



Aspectos Computacionais da Modelagem Bidimensional do Escoamento Superficial em Bacias Hidrográficas

João Luiz Franco

Tese apresentada à Escola de Engenharia de
São Carlos, da Universidade de São Paulo,
como parte dos requisitos para obtenção do
Título de Doutor em Engenharia Hidráulica.

DEDALUS - Acervo - EESC



31100017902

ORIENTADOR: Prof. Dr. Fazal Hussain Chaudhry

São Carlos
1997



Class.	TESE-EESC
Cutt.	2328
Tombo	T136 97

Área: Hidráulica e Saneamento

311 00017902

at 0934019

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC-USP

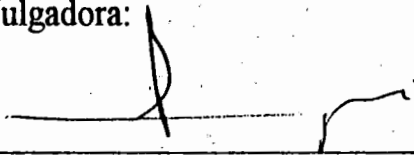
Franco, João Luiz
F825a Aspectos computacionais da modelagem
bidimensional do escoamento superficial em
bacias hidrogáficas / João Luiz Franco. São
Carlos, 1997.

Tese (Doutorado) -- Escola de Engenharia
de São Carlos-Universidade de São Paulo, 1997.
Orientador: Prof. Dr. Fazal Hussain Chaudhry

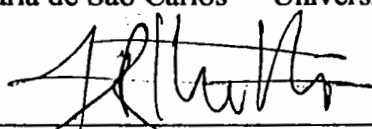
1. Escoamento superficial. 2. Método dos
elementos finitos. 3. Modelo numérico de
terreno. 4. Modelos hidrodinâmicos. I. Título

FOLHA DE APROVAÇÃO

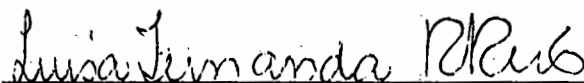
Tese defendida e aprovada em 23-5-1997
pela Comissão Julgadora:



Prof. Titular **FAZAL HUSSAIN CHAUDHRY** (Orientador)
(Escola de Engenharia de São Carlos - Universidade de São Paulo)



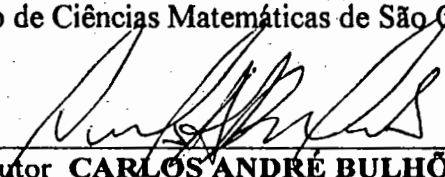
Prof. Associado **ANTONIO MAROZZI RIGHETTO**
(Escola de Engenharia de São Carlos - Universidade de São Paulo)



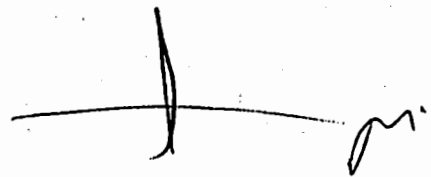
Profa. Doutora **LUISA FERNANDA RIBEIRO REIS**
(Escola de Engenharia de São Carlos - Universidade de São Paulo)



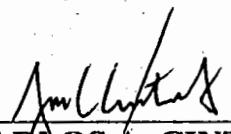
Profa. Titular **MARIA CAROLINA MONARD**
(Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo)



Prof. Doutor **CARLOS ANDRÉ BULHÕES MENDES**
(Universidade Federal do Rio Grande do Sul)



Prof. Titular **FAZAL HUSSAIN CHAUDHRY**
Coordenador da Área de Hidráulica e Saneamento



JOSE CARLOS A. CINTRA
Presidente da Comissão de Pós-Graduação

Dedicatória

A meus pais, pelo carinho, dedicação e exemplo de vida.

Agradecimentos

ao prof. Fazal, pela paciência em todas as fases do trabalho, pela dedicação, pelo apoio nas horas difíceis e por ter proporcionado a oportunidade de realizarmos este trabalho

aos amigos Mauro, Ethel, João Nivaldo e João de Lucca, pelo auxílio, preocupação e companheirismo

a Sandra, pelo amor e carinho; pelo afeto e compreensão em todas as horas

a Márcia e Pavi, pela preocupação em atender a todos com carinho e presteza

a todos que de alguma forma colaboraram neste trabalho

à CAPES, pelo apoio financeiro

a Deus, por haver me doado o dom, a oportunidade e a perseverança

Resumo

A resolução dos problemas de escoamento superficial em uma bacia hidrográfica pode ser efetuada a partir da discretização da região e utilização de métodos numéricos. São utilizados modelos digitais de terreno com base na rede triangular irregular para regiões bidimensionais complexas, pois esta admite elementos de diversos tamanhos e formatos, dada a maleabilidade do triângulo para discretizar regiões. Esta forma inibe a utilização do método das diferenças finitas, mas adequa-se ao método dos elementos finitos, que modela matematicamente cada elemento e apresenta solução para qualquer ponto interno do elemento. A região bidimensional é tratada como tal, utilizando a modelagem bidimensional a partir da região dividida em elementos triangulares. Isto permite representar com mais exatidão as características físicas da região, o que não acontece com a simplificação unidimensional comumente utilizada. Deste modo, obtém-se um resultado mais refinado do comportamento hidrológico da bacia hidrográfica.

A ferramenta computacional desenvolvida divide o cálculo em duas fases, separando a identificação dos canais da quantificação da propagação da vazão. Isto permite obter uma discretização mais elaborada sem sobrecarregar a fase de cálculo. Na fase de quantificação, os elementos são processados de forma independente, conduzindo a soluções eficientes sem exigir a utilização de um equipamento computacional de altíssima performance. As previsões do modelo computacional bidimensional para formas de bacias simples foram testadas através de comparação com soluções analíticas, obtendo resultados exatos em regime permanente e apresentando diferenças mínimas em regime não permanente. A aplicação do programa computacional é exemplificada através de previsão do escoamento superficial de uma bacia hidrográfica real. O modelo apresentado oferece meios para estudo do efeito de modificações da superfície em bacias hidrográficas.

PALAVRAS-CHAVE: modelo numérico de terreno, elementos finitos, escoamento superficial, bacias hidrográficas, modelos hidrodinâmicos

Abstract

The solution of runoff problems in a hydrographic basin can be obtained by discretization of the region of flow and the use of numerical methods. A Digital Elevation Model (DEM) for irregular regions based on Triangular Irregular Network (TIN) is used as it admits elements of diverse sizes and forms in view of the malleability of the triangle in discretization of geographical regions. This shape inhibits the use of finite differences but is highly suited to the finite element method, which mathematically describes each element and produces solution at any point inside the element. The hydrographic basin is treated as a two-dimensional region divided in triangular elements for numerical modelling. This permits a representation, with greater accuracy, of the physical characteristics of a region which would not be possible with one-dimensional simplification. Thus, more refined results on the hydrologic response of basin are obtained.

The computational tool developed in this study divides the calculation in two parts, separating the identification of channel drainage network from the quantification of discharge routing. This facilitates a more elaborate discretization without overloading the calculations. In the second part, the triangular elements are processed independently leading to efficient solutions requiring less computational resources. The predictions of two-dimensional computational model for basins of simple shapes were tested through comparison with the analytical solutions, obtaining exact results for steady-state and very close results for the transient state. Application of the computational program is illustrated through the prediction of the surface runoff of a real basin. The model presented here offers means for the study of the effect of surface modifications on basin response.

KEY WORDS: digital elevation model, finite element method, surface runoff, hydrologic response, hydrodynamic models.

Sumário

1. Introdução.....	1
1.1 Motivação do Trabalho	1
1.2 Organização do Trabalho	3
2. Revisão Bibliográfica.....	6
2.1 Modelos Hidrodinâmicos de Propagação de Cheias	6
2.1.1 O Modelo da Onda Cinemática.....	7
2.1.2 Aplicação dos Modelos Hidrodinâmicos nas Bacias Hidrográficas	11
2.2 Sistemas de Suporte de Decisão Espacial	13
2.2.1 Sistemas Baseados em Conhecimento.....	13
2.2.2 Sistemas de Informação Geográfica.....	16
2.2.3 Integrando GIS e DSS.....	18
2.2.4 Aplicações de SDSS em Recursos Hídricos	20
2.2.5 Desafios dos SDSSs	22
2.3 Modelos de Elevação Digital.....	23
2.3.1 Grade Regular	24
2.3.2 Rede Triangular Irregular (TIN).....	25
2.3.3 Linhas de Contorno (Vetorial).....	26
2.3.4 Aplicações do modelo DEM-TIN.....	26
2.4 Métodos Numéricos para Equações Diferenciais	29
2.4.1 Método das Diferenças Finitas.....	32
2.4.2 Método dos Elementos Finitos	35
2.4.3 Métodos de Runge-Kutta	39
3. Metodologia	40
3.1 Modelo TIN.....	40
3.1.1 Representação de uma Rede Triangular	41
3.1.2 Funções de Aproximação para Elementos Triangulares.....	43
3.2 Modelagem de Fluxo em um DEM-TIN.....	46
3.2.1 Determinação da Direção do Fluxo.....	46
3.2.2 Quantificação da Divisão do Fluxo	48
3.2.3 Identificação da Direção do Fluxo	49
3.2.4 Intersecção de Fluxos.....	51
3.3 Métodos Numéricos Aplicados ao escoamento Superficial.....	52
3.3.1 Método das Diferenças Finitas.....	52
3.3.2 Método dos Elementos Finitos	54
3.3.3 Equação da Continuidade Bidimensional	61

4. Solução Computacional	68
4.1 Objetivos do Estudo Computacional.....	68
4.2 escoamento Superficial Unidimensional	69
4.2.1 Método das Diferenças Finitas.....	70
4.2.2 Método dos Elementos Finitos	71
4.2.3 Exemplos com intensidade de chuva variável	80
4.3 escoamento Superficial Bidimensional	82
4.3.1 Resolução no elemento triangular	83
4.3.2 Comunicação entre elementos	88
4.3.3 escoamento no Canal.....	92
4.3.4 escoamento numa Bacia Hidrográfica	93
4.3.5 Aspectos da Implementação	101
4.3.6 Definição da sequência de processamento dos elementos.....	105
5. Conclusão	109
5.1 Avaliação dos Resultados.....	109
5.2 Trabalhos Futuros	111
Referências Bibliográficas	113
Apêndice A - Listagem dos Programas	124
Apêndice B - Listagem dos Arquivos	141

LISTA DE FIGURAS

Figura 3.1-1: Exemplo de uma Rede Triangular	41
Figura 3.2-1: Passagem do fluxo pela superfície triangular.....	47
Figura 3.2-2: Fluxo saindo por dois lados do triângulo	47
Figura 3.2-3: Divisão do Fluxo de Saída do Triângulo.....	47
Figura 3.2-4: Ponto de Divisão do Triângulo.....	48
Figura 3.2-5: Ângulos formados pelos lados do triângulo com o eixo x	49
Figura 3.2-6: Troca de Fluxo entre Triângulos	51
Figura 3.2-7: Identificação de Canal na Intersecção dos Triângulos	51
Figura 3.3-1: Região Triangular	62
Figura 4.2-1: Solução Numérica por Diferenças Finitas para o Escoamento Superficial por Onda Cinemática (N=3,5,11,21)	71
Figura 4.2-2: Solução numérica para o Escoamento Superficial por Onda Cinemática - Elementos Finitos - Solução 1	72
Figura 4.2-3: Aperfeiçoamento da Solução 1 - Método dos Elementos Finitos - Utilizando Runge-Kutta no tempo	73
Figura 4.2-4: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Velocidade	74
Figura 4.2-5: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Velocidade e Tempo Médio	75
Figura 4.2-6: Solução numérica para o Escoamento Unidimensional - Elementos Finitos - Aproximação Linear para Velocidade - Comparativo entre Diferentes Métodos para o Processamento Temporal.....	76
Figura 4.2-7: Solução numérica para o Escoamento Unidimensional utilizando Elementos Finitos com Aproximação Linear para Velocidade - Comparativo entre Diferentes Métodos para Processamento Temporal no Intervalo a 200.....	80
Figura 4.2-8: Solução numérica para Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Vazão	77
Figura 4.2-9: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Vazão - Runge Kutta no Processamento Temporal	78
Figura 4.2-10: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Vazão - Comparativo entre os Métodos Euler e Runge-Kutta no Processamento Temporal.....	79
Figura 4.2-11: Escoamento unidimensional com término da chuva.....	80
Figura 4.2-12: Intensidade da chuva variando no tempo	81
Figura 4.2-13: Escoamento unidimensional com chuva variando no tempo	81
Figura 4.3-1: Triângulo subdividido em 4.....	84
Figura 4.3-2: Triângulo subdividido em 16.....	84
Figura 4.3-3: Triângulo com base para cima.....	85

Figura 4.3-4: Solução para o triângulo com base para cima	86
Figura 4.3-5: Triângulo com base para baixo	86
Figura 4.3-6: Solução para o triângulo com base para baixo	87
Figura 4.3-7: Região retangular dividida em 3 triângulos	89
Figura 4.3-8: Solução para a região retangular dividida em 3 triângulos.....	90
Figura 4.3-9: Região retangular dividida em 4 triângulos	90
Figura 4.3-10: Solução da região dividida em 4 triângulos.....	91
Figura 4.3-11: Região com Canal de Escoamento.....	92
Figura 4.3-12: Solução da Região com Canal de Escoamento.....	93
Figura 4.3-13: Visualização da Bacia Hidrográfica	95
Figura 4.3-14: Linhas de Contorno da Bacia Hidrográfica	95
Figura 4.3-15: Divisão da região em elementos triangulares.....	96
Figura 4.3-16: Sequência de Processamento dos Elementos e Formação dos Canais	96
Figura 4.3-17: Vazão na saída do canal para o caso da Bacia Hidrográfica: Intensidade da Precipitação constante.....	97
Figura 4.3-18: Intensidade da Precipitação variável no tempo para a Bacia Hidrográfica	98
Figura 4.3-19: Vazão na saída do canal para o caso da Bacia Hidrográfica: Intensidade da Precipitação variável no tempo	98
Figura 4.3-20: Vazão em pontos internos para a Bacia Hidrográfica com Intensidade da Precipitação variável no tempo	99
Figura 4.3-21: Vazão observada na saída da Bacia Hidrográfica	100
Figura 4.3-22: Vazão por unidade de área obtida na saída da Bacia Hidrográfica...	100

Lista de Tabelas

Tabela 3.1-1: Coordenadas dos pontos da região.....	42
Tabela 3.1-2: Definição dos triângulos e vizinhança.....	43
Tabela 3.2-1: Entrada e saída do fluxo de água através dos lados do triângulo.....	50
Tabela 4.2-1: Erro da solução numérica utilizando Diferenças Finitas.....	70
Tabela 4.2-2: Discretização utilizada para o Método dos Elementos Finitos - Caso Unidimensional.....	72
Tabela 4.2-3: Diferença entre o resultado numérico e a solução analítica: comparativo entre três métodos.....	75
Tabela 4.2-4: Erro acumulado para diversos níveis de discretização.....	78
Tabela 4.2-5: Erro acumulado para diversos níveis de discretização utilizando Runge-Kutta no processamento temporal.....	79
Tabela 4.3-1: Coordenadas do Triângulo com base para cima.....	85
Tabela 4.3-2: Coordenadas do Triângulo com base para baixo.....	87
Tabela 4.3-3: Coordenadas da região retangular dividida em 3 triângulos.....	88
Tabela 4.3-4: Coordenadas dos vértices da região retangular discretizada em triângulos.....	91
Tabela 4.3-5: Coordenadas da Região com Canal de Escoamento.....	92
Tabela 4.3-6: Coordenadas da Bacia Completa.....	94

Lista de Símbolos

A	área da seção de escoamento
C	coeficiente de resistência de Chézy
f	saída lateral por unidade de comprimento do canal
g	aceleração da gravidade
h	altura da água em relação ao fundo do canal
i	entrada lateral por unidade de comprimento do canal
L	comprimento
m	expoente do escoamento cinemático
n	fator de resistência de Manning
q	entrada lateral na região
Q	vazão na seção de escoamento
Q_x	vazão na seção de escoamento na direção x
Q_y	vazão na seção de escoamento na direção y
S_0	declividade do fundo do canal
S_f	declividade da linha de energia
t	tempo
v	velocidade média do escoamento
α	coeficiente do escoamento cinemático
ϑ_i	ângulo formado pelo lado i do triângulo com o eixo x
ψ	direção do gradiente

1. Introdução

1.1 Motivação do Trabalho

O crescimento da população mundial e o conseqüente aumento da utilização da superfície terrestre pelo homem fazem das águas, dos rios e das bacias hidrográficas elementos de extrema importância. O uso indevido da água, do solo, ou da vegetação de uma determinada região pode causar efeitos destrutivos na região, bem como aumentar o custo do desenvolvimento e ameaçar a qualidade de vida nas regiões vizinhas de jusante, tanto no ponto de vista da qualidade da água quanto do excesso, que toma a forma de enchentes. Portanto, planejamentos integrados tornam-se cada vez mais necessários ao bom uso do meio ambiente e seus recursos naturais.

Assim sendo, é fundamental o bom entendimento do comportamento hidrológico de uma bacia hidrográfica, e por conseguinte, do hidrograma gerado ou modificado nessa bacia. Por isso, o escoamento superficial tem sido objeto de estudos aprofundados. A ocorrência do escoamento superficial é resultado da água precipitada na superfície da bacia que percorre diversos caminhos até chegar às calhas dos rios, formando assim o escoamento fluvial. Parte da chuva infiltra-se na superfície terrestre e, movendo-se lentamente pelo meio poroso, alimenta as calhas dos rios, garantindo-lhes uma certa sustentação do escoamento fluvial nos períodos não chuvosos. Outra parte, movendo-se mais rapidamente sobre a superfície, chega às calhas fluviais, provocando mudanças mais abruptas no regime dos cursos d'água.

Para a obtenção de hidrogramas, ou seja, transformação de chuva efetiva em vazão superficial, diversos modelos são disponíveis na literatura. A maioria desses modelos está baseada em conceitos lineares e constituem-se em métodos bastante utilizados na engenharia, apesar das limitações de suas aplicações. Quando uma grande bacia está em estudo, em geral, é necessário subdividi-la em áreas menores, onde os referidos modelos são aplicados. Determinados os hidrogramas para cada sub-área, eles devem ser combinados levando-se em conta os efeitos de translação e armazenamento dentro da calha principal ou secundária do rio.

Um dos problemas fundamentais do projeto e operação de obras é estudar a propagação e a mudança na forma de um hidrograma no espaço. À medida que um hidrograma caminha para jusante necessita-se determinar o tempo de propagação e a atenuação para outras seções. As técnicas envolvidas nesse problema são classificadas em duas categorias: modelos hidrológicos e modelos hidrodinâmicos. Os modelos hidrológicos aplicam a equação da continuidade e relações que descrevem o armazenamento e as vazões de entrada e saída. Os modelos hidrodinâmicos utilizam as equações de Saint-Venant, tratando de forma mais adequada a dinâmica do escoamento. A grande vantagem dos modelos hidrodinâmicos reside na possibilidade de incorporar as variações espaciais e temporais nos elementos que caracterizam a resposta da bacia durante o escoamento superficial. Entretanto, a utilização dessas variações envolve manipulação de informações volumosas no espaço.

Este trabalho tem por objetivo modelar o escoamento superficial bidimensional e simular a propagação da onda cinemática, a partir de informações digitais da superfície. A maioria dos modelos existentes não aplica diretamente a modelagem bidimensional. Ao contrário, dividem a bacia em elementos com características unidimensionais, que são conectados de modo a simular um comportamento bidimensional. A utilização desse artifício deve-se à complexidade da modelagem bidimensional, além da limitação de alguns métodos numéricos, que não trabalham com discretizações irregulares.

Neste trabalho, foi desenvolvido uma estrutura computacional para cálculo de vazão nos diversos pontos da bacia, a partir de informações digitais da topografia do terreno e vazões de entrada em forma de águas pluviais. Para o cálculo da vazão e altura do nível d'água nos diversos pontos da bacia, são utilizadas as equações bidimensionais, de forma que os resultados obtidos são fruto de um processamento realmente bidimensional na bacia, e não apenas uma aproximação do caso unidimensional.

1.2 Organização do Trabalho

O trabalho está dividido em 3 partes principais:

No Capítulo 2 - Revisão Bibliográfica - são apresentados de forma sucinta os principais temas envolvidos no trabalho. À primeira vista, alguns temas aparentam não ter relação alguma com os outros; no entanto, ao final verificamos que cada um deles tem papel importante no conjunto.

- Na seção 1 são estudados os Modelos Hidrodinâmicos, e são apresentadas as equações do modelo de onda cinemática utilizadas neste trabalho.
- Na seção 2 são apresentados os Sistemas de Suporte de Decisão Espacial, uma ferramenta para as pessoas envolvidas com metodologias complexas envolvendo informações geográficas. Estes sistemas aliam as facilidades de processamento de informações espaciais dos Sistemas de Informação Geográficas com técnicas de Inteligência Artificial para cálculos envolvendo decisões complexas.

- Na seção 3 descreve-se as diferentes formas de representação digital do relevo, utilizando os Modelos de Elevação Digital, e suas diversas aplicações.
- Na seção 4, são revisados os principais Métodos Numéricos utilizados na solução de equações diferenciais. Estes, junto com os modelos hidrodinâmicos, são a base do processamento numérico do escoamento superficial.

No Capítulo 3 - Metodologia - é feito o desenvolvimento teórico da metodologia utilizada, incluindo métodos de representação, formulação matemática e métodos numéricos.

- Na seção 1 desenvolve-se o Modelo de Elevação Digital com Rede Triangular Irregular, apresentando-se a forma de armazenamento das informações e o desenvolvimento das funções de aproximação dentro de uma superfície triangular.
- Na seção 2 apresenta-se um método para determinar a direção do fluxo dentro de uma região discretizada em triângulos e identificar a formação de canais.
- Na seção 3, as equações do modelo de onda cinemática são desenvolvidas para a aplicação de métodos numéricos, tanto para o caso unidimensional quanto bidimensional. Diferentes formulações são apresentadas, com o objetivo de selecionar a melhor delas para utilizar no modelo bidimensional.

No Capítulo 4 - Solução Computacional - são apresentados diversos problemas e as respectivas soluções obtidas para cada formulação desenvolvida no capítulo 3.

- Na seção 1 trabalha-se sobre a modelagem unidimensional, testando as formulações desenvolvidas, visando encontrar a melhor solução.
- Na seção 2, utiliza-se o modelo bidimensional, verificando sua validade em diversos exemplos. Nesta seção também são discutidos aspectos da implementação, indicando como deve ser feita a entrada de dados e mostrando de forma concisa a resolução de alguns problemas.

No Capítulo 5, são reveladas as conclusões do trabalho, abrindo espaço para trabalhos futuros partir dos resultados obtidos e as ferramentas desenvolvidas neste trabalho.

2. Revisão Bibliográfica

2.1 Modelos Hidrodinâmicos de Propagação de Cheias

Uma bacia hidrográfica é um sistema físico extremamente complexo, contendo, inclusive, entradas estocásticas. No entanto, segundo WOOLHISER & LIGGETT (1967), para uma determinada especificação das características físicas do sistema, o estado inicial e as entradas, a resposta do sistema é essencialmente determinística. O elemento hidráulico básico de uma bacia hidrográfica é um fluxo superficial plano descarregando como afluxo lateral para um canal. O fluxo neste elemento é variável no tempo e no espaço. O afluxo lateral consiste da chuva efetiva menos as perdas por infiltração, que também varia tanto no espaço quanto no tempo.

O desenvolvimento de modelos de simulação são hoje de grande importância e encontram diversas aplicações em hidrologia e recursos hídricos [AZEVEDO & FONTANE (1993)]. Para simular a resposta de uma bacia hidrográfica complexa a partir da precipitação, deve-se construir um modelo matemático combinando componentes mais simples. O problema de fluxo não permanente variando no espaço, sobre uma região planar deve ser resolvido considerando-se determinadas condições de contorno. De posse desse resultado, deve-se então combinar os elementos para obter a resposta do sistema completo [WOOLHISER & LIGGETT (1967)]. O fenômeno da propagação de cheias está diretamente relacionado com a forma da bacia hidrográfica. Os modelos hidrodinâmicos trabalham com a água disponível para o escoamento superficial. Segundo AZEVEDO & FONTANE (1993), para se obter uma relação entre a precipitação e a propagação de cheias, é necessário

calcular também a água que é abstraída por infiltração. Essa relação é obtida através de modelos hidrológicos chuva-vazão, através da análise de dados como tipo de solo, clima e vegetação. Para a transformação de chuva efetiva em vazão superficial diversos modelos foram desenvolvidos. MAGALHAES (1989) afirma que a maioria desses modelos estão baseados em conceitos lineares e constituem-se em métodos bastante utilizados na engenharia, apesar das limitações de suas aplicações. Atualmente, podem ser encontrados diversos modelos de propagação de cheias, em diversos níveis de sofisticação [TUNG (1985)]. Entre os mais sofisticados, destaca-se a modelagem hidrodinâmica de fluxo unidimensional utilizando equações de Saint-Venant [Huang & Song (1985)]. Hsu et al. (1995) mostraram o potencial das redes neurais para simular o procedimento hidrológico em bacias hidrográficas. As redes neurais artificiais são estruturas matemáticas capazes de identificar relacionamentos complexos não-lineares entre conjuntos de dados de entrada e saída. Ichoku et al. (1996) aplicaram as técnicas de fractais para investigar os efeitos dos contornos sobre a rede de canais, extraída por duas técnicas automatizadas: método de acumulação de fluxo e método de estruturação multinível.

2.1.1 O Modelo da Onda Cinemática

O escoamento em rios e canais são classificados como permanentes e não permanentes, dependendo da variação no tempo em relação a uma determinada seção de observação, das principais grandezas que os caracterizam, como vazão, profundidade e velocidade. Os escoamentos naturais são quase sempre não permanentes; porém, quando a variação temporal é bastante lenta, eles geralmente são estudados como permanentes.

A abordagem não permanente, gradualmente variada, é bem mais complexa. Sua teoria foi inicialmente desenvolvida por Laplace e Lagrange, na França, durante o século XVIII e por Saint-Venant e outros pesquisadores no último século [MAGALHAES (1989)]. A descrição matemática do escoamento é composta de equações diferenciais parciais não lineares, baseadas em leis físicas de conservação de massa e quantidade de movimento e conhecidas como equações de Saint-Venant.

2.1.1.1 Equações de Saint-Venant

O desenvolvimento das equações de Saint-Venant para canais naturais considera que o escoamento é unidimensional e se dá no plano longitudinal ao rio. As equações de Saint-Venant que regem o escoamento não permanente e gradualmente variado são:

- Equação da Continuidade

$$\frac{\delta A}{\delta t} + \frac{\delta Q}{\delta x} = i - f \quad (2.1-1)$$

- Equação Dinâmica (Quantidade de Movimento)

$$\frac{\delta v}{\delta t} + v \frac{\delta v}{\delta x} + g \frac{\delta h}{\delta x} + v \frac{i - f}{A} = g(S_0 - S_f) \quad (2.1-2)$$

onde:

- A = área da seção de escoamento
- Q = vazão na seção de escoamento
- h = altura da água em relação ao fundo do canal
- v = velocidade média do escoamento
- i = entrada lateral por unidade de comprimento do canal
- f = saída lateral por unidade de comprimento do canal
- g = aceleração da gravidade
- S_0 = declividade do fundo do canal
- S_f = declividade da linha de energia

Nas equações acima, A , h e v são funções de x e t .

2.1.1.2 Simplificações do Modelo

Quando há interesse no comportamento hidrológico, deve ser feita uma modelagem do fluxo superficial utilizando soluções numéricas das equações de Saint-Venant. Muitas vezes, pode-se reduzir o esforço computacional utilizando-se as formas mais simples das equações de fluxo, segundo as características específicas do canal. Das diversas simplificações, destacam-se o modelo de onda cinemática e modelo de difusão [BLANDFORD & ORMSBEE (1993), TAYFUR et al. (1993)].

O Modelo de Difusão é formado pela equação da continuidade e pela equação de difusão. A equação de difusão é obtida desprezando-se os dois primeiros termos da equação dinâmica:

$$g \frac{\delta h}{\delta x} + v \frac{i - f}{A} = g(S_0 - S_f) \quad (2.1-3)$$

A equação da onda cinemática é obtida desprezando-se os termos à esquerda da equação dinâmica, tornando-se:

$$S_0 = S_f \quad (2.1-4)$$

A equação (2.1-4), juntamente com a equação da continuidade formam o Modelo de Onda Cinemática. Este modelo é aplicável a rios em regime fluvial nos quais os termos envolvendo velocidade são desprezíveis. O modelo pode também ser aplicado a problemas de fluxo superficial em diversas situações práticas encontradas no

gerenciamento de recursos hídricos [VIEUX et al. (1990)]. No modelo da onda cinemática, há uma única relação entre a vazão e a altura do nível d'água, dada pela seguinte equação:

$$Q = \alpha h^m \quad (2.1-5)$$

Na equação (2.1-5), o coeficiente α e o expoente m são definidos por leis de resistência de fluxo uniforme, tais como Chézy ou Manning [STREETER (1966)]:

- Chézy:

$$\alpha = C\sqrt{S_0} \quad (2.1-6)$$

- Manning:

$$\alpha = \frac{1}{n}\sqrt{S_0} \quad (2.1-7)$$

onde:

- C = coeficiente de resistência de Chézy
- n = fator de resistência de Manning
- S_0 = declividade

2.1.2 Aplicação dos Modelos Hidrodinâmicos nas Bacias Hidrográficas

A dedução original dos modelos hidrodinâmicos tratam do escoamento unidimensional. Assim, durante muito tempo, a modelagem de fluxo superficial utilizou a equação da onda cinemática unidimensional para calcular a propagação do fluxo na bacia. O fluxo superficial nas bacias hidrográficas está diretamente relacionado à declividade natural do terreno, o qual raramente apresenta propriedades físicas e hidráulicas homogêneas. Segundo ZHANG & CUNDY (1989), a microtopografia, a rugosidade da superfície as propriedades hidráulicas do solo podem variar com escalas de centímetros a metros. Para incorporar declividades variáveis, pode-se utilizar o método da cascata cinemática, que permite aplicar os modelos hidrodinâmicos a bacias hidrográficas, utilizando o seguinte artifício: a superfície é subdividida em uma série de superfícies planas, cada uma com diferentes declividades. Em seguida, as contribuições de cada superfície são somadas, de acordo com os caminhos lineares ao longo dos quais a água deveria, supostamente, deslocar-se. No entanto, as simplificações utilizadas nesse processo acarretam um processo não determinístico, com um fator considerável de subjetividade envolvida, pois define-se de antemão os caminhos através dos quais a água deverá se deslocar.

O problema da subjetividade pode ser minimizado através da utilização de métodos não determinísticos. Em MOREL-SEYTOUX et al. (1993), procedimentos estatísticos são combinados com a equação de onda cinemática, levando em conta a não linearidade e a dispersão do fenômeno de propagação da onda de cheia. No entanto, para se obter um procedimento hidráulico e hidrológico mais preciso, é necessário utilizar modelos dinâmicos multidimensionais que incorporem a variação espacial da declividade [ZHANG & CUNDY (1989)]. Deste modo, para que se possa atingir resultados compatíveis com a natureza bidimensional do movimento da água na superfície da bacia, é necessário considerar as equações em duas dimensões. Segundo VIEUX et al. (1990), essas equações podem ser obtidas através de uma análise detalhada do escoamento superficial bidimensional, segundo o modelo da onda cinemática. Neste caso, a equação da continuidade torna-se:

$$\frac{\delta h}{\delta t} + \frac{\delta Q_x}{\delta x} + \frac{\delta Q_y}{\delta y} = i - f \quad (2.1-8)$$

onde:

- h = altura da água em relação ao fundo do canal
- Q_x = vazão por unidade de largura na direção x
- Q_y = vazão por unidade de largura na direção y
- i = entrada lateral por unidade de comprimento do canal
- f = saída lateral por unidade de comprimento do canal

A relação entre a vazão e a altura no caso bidimensional é dada pela equação a seguir, onde α_x e α_y são coeficientes nas direções x e y , definidos pela lei de Chezy ou pela lei de Manning.

$$\begin{aligned} Q_x &= \alpha_x h^m \\ Q_y &= \alpha_y h^m \end{aligned} \quad (2.1-9)$$

Grayson et al. (1992a) apresentam o modelo computacional baseado em parâmetros hidrológicos THALES. A aplicação deste modelo a duas bacias hidrográficas forneceu diferentes respostas hidrológicas dominantes. Estes resultados são discutidos em Grayson et al (1992b), onde se discute a viabilidade da simulação hidrológica baseada em modelos físicos. No entanto, Goodrich & Woolhiser (1994) alertam que somente com dados experimentais de alta qualidade é possível determinar a inviabilidade de assumir a topografia como parâmetro hidrológico dominante. Smith et al.(1994) ressaltam que deve-se distinguir os modelos hidrodinâmicos, baseados em características físicas, de um modelo computacional, tal como o THALES, que além de incorporar essas características, possui muitas outras aproximações. O primeiro modelo é uma abstração matemática de procedimentos naturais observados, que inclui simplificações em experimentos controlados. O segundo modelo aplica tais características a partes complexas da natureza, assumindo valores numéricos, topográficos e computacionais ao longo do processamento, possibilitando o acúmulo de erros. Deste modo, não se pode esperar de um modelo computacional a mesma precisão de um experimento controlado, o que não inviabiliza a utilização de modelos físicos em processos de simulação. Por outro lado, Robinson et al. (1995), utilizando modelos geomorfológicos de propagação, mostraram que os parâmetros da bacia variam de acordo com suas dimensões.

2.2 Sistemas de Suporte de Decisão Espacial

Os modelos de simulação vem sendo aprimorados cada vez mais com os avanços na informática, tanto nos equipamentos quanto na diversidade de programas disponíveis, permitindo trabalhar com maiores quantidades de dados e informações. Na maioria dos casos, os problemas são determinísticos. No entanto, existem casos em que a participação de um especialista é fundamental para tomar certas decisões, indicando o melhor caminho a ser seguido entre as possibilidades existentes. Em alguns casos, pode envolver, inclusive, uma parcela de subjetividade significativa.

Os caminhos para solucionar os problemas não determinísticos fazem parte das pesquisas de Inteligência Artificial e Sistemas Especialistas. Em alguns casos, para decidir qual o melhor caminho a ser adotado, pode-se utilizar de Sistemas de Suporte de Decisão (SSD).

2.2.1 Sistemas Baseados em Conhecimento

Conforme descrevem NILSSON (1982) e WINSTON (1984), Inteligência Artificial é a parte da Ciência da Computação que se preocupa em elaborar “sistemas inteligentes”, isto é, sistemas que exibem características normalmente associadas à inteligência humana. FORD (1987) reforça que as pesquisas na área de Inteligência Artificial concentram-se na elaboração de programas para resolver os problemas em que, até o momento, o ser humano supera a máquina.

Segundo HAYES-ROTH et. al (1983) e JACKSON (1983), um dos segmentos mais explorados das pesquisas de Inteligência Artificial são os Sistemas Especialistas. Os Sistemas Especialistas são programas destinados a solucionar problemas em um campo específico do conhecimento, utilizando conhecimentos e procedimentos de inferência na solução de tais problemas [ALTY & COOMBS (1984), CARNOTA &

TEZKIEWIZ (1988)]. Estes programas devem apresentar um desempenho comparável ao dos especialistas humanos no campo em que é aplicado [WATERMAN (1986)].

Os Sistemas Baseados em Conhecimento, ou simplesmente Sistemas de Conhecimento, são programas de computador que resolvem (ou ajudam a resolver) problemas que requerem inteligência humana. Diferenciando-se dos sistemas tradicionais de processamento de dados, que processam grandes volumes de informação algorítmicamente, os Sistemas de Conhecimento geralmente examinam um grande número de possibilidades e, em muitos casos, podem construir uma solução dinamicamente [MONARD & HEBIHARA (1987)]. Outras diferenças entre sistemas convencionais e Sistemas de Conhecimento que podem ser destacadas são:

- a maneira como estão organizados;
- o modo como incorporam o conhecimento;
- a fôrma de execução;
- a impressão que causam aos usuários, pois transmitem uma sensação de raciocínio.

Os Sistemas Especialistas constituem uma classe dos Sistemas de Conhecimento que resolvem problemas muito complexos. Suas principais características são:

- contemplam múltiplas hipóteses simultaneamente;
- interagem com o usuário para buscar informações que estão além do conhecimento do sistema;
- justificam suas conclusões, bem como o motivo pelo qual solicitou alguma informação adicional durante o processamento;
- trabalham com dados incertos e regras imprecisas;
- o procedimento de resolução do problema é de fácil compreensão para o usuário do sistema;
- flexível o suficiente para que seu conhecimento possa ser aplicado a novos tipos de problemas.

Existem algumas características que diferenciam um Sistema Especialista de um Sistema de Conhecimento [JACKSON (1983)], mas basicamente um Sistema Especialista é um Sistema de Conhecimento que apresenta um desempenho comparável ao de um especialista humano. O termo Sistema Baseado em Conhecimento foi introduzido como alternativa para o termo Sistema Especialista, que ganhou uma certa conotação negativa, dado que esse termo é muito pretensioso. Porém, na maioria dos trabalhos nesta área, estes termos continuam sendo usados indistintamente [FRANCO (1991)].

O desenvolvimento de um Sistema Especialista consiste, basicamente, em extrair o conhecimento de um especialista humano e colocá-lo na Base de Conhecimento, no formato adequado. Isto é feito em duas etapas. Primeiramente, o conhecimento é extraído do especialista e modelado, utilizando alguma técnica de representação de conhecimento. Para isto, utiliza-se de um especialista em modelagem de conhecimento, que é chamado de Engenheiro de Conhecimento [MONARD (1986)]. A partir dessa modelagem, o implementador da Base de Conhecimento adiciona o conhecimento na forma exigida pelo sistema.

O processo de aquisição de conhecimento é lento, sendo considerado o ponto crítico na construção de Sistemas Especialistas [SAWAKI (1991)]. Isto ocorre porque, geralmente, os especialistas humanos são peritos em sua área de atuação, mas apresentam certa dificuldade em descrever seus próprios processos de raciocínio.

Em RODRIGUES (1990), pode-se verificar que encontrar um Núcleo de Sistemas Especialistas que seja perfeitamente adequado às necessidades de uma aplicação específica é uma tarefa difícil. Normalmente, os Núcleos disponíveis comercialmente, aos quais temos acesso, são restritos, ou seja, não apresentam muitas das características importantes na implementação de tais sistemas. Por outro lado, um Núcleo que apresentasse todas essas características poderia tornar o sistema lento desnecessariamente, caso a aplicação envolvida não utilize todos os recursos oferecidos pelo Núcleo. Isto ocorre porque, em um Núcleo fechado, as características

desnecessárias não podem ser eliminadas, deixando o sistema lento, devido à quantidade excessiva de código [RODRIGUES & MONARD (1990)]. Em alguns casos, pode ocorrer também a existência de procedimentos inerentes ao núcleo que não se adequam à implementação de um determinado sistema. Por exemplo, o método usado para calcular o fator de certeza feito pelo núcleo pode ser diferente do desejado para uma determinada aplicação [MONARD & PRADO (1989)].

Por tais motivos, pode-se afirmar que, para o projetista de um Sistema Especialista, seria interessante que o Núcleo fosse maleável o bastante para melhor se adequar à tarefa na qual é utilizado. Uma possível solução, para dotar um Núcleo de Sistemas Especialistas com tais características, é montá-lo como um conjunto de subsistemas que possam ser alterados, retirados e, inclusive, trocados por outros subsistemas [MONARD & RODRIGUES (1989), FRANCO & MONARD (1992b)].

2.2.2 Sistemas de Informação Geográfica

Os modelos matemáticos bidimensionais utilizam informações no domínio espacial sobre diversos aspectos da superfície, tais como: altitude, natureza, propriedades hidráulicas, e outros. Atualmente, utilizam-se sistemas computacionais específicos para trabalhar com essa multiplicidade de informações, denominados Sistemas de Informação Geográfica.

Um Sistema de Informação Geográfica (SIG) é um sistema gerenciador de base de dados digitais projetado para armazenar grandes quantidades de informações espacialmente distribuídas, provenientes de fontes diversas. Essas informações podem ser armazenadas, recuperadas, analisadas e apresentadas graficamente, dentro das especificações do usuário. Deste modo, podem ser utilizados para facilitar a compilação de dados espaciais e parâmetros hidrológicos necessários em diversas aplicações [JOHNSON et al. (1988), BHASKAR et al. (1993)].

