



EAMC2019

XII Encontro Acadêmico de Modelagem Computacional

Laboratório Nacional de Computação Científica - LNCC/MCTIC

04 a 07 de Fevereiro de 2019.

Livro de resumos



Laboratório
Nacional de
Computação
Científica

XII Encontro Acadêmico de Modelagem Computacional

Laboratório Nacional de Computação Científica - LNCC/MCTIC

04 a 07 de Fevereiro de 2019. Petrópolis - RJ.

Comitê Organizador

Claudio Daniel Tenório de Barros

Diego Tavares Volpatto

Éden Pereira da Silva

Ingrid Bertin Carneiro

Jorge Morvan Marotte Luz Filho

Lucas dos Anjos

Luis Alonso Mansilla Alvarez

Comitê Científico

Gastão Florêncio Miranda Junior - UFS

Iury Higor Aguiar da Igreja - UFJF

Marcos Garcia Todorov - LNCC

Sicilia Ferreira Ponce Pasini Judice - FAETERJ

Suelen de Souza Rocha - UFPB

Thiago de Oliveira Quinelato - Unicamp

Thiago José Machado - UFPB

Trabalhos completos

	Page
<i>Implementação de software para reconstrução 3D utilizando imagens 2D de microestruturas obtidas através técnica de seccionamento em série</i> Andrade A.B.R., Ferreira J.L.G., Baía D.M., Silva W.L., Neves T.A., Rodríguez Y., Rangel P.	5
<i>Escoamento bifásico com gravidade em um meio heterogêneo de dois tipos de rocha</i> Araujo I.L., Bermúdez P.R., Rodríguez Y.	16
<i>Seleção de modelos para o problema de incrustação em trocadores de calor</i> Bonifácio C.L.C., Silva R.S., Cardozo F.H.P.	25
<i>Análise comparativa de métodos numéricos aplicados ao fluxo de carga no software matlab</i> Cardoso D.L., Montovani D.S.	34
<i>Numerical study of the Swift-Hohenberg equation: Defects dynamics for first and second order temporal discretization</i> Coelho D.L., Pontes J.R.M., Mangiavacchi N.	42
<i>Métodos de volumes finitos aplicado a um problema de transporte de contaminantes</i> Coimbra A.L.S.C., Telles W.R.	52
<i>Análise comparativa entre os métodos algoritmo de colisão de partículas e newton na resolução de sistemas de equações não lineares</i> Dias R.M.M.S., Telles W.R.	54
<i>Utilização do método de Monte Carlo para a resolução de problemas complexos de integração</i> Ferreira V.B., Pereira T.J.	64

<i>Dinâmicas de infecção e tratamento do vírus da hepatite C utilizando modelo matemático com estrutura etária</i>	65
Garbero S.J., Quintela B.M., Weber R., Lobosco M.	
<i>Parameter identification problem in Hodgkin and Huxley model</i>	73
Mandujano J.A.	
<i>Existência e comportamento assintótico de soluções para a equação da onda de 4a ordem envolvendo expoente variável</i>	83
Panni W.S., Ferreira J.	
<i>Aplicação de observadores de estado em um modelo SIR para estimação da população de mosquitos transmissores da febre amarela</i>	91
Rocha F.P., Giesbrecht M. Meyer J.F.C.A.	
<i>Identificação de danos em uma viga de Euler-Bernoulli utilizando redes neurais artificiais: Uma análise em relação à posição do dano na estrutura na presença de ruidos</i>	101
Silva A.A.M., Santos D.T., Portella R.A., Corrêa C.A.	
<i>Algoritmos de inteligência de enxame com busca local baseada em pull move aplicados ao problema de predição de estrutura de proteínas no modelo 2D HP</i>	111
Soares C.R., Cavalcanti D.M.	
<i>Aplicação do método das diferenças finitas em um problema de transferência de calor utilizando as formulações explícita e implícita</i>	121
Tavares A., Telles W.R., Semaan G.S.	
<i>Avaliação dos métodos de otimização Luus-Jaakola e algoritmo de colisão de partículas em um problema de transferência de calor</i>	131
Tavares A., Telles W.R., Semaan G.S.	

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

IMPLEMENTAÇÃO DE SOFTWARE PARA RECONSTRUÇÃO 3D UTILIZANDO IMAGENS 2D DE MICROESTRUTURAS OBTIDAS ATRAVÉS DA TÉCNICA DE SECCIONAMENTO EM SÉRIE

Ana Beatriz Rodrigues de Andrade Graça¹ - anabeatrizgraca@id.uff.br

Jonathas Luis Groetares Ferreira¹ - jonathasferreira@id.uff.br

Diego Magalhães Baía² - diegobaia@id.uff.br

Weslley Luiz da Silva³ - weslleyassis@id.uff.br

Tiago Araújo Neves³ - tneves@id.uff.br

Yoisell Rodríguez Núñez³ - yoisellrn@id.uff.br

Paulo Rangel Rios⁴ - prrios@id.uff.br

¹Aluno de Mestrado - Escola de Engenharia Industrial Metalúrgica de Volta Redonda/UFF

²Aluno de Graduação - Escola de Engenharia Industrial Metalúrgica de Volta Redonda/UFF

³Professor Adjunto - Escola de Engenharia Industrial Metalúrgica de Volta Redonda/UFF

⁴Professor Titular - Escola de Engenharia Industrial Metalúrgica de Volta Redonda/UFF

Resumo. Um material isotrópico é aquele que tem as mesmas propriedades em todas direções. A análise do comportamento de um material desse tipo, baseado apenas em imagens metalográficas bidimensionais, é insuficiente devido à natureza tridimensional, pois não possibilita a observação de um volume microestrutural da amostra. Nesse caso, a aplicação de técnicas de modelagem numérica pode apoiar e ampliar a investigação experimental tridimensional realizada. Sendo assim, este trabalho tem como objetivo principal criar domínios discretos com fidelidade a ambientes naturais. A reconstrução 3D é feita a partir de uma técnica chamada seccionamento em série, empregada para reconstruir uma microestrutura através da aquisição de seções planas bidimensionais de uma amostra real. O programa está sendo desenvolvido seguindo o paradigma da Programação Orientada a Objetos, implementado na linguagem de programação C++. Neste trabalho implementou-se duas abordagens para a determinação dos planos intermediários às seções de entrada: a repetição de planos adjacentes e uma aproximação do que seriam os planos intermediários por meio da média das cores dos planos reais vizinhos interirícos, utilizando o sistema RGB de cores.

Palavras-chave: Computação gráfica, Processamento de imagens, Malhas discretas

1. INTRODUÇÃO

Domínios computacionais (ou volumes de controle) são regiões nas quais se deseja determinar alterações em variáveis específicas que evoluem de maneira dinâmica e são, geralmente, modelados a partir de equações algébricas, representação de computação gráfica, criação de malhas computacionais a partir de eixos cartesianos, dentre outros sistemas. O modo de se gerar malhas discretas possui um papel importante quando associado ao desenvolvimento de modelos computacionais para a simulação de fenômenos da natureza via métodos numéricos.

Porém, a representação computacional de detalhes de domínios com geometrias complexas, como um cérebro com todos os detalhes dos sulcos cingulados, o corpo caloso e o cerebelo, tem sido um problema na maioria dos casos nos quais a geometria põe em risco a integridade dos resultados gerados via análise numérica, pois milímetros de diferença do original podem causar relevantes erros nos resultados finais em determinados casos, como em cálculos de concentrações de células cancerosas, por exemplo. Com o advento da computação, novas técnicas de representação gráfica foram criadas e com elas novas formas de expressar modelos físicos por meio de técnicas computacionais.

A microscopia óptica e a eletrônica fornecem apenas imagens bidimensionais (2D), sem fornecer informações quantitativas diretas sobre a estrutura interna tridimensional (3D) de um material. Cada microestrutura apresenta características particulares, tais como o tamanho, informações geométricas, distribuição de fases e morfologia. E, para entender o comportamento do material envolvido, é necessário estabelecer um vínculo entre a microestrutura e suas propriedades (Wunsch and Chawla, 2003; Zankel et al., 2014). Porém, o material que possui uma natureza tridimensional não deve ser analisado apenas com imagens metalográficas bidimensionais, pois essa análise se torna insatisfatória. Com isso, certas informações geométricas e morfológicas sobre uma microestrutura tridimensional não podem ser deduzidas devido a limitações bidimensionais, pois não possibilita a observação de um volume microestrutural da amostra. Nesse caso, a aplicação de técnicas de modelagem numérica pode apoiar a investigação experimental tridimensional realizada através da reconstrução 3D (de Freitas, 2008; Tewari and Gokhale, 2000). O método utilizado neste trabalho permite a reconstrução das estruturas internas de materiais de modo que o resultado final seja um conjunto de dados 3D que torne possível visualizar e identificar elementos tais como poros, grãos, fases, precipitados, e outros.

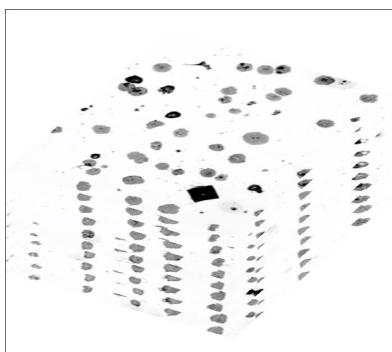
Inicialmente é necessário aplicar a técnica de seccionamento em série, descrita na subseção 2.1, no material de interesse. Esta técnica é empregada para adquirir seções bidimensionais de uma determinada microestrutura. Como resultado desse processo, obtém-se uma sequência de imagens que servirão como dados de entrada para o código computacional que está sendo desenvolvido neste trabalho. O modo como o programa está sendo implementado fornece um arquivo de saída que contém os dados necessários para gerar uma malha discreta que representa a reconstrução tridimensional, já validada e aplicada em diversas áreas da ciência de maneira geral, através de softwares de visualização 3D. Utiliza-se duas abordagens distintas, implementadas no código computacional, para superar a perda de informações que o processo de seccionamento em série causa na hora do desbastamento de uma amostra real.

Sendo assim, neste trabalho as seções abaixo estão organizadas da seguinte maneira: a seção 2. descreve detalhadamente como o processo de seccionamento em série é feito e mostra dois exemplos de uma série de seções bidimensionais, resultantes deste processo laboratorial, que foram empilhadas através da ferramenta computacional principal deste trabalho. Ainda nesta seção encontra-se uma descrição de como o código computacional tem sido implementado e quais softwares de visualização 3D são utilizados para gerar os resultados gráficos a partir do arquivo de saída gerado. Duas abordagens que já foram implementadas, e estão em processo de teste, são relatadas nessa seção: a repetição de planos adjacentes e planos intermediários obtidos por meio da média das cores. Já a seção 3. apresenta um pouco sobre os dois materiais nos quais a técnica de seccionamento em série foi aplicada e, posteriormente, o resultado utilizado como dado de entrada para o software de interesse: de Ferro Fundido Nodular e Aço Inoxidável Duplex. Na seção 4. a reconstrução das microestruturas é realizada utilizando as duas abordagens propostas na seção 2.2. Por fim, encerra-se este artigo com as conclusões na seção 5.

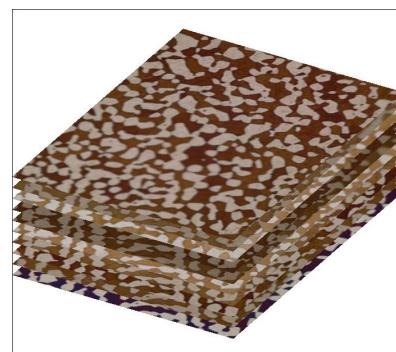
2. METODOLOGIA

2.1 Seccionamento em Série

O processo de seccionamento em série consiste em remover camadas sequenciais do material de interesse. A cada camada removida mediante a lixamento e/ou polimento metalográfico da superfície da amostra é necessário corrigir a amostra e fotografar sua superfície usando microscopia óptica (Tewari and Gokhale, 2000; Spowart et al., 2003). Através desse processo obtém-se imagens 2D de cada camada que são chamadas de *seções planas*, *seções bidimensionais* ou *fatiadas em série* da microestrutura. Após a obtenção das imagens ocorre o empilhamento das mesmas, conforme pode ser visto na Fig. 1.



(a) Ferro Fundido Nodular.



(b) Aço Inoxidável Duplex.

Figura 1- Exemplo de duas pilhas de 10 seções em série de Ferro Nodular (a) e Aço Duplex (b).

Existem alguns métodos mais elaborados de seccionamento em série para investigação e representação 3D da estrutura interna de materiais, como por exemplo o FIB (Focused Ion Beam), que fornece resultados desbastando a amostra do material com um feixe de íons focalizado (Zankel et al., 2014). No entanto, o método utilizado neste trabalho foi o método de metalografia convencional que é realizado pela seguinte sequencia de passos básicos: 1- A preparação da amostra é feita através de ataque químico adequado ao que se deseja analisar na microestrutura. O objetivo deste ataque é obter um bom contraste com relação as fases presentes no material metálico de modo que as fases necessitam estar bem distintas em relação à matriz; 2- É necessário escolher qual a região da amostra que será analisada; 3- Microimpressões Vickers (Wunsch and Chawla, 2003) são feitas sobre a região de interesse escolhida no segundo passo. Elas auxiliam no alinhamento das seções e servem para identificar a exata região que foi selecionada após sucessivos polimentos. Além disso, as Microimpressões Vickers também possibilitam medir precisamente a distância entre as seções consecutivas logo após cada lixamento metalográfico, pois o tamanho da microimpressão diminui conforme o desbastamento ocorre; 4- Somente neste passo do processo adquiri-se a primeira imagem de uma das seções em série, e para isto, com o auxílio do microscópio óptico, ocorre uma tomada de foto da região da amostra que foi escolhida para a análise.

Ao final do quarto passo é feito o polimento e/ou lixamento metalográfico da amostra em uma determinada direção, que é considerado o passo mais crítico do processo, pois existem alguns fatores que interferem na qualidade final da superfície. Dependendo da estrutura do material o polimento é suficiente para este passo. Materiais de maior dureza necessitam também do lixamento. Essa etapa de polimento e/ou lixamento e tomada de foto se repete várias ve-

zes até obter um número de seções suficientemente bom para reconstruir o objeto em estudo (de Freitas, 2008; Wunsch and Chawla, 2003; Tewari and Gokhale, 2000; Li et al., 1998).

Quando as fotografias são obtidas por meio de microscopia óptica é necessário fazer o alinhamento para que, na hora da reconstrução, os objetos de interesse possam ser vistos e analisados corretamente e com maior clareza (Ganti et al., 2018; Madej et al., 2017).

As fotos capturadas são salvas em arquivos digitais de diferentes formatos. Estes arquivos são portadores de algumas informações, tais como: tamanho da imagem, data, densidade de pixels, níveis de vermelho, verde e azul em cada pixel e etc. A organização visual dos elementos é responsável por posicionar cada pixel em seu devido lugar. Deste modo, a representação geométrica da cena é mantida a rigor, e formas geométricas complexas podem ser representadas com maior fidelidade.

Um dos fatores relevantes a ser considerado na técnica de seccionamento em série é a espessura a ser retirada da amostra. Como o desbastamento da amostra real gera grandes perdas de informações, procura-se aplicar métodos computacionais para determinar a microestrutura entre dois planos discretos. Busca-se encontrar os planos intermediários que representem de modo fidedigno as informações que foram perdidas no processo de desbaste da microestrutura. Outro fator importante é trato das imagens (aquisição, arquivamento, manejo, etc), que este trabalho tenta aprimorar, uma vez que falhas neste processo podem causar perda de dados da microestrutura.

2.2 Desenvolvimento do Código Computacional

A ferramenta principal foi desenvolvida com o auxílio do software Microsoft Visual Studio® escrevendo-a na linguagem de programação orientada a objetos C++. A escolha desta linguagem se deve a compatibilidade com diversos sistemas operacionais e a possibilidade de novas funcionalidades serem facilmente incorporadas no código.

Com este código computacional é possível realizar novos estudos com a matriz reconstruída, pois os dados não são descartados. A matriz em três dimensões fica salva em um arquivo de saída e pode ser utilizada por outras ferramentas numéricas e/ou programas de simulação computacional, tais como: Tecplot 360® (Tecplot, 2010) ou equivalentes.

Também utilizou-se a biblioteca OpenCV, desenvolvida pela Intel, disponível em (Intel Corporation, 2000). Ao fim do trabalho obteve-se uma ferramenta computacional que reconstrói em 3D um material seccionado em série e, posteriormente, salva todas as características geométricas deste material para futuras análises. A possibilidade de adquirir um material reconstruído em 3D e reproveitá-lo para realizar simulações de transformação de fase, crescimento de tumores, troca de calor, análise de deformação, dentre outros, torna o programa inédito no mercado e de grande aplicabilidade no meio científico.

Duas técnicas computacionais foram implementadas para usar a sintaxe de uma imagem digital e criar um domínio computacional com referências ao espaço discreto com coordenadas cartesianas em três dimensões.

Repetição de planos adjacentes. Atualmente as cores que são vistas em aparelhos digitais tais como a TV, a tela de um celular, o monitor de um computador, uma câmera fotográfica geram a cor através de luz. Essa formação de cores é composta pelas três cores básicas: vermelho, verde e azul que no inglês são red, green e blue que formam a sigla RGB para esse sistema de cores. Um pixel é formado por essas três cores em conjunto. Ele pode variar entre todas as

cores, desde o branco, que é a mistura de todas as cores, até o preto que é a ausência de cores no sistema RGB.

Considere um plano como sendo uma imagem bidimensional que foi obtida através do processo de seccionamento em série descrito na subseção 2.1. Como o domínio computacional que pretende-se gerar é um domínio discreto, então é possível pensar em um número finito de pontos que estão contidos em cada plano utilizado.

O algoritmo começa percorrendo os planos mais baixos e passa para os planos superiores conforme a quantidade de pontos, que cada plano contém, acaba de ser percorrida. Com isso, o primeiro ponto do primeiro plano está fixo na origem do sistema tridimensional de coordenadas. O próximo ponto vai ser o ponto cuja a coordenada x avança uma unidade no sentido positivo de seu eixo, ou seja, avança para o próximo pixel da imagem. Isso ocorre até que a quantidade de pixels no eixo das abscissas se acabe, pois tem-se um número finito de pixels por imagem. Após essa etapa, avança-se uma unidade de medida no eixo das ordenadas, ou seja, avança-se um pixel na direção y e segue-se este raciocínio para o eixo das cotas até que o código computacional atinja o número que é calculado previamente pela multiplicação entre a altura e a largura da seção bidimensional vezes a quantidade de seções bidimensionais que são consideradas para fazer o empilhamento que, posteriormente, resulta na reconstrução 3D. Durante esse processo, as cores são salvas em um vetor de tamanho $x \times y \times z$, onde x corresponde a largura da imagem, y é a altura e z é a quantidade total de planos que é escolhido para usar na reconstrução. É possível armazenar 256 possíveis tons de R, de G e de B, que totalizam 16.777.216 possibilidades de cores diferentes de pixel.

Como dados de entrada, precisa-se do número inicial de imagens que foi escolhido de acordo com o processo de seccionamento em série e do número de planos intermediários que serão repetidos a cada duas imagens.

O preenchimento dos planos intermediários é feito com o auxílio de comandos condicionais e comandos de repetição padrões da linguagem de programação C++. A ideia principal é que a cor do plano de baixo vai se repetir nos planos intermediários mais próximos que estão em cima, de modo que as coordenadas x e y desse ponto não se alterem. Cada ponto/pixel da imagem possui uma cor. Considere que esses valores são os estados da matriz em um determinado ponto. É preciso repetir os estados dos planos inferiores nos planos superiores, onde os planos intermediários estão vazios. A solução para esse problema foi identificar onde cada plano acaba de ser preenchido, isto é, foi necessário saber em qual ponto a coordenada z se altera. Isto foi necessário porque o código está vetorizado. Para esta identificação, utilizou-se a ideia de contadores. O contador principal serve para contar quantos pontos já foram percorridos através do comando de repetição. E quando esse contador for divisível pela quantidade total de pontos contidos em um plano, isto é, esta divisão possui resto igual zero, sabe-se que um plano acabou de ser preenchido com seus respectivos estados. Assim, repete-se o estado desse plano no plano intermediário acima, através do segundo contador que volta o vetor que contém os valores dos estados dos pontos na posição inicial do plano que pretende-se repetir. E um terceiro contador vai verificar quantos planos foram repetidos. Se esse terceiro contador for igual ao número de planos que deseja-se repetir a cada dois planos, então o primeiro contador não é subtraído pela quantidade total de pontos que contém em um plano: ele apenas continua a preencher o próximo plano que não vai ser repetido dessa vez. Caso contrário, o primeiro contador é subtraído pela quantidade de pontos que um plano contém, com o objetivo de voltar no início dos estados. Essa ideia se repete conforme a quantidade de planos que coloca-se como dado de entrada até o último plano ser totalmente preenchido com seu devido estado.

Planos intermediários obtidos por meio da média das cores. Uma outra alternativa para preencher os planos intermediários é utilizar a média entre as cores. Para obter a média entre duas cores, basta somar os valores de R, os valores de G e os valores de B e dividir cada um por dois. O resultado é uma nova cor que é a média entre duas cores. Todos os números não inteiros são arredondados para o inteiro mais próximo.

O primeiro passo desta técnica é empilhar seções bidimensionais, que são dados de entrada para o software, do mesmo modo que foi feito para construir a repetição de planos. Depois que todos os planos principais são empilhados juntamente com os planos vazios, os planos intermediários serão preenchidos utilizando uma proporção vinculada com a Eq. (1) que é a taxa de variação de uma função. Neste caso, considerou-se uma função $f(x, y, z)$ de três variáveis que representa os níveis de R, de G e de B de cada pixel.

$$T_v = \frac{f_i(x_k, y_k, z_{k+1}) - f_i(x_k, y_k, z_k)}{z_{k+1} - z_k}, \quad i = 1, 2, 3 \quad \text{e} \quad k = 1, 2, \dots, n. \quad (1)$$

A taxa de variação T_v é “o quanto o R, o G e o B distam” um dos outros através de uma proporção que está relacionada a quantidade de planos intermediários que se aspira preencher. As coordenadas x e y são as mesmas para um dado ponto fixo do plano que o algoritmo percorre. O que muda nessa posição é a informação da coordenada z que indica em qual plano esse ponto de coordenada x e y está. Isto significa que as variáveis x_k e y_k são as mesmas para pontos escolhidos na mesma direção (em cima e em baixo). Já as variáveis z_k e z_{k+1} correspondem aos primeiro e segundo planos principais quando $k = 1$, depois correspondem ao segundo e terceiro planos principais quando $k = 2$ e assim por diante até o algoritmo acabar de percorrer todos os planos desejados. A variável n representa a quantidade de pontos que um plano possui. Observou-se que o valor da diferença entre z_{k+1} e z_k é o mesmo valor da quantidade total de planos intermediários mais um. Para $i = 1$ o valor de $f_1(x_k, y_k, z_{k+1})$ representa o valor de R no primeiro ponto do primeiro plano principal e o valor de $f_1(x_k, y_k, z_k)$ representa o valor de R no primeiro ponto do segundo plano principal. Esse raciocínio se propaga analogamente para o G, representado por f_2 , e para B, representado por f_3 . O código preenche todos os planos vazios por blocos com o objetivo de tentar deixar o código paralelizável.

Dessa forma, conseguiu-se obter os planos intermediários utilizando a média das cores. Na Fig. 2 é possível ver três exemplos de média de cores. As cores intermediárias são calculadas de modo que os pontos intermediários que estão mais perto de um ponto que contém um estado específico fiquem mais próximos desse estado e os pontos mais perto de um segundo estado fiquem mais próximos desse segundo estado.

R	G	B	Cor
255	255	255	
213	213	213	
170	170	170	
128	128	128	
85	85	85	
43	43	43	
0	0	0	

(a) Branco e preto.

R	G	B	Cor
127	127	127	cinza escuro
129	106	109	cinza
130	85	92	cinza claro
132	64	74	cinza muito claro
133	42	56	cinza branco
135	21	39	cinza amarelo
136	0	21	cinza vermelho

(b) Cinza e vinho.

R	G	B	Cor
237	28	36	vermelho escuro
240	64	30	vermelho
243	99	24	vermelho laranja
246	135	18	vermelho laranja amarelo
249	171	12	vermelho amarelo
252	206	6	vermelho amarelo amarelo
255	242	0	amarelo

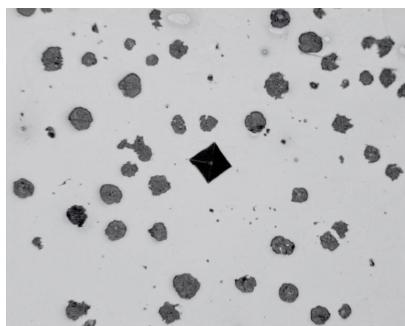
(c) Vermelho e amarelo.

Figura 2- A partir de duas cores principais escolheu-se, como exemplo, cinco cores intermediárias para serem calculados os valores de R, de G e de B, utilizando a média de cores.

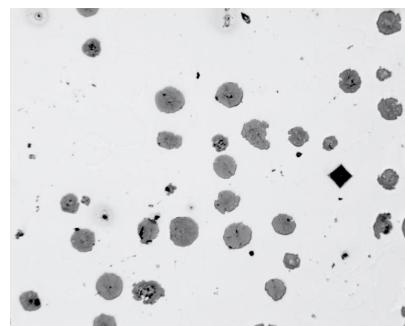
3. MATERIAIS

Um foco maior é dado à reconstrução de amostras de materiais metálicos, em escala microestrutural. Porém, nada impedirá que a técnica seja aplicada a outros materiais. Neste trabalho foram usados Ferro Fundido Nodular e Aço Inoxidável Duplex.

O Ferro Fundido Nodular é uma liga composta de carbono e silício com teor de carbono acima de 2% e silício entre 1 e 3%, podendo conter outros elementos químicos tais como manganes, fósforo, zircônio, estanho, e etc. Geralmente é utilizado na indústria para a confecção de peças que necessitem de maior resistência ao impacto em relação aos ferros fundidos cinzentos, além de maior resistência à tração, resistência ao alongamento e escoamento (de Freitas, 2008; Chiaverini, 1990). A Fig. 3 apresenta duas imagens microscópicas de uma amostra de Ferro Fundido Nodular que foram obtidas através do processo de seccionamento em série. A Microimpressão Vickers em forma de losango é visível em ambas imagens.



(a) Primeira seção arquivada no computador.

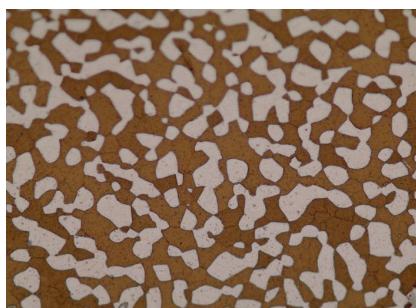


(b) Última seção arquivada no computador.

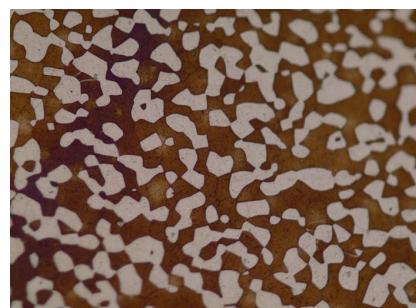
Figura 3- Microscopia de amostra de Ferro Fundido Nodular. Fonte: (de Freitas, 2008).

Já o Aço Duplex é um aço inoxidável composto por dois tipos de microestruturas: Ferríta e Austenita, onde a parcela volumétrica de ambas fases são aproximadamente iguais. Este aço é caracterizado pela resistência alta mecânica e resistência a corrosão (Copalert, 2008).

A Fig. 4 apresenta duas imagens microscópicas de uma amostra de Aço Inoxidável Duplex que foram obtidas através do processo de seccionamento em série.



(a) Primeira seção arquivada no computador.



(b) Última seção arquivada no computador.

Figura 4- Microscopia de amostra de Aço Duplex.

Mais informações sobre estes materiais podem ser encontradas em (Chiaverini, 1990) e em (Copalert, 2008).

4. RESULTADOS E DISCUSSÃO

Testes foram feitos com cores sólidas: cinza, vinho, vermelho, amarelo, verde, azul claro, azul escuro, roxo, rosa e marrom escolhidas ao acaso, porém de modo que o primeiro plano principal possui todos os pixels com o mesmo tom de cinza $rgb(127, 127, 127)$, o segundo plano principal possui o mesmo tom de vinho $rgb(136, 0, 21)$ em todos seus pixels e assim, o raciocínio segue analogamente para os outros oito planos principais. Optou-se por fazer uma reconstrução nessas condições e obteve-se os resultados apresentados na Fig. 5 apenas como teste inicial. É possível observar que com a mesma quantidade de planos principais (dez) e de planos intermediários (seis), obteve-se diferentes resultados com as técnicas propostas.

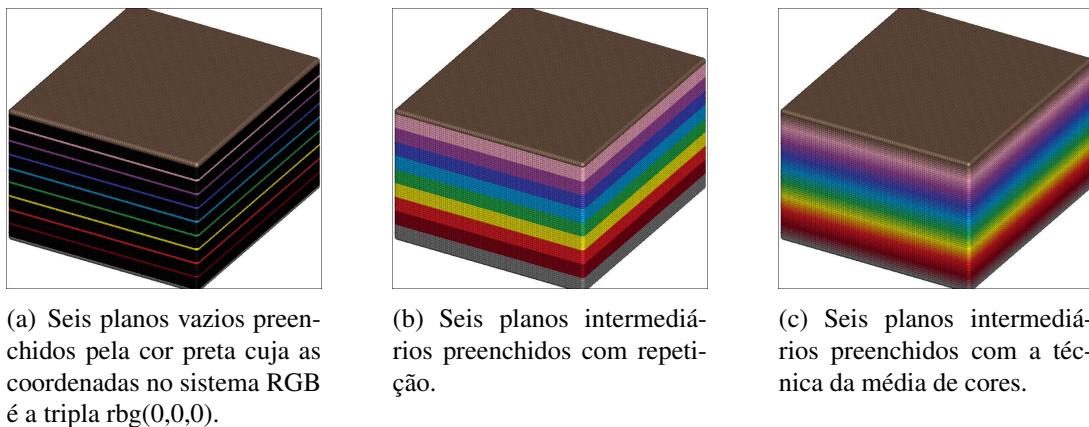


Figura 5- Teste das abordagens que foram implementadas no código desenvolvido.

Nas mesmas condições, testes foram feitos com as imagens de Ferro Fundido Nodular e Aço Duplex. Os resultados obtidos, respectivamente, podem ser vistos na Fig. 6.

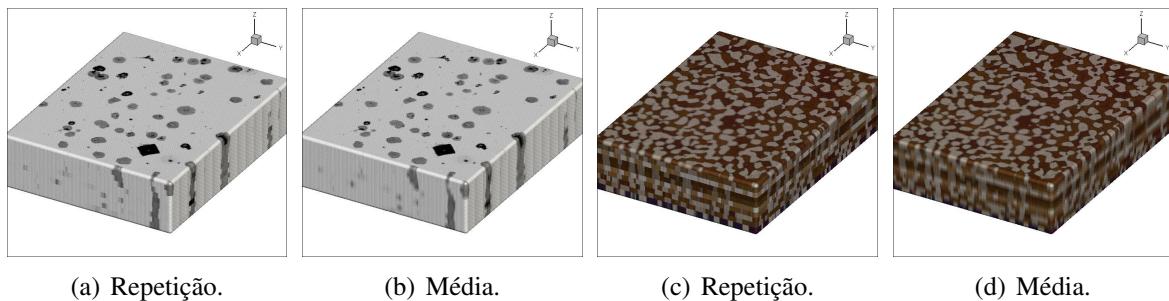


Figura 6- Resultados com Ferro Fundido Nodular e Aço Duplex.

Na Fig. 7 e na Fig. 8 a reconstrução da microestrutura completa é realizada por meio das duas abordagens apresentadas neste trabalho e, após este procedimento, a área de interesse é escolhida. Pode-se observar, através da Fig. 9, detalhes ampliados da imagem que mostram a diferença da borda da microestrutura quando utiliza-se as diferentes técnicas de obtenção de planos intermediários.

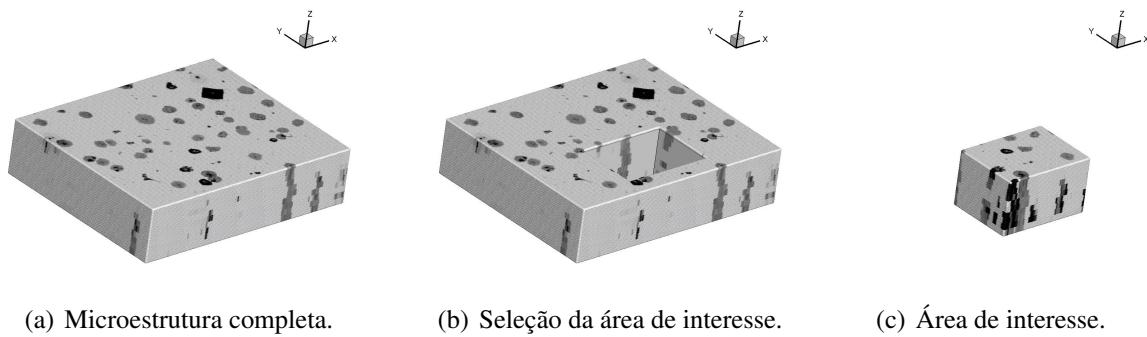


Figura 7- Resultado da técnica de repetição de planos adjacentes.

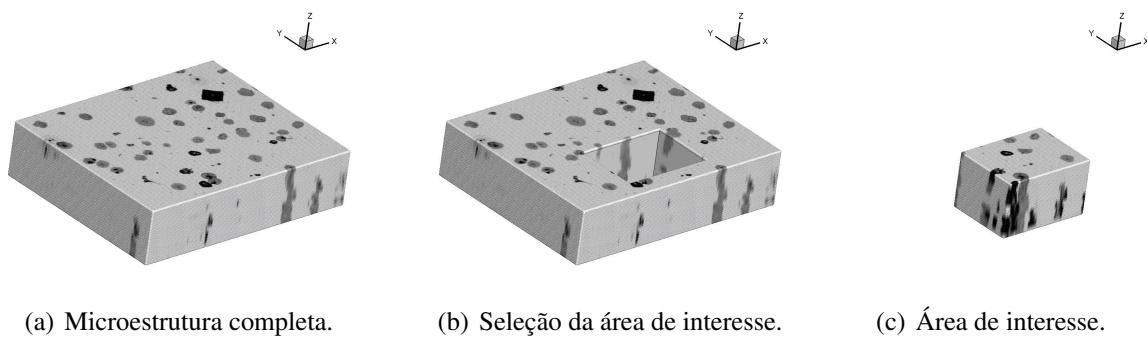


Figura 8- Resultado da técnica de obtenção de planos intermediários por meio da médias das cores.

