#### UNIVERSIDADE ESTADUAL DE CAMPINAS INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA

#### **DOUGLAS AZEVEDO CASTRO**

### ESQUEMAS DE APROXIMAÇÃO EM MULTINÍVEL E APLICAÇÕES

TESE DE DOUTORADO APRESENTADA AO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA DA UNICAMP PARA OBTENÇÃO DO TÍTULO DE DOUTOR EM MATEMÁTICA APLICADA

#### ORIENTADORA: SÔNIA MARIA GOMES COORIENTADOR: JORGE STOLFI

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO DOUGLAS AZÉVEDO CASTRO E ORIENTADA PELOS PROFs. DRs.

MY ônia Maria-Gomes

1 hour non

Jorge Stolfi

CAMPINAS, 2011

#### FICHA CATALOGRÁFICA ELABORADA POR ANA REGINA MACHADO – CRB8/5467 BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA – UNICAMP

Castro, Douglas Azevedo, 1982-C279e Esquemas de aproximação em multinível e aplicações / Douglas Azevedo Castro. - Campinas, SP : [s.n.], 2011. Orientador: Sônia Maria Gomes. Coorientador: | orge Stolfi. Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica. 1. Equações diferenciais - Soluções numéricas. 2. Método dos volumes finitos. 3. Wavelets (Matemática). 4. Estruturas de dados (Computação). 5. Spline, Teoria do. I. Gomes, Sônia Maria, 1952-. II. Stolfi, Jorge, 1950-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em inglês: Multilevel approximation schemes and applications Palavras-chave em inglês: Differential equations - Numerical solutions Finite volume method Wavelets (Mathematics) Data structures (Computer science) Spline theory Área de concentração: Matemática Aplicada Titulação: Doutor em Matemática Aplicada Banca examinadora: Sônia Maria Gomes [Orientador] Maria Cristina de Castro Cunha Helio Pedrini Margarete Oliveira Domingues Helio Côrtes Vieira Lopes Data da defesa: 12-12-2011 Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 12 de dezembro de 2011 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

ms

Prof(a). Dr(a). SÔNIA MARIA GOMES

an

Prof(a). Dr(a). MARIA CRISTINA DE CASTRO CUNHA

Prof(a). Dr(a). HELIO PEDRINI

Prof(a). Dr(a). MARGARETE OLIVEIRA DOMINGUES

Helio Config Vienc forz Prof(a). Dr(a). HELIO CÔRTES VIEIRA LOPES

A meus pais e irmãos.

# Agradecimentos

À profa. Dra. Sônia Maria Gomes e ao prof. Dr. Jorge Stolfi, meus orientadores, que investiram muito do seu tempo em mim e deram valiosas instruções durante as orientações.

À profa. Dra. Anamaria Gomide que, diversas vezes e de várias formas, tanto me ajudou.

Aos professores que participaram da banca de defesa, por suas valiosas dicas.

Ao prof. Dr. Jesus Carlos da Mota, orientador de mestrado e iniciação científica, que me ajudou, em especial, com o despertar do interesse pela pesquisa. Em seu nome agradeço a todos os professores e funcionários do IME/UFG.

À toda minha família, essa batalha vencida também é sua culpa.

Ao Alberto Caiero, espero que não se importe em me emprestrar algumas de suas palavras.

À "última estrela(...), pouso meus olhos em seu trêmulo azular branco" e consigo reanimar o desejo de lutar, ta.

Aos meus irmãos, Danubio e Dennes, por me ajudar com a manipulação de imagens. De verdade, vocês me ajudaram muito, tanto diretamente quanto por meio de zombarias quando iam passear e eu tinha que ficar estudando. Valeu por também manipular aquelas imagens, sabe, aquelas montagens nas fotos de família, me incluindo nas fotografias.

Aos meus pais, a quem devo toda minha conquista, só esse casal sabe o quanto de céu e de terra moveram para ajudar e motivar os seus três rebentos a vencerem. Nós nem imaginamos quantas vezes e de que abriram mão pela gente. "Ô seu" Jeová, "ô dona" Nilza, minha eterna gratidão ainda é insuficiente. Não há um só dia em que não reconheça o que fizeram, o que fazem e o que ainda farão por todos nós.

Ao IMECC, pelo infraestrutura e apoio na divulgação de nosso trabalho. Ao pessoal da secretaria de pós graduação, Cidinha, Tânia, Ednaldo, Lívia pela dedicação em seu trabalho.

Ao CNPq e à FAPESP pelo apoio financeiro.

Aos amigos com quem dividi uma casa em Campinas, Anderson, Alison, Edcarlos, Henrique pelos bons tempos.

Aos outros amigos que fiz em Campinas, Daniela, Denise, Gilcélia, Juliana, Leidy, João, pelo elo fraternal.

Enfim, a todos os meus professores, que nesses 25 anos de instrução contribuiram para minha conquista. No início do ginásio um deles contou à turma sobre o mais alto grau de instrução que se podia chegar. Nem precisa mencionar o quão fascinado fiquei e a partir dali teria mais uma meta em minha vida. Consegui! "O tempo é apenas o rio em que vou pescando. Bebo nele, mas ao beber vejo-lhe o leito de areia e percebo quão raso é. A fina corrente logo se esvai, mas a eternidade permanece. Gostaria de beber mais fundo e de pescar no céu, em cujo leito os seixos são estrelas. Não consigo contá-las. Ignoro a primeira letra do alfabeto. Tenho lamentado sempre não ser sábio como no dia em que nasci. A inteligência é um cutelo que penetra e corta caminho adentro o segredo das coisas. Não desejo ocupar minhas mãos mais do que o necessário. Minha cabeça é mãos e pés. Sinto que minhas melhores faculdades aí se concentram. O instinto me diz que a cabeça é um órgão para escavação, feito o focinho e as patas de certos bichos, e com a qual gostaria de explorar e cavar meu caminho através desses morros. Penso que o filão mais rico está por aí nas redondezas, e assim avalio por meio de varinha de condão e dos finos eflúvios que se levantam. Aqui começarei a minerar."

H. D. Thoreau em Walden ou A vida nos bosques.

## Resumo

O objetivo desta tese é desenvolver algoritmos baseados em malhas e bases funcionais inovadoras usando técnicas de multiescala para aproximação de funções e resolução de problemas de equações diferenciais. Para certas classes de problemas, é possível incrementar a eficiência dos algoritmos de multiescala usando bases adaptativas, associadas a malhas construídas de forma a se ajustarem com o fenômeno a ser modelado. Nesta abordagem, em cada nível da hierarquia, os detalhes entre a aproximação desse nível e a aproximação definida no próximo nível menos refinado pode ser usada como indicador de regiões que necessitam de mais ou menos refinamento. Desta forma, em regiões onde a solução é suave, basta utilizar os elementos dos níveis menos refinados da hierarquia, enquanto que o maior refinamento é feito apenas onde a solução tiver variações bruscas.

Consideramos dois tipos de formulações para representações multiescala, dependendo das bases adotadas: splines diádicos e *wavelets*. A primeira abordagem considera espaços aproximantes por funções splines sobre uma hierarquia de malhas cuja resolução depende do nível. A outra abordagem considera ferramentas da análise *wavelet* para representações em multirresolução de médias celulares. O enfoque está no desenvolvimento de algoritmos baseados em dados amostrais *d*-dimensionais em malhas diádicas que são armazenados em uma estrutura de árvore binária. A adaptatividade ocorre quando o refinamento é interrompido em algumas regiões do domínio, onde os detalhes entre dois níveis consecutivos são suficientemente pequenos. Um importante aspecto deste tipo de representação é que a mesma estrutura de dados é usada em qualquer dimensão, além de facilitar o acesso aos dados nela armezenados. Utilizamos as técnicas desenvolvidas na construção de um método adaptativo de volumes finitos em malhas diádicas para a solução de problemas diferenciais. Analisamos o desempenho do método adaptativo em termos da compressão de memória e tempo de CPU em comparação com os resultados do esquema de referência em malha uniforme no nível mais refinado. Neste sentido, comprovamos a eficiência do método adaptativo, que foi avaliada levando-se em consideração os efeitos da escolha de diferentes tipos de fluxo numérico e dos parâmetros de truncamento.

# Abstract

The goal of this thesis is to develop algorithms based on innovative meshes and functional bases using multiscale techniques for function approximation and solution of differential equation problems. For certain classes of problems, one can increase the efficiency of multiscale algorithms using hierarchical adaptive bases, associated to meshes whose resolution varies according to the local features of the phenomenon to be modeled. In this approach, at each level of the hierarchy the details—differences between the approximation for that level and that of the next coarser level—can be used as indicators of regions that need more or less refinement. In this way, in regions where the solution is smooth, it suffices to use elements of the less refined levels of the hierarchy, while the maximum refinement is used only where the solution has sharp variations.

We consider two classes of formulations for multiscale representations, depending on the bases used: dyadic splines and wavelets. The first approach uses approximation spaces consisting of spline functions defined over a mesh hierarchy whose resolution depends on the level. The other approach uses tools from wavelet analysis for multiresolu-tion representations of cell averages. The focus is on the development of algorithms based on sampled d-dimensional data on dyadic meshes which are stored in a binary tree structures. The adaptivity happens when the refinement is interrupted in certain regions of the domain, where the details between two consecutive levels are sufficiently small. This representation greatly simplifies the access to the data and it can be used in any dimension.

We use these techniques to build an adaptive finite volume method on dyadic grids for the solution of differential problems. We analyze the performance of the method in terms of memory compression and CPU time, comparing it with the reference scheme (which uses a uniform mesh at the maximum refinement level). In these tests, we confirmed the efficiency of the adaptive method for various numeric flow formulas and various choices of the thresholding parameters.

# Sumário

A	gradecimentos	vii
Re	esumo	xi
$\mathbf{A}$	bstract x	iii
Su	ımário	xv
1	Introdução	1
<b>2</b>	Conceitos Básicos em Teoria de Aproximação	9
	2.1 Aspectos Gerais	9
	2.1.1 Critérios de Aproximação	10
	2.1.2 Como Calcular a Aproximação $\hat{u} = Tu?$	11
	2.1.3 Erro de Aproximação	12
	2.2 Exemplos	13
	2.2.1 Spiines Lineares	13
	2.2.2 <i>Wavelets</i> de Haar	23 32
		02
3	Malhas Diádicas e sua Representação	33
	3.1 Conceitos Básicos	33
	3.1.1 Subdivisões $\ldots$	33
	3.1.2 Caixas $\ldots$	34
	3.1.3 Árvores Binárias	35
	3.2 Malhas Diádicas	37
	3.2.1 Construção da Malha	37
	3.3 Malhas $2^d$ -ádicas	44
4	Splines Diádicos	45
	4.1 Splines em Malhas Diádicas Regulares	45
	4.2 Bases para Splines	46

		4.2.1	Base Uniforme de Tendas	48			
		4.2.2	Base Hierárquica de Tendas	49			
		4.2.3	Adaptatividade	50			
		4.2.4	Algoritmo de Interpolação Usando Base de Tendas	51			
	4.3	Exem	plo	52			
<b>5</b>	Ana	álise M	IR para Médias Celulares em Malhas Diádicas	55			
	5.1	Defini	ções e Notações Preliminares	56			
	5.2	Algori	tmos de MR em Malhas Diádicas	58			
	5.3	Previs	ão de Ordem Superior	59			
		5.3.1	Previsão Exata para Polinônimos de Grau $\leq 2$	59			
		5.3.2	Previsão Exata de Ordem Superior em Malhas Diádicas d-dimensional	.s 66			
6	Análise MR para Médias Celulares em Malhas Diádicas: Aspecto Fun-						
	cior	ıal		69			
	6.1	Caso	Unidimensional	70			
	6.2	Caso I	Bidimensional	73			
	6.3	Caso a	d-dimensional	777			
7	Representações MR Adaptativas 81						
	7.1 Operador de Truncamento						
	7.2	Estim	ativa de Erro	82			
	7.3	Exem		83			
		7.3.1	Comparação de Esquemas MR em Malhas Diádicas e <i>Quad-grids</i> .	83			
		7.3.2	Análise MR em Compressao de Imagem	90			
8	Esq	uema	Adaptativo de Volumes Finitos para Malhas Diádicas	93			
	8.1	Discre	tização por Volumes Finitos	94			
		8.1.1	Integração no Tempo	95			
		8.1.2	Fluxo Numérico	96			
		8.1.3	Fluxo em Dimensões Superiores	100			
	8.2	Adapt	atividade e o Esquema de Volumes Finitos	101			
		8.2.1	Estrutura de Dados	103			
	8.3	Imple	mentação do Algoritmo	107			
		8.3.1	Exemplos Unidimensionais	108			
9	Esquema Adaptativo de Volumes Finitos para Malhas Diádicas: Resul-						
	tad	os Nur	néricos	115			
	9.1	Esque	$\max FV/MR-Roe + ENO + RK2 \dots \dots$	116			
		9.1.1	Equação de Advecção	116			
		9.1.2	Equação de Burgers $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	119			
		9.1.3	Equação de Buckley-Leverett	121			

9.2	Esque	ma $FV/MR$ -WENO+RK3	. 130		
	9.2.1	Equação de Advecção	. 130		
	9.2.2	Equação de Burgers	. 133		
	9.2.3	Equação de Buckley-Leverett	. 139		
9.2.1       Equação de Advecção       130         9.2.2       Equação de Burgers       133         9.2.3       Equação de Buckley-Leverett       139         10 Conclusões e Trabalhos Futuros       145         Referências Bibliográficas       148					
Referências Bibliográficas					

# Capítulo 1

# Introdução

O objetivo desta tese é desenvolver algoritmos baseados em malhas e bases funcionais inovadoras usando técnicas de *multiescala* para aproximação de funções e resolução de problemas de equações diferenciais.

Em muitos problemas, técnicas de multiescala permitem obter algoritmos mais confiáveis e/ou eficientes, em comparação com aqueles que trabalham em uma única escala. Na abordagem em multiescala, a função de interesse é representada por uma *hierarquia* de bases funcionais, em escalas ordenadas em progressão geométrica. Tipicamente, a construção dessas bases de multiescala se baseia em uma hierarquia de malhas geométricas do domínio computacional. Nesse contexto, várias estratégias podem ser introduzidas para melhorar a eficiência de algoritmos de aproximação. Por exemplo, a resolução de problemas pode ser feita em cada nível da hierarquia, da versão mais reduzida para a mais detalhada, usando a solução do nível anterior (mais grosseiro) como estimativa inicial para resolver o problema no nível presente (mais detalhado).

Para certas classes de problemas, é possível incrementar a eficiência dos algoritmos em multiescala usando bases *adaptativas*, associadas a malhas construídas de forma a se ajustarem com o fenômeno a ser modelado. Nesta abordagem, em cada nível da hierarquia, a diferença entre a aproximação desse nível e a aproximação definida no próximo nível menos refinado pode ser usada como indicador de regiões que necessitam de mais ou menos refinamento. Desta forma, em regiões onde a solução é relativamente suave poderiam estar presentes os elementos dos níveis menos refinados da hierarquia, enquanto que o maior refinamento ocorreria apenas onde a solução tiver variações bruscas. Espera-se que, com adaptatividade, o volume total de informações a ser armazenado e processado seja reduzido, diminuindo, em consequência, o tempo de CPU.

Como os métodos de multiescala podem fornecer representações de dados que são esparsas, complexas e hierárquicas, dependendo da estrutura de dados utilizada para sua codificação, o desempenho dos resultados pode ser comprometido. Sendo assim, o desenvolvimento de algoritmos em multiescala exige uma ampla gama de técnicas, não somente em teoria de aproximação, mas também em computação científica.

Apesar de significantes avanços já terem sido obtidos nessas linhas de pesquisa, é comum as técnicas serem desenvolvidas e empregadas em aplicações de áreas específicas, com contribuições disjuntas entre elas. Nesse sentido, o enfoque do presente trabalho está no desenvolvimento de novas ferramentas que visem atender às necessidades das técnicas de multiescala, com especial atenção na estrutura de dados adotada, com o objetivo de obter algoritmos computacionais adaptativos eficientes.

No presente trabalho consideramos dois tipos de formulações para representações em multiescala, dependendo das bases adotadas: splines diádicos e *wavelets*.

A primeira abordagem considera espaços aproximantes por funções splines sobre uma hierarquia de malhas cuja resolução depende do nível. Esta abordagem pode ser usada com malhas regulares ou irregulares de geometrias diversas, resultando em uma grande variedade de representações, muitas das quais ainda inexploradas. Especificamente, pretendemos considerar splines lineares em *malhas diádicas*, usadas com sucesso por Cardoso *et al.* [5, 6].

Uma malha diádica é uma hierarquia de malhas obtida, a partir de uma caixa ddimensional, por bissecções múltiplas, sendo que, no nível  $\ell$  da hierarquia, todos os planos de corte são perpendiculares ao eixo ( $\ell \mod d$ ) (o resto da divisão inteira de  $\ell$  por d). A principal vantagem de splines em malhas diádicas é sua simplicidade topológica, que reduz bastante a complexidade dos algoritmos em muitas aplicações, especialmente as que exigem malhas adaptativas e dinâmicas. Um exemplo de representação em malha diádica adaptativa é mostrado na Figura 1.1, em que há um maior refinamento da malha numa certa região do domínio para que seja possível captar com mais precisão as variações de uma certa função.

	K		
	Y	Ð	
		-	

Figura 1.1: Malha diádica adaptativa.

A codificação das informações associadas às representações em malhas diádicas é feita em uma estrutura de dados de árvore binária [31]. Uma árvore binária é um conjunto T de elementos denominados nós, estruturado da seguinte forma: ou o conjunto T é vazio (árvore vazia); ou existe um nó especial r, chamado raiz de T e os demais nós estão divididos em duas árvores binárias  $T_e \in T_d$ , respectivamente, a subárvore esquerda e subárvore direita de r. A cada divisão dos nós, criamos um novo nível na árvore. Aos nós de mesmo nível associamos os elementos correspondentes da malha. Assim, para relacionar os diferentes níveis da escala precisamos das informações em cada nó e em seus vizinhos.

A outra abordagem para a representação multiescala a ser considerada no presente trabalho utiliza ferramentas da análise *wavelet* [12, 16, 24]. Neste contexto, os esquemas permitem a representação de uma função em vários níveis de escala. Operadores de multirresolução que relacionam as informações em diferentes níveis de escala têm papel fundamental no desenvolvimento do assunto. Para passar de um nível mais fino para um mais grosseiro usamos o operador de restrição. Para passar do nível mais grosseiro para o mais fino, usamos o operador de previsão. Os coeficientes *wavelet* são os erros de previsão.

Uma das principais aplicações dos algoritmos de *wavelets* é na compressão de dados. Usando um processamento de multirresolução, por meio de uma aplicação reversível, os dados são transformados em um conjunto de coeficientes *wavelet*, a serem descartados ou não. A compressão é obtida quando ignoramos os coeficientes menores que uma tolerância previamente especificada.

Outra aplicação de interesse refere-se aos métodos de multirresolução, baseados em técnicas de *wavelets*, para a solução numérica adaptativa de equações diferenciais parciais para problemas em que podem ocorrer singularidades ou brusca variação no gradiente das soluções em regiões localizadas. O princípio é acelerar e reduzir o espaço de armazenamento na simulação numérica desse tipo de problema. Os coeficientes wavelet são usados como indicadores de regularidade local, para detectar a posição dessas regiões, permitindo assim, fazer um refinamento local da malha, que pode mudar com o passar do tempo. Desta forma, o refinamento de uma célula da partição do domínio ocorre quando o coeficiente *wavelet* associado a tal célula for significativo. Nas regiões em que a solução for suave, onde os coeficientes *wavelet* são desprezíveis, automaticamente células menos refinadas são atribuídas. A definição de tal tipo de estratégia é baseada em um esquema de referência em que os operadores diferenciais são aproximados por fórmulas para malhas uniformes. Nas discretizações em malhas adaptativas, os operadores de previsão são utilizados para aproximar as fórmulas que dependem de informações de células vizinhas, eventualmene ausentes nas malhas adaptadas. Tal procedimento conduz a uma redução das informações processadas, com baixo custo de armazenamento e tempo de CPU, desde que acompanhado de uma estrutura de dados apropriada. Para uma visão geral sobre métodos de multirresolução, são indicados os livros de Cohen [12], Müller [34] e os artigos de revisão sobre o assunto [40, 19].

No caso específico em que o esquema de referência é de volumes finitos, o trabalho pioneiro nessa área foi publicado em [23], em que A. Harten propôs o uso de esquemas em multirresolução para médias celulares para acelerar o processamento do método em malhas uniformes, aplicado a problemas diferenciais de leis de conservação unidimensionais. Nesse trabalho, os fluxos numéricos são calculados exatamente apenas nas células indicadas pelos coeficientes *wavelet* significativos e interpolados nas demais células, com a evolução sendo feita na malha uniforme, sem introduzir malhas adaptativas. Essa estratégia foi estendida a problemas bidimensionais em Bihari e Harten [1], Bihari [2] e em Chiavassa e Donat[11]. Posterioremente, as idéias de A. Hartem serviram como guia para novas melhorias, com a introdução de adaptabilidade espacial, como proposto em Kaibara e Gomes [29], Cohen et al. [14], Müller [34] e Roussel et al. [39, 38]. Em Domingues et al. [18, 17] e Müller e Stiriba [35], métodos numéricos mais elaborados são utilizados a fim de obter, além da adaptabilidade espacial, uma adaptabilidade no tempo. É importante observar que, ao se usar tais estratégias de multirresolução, o erro de aproximação pode ser controlado pelo parâmetro de truncamento, como provado em [14].

Tradicionalmente, os algoritmos em multirresolução são estruturados em refinamento de malhas  $2^{d}$ -ádicas [29, 14, 39, 38, 18, 17]. Uma malha  $2^{d}$ -ádica é uma hierarquia de malhas obtidas a partir de uma caixa d-dimensional que, em cada nível de refinamento, a caixa é dividida por hiperplanos ortogonais a cada um dos eixos coordenados. No caso bidimensional, a malha  $2^{d}$ -ádica é conhecida como quad-grid e no caso tridimensional, como oct-grid.

Métodos de multirresolução adaptativos geram dados que são esparsos e com estrutura complexa, que requer uma estrutura de dados eficiente em sua manipulação. As principais estruturas adotadas são as quad-trees para d = 2 e oct-trees, para d = 3 [39], ou hash tables [3]. Uma revisão sobre as diferentes técnicas é feita em Latu [32].

#### Proposta da tese

Neste trabalho, propomos desenvolver e implementar esquemas de aproximação em multinível estruturados em malhas diádicas para representar sinais d-dimensionais e para resolver, adaptativamente, problemas diferenciais.

Um dos focos principais desta tese é a proposta de modificar os esquemas de multirresolução (a partir daqui indicados pela sigla MR), armazenando e processando os dados utilizando uma estrutura de árvore binária. Especificamente, visando aplicações em métodos adaptativos de volumes finitos, nos concentramos em algoritmos em MR para dados obtidos de discretizações por médias celulares em malhas diádicas. Cada nó da árvore representa uma célula no nível correspondente da hierarquia de malhas. Os operadores de previsão e restrição acessam os dados contidos nos nós para estimar o valor a ser armazenado em um outro nó, acrescentando elementos à árvore ou eliminando alguns dos elementos. Espera-se que a simplicidade decorrente da codificação das malhas diádicas em árvore binária facilite a programação de algoritmos em MR adaptativos que possam ser aplicados em domínios computacionais *d*-dimensionais arbitários.

### Organização do texto

Este trabalho está organizado da seguinte forma:

No Capítulo 2, fazemos uma revisão de alguns conceitos básicos sobre teoria de aproximação que são centrais em Análise Numérica. Focamos nossa atenção em alguns aspectos gerais, tais como o tipo dos espaços das funções a serem aproximadas, dos espaços das funções aproximantes, possíveis critérios de aproximação e erros de aproximação. Uma vez definido o critério de aproximação, ainda resta escolher a maneira de se calcular tal aproximação, que depende da maneira em que as funções do espaço aproximante são representadas. Ou seja, os algoritmos de cálculo variam de acordo com a base escolhida. Esse efeito é notável se as bases são construídas em um único nível ou em vários níveis. Tais aspectos são ilustrados em dois exemplos unidimensionais simples: splines lineares e *wavelets* de Haar.

No Capítulo 3, expomos sobre a estrutura de dados a ser empregada nas implementações dos esquemas em multinível considerados. Apresentamos os conceitos de árvore binária ([31]), malhas diádicas e o processo de construção destas malhas([5, 6]), além da associação de uma malha diádica com uma árvore binária. A árvore binária é uma ferramenta importante para a codificação de esquemas estruturados em malhas diádicas, já que nos facilita a navegação pelos dados nos distintos níveis.

No Capítulo 4, tratamos o caso de espaços aproximantes por funções splines multilineares em malhas diádicas. Revisamos diferentes formas de construção de bases para tais espaços, tanto em malhas diádicas regulares quanto para malhas irregulares em multinível. Descrevemos o algoritmo de aproximação por interpolação e apresentamos um exemplo ilustrando esse critério de aproximação no caso bidimensional.

Os Capítulos anteriores são apresentações de resultados conhecidos, de caráter didático e de embasamento teórico, como motivação para os capítulos seguintes, os quais contém as contribuições desta tese.

No Capítulo 5, discutimos os esquemas de MR para médias celulares em malhas diádicas. Desenvolvemos algoritmos baseados em operadores de previsão de diferentes ordens e mostramos as semelhanças e diferenças com esquemas tradicionais de MR em malhas  $2^{d}$ -ádicas. Nesta tese, todas as implementações são para condições de contorno periódicas.

No Capítulo 6, fazemos uma revisão do aspecto funcional de análises de multirresolução para médias celulares para o caso unidimensional, conforme indicado no trabalho de Harten [24] e discutido em Kaibara [30]. A seguir, estendemos esta análise para o caso de esquemas MR em malhas diádicas em dimensões superiores.

No Capítulo 7, discutimos a maneira de obter uma representação em multirresolução adaptativa e a estimativa de erro. Também comparamos o desempenho das malhas diádicas e das *quad-grids* mediante a análise de dois exemplos diferentes no domínio bidimensional, usando médias celulares. Também ilustramos o desempenho do esquema MR adaptativo na compressão de uma imagem.

No Capítulo 8, apresentamos os conceitos envolvidos na construção de um método adaptativo que combina discretização por volumes finitos para leis de conservação e análise de MR para médias celulares em malhas diádicas. Para os esquemas de referência em malha uniforme, fazemos a aproximação dos fluxos numéricos de duas formas distintas. A primeira combina o esquema de Roe [37] com o esquema essencialmente não oscilatório (ENO), proposto por Harten e Osher [27]. A outra forma que consideramos é o esquema essencialmente não oscilatório ponderado WENO [33, 28]. A integração no tempo é feita por meio dos Métodos de Runge-Kutta de segunda e terceira ordem. Apresentamos exemplos unidimensionais para a equação de advecção, com intuito de verificar a ordem de convergência dos esquemas.

No Capítulo 9, apresentamos três experimentos numéricos bidimensionais que são

analisados sob a luz dos esquemas de referência para fluxos de Roe + ENO + RK2 e WENO + RK3, considerando os esquemas MR em malhas diádicas. Primeiro, resolvemos o problema de advecção. O segundo exemplo é a equação de Burgers e, finalmente, resolvemos o problema de Buckley-Leverett, equação que modela o escoamento de petróleo em um reservatório. Para cada um dos exemplos, é analisado o efeito da variação do parâmetro de truncamento dos coeficientes *wavelet* na eficiência dos esquemas MR em termos de redução de memória de armazenamento e ganho no tempo de CPU, quando comparados com os esquemas de referência de volumes finitos sobre as malhas uniformes do nível mais refinado.

No Capítulo final, relacionamos as conclusões da pesquisa desta tese e perspectivas de estudos futuros.

# Capítulo 2

# Conceitos Básicos em Teoria de Aproximação

Neste capítulo, fazemos uma revisão dos conceitos gerais em teoria da aproximação, tais como, espaço das funções a serem aproximadas V, espaços das funções aproximantes U, alguns critérios de aproximação e erros de aproximação. Uma vez definido o critério de aproximação, resta definir a maneira de calcular tal aproximação, que depende da escolha das bases do espaço aproximante, e analisar o erro cometido. Com exemplos unidimensionais simples, ilustramos tais aspectos: consideramos splines lineares e *wavelets* de Haar.

Mais detalhes sobre teoria geral de aproximação podem ser obtidos no livros de Hammerlin e Hoffmann [22] e Prenter [36]. Para os conceitos em multinível ver Cohen [12] e Müller [34].

### 2.1 Aspectos Gerais

O estudo da teoria de aproximação de funções envolve a análise de alguns aspectos como o espaço das funções a serem aproximadas V, o espaço das funções aproximantes U, o critério de aproximação (regra para se escolher a função aproximada), os algoritmos de cálculo e análise de erro de aproximação. Tipicamente, V é um espaço vetorial normado de dimensão infinita. Em teoria de aproximação, o interesse é encontrar um representante para uma função  $u \in V$  que seja um elemento de um espaço de dimensão finita U. Esperamos que os elementos de U sejam mais simples que os de V e que U tenha um bom *potencial de aproximação*. Isto quer dizer que um elemento  $u \in V$  é substituído por um elemento  $\tilde{u} \in U$  de forma que, se a dimensão de U tende ao infinito, a norma  $||u - \tilde{u}||$  tende a zero. Para implementar o critério de aproximação adotado, consideramos uma base  $B = \{\phi_0, \dots, \phi_n\}$  para U. Sendo assim, a aproximação para u é uma função  $\tilde{u} = \sum \alpha_j \phi_j$ , em que  $\alpha_j$  depende do critério de aproximação adotado. A escolha das funções de base  $\phi_j \in B$  influencia na definição dos algoritmos de cálculo e na estrutura de dados a ser utilizada.

#### 2.1.1 Critérios de Aproximação

O critério de aproximação é uma regra que, a cada  $u \in V$  atribui uma única função  $\tilde{u} \in U$ . O operador  $T: V \to U$ , em que  $\tilde{u} = Tu$ , pode ou não ser linear. Um exemplo de critério de aproximação é o da *melhor aproximação*. Ou seja,  $\tilde{u}$  é escolhido de forma a minimizar a distância entre a função  $u \in V$  e o espaço  $U \subset V$ . Em outras palavras, procuramos  $\tilde{u} \in U$  tal que  $||u - \tilde{u}|| = \inf_{w \in U} ||u - w||$ .

Formalmente, temos:

**Definição 2.1.** Seja V um espaço vetorial normado, munido da norma || ||. Seja U um subespaço de V. Dizemos que  $\tilde{u} \in U$  é a melhor aproximação para u em U se

$$\|\tilde{u} - u\| \le \|u - v\|, \quad \forall \quad v \in U.$$
 (2.1)

No caso em que a norma  $||v|| = \langle v, v \rangle^{\frac{1}{2}}$  é proveniente de um produto interno, a melhor aproximação é dada pela *projeção ortogonal* e podemos provar a existência e unicidade da melhor aproximação.

**Teorema 2.2.** Seja V um espaço com produto interno e  $U \subset V$  um subespaço de dimensão finita. Então a melhor aproximação de  $u \in V$  em U existe e é única. Além disto, verificase que  $(u - \tilde{u}) \perp V$ . A demonstração desse teorema se encontra em Hammerlin e Hoffmann [22, pág.138].

Outro critério de aproximação muito utilizado é a *interpolação*, que consiste em construir uma função  $\tilde{u}$  que coincide com a função a ser aproximada u em pontos dados no domínio de u. Sendo assim, dados  $t_0, \dots, t_n$ , (n + 1) pontos distintos no domínio das funções  $u \in V$ , o operador  $T: V \to U$  deve satisfazer  $\tilde{u}(t_i) = Tu(t_i) = u(t_i)$ .

Podemos provar que, em um espaço  $U = \text{span}\{\phi_0, \dots, \phi_n\}$ , em que as funções tenham no máximo n zeros, existe uma única função  $\tilde{u} \in U$  que satisfaz a condição de interpolação nos (n + 1) pares de dados  $(t_j, y_j)$ , para  $j = 0, \dots, n$  e  $y_j = u(t_j)$ . Por exemplo, se  $U = \mathbb{P}^n[a, b]$ . Detalhes podem ser vistos em Hammerlin e Hoffmann [22, pág.182].

### **2.1.2** Como Calcular a Aproximação $\tilde{u} = Tu$ ?

Uma vez definido o critério de aproximação T, resta escolher uma maneira de calcular a aproximação. Esse processo envolve a escolha da maneira de representar as funções de U, o que implica na escolha de uma base para o espaço aproximante U. Mais precisamente, supondo que  $B = \{\phi_0, \dots, \phi_n\}$  é uma base para o espaço U, então para cada função  $\tilde{u} \in U$ , obtemos uma única representação na forma

$$\tilde{u} = \sum_{j=0}^{n} \alpha_j \phi_j. \tag{2.2}$$

Assim, para calcular a aproximação  $\tilde{u} = Tu$  precisamos determinar os coeficientes que realizem tal critério de aproximação.

No caso em que o critério escolhido é o da projeção ortogonal, Teorema 2.2, temos que a aproximação é caracterizada pela condição de ortogonalidade  $(u - \tilde{u}) \perp V$ . Então, os coeficientes  $\alpha = (\alpha_j)^T$  são dados pelo sistema normal

$$\sum_{j=0}^{n} \alpha_j \langle \phi_j, \phi_k \rangle = \langle u, \phi_k \rangle, \quad 0 \le k \le n.$$
(2.3)

Na forma matricial, o sistema acima é reescrito como

$$M\alpha = b, \tag{2.4}$$

em que  $M_{kj} = \langle \phi_j, \phi_k \rangle, \ b_k = \langle u, \phi_k \rangle, \ \mathrm{com} \ j, k = 0, \cdots, n.$ 

No caso em que o critério de aproximação escolhido é o da interpolação, a aproximação

$$\tilde{u} = \sum_{j=0}^{n} \alpha_j \phi_j \tag{2.5}$$

deve satisfazer as condições de interpolação  $\tilde{u}(t_j) = u(t_j)$  para  $j = 0, \dots, n$ , e os coeficientes da aproximação satisfazem

$$\sum_{j=0}^{n} \alpha_j \phi_j(t_k) = u(t_k), \quad k = 0, \cdots, n,$$
(2.6)

que é reescrita na forma matricial por

$$M\alpha = b, \tag{2.7}$$

em que  $M_{kj} = \phi_j(t_k), \ b_k = u(t_k), \ {\rm com} \ j, k = 0, \cdots, n.$ 

### 2.1.3 Erro de Aproximação

Um aspecto crucial em teoria de aproximação é a análise do erro cometido em um dado esquema de aproximação. Em geral, o erro depende de três aspectos. O primeiro é o potencial de aproximação dos espaços aproximantes que, tipicamente, é definido por sua capacidade de reprodução de polinômios. Usualmente, o erro na melhor aproximação é da forma

$$||u - Tu|| \approx K N^{-p-1},$$

em que p é o grau máximo dos polinômios representados em U e N é a dimensão de U.