Os SIGs são constituídos por uma série de programas e processos de análise, dentre os quais encontram-se facilidades para entrada de dados geográficos e visualização gráfica, entre outras. Assim, um SIG pode ser visto de forma geral como um sistema composto por:

- uma base de dados, contendo informação espacial
- um conjunto de programas dedicados à execução de operações sobre os dados (análise espacial)
- equipamento computacional especializado

Os SIGs baseiam-se em tecnologia de armazenamento, análise e tratamento de dados espaciais, não espaciais e temporais, além da geração de informações correlatas. As aplicações de SIG vão da coleta e tratamento da informação espacial ao desenvolvimento de novos sistemas e aplicações. A tecnologia envolve equipamentos e programas com diversos níveis de sofisticação e suas aplicações são inúmeras. A característica principal destes sistemas é focalizar o relacionamento de determinado fenômeno da realidade com sua localização espacial, permitindo relacionar fenômenos distintos e analisar aspectos mais complexos, como vizinhança e contiguidade. Além disso, fenômenos distintos podem ser representados de forma interativa e evolutiva, abrangendo todo o sistema, inclusive no aspecto temporal.

Devido à sua versatilidade, os SIGs estão se tornando cada vez mais necessários nas empresas, institutos de pesquisas e órgãos públicos, utilizados em planejamento urbano, gerenciamento de recursos, por cientistas, engenheiros civis e ambientais para análise, estimativa, planejamento e modelagem [BHASKAR et al. (1993)]. Há ainda os SIGS que são desenvolvidos para aplicações específicas, integrando dados digitais a técnicas de análise [VANBLARGAN et al. (1990), DJOCKIC & MAIDMENT (1991)]. No futuro, devido ao crescente aumento da população e a escassez de recursos naturais, os SIGS se tornarão indispensáveis.

Atualmente, os SIGS possuem um complexo sistema de hardware e software, fornecendo um ambiente integrado, através de facilidades de editoração de texto, aquisição de dados e visualização gráfica, auxiliando na elaboração de projetos e monitoramento. Segundo FORTIN & BERNIER (1991) e SCHULTZ (1993), esses sistemas foram possíveis graças aos avanços nas áreas de sensoriamento remoto, computação gráfica e desenvolvimento de ambientes interativos. Uma das tendências mais recentes na evolução da tecnologia de SIG é a inclusão de técnicas de Inteligência Artificial, principalmente Sistemas Especialistas, nas fases de projeto e operação [QIAN et al. (1990), SMITH et al. (1990), ZEHANA et al. (1991)].

2.2.3 Integrando SIG e SSD

Os Sistemas de Informação Geográfica tem sido utilizados em diversas aplicações, tanto isoladamente quanto aliado a tecnologias de Sistemas de Suporte de Decisão (SSD), tais como Sistemas Especialistas (SE), Pacotes Estatísticos, Programas de Otimização e Recursos Gráficos [LEIPNIK et al. (1993)]. Essa integração leva a uma nova classe de sistemas de computador, que auxiliam a tomada de decisão em problemas que apresentam dimensão espacial, chamados Sistemas de Suporte de Decisão Espacial (SSDE).

Os SSD consistem de uma interface com o usuário, uma base de conhecimento e uma base de dados. Eles são projetados para ajudar a resolver problemas complexos e são orientados para auxiliar na tomada de decisões. Os avanços no desenvolvimento de SSD para auxiliar no projeto de recursos hídricos conduzem de modo natural em direção à integração de tais sistemas com SIG [WALSH (1993)].

Os SIGS, por sua vez, caracterizam-se pela capacidade de análise e gerenciamento de dados distribuídos espacialmente, auxiliando no armazenamento de dados, organização, análise e visualização. Tanto SIG quanto SSD têm contribuído para a

pesquisa, o gerenciamento e o desenvolvimento na área de Recursos Hídricos. Deste modo, o poder de análise espacial e visualização, que caracterizam os SIGS, podem aperfeiçoar cada componente de um SSD, evoluindo desta forma para um SSDE [WALSH (1993)].

Normalmente, os SSD, principalmente os Sistemas Especialistas, são desenvolvidos a partir de Núcleos (Shells), os quais, em geral, consistem de pacotes fechados. No entanto, é difícil encontrar um Núcleo que se adeque perfeitamente às necessidades de uma aplicação específica [FRANCO (1991)]. No caso de integração com outros sistemas, o problema pode se agravar ainda mais, já que não se tem acesso às características internas desse Núcleo. Esse problema pode ser contornado construindo-se um Núcleo específico a partir da linguagem Prolog utilizando a meta-programação, obtendo-se, deste modo, o domínio completo das características internas desse Núcleo [FRANCO & MONARD (1992b)]. Além disso, a utilização de Núcleos construídos a partir de meta-interpretadores permite obter maior eficiência após a integração do Núcleo com as regras de conhecimento utilizando Avaliação Parcial [FRANCO & MONARD (1992a)].

A construção de SSDEs utilizando de Núcleos construídos a partir da linguagem Prolog permite também utilizar de modo integrado as facilidades fornecidas pela programação lógica, como por exemplo definir linguagens de programação específicas para aplicações hidrológicas [MONARD & FRANCO (1990)]. A versatilidade da programação lógica, principalmente no que se refere a técnicas de busca, a torna eficaz também em problemas de transportes e de fluxo em redes [FRANCO & MONARD (1993)].

→ Sistema de Suporte de Decisão Espacial

2.2.4 Aplicações de SSDE em Recursos Hídricos

A utilização de SIG na análise de dados hidrológicos tem fornecido muitos benefícios na gestão e projeto de recursos hídricos [ROSS & TARA (1993)]. Graças à facilidade de representar em forma digital as características usadas em modelos hidrológicos, SIG e DEM vem sendo utilizados na análise hidrológica [DEVANTIER & FELDMAN (1993)].

Além das informações relacionadas à descrição da topografia, SIG permite armazenar diversas informações topológicas, como declividade, irregularidade da superfície, tipo de solo e outros. Muitas vezes esses atributos são utilizados para descrever a capacidade de retenção de água em determinada região. Essa capacidade de armazenamento diversificado de SIG permite sua integração com Sistemas Especialistas, obtendo dessa forma um SSDE [DEVANTIER & FELDMAN (1993)].

A utilização de um SSDE auxilia sobremaneira o planejamento a longo prazo. Essa ferramenta permite incorporar regras lógicas e perícias de especialistas em planejamento de Recursos Hídricos aliadas à capacidade de armazenar e analisar dados distribuídos espacialmente [MCKINNEY & MAIDMENT (1993)]. Um SSDE permite ainda aproveitar as características da lógica nebulosa, definindo regras não determinísticas para representar o conhecimento. O método foi utilizado em modelagem eco-hidrológica [DROESEN & GEELLEN (1993)].

No caso de águas subterrâneas, um SSDE pode auxiliar na previsão para abastecimento doméstico, industrial e na agricultura, nos projetos hidráulicos e diversas outras aplicações [FÜRST et. al (1993)]. SSDEs podem ser também utilizados para calcular a extensão da degradação ambiental em uma bacia hidrográfica a partir de dados aeroespaciais de terreno. A partir destes dados, pode-se, por exemplo, modelar a erosão e a sedimentação, calculando-se então a

degradação do terreno e as fontes de sedimentos [MEIJERINK et al. (1993)]. Neste caso, utilizando-se de uma base de dados relacional, a integração com regras de conhecimento torna-se muito simples.

O sistema RAISON é um exemplo de SSDE que integra modelagem, base de dados, SIG, métodos estatísticos, sensoriamento remoto por imagens de satélite e sistema especialista baseado no algoritmo de árvore de decisão ID3 [LAM & SWAYNE (1993)]. O algoritmo ID3, que é um método de aprendizado de máquina idealizado por Quinlan, acha regras a partir de exemplos dados construindo árvores de decisão [CASTIÑEIRA 1990)].

O projeto e a segurança de barragens necessita de técnicas hidrológicas eficazes, principalmente nas estimativas de vazão. O pacote computacional RADMAX utiliza dados de precipitação observados por radares para a transposição e maximização de tempestades severas reais. A precipitação máxima provável e a enchente máxima provável são estimadas utilizando-se sofisticados modelos matemáticos distribuídos, para a transformação da chuva em vazão. Nele, propõe-se a utilização de técnicas de Inteligência Artificial e Sistemas especialistas para reduzir a subjetividade envolvida no processo de escolha de uma vazão a ser adotada para o projeto de uma barragem [PESSOA (1991)].

Segundo FRYSSINGER et. al (1993), o Sistema de Suporte de Decisão Ambiental Sandia constitui-se de um SSDE dirigido para auxiliar as decisões com respeito a aspectos hidrológicos no gerenciamento de resíduos perigosos. O sistema está sendo desenvolvido em conjunto pelos laboratórios da Sandia National e ATT Bell. O sistema tem por objetivo reduzir a subjetividade nos processos decisórios relativos ao meio ambiente utilizando resultados probabilísticos, já que as leis que regularizam tais gerenciamento em geral não possuem um grau de clareza determinístico. Por outro lado, RULAND et al. (1993) utilizou, na resolução de problemas relacionados com hidráulica ambiental, um SIG orientado a objetos, o Smallworld. Esse sistema foi desenvolvido a partir da linguagem de programação orientada a objetos Magik.

A previsão de cheias e a busca de alternativas para o gerenciamento das enchentes também pode ser auxiliada por SSDEs. Uma arquitetura SSDE proposta por SIMONOVIC (1993) utiliza um Sistema Especialista construído sobre o Nexpert Object, utilizando um módulo de base de dados relacional Oracle, um módulo de apresentação utilizando o Sistema de Informações Geográficas GRASS e dois sistemas de modelagem para a construção da base de conhecimento: o sistema de análise de prejuízo de enchentes (SID) e o programa para auxiliar a avaliação econômica no gerenciamento de enchentes (EAD).

Uma outra forma de utilização dos SSDEs é apresentada por FEDRA (1993), que mostra como as tecnologias de hipertexto e multimídia podem ser integradas aos SSDE para fornecer recursos audiovisuais, permitindo sua utilização como ferramenta de ensino.

2.2.5 Desafios dos SSDEs

Os problemas na construção de SSDEs residem no fato da necessidade de especialistas em diversas áreas de conhecimento: programação SIG, modelagem em recursos hídricos, projetos de SSD, gerenciamento de recursos hídricos, planejamento e outros.

Deste modo, SSDEs necessitam de colaboração interdisciplinar para resolver a complexidade da integração das diversas tecnologias. Inicialmente, os projetistas de SSD podem direcionar o trabalho, pois já possuem a experiência em análise, projeto e interação com o usuário [WALSH (1993)]. Neste caso, o projeto pode ser completamente definido de forma a distribuir as funções aos especialistas mais indicados, no projeto da base de dados, na aquisição e representação do conhecimento, na interface com o usuário, na elaboração das formas de entrada de dados e nas formas de apresentação dos resultados, além da integração com outros sistemas.

2.3 Modelos de Elevação Digital

A representação digital da variação espacial do relevo é conhecida como Modelo de Elevação Digital (DEM¹) ou Modelo Digital de Terreno (DTM²). Apesar de originariamente os DEMs terem sido desenvolvidos somente para modelar relevo, eles podem também ser utilizados para modelar a variação contínua de qualquer outro atributo sobre uma superfície bidimensional [BURROUGH (1987)].

Alguns resultados recentes, apresentados a seguir, podem ajudar a visualizar de maneira mais efetiva as aplicações dos Modelos de Elevação Digital. CHAIRAT & DELLEUR (1993) destacam a utilização de modelos hidrológicos baseados em topografia integrados a sistemas de suporte de análise de recursos geográficos, resultando em uma ferramenta poderosa na determinação de parâmetros hidrológicos.

Em ELGY et al. (1993), pode-se verificar que a utilização de Modelos de Elevação Digital na modelagem de rede de drenagem urbana também alcançou resultados significativos, facilitando a geração de mapas, ligações com modelos de simulação e a apresentação gráfica dos resultados.

MOORE et al. (1993) afirmam que a análise de processos hidrológicos, ecológicos e geomorfológicos distribuídos espacialmente em superfícies de relevo complexas, integrada com análise de terreno para calcular declividade, direção de fluxo e bacias hidrográficas, além de outros processos hidrológicos, também torna-se possível com a utilização de DEMs.

¹ Digital Elevation Model

² Digital Terrain Model

Segundo LA BARBERA et al. (1993), a obtenção da rede de canais pode ser simples em alguns casos, mas encontra limitações em casos de estruturas morfológicas complexas. RIEGER (1993) destaca que, na obtenção de bacias de drenagem e rede de canais, algumas vantagens podem ser obtidas na utilização de uma estrutura de dados tipo pirâmide em lugar de grade regular .

O processo de derivação da hidrógrafa unitária a partir do modelo digital da bacia hidrográfica requer a definição de uma grade de direção de fluxo e conseqüente quantificação da velocidade de fluxo. A velocidade pode ser calculada a partir da declividade e da cobertura do solo [MAIDMENT (1993)]. Modelos de fluxos subterrâneos também podem ser combinadas com dados digitais de bacias hidrográficas, acumulação de águas e tempo de permanência na zona não saturada para auxiliar hidrogeólogos no cálculo de zonas de captura de águas subterrâneas [MERKEL & SPERLING (1993)].

A variação da elevação da superfície sobre uma dada área pode ser modelada de diversas formas. As principais formas de representação são grade regular, rede triangular irregular e linhas de contorno.

2.3.1 Grade Regular

É a forma mais comum de DEM, também conhecida como matriz de altitude ou raster. Segundo DEVANTIER & FELDMAN (1993), a grade é constituída de linhas espaçadas regularmente, sendo cada retângulo identificado pelas coordenadas do ponto central. Devido à facilidade de tratamento de matrizes por computadores, a grade regular é a forma que apresenta maiores aplicações. No entanto, apresenta algumas desvantagens:

- áreas de terreno uniforme apresentam muitos dados redundantes;
- dificuldade de adaptação em áreas de relevo de complexidade diferenciada, requerendo alteração no tamanho da grade;
- ênfase exagerada ao longo do eixo da grade em certos tipos de cálculo, tais como linhas de cálculo de visão panorâmica.

2.3.2 Rede Triangular Irregular (TIN³)

Segundo BURROUGH (1987), a rede triangular irregular é um sistema para DEM que evita as redundâncias da grade regular, além de ser mais versátil na representação de terrenos irregulares. No entanto, não deixa de ser versátil no aspecto computacional. O modelo utiliza uma camada de facetas de triângulos contínuos e conectados, baseados na triangulação de nós espaçados irregularmente de Delaunay.

Como as grades regulares, TIN permite o acréscimo de informações extras em áreas de relevo complexo sem a necessidade de acumular dados redundantes nas áreas de relevo simples. Conseqüentemente, o processo de captura de dados em um TIN pode seguir montanhas, linhas de fluxo e outras características topológicas importantes, que podem ser digitalizadas de acordo com a precisão exigida.

É possível extrair automaticamente um modelo TIN a partir de dados regulares, obtidos em forma digital de sensores [FOWLER et al. (1979), BAND (1989)]. Os métodos de geração automática de redes triangulares baseiam-se, em geral, na triangulação de Delaunay [LEE & SCHACHTER (1980), WATSON & PHILIP (1984a), WATSON & PHILIP (1984b)].

O modelo TIN é uma estrutura topológica semelhante à vetorial, no que concerne ao conceito de estruturas totalmente definidas topologicamente, com a diferença que no modelo TIN não é necessário estipular condições para os casos de ilhas ou buracos.

³ Triangular Irregular Network

Este modelo considera os nós da rede como entidades primárias da base de dados, onde as relações topológicas são construídas utilizando ponteiros de cada nó para os nós vizinhos. A lista de vizinhos é ordenada no sentido horário ao redor do nó, iniciando ao norte.

A região fora da área modelada pelo TIN é representada por um nó fictício no lado contrário da esfera topológica sobre a qual o TIN é projetado. Este nó fictício auxilia na descrição da topologia dos pontos do contorno e simplifica seu processamento.

2.3.3 Linhas de Contorno (Vetorial)

A terceira importante forma de representação topológica é o mapeamento por linhas de contorno ou estruturas vetoriais. A representação vetorial de um elemento é uma tentativa de reproduzi-lo o mais exatamente possível, supondo o espaço contínuo, o que permite uma precisão maior na definição das posições, distâncias e áreas. Os métodos vetoriais assumem que as coordenadas dos pontos são matematicamente exatas. TEIXEIRA et al. (1992) afirma que, em forma vetorial, qualquer elemento pode ser reduzido a três formas básicas: ponto, linhas e polígonos.

2.3.4 Aplicações do modelo DEM-TIN

Quando a rede de drenagem e as linhas de fluxo não são conhecidas previamente, pode ser necessário derivá-las a partir do relevo, utilizando para isso métodos computacionais adequados [COUGER et al. (1992), MARTZ & GARBECHT (1992)]. A extração automática da topografia do terreno utilizando DEM é um procedimento que pode ser realizado facilmente para as aplicações gerais de modelos chuva-vazão distribuídos [JENSON & DOMINGUE (1988)].

Os algoritmos geralmente utilizados deduzem a rede de drenagem a partir de modelos de grade regular. Esses algoritmos, no entanto, apresentam uma grande falha: a menos que o terreno seja acidentado, as linhas de fluxo tendem a correr em linhas paralelas, ao longo das direções geradas pela orientação da grade da amostragem [GOODRICH et. al (1991)].

Deve-se levar em conta também que a representação topográfica no modelo baseado em triângulo é mais precisa, enquanto o modelo baseado em retângulos envolve interpolação dos dados originais ponto a ponto, acarretando perda de detalhes [JONES & WRIGHT (1990)]. Além disso, os triângulos podem designar planos ou facetas, sendo adaptável a qualquer configuração de terreno. FAIRFIELD & LEYMARIE (1991) apresenta uma solução para o problema, onde determinam-se as linhas de fluxo de forma independente da orientação da grade, utilizando para isso métodos estocásticos. No entanto, essa solução também apresenta problemas, como convergência lateral ao invés de fluxo paralelo, como aponta COSTA-CABRAL & BURGÉS (1994), que resolve o problema utilizando caminhos de fluxo internos aos retângulos. Uma outra alternativa é apresentada por Tarboton (1997) calcula a direção de fluxo calculando a declividade entre a região retangular e cada um de seus 8 vizinhos. A declividade que apresenta maior valor é adotada para a direção do fluxo.

No entanto, GOODRICH et. al (1991) ressalta uma série de vantagens apresentadas pelo modelo TIN na modelagem da topografia. Em alguns casos, são necessários de 14 a 250 pontos de grade regular para cada ponto de TIN. Além disso, a propagação cinemática sobre uma grade regular apresenta problemas de convergência e divisão de fluxo, enquanto que o modelo TIN supera tais dificuldades.

Muitas vezes, os dados obtidos para designar o terreno são constituídos de um conjunto de pontos que são agrupados em um TIN. Em certos casos, o mapa de contorno é gerado a partir dele. Porém, JONES & WRIGHT (1990) provam que melhores resultados são obtidos trabalhando-se diretamente com o modelo TIN.

Em JONES & WRIGHT (1990), apresenta-se um algoritmo para traçar o caminho de maior declividade a partir de um dado ponto inicial sobre um terreno definido por um TIN. A direção inicial do fluxo dependerá essencialmente do gradiente da superfície e de sua irregularidade. As direções de fluxo subsequente dependerão tanto do gradiente em cada ponto e do momento. Deste modo, calculam-se as linhas de fluxo de água potenciais. A partir das direções de fluxo, é calculada a rede de canais. Em seguida, são delineados os contornos de drenagem e subdivisões em triângulos. Deste modo, pode-se agrupar os triângulos em áreas fonte. Conhecendo-se as áreas fonte, é possível delinear a bacia hidrográfica da seguinte forma: a bacia hidrográfica correspondente a um nó na rede de canais é a área que poderia contribuir potencialmente para o fluxo através do nó; a bacia hidrográfica para um nó delimitado é simplesmente a soma das áreas fontes com fluxo acima do nó. NELSON et al. (1994) declara que este procedimento pode ser utilizado em conjunto com métodos de modelagem hidrológica para criar um ambiente integrado de modelagem hidrológica.

PALACIOS-VELEZ & CUEVAS-RENAUD (1986), também utilizando um modelo de facetas triangulares TIN-DEM, desenvolveram um algoritmo que leva em conta a trajetória do caminho de maior declividade à montante ou à jusante, partindo de qualquer ponto da superfície topográfica. Também leva em conta a definição de bacias e sub-bacias para qualquer ponto do rio. Além disso, mostra que a grade regular pode ser tratada como um caso particular do modelo TIN, generalizando o modelo apresentado. Baseado nesse modelo, PALACIOS-VELEZ & CUEVAS-RENAUD (1986) desenvolveram um sistema computacional baseado em facetas triangulares (denominado SHIFT) que permite:

- edição e visualização de uma bacia armazenada em DEM-TIN;
- entrada e interpolação de dados de solo e chuva;
- cálculo e propagação de fluxo de vazão ao longo das facetas.

Segundo PALACIOS-VELEZ & CUEVAS-RENAUD (1992), o sistema determina a sequência de processamento dos elementos e identifica as facetas que fazem parte da área de contribuição em cada segmento de canal .

2.4 Métodos Numéricos para Equações Diferenciais

Os modelos hidrodinâmicos, seja qual for o grau de simplificação, dificilmente podem ser resolvidos analiticamente. Nestes casos, eles devem ser resolvidos numericamente para aplicações reais. Na solução numérica, o espaço contínuo é dividido em regiões discretas, de modo que a solução para cada região possa ser encontrada.

PRICE (1978) explica que primeiro estudo numérico importante de propagação de cheias utilizando computadores utilizou o método das características, adotando soluções explícitas para as equações características. Neste método, as equações diferenciais parciais são transformadas, cada uma delas, em um par de equações ordinárias [STRELKOFF (1970) , HUANG & SONG (1985)]. Muitos trabalhos foram realizados posteriormente utilizando esse método, adotando diferentes graus de simplificação e diversas formas de tratar o problema [WOOLHISER & LIGGETT (1967), SIVALOGANATHAN (1978)].

VEMURI et al. (1981) apresenta outro método numérico muito utilizado na resolução de problemas envolvendo equações diferenciais: o método das diferenças finitas. Neste método, as derivadas são substituídas por fórmulas discretas, utilizando aproximações da expansão por séries de Taylor. Em seguida, a equação é aplicada às diversas regiões discretas, resultando em um sistema de equações cuja solução fornece o valor numérico aproximado para cada região.

Segundo PRICE (1978), o método das diferenças finitas pode ser dividido em dois esquemas, de acordo com a maneira como é feita a discretização em relação à derivada temporal: esquemas explícitos e implícitos. Os esquemas explícitos, bem como o método das características, apresentam problemas de instabilidade. VIEUX et

al. (1990) explica que o intervalo de tempo e espaço devem obedecer à condição de Courant:

$$\Delta t = \frac{\Delta x}{\sqrt{gh}} \quad (2.4-1)$$

Os inconvenientes causados pela instabilidade podem ser reduzidos pela utilização de esquemas implícitos. Estes métodos, além de serem mais estáveis, permitem a utilização de intervalos de tempo maiores. No entanto, HUANG & SONG (1985) adverte que para se encontrar a solução são necessárias muitas iterações, tornando o procedimento excessivamente lento.

TAYFUR et al. (1993) explica que o método das diferenças finitas também pode ter diferentes esquemas em relação à discretização espacial: esquema progressivo, regressivo e centrado. A maior deficiência do método das diferenças finitas consiste justamente na discretização espacial, que deve ser feita através de malhas retangulares, dificultando sua aplicação a geometrias irregulares. Na solução através do método das diferenças finitas observa-se que, devido à não linearidade das equações de fluxo, freqüentemente não há convergência em superfícies muito irregulares com alterações abruptas na declividade. Nestes casos, deve-se alterar a topografia, trocando-a por uma superfície mais lisa, utilizando somente os valores médios.

Outro método de solução numérica que adquiriu grande importância é o método dos elementos finitos. Neste método, o espaço contínuo é subdividido em regiões discretas (como no método das diferenças finitas). Porém, neste caso, cada região é denominada elemento e definida matematicamente, de modo que, quando a solução para cada elemento é encontrada, é possível saber o valor em cada ponto interno da região representada por esse elemento [VEMURI et al. (1981)].

Diversos estudos utilizaram o método dos elementos finitos na resolução dos modelos hidrodinâmicos para o caso unidimensional [ROSS et al. (1979), TAYLOR

Diversos estudos utilizaram o método dos elementos finitos na resolução dos modelos hidrodinâmicos para o caso unidimensional [ROSS et al. (1979) , TAYLOR et al. (1974)]. Muitas vezes, a solução no domínio do espaço utiliza o método dos elementos finitos, enquanto o problema no domínio do tempo é resolvido utilizando-se o método das diferenças finitas [JAYAWARDENA & WHITE (1977) , JAYAWARDENA & WHITE (1979)].

A propagação de fluxo no caso unidimensional pode ser utilizado para o cálculo de bacias hidrográficas, desde que se considere a bacia como um agrupamento de bacias unidimensionais, que devem ser calculadas individualmente e, ao final, somar as contribuições de cada uma [JAYAWARDENA & WHITE (1979)]. Porém, essa aproximação exige uma interpretação subjetiva na divisão em bacias unidimensionais, que deve ser considerada [GOODRICH et al. (1993)]. A aplicação do método dos elementos finitos a modelos hidrodinâmicos em superfície bidimensional simples, em estudos feitos por TAYLOR et al. (1974) , apresentou estabilidade. Porém, GOODRICH et al. (1993) afirma que nesse estudo não se tratou de aplicações a bacias hidrográficas.

VIEUX et al. (1990) garante que o método dos elementos finitos pode ser aplicado a qualquer modelo de terreno, independente de sua configuração geométrica. Uma outra vantagem desse método é que o sistema de equações final apresenta matrizes diagonais e simétricas, permitindo métodos de solução e armazenamento compactos. Além disso, a ausência de restrições com relação à geometria da região torna esse método mais indicado para estudo de problemas bidimensionais.

A capacidade de tratar problemas maiores e mais complexos tem aumentado as aplicações do método dos elementos finitos. De acordo com GOODRICH et al. (1993), a aplicação desse método ao caso bidimensional apresenta ainda a possibilidade de integração com modelos de elevação digital que utilizam rede triangular irregular (modelo DEM-TIN).

2.4.1 Método das Diferenças Finitas

O método das diferenças finitas consiste em discretizar o espaço-tempo contínuo e aplicar as aproximações obtidas a partir de aproximações por série de Taylor.

Considere-se uma função $f'(x) = g(x, f)$. A expansão de $f(x)$ em uma série de Taylor para o ponto $x = x_0$ é:

$$f(x) = f(x_0) + (x - x_0) \frac{df}{dx} + \frac{(x - x_0)^2}{2!} \frac{d^2 f}{dx^2} + \dots \quad (2.4-2)$$

Seja $x = x_0 + \Delta x$. A equação (2.4-1) pode ser reescrita da seguinte forma:

$$\frac{df}{dx} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} + \xi \quad (2.4-3)$$

O cálculo de uma solução aproximada de uma equação diferencial, utilizando a equação (2.4-2) é chamado Método de Euler. Existem diversos esquemas de utilização do método das diferenças finitas, cuja diferença é a técnica de discretização utilizada, tanto para a variável espacial quanto para a variável temporal. Quanto à dimensão espacial, o esquema de diferenças finitas pode ser progressivo, regressivo ou centrado; quanto à temporal, implícito ou explícito. Em cada um dos esquemas, a substituição da derivada pela diferença entre variáveis discretas é feita de uma maneira distinta.

No esquema regressivo, utiliza-se a diferença entre o ponto atual e seu antecessor imediato:

$$\left. \frac{df}{dx} \right|_{x=i} = \frac{f_i - f_{i-1}}{\Delta x} \quad (2.4-4)$$

No esquema progressivo, utiliza-se a diferença entre o ponto atual e seu sucessor imediato:

$$\left. \frac{df}{dx} \right|_{x=i} = \frac{f_{i+1} - f_i}{\Delta x} \quad (2.4-5)$$

No esquema centrado, utiliza-se a diferença entre o sucessor e o antecessor imediato do ponto atual:

$$\left. \frac{df}{dx} \right|_{x=i} = \frac{f_{i+1} - f_{i-1}}{2\Delta x} \quad (2.4-6)$$

Segundo QUEIROZ (1991), os esquemas explícitos e implícitos diferenciam-se pela utilização dos valores da variável no tempo atual para o cálculo dos novos valores a cada iteração. No esquema explícito, somente uma variável discreta encontra-se no tempo seguinte ($k+1$) e todas as outras no tempo atual (k). Neste caso, o cálculo torna-se simples, pois há uma única variável desconhecida em função de valores conhecidos.

No esquema implícito, utiliza-se uma combinação de valores das variáveis no tempo atual e no tempo seguinte para o cálculo dos novos valores. O esquema pode ser totalmente implícito, quando são utilizados os valores no tempo seguinte para todas as variáveis fora da derivada temporal, ou parcialmente implícito, quando se utilizam os valores médios (entre o tempo atual e o tempo seguinte) para as variáveis fora da variável temporal.

Para exemplificar, considere-se a equação a seguir:

$$\frac{df}{dt} + \frac{df}{dx} = C \quad (2.4-7)$$

O termo $\frac{df}{dx}$ será substituído utilizando esquema regressivo. O termo $\frac{df}{dt}$ é substituído pela diferença entre o valor no tempo seguinte e o valor no tempo atual:

$$\frac{df}{dt} = \frac{f^{k+1} - f^k}{\Delta t} \quad (2.4-8)$$

A equação completa depende do método escolhido: explícito ou implícito. No método explícito, utiliza-se, para as variáveis discretas associadas à derivada espacial, os valores no tempo atual:

$$\frac{f_i^{k+1} - f_i^k}{\Delta t} + \frac{f_i^k - f_{i-1}^k}{\Delta x} = C \quad (2.4-9)$$

No método totalmente implícito, utiliza-se, para as variáveis discretas associadas à derivada espacial, os valores no tempo seguinte:

$$\frac{f_i^{k+1} - f_i^k}{\Delta t} + \frac{f_i^{k+1} - f_{i-1}^{k+1}}{\Delta x} = C \quad (2.4-10)$$

No método parcialmente implícito, utiliza-se, para as variáveis discretas associadas à derivada espacial, combinações entre os valores no tempo atual e o tempo seguinte:

$$\frac{f_i^{k+1} - f_i^k}{\Delta t} + \frac{\bar{f}_i - \bar{f}_{i-1}}{\Delta x} = C \quad (2.4-11)$$

onde:

$$\begin{aligned} \bar{f} &= \alpha f^{k+1} + \beta f^k \\ \alpha + \beta &= 1 \end{aligned} \quad (2.4-12)$$

2.4.2 Método dos Elementos Finitos

O método dos elementos finitos é um procedimento numérico para resolver problemas cuja formulação matemática envolve equações diferenciais da seguinte forma :

$$\begin{aligned} \mathbf{L}u(X) &= \mathbf{0} & X \in \Omega \\ u(X) &= g(X) & X \in \Gamma \end{aligned} \quad (2.4-13)$$

onde:

\mathbf{L} é um operador diferencial

u é a variável independente

Ω é a região

Γ é o contorno

$$X = (x_1, x_2, \dots, x_n)$$

O método dos elementos finitos pode ser dividido em cinco passos:

1. Discretizar a região, definindo os elementos que a compõem
2. Especificar a função de interpolação, escrevendo uma equação para cada elemento
3. Desenvolver o sistema de equações, utilizando as equações individuais descritas para cada elemento
4. Resolver o sistema de equações
5. Calcular os valores de interesse

Da mesma forma como a região é dividida em elementos, a função também deve ser aproximada por uma função discreta para ser aplicada a cada um dos elementos. SEGERLIND (1984) ressalta que a escolha da função de aproximação deve ser criteriosa, pois dela depende o sucesso do método.

Uma forma conveniente de escrever a função de aproximação linear \hat{u} é:

$$\hat{u}(x) = \sum_{i=1}^n u_i N_i(x) \quad (2.4-14)$$

onde:

u_i é o valor da função u no vértice i

n é o número de vértices

$N_i(x)$ são funções básicas de aproximação

2.4.2.1 Métodos de Resíduos Ponderados

No método dos elementos, é necessário formular aproximações das equações. Isto pode ser feito utilizando-se o método variacional ou métodos de resíduos ponderados, dentre os quais destaca-se o método de Galerkin. Nos métodos de resíduos ponderados, a função de aproximação é substituída na equação diferencial. Essa função, sendo uma aproximação da função original, não satisfaz a equação. Assim, haverá uma diferença, que é denominada resíduo (R). Para se obter a melhor solução, deve-se minimizar o resíduo ao longo de toda a região:

$$\text{Minimizar } \int_{\Omega} R \, d\Omega \quad (2.4-15)$$

Para facilitar a tarefa de minimização, utiliza-se um fator de ponderação w junto ao resíduo. Esse fator permite que a minimização atinja o valor zero na integração do resíduo [VEMURI et al. (1981)]. Deste modo, a equação a ser resolvida é:

$$\int_{\Omega} wR \, d\Omega = 0 \quad (2.4-16)$$

Os diversos métodos de resíduos ponderados diferenciam-se pela forma como é escolhido o fator de ponderação w . Dentre eles, o mais importante é o método de Galerkin, que utiliza as mesmas funções de aproximação $N(x)$ como fator de ponderação na integração do resíduo [BREBBIA & FERRANTE (1975), WAIT & MITCHELL (1985)].

$$w_i = N_i(x) \quad (2.4-17)$$

Desta forma, a equação (2.4-15) pode ser escrita da seguinte forma:

$$\int_{\Omega} NR \, d\Omega = 0 \quad (2.4-18)$$

$$\text{onde } N = [N_1 \quad N_2 \quad \dots \quad N_n]^T$$

2.4.2.2 Representação Matricial

Após a resolução da integral na equação (2.4-17), obtém-se um sistema de equações envolvendo todos os nós da superfície. Cada linha do sistema representa a contribuição recebida pelo nó i de todos os outros nós, e pode ser escrita da seguinte forma:

$$\sum_{j=1}^n G_{i,j} u_j = s_i \quad ; \quad i=1..n \quad (2.4-19)$$

O conjunto de equações pode ser representado na seguinte forma matricial:

$$\mathbf{G}\mathbf{u}=\mathbf{s} \quad (2.4-20)$$

onde:

\mathbf{G} é a matriz quadrada dos coeficientes

\mathbf{u} é um vetor coluna contendo os nós da região discretizada (vértices dos elementos)

\mathbf{s} é um vetor coluna com as condições de contorno

Um dos métodos de montar o sistema de equações consiste em analisar cada elemento isoladamente, verificando a contribuição individual de cada nó que compõe o elemento. Para isso, aplica-se, a cada elemento, a equação (2.4-17). A contribuição de cada elemento será somente para os nós que compõem o elemento. Esse conjunto de contribuições pode ser representado por uma matriz quadrada, cujo tamanho corresponde ao número de nós que compõem cada elemento. Se os elementos forem triângulos, a matriz será 3×3 ; se forem quadriláteros, será uma matriz 4×4 .

Considerando elementos com 4 nós (denominados i,j,k,m), a matriz de contribuições para o elemento e será:

$$\mathbf{G}^e = \begin{bmatrix} G_{i,i}^e & G_{i,j}^e & G_{i,k}^e & G_{i,m}^e \\ G_{j,j}^e & G_{j,j}^e & G_{j,k}^e & G_{j,m}^e \\ G_{k,i}^e & G_{k,j}^e & G_{k,k}^e & G_{k,m}^e \\ G_{m,i}^e & G_{m,j}^e & G_{m,k}^e & G_{m,m}^e \end{bmatrix} \quad (2.4-21)$$

Após a avaliação de todos os elementos, a contribuição dos nós é totalizada somando sua contribuição em cada elemento. A matriz G completa é obtida da seguinte forma:

$$G_{l,c} = \sum_e G_{l,c}^e \quad \forall l,c \quad (2.4-22)$$

2.4.3 Métodos de Runge-Kutta

O método de Euler requer uma amplitude de intervalo muito pequena para possibilitar uma precisão razoável. Os métodos de Runge-Kutta procuram obter uma maior precisão e, ao mesmo tempo, evitam a necessidade de derivadas de ordens mais elevadas [CONTE (1965)].

Existem diversos métodos de Runge-Kutta, de diversas ordens e diferentes precisões. Dentre eles, o mais popular e mais comumente usado é o algoritmo de Quarta Ordem, descrito a seguir:

Seja a equação $\frac{df}{dx} = g(x, f)$ e $x_{i+1} = x_i + \Delta x$. As aproximações da função são geradas utilizando a seguinte fórmula de recorrência:

$$\begin{aligned} f_{i+1} &= f_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= g(x_i, f_i)\Delta x \\ k_2 &= g\left(x_i + \frac{\Delta x}{2}, f_i + \frac{k_1}{2}\right)\Delta x \\ k_3 &= g\left(x_i + \frac{\Delta x}{2}, f_i + \frac{k_2}{2}\right)\Delta x \\ k_4 &= g(x_i + \Delta x, f_i + k_3)\Delta x \end{aligned} \quad (2.4-23)$$

3. Metodologia

3.1 Modelo TIN

Atualmente, as técnicas de aquisição de dados sobre elevação de terreno já estão bem desenvolvidas, sendo possível a utilização de diversas técnicas existentes. Segundo PALACIOS-VELEZ & CUEVAS-RENAUD (1986), a representação dessas informações em computadores é feita utilizando-se modelos de elevação digital (DEM), que são estruturas de dados para representar a configuração de uma superfície topográfica. As principais formas de armazenamento são: estruturas vetoriais, grade retangular (estrutura raster) e rede triangular irregular (TIN).

A estrutura raster, ou matriz de altitude, caracteriza-se pela facilidade de manipulação por computador, pois a grade pode ser considerada como uma matriz, onde cada elemento representa um retângulo da grade. Já as estruturas vetoriais podem representar o terreno de forma mais precisa, pois os elementos são compostos por polígonos, de forma que podem ser adaptados a qualquer configuração de terreno.

A rede triangular é uma técnica de representação alternativa com a finalidade de aliar as vantagens de uma representação precisa à facilidade de manipulação por computador. Neste modelo, a superfície é modelada como um conjunto de facetas triangulares adjacentes não sobrepostas, cujos vértices estão localizados adaptativamente nos locais onde a declividade modifica abruptamente. A precisão na representação deve-se ao fato da versatilidade dos triângulos, que pode se adaptar facilmente às mais irregulares configurações, modificando seu formato em cada caso. Apesar de sua

capacidade mutante com relação à forma, todo triângulo pode ser descrito da mesma forma, através das coordenadas de seus vértices. Além disso, qualquer ponto interno pode ser calculado aplicando-se a equação do triângulo às coordenadas desse ponto.

Em termos computacionais, destaca-se a economia de espaço no armazenamento e a relativa facilidade encontrada no cálculo. Segundo estimativas, para a obtenção da mesma precisão na representação de características geomorfológicas, são necessários de 14 a 250 pontos de uma grade retangular para um único ponto de TIN [PALACIOS-VELEZ & CUEVAS-RENAUD (1986)]. No entanto, deve-se observar que no modelo TIN, cada ponto utiliza mais espaço de armazenamento que na estrutura raster.

3.1.1 Representação de uma Rede Triangular

A geração de um modelo TIN é feita em duas fases:

1. a seleção dos pontos com a informação sobre a elevação do terreno
2. a conexão desses pontos em facetas triangulares.

Um exemplo de discretização de uma região utilizando o modelo TIN pode ser visto na Figura 3.1-1.

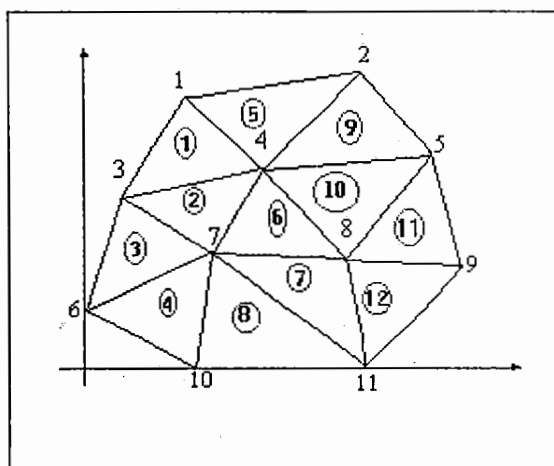


Figura 3.1-1: Exemplo de uma Rede Triangular

Após selecionar os pontos (que serão vértices dos triângulos), eles devem ser enumerados e armazenados com suas coordenadas x, y e z., como pode ser observado na Tabela 3.1-1.