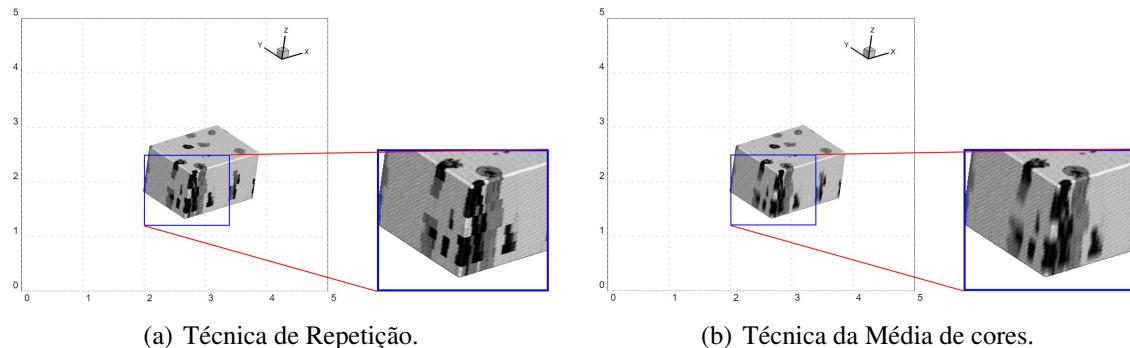


Figura 9- Área de interesse ampliada para mostrar a diferença entre as duas técnicas abordadas.

5. CONCLUSÕES

A escolha de uma linguagem de programação que possibilita empregar o paradigma de orientação a objeto permitiu que códigos de autores fossem utilizados para dar sequência ao desenvolvimento deste software. Logo, permitirá que outros autores prossigam neste trabalho com maior facilidade quando for preciso incorporar novas funcionalidades ao programa.

A reconstrução de uma microestrutura fica mais suave com a utilização da técnica de obtenção de planos intermediários por meio da média das cores do que com a utilização da técnica de repetição de planos. E, com base nos resultados obtidos, o domínio computacional gerado já pode ser aplicado em simulações computacionais tais como Transformação de Fase, Transferência de Calor e Deformação a partir dos métodos Autômato Celular, Cone Causal, Monte Carlo, Elementos Finitos, Diferenças Finitas, dentre outros. Este trabalho ainda está em de-

senvolvimento. Com isso, testes estão sendo feitos e os resultados gerados até o momento são coerentes com o que se espera e com o trabalho de outros autores.

Agradecimentos

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Os autores agradecem o apoio da Pró-Reitoria de Assuntos Estudantis (PROAES) da Universidade Federal Fluminense (UFF), do Mestrado em Modelagem Computacional em Ciência e Tecnologia (MCCT), do Núcleo de Modelamento Microestrutural (NMM). Agradecem também o EAMC e ao Laboratório Nacional de Computação Científica (LNCC) pela oportunidade de publicação deste trabalho.

REFERÊNCIAS

- Chiaverini, V. (1990). *Aços e Ferros Fundidos*. Publicação da Associação Brasileira de Metalurgia e Materiais, São Paulo, 6 edition.
- Copalert, H. (2008). *Metalografia dos Produtos Siderúrgicos Comuns*. Edgard Blücher, São Paulo, 4 edition.
- de Freitas, A. F. P. (2008). Metalografia quantitativa tridimensional dos ferros fundidos nodular e cíngulo. *Dissertação de Mestrado em Engenharia Metalúrgica da Escola de Engenharia Industrial Metalúrgica de Volta Redonda da Universidade Federal Fluminense*.
- Ganti, S., Turner, B., Kirsch, M., Anthony, D., McCoy, B., Trivedi, H., and Sundar, V. (2018). Three-dimensional (3d) analysis of white etching bands (webs) in aisi m50 bearing steel using automated serial sectioning. *Materials Characterization*, 138:1118.
- Intel Corporation, Willow Garage, I. (2000). OpenCV (open source computer vision library). Acessado em 08/2017.
- Li, M., Ghosh, S., Rouns, T. N., Richmond, H. W. O., and Hunt, W. (1998). Serial sectioning method in the construction of 3d microstructures for particlereinforced mmcs. *Materials Characterization*, 41:81–95.
- Madej, L., Mojzeszko, M., Chraponski, J., Roskosz, S., and Cwajna, J. (2017). Digital material representation model of porous microstructure based on 3d reconstruction algorithm. *Archives of Metallurgy and Materials*, 62(2):563–569.
- Spowart, J. E., Mullens, H. E., and Puchala, B. T. (2003). Collecting and analyzing microstructures in three dimensions: a fully automated approach. *JOM*, 55:3537.
- Tecplot, I. (2010). *Users Manual*.
- Tewari, A. and Gokhale, A. M. (2000). Application of three-dimensional digital image processing for reconstruction of microstructural volume from serial sections. *Materials Characterization*, 44:259–269.
- Wunsch, B. and Chawla, N. (2003). Serial sectioning for 3d visualization and modeling of sic particle reinforced aluminum composites. *TMS Outstanding Student*, pages 1–16.
- Zankel, A., Wagner, J., and Poelt, P. (2014). Serial sectioning methods for 3d investigations in materials science. *Micron*, 62:66–78.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

ESCOAMENTO BIFÁSICO COM GRAVIDADE EM UM MEIO HETEROGÊNEO DE DOIS TIPOS DE ROCHA

Isamara Landim Nunes Araujo¹ - isamara-landim@hotmail.com

Panters Rodríguez Bermúdez¹ - pantersrb@id.uff.br

Yoisell Rodríguez Núñez¹ - yoisellrn@id.uff.br

¹Programa de Pós Graduação MCCT – Universidade Federal Fluminense

Resumo. Neste trabalho estudamos o escoamento vertical de dois fluidos imiscíveis, água e óleo, em um meio poroso composto por duas rochas com permeabilidades distintas. Este fenômeno pode ser modelado por uma lei de conservação escalar não linear com função fluxo descontínua. As soluções analíticas entrópicas para esta lei de conservação, com dados iniciais de Riemann, podem ser obtidas considerando uma extensão do critério de entropia de Oleinik, junto com uma escolha adequada de conexão entre os fluxos f_l , f_r (Kaasschieter, 1999; Andreianov and Cancès, 2013). Do ponto de vista numérico, utilizamos o método de diferenças finitas Lagrangian-Eulerian desenvolvido por Abreu e Pérez (Perez, 2015), baseado em uma abordagem Lagrangeana-Euleriana, e o método clássico Lax-Friedrichs que permitem lidar de maneira eficiente com fluxos descontínuos. Os resultados obtidos por ambos os métodos são comparados quanto a precisão para distintos dados de Riemann e regimes de escoamento. As soluções de Riemann para este problema, em geral, são compostas de ondas de choque, rarefações ou combinações destas, e um choque estacionário em $z = 0$, o qual representa a conexão entre as funções fluxos f_l e f_r . As soluções numéricas encontradas pelos métodos Lagrangian-Eulerian e Lax-Friedrichs aproximam de maneira eficiente, nos casos estudados, quando comparadas com as soluções analíticas entrópicas obtidas em (Kaasschieter, 1999).

Palavras-Chaves: Leis de Conservação, Diferenças Finitas, Método Lagrangian-Eulerian, Escoamento Bifásico, Meio Heterogêneo.

1. INTRODUÇÃO

As leis de conservação surgem em uma ampla variedade de modelos, principalmente na Física e Engenharia, que descrevem fenômenos de transporte. O mais interessante sobre estas equações é que quando estas são não-lineares suas soluções comumente são descontínuas ou com derivadas descontínuas embora o dado inicial seja suave; logo o conceito de solução clássica deve ser estendido, dando lugar à formulação integral e às chamadas soluções fracas. Aqui, no entanto, surge uma das principais dificuldades e questões na teoria das leis de conservação: para um determinado problema de valor inicial várias soluções fracas são possíveis, mas somente uma é a solução fisicamente correta chamada de entrópica, de forma que condições adicionais são necessárias para selecionar esta solução (LeVeque, 1992; van Duijn, 2002).

Estas condições adicionais são conhecidas como critérios de entropia. Existem diversos critérios de entropia, tanto para equações escalares quanto para sistemas. Um comumente utilizado é o critério do perfil viscoso, o qual consiste em incluir um pequeno termo difusivo εu_{xx} (viscoso) na equação, de forma que a condição que define a solução entrópica é que esta deve ser limite quando $\varepsilon \rightarrow 0$ das soluções da equação com viscosidade (van Duijn, 2002).

Uma lei de conservação com dados iniciais constantes por partes, com uma descontinuidade, é conhecido como Problema de Riemann. As soluções entrópicas de Riemann para leis de conservação com função de fluxo contínua, em geral, contêm ondas de choque, ondas de rarefações ou combinações destas (LeVeque, 1992). Para fluxos com descontinuidade espacial, comumente há também um choque estacionário na descontinuidade (Kaasschieter, 1999; Andreianov and Cancès, 2013).

Para leis de conservação escalares com função de fluxo contínua o critério de entropia Oleinik pode ser utilizado para se obter as soluções analíticas de Riemann (Oleinik, 1957b,a; Rodríguez-Bermúdez and Marchesin, 2013). Já para funções de fluxo descontínuas, em geral, não existe um critério de entropia simples para se obter as soluções analíticas de Riemann, embora para alguns casos particulares de funções de fluxo existam trabalhos relevantes sobre este assunto (Kaasschieter, 1999; Andreianov and Cancès, 2013).

Do ponto de vista numérico, existem diversos métodos conservativos clássicos de diferenças finitas, volumes finitos e elementos finitos capazes de obter a solução entrópica de forma eficiente para leis de conservação escalares e sistemas com função de fluxo contínua, porém existem poucos resultados teóricos de convergência destes métodos para os casos com função de fluxo descontínua. Por este motivo e pela facilidade de se considerar um termo fonte ou estender o modelo para um caso bidimensional posteriormente, os métodos numéricos possuem uma maior importância.

O método Lagrangian-Eulerian é um esquema localmente conservativo baseado em uma estrutura Lagrangeana-Euleriana. Este esquema é de alta resolução para leis de conservação escalares não lineares, sendo capaz de obter a solução numérica de maneira eficiente mesmo quando estas leis possuem função de fluxo descontínua (Abreu et al., 2018). O método Lagrangian-Eulerian também mostrou evidências de ser “well-balanced”, o qual pode ser muito útil se adicionarmos futuramente um termo fonte, pois ele preserva o equilíbrio entre o termo fonte e o fluxo (Perez, 2015; Abreu and Perez, 2014a,b; Abreu et al., 2017).

Juntamente com o método Lax-Friedrichs, estes são os únicos esquemas de diferenças finitas para o qual foi provada convergência para a solução entrópica para os casos com função de fluxo descontínua (Abreu et al., 2018). Por este motivo, utilizamos estes métodos para resolver o problema do escoamento bifásico em um meio heterogêneo. E comparamos os resultados obtidos por estes métodos.

Neste trabalho estudamos o escoamento em um meio poroso, composto de duas rochas com permeabilidades distintas, de dois fluidos imiscíveis, água e óleo. Consideramos que o escoamento ocorre somente na direção vertical, estando assim sob o efeito da força de gravidade, de forma que este fenômeno pode ser modelado por uma lei de conservação não linear com função de fluxo descontínua. Para mais detalhes veja (Araujo et al., 2018).

2. ESCOAMENTO BIFÁSICO COM GRAVIDADE EM DUAS ROCHAS

Consideramos o escoamento vertical de dois fluidos imiscíveis, água e óleo, em um meio poroso composto de duas rochas, veja a Figura 1, sob o efeito da força gravitacional.

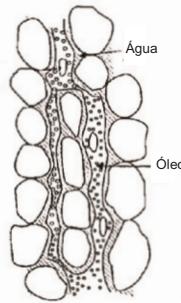


Figura 1- Escoamento bifásico em um meio poroso.

Fonte: (Quintella, 2012).

Assumimos que não há termo fonte, a temperatura, densidades, viscosidades e porosidade são constantes. E negligenciamos os efeitos de compressibilidade e pressão capilar, portanto utilizamos as seguintes equações para modelar o escoamento:

$$\frac{\partial}{\partial t}(\phi s_i) + \frac{\partial}{\partial z}v_i = 0, \quad i = w, o, \quad (1)$$

$$v_i = -K \frac{k_i}{\mu_i} \left(\frac{\partial p_i}{\partial z} - \rho_i g e_z \right), \quad i = w, o. \quad (2)$$

A equação (1) expressa a conservação de massa, onde ϕ é a porosidade, s_i , v_i são a saturação e velocidade da fase i , respectivamente. A velocidade v_i é determinada pela lei de Darcy (2), na qual K é a permeabilidade absoluta da rocha, g é a constante gravitacional, k_i , μ_i , p_i e ρ_i são a permeabilidade relativa, viscosidade, pressão e densidade dos fluidos, respectivamente.

Para prosseguir com o desenvolvimento do modelo assumimos que o meio poroso é totalmente saturado $s_w + s_o = 1$, que as permeabilidades são funções da saturação e que o meio poroso é heterogêneo, isto é $K = K(z)$.

Agora definimos:

$$K(z) = \begin{cases} K_l, & \text{se } z < 0, \\ K_r, & \text{se } z > 0. \end{cases} \quad (3)$$

Restringimos a análise para o modelo com permeabilidades quadráticas, $k_i = s_i^2$. Logo este escoamento é modelado pela seguinte lei de conservação:

$$\frac{\partial s_w}{\partial t} + \frac{\partial}{\partial z}(f(s_w, z)) = 0, \quad (4)$$

com função fluxo

$$f(s_w, z) = \begin{cases} f_l, & \text{se } z < 0, \\ f_r, & \text{se } z > 0, \end{cases} \quad (5)$$

onde

$$f_l = \frac{s_w^2}{s_w^2 + \mu(1 - s_w)^2} [v + (1 - s_w)^2 \mu(1 - \rho)], \quad (6)$$

e

$$f_r = \frac{s_w^2}{s_w^2 + \mu(1-s_w)^2} \left[v + (1-s_w)^2 \mu(1-\rho) \frac{K_r}{K_l} \right], \quad (7)$$

aqui denotamos $\mu = \mu_w/\mu_o$ e $\rho = \rho_o/\rho_w$, s_w é a saturação da água. O parâmetro v está relacionado aos gradientes de pressão.

3. MÉTODOS DE DIFERENÇAS FINITAS

Os métodos de diferenças finitas são muito utilizados na resolução de equações diferenciais, sendo muito úteis para leis de conservação.

Para preservar a propriedade de conservação das soluções os métodos são utilizados em sua forma conservativa (Kröner, 1997).

$$U_j^{n+1} = U_j^n - \frac{k}{h} (G(U_j^n, U_{j+1}^n) - G(U_{j-1}^n, U_j^n)). \quad (8)$$

Onde $U_j^n = U(x_j, t^n)$, $h = x_{j+1} - x_j$ e $k = t^{n+1} - t^n$. A função G é chamada de fluxo numérico.

O esquema Lax-Friedrichs (LF) em sua forma conservativa (8) possui o seguinte fluxo numérico para leis de conservação com função fluxo descontínua:

$$G(U_{j-1}^n, U_j^n) = \frac{1}{2} (f(U_{j-1}^n, x_{j-1}) + f(U_j^n, x_j)) - \frac{1}{2\nu} (U_j^n - U_{j-1}^n), \quad (9)$$

onde

$$\nu = k/h. \quad (10)$$

Este esquema deve satisfazer a seguinte condição CFL (LeVeque, 2007)

$$\max_j |f'(U_j^n, z_j)| \frac{k}{h} \leq 1. \quad (11)$$

O método Lagrangian-Eulerian (LEH) em sua forma conservativa possui o seguinte fluxo numérico:

$$G(U_j^n, U_{j+1}^n) = \frac{1}{4} \left[\frac{h}{k} (U_j^n - U_{j+1}^n) + 2(f(U_{j+1}^n, x_{j+1}) + f(U_j^n, x_j)) \right] \quad (12)$$

e a condição CFL para este esquema é (Perez, 2015; Abreu and Perez, 2014a,b):

$$\max_j |f'(U_j^n, -j)| \frac{k}{h} \leq \frac{\sqrt{2}}{2}. \quad (13)$$

4. SOLUÇÕES NUMÉRICAS E ANALÍTICAS

Neste trabalho resolvemos problemas de Riemann com dados iniciais em (14) para a lei de conservação (4) com função fluxo (5–7), onde o escoamento é puramente gravitacional, isto é, $v = 0$. Selecionamos parâmetros diferentes, de modo que apresentaremos a seguir quatro problemas distintos. Para todos os casos usamos CFL= 0, 7.

$$s_w(z, 0) = \begin{cases} s_l, & z \leq 0, \\ s_r, & z > 0. \end{cases} \quad (14)$$

- Exemplo 1: neste caso escolhemos $\mu = 1/3$ e $\rho = 3/2$ (óleo pesado). As permeabilidades são $k_l = 0,5$ e $k_r = 1$, de forma que $f_l > f_r$.
- Exemplo 2: novamente temos o caso com óleo pesado ($\rho = 3/2$) e selecionamos $\mu = 1/3$. Escolhemos as permeabilidades de forma que $f_r > f_l$, portanto podemos utilizar $k_l = 1$ e $k_r = 0,5$.
- Exemplo 3: para este caso definimos os fluidos como sendo óleo leve e água, portanto $\rho = 0,8$ e $\mu = 1/3$. As rochas possuem permeabilidade $k_l = 0,5$ e $k_r = 1$ ($f_r > f_l$).
- Exemplo 4: finalmente para o último caso estudado, definimos novamente $\rho = 0,8$ e $\mu = 1/3$. Neste caso as permeabilidades das rochas são $k_l = 1$ e $k_r = 0,5$, portanto $f_l > f_r$.

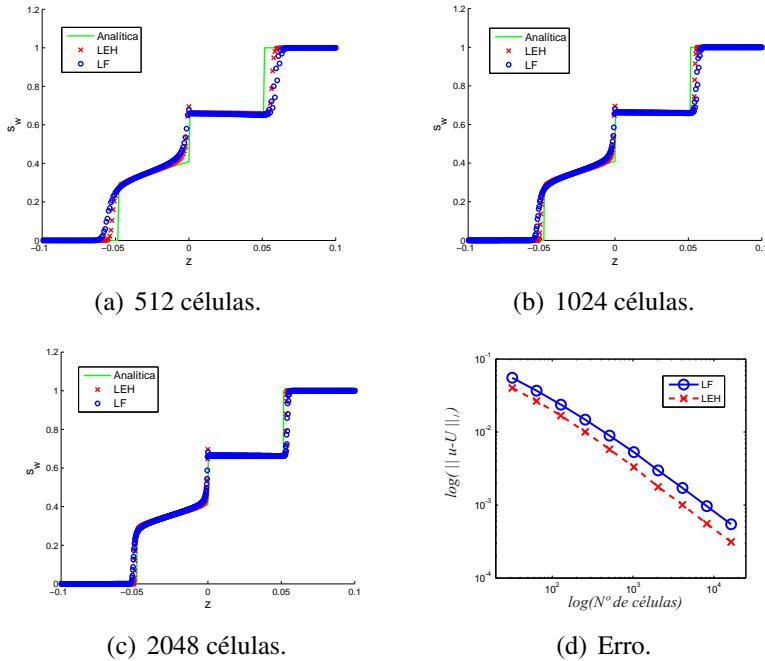


Figura 2- Exemplo 1 para $t = 0,5$ com dados de Riemann $s_l = 0$, $s_r = 1$.

A seguir explicaremos a solução para o exemplo 1 com dados de Riemann $s_l = 0$ e $s_r = 1$, veja as Figuras 2(a)–2(c), a solução é composta de uma onda de choque com velocidade negativa

conectando os estados $s_l = 0$ e $s_1 = 0,298$, seguida de uma onda de rarefação também com velocidade negativa conectando os estados $s_1 = 0,298$ e $S_L = 0,4095$; ambas viajam para o topo do reservatório. Posteriormente há um choque estacionário em $z = 0$ conectando os estados $S_L = 0,4095$ e $S_R = 0,6659$, e por último uma onda de choque com velocidade positiva conectando os estados $S_R = 0,6659$ e $s_r = 1$ viajando para o fundo do reservatório. Já na Figura 2(d) é mostrado o comportamento do erro quando o número de células aumenta: através deste resultado notamos que os erros dos métodos LEH e LF são da mesma ordem, e diminui quando a malha é refinada. Notamos também que o esquema LEH tem maior precisão, visto que este teve um menor erro.

Nas Figuras 3, 4 e 5 são mostradas as soluções do restante dos exemplos.

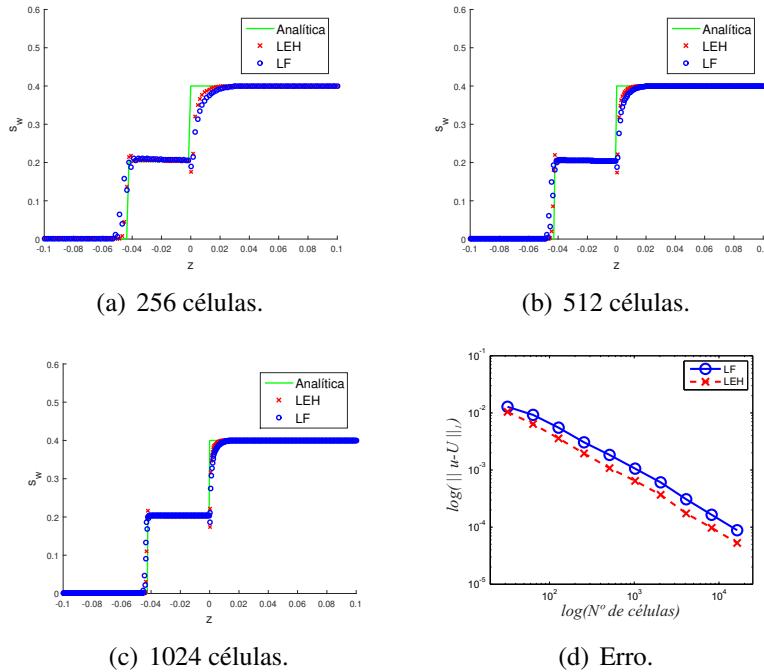


Figura 3- Exemplo 2 para $t = 0,5$ com dados de Riemann $s_l = 0$, $s_r = 0,4$.

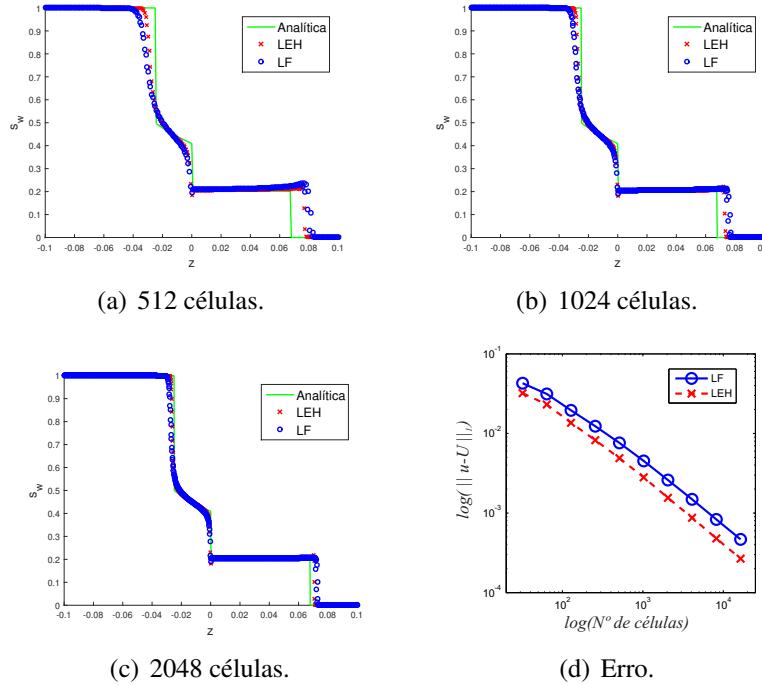


Figura 4- Exemplo 3 para $t = 1$ com dados de Riemann $s_l = 1$, $s_r = 0$.

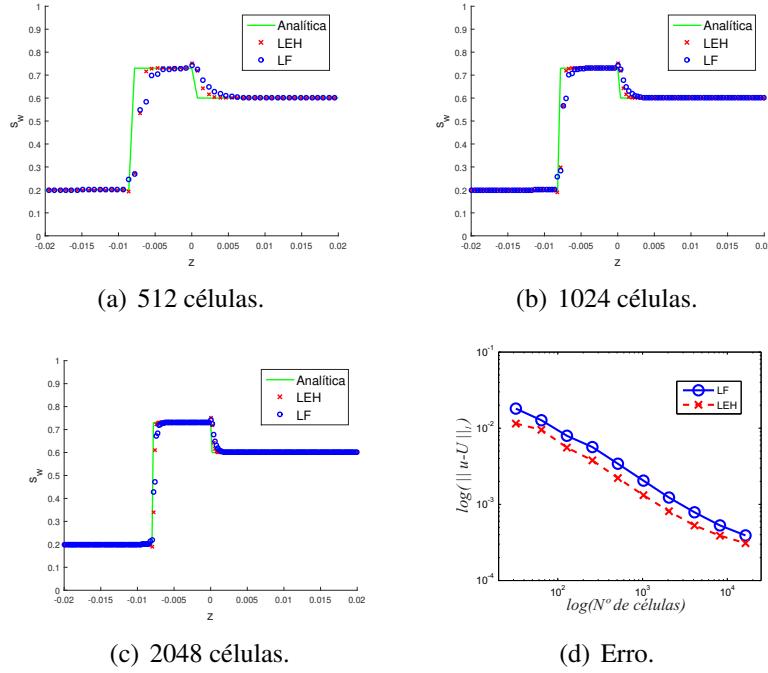


Figura 5- Exemplo 4 para $t = 2$ com dados de Riemann $s_l = 0, 2$, $s_r = 0, 6$.

5. CONCLUSÕES

Neste trabalho fazemos um estudo numérico da lei de conservação que modela o escoamento bifásico com gravidade em duas rochas com permeabilidades distintas, escolhendo diferentes

parâmetros de forma a resolver quatro tipos de problemas de Riemann.

De acordo com os resultados apresentados notamos que os métodos Lagrangian-Eulerian e Lax-Fridrichs foram capazes de obter a solução entrópica de forma eficiente, como pode ser comprovado ao se comparar as soluções numéricas com as analíticas obtidas via critério de entropia desenvolvido em (Kaasschieter, 1999). Notamos também que, em malhas mais refinadas as soluções numéricas obtidas são muito mais precisas, como era de se esperar, o que pode ser verificado ao se comparar os erros calculados.

Em todos os casos apresentados o método Lagrangian-Eulerian mostrou ter maior precisão do que o Lax-Friedrichs, sendo menos difusivo e apresentando um menor erro.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Abreu, E., Lambert, W., Perez, J., and Santo, A. (2017). A new finite volume approach for transport models and related applications with balancing source terms. *Math. Comput. Simulation*, 137:2–28.
- Abreu, E., Matos, V., Perez, J., and Rodriguez-Bermudez, P. (2018). A shock-capturing and high-resolution lagrangian-eulerian method for first order hyperbolic problems with forcing terms. Submetido.
- Abreu, E. C. and Perez, J. A. (2014a). A lagrangian-eulerian algorithm scheme for hyperbolic conservation laws and balance laws. *HYP2014- XV International Conference on Hyperbolic Problems*. <http://www.hyp2014.impa.br>. Acesso em 31 de agosto de 2018.
- Abreu, E. C. and Perez, J. A. (2014b). A new locally conservative lagrangian eulerian method for hyperbolic and balance laws. *VIII Pan-American Workshop Applied and Computational Mathematics*,. <http://www.csic.sdsu.edu/panam2014/index.html>.
- Andreianov, B. and Cancès, C. (2013). Vanishing capillarity solutions of buckley-leverett equation with gravity in two-rocks medium. *Comput. Geosci.*, 17(3):551–572.
- Araujo, I. L. N., Rodriguez-Bermudez, P., and Rodriguez-Nunez, Y. (2018). Numerical study for two-phase flow with gravity in homogeneous and piecewise heterogeneous porous media. Submetido.
- Kaasschieter, E. (1999). Solving the buckley leverett equation with gravity in a heterogeneous porous medium. *Comp. Geosci.*, 3:23–48.
- Kröner, D. (1997). *Numerical Schemes for Conservation Laws*. John Wiley Sons, Stuttgart.
- LeVeque, R. J. (1992). *Numerical Methods for Conservation Laws*. Birkhauser, Basel; Boston; Berlin, second edition.
- LeVeque, R. J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM.
- Oleinik, O. (1957a). Discontinuous solutions of nonlinear differential equations. *Amer. Math. Soc. Trans. Ser.*, 2(26):95–172.
- Oleinik, O. A. (1957b). On the uniqueness of generalized solution of cauchy problem for non linear system of equations occurring in mechanics. *Uspekhi Mat. Nauk*, (12):169–176.

- Perez, J. A. (2015). *Lagrangian-Eulerian approximation methods for balance laws and hyperbolic conservation laws*. Ph.d. thesis.
- Quintella, J. R. F. (2012). Simulação da injeção alternada água-emulsão como processo de recuperação avançada de petróleo. Master's thesis, PUC-Rio.
- Rodríguez-Bermúdez, P. and Marchesin, D. (2013). Riemann solutions for vertical flow of three phases in porous media: simple cases. *Journal of Hyperbolic Differential Equations*, 10(2):335–370.
- van Duijn, C. J. (2002). *An Introduction to Conservation Laws: Theory and Applications to Multi-Phase Flow*. Lecture Notes, Delft University of Technology.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

SELEÇÃO DE MODELOS PARA O PROBLEMA DE INCRUSTAÇÃO EM TROCADORES DE CALOR

Renato S. Silva¹ - rssr@lncc.br

Carmen Lúcia Corrêa Bonifácio² bonicarmen@gmail.com

Fernando Henrique Pereira Cardozo³ - fernando_cardozo@poli.ufrj.br

¹ Laboratório Nacional de Computação Cienífica - LNCC

² Faculdade de Educação Tecnológica do Estado Rio de Janeiro - FAETERJ

³ Univ. Federal do Rio de Janeiro - UFRJ

Resumo. A incrustação em trocadores de calor é um problema sério em diferentes tipos de indústrias motivando o surgimento de diversos modelos numéricos para e previsão da ocorrência da incrustação. As características químicas e mecânicas dos trocadores de calor têm tornado a tarefa da escolha do modelo um processo bastante delicado. Para a escolha do melhor modelo são necessários o conhecimento das hipóteses e os limites de aplicabilidade cada um. A maneira usual para a escolha de um modelo em detrimento de outro é comparar seus desempenhos a partir de um conjunto de dados similares à situação para a qual se deseja a previsão. Assim podemos escolher o modelo com melhor desempenho em um determinado critério. Neste trabalho usamos o método ABC-SMC (Approximate Bayesian Computation - Sequential Monte Carlo) pois este método permite tanto a obtenção da distribuição a posteriori dos modelos quanto a distribuição a posteriori dos parâmetros desses modelos.

Palavras Chaves: Incrustação, ABC-SMC, Seleção de Modelos

INTRODUÇÃO

A incrustação em trocadores de calor ocorre devido a deposição de cristais de sais de substâncias presentes no fluido de refrigeração. Um fator determinante para desenvolvimento de incrustações é a concentração dos sais acima da solubilidade de saturação. Esse fenômeno é agravado com sais formados por íons de Cálcio (Ca_{+2}) e Magnésio (Mg_{+2}) que possuem solubilidade inversa em relação ao aumento de temperatura (Sheikholeslami, 1984). A condutividade térmica dos depósitos de sais é muito menor que a do metal de construção do equipamento. Por isso, quanto maior a espessura da camada de incrustação, maior a resistência térmica da camada, reduzindo a eficiência de troca térmica, assim como aumentando o custo operacional do trocador de calor.

Na literatura existem diversos modelos matemáticos desenvolvidos para explicar o fenômeno do aparecimento e crescimento da incrustação. Dentre eles podemos citar: Kern and Seaton, 1964 - Sheikholeslami, 1984 - Bohent , 1987 - Babuška, Silva e Actor entre outros (Bohnet, 1987),(Müller-Steinhagen, 2011),(Sheikholeslami, 1984),(Babuška *et al.* , 2018).

A escolha do melhor modelo para cada situação exige conhecimento detalhado sobre as variáveis analisadas e os fatores limitantes. A maneira usual da escolha de um modelo em detrimento de outros é comparar o desempenho dos modelos com um conjunto de dados similares ao da situação que se deseja a previsão. Assim podemos escolher o modelo com melhor desempenho em um determinado critério. Entretanto devido à existência de uma grande variabilidade nas condições de operação e variáveis associadas, dificilmente um único modelo será o ideal em todas as situações. Neste trabalho é investigado o uso da inferência Bayesiana no contexto da seleção de modelos através de um método do tipo *Approximate Bayesian Computation* (ABC) para os modelos de incrustação em trocadores de calor com dados experimentais.

MODELOS DE INCRUSTAÇÃO

Os modelos de crescimento de incrustação em trocadores de calor podem ser escritos de uma forma geral como:

$$\frac{dm_f}{dt} = \dot{m}_d - \dot{m}_r, \quad (1)$$

em que m_f é a massa da camada de incrustação, \dot{m}_d é a massa depositada e \dot{m}_r é a massa removida. Modelos diferentes são construídos a partir de diferentes definições dos processos físicos. Neste trabalho consideraremos os modelos linear e o de Kern e Seaton (Babuška *et al.*, 2018).

Modelo Linear

Modelos lineares podem ser encontrados na literatura para descrever a camada de incrustação que apresenta uma dureza elevada. Consideramos aqui o modelo linear que é baseado no processo de transferência de massa, da solução para a superfície do tubo, e que assume somente a existência do processo de deposição, que é função apenas do excesso de $CaCO_3$ existente na água. Este modelo é definido como:

$$\dot{m}_d^L = K_1(c_F - c_s), \quad (2)$$

em que c_F é a concentração de $CaCO_3$ no fluido e c_s é a concentração de saturação do $CaCO_3$.

O Modelo de Kern e Seaton

O modelo de Kern e Seaton foi apresentado em 1959 (Kern & Seaton, 1959). Este modelo foi construído a partir da observação de dados reais em que as curvas de resistência térmica apresentavam comportamento assintótico. Portanto, a fim de representar este comportamento, o modelo de Kern e Seaton apresenta os termos de deposição e o de remoção.

O termo de deposição, \dot{m}_d^{KS} , é proporcional a velocidade do fluido (u), sendo definido como:

$$\dot{m}_d^{KS} = K_2 u c_F. \quad (3)$$

O termo de remoção é função da força de cisalhamento (τ) causada pelo fluido. Entretanto, assume-se que as partículas são removidas em blocos aleatórios que estejam mais fracamente

ligados aos demais, e que as partes mais fracas ocorrem em qualquer profundidade com a mesma probabilidade. Com essas suposições o termo de remoção é definido como:

$$\dot{m}_r^{KS} = K_3 \tau \delta , \quad (4)$$

em que δ é o tamanho atual da camada de incrustação.