Outro aspecto diz respeito à regularidade da função aproximada. Para funções pouco regulares, a ordem máxima p + 1 pode não ser alcançada.

Finalmente, outro aspecto depende do esquema utilizado, o qual poderia não ter a mesma taxa de convergência dada pela melhor aproximação. Nos exemplos a seguir, comentamos esses aspectos.

### 2.2 Exemplos

Para ilustrar os conceitos discutidos nas seções anteriores, apresentamos alguns exemplos simples. O espaço das funções reais a serem aproximadas é V = C[a, b], isto é, as funções reais contínuas no intervalo  $[a, b] \subset \mathbb{R}$ , munido da norma  $||u||_{\infty} = \sup_{t \in [a, b]} |u(t)|$  ou  $V = \mathbb{L}^2[a, b]$  com a norma  $||u||_2 = \sqrt{\langle u, u \rangle}$ , induzida pelo produto interno  $\langle u, v \rangle = \int_a^b u(t)v(t)dt$ .

### 2.2.1 Splines Lineares

Seja  $\gamma = \{c_j : 0 \le j < n\}$  uma partição do intervalo [a, b] por subintervalos  $c_j = (t_j, t_{j+1})$ , em que  $t_0 = a, t_j < t_{j+1}, t_n = b$ , de forma que  $[a, b] = \bigcup_j \overline{c_j}$ .

O espaço das funções aproximantes é o subespaço das funções contínuas lineares por partes, denominado de *splines lineares* sobre a malha  $\gamma$ , definido por

$$S^{1}(\gamma) = \{s : s \in C^{0}[a, b] \in s|_{c_{j}} \in \mathbb{P}^{1}, 0 \le j \le n - 1\}.$$
(2.8)

Considerando uma partição uniforme  $\gamma_h$ , em que  $t_j = a + jh$ , com  $0 \le j \le n$  e h = (b-a)/n, é possível verificar que  $S^1(\gamma_h) = \operatorname{span}\{\phi_j^h : 0 \le j \le n\}$ , em que as funções de base  $\phi_j^h : [a, b] \to [0, 1]$  são definidas por

$$\phi_j^h(t) = \phi(h^{-1}t - j), \tag{2.9}$$

em que

$$\phi(t) = \begin{cases} 1+t, & \text{se } t \in [-1,0], \\ 1-t, & \text{se } t \in [0,1], \\ 0, & \text{caso contrário.} \end{cases}$$
(2.10)

Na Figura 2.1, mostramos o gráfico de alguns elementos da base de  $S^1(\gamma_h)$ , especificamente de  $\phi_0^h$  e  $\phi_n^h$  nos extremos e  $\phi_{j-1}^h$ ,  $\phi_j^h$ ,  $\phi_{j+1}^h$  no interior.

O potencial de aproximação neste caso é da ordem de  $h^2$ , tendo em vista que somente polinômios de grau  $p \leq 1$  podem ser representados exatamente em  $S^1(\gamma_h)$ .

No espaço aproximante  $S^1(\gamma_h)$  podem ser empregados diversos métodos de aproximação, entre eles estão a interpolação e a projeção ortogonal.



Figura 2.1: Elementos da base de  $S^1(\gamma_h)$ .

### Interpolação

Quando buscamos uma aproximação  $\tilde{u}_h \in S^1(\gamma_h)$  que seja interpolatória, o problema a se resolver é simples. De fato, em termos da base de  $S^1(\gamma_h)$ , escrevemos a função  $\tilde{u}_h$  na forma

$$\tilde{u}_h(t) = \sum_{j=0}^n \alpha_j \phi_j^h(t).$$
(2.11)

Lembramos que, para ser interpolatória, a aproximação  $\tilde{u}_h$  satisfaz  $\tilde{u}_h(t_j) = u(t_j)$ , para todo  $0 \leq j \leq n$ . Como  $\phi_j^h(t_k) = \delta_{jk}$ , os coeficientes  $\alpha_j$  da combinação linear (2.11) são dados por  $\alpha_j = u(t_j)$ .

**Exemplo 2.1.** Na Figura 2.2, exibimos os gráficos da função a ser aproximada  $u(t) = \sin(\pi t)$  (curva contínua) e da aproximação  $\tilde{u}_h$  (curva descontínua), em que [a, b] = [-1, 1], n = 4.



Figura 2.2: Exemplo 2.1: gráficos da função u (sólida) e da aproximação interpolatória por spines lineares  $\tilde{u}$  (pontilhada).

Resultados teóricos (ver [22, pág.256]) garantem que, se u é Lipschitz contínua, então a seqüência de splines lineares interpolatórios  $\tilde{u}_h$  converge uniformemente para u com taxa linear com respeito ao espaçamento h. Isto é,

$$||E||_{\infty} = ||u - \tilde{u}_h||_{\infty} \le Kh$$

Por outro lado, se  $u \in C^2[a, b]$ , temos convergência quadrática com respeito a h, ou seja,

$$\|u - \tilde{u}_h\|_{\infty} \le \frac{h^2}{8} \|u''\|_{\infty}.$$
(2.12)

De forma geral, mesmo que consideremos uma função  $u \in C^{q}[a, b]$ , com q > 2, não podemos obter taxa de convergência melhor que a quadrática. Um dos fatos que contribuem para essa limitação é que apenas os polinômios de grau  $p \leq 1$  são exatamente representados em  $S^{1}(\gamma_{h})$ .

Verifiquemos numericamente essas taxas de convergência.

**Exemplo 2.2.** Para ilustrar o caso de convergência linear, consideramos a função Lipschitz contínua  $u(t) = |t^2 - 0.5|$ , com  $t \in [-1, 1]$ . Na Figura 2.3 (b), observamos o gráfico log-log do erro  $||E||_{\infty}$  como função de h. A reta que melhor se ajusta a esses dados (curva contínua) possui coeficiente angular 1,0624.

**Exemplo 2.3.** Considerando a função de classe  $C^2[-1, 1]$ ,  $u(t) = \sin(\pi t)$ , a curva tracejada, na Figura 2.3 (b), é o gráfico log-log de  $||E||_{\infty}$  como função de h. Calculando a inclinação da reta que melhor se ajusta a esses dados, verificamos que a ordem de aproximação numérica é igual a 2,0010.

### Projeção Ortogonal

Seja  $V = \mathbb{L}^2[-1, 1]$  o espaço das funções a serem aproximadas, munido da norma  $|| ||_2$ . Dada uma função u a ser aproximada, sabemos que a sua melhor aproximação  $\hat{u}$  em  $S^1(\gamma_h)$ satisfaz a propriedade da ortogonalidade  $\langle u - \hat{u}_h, \phi_j \rangle = 0$ , que implica em satisfazer a Equação (2.3).



Figura 2.3: (a) Gráfico das funções a serem analisadas: uma Lipschitziana (curva contínua) e uma de classe  $C^2[-1,1]$  (curva descontínua). (b) Gráfico log-log dos erros de interpolação  $||E_h||_{\infty}$  como função de h: as retas de ajuste linear possuem coeficientes angular 1,0624, para o Exemplo 2.2 e 2,0010, para o Exemplo 2.3.

Observadas as propriedades de suporte local dos elementos da base de  $S^1(\gamma_h)$ , verificamos que a matriz de coeficientes M em (2.4) é uma matriz tridiagonal, dada por

**Exemplo 2.4.** Neste exemplo, consideramos [a, b] = [-1, 1], n = 4, h = 0, 5 e a função

 $u(t) = \sin(\pi t)$ . Os coeficientes da combinação linear  $\hat{u}_h = \sum_{j=0}^4 \alpha_j \phi_j$  são obtidos resolvendo o sistema normal

$$\begin{bmatrix} \frac{1}{6} & \frac{1}{12} & 0 & 0 & 0\\ \frac{1}{12} & \frac{1}{3} & \frac{1}{12} & 0 & 0\\ 0 & \frac{1}{12} & \frac{1}{3} & \frac{1}{12} & 0\\ 0 & 0 & \frac{1}{12} & \frac{1}{3} & \frac{1}{12}\\ 0 & 0 & 0 & \frac{1}{12} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_2\\ \alpha_3\\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -0.115668\\ -0.405285\\ 0\\ 0.405285\\ 0.115668 \end{bmatrix}$$

Assim,  $\alpha_0 = -0.0983749$ ,  $\alpha_1 = -1.19126$ ,  $\alpha_2 = 4.48359 * 10^{-17}$ ,  $\alpha_3 = 1.19126$ ,  $\alpha_4 = 0.0983749$ .

Na Figura 2.4, a função a ser aproximada u é representada pela curva contínua e a melhor aproximação  $\hat{u}_h$ , dada pela projeção ortogonal, pela curva tracejada.



Figura 2.4: Exemplo 2.4: gráficos da função u (contínua) e a melhor aproximação, dada pela projeção ortogonal, nos splines lineares  $\hat{u}_h$  (tracejada), com h = 0, 5.

De modo geral, é de interesse saber quão bem a projeção ortogonal  $\hat{u}_h$  representa a função u, à medida que h tende a zero. Para obtermos uma estimativa da taxa de convergência do erro de aproximação obtido pelo critério de projeção ortogonal, usamos a taxa de convergência do erro de interpolação. Consideramos então as aproximações  $\hat{u}_h$ e  $\tilde{u}_h$ , em que esta é a aproximação de u por interpolação e aquela é a projeção ortogonal de u em  $S^1(\gamma_h)$ . Tal relação (ver [22, pág.258]) é dada por

$$||u - \hat{u}_h||_2 \le ||u - \tilde{u}_h||_2 \le \sqrt{(b-a)}||u - \tilde{u}_h||_{\infty}.$$

Segue que, se  $u \in C^2[a, b]$ , a convergência é quadrática com respeito a h, veja Equação (2.12), ou seja,

$$||E_h||_2 = ||u - \hat{u}_h||_2 \le Kh^2.$$

Verificamos numericamente a ordem de aproximação, dada pelo coeficiente angular da reta de ajuste dos dados log  $||E_h||$  versus  $\log(h)$ .

No caso do Exemplo 2.4, em que a função  $u(t) = \sin(\pi t)$  e  $-1 \le t \le 1$ , a ordem de aproximação numérica é 2.01941. Na Figura 2.5, vemos o gráfico log-log do erro  $||E_h||_2$  como função de h.



Figura 2.5: Exemplo 2.4: gráfico log-log do erro  $||E_h||_2$  como função de h.

#### **Base Hierárquica**

Como motivação para a teoria apresentada no Capítulo 4, nesta seção descrevemos outra forma de representar funções de  $S^1(\gamma)$  em termos de uma base hierárquica, no caso em que a partição  $\gamma$  for proveniente de uma malha diádica.

Tomamos uma hierarquia de partições  $\gamma_{\ell}$ , com  $\ell = 0, \ldots$ , em que o comprimento dos subintervalos da partição é  $h_{\ell} = 2^{-\ell}(b-a)$  e contendo  $n_{\ell} = 2^{\ell}$  subintervalos. Na Figura 2.6, esboçamos tal hierarquia de coberturas, de forma que  $\gamma_{\ell+1}$  é obtida a partir de  $\gamma_{\ell}$  pela divisão diádica de cada  $c_j^{\ell} \in \gamma_{\ell}$  em dois subintervalos  $c_{2j}^{\ell+1}$ ,  $c_{2j+1}^{\ell+1}$ , de igual tamanho, conforme indicado na Figura 2.7.

Defindo os espaços  $U^{\ell} = S^1(\gamma_{\ell})$ , podemos verificar a inclusão  $U^{\ell} \subset U^{\ell+1}$ . Sendo


Figura 2.6: Hierarquia de malhas.



Figura 2.7: Intervalos das partições  $\gamma_{\ell} \in \gamma_{\ell+1}$ .

assim, o espaço  $U^{\ell+1}$ pode ser decomposto em soma direta

$$U^{\ell+1} = U^{\ell} \oplus W^{\ell},$$

em que  $W^{\ell} = \{u \in U^{\ell+1} : u(t_j^{\ell}) = 0, 0 \le j \le n_{\ell}\}$  são as funções de  $U^{\ell+1}$  que se anulam na malha  $\gamma_{\ell}$ , que tem por base as funções  $\psi_j^{\ell} = \phi_{2j+1}^{\ell+1}, 0 \le j \le n_{\ell} - 1$ .

Dessa forma, escolhido L como o nível de resolução mais fino, é possível apresentar uma base para  $S^1(\gamma_L)$  sob dois aspectos. No primeiro, discutido na Seção 2.2, em que a base é dada por  $B^L = \{\phi_j^L, 0 \le j \le n_L\}$ . O segundo é a base em dois níveis de resolução, em que

$$B_{MR}^{L} = \{\phi_j^{L-1} : 0 \le j \le n_{L-1}\} \cup \{\psi_j^{L-1} : 0 \le j \le n_{L-1} - 1\},\$$

que é dada pela união disjunta das bases de  $U^{L-1}$  e  $W^{L-1}$ .

**Exemplo 2.5.** Considerando [a, b] = [-1, 1], L = 2,  $n_2 = 4$ , vemos na Figura 2.8, a base hierárquica para o espaço  $U^2 = S^1(\gamma_2)$ . As curvas tracejadas representam os elementos da base de  $U^1 = S^1(\gamma_1)$  e as curvas sólidas representam os elementos da base de  $W^1$ .



Figura 2.8: Elementos da base de  $S^1(\gamma_1)$  (tracejada) e  $W^1$  (sólida).

Assim, podemos escrever  $u \in S^1(\gamma_L)$  em termos da base de um nível

$$u(t) = \sum_{j=0}^{n_L} u(t_j^L) \phi_j^L(t), \qquad (2.14)$$

ou na base de dois níveis

$$u(t) = \sum_{j=0}^{n_{L-1}} u(t_j^{L-1})\phi_j^{L-1}(t) + \sum_{j=0}^{n_{L-1}-1} d_j^{L-1}\psi_j^{L-1}(t), \qquad (2.15)$$

em que

$$d_j^{L-1} = u(t_{2j+1}^L) - \frac{u(t_j^{L-1}) + u(t_{j+1}^{L-1})}{2}.$$

O coeficiente  $d_j^{L-1}$  mede a diferença entre o valor exato para a função u em  $t_{2j+1}^L$  e o valor dado pela interpolação linear dos valores de u nos pontos vizinhos  $t_{j+1}^{L-1}$  e  $t_j^{L-1}$ , como indicado na Figura 2.9.

Introduzimos uma terminologia para os operadores que relacionam as informações entre as diferentes escalas de resolução, em que  $\ell = 0, \dots, L-1$ . Ao operador que permite passar do nível menos refinado  $\ell$  para o nível mais refinado  $\ell + 1$  denominamos *previsão* e tal operador é denotado por  $P_{\ell}^{\ell+1}$ . Para passarmos do nível  $\ell + 1$  para o nível  $\ell$  usamos o operador de *restrição*, denotado por  $P_{\ell+1}^{\ell}$ . Precisamente, sejam  $u^{\ell} = (u_j^{\ell})$  as amostras de u na malha  $\gamma_{\ell}$ . Portanto,

$$(P_{\ell+1}^{\ell} u^{\ell+1})_j = u_{2j}^{\ell+1},$$
  

$$(P_{\ell}^{\ell+1} u^{\ell})_{2j+1} = \frac{u_j^{\ell} + u_{j+1}^{\ell}}{2},$$



Figura 2.9: Representação geométrica do coeficiente  $d_j^{L-1}$ .

O coeficiente  $d_j^{\ell}$  mede a diferença entre o valor exato para a função u em  $t_{2j+1}^{\ell+1}$  e a previsão do valor de u no mesmo ponto.

Os algoritmos que relacionam as informações  $u^L$ ,  $u^{L-1}$  e  $d^{L-1}$  são os algoritmos MR direito e MR inverso.

Para obter os vetores  $u^{L-1}$  <br/>e $d^{L-1}$ a partir de  $u^L$ utilizamos o algoritmo MR direto, dado por

Para 
$$j = 0, \cdots, n_{L-1}$$
  
 $u_j^{L-1} = u_{2j}^L,$   
 $d_j^{L-1} = u_{2j+1}^L - \frac{u_j^{L-1} + u_{j+1}^{L-1}}{2},$ 

o qual é representado pelo diagrama



O algoritmo MR inverso tem como dados de entrada as informações contidas nos vetores  $u^{L-1}$  e  $d^{L-1}$  e recupera  $u^L$  por

Para 
$$j = 0, \cdots, n_{L-1}$$
  
 $u_{2j}^L = u_j^{L-1},$   
 $u_{2j+1}^L = d_j^{L-1} + \frac{u_j^{L-1} + u_{j+1}^{L-1}}{2}$ 

o qual é representado pelo diagrama

$$u^{L-1} \xrightarrow{} u^{L}. \tag{2.17}$$
$$d^{L-1}$$

Tais algoritmos podem ser executados em multinível. O algoritmo MR direto se expressa como

$$u^{L} \xrightarrow{} u^{L-1} \xrightarrow{} u^{L-2} \xrightarrow{} \cdots \xrightarrow{} u^{0}$$

$$(2.18)$$

$$d^{L-1} \quad d^{L-2} \quad d^{L-3} \quad d^{0}$$

e o algoritmo MR inverso se expressa como



Os algoritmos MR direto e inverso correspondem à mudança da bas<br/>e $B^L$ para a base multinível $B^L_{MR}$ para<br/>  $U^L$  dada por

$$B_{MR}^{L} = \{\phi_{j}^{0} : 0 \le j \le n_{0}\} \cup \left\{\bigcup_{\ell=0}^{L-1} \{\psi_{j}^{\ell} : 0 \le j \le n_{\ell} - 1\}\right\}.$$

Ou seja, as funções  $u \in S^1(\gamma_L)$ são representas na base multinível  $B^L_{MR}$  pela expansão

$$u(t) = \sum_{j=0}^{n_0} u(t_j^0) \phi_j^0(t) + \sum_{\ell=0}^{L-1} \sum_{j=0}^{n_{L-1}} d_j^\ell \psi_j^\ell(t).$$
(2.20)

#### Adaptatividade em Multinível

A vantagem em usarmos bases hierárquicas está no fato dessas bases possibilitarem uma representação adaptativa da função a ser aproximada. O processo adaptativo não pode ser feito aleatoriamente, precisa ser guiado por algum parâmetro que indique onde e como fazer a adaptatividade. Vimos que o coeficiente  $d_j^{\ell}$  representa o erro de interpolação linear no ponto  $t_{2j+1}^{\ell+1}$ , interpolação obtida a partir dos dados vizinhos a  $t_j^{\ell}$  na malha  $\gamma_{\ell}$ .

Esperamos então que, em regiões onde a função a ser aproximada é suave, os coeficientes sejam pequenos. De maneira análoga, o erro de interpolação deve ser significante nas regiões onde a função tem variação brusca ou apresenta descontinuidades. Assim, os coeficientes  $d_j^{\ell}$  podem ser usados para estimar a regularidade da função a ser aproximada.

Obtemos a adaptatividade mantendo apenas os termos  $d_j^{\ell}$  da expansão em multinível (2.20) que são significativos. Especificamente, dado um parâmetro de tolerância  $\epsilon > 0$ , obtemos a adaptatividade a partir do truncamento dos coeficientes  $d_j^{\ell}$ , isto é, tomamos

$$\tilde{d}_{j}^{\ell} = \begin{cases} 0 & \text{se } |d_{j}^{\ell}| \leq \epsilon, \\ d_{j}^{\ell} & \text{se } |d_{j}^{\ell}| > \epsilon. \end{cases}$$

$$(2.21)$$

Assim, define-se a aproximação adaptativa

$$u_{\epsilon} = \sum_{j=0}^{n_0} u(t_j^0) \phi_j^0 + \sum_{\ell=0}^{L-1} \sum_{j=0}^{n_\ell - 1} \tilde{d}_j^{\ell} \psi_j^{\ell}.$$
 (2.22)

**Exemplo 2.6.** Na Figura 2.10, à esquerda, exibimos o gráfico de uma função u(t),  $-1 \le t \le 1$ , em que

$$u(t) = \begin{cases} \frac{\cos(\pi t) + 1}{2}, & \text{se } t < -\frac{\pi}{6}, \\ |\sin(0.6\pi t - 1)| & \text{se } t \ge -\frac{\pi}{6}. \end{cases}$$

Note que u é descontínua em  $t = -\frac{\pi}{6}$  e apresenta uma variação brusca em  $t = \frac{10}{6\pi}$ , com descontinuidade na derivada u'. No lado direito, a figura indica a posição dos coeficientes relevantes  $|d_j^{\ell}| > \epsilon$ , em que  $\epsilon = 0.001$  e  $1 \le \ell \le 10$ . Observamos que os coeficientes  $d_j^{\ell}$  significativos indicam as posições onde a função u apresenta variação brusca e/ou descontinuidade.

### 2.2.2 Wavelets de Haar

Como motivação para a teoria a ser apresentada no Capítulo 5, consideramos o exemplo das *wavelets* de Haar. Nesta seção, o espaço das funções a serem aproximadas é  $V = \mathbb{L}^2[a, b]$ , munido da norma  $\|\cdot\|_2$ .

O espaço aproximante é dado por

$$U^{\ell} = S^{0}(\gamma_{\ell}) = \left\{ u \in \mathbb{L}^{2}[a, b] : u|_{c_{j}^{\ell}} = u_{j}^{\ell} \right\},$$





Figura 2.10: Exemplo 2.6: (a) gráfico de u(t) (curva contínua) e da aproximação adaptativa  $u_{\epsilon}(t)$  (+), com  $\epsilon = 0.001$  e L = 10; (b) posição, por nível, dos pontos da malha hierárquica associados aos coeficientes relevantes  $|d_j^{\ell}| > 0.001$ ,  $1 \le \ell \le 10$ .

É fácil ver que as funções  $\phi_j^\ell,$  definidas por

$$\phi_j^\ell(t) = \mathcal{X}_{c_j^\ell}(t) = \begin{cases} 1, & t \in c_j^\ell, \\ 0, & t \notin c_j^\ell, \end{cases}$$
(2.23)

para  $0 \leq j \leq n_{\ell} - 1$ , formam uma base ortogonal para o espaço  $S^0(\gamma_{\ell})$ . Logo

$$S^{0}(\gamma_{\ell}) = \operatorname{span} \left\{ \phi_{j}^{\ell}, 0 \le j \le n_{\ell} - 1 \right\}.$$
(2.24)

A aproximação

$$\hat{u}^{\ell} = P^{\ell} u = \sum_{j=0}^{n_{\ell}-1} u_j^{\ell} \phi_j^{\ell}, \qquad (2.25)$$

em que

$$u_j^\ell = \frac{1}{|c_j^\ell|} \int_{c_j^\ell} u(t) \mathrm{d}t = \frac{1}{|c_j^\ell|} \langle u, \phi_j^\ell \rangle$$

são as médias de u nas células  $c_j^{\ell}$ , corresponde à projeção ortogonal de u sobre  $S^0(\gamma_{\ell})$ .

Como o espaço  $S^0(\gamma)$  representa exatamente apenas os polinômios de grau 0, sabemos que a ordem de aproximação para o exemplo de Haar é dada por ([12])

$$\|u - P^{\ell}u\|_2 = \mathcal{O}(h),$$

em que  $h = 2^{-\ell}$ .

**Exemplo 2.7.** Considerando a função  $u(t) = 1 - t^2$ , com  $-1 \le t \le 1$ , vemos na Figura 2.11, o gráfico de u (curva contínua) e o gráfico da projeção  $\hat{u}^3$  (curva constante por partes) de u sobre o espaço  $S^0(\gamma_3)$ . Na Figura 2.12, temos o gráfico log-log do erro de  $||E||_2$  como função de h.

#### Aproximação em Multinível

Sendo, para cada  $\ell = 0, \cdots, L-1, U^{\ell} \subset U^{\ell+1}$ , o espaço  $U^{\ell+1}$  pode ser escrito como soma direta

$$U^{\ell+1} = U^{\ell} \oplus W^{\ell},$$

em que  $W^{\ell}$  é o complemento ortogonal de  $U^{\ell}$  em  $U^{\ell+1}$ .



Figura 2.11: Exemplo 2.7: gráfico da função  $u(t) = 1 - t^2$  (contínua) e de sua projeção em  $S^0(\gamma_3)$  (constante por partes).



Figura 2.12: Exemplo 2.7: gráfico log-log do erro  $||E||_2$  como função de h, reta com inclinação 1.

Os elementos da base do espaço  $W^{\ell}$ , denominados de *wavelets*, são dados por

$$\psi_j^{\ell}(t) = \mathcal{X}_{c_{2j+1}}^{\ell+1}(t) - \mathcal{X}_{c_{2j}}^{\ell+1}(t) = \phi_{2j+1}^{\ell+1} - \phi_{2j}^{\ell+1}$$

Na Figura 2.13, esboçamos o gráfico de uma wavelet de Haar $\psi_j^\ell.$ Note que as funções  $\phi_j^\ell$ e $\psi_k^\ell$ satisfazem

$$\begin{split} \langle \psi_j^{\ell}, \psi_j^{\ell} \rangle &= |c_j^{\ell}|, \qquad \langle \psi_j^{\ell}, \psi_k^{\ell} \rangle = 0, \quad j \neq k, \\ \langle \phi_j^{\ell}, \psi_k^{\ell} \rangle &= 0, \quad \forall j, k. \end{split}$$

A projeção de  $u \in \mathbb{L}^2[a,b]$  em  $S^0(\gamma_L)$  é dada por

$$P^{L}u = \sum_{j=0}^{n_{L}-1} u_{j}^{L}\phi_{j}^{L} = \sum_{j=0}^{n_{L-1}-1} u_{j}^{L-1}\phi_{j}^{L-1} + \sum_{j=0}^{n_{L-1}-1} d_{j}^{L-1}\psi_{j}^{L-1}.$$
 (2.26)

Observe que o projetor definido na equação (2.26) tem a forma  $P^L = P^{L-1} + Q^{L-1}$ , em que  $Q^{L-1}$  é o projetor ortogonal sobre o espaço  $W^{L-1}$ . Para determinar  $Q^{L-1}$ , resta exibir a maneira de se obter os *coeficientes wavelet*  $d_j^{L-1}$ .



Figura 2.13: Gráfico da função wavelet  $\psi_j^{\ell}(t)$ .

Como as funções  $\psi_j^{L-1}$  formam uma base ortogonal para  $W^{L-1},$  segue que

$$d_j^{L-1} = \frac{\langle u, \psi_j^{L-1} \rangle}{\langle \psi_j^{L-1}, \psi_j^{L-1} \rangle} = \frac{1}{|c_j^{L-1}|} \langle u, \psi_j^{L-1} \rangle = \frac{u_{2j+1}^L - u_{2j}^L}{2}$$

Na representação de  $\hat{u}^L = P^L u = P^{L-1}u + Q^{L-1}u$ , da equação (2.26), há informações relacionadas a dois níveis de refinamento. Os coeficientes  $u_j^{L-1}$  fornecem as informações sobre u na cobertura mais grosseira e podem ser obtidos das informações no nível L pela fórmula

$$u_j^{L-1} = \frac{u_{2j}^L + u_{2j+1}^L}{2}.$$
(2.27)

Os detalhes  $d_j^{L-1}$  são as informações necessárias para recuperar  $u^L$  a partir de  $u^{L-1}$ .

A maneira de relacionar as informações entre os diferentes níveis de resolução,  $\ell = 0, \dots, L-1$ , é feita por meio do uso do operador de *previsão* e do operador de *restrição*. Quando queremos passar de um nível mais fino para um mais grosseiro, usamos o operador de restrição  $P_{\ell+1}^{\ell}$ . Quando passamos do nível mais grosseiro para o mais fino, aplicamos o operador de previsão  $P_{\ell}^{\ell+1}$ . Os coeficientes *wavelet* são os erros de previsão. No caso das *wavelets* de Haar temos,

$$(P_{\ell+1}^{\ell} u^{\ell+1})_j = \frac{u_{2j}^{\ell+1} + u_{2j+1}^{\ell+1}}{2},$$
$$(P_{\ell}^{\ell+1} u^{\ell})_{2j+1} = u_j^{\ell}.$$

#### Algoritmos MR

Os algoritmos MR tem como objetivo a passagem de uma representação em um nível para uma representação em multinível. Por ser um processo reversível, pode-se obter novamente a solução em um único nível. Especificamente, o algoritmo MR direto (análise) converte os dados de um nível em dados que relacionam vários níveis de resolução e tem como dado de entrada o vetor de médias celulares no nível mais refinado da malha  $u^L$ . O dado de saída é  $\{u^0, d^0, \dots, d^{L-1}\}$ , em que  $u^0$  é a média da função em toda raiz, o domínio de  $u \in d^{\ell}, 0 \leq \ell < L - 1$  são os erros de previsão.

De forma análoga, o algoritmo MR inverso (síntese) converte dados que relacionam diferentes níveis de resolução em dados referentes a um único nível e tem como dados de entrada as informações em multinível  $\{u^0, d^0, \dots, d^{L-1}\}$  e devolve o vetor de médias celulares  $u^L$ .

Em resumo, considerando dois níveis de resolução, o algoritmo MR direto (análise) é dado por

Para 
$$j = 0, \dots, n_{L-1} - 1$$
  
 $u_j^{L-1} = \frac{u_{2j}^L + u_{2j+1}^L}{2},$   
 $d_j^{L-1} = \frac{u_{2j+1}^L - u_{2j}^L}{2} = u_{2j+1}^L - u_j^{L-1},$ 

e é representado pelo diagrama

$$u^{L} \xrightarrow{} u^{L-1}$$

$$(2.28)$$

$$d^{L-1}.$$

O algoritmo MR inverso (síntese), em dois níveis de resolução, tem como dados de entrada as informações contidas nos vetores  $u^{L-1} = (u_j^{L-1})$  e  $d^{L-1} = (d_j^{L-1})$  e recupera o vetor  $u^L$  por

Para 
$$j = 0, \cdots, n_{L-1}$$
  
 $u_{2j+1}^L = d_j^{L-1} + u_j^{L-1},$   
 $u_{2j}^L = 2u_j^{L-1} - u_{2j+1}^L,$ 

e é representado pelo diagrama



Tais algoritmos podem ser representados em multinível. O algoritmo MR direto se expressa como

e o algoritmo MR inverso é dado por



A representação da aproximação  $\hat{u} = P^L u$  em multiresolução é dada por

$$P^{L}u = \sum_{j=0}^{n_{0}-1} u_{j}^{0}\phi_{j}^{0} + \sum_{\ell=0}^{L-1} \sum_{j=0}^{n_{\ell}-1} d_{j}^{\ell}\psi_{j}^{\ell}.$$
(2.32)

#### Adaptatividade

A vantagem de usarmos  $\hat{u} = P^L u$  na forma (2.32) é que ela torna possível a adaptatividade na expressão de  $\hat{u}$ . Como, para  $\ell = 0, \dots, L-1$ , o coeficiente  $d_j^{\ell}$  representa o erro da previsão da média celular de uma função u na célula  $c_{2j+1}^{\ell+1}$ , em que  $u_{2j+1}^{\ell+1} \approx u_j^{\ell}$  e  $\overline{c}_j^{\ell} = \overline{c}_{2j}^{\ell+1} \cup \overline{c}_{2j+1}^{\ell+1}$ , cada  $d_j^{\ell}$  é uma medida de regularidade local da função a ser aproximada. Nas regiões onde a função apresenta uma variação brusca ou descontinuidade, esperamos que esses coeficientes sejam grandes. De modo análogo, nas regiões de suavidade, supomos que os coeficientes sejam pequenos. A adaptatividade é obtida pelo truncamento dos coeficientes *wavelet* que estão abaixo de uma tolerância previamente especificada. O processo de truncamento depende do nível de resolução, permitindo um "erro maior" nos níveis de maior resolução e um "erro menor" nos níveis mais grosseiros. Especificamente, dado  $\epsilon > 0$ , Harten [23] propôs que, para cada nível  $\ell$ ,

$$\epsilon_{\ell} = 2^{(\ell - L)} \epsilon,$$

em que L é o nível máximo de resolução da malha. Assim, o truncamento dos coeficientes  $d_i^\ell$  é dado por

$$\tilde{d}_{j}^{\ell} = \begin{cases} 0 & \text{se } |d_{j}^{\ell}| \leq \epsilon_{\ell}, \\ d_{j}^{\ell} & \text{se } |d_{j}^{\ell}| > \epsilon_{\ell}. \end{cases}$$
(2.33)

Então, uma aproximação  $\hat{u}^L_\epsilon$  para  $\hat{u}^L$  é dada por

$$\hat{u}^{L} \approx \sum_{j=0}^{n_{0}} u_{j}^{0} \phi_{j}^{0}(t) + \sum_{\ell=0}^{L-1} \sum_{j=0}^{n_{\ell}-1} \tilde{d}_{j}^{\ell} \psi_{j}^{\ell}(t) = \hat{u}_{\epsilon}^{L}.$$
(2.34)

**Exemplo 2.8.** Na Figura 2.14, à esquerda, plotamos o gráfico da função  $u(t), -1 \le t \le 1$ 

$$u(t) = \begin{cases} \frac{\cos(\pi t) + 1}{2}, & \text{se } t < -\frac{\pi}{6}, \\ |\sin(0.6\pi t - 1)| & \text{se } t \ge -\frac{\pi}{6}. \end{cases}$$

Observe que a função u apresenta descontinuidade em  $t = -\frac{\pi}{6}$  e variação brusca em  $t = \frac{10}{6\pi}$ , com descontinuidade na derivada u'. No lado direito, a figura indica o centro das células da partição sobre as quais os coeficientes wavelets  $|d_j^{\ell}| > \epsilon_{\ell}$ , para níveis de refinamento  $1 \leq \ell \leq 9$  e  $\epsilon = 0, 1$ . Note que, nas regiões onde a função u apresenta descontinuidade ou forte variação no gradiente, as células da partição estão em níveis mais refinados (elevados).