Tabela 3.1-1: Coordenadas dos pontos da região

Vértice	x	y	z
1	2.3	6.8	7.0
2	6.8	7.5	7.3
3	1.0	4.5	5.9
4	4.5	5.2	6.2
5	8.5	5.4	5.8
6	0	1.5	4.2
7	3.2	2.9	4.8
8	6.4	2.9	5.1
9	9.3	2.5	4.5
10	3.0	0	3.3
11	7.0	0	3.1

Após a definição dos pontos, basta definir o relacionamento entre eles, visando a formação dos triângulos. Esse relacionamento pode ser feito de duas formas distintas. Na primeira forma armazena-se, para cada ponto, os pontos vizinhos, obedecendo o sentido horário.

A segunda forma de armazenamento usa a representação pelo método triangular, que enumera cada triângulo, definindo-o através dos vértices que o compõem e dos triângulos adjacentes a ele. Como cada triângulo possui 3 vértices e 3 triângulos adjacentes (com exceção no contorno), são utilizados 6 ponteiros para cada triângulo. Na Tabela 3.1-2 pode-se observar essa forma de armazenamento para a Figura 3.1-1.

Tabela 3.1-2: Definição dos triângulos e vizinhança

Triângulo	Vértices			Adjacentes		
1	1	4	3	5	2	A
2	4	7	3	6	3	1
3	3	7	6	2	4	A
4	7	10	6	8	C	3
5	2	4	1	9	1	B
6	4	8	7	10	7	2
7	8	11	7	12	8	6
8	11	10	7	C	4	7
9	2	5	4	B	10	5
10	5	8	4	11	6	9
11	5	9	8	B	12	10
12	9	11	8	C	7	11

Em termos de espaço requerido para o armazenamento, as estruturas se equivalem. Porém, a segunda forma é mais eficiente no processamento triângulo-a-triângulo e, portanto, mais indicada para o problema em questão neste trabalho.

3.1.2 Funções de Aproximação para Elementos Triangulares

Segundo MITCHELL & WAIT (1977), as funções básicas devem se adequar ao tipo de elemento geométrico escolhido para representar a região. No caso bidimensional, destaca-se o triângulo, devido a sua maleabilidade.

Cada elemento triangular possui três lados e três nós, um em cada vértice. A equação da superfície é:

$$\hat{u} = \alpha x + \beta y + \gamma \quad (3.1-1)$$

Os três nós são i, j e k . Os valores de \hat{u} nesses nós são, respectivamente, $\hat{u}_i, \hat{u}_j, \hat{u}_k$. Suas coordenadas são $(x_i, y_i), (x_j, y_j), (x_k, y_k)$. A substituição desses valores na equação (3.1-1) conduz ao seguinte sistema de equações:

$$\begin{cases} \hat{u}_i = \alpha x_i + \beta y_i + \gamma \\ \hat{u}_j = \alpha x_j + \beta y_j + \gamma \\ \hat{u}_k = \alpha x_k + \beta y_k + \gamma \end{cases} \quad (3.1-2)$$

A solução do sistema é:

$$\begin{cases} \alpha = \frac{(y_j - y_k)\hat{u}_i + (y_k - y_i)\hat{u}_j + (y_i - y_j)\hat{u}_k}{2A} \\ \beta = \frac{(x_k - x_j)\hat{u}_i + (x_i - x_k)\hat{u}_j + (x_j - x_i)\hat{u}_k}{2A} \\ \gamma = \frac{(x_j y_k - x_k y_j)\hat{u}_i + (x_k y_i - x_i y_k)\hat{u}_j + (x_i y_j - x_j y_i)\hat{u}_k}{2A} \end{cases} \quad (3.1-3)$$

Na equação (3.1-3), A é a área do triângulo, dada por:

$$A = \frac{x_i y_j + x_j y_k + x_k y_i - y_i x_j - y_j x_k + y_k x_i}{2} \quad (3.1-4)$$

A substituição de α, β e γ na equação (3.1-1), rearranjando os elementos de forma adequada, transforma-se na seguinte equação:

$$\hat{u} = N_i \hat{u}_i + N_j \hat{u}_j + N_k \hat{u}_k \quad (3.1-5)$$

Na equação (3.1-5), as funções de aproximação linear (N) são dadas por:

$$\begin{cases} N_i = \frac{a_i + b_i x + c_i y}{2A} \\ N_j = \frac{a_j + b_j x + c_j y}{2A} \\ N_k = \frac{a_k + b_k x + c_k y}{2A} \end{cases} \quad (3.1-6)$$

onde:

$a_i = x_j y_k - x_k y_j$	$b_i = y_j - y_k$	$c_i = x_k - x_j$
$a_j = x_k y_i - x_i y_k$	$b_j = y_k - y_i$	$c_j = x_i - x_k$
$a_k = x_i y_j - x_j y_i$	$b_k = y_i - y_j$	$c_k = x_j - x_i$

Esta forma de equacionar o problema facilita o cálculo das derivadas parciais, como pode ser observado nas equações (3.1-7) e (3.1-8).

$$\frac{\delta \hat{u}}{\delta x} = \frac{\delta N_i}{\delta x} \hat{u}_i + \frac{\delta N_j}{\delta x} \hat{u}_j + \frac{\delta N_k}{\delta x} \hat{u}_k = \frac{b_i \hat{u}_i + b_j \hat{u}_j + b_k \hat{u}_k}{2A} \quad (3.1-7)$$

$$\frac{\delta \hat{u}}{\delta y} = \frac{\delta N_i}{\delta y} \hat{u}_i + \frac{\delta N_j}{\delta y} \hat{u}_j + \frac{\delta N_k}{\delta y} \hat{u}_k = \frac{c_i \hat{u}_i + c_j \hat{u}_j + c_k \hat{u}_k}{2A} \quad (3.1-8)$$

3.2 Modelagem de Fluxo em um DEM-TIN

Na modelagem de fluxo, a superfície é subdividida em diversos elementos de superfície. Em geral, essas superfícies elementares são planares, e podem ser descritas pela seguinte fórmula:

$$z(x,y) = ax + by + c \quad (3.2-1)$$

O fluxo da água ocorre na direção da declividade decrescente, isto é, no sentido contrário ao gradiente.

$$-\text{grad } z = (a, b) \quad (3.2-2)$$

O ângulo de fluxo (ψ), no sentido anti-horário em relação a leste, pode ser qualquer valor entre 0 e 360°, e pode ser calculado através da seguinte expressão geral [COSTA-CABRAL & BURGESS (1994)]:

$$\begin{cases} a \neq 0, b \neq 0 \Rightarrow \Psi = \arctg\left(\frac{b}{a}\right) + (B - AB + 2) \cdot 90^\circ \\ a \neq 0, b = 0 \Rightarrow \Psi = (A + 1) \cdot 90^\circ \\ a = 0, b \neq 0 \Rightarrow \Psi = (B + 2) \cdot 90^\circ \end{cases} \quad (3.2-3)$$

$$\text{onde: } \begin{cases} A = \frac{a}{|a|} \\ B = \frac{b}{|b|} \end{cases}$$

3.2.1 Determinação da Direção do Fluxo

No caso da modelagem TIN, o gradiente é calculado a partir das coordenadas dos três vértices, segundo a equação (3.1-3). De posse do ângulo ψ , é possível determinar a passagem do fluxo pela superfície triangular, isto é, de onde vem a água e para onde vai. Para isso, deve-se colocar linhas de fluxo paralelas ao gradiente, utilizando o ângulo ψ , como pode ser visto na Figura 3.2-1.

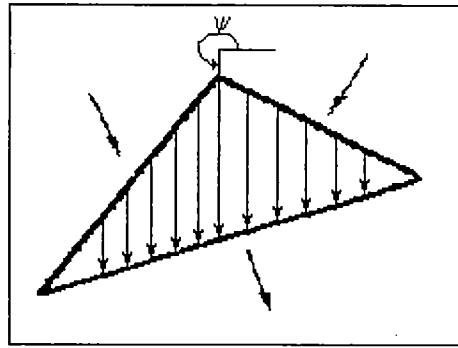


Figura 3.2-1: Passagem do fluxo pela superfície triangular

Em alguns casos, o triângulo pode fornecer água para dois outros triângulos (Figura 3.2-2). Neste caso, é possível determinar a quantidade que vai para cada um deles.

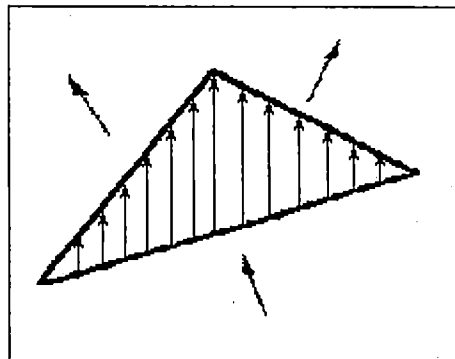


Figura 3.2-2: Fluxo saindo por dois lados do triângulo

Para determinar a divisão da quantidade de água entre os triângulos adjacentes, utiliza-se a linha de fluxo paralela ao gradiente que passa pelo vértice do triângulo, dividindo-o em dois outros triângulos. A fração de água que vai para cada um dos lados é calculada dividindo-se a área de cada sub-triângulo pela área do triângulo original, como mostrado na Figura 3.2-3.

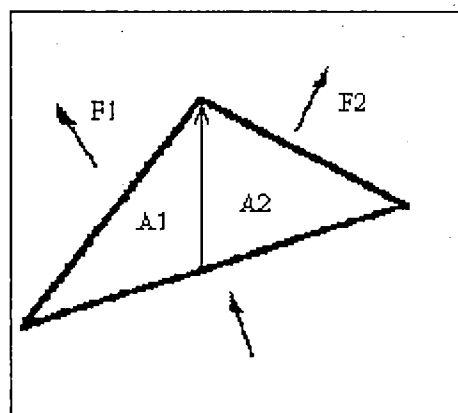


Figura 3.2-3: Divisão do Fluxo de Saída do Triângulo

3.2.2 Quantificação da Divisão do Fluxo

O cálculo da área do triângulo a partir de seus vértices é dado pela equação (3.1-4). Deste modo, para calcular a quantidade de fluxo que passa em cada lado do triângulo, é necessário encontrar o ponto p , onde a linha de fluxo que passa pelo vértice k corta o lado do triângulo oposto a esse vértice Figura 3.2-4.

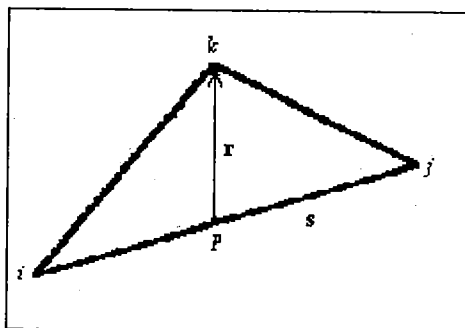


Figura 3.2-4: Ponto de Divisão do Triângulo

A equação da reta correspondente à linha de fluxo paralela ao gradiente que passa pelo vértice k é:

$$r: y - y_k = \operatorname{tg}(\alpha)(x - x_k) \quad (3.2-4)$$

A equação da reta que liga os vértices i e j é:

$$s: y - y_i = \frac{y_j - y_i}{x_j - x_i}(x - x_i) \quad (3.2-5)$$

O ponto p é determinado pela intersecção das retas r e s , e pode ser calculado solucionando o sistema composto pelas equações (3.2-1) e (3.2-2). A solução desse sistema é:

$$\begin{cases} x_p = \frac{(y_k - x_k \cdot \operatorname{tg}(\alpha) - y_i)\Delta x + x_i \Delta y}{\Delta y - \Delta x \cdot \operatorname{tg}(\alpha)} \\ y_p = y_k + (x_p - x_k) \cdot \operatorname{tg}(\alpha) \end{cases} \quad (3.2-6)$$

$$\text{onde: } \begin{cases} \Delta x = x_j - x_i \\ \Delta y = y_j - y_i \end{cases}$$

Utilizando-se a fórmula para cálculo de área do triângulo dada pela equação (3.1-4), pode-se calcular a fração do fluxo que sai pelo lado ik dividindo-se a área do triângulo ipk pela área do triângulo ijk :

$$P_{ik} = \frac{(x_i y_p - x_p y_i) + (x_k y_i - x_i y_k) + (x_p y_k - x_k y_p)}{(x_i y_j - x_j y_i) + (x_k y_i - x_i y_k) + (x_j y_k - x_k y_j)} \quad (3.2-7)$$

A fração de fluxo restante sai através do lado jk :

$$P_{jk} = 1 - P_{ik} \quad (3.2-8)$$

3.2.3 Identificação da Direção do Fluxo

O comportamento do fluxo depende essencialmente da direção do gradiente (ψ) e dos ângulos formados pelos lados do triângulo com o eixo x , denominados ϕ_1 , ϕ_2 e ϕ_3 , correspondendo, respectivamente, aos lados jk , ji e ki (Figura 3.2-5).

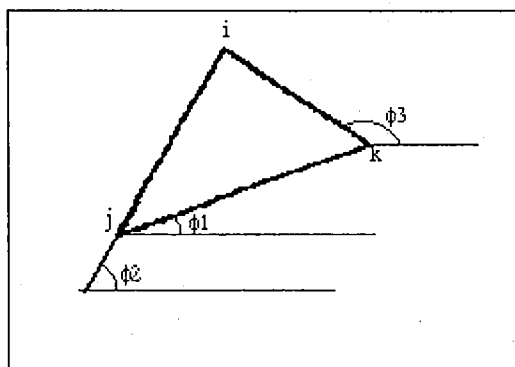
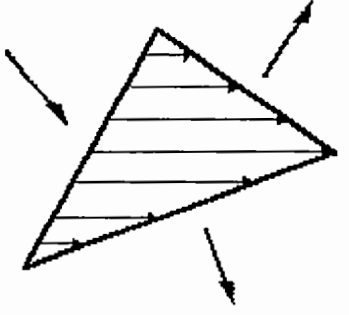
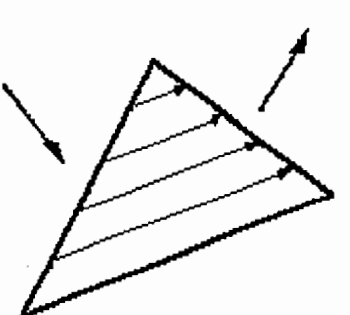
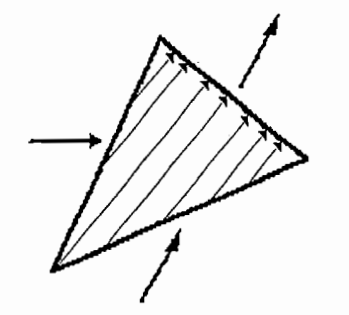
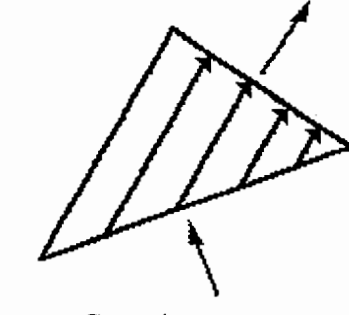
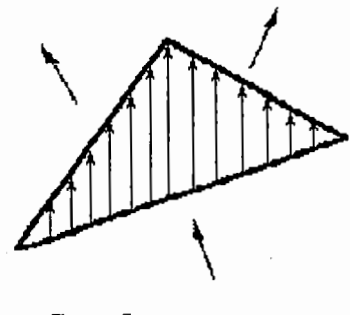
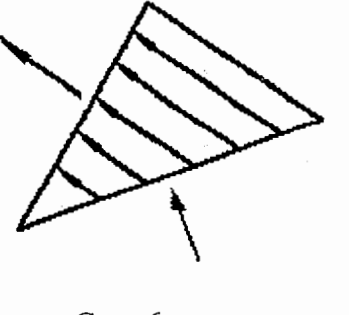
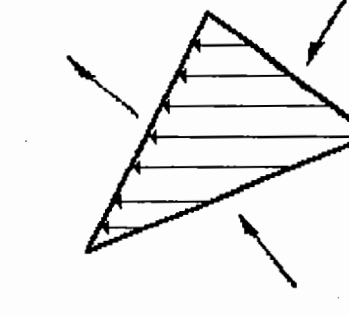
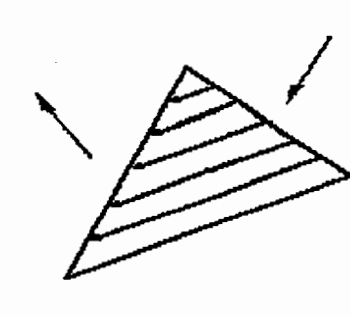
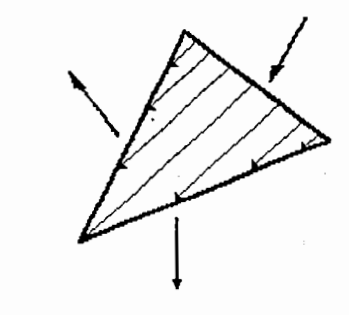
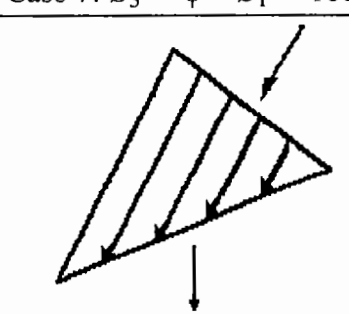
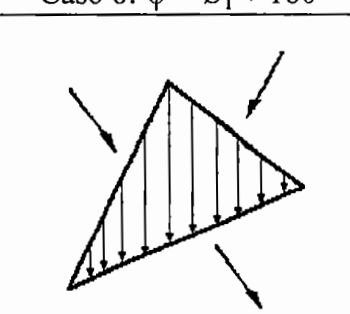
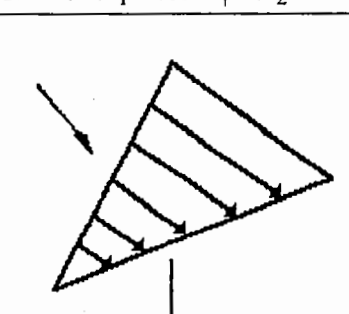


Figura 3.2-5: Ângulos formados pelos lados do triângulo com o eixo x

A entrada e saída do fluxo de água através dos lados do triângulo pode ocorrer de doze modos distintos, que são apresentados na Tabela 3.2-1.

Tabela 3.2-1: Entrada e saída do fluxo de água através dos lados do triângulo

 <p>Caso 1: $\varnothing_3 + 180 < \psi < \varnothing_1$</p>	 <p>Caso 2: $\psi = \varnothing_1$</p>	 <p>Caso 3: $\varnothing_1 < \psi < \varnothing_2$</p>
 <p>Caso 4: $\psi = \varnothing_2$</p>	 <p>Caso 5: $\varnothing_2 < \psi < \varnothing_3$</p>	 <p>Caso 6: $\psi = \varnothing_3$</p>
 <p>Caso 7: $\varnothing_3 < \psi < \varnothing_1 + 180$</p>	 <p>Caso 8: $\psi = \varnothing_1 + 180$</p>	 <p>Caso 9: $\varnothing_1 + 180 < \psi < \varnothing_2 + 180$</p>
 <p>Caso 10: $\psi = \varnothing_2 + 180$</p>	 <p>Caso 11: $\varnothing_2 + 180 < \psi < \varnothing_3 + 180$</p>	 <p>Caso 12: $\psi = \varnothing_3 + 180$</p>

3.2.4 Intersecção de Fluxos

Em alguns casos, um triângulo A envia um fluxo F_A para um triângulo B. Este, por sua vez, envia um fluxo F_B para o triângulo A, como indicado na Figura 3.2-6.

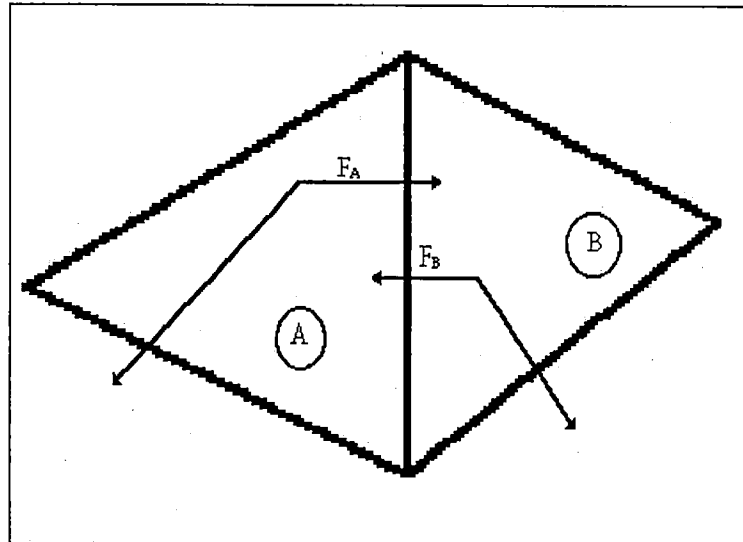


Figura 3.2-6: Troca de Fluxo entre Triângulos

Neste caso, identifica-se a formação de um condutor de fluxo (canal) ao longo da intersecção entre esses dois triângulos (Figura 3.2-7).

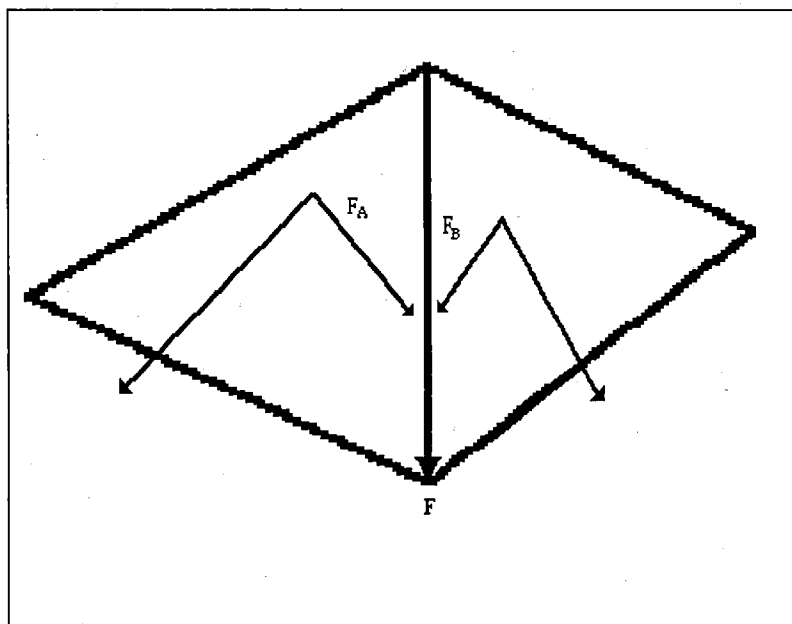


Figura 3.2-7: Identificação de Canal na Intersecção dos Triângulos

O lado que forma o condutor de fluxo liga dois vértices. O sentido do fluxo será do vértice mais elevado em direção ao vértice de menor elevação. Esse vértice receberá um fluxo $F = F_A + F_B$.

3.3 Métodos Numéricos Aplicados ao Escoamento Superficial

O modelo da onda cinemática utiliza a equação da continuidade (2.1-1) e a relação entre a vazão e a altura do nível d'água (2.1-5) :

$$\frac{\delta A}{\delta t} + \frac{\delta Q}{\delta x} = i - f$$
$$Q = \alpha h^m$$

Devido à natureza das equações, raros são os casos em que é possível encontrar solução analítica para simulações do escoamento. Deste modo, deve-se aplicar métodos numéricos para encontrar aproximações para as soluções desejadas.

Nesta seção, as equações serão desenvolvidas para trabalhar com diferentes soluções numéricas para o problema acima, uma formulação utilizando o método das diferenças finitas e três formulações para a aplicação do método dos elementos finitos.

3.3.1 Método das Diferenças Finitas

Tayfur et al. (1993) mostrou como o método das diferenças finitas pode ser utilizado na solução da equação da continuidade para o caso unidimensional. Para a aplicação deste método, a região é discretizada atribuindo valores pontuais às variáveis. Em seguida, as derivadas parciais na equação da continuidade são substituídas pelas suas aproximações, utilizando os valores pontuais.

As aproximações das derivadas parciais podem ser vistas na equação (3.3-1).

$$\begin{aligned}\frac{\partial Q}{\partial x} &= \frac{Q_i - Q_{i-1}}{\Delta x} \\ \frac{\partial h}{\partial t} &= \frac{h^{k+1} - h^k}{\Delta t}\end{aligned}\tag{3.3-1}$$

onde:

- Q_i = Vazão no ponto i ao longo de x
- h^k = Altura no tempo k

Após a substituição, a equação da continuidade torna-se:

$$\frac{Q_i^k - Q_{i-1}^k}{\Delta x} + \frac{h_i^{k+1} - h_i^k}{\Delta t} = q\tag{3.3-2}$$

Isolando-se o termo h no tempo $k+1$, obtém-se:

$$h_i^{k+1} = h_i^k + \left[q - \frac{(Q_i^k - Q_{i-1}^k)}{\Delta x} \right] \Delta t\tag{3.3-3}$$

O cálculo é feito iterativamente da seguinte forma:

- Atribuir um valor inicial para h em cada ponto no tempo inicial
- Repetir para cada intervalo de tempo até atingir o tempo final
 - calcula-se o valor de Q em cada ponto utilizando a relação entre Q e h
 - calcula-se os novos valores para h utilizando a equação (3.3-3).

3.3.2 Método dos Elementos Finitos

O método dos elementos finitos consiste em discretizar a região em diversos elementos. Em seguida, deve-se especificar uma função de interpolação, com uma equação para cada elemento. A variação espacial das quantidades deve ser aproximada por uma função discreta, para ser aplicada a cada elemento. Finalmente, as equações são agrupadas em um sistema, cuja resolução produz a solução do problema físico. O desenvolvimento teórico do método dos elementos finitos encontra-se na seção 2.4.2.

3.3.2.1 Substituição de Q em função de h na equação de continuidade:

Uma maneira de eliminar a função de vazão da equação é substituir $Q(x,t)$ por uma função em $h(x,t)$ [Goodrich et. al (1993)], utilizando a relação entre a vazão (Q) e a altura do nível de água (h) dada pela equação (2.1-5).

Após a substituição, a equação da continuidade é escrita da seguinte forma:

$$\alpha \frac{\partial h^n}{\partial x} + \frac{\partial h}{\partial t} = q$$

$$\alpha n h^{n-1} \frac{\partial h}{\partial x} + \frac{\partial h}{\partial t} = q \quad (3.3-4)$$

$$nV \frac{\partial h}{\partial x} + \frac{\partial h}{\partial t} = q$$

onde:

$$V = \alpha h^{n-1} \quad (3.3-5)$$

Em seguida, aproxima-se h por uma função discreta \hat{h} :

$$\hat{h} = \sum h_i N_i \quad (3.3-6)$$

A substituição de h pela função discreta acima gera um resíduo, dado por:

$$R = nV \dot{\mathbf{N}}_x^T \mathbf{h} + \mathbf{N}^T \dot{\mathbf{h}} - q \quad (3.3-7)$$

Para se obter a melhor solução, deve-se minimizar o resíduo na região. O método de Galerkin utiliza as funções de aproximação $N(x)$ como fator de ponderação [BREBBIA & FERRANTE (1975), WAIT & MITCHELL (1985)].

Desta forma, a equação a ser resolvida é:

$$\int \mathbf{N} R dx = nV \int \mathbf{N} \dot{\mathbf{N}}_x^T dx \mathbf{h} + \int \mathbf{N} \mathbf{N}^T dx \dot{\mathbf{h}} - q \int \mathbf{N} dx = 0 \quad (3.3-8)$$

Assim, obtém-se a seguinte equação:

$$\mathbf{A} \dot{\mathbf{h}} + \mathbf{B} \mathbf{h} + \mathbf{C} = 0 \quad (3.3-9)$$

onde:

$$\begin{aligned} \mathbf{A} &= \int \mathbf{N} \mathbf{N}^T dx = \int_0^L \begin{bmatrix} Ni^2 & NiNj \\ NiNj & Nj^2 \end{bmatrix} dx = \frac{L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \\ \mathbf{B} &= nV \int \mathbf{N} \dot{\mathbf{N}}_x^T dx = nV \int_0^L \frac{1}{L} \begin{bmatrix} -Ni & Ni \\ -Nj & Nj \end{bmatrix} dx = nV \frac{1}{L} \begin{bmatrix} -\frac{L}{2} & \frac{L}{2} \\ -\frac{L}{2} & \frac{L}{2} \end{bmatrix} = \frac{nV}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \\ \mathbf{C} &= -q \int \mathbf{N} dx = -q \int_0^L \begin{bmatrix} Ni \\ Nj \end{bmatrix} dx = -\frac{qL}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned}$$

(3.3-10)

A equação pode ser resolvida utilizando-se o método das diferenças finitas no tempo, substituindo h por valores discretos pontuais:

$$\mathbf{A} \frac{\mathbf{h}^{k+1} - \mathbf{h}^k}{\Delta t} + \mathbf{B}\mathbf{h}^k + \mathbf{C} = \mathbf{0} \quad (3.3-11)$$

$$\mathbf{h}^{k+1} = \mathbf{h}^k - \mathbf{A}^{-1}(\mathbf{B}\mathbf{h}^k + \mathbf{C})$$

Para calcular a matriz \mathbf{B} na equação (3.3-10), é necessário calcular V . Neste caso, utiliza-se, para cada elemento, o valor médio no elemento. Para o cálculo de V , utiliza-se a equação (3.3-5). Dois métodos foram testados nesse cálculo, sem que houvesse diferença significativa entre eles:

$$1) V = \alpha \frac{(h_i^{n-1} + h_j^{n-1})}{2} \quad (3.3-12)$$

$$2) V = \alpha \frac{1}{L} \int h^{n-1} dh \quad (3.3-13)$$

3.3.2.2 Aproximações Lineares para V e h .

Na seção anterior, o sistema de equações foi simplificado substituindo-se a vazão Q pela sua relação com a altura do nível d'água h na equação da continuidade.

$$\boxed{Q = ah^n \quad \text{ou} \quad \begin{cases} Q = Vh \\ V = ah^{n-1} \end{cases}} \quad (3.3-14)$$

Na solução proposta anteriormente, V foi considerado constante em cada elemento. Com o objetivo de melhorar os resultados obtidos, será desenvolvida nesta seção uma formulação utilizada em Taylor et al. (1974) e Vieux et al. (1990), que aproxima V por uma função básica linear discretizada, da mesma forma como foi feito em relação à altura do nível d'água h .

Como visto na seção 3.3.2.1, após a substituição de Q por h , chegou-se à equação (3.3-4):

$$nV \frac{\partial h}{\partial x} + \frac{\partial h}{\partial t} = q$$

Neste ponto, V e h são aproximados por uma função linear:

$$\begin{cases} \hat{h} = \sum h_i N_i \\ \hat{V} = \sum V_i N_i \end{cases} \quad (3.3-15)$$

Deste modo, o produto $V \frac{\partial h}{\partial x}$ pode ser escrito na forma matricial da seguinte forma:

$$V \frac{\partial h}{\partial x} = \mathbf{N}^T \mathbf{V} \mathbf{N}_x^T \mathbf{h} \quad (3.3-16)$$

O resíduo decorrente da substituição de V e h pelas funções básicas lineares \dot{V} e \dot{h} na equação (3.3-4) é:

$$R = n\mathbf{N}^T \mathbf{V} \mathbf{N}_x^T \dot{h} + \mathbf{N}^T \dot{h} - q \quad (3.3-17)$$

Após a aplicação do método de Galerkin para minimizar o resíduo, chegamos à equação diferencial da forma $\mathbf{A} \dot{\mathbf{h}} + \mathbf{B} \mathbf{h} + \mathbf{C} = \mathbf{0}$, onde \mathbf{A} e \mathbf{C} possuem os mesmos valores do caso anterior, enquanto a matriz \mathbf{B} é calculada da seguinte forma:

$$\begin{aligned} \mathbf{B} &= n \int \mathbf{N} \mathbf{N}^T \mathbf{V} \mathbf{N}_x^T dx = n \int_0^L \begin{bmatrix} Ni^2 & NiNj \\ NiNj & Nj^2 \end{bmatrix} \cdot \begin{bmatrix} -\frac{V_i}{L} & \frac{V_i}{L} \\ -\frac{V_j}{L} & \frac{V_j}{L} \end{bmatrix} dx = n \frac{L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \frac{1}{L} \begin{bmatrix} -V_i & V_i \\ -V_j & V_j \end{bmatrix} \\ &= \frac{n}{6} \begin{bmatrix} -(2V_i + V_j) & (2V_i + V_j) \\ -(V_i + 2V_j) & (V_i + 2V_j) \end{bmatrix} = \frac{n}{6} \left(\begin{bmatrix} -2 & 2 \\ -1 & 1 \end{bmatrix} V_i + \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix} V_j \right) \end{aligned} \quad (3.3-18)$$

O cálculo no regime temporal pode ser feito utilizando-se a equação (3.3-11), da mesma forma como se fez na seção anterior.

No entanto, pode-se melhorar a solução utilizando-se um método alternativo para o cálculo no tempo. Neste método, o termo \mathbf{h} , que era utilizado somente no tempo anterior, é substituído pelo valor médio entre o tempo anterior e o tempo atual [Goodrich et. al (1993), Vieux et al. (1990)]:

$$h = \frac{h^{k+1} + h^k}{2} \quad (3.3-19)$$

Deste modo, a equação diferencial é desenvolvida da seguinte forma:

$$\mathbf{A} \dot{\mathbf{h}} + \mathbf{B}\mathbf{h} + \mathbf{C} = \mathbf{0}$$

$$\mathbf{A} \frac{h^{k+1} - h^k}{\Delta t} + \mathbf{B} \frac{h^{k+1} + h^k}{2} + \mathbf{C} = \mathbf{0}$$

$$\mathbf{A}(h^{k+1} - h^k) + \frac{\Delta t}{2} \mathbf{B}(h^{k+1} + h^k) + \Delta t \mathbf{C} = \mathbf{0}$$

$$\left(\mathbf{A} + \frac{\Delta t}{2} \mathbf{B} \right) h^{k+1} - \left(\mathbf{A} - \frac{\Delta t}{2} \mathbf{B} \right) h^k + \Delta t \mathbf{C} = \mathbf{0}$$

$$\Rightarrow h^{k+1} = \mathbf{D}^{-1} \mathbf{E} h^k + \Delta t \mathbf{D}^{-1} \mathbf{C}$$

(3.3-20)

$$\text{onde: } \begin{cases} \mathbf{D} = \mathbf{A} + \frac{\Delta t}{2} \mathbf{B} \\ \mathbf{E} = \mathbf{A} - \frac{\Delta t}{2} \mathbf{B} \end{cases}$$

3.3.2.3 Aproximação linear para Q e h

Foram propostas duas soluções utilizando o método dos elementos finitos para solucionar o problema do escoamento superficial. Nos dois casos, a função de vazão Q foi substituída, na equação da continuidade, pela sua relação com a altura do nível d'água h .

Nesta seção, será proposta uma nova formulação, onde Q não é substituído por h na equação da continuidade. Neste caso, será utilizada uma aproximação linear discreta para Q [JAYAWARDENA & WHITE (1977), Ross et al. (1979)], da mesma forma como já é feito para h :

$$\begin{cases} \hat{h} = \sum h_i N_i \\ \hat{Q} = \sum Q_i N_i \end{cases}$$

Neste caso, a solução da derivada parcial $\frac{\partial Q}{\partial x}$ é semelhante à solução de $\frac{\partial h}{\partial x}$, e a substituição das funções discretizadas na equação da continuidade gera o seguinte resíduo, escrito em forma matricial:

$$R = \dot{\mathbf{N}}_x^T \mathbf{Q} + \mathbf{N}^T \dot{\mathbf{h}} - q \quad (3.3-21)$$

Após a utilização do método de Galerkin, obtém-se a seguinte equação:

$$\int \mathbf{N} R dx = \int \mathbf{N} \dot{\mathbf{N}}_x^T dx \mathbf{Q} + \int \mathbf{N} \mathbf{N}^T dx \dot{\mathbf{h}} - q \int \mathbf{N} dx = 0 \quad (3.3-22)$$

A equação acima pode ser escrita na forma:

$$\mathbf{A} \dot{\mathbf{h}} + \mathbf{B} \mathbf{Q} + \mathbf{C} = 0 \quad (3.3-23)$$

onde:

$$\begin{aligned} \mathbf{A} &= \int \mathbf{N} \mathbf{N}^T dx = \int_0^L \begin{bmatrix} Ni^2 & NiNj \\ NiNj & Nj^2 \end{bmatrix} dx = \frac{L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \\ \mathbf{B} &= \int \mathbf{N} \dot{\mathbf{N}}_x^T dx = \int_0^L \frac{1}{L} \begin{bmatrix} -Ni & Ni \\ -Nj & Nj \end{bmatrix} dx = \frac{1}{L} \begin{bmatrix} -\frac{L}{2} & \frac{L}{2} \\ -\frac{L}{2} & \frac{L}{2} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \\ \mathbf{C} &= -q \int \mathbf{N} dx = -q \int_0^L \begin{bmatrix} Ni \\ Nj \end{bmatrix} dx = -\frac{qL}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned} \quad (3.3-24)$$

Para resolver a equação no tempo, utiliza-se o método das Diferenças Finitas da seguinte forma:

- Calcular vazão inicial, a partir da solução inicial
- Repetir até atingir o tempo final desejado
 - calcular o novo valor para h
 - a partir do novo valor de h , recalculando Q

O cálculo da vazão Q a cada passo é feito utilizando-se a relação entre Q e h (2.1-5):

$$Q = \alpha \cdot h^n$$

3.3.3 Equação da Continuidade Bidimensional

A equação da onda cinemática bidimensional é dada pela equação (2.1-1) :

$$\frac{\delta h}{\delta t} + \frac{\delta Q_x}{\delta x} + \frac{\delta Q_y}{\delta y} = i - f$$

A relação entre a altura do nível d'água e as projeções da vazão nas direções x e y é dada pela equação (2.1-5):

$$Q_x = \alpha_x h^m$$

$$Q_y = \alpha_y h^m$$

A resolução do escoamento superficial para o caso bidimensional utilizará o método dos elementos finitos com aproximações lineares para Q e h :

$$\begin{cases} \hat{h} = \sum h_i N_i \\ \hat{Q}_x = \sum Q_{x_i} N_i \\ \hat{Q}_y = \sum Q_{y_i} N_i \end{cases} \quad (3.3-25)$$

Os valores N_i, N_j, N_k são dados em (3.1-6) e o cálculo das derivadas parciais em (3.1-7) e (3.1-8). A substituição das funções discretizadas na equação da continuidade gera o seguinte resíduo, escrito em forma matricial:

$$R = \dot{\mathbf{N}}_x^T \mathbf{Q}_x + \dot{\mathbf{N}}_y^T \mathbf{Q}_y + \mathbf{N}^T \dot{\mathbf{h}} - q \quad (3.3-26)$$

Após a utilização do método de Galerkin, obtém-se a seguinte equação:

$$\int_{\Omega} \mathbf{N} \mathbf{R} d\Omega = \int_{\Omega} \mathbf{N} \dot{\mathbf{N}}_x^T d\Omega \mathbf{Q}_x + \int_{\Omega} \mathbf{N} \dot{\mathbf{N}}_y^T d\Omega \mathbf{Q}_y + \int_{\Omega} \mathbf{N} \mathbf{N}^T d\Omega \dot{\mathbf{h}} - q \int_{\Omega} \mathbf{N} d\Omega = 0 \quad (3.3-27)$$

A equação acima pode ser escrita na forma:

$$\mathbf{A} \dot{\mathbf{h}} + \mathbf{B}_1 \mathbf{Q}_x + \mathbf{B}_2 \mathbf{Q}_y + \mathbf{C} = 0 \quad (3.3-28)$$

A seguir serão apresentados os cálculos dos coeficientes \mathbf{A} , \mathbf{B}_1 , \mathbf{B}_2 e \mathbf{C} .

