

Universidade Federal Fluminense

ISAMARA LANDIM NUNES ARAUJO

Aplicação de Esquemas de Galerkin Descontínuo em
Problemas Hiperbólicos em 2D

VOLTA REDONDA

2020

ISAMARA LANDIM NUNES ARAUJO

Aplicação de Esquemas de Galerkin Descontínuo em Problemas Hiperbólicos em 2D

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Orientador:

Panters Rodríguez Bermúdez

Coorientador:

Eduardo Cardoso de Abreu
Yoisell Rodríguez Núñez

UNIVERSIDADE FEDERAL FLUMINENSE

VOLTA REDONDA

2020

Ficha catalográfica automática - SDC/BEM
Gerada com informações fornecidas pelo autor

A658a Araujo, Isamara Landim Nunes
Aplicação de Esquemas de Galerkin Descontínuo em
Problemas Hiperbólicos em 2D / Isamara Landim Nunes Araujo ;
Panters Rodríguez Bermúdez, orientador ; Eduardo Cardoso de
Abreu, coorientador. Volta Redonda, 2020.
102 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense,
Volta Redonda, 2020.

DOI: <http://dx.doi.org/10.22409/PPG-MCCT.2020.m.13968204778>

1. Método Galerkin Descontínuo. 2. Runge-Kutta. 3. Shallow
Water. 4. Leis de Conservação. 5. Produção intelectual. I.
Bermúdez, Panters Rodríguez, orientador. II. Abreu, Eduardo
Cardoso de, coorientador. III. Universidade Federal
Fluminense. Escola de Engenharia Industrial e Metalúrgica de
Volta Redonda. IV. Título.

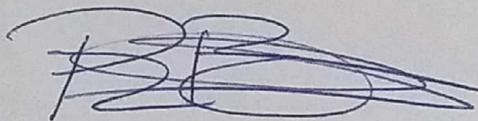
CDD -

Aplicação de Esquemas de Galerkin Descontínuo em Problemas
Hiperbólicos em 2D

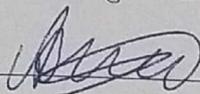
Isamara Landim Nunes Araujo

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

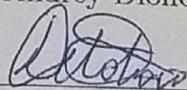
Aprovada por:



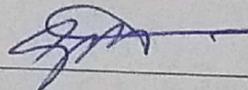
Prof. Panters Rodríguez Bermúdez, D.Sc. / UFF-VR



Prof. Andrey Dione Ferreira, D.Sc. / IFRJ-VR



Prof. Diomar Cesar Lobão, D.Sc. / UFF-VR



Prof. Gustavo Benitez Alvarez, D.Sc. / UFF-VR

Volta Redonda, 20 de Fevereiro de 2020.

Para os meus sobrinhos.

Agradecimentos

Agradeço à Deus por ter me dado forças e paciência diante das dificuldades para concluir esta etapa na minha vida.

Aos meu pais, Roselene e José, por me amarem incondicionalmente, me apoiarem nas minhas escolhas e pela confiança.

Ao meu companheiro, Luan, pelo amor e compreensão nos momentos mais difíceis.

Aos meus sobrinhos por me proporcionarem momentos de paz, ouvir vocês me chamando de “tia” sempre me deixava um bocadinho melhor e transformava o meu dia ruim em um dia maravilhoso.

Aos meus irmãos por estarem sempre ao meu lado, mesmo diante de tantas discussões.

Ao meu orientador Panters Rodríguez Bermúdez pelo incentivo, conhecimentos a mim passados e sugestões ao longo do meu projeto.

Aos meus co-orientadores Professor Eduardo Cardoso de Abreu por tudo que me foi ensinado e pelo esforço em me auxiliar, mesmo com a distância, e Professor Yoisell Rodríguez Núñez pelos conselhos, por me encorajar e pela amizade nestes anos. Agradeço a dedicação de ambos.

À todos os professores do MCCT pelos conhecimentos compartilhados comigo. Em especial, agradeço ao professores Diomar César Lobão e Tiago Araújo Neves por terem sido solícitos e contribuírem para realização deste trabalho.

Agradeço ao Ciro Díaz pela contribuição no aprimoramento do meu código computacional, obrigada por dedicar seu tempo nessa parte crucial do meu trabalho.

Gostaria de agradecer também ao Professor John Perez por ser sido atencioso e contribuído de forma significativa para este trabalho, cedendo seu código para que facilitasse a verificação dos meus resultados.

Aos amigos que fiz no MCCT, em especial Ana Beatriz que sempre teve um pensamento positivo, João Paulo que cooperou com a realização deste trabalho e não poderia esquecer da Larissa, que foi uma excelente amiga e confidente.

À toda a minha família por toda torcida e mensagens de carinho.

Finalmente, agradeço à CAPES pelo suporte financeiro, sem o qual a realização deste trabalho não teria sido possível.

Resumo

Neste trabalho utilizamos o método de elementos finitos Galerkin Descontínuo Runge-Kutta (RKDG) em uma malha triangular para obter soluções numéricas para problemas hiperbólicos em 2D, em particular para leis de conservação escalares e para o sistema de leis de balanço shallow water. Este método consiste na discretização espacial através do método Galerkin Descontínuo e na aplicação da aproximação Runge-Kutta de terceira ordem no tempo. A formulação do método Galerkin Descontínuo utiliza uma abordagem local, visto que a abordagem global, geralmente utilizada em outros métodos mais clássicos de elementos finitos como Galerkin Contínuo, não é capaz de lidar com a formação de choque e a interação das ondas. A utilização deste método requer a aplicação do limitador TVB para retirar as oscilações espúrias geradas, foi proposto neste trabalho um tratamento especial para este limitador nos elementos presentes na fronteira, e nos casos em que a variável conservada/balanceada é limitada será necessário também a aplicação do limitador “positive preserving”. Para gerar a malha triangular não estruturada utilizamos o gerador de malha desenvolvido por Per-Olof Persson [46]. Para calcular as integrais requeridas uma quadratura Gaussiana especial foi aplicada, de forma a satisfazer a condição CFL que garante a convergência do método numérico. Na obtenção das soluções numéricas, aproximações por polinômios lineares foram utilizadas, e as soluções obtidas foram comparadas qualitativamente com as soluções encontradas na literatura a fim de verificá-las. De maneira geral o método RKDG foi capaz de obter as soluções dos problemas hiperbólicos, capturando assim as ondas de rarefação e choque que surgem na solução, e mostrou ter a propriedade “well-balanced” necessária para resolver numericamente as leis de balanço. Porém, malhas com um número elevado de elementos são necessárias gerando assim um grande custo computacional.

Abstract

In this work, a discontinuous Galerkin Runge-Kutta finite element method was implemented on a triangle mesh to obtain numerical solutions for 2D hyperbolic problems, with emphasis on scalar conservation laws and the shallow water balance law system. This method consists on a spacial finite element discretization using Discontinuous Galerkin framework and on the application of a third order Runge-Kutta over time. Discontinuous Galerkin method's uses a local approach, since a global approach used by classical finite element methods like continuous Galerkin and others are not able to handle property with phenomena like shock formations and wave interactions. The implementation of this method requires the use of a TVB limitador in order to reduce spurious oscilations. In this work, a specific treatment for this limitador on the boundaries elements was adopted. For problems like shallow water equations where the high of water variable must be non-negative, we use a positive preserving limitador. This kind of limitador can be extended also for problems in which conserved/balanced variables must be limited. In order to generate a non-structured triangular mesh, we used a mesh generator developed by Per-Olof Persson [46]. To calculate the required integrals, while satisfying the CFL condition, a special Gaussian quadrature were applied. To obtain the numerical solutions aproximations by linear polynomials were used the obtained solutions were qualitatively compared with the solutions found on the literaturie for verification. The RKDG method was able to obtain the entropic solutions for the hyperbolic problems, it was able to capture the rarefaction waves and the shocks that arise on the Riemann solutions. The method has the property of being well-balanced, necessary to numerically solve the balance laws. However, a mesh with a high number of elements is needed to achieve those results, incurring on a high computational cost.

Palavras-chave

1. Método Galerkin Descontínuo
2. Runge-Kutta
3. Shallow Water
4. Leis de Conservação
5. Método bem balanceado.

Glossário

CFL	:	Constante Courant-Friedrichs-Lewy
DG	:	<i>Discontinuous Galerkin</i>
EDO	:	Equação Diferencial Ordinária
EDP	:	Equação Diferencial Parcial
LEH	:	<i>Lagrangian-Eulerian Method for hyperbolic conservation laws</i>
PVI	:	Problema de Valor Inicial
RK	:	<i>Runge-Kutta</i>
RKDG	:	<i>Runge-Kutta Discontinuous Galerkin</i>
TVB	:	<i>Total Variation Bounded</i>
TVBM	:	<i>Total Variation Bounded in the Means</i>
TVD	:	<i>Total Variation Diminishing</i>
TVDM	:	<i>Total Variation Diminishing in the Means</i>

Lista de Símbolos

$b(x, y)$: fundo marítimo (Eqs. shallow water)

b_i : baricentro do elemento K_i ($i = 0$ se refere ao próprio elemento e $i = 1, 2, 3$ aos elementos vizinhos)

e_K^i : aresta i do elemento triangular K , para $i = 1, 2, 3$

$f(u)$: função de fluxo na direção horizontal

$g(u)$: função de fluxo na direção vertical

g : constante da aceleração da gravidade

$h(x, y, t)$: função da altura da água (Eqs. shallow water)

k : grau dos polinômios utilizamos para aproximação a solução

K : elemento triangular

l_K^i : comprimento da aresta i do elemento triangular K , para $i = 1, 2, 3$

L_i :: coordenadas de área, $i = 1, 2, 3$

m_i : ponto médio da aresta i para um elemento K arbitrário, $i = 1, 2, 3$

M : matriz de massa

\vec{n}_K^i : vetor normal unitário em relação à aresta i do elemento K , para $i = 1, 2, 3$

N_i : função de forma

n : quantidade de nós dentro de cada elemento K

N : número de pontos de quadratura Gauss-Lobatto

NE : quantidade de elementos no qual o domínio computacional foi dividido

$s(h, b)$: termo fonte das equações shallow water

- S_K : conjunto de pontos de quadratura no elemento K
- \tilde{S}_K : conjunto de pontos de quadratura presentes na aresta do triângulo K
- \bar{S}_K : conjunto de pontos de quadratura presentes no interior do triângulo K
- t : variável temporal
- $u(x, t)$: no contexto geral é a variável escalar da quantidade conservada, exceto nas equações shallow water onde esta representa a velocidade da água na direção x
- \hat{u} : pontos de quadratura Gauss-Lobatto
- \bar{U}_K : valor médio de U sobre o elemento K
- $u_t, \frac{\partial u}{\partial t}$: derivada parcial de u em relação à t
- $u_x, \frac{\partial u}{\partial x}$: derivada parcial de u em relação à x
- v : pontos de quadratura Gauss
- $v(y, t)$: velocidade da água na direção y
- x : variável espacial com eixo horizontal
- w : peso do ponto de quadratura de Gauss
- \hat{w} : peso do ponto de quadratura de Gauss-Lobatto
- y : variável espacial com eixo vertical
- Φ : velocidade de choque
- Ω : domínio computacional
- T_τ : família de partições do domínio computacional
- τ_K : diâmetro do elemento triangular K
- τ : maior diâmetro dos elementos triangulares do domínio computacional
- Λ : parâmetro do limitador TVB
- C^0 : classe das funções contínuas
- C^2 : classe das funções com derivadas contínuas até segunda ordem

C_0^1 : classe de funções diferenciáveis com “suporte compacto”

L^p : espaço das funções p - integráveis, veja o Apêndice A

L^2 : espaço das funções quadrado integráveis

L_{loc}^1 : espaço das funções localmente integráveis, veja o Apêndice B

\mathbb{R} : conjunto dos números reais

\mathbb{R}^+ : conjunto dos números reais não negativos

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xvi
1 Introdução	17
1.1 Objetivos	22
1.1.1 Objetivo Geral	22
1.1.2 Objetivos Específicos	22
2 Leis de Conservação e Balanço	23
2.1 Leis de Conservação	23
2.1.1 Exemplo de aplicação física: Lei de Conservação de massa em 1D	25
2.1.2 Método das Características	26
2.1.3 Velocidade do Choque e Condição de Rankine-Hugoniot	30
2.1.4 Condição de Entropia	31
2.1.5 Problema de Riemann	33
2.2 Um Exemplo de Leis de Balanço	37
2.2.1 Equações Shallow Water	37
3 Um Estudo do Método de Elementos Finitos Galerkin-Descontínuo	42
3.1 Elementos Triangulares	43
3.1.1 Coordenadas Naturais	43
3.1.2 Funções de Forma	45

3.2	Método RKDG para Leis de Conservação	47
3.2.1	Limitador TVB	50
3.2.1.1	Tratamento do Limitador no Contorno	53
3.2.2	Limitador Positive Preserving para Leis de Conservação	53
3.2.3	O algoritmo para discretização temporal RK	54
3.3	Método RKDG para as Equações Shallow Water	55
3.3.1	O algoritmo para discretização temporal RK	57
3.4	Integração Numérica	60
4	Gerador de Malhas	65
4.1	O Algoritmo	66
4.2	Implementação	68
5	Verificação Computacional do Código e Experimentos Numéricos	72
5.1	Malhas	73
5.2	Teste do Limitador TVB	75
5.3	Leis de Conservação	80
5.4	Equações de Shallow Water	85
6	Conclusões e Trabalhos Futuros	90
6.1	Conclusões	90
6.2	Trabalhos Futuros	91
	Referências	92
	Apêndice A	97
	Apêndice B	98

Lista de Figuras

2.1	Fluxo do fluido em um determinado domínio.	25
2.2	Interseção das características.	27
2.3	Curvas características da equação de Burgers com dado inicial (2.18).	29
2.4	Trajetória $x(t)$ da descontinuidade no tempo.	30
2.5	Dados iniciais para problemas de Riemann.	34
2.6	Curvas características para equação de Burgers com $u_l = 0$ e $u_r = 1$ em (a). E o “preenchimento” das regiões vazia com uma onda de rarefação em (b).	35
2.7	Fluido em uma certa profundidade h	37
2.8	Definição do sistema de coordenadas e contornos.	38
3.1	Particionando o domínio em elementos triangulares.	42
3.2	Exemplo de mudança do sistema cartesiano para o sistema de coordenadas naturais.	44
3.3	Transformação das coordenadas cartesianas para coordenadas de área do ponto $p(x, y)$	44
3.4	Quantidade de nós necessária para polinômios lineares e quadráticos.	46
3.5	Ilustração dos dados necessário para o limitador TVB.	51
3.6	Ilustração do caso em que a aresta 1 do elemento K_0 esteja na fronteira. Note que $\alpha_1 = 1/2$ e $\beta_1 = 0$	54
3.7	Fluxograma do Método Galerkin Descontínuo.	58
3.8	Fluxograma do Método de Discretização Temporal Runge-Kutta.	59
3.9	Ilustração das três projeções para $k = 2$	61
3.10	Pontos de quadratura em um triângulo para $k = 2$	62

3.11	Pontos de quadratura utilizados.	64
4.1	Malhas geradas pelo Distmesh.	66
4.2	Código distmesh2d.m para gerar uma malha 2D.	69
4.3	Procedimento para gerar uma malha triangular não uniforme.	70
5.1	Malha com 5840 elementos ($\tau = 0.287$), onde $(x, y) \in [0, 1] \times [0, 1]$	73
5.2	Malha com 8072 elementos ($\tau = 0.242$), onde $(x, y) \in [0, 1] \times [0, 1]$	73
5.3	Malha com 11924 elementos ($\tau = 0.206$), onde $(x, y) \in [0, 1] \times [0, 1]$	74
5.4	Malha com 23282 elementos ($\tau = 0.143$), onde $(x, y) \in [0, 1] \times [0, 1]$	74
5.5	Malha com 36354 elementos ($\tau = 0.2264$), onde $(x, y) \in [-10, 10] \times [-10, 10]$	75
5.6	Malha com 64600 elementos ($\tau = 1.6942$), onde $(x, y) \in [-100, 100] \times [-100, 100]$	75
5.7	Condição Inicial.	76
5.8	Condição Inicial.	77
5.9	Resultados obtidos para o Teste 1, em (a) $t = 0.05$ e nas figuras (b)–(f) $t = 0.1$	78
5.10	Resultados obtidos para o Teste 2, em (a) $t = 0.015$ e nas figuras (b)–(f) $t = 1/12$	79
5.11	Condição Inicial.	80
5.12	Resultado obtido para o PVI (5.5)–(5.6) para $t = 1/12$	81
5.13	Condição Inicial.	82
5.14	Resultado obtido para o PVI (5.5)–(5.7) para $t = 0.5$	83
5.15	Condição Inicial.	84
5.16	Resultado obtido para o PVI (5.8)–(5.9) para $t = 0.1$	84
5.17	Malha com 189000 elementos, com $(x, y) \in [-100, 100] \times [-100, 100]$	86
5.18	Resultado para o Caso 1 para $t = 4.8027e - 04$, $\tau = 1.6942$ (64600 elementos) e $\Lambda = 0.01$	87
5.19	Condição Inicial.	88

5.20 Resultado obtido para o Caso 2 para $t = 0.1$, $\tau = 0.2264$ (36354 elementos) e $\Lambda = 0.001$	88
---	----

Lista de Tabelas

5.1	Tempo Computacional para as Leis de Conservação.	84
5.2	Erro ε_{L_1} para o problema estacionário.	85
5.3	Tempo Computacional para as Equações Shallow Water.	89

Capítulo 1

Introdução

A modelagem de diversos fenômenos físicos é frequentemente baseada no fato de que certas quantidades são conservadas ou balanceadas. As equações que expressam a conservação e balanceamento de uma quantidade física são conhecidas como leis de conservação e balanço, respectivamente.

Problemas de interesse prático que são modelados por leis de conservação ou balanço surgem em aplicações diversas como meteorologia, dinâmica dos fluidos, acústica, transporte de fluidos em meios porosos, modelagem de tráfego nas cidades, armazenamento de CO₂, hidrologia, oceanografia e muitas outras. Dentre as muitas leis de conservação e balanço conhecidas podemos citar as equações de advecção linear, de Burgers, de Buckley-Leverett e o sistema “shallow water” (SW). A equação de Burgers é a lei de conservação não linear mais simples, tipicamente usada como exemplo para descrever a formação de ondas de choques em distintas áreas da física. Já a equação de Buckley-Leverett é utilizada para modelar escoamento bifásico imiscível em meios porosos, com muitas aplicações na engenharia petrolífera. As equações shallow water são utilizadas para modelar problemas de ondas aquáticas onde o campo de fluxo tem uma componente que é desprezível em relação às outras duas, ou seja, a componente de velocidade vertical é pequena em relação à horizontal. Esse sistema têm sido amplamente utilizado na modelagem de diversos fenômenos, como fluxos em rios e áreas costeiras, reservatórios e escoamentos de canal aberto; fluxos de maré na regiões de águas costeiras e estuários; entre outros. Dentre as aplicações mais recentes podemos citar a modelagem do oceano em grande escala [36], circulação atmosférica [32], morfodinâmica de rios [37, 69], avalanches [31], vazão de barragem e escoamentos granulares [29, 30, 51]. As equações SW foram introduzidas no século XIX quando Adhémar Jean Claude Barré de Saint-Venant aproximou a propagação de marés através das equações shallow water unidimensionais

[52], que ficaram conhecidas como equações Saint-Venant.

Devido a grande aplicabilidade de leis de conservação e balanço a criação de métodos robustos, precisos e eficientes para resolver numericamente estes é de considerável importância, atraindo assim o interesse de muitos pesquisadores e profissionais. No entanto, a solução destas equações comumente tem descontinuidades, devido à não linearidade das funções de fluxo. O tratamento destas descontinuidades representa um desafio para os distintos esquemas numéricos que existem na literatura. Assim, o método deve ser robusto o suficiente para capturar as descontinuidades presentes na solução, e não introduzir oscilações espúrias que prejudiquem a qualidade da aproximação.

No caso das leis de balanço outra dificuldade adicional é o tratamento do termo fonte, esquemas numéricos tradicionais com uma manipulação direta do termo fonte não são capazes de preservar o equilíbrio entre o termo fonte e o fluxo, e geralmente falham em capturar bem o estado estacionário, sendo necessário que o método seja “well-balanced”. Se tratando das equações SW outra dificuldade importante frequentemente encontrada nas simulações é o aparecimento de áreas secas. Muitas aplicações em shallow water envolvem interfaces que se movem rapidamente entre áreas úmidas e secas, tais como quebras de barragens e ondas de inundação. Atenção especial deve ser dada, caso contrário os métodos podem falhar perto da frente seca/molhada e podem produzir uma altura de água negativa inaceitável, e tornar o sistema não hiperbólico e mal posto [65, 66].

Recentemente, vários métodos “bem balanceados” foram desenvolvidos com sucesso para as equações de Shallow Water [13, 14, 15, 40, 42, 43, 44, 16, 59, 60, 61, 62, 63, 64]. Dentre eles podemos destacar o método dos elementos finitos Galerkin Descontínuo (DG) que é capaz de obter a solução estacionária, e ao mesmo tempo preservar a positividade da altura da água. Uma vez que estamos trabalhando com problemas hiperbólicos, de forma a obtermos formação de choques e interação de ondas, o método Galerkin Contínuo não é capaz de lidar com essas soluções descontínuas. Com efeito o método Galerkin Descontínuo foi desenvolvido para corrigir essa falha, de maneira que o método DG use uma abordagem local e não a formulação global, geralmente utilizadas em métodos de elementos finitos clássicos. O uso de uma abordagem global, nesses métodos clássicos para problemas hiperbólicos, requer que o método seja estabilizado, tais métodos são estabilizados ad hoc, ou seja faz-se a inclusão de uma difusão numérica artificial para forçar uma suavidade na solução, porém essa formulação só funciona bem em alguns casos particulares. Por tal motivo, neste trabalho usamos o método Galerkin Descontínuo [65, 27, 68] que utiliza uma formulação local.

O método DG foi desenvolvido por Reed e Hill em 1973 como uma técnica para resolver problemas de transporte de nêutrons [48]. Em 1974, LaSaint e Raviart [38] apresentaram a primeira análise do método Galerkin Descontínuo para a equação Advecção Linear. Em seguida Johnson e Pitkaränta [33], Richter [49] e Peterson [47] também estudaram e aplicaram este método, todos estes foram para equações lineares.

Uma das dificuldades encontradas pelos pesquisadores foi a discretização implícita no tempo. A fim de evitar essa dificuldade, em 1982, Chavent e Salzano [18], construíram uma versão explícita do método DG para o caso de uma lei de conservação escalar unidimensional. Eles discretizaram o espaço usando o método DG com elementos lineares por partes e depois discretizaram o tempo usando o método de Euler avançado. Infelizmente uma análise clássica de von Neumann mostra que o método resultante é incondicionalmente instável quando a razão $\Delta t/\Delta x$ é mantida constante. Essa condição é muito restritiva quando se trata de problemas hiperbólicos, tornando assim este método inviável para estes problemas. Para melhorar a estabilidade deste esquema, em 1989, Chavent e Cockburn [17] o modificaram, introduzindo um limitador adequadamente definido, seguindo as idéias introduzidas em 1974 por van Leer [57]. Eles assim obtiveram um esquema que provou ser TVDM (*Total Variation Diminishing in the Means*) e TVB (*Total Variation Bounded*), desde que o número de CFL, $f' \frac{\Delta t}{\Delta x}$, seja menor ou igual a $1/2$ é garantida a convergência. Embora os resultados numéricos indiquem convergência para as soluções entrópicas corretas, o esquema é apenas de primeira ordem no tempo. Além disso, o limitador deve equilibrar as oscilações espúrias em regiões suaves causadas por instabilidade linear, afetando a qualidade da aproximação nessas regiões.

Essas dificuldades foram superadas por Cockburn e Shu em [25], onde foi introduzido o primeiro método RKDG. Esse esquema foi construído (i) mantendo a método DG linear por partes para a discretização do espaço, (ii) usando uma discretização explícita especial TVD do tipo Runge-Kutta de segunda ordem introduzida por Shu e Osher em 1988 [53] e 1989 [54], e (iii) modificando o limitador para manter a precisão formal do esquema nos extremos. O esquema explícito resultante foi então provado ser linearmente estável para números de CFL inferiores a $1/3$, de segunda ordem no espaço e no tempo, e TVBM. Os resultados numéricos mostraram convergência de segunda ordem em regiões suaves, transições de choque (geralmente em um ou dois elementos) sem oscilações e convergência para soluções entrópicas, mesmo para fluxos não convexos.

Em 1989, Cockburn e Shu [26] generalizaram essa abordagem e construíram métodos RKDG precisos de alta ordem para a lei de conservação hiperbólica escalar. Para criar

métodos RKDG da ordem $k + 1$, eles usaram (i) o método de discretização DG com polinômios de grau k para a discretização espacial, (ii) um Runge-Kutta TVD $(k + 1)$ para discretizar o tempo, e (iii) um limitador generalizado. Esse limitador foi cuidadosamente planejado para reforçar a propriedade TVBM sem destruir a precisão do esquema. Os resultados numéricos, para $k = 1, 2$, indicam ordem de $(k + 1)$ em regiões suaves, longe de descontinuidades, bem como transições de choque nítidas sem oscilações, convergência para as soluções entrópicas foi observada em todos os testes.

Esses esquemas RKDG foram estendidos a sistemas unidimensionais em 1989 por Cockburn, Lin e Shu [23].

A extensão do método RKDG ao caso multidimensional escalar foi feita em 1990 por Cockburn, Rou e Shu [20]. As principais contribuições dessa extensão são (i) algumas considerações de precisão e (ii) a extensão do limitador generalizado.

Em 1991, Cockburn e Shu em [24] iniciaram a extensão dos métodos RKDG para sistemas multidimensionais gerais, que foi concluída em 1998 em [27], onde aplicações às equações de Euler da dinâmica de gás foram exibidas. Uma das contribuições de [27] é a construção de um novo e prático limitador generalizado que funciona muito bem em triângulos e retângulos, e com elementos lineares e quadráticos por partes. Experimentos numéricos para as equações de Euler da dinâmica dos gases foram realizados em 1991 por Bey e Oden [11], em 1997 por Bassi e Rebay [6], em 1998 por Baumann e Oden [9] e por Warburton, Lomtev, Kirby e Karniadakis [55]. Para uma discussão mais detalhada do processo histórico de utilização do DG veja [21].

Desde então, os métodos dos elementos finitos Galerkin Descontínuo tornaram-se uma escolha particularmente popular e tem sido amplamente utilizada em uma ampla gama de aplicações. Devido a suas várias vantagens, além das já citadas, incluindo sua precisão, alta eficiência paralela, flexibilidade para adaptabilidade geométrica e malhas arbitrárias, o tornaram particularmente adequados para obter as soluções numéricas de leis de conservação e das equações shallow water.