Para os modelos descritos, cabe ressaltar que os parâmetros K_1 , K_2 e K_3 são constantes que devem ser calibradas para o cenário considerado.

SELEÇÃO DE MODELOS VIA ABC

A seleção de modelos é um tópico importante na ciência preditiva, especialmente quando existem vários modelos com físcas diferentes que podem descrever o fenômeno que os dados experimentais representam. Quando não são conhecidos os parâmentros dos modelos essa escolha é ainda mais complexa, aumentando a incerteza relativa à escolha do modelo.

Seja m_i , $i = 1, \dots, n$, um conjunto de modelos parametrizados por $\theta_i \in \Theta$, e x_0 um conjunto de dados. A inferência Bayesiana nos permite escrever a probabilidade *a posteriori* de cada modelo representar os dados como:

$$\pi(m_i, \theta_i | x_0) \approx \pi(x_0 | m_i, \theta_i) \pi(m_i | \theta_i) \pi(\theta_i) , \quad (5)$$

em que $\pi(x_0 | m_i, \theta_i)$ é a função de verossimilhança, $\pi(m_i | \theta_i)$ é a probabilidade *a priori* do modelo m_i dado θ_i , e $\pi(\theta_i)$ é a probabilidade *a priori* dos parâmetros θ_i do modelo m_i . Entretanto para obter a probabilidade do modelo m_i dado o conjunto de dados x_0 é necessário marginalizar a Eq.5 obtendo-se:

$$\pi(m_i | x_0) = \int_{\Theta} \pi(m_i, \theta_i | x_0) d\theta . \quad (6)$$

Em muitos casos a função de verossimilhança não é conhecida ou é muito complexa para ser avaliada. Para contornar estas dificuldades foram criados os métodos de calibração *Approximate Bayesian Computation* (ABC) (T. & M.P., 2009). Nesta metodologia, a avaliação da função de verossimilhança é substituída pela avaliação de uma distância entre os dados experimentais e os valores simulados, aqui denotados por x^* (Turner & Zandt, 2012). Isto significa que os métodos ABC adotam

$$\pi(x_0 | m_i, \theta_i) \approx \pi(x_0 | d(x_0, x^*) < \varepsilon), \quad (7)$$

de modo que quanto menor ε melhor será a aproximação. A distância utilizada neste trabalho é apresentada pela seguinte equação

$$d(x^*, x_0) = \sum (x^* - x_0)^2 . \quad (8)$$

De posse da aproximação da distribuição *a posteriori*, a Eq.6 pode ser integrada pelo Método de Monte Carlo resultando em

$$\pi(m_i | x_0) \propto \frac{\# \text{modelos } m_i \text{ aceitos}}{\text{total de modelos aceitos}} . \quad (9)$$

Neste trabalho utilizamos o método ABC-SMC (T. & M.P., 2009) apresentado no Algoritmo 1. Adotamos probabilidades *a priori* uniformes tanto para os modelos quanto para os parâmetros. As tolerâncias foram calculadas utilizando um percentil pré determinado da população anterior.

Algoritmo 1 ABC-SMC

```

1: procedure ABC-SMC
2:   Inicializar  $\varepsilon_1, \dots, \varepsilon_T$ 
3:    $t = 0, i = 1$ 
4:   Amostrar  $m^*$  de  $\pi(m)$ 
5:   if  $t=0$  then
6:     Amostre  $\theta^{**}$  de  $\pi(\theta(m^*))$ 
7:   else
8:     Amostre  $\theta^*$  da população  $t - 1 \{(\theta(m^*))_{t-1}\}$  com pesos  $w(m^*)_{t-1}$ 
9:     Pertube  $\theta^*$  para obter  $\theta^{**} \sim K_t(\theta|\theta^*)$ 
10:    if  $\pi(\theta^{**}) = 0$  then
11:      retorna para 4
12:    end if
13:    end if
14:    Simular  $x^* \sim f(x|\theta^{**}, m^*)$ 
15:    if  $d(x^*, x_0) \geq \varepsilon_t$  then
16:      retorna a 4
17:    else
18:       $m_t^{(i)} = m^*$ 
19:      Adicione  $\theta^{**}$  a população  $\{(\theta(m^*))_t\}$ 
20:      Calcule o peso como:
21:      
$$w_t^{(i)} = \begin{cases} 1 & \text{If } t = 0, \\ \frac{\pi(\theta^{**})}{\sum_{j=1}^N w_{t-1}^{(i)} K_t(\theta_{t-1}^{(j)}|\theta^{**})} & \text{If } t > 0 \end{cases}$$

22:    end if
23:    If  $i < N$ ,  $i = i + 1$ , retorna 4
24:    Normalizar os  $w$  para todo  $m$ 
25:    If  $t < T$ ,  $t = t + 1$ , retorna 3
26: end procedure

```

RESULTADOS

Utilizamos os dados experimentais *Run 20* e *Run 23* extraídos de (Sheikholeslami, 1984). Numa primeira etapa, com o intuito de verificar o método ABC, calibraramos os modelos linear e o modelo KS com o método de otimização de Luus and Jaakola (LJ) (Silva Neto *et al.*, 2016), minimizando o valor da raiz quadrada do erro médio (RMSE). Os valores dos parâmetros e o RMSE obtidos são apresentados na Tab.1 e as curvas geradas são apresentadas na Fig.1.

Para a utilização do método ABC-SMC em ambos os casos utilizamos cinco populações com 800 membros em cada. A tolerância inicial $\varepsilon_0 = 5.0 \times 10^{-6}$ é preestabelecida e os demais valores são obtidos do percentil de aproximadamente 16% da população anterior.

Tabela 1- Parâmetros dos modelos calibrados via LJ e RSME.

	Parâmetros			RSME	
	Linear		KS	Linear	KS
	K_1	K_2	K_3		
Run 20	0.3074E-05	0.5486E-05	0.4289E-01	0.7633E-05	0.7049E-05
Run 23	0.9377E-05	0.4335E-04	0.2621	0.4190E-04	0.1905E-04

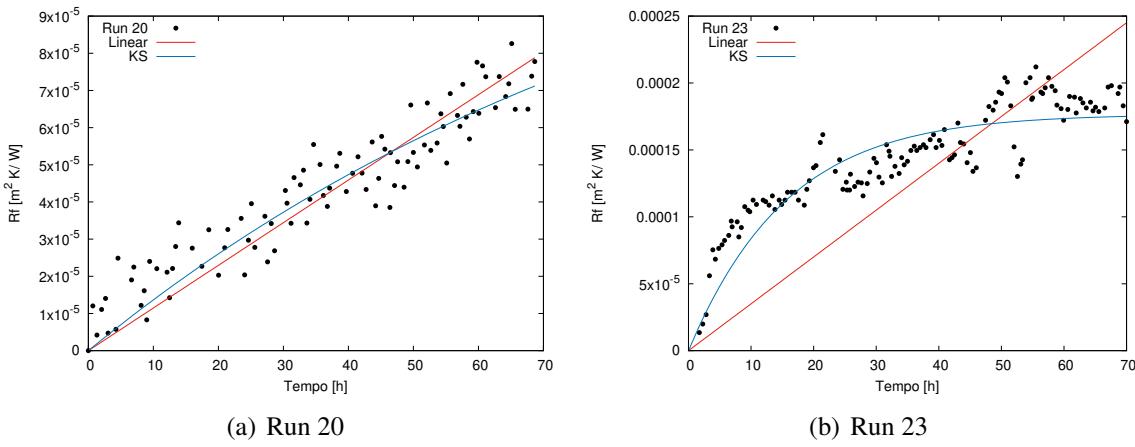


Figura 1- Modelos calibrados via o método LJ.

Run 20

Na Fig.2(a) apresentamos as simulações dos modelos calibrados com o método ABC-SMC. Observando a Fig.2(a) e a Tab.2, podemos ver que ambos os modelos apresentam um bom ajuste em relação aos pontos experimentais. A Fig.2(b) apresenta a evolução dos valores das tolerâncias ε_i utilizadas pelo ABC-SMC.

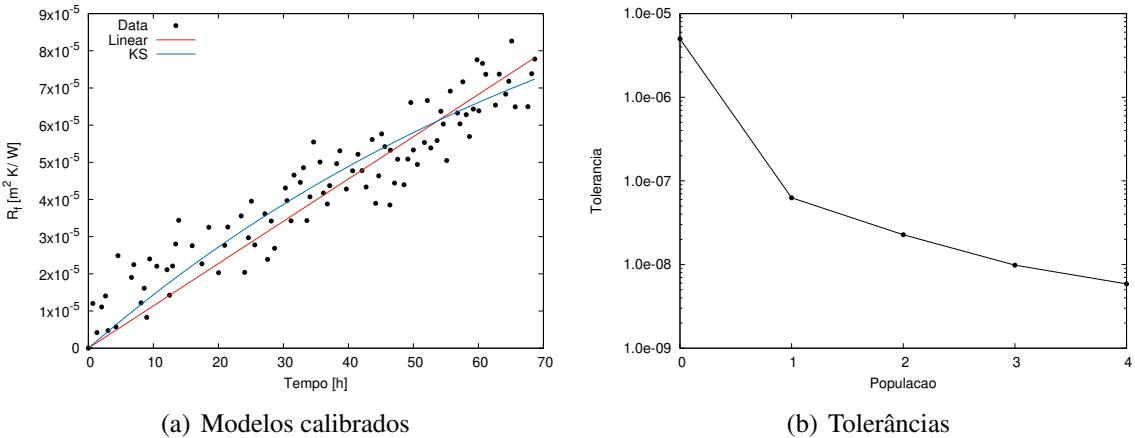
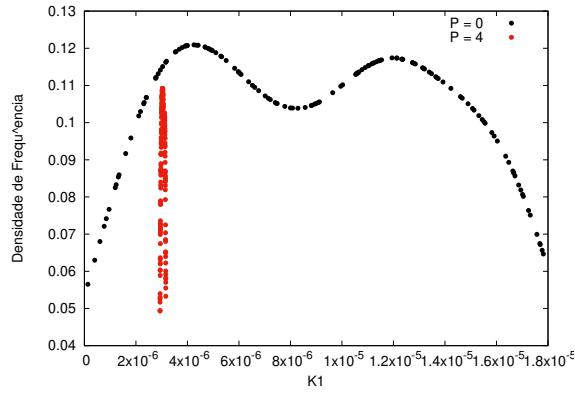


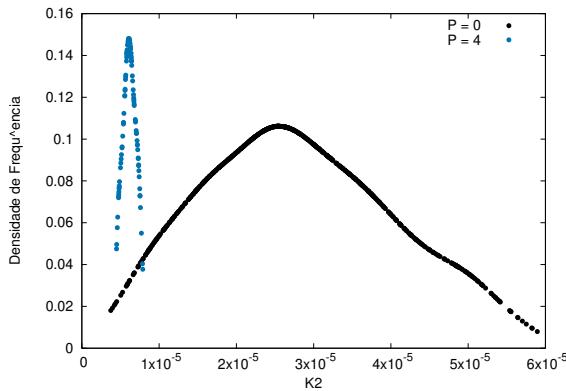
Figura 2- Resultados para Run 20.

A Figura 3 apresenta as estimativas de densidade do kernel para a população inicial ($P=0$) e para população final ($P=4$), para cada parâmetro dos modelos. Nota-se que não só há convergência do valor esperado como a redução significativa da largura da variabilidade.

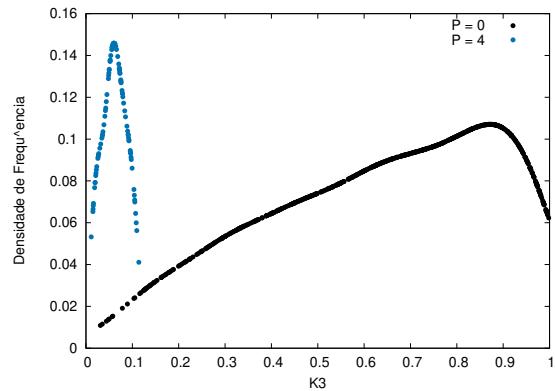
A Fig.4 apresenta a evolução da distribuição $\pi(m_i|x_0)$. Apesar de ambos os modelos aproximarem bem os dados, a distribuição $\pi(m_i|x_0)$ ainda não está representando este fato, não permitindo distinguir os modelos. Este comportamento indica que o número de populações e/ou o número de pontos não é suficiente para representar $\pi(m_i|x_0)$, características associadas ao ABC-SMC que não são conhecidas *a priori*.



(a) Modelo Linear (K_1)

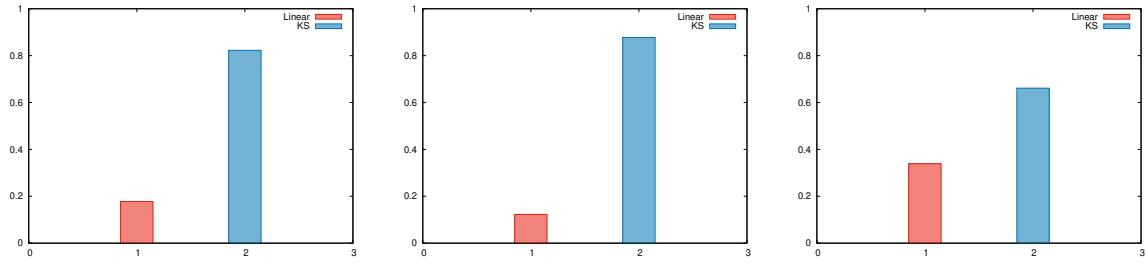


(b) Modelo KS (K_2)



(c) Modelo KS (K_3)

Figura 3- Densidade de Kernel inicial e final para os parâmetros dos modelos para Run 20.



(a) População 0

(b) População 1

(c) População 2

(d) População 3

(e) População 4

Figura 4- Evolução de aproximação da $\pi(m_i|x_0)$ para Run 20.

Run 23

Com este conjunto de dados e com o modelo linear não foi possível obter parâmetros que satisfizessem a tolerância igual ou menor 0.927×10^{-7} e portanto o método ABC descartou o modelo linear. Isto significa que o modelo linear não foi capaz de representar a física descrita pelos dados, enquanto que somente o modelo KS é capaz de representá-los.

Na Fig.5(a) apresentamos a simulação obtida com o método ABC-SMC para o modelo KS e os respectivos parâmetros estimados são apresentados na Tabela 2. A Fig.5(b) apresenta os valores das tolerâncias ε_i utilizadas.

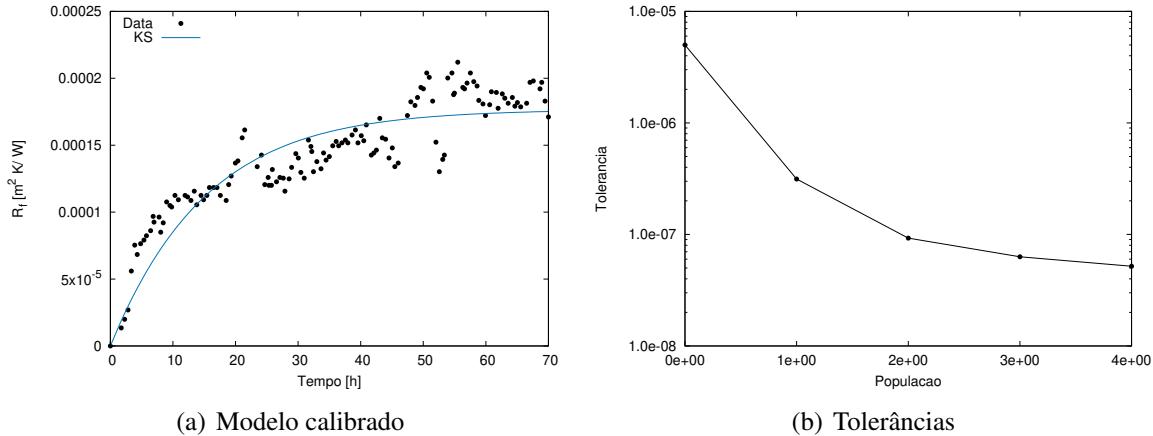
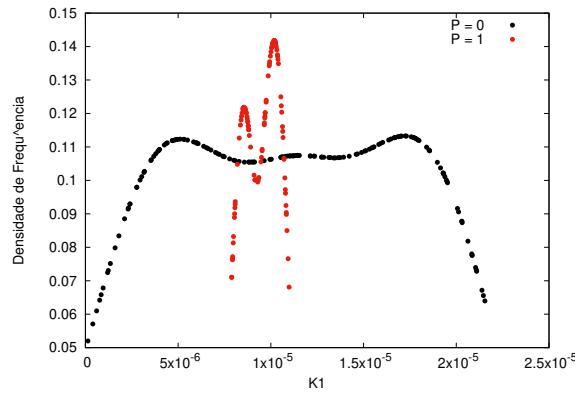


Figura 5- Resultados para Run 23.

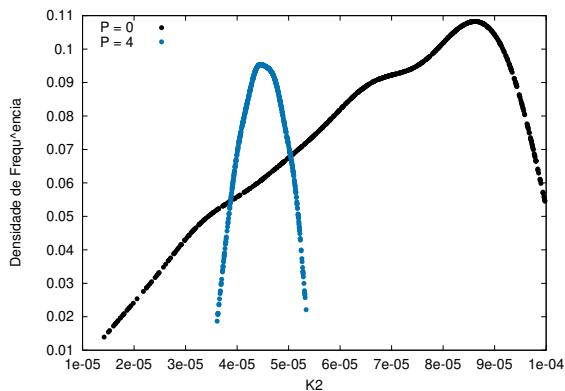
A Fig. 6 apresenta as estimativas de densidade do kernel para a população inicial ($P=0$) e para a população final ($P=4$), no caso de KS, e para a população $P=1$, no caso do modelo linear. A Fig.7 apresenta a evolução da distribuição $\pi(m_i|x_0)$ em relação ao aumento do número de populações.

Tabela 2- Parâmetros calibrados dos modelos via ABC-SMC.

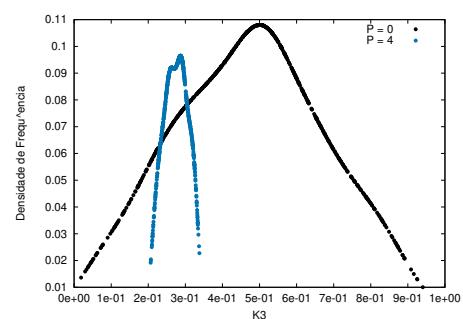
	Parâmetros		
	Linear		KS
	K_1	K_2	K_3
Run 20	0.3048E-05	0.5802E-05	0.4835E-01
Run 23	-	0.4359E-04	0.2636



(a) Modelo Linear (K_1)



(b) Modelo KS (K_2)



(c) Modelo KS (K_3)

Figura 6- Densidade de Kernel inicial e final para os parâmetros dos modelos para Run 23.

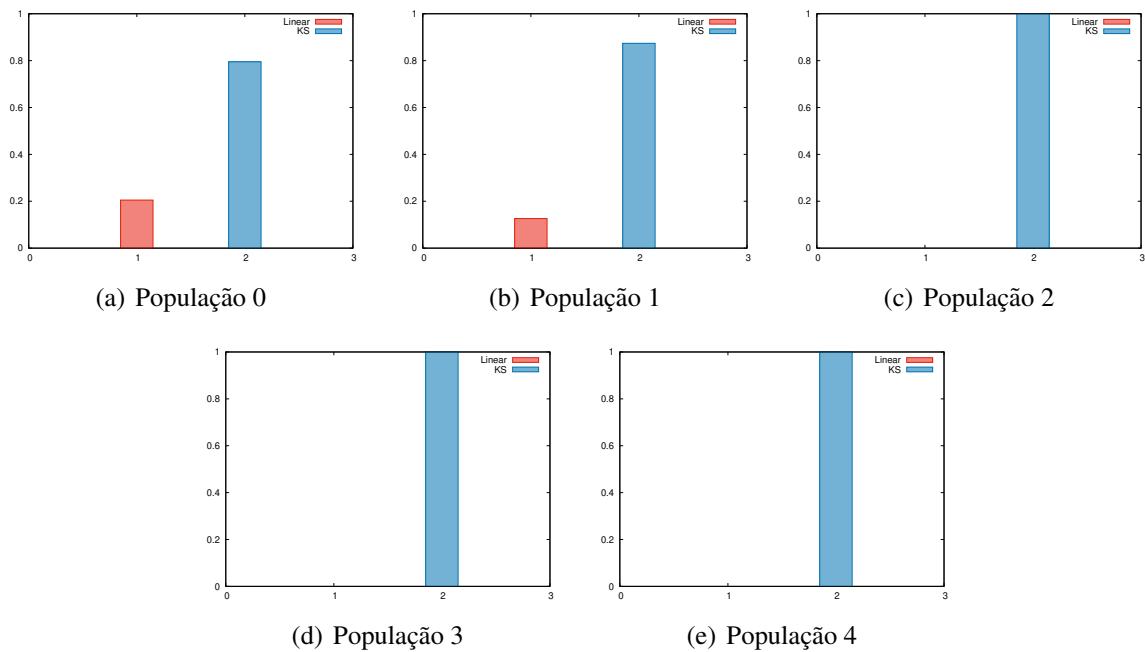


Figura 7- Evolução de aproximação da $\pi(m_i|x_0)$ para Run 23.

CONCLUSÕES

O métdo ABC-SMC é útil na determinação do melhor modelo, tendo sido capaz de selecionar o modelo KS para os dados da Run 23. Para Run 20, selecionou os dois, não fazendo distinção entre eles, o que está de acordo com o resultado da Tabela 1 obtida com um método de otimização. Nota-se também que os parâmentros encontrados pelo método ABC estão bem próximos aos obtidos via o método de otimização. Entretanto trabalhos futuros serão necessários para uma avaliação mais precisa dos custos computacionais envolvidos no método ABC para a seleção de modelos.

Agradecimentos

O presente trabalho foi realizado com apoio do programa PIBIC/CNPq - LNCC/MCTI.

REFERÊNCIAS

- Babuška, I., Silva, R. S. and Actor, J. 2018. Break-off model for CaCO₃ fouling in heat exchangers. *International Journal of Heat and Mass Transfer*, **116**, 104 – 114.
- Bohnet, M. W. 1987. Fouling of Heat Transfer Surfaces. *Chemical Engineering Tecnology*, **10**, 113–125.
- Kern, D. Q. and Seaton, R. E. 1959. A theoretical analysis of thermal surface fouling. *British Chemical Engineering*, **4**(5), 258–262.
- Steinhagen, M., 2011. Heat Transfer Fouling: 50 Years After the Kern and Seaton Model. *Heat Transfer Engineering*, **32**(1), 1–13.
- Sheikholeslami, R. 1984. *Tube Material and Augmented Surface Effects in Heat Exchanger Scaling*. M.Phil. thesis, The University of British Columbia.
- Silva Neto, A. J., Becceneri J. C. and Campos Velho, H. F. (eds). 2016. *Inteligência computacional aplicada a problemas inversos em transferência radiativa*. UERJ.
- Toni, T. and Stumpf, M. P. 2009. Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, **26**(1), 104–110.
- Turner, B. M., and Zandt, T. V. 2012. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, **56**(2), 69–85.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

ANÁLISE COMPARATIVA DE MÉTODOS NUMÉRICOS APLICADOS AO FLUXO DE CARGA NO SOFTWARE MATLAB®

Diogo Cardoso Lima¹ - cardoso_diogo@yahoo.com.br

Diovani de Souza Montovani² - diomontovani@gmail.com

¹Pontifícia Universidade Católica do Rio de Janeiro – Departamento de Engenharia Elétrica

²Universidade do Estado do Rio de Janeiro – Faculdade de Engenharia

Resumo. Os estudos elétricos envolvendo fluxo de potência em sistemas elétricos são desenvolvidos, sobretudo, para auxiliar o planejamento e os reforços da rede existente, a minimização de perdas, e para traçar estratégias visando a economicidade da operação do sistema. Ao longo do tempo, diferentes técnicas foram desenvolvidas para obtenção das soluções envolvendo as equações e inequações não lineares que representam o modelo de rede, de modo que se obtêm ao fim de sucessivas iterações as tensões e ângulo nodais. O presente artigo visa apresentar uma comparação entre os métodos de Gauss-Seidel, Gauss-Seidel acelerado e Newton-Raphson implementados no software MATLAB® aplicados à solução de fluxo de carga em redes elétricas.

Palavras-chave: Sistemas elétricos, Fluxo de potência, Gauss-Seidel, Newton-Raphson

1. INTRODUÇÃO

A análise de fluxo de carga em redes elétricas é essencial para sua correta operação e planejamento, e permite avaliar, entre outros aspectos:

- A geração necessária para suprir a carga e as perdas inerentes ao sistema;
- As tensões em cada barra, avaliando se as tensões encontram-se dentro de faixas aceitáveis;
- Se os geradores operam dentro de seus limites;
- Quantificar as potências ativa e reativa que fluem nas linhas de transmissão e nos transformadores.

Para tanto, sua formulação emprega um conjunto de equações algébricas, dado que o processo ocorre em regime estacionário, e não-lineares, devido aos parâmetros associados a própria da rede, além de algumas restrições que devem ser atendidas durante o processo iterativo. Segundo (Stott, 1974), apenas na segunda metade da década de 50, que as primeiras abordagens para o cálculo de fluxo de carga tornaram-se atrativas para serem implementadas computacionalmente.

2. IMPLEMENTAÇÃO COMPUTACIONAL DOS MÉTODOS

Os métodos implementados neste trabalho recebem inicialmente como dados de entrada as informações do sistema, tais como os dados de barra que incluem o nome, tipo, limites de tensão, valores especificados de módulo e ângulo das tensões nodais, potência ativa e reativa gerada, e potência ativa e reativa drenada pelas cargas. Os dados de linha e transformadores incluem as conexões, a capacidade do circuito (em MVA), sua impedância e o ajuste do *tap*. Cada barra pertence a um grupo bem definido, onde:

- Barra do tipo 1 (PV) – Denominada barra de tensão controlada. São especificados o módulo da tensão $|\dot{V}_k|$ e a potência ativa P_k , e ao final da solução do fluxo de potência obtém-se θ_k e Q_k ;
- Barra do tipo 2 (PQ) – Denominada barra de carga, são as mais comuns. São definidas para estas barras as variáveis P_k e Q_k , e ao final da simulação computam-se as variáveis $|\dot{V}_k|$ e θ_k ;
- Barra do tipo 3 ($V\theta$) – Denominada barra *swing* (ou *slack*), representa a barra de referência para o sistema, convenientemente especificada com o subíndice s . Para esta barra são definidos o módulo e ângulo da tensão, e ao final do processo iterativo obtém-se as variáveis P_s e Q_s .

Deste modo, em síntese, cada barra está associado às variáveis $|\dot{V}_k|$, θ_k , P_k , e Q_k , sendo para cada barra, conhecida duas variáveis. A experiência em simulações deste tipo demonstram que a convergência pode ser acelerada em muitos casos adotando-se como estimativa inicial $|\dot{V}_k|^{(0)} = 1,0$ [pu] para as barras do tipo PQ, e $\theta_k^{(0)} = 0^\circ$ para as barras do tipo PQ e PV (*flat start*). As variáveis P_k^{esp} e Q_k^{esp} representam, respectivamente, as potências ativa e reativa líquidas injetadas na barra k , de tal sorte que, considerando a Fig. 1, obtém-se:

$$P_k^{\text{esp}} = P_{G_k} - P_{D_k} \quad (1)$$

$$Q_k^{\text{esp}} = Q_{G_k} - Q_{D_k} \quad (2)$$

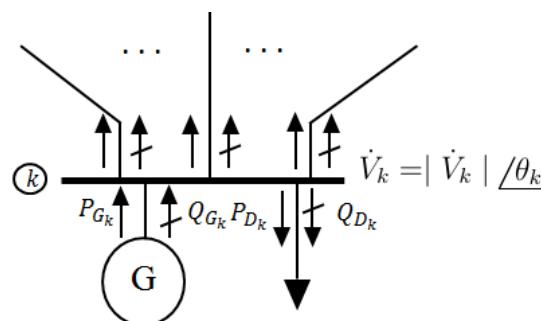


Figura 1- Variáveis associadas à barra k .

Testa-se a convergência do método comparando o máximo resíduo encontrado com uma tolerância pré-especificada.

Durante a execução do processo iterativo deseja-se que as tensões das barras PV sejam

mantidas nos valores especificados. No entanto, pode-se verificar a partir de um determinado número de iterações, se:

$$Q_{G_k}^{\min} \leq Q_{G_k}^{\nu} \leq Q_{G_k}^{\max} \quad (3)$$

Caso a violação ocorra, substitui-se $Q_{G_k}^{\nu}$ pelo limite máximo ou mínimo violado, e trata-se a partir de então a barra como do tipo PQ, fazendo $|\dot{V}_k|$ variar durante as iterações seguintes.

Para as barras de carga, deseja-se que a tensão esteja dentro de uma faixa tolerável:

$$|\dot{V}_k|^{\min} \leq |\dot{V}_k|^{\nu} \leq |\dot{V}_k|^{\max} \quad (4)$$

Caso a violação ocorra, substitui-se $|\dot{V}_k|^{\nu}$ pelo limite máximo ou mínimo violado, e trata-se a partir de então a barra como do tipo PV. Segundo (Ramos & Dias, 1983), alguns programas desenvolvidos podem retornar as barras ao tipo anterior durante o processo, mas esta possibilidade foi abandonada nesta simulação.

2.1 Método de Gauss-Seidel

O método de Gauss-Seidel representa um aprimoramento do método de Gauss, pois emprega, assim que obtida, a estimativa mais recente na iteração atual. Deste modo, considera-se a corrente \dot{I}_k injetada na barra k , conforme ilustrado na Fig. 2:

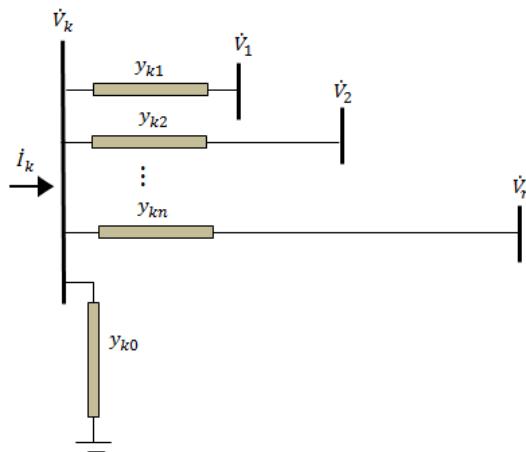


Figura 2- Corrente injetada na barra k .

onde:

$$\dot{I}_k = \frac{P_k - jQ_k}{\dot{V}_k^*} = \dot{V}_k \sum_{\substack{m=0 \\ m \neq k}}^n y_{km} - \sum_{\substack{m=1 \\ m \neq k}}^n y_{km} \dot{V}_m \quad (5)$$

Tendo em vista a matriz quadrada, e em geral esparsa, de admitância nodal \mathbf{Y} , construída de tal modo que $Y_{km} = -y_{km}$ e $Y_{kk} = \sum_{m=0}^n y_{km}$, a Eq. (5) adaptada ao método iterativo de Gauss-

Seidel torna-se, isolando a tensão \dot{V}_k :

$$x_k^{\nu+1} = g(x_1^{\nu+1}, \dots, x_{k-1}^{\nu+1}, x_k^\nu, x_{k+1}^\nu, \dots, x_n^\nu) \Rightarrow \dot{V}_k^{\nu+1} = \frac{1}{Y_{kk}} \left(\frac{P_k - jQ_k}{(\dot{V}_k^\nu)^*} - \sum_{m=1}^{k-1} Y_{km} \dot{V}_m^{\nu+1} - \sum_{m=k}^n Y_{km} \dot{V}_m^\nu \right) \quad (6)$$

Assume-se que o método converge quando o valor absoluto das estimativas para todas as barras, entre as sucessivas iterações, é menor ou igual a tolerância especificada, ou seja, $|x_k^{\nu+1} - x_k^\nu| \leq \epsilon$. Para as barras de tensão controlada, calcula-se iterativamente a potência reativa verificando a condição imposta pela Eq. (3).

Devido a sua formulação, o fraco acoplamento entre as barras do sistema impõe que a estimativa atual de uma barra possui efeito local, impactando diretamente nas barras a ela conectadas. Isto se traduz em uma lenta convergência, especialmente em sistemas de grande porte (Stott, 1974).

Contudo, o processo iterativo pode ser substancialmente acelerado através do fator α ($\alpha > 1$), fazendo:

$$x_k^{\nu+1} = x_k^\nu + \alpha [g(x_1^{\nu+1}, \dots, x_{k-1}^{\nu+1}, x_k^\nu, x_{k+1}^\nu, \dots, x_n^\nu) - x_k^\nu] \Rightarrow \dot{V}_k^{\nu+1} = \dot{V}_k^\nu + \alpha \Delta \dot{V}_k^\nu \quad (7)$$

A escolha adequada de α , obtido empiricamente para cada caso, em geral produz uma taxa de convergência duas vezes maior, ou mais. Por outro lado, uma escolha indiscriminada de α pode levar o método a divergência. Em (Elgerd, 1976), recomenda-se valor de α entre 1,5 e 1,7 para estudos desta natureza.

