

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Civil

**Simulação e Controle De Enchentes Usando as Equações de Águas
Rasas e a Teoria Do Controle Ótimo**

Malú Grave

Porto Alegre
2016

MALÚ GRAVE

**SIMULAÇÃO E CONTROLE DE ENCHENTES USANDO AS
EQUAÇÕES DE ÁGUAS RASAS E A TEORIA DO CONTROLE
ÓTIMO**

Dissertação apresentada ao Programa de Pós-Graduação em
Engenharia Civil da Universidade Federal do Rio Grande do Sul,
como parte dos requisitos para obtenção do título de Mestre em
Engenharia

Porto Alegre
2016

CIP - Catalogação na Publicação

Grave, Malú
Simulação e Controle De Enchentes Usando as
Equações de Águas Rasas e a Teoria Do Controle Ótimo /
Malú Grave. -- 2016.
103 f.

Orientador: Armando Miguel Awruch.

Dissertação (Mestrado) -- Universidade Federal do
Rio Grande do Sul, Escola de Engenharia, Programa de
Pós-Graduação em Engenharia Civil, Porto Alegre, BR-
RS, 2016.

1. Dinâmica dos Fluidos Computacional. 2. Equações
de Águas Rasas. 3. Método dos Elementos Finitos. 4.
Controle Ótimo. I. Awruch, Armando Miguel, orient.
II. Título.

MALÚ GRAVE

**SIMULAÇÃO E CONTROLE DE ENCHENTES USANDO AS
EQUAÇÕES DE ÁGUAS RASAS E A TEORIA DO
CONTROLE ÓTIMO**

Esta Dissertação de Mestrado foi julgada adequada para a obtenção do título de MESTRE EM ENGENHARIA, Área de Concentração de ESTRUTURAS, e aprovada em sua forma final pelo professor orientador e pelo Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul.

Porto Alegre, 14 de Janeiro de 2016

Prof. Armando Miguel Awruch
Dr. pela COPPE/UFRJ
Orientador

Prof. Dr. Carlos Torres Formoso
Ph.D. pela University of Salford
Coordenador do PPGEC/UFRGS

BANCA EXAMINADORA

Profa. Adriane Prisco Petry (UFRGS)
Dra. pela PROMEC/UFRGS

Prof. Alexandre Luis Braun (UFRGS)
Dr. pela PPGEC/UFRGS

Profa. Flávia Schwarz Franceschini Zinani (UNISINOS)
Dra. pela PROMEC/UFRGS

AGRADECIMENTOS

Agradeço ao professor Armando Miguel Awruch pela orientação, incentivo e conhecimento transmitido ao longo do desenvolvimento deste trabalho.

Ao Renato Linn, pelo auxílio para a realização deste trabalho e constante acompanhamento nas etapas de elaboração do mesmo.

Aos meus pais, Margrit e Lutero, pelo apoio, incentivo e todas as oportunidades que me concederam ao longo da minha vida.

Ao meu namorado e amigo Gabriel, pelo carinho diário, suporte e paciência durante o período de realização deste trabalho.

À minha prima Melina, pela ajuda e incentivo a continuar a carreira acadêmica.

Aos amigos e colegas, sem os quais o percurso para se chegar até aqui não seria tão agradável.

A todos os professores que de alguma forma contribuíram com a minha formação.

Ao meu gato Timo pelo companheirismo.

A todos que de alguma forma colaboraram para eu chegar até aqui.

RESUMO

GRAVE, M. Simulação e Controle de Enchentes Usando as Equações de Águas Rasas e a Teoria do Controle Ótimo. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 2016.

Esta dissertação tem por objetivo a implementação de um código para simular problemas hidrodinâmicos, bem como a possibilidade de controlar as elevações de onda resultantes numa determinada região por meio de uma vazão ótima controlada dentro do sistema estudado. O algoritmo implementado é baseado nas equações de águas rasas, as quais são aplicáveis em situações onde a altura d'água é de ordem muito menor do que as dimensões do sistema, que é discretizado espacial e temporalmente pelo Método dos Elementos Finitos e pelo método CBS (*Characteristic Based-Split*), respectivamente. O método de controle consiste na busca de uma curva de vazão de controle ótima que minimize a função objetivo, a qual compara os valores de altura d'água que se deseja encontrar em uma região especificada com os calculados pela simulação numérica. Para isso, utiliza-se um algoritmo evolutivo SCE-UA (*Shuffled Complex Evolution – University of Arizona*), que busca otimizar parâmetros de geração das curvas de vazão de controle, podendo estas serem modeladas por NURBS (*Non-Uniform Rational B-Splines*), que são capazes de encontrar a solução ótima, ou modeladas com curvas de forma triangular (linear) ou parabólica (quadrática) que apresentam uma solução aproximada de fácil implementação. Por fim, várias aplicações são realizadas, tanto para a simples simulação, quanto para o controle de problemas hidrodinâmicos, a fim de validar os algoritmos desenvolvidos e os resultados obtidos mostraram que os objetivos foram alcançados, encontrando uma forma eficiente de se fazer o controle de enchentes.

Palavras-chave: Dinâmica dos Fluidos Computacional; Equações de Águas Rasas; Método dos Elementos Finitos; Controle Ótimo.

ABSTRACT

GRAVE, M. Simulation and Flood Control Using the Shallow Water Equations and the Optimal Control Theory. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 2016.

Implementation of a computational code for the numerical simulation of hydrodynamic problems as well as the ability to control the resulting wave elevations in a specific area, using an optimal flow controlled within the studied system are the aims of this work. The implemented algorithm is based on the shallow waters equations, which are applicable in situations where the water height is much smaller than the system dimensions, and are spatial and temporally discretized by the Finite Element Method and the CBS method (Characteristic Based-Split), respectively. The control method consists in finding an optimal control flow curve that minimizes the objective function, which compares the objective value of water elevations in a specified region with those calculated by numerical simulation. An evolutionary algorithm called SCE-UA (Shuffled Complex Evolution - University of Arizona), which looks for optimize parameters of control flow curves generation, is used. These curves may be modeled by NURBS (Non-Uniform Rational B-Splines) which are able to find the optimal solution, or by curves of triangular (linear) or parabolic quadratic forms, which are an approximate solution easy to implement. Finally, several applications are performed for both simulation and control of hydrodynamic problems in order to validate the developed algorithms, and the results showed that the aims of this work were reached, finding an efficient way to control floods.

Key-words: Computational Fluid Dynamics; Shallow Water Equations; Finite Element Method; Optimal Control.

SUMÁRIO

1 INTRODUÇÃO	17
1.1 JUSTIFICATIVA.....	17
1.2 OBJETIVOS	19
1.3 METODOLOGIA	20
1.4 REVISÃO BIBLIOGRÁFICA	21
2 EQUAÇÕES DE ÁGUAS RASAS	24
2.1 EQUAÇÕES GOVERNANTES	25
2.2 CONDIÇÕES DE CONTORNO	28
2.3 DISCRETIZAÇÃO TEMPORAL: O MÉTODO CBS.....	29
2.4 DISCRETIZAÇÃO ESPACIAL	32
2.5 CAPACIDADE DE SIMULAR INUNDAÇÕES E SECAGEM	35
2.6 ESTABILIDADE NUMÉRICA	37
3 MÉTODOS DE OTIMIZAÇÃO	39
3.1 EVOLUÇÃO DOS COMPLEXOS MISTURADOS (SCE-UA)	42
3.2 APLICAÇÃO DO SCE-UA PARA OTIMIZAÇÃO DE FUNÇÕES	44
3.3 EQUAÇÕES DE ÁGUAS RASAS ACOPLADAS AO MÉTODO DE OTIMIZAÇÃO	47
4 GERAÇÃO DE CURVAS	48
4.1 CURVAS <i>B-SPLINES</i>	48
4.2 DEFINIÇÃO DE NURBS	50
4.3 GERAÇÃO DE CURVAS NURBS	51
4.4 OUTROS TIPOS DE CURVAS	53
5 TEORIA DO CONTROLE ÓTIMO APLICADA A ÁGUAS RASAS	56
5.1 OTIMIZAÇÃO DE CURVAS UTILIZANDO NURBS E SCE-UA	56
6 EXEMPLOS DE APLICAÇÃO DAS EQUAÇÕES DE ÁGUAS RASAS	60
6.1 ESCOAMENTO EM CANAL COM TRECHO DE ELEVAÇÃO PARABÓLICA DO FUNDO	60
6.2 SALTO HIDRÁULICO DEVIDO À PAREDE OBLÍQUA	62
6.3 ROMPIMENTO PARCIAL DE BARRAGEM	64
6.4 ROMPIMENTO DE BARRAGEM EM TERRENO COM TRÊS ILHAS CÔNICAS	66
6.5 ROMPIMENTO PARCIAL DE BARRAGEM EM TERRENO INCLINADO.....	70
6.6 TIGELA PARABÓLICA	72
6.7 INUNDAÇÃO EM ZONAS URBANAS	74

7 EXEMPLOS DE APLICAÇÃO DO ALGORITMO DE CONTROLE	80
7.1 CANAL RETANGULAR COM VAZÃO DE CONTROLE A JUSANTE	81
7.2 RESERVATÓRIO MORIYOSHIZAN	89
8 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	97
8.1 CONCLUSÕES	97
8.2 SUGESTÕES PARA TRABALHOS FUTUROS	98
REFERÊNCIAS	100

LISTA DE FIGURAS

Figura 1: Modelo de águas rasas	26
Figura 2: Distribuição das velocidades	27
Figura 3: Regiões de contorno de um domínio genérico	29
Figura 4: Elemento triangular de três nós	32
Figura 5: Operações utilizadas para formar novos indivíduos reais num espaço com duas variáveis de controle (y_1 e y_2)	41
Figura 6: Função Schwefel bidimensional	45
Figura 7: Função Rastrigin bidimensional	46
Figura 8: Computação triangular para funções de base <i>B-spline</i>	49
Figura 9: Geração de curva NURBS	52
Figura 10: Variação do peso w_2	52
Figura 11: Vazão de controle na forma linear	54
Figura 12: Vazão de controle na forma parabólica	55
Figura 13: Ajuste de curva com o método SCE-UA	57
Figura 14: Definição dos melhores parâmetros da curva com o método SCE-UA	59
Figura 15: Comparação entre solução numérica e analítica do nível d'água do canal com elevação parabólica de fundo	61
Figura 16: Comparação entre solução numérica e analítica da descarga do canal com elevação parabólica de fundo	62
Figura 17: Geometria do exemplo de salto hidráulico devido à parede oblíqua	63
Figura 18: Altura d'água do exemplo de salto hidráulico devido à parede oblíqua	63
Figura 19: Geometria do exemplo de rompimento parcial de barragem	64
Figura 20: Resultado do exemplo de rompimento de barragem em terreno seco em $t=7,2s$	65
Figura 21: Contornos de nível da profundidade d'água do exemplo de rompimento de barragem (seco) em $t=7,2s$	65
Figura 22: Resultado do exemplo de rompimento de barragem em terreno molhado em $t=7,2s$	66

Figura 23: Contornos de nível da profundidade d'água do exemplo de rompimento de barragem (molhado) em $t=7,2s$	66
Figura 24: Geometria do exemplo de três ilhas cônicas	67
Figura 25: Resultados três ilhas em $t = 2 s$	68
Figura 26: Resultados três ilhas em $t = 6 s$	68
Figura 27: Resultados três ilhas em $t = 12 s$	68
Figura 28: Resultados três ilhas em $t = 30 s$	69
Figura 29: Comparação dos resultados obtidos em $t = 12 s$ com Nikolos e Delis (2009)	69
Figura 30: Comparação dos resultados obtidos em $t = 12 s$ com Liang e Borthwick (2009)	70
Figura 31: Rompimento parcial de barragem com terreno inclinado	71
Figura 32: Resultados do exemplo de rompimento parcial de barragem com terreno inclinado	71
Figura 33: Resposta analítica e numérica do exemplo da tigela parabólica	73
Figura 34: Resultados da tigela parabólica em $t=1756,5$ encontrados por Bunya et al. (2009)	74
Figura 35: Inundação em zonas urbanas: geometria	76
Figura 36: Inundação em zonas urbanas: malha	76
Figura 37: Inundação em zonas urbanas: estações de análise	77
Figura 38: Inundação em zonas urbanas: resultados nas estações de análise	78
Figura 39: Inundação em zonas urbanas: simulação aos 3 s em 2D	79
Figura 40: Inundação em zonas urbanas: simulação aos 3 s em 3D	79
Figura 41: Fluxograma do processo de otimização da curva de controle	80
Figura 42: Exemplo 1 de controle	81
Figura 43: Fluxo de entrada do exemplo 1 de controle	81
Figura 44: Vazão de controle do exemplo 1 de controle	83
Figura 45: Altura d'água nos pontos objetivos do exemplo 1 de controle	84

Figura 46: Comparação entre alturas d'água do exemplo 1 com controle e sem controle	84
Figura 47: Vazão de controle do exemplo 1 de controle na forma ótima linear	87
Figura 48: Comparação das curvas de controle do exemplo 1.....	87
Figura 49: Comparação da altura d'água nos pontos objetivos para cada tipo de curva de controle do exemplo 1	88
Figura 50: Geometria do exemplo 2 de controle	89
Figura 51: Vazão de entrada do exemplo 2 de controle	90
Figura 52: Suavização dos resultados do exemplo 2 de controle no ponto A	91
Figura 53: Vazão de controle do exemplo 2 de controle	92
Figura 54: Comparação de altura d'água no ponto A aplicando as vazões de controle ótima encontradas por Kawahara e Shimada (1994) e o algoritmo desenvolvido	93
Figura 55: Possíveis vazões de controle do exemplo 2 de controle	95
Figura 56: Legenda para as figuras 54, 55 e 56	95
Figura 57: Comparação da altura d'água no ponto A do exemplo 2 de controle	95
Figura 58: Comparação da altura d'água no ponto B do exemplo 2 de controle	96
Figura 59: Comparação da altura d'água no ponto C do exemplo 2 de controle	96

LISTA DE TABELAS

Tabela 1: Pontos de controle de uma curva NURBS	51
Tabela 2: : Coordenadas dos pontos de controle da curva que representa a vazão ótima do exemplo 1 de controle	83
Tabela 3: Coordenadas dos pontos de controle da curva que representa a vazão ótima do exemplo 2 de controle.....	92
Tabela 4: Comparação das funções objetivo entre formas de curva de controle do exemplo 2	94

LISTA DE SIGLAS

WRI – *World Resources Institute*

CBS – *Caracteristic Based-Split*

NURBS – *Non-Uniform Rational B-Splines*

AG – *Algoritmos Genéticos*

SCE-UA - (*Shuffled Complex Evolution – University of Arizona*)

LISTA DE SÍMBOLOS

Para equações de águas rasas:

u_i – componentes do vetor de velocidades (m/s);

x_i – coordenadas cartesianas (m);

t – tempo (s);

ρ – densidade do fluido (kg/m^3);

p – pressão do fluido (m^3/s^2);

τ_{ij} – componentes do tensor cisalhante (m^3/s^2);

g_j – componente do vetor de gravidade (m/s^2);

h – altura total d'água (m);

U_i – velocidade média desenvolvida ao longo da profundidade u_i multiplicada pela altura total d'água h (m^2/s);

F_{ij} – a componente i do j -ésimo vetor fluxo (m^2/s);

Q_i – representa a i -ésima componente do vetor fonte (m^2/s);

H – altura d'água em relação ao leito medida a partir de um sistema de referência arbitrário horizontal (m);

η – elevação da superfície livre com relação ao nível previamente definido (m);

g – módulo da aceleração da gravidade (m/s^2);

C – coeficiente de Chèzy-Manning;

n – coeficiente de atrito;

r_i – forças de Coriolis (m/s^2);

f – coeficiente de Coriolis;

τ_i – forças de tração devido ao vento (m/s^2);

c_d – coeficiente de arrasto;

W_i – velocidade do vento na direção i (m/s);

c – celeridade da onda (m/s);

u_i^b – componente do vetor de velocidades no leito (m/s);

θ_1, θ_2 – parâmetros do método CBS;

ΔU_i^* – variável intermediária da variação de *momentum* do método CBS;

ΔU_i^{**} – variável intermediária da variação de *momentum* do método CBS;

N_i – funções de forma;

a_i – parâmetro da função de forma de um elemento triangular de 3 nós;

b_i – parâmetro da função de forma de um elemento triangular de 3 nós;

c_i – parâmetro da função de forma de um elemento triangular de 3 nós;

A – área do elemento triangular;

q – vazão por unidade de comprimento (m^2/s);

h_{min} – nível mínimo que um elemento possui quando considerado seco;

Z_b – topografia do fundo;

Φ – entidade a ser suavizada;

Φ_s – entidade suavizada;

α_s – parâmetro de suavização;

M_D – matriz M sem os termos não-diagonais.

Para o algoritmo SCE-UA:

G – centroide dos candidatos à solução com exceção daquele com pior valor de função objetivo;

u_q – indivíduo de pior aptidão;

r – reflexão;

e – expansão;

c^+ - contração positiva;

c^- - contração negativa;

p – número de complexos;

m – número de pontos em cada complexo;

D – conjunto formado por todos indivíduos em forma crescente de acordo com o valor da função objetivo;

A^k – conjunto formado pelos indivíduos de cada complexo k ;

f – valor da função objetivo

q – número de indivíduos num subcomplexo;

α' – número de vezes que se geram novos indivíduos nos subcomplexos;

β – número de vezes que se aplicam as operações na evolução dos complexos;

SCR – solução candidata de reflexão do algoritmo;

SCC – solução candidata de contração do algoritmo;

SCP – solução candidata pior do algoritmo;

SCA – solução candidata aleatória do algoritmo;

FO – função objetivo;

n – número de variáveis;

Para funções NURBS:

\mathbf{u} – vetor de nós de uma curva NURBS;

k – multiplicidade de um nó;

p em NURBS – grau da curva;

$N_{i,p}(u)$ – função de base de uma B-spline para os i pontos de controle e grau p ;

$C(u)$ – curva B-spline;

w_i – pesos de uma curva NURBS relativos aos i pontos de controle;

$Q(u)$ – curva NURBS.

Para teoria do controle ótimo:

J – função de performance;

η_{obj} – elevação da onda que se deseja encontrar;

x_{nurbs} – coordenada em x encontrada pela curva NURBS;

y_{nurbs} – coordenada em y encontrada pela curva NURBS.

1 INTRODUÇÃO

O início das grandes civilizações se deve ao estabelecimento das tribos nômades em regiões onde era possível se desenvolver a agricultura. Para isso, são necessários dois recursos básicos: terra e água. Onde houvesse água em abundância, haveria terra fértil, e é por isso que as grandes cidades foram erguidas em torno dos rios. Estes fornecem muitas funções essenciais para a vida da sociedade, fauna e flora, não apenas na agricultura ou o próprio consumo d'água, mas também na pecuária, pesca, transporte, energia elétrica, indústria.

Porém, a água, mesmo tão fundamental para a vida, começou a se mostrar uma ameaça quando participante de desastres ecológicos. Tempestades, tsunamis, rompimento de barragens ou diques, tornados, dentre outras situações que podem causar grandes enchentes, colocam em risco a vida das pessoas, como também causam significativas perdas econômicas e ambientais.

O grande crescimento das cidades e a urbanização desencadeou uma série de consequências que aumentaram os riscos de inundação. Várzeas começaram a ser ocupadas e populações vivem com constantes possibilidades de sofrerem prejuízos pelas cheias onde antes era feito o controle natural dos rios. O solo foi sendo impermeabilizado por camadas de concreto e asfalto, diminuindo a capacidade de infiltração d'água e aumentando o volume a ser escoado na superfície. Junto disso, há também o alto grau de desflorestamento, que está ocasionando grandes alterações climáticas, como evaporação e precipitações, derretimento de camadas polares, entre outros fenômenos já citados anteriormente.

Existem inúmeros motivos, tanto naturais como antrópicos, para que hajam inundações e enchentes no meio urbano, e, sendo os prejuízos tão grandes economicamente e em vidas humanas, uma série de estudos buscando formas mitigadoras de situações extremas começaram a serem feitos nessa área.

1.1 JUSTIFICATIVA

Recentes pesquisas do *World Resources Institute* (WRI) mostraram que, em decorrência de estragos causados por inundações, o Brasil arca com um custo anual estimado de US\$ 3

1 INTRODUÇÃO

O início das grandes civilizações se deve ao estabelecimento das tribos nômades em regiões onde era possível se desenvolver a agricultura. Para isso, são necessários dois recursos básicos: terra e água. Onde houvesse água em abundância, haveria terra fértil, e é por isso que as grandes cidades foram erguidas em torno dos rios. Estes fornecem muitas funções essenciais para a vida da sociedade, fauna e flora, não apenas na agricultura ou o próprio consumo d'água, mas também na pecuária, pesca, transporte, energia elétrica, indústria.

Porém, a água, mesmo tão fundamental para a vida, começou a se mostrar uma ameaça quando participante de desastres ecológicos. Tempestades, tsunamis, rompimento de barragens ou diques, tornados, dentre outras situações que podem causar grandes enchentes, colocam em risco a vida das pessoas, como também causam significativas perdas econômicas e ambientais.

O grande crescimento das cidades e a urbanização desencadeou uma série de consequências que aumentaram os riscos de inundação. Várzeas começaram a ser ocupadas e populações vivem com constantes possibilidades de sofrerem prejuízos pelas cheias onde antes era feito o controle natural dos rios. O solo foi sendo impermeabilizado por camadas de concreto e asfalto, diminuindo a capacidade de infiltração d'água e aumentando o volume a ser escoado na superfície. Junto disso, há também o alto grau de desflorestamento, que está ocasionando grandes alterações climáticas, como evaporação e precipitações, derretimento de camadas polares, entre outros fenômenos já citados anteriormente.

Existem inúmeros motivos, tanto naturais como antrópicos, para que hajam inundações e enchentes no meio urbano, e, sendo os prejuízos tão grandes economicamente e em vidas humanas, uma série de estudos buscando formas mitigadoras de situações extremas começaram a serem feitos nessa área.

1.1 JUSTIFICATIVA

Recentes pesquisas do *World Resources Institute* (WRI) mostraram que, em decorrência de estragos causados por inundações, o Brasil arca com um custo anual estimado de US\$ 3

Com os avanços tecnológicos, é possível desenvolver um modelo numérico computacional que represente adequadamente essas situações. Inundações podem ser simuladas pelas equações de águas rasas, que têm como característica principal a escala do seu comprimento ser muito maior do que a de sua profundidade. Nesse caso, a aceleração vertical pode ser negligenciada, e conseqüentemente, o campo de pressões vertical considerado hidrostático. Com base nas equações de conservação de quantidade de movimento e massa, pode-se resolver os problemas de hidrodinâmica para estes modelos.

As equações de águas rasas são amplamente utilizadas para simulações de inundações. Neste trabalho foi desenvolvido um código que faz o controle da altura do corpo d'água em determinado local, com o auxílio de uma região com fluxo de saída controlado para se chegar a esse resultado. Esse controle pode ser feito através da construção de reservatórios, aquedutos, barreiras móveis, comportas, entre outras soluções, que podem ser determinadas a partir da vazão necessária para chegar à solução ideal do problema. Essa é uma das formas de prevenir grandes inundações, fazendo o controle d'água que pode chegar à cidade.

1.2 OBJETIVOS

Os objetivos deste trabalho estão divididos em principais e secundários. Os objetivos principais são:

- a) desenvolver um algoritmo que resolva as equações de águas rasas, utilizando o método dos elementos finitos para discretização espacial, unido a uma discretização temporal explícita pelo método CBS (*Characteristic Based-Split*), capaz de lidar com topografias complexas e situações de inundação e secagem;
- b) desenvolver um algoritmo capaz de controlar enchentes por meio do controle de uma vazão alternativa ao curso natural do sistema em estudo, buscando a solução ótima por meio de diferentes curvas de vazão de controle.

E os objetivos secundários são:

- a) desenvolver um algoritmo capaz de gerar diferentes tipos de curvas, utilizando tanto a metodologia de curvas paramétricas NURBS, como também a geração de curvas de forma triangular (linear) e parabólica (quadrática), para estas serem aplicadas como curvas de controle a serem otimizadas;
- b) desenvolver um algoritmo de otimização evolutivo capaz de minimizar as mais complexas funções, com diversos números de variáveis, a fim de aplicá-lo no problema de otimização do proposto trabalho.

1.3 METODOLOGIA

É utilizada a linguagem Fortran 90/95 para implementar os objetivos citados. Foi desenvolvido um algoritmo para o cálculo das equações de águas rasas, onde foram implementados códigos para a aplicação do método dos elementos finitos com elementos triangulares lineares unidos a uma discretização temporal explícita pelo método CBS, capazes de resolver essas equações. Para a geração de malhas, utiliza-se um *software* auxiliar chamado GiD (2015), pelo qual são definidos vários parâmetros para o funcionamento correto do código. O algoritmo de águas rasas foi validado a partir da sua aplicação em vários testes disponíveis na bibliografia, podendo-se comparar os resultados aos de outros autores.

O algoritmo de águas rasas é então acoplado a um algoritmo de otimização evolutivo chamado SCE-UA desenvolvido por Duan *et al.* (1992). O que se busca otimizar é a curva ideal da vazão para que em uma área determinada a altura d'água tenha um comportamento pré-definido como objetivo. As curvas que representam a vazão de controle são geradas a partir de um terceiro algoritmo que pode gerar curvas NURBS (*Non-Uniform Rational B-Splines*), das quais as coordenadas y dos pontos de controle são os parâmetros variáveis para encontrar a solução ideal, bem como curvas lineares de forma triangular e quadráticas de forma parabólica através de uma formulação mais simplificada. Num primeiro momento, cada algoritmo é testado separadamente. O algoritmo de otimização pelo método SCE-UA é validado quando aplicado na busca por mínimos de funções com vários parâmetros a serem otimizados. Já o algoritmo de geração de curvas, tem os resultados comparados aos disponíveis na bibliografia sobre o assunto. Com todos os algoritmos funcionando corretamente, pode-se fazer o acoplamento destes, implementando uma forma de controle que funciona da seguinte forma:

- a) Geração de valores aleatórios para as variáveis de otimização: num primeiro momento o algoritmo de otimização gera valores aleatórios para as variáveis que se deseja otimizar, variáveis estas que são parâmetros para a geração de curvas;
- b) Geração de curvas: com os valores dos parâmetros de otimização definidos, geram-se curvas que representarão a vazão de controle a ser aplicada no algoritmo de águas rasas;
- c) Cálculo da função objetivo: cada curva gerada é aplicada como vazão de controle no algoritmo de águas rasas, e a partir dos resultados encontrados com essa situação de estudo, é calculada a função objetivo que depende das respostas encontradas pela aplicação desta;

- d) Otimização dos parâmetros: com o cálculo da função objetivo para cada grupo de variáveis aleatórias, é possível começar o processo evolutivo até se encontrar os parâmetros que representam a curva de controle ótima, ou seja, que minimize a função objetivo.
- e) Convergência: gerados os novos valores das variáveis de otimização a partir do processo evolutivo, volta-se para o segundo passo, até se chegar numa solução considerada satisfatória.

1.4 REVISÃO BIBLIOGRÁFICA

Sendo os prejuízos causados por inundações e enchentes cada vez maiores, estudos buscando formas mitigadoras, especificamente este caso de controle das cheias, vêm crescendo consideravelmente.

Em 2013, Ferreira (2013) fez uma abordagem numérica das equações de águas rasas utilizando o método dos elementos finitos e uma discretização temporal semi-implícita, na qual ele buscou encontrar o par de elementos adequado que represente a velocidade e a elevação da onda para calcular corretamente a solução discreta. No modelo de águas rasas utilizado, foi considerado que o fluido está disposto sobre uma superfície plana, cujo atrito é desprezível.

Já Pereira (2010) desenvolveu um modelo hidrodinâmico e de transporte de escalares, utilizando as equações de águas rasas, capaz de simular os padrões de circulação d'água e transporte de poluentes. Ele fez um estudo de caso do Rio Guaíba, localizado no estado do Rio Grande do Sul, que mostrou a potencialidade do modelo como instrumento de ajuda ao gerenciamento dos recursos hídricos e na avaliação dos impactos do lançamento de cargas poluidoras.

Em 1992, Kawahara *et al.* (1992), já estudavam formas de controle de problemas hidrodinâmicos utilizando as equações de águas rasas. Para a solução do método de controle utilizaram o método dos gradientes conjugados, que procura o menor valor para a função objetivo através da direção conjugada do gradiente dessa função, e o método Sakawa-Shindo, que modifica a função Hamiltoniana de forma a assegurar a estabilidade do problema de otimização. Nos seus primeiros estudos, analisaram a altura d'água em canais com diversas geometrias, onde o controle era feito por comportas na barragem a jusante. Concluíram que com os métodos utilizados para a otimização unidos a uma modelagem em elementos finitos

poderia se controlar a elevação d'água em todo sistema ajustando uma curva de saída de fluxo nessas comportas.

Em 1994, Kawahara e Shimada (1994) avançaram nestes estudos, comprovando que a duração do mecanismo de controle não precisa ser necessariamente a mesma duração do fluxo de entrada do domínio em estudo. Utilizaram também os mesmos métodos baseados no gradiente e Sakawa-Shindo para chegar a essa conclusão, e foi estudado o reservatório Moriyoshizan, no Japão. Foi controlado o fluxo de saída deste reservatório para que a elevação d'água em todo reservatório tivesse o mínimo de alteração com a entrada de uma vazão de água especificada.

Recentemente, em 2012, Samizo (2012) fez um estudo da barragem do lago Ikari no Japão, o qual se localiza perto do reservatório Kawaji. Existe uma área de *resort* localizada a jusante do lago Ikari, e em caso de grandes cheias essas áreas podem ser atingidas. Existe também um aqueduto que conecta os dois lagos e permite que o fluxo de água passe do reservatório Ikari para o Kawaji. É através deste aqueduto que se faz o estudo de controle, verificando qual a vazão necessária para que haja uma transferência de água de um lago para o outro, de forma que na região dos *resorts* não haja inundação. Para isso, Samizo, utilizou uma formulação adjunta de primeira ordem no qual o processo de controle foi feito por um método baseado em gradientes. A vazão de controle é então derivada de forma a otimizar a função objetivo utilizando os métodos já citados, e com os resultados encontrados, pode-se fazer um plano de prevenção de inundações na área estudada.

Sanders e Katopodes (1999) desenvolveram uma formulação adjunta sensitiva para o controle que também utiliza gradientes, e a aplicaram para realizar o controle de um canal por meio de uma saída lateral, podendo essa saída ser um reservatório, por exemplo.

Pode-se perceber que o método de controle utilizado pela grande maioria dos autores depende de derivadas para ser realizado, diferente do que é aplicado neste trabalho, onde é utilizado um método evolutivo. Não foram encontradas referências bibliográficas nas quais o controle de um problema de águas rasas foi realizado desta maneira. A vantagem deste procedimento em relação a um método de controle baseado em gradientes é que este não é sensível a imprecisões ocasionadas pela necessidade de se calcular derivadas, pois estas não são empregadas pelo algoritmo evolutivo.

Como pôde ser visto, vários estudos sobre esse assunto já foram realizados, e nesse trabalho se procura uma forma alternativa de se chegar a esses resultados. Além de buscar a curva ótima, ao se utilizar um método de otimização evolutivo, é possível definir a característica que terá a curva de vazão de controle, o que os métodos baseados em gradientes não conseguem fazer. Nem sempre será possível reproduzir na prática, com uma comporta, a vazão encontrada como solução ótima do sistema, visto que ela pode ter qualquer formato. Reproduzir em campo uma curva de vazão linear ou quadrática é de mais fácil aplicação, então, buscou-se encontrar as curvas de forma triangular (linear) e parabólica (quadrática) que otimizam o objetivo do problema, além da curva ótima, para assim se comparar os resultados obtidos.