Neste trabalho foi desenvolvido um algoritmo do método RKDG de acordo com Xing e Zhang [65], Zhang et al [68], Cockburn e Shu [27] para resolver leis de conservação em 2D, em particular a equação Advecção Linear e a equação de Burgers, e o sistema de leis de balanço Shallow Water em 2D. Dentre as dificuldades encontradas por nós, podemos destacar que devido ao fato do método DG utilizar uma formulação local isto gerou um grande custo computacional. A aplicação do limitador TVB é indispensável, visto que sem ele o método gera oscilações espúrias e se torna instável, porém sua implementação não

é trivial, pois informações importantes não foram encontradas detalhadas na literatura, como por exemplo a aplicação deste nos elementos presentes no contorno e como aplicá-lo quando utilizamos polinômios quadráticos. Para garantir a convergência da solução numérica, via condição CFL, uma quadratura especial é necessária.

A seguir explicaremos brevemente como este trabalho é organizado.

O capítulo 2 é dividido em duas seções, na primeira seção definimos alguns conceitos básicos sobre leis de conservação em 1D e 2D. Nesta seção também mostramos conceitos teóricos sobre leis de conservação em 1D que podem ser utilizados para entender o comportamento qualitativo das soluções em 2D, e apresentamos alguns exemplos de leis de conservação em 2D. Na segunda seção, mostramos como o sistema de leis de balanço shallow water é deduzido e mencionamos algumas de suas aplicações.

No capítulo 3, são mostrados alguns conceitos fundamentais sobre os métodos dos elementos finitos afim de se ter a base necessária para se aplicar o método RKGD. Em seguida, explica-se como o método Galerkin Descontínuo é aplicado localmente em cada elemento na discretização espacial bem como os limitadores utilizados e como a discretização temporal é realizada através do método Runge-Kutta de terceira ordem para leis de conservação e para as equações shallow water. Também neste capítulo, mostra-se como calcular numericamente as integrais necessárias, assim como obter os pontos de quadratura fundamentais para a integração, de forma que esta quadratura conserve a estabilidade do método com a condição CFL dada em [68], visto que estes pontos não são os usuais encontrados na literatura.

No capítulo 4, apresenta-se o gerador de malha utilizado, explicando como Per-Olof Persson desenvolveu o algoritmo deste gerador e alguns conceitos básicos para utilizá-lo.

No capítulo 5, são apresentados os resultados numéricos obtidos para leis de conservação escalares 2D, com fluxos e dados iniciais diferentes, e para as equações shallow water. Também neste capítulo apresenta-se alguns testes realizados para verificar se o tratamento do limitador TVB realizado nos elementos do contorno funciona, e como o parâmetro Λ do limitador TVB influencia a solução numérica obtida, visto que este é fundamental na retirada das oscilações espúrias.

No capítulo 6, são expostas as conclusões sobre este trabalho, como veremos o método RKDG é capaz de obter a solução entrópica, porém para isto é necessário malhas com um número relativamente alto de elementos, causando um alto custo computacional.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é resolver numericamente leis de conservação e as equações Shallow Water em 2D.

1.1.2 Objetivos Específicos

- Estudar ferramentas numéricas e teóricas para problemas hiperbólicos em 1D e 2D;
- Obter a solução numérica para problemas em 2D, através do método de elementos finitos Galerkin Descontínuo Runge-Kutta;
- Implementar o método, utilizando um software interativo de alta performance voltado para o cálculo numérico chamado MATLAB[®].

Capítulo 2

Leis de Conservação e Balanço

As leis de conservação e balanço são muito comuns na física visto que estas expressam a conservação e balanceamento, respectivamente, de uma certa quantidade. Sendo bastante utilizadas na modelagem de diversos fenômenos, que ocorrem por exemplo na dinâmica dos fluidos, acústica, engenharia hidráulica e petrolífera, entre outros.

O mais interessante sobre essas equações hiperbólicas é que a solução destas pode ser descontínua, embora a condição inicial seja suave. Isto faz com que seja necessário utilizar o conceito de solução fraca, e estudar em detalhe a formação de ondas de choque e a interação deste tipo de ondas e outras.

2.1 Leis de Conservação

Embora, neste trabalho, estejamos interessados em leis de conservação em 2D, a teoria sobre as leis de conservação em 1D nos dá uma base importante para poder entender as principais definições, como ondas elementares que costumam aparecer como parte das soluções devido à não linearidade (choques e rarefações), conceitos importantes como aquele de solução entrópica, e até em muitas ocasiões poder entender o comportamento qualitativo das soluções no caso 2D. Por isto nós dedicamos o início do presente capítulo a introduzir de maneira breve os principais conceitos e definições para o caso unidimensional.

Uma lei de conservação escalar em uma dimensão espacial apresenta a seguinte forma:

$$\frac{\partial}{\partial t}(u(x, t)) + \frac{\partial}{\partial x}(f(u)) = 0, \quad \forall t > 0, \quad x \in \mathbb{R} \quad (2.1)$$

com condição inicial

$$u(x, 0) = u_0(x), \quad \forall x \in \mathbb{R} \quad (2.2)$$

onde $u : \mathbb{R} \times \mathbb{R}^+ \mapsto \mathbb{R}$ representa uma quantidade conservada escalar e $f(u)$ corresponde à função de fluxo. Essas equações geralmente descrevem fenômenos de transporte.

Em geral, como será explicado neste capítulo, devido à não linearidade das funções de fluxo na maioria dos casos, soluções clássicas não existem para o problema (2.1)-(2.2), portanto o conceito de soluções fracas é utilizado. Porém, normalmente soluções fracas não são únicas, então para obter a solução fraca correta, ou seja, a solução fisicamente correta (entrópica) condições de entropia são utilizadas.

Definição 2.1 (*Solução fraca para leis de conservação em 1D*)

Seja $u_0 \in L^\infty(\mathbb{R})$. Então u é chamada de uma solução fraca de (2.1)-(2.2) se

$$\int_0^{+\infty} \int_{-\infty}^{+\infty} (u\phi_t + f(u)\phi_x) dx dt + \int_{-\infty}^{+\infty} \phi(x, 0)u_0(x)dx = 0 \quad (2.3)$$

para todas as funções continuamente diferenciáveis com suporte compacto $\phi \in C_0^1$, para detalhes veja [34].

Em geral, uma lei de conservação em 2D é dada pela seguinte equação:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0 \quad \text{em } \mathbb{R}^2 \times \mathbb{R}^+, \quad (2.4)$$

$$u(x, y, 0) = u_0(x, y) \quad \text{em } \mathbb{R}^2 \quad (2.5)$$

para dadas funções de fluxo f , g e condição inicial u_0 . Assume-se que f e g são suficientemente suaves.

Definição 2.2 (*Solução fraca para leis de conservação em 2D*)

Seja $u_0 \in L^\infty(\mathbb{R}^2)$. Então u é chamada de solução fraca de (2.4)- (2.5) se $u \in L_{\text{loc}}^1(\mathbb{R}^2 \times \mathbb{R}^+)$ e se

$$\int_{\mathbb{R}^2} \int_{\mathbb{R}^+} [u\varphi_t + f(u)\varphi_x + g(u)\varphi_y] dt d\mathbf{x} + \int_{\mathbb{R}^2} u_0\varphi(\cdot, 0)d\mathbf{x} = 0$$

para todo $\varphi \in C_0^\infty(\mathbb{R}^2 [0, \infty[)$, para detalhes ver [34].

Como no caso 1D as soluções fracas não são únicas para um dado problema, apesar de haver somente uma solução fisicamente correta (entrópica). Uma condição de entropia é usada para selecionar a solução correta. A condição de entropia pode ser derivada usando o critério do perfil viscoso, veja [35].

Teorema 2.1 (*Existência e unicidade de uma solução entrópica em 2D*)

Seja $u_0 \in L^1(\mathbb{R}^2) \cap L^\infty(\mathbb{R}^2)$. Então existe uma e somente uma solução entrópica u de (2.4)- (2.5) e $u \in C^0(L^1(\mathbb{R}^2), [0, T]) \cap L^\infty(\mathbb{R}^2 \times [0, T])$ [34].

Veja a prova em [35].

2.1.1 Exemplo de aplicação física: Lei de Conservação de massa em 1D

Considere um fluido que escoa através de um tubo uniforme com seção muito fina, de forma que o escoamento possa ser considerado unidimensional. Assume-se que as propriedades deste fluido são constantes através de cada seção do tubo. Sejam $u = u(x, t)$ a densidade (massa por unidade de comprimento), e $f(u(x, t))$ o fluxo (massa por unidade de comprimento, por unidade de tempo) do fluido. Considere um segmento arbitrário do tubo entre as seções transversais localizadas em $x = a$ e em $x = b$, veja a Figura 2.1.

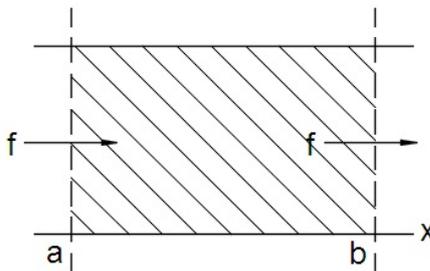


Figura 2.1: Fluxo do fluido em um determinado domínio.

Fonte: Imagem do autor.

A massa total de fluido dentro deste domínio no instante de tempo t é:

$$\text{massa} = \int_a^b u(x, t) dx. \quad (2.6)$$

Assume-se que a massa não é criada e nem destruída, então a massa neste domínio pode mudar apenas devido ao fluxo através dos pontos no contorno (diferença entre a quantidade que entra no domínio e a quantidade que sai do domínio). Assim, a taxa de mudança de massa no domínio é dada por:

$$\frac{d}{dt} \int_a^b u(x, t) dx = f(u(a, t)) - f(u(b, t)). \quad (2.7)$$

A equação acima é conhecida como forma integral da lei de conservação de massa.

Se u e f são funções suaves então podemos escrever a equação (2.7) da seguinte forma:

$$\frac{d}{dt} \int_a^b u(x, t) dx = - \int_a^b (f(u))_x dx,$$

$$\int_a^b u_t dx = - \int_a^b (f(u))_x dx.$$

Deste modo obtemos a seguinte equação:

$$\int_a^b [u_t + (f(u))_x] dx = 0 \quad \forall x \in (a, b). \quad (2.8)$$

A equação (2.8) também é uma forma integral da lei da conservação de massa. Se as funções envolvidas são diferenciáveis em todos os pontos, então podemos escrever a equação acima como:

$$u_t + (f(u))_x = 0. \quad (2.9)$$

Esta é conhecida como forma diferencial da lei de conservação da massa.

2.1.2 Método das Características

O método das características consiste em obter uma curva, chamada de curva característica, no plano (x, t) , ao longo da qual a EDP se transforma em uma EDO. As curvas características para uma lei de conservação na forma (2.1) são dadas pela equação diferencial

$$\frac{dx(t)}{dt} = f'(u(x, t)). \quad (2.10)$$

Podemos derivar u em relação a t ao longo de uma curva característica e obtemos

$$\frac{d}{dt} u(x, t) = \frac{dx}{dt} u_x + u_t = u_t + f'(u) u_x = 0. \quad (2.11)$$

Notamos que, de acordo com a equação acima, se u é uma solução clássica de (2.1) então u é constante ao longo das curvas características. Logo,

$$\frac{dx}{dt}(t) = f'(u(x_0)). \quad (2.12)$$

Segue-se que as curvas são retas dadas por:

$$x(t) = f'(u(x_0))t + x_0 \quad (2.13)$$

Como u é constante ao longo das curvas características $u(x, t) = u_0(x_0)$, onde x_0 pode ser obtido através da equação (2.13). Assim, a solução da lei de conservação (2.1) com dado inicial $u(x, 0) = u_0(x)$ é dada implicitamente por

$$u(x, t) = u_0(x - f'(u(x_0))t), \quad (2.14)$$

para t suficientemente pequeno.

Note que, para f não linear e convexa, $f \in C^2(\mathbb{R})$, $f''(u) > 0$, em geral, as características vão se cruzar. De fato suponha que existem pontos (x_1, t_1) , (x_2, t_1) , onde $x_2 > x_1$, tal que

$$u_1 := u(x_1, t_1) > u(x_2, t_1) =: u_2.$$

Então $f'(u_1) > f'(u_2)$ para todo $t > 0$. Consequentemente, as características se interceptam em algum $t_2 > t_1$, veja a Figura 2.2, contradizendo a suavidade de u , se u for decrescente.

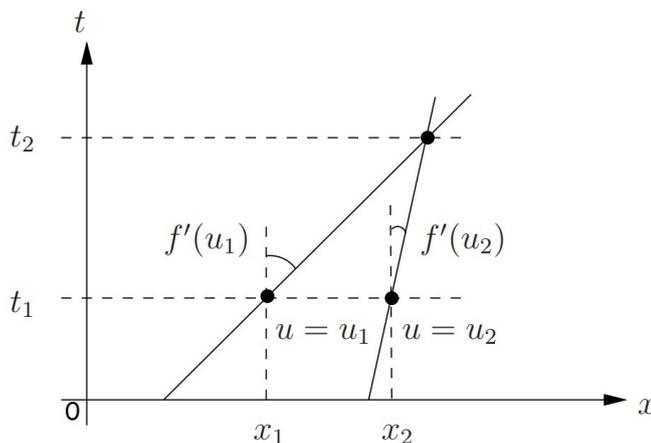


Figura 2.2: Interseção das características.

Fonte: van Duijn [56].

Lembrando que:

- u é constante ao longo de qualquer curva característica $x(t)$;
- $\frac{dx}{dt} = f'(u(x, t))$ para $t > 0$.

Mesmo quando $u_0 \in C^1(\mathbb{R})$, pode ocorrer a formação de um choque ou descontinuidade, com u_0 e u'_0 limitados sobre \mathbb{R} , usamos o teorema da função implícita para resolver esta equação para u como função diferenciável de x e t (com t suficientemente pequeno) [56]. Em particular

$$u_t = u'_0(x) [-f''(u)u_t t - f'(u)] \Rightarrow u_t = -\frac{f'(u)u'_0}{1 + f''(u)u'_0 t}.$$

e

$$u_x = u'_0(x) [1 - f''(u)u_x t] \Rightarrow u_x = -\frac{u'_0}{1 + f''(u)u'_0 t}.$$

Destas expressões, se $f''(u)u'_0 \geq 0$, então u_t e u_x permanecem delimitados: as características divergem e nenhuma descontinuidade ocorre. Por outro lado, se $f''(u)u'_0 \leq 0$, então as derivadas explodem quando $1 + f''(u)u'_0 t \rightarrow 0$. Como $f'' > 0$, isto ocorre se houver pontos onde $u'_0 < 0$, ou seja, se a função da condição inicial for decrescente em algum intervalo [56].

Exemplo: Equação de Burgers

$$\begin{cases} u_t + uu_x = 0 \\ u(x, 0) = u_0(x_0) \end{cases} \quad (2.15)$$

Usando as observações anteriores, o primeiro passo é definir uma curva característica para este problema. Então temos:

$$\frac{dx}{dt} = u, \quad x(0) = x_0. \quad (2.16)$$

Temos que ao longo da curva característica:

$$\begin{aligned} \frac{du}{dt} &= u_x \frac{dx}{dt} + u_t, \\ &= u_x u + u_t \\ &= 0. \end{aligned}$$

Então a solução de (2.16) é uma reta com inclinação $u_0(x_0)$,

$$x(t) = u_0(x_0)t + x_0. \quad (2.17)$$

Como u é constante ao longo da curva característica, temos que $u(x(t), t) = u_0(x_0)$, onde x_0 é dado implicitamente pela equação (2.17).

Pelo Teorema da função implícita para encontrar x_0 é necessário que

$$\frac{d}{dx_0}(u_0(x_0)t + x_0) = \frac{d}{dx_0}(u_0(x_0))t + 1 \neq 0.$$

Se $t = -1/u'_0(x)$ o teorema da função implícita não é válido para este valor de t .

Pode-se encontrar o valor de x_0 apenas para valores pequenos de t , sempre que $\frac{d}{dx_0}u_0(x_0)$ é limitado, pois evita fenômenos relacionado à velocidade infinita de propagação. Mas para t suficientemente grande podemos chegar a uma contradição. Tomemos, por exemplo, a condição inicial

$$u_0(x) = \begin{cases} 1, & \text{se } x < 0, \\ 0, & \text{se } x > 0, \end{cases} \quad (2.18)$$

então obtemos a seguinte solução:

- Se $x_0 < 0$, $u_0(x_0) = 1$ então $x = t + x_0$ e assim $u(x, t) = u_0(x - t)$.
- Se $x_0 > 0$, $u_0(x_0) = 0$ então $x = x_0$ e assim $u(x, t) = u_0(x_0) = 0$.

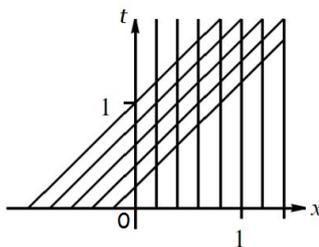


Figura 2.3: Curvas características da equação de Burgers com dado inicial (2.18).

Fonte: Leveque [39].

Logo, as curvas características se cortam a partir de $t = 0$, veja a Figura 2.3, e assim a descontinuidade já se propaga desde o tempo $t = 0$. Mesmo para o caso de uma condição inicial suave, se a função u_0 for decrescente em algum intervalo, podemos calcular o tempo de formação do choque (onde a solução deixa de ser clássica), nesse caso $u'_0(x)$ é negativo para algum x então há um tempo t_B para o qual se forma uma singularidade

$$t_B = \frac{-1}{\min\{u'_0(x)\}} \quad (2.19)$$

a partir da qual as características se interceptam e a solução é multivalorada. Isto nos leva a definição do tempo de existência de uma solução clássica e a necessidade de estender a definição de solução para considerar situações mais gerais.

2.1.3 Velocidade do Choque e Condição de Rankine-Hugoniot

Sabemos que, de acordo com a seção 2.1.1, a lei de conservação pode ser escrita na seguinte forma integral

$$\frac{d}{dt} \int_a^b u(x, t) dx = f(u(a, t)) - f(u(b, t)). \quad (2.20)$$

Suponha agora que $u(x, t)$ é descontínua através de uma curva suave $x(t)$ e que a equação diferencial seja válida em ambos os lados da curva, veja a figura 2.4.

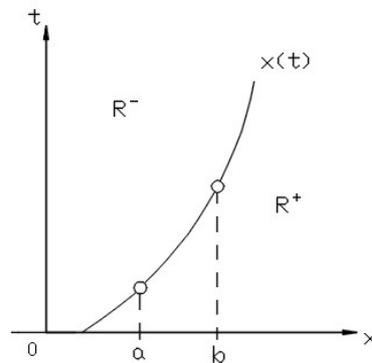


Figura 2.4: Trajetória $x(t)$ da descontinuidade no tempo.

Fonte: Imagem do autor.

Assim, a equação (2.20) pode ser escrita da seguinte forma:

$$\frac{d}{dt} \left\{ \int_a^{x(t)} u(z, t) dz + \int_{x(t)}^b u(z, t) dz \right\} = f(u(a, t)) - f(u(b, t)). \quad (2.21)$$

Para realizar os cálculos necessários vamos simplificar a notação definindo:

$$\int_{x_0}^x u(z, t) dz = U(x^-, t) \quad \text{e} \quad \int_x^{x_0} u(z, t) dz = -U(x^+, t) \quad (2.22)$$

Substituindo a definição (2.22) na equação (2.21), obtemos:

$$\frac{d}{dt} (U(x^-(t), t) - U(a, t)) + \frac{d}{dt} (U(b, t) - U(x^+(t), t)) = f(u(a, t)) - f(u(b, t)),$$

e assim,

$$U_t(x^-, t) + U_x(x^-, t) \frac{dx^-}{dt} - U_t(a, t) + U_t(b, t) - U_t(x^+, t) - U_x(x^+, t) \frac{dx^+}{dt} = f(u(a, t)) - f(u(b, t)),$$

onde x^- e x^+ são os limites pela esquerda e pela direita, respectivamente, de $x(t)$, veja a Figura 2.4.

De (2.22) temos:

$$U_t = \int_0^x u_t dz.$$

$$U_x = u(x, t) - u(x_0, t).$$

Desta forma,

$$\int_a^{x(t)} u_t dx + u(x^-, t) \frac{dx^-}{dt} + \int_{x(t)}^b u_t dx - u(x^+, t) \frac{dx^+}{dt} = f(u(a, t)) - f(u(b, t)).$$

Fazendo $a \rightarrow x^-(t)$ e $b \rightarrow x^+(t)$, obtemos:

$$\frac{dx}{dt}(u_l - u_r) = f(u(x^-(t), t)) - f(u(x^+(t), t)),$$

$$\frac{dx}{dt}(u_l - u_r) = f_l - f_r.$$

Consequentemente obtemos a condição de Rankine-Hugoniot:

$$\Phi = \frac{dx}{dt} = \frac{f_r - f_l}{u_r - u_l} \quad (2.23)$$

2.1.4 Condição de Entropia

Soluções fracas, em geral, não são únicas. Assim para uma lei de conservação com uma determinada condição inicial é possível que exista mais de uma solução fraca. Por outro lado, a condição de Rankine-Hugoniot não é suficiente para garantir a unicidade das ondas de choque como soluções fracas do problema de Riemann.

A unicidade da solução (fisicamente relevante) do problema de Riemann para leis de conservação pode ser obtida introduzindo-se restrições sobre as soluções fracas. Tais restrições são denominadas condições de entropia. Existem várias formas de selecionar as soluções entrópicas. A seguir expomos algumas delas [39].

Condição 1 (*Condição de entropia de Lax*)

Uma descontinuidade que se propaga com velocidade Φ , que satisfaz a condição de Rankine-Hugoniot (2.23), satisfaz a condição de entropia de Lax se

$$f'(u_l) > \Phi > f'(u_r). \quad (2.24)$$

Note que $f'(u)$ é a inclinação da curva característica, denominada então velocidade característica. Se f é convexa o critério de Lax se reduz simplesmente ao requisito:

$$u_l > u_r. \quad (2.25)$$

Condição 2 (Condição de entropia de Lax-Oleinik) A solução $u(x, t)$ é uma solução entrópica se todas as descontinuidades satisfazem

$$\frac{f(u) - f(u_l)}{u - u_l} \geq \frac{f(u) - f(u_r)}{u - u_r}, \quad \forall u_l < u < u_r. \quad (2.26)$$

Para f convexo este requisito (2.26) se reduz a (2.25).

Outra forma da condição de entropia é baseada na disseminação da característica em um leque de rarefação. Se $u(x, t)$ é uma função crescente de x em alguma região, então as características se espalham se $f'' > 0$. A taxa de propagação pode ser quantificada e dá a seguinte condição, também devido a Oleinik [39]. Assim, temos a seguinte condição.

Condição 3 (Condição de entropia de Oleinik)

A solução $u(x, t)$ é uma solução entrópica se houver uma constante $E > 0$ tal que para todos $a > 0$, $t > 0$ e $x \in \mathbb{R}$ tem-se

$$\frac{u(x + a, t) - u(x, t)}{a} \leq \frac{E}{t}. \quad (2.27)$$

Temos a seguinte definição:

Definição 2.3 Uma função suave $(\eta; q): \mathbb{R} \rightarrow \mathbb{R}^2$ se chama um par entrópico da equação (2.1), se qualquer solução continuamente diferenciável de (2.1) satisfaz a lei de conservação

$$\eta(u)_t + q(u)_x = 0.$$

No sentido distributivo. As funções η e q se chamam entropia e fluxo de entropia, respectivamente.

Note que se u é uma solução clássica do problema de Cauchy (2.1)-(2.2) e $(\eta; q)$ é um

par de entropia para (2.1), então

$$0 = \eta(u)_t + q(u)_x = \eta'(u)u_t + q'(u)u_x \quad (2.28)$$

Também u satisfaz $0 = u_t + f(u)_x = u_t + f'(u)u_x$ e multiplicando por $\eta'(u)$ se obtém

$$0 = \eta'(u)u_t + \eta'(u)f'(u)u_x. \quad (2.29)$$

De (2.28) e (2.29), pode-se concluir que $(\eta; q)$ é um par de entropia para (2.1) se e somente se $\eta'(u)f'(u) = q'(u)$.

Para o caso

$$\begin{cases} \eta(z) = |z - k|, \\ q(z) = (f(z) - f(k))\text{sgn}(z - k), \end{cases} \quad (2.30)$$

se têm a seguinte definição [41]

Definição 2.4 (Entropia de Krushkov) Uma solução entrópica de (2.1)-(2.2) é uma aplicação contínua $u : [0; \infty] \rightarrow L^1_{loc}(\mathbb{R})$ qual satisfaz (2.2) junto com

$$\int_0^{+\infty} \int_{-\infty}^{+\infty} \{|u - k|\phi_t + \text{sgn}(u - k)[f(u) - f(k)]\phi_x\} dx dt + \int_{\mathbb{R}} \{|u_0(x) - k|\phi(x, 0)\} dx \geq 0, \quad (2.31)$$

para qualquer função teste não negativa ϕ em $[0, \infty) \times \mathbb{R}$, e qualquer constante k . As entropias (2.30) se chamam entropias de Krushkov [41].

Definição 2.5 (Solução Entrópica de Krushkov em 2D) [34]

A solução fraca de (2.4)- (2.5) é chamada uma solução entrópica se temos para todo $\varphi \in C_0^\infty(\mathbb{R}^2 \times \mathbb{R}^+)$, $\varphi \geq 0$ e para todo $k \in \mathbb{R}$

$$\int_{\mathbb{R}^2} \int_{\mathbb{R}^+} \{\varphi_t |u - k| + \varphi_x \text{sign}(u - k) [f(u) - f(k)] + \varphi_y \text{sign}(u - k) [g(u) - g(k)]\} dt dx \geq 0.$$

Kruzkov [35] também mostra que toda solução entrópica pode ser considerada como um limite de viscosidade.

As soluções fracas que satisfazem alguns dos critérios de entropia, explicados acima, são chamadas soluções entrópicas fracas.

2.1.5 Problema de Riemann

Chama-se problema de Riemann um problema de valor inicial composto por uma lei de conservação e um dado inicial constante por partes, com uma única descontinuidade

[39]

$$u(x, 0) = \begin{cases} u_l & x < 0, \\ u_r & x \geq 0, \end{cases} \quad (2.32)$$

onde u_l e u_r são valores constantes. No caso, u_l é chamado de estado inicial à esquerda e u_r de estado inicial à direita, veja a Figura 2.5.

