Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Igor Morgado

Métodos miméticos aplicados em equações diferenciais parciais

Rio de Janeiro
2019

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC-A

M847 Morgado, Igor.
Métodos miméticos aplicados em equações diferenciais parciais / Igor Morgado. – 2019.
92f.: il.

Orientadora: Cristiane de Oliveira Faria.
Dissertação (Mestrado em Ciências Computacionais) - Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística.

1. Equações diferenciais parciais - Teses. 2. Diferenças finitas - Teses. I. Faria, Cristiane de Oliveira. II. Universidade do Estado do Rio de Janeiro. Instituto de Matemática e Estatística. III. Título.

CDU 517.95

Rosalina Barros *CRB-7 / 4204* - Bibliotecária responsável pela elaboração da ficha catalográfica.

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

_____ _____

Assinatura Data

Igor Morgado

Métodos miméticos aplicados em equações diferenciais parciais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro. Área de concentração: Ciências exatas e da terra.

Orientador: Prof.ª Dra. Cristiane de Oliveira Faria

Rio de Janeiro

2019

Igor Morgado

Métodos miméticos aplicados em equações diferenciais parciais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro. Área de concentração: Ciências exatas e da terra.

Aprovada em 22 de fevereiro de 2019.

Banca Examinadora:

Prof.ª Dra. Cristiane de Oliveira Faria (Orientador)
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Eduardo Gomes Dutra do Carmo
Universidade Federal do Rio de Janeiro

Prof. Dr. Abimael Fernando Dourado Loula
Laboratório Nacional de Computação Científica

Prof. Dr. Leandro Di Bartolo
Observatório Nacional

Prof. Dr. Carlos Frederico Fragoso de Barros e Vasconcellos
Instituto de Matemática e Estatística - UERJ

Rio de Janeiro

2019

AGRADECIMENTOS

A todos os professores do CComp-IME que direta ou indiretamente me ajudaram nesta jornada,seja nas aulas, seja em conversas ou por que me passaram a paixão que sentem pela Matemática:

Ao Prof. Carlos Frederico (aka Fred) pela sua apaixonante aula sobre Cálculo Variacional;

Ao Prof. Fabiano Oliveira por me fazer olhar novamente para algoritmos e estrutura de dados com o devido respeito;

Ao Prof. Guilherme Mota pelas aulas profundas e conversas sobre visão computacional e suas aplicações;

Ao Prof. Leandro Marzulo (que o Google o tenha) e a Prof.ª Maria Clícia pelas aulas de programação paralela e grafos acíclicos dirigidos;

À Prof.ª Roseli Wedemann por expor de forma matematicamente correta o funcionamento das redes neuronais o que me abriu os olhos para diversas possibilidades,

À Prof.ª Zochil Arenas pela aula de Métodos Numéricos e a possibilidade de expor novas ideias durante as aulas;

E a minha professora e orientadora Cristiane Faria, pelo tempo dispendido durante as semanas, as aulas e tudo mais.

Externamente ao CComp gostaria de agradecer ao Prof. Leandro Di Bartolo, professor do Observatório Nacional por responder minhas questões sobre malhas escalonadas quando eu já estava perdendo todas as esperanças.

Aos colegas do programa em especial ao Artur Grover (aka Arthur) pelos estudos em conjunto, idéias e ler os primeiros manuscritos desta dissertação.

Agradeço também as pessoas que convivem comigo diariamente (também conhecido como família). Ericka Hochleitner por aturar minha dedicação quase religiosa aos estudos e minha filha Juliana Pantoja/Pereira.

Don't panic.

— Douglas Adams, The Hitch hiker's Guide to the Galaxy

RESUMO

MORGADO, I. *Métodos miméticos aplicados em equações diferenciais parciais*. 2019. 92 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

Neste trabalho será estudado o Método Mimético de Castillo-Grone aplicado à um problema hiperbólico, unidimensional sobre malha escalonada homogênea. Este método baseia-se no Método de Diferenças Finitas na elaboração da malha discreta, mas diferencia-se na elaboração do operador diferencial, que, em particular, é formulado para que as características da mecânica do contínuo sejam preservadas. Esta formulação permite que este operador seja aplicado a problemas onde o Método de Diferenças Finitas não obtém convergência satisfatória e nem permite um tratamento das condições de fronteira efetivo. Para validação será modelado o problema da corda elástica com extremidades fixas atendendo as condições de contorno de Dirichlet e Neumann. Será estudado também o comportamento da alteração da resolução da malha temporal e espacial utilizando um esquema de diferenças espacialmente centralizado e progressivo no tempo (FCTS). Dadas as características oscilatórias, para malha temporal utilizamos um modelo do tipo Leapfrog. Um estudo comparativo com o Método de Diferenças Finitas e a solução analítica será apresentado. O MMCG demonstrou convergência levemente superior ao MDF na maior parte dos casos. Além disso apresentamos o formulação computacional para o problema, algoritmos e um código fonte para ser usado como referência.

Palavras-chave: Métodos Miméticos. Equações diferenciais parciais. Métodos Numéricos. Equação da Onda.

ABSTRACT

MORGADO, I. *Mimetic methods applied to partial differential equations*. 2019. 92 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

In this work we will present the Castillo-Grone mimetic method that will be applied to a hyperbolic problem over an uni dimensional, homogeneous and staggered grid. This method is based upon the Finite Differences Method in grid discretization, but stand apart in how the differential operator is built, where the continuum mechanics is preserved. This method is applicable in situations where the standard Finite Difference Method cannot be applied satisfactorily and neither allows a correct handling of boundary conditions. To validate this method we will use an elastic string model with fixed boundaries satisfying the Dirichlet and Neumann boundary conditions. It will be also studied the model behavior while changing the grid resolution in time and space. For the time-space discretization it will be used a Forward-Time Central-Space (FTCS) model. Given the oscillatory behavior, it will be used a Leapfrog model in the time discretization. An analysis will be done, comparing the Castillo-Grone Mimetic Method, the Finite Difference and the analytical solution. In this analysis the Castillo-Grone method has shown slightly better results in most of the analyzed cases. Besides that we also present a computational formulation with algorithms and a source code to be used as reference.

Keywords: Mimethic methods. Partial differential equation. Numerical methods. Wave equation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Função seno discretizada 18
Figura 2 - Malha intercalada uniforme. 19
Figura 3 - Comparação entre os métodos de integração numérica na malha intercalada 24
Figura 4 - Elementos não zero dos operadores 54
Figura 5 - Método Leapfrog para as 5 primeiras iterações 55
Figura 6 - Comparativo da onda em $t = 1,26s$. 59
Figura 7 - Erro de truncamento em $t = 1,26$. 59
Figura 8 - Comparativo da onda em $t = 2,10s$. 60
Figura 9 - Erro de truncamento em $t = 2,10$. 60
Figura 10 - Comparativo da onda em $t = 3,00s$. 61
Figura 11 - Erro de truncamento em $t = 3,00$. 61
Figura 12 - Comparativo da onda em $t = 3,20s$. 62
Figura 13 - Erro de truncamento em $t = 3,20$. 62
Figura 14 - Visualisação de erro e estabilidade. 64
Figura 15 - Análise de convergência a priori dos modelos numéricos. 65

LISTA DE TABELAS

Tabela 1 - Erro máximo. 63
Tabela 2 - Coeficientes de convergência. 66

LISTA DE ALGORITMOS

Algoritmo 1 - Calcula os pontos vizinhos na malha intercalada. 40
Algoritmo 2 - Computa o estêncil utilizando as matrizes de Vandermonde. 41
Algoritmo 3 - Cálculo da matriz para estêncil. 42
Algoritmo 4 - Calcula vetor gerador do operador Divergente no contorno. 43
Algoritmo 5 - Calcula vetor gerador do operador Gradiente no contorno. 44
Algoritmo 6 - Calcula o estêncil na borda do operador. 45
Algoritmo 7 - Construção da matriz $\tilde{\Pi}$. 47
Algoritmo 8 - Computa vetor mimético da submatriz de contorno. 48
Algoritmo 9 - Calcula a submatriz $A(\alpha)$. 49
Algoritmo 10 - Calcula o Operador mimético de Castillo-Grone. 50
Algoritmo 11 - Computa o erro máximo. 63

LISTA DE ABREVIATURAS E SIGLAS

MDF	Método das Diferenças Finitas
MVF	Método dos Volumes Finitos
MEF	Método dos Elementos Finitos
MMCG	Método Mimético Castillo-Grone
EDP	Equações Diferenciais Parciais
EDO	Equações Diferenciais Ordinárias

LISTA DE SÍMBOLOS

$\mathbf{D}$	Operador mimético do divergente
$\mathbf{G}$	Operador mimético do gradiente
∇	Operador gradiente contínuo
Δ	Operador Laplaciano contínuo
∂	Derivada parcial
$\mathbf{1}_N$	Vetor representando uma função constante N dimensional
$\mathbb{1}_N$	Matriz diagonal com elementos unitários de ordem N.

SUMÁRIO

	INTRODUÇÃO	14
1	**OPERADORES MIMÉTICOS**	18
1.1	**Discretização do domínio: A Malha Escalonada Uniforme Unidimensional**	18
1.2	**Condições miméticas**	21
1.2.1	**CM.1**: Mimetizando a derivada da função constante	22
1.2.2	**CM.2**: A versão discreta do Teorema Fundamental do Cálculo	22
1.2.3	**CM.3**: Integração por partes: Relacionando os operadores **D** e **G**	24
1.3	**Acurácia uniforme em todo domínio**	26
1.4	**Construindo o operador mimético de Castillo-Grone**	28
1.4.1	Estêncil de k-ésima ordem	28
1.4.2	Operador diferencial central	29
1.4.3	Contorno mimético	32
1.4.3.1	A derivada deve ser zero para função constante	33
1.4.3.2	O Teorema fundamental do Cálculo deve ser válido	33
1.4.3.3	Produtos internos ponderados	34
1.5	**Características do operador**	37
2	**ALGORITMOS**	39
2.1	**Discretização do domínio**	39
2.2	**Cálculo do estêncil**	39
2.3	**Cálculo do operador de suporte**	42
2.3.1	Estêncil na borda	45
2.4	**Condições miméticas**	46
3	**METODOLOGIA**	51
3.1	**Modelo matemático**	51
3.2	**Discretização do domínio**	52
3.3	**Operador mimético discreto**	52
3.4	**Condições de contorno**	54
3.5	**Método Leapfrog**	54
3.6	**Comparativos**	55
3.7	**Ambiente de execução**	57
4	**RESULTADOS**	58
	CONCLUSÃO	67
	TRABALHOS FUTUROS	69
	REFERÊNCIAS	70
	APÊNDICE A – Conceitos	73

APÊNDICE B – Demonstrações 81
APÊNDICE C – Código fonte 84

INTRODUÇÃO

Na natureza ocorrem diversos fenômenos que tentamos representar através da linguagem matemática, sendo modelados através do uso de equações diferenciais parciais (EDP). Estas equações são caracterizadas por apresentar a derivada da função solução como parte de seus termos. Por exemplo a equação

$$\frac{\partial^2 u\left(\mathbf{x},t\right)}{\partial t^2} = c^2 \nabla^2 u\left(\mathbf{x},t\right) \tag{1}$$

descreve a propagação da onda $u(\mathbf{x},t)$ sobre um meio com velocidade constante c em função do tempo t e de uma posição $\mathbf{x}$ no espaço.

Resolver a EDP (1) em questão, significa encontrar uma função u que satisfaça a igualdade apresentada. A demonstração da obtenção dessa função pode ser encontrada em Iorio (2010) e é dada por

$$u(x,t) = \varphi(x+ct) + \psi(x-ct). \tag{2}$$

Considerando as funções φ e ψ que compõem a solução geral u como sendo curvas do tipo senoidal, a equação (2) torna-se

$$u(x,t) = \frac{\operatorname{sen}\left(x+ct\right) + \operatorname{sen}\left(x-ct\right)}{2}, \tag{3}$$

que é representado por duas ondas senoidais com metade da energia inicial cada, velocidade constante e movendo-se em direções opostas.

Por outro lado, em uma grande família de problemas a solução analítica pode ser extremamente complexa ou ainda pior, inexistente. Por exemplo, em

$$\frac{\partial^2 u\left(\mathbf{x},t\right)}{\partial t^2} = c^2 \nabla^2 u\left(\mathbf{x},t\right) + F\left(\mathbf{x},t\right), \tag{4}$$

onde F é uma função qualquer. Para um grande conjunto de funções F, principalmente quando F não atende os critérios de integrabilidade, ou u não é suficientemente contínua, as soluções analíticas podem não existir.

Para estes casos, métodos foram criados para aproximar a solução de uma equação qualquer, ou seja, encontrar resultados que estão próximos o suficiente da solução analítica, sem necessariamente conhecer sua solução, estes métodos são denominados de métodos numéricos.

Um dos métodos numéricos amplamente utilizados é o Método das Diferenças Finitas (MDF), que foi introduzido por Lewy, Friedrichs e Courant (1928) e amplamente discutido posteriormente com exemplos emblemáticos em Trefethen (1996), Hyman e Lar-

routurou (1992), Yee (1966), Reynolds (1978) e Shashkov (1996).

O MDF é construído a partir da discretização do operador diferencial analítico sobre uma malha pontualmente discretizada. Estas técnicas permitem aproximar-se do resultado analítico para uma dada precisão desejada, mas quanto mais preciso, mais caro computacionalmente o problema se torna. Infelizmente estas discretizações normalmente apresentam artefatos numéricos de natureza não física que podem dar origem à soluções oscilatórias ou relacionadas com a propagação do erro no passo temporal, por isso se faz necessária uma análise aprofundada das condições de convergência, estabilidade e consistência de um dado modelo.

No caso da existência de descontinuidades na solução de uma EDP foi desenvolvido o Método de Volumes Finitos, que é baseado na forma integral ao invés da equação diferencial. Ao invés de tomar pontos discretizados em uma malha, utiliza-se de células sobre a malha que aproximam a integral total de uma função f sobre um determinado intervalo na malha utilizando a média de uma célula, detalhes dos volumes finitos aplicados a problemas hiperbólicos pode ser visto em LeVeque (2002).

Em geral quando uma forma variacional do problema é apresentada, o uso do Método dos Elementos Finitos pode ser aplicado, que normalmente resulta num sistema de equações algébricas, onde a quantidade de equações está diretamente ligada aos graus de liberdade do modelo relacionado. Estas equações que buscam minimizar o erro do sistema algébrico com relação a função que ele aproxima(BATHE; WILSON, 1976).

Os Métodos Miméticos de Castillo-Grone (MMCG) estão inclusos na categoria de métodos de diferenças finitas, sendo sua principal característica ser baseado na construção de um operador matemático que modela a mecânica do contínuo do sistema físico, isto é, propõe-se a modelar as leis físicas constituintes como a lei de conservação de energia mimetizando as restrições das leis físicas. O MMCG baseia-se numa extensão do teorema da divergência de Gauss e utiliza-se deste para impor as condições dos operadores diferenciais do contínuo e das integrais de fluxo, sendo válidos invariavelmente em modelos de até três dimensões (CASTILLO; MIRANDA, 2013).

O principal ponto de diferença é que os métodos tradicionais discretizam o problema e somente então verificam sua estabilidade. O MMCG busca operadores que sejam intrinsecamente estáveis pois são limitados pelas leis que regem o problema contínuo, e este processo diretamente contribui para a estabilidade.

É importante ressaltar que no MMCG devemos discretizar o problema físico como um todo baseando-se na teoria do contínuo que formula o problema, e por outro lado, os métodos tradicionais preocupam-se somente em discretizar o operador. Consequentemente é muito provável que se o problema colocado for estável, seu análogo mimético provavelmente será.

Os MMCG tiveram sua primeira apresentação no artigo "A matrix analysis approach to higher order approximations for divergence and gradients satisfying a global conservation

law" (CASTILLO; GRONE, 2003). Foi utilizado para malhas não retangulares (BREZZI; LIPNIKOV; SHASHKOV, 2005), não uniformes unidimensionais (BATISTA; CASTILLO, 2009) e bidimensionais (CASTILLO et al., 1995), em discretizações de alta ordem com formulações regulares ou compactas(ABOUALI; CASTILLO, 2012).

Foi aplicado em problemas encontrados na sísmica como a propagação de ondas acústicas 2D (CORDOVA et al., 2014), ondas Rayleigh com baixa dispersividade (ROJAS et al., 2013) e propagação em modelos com rupturas e fricção (ROJA et al., 2007) (ROJAS et al., 2009), modelagem sísmica anisotrópica viscoelástica 3D (FERRER et al., 2014) e simulação de cenários de sequestramento de carbono(SANCHEZ; PAOLINI; CASTILLO, 2014). Também utilizado em problemas de processamento de imagens e visão computacional (BAZAN et al., 2011).

O objetivo principal desta dissertação é construir um texto que introduza o MMCG de forma didática e gradual desde sua concepção teórica sobre malhas ortogonais regulares unidimensionais, a construção dos operadores gradiente, divergente e laplaciano; demonstrar um algoritmo funcional, bem como um código fonte capaz de reproduzir o algoritmo usando a linguagem de programação Python; modelar um problema hiperbólico e para validar todas as etapas citadas acima e compará-lo com a solução baseada no Método de Diferenças Finitas.

Esta abordagem foi tomada por diversos motivos: o primeiro deles é que muitos dos artigos referenciados dão enfase a uma parte ou outra do método sem haver em um único material com todo o processo, isso dificulta a compreensão pois muitas vezes os exemplos trabalhados entre os diversos artigos são diferentes e em muitas vezes a notação não é consistente. O segundo motivo é a inexistência de material deste tópico em português e pode se tornar uma referência para alunos lusófonos interessados em modelagem matemática, principalmente alunos de curso de graduação. Assim sendo este material foi dividido em seis capítulos e três apêndices.

O Capítulo 1 apresenta os operadores miméticos, suas condições bem como apresenta o processo de construção dos operadores ea modelagem dos operadores miméticos e suas características.

O Capítulo 2 apresenta a forma computacional de obtenção do mesmo operador mimético apresentado no Capítulo 1 e assim sendo uma base necessária para preencher algumas das lacunas na passagem entre o modelo algébrico e o modelo computacional.

O Capítulo 3 apresenta a modelagem de um problema hiperbólico utilizando os métodos miméticos. Alguns conceitos apresentados anteriormente foram reintroduzidos, mas neste capítulo um cuidado especial foi dado para que toda notação apresentada seja estritamente relacionada com o problema proposto evitando eventuais confusões em nomeclaturas.

O Capítulo 4 apresenta os resultados obtidos no capítulo três e compara o método mimético de Castillo-Grone com o MDF.

Os Capítulos e apresentam resumidamente as conclusões e tópicos para serem trabalhados no futuro.

Os apêndices foram divididos em três partes, o apêndice A apresenta alguns teoremas que são utilizados durante a elaboração do operador e o apêndice B apresenta algumas demonstrações de afirmações feitas durante o texto. Estas partes foram separadas para tornar a leitura do texto mais direta.

O apêndice C apresenta uma uma implementação referência em Python 3, que é totalmente funcional, que apesar de não otimizada pode ser um ponto de partida para implementações futuras, otimizações e reescrita em outras linguagens como C ou Fortran.

1 OPERADORES MIMÉTICOS

Iremos nas próximas sessões demonstrar como obter operadores miméticos necessários para resolver equações diferenciais numericamente. Um grande número de equações diferenciais são criadas usando operadores diferenciais de primeira ordem: divergentes, rotacionais e gradientes.

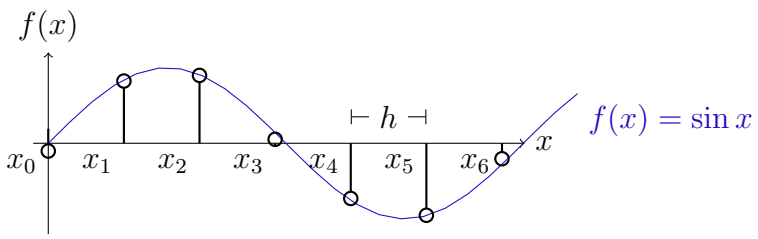
1.1 Discretização do domínio: A Malha Escalonada Uniforme Unidimensional

Como não computacionalmente possível representar as noções de infinitesimal e infinito nos computadores, é necessário transformar o espaço contínuo de um intervalo sobre a reta em uma região discreta. Esta transformação toma valores de uma função contínua em pontos discretos, isto é, sem valores em sua vizinhança (LIMA, 2012a). Assim, neste trabalho dada uma função qualquer, tomaremos valores desta função igualmente espaçados, com espaçamento fixo que chamaremos de h, como mostrado na Figura 1. Seja então o intervalo M sobre a reta real, dado por $M = [0, 1]$, particionaremos M em N intervalos, igualmente espaçados, com distância $h = \frac{1}{N}$.

Cada intervalo $[x_i, x_{i+1}]$, com $0 \leq i \leq N - 1$, será chamado de *célula* e os extremos de cada célula será chamado de *nó* e indicado pela notação x_i. Em particular, dois nós (x_0 e x_N) são especialmente nomeados como *nós de contorno*, por estarem no extremos do intervalo. Endereçamos os centros das células por $x_{i+\frac{1}{2}}$ que é exatamente o ponto médio da célula, isto é, $x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1})$ para $0 \leq i \leq n - 1$.

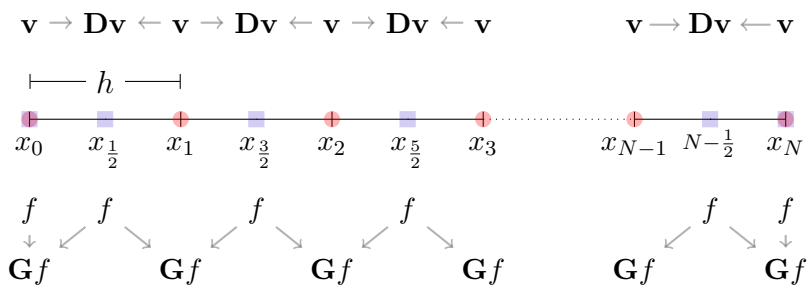
A união de todos estes nós, células e extremos é chamada de *malha intercalada*

Figura 1 - Função seno discretizada



Fonte: O autor, 2019.

Figura 2 - Malha intercalada uniforme.



Legenda: Malha intercalada uniforme (M), discretizada em N partes igualmente espaçadas (h), onde $\tilde{v}$ é calculado nos nós (mostrados como um círculo vermelho) e $\tilde{f}$ é calculado nos centros das células e nos pontos de contorno(mostrado como um quadrado azul).

Fonte: O autor, 2019.

uniforme. Chamamos de *intercalada*[1] pois queremos aproximar as derivadas dos valores nos centros das células com valores em seus nós (e vice-versa), conforme podemos ver na Figura 2.

Tomaremos então duas funções $\mathbf{v}(x)$ e $f(x)$ sobre o intervalo M, onde $\mathbf{v}(x)$ é uma função vetorial avaliada sobre os nós do intervalo e $f(x)$ uma função escalar calculada nos centros das células e nos nós de contorno (veja Figura 2).