2 EQUAÇÕES DE ÁGUAS RASAS

O trabalho é baseado na teoria de águas rasas, a qual descreve um escoamento incompressível de superfície livre onde a escala vertical (profundidade) tem dimensões muito menores do que a escala horizontal, ou seja, a extensão do corpo d'água. Neste capítulo serão apresentadas as equações que descrevem o comportamento d'água a partir dessa teoria, bem como os métodos e considerações utilizadas para desenvolver o algoritmo.

A teoria de águas rasas resulta da integração ao longo da profundidade das equações de Navier Stokes, as quais são derivadas das equações de conservação de massa e quantidade de movimento, como apresentado a seguir nas equações 1 e 2 (WIJAYA *et al.*, 2008):

Equação de conservação de massa:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1)$$

Equação de balanço de quantidade de movimento:

$$\frac{\partial u_j}{\partial t} + \frac{\partial}{\partial x_i} (u_i u_j) + \frac{1}{\rho} \frac{\partial p}{\partial x_j} - \frac{1}{\rho} \frac{\partial}{\partial x_i} \tau_{ij} - g_j = 0 \quad (2)$$

com i e $j = 1, 2$ e 3 .

Onde:

u_i : componentes do vetor de velocidades;

x_i : coordenadas cartesianas;

t : tempo;

ρ : densidade do fluido;

p : pressão do fluido;

τ_{ij} : componentes do tensor cisalhante;

g_j : componente do vetor de gravidade.

A equação de conservação de massa, também chamada de equação da continuidade, expressa a relação entre a massa acumulada no interior de um volume de controle com o fluxo, através do seu contorno. Já as equações de balanço de quantidade de movimento são derivadas da segunda Lei de Newton. Estudos mais aprofundados sobre as equações de conservação na Mecânica dos Fluidos podem ser encontrados em Schlichting (1979) e White (1991).

2.1 EQUAÇÕES GOVERNANTES

As equações de águas rasas na sua forma integrada ao longo da profundidade podem ser escritas conforme as equações 3 e 4 (ZIENKIEWICZ *et al.*, 2011):

$$\frac{\partial h}{\partial t} + \frac{\partial U_i}{\partial x_i} = 0 \quad (3)$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial F_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} + Q_i = 0 \quad (4)$$

onde $i, j = 1, 2$ e $U_i = hu_i$ é definido como a componente da velocidade média desenvolvida ao longo da profundidade u_i multiplicada pela altura total d'água h . As variáveis h e U_i são as incógnitas do problema hidrodinâmico. Os fluxos F_{ij} são definidos como $F_{ij} = hu_i u_j$, onde F_{ij} é a componente i do vetor de fluxo j , e a variável p (pressão) é dada pela equação 5 (MORANDI-CECCHI; VENTURIN, 2006):

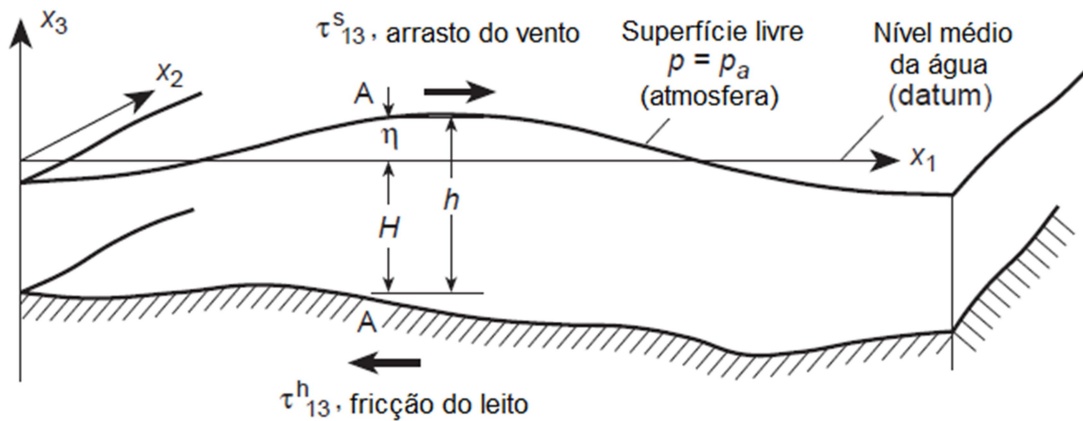
$$p = \frac{1}{2} g(h^2 - H^2) \quad (5)$$

sendo H o nível médio d'água em relação ao leito medida a partir de um sistema de referência arbitrário horizontal (nível de referência). As variáveis h e H se relacionam por meio de $h = H$

$+\eta$ (Figura 1), onde η é a elevação da superfície livre com relação ao nível previamente definido. O termo g indica a aceleração da gravidade e Q_i representa a i componente do vetor fonte definido pela equação 6:

$$Q_i = -g(h - H) \frac{\partial H}{\partial x_i} + \frac{g u_i |u|}{C^2 h} + r_i - \frac{\tau_i}{\rho} + \frac{h \partial p_a}{\rho \partial x_i} \quad (6)$$

Figura 1: Modelo de águas rasas



(fonte: adaptado de ZIENKIEWICZ *et al.*, 2011)

Estes termos surgem a partir da formulação baseada na integração ao longo da profundidade, sendo devidos: a) ao declive e à fricção do fundo, onde o coeficiente de Chezy-Manning é dado por $C = n^{-1}h^{1/6}$, sendo n o coeficiente de atrito; b) às forças de Coriolis, cujas componentes são $r_1 = -fU_2$ e $r_2 = fU_1$, e f é o coeficiente de Coriolis dado por $f = 2\Omega \sin\theta$ (onde θ é a latitude do fluido e $\Omega = 7 \times 10^{-5} \text{ s}^{-1}$ para a Terra); c) às forças de tração devidas ao vento $\tau_i = c_d \rho W_i^2$, onde c_d é um coeficiente de arrasto, ρ é a densidade do fluido e W_i a velocidade do vento na direção i ; e d) à pressão atmosférica definida como p_a . Por fim, a celeridade de onda c se relaciona com a elevação total d'água, para ondas longas, através da equação 7 (ORTIZ *et al.*, 2006):

$$c^2 = \frac{dp}{dh} = gh \quad (7)$$

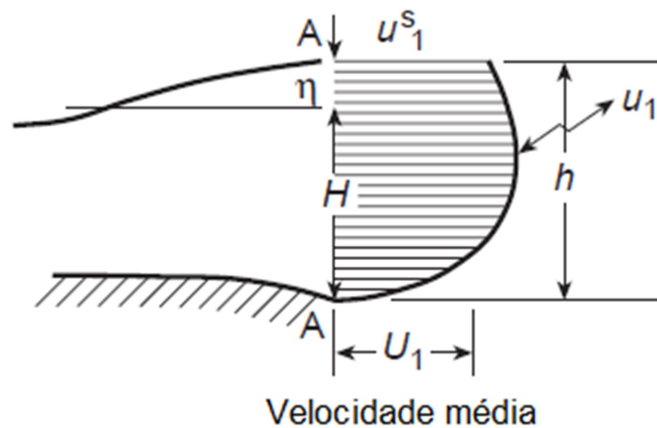
As premissas consideradas para chegar nessa formulação são (ZIENKIEWICZ *et al.*, 2011):

- a) para águas rasas, a profundidade tem dimensões muito menores do que a extensão do corpo d'água;
- b) a velocidade vertical u_3 é pequena e sua aceleração correspondente pode ser negligenciada;
- c) a pressão é considerada com distribuição hidrostática;
- d) assume-se que a topografia do fundo não varia com o tempo;
- e) assume-se que não ocorram escorregamentos no leito, $u_1^b = u_2^b = 0$, e, conseqüentemente, $u_3^b = 0$.
- f) as velocidades u_1 e u_2 são desconhecidas e não uniformes conforme apresentado na figura 2, então é conveniente introduzir a notação de velocidade média, definida na equação 8:

$$\int_{-H}^{\eta} u_i dx_3 = U_i(H + \eta) = U_i h \quad (8)$$

com $i = 1, 2$.

Figura 2: Distribuição das velocidades



(fonte: adaptado de ZIENKIEWICZ *et al.*, 2011)

Apesar de um escoamento de águas rasas ser praticamente horizontal, o que implica em muitas simplificações na formulação matemática e a possibilidade de se assumir uma pressão com distribuição hidrostática em simulações numéricas, esse tipo de escoamento não é exatamente bidimensional. Diferenças na temperatura ou salinidade, por exemplo, podem criar variações na densidade ao longo do eixo vertical. Na maioria dos casos, esses efeitos

podem ser desprezados, pois não são essenciais. Então, é suficiente considerar uma formulação que calcula as médias ao longo da profundidade, como é feito nessa teoria (WIJAYA *et al.*, 2008).

O desenvolvimento completo da dedução das equações de águas rasas pode ser visto no em Wijaya *et al.* (2008).

2.2 CONDIÇÕES DE CONTORNO

As equações de águas rasas precisam de condições iniciais e de contorno aplicadas para a sua resolução. As condições de contorno empregadas neste trabalho consistem em (ORTIZ *et al.*, 2006):

- a) prescrever os valores da elevação da onda $\bar{\eta}$, em uma porção do contorno Γ_{η} , tal como na equação 9;

$$\eta = \bar{\eta} \quad \text{em } \Gamma_{\eta} \in \Gamma \quad (9)$$

- b) prescrever os valores da vazão $\bar{q} = hu_i$, em uma porção do contorno Γ_q , tal como na equação 10;

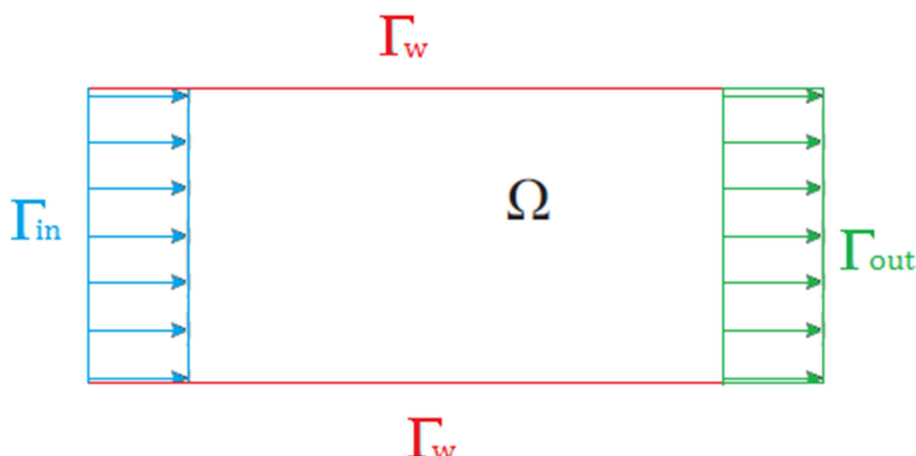
$$hu_i = \bar{q} \quad \text{em } \Gamma_q \in \Gamma \quad (10)$$

- c) condições de contorno de escorregamento em paredes sólidas totalmente reflexivas, tal como na equação 11.

$$U_i h . n_i = 0 \quad \text{em } \Gamma_w \in \Gamma \quad (11)$$

Em alguns casos, pode-se prescrever mais de uma condição de contorno daquelas descritas acima numa mesma fronteira, como, por exemplo, uma vazão associada a uma altura pré-determinada. Então, pode-se dividir os contornos como as características de um fluxo de entrada, Γ_{in} , as características de um fluxo de saída, Γ_{out} , e de parede sólida, Γ_w . Um domínio genérico é apresentado na figura 3 para representar os possíveis contornos disponíveis.

Figura 3: Regiões de contorno de um domínio genérico.



Existem também as condições de contorno para fronteiras abertas, que simulam a não reflexão do escoamento quando este chega nesse contorno. Esta condição não foi aplicada neste trabalho, no qual foi estendido o domínio em estudo para a sua representação.

As condições iniciais que devem ser fornecidas são a altura d'água em cada nó, bem como sua velocidade inicial e batimetria. Esses valores, unidos às condições de contorno, tornam o problema bem definido para dar início a uma simulação. Condições iniciais ou de contorno insuficientes podem levar o resultado da análise a soluções errôneas, ou, inclusive não convergir para nenhuma solução.

2.3 DISCRETIZAÇÃO TEMPORAL: O MÉTODO CBS

Os métodos CBS (*Characteristic-Based Split method*) têm sido muito utilizados em diversas áreas da mecânica dos fluidos e da mecânica dos solos. Desenvolvido por Zienkiewicz *et al.* (2011), este método tem sua base no procedimento dos passos fracionados introduzido por Chorin (1968 e 1969) e Temam (1969) para as equações incompressíveis de Navier-Stokes no contexto de diferenças finitas. A principal ideia do método consiste em tornar a equação de conservação de momento auto-adjunta, dentro de uma sequência de dois ou mais passos, que divide o problema numérico em subproblemas relativamente mais fáceis de serem resolvidos. O objetivo é que essa separação proposta forneça uma forma de calcular uma solução (U_i^{n+1}, p^{n+1}) no tempo t^{n+1} a partir dos valores (U_i^n, p^n) no tempo t^n . Aplicando-se esse método sobre o sistema de equações governantes de problemas de águas rasas, tem-se um sistema de equações auto-adjunto o qual pode ser discretizado de forma ótima pelo método de Galerkin (LINN, 2013).

A aplicação do método CBS nas equações de conservação é feita separando os termos convectivos e difusivos do problema. Inicialmente, se considera a equação da continuidade juntamente com a equação de quantidade de movimento a serem resolvidas para as variáveis u_i e p . O gradiente de pressão presente das equações governantes é considerado como um termo de fonte, e será calculado de forma independente no tempo t^{n+1} , com incremento de tempo $\Delta t = t^{n+1} - t^n$. As equações de conservação podem ser escritas conforme as equações 12 e 13 (ZIENKIEWICZ *et al.*, 2011):

$$\frac{1}{c^2} \frac{\Delta p}{\Delta t} + \frac{\partial U_i^n}{\partial x_i} + \frac{\partial(\Delta U_i)}{\partial x_i} = 0 \quad (12)$$

$$\frac{\Delta U_i}{\Delta t} = - \left[\frac{\partial F_{ij}}{\partial x_i} + Q_i \right]^n + \frac{\Delta t}{2} \left[u_k \frac{\partial}{\partial x_k} \left(\frac{\partial F_{ij}}{\partial x_j} + Q_i \right) \right]^n - \frac{\partial p^{n+\theta_2}}{\partial x_i} \quad (13)$$

onde:

$$\frac{\partial p^{n+\theta_2}}{\partial x_i} = (1 - \theta_2) \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial p^{n+1}}{\partial x_i} - (1 - \theta_2) \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial p^n}{\partial x_i} \right) \quad (14)$$

Então, as velocidades são calculadas dividindo a equação de quantidade de movimento resultante em duas partes, como mostrado na equação 15 (WIJAYA *et al.*, 2008).

$$\Delta U_i = \Delta U_i^* + \Delta U_i^{**} \quad (15)$$

onde:

$$\Delta U_i^* = \Delta t \left[-\frac{\partial F_{ij}}{\partial x_j} - Q_i \right]^n + \frac{\Delta t^2}{2} \left[u_k \frac{\partial}{\partial x_k} \left(\frac{\partial F_{ij}}{\partial x_j} + Q_i \right) \right]^n \quad (16)$$

e:

$$\Delta U_i^{**} = -\frac{\partial p^{n+\theta_2}}{\partial x_i} \quad (17)$$

Com $0 \leq \theta_1, \theta_2 \leq 1$, sendo $\theta_2 = 0$ para um esquema explícito e $\theta_2 > 0$ para esquemas semi-implícitos. Eliminando-se ΔU_i na equação da continuidade (equação 12), através do cálculo da divergência da equação 13 e substituindo-a na anterior, tem-se a seguinte equação auto-adjunta para a variável p :

$$\begin{aligned} & \left(\frac{1}{c^2} \right)^n \Delta p - \Delta t^2 \theta_1 \theta_2 \frac{\partial}{\partial x_i} \left(\frac{\partial (\Delta p)}{\partial x_i} \right) \\ & = -\Delta t \frac{\partial}{\partial x_i} [U_i^n + \theta_1 \Delta U_i^*] + \Delta t^2 \theta_1 \frac{\partial}{\partial x_i} \left(\frac{\partial p^n}{\partial x_i} \right) \end{aligned} \quad (18)$$

Observa-se que a variável intermediária ΔU_i^* é completamente determinada no tempo $t^n = n\Delta t$, enquanto que a pressão é calculada em $t^{n+\theta_2}$. Neste ponto, há pelo menos duas alternativas. Uma é eliminar a pressão num cálculo inicial para esta ser posteriormente introduzida, e, a segunda, utilizada neste trabalho, é omitir apenas o gradiente de pressão no início do passo de tempo, sendo introduzido num passo de tempo posterior dos cálculos (ZIENKIEWICZ *et al.*, 2011). Para se calcular p^{n+1} e U_i^{n+1} , no tempo $(n+1)\Delta t$, deve-se seguir os seguintes passos:

- a) resolver as equações de conservação de quantidade de movimento, omitindo o gradiente da pressão no início do passo, para calcular a variável intermediária ΔU_i^* (equação 16);

- b) calcular o incremento de massa $\Delta\rho$ ou de pressão Δp (equação 18);
- c) corrigir as velocidades ΔU_i^{**} para finalmente calcular o valor de U_i no tempo t^n (equação 17).

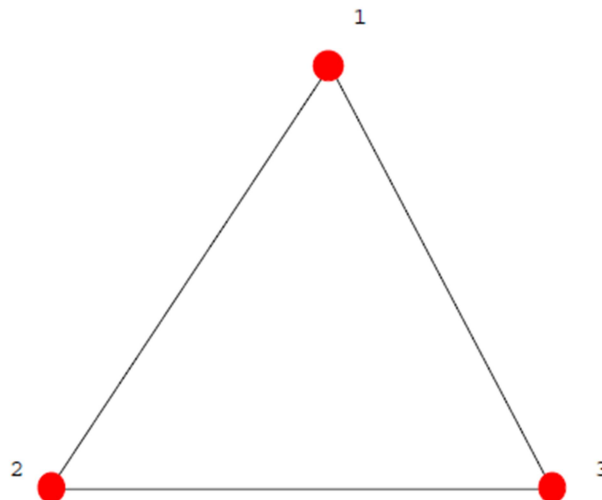
Como já foi mencionado, o método proporciona um operador auto-adjunto nas variáveis Δp (ou $\Delta\rho$) para uma discretização espacial do tipo Galerkin, e, portanto, tal discretização é empregada de forma otimizada.

2.4 DISCRETIZAÇÃO ESPACIAL

O método dos Elementos Finitos é um método de aproximação na análise de diversos fenômenos físicos que podem ocorrer num meio contínuo. Sua ideia principal consiste em subdividir o domínio do problema em questão em sub-regiões onde é possível realizar os cálculos de forma aproximada. Para isso, são empregados elementos que representam essas subdivisões, e, dependendo das suas funções de forma, alguns são considerados mais simples ou mais complexos, e a sua escolha e a do seu tamanho dentro da malha deve ser analisada para que a aproximação esteja representando adequadamente o comportamento do domínio analisado.

O elemento utilizado para fazer a discretização espacial das equações de águas rasas é o triângulo linear de três nós conforme apresentado na figura 4:

Figura 4: Elemento triangular de três nós.



Este é um elemento considerado simples, mas que é suficiente para representar o comportamento do problema em questão. As funções de forma N de um elemento triangular linear dependem das coordenadas cartesianas dos seus vértices (x_i, y_i) , e, juntamente com a área do triângulo, são apresentadas conforme as equações 19 e 20:

$$N_i = \frac{1}{2A}(a_i + b_i x + c_i y) \quad (i = 1,2,3) \quad (19a)$$

onde:

$$\begin{aligned} a_1 &= x_2 y_3 - x_3 y_2 & b_1 &= y_2 - y_3 & c_1 &= x_3 - x_2 \\ a_2 &= x_3 y_1 - x_1 y_3 & b_2 &= y_3 - y_1 & c_2 &= x_1 - x_3 \\ a_3 &= x_1 y_2 - x_2 y_1 & b_3 &= y_1 - y_2 & c_3 &= x_2 - x_1 \end{aligned} \quad (19b)$$

$$A = \frac{1}{2}(b_1 c_2 - b_2 c_1) = \frac{1}{2}(b_2 c_3 - b_3 c_2) = \frac{1}{2}(b_3 c_1 - b_1 c_3) \quad (20)$$

A partir disso, a seguinte aproximação espacial é aplicada (NITHIARASU, 2002):

$$U_i = N^l \tilde{U}_i \quad \Delta U_i^* = N^l \Delta \tilde{U}_i^* \quad \Delta U_i^{**} = N^l \Delta \tilde{U}_i^{**} \quad h = N^l \tilde{h} \quad (21)$$

Onde o til (\sim) acima da variável indica uma quantidade nodal, sendo N^l as funções de interpolação.

Assim, as equações que descrevem o método CBS apresentadas anteriormente podem ser discretizadas no espaço através do método dos elementos finitos, e ao escrever a pressão em termos da altura total d'água elas são reescritas da seguinte maneira (Ortiz *et al.*, 2006):

Passo 1: Equações de quantidade de movimento intermediárias

$$\int_{\Omega} N^l \frac{\Delta U_i^*}{\Delta t} d\Omega = \quad (22)$$

$$\left[- \int_{\Omega} N^l \frac{\partial F_{ij}}{\partial x_j} d\Omega - \int_{\Omega} N^l Q d\Omega - \frac{\Delta t}{2} \int_{\Omega} \frac{\partial}{\partial x_k} (N^l u_k) \frac{\partial F_{ij}}{\partial x_j} d\Omega + \right.$$

$$\left. \frac{\Delta t}{2} \int_{\Omega} N^l u_k \frac{\partial Q}{\partial x_k} d\Omega + \frac{\Delta t}{2} \int_{\partial\Omega} N^l u_k \frac{\partial F_{ij}}{\partial x_j} n_k d\Gamma \right]^n$$

onde Ω é o domínio do fluido delimitado por $\partial\Omega$, $i, j, k = 1, 2$ e n_k é a componente k da direção normal à fronteira $\partial\Omega$.

Passo 2: Equação da pressão: cálculo do incremento da altura d'água

$$\int_{\Omega} N^l \frac{\Delta h}{\Delta t} d\Omega + \Delta t \theta_1 \theta_2 \int_{\Omega} \frac{\partial N^l}{\partial x_i} \tilde{c}^2 \frac{\partial(\Delta h)}{\partial x_i} d\Omega = - \int_{\Omega} N^l \frac{\partial U_i^n}{\partial x_i} d\Omega + \quad (23)$$

$$\theta_1 \int_{\Omega} \frac{\partial N^l}{\partial x_i} \left(\Delta U_i^* - \Delta t (c^2)^n \frac{\partial h^n}{\partial x_i} \right) d\Omega - \theta_1 \int_{\partial\Omega} N^l \left[\Delta U_i^* - \right.$$

$$\left. \Delta t \left((c^2)^n \frac{\partial h^n}{\partial x_i} + \theta_2 \tilde{c}^2 \frac{\partial(\Delta h)}{\partial x_i} \right) \right] n_i d\Gamma$$

onde \tilde{c} é uma media da celeridade para todos passos de tempo (com $c^2 = gh$ para ondas longas) e $(c^2)^n$ é calculada no tempo t^n .

Passo 3: Correção das velocidades

$$\begin{aligned}
 \int_{\Omega} N^l \frac{u_i^{**}}{\Delta t} d\Omega = & \quad (24) \\
 -(1 - \theta_2) \int_{\Omega} N^l (c^2)^n \frac{\partial h^n}{\partial x_i} d\Omega - \theta_2 \int_{\Omega} N^l (c^2)^{n+1} \frac{\partial h^{n+1}}{\partial x_i} d\Omega - \\
 (1 - \theta_2) \frac{\Delta t}{2} \int_{\Omega} \frac{\partial N^l}{\partial x_k} u_k \left((c^2)^n \frac{\partial h^n}{\partial x_i} \right) d\Omega - \\
 (1 - \theta_2) \frac{\Delta t}{2} \int_{\Omega} N^l \frac{\partial u_k}{\partial x_k} \left((c^2)^n \frac{\partial h^n}{\partial x_i} \right) d\Omega (1 - \\
 \theta_2) \frac{\Delta t}{2} \int_{\partial\Omega} N^l u_k \left((c^2)^n \frac{\partial h^n}{\partial x_i} \right) d\Gamma
 \end{aligned}$$

O desenvolvimento completo dessas equações pode ser obtido em Zienkiewicz *et al.* (2011) e Ortiz *et al.* (2004).

2.5 CAPACIDADE DE SIMULAR INUNDAÇÕES E SECAGEM

Em diversas situações reais, têm-se problemas onde a fronteira entre a região molhada e a seca é não permanente, como no caso de inundações e passagem de rios e estuários. Portanto, tem-se a necessidade de disponibilizar ao código de águas rasas a capacidade de simular elementos tanto secos como molhados, e cuja situação muda com o tempo. Ao se representar essa condição numericamente, ela será relevante principalmente para os elementos que pertencem a essa fronteira entre o terreno inundado ou não, pois são nesses pontos em que um elemento pode vir a se tornar seco ou molhado, dependendo do fluxo do problema em questão.

Num problema deste tipo, podem surgir erros numéricos significativos que afetam a estabilidade da solução se não forem modelados adequadamente, e, por esse motivo, muitos modelos bidimensionais não permitem simulações que possuam esse tipo de condição (FROEHLICH, 2003). Devido à presença de elementos parcialmente molhados, há perda ou ganho de água no balanço de massa, tendo em vista que estes elementos são considerados

secos no processo de cálculo. Problemas no balanço de massa, juntamente com problemas de estabilidade, podem gerar erros que geralmente produzem altura d'água negativa, altas velocidades artificiais e grandes erros de conservação do volume (BRUFAU *et al.*, 2002).

Para a realização de uma simulação que permita topografia variável onde ocorram situações de secagem ou inundação, diversas técnicas podem ser encontradas na literatura, incluindo ajuste nas condições de fronteira (AKANBI, KATOPODES, 1988), modificação das equações governantes em áreas de água muito rasa (MESELHE; HOLLY, 1993) e uso de contornos móveis ou deformação de malha (LYNCH; GRAY, 1980).

Contornos móveis é uma técnica Lagrangeana, na qual a definição da malha no domínio é constantemente recalculada e gerada para se adaptar ao domínio molhado da simulação. Essa técnica utiliza algoritmos bastante complicados para a geração de malhas, sendo assim mais custosa computacionalmente. Já os outros dois métodos utilizam uma malha Euleriana fixa, pela qual o algoritmo deve ser capaz de identificar quais áreas estão secas ou inundadas. Dentre estas, o método utilizado neste algoritmo é explicitado a seguir.

As equações de águas rasas não podem ser resolvidas diretamente caso o nível total seja nulo ($h = 0$). Para isto, distinguem-se dois tipos de nós: secos ($h = h_{min}$) e molhados ($h > h_{min}$), onde h_{min} é um nível baixo tomado como referência mínima e apenas nós molhados são levados em consideração pelo método na solução das equações. Ademais, distinguem-se dois tipos de elementos: secos (apresentam pelo menos 1 nó seco) e molhados (apresentam todos os nós molhados). Apenas elementos molhados são considerados no algoritmo. Existe também a possibilidade de um elemento seco vir a tornar-se molhado, quando um elemento seco possuir um elemento vizinho com um fluxo mínimo capaz de elevar o nível do elemento seco para além do nível mínimo h_{min} (MEDEIROS; HAGEN, 2012). Isto é verificado ao se calcular a média da altura d'água de dois dos três nós que formam um elemento seco; se esta for maior do que h_{min} , o elemento passa a ser considerado molhado, e é atribuído ao seus nós um valor de altura d'água igual a média dos valores encontrados nestes anteriormente.

De forma inversa, pode-se ter o efeito contrário, de secagem do elemento, quando o fluxo reduz o nível de um nó para menos do que o nível mínimo h_{min} . Assim, o elemento é considerado seco e não é considerado no cálculo das equações.

2.6 ESTABILIDADE NUMÉRICA

Para haver estabilidade numérica, existe um valor de passo de tempo máximo que deve ser utilizado em cada iteração das simulações de problemas de águas rasas. Se for utilizado um passo de tempo maior do que este valor limite, o algoritmo pode não convergir para um resultado, ou, ainda, apresentar resultados errôneos.

O algoritmo desenvolvido pode resolver as equações de forma explícita ou semi-implícita, dependendo da escolha dos valores de θ_1 e θ_2 , e o critério de estabilidade depende do esquema escolhido a ser aplicado. Como já mencionado, para um esquema explícito temos (WIJAYA *et al.*, 2008).

$$\frac{1}{2} \leq \theta_1 \leq 1 \quad \theta_2 = 0 \quad (25)$$

E para este caso, a estabilidade é obtida quando o incremento de tempo é menor do que o valor limite definido pela equação 26:

$$\Delta t \leq \Delta t_\sigma = \frac{a}{c + |u|} \quad (26)$$

Onde:

a : tamanho do elemento;

c : celeridade da onda;

$$|u| = \sqrt{U_i U_i}$$

Para o algoritmo deste trabalho é utilizada a forma explícita com $\theta_1 = 0,5$ e $\theta_2 = 0$ com precisão de segunda ordem no tempo para o cálculo das velocidades (esquema de Crank-Nicolson), enquanto que a pressão possui precisão de primeira ordem no tempo, embora seja função explícita do tempo. A respeito de informação, na forma semi-implícita, as quantidades de θ_1 e θ_2 são definidas conforme apresentadas na equação 27:

$$\frac{1}{2} \leq \theta_1 \leq 1 \qquad \frac{1}{2} \leq \theta_2 \leq 1 \qquad (27)$$

e os três limites que devem ser considerados para assegurar esta estabilidade são:

$$\Delta t \leq \Delta t_\sigma = \frac{a}{|u|} \qquad \Delta t \leq \Delta t_\nu = \frac{a^2}{2\nu} \qquad \Delta t \leq \frac{\Delta t_\sigma \Delta t_\nu}{\Delta t_\sigma + \Delta t_\nu} \qquad (28)$$

onde ν é a viscosidade cinemática.

Calculado o passo de tempo mínimo necessário em cada interação, é aplicado ainda um coeficiente que reduz este valor, com a finalidade de aumentar a segurança no que se trata de estabilidade numérica. Os valores aplicados neste trabalho foram entre 20 e 5% do valor calculado.

Com essas condições definidas, é possível implementá-las no algoritmo das equações de águas rasas e após validá-lo, passar para a etapa de controle, para a qual é importante entender o funcionamento do método de otimização evolutivo utilizado e o método de geração de curvas.

3 MÉTODOS DE OTIMIZAÇÃO

Otimização é a busca do valor de uma ou n variáveis que proporcionem o melhor resultado para uma função dependente dessas variáveis, alcançando um objetivo desejado enquanto cumpre restrições impostas ao problema. Neste trabalho, a função a ser otimizada está integrada às equações de águas rasas, onde se busca alcançar um objetivo que depende da solução destas.