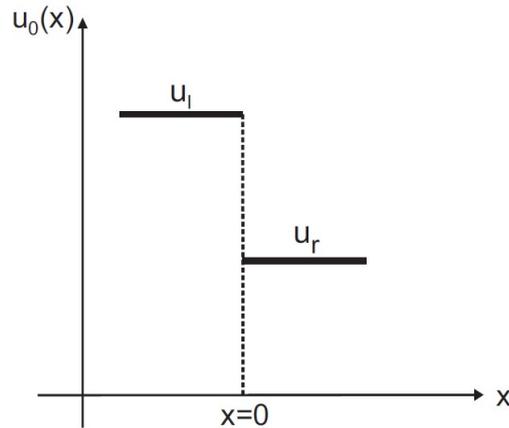


Figura 2.5: Dados iniciais para problemas de Riemann.

Fonte: Yamashita [67].

Como já foi dito na seção 2.1.4 (condições de Lax e Lax-Oleinik), se f é de classe C^2 e estritamente convexa, para $u_l > u_r$ então a única solução entrópica do problema de Riemann (2.1), (2.32) é uma onda de choque:

$$u(x, t) = \begin{cases} u_l, & x < \Phi t, \\ u_r, & x \geq \Phi t, \end{cases} \quad (2.33)$$

onde Φ é a velocidade do choque definida na seção 2.1.3. A onda de choque ocorre quando as curvas características se cruzam, veja a Figura 2.3.

Por outro lado, se $u_r > u_l$ então a única solução entrópica deste Problema de Riemann é a onda de rarefação:

$$u(x, t) = \begin{cases} u_l, & x < f'(u_l)t, \\ g(x/t), & f'(u_l)t \leq x \leq f'(u_r)t, \\ u_r, & x > f'(u_r)t, \end{cases} \quad (2.34)$$

onde $g = (f')^{-1}$. As ondas de rarefações surgem quando existem regiões no plano xt onde as curvas características não são definidas, veja a Figura 2.6. Uma das formas (não é

a única, porém é a única entrópica) de se “preencher” essas regiões, de modo a obter a solução fraca do PVI, são as chamadas ondas de rarefações.

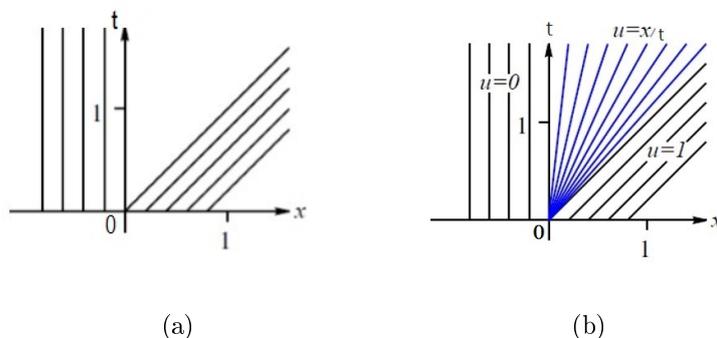


Figura 2.6: Curvas características para equação de Burgers com $u_l = 0$ e $u_r = 1$ em (a). E o “preenchimento” das regiões vazia com uma onda de rarefação em (b).

Fonte: de Lima [28].

A seguir serão apresentados alguns exemplos de leis de conservação bidimensionais, algumas das quais serão estudadas no presente trabalho.

Equação de Advecção Linear em 2D

Esta é lei de conservação linear muito simples amplamente conhecida na literatura e com grande utilidade.

$$\frac{\partial u}{\partial t} + v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} = 0, \quad (2.35)$$

onde v_1 e v_2 são constantes que denotam as velocidades, com que a onda se propaga, nas direções x e y , respectivamente.

A solução exata desta equação é a translação da condição inicial na direção do vetor de velocidades (constantes), se para $t = 0$ temos que

$$u(x, y, 0) = u_0(x, y),$$

então

$$u(x, y, t) = u_0(x - v_1 t, y - v_2 t). \quad (2.36)$$

Podemos obter essa solução de forma análoga ao método das características em 1D explicado anteriormente, veja a equação (2.14).

Equação de Burgers em 2D

Um exemplo de grande importância dentro das leis de conservação, é a Equação de Burgers, também conhecida como Equação de Burgers sem viscosidade. Esta equação é

utilizada em várias áreas da matemática aplicada, tal como mecânica dos fluídos, acústica não linear, dinâmica do gases, fluxo do tráfego nas cidades. A equação de Burgers foi introduzida primeiro por Harry Bateman em 1915 [7] e depois estudada por Johannes Martinus Burgers em 1948 [12].

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) + \frac{\partial}{\partial y} \left(\frac{u^2}{2} \right) = 0. \quad (2.37)$$

Equação de Buckley-Leverett em 2D

A equação de Buckley-Leverett é uma lei de conservação usada para modelar o fluxo bifásico em meios porosos. Ela descreve um processo imiscível de deslocamento, como o deslocamento de óleo pela água. Este modelo matemático pode ser aplicado na recuperação de óleo em aplicações de engenharia.

$$\frac{\partial s_w}{\partial t} + \frac{\partial f(s_w)}{\partial x} + \frac{\partial g(s_w)}{\partial y} = 0, \quad (2.38)$$

com

$$f(s_w) = g(s_w) = \frac{s_w^2}{s_w^2 + \frac{\mu_w}{\mu_o}(1 - s_w)^2}, \quad (2.39)$$

onde μ_w e μ_o são as viscosidades da água e do óleo, respectivamente.

Se considerarmos a ação da força gravitacional sobre o fluxo, as funções de fluxos são as seguintes:

$$g(s_w) = f(s_w) \{1 - C_g(1 - s_w)^2\}, \quad (2.40)$$

onde C_g é uma constante que envolve o termo gravitacional, veja [4, 50] para detalhes.

2.2 Um Exemplo de Leis de Balanço

2.2.1 Equações Shallow Water

As equações “Shallow Water” são um conjunto de leis de balanço que descrevem o comportamento de um fluido, geralmente a água, de uma certa profundidade h , veja a Figura 2.7. Estas com uma topografia do fundo não plana desempenham um papel crítico na modelagem e simulação de fluxos em oceanos, rios, lagos e áreas costeiras. Elas tem ampla aplicações na engenharia hidráulica, modelagem atmosférica e podem ser usadas para prever marés, níveis de tempestades e mudanças no litoral de furacões, e até mesmo para simular a propagação do tsunami [65, 3].

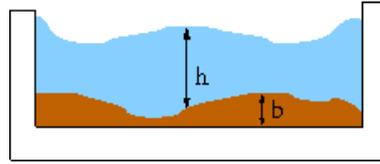


Figura 2.7: Fluido em uma certa profundidade h .
Fonte: SWE [3].

Essas equações são derivadas das equações de conservação da massa e do momento linear (as equações de Navier-Stokes) [58]. A equação de conservação do momento é expressada como:

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2) + \frac{\partial}{\partial y}(\rho uv) + \frac{\partial}{\partial z}(\rho uw) - \rho f v + \frac{\partial p}{\partial x} - \frac{\partial \tau_{xx}}{\partial x} - \frac{\partial \tau_{xy}}{\partial y} - \frac{\partial \tau_{xz}}{\partial z} = 0, \quad (2.41)$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2) + \frac{\partial}{\partial z}(\rho vw) + \rho f u + \frac{\partial p}{\partial y} - \frac{\partial \tau_{xy}}{\partial x} - \frac{\partial \tau_{yy}}{\partial y} - \frac{\partial \tau_{yz}}{\partial z} = 0, \quad (2.42)$$

$$\frac{\partial}{\partial t}(\rho w) + \frac{\partial}{\partial x}(\rho uw) + \frac{\partial}{\partial y}(\rho vw) + \frac{\partial}{\partial z}(\rho w^2) + \frac{\partial p}{\partial z} + \rho g - \frac{\partial \tau_{xz}}{\partial x} - \frac{\partial \tau_{yz}}{\partial y} - \frac{\partial \tau_{zz}}{\partial z} = 0. \quad (2.43)$$

Onde (x, y, z) é o sistema de coordenadas, veja a Figura 2.8, (u, v, w) são as componentes da velocidade, t , p , ρ e g denotam o tempo, pressão, densidade e aceleração gravitacional, respectivamente. E $f = 2\Omega \sin \phi$ é o parâmetro de Coriolis, indicando o efeito de rotação da Terra (Ω é a taxa de angulação da revolução, ϕ é a latitude geográfica).

A tensão viscosa τ_{ij} podem ser expressadas em termos da taxa de deformação do fluido como

$$\frac{\tau_{ij}}{\rho} = \nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (2.44)$$

onde ν é a viscosidade cinemática em (m^2/s) .

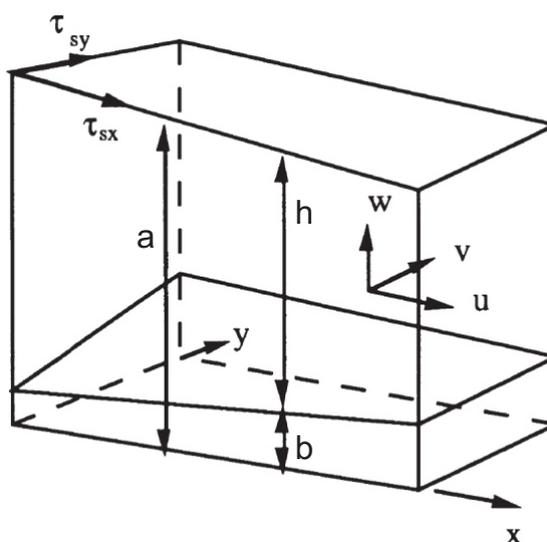


Figura 2.8: Definição do sistema de coordenadas e contornos.
Fonte: Vreugdenhil [58] (Modificada).

A equação de conservação da massa para um fluido é:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0. \quad (2.45)$$

Fazendo algumas considerações em (2.45), veja [58] para detalhes, obtemos a equação de balanço de massa (ou equação da continuidade) para fluidos incompressíveis:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \quad (2.46)$$

Algumas condições são necessárias para realizar esta modelagem, estas são as condições de superfície e fundo, que são de dois tipos. As condições cinemáticas dizem que as partículas de água não cruzarão as fronteiras. Para um fundo sólido, isso significa que a componente da velocidade normal deve desaparecer:

$$u \frac{\partial b}{\partial x} + v \frac{\partial b}{\partial y} - w = 0, \quad z = b, \quad (2.47)$$

onde b é o nível inferior, veja a Figura 2.8. Na superfície livre, é um pouco mais complicado, pois a superfície pode estar se movendo sozinha. Então a velocidade normal relativa deve desaparecer

$$\frac{\partial a}{\partial t} + u \frac{\partial a}{\partial x} + v \frac{\partial a}{\partial y} - w = 0, \quad z = a, \quad (2.48)$$

onde a é o nível da superfície, veja a Figura 2.8.

Na superfície livre, presume-se que as tensões no fluido imediatamente abaixo da

superfície livre sejam as mesmas que as do ar imediatamente abaixo, isto é, a tensão superficial não é levada em consideração e dessa forma temos:

$$p = p_a, \quad (2.49)$$

onde p_a é a pressão atmosférica.

Para simplificar o que queremos dizer com as águas rasas, devemos considerar escalas típicas. Considere três escalas típicas principais H , L e U . A suposição básica na teoria das águas rasas é que qualquer escala vertical H é muito menor que a horizontal L , quão pequena a proporção H/L deve ser não é fácil de dizer. Se x, y possuem escalas típicas L , e as componentes de velocidade horizontal u, v são de ordem U , cada um dos dois primeiros termos em (2.46) é da ordem U/L . No entanto, eles normalmente não cancelam com o outro, o que significa que o terceiro termo também deve ser dessa ordem; caso contrário um equilíbrio entre os três não é possível. Assumindo que a escala de comprimento vertical é a profundidade h da água, então a consequência é que a componente de velocidade vertical é da ordem $w = (Uh)/L$, isto é, é menor que as componentes horizontais pelo mesmo fator que as escalas de comprimento.

Agora, considerando a equação do momento vertical (2.43). Usando as escalas de comprimento e velocidade, lembrando que a viscosidade geralmente é insignificante, podemos estimar todos os termos, exceto o gradiente de pressão. Note que, todos os termos são pequenos em relação à aceleração gravitacional e apenas o gradiente de pressão permanece para balancear, e (2.43) simplifica a distribuição da pressão hidrostática:

$$\frac{\partial p}{\partial z} = -\rho g \quad (2.50)$$

Ao integrar (2.50) usando a condição de contorno (2.49), obtemos:

$$p = g \int_z^a \rho dz + p_a. \quad (2.51)$$

Assumindo que a densidade é constante ao longo da profundidade, temos:

$$p = \rho g(a - z) + p_a. \quad (2.52)$$

A partir de (2.52), os gradientes de pressão em (2.41)-(2.42) podem ser determinados

$$\frac{\partial p}{\partial x} = \rho g \frac{\partial a}{\partial x} + g(a - z) \frac{\partial \rho}{\partial x} + \frac{\partial p_a}{\partial x}. \quad (2.53)$$

Com os resultados obtidos até agora as equações de momento se tornam:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(u^2) + \frac{\partial}{\partial y}(uv) + \frac{\partial}{\partial z}(uw) - fv + g \frac{\partial a}{\partial x} + \frac{g}{\rho_0}(a-z) \frac{\partial \rho}{\partial x} + \frac{1}{\rho_0} \frac{\partial p_a}{\partial x} - \frac{1}{\rho_0} \left(\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \right) = 0, \quad (2.54)$$

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}(uv) + \frac{\partial}{\partial y}(v^2) + \frac{\partial}{\partial z}(vw) - fu + g \frac{\partial a}{\partial y} + \frac{g}{\rho_0}(a-z) \frac{\partial \rho}{\partial y} + \frac{1}{\rho_0} \frac{\partial p_a}{\partial y} - \frac{1}{\rho_0} \left(\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} \right) = 0. \quad (2.55)$$

A etapa final em direção as equações shallow water em 2D envolve a integração das equações de momento (2.54)-(2.55) e a equação da continuidade (2.46) sobre a profundidade $h = a - b$. Integrando a equação da continuidade:

$$\int_b^a \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) dz + [w]_b^a = \frac{\partial}{\partial x}(h\bar{u}) + \frac{\partial}{\partial y}(h\bar{v}) + [w]_b^a - \frac{\partial a}{\partial x} u_s - \frac{\partial a}{\partial y} v_s + \frac{\partial b}{\partial x} u_b + \frac{\partial b}{\partial y} v_b = 0, \quad (2.56)$$

onde \bar{u}, \bar{v} indicam as velocidades médias na profundidade. Note que os termos de superfície (s) e de fundo (b) são termos de correção que podem ser obtidos após a troca das operações de integração e diferenciação presentes no lado esquerdo da equação (2.56). Usando as condições de contorno (2.47) e (2.48), os termos do fundo são cancelados e os da superfície produzem a taxa de mudança de nível da superfície:

$$\frac{\partial a}{\partial t} + \frac{\partial}{\partial x}(h\bar{u}) + \frac{\partial}{\partial y}(h\bar{v}) = 0. \quad (2.57)$$

Uma operação semelhante é aplicada a (2.54) e (2.55), veja [58] para detalhes, e o resultado da integração é:

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2) + \frac{\partial}{\partial y}(huv) - fhv + gh \frac{\partial a}{\partial x} + \frac{gh^2}{2\rho_0} \frac{\partial \rho}{\partial x} - \frac{1}{\rho_0} \tau_{bx} - \frac{\partial}{\partial x}(hT_{xx}) - \frac{\partial}{\partial y}(hT_{xy}) = F_x, \quad (2.58)$$

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}(hv^2) - fhv + gh \frac{\partial a}{\partial y} + \frac{gh^2}{2\rho_0} \frac{\partial \rho}{\partial y} - \frac{1}{\rho_0} \tau_{by} - \frac{\partial}{\partial x}(hT_{xy}) - \frac{\partial}{\partial y}(hT_{yy}) = F_y. \quad (2.59)$$

Onde T_{ij} incluem fricção viscosa, fricção turbulenta e advecção diferencial. As forças motrizes $F_{x,y}$ incluem o estresse do vento, o gradiente de pressão atmosférica e as tensões de radiação por ondas (curtas). Note que a barra indicando as médias de profundidade foram omitidas.

Negligenciando o parâmetro de Coriolis, $\tau_{bx,by}$, T_{ij} e $F_{x,y}$, considerando que a densidade é constante ao longo da direção horizontal e lembrando que $b = b(x, y)$, obtemos a equação shallow water em duas dimensões:

$$\begin{cases} h_t + (hu)_x + (hv)_y = 0 \\ (hu)_t + (hu^2 + \frac{1}{2}gh^2)_x + (huv)_y = -ghb_x \\ (hv)_t + (huv)_x + (hv^2 + \frac{1}{2}gh^2)_y = -ghb_y \end{cases} \quad (2.60)$$

onde h denota a altura da água, $(u, v)^T$ é o vetor velocidade, b representa a topografia do fundo e g é a constante gravitacional.

O jacobiano para as equações SW é dado a seguir

$$\left(\begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + gh & 0 & 2v \end{bmatrix} \right). \quad (2.61)$$

Para as equações SW, a solução de estado estacionário representa uma superfície de água ainda plana, e muitas vezes é referida como solução "lago em repouso":

$$u = v = 0 \quad \text{e} \quad h + b = \text{const.} \quad (2.62)$$

Capítulo 3

Um Estudo do Método de Elementos Finitos Galerkin-Descontínuo

Os métodos dos elementos finitos são procedimentos utilizados para obter soluções aproximadas para equações diferenciais, estes são muito empregados na mecânica de sólidos, estruturas, transferência de calor e fluidos, de fato são úteis em praticamente todos os campos da análise de engenharia [8].

A ideia básica por trás destes métodos consiste em dividir o domínio no qual a equação diferencial é definida, em regiões menores conhecidas como elementos, veja a Figura 3.1, e a solução da equação diferencial é aproximada em cada um desses elementos usando uma função polinomial e o conceito variacional [10, 19].



Figura 3.1: Particionando o domínio em elementos triangulares.
Fonte: DistMesh [1].

Uma classe de métodos dos elementos finitos, o Galerkin Descontínuo (DG), foi introduzido em 1973 por Reed e Hill [48] para resolver a equação de transporte de nêutron, mas apenas recentemente que eles evoluíram e se tornaram adequados para o uso. E

rapidamente encontrou-se sendo empregado em diversas aplicações como em escoamentos turbulentos, simulação de recuperação de petróleo, modelagem de águas rasas, transporte de contaminante em meios porosos, fluxos viscoelásticos, entre muitas outras [22].

O método DG combina características dos métodos dos elementos finitos clássicos e dos volumes finitos, e usa um espaço polinomial por partes completamente descontínuo para obter as soluções numéricas. Várias vantagens tornam este método adequado para as leis de conservação e balanço, em particular para as equações shallow water, dentre elas podemos destacar que o método DG é capaz de preservar o equilíbrio entre o termo fonte e fluxo, sendo assim apto a encontrar a solução estacionária e tratar de forma adequada o termo fonte. Bem como é capaz de preservar a positividade da altura da água e também não introduz oscilações espúrias na solução aproximada.

3.1 Elementos Triangulares

Domínios bidimensionais comumente são divididos em elementos retangulares ou triangulares. Os elementos retangulares são convenientes para geometrias mais regulares, não sendo adequados para limites curvos e geometrias mais complexas. Isso fez com que os elementos triangulares fossem mais utilizados na literatura, visto que estes representam uma geometria arbitrária com muita facilidade e geram soluções com maior acurácia [2]. Por este motivo usaremos os elementos triangulares neste trabalho. A seguir será apresentando alguns conceitos necessários para se utilizar os elementos triangulares.

3.1.1 Coordenadas Naturais

O sistema de coordenadas naturais é muito utilizado no método de elementos finitos, pois este facilita o cálculo das integrais necessárias. Este sistema é definido de tal forma que as coordenadas de um ponto qualquer, no elemento, tome valores entre 0 e 1, veja a Figura 3.2, para mais detalhes veja [2].

As coordenadas naturais utilizadas para os elementos triangulares são chamadas de coordenadas de área, estas são relacionadas com a área de cada triângulo. Para definir as coordenadas de área do ponto $p(x, y)$, veja a Figura 3.3, em termos do sistema de coordenadas naturais é necessário calcular as novas coordenadas L_1 , L_2 e L_3 , quais são definidas em termos da subárea triangular isto é,

$$L_1 = \frac{A_1}{A}, \quad L_2 = \frac{A_2}{A}, \quad L_3 = \frac{A_3}{A}, \quad (3.1)$$

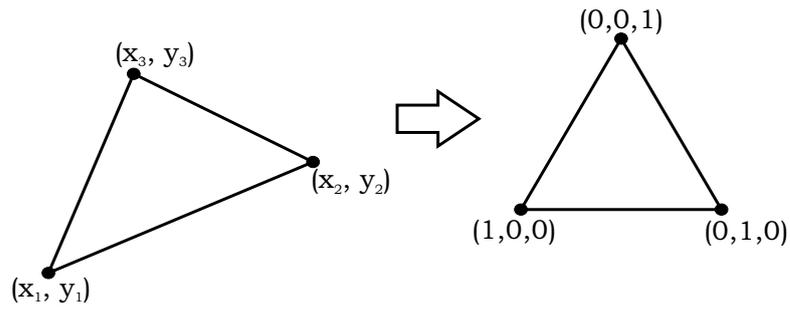


Figura 3.2: Exemplo de mudança do sistema cartesiano para o sistema de coordenadas naturais.

Fonte: Imagem do autor.

onde $A = A_1 + A_2 + A_3$, logo $L_1 + L_2 + L_3 = 1$.

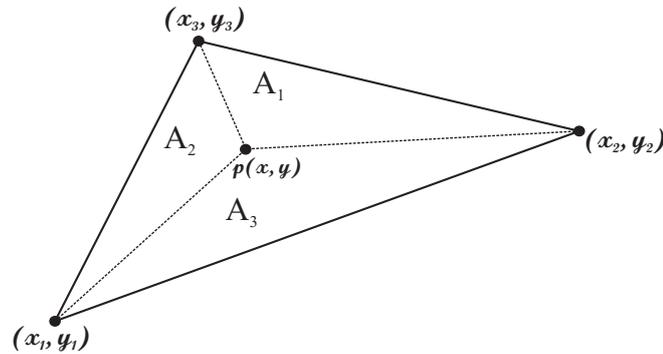


Figura 3.3: Transformação das coordenadas cartesianas para coordenadas de área do ponto $p(x, y)$.

Fonte: Imagem do autor.

As áreas dos triângulos podem ser calculadas usando as coordenadas cartesianas do ponto e dos vértices do triângulo:

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}, \quad A_1 = \frac{1}{2} \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix},$$

$$A_2 = \frac{1}{2} \begin{vmatrix} 1 & x & y \\ 1 & x_3 & y_3 \\ 1 & x_1 & y_1 \end{vmatrix}, \quad A_3 = \frac{1}{2} \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix}.$$

A relação entre os dois sistemas de coordenadas pode ser definida como

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix}. \quad (3.2)$$

A inversa da relação entre as coordenadas dada pela equação (3.2) pode ser expressada como

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2y_3 - x_3y_2 & y_2 - y_3 & x_3 - x_2 \\ x_3y_1 - x_1y_3 & y_3 - y_1 & x_1 - x_3 \\ x_1y_2 - x_2y_1 & y_1 - y_2 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}. \quad (3.3)$$

3.1.2 Funções de Forma

Uma função $\phi(x, y)$ pode ser aproximada pelo polinômio $\phi_h(x, y)$ utilizando funções de forma (polinômios) e os valores nodais.

$$\phi_h(x, y) = \sum_{i=1}^n \phi_i N_i(x, y), \quad (3.4)$$

onde n é a quantidade de nós de um triângulo e $\phi_i = \phi_h(x_i, y_i)$. Aqui, cada função de forma N_i (polinômio de um certo grau) deve ser tal que seu valor seja um se avaliado em seu nó relacionado e zero em qualquer um dos outros $n - 1$ nós, isto é

$$N_i(x_j, y_j) = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j, \end{cases} \quad 1 \leq i, j \leq n. \quad (3.5)$$

A função de forma para os elementos triangulares podem ser lineares, quadráticas ou cúbicas e para cada caso é necessário um determinado número de nós, veja a Figura 3.4. Estas podem ser determinadas através das coordenadas cartesianas ou de área, porém a fim de facilitar cálculos define-se as funções de forma por meio das coordenadas de área.

Portanto, utilizando funções de forma lineares, obtemos a seguinte interpolação

$$\phi_h(x, y) = \sum_{i=1}^3 \phi_i N_i(x, y), \quad (3.6)$$

onde $N_1(x, y) = L_1 + L_2 - L_3$, $N_2(x, y) = -L_1 + L_2 + L_3$ e $N_3(x, y) = L_1 - L_2 + L_3$.

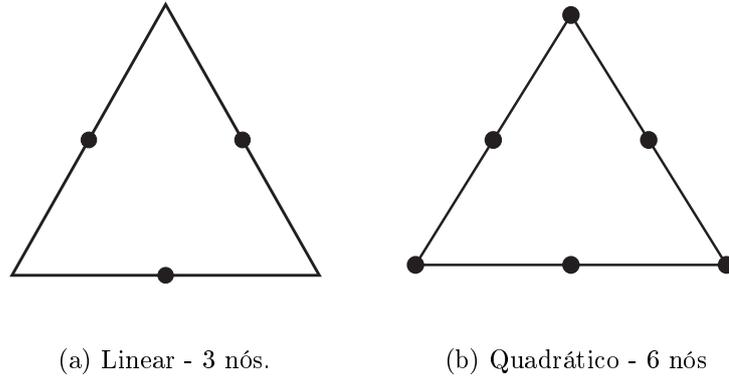


Figura 3.4: Quantidade de nós necessária para polinômios lineares e quadráticos.
Fonte: Imagem do autor.

E para funções de forma quadráticas,

$$\phi_h(x, y) = \sum_{i=1}^6 \phi_i N_i(x, y), \quad (3.7)$$

onde

$$\begin{aligned} N_1 &= L_1(2L_1 - 1), \\ N_2 &= L_2(2L_2 - 1), \\ N_3 &= L_3(2L_3 - 1), \\ N_4 &= 4L_1L_2, \\ N_5 &= 4L_2L_3, \\ N_6 &= 4L_3L_1. \end{aligned} \quad (3.8)$$

No que se segue será necessário avaliar as derivadas das funções de forma, definidas em coordenadas de área, em relação às coordenadas cartesianas. Essas derivadas podem ser obtidas utilizando a regra da cadeia. Assim, para funções lineares obtemos o seguinte operador

$$\begin{aligned} \frac{\partial}{\partial x} &= \frac{\partial}{\partial L_1} \frac{\partial L_1}{\partial x} + \frac{\partial}{\partial L_2} \frac{\partial L_2}{\partial x} + \frac{\partial}{\partial L_3} \frac{\partial L_3}{\partial x}, \\ &= \frac{c_1}{2A} \frac{\partial}{\partial L_1} + \frac{c_2}{2A} \frac{\partial}{\partial L_2} + \frac{c_3}{2A} \frac{\partial}{\partial L_3}, \\ &= \sum_{i=1}^3 \frac{c_i}{2A} \frac{\partial}{\partial L_i}. \end{aligned} \quad (3.9)$$

Onde $c_1 = y_2 - y_3$, $c_2 = y_3 - y_1$ e $c_3 = y_1 - y_2$. Analogamente, obtemos

$$\frac{\partial}{\partial y} = \sum_{i=1}^3 \frac{d_i}{2A} \frac{\partial}{\partial L_i}, \quad (3.10)$$

onde $d_1 = x_3 - x_2$, $d_2 = x_1 - x_3$ e $d_3 = x_2 - x_1$.

