ são as declividades das retas tangentes ao eixo x e y neste ponto. Se estes valores tendem ao infinito é porque o ângulo que as retas tangentes formam com os eixos tende a 90° e não existe tangente de 90°. Logo a superfície não - diferenciável neste ponto.

Quando os valores dos parâmetros g_i se aproximam de 1 (um) a superfície apresenta picos e encostas íngremes, quando os valores de g_i são iguais a zero então a superfície recai na interpolação por planos.

Para que a superfície seja diferenciável os parâmetros g_i's terão que ser menores que a_i e d_i para que:

$$\left|\frac{g_i}{a_i}\right| < 1 \qquad e \qquad \left|\frac{g_i}{d_i}\right| < 1$$

Para que a superfície seja diferenciável em qualquer ponto é necessário que numa dada vizinhança de qualquer ponto desta superfície seja possível traçar um plano tangente a este ponto, ou seja,numa dada vizinhança esta superfície é linear.

O algorítmo de interpolação fractal tem como principal característica permitir que o padrão de rugosidade da superfície se repita em qualquer escala. Se

o padrão de rugosidade se repitir em qualquer escala não será possível definir uma vizinhança linear qualquer que seja a escala, ou seja, se a superfície é diferenciável então não reproduz a rugosidade em qualquer escala.

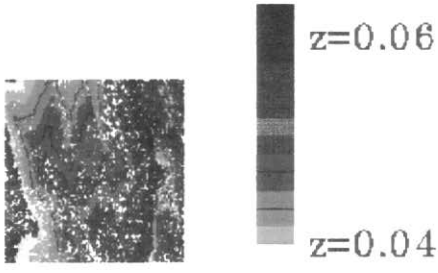


Figura 2.5a-Detalhe da região assinalada na figura 2.4a.

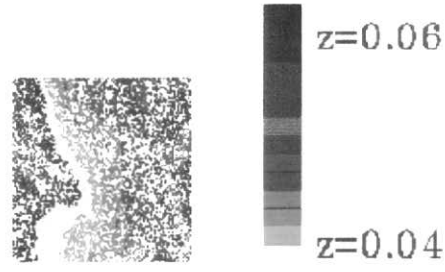


Figura 2.5b-Detalhe da região assinalada na figura 2.4b.

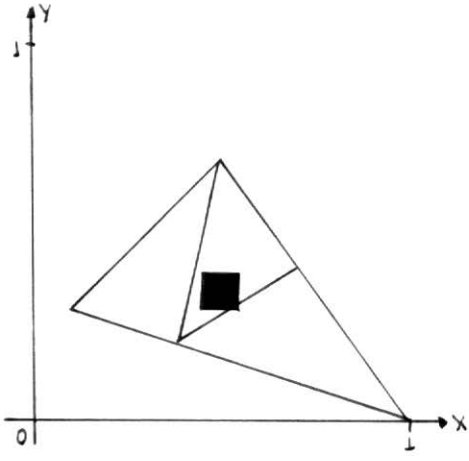


Figura 2.4a - Em negrito a região onde é feito o detalhe.

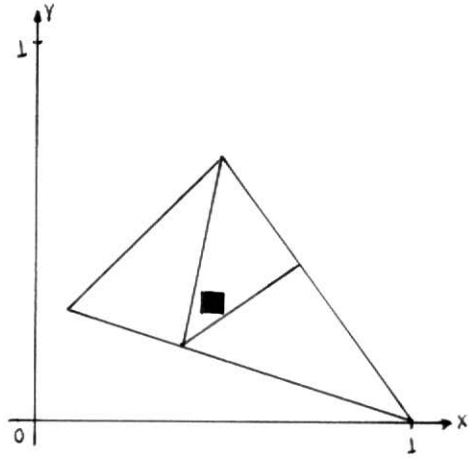


Figura 2.4b - Em negrito a região onde é feito o detalhe

II.5 - RUGOSIDADE DA SUPERFÍCIE.

Como visto na seção anterior, para se preservar o padrão de rugosidade é necessário estabelecer os valores dos parâmetros $|g_i|$ de tal forma que sejam maiores que os valores dos parâmetros $|a_i|$ e $|d_i|$ para todo $i = 1,2,...N$.

Na figura 2.5a e 2.5b é mostrado detalhe da superfície correspondente às áreas assinaladas nas figuras 2.4a e 2.4b.

II.6 - ESQUEMA DO ALGORÍTMO:

Para maior entendimento do algorítmo de interpolação fractal utilizado neste trabalho é exposto um esquema do algorítmo.

Quando o desejado é um determinado número de pontos (x,y,z) da superfície é utilizado o seguinte esquema:

1º - com a triângulação e os parâmetros g_i calcular os demais parâmetros.

2º - arbitrar valores para x_o ,y_o ,z_o no intervalo [0,1].

3º - escolher qualquer uma das transformações w_i

4º- aplicar essa transformação ao ponto (x_o,y_o,z_o) obtendo o ponto (x_1,y_1,z_1)

$$x_1 = a_i x_o + b_i y_o + u_i ,$$
$$y_1 = c_i x_o + d_i y_o + v_i ,$$
$$z_1 = e_i x_o + f_i y_o + g_i z_o + t_i .$$

5º - escolher qualquer uma das transformações w_i e aplicar ao ponto (x_1,y_1,z_1) obtendo o ponto (x_2,y_2,z_2).

Assim sucessivamente até um número m de iterações.

Todos os pontos gerados após aproximadamente 10 iterações pertencem à superfície.

Quando o desejado ' o valor da altitude em um determinado ponto (x,y) da superfície o esquema do algorítmo é o seguinte:

1º - com a triângulação e os parâmetros g_i calcular os demais parâmetros e os parâmetros das transformações inversas ($a_i^{-1}, b_i^{-1}, c_i^{-1}, d_i^{-1}$, são os parâmetros da inversa da transformação w_i).

2º - nos pontos (x,y) que se deseja calcular z fazer:

- $x_m = x$ e $y_m = y$

1 - ver a que triângulo da interpolação o ponto (x_m, y_m) pertence. Seja i o índice deste triângulo.

2 - $x_{m-1} = a_i^{-1} x_m + b_i^{-1} y_m - u_i^{-1}$,

$y_{m-1} = c_i^{-1} x_m + d_i^{-1} y_m - v_i^{-1}$,

3 - guardar o índice i em um vetor.

4 - $x_m = x_{m-1}$ e $y_m = y_{m-1}$.

5 - voltar ao passo 1 um número m de iterações.

6 - arbitrar ou estimar o valor de z_o.

7 - fazer : $x_o = x_m$, $y_o = y_m$

8 - para k = 1,..m fazer:

1 - resgatar o último valor do vetor que guarda os índices i.

2 - $x_k = a_i x_{k-1} + b_i y_{k-1} + u_i$,

$y_k = c_i x_{k-1} + b_i y_{k-1} + v_i$,

$z_k = e_i x_{k-1} + f_i y_{k-1} + g_i z_{k-1} + t_i$.

9 - o valor de z no ponto desejado ' z_k quando k = m.

Deste modo pode-se calcular o valor da altitude para qualquer ponto da superfície dentro do triângulo definido como domínio.

CAPÍTULO III

MODELO CHUVA-VAZÃO DISTRIBUÍDO

III.1 - INTRODUÇÃO.

Por ser o interesse deste trabalho a introdução do modelo digital de terreno com interpolação fractal na simulação hidrológica, foi utilizado um modelo chuva-vazão distribuído bastante simplificado. Assim, não foi preocupação deste trabalho ter uma boa representatividade de todas as fases do ciclo hidrológico dentro do modelo.

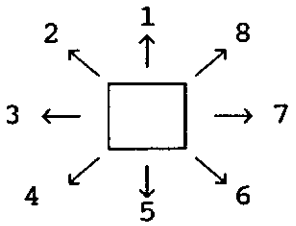
O modelo chuva-vazão proposto neste trabalho apresenta duas partes distintas: o modelo hidrológico e o cálculo dos parâmetros derivados do modelo digital de terreno que são utilizados no modelo hidrológico.

No modelo hidrológico são simulados os processos físicos: a distribuição da chuva pela bacia, as perdas por evapotranspiração e infiltração, e o escoamento superficial. O escoamento sub-superficial e subterrâneo não são considerados por esta versão do modelo.

Para calcular os parâmetros para a execução do modelo hidrológico foi feito um programa separado que utiliza o modelo digital de terreno (MDT) exposto no capítulo II.

A bacia hidrográfica é discretizada em uma malha quadrada. Essa discretização pressupõe que as características hidrológicas sejam homogêneas para toda a área da quadrícula. Assim são usados valores médios para todos os parâmetros e uma única direção de escoamento referentes a cada quadrícula.

Cada elemento da malha possui uma direção de escoamento que pode ser qualquer uma das oito quadrículas vizinhas .



Os dados de entrada do modelo são a precipitação e os parâmetros calculados a partir do MDT incluindo a direção de escoamento.

O resultado da utilização do modelo é a hidrógrafa produzida pelo escoamento superficial de um evento chuvoso.

Em cada elemento da malha são simuladas as perdas por evaporação e infiltração. O volume restante então sofre retenção conforme um parâmetro antes de escoar para a quadrícula jusante. Desse modo é simulado o efeito de amortecimento na bacia.

Em cada intervalo de tempo é feito o balanço hídrico em todas as quadrículas e calculada a vazão na quadrícula correspondente ao ponto de controle. Desse modo é simulado o efeito de translação do escoamento na bacia.

A simulação da hidrógrafa é feita do seguinte modo:

Em cada passo de tempo fazer:

1º - distribuir a precipitação em todas as quadrículas.

2º - em cada quadrícula:

- acrescentar a parcela de contribuição da quadrícula à montante que escoou no tempo anterior.
- calcular as perdas por evaporação e infiltração.
- fazer o balaço hídrico: o volume superficial é igual ao volume existente na quadrícula mais o volume de chuva menos as perdas por evaporação e infiltração.
- determinar a parcela de ägua que escoa e a parcela que fica retida na quadrícula pelo efeito de amortecimento.

3º - calcular a vazão neste passo de tempo como sendo o

volume que chegou à quadrícula que corresponde ao ponto de controle.

Para este modelo são feitas as seguintes simplificações:

- A precipitação tem intensidade constante dentro do intervalo de tempo adotado para a execução do modelo.
- A precipitação tem a mesma intensidade para toda área da quadrícula num dado intervalo de tempo.
- As perdas por transpiração e retenção vegetal são desprezadas.
- As perdas por evaporação e infiltração são retiradas do volume precipitado no final do intervalo de tempo e não continuamente no tempo.
- A parcela do volume superficial retirado o amortecimento somente escoa no final do intervalo de tempo.

O intervalo de tempo tem que ser compatível com a dimensão das quadrículas da malha para que seja reproduzida a velocidade do escoamento e o efeito de amortecimento característicos da bacia.

Para se determinar o intervalo de tempo adequado para uma dada dimensão da malha pode-se utilizar a velocidade média da bacia :

$$\Delta t = \frac{\Delta s}{v_m}$$

onde : Δs o comprimento do lado das quadrículas,

Δt o intervalo de tempo usado na simulação,

v_m a velocidade média característica da bacia.

A velocidade média da bacia é' calculada utilizando-se o tempo de concentração e o máximo percurso da água pela bacia:

$$v_m = \frac{s_{max}}{t_c}$$

onde : v_m ' a velocidade média da bacia,

s_{max} ' o maior percurso que a água faz dentro da bacia,

t_c ' o tempo de concentração da bacia.

III.2 - SIMULAÇÃO DOS PROCESSOS FÍSICOS.

III.2.1 - Distribuição da precipitação sobre a bacia:

Para a distribuição da intensidade da chuva sobre a bacia a técnica escolhida foi da média aritmética das alturas de precipitação nos postos pertencentes à bacia.

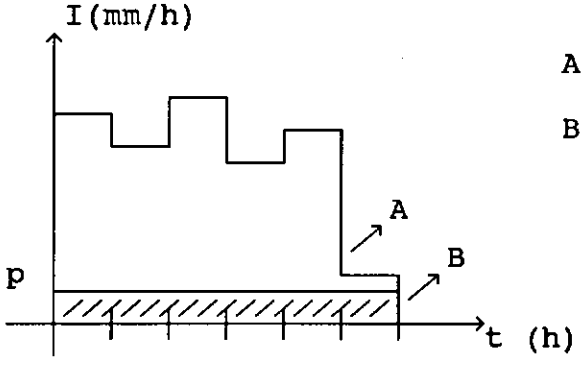
$$P_{(i,j)} = \frac{1}{n} \sum_{k=1}^{n} P_k$$

onde $P_{(i,j)}$ ' a altura de chuva na quadrícula (i,j)
n ' o número de postos da bacia
P_k ' a altura de chuva no posto k ,
$k = 1,2,..n.$

III.2.2 - Perdas por evaporação.

As perdas por evaporação são consideradas constantes para todos os intervalos do tempo de simulação.

Essas perdas são calculadas a partir de um percentual sobre o volume de precipitação do evento chuvoso. O percentual adotado para este trabalho ' de 10% (dez por cento) do volume total de chuva na bacia.



A = volume total de chuva

B = volume total que evapora na bacia

$$evp_{(i,j)} = \left(\frac{vol}{nt} \cdot p \right) / nq$$

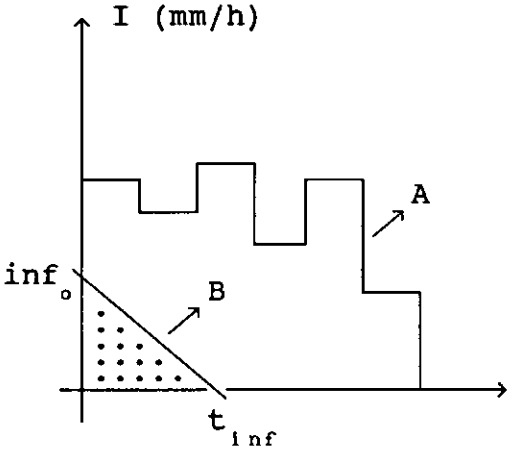
sendo $evp_{(i,j)}$ o volume que evapora na quadrícula (i,j) em qualquer intervalo de tempo,
vol volume total de precipitação na bacia durante o evento chuvoso,
nq número de quadrículas pertencentes ' bacia,

nt número de intervalos de tempo da simulação.

p percentual do volume total que evapora.

III.2.3 - Perdas por infiltração.

As perdas por infiltração são calculadas por uma função linear utilizando-se o volume total de chuva na bacia num dado evento chuvoso e o parâmetro tempo de duração da infiltração na quadrícula. A separação da chuva efetiva é feita do seguinte modo:



I = intensidade de chuva ou intensidade da infiltração

A = linha que determina o volume total de chuva

B = função que define a intensidade de infiltração no tempo

A linha B é representada pela equação:

$$I_{(i,j)} = \frac{inf_o}{t_{inf}} \cdot t + inf_o \qquad (3.1)$$

onde: $I_{(i,j)}$ é a intensidade de infiltração no tempo t na quradrícula (i,j),

t_{inf} ' o tempo de infiltração da quadrícula (i,j),

inf_o ' a intensidade inicial da infiltração.

O valor de inf_o é determinado a partir do volume total de precipitação na bacia de tal forma que a infiltração total na bacia seja igual ou inferior a uma percentagem do volume total do evento chuvoso. O volume

que infiltra em uma quadrícula é igual a área sob a linha B:

$$vol \cdot p / nq = \frac{1}{2} \left| t_{inf} \cdot inf_o \right|$$

sendo: vol o volume total da chuva na bacia em todo o evento chuvoso,
p percentual do volume total vol,
nq número de quadrículas da malha.

Assim podemos calcular o valor de inf_o para cada quadrícula (i,j) em função do parâmetro t_{inf} :

$$inf_o = \frac{2\ p\ vol}{nq\ t_{inf}} \qquad (3.2)$$

Substituindo a equação (3.2) na equação (3.1) temos a infiltração $I_{(i,j)}$ na quadrícula (i,j) em função do tempo:

$$I_{(i,j)} = \frac{2\ p\ vol}{nq} \left(\frac{1}{t_{inf}} - \frac{t}{t^2_{inf}} \right)$$

O parâmetro t_{inf} é calculado, a partir do modelo digital de terreno, para todas as quadrículas da malha.

III.2.4 - Escoamento superficial.

O escoamento superficial é considerado unidimensional e de velocidade constante na direção de escoamento de cada quadrícula.

A direção de escoamento de uma quadrícula define a quadrícula a jusante dessa quadrícula.

O volume superficial, depois de retiradas as perdas por evaporação e infiltração, sofre amortecimento antes de escoar para a quadrícula jusante. Do volume superficial é retirada a parcela de água que escoa como função de um parâmetro de retenção que é constante para cada quadrícula e calculada em função da declividade e rugosidade da mesma.

A parcela de água que escoa é' calculada pelas equações baseadas na equação de deplecionamento de reservatório superficial do modelo SMAP (LOPES et al, 1981).

$$q_{(i,j)} = V_{sup}(K_{(i,j)}) \qquad (3.3)$$

quando a velocidade da quadrícula é igual ou menor à' velocidade m'dia da bacia.

$$q_{(i,j)} = V_{sup}(K_{(i,j)} - 1) \qquad (3.4)$$

quando a velocidade da quadrícula é maior que a velocidade média da bacia.

onde: $q_{(i,j)}$ é a vazão que escoa da quadrícula (i,j) para a quadrícula jusante,

V_{sup} é o volume superficial da quadrícula (i,j),

$K_{(i,j)}$ é a constante de recessão da quadrícula (i,j).

A cada passo de tempo é calculado o volume $q_{(i,j)}$ que escoa de cada quadrícula e esse volume é' adicionado ao volume existente na quadrícula jusante no próximo passo de tempo. É feito então um balanço hídrico em todas as quadrículas.

O parâmetro de retenção K é calculado pela relação entre a velocidade média de escoamento da bacia e a velocidade do escoamento na quadrícula.

$$K_{(i,j)} = v_{(i,j)} / v_m \qquad (3.5)$$

onde: $v_{(i,j)}$ ' a velocidade da quadrícula (i,j)

Pelas equações (3.3) e (3.4) temos que:

1º - Se a velocidade da quadrícula é igual a velocidade média da bacia, neste caso $k_{(i,j)} = 1$, então o volume de água acumulado num intervalo de tempo dentro daquela quadrícula escoa totalmente no final do intervalo..

2º - Se a velocidade da quadrícula é menor que a velocidade média da bacia, neste caso $k_{(i,j)} < 1$, então o volume desta quadrícula escoa parcialmente respeitando a relação entre as velocidades.