Para se fazer o controle de inundações, foi definida a utilização de um algoritmo de otimização evolutivo, diferente da maioria dos métodos utilizados na bibliografia para resolver esse tipo de questão, com intuito de se obter uma forma alternativa ao que já foi pesquisado. Métodos evolutivos são de natureza iterativa, os quais começam com uma avaliação inicial de uma função objetivo, e buscam melhorar esse valor de maneira sequencial, fazendo operações até encontrar a solução ótima.

Os métodos evolutivos conseguem trabalhar tanto com codificação contínua como discreta das variáveis, e não precisam que a função seja diferenciável, conseguindo encontrar mínimos globais em um amplo domínio de busca. Pode-se dizer que algoritmos evolutivos são eficientes para otimizar a maioria dos problemas existentes.

Um método evolutivo bastante interessante são os chamados Algoritmos Genéticos (AG), que são métodos computacionais de otimização inspirados nos mecanismos da evolução natural e genética, propostos por Darwin em 1858. Eles combinam os conceitos de adaptação seletiva, troca de material genético e sobrevivência dos indivíduos mais capazes, seguindo a linha da seleção natural.

Sendo propostos inicialmente por Holland (1975) em 1975, foi nos anos 80 que seu aluno Goldberg (1989) conseguiu os primeiros sucessos em aplicações industriais de AG. Desde então, estes algoritmos vêm sendo modificados e aplicados com sucesso nos mais diversos problemas de otimização.

Adotando a nomenclatura utilizada na genética, seguem algumas definições relacionadas aos AG (GOLDBERT, 1989):

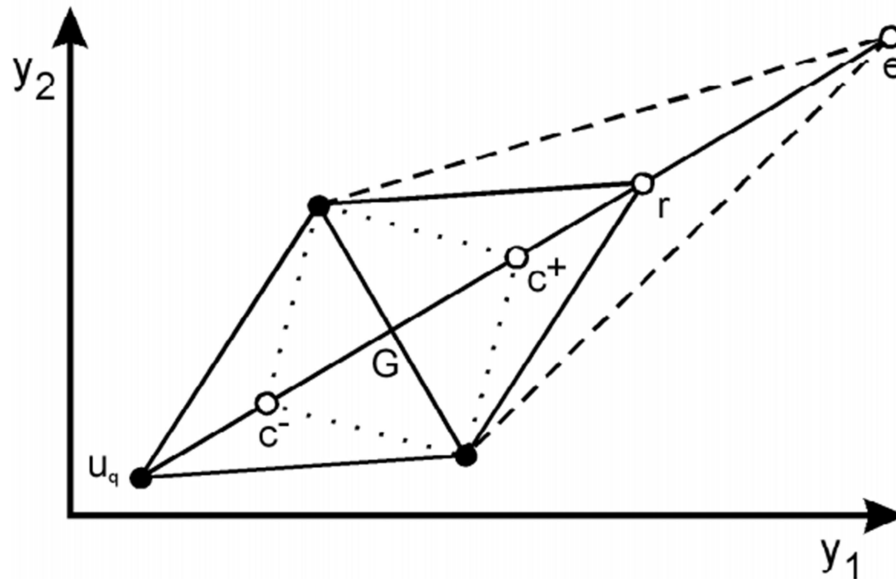
- a) cromossomo (indivíduo): cadeia de caracteres representando alguma informação relativa às variáveis do problema. Deste modo, cada cromossomo representa uma possível solução do problema;
- b) gene: é a unidade básica do cromossomo. Cada cromossomo tem um certo número de genes, cada um descrevendo uma variável;
- c) população: conjunto de cromossomos;
- d) geração: número de iterações que o algoritmo executa;
- e) operações genéticas: operações que o algoritmo realiza sobre cada cromossomo.

O funcionamento desse processo de otimização se inicia com a geração de uma população aleatória, na forma de cromossomos, formada por um conjunto de indivíduos que podem ser vistos como possíveis soluções do problema. Acontece, então, o processo evolutivo, onde esta população é avaliada e cada indivíduo recebe um índice que reflete sua adaptabilidade a um determinado ambiente. Uma porcentagem dos indivíduos mais adaptados é mantida, enquanto os outros são descartados. Aqueles mantidos pela seleção podem sofrer modificações em suas características fundamentais, gerando descendentes para a próxima geração. Esse processo chamado de reprodução é repetido até o número de gerações que forneça uma solução satisfatória (DIMCIC, 2011).

Para dar início a esse processo, é necessária uma codificação para fazer a transformação das variáveis do problema em cromossomos, e assim estes serem manipulados adequadamente. Para isso, existem dois tipos de codificação dos parâmetros de algoritmos genéticos: puros e reais. Os puros utilizam a representação binária dos valores dos parâmetros. Nessa codificação cada cromossomo é um vetor composto por zeros e uns, com cada bit representando um gene. As operações utilizadas para a formação de novos indivíduos desta codificação são o cruzamento (*crossover*) e a mutação.

Já a codificação real utiliza os valores dos genes de cada indivíduo diretamente para gerar novos indivíduos dentro do espaço de validade do domínio. As operações utilizadas para a formação de novos indivíduos podem ser os mecanismos de reflexão, expansão, contração positiva e contração negativa. Na figura 5, estão representados esses processos no espaço bidimensional, ou seja, de duas variáveis de controle. Uma situação de mais variáveis é de difícil representação geométrica, mas a abordagem matemática básica é a mesma a que a apresentada a seguir (WATANABE *et al.*, 2003).

Figura 5: Operações utilizadas para formar novos indivíduos reais num espaço com duas variáveis de controle (y_1 e y_2)



(fonte: WATANABE *et al*, 2003).

onde:

- a) reflexão: definida por $r = 2G - u_q$, onde G é o vetor centroide dos indivíduos que formam o triângulo principal, sendo u_q o indivíduo de pior aptidão;
- b) expansão: definida por $e = 3G - 2u_q$;
- c) contração positiva: definida por $c^+ = (3G - u_q)/2$;
- d) contração negativa: definida por $c^- = (G + u_q)/2$.

Ademais, é definida uma mutação gerada aleatoriamente dentro do domínio de busca. Vale ressaltar que algumas dessas opções de busca podem não serem utilizadas pelo método de otimização adotado e nem todas elas precisam ser implementadas em um algoritmo.

Dada essa introdução sobre os métodos evolutivos e a base do funcionamento de Algoritmos Genéticos, será descrito resumidamente o algoritmo utilizado neste trabalho, chamado SCE-UA (*Shuffled Complex Evolution – University of Arizona*), que pode ser traduzido como Evolução dos Complexos Misturados. Esse algoritmo opera com uma população de pontos que evolui em direção ao resultado ótimo por meio de sucessivas iterações e avaliações da função objetivo.

3.1 EVOLUÇÃO DOS COMPLEXOS MISTURADOS (SCE-UA)

O algoritmo SCE-UA foi inicialmente descrito por Duan *et al.* (1992), combinando técnicas de busca aleatória, Algoritmos Genéticos e otimização local. Ao analisar a superfície de resposta de funções objetivo aplicadas a modelos hidrológicos, Duan *et al.* (1992) constataram que, por apresentar um grande número de mínimos locais, os métodos de calibração automática de busca local não conseguiam encontrar o mínimo global, e assim, desenvolveram esse método alternativo.

O algoritmo SCE-UA imita o processo de seleção natural. Durante o procedimento, é gerada uma amostra de pontos espalhados no espaço hiperdimensional que define um domínio onde os parâmetros são considerados válidos. Cada ponto representa uma solução candidata e constitui a população de uma espécie. Esta população pode estar dividida em diversas comunidades, que são os complexos, e em cada complexo a população pode evoluir de forma independente das outras. Após um determinado período, as comunidades são misturadas e uma nova divisão é feita. Este processo permite o compartilhamento de informações que cada comunidade adquiriu separadamente (SOROOSHIAN; GUPTA, 1995).

Cada um dos membros de uma comunidade é candidato a participar do processo de reprodução. Para garantir que o processo seja competitivo, é necessário que os indivíduos com maior aptidão tenham maior probabilidade de serem escolhidos para participar desse processo. Sendo assim, a aptidão dos indivíduos é avaliada pelo valor da função objetivo: quanto menor este valor, maior será a aptidão, visto que este algoritmo foi desenvolvido para problemas de minimização.

Ao longo do procedimento, a amostra de pontos evolui em conjunto, até que os pontos se concentrem em uma região muito pequena, ou até mesmo sobre um ponto no espaço, em que está localizado o resultado ótimo. Quando isso ocorrer, pode-se dizer que o resultado convergiu para uma solução.

Os passos do algoritmo, conforme descrito por Duan *et al.* (1992), são apresentados a seguir:

1) **Início**: são selecionados os valores dos parâmetros p e m , que definem respectivamente, o número de complexos e o número de pontos em cada complexo. Sendo n o número de variáveis de otimização, os valores destes parâmetros devem ser tais que: $p \geq 1$ e $m \geq n + 1$. O tamanho da amostra s é igual ao produto de p e m ;

2) Geração da população: são gerados s grupos de parâmetros do modelo de simulação, entre os limites máximo e mínimo definidos para cada parâmetro. Esses conjuntos geralmente são definidos de forma aleatória, utilizando uma distribuição de probabilidade uniforme, mas é possível também indicar parâmetros próximos a uma resposta que se espera para agilizar o processo evolutivo;

3) Hierarquização dos indivíduos: os indivíduos gerados são avaliados com a função objetivo e reorganizados em ordem crescente, formando o conjunto D , de maneira que o primeiro ponto apresente o menor (melhor) valor;

4) Divisão em complexos: a população D é dividida em p complexos, cada um com m indivíduos. A divisão é feita de tal forma a garantir que cada complexo esteja integrado por certo número de indivíduos com boa aptidão e outros com má aptidão. A divisão em complexos segue a regra $A^k = \{x_j^k, f_j^k\}$ onde $x_j^k = x_{k+p(j-1)}$ e $f_j^k = f_{k+p(j-1)}$, onde k é o número do complexo que varia de 1 a p , j é o número dentro do complexo que varia de 1 a m , x é o grupo de parâmetros e f é o valor da função objetivo, sendo que x e f estão originalmente organizados numa amostra do tamanho s , isto é, x_j e f_j ($j = 1, \dots, s$);

5) Evolução dos complexos: cada complexo é submetido a um processo de evolução independente baseado no método de Evolução Competitiva Complexa, descrito por Duan *et al.* (1992) como segue:

- a) Início: são selecionados os valores para os parâmetros do SCE-UA q, α' e β . Os valores são escolhidos de forma que $2 \leq q \leq m, \alpha' \geq 1$ e $\beta \geq 1$, e é delimitado o hipercubo que irá conter os novos indivíduos;
- b) Atribuição de pesos: cada um dos indivíduos do complexo recebe um peso proporcional a sua posição hierárquica do complexo;
- c) Seleção dos pais: um subcomplexo de q indivíduos é escolhido de forma aleatória dentro do complexo, respeitando as probabilidades definidas em b);
- d) Geração de novos indivíduos: os indivíduos do subcomplexo (u_j para $j = 1, \dots, q$) são organizados de forma crescente de função objetivo e é calculado o centroide dos $q - 1$ melhores indivíduos. Em seguida, as coordenadas de um novo indivíduo são calculadas como segue, considerando o valor da função objetivo (FO) associada ao pior indivíduo (SCP) do subcomplexo, igual a FO(SCP);
 - i) Um novo indivíduo é criado por reflexão com base no centroide definido por $G = \frac{1}{q-1} \sum_{j=1}^{q-1} u_j$ e é chamado de Solução Candidata de Reflexão (SCR),

- ii) Se SCR estiver contido dentro do domínio é calculada a função objetivo FO(SCR) e o procedimento continua no passo iii. Caso contrário, é definido o menor hipercubo que contém todos os indivíduos do complexo e gerado um indivíduo de forma aleatória, denominado Solução Candidata Aleatória (SCA), no interior desse hipercubo. Esse é conhecido como o passo de mutação, onde se atribui o $SCR=SCA$ e $FO(SCR)=FO(SCA)$,
- iii) Se $FO(SCR) < FO(SCP)$, SCP é substituído por SCR e o procedimento continua no passo v. Caso contrário, é feito um procedimento de contração cujo novo indivíduo é chamado de Solução Candidata por Contração (SCC),
- iv) Se $FO(SCC) < FO(SCP)$, SCP é substituído por SCC e o procedimento continua no passo v. Caso contrário, novamente se gera um indivíduo de forma aleatória dentro do menor hipercubo que contém os indivíduos do complexo, SCA e se atribui $SCP=SCA$,
- v) Repetir os passos i-iv α' vezes.

Os passos b, c e d são repetidos β vezes.

6) Mistura dos complexos: os complexos são novamente agrupados na população D , e os indivíduos reorganizados, em ordem crescente de função objetivo;

7) Teste de convergência: Se o critério de convergência é satisfeito, o algoritmo se encerra. Caso o contrário, o procedimento retorna ao passo 3. Geralmente, para algoritmos evolutivos, o critério de convergência é um número de gerações que devem ser feitas. Toda vez que se chega na etapa 7, uma geração foi percorrida.

No algoritmo desenvolvido foram utilizados valores de $\alpha' = 1$ e $q = 3,4n + 1$ recomendados pelos autores do método como fixos, e o usuário pode variar o número de variáveis, número da população, os número de complexos, β e o domínio do problema em questão. Os autores também recomendam a utilização de um valor de $\beta = 2n + 1$.

3.2 APLICAÇÃO DO SCE-UA PARA OTIMIZAÇÃO DE FUNÇÕES

O algoritmo de otimização foi desenvolvido paralelamente ao programa de águas rasas. Assim, é possível aplicá-lo em diferentes problemas que não se limitam apenas a este trabalho, bem como facilita a sua verificação e funcionamento.

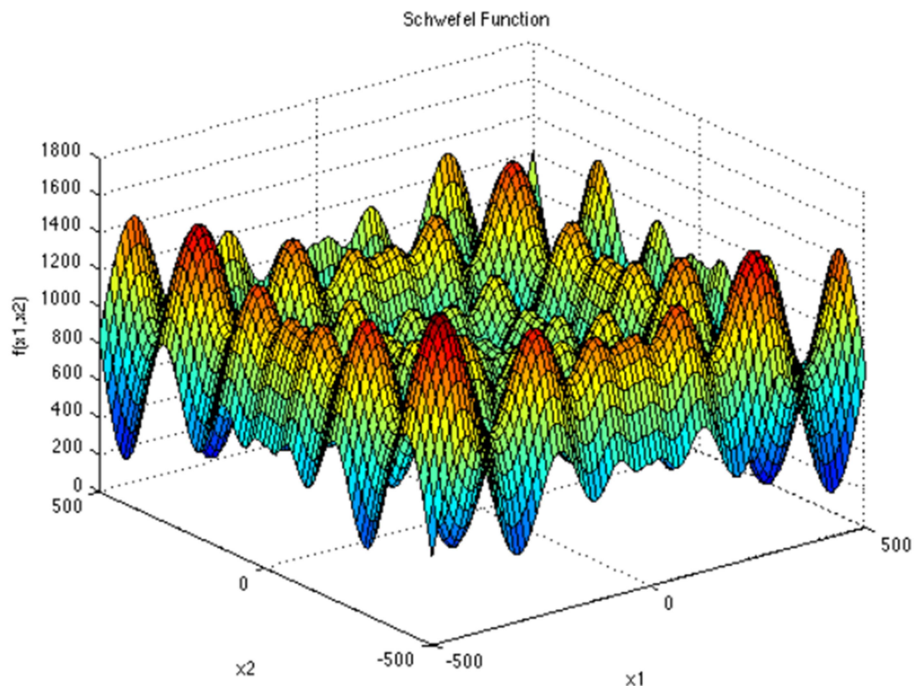
Para testar a aplicação do SCE-UA com o código desenvolvido, num primeiro momento foram otimizadas funções que possuem uma solução analítica conhecida. Buscou-se otimizar funções com um grande número de mínimos locais como as funções de Schwefel e de

Rastrigin apresentadas a seguir (VIRTUAL LIBRARY OF SIMULATION EXPERIMENTS, acesso em 2015).

A função Schwefel é multidimensional e possui muitos mínimos locais como pode ser visto na sua forma bidimensional apresentada na figura 6. A sua formulação é apresentada na equação 29 e esta foi testada até a dimensão $d = 10$, com dez variáveis a serem otimizadas. Utilizando uma amostra de 30 indivíduos divididos em apenas 1 complexo, um valor de $\beta = 400$, que é extremamente alto comparado ao recomendado pelos autores do método $(2n + 1)$ e dentro do domínio limitado pelo hipercubo $x_i \in [-500, 500]$, foram necessárias 16720 interações para se chegar ao mínimo global cuja resposta é 0, no ponto $x_i = 420,9687$ com $i = 1, 2, \dots, 10$.

$$f(x) = 418,9829 - \sum_{i=1}^d x_i \text{sen}(\sqrt{|x_i|}) \quad (29)$$

Figura 6: Função Schwefel bidimensional



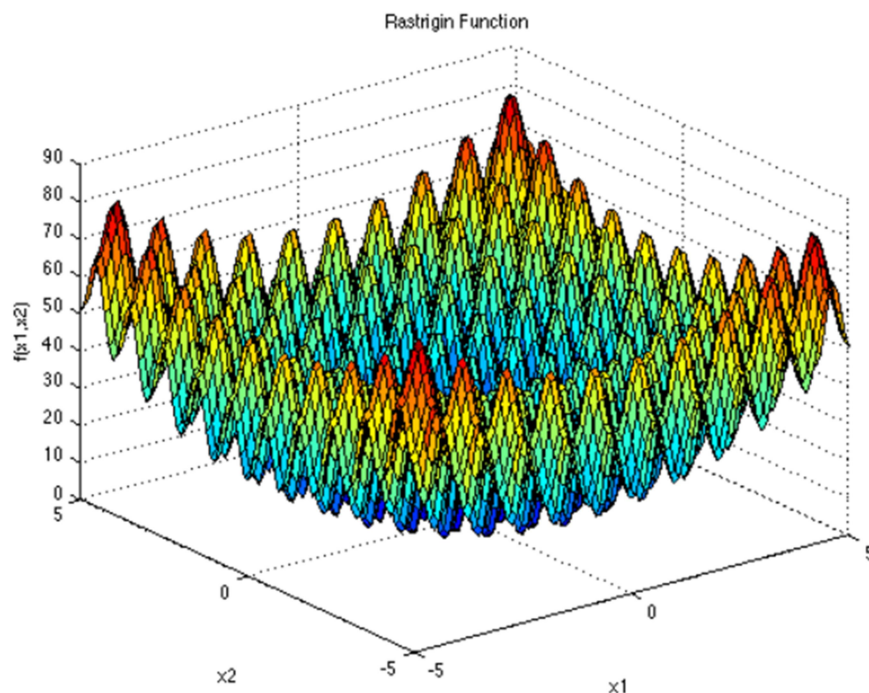
(fonte: VIRTUAL LIBRARY OF SIMULATION EXPERIMENTS, acesso em 2015).

Comparado a outras funções testadas, esta foi a que precisou de mais interações para convergir. Isto se explica pela quantidade de variáveis testadas, grande número de mínimos locais existentes, como também o amplo domínio de busca da solução, sendo um ótimo teste para algoritmos de otimização.

A função Rastrigin, por sua vez, possui comportamento semelhante, com vários mínimos locais e a característica de ser multidimensional. Porém, o domínio se limita a $x_i \in [-5.12, 5.12]$, o que fez o resultado de uma análise para 10 dimensões, com os parâmetros $m = 30, p = 1$ e $\beta = 75n$, convergir em apenas 84 interações, sendo o resultado do mínimo global igual a 0 para o ponto $x_i = 0$ com $i = 1, 2, \dots, 10$. A formulação da função Rastrigin pode ser vista na equação 30, e sua forma bidimensional está plotada na figura 7.

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)] \quad (30)$$

Figura 7: Função Rastrigin bidimensional



(fonte: VIRTUAL LIBRARY OF SIMULATION EXPERIMENTS, acesso em 2015).

A escolha dos parâmetros para cada exemplo se deu após vários testes, nos quais se concluiu que utilizar a menor população permitida (que depende do valor q definido para os subcomplexos), e quanto maior o número β (que faz com que haja maiores tentativas de melhorar a resposta), menor será o número de gerações necessário para o resultado convergir. Porém, estas condições dependem da função testada, visto que houve algumas exceções como, por exemplo, a própria função Schwefel apresentada. Ao se definir o número β igual a $75n$, como o que foi aplicado na função Ratrington, o algoritmo não conseguiu encontrar a resposta ótima num período de 50000 gerações.

Em geral, o algoritmo SCE-UA resolve a busca de mínimos de funções com pouquíssimas interações. Para alguns casos testados foram necessárias apenas 4 ou 5 gerações para se encontrar o resultado mínimo. Buscaram-se funções complexas como as apresentadas para testar o algoritmo em problemas que são realmente de difícil otimização, e para estes, os resultados também foram satisfatórios. Alterando a função objetivo, este método pode ser utilizado para diversos problemas de otimização, com uma vasta gama de aplicações em problemas de engenharia.

3.3 EQUAÇÕES DE ÁGUAS RASAS ACOPLADAS AO MÉTODO DE OTIMIZAÇÃO

O algoritmo evolutivo de otimização é empregado neste trabalho para o controle de enchentes e inundações através de uma região do domínio chamada de fronteira de controle, que vai determinar a evolução da vazão ao longo do tempo necessária para que em determinado local a altura d'água não ultrapasse um valor objetivo pré-definido. Para isso, serão modificados os parâmetros (variáveis de otimização) que definem uma curva *vazão x tempo*. Para essas curvas, o eixo y representa a vazão, enquanto que o eixo x representa o tempo. Essa curva poderá ser modelada como curva NURBS, metodologia que permite representar formas arbitrárias por meio de uma base matemática simples, possuindo algoritmos rápidos e numericamente estáveis, como também poderá ser modelada por uma metodologia mais simples, com curvas lineares de forma triangular e quadráticas de forma parabólica. Para essas curvas, o eixo y representa a vazão, enquanto que o eixo x representa o tempo. A metodologia de curvas NURBS e a definição dos seus parâmetros serão explicados no próximo capítulo.

4 GERAÇÃO DE CURVAS

Três formas de curva são empregadas neste trabalho para representar a vazão de controle: NURBS, lineares e quadráticas. As duas formas mais comuns de representar curvas em modelação geométrica são as equações implícitas e as funções paramétricas. As equações implícitas se situam no plano xy na forma $f(x, y) = 0$, onde a equação descreve uma relação entre as coordenadas x e y dos pontos da curva. Já na forma paramétrica, cada coordenada de um ponto da curva é representada separadamente com uma função explícita de um parâmetro independente (PIEGL; TILLER, 1997). Como exemplo, pode-se descrever um círculo de raio unitário centrado na origem, apresentado na forma implícita pela equação 31 e na forma paramétrica pela equação 32.

$$f(x, y) = x^2 + y^2 - 1 = 0 \quad (31)$$

$$\begin{aligned} x(u) &= \cos(u) \\ y(u) &= \sin(u) \end{aligned} \quad (32)$$

As curvas NURBS são um tipo de curvas paramétricas. Generalizações de curvas *B-splines* e Bézier, elas podem representar formas complexas com poucos dados. Curvas *B-splines* e Bézier não conseguem representar um círculo exato, como o exemplo citado acima, por serem curvas paramétricas polinomiais; já curvas NURBS são racionais e fazem precisamente essa representação.

4.1 CURVAS *B-SPLINES*

É importante entender as curvas *B-splines* para, a partir delas, se chegar nas curvas NURBS. Define-se u um conjunto de $m + 1$ números reais onde $u_1 \leq u_2 \leq u_3 \leq \dots \leq u_m$. Os u_i são chamados de nós e o conjunto \mathbf{u} de vetor de nós. Se os nós são igualmente espaçados, ou seja, $u_{i+1} - u_i$ é uma constante, o vetor de nós é dito uniforme, caso contrário, não uniforme. Um

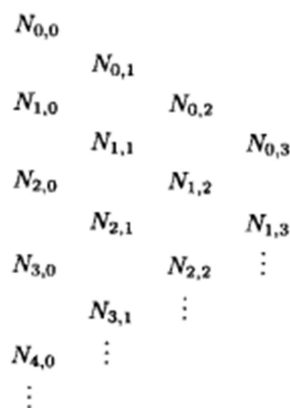
nó pode ser simples, quando ele aparece apenas uma vez no intervalo, ou ter multiplicidade k , ao se repetir k vezes. Os nós podem ser considerados pontos de subdivisão do domínio $[u_0, u_m]$ em períodos constituídos por dois nós. Além destes parâmetros, para definir as funções bases de uma B -spline é necessário definir o grau p dessas funções. A partir disso, a i -ésima função base B -spline de grau p , é dada por:

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq u \leq u_{i+1} \\ 0 & \text{caso contrário} \end{cases} \quad (33)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (34)$$

Se o grau é zero ($p = 0$), essas funções de base são todas funções de passo, isto é, a função de base $N_{i,0}(u)$ é igual a 1 apenas se u está no i -ésimo período constituído pelos nós $[u_i, u_{i+1})$. Para compreender o modo de computar $N_{i,p}(u)$ para p maior que 0, pode-se usar o esquema de computação triangular representado na figura 8 (PIEGL; TILLER, 1997).

Figura 8: Computação triangular para funções de base B -spline



(fonte: PIEGL; TILLER, 1997).

Além do vetor de nós u , uma curva B -spline também tem de ter $n + 1$ pontos de controle $p_0, p_1, p_2, \dots, p_n$, e a partir desses, uma curva B -spline de grau p é dada por:

$$C(u) = \sum_{i=0}^n N_{i,p}(u)P_i \quad (35)$$

onde $N_{i,p}(u)$ são as funções de base citadas anteriormente e P_i os pontos de controle. Os pontos P_i formam uma rede de controle e as funções de base *B-splines* são definidas sobre o vetor de nós \mathbf{u} , por:

$$\mathbf{u} = \left\{ \underbrace{0, \dots, 0}_{p+1}, \hat{u}_{p+1}, \dots, \hat{u}_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\} \quad (36)$$

Para definir uma curva *B-spline* de grau p , com $n + 1$ pontos de controle, será preciso fornecer $n + p + 2$ nós. Por outro lado, se definido um vetor de nós de $m + 1$ nós e $n + 1$ pontos de controle, o grau da curva será $p = m - n - 1$.

4.2 DEFINIÇÃO DE NURBS

Como já mencionado, curvas *B-splines* são polinomiais e apesar da sua flexibilidade e propriedades úteis para gerar curvas, elas não conseguem representar curvas simples como círculos e elipses. Círculos podem ser representados por funções racionais, e é assim, generalizando curvas *B-splines* para curvas racionais é que se chega às curvas NURBS.

Assumindo que os nós se encontram no intervalo de $[0,1]$, uma curva NURBS pode ser descrita como:

$$Q(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_iP_i}{\sum_{i=0}^n N_{i,p}(u)w_i} \quad (37)$$

onde w_i são os pesos vindos do conceito de coordenadas homogêneas. Se $w_i = 1$ para todos pontos de controle, a curva se resume a uma *B-spline*. Quanto maior o valor de w_i , mais a curva se aproximará do seu respectivo ponto de controle.

Concluindo, para se definir uma curva NURBS é necessário definir os pontos de controle, o vetor de nós, e o peso para cada ponto de controle. Para a otimização do problema deste trabalho, o grau da curva e as coordenadas no eixo x dos pontos de controle serão fixados, onde os nós serão distribuídos uniformemente sobre este eixo. Então, as variáveis de otimização serão as coordenadas no eixo y dos pontos de controle e os respectivos pesos, buscando assim a curva ideal de fluxo de saída da fronteira de controle de um problema de águas rasas.

4.3 GERAÇÃO DE CURVAS NURBS

Da mesma forma que para SCE-UA, foi desenvolvido um código para o cálculo de curvas NURBS independente do problema de águas rasas. Neste algoritmo o usuário informa o grau da curva e o número de pontos de controle com suas respectivas coordenadas e pesos, e a partir disso, o programa cria um arquivo de saída com os dados da curva resultante dessas condições fornecidas.

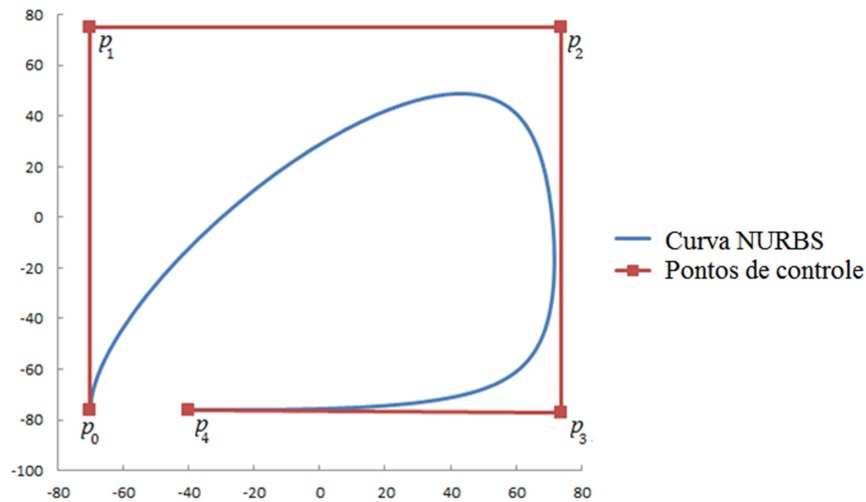
Como exemplo ilustrativo da aplicação do programa, foi gerada uma curva de grau 3, definida pelos 5 pontos de controle no plano x - y dados por:

Tabela 1: Pontos de controle de uma curva NURBS.

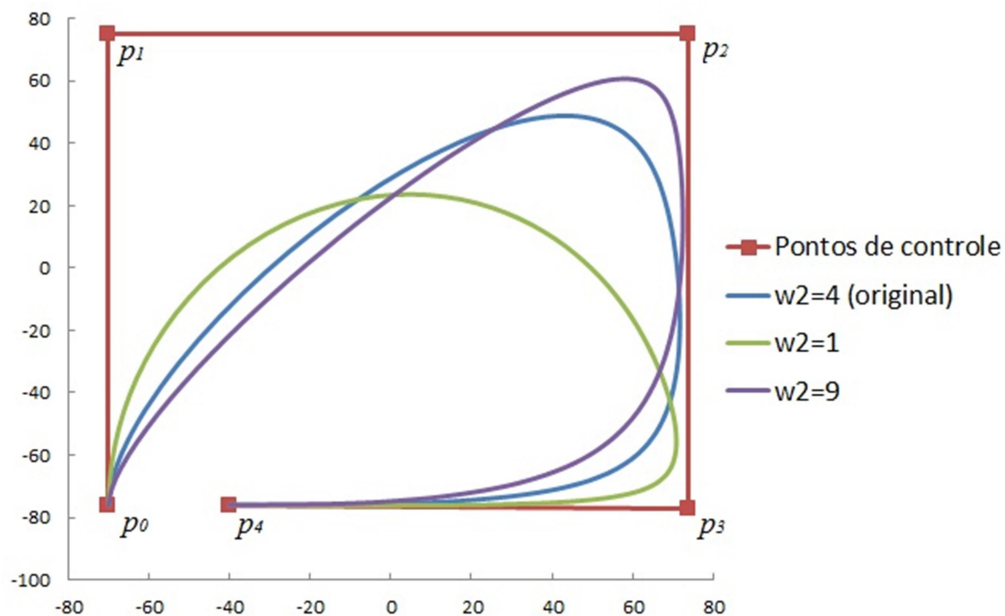
	X	Y	W
p₀	-70	-76	1
p₁	-70	75	0.5
p₂	74	75	4
p₃	74	-77	5
p₄	-40	-76	1

O resultado é apresentado na figura 9:

Figura 9: Geração de curva NURBS.