A seguir será explicado brevemente o método RKDG para as Leis de Conservação, seguido de uma explicação mais detalhada para Leis de Balanço, utilizando as equações shallow water para descrever o método. Utilizamos as seguintes referências como base para apresentar o método [65, 27, 66, 68].

3.2 Método RKDG para Leis de Conservação

Considere a seguinte lei de conservação bidimensional

$$U_t + \nabla \cdot \mathbf{F}(U) = 0 \quad (x, y) \in \Omega \in \mathbb{R}^2, \quad t \geq 0, \quad (3.11)$$

onde $U = (u_1, u_2, \dots, u_m)^T$ e $\mathbf{F}(U) = (f(U), g(U))$, com condição inicial $U(x, y, 0) = U_0(x, y)$ e talvez alguma condição de contorno, por exemplo de tipo Dirichlet $U|_{\partial\Omega} = \gamma$.

Seja T_τ a família de partições do domínio computacional Ω parametrizado por $\tau > 0$. Para qualquer triângulo $K \in T_\tau$, definimos $\tau_K = \text{diam}(K)$, $\tau = \max_{K \in T_\tau} \tau_K$ e $|K|$ a sua área. Para cada lado e_K^i ($i = 1, 2, 3$) do triângulo K , denotamos seu comprimento l_K^i e seu vetor normal unitário por \vec{n}_K^i . Buscamos uma solução aproximada de (3.11) que pertença ao espaço dimensional finito

$$V_\tau = V_\tau^k = \{w \in L^2(\Omega); w|_K \in P^k(K) \forall K \in T_\tau\}, \quad (3.12)$$

onde $P^k(K)$ denota o espaço de polinômios em K de grau no máximo k . Seguindo [27, 65], para a derivação de um esquema DG, a equação (3.11) é multiplicada por uma função teste $w(x, y) \in V_\tau$ e integramos sobre K :

$$\iint_K U_t w \, d\mathbf{x} + \iint_K (\nabla \cdot \mathbf{F}) w \, d\mathbf{x} = 0, \quad (3.13)$$

onde $\mathbf{x} = (x, y)$. Aplicando o teorema da Divergência na equação acima, obtemos:

$$\iint_K U_t w \, d\mathbf{x} - \iint_K \mathbf{F} \cdot \nabla w \, d\mathbf{x} + \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}|_{e_K^i} \cdot \vec{n}_K^i w \, ds = 0. \quad (3.14)$$

O fluxo numérico $\hat{\mathbf{F}}$ é definido ao longo das arestas do elemento K de forma que

$$\hat{\mathbf{F}}|_{e_K^i} \cdot \vec{n}_K^i = \mathcal{F} \left(U_i^{\text{int}(K)}, U_i^{\text{ext}(K)}, \vec{n}_K^i \right), \quad (3.15)$$

onde $U_i^{int(K)}$ é o valor obtido no interior do elemento K como

$$U_i^{int(K)} \equiv U(\mathbf{x}^{int(K)}, t) = \lim_{\mathbf{z} \rightarrow \mathbf{x}, \mathbf{z} \in K} U(\mathbf{z}, t) \quad \text{para } t \text{ fixo, } \mathbf{x} \in e_K^i \quad i = 1, 2, 3, \quad (3.16)$$

e $U_i^{ext(K)}$ é o valor obtido no exterior do elemento K como

$$U(\mathbf{x}^{ext(K)}, t) = \begin{cases} \gamma(\mathbf{x}, t), & \text{se } \mathbf{x} \in \partial\Omega, \\ \lim_{\mathbf{z} \rightarrow \mathbf{x}, \mathbf{z} \notin K} U(\mathbf{z}, t) & \text{em caso contrário,} \end{cases} \quad \text{para } t \text{ fixo,} \quad (3.17)$$

no caso de condição de contorno de Dirichlet ($U|_{\partial\Omega} = \gamma$), para detalhes veja [27]. A função \mathcal{F} pode ser definida usando o fluxo global Lax-Friedrichs

$$\begin{aligned} \mathcal{F}(a_1, a_2, \vec{n}) &= \frac{1}{2} [\mathbf{F}(a_1) \cdot \vec{n} + \mathbf{F}(a_2) \cdot \vec{n} - \alpha(a_2 - a_1)], \\ \alpha &= \max |\mathbf{F}'(U) \cdot \vec{n}|, \end{aligned} \quad (3.18)$$

onde o máximo é tomado sobre o domínio Ω . E este fluxo satisfaz a conservatividade e consistência

$$\mathcal{F}(a_1, a_2, \vec{n}) = -\mathcal{F}(a_2, a_1, -\vec{n}), \quad \mathcal{F}(a_1, a_1, \vec{n}) = \mathbf{F}(a_1) \cdot \vec{n}. \quad (3.19)$$

Podemos reescrever a equação (3.14) na forma de um sistema de EDO's (d/dt) $U = \mathcal{L}(U)$, para isso definimos (usando as funções de forma N_i):

$$U(x, y, t) = \sum_{l=1}^n U_l(t) N_l(x, y) \quad \text{e} \quad w = \sum_{j=1}^n c_j N_j(x, y) \quad (3.20)$$

Substituindo (3.20) em (3.14) obtemos:

$$\begin{aligned} & \iint_K \frac{d}{dt} \left[\sum_{l=1}^n U_l(t) N_l(x, y) \right] \sum_{j=1}^n c_j N_j(x, y) \, d\mathbf{x} - \iint_K \mathbf{F} \cdot \nabla \left(\sum_{j=1}^n c_j N_j(x, y) \right) \, d\mathbf{x} \\ & + \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}|_{e_K^i} \cdot \vec{n}_K^i \sum_{j=1}^n c_j N_j(x, y) \, ds = 0 \end{aligned} \quad (3.21)$$

ou equivalente,

$$\sum_{j=1}^n c_j \left(\sum_{l=1}^n \left\{ \iint_K \frac{dU_l}{dt} N_l N_j \, d\mathbf{x} - \iint_K \mathbf{F} \cdot \nabla N_j \, d\mathbf{x} + \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}|_{e_K^i} \cdot \vec{n}_K^i N_j \, ds \right\} \right) = 0. \quad (3.22)$$

Considerando uma escolha natural para o grupo de parâmetros c_j : $c_j = 1$, $c_p = 0$

para $p \neq j$ e após alguns cálculos, para detalhes veja [10], podemos escrever (3.22) em uma forma mais compacta

$$\frac{dU(t)}{dt} = M^{-1} \left(\iint_K \mathbf{F} \cdot \nabla N \, d\mathbf{x} - \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}|_{e_K^i} \cdot \vec{n}_K^i N \, ds \right). \quad (3.23)$$

Onde M é a matriz de massa (local do elemento K) e seus coeficientes são definidos como $M_{lj} = 2|K| \int_0^1 \int_0^{1-L_1} N_l N_j \, dL_1 \, dL_2$ e esta é

$$M = |K| \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}, \quad (3.24)$$

para polinômios lineares ($k = 1$). Caso queiramos usar os polinômios quadráticos ($k = 2$) obtemos:

$$M = |K| \begin{bmatrix} \frac{1}{30} & -\frac{1}{180} & -\frac{1}{180} & 0 & -\frac{1}{45} & 0 \\ -\frac{1}{180} & \frac{1}{30} & -\frac{1}{180} & 0 & 0 & -\frac{1}{45} \\ -\frac{1}{180} & -\frac{1}{180} & \frac{1}{30} & -\frac{1}{45} & 0 & 0 \\ 0 & 0 & -\frac{1}{45} & \frac{8}{45} & \frac{4}{45} & \frac{4}{45} \\ -\frac{1}{45} & 0 & 0 & \frac{4}{45} & \frac{8}{45} & \frac{4}{45} \\ 0 & -\frac{1}{45} & 0 & \frac{4}{45} & \frac{4}{45} & \frac{8}{45} \end{bmatrix}. \quad (3.25)$$

Agora podemos usar método de discretização no tempo Runge-Kutta de terceira ordem para resolver a EDO (3.23)

$$\begin{aligned} U^{(1)} &= U^n + \Delta t \mathcal{L}(U^n), \\ U^{(2)} &= \frac{3}{4} U^n + \frac{1}{4} (U^{(1)} + \Delta t \mathcal{L}(U^{(1)})), \\ U^{n+1} &= \frac{1}{3} U^n + \frac{2}{3} (U^{(2)} + \Delta t \mathcal{L}(U^{(2)})). \end{aligned} \quad (3.26)$$

Onde $\mathcal{L}(U)$ é o operador espacial, definido como o lado direito da equação (3.23).

Dessa forma, temos que o método RKDG utiliza uma formulação local, ou seja, os cálculos realizados devem ser feitos elemento por elemento. Assim, calcula-se as integrais em (3.23) em cada elemento, e resolve-se um sistema de EDO's com m equações (as m componentes de U) via Runge-Kutta, utilizando para isso a matriz de massa 3×3 (caso linear) em (3.24) ou uma matriz de massa 6×6 (caso quadrático) em (3.25). Esta é uma diferença importante entre este método e os métodos de elementos finitos clássicos, no qual se utiliza uma formulação global e assim uma matriz de massa global. Esta formulação

local faz com que para problemas estritamente hiperbólicos seja possível capturar as ondas geralmente sem a necessidade de considerar condições de contorno.

Teorema 3.1 *O método RKDG com o polinômio $U_K(\mathbf{x})$ de grau k deve satisfazer o princípio do máximo*

$$A \leq \bar{U}_K^{n+1} \leq B, \quad (3.27)$$

onde $A = \min_{\mathbf{x}} U_0$, $B = \max_{\mathbf{x}} U_0$ e \bar{U}_K é a média de U no elemento K , o super índice $n+1$ indica o próximo espaço no tempo. Uma condição suficiente é que cada $U_K(\mathbf{x})$ satisfaça $U_K(\mathbf{x}) \in [A, B]$, $\forall \mathbf{x} \in S_K$, onde S_K é o conjunto dos pontos de quadratura, sob a condição CFL

$$\alpha \frac{\Delta t}{|K|} \sum_{i=1}^3 l_K^i \leq \frac{2}{3} \hat{w}_1 \quad (3.28)$$

onde \hat{w}_1 é o peso do primeiro ponto de quadratura Gauss-Lobatto no intervalo $[-1/2, 1/2]$, para mais detalhes veja [68].

O método RKDG, quando aplicado em problemas com soluções descontínuas, pode gerar oscilações e até instabilidade não linear. De modo que torne-se necessário a utilização do limitador TVB para controlar tais soluções [65, 27, 66]. Em alguns casos, há também a necessidade de que a variável conservada ou balanceada esteja contida dentro de um certo intervalo, veja o Teorema 3.1, sendo preciso também a aplicação do limitador “positive preserving”.

3.2.1 Limitador TVB

Neste trabalho é utilizado o limitador TVB clássico com função minmod corrigida definida em [27] por:

$$\bar{m}(a_1, \dots, a_n) = \begin{cases} a_1, & \text{se } |a_1| \leq \Lambda \tau^2, \\ m(a_1, \dots, a_n), & \text{em caso contrário,} \end{cases} \quad (3.29)$$

onde Λ é o parâmetro TVB a ser escolhido adequadamente em termos da segunda derivada da função, veja [26], se Λ é escolhido pequeno demais, mais células do que o necessário serão identificadas contendo oscilações, aumentando o custo computacional; no entanto, se Λ é escolhido muito grande, oscilações espúrias surgirão. A função minmod m é dada por

$$m(a_1, \dots, a_n) = \begin{cases} s \min_i |a_i|, & \text{se } s = \text{sign}(a_1) = \dots = \text{sign}(a_n), \\ 0, & \text{em caso contrário.} \end{cases} \quad (3.30)$$

De maneira geral, este limitador é aplicado em U após cada passo interno da discretização de tempo Runge-Kutta. O procedimento do limitador TVB envolve duas etapas: a primeira é verificar se alguma limitação é necessária, em um determinado elemento; se um certo elemento necessitar do limitador, então o segundo passo é aplicar este sobre o respectivo vetor [66]. Para detectar as células com problemas usando o limitador descrito acima, declaramos que se função descrita em (3.29) retornar um valor diferente do primeiro argumento, esta célula é marcada como uma célula problemática e deve-se aplicar o limitador.

Para construir o limitador TVB para os elementos triangulares nós procedemos como em [27]. Considere o triângulo K_0 , e seus três vizinhos K_1, K_2 e K_3 como na Figura 3.5, onde m_i é o ponto médio da aresta $e_{K_0}^i$ do elemento K_0 e b_j denota o baricentro de cada triângulo K_j para $j = 0, 1, 2, 3$, e seja U uma função linear.

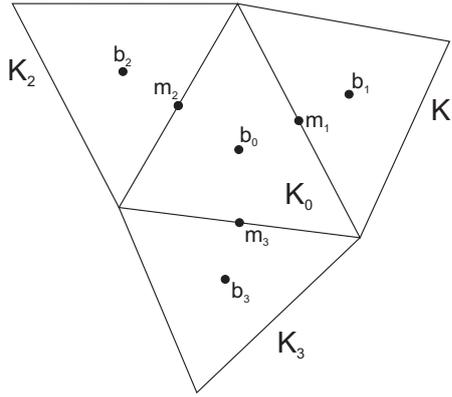


Figura 3.5: Ilustração dos dados necessário para o limitador TVB.
Fonte: Cockburn et al [27] (Modificada).

Para os elementos triangulares limitamos a grandeza Δ_i definida abaixo:

$$\Delta_i = \overline{m} \left(\hat{U}(m_i, K_0), \nu \Delta \bar{U}(m_i, K_0) \right), \quad i = 1, 2, 3. \quad (3.31)$$

Onde \overline{m} é o limitador TVB (3.29), e $\nu > 1$. Nós tomamos $\nu = 1.5$, como em [27]. As variáveis $\hat{U}(m_i, K_0)$ e $\Delta \bar{U}(m_i, K_0)$ são definidas a seguir.

$$\hat{U}(m_i, K_0) = U(m_i) - \bar{U}_{K_0}, \quad i = 1, 2, 3, \quad (3.32)$$

ou seja, $\hat{U}(m_i, K_0)$ é a diferença entre os valores de U no ponto médio da aresta $e_{K_0}^i$ e o valor médio no elemento K_0 , que coincide com o valor de U no baricentro b_0 . Por outro

lado, define-se

$$\begin{aligned}\Delta\bar{U}(m_1, K_0) &= \alpha_1(\bar{U}_{K_1} - \bar{U}_{K_0}) + \beta_1(\bar{U}_{K_2} - \bar{U}_{K_0}), \\ \Delta\bar{U}(m_2, K_0) &= \alpha_2(\bar{U}_{K_2} - \bar{U}_{K_0}) + \beta_2(\bar{U}_{K_3} - \bar{U}_{K_0}), \\ \Delta\bar{U}(m_3, K_0) &= \alpha_3(\bar{U}_{K_3} - \bar{U}_{K_0}) + \beta_3(\bar{U}_{K_1} - \bar{U}_{K_0}).\end{aligned}\tag{3.33}$$

Onde \bar{U}_K é o valor médio de U sobre o elemento K , os coeficientes α_i e β_i , $i = 1, 2, 3$ são tais que:

$$\begin{aligned}m_1 - b_0 &= \alpha_1(b_1 - b_0) + \beta_1(b_2 - b_0), \\ m_2 - b_0 &= \alpha_2(b_2 - b_0) + \beta_2(b_3 - b_0), \\ m_3 - b_0 &= \alpha_3(b_3 - b_0) + \beta_3(b_1 - b_0).\end{aligned}\tag{3.34}$$

Estes coeficiente α_i e β_i existem, pois os vetores $b_i - b_0$ e $b_j - b_0$ ($i \neq j$) são linearmente independentes em \mathbb{R}^2 e formam uma base.

Uma vez limitados os Δ_i em (3.31), se $\sum_{i=1}^3 \Delta_i = 0$ define-se

$$U(x, y, t) = \sum_{i=1}^3 (\Delta_i + \bar{U}_{K_0}) N_i(x, y).\tag{3.35}$$

Note que caso a limitação não fosse necessária isto devolve a solução original, pois $\Delta_i = \hat{U}(m_i, K_0)$ e $\Delta_i + \bar{U}_{K_0} = U(m_i)$.

Se $\sum_{i=1}^3 \Delta_i \neq 0$, calculamos

$$pos = \sum_{i=1}^3 \max(0, \Delta_i), \quad neg = \sum_{i=1}^3 \max(0, -\Delta_i),$$

e

$$\theta^+ = \min\left(1, \frac{neg}{pos}\right), \quad \theta^- = \min\left(\frac{pos}{neg}\right).$$

Então, definimos

$$U(x, y, t) = \sum_{i=1}^3 (\hat{\Delta}_i + \bar{U}_{K_0}) N_i(x, y)\tag{3.36}$$

onde

$$\hat{\Delta}_i = \theta^+ \max(0, \Delta_i) - \theta^- \max(0, -\Delta_i).$$

Para sistemas, realizamos a limitação nas variáveis características locais [27]. Neste caso para limitar o a célula procedemos da seguinte forma:

- Encontramos a matriz R e sua inversa R^{-1} qual diagonaliza o Jacobiano,

$$R^{-1} \left(\frac{\partial}{\partial U} \mathbf{F}(\bar{U}_{K_0}) \cdot \frac{m_i - b_0}{|m_i - b_0|} \right) R = D, \quad (3.37)$$

onde D é uma matriz diagonal que contém os autovalores do Jacobiano.

- Transformamos todas as quantidades necessárias para limitação multiplicando-as a esquerda por R^{-1} .
- Aplicamos o limitador escalar (3.29) em cada uma das componentes das quantidades transformadas.
- O resultado é transformado ao espaço original multiplicando-o a esquerda por R .

Para mais detalhes veja [66, 27].

3.2.1.1 Tratamento do Limitador no Contorno

Os trabalhos [66, 27] não informam como proceder caso uma ou mais arestas do elemento K_0 esteja na fronteira. A solução encontrada por nós foi definir um “baricentro fictício” simétrico ao baricentro b_0 em relação ao ponto médio da aresta presente na fronteira. Por exemplo, suponha que a aresta 1 do elemento K_0 pertença a fronteira, então pegamos um elemento vizinho fictício à aresta 1, onde o seu baricentro b_1^* é simétrico ao b_0 em relação à m_1 , veja a Figura 3.6.

Da construção acima, obtemos

$$2(m_1 - b_0) = (b_1)^* - b_0,$$

portanto $(b_1)^* = 2m_1 - b_0$. De forma geral, se uma aresta j estiver na fronteira temos que:

$$(b_j)^* = 2m_j - b_0. \quad (3.38)$$

Observe que se substituirmos $b_j = (b_j)^*$ em (3.34), obtemos que $\alpha_j = 1/2$ e $\beta_j = 0$.

3.2.2 Limitador Positive Preserving para Leis de Conservação

Para que a condição dada pelo Teorema 3.1 seja satisfeita, nós precisamos modificar o polinômio $U_K(\mathbf{x})$ de forma que $U_K(\mathbf{x}) \in [A, B]$, $\forall \mathbf{x} \in S_K$. Para todos os K , assumimos

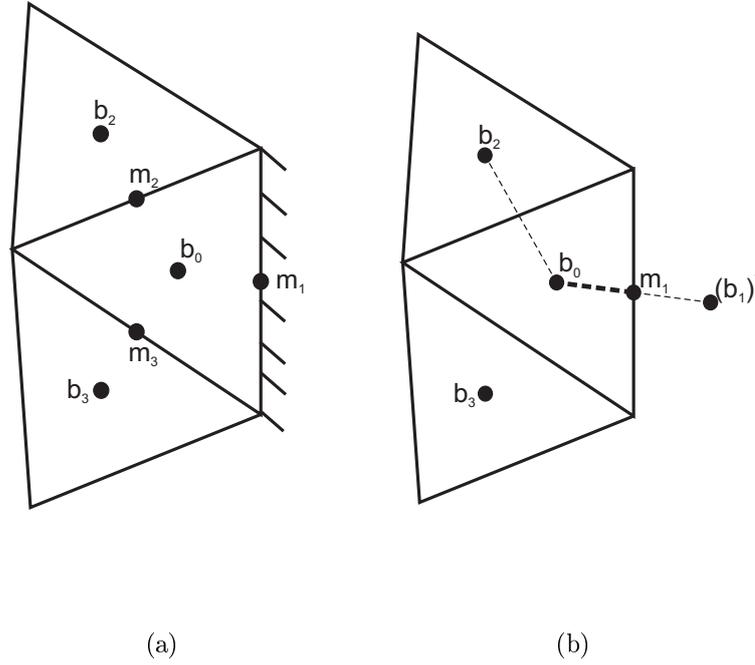


Figura 3.6: Ilustração do caso em que a aresta 1 do elemento K_0 esteja na fronteira. Note que $\alpha_1 = 1/2$ e $\beta_1 = 0$.

Fonte: Imagem do autor.

que $\bar{U}_K^n \in [A, B]$, como em [68] usamos o seguinte polinômio $\tilde{U}_K(\mathbf{x})$ em vez de $U_K(\mathbf{x})$:

$$\tilde{U}_K(\mathbf{x}) = \theta(U_K(\mathbf{x}) - \bar{U}_K^n) + \bar{U}_K^n, \quad (3.39)$$

onde

$$\theta = \min \left\{ \left| \frac{B - \bar{U}_K^n}{M_K - \bar{U}_K^n} \right|, \left| \frac{A - \bar{U}_K^n}{m_K - \bar{U}_K^n} \right|, 1 \right\},$$

com

$$M_K = \max_{\mathbf{x} \in S_K} U_K(\mathbf{x}), \quad m_K = \min_{\mathbf{x} \in S_K} U_K(\mathbf{x}).$$

3.2.3 O algoritmo para discretização temporal RK

Podemos esquematizar a discretização Runge-Kutta (3.26) para as leis de conservação da seguinte forma:

1. Dado o polinômio DG de $U_K(\mathbf{x})$ no passo de tempo n , então calcula-se $\alpha = \max |\mathbf{F}'(U) \cdot \vec{n}|$, onde o máximo é tomado sobre o conjunto de pontos de quadratura nas arestas, \tilde{S}_K , da malha. E assim determina-se o intervalo de tempo (ver condição

CFL em (3.28)):

$$\Delta t = \min \frac{2}{3} \frac{\hat{w}_1 |K|}{\alpha \sum_{i=1}^3 l_k^i},$$

onde α para as leis de conservação é dado pela equação (3.18).

2. Calcula-se o primeiro estágio com $U_K(\mathbf{x})$ e essa solução é denotada por U_K^1 . Modifica-se essa solução aplicando primeiro o limitador TVB e, se necessário, posteriormente o limitador positive preserving e assim atualiza-se o valor de $U_K^1 = \tilde{U}_K^1(\mathbf{x})$.
3. Calcula-se o segundo estágio com $U_K^1(\mathbf{x})$, e obtém-se $U_K^2(\mathbf{x})$, aplica-se o limitador TVB e, se necessário o positive preserving e obtém-se $U_K^2 = \tilde{U}_K^2(\mathbf{x})$.
4. Calcula-se o terceiro estágio com $U_K^2(\mathbf{x})$, e denota-se essa solução por $U_K^3(\mathbf{x})$, aplica-se o limitador TVB e o positive preserving, se necessário, e obtém-se a solução no tempo $n + 1$.

3.3 Método RKDG para as Equações Shallow Water

Podemos escrever as equações shallow water (2.60) na seguinte forma:

$$U_t + \nabla \cdot \mathbf{F}(U) = s(h, b) \quad (3.40)$$

onde $U = (h, hu, hv)^T$ e $\mathbf{F}(U) = (f(U), g(U))$. Procedendo de forma análoga que as leis de conservação obtemos [65, 27]:

$$\iint_K U_t w \, d\mathbf{x} - \iint_K \mathbf{F} \cdot \nabla w \, d\mathbf{x} + \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}|_{e_K^i} \cdot \vec{n}_K^i w \, ds = \iint_K s(h, b) w \, d\mathbf{x}. \quad (3.41)$$

onde $\hat{\mathbf{F}}$ é dado pelas equações (3.15)–(3.18), para as equações Shallow Water temos que:

$$\alpha = \max \left(\left(|u| + \sqrt{gh}, |v| + \sqrt{gh} \right) \cdot \vec{n} \right) \quad (3.42)$$

Afim de obter um método bem balanceado, isto é manter o equilíbrio entre o fluxo e o termo fonte, preservando a solução estacionária (2.62), o esquema (3.41) é modificado e assume a forma:

$$\iint_K U_t w \, d\mathbf{x} - \iint_K \mathbf{F} \cdot \nabla w \, d\mathbf{x} + \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}^*|_{e_K^i} \cdot \vec{n}_K^i w \, ds = \iint_K s(h, b) w \, d\mathbf{x}. \quad (3.43)$$

O fluxo $\hat{\mathbf{F}}^*$ é calculado com base na técnica de reconstrução hidrostática [5]. De posse dos valores $U_i^{int(K)}$ e $U_i^{ext(K)}$ na aresta e_K^i , define-se

$$\begin{aligned} h_i^{*,int(K)} &= \max\left(0, h_i^{int(K)} + b_i^{int(K)} - \max\left(b_i^{int(K)}, b_i^{ext(K)}\right)\right) \\ h_i^{*,ext(K)} &= \max\left(0, h_i^{ext(K)} + b_i^{ext(K)} - \max\left(b_i^{int(K)}, b_i^{ext(K)}\right)\right) \end{aligned} \quad (3.44)$$

e redefinimos os valores interiores e exteriores de U como:

$$\begin{aligned} U_i^{*,int(K)} &= \frac{h_i^{*,int(K)}}{h_i^{int(K)}} U_i^{int(K)}, \\ U_i^{*,ext(K)} &= \frac{h_i^{*,ext(K)}}{h_i^{ext(K)}} U_i^{ext(K)}. \end{aligned} \quad (3.45)$$

Logo, o fluxo $\hat{\mathbf{F}}^*$ é tal que

$$\hat{\mathbf{F}}^*|_{e_K^i} \cdot \vec{n}_K^i = \mathcal{F}\left(U_i^{*,int(K)}, U_i^{*,ext(K),n_K^i}\right) + \langle \delta_{i,x}^*, \delta_{i,y}^* \rangle \cdot \vec{n}_K^i, \quad (3.46)$$

onde

$$\delta_{i,x}^* = \left(0, \frac{g}{2}(h_i^{int(K)})^2 - \frac{g}{2}(h_i^{*,int(K)})^2, 0\right)^T, \quad \delta_{i,y}^* = \left(0, 0, \frac{g}{2}(h_i^{int(K)})^2 - \frac{g}{2}(h_i^{*,int(K)})^2\right)^T.$$

O símbolo $\langle \cdot, \cdot \rangle$ em (3.46) é utilizado eventualmente para definir uma operação entre uma matriz 3×2 , cujas colunas são os vetores $\delta_{i,x}^*$ e $\delta_{i,y}^*$, e o vetor normal \vec{n}_K^i , de forma que é realizado um produto interno entre cada linha dessa matriz e o vetor \vec{n}_K^i .