3º - Se a velocidade da quadrícula é maior que a velocidade média da bacia, neste caso $k_{(i,j)} > 1$, então significa que o volume de água da quadrícula escoa antes de completar o intervalo de tempo. Ou que o percurso percorrido pelo escoamento ' maior que o comprimento da quadrícula. Neste caso o volume superficial escoa para a quadrícula jusante da quadrícula jusante porporcionalmente ' relação entre as velocidades. O volume que escoa de uma quadrícula pode escoar no máximo por duas quadrículas no mesmo passo de tempo.

III.3 - CÁCULO DOS PARÂMETROS FISIOGRÁFICOS A PARTIR DO MODELO DIGITAL DE TERRENO.

III.3.1 - Introdução.

Alguns parâmetros utilizados pelos modelos chuva-vazão distribuídos podem ser derivados diretamente da topografia, outros parâmetros recebem influência de outros fatores além do relevo. É exemplo do primeiro caso a declividade, no segundo caso temos a capacidade de infiltração e evaporação, entre outros.

Como conseqüência da proposta deste trabalho optou-se por derivar os parâmetros somente a partir do relevo sem considerar a influência de outros fatores.

Os parâmetros derivados do modelo digital de terreno são taxa de infiltração e velocidade do escoamento superficial. Foi implementado também um algorítmo de derivação de rede de rios a partir de modelo digital de terreno. Estes parâmetros são calculados para cada elemento da malha do modelo distribuído. A malha utilizada no modelo distribuído não tem nenhuma relação com a malha utilizada na triângulação do modelo digital de terreno. A malha do modelo distribuído é definida a partir

da dimensão escolhida para as quadrículas. Em qualquer ponto desta malha pode ser calculada a altitude utilizando o algorítmo de interpolação fractal exposto na seção 2, capítulo II.

III.3.2 - Declividade das quadrículas da malha.

O algorítmo de interpolação fractal permite que a superfície seja diferenciável dependendo dos valores dos parâmetros g_i (ver seção 6,capítulo II). Entretanto, por serem estes parâmetros responsáveis pela rugosidade da superfície, essa restrição torna o algorítmo pouco abrangente.

Optou-se então por não restringir os valores de g_i e por calcular a declividade m'dia da quadrícula através de ajuste de um plano a alguns pontos da superfície pertencentes ' quadrícula.

A técnica de ajuste escolhida foi o método dos mínimos quadrados:

- calcula-se a altitude em nove pontos da quadrícula: os vértices, o ponto central e os pontos médios dos lados da quadrícula.
- ajusta-se um plano a estes pontos por método dos mínimos quadrados: dados os pontos (x_i, y_i, z_i), $i = 1,2,...9$ obtemos

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \cdot \\ \cdot \\ \cdot \\ z_9 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & x_9 & y_9 \end{bmatrix} \begin{bmatrix} \beta_o \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_9 \end{bmatrix}$$

Temos entro $i = 1,2...,9$ equações da forma:

$$z_i = \beta_0 + \beta_1 x_i + \beta_2 y_i + \varepsilon_i$$

onde vamos minimizar :

$$\text{Min} \sum_{i=1}^{9} \varepsilon^2$$

O gradiente deste plano ajustado a estes pontos é : (β_1, β_2)

III.3.3 - Direção de escoamento:

A direção de escoamento de cada quadrícula é' calculada utilizando-se as altitudes médias das quadrículas. As quadrículas vizinhas são numeradas de 1 a 8 no sentido anti-hor±rio. Para cada quadrícula verifica-se qual das oito quadrículas vizinhas possui menor altitude, a direção de escoamento será o número da quadrícula de menor altitude.

Este procedimento pode gerar buracos inexistentes na bacia, o que provoca descontinuidades nos caminhos do escoamento, necessita-se, portanto, de procedimento de correção desse erros. Neste trabalho a correção é feita manualmente, mudando-se as direções de escoamento que provocam descontinuidade do seguinte modo: se uma quadrícula apresenta buraco, altitude menor que as suas vizinhas, compara-se com o mapa da bacia e atribui-se a ela outra direção de escoamento que seja compatível com a os caminhos dos escoamentos da bacia.

III.3.4 - Taxa de infiltração:

A taxa de infiltração é variável para cada evento chuvoso, porque depende do volume total da precipitação do evento chuvoso. O parâmetro tempo de infiltração (t_{inf}) é constante para uma dada quadrícula para qualquer evento chuvoso e ' calculado assumindo-se que ' função apenas da declividade da quadrícula.

$$t_{inf} = - \frac{t_{max}}{90^o} . \theta_{(i,j)} + t_{max}$$

onde: t_{inf} é o tempo de infiltração em função da declividade da quadrícula,

$\theta_{(i,j)}$ é o ângulo de declividade da quadrícula

t_{max} é o valor máximo para t_{inf} , único para toda a bacia.

III.3.5 - Velocidade de escoamento superficial nas quadrículas:

A velocidade de escoamento $v_{(i,j)}$ da quadrícula (i,j) é calculada em função da declividade e da rugosidade da superfície da quadrícula pela equação :

$$v_{(i,j)} = m_{(i,j)} \cdot \theta_{(i,j)} \qquad (3.6)$$

onde: $v_{(i,j)}$ é a velocidade do escoamento na quadrícula,

$\theta_{(i,j)}$ é o ângulo que a declividade da quadrícula (i,j) faz com F(x,y) = 0,

$m_{(i,j)}$ é determinado pelo parâmetro que define a rugosidade da quadrícula.

Para que a velocidade nas quadrículas seja compatível com a velocidade média na bacia o valor de $m_{(i,j)}$ é determinado por um m_{med} que é uma relação entre a velocidade na bacia e a declividade média das quadrículas:

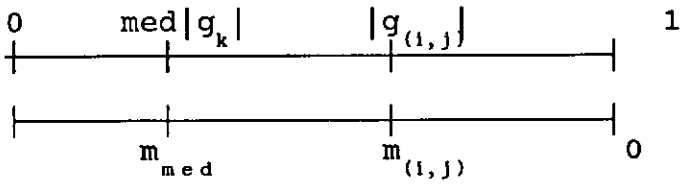
$$m_{med} = \frac{v_m}{\theta_{med}} \qquad (3.7)$$

onde: m_{med} é a declividade característica da bacia

v_m é a velocidade média da bacia,

θ_{med} é a média dos ângulos de declividade das quadrículas.

O valor de m_{med} corresponde à média do valor absoluto do parâmetro g_k, k = 1,2,3..N, onde N é o número de triângulos utilizados na triângulação dos pontos de interpolação, ou seja :



onde $med|g_k|$ é a média dos parâmetros das transformações w_k pertencentes ao sistema iterativo de funções $\{ \mathbb{R}^3; w_k, k = 1,2,3,...N \}$,

$g_{(i,j)}$ é o parâmetro livre g_k associado ao triângulo da interpolação ao qual a quadrícula (i,j) pertence.

$$med|g_k| = \sum_{k=1}^{N} |g_k|$$

então temos que :

$$\frac{1- med|g_k|}{m_{med}} = \frac{1 - |g_{(i,j)}|}{m_{(i,j)}}$$

Tirando o valor de $m_{(i,j)}$ temos:

$$m_{(i,j)} = m_{med} \frac{(1 - |g_{(i,j)}|)}{(1 - med\ |g_k|\ )} \qquad (3.8)$$

Substituindo as equações (3.8) e (3.7) na equação (3.6) temos:

$$v_{(i,j)} = \frac{v_m\ (1 - |g_{(i,j)}|)}{\theta_{med}\ (1 - med|\ g_k\ |)} \cdot \theta_{(i,j)}$$

Assim a velocidade da quadrícula é calculada em função da declividade expressa em $\theta_{(i,j)}$ e da rugosidade expressa em $g_{(i,j)}$.

III.3.6 - Determinação automática da rede de rios.

Foi implementado um algorítmo de derivação da

rede de drenagem proposto por O' CALLAGHAN e MARK (1984).

A rede de drenagem é determinada a partir de uma matriz que representa a elevação média em cada quadrícula e a direção de escoamento de cada quadrícula do seguinte modo :

- ordenar a matriz de elevação da maior altitude para a menor altitude guardando os índices i,j da matriz original. Esta matriz com os índices define a posição que a altitude de determinada quadrícula ocupa no ranking de elevação.
- calcular uma matriz de fluxo acumulado:
  - assumir o valor de 1(um) para cada elemento de uma matriz de fluxo acumulado que corresponde às quadrículas da malha.
  - começando da quadrícula de maior elevação para a de menor elevação fazer:
    - encontrar a quadrícula jusante desta quadrícula através da matriz direção de escoamento.
    -somar o conteúdo do elemento da matriz de fluxo acumulado desta quadrícula ao conteúdo da quadrícula jusante.

Ao final deste processo é obtida a matriz de fluxo acumulado que representa o total de quadrículas que drenam para cada quadrícula da malha. Para quadrículas que não recebem escoamento de nenhuma outra este valor é 1 (um) e significa que elas são os divisores de drenagem. A rede de drenagem ' definida como sendo as quadrículas que recebem contribuição de um certo número de quadrículas. Este número é arbitrário e sua determinação varia de autor para autor(TRIBE, 1992), (DONKER, 1992).

O principal problema relacionado com este algorítmo é a descontinuidade da rede de drenagem . Isso é devido ao fato de que os modelos digitais de terreno podem gerar buracos ou regiões planas inexistentes na superfície da qual foram extraídos os dados de elevação.

Para contornar este problema, nesta atual fase da pesquisa, foi adotado um procedimento que consite de examinar os caminhos de escoamento obtidos através do algorítmo acima e corrigir as descontinuidades manualmente.

CAPÍTULO IV

APLICAÇÃO DO MODELO

IV.1 - REGIÃO ESCOLHIDA .

Foi escolhida, para aplicação do modelo, uma sub-bacia do rio Paraitinga até o exutório situado em São Luiz do Paraitinga.

O rio Paraitinga é um dos formadores do rio Paraíba do Sul. Essa sub-bacia está situada parte no estado de São Paulo e parte no estado do Rio de Janeiro compreendendo uma área de 1956 km^2. É formada em sua maior parte de solo vegetado, numa região montanhosa, sendo que um dos divisores da área de drenagem é a serra do Mar.

Os dados utilizados neste estudo foram coletados de mapas topográficos e um posto de medição de vazão em São Luiz do Paraitinga.

IV.2 - SIMULAÇÃO DO RELEVO .

Foram coletadas altitudes em dezenove pontos de mapas topográficos em escala 1 : 250000. Destes dezenove pontos treze pertencem à bacia e foram escolhidos por pertecerem ao talvegue principal e ao divisor de drenagem da bacia. Os seis pontos restantes foram coletados para satisfazer o algorítmo de interpolação fractal.

A partir destes pontos foi feita uma triângulação obedecendo os critérios do algorítmo de interpolação. Os pontos coletados e a conectividade dos triângulos se encontram nas tabelas I e II do APÊNDICE I. A disposição dos pontos coletados na bacia de estudo é' mostrada nas figuras 4.1b. A bacia foi inserida em um plano cartesiano onde à ordenada e à abscissa de valor igual a 1 (um) corresponde o valor de 95250 metros a partir da origem do plano cartesiano (figura 4.1a).

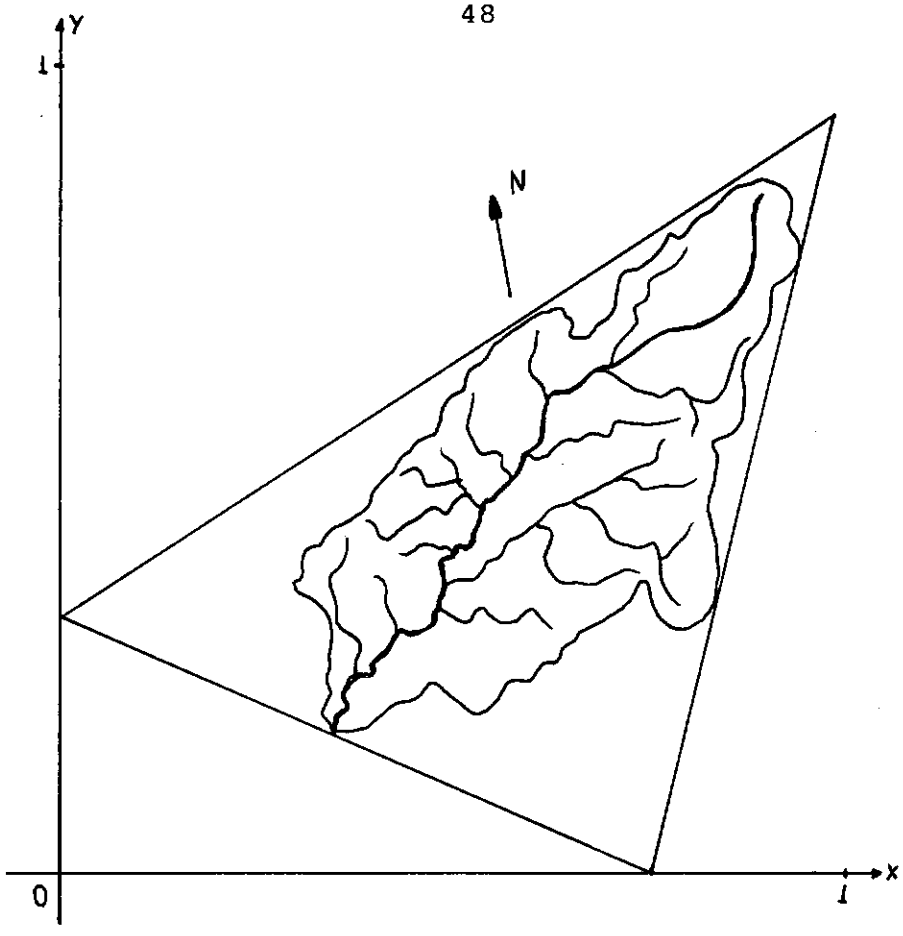


Figura 4.1a - Bacia inserida no triângulo domínio.

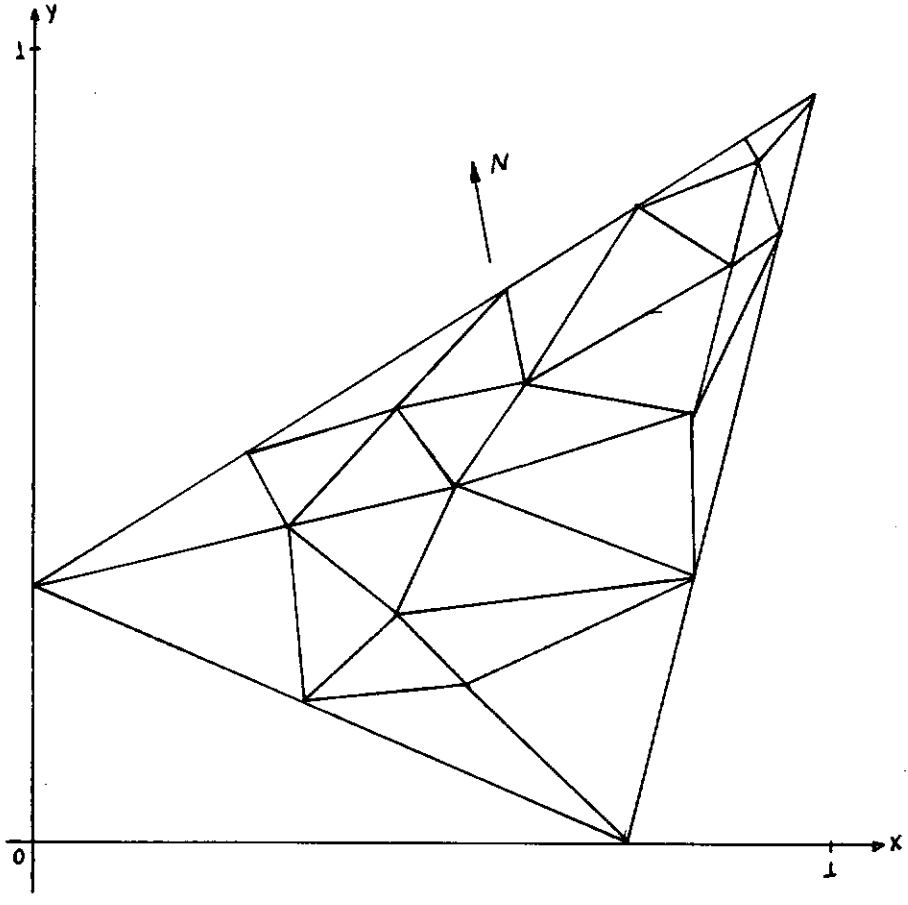


Figura 4.1b - Triângulação com os pontos de interpolação

As abscissas e ordenadas dos pontos de interpolação correspondem às coordenadas x e y deste plano cartesiano e as cotas dos pontos de interpolação correspondem às altitudes destes pontos. Como para esta bacia as altitudes maiores são de 1800 m, foi necessário multiplicá-las por um fator de 1×10^{-4} para que as cotas permaneçam no intervalo [0,1].

A determinação do relevo é feita a partir da triângulação e da escolha dos parâmetros livres g_i. Com a mesma triângulação podemos obter diversas superfícies diferentes, bastando mudar os parâmetros g_i.

Foram testados 3 (três) conjuntos de parâmetros diferentes. Os valores dos parâmetros g_i para o terceiro caso estão na tabela IV do APÊNDICE I. Para o segundo caso foi testado o valor de 0,2 para todos os parâmetros g_i. Para o primeiro caso o valor utilizado foi de 0,3 para todos os parâmetros g_i. As figuras 4.2a, 4.2b, 4.2c mostram os resultados das simulações do relevo para os diferentes conjuntos de parâmetros.