- As integrais são calculadas sobre a região Ω , que, neste caso, é um triângulo:

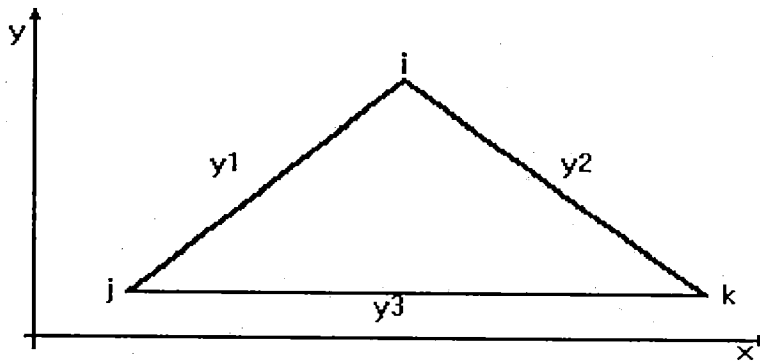


Figura 3.3-1: Região Triangular

- Seja G o valor dentro da integral. A integral na região pode ser desmembrada da seguinte forma:

$$\int_{\Omega} G d\Omega = \int_j^i \int_0^{y1} G dy dx + \int_i^k \int_0^{y2} G dy dx - \int_j^k \int_0^{y3} G dy dx \quad (3.3-29)$$

- A matriz \mathbf{A} da equação (3.3-28) é, para cada elemento:

$$\mathbf{A} = \int_{\Omega} \mathbf{N} \mathbf{N}^T d\Omega = \begin{bmatrix} \int N_i^2 & \int N_i N_j & \int N_i N_k \\ \int N_j N_i & \int N_j^2 & \int N_j N_k \\ \int N_k N_i & \int N_k N_j & \int N_k^2 \end{bmatrix} \quad (3.3-30)$$

O produto $N_i N_j$ é dado por:

$$N_i N_j = \frac{1}{4A^2} (a_i a_j + b_i b_j x^2 + c_i c_j y^2 + (a_i b_j + a_j b_i)x + (a_i c_j + a_j c_i)y + (b_i c_j + b_j c_i)xy) \quad (3.3-31)$$

A integral do produto $N_i N_j$ em função de y é:

$$\int_0^y N_i N_j dy = \frac{1}{4A^2} \left[\begin{aligned} & [a_i a_j + b_i b_j x^2 + (a_i b_j + a_j b_i)x]y + \\ & [a_i c_j + a_j c_i + (b_i c_j + b_j c_i)x] \frac{y^2}{2} + c_i c_j \frac{y^3}{3} \end{aligned} \right] \quad (3.3-32)$$

Seja a reta y ligando os pontos (x_0, y_0) e (x_f, y_f) . A equação da reta pode ser escrita da seguinte forma:

$$y = mx + t \quad (3.3-33)$$

onde:

$$\begin{cases} m = \frac{\Delta y}{\Delta x} = \frac{y_f - y_0}{x_f - x_0} \\ t = y_0 - mx_0 \end{cases}$$

Substituindo (3.3-33) na equação (3.3-32) obtém-se:

$$\int_0^{mx+t} N_i N_j dy = \frac{1}{4A^2} \left[a_i a_j mx + a_i a_j t + b_i b_j mx^3 + b_i b_j tx^2 + (a_i b_j + a_j b_i) mx^2 + \right. \\ \left. (a_i b_j + a_j b_i) tx + (a_i c_j + a_j c_i) \frac{m^2}{2} x^2 + (a_i c_j + a_j c_i) mtx + \right. \\ \left. (a_i c_j + a_j c_i) \frac{t^2}{2} + (b_i c_j + b_j c_i) \frac{m^2}{2} x^3 + (b_i c_j + b_j c_i) mtx^2 + \right. \\ \left. (b_i c_j + b_j c_i) \frac{t^2}{2} x + c_i c_j \frac{m^3}{3} x^3 + c_i c_j m^2 tx^2 + c_i c_j mt^2 x + \frac{c_i c_j}{3} t^3 \right] \quad (3.3-34)$$

Deste modo, a integral na região torna-se:

$$\int_{x_0}^{x_f} \int_0^{mx+t} N_i N_j dy dx = \frac{1}{4A^2} \left[\left[b_i b_j m + c_i c_j \frac{m^3}{3} + (b_i c_j + b_j c_i) \frac{m^2}{2} \right] \frac{x_f^4 - x_0^4}{4} + \right. \\ \left[b_i b_j t + (a_i b_j + a_j b_i) m + (a_i c_j + a_j c_i) \frac{m^2}{2} + (b_i c_j + b_j c_i) m t + c_i c_j m^2 t \right] \frac{x_f^3 - x_0^3}{3} + \\ \left[a_i a_j m + (a_i b_j + a_j b_i) t + (a_i c_j + a_j c_i) m t + (b_i c_j + b_j c_i) \frac{t^2}{2} + c_i c_j m t^2 \right] \frac{x_f^2 - x_0^2}{2} + \\ \left. \left[a_i a_j t + (a_i c_j + a_j c_i) \frac{t^2}{2} + c_i c_j \frac{t^3}{3} \right] (x_f - x_0) \right] \quad (3.3-35)$$

- A matriz \mathbf{B}_1 na equação (3.3-28) é dada por:

$$\mathbf{B}_1 = \int_{\Omega} \mathbf{N} \dot{\mathbf{N}}_x^T d\Omega = \begin{bmatrix} \int N_i \dot{N}_{x_i} & \int N_i \dot{N}_{x_j} & \int N_i \dot{N}_{x_k} \\ \int N_j \dot{N}_{x_i} & \int N_j \dot{N}_{x_j} & \int N_j \dot{N}_{x_k} \\ \int N_k \dot{N}_{x_i} & \int N_k \dot{N}_{x_j} & \int N_k \dot{N}_{x_k} \end{bmatrix} \quad (3.3-36)$$

A derivada parcial $\frac{\delta N_i}{\delta x}$ é calculada em (3.1-7). Assim, obtém-se o seguinte resultado:

$$\int_{\Omega} N_i \dot{N}_{x_j} d\Omega = \int_{x_0}^{x_f} \int_0^y N_i b_j dy dx \quad (3.3-37)$$

Calculando a integral interna, obtém-se:

$$\int_0^y N_i b_j dy = \int_0^y \frac{1}{2A} (a_i + b_i x + c_i y) b_j dy = \frac{1}{2A} \left(a_i b_j y + b_i b_j xy + c_i b_j \frac{y^2}{2} \right) \quad (3.3-38)$$

Substituindo a equação da reta (3.3-33) na equação (3.3-38), obtém-se:

$$\int_0^{mx+t} N_i b_j dy = \frac{1}{2A} \left[\left(b_i b_j m + c_i b_j \frac{m^2}{2} \right) x^2 + (a_i b_j m + b_i b_j t + c_i b_j m t) x + \left(a_i b_j t + c_i b_j \frac{t^2}{2} \right) \right] \quad (3.3-39)$$

O cálculo da integral em toda a região é feito substituindo-se a equação (3.3-39) em (3.3-37):

$$\int_{x_0}^{x_f} \int_0^{mx+t} N_i b_j dy dx = \frac{1}{2A} \left[\left(b_i b_j m + c_i b_j \frac{m^2}{2} \right) \frac{(x_f^3 - x_0^3)}{3} + (a_i b_j m + b_i b_j t + c_i b_j m t) \frac{(x_f^2 - x_0^2)}{2} + \left(a_i b_j t + c_i b_j \frac{t^2}{2} \right) (x_f - x_0) \right] \quad (3.3-40)$$

- A matriz \mathbf{B}_2 na equação (3.3-28) é dada por:

$$\mathbf{B}_2 = \int_{\Omega} \mathbf{N} \dot{\mathbf{N}}_y^T d\Omega = \begin{bmatrix} \int N_i \dot{N}_{y_i} & \int N_i \dot{N}_{y_j} & \int N_i \dot{N}_{y_k} \\ \int N_j \dot{N}_{y_i} & \int N_j \dot{N}_{y_j} & \int N_j \dot{N}_{y_k} \\ \int N_k \dot{N}_{y_i} & \int N_k \dot{N}_{y_j} & \int N_k \dot{N}_{y_k} \end{bmatrix}$$

A derivada parcial $\frac{\delta N_i}{\delta y}$ é calculada em (3.1-8). Assim, obtém-se o seguinte resultado:

$$\int_{\Omega} N_i \dot{N}_{y_j} d\Omega = \int_{x_0}^{x_f} \int_0^y N_i c_j dy dx$$

O desenvolvimento dessa equação é feito de forma semelhante à utilizada na matriz \mathbf{B}_1 , obtendo-se o seguinte resultado:

$$\int_{x_0}^{x_f} \int_0^{mx+t} N_i c_j dy = \frac{1}{2A} \left[\begin{aligned} & \left(b_i c_j m + c_i c_j \frac{m^2}{2} \right) \frac{(x_f^3 - x_0^3)}{3} + \left(a_i c_j m + b_i c_j t + c_i c_j m t \right) \frac{(x_f^2 - x_0^2)}{2} \\ & + \left(a_i c_j t + c_i c_j \frac{t^2}{2} \right) (x_f - x_0) \end{aligned} \right]$$

(3.3-41)

- A matriz C na equação (3.3-28) é dada por:

$$C = \int_{\Omega} \mathbf{N} d\Omega = \begin{bmatrix} \int N_i \\ \int N_j \\ \int N_k \end{bmatrix}$$

O cálculo dos elementos em C é apresentado a seguir:

$$\int_{\Omega} N_i d\Omega = \int_{x_0}^{x_f} \int_0^y N_i dy dx$$

$$\int_0^y N_i dy = \frac{1}{2A} \int_0^y (a_i + b_i x + c_i y) dy = \frac{1}{2A} \left(a_i y + b_i xy + c_i \frac{y^2}{2} \right)$$

Substituindo-se y pela equação da reta (3.3-33):

$$\int_0^{mx+t} N_i dy = \frac{1}{2A} \left(a_i mx + a_i t + b_i mx^2 + b_i tx + \frac{c_i}{2} m^2 x^2 + c_i m tx + \frac{c_i}{2} t^2 \right)$$

A integral na região, deste modo, é:

$$\int_{x_0}^{x_f} \int_0^{mx+t} N_i dy = \frac{1}{2A} \left[\left(b_i m + \frac{c_i}{2} m^2 \right) \frac{x_f^3 - x_0^3}{3} + (a_i m + b_i t + c_i mt) \frac{x_f^2 - x_0^2}{2} + \left(a_i t + \frac{c_i}{2} t^2 \right) (x_f - x_0) \right]$$

(3.3-42)

4. Solução Computacional

4.1 Objetivos do Estudo Computacional

O estudo computacional visa criar condições para que se possa simular numericamente o comportamento do escoamento superficial em bacias hidrográficas, principalmente nos casos onde não é possível encontrar a solução analítica. Para validar um determinado método, são utilizados casos excepcionais, onde a solução analítica é conhecida. Deste modo, foi possível comparar diversos métodos, utilizando o escoamento unidimensional, para encontrar o mais indicado ao escoamento bidimensional.

Para que seja possível entender de maneira clara os caminhos trilhados até a implementação final do escoamento bidimensional, este capítulo divide-se em duas partes. Na primeira parte, são analisadas diferentes formulações para o caso unidimensional, utilizando um problema que possui solução analítica. Isto é feito com o objetivo de comparar a solução computacional de cada uma das formulações, de modo que se possa obter a que apresenta valores mais próximos à solução analítica. Na segunda parte, utiliza-se a melhor formulação encontrada para o caso unidimensional para aplicá-la ao caso bidimensional. Neste caso, serão analisados diversos casos para verificar a validade dessa metodologia. No caso bidimensional, há outros aspectos envolvidos, além dos métodos numéricos. De acordo com o método proposto, os triângulos advindos da discretização devem ser processados individualmente. Para isso, há a necessidade de uma definição clara da ordem do processamento e a forma de comunicação entre eles.

4.2 Escoamento Superficial Unidimensional

Nesta seção, serão analisados os diversos métodos numéricos para o resolução do problema do escoamento unidimensional utilizando o modelo da onda cinemática. Para este fim, será utilizada uma região planar retangular, com as seguintes características:

- Comprimento: $L = 100$ m
- Coeficientes da equação da onda cinemática:
 - $\alpha = 0,6 \text{ mm}^{0,5}/\text{s}$
 - $m=1,5$
- Intensidade da chuva: $q = 50 \text{ mm/h}$

Os resultados obtidos serão apresentados graficamente (destacando a vazão por unidade de largura na saída do canal) e comparados à solução analítica, calculada utilizando os resultados apresentados por STREETER (1966).

A vazão de saída do canal é proporcional à intensidade da chuva e ao tempo, até atingir o ponto de saturação ($t = t_s$), quando a vazão será então constante e proporcional ao comprimento do canal. Esse valor é calculado pela fórmula a seguir:

$$Q_{x=L} = \begin{cases} \alpha(q \cdot t)^m & 0 < t < t_s \\ q \cdot L & t \geq t_s \end{cases} \quad (4.2.1-1)$$

O tempo necessário para que o ponto de saturação seja atingido é dado pela fórmula a seguir:

$$t_s = \left(\frac{L}{\alpha \cdot q^{m-1}} \right)^{1/m} \quad (4.2.1-2)$$

Em alguns dos casos que serão estudados, será utilizado um cálculo de erro, isto é, diferença entre a solução numérica e a solução analítica. Esse erro será calculado totalizando-se a diferença entre a solução numérica e a solução analítica em cada intervalo de tempo.

$$\xi = \sum_{t=0}^{T_{max}} |Q_t^N - Q_t^A| \quad (4.2.1-3)$$

4.2.1 Método das Diferenças Finitas

Para resolver numericamente o problema do escoamento unidimensional pelo modelo de onda cinemática, será utilizada a formulação apresentada na seção 3.3.1.

Foram realizados diversos testes, modificando, em cada um deles, o espaçamento entre os pontos. Os resultados são apresentados na Tabela 4.2-1, onde N é o número de pontos na discretização:

Tabela 4.2-1: Erro da solução numérica utilizando Diferenças Finitas

Δx	N	Erro Total
100	2	149,53
25	5	46,07
10	11	19,20
5	21	9,78
1	101	1,93

O gráfico na Figura 4.2-1 mostra a curva de vazão na saída do canal em cada um dos casos, juntamente com a solução analítica.

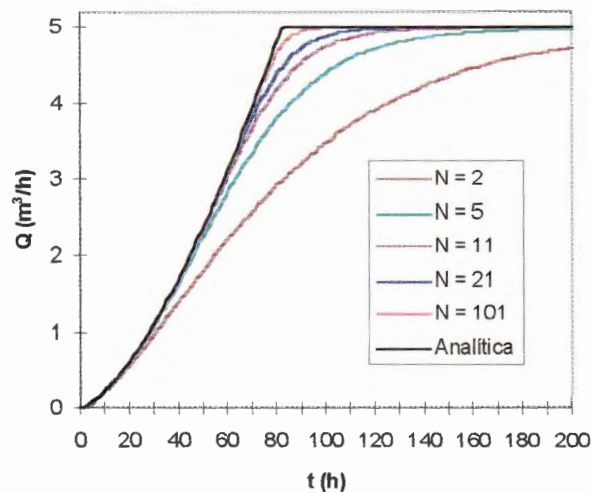


Figura 4.2-1: Solução Numérica por Diferenças Finitas para o Escoamento Superficial por Onda Cinemática ($N=3,5,11,21, 101$)

Pode-se concluir que a solução aproxima-se da solução analítica quando aumentamos o número de pontos (N). Porém, esse refinamento das subdivisões está limitado à memória primária do computador. Além disso, o aumento do número de pontos reflete no tempo de processamento, pois o aumento de variáveis pode comprometer a performance computacional.

4.2.2 Método dos Elementos Finitos

O método dos elementos finitos consiste em discretizar a região em diversos elementos. O desenvolvimento das equações do modelo da onda cinemática para a aplicação do método dos elementos finitos foi apresentada na seção 3.3.2, onde foram mostrados três modos distintos de equacionar o problema. Nesta seção, os resultados obtidos em cada um dos três casos serão comparados a fim de obter a melhor formulação para encontrar a solução numérica do escoamento superficial.

4.2.2.1 Substituição de Q em função de h na equação de continuidade:

Este primeiro método utiliza a formulação descrita na seção 3.3.2.1, que substitui a vazão pela sua relação com a altura do nível d'água, obtendo a velocidade do escoamento (V), que, neste caso, é considerada constante dentro do elemento.

Foram feitos testes utilizando 1 elemento (2 pontos), 4 elementos (5 pontos) e 10 elementos (11 pontos). Os valores de Δx correspondentes estão na Tabela 4.2-2. O erro não foi calculado porque a solução encontrada é muito distante da solução analítica.

Tabela 4.2-2: Discretização utilizada para o Método dos Elementos Finitos - Caso Unidimensional

Elementos	Δx	N
1	50	2
4	25	5
10	10	11

Utilizando-se o mesmo exemplo proposto, foram obtidos para a vazão de saída os resultados apresentados na Figura 4.2-2, comparados com a solução analítica.

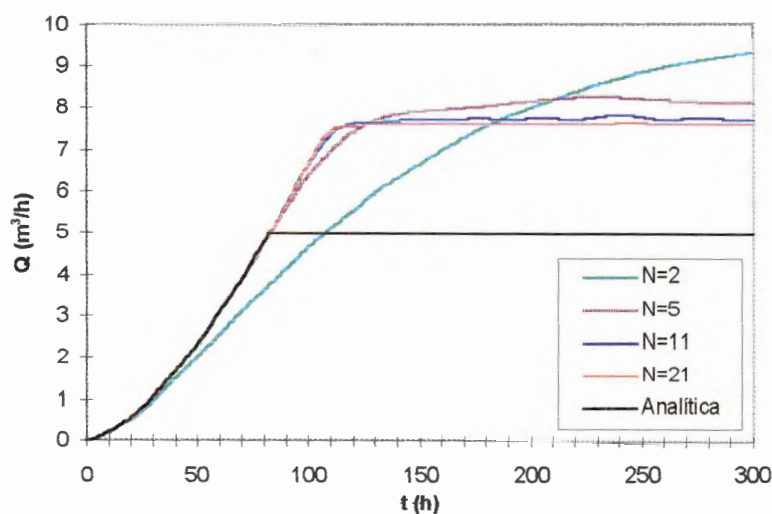


Figura 4.2-2: Solução numérica para o Escoamento Superficial por Onda Cinemática - Elementos Finitos - Solução 1

Como se pode constatar, esta solução apresenta resultados muito pouco satisfatórios. Na tentativa de se melhorar o resultado obtido, utilizou-se o método de Runge-Kutta, ordem 4, para o cálculo em função do tempo. Porém, o resultado obtido também não foi satisfatório, como pode ser constatado no gráfico apresentado na Figura 4.2-3.

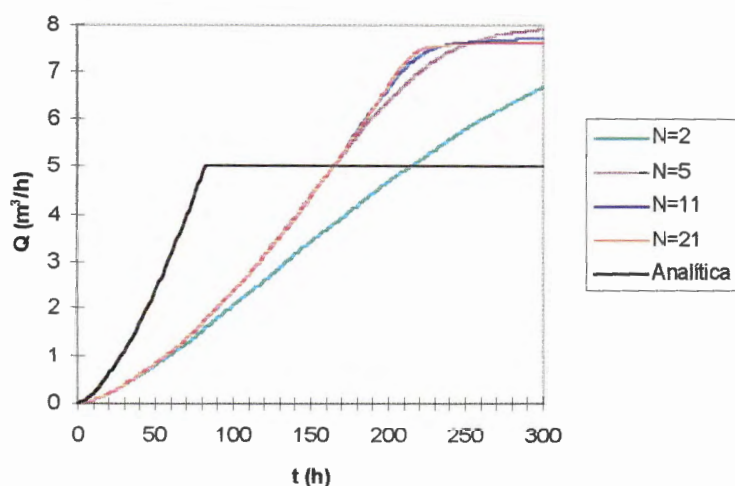


Figura 4.2-3: Aperfeiçoamento da Solução 1 - Método dos Elementos Finitos - Utilizando Runge-Kutta no tempo

4.2.2.2 Utilizando função básica linear para V e h .

Na seção anterior, o sistema de equações foi simplificado substituindo-se a vazão Q pela sua relação com a altura do nível d'água h na equação da continuidade, obtendo a velocidade do escoamento (V) constante em cada elemento. Com o objetivo de melhorar os resultados obtidos, foi desenvolvida uma formulação que aproxima V por uma função básica linear discretizada, da mesma forma como já foi feito com h . O desenvolvimento dessa formulação encontra-se na seção 3.3.2.2.

Para a solução da equação diferencial $A \dot{h} + Bh + C = 0$ em regime temporal, utilizou-se o método das diferenças finitas, da mesma forma como na seção anterior. Os resultados obtidos podem ser observados no gráfico a seguir:

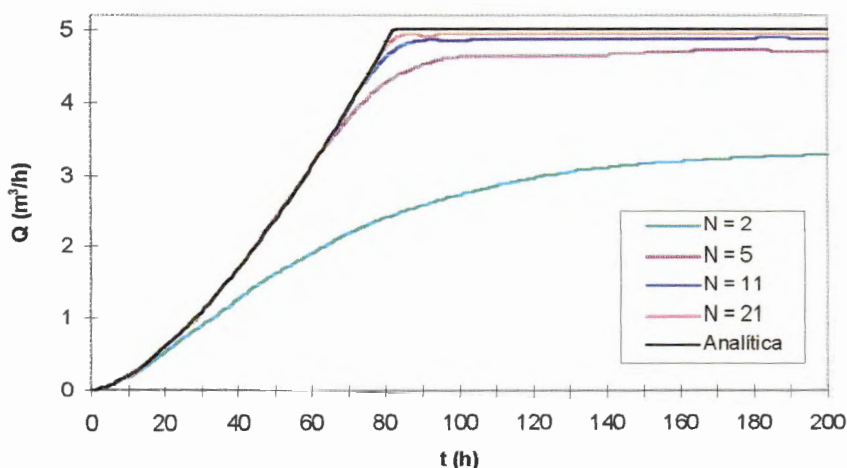


Figura 4.2-4: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Velocidade

Como se pode observar, a solução obtida aproxima-se da solução analítica à medida que se reduz o tamanho dos elementos. Porém, deve-se ressaltar que mesmo aumentando a discretização, o método utilizado não atinge o valor da solução analítica em regime permanente.

No cálculo anterior, a discretização temporal foi feita utilizando-se diferenças finitas utilizando o tempo anterior para calcular o valor atual. No sentido de melhorar a solução, um método alternativo de cálculo no tempo é utilizado. Neste método, o termo h , que era utilizado somente no tempo anterior, é substituído pelo valor médio entre o tempo anterior e o tempo atual (3.3-19). O resultado obtido com essa formulação pode ser observado na Figura 4.2-5.

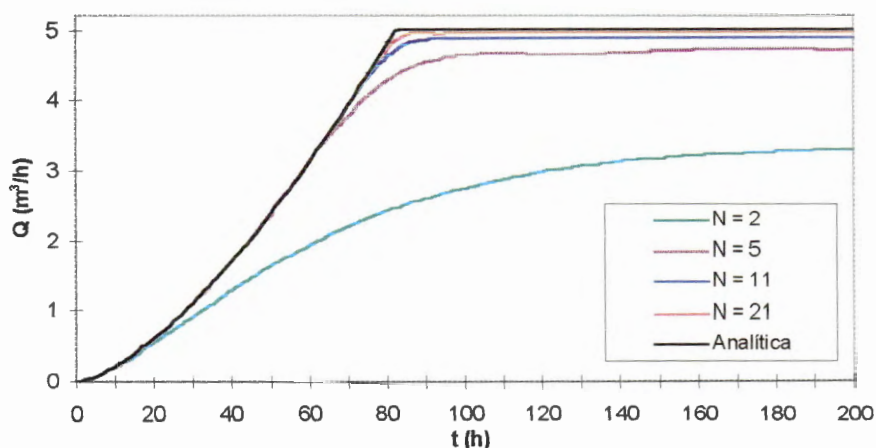


Figura 4.2-5: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Velocidade e Tempo Médio

O método de Runge-Kutta foi utilizado na tentativa de melhorar ainda mais a solução. No entanto, o resultado foi inferior ao mostrado no gráfico acima.

A diferença acumulada entre a solução analítica e os valores obtidos com a discretização em 20 elementos em cada um dos 3 casos pode ser observada na Tabela 4.2-3:

Tabela 4.2-3: Diferença entre o resultado numérico e a solução analítica: comparativo entre três métodos

Método	Erro
Diferenças Finitas	8,69
Tempo Intermediário	7,63
Runge-Kutta	8,77

No gráfico apresentado na Figura 4.2-6, pode-se observar as curvas de vazão de saída em cada um dos métodos acima comparados à solução analítica.

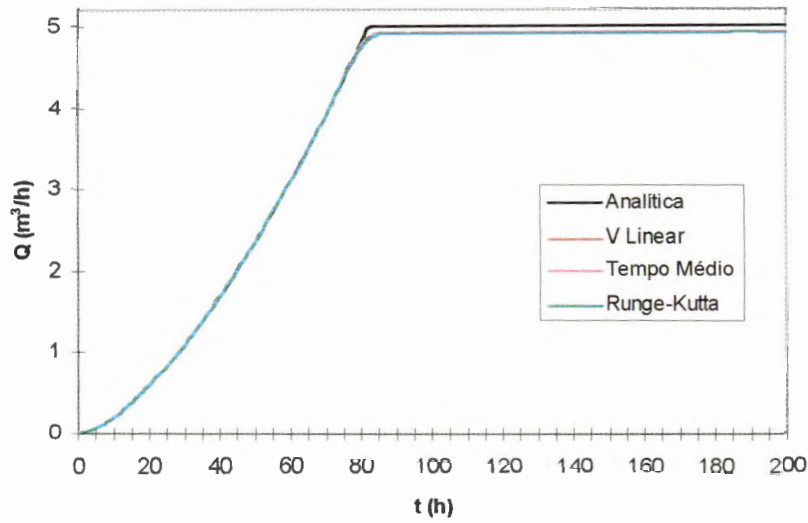


Figura 4.2-6: Solução numérica para o Escoamento Unidimensional - Elementos Finitos - Aproximação Linear para Velocidade - Comparativo entre Diferentes Métodos para o Processamento Temporal

Para observar melhor a diferença das soluções em regime permanente, na Figura 4.2-7 utiliza-se uma ampliação do gráfico no intervalo de tempo 80 a 200:

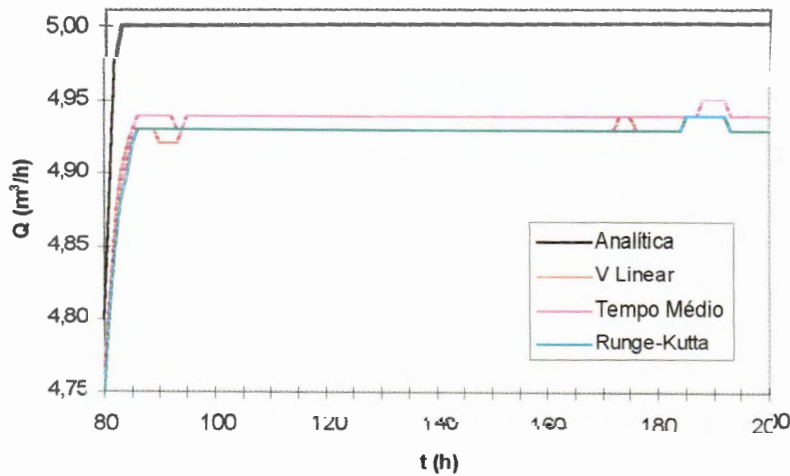


Figura 4.2-7: Solução numérica para o Escoamento Unidimensional utilizando Elementos Finitos com Aproximação Linear para Velocidade - Comparativo entre Diferentes Métodos para o Processamento Temporal no Intervalo 80 a 200

4.2.2.3 Utilizando aproximação linear para Q e h

Na formulação utilizada para a obtenção dos resultados apresentados nas seções anteriores, a função de vazão Q foi substituída, na equação da continuidade, pela sua relação com a altura do nível d'água h . Nesta seção, será utilizada a formulação apresentada na seção 3.3.2.3, onde Q não é substituído por h na equação da continuidade, mas sim por uma aproximação linear discreta.

A solução obtida com a utilização deste método pode ser observada no gráfico a seguir:

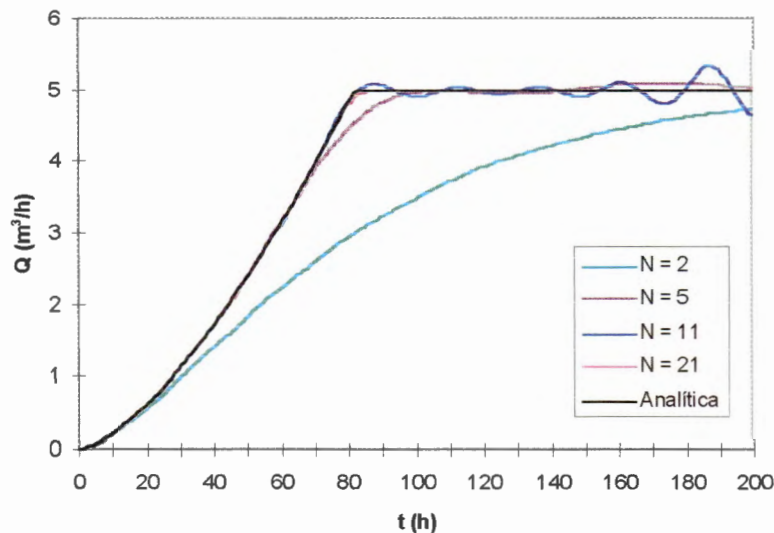


Figura 4.2-8: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Vazão

É importante ressaltar que neste método, a solução em regime permanente é atingida mesmo utilizando um único elemento. Já a solução em regime não permanente melhora com o aumento da discretização. O erro acumulado em cada uma das discretizações pode ser observado a seguir na Tabela 4.2-4.

Tabela 4.2-4: Erro acumulado para diversos níveis de discretização

Δx	N	Erro
100	2	147,76
25	5	11,82
10	11	10,93
5	21	0,78

Nesta forma de solução adotada, não é possível utilizar o tempo intermediário para melhorar a solução, da mesma forma como foi feito nos casos anteriores. Nesta formulação, o termo h não aparece na equação, pois Q foi mantido da forma original. É lógico que poderia ser utilizado o valor médio para Q , como em VIEUX et al. (1990), mas a equação não teria solução direta, sendo necessário utilizar um método iterativo de solução, que não consta dos objetivos deste trabalho.

Porém, pode-se alcançar melhorias significativas utilizando-se o método de Runge-Kutta, que praticamente mantém a mesma curva, mas inibindo a instabilidade apresentada na solução utilizando Diferenças Finitas no tempo. No gráfico a seguir, pode-se constatar essa afirmação:

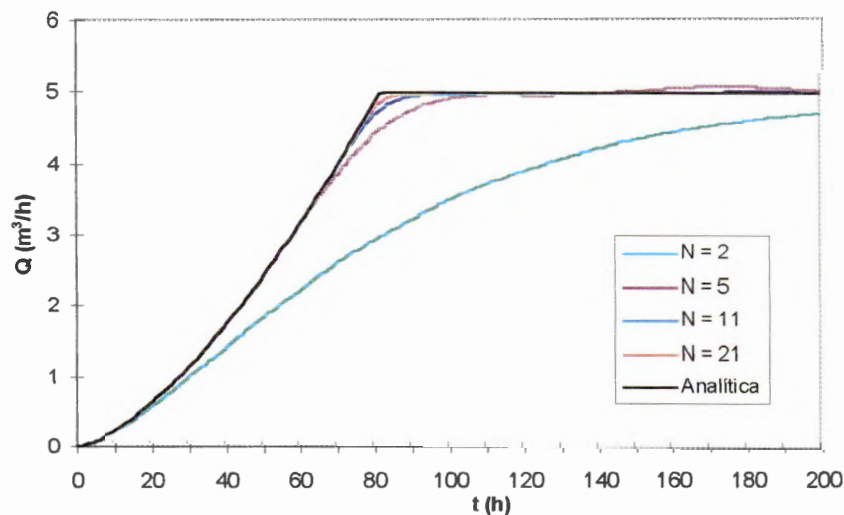


Figura 4.2-9: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Vazão - Runge Kutta no Processamento Temporal

Observando-se o erro acumulado nesta última formulação (Tabela 4.2-5), pode-se afirmar que este método apresenta soluções muito próximas à solução analítica.

Tabela 4.2-5: Erro acumulado para diversos níveis de discretização utilizando Runge-Kutta no processamento temporal

Δx	N	Erro
100	2	149,76
25	5	13,32
10	11	2,73
5	21	0,89

Para reiterar a melhor performance do método de Runge-Kutta, utilizou-se uma discretização com 40 elementos ($N=41$). Na figura pode-se verificar a instabilidade da curva com o aumento do tempo no método de Euler, enquanto o método de Runge-Kutta mantém a curva estável, e o erro acumulado diminui para 0,42.

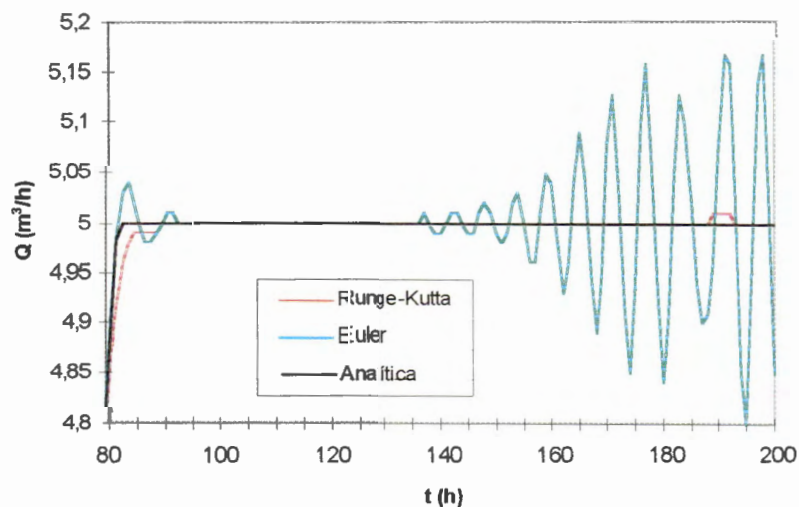


Figura 4.2-10: Solução numérica para o Escoamento Superficial Unidimensional - Método dos Elementos Finitos - Aproximação Linear para Vazão - Comparativo entre os Métodos Euler e Runge-Kutta no Processamento Temporal

4.2.3 Exemplos com intensidade de chuva variável

Os programas desenvolvidos para resolver numericamente o problema do escoamento unidimensional pelo modelo da onda cinemática foram testados comparando os resultados com a solução analítica, utilizando para isto intensidade de chuvas constante. No entanto, o programa permite utilizar também variações da intensidade da chuva no tempo.

Nesta seção, serão apresentados exemplos utilizando essa característica. O programa utilizado para a geração dos resultados é a última versão, que utiliza uma aproximação linear para a função de vazão e método de Runge-Kutta para cálculo no tempo.

Exemplo 1: Fim da chuva

Neste primeiro exemplo, será considerado intensidade de chuvas constante e terminando abruptamente, conforme a definição a seguir:

$$q(t) = \begin{cases} 50 \text{ mm/h} & \text{se } t < 80 \text{ h} \\ 0 & \text{se } t \geq 80 \text{ h} \end{cases}$$

A curva de vazão na saída do canal para este exemplo pode ser observada na Figura 4.2-11.

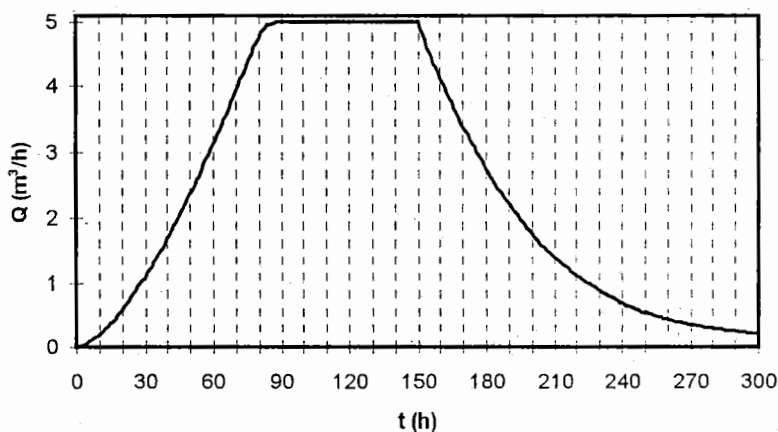


Figura 4.2-11: Escoamento unidimensional com término da chuva

Exemplo 2: Intensidade de chuva variando no tempo

Neste exemplo, será considerado uma variação maior na intensidade da chuva, conforme o gráfico mostrado na Figura 4.2-12. Neste caso, a chuva começa e termina em zero, e não há variação abrupta na intensidade da chuva.

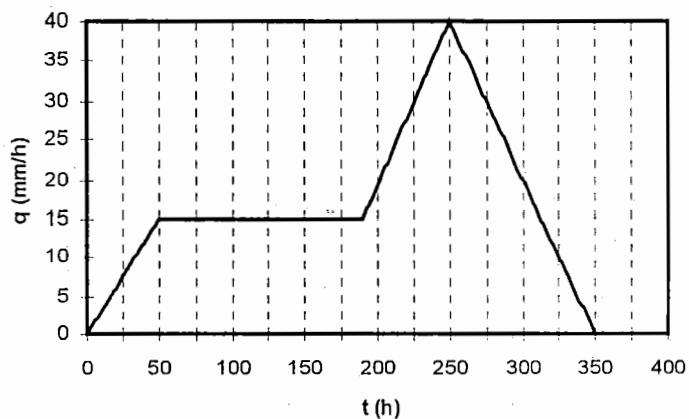


Figura 4.2-12: Intensidade da chuva variando no tempo

A solução numérica obtida para este exemplo é apresentada na Figura 4.2-13.

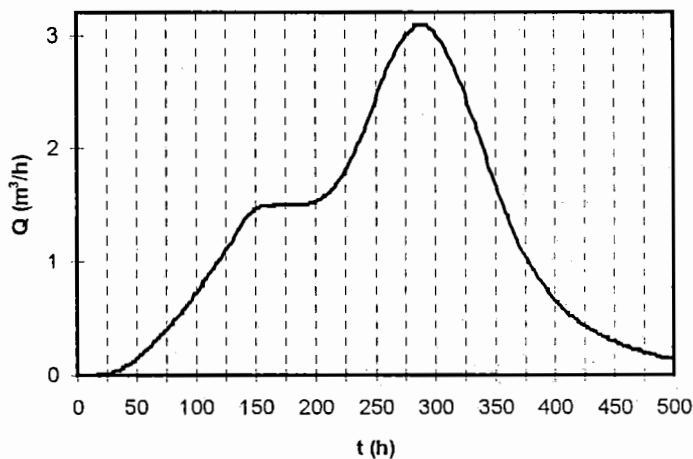


Figura 4.2-13: Escoamento unidimensional com chuva variando no tempo

4.3 Escoamento Superficial Bidimensional

Para encontrar a solução computacional sobre o modelo TIN, será utilizado um método semelhante ao proposto por GOODRICH et. al (1993), que utiliza o processamento faceta-por-faceta. Na maioria dos trabalhos utilizando elementos finitos, o processamento é feito sobre todos os elementos de forma conjunta, mesmo em estudos de escoamento superficial com modelagem TIN [VIEUX et al. (1990)]. Segundo GOODRICH et. al (1993), quando se utiliza escoamento superficial com o modelo da onda cinemática, não há razão para empregar uma solução global porque a natureza das equações não incorpora a influência das condições de contorno a jusante.

No modelo faceta-por-faceta, cada triângulo é processado individualmente, a partir dos elementos contendo os pontos mais elevados até os elementos que fazem parte da saída da bacia¹. A hidrógrafa de saída para cada elemento triangular deve ser armazenada, de modo que possa ser utilizada como condições de contorno para os elementos adjacentes. Quando a saída de um elemento se dá para um canal, a vazão de saída do elemento torna-se entrada lateral para o canal.

Deste modo, o método de resolução pode ser dividido nos seguintes passos:

1. Dividir a região em elementos triangulares
2. Definir a ordem de processamento dos elementos, de acordo com a altura de seus vértices
3. Identificar a rede de canais
4. Processar os elementos individualmente armazenando a hidrógrafa de saída de cada lado
5. Processar os lados que formam canais (escoamento unidimensional)

¹ Considera-se que a topografia é a propriedade dominante na formação da rede de drenagem

Para o processamento individual dos triângulos, o elemento deve ser subdividido, de modo a obter triângulos e pontos internos locais com as mesmas dimensões. A subdivisão é necessária devido ao formato do elemento, que permite a possibilidade de existirem dois lados do triângulo no contorno da bacia hidrográfica. Neste caso, os 3 vértices do triângulo devem ser considerados condições de contorno, com fluxo e altura nulos, e não haveria solução para esse triângulo.