A operação de truncamento diz que os coeficientes  $d_j^{\ell}$  desprezíveis não precisam ser armazenados, acarretando na compressão de dados. Em termos práticos, como mostrado em [30], a aproximação adaptativa (2.34) pode ser obtida apenas com as médias celulares nas células associadas aos coeficientes  $\tilde{d}_j^{\ell}$  significativos. No Capítulo 3, veremos como a adaptatividade da função em MR está associada com a adaptatividade na estrutura de dados empregada para codificação da malha adaptada. Isto é, criamos campos na estrutura de dados apenas para armazenar os valores das médias celulares associadas aos coeficientes wavelet significativos, acarretando na compressão de dados.



Figura 2.14: Exemplo 2.8: (a) u(t) (curva contínua) e aproximação adaptativa  $u_{\epsilon}(t)$ , com L = 9 e  $\epsilon = 0.1$ ; (b) distribuição em cada nível da posição dos coeficientes wavelet significativos.

Para o Exemplo 2.8, medimos a taxa de compressão como a porcentagem de células da da malha adaptativa com relação ao total de células da partição uniforme do nível mais refinado. Na Tabela 2.1 apresentamos a taxa de compressão e o erro entre a projeção  $\hat{u}$  e a aproximação  $\hat{u}_{\epsilon}$ , na norma  $L_1$ , para diferentes valores de  $\epsilon$ . Para cada valor de  $\epsilon$ , consideramos uma malha uniforme inicial com L = 10 níveis de refinamento, contendo  $2^{10} = 1024$  células em seu nível mais refinado. Observamos que a diferença entre a projeção e a representação adaptativa é proporcional ao parâmetro de truncamento  $\epsilon$ . A compressão diminui com o parâmetro  $\epsilon$ .

$\epsilon$	Compressão	$\ \hat{u} - \hat{u}_{\epsilon}\ _1$
$10^{-1}$	$89,\!453\%$	0.011938
$10^{-2}$	63.476%	0.003527
$10^{-3}$	11.23%	0.000211
$10^{-4}$	1.074%	0.000106

Tabela 2.1: Exemplo 2.8: Taxa de compressão, como porcentagem de células da malha adaptativa, quando comparadas com a malha uniforme do nível mais refinado, e diferença entre a projeção  $\hat{u}$  e a aproximação  $\hat{u}_{\epsilon}$  na norma  $L_1$  para diferentes valores de  $\epsilon$ .

## 2.3 Estrutura de Dados

Para codificarmos as malhas adaptativas e os esquemas MR de maneira eficiente, necessitamos de uma estrutura de dados que seja dinâmica, tenha baixo custo de armazenamento e que permita um rápido acesso aos dados nela contidos. Uma estrutura de dados dinâmica muda de forma e tamanho durante o tempo de execução de um algoritmo. No próximo capítulo, propomos o uso de uma árvore binária para codificação das malhas e esquemas MR baseados em malhas diádicas.

# Capítulo 3

# Malhas Diádicas e sua Representação

Neste capítulo, apresentamos os conceitos de malhas diádicas, o processo de construção destas malhas, inicialmente proposto por Cardoso *et al.* [5, 6]. Revisamos os aspectos importantes da estrutura de árvore binária [31], empregada nas implementações de esquemas baseados em malhas diádicas, como aqueles adotados no presente trabalho.

O interesse principal está em uma estrutura dinâmica de árvore binária, permitindo fácil acesso aos dados nela armazenados. Ao final do capítulo, introduzimos o conceito de malhas  $2^{d}$ -ádicas [20], para uma futura comparação com as malhas diádicas.

# 3.1 Conceitos Básicos

Nesta seção discutimos alguns conceitos necessários para as seções subseqüentes.

### 3.1.1 Subdivisões

Uma subdivisão de dimensão d é uma coleção C de subconjuntos disjuntos de  $\mathbb{R}^d$  (os elementos da subdivisão), com as seguintes propriedades:

- (1) cada elemento  $c \in \mathcal{C}$  é um conjunto homeomorfo a uma bola aberta  $\mathbb{B}^k$ ,  $0 \leq k \leq d$ .
- (2) a fronteira de cada elemento c é a união de um número finito de elementos de C.

Em particular, os elementos de uma subdivisão com dimensão k = 0, 1, 2, 3 que são chamados, respectivamente, de vértices, arestas, paredes e blocos da subdivisão. Em qualquer espaço  $\mathbb{R}^d$ , os elementos de dimensão d são também chamados de células da subdivisão. Os elementos da subdivisão C que compõem a fronteira de um elemento csão ditos as faces de c. Se um elemento c tem dimensão k, suas faces de dimensão k - 1são as facetas. Note que os elementos de uma subdivisão C do conjunto R formam uma partição para R, o domínio da subdivisão.

#### 3.1.2 Caixas

Vamos numerar os eixos de coordenadas de  $\mathbb{R}^d$  de 0 a d-1, de modo que um ponto  $x \in \mathbb{R}^d$ tem coordenadas  $x_0, x_1, \dots, x_{d-1}$ .

Uma caixa k-dimensional  $c \in \mathbb{R}^d$  é o produto cartesiano de d intervalos  $I_0, I_1, \dots, I_{d-1}$  $c = \prod_{i=0}^{d-1} I_i$ , em que k são intervalos abertos limitados e os demais d-k desses intervalos são intervalos fechados degenerados a um ponto.

Dizemos que o eixo *i espeta* a caixa *c* se o intervalo  $I_i$  é um ponto, e *varre* a caixa se  $I_i$  não é degenerado. A Figura 3.1 mostra um exemplo no  $\mathbb{R}^2$ .



Figura 3.1: Três caixas  $c_1 = \{4\} \times (1,5), c_2 = (1,4) \times (1,5)$  e  $c_3 = (4,7) \times (1,5)$ . O eixo 0 espeta e o eixo 1 varre a caixa unidimensional  $c_1$ ; ambos os eixos varrem as caixas  $c_2$  e  $c_3$ .

## 3.1.3 Árvores Binárias

**Definição 3.1.** Uma árvore binária  $\acute{e}$  um conjunto T de elementos denominados nós, estruturado da seguinte forma:

- 1- ou o conjunto T é vazio (árvore vazia);
- 2- ou existe um nó especial r, chamado raiz de T, e os demais nós estão divididos em duas árvores binárias  $T_e \ e \ T_d$  disjuntas, chamadas respectivamente, subárvore esquerda e subárvore direita de r.
- A Figura 3.2 ilustra o conceito de árvore binária.



Figura 3.2: Árvore binária.

Em qualquer nó da árvore, as raízes das subárvores esquerda e direita, quando existem, são chamadas de *filho esquerdo* e *filho direito*, respectivamente, desse nó. Os nós que não têm filhos são chamados de *folhas*. Neste trabalho consideraremos apenas árvores binárias não vazias ditas *árvores de tipo*  $\{0, 2\}$  nas quais cada nó ou não tem filhos ou tem 2 filhos.

#### Árvores Completas, Altura, Profundidade, Nível

Cada nó da árvore binária possui uma *profundidade*. Por definição, a profundidade da raiz é 0, e cada filho de um nó de profundidade  $\ell$  tem profundidade  $\ell + 1$ . O *nível*  $\ell$  de

uma árvore binária é o conjunto de nós com profundidade  $\ell$ . Esse nível pode conter, no máximo,  $2^{\ell}$  nós. Na Figura 3.3 destacamos os nós do nível 4.



Figura 3.3: Nível 4 de uma árvore binária.

A *altura* de uma árvore binária é definida como a quantidade de níveis não vazios acrescida de uma unidade; ou seja, é o menor  $\ell$  tal que o nível  $\ell$  é vazio.

Uma árvore binária é *completa* se todos os nós folha tem a mesma profundidade. Um exemplo pode ser visto na Figura 3.4. Caso contrário dizemos que a árvore binária é *incompleta*. Veja um exemplo na Figura 3.5.



Figura 3.4: Árvore binária completa.



Figura 3.5: Árvore binária incompleta.

#### Índices dos Nós

A cada nó n de uma árvore atribuímos um número inteiro positivo I(n), denominado *índice* do nó. A atribuição dos índices é feita da seguinte forma recursiva: para todo nó n

$$I(n) = \begin{cases} 1 & \text{se } n \text{ \'e raiz,} \\ 2I(p) & \text{se } n \text{ \'e filho esquerdo de } p, \\ 2I(p) + 1 & \text{se } n \text{ \'e filho direito de } p. \end{cases}$$
(3.1)

A Figura 3.6 ilustra esta enumeração para uma árvore incompleta. Observe que o índice de um nó depende apenas do caminho da raiz até ele e independe do fato da árvore ser completa ou não.



Figura 3.6: Distribuição de índices na árvore binária incompleta.

Observe também que cada nó do nível  $\ell$  tem índice entre  $2^{\ell} e 2^{\ell+1} - 1$ , inclusive. A forma sistemática de calcular a profundidade a partir do índice I de um nó é

$$\ell = \lfloor \log_2 I \rfloor.$$

A justificativa para esta escolha é que todo índice I está entre  $2^r$  e  $2^{r+1}$ , para algum r, isto é,  $2^r \leq I < 2^{r+1}$ . Como o log é uma função crescente, vale a relação  $r \leq \log_2 I < r+1$ . Deste modo, a profundidade é tomada como sendo  $\ell = r$ .

# 3.2 Malhas Diádicas

Nesta seção discutimos o conceito de *malhas diádicas*, que podem ser *regulares* ou *irregulares*. Estas oferecem uma divisão do domínio dos sinais a serem discretizados, respectivamente uniforme e heterogênea.

## 3.2.1 Construção da Malha

Nesta seção estabelecemos o processo de construção das malhas diádicas regulares. Para tal, consideramos uma caixa d-dimensional  $R \subset \mathbb{R}^d$ , denominada célula raiz. Uma malha

diádica infinita é uma hierarquia de subdivisões  $\mathcal{G} = \mathcal{G}_0 \cup \mathcal{G}_1 \cup \cdots$  obtidas a partir da célula raiz por bisecções sucessivas das células de maneira cíclica. Mais precisamente, para passar do nível  $\ell$  da hierarquia para o nível  $\ell + 1$ , todas as células são cortadas ao meio por planos perpendiculares ao eixo ( $\ell \mod d$ ). Este processo é ilustrado na Figura 3.7 para o caso bidimensional e na Figura 3.8 para o caso tridimensional.

Note que cada nova divisão é ortogonal a um eixo de coordenadas  $i = (\ell \mod d)$ distinto, acarretando em um refinamento gradativo da malha. As duas metades de uma célula da malha são chamadas de *células filhas*, que têm a metade do volume da célula original (sua *célula mãe*). O mesmo ocorre com todo elemento de dimensão menor que d que é varrido pelo eixo *i*.

Como conseqüência do refinamento gradual (diádico), a quantidade de células dobra ao passar do nível  $\ell$  para o nível  $\ell + 1$ .

Ao longo de cada eixo i, o número de células é  $2^{\nu(\ell,i)}$ , em que

$$\nu(\ell, i) = \left\lfloor \frac{\ell}{d} \right\rfloor + \begin{cases} 1 & \text{se } (\ell \mod d) > i, \\ 0 & \text{se } (\ell \mod d) \le i. \end{cases}$$
(3.2)



Figura 3.7: Os 8 primeiros níveis da malha diádica bidimensional.

A *profundidade* de uma célula c da malha diádica  $\mathcal{G}$  é o índice  $\ell$  do nível a qual ela pertence.

Para fins computacionais é muito útil considerar a célula raiz  $R = [0, 1] \times [0, 1]$  com a topologia do toro  $\mathbb{T}^2$ . Isto é, como indicado na Figura 3.9, identificamos os lados opostos de R. Com esta convenção, todas as células de mesmo nível da malha diádica



Figura 3.8: Os 8 primeiros níveis da malha diádica tridimensional.



Figura 3.9: Toro  $\mathbb{T}^2$  como subconjunto do plano  $\mathbb{R}^2$  com lados opostos identificados (esquerda) e uma representação topologicamente equivalente de  $\mathbb{T}^2$  como superfície no espaço  $\mathbb{R}^3$  (direita).

são perfeitamente equivalentes; em particular, toda célula tem 2d células adjacentes (que compartilham facetas). Da mesma forma, todos os elementos do mesmo nível  $\ell$  que são varridos e espetados pelos mesmos eixos são equivalentes em termos de número e arranjo de elementos vizinhos. Em particular, todo elemento de dimensão k está na fronteira de  $2^{d-k}$  elementos distintos.

#### Malhas Diádicas Finitas

Na prática usamos malhas diádicas finitas, em que o processo termina depois de um número finito de divisões.

**Definição 3.2.** Uma malha diádica finita é um subconjunto finito G da malha diádica  $\mathcal{G}$ 

- (a) a mãe de todo elemento c ∈ G está em G, exceto quando c é a célula raiz ou uma de suas faces;
- (b) os filhos esquerdo e direito de todo elemento c ∈ G estão ou ambos presentes em G,
   ou ambos ausentes de G;
- (c) se c é uma célula de G, todas as faces de c estão em G;
- (d) todo elemento de G que não é célula é face de alguma célula de G.

Note que G não é uma subdivisão no sentido da definição apresentada no início da seção 3.2.1, mas cada nível de G é.

Um elemento c de uma malha diádica finita G é um *elemento folha* se não sofreu divisão, isto é, seus elementos filhos não estão em G. Verifica-se que os elementos folha de uma malha diádica finita são uma subdivisão cujo domínio é a célula raiz. Uma malha diádica é *completa* se todas as células folha têm a mesma profundidade.

Exemplos de malhas diádicas completas podem ser vistos na Figura 3.7 para o caso bidimensional e na Figura 3.8 para o caso tridimensional.

Se as células folha aparecem em níveis diferentes da malha, dizemos que a malha é *incompleta* ou *irregular*. Na Figura 3.10 exibimos as folhas de uma malha diádica incompleta.



Figura 3.10: Células folha de uma malha diádica incompleta.

Uma malha diádica G pode ser associada a uma árvore binária do tipo  $\{0, 2\}$ . Para este fim, introduzimos uma relação parcial de ordem nas células da malha. Mais precisamente, sejam a, b células de um mesmo nível de G e sejam  $a_i e b_i$  as projeções ortogonais das células a e b, respectivamente, sobre o eixo i. Dizemos que a é menor que b na direção i, denotado por  $a <_i b$ , se sup  $a_i \leq \inf b_i$ . Também dizemos que a é igual a b no eixo i, denotado por  $a =_i b$ , se  $a_i = b_i$ . Veja exemplo na Figura 3.11. Em geral, usamos a relação parcial de ordem  $<_i$  apenas entre elementos do mesmo nível. Em particular, com esta relação, podemos distinguir o *filho menor* a e o *filho maior* b de um elemento c, pela condição  $a <_i b$ , em que i é o eixo perpendicular ao hiperplano de corte.



Figura 3.11: Relação de ordem entre células. Neste caso,  $a =_0 b$  mas  $a <_1 b$ .

Podemos usar  $<_i$  entre vértices e células. Em particular, definimos o *vértice inferior* de uma célula c como sendo o vértice v que satisfaz  $v \leq_i c$ , para todo i em  $\{0, \dots, d-1\}$ . Nesta tese, para simplificar, supomos que o vértice inferior da célula raiz está na origem dos eixos coordenados de  $\mathbb{R}^d$ .

#### Índices das Células

A cada célula da malha diádica  $\mathcal{G}$  atribuímos um índice inteiro como segue. Por definição, o índice da célula raiz é 1. As filhas menor e maior da célula de índice I possuem índices  $2I \in 2I + 1$ , respectivamente.

A distribuição dos índices na malha diádica é ilustrada na Figura 3.12 e na Figura 3.13. Podemos então identificar uma malha diádica finita G com uma árvore binária do tipo  $\{0,2\}$ , através dos índices. Especificamente, cada nó da árvore binária com índice I



Figura 3.12: Distribuição dos índices nos 3 primeiros níveis de uma malha diádica bidimensional.

representa a célula da malha diádica G que tem índice I. A Figura 3.13 ilustra essa correspondência.



Figura 3.13: Índices na árvore e na malha.

#### Posição e Tamanho de uma Célula

A posição de uma célula c da malha diádica  $\mathcal{G}$  com profundidade  $\ell$  é um vetor de inteiros  $\delta(c) = (\delta_0, \delta_1, \dots, \delta_{d-1})$ , sendo que  $\delta_i$  é a quantidade de células a de  $\mathcal{G}$  com a mesma profundidade  $\ell$  tais que  $a <_i c$  e  $a =_j c$  para todo  $j \neq i$ . Verifica-se que a posição  $\delta_i$  varia entre 0 e  $2^{\nu(\ell,i)} - 1$ .

Dado o índice de uma célula podemos determinar tanto sua posição quanto a profundidade. Para isto, procedemos da seguinte forma:

- 1. Calculamos a representação binária de I.
- 2. Eliminamos o dígito binário mais significativo. O número de dígitos restante é a profundidade  $\ell$ .

- 3. Separamos os dígitos restantes, da esquerda para a direita, em d grupos, atribuindo cada dígito a um grupo, alternadamente. Isto é, o dígito na posição k, contando de 0, é atribuído ao grupo ( $k \mod d$ ). Os dígitos de cada grupo i na mesma ordem formam a representação binária de  $\delta_i$ .
- 4. Transformamos as representações de cada  $\delta_i$  para a base decimal.

Por outro lado, dados o vetor posição  $\delta(c)$  e a profundidade  $\ell$  de uma célula é possível determinar a célula e seu índice.

- 1. Determinamos quantos bits binários  $\nu(\ell, i)$  são necessários para representar cada entrada  $\delta_i$  do vetor posição  $\delta_i$ , dado pela fórmula (3.2);
- 2. A seguir, montamos as representações binárias de  $\delta_i$ , completando com zeros à esquerda até obter  $\nu(\ell, i)$  dígitos;
- 3. Consideramos o primeiro dígito do índice I igual a 1
- 4. Da esquerda para a direita, tomamos um bit de cada  $\delta_i$ , alternadamente; ou seja, o bit k do índice I é o bit  $\lfloor k/d \rfloor$  de  $\delta_{(k \mod d)}$  contando de 0.
- 5. Convertemos o índice I para decimal.

**Exemplo 3.1.** Suponha que d = 2 e que queremos descobrir a posição  $\delta$  da célula de índice I = 147. Na base 2, temos I = 10010011. Note que, nesta representação binária, contamos 7 bits depois do bit mais significativo, isso quer dizer que a profundidade desta célula é 7.

Para montar o vetor  $\delta(c)$  fazemos o seguinte:

- Eliminamos o dígito 1 mais significativo (marcado com '\*').
- Alternadamente escolhemos os dígitos para  $\delta_0$  (marcados com '0') e  $\delta_1$  (marcados com '1'):  $I = \stackrel{*01010010}{10010011_2}$ . Ou seja, temos  $\delta_0 = 0101$  e  $\delta_1 = 001$ .
- Convertendo para base decimal, temos  $\delta_0 = 5$  e  $\delta_1 = 1$ , ou seja  $\delta(c) = (5, 1)$ .

Por outro lado, dados o vetor posição  $\delta(c) = (5, 1)$  e a profundidade  $\ell = 7$ , para recuperar o índice da célula c calculamos  $q_0 = 3 + 1$  e  $q_1 = 3 + 0$ . Escrevemos a componente  $\delta_0$  do vetor posição  $\delta$  com 4 dígitos binários e a componente  $\delta_1$  com 3 dígitos, obtendo  $\delta(c) = (0101, 001)$ . Intercalando esses dígitos, obtemos  $I = \stackrel{*010101010}{100100112} = 147_{10}$ 

# **3.3** Malhas 2<sup>d</sup>-ádicas

Nesta seção apresentamos a relação entre as malhas diádicas d-dimensionais e as malhas  $2^{d}$ -ádicas, freqüentemente usadas em aproximações multiescala.

Como na seção anterior, consideramos uma caixa d-dimensional  $R \subset \mathbb{R}^d$ .

A malha  $2^d$ -ádica é uma hierarquia infinita de subdivisões  $\mathcal{H} = \mathcal{H}_0 \cup \mathcal{H}_1 \cup \cdots$  obtidas dividindo perpendicularmente cada célula em  $2^d$  células filho iguais por d hiperplanos ortogonais a cada um dos eixos coordenados. No caso bidimensional, a malha  $2^d$ -ádica é conhecida como quad-grid e no caso tridimensional como oct-grid.

Uma malha  $2^{d}$ -ádica tem  $2^{d\ell}$  células no nível  $\ell$ . Pode-se ver que o mesmo nível  $\ell$  de uma malha  $2^{d}$ -ádica é idêntico ao nível  $d\ell$  da malha diádica. Mais genericamente, uma malha  $2^{d}$ -ádica finita irregular coincide com uma malha diádica finita cujas folhas têm profundidades múltiplas de d.

Nesse sentido, as malhas diádicas são mais gerais que as malhas  $2^{d}$ -ádicas para  $d \ge 2$ , visto que aquelas permitem refinamentos intermediários. Vale observar que, para o caso unidimensional, os dois tipos de malhas são idênticos.

# Capítulo 4

# **Splines Diádicos**

Em capítulos anteriores, apresentamos os elementos da teoria de aproximação. Vimos que precisamos escolher, por exemplo, espaço aproximante e a base para tal espaço.

Nesse capítulo, síntese do trabalho de Cardoso *et al.* [5, 6], discutimos o espaço dos splines multilineares em malhas diádicas sobre uma região  $R \subset \mathbb{R}^d$ . Provamos que uma base para tal espaço é obtida pelo produto tensorial dos elementos da base do espaço dos splines lineares definidos sobre a partição de um intervalo, apresentado na Seção 2.2.

Fazemos uma revisão sobre o espaço das funções splines multilineares em malhas diádicas e construímos diferentes bases, tanto em malhas diádicas regulares quanto em malhas irregulares de multinível.

Apresentamos o algoritmo para a construção da aproximação das funções nestes espaços e apresentamos alguns exemplos de interesse que ilustram as diferentes bases, tanto em um quanto em vários níveis, bem como as aproximações de algumas funções usando tais bases.

## 4.1 Splines em Malhas Diádicas Regulares

Para simplificar, supomos nessa seção qua a célula raiz é o hipercubo  $R = [0, 1]^d$ . Por mudança de escala em cada eixo, os conceitos e resultados apresentados valem para malhas diádicas cuja célula raiz é qualquer caixa fechada d-dimensional de  $\mathbb{R}^d$ . Seja G uma malha diádica sobre R. Como observado na Seção 3.2.1, G é uma subdivisão da caixa R e, portanto, uma partição de R. Podemos considerar G como o produto de partições  $C_0, C_1, \dots, C_{d-1}$ , em que cada  $C_i$  é uma subdivisão diádica regular do intervalo  $[0, b_i]$ . Ou seja, cada elemento c de G é o produto cartesiano  $c_0 \times c_1 \times \cdots \times c_{d-1}$ , em que cada  $c_i$  é um elemento de  $C_i$ .

Seja  $\mathbb{P}^1$  o espaço de todos os polinômios reais de variável real com grau menor ou igual a 1. Define-se

$$S^{1}(G) = \{ f \in C^{0}(R) : f|_{c}(x) = \prod_{i=0}^{d-1} g_{i}(x_{i}), g_{i} \in \mathbb{P}^{1} \}.$$

Em outras palavras, o espaço  $S^1(G)$  é formado por funções que, quando restritas a cada célula de G, são descritas por polinômios de grau menor ou igual 1 em cada variável. A restrição a uma célula de G da função  $f \in S^1(G)$  é denominada *retalho* de f. Cada função  $f \in S^1(G)$  é denominada de *spline multilinear diádico*.

Nas aplicações, consideramos a condição de contorno periódica, impondo a continuidade também na fornteira da caixa R, respeitando a suposição de topologia toroidal (Seção 3.2.1).

## 4.2 Bases para Splines

Para trabalhar com o espaço  $S^1(G)$ , é interessante escolher uma base para o mesmo. Neste capítulo usaremos bases constituídas de elementos chamados funções tendas (ou funções chapéu).

O suporte de cada tenda é composto por  $2^d$  células de igual profundidade, agrupadas em torno de um *vértice central* v, e das faces compartilhadas por essas células. No vértice central, a tenda vale 1, e cai linearmente ao longo de cada eixo até assumir o valor 0, na fronteira do suporte.

Cada tenda da base pode ser obtida por devidas translações e mudança de escala (contração) de uma *tenda mestre*  $\Lambda^d$ , dada pelo produto tensorial  $\Lambda^d(x) = \prod_{i=0}^{d-1} \Lambda(x_i)$ ,

em que  $\Lambda$  é a tenda mestre unidimensional

$$\Lambda(z) = \begin{cases} 1-z & \text{se } 0 \le z \le 1, \\ 0 & \text{se } |z| > 1, \\ 1+z & \text{se } -1 \le z < 0. \end{cases}$$
(4.1)

Na Figura 4.1, exibimos as tendas mestre nos casos uni e bidimensionais.



Figura 4.1: Função tenda, em uma e duas dimensões.

A profundidade de uma tenda  $\tau$  é, por definição, a profundidade das células em seu suporte. Cada tenda pode ser identificada, unicamente, pela célula superior do seu suporte (ver definição da relação parcial de ordem para células na Seção 3.2.1). Então, seja c uma célula folha da malha diádica regular G com posição  $\delta(c)$ . A fim de construir a tenda associada a c, construímos uma matriz diagonal Q cujo elemento  $Q_{ii}$  é a quantidade de células na direção i, ou seja,

$$Q_i = 2^{\nu(\ell,i)},$$

em que  $\nu(\ell, i)$  é dado em (3.2). Determinamos também a posição  $\delta(c) = (\delta_0, \dots, \delta_{d-1})$  da célula, como na Seção 3.2.1. Assim, a função tenda da célula c é dada por

$$\tau_c(x_0,\cdots,x_{d-1}) = \Lambda^d(Qx - \delta(c)) = \prod_{i=0}^{d-1} \Lambda(Q_i x_i - \delta_i).$$

$$(4.2)$$

Na translação da tenda mãe, é importante observar a topologia toroidal de R. Em particular, se  $\delta_i = 0$ , o suporte da tenda inclui células com  $\delta_i = Q_i - 1$ . Além disso, se  $Q_i = 1$ , a tenda (e qualquer spline em  $S^1(G)$ ) se dobra sobre si mesma e reduz-se a uma função constante na direção i.

## 4.2.1 Base Uniforme de Tendas

Uma maneira de construir uma base para  $S^1(G)$ , em que G tem profundidade L, é tomar todas as tendas de mesma profundidade L. Os elementos (tendas) dessa base são apresentadas na Figura 4.2.

Precisamos mostrar que o conjunto das tendas  $\{\tau_c : c \in G\}$  de fato é uma base para  $S^1(G)$ . Para isso, usamos o fato de que o conjunto

$$\Phi_i = \{\phi_j^i, j = 0, \cdots, n_i\}, \quad i = 0, \cdots, d-1,$$
(4.3)

em que  $\phi_j^i$  é definida por

$$\phi_i^i(x) = \Lambda(Q_i x_i - \delta_i), \tag{4.4}$$

é uma base para  $S^1(C_i)$ . Definimos

$$\tau_c(x) = \prod_{i=0}^{d-1} \phi^i_{\delta_i}(x_i), \quad \delta_i = 0, \cdots, Q_i - 1$$
(4.5)

em que  $(\delta_0, \dots, \delta_{d-1})$  é a posição de c. Observe que, para cada célula c, com vértice inferior  $v_c$ , a tenda  $\tau_c$  satisfaz

$$\tau_c(v) = \begin{cases} 1, & v = v_c, \\ 0, & \text{caso contrário,} \end{cases}$$

em que v é um vértice qualquer das células de G.

Assim, para tendas em um único nível, definimos a interpolação

$$\tilde{u}(x) = \sum_{c \in G} u(v_c) \tau_c(x).$$

Verifica-se que  $\tilde{u}(x) = u(x)$ , para toda função  $u \in S^1(G)$ . Consequentemente, vale o resultado.

**Teorema 4.1.** Seja G uma malha diádica uniforme. Então  $S^1(G) = \operatorname{span}\{\tau_c : c \in G\}$ .



Figura 4.2: Os elementos da base do espaço  $S^1(G)$ , em que G é uma malha diádica regular com profundidade 4. Cada tenda  $\tau_c$  é representada por suas curvas de nível.

## 4.2.2 Base Hierárquica de Tendas

Denotamos por  $G_{\ell}$  a malha diádica uniforme do nível  $\ell \in \mathcal{V}(\ell)$  o conjunto dos vértices das células de  $G_{\ell}$ . Outra forma de construir uma base para  $S^1(G_L)$  é escolher, para cada vértice  $v \in \mathcal{V}(L)$ , a tenda centrada em v, de menor profundidade possível [6, 5]. Ou seja, tal processo equivale a escolher a tenda  $\tau_c^{\ell+1}$  associada à célula  $c \in G_{\ell+1}$  cujo vértice inferior  $v_c$  aparece pela primeira vez na malha. Precisamente, se v é um vértice de profundidade  $\ell + 1$ , isto é,  $v \in \mathcal{V}(\ell + 1) - \mathcal{V}(\ell)$ , consideramos a tenda  $\tau_c^{\ell+1}$ , em que  $c \in G_{\ell+1}$  é a célula tendo como vértice inferior  $v_c = v$ . Se v é o vértice com profundidade igual a 0, associamos a função  $\tau_c^0 = 1$ , em que c é a célula raiz.

Assim, temos dois tipos de representação para uma uma função de  $S^1(G_L)$ , em um ou

em vários níveis, dados respectivamente por

$$u = \sum_{c \in G_L} u(v_c) \tau_c^L$$
  
=  $\sum_{c \in G_0} u(v_c) \tau_c^0 + \sum_{\ell=0}^{L-1} \sum_{c \in D(\ell)} d_c^\ell \tau_c^{\ell+1},$  (4.6)

em que  $D(\ell) = \{c \in G_{\ell+1} : v_c \in \mathcal{V}(\ell+1) - \mathcal{V}(\ell)\}$ . Verifica-se que

$$d_c^{\ell} = u(v_c) - \frac{u(v_e) + u(v_d)}{2},$$

em que  $v_e \in v_d$  são os vértices da aresta de uma célula de  $G_\ell$ , perpendicular ao eixo de subdivisão ( $\ell \mod d$ ), tendo  $v_c$  como ponto médio.

Na representação em um nível, utilizamos a base  $B(G_L) = \{\tau_c^L : c \in G_L\}$ , enquanto que na representação em multinível a base é

$$B_{MR}(G_L) = \{\tau_c^0 : c \in G_0\} \cup \left\{\bigcup_{\ell=0}^{L-1} \{\tau_c^{\ell+1} : c \in D(\ell)\}\right\}.$$

Na Figura 4.3, exibimos os elementos da base hierárquica para o espaço  $S^1(G_4)$ , organizadas por nível.

### 4.2.3 Adaptatividade

Novamente, a vantagem em usarmos bases hierárquicas está no fato dessas bases possibilitarem uma representação adaptativa da função a ser aproximada. O processo adaptativo não pode ser feito aleatoriamente, precisa ser guiado por algum parâmetro que indique onde e como fazer a adaptatividade.

Como erro de interpolação linear, esperamos que os coeficientes  $d_c^{\ell}$  sejam pequenos em regiões onde a função a ser aproximada é suave e significativos nas regiões onde a função tem variação brusca ou apresenta descontinuidades. Assim, os coeficientes  $d_c^{\ell}$  podem ser usados para estimar a regularidade da função a ser aproximada.

Obtemos a adaptatividade mantendo apenas os termos  $d_c^{\ell}$  da expansão em multinível (4.6) que são significativos. Especificamente, dado um parâmetro de tolerância  $\epsilon > 0$ ,



Figura 4.3: Os elementos da base hierárquica para o espaço  $S^1(G)$ , em que G é uma malha diádica regular de profundidade 4, ou 5 níveis.

obtemos a adaptatividade a partir do truncamento dos coeficientes  $d_a^\ell,$ isto é, tomamos

$$\tilde{d}_c^\ell = \left\{ \begin{array}{ll} 0 & \mathrm{se} \quad |d_c^\ell| \leq \epsilon, \\ \\ d_c^\ell & \mathrm{se} \quad |d_c^\ell| > \epsilon. \end{array} \right.$$

Assim, define-se a aproximação adaptativa

$$u_{\epsilon} = \sum_{c \in G_0} u(v_c)^0 \tau_c^0 + \sum_{\ell=0}^{L-1} \sum_{c \in D(\ell)} \tilde{d}_c^{\ell} \tau_c^{\ell+1}.$$

## 4.2.4 Algoritmo de Interpolação Usando Base de Tendas

Escolhidas as tendas que formam a base do espaço de aproximação, resta determinar uma forma eficiente de calcular a interpolação nesse espaço de aproximação.

Sejam G uma malha diádica d-dimensional, B uma base de tendas e u uma função definida sobre G. O algoritmo Interpola, descrito por Cardoso *et al.* [5], encontra uma combinação linear

$$\tilde{u} = \sum_{j=0}^{n} q_j \tau_j$$

dos elementos de B que interpola a função u no conjunto de pontos V que são centros das tendas de B(G).

#### Interpola(B, u)

Seja V o conjunto dos vértices centrais das tendas de B;

Ordene *B* em ordem de profundidade crescente, obtendo  $\tau_0, \dots, \tau_n$ ;

for  $j = 0 \rightarrow n$  do

Seja v o vértice central de  $\tau_i$ ;

if 
$$v \notin V$$
 then  
 $q_j \leftarrow 0$ ;  
else  
 $q_j \leftarrow u(v) - \left(\sum q_i \tau_i : i < j, v \in Dom(\tau_i)\right)$ ;  
 $V \leftarrow V - \{v\}$ ;  
end if  
end for  
(4.7)

## 4.3 Exemplo

Nesta seção, usamos o algoritmo adaptativo de *splines* diádicos para representação da imagem de uma paisagem. Consideramos a imagem do castelo de Lichtenstein, em escala de cinza variando entre 0 (preto) e 255 (branco), com  $512 \times 512$  pixels, apresentada na Figura 4.4.