Para simplificar nossa notação consideraremos os vetores $\mathbf{v}$ e $\mathbf{f_{cb}}$ dados por:

$$\begin{aligned} \mathbf{v} &\coloneqq (v(x_0), v(x_1), \ldots, v(x_{n-1}), v(x_n))^T, \\ \mathbf{f_{cb}} &\coloneqq \big(f(x_0), f(x_{1/2}), f(x_{3/2}), \ldots, f(x_{n-1/2}), f(x_n)\big)^T. \end{aligned} \tag{5}$$

Assim sendo, $\mathbf{v}$ representa os valores de $\mathbf{v}(x)$ em cada nó e tem dimensão $N+1$. Da mesma forma $\mathbf{f_{cb}}$ representa o valor de f no centro de cada célula e na região de contorno, que por sua vez tem dimensão $N+2$.

Iremos representar o operador do divergente como $\mathbf{D}$ e o operador gradiente como $\mathbf{G}$. $\mathbf{D}$ é atribuído ao centro da célula e é calculado com os valores de $\mathbf{v}$ obtidos pelos valores dos nós. Em contrapartida $\mathbf{G}$ é atribuido aos nós e calculado pelos valores f obtidos dos centros das células e dos pontos de contorno. Em outras palavras $\mathbf{D}$ mapeia os nós aos centros das células e $\mathbf{G}$ mapeia dos centros das células e pontos de contorno nos nós (veja Figura 2).

[1] Podemos utilizar de forma equivalente a intercalada os termos *intercalada* ou *deslocada*. Neste trabalho optamos utilizar o termo intercalada por ser uma tradução direta do inglês *staggered*. Não foi encontrada na literatura em português referências a este tipo de malha.

Queremos que os operadores diferenciais criados não sejam somente aproximações baseadas nos valores vizinhos, mas também que possuam propriedades adicionais que sejam matematicamente consistentes. Estas são:

CM.1 A derivada da função constante deve ser zero;

CM.2 O Teorema Fundamental do Cálculo deve ser satisfeito:

$$\int_0^1 \frac{d}{dx} f(x)dx = f(1) - f(0); \tag{6}$$

CM.3 Integração por partes deve ser satisfeita:

$$\int_0^1 \frac{dv}{dx} f(x)dx + \int_0^1 v(x)\frac{df}{dx}dx = v(1)f(1) - v(0)f(0). \tag{7}$$

Chamamos estas de **condições miméticas (CM)** e são necessárias para que um operador seja definido como mimético (CASTILLO; GRONE, 2003). Tomando a derivada baseando-se nos pontos vizinhos da malha intercalada obtemos os seguintes operadores discretos, baseado nas diferenças finitas centrais:

$$\mathbf{D}\, v_{i+\frac{1}{2}} = \frac{v_{i+1} - v_i}{h} \; \forall i \in [0, N-1], \tag{8}$$

$$\mathbf{G}\, f_i = \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{h} \; \forall i \in [1, N-1]. \tag{9}$$

Nos nós de contorno calculamos $\mathbf{G}\, f$ baseado nas diferenças finitas laterais como

$$\mathbf{G}\, f_0 = \frac{f_{\frac{1}{2}} - f_0}{\frac{h}{2}}, \tag{10}$$

$$\mathbf{G}\, f_N = \frac{f_N - f_{N-\frac{1}{2}}}{\frac{h}{2}}. \tag{11}$$

Nestas aproximações o erro da diferença central é da ordem $O(h^2)$ e da lateral é $O(h)$, como demonstrado em Trefethen (1996) e Shashkov (1996).

Como $\mathbf{D}$ e $\mathbf{G}$ são operadores lineares podemos então representá-los como matrizes $\mathbf{D}$ e $\mathbf{G}$, aplicadas sobre os vetores $\mathbf{v}$ e $\mathbf{f_{cb}}$ respectivamente, onde $\mathbf{D}$ tem dimensões $N \times N+1$ e $\mathbf{G}$ tem dimensões $(N+1) \times (N+2)$. Estas dimensões permitem a composição dos operadores para obter operadores diferenciais de segunda ordem que mapeiam funções sobre os pontos dos nós e dos extremos em pontos nos centros das células.

Desta forma, $\mathbf{D}$ é um mapa linear $\mathbb{R}^{N+1} \to \mathbb{R}^N$

$$(\mathbf{D})_{N\times N+1} (\mathbf{v})_{N+1\times 1} = (\mathbf{D}\,\mathbf{v})_N \,. \tag{12}$$

Fazendo com que $\mathbf{D}\,\mathbf{v}$ represente a aproximação discreta do operador divergente $\nabla \cdot \mathbf{v}$, ou

na forma matricial como

$$\mathbf{D} = \frac{1}{h}\begin{bmatrix} -1 & 1 & & & & & \\ & -1 & 1 & & & & \\ & & -1 & 1 & & & \\ & & & \ddots & \ddots & & \\ & & & & -1 & 1 & \\ & & & & & -1 & 1 \\ & & & & & & -1 & 1 \end{bmatrix}. \tag{13}$$

Da mesma forma, a matriz $\mathbf{G}$ é um mapa linear $\mathbb{R}^{N+2} \to \mathbb{R}^{N+1}$

$$(\mathbf{G})_{(N+1)\times(N+2)}\,(\mathbf{f_{cb}})_{(N+2)\times 1} = (\mathbf{G}\,\mathbf{f_{cb}})_{(N+1)\times 1}\,, \tag{14}$$

que aplicado sobre $\mathbf{f_{cb}}$ representa a aproximação discreta do operador gradiente ∇f, isto é $\mathbf{G}\ \mathbf{f_{cb}} \approx \nabla f$, assim reescrevemos a matriz $\mathbf{G}$ como

$$\mathbf{G} = \frac{1}{h}\begin{bmatrix} -2 & 2 & & & & & \\ & -1 & 1 & & & & \\ & & -1 & 1 & & & \\ & & & \ddots & \ddots & & \\ & & & & -1 & 1 & \\ & & & & & -1 & 1 \\ & & & & & & -2 & 2 \end{bmatrix}. \tag{15}$$

Repare que a primeira e última linha de $\mathbf{G}$, tem agregada a condição de contorno do gradiente conforme definido em (10) e (11).

1.2 Condições miméticas

A representação na forma matricial, mesmo já incluídas as condições de contorno não são suficientes para a construção dos operadores miméticos. Devemos também adicionar as condições miméticas definidas em **CM.1**, **CM.2** e **CM.3**, como discutido a seguir.

1.2.1 **CM.1**: Mimetizando a derivada da função constante

O primeiro critério que deve ser satisfeito é que a derivada de uma função constante deve ser zero, conforme definimos em **CM.1**. Seja uma função N-dimensional representada pelo vetor

$$\mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^N \tag{16}$$

então a condição **CM.1** discretizada pode ser escrita como

$$D\mathbf{1} = 0. \tag{17}$$

Esta equação equivale dizer que a soma de todos elementos das colunas de cada linha de D deve ser zero. Equivalentemente,

$$G\mathbf{1} = 0. \tag{18}$$

Podemos perceber que ambos operadores matriciais D e G gozam desta propriedade, como podemos ver em (13) e (15). Assim, satisfazem a primeira condição mimética.

1.2.2 **CM.2**: A versão discreta do Teorema Fundamental do Cálculo

A segunda condição representa uma equivalência discreta do Teorema Fundamental do Cálculo. Para tal precisamos definir a integral sobre uma malha discreta. Iremos utilizar duas estratégias, uma para aproximar a integral de uma função definida nos centros das células ($\mathbf{f_{cb}}$), outra para aproximar a integral da função definida nos nós ($\mathbf{v}$).

Usaremos a notação $\langle \cdot, \cdot \rangle$ para representar o produto interno euclidiano que aparece de maneira natural ao tomarmos a aproximação discreta da integral como veremos. Equivalentemente podemos utilizar uma multiplicação vetorial fazendo $\langle \mathbf{a}, \mathbf{b} \rangle \equiv \mathbf{b}^T\mathbf{a}$ por construção.

Para as funções $f(x)$ definidas nos centros utilizaremos a regra do ponto médio (conforme Figura 3b) que é dada por:

$$\int_0^1 f(x)dx \approx \sum_{i=0}^{N-1} f_{i+\frac{1}{2}} \cdot h = \langle \mathbf{f_{cb}}, h\mathbf{1}_{N+2} \rangle. \tag{19}$$

Da mesma forma a operação $\mathbf{D}\,\mathbf{v}$ também é mapeada nos centros das células.

Utilizando a mesma fórmula obtemos:

$$\int_0^1 \frac{d}{dx} v(x)dx \approx \langle \mathbf{D}\,\mathbf{v}, h\mathbf{1}\rangle = h\mathbf{1}^T\,\mathbf{D}\,\mathbf{v} = \sum_{i=0}^{N-1} \frac{v_{i+1}-v_i}{h} h = \mathbf{v}_N - \mathbf{v}_0 = v(1) - v(0). \tag{20}$$

Note também que o operador $\mathbf{D}$ em conjunto com a regra do ponto médio para integração numérica satisfaz exatamente a versão discreta do Teorema Fundamental do Cálculo. Iremos então expressar esta condição de forma matricial como

$$h\mathbf{1}^T\,\mathbf{D} = (-1, 0, \cdots, 0, 1)\,, \tag{21}$$

o que implica em

$$\langle \mathbf{D}\,\mathbf{v}, h\mathbf{1}\rangle = v_N - v_0 \;\; \forall v \in \mathbb{R}^{N+1}, \tag{22}$$

que equivale a restringir as somas das colunas de $h\,\mathbf{D}$ sendo iguais a $(-1, 0, \ldots, 0, 1)$, que é verdadeiro para $\mathbf{D}$ apresentado em (13).

Como para a função $v(x)$ mapeada aos nós da malha é mais conveniente utilizar a regra trapezoidal (mostrada na Figura 3a) de integração numérica(CONTE; BOOR, 1980) dada por

$$\int_0^1 v(x)dx \approx \sum_{i=0}^{N-1} \frac{1}{2}\left(v_{i+1} + v_i\right) h. \tag{23}$$

Podemos então reescrever a regra (23) sobre a malha intercalada M como

$$\sum_{i=0}^{N-1} \frac{1}{2}\left(v_{i+1} - v_i\right) h = \left(\frac{1}{2}v_0 + \sum_{i=1}^{N-1} \frac{1}{2}\left(v_{i+1} - v_i\right) + \frac{1}{2}v_n\right) h = \langle \mathbf{v}, h\mathbf{1}\rangle_P \tag{24}$$

com P sendo a matriz de pesos associada ao produto interno ponderado[2] dada por

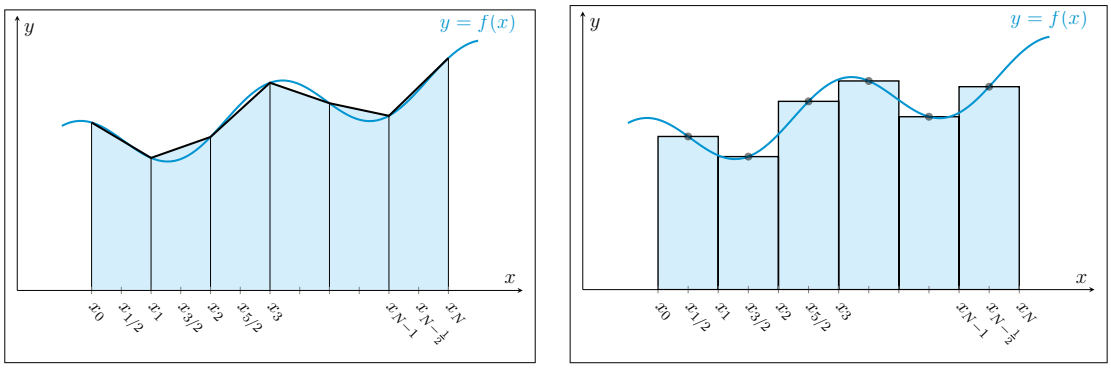
$$P = DIAG\left(\frac{1}{2}, 1, 1, \ldots, 1, 1, \frac{1}{2}\right) \in \mathbb{R}^{(N+1)\times(N+1)}. \tag{25}$$

Da mesma forma o operador $\mathbf{G}$ é mapeado aos nós do intervalo discreto, por isso também usaremos a integração numérica pela regra do trapézio assim como foi feito com a função $\mathbf{v}(x)$. Desta forma obtemos

$$\langle \mathbf{G}\,\mathbf{f_{cb}}, h\mathbf{1}\rangle_P = h\mathbf{1}^T P\,\mathbf{G}\,\mathbf{f_{cb}} = f_N - f_0. \tag{26}$$

[2] O produto interno ponderado é introduzido com mais detalhes no apêndice A.1.

Figura 3 - Comparação entre os métodos de integração numérica na malha intercalada



Legenda: A Figura (a) mostra os polígonos utilizados para a integração numérica através da regra do trapézio. A Figura (b) mostra os retângulos utilizados para a regra do ponto médio calculados nos centros das células da malha intercalada.

Fonte: O autor, 2019.

Expressando em função de P e $\mathbf{G}$, temos

$$h\mathbf{1}^T P\,\mathbf{G} = (-1, 0, 0, \ldots, 0, 0, 1) \tag{27}$$

demonstrando que $h\mathbf{1}^T P\,\mathbf{G}$ satisfaz o Teorema Fundamental do Cálculo para qualquer $\mathbf{f_{cb}}$ dado. Esta restrição é equivalente a exigir que as somas das colunas de $hP\,\mathbf{G}$ seja igual a $(-1, 0, \ldots, 0, 1)$.

Concluindo: com os operadores $\mathbf{D}$ e $\mathbf{G}$ dados em (13) e (15) respectivamente, mostramos que

$$hP\,\mathbf{G} = h\,\mathbf{D} \tag{28}$$

e que a soma das colunas de $hP\,\mathbf{G}$ e $h\,\mathbf{D}$ resulta em $(-1, 0, \ldots, 0, 1)$, isto é, satisfazem o Teorema Fundamental do Cálculo.

1.2.3 **CM.3**: Integração por partes: Relacionando os operadores $\mathbf{D}$ e $\mathbf{G}$

Como último critério a ser satisfeito temos a integração por partes (ver **CM.3**) que relaciona os operadores $\mathbf{D}$ e $\mathbf{G}$. O primeiro problema a ser resolvido é a diferença das dimensões de $\mathbf{D}\,\mathbf{v} \in \mathbb{R}^N$ e $\mathbf{f_{cb}} \in \mathbb{R}^{N+2}$, conforme mostrado em (13) e (15) respectivamente. Adicionaremos ao operador $\mathbf{D}$ condições de contorno de Neumann, isto é, a derivada da

função $\mathbf{v}(x)$ é nula nas bordas,

$$\begin{cases} v_0' = 0 \\ v_N' = 0 \end{cases} \tag{29}$$

Na forma matricial criaremos o operador $\hat{\mathbf{D}}$ que mapeia o operador $\mathbf{D}$ já obtido, acrescentando as condições de Neumann, ou seja

$$\hat{\mathbf{D}} = \begin{bmatrix} 0 & \cdots & 0 \\ & \mathbf{D} & \\ 0 & \cdots & 0 \end{bmatrix} \tag{30}$$

fazendo com que independente dos valores do primeiro e últimos elementos de $\mathbf{v}$, a sua derivada é sempre nula. Resolvido o problema de dimensionalidade pois $\hat{\mathbf{D}} \in \mathbb{R}^{(N+2)\times(N+1)}$ podemos reescrever a integração por partes (**CM.3**), como

$$\begin{aligned} \langle \hat{\mathbf{D}}\mathbf{v}, \mathbf{f_{cb}} \rangle + \langle \mathbf{v}, \mathbf{G}\,\mathbf{f_{cb}} \rangle &= v_N f_N - v_0 f_0 \\ \langle \hat{\mathbf{D}}\mathbf{v}, \mathbf{f_{cb}} \rangle + \langle \mathbf{v}, \mathbf{G}\,\mathbf{f_{cb}} \rangle &= \langle B\mathbf{v}, \mathbf{f_{cb}} \rangle \\ h \left\langle \left(\hat{\mathbf{D}} + \overset{T}{\mathbf{G}}\, P \right) v, \mathbf{f_{cb}} \right\rangle &= \langle B\mathbf{v}, \mathbf{f_{cb}} \rangle \end{aligned} \tag{31}$$

que nos leva a igualdade

$$\begin{aligned} h \left(\hat{\mathbf{D}} + \overset{T}{\mathbf{G}}\, P \right) &= B \\ \left(\hat{\mathbf{D}} + \overset{T}{\mathbf{G}}\, P \right) &= \frac{1}{h} B \end{aligned} \tag{32}$$

com

$$B = \begin{pmatrix} -1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(N+2)\times(N+1)}, \tag{33}$$

que é a condição mimética para satisfazer a integração por partes (**CM.3**). Sendo também

equivalente dizer que $\mathbf{G}$ deve ter a forma:

$$\mathbf{G} = \begin{pmatrix} -1 & & 0 \\ 0 & & 0 \\ \vdots & -\mathbf{D}^T & \vdots \\ 0 & & 0 \\ 0 & & 1 \end{pmatrix} \tag{34}$$

1.3 Acurácia uniforme em todo domínio

Nesta sessão iremos incluir a acurácia uniforme ao operador em todo domínio, isto é, a acurácia nos pontos de contorno será a mesma que a existente nos pontos internos, assim a acurácia será a mesma em todos os pontos onde o operador é definido.

Adicionalmente o Método Mimético de Castillo-Grone (MMCG) permite criar operadores discretos que são uniformemente precisos até a k-ésima ordem, para qualquer k par. Nas próximas sessões iremos obter os operadores de quarta ordem, conforme mostrados em Castillo et al. (2000), Runyan (2011) e Abouali e Castillo (2012),

Para ilustrar, tomemos como exemplo o operador mimético de Castillo-Grone (que ainda não aprendemos a encontrar) de segunda ordem dado pelas matrizes

$$P = DIAG\left(\frac{3}{8}, \frac{9}{8}, 1, \ldots, 1, \frac{9}{8}, \frac{3}{8}\right) \in \mathbb{R}^{(N+1)\times(N+1)} \tag{35}$$

$$\mathbf{G} = \frac{1}{h}\begin{pmatrix} \frac{8}{3} & 3 & \frac{-1}{3} & & & & & \\ 0 & -1 & 1 & & & & & \\ & 0 & -1 & 1 & & & & \\ & & & \ddots & \ddots & & & \\ & & & & -1 & 1 & 0 & \\ & & & & & -1 & 1 & 0 \\ & & & & & \frac{1}{3} & -3 & \frac{8}{3} \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+2)} \tag{36}$$

$$\mathbf{D} = \frac{1}{h}\begin{pmatrix} 0 & \cdots & & & & & \\ -1 & 1 & & & & & \\ & -1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & -1 & 1 & & \\ & & & & -1 & 1 & \\ & & & & & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{(N+2)\times(N+1)} \tag{37}$$

$$B = \begin{pmatrix} -1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(N+2)\times(N+1)} \tag{38}$$

é fácil ver que

$$h\left(\hat{\mathbf{D}} + \overset{T}{\mathbf{G}}\, P\right) \neq B. \tag{39}$$

Por isso definimos a matriz $\tilde{B}$ dada por

$$\tilde{B} \equiv h\left(\hat{\mathbf{D}} + \overset{T}{\mathbf{G}}\, P\right). \tag{40}$$

Desta forma por definição reescrevemos (31) como

$$h\left\langle \left(\hat{\mathbf{D}} + \overset{T}{\mathbf{G}}\, P\right) v, f \right\rangle = \langle \tilde{B} v, f \rangle \tag{41}$$

que nos dá

$$\tilde{B} = \begin{pmatrix} -1 & & & & & & \\ \frac{1}{8} & -\frac{1}{8} & & & & & \\ -\frac{1}{8} & \frac{1}{8} & 0 & & & & \\ & & 0 & \ddots & & & \\ & & & \ddots & 0 & & \\ & & & & 0 & -\frac{1}{8} & \frac{1}{8} \\ & & & & 0 & \frac{1}{8} & -\frac{1}{8} \\ & & & & & & 1 \end{pmatrix} \tag{42}$$

Os operadores **D** e **G** apresentados possuem acurácia de segunda ordem em todo domínio (inclusive nas bordas), veremos agora como construir este operador.

1.4 Construindo o operador mimético de Castillo-Grone

A contribuição de Castillo e Grone (2003)[3] aos operadores miméticos foi a elaboração dos operadores que além de atender às restrições das leis físicas constituintes, possuem a mesma ordem de acurácia em sua região de contorno.

O MMCG, permite construir operadores de ordem arbitrária k, para qualquer k par (CASTILLO; GRONE, 2017). Neste exemplo construiremos os operadores divergente e gradiente de quarta ordem ($k = 4$).[4]

1.4.1 Estêncil de k-ésima ordem

Iniciaremos elaborando um estêncil que aproxima a derivada em cada ponto de interesse com precisão arbitrária k, devido a simetria esperada nas formulações miméticas construiremos coeficientes centrais.

É importante que, como estamos em uma malha intercalada, para obter o operador *gradiente* utilizaremos os nós da malha e para obter o operador *divergente* utilizaremos os centros das células (veja Figura 2). Por construção as funções vetoriais são calculados nos nós e as funções escalares são computadas nos centros das células.

[3] A presente sessão baseia nos algoritmos e técnicas descritas em Runyan (2011) e Sanchez e Castillo (2013).

[4] O código fonte em Python pode ser encontrado no apêndice C.

Aproximaremos as funções em torno dos pontos de interesse utilizando as expansões de Taylor (LIMA, 2012b), de forma que possamos encontrar a melhor aproximação dada a acurácia desejada [5].

Para encontrar os coeficientes de Taylor utilizaremos o método apresentado por Spitzbart e Macon (1957) e Gregory (1957), que consistem em resolver um sistema linear cujo a matriz de coeficientes é dado por uma matriz de Vandermonde[6].

Desta forma, os coeficientes do estêncil do divergente de ordem 4 da diferença finita central é encontrado ao resolver o seguinte sistema linear

$$Ts = \mathbf{e}_n, \tag{43}$$

onde T é a matriz de Vandermonde cheia gerada pelo vetor $\mathbf{g}$, uma outra forma de escrever é $T = V(\mathbf{g})$, que é dado pelos k pontos vizinhos mais próximos *relativos* ao ponto que queremos aproximar, em particular $\mathbf{g} = \left(-\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}\right)$. Continuando, $\mathbf{s}$ é o vetor do estêncil que queremos encontrar e $\mathbf{e}_n$ é um vetor na forma que possui 0 em todo lugar, exceto no índice da função que queremos calcular, isto é $\mathbf{e}_n = \mathbf{e}_0 = (1, 0, 0, \cdots)$, caso desejemos aproximar a função, $\mathbf{e}_n = \mathbf{e}_1 = (0, 1, 0, \cdots)$, para o caso de queremos aproximar a função primeira derivada, $\mathbf{e}_n = \mathbf{e}_2 = (0, 0, 1, 0, \cdots)$ caso desejemos aproximar a função da segunda derivada. É importante notar que neste caso não é possível aproximar um grau de derivada maior do que a ordem do estêncil. Logo para encontrar os coeficientes do estêncil que aproximam a primeira derivada da função com acurácia $k = 4$, precisamos resolver

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -\frac{3}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{3}{2} \\ \left(-\frac{3}{2}\right)^2 & \left(-\frac{1}{2}\right)^2 & \left(\frac{1}{2}\right)^2 & \left(\frac{3}{2}\right)^2 \\ \left(-\frac{3}{2}\right)^3 & \left(-\frac{1}{2}\right)^3 & \left(\frac{1}{2}\right)^3 & \left(\frac{3}{2}\right)^3 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{44}$$

Cujo a solução $\mathbf{s} = \left(\frac{1}{24}, -\frac{9}{8}, \frac{9}{8}, -\frac{1}{24}\right)$.