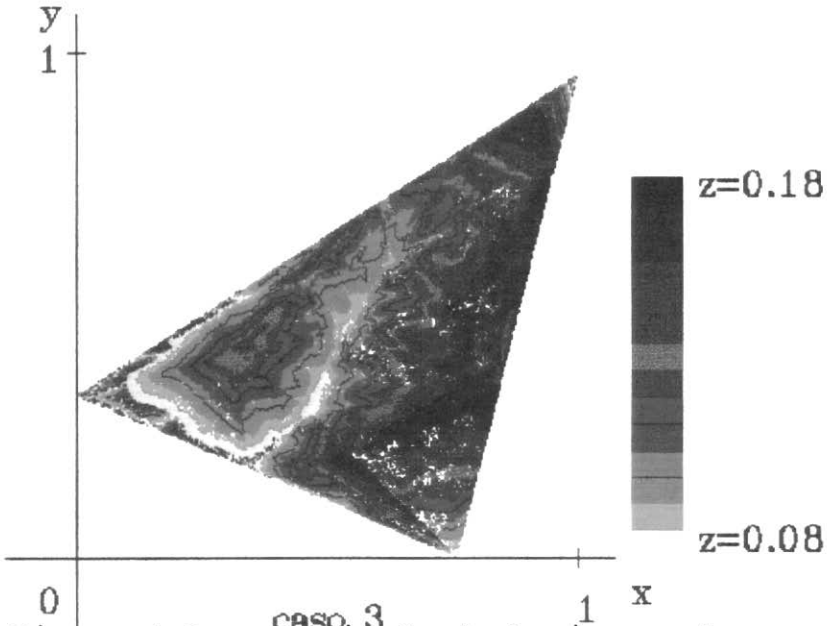


Figura 4.2a - Projeção da bacia no plano z= 0. As altitudes estão representadas em cores. Parâmetros g_i variáveis.

Para esta simulação do relevo foram usadas 10 (dez) iterações e a estimativa de z inicial como o valor médio das altitudes nos pontos de interpolação.

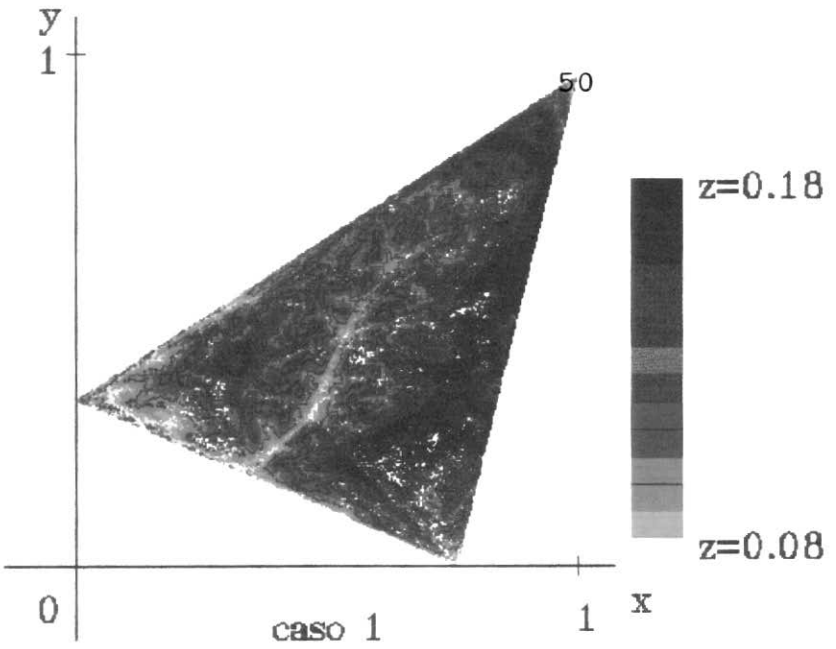


Figura 4.2a - Projeção da bacia no plano z= 0. As altitudes estão representadas em cores. Parâmetros g_i = 0,3.

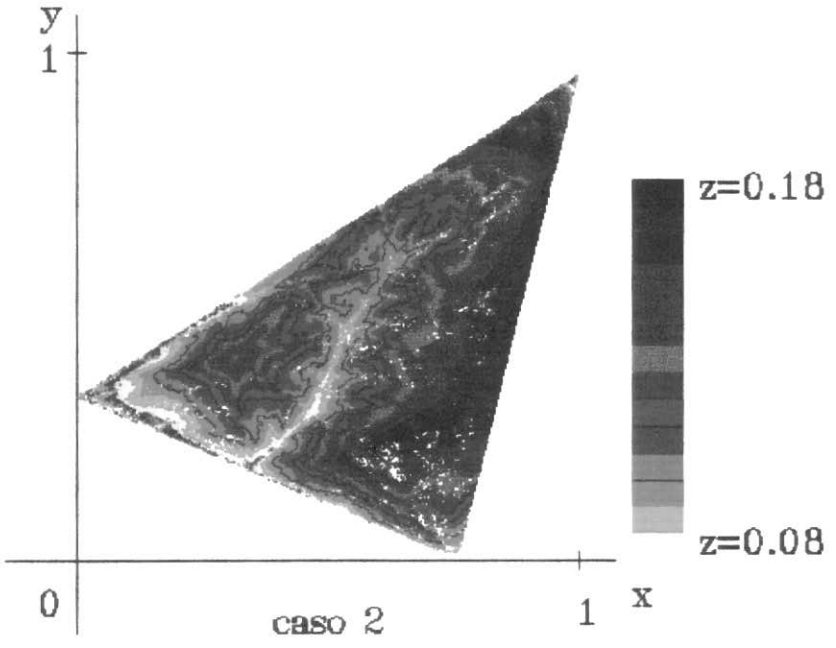


Figura 4.2a - Projeção da bacia no plano z= 0. As altitudes estão representadas em cores. Parâmetros g_i = 0,2.

IV.3 - CÁLCULO DOS PARÂMETROS PARA O MODELO HIDROLÓGICO.

Para calcular os parâmetros que são utilizados no modelo hidrológico, alguns dados foram retirados de dados reais e outros dados foram arbitrados:

- para o comprimento máximo de percurso da água na bacia foi utilizado o comprimento do talvegue principal do exutório até o divisor de drenagem mais longínquo da bacia. Esta medida é de 88750 m.
- o tempo de concentração da sub-bacia foi calculado utilizando-se a fórmula de Ribeiro (PITTHAN,1973):

$$t_c = \frac{16\ L}{60\ (1{,}05 - 0{,}2\ p)\ (100\ S)^{0{,}04}}$$

sendo as variáveis:

t_c tempo de concentração em horas,
L comprimento do curso d'água principal em km,
S declividade do curso d'água principal
p percentagem da bacia com cobertura vegetal.

Foram utilizados os seguintes valores:

L = 88,75 km
p = 0,9
S = h / L = 0,012
onde h = 1,066km ' a diferença entre as cotas do ponto mais alto do curso d'água principal e o ponto do exutório.

- para estes valores o tempo de concentração encontrado foi de 27 horas.
- a velocidade média de escoamento na bacia foi então estimada, a partir do tempo de concentração t_c e do comprimento máximo de percurso da água L, em torno de 0,9 m/s.
- para o valor máximo do parâmetro responsável pela infiltração (t_max) foi arbitrado o valor de t_max = 20 horas.
- para o parâmetro que limita as velocidades nas quadrículas (θ_{med}) é calculada a média dos ângulos das declividades das quadrículas.
- para o parâmetro que representa rugosidade média da bacia

(med |g|) foi utilizada a média dos parâmetros g_i.
- foi definida uma malha quadrada cuja origem corresponde ao ponto (0,314; 0,157) do plano cartesiano. A dimensão das quadrículas da malha foi calculada para três casos :
1º- quadrículas de 6250m x 6250m, com malha composta de 10x10 quadrículas.
2º - quadrículas de 2500m x 2500m, com malha composta de 27x25 quadrículas.
3º- quadrículas de 1250m x 1250m, com malha composta de 50x54 quadrículas.

O modelo que calcula os parâmetros fisiográficos (mod_terr) foi executado com os dados acima para os três conjuntos de parâmetros g_i, portanto com três superfícies diferentes. Os resultados para a malha de 27x25 quadrículas estão nas figuras 4.3a, 4.3b, 4.3c, os resultados para malha de 54x50 quadrículas estão nas figuras 4.4a, 4.4b, e 4.4c onde são apresentadas as redes de escoamento da bacia obtidas através do modelo rede_rio.

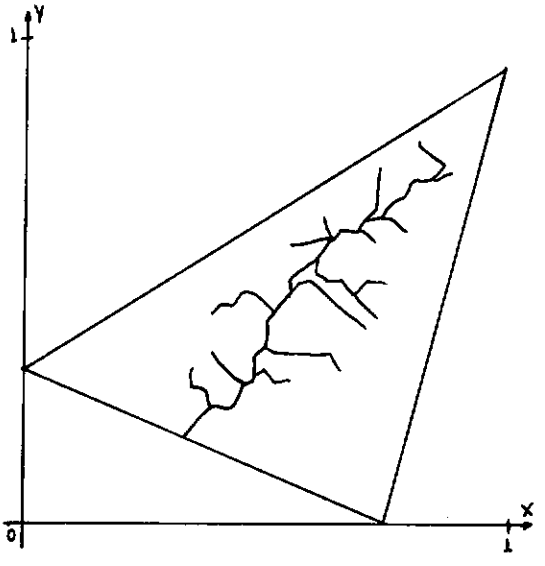


Figura 4.3a - Rede de rios gerada com malha 27x25 e parâmetros g_i variáveis

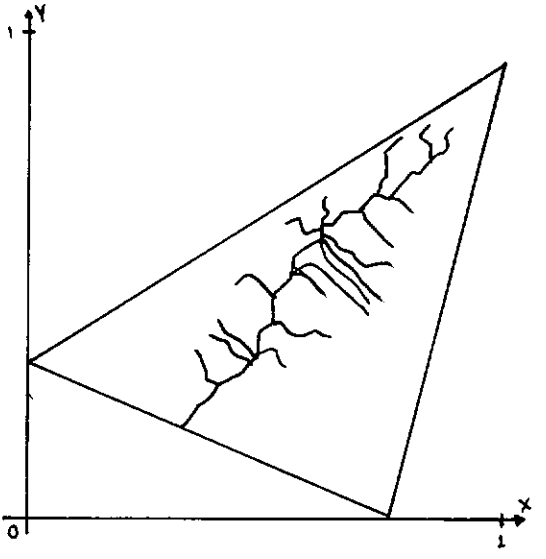


Figura 4.3b - Rede de rios gerada com malha 27x25 e parâmetros g_i = 0,2.

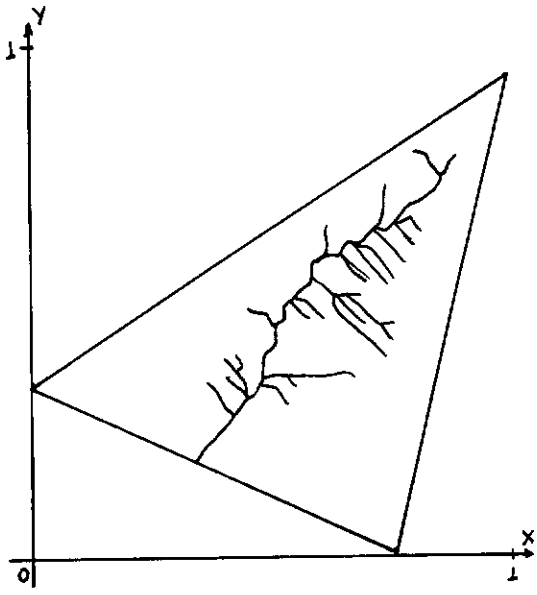


Figura 4.3c - Rede de rios gerada com malha 27x25 e parâmetros g_i = 0,3

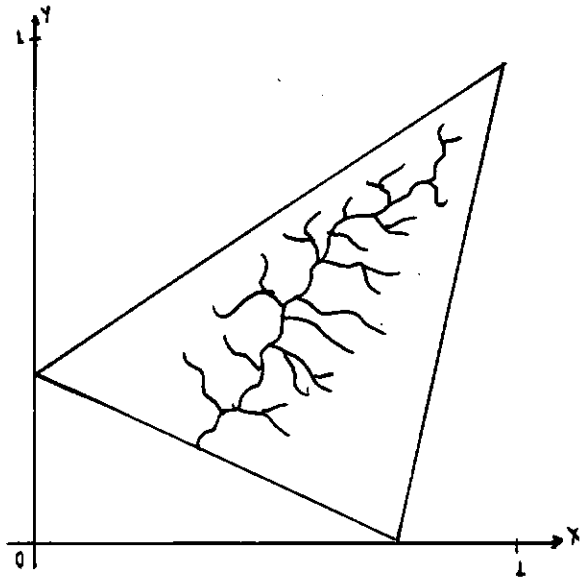


Figura 4.4a - Rede de rios gerada com malha 54x50 e parâmetros g_i variáveis

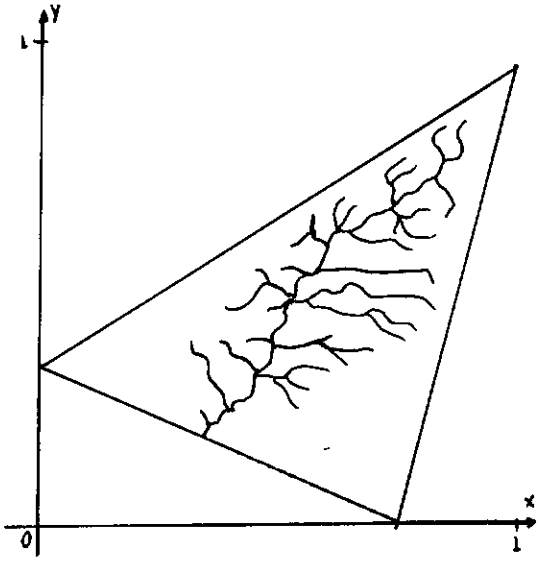


Figura 4.4b - Rede de rios gerada com malha 54x50 e parâmetros g_i = 0,2

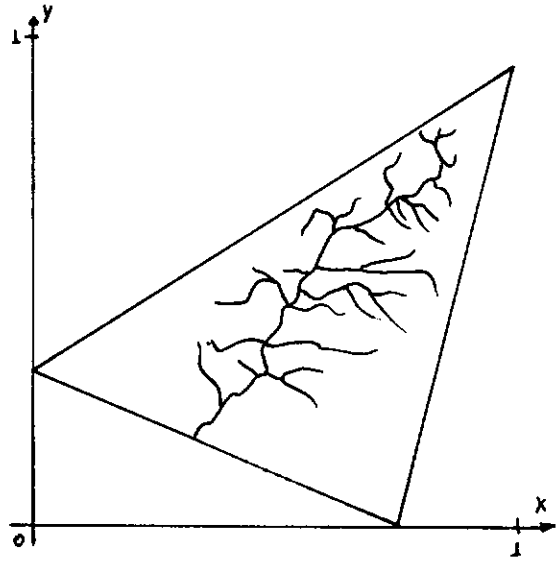


Figura 4.4c - Rede de rios gerada com malha 54x50 e parâmetros g_i = 0.3.

Para a comparação das redes geradas automaticamente e a rede da bacia real é reproduzida na figura 4.5 a bacia real.

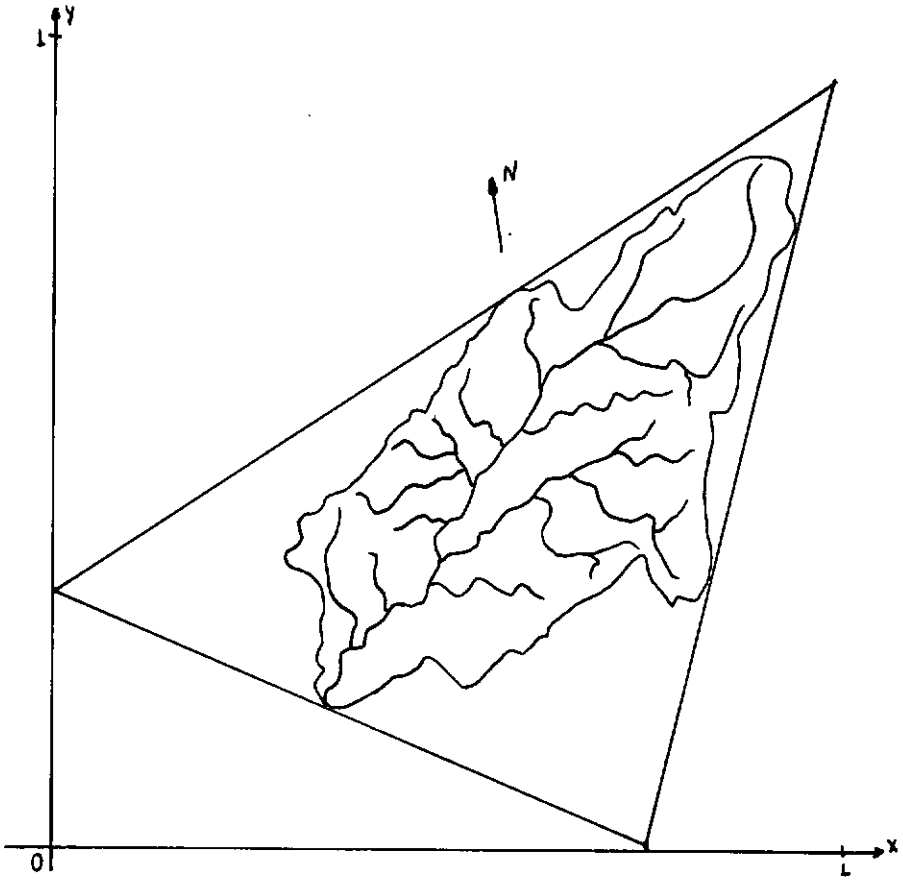


Figura 4.5 - Mapa da bacia do rio Paraitinga.

Para corrigir as descontinuidades das redes de escoamento geradas automaticamente devido a buracos inexistentes, foi necessário fazer algumas correções nas direções de escoamento :

- para a malha de 10x10 quadrículas, das 43 quadrículas pertencentes à bacia foram necessárias em média 3 correções, ou seja, 7% de direções de escoamento indevidas.
- para a malha de 27x25 quadrículas, das 304 quadrículas pertencentes à bacia foram necessárias em média 8 correções, ou seja, 2.6% .
- para malha de 54x50 quadrículas, das 1262 pertencentes à bacia foram necessárias em média 30 correções, ou seja, 2,4% .

Para malhas mais refinadas uma percentagem pequena de quadrículas apresentou direção de escoamento

indevidas e a maior parte desses buracos, ou seja, quadrículas cuja altitude é menor que todas as suas quadrículas vizinhas, aconteceu no talvegue principal.

IV.4 - SIMULAÇÃO DE UM EVENTO CHUVOSO.

O tamanho do intervalo de tempo utilizado na simulação variou conforme a dimensão das quadrículas. Este tamanho de intervalo obedeceu ao critério exposto na seção 1, capítulo III :

$$\Delta t = \frac{\Delta s}{v_m}$$

onde Δt o intervalo em segundos,
Δs o comprimento da quadrícula,
v_m a velocidade média do escoamento na bacia.