Não é difícil comprovar que o método (3.43), combinado com a escolha de fluxos (3.46), é bem balanceado, mostrando que no estado estacionário (2.62) das equações de shallow water o fluxo numérico $\hat{\mathbf{F}}^*$ se torna $\mathbf{F}(U_i^{int(K)})$ na aresta e_K^i , veja [65].

Seguindo o passo a passo do método RKDG para leis de conservação, escrevemos então a equação (3.43) em um sistema de equações diferenciais ordinárias

$$\frac{dU(t)}{dt} = \mathcal{L}(U(t)), \quad (3.47)$$

onde o operador espacial neste caso é:

$$\mathcal{L}(U(t)) = M^{-1} \left(\iint_K \mathbf{F} \cdot \nabla N \, d\mathbf{x} - \sum_{i=1}^3 \int_{e_K^i} \hat{\mathbf{F}}^*|_{e_K^i} \cdot \vec{n}_K^i N \, ds + \iint_K s(h, b) N \, d\mathbf{x} \right) \quad (3.48)$$

Podemos então, como explicado na seção 3.2, usar o método de discretização no tempo Runge-Kutta de terceira ordem (3.26).

No caso das equações SW, o princípio do máximo dado pelo Teorema 3.1 se resume em $h_K(\mathbf{x}) \geq 0$, $\forall \mathbf{x} \in S_K$. Podemos então usar o limitador positive preserving [65] dado pela equação (3.39) com

$$\theta = \min \left\{ 1, \frac{\bar{h}_j^n}{\bar{h}_j^n - m_{i,j}} \right\}, \quad (3.49)$$

onde

$$m_{i,j} = \min_{(x,y) \in S_K} h_{ij}^n(x,y). \quad (3.50)$$

Assim, calculamos o polinômio modificado \tilde{U}_j^n e o usamos em vez de U_j^n no esquema RKDG. Note que, este limitador não terá efeito em soluções estacionárias (2.62), portanto este não afeta a propriedade bem balanceada do método [65]. Observe também que na região molhada, onde $\theta = 1$, o limitador não tem efeito, ou seja, $\tilde{U}_j^n = U_j^n$. Assim sendo este limitador é ativado apenas na região seca ou quase seca, onde $\theta < 1$, veja [66].

O limitador TVB se não utilizado corretamente pode destruir a solução estacionária. Para evitar isso, como explicado em [66, 65], verifica-se então se este é necessário baseado em $(h + b, (hu), (hv))^T$ se $m_{i,j} \geq 0$ (região molhada) ou baseado em $(h, (hu), (hv))^T$ caso contrário; se necessitar a sua aplicação procedemos conforme descrito na seção 3.2.1.

3.3.1 O algoritmo para discretização temporal RK

A fim de resolver numericamente as equações SW precisamos que o método numérico preserve a positividade de h , para obter esta condição na discretização de tempo RK um recurso eficiente é realizar um cálculo preliminar para verificar se o próximo intervalo de tempo $n + 1$ irá produzir uma altura negativa. Se sim, o cálculo será reiniciado a partir do passo de tempo n com metade do passo de tempo. Dessa forma podemos esquematizar a discretização Runge-Kutta, veja a Figura 3.8, da seguinte forma:

1. Dado o polinômio DG de $U_K(\mathbf{x})$ no passo de tempo n satisfazendo $\bar{h} \geq 0$ e $h_K(\mathbf{x}) \geq 0$, $\forall \mathbf{x} \in \tilde{S}_K$, onde \tilde{S}_K é o conjunto dos pontos de quadratura presentes nas arestas (da malha toda). Então calcula-se $\alpha = \max((|u| + \sqrt{gh}, |v| + \sqrt{gh}) \cdot \vec{n})$, onde o máximo é tomado sobre \tilde{S}_K na malha. E assim determina-se o intervalo de tempo (veja a condição CFL em (3.28)):

$$\Delta t = \min \frac{2}{3} \frac{\hat{w}_1 |K|}{\alpha \sum_{i=1}^3 l_k^i},$$

onde para as equações shallow water α é dado pela equação (3.42).

2. Calcula-se o primeiro estágio com $U_K(\mathbf{x})$ e essa solução é denotada por U_K^1 . Modica-se essa solução aplicando primeiro o limitador TVB e posteriormente o limitador para preservar a positividade, e assim obtém-se $\tilde{U}_K^1(\mathbf{x})$.
3. Calcula-se o segundo estágio com $\tilde{U}_K^1(\mathbf{x})$, e obtém-se $U_K^2(\mathbf{x})$. Se o valor médio da altura da água é negativo, então volta-se ao passo 2 e reinicia-se dividindo Δt pela metade; caso contrário, aplica-se os limitadores e obtém-se $\tilde{U}_K^2(\mathbf{x})$.
4. Calcula-se o terceiro estágio com $\tilde{U}_K^2(\mathbf{x})$, e denota-se essa solução por $U_K^3(\mathbf{x})$. Se o valor médio da altura da água é negativo, então volta-se no passo 2 e reinicia-se dividindo Δt pela metade; caso contrário, aplica-se os limitadores e obtém-se a solução no tempo $n + 1$.

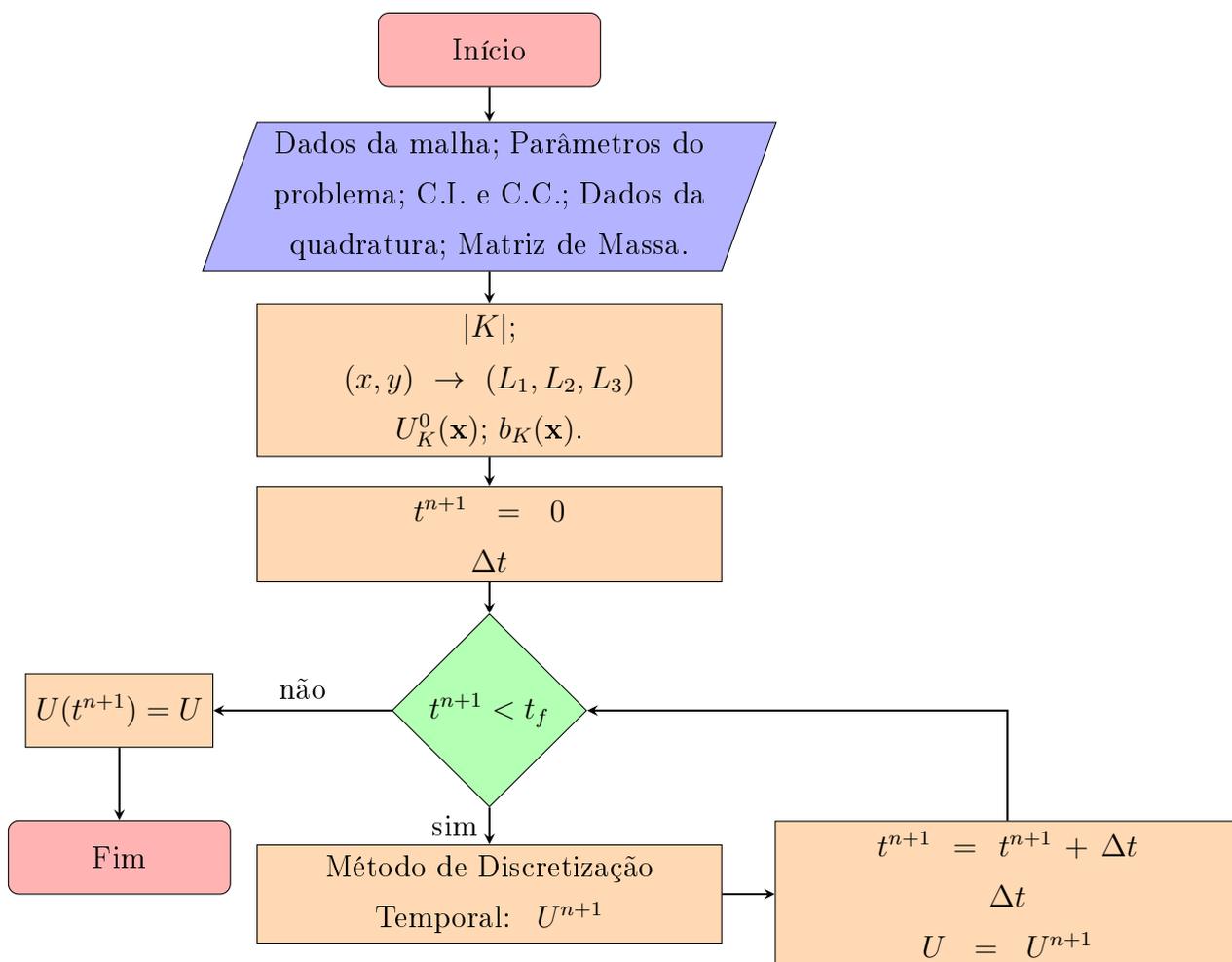


Figura 3.7: Fluxograma do Método Galerkin Descontínuo.

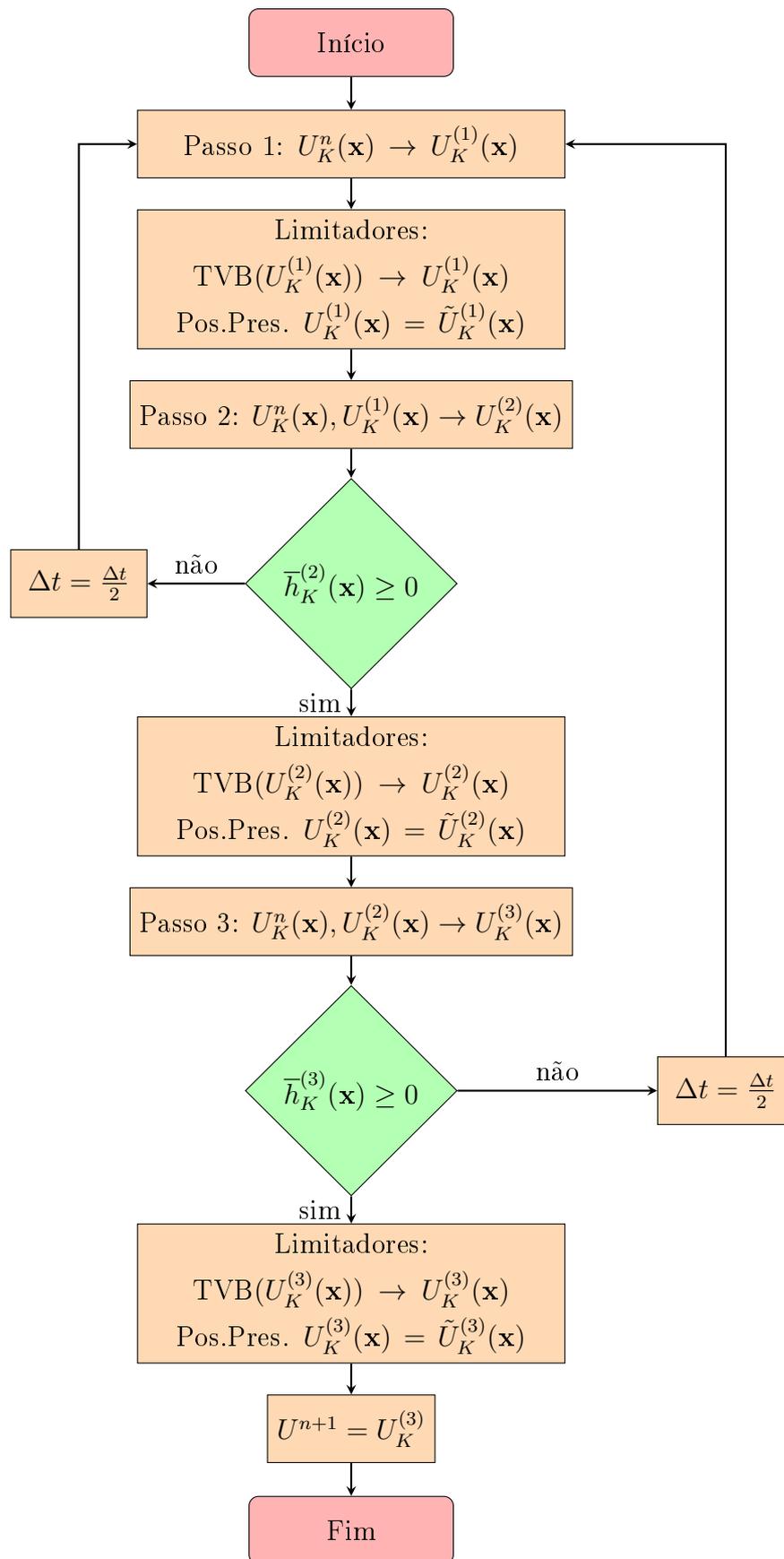


Figura 3.8: Fluxograma do Método de Discretização Temporal Runge-Kutta.

3.4 Integração Numérica

As integrais que aparecem nas equações (3.23) e (3.48) podem ser calculadas numericamente usando regras de quadratura, neste trabalho como em [65] usamos uma regra de quadratura Gaussiana especial a fim de que a condição CFL e o Teorema 3.1 sejam satisfeitos, para detalhes veja [68].

Assume-se que o polinômio DG no triângulo K é $p_K(x, y)$ de grau k . Seguindo a metodologia de [68], a primeira etapa é decompor o valor médio do elemento K como uma combinação convexa de valores pontuais do polinômio $p_K(x, y)$ por uma regra de quadratura satisfazendo:

- A regra de quadratura é exata para integração de $p_K(x, y)$ em K ;
- Todos os pesos de quadratura devem ser positivos;
- Os pontos de quadratura incluem todos os pontos da quadratura de Gauss para cada aresta e_K^i . Também é preciso encontrar os pesos de quadratura destes pontos, já que a condição CFL do método RKDG depende deles.

Para obter esta regra de quadratura especial primeiro considere uma regra de quadratura em um quadrado unitário Q com vértices $(-\frac{1}{2}, -\frac{1}{2})$, $(\frac{1}{2}, -\frac{1}{2})$, $(\frac{1}{2}, \frac{1}{2})$ e $(-\frac{1}{2}, \frac{1}{2})$ no plano (u, v) . Seja $\{\hat{u}^\alpha : \alpha = 1, \dots, N\}$ os pontos de quadratura Gauss-Lobatto no intervalo $[-\frac{1}{2}, \frac{1}{2}]$, onde N é o menor inteiro tal que $2N - 3 \geq k$, com pesos \hat{w}_α , e $\{v^\beta : \beta = 1, \dots, k + 1\}$ denota os pontos de quadratura Gauss em $[-\frac{1}{2}, \frac{1}{2}]$ com pesos w_β . Para o polinômio $p(u, v)$, pode-se usar o produto tensorial dos N pontos de Gauss-Lobatto para u e $k + 1$ pontos de Gauss para v como a regra de quadratura no quadrado, então os pontos podem ser escritos como $S_k = \{(\hat{u}^\alpha, v^\beta) : \alpha = 1, \dots, N; \beta = 1, \dots, k + 1\}$, veja a Figura 3.9(a) para S_2 . Esta é exata para um polinômio $p(u, v)$ se o grau de $p(u, v)$ com respeito à u não é maior que k e o grau com respeito à v não é maior que $2k + 1$.

Sejam os vértices \mathbf{V}_1 , \mathbf{V}_2 e \mathbf{V}_3 , no sentido horário, do elemento K e o vetor posição \mathbf{P} de um ponto arbitrário P em K que pode ser especificado pelas coordenadas de área $(\zeta_1, \zeta_2, \zeta_3)$, $\mathbf{P} = \zeta_1 \mathbf{V}_1 + \zeta_2 \mathbf{V}_2 + \zeta_3 \mathbf{V}_3$. Então, define-se as seguintes três funções:

$$\mathbf{g}_1(u, v) = \left(\frac{1}{2} + v\right) \mathbf{V}_1 + \left(\frac{1}{2} + u\right) \left(\frac{1}{2} - v\right) \mathbf{V}_2 + \left(\frac{1}{2} - u\right) \left(\frac{1}{2} - v\right) \mathbf{V}_3,$$

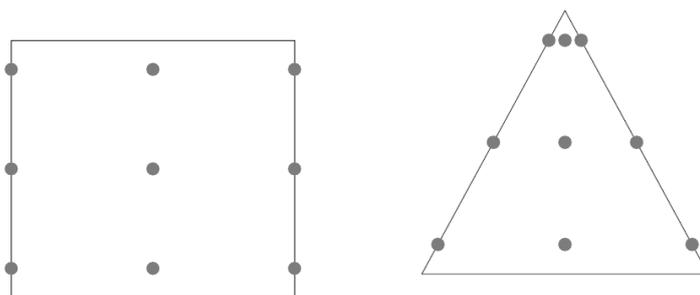
$$\mathbf{g}_2(u, v) = \left(\frac{1}{2} + v\right) \mathbf{V}_2 + \left(\frac{1}{2} + u\right) \left(\frac{1}{2} - v\right) \mathbf{V}_3 + \left(\frac{1}{2} - u\right) \left(\frac{1}{2} - v\right) \mathbf{V}_1,$$

$$\mathbf{g}_3(u, v) = \left(\frac{1}{2} + v\right) \mathbf{V}_3 + \left(\frac{1}{2} + u\right) \left(\frac{1}{2} - v\right) \mathbf{V}_1 + \left(\frac{1}{2} - u\right) \left(\frac{1}{2} - v\right) \mathbf{V}_2.$$

Cada uma destas funções é uma projeção do quadrado sobre o elemento K , veja a Figura 3.9. Usa-se \mathbf{g}_i ($i = 1, 2, 3$) e S_k para construir a quadratura triangular. Seja $p_k(x, y)$ o polinômio DG de duas variáveis e grau k com valor médio \bar{U}_K^n definido no triângulo K , logo

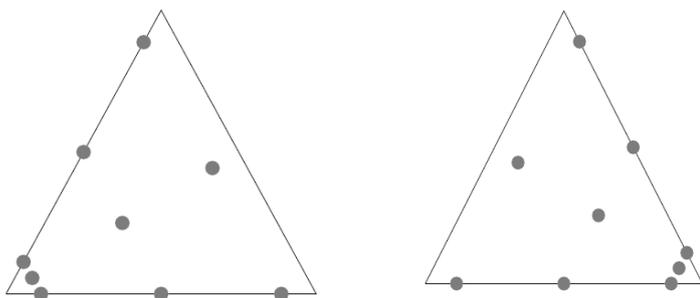
$$\bar{U}_K^n = \frac{1}{|K|} \iint_K p_K(x, y) dA(x, y) = \frac{1}{|K|} \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} p_k(\mathbf{g}_i(u, v)) \left| \frac{\partial \mathbf{g}_i(u, v)}{\partial(u, v)} \right| du dv \quad i = 1, 2, 3. \quad (3.51)$$

Lema 3.1 *Se a orientação dos três vértices \mathbf{V}_1 , \mathbf{V}_2 e \mathbf{V}_3 é horária, então o Jacobiano $\left| \frac{\partial \mathbf{g}_i(u, v)}{\partial(u, v)} \right| = 2|K| \left(\frac{1}{2} - v\right)$.*



(a) Pontos de Gauss e de Gauss-Lobatto no quadrado.

(b) Projeção 1.



(c) Projeção 2.

(d) Projeção 3.

Figura 3.9: Ilustração das três projeções para $k = 2$.

Fonte: Zhang et al [68].

Define-se $\hat{p}_K^i = p_K(\mathbf{g}_i(u, v)) \left| \frac{\partial \mathbf{g}_i(u, v)}{\partial(u, v)} \right|$, então pode-se calcular a integral em (3.51)

numericamente usando a quadratura. Logo,

$$\bar{U}_K^n = \frac{1}{|K|} \sum_{\alpha=1}^N \sum_{\beta=1}^{k+1} \hat{p}_K^i(\hat{u}^\alpha, v^\beta) w_\alpha \hat{w}_\beta = \sum_{\alpha=1}^N \sum_{\beta=1}^{k+1} p_K(\mathbf{g}_i(\hat{u}^\alpha, v^\beta)) 2 \left(\frac{1}{2} - v^\beta \right) w_\alpha \hat{w}_\beta \quad (3.52)$$

Então tem-se três regras de quadratura diferentes para $p_K^i(x, y)$ sobre K , e os pontos de quadratura são $\mathbf{g}_i(S_k)(i = 1, 2, 3)$ com pesos de quadratura positivos, veja as Figuras 3.9(b), 3.9(c), 3.9(d) para $k = 2$. Combinando os pontos destas três regras de quadratura, obtemos uma regra de quadratura com $3(N - 1)(k + 1)$ pontos, veja a Figura 3.10. Em coordenadas de área o conjunto S_k^K destes pontos de quadratura pode ser escrito como

$$\begin{aligned} S_k^K = & \left\{ \left(\frac{1}{2} + v^\beta, \left(\frac{1}{2} + \hat{u}^\alpha \right) \left(\frac{1}{2} - v^\beta \right), \left(\frac{1}{2} - \hat{u}^\alpha \right) \left(\frac{1}{2} - v^\beta \right) \right), \right. \\ & \left(\left(\frac{1}{2} - \hat{u}^\alpha \right) \left(\frac{1}{2} - v^\beta \right), \frac{1}{2} - v^\beta, \left(\frac{1}{2} + \hat{u}^\alpha \right) \left(\frac{1}{2} - v^\beta \right) \right), \\ & \left. \left(\left(\frac{1}{2} + \hat{u}^\alpha \right) \left(\frac{1}{2} - v^\beta \right), \left(\frac{1}{2} - \hat{u}^\alpha \right) \left(\frac{1}{2} - v^\beta \right), \frac{1}{2} - v^\beta \right) \right\}, \\ & \alpha = 1, \dots, N, \quad \beta = 1, \dots, k + 1. \end{aligned} \quad (3.53)$$

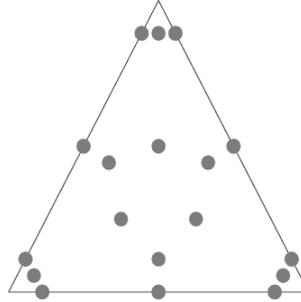


Figura 3.10: Pontos de quadratura em um triângulo para $k = 2$.

Fonte: Zhang et al [68].

Agora a média celular pode ser decomposta como:

$$\bar{U}_K^n = \frac{1}{3} \sum_{i=1}^3 \bar{U}_K^n = \sum_{i=1}^3 \sum_{\alpha=1}^N \sum_{\beta=1}^{k+1} p_K(\mathbf{g}_i(\hat{u}^\alpha, v^\beta)) \frac{2}{3} \left(\frac{1}{2} - v^\beta \right) w_\alpha \hat{w}_\beta = \sum_{\mathbf{x} \in S_k^K} p_K(\mathbf{x}) w_{\mathbf{x}}. \quad (3.54)$$

Em seguida, é necessário determinar os pesos de quadratura $w_{\mathbf{x}}$ para os pontos localizados na aresta e_K^1 que são $(0, \frac{1}{2} + v^\beta, \frac{1}{2} - v^\beta)$, os pontos $(\frac{1}{2} - v^\beta, 0, \frac{1}{2} + v^\beta)$ em e_K^2 , e os pontos $(0, \frac{1}{2} + v^\beta, \frac{1}{2} - v^\beta)$ em e_K^3 . Note que $\mathbf{g}_2(\frac{1}{2}, v^\beta)$ e $\mathbf{g}_3(-\frac{1}{2}, -v^\beta)$ são o mesmo

ponto $(0, \frac{1}{2} + v^\beta, 0, \frac{1}{2} - v^\beta)$. Visto que $\hat{w}_1 = \hat{w}_N$, o peso de $(0, \frac{1}{2} + v^\beta, 0, \frac{1}{2} - v^\beta)$ é

$$\frac{2}{3} \left(\frac{1}{2} + v^\beta \right) w_\beta \hat{w}_1 + \frac{2}{3} \left(\frac{1}{2} - v^\beta \right) w_\beta \hat{w}_N = \frac{2}{3} w_\beta \hat{w}_1. \quad (3.55)$$

Similarmente obtemos que os pesos para os demais pontos são $\frac{2}{3} w_\beta \hat{w}_1$.

Os valores de $p_K(x, y)$ nos pontos de quadratura nas arestas e_K^i podem ser denotados como $U_{i,\beta}^{int(K)}$. Podemos então escrever (3.54) como uma combinação linear de p_K nos pontos de quadratura S_k^K .

$$\bar{U}_K^n = \sum_{i=1}^3 \sum_{\beta=1}^{k+1} \frac{2}{3} w_\beta \hat{w}_1 U_{i,\beta}^{int(K)} + \sum_{\mathbf{x} \in \bar{S}_k^K} U_K(\mathbf{x}) w_{\mathbf{x}}, \quad (3.56)$$

onde \bar{S}_k^K são os pontos em S_k^K que estão no interior do triângulo K e

$$w_{\mathbf{x}} = \frac{2}{3} \left(\frac{1}{2} - v^\beta \right) w_\alpha \hat{w}_\beta.$$

Desse modo, podemos calcular as integrais duplas em (3.23) e (3.48) de forma análoga, por exemplo:

$$\iint_K \mathbf{F} \cdot \nabla N \, d\mathbf{x} = |K| \left(\sum_{i=1}^3 \sum_{\beta=1}^{k+1} \frac{2}{3} w_\beta \hat{w}_1 \mathbf{F}(U_{i,\beta}^{int(K)}) \cdot \nabla N(\mathbf{x}_{i,\beta}^{int(K)}) + \sum_{\mathbf{x} \in \bar{S}_k^K} \mathbf{F}(U_K(\mathbf{x})) \cdot \nabla N(\mathbf{x}) w_{\mathbf{x}} \right) \quad (3.57)$$

Já para a integral de linha presente em (3.23) podemos calcular da forma calculada abaixo:

$$\sum_{i=1}^3 \int_{e_K^i} \mathbf{F} \cdot n_K^i N \, ds = \sum_{i=1}^3 \sum_{\beta=1}^{k+1} w_\beta \mathbf{F}(U_{i,\beta}^{int(K)}) \cdot n_K^i N(\mathbf{x}_{i,\beta}) l_K^i \quad (3.58)$$

Em particular, nos testes numéricos realizados neste trabalho usamos polinômios lineares ($k = 1$), sendo necessários e suficientes 3 pontos de quadratura Gauss-Lobatto e 2 de Gauss, como segue

$$\hat{u} = \left\{ -\frac{1}{2}, 0, \frac{1}{2} \right\}, \quad v = \left\{ -\frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{3}} \right\}$$

com os seus respectivos pesos

$$\hat{w} = \left\{ \frac{1}{6}, \frac{2}{3}, \frac{1}{6} \right\}, \quad w = \left\{ \frac{1}{2}, \frac{1}{2} \right\},$$

no intervalo $[-1/2, 1/2]$. Dessa forma o conjunto S_K é formado por 12 pontos de

quadratura, onde \tilde{S}_K contém 6 destes pontos, dois em cada aresta, e \bar{S}_K também contém 6 pontos, veja a Figura 3.11.