Este exemplo foi baseado numa curva apresentada por Minetto (2003), sendo uma das diversas curvas testadas para a validação do algoritmo desenvolvido para a geração das mesmas. Foram modificados parâmetros como grau da curva, coordenadas, pesos e número dos pontos de controle com intuito de ter a certeza de que o programa funciona corretamente. Para se entender a sensibilidade de uma curva NURBS quanto aos pesos avaliados em cada ponto de controle, a figura 10 apresenta novas curvas geradas a partir de diferentes valores atribuídos ao peso do ponto de controle p_2 .

Figura 10: Variação do peso w_2 .

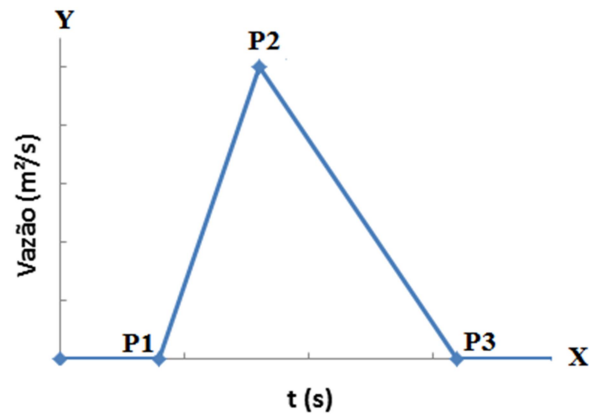
Pode-se notar que quanto maior o peso atribuído a um ponto de controle, mais a curva tende a se aproximar do mesmo. Com o peso $w_2 = 9$, a curva tendeu a se aproximar consideravelmente do ponto p_2 , principalmente quando comparada as outras curvas produzidas com pesos w_2 menores do que este. Outro aspecto relevante que pode ser observado, é que a variação do peso num único ponto pode causar uma mudança significativa na forma da curva, afetando, neste caso, completamente a geometria resultante. Neste exemplo, essa mudança geral da curva se deu devido aos poucos pontos de controle que a formavam. Num caso com mais pontos de controle, as mudanças feitas em um deles irão afetar apenas uma região da curva, e não ela por completo. Isso tudo depende do grau da curva e número de pontos de controle que a formam. Maiores informações sobre curvas NURBS e suas propriedades podem ser encontradas em Minetto (2003) e Piegl e Tiller (1997).

4.4 OUTROS TIPOS DE CURVAS

Uma vantagem da otimização por um algoritmo evolutivo é não se limitar a encontrar somente a curva ótima, e sim poder definir qual o tipo de curva que se deseja otimizar. Na prática, representar uma curva linear ou quadrática é muito mais simples de se fazer com a abertura e fechamento de uma comporta do que controlar esse sistema para uma curva arbitrária que pode surgir da otimização de NURBS. Então, buscou-se encontrar não apenas a curva ótima encontrada pela metodologia de uma curva NURBS, mas também aquelas encontradas pela otimização de parâmetros que encontram uma curva triangular (linear) e uma parábola (quadrática) que mais se aproximam do objetivo.

Para encontrar uma vazão de controle na forma de um triângulo, são necessários encontrar as coordenadas de 3 pontos, conforme ilustrado na figura 11. Sabe-se que para P1 e P3 a coordenada y é igual a zero. Então restam 4 parâmetros a serem encontrados e otimizados: a coordenada em x do P1, as coordenadas x e y do P2 e a coordenada em x de P3.

Figura 11: Vazão de controle na forma linear



As equações utilizadas para obter os valores da curva de forma analítica a partir das coordenadas desses três pontos a serem otimizados são:

$$\begin{aligned}
 y &= 0 && \text{para } x \leq x_1 \\
 y &= A_1 (x - x_1) && \text{para } x_1 < x \leq x_2 \\
 y &= A_2 (x - x_2) + y_1 && \text{para } x_2 < x \leq x_3 \\
 y &= 0 && \text{para } x > x_3
 \end{aligned} \tag{38}$$

onde:

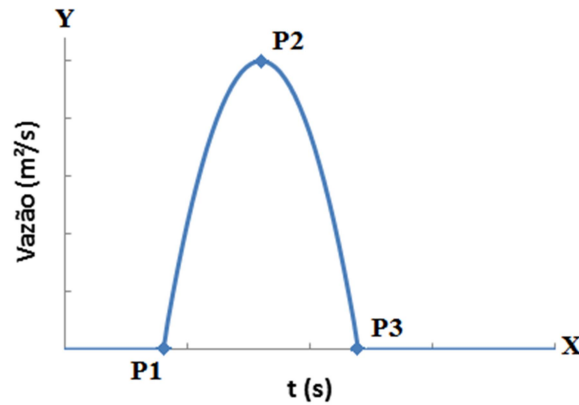
$$\begin{aligned}
 A_1 &= \frac{y_2 - y_1}{x_2 - x_1} \\
 A_2 &= \frac{y_3 - y_2}{x_3 - x_2}
 \end{aligned} \tag{39}$$

Com essas informações é possível começar o processo de otimização dos parâmetros, em busca da curva de controle de forma linear que minimiza o erro entre a altura d'água objetivo e aquela computada.

Outra forma de curva testada foi a parábola. Neste caso também precisam ser definidas as coordenadas de 3 pontos para encontrar a equação da curva conforme a figura 12. Como uma parábola é simétrica em relação ao eixo que passa pelo seu valor máximo, as coordenadas do ponto P3 podem ser encontradas a partir das coordenadas de P1 e P2. Ademais, a coordenada em y de P1 será sempre 0, então as coordenadas que precisam ser encontradas e,

consequentemente, otimizadas são: a coordenada em x do ponto P1 e as coordenadas em x e y do ponto P2.

Figura 12: Vazão de controle na forma parabólica



Definido os três pontos da curva, é possível calcular os parâmetros A , B e C de uma parábola da forma $Ax^2 + Bx + C$, e com essa formulação encontrar as coordenadas para cada instante conforme a equação 40.

$$\begin{aligned} y &= 0 && \text{para } x \leq x_1 \\ y &= Ax^2 + Bx + C && \text{para } x_1 < x \leq x_3 \\ y &= 0 && \text{para } x > x_3 \end{aligned} \quad (40)$$

Assim, além da busca pela curva ótima que deve ser encontrada com NURBS, tem-se duas opções alternativas que possuem menos variáveis de otimização, e são de fácil aplicação em campo ao se trabalhar com a abertura e fechamento de comportas.

Com os algoritmos de otimização, geração de curvas e cálculo de problemas de águas rasas funcionando, pode-se passar para a etapa de controle, que une em um só programa tudo que foi desenvolvido até então.

5 TEORIA DO CONTROLE ÓTIMO APLICADA A ÁGUAS RASAS

A proposta deste trabalho é apresentar formas de controlar possíveis enchentes em zonas urbanas por meio do controle ótimo de uma vazão alternativa ao curso natural do sistema em estudo, que faça com que o objetivo da altura d'água não ultrapassar um certo valor em dada região, se cumpra. Essa vazão alternativa se daria por meio de obras no curso d'água, seja a abertura de comportas, construção de reservatórios, aquedutos, entre outras formas de projeto que colocariam em prática o estudo do controle ótimo.

O problema de controle a ser aplicado neste trabalho é encontrar a vazão que minimize a função de performance (função objetivo) sobre as restrições das equações de águas rasas. A função de performance J é definida pela soma ao quadrado da diferença entre o valor computado e o valor objetivo da elevação d'água nos pontos marcados, e está apresentada na equação 41. Minimizar a função de performance J significa que a elevação d'água nos pontos marcados se torna a mais próxima possível da elevação d'água objetivo.

$$J = \frac{1}{2} \int_{t_0}^{t_f} \int_{\Omega} (\eta - \eta_{obj}) [\delta(x - x_o)(y - y_o)] (\eta - \eta_{obj}) d\Omega dt \quad (41)$$

onde $\delta(x - x_o)(y - y_o)$ é o delta de Dirac, sendo $\delta = 1$ para $x = x_o$ e $y = y_o$ e $\delta = 0$ para $x \neq x_o$ e $y \neq y_o$.

A função de performance é mínima quando se encontrar uma curva de vazão ótima nos pontos de controle. Essa curva de vazão é gerada por NURBS e é variando os parâmetros que geram esta curva que ela é otimizada pelo algoritmo SCE-UA. Este é o método utilizado neste trabalho para se fazer o controle ótimo em problemas de enchente em zonas urbanas.

5.1 OTIMIZAÇÃO DE CURVAS UTILIZANDO NURBS E SCE-UA

Antes de aplicar os algoritmos no problema de controle de águas rasas em si, é importante verificar o funcionamento da otimização por SCE-UA unida a geração de curvas NURBS.

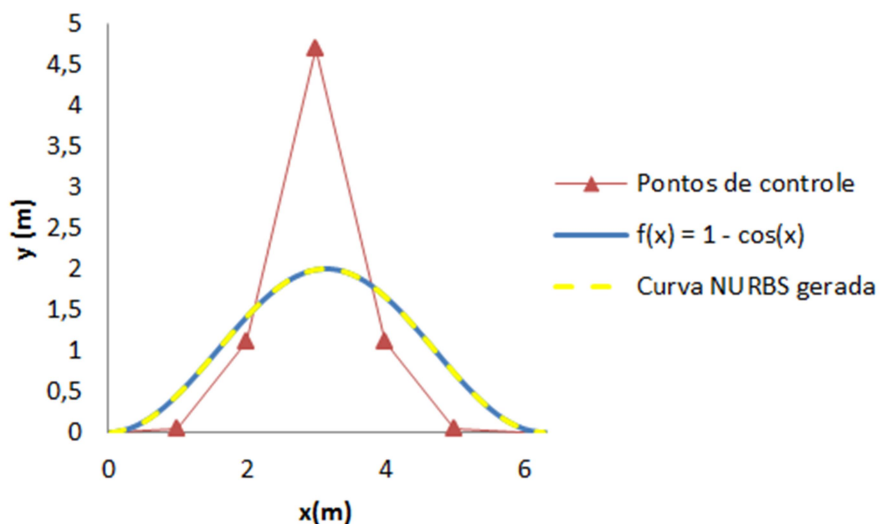
Para isso, um problema simples pode ser testado, o qual consiste num ajuste de uma curva cosseno do tipo $f(x) = 1 - \cos(x)$, por meio da otimização de uma curva NURBS. Neste problema, a função objetivo seria a diferença ao quadrado entre o valor exato da função cosseno com o valor computado pela curva gerada pelo programa naquele ponto (equação 42). A função objetivo será mínima quando a curva NURBS gerada se aproximar ao máximo dessa curva cosseno.

$$f = \int_{\Omega} [y_{nurbs} - (1 - \cos(x_{nurbs}))]^2 d\Omega \quad (42)$$

onde x_{nurbs} e y_{nurbs} são as coordenadas da curva gerada.

As variáveis de otimização para este problema seriam as coordenadas em y dos pontos de controle e seus respectivos pesos. A otimização funciona da seguinte forma: num primeiro momento são gerados valores aleatórios para as variáveis de otimização, sendo que estes estão dentro do domínio fornecido ao problema. Assim, várias curvas aleatórias são geradas e é calculado o valor da função objetivo para cada uma delas. A curva que tiver o menor valor da função objetivo é considerada a melhor aproximação e a partir dela começa o processo evolutivo de SCE-UA que tende a minimizar o erro. O resultado encontrado pode ser visualizado na figura 13, no qual se percebe excelente concordância entre a curva gerada e a resposta analítica da função cosseno fornecida.

Figura 13: Ajuste de curva com o método SCE-UA



Essa não é uma solução única. Com diferentes pontos de controle pode-se chegar ao mesmo resultado apresentado. Por esse motivo foi possível fazer uma análise de convergência para diferentes combinações de parâmetros da curva. Estes foram variados até se achar a considerada melhor combinação, na qual a solução mínima acontece no menor número de gerações possível. A partir disso se definiu como os parâmetros seriam aplicados no problema de águas rasas. Nessa etapa foram variados os graus da curva, número de variáveis de otimização e se estes seriam as coordenadas em y dos pontos de controle juntamente com seus pesos, ou apenas as coordenadas em y enquanto os pesos seriam fixados em 1. Considerou-se essa possibilidade de fixar os pesos levando em consideração o fato de que ao mantê-los como variáveis, os números de variáveis de otimização dobrariam – cada ponto de controle teria uma coordenada e um peso para ser encontrado. Ademais, no algoritmo desenvolvido, todas as variáveis possuem o mesmo domínio e em buscas de solução em grandes áreas, a mesma solução pode ser encontrada com diferentes combinações de coordenadas e pesos, fazendo com que o algoritmo demore a convergir ou nem venha a encontrar uma solução.

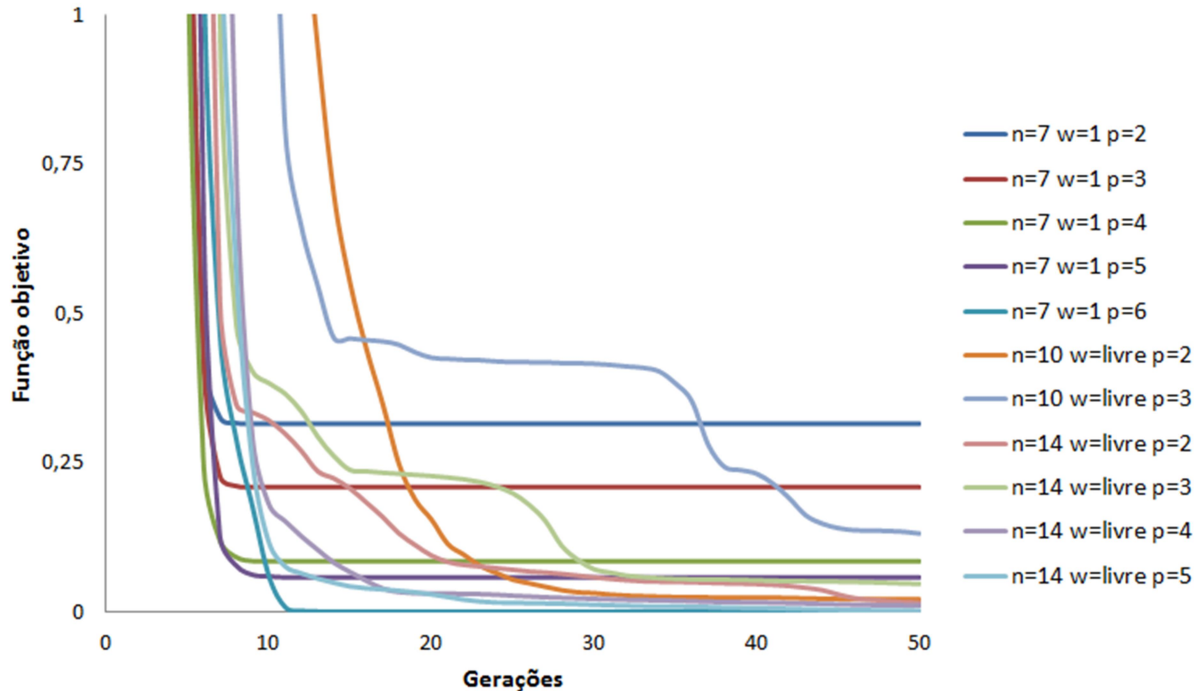
Na figura 14 são apresentados os resultados de diferentes combinações, nas quais se utilizou 7 pontos de controle igualmente espaçados no eixo x para se achar a solução. Em alguns casos estes estavam com peso fixo, em outros com peso livre, e ainda dois testes foram feitos no qual o número de pontos de controle diminuiu para 5 com pesos variáveis (10 variáveis de otimização). Na legenda da imagem n é o número de variáveis de otimização utilizado, w é o peso, sendo que livre significa que ele também era uma variável de otimização, e p é o grau da curva. Como foram definidos 7 pontos de controle, o grau máximo que pode se chegar é 6.

Analisando os resultados obtidos a partir dessas diferentes combinações de parâmetros, pôde-se concluir que variar os pesos é menos efetivo do que variar apenas as coordenadas em y dos pontos de controle. Na figura 14 é nítido que as otimizações feitas apenas com as coordenadas em y como incógnita convergiram mais rápido. E a melhor resposta, que inclusive é a que foi apresentada na figura 13, é com grau $p = 6$, $w = 1$ e $n = 7$.

Desse modo estariam sendo geradas curvas *B-splines* ao invés de curvas NURBS, tendo em vista que uma curva NURBS com pesos unitários em todos os pontos de controle acaba por se tornar uma curva *B-spline*. Apesar de que NURBS possuem aplicação muito maior que *B-splines*, estas se mostraram eficientes para resolver o problema em questão. Ao representar com precisão uma curva cosseno, sendo que esta tem forma muito semelhante às apresentadas

por curvas de vazão, acredita-se que seja capaz de representar a maioria das curvas de vazão em problemas de controle.

Figura 14: Definição dos melhores parâmetros da curva com o método SCE-UA.



É também importante ressaltar que o número de pontos de controle foi definido como o menor número que conseguisse chegar à resposta desejada. Quanto menos variáveis de otimização, mais rápido o problema converge. Porém é necessário um número mínimo de pontos de controle para se conseguir representar uma curva, e para este caso, este número foi 7.

Concluindo, com essa seção se definiu que as variáveis de otimização de um problema de controle de águas rasas serão apenas as coordenadas em y dos pontos de controle da curva que representa a vazão a ser otimizada, que utilizar uma curva de grau $p = 6$ já é suficiente para se obter bons resultados, e que o algoritmo de geração de curvas funciona bem com o método SCE-UA. A próxima etapa consiste em aplicar os algoritmos de otimização e geração de curvas juntamente com as equações de águas rasas. Nos próximos capítulos serão apresentados resultados da aplicação tanto para simulação de problemas de águas rasas isoladamente, quanto para o seu controle.

6 EXEMPLOS DE APLICAÇÃO DAS EQUAÇÕES DE ÁGUAS RASAS

Neste capítulo serão apresentados alguns exemplos analisados com intuito de validar o código desenvolvido para a resolução de problemas utilizando as equações de águas rasas.

Para o pré-processamento, optou-se pela utilização do *software* GiD (2015), um pré-processador de malhas no qual é possível: a) atribuir a geometria do domínio de cada exemplo testado; b) a criação da malha conforme as necessidades dos problemas; c) a aplicação das condições de contorno nas fronteiras; d) entre outros dados necessários para dar entrada no programa.

Com a malha e propriedades do problema definidos, é preciso atribuir as condições iniciais do problema dentro do código. Lá se definem as velocidades iniciais, altura d'água e batimetria para cada nó do domínio, bem como se atribuem aos contornos de entrada ou saída qual vazão ocorre a cada passo de tempo. A vazão é aplicada por unidade de comprimento, sendo representada pela multiplicação da altura d'água e da velocidade, ou seja, possui unidade de m^2/s . Outros dados importantes, tais como o coeficiente de Manning e a altura d'água mínima também devem ser atualizadas dentro do código. O valor da gravidade foi definido como $g = 9,806 m/s^2$ em todos os exemplos. Os passos de tempo dos exemplos foram calculados automaticamente seguindo o limite máximo necessário para a estabilidade.

O programa gera como saída arquivos, os quais apresentam os resultados de altura d'água e velocidades a cada instante de tempo definido. Com esses resultados se pôde comparar os resultados obtidos em cada exemplo com os de outros pesquisadores e assim validar o algoritmo desenvolvido.

6.1 ESCOAMENTO EM CANAL COM TRECHO DE ELEVAÇÃO PARABÓLICA DO FUNDO

Este primeiro exemplo representa um canal com topografia variável por onde entra uma vazão constante e o objetivo é analisar o comportamento da elevação da onda com essas condições. As características deste exemplo permitem validar um escoamento passando por uma batimetria que não é constante, além de validar se o código representa corretamente transições

críticas, que neste caso ocorrem devido à velocidade do escoamento em conjunto com a elevação parabólica do leito no eixo do canal.

O escoamento ocorre em um canal de 25 metros de comprimento e 1 metro de largura, cujo leito é considerado sem fricção e é descrito pela função:

$$Z_b = \begin{cases} 0 & \text{se } x < 8\text{m} \\ 0,2 - 0,05(x - 10)^2 & \text{se } 8 \leq x \leq 12\text{ m} \\ 0 & \text{se } x > 12\text{ m} \end{cases} \quad (43)$$

As condições de contorno aplicadas são que o fluxo de entrada (em $x = 0$) é igual a $0,18\text{ m}^3/\text{s}$, na saída (em $x = 25\text{ m}$) o nível de água é igual a $0,33\text{ m}$, e as laterais do canal atuam com a condição de contorno de parede.

O modelo utilizado possui uma malha de 25233 nós e 48756 elementos. A malha bem refinada é necessária para simular a transição crítica que o problema propõe. Foram gerados elementos da ordem de $0,075\text{ m}$ nas regiões mais afastadas da elevação do fundo, enquanto que na região que passa por essa elevação (dos 8 aos 12 m) foram gerados elementos de ordem $0,025\text{ m}$.

A figura abaixo apresenta uma comparação entre os resultados numéricos obtidos no eixo do canal por este trabalho e a solução analítica, que é dada por Goutal e Maurel (1997).

Figura 15: Comparação entre solução numérica e analítica do nível d'água do canal com elevação parabólica de fundo.

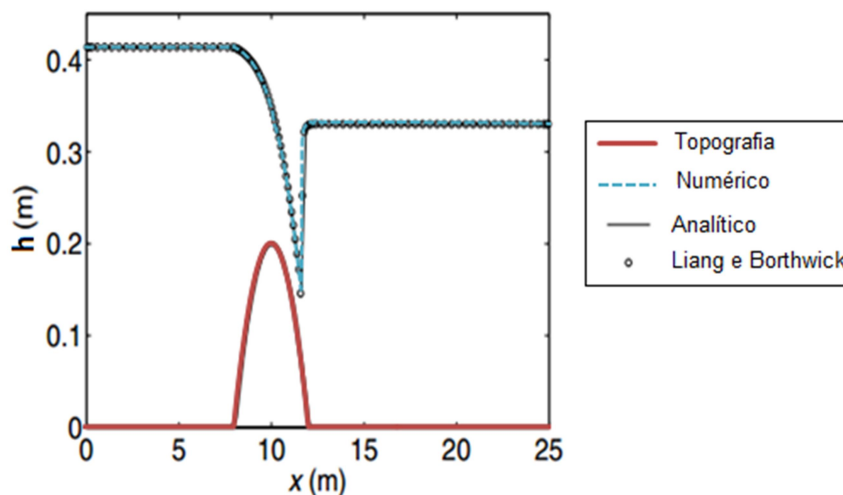
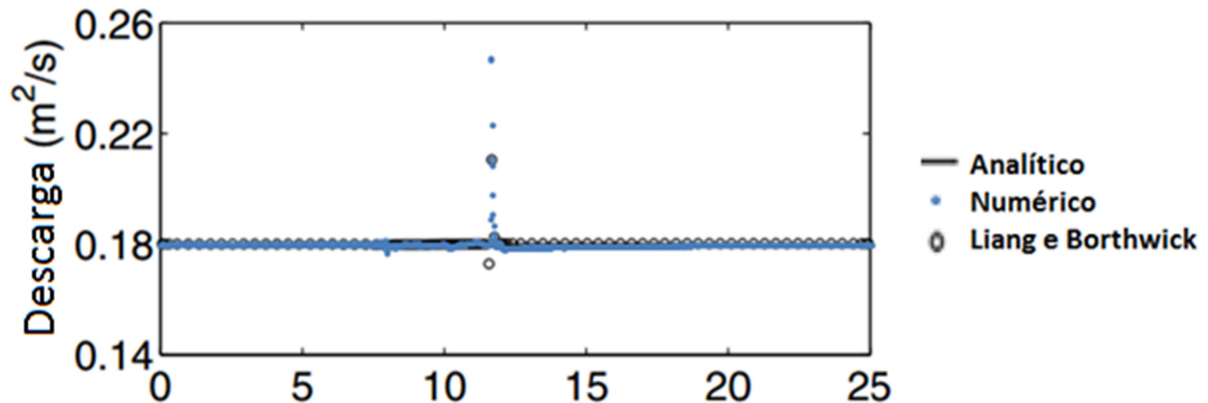


Figura 16: Comparação entre solução numérica e analítica da descarga do canal com elevação parabólica de fundo.



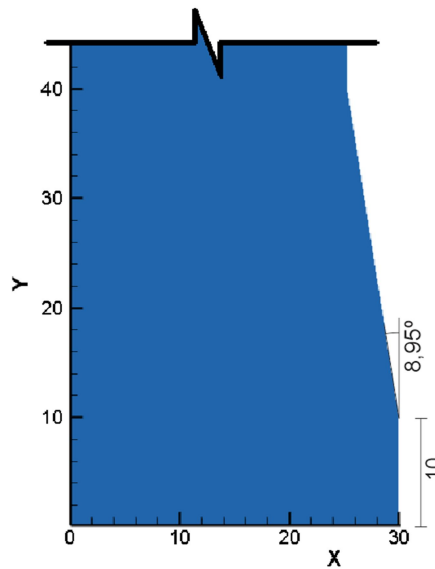
Nas comparações foram também utilizados resultados numéricos e experimentais apresentados pelos autores Liang e Borthwick (2009). Pode-se ver na figura 16 que tanto no resultado numérico encontrado pelo algoritmo CBS desenvolvido, quanto no dos autores apresentado como referência, houve uma discrepância no gráfico de descarga logo após a elevação do fundo. Esta diferença em relação ao resultado analítico também foi percebida por outros pesquisadores. Este erro é aceitável por se tratar de um método numérico, e ocorre numa parte crítica do problema. Já o nível d'água teve um bom comportamento ao ser comparado e pode-se considerar que a resposta do programa está de acordo com o resultado analítico.

6.2 SALTO HIDRÁULICO DEVIDO À PAREDE OBLÍQUA

Este exemplo analisa um salto hidráulico formado a partir de um escoamento supercrítico num canal de parede oblíqua, a qual forma um ângulo de $8,95^\circ$ com a direção da corrente ainda não perturbada. O fundo do canal é considerado sem fricção e as condições de contorno aplicadas foram a vazão de entrada (em $y = 0$) igual a $8,57 \text{ m}^3/\text{s}$, a condição de parede nas laterais do canal, incluindo a parede oblíqua, e na saída foi modelado um canal longo suficiente para o problema convergir, visto que no algoritmo desenvolvido não foi modelada condição de contorno de fronteira aberta que possui características não reflexivas (neste caso, foi necessário expandir o canal 65 metros na direção y).

A geometria é apresentada na figura 17, focando somente no domínio representado pelo retângulo de $30 \times 40 \text{ m}$, pois os valores de y maiores que 40 m não são de interesse.

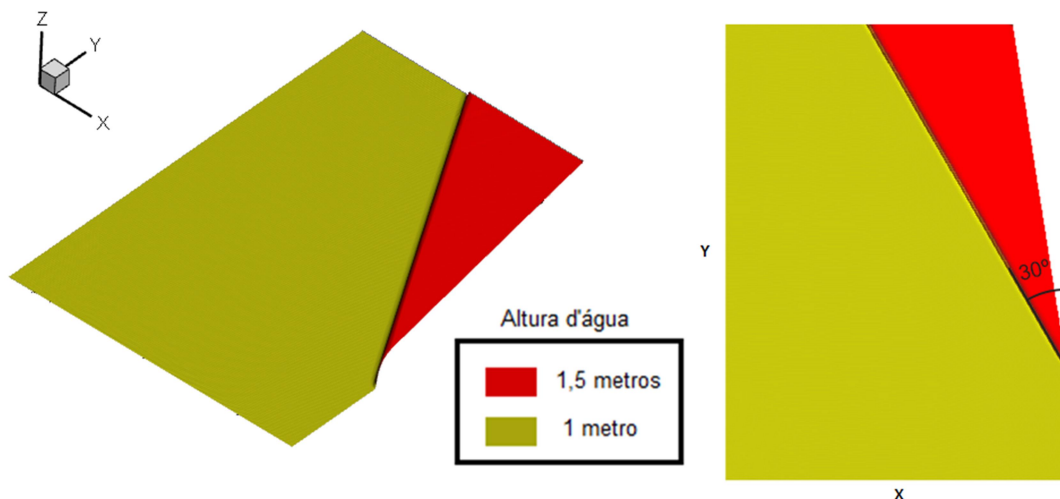
Figura 17: Geometria do exemplo de salto hidráulico devido à parede oblíqua.



A malha utilizada possui 32671 nós e 64662 elementos, ou seja, uma malha bem refinada, que foi necessária para representar adequadamente o salto hidráulico, onde os elementos são da ordem de 0,1 m. No alongamento do canal os elementos foram aumentando gradativamente de tamanho, pois naquela região não era necessário um refinamento tão grande.

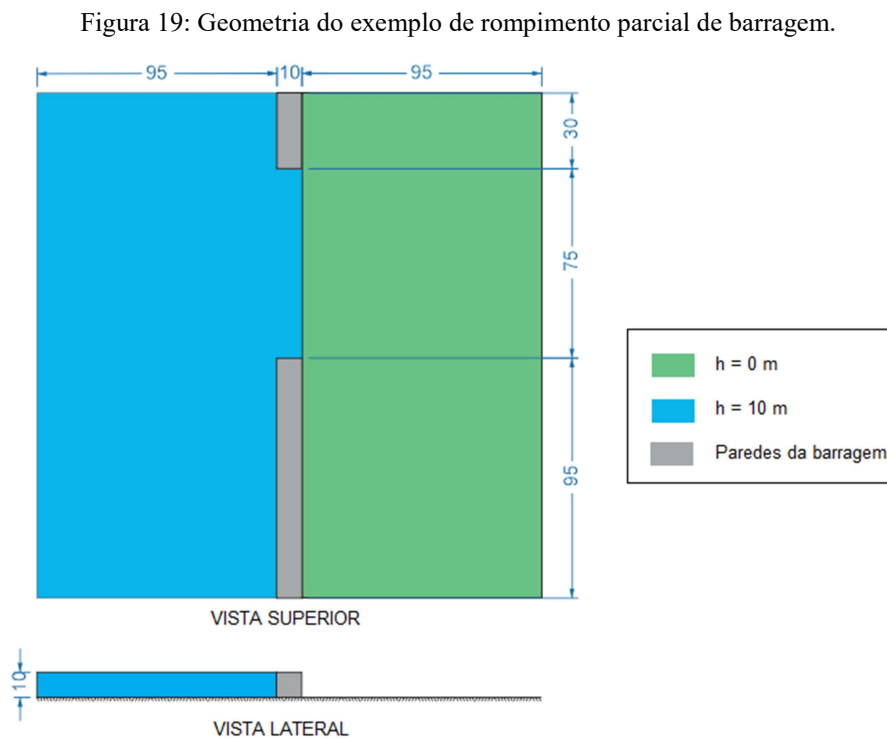
Os resultados analíticos mostram que o salto começa em $y = 10$ m, convergindo num ângulo de 30° com a direção y , onde nesse espaço a água sobe de 1 metro para 1,5 metros de altura (ROGERS *et al.*, 2001). Neste caso, os resultados numéricos foram exatos, e estão apresentados na figura seguinte:

Figura 18: Altura d'água do exemplo de salto hidráulico devido à parede oblíqua.