GOODRICH et. al (1993) sugere a divisão em 16 triângulos internos, gerando 15 pontos locais do triângulo (Figura 4.3-2). Para este nível de discretização, o número máximo de valores desconhecidos é 10, caso em que existe somente um lado como condição de contorno e a saída ocorre através de dois lados. Quando a situação é invertida (dois lados no contorno e um na saída), existem valores desconhecidos em 6 pontos. Os três lados no contorno ou na saída não devem ser considerados, pois isto não deve ocorrer no caso do modelo da onda cinemática.

4.3.1 Resolução no elemento triangular

Nesta seção, serão avaliados os resultados da implementação quando aplicada a um único triângulo isoladamente.

Na solução adotada, os seguintes elementos são considerados como dados de entrada:

- As coordenadas (x,y,z) dos vértices do triângulo
- Os lados que devem ser considerados contorno
- Os lados sobre os quais há fluxo de saída do triângulo

O triângulo é, em seguida, subdividido em diversos outros triângulos, de modo a gerar pontos internos no triângulo. A seguir, pode-se observar duas diferentes

subdivisões internas para o triângulo, cada uma delas com maior discretização. Na Figura 4.3-1, utilizam-se 6 pontos e na Figura 4.3-2, 15 pontos.

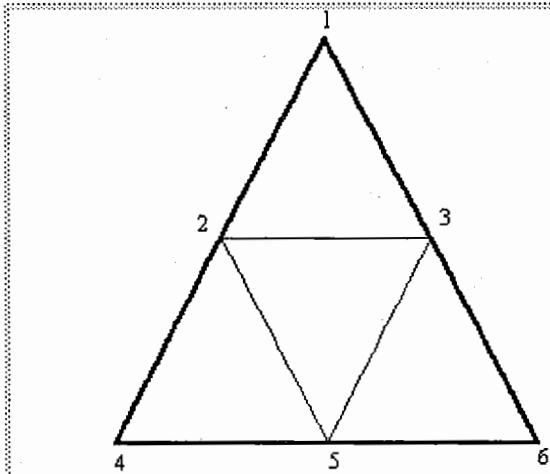


Figura 4.3-1: Triângulo subdividido em 4

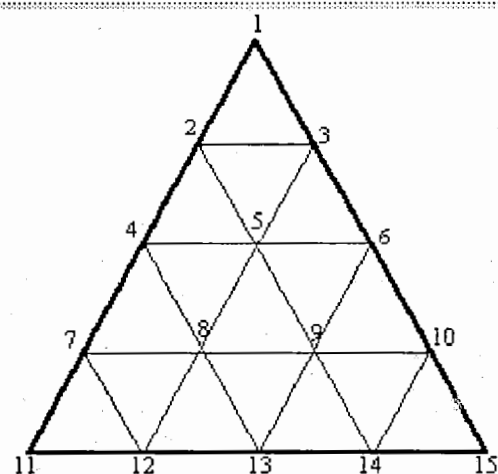


Figura 4.3-2: Triângulo subdividido em 16

Os triângulos internos são elementos finitos triangulares com dimensões idênticas e mesma declividade. O aumento na discretização interna diminui o efeito dos pontos pertencentes aos lados que são contorno, principalmente quando existem dois lados nessa condição. Porém, o programa torna-se mais lento, pois trabalha com matrizes maiores e exige um intervalo de tempo (Δt) menor, segundo a condição de Courant (equação 2.4-1).

Nesta seção, serão avaliados dois casos: triângulo de 6 pontos (Figura 4.3-1) e triângulo de 15 pontos (Figura 4.3-2).

Exemplo 1: Triângulo com base para cima

As coordenadas do elemento triangular utilizado neste exemplo (Figura 4.3-3) são dadas na Tabela 4.3-1.

Tabela 4.3-1: Coordenadas do Triângulo com base para cima

Vértice	x	y	z
1	5	0	70
2	0	100	90
3	10	100	90

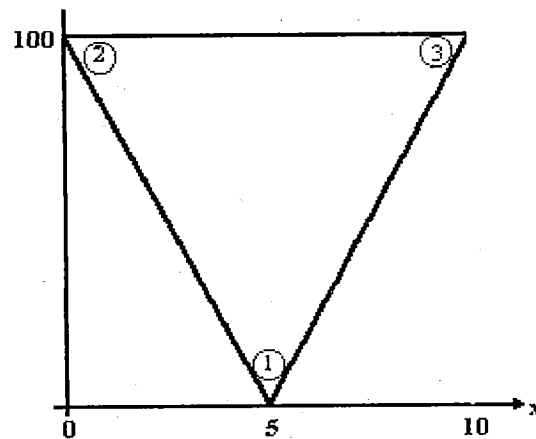


Figura 4.3-3: Triângulo com base para cima

Observa-se que, para os fins de teste de resultado com formulação analítica, o triângulo pode ser considerado como parte de uma superfície retangular com a mesma declividade. Assim, a solução obtida para o triângulo pode ser comparável à solução analítica no contorno de jusante da área retangular (seção 4.2).

A solução encontrada para os dois esquemas de discretização (6 e 15 pontos) são mostrados na Figura 4.3-4, juntamente com a solução analítica. Assim, a solução obtida para o triângulo é comparável com a solução analítica no contorno jusante da área retangular. Analisando o gráfico, pode-se constatar que ambas chegam ao resultado esperado em regime permanente.

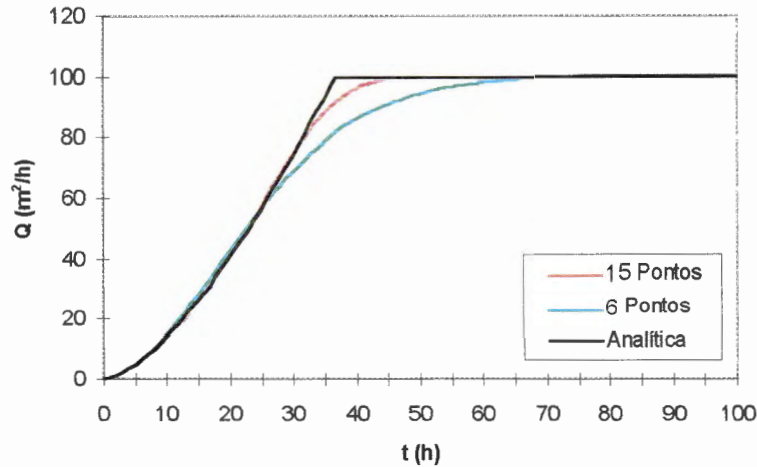


Figura 4.3-4: Solução para o triângulo com base para cima

Exemplo 2: Triângulo com base para baixo

O elemento triangular utilizado neste exemplo pode ser visto na Figura 4.3-5. Suas coordenadas são dadas na Tabela 4.3-2.

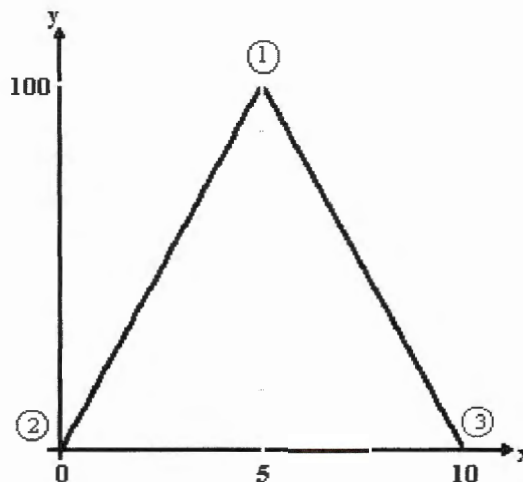


Figura 4.3-5: Triângulo com base para baixo

Tabela 4.3-2: Coordenadas do Triângulo com base para baixo

Vértice	x	y	z
1	5	100	90
2	0	0	70
3	10	0	70

A solução encontrada para este exemplo encontra-se na Figura 4.3-6. Para o triângulo de 6 pontos, somente o ponto interno 5 (Figura 4.3-1) apresenta valor não nulo, pois todos os outros pertencem ao contorno. Para o triângulo de 15 pontos, os pontos internos 5, 8, 9, 12, 13 e 14 (Figura 4.3-2) não pertencem ao contorno. Somente os valores pertencentes à base do triângulo foram apresentados na solução, juntamente com a solução analítica².

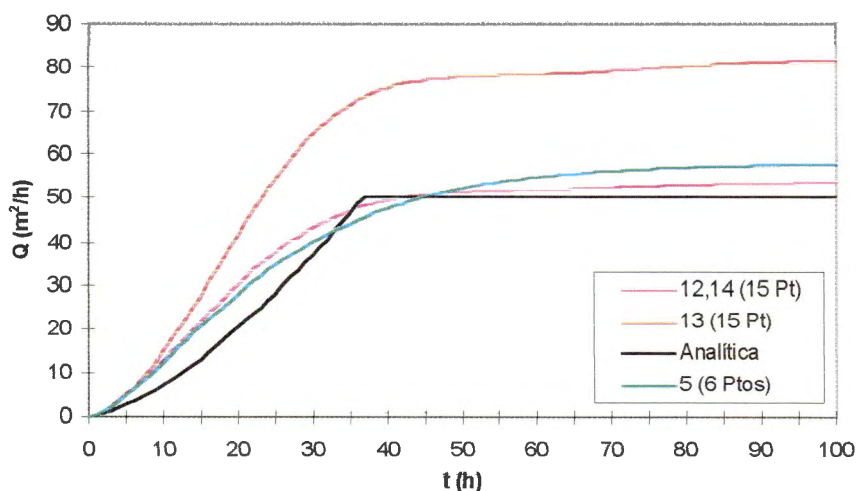


Figura 4.3-6: Solução para o triângulo com base para baixo

Observa-se que em nenhum dos dois esquemas de discretização a solução em regime permanente foi atingida com exatidão. Isto ocorre pela influência das condições de

² Obtida utilizando-se um retângulo com a mesma área do triângulo (10 x50).

Observa-se que em nenhum dos dois esquemas de discretização a solução em regime permanente foi atingida com exatidão. Isto ocorre pela influência das condições de contorno sobre os pontos internos. Porém, a utilização do ponto 5 (para o triângulo de 6 pontos) ou do ponto 12 ou 14 (para o triângulo de 15 pontos), podem ser utilizados como aproximações para a vazão.

Deve-se ressaltar, entretanto, que este problema só ocorre quando existem dois lados do triângulo no contorno da região. Se o triângulo recebe fluxo de outros elementos, o erro apresentado não ocorre, como será mostrado a seguir.

4.3.2 Comunicação entre elementos

Após analisar o processamento sobre cada elemento triangular, é necessário verificar a passagem do fluxo de um triângulo para outro. Todo triângulo, ao iniciar o processamento, deve verificar os fluxos incidentes para tomar os valores armazenados e inicializar as condições de contorno a cada iteração. Ao final do processamento, os valores calculados devem ser armazenados de forma adequada, de modo que possam ser utilizados em processamentos subsequentes.

Para validar o esquema de implementação em estudo, será utilizada uma região planar retangular, de escoamento unidimensional, dividida em 3 triângulos, conforme mostrado na Figura 4.3-7. As coordenadas dos vértices são apresentadas na Tabela 4.3-2.

Tabela 4.3-3: Coordenadas da região retangular dividida em 3 triângulos

Vértice	x	y	z
1	0	0	70
2	10	0	70
3	0	100	90
4	0	100	90
5	10	100	90

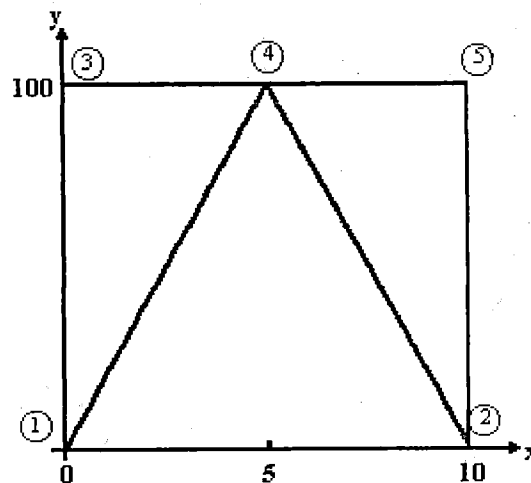


Figura 4.3-7: Região retangular dividida em 3 triângulos

O programa foi processado na região utilizando os seguintes valores:

- Intensidade de Chuva: $q = 1 \text{ m/h}$
- Coeficiente de Chézy: $C = 6$
- Coeficiente do escoamento cinemático: $m=1,5$

A solução obtida para o exemplo apresentado pode ser observada na Figura 4.3-8, onde são comparados os valores obtidos com a discretização em 6 pontos aos obtidos com a discretização em 15 pontos. Como já se havia constatado no caso unidimensional, o resultado obtido com a maior discretização se aproxima mais da solução analítica.

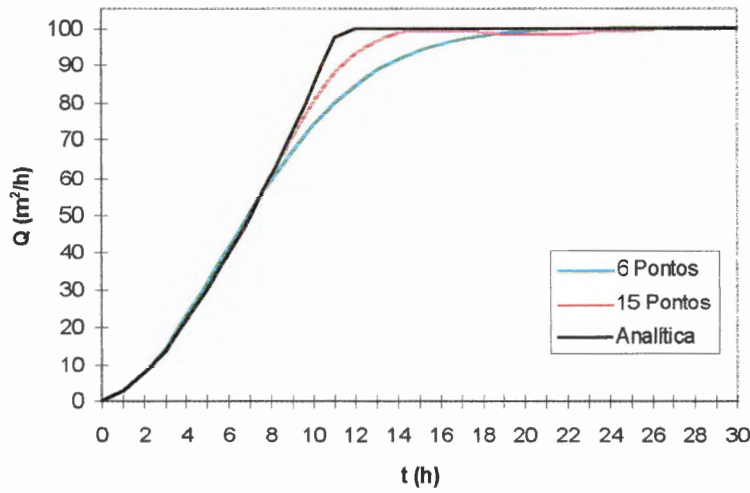


Figura 4.3-8: Solução para a região retangular dividida em 3 triângulos

A seguir, será apresentado um outro exemplo utilizando uma região retangular dividida em triângulos, porém utilizando uma forma diferente de discretização. A região planar retangular, de escoamento unidimensional, é dividida em 4 triângulos, conforme mostrado na Figura 4.3-9. As coordenadas dos vértices são apresentadas na Tabela 4.3-4.

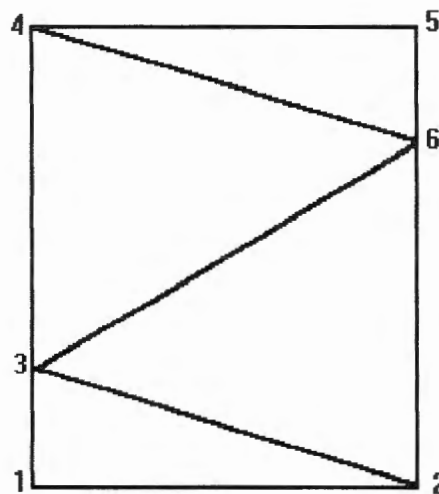


Figura 4.3-9: Região retangular dividida em 4 triângulos

Tabela 4.3-4: Coordenadas dos vértices da região retangular discretizada em triângulos

Vértice	x	y	z
1	0	0	60
2	10	0	60
3	0	25	70
4	0	100	100
5	10	100	100
6	10	75	90

Foram ainda utilizados os seguintes valores:

- Intensidade de Chuva: $q = 1 \text{ m/h}$
- Coeficiente de Chézy: $C = 6$
- Coeficiente do escoamento cinemático: $m=1,5$

Para o processamento dos triângulos, foram testados dois casos: triângulo de 6 pontos e triângulo de 15 pontos. Em ambos os casos, a solução em regime permanente foi atingida. Em relação ao regime não-permanente, o triângulo de 15 pontos apresentou melhor resultado, como pode ser conferido na Figura 4.3-10.

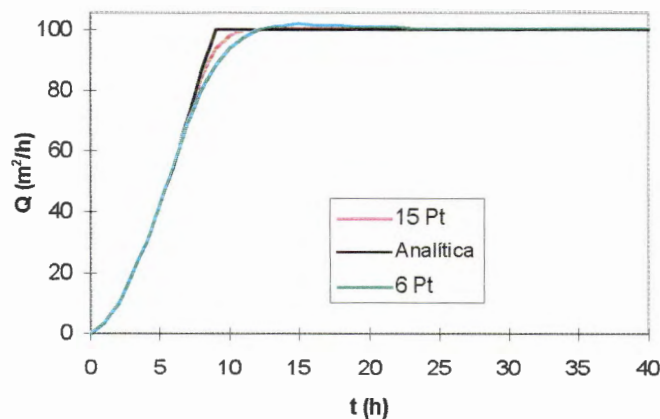


Figura 4.3-10: Solução da região dividida em 4 triângulos

4.3.3 Escoamento no Canal

Na maioria dos casos, as águas que escoam em uma superfície concentram-se em canais. Nesta seção, será utilizado um exemplo onde isso ocorre, como pode ser observado na Figura 4.3-11. As coordenadas correspondentes estão na Tabela 4.3-5 e os valores utilizados são:

- Intensidade de Chuva: $q = 1 \text{ m/h}$
- Coeficiente de Chézy: $C = 10$
- Coeficiente do escoamento cinemático: $m=1,5$

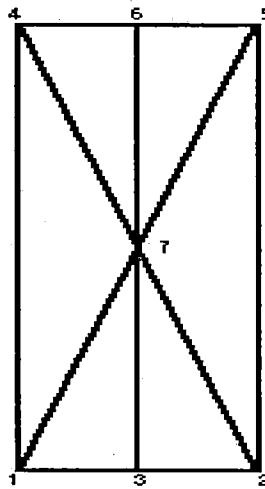


Figura 4.3-11: Região com Canal de Escoamento

Tabela 4.3-5: Coordenadas da Região com Canal de Escoamento

Vértice	x	y	z
1	0	0	70
2	10	0	70
3	5	0	50
4	0	100	200
5	10	100	200
6	5	100	180
7	5	50	115

A solução encontrada para este exemplo está na Figura 4.3-12, onde são apresentados as curvas de vazão obtidas na saída da bacia utilizando as discretizações em 6 pontos e 15 pontos.

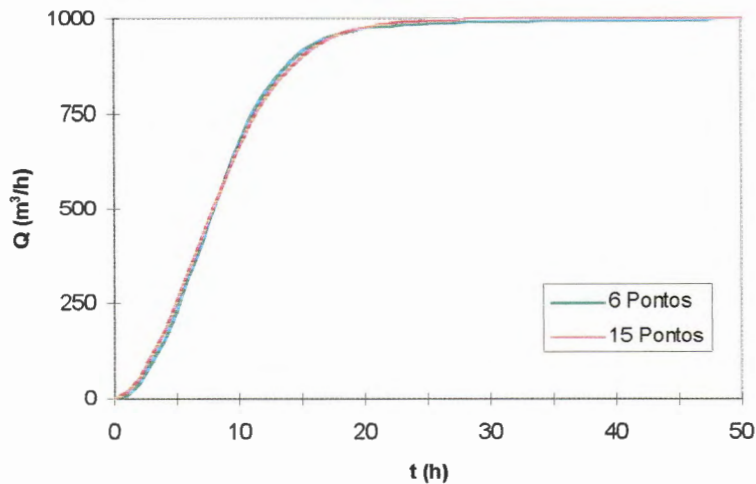


Figura 4.3-12: Solução da Região com Canal de Escoamento

4.3.4 Escoamento numa Bacia Hidrográfica

Nesta seção, será apresentada a solução para o escoamento em uma bacia hidrográfica completa. A região utilizada foi extraída de MOORE & GRAYSON (1991), e pode ser visualizada na Figura 4.3-13. Para a obtenção dos pontos discretos, foram selecionados alguns pontos ao longo das curvas de nível, utilizando valores pontuais de altitude, obtidos a partir das linhas de contorno (Figura 4.3-14). Esses valores são apresentados na Tabela 4.3-6.

Como nos casos anteriores, a bacia é dividida em triângulos (que são processados individualmente) cuja contribuição é levada aos canais (que são processados de forma unidimensional). A região discretizada pode ser observada na Figura 4.3-15. A partir desse modelo discretizado, o comportamento do fluxo de água em cada elemento triangular é obtido a partir de sua declividade. De acordo com esse comportamento, determina-se a sequência de processamento dos triângulos e identifica-se os canais. O resultado dessa análise pode ser observado na Figura 4.3-16.

Tabela 4.3-6: Coordenadas da Bacia Completa

Vértice	x	y	z
1	1,0	9,3	100,0
2	1,2	8,5	98,0
3	0,6	7,2	94,0
4	0,6	5,1	88,0
5	0,9	1,1	80,0
6	3,5	1,5	70,0
7	5,3	0,3	75,0
8	6,2	2,3	82,0
9	6,4	4,2	88,0
10	7,5	7,5	100,0
11	8,5	9,0	105,0
12	9,7	10,8	115,0
13	7,4	11,2	106,0
14	4,3	11,3	96,0
15	2,4	10,4	95,0
16	3,6	9,7	92,5
17	6,4	9,5	99,0
18	2,4	8,0	94,1
19	3,0	6,5	86,2
20	4,6	6,8	90,0
21	5,9	6,8	93,0
22	3,5	5,0	81,0
23	5,2	4,4	86,2
24	1,7	2,7	79,8
25	3,1	2,7	72,0
26	4,9	2,7	81,0

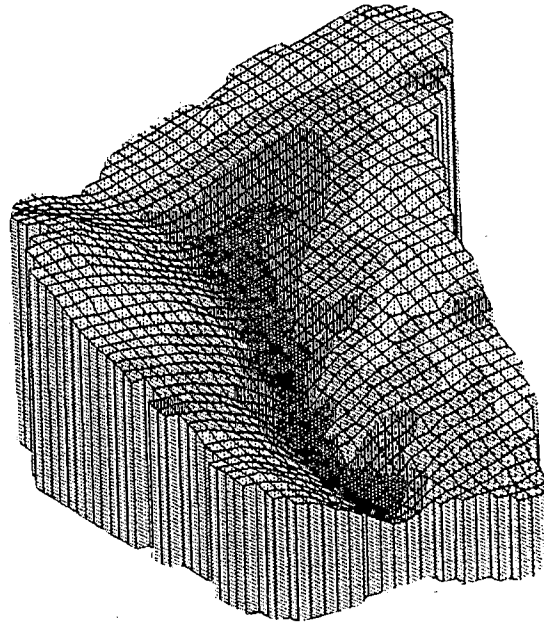


Figura 4.3-13: Visualização da Bacia Hidrográfica

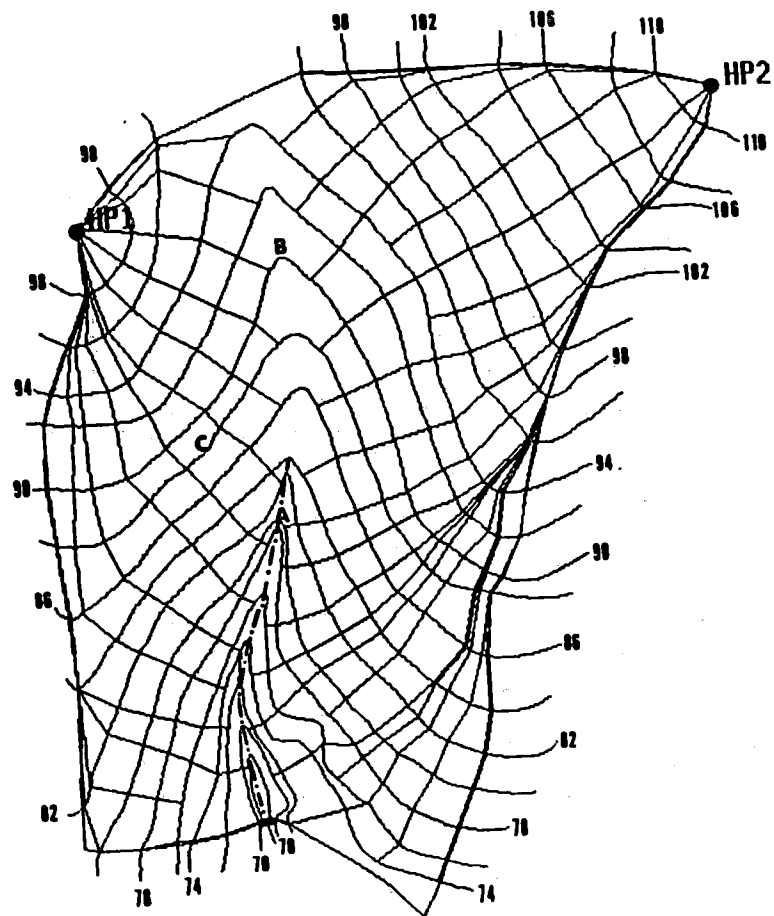


Figura 4.3-14: Linhas de Contorno da Bacia Hidrográfica

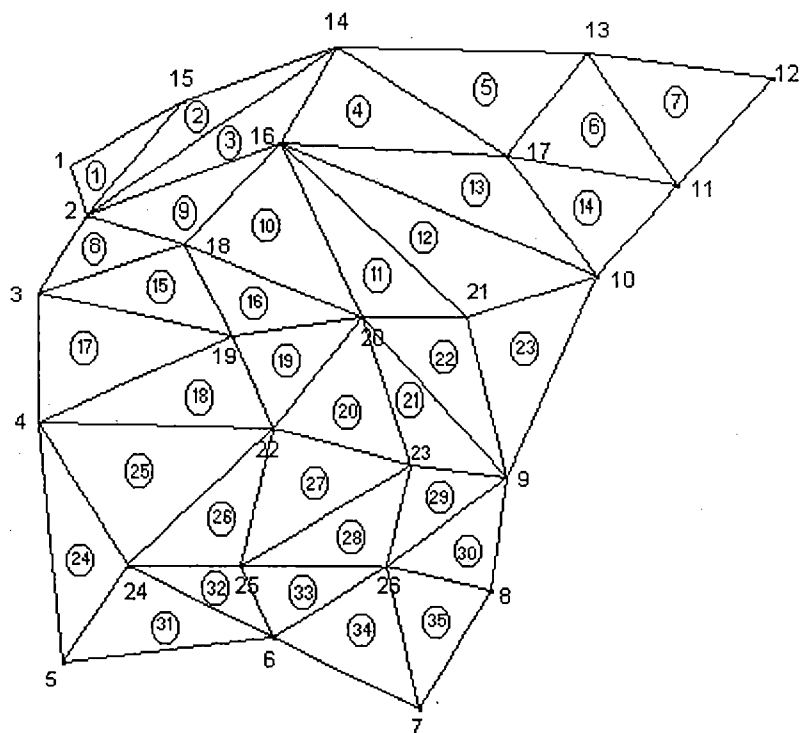


Figura 4.3-15: Divisão da região em elementos triangulares

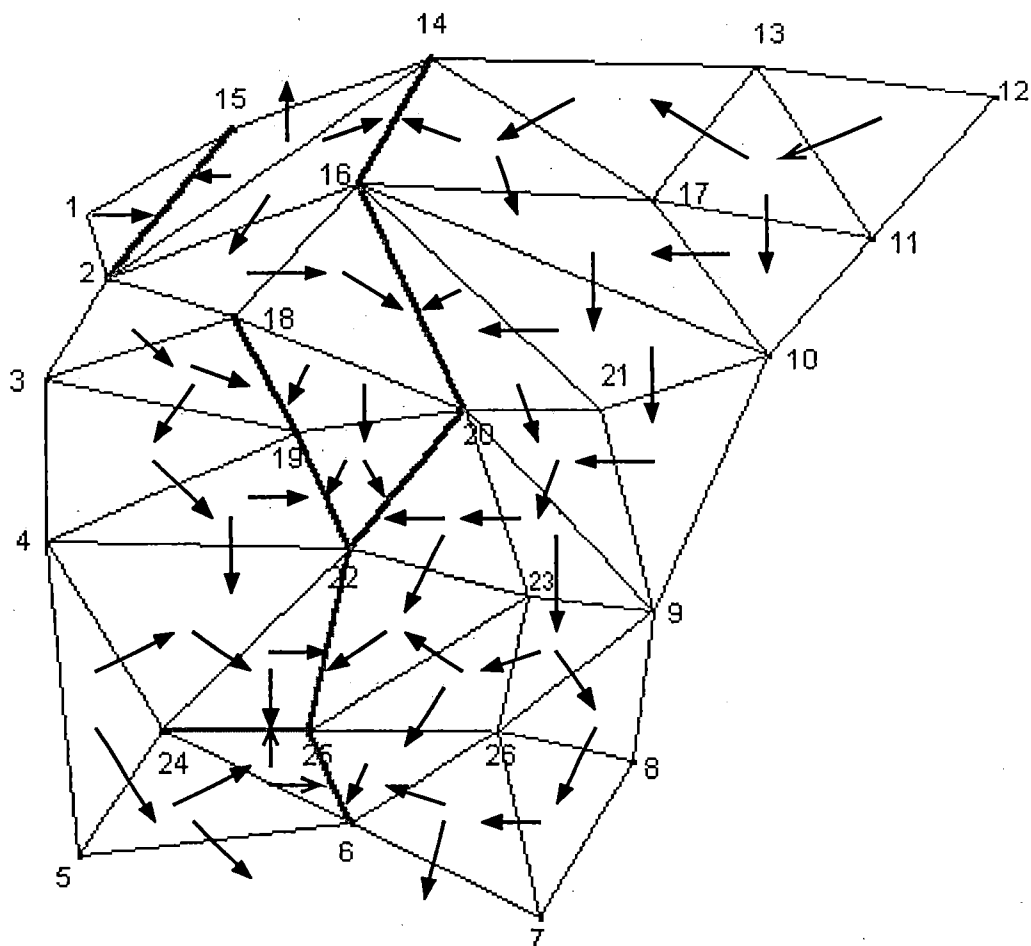


Figura 4.3-16: Sequência de Processamento dos Elementos e Formação dos Canais

Para encontrar a relação entre a vazão e o nível d'água será utilizada a equação de Manning,, juntamente com os seguintes valores:

- Coeficiente de resistência de Manning: $n = 0,11$
- Coeficiente da equação da onda cinemática: $m = 5/3$

Inicialmente, utilizou-se como entrada uma precipitação com intensidade da chuva constante: $q = 50 \text{ mm/h}$. A resposta obtida na saída do canal, em dois esquemas de discretização dos elementos, é mostrada na Figura 4.3-17.

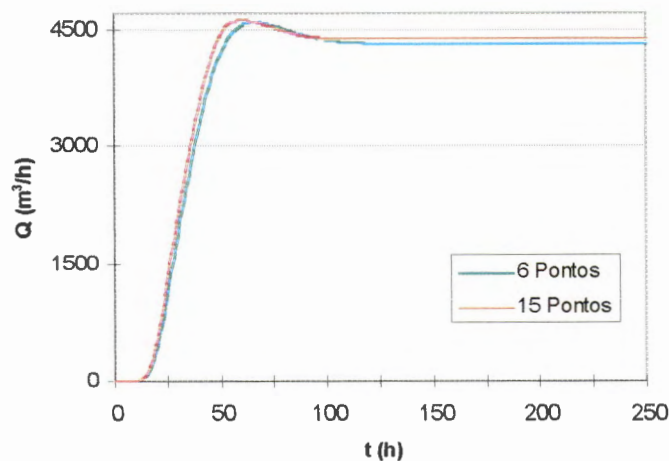


Figura 4.3-17: Vazão na saída do canal para o caso da Bacia Hidrográfica:
Intensidade da Precipitação constante

Para melhor analisar o comportamento da resposta gerada pelo programa face a casos reais, será utilizado o evento de chuva observado na região, destacado em MOORE & GRAYSON (1991), e apresentado na Figura 4.3-18. As respostas obtidas para a saída da bacia em dois esquemas de discretização pode ser observada na Figura 4.3-19.

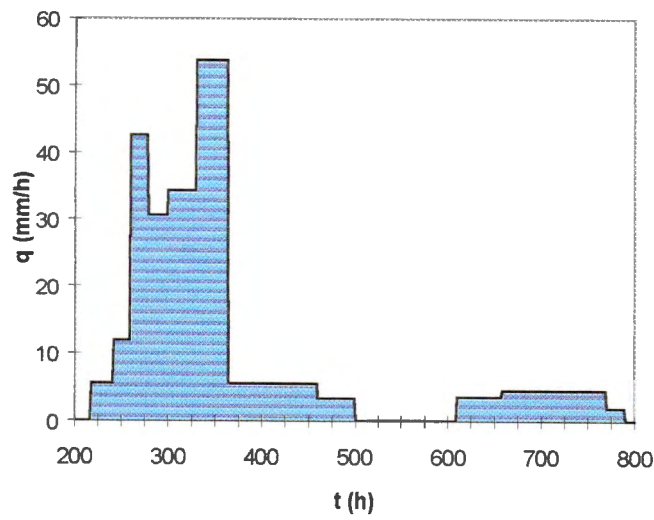


Figura 4.3-18: Intensidade da Precipitação variável no tempo para a Bacia Hidrográfica.

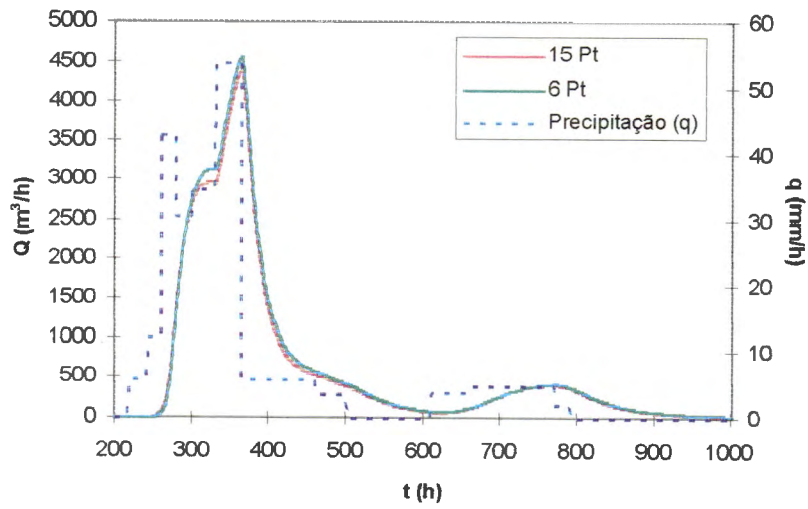


Figura 4.3-19: Vazão na saída do canal para o caso da Bacia Hidrográfica: Intensidade da Precipitação variável no tempo

Na Figura 4.3-20 pode-se observar diferentes curvas de vazão, obtidas para pontos internos da bacia. Estes valores correspondem ao esquema de discretização dos triângulos em 6 pontos.

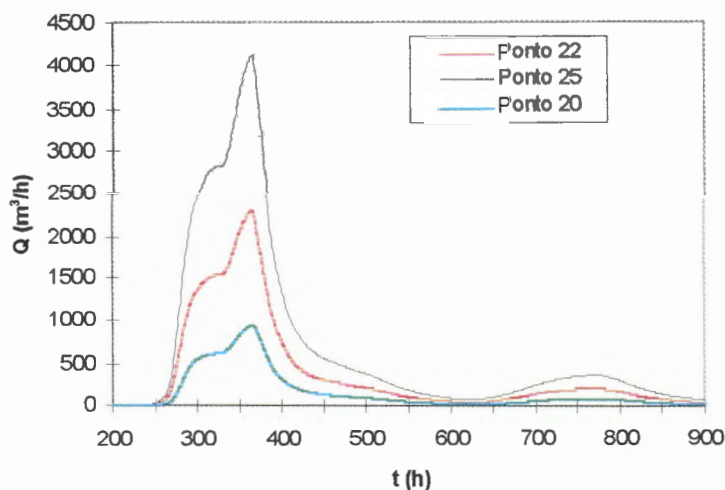


Figura 4.3-20: Vazão em pontos internos para a Bacia Hidrográfica com Intensidade da Precipitação variável no tempo

Na Figura 4.3-21 encontram-se os valores observados na saída da bacia, bem como os valores obtidos por MOORE & GRAYSON (1991). Observe-se que a vazão é apresentada em mm/h (vazão por unidade de área), a mesma medida utilizada na medida da precipitação. Para melhor comparar a solução obtida pelo programa aqui apresentado com os valores observados, as curvas de vazão apresentadas na Figura 4.3-19 serão divididas pela área total da região. O resultado obtido é apresentado na Figura 4.3-22. Deve-se observar que, na vazão observada, existe perda de água por infiltração, o que não ocorre no resultado obtido pelo programa, que não incorpora tais perdas. Deste modo, os resultados obtidos foram satisfatórios, comprovando a viabilidade da utilização do método apresentado em bacias hidrográficas, mesmo em regime não permanente.

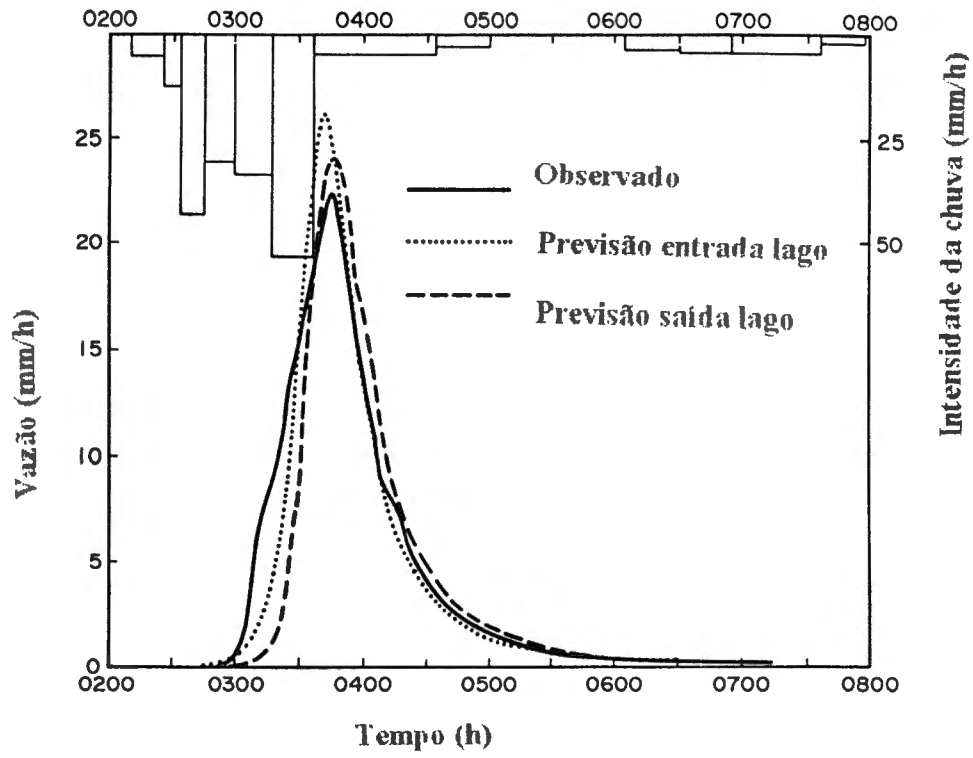


Figura 4.3-21: Vazão observada na saída da Bacia Hidrográfica

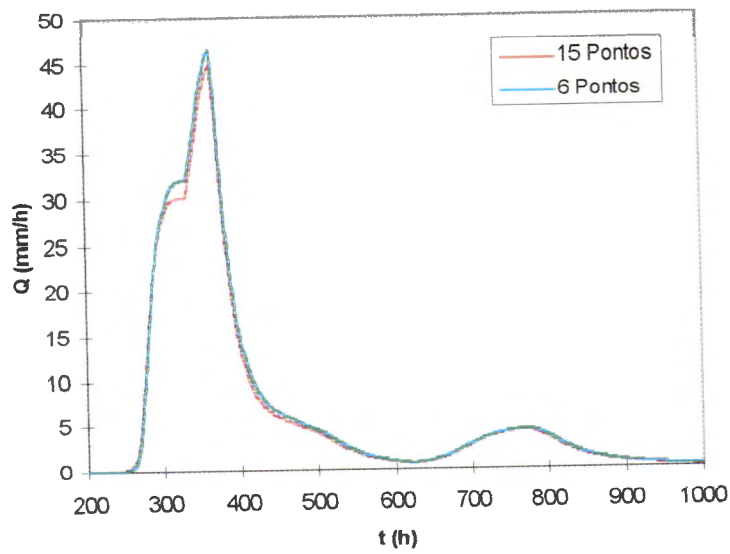


Figura 4.3-22: Vazão por unidade de área obtida na saída da Bacia Hidrográfica

4.3.5 Aspectos da Implementação

O programa que soluciona o problema da quantificação do escoamento bidimensional utilizando o método dos elementos finitos foi implementado na linguagem Pascal, utilizando o compilador Borland Turbo Pascal.