Os intervalos de tempo obtidos da relação acima foram:
- 1:56 horas para a malha de 10x10 quadrículas.
- 46 minutos para a malha de 27x25 quadrículas.
- 23 minutos para a malha de 54x50 quadrículas.

O volume de água que infiltra foi estimado em 25% do volume total de chuva do evento. O volume que evapora foi estimado em 10% do volume total de chuva do evento.

Foi simulada uma chuva de um único dia de duração com intensidade constante (intensidade de 4 mm/h) para testar o tempo de concentração obtido da hidrógrafa gerada pelo modelo chuva-vazão a partir de diferentes valores para o parâmetro de rentenção superficial (k) utilizando a malha 27x25:
- com os parâmetros de retenção (k) de todas as quadrículas iguais a 0,5, obteve-se o tempo de concentração de 42 horas aproximadamente.
- com os parâmetros k de todas as quadrículas iguais a 1(um) o tempo de concentração obtido foi de 28:30 horas.
- com os parâmetros k de todas as quadrículas iguais a 1,5, o tempo de concentração obtido foi de 24:40 horas.

Os resultados são apresentados na figura 4.6.

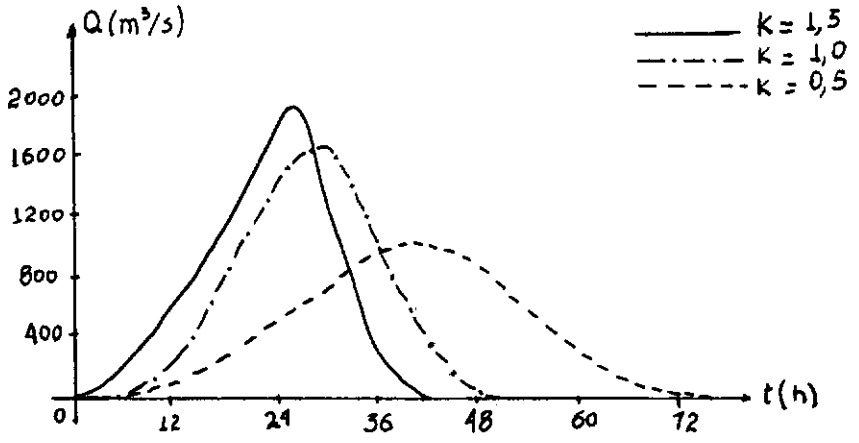


Figura 4.6 - Hidrógrafa das vazões calculadas para três valores do parâmetro de retenção k: 0,5; 1,0; 1,5.

O mesmo evento chuvoso foi utilizado para simular as hidrógrafas utilizando os diferentes tamanhos de malha e os parâmetros de infiltração e retenção superficial obtidos pelo modelo digital de terreno (a superfície utilizada foi a obtida pelos parâmetros g_i iguais a 0,3), os resultados são mostrados nas figuras 4.7a, 4.7b, 4.7c e na tabela V do APÊNDICE I:

- para a malha 10x10 foi obtido o tempo de concentração de 27 horas.
- para a malha 27x25 foi obtido o tempo de concentração de 33:50 horas.
- para a malha 54x50 foi obtido o tempo de concentração de 27:44 horas.

Nas hidrógrafas obtidas pode-se observar que o tempo de concentração, que vai do instante t = 0 até o instante do pico de vazão da hidrógrafa, ficou em torno do tempo de concentração estimado para a bacia que é de 27 horas. Para malhas de 10x10 e 27x25 quadrículas os picos de vazão são mais acentuados do que para a malha 54x50 quadrículas embora o volume de água, área sob a hidrógrafa, tenha sido preservado. Na bacia discretizada em 54x50 quadrículas o escoamento sofreu maior retenção amortecendo assim, o pico da hidrógrafa.

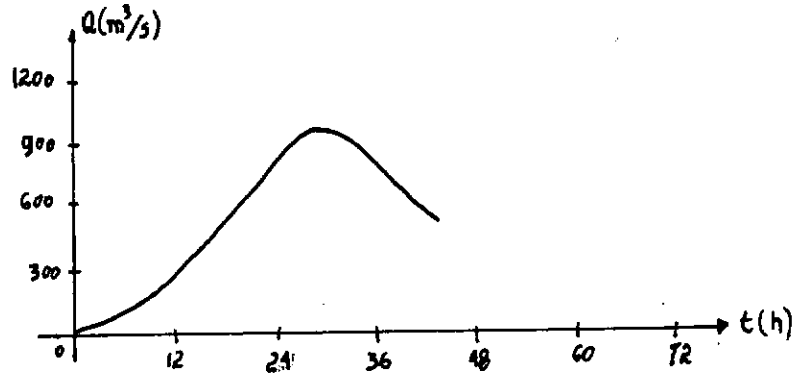


Figura 4.7a - Hidrógrafa aberta de vazão da bacia discretizada em malha 10x10

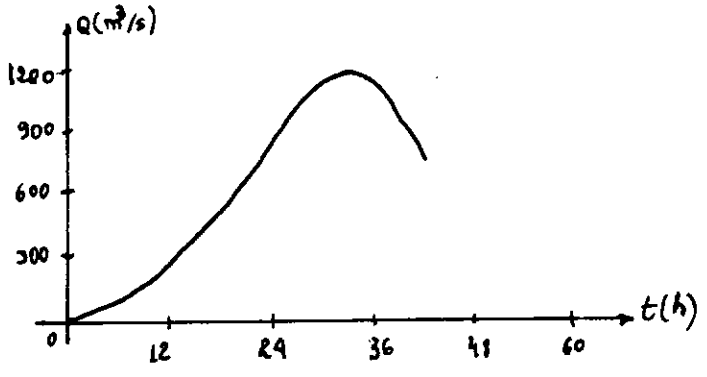


Figura 4.7b - Hidrógrafa aberta de vazão da bacia discretizada em malha 27x25

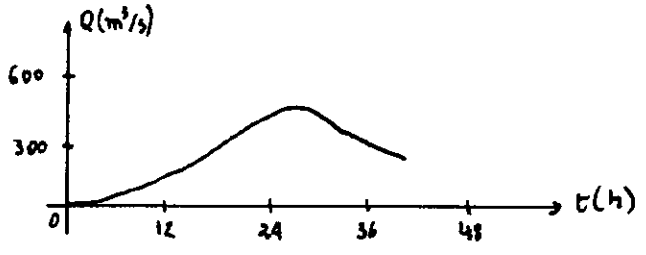


Figura 4.7c - Hidrógrafa aberta de vazão da bacia discretizada em malha 54x50

CAPÍTULO V

CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

V.1 - CONCLUSÃO.

A partir dos resultados obtidos na aplicação do modelo podemos concluir que a simulação do relevo feita através de algorítmo fractal consegue representar as características de irregularidade do relevo com uma base de dados pequena e de fácil manipulação. A superfície modelada pelo algorítmo não foi ajustada à superfície real com a precisão de um mapa , apenas simulou de modo geral a irregularidade da superfície. Para que seja possível reproduzir a superfície real com maior precisão é necessário um número maior de pontos de interpolação.

Das figuras 4.2a,4.2b e 4.2c deduz-se que mesmo para uma triângulação pouco refinada como a que foi utilizada neste trabalho, é possível obter caminhos de escoamento representativos da bacia.

O que se deseja para a aplicação de modelos hidrológicos distribuídos é que a definição dos caminhos do escoamento pela bacia, obtidos das direções de escoamento, sejam uma aproximação razoável da rede de drenagem. Isto o presente algorítmo foi capaz de obter.

As redes de drenagem obtidas automaticamente mostram caminhos de escoamento paralelos, mas isto é devido à escolha da forma da malha e das aproximações feitas para se obter as direções de escoamento das quadrículas. Estas simplificações implicam em suavização das características de irregularidades obtidas através do algorítmo de interpolação fractal.

A obtenção dos parâmetros fisiográficos responsáveis pela infiltração obedeceu a critérios arbitrários com pouca fundamentação física.

O modelo hidrológico proposto neste trabalho é uma simplificação dos modelos hidrológicos chuva-vazão

distribuídos e foi desenvolvido com o intuito de exemplificar a aplicação do algorítmo de simulação do relevo de uma bacia à hidrologia e não simula aspectos importantes como o escoamento subterrâneo.

Apesar das simplificações, este modelo chuva-vazão conseguiu simular o tempo de concentração de uma bacia. Entretanto, não conseguiu simular a diferença entre a curva de ascenção e a curva de recessão de uma hidrógrafa. O modelo mostrou que a curva de recessão é quase simétrica à curva de ascenção para todos os casos simulados.

2- SUGESTÕES PARA TRABALHOS FUTUROS.

Pesquisar uma medida de aderência com o objetivo de ajustar a superfície obtida pelo algorítmo de interpolador fractal à superfícies reais.

Desenvolver o modelo chuva-vazão distribuído de maneira que possa ser incluído o escoamento subterrâneo, além de modelar a infiltração utilizando outros recursos fisicamente fundamentados.

Capacitar a utilização no modelo chuva-vazão de dados de cobertura vegetal, umidade e tipo de solo, etc, para uma melhor representatividade do regime hidrológico de uma bacia.

BIBLIOGRAFIA

BARNSLEY, M., Fractals Everywhere, Academic Press, Inc, 1988.

BEVEN, K. J, "On the Generalized Kinematic Routing Method", Water Resources Reserch, vol 15(5), 1238-1242, 1979.

BEVEN, K. J, "Distributed Models", of " Hydrological Forecasting", M. G. Anderson and T.P. Burt, John Viley & Sons ltd, 1985

CANEDO, P. M., "Hidrologia Superficial", ABRH, Ed. UFRJ, 1989.

COLLINS, S. H., MOON, G. C., "Algorithms for Dense Digital Terrain Models", Photogram Eng. and Remote Sensing, vol 47, 71-76, 1981.

DONKER, N.H.W,"Automatic Extraction of Catchment Hydrologic Properties from Digital Elevation Data", ITC Journal,vol 3, 257-265, 1992.

FETT, W., NEUMANN, P., SCHULTZ, G. A., "Hydrological Model Based on Satellite Imagery and GIS", Int. Symposium on Remote Sensing and Water Resources, Enschede, The Netherlands, 1990.

GILL, P., MURRAY, W., WRIZH, M.H., "Practical Optimization", London, Academic Press, 1981.

HADLEY, G, "Linear Algebra", Reading Mass: Addison-Wesley,1973.

JÜRGENS, H., PEITGEN, H., SAUPE, D., "The Language of Fractals", Scientific American, August, 1990.

LA BARBERA, P., ROSSO, R., "On the Fractal Dimension of Stream Network", Water Resources Reserch, vol 25(4), 735-741, 1989.

LOPES, J. E. G., BRAGA, B. P. F., CONEJO, J. G. L., "SMAP- A Simplified Hydrological Model", Proceedings of the International Symposium on Applied Modelling, Mississippi- USA, 1981.

LOVEJOY, S., "Fractal Properties of Rain, and a Fractal Model", Tellus, vol 37 A, 209-232, 1985.

MANDELBROT, B. B., "Fractal: Form, Chance and Dimension", W. H. Freeman and Co, 1977.

MANDELBROT, B. B., "The Fractal Geometry of Nature", W.H. Freeman & Co, 1983.

MOORE, I. D., GRAYSON, R. B., "Terrain-Based Catchment Partitioning and Runoff Prediction Using Vector Elevation Data", Water Resources Reserch, vol 27(6), 1177-1191, 1991.

MOORE, I. D., GRAYSON, R. B., LADSON, A. R., "Digital Terrain Modelling: a Review of Hydrological, Geomorphological and Biological Applications", Hydrological Processes, vol 5, 3-30, 1991.

NEUMANN, P, FETT, W, and SCHULTZ,G. A, "A Geographic Information System as Data Base for Distributed Hydrological Models", Int. Symposium on Remote Sensing and Water Resources, Enschede, The Netherlands, 1990.

NIKORA, V. I.,"Fractal Structures of River Plan Forms", Water Resources Reserch, vol 27(6), 1327-1333, 1991.

O' CALLAGHAN ,J.F, and MARK,D.M,"The Extration of Drainage Networks from Digital Elevation Data", Comp. Vis., Graphics and Image Proc, vol 28, 323-344, 1984.

PALACIOS-VELEZ, O. L., CUEVAS-RENAUD, B., "SHIFT: A Distributed Runoff Model Using Irregular Triangular Facets", Journal of Hydrology, vol 134, 35-55, 1992.

PITTHAN, R. O. "Tempo de Concentração de Pequenas Bacias Hidrográficas: Análise Comparativa dos Diferentes Métodos e Uso de Traçadores Radioativos", Tese de Mestrado, COPPE/UFRJ, 1973.

THEILER, J., "Estimating Fractal Dimension", Journal Opt. Soc. Am. A, vol 7 (6), 1055-1073, 1990.

TORBOTON, D. G., BRAS, R. L., RODRIGUEZ-ITURBE, I., "The Fractal Nature of River Networks", Water Resources Reserch, vol 24(8), 1317-1322, 1988.

TORBOTON D. G., BRAS R. L., RODRIGUEZ-ITURBE I., "Of the Extraction of Channel Networks from Digital Elevation Data", Hydrological Processes, vol 5, 81-100, 1991.

TRIBE, A, "Automated Recognition of Valley Lines and Drainage Networks from Grid Digital Elevation Models: a Review and a New Method", Journal of Hydrology, vol 139(1-4), 263-293, 1992.

APÊNDICE I

TABELAS DE DADOS E RESULTADOS

Coordenadas dos pontos de interpolação:

- x e y são as coordenadas relativas à posição dos pontos no plano cartesiano no qual está contida a bacia.
- z é a altitude coletada no ponto (x,y) multiplicada por 10000. Isto é necessário para que as cotas z estejam dentro do domínio [0,1].

TABELA I

i	x_i	y_i	z_i
1	0.000	0.323	0.0550
2	0.318	0.404	0.1300
3	0.273	0.495	0.0700
4	0.341	0.178	0.0734
5	0.467	0.289	0.0800
6	0.549	0.202	0.1500
7	0.457	0.551	0.1000
8	0.535	0.454	0.0850
9	0.627	0.583	0.0900
10	0.601	0.701	0.1000
11	0.840	0.543	0.1500
12	0.887	0.735	0.1200
13	0.845	0.341	0.1700
14	0.759	0.000	0.0800
15	0.955	0.777	0.1500
16	0.774	0.811	0.1250
17	0.913	0.866	0.1800
18	1.000	0.953	0.0550
19	0.900	0.890	0.1250

Coordenadas do triângulo do domínio (que contem a bacia):

- vértice 1 : (0.000,0.375,0.0550),
- vértice 2 : (0.759,0.000,0.0800),
- vértice 3 : (1.000,0.000,0.0550);

Fator de correção :

- para x e y : 95250 m
- para z : 10000 m

Conectividade dos triângulos : vértice 1, vértice 2, vértice 3 dos triângulos n correspondem aos números dos pontos na tabela anterior.

TABELA II

n	$vert1_n$	$vert2_n$	$vert3_n$
1	1	4	2
2	1	3	2
3	3	7	2
4	8	7	2
5	5	8	2
6	4	5	2
7	4	5	6
8	4	14	6
9	14	6	13
10	5	6	13
11	5	8	13
12	8	11	13
13	8	9	11
14	8	9	7
15	9	7	10
16	9	10	16
17	9	12	16
18	9	12	11
19	13	15	11
20	12	11	15
21	12	17	16
22	18	16	15
23	16	19	17
24	12	15	17
25	3	7	10
26	17	18	19

Cálculo do valor de z para determinados pontos (x,y) utilizando diferentes conjuntos de parâmetros e diferentes estimativas de z inicial.

Pontos (x,y) utilizados:

i	x_i	y_i
1	0.4	0.3
2	0.7	0.25
3	0.5	0.4

TABELAS III

Para cada conjunto (g_1,g_2,g_3) diferente foi calculado o valor de z_i para os pontos listados acima utilizando três estimativas de z_o inicial:

1º) $z_o = z_{plano}$ - o valor da cota no ponto (x_o, y_o) calculado utilizando-se a equação do plano associado ao triângulo ao qual o ponto (x_o,y_o) pertence.

2º) $z_o = 0$

3º) $z_o = z_{m\text{-}dio}$ - a média aritimética das cotas dos pontos de interpolação.

O número de iterações utilizado para calcular o valor de z_i foi designado por m.

TABELAS III.A

Parâmetros g_i= (0.1, 0.15, 0.1)

$z_o = z_{plano}$				
	z_i			
i	m = 5	m = 10	m = 15	m = 50
1	0.0579330	0.05479328	0.05479328	0.05479328
2	0.07877307	0.07877304	0.07877304	0.07877304
3	0.06218984	0.06218986	0.06218986	0.06218986

$z_o = 0$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.05479225	0.05479328	0.05479328	0.05479328
2	0.07877114	0.07877304	0.07877304	0.07877304
3	0.06218764	0.06218986	0.06218986	0.06218986

$z_o = z_{m-dio}$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.05480275	0.05479328	0.05479328	0.05479328
2	0.07879476	0.07877304	0.07877304	0.07877304
3	0.06220339	0.06218986	0.06218986	0.06218986

TABELAS III.B

Parâmetros g_i = (0.25, 0.3, 0.2)

$z_o = z_{plano}$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.05686819	0.05686692	0.05686692	0.05686692
2	0.07833729	0.07833349	0.07833349	0.07833349
3	0.06092142	0.06092382	0.06092382	0.060992382

$z_o = 0$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.05682596	0.05686689	0.05686692	0.05686692
2	0.0782601	0.07833345	0.07833349	0.07833349
3	0.06083315	0.06092375	0.06092382	0.06092382

$z_o = z_{m-dio}$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.05724596	0.05686714	0.05686692	0.056866692
2	0.0792051	0.07833392	0.07833349	0.07833349
3	0.06146315	0.06092443	0.060992382	0.06092382

TABELA III.C

Parâmetros g_i = (0.55, 0.35, 0.6)

$z_o = z_{plano}$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.06705299	0.06705516	0.06705859	0.06705783
2	0.07671999	0.07652161	0.07651614	0.07651623
3	0.0668938	0.06711503	0.06711312	0.06711540

$z_o = 0$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.06411265	0.06692712	0.06705592	0.06705783
2	0.07591099	0.07646757	0.07638181	0.07651623
3	0.06451483	0.06708879	0.06711312	0.06711540

$z_o = z_{m-dio}$				
i	z_i			
	m = 5	m = 10	m = 15	m = 50
1	0.09323251	0.06813734	0.06708527	0.06705783
2	0.08581512	0.07711471	0.07655057	0.07651623
3	0.08149339	0.06735086	0.06713350	0.06711540

TABELA IV

Tabela dos parâmetros g_i variáveis para a interpolação

i	g_i
1	0.1
2	0.1
3	0.17
4	0.18
5	0.11
6	0.1
7	0.19
8	0.42
9	0.14
10	0.23
11	0.22
12	0.37
13	0.21
14	0.1
15	0.18
16	0.1
17	0.27
18	0.3
19	0.36
20	0.29
21	0.1
22	0.1
23	0.17
24	0.1
25	0.29
26	0.16

TABELA V

Resultados do modelo mvazao para três dimensões de malha.

tempo	vazão (m^3/ segundo)		
t(hora)	malha 10x10 quad.	malha 27x25 quad.	malha 54x50 quad.
6	139.1	31.52	31.15
12	314.3	254.5	190.1
18	600.1	582.5	345.8
24	864.5	988.1	430.2
6	915.8	1183.1	405.2
12	838.3	1187.4	233.2
18	638.2	921.6	87.4
24	439.2	440.7	11.7
6	215.4	111.3	0.6
12	105.1	14.7	0.03
18	48.3	3.4	-
24	19.5	-	-
6	7.5	-	-

APÊNDICE II

PROGRAMAS DESENVOLVIDOS

1 - PARA SIMULAR O HIDROGRAMA SUPERFICIAL.