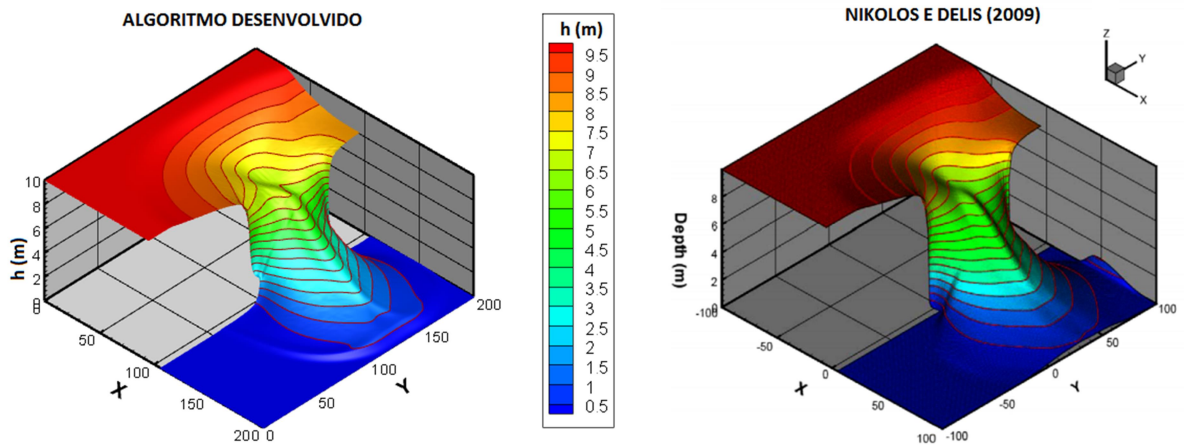
6.3 ROMPIMENTO PARCIAL DE BARRAGEM

O objetivo deste exemplo é verificar a habilidade do código de reproduzir a propagação de uma onda por um terreno seco. Foi simulado então o rompimento parcial de uma barragem, onde o terreno a jusante se encontra sem altura d'água alguma. O domínio espacial é representado por uma malha de 200x200 metros, onde a barragem e suas dimensões estão apresentadas na figura 19.



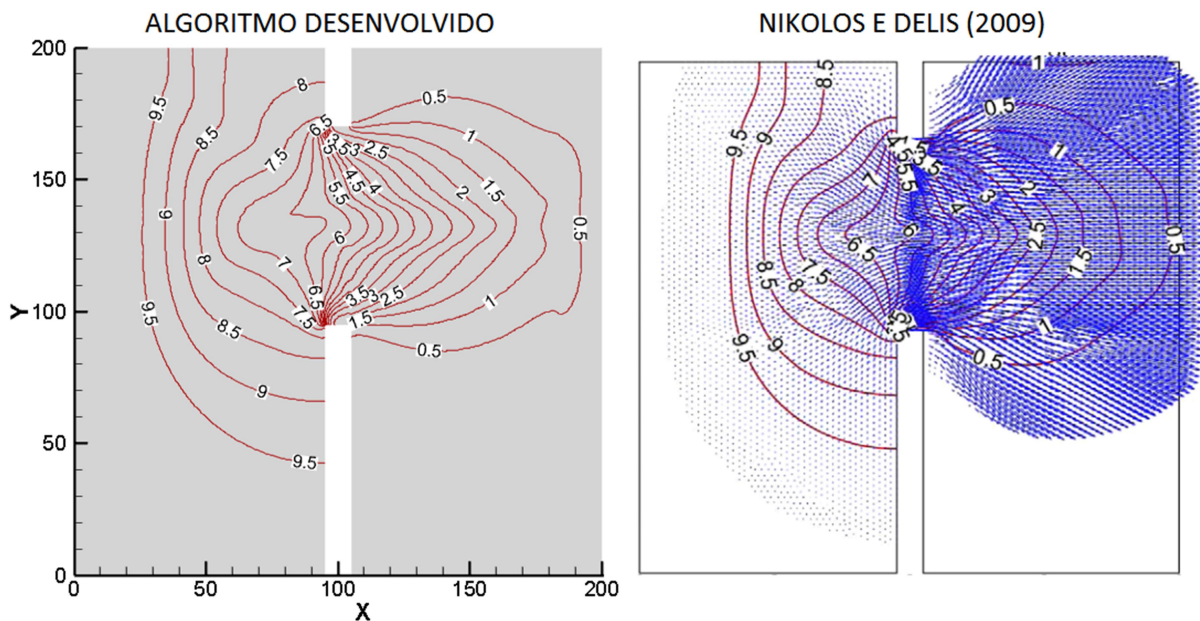
A malha tem 39276 nós e 77500 elementos, que possuem dimensão na ordem de 1 m. O fundo é considerado sem fricção e as condições de contorno aplicadas foram a representação de paredes em todas as fronteiras, inclusive nos muros da barragem. Foi atribuído um valor de $h_{min} = 0,01 \text{ m}$ para limitar regiões consideradas secas ou molhadas. Não existe referência que apresente o resultado analítico para este problema, mas resultados numéricos de vários autores estão disponíveis como os de Caleffi *et al.* (2003) e Nikolos e Delis (2009), que tiveram resultados semelhantes ao do algoritmo desenvolvido. Os resultados da simulação foram comparados com os obtidos por Nikolos e Delis (2009) em $t = 7,2 \text{ s}$.

Figura 20: Resultado do exemplo de rompimento de barragem em terreno seco em $t=7,2s$



(fonte: adaptado de Nikolos e Delis, 2009)

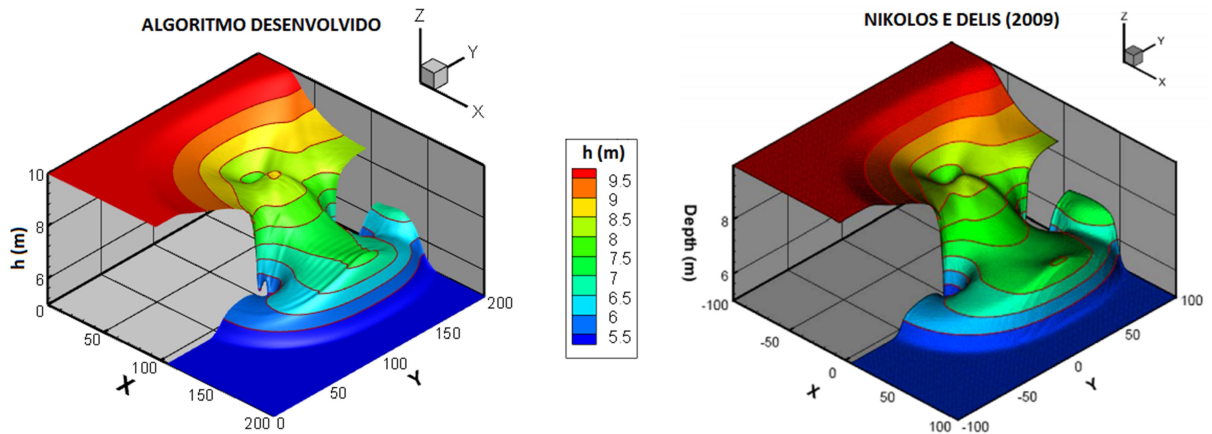
Figura 21: Contornos de nível da profundidade d'água (m) do exemplo de rompimento de barragem (seco) em $t=7,2s$



(fonte: adaptado de Nikolos e Delis, 2009)

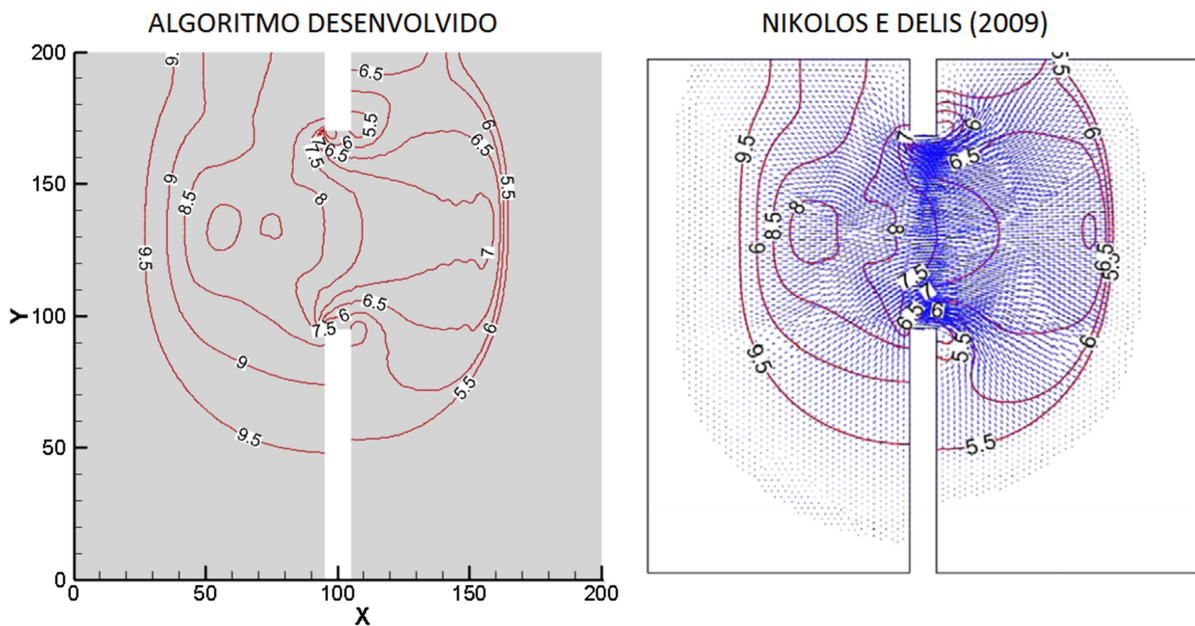
Utilizando o mesmo domínio espacial, pode-se fazer um teste para um rompimento de barragem no qual a propagação da onda se dá por um terreno já inundado. Em vez de o terreno a jusante estar seco, foi assumido que este possui um nível d'água de 5 metros. Este teste também não possui referência de solução analítica disponível, mas é possível comparar os resultados com diversos autores, tais como Caleffi *et al.* (2003) e Nikolos e Delis (2009). Mais uma vez os resultados foram satisfatórios.

Figura 22: Resultado do exemplo de rompimento de barragem em terreno molhado em $t=7,2s$



(fonte: adaptado de Nikolos e Delis, 2009)

Figura 23: Contornos de nível da profundidade d'água do exemplo de rompimento de barragem (molhado) em $t=7,2s$



(fonte: adaptado de Nikolos e Delis, 2009).

6.4 ROMPIMENTO DE BARRAGEM EM TERRENO COM TRÊS ILHAS CÔNICAS

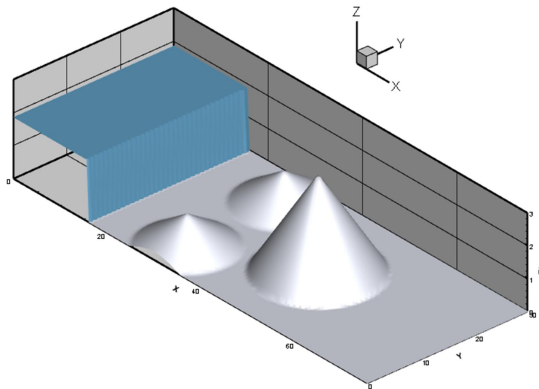
Este exemplo simula o rompimento total de uma barragem, cuja onda formada passa por um terreno seco que possui três elevações cônicas na sua topografia. Este teste foi proposto por Kawahara e Umetsu (1986), e reconsiderado por vários autores como Liang e Borthwick (2009) e Nikolos e Delis (2009). O domínio espacial é um retângulo de 75x30 metros, cuja topografia é dada por:

$$Z_b = \max \begin{cases} 0 \\ 1 - \frac{1}{8} \sqrt{(x - 30)^2 + (y - 6)^2} \\ 1 - \frac{1}{8} \sqrt{(x - 30)^2 + (y - 24)^2} \\ 3 - \frac{3}{10} \sqrt{(x - 47,5)^2 + (y - 15)^2} \end{cases} \quad (44)$$

A malha possui 40816 nós e 81000 elementos cuja dimensão é da ordem de 0,4 m em quase todo domínio e em torno da ilha maior, onde foi feito um refinamento, da ordem de 0,2 m. A rugosidade do fundo é representada pelo coeficiente de Manning $n = 0,018 \text{ s/m}^3$, e as condições de contorno das fronteiras são as de paredes. Foi atribuído um valor de $h_{\min} = 0,01 \text{ m}$ para limitar regiões consideradas secas ou molhadas.

Inicialmente existe uma barragem em $x = 16 \text{ m}$, na qual o nível d'água se encontra em 1,875 m, conforme apresentado na figura 24. Em $t = 0$, a barragem instantaneamente desaparece e a simulação começa.

Figura 24: Geometria do exemplo de três ilhas cônicas



As figuras seguintes ilustram o comportamento da onda em diferentes tempos. Em $t = 2 \text{ s}$, a onda chega às duas pequenas ilhas e começa a crescer sobre elas. Em $t = 6 \text{ s}$, as pequenas ilhas estão totalmente submersas, e a onda chega à grande ilha. Em $t = 12 \text{ s}$, a onda passa pelos lados da grande ilha e começa a inundar o trecho a sua jusante. Nos passos de tempo seguintes, a onda reflete nas paredes e juntamente com a fricção do fundo e interação com as ilhas começa a tender a sua estabilidade, como pode ser visto em $t = 30 \text{ s}$. Em $t = 300 \text{ s}$ a simulação já se apresenta estabilizada no seu domínio espacial.

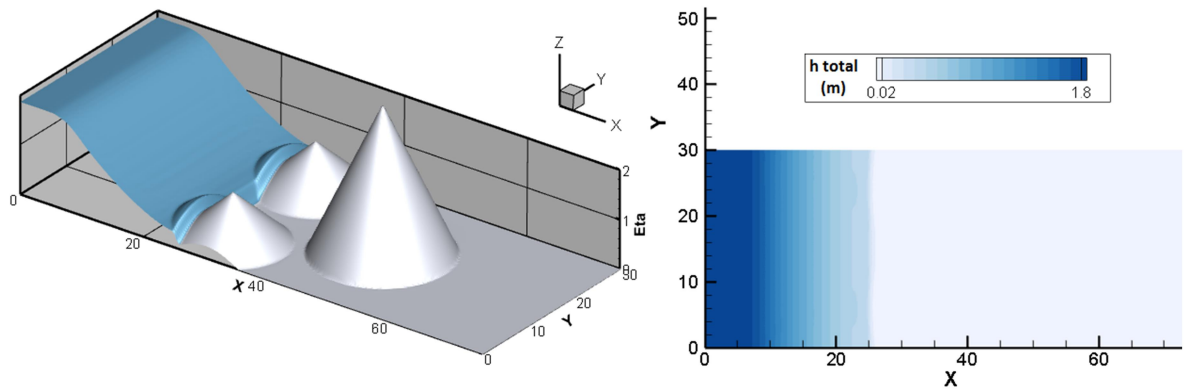
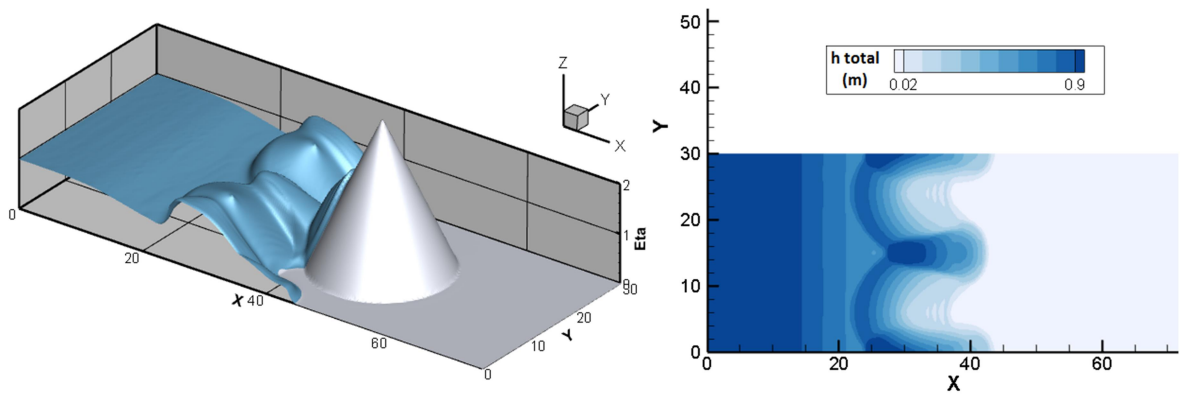
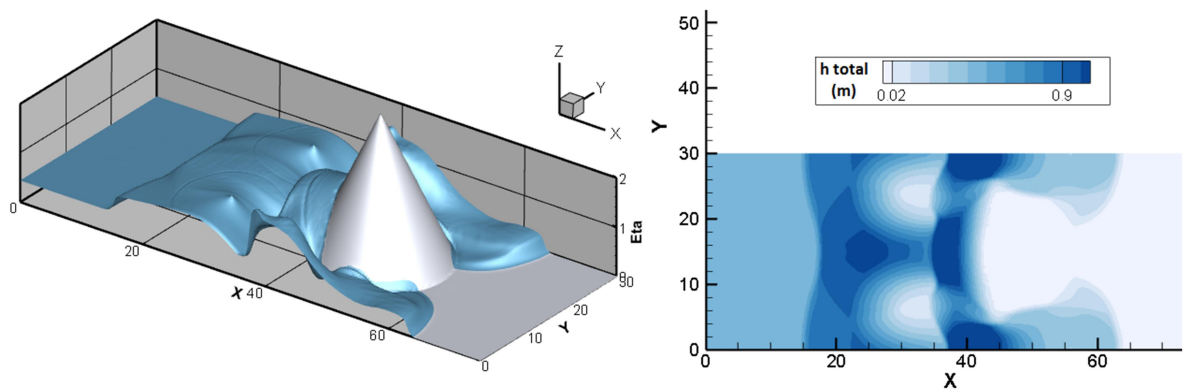
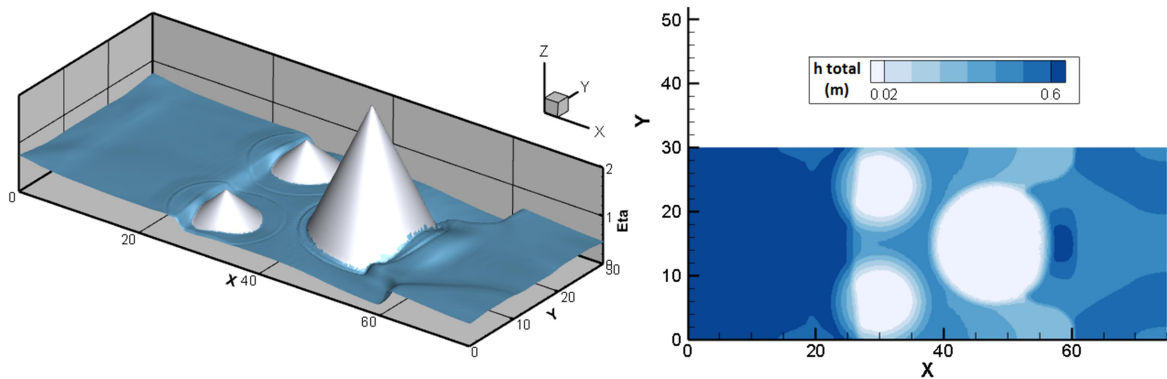
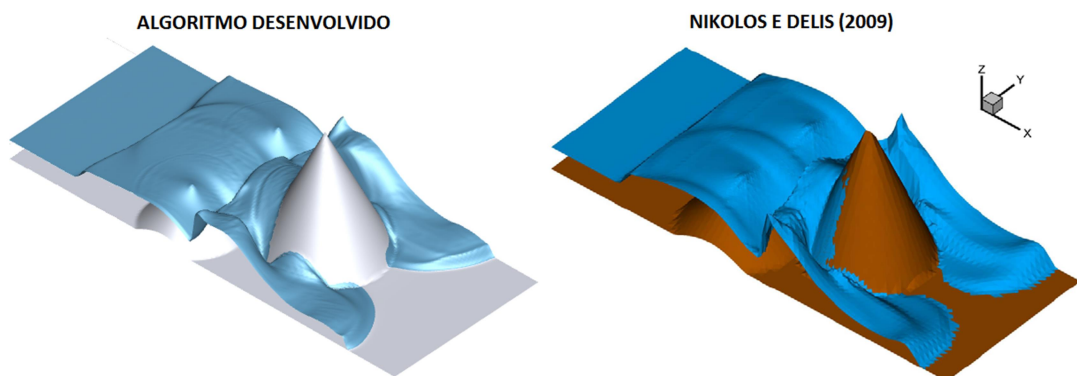
Figura 25: Resultados três ilhas em $t = 2$ sFigura 26: Resultados três ilhas em $t = 6$ sFigura 27: Resultados três ilhas em $t = 12$ s

Figura 28: Resultados três ilhas em $t = 30$ s

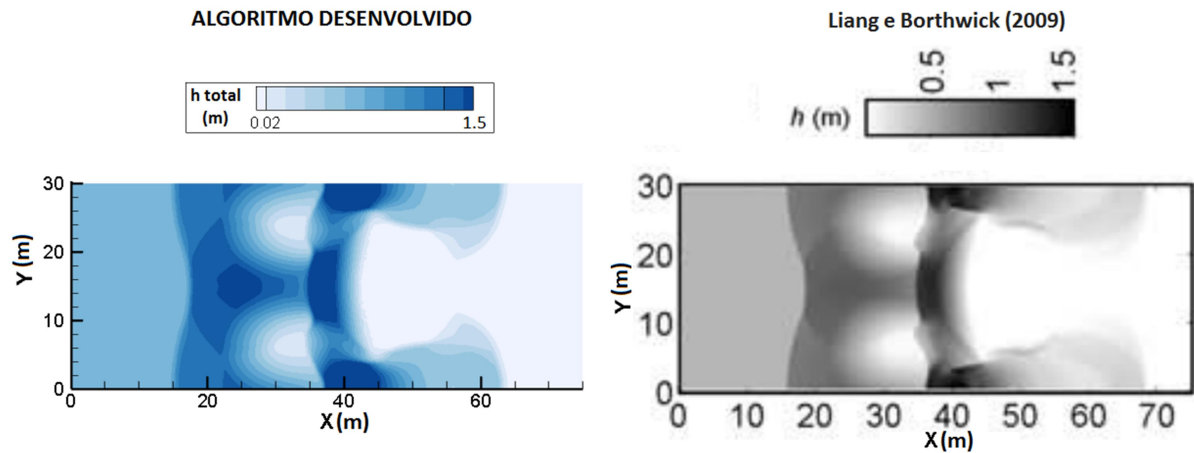
Este exemplo possui uma característica peculiar em relação à formulação utilizada para a representação de elementos secos e molhados. Como a topografia em formato de cone possui uma inclinação bastante elevada, se a malha não for suficientemente refinada, existirá uma grande oscilação de secagem e molhagem dos elementos que participam dessa transição. Isso ocorre pela formulação utilizada, que faz uma média das alturas em cada nó quando este se torna molhado, unida à inclinação do terreno, que faz com que a tendência seja daquele elemento secar devido à gravidade. O problema é resolvido com um refinamento de malha naquela região, levando em consideração o fato de que elementos menores possuem nós mais próximos e mesmo que a oscilação persista, será numa área reduzida.

Apesar de não existir solução analítica para este problema, os resultados encontrados foram muito semelhantes aos encontrados por outros autores, e podem ser vistos em Kawahara e Umetsu, (1986), Liang e Borthwick (2009) e Nikolos e Delis (2009). Duas comparações em $t = 12$ s estão apresentadas nas figuras 29 e 30.

Figura 29: Comparação dos resultados obtidos em $t = 12$ s com Nikolos e Delis (2009)

(fonte: adaptado de Nikolos e Delis, 2009)

Figura 30: Comparação dos resultados obtidos em $t = 12$ s com Liang e Borthwick (2009)



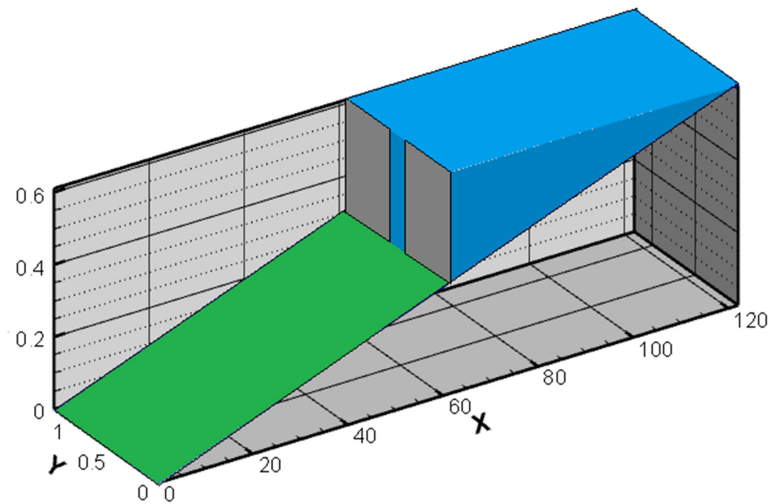
(fonte: adaptado de Liang e Borthwick, 2009)

6.5 ROMPIMENTO PARCIAL DE BARRAGEM EM TERRENO INCLINADO

Outro modelo de rompimento de barragem pôde ainda ser testado, sendo que este se diferencia dos anteriores por possuir topografia em declive. Este problema foi simulado numericamente por Bradford e Sanders (2002) e comparado com resultados experimentais realizados por *Waterways Experimental Station* (WES, 1960). Este é um problema interessante, pois valida a metodologia utilizada para problemas de inundação a medida que a onda avança no declive, bem como a aplicação de rugosidade de fundo juntamente com esta metodologia.

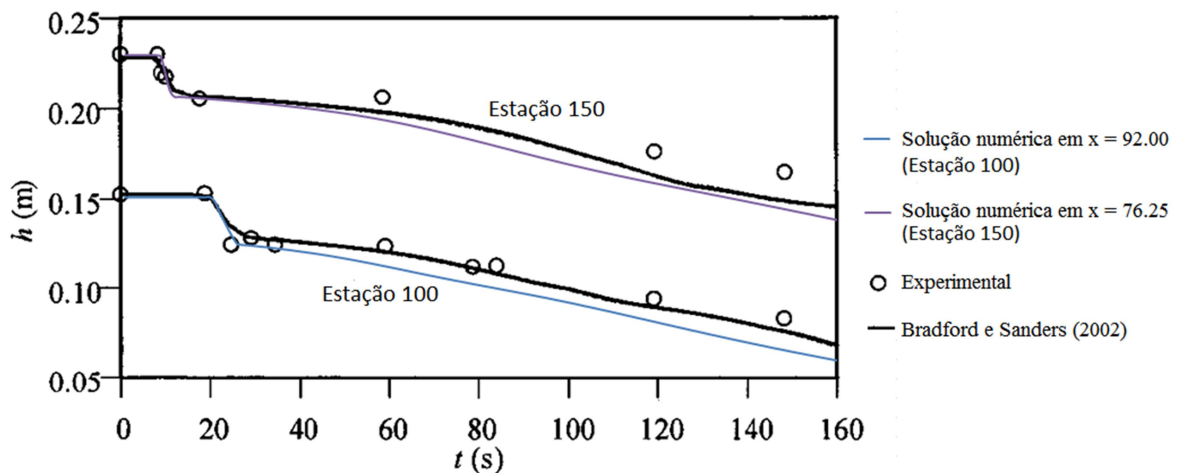
É simulado um canal de 122 m de comprimento e 1,22 m de largura que possui uma declividade de 0,005. O leito possui uma rugosidade representada pelo coeficiente de Manning $n = 0,009 \text{ s/m}^3$ e foi atribuído um valor de $h_{\text{mín}} = 0,01 \text{ m}$. A barragem está localizada na metade do canal, possuindo uma abertura central de 0,122 m, e água com profundidade de 0,305 m exatamente atrás desta. O nível d'água varia acompanhando a declividade do terreno, chegando a zero na fronteira direita. A barragem possui espessura infinitesimal, e para o modelo utilizado neste trabalho, ela foi simulada com paredes variáveis de 0 a 1 mm, aumentando a espessura da abertura central até as paredes externas. No instante $t = 0$, o problema pode ser visualizado na figura 31.

Figura 31: Rompimento parcial de barragem com terreno inclinado



No modelo utilizado neste trabalho, a malha possui 158600 elementos e 82376 nós com elementos da ordem de 0,05 m. Todas as fronteiras possuem característica de serem paredes reflexivas, e assim como os outros problemas de rompimento de barragem a única força atuante é a gravidade. Os resultados podem ser vistos na figura 32.

Figura 32: Resultados do exemplo de rompimento parcial de barragem com terreno inclinado



Mais uma vez as respostas foram muito semelhantes às encontradas pelos resultados experimentais e simulados por Bradford e Sanders (2002).