Entrada de Dados

A entrada de dados é feita a partir de 4 arquivos, contendo todas as informações necessárias à identificação da região sobre a qual o escoamento deve se desenvolver. As informações armazenadas em cada arquivo são descritas a seguir:

- Coord.com

Este arquivo possui todos os pontos da região, que correspondem a vértices dos triângulos, identificando a localização espacial (coordenadas x, y e z). Os registros possuem o seguinte formato:

Número do vértice	Coordenada X	Coordenada Y	Coordenada Z
-------------------	--------------	--------------	--------------

- Triang.com

Identifica cada triângulo, a partir de seus 3 vértices.

Número do triângulo	Vértice i	Vértice j	Vértice k
---------------------	-----------	-----------	-----------

- Lados.com

Identifica cada lado que pertença a um triângulo, a partir de seus dois pontos extremos

Número do lado	Vértice 1	Vértice 2
----------------	-----------	-----------

- Ordem.com

Neste arquivo encontram-se as informações relativas à ordem de processamento dos triângulo e como é feita a comunicação entre triângulos.

Possui o seguinte formato:

Tipo Elemento	Número Elemento	Qtidade Informações	Registro Informações
---------------	-----------------	---------------------	----------------------

Tipo Elemento é um caracter que identifica se o elemento a ser processado é um triângulo (T) ou um lado (L), que corresponde a um trecho do canal. No primeiro caso, utiliza-se escoamento bidimensional; no segundo, escoamento unidimensional.

Número Elemento é o número do triângulo ou do lado que deve ser processado.

Qtidade Informações indica se Registro Informações armazena 1, 2 ou 3 informações.

Registro Informações apresenta informações a respeito das interfaces de comunicação do triângulo. Cada elemento de comunicação pode ser um lado ou um vértice, e é representado da seguinte forma:

Número	Tipo de interface
--------	-------------------

Número pode ser o número de um lado ou de um vértice

Tipo de interface pode ter os valores **e** (entrada), **s** (saída) ou **v** (vértice). Os valores **e** e **s** indicam que se trata de um lado. No caso de um vértice (**v**), indica sempre entrada.

Processamento dos arquivos de entrada

O processamento é feito da seguinte forma:

- No arquivo Ordem.com, é selecionado o elemento a ser processado e suas interfaces.
- Se o elemento da interface é um vértice ou uma entrada, o arquivo associado ao elemento é aberto para leitura. A cada iteração no processamento é feita uma leitura no arquivo apropriado e os valores lidos são utilizados como condição de contorno.
- Se o elemento é saída, o arquivo associado ao elemento é aberto para escrita. A cada iteração, o valor calculado é armazenado no arquivo.
- Os arquivos de interface são criados e gerenciados pelo próprio programa; eles são nomeados da seguinte forma:

1. Lados: Y + NroLado + .LIG
2. Vértices: P + NroVertice + .PTO

Comunicação entre elementos

A comunicação é feita utilizando os valores da vazão em cada ponto. No instante do armazenamento, os valores da altura são transformados em vazão de acordo com o valor α (que é proporcional à declividade) do triângulo sobre o qual os valores estão sendo calculados.

Alguns destes valores tornam-se condições de contorno para triângulos que serão processados posteriormente. Quando um desses outros triângulos é processado, ocorre o processo inverso, transformando a vazão armazenada em altura, de acordo com o valor α deste novo triângulo. Quando o lado é tratado como canal, os valores que foram armazenados na saída do triângulo tornam-se entrada lateral para o canal. Neste caso, não há necessidade de conversão, pois na entrada lateral utiliza-se diretamente a vazão.

Armazenamento dos Resultados

O programa cria arquivos de saída em formato ASCII para que os resultados obtidos possam ser observados e utilizados. Para cada elemento processado é criado um arquivo do seguinte modo:

ALT + Nro_Triângulo + .DAT

Nesse arquivo são armazenados todos os valores em cada um dos pontos internos do triângulo a cada intervalo de tempo. Os valores são organizados de forma que cada linha representa um intervalo de tempo e cada coluna um ponto interno, facilitando a observação e utilização dos valores, permitindo facilmente editá-los e representá-los graficamente.

Da mesma forma, cria-se um arquivo para cada ponto da região (correspondente a um vértice do triângulo). Esses arquivos possuem a seguinte forma:

P + Nro_Vértice + .PT

A organização é semelhante aos arquivos de elementos, porém, estes possuem uma única coluna, já que apresentam um único valor a cada intervalo.

Em ambos os arquivos, os valores armazenados podem ser vazão ou altura do nível d'água. Deste modo, pode-se colocar a vazão em ambos, a altura em ambos, ou a vazão em um e altura em outro. Para alterar de um modo para outro, deve-se modificar uma linha de comando no procedimento **Grava_Vetor**.

Vazão de Entrada do Sistema

A vazão de entrada do sistema, representada pela intensidade da chuva, é simulada por uma única função: **QChuva**. Deste modo, pode-se facilmente alterar a função para utilizar os seguintes valores na entrada:

- um valor constante
- uma função no tempo $f(t)$
- uma função no espaço $f(x,y)$
- uma função no espaço-tempo $f(x,y,t)$
- valores discretos obtidos de arquivo

4.3.6 Definição da sequência de processamento dos elementos

Como visto na seção anterior, para o processamento bidimensional, necessita-se da presença de 4 arquivos: **Coord.com**, **Triang.com**, **Lados.com** e **Ordem.com**. Os dois primeiros retratam a configuração geográfica do ambiente e a discretização TIN. Os dois últimos são utilizados na sequência de processamento na comunicação entre os elementos.

A definição dos arquivos de entrada para o processamento bidimensional apresentada na seção anterior pode ser uma tarefa não muito complexa quando trabalhamos com poucos elementos. No entanto, com o aumento da discretização da região, gerando triângulos com declividades distintas, a geração dos arquivos torna-se muito complexa, principalmente o arquivo que dirige todo o processamento (**Ordem.com**), que precisa retratar claramente, além da ordem de processamento, todas as interfaces de entrada e saída com os triângulos adjacentes.

- um arquivo contendo as coordenadas dos pontos (Coord.com),
- um arquivo contendo os triângulos em função dos pontos (Triang.com),
- Um arquivo contendo os lados que pertencem ao contorno da região.

A partir dessas informações, o programa realiza toda a parte “inteligente” do processamento, gerando os arquivos necessários para o programa descrito na seção anterior, que realiza somente o processamento numérico.

Para o desenvolvimento do programa que define a sequência de processamento dos elementos, foi utilizada a linguagem de programação lógica Prolog. A escolha de uma linguagem lógica deve-se ao seguinte fato: o programa em questão utiliza muito pouco de processamento numérico, pois o único cálculo envolvido é o cálculo do gradiente e área de triângulos. Por outro lado, utiliza-se uma grande quantidade de tomadas de decisão e processamento de listas. A linguagem Prolog, devido a seu considerável poder de expressão [FRANCO & MONARD (1992b)], é a linguagem mais adequada para o processamento simbólico, decisões e para trabalhar com listas.

O programa trabalha da seguinte forma:

1. As informações contidas nos 3 arquivos são lidas e transformadas em relações Prolog
2. As informações contidas no arquivo **Triang.com** são processadas de modo que os vértices fiquem na posição correta para o processamento dos elementos. Este passo envolve um processo de decisão complexo, pois não basta que os vértices estejam em sentido anti-horário: existem regras para definir os vértices como **i**, **j** ou **k**, fundamentais ao processamento, mas que não é importante detalhar neste ponto. Isto é feito pela relação **acerta_vertices/4**.
3. É gerada a relação adjacentes/2, onde se define, para cada triângulo, os triângulos adjacentes. Este passo necessita da informações dos lados que

definir os vértices como **i**, **j** ou **k**, fundamentais ao processamento, mas que não é importante detalhar neste ponto. Isto é feito pela relação **acerta_vertices/4**.

3. É gerada a relação **adjacentes/2**, onde se define, para cada triângulo, os triângulos adjacentes. Este passo necessita das informações dos lados que são contorno. Este passo também facilita a tarefa do usuário, pois é muito mais simples informar os lados que são contorno que informar todas as vizinhas de cada triângulo. A identificação da vizinhança é, portanto, feita pelo programa.
4. Calcula-se, para cada triângulo, o gradiente e os ângulos ψ , ϕ_1 , ϕ_2 e ϕ_3 . O ângulo ψ depende do gradiente, e é calculado de acordo com a equação (4.3-2). Os ângulos ϕ_i são relativos aos lados do triângulo (Figura 4.3-23).
5. Calcula a direção do fluxo de água no triângulo, indicando os lados pelos quais o fluxo entra e os lados por onde a água deixa a região. Isto é feito de acordo com os esquemas mostrados na seção 3.2.3. Quando há saída de fluxo por dois lados, o programa calcula ainda o percentual que vai para cada um deles, de acordo com as equações (4.3-4) e (4.3-5). Os valores calculados são armazenados na relação **fluxo/3**, que contém o triângulo atual, o triângulo para onde o fluxo de água vai e a porcentagem.
6. Quando todos os triângulos já foram processados, identificam-se os condutores de água (trechos do canal). Isto é feito utilizando-se a relação **fluxo/3** e o método apresentado na seção 3.2.4.
7. São criadas as relações **envia/2** e **recebe/2**, contendo, para cada triângulo, uma lista de todos os triângulos para os quais envia ou recebe fluxo. A partir destas, são feitas as relações **envia_lado/2** e **recebe_lado/2**, indicando, para cada triângulo, os lados para os quais envia e dos quais recebe fluxo de água.
8. As relações **envia/2** e **recebe/2** são utilizadas para gerar a ordem de processamento dos triângulos. O processo é feito da seguinte forma:

- Utilizando a relação **recebe/2**, são selecionados os triângulos que não recebem fluxo de nenhum outro.
- Para o triângulo selecionado, verifica-se na relação **envia/2** os triângulos que recebem fluxo dele. Para cada um destes triângulos, retira-se da relação **recebe/2** o triângulo recém-selecionado, diminuindo a lista.
- As listas de triângulos na relação **recebe/2** vão diminuindo até tornarem-se vazias. Quando isto ocorre, o triângulo já pode ser selecionado.
- Os condutores são tratados da mesma forma, apresentando condições de seleção a partir do momento que os triângulos adjacentes já foram selecionados.
- O processo se repete até que todos os triângulos e condutores sejam selecionados.
- A ordem de processamento é armazenada na relação **ordem/1**.

9. São identificados todos os lados na região. Os lados são numerados e armazenados na relação **lados/3**.

10. São gerados os arquivos **LADOS.COM** e **ORDEM.COM**

- A geração do arquivo **LADOS.COM** é direta, a partir da relação **lados/3**.
- Para gerar o arquivo **ORDEM.COM**, a lista em **ordem/1** é percorrida, de modo que cada elemento torna-se uma linha no arquivo.
- Utilizando-se as relações **envia_lado/2** e **recebe_lado/2**, é possível gerar o arquivo da maneira desejada, conforme apresentado na seção 4.3.5.

5. Conclusão

5.1 Avaliação dos Resultados

Neste trabalho, desenvolveu-se uma ferramenta computacional para previsão do escoamento superficial em uma bacia hidrográfica. A ferramenta realiza a identificação da direção do fluxo e a quantificação da vazão, permitindo que se possa simular o comportamento hidrológico da bacia hidrográfica a partir de dados digitais da topografia da região, facilitando o controle de recursos hídricos, projetos hidráulicos e previsão de cheias.

A identificação da direção do fluxo envolve uma pequena parcela de cálculo matemático, somente o necessário para calcular o gradiente em cada região triangular e a inclinação de cada lado do triângulo. No entanto, a partir dessas informações, existem diversos aspectos envolvendo decisões em um grande número de variáveis. Por este motivo, nesta fase são utilizadas técnicas de Inteligência Artificial, e o programa é desenvolvido na linguagem de programação Prolog, mais adequada que as linguagens procedurais para solucionar o problema apresentado. O programa trabalha diretamente sobre o modelo de elevação digital representado como rede triangular irregular e gera todas as informações necessárias para o processamento subsequente, indicando a ordem de processamento dos elementos e identificando os canais para o processamento unidimensional do escoamento.

A quantificação da vazão utiliza as equações do modelo de onda cinemática e métodos numéricos para calcular a vazão dentro da região. O programa foi

implementado utilizando-se a linguagem Pascal, e o processo é gerenciado por um arquivo contendo informações quanto à ordem do processamento de cada elemento e suas comunicações com outros elementos ou canais. Estas informações são geradas pelo programa anterior, de modo que este preocupa-se somente com o processamento numérico. O programa trabalha com informações espaciais e temporais, de modo que pode-se obter a vazão em qualquer ponto da região em qualquer tempo. Para o processamento das informações no espaço, utiliza-se o método dos elementos finitos, tanto no caso bidimensional, para o processamento dos elementos triangulares, quanto no caso unidimensional para o processamento dos canais. Para o processamento no tempo, utiliza-se o método das diferenças finitas.

A informação sobre a vazão incidente na região sob a forma de entrada lateral e águas pluviais é gerada por uma função, que tanto pode assumir um valor constante, como uma função do tempo e do espaço, como também informações discretas obtidas por sensores ou um arquivo de entrada. Para cada um dos casos, é necessário somente alterar o corpo da função geradora das informações, sem que se necessite alterar o restante do programa. Essa função foi implementada de forma a passar ao programa a entrada lateral efetiva em cada sub-região a cada intervalo de tempo.

Deve-se destacar ainda que muitos trabalhos desenvolvidos para o processamento bidimensional não utilizam as equações bidimensionais no cálculo. Nestes casos, são utilizadas as equações unidimensionais e os elementos triangulares ou retangulares são rotacionados e vistos em uma única dimensão. Estes procedimentos simplificam os cálculos bidimensionais, que são realmente muito complexos, tanto no desenvolvimento matemático quanto no processamento computacional. Porém, em alguns casos, essa simplificação apresenta alguns problemas na representação das informações, como vazão de entrada variando no espaço ou obtenção da vazão de saída em qualquer ponto do espaço.

Foram realizados diversos testes para validação da metodologia utilizada. Inicialmente, foram utilizados exemplos que possuem solução analítica. Os bons

resultados obtidos pelo processamento numérico viabilizaram sua utilização geral. Nos testes seguintes também foram obtidos resultados satisfatórios, comprovando ser viável a utilização da modelagem bidimensional.

A divisão do programa em duas partes simplifica o processamento numérico, pois cada elemento triangular é processado individualmente e pode ainda ser subdividido, de forma a aumentar a discretização para a obtenção de resultados mais apurados. Deste modo, foi possível executar o programa em um microcomputador IBM-PC 386 Dx 33Mhz. O processamento da região completa, com todos os elementos integrados na mesma matriz, exigiria a utilização de equipamentos mais poderosos, como supercomputadores. A primeira parte do processamento permite ainda avaliar a discretização utilizada, pois permite visualização dos canais e áreas de contribuição, sendo possível aumentar a discretização se o resultado não for satisfatório ou diminuí-la para acelerar o processamento numérico.

A divisão do sistema em duas partes também agiliza o processo de simulação, pois após o processo de identificação do direcionamento do fluxo, é possível obter diversos hidrogramas, alterando-se as características hidrológicas da bacia (tipo de solo, vegetação, etc.) ou os dados de entrada (pluviogramas) sem que seja necessário processar novamente a primeira parte do sistema.

5.2 Trabalhos Futuros

Os métodos desenvolvidos neste trabalho e os resultados obtidos abrem espaço para diversos outros trabalhos, tanto relacionados a melhorias nos programas já desenvolvidos quanto à sua utilização. Com relação às melhorias, pode-se destacar:

- adaptação do programa utilizando a linguagem de programação visual Delphi, que possui programação básica em Pascal, mas permite implementação de facilidades para interface com o usuário e acesso a diferentes bases de dados.
- ampliação do programa, transformando-o em um Sistema de Informação Geográfica. Para isto, deve-se facilitar a entrada dos dados e permitir acesso às informações de saída em forma gráfica.
- modificação do esquema para processamento unidimensional nos canais, aumentando a discretização e interpolando valores na entrada lateral para atingir valores mais precisos na saída do canal.
- introdução de características de solo e cálculos hidrológicos, como infiltração e evaporação.

Outros trabalhos que podem ser desenvolvidos a partir deste são relacionados a testes de campo, utilizando-se dados reais obtidos de forma digital e verificando os resultados obtidos para bacias hidrográficas conhecidas. Deste modo, é possível comparar esses resultados aos obtidos através de outros métodos com relação aos valores reais.

Dentro da mesma linha de trabalhos, é possível a utilização dos métodos e programas desenvolvidos em bacias sobre as quais não se conhece o comportamento hidrológico, ou até mesmo regiões de difícil acesso, permitindo controle, simulação e previsão de cheias.

Acreditamos ainda que, em futuro próximo, será possível adequar o comportamento hidrológico de cada região às necessidades de todas as espécies que com ela interagem. Para que esse objetivo possa ser atingido, é fundamental uma ferramenta que possibilite simular o reflexo nas características hidrológicas causadas por alterações na superfície, evitando os efeitos colaterais decorrentes de projetos envolvendo modificações no relevo ou tipo de solo.

Referências Bibliográficas

- ALTY & COOMBS (1984) Alty, J.C.; Coombs, M.J. ; *Expert Systems: Concepts and Examples.*; NCC PUBLICATIONS, 1984.
- AZEVEDO & FONTANE (1993) Azevedo, L.G.; Fontane, D.G.; Porto, R.L. *Expert System for the Calibration of SMAP.* WATER INTERNATIONAL, VOL. 18, NRO 2, 1993.
- BAND (1989) Band, L.E. *Terrain-Based Watershed Information System.* HYDROLOGICAL PROCESSES, VOL. 3, Nº 2, PP 151-162, 1989.
- BHASKAR et al. (1993) Bhaskar, N.R.; James W.P.; Devulapalli, R.S. *Hydrologic Parameter Estimation Using Geographic Information System.* JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT, VOL. 118, NRO 5, PP 492-512, 1993.
- BLANDFORD & ORMSBEE (1993) Blandford, G.E.; Ormsbee, L.E. *A Diffusion Wave Finite Element Model for Channel Networks.* JOURNAL OF HYDROLOGY, 142, PP 99-120, 1993.
- BREBBIA & FERRANTE (1975) Brebbia, C.A.; Ferrante A.J. *The Finite Element Technique.* ED. DA UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 1975.
- BURROUGH (1987) Burrough, P.A. *Principles of Geografic Information Systems for Land Resources Assessment.* CLARENDON PRESS, OXFORD, 1987.
- CARNOTA & TEZKIEWIZ (1988) Carnota, R.J.; Tezkiewiz, A.D. *Sistemas Expertos y Representación del Conocimiento.* ED. EBAI, 1988.
- CASTIÑEIRA 1990) Castiñeira, M.I. *Aprendizado de Máquina por Exemplos: Família TDIDT.* TESE DE MESTRADO, ICMSC-USP, 1990.

- CHAIRAT & DELLEUR (1993) Chairat, S.; Delleur, J.W. *Integrating a Physically Based Hydrological Model with GRASS*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 143-150, 1993.
- CONTE (1965) Conte, S.D. *Elementary Numerical Analysis*. MCGRAW-HILL, INC. - 1965.
- COUGER et al. (1992) Couger, G.; Wilson, B.N.; Rice, C.T. *Determination of Drainage Networks from Plot-size and Basin-size Areas*. APPLIED ENGINEERING IN AGRICULTURE VOL. 8, N° 2, PP 185-189, 1992.
- COSTA-CABRAL & BURGESS (1994) Costa-Cabral, M.C.; Burgess, S.J. *Digital Elevation Model Networks (DEMON): A Model of Flow over Hillslopes for Computation of Contributing and Dispersal Areas*. WATER RESOURCES RESEARCH, VOL. 30, N° 6, PP 1681-1692, 1994.
- DEVANTIER & FELDMAN (1993) DeVantier, B.A.; Feldman, A.D. *Review of GIS Applications in Hydrologic Modeling*. JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT, VOL. 119, NRO 2, PP 246-261, 1993.
- DJOCKIC & MAIDMENT (1991) Djockic, D.; Maidment, D.R. *Terrain Analysis for Urban Stormwater Modelling*. HYDROLOGICAL PROCESSES, VOL. 5, N° 1, PP 115-124, 1991.
- DROESEN & GEELLEN (1993) Droesen, W.J.; Geelen L.H.W.T. *Application of Fuzzy Sets in Ecohydrological Expert Systems*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 3-11, 1993.
- ELGY et al. (1993) Elgy, J.; Maksimovic, C.; Prodanovic, D. *Matching Standard GIS Packages with Urban Storm Drainage Simulation Software*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 151-160, 1993.

- FAIRFIELD & LEYMARIE (1991) Fairfield, J.; Leymarie P. *Drainage Network From Grid Digital Elevation Models*. WATER RESOURCES RESEARCH, VOL. 27, NRO 5, PP 709-717, 1991.
- FEDRA (1993) Fedra, K. *Models, GIS and Expert Systems: Integrated Water Resources Models*. HYDROGIS 93, IAHS PUBLICATION, N° 21, PP 297-308, 1993.
- FOWLER et al. (1979) Fowler, R.J.; Little, J.J. *Automatic Extraction of Irregular Network Digital Terrain Models*. COMPUT. GRAPHICS SIGGRAPH-ACM, N° 13, PP 199-207, 1979.
- FORD (1987) Ford, N. *How Machines Think*. JOHN WILEY AND SONS, 1987.
- FORTIN & BERNIER (1991) Fortin, J.P., Bernier, M. *Processing of Remotely Sensed Data to Derive Useful Input Data for the Hydrotel Hydrological Model*. INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), VOL. 1, PP 63-65, 1991.
- FRANCO (1991) Franco, J.L. *Sobre Avaliação Parcial de Meta-Interpretadores e Geração de Núcleos Específicos de Sistemas Especialistas*. DISSERTAÇÃO DE MESTRADO, ICMSC - USP, 1991.
- FRANCO & MONARD (1992a) Franco, J.L.; Monard, M.C. *Avaliação Parcial de Programas Prolog e sua Utilização na Meta-Programação*. SÉRIE CIÊNCIA & TECNOLOGIA, ILTC, VOL. 1, NRO 13, PP 1-35, 1992.
- FRANCO & MONARD (1992b) Franco, J.L.; Monard, M.C. *Meta-Interpretadores Prolog*. NOTAS DIDÁTICAS DO ICMSC-USP, NRO 6, 1992.
- FRANCO & MONARD (1993) Franco, J.L.; Monard, M.C. *Programação Lógica e Fluxo em Redes*. NOTAS DO ICMSC-USP, NRO 129, 1993.
- FRYSINGER et. al (1993) Frysinger, S.P.; Thomas, R.P.; Parsons, A.M. *Hydrological Modelling and GIS: the Sandia Environmental Decision Support Systems*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 45-50, 1993.

- FÜRST et. al (1993) Fürst, J.; Girstmair, G.; Nachtnebel, H.P. *Application of GIS in Decision Support Systems for Groundwater*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 13-21, 1993.
- GRAYSON et al.(1992a) Grayson, R.B.; Moore, I.D.; McMahon, T.A. *Physically based hydrologic modeling 1, A Terrain-Based Model for Investigative Purposes*. WATER RESOURCES RESEARCH, VOL. 28, NRO 10, PP 2639-2658, 1992.
- GRAYSON et al.(1992b) Grayson, R.B.; Moore, I.D.; McMahon, T.A. *Physically based hydrologic modeling 2, Is the concept realistic?* WATER RESOURCES RESEARCH, VOL. 28, NRO 10, PP 2659-2666, 1992.
- GOODRICH et. al (1991) Goodrich, D.C.; Woolhiser, D.A.; Keefer, T.O. *Kinematic Routing Using Finite Elements on a Triangular Irregular Network*. WATER RESOURCES RESEARCH, VOL. 27, NRO 6, PP 995-1003, 1991.
- GOODRICH & WOOLHISER (1994) Goodrich, D.C.; Woolhiser, D.A. *Comment on Physically based hydrologic modeling, 1, A Terrain-Based Model for Investigative Purposes*. WATER RESOURCES RESEARCH, VOL. 30, NRO 3, PP 845-847, 1994.
- HAYES-ROTH et. al (1983) Hayes-Roth, F.; Waterman, D.A.; Lenat, D.B. *Building Expert Systems*. ADDISON-WESLEY PUBLISHING COMPANY, 1983.
- HSU et al. (1995) Hsu, K.; Gupta, H.V.; Sorooshian, S. *Artificial Neural Network Modeling of the Rainfall-Runoff Process*. WATER RESOURCES RESEARCH, VOL. 31, N° 10, PP 2517-2530, 1995.
- HUANG & SONG (1985) Huang, J.; Song, C.C.S. *Stability of Dynamic Flood Routing Schemes*. JOURNAL OF HYDRAULIC ENGINEERING, VOL. 111, NRO 12, PP 1497-1506, 1985.
- ICHOKU et. al (1996) Ichoku, C.; Karnieli, A.; Verchovsky, I. *Application of Fractal Techniques to the Comparative Evaluation of Two Methods of Extracting Channel Networks from Digital Elevation Models*. WATER RESOURCES RESEARCH, VOL. 32, NRO 2, PP 389-399, 1996.

- JACKSON (1983) Jackson, P. *Introduction to Expert Systems*. ADDISON-WESLEY PUBLISHING COMPANY, 1983.
- JAYAWARDENA & WHITE (1977) Jayawardena, A.W.; White, J.K. *A Finite Element Distributed Catchment Model I - Analytical Basis*. JOURNAL OF HYDROLOGY, VOL. 34, PP 269- 286, 1977.
- JAYAWARDENA & WHITE (1979) Jayawardena, A.W.; White, J.K. *A Finite Element Distributed Catchment Model I - Application to Real Catchments*. JOURNAL OF HYDROLOGY, VOL. 42, PP 231- 249, 1979.
- JENSON & DOMINGUE (1988) Jenson, S.K.; Domingue, J.O. *Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis*. PHOTOGRAMMETRIC ENGINEERING AND REMOTE SENSING, VOL. 54, N° 11, PP 1593-1600, 1988.
- JOHNSON et al. (1988) Johnson, L.E.; Dallmann, J.; Toms, E.; Huffman C. *Geographic Information Systems for Hydrologic Modeling*. COMPUTER DECISION SUPPORT SYSTEMS WATER MANAGE: PROCEEDINGS OF THE 3RD WATER RESOURCES OPERATIONS MANAGEMENT WORKSHOP, PP 736-749, 1988.
- JONES & WRIGHT (1990) Jones, N.L.; Wright, S.C. *Watershed Delineation With Triangle-Based Terrain Models*. JOURNAL OF HYDRAULIC ENGINEERING, VOL. 116, NRO 10, PP 1232-1251, 1990.
- LA BARBERA et al. (1993) La Barbera P.; Lanza, L.; Siccardi, F. *Hydrologically Oriented GIS and Application to Rainfall-Runoff Distributed Modelling: Case Study of the Arno Basin*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 171-179,1993.
- LAM & SWAYNE (1993) Lam, D.C.L.; Swayne, D.A. *An Expert System Approach of Integrating Hydrological Databases, Models and GIS: application of the RAISON System*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 23-33,1993.

- LEE & SCHACHTER (1980) Lee, D.T.; Schachter, B.J. *Two Algorithms for Constructing a Delaunay Triangulation*. INTERNATIONAL JOURNAL OF COMPUTER AND INFORMATION SCIENCES, VOL. 9, N° 3, 1980.
- LEIPNIK et al. (1993) Leipnik, M.R.; Kemp, K.K.; Loaiciga, H.A. *Implementation of GIS for Water Resources Planning and Management*. JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT, VOL. 119, NRO 2, PP 184-205, 1993.
- MAGALHAES (1989) Magalhães, P.C. *Hidrologia Superficial*. ENGENHARIA HIDROLÓGICA, ABRH, ED. UFRJ, PP 201-289, 1989.
- MAIDMENT (1993) Maidment, D.R. *Developing a Spatially Distributed Unit Hydrograph by Using GIS*. HIDROGIS 93, IAHS PUBLICATION, NRO 21, PP 181-192, 1993.
- MARTZ & GARBECHT (1992) Martz, L.W.; Garbecht, J. *Numerical Definition of Drainage Network and Subcatchment Areas from Digital Elevation Models*. COMPUTERS & GEOSCIENCES VOL. 18, N° 6, PP 747-761, 1992.
- MCKINNEY & MAIDMENT (1993) McKinney, D.C.; Maidment, D.R. *Expert Geographic Information System for Texas Water Planning*. JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT VOL. 119, NRO 2, PP 170-183, 1993.
- MEIJERINK et al. (1993) Meijerink, A.M.J.; Mannaerts, C.M.; Brouwer, H.A.; Valenzuela, C.R. *Application of ILWIS to Decision Support in Watershed Management*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 35-44, 1993.
- MERKEL & SPERLING (1993) Merkel, B.; Sperling, B. *Raster Based Modelling of Watershed and Flow Accumulation*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 193-199, 1993.
- MITCHELL & WAIT (1977) Mitchel, A.R.; Wait, R. *The Finite Element Method in Partial Differential Equations*. JOHN WILEY AND SONS, INC, 1977.

- MONARD (1986) Monard, M.C. *Um Sistema de Representação de Conhecimento e Raciocínio por Default*. TESE DE LIVRE DOCÊNCIA, ICMSC-USP, 1986.
- MONARD & HEBIHARA (1987) Monard, M.C.; Hebihara, S.M. *Um Sistema que Utiliza Raciocínio por Default*. ANAIS IV SBIA, PP 143-156, 1987.
- MONARD & PRADO (1989) Monard, M.C.; Prado A.H.A. *Uso de Incertezas em Sistemas Baseados em Conhecimento*. ILTC, 1989.
- MONARD & RODRIGUES (1989) Monard, M.C.; Rodrigues S.R. *Desenvolvimento de Sistemas Especialistas em Prolog*. ILTC, 1989.
- MONARD & FRANCO (1990) Monard, M.C.; Franco, J.L. *Uso de Programação Lógica no Desenvolvimento de Compiladores*. RELATÓRIO TÉCNICO, ICMSC-USP, NRO 8, 1990.
- MOORE & GRAYSON (1991) Moore, I.D.; Grayson, R.B. *Terrain-Based Catchment Partitioning and Runoff Prediction Using Vector Elevation Data*. WATER RESOURCES RESEARCH, VOL. 27, N° 6, PP 1177-1191, 1991.
- MOORE et al. (1993) Moore, I.D.; Gallant, J.C.; Guerra, L.; Kalma, J.D. *Modelling the Spatial Variability of Hydrological Process Using GIS*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 161-169, 1993.
- MOREL-SEYTOUX et al (1993) Morel-Seytoux H.J.; Fahmy, H.; Lamagat, J.P. *A Composite Hydraulic and Statistical Flow-Routing Method*. WATER RESOURCES RESEARCH, VOL. 29, NRO 2, PP 413-418, 1993.
- NELSON et al (1994) Nelson, E.J.; Jones, N.L.; Miller, A.W. *Algorithm for Precise Drainage-Basin Delineation*. JOURNAL OF HYDRAULIC ENGINEERING, VOL. 120, N° 3, 1994.
- NILSSON (1982) Nilsson, N.J. *Principles of Artificial Intelligence*. SPRING-VERLAG, 1982.

- PALACIOS-VELEZ & CUEVAS-RENAUD (1986) Palacios-Vélez, O.L.; Cuevas-Renaud, B. *Automated River Course, Ridge and Basin Delineation from Digital Elevation Data*. JOURNAL OF HYDROLOGY, VOL. 86, PP 299-314, 1986.
- PALACIOS-VELEZ & CUEVAS-RENAUD (1992) Palacios-Vélez, O.L.; Cuevas-Renaud, B. *SHIFT: A Distributed Runoff Model Using Irregular Triangular Facets*. JOURNAL OF HYDROLOGY, VOL. 134, PP 35-55, 1992.
- PESSOA (1991) Pessoa, M.L. *RADMAX: an Intelligent Computer Package for Dam Design and Dam Safety using Radar Observations*. REVISTA DE INFORMÁTICA - TEÓRICA E APLICADA, VOL. 1, NRO 3, 1991.
- PRICE (1978) Price, R.K. *Comparison of Four Numerical Methods for Flood Routing*. JOURNAL OF THE HYDRAULICS DIVISION, VOL. 100, NRO HY7, PP 879-899, 1978.
- QUEIROZ (1991) Queiroz, O.M. *Propagação de Cheias: Aplicação do Modelo de Onda Cinemática ao Ribeirão do Onça - SP*. DISSERTAÇÃO DE MESTRADO - EESC - USP - DEZEMBRO, 1991.
- QIAN et al. (1990) Qian, J.; Ehrich, R.W.; Campbell, J.B. *DNESYS - An Expert System for Automatic Extraction of Drainage Networks from Digital Elevation Data*. IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING VOL. 28, N° 1, PP 29-45, 1990.
- RIEGER (1993) Rieger, W. *Hydrological Terrain Features Derived from a Pyramid Raster Structure*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 201-210, 1993.
- ROBINSON et. al (1995) Robinson, J.S.; Murugesu, S.; Snell, J.D. *On the Relative Roles of Hillslope Processes, Channel Routing, and Network Geomorphology in the Hydrologic Response of Natural Catchments*. WATER RESOURCES RESEARCH, VOL. 31, NRO 12, PP 3089-3101, 1995.

- RODRIGUES (1990) Rodrigues, S.R. *Um Ambiente para Auxiliar a Construção de Núcleos de Sistemas Especialistas*. TESE DE MESTRADO, ICMSC-USP, 1990.
- RODRIGUES & MONARD (1990) Rodrigues, S.R.; Monard, M.C. *Uso de Meta-Interpretadores no Desenvolvimento de Núcleos de Sistemas Especialistas*. ILTC, 1990.
- ROSS et al. (1979) Ross, B.B.; Contractor, D.N.; Shanholtz, V.O. *A Finite Element Model of Overland and Channel Flow for Assessing the Hydrologic Impact of Land-Use Change*. JOURNAL OF HYDROLOGY, VOL. 41, PP 11-30, 1979.
- ROSS & TARA (1993) Ross, M.A.; Tara, P.D. *Integrated Hydrologic Modeling with Geographic Information Systems*. JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT, VOL. 119, NRO 2, PP 129-140, 1993.
- RULAND et al. (1993) Ruland, P.; Arnold, U.; Rouvé, G. *An Integrated Information System for Environmental Hydraulics Using Smallworld GIS*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 51-60, 1993.
- SAWAKI (1991) Sawaki, J. *Um Sistema Baseado em Conhecimento para Apoio aos Usuários da Biblioteca NAG*. TESE DE MESTRADO, ICMSC-USP, 1991.
- SCHULTZ (1993) Schultz, G.A. *Application of GIS and Remote Sensing in Hydrology*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 127-140, 1993.
- SEGERLIND (1984) Segerlind, L.J. *Applied Finite Element Analysis*. JOHN WILEY AND SONS, INC, 1984.
- SIMONOVIC (1993) Simonovic, S.P. *Flood Control Management by Integrating GIS with Expert Systems: Winnipeg City Case Study*. HYDROGIS 93, IAHS PUBLICATION, NRO 21, PP 61-72, 1993.

- SIVALOGANATHAN (1978) Sivaloganathan, K. *Flood Routing by Characteristic Methods*. JOURNAL OF THE HYDRAULICS DIVISION, VOL. 104, NRO HY7, PP 1027-1044, 1978.
- SMITH et al. (1990) Smith, T.R.; Zhan, C.; Gao, P. *Knowledge-Based, Two-Steps Procedure for Extracting Channel Networks from Noisy DEM Data*. COMPUTERS & GEOSCIENCES VOL. 16, N° 6, PP 777-786, 1990.
- SMITH et al. (1994) Smith, R.E.; Goodrich, D.R.; Woolhiser, D.A.; Simanton, J.R. *Comment On Physically based hydrologic modeling, 2, Is the concept realistic?* WATER RESOURCES RESEARCH, VOL. 30, N° 3, PP 851-854, 1994.
- STRELKOFF (1970) Strelkoff, T. *Numerical Solution of Saint-Venant Equations*. JOURNAL OF THE HYDRAULICS DIVISION, VOL. 96, NRO HY1, PP 223-252, 1970.
- STREETER (1966) Streeter, V.L. *Fluid Mechanics*. MCGRAW-HILL BOOK COMPANY, NY, 1966.
- TARBOTON (1997) Tarboton, D.G. *A New Method for the Determination of Flow Directions and Upslope Areas in Grid Digital Elevation Models*. WATER RESOURCES RESEARCH, VOL. 33, N° 2, PP 309-319, 1997.
- TAYFUR et al. (1993) Tayfur, G.; Kavvas, M.L.; Govindaraju, R.S.; Storm, D.E. *Aplicability of St. Venant Equations for Two-Dimensional Overland Flows Over Rough Infiltrating Surfaces*. JOURNAL OF HYDRAULIC ENGINEERING, VOL. 119, NRO 1, PP 51-63, 1993.
- TAYLOR et al. (1974) Taylor C.; Al-Mashidani, G.; Davis, J.M. *A Finite Element Approach to Watershed Runoff*. JOURNAL OF HYDROLOGY, VOL. 21, PP 231-246, 1974.
- TEIXEIRA et al. (1992) Teixeira, A.L.A.; Moretti, E.; Christofolletti, A. *Introdução aos Sistemas de Informação Geográfica*. EDIÇÃO DO AUTOR, RIO CLARO, 1992.
- TUNG (1985) Tung, Y. *River Flood Routing by Nonlinear Muskingum Method*. JOURNAL OF HYDRAULIC ENGINEERING, VOL. 111, NRO 12, PP 1447-1460, 1985.

- VANBLARGAN et al. (1990) VanBlargan, E.J.; Ragan, R.M., Schaake, J.C. *Hydrologic Geographic Information System*. TRANSPORTATION RESEARCH RECORD, N° 1261, PP 44-51, 1990.
- VEMURI et al. (1981) Vemuri, V.; Karplus, W. *Digital Computer Treatment of Partial Differential Equations*. PRENTICE-HALL, INC, ENGLEWOODS CLIFFS, 1981.
- VIEUX et al. (1990) Vieux, B.E.; Bralts, V.F.; Segerlind, L.J.; Wallace R.B. *Finite Element Watershed Modeling: One-Dimensional Elements*. JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT VOL. 116, NRO 6, PP 803-819, 1990.
- WAIT & MITCHELL (1985) Wait, R.; Mitchell, A.R. *Finite Element Analysis and Applications*. JOHN WILEY AND SONS, INC, 1985.
- WALSH (1993) Walsh, M.R. *Towards Spatial Decision Support Systems in Water Resources*. JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT VOL. 119, NRO 2, PP 158-183, 1993.
- WATERMAN (1986) Waterman, D.A. *A Guide to Expert Systems*. ADDISON-WESLEY, 1986.
- WATSON & PHILIP (1984a) Watson, D.F.; Philip, G.M. *Systematic Triangulations*. COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, N° 26, PP 217-223, 1984.
- WATSON & PHILIP (1984b) Watson, D.F.; Philip, G.M. *Triangle Based Interpolation*. MATHEMATICAL GEOLOGY, VOL. 16, N° 8, 1984.
- WINSTON (1984) Winston, P.H. *Artificial Intelligence*. ADDISON-WESLEY, 1984.
- WOOLHISER & LIGGETT (1967) Woolhiser, D.A.; Liggett, J.A. *Unsteady, One-Dimensional Flow over a Plane --- the Rising Hydrograph*. WATER RESOURCES RESEARCH, VOL. 3, NRO 3, PP 753-770, 1967.
- ZEHANA et al. (1991) Zehana, M.; Desachy, J.; Zahzah, E.H. *DEM Features for Remote Sensing Image Analysis by an Expert System: Stream Networks and Real Distances Computation*. INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), VOL. 3, PP1861-1864, 1991.
- ZHANG & CUNDY (1989) Zhang, W.; Cundy, T.W. *Modeling of Two-Dimensional Overland Flow*. WATER RESOURCES RESEARCH, VOL.25, NRO 9, 1989.