2.2 Método de Newton-Raphson

O método de Newton-Raphson inicializa-se resolvendo o sub-sistema 1 contendo $2N_{PQ} + N_{PV}$ equações, sendo N_{PQ} o número de barras PQ, e N_{PV} o número de barras PV (Monticelli, 1983). Calculam-se os resíduos de potência ativa para as barras PQ e PV:

$$\Delta P_k^\nu = P_k^{\text{esp}} - P_k^{\text{calc}^\nu} = P_k^{\text{esp}} - |\dot{V}_k|^\nu \sum_{m=1}^n |\dot{V}_m|^\nu (G_{km} \cos \theta_{km}^\nu + B_{km} \sin \theta_{km}^\nu) \quad (8)$$

e os resíduos de potência reativa para as barras PQ:

$$\Delta Q_k^\nu = Q_k^{\text{esp}} - Q_k^{\text{calc}^\nu} = Q_k^{\text{esp}} - |\dot{V}_k|^\nu \sum_{m=1}^n |\dot{V}_m|^\nu (G_{km} \sin \theta_{km}^\nu - B_{km} \cos \theta_{km}^\nu) \quad (9)$$

Testa-se a convergência do método comparando o máximo resíduo encontrado com uma tolerância especificada, e busca-se que $\max \{\Delta P_k^\nu\} \leq \epsilon_P$ e $\max \{\Delta Q_k^\nu\} \leq \epsilon_Q$. Caso algum resíduo viole esta condição, calcula-se a matriz jacobiano de dimensão $(2N_{PQ} + N_{PV}) \times (2N_{PQ} + N_{PV})$.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial P}{\partial \underline{P}} & \frac{\partial P}{\partial \underline{Q}} \\ \frac{\partial Q}{\partial \underline{Q}} & \frac{\partial Q}{\partial \underline{V}} \end{bmatrix} \quad (10)$$

onde:

- $\frac{\partial \underline{P}}{\partial \underline{\theta}}$ é a partição de \mathbf{J} de dimensão $(N_{PQ} + N_{PV}) \times (N_{PQ} + N_{PV})$;
- $\frac{\partial \underline{P}}{\partial |\dot{\underline{V}}|}$ é a partição de \mathbf{J} de dimensão $(N_{PQ} + N_{PV}) \times (N_{PV})$;
- $\frac{\partial \underline{Q}}{\partial \underline{\theta}}$ é a partição de \mathbf{J} de dimensão $(N_{PQ}) \times (N_{PQ} + N_{PV})$;
- $\frac{\partial \underline{Q}}{\partial |\dot{\underline{V}}|}$ é a partição de \mathbf{J} de dimensão $(N_{PQ}) \times (N_{PV})$;

o incremento no vetor de estado será dado por:

$$\Delta \underline{x}^\nu = -[\mathbf{J}(\underline{x}^\nu)]^{-1} \underline{f}(\underline{x}^\nu) \Rightarrow \begin{bmatrix} \Delta \underline{\theta} \\ \Delta |\dot{\underline{V}}| \end{bmatrix}^\nu = \begin{bmatrix} \frac{\partial \underline{P}}{\partial \underline{\theta}} & \frac{\partial \underline{P}}{\partial |\dot{\underline{V}}|} \\ \frac{\partial \underline{Q}}{\partial \underline{\theta}} & \frac{\partial \underline{Q}}{\partial |\dot{\underline{V}}|} \end{bmatrix}^{\nu-1} \begin{bmatrix} \Delta \underline{P}^\nu \\ \Delta \underline{Q}^\nu \end{bmatrix} \quad (11)$$

atualiza-se então o vetor de estado, fazendo:

$$\underline{x}^{\nu+1} = \underline{x}^\nu + \Delta \underline{x}^\nu \Rightarrow \begin{bmatrix} \underline{\theta} \\ |\dot{\underline{V}}| \end{bmatrix}^{\nu+1} = \begin{bmatrix} \underline{\theta} \\ |\dot{\underline{V}}| \end{bmatrix}^\nu + \begin{bmatrix} \Delta \underline{\theta} \\ \Delta |\dot{\underline{V}}| \end{bmatrix}^\nu \quad (12)$$

verifica-se a condição (3), e em caso de violação, Q_k é embutido no subsistema 1 e altera-se a dimensão da matriz jacobiano.

Após a solução do subsistema 1, o programa inicializa o subsistema 2 de dimensão $N_{PV} + 2$ equações, calculando a potência ativa na barra de referência (*swing*), e a potência reativa nas barras PV e de referência. Após este passo, pode-se calcular os fluxo nas linhas e transformadores, as perdas do sistema, entre outros dados que se julgarem pertinentes.

3. RESULTADOS

Os métodos descritos foram implementados no *software* MATLAB®, considerando sistemas de diversas dimensões encontrados na literatura. Considerou-se para todos os casos o critério de violação estipulado em (3) através dos dados limites de geração de potência reativa. Em sistemas em que tais informações não estão disponíveis, adotou-se $Q_G^{\min} = -9,999$ [pu] e $Q_G^{\max} = 9,999$ [pu].

Os resultados apresentados nas Tab. 1 e 2, demonstram que o fator de aceleração α , tomado na simulação como 1,7 apresentou satisfatório ganho computacional em relação ao método de Gauss-Seidel para a maioria dos casos. Contudo, o método de Gauss-Seidel acelerado não foi capaz de convergir para o sistema IEEE 300 barras, refletindo que a escolha de $\alpha = 1,7$ não é adequada para este sistema.

Por sua vez, o método de Newton-Raphson apresenta um número de iterações praticamente constante, e sua superioridade pode ser verificada através do aumento gradativo das dimensões e da complexidade do sistema investigado, o que justifica sua aplicação em sistemas reais de grande porte.

Tabela 1- Número de iterações requeridas para convergência

Sistema	Número de iterações		
	Newton-Raphson	Gauss-Seidel	Gauss-Seidel Acelerado [†]
Stagg & El-Albiad	3	54	48
WSCC 9 barras	3	131	60
IEEE 14 barras	3	161	53
IEEE 30 barras	4	392	62
New England	3	1219	223
IEEE 57 barras	4	531	93
Nordic	5	5055	1031
IEEE 118 barras	5	2323	482
IEEE 300 barras	5	18201	∞

¹ Sistema 5 barras extraído de (Stagg & El-Albiad, 1968)

² Sistema 39 barras extraído de (Ajjarapu, 2006)

³ Sistema 74 barras extraído de (IEEE Power & Energy Society, 2015)

† Fator de aceleração $\alpha = 1,7$

Tabela 2- Tempo de execução para convergência dos métodos

Sistema	Tempo de simulação [s]		
	Newton-Raphson	Gauss-Seidel	Gauss-Seidel Acelerado [†]
Stagg & El-Albiad	0,094	0,054	0,059
WSCC 9 barras	0,100	0,077	0,076
IEEE 14 barras	0,110	0,078	0,074
IEEE 30 barras	0,121	0,148	0,089
New England	0,116	0,396	0,219
IEEE 57 barras	0,132	0,265	0,127
Nordic	0,406	2,696	0,703
IEEE 118 barras	0,336	2,126	0,655
IEEE 300 barras	1,413	50,461	∞

A fim de demonstrar o comportamento de acomodação em torno da solução de tensão pelos métodos utilizados, a Fig. 3 ilustra a convergência na solução da magnitude de tensão da barra de carga #1 do sistema Nordic em 0,9988 [pu].

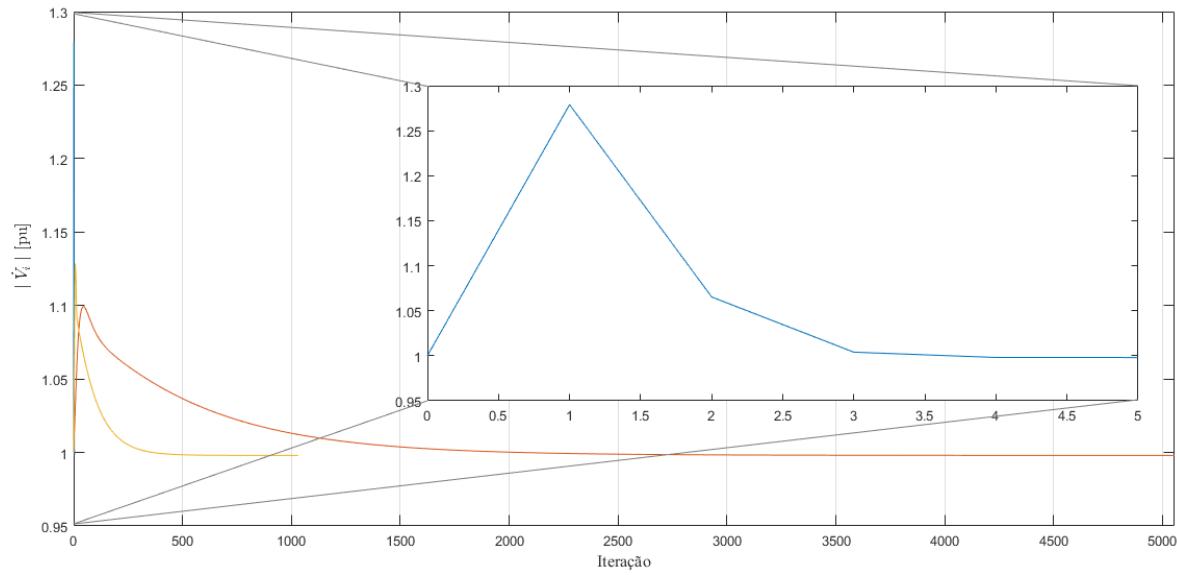


Figura 3- Convergência da tensão da barra #1 no sistema Nordic.

4. CONCLUSÃO

A partir dos resultados apresentados neste artigo, verifica-se que o fator de aceleração α introduz um aperfeiçoamento ao método de Gauss-Seidel, oferecendo um tempo de simulação 3,84 vezes menor para o pior dos casos analisados em que se constatou a convergência de ambos. Com efeito, uma escolha inadequada de α pode fornecer resultados insatisfatórios de não convergência, como observado no último caso para o sistema IEEE 300 barras. Portanto, a necessidade de um estudo prévio para sua determinação torna-o impraticável.

Em contrapartida, o método de Newton-Raphson, embora apresente maior complexidade e exija maior custo computacional, oferece um tempo de simulação aceitável para todos os casos, demonstrando que o método é insensível às dimensões do sistema.

Ademais, o método fornece informações relevantes através dos elementos da matriz jacobiano que podem ser utilizados para outros estudos, como em índices de avaliação das condições de estabilidade de tensão.

Referências

- [1] Ajjarapu, V. 2006. *Computational techniques for voltage stability assessment and control*. Springer, Iowa.
- [2] Elgerd, O. I. 1976. *Introdução à teoria de sistemas de energia elétrica*. McGraw-Hill Ltda, São Paulo.
- [3] IEEE Power & Energy Society. 2015. *Test systems for voltage stability analysis and security assessment*. pp. 1 – 161.
- [4] Monticelli, A. 1968. *Fluxo de carga em redes de energia elétrica*. Editora Edgard Blücher Ltda, São Paulo.
- [5] Ramos, D. S. & Dias, E. M. 1983. *Sistemas elétricos de potência – Regime permanente*, vol. 2. Editora Guanabara Dois S. A., Rio de Janeiro.

- [6] Stagg, G. W. & El-Abiad, A. H. 1968. *Computer methods in power system analysis*. McGraw-Hill Inc.
- [7] Stott, B. 1974. *Review of load flow calculation methods*. *Proceedings of the IEEE*. vol. 62, No. 7, pp. 916 – 929.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

NUMERICAL STUDY OF THE SWIFT-HOHENBERG EQUATION: DEFECTS DYNAMICS FOR FIRST AND SECOND ORDER TEMPORAL DISCRETIZATION

Daniel Lessa Coelho¹ - danielcoelho.uerj@gmail.com

José da Rocha Miranda Pontes¹ - jose.pontes@uerj.br

Norberto Mangiavacchi¹ - norberto@uerj.br

¹GESAR - Grupo de Estudos e Simulações Ambientais de Reservatórios

(UERJ - Universidade do Estado do Rio de Janeiro, Rua Fonseca Teles 121, 21940-903, Rio de Janeiro, RJ)

Abstract. The present paper considers a generalized nonlinear diffusion equation known as the Swift-Hohenberg (SH) equation. Applications of this equation includes the study of pattern formation in physical problems outside equilibrium, described by phase field models, such as the hydrodynamic instability due to buoyancy or surface tension effects or even the structural evolution of nanostructured copolymers. Numerical solutions of the SH equation can present patterns with a large number of topological defects, e.g., grain boundaries, dislocations and disclinations; so computational modelling of such mesoscopic effects can help us understand, explain and predict macroscopic properties of an extended system of interest. A semi-implicit finite difference scheme with temporal discretization of first and second order was implemented in the discretization of the governing equation in two dimensions. Simulations were performed with parameters found in the literature and some of them with a quadratic nonlinearity. Bidimensional rolls patterns, such as hexagonals and stripes were obtained from random initial conditions for certain values of the bifurcation parameter and the coefficient associated to the quadratic nonlinearity. Numerical results obtained from the first and second order schemes suggest that not only the number of defects decays as a power-law with time, but defects dynamics also responds to the implemented numerical approximation.

Keywords: Swift-Hohenberg Equation, Pattern Formation, Topological Defects, Implicit Methods, Finite Difference Methods

1. INTRODUCTION

Several works have been developed regarding pattern dynamics far from equilibrium such as thermal convection, chemical reactions, biological pattern formation and materials science. Two examples of such phenomena is the instability in Rayleigh-Bénard convection and spatially ordered compositional patterns in block copolymers. The Swift-Hohenberg (SH) equation is well known for modelling thermal convection (Swift and Hohenberg, 1977). The equation first appeared in the framework of Bénard thermal convection between two infinite horizontal surfaces with temperatures T_0 and $T_0 + \Delta T$. When the Rayleigh number R , which is a dimensionless temperature difference ΔT , is larger than a critical value R_c , there is emergence of convection

rolls, which are illustrated below. Obviously, the pattern formation can be seen from the top view of the x-y plane, and not the following side view (x-z plane) of the experiment:

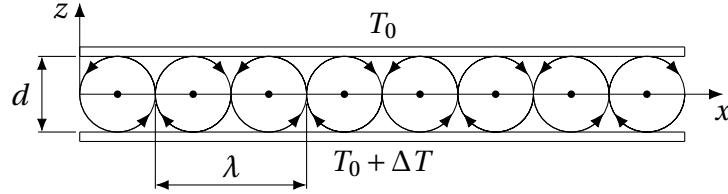


Figure 1- Representation of convection rolls. Four wavelengths can be observed between the plates at different temperatures, distanced by d .

2. THE PHYSICAL MODEL

Some dynamical systems, namely *potencial* or *gradient* systems, are of particular interest because their behavior is simpler than the general case, and because they are frequently encountered in approximate treatments of physical systems. The SH has this important property that is so called *potential* dynamics, which means there is a potential (also called a *Lyapunov* functional) associated with the SH order parameter field $\psi(x, y, t)$ that has the property of decreasing monotonically during the dynamics. This non-increasing behaviour of the *Lyapunov* functional \mathcal{F} for SH is clearly demonstrated by Christov and Pontes, in 2002 [2]. Adopting the following time-dependent Ginzburg–Landau equation with conserved dynamics for the order parameter $\psi(x, y, t)$ and neglecting the advection term, we have:

$$\frac{\partial \psi}{\partial t} = -\frac{\delta \mathcal{F}}{\delta \psi} \quad (1)$$

Discussions about this model applied to block copolymers dynamics is performed by Viñals, in 2009 [5]. Then, for example, a block copolymer melt can be described by a simpler coarse-grained free energy (the Brazovskii or Leibler energy)[5]:

$$\mathcal{F}[\psi] = \int_D d\mathbf{r} \left\{ -\frac{1}{2} \epsilon \psi^2 + \frac{1}{2} [(k_0^2 + \nabla^2)\psi]^2 - \frac{1}{3} g_1 \psi^3 + \frac{1}{4} \psi^4 \right\} \quad (2)$$

Where D represents the domain whose size commensurate with the length scales of the patterns. We considered a rectangle $x \in [0, L_x]$, $y \in [0, L_y]$ and generalized Dirichlet boundary conditions, $\psi = \partial\psi/\partial n$, where n stands for the outward normal direction to the boundary.

The dimensionless parameter ϵ ($\epsilon > 0$) is the bifurcation (control) parameter, which represents the distance to threshold (how close R is to the critical value R_c) in the form $\epsilon = (R - R_c)/R_c$ (assuming $R_c \neq 0$), e.g., in the case of Rayleigh-Bénard convection. The onset of linear instability $R = R_c$ is represented by $\epsilon = 0$. The other parameter g_1 ($0 < g_1 < 1$) acts on the magnitude of the cubic term, which is present merely to draw attention to its participation in the system symmetry break, as will be shown in the simulations. For the finite critical wave number, we adopted $k_0 = 1$, also dimensionless.

Substituting Eq. (2) in the Ginzburg-Landau equation (1) and rescaling all quantities to be dimensionless, we obtain the so-called Swift–Hohenberg equation:

$$\frac{\partial \psi}{\partial t} = \epsilon \psi - (k_0^2 + \nabla^2)^2 \psi + g_1 \psi^2 - \psi^3 \quad (3)$$

Where the laplacian operator is: $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Developing the term $(k_0^2 + \nabla^2)^2 \psi$, we have:

$$\begin{aligned}\frac{\partial \psi}{\partial t} &= \epsilon \psi - k_0^4 \psi - 2k_0^2 \nabla^2 \psi - \nabla^4 \psi + g_1 \psi^2 - \psi^3 \\ &= \epsilon \psi - k_0^4 \psi - 2k_0^2 \frac{\partial^2 \psi}{\partial x^2} - 2k_0^2 \frac{\partial^2 \psi}{\partial y^2} - \frac{\partial^4 \psi}{\partial x^4} - 2 \frac{\partial^4 \psi}{\partial x^2 \partial y^2} - \frac{\partial^4 \psi}{\partial y^4} + g_1 \psi^2 - \psi^3\end{aligned}\quad (4)$$

3. DIFFERENCE SCHEME

The present work consisted of integrating the uniformly forced SH equation with and without the quadratic nonlinearity by the finite differences method. Here, we develop only the second order temporal discretization according to a semi-implicit scheme described in the following sections (used by Christov and Pontes in 2002 [2]), since the first-order discretization is a simpler variation of the present difference scheme. All simulations presented in the next section were done in a rather coarse spatial grid, consisting of 16 points per wavelength (*mesh*). We adopted the following “*Staggered*” grid, initial conditions and the generalized dirichlet boundary condition (function’s value and first derivative equals zero) considered bellow:

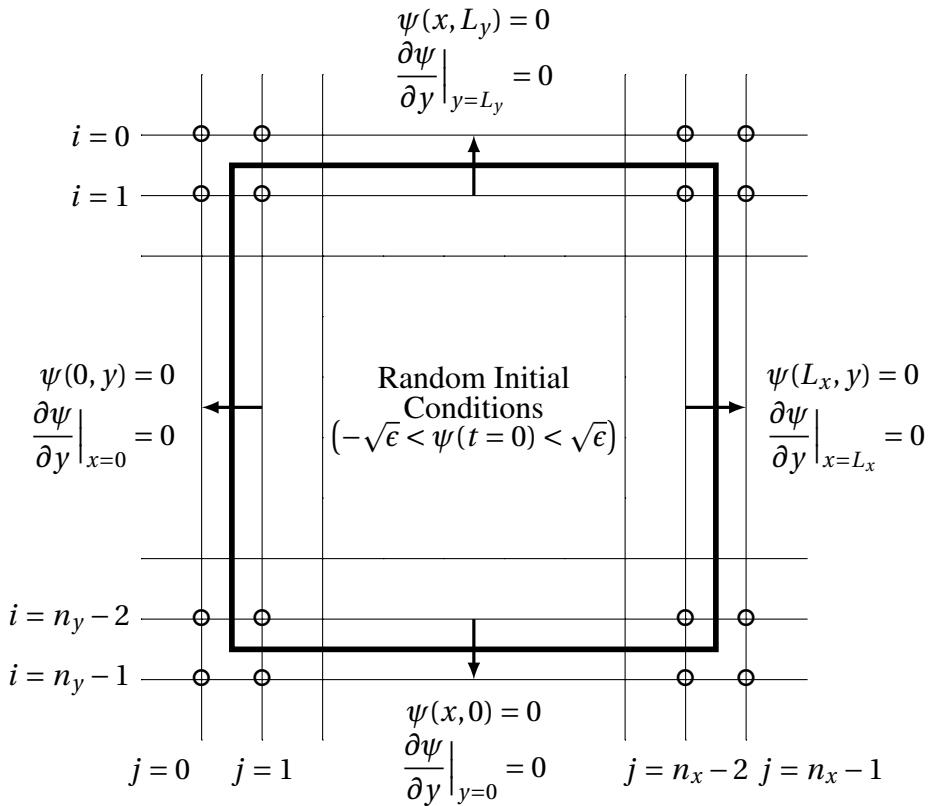


Figure 2- The “*Staggered*” grid.

3.1 Implicit Time-Stepping

The implicit scheme is as follows:

$$\frac{\psi^{n+1} - \psi^n}{\Delta t} = \Lambda_X (\psi^{n+1} + \psi^n) + \Lambda_Y (\psi^{n+1} + \psi^n) + f^{n+1/2} \quad (5)$$

The operators Λ_x , Λ_y and $f^{n+1/2}$ are defined as:

$$\Lambda_X = \frac{1}{2} \left[-\frac{\partial^4}{\partial x^4} - \frac{k_0^4}{2} - \frac{(\psi^n)^2}{2} \right]; \quad \Lambda_Y = \frac{1}{2} \left[-\frac{\partial^4}{\partial y^4} - \frac{k_0^4}{2} - \frac{(\psi^n)^2}{2} \right]$$

$$f^{n+1/2} = \frac{1}{2} \left[\epsilon + \frac{g_1}{2} (\psi^{n+1} + \psi^n) - 2k_0^2 \frac{\partial^2}{\partial x^2} - 2k_0^2 \frac{\partial^2}{\partial y^2} - 2 \frac{\partial^4}{\partial x^2 \partial y^2} \right] (\psi^{n+1} + \psi^n)$$

3.2 Internal Iterations

Since the operators $\Lambda_X^{n+1/2}$, $\Lambda_Y^{n+1/2}$ and the function $f^{n+1/2}$ contain terms in the new stage, we do internal iterations. Besides, internal iterations at each time step are required to secure the approximation for the nonlinearities taking part in the scheme of Equation 5. The iterations loop will continue until convergence is attained from monitoring the L_∞ norm. There is a trade-off related to the time step Δt : for a larger Δt , convergence will be impaired and the number of internal iterations will increase, while a smaller Δt will impact on a smaller number of iterations, but it will imply on a greater number of time steps. The internal iterations scheme reads:

$$\frac{\bar{\psi}^{n,p+1} - \psi^n}{\Delta t} = \Lambda_X (\bar{\psi}^{n,p+1} + \psi^{n,p}) + \Lambda_Y (\bar{\psi}^{n,p+1} + \psi^{n,p}) + f^{n+1/2} \quad (6)$$

where the index (p) refers to the internal iteration number. The superscript $(n, p+1)$ identifies the new iteration, while (n) are the values of the previous time step. The superscript $(n+1)$ for the nonlinear term in the function $f^{n+1/2}$ will be replaced by (n, p) , which stands for the values obtained from the previous iteration. The operators $\Lambda_X^{n+1/2}$, $\Lambda_Y^{n+1/2}$ function $f^{n+1/2}$ are redefined as:

$$\Lambda_X = \frac{1}{2} \left[-\frac{\partial^4}{\partial x^4} - \frac{k_0^2}{2} - \frac{(\psi^n)^2}{2} \right]; \quad \Lambda_Y = \frac{1}{2} \left[-\frac{\partial^4}{\partial y^4} - \frac{k_0^2}{2} - \frac{(\psi^n)^2}{2} \right]$$

$$f^{n+1/2} = \frac{1}{2} \left[\epsilon + \frac{g_1}{2} (\psi^{n,p} + \psi^n) - 2k_0^2 \frac{\partial^2}{\partial x^2} - 2k_0^2 \frac{\partial^2}{\partial y^2} - 2 \frac{\partial^4}{\partial x^2 \partial y^2} \right] (\psi^{n,p} + \psi^n)$$

The iterations proceed until the following criterion for the L_∞ norm is satisfied with $\delta = 10^{-6}$:

$$L_\infty = \frac{\max |\psi^{n,p+1} - \psi^{n,p}|}{\max |\psi^{n,p+1}|} < \delta$$

The last iteration gives the sought function ψ in the new time

$$\psi^{n+1} \stackrel{\text{def}}{=} \psi^{n,p+1}$$

3.3 The Splitting

The *splitting* of Eq. (5) is made according to the Douglas second scheme (also known as “scheme of stabilizing correction”, shown by Christov et al. 1997 [3]). This strategy was chosen to deal with the costly procedure of solving Eq. (6); even though we are dealing with sparse

matrices for the operators, the internal iterations makes the process to be repeated several times during each time step. The scheme is designed as follows:

$$\frac{\tilde{\psi} - \psi^n}{\Delta t} = \Lambda_X \tilde{\psi} + \Lambda_Y \psi^n + f^{n+1/2} + (\Lambda_X + \Lambda_Y) \psi^n \quad (7)$$

$$\frac{\psi^{n,p+1} - \tilde{\psi}}{\Delta t} = \Lambda_Y (\psi^{n,p+1} - \psi^n) \quad (8)$$

Here, ψ is the function for the half-time step. In order to show that the splitting represents the original scheme, we rewrite Eqs. (7) and (8) in the form:

$$(E - \Delta t \Lambda_X) \tilde{\psi} = (E + \Delta t \Lambda_X) \psi^n + 2\Delta t \Lambda_Y \psi^n + \Delta t f^{n+1/2} \quad (9)$$

$$(E - \Delta t \Lambda_Y) \psi^{n,p+1} = \tilde{\psi} + \Delta t \Lambda_Y \psi^n \quad (10)$$

where E is the unity operator. The intermediate variable $\tilde{\psi}$ is eliminated by applying the operator $(E - \Delta t \Lambda_X)$ to Eq. (10) and adding the result to Eq. (9). The result may be rewritten as:

$$(E + \Delta t^2 \Lambda_X \Lambda_Y) \frac{\psi^{n,p+1} - \psi^n}{\Delta t} = (\Lambda_X + \Lambda_Y) (\psi^{n,p+1} + \psi^n) + f^{n+1/2} \quad (11)$$

A comparison with Eq. (5) shows that Eq.(11) is actually equivalent to the first one except by the defined positive operator having a norm greater than one:

$$B \equiv E + \Delta t^2 \Lambda_X \Lambda_Y = E + O(\Delta t^2) \quad (12)$$

which acts on the term $(\psi^{n,p+1} - \psi^n)/\Delta t$. This means that this operator does not change the steady state solution. Furthermore, since $\|B\| > 1$, the scheme given by Eqs. (7)-(8) is more stable than the target one (Eq. (5)).

3.3 Spatial Discretization

In order to apply also a second order approximation in space, we make a Taylor development of a function ψ in the points of a uniform grid, as follows:

$$\begin{aligned} \psi_{i+1} &= \psi_i + \Delta x \psi' + \frac{\Delta x^2}{2!} \psi'' + \frac{\Delta x^3}{3!} \psi''' + \frac{\Delta x^4}{4!} \psi'''' + O\left(\frac{\Delta x^5}{5!}\right) \\ \psi_{i+2} &= \psi_i + 2\Delta x \psi' + \frac{4\Delta x^2}{2!} \psi'' + \frac{8\Delta x^3}{3!} \psi''' + \frac{16\Delta x^4}{4!} \psi'''' + O\left(\frac{32\Delta x^5}{5!}\right) \\ \psi_{i-1} &= \psi_i - \Delta x \psi' + \frac{\Delta x^2}{2!} \psi'' - \frac{\Delta x^3}{3!} \psi''' + \frac{\Delta x^4}{4!} \psi'''' - O\left(\frac{\Delta x^5}{5!}\right) \\ \psi_{i-2} &= \psi_i - 2\Delta x \psi' + \frac{4\Delta x^2}{2!} \psi'' - \frac{8\Delta x^3}{3!} \psi''' + \frac{16\Delta x^4}{4!} \psi'''' - O\left(\frac{32\Delta x^5}{5!}\right) \end{aligned}$$

Considering the function ψ in a bidimensional domain, we can define the derivatives by truncating the Taylor's development and arranging the equations. The second derivative with second order accuracy can be written as

$$\frac{\partial^2 \psi_{i,j}}{\partial x^2} = \frac{\psi_{i,j-1} - 2\psi_{i,j} + \psi_{i,j+1}}{\Delta x^2}; \quad \frac{\partial^2 \psi_{i,j}}{\partial y^2} = \frac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{\Delta y^2}$$

and the fourth derivative with second order accuracy as

$$\begin{aligned}\frac{\partial^4 \psi_{i,j}}{\partial x^4} &= \frac{\psi_{i,j-2} - 4\psi_{i,j-1} + 6\psi_{i,j} - 4\psi_{i,j+1} + \psi_{i,j+2}}{\Delta x^4} \\ \frac{\partial^4 \psi_{i,j}}{\partial y^4} &= \frac{\psi_{i-2,j} - 4\psi_{i-1,j} + 6\psi_{i,j} - 4\psi_{i+1,j} + \psi_{i+2,j}}{\Delta y^4} \\ \frac{\partial^4 \psi_{i,j}}{\partial x^2 \partial y^2} &= \frac{1}{\Delta x^2 \Delta y^2} \left(\psi_{i-1,j-1} - 2\psi_{i,j-1} + \psi_{i+1,j-1} - 2\psi_{i-1,j} + 4\psi_{i,j} - 2\psi_{i+1,j} + \right. \\ &\quad \left. + \psi_{i-1,j+1} - 2\psi_{i,j+1} + \psi_{i+1,j+1} \right)\end{aligned}$$

4. NUMERICAL RESULTS

The numerical simulations conducted in the framework of the SH equation considered the following common parameters: $k_0 = 1.0$, $\epsilon = 0.3$, $\Delta t = 1.0$ and $L_x = L_y = 50.265$ a.u.. The domain size corresponds to 8 wavelenghts (λ) obtained from the wavenumber (k_0) trough the relation:

$$\lambda = \frac{2\pi}{k_0} = 2\pi \quad ; \quad 8\lambda = 16\pi \approx 50.265 \text{ a.u.} \quad (13)$$

We assess the rate of evolution of the pattern during the simulation by monitoring the relative L_1 norm defined as:

$$L_1 = \frac{1}{\Delta t} \frac{\sum_{i,j} |\psi^{n+1} - \psi^n|}{\sum_{i,j} |\psi^{n+1}|} \quad (14)$$

which roughly corresponds to the ratio between the spatial average of the modulus of time derivative $\frac{\partial \psi}{\partial t}$ and the spatial average of the modulus of the function itself. The calculations begin from the a random initial condition and proceeded until $L_1 \leq 1 \times 10^{-6}$, when it can be assumed that the state is steady. The first group of simulations corresponds to the *stripes*, which contains the structure for its respective time shown below:

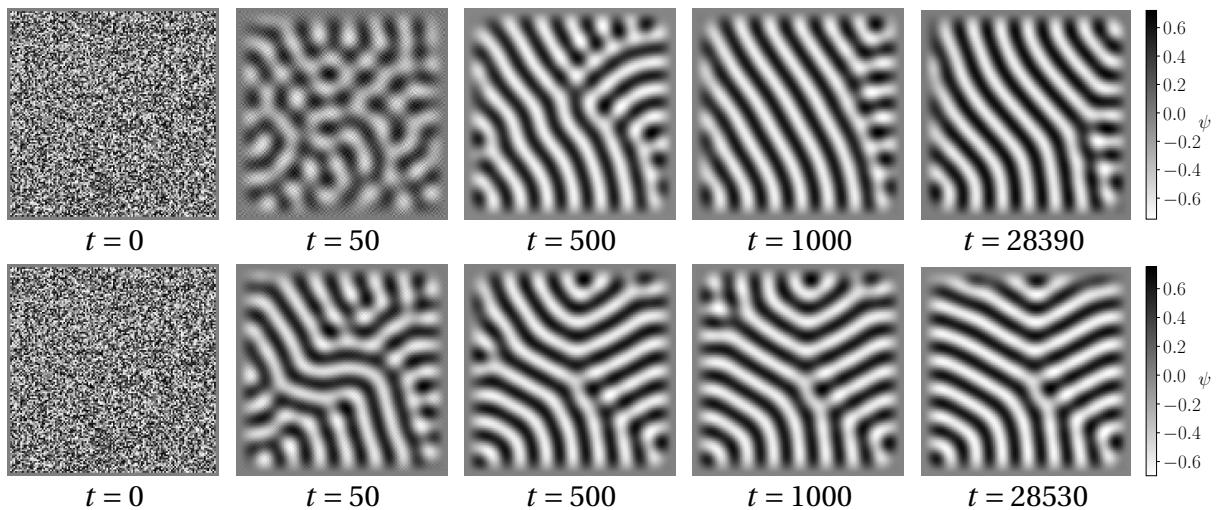


Figure 3- First group of simulations. Evolution of the pattern (*stripes*) in the 128×128 square uniformly forced with $\epsilon = 0.3$ and $g = 0.0$ for first (top row) and second (bottom row) order temporal approximation.

The second group of simulations corresponds to the *hexagons*, which contains the structure for its respective time shown below:

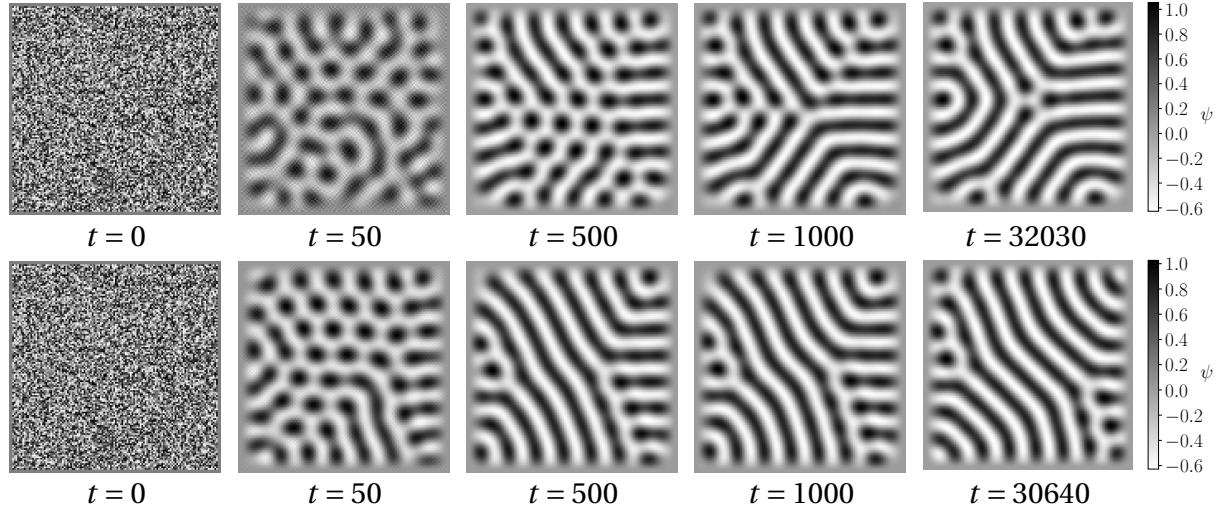


Figure 4- Second group of simulations. Evolution of the pattern (initially *hexagons*) in the 128×128 square uniformly forced with $\epsilon = 0.3$ and $g = 0.5$ for first (top row) and second (bottom row) order temporal approximation.