1.1 - O programa que simula o hidrograma superficial (mvazao).

1.2 - O arquivo de entrada de dados para este programa (um exemplo).

2 - PARA GERAR SUPERFÍCIE

2.1 - O programa para gerar uma superfície dado uma triângulação e plotar na tela os pontos x e y numa escala de cores variando conforme a altitude.

3 - PARA GERAR PARÂMETROS PARA O MODELO HIDROLÓGICO.

3.1 - O programa que gera os parâmetros para o modelo hidrológico e os respectivos arquivos de entrada de dados (mod_terr).

3.2 - O programa para a determinação da rede de drenagem (rede_rio).

3.3 - O programa que permite correção nas direções de drenagem obtidas do programa mod_terr (cor_dir).

1 1- PROGRAMA PARA O MODELO HIDROLÓGICO

```
PROGRAM MVAZAO;
(* ---------------------------------------
   programa para o modelo hidrologico.
   dados de entrada :
    - intervalo de tempo da simulacao,
    - numero de intervalos de tempo,
    - comprimento do lado das quadriculas,
    - numero de quadriculas pertencentes
      a bacia
    - numero de quadriculas nos eixos x e y,
    - fator de correcao de x e y em metros,
    - quadricula correspondente ao outlet,
    - percentagem do volume total que infiltra,
    - percentagem do volume total que evapora,
    - vetor de intensidade de chuva (metros /hora),
    - matriz de direcao de escoamento,
    - matriz de parametro de recessao,
    - matriz de parametros de infiltracao,
    - matriz indice para saber se as
      quadriculas pertencem ou nao a bacia.
   dados de saida:
    - vetor de vazao[t] correspondente a
      vazao do hidrograma superficial.
    - volume total de precipitacao na bacia,
    - volume retido na bacia apos simulacao
      do hidrograma,
    - volume total = volume retido + volume que saiu
  ----------------------------------------*)

uses
  crt,printer,graph,dos;
type
  str=string[15];
  vet2=array[0..50] of double;
  mat=array[0..51] of record df     : integer;
                      indice :integer;
                               ksup   :double;
```

```
                              tinf    :double;
                              alt     :double;
    end;
  blinha=array[0..56] of ^ vet2;
  clinha=array[0..56] of ^ mat;
  vet3=array[1..360] of double;
var
  arq1                    :file of mat;
  arq2,arq3,arq4          :text;
  param                   :clinha;
  vazao                   :vet3;
  chegou                  :blinha;
  vol                     :blinha;
  int_ch                  :vet3;
  nom1,nom2,nom3          :str;
  i,j,k,t,auxii,auxjj     :integer;
  auxi,auxj,ki,kj         :integer;
  nx,ny,ioutlet,joutlet   :integer;
  ot,cont,nq,nt           :integer;
  deltat,deltas           :double;
  evp,inf,q1,q2,v         :double;
  vol_inf,vol_sup,area    :double;
  chuva,fator             :double;
  soma,vol_t,p1,p2        :double;


PROCEDURE INFILTRA ( var  ti,tf :integer;tinf :double);
(*-------------------------------------------------------
 calcula as perdas por infiltracao.
 dados de entrada:
 v = percentagem do volume total de chuva que cai na bacia
 dividido pelo numero de quadriculas(entrar em milimetros),
 tinf  = parametro tempo de infiltracao.
 dado de saida:
 inf = volume que infiltra na quadricula ( em milimetros).
-------------------------------------------------------*)
 var
   aux       :integer;
```

```
 BEGIN
  aux:=trunc(tinf);
  if ((tinf<>0) and (aux>=tf)) then
  begin
   inf:=(v/sqr(tinf))*(sqr(ti)-sqr(tf))+(2*v/tinf)*(tf-ti);
  end
  else   inf:=0;
END; (* ------- fim da procedure infiltra ------*)

PROCEDURE INDEX_IJ ( var l,q,k :integer);
(* ------------------------------------------------
  calcula qual a quadricula jusante de (i,j) dada
  a direcao de escoamento.
  dados de entrada:
  l = a posicao i da malha,
  q = a posicao j da malha,
  k = parametro direcao de escoamento.
  dado de saida:
  ki = a posicao i da quad.jusante,
  kj = a posicao j da quad. jusante.
 ------------------------------------------------*)
  var
    aux1,aux2     :integer;

 BEGIN
    if (k=1) then
    begin
      aux1:=l+1;aux2:=q;
    end;
    if (k=2) then
    begin
      aux1:=l+1;aux2:=q-1;
    end;
    if (k=3) then
    begin
      aux1:=l;aux2:=q-1;
    end;
    if (k=4) then
    begin
```

```
      aux1:=l-1;aux2:=q-1;
    end;
    if (k=5) then
    begin
      aux1:=l-1;aux2:=q;
    end;
    if (k=6) then
    begin
      aux1:=l-1;aux2:=q+1;
    end;
    if (k=7) then
    begin
      aux1:=l;aux2:=q+1;
    end;
    if (k=8) then
    begin
      aux1:=l+1;aux2:=q+1;
    end;
    if (k=0) then
    begin
      aux1:=l;aux2:=q;
    end;
    ki:=aux1;kj:=aux2;
 END; (* ----- fim da procedure index_ij ------*)

BEGIN
  clrscr;
  writeln(' entre com o nome do arquivo de dados:');
  readln(nom1);
  writeln( ' entre com o arquivo de parametros :');
  readln(nom3);
  writeln(' entre com o nome do arquivo de saida :');
  readln(nom2);
  assign(arq2,nom1);
  assign(arq3,nom2);
  reset(arq2);
  rewrite(arq3);
```

```
(*-------------------------------------------
le os dados de entrada
a intensidade esta em metros/hora
 -------------------------------------------*)
for i:=1 to 2 do readln(arq2);
readln(arq2);
readln(arq2);
readln(arq2,deltat);
readln(arq2);
readln(arq2,nt);
readln(arq2);
readln(arq2,deltas);
readln(arq2);
readln(arq2,nq);
readln(arq2);
readln(arq2,nx,ny);
readln(arq2);
readln(arq2,fator);
readln(arq2);
readln(arq2,ioutlet,joutlet);
readln(arq2);
readln(arq2,p1);
readln(arq2);
readln(arq2,p2);
for i:=1 to 2 do readln(arq2);
cont:=0;
soma:=0;
for i:=1 to nt do
begin
  readln(arq2,int_ch[i]);
  soma:=soma+int_ch[i];
  if (int_ch[i]<>0) then cont:=cont+1;
end;


{ v= volume total*percentual/numero de quad}
(* ----------------------------------------
    abre as matrizes de ponteiros
 ----------------------------------------*)
```

```
assign(arq1,nom3);
reset(arq1);

for i:=0 to ny do
begin
  NEW(vol[i]);
  NEW(chegou[i]);
end;
for i:=1 to ny do
begin
  NEW(param[i]);
end;
(* ----------------------------------------
    le a matriz de parametros e zera as
    matrizes de trabalho
  ----------------------------------------*)
for i:=1 to ny do
begin
  read(arq1,param[i]^);
end;
for i:=0 to ny do
begin
  for j:=0 to nx do
  begin
    vol[i]^[j]:=0;
    chegou[i]^[j]:=0;
  end;
end;

(*-----------------------------------------------
   calcula a area das quadriculas em metros,
   calcula o volume total da precipitacao,
   calcula o volume que infiltra em cada quadriculas,
   calcula o volume que evapora em cada quadricula.
-----------------------------------------------*)
area:=sqr(fator)*sqr(deltas);
vol_t:=soma*deltat*area*nq;
v:=(vol_t*p1)/nq;
evp:=(vol_t*p2)/(cont*nq);
```

```
 (*-----------------------------------------------------
   loop que calcula a vazao em cada passo de tempo t
 -----------------------------------------------------*)

  for t:=1 to nt do
  begin

 (* -----------------------------------------------------
    calcula a precipitacao na quadricula, retira as perdas
    por evaporacao,retira as perdas por infiltracao,
    acrescenta o volume que saiu no tempo anterior a cada
    quadricula ( matriz chegou) e calcula o volume
    superficial.
  -----------------------------------------------------*)
     soma:=0;
     for i:=1 to ny do
     begin
for j:=1 to nx do
begin
  if (param[i]^[j].indice<>0) then
        begin
    chuva:=(int_ch[t]*deltat*area);
    if ( evp<=chuva ) then  vol_inf:=chuva-evp
               else  vol_inf:=0;
          ot:=t-1;
          infiltra(ot,t,param[i]^[j].tinf);
    if ( inf<= vol_inf ) then vol_sup:=vol_inf-inf
                  else vol_sup:=0;
          vol[i]^[j]:=vol[i]^[j]+vol_sup+chegou[i]^[j];
    chegou[i]^[j]:=0;
          soma:=soma+vol_sup;
        end;

end; (* ------ fim do for j ---------*)
     end;  (* ------- fim do for i  ----------*)
 (* -----------------------------------------------
    calcula a parcela que escoa de cada quadricula
    e a acrescenta a uma matriz auxiliar chegou
  -----------------------------------------------*)
```

```
      for i:=1 to ny do
      begin
        for j:=1 to nx do
        begin
          if (param[i]^[j].indice<>0) then
          begin
            index_ij (i,j,param[i]^[j].df);
            auxi:=ki;
            auxj:=kj;
            if ( param[i]^[j].ksup<=1 ) then
            begin
              q1:=vol[i]^[j]*param[i]^[j].ksup;
              chegou[auxi]^[auxj]:=chegou[auxi]^[auxj]+q1;
              vol[i]^[j]:=vol[i]^[j]-q1;
            end
            else
            begin
              q1:=vol[i]^[j]*(param[i]^[j].ksup - 1);
              q2:=vol[i]^[j]-q1;
              index_ij (auxi,auxj,param[auxi]^[auxj].df);
              auxii:=ki;
              auxjj:=kj;
              chegou[auxi]^[auxj]:=chegou[auxi]^[auxj]+q2;
              chegou[auxii]^[auxjj]:=chegou[auxii]^[auxjj]+q1;
              vol[i]^[j]:=0;
            end;
          end; (*------ fim do if indice ------------*)
        end; (* ------ fim do for j -------*)
      end;  (* ------ fim do for i -------*)
  (* ------------------------------------------------
     calcula a vazao neste poasso de tempo:
    vazao[t] = vazao no intervalo t em metros cubicos
   ------------------------------------------------*)
     vazao[t]:=chegou[ioutlet]^[joutlet];
     chegou[ioutlet]^[joutlet]:=0;
   end; (* ---- fim do for t ------------*)
(*-------------------------------------
  impressao dos resultados
  -------------------------------------*)
```

```
writeln(arq3,' RESULTADOS DO PROGRAMA MVAZAO');
for i:=1 to 2 do writeln(arq3);
writeln(arq3,' caso : ',nom3:15);
writeln(arq3,' numero de quadriculas da bacia : ',nq:6);
writeln(arq3,' intervalo de tempo :',deltat:5:2,'horas');
writeln(arq3,' area de cada quadricula da bacia :
               ',area:10:3,' m2 ');
writeln(arq3);
writeln(arq3,' volume total de chuva durante o evento:
               ',vol_t:10:3);
writeln(arq3);
writeln(' vetor de vazoes em metros cubicos');
writeln(arq3,' vetor de vazoes em metros cubicos');
writeln(arq3,' vazao no intervalo de tempo t=1,2,..nt ');
for t:=1 to nt do
begin
  writeln(arq3,' vazao[',t,'] = ',vazao[t]:7:4);
  writeln(vazao[t]:8:4);
end;

soma:=0;
for i:=1 to ny do
begin
  for j:=1 to nx do
  begin
    soma:=soma+vol[i]^[j];
  end;
end;

for i:=1 to ny do
begin
  for j:=1 to nx do
  begin
    soma:=soma+chegou[i]^[j];
  end;
end;
writeln(arq3);
writeln(arq3,'volume de agua que ficou retida na bacia');
writeln(arq3,'  no ultimo intervalo de tempo :
```

```
                ',soma:15:7,' metros cubicos

  for i:=1 to nt do
  begin
    soma:=soma+vazao[i];
  end;
  writeln(' volume total= ',soma:10:3);
  writeln(arq3);
  writeln(arq3,' volume que ficou mais o que saiu em
                  forma de vazao:');
  writeln(arq3,' volume total descontadas as perdas :
      ',soma:15:7,' metros cubi  repeat until keypressed;
  (* ----------------------------------------
      libera as matrizes de ponteiros
  ----------------------------------------*)
  for i:=0 to ny do
  begin
    DISPOSE(vol[i]);
    DISPOSE(chegou[i]);
  end;
  for i:=1 to ny do
  begin
    DISPOSE (param[i]);
  end;
  close(arq1);
  close(arq2);
  close(arq3);
END.
```

1.2 - ARQUIVO DE ENTRADA DE DADOS EM TEXTO PARA O MODELO CHUVA-VAZÃO.

É necessário também um arquivo em binário com os parâmetros fisiográficos gerado pelo programa mod_terr.

```
arquivo de dados para o program vazao
caso de exemplo
'p31.bin'          (* arquivo de parametros  *)
intervalo de tempo
0.77    (* deltat  em horas  *)
numero de intervalos de tempo
80    (*  nt  *)
comprimento do lado da quad em x,y
0.026247    (*  deltas   *)
numero de quadriculas pertencentes a bacia
304     (*  nq   *)
numero de quadriculas no eixo x e eixo y
25 27    (*  nx,ny  *)
fator de correcao do plano x,y em metros
95250     (*  fator  *)
quadricula correspondente ao outlet
2  2    (*  ioutlet, joutlet *)
percentagem do volume total que infiltra
0.25   (*  p1  *)
percentagem do volume que evapora
0.1    (*   p2   *)
intensidade de chuva nos intervalos de tempo
formato : int_ch[i]; i = 1,2...nt  em metros/horas
0.004
0.004
0.004
0.004
.
.
.
.
```