1.4.2 Operador diferencial central

O próximo passo é construir o operador diferencial. Como estamos interessados em aproximar as derivadas nos pontos de contorno, devemos começar por estes pontos.

[5] A expansão detalhada da aproximação de Taylor para o problema exposto pode ser vista no apêndice A.4.

[6] A abordagem utilizada relacionada as matrizes de Vandermonde são explicadas no apêndice A.2.

Iremos então gerar uma matriz $\mathbf{D}(A)$ (ou $\mathbf{G}(B)$), cujo a maior parte da sua diagonal é composta pelos elementos do estêncil obtido, isto é:

$$\mathbf{D}(A) = \frac{1}{h}\begin{bmatrix} A & & \\ & \mathbf{D} & \\ & & A^* \end{bmatrix}. \tag{45}$$

As dimensões da submatriz A (e analogamente B)[7], dependem exclusivamente da acurácia do operador sendo construido. Em geral a submatriz A tem dimensões $k \times l$, com $l = \frac{3}{2}k$, que no caso de quarta ordem ($k = 4$) temos $l = 6$, então para o operador D gerado pelo estêncil s teríamos:

$$\mathbf{D}(A) = \frac{1}{h}\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & & & \cdots \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & & & \cdots \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & & & \cdots \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & & & \cdots \\ & & & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & & \cdots \\ & & & & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & \cdots \\ & & & & & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & \cdots \\ & & & & & & & & \ddots \end{bmatrix} \tag{46}$$

A matriz (46) está sendo representada somente na sua porção superior esquerda, como vimos anteriormente $\mathbf{D}(A) \in \mathbb{R}^{(N+2)\times(N+1)}$, devemos ajustar a submatriz A dentro de $\mathbf{D}$ de forma que $\mathbf{D}$ possua as propriedades miméticas e tenha acurácia similar ao operador nos pontos interiores, que no caso em questão é de quarta ordem.

Os elementos de A ainda devem ser descobertos. Da mesma forma iremos utilizar as expansões de Taylor, aplicadas com as matrizes de Vandermonde para encontrar cada uma das linhas de A. Para a primeira linha em particular queremos encontrar $\mathbf{D}\,\mathbf{v}$ ao redor do indice $x_{\frac{3}{2}}$, para tal utilizaremos os l pontos vizinhos na malha intercalada que contribuem para o divergente (veja Figura 2), isto é os pontos $x_1, x_2, x_3, x_4, x_5, x_6$, cujo a distância relativa ao índice de interesse é dado por $\left(1-\frac{3}{2}, 2-\frac{3}{2}, 3-\frac{3}{2}, 4-\frac{3}{2}, 5-\frac{3}{2}, 6-\frac{3}{2}\right) = \left(-\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \frac{9}{2}\right) = g_1$, nosso vetor gerador, entao precisamos resolver

$$V(g_1)A_1 = \mathbf{e}_1 \tag{47}$$

[7] Devido a simetria do operador diferencial $D(A)$, A^* é encontrada em conjunto com a matriz A e será discutido adiante. O mesmo vale para $G(B)$.

com

$$g_1 = \left(-\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \frac{9}{2}\right) \tag{48}$$

isto é

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -\frac{1}{2} & \frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2} & \frac{9}{2} \\ \left(-\frac{1}{2}\right)^2 & \left(\frac{1}{2}\right)^2 & \left(\frac{3}{2}\right)^2 & \left(\frac{5}{2}\right)^2 & \left(\frac{7}{2}\right)^2 & \left(\frac{9}{2}\right)^2 \\ \left(-\frac{1}{2}\right)^3 & \left(\frac{1}{2}\right)^3 & \left(\frac{3}{2}\right)^3 & \left(\frac{5}{2}\right)^3 & \left(\frac{7}{2}\right)^3 & \left(\frac{9}{2}\right)^3 \\ \left(-\frac{1}{2}\right)^4 & \left(\frac{1}{2}\right)^4 & \left(\frac{3}{2}\right)^4 & \left(\frac{5}{2}\right)^4 & \left(\frac{7}{2}\right)^4 & \left(\frac{9}{2}\right)^4 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \\ a_{16} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{49}$$

Para simplificar a notação faremos referência a i-ésima linha da submatriz A por

$$A_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}, a_{i6}) \;\; 0 \leq i \leq k \tag{50}$$

Em geral a matrizes de Vandermonde cheias (isto é quadradas) são não singulares, isto é, possuem solução única (MACON; SPITZBART, 1958) (VERDE-STAR, 1988). Este não é o caso em (49), pois não é quadrada[8].

Neste caso em particular a solução para A_i é dada por:

$$A_i^T = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \\ a_{16} \end{bmatrix} = \begin{bmatrix} -\frac{11}{12} \\ \frac{17}{24} \\ \frac{3}{8} \\ -\frac{5}{24} \\ \frac{1}{24} \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} -1 \\ 5 \\ -10 \\ 10 \\ -5 \\ 1 \end{bmatrix}, \tag{51}$$

onde α_1 é um escalar real qualquer e $(-1, 5, -10, 10, -5, 1)$, é o núcleo de $V(g_1)$A.2. Faremos o mesmo para as outras 3 linhas de A, isto é A_2, A_3 e A_4, com os respectivos

[8] No apêndice A.2.1 algumas características das matrizes de Vandermonde utilizadas nesta sessão.

geradores g_i.

$$A_1^T = \begin{bmatrix} -\frac{11}{12} & \frac{17}{24} & \frac{3}{8} & -\frac{5}{24} & \frac{1}{24} & 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix} \tag{52}$$

$$A_2^T = \begin{bmatrix} \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix} \tag{53}$$

$$A_3^T = \begin{bmatrix} 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix} \tag{54}$$

$$A_4^T = \begin{bmatrix} 0 & 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} \end{bmatrix} + \alpha_4 \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix}. \tag{55}$$

de forma resumida, podemos reescrever como

$$A(\alpha) = \Pi + \alpha\nu^T, \tag{56}$$

com

$$\Pi = \begin{bmatrix} -\frac{11}{12} & \frac{17}{24} & \frac{3}{8} & -\frac{5}{24} & \frac{1}{24} & 0 \\ \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & 0 \\ 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 \\ 0 & 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} \end{bmatrix} \tag{57}$$

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix} \tag{58}$$

$$\nu = \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix}. \tag{59}$$

Existe um conjunto infinito de soluções possíveis para α e consequentemente para $A(\alpha)$, destes deveremos procurar as submatrizes $A(\alpha)$ que satisfazem as condições mimé-ticas de Castillo-Grone.[9].

1.4.3 Contorno mimético

O próximo passo é encontrar os coeficientes α_i eq. (58) da submatriz eq. $A(\alpha)$ (56) da matriz eq. $\mathbf{D}(A)$ (46), que tornam $\mathbf{D}(A)$ mimético e preservam a acurácia do operador em todo domínio.

[9] As soluções para este problema foram encontradas utilizando o software de computação simbólica SymPy (MEURER et al., 2017) baseado na linguagem Python (ROSSUM, 1995).

1.4.3.1 A derivada deve ser zero para função constante

Queremos então que dada uma função constante a derivada seja sempre zero. Representamos uma função constante sobre uma malha discreta como o vetor $\mathbf{v}$ cujo todos os elementos sejam iguais, em particular tomemos

$$\mathbf{v} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{60}$$

que quando aplicada sobre a submatriz A, deve ser 0 sempre isto é:

$$\mathbf{v} \cdot A_i = 0 \; \forall i \tag{61}$$

que é equivale dizer que a soma das colunas de A deve ser sempre zero.

1.4.3.2 O Teorema fundamental do Cálculo deve ser válido

Para que o Teorema Fundamental do Cálculo seja verdadeiro, precisamos encontrar um operador $\mathbf{D}$ (ou análogamente $\mathbf{G}$) que satisfaz:

$$\mathbf{D}\,\mathbf{v} = v(1) - v(0) \tag{62}$$

em particular para a função constante discretizada temos

$$\mathbf{1}^T\,\mathbf{D}(A(\alpha)) = \begin{bmatrix} -1 & 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{63}$$

Tomando a equação (46), percebe-se que seus pontos interiores são simétricos e a soma de cada linha é 0. Por isso devemos nos preocupar somente com as colunas de $\mathbf{D}$ que também possuam elementos da submatriz A.

Queremos que a soma das $l = 6$ primeiras colunas de $\mathbf{D}(A)$ sejam $(-1, 0, 0, 0, 0, 0)$ que é um dos extremos do intervalo do Teorema Fundamental do Cálculo discreto (o outro extremo do intervalo é obtido por anti-simetria, como veremos adiante), mas queremos fazer isso somente em função de A sem necessitar de toda matriz $\mathbf{D}(A)$.

Queremos então que a soma das colunas de A seja dada por

$$\mathbf{1}^T A = \left(-1, 0, 0, -\frac{1}{24}, \frac{13}{12}, -\frac{1}{24}\right) \equiv \mathbf{d}^T. \tag{64}$$

É interessante notar que os elementos da soma das colunas de D que não dependem somente da submatriz A (da coluna 3 à 6), devem ter suas somas compensadas pelos

valores do estêncil dos pontos internos[10].

Baseado no resultado da equação (64), devemos então procurar os coeficientes α_i da submatriz A que satisfazem esta restrição. De (56) e (64) temos

$$\mathbf{1}^T A(\alpha) = \mathbf{1}^T \left(\Pi + \alpha\nu^T\right) = \mathbf{1}^T\Pi + \left(\mathbf{1}^T\alpha\right)\nu^T. \tag{65}$$

Reescrevendo (65) substituindo (64), temos

$$\begin{aligned} \mathbf{1}^T\Pi + \left(\mathbf{1}^T\alpha\right)\nu^T &= \mathbf{d}^T \\ \mathbf{1}^T\alpha\nu^T &= \mathbf{d}^T - \mathbf{1}^T\Pi \\ k\nu^T &= \mathbf{d}^T - \mathbf{1}^T\Pi. \end{aligned} \tag{66}$$

pois $\mathbf{1}^T\alpha \in \mathbb{R}$, uma constante diferente de zero (a soma de todos os α_i). É demonstrado em Castillo e Grone (2003) que a equação (66) não possui solução com relação ao produto interno padrão. Em particular podemos notar que $d^T - \mathbf{1}^T\Pi$ tem elementos somente em um subespaço de $\mathbb{R}^6$ enquanto ν possui elementos em todo $\mathbb{R}^6$, assim é impossível a igualdade e é exatamente baseado neste argumento que a demonstração é apresentada.

Dado esta impossibilidade teremos que procurar outros espaços onde este produto interno nos resulte em vetores paralelos. Usaremos então os conceitos apresentados no apêndice A.1 para gerar tais produtos.

1.4.3.3 Produtos internos ponderados

Iremos então construir uma matriz quadrada $N \times N$ definida como:

$$Q = DIAG(q_1, q_2, \ldots, q_{k-1}, q_k, 1, \ldots, 1, q_k, q_{k-1}, \ldots, q_2, q_1) \tag{67}$$

que chamaremos de matriz de peso.

Desta forma, ao invés de resolver a equação (63) resolveremos

$$\mathbf{1}^T Q\,\mathbf{D}(A(\alpha)) = \frac{1}{h}\begin{bmatrix} -1 & 0 & \ldots & 0 & 1 \end{bmatrix} \tag{68}$$

que equivale a resolver a condição mimética

$$\mathbf{1}^T Q A(\alpha) = \mathbf{d}^T. \tag{69}$$

[10] Uma abordagem algoritmica será apresentada no capítulo 2, bem como o código fonte deste algoritmo que é visto no apêndice C.

Fazendo

$$\mathbf{1}^T Q = \mathbf{q}^T \tag{70}$$

podemos reescrever (69) como

$$\begin{aligned} \mathbf{q}^T A(\alpha) &= \mathbf{d}^T \\ A(\alpha)^T \mathbf{q} &= \mathbf{d} \end{aligned} \tag{71}$$

Substituindo com (56)

$$\begin{aligned} A(\alpha)^T \mathbf{q} &\equiv \left(\Pi^T + \nu\alpha^T\right) \mathbf{q} \\ &= \Pi^T \mathbf{q} + \nu \left(\alpha^T \mathbf{q}\right) \end{aligned} \tag{72}$$

$\alpha^T \mathbf{q}$ é escalar pois $\alpha \in \mathbb{R}^{1,k}$ e $\mathbf{q} \in \mathbb{R}^{k,1}$ (k é a ordem de acurácia) definiremos como

$$\alpha^T \mathbf{q} = \lambda \tag{73}$$

podendo então reescerver (72) substituindo (73)

$$A(\alpha)^T \mathbf{q} \equiv \Pi^T \mathbf{q} + \lambda\nu \tag{74}$$

Iremos reordenar os termos de (74) de forma que possam ser expressos por um produto de matrizes e também para isolar os termos desconhecidos. Assim teremos

$$\left[\begin{array}{c|c} \Pi^T & \nu \end{array}\right] \left[\begin{array}{c} \mathbf{q} \\ \hline \lambda \end{array}\right] \tag{75}$$

que são as matrizes Π^T aumentada com o vetor ν a direita e o vetor $\mathbf{q}$. Pelo exposto na eq. (75) e a igualdade (71), temos então que resolver

$$\tilde{\Pi}\tilde{\mathbf{q}} = \mathbf{d} \tag{76}$$

com $\tilde{\Pi}$ e $\tilde{\mathbf{q}}$ sendo a primeira e segunda matrizes de (75) respectivamente. Dados explici-

tamente no nosso problema por

$$\begin{bmatrix} -\frac{11}{12} & \frac{1}{24} & 0 & 0 & -1 \\ \frac{17}{24} & -\frac{9}{8} & \frac{1}{24} & 0 & 5 \\ \frac{3}{8} & \frac{9}{8} & -\frac{9}{8} & \frac{1}{24} & -10 \\ -\frac{5}{24} & -\frac{1}{24} & \frac{9}{8} & -\frac{9}{8} & 10 \\ \frac{1}{24} & 0 & -\frac{1}{24} & \frac{9}{8} & -5 \\ 0 & 0 & 0 & -\frac{1}{24} & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \lambda \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ -\frac{1}{24} \\ \frac{13}{12} \\ -\frac{1}{24} \end{bmatrix}. \tag{77}$$

A solução para este sistema é única dada por:

$$\tilde{\mathbf{q}}^T = \begin{bmatrix} \frac{649}{576} & \frac{143}{192} & \frac{75}{64} & \frac{551}{576} & -\frac{25}{13824} \end{bmatrix}. \tag{78}$$

Desta forma encontramos a matriz Q e a constante λ são dadas por:

$$Q = DIAG\left(\frac{649}{576}, \frac{143}{192}, \frac{75}{64}, \frac{551}{576}, 1, \ldots, 1, \frac{551}{576}, \frac{75}{64}, \frac{143}{192}, \frac{649}{576}\right) \tag{79}$$

e

$$\lambda = -\frac{25}{13824}. \tag{80}$$

Assim como

$$\alpha^T \mathbf{q} = \lambda, \tag{81}$$

podemos escrever

$$\alpha^T \mathbf{q} = \lambda, \tag{82}$$

$$\begin{bmatrix} \frac{649}{576} & \frac{143}{192} & \frac{75}{64} & \frac{551}{576} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = -\frac{25}{13824}, \tag{83}$$

$$\frac{649}{576}\alpha_1 + \frac{143}{192}\alpha_2 + \frac{75}{64}\alpha_3 + \frac{551}{576}\alpha_4 = -\frac{25}{13824}. \tag{84}$$

Esta equação define a hiper superfície 3D em $\mathbb{R}^4$ no qual α deve pertencer, desta forma $D(A(\alpha))$ é uma família de 3 parâmetros.

Em particular se tomarmos

$$\alpha_2 = \alpha_3 = \alpha_4 = 0 \tag{85}$$

de forma que cada linha exceto a primeira e a última sejam as derivadas de quarta ordem

nas diferenças finitas centrais, teremos

$$\alpha^T q = \frac{649}{576}\alpha_1 = -\frac{25}{13824} \tag{86}$$

$$\alpha_1 = -\frac{25}{15576} \tag{87}$$

Assim podemos rescrever (88) como

$$\mathbf{D}(A) = \frac{1}{h}\begin{bmatrix} -\frac{4751}{5192} & \frac{909}{1298} & \frac{6091}{15576} & -\frac{1165}{5192} & \frac{129}{2596} & -\frac{25}{15576} & 0 & 0 & 0 & \cdots \\ \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & \cdots \\ & & & & & & & & & \ddots \end{bmatrix} \tag{88}$$

dados

$$\mathbf{q} = \left(\frac{649}{576}, \frac{143}{192}, \frac{75}{64}, \frac{551}{576}\right) \tag{89}$$

e

$$Q = DIAG\left(q_1, q_2, q_3, q_4, 1, 1, \ldots, 1, 1, q_4, q_3, q_2, q_1\right). \tag{90}$$

1.5 Características do operador

Adicionalmente as condições miméticas, existem outras propriedes esperadas que o operador possua dada suas características geométricas.

A primeira delas é que a matriz do operador seja uma matriz esparsa em comparação com a ordem da matriz, visto que somente os valores diagonais e adjacentes serão diferentes de zero, do contrário o custo computacional do operador será elevado.

O segundo é que a computação de uma aproximação qualquer é dada somente pelos pontos vizinhos, assim a matriz do operador é formada em banda.

O terceiro ponto é a similaridade com uma matriz do tipo Toeplitz, isto é, os valores das diagonais são repetidos, que pode ser utilizado como uma vantagem computacional, tanto para armazenamento como para computabilidade.

A quarta característica é a simetria, isto pode ser melhor representado dado uma

matriz de permutação de ordem n

$$P_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{91}$$

e um operador mimético M, é verdadeira a equação:

$$P_n M P_m = -M \tag{92}$$

com n e m sendo a dimensão necessária para que a multiplicação matricial faça sentido. Chamamos a matriz que satisfaz a condição (92) de centro simétrico deslocada.

Assim sendo as características esperadas do operador mimético M (divergente ou gradiente) são a saber:

- M tem soma de linhas zero;
- M tem soma de colunas $-1, 0, \ldots, 0, 1$;
- M é em banda;
- M tem estrutura Toeplitz em suas linhas interiores;
- M é centro simétrico deslocada.

2 ALGORITMOS

Nesta sessão iremos apresentar os algoritmos para resolver alguns dos problemas encontrados. Estes algoritmos serão utilizados para a elaboração do programa de computador capaz de realizar as diferenças finitas através dos métodos miméticos. O algoritmo é dividido nas seguintes partes:

1. Discretização do domínio;
2. Cálculo do Estêncil (MDF);
3. Cálculo do operador de Suporte (MOS); e
4. Cálculo do operador mimético de Castilho Grone (MCG).

Iremos apresentar cada um deles de forma objetiva fazendo referência aos códigos fontes do apêndice C e da descrição do método apresentada no capítulo 1.

2.1 Discretização do domínio

É necessário que o número de pontos discretizados da malha N seja maior que $3k-1$, onde k é a ordem do operador. Esta se faz necessária para que durante o cálculo do operador mimético o sistema linear da submatriz dos pontos de contorno de um dos lados do intervalo não se sobreponha ao outro lado do intervalo. No caso exemplo, precisaremos de uma discretização mínima de 11 pontos na malha.

2.2 Cálculo do estêncil

O estêncil tem como parâmetro a acurácia do operador que pretendemos obter. Quanto maior acurácia, maior o operador e consequentemente maior o custo computacional para obter o operador e computar cada passo temporal.

Apesar do MDF permitir a criação de operadores de qualquer acurácia, o MMCG somente existe para valores pares de acurácia, por isso esta restrição deve ser levada em conta quando gerado o estêncil inicial. Para representar a acurácia (também chamada de ordem), utilizaremos a constante k, que não será alterada durante todo o processo. Ao expandir nossos exemplos utilizaremos sempre $k = 4$. Pois $k = 2$ é trivial e $k \geq 6$ cria operadores demasiadamente grandes para serem confortavelmente apresentados em uma dissertação, sem acrescentar em nada no método, visto que os algoritmos neste capítulo serão generalizados para um k par qualquer.

Algoritmo 1 - Calcula os pontos vizinhos na malha intercalada.

DOCUMENTAÇÃO

TÍTULO

Pontos vizinhos na malha intercalada

PROPÓSITO

Calcular os pontos em uma malha intercalada vizinhos ao ponto de interessa para obtenção da aproximação de Taylor.

ENTRADAS

k: Ordem/Acurácia do operador

h: Tamanho do intervalo entre pontos

SAÍDAS

s: Pontos vizinhos para o operador de ordem k

OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS

Para que o estêncil gerado possa ser transformado em operador mimético de Castillo-Grone é necessário que a ordem k seja par. Não iremos colocar esta restrição no algoritmo para simplificação mas ela deve constar em qualquer implementação real.

FIM DOCUMENTAÇÃO

ALGORITMO VETORGERADOR

declarar k, h, i **numéricos**

declarar g **vetor**

1. $i \leftarrow 1$
2. **para** x **de** $\frac{h-k}{2}$ **até** $\frac{k}{2}$ **passo** h, **fazer**
3. | $g_i \leftarrow x$
4. | $i \leftarrow i + 1$
5. **fim para**

FIM ALGORITMO

Legenda: O algoritmo apresentado tem como objetivo dados a ordem de acurácia k e o espaçamento uniforme h entre os pontos da malha intercalada, retornar as posições vizinhas a um ponto qualquer para a obtenção das aproximações de Taylor para a derivada.

Definida a ordem, precisamos identificar os pontos na malha que serão os vizinhos ao ponto sendo calculado na malha intercalada. Tomando a distância entre pontos como sendo h (quando representado numericamente utilizaremos a constante $h = 1$, pois a malha é uniforme), o operador mimético tem como característica ser centro-simétrico por isso optaremos por utilizar um estencil de diferenças finitas centrais, para tal obteremos $\frac{k}{2}$ pontos a esquerda e $\frac{k}{2}$ pontos a direita. do ponto de interesse onde estamos calculando o operador derivada.