APÊNDICE A - Listagem dos Programas

Aqui serão apresentados os programas responsáveis pelo processamento bidimensional, tanto na fase de identificação do fluxo, que é feito utilizando programação lógica, quanto na fase de quantificação, onde se utiliza programação procedural.

Criação da Base de Dados Prolog

O programa abaixo lê os arquivos contendo as coordenadas dos vértices, os contornos da região e a definição dos triângulos e armazena-os em uma base de dados Prolog. Além de transformar os dados, ele identifica os triângulos adjacentes e armazena essa informação. As principais partes do programa são descritas a seguir:

- **le_dados:** lê o arquivo com as coordenadas dos vértices e gera as relações no formato coordenadas(NroVert, ValorX, ValorY, ValorZ).
- **le_triang:** lê o arquivo com a definição dos triângulos em função dos vértices, identifica a melhor forma de posicionar esses vértices e gera relações no formato triangulo(NroTri,vert(V1,V2,V3)).
- **le_contorno:** lê o arquivo com os contornos e armazena-os em relações na forma contorno(NroLado,P1,P2).
- **pega_adjacentes:** seleciona os triângulos um a um e identifica os triângulos adjacentes, criando uma relação da forma adjacentes(NroTri ,adj(T1,T2,T3)).

As relações coordenadas/4, triangulo/2 e adjacentes/2 são armazenadas no arquivo **dados.ari**.

```

cria_arqs(Nome) :-
  abre_arquivos(Nome,Arq1,Arq2,Arq3), le_dados(Arq1),
  le_triangu(Arq2),
  le_contorno(Arq3),
  findall(N,triangulo(N,_,L),
  pega_adjacentes(L),
  file_list('dados.ari',[coordenadas/4,
  triangulo/2,adjacentes/2]),!).

abre_arquivos(Nome,Arq1,Arq2,Arq3) :-
  atom_string(Nome,Str),
  string_length(Str,Comp),
  concat(Str,$.vrt$,S1),
  open(Arq1,S1,r),
  concat(Str,$.tri$,S2),
  open(Arq2,S2,r),
  concat(Str,$.con$,S3),
  open(Arq3,S3,r).

le_dados(Arq) :-
  read_line(Arq,Str),
  substring(Str,0,2,Sn),
  int_text(N,Sn),!,
  le_dados1(N,Str,Arq).

le_dados1(0,_,Arq) :- close(Arq),!.
le_dados1(N,Str,Arq) :-
  substring(Str,2,6,Sx),
  substring(Str,8,6,Sy),
  substring(Str,14,6,Sz),
  float_text(X,Sx,_),
  float_text(Y,Sy,_),
  float_text(Z,Sz,_),
  assert(coordenadas(N,X,Y,Z)),!,
  le_dados(Arq).

le_triangu(Arq) :-
  read_line(Arq,Str),
  substring(Str,0,2,Sn),
  int_text(N,Sn),!,
  le_triangu1(N,Str,Arq).

le_triangu1(0,_,Arq) :- close(Arq),!.
le_triangu1(N,Str,Arq) :-
  divide_linha(Str,A,B,C),
  acerta_pontos(A,B,C,Vert),
  assert(triangulo(N,Vert)),!,
  le_triangu(Arq).

divide_linha(Str,A,B,C) :-
  substring(Str,3,2,Si),
  substring(Str,6,2,Sj),
  substring(Str,9,2,Sk),
  int_text(A,Si),
  int_text(B,Sj),
  int_text(C,Sk).

acerta_pontos(A,B,C,Vert) :-
  coordenadas(A,Xa,Ya,_),
  coordenadas(B,Xb,Yb,_),
  coordenadas(C,Xc,Yc,_),
  acerta_vertices((Xa,Ya),(Xb,Yb),(Xc,Yc),Trig),!,
  posiciona_vertices(A,B,C,Trig,Vert).

acerta_vertices(A,B,C,jki) :- menor(A,B,C).
acerta_vertices(A,B,C,ijk) :- menor(B,A,C).
acerta_vertices(A,B,C,kij) :- menor(C,A,B).
acerta_vertices(A,B,C,ijk) :- maior(A,B,C).
acerta_vertices(A,B,C,kij) :- maior(B,A,C).
acerta_vertices(A,B,C,jki) :- maior(C,A,B).

acerta_vertices(A,B,C,ijk) :-
  maior_y(A,B,C),
  menor_x(A,B,C).
acerta_vertices(A,B,C,kij) :-
  maior_y(B,C,A),
  menor_x(B,C,A).
acerta_vertices(A,B,C,jki) :-
  maior_y(C,A,B),
  menor_x(C,A,B).

acerta_vertices(A,B,C,ijk) :-
  maior_y(A,B,C).
acerta_vertices(A,B,C,kij) :-
  maior_y(B,C,A).
acerta_vertices(A,B,C,jki) :-
  maior_y(C,A,B).

menor((X1,Y1),(X2,Y2),(X3,Y3)) :-
  X1 =< X2, X1 =< X3,
  Y1 =< Y2, Y1 =< Y3.

maior((X1,Y1),(X2,Y2),(X3,Y3)) :-
  X1 >= X2, X1 >= X3,
  Y1 >= Y2, Y1 >= Y3.

maior_y((X1,Y1),(X2,Y2),(X3,Y3)) :-
  Y1 >= Y2, Y1 >= Y3.
menor_x((X1,Y1),(X2,Y2),(X3,Y3)) :-
  X1 =< X2, X1 =< X3.

posiciona_vertices(A,B,C,ijk,vert(A,B,C)).
posiciona_vertices(A,B,C,jki,vert(C,A,B)).
posiciona_vertices(A,B,C,kij,vert(B,C,A)).

pega_adjacentes({}).
pega_adjacentes([N|L]) :-
  pega_adj(N),
  pega_adjacentes(L).

pega_adj(N) :-
  triangulo(N,vert(I,J,K)),
  vizinho(N,I,J,V1),
  vizinho(N,J,K,V2),
  vizinho(N,K,I,V3),
  assert(adjacentes(N,adj(V1,V2,V3))).

vizinho(_,P1,P2,V) :- el_contorno(P1,P2,V),!.
vizinho(N,P1,P2,V) :- el_triangu(N,P1,P2,V).

el_triangu(N,P1,P2,V) :-
  triangulo(V,vert(I,J,K)),
  N \= V,
  L = [I,J,K],
  pertence(P1,L),
  pertence(P2,L).

el_contorno(P1,P2,I(P1,P2)) :- contorno(_,P1,P2),P1 < P2.
el_contorno(P1,P2,I(P2,P1)) :- contorno(_,P1,P2).
el_contorno(P1,P2,I(P1,P2)) :- contorno(_P2,P1),P1 < P2.
el_contorno(P1,P2,I(P2,P1)) :- contorno(_P2,P1).

le_contorno(Arq) :-
  read_line(Arq,Str),
  substring(Str,0,2,Sn),
  int_text(N,Sn),!,
  le_contorno1(N,Str,Arq).

le_contorno1(0,_,Arq) :- close(Arq),!.
le_contorno1(N,Str,Arq) :-
  substring(Str,3,2,S1),
  substring(Str,6,2,S2),
  int_text(P1,S1),
  int_text(P2,S2),
  assert(contorno(N,P1,P2)),!,
  le_contorno(Arq).

```

Identificação dos Fluxos

O programa a seguir utiliza a base de dados **dados.ari**, gerada pelo programa anterior, e calcula a direção do fluxo. As principais partes do programa são:

- **calcula_triangulo**: a partir de um triângulo selecionado, calcula-se os ângulos dos lados e do gradiente com relação ao eixo das abcissas. Em seguida, a partir desses 4 ângulos, identifica o comportamento do fluxo nesse triângulo. A partir dessa informação, gera relações da forma fluxo(NroT, T1, P), onde NroT é o triângulo em análise, T1 é um triângulo vizinho que recebe fluxo e P é a porcentagem do fluxo que esse triângulo recebe.
- **calcula_interseccoes**: a partir da informação sobre a comunicação entre os triângulos, identifica os lados que são condutores de fluxo (canais).
- **processa_ordem**: identifica a ordem de processamento nos triângulos. Isto é feito selecionando os triângulos que não recebem fluxo de nenhum outro triângulo. Os triângulos selecionados são retirados da lista de entrada dos outros triângulos. Quando a lista se esvazia, o triângulo já pode ser selecionado, e o processo se repete até que todos os triângulos sejam selecionados.
- **identifica_lados**: numera todos os lados, tanto do contorno quanto internos, e armazena-os na relação lado(NroLado, P1, P2).

Dois arquivos são criados:

- **fluxos.ari**: armazena as relações fluxo/3 e condutor/2.
- **comunica.ari**: armazena as relações ordem/1, envia_lado/2, recebe_lado/2 e lado/3.

```

calcula_fluxos :-
  findall(N,triangulo(N,_),L),
  calcula_todos(L),
  calcula_intersecoes,
  aglutina_iguais,
  seleciona_fluxo_io(L),
  indica_lados_io(L),
  processa_ordem,
  identifica_lados,
  file_list('fluxos',[fluxo/3,condutor/2]),
  file_list('comunica',[envia_lado/2,recebe_lado/2,
  ordem/1,lado/3]),!.

calcula_todos([N|_]) :-
  calcula_triangulo(N),!,
  calcula_todos(L).
calcula_todos([]).

calcula_triangulo(N) :-
  triangulo(N,Vert),
  calcula_pontos_interligados(Vert),!,
  calcula_grad(Vert,Alfa),
  calcula_phi(Vert,F1,F2,F3),
  assert( angulos(N,phi(F1,F2,F3),alfa(Alfa)) ),
  calcula_direcoes(N,Alfa,F1,F2,F3,Direcoes),!,
  armazena_fluxos(N,Direcoes).

calcula_grad(vert(I,J,K),Alfa) :-
  coordenadas(I,Xi,Yi,Zi),
  coordenadas(J,Xj,Yj,Zj),
  coordenadas(K,Xk,Yk,Zk),
  AA is Xi*Yj+Xj*Yk+Xk*Yi-Yi*Xj-Yj*Xk-Yk*Xi,
  Fa is (Zi*(Yj - Yk) + Zj*(Yk - Yi) + Zk*(Yi - Yj))/AA,
  Fb is (Zi*(Xk - Xj) + Zj*(Xi - Xk) + Zk*(Xj - Xi))/AA,
  calcula_alfa(Fa,Fb,Alfa).

calcula_alfa(Fa,Fb,Alfa) :-
  Fa =:= 0,!,
  B is Fb/abs(Fb),
  Alfa is (B+2)*90.
calcula_alfa(Fa,Fb,Alfa) :-
  Fb =:= 0,!,
  A is Fa/abs(Fa),
  Alfa is (A+1)*90.
calcula_alfa(Fa,Fb,Alfa) :-
  A is Fa/abs(Fa),
  B is Fb/abs(Fb),
  arctg(Fb/Fa,Angulo),
  Alfa is Angulo + (B - A*B + 2)*90.

calcula_phi(vert(I,J,K),F1,F2,F3) :-
  calcula_angulo(J,K,FF1),
  calcula_angulo(J,I,FF2),
  calcula_angulo(K,I,FF3),
  acerta_angulo(FF2,F2),
  acerta_angulo1(F2,FF1.F1),
  acerta_angulo(FF3,FF3),
  acerta_angulo2(FF3,F2,F3).

calcula_angulo(P1,P2,Angulo) :-
  coordenadas(P1,X1,Y1,Z1),
  coordenadas(P2,X2,Y2,Z2),
  calc_ang(X1,X2,Y1,Y2,Angulo).

calc_ang(X,X_,_90) :- !.
calc_ang(X1,X2,Y1,Y2,Angulo) :-
  M is (Y2 - Y1)/(X2 - X1),
  arctg(M,Angulo).

acerta_angulo(F,F) :- F >= 0,!.
acerta_angulo(FF,F) :- F is FF + 180.
acerta_angulo1(F2,FF,F) :- FF < 0, F is FF + 180, F <
F2,!.
acerta_angulo1(_F,F).
acerta_angulo2(F3,F2,F3) :- F2 < F3,!.

```

```

acerta_angulo2(FF3,_F3) :- F3 is FF3 + 180.

calcula_direcoes(_Alfa,F1,_F2,flx(c,l,@)) :- % Caso 2
Alfa =:= F1.
calcula_direcoes(_Alfa,_F2,flx(c,l,@)) :- % Caso 4
Alfa =:= F2.
calcula_direcoes(_Alfa,_F3,flx(a,l,@)) :- % Caso 6
Alfa =:= F3.

calcula_direcoes(_Alfa,F1,_F2,flx(a,l,@)) :- % Caso 8
Alfa =:= F1 + 180; Alfa =:= F1 - 180.
calcula_direcoes(_Alfa,_F2,flx(b,l,@)) :- % Caso 10
Alfa =:= F2 + 180; Alfa =:= F2 - 180.
calcula_direcoes(_Alfa,_F3,flx(b,l,@)) :- % Caso 12
Alfa =:= F3 + 180; Alfa =:= F3 - 180.
calcula_direcoes(_Alfa,F1,F2,flx(c,l,@)) :- % Caso 3
Alfa > F1, Alfa < F2.
calcula_direcoes(_Alfa,F1,F2,flx(c,l,@)) :- % Caso 3
Alfa > F1 + 360.

calcula_direcoes(_Alfa,F1,_F3,flx(a,l,@)) :- % Caso 7
Alfa > F3, Alfa < F1 + 180.
calcula_direcoes(_Alfa,_F2,F3,flx(b,l,@)) :- % Caso 11
Alfa > F2 + 180, Alfa < F3 + 180.
calcula_direcoes(N,Alfa,_F2,F3,flx(a,P,c)) :- % Caso 5
Alfa > F2, Alfa < F3,!.
triangulo(N,vert(I,J,K)),
calcula_porcentagem(J,K,I,Alfa,P).
calcula_direcoes(N,Alfa,F1,F2,flx(b,P,a)) :- % Caso 9
Alfa > F1 + 180, Alfa < F2 + 180,!,
triangulo(N,vert(I,J,K)),
calcula_porcentagem(K,I,J,Alfa,P).
calcula_direcoes(N,Alfa,_F2,F3,flx(c,P,b)) :- % Caso 1
triangulo(N,vert(I,J,K)),
calcula_porcentagem(I,J,K,Alfa,P).

calcula_porcentagem(I,J,K,Alfa,P) :-
  coordenadas(I,Xi,Yi,_),
  coordenadas(J,Xj,Yj,_),
  coordenadas(K,Xk,Yk,_),
  Dx is Xj - Xi, Dy is Yj - Yi,
  tg(Alfa,M),
  Xp is ((Yk - Xk*M - Yi)*Dx + Xi*Dy)/(Dy - Dx*M),
  Yp is Yk + (Xp - Xk)*M,
  Aijk is Xi*Yj+Xj*Yk+Xk*Yi-Yi*Xj-Yj*Xk-Yk*Xi,
  Aipk is Xi*Yp+Xp*Yk+Xk*Yi-Yi*Xp-Yp*Xk-Yk*Xi,
  P is Aipk/Aijk.

armazena_fluxos(N,flx(V1,P1,V2)) :-
  adjacentes(N,Adj),
  pega_vizinho(V1,Adj,N1),
  pega_vizinho(V2,Adj,N2),
  armazena(N,N1,N2,P1).

armazena(N,N1,N2,1) :-
  assert(fluxo(N,N1,1)),!.

armazena(N,N1,N2,P1) :-
  assert(fluxo(N,N1,P1)),
  P2 is 1 - P1,
  assert(fluxo(N,N2,P2)).

pega_vizinho(a,adj(N,_),N).
pega_vizinho(b,adj(_N,_),N).
pega_vizinho(c,adj(_N,N),N).
pega_vizinho(@,@,@).
pega_vizinho(N,_N).

calcula_intersecoes :-
  fluxo(N1,N2,P1),
  fluxo(N2,N1,P2),
  calcula_direcao(N1,N2,No1,No2),
  assert(condutor(No1,No2)),
  assert(fluxo(N1,cond(No1,No2),P1)).

```

```

assert(fluxo(N2,cond(No1,No2),P2)),
assert(fluxo_no(N1,ponto(No2),P1)),
assert(fluxo_no(N2,ponto(No2),P2)),
retract(fluxo(N1,N2,P1)),
retract(fluxo(N2,N1,P2)),!,
calcula_interseccoes.

calcula_interseccoes.

calcula_direcao(N1,N2,No1,No2) :-
triangulo(N1,V1),
triangulo(N2,V2),
encontra_par(V1,V2,P1,P2),
calc_dir(P1,P2,No1,No2).

encontra_par(vert(A1,B1,C1),vert(A2,B2,C2),A1,P2) :-
L = [A2,B2,C2],
pertence(A1,L),!,
encontra_par2(B1,C1,L,P2).
encontra_par(vert(_,P1,P2),_,P1,P2).

encontra_par2(P2,_,L,P2) :- pertence(P2,L),!.
encontra_par2(_,P2,_,P2).

calc_dir(P1,P2,P1,P2) :-
coordenadas(P1,_,Z1),
coordenadas(P2,_,Z2),
Z1 > Z2,!.
calc_dir(P1,P2,P2,P1).

calcula_fluxo_dos_nos :-
setof(No, Y^P^fluxo(Y,ponto(No),P),L),
calcula_fluxo_nos(L).

calcula_fluxo_nos([No|L]) :-
lista_adjacentes(No,Adj),
coordenadas(No,X,Y,Z),
nos_abaxo(Z,Adj,Abaxo),!,
calcula_fluxo_nos(L).
calcula_fluxo_nos([]).

nos_abaxo(H,[N|L],[N|R]) :-
coordenadas(N,_,Z),
Z < H,!,
nos_abaxo(H,L,R).
nos_abaxo(H,[N|L],R) :-
nos_abaxo(H,L,R).
nos_abaxo(_,[],[]).

aglutina_iguais :-
fluxo_no(N1,N2,P1),
fluxo_no(N1,N2,P2),
P1 \= P2,!,
P is P1 + P2,
assert(fluxo_no(N1,N2,P)),
retract(fluxo_no(N1,N2,P1)),
retract(fluxo_no(N1,N2,P2)),!,
aglutina_iguais.
aglutina_iguais.

calcula_pontos_interligados(vert(A,B,C)) :-
acrescenta_pontos(A,B,C),
acrescenta_pontos(B,A,C),
acrescenta_pontos(C,A,B).

acrescenta_pontos(A,B,C) :-
pega_lista(A,L),
insere(B,L,L1),
insere(C,L1,L2),
armazena_lista(A,L2).

pega_lista(N,L) :- retract(lista_adjacentes(N,L)),!.
pega_lista(N,[]).

armazena_lista(N,L) :- assert(lista_adjacentes(N,L)).
tg(A,T) :- T is tan(A*3.1415927/180).

arctg(T,A) :- AR is atan(T), A is
round(AR*180/3.1415927.1).

insere(X,[X|L],[X|L]) :-!.
insere(X,[Y|L],[X,Y|L]) :- X < Y,!.
insere(X,[Y|L],[Y|R]) :- insere(X,L,R).
insere(X,[],[X]).

retira(X,[X|L],L) :-!.
retira(X,[Y|L],[Y|R]) :- retira(X,L,R).

seleciona_fluxo_io([Tri|L]) :-
findall(X,fluxo(Tri,X,_)Env),
findall(Y,fluxo(Y,Tri,_)Rec),
assert(envia(Tri,Env)),
assert(recebe(Tri,Rec)),
seleciona_fluxo_io(L).
seleciona_fluxo_io([]).

triangulo_livre(Tri) :-
findall(X,recebe(X,[]),L),
pega_mais_alto(L,Tri).

max(A,B,A) :- A > B,!.
max(A,B,B).

ponto_mais_alto(V1,V2,V3,Z) :-
coordenadas(V1,_,Z1),
coordenadas(V2,_,Z2),
coordenadas(V3,_,Z3),
max(Z1,Z2,Z3), max(Z3,Z3,Z).

pega_mais_alto([Tri],Tri):-!.
pega_mais_alto([T|L],Tri) :-
triangulo(T,vert(V1,V2,V3)),
ponto_mais_alto(V1,V2,V3,Z),
pega_maior(T,Z,L,Tri).

pega_maior(T1,Z1,[T2|L],Tri) :-
triangulo(T2,vert(V1,V2,V3)),
ponto_mais_alto(V1,V2,V3,Z2),
Z2 > Z1,!,
pega_maior(T2,Z2,L,Tri).
pega_maior(T1,Z1,[],Tri) :- pega_maior(T1,Z1,L,Tri).
pega_maior(Tri,_,[],Tri).

processa_ordem :-
processa_ordem(Ord),
assert(ordem(Ord)).

processa_ordem([Condutor|L]) :-
retract(pode_processar(Condutor)),!,
processa_ordem(L).

processa_ordem([Tri|L]) :-
triangulo_livre(Tri),
retract(recebe(Tri,[])),!.
envia(Tri,Lout),
rearranja_io(Tri,Lout),
processa_ordem(L).

processa_ordem([]).

processa_lado(L) :- L = [_,_] ,!.
processa_lado(Cond) :-
retract(logo_processar(Cond)),!,
assert(pode_processar(Cond)).

processa_lado(Cond) :- assert(logo_processar(Cond)).

rearranja_io(Tri,[X|Lout]) :-
(X = l(Cond) : X = cond(Cond)),!.
processa_lado(X),
rearranja_io(Tri,Lout).

rearranja_io(Tri,[X|Lout]) :-
retract(recebe(X,Lin)),
retira(Tri,Lin,NLin),
assert(recebe(X,NLin)).

```

```

rearranja_io(Tri,Lout).
rearranja_io(_,[]).

indica_lados_io([Tri|L]) :-
  envia(Tri,Lout),
  processa_inter(Tri,Lout,LLatO),
  assert(envia_lado(Tri,LLatO)),
  recebe(Tri,Lin),
  processa_inter(Tri,Lin,LLatI),
  assert(recebe_lado(Tri,LLatI)),!,
  indica_lados_io(L).
indica_lados_io([]).

processa_inter(Tri,[X|L],[X|R]) :-
  X = l(_,_),!,
  processa_inter(Tri,L,R).

processa_inter(Tri,[X|L],[Lado|R]) :-
  X = cond(N1,N2),!,
  monta_lado(N1,N2,Lado),
  processa_inter(Tri,L,R).

```

```

processa_inter(Tri,[X|L],[Lado|R]):-
  calcula_direcao(Tri,X.N1.N2),
  monta_lado(N1,N2,Lado),
  processa_inter(Tri,L,R).

processa_inter(Tri,[],[]).

monta_lado(N1,N2,l(N1,N2)) :- N1 < N2,!.
monta_lado(N1,N2,l(N2,N1)).

identifica_lados :-
  ident_lados(1,0).

ident_lados(No.Cont) :-
  retract(lista_adjacentes(No.L)),
  processa_adjacentes(No.L.Cont,Cont1),
  N1 is No + 1,!,
  ident_lados(N1,Cont1).
ident_lados(_,_).

processa_adjacentes(No.[X|L],C.NC) :-
  X > No,
  C1 is C+1,
  assert(lado(C1.No.X)),
  processa_adjacentes(No.L,C1.NC).

processa_adjacentes(No,[_|L],C.NC) :-
  processa_adjacentes(No.L,C.NC).
processa_adjacentes(.,_|C.C).

```

Geração dos arquivos de comunicação

O arquivo **comunica.ari** contém as informações necessárias para o processamento numérico dos triângulos: a ordem do processamento e as interfaces de comunicação entre os triângulos. O programa a seguir transforma essas informações e gera informações em formato texto, de modo que elas possam ser armazenadas em arquivos e, posteriormente, utilizadas pelo programa Pascal responsável pela quantificação do escoamento.

As principais partes do programa são:

- **gera_arq**: cria o arquivo ordem.com
- **armazena_lados**: cria o arquivo lados.com

```

monta_arqpas :-
reconsult(comunica).
create(Arq,'ordem.com'),
ordem(Ord),
gera_arq(Arq,Ord),
create(Ark,'lados.com'),
armazena_lados(Ark).

armazena_lados(Arq) :-
retract(lado(Nro,No1,No2)),
printf(Arq,Nro,3),
printf(Arq,No1,3),
printf(Arq,No2,3),
nl(Arq),!,
armazena_lados(Arq).
armazena_lados(Arq) :- write(Arq,'000'),close(Arq).

gera_arq(Arq,[TriL]) :-
processa_tri(Arq,Tri),!,
gera_arq(Arq,L).
gera_arq(Arq,[]) :- write(Arq,'00'),close(Arq).

pega_lado(A,B,N,0) :- lado(N,A,B).
pega_lado(A,B,N,1) :- lado(N,B,A).

pega_vertice([!(P1,P2),!(P3,P4)],V) :-
pega_vert(P1,P2,P3,P4,V).

pega_vert(V,_,V,_,V).
pega_vert(V,_,V,V,V).
pega_vert(_,V,V,_,V).
pega_vert(_,V,_,V,V).

processa_tri(Arq,cond(L,J)):-
pega_lado(L,J,N,Sentido),
write(Arq,'L'),
printf(Arq,N,2),
write(Arq,Sentido), % Sentido = 0 (normal) ou 1
(invertido)
nl(Arq).

processa_tri(Arq,Tri):-
recebe_lado(Tri,Lin),
Lin = [_,_],!,

```

```

pega_vertice(Lin,Vert),
write(Arq,'T'),
printf(Arq,Tri,2),
write(Arq,2),
printf(Arq,Vert,2),
write(Arq,v),
envia_lado(Tri,Lout),
imprime(Arq,Lout,s),
nl(Arq).

processa_tri(Arq,Tri) :-
envia_lado(Tri,Lout),
recebe_lado(Tri,Lin),
nro_el(Lin,N1),
nro_el(Lout,N2),
N is N1 + N2,
write(Arq,'T'),
printf(Arq,Tri,2),
write(Arq,N),
imprime(Arq,Lin,e),
imprime(Arq,Lout,s),
nl(Arq).

imprime(Arq,[!(N1,N2)]L,Tipo) :-
lado(Nro,N1,N2),
printf(Arq,Nro,2),
write(Arq,Tipo),!,
imprime(Arq,L,Tipo),
imprime(Arq,[],_).

printf(Arq,Nro,N) :-
Val is 10^(N-1),
quantifica_espacos(Nro,Val,Esp),
tab(Arq,Esp),
write(Arq,Nro).

quantifica_espacos(Nro,Val,0) :- Val =< Nro,!.
quantifica_espacos(Nro,Val,Esp) :-
quantifica_espacos(Nro*10,Val,Esp1),
Esp is Esp1 + 1.

```

Processamento Numérico Bidimensional

O programa a seguir quantifica a propagação bidimensional do escoamento na bacia hidragráfica e a propagação unidimensional no canal. As informações sobre os vértices, lados, triângulos, ordem do processamento dos triângulos e a comunicação entre eles encontram-se nos arquivos coord.com, triang.com, lados.com e ordem.com.

Os principais procedimentos são destacados a seguir:

- **Le_ordem_Processamento:** Faz a leitura dos arquivos citados (*.com), contendo as informações sobre o processamento da região e armazena-os em vetores.
- **Calcula_Triangulo:** Calcula as matrizes A_i , B_i e C_i dos triângulos internos de cada região triangular.
- **Monta_Matriz_Geral:** Totaliza as matrizes internas A_i , B_i e C_i , gerando as matrizes A , B e C (que são elementos das equações integrais do método dos elementos finitos).
- **Calcula_Onda:** Cálcula os novos valores de h , em função dos valores no tempo anterior. Utiliza o método de Runge-Kutta ordem 4. A condição de contorno é definida pelo procedimento **Preenche_Contorno**. A comunicação para os elementos a jusante é feita pelo procedimento **Preenche_Saida**.

Os procedimentos descritos estão presentes tanto no caso unidimensional quanto no bidimensional, trabalhando praticamente da mesma forma.

<pre> program Elementos_Finitos; uses crt,printer,dos,jlf,graph; const NPt = 6; NQTri = 99; N_Chezy = 1.5; type dado = real; tri = 1..3; QTRI = 1..NQTri; Pontos = 1..NPt; lista_adjacentes = array[1..3] of record nro_adj : QTRI; com : -1..1; end; var DeltaT : real; Area_Tri,Area_Total:real; NroTriang,MaxTri : Qtri; ContTri,lixo:integer; l : array[1..3] of record nro : QTRI; com : -1..2; {2 indica entrada e saida} normal : boolean; pos : array[1..5] of integer; end; vl : array[1..3] of record local : Pontos; </pre>	<pre> global : Qtri; val:real; end; VTriang : array[QTRI] of record vi,vj,vk : integer; end; Lado : array[QTRI] of record vl,v2 : Qtri; ja_process:boolean; end; temp : integer; Ordem : array[QTRI] of record nro_tri : QTRI; qtd_adj : 1..3; list_adj : lista_adjacentes; elemento : char; end; Coord : array[QTRI] of record px.py.pz : real; fonte:boolean; end; procedure Le_Dados_Iniciais; Begin write('Digite Chuva : ').readln(Chuva); write('Digite Tempo Maximo : ').readln(Tmax); write('Digite Intervalo de tempo : ').readln(DeltaT); </pre>
--	--


```

End;

procedure Le_Ordem_Processamento;
var arq : text;
    numtri,i : integer;
    j,n : tri;
    x : Qtri;
    tipo,tl : char;

function le_nro(n:integer):integer;
var c:char;
    x:0..9;
begin
    read(arq,c);
    if ord(c) = 32 then x := 0 else x := ord(c) - 48;
    if n = 1 then le_nro := x
    else le_nro := x*Trunc(Pot(10,n-1)) + le_nro(n-1);
end;

function le_real(n1,n2:integer):real;
var x1,x2,fator:real;
    i:integer;
    c:char;
begin
    x1 := le_nro(n1);
    read(arq,c);
    fator := 10000;
    x2 := le_nro(n2) + fator;
    for i := 1 to n2 do begin x2 := x2/10; fator := fator/10;
end;
    le_real := x1 + x2 - fator;
end;

Begin

/* Leitura do arquivo de ordem de processamento dos
   triangulos */

assign(arq,'ordem.com');
reset(arq);
i := 0;
read(arq,tl);
while tl <> '0' do begin
    i := i + 1;
    Ordem[i].elemento := tl;
    numtri := le_nro(2);
    Ordem[i].nro_tri := numtri;
    if tl = 'T' then begin
        n := le_nro(1);
        Ordem[i].qtd_adj := n;
        for j := 1 to n do
            begin
                x := le_nro(2);
                Ordem[i].list_adj[j].nro_adj := x;
                read(arq,tipos);
                if tipos = 'e' then Ordem[i].list_adj[j].com := 1
                else Ordem[i].list_adj[j].com := -1;
            end;
        end;
        readln(arq);
        read(arq,tl);
    end;
    Maxtri := i;
    close(arq);

/* Leitura do arquivo com os lados na malha triangular
   */

assign(arq,'lados.com');

```

```

reset(arq);
numtri := le_nro(3);
while numtri <> 0 do begin
    Lado[numtri].v1 := le_nro(3);
    Lado[numtri].v2 := le_nro(3);
    Lado[numtri].ja_process := False;
    readln(arq);
    numtri := le_nro(3);
end;
close(arq);

/* Leitura do arquivo com as coordenadas dos pontos na
   malha triangular */
assign(arq,'coord.com');
reset(arq);
numtri := le_nro(2); { Neste caso. numtri = numero do
   vertice }
while numtri <> 0 do begin
    Coord[numtri].px := le_real(4,1);
    Coord[numtri].py := le_real(4,1);
    Coord[numtri].pz := le_real(4,1);
    Coord[numtri].fonte := false;
    readln(arq);
    numtri := le_nro(2)
end;
close(arq);

/* Leitura do arquivo com as definicoes dos triangulos
   */
assign(arq,'triang.com');
reset(arq);
numtri := le_nro(2);
while numtri <> 0 do
    with VTriang[numtri] do begin
        vi := le_nro(3);
        vj := le_nro(3);
        vk := le_nro(3);
        readln(arq);
        numtri := le_nro(2);
    end;
close(arq);
End;

procedure Definir_Posicoes_Locais;
var t1,t2:1..3;
Begin
    for t1 := 1 to 3 do
        with l[t1] do
            case t1 of
                1 : begin { * i j * }
                    pos[1] := 1; pos[2] := 2; pos[3] := 4;
                    end;
                2 : for t2 := 1 to 3 do pos[t2] := t2+3; { * j k * }
                3 : begin { * i k * }
                    pos[1] := 1; pos[2] := 3; pos[3] := 6;
                    end;
            end;
        vl[1].local := 1; vl[2].local := 4; vl[3].local := 6;
    End;

function Arredonda(N:dado):dado;
var r:dado;
begin
    r := int(1e12*N)/1e12;
    if abs(r) < 1e-5 then Arredonda:=0
    else Arredonda := r;
end;

PROCEDURE PROCESSA_BIDIMENSIONAL;

```

```

const wi = 1; wj = 2; wk = 3;
  Ax = 1; Ay = 2; Af = 3;
  NTri = 4;

type vetor = array[Pontos] of dado;
  matriz = array[Pontos,Pontos] of dado;
  vetor3 = array[tri] of dado;
  matriz3 = array[tri,tri] of dado;
  Reg = array[1..3] of real;
  file_com = file of Reg;
  file_real = file of real;
var p : tri;
  Tmax, DeltaT, Chuva: real;
  X, Y: vetor3;
  hh, h, BB, CC, NN : vetor;
  AA, AI, Nb, Nc : matriz;
  i : tri;
  tempo: real;
  alfa, alfa_x, alfa_y, Mh, AAt : real; (*Mh = m (Q = alfa*h^m)*)
  (*AAt = 2A (A = area triangulo) *)

  arq: text;
  arqp: array[1..3] of file_real;
  larq, xarq : array[1..3] of file_com;
  Rx : array[Tri] of Reg;

function Vazao(Direcao: tri; h: real): real;
var alfa_geral: real;
Begin
  case Direcao of
    Ax : alfa_geral := alfa_x;
    Ay : alfa_geral := alfa_y;
    Af : alfa_geral := alfa;
  end;
  Vazao := alfa_geral * Pot(h, Mh);
End;

function Q: real;
begin
  Q := Chuva;
end;

Procedure Inicializa_Vet(var v: vetor);
var l: Pontos;
Begin
  for l := 1 to NPt do v[l] := 0;
End;
Procedure Inicializa_Mat(var M: matriz);
var i, j: Pontos;
Begin
  for i := 1 to NPt do
    for j := 1 to NPt do M[i, j] := 0;
  End;

procedure Grava_Vetor(h: vetor);
var l: Pontos;
Begin
  write(arq, ' ', tempo: 3:0, ' ');
  for l := 1 to NPt do write(arq, Vazao(af, h[l]): 8:3, ' ');
  writeln(arq);
End;

Procedure Monta_Comunicacoes(NroTriang, Cont: QTri);
var i, p1, p2, p3: integer;
  R: lista_adjacentes;
  pos : tri;
  N_nro_lado : Qtri;
  sit: boolean;

function Busca(p1, p2: QTri; var sit: boolean; var
  num: Qtri): integer;

```

```

var i: integer;
  l1, l2: Qtri;
  achou : boolean;
Begin
  i := 0;
  achou := false;
  repeat
    i := i + 1;
    num := R[i].nro_adj;
    l1 := Lado[num].v1;
    l2 := Lado[num].v2;
    achou := ((l1 = p1) and (l2 = p2)) OR
      ((l1 = p2) and (l2 = p1));
  until achou or (i = N);
  if achou then begin
    sit := (l1 = p1);
    Busca := i;
  end
  else Busca := 0;
End;

Begin
  N := Ordem[Cont].qtd_adj;
  R := Ordem[Cont].list_adj;
  with VTriang[NroTriang] do
    for i := 1 to 3 do
      begin
        case i of
          1 : begin p1 := vi; p2 := vj; end;
          2 : begin p1 := vj; p2 := vk; end;
          3 : begin p1 := vi; p2 := vk; end;
        end;
        pos := Busca(p1, p2, sit, nro_lado);
        l[i].nro := nro_lado;
        if pos = 0 then l[i].com := 0
        else l[i].com := R[pos].com;
        l[i].normal := sit;
      end;
    end;
  End;

Procedure Leia_Pontos(var X: vetor3; var
  Y: vetor3; NroTriang: QTri);
const i = 1; j = 2; k = 3; C = -1;
var Z: vetor3;
  So_x, So_y: real;
Begin
  with VTriang[NroTriang] do begin
    X[1] := Coord[vi].px; Y[1] := Coord[vi].py; Z[1] :=
    Coord[vi].pz;
    X[2] := Coord[vj].px; Y[2] := Coord[vj].py; Z[2] :=
    Coord[vj].pz;
    X[3] := Coord[vk].px; Y[3] := Coord[vk].py; Z[3] :=
    Coord[vk].pz;
  end;
  Area_tri := X[i]*Y[j]+X[j]*Y[k]+X[k]*Y[i]-Y[i]*X[j]-
  Y[j]*X[k]-Y[k]*X[i];
  So_x := (Z[i]*(Y[j] - Y[k]) + Z[j]*(Y[k] - Y[i]) +
  Z[k]*(Y[i] - Y[j]))/Area_tri;
  So_y := (Z[i]*(X[k] - X[j]) + Z[j]*(X[i] - X[k]) +
  Z[k]*(X[j] - X[i]))/Area_tri;
  if So_x > 0 then Alfa_x := C*sqrt(So_x)
  else Alfa_x := -C*sqrt(-So_x);
  if So_y > 0 then Alfa_y := C*sqrt(So_y)
  else Alfa_y := -C*sqrt(-So_y);
  Alfa := sqrt(sqrt(Alfa_x) + sqrt(Alfa_y));
  Mh := N_Chezy;
  Area_Total := Area_Total + Area_tri/2;
End;

```

```

procedure Calcula_Triangulo(X,Y:vetor3; var
AA,NB,NC:matriz3; var NN:vetor3);
var a,b,c:vetor3;

procedure Imprime_Mat3(M:matriz3);
var i,j:Pontos;
Begin
for i := 1 to 3 do begin
for j := 1 to 3 do write(M[i,j]:10:2);
writeln;
end;
End;
procedure Calcula_Constantes;
var pi,pj,pk:tri;
Begin
AAAt := 0;
for pi := 1 to 3 do begin
pj := pi mod 3 + 1;
pk := pj mod 3 + 1;
a[pi] := X[pj]*Y[pk] - X[pk]*Y[pj];
b[pi] := Y[pj] - Y[pk];
c[pi] := X[pk] - X[pj];
AAAt := AAAt + a[pi];
end;
End;

procedure Calcula_Matriz_h2(Var AA:matriz3);
var pi,pj : tri;

function Integral_Regiao(pi,pj:tri):dado;
var I1,I2,I3:dado;

function Integral_Reta(i,j,o,f:tri):dado;
var m,t,I1,I2,I3,I4,Dx1,Dx2,Dx3,Dx4:dado;
Begin
if X[f] = X[o] then Integral_Reta := 0
else begin
m := (Y[f] - Y[o])/(X[f] - X[o]);
t := Y[o] - m*X[o];
I4 := b[i]*b[j]*m + c[i]*c[j]*m*m*m/3 +
(b[i]*c[j]+b[j]*c[i])*m*m*m/2;
I3 := b[i]*b[j]*t + (a[i]*b[j] + a[j]*b[i])*m +
(a[i]*c[j]+a[j]*c[i])*m*m/2 +
(b[i]*c[j]+b[j]*c[i])*m*t + c[i]*c[j]*m*m*t;
I2 := a[i]*a[j]*m + (a[i]*b[j] + a[j]*b[i])*t +
(a[i]*c[j] + a[j]*c[i])*m*t +
(b[i]*c[j] + b[j]*c[i])*t*t/2 + c[i]*c[j]*m*t*t;
I1 := a[i]*a[j]*t + (a[i]*c[j]+a[j]*c[i])*t*t/2 +
c[i]*c[j]*t*t*t/3;
Dx4 := X[f]*X[i]*X[j]*X[f]*X[f] -
X[o]*X[o]*X[o]*X[o];
Dx3 := X[f]*X[i]*X[j] - X[o]*X[o]*X[o];
Dx2 := X[f]*X[i] - X[o]*X[o];
Dx1 := X[f] - X[o];
Integral_Reta := I1*Dx1 + I2*Dx2/2 + I3*Dx3/3 +
I4*Dx4/4;
end;
End;

Begin { * Integral_Regiao * }
I1 := Integral_Reta(pi,pj,wj,wi);
I2 := Integral_Reta(pi,pj,wi,wk);
I3 := Integral_Reta(pi,pj,wj,wk);
Integral_Regiao := (I1 + I2 - I3)/(AAAt*AAAt);
End;

Begin { * Calcula_Matriz_h2 * }
for pi := 1 to 3 do
for pj := 1 to 3 do
if pi > pj then AA[pi,pj] := AA[pj,pi]