6.6 TIGELA PARABÓLICA

O exemplo da tigela parabólica consiste num corpo de fundo parabólico axissimétrico, que num estado inicial possui elevação de água também em uma forma parabólica e velocidade nula. Com base no modelo de Bunya *et al.* (2009), o fundo é prescrito pela equação 45.

$$Z_b = \alpha r^2 \quad (45)$$

A solução exata da profundidade d'água e sua velocidade com o tempo é dada por:

$$h(r, t) = \frac{1}{X + Y \cos(\omega t)} + \frac{\alpha(Y^2 - X^2)r^2}{(X + Y \cos(\omega t))^2} \quad (46)$$

$$u(x, y, t), v(x, y, t) = \frac{Y \omega \sin(\omega t)}{X + Y \cos(\omega t)} \left(\frac{x}{2}, \frac{y}{2} \right) \quad (47)$$

onde:

$$r = \sqrt{x^2 + y^2} \text{ m};$$

$$\alpha = 1.6 \times 10^{-7} \text{ m}^{-1};$$

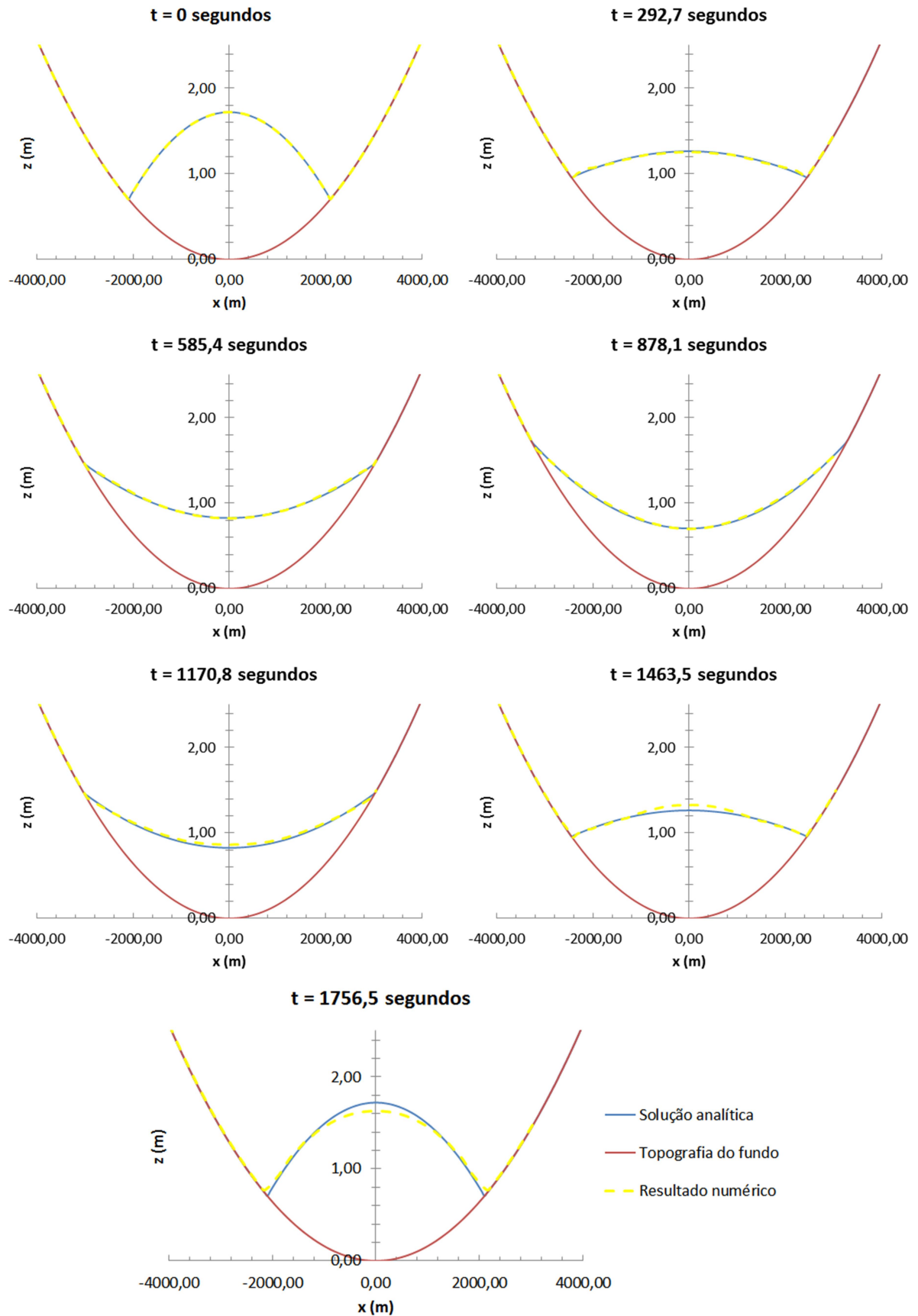
$$X = 1 \text{ m}^{-1};$$

$$Y = -0.41884 \text{ m}^{-1};$$

$$\omega = \sqrt{8g\alpha} \text{ sendo } g \text{ a aceleração da gravidade.}$$

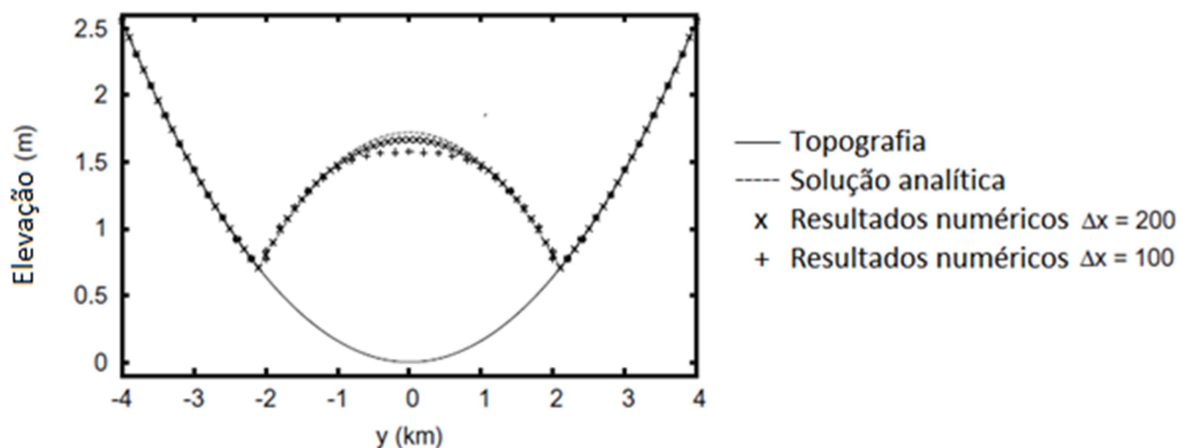
O domínio do problema é um quadrado de lado de 8000 m centralizado na posição (0,0), do qual foi simulado apenas um quarto deste domínio com condições de simetria nas fronteiras que interceptavam o centro, e com condições de parede reflexiva nas bordas externas. A malha utilizada possui 51521 nós e 102400 elementos e o fundo é considerado sem fricção com $h_{\min} = 0,01 \text{ m}$. Abaixo estão ilustrados os resultados numéricos junto com a resposta analítica a cada 1/6 período de tempo do problema.

Figura 33: Resposta analítica e numérica do exemplo da tigela parabólica



Pode-se observar que em 1756,5 s, quando a onda termina um ciclo do período, houve alguma dissipação numérica, pois a elevação d'água não alcança a altura que o resultado analítico apresenta. Isto também ocorre para Bunya *et al.* (2009) como pode ser visto na figura 34, sendo este erro aceitável para ainda assim considerar que o algoritmo fornece bons resultados.

Figura 34: Resultados da tigela parabólica em $t=1756,5$ encontrados por Bunya *et al.* (2009)



(fonte: adaptado de Bunya *et al.*, 2009)

6.7 INUNDAÇÃO EM ZONAS URBANAS

Com o algoritmo das equações de águas rasas desenvolvido, também é possível fazer a simulação de inundação em zonas urbanas, o que pode ser uma ferramenta importante para prevenir os riscos causados por inundações. A simulação de inundação em zonas urbanas permite criar um mapa de risco da região, isto é, identificar as zonas que serão mais atingidas por uma enchente, bem como permite desenvolver um plano de controle, identificar informações importantes como tempo de chegada d'água na área a ser inundada, a máxima profundidade e a velocidade d'água que chega na área inundada, e a duração da inundação. Tendo essas informações em mãos, pode-se planejar formas de prevenção ou ação quando houver um alagamento.

Porém, diferente dos outros exemplos apresentados, um meio urbano é um meio complexo do ponto de vista geométrico para se aplicar as equações de águas rasas. É preciso representar as diversas edificações que estarão na área de estudo a ser inundada e, para isso, existem diferentes técnicas que podem ser utilizadas, segundo Alcrudo (2004):

- a) rugosidade local diferenciada representando os prédios e obstáculos;
- b) técnica de elevação de topografia;
- c) técnica de paredes verticais.

A técnica de rugosidade é eficiente se a resposta que se deseja encontrar diz respeito somente às velocidades e altura d'água de forma geral no meio urbano. Para isso, funciona bem aplicar uma rugosidade equivalente a que o meio urbano representa, sem se preocupar com as paredes dos prédios. Porém, essa técnica não representa o comportamento local em torno das edificações, e se o objetivo for esse, as técnicas de elevação da topografia e paredes verticais são mais eficientes. A técnica de elevação da topografia é a representação dos prédios por meio da batimetria do leito, que é elevado no formato dos prédios, semelhante ao que foi feito no exemplo das três ilhas cônicas. O problema desta técnica é que a representação da parede nunca será vertical, visto que, para isso, dois nós teriam as mesmas coordenadas xy com elevação no eixo z diferente. Isso não pode ocorrer num modelo de águas rasas em duas dimensões, e para uma representação adequada, a malha teria de ser muito refinada. Além disso, já foi visto no exemplo das três ilhas cônicas que uma variação grande da topografia pode ter problemas com o método aplicado para elementos secos e molhados, então, para o algoritmo desenvolvido neste trabalho, está técnica não é a mais recomendada.

A terceira técnica, e aquela que foi empregada neste último exemplo de aplicação do algoritmo desenvolvido, é a de paredes verticais. Aqui são feitos “furos” na malha onde estão localizados os prédios, e como contorno, para fechar essas fronteiras, são aplicadas as condições de paredes reflexivas. Com essa condição, as inundações ficam limitadas a não ultrapassarem o topo das edificações, o que nesse caso seria impossível de simular. Porém, com pequenas exceções de grandes tsunamis, é isso que ocorre nas zonas urbanas: o escoamento por entre os prédios e não sob eles, sendo esta limitação irrelevante para a maioria dos casos.

O teste reproduzido é baseado em Szydłowski e Twaróg (2006), que fizeram tanto um modelo numérico, como também um modelo experimental para validar os resultados encontrados. Eles estudaram um modelo reduzido que representa o rompimento de uma barragem que causaria inundações numa zona urbana. No domínio são dispostos 30 prédios representados esquematicamente no modelo com as dimensões 0,1 x 0,1 m e espaçados a cada 0,1 m, conforme a figura 35. Estes farão parte de uma inundação ocorrida pelo rompimento parcial de uma barragem representada por uma altura d'água inicial de 0,21 m. A barragem possui

uma parede de 0,12 m, e possui dimensões de 3 metros de largura e 3 metros de comprimento. O domínio da parte inicialmente seca e onde se localizam os prédios foi feito suficientemente grande para reproduzir as fronteiras abertas não desenvolvidas no algoritmo desse trabalho. Enquanto que a largura desta região foi modelada com 3 m pelos autores, neste trabalho ela foi estendida para 10 m a fim de cumprir esta condição. Já o comprimento dessa região foi estendido de 3,75 m para 4,5 m.

A malha utilizada foi refinada em torno dos prédios, e os elementos foram gradativamente aumentando conforme se afastavam destes, possuindo 243392 elementos e 124501 nós (figura 36). O leito de todo domínio possui rugosidade equivalente a um coeficiente de Manning igual a $n = 0,018 s/m^3$ e não possui variação de batimetria. Foi atribuído um valor de $h_{min} = 0,01 m$. No instante $t = 0$, uma abertura centralizada na barragem de 0,5 m de largura surge, dando início a simulação.

Figura 35: Inundação em zonas urbanas: geometria.

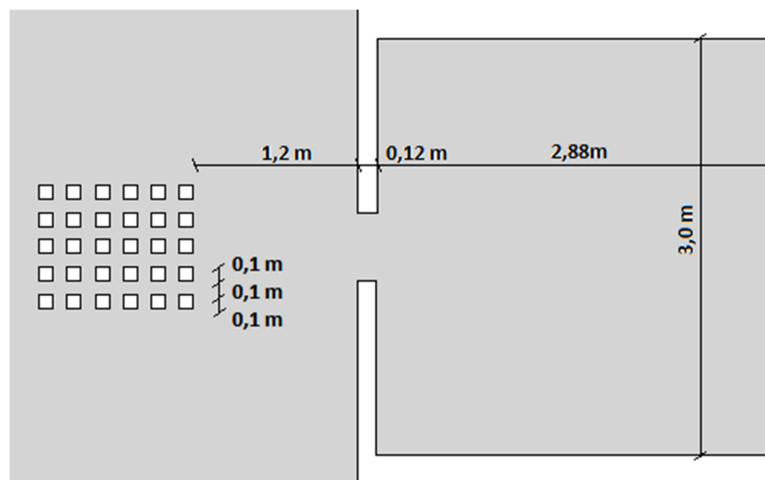
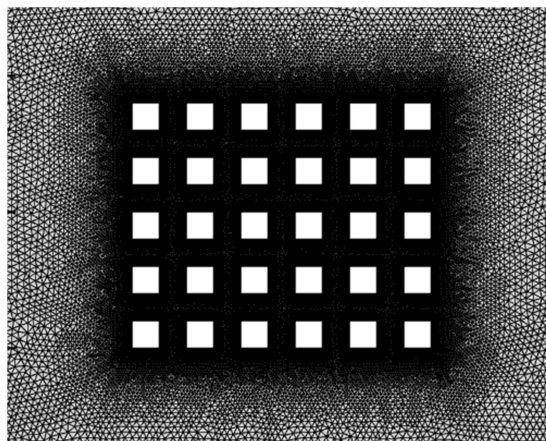
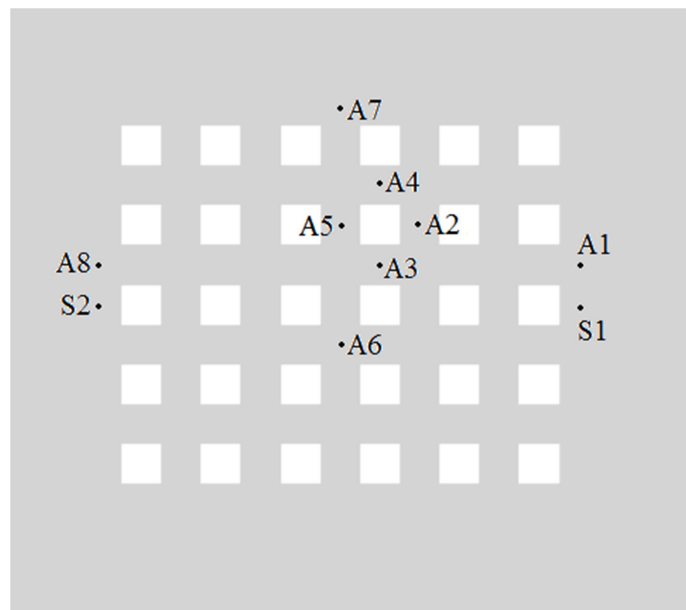


Figura 36: Inundação em zonas urbanas: malha.



Nas figuras seguintes estão apresentados os resultados obtidos para 20 segundos de simulação, comparado aos encontrados por Szydłowski e Twaróg (2006) em 10 estações espalhadas pelo domínio conforme a figura 37. Mais uma vez houve boa concordância entre resultados experimentais e numéricos obtidos, sendo que as pequenas diferenças encontradas se devem ao tipo de método utilizado por cada autor, malha, e alguma diferença nos dados iniciais visto que as informações do experimento de Szydłowski e Twaróg (2006) não estão claras na sua totalidade.

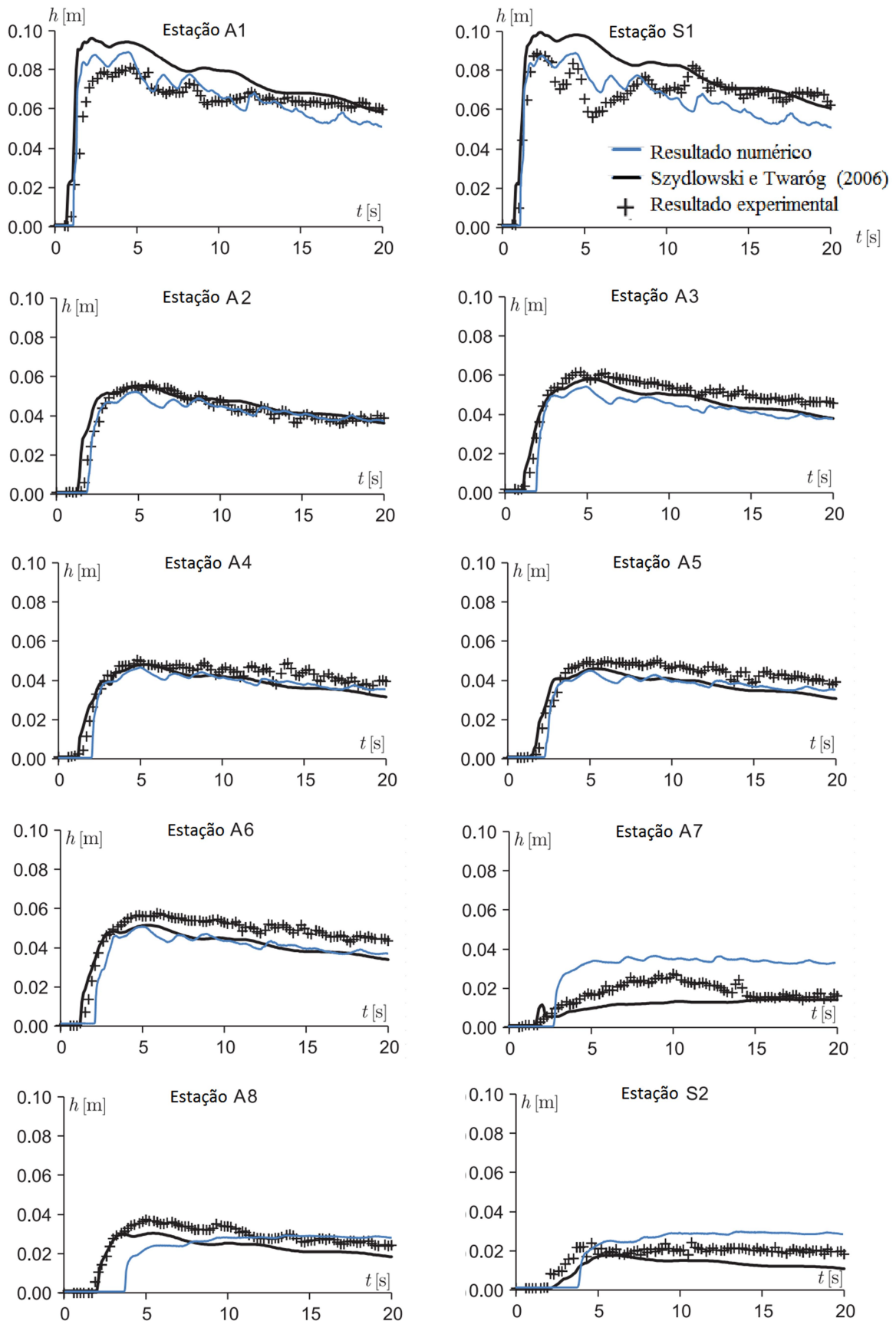
Figura 37: Inundação em zonas urbanas: estações de análise.



Nas figuras 39 e 40 também podem ser vistas imagens em 2 e 3D da simulação no tempo 3 segundos.

Com estes resultados, é finalizada a verificação do algoritmo desenvolvido, o qual se mostrou eficiente para diversas aplicações, incluindo análises de inundações urbanas, mostrando que tem potencial para ser utilizado em estudos de casos reais. É possível reproduzir qualquer problema que se encaixe nas restrições das equações de águas rasas.

Figura 38: Inundação em zonas urbanas: resultados nas estações de análise



(fonte: adaptado de Szydłowski e Twaróg, 2006)

Figura 39: Inundação em zonas urbanas: simulação aos 3 s em 2D.

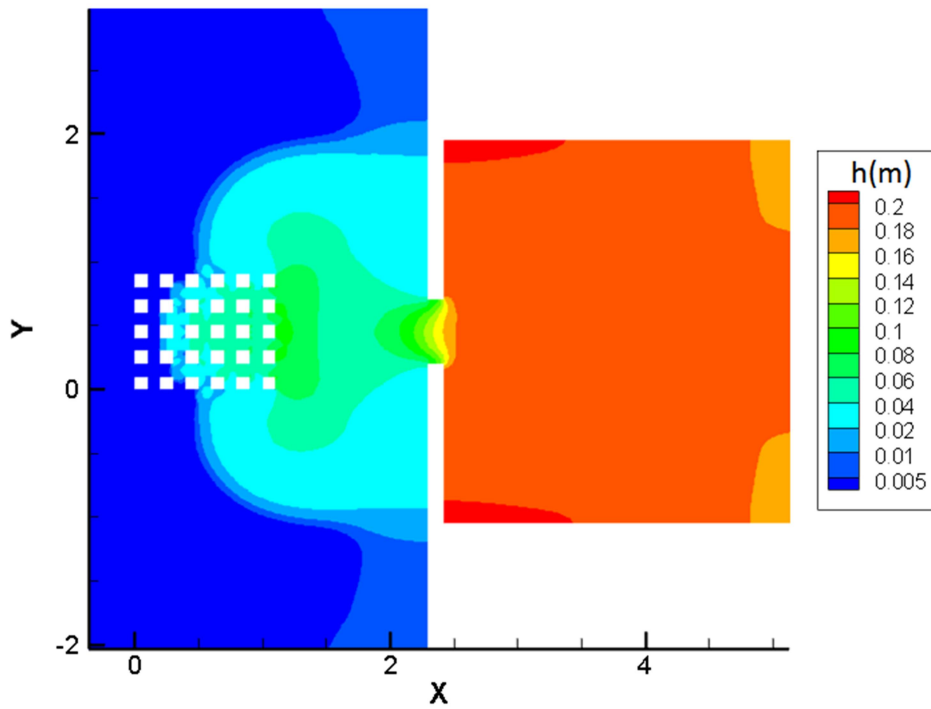
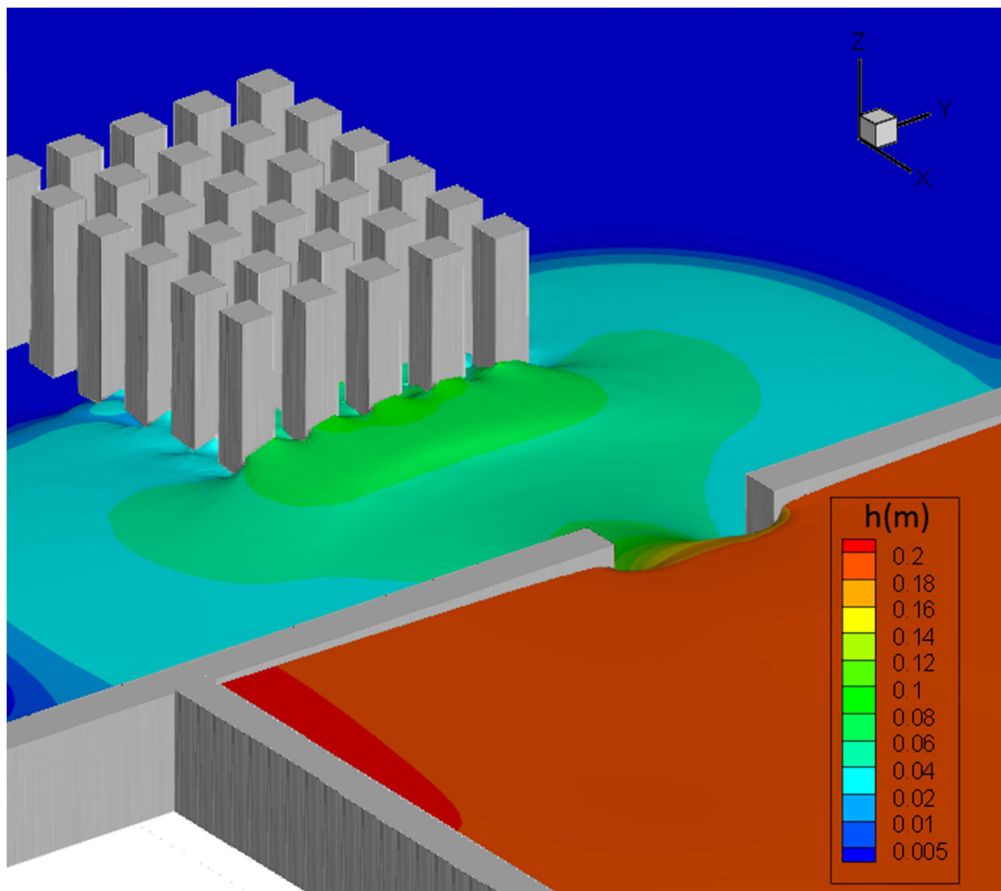


Figura 40: Inundação em zonas urbanas: simulação aos 3 s em 3D.

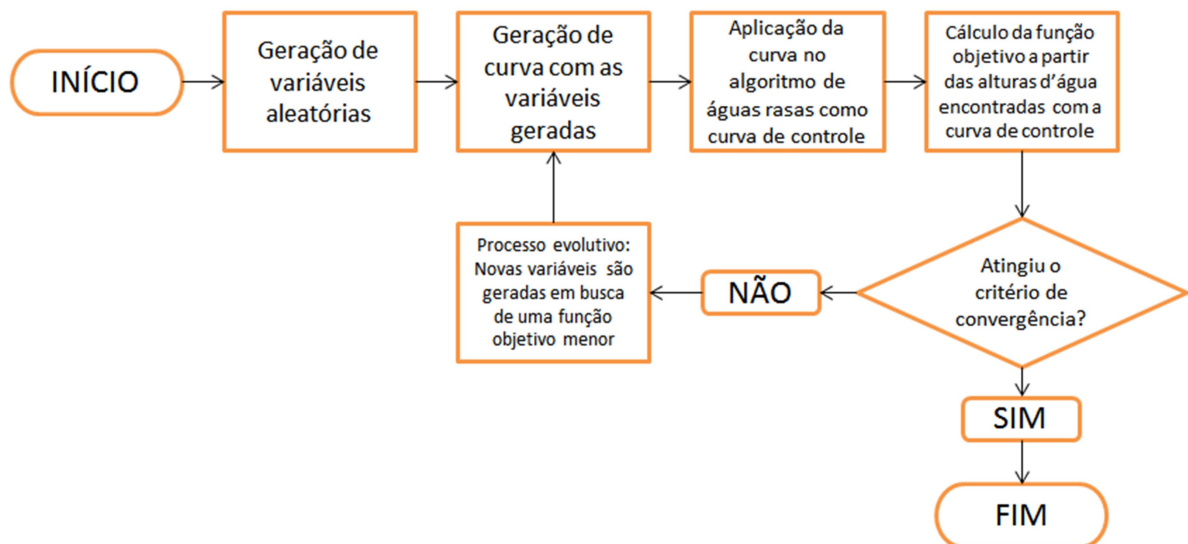


7 EXEMPLOS DE APLICAÇÃO DO ALGORITMO DE CONTROLE

Neste capítulo serão apresentados exemplos da aplicação do algoritmo de otimização do controle de águas rasas para confirmar se a técnica utilizada funciona corretamente.

O algoritmo funciona da seguinte forma: inicialmente, é chamada a rotina que inicializa os parâmetros da otimização, gerando valores das variáveis de otimização aleatórios. Esses valores são introduzidos numa sub-rotina que gera as curvas respectivas a essas variáveis, e estas são atribuídas a uma vazão localizada na fronteira de controle do problema analisado. Para cada curva de vazão gerada, é calculada a função objetivo a partir dos resultados do programa de águas rasas, visto que a função objetivo compara a altura d'água desejada com aquela calculada para determinada região. A partir disso começa o processo evolutivo, até se achar uma curva ideal que minimiza a função objetivo. Então, para cada geração de um novo indivíduo, ou seja, um possível candidato a solução, deve-se gerar uma nova curva e se calcular a resposta do algoritmo de águas rasas tendo ela como vazão de controle, para, assim, se obter a função objetivo daquela candidata a solução. Um fluxograma deste processo pode ser visualizado na figura 41.

Figura 41: Fluxograma do processo de otimização da curva de controle



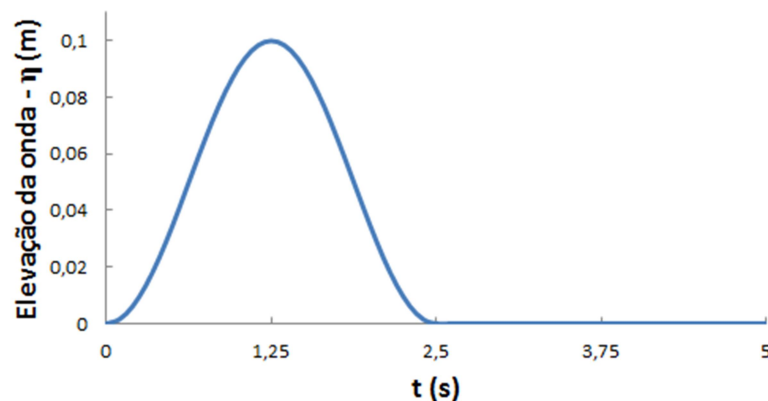
7.1 CANAL RETANGULAR COM VAZÃO DE CONTROLE A JUSANTE

O primeiro exemplo testado foi a aplicação do método num canal retangular por onde entra uma onda descrita pela figura 43. O canal possui 50 metros de comprimento e 2 metros de largura. A altura d'água é 10 m em todo o domínio. A fronteira de entrada, de controle e os pontos marcados estão localizados à esquerda, direita e no centro do domínio computacional, respectivamente. O objetivo do problema é que nos pontos marcados a elevação de onda seja sempre nula. Isso significa que será controlada a vazão de saída do canal, para que no seu centro não haja elevação de onda durante toda a simulação diante de um fluxo de entrada aplicado na outra extremidade do canal. A malha utilizada possui 400 elementos e 253 nós sendo que um esquema do problema pode ser visto na figura 42.

Figura 42: Exemplo 1 de controle



Figura 43: Fluxo de entrada do exemplo 1 de controle



Os nós participantes da linha central onde se busca atingir o objetivo são os nós 126, 127 e 128. Assim, foi atribuído que para cada passo de tempo e para cada um destes nós a altura objetivo fosse igual a $h_{obj} = 10,0$ m. Sabe-se que a resposta analítica para este exemplo é uma vazão na saída de mesma magnitude da vazão de entrada aplicada. Saindo a mesma quantidade de água que entra, fará com que no centro não haja variação da altura d'água.

Um detalhe importante para este exemplo, é que o que acontece na saída não afeta imediatamente a linha central do domínio, ou seja, pode-se encontrar uma curva que após um certo tempo da simulação não cause interferência nos pontos objetivos, pois a onda demora um tempo para chegar até lá. Para avaliar bem pelo menos 5 segundos de simulação como apresentado na curva de entrada, foram simulados 7.5 segundos do problema. Assim, a curva encontrada para os 5 primeiros segundos vai afetar todo domínio do problema, e é esta curva que busca-se para se comparar com a resposta analítica.

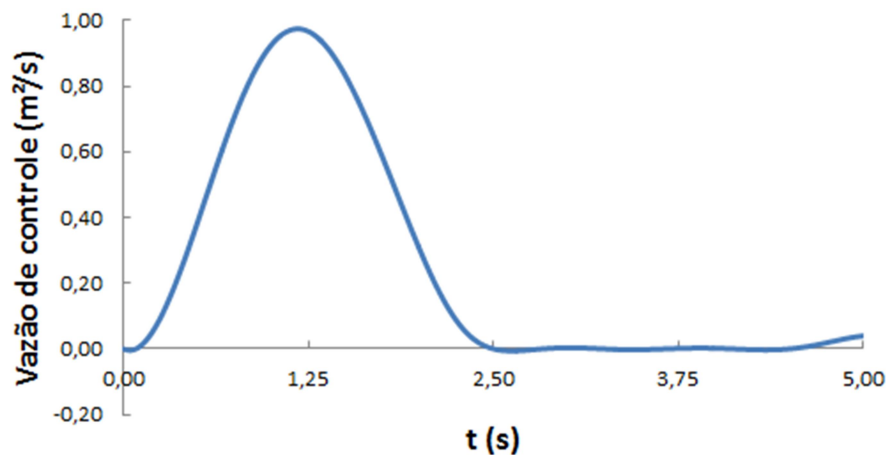
Para a curva NURBS que representa a vazão de controle, foram utilizados 19 pontos de controle igualmente espaçados no eixo x , que representam os 7.5 segundos simulados. Destes, apenas as coordenadas em y dos 13 primeiros, que representam os 5 primeiros segundos, serão otimizadas, enquanto que para as restantes foram atribuídos valores nulos de vazão. Para o primeiro ponto, localizado em $x = 0$, também foi atribuído o valor em $y = 0$ para respeitar a continuidade do problema, visto que a simulação parte de uma situação inicial de vazão nula naquele ponto. As outras 12 primeiras coordenadas em y dos pontos de controle são as variáveis do problema de otimização.

Sabendo que o valor máximo da vazão para gerar a onda do fluxo de entrada é da ordem de $1 \text{ m}^2/\text{s}$, definiu-se um domínio de busca das variáveis de otimização entre -0.5 e $2 \text{ m}^2/\text{s}$, visto que é sabido que a resposta vai estar em torno deste valor. Utilizou-se uma população de 50 indivíduos, apenas 1 complexo e $\beta = 10$, e após 323 gerações se chegou no resultado da curva de controle dada conforme pontos de controle apresentados na tabela 2 e plotada na figura 44. Neste trabalho o critério de convergência que considera a função objetivo encontrada como solução ótima é quando todos candidatos a solução de um subcomplexo atingem um mesmo valor de função objetivo. Depois desse momento não são gerados diferentes valores de função objetivo e a otimização termina. O número de gerações apresentado segue esse critério.