Para o caso exemplo o algoritmo 1 nos dá o vetor $\mathbf{g} = \begin{pmatrix} -3/2 & -1/2 & 1/2 & 3/2 \end{pmatrix}$. O próximo passo é obter os coeficientes do estêncil das diferenças finitas, utilizaremos a matriz $T = V(\mathbf{g})$ de Vandermonde cheia gerada pelo vetor $\mathbf{g}$ conforme mostrado no

apêndice A.2. Como queremos obter as aproximações da primeira derivada utilizaremos o vetor resultado $\mathbf{o} = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}$. Agora precisamos resolver

$$T\,\mathbf{s} = \mathbf{o}. \tag{93}$$

Para obter a solução de (93) podemos utilizar tanto o método dos mínimos quadrados (detalhado em Trefethen (1997)), como uma solução simbólica (vistos em Gathen (2013) e Geddes Stephen R. Czapor (1992)), ambas resultam no mesmo vetor $\mathbf{s} = \begin{pmatrix} 1/24 & -9/8 & 9/8 & -1/24 \end{pmatrix}$, pois T é sempre não singular (MACON; SPITZBART, 1958) para qualquer valores de $\mathbf{g}$ diferentes entre si. Desta forma computacionalmente obteríamos o algoritmo 2.

Algoritmo 2 - Computa o estêncil utilizando as matrizes de Vandermonde.

DOCUMENTAÇÃO
  TÍTULO
    Estencil de diferenças finitas centrais
  PROPÓSITO
    Obter o estêncil de ordem k das diferenças finitas centrais para a primeira derivada
  ENTRADAS
    k: Ordem do operador
    $\mathbf{g}$: Vetor gerador
  SAÍDAS
    s: Estêncil de ordem k associado a diferença finita central da primeira derivada
  OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS
    As funções *vandermonde* e *linsolve* são definidas no pacote matemático.
FIM DOCUMENTAÇÃO
ALGORITMO ESTENCILMDF
  declarar k, i **numéricos**
  declarar g, o, s **vetor**
  declarar T **matriz**
1. **para** i **de** 1 **até** k , **fazer**
2. | **se** $(i \neq 1)$, **então**
3. | | $o_i \leftarrow 0$
4. | **senão**
5. | | $o_i \leftarrow 1$
6. | **fim se**
7. **fim para**
8. $T \leftarrow \text{vandermonde}(g)$
9. $s \leftarrow \text{linsolve}(T, o)$
FIM ALGORITMO

De posse do vetor $\mathbf{s}$ podemos então fabricar a matriz D do operador diferencial para os N pontos da malha discreta, conforme mostrado no algoritmo 3.

Algoritmo 3 - Cálculo da matriz para estêncil.

DOCUMENTAÇÃO

TÍTULO

Matriz do estêncil de diferenças finitas

PROPÓSITO

Obter a matriz do operador diferencial aplicado aos N pontos da malha discretizada.

ENTRADAS

s: Vetor estêncil

N: Número de pontos da malha discretizada

SAÍDAS

O: Matriz do operador de diferenças finitas para os N pontos da malha intercalada.

OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS

As dimensões da matriz O de saida dependem diretamente do tamanho do estencil s e do número de pontos discretizados N. Para o gradiente temos $N = n + 2$ e para o divergente $N = n + 1$, com n sendo o número de nós da malha intercalada.

FIM DOCUMENTAÇÃO

ALGORITMO MATRIZESTENCIL

declarar N, l **numéricos**

declarar g **vetor**

declarar O **matriz**

1. $l \leftarrow tamanho(g)$
2. $O_{i,j} \leftarrow 0 \; \forall i, j \mid 1 \leq i \leq N$ e $1 \leq j \leq N + l$ $\{Inicializa\ O\}$
3. **para** i **de** 1 **até** N , **fazer**
4. | **para** j **de** 1 **até** l , **fazer**
5. | | $O_{i,i+j} \leftarrow g_j$
6. | **fim para**
7. **fim para**

FIM ALGORITMO

Legenda: Obtêm a matriz baseada no estêncil s para o cálculo do operador central das diferenças finitas nos N pontos da malha discretizada.

2.3 Cálculo do operador de suporte

Devemos então modificar o operador O obtido no algorítmo 3, de forma que satisfaça as condições miméticas em especial nos pontos de contorno. Este método é ligeiramente diferente caso desejemos obter o operador gradiente ou divergente, principalmente por que são computados em pontos diferentes da malha intercalada conforme mostrado na Figura 2.

Operador divergente

Nos pontos interiores o operador O satisfaz as condições miméticas, mas na borda é necessário um tratamento adicional. Computaremos as matrizes de Vandermonde nestes pontos em especial, para as k linhas da borda, serão necessário computar $\frac{3}{2}k$ colunas, chamaremos este valor de l para simplificar, isto é $l = \frac{3}{2}k$. Serão criados então k vetores geradores com $1 \leq j \leq k$, e j o índice para os vetores gerados. Conforme pode ser visto no algoritmo 4.

Algoritmo 4 - Calcula vetor gerador do operador Divergente no contorno.

DOCUMENTAÇÃO

TÍTULO

Vetor gerador para o contorno do Divergente

PROPÓSITO

Calcular o vetor gerador da matriz de Vandermonde associada aos pontos de contorno do domínio discretizado.

ENTRADAS

k: Ordem/Acurácia do operador.
l: Número de colunas da submatriz.
h: Tamanho do intervalo entre pontos
i: Índice do i-ésimo elemento do gerador. $1 \leq i \leq l$
j: Índice da j-ésimo vetor gerado. $1 \leq j \leq k$

SAÍDAS

g^j: j-ésimo vetor gerador.

OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS

Para que o estêncil gerado possa ser transformado em operador mimético de Castillo-Grone é necessário que a ordem k seja par. Não iremos colocar esta restrição no algoritmo para simplificação mas ela deve constar em qualquer implementação real.

FIM DOCUMENTAÇÃO

ALGORITMO GERADORCONTORNODIVERGENTE

declarar k, l, h, i, j **numéricos**
declarar g **vetor**
1. $l \leftarrow \frac{3}{2}k$
2. $i \leftarrow 1$
3. **para** x **de** $-\frac{h}{2}$ **até** $h\left(l - \frac{1}{2}\right)$ **passo** h, **fazer**
4. | $g_i^j \leftarrow x - j$
5. | $i \leftarrow i + 1$
6. **fim para**

FIM ALGORITMO

Operador gradiente

Os pontos de fronteira do operador gradiente são obtidas de forma similar, exceto que, para o primeiro e último pontos da malha o gradiente possui tratamento especial, assim sendo esta condição deve ser inserida no vetor gerador. O algoritmo 5 mostra esta abordagem.

Algoritmo 5 - Calcula vetor gerador do operador Gradiente no contorno.

DOCUMENTAÇÃO
TÍTULO
Vetor gerador para o contorno do Gradiente

PROPÓSITO
Calcular o vetor gerador da matriz de Vandermonde associada aos pontos de contorno do domínio discretizado.
ENTRADAS
k: Ordem/Acurácia do operador.
l: Número de colunas da submatriz.
h: Tamanho do intervalo entre pontos
i: Índice do i-ésimo elemento do gerador. $1 \leq i \leq l$
j: Índice da j-ésimo vetor gerado. $1 \leq j \leq k$
SAÍDAS
g^j: j-ésimo vetor gerador.
OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS
Para que o estêncil gerado possa ser transformado em operador mimético de Castillo-Grone é necessário que a ordem k seja par.
FIM DOCUMENTAÇÃO
ALGORITMO GERADORCONTORNOGRADIENTE
declarar k, l, h, i, j **numéricos**
declarar g **vetor**
1. $l \leftarrow \frac{3}{2}k$
2. $g_1 \leftarrow 0$
3. $i \leftarrow 2$ {*O laço* ***para*** *começa no segundo elemento*}
4. **para** x **de** $\frac{h}{2}$ **até** $h\left(l - \frac{1}{2}\right)$ **passo** h, **fazer**
5. | $g_i^j \leftarrow x - j$
6. | $i \leftarrow i + 1$
7. **fim para**

FIM ALGORITMO

2.3.1 Estêncil na borda

Sanchez e Castillo (2013) mostram que é necessário computar somente os d primeiro pontos de contorno, onde d é a dimensão do núcleo (nulidade) da matriz de Vandermonde[11] gerada pelo primeiro ponto de contorno. Para o gradiente o posto é $d = \frac{k}{2}$ e para o divergente $d = \frac{k}{2} - 1$. Que no nosso caso temos $d_G = 2$ e $d_D = 1$, como pode ser visto no algoritmo 6.

Algoritmo 6 - Calcula o estêncil na borda do operador.

DOCUMENTAÇÃO
TÍTULO
Estêncil nos pontos de contorno para o Divergente

PROPÓSITO
Calcular os coeficientes de Taylor para os pontos do contorno do operador divergente.
ENTRADAS
k: Ordem/Acurácia do operador.
h: Intervalo entre pontos.
d: Dimensão do núcleo da matriz de Vandermonde associada ao contorno.
g: Matriz dos geradores da matriz de vandermonde dos pontos do contorno.
i: Contador auxiliar.
r: Vetor auxiliar.
SAÍDAS
Pi: Estencil da borda do operador.
Nu: Núcleo da matriz de Vandermonde associada ao estêncil do contorno.
FIM DOCUMENTAÇÃO
ALGORITMO CONTORNODIVERGENTE
declarar k, h, i, d **numéricos**
declarar g **matriz**
declarar Π **matriz**
declarar ν **matriz**
1. $d = \frac{k}{2} - 1$
2. $o = VetorSeletor(k+1)$
3. $g = GeradorContornoDivergente(k, h)$
4. **para** i **de** 1 **até** d , **fazer**
5. | $V \leftarrow Vandermonde(g^i)$
6. | $r \leftarrow linsolve(V, o)$
7. | $linha_i(\Pi) = r$
8. | $linha_i(\nu) = Nucleo(V)$
9. **fim para**

FIM ALGORITMO

[11] A demonstração pode ser vista em A.2.1.

2.4 Condições miméticas

Obtemos os elementos da matriz Π (da equação (57)) que diferem do estêncil padrão do operador de diferenças finitas centrais ao utilizar o algoritmo e o núcleo ν pelo algoritmo 6. Estamos próximos de construir a matriz que será o operador definitivo, mas antes precisamos forçar as condições miméticas no operador de forma que satisfaçam **CM.1**, **CM.2** e **CM.3**, para tal devemos então encontrar os coeficientes α (58) corretos de $A(\alpha)$. Tal sistema não possui solução, construiremos então um sistema linear que coloca os valores conhecidos separado dos desconhecidos como mostrado na equação (75). Vamos direto à construção deste sistema, pois todo desenvolvimento já foi realizado no capítulo 1. Para tal devemos resolver o sistema linear $\tilde{\Pi}\tilde{q} = d$. Podemos obter $\tilde{\Pi}$ através do algoritmo 7, e para garantir as propriedades miméticas encontraremos o vetor d pelo algoritmo 8.

Uma vez definidos todas as funções, o código principal para o cálculo do operador é dado pelo algoritmo 10.

Obtidos $\tilde{\Pi}$ e d, nos resta resolver o sistema linear de forma que os pesos q_i e λ sejam encontrados. Com estes poderemos obter os α_i que preservam as propriedade miméticas e obter a submatriz $A(\alpha)$ através do algoritmo 9 para o operador desejado.

Algoritmo 7 - Construção da matriz $\tilde{\Pi}$.

DOCUMENTAÇÃO
TÍTULO
Construção da matriz $\tilde{\Pi}$
PROPÓSITO
Construir a matriz $\tilde{\Pi}$ que será utilizada para encontrar os pesos para satisfazer a condição mimética.
ENTRADAS
Π: Submatriz com o estêncil de borda
ν: Núcleo da matriz de Vandermonde da borda
s: Estêncil de diferenças finitas
SAÍDAS
$\tilde{\Pi}$: Matriz estendida com o núcleo
FIM DOCUMENTAÇÃO
ALGORITMO MATRIZPITILDE
declarar $\Pi, \tilde{\Pi}, \nu$ **matriz**
declarar s **vetor**
declarar k, l, n, m, o, p **numéricos**
$k \leftarrow tamanho(s)$
$l \leftarrow \frac{3}{2}k$
$m \leftarrow \#linhas(\Pi)$ $\{$*Número de linhas de* $\Pi\}$
$n \leftarrow tamanho(\nu)$ $\{$*Nulidade de* $\nu\}$
$\tilde{\Pi}_{k,l} \leftarrow 0$ $\{$*Inicializa a matriz com* $0\}$
para i **de** 0 **até** m , **fazer**
| **para** j **de** 0 **até** l , **fazer**
| | $\tilde{\Pi}_{i,j} \leftarrow \Pi_{i,j}$
| **fim para**
fim para
$o \leftarrow m$ $\{$*Deslocamento de linhas*$\}$
$p \leftarrow m - 1$ $\{$*Deslocamento de colunas*$\}$
para i **de** 0 **até** $k - o$, **fazer**
| **para** j **de** 0 **até** $i + p$, **fazer**
| | $\tilde{\Pi}_{i+o,j} \leftarrow 0$
| **fim para**
| **para** j **de** $i + p$ **até** $i + k$, **fazer**
| | $\tilde{\Pi}_{i+o,j} \leftarrow s_{j-i}$
| **fim para**
| **para** j **de** $i + k$ **até** l , **fazer**
| | $\tilde{\Pi}_{i+o,j} \leftarrow 0$
| **fim para**
fim para
$o \leftarrow k$ $\{$*Deslocamento de linhas*$\}$
para i **de** 0 **até** n , **fazer**
| **para** j **de** 0 **até** l , **fazer**
| | $\tilde{\Pi}_{i+o,j} \leftarrow \nu_{i,j}$
| **fim para**
fim para
FIM ALGORITMO

Algoritmo 8 - Computa vetor mimético da submatriz de contorno.

DOCUMENTAÇÃO

TÍTULO

Obtenção do vetor mimético da submatriz $A(\alpha)$

PROPÓSITO

Obter o vetor que garante as condições miméticas da submatriz $A(\alpha)$.

ENTRADAS

s: Estêncil de diferenças finitas

SAÍDAS

d: Vetor mimético para submatriz $A(\alpha)$

OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS

A obtenção deste vetor independe do operador sendo modelado

FIM DOCUMENTAÇÃO

ALGORITMO VETORMIMETICO

1. $\mathbf{d} \leftarrow \mathbf{0}$ $\{$*Inicializa vetor* $\mathbf{d}$ *com* $0\}$
2. $\mathbf{d}_0 \leftarrow -1$
3. **para** i **de** 1 **até** $\frac{3}{4}k$, **fazer**
4. | soma $\leftarrow 0$
5. | **para** j **de** 1 **até** i , **fazer**
6. | | soma $\leftarrow$ soma $+s_j$
7. | **fim para**
8. | $d_{i+\frac{k}{2}} \leftarrow$ -soma
9. **fim para**

FIM ALGORITMO

Algoritmo 9 - Calcula a submatriz $A(\alpha)$.

DOCUMENTAÇÃO

TÍTULO

Cálculo da submatriz $A(\alpha)$

PROPÓSITO

Calcular a submatriz $A(\alpha)$ na borda do operador diferencial.

ENTRADAS

$\tilde{\Pi}$: Matriz $\tilde{\Pi}$ utilizada para obtenção dos pesos .

ν: Núcleo da matriz $\tilde{\Pi}$.

q: Pesos para ajustar o produto interno.

λ_i: Constantes de proporcionalidade dos pesos.

SAÍDAS

$A(\alpha)$: Submatriz da borda do operador diferencial.

FIM DOCUMENTAÇÃO

ALGORITMO SUBMATRIZA

declarar k **numérico**
declarar n **numérico**
declarar $A(\alpha)$ **matriz**
declarar α **vetor**

$k \leftarrow$ tamanho(q)
$n \leftarrow$ tamanho(λ)
para i **de** 0 **até** k , **fazer**
| $linha(A_i) \leftarrow linha(\tilde{\Pi}_i)$
fim para
para i **de** 0 **até** n , **fazer**
| $\alpha_i \leftarrow \frac{\lambda_i}{q_i}$
fim para
para i **de** 0 **até** n , **fazer**
| $linha(A_i) \leftarrow linha(A_i) + \alpha_i \nu_i$
fim para

FIM ALGORITMO

Algoritmo 10 - Calcula o Operador mimético de Castillo-Grone.

DOCUMENTAÇÃO

TÍTULO

Estêncil nos pontos de contorno para o Divergente

PROPÓSITO

Calcular os coeficientes de Taylor para os pontos do contorno do operador divergente.

ENTRADAS

k: Ordem/Acurácia do operador.

N: Número de pontos discretizados

SAÍDAS

O: Operador diferencial mimético de Castillo-Grone

OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS

N deve ser no mínimo $3k - 1$, do contrário não é possível calcular os operadores.

A função CONTORNOOPERADOR pode ser tanto do gradiente ou do divergente.)

FIM DOCUMENTAÇÃO

ALGORITMO PRINCIPAL

1. $g \leftarrow MalhaIntercalada(k)$
2. $s \leftarrow EstencilMDF(g, k)$
3. $d \leftarrow VetorMimetico(s)$
4. $\Pi, \nu \leftarrow ContornoOperador(k, h)$ $\{Operador\ pode\ ser\ divergente\ ou\ gradiente\}$
5. $\tilde{\Pi} = MatrizPiTilde(\Pi, \nu, s)$
6. $q, \lambda \leftarrow VetorPeso(\tilde{\Pi}, d)$
7. $A \leftarrow MatrizA(\tilde{\Pi}, \nu, q, \lambda)$
8. $O \leftarrow Operador(A, N)$

FIM ALGORITMO

3 METODOLOGIA

Para este trabalho resolvemos aplicar os métodos miméticos em um problema com solução analítica, desta forma é possivel analisar e comparar os resultados obtidos com suas soluções conhecidas.

3.1 Modelo matemático

O problema escolhido foi da corda elástica com densidade homogênea e com as extremidades fixas, apresentado por Iorio (2010). O modelo matemático é representado por uma equação diferencial parcial hiperbólica[12] com condições de Neumann e Dirichelet, o problema é apresentado como

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial t} = \frac{\partial u}{\partial x} \end{cases} \tag{94}$$

e velocidade (c)

$$c^2 = 1 = \frac{\kappa}{\rho} \tag{95}$$

no intervalo $M = [-1, 1]$, com uma solução analítica

$$\begin{aligned} u &= \text{sen}\,(2\pi x)\cos(2\pi t)\,, \\ v &= \cos(2\pi x)\,\text{sen}\,(2\pi t)\,, \end{aligned} \tag{96}$$

condições iniciais $(t = 0)$

$$\begin{aligned} u(x, 0) &= \text{sen}\,(2\pi x)\,, \\ v(x, 0) &= 0 \end{aligned} \tag{97}$$

e condições de contorno

$$u(-1, t) = u(1, t) = 0. \tag{98}$$

[12] A equivalência das formas da equação da onda é apresentada com mais detalhes no apêndice B.2.

3.2 Discretização do domínio

O intervalo M foi discretizado em N intervalos regulares (Δx), conforme representado pela Figura 2, com os nós dados por $x_i = i\Delta x$ $0 \leq i \leq N$, isto é, teremos $N+1$ nós que chamaremos de malha regular. Da mesma forma obtemos uma malha sobre os centros dos intervalos (chamados de células) dado os semi interios $x_{i+\frac{1}{2}} = \frac{1}{2}\left(x_i + x_{i+1}\right)$ $0 \leq i \leq N-1$, obtendo assim N pontos intervalares, chamaremos de estes de malha auxiliar. A união da malha regular com a malha auxiliar chamamos de malha intercalada uniforme. Podemos adicionalmente extender a malha auxiliar com os pontos dos extremos da malha regular, chamaremos esta de malha auxiliar extendida e analogamente a união da malha auxiliar extendida com a malha regular, teremos a malha intercalada uniforme extendida.

A função v será computada sobre a malha regular e a função u será computado sobre a malha auxiliar extendida, que para simplicidade será operada como um vetor isto é

$$v = (v(x_0), v(x_1), v(x_2), \cdots, v\left(x_{N-1}\right), v\left(x_N\right)) \tag{99}$$

$$u = \left(u(x_0), u\left(x_{1/2}\right), u\left(x_{3/2}\right), \cdots, u\left(x_{N-1/2}\right), u\left(x_N\right)\right) \tag{100}$$

É importante notar que $v \in \mathbb{R}^{N+1}$ e $u \in \mathbb{R}^{N+2}$.

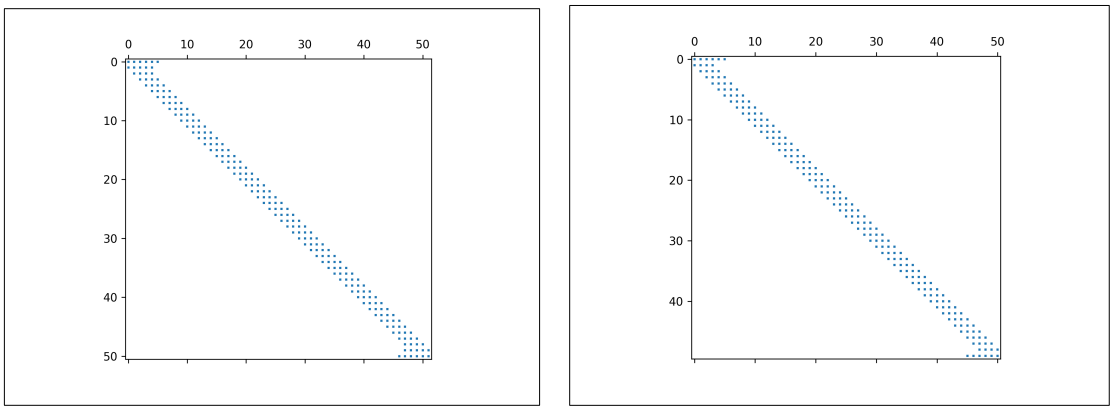
3.3 Operador mimético discreto

Obtemos o operador discreto divergente e gradiente conforme apresentado no capitulo 1, que é apresentado de forma arredondada com quatro casas decimais significativas

$$
\mathbf{D} = \frac{1}{\Delta x}\begin{bmatrix}
-1.031 & 0.7891 & 0.440 & -0.2528 & 0.056 & -0.0018 & & & & & \\
0.031 & -0.8379 & 0.8379 & -0.031 & & & & & & & \\
 & 0.0488 & -1.3184 & 1.3184 & -0.0488 & & & & & & \\
 & & 0.0399 & -1.0762 & 1.0762 & -0.0399 & & & & & \\
 & & & 0.0417 & -1.125 & 1.125 & -0.0417 & & & & \\
 & & & & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & & & 0.0417 & -1.125 & 1.125 & -0.0417 & & \\
 & & & & & & 0.0399 & -1.0762 & 1.0762 & -0.0399 & \\
 & & & & & & & 0.0488 & -1.3184 & 1.3184 & -0.0488 \\
 & & & & & & & & 0.031 & -0.8379 & 0.8379 & -0.031 \\
 & & & & & & 0.0018 & -0.056 & 0.2528 & -0.440 & -0.7891 & 1.031
\end{bmatrix} \tag{101}
$$

$$
\mathbf{G} = \frac{1}{\Delta x}\begin{bmatrix}
-1.1877 & 1.5538 & -0.5261 & .1952 & -0.0362 & 0.0009 & & & & & \\
0.1877 & -1.591 & 1.4884 & -0.0924 & 0.0073 & & & & & & \\
 & 0.0372 & -1.0049 & 1.0049 & -0.0372 & & & & & & \\
 & & 0.0426 & -1.1494 & 1.1494 & -0.0426 & & & & & \\
 & & & 0.0417 & -1.125 & 1.125 & -0.0417 & & & & \\
 & & & & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & & & 0.0417 & -1.125 & 1.125 & -0.0417 & & \\
 & & & & & & 0.0426 & -1.1494 & 1.1494 & -0.0426 & \\
 & & & & & & & 0.0372 & -1.0049 & 1.0049 & -0.0372 \\
 & & & & & & & -0.0073 & 0.0924 & -1.4884 & 1.591 & -0.1877 \\
 & & & & & & -0.0009 & 0.0362 & -0.1952 & 0.5261 & -1.5538 & 1.1877
\end{bmatrix} \tag{102}
$$

Figura 4 - Elementos não zero dos operadores



(a) (b)

Legenda: A figura (a) mostra os elementos não zero da matriz do operador gradiente, enquanto a Figura (b) mostra para o divergente. Fica evejante o quão esparsa é a matriz dos operadores.