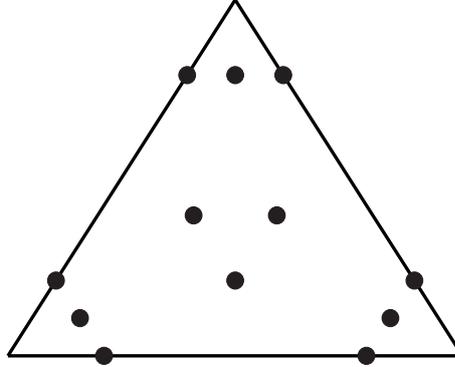


Figura 3.11: Pontos de quadratura utilizados.

Fonte: Imagem do autor.

Onde,

$$\tilde{S}_K = \left\{ \begin{array}{l} (0.7887, 0.2113, 0); (0.2113, 0.7887, 0); \\ (0, 0.7887, 0.2113); (0, 0.2113, 0.7887); \\ (0.2113, 0, 0.7887); (0.7887, 0, 0.2113) \end{array} \right\} \quad (3.59)$$

$$\bar{S}_K = \left\{ \begin{array}{l} (0.2113, 0.3943, 0.3943); (0.3943, 0.2113, 0.3943); \\ (0.3943, 0.3943, 0.2113); (0.7887, 0.1057, 0.1057); \\ (0.1057, 0.7887, 0.1057); (0.1057, 0.1057, 0.7887) \end{array} \right\} \quad (3.60)$$

Capítulo 4

Gerador de Malhas

O primeiro passo para se usar o método Galerkin Descontínuo ou qualquer outro método de elementos finitos é gerar uma malha não estruturada, para isso utilizados o gerador de malhas Distmesh desenvolvido por Per-Olof Persson [46].

O gerador Distmesh foi elaborado baseado nas seguintes ideias:

- Uma malha não estruturada requer uma escolha de nós, os vértices, e uma triangulação;
- A malha deve ter pequenos elementos para resolver os detalhes da geometria, mas tamanhos maiores, sempre que possível, para reduzir o número total de nós;
- É necessário elementos de alta qualidade (por exemplo, triângulos que são quase equiláteros) para obter precisão discretizações e sistemas lineares bem condicionados.

Para obter as condições acima, o gerador foi desenvolvido utilizando o método de refinamento Delaunay, que parte de uma triangulação inicial e se refina até a qualidade do elemento seja suficientemente alta. Durante os movimentos do nó, altera-se a conectividade da malha para melhorar as qualidades do elemento, calculando novamente a triangulação de Delaunay ou atualizando-a localmente. Em equilíbrio, os elementos da malha tendem a ser de qualidade muito alta, e o método se generaliza para dimensões mais altas, veja alguns exemplos de malha geradas pelo Distmesh na Figura 4.1. O gerador foi implementado na linguagem MATLAB[®] [1].

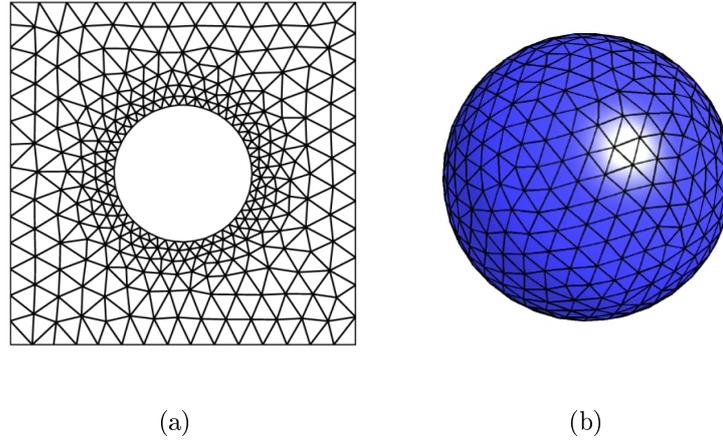


Figura 4.1: Malhas geradas pelo Distmesh.
Fonte: Persson [46].

4.1 O Algoritmo

O algoritmo Distmesh é baseado em uma simples analogia mecânica entre uma malha triangular e uma estrutura de treliça 2-D, ou equivalentemente a uma estrutura de molas. No modelo físico, as arestas dos triângulos correspondem as barras, e os pontos equivalem as juntas da treliça. Cada barra tem uma relação de deslocamento de força $f(l, l_0)$ dependendo de seu comprimento atual l e seu comprimento não estendido l_0 .

Em cada nó na fronteira, há uma força de reação atuando normal à fronteira. A magnitude dessa força é grande o suficiente para impedir que o nó se mova para fora. As posições das juntas são encontradas através de uma força de equilíbrio na estrutura. Para resolver o equilíbrio da força, o gerador coleta as coordenadas x e y de todos os N pontos na malha em uma matriz \mathbf{p} , de tamanho N por 2 :

$$\mathbf{p} = [\mathbf{x} \quad \mathbf{y}]. \quad (4.1)$$

O vetor força $\mathbf{F}(\mathbf{p})$ tem componentes horizontal e vertical em cada ponto da malha:

$$\mathbf{F}(\mathbf{p}) = [\mathbf{F}_{int,x}(\mathbf{p}) \quad \mathbf{F}_{int,y}(\mathbf{p})] + [\mathbf{F}_{ext,x}(\mathbf{p}) \quad \mathbf{F}_{ext,y}(\mathbf{p})], \quad (4.2)$$

onde \mathbf{F}_{int} contém as forças internas da barra, e \mathbf{F}_{ext} são as forças externas.

O algoritmo Delaunay determina triângulos não sobrepostos que preenchem o domínio desejado, de modo que cada aresta é compartilhada por no máximo dois triângulos. Para isso o sistema $\mathbf{F}(\mathbf{p}) = 0$ deve ser resolvido para um conjunto de posições de equilíbrio \mathbf{p} .

A fim de resolver $\mathbf{F}(\mathbf{p}) = 0$ introduziu-se uma dependência artificial de tempo. Para alguns $\mathbf{p}(0) = \mathbf{p}_0$, considera-se o sistema de EDOs (em unidades não físicas!)

$$\frac{d\mathbf{p}}{dt} = \mathbf{F}(\mathbf{p}), \quad t \geq 0. \quad (4.3)$$

Se uma solução estacionária é encontrada, ela satisfaz nosso sistema $\mathbf{F}(\mathbf{p}) = 0$. O sistema (4.3) é aproximado usando o método Euler avançado. No tempo discreto $t_n = n\Delta t$, a solução aproximada $\mathbf{p}_n \approx \mathbf{p}(t_n)$ é atualizada por

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \Delta t \mathbf{F}(\mathbf{p}_n). \quad (4.4)$$

Ao avaliar a função de força, as posições \mathbf{p}_n são conhecidas e, portanto, também a triangulação do conjunto de pontos atual. As forças de reação externas entram da seguinte maneira: todos os pontos que saem da região durante a atualização de \mathbf{p}_n para \mathbf{p}_{n+1} são movidos de volta para o ponto de fronteira mais próximo.

A função $k(l_0 - l)$ modela as molas lineares comuns. A implementação feita usa essa resposta linear para as forças repulsivas, mas não permite forças atrativas:

$$f(l, l_0) = \begin{cases} k(l_0 - l) & \text{se } l < l_0, \\ 0 & \text{se } l \geq l_0. \end{cases} \quad (4.5)$$

É razoável exigir $f = 0$ para $l = l_0$. O tratamento proposto para as fronteiras significa que nenhum ponto é forçado a permanecer na fronteira, eles são simplesmente impedidos de atravessá-lo. É importante que a maioria das barras dê forças repulsivas $f > 0$, para ajudar os pontos espalhados por toda a geometria. Isto significa que $f(l, l_0)$ deve ser positivo quando estiver perto do comprimento desejado, o que pode ser obtido escolhendo l_0 ligeiramente maior que o comprimento que realmente desejamos.

Nessa implementação, a distribuição de comprimento de aresta desejada é fornecida pelo usuário como uma função de tamanho de elemento $h(x, y)$, ele fornece a distribuição relativa sobre o domínio. Isso evita uma conexão implícita com o número de nós, que o usuário não é solicitado a especificar. Para encontrar a escala, calculamos a relação entre a área da malha a partir dos comprimentos l_i da aresta atual e o “tamanho desejado” (de $h(x, y)$ nos pontos médios (x_i, y_i) das barras):

$$\text{Fator Escala} = \left(\frac{\sum l_i^2}{\sum h(x_i, y_i)^2} \right)^{1/2}. \quad (4.6)$$

As posições iniciais do nó \mathbf{p}_0 podem ser escolhidas de várias maneiras. Uma distribuição aleatória dos pontos geralmente funciona bem. Para malhas destinadas a ter tamanhos de elementos uniformes (e para geometrias simples), bons resultados são obtidos a partir de pontos igualmente espaçados. Quando uma distribuição de tamanho não uniforme $h(x, y)$ é desejada, a convergência é mais rápida se a distribuição inicial for ponderada por probabilidades proporcionais a $1/h(x, y)^2$. O método de rejeição começa com uma malha inicial uniforme dentro do domínio e descarta pontos usando essa probabilidade.

4.2 Implementação

O código fonte completo para o gerador de malha bidimensional, veja a Figura 4.2, será explicado em detalhes abaixo, seguimos [46].

A primeira linha especifica a sintaxe de chamada para a função `distmesh2d`:
`function [p, t] = distmesh2d(fd, fh, h0, bbox, pfix, varargin)`

Essa função produz as seguintes saídas:

- p : uma matriz N por 2, que contém as coordenadas x e y para cada um dos N nós da malha;
- t : conhecido como matriz de conexão, cada linha está associada a um triângulo e as colunas especificam que nós da malha compõe o elemento.

Os argumentos de entrada são os seguintes:

- A geometria é dada como uma função de distância `fd`;
- A função de comprimento de aresta desejada, $h(x, y)$ é dada como uma função `fh`, que retorna h para todos os pontos de entrada;
- O parâmetro `h0` é a distância entre os pontos na distribuição inicial p_0 ;
- A caixa delimitadora da região é uma matriz `bbox = [xmin, ymin; xmax, ymax]`;
- As posições de nós fixos são dadas como uma matriz `pfix` com duas colunas, onde essas correspondem as coordenadas x e y , respectivamente;
- Parâmetros adicionais para as funções `fd` e `fh` podem ser dados no último argumentos `varargin` (para mais informações use o `help` do MATLAB).

```

function [p,t]=distmesh2d(fd,fh,h0,bbox,pfix,varargin)

dptol=.001; ttol=.1; Fscale=1.2; deltat=.2; geps=.001*h0; deps=sqrt(eps)*h0;

% 1. Create initial distribution in bounding box (equilateral triangles)
[x,y]=meshgrid(bbox(1,1):h0:(bbox(2,1)+h0),bbox(1,2):h0*sqrt(3)/2:(bbox(2,2)));
x(2:2:end,:)=x(2:2:end,:)+h0/2; % Shift even rows
p=[x(:),y(:)]; % List of node coordinates

% 2. Remove points outside the region, apply the rejection method
p=p(feval(fd,p,varargin{:})<geps,:); % Keep only d<0 points
r0=1./feval(fh,p,varargin{:}).^2; % Probability to keep point
p=[pfix; p(rand(size(p,1),1)<r0./max(r0),:)]'; % Rejection method
N=size(p,1); % Number of points N

pold=inf; % For first iteration
while 1
% 3. Triangulation by the Delaunay algorithm
if max(sqrt(sum((p-pold).^2,2))/h0)>ttol % Any large movement?
pold=p; % Save current positions
t=delaunayn(p); % List of triangles
pmid=(p(t(:,1),:)+p(t(:,2),:)+p(t(:,3),:))/3; % Compute centroids
t=t(feval(fd,pmid,varargin{:})<-geps,:); % Keep interior triangles
% 4. Describe each bar by a unique pair of nodes
bars=[t(:, [1,2]); t(:, [1,3]); t(:, [2,3])]; % Interior bars duplicated
bars=unique(sort(bars,2), 'rows'); % Bars as node pairs
% 5. Graphical output of the current mesh
trimesh(t,p(:,1),p(:,2),zeros(N,1))
view(2),axis equal,axis off,drawnow
end

% 6. Move mesh points based on bar lengths L and forces F
barvec=p(bars(:,1),:)-p(bars(:,2),:); % List of bar vectors
L=sqrt(sum(barvec.^2,2)); % L = Bar lengths
hbars=feval(fh,(p(bars(:,1),:)+p(bars(:,2),:))/2,varargin{:});
L0=hbars*Fscale*sqrt(sum(L.^2)/sum(hbars.^2)); % L0 = Desired lengths
Fvec=F./L*[1,1].*barvec; % Bar forces (x,y components)
Ftot=full(sparse(bars(:, [1,1,2,2]),ones(size(F))*[1,2,1,2],[Fvec,-Fvec],N,2));
Ftot(1:size(pfix,1),:)=0; % Force = 0 at fixed points
p=p+deltat*Ftot; % Update node positions

% 7. Bring outside points back to the boundary
d=feval(fd,p,varargin{:}); ix=d>0; % Find points outside (d>0)
dgradx=(feval(fd,[p(ix,1)+deps,p(ix,2)],varargin{:})-d(ix))/deps; % Numerical
dgrady=(feval(fd,[p(ix,1),p(ix,2)+deps],varargin{:})-d(ix))/deps; % gradient
p(ix,:)=p(ix,:)-[d(ix).*dgradx,d(ix).*dgrady]; % Project back to boundary

% 8. Termination criterion: All interior nodes move less than dptol (scaled)
if max(sqrt(sum(deltat*Ftot(d<-geps,:).^2,2))/h0)<dptol, break; end
end

```

Figura 4.2: Código distmesh2d.m para gerar uma malha 2D.

Fonte: Persson [46].

O algoritmo parará quando todos os movimentos em uma iteração (em relação ao comprimento médio da barra) forem menores que dptol. Da mesma forma, o ttol controla

até que ponto os pontos podem se mover (relativamente) antes de uma nova triangulação Delaunay. A “pressão interna” é controlada pela F_{scale} . O intervalo de tempo no método de Euler (4.4) é Δt e ϵ é a tolerância nas avaliações de geometria. A raiz quadrada de ϵ é a tolerância da máquina é Δx na equação numérica diferencial da função da distância.

Vamos descrever os passos de 1 ao 8 do algoritmo `distmesh2d`, que basicamente consiste em distribuir os pontos, realizar a triangularização e finalmente determinar a força de equilíbrio, veja a Figura 4.3.

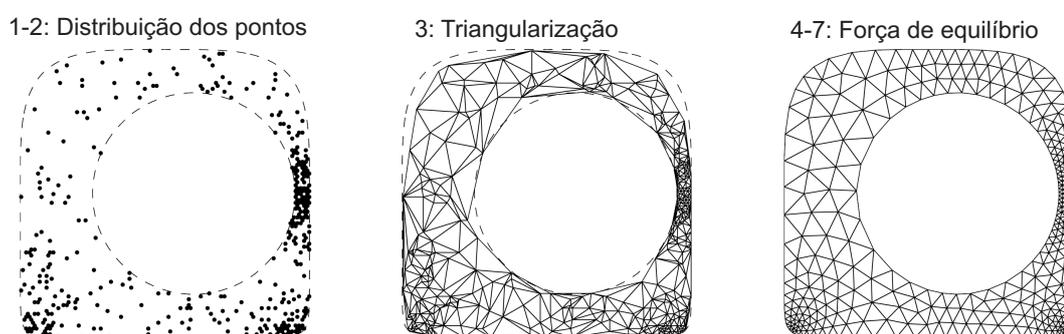


Figura 4.3: Procedimento para gerar uma malha triangular não uniforme.

Fonte: Persson [46].

1. O primeiro passo cria uma distribuição uniforme dos nós dentro da caixa delimitadora da geometria. A função `meshgrid` gera uma grade retangular, dada como dois vetores x e y das coordenadas do nó. Inicialmente as distâncias são $\sqrt{3}h_0/2$ na direção y . Ao deslocar cada segunda linha $h_0/2$ para a direita, todos os pontos estarão a uma distância h_0 de seus vizinhos mais próximos. As coordenadas são armazenadas na matriz `p`.
2. A próxima etapa remove todos os nós fora da geometria desejada: `feval` chama a função de distância `fd`, com as posições de nó `p` e os argumentos adicionais `varargin` como entradas. O resultado é um vetor coluna de distâncias dos nós até a fronteira da geometria. Somente os pontos interiores com distâncias negativas (permitindo uma tolerância `gep`) são mantidos. Então, avaliamos $h(x, y)$ em cada nó e rejeitamos pontos com uma probabilidade proporcional a $1/h(x, y)^2$. A matriz de nós fixos do usuário é colocada nas primeiras linhas de `p`.
3. Agora o código entra no loop principal, onde a localização dos N pontos é melhorada iterativamente. Inicializa-se a variável `pold` para a primeira iteração e inicia o loop

(o critério de finalização vem depois). Para economizar tempo de computação, uma nova triangulação é necessária quando o deslocamento máximo desde a última triangulação é maior que o t_{tol} (relativo ao tamanho aproximado do elemento l_0). As localizações dos nós após a nova triangulação são armazenadas em $pold$ e todas as iterações comparam os locais atuais p com $pold$. A função Delaunay do MATLAB gera uma triangulação t , e os triângulos fora da geometria devem ser removidos. Usamos uma solução simples aqui - se o centroide de um triângulo tiver $d > 0$, esse será removido.

4. Descreve cada barra por um único par de nós. Ao criar uma lista de arestas, cada triângulo contribui com três pares de nós. Como a maioria dos pares aparecerá duas vezes (as arestas estão em dois triângulos), as duplicatas são removidas nesta etapa.
5. Saída gráfica da malha atual.
6. Cada barra é um vetor de duas componentes em $barvec$; seu comprimento está em L . Os comprimentos desejados L_0 vêm avaliando $h(x, y)$ no ponto médio de cada barra. Nós multiplicamos pelo fator de escala em (4.6) e pelo fator fixo $Fscale$, para assegurar que a maioria das barras dê forças repulsivas $f > 0$ em F . A força resultante F_{tot} é a soma dos vetores de força em $Fvec$, de todas as barras que se encontram em um nó. Uma força de alongamento tem sinal positivo e sua direção é dada pelo vetor $bars$ de duas componentes. Observe que o F_{tot} para os nós fixos é definido como zero.
7. Se um ponto terminar fora da geometria após a atualização de p , ele será movido de volta para o ponto mais próximo na fronteira (usando a função de distância). Isso corresponde a uma força de reação normal à fronteira. O gradiente de $d(x, y)$ dá a direção (negativa) ao ponto de fronteira mais próximo.
8. Finalmente o último passo, o critério de terminação é baseado no movimento máximo do nó na iteração atual (excluindo os pontos de fronteira). Este critério é por vezes muito restritivo, e uma malha de alta qualidade é muitas vezes conseguida muito antes da terminação.

Capítulo 5

Verificação Computacional do Código e Experimentos Numéricos

Neste capítulo serão apresentados resultados numéricos obtidos pelo método de elementos finitos RKDG, descrito no capítulo 3, quando utilizados os métodos Galerkin Descontínuo e Runge-Kutta de terceira ordem na discretização espacial e temporal, respectivamente, para algumas leis de conservação e as equações de Shallow Water. Realizamos também alguns testes numéricos visando a verificação da propriedade “well-balanced” do método e a importância da aplicação do limitador TVB, bem como a abordagem adotada no seu emprego em elementos presentes na fronteira.

Na implementação do método foram utilizados polinômios lineares ($k = 1$), visto que os trabalhos, usados como referência na aplicação do limitador TVB, não expõem de forma clara como empregar o limitador quando se utiliza aproximações por polinômios quadráticos. Sendo assim foram necessários dois pontos de quadratura de Gauss e três de Lobatto, totalizando doze pontos de quadratura em S_K , veja a Figura 3.11.

O valor do parâmetro Λ , necessário no limitador TVB, foi estimado baseado em testes numéricos realizados.

A fim de verificar as soluções numéricas obtidas, para o caso das leis de conservação escalares em 2D estas foram comparadas qualitativamente com as obtidas pelo método Lagrangian-Eulerian (LEH), desenvolvido por Pérez e Abreu [45]. Os resultados obtidos através do método LEH foram adquiridos por meio do código computacional cedido pelo Dr. John Pérez. Para o caso das equações “Shallow Water”, estas foram comparadas qualitativamente com os resultados obtidos pelo Xing e Zhang em [65].

Na geração dos resultados foi utilizado um notebook Dell com processador Intel Core

i5, memória 4GB RAM e disco rígido de 1 TB. O método foi implementado na linguagem computacional Matlab.

5.1 Malhas

Nesta seção iremos mostrar as malhas utilizadas para obter as soluções numéricas em cada problema.

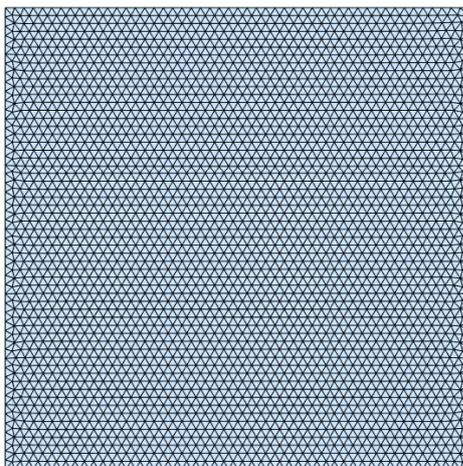


Figura 5.1: Malha com 5840 elementos ($\tau = 0.287$), onde $(x, y) \in [0, 1] \times [0, 1]$.

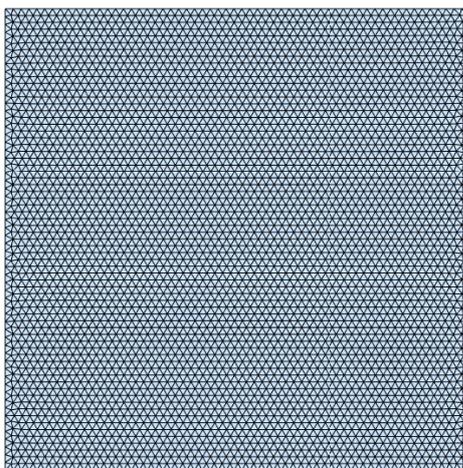


Figura 5.2: Malha com 8072 elementos ($\tau = 0.242$), onde $(x, y) \in [0, 1] \times [0, 1]$.

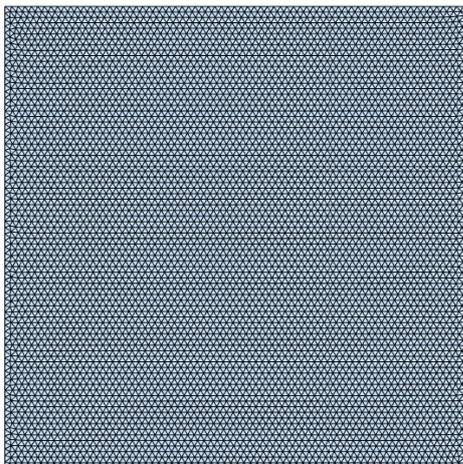


Figura 5.3: Malha com 11924 elementos ($\tau = 0.206$), onde $(x, y) \in [0, 1] \times [0, 1]$.

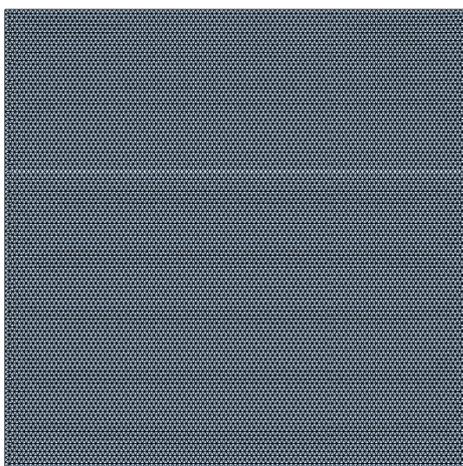


Figura 5.4: Malha com 23282 elementos ($\tau = 0.143$), onde $(x, y) \in [0, 1] \times [0, 1]$.

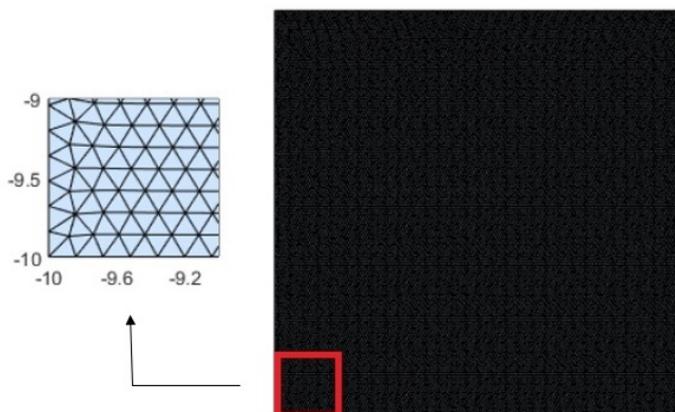


Figura 5.5: Malha com 36354 elementos ($\tau = 0.2264$), onde $(x, y) \in [-10, 10] \times [-10, 10]$.

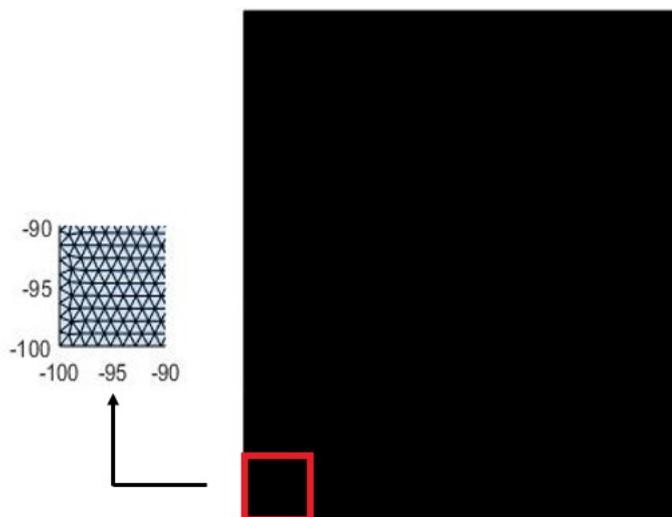


Figura 5.6: Malha com 64600 elementos ($\tau = 1.6942$), onde $(x, y) \in [-100, 100] \times [-100, 100]$.