```

```

else AA[pi,pj] := Integral_Regiao(pi,pj);
End;

procedure Calcula_NB(var Nb:matriz3);
var pi,pj:tri;

function CalculaNB_regiao(pi,pj:tri):dado;
var NB1,NB2,NB3 : dado;

function CalculaNB_reta(l,p,o,f:tri):dado;
var m,t,I1,I2,I3,Dx1,Dx2,Dx3:dado;
Begin
if X[f] = X[o] then CalculaNB_reta := 0
else begin
m := (Y[f] - Y[o])/(X[f] - X[o]);
t := Y[o] - m*X[o];
I3 := b[l]*b[p]*m + c[l]*b[p]*m*m/2;
I2 := a[l]*b[p]*m + b[l]*b[p]*t + c[l]*b[p]*m*t;
I1 := a[l]*b[p]*t + c[l]*b[p]*t*t/2;
Dx3 := X[f]*X[l]*X[p] - X[o]*X[o]*X[o];
Dx2 := X[f]*X[l] - X[o]*X[o];
Dx1 := X[f] - X[o];
CalculaNB_reta := I1*Dx1 + I2*Dx2/2 + I3*Dx3/3;
end;
End;

Begin { * CalculaNB_regiao * }
NB1 := CalculaNB_reta(pi,pj,wj,wi);
NB2 := CalculaNB_reta(pi,pj,wi,wk);
NB3 := CalculaNB_reta(pi,pj,wj,wk);
CalculaNB_regiao := (NB1 + NB2 - NB3)/(AAAt*AAAt);
End;

Begin { * Calcula_NB * }
for pi := 1 to 3 do
for pj := 1 to 3 do
Nb[pi,pj] := CalculaNB_regiao(pi,pj);
End;

procedure Calcula_NC(var Nc:matriz3);
var pi,pj:tri;
function CalculaNC_regiao(pi,pj:tri):dado;
var NC1,NC2,NC3 : dado;

function CalculaNC_reta(l,p,o,f:tri):dado;
var m,t,I1,I2,I3,Dx1,Dx2,Dx3:dado;
Begin
if X[f] = X[o] then CalculaNC_reta := 0
else begin
m := (Y[f] - Y[o])/(X[f] - X[o]);
t := Y[o] - m*X[o];
I3 := b[l]*c[p]*m + c[l]*c[p]*m*m/2;
I2 := a[l]*c[p]*m + b[l]*c[p]*t + c[l]*c[p]*m*t;
I1 := a[l]*c[p]*t + c[l]*c[p]*t*t/2;
Dx3 := X[f]*X[l]*X[p] - X[o]*X[o]*X[o];
Dx2 := X[f]*X[l] - X[o]*X[o];
Dx1 := X[f] - X[o];
CalculaNC_reta := I1*Dx1 + I2*Dx2/2 + I3*Dx3/3;
end;
End;

Begin { * CalculaNC_regiao * }
NC1 := CalculaNC_reta(pi,pj,wj,wi);
NC2 := CalculaNC_reta(pi,pj,wi,wk);
NC3 := CalculaNC_reta(pi,pj,wj,wk);
CalculaNC_regiao := (NC1 + NC2 - NC3)/(AAAt*AAAt);
End;

Begin { * Calcula_NC * }

```

```

for pi := 1 to 3 do
  for pj := 1 to 3 do
    Nc[pi,pj] := CalculaNC_regiao(pi,pj);
  End;
End;

procedure Calcula_Vetor_NN(var NN:vetor3);
var l : tri;

function CalculaN_regiao(l:tri):dado;
var N1,N2,N3 : dado;

function CalculaN_reta(l,o,f:tri):dado;
var m,t,l1,l2,l3,Dx1,Dx2,Dx3:dado;
Begin
  if X[f] = X[o] then CalculaN_reta := 0
  else begin
    m := (Y[f] - Y[o])/(X[f] - X[o]);
    t := Y[o] - m*X[o];
    l3 := b[l]*m + c[l]*m*m/2;
    l2 := a[l]*m + b[l]*t + c[l]*m*t;
    l1 := a[l]*t + c[l]*t*t/2;
    Dx3 := X[f]*X[f]*X[f] - X[o]*X[o]*X[o];
    Dx2 := X[f]*X[f] - X[o]*X[o];
    Dx1 := X[f] - X[o];
    CalculaN_reta := l1*Dx1 + l2*Dx2/2 + l3*Dx3/3;
  end;
End;

Begin { * CalculaN_regiao * }
N1 := CalculaN_reta(l,wj,wi);
N2 := CalculaN_reta(l,wi,wk);
N3 := CalculaN_reta(l,wj,wk);
CalculaN_regiao := (N1 + N2 - N3)/AA;
End;

Begin { * Calcula_Vetor_NN * }
for l := 1 to 3 do NN[l] := CalculaN_regiao(l);
End;

Begin { * Proc. Calcula Triangulo * }
Calcula_Constantes;
Calcula_Matriz_h2(AA);
Calcula_NB(Nb);
Calcula_NC(Nc);
Calcula_Vetor_NN(NN);
End;

procedure Pega_Coordenadas_Tri(N:integer;var
i,j,k:Pontos);
var R:string[3];
Begin
  case N of
    1 : R := '123';
    2 : R := '245';
    3 : R := '253';
    4 : R := '356';
  end;
  i := ord(R[1]) - 48;
  j := ord(R[2]) - 48;
  k := ord(R[3]) - 48;
End;

procedure Calcula_Regiao(X,Y:vetor3);
type Coord = record cx,cy : real; end;
  Trig = array [1..Npt] of Coord;
var V : Trig;
  i : integer;

procedure Calcula_Pontos_Internos(X,Y:vetor3; var
V:Trig);

```

```

const Da=1;Db=2;Dc=3;
var D : array[tri] of Coord;
procedure Calcula_Diferenca(n,i,j:tri);
Begin
  D[n].cx := (x[i]-x[j])/2;
  D[n].cy := (y[i]-y[j])/2;
End;

procedure Pontos_Globais(n:Pontos:i:tri);
Begin
  V[n].cx := X[i];
  V[n].cy := Y[i];
End;

procedure Calcula_Pontos(n.p:Pontos;t:tri);
Begin
  V[n].cx := V[p].cx + D[t].cx;
  V[n].cy := V[p].cy + D[t].cy;
End;

Begin { *Calcula_Pontos_Internos * }
Calcula_Diferenca(Da,wi,wj);
Calcula_Diferenca(Db,wk,wi);
Calcula_Diferenca(Dc.wk,wj);

Pontos_Globais(1,wi);
Pontos_Globais(4,wj);
Pontos_Globais(6,wk);

Calcula_Pontos(2,4,Da);
Calcula_Pontos(3,1,Db);
Calcula_Pontos(5,4,Dc);
End;

procedure Monta_Matriz_Geral(pi,pj,pk:Pontos;
Ai,Nbi,Nci:matriz3;Ni:vetor3);
var W:array[1..3] of Pontos;
  k1,k2:Tri;
  i,j : Pontos;
Begin
  W[wi] := pi; W[wj] := pj; W[wk] := pk;
  for k1 := wi to wk do begin
    i := w[k1];
    for k2 := wi to wk do begin
      j := w[k2];
      AA[i,j] := Arredonda(AA[i,j] + Ai[k1,k2]);
      Nb[i,j] := Arredonda(Nb[i,j] + Nbi[k1,k2]);
      Nc[i,j] := Arredonda(Nc[i,j] + Nci[k1,k2]);
    end;
    NN[i] := Arredonda(NN[i] + Ni[k1]);
  end;
End;

procedure Processa_Triangulo(N:integer);
var pi,pj,pk : pontos;
  AAi,Nbi,Nci : matriz3;
  NNi : vetor3;
Begin
  Pega_Coordenadas_Tri(N,pi,pj,pk);
  X[wi] := V[pi].cx; Y[wi] := V[pi].cy;
  X[wj] := V[pj].cx; Y[wj] := V[pj].cy;
  X[wk] := V[pk].cx; Y[wk] := V[pk].cy;
  Calcula_Triangulo(X,Y,AAi,Nbi,Nci,NNi);
  Monta_Matriz_Geral(pi,pj,pk,AAi,Nbi,Nci,NNi);
End;

Begin { * Calcula Regiao * }
Calcula_Pontos_Internos(X,Y,V);
for i := 1 to NTri do Processa_Triangulo(i);

```

```

End;
procedure Calcula_Vetor_h1(h:vetor;var BB:vetor);
var pi,pj,pk,i,j,t : Pontos;
    hi,hk,hj : real;
    Qx,Qy:vetor;
    Somat:dado;
Begin { * Calcula_Matriz_h1 * }
for i := 1 to Npt do begin
    Qx[i] := Arredonda(Vazao(Ax,h[i]));
    Qy[i] := Arredonda(Vazao(Ay,h[i]));
end;
for i := 1 to Npt do begin
    Somat := 0;
    for j := 1 to Npt do
        Somat := Somat + Nb[i,j]*Qx[j] + Nc[i,j]*Qy[j];
    BB[i] := Arredonda(Somat);
end;
End;

procedure Calcula_Vetor_h0(Var CC:vetor;Q:dado);
var i : Pontos;
Begin
for i := 1 to NPT do CC[i] := -NN[i]*Q;
End;

procedure CalculaK(h:vetor;var k:vetor);
var T,BQ:vetor;
    i,j:Pontos;
    Somat:dado;
Begin
    Calcula_vetor_h1(h,BQ);
for i := 1 to Npt do
    T[i] := BQ[i] + CC[i]; { T = BQ + C }
for i := 1 to Npt do
begin
    Somat := 0;
    for j := 1 to Npt do Somat := Somat + AI[i,j]*T[j];
    {Somat = AI*(Bh + C)}
    k[i] := -Somat;
end;
End;

procedure Calcula_Onda;
var i,j:Pontos;
    k1,k2,k3,k4:vetor;
    Somat : dado;
    NCont : array[1..Npt] of boolean;

procedure Preenche_Contorno;
var i,j,vert : 1..Npt;
begin
for i := 1 to 3 do
with I[i] do
case com of
0 : for j := 1 to 3 do begin
vert := pos[j];
h[vert] := 0;
NCont[vert] := false;
end;
1 : if normal then for j := 1 to 3 do begin
vert := pos[j];
h[vert] := Pot(Rx[j,i]/Alfa,1/mh);
NCont[vert] := false;
end
else for j := 3 downto 1 do begin
vert := pos[4-j];
h[vert] := Pot(Rx[j,i]/Alfa,1/mh);
NCont[vert] := false;
end
end;
end;

procedure Preenche_Saida;

```

```

var i,j : 1..Npt;
    Raux : real;

function contor(i:integer):integer;
begin
    if i = 1 then contor := 0
    else contor := 1;
end;

begin
for i := 1 to 3 do
with I[i] do
if (com = -1) or (com = 2) then
if normal then for j := 1 to 3 do
    Rx[i,j] := contor(j)*Rx[i,j]*(com+1)/3 +
Vazao(Af,h[pos[j]])
else for j := 3 downto 1 do
    Rx[i,j] := contor(j)*Rx[i,j]*(com+1)/3 +
Vazao(Af,h[pos[4-j]]);
for i := 1 to 3 do
end; {O fator que multiplica Rx eh 0 se com = -1 e
1 se com = 2}

Begin { * Calcula Onda * }
CalculaK(h,k1);
for i := 1 to Npt do hh[i] := h[i] + DeltaT*k1[i]/2;
CalculaK(hh,k2);
for i := 1 to Npt do hh[i] := h[i] + DeltaT*k2[i]/2;
CalculaK(hh,k3);
for i := 1 to Npt do hh[i] := h[i] + DeltaT*k3[i];
CalculaK(hh,k4);
Preenche_Contorno;
for i := 1 to Npt do
if NCont[i] then
h[i] := h[i] +(k1[i]+2*k2[i]+2*k3[i]+k4[i])*DeltaT/6;
Preenche_Saida;
End;

procedure Calcula_Inversa(M:matriz;var W:matriz);
var H:array[1..6,1..12] of dado;
    pt,pj,i,j,l:1..12;
    S : dado;
Begin
for i := 1 to 6 do
for j := 1 to 6 do
begin
H[i,j] := M[i,j];
if i = j then H[i,j+6] := 1
else H[i,j+6] := 0;
end;
for i := 1 to 6 do begin
S := H[i,i];
for j := i to 12 do H[i,j] := H[i,j]/S;
for l := i + 1 to 6 do begin
S := H[l,i];
for j := 1 to 12 do H[l,j] := H[l,j] - H[i,j]*S;
end;
end;
for i := 2 to 6 do
for l := 1 to i-1 do begin
S := H[l,i];
for j := i to 12 do H[l,j] := H[l,j] - H[i,j]*S;
end;
for i := 1 to 6 do
for j := 1 to 6 do W[i,j] := H[i,j+6];
End;

procedure Multiplica(A,B:matriz; var C:matriz);
var i,j,k:1..12;
begin

```

```

for i := 1 to 6 do
  for j := 1 to 6 do begin
    C[i,j] := 0;
    for k := 1 to 6 do C[i,j] := C[i,j] + A[i,k]*B[k,j];
    end;
  end;

procedure Abre_Arquivos;
var i:tri;
    Si,Sj,Sk,S,Nome,SNum : string[9];
begin
  str(NroTriang,S);
  S := 'alt'+S+'.dat';
  assign(arq,S);
  rewrite(arq);
  for i := 1 to 3 do
    with l[i] do
      if com <> 0 then begin
        str(nro,Snum);
        Nome := 'Y' + SNum + '.lig';
        assign(Larq[i],Nome);
        if com = 1 then reset(Larq[i])
        else with Lado[nro] do
          if ja_process then begin
            reset(Larq[i]);
            Nome := 'K Y'+Snum+'.dup';
            assign(Xarq[i],Nome);
            rewrite(Xarq[i]);
            com := 2;
          end
          else begin
            rewrite(Larq[i]);
            ja_process := True;
          end;
        end;
      with Vtriang[NroTriang] do begin
        str(vi,Si); str(vj,Sj); str(vk,Sk);
        vl[1].global := vi; vl[2].global := vj; vl[3].global := vk;
      end;
    end;

procedure Fecha_Arquivos;
var i:tri;
    Nome,SNum : string[8];
    R:Reg;
Begin
  for i := 1 to 3 do if l[i].com <> 0 then close(Larq[i]);
  for i := 1 to 3 do
    if l[i].com = 2 then begin
      close(Xarq[i]);
      str(l[i].nro,Snum);
      Nome := 'Y' + SNum + '.lig';
      reset(Xarq[i]);
      rewrite(Larq[i]);
      while not eof(Xarq[i]) do begin
        read(Xarq[i],R);
        write(Larq[i],R);
      end;
      close(Larq[i]);
      close(Xarq[i]);
    end;
  close(arq);
End;

procedure Inicializa_Todos;
Begin
  Inicializa_Mat(AA);
  Inicializa_Mat(Nb);
  Inicializa_Mat(NC);

```

```

  Inicializa_Vet(NN);
  Inicializa_Vet(h);
End;

BEGIN {PROCESSA_BIDIMENSIONAL}
  NroTriang := Ordem[ContTri].nro_tri; { Pega novo triangulo }
  Leia_Pontos(X.Y,NroTriang); { Pega coordenadas do triang }
  Monta_Comunicacoes(NroTriang,ContTri); { Monta o vetor l[3], que indica }
  Abre_Arquivos;
  Inicializa_Todos;
  Calcula_Regiao(X,Y);
  Calcula_Inversa(AA,Al);

  tempo := 0;
  writeln(arq, ' alfa_x = 'alfa_x:5:2,' alfa_y = 'alfa_y:5:2);
  write(arq, ' tempo ');
  for lixo := 1 to NPt do write(arq,lixo:6,' ');
  writeln(arq);
  Grava_Vetor(h);

  for temp := 1 to round(Tmax/DeltaT) do
  begin
    clrscr;
    tempo := temp*DeltaT;
    writeln;
    write(' Triangulo Nro ',NroTriang,' Tmax = ',Tmax);
    writeln(' Alfa_x = ',Alfa_x:5:2,' Alfa_y = ',Alfa_y:5:2);
    WRITELN('DT = ',DELTAT:10:5);
    writeln;
    writeln(1st,* tempo = ',tempo:4:1,' # q = ',Chuva:5:2,' *');
    for i := 1 to 3 do if l[i].com > 0 then
      read(Larq[i],Rx[i]);
      Calcula_Vetor_h0(CC,q);
      Calcula_Onda;
      for i := 1 to 3 do case l[i].com of
        -1 : write(Larq[i],Rx[i]);
        2 : write(Xarq[i],Rx[i]);
      end;
      if frac(tempo) = 0 then Grava_Vetor(h);
    end;
  Fecha_Arquivos;
END;

procedure Processa_Unidimensional;
const wi = 1; wj = 2; wk = 3;
      NPt = 2; NEle = 1;
      Larg = 1;

type
  tri = 1..2;
  Pontos = 1..NPt;
  Reg = array[1..3] of dado;
  vetor = array[Pontos] of dado;
  matriz = array[Pontos,Pontos] of dado;
  vetor2 = array[tri] of dado;
  matriz2 = array[tri,tri] of dado;

var p : tri;
    NroLado : QTri;
    pto_entrada.pto_saida : Pontos;
    X :vetor2; L : real;
    hh,h,BQ,CC,NN : vetor;
    AA,Al,BB : matriz;
    temp : integer;

```

```

tempo:real;
alfa,Mh : real;
arq : text;
arqent,arqsai : file of real;
link1,link2 : file of Reg;
med,saida : array[0..2] of file of real;
tem_entrada,pto_juncao:boolean;
Times integer

function Hc:dado;
var Qc,qi:dado;
begin
  read(saida[times - 1],Qc);
  if times = 1 then
    if tem_entrada then begin
      read(arqent,qi);
      Qc := Qc + qi;
    end;
    Hc := Pot(Qc/alfa,1/Mh);
  end;

Procedure Inicializa_Vet(var v:vetor);
var l:Pontos;
Begin
  for l := 1 to NPt do v[l] := 0;
End;

Procedure Inicializa_Mat(var M:matriz);
var i,j:Pontos;
Begin
  for i := 1 to NPt do
    for j := 1 to NPt do M[i,j] := 0;
  End;

procedure Grava_Vetor(h:vetor);
var l:Pontos;
QX : dado;
Begin
  write(arq,tempo:4:0,' ',q:5:2);
  for l := 1 to NPt do
    begin
      Qx := Pot(h[l],Mh)*alfa;
      write(arq,Qx:8:2,' ');
    end;
  writeln(arq);
End;

Procedure Leia_Pontos(var X:vetor2);
var x1,x2,y1,y2,z1,z2,dd : dado;
Begin
  with Lado[NroLado] do begin
    x1 := Coord[v1].px;
    y1 := Coord[v1].py;
    z1 := Coord[v1].pz;
    x2 := Coord[v2].px;
    y2 := Coord[v2].py;
    z2 := Coord[v2].pz;
    if z1 > z2 then begin
      pto_entrada := v1;
      pto_saida := v2;
    end
    else begin
      pto_entrada := v2;
      pto_saida := v1;
    end;
  end;
  dd := sqrt(y2 - y1) + sqrt(x2 - x1);
  X[1] := 0;
  X[2] := sqrt(dd);
  Mh := N_Chezy;

  alfa := 0.6;
End;

procedure Calcula_Elemento(X:vetor2; var
AA,BB:matriz2; var NN:vetor2);
var a,b,c:vetor2;

procedure Imprime_mat2(M:matriz2);
var i,j:Pontos;
Begin
  for i := 1 to 2 do begin
    for j := 1 to 2 do write(M[i,j]:10:2);
    writeln;
  end;
End;
procedure Calcula_AA(Var AA:matriz2);
var pi,pj : tri;
Begin (* Calcula_Matriz_h2 *)
  for pi := 1 to 2 do
    for pj := 1 to 2 do
      if pi = pj then AA[pi,pj] := L/3
      else AA[pi,pj] := L/6;
    End;
  End;

procedure Calcula_BB(var BB:matriz2);
var pi,pj:tri;
Begin
  For pi := 1 to 2 do
    begin
      BB[pi,1] := -1/2;
      BB[pi,2] := 1/2;
    end;
  End;

procedure Calcula_Vetor_NN(var NN:vetor2);
var l : tri;
Begin (* Calcula_Vetor_NN *)
  for l := 1 to 2 do NN[l] := 1/2;
End;

Begin (* Proc. Calcula Elemento *)
  Calcula_AA(AA);
  Calcula_BB(BB);
  Calcula_Vetor_NN(NN);
End;

procedure Calcula_Regiao(X:vetor2);
type Trig = array [1..NPt] of dado;
var V : Trig;
  i : integer;

procedure Calcula_Pontos_Internos(X:vetor2; var
V:Trig);
var i:integer;
Begin
  L := (X[2] - X[1])/(NEle+1);
  V[1] := X[1];
  for i := 1 to NEle do V[i+1] := V[i] + L;
End;

procedure Monta_Matriz_Geral(N:Pontos;Ai,Bi:matriz2;
Ni:vetor2);
var i,j,k1,k2 : Pontos;
Begin
  for k1 := 1 to 2 do
    begin
      i := k1 + N - 1;
      for k2 := 1 to 2 do begin
        j := k2 + N - 1;
        AA[i,j] := AA[i,j] + Ai[k1.k2];
        BB[i,j] := BB[i,j] + Bi[k1.k2];
      end;
    end;
  End;

```

```

end;
NN[i] := NN[i] + Ni[k1];
end;
End;

procedure Processa_Triangulo(N:integer);
var pi,pj,pk : pontos;
    AAI,BBi : matriz2;
    NNi : vetor2;
Begin
pi := N; pj := N + 1;
X[wi] := V[pi];
X[wj] := V[pj];
Calcula_Elemento(X,AAi,BBi,NNi);
Monta_Matriz_Geral(N,AAi,BBi,NNi);
End;
Begin { * Calcula Regiao * }
Calcula_Pontos_Internos(X,V);
for i := 1 to NEle do Processa_Triangulo(i);
End;

procedure Calcula_Vetor_h1(h:vetor;BB:matriz;var
BQ:vetor);
var pi,pj,pk,i,j,t : Pontos;
    Q:vetor;
    Somat:dado;
Begin { * Calcula_Vetor h1 * }
for i := 1 to Npt do Q[i] := alfa*pot(h[i],Mh);
for i := 1 to Npt do begin
Somat := 0;
for j := 1 to Npt do Somat := Somat + BB[i,j]*Q[j];
BQ[i] := Somat;
end;
End;

procedure Calcula_Vetor_h0(Var CC:vetor);
var i : Pontos;
    qm:Real;
Begin
read(med[times],qm);
for i := 1 to Npt do CC[i] := -NN[i]*qm*L*Larg;
End;

procedure Calcula_Onda(AI:matriz;BQ,CC:vetor);
var i,j:Pontos;
    T:vetor;
    Somat: dado;
    QQ,QI : real;
Begin
for i := 1 to Npt do
T[i] := BQ[i] + CC[i]; { T = BQ + C }
h[1] := Hc;
for i := 2 to Npt do
begin
Somat := 0;
for j := 1 to Npt do Somat := Somat + AI[i,j]*T[j];
{Somat =
AI*(Bh + C)}
h[i] := h[i] - Somat*DeltaT;
if h[i] < 0 then for lixo := 1 to 5 do beep;
end;
QQ := alfa*pot(h[npt],Mh);
write(saida[times],QQ);
if times = 2 then
begin
if pto_juncao then begin
read(arqsai,QI);
QQ := QI + QQ;
seek(arqsai,filepos(arqsai)-1);
end;
write(arqsai,QQ);

```

```

end;
End;

procedure Calcula_Inversa(M:matriz;var W:matriz);
var H:array[1..Npt,1..Npt*2] of dado;
    pi,pj,i,j,l:1..Npt*2;
    S : dado;

Begin
for i := 1 to Npt do
for j := 1 to Npt do
begin
H[i,j] := M[i,j];
if i = j then H[i,j+Npt] := 1
else H[i,j+Npt] := 0;
end;
for i := 1 to Npt do begin
S := H[i,i];
for j := i to Npt*2 do H[i,j] := H[i,j]/S;
for l := i + 1 to Npt do begin
S := H[l,i];
for j := 1 to Npt*2 do H[l,j] := H[l,j] - H[i,j]*S;
end;
end;
for l := 2 to Npt do
for i := 1 to i-1 do begin
S := H[l,i];
for j := i to Npt*2 do H[l,j] := H[l,j] - H[i,j]*S;
end;
for i := 1 to Npt do
for j := 1 to Npt do W[i,j] := H[i,j+Npt];
End;

procedure Multiplica(A,B:matriz; var C:matriz);
var i,j,k:1..Npt;
begin
for i := 1 to Npt do
for j := 1 to Npt do begin
C[i,j] := 0;
for k := 1 to Npt do C[i,j] := C[i,j] + A[i,k]*B[k,j];
end;
end;

procedure Abre_Arquivos;
var Nome,S,Snum,Str_entrada,Str_saida : string;
    i:integer;
Begin
str(NroLado,Snum);
Nome := 'Y'+Snum+'.LIG';
assign(link1,Nome);
assign(med[1],'Media1.xxx');
assign(med[2],'Media2.xxx');
assign(saida[0],'saida0.xxx');
assign(saida[1],'saida1.xxx');
assign(saida[2],'saida2.xxx');
S := 'L'+Snum+'.dat';
assign(arq,S);
rewrite(arq);
for i := 0 to 2 do rewrite(saida[i]);
str(pto_saida,Str_saida);
Str_saida := 'P' + Str_saida + '.pto';
assign(arqsai,Str_saida);
{$I-}
reset(arqsai);
{$I+}
pto_juncao := (IOResult = 0);
if not pto_juncao then rewrite(arqsai);
tem_entrada := Coord[pto_entrada].fonte;
if tem_entrada then begin
str(pto_entrada,Str_entrada);

```



```

    Str_entrada := 'P' + Str_entrada + '.pto';
    assign(arqent,Str_entrada);
    reset(arqent);
end;
End;

```

```

procedure Fecha_Arquivos;
var Rx : Reg;
    i : integer;
Begin
    rewrite(link1);
    while not eof(saida[0]) do begin
        for i := 1 to 3 do read(saida[i-1],Rx[i]);
            write(link1,Rx);
        end;
        for i := 1 to 2 do close(med[i]);
        for i := 0 to 2 do close(saida[i]);
        close(link1);
        close(arq);
        if tem_entrada then close(arqent);
        close(arqsai);
        Coord[pto_saida].fonte := true;
    End;

```

```

procedure Cria_Arq_Media;
var qm1,qm2:real;
    Rx : Reg;
Begin
    reset(link1);
    rewrite(med[1]);
    rewrite(med[2]);
    while not eof(link1) do begin
        read(link1,Rx);
        qm1 := (Rx[1] + Rx[2])/2;
        qm2 := (Rx[2] + Rx[3])/2;
        write(med[1],qm1);
        write(med[2],qm2);
        write(saida[0],Rx[1]);
    end;
    close(link1);
    reset(med[1]);
    reset(med[2]);
End;

```

```

BEGIN { Processa Unidimensional }
    NroLado := Ordem[ContTri].nro_tri;
    Leia_Pontos(X);
    Abre_Arquivos;
    Cria_Arq_Media;

```

```

    clrscr;
    for times := 1 to 2 do begin
        reset(saida[times-1]);
        Inicializa_Mat(AA);
        Inicializa_Mat(BB);
        Inicializa_Vet(NN);
        Inicializa_Vet(h);
        Calcula_Regiao(X);
        Calcula_Inversa(AA,AI);
        tempo := 0;
        writeln(arq,' alfa = ',alfa:5:2);
        write(arq,' tempo * q');
        for lixo := 1 to NPt do write(arq,lixo:6,' ');
        writeln(arq);
        Grava_Vetor(h);
        for temp := 1 to round(Tmax/DeltaT) do
            begin
                tempo := temp*DeltaT;
                writeln;
                clrscr;
                writeln(1st,' Lado ', NroLado);writeln;
                writeln(1st,'* tempo = ',tempo:4:1,' # q = ',q:5:2,' *');
                Calcula_vetor_h1(h,BB,BQ);
                Calcula_Vetor_h0(CC);
                Calcula_Onda(AI,BQ,CC);
                writeln('* Vetor h *');
                Imprime_Vetor(h);
                if (frac(tempo) = 0) then Grava_Vetor(h);
            end;
        end;
        Fecha_Arquivos;
    END;

```

```

BEGIN
    Area_Total := 0;
    Le_Dados_Iniciais;
    Le_Ordem_Processamento;
    Definir_Posicoes_Locais;
    ContTri := 0;
    For Temp := 1 TO NQTri do Lado[Temp].Ja_Process :=
    False;
    Repeat
        ContTri := ContTri + 1;
        case Ordem[ContTri].elemento of
            'T' : Processa_Bidimensional;
            'L' : Processa_Unidimensional;
        end;
    Until ContTri >= MaxTri;
END.

```

APÊNDICE B - Listagem dos Arquivos

Nesta seção serão colocados os arquivos utilizados no exemplo utilizado para o processamento bidimensional em uma bacia hidrográfica.

Arquivos de Comunicação

Coord.com				Triang.com				Lados.com						Ordem.com
1	1.0	9.3	100.0	1	1	2	15	1	1	2	55	22	24	T 33 4c 6s41s
2	1.2	8.5	98.0	2	15	2	14	2	1	15	56	22	25	T 7212v35s
3	0.6	7.2	94.0	3	14	2	16	3	2	3	57	23	25	T 6335e36s39s
4	0.6	5.1	88.0	4	14	16	17	4	2	14	58	23	26	T 5338c39e42s
5	0.9	1.1	80.0	5	14	17	13	5	2	15	59	24	25	T14236e32s
6	3.5	1.5	70.0	6	13	17	11	6	2	16	60	25	26	T 4342e43s41s
7	5.3	0.3	75.0	7	13	11	12	7	2	18	00			T13217v31s
8	6.2	2.3	82.0	8	2	3	18	8	3	4				T12331e33s46s
9	6.4	4.2	88.0	9	2	18	16	9	3	18				T11346e51s45s
10	7.5	7.5	100.0	10	16	18	20	10	3	19				T23233e27s
11	8.5	9.0	105.0	11	16	20	21	11	4	5				L410
12	9.7	10.8	115.0	12	16	21	10	12	4	19				T 82 2v 9s
13	7.4	11.2	106.0	13	16	10	17	13	4	22				T 92 6e44s
14	4.3	11.3	96.0	14	17	10	11	14	4	24				T16348c49s47s
15	2.4	10.4	95.0	15	3	19	18	15	5	6				T153 9e10s47s
16	3.6	9.7	92.5	16	18	19	20	16	5	24				L470
17	6.4	9.5	99.0	17	3	4	19	17	6	7				T10244e45s
18	2.4	8.0	94.1	18	19	4	22	18	6	24				T17210e12s
19	3.0	6.5	86.2	19	19	22	20	19	6	25				L450
20	4.6	6.8	90.0	20	20	22	23	20	6	26				T22221v26s
21	5.9	6.8	93.0	21	20	23	9	21	7	8				T19349e50s52s
22	3.5	5.0	81.0	22	20	9	21	22	7	26				T21326e28s53s
23	5.2	4.4	86.2	23	21	9	10	23	8	9				T20353e54s52s
24	1.7	2.7	79.8	24	4	5	24	24	8	26				L520
25	3.1	2.7	72.0	25	4	24	22	25	9	10				T24311e14s16s
26	4.9	2.7	81.0	26	22	24	25	26	9	20				T18312e13s50s
00				27	22	25	23	27	9	21				L500
				28	23	25	26	28	9	23				T29328e29s58s
				29	23	26	9	29	9	26				T252 4v55s
				30	9	26	8	30	10	11				T30229e24s
				31	24	5	6	31	10	16				T28358e60s57s
				32	24	6	25	32	10	17				T27254e56s
				33	25	6	26	33	10	21				T35224e22s
				34	26	6	7	34	11	12				T26355e56s59s
				35	26	7	8	35	11	13				L560
				00				36	11	17				T34322e17s20s
								37	12	13				T33226v19s
								38	13	14				T31316e18s15s
								39	13	17				T32318e59s19s
								40	14	15				L590
								41	14	16				L191
								42	14	17				00
								43	16	17				
								44	16	18				
								45	16	20				
								46	16	21				
								47	18	19				
								48	18	20				
								49	19	20				
								50	19	22				
								51	20	21				
								52	20	22				
								53	20	23				
								54	22	23				

Arquivos gerados pelo programa Prolog

Comunica.ari

ordem([7,6,5,14,1,4,13,12,23,2,cond(2,15),3,cond(14,16),8,9,16,15,cond(18,19),10,17,11,cond(16,20),22,19,21,20,cond(20,22),24,18,cond(19,22),29,25,30,28,27,35,26,cond(22,25),34,33,31,32,cond(24,25),cond(25,6)]).

envia_lado(1,[l(2,15)]).
 envia_lado(2,[l(14,15),l(2,15)]).
 envia_lado(3,[l(2,16),l(14,16)]).
 envia_lado(4,[l(16,17),l(14,16)]).
 envia_lado(5,[l(14,17)]).
 envia_lado(6,[l(11,17),l(13,17)]).
 envia_lado(7,[l(11,13)]).
 envia_lado(8,[l(3,18)]).
 envia_lado(9,[l(16,18)]).
 envia_lado(10,[l(16,20)]).
 envia_lado(11,[l(20,21),l(16,20)]).
 envia_lado(12,[l(10,21),l(16,21)]).
 envia_lado(13,[l(10,16)]).
 envia_lado(14,[l(10,17)]).
 envia_lado(15,[l(3,19),l(18,19)]).
 envia_lado(16,[l(19,20),l(18,19)]).
 envia_lado(17,[l(4,19)]).
 envia_lado(18,[l(4,22),l(19,22)]).
 envia_lado(19,[l(19,22),l(20,22)]).
 envia_lado(20,[l(22,23),l(20,22)]).
 envia_lado(21,[l(9,23),l(20,23)]).
 envia_lado(22,[l(9,20)]).
 envia_lado(23,[l(9,21)]).
 envia_lado(24,[l(4,24),l(5,24)]).
 envia_lado(25,[l(22,24)]).
 envia_lado(26,[l(22,25),l(24,25)]).
 envia_lado(27,[l(22,25)]).
 envia_lado(28,[l(25,26),l(23,25)]).
 envia_lado(29,[l(9,26),l(23,26)]).
 envia_lado(30,[l(8,26)]).
 envia_lado(31,[l(6,24),l(5,6)]).
 envia_lado(32,[l(24,25),l(6,25)]).
 envia_lado(33,[l(6,25)]).
 envia_lado(34,[l(6,7),l(6,26)]).
 envia_lado(35,[l(7,26)]).

recebe_lado(1,[]).
 recebe_lado(2,[]).
 recebe_lado(3,[]).
 recebe_lado(4,[l(14,17)]).
 recebe_lado(5,[l(13,17)]).
 recebe_lado(6,[l(11,13)]).
 recebe_lado(7,[]).
 recebe_lado(8,[]).
 recebe_lado(9,[l(2,16)]).
 recebe_lado(10,[l(16,18)]).
 recebe_lado(11,[l(16,21)]).
 recebe_lado(12,[l(10,16)]).
 recebe_lado(13,[l(16,17),l(10,17)]).
 recebe_lado(14,[l(11,17)]).
 recebe_lado(15,[l(3,18)]).
 recebe_lado(16,[]).
 recebe_lado(17,[l(3,19)]).
 recebe_lado(18,[l(4,19)]).
 recebe_lado(19,[l(19,20)]).
 recebe_lado(20,[l(20,23)]).
 recebe_lado(21,[l(9,20)]).
 recebe_lado(22,[l(20,21),l(9,21)]).
 recebe_lado(23,[l(10,21)]).
 recebe_lado(24,[]).
 recebe_lado(25,[l(4,22),l(4,24)]).
 recebe_lado(26,[l(22,24)]).
 recebe_lado(27,[l(22,23),l(23,25)]).
 recebe_lado(28,[l(23,26)]).
 recebe_lado(29,[l(9,23)]).
 recebe_lado(30,[l(9,26)]).
 recebe_lado(31,[l(5,24)]).
 recebe_lado(32,[l(6,24)]).
 recebe_lado(33,[l(25,26),l(6,26)]).
 recebe_lado(34,[l(7,26)]).
 recebe_lado(35,[l(8,26)]).

lado(1,1,2).
lado(2,1,15).
lado(3,2,3).
lado(4,2,14).
lado(5,2,15).
lado(6,2,16).
lado(7,2,18).
lado(8,3,4).
lado(9,3,18).
lado(10,3,19).
lado(11,4,5).
lado(12,4,19).
lado(13,4,22).
lado(14,4,24).
lado(15,5,6).
lado(16,5,24).
lado(17,6,7).
lado(18,6,24).
lado(19,6,25).
lado(20,6,26).
lado(21,7,8).
lado(22,7,26).
lado(23,8,9).
lado(24,8,26).
lado(25,9,10).
lado(26,9,20).
lado(27,9,21).
lado(28,9,23).
lado(29,9,26).
lado(30,10,11).

lado(31,10,16).
lado(32,10,17).
lado(33,10,21).
lado(34,11,12).
lado(35,11,13).
lado(36,11,17).
lado(37,12,13).
lado(38,13,14).
lado(39,13,17).
lado(40,14,15).
lado(41,14,16).
lado(42,14,17).
lado(43,16,17).
lado(44,16,18).
lado(45,16,20).
lado(46,16,21).
lado(47,18,19).
lado(48,18,20).
lado(49,19,20).
lado(50,19,22).
lado(51,20,21).
lado(52,20,22).
lado(53,20,23).
lado(54,22,23).
lado(55,22,24).
lado(56,22,25).
lado(57,23,25).
lado(58,23,26).
lado(59,24,25).
lado(60,25,26).

Fluxos.ari

fluxo(2,l(14,15),0.44523223).
 fluxo(3,9,0.61527745).
 fluxo(4,13,0.98091515).
 fluxo(5,4,1).
 fluxo(6,14,0.96841034).
 fluxo(6,5,0.03158966).
 fluxo(7,6,1).
 fluxo(8,15,1).
 fluxo(9,10,1).
 fluxo(11,22,0.29654221).
 fluxo(12,23,0.19573877).
 fluxo(12,11,0.80426123).
 fluxo(13,12,1).
 fluxo(14,13,1).
 fluxo(15,17,0.21443194).
 fluxo(16,19,0.0340724).
 fluxo(17,18,1).
 fluxo(18,25,0.69224453).
 fluxo(20,27,0.96806518).
 fluxo(21,29,0.30338013).
 fluxo(21,20,0.69661987).
 fluxo(22,21,1).
 fluxo(23,22,1).
 fluxo(24,25,0.01222704).
 fluxo(24,31,0.98777296).
 fluxo(25,26,1).
 fluxo(28,33,4.09853662).
 fluxo(28,27,-3.09853662).
 fluxo(29,30,-0.39738851).
 fluxo(29,28,1.39738851).
 fluxo(30,35,1).
 fluxo(31,32,-0.79549286).
 fluxo(31,l(5,6),1.79549286).

fluxo(34,l(6,7),1.29975958).
 fluxo(34,33,-0.29975958).
 fluxo(35,34,1).
 fluxo(1,cond(2,15),1).
 fluxo(2,cond(2,15),0.55476777).
 fluxo(3,cond(14,16),0.38472255).
 fluxo(4,cond(14,16),0.1908485).
 fluxo(10,cond(16,20),1).
 fluxo(11,cond(16,20),0.70345779).
 fluxo(15,cond(18,19),6.78556806).
 fluxo(16,cond(18,19),0.9659276).
 fluxo(18,cond(19,22),0.30775547).
 fluxo(19,cond(19,22),4.87007547).
 fluxo(19,cond(20,22),-3.87007547).
 fluxo(20,cond(20,22),0.03193482).
 fluxo(26,cond(22,25),0.63288293).
 fluxo(27,cond(22,25),1).
 fluxo(26,cond(24,25),0.36711707).
 fluxo(32,cond(24,25),0.76008266).
 fluxo(32,cond(25,6),0.23991734).
 fluxo(33,cond(25,6),1).

condutor(2,15).
 condutor(14,16).
 condutor(16,20).
 condutor(18,19).
 condutor(19,22).
 condutor(20,22).
 condutor(22,25).
 condutor(24,25).
 condutor(25,6).