As matrizes dos operadores $\mathbf{D}$ e $\mathbf{G}$ são construídas de forma que $\mathbf{D}$ seja um mapa de $\mathbf{v}$ para u e $\mathbf{G}$ seja uma mapa de u para $\mathbf{v}$, isto significa que as dimensões de $\mathbf{D}$ deve ser $N+1 \times N+2$ e $\mathbf{G}$ deve ser $N+2 \times N+1$. Pela Figura 4 podemos perceber que a matriz dos operadores é esparsa, permitindo otimizações para armazenamento e computação do operador.

3.4 Condições de contorno

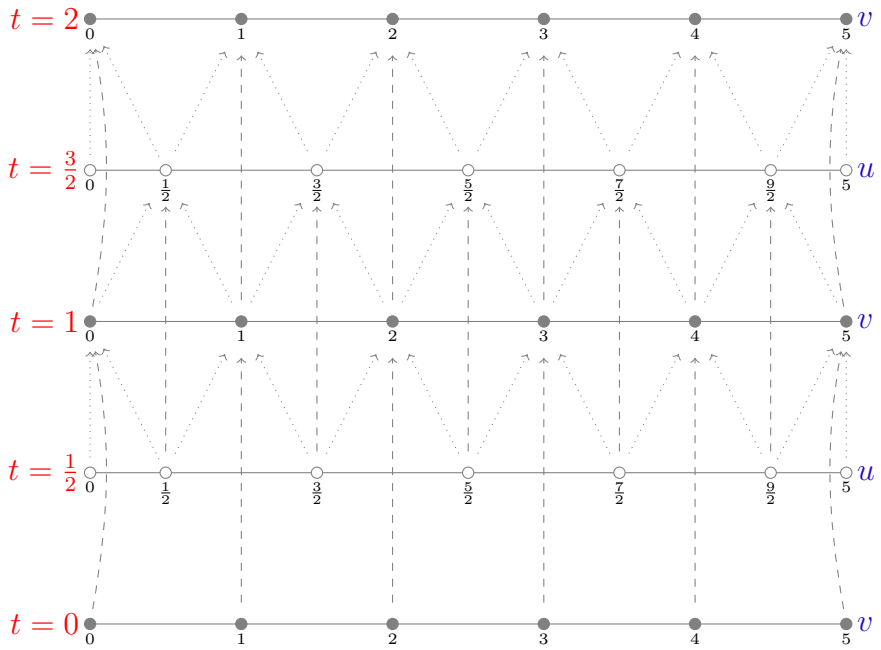
As condições de contorno de $u(1,t)$ e $u(-1,t)$ para o modelo são reforçadas aplicando-se 0 nos elementos u_0 e u_N a cada passo da iteração.

3.5 Método Leapfrog

O método Leapfrog é um método utilizado para integração numérica de equações diferencias. É um método de segunda ordem, enquanto o método de Euler é de primeira ordem apesar de requerer o mesmo número de computações por passo temporal. De acordo com Birdsall e Langdon (1985) o método é estável para movimentos oscilatórios onde o método de Euler falha desde que Δt seja constante e satisfaça $\Delta t \leq \frac{2}{\omega}$ com ω sendo a frequência.

Utiliza uma malha intercalada temporal, onde a equação espacial é computada nos instantes inteiros e a derivada do movimento é calculada no instante semi-inteiro.

Figura 5 - Método Leapfrog para as 5 primeiras iterações



Legenda: Cada linha representa a discretização do domínio em 5 intervalos ($N = 5$) em um determinado passo temporal. Em $t = 0$ e $t = \frac{1}{2}$, temos as condições iniciais para as funções v e u respectivamente. As setas mostram o esténcil para o calculo dos elementos em cada ponto. As condições de contorno de u são dadas pelo modelo, por isso não são calculadas (não há setas em direção aos pontos de contorno.). A função u é computada nos passos semi-inteiros enquanto a função v é computada nos passos inteiros. Para simplicidade utilizamos um diferencial de segunda ordem tanto para o tempo quanto para o espaço. As setas pontilhadas representam o operador diferencial espacial enquanto as setas tracejadas representam o operador diferencial temporal.

Fonte: O autor, 2019.

Discretizado como

$$V_i^{t+1} = V_i^t + \Delta t D U^{t+1/2} \tag{103}$$

$$U_i^{t+\frac{3}{2}} = U_i^{t+\frac{1}{2}} + \Delta t G V^{t+1} \tag{104}$$

onde o índice superior é o passo temporal e o inferior é a posição espacial e Δt o intervalo de tempo. A Figura 5 mostra os cinco primeiros passos temporais do método.

3.6 Comparativos

Para validação empírica do método mimético dois suportes foram tomados, o primeiro deles é uma implementação referência utilizando as diferenças finitas centrais de

quarta ordem nos pontos interiores e de segunda ordem nos pontos de contorno devido a falta de pontos suficientes para a construção de um operador de quarta ordem, este operador é mostrado em (105) e (106), chamaremos este modelo de MDF42, quarta ordem no centro, segunda ordem nas bordas.

$$D_{mdf42} = \frac{1}{\Delta x}\begin{bmatrix} -1 & 1 & & & & & & \\ 0.0417 & -1.125 & 1.125 & -0.0417 & & & & \\ & 0.0417 & -1.125 & 1.125 & -0.0417 & & & \\ & & \ddots & \ddots & \ddots & \ddots & & \\ & & & 0.0417 & -1.125 & 1.125 & -0.0417 & \\ & & & & 0.0417 & -1.125 & 1.125 & -0.0417 \\ & & & & & & -1 & 1 \end{bmatrix} \tag{105}$$

$$G_{mdf42} = \frac{1}{\Delta x}\begin{bmatrix} -2 & 2 & & & & & & \\ 0.0417 & -1.125 & 1.125 & -0.0417 & & & & \\ & 0.0417 & -1.125 & 1.125 & -0.0417 & & & \\ & & \ddots & \ddots & \ddots & \ddots & & \\ & & & 0.0417 & -1.125 & 1.125 & -0.0417 & \\ & & & & 0.0417 & -1.125 & 1.125 & -0.0417 \\ & & & & & & -2 & 2 \end{bmatrix} \tag{106}$$

Foi também elaborado um operador de quarta ordem utilizando vizinhos deslocados nos pontos próximos ao contorno do domínio, este modelo será nomeado como MDF4 e os operadores são:

$$D_{mdf4} = \frac{1}{\Delta x}\begin{bmatrix} -0.9583 & 0.875 & 0.125 & -0.0417 & & & & \\ 0.0417 & -1.125 & 1.125 & -0.0417 & & & & \\ & 0.0417 & -1.125 & 1.125 & -0.0417 & & & \\ & & \ddots & \ddots & \ddots & \ddots & & \\ & & & 0.0417 & -1.125 & 1.125 & -0.0417 & \\ & & & & 0.0417 & -1.125 & 1.125 & -0.0417 \\ & & & & 0.0417 & -0.125 & -0.875 & 0.9583 \end{bmatrix} \tag{107}$$

$$G_{mdf4} = \frac{1}{\Delta x}\begin{bmatrix} -3.0667 & 3.75 & -0.833 & 0.15 & & & & \\ 0.1333 & -1.125 & 1.1667 & -0.05 & & & & \\ & 0.0417 & -1.125 & 1.125 & -0.0417 & & & \\ & & \ddots & \ddots & \ddots & \ddots & & \\ & & & 0.0417 & -1.125 & 1.125 & -0.0417 & \\ & & & & 0.05 & -1.1667 & 1.25 & -0.1333 \\ & & & & -0.15 & 0.8333 & -3.75 & 3.0667 \end{bmatrix} \tag{108}$$

O segundo suporte foi utilizar a solução analítica da equação (94) e analizar o erro de truncamento apresentado. Fazendo também uma comparação da solução analítica contra o modelo MDF42 e MDF4.

Para computar o erro de truncamento utilizamos a norma

$$\|err(t)\|_2 = \left(\sum(\mathbf{u}(t) - \hat{\mathbf{u}}(t))^2 \Delta x\right)^{\frac{1}{2}} \tag{109}$$

onde $\mathbf{u}(t)$ é a solução analítica e $\hat{\mathbf{u}}(t)$ é uma solução numérica obtida em um determinado instante t do tempo de simulação.

3.7 Ambiente de execução

As simulação foram realizadas em um computador Intel i7-8700k, com 4.1GHz de clock, 64GB de memória RAM DDR5, com disco Samsung Electronics Co Ltd NVMe SSD Controller SM961/PM961 dedicado ao sistema operacional e disco de dados Hitachi Deskstar 7k3000 com 3TB, destes 2TB foram utilizados para armazenagem dos testes e simulações e sistema operacional Debian GNU/Linux 10.0 Buster. Todo algoritmo foi desenvolvido para execução sequencial e local, nenhum paralelismo foi aplicado. Não foram medidos tempos de execução neste trabalho. Todo software foi desenvolvido na linguagem Python3 (ROSSUM, 1995) e executado sobre CPython 3.7.1 e está disponibilizado online (MORGADO, 2018b) e a biblioteca PyMTK(MORGADO, 2018a) onde foi utilizado variáveis de precisão dupla para toda computação numérica.

4 RESULTADOS

Considerando a equação diferencial parcial hiperbólica (94), os operadores MMCG4 (101) (102), MDF42 (105) (106) e MDF4 (107) (108), e a modelagem leapfrog (104) sobre a malha intercalada apresentada no intervalo $M = [-1, 1]$ foram obtidos esperados de acordo com a literatura existente sobre o método mimético.

As Figuras 6, 8, 10 e 12 apresentam resultados da simulação da modelagem da equação (94) utilizando uma malha com 256 intervalos espacials ($\Delta x \approx 0.0078$) e número CFL 0.8 que é próximo ao limite do MMCG modelado neste trabalho. Cada quadro representa um dos quatro instantes distintos no tempo $t = \{1.26, 2.1, 3.0, 3.2\}$ escolhidos arbitrariamente, onde foi realizado um zoom de 64× em pontos críticos. O MMCG4 apresenta os melhores resultados em todos os pontos analisados, com resultados muito próximos ao modelo utiliando o operador MDF4.

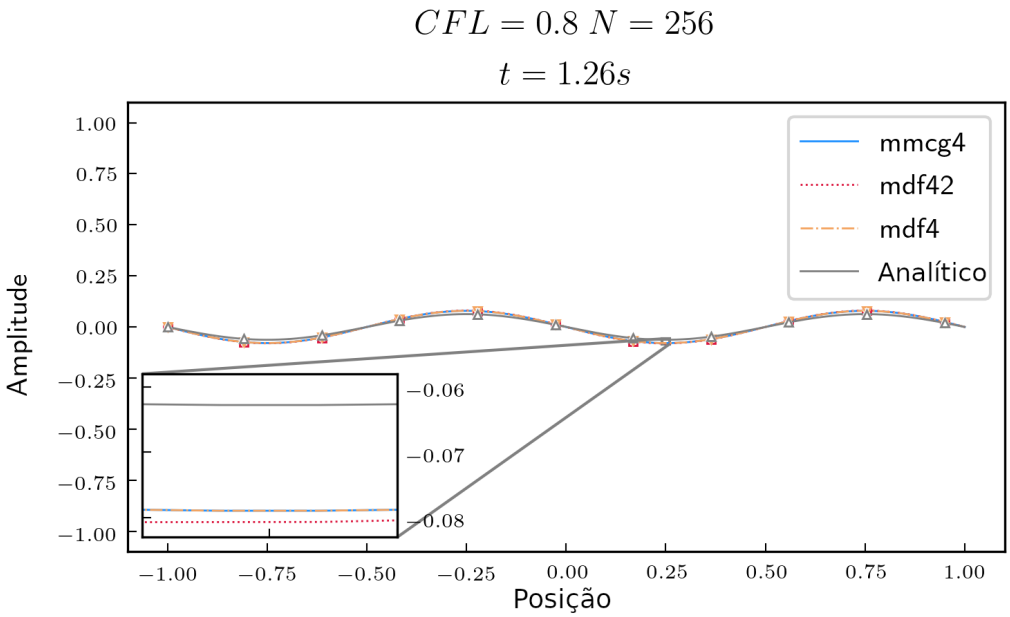
As Figuras 7, 9, 11 e 13 apresenta, o mesmo modelo sobre a perspectiva do erro de truncamento das diversas soluções numéricas. Para esta análise tomamos a diferença entre a solução numérica e a analítica computada, o valor absoluto desta diferença é então apresentado nos quadros da figura nos mesmos instantes de tempo das Figuras 6, 8, 10 e 12. A solução MDF42 apreenta oscilações de maior frequência do que as soluções MDF4 e MMCG4.

Uma análise do comportamento do modelo com relação as variações dos intervalos espaciais discretizados e o número CFL. Nesta análise, foi realizada uma simulação de 5 segundos para cada par de parametros N (número de intervalos espaciais) e CFL (número CFL). Com a simulação realizada foi computado o erro em cada posição espacial para cada passo temporal. Desta forma foi possível calcular o erro de truncamento a cada instante de tempo e foi exibido o erro máximo dada a norma L_2 (109), cnforme mostrado no algoritmo 11.

Desta forma a Figura 14 apresenta com a escala de cores os valores para o erro para este algoritmo em cada um dos modelos trabalhados.

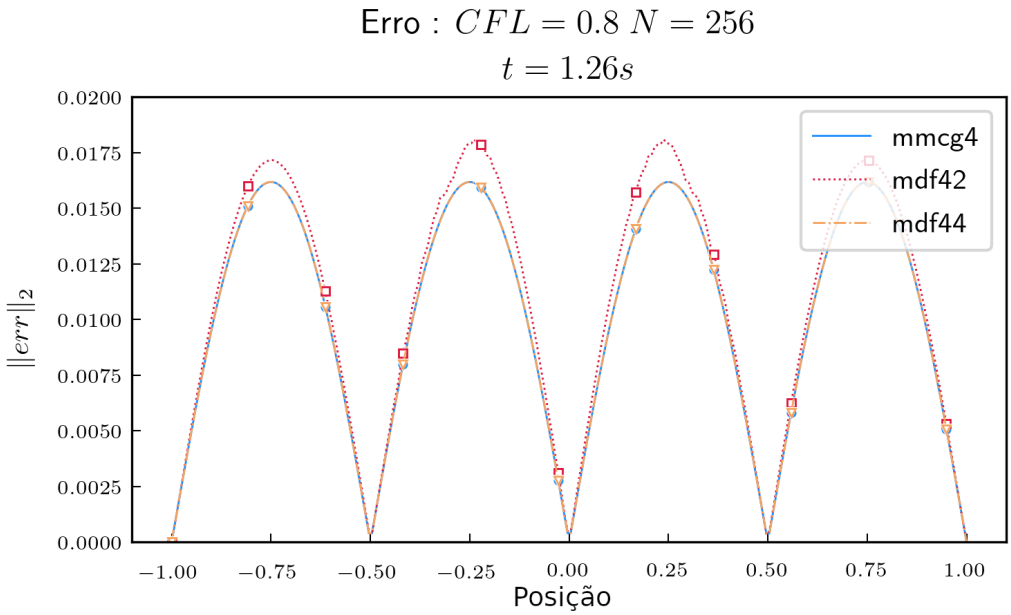
Nestes modelos é esperado que quanto mais fina a malha melhor seja a aproximação, consequentemente menor o erro de truncamento computado. Para o operador MMCG4 é esperada uma aproximação de quarta ordem(CASTILLO; GRONE, 2003) (CASTILLO; MIRANDA, 2013), isto é a razão entre a redução da malha deve apresentar uma redução do erro que seja até quatro ordens de magnitude. Problemas como arredondamento numérico também influenciam a precisão do operador computado. Podemos ver na tabela 2 que as taxas de convergência do MMCG4 é muito próxima as do MDF4 e normalmente o dobro do MDF42.

Figura 6 - Comparativo da onda em $t = 1,26s$.



Fonte: O autor, 2019.

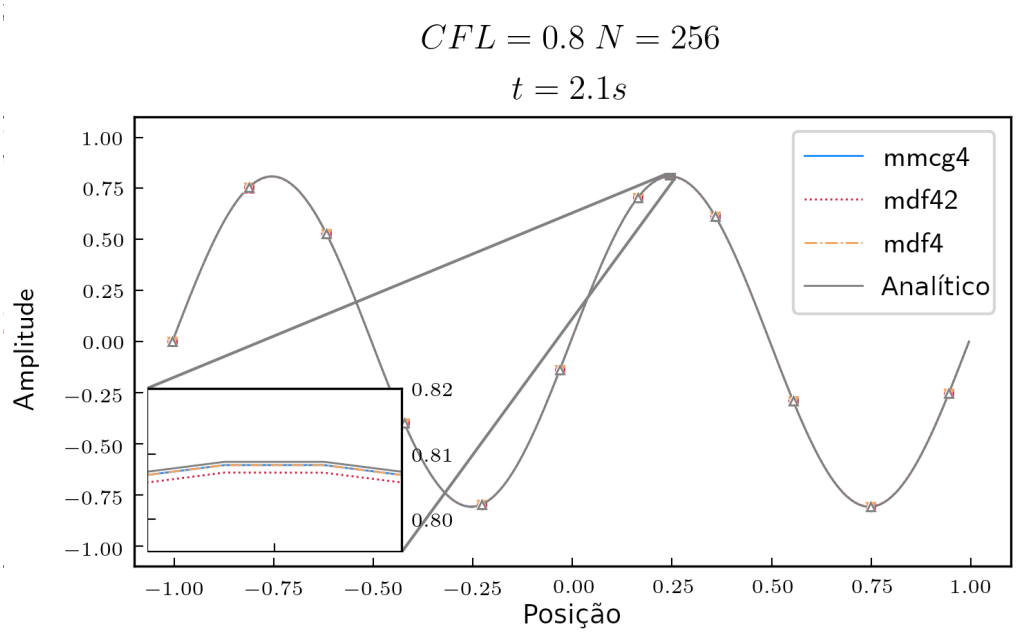
Figura 7 - Erro de truncamento em $t = 1,26$.



Legenda: A figura mostra o erro da equação (94), apresentado nas mesmas condições da Figura 6.

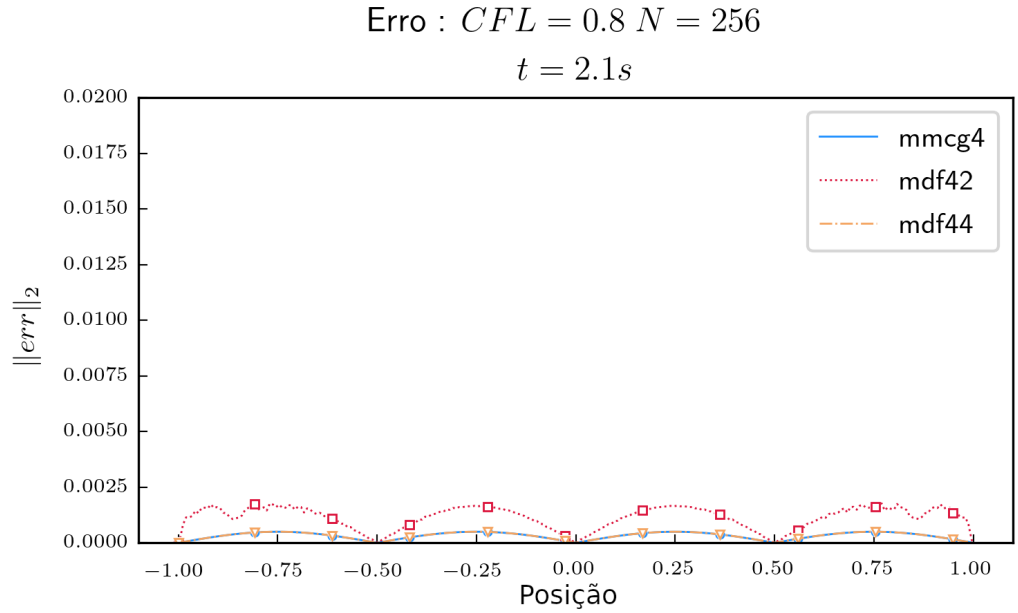
Fonte: O autor, 2019.

Figura 8 - Comparativo da onda em $t = 2,10s$.



Fonte: O autor, 2019.

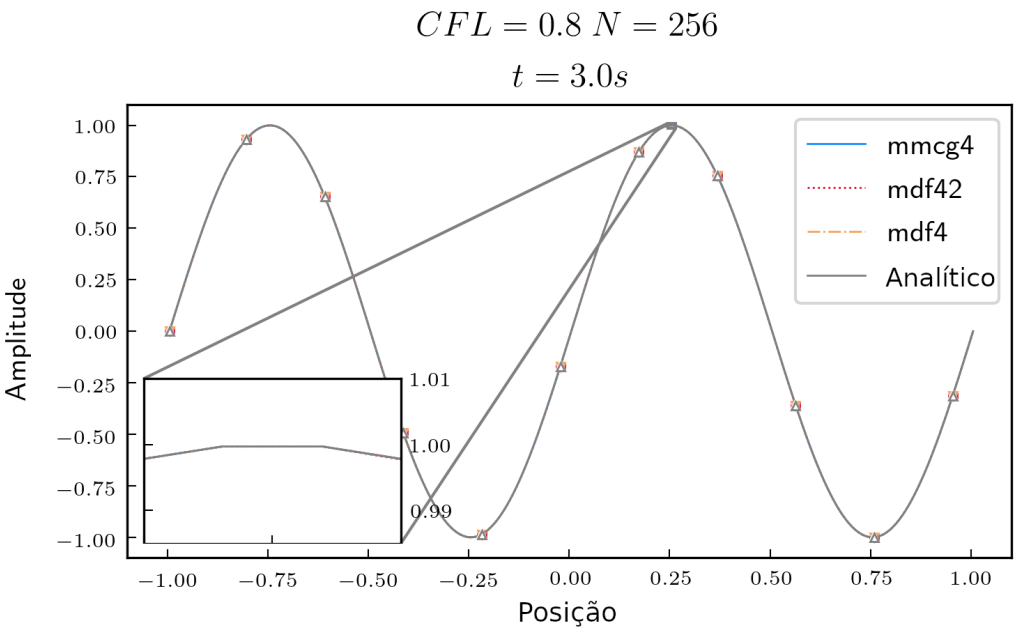
Figura 9 - Erro de truncamento em $t = 2,10$.