2.1 - PROGRAMA PARA GERAR ALTITUDES

```
PROGRAM GERA3DRD;

{*********************************************************
 PROGAMA PARA GERAR ALTITUDES  RANDÔMICAMENTE
 ESTE PROGRAMA NECESSITA DE UMA TPU E DE UM ARQUIVO BGI
 ESPECIAIS QUE O PERMITE TRABALHAR COM 256 CORES.
 ESTES ARQUIVOS SÃO : SVGA254.BGI E ESC_COR.TPU.
*********************************************************}

  USES
  CRT,PRINTER, DOS,GRAPH,ESC_COR;
  TYPE
   st2=string[15];
   st1=string[4];
   MAT1=array[1..70,1..3] of double;
   mat2=array[1..70,1..10] of double;
   mat3=array[1..70,1..6] of double;
   mat5=array[0..40,0..40] of DOUBLE;
   mat7=array[1..3,1..4] of double;
   MAT6=ARRAY[1..70,1..9] OF double;
   VET=array[1..20] of integer;
   VET1=ARRAY[1..3] OF double;
   VET2=ARRAY[1..70] OF double;
     vet3=array[1..6] of double;
  CONST
    N=3;  {dimensao da matriz dos triangulos do dominio}
  ITERA=10; {numero de iteracoes da IFS}
  UserCharsize=0;

  VAR
  J,K,L,Q,T,j1,DELTA,CONTADOR,M,COR,CORINF      :INTEGER;
  I                                             :longint;
  TETA,TETA1,TETA2,K1,T1,sab,iaux,FX,FY,D       :INTEGER;
  nom1                                          :st2;
  no1,no2                                       :st1;
  dc,xii,p                                      :integer;
```

```
corsup,intcor,DX,DY                                    :integer;
intz,zsup,zinf                                         :double;
VZ,AUX1,AUX2,AUX3,FAT,AUX,soma,U                       :DOUBLE;
VALOR1,VALOR2,VALOR3,xobs,yobs,A1,A2                   :DOUBLE;
xi,yf,XF,YI                                            :double;
zoom                                                   :integer;
G,GAUX                                                 :MAT7;
COEF                                                   :MAT1;
VETDELTA                                               :VET;
IFS                                                    :MAT2;
PX,PY,PZ                                               :MAT5;
X                                                      :MAT6;
 XO,YO,ZO                                              :VET1;
GN,detx                                                :VET2;
v                                                      :vet3;
ARQ1,ARQ2,arq3,ARQ4                                    :TEXT;

 BEGIN
 CLRSCR;
 RANDOMIZE;
 writeln(' entre com o arquivo de entrada');
 readln(nom1);
 ASSIGN(ARQ1,nom1);
 RESET(ARQ1);
 writeln(' entre com o fator zoom');
 readln(zoom);
 if (zoom>1) then
 begin
 writeln(' entre com o ponto inicial (xi,yi)');
 readln(xi,yi);
 writeln(' entre com o ponto final (xf,yf)');
 readln(xf,yf);
 end;
 writeln(' entre com os valores de zinf e zsup');
   readln(zinf,zsup);

   (*-----------------------------------------------
         LE OS ARQUIVOS DE ENTRADA
    -----------------------------------------------*)
```

```
 FOR I:=1 TO 2 DO READLN(ARQ1);
 READLN(ARQ1,M);
 FOR I:=1 TO M DO
 BEGIN
FOR J:=1 TO 9 DO
  BEGIN
    READ(ARQ1,X[I,J]);
  END;

  READLN(ARQ1);
END;
   READLN(ARQ1);
   FOR I:=1 TO 3 DO
     BEGIN
READ(ARQ1,XO[I],YO[I],ZO[I]);
 END;
 for i:=1 to 2 do readln(arq1);
 FOR I:=1 TO M DO
 BEGIN
  READln(arq1,GN[I]);
 END;


  (*-----------------------------------------------------
 calcula a IFS  para cada triangulo da regiao
 -----------------------------------------------------*)
 FOR L:=1 TO M DO
 BEGIN
TETA:=0;
K:=1;
FOR Q:=1 TO 2 DO
(*-----------------------------------------------------
  calcula a,b,u quando q=1 ; c,d,v quando q=2
 -----------------------------------------------------*)
  BEGIN
     FOR I:=1 TO 3 DO
      BEGIN
       G[I,1]:=XO[I];
       G[I,2]:=YO[I];
```

```
      G[I,3]:=1;
     END;
   G[1,4]:=X[L,K];
   G[2,4]:=X[L,K+3];
   G[3,4]:=X[L,K+6];

   FOR T:=1 TO N DO
      BEGIN
K1:=T;
AUX:=G[T,T];
FOR T1:=T+1 TO N DO
  BEGIN
    IF (G[T1,T]>AUX) THEN
         BEGIN
    AUX:=G[T1,T];
    K1:=T1;
         END;
  END;

FOR J:=1 TO N+1 DO
  BEGIN
    GAUX[T,J]:=G[T,J];
    G[T,J]:=G[K1,J];
    G[K1,J]:=GAUX[T,J];
  END;
 FAT:=G[T,T];
 FOR J:=1 TO N+1 DO
   BEGIN
     G[T,J]:=G[T,J]/FAT;
   END;
 FOR I:=1 TO N DO
   BEGIN
     IF(I<>T) THEN
       BEGIN
  FAT:=G[I,T];
  FOR J:=1 TO N+1 DO
    BEGIN
      G[I,J]:=G[I,J]-G[T,J]*FAT;
    END;
```

```
   END;
      END;
        END;

     FOR I:=1 TO 3 DO
       BEGIN
  IFS[L,I+TETA]:=G[I,4];
       END;

     K:=K+1;
     TETA:=TETA+3;
   END; { fim do for q}

 END; { fim do for L }

{ for i:=1 to m do
 begin
   for j:=1 to 6 do
   begin
    write(ifs[i,j]:10:3);
  end;
  writeln;
  readln;
  end;
  readln;  }
  (*-----------------------------------------------------
 calcula os valores de:e,f,t para determinado parametro g
      -----------------------------------------------------*)
      FOR L:=1 TO M DO
 BEGIN
   FOR I:=1 TO 3 DO
    BEGIN
      G[I,1]:=XO[I];
      G[I,2]:=YO[I];
      G[I,3]:=1;
    END;
   FOR J:=1 TO 3 DO
     BEGIN
       G[J,4]:=X[L,(J+2*J)]-GN[L]*ZO[J];
```

```
   END;

 FOR T:=1 TO N DO
   BEGIN
K1:=T;
AUX:=G[T,T];
FOR T1:=T+1 TO N DO
  BEGIN
    IF (G[T1,T]>AUX) THEN
         BEGIN
    AUX:=G[T1,T];
    K1:=T1;
         END;
  END;

FOR J:=1 TO N+1 DO
  BEGIN
    GAUX[T,J]:=G[T,J];
    G[T,J]:=G[K1,J];
    G[K1,J]:=GAUX[T,J];
  END;
     FAT:=G[T,T];
     FOR J:=1 TO N+1 DO
BEGIN
  G[T,J]:=G[T,J]/FAT;
END;
     FOR I:=1 TO N DO
BEGIN
  IF(I<>T) THEN
    BEGIN
      FAT:=G[I,T];
      FOR J:=1 TO N+1 DO
 BEGIN
   G[I,J]:=G[I,J]-G[T,J]*FAT;
 END;
    END;
 END;
   END;
```

```
  FOR I:=1 TO 3 DO
    BEGIN
      IFS[L,I+6]:=G[I,4];
    END;
  IFS[L,10]:=GN[L];
END;  { fim do segundo  for L }
(*----------------------------------------
  para fazer zoom
 ----------------------------------------*)
 if(zoom>1) then
 begin
  for i:=1 to m do
  begin
  ifs[i,3]:=ifs[i,3]*zoom;
  ifs[i,6]:=ifs[i,6]*zoom;
  ifs[i,9]:=ifs[i,9]*zoom;
  end;
  FX:=1;
  FY:=1;
  DY:=trunc(yf*zoom);
  DX:=trunc(xi*zoom);
 end
 else
 begin
 fx:=450;
 fy:=450;
 dy:=500;
 dx:=100;
 end;

  CORINF:=50;
  corsup:=250;
  intz:=zsup-zinf;
  intcor:=corsup-corinf;
  AUX1:=0;
  AUX2:=0;
  AUX3:=0;
  INIT(2);
  ESCALAARCO(100,100);
```

```
 line(0+dx,trunc(dy-0*fy),0+dx,trunc(dy-1*fy));
line(0+dx,trunc(dy-0*fy),trunc(1*fx+dx),trunc(dy-0*fy));

xii:=200;
dc:=getmaxx div 450;
for cor:=corinf to corsup do
begin
setFillStyle(solidFill,cor);
bar(xii,80,xii+dc,100);
inc(xii,dc);
end;

 FOR I:=1 TO 60000 DO
 BEGIN
  U:=RANDOM;
  FOR J:=1 TO M DO
   BEGIN
  A1:=J*(1/M)-1/M;
  A2:=1/M+(J-1)*1/M;
  IF (( U> A1) AND (U<=A2)) THEN K:=J;
   END;
  VALOR1:=IFS[K,1]*AUX1+IFS[K,2]*AUX2+IFS[K,3];
  VALOR2:=IFS[K,4]*AUX1+IFS[K,5]*AUX2+IFS[K,6];
  VALOR3:=IFS[K,7]*AUX1+IFS[K,8]*AUX2
  +IFS[K,10]*AUX3+IFS[K,9];
  AUX1:=VALOR1;
  AUX2:=VALOR2;
  AUX3:=VALOR3;
  IF (ZOOM=1) THEN AUX:=AUX3
  ELSE AUX:=AUX3/ZOOM;

 IF (I>10) THEN
 BEGIN
  IF (ZOOM>1) THEN
  BEGIN
 IF((AUX1<=XF*ZOOM) AND (AUX2>=YI*ZOOM)) THEN
 BEGIN
  COR:=TRUNC(corinf+(intcor*(aux-zinf))/intz);
  PUTPIXEL(TRUNC(AUX1*FX+DX),TRUNC(DY-AUX2*FY),COR);
```

```
 END;
  END
  ELSE
  BEGIN
  COR:=TRUNC(corinf+(intcor*(aux-zinf))/intz);
  if (cor>0) then begin
  PUTPIXEL(TRUNC(AUX1*FX+DX),TRUNC(DY-AUX2*FY),COR);
   end;
  END;
 END;
END;
str(zinf:4:2,no1);
str(zsup:4:2,no2);
settextstyle(1,0,4);
outtextxy(190,70,'z=');
outtextxy(210,70,no1);
outtextxy(360,70,'z=');
outtextxy(380,70,no2);
outtextxy(dx-10,dy+10,'0');
outtextxy(dx-15,dy-fy,'1');
outtextxy(fx+dx,dy+15,'1');
if (nom1='parait1.dat') then
outtextxy(250,550,'caso 1 ');
if (nom1='parait2.dat') then
outtextxy(200,550,'caso 2 ');
if (nom1='parait3.dat') then
outtextxy(200,550,'caso 3 ');
write(chr(7));
  repeat until keypressed;
CLOSE(ARQ1);
CLOSEGRAPH;

  END.
```