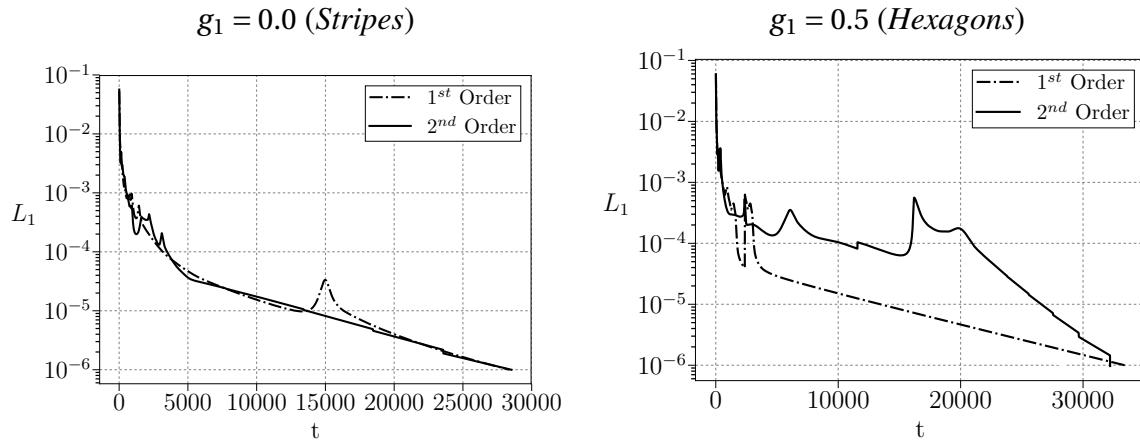


Figure 5- L_1 norm vs. time (t) for both groups of simulations with details and parameters indicated.

There are several simulation results regarding the SH equation. Particularly, there is a special group of simulations that can be accessed through online *java* demonstrations provided by Michael Cross (last modified in 2008). There, one can change the main parameters of the equation and quickly get simulation results for the long time evolution until the stationary patterns. Considering that M. Cross used periodic boundary condition for all simulations, we can acquire qualitative data for comparison with our results, mainly the function's maximum and minimum.

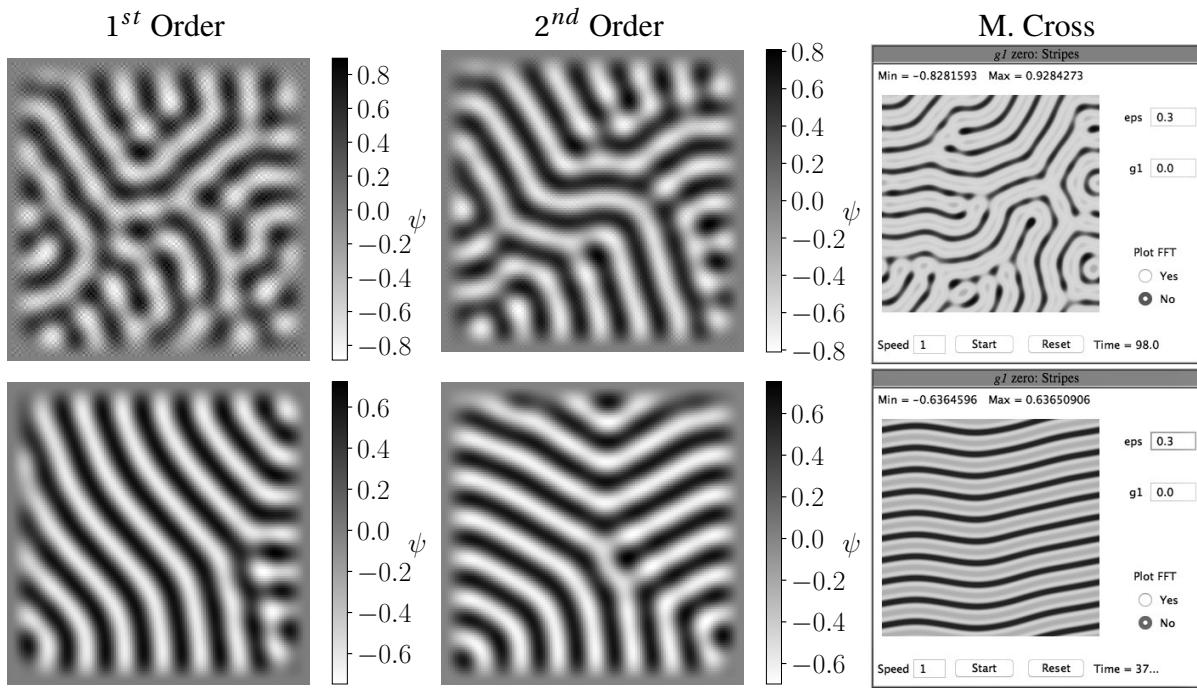


Figure 6- First group of simulations. Transient (first row) and steady state (second row) patterns comparison with Michael Cross work, in 2008.

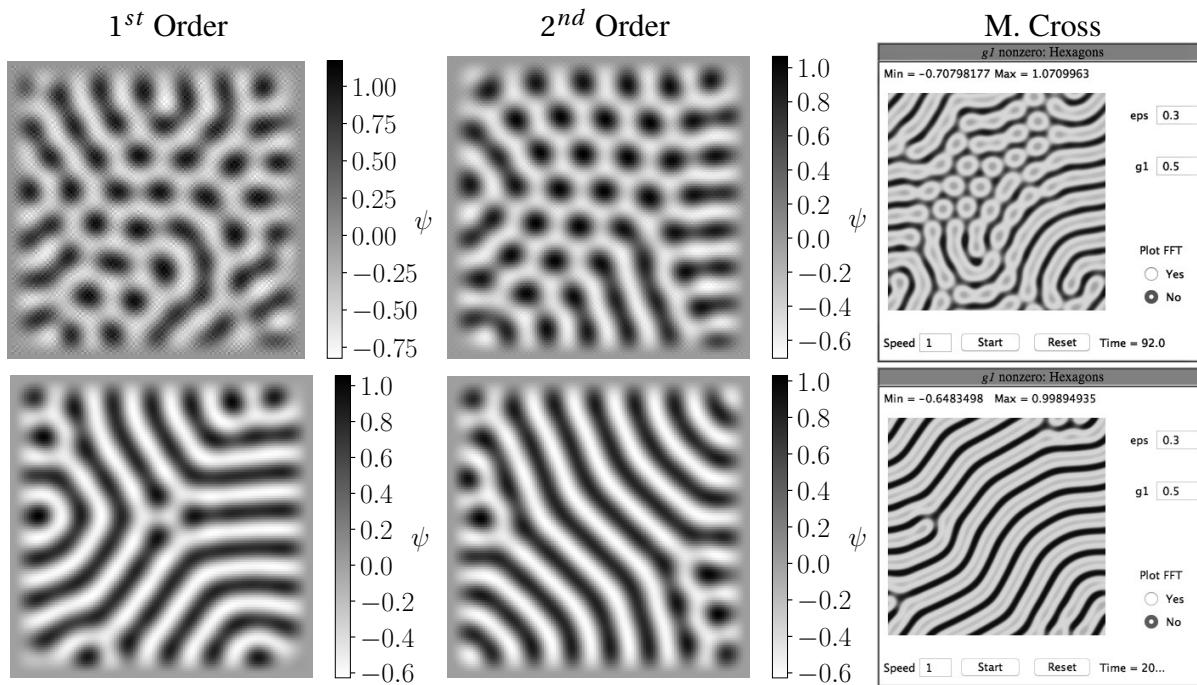


Figure 7- Second group of simulations. Transient (first row) and steady state (second row) patterns comparison with Michael Cross work, in 2008.

5. CONCLUSION

In the last decades, materials science has attracted interdisciplinary attentions due to the challenge of modeling pattern formation regarding phase transitions and dynamics of microstructures in distinct materials. In general, differential equations of higher orders with non-linear terms are involved. In this area, some authors mentioned in this work were able to study theoretical and numerical dynamics that lead to the formation of periodic structures, such as Vitral et al. [9], Pontes et al. [7], Walgraef et al. [8], Provatas et al. [6] and Viñals [5], and therefore the objectives of the present work was to achieve a reliable numerical approach to one of those equations involved in such physical problems.

All simulations were uniformly forced 128×128 squares. Both simulations (first and second approximations) from the same group started from the exact same random initial conditions and presented the long evolution of roll patterns with specific wavenumber (k_0) seen in the previous section.

In a general matter, all simulations took almost the same physical time to attain the steady state (S.S.); which draws our attention to a possible strong relation between reaching the uniform state and the system size. There is also a tendency of rolls to align perpendicularly to the side walls, which concerns the relevance of the boundary conditions employed and this is sufficiently strong to change one of the qualitative features of patterns formed in entirely supercritical systems. Although we didn't consider a large domain (i.e. approximately 20×20 wavelengths), we can still capture the main features of the SH equation, such as the amplitude grain boundary (S.S. for the first group - first order and S.S. for the second group - second order), phase grain boundary (S.S. for the first group - second order) and disclinations (S.S. for the second group - first order).

Defects dynamics responds very differently in first and second order approximations. The L_1 norm help us detect not only the growth of early stages but also the slow displacement of an almost saturated pattern that still moves, due to phase dynamics, which is essentially diffusional (slow). Through the L_1 norm analysis, in the second group with second order discretization we note a totally different dynamics, which leads the system to a fewer number of defects in the final state, compared to the first order.

In the first group (*stripes*), the defects can be noted earlier because of the structure formation with respect to the nonlinear saturation and are eliminated in time, suggesting an adjustment of the structure through time obeying the phase evolution which is mainly driven by diffusion. Second order approximation for the first group preserves more defects than the first order, in this simulation. *Hexagonal* patterns seen in the second group give place to *stripes* for long times and for the first order approximation a curious pattern stays with the S.S., usually called as a concave disclination.

Considering other works, there is considerable qualitative agreement between the present papers and the results obtained by Christov et al. 2002 [2] and Michael Cross et al. 2008. This encourages the current development of semi-implicit methods capable of integrating efficiently fourth-order parabolic equations with nonlinear terms such as the Swift-Hohenberg; and applying such method to help us understand, explain and predict macroscopic properties and nonlinear behaviour of extended systems.

Acknowledgements

The authors would like to thank the support from CNPq and FAPERJ.

REFERENCES

- [1] Swift, J. and Hohenberg, P. C. 1977. Hydrodynamic fluctuations at the convective instability. *Phys. Rev. A*, 15(1):319–328.
- [2] Christov, C. I., Pontes, J. 2002. Numerical scheme for Swift-Hohenberg equation with strict implementation of Lyapunov functional. *Mathematical and Computer Modelling*, Elsevier Science, 35:87–99.
- [3] Christov, C. I., Pontes, J., Walgraef D. and Velarde, M. G. 1997. Implicit time splitting for fourth-order parabolic equations. *Comp. Meth. Appl. Meth. & Eng.*, Elsevier Science, 148, pp.209-224.
- [4] Cross, M. C. and Hohenberg, P. C. 1993. Pattern formation outside of equilibrium. *Rev. Mod. Phys.*, 65(3):851.
- [5] Viñals, J. 2009. Defect Dynamics in Mesophases. *Journal of the Physical Society of Japan*, 78:041011–.
- [6] Provatas N., Elder K. 2010. Phase-Field Methods in Materials Science and Engineering. *Wiley* 3527407472,9783527407477.
- [7] Pontes, J., Walgraef, D. and Christov, C.I. 2008. Pattern formation in spatially ramped Rayleigh-Bénard systems *Journal of Computational Interdisciplinary Sciences*, 11–32.
- [8] Walgraef, D. and Aifantis, E. C. 1985. Dislocation patterning in fatigued metals as a result of dynamical instabilities. *Journal of Applied Physics*, vol. 58, page 688–.
- [9] Vitral, E., Walgraef D., Pontes , J., Anjos, G.R. and Mangiavacchi, N. 2018. Nano-patterning of surfaces by ion sputtering: Numerical study of the anisotropic damped Kuramoto-Sivashinsky equation. *Computational Materials Science*, vol. 146, page 193–203.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

MÉTODO DE VOLUMES FINITOS APLICADO A UM PROBLEMA DE TRANSPORTE DE CONTAMINANTES

Anna Luisa Soares Castro Coimbra¹ - anna_luisa@id.uff.br

Wagner Rambaldi Telles^{1,2} - wtelles@id.uff.br

¹ Universidade Federal Fluminense, Instituto do Noroeste Fluminense de Educação Superior, INFES, Santo Antônio de Pádua, RJ, Brasil

²Universidade Federal Fluminense, Pós-Graduação em Modelagem Computacional em Ciência e Tecnologia, MCCT, Volta Redonda, RJ, Brasil

O presente trabalho consiste em modelar o transporte de contaminantes em um rio, visto que boa parte da água potável proveniente do mesmo sofre constante ameaçada de poluição, devido à expansão urbana, aliada ao desenvolvimento da indústria e das atividades agrícolas. Segundo Magossi (1997) “no Brasil, quase todos os rios possuem algum tipo de poluição”. Para o problema em questão, será analisado o lançamento de um traçador salino, em um trecho do rio São Pedro, pequeno curso d’água situado na bacia hidrográfica do rio Macaé, centro-norte do Estado do Rio de Janeiro. Para esse estudo utilizar-se-ão os dados do trabalho de campo realizado por Souza (2009), onde o experimento teve o intuito de simular um cenário de poluição acidental no curso de água em questão. A partir dos dados obtidos no campo, modela-se o problema a fim de obter a equação mais representativa possível, para isso utiliza-se o Método do Volumes Finitos (MVF), que tem sido utilizado nas mais diversas áreas da ciência e, segundo OSSES (2016), trata-se do método preferido por cientistas e engenheiros que trabalham com a mecânica de fluidos. Esse método consiste em determinar um limite para a região investigada, em seguida essa região é dividida em um número finito de volumes, denominados volumes de controle, os quais, quanto menores, melhor torna-se o resultado da aproximação. Após esse processo, é feito o cálculo da integral múltipla da equação aplicada a cada volume, aproximando os valores das variáveis nas faces e as derivadas com a informação das variáveis nodais, obtendo um sistema algébrico ao final. A partir desse ponto, é construída uma matriz representando o sistema algébrico a ser implementado computacionalmente utilizando o software Scilab. Espera-se que este trabalho, além de auxiliar na tomada de decisões, possibilite, principalmente, a obtenção de um modelo computacional mais robusto que sirva como base para futuros trabalhos de modelagem de fluidos na região de interesse.

Palavras-chave: Poluição Hídrica, Método do Volumes Finitos, Modelagem Computacional, Transporte de Contaminantes

REFERÊNCIAS

- [1] Magossi, L. R. e Bonacella, P. H. Poluição das águas. São Paulo: Editora Moderna, 1997.
- [2] Osses, J. A. **O método de volumes finitos.** Disponível em: <<https://www.esss.co/blog/o-metodo-de-volumes-finitos/>>. Acesso em: 25 de set. de 2018.
- [3] Souza, E. P. **Avaliação de mecanismos dispersivos em rios atravésde problemas inversos.** Dissertação (Mestrado em Modelagem Computacional), Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Nova Friburgo, RJ, 2009.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

ANÁLISE COMPARATIVA ENTRE OS MÉTODOS ALGORITMO DE COLISÃO DE PARTÍCULAS E NEWTON NA RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES NÃO LINEARES

Rennan Mendes de Moraes dos Santos Dias¹ - rennandias@id.uff.br

Wagner Rambaldi Telles^{1,2} - wtelles@id.uff.br

¹Universidade Federal Fluminense, Instituto do Noroeste Fluminense de Educação Superior, INFES, Santo Antônio de Pádua, RJ, Brasil

²Universidade Federal Fluminense, Pós-Graduação em Modelagem Computacional em Ciência e Tecnologia, MCCT, Volta Redonda, RJ, Brasil

Resumo. No presente trabalho é analisada a viabilidade da aplicação dos métodos Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA) para resolução de sistemas de equações não lineares em comparação ao método de Newton, comumente usado para essa classe de problemas. A escolha desta abordagem é justificada pelo não uso de derivadas e baixa dependência da estimativa inicial dos métodos estocásticos, em contraponto aos métodos determinísticos. Com isso, pôde-se concluir que os métodos de otimização estocásticos, aqui analisados, estão aptos a serem aplicados a esse tipo de problemática.

Keywords: Sistemas de equações não lineares, Algoritmo de Colisão de Partículas, Algoritmo de Colisão de Múltiplas Partículas, Método de Newton

1. INTRODUÇÃO

Sistemas de equações não-lineares são extremamente comuns durante a modelagem de problemas práticos relacionados à diferentes áreas, como: engenharia, física, ciência da computação, biologia, entre outras. Dentre elas, destacam-se as soluções de problemas de equilíbrio, etapas na solução de problemas de otimização, discretização de sistemas de equações diferenciais parciais, assim como modelos complexos de transporte e modelagem estatística (Santos, 2016). Neste cenário, ainda se fazem necessários os estudos de diferentes técnicas de resolução desses sistemas.

No processo de resolução por métodos iterativos de um sistema de equações, tem-se a possibilidade de transforma-lo em um problema de otimização, submetido à minimização da função objetivo equivalente às soluções do sistema original (Viana, 2017).

Neste artigo são usados dois sistemas de equações não lineares. A partir destes é feita uma análise do desempenho dos métodos Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA) frente ao método de Newton. Desta forma, pretende-se encontrar todas as soluções de ambos os sistemas após um processo investigativo no que se refere à configuração mais adequada visando as execuções dos algoritmos, verificando assim, a aplicabilidade dos métodos para esse tipo de problemática.

2. SISTEMAS DE EQUAÇÕES NÃO LINEARES

Seja um sistema de equações não lineares qualquer,

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (1)$$

Pode-se escrever, na forma vetorial, cada equação representada pelas f_n funções. Assim, tem-se:

$$\mathbf{f} = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2)$$

Na transformação de um sistema de equações não lineares em um problema de minimização não linear sem restrições, necessita-se da criação de uma função objetivo $F(\mathbf{x})$ para obtenção do ponto ótimo, sendo essa gerada a partir da função vetorial descritas em (2). Esse processo é realizado pela norma L_p de Holder, definida como:

$$F(\mathbf{x}) \equiv L_p = \left\{ \sum_{k=1}^n |f_k(\mathbf{x})|^p \right\}^{\frac{1}{p}} \quad (3)$$

onde $p \in \mathbb{N}$ e n é a dimensão do espaço vetorial.

Para o parâmetro p , em problemas que envolvem esse tipo de processo, tem-se três casos mais clássicos: $p = 1$, $p = 2$ e $p \rightarrow +\infty$. Nesse trabalho, para os dois métodos estocásticos estudados, utilizou-se $p = 2$ e omitiu-se a raiz, ou seja, é minimizada a soma dos quadrados das componentes. Para maiores detalhes sobre a utilização da norma de Holder em problemas de otimização, vide Viana (2017). Logo, a Equação (3) é denotada por:

$$F(\mathbf{x}) \equiv L_2 = \sum_{k=1}^n |f_k(\mathbf{x})|^2 \quad (4)$$

3. METODOLOGIA

3.1 Métodos de Otimização

O conceito de otimização é conhecido como o problema de achar o valor máximo ou mínimo de uma função objetivo sujeita, na maioria dos casos, a um conjunto de restrições. Dentro deste conceito trabalham-se com duas classes de métodos: determinísticos e estocásticos (Holtz, 2005). A primeira destas, geralmente faz uso de pelo menos a primeira derivada da função objetivo, que por sua vez, deve ser contínua e diferenciável no espaço de busca. Além disso,

os métodos determinísticos são extremamente dependentes do ponto de partida fornecido, mas apresentam teoremas que lhes garantem a convergência para uma solução ótima, que não é necessariamente a solução ótima global, mostrando assim uma dificuldade ao se trabalhar com funções que possuem vários ótimos.

Já a segunda, classe que engloba os métodos aqui estudados, Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA), usa somente a avaliação da função objetivo e parâmetros estocásticos, que garantem a fuga de mínimos (ou máximos) locais, não requerendo que a função objetivo seja contínua ou diferenciável. Outra vantagem é não haver restrição alguma quanto ao ponto de partida dentro do espaço de busca da solução.

Nas seções a seguir, são descritos os métodos estocásticos Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA), bem como o método determinístico de Newton.

3.2 Algoritmo de Colisão de Partículas (PCA)

Proposto por Wagner Sacco (2006), o Algoritmo de Colisão de Partículas (PCA) é um algoritmo do tipo Metropolis, inspirado no processo de colisão de partículas nucleares, particularmente na ideia de espalhamento e absorção. Uma das vantagens desse método é a grande quantidade de parâmetros definidos por rotinas aleatórias, e não pelo usuário ao configurar o início da execução. Nele é necessário definir apenas os intervalos de busca para as variáveis do problema tratado, além do número de vezes que ocorrerá o processo iterativo (n_{PCA}) e exploratório em torno da melhor solução (n_{Exp}). O processo de busca pelo ótimo da função objetivo se dá, resumidamente, de acordo com os passos abaixo:

Passo 1. Defina o número de vezes em que o processo iterativo irá ocorrer (n_{PCA}) e o número de explorações em torno da melhor solução (n_{Exp});

Passo 2. Defina os limites superior (**U**) e inferior (**L**) do espaço de busca para cada uma das variáveis do problema;

Passo 3. Atribua 1 como o valor inicial para a variável contador (c);

Passo 4. Gere uma solução inicial aleatória dentro do domínio de busca da solução (**OldConfig**) e faça **BestConfig** \leftarrow **OldConfig**;

Passo 5. Caso $c \geq n_{PCA}$ encerre o algoritmo. Caso contrário, faça $c = c + 1$ e gere uma nova solução, **NewConfig**, calculada a partir de uma perturbação estocástica na solução **OldConfig** de acordo com a fórmula: $\text{NewConfig} = \text{OldConfig} + [(\text{U} - \text{OldConfig})rand - (\text{OldConfig} - \text{L})(1 - rand)]$. Se $F(\text{NewConfig}) < F(\text{OldConfig})$, então **OldConfig** \leftarrow **NewConfig** e vá para o Passo 6. Caso contrário, vá para o Passo 7. Obs: Se $F(\text{NewConfig}) < F(\text{BestConfig})$, então **BestConfig** também é atualizado com os valores contidos em **NewConfig**;

Passo 6. Faça o procedimento de Exploração n_{Exp} vezes gerando novas estimativas **NewConfig** de acordo com a fórmula: $\text{U}' = (1+0, 2rand)\text{OldConfig}; \text{L}' = (1-0, 2rand)\text{OldConfig}; \text{NewConfig} = \text{OldConfig} + [(\text{U}' - \text{OldConfig})rand - (\text{OldConfig} - \text{L}')(1 - rand)]$. Caso $F(\text{NewConfig}) < F(\text{OldConfig})$, então **OldConfig** \leftarrow **NewConfig**. Ao final do

processo de Exploração, volte para o Passo 5. Obs: Se $F(\text{NewConfig}) < F(\text{BestConfig})$, então **BestConfig** também é atualizado com os valores contidos em **NewConfig**;

Passo 7. Calcule a probabilidade de espalhamento, $p_{scattering}$, de acordo com a fórmula: $p_{scattering} = 1 - (F(\text{BestConfig})/F(\text{NewConfig}))$. Gere um número aleatório $rand$. Se $p_{scattering} < rand$, então **OldConfig** recebe o valor de **NewConfig** e vá para o Passo 6. Caso contrário, gere uma nova solução aleatória **OldConfig** e vá para o Passo 5.

Ao término do Algoritmo de Colisão de Partículas (PCA), a estimativa do mínimo da função objetivo é dada pela solução **BestConfig**. Para visualizar os detalhes do método, vide Sacco, Oliveira e Pereira (2006) e Telles (2014).

3.3 Algoritmo de Colisão de Múltiplas Partículas (M-PCA)

Baseado no Algoritmo de Colisão de Partículas (PCA), o Algoritmo de Colisão de Múltiplas Partículas (M-PCA) introduz a ideia do uso de multi-partículas. Quando implementado num ambiente de computação de alto desempenho, tem o seu processo agilizado ao abrir um leque maior de exploração do espaço de busca (Luz, Becceneri e Campos Velho, 2008).

O Algoritmo de Colisão de Múltiplas Partículas (M-PCA), em sua essência, possui exatamente o mesmo procedimento adotado no Algoritmo de Colisão de Partículas (PCA) canônico. Porém, com a adição de n_p partículas executando a busca no mesmo espaço de soluções de maneira independente, porém relativamente colaborativa, tem-se a necessidade de implementar um mecanismo de comunicação indireta entre as partículas, capacitando uma coordenação das atividades (Luz, Becceneri e Campos Velho, 2008). Esse processo é feito incluindo mais um contador no Passo 3 (c_p inicializado em 1) e um novo *loop* após o Passo 4, fazendo o algoritmo executar o processo n_p vezes.

3.4 Método de Newton para Sistemas de Equações Não Lineares

O método determinístico de Newton aproxima uma função não linear por meio de um modelo linear local, na vizinhança de um ponto \mathbf{x}_k , onde busca-se encontrar soluções para o sistema de equações não lineares. Aplicado à funções de uma variável real, o modelo é gerado a partir da reta tangente da função no ponto. Quando expandido para funções com mais de uma variável, surge a necessidade do cálculo das derivadas parciais, como consequência, da matriz Jacobiana. Tem-se a seguinte equação que resume o método de Newton,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J^{-1}(\mathbf{x}_k)F(\mathbf{x}_k) \quad (5)$$

onde J^{-1} corresponde a inversa da matriz Jacobiana e $k = 0, 1, 2, \dots$. O processo continua até que um critério de parada pré-estabelecido seja atendido. Esse método determinístico é comumente utilizado na resolução de sistemas de equações lineares por possuir uma boa precisão (Oliveira e Martinez, 2013).

Maiores detalhes sobre o método de Newton e seu respectivo algoritmo podem ser vistos em Burden et al. (2016) e Cunha (2017).

4. PROBLEMAS PROPOSTOS

Neste trabalho foram utilizados dois problemas que envolvem sistemas de equações não lineares.

O primeiro trata-se de um problema envolvendo equações trigonométricas. Possui 13 soluções e foi usado para testar o algoritmo *Continuous-Greedy Randomized Adaptive Search Procedure* (C-GRASP), desenvolvido por Hirsh et al. (2006). Seguem as equações que representam o sistema,

$$\begin{cases} -\sin(x_1)\cos(x_2) - 2\cos(x_1)\sin(x_2) = 0 \\ \cos(x_1)\sin(x_2) - 2\sin(x_1)\cos(x_2) = 0 \\ 0 \leq x_1, x_2 \leq 2\pi \end{cases} \quad (6)$$

Enquanto o segundo problema foi proposto por Bini e Mourrain em 2004. Contém 8 soluções e está representado abaixo,

$$\begin{cases} -x_2^2x_3^2 - x_2^2 + 20x_2x_3 - x_3^2 - 13 = 0 \\ -x_1^2x_3^2 - x_1^2 + 24x_1x_3 - x_3^2 - 12 = 0 \\ -x_1^2x_2^2 - x_1^2 + 24x_1x_2 - x_2^2 - 13 = 0 \\ 0 \leq x_1, x_2, x_3 \leq 20 \end{cases} \quad (7)$$

Ambos os exemplos dispostos acima foram retirados da tese de Silva (2017), a qual serviu como mecanismo de comparação para a análise dos resultados obtidos.

5. RESULTADOS

Para obtenção dos resultados a seguir, foi utilizado um computador AMD Phenom II X4 830 e memória RAM de 4,0 GB. Os métodos Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA) foram implementados na linguagem de programação C, enquanto o método de Newton foi implementado em SciLab. Apesar de terem sido usadas linguagens de programação diferentes, essas não apresentam influência sobre os resultados apresentados e sobre os critérios considerados para comparação adotados neste trabalho, sendo esses: número de raízes encontradas e quantidade máxima do número de avaliações da função objetivo (NAF). Além disso, foram usadas as mesmas estimativas iniciais para cada uma das execuções dos referidos métodos de otimização.

5.1 Método de Otimização Algoritmo de Colisão de Partículas (PCA)

Nessa seção são mostrados os resultados obtidos para ambos os sistemas, sob os seguintes parâmetros: $n_{PCA} = 100$; $n_{Exp} = 150$; Número de execuções = 100. Na Tabela 1 são apresentadas as soluções para o sistema de equações proposto (6), obtidos por meio do Algoritmo de Colisão de Partículas (PCA) ao longo das 100 execuções.

Tabela 1- Raízes do primeiro sistema de equações não lineares encontradas pelo PCA.

Tempo total das 100 execuções = 106,716s.				
Raízes	x_1	x_2	$F(x_1, x_2)$	NAF
1	0.001350	3.141427	0.000009	14380
2	0.000837	0.001092	0.000009	434
3	3.140590	3.142283	0.000007	3071
4	1.570544	4.711289	0.000006	2684
5	1.571064	1.571038	0.000001	481
6	0.001311	6.282819	0.000009	12976
7	4.712910	1.570205	0.000003	9815
8	4.711482	4.712109	0.000005	14653
9	6.281791	3.143401	0.000026	14951

Já na Tabela 2 são apresentadas as soluções para o sistema de equações proposto (7), obtidos por meio do Algoritmo de Colisão de Partículas (PCA) ao longo das 100 execuções.

Tabela 2- Raízes do segundo sistema de equações não lineares encontradas pelo PCA.

Tempo total das 100 execuções = 193,421s.					
Raízes	x_1	x_2	x_3	$F(x_1, x_2, x_3)$	NAF
1	0.331754	4.652733	4.597399	0.011332	15100
2	0.779913	0.778825	0.780500	0.000503	15100
3	4.627805	4.621725	4.622936	0.307206	14951
4	0.775791	10.840621	0.776109	0.013860	15100
5	4.655184	0.331474	4.594826	0.018941	14951
6	10.860191	0.779808	0.778585	0.013300	14951
7	0.776954	0.775200	10.843573	0.023913	14951

5.2 Método de Otimização Algoritmo de Colisão de Múltiplas Partículas (M-PCA)

Nessa seção são mostrados os resultados obtidos, para ambos os sistemas, sob os seguintes parâmetros: $n_{PCA} = 100$; $n_{Exp} = 150$; $n_p = 5$; Número de execuções = 100. Na Tabela 3 são apresentadas as soluções para o sistema de equações proposto (6), obtidos por meio do Algoritmo de Colisão de Múltiplas Partículas (M-PCA) ao longo das 100 execuções.

Tabela 3- Raízes do primeiro sistema de equações não lineares encontradas pelo M-PCA.

Tempo total das 100 execuções = 188,301s.				
Raízes	x_1	x_2	$F(x_1, x_2)$	NAF
1	1.571305	4.712110	0.000002	14200
2	3.141750	3.141498	0.000000	4517
3	4.712934	1.569970	0.000005	67899
4	0.000891	0.001086	0.000010	1648
5	0.001348	6.282898	0.000009	16292
6	4.713564	4.713009	0.000009	20141
7	1.571842	1.571240	0.000006	26800
8	6.282559	6.282241	0.000006	50879
9	6.281904	3.141088	0.000009	26858
10	3.140982	6.282034	0.000008	36914
11	0.000458	3.142372	0.000004	202

Por outro lado, na Tabela 4 são apresentadas as soluções para o sistema de equações proposto (7), obtidos por meio do Algoritmo de Colisão de Múltiplas Partículas (M-PCA) ao longo das 100 execuções.

Tabela 4- Raízes do segundo sistema de equações não lineares encontradas pelo M-PCA.

Tempo total das 100 execuções = 1013,856 s.					
Raízes	x_1	x_2	x_3	$F(x_1, x_2, x_3)$	NAF
1	0.332288	4.623506	4.627188	0.001864	75351
2	4.605072	0.331706	4.645037	0.007728	75202
3	0.778370	0.778963	10.858101	0.012425	75351
4	4.628575	4.621661	4.627229	0.294667	75202
5	10.855345	0.779133	0.779047	0.000269	75351
6	0.779015	10.854913	0.778377	0.002641	75351
7	4.638161	4.612723	0.332516	0.007440	75202
8	0.779221	0.778819	0.780038	0.000314	74904

5.3 Método de Newton

Nessa seção são mostrados os resultados obtidos, para ambos os sistemas, com critério de parada da norma infinita $< 10^{-7}$ (entre duas iterações sucessivas) e número de execuções = 100. Na Tabela 5 são apresentadas as soluções para o sistema de equações proposto (6), obtidos por meio do Método de Newton ao longo das 100 execuções.

Tabela 5- Raízes do primeiro sistema de equações não lineares encontradas pelo Método de Newton.

Tempo total das 100 execuções = 0,164s.			
Raízes	x_1	x_2	NAF
1	1.5707963	4.712389	30
2	0	3.1415927	24
3	0	0	24
4	3.1415927	3.1415927	36
5	0	6.2831853	18
6	6.2831853	6.2831853	48
7	4.712389	1.5707963	42
8	1.5707963	1.5707963	30
9	3.1415927	6.2831853	36
10	3.1415927	0	48
11	6.2831853	3.1415927	60

Na sequência, Tabela 6, são apresentadas as soluções para o sistema de equações proposto (7), obtidos por meio do Método de Newton ao longo das 100 execuções.

Tabela 6- Raízes do segundo sistema de equações não lineares encontradas pelo Método de Newton.

Tempo total das 100 execuções = 0,3879041 s.				
Raízes	x_1	x_2	x_3	NAF
1	0.3320731	4.6251816	4.6251816	96
2	4.6251816	0.3320731	4.6251816	144
3	4.6251816	4.6251816	4.6251816	132
4	0.779548	0.779548	10.857704	72
5	0.779548	0.779548	0.779548	156
6	0.779548	10.857704	0.779548	168

A partir dos resultados exibidos acima, pode-se construir a Tabela 7, a qual permite realizar uma análise comparativa entre os métodos apresentados nesse trabalho.

Tabela 7- Resultados encontrados pelos métodos PCA, M-PCA e Newton.

Sistemas	Quantidade de raízes encontradas			Total de raízes possíveis
	PCA	M-PCA	Newton	
Sistema Trigonométrico (6)	9	11	11	13
Bini e Mourrain (7)	7	8	6	8

6. CONCLUSÕES

Diante dos resultados obtidos, pode-se observar que os métodos de otimização estocásticos Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA) são viáveis para encontrar soluções de um sistema de equações não linear, através da abordagem sugerida, visto que foram capazes de encontrar soluções para os dois sistemas propostos.

Quanto ao número de raízes encontradas, para o primeiro problema, o Algoritmo de Colisão de Múltiplas Partículas (M-PCA) e o Método de Newton tiveram os resultados mais satisfatórios. Já para o segundo problema, os métodos estocásticos tiveram melhor desempenho, com destaque para o Algoritmo de Colisão de Múltiplas Partículas (M-PCA) que encontrou todas as raízes.

No que tange ao número de avaliações da função objetivo, fica evidente a superioridade do Método de Newton, tanto pelo baixo número indicado pelo NAF, quanto pelo tempo total de execução. Ainda nesse tópico, cabe ressaltar que o número alto de execuções do Algoritmo de Colisão de Múltiplas Partículas (M-PCA), mesmo em relação ao Algoritmo de Colisão de Partículas (PCA), se dá devido ao não uso de uma computação de alto desempenho. Nesta situação, espera-se um resultado ainda mais satisfatório do Algoritmo de Colisão de Múltiplas Partículas (M-PCA).