Legenda: A figura mostra o erro da equação (94), apresentado nas mesmas condições da Figura 8.

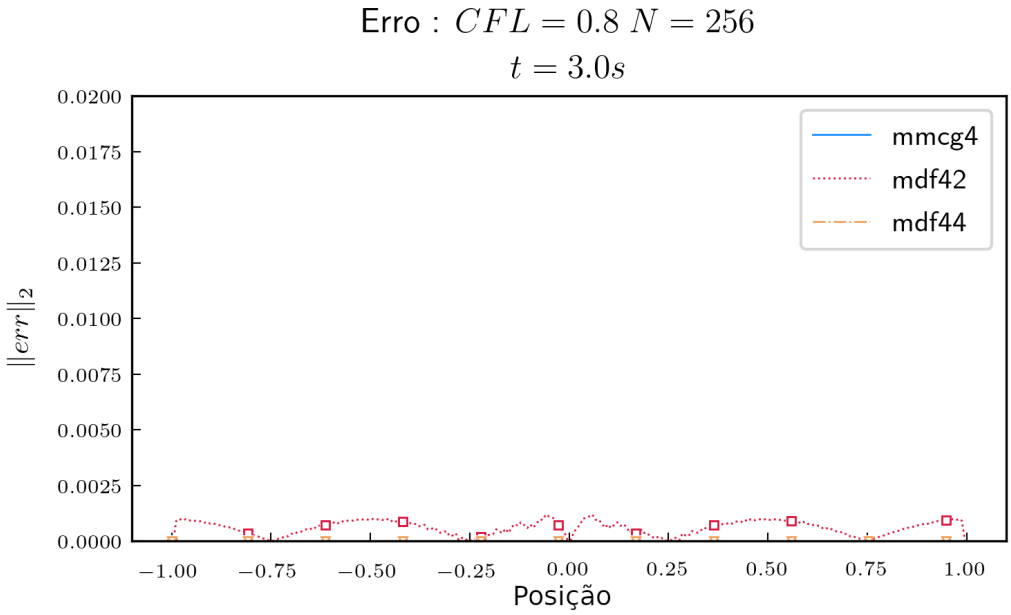
Fonte: O autor, 2019.

Figura 10 - Comparativo da onda em $t = 3,00s$.



Fonte: O autor, 2019.

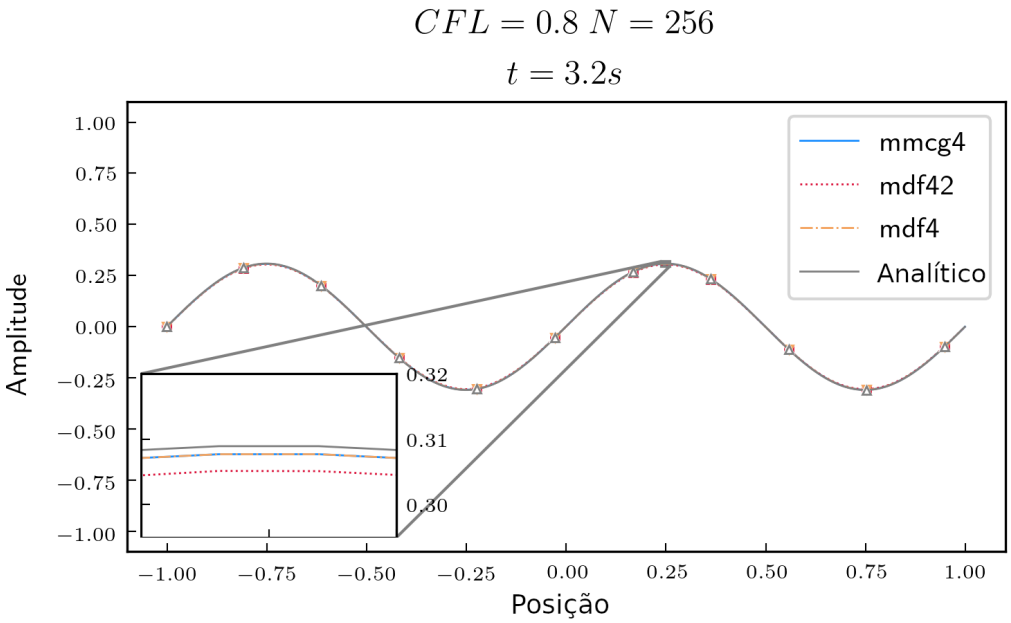
Figura 11 - Erro de truncamento em $t = 3,00$.



Legenda: A figura mostra o erro da equação (94), apresentado nas mesmas condições da Figura 10.

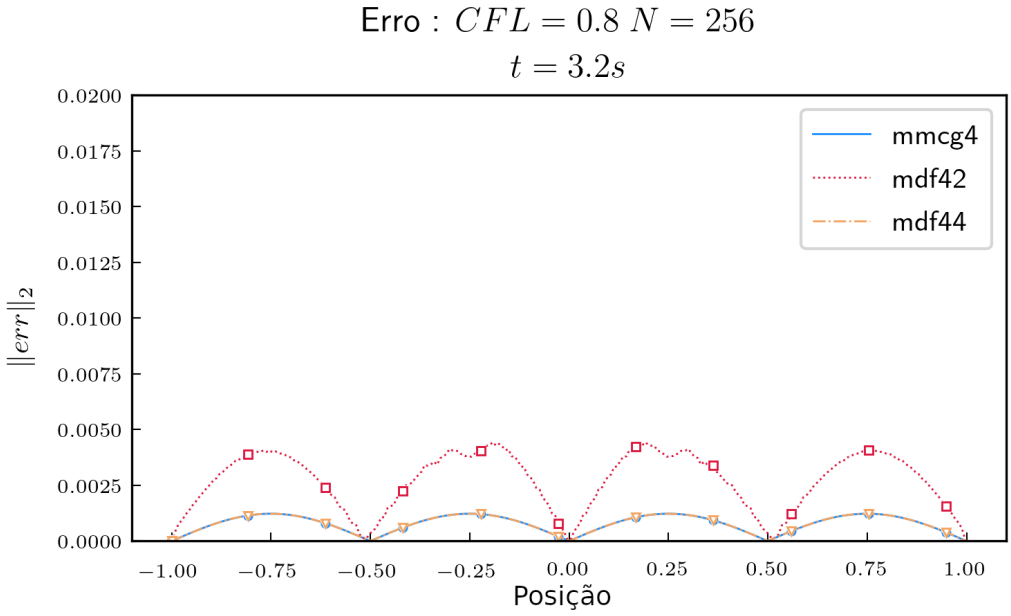
Fonte: O autor, 2019.

Figura 12 - Comparativo da onda em $t = 3,20s$.



Fonte: O autor, 2019.

Figura 13 - Erro de truncamento em $t = 3,20$.



Legenda: A figura mostra o erro da equação (94), apresentado nas mesmas condições da Figura 12.

Fonte: O autor, 2019.

Tabela 1 - Erro máximo.

		$max_{t\in[0,4]}(\|err\|_2)$		
Δx	Δt	$MDF42$	$MDF4$	$MMCG4$
$\frac{1}{20}$	$\frac{1}{180}$	1.9840×10^{-2}	5.6920×10^{-4}	$\mathbf{2.7246 \times 10^{-5}}$
	$\frac{1}{250}$	1.9302×10^{-2}	1.2187×10^{-4}	$1.9750 \times 10^{+0}$
$\frac{1}{40}$	$\frac{1}{400}$	1.1561×10^{-2}	1.9027×10^{-4}	1.7132×10^{-4}
	$\frac{1}{770}$	1.1384×10^{-2}	$\mathbf{8.6564 \times 10^{-6}}$	1.0467×10^{-5}
$\frac{1}{80}$	$\frac{1}{770}$	5.9363×10^{-3}	5.6874×10^{-5}	5.6270×10^{-5}
	$\frac{1}{2500}$	5.8829×10^{-3}	1.7256×10^{-6}	$\mathbf{1.1211 \times 10^{-6}}$
$\frac{1}{120}$	$\frac{1}{1250}$	3.9694×10^{-3}	2.6168×10^{-5}	2.6088×10^{-5}
	$\frac{1}{5000}$	3.9444×10^{-3}	3.0955×10^{-7}	$\mathbf{2.2968 \times 10^{-7}}$

Legenda: Erros máximos para cada modelo em variações distintas tanto na malha espacial como na temporal.

Fonte: O autor, 2019.

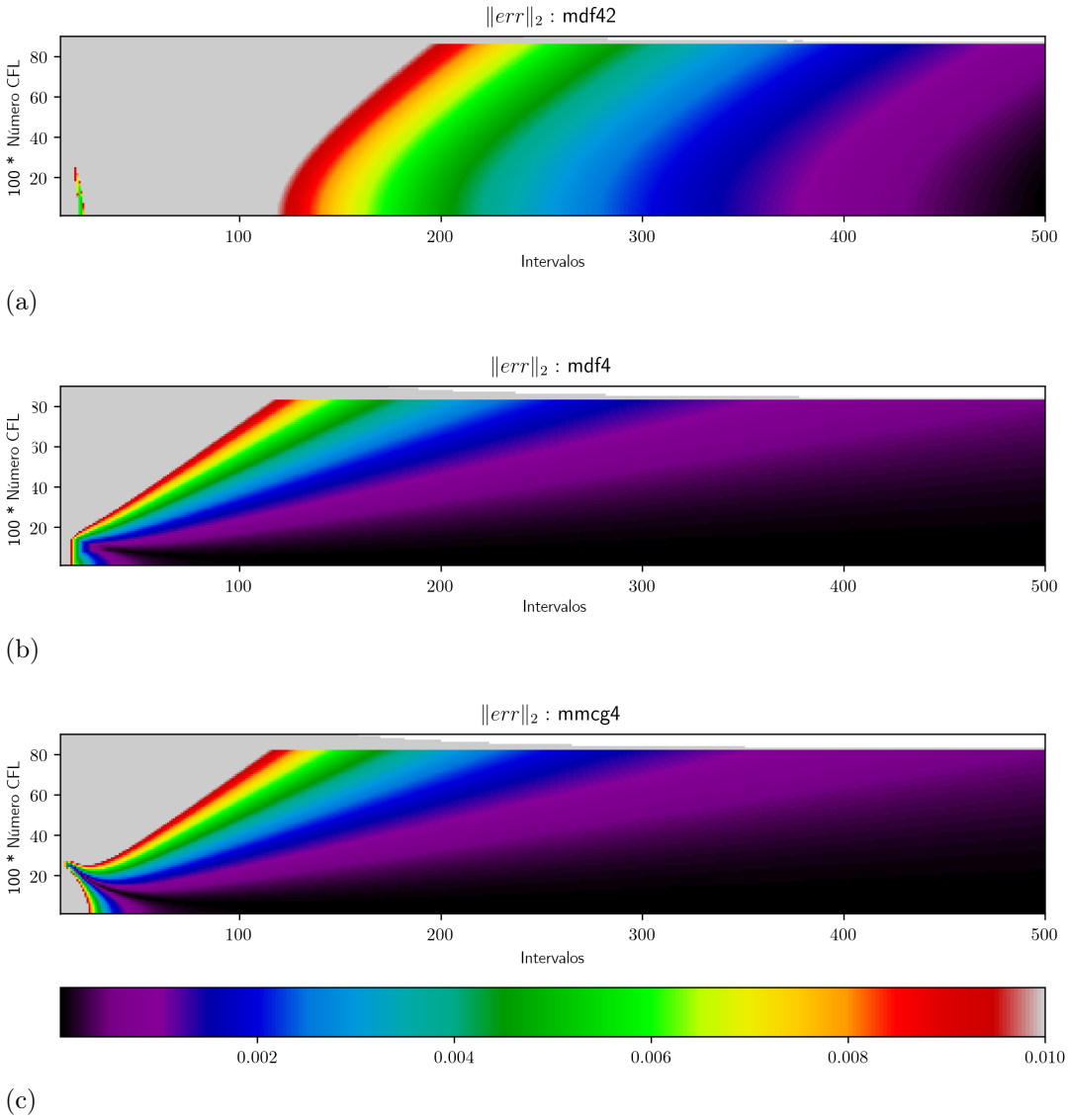
Algoritmo 11 - Computa o erro máximo.

ALGORITMO ERRO N $\times$ CFL

1. **para** n **de** 1 **até** 500 , **fazer**
2. | **para** c **de** 0.01 **até** 0.90 **passo** Número CFL variando em centésimos, **fazer**
3. | | **para** t **de** 0 **até** 4 **passo** Em segundos, **fazer**
4. | | | $\epsilon^t_{(n,c)} \leftarrow \left|\hat{U}^t_{(n,c)} - U^t_{(n,c)}\right|$
5. | | | $\|err\|^t_{(n,c)} \leftarrow \|\epsilon^t_{(n,c)}\|_2$
6. | | **fim para**
7. | | $\|err\|^{max}_{(n,c)} \leftarrow max\left(\|err\|^t_{(n,c)}\right) \forall t \in [0,4]$
8. | **fim para**
9. **fim para**

FIM ALGORITMO

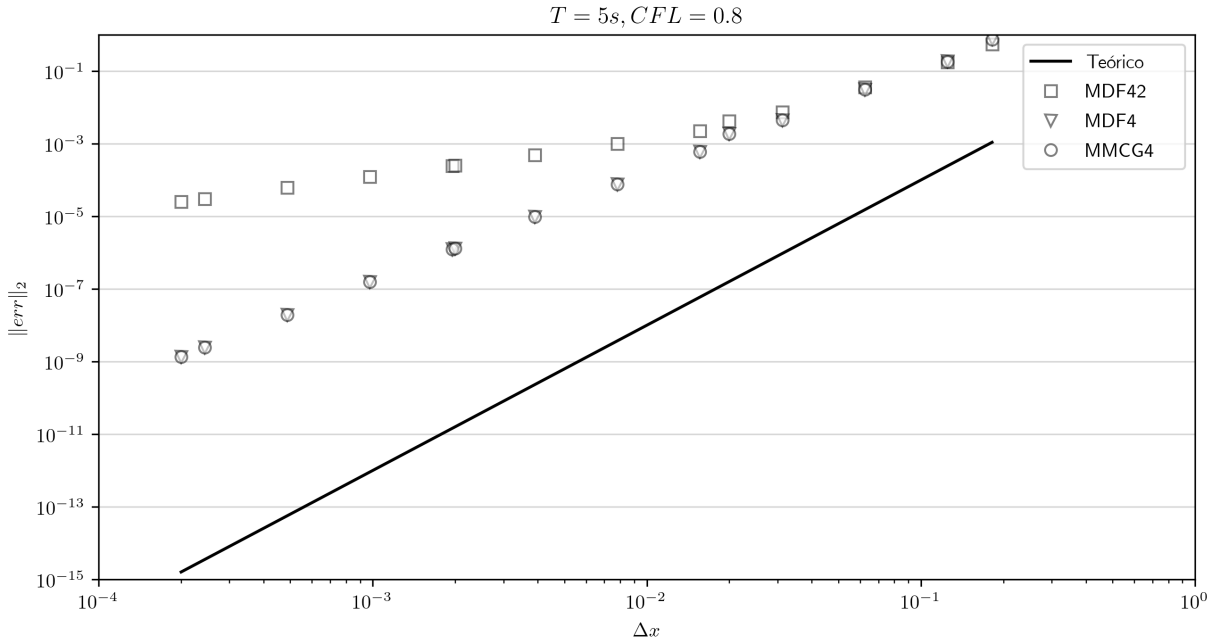
Figura 14 - Visualisação de erro e estabilidade.



Legenda: Nas figuras o eixo horizontal representa a variação do número de intervalos discretizados na malha intercalada com $N = [11, 500]$, onde 11 é o menor valor possível para o modelo mimético de quarta ordem, o eixo vertical apresenta a variação em centésimos do número CFL que varia no intervalo $CFL = [0.01, 0.90]$. As cores representam o erro da norma L_2 (109) para cada par (N, CFL). Os valores máximos e mínimos são os mesmos em todas figuras e podem ser comparados pela barra de cores, onde preto são os locais de menor erro e cinza os de erro acima da faixa aceitável. Os pontos em brancos são pares (N, CFL) onde a estabilidade não foi obtida e os valores da simulação foram rápidamente para infinito. O número CFL máximo para o modelo MDF42 foi 0.85, para o MDF4 0.82 e para o modelo MMCG4 foi 0.81.

Fonte: O autor, 2019.

Figura 15 - Análise de convergência a priori dos modelos numéricos.



Legenda: A figura mostra a análise de convergência a priori apresentado em escala logarítmica em ambos os eixos. O eixo horizontal mostra os diferentes valores de Δx e o eixo vertical mostra o erro obtido pela norma L_2 do erro de truncamento (conforme equação (109)). As marcações círculares são do modelo MMCG4, as triangulares o modelo MDF4 e as marcações quadradas do modelo MDF42. Uma linha da convergência teórica para a aproximação de quarta ordem é exibida para fins de comparação. O erro é similar em ambos os modelos para discretizações grosseiras e vai apresentando melhor convergência no modelo MMCG conforme a malha se refina. O coeficiente angular do modelo MMCG4 é 2.9380, do modelo MDF4 2.9333 é enquanto do modelo MDF42 é 1.9379. A análise foi realizada para o número CFL 0.80 no instante $t = 5$.

Fonte: O autor, 2019.

Tabela 2 - Coeficientes de convergência.

		Coef. Convergência		
CFL	Tempo	MDF42	MDF4	MMCG4
0.4	1	1.01483	2.95336	3.04790
	5	1.10736	2.88369	2.93288
	10	1.75331	2.75412	2.78644
	20	1.81692	2.74627	2.75216
0.7	1	1.11439	2.95967	2.97655
	5	1.33508	2.75810	2.81868
	10	1.93790	2.69276	2.79715
	20	1.64006	2.79632	2.83537
0.8	1	1.01483	2.93999	2.95234
	5	1.10736	2.93304	2.93800
	10	1.75331	2.72820	2.74980
	20	1.81692	2.78846	2.79232

Legenda: Apresenta as taxas de convergência dos modelos MDF42, MDF4 e MMCG4 para diferentes condições CFL e em diferentes instantes de tempo.

Fonte: O autor, 2019.

CONCLUSÃO

Este trabalho permitiu a compreensão dos conceitos que envolvem a construção de um operador de diferenças finitas mimético de acordo com o Método Mimético de Castillo-Grone cujo as principais vantagens com relação ao método de diferenças finitas tradicional são: a. manter as propriedades matemáticas dos operadores diferenciais analíticos (condições miméticas)(Seção 1.1); e b. preservar a acurácia do operador discreto em todo domínio (Seção 1.3). Além disso, apresentou algoritmos (Capítulo 2) independente de linguagem para a implementação computacional (Apêndice C) do método, o que permitiu adicionar aos apêndices uma implementação de referência utilizando a linguagem Python.

Assim, foi possível criar uma simulação (Seção 3.1) para validar os operadores obtidos bem como os algoritmos elaborados. Simulação esta que foi comparada com dois outros operadores de diferenças finitas convencional com a mesma ordem de acurácia, este teste mostrou que o MMCG não só responde de forma esperada como obtém resultados com menor erro global do que o MDF clássico.

Foi analisado o campo de pressão entre o MMCG (Seção 3.3) e duas versões distintas do MDF (Seção 3.6) uma utilizando quarta ordem nos pontos interiores e segunda ordem no contorno (MDF42) (eqs. (105) (106)), a outra utilizando quarta ordem central nos pontos interiores e quatro pontos em diferença progressiva ou regressiva nos pontos de contorno (MDF4) (eqs. (107) (108)), também foi calculado o campo analítico como referência e para cálculo do erro de truncamento (eq. (109)). Para fins de apresentação mostramos o campo de pressão em diferentes instantes temporais (Figuras 6, 8, 10 e 12), bem como o erro de truncamento (Figuras 7, 9, 11 e 13), em todos os casos o MMCG obteve melhores resultados.

Os erros globais máximos (Tabela 1) foram estudados em um intervalo para diferentes discretizações espaciais e temporais. Em quase todos os casos o MMCG obteve melhores resultados, inclusive existem casos onde o MMCG obtém menor erro mesmo com uma discretização temporal mais grossa.

Foi feito uma mapa (Figura 14) comparando o comportamento da convergência do modelo ao variar a malha espacial e o número CFL, as convergências do MMCG são similares (levemente superiores) as mostradas pelo MDF4 e largamente superiores ao MDF42. Exceto onde o número CFL é muito baixo para uma malha bastante grossa (Figura 14c). Um estudo mais aprofundado deste comportamento não foi feito e é um ponto interessante a ser tomado.

Analisamos também os valores da convergência a priori (Figura 15) do modelo, o MMCG apresenta o melhor resultado, seguido do MDF4 e do MDF42.

Os resultados foram consistentes (Tabela 2) com os já obtidos em outros artigos,

assim, validando nosso estudo e permitindo um aprofundamento no campo de métodos miméticos.

O problema modelo (Seção 3.1) utilizado é extremamente simples, sua escolha foi dada principalmente por atender ao objetivo principal do trabalho que é estudar e consolidar o processo de elaboração dos operadores miméticos de Castillo-Grone.

Para finalizar foi demonstrado com este trabalho que o MMCG é uma alternativa possível aos operadores convencionais de diferenças finitas, sem elevar o custo computacional da simulação.

TRABALHOS FUTUROS

Explorar mais profundamente os operadores miméticos tanto modelos 2D e 3D é o próximo passo natural como trabalhado em Castillo e Miranda (2013), Ferrer et al. (2014) e Sanchez, Paolini e Castillo (2014), conjuntamente trabalhar com outros operadores como rotacional e laplaciano aplicados a outros problemas de EDP.

Existem alguns pontos em aberto que podem ser trabalhados em uma tese de doutorado. Encontrar melhores paramêtros para a matriz de contorno que tenham a convergência ótima, será que existe a possibilidade de encontrar uma relação entre os paramêtros de borda e esta matriz ótima, Runyan (2011) especula que a matriz ótima para A deve ser diferente para cada situação, conforme o problema sendo modelado.