5.2 Teste do Limitador TVB

A fim de mostrar a importância do limitador TVB, validar o nosso código e conferir se o tratamento, desenvolvido por nós, nos elementos que pertencem à fronteira funciona, iremos resolver dois problemas de valor inicial distintos. Para realizar esse estudo iremos obter a solução numérica dos PVIs abordando o limitador das seguintes formas:

- Limitador não é aplicado;

- Limitador é aplicado somente nos elementos que não pertençam a fronteira;
- Limitador é aplicado em todos os elementos e mudamos o valor de Λ de maneira adequada.

Teste 1: Equação de Advecção Linear

Consideramos a equação Adveção Linear

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \quad (5.1)$$

e o dado inicial:

$$u(x, y, 0) = \sin(\pi(x + y)), \quad (5.2)$$

com $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 0.5]$ [45].

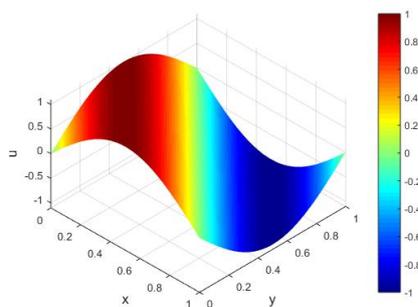


Figura 5.7: Condição Inicial.

Teste 2: Equação de Burgers

Consideramos neste exemplo a seguinte lei de conservação

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) + \frac{\partial}{\partial y} \left(\frac{u^2}{2} \right) = 0. \quad (5.3)$$

e o seguinte dado inicial associado à equação:

$$u(x, y, 0) = \begin{cases} 2, & x < 0.25, y < 0.25 \\ 3, & x > 0.25, y > 0.25 \\ 1, & \text{caso contrário,} \end{cases} \quad (5.4)$$

com $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 0.5]$ [45].

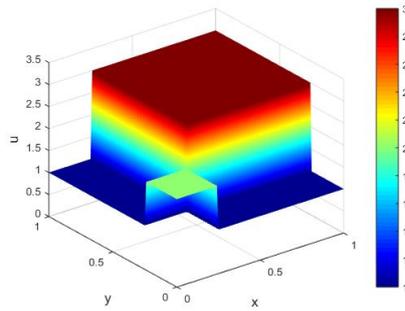


Figura 5.8: Condição Inicial.

De acordo com os resultados apresentados, veja as Figuras 5.9 e 5.10, notamos que o uso do limitador é crucial para obter uma solução numérica correta e que a sua não aplicação pode gerar muitas oscilações, de forma que estas prejudiquem a solução numérica e até impossibilitem a obtenção das soluções em determinados tempos. De fato isto impossibilitou a obtenção do resultado numérico no tempo desejado, nas Figuras 5.9(a) e 5.10(a) temos que o tempo é menor do que as demais, pois as oscilações espúrias estragaram a solução.

Podemos observar também que o tratamento desenvolvido por nós para tratar os elementos presentes no contorno funcionou de forma correta, veja as Figuras 5.9(d) e 5.10(d).

Como já explicado anteriormente a escolha do Λ é muito importante, visto que se este for muito grande oscilações espúrias relevantes podem não ser eliminadas, veja as Figuras 5.9(c) e 5.10(c). Nas Figuras 5.9(d) e 5.10(d) vemos que $\Lambda = 0.5$ parece ser adequado, pois este foi capaz de retirar as oscilações presentes quando este toma o valor igual a 50.

Logo, de acordo com os teste realizados podemos perceber que nosso algoritmo conseguiu eliminar as oscilações espúrias relevantes e funcionou bem com o refinamento de malha, veja as Figuras 5.9(e)-5.9(f) e 5.10(e)-5.10(f).

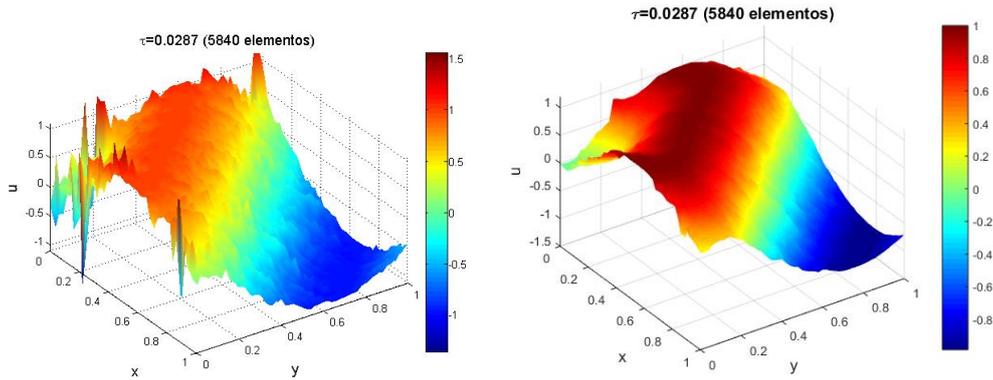
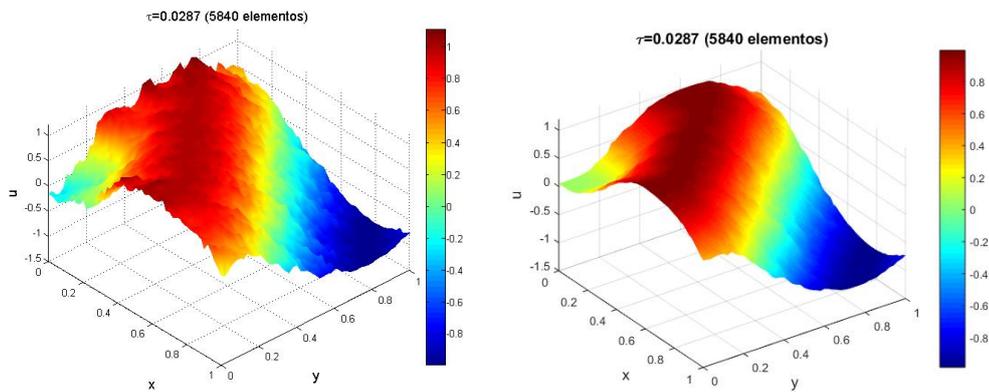
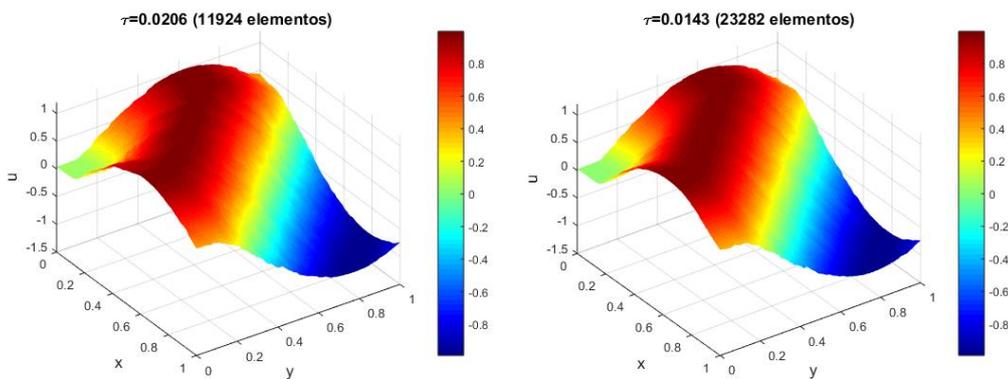
(a) Limitador não aplicado, $t = 0.05$.(b) Limitador aplicado somente nos elementos que não pertencem ao contorno ($\Lambda = 0.5$).(c) Limitador aplicado em todos os elementos ($\Lambda = 50$).(d) Limitador aplicado em todos os elementos ($\Lambda = 0.5$).(e) Limitador aplicado em todos os elementos ($\Lambda = 0.5$).(f) Limitador aplicado em todos os elementos ($\Lambda = 0.5$).

Figura 5.9: Resultados obtidos para o Teste 1, em (a) $t = 0.05$ e nas figuras (b)–(f) $t = 0.1$.

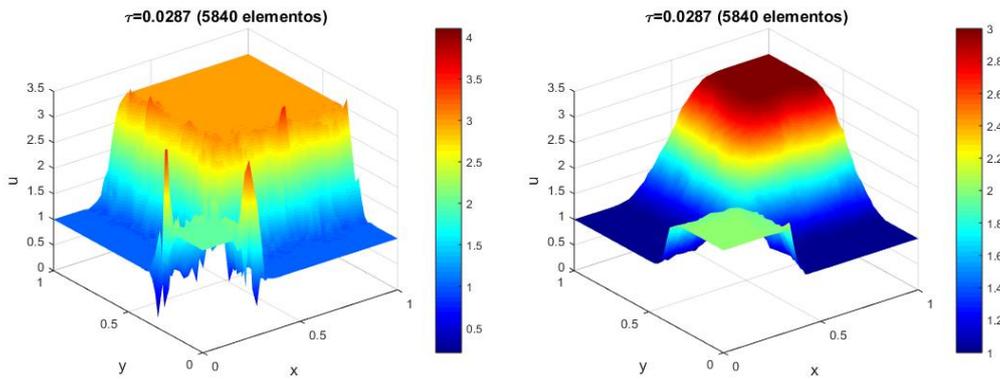
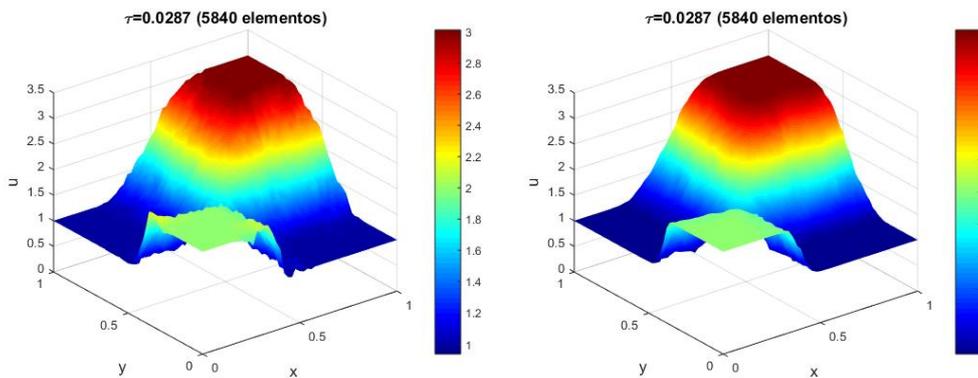
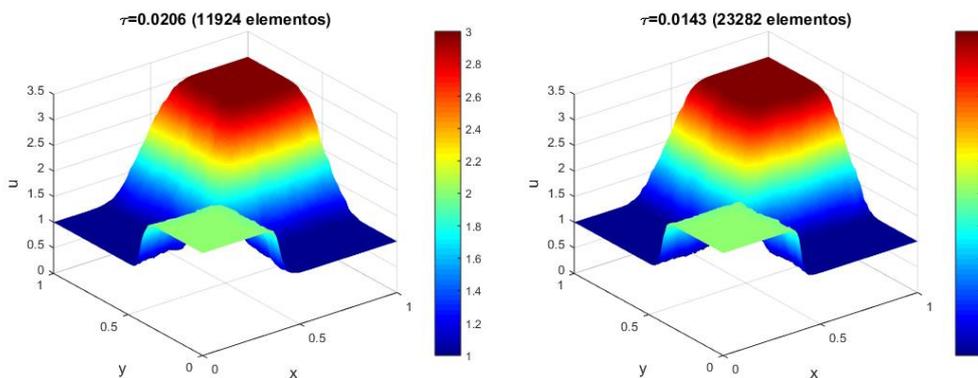
(a) Limitador não é aplicado, $t = 0.015$.(b) Limitador aplicado somente nos elementos que não pertencem ao contorno ($\Lambda = 0.5$).(c) Limitador aplicado em todos os elementos ($\Lambda = 50$).(d) Limitador aplicado em todos os elementos ($\Lambda = 0.5$).(e) Limitador aplicado em todos os elementos ($\Lambda = 0.5$).(f) Limitador aplicado em todos os elementos ($\Lambda = 0.5$).

Figura 5.10: Resultados obtidos para o Teste 2, em (a) $t = 0.015$ e nas figuras (b)–(f) $t = 1/12$.

5.3 Leis de Conservação

Problema 1. Equação de Burgers Bidimensional

Consideramos o seguinte problema de valor inicial bidimensional associado à equação de Burgers:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) + \frac{\partial}{\partial y} \left(\frac{u^2}{2} \right) = 0, \quad (5.5)$$

com $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 0.5]$, e dado inicial:

$$u(x, y, 0) = \begin{cases} 2, & x < 0.25, y < 0.25 \\ 3, & x > 0.25, y > 0.25 \\ 1, & \text{caso contrário.} \end{cases} \quad (5.6)$$

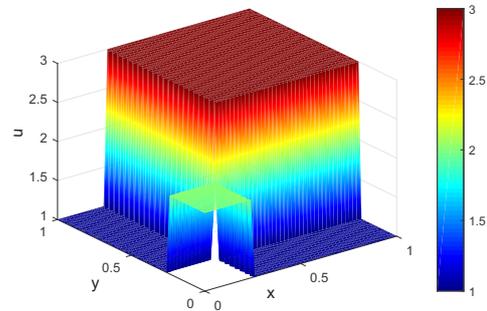


Figura 5.11: Condição Inicial.

Nesta lei de conservação temos que os fluxos $f(u) = g(u) = \frac{u^2}{2}$ são convexos, podemos fazer uma análise qualitativa utilizando a teoria sobre leis de conservação em 1D explicada no capítulo 2, assim vemos que para $u_l = 2$ (degrau de cor verde) e $u_r = 1$ (degrau de cor azul) é esperado que surja uma onda de choque, pois $u_l > u_r$, já para $u_l = 1$ (degrau de cor azul) e $u_r = 3$ (degrau de cor vermelha) é esperado uma onda de rarefação, porque $u_l < u_r$.

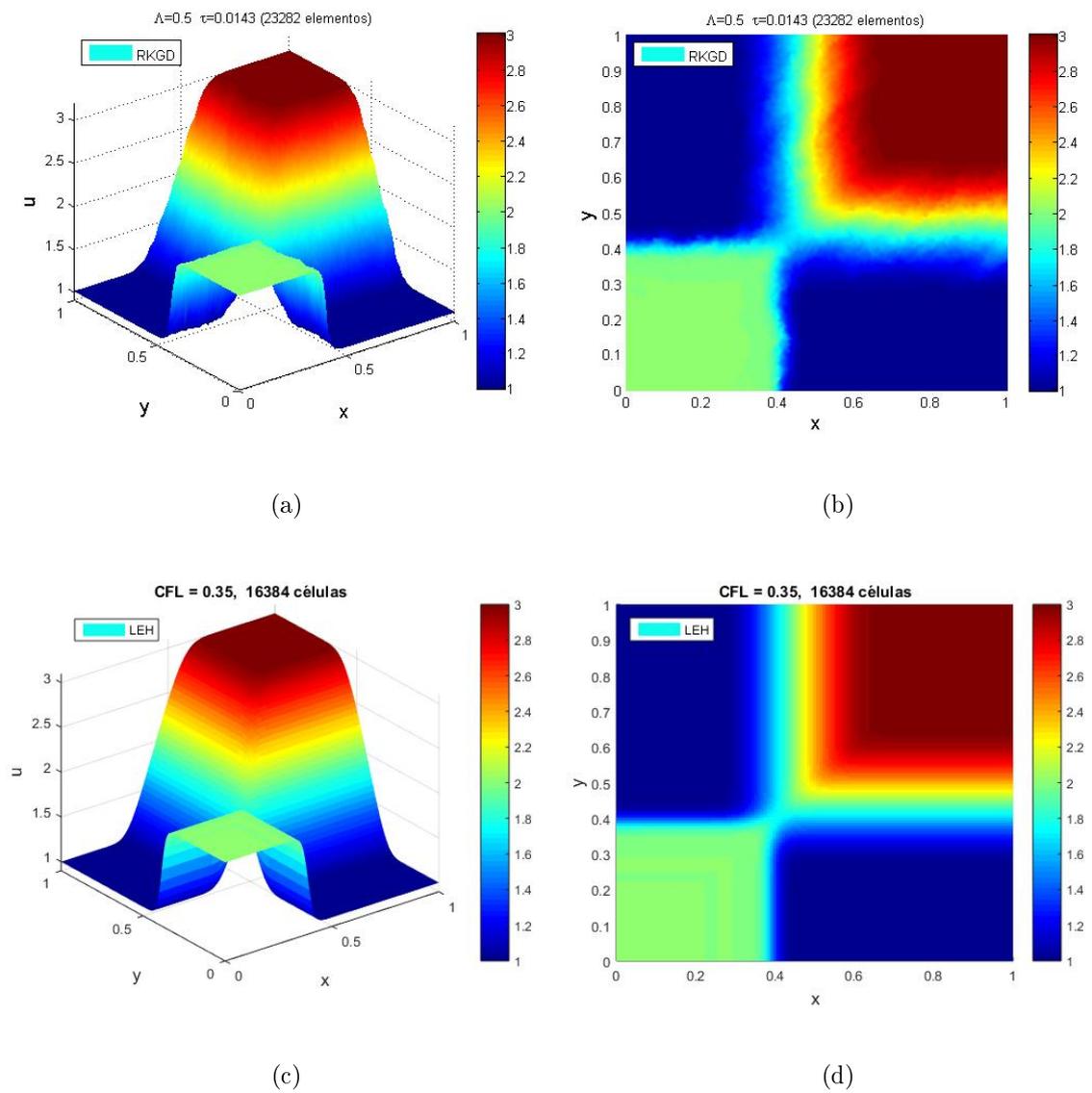


Figura 5.12: Resultado obtido para o PVI (5.5)–(5.6) para $t = 1/12$.

Problema 2. Equação de Burgers

Para a Equação de Burgers também consideramos a seguinte condição inicial por partes:

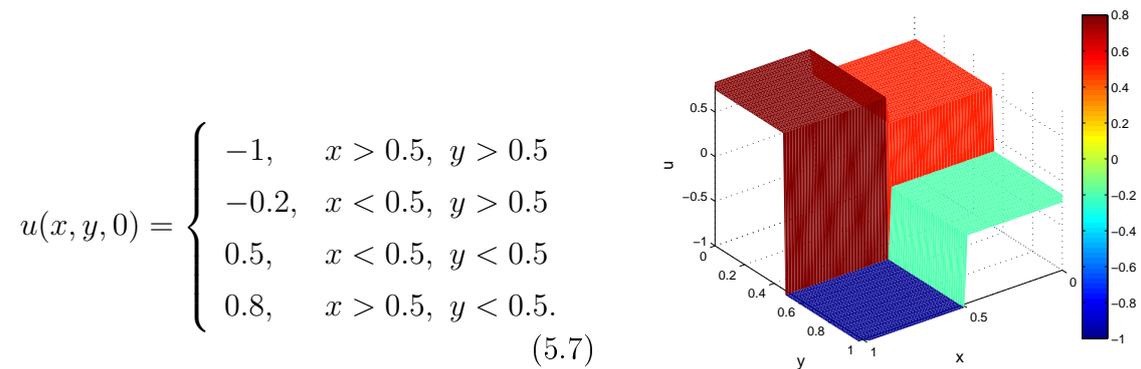


Figura 5.13: Condição Inicial.

com $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 0.5]$.

Para este dado inicial analisando ao longo do eixo x esperamos que surja uma onda de choque entre os dados iniciais $u_l = -0.2$ (degrau de cor verde) e $u_r = -1$ (degrau de cor azul), uma onda de rarefação entre $u_l = 0.5$ (degrau de cor vermelha) e $u_r = 0.8$ (degrau de cor vinho). E ao longo do eixo y ondas de choque entre os dados iniciais $u_l = 0.8$ (degrau de cor vinho) e $u_r = -1$ (degrau de cor azul), e $u_l = 0.5$ (degrau de cor vermelha) e $u_r = -0.2$ (degrau de cor verde).

De acordo com o resultado obtido, através do método Galerkin Descontínuo, podemos perceber que as oscilações espúrias que surgiram estragaram a solução, veja a Figura 5.14(a), de forma que esta não seja muito boa, de fato podemos notar que estas oscilações espúrias prejudicaram principalmente a captura das ondas de choque, visto que a onda de rarefação foi obtida de forma correta. O ideal para solucionar este problema talvez seria uma melhor estimativa do parâmetro Λ do limitador TVB, refinamento de malha ou utilizar polinômios quadráticos.

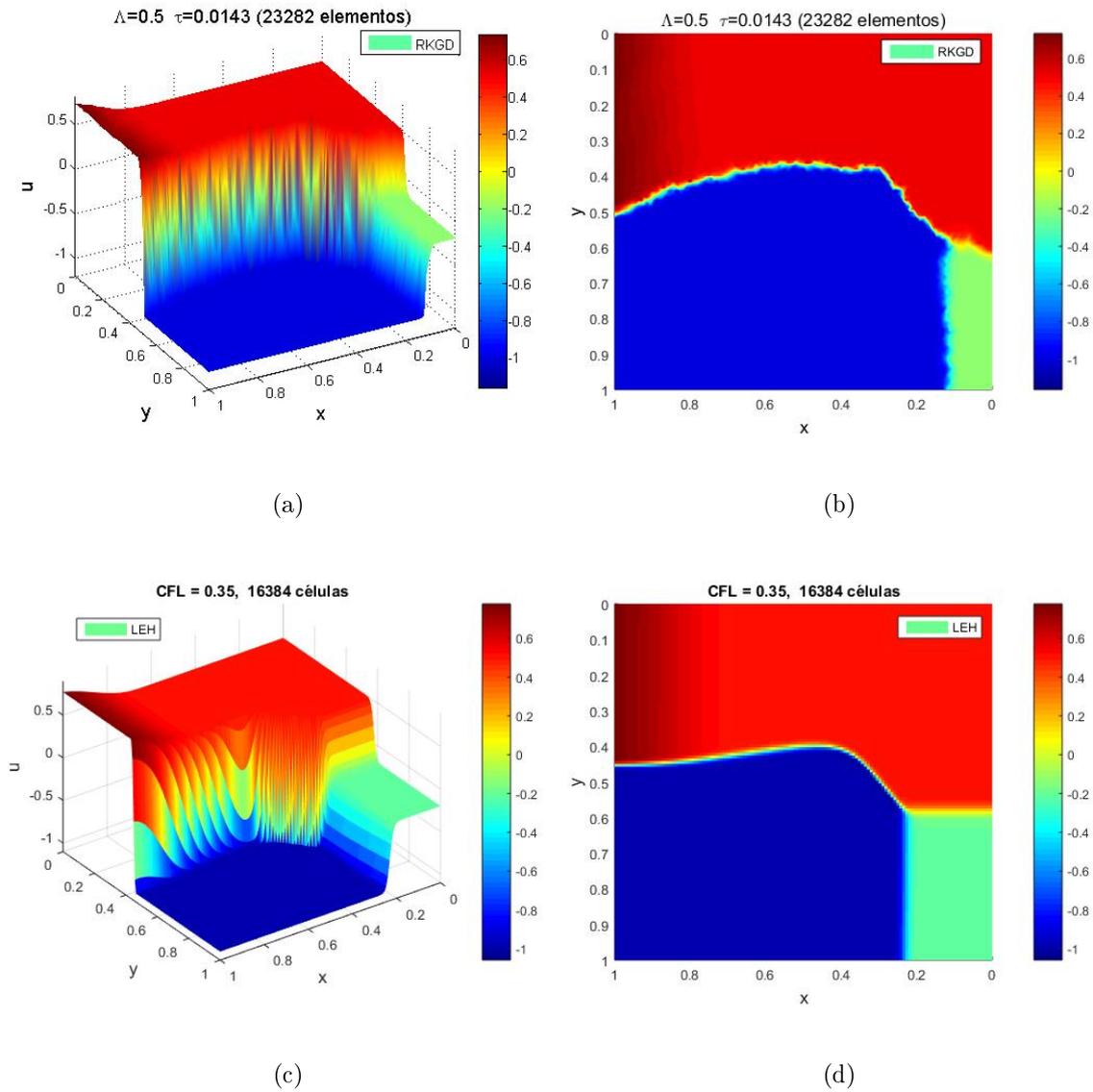


Figura 5.14: Resultado obtido para o PVI (5.5)–(5.7) para $t = 0.5$.

Problema 3. Equação de Advecção Linear

Consideramos o seguinte problema de valor inicial bidimensional associado à equação de Advecção Linear:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \quad (5.8)$$

com $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 0.5]$, e dado inicial:

$$u(x, y, 0) = \begin{cases} 1, & 0.1 \leq x \leq 0.6 \text{ e } 0.1 \leq y \leq 0.6 \\ 0, & \text{caso contrário.} \end{cases} \quad (5.9)$$

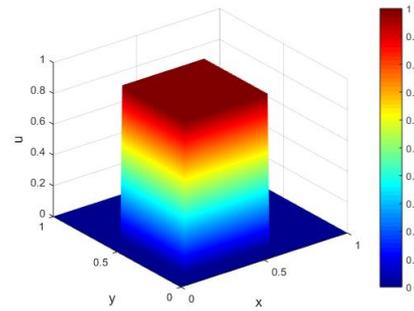


Figura 5.15: Condição Inicial.

No caso da equação Advecção Linear a solução exata é conhecida, veja a equação (2.36), então a solução desta é a condição inicial viajando na direção do vetor velocidade, que neste caso é o vetor $(1, 1)$, ou seja a bissetriz do plano (x, y) . Veja as soluções numéricas na Figura 5.16, elas correspondem com o esperado da teoria.