Tabela 2: Coordenadas dos pontos de controle da curva que representa a vazão ótima do exemplo 1 de controle

Pontos de controle	Coord. X	Coord. Y
1	0	0
2	0,417	-9.46E-02
3	0,833	1.30
4	1,25	1.18
5	1,67	0.74
6	2,083	8.10E-02
7	2,5	-5.03E-02
8	2,917	3.59E-02
9	3,333	-2.32E-02
10	3,75	1.10E-02
11	4,167	2.72E-03
12	4,583	-2.52E-02
13	5	6.80E-02

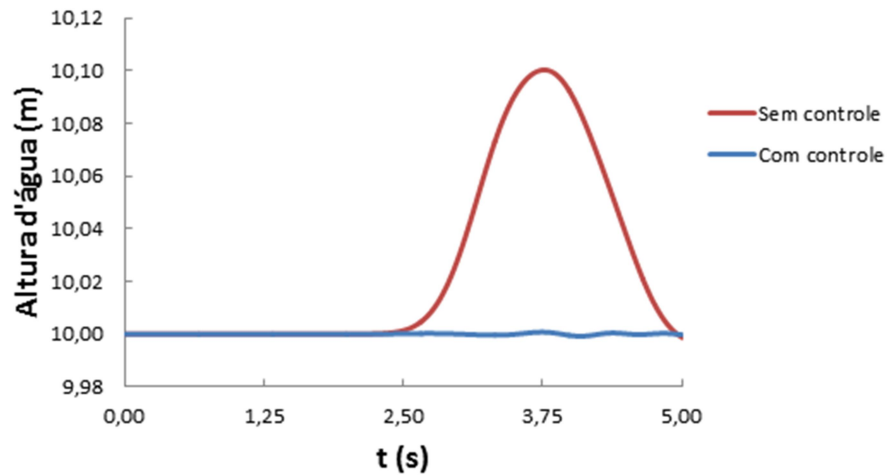
Figura 44: Vazão de controle do exemplo 1 de controle



Pode-se notar que a vazão começa a crescer quando chega perto de 5 segundos. Isso ocorre, como comentado anteriormente, devido à vazão que a partir desse ponto pode ter qualquer valor, não afetando a altura d'água nos pontos objetivos.

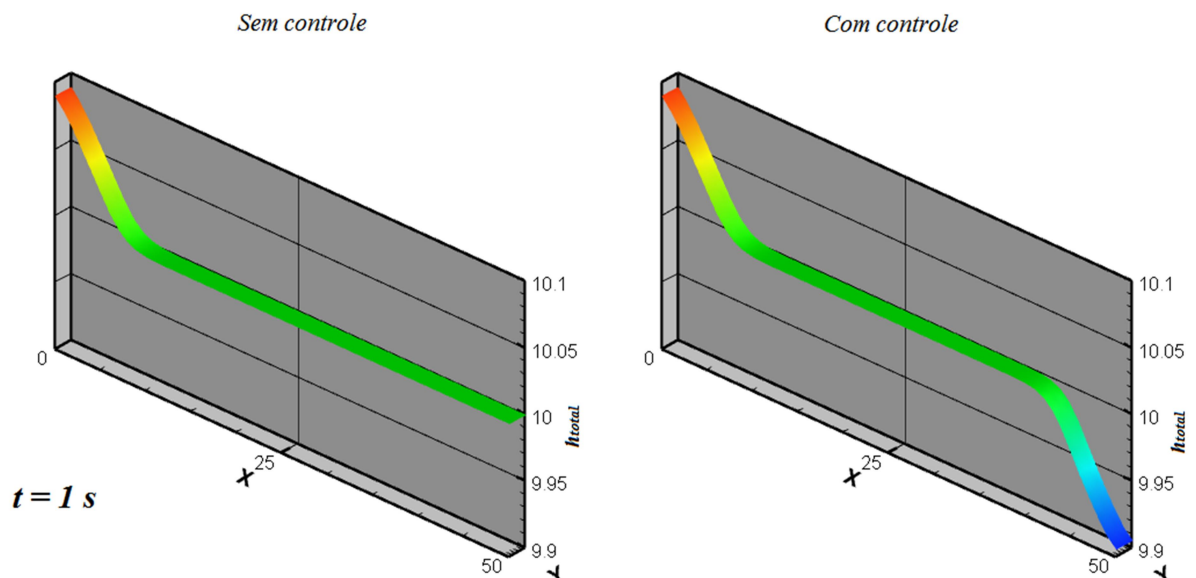
A função objetivo foi normalizada na forma $f^* = f/f_0$, onde f_0 é a função objetivo calculada sem controle. O valor da função objetivo encontrado foi de $f^* = 6,37 \times 10^{-5}$, sendo que a partir da 100ª geração a função objetivo já possuía valores menores do que 0,1. Isto significa que chegou muito próximo do resultado esperado, que seria 0. A variação da altura d'água no nó 127, que representa o nó central do domínio, está plotada na figura 45, com a representação anterior e posterior ao controle.

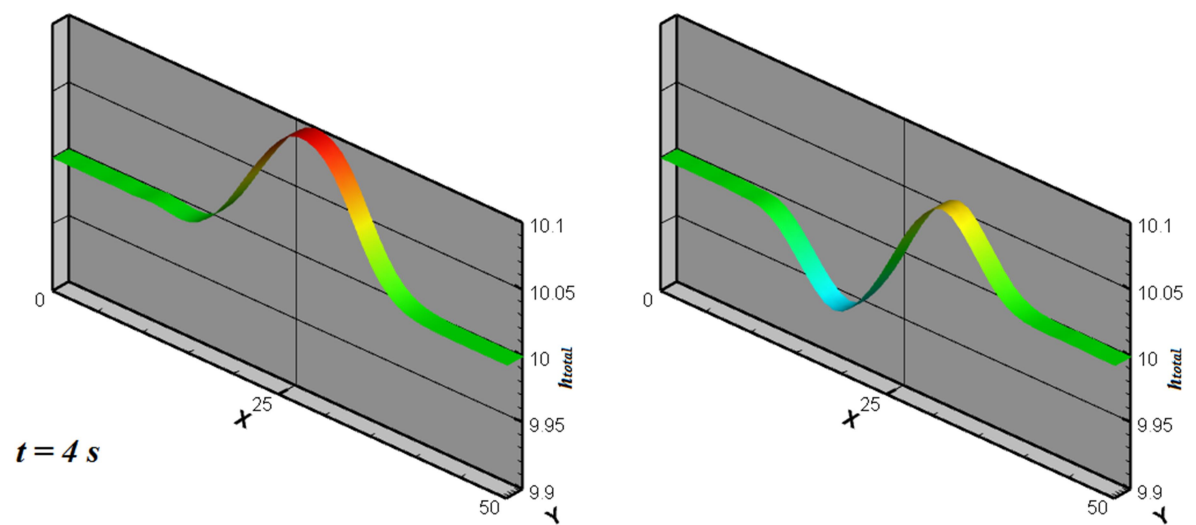
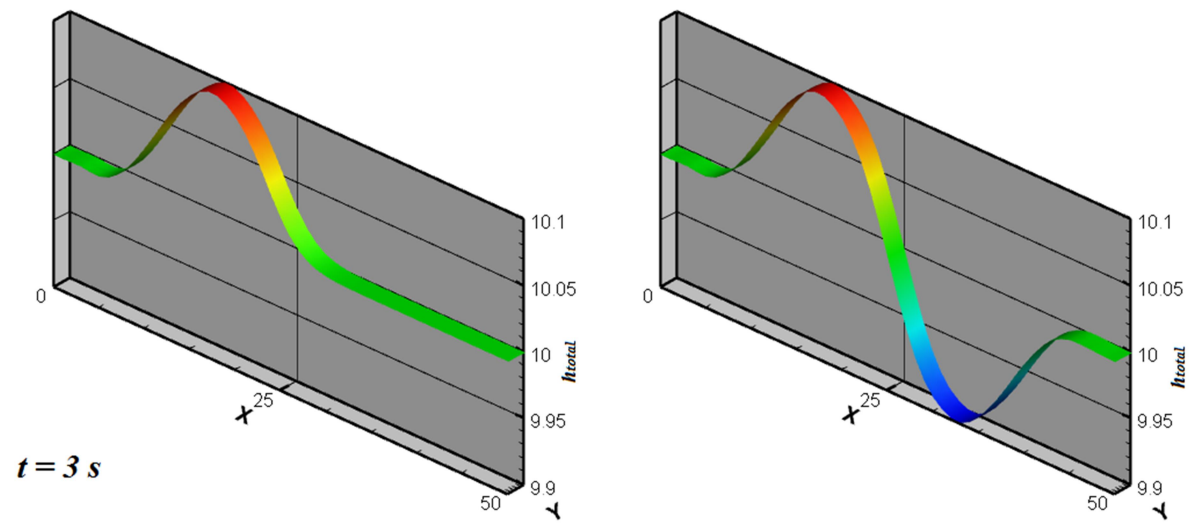
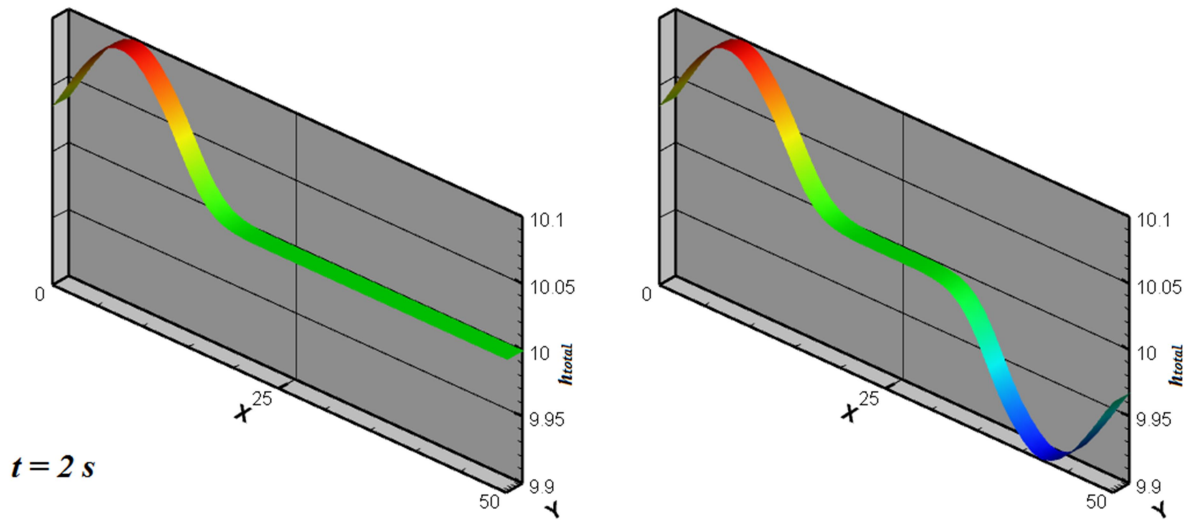
Figura 45: Altura d'água nos pontos objetivos do exemplo 1 de controle

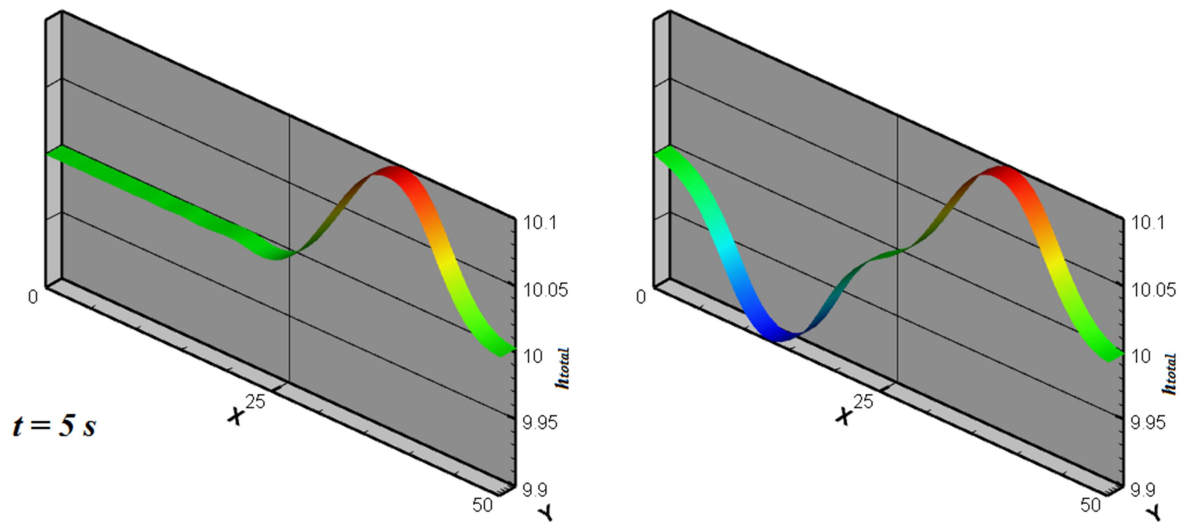


Pode-se ver na figura 45 que o objetivo foi atingido. A altura d'água nos pontos objetivos não varia durante a simulação. Na figura 46 é apresentada a simulação em 5 passos de tempo comparando o que acontece no domínio todo com a solução sem controle e a solução com controle. Mais uma vez se pode observar que o algoritmo de controle forneceu resultados satisfatórios.

Figura 46: Comparação entre alturas d'água do exemplo 1 com controle e sem controle







Simulações semelhantes a esta podem ser encontradas em Samizo (2012) e Samizo e Kawahara (2014). A vazão de controle encontrada condiz com o resultado esperado pela solução analítica, e o método utilizado mostrou funcionar bem.

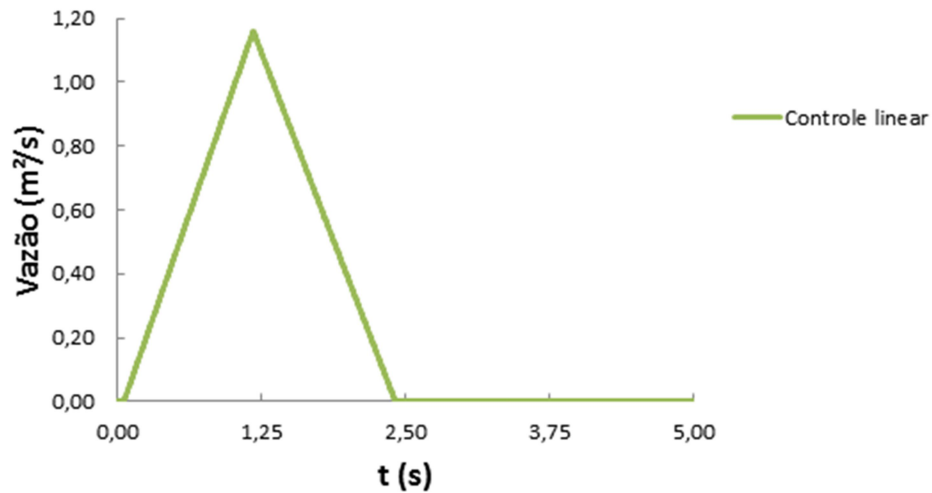
Para este exemplo também foram testadas as curvas lineares (de forma triangular) e quadráticas (de forma parabólica) a fim de comparar resultados. Nestes casos, optou-se por buscar um valor igual ou menor ao valor objetivo, utilizando penalidade para valores que resultarem acima deste, visto que se sabe que com estas formulações não é possível determinar a solução ideal. Então, definiu-se que o resultado se não for igual ao objetivo, deve ser menor que este.

Para a vazão na forma triangular, o domínio de busca dos valores dos parâmetros utilizado foi de 0 a 3 para a coordenada y , enquanto que para os valores da coordenada x , tomou-se o cuidado para que as coordenadas mantivessem uma ordem coerente, visto que $P1 \leq P2 \leq P3$, e espaçadas entre o valor total do tempo de simulação. Além disso, ao ter um número de variáveis de otimização menor (4 comparada a 12 da situação anterior), o número de indivíduos da população pode ser menor, e por isso, o número da população foi definido como 20. Os outros parâmetros de SCE-UA continuaram os mesmos.

Com apenas 128 gerações já houve uma convergência da resposta, como pode ser vista na figura 47. O valor da função objetivo normalizada encontrada foi igual a $f^* = 1,14 \times 10^{-2}$. Mesmo que não esteja tão próximo de zero como a resposta ótima encontrada por NURBS,

ainda assim é um resultado excelente, tanto do ponto de vista de rapidez para convergir, quanto do ponto de vista do resultado em si, que quase atingiu o objetivo.

Figura 47: Vazão de controle do exemplo 1 de controle na forma ótima linear.



Já para a forma parabólica, o domínio utilizado foi de 0 a 2, visto que para os dois exemplos anteriores o valor máximo de uma coordenada não passou de 2, e os outros parâmetros se mantiveram iguais aos utilizados para a forma linear. A otimização buscando a curva de controle em forma parabólica convergiu após 139 e a função objetivo resultou em $f^* = 5,39 \times 10^{-2}$. Este resultado foi o que menos chegou perto do objetivo $f^* = 0$, mas que se mostrou bom se essa curva for a desejada para se colocar em prática. Uma comparação das 3 curvas estudadas e respectivas alturas d'água nos pontos objetivos podem ser vistas nas figuras 48 e 49.

Figura 48: Comparação das curvas de controle do exemplo 1.

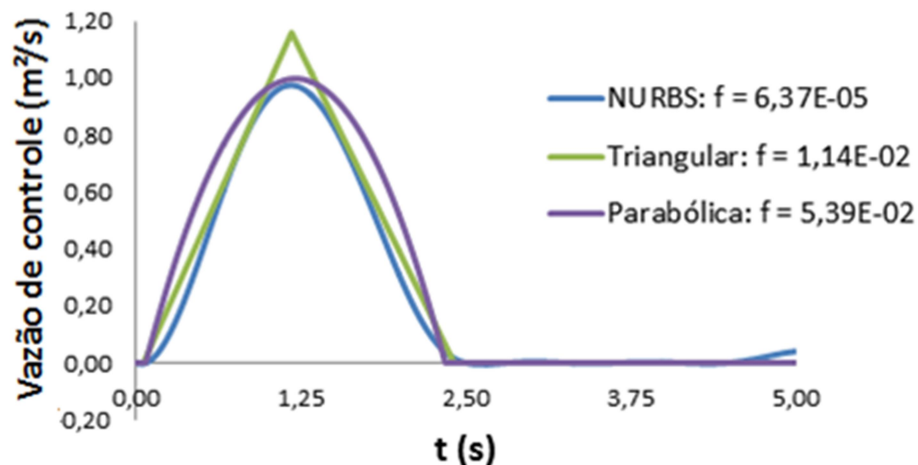
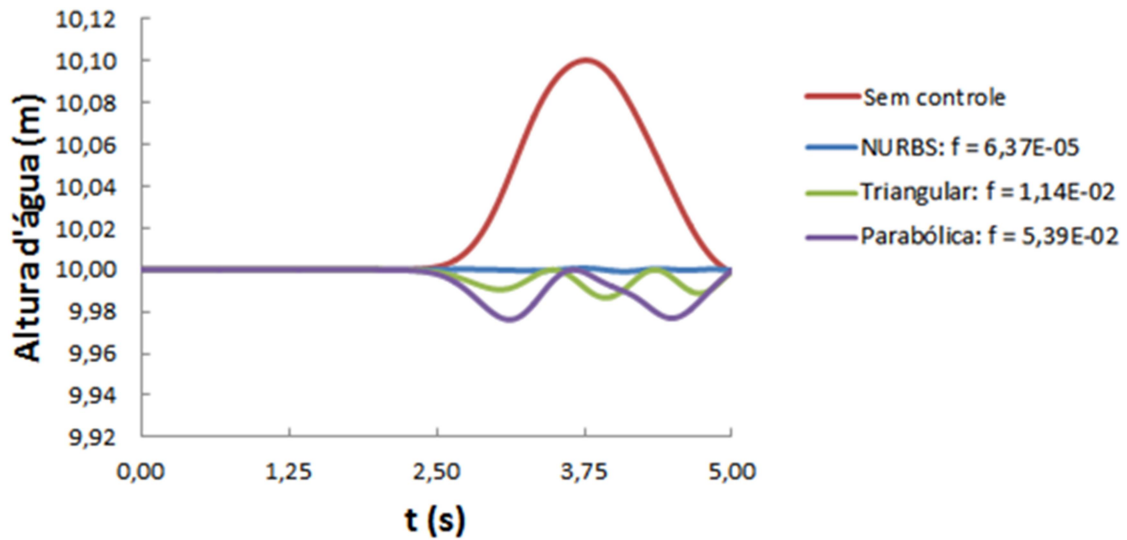


Figura 49: Comparação da altura d'água nos pontos objetivos para cada tipo de curva de controle do exemplo 1.



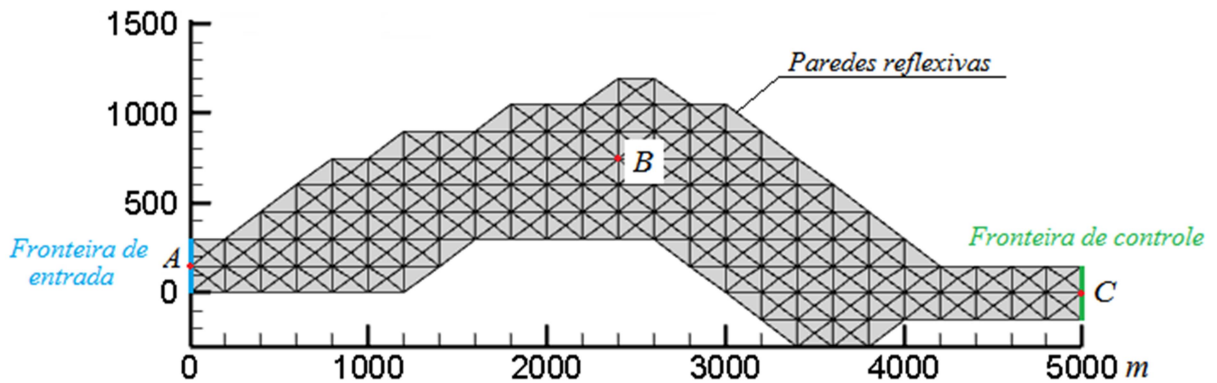
É importante ressaltar que a função objetivo é calculada de forma com que a elevação de onda seja mínima, tanto para valores maiores quanto para valores menores do que o nível inicial de 10 m. Além disso, buscou-se encontrar valores de altura d'água que se aproximem de 10 m, porém não ultrapassem este valor. Por este motivo, mesmo que a altura d'água encontrada por meios de uma vazão de controle de forma parabólica tenha apresentado valores menores, a função objetivo encontrada foi maior. O objetivo é se aproximar de 10 m, e não minimizar a altura d'água.

Apesar de que com uma curva NURBS foram encontrados valores de função objetivo menores àqueles encontrados pelas duas outras formulações num número de gerações menor, a convergência final demorou um número muito maior de gerações para ocorrer. Em situações mais complexas pode ser vantajoso definir a curva de otimização como linear ou parabólica, visto que o número de incógnitas é pequeno e fixo, diferente da metodologia que utiliza NURBS, que pode precisar de mais ou menos pontos de controle para encontrar um resultado ideal, e, conseqüentemente, um maior número de variáveis de otimização. Vale lembrar que quanto menor o número de variáveis de otimização, mais rápido a solução converge para um resultado. Além disso, como já foi mencionado, a aplicação prática de uma curva de primeira ou segunda ordem é muito mais simples do que controlar uma comporta que representa a vazão de controle gerada por uma curva complexa como pode ser encontrada por NURBS. No próximo exemplo serão avaliadas as três formulações novamente, em busca de mais resultados comparativos.

7.2 RESERVATÓRIO MORIYOSHIZAN

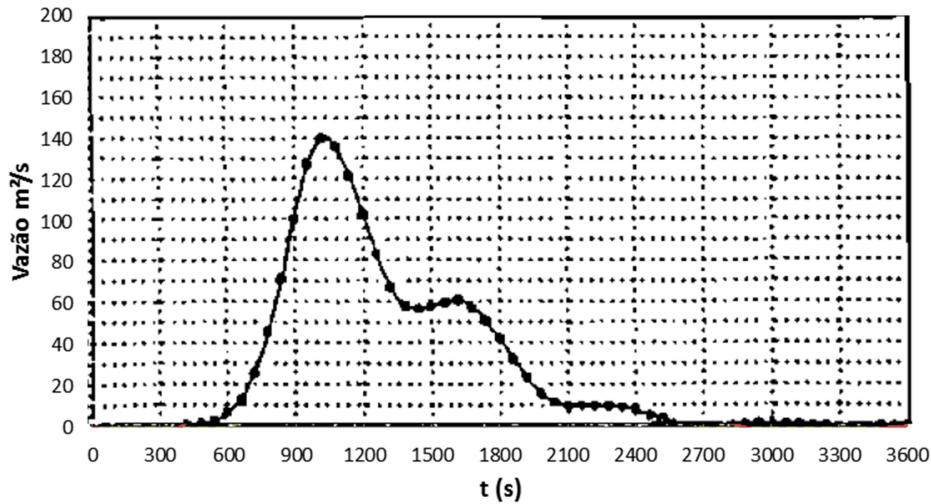
Neste segundo exemplo, estudou-se o reservatório Moriyoshizan, construído no Japão com propósitos de irrigação. Este reservatório foi estudado por Kawahara e Shimada (1994) com o mesmo intuito de controle ótimo, então os resultados encontrados podem ser comparados para validar o método utilizado neste trabalho. O reservatório possui geometria conforme a figura 50, onde já foi ilustrada a malha utilizada, com comprimento de aproximadamente 5 km e 700 m de largura. A malha possui 428 elementos e 242 nós e possui elementos de lado da ordem de 200 m (metade do tamanho utilizado pelos autores citados).

Figura 50: Geometria do exemplo 2 de controle.



Uma comporta está localizada no ponto C, e é a vazão que passa por esta que se deseja controlar quando entra no reservatório uma vazão localizada no ponto A. Quando a comporta está fechada, o reservatório pode não ter capacidade de suportar toda a água que chega. Então, o objetivo deste exemplo é que haja menor alteração possível do nível d'água em todo o reservatório, quando aplicada uma vazão de entrada típica conforme a figura 51. O nível d'água inicial é de 60 m de profundidade, com batimetria constante em todo domínio. O tempo de simulação é de 3600 segundos.

Figura 51: Vazão de entrada do exemplo 2 de controle.



Neste exemplo foi adicionado ao algoritmo de águas rasas um método de suavização dos resultados proposto por Thomas e Nithiarasu (2005), visto que a resposta sem este variava a cada passo de tempo em torno da resposta apresentada por Kawahara e Shimada (1994). A suavização proposta pelos autores é dada pela equação 48:

$$\Phi_s = \left[\frac{1}{1 + 0,5\alpha_s} \Phi + \frac{\alpha_s}{1 + 0,5\alpha_s} (M - M_D)M_D^{-1}\Phi \right] \quad (48)$$

Onde:

Φ : entidade a ser suavizada;

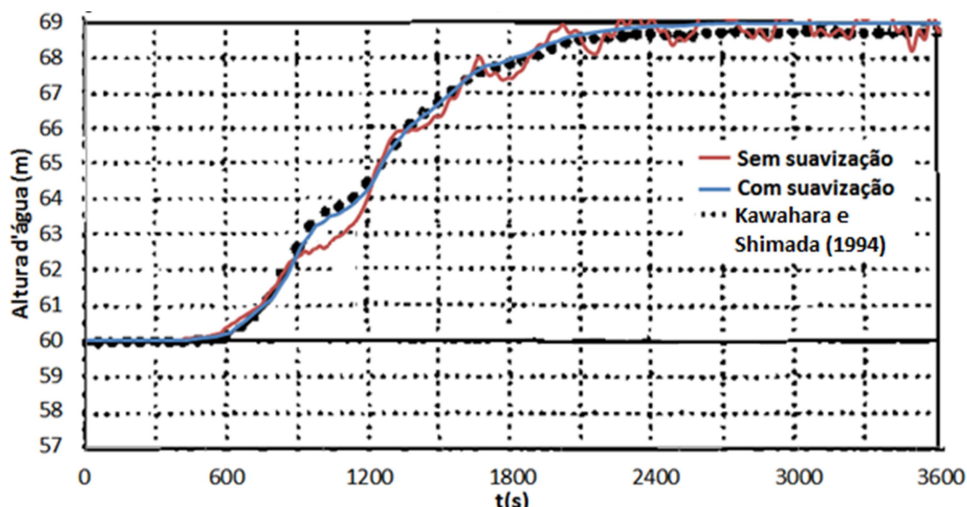
Φ_s : entidade suavizada;

α_s : parâmetro de suavização;

M_D : matriz M sem os termos não-diagonais.

Através de um coeficiente de suavização adequado, neste caso 0,5, os resultados encontrados ficaram bem condizentes ao que foi apresentado pelos autores. Na figura 52 pode ser visto os resultados com e sem a suavização para a altura d'água no ponto A, quando a comporta se encontra fechada, ou seja, resultado apenas do algoritmo de águas rasas.

Figura 52: Suavização dos resultados do exemplo 2 de controle no ponto A.



Para encontrar a curva ideal que minimiza a variação de altura d'água de todo o domínio do problema, os parâmetros de otimização utilizados foram: $m = 50$, $p = 1$ e $\beta = 10$, sendo que foram definidos 13 pontos de controle espaçados igualmente no eixo do tempo, dos quais 12 coordenadas em y são as variáveis de otimização. Assim como no exemplo anterior, a coordenada em y no instante igual a zero deve ser nula para respeitar a continuidade do problema. O intervalo que cada variável de otimização pode se encontrar está entre -100 e 400 m^2/s . Devido a este grande intervalo, algumas soluções aleatórias geradas no início da criação dos indivíduos da população podem resultar em valores incompatíveis ao algoritmo de águas rasas. Por exemplo, uma solução aleatória pode ter todas as variáveis com valores elevados, o que secaria o reservatório, e geraria um erro no programa. De forma a evitar este tipo de situação, e tentar acelerar o processo de otimização, a população inicial foi gerada buscando diferentes curvas que se aproximam da curva de entrada fornecida, dentro do espaço de tempo da simulação. Buscou-se atribuir valores para os indivíduos de forma a gerar diversas curvas com diferentes tempos de duração e diferentes magnitudes que por intuito podem se aproximar da forma final. Assim, após 453 gerações, foi encontrada uma curva ótima, conforme a figura 53, com valor da função objetivo normalizada $f^* = 1,79 \times 10^{-3}$, cujos pontos de controle são apresentados na tabela 3.