Na Figura 4.5, à esquerda, vemos a malha adaptativa relativa ao parâmetro de truncamento  $\epsilon = 20$ . Neste caso, a malha adaptativa possui 72324 células folha, correspondendo



Figura 4.4: Imagem do castelo de Lichtenstein, com  $512 \times 512$  pixels em escala de cinza.

a 27,58% do número de células da malha uniforme original, isto é, uma compressão de 72,42%. Observamos que a adaptatividade ocorre de maneira a preservar a posição do castelo, nuvens, mata e regiões que necessitam de mais detalhes. A imagem recuperada, após o truncamento, por meio da interpolação por tendas é mostrada na Figura 4.5, à direita. Observamos que, mesmo com a alta taxa de compressão, ainda obtemos uma imagem boa, em que, por exemplo, os efeitos degradê ainda são preservados.



Figura 4.5: À esquerda, malha adaptativa relativa ao parâmetro de truncamento  $\epsilon = 20$ . À direita, imagem recuperada.

Já na Figura 4.6, à esquerda, vemos a malha adaptativa relativa ao parâmetro de truncamento  $\epsilon = 10$ . Neste caso, o número de células folha na malha adaptativa é de 101557 células, correspondendo a 38,74% do número de células da imagem uniforme, isto é, uma compressão de 61,26%. Observamos que o esquema adaptativo mantém a malha refinada nas regiões que necessitam de mais detalhes. A imagem recuperada, após o truncamento, é mostrada na Figura 4.6, à direita. Com a taxa de compressão um pouco menor que na análise anterior, observamos uma melhora na nitidez da imagem recuperada.



Figura 4.6: À esquerda, malha adaptativa relativa ao parâmetro de truncamento  $\epsilon=10.$ À direita, imagem recuperada.
# Capítulo 5

# Análise MR para Médias Celulares em Malhas Diádicas: Aspecto Discreto

Como descrito no Capítulo 2, esquemas MR permitem a representação de dados em vários níveis de escala. Para que seja possível navegar nos diferentes níveis, utilizamos operadores que relacionam informações em diferentes escalas de resolução. Esses operadores têm papel fundamental no desenvolvimento do assunto.

Tradicionalmente, esquemas de MR são construídos para dados amostrados em malhas  $2^d$ -ádicas. No presente capítulo, propomos e discutimos esquemas de MR para discretizações por médias celulares em malhas diádicas, em qualquer dimensão. Especificamente, consideramos esquemas de MR de Haar (com previsão exata para constantes) e esquemas de ordem superior. Mostramos que os operadores de previsão para as médias celulares em 1D, exatos para polinômios de grau menor ou igual a 2s, em que s é um número inteiro positivo, geram fórmulas de previsão exatas para as médias celulares em uma malha diádica d-dimensional, sendo exatas para polinômios de grau menor ou igual a 2s em cada uma das variáveis. Para o caso bidimensional, fazemos uma comparação com os tradicionais esquemas MR para quad-grids.

## 5.1 Definições e Notações Preliminares

Recordamos algumas notações introduzidas no Capítulo 3. Consideramos uma hierarquia de malhas diádicas regulares  $G_{\ell}$  em  $\mathbb{R}^d$ ,  $\ell = 0, 1, \dots, L$ . Denotamos por R a célula raiz, que é uma caixa d-dimensional em  $\mathbb{R}^d$ . A malha  $G_{\ell+1}$  é obtida por meio da divisão de cada célula de  $G_{\ell}$  por um hiperplano perpendicular ao eixo  $i = (\ell \mod d)$ . Exemplos destas hierarquias de malhas podem ser vistos na Figura 3.7, para o caso bidimensional e na Figura 3.8, para o caso tridimensional.

As seguintes notações são utilizadas:

- A profundidade de uma célula da malha diádica  $G_{\ell} \in \{0, \dots, L\}$ .
- $\mathcal{K}(\ell)$  é conjunto de posições das células de profundidade  $\ell$ .
- $c_{\alpha}^{\ell}$  é a célula da malha diádica  $G_{\ell}$ , com índice  $\alpha$  e profundidade  $\ell$ .
- $c_{\alpha_L}^{\ell+1}$  célula filho menor da célula  $c_{\alpha}^{\ell}$  (ver Seção 3.2.1).
- $c_{\alpha_R}^{\ell+1}$  célula filho maior da célula  $c_\alpha^\ell.$

Sabemos que uma malha diádica  $G_{\ell}$  forma uma partição para a caixa R. Usando as notações acima,

$$\overline{R} = \bigcup_{\alpha \in \mathcal{K}(\ell)} \overline{c_{\alpha}^{\ell}}, \ e \ c_{\alpha}^{\ell} \cap c_{\beta}^{\ell} = \emptyset, \ se \ \alpha \neq \beta.$$
(5.1)

Seja  $V = \mathbb{L}^2(R)$  o espaço das funções a serem aproximadas. Denotamos por  $u^{\ell}$  o vetor com as informações no nível de resolução  $\ell$  formado com as médias celulares de  $u \in V$  na malha diádica  $G_{\ell}$ 

$$u_{\alpha}^{\ell} = \frac{1}{|c_{\alpha}^{\ell}|} \int_{c_{\alpha}^{\ell}} u(x) \mathrm{d}x.$$

Sendo assim, temos o operador de *discretização* por médias celulares  $\mathcal{D}^{\ell}: V \to X^{\ell}$  que é definido por

$$\mathcal{D}^{\ell} u = u^{\ell} = (u^{\ell}_{\alpha}), \tag{5.2}$$

em que  $X^{\ell}$  é o espaço vetorial real de dimensão  $\#\mathcal{K}(\ell)$ .

Para que seja possível a representação de uma função por suas médias celulares em vários níveis de escala e o relacionamento dessas informações nas diferentes escalas, são definidos operadores interescala que desempenham um papel fundamental no desenvolvimento do assunto. Quando queremos passar de um nível mais fino para um mais grosseiro, usamos o operador de restrição  $P_{\ell+1}^{\ell}$ . Para passar do nível mais grosseiro para o mais fino, aplicamos o operador de previsão  $P_{\ell}^{\ell+1}$ .

Como vimos no Capítulo 3, os elementos das malhas diádicas estão bijetivamente relacionados com os nós de árvores binárias. Sendo assim, os operadores de previsão e restrição são aplicados sobre dados armazenados na estrutura de dados de árvores binárias. Em resumo, os nós do nível  $\ell$  contêm os dados das células da malha diádica de mesmo nível, os quais são usados para prever os dados do nível  $\ell + 1$ . Da mesma forma, os dados contidos nos nós da escala  $\ell + 1$  são usados para calcular os dados na escala  $\ell$ .

Para o caso específico de discretização por médias celulares em malhas diádicas, o operador de projeção (restrição)  $P_{\ell+1}^{\ell}: X^{\ell+1} \to X^{\ell}$  é definido por

$$u_{\alpha}^{\ell} = \frac{u_{\alpha_L}^{\ell+1} + u_{\alpha_R}^{\ell+1}}{2}.$$
(5.3)

O operador de previsão  $P_{\ell}^{\ell+1}: X^{\ell} \to X^{\ell+1}$  deve ser a inversa direita do operador de restrição, isto é,

$$P_{\ell+1}^{\ell} P_{\ell}^{\ell+1} = I^{\ell}.$$

O exemplo mais simples de operador de previsão é o caso de Haar [21], que é dado pelas fórmulas

$$(P_{\ell}^{\ell+1}u^{\ell})_{\alpha_R} = u_{\alpha}^{\ell},$$

$$(P_{\ell}^{\ell+1}u^{\ell})_{\alpha_L} = u_{\alpha}^{\ell}.$$

$$(5.4)$$

Observe que essa previsão fornece uma aproximação  $\tilde{u}^{\ell+1}$  para as médias celulares  $u^{\ell+1}$ , sendo exata para funções constantes. Na Seção 5.3, são considerados operadores de previsão mais complexos, exatos para polinômios de ordem superior.

De posse dos operadores de previsão e restrição, os algoritmos de multirresolução, que relacionam os diferentes níveis de resolução, podem ser definidos.

### 5.2 Algoritmos de MR em Malhas Diádicas

Um operador de previsão utiliza os dados contidos nos nós do nível  $\ell$  para estimar os dados no nível  $\ell + 1$ . Em malhas diádicas, ao passarmos do nível de refinamento  $\ell$  para o nível  $\ell + 1$ , cada célula da malha é dividida em duas células filho. Então é possível obter dois erros de previsão por célula. Para evitar redundâncias, basta guardar o erro de previsão de apenas uma das células filho, por exemplo, do filho maior, conforme relação de ordem ilustrada na Figura 3.11. Isto é, para cada  $c_{\alpha}^{\ell}$  armazenamos apenas

$$d^{\ell}_{\alpha} = u^{\ell+1}_{\alpha_R} - (P^{\ell+1}_{\ell}u^{\ell})_{\alpha_R},$$

denominado coeficiente *wavelet*. De fato, tendo os valores de  $u_{\alpha}^{\ell} \in d_{\alpha}^{\ell}$ , recuperamos  $u_{\alpha_R}^{\ell+1} = d_{\alpha}^{\ell} + \tilde{u}_{\alpha_R}^{\ell+1} \in u_{\alpha_L}^{\ell+1} = 2u_{\alpha}^{\ell} - u_{\alpha_R}^{\ell+1}$ . Sendo assim, procedendo em vários níveis, podemos estabelecer uma bijeção  $u^L \leftrightarrow MR(u^L) = (u^0, d^0, \cdots, d^{L-1})$ , entre os dados  $u^L$ , no nível máximo de resolução da malha, e os dados  $MR(u^L)$  em multirresolução.

Para o algoritmo de MR direto, o dado de entrada é o conjunto de médias celulares da função u no nível L e os dados de saída são os vetores  $d^{\ell}$ ,  $\ell = L - 1, \dots, 0$ , e o vetor  $u^0$ , que é a média de u na caixa R. Precisamente, o algoritmo MR direto é dado por

Para 
$$\ell = L - 1, \cdots, 0$$
  
Para  $\alpha \in \mathcal{K}(\ell)$   
 $u_{\alpha}^{\ell} = \frac{u_{\alpha L}^{\ell+1} + u_{\alpha R}^{\ell+1}}{2},$   
 $\tilde{u}_{\alpha R}^{\ell+1} = \left(P_{\ell}^{\ell+1}u^{\ell}\right)_{\alpha R}$   
 $d_{\alpha}^{\ell} = u_{\alpha R}^{\ell+1} - \tilde{u}_{\alpha R}^{\ell+1}.$ 

Os dados de entrada para o algoritmo MR inverso são as informações contidas nos detalhes  $d^{\ell}$ , para  $\ell = 0, \dots, L - 1 \in u^0$ . O dado de saída é o vetor  $u^L$ . O algoritmo MR

inverso é dado por

Para 
$$\ell = 0, \cdots, L-1$$
  
Para  $\alpha \in \mathcal{K}(\ell)$   
 $\tilde{u}_{\alpha_R}^{\ell+1} = \left(P_\ell^{\ell+1} u^\ell\right)_{\alpha_R}$   
 $u_{\alpha_R}^{\ell+1} = d_\alpha^\ell + \tilde{u}_{\alpha_R}^{\ell+1},$   
 $u_{\alpha_L}^{\ell+1} = 2u_\alpha^\ell - u_{\alpha_R}^{\ell+1}.$ 

#### 5.3 Previsões de Ordem Superior

Nesta seção apresentamos esquemas de previsão para médias celulares em malhas diádicas que sejam exatos para polinômios de grau superior.

#### 5.3.1 Previsão Exata para Polinônimos de Grau $\leq 2$

Para reprodução exata de polinômios de grau menor ou igual a dois, procuramos um operador de previsão da forma

$$\tilde{u}_{\alpha_R}^{\ell+1} = u_{\alpha}^{\ell} + \lambda [u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell}],$$
(5.5)

em que as médias consideradas são referentes à célula  $c_{\alpha}^{\ell}$  e seus dois vizinhos mais próximos na direção  $i = (\ell \mod d)$ , satisfazendo  $c_{\alpha-1}^{\ell} < c_{\alpha}^{\ell} < c_{\alpha+1}^{\ell}$ .

Para ser conservativo,  $\tilde{u}_{\alpha_L}^{\ell+1} = 2u_{\alpha}^{\ell} - \tilde{u}_{\alpha_R}^{\ell+1}$ , o que implica

$$\tilde{u}_{\alpha_L}^{\ell+1} = u_{\alpha}^{\ell} - \lambda [u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell}].$$
(5.6)

Na Figura 5.1, à esquerda, vemos a posição das células no caso unidimensional e à direita, a posição das células no caso bidimensional, com subdivisão na vertical.

#### **Caso Unidimensional**

Para determinar o coeficiente  $\lambda$  na equação (5.5), seguimos algumas etapas. Primeiro calculamos o valor das médias celulares dos monômios  $u = 1, x, x^2$  na célula  $c_{\alpha_R}^{\ell+1}$ . Posteriormente, determinamos as previsões das médias celulares de tais monômios usando a



Figura 5.1: Posição das células  $c_{\alpha}^{\ell}$ ,  $c_{\alpha+1}^{\ell}$ ,  $c_{\alpha-1}^{\ell}$  e  $c_{\alpha_R}^{\ell+1}$  na malha diádica unidimensional(esquerda) e bidimensional com subdivisão na vertical (direita).

equação (5.5). O valor de  $\lambda$  é obtido quando impomos a condição  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$ , em que  $u_{\alpha_R}^{\ell+1}$  é a média de u em  $c_{\alpha_R}^{\ell+1}$  e  $\tilde{u}_{\alpha_R}^{\ell+1}$  é a previsão da média de u na mesma célula  $c_{\alpha_R}^{\ell+1}$ .

Para que a escrita fique mais econômica, supomos que o vértice inferior de  $c_{\alpha_R}^{\ell+1}$  esteja na origem da reta real. Denotamos por  $\Delta x$  o comprimento de  $c_{\alpha}^{\ell}$  e definimos  $\delta x = \frac{\Delta x}{2}$ .

• Para u = 1,

$$u_{\alpha_R}^{\ell+1} = \frac{1}{|c_{\alpha_R}^{\ell+1}|} \int_0^{\delta x} \mathrm{dx} = 1,$$
(5.7)

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{1}{|c_{\alpha}^{\ell}|} \int_{-\delta x}^{\delta x} \mathrm{dx} + \lambda \left[ \frac{1}{|c_{\alpha+1}^{\ell}|} \int_{\delta x}^{3\delta x} \mathrm{dx} - \frac{1}{|c_{\alpha-1}^{\ell}|} \int_{-3\delta x}^{-\delta x} \mathrm{dx} \right] = 1.$$
(5.8)

Portanto, para uma função constante temos  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$ .

• Para u = x, aparece a restrição sobre  $\lambda$ . Neste caso,

$$u_{\alpha_R}^{\ell+1} = \frac{\delta x}{2},\tag{5.9}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = 4\lambda\delta x. \tag{5.10}$$

Logo a relação  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$  é satisfeita se

$$\lambda = \frac{1}{8}.\tag{5.11}$$

• Para  $u = x^2$  não há restrições sobre  $\lambda$ , pois

$$u_{\alpha_R}^{\ell+1} = \frac{\delta x^2}{3},\tag{5.12}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{\delta x^2}{3}.\tag{5.13}$$

Logo, para  $u = x^2$  a relação  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$  é válida para qualquer valor de  $\lambda$ .

Assim, com  $\lambda = \frac{1}{8}$ , como a reconstrução é exata para os monômios  $1, x, x^2$ , concluímos que a reconstrução também é exata para qualquer polinômio de grau menor ou igual a 2. Observa-se que essa fórmula de previsão coincide com a fórmula proposta por A. Harten [26], usando interpolação polinomial de médias celulares em malhas diádicas unidimensionais.

#### Caso Bidimensional

Basta considerar o caso em que a subdivisão da célula é feita perpendicularmente ao eixo 0. De fato, para refinamento por uma reta perpendicular ao eixo 1, o resultado segue por simetria. Assim, sem perda de generalidade, assume-se que  $c_{\alpha_L}^{\ell+1}$  e  $c_{\alpha_R}^{\ell+1}$  estão separadas pelo eixo 1 e que o vértice inferior de  $c_{\alpha_R}^{\ell+1}$  está na origem do espaço euclidiano  $\mathbb{R}^2$ , ver Figura 5.1. Denota-se por  $\Delta x_i$  o comprimento da aresta de  $c_{\alpha}^{\ell}$  na direção *i*, em que i = 0, 1, e define-se  $\delta x_i = \frac{\Delta x_i}{2}$ .

Para u=1,tem-se $u_{\alpha_R}^{\ell+1}=u_\alpha^\ell=u_{\alpha+1}^\ell=u_{\alpha-1}^\ell=1$ e

$$\tilde{u}_{\alpha_R}^{\ell+1} = u_{\alpha}^{\ell} + \lambda [u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell}] = 1.$$

Portanto, para uma função u = 1 tem-se  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$ .

Considerando  $u = x_1$ , tem-se  $u_{\alpha_R}^{\ell+1}$  e  $\tilde{u}_{\alpha_R}^{\ell+1}$  são dados por

$$u_{\alpha_R}^{\ell+1} = \frac{\Delta x_1}{2},\tag{5.14}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{\Delta x_1}{2}.$$
(5.15)

Logo, para  $u = x_1$ , tem-se  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$ .

Analogamente, se  $u = x_1^2$ , tem-se

$$u_{\alpha_R}^{\ell+1} = \frac{\Delta x_1^2}{3},\tag{5.16}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{\Delta x_1^2}{3}.$$
(5.17)

Logo, para  $u = x_1^2$ , tem-se  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$ .

Para  $u = x_0$ , aparece a primeira restrição sobre  $\lambda$ . Neste caso,

$$u_{\alpha_R}^{\ell+1} = \frac{\delta x_0}{2},$$
 (5.18)

$$\tilde{u}_{\alpha_R}^{\ell+1} = 2\lambda \Delta x_0. \tag{5.19}$$

Mas, para que se tenha  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$ , o parâmetro  $\lambda$  é escolhido de forma que

$$\frac{\delta x_0}{2} = 2\lambda \Delta x_0. \tag{5.20}$$

Como  $\delta x_0 = \frac{\Delta x_0}{2}$ , tem-se

$$\frac{\delta x_0}{2} = 4\lambda \delta x_0,\tag{5.21}$$

que implica em

$$\lambda = \frac{1}{8}.\tag{5.22}$$

Logo, para  $u = x_0$ , tem-se que  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1} \iff \lambda = \frac{1}{8}$ .

No caso em que  $u = x_0 x_1$ , tem-se

$$u_{\alpha_R}^{\ell+1} = \frac{\delta x_0 \Delta x_1}{4},\tag{5.23}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = 2\lambda \Delta x_1 \delta x_0, \qquad (5.24)$$

Mais uma vez, para que se tenha  $u_{\alpha_R}^{\ell+1}=\tilde{u}_{\alpha_R}^{\ell+1},$  é necessário que  $\lambda$  satisfaça

$$\frac{\delta x_0 \Delta x_1}{4} = 2\lambda \Delta x_1 \delta x_0, \tag{5.25}$$

ou seja,

$$\lambda = \frac{1}{8}.\tag{5.26}$$

Logo, para  $u = x_0 x_1$ , tem-se  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1} \iff \lambda = \frac{1}{8}$ . Note que essa é a mesma exigência do caso em que  $u = x_0$ .

Para  $u = x_0^2$ , também não há restrições sobre  $\lambda$ , pois

$$u_{\alpha_R}^{\ell+1} = \frac{\Delta x_0^2}{12},\tag{5.27}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{\Delta x_0^2}{12}.$$
(5.28)

Logo, para  $u = x_0^2$ , tem-se  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$  para todo  $\lambda$ .

Para  $u = x_0 x_1^2$ , tem-se

$$u_{\alpha_R}^{\ell+1} = \frac{\Delta x_0 \Delta x_1^2}{12},$$
 (5.29)

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{2\lambda\Delta x_0 \Delta x_1^2}{3}.$$
(5.30)

Logo, para  $u = x_0 x_1^2$ , a identidade  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$  é válida se  $\lambda = \frac{1}{8}$ . Para  $u = x_0^2 x_1^2$ , tem-se

$$u_{\alpha_R}^{\ell+1} = \frac{\Delta x_0^2 \Delta x_1^2}{36},\tag{5.31}$$

$$\tilde{u}_{\alpha_R}^{\ell+1} = \frac{\Delta x_0^2 \Delta x_1^2}{36},$$
(5.32)

Logo, a identidade  $u_{\alpha_R}^{\ell+1} = \tilde{u}_{\alpha_R}^{\ell+1}$  é válida para todo  $\lambda$ .

Assim, com  $\lambda = \frac{1}{8}$ , a fórmula

$$\tilde{u}_{\alpha_R}^{\ell+1} = u_{\alpha}^{\ell} + \lambda \left[ u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell} \right]$$
(5.33)

é exata para  $u = 1, x_0, x_1, x_0 x_1, x_0^2, x_1^2, x_0 x_1^2, x_0^2 x_1^2$ . Isto implica a reconstrução exata da média celular para polinômios de grau menor ou igual a 2 em cada variável. Observase que esse operador de previsão para malhas diádicas bidimensionais utiliza a mesma fórmula do caso unidimensional.

#### Relação entre os Operadores para Malhas Quad-grid e Malhas Diádicas

Antes de prosseguir, fazemos uma breve pausa para descrever o esquema MR proposto por Birari e Harten em [2], que é desenvolvido para malhas  $2^d$ -ádicas, com d = 2.

Para efeito de comparação com os esquemas para malhas diádicas bidimensionais, apresentamos em seguida a notação utilizada no decorrer desta seção. Sem perda de generalidade, consideremos o conjunto de células quadradas  $c^{\ell}_{\alpha}$ ,  $\alpha = 0, 1, \dots, 8$ , conforme Figura 5.2 (1), presentes no nível  $\ell$  de uma malha diádica. Na Figura 5.2 (2), apresentamos os filhos, em particular, da célula  $c^{\ell}_{4}$ , mas a determinação dos filhos de  $c^{\ell}_{\alpha}$  para algum  $\alpha \in \{0, 1, \dots, 8\}$  é análoga. Representamos o filho esquerdo por  $c^{\ell+1}_{\alpha_L}$  e o filho direito por  $c_{\alpha_R}^{\ell+1}$ . Na Figura 5.2 (3), vemos os netos da célula  $c_4^{\ell}$ . O filho esquerdo de  $c_{\alpha_L}^{\ell+1}$  é denotado por  $c_{\alpha_L}^{\ell+2}$  e o filho direito por  $c_{\alpha_R}^{\ell+2}$ , assim como o filho esquerdo de  $c_{\alpha_R}^{\ell+1}$  é denotado por  $c_{\alpha_R}^{\ell+2}$  e o filho direito por  $c_{\alpha_R}^{\ell+2}$ . É importante observar que as células  $c_{\alpha_L}^{\ell+2}$ ,  $c_{\alpha_R}^{\ell+2}$ ,  $c_{\alpha_R}^{\ell+2}$  e  $c_{\alpha_R}^{\ell+2}$ coincidem com os filhos das células  $c_{\alpha}^{\ell}$ ,  $\alpha = 0, 1, \dots, 8$ , caso o refinamento adotado seja o das malhas  $2^d$ -ádicas.

Para expressar a média celular de uma função u na célula  $c^{\ell}_{\alpha}$ , escrevemos  $u^{\ell}_{\alpha}$ .



Figura 5.2: Nomenclatura das células.

De acordo com [2], sabemos que o operador de restrição nas malhas  $2^d$ -ádicas é dado por

$$u_4^{\ell} = \frac{1}{4} (u_{4_R}^{\ell+2} + u_{4_R}^{\ell+2} + u_{4_R}^{\ell+2} + u_{4_L}^{\ell+2}).$$

Por outro lado, pela aditividade da integral, temos, para dois níveis de refinamento, que

$$u_{4}^{\ell} = \frac{1}{2} \left( u_{4_{L}}^{\ell+1} + u_{4_{R}}^{\ell+1} \right) = \frac{1}{2} \left[ \frac{1}{2} \left( u_{4_{L}}^{\ell+2} + u_{4_{L}}^{\ell+2} \right) + \frac{1}{2} \left( u_{4_{R}}^{\ell+2} + u_{4_{L}}^{\ell+2} \right) \right]$$
  
$$= \frac{1}{4} (u_{4_{L}}^{\ell+2} + u_{4_{L}}^{\ell+2} + u_{4_{R}}^{\ell+2} + u_{4_{R}}^{\ell+2} + u_{4_{R}}^{\ell+2}).$$
 (5.34)

Isto prova que, para 2 níveis de refinamento, os operadores de restrição coincidem.

Reciprocamente, para fazermos previsões das médias  $u_{4_L^R}^{\ell+2}$ ,  $u_{4_L^L}^{\ell+2}$ ,  $u_{4_R^R}^{\ell+2}$  e  $u_{4_R^R}^{\ell+2}$  a partir de  $u^{\ell} = \{\cdots, c_0^{\ell}, c_1^{\ell}, c_2^{\ell}, c_3^{\ell}, c_6^{\ell}, c_7^{\ell}, c_8^{\ell}, \cdots\}$ , as fórmulas de [2], para  $\lambda = \frac{1}{8}$ , são

$$\begin{split} \tilde{u}_{4_{R}}^{\ell+2} &= u_{4}^{\ell} + \lambda(u_{5}^{\ell} - u_{3}^{\ell}) + \lambda(u_{1}^{\ell} - u_{7}^{\ell}) + \lambda^{2}(u_{2}^{\ell} - u_{8}^{\ell} - u_{0}^{\ell} + u_{6}^{\ell}), \\ \tilde{u}_{4_{L}}^{\ell+2} &= u_{4}^{\ell} - \lambda(u_{5}^{\ell} - u_{3}^{\ell}) + \lambda(u_{1}^{\ell} - u_{7}^{\ell}) - \lambda^{2}(u_{2}^{\ell} - u_{8}^{\ell} - u_{0}^{\ell} + u_{6}^{\ell}), \\ \tilde{u}_{4_{R}}^{\ell+2} &= u_{4}^{\ell} + \lambda(u_{5}^{\ell} - u_{3}^{\ell}) - \lambda(u_{1}^{\ell} - u_{7}^{\ell}) - \lambda^{2}(u_{2}^{\ell} - u_{8}^{\ell} - u_{0}^{\ell} + u_{6}^{\ell}), \\ \tilde{u}_{4_{L}}^{\ell+2} &= u_{4}^{\ell} - \lambda(u_{5}^{\ell} - u_{3}^{\ell}) - \lambda(u_{1}^{\ell} - u_{7}^{\ell}) + \lambda^{2}(u_{2}^{\ell} - u_{8}^{\ell} - u_{0}^{\ell} + u_{6}^{\ell}). \end{split}$$
(5.35)

Consideremos agora o esquema de previsão de malhas diádicas fazendo primeiro a divisão de células perpendicular ao eixo 0, seguida da divisão perpendicular ao eixo-1, conforme indicadas na Figura 5.2 (2). Segundo o esquema de previsão (5.5), as médias celulares do filho esquerdo e do filho direito das células  $c_7^\ell$ ,  $c_4^\ell$  e  $c_1^\ell$ , são

$$\begin{split} \tilde{u}_{7_L}^{\ell+1} &= u_7^{\ell} - \lambda (u_8^{\ell} - u_6^{\ell}), \\ \tilde{u}_{7_R}^{\ell+1} &= u_7^{\ell} + \lambda (u_8^{\ell} - u_6^{\ell}), \\ \tilde{u}_{4_L}^{\ell+1} &= u_4^{\ell} - \lambda (u_5^{\ell} - u_3^{\ell}), \\ \tilde{u}_{4_R}^{\ell+1} &= u_4^{\ell} + \lambda (u_5^{\ell} - u_3^{\ell}), \\ \tilde{u}_{1_L}^{\ell+1} &= u_1^{\ell} - \lambda (u_2^{\ell} - u_0^{\ell}), \\ \tilde{u}_{1_R}^{\ell+1} &= u_1^{\ell} + \lambda (u_2^{\ell} - u_0^{\ell}). \end{split}$$
(5.36)

Agora, basta aplicar o operador de previsão (5.5) aos filhos de  $c_{4_L}^{\ell+1}$  e  $c_{4_R}^{\ell+1}$ . Usando os valores previstos em (5.36), temos

$$\begin{split} \tilde{u}_{4L}^{\ell+2} &= u_4^{\ell} - \lambda(u_5^{\ell} - u_3^{\ell}) - \lambda[u_1^{\ell} - \lambda(u_2^{\ell} - u_0^{\ell}) - (u_7^{\ell} - \lambda(u_8^{\ell} - u_6^{\ell}))] \\ &= u_4^{\ell} - \lambda(u_5^{\ell} - u_3^{\ell}) - \lambda(u_1^{\ell} - u_7^{\ell}) + \lambda^2(u_2^{\ell} - u_0^{\ell} - u_8^{\ell} + u_6^{\ell}), \\ \tilde{u}_{4L}^{\ell+2} &= u_4^{\ell} - \lambda(u_5^{\ell} - u_3^{\ell}) + \lambda[u_1^{\ell} - \lambda(u_2^{\ell} - u_0^{\ell}) - (u_7^{\ell} - \lambda(u_8^{\ell} - u_6^{\ell}))] \\ &= \tilde{u}_4^{\ell} - \lambda(u_5^{\ell} - u_3^{\ell}) + \lambda(u_1^{\ell} - u_7^{\ell}) - \lambda^2(u_2^{\ell} - u_0^{\ell} - u_8^{\ell} + u_6^{\ell}), \\ u_{4L}^{\ell+2} &= u_4^{\ell} + \lambda(u_5^{\ell} - u_3^{\ell}) - \lambda[u_1^{\ell} + \lambda(u_2^{\ell} - u_0^{\ell}) - (u_7^{\ell} + \lambda(u_8^{\ell} - u_6^{\ell}))] \\ &= \tilde{u}_4^{\ell} + \lambda(u_5^{\ell} - u_3^{\ell}) - \lambda(u_1^{\ell} - u_7^{\ell}) - \lambda^2(u_2^{\ell} - u_0^{\ell} - u_8^{\ell} + u_6^{\ell}), \\ u_{4R}^{\ell+2} &= u_4^{\ell} + \lambda(u_5^{\ell} - u_3^{\ell}) + \lambda[u_1^{\ell} + \lambda(u_2^{\ell} - u_0^{\ell}) - (u_7^{\ell} + \lambda(u_8^{\ell} - u_6^{\ell}))] \\ &= u_4^{\ell} + \lambda(u_5^{\ell} - u_3^{\ell}) + \lambda[u_1^{\ell} + \lambda(u_2^{\ell} - u_0^{\ell}) - (u_7^{\ell} + \lambda(u_8^{\ell} - u_6^{\ell}))] \\ &= u_4^{\ell} + \lambda(u_5^{\ell} - u_3^{\ell}) + \lambda(u_1^{\ell} - u_7^{\ell}) + \lambda^2(u_2^{\ell} - u_0^{\ell} - u_8^{\ell} + u_6^{\ell}), \end{split}$$

que coincidem com as previsões (5.35) de [2]. Ou seja, vale o seguinte teorema.

**Teorema 5.1.** Para malhas diádicas em um espaço bidimensional, o esquema de previsão (5.5), partindo de uma malha quadrada, e depois de dois níveis de refinamento, é equivalente ao esquema de previsão dado pelas equações (5.35), propostas em [2].

# 5.3.2 Previsão Exata para Polinômios de Grau $\leq 2s$ em Malhas Diádicas d-dimensionais

Motivados pelos resultados do operador de previsão derivado na seção anterior, propomos esquemas de previsão da forma

$$\left(P_{\ell}^{\ell+1}u^{\ell}\right)_{\alpha_{R}} = u_{\alpha}^{\ell} + \sum_{j=1}^{s} \lambda_{j} \left[u_{\alpha+j}^{\ell} - u_{\alpha-j}^{\ell}\right], \qquad (5.38)$$

em que  $s \ge 1$  e as médias consideradas são referentes à célula  $c_{\alpha}^{\ell}$  e seus 2s vizinhos mais próximos na direção  $i = (\ell \mod d)$ , satisfazendo  $c_{\alpha-s}^{\ell} < \cdots < c_{\alpha-1}^{\ell} < c_{\alpha}^{\ell} < c_{\alpha+1}^{\ell} \cdots < c_{\alpha+s}^{\ell}$ . Lembramos que, para o caso de s = 1,  $\lambda_1 = \frac{1}{8}$ . Para s = 2, os valores dos coeficientes são

$$\lambda_1 = \frac{11}{8}, \lambda_2 = -\frac{3}{128}.$$

Para ser conservativo, devemos ter

$$(P_{\ell}^{\ell+1}u^{\ell})_{\alpha_{L}} = u_{\alpha}^{\ell} - \sum_{j=1}^{s} \lambda_{j} \left[ u_{\alpha+j}^{\ell} - u_{\alpha-j}^{\ell} \right].$$
 (5.39)

**Teorema 5.2.** Suponha que os coeficientes  $\lambda_j$  sejam escolhidos de forma que o operador seja exato, no caso unidimensional, para polinômios de grau  $\leq 2s$ , como os obtidos por Harten [25]. Ou seja, assumimos que, para  $0 \leq p \leq 2s$ ,

$$2\int_{c_{\alpha_R}^{\ell+1}} x^p dx = \int_{c_{\alpha}^{\ell}} x^p dx + \sum_{j=1}^s \lambda_j \left[ \int_{c_{\alpha+j}^{\ell}} x^p dx - \int_{c_{\alpha-j}^{\ell}} x^p dx \right].$$
 (5.40)

Então, o operador de previsão (5.38) para malhas diádicas d-dimensionais é exato para qualquer polinômio de grau  $\leq 2s$  em cada uma das variáveis.