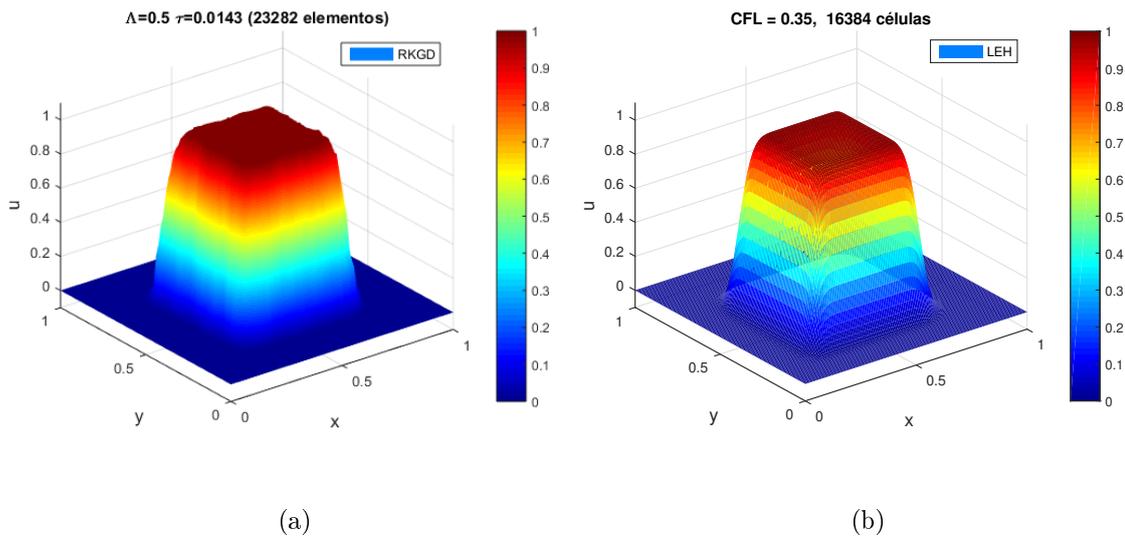
Figura 5.16: Resultado obtido para o PVI (5.8)–(5.9) para $t = 0.1$.

Tabela 5.1: Tempo Computacional para as Leis de Conservação.

Problema	Tempo Computacional (RKGD)	Tempo Computacional (LEH)
Problema 1	69.2 horas	1.2029 minutos
Problema 2	4.5 dias	2.3549 minutos
Problema 3	34.18 horas	0.9738 segundos

5.4 Equações de Shallow Water

Problema 1: Teste Propriedade “Well-Balanced”

Para verificar que o método implementado possui a propriedade “well-balanced” testamos se este é capaz de preservar o estado estacionário ($h + b = \text{cte}$ e $u = v = 0$). Este exemplo foi tomado de Xing [65].

Consideramos um domínio computacional retangular $[0, 1] \times [0, 1]$ e a seguinte função do fundo:

$$b(x, y) = \max(0, 1 - (10x - 5)^2 - (10y - 5)^2), \quad (5.10)$$

e a condição inicial é uma solução estacionária:

$$h(x, y, 0) = 2 - b(x, y), \quad hu(x, y, 0) = hv(x, y, 0) = 0.$$

Uma condição de contorno periódica foi usada. Para que o método possua a propriedade “well-balanced” o estado estacionário deve ser preservado e a superfície deve permanecer plana ($h + b = \text{cte}$), de forma que o erro obtido seja da ordem do erro de arredondamento. Calculamos a solução até $t = 0.5$, em uma malha com $\tau = 0.0242$ (8072 elementos) e utilizamos $\Lambda = 0.5$. Na tabela 5.2 é mostrado o erro ε_{L_1} obtido, onde podemos ver claramente que todos os erros estão no nível de erros de arredondamento, que verifica a propriedade “well-balanced”.

Tabela 5.2: Erro ε_{L_1} para o problema estacionário.

h	hu	hv
1.723E-14	5.443E-14	5.595E-14

Para calcular este erro para um tempo fixo utilizamos a seguinte equação

$$L_1(t) = \sum_{i=1}^{NE} \iint_{K_i} |U_{\text{exato}}^i - U_{\text{numérico}}^i| dx dy, \quad (5.11)$$

onde NE é o número de elementos. A integral acima é calculada como explicado na seção 3.4, ou seja utilizamos o pontos de quadratura S_K , veja a Figura 3.11, e a mesma fórmula utilizada para calcular as integrais duplas dentro do método RKDG.

Problema 2: Barragem Circular

- Caso 1 (Problema resolvido em [65]):

Consideramos um domínio computacional quadrado $[-100, 100] \times [-100, 100]$ com topografia de fundo plana (ou seja, $b = 0$). A barragem está localizada em $r = x^2 + y^2 = 60$ e a altura da água h é inicialmente definida como dez dentro da barragem e zero fora. Ambas as componentes de velocidade u e v são definidas inicialmente como zero. No instante $t = 0$, a parede circular que forma a barragem entra em colapso.

Em [65] os autores utilizam uma malha triangular com diâmetro $\tau = 1$ e polinômios quadráticos para obter a solução em $t = 1.75$. Gerando uma malha através do gerador Distmesh obtemos uma malha com 189000 elementos ($\tau = 0.9947$), veja a Figura 5.4.

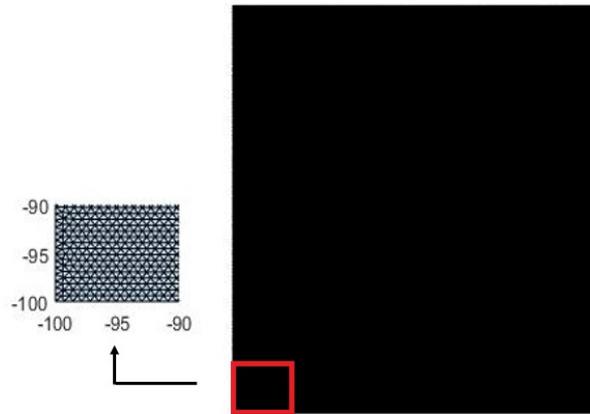


Figura 5.17: Malha com 189000 elementos, com $(x, y) \in [-100, 100] \times [-100, 100]$.

Infelizmente com os computadores disponíveis não foi possível obter a solução numérica para este problema devido ao número elevado de elementos, sendo assim utilizamos uma malha mais grossa, veja a Figura 5.6, a fim de se ter uma ideia do tempo computacional necessário.

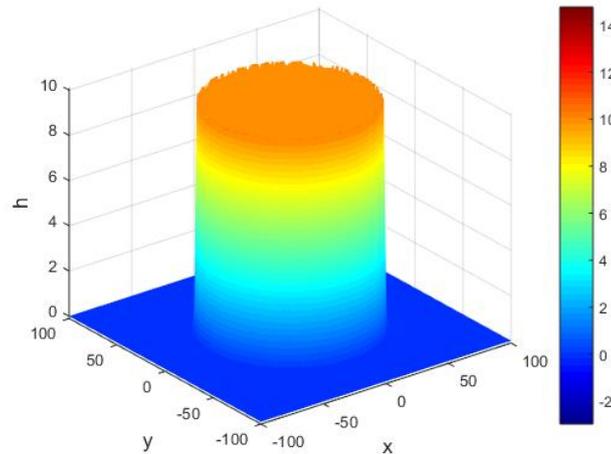


Figura 5.18: Resultado para o Caso 1 para $t = 4.8027e-04$, $\tau = 1.6942$ (64600 elementos) e $\Lambda = 0.01$.

Para obter o resultado do Caso 1, veja a Figura 5.18, foi realizada 33 iterações sendo necessário 13.78 horas. Todavia o primeiro Δt calculado é $8.9792e-04$ e os demais estão entre $1e-08$ e $1e-07$, portanto para obter a solução em $t = 1.75$ seria necessário meses.

- Caso 2 (Problema adaptado para diminuir o custo computacional):

Para obter a solução do problema acima é necessário um tempo computacional muito grande, afim de diminuir este tempo modificamos o problema de forma a obtermos um semelhante. Assim, consideramos um domínio computacional quadrado $[-10, 10] \times [-10, 10]$, e a barragem localizada em $r = x^2 + y^2 = 6$, supomos a altura da água h é inicialmente definida como 1.5 dentro da barragem e 0.5 fora, e $u = v = 0$. De novo consideramos topografia plana, $b = 0$.

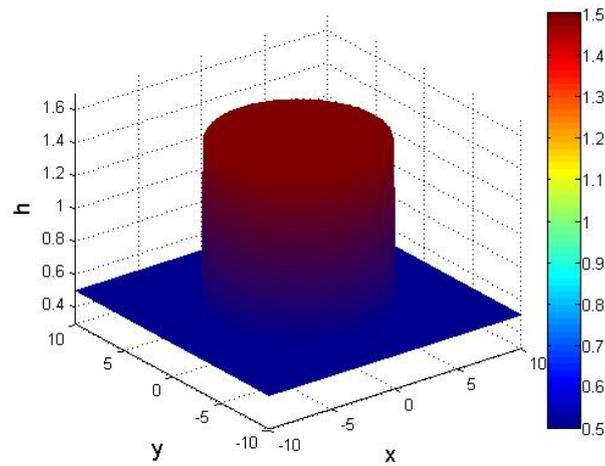
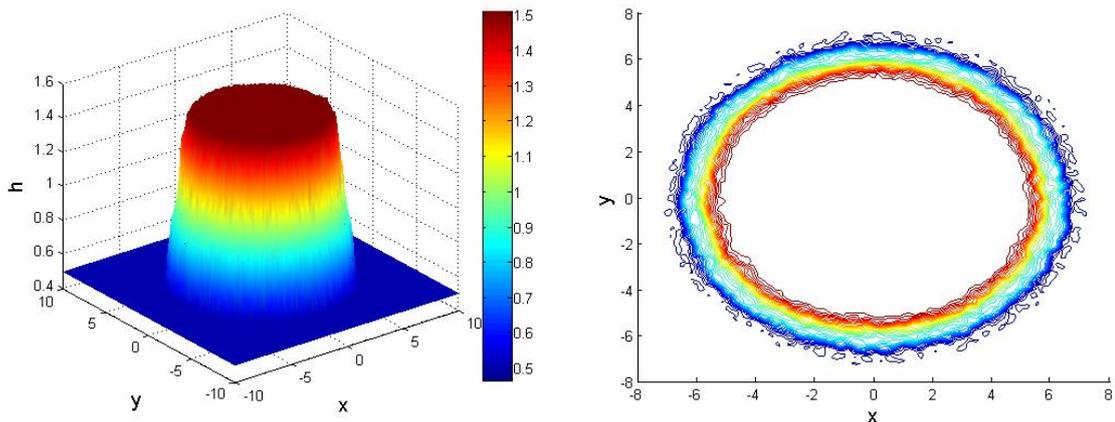


Figura 5.19: Condição Inicial.



(a) Solução em 3D

(b) Curvas de nível com 30 faixas de nível uniformemente espaçadas entre 0.5 e 1.5.

Figura 5.20: Resultado obtido para o Caso 2 para $t = 0.1$, $\tau = 0.2264$ (36354 elementos) e $\Lambda = 0.001$.

Podemos comparar de maneira qualitativa os resultados obtidos mostrados na Figura 5.20 com os obtidos pelo Xing em [65], de forma que podemos ver uma similaridade entre estes resultados e aqueles obtidos por Xing em [65] para o problema do caso 1. Veja como o parâmetro Λ do limitador TVB, embora escolhido muito pequeno $\Lambda = 0.001$, ainda aparecem oscilações espúrias na solução, isto é visível sobretudo nas curvas de nível mostradas na Figura 5.20(b).

A seguir será apresentada uma tabela mostrando o tempo computacional necessário

para obter as soluções numéricas para a equação de Shallow Water, utilizando o nosso código.

Tabela 5.3: Tempo Computacional para as Equações Shallow Water.

Problema	Tempo Computacional
Problema 1	5 dias
Problema 2 (Caso 1)	13.78 horas
Problema 2 (Caso 2)	14.4823 horas

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

No presente trabalho foi estudado e implementado o método de elementos finitos Galerkin Descontínuo utilizado para obter soluções numéricas para problemas estritamente hiperbólicos, leis de conservação lineares e não lineares em 2D, assim como o sistema de leis de balanço shallow water também em 2D.

Para utilizar o método RKDG a aplicação de dois limitadores é fundamental, visto que o método gera oscilações espúrias e este deve satisfazer o princípio do máximo, veja o Teorema 3.1. No caso do limitador TVB foi preciso acrescentar um tratamento especial quando este é aplicado nos elementos do contorno, pois os artigos utilizados como referência não especificam como tratar este caso particular. O tratamento adotado mostrou-se eficiente, como pode ser visto nos testes realizados.

Dentre as dificuldades numéricas encontradas no trabalho temos que este método necessita de uma malha com um número elevado de elementos para obter uma solução boa ou aceitável, dependendo do diâmetro da malha muitas oscilações espúrias são geradas e o limitador TVB pode não ser capaz de retirar todas, de forma que estas se propaguem ao longo do tempo levando a uma solução incorreta. A escolha do parâmetro Λ não é trivial, então testes numéricos foram necessários para escolher de forma conveniente este parâmetro, pois este é de fundamental importância na retirada das oscilações espúrias. Quando escolhido de maneira correta, este mostrou-se suficiente para malhas de diferentes diâmetros, não sendo então necessária a sua alteração juntamente com a mudança do diâmetro da malha.

Informações sobre o valor de U_{ext} (3.17) quando o elemento pertence ao contorno não

foram encontradas na literatura, visto que geralmente quando estamos trabalhando com problemas estritamente hiperbólicos condições de contorno não são sempre necessárias. Se o valor de U_{ext} no contorno for atribuído de maneira diferente aos exemplos dos artigos referentes, isto pode acarretar um erro que se propaga para dentro do domínio. Dessa forma, em alguns casos não foi possível reproduzir a solução numérica como em artigos da bibliografia, em especial Xing [65], visto que a aplicação do método RKDG no contorno em [65] não é trivial e nem clara para alguns exemplos.

Em termos gerais o método apresentou bom comportamento convergindo para as soluções entrópicas, mostrando assim ser capaz de capturar as ondas de rarefações e ondas de choque, assim como outros tipos de ondas para condições gerais de Cauchy. Porém devido ao fato deste método necessitar de malhas com um número elevado de elementos, isto gerou um grande custo computacional e a impossibilidade prática de reproduzir alguns resultados obtidos em [65] para a equação shallow water.

6.2 Trabalhos Futuros

Por fim propõe-se o estudo dos seguintes tópicos para dar continuidade ao presente trabalho:

- Finalizar o método RKGD para polinômios quadráticos;
- Diminuir o tempo computacional, vetorizando, paralelizando ou mudando a linguagem do código;
- Obter a solução numérica da lei de conservação que modela o escoamento bifásico em um meio poroso em 2D com termo fonte através do método RKGD.

Referências

- [1] DistMesh Gallery. <https://popersson.github.io/distmesh/>. Acessado em 16 de janeiro de 2020.
- [2] NPTEL-Finite Element Analysis (Web). <https://nptel.ac.in/courses/105105041/>. Acessado em 16 de janeiro de 2020.
- [3] SWE. <https://www5.in.tum.de/SWE/doxy/index.html>. Acessado em 16 de janeiro de 2020.
- [4] New non-oscillatory central schemes on unstructured triangulations for hyperbolic systems of conservation laws. *Journal of Computational Physics* 227, 11 (2008), 5736 – 5757.
- [5] AUDUSSE, E., BOUCHUT, F., BRISTEAU, M.-O., KLEIN, R., PERTHAME, B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *Siam Journal on Scientific Computing* 25 (2004), 2050–2065.
- [6] BASSI, F., REBAY, S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.* 138 (1997), 251–285.
- [7] BATEMAN, H. Some recent researches on the motion of fluids. *Monthly Weather Review* 43, 4 (1915), 163–170.
- [8] BATHE, K.-J. *Finite Element Procedures*. Prentice Hall, 1996.
- [9] BAUMANN, C., ODEN, J. A discontinuous hp finite element method for the solution of the euler equation of gas dynamics. *In 10th International Conference on Finite Element in Fluids* (1998).
- [10] BECKER, E. B., CAREY, G. F., ODEN, J. T. *Finite Elements an Introduction*, vol. 1. Prentice Hall, 1981.
- [11] BEY, K. S., ODEN., J. A Runge-Kutta discontinuous Galerkin finite element method for high speed flows. *In 10th AIAA Computational Fluid Dynamics Conference* (June 1991), 24–27.
- [12] BURGERS, J. A mathematical model illustrating the theory of turbulence. *Advances in applied mechanics* 1 (1948), 171–199.
- [13] CALEFFI, V., VALIANI, A., BERNINI, A. Fourth-order balanced source term treatment in central WENO schemes for shallow water equations. *J. Comput. Phys.* 218 (2006), 228–245.

-
- [14] CANESTRELLI, A., SIVIGLIA, A., DUMBSER, M., TORO, E. F. Well-balanced high-order centred schemes for non-conservative hyperbolic systems. applications to shallow water equations with fixed and mobile bed. *Adv. in Water Resour.* 32 (2009), 834–844.
- [15] CASTRO, M., GALLARDO, J., MADROÑAL, C. High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. applications to shallow-water systems. *Math. Comput.* 75 (2006), 1103–1134.
- [16] CASTRO, M., PARÉS, C. On the well-balance property of Roe’s method for nonconservative hyperbolic systems. applications to shallow-water systems. *Mathematical Modelling and Numerical Analysis* 38 (2004), 821–852.
- [17] CHAVENT, G., COCKBURN, B. The local projection P^0 P^1 -discontinuous- Galerkin finite element method for scalar conservation laws. *RAIRO Model Math. Anal. Numer.* 23 (1989), 565–592.
- [18] CHAVENT, G., SALZANO, G. A finite element method for the 1D water flooding problem with gravity. *J. Comput. Phys.* 45 (1982), 307–344.
- [19] CIARLET, P. G., LIONS, J. L. *Finite Element Methods (Part 1)*, vol. 2. Elsevier Science, 1990.
- [20] COCKBURN, B., HOU, S., SHU, C. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.* 54 (1990), 545–581.
- [21] COCKBURN, B., KARNIADAKIS, G., SHU, C.-W. *Discontinuous Galerkin Methods: Theory, Computation and Application*. 2000.
- [22] COCKBURN, B., KARNIADAKIS, G. E., SHU, C.-W. *Discontinuous Galerkin Methods: Theory, Computation and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2000.
- [23] COCKBURN, B., LIN, S., SHU, C. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. *J. Comput. Phys.* 84 (1989), 90–113.
- [24] COCKBURN, B., SHU, C. The P^1 -RKDG method for two-dimensional Euler equations of gas dynamics. *Technical Report ICASE* (1991), 91–32.
- [25] COCKBURN, B., SHU, C. The Runge-Kutta local projection P^1 - discontinuous Galerkin method for scalar conservation laws. *RAIRO Model Math. Anal. Numer.* 25 (1991), 337–361.
- [26] COCKBURN, B., SHU, C.-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math. Comp.* 52 (1989), 411–435.
- [27] COCKBURN, B., SHU, C.-W. The Runge-Kutta discontinuos Galerkin method for conservation laws V: multidimensiona systems. *J. Comput. Phys.* 141 (1998), 199–224.

- [28] DE LIMA, P. C. Equações diferenciais parciais i. <http://www.mat.ufmg.br/lima/apostilas/EDP1-aulas-22-8-13.pdf>, 2013. Acessado em 08 de setembro de 2017.
- [29] FRACCAROLLO, L., CAPART, H. Riemann wave description of erosional dam-break flows. *Journal of Fluid Mechanics* 461 (2002), 1–46.
- [30] GEORGE, D., M. IVERSON, R. A depth-averaged debris-flow model that includes the effects of evolving dilatancy. ii. numerical predictions and experimental tests. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 470 (2014), 1–31.
- [31] GRAY, J., WIELAND, M., HUTTER, C. Gravity-driven free surface flow of granular avalanches over complex basal topography. *Royal Society of London Proceedings Series A* 455 (1999), 1841–1874.
- [32] HOLTON, J. R. *An Introduction to Dynamic Meteorology*. Elsevier Academic Press, 2004.
- [33] JOHNSON, C., PITKARANTA, J. An analysis of the discontinuous galerkin method for a scalar hyperbolic equation. *Math. Comp.* 46, 1 (1986).
- [34] KRÖNER, D. *Numerical Schemes for Conservation Laws*. Wiley-Teubner Series, Advances in Numerical Mathematics, 1997.
- [35] KRUKOV, S. N. First order quasilinear equations in several independent variables. *Mat. Sbornik* 81 (1970), 217–243.
- [36] L. HIGDON, R. Numerical modelling of ocean circulation. *Acta Numerica* 15 (2006), 385–470.
- [37] LANZONI, S., SIVIGLIA, A., FRASCATI, A., SEMINARA, G. Long waves in erodible channels and morphodynamic influence. *Water Resources Research - WATER RESOUR RES* 42 (2006).
- [38] LESAIN, P., RAVIART, P. On a finite element method for solving the neutron transport equation. In *Mathematical Aspects of Finite Elements in Partial Differential Equations*, C. de Boor, Ed. Academic Press, 1974, p. 89 – 123.
- [39] LEVEQUE, R. J. *Numerical Methods for Conservation Laws*, second ed. Birkhauser, 1992.
- [40] LUKACOVA-MEDVIDOVA, M., NOELLE, S., KRAFT, M. Well-balanced finite volume evolution galerkin methods for the shallow water equations. *Journal of Computational Physics* 221 (2007), 122–147.
- [41] MUÑOZ, N. L. R. Tasas de convergencia en el esquema de Glimm. Dissertação de Mestrado, Universidad Pedagógica y Tecnológica de Colombia, 2019.
- [42] NOELLE, S., PANKRATZ, N., PUPPO, G., NATVIG, J. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *Journal of Computational Physics* 213 (2006), 474–499.

- [43] NOELLE, S., XING, Y., SHU, C.-W. High-order well-balanced finite volume weno schemes for shallow water equation with moving water. *Journal of Computational Physics* 226 (2007), 29–58.
- [44] PARÉS, C. Numerical methods for nonconservative hyperbolic systems: A theoretical framework. *SIAM J. Numerical Analysis* 44 (2006), 300–321.
- [45] PEREZ, J. A. *Lagrangian-Eulerian approximation methods for balance laws and hyperbolic conservation laws*. Ph.d. thesis, 2015.
- [46] PERSSON, P.-O. *Mesh Generation for Implicit Geometries*. Tese de doutorado, MIT, 2005.
- [47] PETERSON, T. A note on the convergence of the discontinuous Galerkin method for a scalae hyperbolic equation. *SIAM J. Numer. Anal.* 28, 133 (1991).
- [48] REED, W., HILL, T. Triangular mesh methods for the neutron transport equation. *Proceedings of the American Nuclear Society* (1973).
- [49] RICHTER, G. R. An optimal-order error estimate for the discontinuous Galerkin method. *Math. Comp.* 50, 75 (1988).
- [50] RODRÍGUEZ-BERMÚDEZ, P. *Buoyancy Driven Three-Phase Flow in Porous Media*. PhD thesis, IMPA, 2010.
- [51] ROSATTI, G., BEGNUDELLI, L. Two-dimensional simulation of debris flows over mobile bed: Enhancing the TRENT2D model by using a well-balanced generalized Roe-type solver. *Computers and Fluids* 71 (2013), 179–195.
- [52] SAINT-VENANT, A. Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l’introduction des marées dans leur lit. *Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences, Paris* 73 (01 1871).
- [53] SHU, C.-W., OSHER, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* 77 (1988), 439–471.
- [54] SHU, C.-W., OSHER, S. Efficient implementation of essentially non-oscillatory shock capturing schemes, ii. *J. Comput. Phys.* 83 (1989), 32–78.
- [55] T.C. WARBURTON, I. LOMTEV, R. K., KARNIADAKIS, G. A discontinuous Galerkin method for the Navier-Stokes equations in hybrid grids. *In 10th International Conference on Finite Element in Fluids* (1998).
- [56] VAN DUIJN, C. J. *An Introduction to Conservations Laws: Theory and Applications to Multi-Phase Flow*. Lecture Notes, Delft University of Technology, 2002.
- [57] VAN LEER, B. Towards the ultimate conservative difference scheme, ii. *Journal of Computational Physics* 14, 4 (1974), 361–376.
- [58] VREUGDENHIL, C. B. *Numerical Methods for Shallow Water Flow*. Kluwer Academic Publishers, 1994.

- [59] VUKOVIC, S., SOPTA, L. ENO and WENO schemes with the exact conservation property for one-dimensional shallow water equations. *Journal of Computational Physics* 179 (2002), 593–621.
- [60] XING, Y., SHU, C.-W. High order finite difference WENO schemes with the exact conservation property for the shallow water equations. *Journal of Computational Physics* 208 (2005), 206–227.
- [61] XING, Y., SHU, C.-W. High-order well-balanced finite difference WENO schemes for a class of hyperbolic systems with source terms. *J. Sci. Comput.* 27 (2006), 477–494.
- [62] XING, Y., SHU, C.-W. High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *Journal of Computational Physics* 214 (2006), 567–598.
- [63] XING, Y., SHU, C.-W. A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *Communications in Computational Physics* 1 (2006), 100–134.
- [64] XING, Y., SHU, C.-W., NOELLE, S. On the advantage of well-balanced schemes for moving-water equilibria of the shallow water equations. *Journal of Scientific Computing* 48 (2010), 339–349.
- [65] XING, Y., ZHANG, X. Positivity-preserving well-blanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes. *J. Sci. Comput.* 57 (2013), 19–41.
- [66] XING, Y., ZHANG, X., SHU, C.-W. Positivity-preserving high order well-balanced discontinuous galerkin methods for the shallow water equations. *Adv. Water Resources* 33 (2010), 1476–1493.
- [67] YAMASHITA, W. M. S. Introdução as leis de conservação e aplicações. Monografia (Bacharel em Matemática), Universidade Federal de Juiz de Fora, 2014.
- [68] ZHANG, X., XIA, Y., SHU, C.-W. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing* 50 (2012), 29–62.
- [69] ZOLEZZI, G., SEMINARA, G. Downstream and upstream influence in river meandering. part 1. general theory and application to overdeepening. *Journal of Fluid Mechanics* 438 (2001), 183–211.

APÊNDICE A

Seja f uma função definida em um domínio D mensurável

- Se $p \in [1, \infty)$, f é dita p integrável e pertence ao espaço L^p se sua norma L^p for finita

$$\|f\|_{L^p} = \left(\int_D |f(x)^p dx \right)^{1/p}$$

- Se $p = \infty$, f é dita essencialmente limitada e pertence ao espaço L^∞ se existir uma constante C real tal que

$$\mu \{x \in D : |f| > C\} = 0$$

ou seja, $|f| \leq C$ exceto em conjunto de medida zero.

A norma $\|f\|_{L^\infty(D)}$ é a menor das constantes com a propriedade acima, ou seja:

$$\|f\|_{L^\infty(D)} = \inf \{C : \mu \{|f(x)| > C\} = 0\}$$

APÊNDICE B

Seja $f : D \rightarrow \mathbb{R}$, dizemos que $f \in L^1_{\text{loc}}(E)$ se E é um subconjunto mensurável de E e vale que:

- $\forall V \subseteq \bar{V} \subseteq E$ com \bar{V} compacto então $f \in L^1(V)$

Esta definição pode ser generalizada para os espaços $L^p_{\text{loc}}(V)$.