Também é valido atentar para a praticidade de implementação e execução dos métodos estocásticos, visto que estes não necessitam do cálculo das derivadas parciais das funções do sistema. Por exemplo, para o segundo problema (7) usam-se 9 derivadas parciais. Além de apresentarem baixa dependência da estimativa inicial (fato que justifica as raízes não encontradas pelo método de Newton ao longo das 100 execuções).

Dessa forma, confrontando os resultados exibidos (número raízes encontradas e NAF), os tempos de execução e a praticidade de implementação dos métodos, conclui-se que ambos Algoritmo de Colisão de Partículas (PCA) e Algoritmo de Colisão de Múltiplas Partículas (M-PCA) estão aptos a serem aplicados a este tipo de problemática.

Como expectativa para trabalhos futuros, pretende-se utilizar os métodos estocásticos descritos nesse trabalho (Algoritmo de Colisão de Partículas (PCA) e sua variante) em aplicações de maior relevância, tais como na estimativa de parâmetros em corpos hídricos e em problemas de transferência de calor, dentre outros.

Agradecimentos

Os autores deste trabalho agradecem à Universidade Federal Fluminense pela bolsa de Iniciação Científica (PIBIC), Edital PIBIC/ CNPq/ UFF 2018/2019.

Referências

- [1] Burden, R. L., Faires, D. J.; Burden, A. M. *Numerical Analysis*.10.ed. Boston: Cengage Learning, 2016.
- [2] Cunha, M. C. C. *Métodos Numéricos*.2.ed. São Paulo: Unicamp, 2017, 81-86.
- [3] Holtz, G. C. C. *Traçado automático de envoltórias de esforços em estruturas planas utilizando um algoritmo evolucionário*. Dissertação de Pós-Graduação do título de Mestre em Engenharia Civil, PUC-RJ, RJ, Brasil, 2005, 35-38.
- [4] Luz, E. F. P.; Becceneri, J. C.; Campos Velho, H. F. *Uma nova metaheurística baseada em algoritmo de colisão de múltiplas partículas*. Simpósio de Pesquisa Operacional da Marinha (SPOLM), Rio de Janeiro, RJ, Brasil, 2008.
- [5] Oliveira e Martinez, G. L. e A. L. M. *Um estudo introdutório do método de Newton para minimização funções de uma ou várias variáveis*. CMAC Sudeste 2013, Bauru, São Paulo, Brasil.
- [6] Sacco, W. F.; Oliveira, C. R. E.; Pereira, C. M. N. *Two stochastic optimization algorithms applied to nuclear reactor core design*. In: Nuclear Energy, v. 48, n. 6, 2006, 525-539.
- [7] Santos, T. M. dos. *Um estudo sobre a resolução de sistemas não lineares: perspectivas teóricas e aplicações*. Tese de Pós-Graduação do título de Doutora em Matemática Aplicada, Unicamp, SP, Brasil, 2016.
- [8] Silva, M. R. *Aplicação de um algoritmo híbrido estocástico-determinístico na solução de um problema de direção automotiva*. XXXVII Encontro nacional de Engenharia de Produção, 2017, 80-101.
- [9] Telles, W. R. *Previsão do comportamento hidráulico de um rio com base na estimativa de coeficientes que controlam seu escoamento. Estudo de caso: Rio Bengalas, Nova Friburgo-RJ*. Tese de Pós-Graduação do título de Doutor em Modelagem Computacional, Instituto Politécnico, UERJ, Nova Friburgo, RJ, 2014, 77-83.
- [10] Viana, P. E. S. *Método de Simulação Estocástica Aplicados em Problemas Não Lineares*. Dissertação de Pós-Graduação do título de Mestre em Modelagem Computacional, UFF, Volta Redonda, RJ, 2017, 12-31.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

UTILIZAÇÃO DO MÉTODO DE MONTE CARLO PARA A RESOLUÇÃO DE PROBLEMAS COMPLEXOS DE INTEGRAÇÃO

Vinícius Brum Ferreira¹ - brum_vinicio@id.uff.br

Thiago Jordem Pereira² - tjordem@id.uff.br

^{1,2} Departamento de Ciências Exatas, Biológicas e da Terra, INFES, UFF, Santo Antônio de Pádua, RJ

Um dos conceitos mais importantes do cálculo é o de integral, que por sua vez possui aplicações em vários problemas em diversas áreas do conhecimento. Geralmente, alguns problemas típicos de integração envolvem funções complexas e multidimensionais, de modo que suas soluções são difíceis ou inviáveis de serem obtidas através de técnicas de integração convencionais. Neste caso, pode-se recorrer ao método de Monte Carlo que é uma alternativa eficiente para problemas de integração complexos. Este método de simulação estocástica baseia-se em técnicas de amostragens aleatórias, nas quais geram-se amostras a partir de uma distribuição de probabilidades para aproximar o problema de interesse. O objetivo deste trabalho é estudar a eficiência do método de Monte Carlo na resolução de problemas complexos de integração, onde se analisa a quantidade de amostras geradas, o erro de aproximação e o esforço computacional.

Palavras-chave Integrais, Simulação Estocástica, Método de Monte Carlo

Referências

- [1] Andrieu, C. and Freitas, N. D. and Doucet, A. and Jordan, M. I. 2003. An Introduction to MCMC for Machine Learning, *Machine Learning*, nº 50, pp. 5–43.
- [2] Leithold, L. 2004. *O Cálculo Com Geometria Analítica*. 3º ed., v. 1, Rio de Janeiro: Harbra.
- [3] Metropolis, N. and Ulam, S. 1949. The Monte Carlo Method, *Journal of the American Statistical Association*, v. 44, nº 247, pp. 335–341.
- [4] Robert, C. P. and Casella, G. 2002. *Monte Carlo Statistical Methods*. 2º ed., v. 1, USA: Springer Texts in Statistics.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

DINÂMICAS DE INFECÇÃO E TRATAMENTO DO VÍRUS DA HEPATITE C UTILIZANDO MODELO MATEMÁTICO COM ESTRUTURA ETÁRIA

Jéssica Garbero da Silva¹ - jessica.garbero@ice.ufjf.br

Bárbara de Melo Quintela² - barbara@ice.ufjf.br

Rodrigo Weber dos Santos² - rodrigo.weber@ufjf.edu.br

Marcelo Lobosco² - marcelo.lobosco@ufjf.edu.br

¹Instituto de Ciências Exatas (ICE), Universidade Federal de Juiz de Fora (UFJF)

²Laboratório de Fisiologia Computacional (FISIOCOMP), DCC/ICE/UFJF.

Abstract. According to the World Health Organization, infectious diseases caused by pathogens remain among the main causes of death and affect mainly low income countries. Host-pathogen interaction is extremely complex and its better understanding is crucial to human health improvement. This work aims to use an age-based mathematical model, previously validated, to better understand the dynamics of hepatitis C virus (HCV). The proposed model was able to fit patient data treated with Direct-acting Antiviral Agents, DAA. However, it was not able to fully represent experimental data of patients treated with a combination of DAA drugs. The model was improved to make the parameter that represents the infected cells decay to be dependent of the age of infection. We expect to use this model to better understand the influence of host factors specifically related to immune response to HCV and improve therapy. Moreover, it is expected to benefit from this model to understand other viruses of the same family as Yellow Fever virus (YFV), Dengue Virus (DNV), Zika Virus (ZNV) and Chikungunya Virus (CHV).

Keywords: Viral Dynamics, Computational Biology, Computational Immunology, Mathematical Modeling, Differential Equations

1. INTRODUÇÃO

A hepatite C crônica é uma doença que atinge entre 130–150 milhões de pessoas no mundo e é a principal causa de cirrose e câncer no fígado (WHO, 2016). O vírus da hepatite C (HCV) é constituído de uma molécula de RNA de orientação positiva (RNA+) linear de filamento único com cerca de 9600 nucleotídeos em seu genoma e foi classificada como pertencente ao gênero *hepacivirus* e família *flaviridae* (GASTAMINZA et al., 2008).

Por muitos anos a replicação do HCV não era completamente compreendida devido a dificuldade de desenvolver um sistema de cultura in vitro do HCV. Entretanto, o desenvolvimento nas últimas décadas de um sistema de cultura de células infectadas pelo HCV permitiu a investigação de processos que governam a replicação e outras características do ciclo de vida do HCV (APPEL et al., 2006; ELLIOT; ARMSTRONG; MCLAUCHLAN, 2009; AFZAL et al., 2015). Além disso, novos métodos de distinção e quantificação de ambas as moléculas positivas e negativas foram desenvolvidos e melhorados o que permite estudar seus comportamentos (BESSAUD et al., 2001).

O HCV afeta principalmente as células do fígado, *hepatócitos*. Após a entrada no hepatócito, o RNA+ é liberado e traduzido em uma poliproteína a partir da qual todas as proteínas necessárias para a replicação do HCV são produzidas (QUINKERT; BARTENSCHLAGER; LOHMANN, 2005). As proteínas formam complexos de replicação junto com uma cópia de RNA de orientação negativa (RNA-) que é utilizado como modelo para a replicação de novos RNA+. As cópias de RNA+ criadas podem ser usadas para serem novamente traduzidas, replicadas, criando novos complexos de replicação ou montadas em partículas virais para serem exportadas (QUINKERT; BARTENSCHLAGER; LOHMANN, 2005).

Modelos baseados em estrutura etária tem sido amplamente utilizados no estudo epidemiológico de doenças infecciosas como o HIV (THIEME; CASTILLO-CHAVEZ, 1993), hepatite C (MARTCHEVA; CASTILLO-CHAVEZ, 2003) e tuberculose (CASTILLO-CHAVEZ; FENG, 1997; THIEME; CASTILLO-CHAVEZ, 2002). Uma vantagem de usar tal abordagem é a possibilidade de considerar que indivíduos ou células com idades distintas poderiam se comportar de maneira diferente (LI; BRAUER, 2008). Usando essa abordagem na modelagem da dinâmica do vírus dentro de um hospedeiro permite uma representação realista da biologia da infecção, na qual a taxa de produção e liberação de novos vírus não é constante, mas depende do tempo que a célula está infectada. Além disso, o modelo também pode contabilizar taxa de morte de uma célula infectada que depende do tempo que a célula está infectada.

Trabalhos anteriores utilizaram um modelo baseado em estrutura etária para representar a replicação do RNA+ dentro de uma célula infectada e o seu comportamento durante o tratamento com medicamentos de ação direta (*Direct-acting antiviral agents, DAAs*) (GUEDJ et al., 2013; RONG et al., 2013; RONG; PERELSON, 2013). Foi acrescentada uma equação ao modelo intracelular para representar também o RNA- no complexo de replicação (QUINTELA, 2015). Foi mostrado que apenas as duas equações não foram suficientes para representar dados experimentais de transfecção. Para tanto, foi necessário incluir uma terceira equação para representar o RNA+ assim que entra na célula, no processo de tradução, para ajustar os parâmetros do modelo aos dados disponíveis (QUINTELA et al., 2017, 2018).

O presente trabalho apresenta uma continuidade dos trabalhos anteriores supracitados e sua principal contribuição é o estudo da influência dos parâmetros dependentes do tempo de infecção. Mais especificamente, são apresentadas as vantagens de utilizar parâmetros dependentes do tempo de infecção para melhor representar algumas características biológicas. São analisados dois parâmetros neste trabalho: ρ e δ , que respectivamente representam a taxa de exportação de vírus a partir de uma célula infectada e a taxa de decaimento de uma célula infectada. A escolha dos parâmetros se baseia em experimentos que mostra que os vírus só começam a ser exportados após 12h que o RNA do vírus entra na célula (KEUM et al., 2012) e, que mostram que a taxa de morte dos hepatócitos infectados pelo vírus da hepatite C é maior que a de células não infectadas (LIM et al., 2014; MASALOVA et al., 2017).

2. MATERIAIS E MÉTODOS

O modelo com três equações para a dinâmica intracelular de RNA+ e RNA- do HCV tanto disponível para tradução quanto sua dinâmica no complexo de replicação que foi utilizado no presente trabalho é apresentado a seguir.

2.1 Modelo de Infecção Intracelular

$$\begin{cases} \frac{dR_t}{dt} = \theta R_p - (\sigma + \rho(a) + \mu_t) R_t, \\ \frac{dR_p}{dt} = \alpha R_n + \sigma R_t - (\theta + \rho(a) + \mu_c) R_p, \\ \frac{dR_n}{dt} = r(1 - \frac{R_n}{R_{max}}) R_p - \mu_c R_n, \\ R_t(0) = R_{t_0}, R_p(0) = R_{p_0}, R_n(0) = R_{n_0}, \end{cases} \quad (1)$$

onde R_t representa o RNA+ em tradução, R_p representa o RNA+ no complexo de replicação e R_n representa o RNA- no complexo de replicação. Foi considerado que tanto RNA+ quanto RNA- dentro do complexo de replicação possuem mesmo decaimento μ_c . O RNA+ em tradução possui decaimento μ_t . Os parâmetros que mais influenciam na quantidade de vírus sendo produzidos são a taxa de replicação de RNA+, α e a taxa de replicação de RNA-, r , que foi considerada em um termo logístico para delimitar o número de complexos de replicação que pode ser formado, dado pelo parâmetro R_{max} . A taxa de exportação ρ também influencia na quantidade de vírus que está no plasma. Outros parâmetros menos influentes são as taxas em que o RNA+ migra entre tradução e replicação, θ e σ .

As condições iniciais dependem do cenário de simulação. Para simular uma infecção *in vivo*, considera-se que apenas um vírus é capaz de atravessar a membrana da célula e que existem mecanismos que impedem outros vírus do mesmo tipo de entrar na mesma célula (BLACKARD; SHERMAN, 2007; ALIZON, 2013). Portanto, para representar tal cenário as condições iniciais seriam $R_{t_0} = 1$, $R_{p_0} = 0$ e $R_{n_0} = 0$. Já para simular uma transfecção de vírus em uma cultura de células pode-se assumir como valor inicial de R_{t_0} a quantidade que foi adotada no experimento *in vitro* e os outros valores permanecem iguais a zero.

2.2 Modelo de Infecção e Tratamento Baseado em Estrutura Etária

Para analisar os efeitos do tratamento juntamente com as dinâmicas intracelulares junta-se o modelo mostrado em 2.1 a um modelo de infecção para chegar no modelo baseado em estrutura etária mostrado abaixo:

$$\begin{cases} \frac{dT}{dt} = s - \beta V T - dT \\ \frac{dV}{dt} = (1 - e_s) \int_0^\infty \rho(a) R_s I da - cV \\ \frac{\partial I}{\partial t} + \frac{\partial R_n}{\partial a} = -\delta(a) I \\ \frac{\partial R_t}{\partial t} + \frac{\partial R_t}{\partial a} = \theta R_p - (\sigma + (1 - \epsilon_s)\rho(a) + \kappa_t \mu_t) R_t, \\ \frac{\partial R_p}{\partial t} + \frac{\partial R_p}{\partial a} = (1 - \epsilon_\alpha) \alpha R_n + \sigma R_t - (\theta + (1 - \epsilon_s)\rho(a) + \kappa_c \mu_c) R_p, \\ \frac{\partial R_n}{\partial t} + \frac{\partial R_n}{\partial a} = (1 - \epsilon_r)r(1 - \frac{R_n}{R_{max}}) R_p - \kappa_c \mu_c R_n, \\ T(0) = T_0, V(0) = V_0, I(0, t) = \frac{dI}{da}, I(a, 0) = \bar{I}(a) \\ R_t(0, t) = R_{t_0}, R_t(a, 0) = \bar{R}_t(a), \\ R_p(0, t) = R_{p_0}, R_p(a, 0) = \bar{R}_p(a), \\ R_n(0, t) = R_{n_0}, R_n(a, 0) = \bar{R}_n(a) \end{cases} \quad (2)$$

onde os parâmetros ϵ_x representam a eficácia do medicamento em bloquear ou reduzir replicação e exportação do vírus e os parâmetros κ_x representam um fator de multiplicação para o decaimento.

Para o modelo de infecção considera-se que células alvo T se tornam infectadas I a partir do contato com vírus V a uma taxa β . Como as células do fígado se regeneram rápido adotou-se valor constante s para fonte e um decaimento d a partir de estimativas feitas em trabalhos anteriores (GUEDJ et al., 2013). A quantidade de vírus no plasma é calculada a partir de quantos vírus estão sendo exportados pelas células infectadas. Usa-se uma constante c estimada para eliminação dos vírus pelo sistema imune (GUEDJ et al., 2013).

As células infectadas apenas apresentam decaimento a uma taxa δ . As demais equações representam o RNA dentro das células infectadas e foram explicadas na seção 2.1.

Para simular o tratamento, primeiramente simulamos uma infecção com apenas as equações diferenciais ordinárias (Eq. 1) e utilizamos os valores de R_t , R_p e R_n como as condições iniciais $\overline{R_t}$, $\overline{R_p}$, e $\overline{R_n}$.

2.3 Resolução Numérica

O domínio adotado para a estrutura etária foi de 50 dias, que é considerado um tempo razoável de vida para uma célula do fígado. As simulações apresentadas no presente trabalho foram de dois dias de tratamento, por ser um período de simulação validado anteriormente com dados experimentais (QUINTELÀ et al., 2017). A Tabela 2.3 apresenta os parâmetros base utilizados nas simulações.

Table 1- Tabela com parâmetros que foram utilizados nas simulações. São apresentados seus nomes, valores, unidades e significado biológico.

Parâmetros	Valor	Unidade	Significado biológico
d	0.01	dia ⁻¹	decaimento de células alvo
s	$1.3e^5$	dia ⁻¹	fonte de células alvo
δ	0.62	dia ⁻¹	decaimento de células infectadas
β	$5e^{-8}$	dia ⁻¹	taxa de infecção
c	22.3	dia ⁻¹	eliminação de vírus pelo sistema imune
ρ	8.18	dia ⁻¹	exportação de vírus para o plasma
α	30	dia ⁻¹	replicação do RNA+
R_{max}	50	moléculas	limite para complexos de replicação (RNA-)
r	1.61	dia ⁻¹	replicação de RNA-
τ	0.5	dia	atraso para exportação de vírus
k	0.8	-	coeficiente da função exponencial
μ_t	0.8	dia ⁻¹	decaimento do RNA+ em tradução
μ_c	0.89	dia ⁻¹	decaimento de RNA+ e RNA- no complex. de repl.
σ	1.3	dia ⁻¹	taxa que RNA+ migra para compl. de repl.
θ	1.2	dia ⁻¹	taxa que RNA+ migra para tradução

Utilizou-se Euler explícito para as EDOs e para a resolução das EDPs foi utilizado o Método das Linhas (*Method of Lines*, MOL), e as derivadas foram aproximadas por diferenças finitas de primeira ordem. Usou-se para o tempo de simulação, o passo $\Delta t = 0.01$ e para as classes etárias (ages) $\Delta a = 0.1$, mantendo-se a relação necessária para estabilidade do método.

Para cenários de 2 dias de tratamento utilizando linguagem de programação C++ em um processador Intel core i5 2.9GHz as simulações levam menos de 40ms.

3. RESULTADOS E DISCUSSÃO

O modelo matemático baseado em estrutura etária formado pelas 6 equações foi capaz de ajustar dados de pacientes humanos para 2 dias de tratamento com drogas antivirais de ação direta (QUINTELÀ et al., 2018). Como não foi capaz de representar os resultados para um tempo maior de simulação investigou-se efeitos do decaimento da célula infectada que nos trabalhos anteriores foi mantido constante.

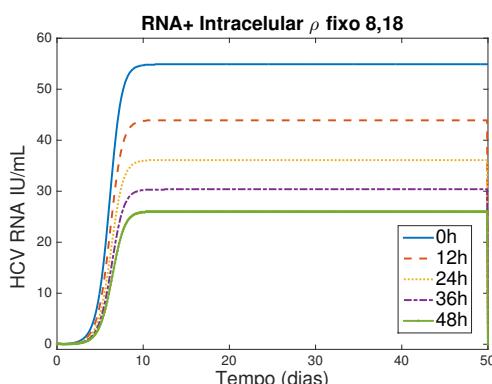
3.1 Variação de ρ

O parâmetro ρ representa a exportação dos vírus da célula infectada. A partir de observações de que os vírus só começam a ser exportados após 12h de infecção (KEUM et al., 2012), optou-se por tornar o ρ dependente da estrutura etária utilizando a função exponencial mostrada a seguir para evitar descontinuidades.

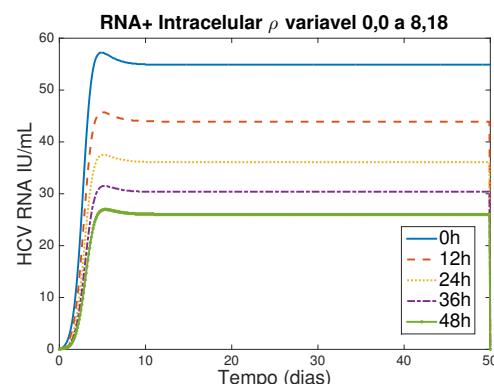
$$\rho(a) = \begin{cases} 0, & a < \tau, \\ (1 - e^{-k(a-\tau)})\rho, & a \geq \tau, \end{cases}$$

onde $k = 0.8$ foi estimado e τ representa o tempo que inicia a exportação (QUINTELÀ et al., 2018).

As Figuras 1(a) e (b) mostram a variação dos valores de RNA+ dentro de uma célula em diferentes passos do tempo de simulação do tratamento. A duração do tratamento são 2 dias e, portanto, é mostrado os valores nas 50 classes etárias no tempo inicial, após 12h, 24h, 36h e 48h de simulação.



(a) Parâmetro ρ constante em 8.18.



(b) Parâmetro ρ variável de 0,0 a 8,18.

Figure 1- Diferença no perfil de RNA+ intracelular se mantiver o parâmetro ρ em um valor constante. O valor 8.18 foi estimado em trabalhos anteriores.

A influência de se manter o ρ constante ou dependente da estrutura etária na carga viral pode ser vista na Figura 2.

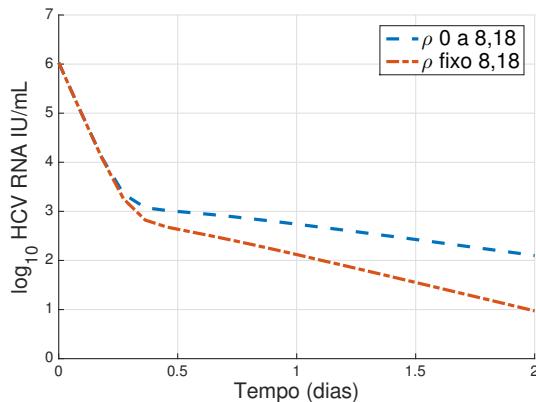


Figure 2- Variação da quantidade de vírus no plasma fazendo-se ρ constante igual a 8.18 ou variável iniciando com zero e aumentando exponencialmente a partir de 12h de simulação até atingir o valor estimado 8.18.

3.2 Variação de δ

Um hepatócito infectado pelo HCV pode ter sua morte programada (apoptose) por influência do vírus (LIM et al., 2014; MASALOVA et al., 2017). Os processos não são completamente compreendidos e se torna interessante poder utilizar o modelo para investigar o aumento da taxa de decaimento das células com o tempo de infecção.

A Figura 3 mostra os efeitos na carga viral ao tornar os valores de δ dependentes do tempo de infecção. A curva inicial não apresenta diferença assim como na variação do parâmetro ρ . Observa-se que a influência se torna maior com o tempo, conforme esperado. Ao aumentar o valor de δ , mais células morrem com o tempo o que faz com que um número menor continue exportanto vírus. Com isso observa-se a diminuição dos vírus no plasma com o tempo.

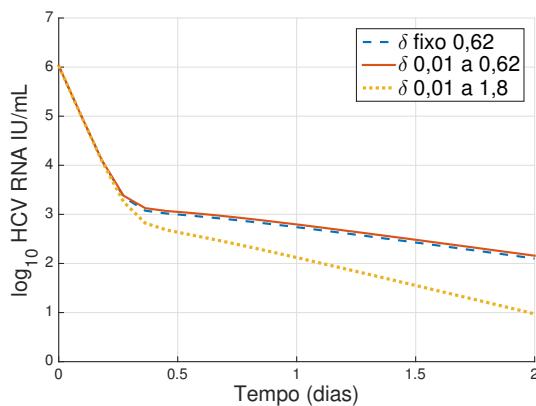


Figure 3- Variação da quantidade de vírus no plasma fazendo-se δ constante igual a 0.62 ou variável iniciando com 0.01 e aumentando exponencialmente até atingir o valor estimado 0.68 e iniciando com 0.01 e aumentando até 1.8. Usou-se o valor inicial como 0.01 por ser a taxa de decaimento usada para uma célula alvo antes de ser infectada.

É importante observar que a indução da morte da célula infectada causa danos ao fígado

principalmente a longo prazo.

4. CONCLUSÕES

O presente trabalho apresentou a influência de tornar parâmetros dependentes do tempo na representação das dinâmicas do vírus da hepatite C. Foram utilizados como exemplos a taxa de exportação de vírus e a taxa de morte de uma célula infectada em um modelo matemático de equações diferenciais com estrutura etária.

Espera-se que o mesmo modelo possa ser utilizado para melhor compreensão dos processos intracelulares da dinâmica desse vírus como também de outros vírus da mesma família *flaviridae* como os vírus da Dengue, Febre Amarela, Zika e Chikungunya. Medicamentos utilizados para tratamento da hepatite C foram utilizados em tratamentos de Zika (SACRAMENTO et al., 2017) e Chikungunya (FERREIRA et al., 2018) com sucesso. No entanto, o comportamento desses vírus e suas influências no organismo não são completamente conhecidos.

Agradecimentos

Os autores gostariam de agradecer a CAPES, CNPq, FAPEMIG e UFJF pelo apoio a execução do projeto.

REFERENCES

- AFZAL, M. S. et al. Regulation of Core Expression During the hepatitis C Virus Life Cycle. *J Gen Virol*, v. 96, p. 311–21, 2015.
- ALIZON, S. Co-infection and super-infection models in evolutionary epidemiology. *Interface Focus*, v. 3, 2013.
- APPEL, N. et al. From Structure to Function: New Insights into hepatitis C Virus RNA Replication. *J Biol Chem*, v. 281, n. 14, p. 9833–9836, 2006.
- BESSAUD, M. et al. Development of a Taqman RT-PCR assay for the detection and quantification of negatively stranded RNA of human enteroviruses: evidence for false-priming and improvement by tagged RT-PCR. *J Virol Methods*, v. 153, p. 182–9, 2001.
- BLACKARD, J. T.; SHERMAN, K. E. Hepatitis c virus coinfection and superinfection. *The Journal of Infectious Diseases*, v. 145, p. 519–24, 2007.
- CASTILLO-CHAVEZ, C.; FENG, Z. To treat or not to treat: The case of tuberculosis. *JMB*, v. 35, p. 629–56, 1997.
- ELLIOT, R. M.; ARMSTRONG, V. J.; MCLAUCHLAN, J. Struct mol virol. In: _____. [S.l.]: Intl Medical Press, 2009. (Human Virus Guides), cap. 2.
- FERREIRA, A. C. et al. Beyond members of the flaviviridae family, sofosbuvir also inhibits chikungunya virus replication. *Antimicrob Agents Chemother*, 2018.
- GASTAMINZA, P. et al. Cellular Determinants of Hepatitis C Virus Assembly, Maturation, Degradation, and Secretion. *J Virol*, v. 82, n. 5, p. 2120–2129, 2008.
- GUEDJ, J. et al. Modeling Shows that the NS5A inhibitor dacatasvir has two modes of action and yields a shorter estimate of the hepatitis C virus half-life. *Proc Natl Acad Sci USA*, v. 110, n. 10, p. 3991–3996, 2013.

- KEUM, S. et al. The specific infectivity of hepatitis C virus changes through its life cycle. *Virol*, v. 433, p. 462–470, Nov 2012.
- LI, J.; BRAUER, F. Continuous-time age-structured models in population dynamics and epidemiology. In: BRAUER, F.; DRIESSCHE, P. van den; WU, J. (Ed.). *Math Epidemiol*. [S.l.]: Springer Berlin Heidelberg, 2008, (Lecture Notes in Mathematics, v. 1945). p. 205–227. ISBN 978-3-540-78910-9.
- LIM, E. U. et al. Hepatitis c virus-induced hepatocyte cell death and protection by inhibition of apoptosis. *Journal of General Virology*, 2014.
- MARTCHEVA, M.; CASTILLO-CHAVEZ, C. Diseases with chronic stage in population with varying size? *Math Biosci*, v. 182, p. 1–25, 2003.
- MASALOVA, O. V. et al. Modulation of cell death pathways by hepatitis c virus proteins in huh7.5 hepatoma cells. *International Journal of Molecular Science*, 2017.
- QUINKERT, D.; BARTENSCHLAGER, R.; LOHMANN, V. Quantitative Analysis of the Hepatitis C Virus Replication Complex. *J Virol*, v. 79, p. 13594–13605, 2005.
- QUINTELA, B. de M. *Acoplamento de Modelos Computacionais de Doenças Infecciosas*. Tese (Doutorado) — Programa de Pós-Graduação em Modelagem Computacional, Juiz de Fora-MG, 3 2015.
- QUINTELA, B. M. et al. An Age-based Multiscale Mathematical Model of the Hepatitis C Virus Life-cycle During Infection and Therapy: Including Translation and Replication. In: TORRES, I. in;; BUSTAMANTE, J.; SIERRA, D. (Ed.). *VII Latin American Congress on Biomedical Engineering CLAIB 2016*. Bucaramanga, Santander, Colombia: IFMBE Proceedings, 2017. v. 60, p. 508–511.
- QUINTELA, B. M. et al. A new age-structured multiscale model of the hepatitis c virus life-cycle during infection and therapy with direct-acting antiviral agents. *Frontiers in Microbiology*, v. 9, 2018.
- RONG, L. et al. Analysis of Hepatitis C Virus Decline during Treatment with the Protease Inhibitor Danoprevir Using a Multiscale Model. *PLOS Comput Biol*, v. 9, n. 3, p. e1002959, 2013.
- RONG, L.; PERELSON, A. S. Mathematical analysis of multiscale models for hepatitis C virus dynamics under therapy with direct-acting antiviral agents. *Math Biosci*, v. 245, n. 1, p. 22 – 30, 2013. ISSN 0025-5564. {SI} : {BIOCOMP} 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0025556413001223>>.
- SACRAMENTO, C. Q. et al. The clinically approved antiviral drug sofosbuvir inhibits zika virus replication. *Scientific Reports*, 2017.
- THIEME, H.; CASTILLO-CHAVEZ, C. How may infection-age-dependent infectivity affect the dynamics of HIV/AIDS? *SIAM J Appl Math*, v. 53, p. 1337–79, 1993.
- THIEME, H.; CASTILLO-CHAVEZ, C. A two-strain tuberculosis model with age infection. *SIAM J Appl Math*, v. 62, p. 1634–56, 2002.
- WHO. *Guidelines for the screening, care and treatment of persons with chronic hepatitis C infection*. 2016. Date accessed: July, 2016. Disponível em: <<http://www.who.int/mediacentre/factsheets/fs164/en/>>.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

DISCRETE INVERSE PROBLEM IN THE HODGKIN AND HUXLEY MODEL

Jemy Alex Mandujano Valle - jhimyunac@gmail.com

National Laboratory of Scientific Computation (LNCC), Brazil.

Abstract. *The Hodgkin–Huxley (H–H) model is a system of ordinary differential equations (ODE) that describes the action potential behavior in the squid neuron. In this work, we propose the Landweber iteration to solve the inverse problem, that is, the estimation of maximum conductances in the discrete H–H model. To obtain the discrete model, we apply the finite differences method in the continuous H–H model.*

Keywords: *Hodgkin and Huxley model, Finite differences method, Inverse problems, Landweber Iteration.*

1. INTRODUCTION

In 1952, a mathematical model was developed to describe the initiation and propagation of an action potential in a neuron. Thenceforth, the then called Hodgkin-Huxley model, named after its creators, has been used vastly in the world of physiology. To obtain this model, the authors used two experimental techniques known as space clamp and voltage clamp, achieving significant progress in the understanding of nerve cells. However, while some variables, such as the membrane potential, are easily measured by voltage-clamp experiments, many parameters are obtained through a tedious combination of experiments and data tuning.

Some recent works (Sun et al. 2011 and Fang et al. 2016) estimate maximum conductances in the H–H model. Both proposed an adaptive observer to determine unknown parameters. However, the observer proposed by Fang et al. (2016) is less conservative than the observer proposed by Sun et al. (2011). Buhry et al. (2011) and (2012) estimate the constant parameters of an ion channel (H–H model) using the differential evolution algorithm. Finally, Roudolph et al. (2004) and Pospischil et al. (2007), estimate synaptic conductances, given the potential of the membrane, on the passive membrane equation.

Thereby, the goal of the present paper is to apply the *Nonlinear Landweber method* to solve the inverse problem of recovering the maximum conductances in the H–H model. Madureira et al. (2017), Valle et al. (2017) and Valle et al. (2018), also used the same method but applied to a different problem (passive cable equation).

We give a brief overview of the paper. In Section 2, we describe the continuous and discrete H–H models. In Section 3, we present the inverse problem for the discrete case and compute the adjoint operator of the Landweber iteration. Section 4, we provide the related numerical results. Finally, we discuss our findings and show some conclusions in Section 5.