Demonstração. No caso d-dimensional, suponha que  $c_{\alpha}^{\ell} = \prod_{i=0}^{d-1} c_{\alpha_i}^{\ell}$ , em que  $c_{\alpha_i}^{\ell}$  correspondem a células unidimensionais no eixo *i*, com longitude  $\Delta x_i$ , e que a subdivisão da

célula é feita perpendicularmente ao eixo n  $(\overrightarrow{0x_n})$ . Sendo  $u(x) = \prod_{i=0}^{d-1} x_i^{p_i}, 0 \le p_i \le 2s$ , e tendo em consideração que  $|c_{\alpha+1}^{\ell}| = |c_{\alpha}^{\ell}| = 2|c_{\alpha_R}^{\ell+1}|$ , segue que

$$u_{\alpha_R}^{\ell+1} = \frac{2}{|c_{\alpha}^{\ell}|} \int_{c_{\alpha_n R}^{\ell+1}} x_n^{p_n} dx_n \prod_{i=0, i \neq n}^{d-1} \int_{c_{\alpha_i}^{\ell}} x_i^{p_i} dx_i$$

е

$$u_{\alpha+j}^{\ell} = \frac{1}{|c_{\alpha}^{\ell}|} \int_{c_{\alpha_{n+j}}^{\ell}} x_n^{p_n} dx_n \prod_{i=0, i \neq n}^{d-1} \int_{c_{\alpha_i}^{\ell}} x_i^{p_i} dx_i.$$

Portanto, usando (5.40), segue que

$$\widetilde{u}_{\alpha_{R}}^{\ell+1} = \frac{1}{|c_{\alpha}^{\ell}|} \prod_{i=0, i \neq n}^{d-1} \int_{c_{\alpha_{i}}^{\ell}} x_{i}^{p_{i}} dx_{i} \left[ \int_{c_{\alpha_{n}}^{\ell}} x_{n}^{p_{n}} dx_{n} + \sum_{j=1}^{2s} \lambda_{j} \left( \int_{c_{\alpha_{n+j}}^{\ell}} x_{n}^{p_{n}} dx_{n} - \int_{c_{\alpha_{n-j}}^{\ell}} x_{n}^{p_{n}} dx_{n} \right) \right] \\
= \frac{2}{|c_{\alpha}^{\ell}|} \prod_{i=0, i \neq n}^{d-1} \int_{c_{\alpha_{i}}^{\ell}} x_{i}^{p_{i}} dx_{i} \int_{c_{\alpha_{n}R}^{\ell+1}} x_{n}^{p_{n}} dx_{n} \\
= u_{\alpha_{R}}^{\ell+1}.$$
(5.41)

Pela equação acima, concluímos que o operador de previsão para as médias celulares em uma malha diádica d-dimensional pode ser construído a partir de uma fórmula unidimensional, sendo exato para polinômios de grau menor ou igual a 2s em cada uma das variáveis.

Lembramos que, uma vez definido o operador de previsão, podemos medir o erro dessa previsão em relação ao valor exato da média celular. Isto é,

$$d^{\ell}_{\alpha} = u^{\ell+1}_{\alpha_R} - \tilde{u}^{\ell+1}_{\alpha_R}.$$
 (5.42)

Por exemplo, para os esquemas de previsão definidos na presente seção, os algoritmos MR direto e inverso, em qualquer dimensão, executam

Para 
$$\ell = L - 1, \dots, 0$$
  
Para  $\alpha \in \mathcal{K}(\ell)$   
 $u_{\alpha}^{\ell} = \frac{u_{\alpha_{L}}^{\ell+1} + u_{\alpha_{R}}^{\ell+1}}{2},$ 

$$d_{\alpha}^{\ell} = u_{\alpha_{R}}^{\ell+1} - \begin{cases} u_{\alpha}^{\ell} + \frac{1}{8} \left[ u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell} \right], & \text{para } s = 1, \\ u_{\alpha}^{\ell} + \frac{11}{8} \left[ u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell} \right] - \frac{3}{128} \left[ u_{\alpha+2}^{\ell} - u_{\alpha-2}^{\ell} \right], & \text{para } s = 2, \end{cases}$$
(5.43)

$$\begin{split} \text{Para} \quad & \ell = 0, \cdots, L-1 \\ \text{Para} \quad & \alpha \in \mathcal{K}(\ell) \\ & u_{\alpha R}^{\ell+1} = d_{\alpha}^{\ell} + \begin{cases} u_{\alpha}^{\ell} + \frac{1}{8} \left[ u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell} \right], & \text{para } s = 1, \\ u_{\alpha}^{\ell} + \frac{11}{8} [u_{\alpha+1}^{\ell} - u_{\alpha-1}^{\ell}] - \frac{3}{128} [u_{\alpha+2}^{\ell} - u_{\alpha-2}^{\ell}], & \text{para } s = 2, \\ u_{\alpha L}^{\ell+1} = 2u_{\alpha}^{\ell} - u_{\alpha R}^{\ell+1}. \end{split}$$

#### ou

# Capítulo 6

# Análise MR para Médias Celulares em Malhas Diádicas: Aspecto Funcional

Esquemas MR também podem ser interpretados em um contexto funcional, no qual as funções são representadas em espaços de aproximação em uma hierarquia de diferentes escalas  $U^0 \subset U^1 \subset \cdots \subset U^{\ell} \subset U^{\ell+1} \cdots$  Em um determinado nível  $\ell$ , as funções de  $U^{\ell}$ podem ser representadas de diferentes formas, dependendo da escolha da base do referido espaço, podendo ser uma base em um único nível ou uma base multinível. Os algoritmos MR correspondem às operações de mudança de uma base para a outra.

Neste capítulo, como base para a teoria que desenvolvemos no contexto de esquemas MR em malhas diádicas, fazemos uma revisão do aspecto funcional de análises de multirresolução para médias celulares para o caso unidimensional, conforme indicado no trabalho de Harten [24] e discutido em Kaibara [30]. A seguir, como contribuição, construímos os espaços aproximantes  $U^{\ell}$  associados aos esquemas MR para médias celulares em malhas diádicas em duas dimensões. Generalizamos o resultado, estendendo esta análise para o caso de esquemas MR em malhas diádicas em dimensões superiores.

### 6.1 Caso Unidimensional

Sejam R = [0, 1], G uma malha diádica em  $R \in P_{\ell}^{\ell+1}$  um esquema de previsão para médias celulares na malha G. Para a definição de um contexto funcional de análise de MR, é fundamental o conceito de convergência do operador de previsão  $P_{\ell}^{\ell+1}$ .

**Definição 6.1.** O operador de previsão é convergente se, a partir de qualquer amostra  $s^{\ell}$ de médias celulares no nível  $\ell$ , existe uma única função integrável s(t) tal que  $\mathcal{D}^{\ell}s = s^{\ell}$ e suas médias celulares em qualquer nível  $j \ge \ell + 1$  coincidem com os dados de saída da aplicação iterativa do esquema de previsão. Ou seja

$$\mathcal{D}^j s = \prod_{i=\ell}^{j-1} P_i^{i+1} s^\ell.$$

Supondo que o operador de previsão seja convergente, para cada célula  $c_{\alpha}^{\ell} \in G_{\ell}$ , definese a função básica  $\phi_{\alpha}^{\ell}(t)$  como resultado do processo de refinamento iterativo da amostra  $s_{\beta}^{\ell} = \delta_{\alpha\beta}$ . O espaço  $U^{\ell}$  é definido como

$$U^{\ell} = \operatorname{span}\{\phi^{\ell}_{\alpha}(t), \alpha \in \mathcal{K}(\ell)\}.$$

Fica claro que  $U^\ell \subset U^{\ell+1}$  e que

$$\sum_{\alpha \in \mathcal{K}(\ell)} u_{\alpha}^{\ell} \phi_{\alpha}^{\ell}(t) = \sum_{\alpha \in \mathcal{K}(\ell+1)} \tilde{u}_{\alpha}^{\ell+1} \phi_{\alpha}^{\ell+1}(t),$$

em que  $\tilde{u}^{\ell+1} = P_{\ell}^{\ell+1} u^{\ell}$ .

Os operadores de previsão de médias celulares em malhas diádicas unidimensionais definidos em Harten [24] resultam ser convergentes, gerando espaços de multirresolução correspondentes a uma das famílias biortogonais descritas em Cohen *et al.* [13].

As funções  $\phi_{\alpha}^{\ell}(t)$ , chamadas funções de escalonamento, verificam as seguintes propriedades ([30]).

1. Interpolação de médias celulares:

$$\frac{1}{|c_{\alpha}^{\ell}|} \int_{c_{\alpha}^{\ell}} \phi_{\beta}^{\ell}(t) dt = \begin{cases} 1, & \text{se } \beta = \alpha \\ 0, & \text{se } \beta \neq \alpha. \end{cases}$$
(6.1)

- 2. Regularidade: a regularidade de  $\phi_{\alpha}^{\ell}(t)$  aumenta com a ordem do operador de previsão  $P_{\ell}^{\ell+1}$  (para  $s = 1, \ \phi_{\alpha}^{\ell} \in C^0$ , para  $s = 2, \ \phi_{\alpha}^{\ell} \in C^1$ ).
- 3. Suporte local: o suporte de  $\phi_{\alpha}^{\ell}(t)$  está limitado em um intervalo contendo  $c_{\alpha}^{\ell}$ , de longitude  $\approx 2^{-\ell}$ .
- 4. Relação de escala: a função  $\phi^{\ell}_{\alpha}(t)$  satisfaz a equação

$$\phi_{\alpha}^{\ell}(t) = \sum_{\beta \in \mathcal{K}(\ell+1)} q_{\alpha\beta} \phi_{\beta}^{\ell+1}(t), \qquad (6.2)$$

em que

$$q_{\alpha\beta} = P_{\ell}^{\ell+1} \delta_{\alpha\beta}. \tag{6.3}$$

**Exemplo 6.1.** Consideremos a amostra de médias celulares  $s^2 = (0, 1, 0, 0)$  e o operador de previsão exato para polinômios de grau  $\leq 2$ , conforme indicados nas fórmulas (5.5) e (5.6), com  $\lambda = \frac{1}{8}$ . Neste exemplo, consideramos uma condição de contorno periódica e, aplicando o operador de previsão, obtemos os coeficientes  $q_{\alpha\beta}$  da equação (6.3) dados por  $(-\lambda, \lambda, 1, 1, \lambda, -\lambda, 0, 0)$ .

Seja  $W^{\ell} \subset U^{\ell+1}$  o núcleo do operador  $\mathcal{D}^{\ell}$ . Verifica-se que

$$W^{\ell} = \operatorname{span}\{\psi^{\ell}_{\alpha}(t), \alpha \in \mathcal{K}(\ell)\},\$$

em que a função  $wavelet~\psi^\ell_\alpha(t)$ é definida por

$$\psi_{\alpha}^{\ell}(t) = \phi_{\alpha_R}^{\ell+1}(t) - \phi_{\alpha_L}^{\ell+1}(t),$$

 $\alpha_R \in \alpha_L$  sendo os índices das células filho de  $c^{\ell}_{\alpha}$ . De fato, em primeiro lugar, podemos ver que o conjunto  $\{\psi^{\ell}_{\alpha}(t), \alpha \in \mathcal{K}(\ell)\}$  é linearmente independente. Sendo

$$\sum_{\alpha \in \mathcal{K}(\ell)} d^{\ell}_{\alpha} \psi^{\ell}_{\alpha}(t) = 0,$$

e como

$$\left(\mathcal{D}^{\ell+1}\psi_{\alpha}^{\ell}\right)_{\gamma} = \left(\mathcal{D}^{\ell+1}\phi_{\alpha_{R}}^{\ell+1}\right) - \left(\mathcal{D}^{\ell+1}\phi_{\alpha_{L}}^{\ell+1}\right)_{\gamma} = \delta_{\gamma\alpha_{R}} - \delta_{\gamma\alpha_{L}},$$

segue que  $d_{\alpha}^{\ell}=0$  para todo  $\alpha.$  Além disso, sendo

$$\left(\mathcal{D}^{\ell}\psi_{\alpha}^{\ell}\right)_{\gamma} = \left(\mathcal{D}^{\ell}\phi_{\alpha_{R}}^{\ell+1}(t)\right)_{\gamma} - \left(\mathcal{D}^{\ell}\phi_{\alpha_{L}}^{\ell+1}(t)\right)_{\gamma},$$

pela aditividade da integral, segue que

$$\left(\mathcal{D}^{\ell}\phi_{\alpha_{R}}^{\ell+1}(t)\right)_{\gamma} = \frac{1}{|c_{\gamma}^{\ell}|} \int_{c_{\gamma_{R}}^{\ell}} \phi_{\alpha_{R}}^{\ell+1}(t)dt + \frac{1}{|c_{\gamma}^{\ell}|} \int_{c_{\gamma_{L}}^{\ell}} \phi_{\alpha_{R}}^{\ell+1}(t)dt = \frac{1}{|c_{\gamma}^{\ell}|} \left(\delta_{\alpha_{R}\gamma_{R}}|c_{\gamma_{R}}^{\ell}| - \delta_{\alpha_{R}\gamma_{L}}|c_{\gamma_{L}}^{\ell}|\right)$$

е

$$\left(\mathcal{D}^{\ell}\phi_{\alpha_{L}}^{\ell+1}(t)\right)_{\gamma} = \frac{1}{|c_{\gamma}^{\ell}|} \int_{c_{\gamma_{R}}^{\ell}} \phi_{\alpha_{L}}^{\ell+1}(t)dt + \frac{1}{|c_{\gamma}^{\ell}|} \int_{c_{\gamma_{L}}^{\ell}} \phi_{\alpha_{L}}^{\ell+1}(t)dt = \frac{1}{|c_{\gamma}^{\ell}|} \left(\delta_{\alpha_{L}\gamma_{R}}|c_{\gamma_{R}}^{\ell}| - \delta_{\alpha_{L}\gamma_{L}}|c_{\gamma_{L}}^{\ell}|\right).$$

Portanto, também verificamos que  $(\mathcal{D}^{\ell}\psi^{\ell}_{\alpha})_{\gamma} = 0$ . Sendo assim, o subespaço  $W^{\ell}$  forma um espaço complementar de  $U^{\ell}$  em  $U^{\ell+1}$ , verificando a soma direta

$$U^{\ell+1} = U^{\ell} \oplus W^{\ell}.$$

Considere uma função  $u \in U^{\ell+1}$  escrita na forma

$$u(t) = \sum_{\alpha \in \mathcal{K}(\ell)} u_{\alpha}^{\ell} \phi_{\alpha}^{\ell}(t) + \sum_{\alpha \in \mathcal{K}(\ell)} d_{\alpha}^{\ell} \psi_{\alpha}^{\ell}(t).$$

Fica claro que  $(\mathcal{D}^{\ell}u)_{\alpha} = u_{\alpha}^{\ell} = \frac{u_{\alpha R}^{\ell+1} + u_{\alpha L}^{\ell+1}}{2}$ . Para determinar  $d_{\alpha}^{\ell}$ , observamos que

$$u_{\alpha_R}^{\ell+1} = \left(\mathcal{D}^{\ell+1}u(t)\right)_{\alpha_R} = \left(P_\ell^{\ell+1}u^\ell\right)_{\alpha_R} + \sum_{\alpha\in\mathcal{K}(\ell)} d_\alpha^\ell \left(\mathcal{D}^{\ell+1}\psi_\alpha^\ell(t)\right)_{\alpha_R}.$$

Como

$$\left(\mathcal{D}^{\ell+1}\psi^{\ell}_{\alpha}(t)\right)_{\alpha_{R}} = \left(\mathcal{D}^{\ell+1}\phi^{\ell+1}_{\alpha_{R}}(t)\right)_{\alpha_{R}} - \left(\mathcal{D}^{\ell+1}\phi^{\ell+1}_{\alpha_{L}}(t)\right)_{\alpha_{R}} = \delta_{\alpha_{R}\alpha_{R}} - \delta_{\alpha_{L}\alpha_{R}}$$

segue

$$u_{\alpha_R}^{\ell+1} = (P_\ell^{\ell+1} u^\ell)_{\alpha_R} + d_\alpha^\ell.$$

Isto é,

$$d^{\ell}_{\alpha} = u^{\ell+1}_{\alpha_R} - (P^{\ell+1}_{\ell} u^{\ell})_{\alpha_R}.$$
 (6.4)

Portanto, acabamos de mostrar que os esquemas MR correspondem a uma mudança entre a base { $\phi_{\alpha}^{\ell+1} : \alpha \in \mathcal{K}(\ell+1)$ } e { $\phi_{\beta}^{\ell} : \beta \in \mathcal{K}(\ell)$ }  $\cup$  { $\psi_{\beta}^{\ell} : \beta \in \mathcal{K}(\ell)$ }. De forma geral, para um determinado nível L > 0, dada a função  $u \in U^L$  podemos escrever sua expansão em multinível

$$u(t) = \sum_{\alpha \in \mathcal{K}(0)} u^0_\alpha \phi^0_\alpha(t) + \sum_{0 \leq \ell \leq L-1} \sum_{\alpha \in \mathcal{K}(\ell)} d^\ell_\alpha \psi^\ell_\alpha(t),$$

que corresponde a soma direta

$$U^L = U^0 \oplus W^0 \oplus \dots \oplus W^{L-1}.$$

Assim, no contexto funcional, os esquemas MR correspondem à mudança de base de um único nível e uma base em mais níveis de escala.

Na Figura 6.1, apresentamos o gráfico da função de escalonamento para uma célula do nível  $\ell = 7$ , à esquerda, e o gráfico da função *wavelet* no mesmo nível, à direita, correspondentes ao operador de previsão (5.5), exato para polinômios de grau  $\leq 2$ .



Figura 6.1: Função de escalonamento  $\phi_{\alpha}^{7}(t)$  e função wavelet  $\psi_{\alpha}^{7}(t)$ .

### 6.2 Caso Bidimensional

Consideramos  $R = [0,1] \times [0,1] \subset \mathbb{R}^2$ , G uma malha diádica sobre R. Em cada nível  $\ell$ , seja  $\ell_i$  a quantidade de divisões sobre o eixo i, na malha diádica  $G_\ell$ , de tal forma que  $\ell = \ell_0 + \ell_1$ . Podemos ter dois casos:  $\ell$  é par, quando  $\ell_0 = \ell_1$  ou,  $\ell$  é ímpar, quando  $\ell_0 = \ell_1 + 1$ . Sendo assim, cada célula  $c_{\alpha}^{\ell} \in G_{\ell}$  pode ser escrita como o produto cartesiano  $c_{\alpha}^{\ell} = c_{\alpha_0}^{\ell_0} \times c_{\alpha_1}^{\ell_1}$ , Desta forma, definimos a função básica  $\phi_{\alpha}^{\ell}(x)$  como resultado do produto tensorial das funções de base unidimensionais. Especificamente, a função escala é definida por

$$\phi_{\alpha}^{\ell}(x) = \phi_{\alpha_0}^{\ell_0}(x_0)\phi_{\alpha_1}^{\ell_1}(x_1), \tag{6.5}$$

em que  $x = (x_0, x_1)$  e  $\phi_{\alpha_i}^{\ell_i}(x_i)$  é um elemento da base unidimensional no intervalo sobre o eixo *i*, para i = 0, 1. Então, define-se o espaço  $U^{\ell}$  por

$$U^{\ell} = \operatorname{span}\{\phi^{\ell}_{\alpha}(x), \alpha \in \mathcal{K}(\ell)\}.$$

Ou seja,  $\mathbf{U}^{\ell} = U^{\ell_0} \times U^{\ell_1}$ . Observamos que, no caso em que  $\ell_0 = \ell_1$ , obtemos os espaços usuais que ocorrem nos esquemas MR para malhas quadradas.

Verificamos que as funções de escalonamento  $\phi^{\ell}_{\alpha}(x)$  satisfazem as seguintes propriedades.

 Interpolação de médias celulares: Usando a Equação (6.1), a função de escalonamento (6.5) satisfaz

$$\int_{c_{\beta}^{\ell}} \phi_{\alpha}^{\ell}(x) dx = \int_{c_{\beta_0}^{\ell_0}} \phi_{\alpha_0}^{\ell_0}(x_0) dx_0 \int_{c_{\beta_1}^{\ell_1}} \phi_{\alpha_1}^{\ell_1}(x_1) dx_1 = |c_{\alpha}^{\ell}| \delta_{\alpha\beta}.$$
(6.6)

- 2. Regularidade:  $\phi_{\alpha}^{\ell}(x)$  possui o mesmo grau de regularidade do caso unidimensional correspondente.
- Relação de escala: há dois casos a se considerar. Primeiro, quando ℓ<sub>0</sub> = ℓ<sub>1</sub>, a próxima subdivisão ocorre perpendicular ao eixo 0 e o nível seguinte ℓ+1 corresponde a ℓ<sub>0</sub>+1 e ℓ<sub>1</sub>. Pela definição da função escala e a relação de escala no caso unidimensional,

$$\phi_{\alpha}^{\ell}(x) = \phi_{\alpha_{0}}^{\ell_{0}}(x_{0})\phi_{\alpha_{1}}^{\ell_{1}}(x_{1}) = \left(\sum_{\beta_{0}\in\mathcal{K}(\ell_{0}+1)}q_{\alpha_{0}\beta_{0}}\phi_{\beta_{0}}^{\ell_{0}+1}(x_{0})\right)\phi_{\alpha_{1}}^{\ell_{1}}(x_{1}) \\
= \sum_{\beta\in\mathcal{K}(\ell+1)}q_{\beta}\phi_{\beta}^{\ell+1}(x),$$
(6.7)

em que para  $\beta = (\beta_0, \alpha_1) \in \mathcal{K}(\ell + 1)$  e o coeficiente  $q_\beta = q_{\alpha_0\beta_0}$  é dado pela equação (6.3).

De maneira análoga, quando  $\ell_0 = \ell_1 + 1$  e a próxima divisão é perpendicular ao eixo 1, no nível  $\ell + 1$  os níveis em cada coordenada são iguais a  $\ell_0$ . Portanto

$$\phi_{\alpha}^{\ell}(x) = \phi_{\alpha_{0}}^{\ell_{0}}(x_{0})\phi_{\alpha_{1}}^{\ell_{1}}(x_{1}) = \phi_{\alpha_{0}}^{\ell_{0}}(x_{0}) \left(\sum_{\beta_{1}\in\mathcal{K}(\ell_{1}+1)} q_{\beta_{1}\alpha_{1}}\phi_{\beta_{1}}^{\ell_{1}+1}(x_{1})\right)$$

$$= \sum_{\beta\in\mathcal{K}(\ell+1)} q_{\beta}\phi_{\beta}^{\ell+1}(x),$$
(6.8)

em que para  $\beta = (\alpha_0, \beta_1) \in \mathcal{K}(\ell + 1)$  e o coeficiente  $q_\beta = q_{\beta_1 \alpha_1}$  é dado pela equação (6.3).

Desta forma, fica claro que  $U^\ell \subset U^{\ell+1}$ e que

$$\sum_{\alpha \in \mathcal{K}(\ell)} u_{\alpha}^{\ell} \phi_{\alpha}^{\ell}(x) = \sum_{\alpha \in \mathcal{K}(\ell+1)} \tilde{u}_{\alpha}^{\ell+1} \phi_{\alpha}^{\ell+1}(x),$$

em que  $\tilde{u}^{\ell+1} = P_{\ell}^{\ell+1} u^{\ell}.$ 

Como no caso unidimensional, seja  $W^{\ell} \subset U^{\ell+1}$  o núcleo do operador  $\mathcal{D}^{\ell}$ . Verifica-se que

$$W^{\ell} = \operatorname{span}\{\psi^{\ell}_{\alpha}(x), \alpha \in \mathcal{K}(\ell)\},\$$

em que a função wavelet  $\psi^{\ell}_{\alpha}(x)$  é definida por

$$\psi_{\alpha}^{\ell}(x) = \begin{cases} \phi_{\alpha_1}^{\ell_1}(x_1)\psi_{\alpha_0}^{\ell_0}(x_0), & \text{se} \quad (\ell \mod 2) = 0, \\ \phi_{\alpha_0}^{\ell_0}(x_0)\psi_{\alpha_1}^{\ell_1}(x_1), & \text{se} \quad (\ell \mod 2) = 1. \end{cases}$$

Sendo núcleo do operador de discretização, o subespaço  $W^{\ell}$ , forma um espaço complementar de  $U^{\ell}$  em  $U^{\ell+1}$ , verificando a soma direta

$$U^{\ell+1} = U^{\ell} \oplus W^{\ell}.$$

Seja $u \in U^{\ell+1}$ uma função escrita na forma

$$u(x) = \sum_{\alpha \in \mathcal{K}(\ell)} u_{\alpha}^{\ell} \phi_{\alpha}^{\ell}(x) + \sum_{\alpha \in \mathcal{K}(\ell)} d_{\alpha}^{\ell} \psi_{\alpha}^{\ell}(x).$$

Fica claro que  $(\mathcal{D}^{\ell}u)_{\alpha} = u_{\alpha}^{\ell} = \frac{u_{\alpha R}^{\ell+1} + u_{\alpha L}^{\ell+1}}{2}$ . Para determinar  $d_{\alpha}^{\ell}$ , observamos que

$$u_{\alpha_R}^{\ell+1} = \left(\mathcal{D}^{\ell+1}u(x)\right)_{\alpha_R} = \left(P_\ell^{\ell+1}u^\ell\right)_{\alpha_R} + \sum_{\alpha\in\mathcal{K}(\ell)} d_\alpha^\ell \left(\mathcal{D}^{\ell+1}\psi_\alpha^\ell(x)\right)_{\alpha_R}.$$

Como, para a próxima divisão ortogonal ao eixo n,

$$\left(\mathcal{D}^{\ell+1}\psi_{\alpha}^{\ell}(x)\right)_{\alpha_{R}} = \left(\mathcal{D}^{\ell+1}(\phi_{\alpha_{i}}^{\ell}(x_{i})\psi_{\alpha_{n}}^{\ell}(x_{n}))\right)_{\alpha_{R}} = \left(\delta_{\alpha_{nR}\alpha_{nR}} - \delta_{\alpha_{nL}\alpha_{nR}}\right)\delta_{\alpha_{i}\alpha_{i}}$$

segue que

$$u_{\alpha_R}^{\ell+1} = (P_\ell^{\ell+1} u^\ell)_{\alpha_R} + d_\alpha^\ell.$$

Isto é,

$$d^{\ell}_{\alpha} = u^{\ell+1}_{\alpha_R} - (P^{\ell+1}_{\ell} u^{\ell})_{\alpha_R}.$$
(6.9)

Portanto, acabamos de mostrar que os esquemas MR correspondem a uma mudança entre a base  $\{\phi_{\alpha}^{\ell+1} : \alpha \in \mathcal{K}(\ell+1)\}$  e  $\{\phi_{\beta}^{\ell} : \beta \in \mathcal{K}(\ell)\} \cup \{\psi_{\beta}^{\ell} : \beta \in \mathcal{K}(\ell)\}.$ 

De forma geral, para um determinado nível L > 0, dada a função  $u \in U^L$  podemos escrever sua expansão em multinível

$$u(x) = \sum_{\alpha \in \mathcal{K}(0)} u_{\alpha}^{0} \phi_{\alpha}^{0}(x) + \sum_{0 \le \ell \le L-1} \sum_{\alpha \in \mathcal{K}(\ell)} d_{\alpha}^{\ell} \psi_{\alpha}^{\ell}(x),$$

que corresponde à soma direta

$$U^L = U^0 \oplus W^0 \oplus \dots \oplus W^{L-1}.$$

Assim, no contexto funcional, os esquemas MR correspondem à mudança de base de um único nível e uma base em mais níveis de escala.

Na Figura 6.2, apresentamos, para o caso bidimensional, os gráficos das funções de escalonamento para uma célula do nível  $\ell = 7$ , à esquerda, e o gráfico da função *wavelet* no mesmo nível, à direita.



Figura 6.2: Caso bidimensional: função de escalonamento  $\phi_{\alpha}^{7}(x)$  (esquerda) e função wavelet  $\psi_{\alpha}^{7}(x)$  (direita).

### 6.3 Caso *d*-dimensional

O estudo do caso d-dimensional segue a mesma linha que o caso bidimensional, isto é, definimos as funções escala como produto tensorial das funções unidimensionais, escalonadas apropriadamente. Consideramos  $R = [0,1]^d \subset \mathbb{R}^d$ , G uma malha diádica sobre R. Em cada nível  $\ell$ , seja  $\ell_i$  a quantidade de divisões sobre o eixo i, na malha diádica  $G_\ell$ , de tal forma que  $\ell = \sum_{i=0}^{d-1} \ell_i$ . Assim, cada célula  $c_{\alpha}^{\ell} \in G_{\ell}$  é dada pelo produto cartesiano  $c_{\alpha}^{\ell} = c_{\alpha_0}^{\ell_0} \times \cdots \times c_{\alpha_{d-1}}^{\ell_{d-1}}$ . Tomando  $c_{\alpha}^{\ell}$  como célula de referência, definimos a função básica  $\phi_{\alpha}^{\ell}(x)$  como resultado do produto tensorial das funções de base unidimensionais

$$\phi_{\alpha}^{\ell}(x) = \prod_{i=0}^{d-1} \phi_{\alpha_i}^{\ell_i}(x_i), \tag{6.10}$$

em que  $\phi_{\alpha_i}^{\ell_i}(x_i)$  são elementos da base unidimensional no intervalo sobre o eixo *i*, no nível  $\ell_i$ . Então, define-se o espaço  $U^{\ell}$  por

$$U^{\ell} = \operatorname{span}\{\phi^{\ell}_{\alpha}(x), \alpha \in \mathcal{K}(\ell)\}.$$

Ou seja,  $U^{\ell} = \prod_{i=0}^{d-1} U^{\ell_i}$ .

Como nos casos anteriores, verificamos que as funções de escalonamento  $\phi_{\alpha}^{\ell}(x)$  satisfazem as seguintes propriedades.

1. Interpolação de médias celulares:

$$\int_{c_{\beta}^{\ell}} \phi_{\alpha}^{\ell}(x) dx = |c_{\alpha}^{\ell}| \delta_{\beta\alpha}.$$
(6.11)

- 2. Regularidade:  $\phi_{\alpha}^{\ell}(x)$  possui o mesmo grau de regularidade do caso unidimensional correspondente.
- 3. Relação de escala: supondo que o refinamento para passar do nível  $\ell$  para  $\ell+1$  se dá perpendicularmente ao eixo n, os índices  $\beta$  de  $\mathcal{K}(\ell+1)$  em cada coordenada  $i \neq n$ correspondem aos índices de  $\mathcal{K}(\ell_i)$ , enquanto que na coordenada n correspondem aos índices de  $\mathcal{K}(\ell_n+1)$ . Portanto,

$$\phi_{\alpha}^{\ell}(x) = \prod_{i=0}^{d-1} \phi_{\alpha_i}^{\ell_i}(x_i) = \phi_{\alpha_n}^{\ell_n}(x_n) \prod_{i=0, i \neq n}^{d-1} \phi_{\alpha_i}^{\ell_i}(x_i)$$
$$= \left(\sum_{\beta_n \in \mathcal{K}(\ell_n+1)} q_{\beta_n \alpha_n} \phi_{\beta_n}^{\ell_n+1}(x_n)\right) \prod_{i=0, i \neq n}^{d-1} \phi_{\alpha_i}^{\ell_i}(x_i)$$
$$= \sum_{\beta \in \mathcal{K}(\ell+1)} q_{\beta} \phi_{\beta}^{\ell+1}(x),$$
(6.12)

em que  $q_{\beta} = q_{\beta_n \alpha_n}$ .

Novamente, fica claro que  $U^\ell \subset U^{\ell+1}$ e que

$$\sum_{\alpha \in \mathcal{K}(\ell)} u_{\alpha}^{\ell} \phi_{\alpha}^{\ell}(x) = \sum_{\alpha \in \mathcal{K}(\ell+1)} \tilde{u}_{\alpha}^{\ell+1} \phi_{\alpha}^{\ell+1}(x),$$

em que  $\tilde{u}^{\ell+1} = P_{\ell}^{\ell+1} u^{\ell}.$ 

Seja $W^\ell \subset U^{\ell+1}$ o núcleo do operador  $\mathcal{D}^\ell.$  Verifica-se que

$$W^{\ell} = \operatorname{span}\{\psi^{\ell}_{\alpha}(x), \alpha \in \mathcal{K}(\ell)\},\$$

em que a função wavelet  $\psi^\ell_\alpha(x)$  depende da direção do refinamento e é definida por

$$\psi_{\alpha}^{\ell}(x) = \psi_{\alpha_n}^{\ell_j}(x_n) \prod_{i=0, i \neq n}^{d-1} \phi_{\alpha_i}^{\ell_i}(x_i).$$

Sendo assim, o subespaço  $W^\ell$  forma um espaço complementar de  $U^\ell$  em  $U^{\ell+1},$  verificando a soma direta

$$U^{\ell+1} = U^{\ell} \oplus W^{\ell}.$$

Seja $u \in U^{\ell+1}$ uma função escrita na forma

$$u(x) = \sum_{\alpha \in \mathcal{K}(\ell)} u_{\alpha}^{\ell} \phi_{\alpha}^{\ell}(x) + \sum_{\alpha \in \mathcal{K}(\ell)} d_{\alpha}^{\ell} \psi_{\alpha}^{\ell}(x).$$

Fica claro que  $(\mathcal{D}^{\ell}u)_{\alpha} = u_{\alpha}^{\ell}$ . Além disso,

$$u_{\alpha_R}^{\ell+1} = \left(\mathcal{D}^{\ell+1}u(x)\right)_{\alpha_R} = \left(P_\ell^{\ell+1}u^\ell\right)_{\alpha_R} + \sum_{\alpha\in\mathcal{K}(\ell)} d_\alpha^\ell \left(\mathcal{D}^{\ell+1}\psi_\alpha^\ell(x)\right)_{\alpha_R}.$$

Sabendo que

$$\left(\mathcal{D}^{\ell+1}\psi_{\alpha}^{\ell}(x)\right)_{\alpha_{R}} = \left(\mathcal{D}^{\ell+1}(\psi_{\alpha_{n}}^{\ell}(x_{n})\prod_{i=0,i\neq n}^{d-1}\phi_{\alpha_{i}}^{\ell}(x_{i}))\right)_{\alpha_{R}} = \left(\delta_{\alpha_{nR}\alpha_{nR}} - \delta_{\alpha_{nL}\alpha_{nR}}\right)\prod_{i=0,i\neq n}^{d-1}\delta_{\alpha_{i}\alpha_{i}},$$

segue que

$$u_{\alpha_R}^{\ell+1} = (P_\ell^{\ell+1} u^\ell)_{\alpha_R} + d_\alpha^\ell.$$

Isto é,

$$d_{\alpha}^{\ell} = u_{\alpha_R}^{\ell+1} - (P_{\ell}^{\ell+1} u^{\ell})_{\alpha_R}.$$
(6.13)

Portanto, acabamos de mostrar que os esquemas MR correspondem a uma mudança entre a base  $\{\phi_{\alpha}^{\ell+1} : \alpha \in \mathcal{K}(\ell+1)\}$  e  $\{\phi_{\beta}^{\ell} : \beta \in \mathcal{K}(\ell)\} \cup \{\psi_{\beta}^{\ell} : \beta \in \mathcal{K}(\ell)\}.$ 

De forma geral, para um determinado nível L > 0, dada a função  $u \in U^L$ , podemos escrever sua expansão em multinível

$$u(x) = \sum_{\alpha \in \mathcal{K}(0)} u^0_{\alpha} \phi^0_{\alpha}(x) + \sum_{0 \le \ell \le L-1} \sum_{\alpha \in \mathcal{K}(\ell)} d^{\ell}_{\alpha} \psi^{\ell}_{\alpha}(x),$$

que corresponde à soma direta

$$U^L = U^0 \oplus W^0 \oplus \dots \oplus W^{L-1}.$$

Assim, no contexto funcional, os esquemas MR correspondem à mudança de base de uma base de único nível para uma base de mais níveis de escala.