3.1 - PARA GERAR PARÂMETROS PARA O MODELO HIDROLÓGICO

```
PROGRAM MOD_TER2;

(*********************************************************
  PROGRAMA PARA GERAR PARAMETROS PARA O MODELO HIDROLOGICO
  DADOS DE ENTRADA:
  - MATRIZ DE PONTOS DE TRIANGULACAO,
  - VERTICES DO TRIANGULO DOMINIO,
  - PARAMETROS LIVRES G = { IFS[I,10]},
  - MATRIZ PONT { PARA USO NA PROCEDURE LOCALIZA },
 - MATRIZ INDICE : DETERMINA SE A QUADRICULA (I,J) PERTENCE
    A BACIA.
  - VARIAVEIS : T_MAX { VALOR MAXIMO DO PARAMETRO T_INF },
            ZMEDIO {MEDIA DOS VALORES DAS COTAS DOS PONTOS
                    DE INTERPOLACAO },
          X0,Y0 { COORDENADAS DO PONTO INICIAL DA MALHA }
        DS  { COMPRIMENTO DO LADO DA QUADRICULA EM METROS
              CONVERTIDOS PARA O PLANO CARTESIANO X,Y },
            VM { VELOCIDADE MEDIA DA BACIA }
          NX { NUMERO DE DIVISOES EM J ( EIXO DOS X ) }
          NY { NUMERO DE DIVISOES EM I ( EIXO DOS Y ) }
       MED_ANG { ANGULO MEDIO DE DECLIVIDADE DA BACIA }

  DADOS DE SAIDA:
- ARQUIVO PARAMET.BIN { ARQUIVO DE PARAMETROS PARA O MODELO
                        HIDROLOGICO }
- ARQUIVO REDE.BIN  { ARQUIVO DE DADOS PARA O PROGRAMA
  REDE_RIO}
- ARQUIVO MOD_TERR.DAT { ARQUIVO DE SAIDA EM ASCII }

  *********************************************************)

  USES
    CRT,PRINTER, DOS,MATH;
  TYPE
  STR= STRING[15];
  MAT1=ARRAY[0..51] OF RECORD  DF     :INTEGER;
           INDICE :INTEGER;
```

```
          KSUP    :DOUBLE;
          TINF    :DOUBLE;
          AT      :DOUBLE;
         END;
 MATRIZ=ARRAY[0..56] OF ^ MAT1;
 MAT2=ARRAY[1..50,1..10] OF DOUBLE;
 VET1=ARRAY[1..50] OF DOUBLE;
 VET2=ARRAY[1..3] OF DOUBLE;
 MAT3=ARRAY[1..50,1..9] OF DOUBLE;
 MAT4=ARRAY[1..50,1..4] OF INTEGER;
 MAT6=ARRAY[1..55,1..50] OF INTEGER;
 MAT7=ARRAY[1..55,1..50] OF DOUBLE;
 VAR
 I,J,NX,NY,NM,DESC,NQ          :INTEGER;
 MED_G,MED_ANG,VM,DS           :DOUBLE;
 T_MAX,ALT_MED,INF,RG          :DOUBLE;
 X0,Y0,ZMEDIO,XX,YY,ANG        :DOUBLE;
 GRAD1,GRAD2,KS,SOMA           :DOUBLE;
 NOM1,NOM2,NOM3,NOM4,NOM5      :STR;
 PARAM                         :MATRIZ;
 IFS,INV                       :MAT2;
 X                             :MAT3;
 PONT                          :MAT4;
 PERT                          :MAT6;
 AGL,RUG                       :MAT7;
 DETX                          :VET1;
 XO,YO,ZO                      :VET2;
 ARQ1,ARQ2,ARQ4,arq7                :TEXT;
 ARQ3                          :FILE OF MAT1;

PROCEDURE PARAMETROS ( VAR M :INTEGER);
(* ------------------------------------------------------
  - CALCULA OS PARAMETROS PARA O ALGORITMO DE INTERPOLACAO
    FRACTAL : IFS[I,J], I: INDICE DOS TRIANGULOS DA
 INTERPOLACAO, J=1,2,..10  PARAMETROS PARA CADA UNIDADE DA
    TRIANGULACAO.
  - CALCULA O VETOR DETX : DETERMINANTE DOS TRIANGULOS DA
  INTERPOLACAO PARA SER UTILIZADO NA PROCEDURE LOCALIZA
 ------------------------------------------------------*)
```

```
TYPE
  MAT1=ARRAY[1..3,1..4] OF DOUBLE;
  MAT2=ARRAY[1..2,1..4] OF DOUBLE;
  VET=ARRAY[1..6] OF DOUBLE;
VAR
  L,I,J,K,T,K1,T1,I1,Q    :INTEGER;
  TETA,TETA1,TETA2,IAUX   :INTEGER;
  AUX,FAT,SOMA            :DOUBLE;
  G,GAUX                  :MAT1;
  B,BAUX                  :MAT2;
  V                       :VET;
BEGIN
 (*-----------------------------------------------
  calcula a IFS somente de x e y e a inversa para cada
     triangulo da regiao
  -----------------------------------------------*)
  FOR L:=1 TO M DO
    BEGIN
      TETA:=0;
      K:=1;
      FOR Q:=1 TO 2 DO
      (*-----------------------------------------
        calcula a,b,u quando q=1 ; c,d,v quando q=2
       -----------------------------------------*)
        BEGIN
           FOR I:=1 TO 3 DO
            BEGIN
             G[I,1]:=XO[I];
             G[I,2]:=YO[I];
             G[I,3]:=1;
            END;
          G[1,4]:=X[L,K];
          G[2,4]:=X[L,K+3];
          G[3,4]:=X[L,K+6];

          FOR T:=1 TO 3 DO
             BEGIN
              K1:=T;
```

```
      AUX:=G[T,T];
      FOR T1:=T+1 TO 3 DO
        BEGIN
          IF (G[T1,T]>AUX) THEN
                                BEGIN
                                  AUX:=G[T1,T];
                                  K1:=T1;
                                END;
        END;

      FOR J:=1 TO 4 DO
        BEGIN
          GAUX[T,J]:=G[T,J];
          G[T,J]:=G[K1,J];
          G[K1,J]:=GAUX[T,J];
        END;
       FAT:=G[T,T];
       FOR J:=1 TO 4 DO
         BEGIN
           G[T,J]:=G[T,J]/FAT;
         END;
       FOR I:=1 TO 3 DO
         BEGIN
           IF(I<>T) THEN
             BEGIN
               FAT:=G[I,T];
               FOR J:=1 TO 4 DO
                 BEGIN
                   G[I,J]:=G[I,J]-G[T,J]*FAT;
                 END;
             END;
         END;
    END;

FOR I:=1 TO 3 DO
  BEGIN
    IFS[L,I+TETA]:=G[I,4];
  END;
```

```
    FOR I:=1 TO 2 DO
      BEGIN
        B[K,I]:=G[I,4];
      END;

    K:=K+1;
    TETA:=TETA+3;
  END; { fim do for q}

(*-----------------------------------------
    calcula a inversa de a b c d
-----------------------------------------*)
FOR  I:=1 TO 2 DO
  BEGIN
    FOR J:=3 TO 4 DO
      BEGIN
        IF (I+J=5) THEN B[I,J]:=0
                   ELSE B[I,J]:=1;
      END;
  END;

FOR T:=1 TO 2 DO
  BEGIN
        K1:=T;
        AUX:=B[T,T];
        FOR T1:=T+1 TO 2 DO
          BEGIN
            IF (B[T1,T]>AUX) THEN
                                BEGIN
                                  AUX:=B[T1,T];
                                  K1:=T1;
                                END;
          END;

        FOR J:=1 TO 4 DO
          BEGIN
            BAUX[T,J]:=B[T,J];
            B[T,J]:=B[K1,J];
            B[K1,J]:=BAUX[T,J];
```

```
                END;
            FAT:=B[T,T];
            FOR J:=1 TO 4 DO
              BEGIN
                B[T,J]:=B[T,J]/FAT;
              END;
            FOR I:=1 TO 2 DO
              BEGIN
                IF(I<>T) THEN
                  BEGIN
                    FAT:=B[I,T];
                    FOR J:=1 TO 4 DO
                      BEGIN
                        B[I,J]:=B[I,J]-B[T,J]*FAT;
                      END;
                  END;
              END;
          END;

        TETA:=0;
        FOR I:=1 TO 2 DO
        (*-----------------------------------------
       armazena a inversa da matriz abcd como inv[i,j]
        -----------------------------------------*)
          BEGIN
            FOR J:=1 TO 2 DO
              BEGIN
                INV[L,J+TETA]:=B[I,J+2];
              END;
            TETA:=TETA+3;
          END;

        FOR I:=1 TO 2 DO
  (*-----------------------------------------------
multiplica a inversa por u,v apar obter inversa de u e v
 -----------------------------------------------*)
   BEGIN
            TETA:=0;
            AUX:=0;
```

```
            FOR J:=3 TO 4 DO
              BEGIN
                AUX:=AUX+B[I,J]*IFS[L,J+TETA];
                TETA:=TETA+2;
              END;
            INV[L,(I+I*2)]:=AUX;
          END;
      END; { fim do for L }

(*-----------------------------------------------------
calcula os valores de:e,f,t para determinado parametro g
-----------------------------------------------------*)
      FOR L:=1 TO M DO
        BEGIN
          FOR I:=1 TO 3 DO
           BEGIN
             G[I,1]:=XO[I];
             G[I,2]:=YO[I];
             G[I,3]:=1;
           END;
          FOR J:=1 TO 3 DO
            BEGIN
              G[J,4]:=X[L,(J+2*J)]-IFS[L,10]*ZO[J];
            END;

          FOR T:=1 TO 3 DO
            BEGIN
                K1:=T;
                AUX:=G[T,T];
                FOR T1:=T+1 TO 3 DO
                  BEGIN
                    IF (G[T1,T]>AUX) THEN
                                        BEGIN
                                          AUX:=G[T1,T];
                                          K1:=T1;
                                        END;
                  END;

                FOR J:=1 TO 4 DO
```

```
                BEGIN
                  GAUX[T,J]:=G[T,J];
                  G[T,J]:=G[K1,J];
                  G[K1,J]:=GAUX[T,J];
                END;
              FAT:=G[T,T];
              FOR J:=1 TO 4 DO
                BEGIN
                  G[T,J]:=G[T,J]/FAT;
                END;
              FOR I:=1 TO 3 DO
                BEGIN
                  IF(I<>T) THEN
                    BEGIN
                      FAT:=G[I,T];
                      FOR J:=1 TO 4 DO
                        BEGIN
                          G[I,J]:=G[I,J]-G[T,J]*FAT;
                        END;
                    END;
                END;
            END;

          FOR I:=1 TO 3 DO
            BEGIN
              IFS[L,I+6]:=G[I,4];
            END;
        END;  { fim do segundo  for L }

 (*-----------------------------------------------------
  calcula o vetor detx : determinante dos triangulos da
 triangulacao, para serem utilizados em localiza.
-----------------------------------------------------*)
        for i:=1 to m do
          begin
(*-- monta a matriz do triangulo i  ------------------*)

          teta:=0;
          for j:=1 to 3 do
```

```
            begin
              G[1,j]:=x[i,j+teta];
              G[2,j]:=x[i,j+teta+1];
              G[3,j]:=1;
              teta:=teta+2;
            end;
          (*-------------------------*)
(*----- calcula o determinante da matriz G -------------*)

              v[1]:=G[1,1]*G[2,2];
              v[2]:=G[1,3]*G[2,1];
              v[3]:=G[1,2]*G[2,3];
              v[4]:=-G[2,2]*G[1,3];
              v[5]:=-G[1,1]*G[2,3];
              v[6]:=-G[1,2]*G[2,1];

(*---- otimiza a soma dos termos do determinante -------*)

              iaux:=1;
              for l:=1 to 4 do
                begin
                  repeat
                    teta:=0;
                    for j:=iaux to 5 do
                      begin
                      if ( abs(v[j+1]) < abs(v[j]) )  then
                          begin
                            teta:=1;
                            aux:=v[j];
                            v[j]:=v[j+1];
                            v[j+1]:=aux;
                          end;
                      end;
                  until ( teta=0 );
                  soma:=v[iaux]+v[iaux+1];
                  v[iaux+1]:=soma;
                  iaux:=iaux+1;
                end;    { fim do for l }
              detx[i]:=v[5]+v[6];
```

```
(*------- fim do calculo do determinante de Ai1 -------*)
            end;    { fim do for i }

      END; (*---- FIM DA PROCEDURE PARAMETROS -----*)

PROCEDURE GRADIENTE ( var  xi,yi  :double);
(* ------------------------------------------------
   calcula o vetor de maior declividade na quadricula
   calcula a altitude media da quadricula para o
   programa rede de rios.
   dados de entrada:
    - (xi,yi) = coordenadas do primeiro vertice esquerdo
      da quadricula,
    - ds = comprimento do lado da quadricula (metros),
    dados de saida:
    - (grad1,grad2) = vetor direcao de maior declividade
    - alt_med = altitude media da quadricula.
 ------------------------------------------------*)

  type
    vet=array[1..9] of double;
  var
    i,j,cont,delta,c          :integer;
    dx,soma,bx,by,pz,sm       :double;
    x1,y,z,zz,px,py           :vet;

 PROCEDURE LOCALIZA ( var vx,vy :double);
 (*---- localiza em qual triangulo o ponto vx,vy esta---
 utiliza : matriz de vertices dos triangulos X,
    determinante de cada um desses triangulos detx,
    numero de triangulos m,
 retorna : a variavel inteira delta ,que representa o
    triangulo ao qual o ponot pertence.
   ------------------------------------------------*)
   type
     mat1=array[1..3,1..3] of double;
     vet1=array[1..6] of double;
     vet2=array[1..3] of double;
   var
```

```
    i,j,i1,l,indice,iaux,jaux,teta,troc,p,k1,j1 :integer;
    v                                           :vet1;
    a,g                                         :mat1;
    det,lam                                     :vet2;
    soma,aux                                    :double;
  BEGIN
    jaux:=0;
    indice:=0;
    i:=1;
   if (vx<=0.5 ) then
   begin
     if (vy<=0.5) then p:=1
    else p:=2;
    end
   else
     begin
       if (vy<=0.5 ) then p:=3
       else p:=4;
     end;


    while ( (indice=0 ) and (i<=nm) ) do
      begin
for k1:=1 to 4 do
  begin
   if (pont[i,k1]=p) then
    begin
(*------ monta a matriz do triangulo i  ----------------*)

teta:=0;
for j:=1 to 3 do
  begin
    G[1,j]:=x[i,j+teta];
    G[2,j]:=x[i,j+teta+1];
    G[3,j]:=1;
    teta:=teta+2;
  end;   {* End - J *}
(*-------------------------------------------------------*)
```

```
for i1:=1 to 3 do
  begin
   (*--------- monta a matriz A1 ---------------------*)
    for j:=1 to 3 do
      begin
 if (j<>i1) then
   begin
     for j1:=1 to 3 do
       begin
  A[j1,j]:=G[j1,j];
       end; {* End - J1 *}
   end; {* End - IF *}
      end; {* End - J *}
    A[1,i1]:=vx;
    A[2,i1]:=vy;
    A[3,i1]:=1;
   (*-------------------------------------------------*)
    (*------- calcula o determinante da matriz Ai1 --*)

    v[1]:=A[1,1]*A[2,2];
    v[2]:=A[1,3]*A[2,1];
    v[3]:=A[1,2]*A[2,3];
    v[4]:=-A[2,2]*A[1,3];
    v[5]:=-A[1,1]*A[2,3];
    v[6]:=-A[1,2]*A[2,1];

(*---- otimiza a soma dos termos do determinante ---*)

    iaux:=1;
    for l:=1 to 4 do
      begin
 repeat
   troc:=0;
   for j:=iaux to 5 do
     begin
       if ( abs(v[j+1]) < abs(v[j]) )  then
  begin
    troc:=1;
    aux:=v[j];
```

```
    v[j]:=v[j+1];
    v[j+1]:=aux;
  end; {* End - If *}
     end; {* End - J *}
 until ( troc=0 );
 soma:=v[iaux]+v[iaux+1];
 v[iaux+1]:=soma;
 iaux:=iaux+1;
      end;    {* End - L*}
    det[i1]:=v[5]+v[6];
    (*------- fim do calculo do determinante de Ai1 -*)
  end;   {* End - I1 *}

(*---------- calculo dos lambidas--------------------*)
 soma:=0;
 for j:=1 to 3 do
   begin
     lam[j]:=det[j]/detx[i];
     soma:=soma+abs(lam[j]);
   end; {* End - J *}
 (*-------------------------------------------------*)
 if (soma<=1.000000001 ) then
    indice:=1
 else
    indice:=0;
 if (indice=0) then
    delta:=0
 else
    delta:=i;
end;  {* End If - pont=p *}
     end;   {* fim do for k1 *}
 i:=i+1;
     end;    {* fim do while *}
 END;  (*----- FIM DA PROCEDURE LOCALIZA-----*)



PROCEDURE ALTITUDE ( VAR X,Y : DOUBLE);
```

```
(* ------------------------------------------------
   CALCULA A ALTITUDE NO PONTO X,Y UTILIZANDO SISTEMA
   DINAMICO DE SHIFT RANDOMICO
 ------------------------------------------------*)
TYPE
  VET=ARRAY[1..15] OF INTEGER;
VAR
  J,K,CONTADOR          :INTEGER;
  AUX1,AUX2,AUX3        :DOUBLE;
  VALOR1,VALOR2,VALOR3  :DOUBLE;
  VETDELTA              :VET;
        ITERA                   :INTEGER;

    BEGIN
  LOCALIZA(X,Y);
        ITERA:=10;
  IF (DELTA=0) THEN PZ:=0
  ELSE
        BEGIN
      VETDELTA[1]:=DELTA;
      CONTADOR:=1;
      J:=1;
      AUX1:=X;
      AUX2:=Y;
 (*-----------------------------------------------
  ITERA ATE UM LIMITE DESEJADO COM A IFS INVERSA
      E DETREMINA A ORBITA INVERSA DO PONTO X,Y
 ------------------------------------------------*)
       WHILE ((delta<>0) AND (J<=ITERA)) DO
 BEGIN
 VALOR1:=INV[DELTA,1]*AUX1+INV[DELTA,2]*AUX2-INV[DELTA,3];
 VALOR2:=INV[DELTA,4]*AUX1+INV[DELTA,5]*AUX2-INV[DELTA,6];
  LOCALIZA(VALOR1,VALOR2);
  VETDELTA[J+1]:=DELTA;
  CONTADOR:=CONTADOR+1;
  J:=J+1;
  AUX1:=VALOR1;
  AUX2:=VALOR2;
 END;
```

```
      AUX3:=ZMEDIO;
(*-----------------------------------------------
ITERA NOVAMENTE COM A IFS ORIGINAL  UITLIZANDO
               A ORBITA GUARDADA EM VETDELTA
-----------------------------------------------*)
      IF(CONTADOR-1=ITERA) THEN
 BEGIN
  FOR J:=ITERA DOWNTO 1 DO
   BEGIN
    K:=VETDELTA[J];
    VALOR1:=IFS[K,1]*AUX1+IFS[K,2]*AUX2+IFS[K,3];
    VALOR2:=IFS[K,4]*AUX1+IFS[K,5]*AUX2+IFS[K,6];
    VALOR3:=IFS[K,7]*AUX1+IFS[K,8]*AUX2
  +IFS[K,10]*AUX3+IFS[K,9];
    AUX1:=VALOR1;
    AUX2:=VALOR2;
    AUX3:=VALOR3;
   END;
                PZ:=AUX3;
 END
ELSE
BEGIN
  FOR J:=CONTADOR-1 DOWNTO 1 DO
   BEGIN
     K:=VETDELTA[J];
     VALOR1:=IFS[K,1]*AUX1+IFS[K,2]*AUX2+IFS[K,3];
     VALOR2:=IFS[K,4]*AUX1+IFS[K,5]*AUX2+IFS[K,6];
     VALOR3:=IFS[K,7]*AUX1+IFS[K,8]*AUX2
     +IFS[K,10]*AUX3+IFS[K,9];
     AUX1:=VALOR1;
     AUX2:=VALOR2;
     AUX3:=VALOR3;

    END;
                PZ:=AUX3;
END;
 END;     (* IF DELTA=0 *)
  END;  (* ----- FIM DA PROCEDURE ALTITUDE -----*)
```

```
PROCEDURE AJUSTE (var  na :integer);
(* ------------------------------------------------------
  ajusta um plano a um conjunto de na pontos (xx,yy,elev)
  usando minimos quadrados
------------------------------------------------------*)
var
 i,j                                          :integer;
 sx,sy,sz,sxx,syy,sxy,sxz,syz,det,bo          :double;
BEGIN
  sx:=0;
  sy:=0;
  sz:=0;
  sxx:=0;
  syy:=0;
  sxy:=0;
  sxz:=0;
  syz:=0;

  for i:=1 to na do
  begin
    sx:=sx+px[i];
    sy:=sy+py[i];
    sz:=sz+zz[i];
    sxx:=sxx+px[i]*px[i];
    syy:=syy+py[i]*py[i];
    sxy:=sxy+px[i]*py[i];
    sxz:=sxz+px[i]*zz[i];
    syz:=syz+py[i]*zz[i];
  end;
  sx:=sx/na;
  sy:=sy/na;
  sz:=sz/na;

  sxx:=sxx-na*sx*sx;
  syy:=syy-na*sy*sy;
  sxy:=sxy-na*sx*sy;
  sxz:=sxz-na*sx*sz;
  syz:=syz-na*sy*sz;
```

```
   det:=sxx*syy-sxy*sxy;
   if ( abs(det) <=1E-10) then
   begin
      writeln('determinante de matriz singular');
      exit;
   end;
   bx:=(sxz*syy-sxy*syz)/det;
   by:=(sxx*syz-sxz*sxy)/det;
   bo:=sz-bx*sx-by*sy;

END; (*---- fim da procedure ajuste ----------------*)

BEGIN
   dx:=ds/2;
   (* ------------------------------------------------
      calcula um vetor com nove pontos (x1,y,z) que
      serao utilizados para ajustar um plano a quadricula
     ------------------------------------------------*)
   for i:=1 to 3 do
   begin
     x1[i]:=xi;
     y[i]:=yi+(i-1)*dx;
     altitude(x1[i],y[i]);
     z[i]:=pz;
     x1[i+3]:=xi+dx;
     y[i+3]:=y[i];
     altitude(x1[i+3],y[i+3]);
     z[i+3]:=pz;
     x1[i+6]:=xi+2*dx;
     y[i+6]:=y[i];
     altitude(x1[i+6],y[i+6]);
     z[i+6]:=pz;
   end;

   (* ------------------------------------------------
      transforma  o vetor anterior em um vetor cujas
      altitudes sejam diferentes de zero
   ------------------------------------------------*)
   soma:=0;
```

```
cont:=0;
j:=1;
for i:=1 to 9 do
begin
  if (z[i]<>0) then
  begin
    zz[j]:=z[i];
    px[j]:=x1[i];
    py[j]:=y[i];
    cont:=cont+1;
    j:=j+1;
    soma:=soma+z[i];
  end;
end;

(* ------------------------------------------------
   calcula o parametro rg : localiza os vertices da
   quadricula na triangulacao e faz uma media com os
   parametros livres g = IFS[i,10]
 ------------------------------------------------*)
c:=0;
sm:=0;
localiza (x1[1],y[1]);
if ( delta<>0 ) then
begin
  c:=c+1;
  sm:=sm+ifs[delta,10];
end;
localiza (x1[3],y[3]);
if (delta<>0 ) then
begin
  c:=c+1;
  sm:=sm+ifs[delta,10];
end;
localiza (x1[7],y[7]);
if (delta<>0 ) then
begin
  c:=c+1;
  sm:=sm+ifs[delta,10];
```

```
  end;
  localiza (x1[9],y[9]);
  if ( delta<>0 ) then
  begin
    c:=c+1;
    sm:=sm+ifs[delta,10];
  end;
  rg:=sm/c;

  (* ------------------------------------------------
     calcula a altitude media da quadricula
  ------------------------------------------------*)

  alt_med:=soma/cont;

  (* ------------------------------------------------
     calcula o gradiente
    ------------------------------------------------*)
  ajuste(cont);
  grad1:=-bx;
  grad2:=-by;

END; (* --------- fim da procedure gradiente -------- *)


PROCEDURE ANGULO ( var a,b  :double);
(* ------------------------------------------------
  calcula o angulo que o plano da quadricula faz com a
  horizontal( f(x,y)=0). Parametros de entrada:
  coeficientes a e b do plano z = ax + by + c
 ------------------------------------------------*)
     var
       aux,aux1         :double;

     BEGIN
       aux:=sqrt(sqr(a)+sqr(b)+1);
       aux1:=1/aux;
       ang:=arccos(aux1);
```

```
    END;  (* ------ FIM DA PROCEDURE ANGULO -------*)

PROCEDURE DFLOW (var l,q :integer);
(* ------------------------------------------------------
 determina a direcao de escoamento utilizando o gradiente
  do plano ajustado
  Parametros de entrada: a e b da equacao do plano
  z = ax + by + c   ajustado a quadricula
------------------------------------------------------*)
 var
min,aux1              :double;
 BEGIN
 min:=param[l]^[q].at;
 desc:=0;
 aux1:=param[l+1]^[q].at;
 if ((aux1< min) and (aux1<>0)) then
 begin
 min:=aux1;
 desc:=1;
 end;
 aux1:=param[l+1]^[q-1].at;
 if ((aux1< min) and (aux1<>0)) then
 begin
 min:=aux1;
 desc:=2;
 end;
 aux1:=param[l]^[q-1].at;
 if((aux1< min) and (aux1<>0)) then
 begin
 min:=aux1;
 desc:=3;
 end;
 aux1:=param[l-1]^[q-1].at;
 if((aux1< min) and (aux1<>0)) then
 begin
 min:=aux1;
 desc:=4;
 end;
 aux1:=param[l-1]^[q].at;
```

```
  if ((aux1< min) and (aux1<>0)) then
  begin
  min:=aux1;
  desc:=5;
  end;
  aux1:=param[l-1]^[q+1].at;
  if ((aux1< min) and (aux1<>0)) then
  begin
  min:=aux1;
  desc:=6;
  end;
  aux1:=param[l]^[q+1].at;
  if ((aux1< min) and (aux1<>0)) then
  begin
  min:=aux1;
  desc:=7;
  end;
  aux1:=param[l+1]^[q+1].at;
  if ((aux1< min) and (aux1<>0)) then
  begin
  min:=aux1;
  desc:=8;
  end;
 END;
(*  o numero nove (0) significa um buraco
       ou lugar plano --*)
 (* --- fim  da procedure dflow --------*)



  PROCEDURE T_INFILT (var  a  :double);
  (* ------------------------------------------------
    calcula o tempo de infiltracao da quadricula
    dados de entrada:
    a  = angulo que a maxima declividade faz com z=0,
    t_max = valor maximo para o parametro inf.
   ------------------------------------------------*)

   BEGIN
```

```
    inf:= - (t_max/1.571)* a + t_max;
  END;  (*------ fim da procedure t_infilt -------*)

PROCEDURE VELOCIDADE ( var r,a   :double);
(* ------------------------------------------------
  calcula a velocidade da quadricula
  dados de entrada:
  vm = velocidade media  bacia,
  med_ang = angulo referente a maxima declividade da
            bacia,
  med_g = menor valor absoluto dos parametros g,
  r   = parametro g referente a quadricula,
  a  = angulo que a declividade faz com z = 0 .
 ------------------------------------------------*)

 var
   aux1,aux2,vl    :double;

 BEGIN
   aux1:=med_ang*( 1 - med_g);
   aux2:=vm*( 1 - r);
 vl:=(aux2/aux1)*a;
   ks:=vl/vm;
 END;  (* ----- fim da procedure velocidade ---------*)


 BEGIN
 CLRSCR;
 writeln(' entre com os nomes dos arquivos :');
 writeln;
 writeln(' arquivo de entrada com a triangulacao: ');
 readln(nom1);
 writeln(' arquivo de entrada com dados da bacia: ');
 readln(nom2);
 writeln(' arquivo de saida para o mvazao e rede_rio: ');
 readln(nom3);
 writeln(' arquivo de saida com os dados em texto :');
 readln(nom4);
 assign(arq1,nom1);
```

```
  assign(arq2,nom2);
  assign(arq3,nom3);
  assign(arq4,nom4);
  reset(arq1);
  reset(arq2);
  rewrite(arq3);
  rewrite(arq4);
  fillchar(rug,sizeof(rug),0);
  fillchar(agl,sizeof(agl),0);
  (*-------------------------------------------------------
      le o arquivo com a triangulacao da interpolacao
   -------------------------------------------------------*)
   FOR I:=1 TO 2 DO    READLN(ARQ1);
   READLN(ARQ1,NM);
   FOR I:=1 TO NM DO
     BEGIN
FOR J:=1 TO 9 DO
  BEGIN
    READ(ARQ1,X[I,J]);
  END;

  READLN(ARQ1);
 END;
   READLN(ARQ1);
   FOR I:=1 TO 3 DO
     BEGIN
READ(ARQ1,XO[I],YO[I],ZO[I]);
     END;
   READLN(ARQ1);
   READLN(ARQ1);
   FOR I:=1 TO NM DO
   BEGIN
     READLN(ARQ1,IFS[I,10]);
   END;
   readln(arq1);
   for i:=1 to nm do
   begin
     for j:=1 to 4 do
     begin
```

```
      read(arq1,pont[i,j]);
    end;
    readln(arq1);
  end;

 (* ------------------------------------------------
    le o arquivo com a matriz indice e demais dados
    de entrada
 ------------------------------------------------*)

  for i:=1 to 2 do readln(arq2);
  readln(arq2,t_max);
  readln(arq2);
  readln(arq2,x0,y0);
  readln(arq2);
  readln(arq2,nx,ny);
  readln(arq2);
  readln(arq2,ds);
  for i:=1 to 3 do  readln(arq2);
  readln(arq2,vm);
  readln(arq2);
readln(arq2,zmedio);
for i:=1 to 5 do  readln(arq2);
for i:=ny downto 1 do
  begin
    for j:=1 to nx do
    begin
      read(arq2,pert[i,j]);
    end;
    readln(arq2);
  end;

 (*---------------------------------------
 calcula os parametros da ifs
----------------------------------------*)
parametros(nm);

    (* ---------------------------------------
       cria as marizes de ponteiros
```

```
      ----------------------------------------*)

for i:=0 to ny+1 do
begin
  NEW(param[i]);
end;
for i:=0 to ny+1 do
begin
  for j:=0 to nx+1 do
  begin
  param[i]^[j].at:=0;
  param[i]^[j].df:=0;
  param[i]^[j].ksup:=0;
  param[i]^[j].indice:=0;
  param[i]^[j].tinf:=0;
  end;
end;

    (* ---------------------------------
       calcula med_g : media dos parametros g
      ----------------------------------------*)
    soma:=0;
    for i:=1 to nm do
    begin
      soma:=soma+ifs[i,10];
    end;
    med_g:=soma/nm;
soma:=0;
(* ----------------------------------------
 calcula todos os parametros fisiograficos
 para utilizacao no modelo hidrologico
 para todas as quadriculas (i,j)
 i: eixo dos y,
 j: eixo dos x,
 ----------------------------------------*)
  clrscr;
  for i:=1 to 12 do writeln;
  writeln('                    PROCESSANDO');
  writeln(' FAVOR NAO MEXER. OBRIGADO.');
```

```
  for i:=1  to 3 do writeln;
  writeln('                    ROSA');
for i:=1 to ny do
begin
  for j:=1 to nx do
  begin
  xx:=x0+(j-1)*ds;
  yy:=y0+(i-1)*ds;
  if ( pert[i,j]=1 ) then
  begin
  gradiente(xx,yy);
  angulo(grad1,grad2);
  agl[i,j]:=ang;
  rug[i,j]:=rg;
  t_infilt(ang);
  param[i]^[j].indice:=pert[i,j];
  param[i]^[j].tinf:=inf;
  param[i]^[j].at:=alt_med;
  end
  else
  begin
  param[i]^[j].df:=0;
  param[i]^[j].indice:=pert[i,j];
  param[i]^[j].ksup:=0;
  param[i]^[j].tinf:=0;
  param[i]^[j].at:=0;
  end;

      end; (* ---- fim do for j --------*)
    end;  (* ------ fim do for i -------*)
(*---------------------------------------*)

soma:=0;
nq:=0;
for i:=1 to ny do
begin
  for j:=1 to nx do
  begin
  soma:=soma+agl[i,j];
```

```
  if( pert[i,j] =1 ) then nq:=nq+1;
  end;
end;
med_ang:=soma/nq;

for i:=1 to ny do
begin
  for j:=1 to nx do
  begin
 if( pert[i,j]=1 ) then
 begin
  velocidade(rug[i,j],agl[i,j]);
  param[i]^[j].ksup:=ks;
  dflow(i,j);
  param[i]^[j].df:=desc;
 end;
  end;
end;



    (* ----------------------------------------
       grava os parametros nos arquivos de saida
     ----------------------------------------*)
writeln(arq4,' resultado do programa mod_terr');
writeln(arq4);
writeln(arq4,'ponto inicial da malha :',x0:10:5,y0:10:5);
for i:=0 to ny+1 do
begin
  write(arq3,param[i]^);
end;
writeln(arq4,' parametros g da ifs');
for i:=1 to nm do
begin
  write(arq4,ifs[i,10]:5:2);
end;
writeln(arq4);
writeln(arq4,'as matriz estao na ordem i=ny+1,0;
              j=0,nx+1');
```

```
writeln(arq4,' matriz direcao de escoamento');
for i:=ny+1 downto 0 do
begin
  for j:=0 to nx+1 do
  begin
  write(arq4,param[i]^[j].df:4);
  end;
  writeln(arq4);
end;
writeln(arq4);
writeln(arq4,' matriz constante de recessao em
              cada quadricula');
writeln(arq4);
for i:=ny+1 downto 0 do
begin
  for j:=0 to nx+1 do
  begin
  write(arq4,param[i]^[j].ksup:6:3);
  end;
  writeln(arq4);
end;
writeln(arq4);
writeln(arq4,' matriz altitude media em cada
                   quadricula');
writeln(arq4);
for i:=ny+1 downto 0 do
begin
  for j:=0 to nx+1 do
  begin
  write(arq4,param[i]^[j].at:6:3);
  end;
  writeln(arq4);
end;
(* ---------------------------------------------
  libera a matriz de ponteiros
  ---------------------------------------------*)
for i:=0 to ny+1 do
begin
```

```
  DISPOSE(param[i]);
end;

    close(arq1);
    close(arq2);
    close(arq3);
    close(arq4);

  END.
```