Figura 53: Vazão de controle do exemplo 2 de controle

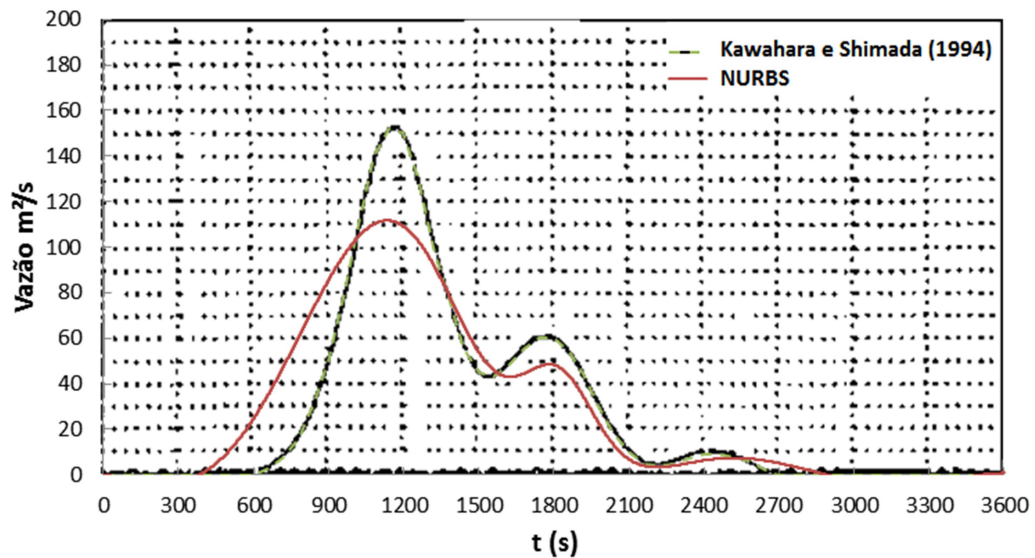


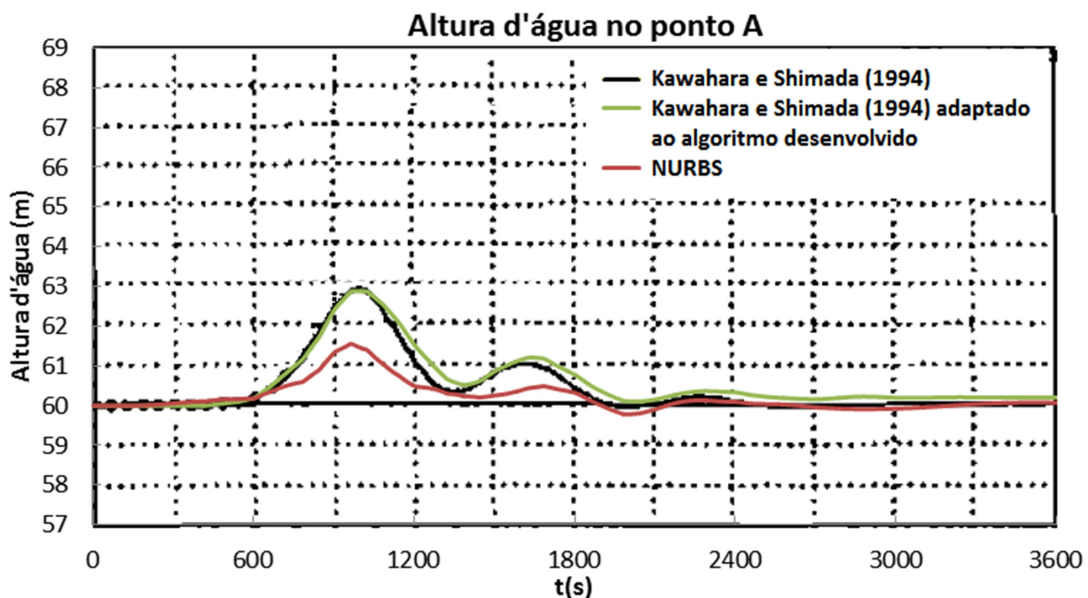
Tabela 3: Coordenadas dos pontos de controle da curva que representa a vazão ótima do exemplo 2 de controle.

Pontos de controle	Coord. X	Coord. Y
1	0	0
2	300	-3.54
3	600	-35.93
4	900	122.91
5	1200	206.27
6	1500	-37.13
7	1800	85.09
8	2100	-48.71
9	2400	27.88
10	2700	17.28
11	3000	-21.15
12	3300	-2.22E-02
13	3600	0.78

Pode-se notar que o resultado não foi igual ao encontrado por Kawahara e Shimada (1994), mas, como os métodos utilizados são diferentes, não é possível uma comparação direta dos resultados. Por isso, foi simulada uma vazão de saída conforme resultado encontrado pelos autores da referência citada, e verificada a altura d'água encontrada para ver se os resultados encontrados pelo programa desenvolvido seriam os mesmos apresentados por eles. Como pode ser visto na figura 54, para o ponto A, a altura d'água devido a uma vazão conforme considerada ótima encontrada por Kawahara e Shimada (1994) foi muito próxima ao que eles encontraram, confirmando novamente que o algoritmo de águas rasas funciona corretamente.

Juntamente com essas informações, estão apresentados os valores de altura d'água para a solução ótima encontrada por NURBS, e, pelo menos, para este ponto, a variação da elevação da onda foi menor que a encontrada pelos autores. A fim de comparação, foi calculado o valor da função objetivo para este caso onde foi utilizada a vazão considerada ótima pelos autores, resultando em $f^* = 7,47 \times 10^{-3}$, valor este maior do que o encontrado pelo algoritmo desenvolvido neste trabalho, ou seja, a resposta encontrada pela otimização por SCE-UA foi mais próxima do objetivo do que a dos autores.

Figura 54: Comparação de altura d'água no ponto A aplicando as vazões de controle ótima encontradas por Kawahara e Shimada (1994) e o algoritmo desenvolvido



Foram encontradas também as vazões de forma triangular (linear) e parabólica (quadrática) para este exemplo. Os parâmetros de otimização utilizados nestes casos foram $m = 30$, $p = 1$ e $\beta = 10$, num intervalo de valores entre 0 e 200 formando o domínio do problema. Vale lembrar que na forma em que foram implementadas as equações destes casos, houve um cuidado para que no eixo x , representado pelo tempo, as coordenadas sigam uma ordem coerente e se adaptem a todo tempo de simulação. Os valores da função objetivo encontrados estão apresentados na tabela 4, onde é possível fazer uma comparação com o que foi obtido por NURBS.

Tabela 4: Comparação das funções objetivo entre formas de curva de controle do exemplo 2.

Forma	Número de variáveis	Gerações	Função objetivo
Parábola	4	112	2,60E-03
Triangular	3	85	1,82E-03
NURBS	12	453	1,79E-03

Os resultados encontrados pela forma triangular obtiveram uma função objetivo muito próxima a encontrada por NURBS, sendo que precisou de quase 6 vezes menos gerações para convergir. Vale lembrar que na 244ª geração da otimização de uma curva NURBS, a função objetivo já era equivalente à função objetivo final da forma triangular. Mas ainda assim, como esperado, a forma triangular convergiu muito mais rápido, apresentando uma solução eficiente alternativa ao cálculo da forma ótima. Para este caso a forma parabólica não se mostrou tão eficiente quanto a forma triangular, mas também obteve convergência rápida se comparada à busca com NURBS, e apresentou bons resultados.

Neste exemplo, é impossível não haver o mínimo de elevação de onda, tanto positiva quanto negativa, dentro desse sistema ao se aplicar uma vazão na fronteira de entrada unida a um controle na saída. A função objetivo se resume a soma das diferenças ao quadrado entre a altura d'água calculada e a altura d'água objetivo em cada ponto do domínio e para cada passo de tempo. O objetivo desse problema é que essas elevações sejam mínimas, mas neste caso, devido à configuração do problema, a função objetivo nunca chegará em zero.

Os resultados das curvas de controle obtidas estão apresentadas na figura 55, e a comparação da altura d'água nos pontos A, B e C, apresentando os resultados quando a comporta se encontra fechada, quando se utiliza a curva de controle obtida por Kawahara e Shimada (1994), e as três formas de controle obtidas pelo algoritmo desenvolvido, estão apresentadas nas figuras 56, 57, 58 e 59.

Figura 55: Possíveis vazões de controle do exemplo 2 de controle

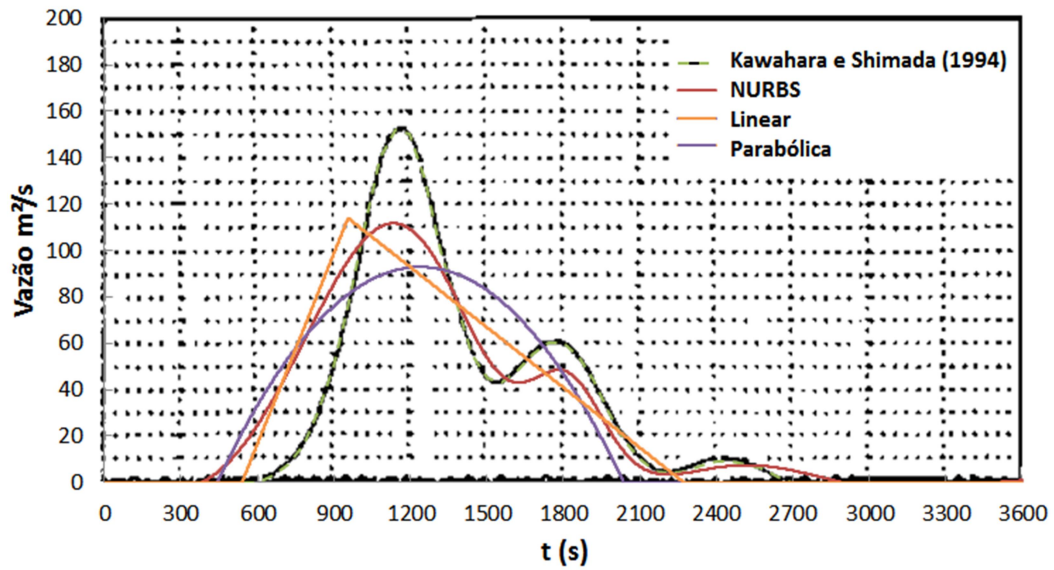


Figura 56: Legenda para as figuras 57, 58 e 59.

- Comporta fechada
- Comporta fechada Kawahara e Shimada (1994)
- Kawahara e Shimada adaptado ao algoritmo desenvolvido
- Kawahara e Shimada (1994)
- Forma parabólica
- Forma linear
- NURBS

Figura 57: Comparação da altura d'água no ponto A do exemplo 2 de controle

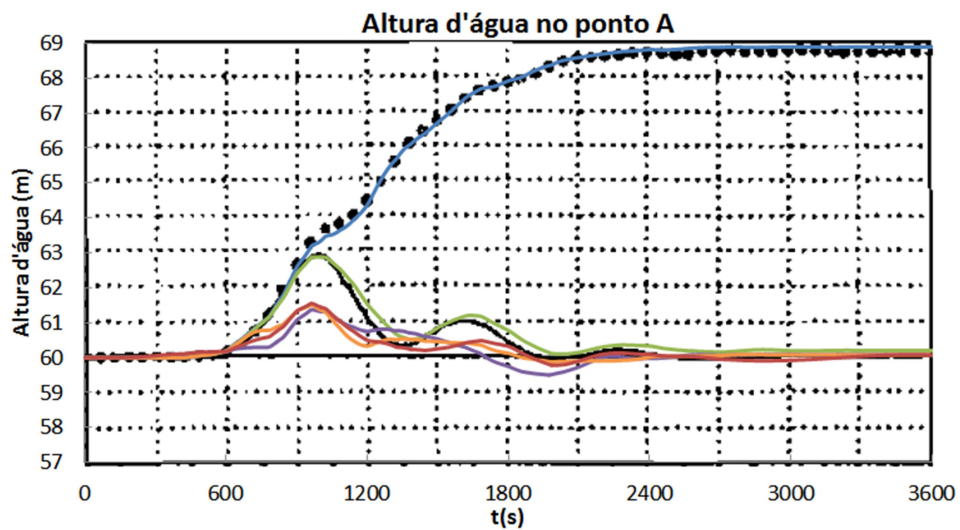


Figura 58: Comparação da altura d'água no ponto B do exemplo 2 de controle

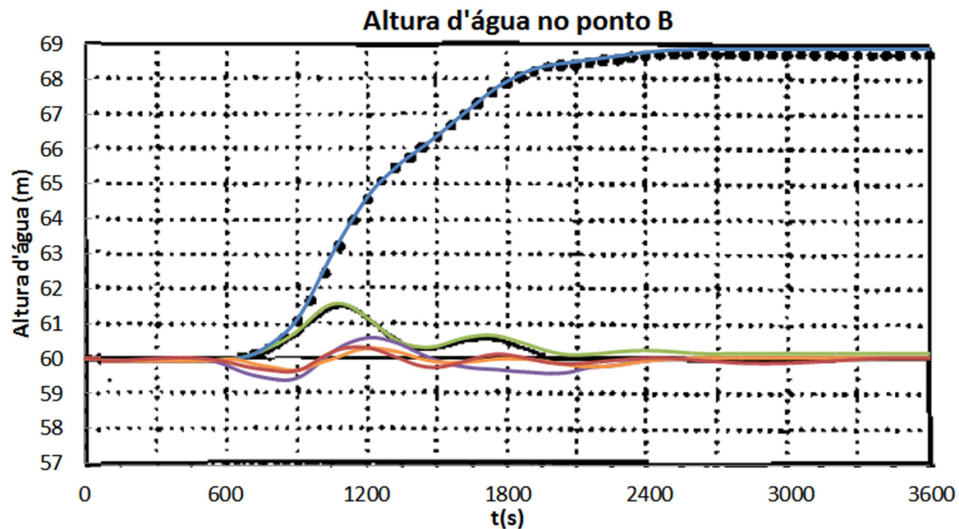
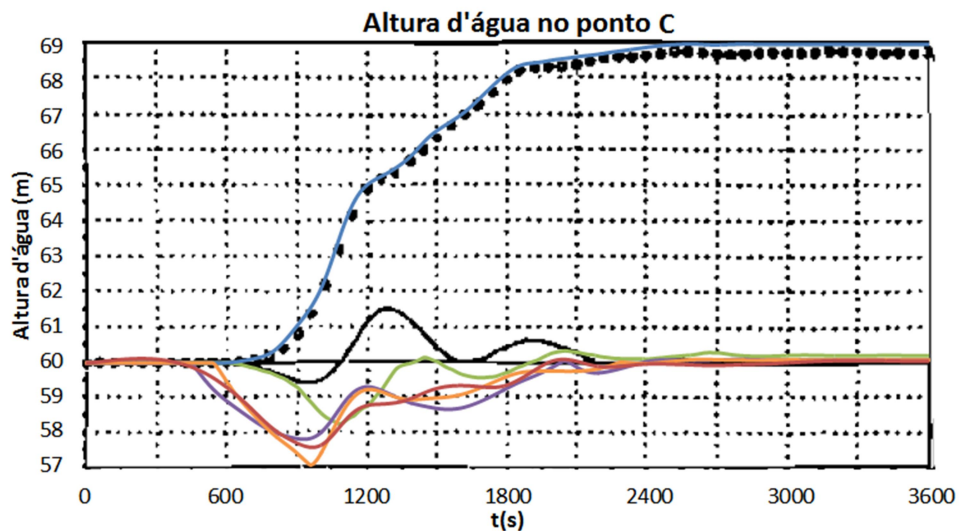


Figura 59: Comparação da altura d'água no ponto C do exemplo 2 de controle



No ponto C pode-se ver uma pequena discrepância dos resultados obtidos por Kawahara e Shimada (1994) em relação aos encontrados pelo algoritmo de águas rasas quando aplicada a vazão que por eles foi considerada ótima. Isso se deve a diversos fatores, como, o método utilizado para modelar as equações de águas rasas, unido ao fato de as curvas de entrada e saída serem uma aproximação visual das imagens fornecidas por estes autores (não há uma fórmula analítica) e a geometria exata do reservatório não foi informada pelos autores, sendo feita também uma aproximação visual. Essas pequenas diferenças podem ter apresentado estes erros, mas não comprometem o resultado final, no qual se conseguiu otimizar a vazão que deve sair pelo ponto C.

8 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

A súmula dos objetivos e resultados obtidos ao longo deste trabalho são resumidos neste capítulo, assim como são apontadas sugestões para trabalhos futuros.

8.1 CONCLUSÕES

Foi realizada a implementação de um código para a simulação de problemas hidrodinâmicos utilizando as equações de águas rasas, e, em seguida, a implementação de um algoritmo de controle, utilizando um método de otimização evolutivo SCE-UA acoplado a geração de curvas, para se determinar uma curva ideal de vazão, que dentro do sistema forneça os melhores resultados ao se buscar alcançar um objetivo pré-determinado por uma função performance. Ambos os códigos obtiveram resultados satisfatórios, atingindo os objetivos principais deste trabalho.

Para a implementação do algoritmo que resolve problemas de águas rasas, utilizou-se uma discretização espacial por meio de elementos finitos triangulares lineares, que são elementos muito simples e requereram a utilização de malhas mais refinadas para a resolução dos problemas. Isto acabou aumentando o tempo para finalizar a simulação destes, visto que o cálculo do passo de tempo de cada iteração depende do tamanho do menor elemento utilizado no domínio. Já na questão de discretização temporal, o método CBS implementado funcionou corretamente, e os resultados obtidos na resolução de diversos problemas, ao serem comparados com outros autores, se mostraram excelentes.

A metodologia utilizada para representar situações de inundação e secagem se mostrou eficiente. Foi possível reproduzir situações em que parte dos elementos se encontravam “secos”, e dependendo do fluxo dos seus elementos vizinhos, num próximo passo de tempo eles poderiam vir a se tornar molhados ou não. A reprodução de elementos secos e molhados é muito importante para abrir o leque de aplicação deste algoritmo, e os resultados mostram que o que foi implementado funciona corretamente.

Do ponto de vista das condições de contorno, sentiu-se a falta da condição de fronteira aberta. Existe a necessidade de um código especial para a aplicação deste tipo de fronteira. Num

primeiro momento, achou-se que não seriam necessárias tais condições, mas só foi possível a reprodução de alguns exemplos encontrados na bibliografia ao se aumentar o domínio do problema em questão. Dessa forma, a fronteira aberta era representada por um canal grande o suficiente para não perturbar o domínio de interesse de estudo. Apesar de a simulação demorar mais a ficar pronta devido à uma malha utilizada maior do que a seria necessária podendo aplicar este tipo de contorno, os resultados não foram afetados por isso.

O método de controle utilizado neste trabalho é diferente daqueles utilizados pela maioria dos autores que fizeram esse tipo de estudo. Buscou-se otimizar uma curva que representa a vazão de controle, afim de encontrar a vazão ótima que fornece os melhores resultados segundo uma função performance determinada. O método utilizado não se limita a encontrar apenas a vazão considerada ótima para resolver o problema. É possível também fornecer o tipo de curva que se deseja implementar, como as apresentadas na forma linear e quadrática. Por estas formas alternativas possuíram menos parâmetros de otimização do que NURBS, elas fornecem um resultado muito mais rápido do que a busca pela forma ótima por NURBS. Além desta vantagem, são curvas de vazões de fácil aplicação prática, com a abertura e fechamento de comportas, por exemplo. Por este motivo o método se torna uma boa alternativa, e não se limita a um tipo de curva de vazão.

A deficiência do método desenvolvido é a necessidade de muitas simulações do algoritmo de águas rasas para se chegar no resultado. Cada geração do processo evolutivo precisa que o algoritmo de águas rasas encontre a solução um número β de vezes, e isso torna o processo lento visto que alguns problemas de águas rasas estudados demoraram um número grande de horas para terminar apenas uma solução. Os exemplos apresentados foram simulados com uma máquina de processador Core-i5 com 8GB de memória RAM, e aqueles apresentados para a busca do controle ótimo possuíam simulações rápidas, da ordem de minutos. Porém, dependendo do número de gerações necessárias para se achar o resultado final, a vazão ótima pode demorar muito tempo (horas e até dias) para ser encontrada.

8.2 SUGESTÕES PARA TRABALHOS FUTUROS

A partir do que foi desenvolvido neste trabalho, algumas sugestões de melhorias e aplicações em trabalhos futuros podem ser sugeridas como segue:

- a) adição das equações de transporte: é possível acrescentar as equações de transporte ao algoritmo de forma simples, com intuito de fazer estudos de transporte de sedimentos e poluentes. Dessa forma, se poderia estudar o controle de efluentes despejados num rio ou lago, a partir de um tratamento que limitaria a quantidade de efluentes despejados naquela região. Da mesma forma que foi feito para a vazão de controle, poderia se buscar uma curva de concentração de poluentes despejados por tempo a ser otimizada.
- b) estudo de caso: o método de controle aplicado mostrou fornecer bons resultados para os exemplos testados, bem como o algoritmo que resolve as equações de águas rasas. A partir destes, seria possível fazer o estudo de uma região, e assim se propor uma solução mitigadora contra enchentes neste estudo de caso;
- c) implementação de adaptação de malha: seria interessante implementar uma rotina de adaptação de malhas, que possibilitam maior precisão quanto aos resultados, visto que fornecem o refinamento necessário nas regiões críticas, bem como podem ser eficientes do ponto de vista de rapidez para terminar uma simulação, com elementos refinados somente onde necessário, diminuindo a quantidade de elementos em malhas superestimadas.
- d) implementação de condição de fronteira aberta: como já foi comentado, sentiu-se a falta dessa condição de contorno para algumas simulações, a qual representa a condição de escoamento livre naquela fronteira, obrigando a geração de malhas maiores do que o necessário para resolver os problemas propostos. Seria interessante a implementação dessa condição de contorno para simplificar as malhas a serem geradas, bem como ter a possibilidade de reproduzir outros exemplos disponíveis na bibliografia que utilizam essa condição;
- e) computação paralela: visto que o algoritmo de controle tem a necessidade de realizar a simulação do problema de águas rasas diversas vezes, e por isso, há um consumo grande de tempo para se encontrar a solução, seria interessante encontrar formas que agilizassem esse processo. Uma forma poderia ser a implementação de computação paralela, na qual mais de um processador está trabalhando em um problema ao mesmo tempo. Acredita-se que para um estudo de caso essa implementação seria necessária.

REFERÊNCIAS

- AKANBI, A.; KATOPODES, N. Model for flood propagation on initially dry land. **Journal of Hydraulic Engineering**, V. 114, 689-706, 1988.
- ALCRUDO, F. Mathematical Modelling Techniques for Flood Propagation in Urban Areas. IMPACT Project Technical Report IMPACT Project WP3; Flood Propagation: Zaragoza, Spain, 2004.
- BUNYA, S.; KUBATKO, E.; WESTERINK, J.; DAWSON, C. A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations. **Computer Methods in Applied Mechanics and Engineering**, n. 198, p. 1548-1562, 2009.
- BRADFORD, S. B.; SANDERS, B. F. Finite-Volume Model for Shallow-Water Flooding of Arbitrary Topography. **Journal of Hydraulic Engineering**, V. 128, 289-298, 2002.
- BRUFAU, P.; VÁZQUEZ-CEDÓN, M.; GARCÍA-NAVARRO, P. A numerical model for the flooding and drying of irregular domains. **International Journal for Numerical Methods in Fluids**, V.39, 247-275, 2002.
- CALEFFI, V.; VALIANI, A.; ZANNI, A. Finite volume method for simulating extreme flood events in natural channels. **Journal of Hydraulic Research**, v. 41, n. 2, p. 167-177, 2003.
- CHORIN, A. J. Numerical solution of the Navier-Stokes equations. **Mathematics of computation**, v. 22, n. 104, p. 745-762, 1968.
- _____. On the convergence of discrete approximations to the Navier-Stokes equations. **Mathematics of Computation**, v. 23, n. 106, p. 341-353, 1969.
- DAVIES, R. Flooding in China, 2010. Disponível em <<http://floodlist.com/asia/flooding-china-2010>>. Acesso em: 21 de janeiro de 2016.
- DIMCIC, M. **Structural optimization of grid shells based on genetic algorithms**. 2011.
- DUAN, Q.; SOROOSHIAN, S.; GUPTA, V. Effective and eficiente global optimization for conceptual rainfall-runoff models. *Water resource Research*, Vol. 28, No. 4, pp. 1015-1031, 1992.
- FERREIRA, M. R. S. **Método dos Elementos Finitos aplicado às Equações de Águas Rasas**. 2013. 94 f. Dissertação (Mestrado em Engenharia Civil) – PPMaP, UFRGS, Porto Alegre. 2013.
- FROEHLICH, D. User Manual for FESWMSFST2DH: Two-dimensional depthaveraged flow and sediment transport model. Release 3. Report FHWA-RD-03-053, 2003.
- GID: The personal pre and post processor. Versão 12.0.3. Barcelona: CIMNE, 2015. Disponível em: <<http://www.gidhome.com>>. Acesso em: 10/03/2015.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison Wesley Longman Inc.: New York, 1989.

GOUTAL, N.; MAUREL, F. Proceedings of the 2nd workshop on dam-break wave simulation. **Electricité de France. Direction des études et recherches**, 1997.

HOLLAND, J. H. **Adaptation in natural and Artificial Systems**. Michigan: Ann Arbor: The University of Michigan Press, 1975.

KAWAHARA, M.; UMETSU, T. Finite element method for moving boundary problems in river flow. **International Journal for Numerical Methods in Fluids**, v. 6, n. 6, p. 365-386, 1986.

KAWAHARA, M.; UMETSU, T.; TANAKA, Y. Optimal Control of Flood Using Finite Element Method. **Proc. JSCE**, 446/1-19, 11s-20s, 1992.

KAWAHARA, M.; SHIMADA, Y. Gradient Method of Optimal Control Applied to the Operation of a Dam Water Gate. **International Journal for Numerical Methods in Fluids**, v. 19, p. 463-477, 1994.

LIANG, Q.; BORTHWICK, A. G. L. Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography. **Computers & Fluids**, v. 38, n. 2, p. 221-234, 2009.

LINN, R. V. **Simulação Computacional de escoamentos Compressíveis Utilizando Adaptação de Malhas Anisotrópica**. 2013. 180 f. Dissertação (Mestrado em Engenharia Civil) – PPGEC, UFRGS, Porto Alegre. 2013.

LYNCH, D. R.; GRAY, W. G. Finite element simulation of flow in deforming regions. **Journal of Computational Physics**, V.36, 135-153, 1980.

LUO, T.; MADDOCKS, A.; ICELAND, C.; WARD, P.; WINSEMIUS, H. World's 15 Countries with the Most People Exposed to River Floods. **World Resources Institute**, disponível em: < <http://www.wri.org/blog/2015/03/world%E2%80%99s-15-countries-most-people-exposed-river-floods>>. Acesso em 17 de agosto de 2015.

MEDEIROS S. C., HAGEN S. C. Review of Wetting and Drying Algorithms for Numerical Tidal Flow Models. **International Journal for Numerical Methods in Fluids**, v. 71, p. 473-487, 2012.

MESELHE, E.; HOLLY, F. J. Simulation of unsteady flow in irrigation canals with dry bed. **Journal of Hydraulic Engineering**, 119 (9), 1021-1039, 1993.

MINETTO, C. F. **Um estudo sobre curvas NURBS**. 2003. 114 f. Dissertação (Mestrado em Matemática Aplicada) – PPGMAp, UFRGS, Porto Alegre. 2003.

MORANDI-CECCHI, M.; VENTURIN, M. Characteristic-based split (CBS) algorithm finite element modelling for shallow waters in the Venice lagoon. **International journal for numerical methods in engineering**, v. 66, n. 10, p. 1641-1657, 2006.

NINDANG, S.; ALLEN, T. **Ahead of Flood Season, Thailand's Communities Demand Greater Preparedness**. Disponível em < <http://asiafoundation.org/in-asia/tag/thailand-floods/>>. Acesso em 21 de janeiro de 2016.

NIKOLOS, I. K.; DELIS, A. I. An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. **Computer Methods in Applied Mechanics and Engineering**, v. 198, n. 47, p. 3723-3750, 2009.

NITHIARASU, P. On boundary conditions of the characteristic based split (CBS) algorithm for fluid dynamics. **International journal for numerical methods in engineering**, v. 54, n. 4, p. 523-536, 2002.

ORTIZ, P.; ZIENKIEWICZ, O. C.; SZMELTER, J. CBS Finite Element Modelling of Shallow Waters and Transport Problems. **Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS**, Jyväskylä, p. 24-28, 2004.

_____. Hydrodynamics and transport in estuaries and rivers by the CBS finite element method. **International journal for numerical methods in engineering**, v. 66, n. 10, p. 1569-1586, 2006.

PEREIRA, F. F. **Modelo Hidrodinâmico e de Transporte Bidimensional de Grade Não Estruturada para Lagos Rasos**. 2010. 96 f. Dissertação (Mestrado em Engenharia Civil) – PPGIPH, UFRGS, Porto Alegre. 2010.

PIEGL, L.; TILLER, W. **The NURBS Book**. 2^a ed, Nova Iorque: Springer-Verlag, 1997.

ROGERS, B.; FUJIHARA, M.; BORTHWICK, A. G. L. Adaptive Q-tree Godunov-type scheme for shallow water equations. **International Journal for Numerical Methods in Fluids**, v. 35, n. 3, p. 247-280, 2001.

SAMIZO, Y. Optimal Control of Shallow Water Flows Using Adjoint Equation Model. **Kawahara laboratory**, 2012.

SAMIZO, Y.; KAWAHARA, M. Input disturbance predictive control of shallow water flows. **Computer Methods in Applied Mechanics and Engineering**, v. 278, p. 80-100, 2014.

SANDERS, B.; KATOPODES, N. Active Flood Hazard Mitigation. I: Bidirectional Wave Control. **J. Hydr. Engrg.**, ASCE, 125(10), p. 1057-1070, 1999.

SCHLICHTING, H. **Boundary-Layer Theory**. McGraw-Hill: New York, 7th edition, 1979.

SOROOSHIAN, S; GUPTA, V. K. Model Calibration In: Singh, V. J. (editor) Computer models of watershed hydrology. **Water Resources Publications**, Highlands Ranch. 1130 p., 1995.

SZYDŁOWSKI, M.; TWARÓG, B. Numerical investigation of flooding of real-topography developed areas following river embankment failure. **TASK Quarterly**, v. 10, n. 3, p. 321-338, 2006.

TEMAM, R. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I). **Archive for Rational Mechanics and Analysis**, v. 32, n. 2, p. 135-153, 1969.

THOMAS, C. G.; NITHIARASU, P. Influences of element size and variable smoothing on inviscid compressible flow solution. **International Journal of Numerical Methods for Heat & Fluid Flow**, v. 15, n. 5, p. 420-428, 2005.

VIRTUAL LIBRARY OF SIMULATION EXPERIMENTS. **Test Functions and Datasets**. Disponível em <<http://www.sfu.ca/~ssurjano/index.html>>. Acesso em 14 de dezembro de 2015.

WHITE, F. W. **Viscous Fluid Flow**. McGraw-Hill: New York, 2th edition, 1991.

WATANABE, M.; SANTOS, C. A. G.; SUZUKI, K. Modificação No Algoritmo Genético SCE-UA E Sua Aplicação A Um Modelo Hidrossedimentológico. **Revista Brasileira de Recursos Hídricos**. Vol 8, No, 1, 2003.

WATERWAYS EXPERIMENT STATION (WES). Floods resulting from suddenly breached dams. Miscellaneous Paper No, 2-374, **U. S. Army Corps of Engineers, Rep. 1: Conditions of minimum resistance**. Vicksburg, Miss, 1960.

WIJAYA, G. B.; BUI, T.; KANOK-NUKULCHAI, W. Numerical simulation of tsunami propagation using the characteristic-based split method. **Journal of Mechanics of Materials and Structures**, v. 3, n. 10, p. 1939-1962, 2008.

ZIENKIEWICZ, O. C.; TAYLOR, R. L.; NITHIARASU, P. The Finite Element Method For Fluid Dynamics. **Butterworth-Heinemann**, 6th edition, 2011.