# Capítulo 7

## Representações MR Adaptativas

Em análise wavelet, a representação de uma função em mutirresolução permite obter uma economia de espaço e tempo de processamento eliminando os coeficientes wavelet que não contribuem de maneira significativa para a reconstrução da função no nível mais refinado. No contexto de esquemas MR para médias celulares em malhas diádicas, essa compressão de dados significa que o processo de refinamento é interrompido nas células  $c_{\alpha}^{\ell}$  em que  $|d_{\alpha}^{\ell}|$ seja menor que uma tolerância  $\epsilon_{\ell}$ , previamente especificada. Neste capítulo apresentamos essa maneira de se obter uma representação em multirresolução adaptativa e estudamos a estimativa do erro cometido. Também comparamos a eficiência dos esquemas MR adaptativos baseados em malhas diádicas e em quad-grids aplicados em dois exemplos num domínio bidimensional com diferentes padrões de variação: um pulso Gaussiano, uma frente vertical. Também aplicamos o esquema MR adaptativo na compressão da imagem de uma paisagem.

## 7.1 Operador de Truncamento

Seja  $MR(u^L) = (u^0, d^0, \dots, d^{L-1})$  a representação MR das médias celulares de uma função u numa malha diádica  $G_L$ . A compressão implica em definir como zero os detalhes que tem valor absoluto menor que uma tolerância. Em outras palavras,

$$\tilde{d}^{\ell}_{\alpha} = \begin{cases} 0 & \text{se } |d^{\ell}_{\alpha}| \leq \epsilon_{\ell}, \\ d^{\ell}_{\alpha} & \text{se } |d^{\ell}_{\alpha}| > \epsilon_{\ell}. \end{cases}$$

Assim, a representação MR adaptativa de  $u^L$  é dada por  $MR_{\epsilon}(u^L)(u^0, \tilde{d}^0, \dots, \tilde{d}^{L-1})$ . Nesta tese, o parâmetro considerado é dado por

$$\epsilon_{\ell} = 2^{(\ell - L)} \epsilon, \tag{7.1}$$

em que  $\epsilon > 0$  é previamente especificado, como sugerido em [26].

#### 7.2 Erro de Truncamento

Como estudado no Capítulo 6, fazendo uso da interpretação dos esquemas MR no contexto funcional, supondo um esquema de previsão convergente, a representação MR das médias celulares de uma função u numa malha diádica  $G_L$ ,  $MR(u^L) = (u^0, d^0, \dots, d^{L-1})$ , corresponde à expansão em multinível

$$u^{L}(x) = u^{0}(x) + \sum_{\ell=0}^{L-1} \sum_{\alpha \in \mathcal{K}(\ell)} d^{\ell}_{\alpha} \psi^{\ell}_{\alpha}(x),$$

em que  $\psi_{\alpha}^{\ell}$  são as funções wavelet, base para o espaço  $W^{\ell} = U^{\ell+1} \setminus U^{\ell}$ .

Analogamente, após a operação de tuncamento, a representação adaptativa da função $MR_{\epsilon}(u^{L})(u^{0}, \tilde{d}^{0}, \cdots, \tilde{d}^{L-1}) \text{ corresponde à expansão incompleta}$ 

$$u_{\epsilon}^{L}(x) = u^{0}(x) + \sum_{\ell=0}^{L-1} \sum_{\alpha \in \tilde{\mathcal{K}}(\ell)} d_{\alpha}^{\ell} \psi_{\alpha}^{\ell}(x),$$

em que  $\tilde{\mathcal{K}}(\ell) = \{ \alpha \in \mathcal{K}(\ell) : |d_{\alpha}^{\ell}| > \epsilon_{\ell} \}.$ 

Assim, estimamos o erro de truncamento na norma  $L_1$ , por

$$\begin{aligned} \|u^{L} - u^{L}_{\epsilon}\|_{1} &= \left\| \sum_{\ell=0}^{L-1} \sum_{\alpha \notin \tilde{\mathcal{K}}(\ell)} d^{\ell}_{\alpha} \psi^{\ell}_{\alpha} \right\|_{1} \leq \sum_{\ell=0}^{L-1} \sum_{\alpha \notin \tilde{\mathcal{K}}(\ell)} \|d^{\ell}_{\alpha} \psi^{\ell}_{\alpha}\|_{1} \\ &= \sum_{\ell=0}^{L-1} \sum_{\alpha \notin \tilde{\mathcal{K}}(\ell)} |d^{\ell}_{\alpha}| \, \|\psi^{\ell}_{\alpha}\|_{1} \leq \sum_{\ell=0}^{L-1} 2^{\ell-L} \epsilon \sum_{\alpha \notin \tilde{\mathcal{K}}(\ell)} \|\psi^{\ell}_{\alpha}\|_{1}. \end{aligned}$$

Sabendo que  $\psi_{\alpha}^{\ell}$  é uma função integrável (é pelo menos contínua para previsões de ordem superior) e tem suporte local, com medida proporcional a  $|c_{\alpha}^{\ell}|$ , resulta que  $\|\psi_{\alpha}^{\ell}\|_{1} \approx |c_{\alpha}^{\ell}|$ . Logo,

$$\begin{aligned} \|u^L - u^L_\epsilon\|_1 &\leq \epsilon \sum_{\ell=0}^{L-1} 2^{\ell-L} \sum_{\alpha \in \mathcal{K}(\ell)} |c^\ell_\alpha| \leq \sum_{\ell=0}^{L-1} 2^{\ell-L} \epsilon |R| \\ &\leq \epsilon |R| \sum_{\ell=0}^{L-1} 2^{\ell-L} \leq C \epsilon. \end{aligned}$$

Vemos que o erro de reconstrução é proporcional ao parâmetro de truncamento  $\epsilon$ . Ou seja, a propriedade de convergência do operador de previsão implica que o esquema MR é estável, no sentido de que perturbarções nos coeficientes *wavelet* não são amplificadas sem controle pela aplicação iterativa do operador de previsão.

#### 7.3 Exemplos

Apresentamos a seguir alguns exemplos bidimensionais para ilustrar a compressão de dados em representações MR adaptativas.

# 7.3.1 Comparação de Esquemas MR em Malhas Diádicas e Quad-grids

Para comparar a eficiência dos esquemas MR em malhas diádicas com os esquemas MR tradicionais em *quad-grids*, fazemos a análise multirresolução de dois exemplos diferentes em 2D, usando discretizações por médias celulares. A análise é feita tanto para o esquema MR de Haar quanto para o esquema MR com previsão exata para polinômios de grau  $\leq 2$ .

Em todos os testes, consideramos a célula raiz  $[-1, 1] \times [-1, 1]$ , e a árvore inicial representando uma malha uniforme com  $2^{10} \times 2^{10} = 2^{20} = 1,048,576$  células folha. Isto corresponde a estruturas de árvore com L = 20 e L = 10 níveis de resolução para malhas diádicas e quad-grids, respectivamente. As médias celulares  $u_{\alpha}^{L}$  são calculadas, para toda célula folha  $c_{\alpha}^{L}$ , usando quadratura Gaussiana com 5 × 5 pontos. As árvores são podadas como descrito na seção anterior, considerando os parâmetros  $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$  para as malhas diádicas e para as *quad-grids* os parâmetros são escolhidos de forma a se obter erro de reconstrução de mesma ordem que na malha diádica.

#### Compressão

Definimos como compressão a porcentagem de células folha desprezadas em relação à malha uniforme. Isto é, a compressão é dada por

$$C = 100 - \frac{m * 100}{2^{20}},$$

em que m é o número de folhas na árvore.

#### Custo de Armazenamento de uma Árvore: Memória Usada

Para o cálculo da memória usada nas representações adaptativas, consideramos árvores em que todos os nós têm os mesmos campos de armazenamento de dados. No caso em que as folhas são explicitamente representadas na estrutura de dados, o espaço em bytes E usado pela estrutura é

$$E = (pA + B)n, (7.2)$$

em que n é o número de nós, p é o número de ponteiros em cada nó, A é o tamanho de um ponteiro em bytes, B é o tamanho de qualquer outra informação adicional armazenada em cada um dos nós (por exemplo as médias celulares ou os detalhes  $d^{\ell}_{\alpha}$ ). Em cada árvore tem-se  $n = (pm - 1)/(p - 1) \approx mp/(p - 1)$ , em que m é a quantidade de células folha. Nesta tese, supomos A = 4 e B = 8 bytes. Para malhas diádicas, p = 2, enquanto que p = 4, para quad-grids. Nos testes apresentados, indicamos por  $E_d$  e  $E_q$  a memória usada, em bytes, para os esquemas em malha diádica e quad-grid, respectivamente.

Primeiro Teste: No primeiro teste consideramos a função

$$u(x_0, x_1) = 0.9 * \exp(-100(x_0^2 + x_1^2)), \tag{7.3}$$

que descreve uma Gaussiana com centro em (0,0) (veja Figura 7.1).



Figura 7.1: Primeiro Teste: gráfico da função descrita pela Equação (7.3).

Nas Tabelas 7.1 e 7.2, listamos os parâmetros de truncamento  $\epsilon$  usados nas malhas diádicas e nas *quad-grids* para obtermos erros de reconstrução da mesma ordem, bem como a taxa de compressão obtida em cada um dos casos.

Na Figura 7.2, à esquerda, exibimos o erro de reconstrução em função da compressão usando esquemas MR de Haar e MR de alta ordem. À direita, apresentamos a razão entre as memórias necessárias para armazenar a malha diádica e *quad-grid* em função do parâmetro de truncamento. Observamos que, tanto para a MR de Haar quanto para MR de alta ordem, as malhas diádicas se mostraram mais eficientes, em termos da memória usada, para representar a função. No entanto, o ganho de memória no esquema MR de alta ordem é superior, da ordem de 20%, enquanto que com Haar o ganho é reduzido à medida que  $\epsilon$  diminui.

**Segundo Teste:** O segundo teste é feito tomando-se a função que descreve uma frente vertical

$$u(x_0, x_1) = 1 - \tanh(100 * x_0) \tag{7.4}$$

	Malha diádica				
$\epsilon$	# folhas usadas	$\operatorname{compressão}(\%)$	erro	memória usada $(E_d)$	
$1.0 * 10^{-1}$	3228	99.69	$7.9 * 10^{-4}$	103280	
$1.0 * 10^{-2}$	15392	98.53	$2.5 * 10^{-4}$	492528	
$1.0 * 10^{-3}$	38380	96.33	$2.2 * 10^{-5}$	1228144	
$1.0*10^{-4}$	59348	94.34	$2.1 * 10^{-6}$	1899120	
	Quad- $grid$				
$\epsilon$	# folhas usadas	$\operatorname{compressão}(\%)$	erro	memória usada $(E_q)$	
$0.5 * 10^{-1}$	4135	99.60	$8.0 * 10^{-4}$	132312	
$1.2 * 10^{-2}$	16789	98.39	$2.6 * 10^{-4}$	537240	
$1.2 * 10^{-3}$	40084	96.17	$2.2 * 10^{-5}$	1282680	

Tabela 7.1: Primeiro Teste e esquemas MR de Haar. Com L = 20 e L = 10 níveis de resolução para malhas diádicas e quad-grids, respectivamente.

	Malha diádica			
$\epsilon$	# folhas usadas	$\operatorname{compressão}(\%)$	erro	memória usada $(E_d)$
$1.0 * 10^{-1}$	616	99.94	$8.8 * 10^{-5}$	19696
$1.0 * 10^{-2}$	1464	99.86	$2.1 * 10^{-5}$	46832
$1.0 * 10^{-3}$	3412	99.67	$4.9 * 10^{-6}$	109168
$1.0 * 10^{-4}$	8940	99.15	$1.1 * 10^{-6}$	286064
	Quad-grid			
$\epsilon$	# folhas usadas	$\operatorname{compressão}(\%)$	erro	memória usada $(E_q)$
$0.14 * 10^{-1}$	766	99.92	$8.3 * 10^{-5}$	24504
$0.15 * 10^{-2}$	1858	99.82	$2.2 * 10^{-5}$	59448
$0.27 * 10^{-3}$	5065	99.51	$4.7 * 10^{-6}$	162072
$0.40 * 10^{-4}$	11200	98.93	$1.1 * 10^{-6}$	358392

Tabela 7.2: Primeiro Teste e esquemas MR de alta ordem. Com L = 20 e L = 10 níveis de resolução para malhas diádicas e *quad-grids*, respectivamente.



Figura 7.2: Primeiro Teste: à esquerda, parâmetro de truncamento em função da compressão usando esquemas de MR de Haar e MR de alta ordem. À direita, apresentamos a razão entre as memórias necessárias para armazenar a malha diádica e *quad-grid* em função do parâmetro de truncamento.

Os resultados dos testes podem ser vistos nas Tabelas 7.3 e 7.4.

Na Figura 7.4, à esquerda, exibimos o erro de reconstrução em função da compressão usando esquemas MR de Haar e MR de alta ordem. À direita, apresentados a razão entre as memórias necessárias para armazenar a malha diádica e *quad-grid* em função do parâmetro de truncamento. Observamos que a malha diádica é mais eficiente, em termos da memória usada, que a *quad-grid* para representar essa função. O ganho em memória fica em torno dos 50%, tanto com o esquema de Haar quanto com o esquema de alta ordem.

A análise desses testes revela que os esquemas MR em malhas diádicas são mais econômicos que as os equemas MR usuais em *quad-grids*, tanto em espaço de armazenamento quanto no número de folhas usadas para representar cada um dos sinais. Essa eficiência é mais significativa no caso do Segundo Teste, com um padrão de singularidade anisotrópico, em que o esquema em malha diádica é mais favorável que o esquema em malhas *quad-grids*. A economia no número de folhas tem papel importante quando buscamos a solução numérica de problemas de evolução, como estudado nos próximos capítulos.



Figura 7.3: Segundo Teste: gráfico da função descrita pela Equação (7.4).

	Malha diádica			
$\epsilon$	# folhas usadas	$\operatorname{compressão}(\%)$	erro	memória usada $(E_d)$
$1.0 * 10^{-1}$	8252	99.21	$7.6 * 10^{-4}$	264048
$1.0 * 10^{-2}$	14780	98.59	$7.4 * 10^{-5}$	472944
$1.0 * 10^{-3}$	20540	98.04	$7.5 * 10^{-6}$	657264
$1.0*10^{-4}$	26972	97.42	$7.2 * 10^{-7}$	863088
			-	
	Quad-grid			
$\epsilon$	Quad-grid # folhas usadas	compressão(%)	erro	memória usada $(E_q)$
$\epsilon 0.5 * 10^{-1}$	Quad-grid # folhas usadas 16120	compressão(%) 98.46	erro $7.9 * 10^{-4}$	$\frac{1}{515832}$ memória usada $(E_q)$
$\epsilon$ 0.5 * 10 <sup>-1</sup> 0.5 * 10 <sup>-2</sup>	$\begin{array}{c} Quad-grid\\ \# \text{ folhas usadas}\\ 16120\\ 28024 \end{array}$	compressão(%) 98.46 97.32	erro $7.9 * 10^{-4}$ $8.5 * 10^{-5}$	$\begin{array}{c} \text{memória usada}(E_q) \\ 515832 \\ 896760 \end{array}$
$\begin{array}{c} \epsilon \\ 0.5 * 10^{-1} \\ 0.5 * 10^{-2} \\ 0.5 * 10^{-3} \end{array}$	Quad-grid # folhas usadas 16120 28024 40312	compressão(%) 98.46 97.32 96.15	$\begin{array}{c} \text{erro} \\ 7.9 * 10^{-4} \\ 8.5 * 10^{-5} \\ 8.6 * 10^{-6} \end{array}$	$\begin{array}{c} \mbox{memória usada}(E_q) \\ 515832 \\ 896760 \\ 1289976 \end{array}$

Tabela 7.3: Segundo Teste e esquemas MR de Haar. Com L = 20 e L = 10 níveis de resolução para malhas diádicas e quad-grids, respectivamente.

	Malha diádica			
$\epsilon$	# folhas usadas	$\operatorname{compressão}(\%)$	erro	memória usada $(E_d)$
$1.0 * 10^{-1}$	4208	99.59	$2.3 * 10^{-4}$	134640
$1.0 * 10^{-2}$	6704	99.36	$1.1 * 10^{-4}$	214512
$1.0 * 10^{-3}$	17840	98.29	$4.6 * 10^{-6}$	570864
$1.0 * 10^{-4}$	23600	97.74	$5.3 * 10^{-7}$	755184
	$\bigcirc$ 1 1			
	Quad-grid			
e	Quad-grid # folhas usadas	compressão(%)	erro	memória usada $(E_q)$
$\epsilon$ 0.10 * 10 <sup>-1</sup>	Quad-grid # folhas usadas 8416	compressão(%) 99.19	erro $2.3 * 10^{-4}$	$\begin{array}{c} \hline \\ \text{memória usada}(E_q) \\ \hline 269304 \end{array}$
$\frac{\epsilon}{0.10 * 10^{-1}} \\ 0.05 * 10^{-2}$	Quad-grid # folhas usadas 8416 13024	compressão(%) 99.19 98.75	erro $2.3 * 10^{-4}$ $1.2 * 10^{-4}$	$ \begin{array}{c c}     memória usada(E_q) \\     269304 \\     416760 \\ \end{array} $
$\begin{array}{c} \epsilon \\ \hline 0.10 * 10^{-1} \\ 0.05 * 10^{-2} \\ 0.30 * 10^{-3} \end{array}$	Quad-grid # folhas usadas 8416 13024 35680	compressão(%) 99.19 98.75 96.59	$\begin{array}{c} \text{erro} \\ 2.3 * 10^{-4} \\ 1.2 * 10^{-4} \\ 4.6 * 10^{-6} \end{array}$	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$

Tabela 7.4: Segundo Teste: esquemas MR de alta ordem. Com L = 20 e L = 10 níveis de resolução para malhas diádicas e *quad-grids*, respectivamente.



Figura 7.4: Segundo Teste: À esquerda, parâmetro de truncamento em função da compressão usando esquemas de MR de Haar e MR de alta ordem. À direita, apresentamos a razão entre as memórias necessárias para armazenar a malha diádica e *quad-grid* em função do parâmetro de truncamento.

#### 7.3.2 Análise MR em Compressão de Imagem

Consideramos a imagem da Figura 4.4 e aplicamos o esquema MR para médias celulares em malhas diádicas, com previsão exata para polinômios de grau menor ou igual a 2.

Na Tabela 7.5, apresentamos os resultados de compressão e erro para os parâmetros de truncamento variando de 80 a 5, dividindo por 2. Observamos que a compressão diminui de 94,8% a 59,4%.

$\epsilon$	erro $(L_1)$	$\operatorname{compressão}(\%)$
80	7,73	94,8
40	5,81	$89,\! 6$
20	3,52	80,3
10	1,77	69,4
5	0,84	$59,\!4$

Tabela 7.5: Imagem da Figura 4.4: erro de reconstrução e compressão de dados, usando esquema MR em malhas diádicas, em função do parâmetro de truncamento.

Nas Figuras 7.5 e 7.6, apresentamos, à esquerda, a malha diádica adaptativa e, à direita, a imagem reconstruída, para  $\epsilon = 5$  e  $\epsilon = 80$ , respectivamente. Podemos observar o ganho de nitidez com o maior refinamento ao se usar um  $\epsilon$  menor.


Figura 7.5: À esquerda, malha diádica com  $\epsilon=$  5. À direita, imagem correspondente recuperada.



Figura 7.6: À esquerda a malha diádica com  $\epsilon=$  80, à direita imagem correspondente recuperada.

# Capítulo 8

# Esquema Adaptativo de Volumes Finitos para Malhas Diádicas

Como indicado no capítulo introdutório, uma aplicação de interesse dos esquemas MR refere-se aos métodos MR para a solução numérica adaptativa de equações diferenciais parciais para problemas em que podem ocorrer singularidades ou brusca variação no gradiente das soluções em regiões localizadas.

Neste capítulo apresentamos os conceitos envolvidos na construção de um método MR adaptativo que combina discretização por volumes finitos (FV) para leis de conservação e análise de MR para médias celulares, denominado esquema FV/MR. Em publicações anteriores, como descrito em Kaibara e Gomes [29], Cohen *et al.* [14], Müller [34] e Roussel *et al.* [39, 38], Domingues *et al.* [18, 17] e Müller e Stiriba [35], são usados esquemas MR clássicos, baseados em malhas  $2^d$ -ádicas. Nosso objetivo é estender esses esquemas para malhas diádicas e analisar o efeito desta nova forma de estrutura de dados.

Para os esquemas de referência em malha uniforme, fazemos a aproximação dos fluxos numéricos de duas formas distintas. A primeira combina o esquema de Roe [37] com o esquema essencialmente não oscilatório (ENO), proposto por Harten e Osher [27]. A outra forma que consideramos é o esquema essencialmente não oscilatório ponderado WENO [33, 28]. A integração no tempo é feita por meio dos Métodos de Runge-Kutta de segunda e terceira ordem. Apresentamos exemplos unidimensionais para a equação de advecção, com intuito de verificar a ordem de convergência dos esquemas.

## 8.1 Discretização por Volumes Finitos

Consideramos os problemas de leis de conservação escalares da forma

$$v_t = -\nabla \cdot f(v), \tag{8.1}$$

em que v = v(x,t) e  $(x,t) \in R \times [0,\infty)$ , com  $R \subset \mathbb{R}^d$ . Escolhemos as condições de contorno e iniciais de forma apropriada.

Fazemos a discretização espacial do problema (8.1) pelo método dos volumes finitos em sua forma conservativa. Isto é, considerando que a malha diádica  $G_L$  seja uma partição de R, então em cada célula  $c^L_{\alpha}$  temos

$$\frac{1}{|c_{\alpha}^{L}|} \int_{c_{\alpha}^{L}} v_{t} \mathrm{dx} = -\frac{1}{|c_{\alpha}^{L}|} \int_{c_{\alpha}^{L}} \nabla \cdot f(v) \mathrm{dx}$$

que, pelo teorema da divergência, pode ser reescrita na forma

$$\frac{\partial \bar{v}_{\alpha}}{\partial t} = \mathcal{D}_{\alpha}(v), \tag{8.2}$$

em que

$$\bar{v}_{\alpha}(t) = \frac{1}{|c_{\alpha}^{L}|} \int_{c_{\alpha}^{L}} v(x,t) \mathrm{dx}$$

é a média celular da solução em  $c_{\alpha}^{L}$  e

$$\mathcal{D}_{\alpha}(v) = -\frac{1}{|c_{\alpha}^{L}|} \int_{\partial c_{\alpha}^{L}} f(v) \cdot \sigma_{\alpha} \mathrm{ds}$$
(8.3)

é o fluxo na fronteira da célula, sendo  $\sigma_{\alpha}$  é o vetor normal unitário exterior a  $c_{\alpha}^{L}$ .

A solução numérica do problema diferencial (8.1) na malha  $G_L$  é denotado por  $u = u^L = (u_\alpha)$ , em que u é o vetor contendo as médias celulares aproximadas  $u_\alpha(t) \approx \bar{v}_\alpha(t)$  da solução de (8.1). Assim, considerando a equação (8.2) sobre todas as células folha, temos o sistema de equações diferenciais ordinárias

$$\frac{\partial u}{\partial t} = \bar{\mathcal{D}}(u), \tag{8.4}$$

em que  $\bar{\mathcal{D}}_{\alpha}(u) \approx \mathcal{D}_{\alpha}(\bar{v})$  denota o fluxo numérico.

Ainda resta resolver a questão da integração no tempo.

#### 8.1.1 Integração no Tempo

Para discretizar a equação (8.4) consideramos o esquema explícito de Runge-Kutta de segunda ordem e o esquema explícito de Runge-Kutta de terceira ordem. Para uma sequência  $t^n = n\Delta t$  de valores discretos de tempo, em que  $\Delta t$  é o comprimento de passo no tempo, denotamos por  $u^n$  a aproximação de u em  $t^n$ , isto é,  $u^n = u(t^n)$ . Com essa notação, o esquema de Runge-Kutta de segunda ordem tem a forma

$$u^{*} = u^{n} + \Delta t \bar{\mathcal{D}}(u^{n})$$
  

$$u^{n+1} = \frac{1}{2} \left[ u^{n} + u^{*} + \Delta t \bar{\mathcal{D}}(u^{*}) \right],$$
(8.5)

e o esquema explícito de Runge-Kutta de terceira ordem é dado por

$$u_{\alpha}^{(0)} = u_{\alpha}^{n},$$

$$u_{\alpha}^{(1)} = u_{\alpha}^{(0)} + \bar{\mathcal{D}}_{\alpha}(u_{\alpha}^{(0)}),$$

$$u_{\alpha}^{(2)} = \frac{3}{4}u_{\alpha}^{(0)} + \frac{1}{4}u_{\alpha}^{(1)} + \frac{1}{4}\bar{\mathcal{D}}_{\alpha}(u_{\alpha}^{(1)})$$

$$u_{\alpha}^{n+1} = \frac{1}{3}u_{\alpha}^{(0)} + \frac{2}{3}u_{\alpha}^{(2)} + \frac{2}{3}\bar{\mathcal{D}}_{\alpha}(u_{\alpha}^{(2)}).$$
(8.6)

Observamos que os esquemas Runge-Kutta são TVD (Total Variation Diminishing), ou seja, a variação total no tempo n é menor ou igual à variação total no tempo n + 1. Em símbolos,

$$TV(u^n) = \sum_{\alpha} |u_{\alpha+1} - u_{\alpha}| \ge TV(u^{n+1}).$$
 (8.7)

Mais detalhes podem ser encontrados em Shu e Osher [41]. Tal característica faz com que a solução numérica se dissipe ao longo do tempo. Por exemplo, com o passar do tempo, a solução numérica vai ficando cada vez mais suave nas regiões de descontinuidade e as soluções contínuas têm seu valor de máximo (ou mínimo) diminuído (aumentado). Isto é, no passo de tempo  $t^{n+1}$ , o valor de máximo da solução é menor que o valor de máximo da solução em  $t^n$ .

#### 8.1.2 Fluxo Numérico

Para as operações efetuadas no decorrer desta seção, supomos que o tempo  $t^n$  está fixo, e que todas as células estão no mesmo nível de refinamento L, o que torna desnecessário o uso dos índices  $n \in L$ . Assim, a célula  $c_{\alpha}$  é dada por um intervalo na reta da forma  $(x_{\alpha-\frac{1}{2}}, x_{\alpha+\frac{1}{2}})$  e denotamos por  $\Delta x_{\alpha} = (x_{\alpha+\frac{1}{2}} - x_{\alpha-\frac{1}{2}})$  o volume de  $c_{\alpha}$ . A fim de simplificar a exposição dos cálculos do fluxo numérico, consideramos inicialmente um domínio espacial unidimensional. Como visto mais adiante, a aproximação do fluxo numérico em dimensões superiores se baseia no caso unidimensional, por meio de produto tensorial.

No caso unidimensional, a equação (8.2) é dada por

$$\frac{\partial u_{\alpha}}{\partial t} = -\frac{f(u(x_{\alpha+\frac{1}{2}},t)) - f(u(x_{\alpha-\frac{1}{2}},t))}{\Delta x_{\alpha}} = \mathcal{D}_{\alpha}(u).$$
(8.8)

Denotando por  $F_{\alpha+\frac{1}{2}}$  uma aproximação para  $f(u(x_{\alpha+\frac{1}{2}},t))$  temos a equação

$$\frac{\partial u_{\alpha}}{\partial t} = -\frac{1}{\Delta x_{\alpha}} \left[ F_{\alpha + \frac{1}{2}} - F_{\alpha - \frac{1}{2}} \right].$$

Em geral, o cálculo de  $F_{\alpha \pm \frac{1}{2}}$  envolve dois ou mais valores de  $u_{\alpha}(t)$ , fornecendo um sistema acoplado de equações diferenciais ordinárias como em (8.4).

Dessa forma, da equação (8.8) temos

$$\bar{\mathcal{D}}_{\alpha} = -\frac{1}{\Delta x_{\alpha}} \left[ \bar{F}_{\alpha + \frac{1}{2}} - \bar{F}_{\alpha - \frac{1}{2}} \right], \qquad (8.9)$$

em que  $\bar{F}_{\alpha \pm \frac{1}{2}}$  são os fluxos numéricos nas interfaces esquerda (-) e direita (+) da célula  $c_{\alpha}$ .

Para a definição do fluxo numérico, usamos dois esquemas: esquema de Roe [37] combinado com esquema essencialmente não oscilatório (ENO), proposto por Harten e Osher [27]. Mais tarde, Bihari [1] mostra que tal combinação dá um esquema interpolatório de segunda ordem. Outra forma de definir os fluxos numéricos que consideramos é o esquema essencialmente não oscilatório ponderado WENO [33, 28].

#### Esquema de Roe + ENO

O fórmula do fluxo de Roe é dada por

$$\bar{F}_{\alpha+\frac{1}{2}} = f_R(u_{\alpha+\frac{1}{2}}^-, u_{\alpha+\frac{1}{2}}^+), \tag{8.10}$$

em que o termo  $f_R$  representa a solução aproximada de Roe para o problema de Riemann com valores para u dados por  $u^-$  (esquerda) e  $u^+$  (direita) da interface  $x_{\alpha+\frac{1}{2}}$ , veja Figura 8.1. Em sua versão escalar a solução para o problema de Riemann é dada por



Figura 8.1: Reconstrução linear por partes de  $u(x_0)$ , usada para definir os valores  $u_{\alpha+\frac{1}{2}}^-$  e  $u_{\alpha+\frac{1}{2}}^+$  na interface  $c_{\alpha+\frac{1}{2}}$ .

$$f_R(u^-, u^+) = \frac{1}{2} \left[ f(u^-) + f(u^+) - |a(u^-, u^+)|(u^+ - u^-) \right], \qquad (8.11)$$

em que

$$a(u^{-}, u^{+}) = \begin{cases} \frac{f(u^{+}) - f(u^{-})}{u^{+} - u^{-}} & \text{se } u^{+} \neq u^{-}, \\ f'(u^{-}) & \text{se } u^{+} = u^{-}. \end{cases}$$

Para determinar completamente a equação (8.10), ainda nos resta escolher os termos  $u_{\alpha+\frac{1}{2}}^+ e u_{\alpha+\frac{1}{2}}^-$ . Essa tarefa é desempenhada pelo método de interpolação ENO de segunda ordem dado por (ver Bihari [1])

$$u_{\alpha+\frac{1}{2}}^{-} = u_{\alpha} + M(u_{\alpha+1} - u_{\alpha}, u_{\alpha} - u_{\alpha-1}),$$
  

$$u_{\alpha+\frac{1}{2}}^{+} = u_{\alpha+1} - M(u_{\alpha+2} - u_{\alpha+1}, u_{\alpha+1} - u_{\alpha}),$$
(8.12)

em que

$$M(\omega, \rho) = \begin{cases} \omega & \text{se } |\omega| \le |\rho| \\ \rho & \text{se } |\omega| > |\rho|, \end{cases}$$
(8.13)

mede a menor inclinação entre os lados direito e esquerdo. De forma geral, a idéia central dos esquemas ENO é a de escolher o *gabarito* mais suave entre alguns candidatos usados para aproximar os fluxos na fronteira de uma célula, a fim de evitar oscilações e, ao mesmo tempo, obter uma aproximação de alta ordem nas regiões próximas aos choques.

#### Esquemas WENO

Os esquemas WENO foram introduzidos por Liu *et al.* [33] e, posteriormente, modificado por Jiang *et al.* [28], entre outros. Nos esquemas WENO, para definirmos o fluxo numérico, ao invés de usar apenas um dos candidatos a gabarito, usamos uma combinação convexa de todos os candidatos. A cada candidato é atribuído um peso que determina sua contribuição à aproximação do fluxo numérico. Os pesos são escolhidos de forma a se obter aproximações de alta ordem em regiões de suavidade (r-ésima ordem para esquemas ENO [33] e 2r - 1-ésima ordem para esquemas WENO [33], em que r, para o caso 1D, é a quantidade de pontos no gabarito subtraído uma unidade) e para regiões próximas à descontinuidade, os gabaritos que contêm as descontinuidades recebem peso próximo a zero.

Para descrever o esquema WENO, consideramos a lei de conservação escalar unidimensional. Longe dos pontos sônicos, existem dois casos para serem considerados na definição de  $\mathcal{D}_{\alpha}(u)$ .