2. THE HODGKIN-HUXLEY MATHEMATICAL MODEL

The H-H model with initial conditions consists of the following ODE

$$\begin{cases} C \frac{dV}{dt} = I_{\text{ext}} - G_{Na}m^3h(V - E_{Na}) - G_Kn^4(V - E_K) - G_L(V - E_L); & t \in (0, T] \\ \frac{d\mathcal{X}}{dt} = (1 - \mathcal{X})\alpha_{\mathcal{X}}(V) - \mathcal{X}\beta_{\mathcal{X}}(V); & \mathcal{X} = m, n, h; \quad t \in (0, T] \\ V(0) = V_0, \quad m(0) = m_0, \quad n(0) = n_0, \quad h(0) = h_0, \end{cases} \quad (1)$$

where C is the membrane capacitance, V is the membrane potential, dV/dt is the rate of change in membrane potential, I_{ext} is the membrane external current. The constants G_{Na} , G_K and G_L are the maximal conductances for Na^+ , K^+ and leakage channels respectively; E_{Na} , E_K , E_L are the Nernst equilibrium potentials. The functions n , m and h represent the so-called activation term of the potassium channel and the activation and inactivation terms of the sodium channel, respectively. The terms V_0 , m_0 , n_0 and h_0 are the initial conditions for V , m , n and h , respectively. Also, the functions $\alpha_{\mathcal{X}}$ and $\beta_{\mathcal{X}}$ ($\mathcal{X} = m, n, h$) are written as:

$$\begin{aligned} \alpha_m &= \frac{(25-V)/10}{\exp((25-V)/10)-1}; & \alpha_h &= 0.07 \exp(-V/20); & \alpha_n &= \frac{(10-V)/100}{\exp((10-V)/10)-1}; \\ \beta_m &= 4 \exp(-V/18); & \beta_h &= \frac{1}{\exp((30-V)/10)+1}; & \beta_n &= 0.125 \exp(-V/80). \end{aligned}$$

We partition the domain in time $([0, T])$ using a mesh t_1, t_2, \dots, t_{nt} . We assume a uniform partition, so the difference between two consecutive points in time will be Δt . The point U_i represents the numerical approximation of $U(t_i)$. Here $t_i = (i - 1)\Delta t$ for $i = 1, \dots, nt$ and $\Delta t = T/(nt - 1)$. Applying finite differences (Forward Euler Method) in equation (1), for $i = 1, \dots, nt - 1$, we have the following Hodgkin and Huxley discrete model,

$$\begin{cases} C \frac{V_{i+1} - V_i}{\Delta t} = I_{\text{ext}} - G_{Na}m_i^3h_i(V_i - E_{Na}) - G_Kn_i^4(V_i - E_K) - G_L(V_i - E_L); \\ \frac{\mathcal{X}_{i+1} - \mathcal{X}_i}{\Delta t} = (1 - \mathcal{X}_i)\alpha_{\mathcal{X}_i}(V_i) - \mathcal{X}_i\beta_{\mathcal{X}_i}(V_i); & \mathcal{X} = m, n, h; \\ V_1 = V_0, \quad m_1 = m_0, \quad n_1 = n_0, \quad h_1 = h_0. \end{cases} \quad (2)$$

Functions $\alpha_{\mathcal{X}_i}$ and $\beta_{\mathcal{X}_i}$ satisfy the following equations:

$$\begin{aligned} \alpha_{m_i} &= \frac{(25-V_i)/10}{\exp((25-V_i)/10)-1}; & \alpha_{h_i} &= 0.07 \exp(-V_i/20); & \alpha_{n_i} &= \frac{(10-V_i)/100}{\exp((10-V_i)/10)-1}; \\ \beta_{m_i} &= 4 \exp(-V_i/18); & \beta_{h_i} &= \frac{1}{\exp((30-V_i)/10)+1}; & \beta_{n_i} &= 0.125 \exp(-V_i/80). \end{aligned}$$

We denote $\mathbf{V} = (V_1, \dots, V_{nt})$, $\mathbf{m} = (m_1, \dots, m_{nt})$, $\mathbf{n} = (n_1, \dots, n_{nt})$, $\mathbf{h} = (h_1, \dots, h_{nt})$ and $\mathbf{G} = (G_{Na}, G_K, G_L)$.

The objective of this work is to estimate \mathbf{G} , from (2), given the potential membrane measurement.

3. DISCRETE INVERSE PROBLEM

Consider the nonlinear operator $F : \mathbb{R}^3 \rightarrow \mathbb{R}^{nt}$, defined by

$$F(\mathbf{G}) = \mathbf{V}, \quad (3)$$

where \mathbf{V} solves (2). In practical terms, the data \mathbf{V} are obtained by measurements. We denote such measurements by \mathbf{V}^δ , where the noise level δ is assumed to be known and satisfies

$$\|\mathbf{V} - \mathbf{V}^\delta\|_{\mathbb{R}^{nt}} \leq \delta. \quad (4)$$

In this paper, we admit the existence of the inverse operator F^{-1} , but not its stability. To control instability, we use an iterative method of regularization (Landweber iteration).

Here, for $x = (x_1, \dots, x_{nt})$ and $y = (y_1, \dots, y_{nt})$, the inner product and the norm are defined by the following equations,

$$\langle x, y \rangle = \Delta t^3 \sum_{i=1}^{nt} x_i y_i \quad \text{and} \quad \|x\| = \sqrt{\Delta t^3 \sum_{i=1}^{nt} |x_i|^2}. \quad (5)$$

From the system of equations (2), we consider the inverse problem of finding an approximation for \mathbf{G} , given the noisy data \mathbf{V}^δ .

Given the initial guess $\mathbf{G}^{1,\delta}$, the Landweber approximation (Hanke et al. 1995, Bende et al. 1996 and Neubauer 2000), to estimate \mathbf{G} , is defined as following:

$$\mathbf{G}^{k+1,\delta} = \mathbf{G}^{k,\delta} + F'(\mathbf{G}^{k,\delta})^*. (\mathbf{V}^\delta - F(\mathbf{G}^{k,\delta})), \quad (6)$$

where $\mathbf{G}^{k,\delta} = (G_{Na}^{k,\delta}, G_K^{k,\delta}, G_L^{k,\delta})$ and $F'(\cdot)^*(\cdot)$ is adjoint of the directional derivative.

The iteration (6) stops at the minimum $k_* = k(\delta, \mathbf{V}^\delta)$, such that, for a given $\tau > 2$,

$$\|\mathbf{V}^\delta - F(\mathbf{G}^{k_*,\delta})\|_{\mathbb{R}^{nt}} \leq \tau \delta. \quad (7)$$

It is possible to show that, under certain conditions, $\mathbf{G}^{k_*,\delta}$ converges to a solution to $F(\mathbf{G}) = \mathbf{V}$ as $\delta \rightarrow 0$ (see [7], Theorem 2.6).

In the next subsection, we calculate, from (6), the adjoint of the directional derivative.

3.1 The adjoint operator

In Theorem 3.1.1, we calculate the operator $F'(\cdot)^*(\cdot)$ from Landweber iteration (6). Then, from Theorem 3.1.1 and algorithm (6), we have the following iteration

$$\left\{ \begin{array}{l} G_{Na}^{k+1,\delta} = G_{Na}^{k,\delta} + \Delta t^3 \sum_{i=1}^{nt} \left(m_i^{k,\delta} \right)^3 h_i^{k,\delta} (V_i^{k,\delta} - E_{Na}) U_i^{k,\delta}, \\ G_K^{k+1,\delta} = G_K^{k,\delta} + \Delta t^3 \sum_{i=1}^{nt} \left(n_i^{k,\delta} \right)^4 (V_i^{k,\delta} - E_K) U_i^{k,\delta}, \\ G_L^{k+1,\delta} = G_L^{k,\delta} + \Delta t^3 \sum_{i=1}^{nt} (V_i^{k,\delta} - E_L) U_i^{k,\delta}, \end{array} \right. \quad (8)$$

where $\mathbf{V}^{k,\delta}$, $\mathbf{m}^{k,\delta}$, $\mathbf{n}^{k,\delta}$, $\mathbf{h}^{k,\delta}$ solve (2), replacing \mathbf{G} by $\mathbf{G}^{k,\delta}$. Also, $U_i^{k,\delta}$ solves (13), replacing \mathbf{G} by $\mathbf{G}^{k,\delta}$ and \mathbf{V} by $\mathbf{V}^{k,\delta}$.

In this work, to obtain an approximation of \mathbf{G} , we used iteration (8).

Theorem 3.1.1. Consider the nonlinear operator F defined in (3), then

$$F'(\mathbf{G})^* \cdot (\mathbf{V}^\delta - F(\mathbf{G})) = \Delta t^3 \left(\sum_{i=1}^{nt} m_i^3 h_i (V_i - E_{Na}) U_i, \sum_{i=1}^{nt} n_i^4 (V_i - E_K) U_i, \sum_{i=1}^{nt} (V_i - E_L) U_i \right).$$

Proof. Evaluating \mathbf{G} in the operator F and from (2) we have \mathbf{V} , \mathbf{m} , \mathbf{n} and \mathbf{h} .

Let the vector $\boldsymbol{\theta} = (\theta_{Na}, \theta_K, \theta_L) \in \mathbb{R}^3$ and $\lambda \in \mathbb{R}$, then evaluating $\mathbf{G} + \lambda\boldsymbol{\theta}$ in the operator F , we have $F(\mathbf{G} + \lambda\boldsymbol{\theta}) = \mathbf{V}^\lambda$, where \mathbf{V}^λ , \mathbf{m}^λ , \mathbf{n}^λ and \mathbf{h}^λ solve

$$\begin{cases} C \frac{V_{i+1}^\lambda - V_i^\lambda}{\Delta t} = I_{ext} - (G_{Na} + \lambda\theta_{Na}) (m_i^\lambda)^3 (h_i^\lambda) (V_i^\lambda - E_{Na}) \\ \quad - (G_K + \lambda\theta_K) (n_i^\lambda)^4 (V_i^\lambda - E_K) - (G_L + \lambda\theta_L) (V_i^\lambda - E_L), \\ \frac{\mathcal{X}_{i+1} - \mathcal{X}_i}{\Delta t} = (1 - \mathcal{X}_i) \alpha_{\mathcal{X}_i}(V_i) - \mathcal{X}_i \beta_{\mathcal{X}_i}(V_i); \quad \mathcal{X} = m^\lambda, n^\lambda, h^\lambda, \\ V_1^\lambda(0) = V_0; \quad m_1^\lambda(0) = m_0; \quad n_1^\lambda(0) = n_0; \quad n_1^\lambda(0) = n_0. \end{cases} \quad (9)$$

We denote $\mathbf{W} = (W_1, \dots, W_{nt})$, $\mathbf{M} = (M_1, \dots, M_{nt})$, $\mathbf{N} = (N_1, \dots, N_{nt})$ and $\mathbf{H} = (H_1, \dots, H_{nt})$. The directional derivative of F at \mathbf{G} in the direction $\boldsymbol{\theta}$ is given by

$$\mathbf{W} = F'(\mathbf{G}).(\boldsymbol{\theta}) = \lim_{\lambda \rightarrow 0} \frac{F(\mathbf{G} + \lambda\boldsymbol{\theta}) - F(\mathbf{G})}{\lambda} = \langle \nabla F, \boldsymbol{\theta} \rangle. \quad (10)$$

Also, we denote the following limits

$$\mathbf{M} = \lim_{\lambda \rightarrow 0} \frac{\mathbf{m}^\lambda - \mathbf{m}}{\lambda}, \quad \mathbf{N} = \lim_{\lambda \rightarrow 0} \frac{\mathbf{n}^\lambda - \mathbf{n}}{\lambda}, \quad \mathbf{H} = \lim_{\lambda \rightarrow 0} \frac{\mathbf{h}^\lambda - \mathbf{h}}{\lambda}, \quad (11)$$

where \mathbf{M} , \mathbf{N} and \mathbf{H} are the directional derivatives of \mathbf{m} , \mathbf{n} and \mathbf{h} , respectively.

Considering the difference between (2) and (9), dividing by λ and taking the limit $\lambda \rightarrow 0$, we have the following equation

$$\begin{cases} C \frac{W_{i+1} - W_i}{\Delta t} + (G_{Na} m_i^3 h_i + G_K n_i^4 + G_L) W_i = -3G_{Na} m_i^2 M_i h_i (V_i - E_{Na}) \\ \quad - G_{Na} m_i^3 H_i (V_i - E_{Na}) - 4G_K n_i^3 N_i (V_i - E_K) \\ \quad - \theta_{Na} m_i^3 h_i (V_i - E_{Na}) - \theta_K n_i^4 (V_i - E_K) - \theta_L (V_i - E_L), \\ \frac{\mathcal{Y}_{i+1} - \mathcal{Y}_i}{\Delta t} + [\alpha_{\mathcal{X}_i}(V_i) + \beta_{\mathcal{X}_i}(V_i)] \mathcal{Y}_i = [(1 - \mathcal{X}_i) \alpha'_{\mathcal{X}_i}(V_i) - \mathcal{X}_i \beta'_{\mathcal{X}_i}(V_i)] W_i; \\ (\mathcal{X}, \mathcal{Y}) = (m, M), (n, N), (h, H), \\ W_1 = 0; \quad M_1 = 0; \quad N_1 = 0; \quad H_1 = 0. \end{cases} \quad (12)$$

This last equation is another system of coupled nonlinear differential equations, depending on the parameter $\boldsymbol{\theta} = (\theta_{Na}, \theta_K, \theta_L)$. Note that the variable $\boldsymbol{\theta}$ represents any point in space \mathbb{R}^3 .

We define the following system of equations

$$\left\{ \begin{array}{l} C \frac{U_i - U_{i-1}}{\Delta t} - (G_{Na} m_i^3 h_i + G_K n_i^4 + G_L) U_i \\ \quad - [(1 - m_i) \alpha'_{m_i}(V_i) - m_i \beta_{m_i}(V_i)] P_i \\ \quad - [(1 - n_i) \alpha'_{n_i}(V_i) - n_i \beta'_{n_i}(V_i)] Q_i \\ \quad - [(1 - h) \alpha'_{h_i}(V_i) - h \beta'_{h_i}(V_i)] R_i = V_i^\delta - V_i \\ \frac{P_i - P_{i-1}}{\Delta t} - [\alpha_{m_i}(V_i) + \beta_{m_i}(V_i)] P_i = -3G_{Na} m_i^2 h_i (V_i - E_{Na}) U_i \\ \frac{Q_i - Q_{i-1}}{\Delta t} - [\alpha_{n_i}(V_i) + \beta_{n_i}(V_i)] Q_i = -4G_K n_i^3 (V_i - E_K) U_i \\ \frac{R_i - R_{i-1}}{\Delta t} - [\alpha_{h_i}(V_i) + \beta_{h_i}(V_i)] R_i = -G_{Na} m_i^3 (V_i - E_{Na}) U_i \\ U_{nt} = 0; \quad P_{nt} = 0; \quad Q_{nt} = 0; \quad R_{nt} = 0. \end{array} \right. \quad (13)$$

We denote $\mathbf{U} = (U_1, \dots, U_{nt})$, $\mathbf{P} = (P_1, \dots, P_{nt})$, $\mathbf{Q} = (Q_1, \dots, Q_{nt})$ and $\mathbf{R} = (R_1, \dots, R_{nt})$.

By the definition of adjoint operator, we have

$$\langle F'(\mathbf{G})^* (\mathbf{V}^\delta - F(\mathbf{G})) , \boldsymbol{\theta} \rangle_{\mathbb{R}^3} = \langle \mathbf{V}^\delta - F(\mathbf{G}) , F'(\mathbf{G}).(\boldsymbol{\theta}) \rangle_{\mathbb{R}^{nt}}.$$

Combining the previous equation and (10) gives

$$\langle F'(\mathbf{G})^* (\mathbf{V}^\delta - F(\mathbf{G})) , \boldsymbol{\theta} \rangle_{\mathbb{R}^3} = \langle \mathbf{V}^\delta - F(\mathbf{G}) , \mathbf{W} \rangle_{\mathbb{R}^{nt}}.$$

By the definition of inner product (5), we have

$$\langle F'(\mathbf{G})^* (\mathbf{V}^\delta - F(\mathbf{G})) , \boldsymbol{\theta} \rangle_{\mathbb{R}^3} = \Delta t^3 \sum_{i=1}^{nt} (V_i^\delta - V_i) W_i.$$

We denote the last equality by Φ , then

$$\Phi = \frac{1}{\Delta t^3} \langle F'(\mathbf{G})^* (\mathbf{V}^\delta - F(\mathbf{G})) , \boldsymbol{\theta} \rangle_{\mathbb{R}^3} = \sum_{i=1}^{nt} (V_i^\delta - V_i) W_i. \quad (14)$$

From the previous equation and the first equality from (13), we obtain the following expression

$$\begin{aligned} \Phi &= \sum_{i=1}^{nt} C \frac{U_i - U_{i-1}}{\Delta t} W_i - \sum_i (G_{Na} m_i^3 h_i + G_K n_i^4 + G_L) U_i W_i \\ &\quad - \sum_{i=1}^{nt} [(1 - m_i) \alpha'_{m_i}(V_i) - m_i \beta_{m_i}(V_i)] P_i - \sum_{i=1}^{nt} [(1 - n_i) \alpha'_{n_i}(V_i) - n_i \beta'_{n_i}(V_i)] Q_i \\ &\quad - \sum_{i=1}^{nt} [(1 - h) \alpha'_{h_i}(V_i) - h \beta'_{h_i}(V_i)] R_i. \end{aligned} \quad (15)$$

From equations (12) and (13), the vector (W_1, U_{nt}) is equal to $(0, 0)$. Then, we obtain

$$\sum_{i=1}^{nt} C \frac{U_i - U_{i-1}}{\Delta t} W_i = - \sum_{i=1}^{nt} C \frac{W_{i+1} - W_i}{\Delta t} U_i. \quad (16)$$

Substituting (16) into (15), we have the following equality

$$\begin{aligned} \Phi = & - \sum_{i=1}^{nt} \left(C \frac{W_{i+1} - W_i}{\Delta t} + (G_{Na} m_i^3 h_i + G_K n_i^4 + G_L) W_i \right) U_i \\ & - \sum_{i=1}^{nt} [(1 - m_i) \alpha'_{m_i}(V_i) - m_i \beta_{m_i}(V_i)] P_i W_i - \sum_{i=1}^{nt} [(1 - n_i) \alpha'_{n_i}(V_i) - n_i \beta'_{n_i}(V_i)] Q_i W_i \\ & - \sum_{i=1}^{nt} [(1 - h) \alpha'_{h_i}(V_i) - h \beta'_{h_i}(V_i)] R_i W_i. \end{aligned}$$

Replacing the first equality from (12) in the previous equation, leads to

$$\begin{aligned} \Phi = & \sum_{i=1}^{nt} 3G_{Na} m_i^2 M_i h_i (V_i - E_{Na}) U_i + \sum_{i=1}^{nt} G_{Na} m_i^3 H_i (V_i - E_{Na}) U_i \\ & + \sum_{i=1}^{nt} 4G_K n_i^3 N_i (V_i - E_K) U_i + \sum_{i=1}^{nt} \theta_{Na} m_i^3 h_i (V_i - E_{Na}) U_i \\ & + \sum_{i=1}^{nt} \theta_K n_i^4 (V_i - E_K) U_i + \sum_{i=1}^{nt} \theta_L (V_i - E_L) U_i \\ & - \sum_{i=1}^{nt} [(1 - m_i) \alpha'_{m_i}(V_i) - m_i \beta_{m_i}(V_i)] P_i W_i - \sum_{i=1}^{nt} [(1 - n_i) \alpha'_{n_i}(V_i) - n_i \beta'_{n_i}(V_i)] Q_i W_i \\ & - \sum_{i=1}^{nt} [(1 - h) \alpha'_{h_i}(V_i) - h \beta'_{h_i}(V_i)] R_i W_i. \quad (17) \end{aligned}$$

Let $(\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i) \in \{(m_i, M_i, P_i), (n_i, N_i, Q_i), (h_i, H_i, R_i)\}$. Then, multiplying the second equation from (12) by \mathcal{Z}_i ,

$$\begin{aligned} \sum_{i=1}^{nt} \left(\frac{\mathcal{Y}_{i+1} - \mathcal{Y}_i}{\Delta t} + [\alpha_{\mathcal{X}_i}(V_i) + \beta_{\mathcal{X}_i}(V_i)] \mathcal{Y}_i \right) \mathcal{Z}_i \\ = \sum_{i=1}^{nt} [(1 - \mathcal{X}_i) \alpha'_{\mathcal{X}_i}(V_i) - \mathcal{X}_i \beta'_{\mathcal{X}_i}(V_i)] W_i \mathcal{Z}_i. \quad (18) \end{aligned}$$

From equations (12) and (13), the vector $(\mathcal{Y}_1, \mathcal{Z}_{nt}) \in \{(M_1, P_{nt}), (N_1, Q_{nt}), (H_1, R_{nt})\}$ equals $(0, 0)$. Then, we have

$$\sum_{i=1}^{nt} \frac{\mathcal{Y}_{i+1} - \mathcal{Y}_i}{\Delta t} \mathcal{Z}_i = - \sum_{i=1}^{nt} \frac{\mathcal{Z}_i - \mathcal{Z}_{i-1}}{\Delta t} \mathcal{Y}_i. \quad (19)$$

Replacing equation (19) in (18), we obtain

$$-\sum_{i=1}^{nt} \left(\frac{\mathcal{Z}_i - \mathcal{Z}_{i-1}}{\Delta t} - [\alpha_{\mathcal{X}_i}(V_i) + \beta_{\mathcal{X}_i}(V_i)]\mathcal{Z}_i \right) \mathcal{X}_i = \sum_{i=1}^{nt} [(1 - \mathcal{X}_i)\alpha'_{\mathcal{X}_i}(V_i) - \mathcal{X}_i\beta'_{\mathcal{X}_i}(V_i)]W_i\mathcal{Z}_i. \quad (20)$$

For $(\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i) = (m_i, M_i, P_i)$ into (20), leads to

$$-\sum_{i=1}^{nt} \left(\frac{P_i - P_{i-1}}{\Delta t} - [\alpha_{m_i}(V_i) + \beta_{m_i}(V_i)]P_i \right) M_i = \sum_{i=1}^{nt} [(1 - m_i)\alpha'_{m_i}(V_i) - m_i\beta'_{m_i}(V_i)]W_iP_i. \quad (21)$$

Substituting the second equation from (13) into (21),

$$\sum_{i=1}^{nt} 3G_{Na}m_i^2h_i(V_i - E_{Na})U_iM_i = \sum_{i=1}^{nt} [(1 - m_i)\alpha'_{m_i}(V_i) - m_i\beta'_{m_i}(V_i)]W_iP_i. \quad (22)$$

For $(\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i) = (n_i, N_i, Q_i)$ into (20), we have

$$-\sum_{i=1}^{nt} \left(\frac{Q_i - Q_{i-1}}{\Delta t} - [\alpha_{n_i}(V_i) + \beta_{n_i}(V_i)]Q_i \right) N_i = \sum_{i=1}^{nt} [(1 - n_i)\alpha'_{n_i}(V_i) - n_i\beta'_{n_i}(V_i)]W_iQ_i. \quad (23)$$

Substituting the third equation from (13) into (23),

$$\sum_{i=1}^{nt} 4G_Kn_i^3(V_i - E_K)U_iN_i = \sum_{i=1}^{nt} [(1 - n_i)\alpha'_{n_i}(V_i) - n_i\beta'_{n_i}(V_i)]W_iQ_i. \quad (24)$$

For $(\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i) = (h_i, H_i, R_i)$ into (20), we obtain

$$-\sum_{i=1}^{nt} \left(\frac{R_i - R_{i-1}}{\Delta t} - [\alpha_{h_i}(V_i) + \beta_{h_i}(V_i)]R_i \right) H_i = \sum_{i=1}^{nt} [(1 - h_i)\alpha'_{h_i}(V_i) - h_i\beta'_{h_i}(V_i)]W_iR_i. \quad (25)$$

Substituting the fourth equation from (13) into (25),

$$\sum_{i=1}^{nt} G_{Na}m_i^3(V_i - E_{Na})U_iH_i = \sum_{i=1}^{nt} [(1 - h_i)\alpha'_{h_i}(V_i) - h_i\beta'_{h_i}(V_i)]W_iR_i. \quad (26)$$

Substituting equations (22), (24) and (26) in (17), we have

$$\Phi = \sum_{i=1}^{nt} \theta_{Na} m_i^3 h_i (V_i - E_{Na}) U_i + \sum_{i=1}^{nt} \theta_K n_i^4 (V_i - E_K) U_i + \sum_{i=1}^{nt} \theta_L (V_i - E_L) U_i.$$

By the definition of inner product

$$\Phi = \left\langle \left(\sum_{i=1}^{nt} m_i^3 h_i (V_i - E_{Na}) U_i, \sum_{i=1}^{nt} n_i^4 (V_i - E_K) U_i, \sum_{i=1}^{nt} (V_i - E_L) U_i \right), \boldsymbol{\theta} \right\rangle_{\mathbb{R}^3}. \quad (27)$$

From equations (14) and (27), since $\boldsymbol{\theta} \in \mathbb{R}^3$ is arbitrary, we obtain

$$\begin{aligned} F'(\mathbf{G})^* (\mathbf{V}^\delta - F(\mathbf{G})) &= \\ \Delta t^3 \left(\sum_{i=1}^{nt} m_i^3 h_i (V_i - E_{Na}) U_i, \sum_{i=1}^{nt} n_i^4 (V_i - E_K) U_i, \sum_{i=1}^{nt} (V_i - E_L) U_i \right). \end{aligned}$$

□

4. NUMERICAL RESULTS

In this section, we show numerical results for the discrete inverse problem using the Landweber iteration. The parameters for the H-H model (2) are: $\Delta t = 0.01$, $nt = 1000$, $C = 1 [\mu F/cm^2]$, $I_{ext} = 10 [\mu A/cm^2]$, $E_{Na} = 50 [mV]$, $E_{Na} = 50 [mV]$, $E_K = -77 [mV]$, $E_L = -54.387 [mV]$, $V_0 = -15 [mV]$, $m_0 = 0.6$, $n_0 = 0.4$ and $h_0 = 0.4$. The goal of this section is to find $G_{Na} = 120 mS/cm^2$, $G_K = 36 mS/cm^2$ and $G_L = 0.3 mS/cm^3$.

To compare the results obtained, first we calculate \mathbf{V} from (2) given $\mathbf{G} = (120, 36, 0.3)$. We obtain \mathbf{V}^δ , for $\delta = 10^{-3}$, from the following equation

$$\mathbf{V}^\delta(t) = \mathbf{V}(t) + rand_\delta, \quad \text{for all } t \in [0, T],$$

where $rand_\delta$ is a uniformly distributed random variable in the interval $[-\delta, \delta]$. Now, we consider \mathbf{V} and \mathbf{G} unknown.

To obtain \mathbf{G} we use (8), given the initial guess $\mathbf{G}^{1,\delta} = (0, 0, 0)$. For $\tau = 4$ in the stopping criterion (7), the algorithm stops when $k_* = 853799$.

Figures 1-A, 1-B and 1-C show estimates for the maximum sodium, potassium, and leakage conductances, respectively.

Figure 2 shows the relative error between the actual and estimated values for the algorithm iterations. The relative error of \mathbf{G} is defined as

$$Error_k^{\mathbf{G}} = \frac{\|\mathbf{G} - \mathbf{G}^{k,\delta}\|_{\mathbb{R}^3}}{\|\mathbf{G}\|_{\mathbb{R}^3}} \times 100\%, \quad k = 1, \dots, 853799. \quad (28)$$

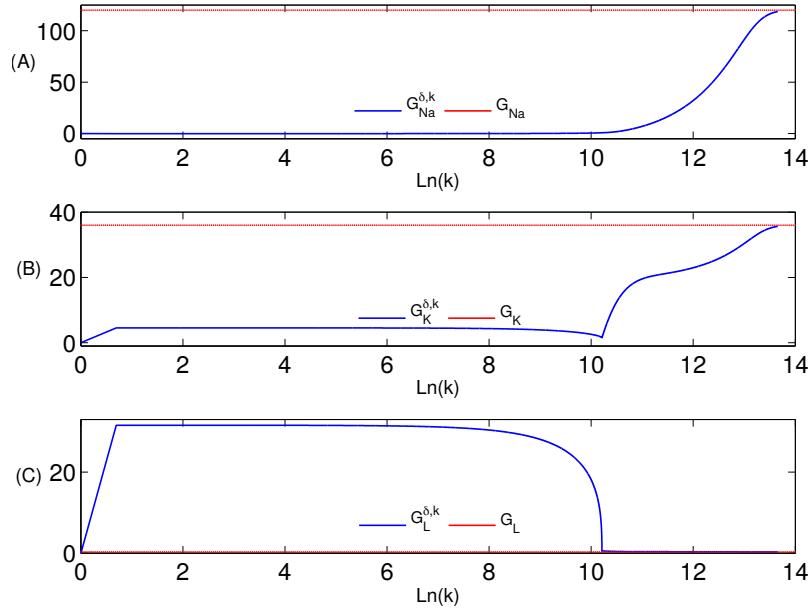


Figure 1- Estimation of the conductances G_{Na} (Subplot-A), G_K (Subplot-B) and G_L (Subplot-C).

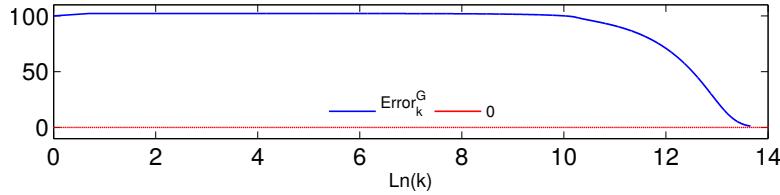


Figure 2- The relative error defined in (28).

5. CONCLUSIONS

In this work, we discretized the H-H model using the explicit Euler method. We considered the inverse problem of determining maximal conductances in the discrete model. To obtain the unknown parameters, we used the Landweber iteration. An essential contribution of the present work is the calculation of the adjoint operator of the Landweber iteration.

Compared to works Sun et al. (2011) and Fang et al. 2016., our approach has the advantage of allowing recovering the unknown parameters from noisy data. The computational results show that the method obtains a reasonable estimate for the parameters. It is important to notice that this work considers only computational data. In future studies, we plan to consider biological data as well.

Acknowledgements

The author would like to thank the support provided by CAPES.

REFERENCES

- [1] Buhry, Laure et al., 2011. Automated parameter estimation of the Hodgkin-Huxley model using the differential evolution algorithm: application to neuromimetic analog integrated circuits. *Neural computation*, v. 23, n. 10, p. 2599-2625.
- [2] Buhry, L., Pace, M. and Saïghi, S., 2012. Global parameter estimation of a Hodgkin-Huxley formalism using membrane voltage recordings: Application to neuro-mimetic analog integrated circuits. *Neurocomputing*, v. 81, p. 75-85.
- [3] Binder, A., Hanke, M., and Scherzer, O. 1996. On the Landweber iteration for nonlinear ill-posed problems. *Journal of Inverse and Ill-posed Problems*, v. 4, n. 5, p. 381-390.
- [4] Fang, L. I. A. O., and LOU, X. Y., 2016. State Estimation of Hodgkin-Huxley Model Based on Membrane Potential. *DEStech Transactions on Engineering and Technology Research*.
- [5] Hodgkin, A. L., and Huxley, A. F. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4), 500-544.
- [6] Hanke, M., Neubauer, A., and Scherzer, O., 1995. A convergence analysis of the Landweber iteration for nonlinear ill-posed problems. *Numerische Mathematik*, v. 72, n. 1, p. 21-37.
- [7] Kaltenbacher, B., Neubauer, A., and Scherzer, O. 2008. *Iterative regularization methods for nonlinear ill-posed problems* (Vol. 6). Walter de Gruyter.
- [8] Madureira, A. L. and Valle, J. A. M., 2017. Solução Númerica de um Problema Inverso em Neurociência via o Método de Landweber Não Linear. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 5, n. 1.
- [9] Neubauer, A., 2000. On Landweber iteration for nonlinear ill-posed problems in Hilbert scales. *Numerische Mathematik*, v. 85, n. 2, p. 309-328.
- [10] Pospischil, M., Piwkowska, Z., Rudolph, M., Bal, T., and Destexhe, A. 2007. Calculating event-triggered average synaptic conductances from the membrane potential. *Journal of Neurophysiology*, v. 97, n. 3, p. 2544-2552, 2007.
- [11] Pospischil, M., et al., 2007. Calculating event-triggered average synaptic conductances from the membrane potential. *Journal of Neurophysiology*, 97(3), 2544-2552.
- [12] Rudolph, M., et al., 2004. A method to estimate synaptic conductances from membrane potential fluctuations. *Journal of neurophysiology*, v. 91, n. 6, p. 2884-2896.
- [13] Sun, J., Deng, B., Wei, X., Jia, C., Wang, J., and Zhao, J., 2011. Parameter estimation in Hodgkin-Huxley model with adaptive method. In *Biomedical Engineering and Informatics (BMEI), 2011 4th International Conference on* (Vol. 4, pp. 1853-1857). IEEE.
- [14] Valle, J. A. M. and Madureira, A. L., 2017. Determinação de Parâmetros com Distribuição Espacial não Uniforme num Modelo de Neurociência via o Método de Landweber não Linear. *X Encontro Acadêmico de Modelagem Computacional*.
- [15] Valle, J. A. M., Madureira, A. L., and Leitão, A., 2018. A Computational Approach for the inverse problem of neuronal conductances determination. *arXiv preprint arXiv: 1810.05887*.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

EXISTÊNCIA E COMPORTAMENTO ASSINTÓTICO DE SOLUÇÕES PARA A EQUAÇÃO DA ONDA DE 4^a ORDEM ENVOLVENDO EXPOENTE VARIÁVEL

Willian dos Santos Panni¹ - willianpanni@id.uff.br

Jorge Ferreira¹ - ferreirajorge2012@gmail.com

¹Universidade Federal Fluminense

Resumo. Neste trabalho estudamos a existência de soluções fracas para uma equação de onda de quarta ordem envolvendo expoente variável não linear sobre um domínio limitado. Além disso, obtivemos um resultado relacionado com o decaimento exponencial e polinomial das soluções. Para demonstrar a existência de soluções fracas utilizamos o método de Faedo-Galerkin acoplado com diversos resultados clássicos de Análise Funcional, espaços de Lebesgue e Sobolev com expoente variável que podem ser encontrados em Fan e Zhao (2001) e desigualdades clássicas para o referido problema. Para o comportamento assintótico usamos uma técnica introduzida por Nakao (1995), tal método foi aplicado para uma equação de onda não linear e no presente trabalho fizemos uma adaptação para utilizarmos em nosso problema. Está em fase de conclusão a análise numérica e simulação da solução. Ressaltamos que a unicidade da solução fraca é um problema em aberto, entretanto estamos trabalhando para solucionar a aludida conjectura. Nosso trabalho está organizado da seguinte forma, na seção 1 definimos as hipóteses e mencionamos vários trabalhos envolvendo os operadores p , $p(x)$ e $\tilde{p}(x,t)$ -Laplaciano. Na seção 2 provamos a existência de soluções fracas. Na 3^a seção enunciamos o decaimento de energia. Finalmente, na seção 4 apresentamos as conclusões e os trabalhos futuros.

Palavras-chave: Equação da onda, Expoente variável, Método de Faedo-Galerkin, Existência de soluções fracas, Comportamento assintótico.

1. INTRODUÇÃO

Equações Diferenciais Parciais com expoente variável surgiram há aproximadamente duas décadas e mais precisamente iniciadas por matemáticos russos e atualmente aqui no Brasil existem poucos matemáticos trabalhando na referida área. Problemas envolvendo expoente variável tem inúmeras aplicações nas ciências aplicadas, por exemplo em fluidos eletro-reológicos, fluidos termo-reológicos de fluxo viscoso não Newtoniano, problemas em elasticidade não linear, restauração de imagem, equações de viga não lineares, processamento de imagens, entre outras.

Chipot e Savitska (2014) demonstraram a existência, unicidade e o comportamento assintótico para uma classe de problemas de difusão não local e não linear associados a um operador do tipo p -Laplaciano, onde uma quantidade não local estava presente no coeficiente de difusão.