3.2 - DETERMINAÇÃO DA REDE DE DRENAGEM

```
PROGRAM REDE_RIO;
 (*-----------------------------------------------------
   programa para gerar rede de drenagem a partir das
   direcoes de escoamento de cada quadricula e altitude
   das quadriculas.
   Dados de entrada :
    - uma matriz contendo a altitude em cada quadricula
    - uma matriz contendo a direcao de escoamento de cada
      quadricula.
   Dados de saida :
    - uma matriz de fluxo acumulado onde  cada elemento
      representa o numero de quadriculas que drenam para
      essa quadricula.
  -----------------------------------------------------*)

  uses
  crt,graph;
  type
  nome=string[15];
  vet1=array[0..51] of record df     :integer;
                    indice :integer;
      ksup   :double;
      tinf   :double;
      at     :double;
     end;
  vet2=array[1..50] of integer;
  alinha=array[0..56] of ^vet1;
  blinha=array[1..55] of ^ vet2;
  var
 x,y,ki,kj        :integer;
 nx,ny            :integer;
 dy,dx,f,t,p          :integer;
 xi,yi,xf,yf      :double;
 fat              :double;
 xm1,xm2,ym1,ym2 :integer;
 ds,xo,yo         :double;
 limiar           :integer;
```

```
 par              :alinha;
 rank,acum        :blinha;
 i,j              :integer;
 arq1             :file of vet1;
 arq2             :text;
 st1,st2          :nome;
 index            :integer;
 iaux,jaux,aux    :integer;

 PROCEDURE VER_IJ ( var k :integer);
  (* ---------------------------------------------
    procedure para transformar uma posicao do vetor
    k em uma posicao dentro da matriz (i,j)
   ---------------------------------------------*)
   var
    a,b    :integer;

   BEGIN
     a:= k div ny;
     b:= k mod ny;

     if (b <> 0) then
     begin
       ki:=b;
       kj:=a+1;
     end
     else
     begin
       ki:=ny;
       kj:=a;
     end;
 END; (* ------ fim da procedure ver_ij --------*)

PROCEDURE INDEX_IJ ( var l,q,k :integer);
(* ---------------------------------------------
  calcula qual a quadricula jusante de (i,j) dada
  a direcao de escoamento.
  dados de entrada:
  l = a posicao i da malha,
```

```
 q = a posicao j da malha,
 k = parametro direcao de escoamento.
 dado de saida:
 ki = a posicao i da quad. jusante,
 kj = a posicao j da quad. jusante.
-----------------------------------------------*)
 var
   aux1,aux2     :integer;

BEGIN
   if (k=1) then
   begin
 aux1:=l+1;aux2:=q;
 end;
 if (k=2) then
 begin
 aux1:=l+1;aux2:=q-1;
 end;
 if (k=3) then
 begin
 aux1:=l;aux2:=q-1;
 end;
 if (k=4) then
 begin
 aux1:=l-1;aux2:=q-1;
 end;
 if (k=5) then
 begin
 aux1:=l-1;aux2:=q;
 end;
 if (k=6) then
 begin
 aux1:=l-1;aux2:=q+1;
 end;
 if (k=7) then
 begin
 aux1:=l;aux2:=q+1;
 end;
 if (k=8) then
```

```
 begin
 aux1:=l+1;aux2:=q+1;
 end;
   if (k=9) then
   begin
 aux1:=l;aux2:=q;
 end;

 ki:=aux1;kj:=aux2;
END; (* ----- fim da procedure index_ij ------*)


 PROCEDURE ORDENA;
 (*-----------------------------------------
 ordena a matriz de altitude da menor altitude
 para a maior altitude e guarda na matriz rank
 a posicao original da altitude na malha
-----------------------------------------*)
 var
  troc,i,i1,i2         :integer;
  j1,j2,iaux,raux      :integer;
  aux                  :real;

   BEGIN
    repeat
      troc:=0;
      for i:=1 to nx*ny-1 do
      begin
        ver_ij (i);
        i1:=ki;j1:=kj;
        iaux:=i+1;
        ver_ij (iaux);
        i2:=ki;j2:=kj;
  if (par[i2]^[j2].at < par[i1]^[j1].at ) then
  begin
          troc:=1;
    aux:=par[i1]^[j1].at;
    par[i1]^[j1].at:=par[i2]^[j2].at;
    par[i2]^[j2].at:=aux;
```

```
      raux:=rank[i1]^[j1];
      rank[i1]^[j1]:=rank[i2]^[j2];
      rank[i2]^[j2]:=raux;
    end;
       end;
      until ( troc=0 );
    END;(* ------- fim da procedure ordena ----------*)

  BEGIN
 clrscr;
writeln('ENTRE COM O NOME DO ARQUIVO DE ENTRADA DE DADOS:');
 readln(st1);
 writeln(' ENTRE COM O NOME DO ARQUIVO DE SAIDA DE DADOS:');
 readln(st2);
 assign(arq1,st1);
 reset(arq1);
 assign(arq2,st2);
 rewrite(arq2);
 writeln(' ENTRE COM O NUMERO DE QUADRICULAS NO EIXO X ');
 writeln(' Limite maximo = 50');
 readln(nx);
 writeln(' ENTRE COM O NUMERO DE QUADRICULAS NO EIXO Y ');
 writeln(' Limite maximo = 55');
 readln(ny);
 writeln(' ENTRE COM O VALOR DE DS (LADO DA QUADRICULA):');
 writeln(' ( O VALOR DEVE ESTAR CONVERTIDO PARA O
            PLANO X,Y )');
 readln(ds);
{ writeln(' ENTRE COM O VALOR DO PONTO (X,Y) INICIAL
            DA MALHA: ');
 readln(xo,yo);}
 writeln(' ENTRE COM O LIMIAR : ');
 readln(limiar);
{ st1:='lnovo.bin';
 st2:='acum21.sai';
 nx:=10;ny:=10;
 ds:=0.0656175;}
 xo:=0.31;yo:=0.15;
```

```
assign(arq1,st1);
reset(arq1);
assign(arq2,st2);
rewrite(arq2);


(*-----------------------------------
 abre as matrizes de ponteiros
 -----------------------------------*)

for i:=1 to ny do
begin
  NEW(rank[i]);
  NEW(acum[i]);
end;
for i:=0 to ny+1 do
begin
  NEW(par[i]);
end;
(*-----------------------------------------
le os parametros e determina a matriz rank
 -----------------------------------------*)
for i:=1 to ny do
begin
  read(arq1,par[i]^);
  for j:=1 to nx do
  begin
  rank[i]^[j]:=(j-1)*ny+i;
  acum[i]^[j]:=1;
  end;
end;

(*---------------------------------------
  ordena a matriz da maior altitude para
  menor altitude
 -----------------------------------*)
 ordena;

 (*---------------------------------------
```

```
calcula a matriz de fluxo acumulado
 ------------------------------------*)
for j:=nx downto 1 do
begin
for i:=ny downto 1 do
begin
 index:=rank[i]^[j];
 ver_ij ( index);
 iaux:=ki;
 jaux:=kj;
 aux:=par[iaux]^[jaux].df;
 if (aux <> 0) then
 begin
   index_ij (iaux,jaux,aux);
   acum[ki]^[kj]:=acum[ki]^[kj]+acum[iaux]^[jaux];
 end;
end;
end;

(*-------------------------------------------------
      grava um arquivo texto de saida com a matriz acum
 -------------------------------------------------*)
writeln(arq2,' arquivo de fluxo acumulado');
writeln(arq2,' obtido do arquivo de entrada: ',st1:15);
writeln(arq2,' nx = ',nx:3,'  ny = ',ny:3);
writeln(arq2,' ponto inicial : ',xo:10:5,', ',yo:10:5);
writeln(arq2,' valor do lado da quadricula : ',ds:10:5);
writeln(arq2);
writeln(arq2,' formato da matriz : i = ny ate 1,
                  j = 1 ate nx');
writeln(arq2);
for i:=ny downto 1 do
begin
for j:=1 to nx do
begin
  write(arq2,acum[i]^[j]:5);
end;
writeln(arq2);
end;
```

```
(* ----------------------------------------
  plota a rede de rios utilizando um limiar
  se o valor do fluxo acumulado de uma
  quadricula e maior que um limiar entao
  plota-se um segmento compreendido entre
  o centro desta quadricula e o centro da
  sua jusante .
  ----------------------------------------*)
  dy:=400;
  dx:=40;
  f:=250;
     (* ----------------------------------------
        fat = fator de conversao para tornar o eixo
        x e o eixo y de mesmo tamanho na impressao
     ----------------------------------------*)
{   fat:=8.8/10.1;}
  fat:=1;
  detectgraph(t,p);
  initgraph(t,p,'g:/LNG/TP/BGI');

  line(0,trunc(d-0),trunc(1*f),trunc(d-0));
  line(0,d,0,trunc(d-1*f*fat));
  line(0,trunc(d-0.32*f*fat),trunc(0.76*f),d);
  line(0,trunc(d-0.32*f*fat),trunc(1*f),trunc(d-0.95*f*fat));
  line(trunc(0.76*f),d,trunc(1*f),trunc(d-0.95*f*fat));

  for i:=1 to ny do
  begin
  for j:=1 to nx do
  begin
  if (acum[i]^[j]>=limiar ) then
  begin
    xi:=xo+(j-1)*ds;
    yi:=(yo+(i-1)*ds);
    xf:=xi+ds;
    yf:=(yi+ds);
    xm1:=trunc(f*(xi+xf)/2);
    ym1:=trunc(d-(f*(yi+yf)/2)*fat);
```

```
    aux:=par[i]^[j].df;
    if (aux<>0) then
    begin
    index_ij(i,j,aux);
    xi:=xo+(kj-1)*ds;
    yi:=yo+(ki-1)*ds;
    xf:=xi+ds;
    yf:=yi+ds;
    xm2:=trunc(f*(xi+xf)/2);
    ym2:=trunc(d-(f*(yi+yf)/2)*fat);
    line(xm1,ym1,xm2,ym2);
    end;
  end;
  end;
  end;


(*----------------------------------------
 libera as matrizes de ponteiros
  ----------------------------------------*)
for i:=1 to ny do
begin
  DISPOSE(rank[i]);
  DISPOSE(acum[i]);
end;
for i:=0 to ny+1 do
begin
 DISPOSE(par[i]);
end;
readln;
close(arq1);
close(arq2);
closegraph;
end.
```

```
PROGRAM COR_DIR;
(*--------------------------------------
  programa para corrigir as direcoes de
  escoamento para que a rede de rios nao
  tenha descontinuidade
 ---------------------------------------*)
uses
 crt;
type
  vet=array[0..51] of record df     :integer;
           indice :integer;
           ksup   :double;
           tinf   :double;
           at     :double;
         end;
  alinha=array[0..56] of ^vet;
  vet1=array[1..60] of integer;
  st=string[15];
var
  i,j,k,nx,ny,n  :integer;
  s              :integer;
  s1             :double;
  par            :alinha;
  auxi,auxj,d    :vet1;
  nom1,nom2      :st;
  arq1,arq2      :file of vet;

BEGIN
  clrscr;
  writeln('entre com o nome do arquivo de entrada:');
  readln(nom1);
  writeln('entre com o nome do arquivo de saida:');
  readln(nom2);
  writeln('entre com nx e  ny :');
  readln(nx,ny);
  assign(arq1,nom1);
  assign(arq2,nom2);
  reset(arq1);
  rewrite(arq2);
```

```
 (*-------------------------------------
 abre matriz de ponteiros e le os dados
 -------------------------------------*)
 for i:=0 to ny+1 do
 begin
 NEW(par[i]);
 end;
 (*---------------------------------
 zera a matriz para nao pegar lixo
---------------------------------*)

 for i:=0 to ny+1 do
 begin
 read(arq1,par[i]^);
 end;
 (*-------------------------------------
 le na tela as direcoes a serem mudadas
-------------------------------------*)

writeln(' entre com o numero de correcoes:');
readln(n);
writeln('formato de entrada : i,j,df(i,j)');
for i:=1 to n do
begin
  writeln(' entre com a correcao ',i:3);
  readln(auxi[i],auxj[i],d[i]);
end;

for i:=1 to ny do
begin
  for j:=1 to nx do
  begin
  for k:=1 to n do
  begin
  if ((i=auxi[k]) and (j=auxj[k])) then
  begin
    par[i]^[j].df:=d[k];
  end;
  end;
```

```
    end;
  end;

  for i:=0 to ny+1 do
  begin
    write(arq2,par[i]^);
  end;

  s:=0;
  s1:=0;
  for i:=1 to ny do
  begin
    for j:=1 to nx do
    begin
    if (par[i]^[j].indice=1) then s:=s+1;
    s1:=s1+par[i]^[j].ksup;
    end;
  end;
  s1:=s1/s;
  writeln('numero de quadriculas da bacia :',s:5);
  writeln('media de ksup = ',s1:6:4);



   for i:=0 to ny+1 do
   begin
   DISPOSE(par[i]);
   end;

   close(arq1);
   close(arq2);
END.
```