• Se  $f'(\mathcal{R}(x)) > 0$ 

$$\bar{\mathcal{D}}_{\alpha}(u) = -\frac{1}{\Delta x_{\alpha}} [f(\mathcal{R}_{\alpha}(x_{\alpha+\frac{1}{2}})) - f(\mathcal{R}_{\alpha-\frac{1}{2}}(x_{\alpha-\frac{1}{2}}))]; \qquad (8.14)$$

• Se  $f'(\mathcal{R}(x)) < 0$ 

$$\bar{\mathcal{D}}_{\alpha}(u) = -\frac{1}{\Delta x_{\alpha}} [f(\mathcal{R}_{\alpha+\frac{1}{2}}(x_{\alpha+\frac{1}{2}})) - f(\mathcal{R}_{\alpha}(x_{\alpha-\frac{1}{2}}))], \qquad (8.15)$$

em que a função  $\mathcal{R}$  é definida de acordo com a ordem de aproximação que se queira.

Em geral,

$$\bar{\mathcal{D}}_{\alpha}(u) = -\frac{1}{\Delta x_{\alpha}} [\tilde{h}(\mathcal{R}_{\alpha}(c_{\alpha+\frac{1}{2}}), \mathcal{R}_{\alpha+1}(c_{\alpha+\frac{1}{2}})) - \tilde{h}(\mathcal{R}_{\alpha-1}(c_{\alpha-\frac{1}{2}}), \mathcal{R}_{\alpha}(c_{\alpha-\frac{1}{2}}))], \qquad (8.16)$$

em que  $\tilde{h}$  é uma função Lipschitz nas duas variáveis, monótona não decrescente no primeiro argumento e não crescente para o segundo argumento.

Algumas escolhas possíveis são

$$\tilde{h}(a,b) = \begin{cases} \min_{a \le u \le b} f(u) & \text{se } a \le b, \\ \max_{a \le u \le b} f(u) & \text{se } a > b, \end{cases}$$

$$\tilde{h}(a,b) = \begin{cases} f(a) & \text{se } f'(u) \ge 0, \min(a,b) \le u \le \max(a,b), \\ f(b) & \text{se } f'(u) \le 0, \min(a,b) \le u \le \max(a,b), \\ h(a,b) & \text{caso contrário}, \end{cases}$$

$$(8.17)$$

em que

$$h(a,b) = \frac{1}{2} \left[ f(a) + f(b) - \beta(b-a) \right],$$
(8.19)

$$\beta = \max_{\min(a,b) \le u \le \max(a,b)} |f'(u)| \tag{8.20}$$

#### Esquema WENO para r = 2

Apresentamos o esquema de aproximação  $\mathcal{R}$  no caso em que r = 2, introduzido por Liu *et al.* [33].

Para cada célula  $c_{\alpha} = (x_{\alpha-\frac{1}{2}}, x_{\alpha+\frac{1}{2}})$  consideramos os gabaritos  $S_{\alpha} = (x_{\alpha-\frac{3}{2}}, x_{\alpha-\frac{1}{2}}, x_{\alpha+\frac{1}{2}})$ ,  $S_{\alpha+1} = (x_{\alpha-\frac{1}{2}}, x_{\alpha+\frac{1}{2}}, x_{\alpha+\frac{3}{2}})$ , nos quais definimos os seguintes polinômios interpolatórios

$$p_{\alpha}(x) = u_{\alpha} + \frac{u_{\alpha} - u_{\alpha-1}}{\Delta x_0} (x - x_{\alpha-1}),$$
  

$$p_{\alpha+1}(x) = u_{\alpha} + \frac{u_{\alpha+1} - u_{\alpha}}{\Delta x_0} (x - x_{\alpha}),$$
(8.21)

e dois indicadores de regularidade  $S_{\alpha} = (u_{\alpha} - u_{\alpha-1})^2$  e  $S_{\alpha+1} = (u_{\alpha+1} - u_{\alpha})^2$ .

Com uma combinação convexa dos polinômios (8.21) obtemos a expressão de  $\mathcal{R}_{\alpha}(x)$ , isto é,

$$\mathcal{R}_{\alpha}(x) = \frac{\nu_0^{\alpha}}{\nu_0^{\alpha} + \nu_1^{\alpha}} p_{\alpha}(x) + \frac{\nu_1^{\alpha}}{\nu_0^{\alpha} + \nu_1^{\alpha}} p_{\alpha+1}(x), \qquad (8.22)$$

em que os coeficientes  $\nu_0^{\alpha}$  e  $\nu_1^{\alpha}$  são avaliados da seguinte forma: Caso 1. Se  $f'(u_{\alpha}) > 0$ 

$$\nu_0^{\alpha} = \frac{1}{2(\epsilon + S_{\alpha})^2},$$

$$\nu_1^{\alpha} = \frac{1}{(\epsilon + S_{\alpha+1})^2},$$
(8.23)

em que  $\epsilon = 0.000001$ .

Caso 2. Se  $f'(u_{\alpha}) \leq 0$ 

$$\nu_0^{\alpha} = \frac{1}{2(\epsilon + S_{\alpha})^2},$$

$$\nu_1^{\alpha} = \frac{1}{(\epsilon + S_{\alpha+1})^2},$$
(8.24)

em que  $\epsilon = 0.000001$ .

#### 8.1.3 Fluxo em Dimensões Superiores

A extensão do cálculo do fluxo numérico para dimensões superiores é feita pelo produto tensorial do cálculo do fluxo unidimensional. Em outras palavras, aplicamos o método de cálculo do fluxo unidimensional em cada uma das direções dos eixos coordenados.

No caso bidimensional, a forma de escrever a equação (8.4) permanece inalterada. O comprimento da aresta da célula  $c_{\alpha}$  na direção i = 0, 1 é dado por  $\Delta x_i$ .

Neste caso,

$$\bar{\mathcal{D}}_{\alpha} = -\frac{1}{\Delta x_0} \left( \bar{F}^0_{\alpha + \frac{1}{2}} - \bar{F}^0_{\alpha - \frac{1}{2}} \right) - \frac{1}{\Delta x_1} \left( \bar{F}^1_{\alpha + \frac{1}{2}} - \bar{F}^1_{\alpha - \frac{1}{2}} \right),$$

em que o fluxo numérico através da fronteira superior (ver Seção 3.2.1), em relação ao eixo i, é dado por

$$\bar{F}^{i}_{\alpha+\frac{1}{2}} = f^{i}_{R}(u^{-}_{\alpha+\frac{1}{2}}, u^{+}_{\alpha+\frac{1}{2}}),$$

sendo que o termo  $f_R^0$  representa a solução aproximada para o problema de Riemann na aresta direita de  $c_{\alpha}$ , com valores para u dados por  $u^-$  (esquerda da aresta) e  $u^+$  (direita da aresta) e  $f_R^1$  representa a solução aproximada para o problema de Riemann na aresta superior de  $c_{\alpha}$ , com valores para u dados por  $u^-$  (abaixo da aresta) e  $u^+$  (acima da aresta). Ambos termos são dados por (8.11) para o fluxo de Roe.

No caso WENO,  $\bar{F}^{i}_{\alpha+\frac{1}{2}}$ , i = 0, 1, são calculados por (8.17) ou por (8.18).

Para dimensões superiores o procedimento é análogo.

## 8.2 Adaptatividade e o Esquema de Volumes Finitos

Para melhorar o desempenho do método dos volumes finitos, tanto em tempo de execução dos algoritmos quanto na compressão de dados, é comum representar a solução  $u^n$  em malhas esparsas adaptativas, ao invés de malhas regulares. No esquema MR, a solução numérica  $u_{MR}^n$  é formada pelas médias celulares em uma malha adaptativa  $G^n$ . A malha adaptativa  $G^n$  está associada a uma estrutura de árvore incompleta, em que o refinamento de uma célula pode ser interrompido antes que o nível máximo de resolução seja alcançado.

Para obtermos a solução  $u_{MR}^{n+1}$  a partir da solução  $u_{MR}^n$  são necessários três estágios. A saber: refinamento, evolução e truncamento.

**Refinamento.** O primeiro dos três passos para obtermos a solução  $u_{MR}^{n+1}$  a partir da solução  $u_{MR}^n$  é o refinamento da malha  $G^n$ .

Essa etapa é necessária porque a malha  $G^n$ , que é boa para representar  $u_{MR}^n$ , pode não ser indicada para representar  $u_{MR}^{n+1}$ . É de se esperar, por exemplo, que as posições das singularidades na solução de (8.1) mudem ao se passar de  $t^n$  para o tempo  $t^{n+1}$ .

Assim, antes de que seja feita a evolução no tempo, fazemos o refinamento da malha  $G^n$  e obtemos uma malha  $G^{n+}$ . Para construir  $G^{n+}$ , cada célula folha  $c^{\ell}_{\alpha}$  de profundidade  $\ell < L$ , contida na malha  $G^n$ , é dividida por um hiperplano ortogonal ao eixo  $i = (\ell \mod d)$ . Esperamos que o passo de tempo  $\Delta t$  seja pequeno o suficiente para que a malha  $G^{n+1}$  seja obtida a partir de  $G^{n+}$ .

A operação de refinamento é identificada por  $\mathbf{R}$  e  $G^{n+} = \mathbf{R}G^n$ . Os valores das médias celulares nas folhas de  $G^{n+}$  são preenchidos usando o operador de previsão (5.5) e a solução numérica sobre  $G^{n+}$  é dada por  $u^{n+} = \mathbf{R}u^n_{MR}$ .

**Evolução.** O operador de evolução discreta  $\mathbf{E}_{MR} = \mathbf{E}_{MR}(\Delta t)$  é aplicado nas folhas de  $G^{n+}$ . Tal operador é dado pelo esquema de Runge-Kutta de segunda ordem (8.5), que pode ser escrito na seguinte forma,

$$\mathbf{E}_{MR} = \mathbf{I} + \frac{\Delta t}{2} \left[ \bar{\mathcal{D}}_a + \bar{\mathcal{D}}_a (\mathbf{I} + \Delta t \bar{\mathcal{D}}_a) \right],$$

ou pelo esquema de Runge-Kutta de terceira ordem (8.6)

$$\mathbf{E}_{MR} = \left(1 + \frac{2\Delta t}{3}\right)\mathbf{I} + \frac{\Delta t}{6}\left[\bar{\mathcal{D}}_{\alpha} + \Delta t\bar{\mathcal{D}}_{\alpha} + \bar{\mathcal{D}}_{\alpha}\left(\mathbf{I} + \Delta t\bar{\mathcal{D}}_{\alpha}\right) + \Delta t\bar{\mathcal{D}}_{\alpha}\left(\mathbf{I} + \Delta t\bar{\mathcal{D}}_{\alpha}\right)\right],$$

em que I é o operador identidade e  $\overline{\mathcal{D}}_a$  é calculado em cada folha da seguinte forma (ver Figura 8.4).

- Caso todos os vizinhos da célula folha  $c^{\ell}_{\alpha}$  sejam também folhas e que estejam em nível igual ou inferior ao nível de  $c^{\ell}_{\alpha}$ , os fluxos são calculados como na malha uniforme de mesmo nível, interpolando as médias celulares ausentes, se necessário.
- Para cada direção *i*, caso os vizinhos (de  $c_{\alpha}^{\ell}$ )  $c_{\beta}^{\ell} <_i c_{\alpha}^{\ell} <_i c_{\gamma}^{\ell}$  sejam refinados, definimos os fluxos de saída de  $c_{\alpha}^{\ell}$  na direção *i* pela soma dos fluxos de entrada nos descendentes de  $c_{\gamma}^{\ell}$  que sejam folhas e que a interseção da fronteira de  $c_{\alpha}^{\ell}$  com a fronteira desses descendentes é não vazia. Analogamente, os fluxo de entrada em  $c_{\alpha}^{\ell}$  na direção *i* é definido pela soma dos fluxos de saída dos descendentes de  $c_{\beta}^{\ell}$  que sejam folhas e que a interseção da fronteira de secendentes de calcular entrada em calcular entrada en calcular entrada entrada entrada entrada entrada entrada enter entrada en

Assim, a solução numérica no tempo  $t^{n+1}$  sobre a malha  $G^{n+}$ , em termos do operador de evolução, é dada por

$$\tilde{u}^{n+1} = \mathbf{E}_{MR} u^{n+1}.$$

**Truncamento.** Finalmente, para obter a solução  $u_{MR}^{n+1}$  a partir de  $\tilde{u}^{n+1}$ , usamos o operador de restrição para atualizar os valores dos nós da árvore e, em seguida, o operador de truncamento  $\mathbf{T}(\epsilon)$ . Isto é, para cada célula  $c_{\alpha}^{\ell}$ ,  $u_{\alpha}^{\ell} = \frac{u_{\alpha_R}^{\ell+1} + u_{\alpha L}^{\ell+1}}{2}$ . O operador  $\mathbf{T}$  varre a malha  $G^{n+}$  do nível mais fino para o mais grosso, nível por nível, apagando as células folha que são desprezíveis para a representação de  $u_{MR}^{n+1}$ .

Em resumo, para obtermos  $u_{MR}^{n+1}$  a partir de  $u_{MR}^n$ , basta seguir o seguinte procedimento

$$u_{MR}^{n+1} = \mathbf{T}(\epsilon) \mathbf{E}_{MR} \mathbf{R} u_{MR}^n.$$

#### 8.2.1 Estrutura de Dados

Consideramos que  $R \subset \mathbb{R}^d$  é uma caixa *d*-dimensional. A estrutura de dados que representa uma malha diádica adaptativa em R é uma árvore binária do tipo  $\{0, 2\}$ , ver Seção 3.1. Esta estrutura é dinâmica, isto é, sua forma depende da malha a qual representa.

Na estrutura de árvore, fazer a compressão de dados significa apagar os nós folha onde os detalhes são menores que uma tolerância previamente especificada. Em outras palavras, o operador (de truncamento)  $\mathbf{T}(\epsilon)$ , definido na seção anterior, apaga as folhas nas quais os detalhes são desprezíveis. O único requisito a ser cumprido pelo operador  $\mathbf{T}(\epsilon)$  é que, se uma célula é folha, então sua célula irmã deve ser mantida na árvore. Tal exigência garante que a ávore binária continue do tipo  $\{0, 2\}$  e a malha diádica que é representada por essa árvore forme uma partição para R.

**Estrutura da Malha.** Diferentes propriedades podem ser exigidas da estrutura de árvore para incrementar a eficiência do método. Por exemplo, em Cohen *et al.* [12, 15], Domingues et al. [17, 18], Müller [34], Roussel *et al* [39, 38] é adotada uma estrutura denominada *estrutura gradual*, em que o valor absoluto da diferença entre os níveis de células folha vizinhas é menor ou igual 1, e que será adotada nos experimentos do próximo capítulo.

**Pacotes de Células.** Para trabalhar com árvores incompletas, as células necessárias ao cálculo do fluxo numérico podem não estar na árvore e esses valores precisam ser interpolados. Para tratar esses casos no caso do esquema MR de alta ordem em malhas diádicas, propomos uma rotina que calcula o fluxo numérico na célula  $c_{\alpha}^{\ell}$  recebendo um pacote de dados, em que cada entrada representa uma célula vizinha a  $c_{\alpha}^{\ell}$ . Por exemplo, com previsão exata para polinômios de grau  $\leq 2$ , usamos um pacote de dados com 5<sup>d</sup> entradas. Na Figura 8.2, à esquerda, destacamos um pacote de dados em uma árvore binária para o caso unidimensional. No lado esquerdo da Figura 8.2, as setas indicam o caminho a ser percorrido na árvore para obtermos o pacote em destaque. Os nós 104

utilizados na composição do pacote em destaque estão representados por círculos e as curvas pontilhadas representam os nós que precisaram ter valores interpolados.

Na Figura 8.2, à direita, apresentamos o pacote de células em duas dimensões. No centro do pacote, está a célula  $c_{\alpha}^{\ell}$  e, à sua volta, estão os dois vizinhos mais próximos em cada direção. A justificativa de usarmos um conjunto de dados com tantas entradas quanto mencionadas acima vem da necessidade de fazermos uma eventual previsão das médias celulares dos descendentes de  $c_{\alpha}^{\ell}$ .

Previsão de Médias Celulares. Para ilustrarmos o processo de previsão das médias celulares nos descendentes de  $c_{\alpha}^{\ell}$ , consideramos o pacote de células da Figura 8.2 e, sem perda de generalidade, supomos que a próxima divisão das células é feita por hiperplanos ortogonais ao eixo i = 0.

Sabemos que a previsão da média celular de uma função u na célula  $c_{\alpha R}^{\ell+1}$ , filho superior de  $c_{\alpha}^{\ell}$ , é feita usando as médias celulares de  $c_{\gamma_0}^{\ell}$  e  $c_{\beta_0}^{\ell}$  (veja Figura 8.3 (a) e equação (5.5)). Suponha que, no próximo nível de refinamento, seja necessário prever a média celular do filho superior de  $c_{\alpha R}^{\ell+1}$ . Para tal, precisamos conhecer as médias celulares dos filhos de  $c_{\gamma_1}^{\ell}$  e  $c_{\beta_1}^{\ell}$ , conforme mostrado na Figura 8.3 (c). No entanto, caso precisamos prever as médias celulares dos filhos de  $c_{\gamma_1}^{\ell}$  e  $c_{\beta_1}^{\ell}$ , é fundamental conhecer as médias dos dois vizinhos mais próximos a  $c_{\gamma_1}^{\ell}$  e  $c_{\beta_1}^{\ell}$ , (veja Figura 8.3 (b)). Assim, para interpolar a média celular de duas gerações de descendentes é necessário um pacote de dados com 3<sup>d</sup> informações. Isto sugere que passar um conjunto de 5<sup>d</sup> dados é desperdício de recursos. No entanto, para montar o pacote de 3<sup>d</sup> células com centro em  $c_{\alpha R}^{\ell+1}$  precisamos, por exemplo, conhecer a média dos filhos de  $c_{\gamma_0}^{\ell}$  e para interpolar tais valores, a média de  $c_{\gamma}^{\ell}$  precisa ser conhecida (veja Figura 8.3 (d)). Portanto, usando um argumento análogo para cada uma das células do pacote que tem fronteira comum com  $c_{\alpha}^{\ell}$ , vemos que há necessidade de um pacote de dados com 5<sup>d</sup> células.

Cálculo Adaptativo do Fluxo Numérico. Para ilustrar o cálculo adaptativo do fluxo no caso bidimensional, consideramos dois casos.



Figura 8.2: Pacotes de células. À esquerda, um pacote unidimensional sobre uma árvore em que os círculos representam os nós usados para construção do pacote em destaque, as setas indicam o caminho percorrido para a construção do pacote e a linha pontilhada representa os valores que precisaram ser interpolados. À direita, um pacote de células bidimensional centrado em  $c_{\alpha}^{\ell}$ .

O primeiro, na coluna (a) da Figura 8.4, as células folha  $c_{\alpha_L}^{\ell+1}$ ,  $c_{\alpha_R}^{\ell+1} \in c_{\gamma}^{\ell}$  estão em níveis diferentes de refinamento e a diferença entre os níveis é de apenas uma unidade. Neste caso, para que haja conservação dos fluxos, definimos que a soma dos fluxos que saem de  $c_{\alpha_R}^{\ell+1} \in c_{\alpha_L}^{\ell+1}$  é igual ao fluxo que entra em  $c_{\gamma}^{\ell}$ . Isto quer dizer que,

$$F_{c_{\alpha}^{\ell}}^{c_{\gamma}^{\ell}} = F_{c_{\alpha_R}^{\ell+1}}^{c_{\gamma}^{\ell}} + F_{c_{\alpha_L}^{\ell+1}}^{c_{\gamma}^{\ell}},$$

em que  $F_{c_{\alpha_R}^{\ell+1}}^{c_{\alpha}^{\ell}}$  presenta o fluxo que vai da célula  $c_{\alpha_R}^{\ell+1}$  para a célula  $c_{\alpha}^{\ell}$ .

De maneira análoga, na coluna (b) da Figura 8.4, as células folha  $c_{\alpha_L}^{\ell+1}$ ,  $c_{\alpha_R}^{\ell+1}$  e  $c_{\gamma}^{\ell}$  estão em níveis diferentes de refinamento, com diferença entre os níveis de uma unidade. Para que tenhamos a conservação dos fluxos, definimos

$$F_{c_{\alpha}^{\ell}}^{c_{\gamma}^{\ell}} = F_{c_{\alpha_R}^{\ell+1}}^{c_{\gamma}^{\ell}}$$

em que  $F_{c_{\alpha_R}^{\ell_{\alpha}}}^{c_{\alpha}^{\ell}}$  presenta o fluxo que vai da célula  $c_{\alpha_R}^{\ell+1}$  para a célula  $c_{\alpha}^{\ell}$ .



Figura 8.3: (a) Células usadas para prever o valor da média de um sinal em  $c_{\alpha_R}^{\ell+1}$ . (b) Células usadas para prever o valor da média de um sinal nos filhos de  $c_{\gamma_1}^{\ell} e c_{\beta_1}^{\ell}$ . (c) Células usadas para prever o valor da média de um sinal no filho de  $c_{\alpha_R}^{\ell+1}$ . (d) Células usadas para prever o valor da média de um sinal no filho de  $c_{\gamma_0}^{\ell}$ .



Figura 8.4: Cálculo do fluxo numérico adaptativo e conservativo quando as células estão em níveis de resolução consecutivos quando (a) há duas células vizinhas e (b) há uma única célula vizinha.

### 8.3 Implementação do Algoritmo

O algoritmo de integração pode ser descrito como segue. Ele depende de seis parâmetros principais — a dimensão d, a célula raiz R, o tempo total de integração T, o fator CFL  $\sigma = \Delta t / \Delta x$ , o nível máximo de resolução L, e o parâmetro de truncamento  $\epsilon$  — bem como a função fluxo f e o estado inicial  $v^0(x) = v(x, 0)$ .

- Calcula o comprimento máximo I do domínio R em cada direção, ao longo de cada eixo, o tamanho mínimo de cada célula Δx = I/2<sup>L</sup>, o número de iterações N = [T/(σΔx)], e passo no tempo Δt = T/N.
- Cria uma estrutura de dados (árvore binária) uniforme com profundidade L. Grava em cada célula a média do estado inicial v<sup>0</sup>. Poda os nós desta árvore que têm detalhes menores que o parâmetro de truncamento ε, como descrito na Seção 8.2.
- Para n, de 0 até N 1, faz:
  - Divide cada célula folha da malha que não esteja no nível máximo L. Estima as médias celulares nestas células usando operador de previsão (5.5).
  - Calcula o fluxo numérico em cada célula folha, como detalhado na Seção 8.2.

- Aplica o esquema de Runge-Kutta de segunda ou terceira ordem (8.5) em cada célula folha, para estimar a média celular no tempo n + 1.
- Atualiza a média celular de cada nó interno da estrutura, do nível mais fino para o mais grosso, pelo operador de restrição.
- Poda os nós desta árvore que estão associados a detalhes abaixo do parâmetro de truncamento  $\epsilon$ , como descrito na Seção 8.2.

#### 8.3.1 Exemplos Unidimensionais

Apresentamos alguns exemplos para verificar numericamente a ordem de convergência dos métodos de Roe + ENO + RK2 e WENO + RK3, este com ordem 3 e aquele com ordem 2, aplicados na resolução da equação de advecção

$$u_t + u_x = 0,$$
  $(x, t) \in (-1, 1) \times \{t > 0\},$   
 $u(x, 0) = u_0(x),$   $u(-1, t) = u(1, t),$ 

em que as condições de contorno são periódicas.

Esquema Roe + ENO + RK2 Resolvemos o problema de advecção com a condição inicial dada por

$$u_0(x) = \exp(-100x^2),$$

no instante final  $T = 0, 5 \in CFL = 0, 5$ , com diferentes níveis de resolução.

Na Tabela 8.1, apresentamos os erros em função do nível de resolução. Na coluna referente à inclinação, verificamos que a taxa de convergência se aproxima de 2, em acordo com os resultados teóricos [1].

Na Figura 8.5, na parte superior, apresentamos os gráficos da solução exata e da solução numérica de referência sobre a malha uniforme de  $2^{12} = 4096$  células. Para comparação, plotamos também a solução adaptativa usando  $\epsilon = 0,001$ , mostrando uma boa aproximação de ambas soluções numéricas com a solução exata. Na parte inferior,



Figura 8.5: Equação de advecção 1D: na parte superior, solução exata, solução numérica de referência e solução adaptativa, usando esquema de Roe + ENO + RK2, no instante T = 0, 5, malha uniforme com  $2^{12}$  células, CFL = 0, 5 e  $\epsilon = 0,001$ . Na parte inferior, diferença entre solução adaptativa e solução exata.

Nível	erro	h	inclinação
9	$2,13*10^{-3}$	$3,90*10^{-3}$	
10	$6,04*10^{-4}$	$1,95*10^{-3}$	1,82
11	$1,64*10^{-4}$	$9,76*10^{-4}$	1,88
12	$4,39*10^{-5}$	$4,88 * 10^{-4}$	1,90
13	$1,15*10^{-5}$	$2,44*10^{-4}$	1,92

Tabela 8.1: Equação de Advecção 1D: taxa de decaimento do erro entre soluções exata e FV em função do nível de resolução. Esquema de Roe + ENO + RK2 com CFL = 0, 4 e T = 0, 5.

vemos a diferença entre a solução analítica e a solução adaptativa, da ordem do parâmetro de truncamento.

Na Figura 8.6, apresentamos os gráficos que ilustram o comportamento da eficiência do esquema adaptativo em termos de tempo de CPU, compressão de dados e erro de reconstrução em função do parâmetro de truncamento. Observamos que, à medida que o parâmetro de truncamento fica menor, o método de MR tem um ganho de 50 a 15 vezes no tempo de CPU. Este sobre malha MR com  $\epsilon = 0,001$  e aquele com  $\epsilon = 0,1$ , correspondente a uma compressão de dados de 96% e 99%, respectivamente. Ao centro, vemos que a diferença entre a solução adaptativa e a solução de referência, medida na norma  $L_1$ , fica da mesma ordem da variação do parâmetro de truncamento.

**Esquema WENO + RK3** Resolvemos o problema de advecção unidimensional com a condição inicial e condição de contorno dadas, respectivamente, por

$$u_0(x) = \sin(\pi x),$$
 (8.25)

no instante final T = 1 e CFL = 0, 8, com diferentes níveis de resolução.

Liu *et al.* [33] mostram que a ordem de convergência teórica é 3. Contudo, na Tabela 8.2, na coluna referente à inclinação, vemos que a ordem 4 da taxa de convergência é alcançada, o que significa que podemos obter uma ordem de aproximação melhor que a garantida pelos resultados teóricos.

Na Figura 8.7, na parte superior, apresentamos os gráficos da solução exata da solução numérica de referência, sobre a malha uniforme de  $2^{12} = 4096$  células. Para comparação,



Figura 8.6: Equação de Advecção 1D: ganho de CPU do algoritmo de MR (acima), compressão de dados (centro) e erro (abaixo) como função do parâmetro de truncamento, usando esquema de Roe + ENO + RK2, malha mais refinada de  $2^{12}$  células e CFL = 0, 5.

Nível	erro	h	inclinação
6	$1,54*10^{-2}$	$3,12*10^{-2}$	
7	$2,40*10^{-3}$	$1,56*10^{-2}$	2,68
8	$2,01*10^{-4}$	$7,81*10^{-3}$	3,57
9	$1,16*10^{-5}$	$3,90*10^{-3}$	4, 12
10	$5,97*10^{-7}$	$2,44*10^{-4}$	4,27
11	$2,53*10^{-8}$	$9,76*10^{-4}$	4,55

Tabela 8.2: Equação de Advecção 1D: taxa de decaimento do erro entre soluções exata e FV em função do nível de resolução. Esquema WENO + RK3 para  $T = 1, 0 \in CFL = 0, 8$ .

plotamos também a solução adaptativa usando  $\epsilon = 0,001$ , mostrando uma boa aproximação de ambas soluções numéricas com a solução exata. Na parte inferior, vemos a diferença entre a solução analítica e a solução adaptativa, da ordem do parâmetro de truncamento.

Na Figura 8.8, apresentamos os gráficos que ilustram o comportamento da eficiência e diferença entre solução MR e FV, na norma  $L_1$ , em função do parâmetro de truncamento. À direita, observamos que o ganho é 68 vezes quando  $\epsilon = 0, 1$  e de 20 vezes quando  $\epsilon = 0,001$ . Ao centro, vemos que a diferença entre a solução FV e a solução MR, na norma  $L_1$ , é proporcional ao parâmetro de truncamento. À esquerda, observamos que a compressão é de 99% para  $\epsilon = 0, 1$  e 96% para  $\epsilon = 0,001$ .



Figura 8.7: Equação de Advecção 1D: na parte superior, solução exata, solução numérica de referência e solução adaptativa, usando esquema WENO + RK3, no instante T = 1, malha uniforme com  $2^{12}$  células, CFL = 0, 8 e  $\epsilon = 0,001$ . Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.



Figura 8.8: Equação de Advecção 1D: ganho de CPU do algoritmo de MR (abaixo), compressão de dados (acima) e diferença entre MR e FV (centro) como função do parâmetro de truncamento, usando esquema WENO + RK3, malha mais refinada de  $2^{12}$  células e CFL = 0, 8.

# Capítulo 9

# Esquema FV/MR para Malhas Diádicas: Resultados Numéricos

O objetivo deste capítulo é fazer um estudo da eficiência do esquema FV/MR adaptativo utilizando a estrutura de dados de árvore binária de malhas diádicas bidimensionais.

Para o esquema de volumes finitos de referência, consideramos diferentes versões, dependendo da escolha do fluxo numérico e do esquema de evolução temporal, como descrito no capítulo anterior. A saber, usamos o esquema de Roe + ENO + RK2 (Seção 8.1.2) e o esquema WENO + RK3 (Seção 8.1.2).

Para os testes, consideramos problemas de valor inicial e de contorno para a equação de advecção, a equação de Burgers e a equação de Buckley-Leverett, em 2D.

Em cada caso, a solução numérica adaptativa é comparada com a solução numérica de referência, esta calculada sobre a malha uniforme do nível mais refinado. Analisamos a compressão de dados, o erro com relação à solução de referência e o ganho de tempo de CPU do algoritmo MR. Na implementação dos algoritmos MR/FV, consideramos o operador de previsão dado na Equação (5.5), exato para polinômios de grau menor ou igual a dois.

A compressão, no instante final, é calculada pela fórmula

$$C = 100 - \frac{100 A}{U},$$

em que A é o número de folhas na árvore adaptativa e U a quantidade de folhas na malha uniforme. Em todos os testes, a malha uniforme mais refinada contém  $2^{10} \times 2^{10}$  células e, no parâmetro  $CFL = \frac{\Delta t}{\Delta x}$ ,  $\Delta x$  refere-se a esse nível de resolução.

Calculamos o erro pela norma  $L_1$ 

$$E = ||u(.,T) - u^{MR}(.,T)||_{L_1}$$

O ganho é dado pela razão entre o tempo de CPU necessário para calcular a solução numérica de referência no instante final da simulação  $(CPU_u)$  e o tempo de CPU necessário para calcular a solução adaptativa  $(CPU_{MR})$ . Isto é, o ganho é

$$G = \frac{CPU_u}{CPU_{MR}}.$$

## 9.1 Esquema FV/MR-Roe + ENO + RK2

Nas aplicações abaixo, o esquema FV de referência combina fluxos Roe + ENO com RK2 para integração temporal e consideramos o instante final T = 0, 5 e CFL = 0, 4.

#### 9.1.1 Equação de Advecção

Como primeiro exemplo, consideramos o problema de valor inicial e de contorno

$$u_t + \sum_{i=0}^{d-1} u_{x_i} = 0, \quad (x,t) \in R \times (0,\infty),$$
  
$$u(x,0) = u_0(x),$$
  
(9.1)

em que  $R = [-1, 1]^d$ , com condições de contorno periódicas e a condição inicial

$$u_0(x_0, x_1) = \exp\left(-100(x_0^2 + x_1^2)\right), \quad (x_0, x_1) \in R.$$
(9.2)

Na Figura 9.1, apresentamos as curvas de nível das soluções numéricas obtidas. As curvas de nível da solução sobre as malhas uniforme e MR variam de 0, 1 a 0, 8. Em todos os casos, o intervalo é de 0, 1 entre cada uma das curvas. A solução de referência na malha uniforme encontra-se no canto superior esquerdo, e as soluções adaptativas são mostradas no canto superior direito, com  $\epsilon = 0, 1$ , na linha inferior, à esquerda, com  $\epsilon = 0, 01$  e à direita, com  $\epsilon = 0,001$ . Observamos que as curvas da solução adaptativa se ajustam com aquelas da solução de referência, à medida que  $\epsilon$  diminui. Exibimos na Figura 9.2, na parte superior, as curvas representando o corte da solução na malha uniforme e da solução MR com  $\epsilon = 0,001$ . Na parte inferior, a diferença entre as duas curvas.



Figura 9.1: Equação de advecção 2D: curvas de nível em T = 0, 5 da solução de referência (acima, à esquerda), solução adaptativa com  $\epsilon = 0, 1$  (acima, à direita),  $\epsilon = 0, 01$  (abaixo, à esquerda) e  $\epsilon = 0,001$  (abaixo à direita). Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 4. Curvas de nível variando de 0,1 a 0,9, com espaçamento 0,1.

Na Figura 9.3, apresentamos as malhas adaptativas das soluções representadas pelas



Figura 9.2: Equação de advecção 2D: na parte superior, perfil em  $x_0 = 0, 5$  da solução de referência e da solução adaptativa com  $\epsilon = 0,001$ . Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 4. Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.

curvas de nível da Figura 9.1, à esquerda, com  $\epsilon = 0, 1$ , ao centro com  $\epsilon = 0, 01$  e à direita com  $\epsilon = 0,001$ . Como esperado, as malhas adaptativas ficam mais refinadas à medida que  $\epsilon$  diminui e acompanham a advecção da condição inicial. Devido à condição de periodicidade, quando o pulso se aproxima da fronteira, o efeito é observado pelo refinamento das células próximas aos lados opostos.

A Figura 9.4, na parte superior, mostra a compressão para diferentes valores de  $\epsilon$ . Observamos que, para  $\epsilon = 0, 1$ , a compressão é superior a 99%. À medida que  $\epsilon$  decresce, a malha MR fica mais refinada, com a compressão decrescendo até alcançar 98,8%, quando



Figura 9.3: Equação de advecção 2D: malhas adaptativas em T = 0, 5 obtidas com  $\epsilon = 0, 1$ (à esquerda),  $\epsilon = 0,01$  (ao centro) e  $\epsilon = 0,001$  (à direita). Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 4.