Otimizar o algoritmo sequencial para o cálculo do operador e buscar solução paralelas para solucionar um operador mimético não foi explorado (SANCHEZ; CASTILLO, 2013). Acredito que os algorítmos clássicos não possam ser imediatamente aplicados visto que nos pontos de contorno o operador não apresenta característica tridiagonal, além da matriz do operador ser quase Toeplitz.

Avaliar as matrizes do operador resultantes de uma solução puramente numérica, os resultados obtidos para construção dos operadores de borda contam com suporte de ferramentas de álgebra computacional, analisar a o espaço da solução de mínimos quadrados com as soluções α do hiperplano como mostrado em (84).

REFERÊNCIAS

ABOUALI, Mohammad; CASTILLO, Jose E. *High-Order Compact Castillo-Groneś Mimetic Operators*. College of Sciences, San Diego, 2012.

BATHE, Klaus-Jurgen; WILSON, Edward L. *Numerical methods in finite element*. [S.l.]: Prentice-Hall,, 1976. (Prentice-Hall civil engineering and engineering mechanics series analysis). ISBN 0136271901.

BATISTA, ED; CASTILLO, Jose E. Mimetic schemes on non-uniform structured meshes. *Electronic Transactions on Numerical Analysis*, v. 34, n. 1, p. 152–162, 2009.

BAZAN, Carlos et al. Mimetic finite difference methods in image processing. *Computational & Applied Mathematics*, SciELO Brasil, v. 30, n. 3, p. 701–720, 2011.

BIRDSALL, C. K.; LANGDON, A. B. *Plasma Physics via Computer Simulations*. [S.l.]: McGraw-Hill Book Company, 1985.

BREZZI, Franco; LIPNIKOV, Konstantin; SHASHKOV, Mikhail. Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM*, 2005.

CASTILLO; GRONE. A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law. SIAM, 2003.

CASTILLO, J. E.; GRONE, R. D. Using kronecker products to construct mimetic gradients. 2017. Disponível em: <http://dx.doi.org/10.1080/03081087.2017.1298564>.

CASTILLO, Jose E. et al. Fourth and sith order conservative finite difference approximations of the divergence and gradient. Elsevier, 2000.

CASTILLO, J. E. et al. High-order mimetic finite difference methods on non uniform grids. *Houston Journal of Mathematics*, 1995.

CASTILLO, José E; MIRANDA, Guillermo F. *Mimetic discretization methods*. [S.l.]: CRC Press, 2013.

CONTE, S. D.; BOOR, Carl de. *Elementary Numerical Analysis*. [S.l.]: Springer, 1980.

CORDOVA, Luis J. et al. Compact finite difference modeling of 2-d acoustic wave propagation. 2014.

FERRER, Miguel et al. 3d viscoelastic anisotropic seismic modeling with high-order mimetic finite differences. Springer, 2014.

GATHEN, Jürgen Gerhard Joachim von zur. *Modern Computer Algebra*. 3rd. ed. [S.l.]: Cambridge University Press, 2013. ISBN 1107039037,9781107039032.

GEDDES STEPHEN R. CZAPOR, George Labahn Keith O. *Algorithms for computer algebra*. 1. ed. [S.l.]: Kluwer Academic, 1992. ISBN 0792392590,9780792392590,9780585332475.

GREGORY, R. T. A method for defining numerical differentiation formulas. Mathematical Association of America, 1957.

HOFFMAN, Kenneth; KUNZE, Ray. *Algebra Linear*. [S.l.]: LTC, 1979.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

HYMAN, J.M.; LARROUTUROU, B. The numerical differentiation of discrete functions using polynomial interpolation methods. Appl. Math. and Compo 10/11, 487-506, 1992.

IORIO, Valeria. *EDP, um curso de graduação*. [S.l.]: IMPA, 2010.

JONES, Eric et al. *SciPy: Open source scientific tools for Python*. 2001–. [Online; accessed <today>]. Disponível em: <http://www.scipy.org/>.

KOIRAN, Pascal; PORTIER, Natacha; VILLARD, Gilles. A rank theorem for vandermonde matrices. Elsevier, 2004.

LEVEQUE, Randall J. *Finite Volume Methods for Hyperbolic Problems*. 1. ed. [S.l.]: Cambridge University Press, 2002. (Cambridge texts in applied mathematics). ISBN 0521810876,0521009243,9780521810876,9780511042195.

LEWY, H.; FRIEDRICHS, K.; COURANT, R. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, v. 100, p. 32–74, 1928. Disponível em: <http://eudml.org/doc/159283>.

LIMA, Elon. *Curso de Análise volume 1*. [S.l.]: IMPA, 2012.

_____. *Curso de Análise volume 2*. [S.l.]: IMPA, 2012.

LIMA, Elon Lages. *Algebra Linear*. [S.l.]: IMPA, 2011.

MACON, N.; SPITZBART, A. Inverses of vandermonde matrices. Mathematical Association of America, 1958.

MEURER, Aaron et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, v. 3, p. e103, jan. 2017. ISSN 2376-5992. Disponível em: <https://doi.org/10.7717/peerj-cs.103>.

MORGADO, Igor. *PyMTK: A mimetic method library for Python*. 2018. Disponível em: <https://github.com/igormorgado/pymtk/>.

_____. *Software para simulação de ondas usando MMCG*. 2018. Disponível em: <https://github.com/igormorgado/mimeticos>.

OLIPHANT, Travis E. *A guide to NumPy*. [S.l.]: Trelgol Publishing, 2006.

REYNOLDS, Albert C. Boundary conditions for the numerical solution of wave propagation problems. *Geophysics*, v. 43, p. 1099–1110, 1978. Disponível em: <http://dx.doi.org/10.1190/1.1440881>.

ROJA, Otilio et al. Modelling rupture propagation using high-order mimetic finite differences. Geophysics Journal, 2007.

ROJAS, Otilio et al. Finite difference modeling of rupture propagation with strong velocity-weakening friction. Geophysics Journal, 2009.

ROJAS, O. et al. Low dispersive modeling of rayleigh waves on partly staggered grids. Springer, 2013.

ROSSUM, Guido van. *Python tutorial*. Amsterdam, 1995.

RUNYAN, Justin Brent. *A novel higher order finite difference time domain method based on the Castillo-Grone mimetic curl operator with applications concerning the time-dependent Maxwell equations*. 2011. 68 f. Dissertação (Master of Science in Computational Science) — San Diego State University, California, 2011.

SANCHEZ, Eduardo; CASTILLO, Jose E. *An algorithmic study of the construction of Higher-order one-dimensional Castillo-Grone mimetic gradient and divergence operators*. College of Sciences, San Diego, 2013.

SANCHEZ, Eduardo J.; PAOLINI, Christopher P.; CASTILLO, Jose E. Analyzing diffusive-advective-reactive processes using mimetic finite differences with implications in carbon dioxide geologic storage. 2014.

SHASHKOV, Mikhail. *Conservative Finite-Difference Methods on General Grids*. [S.l.]: CRC Press, 1996.

SPITZBART, A.; MACON, N. Numerical differentiation formulas. Mathematical Association of America, 1957.

TREFETHEN, David Bau III Lloyd N. *Numerical linear algebra*. [S.l.]: Society for Industrial and Applied Mathematics, 1997. ISBN 9780898713619,0898713617.

TREFETHEN, LLoyd N. *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*. [S.l.]: Cornell University, 1996.

VERDE-STAR, Luis. Inverses of generalized vandermonde matrices. *Journal of Mathematical Analysis and Applications*, 1988.

YEE, Kane S. Numerical solution of initial boundary value problems involving maxwell equations in isotropic media. *IEEE Transactions on Antennas and propagation*, v. 14, n. 3, p. 302–307, 1966.

APÊNDICE A – Conceitos

A.1 Matrizes de pesos e produtos internos ponderados

Em (LIMA, 2011) é definido o *produto interno* como um funcional bilinear, simétrico e positivo em um espaço vetorial que associa um par de vetores u e v a um número real[13]. No espaço euclidiano a forma canônica do produto interno é dada por

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{n} u_i \cdot v_i \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \tag{110}$$

outra forma de escrever a mesma operação é dado pelo produto matricial

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^T \mathbf{u} = \mathbf{v}^T \mathbb{1}_N \mathbf{v} = \langle \mathbb{1}_N \mathbf{u}, \mathbf{v} \rangle \tag{111}$$

Onde $\mathbb{1}_N$ é a matriz onde todos seus elementos da diagonal são 1. Este é também chamado como produto interno canônico.

Uma generalização do produto interno é possível ao utilizarmos, uma matriz Q diagonal positiva definida. Usaremos a notação $\langle u, v \rangle_Q$, para expressar o produto interno utilizando a matriz Q no produto, esta matriz será chamada de *matriz de pesos* e o produto interno derivado de uma matriz de pesos será chamado de *produto interno ponderado*, dado por

$$\langle \mathbf{u}, \mathbf{v} \rangle_Q = \mathbf{v}^T Q \mathbf{u} \tag{112}$$

ou de forma matricial

$$\begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix} \begin{bmatrix} q_1 & & & 0 \\ & q_2 & & \\ & & \ddots & \\ 0 & & & q_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \sum_{i=1}^{n} q_i \cdot v_i \cdot u_i = k \tag{113}$$

com $Q = diag(q_1, q_2, \dots, q_n)$ e $k \in \mathbb{R}$.

Para que esta operação seja considerada um produto interno é necessário que a matriz Q seja positiva definida. Por definição a matriz Q é diagonal e todos seus elementos são positivos não nulos o que faz da matriz Q ser positiva definida por construção.

No caso particular dos métodos miméticos, os produtos internos ponderados são

[13] Todas propriedades do produto interno podem ser vistas em Lima (2011) e Hoffman e Kunze (1979).

necessários para encontrar soluções aos sistemas lineares que normalmente não possui solução no espaços euclidianos de base canônica.

A.2 Matrizes de Vandermonde

Dados n valores distintos dados por $\mathbf{g} = (g_1, g_2, \dots g_n)$, que chamaremos de *vetor gerador*. A matriz gerada associando cada i-ésima linha desta matriz a uma potência positiva dos elementos do vetor gerador $\mathbf{g}$, é chamada de *matriz de Vandermonde* e notaremos como $V(m; \mathbf{g})$ sendo m chamado de *ordem* da matriz, que limita o máximo expoente. Assim sendo a matriz $V(m; \mathbf{g})$ tem $m+1$ linhas e n colunas. De forma mais clara a matriz é escrita como

$$V(m;\mathbf{g}) = \begin{bmatrix} g_1^0 & g_2^0 & \dots & g_n^0 \\ g_1^1 & g_2^1 & \dots & g_n^1 \\ g_1^2 & g_2^2 & \dots & g_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ g_1^m & g_2^m & \dots & g_n^m \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ g_1 & g_2 & \dots & g_n \\ g_1^2 & g_2^2 & \dots & g_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ g_1^m & g_2^m & \dots & g_n^m \end{bmatrix} \tag{114}$$

Quando $n = m+1$ a matriz de Vandermonde é quadrada e é inversível, a demonstração pode ser encontrada em Macon e Spitzbart (1958) e Verde-Star (1988).

As matrizes de Vandermonde tem uso prático para a interpolação polinomial onde g_i, são os pontos sendo interpolados, e para a formulação das expansões de Taylor e diferenciação, este uso foi demonstrado profundamente em Hyman e Larrouturou (1992), Gregory (1957) e Spitzbart e Macon (1957).

Esta formulação bem como suas aplicações tem uso fundamental dentro da geração dos operadores miméticos pois permite uma formulação matricial para as aproximações da derivada.

A.2.1 Núcleo das matrizes de Vandermonde

Demonstraremos aqui a dimensão do núcleo das matrizes de Vandermonde pois seu resultado é necessário para a elaboração do operador miméticos. Em geral, o núcleo da matriz de Vandermonde de ordem k tem dimensão $\frac{1}{2}k - 1$ pois consideremos $V(m, G) \in M_{m+1,n}$ onde assumimos que $n \geq m+1$.

Assim, $V(m, G) : \mathbb{R}^n \to \mathbb{R}^{m+1}$. As primeiras $m+1$ colunas de V são a transposta da matriz de Vandermonde clássica (chamaremos de W) que é inversível.

Desta forma $posto(V) = m+1$ (V é sobrejetiva) e $dim(\ker(V)) = n-m-1$. Como $n = \frac{3m}{2}$ temos que $dim(\ker(V)) = m/2 - 1$.

Seja então V_i uma família de matrizes de Vandermonde, geradas por g_i igualmente espaçados e $\mathbf{b}$ um vetor resultante qualquer, temos então

$$V_i a_i^T = \mathbf{b} \tag{115}$$

cujo a solução a_i é dada por

$$a_i = a_0 + \alpha_i \ker(V_i) \tag{116}$$

Pois como V é sobrejetiva existe uma solução particular que chamaremos de a_0. Se a_i é uma solução, então

$$\begin{aligned} V(a_0) = V(a_i) &\iff \\ V(a_i - a_0) = 0 &\iff \\ a_i - a_0 \in \ker(V) &\iff \\ a_i \in a_0 + \ker(V) &\implies \\ \exists \alpha_i \in \mathbb{R} : a_i = a_0 + \alpha_i \ker(V) & \end{aligned} \tag{117}$$

Como queríamos demonstrar.[14]

A.3 Interpoladores de Lagrange

Um polinômio interpolante de Lagrange pode ser diferenciado para dar uma aproximação de alta ordem precisa para a derivada de uma função definida em uma grade não uniforme.(HYMAN; LARROUTUROU, 1992)

Em uma grade não uniforme, as aproximações das diferenças de **grad** e **div** geradas pela interpolação de Lagrange são raramente miméticas. Se por outro lado a função e a malha são primeiro mapeadas para uma malha uniforme, as derivadas são aproximadas no local usando a interpolação de Lagrange e estes resultados são então mapeados de volta para a grade não uniforme, resultando em aproximações de diferenças finitas que podem ser demonstradamente miméticas visto que cada uma das etapas dos processos do relacionamento de simetria são preservados.

O erro estimado para um interpolante de Lagrange de ordem n(usando $n+1$ pontos)

[14] Uma demonstração mais extensa e detalhada pode ser vista em Koiran, Portier e Villard (2004).

para uma função suave f é:

$$|f(x) - L(x)| \leq \frac{max_{\tilde{x}}\left(f^{n+1}\left(\tilde{x}\right)\right)}{(n+1)!} R\left(\Delta x\right)^{n+1}, \tag{118}$$

$$\frac{df}{dx}(x) = \frac{dL_n}{dx}(x) + O\left(\Delta x\right)^n \tag{119}$$

onde $\tilde{x}$ é o ponto no intervalo de interpolação e R é alguma constante que depende dos pontos de interpolação e cresce na ordem h^3. Assim L_3 nos da a aproximação de terceira ordem para a primeira derivada em grades não uniformes. (em grades regulares isto torna-se uma aproximação de quarta ordem nos pontos médios da célula pelo erro de cancelamento).

A.3.1 Construção do interpolador de Lagrangre

Sejam os pontos x_0, x_1, x_2, e x_3 quatro pontos na grade e $f_i = f(x_i)$, então:

$$L_3(x) = D_0 + (x - x_0)D_{01} + (x - x_0)(x - x_1)D_{012} + (x - x_0)(x - x_1)(x - x_2)D_{0123} \tag{120}$$

onde

$$\begin{aligned} D_0 &= f_0 \\ D_{01} &= \frac{f_1 - f_0}{x_1 - x_0} \\ D_{012} &= \frac{D_{12} - D_{01}}{x_2 - x_0} \\ D_{0123} &= \frac{D_{123} - D_{012}}{x_3 - x_0} \end{aligned} \tag{121}$$

são as diferenças divididas. A primera derivada do interpolante é

$$\begin{aligned} \frac{dL_3(x)}{dx} &= D_{01} \\ &+ \left[(x - x_0) + (x - x_1)\right] D_{012} \\ &+ \left[(x - x_0)(x - x_1) + (x - x_0)(x - x_2) + (x - x_1)(x - x_2)\right] D_{0123}. \end{aligned} \tag{122}$$

Em uma grade uniforme, com $x_0 = x_{i-1}, x_1 = x_i, x_2 = x_{i+1}, x_3 = x_{i+2}$ e $x = x_{i+\frac{1}{2}}$, isto se torna:

$$(D_x f)_{i+\frac{1}{2}} = \frac{-f_{i+2} + 27f_{i+1} - 27f_i + f_{i-1}}{24\Delta x}. \tag{123}$$

A.4 Expansões de Taylor

Demonstraremos nesta sessão o uso das matrizes de Vandermonde para obtenção dos coeficientes de uma expansão de Taylor. Tomemos então a expansão de Taylor que define uma aproximação boa suficiente para um ponto x próximo de a como:

$$f(x) = f(a) + (x-a)\frac{f^{(1)}}{1!} + (x-a)^2\frac{f^{(2)}}{2!} + (x-a)^3\frac{f^{(3)}}{3!} + (x-a)^4\frac{f^{(4)}}{4!} + O\left((x-a)^5\right). \quad (124)$$

Queremos então encontrar o valor para um ponto proximo de x dado a distância $x - ch$. Substituindo a por $x - ch$ na expansão de Taylor temos:

$$\begin{aligned} f(x-ch) &= f(x) + (x-(x-ch))\frac{f^{(1)}}{1!} + (x-(x-ch))^2\frac{f^{(2)}}{2!} + \\ &\quad + (x-(x-ch))^3\frac{f^{(3)}}{3!} + (x-(x-ch))^4\frac{f^{(4)}}{4!} + O\left((x-(x-ch))^5\right) \\ f(x-ch) &= f(x) + (ch)\frac{f^{(1)}}{1!} + (ch)^2\frac{f^{(2)}}{2!} + (ch)^3\frac{f^{(3)}}{3!} + (ch)^4\frac{f^{(4)}}{4!} + O\left(h^5\right). \end{aligned} \quad (125)$$

Podemos então substituir c pelos pontos igualmente espaçados $\pm\frac{1}{2}$ e $\pm\frac{3}{2}$. Nos dando a igualdade (aqui colocaremos somente o caso $c = \frac{1}{2}$).

$$f\left(x - \frac{1}{2}h\right) = f(x) + \left(\frac{1}{2}h\right)\frac{f^{(1)}}{1!} + \left(\frac{1}{2}h\right)^2\frac{f^{(2)}}{2!} + \left(\frac{1}{2}h\right)^3\frac{f^{(3)}}{3!} + \left(\frac{1}{2}h\right)^4\frac{f^{(4)}}{4!} + O\left(h^5\right). \quad (126)$$

Podemos então substituir os termos da matriz

$$f'(x) + O(h^4) = \sigma \begin{bmatrix} f\left(x - \frac{3}{2}h\right) \\ f\left(x - \frac{1}{2}h\right) \\ f\left(x + \frac{1}{2}h\right) \\ f\left(x + \frac{3}{2}h\right) \end{bmatrix} \quad (127)$$

por suas aproximações por expansões de Taylor resultando em

$$\begin{bmatrix} f\left(x-\frac{3}{2}h\right) \\ f\left(x-\frac{1}{2}h\right) \\ f\left(x+\frac{1}{2}h\right) \\ f\left(x+\frac{3}{2}h\right) \end{bmatrix} = \begin{bmatrix} f(x) - \left(\frac{3}{2}h\right)\frac{f^{(1)}}{1!} + \left(\frac{3}{2}h\right)^2\frac{f^{(2)}}{2!} - \left(\frac{3}{2}h\right)^3\frac{f^{(3)}}{3!} + \left(\frac{3}{2}h\right)^4\frac{f^{(4)}}{4!} + O\left(h^5\right) \\ f(x) - \left(\frac{1}{2}h\right)\frac{f^{(1)}}{1!} + \left(\frac{1}{2}h\right)^2\frac{f^{(2)}}{2!} - \left(\frac{1}{2}h\right)^3\frac{f^{(3)}}{3!} + \left(\frac{1}{2}h\right)^4\frac{f^{(4)}}{4!} + O\left(h^5\right) \\ f(x) + \left(\frac{1}{2}h\right)\frac{f^{(1)}}{1!} + \left(\frac{1}{2}h\right)^2\frac{f^{(2)}}{2!} + \left(\frac{1}{2}h\right)^3\frac{f^{(3)}}{3!} + \left(\frac{1}{2}h\right)^4\frac{f^{(4)}}{4!} + O\left(h^5\right) \\ f(x) + \left(\frac{3}{2}h\right)\frac{f^{(1)}}{1!} + \left(\frac{3}{2}h\right)^2\frac{f^{(2)}}{2!} + \left(\frac{3}{2}h\right)^3\frac{f^{(3)}}{3!} + \left(\frac{3}{2}h\right)^4\frac{f^{(4)}}{4!} + O\left(h^5\right) \end{bmatrix}$$

$$\begin{bmatrix} f\left(x-\frac{3}{2}h\right) \\ f\left(x-\frac{1}{2}h\right) \\ f\left(x+\frac{1}{2}h\right) \\ f\left(x+\frac{3}{2}h\right) \end{bmatrix} = \begin{bmatrix} f(x) - \left(\frac{3}{2}h\right)\frac{f^{(1)}(x)}{1!} + \left(\frac{3}{2}h\right)^2\frac{f^{(2)}(x)}{2!} - \left(\frac{3}{2}h\right)^3\frac{f^{(3)}(x)}{3!} + \left(\frac{3}{2}h\right)^4\frac{f^{(4)}(x)}{4!} \\ f(x) - \left(\frac{1}{2}h\right)\frac{f^{(1)}(x)}{1!} + \left(\frac{1}{2}h\right)^2\frac{f^{(2)}(x)}{2!} - \left(\frac{1}{2}h\right)^3\frac{f^{(3)}(x)}{3!} + \left(\frac{1}{2}h\right)^4\frac{f^{(4)}(x)}{4!} \\ f(x) + \left(\frac{1}{2}h\right)\frac{f^{(1)}(x)}{1!} + \left(\frac{1}{2}h\right)^2\frac{f^{(2)}(x)}{2!} + \left(\frac{1}{2}h\right)^3\frac{f^{(3)}(x)}{3!} + \left(\frac{1}{2}h\right)^4\frac{f^{(4)}(x)}{4!} \\ f(x) + \left(\frac{3}{2}h\right)\frac{f^{(1)}(x)}{1!} + \left(\frac{3}{2}h\right)^2\frac{f^{(2)}(x)}{2!} + \left(\frac{3}{2}h\right)^3\frac{f^{(3)}(x)}{3!} + \left(\frac{3}{2}h\right)^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} + \begin{bmatrix} O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \end{bmatrix} \tag{128}$$