 $\epsilon = 0,001$ . Como conseqüência da diminuição da taxa de compressão, observamos, no gráfico da parte inferior, uma queda no ganho em tempo de CPU. Por exemplo, o ganho é de 8 vezes para  $\epsilon = 0, 1$  e de 1,4 vez para  $\epsilon = 0,001$ . A diferença entre a solução adaptativa e a solução de referência, medida na norma  $L_1$ , mantém-se proporcional ao parâmetro de truncamento, como vemos na Figura 9.4, no centro. A Tabela 9.1 contém os dados relativos à Figura 9.4.

έ	$CPU_{MR}$ (s)	ganho	erro $(L_1)$	$\operatorname{compressão}(\%)$
$1,0*10^{-1}$	75, 2	8,0	$3,38*10^{-3}$	99,8
$1,0*10^{-2}$	225, 4	2, 6	$1,04*10^{-3}$	99,3
$1,0*10^{-3}$	425, 3	1, 4	$3,44*10^{-4}$	$98,\!8$
$1,0*10^{-5}$	796, 7	0,7	$6,08*10^{-5}$	$95,\!9$
$1,0*10^{-5}$	1091, 0	0, 5	$1,89*10^{-5}$	93,7
$1,0*10^{-6}$	1372, 0	0, 4	$1,09*10^{-5}$	91,7

Tabela 9.1: Equação de Advecção 2D: dados das soluções FV e FV/MR em função do nível de resolução. Esquema de Roe + ENO + RK2 com CFL = 0, 4 e T = 0, 5. Na malha uniforme  $CPU_u = 594.6$ .

#### 9.1.2 Equação de Burgers

Consideramos a equação de Burgers

$$u_t + (f(u))_{x_0} + (f(u))_{x_1} = 0, \quad (x_0, x_1) \in R,$$
(9.3)



Figura 9.4: Equação de advecção 2D: compressão, erro e ganho de CPU no método MR, em função do parâmetro de truncamento. Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células, instante final T = 0, 5 e CFL = 0, 4.

em que  $f(u) = \frac{1}{2}u^2$  e  $R = [-1, 1]^2$ , com condição de contorno periódica e condição inicial

$$u_0(x_0, x_1) = \exp\left(-100(x_0^2 + x_1^2)\right), \quad (x_0, x_1) \in R.$$
(9.4)

Na Figura 9.5, apresentamos as curvas de nível das soluções numéricas do problema de Burgers, variando de 0,1 a 0,6, com espaçamento 0,1 entre cada uma das curvas. A solução de referência na malha uniforme encontra-se no canto superior esquerdo, e as soluções adaptativas são mostradas no canto superior direito, com  $\epsilon = 0, 1$ , na linha inferior à esquerda, com  $\epsilon = 0, 01$  e à direita, com  $\epsilon = 0, 001$ . Como no caso do problema de adveção, observamos que as curvas da solução adaptativa do problema de Burgers se ajustam com aquelas da solução de referência, à medida que  $\epsilon$  diminui. Na Figura 9.6, na parte superior, apresentamos as curvas representando o corte da solução na malha uniforme (FV) e da solução MR com  $\epsilon = 0,001$ . A diferença entre as curvas é mostrada na parte inferior.

Na Figura 9.7, apresentamos as malhas adaptativas das soluções representadas pelas curvas de nível da Figura 9.5, à esquerda com  $\epsilon = 0, 1$ , ao centro com  $\epsilon = 0, 01$  e à direita, com  $\epsilon = 0,001$ .

A Figura 9.8, na parte superior, mostra que a compressão varia entre 99,7%, com  $\epsilon = 0, 1, e 96,8\%$ , com  $\epsilon = 0,001$ . A curva da parte inferior mostra como o ganho em tempo de CPU se comporta neste caso. O ganho é de 5,6 vezes para  $\epsilon = 0, 1$  e não há ganho de CPU para  $\epsilon = 0,001$ . Ao centro, vemos que a diferença entre a solução adaptativa e a solução de referência, medida na norma  $L_1$ , também fica proporcional à variação do parâmetro de truncamento. A Tabela 9.2 contém os dados relativos à Figura 9.8.

#### 9.1.3 Equação de Buckley-Leverett

Resolvemos numericamente o problema diferencial que modela a recuperação de óleo em um reservatório petrolífero [4]. Consideramos a equação

$$u_t + f(u)_{x_0} + g(u)_{x_1} = 0, \quad (x_0, x_1) \in R.$$
 (9.5)



Figura 9.5: Equação de Burgers 2D: curvas de nível em T = 0, 5 da solução de referência (acima, à esquerda), solução adaptativa com  $\epsilon = 0, 1$  (acima, à direita),  $\epsilon = 0, 01$  (abaixo, à esquerda) e  $\epsilon = 0,001$  (abaixo à direita). Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 4. Curvas de nível variando de 0,1 a 0,6, com espaçamento 0,1.



Figura 9.6: Equação de Burgers 2D: na parte superior, perfil em  $x_0 = 0$  da solução de referência e da solução adaptativa com  $\epsilon = 0,001$ . Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células, T = 0,5 e CFL = 0,4. Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.



Figura 9.7: Equação de Burgers 2D: malhas adaptativas em T = 0, 5 obtidas com  $\epsilon = 0, 1$ (à esquerda),  $\epsilon = 0,01$  (ao centro) e  $\epsilon = 0,001$  (à direita).Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 4.



Figura 9.8: Equação de Burgers 2D: compressão, erro e ganho de CPU no método MR, em função do parâmetro de truncamento. Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células, instante final T = 0, 5 e CFL = 0, 4.

$\epsilon$	$CPU_{MR}$ (s)	ganho	erro $(L_1)$	$\operatorname{compressão}(\%)$
$1,0*10^{-1}$	108, 8	5, 6	$1,08*10^{-3}$	99,7
$1,0*10^{-2}$	291, 5	2, 1	$3,89*10^{-4}$	99,0
$1,0*10^{-3}$	663, 9	0,9	$8,49*10^{-5}$	96,8
$1,0*10^{-5}$	962, 8	0, 6	$2,99 * 10^{-5}$	$94,\!3$
$1,0*10^{-5}$	1355, 5	0, 4	$1,24*10^{-5}$	92,2
$1,0*10^{-6}$	1658, 2	0,3	$2,37*10^{-6}$	90,4

Tabela 9.2: Equação de Burgers 2D: dados das soluções FV e FV/MR em função do nível de resolução. Esquema de Roe + ENO + RK2 com CFL = 0, 4 e T = 0, 5. Na malha uniforme,  $CPU_u = 617, 4$ .

em que  $R \in [-1, 1]^2$ , a condição de contorno é periódica e a condição inicial é

$$u_0(x,y) = \exp(-4\sin^2(2x_0) - 4\sin^2(2x_1)).$$
(9.6)

Os fluxos são dados por

$$f(u) = \frac{u^2}{u^2 + (1-u)^2},$$
  

$$g(u) = -u(1-u).$$

Na Figura 9.9, apresentamos as curvas de nível das soluções numéricas do problema de Buckley-Leverett, variando de 0,02 a 0,74, com espaçamento 0,02. Como nos casos anteriores, observamos que as curvas da solução adaptativa do problema de Buckley-Leverett se ajustam com aquelas da solução de referência, à medida que  $\epsilon$  diminui.

Na Figura 9.10, apresentamos as malhas adaptativas das soluções representadas pelas curvas de nível da Figura 9.9, para  $\epsilon = 0, 1$  e  $\epsilon = 0, 01$ . Na Figura 9.22, à esquerda, encontra-se a malha adaptativa da solução com  $\epsilon = 0, 1$  no instante T = 0, 5 e CFL = 0, 4. À direita, malha adaptativa para  $\epsilon = 0, 01$ . A malha com  $\epsilon = 0, 001$  é demasiadamente cheia para mostrarmos.

Na Figura 9.11, na parte superior, exibimos o corte da solução da equação de Buckley-Leverett em  $x_0 = 0$ . Consideramos a solução FV e MR com  $\epsilon = 0,001$ . Na parte inferior, é mostrada a diferença das duas curvas.

Na Figura 9.12, na parte superior, vemos que a compressão varia entre 99% (malha com  $\epsilon = 0, 1$ ) e 5%( $\epsilon = 0,000001$ ). Na malha com  $\epsilon = 0,001$ , a compressão é de 89%. Ao



Figura 9.9: Equação de Buckley-Leverett 2D: curvas de nível da solução de referência(sup. esq.), solução adaptativa com  $\epsilon = 0, 1(\text{sup. dir.}), \epsilon = 0, 01(\text{inf. esq.})$  e  $\epsilon = 0, 001(\text{inf. dir.})$ . Malha uniforme com  $2^{10} \times 2^{10}$  células. Curvas de nível variando de 0,02 a 0,74, com espaçamento 0,02.


Figura 9.10: Equação de Buckley-Leverett 2D: malhas adaptativas com parâmetro de truncamento  $\epsilon = 0, 1$  (esquerda) e  $\epsilon = 0, 01$  (direita). Instante final T = 0, 5 e CFL = 0, 4.

centro vemos que a diferença entre a solução adaptativa e a solução de referência, medida na norma  $L_1$ , é proporcional à variação do parâmetro de truncamento. Na parte inferior, vemos que o ganho em tempo de CPU é de 2,2 vezes para  $\epsilon = 0, 1$  e não há ganho para  $\epsilon = 0,01$ . A Tabela 9.3 contém os dados relativos à Figura 9.12.

$\epsilon$	$CPU_{MR}$ (s)	ganho	erro $(L_1)$	$\operatorname{compressão}(\%)$
$1,0*10^{-1}$	298, 2	2,20	$5,14*10^{-3}$	99,3
$1,0*10^{-2}$	997, 3	0, 60	$1,53*10^{-3}$	97,5
$1,0*10^{-3}$	3236, 6	0, 20	$5,72*10^{-4}$	$89,\! 6$
$1,0*10^{-5}$	7038, 5	0, 10	$9,98*10^{-5}$	$65,\!9$
$1,0*10^{-5}$	10391, 7	0,06	$4,47*10^{-5}$	30,7
$1,0*10^{-6}$	12040, 0	0,05	$3,36*10^{-6}$	$_{6,0}$

Tabela 9.3: Equação de Buckley-Leverett 2D: dados das soluções FV e FV/MR em função do nível de resolução. Esquema de Roe + ENO + RK2 com CFL = 0, 4 e T = 0, 5. Na malha uniforme,  $CPU_u = 669, 6$ .



Figura 9.11: Equação de Buckley-Leverett 2D: na parte superior, perfil em  $x_0 = 0,0$  da solução de referência e solução adaptativa com  $\epsilon = 0,001$ . Esquema de referência usando esquema Roe + ENO + RK2, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0,4. Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.



Figura 9.12: Equação de Buckley-Leverett 2D: compressão, erro e ganho em função do parâmetro de truncamento usando esquema de Roe + ENO + RK2. Malha uniforme com  $2^{10} \times 2^{10}$  células e  $\epsilon = 0, 1$ , inicialmente. Instante final T = 0, 5 e CFL = 0, 4.

### 9.2 Esquema FV/MR-WENO+RK3

Nas próximas aplicações, o esquema FV de referência combina fluxos WENO com RK3 para integração temporal, usando CFL = 0, 5. O objetivo é avaliar o efeito da adaptatividade com um esquema mais complexo, exigindo maior esforço computacional. Para comparação com o esquema Roe + ENO + K2, repetimos os exemplos analisados na seção anterior.

#### 9.2.1 Equação de Advecção

Consideramos o problema de valor inicial e de contorno, dado pela Equação (9.1), em que  $R = [-1, 1]^2$ , com condições de contorno periódicas. A condição inicial considerada é dada pela Gaussiana, descrita na Equação (9.2).

Na Figura 9.13, apresentamos as curvas de nível das solucões numéricas obtidas. A solução de referência na malha uniforme encontra-se no canto superior esquerdo, e as soluções adaptativas são mostradas no canto superior direito, com  $\epsilon = 0, 1$ , na linha inferior, à esquerda, com  $\epsilon = 0, 01$  e à direita, com  $\epsilon = 0, 001$ . Observamos que as curvas da solução adaptativa se ajustam com aquelas da solução de referência à medida que  $\epsilon$  diminui. As curvas de nível da solução sobre as malhas uniforme e MR variam de 0, 1 a 0, 8. Em todos os casos, o intervalo é de 0, 1 entre cada uma das curvas. Exibimos, na Figura 9.14, na parte superior, as curvas representando o corte da solução na malha uniforme e da solução MR com  $\epsilon = 0,001$ . Vemos, na parte inferior, a diferença entre as duas curvas.

Na Figura 9.15, apresentamos as malhas adaptativas das soluções representadas pelas curvas de nível da Figura 9.13, à esquerda com  $\epsilon = 0, 1$ , ao centro com  $\epsilon = 0, 01$  e à direita com  $\epsilon = 0, 001$ . Como esperado, as malhas adaptativas ficam mais refinadas à medida que  $\epsilon$  diminui e acompanham a advecção da condição inicial.

A Figura 9.16 mostra a compressão, erro e ganho nas simulações do esquema MR, com comportamento similar ao caso do fluxo de Roe + ENO + RK2. Como esperado, o ganho de CPU é superior quando se usa o esquema de referência com fluxo WENO + RK3, que



Figura 9.13: Equação de advecção 2D: curvas de nível em T = 0, 5 da solução de referência (acima, à esquerda), solução adaptativa com  $\epsilon = 0, 1$  (acima, à direita),  $\epsilon = 0, 01$  (abaixo, à esquerda) e  $\epsilon = 0, 001$  (abaixo à direita). Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 5. Curvas de nível variando de 0,1 a 0,9, com espaçamento 0,1.



Figura 9.14: Equação de advecção 2D: na parte superior, perfil em  $x_0 = 0, 5$  da solução de referência e da solução adaptativa com  $\epsilon = 0,001$ . Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células T = 0, 5 e CFL = 0, 5. Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.



Figura 9.15: Equação de advecção 2D: malhas adaptativas em T = 0, 5 obtidas com  $\epsilon = 0, 1$  (à esquerda),  $\epsilon = 0, 01$  (ao centro) e  $\epsilon = 0, 001$  (à direita). Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 5. Curvas de nível a partir de 0, 1 com intervalo entre elas de 0, 1.

é computacionalmente mais caro. A Tabela 9.4 contém os dados relativos à Figura 9.16.

$\epsilon$	$CPU_{MR}$ (s)	ganho	erro $(L_1)$	$\operatorname{compressão}(\%)$
$1,0*10^{-1}$	91, 0	31, 8	$3,64*10^{-3}$	$99,\!8$
$1,0*10^{-2}$	294, 7	9,8	$1,06*10^{-3}$	99,2
$1,0*10^{-3}$	583, 6	4, 9	$3,01*10^{-4}$	98,4
$1,0*10^{-5}$	972, 2	2,9	$1,28*10^{-4}$	96,5
$1,0*10^{-5}$	1399, 0	2, 0	$3,34*10^{-5}$	94,0
$1,0*10^{-6}$	1770, 6	1, 6	$9,69*10^{-6}$	$91,\!6$

Tabela 9.4: Equação de Advecção 2D: dados das soluções FV e FV/MR em função do nível de resolução. Esquema de WENO + RK3 com CFL = 0, 5 e T = 0, 5. Na malha uniforme,  $CPU_u = 2895, 76$ .

#### 9.2.2 Equação de Burgers

Consideramos a equação de Burgers (9.3) com condição de contorno periódica e condição inicial (9.4).

Na Figura 9.17, apresentamos as curvas de nível das soluções numéricas para este problema. As curvas de nível da solução sobre as malhas uniforme e MR variam de 0, 1 a 0, 6. Em todos os casos, o intervalo é de 0, 1 entre cada uma das curvas. A solução de referência na malha uniforme encontra-se no canto superior esquerdo, e as soluções adaptativas são mostradas no canto superior direito, com  $\epsilon = 0, 1$ , na linha inferior à



Figura 9.16: Equação de advecção 2D: compressão, erro e ganho de CPU no método MR, em função do parâmetro de truncamento. Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células, instante final T = 0, 5 e CFL = 0, 5.

esquerda, com  $\epsilon = 0,01$  e à direita, com  $\epsilon = 0,001$ . Como no caso do problema de advecção, observamos que as curvas da solução adaptativa do problema de Burgers se ajustam com aquelas da solução de referência, à medida que  $\epsilon$  diminui. Exibimos, na Figura 9.18, na parte superior, as curvas representando o corte da solução na malha uniforme e da solução MR com  $\epsilon = 0,001$ . Na parte inferior, plotamos a diferença entre as curvas.



Figura 9.17: Equação de Burgers 2D: curvas de nível em T = 0, 5 da solução de referência (acima, à esquerda), solução adaptativa com  $\epsilon = 0, 1$  (acima, à direita),  $\epsilon = 0, 01$  (abaixo, à esquerda) e  $\epsilon = 0, 001$  (abaixo à direita). Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 5. Curvas de nível variando de 0,1 a 0,6, com espaçamento 0,1.



Figura 9.18: Equação de Burgers 2D: na parte superior, perfil em  $x_0 = 0$  da solução de referência e da solução adaptativa com  $\epsilon = 0,001$ . Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células e T = 0,5 CFL = 0,5. Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.

Na Figura 9.19, apresentamos as malhas adaptativas das soluções representadas pelas curvas de nível da Figura 9.17, à esquerda com  $\epsilon = 0, 1$ , ao centro com  $\epsilon = 0, 01$  e à direita, com  $\epsilon = 0, 001$ .



Figura 9.19: Equação de Burgers 2D: malhas adaptativas em T = 0, 5 obtidas com  $\epsilon = 0, 1$ (à esquerda),  $\epsilon = 0, 01$  (ao centro) e  $\epsilon = 0, 001$  (à direita). Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células e CFL = 0, 5 em T = 0, 5.

A Figura 9.20 apresenta as curvas referentes à compressão, ao erro e ao ganho de CPU. Neste caso, com o fluxo numérico WENO + RK3, computacionalmente mais caro que o fluxo Roe + ENO + RK2, considerado na seção anterior, o ganho em tempo de CPU prevaleceu em todos os cenários considerados, enquanto que a compressão e o erro se mantiveram da mesma ordem. A Tabela 9.5 contém os dados relativos à Figura 9.20.

$\epsilon$	$CPU_{MR}$ (s)	ganho	erro $(L_1)$	$\operatorname{compressão}(\%)$
$1,0*10^{-1}$	131, 7	25, 9	$1,38*10^{-3}$	99,7
$1,0*10^{-2}$	371, 3	9, 2	$3,52*10^{-4}$	99,0
$1,0*10^{-3}$	792, 9	4, 3	$6,98*10^{-5}$	96,8
$1,0*10^{-5}$	1205, 2	2, 8	$2,81 * 10^{-5}$	94,3
$1,0*10^{-5}$	1569, 7	2, 1	$6,37*10^{-6}$	92,2
$1,0*10^{-6}$	1919,9	1,7	$1,97*10^{-6}$	90,4

Tabela 9.5: Equação de Burgers 2D: dados das soluções FV e FV/MR em função do nível de resolução. Esquema de WENO + RK3 com CFL = 0, 5 e T = 0, 5. Na malha uniforme,  $CPU_u = 3418, 9$ .



Figura 9.20: Equação de Burgers 2D: compressão, erro e ganho de CPU no método MR, em função do parâmetro de truncamento. Esquema de referência usando esquema WENO + RK3, malha uniforme com  $2^{10} \times 2^{10}$  células, instante final T = 0, 5 e CFL = 0, 5.

#### 9.2.3 Equação de Buckley-Leverett

Consideramos a equação (9.5) em que a condição de contorno é periódica e a condição inicial é (9.6).

Na Figura 9.21, apresentamos as curvas de nível das soluções numéricas deste problema. As curvas de nível da solução sobre as malhas uniforme e MR variam de 0,02 a 0,8. Em todos os casos, o intervalo é de 0,02 entre cada uma das curvas. No canto superior esquerdo, a solução de referência na malha uniforme. No canto superior direito, a solução com  $\epsilon = 0, 1$ . Na linha inferior, à esquerda, solução com  $\epsilon = 0,01$  e à direita solução com  $\epsilon = 0,001$ .

Na Figura 9.22, apresentamos as malhas adaptativas das soluções representadas pelas curvas de nível da Figura 9.21, para  $\epsilon = 0, 1$  e  $\epsilon = 0, 01$ , respectivamente. Apesar da alta compressão, a malha com  $\epsilon = 0,001$  é demasiadamente cheia para mostrarmos.

Na Figura 9.23, na parte superior, exibimos o corte da solução da equação de Buckley-Leverett em  $x_0 = 0$ . Na parte inferior, plotamos a diferença entre as curvas.

Na Figura 9.24, plotamos os gráficos da compressão, do erro e do ganho de CPU. Neste caso, o erro e a compressão se mantiveram como no caso do fluxo Roe + ENO + RK2. O efeito de se usar um esquema com fluxo numérico computacionalmente mais caro melhora o ganho de CPU somente para altas taxas de compressão.

А	Tabela	9.6	contém	$\mathbf{OS}$	dados	rel	lativos	à	Figura	9.24	•
---	--------	-----	--------	---------------	-------	-----	---------	---	--------	------	---

$\epsilon$	$CPU_{MR}$ (s)	ganho	erro $(L_1)$	$\operatorname{compressão}(\%)$
$1,0*10^{-1}$	382, 1	11, 4	$7,07*10^{-3}$	99,2
$1,0*10^{-2}$	1342, 9	3, 2	$2,08*10^{-3}$	97,5
$1,0*10^{-3}$	4748, 6	0,9	$8,47*10^{-4}$	88,8
$1,0*10^{-5}$	9889, 6	0, 4	$1,46*10^{-4}$	$67,\! 6$
$1,0*10^{-5}$	14552, 2	0,3	$1,84*10^{-5}$	$_{30,3}$
$1,0*10^{-6}$	16890, 6	0, 2	$8,00*10^{-6}$	$5,\!6$

Tabela 9.6: Equação de Buckley-Leverett 2D: dados das soluções FV e FV/MR em função do nível de resolução. Esquema de WENO + RK3 com CFL = 0, 5 e T = 0, 5. Na malha uniforme,  $CPU_u = 4371, 7$ .



Figura 9.21: Equação de Buckley-Leverett 2D: curvas de nível da solução de referência, solução adaptativa com  $\epsilon = 0, 1(\text{sup. dir.}), \epsilon = 0, 01(\text{inf. esq.}) e \epsilon = 0, 001(\text{inf. dir.}).$ Malha uniforme com  $2^{10} \times 2^{10}$  células e esquema WENO + RK3. Consideramos CFL = 0, 5 e T = 0, 5. Curvas de nível variando de 0, 02 a 0, 74, com espaçamento 0, 02.



Figura 9.22: Equação de Buckley-Leverett 2D: malhas adaptativas com parâmetro de truncamento  $\epsilon = 0, 1$ (esquerda) e  $\epsilon = 0, 01$ (direita). Esquema WENO + RK3, instante final T = 0, 5 e CFL = 0, 5.



Figura 9.23: Equação de Buckley-Leverett 2D: na parte superior, perfil das soluções em  $x_0 = 0$ . Solução de referência e solução adaptativa com  $\epsilon = 0,001$ . Malha uniforme com  $2^{10} \times 2^{10}$  células. Esquema de WENO + RK3 com instante final T = 0,5 e CFL = 0,5. Na parte inferior, gráfico do erro entre solução adaptativa e solução exata.



Figura 9.24: Equação de Buckley-Leverett 2D: compressão, erro e ganho em função do parâmetro de truncamento usando esquema de WENO + RK3. Malha uniforme com  $2^{10} \times 2^{10}$  células e  $\epsilon = 0, 1$ , inicialmente. Instante final T = 0, 5 e CFL = 0, 5.

# Capítulo 10

## Conclusões e Trabalhos Futuros

A proposta desta tese é a de apresentar novas metodologias de aproximação em multiescala, baseadas em informações mapeadas em malhas diádicas, como uma alternativa ao uso das tradicionais malhas  $2^d$ -ádicas, para representações de funções d-dimensionais e na solução numérica de problemas de valor inicial e de contorno para equações diferenciais. Ao mesmo tempo em que propomos novas alternativas para essas representações, buscamos obter melhorias na eficiência computacional dos métodos, tanto na compressão de dados como em ganhos de tempo de CPU.

A representação de funções d-dimensionais usando esquemas baseados em malhas diádicas oferece algumas vantagens sobre representações similares, baseadas em malhas  $2^{d}$ -ádicas, veja Castro *et al.* [7]. Por exemplo, as malhas diádicas são mais simples de codificar pois, enquanto em uma malha  $2^{d}$ -ádica os nós não folha possuem  $2^{d}$  filhos, em uma malha diádica os nós possuem apenas dois filhos, independente da dimensão do domínio computacional. Além disso, o refinamento mais gradual das malhas diádicas permite antecipar a conclusão do processo, principalmente para funções com um padrão de singularidades alinhadas com os eixos coordenados.

Nesse sentido, propomos novos esquemas MR, de ordem arbitrária, para médias celulares em malhas diádicas *d*-dimensionais. Um aspecto interessante desses esquemas, que facilita sua implementação, é que os operadores de previsão são definidos em termos de fórmulas unidimensionais, aplicadas na direção em que o refinamento está sendo efetuado. Motivados pelo desempenho dos esquemas MR em malhas diádicas, também exploramos as vantagens de se usar essas técnicas para derivar esquemas adaptativos de volumes finitos para problemas de leis de conservação, como alternativa aos conhecidos esquemas FV/MR para malhas  $2^d$ -ádicas.

Um benefício a ser destacado é a facilidade com que a implementação dos esquemas FV/MR para malhas diádicas é generalizada para dimensões superiores, de forma quase imediata, visto que não há necessidade de escrevermos um código para cada dimensão considerada. A forma como os algoritmos estão implementados também possibilita o uso de esquemas para cálculo do fluxo com ordem superior.

Quando o problema analisado possui estrutura localizada, o método FV/MR trabalha com uma alta compressão de dados e mantém a diferença entre as soluções FV/MR e FV sob controle, na ordem do parâmetro de truncamento. Com representações em malhas adaptativas, que usam uma quantidade menor de células folha para representar a solução numérica, procura-se reduzir o volume de cálculo de fluxos numéricos para as integrações no tempo. Nos experimentos realizados, mostramos ser possível obter significante ganho de tempo de CPU e de memória utilizados no algoritmo FV/MR em malhas diádicas, quando comparados com o esquema FV de referência baseado na malha uniforme do nível mais refinado. Além disso, ao usarmos o fluxo numérico WENO+RK3, computacionalmente mais caro que o fluxo Roe+ENO+RK2, o ganho em tempo em CPU aumenta.

Em resumo, as principais contribuições desta tese são:

- Definição de esquemas MR e ordem arbitrária para médias celulares em malhas diádicas d-dimensionais.
- Formulação de análise de multirresolução de espaços funcionais correspondentes.
- Implementação dos esquemas MR para médias celulares utilizando uma estrutura de dados de árvore binária.
- Implementação de esquemas adaptativos FV/MR, combinando os esquemas de MR

para malhas diádicas com o método de volumes finitos, para simulação de leis de conservação escalares, permitindo a implementação de fluxos numéricos de alta ordem.

 Aplicação do método FV/MR na simulação de problemas para as equações de advecção, Burges e Buckley-Leverett, em 1D e 2D, com análise de seu desempenho em termos dos diferentes parâmetros de truncamento dos coeficientes wavelet e para dois esquemas FV de referência.

Os resultados foram publicados em congressos internacionais [7, 8, 9] e em revista especializada [10].

Em decorrência dos estudos feitos nesta tese, visualizamos algumas melhorias que podem ser feitas em trabalhos futuros. Dentre elas, consideramos as seguintes metas.

- Aplicações dos esquemas em problemas 3D;
- Implementação de outros tipos de condições de contorno ( não somente a condição periódica);
- Implementação do método FV/MR para sistemas de equações diferenciais, usando estrutura de árvore binária;
- Implementação do esquema FV/MR com previsão exata para polinômios de grau  $\leq 4$  e fluxos WENO com r = 3.
- Aplicações de esquemas multinível MR e de splines diádicos em computação gráfica;
- Comparação com outras formas de estruturas de dados, como quad-trees, hash tables.

### **Referências Bibliográficas**

- B. L. Bihari. Multiresolution schemes for sonservations laws with viscosity. J. Comput. Phys., 123:207–225, 1996.
- B. L. Bihari and A. Harten. Multiresolution schemes for the numerical solution of 2-d conservation laws. SIAM J. Sci. Comput., 18(2):315–354, 1997.
- K. Brix, S. Melian, S. Müller, and M. Bachmann. Adaptive multiresolution methods. ESAIM: Proceedings, 34:151–183, December 2011.
- [4] S. E. Buckley and M. C. Leverett. Mechanism of fluid displacements in sands. Transactions of the AIME, (146):107–116, 1942.
- [5] C. G. S. Cardoso. Grades diádicas adaptativas para simulação de escoamento de petróleo. Master's thesis, IC – Unicamp, Campinas, São Paulo, Dezembro 2004.
- [6] C. G. S. Cardoso, M. C. Cunha, A. Gomide, D. J. Schiozer, and J. Stolfi. Finite elements on dyadic grids for oil reservoir simulation. *Math. Comput. Simulat.*, 73(1):87–104, 2006.
- [7] D. A. Castro, S. M. Gomes, A. Gomide, A. Oliveira, and J. Stolfi. Multiresolution analysis on multidimensional dyadic grids. *Proceedings of SAMPTA* '09, 2010.
- [8] D. A. Castro, S. M. Gomes, and J. Stolfi. Multiresolution scheme for numerical solution of hyperbolic conservation laws on dyadic grids. *Proceedings of ICCAM* 2010, 2010.

- [9] D. A. Castro, S. M. Gomes, and J. Stolfi. A multiresolution analysis on dyadic grids with applications to adaptive finite volume schemes. *Proceedings of ICIAM 2011*, 2011.
- [10] D. A. Castro, S. M. Gomes, and J. Stolfi. An adaptive multiresolution method on dyadic grids: Application to transport equations. J. Comput. Appl Math., doi:10.1016/j.cam.2011.05.044.
- [11] G. Chiavassa and R. Donat. Point value multi-scale algorithms for 2d compressible flow. SIAM J. Sci. Comput., 23(3):805–823, 2001.
- [12] A. Cohen. Wavelet nethods in numerical analysis: in handbook of numerical analysis, volume VII. Ph. Ciarlet and J. L. Lions eds., 1998.
- [13] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. Pur. Appl. Math.*, 45(5):485–560, 1992.
- [14] A. Cohen, S. M. Kaber, S. Müller, and M. Postel. Fully adaptive multirresolution finite volume schemes for conservation laws. *Math. Comp.*, 72:183–225, 2003.
- [15] A. Cohen and M. Postel S. M. Kaber, S. Müller. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math Comp*, 72(241):183–225, 2003.
- [16] I. Daubechies. Ten lectures on wavelets. SIAM, Philadelphia, PA, 1992.
- [17] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. J. Comput. Phys., 227(8):3758–3780, 2008.
- [18] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. Space-time adaptive multiresolution methods for hyperbolic conservation laws: Applications to compressible euler equations. *Appl. Numer. Math.*, 59(9):2303–2321, 2009.
- [19] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. Adaptive multiresolution methods. *ESAIM: Proceedings*, 34:1–96, December 2011.

- [20] R. Finkel and J.L. Bentley. Quad trees: a data structure for retrieval on composite keys. Acta Inform., 4(1):1–9, 1974.
- [21] A. Haar. On the theory of orthogonal function systems. J. Comput. Phys., 188(2):493–523, 2003.
- [22] G. E. Hammerlin and K. H. Hoffmann. Numerical mathematics. Springer-Verlag, New York, 1991.
- [23] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. Commun. Pur. Appl. Math, 48:1305–1342, 1995.
- [24] A. Harten. Multiresolution representation of data: a general framework. SIAM J. Numer. Anal., 33(3):1205–1256, 1996.
- [25] A. Harten. Multiresolution representation of cell-averaged data. Signal and Image Rep. in Combined Spaces, Y.Y. Zeevi and R.R. Coifman(Eds.), pages 361–391, 1998.
- [26] A. Harten. Multiresolution representation of cell-averaged data. Technical Report CAM/Report/94-21, UCLA, Los Angeles, US, July 1994.
- [27] A. Harten and S. Osher. Uniformly high-order accurate nonoscillatory schemes. SIAM J. Numer. Anal., 24(2):279–309, 1987.
- [28] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. J. Comput. Phys., 126:202–228, 1996.
- [29] M. Kaibara and S. M. Gomes. A fully adaptive multiresolution scheme for shock computations. In E. F. Toro, editor, Godunov Methods: Theory and Applications, Klumer Academic/Plenum Publishers, Princeton, 2003.
- [30] M. K. Kaibara. Análise de multi-resolução para leis de conservação em malhas adaptativas. PhD em Matemática Aplicada, IMECC – Unicamp, Campinas, São Paulo, Maio 2000.

- [31] D. Knuth. The art of computer programming vol 1. Fundamental algorithms. Addison-Wesley, 3rd edition, 1997.
- [32] Guillaume Latu. Sparse data structure design for wavelet-based methods. ESAIM: Proceedings, 34:240–276, December 2011.
- [33] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. J. Comput. Phys., 115:200–212, 1994.
- [34] S. Müller. Adaptive multiscale schemes for conservation laws, volume 27 of Lectures notes in computational science and engineering. Springer, Heidelberg, 2003.
- [35] S. Müller and Y. Stiriba. Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. J. Sci. Comput., 30(3):493–531, 2007.
- [36] P. M. Prenter. Splines and variational methods. John Wiley and Sons, New York, 1975.
- [37] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes.
  J. Comput. Phys., 43:357–372, 1981.
- [38] O. Roussel and K. Schneider. An adaptive multiresolution method for combustion problems: application to flame ball-vortex interaction. *Comp. Fluids*, 34(7):817–831, 2005.
- [39] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. J. Comput. Phys., 188(2):493–523, 2003.
- [40] K. Schneider and O. Vasilyev. Wavelet methods in computational fluid dynamics. Annu. Rev. Fluid Mech., 42:473–503, 2010.
- [41] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schames. J. Comput. Phys., 77:439–471, 1988.