Separando em um produto matricial temos

$$\begin{bmatrix} f\left(x-\frac{3}{2}h\right) \\ f\left(x-\frac{1}{2}h\right) \\ f\left(x+\frac{1}{2}h\right) \\ f\left(x+\frac{3}{2}h\right) \end{bmatrix} = \begin{bmatrix} 1 & \left(-\frac{3}{2}\right) & \left(-\frac{3}{2}\right)^2 & \left(-\frac{3}{2}\right)^3 & \left(-\frac{3}{2}\right)^4 \\ 1 & \left(-\frac{1}{2}\right) & \left(-\frac{1}{2}\right)^2 & \left(-\frac{1}{2}\right)^3 & \left(-\frac{1}{2}\right)^4 \\ 1 & \left(\frac{1}{2}\right) & \left(\frac{1}{2}\right)^2 & \left(\frac{1}{2}\right)^3 & \left(\frac{1}{2}\right)^4 \\ 1 & \left(\frac{3}{2}\right) & \left(\frac{3}{2}\right)^2 & \left(\frac{3}{2}\right)^3 & \left(\frac{3}{2}\right)^4 \end{bmatrix} \begin{bmatrix} f(x) \\ h\frac{f^{(1)}(x)}{1!} \\ h^2\frac{f^{(2)}(x)}{2!} \\ h^3\frac{f^{(3)}(x)}{3!} \\ h^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} + \begin{bmatrix} O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \end{bmatrix}, \tag{129}$$

que pode ser substituída pela transposta da matriz de *Vandermonde*

$$\begin{bmatrix} f\left(x-\frac{3}{2}h\right) \\ f\left(x-\frac{1}{2}h\right) \\ f\left(x+\frac{1}{2}h\right) \\ f\left(x+\frac{3}{2}h\right) \end{bmatrix} = V^T \begin{bmatrix} f(x) \\ hf^{(1)}(x) \\ h^2\frac{f^{(2)}(x)}{2!} \\ h^3\frac{f^{(3)}(x)}{3!} \\ h^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} + \begin{bmatrix} O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \end{bmatrix}, \tag{130}$$

como

$$\sigma = \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 \end{bmatrix} \tag{131}$$

podemos então multiplicar (130) por σ à esquerda e obter

$$\sigma \begin{bmatrix} f\left(x-\frac{3}{2}h\right) \\ f\left(x-\frac{1}{2}h\right) \\ f\left(x+\frac{1}{2}h\right) \\ f\left(x+\frac{3}{2}h\right) \end{bmatrix} = \sigma V^T \begin{bmatrix} f(x) \\ hf^{(1)}(x) \\ h^2\frac{f^{(2)}(x)}{2!} \\ h^3\frac{f^{(3)}(x)}{3!} \\ h^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} + \sigma \begin{bmatrix} O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \end{bmatrix}. \tag{132}$$

Substituindo (127) em (132) temos

$$f'(x) + O(h^4) = \sigma V^T \begin{bmatrix} f(x) \\ hf'(x) \\ h^2\frac{f^{(2)}(x)}{2!} \\ h^3\frac{f^{(3)}(x)}{3!} \\ h^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} + \sigma \begin{bmatrix} O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \end{bmatrix} \tag{133}$$

e como

$$f'(x) = \frac{hf'(x)}{h}, \tag{134}$$

então para aproximar $f'(x)$ pela combinação linear

$$f'(x) + O(h^4) = \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 \end{bmatrix} \begin{bmatrix} f\left(x-\frac{3}{2}h\right) \\ f\left(x-\frac{1}{2}h\right) \\ f\left(x+\frac{1}{2}h\right) \\ f\left(x+\frac{3}{2}h\right) \end{bmatrix} \tag{135}$$

precisamos que

$$\sigma V^T = \begin{bmatrix} 0 & \frac{1}{h} & 0 & 0 & 0 \end{bmatrix}. \tag{136}$$

De forma que

$$\sigma V^T \begin{bmatrix} f(x) \\ hf'(x) \\ h^2\frac{f^{(2)}(x)}{2!} \\ h^3\frac{f^{(3)}(x)}{3!} \\ h^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{h} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f(x) \\ hf'(x) \\ h^2\frac{f^{(2)}(x)}{2!} \\ h^3\frac{f^{(3)}(x)}{3!} \\ h^4\frac{f^{(4)}(x)}{4!} \end{bmatrix} = \frac{hf'(x)}{h} = f'(x), \tag{137}$$

devemos manter ambos os lados da equação (133) na mesma ordem de magnitude de h, para tal tornamos σ proporcional a $\frac{1}{h}$ e criamos para tal a matriz $s = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \end{bmatrix}$,

independente de h tal que $\sigma_i = \frac{s_i}{h}$. Assim a igualdade

$$\sigma \begin{bmatrix} O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \\ O\left(h^5\right) \end{bmatrix} = s \begin{bmatrix} O\left(h^4\right) \\ O\left(h^4\right) \\ O\left(h^4\right) \\ O\left(h^4\right) \end{bmatrix} \tag{138}$$

é verdadeira da mesma forma que para a primeira derivada temos

$$\sigma V^T = \begin{bmatrix} 0 & \frac{1}{h} & 0 & 0 & 0 \end{bmatrix} \tag{139}$$

$$\frac{1}{h} s V^T = \begin{bmatrix} 0 & \frac{1}{h} & 0 & 0 & 0 \end{bmatrix} \tag{140}$$

$$V s^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T . \tag{141}$$

Substituindo nos dá o sistema

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ \left(-\frac{3}{2}\right) & \left(-\frac{1}{2}\right) & \left(\frac{1}{2}\right) & \left(\frac{3}{2}\right) \\ \left(-\frac{3}{2}\right)^2 & \left(-\frac{1}{2}\right)^2 & \left(\frac{1}{2}\right)^2 & \left(\frac{3}{2}\right)^2 \\ \left(-\frac{3}{2}\right)^3 & \left(-\frac{1}{2}\right)^3 & \left(\frac{1}{2}\right)^3 & \left(\frac{3}{2}\right)^3 \\ \left(-\frac{3}{2}\right)^4 & \left(-\frac{1}{2}\right)^4 & \left(\frac{1}{2}\right)^4 & \left(\frac{3}{2}\right)^4 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} , \tag{142}$$

cujo a solução é

$$s = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} \end{bmatrix} \tag{143}$$

que é o estêncil de quarta ordem das diferenças finitas centrais.

APÊNDICE B – Demonstrações

B.1 Produtos internos discretos e integrais contínuas

Seja f uma função escalar e $\mathbf{v}$ uma função vetoria. Podemos escrever seu produto como:

$$f\mathbf{v} = \mathbf{F}. \tag{144}$$

Aplicando a derivada ao lado esquerdo, pela regra da cadeia podemos obter

$$\nabla \cdot (f\mathbf{v}) = f\,\nabla \cdot \mathbf{v} + \mathbf{v}\,\nabla f \tag{145}$$

integrando

$$\int \nabla \cdot (f\mathbf{v}) = \int f\,\nabla \cdot \mathbf{v} + \int \mathbf{v}\,\nabla f \tag{146}$$

integral de funções contínuas no intervalo como produto de funções

$$\int \mathbf{D}(f\mathbf{v}) = \langle f, \mathbf{D}\mathbf{v}\rangle + \langle \mathbf{v}, \mathbf{G}f\rangle \tag{147}$$

onde $\mathbf{D}$ e $\mathbf{G}$ são os operadores divergente e gradiente. Nos leva a

$$\int \mathbf{DF} = \langle f, \mathbf{D}\mathbf{v}\rangle + \langle \mathbf{v}, \mathbf{G}f\rangle \tag{148}$$

e pelo Teorema Fundamental do Cálculo

$$\mathbf{F}|_{\partial\Omega} = \langle f, \mathbf{D}\mathbf{v}\rangle + \langle \mathbf{v}, \mathbf{G}f\rangle \tag{149}$$

onde $\partial\Omega$ e a fronteira da região.

B.2 Equivalência de formas da equação da onda

Seja a EDP da equação da onda com força externa dada por

$$\frac{\partial^2 u}{\partial t^2} - c^2\Delta u = f(\mathbf{x}, t) \tag{150}$$

onde t é a variável temporal e $\mathbf{x}$ a espacial.

Podemos reescreve-la sem o uso direto do operador Laplaciano como

$$\begin{cases} \frac{1}{k}\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{v} + f \\ \rho\frac{\partial \vec{v}}{\partial t} = -\nabla u \end{cases} \tag{151}$$

Com $\mathbf{v}$ sendo o vetor velocidade e u o campo de pressão.

Demonstração

Derivando a equação (150) e a primeia parte de (151) com respeito a variável t obtemos

$$\begin{aligned} \frac{1}{k}\frac{\partial^2 u}{\partial t^2} &= -\nabla \cdot \frac{\partial \mathbf{v}}{\partial t} + \frac{\partial f}{\partial t} \\ &= \frac{1}{\rho}\nabla \cdot (\nabla u) + \frac{\partial f}{\partial t}, \end{aligned} \tag{152}$$

considerando a densidade homogênea, sendo

$$\begin{aligned} \nabla \cdot (\nabla u) &= \Delta u \\ \frac{k}{\rho} &= c^2 \\ \frac{\partial f}{\partial t} &= f(x,t). \end{aligned} \tag{153}$$

Como queríamos demonstrar.

B.3 Equivalência do Teorema Fundamental do Cálculo e o produto interno com o vetor unitário sobre o teorema do divergente

Seja a forma geral do Teorema do divergente

$$\int_\Omega \nabla \cdot \mathbf{v}\, f\, dV + \int_\Omega \mathbf{v} \cdot \nabla f\, dV = \int_{\partial\Omega} f\, \mathbf{v} \cdot \hat{n}\, dS \tag{154}$$

quando $\mathbf{v} \equiv 1$ podemos reescrever a forma discreta do divergente $\mathbf{D}\,1 = 0$ como:

$$\langle 1, \mathbf{G} f \rangle = f_n - f_0 \tag{155}$$

Demonstração:

Seja f uma função escalar e $\mathbf{v}$ uma função vetorial. Podemos escrever o produto como:

$$f\mathbf{v} = \mathbf{F} \tag{156}$$

Se diferenciamos o lado esquerdo da equação e aplicarmos a regra da cadeia obtemos

$$\nabla \cdot (f\mathbf{v}) = f\,\nabla \cdot \mathbf{v} + \mathbf{v}\,\nabla f \tag{157}$$

integrando

$$\int \nabla \cdot (f\mathbf{v}) = \int f\,\nabla \cdot \mathbf{v} + \int \mathbf{v}\,\nabla f \tag{158}$$

aplicando a soma de Riemann em um intervalo discreto

$$\int \nabla \cdot (f\mathbf{v}) = \sum f\,\nabla \cdot \mathbf{v} + \sum \mathbf{v}\,\nabla f \tag{159}$$

que é equivalente em forma de produto interno

$$\int \mathbf{D}(f\mathbf{v}) = \langle f, \mathbf{D}\mathbf{v}\rangle + \langle \mathbf{v}, \mathbf{G}f\rangle \tag{160}$$

onde $\mathbf{D}$ e $\mathbf{G}$ são operadores divergente e gradiente, que nos leva a

$$\int \mathbf{DF} = \langle f, \mathbf{D}\mathbf{v}\rangle + \langle \mathbf{v}, \mathbf{G}f\rangle \tag{161}$$

e pelo Teorema Fundamental do Cálculo

$$\mathbf{F}|_{\partial\Omega} = \langle f, \mathbf{D}\mathbf{v}\rangle + \langle \mathbf{v}, \mathbf{G}f\rangle \tag{162}$$

onde $\partial\Omega$ é a fronteira da região. Ao fazermos $\mathbf{v} \equiv 1$ sobre uma região unidimensional

$$\begin{aligned} F_n - F_0 &= \langle f, \mathbf{D}1\rangle + \langle 1, \mathbf{G}f\rangle \\ &= \langle f, 0\rangle + \langle 1, \mathbf{G}f\rangle \\ &= 0 + \langle 1, \mathbf{G}f\rangle \\ &= \langle 1, \mathbf{G}f\rangle. \end{aligned} \tag{163}$$

Conforme queriamos demonstrar.

APÊNDICE C – Código fonte

Todos os códigos a seguir foram feitos com Python 3.7.1 (ROSSUM, 1995), Numpy 1.16.0 (OLIPHANT, 2006), Scipy 1.2.0 (JONES et al., 2001–), SymPy 1.3 (MEURER et al., 2017) e Matplotlib 2.2.3 (HUNTER, 2007), serão idiomáticos, para referência de implementação é recomendado utilizar os algoritmos do capítulo 2.

C.1 Malha intercalada

vetorgerador.py

```
# Algoritmo 1 - Vetor Gerador dos pontos da malha intercalada

from numpy import arange


def VetorGerador(k):
    assert (k % 2 == 0 and k > 0), "Order␣must␣be␣an␣even␣number␣>␣0"
    return arange((1-k)/2, k/2, 1)
```

C.2 Matriz do estencil MDF central

estencilmdf.py

```
# Algoritmo 2 - Calcula o estencil do MDF pelas matrizes de Vandermonde

from numpy import vander
from scipy.linalg import lstsq
from vetorseletor import VetorSeletor


def EstencilMDF(g):
    # Matriz de vandermonde associada
    T = vander(g, increasing=True).T

    # Vetor seletor, para primeira derivada
    o = VetorSeletor(len(g))

    # Estencil de diferencas finitas centrais da primeira derivada
    s = lstsq(T, o)[0]

    return s
```

matrizestencil.py

```
# Algoritmo 3 - Cria matriz bandada do estencil MDF

from numpy import asarray, zeros, concatenate
from numpy.lib.stride_tricks import as_strided as strided
from scipy.sparse import diags

def MatrizEstencil(g, N):
    n=len(g)
    T = zeros((N,N+n-1))
    for x in range(N):
        T[x][x:x+n]=g
    return T


def MatrizEstencilEsparsa(g, N):
    n=len(g)
    return diags(g, range(N), shape=(N,N+n-1)).toarray()


def MatrizEstencilRapida(g, N):
    """Creates a ‘g‘ generated banded matrix with ’N’ rows."""
    g = asarray(g)
    p = zeros(N-1, dtype=g.dtype)
    b = concatenate((p, g, p))
    s = b.strides[0]
    n = len(g)
    return strided(b[N-1:], shape=(N, n+N-1), strides=(-s, s))
```

geradordivergente.py

```
# Algoritmo 4 - Vetor gerador para matriz de Vandermonde no
#               contorno do dominio para o operador divergente

from numpy import linspace


def GeradorContornoDivergente(k):
    """Returns a matriz with ’k’ generators for
    divergent near boundaries"""
    l = int((3/2) * k)
    g0 = linspace(-.5, (l-.5), l, endpoint=False)
    sub = linspace(0, k, k, endpoint=False)
    return g0 - sub.reshape(sub.shape[0],1)
```

geradorgradiente.py

```
# Algoritmo 5 - Vetor gerador para matriz de Vandermonde no
```

```
#                contorno do dominio para o operador gradiente

from numpy import linspace, zeros, array


def GeradorContornoGradiente(k):
    """Returns a matriz with 'k' generators for divergent near
       boundaries"""
    l = int((3/2) * k)
    g = []
    for i in range(int(l/2)):
        # These are the points near the boundary
        g_ = zeros(l)
        g_[0] = 0
        g_[1:] = linspace(1/2, (l-.5), l-1, endpoint=False)
        g.append(g_ - i)

    for i in range(int(l/2), k):
        # These are the interior points
        g.append(linspace(-1/2, (l-.5), l, endpoint=False)-(i-1))

    return array(g, dtype=float)
```

bordadivergente.py

```
# Algoritmo 6 - Calcula o estencil e o nucleo dos pontos da
#                borda do operador divergente

from numpy import zeros, vander, array, newaxis, concatenate
from sympy import linsolve, Matrix
from numpy import float_

from vetorseletor import VetorSeletor
from geradordivergente import GeradorContornoDivergente


def ContornoDivergente(k):
    """Returns the Stencil near the boundaries

    Where:
        k: Order of accuracy

    Returns: PI, NU
        Mimetic submatrix at boundary
        Kernel of boundary Vandermonde matrix
    """
    # Vetor seletor de ordem da primeira derivada
```

```
    o = VetorSeletor(k+1)

    # Kernel rank of Vandermonde matrix at boundary
    d = int(k/2 - 1)

    # Constroi os pontos do operador divergente proximo ao contorno
    g = GeradorContornoDivergente(k)

    l = 3*k//2
    Pi = zeros([d, l])
    N = []
    for i in range(d):
        V_tmp = vander(g[i], k+1, increasing=True).T
        #r_tmp = linsolve(Matrix(c_[V_tmp, o]))
        r_tmp = linsolve(Matrix(concatenate((V_tmp, o[:,newaxis]), axis
           =1)))
        r_tmp = r_tmp.subs([(s, 0) for s in r_tmp.free_symbols])
        Pi[i] = array(*[e for e in r_tmp]).astype(float_)
        N.append(Matrix(V_tmp).nullspace())

    # Aqui tem que retornar a lista de array com o nullspace completo
    Nu = zeros([len(N[0]), l])
    for i,n in enumerate(N[0]):
        Nu[i] = array(n.T).astype(float_)[0]

    return Pi, Nu
```

bordagradiente.py

```
# Algoritmo 6 - Calcula o estencil e o nucleo dos pontos da
#               borda do operador gradiente

from numpy import zeros, vander, array, concatenate, newaxis
from numpy import float_

from sympy import linsolve, Matrix

from vetorseletor import VetorSeletor
from geradorgradiente import GeradorContornoGradiente


def ContornoGradiente(k):
    """Returns the Stencil near the boundaries to Gradient

    Where:
        k: Order of accuracy
```

```
    Returns: PI, NU
        Mimetic submatrix at boundary
        Kernel of boundary Vandermonde matrix
    """
    # Vetor seletor de ordem da primeira derivada
    o = VetorSeletor(k+1)

    # Kernel rank of Vandermonde matrix at boundary
    d = int(k/2)

    # Constroi os pontos do operador divergente proximo ao contorno
    g = GeradorContornoGradiente(k)

    l = 3*k//2
    Pi = zeros([d, l])
    N = []
    for i in range(d):
        V_tmp = vander(g[i], k+1, increasing=True).T
        #r_tmp = linsolve(Matrix(c_[V_tmp, o]))
        r_tmp = linsolve(Matrix(concatenate((V_tmp, o[:,newaxis]), axis
           =1)))
        r_tmp = r_tmp.subs([(s, 0) for s in r_tmp.free_symbols])
        Pi[i] = array(*[e for e in r_tmp]).astype(float_)
        N.append(Matrix(V_tmp).nullspace())

    Nu = zeros([len(N[0]), l])
    for i,n in enumerate(N[0]):
        Nu[i] = array(n.T).astype(float_)[0]

    return Pi, Nu
```

pitilde.py

```
# Algoritmo 7 - Constroe a matriz \tilde(\Pi)

from matrizestencil import MatrizEstencil
from numpy import concatenate


def MatrizPiTilde(Pi, Nu, s):
    """Returns the PiTilde matrix given the Pi matrix, Nu Nullspace
    and the 's' stencil"""
    missingRowsCount = s.shape[0] - Pi.shape[0]
    N = Pi.shape[1] - s.shape[0] + 1
    A = MatrizEstencil(s, N)[-missingRowsCount:]
    PiT = concatenate((Pi, A, Nu)).T
    return PiT
```

vetormimetico.py

```
# Algoritmo 8 - Calcula vetor mimetico da submatriz A(alpha)

from numpy import zeros, float_


def VetorMimetico(s):
    """Returns the resultant mimetic vector for the submatrix A"""

    k = len(s)
    l = 3*k//2

    h = zeros(l,dtype=float_)
    h[0] = 1

    i = k//2 + 1
    h[i:]=s.cumsum()[:(l-i)]

    return -h
```

pesos.py

```
# Resolve o sistema linear \tilde{Pi} \tilde{q} = d
from scipy.linalg import lstsq


def VetorPeso(Pi, d):
    """Returns the weights for the diagonal matrix and the lambda"""

    k = len(d)*2//3
    i = k//2-1

    solution = lstsq(Pi, d)[0]

    weights = solution[:-i]
    lambda_ = solution[-i:]
    return weights, lambda_
```

submatriza.py

```
# Algoritmo 9 - Calcula a submatriz A(alpha)


def MatrizA(PiTilde, Nu, q, lambda_):

    k = len(q)
    A = PiTilde.T[:k]
```

```
    alpha = lambda_ / q[:len(lambda_)]
    for i in range(len(alpha)):
        A[i] += alpha[i] * Nu[i]

    return A
```

C.3 Código principal

principal.py

```
#!/usr/bin/env python3
################################################################
# Algoritmo X - Codigo Principal
################################################################

import sys
from sympy import pprint

from utils import rationalize
from vetorgerador import VetorGerador
from estencilmdf import EstencilMDF
from vetorseletor import VetorSeletor
from bordagradiente import ContornoGradiente
from bordadivergente import ContornoDivergente
from pitilde import MatrizPiTilde
from vetormimetico import VetorMimetico
from pesos import VetorPeso
from submatrizA import MatrizA
from operadormimetico import OCG

k = 4                # Ordem de precisao do operador
h = 1                # Espacamento da celula
N = 3*k-1            # Menor numero de pontos possivel
l = int(3/2*k)

# Obtendo os pontos vizinhos para aproximacao de Taylor
g = VetorGerador(k)

# Obtendo o estencil do MDF
s = EstencilMDF(g)

# Constroi o vetor resultado 'd' (que independe do operador)
d = VetorMimetico(s)


# Obtem operador Divergente
```

```
Pi_D, Nu_D = ContornoDivergente(k)
PiT_D = MatrizPiTilde(Pi_D, Nu_D, s)
weights_D, lambda_D = VetorPeso(PiT_D, d)
D_A = MatrizA(PiT_D, Nu_D, weights_D, lambda_D)

# Obtem operador Gradiente
Pi_G, Nu_G = ContornoGradiente(k)
PiT_G = MatrizPiTilde(Pi_G, Nu_G, s)
weights_G, lambda_G = VetorPeso(PiT_G, d)
G_A = MatrizA(PiT_G, Nu_G, weights_G, lambda_G)

# Constroi a matriz dos operadores
D = OCG(D_A, s, N)
G = OCG(G_A, s, N+3)
# L = np.dot(D, G)



sys.exit(0)

# Debug info
print("\nPi_D␣=␣")
pprint(rationalize(Pi_D))

print(f"\nNu_D␣=␣")
pprint(rationalize(Nu_D))

print("\nPi_G␣=␣")
pprint(rationalize(Pi_G))

print(f"\nNu_G␣=␣")
pprint(rationalize(Nu_G))

print(f"\nd^T␣=␣")
pprint(rationalize(d))
```

C.4 Funções de apoio

vetorseletor.py

```
import numpy as np

def VetorSeletor(N, n=1):
    """Returns the N-dimensional selector vector to the n-th derivative
        """
    o = np.zeros(N)
    o[n] = 1
```

```
    return o
```

utils.py

```
from sympy import nsimplify, Matrix


def rationalize(x, tolerance=10e-9):
    """Returns a Matrix with symbolic rational numbers"""
    try:
        Mx = Matrix(x)
    except TypeError:
        raise("Input␣isn't␣a␣valid␣matrix␣type.␣Need␣to␣be␣a␣sequence")

    shape = Mx.shape
    rationalized = [nsimplify(e, tolerance=tolerance) for e in Mx]
    return Matrix(rationalized).reshape(*shape)
```