Chabrowski e Fu (2005) estudaram a existência de soluções para problemas elípticos envolvendo o operador $p(x)$ -Laplaciano em um domínio limitado. Zhang e Zhou (2009) demonstra-

ram a existência, unicidade e o comportamento assintótico de soluções fracas para uma equação parabólica degenerada de quarta ordem com expoente variável, além do mais existem diversos artigos russos, chineses, alemães, árabes e brasileiros que estudaram a existência de soluções, unicidade, comportamento assintótico e blow up para problemas envolvendo o operador $p(x)$ -Laplaciano, por exemplo Antontsev, Chipot e Xie (2007), Alaoui, Messaoudi e Khenous (2014), Pinasco (2009), Gao e Fu (1999), Andrade, Silva e Fu (2012) e suas referências.

Os primeiros resultados na literatura relacionados com equações de onda e vigas envolvendo o operador $\vec{p}(x, t)$ -Laplaciano foram publicados por Antontsev (2011), Antontsev e Ferreira (2013), Antontsev e Ferreira (2014) e Ferreira e Messaoudi (2014).

Motivados por tais aplicações e trabalhos mencionados acima, estudamos o seguinte problema de valor inicial

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial t^2} + \Delta (|\Delta u|^{p(x)-2} \Delta u) - \Delta \frac{\partial u}{\partial t} + f \left(x, t, \frac{\partial u}{\partial t} \right) = g(x, t) \text{ em } Q_T, \\ u = \Delta u = 0 \text{ em } \partial\Omega \times (0, T), \\ u(x, 0) = u_0(x), \frac{\partial u(x, 0)}{\partial t} = u_1(x) \text{ em } Q_T. \end{array} \right. \quad (1)$$

Onde $\Omega \subset \mathbb{R}^N$ ($N \geq 3$) é um domínio limitado com fronteira $\partial\Omega$ suave, $0 < T < \infty$, $Q_T = \Omega \times (0, T)$ e as funções p , f , g , u_0 e u_1 satisfazem as seguintes hipóteses:

(H.1) $p : \bar{\Omega} \rightarrow (1, \infty)$ é log-Hölder contínua, isto é, existe uma constante $c > 0$ tal que

$$|p(x) - p(y)| \log |x - y| \leq c \quad \forall x, y \in \bar{\Omega} \quad (2)$$

e satisfaz

$$1 < p^- = \inf_{\bar{\Omega}} p(x) \leq p^+ = \sup_{\bar{\Omega}} p(x) < \frac{N}{2} \quad \forall x \in \bar{\Omega} \quad (3)$$

onde $\bar{\Omega}$ denota o fecho de Ω ;

(H.2) $f(x, t, s) \in C(\Omega \times [0, \infty) \times \mathbb{R})$ e existem constantes positivas c_1 , c_2 e c_3 tais que

$$f(x, t, s)s \geq c_1 |s|^{q(x)} - c_3 \quad (4)$$

e

$$|f(x, t, s)| \leq c_2 (|s|^{q(x)-1} + 1) \quad (5)$$

para todo $(x, t, s) \in \Omega \times [0, \infty) \times \mathbb{R}$, onde $q : \bar{\Omega} \rightarrow (1, \infty)$ é log-Hölder contínua satisfazendo

$$1 < q^- = \inf_{\bar{\Omega}} q(x) \leq q(x) < \frac{Np(x)}{N - 2p(x)} \quad \forall x \in \bar{\Omega}; \quad (6)$$

(H.3) $u_0 \in W^{2,p(x)}(\Omega) \cap W_0^{1,2}(\Omega)$, $u_1 \in L^2(\Omega)$ e $g \in L^{q'(x)}(Q_T)$ onde

$$\frac{1}{q(x)} + \frac{1}{q'(x)} = 1 \quad \forall x \in \bar{\Omega}. \quad (7)$$

O Problema (1) é uma extensão da equação de viga de Euler-Bernoulli com dissipação forte. Além de ser aplicável em fluidos eletro-reológicos e fluidos termo-reológicos, que possuem a característica de que sua viscosidade depende do campo elétrico do fluido e de sua temperatura. Podemos citar também outros campos de aplicações como por exemplo equações de onda, processos de filtração através de meios porosos, restauração e segmentos de imagens.

2. EXISTÊNCIA DE SOLUÇÕES FRACAS

Nessa seção definimos o conceito de solução fraca para o Problema (1), posteriormente aplicamos o método de Faedo-Galerkin no referido problema e enunciamos dois resultados importantes que auxiliam na demonstração do principal teorema dessa seção envolvendo a existência de soluções fracas.

Definição 1: A função $u : Q_T \rightarrow \mathbb{R}$ é uma solução fraca do Problema (1) se

$$u \in L^\infty \left(0, T; W_0^{2,p(x)}(\Omega) \right) \cap C \left(0, T; W_0^{1,2}(\Omega) \right), \quad (8)$$

$$\frac{\partial u}{\partial t} \in L^\infty \left(0, T; L^2(\Omega) \right) \cap L^2 \left(0, T; W_0^{1,2}(\Omega) \right) \cap L^{q(x)}(Q_T) \quad (9)$$

e

$$\begin{aligned} & - \int_{\Omega} \frac{\partial u(x, 0)}{\partial t} \varphi(x, 0) dx - \int_{Q_T} \frac{\partial u}{\partial t} \frac{\partial \varphi}{\partial t} dx dt + \int_{Q_T} |\Delta u|^{p(x)-2} \Delta u \Delta \varphi dx dt + \\ & \int_{Q_T} \nabla \frac{\partial u}{\partial t} \nabla \varphi dx dt + \int_{Q_T} f \left(x, t, \frac{\partial u}{\partial t} \right) \varphi dx dt = \int_{Q_T} g(x, t) \varphi dx dt, \end{aligned} \quad (10)$$

para todo $\varphi \in C^1(0, T; C_0^\infty(\Omega))$ com $\varphi(x, T) = 0$.

A existência de solução fraca para o Problema (1) é demonstrada através do método clássico de Faedo-Galerkin.

Suponhamos a seguinte sequência $\{\omega_i\}_{i=1}^\infty \subset C_0^\infty(\Omega)$ tal que $C_0^\infty(\Omega) = \overline{\bigcup_{n=1}^\infty V_n}^{C^2(\bar{\Omega})}$ e que $\{\omega_i\}_{i=1}^\infty$ seja uma base ortonormal completa em $L^2(\Omega)$, onde

$$V_n := \text{span} \{ \omega_1, \omega_2, \dots, \omega_n \} = [\omega_1, \omega_2, \dots, \omega_n].$$

Buscamos uma sequência de soluções aproximadas da seguinte forma

$$u_n(x, t) = \sum_{i=1}^n \eta_{ni}(t) \omega_i(x) \quad (11)$$

tal que $u_n(x, t) \in V_n$.

As funções $\eta_{ni}(t)$ são determinadas por um sistema de equações diferenciais ordinárias não lineares em relação a variável t . Definimos

$$P_n(t, \mu, v) : [0, T] \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (12)$$

tal que

$$\begin{aligned} (P_n(t, \mu, v))_j &= \int_{\Omega} \left| \sum_{i=1}^n \mu_i \Delta \omega_i \right|^{p(x)-2} \left(\sum_{i=1}^n \mu_i \Delta \omega_i \right) \omega_j dx + \int_{\Omega} \left(\sum_{i=1}^n v_i \nabla \omega_i \right) \nabla \omega_j dx + \\ & \int_{\Omega} f \left(x, t, \sum_{i=1}^n v_i \nabla \omega_i \right) \omega_j dx \end{aligned} \quad (13)$$

com $j = 1, \dots, n$. Para o Problema (1) temos o seguinte problema de Cauchy

$$\begin{cases} \eta''(t) + P_n(t, \eta(t), \eta'(t)) = F_n(t) \\ \eta(0) = U_{0n} \\ \eta'(0) = U_{1n} \end{cases} \quad (14)$$

onde

$$(U_{0n})_j = \int_{\Omega} \psi_n \omega_j dx, \quad (15)$$

$$(U_{1n})_j = \int_{\Omega} \phi_n \omega_j dx, \quad (16)$$

$$F_n = \int_{\Omega} g_n \omega_j dx, \quad (17)$$

tais que $\psi_n \in V_n$, $\phi_n \in V_n$, $g_n \in C_0^\infty(Q_T)$, $\psi_n \rightarrow u_0$ em $W^{2,p(x)}(\Omega) \cap W_0^{1,2}(\Omega)$, $\phi_n \rightarrow u_1$ em $L^2(\Omega)$ e $g_n \rightarrow g$ em $L^{q'(x)}(Q_T)$.

A seguir enunciaremos uma proposição e um lema que auxiliam na demonstração do nosso principal teorema sobre a existência de soluções fracas para o Problema (1).

Proposição 1: (ver Fu e Pan (2010)) Seja $p(x) : \bar{\Omega} \rightarrow \mathbb{R}$ uma função log-Hölder e limitada por $p^- > 1$. Se $\{u_n\}_{n=1}^\infty$ é limitada em $L^{p(x)}(Q_T)$ e $u_n \rightarrow u$ quase sempre em Q_T quando $n \rightarrow \infty$, então existe uma subsequência de $\{u_n\}$, ainda denotada por $\{u_n\}$, tal que $u_n \rightharpoonup u$ fracamente em $L^{p(x)}(Q_T)$ quando $n \rightarrow \infty$.

Lema 1: As seguintes estimativas são uniformes em relação a n .

$$\int_{\Omega} \left| \frac{\partial u_n(x, t)}{\partial t} \right|^2 dx + \int_{\Omega} |\Delta u_n(x, t)|^{p(x)} dx + \int_{\Omega} |\nabla u_n(x, t)|^2 dx \leq c_4, \quad (18)$$

$$\int_{Q_T} \left| \frac{\partial u_n}{\partial t} \right|^{q(x)} dx dt + \int_{Q_T} |\Delta u_n|^{p(x)} + \left| \nabla \frac{\partial u_n}{\partial t} \right|^2 dx dt \leq c_5, \quad (19)$$

$$\| |\Delta u_n|^{p(x)-2} \Delta u_n \|_{L^{p'(x)}(Q_T)} + \left\| f \left(x, t, \frac{\partial u_n}{\partial t} \right) \right\|_{L^{q'(x)}(Q_T)} \leq c_6. \quad (20)$$

Teorema 1: Sob as hipóteses **(H.1)**, **(H.2)** e **(H.3)** o Problema (1) tem solução fraca.

Prova:

De fato, utilizando a Proposição (1) e o Lema (1), existe uma subsequência de $\{u_n\}$ (ainda denotada por $\{u_n\}$), u , ξ e τ tal que

$$\left\{ \begin{array}{ll} \frac{\partial u_n}{\partial t} \rightharpoonup \frac{\partial u}{\partial t} & \text{fracamente } * \text{ em } L^\infty(0, T; L^2(\Omega)), \\ u_n \rightharpoonup u & \text{fracamente } * \text{ em } L^\infty(0, T; W_0^{2,p(x)}(\Omega)) \cap L^\infty(0, T; W_0^{1,2}(\Omega)), \\ \frac{\partial u_n}{\partial t} \rightharpoonup \frac{\partial u}{\partial t} & \text{fracamente em } L^{q(x)}(Q_T) \cap L^2(0, T; W_0^{1,2}(\Omega)), \\ |\Delta u_n|^{p(x)-2} \Delta u_n \rightharpoonup \xi & \text{fracamente em } L^{p'(x)}(Q_T), \\ f \left(x, t, \frac{\partial u_n}{\partial t} \right) \rightharpoonup \tau & \text{fracamente em } L^{q'(x)}(Q_T), \end{array} \right.$$

onde

$$\xi = |\Delta u|^{p(x)-2} \Delta u \quad (21)$$

e

$$\tau = f \left(x, t, \frac{\partial u}{\partial t} \right). \quad (22)$$

Com essas convergências podemos passar o limite em n e consequentemente provamos que $u(x, t)$ é solução do Problema (1) no sentido da Definição (1).

3. DECAIMENTO DE ENERGIA PARA AS SOLUÇÕES FRACAS

Existem pouquíssimos resultados sobre o comportamento assintótico da solução fraca ou forte para Equações Diferenciais Parciais hiperbólicas com expoente variável. Para garantir a regularidade das soluções fracas, utilizarmos o método desenvolvido por Nakao (1995) e determinarmos o decaimento de energia das soluções fracas substituiremos as condições de crescimento da função f em **(H.2)** por:

(H.4) $f(x, t, s) \in C(\Omega \times [0, \infty) \times \mathbb{R})$ e existem constantes c_7 e c_8 tais que

$$f(x, t, s) s \geq c_7 |s|^{q(x)} \quad (23)$$

e

$$|f(x, t, s)| \leq c_8 |s|^{q(x)-1} \quad (24)$$

para todo $(x, t, s) \in \Omega \times [0, \infty) \times \mathbb{R}$.

Corolário 1: Sejam as hipóteses **(H.1)**, **(H.3)** e **(H.4)** e suponhamos que $g(x, t) \equiv 0$. Então para cada $u_0 \in W^{2,p(x)}(\Omega) \cap W_0^{1,2}(\Omega)$, existe $u(x, t) : \Omega \times (0, \infty) \rightarrow \mathbb{R}$ tal que $\forall T > 0$ temos

$$u \in L^\infty(0, T; W_0^{2,p(x)}(\Omega)) \cap L^\infty(0, T; W_0^{1,2}(\Omega)), \quad (25)$$

$$\frac{\partial u}{\partial t} \in L^\infty(0, T; L^2(\Omega)) \cap L^2(0, T; W_0^{1,2}(\Omega)) \cap L^{q(x)}(Q_T) \quad (26)$$

e

$$-\int_{\Omega} \frac{\partial u(x, 0)}{\partial t} \varphi(x, 0) dx - \int_{Q_T} \frac{\partial u}{\partial t} \frac{\partial \varphi}{\partial t} dx dt + \int_{Q_T} |\Delta u|^{p(x)-2} \Delta u \Delta \varphi dx dt + \\ \int_{Q_T} \nabla \frac{\partial u}{\partial t} \nabla \varphi dx dt + \int_{Q_T} f \left(x, t, \frac{\partial u}{\partial t} \right) \varphi dx dt = 0, \quad (27)$$

para todo $\varphi \in C^1(0, T; C_0^\infty(\Omega))$ com $\varphi(x, T) = 0$.

Para demonstrarmos o Corolário (1) usamos os mesmos argumentos do Teorema (1).

Lema 2: (ver Nakao (1995)) Seja $\Psi : (0, \infty) \rightarrow \mathbb{R}$ uma função não negativa e limitada. Se existem duas constantes $\alpha > 0$ e $\beta \geq 0$ tais que

$$\sup_{t \leq s \leq t+1} \Psi^{1+\beta}(s) \leq \alpha [\Psi(t) - \Psi(t+1)], \quad \forall t \geq 0. \quad (28)$$

Então existem constantes positivas C e γ tais que

$$\begin{cases} \Psi(t) \leq Ce^{-\gamma t}, & \forall t \geq 0, \text{ quando } \beta = 0, \\ \Psi(t) \leq C(t+1)^{-\frac{1}{\beta}}, & \forall t \geq 0, \text{ quando } \beta > 0. \end{cases} \quad (29)$$

A seguir obtivemos o teorema principal sobre o comportamento assintótico da solução do Problema (1) e a sua demonstração é feita usando o Lema (2) desenvolvido por Nakao (1995).

Teorema 2: Seja $p^- > \max \left\{ 1, \frac{2N}{N+2} \right\}$. Então existem constantes $C > 0$ e $\gamma > 0$ tais que as soluções fracas obtidas pelo Corolário 1 satisfazem:

Se $q^- \geq 2$, então

$$\int_{\Omega} \left| \frac{\partial u(x, t)}{\partial t} \right|^2 dx + \int_{\Omega} |\Delta u(x, t)|^{p(x)} dx \leq \begin{cases} Ce^{-\gamma t}, & \forall t \geq 0, \text{ se } p^+ = 2, \\ C(t+1)^{-\frac{p^+}{p^+-2}}, & \forall t \geq 0, \text{ se } p^+ > 2; \end{cases} \quad (30)$$

Se $1 < q^- < 2$, então

$$\int_{\Omega} \left| \frac{\partial u(x, t)}{\partial t} \right|^2 dx + \int_{\Omega} |\Delta u(x, t)|^{p(x)} dx \leq \begin{cases} C(t+1)^{-\frac{p^+(q^- - 1)}{p^+ - q^-}}, & \forall t \geq 0, \text{ se } p^+ < q^-, \\ Ce^{-\gamma t}, & \forall t \geq 0, \text{ se } p^+ \geq q^-. \end{cases} \quad (31)$$

Quando a função $g \neq 0$ utiliza-se hipóteses apropriadas de crescimento, para mais detalhes veja Ferreira (1991).

4. CONCLUSÕES E TRABALHOS FUTUROS

Nesse trabalho estudamos a existência de soluções fracas para a equação de onda de quarta ordem com expoentes variáveis não lineares em um domínio limitado. Também investigamos o comportamento assintótico das soluções fracas e obtivemos o decaimento exponencial e polinomial.

Em trabalhos futuros demonstraremos que o Problema (1) possui uma única solução fraca.

Devido ao fato de não existir resultados na literatura referentes a análise numérica para problemas hiperbólicos com expoente variável estamos conjecturando utilizar o Método de Elementos Finitos, Método das Diferenças Finitas ou o Método de Ruth para analisarmos numericamente o referido problema. Para as simulações computacionais usaremos o software MATLAB supondo $p(x) = 2$, $f \equiv 0$ e $g \equiv 0$. Notemos que, se $p(x) = 2$ então o termo envolvendo expoente variável no Problema (1) será

$$\Delta(|\Delta u|^{p(x)-2}\Delta u) = \Delta(|\Delta u|^{2-2}\Delta u) = \Delta(\Delta u) = \Delta^2 u.$$

Essas suposições simplificarão a implementação computacional, pois nossa equação assumirá a seguinte forma

$$\frac{\partial^2 u}{\partial t^2} + \Delta^2 u - \Delta \frac{\partial u}{\partial t} = 0.$$

Além disso, nosso objetivo é obter as mesmas propriedades do Problema (1) em domínio não limitado e com fronteira móvel. Também serão investigadas as desigualdades variacionais relacionadas ao referido problema e suas ramificações, tentaremos também obter um resultado substancial para a equação de onda de quarta ordem envolvendo expoente variável não linear sem o termo dissipativo forte $-\Delta \frac{\partial u}{\partial t}$. Dentre outros problemas que surgirão durante nossas pesquisas.

REFERÊNCIAS

- [1] Andrade, D.; Silva, M. A. J.; Fu, M. T. 2012. Exponential stability for a plate equation with p -Laplacian and memory terms. *Mathematical Methods in the Applied Sciences*, v. 35, p. 417-426.
- [2] Alaoui, M. K.; Messaoudi, S. A.; Khenous, H. B. 2014. A blow-up result for nonlinear generalized heat equation. *Computers and Mathematics with Applications*, v. 68, p. 1723-1732.
- [3] Antontsev, S. 2011. Wave equation with $\vec{p}(x, t)$ -Laplacian and damping term: existence and blow-up. *Journal Differential Equations & Applications*, v. 3, p. 503-525.
- [4] Antontsev, S.; Ferreira, J. 2013. Existence, uniqueness and blow up for hyperbolic equations with nonstandard growth conditions. *Nonlinear Analysis: Theory, Methods and Applications*, v. 93, p. 62-77.
- [5] Antontsev, S.; Ferreira, J. 2014. On a viscoelastic plate equation with strong damping and $\vec{p}(x, t)$ -Laplacian existence and uniqueness. *Journal Differential and Integral Equations*, v. 27, p. 1147-1170.
- [6] Antontsev, S.; Chipot, M.; Xie, Y. 2007. Uniqueness results for equations of the $p(x)$ -Laplacian type. *Advances in Mathematical Sciences and Applications*, v. 17.
- [7] Chabrowski, J.; Fu, Y. 2005. Existence of solutions for $p(x)$ -Laplacian problems on a bounded domain. *Journal of Mathematical Analysis and Applications*, v. 306, p. 604-618.
- [8] Chipot, M.; Savitska, T. 2014. Nonlocal p -Laplace equations depending on the L^p norm of the gradient. *Journal Advances in Differential Equations*, v. 19, p. 997-1020.
- [9] Fan, X. L.; Zhao, D. 2001. On the spaces $L^{p(x)}(\Omega)$ and $W^{m,p(x)}(\Omega)$. *Journal of Mathematical Analysis and Applications*, v. 263, p. 424-446.
- [10] Ferreira, J. 1991. Sobre uma equação hiperbólica-parabólica semilinear. *Tese de Doutorado - Universidade Federal do Rio de Janeiro*.

- [11] Ferreira, J.; Messaoudi, S. A. 2014. On the general decay of a nonlinear viscoelastic plate equation with a strong damping and $\vec{p}(x, t)$ -Laplacian. *Nonlinear Analysis: Theory, Methods and Applications*, v. 104, p. 40-49.
- [12] Fu, Y.; Pan, N. 2010. Existence of solutions for nonlinear parabolic problems with $p(x)$ -growth. *Journal of Mathematical Analysis and Applications*, v. 362, p. 313-326.
- [13] Gao, H.; Fu, M. T. 1999. Global solutions for a nonlinear wave equation with the p -Laplacian operator. *Electronic Journal on the Qualitative Theory of Differential Equations*, v. 11, p. 1-13.
- [14] Nakao, M. 1995. Energy decay for the quasilinear wave equation with viscosity. *Mathematische Zeitschrift*, v. 219, p. 289-299.
- [15] Pinasco, J. P. 2009. Blow-up for parabolic and hyperbolic problems with variable exponents. *Nonlinear Analysis*, v. 71, p. 1094-1099.
- [16] Zhang, C.; Zhou, S. 2009. A fourth-order degenerate parabolic equation with variable exponent. *Journal of Partial Differential Equations*, v. 22, p. 376-392.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

APLICAÇÃO DE OBSERVADORES DE ESTADO EM UM MODELO SIR PARA ESTIMAÇÃO DA POPULAÇÃO DE MOSQUITOS TRANSMISSORES DA FEBRE AMARELA

Fernanda Paula Rocha¹ - fernanda.rocha507@gmail.com

Mateus Giesbrecht¹ - mateus@fee.unicamp.br

João Frederico da Costa Azevedo Meyer² - joni@ime.unicamp.br

¹Faculdade de Engenharia Elétrica e de Computação (FEEC), UNICAMP

²Instituto de Matemática, Estatística e Computação Científica (IMECC), UNICAMP

Resumo. A febre amarela é uma doença infecciosa preocupante da atualidade devido à crescente quantidade de casos nos últimos anos e à alta taxa de mortalidade. Com o recente surto no Brasil e em outros países, e com a eficiência dos métodos de controle tradicionais, o impacto ambiental e a resistência em progresso a pesticidas, destaca-se a necessidade de desenvolver ferramentas para o controle dos mosquitos transmissores. Pretende-se, neste artigo, estimar a população dos mosquitos transmissores da febre amarela no país através de observadores de estado aplicados a um modelo epidêmico SIR. No modelo são utilizados dados estatísticos informados por órgãos oficiais do governo e taxas de natalidade, mortalidade, infecção e recuperação. A partir desses dados, consegue-se estimar a população de mosquitos. Considerando a dificuldade na medição dessa população, ferramentas que realizem sua estimativa como a proposta neste artigo, são fundamentais para entender a dinâmica dos vetores. Com isso, consegue-se planejar estratégias e definir políticas públicas além de atividades de controle eficazes.

Keywords: modelo SIR, modelo epidemiológico, febre amarela, dinâmica populacional de mosquitos, estimativas de estado

1. INTRODUÇÃO

A Epidemiologia consiste no estudo dos fatores relacionados à saúde de uma determinada população, visando prevenir e controlar problemas de saúde. Desta forma, esta ciência tem sido fundamental ao longo dos anos, uma vez que permite a compreensão do comportamento de uma doença em relação a uma população específica.

Para estudar analítica e numericamente determinada população exposta a uma doença, é necessário compreender previamente o impacto desta última sobre os indivíduos acometidos. Partindo de informações empíricas, observações biológicas, acerca da ação de risco, é então possível decidir em quantos e quais comportamentos é ideal subdividir a população a fim de obter uma caracterização rica o suficiente para que o modelo possa fornecer bons resultados, que indiquem o comportamento futuro da doença, e simples o suficiente para que o problema seja matematicamente tratável, criando simulações úteis para a Saúde Pública.

Neste trabalho versa-se sobre a análise do comportamento da febre amarela nos últimos 38 anos, com o monitoramento realizado pelo Ministério da Saúde (MS) e pelo Instituto Brasileiro de Geografia e Estatística (IBGE), conforme pode ser visto na Fig. 1, em que cada unidade da população de suscetíveis equivale a um milhão de indivíduos e cada unidade das populações de infectados e recuperados equivale a apenas um indivíduo. Através desta análise, elaborou-se um modelo no qual foram aplicados os observadores de estados, realizando as estimações das populações de mosquitos portadores e não portadores da doença em questão, o qual é o único e principal foco do trabalho.

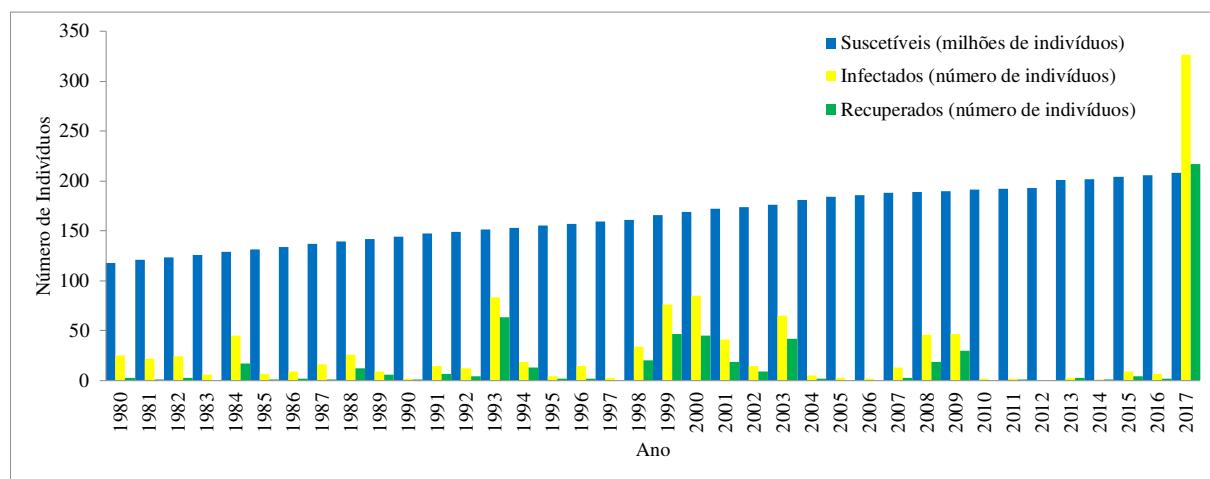


Figura 1- Dados obtidos do Ministério da Saúde e do Instituto Brasileiro de Geografia e Estatística, onde cada unidade da população de suscetíveis equivale a um milhão.

1.1 Características da doença

A febre amarela é um doença infecciosa aguda, transmitida pela picada dos mosquitos transmissores infectados. Esta doença pode apresentar dois ciclos epidemiológicos, sendo eles o ciclo silvestre e ciclo urbano. Na forma silvestre os transmissores são primariamente mosquitos do gênero *Haemagogus* e *Sabathes*, enquanto na forma urbana o vetor é o mesmo da dengue, o *Aedes aegypti*, mas casos de transmissão urbana, no entanto, não são registrados no país desde 1942.

Para realizar a construção do modelo matemático neste trabalho, precisou-se considerar alguns traços sobre a febre amarela. A mesma tem como características que a prole dos mosquitos infectados nasce infectada, e que o ser humano que consegue se recuperar da doença se torna imune. Os mosquitos transmissores da doença infectam os macacos, mas estes não foram considerados neste modelo, apesar desse animal ser um dos principais hospedeiros do vírus no ciclo de transmissão silvestre. A morte desses animais próximos aos locais habitados por humanos, mostra que a doença não está limitada ao ambiente das matas, estando mais próxima das áreas urbanas.

2. MODELO MATEMÁTICO PARA DESCRIÇÃO DE SISTEMAS DINÂMICOS

Com o intuito de prever ou estimar o crescimento de uma dada população, é necessário um modelo dinâmico, tais modelos também podem ser úteis caso pretenda-se estimar a densidade de mosquitos aedes aegypti em uma comunidade afetada pela dengue como realizado em Massad et al. (2017). A maneira como as estimações das populações de mosquitos portadores e não portadores da doença da febre amarela é realizada neste trabalho se torna inovadora, devido as estimativas serem realizadas pelo Filtro de Kalman Estendido (FKE), este sendo apresentado por Smith et al. (1962), que tinha a finalidade de estimar as trajetórias do veículo lunar da Apollo 11.

É preciso entender que um modelo matemático é, na melhor das hipóteses, um conjunto de fórmulas e/ou equações independentes, baseado em uma descrição quantitativa aproximada dos fenômenos reais e criado na esperança de que o comportamento previsto seja consistente com o comportamento real em que se baseia. Sua precisão está sujeita às suposições e exigências feitas pelo pesquisador. Neste caso, usa-se um modelo compartmental SIR, o qual será melhor apresentado na próxima seção, com interpretações próximas as características gerais da doença, a fim de obter uma melhor modelagem do problema.

Uma boa ferramenta de modelagem de sistema dinâmicos multivariáveis é a abordagem em espaço de estados. Esse método é utilizado neste trabalho conforme exposto na seção 2.2.

2.1 Descrição do modelo matemático

O modelo escolhido para descrever a dinâmica de transmissão da febre amarela em uma população, é o modelo clássico do tipo *SIR* tendo como referência Leah (2005) e Murray (2002).

A população de humanos é dividida em três compartimentos: suscetíveis S , infectados I e recuperados R . O indivíduo recupera-se, adquirindo imunidade e permanecendo nesse compartimento. A população de mosquitos é dividida em dois compartimentos, indicados por N como a população de mosquitos não portadores e P a população de mosquitos portadores. Quando um indivíduo suscetível é picado por um mosquito portador ele passa para o compartimento dos infectados com taxa β_H e quando o mosquito não portador pica um indivíduo infectado passa para o compartimento dos portadores com a taxa β_I . O indivíduo que contrai a doença pode se recuperar após o tratamento com taxa δ_H ou então morrer pela doença com taxa α_H , enquanto que o mosquito portador permanece assim pelo resto da sua vida e morre com taxa μ_N que é a mortalidade natural, ou morre com taxa μ_I que é a mortalidade induzida antropicamente, isto é, através da ação humana. A taxa de natalidade dos humanos é λ_H , com μ_H sendo a taxa dos indivíduos que morrem naturalmente e λ_I a taxa de natalidade dos mosquitos. Todos os parâmetros anteriores estão melhor esboçados na Tabela 1 para melhor visualização e compreensão do leitor.

Para as populações de humanos e de mosquitos considerou-se que competem por uma quantidade limitada de recursos e, além disso, considera-se que os nascidos de humanos infectados e recuperados nascem suscetíveis.

O comportamento das populações é expresso matematicamente pelo sistema não linear de cinco equações diferenciais ordinárias, descrito em (1), considerando uma dinâmica populacional malthusiana.

Tabela 1- Siglas utilizadas no modelo proposto.

Parâmetro	Significado dos parâmetros
$S(t)$	População de suscetíveis
$I(t)$	População de infectados
$R(t)$	População de recuperados
$N(t)$	População de mosquitos não portadores da doença
$P(t)$	População de mosquitos portadores da doença
λ_H	Taxa de natalidade da população humana
β_H	Taxa de suscetíveis que contraem a doença ao serem picados por mosquitos infectados
μ_H	Taxa na qual os humanos morrem naturalmente
α_H	Taxa com a qual indivíduos infectados morrem pela doença
δ_H	Taxa da população de indivíduos infectados que se recuperam da doença
λ_I	Taxa de natalidade da população de mosquitos
β_I	Taxa dos mosquitos não portadores que contraem o vírus ao picarem indivíduos infectados
μ_I	Taxa de mortalidade da população de mosquitos pela ação humana
μ_N	Taxa de mortalidade natural da população de mosquitos

$$\left\{ \begin{array}{l} \frac{dS(t)}{dt} = -\beta_H S(t)P(t) - \mu_H S(t) + \lambda_H(S(t) + I(t) + R(t)) \\ \frac{dI(t)}{dt} = \beta_H S(t)P(t) - \delta_H I(t) - (\mu_H + \alpha_H)I(t) \\ \frac{dR(t)}{dt} = \delta_H I(t) - \mu_H R(t) \\ \frac{dN(t)}{dt} = -\beta_I N(t)I(t) + \lambda_I N(t) - \mu_I N(t) - \mu_N N(t) \\ \frac{dP(t)}{dt} = \beta_I N(t)I(t) + \lambda_I P(t) - \mu_I P(t) - \mu_N P(t) \end{array} \right. \quad (1)$$

Sendo $T(t) = S(t) + I(t) + R(t)$, o número total de indivíduos da população de humanos com condições iniciais $S(0) \geq 0$, $I(0) \geq 0$, $R(0) \geq 0$, $N(0) \geq 0$ e $P(0) \geq 0$. A variação no número de indivíduos da população em relação ao tempo é dada pela relação apresentada na equação (2).

$$\frac{dT(t)}{dt} = \frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt}, \quad (2)$$

com $t \in J = [0, T_f]$ o domínio temporal em que T_f é o tempo final.

A partir daqui, é introduzido um sistema dinâmico em que o estado evolui em etapas de tempo discreto, isto é, o modelo de tempo contínuo será modificado para um modelo de tempo discreto, através da aproximação de diferenças finitas progressivas.

A ideia é que existem ações relativamente curtas e sincronizadas (por exemplo, estações de reprodução) que permitem ignorar o comportamento dentro do período de tempo para o propósito do modelo. Uma visão alternativa de modelos discretos é que eles são discretizações de modelos de tempo contínuo. A discretização do sistema apresentado na equação (1) na

equação (3).

$$\begin{cases} S(k+1) = (-\beta_H h P(k) - \mu_H h + \lambda_H h + 1)S(k) + \lambda_H h(I(k) + R(k)), \\ I(k+1) = (1 - \delta_H h - \mu_H h - \alpha_H h)I(k) + \beta_H h S(k)P(k), \\ R(k+1) = \delta_H h I(k) + (1 - \mu_H h)R(k), \\ N(k+1) = (-\beta_I h I(k) + \lambda_I h - \mu_I h - \mu_N h + 1)N(k) \text{ e} \\ P(k+1) = \beta_I h N(k)I(k) + (\lambda_I h - \mu_I h - \mu_N h + 1)P(k). \end{cases} \quad (3)$$

O intervalo de tempo é expresso através do parâmetro h na equação (3).

Explora-se o uso desse sistema na modelagem de fenômenos biológicos como a dinâmica populacional e a epidemiologia, mostrando o efeito no comportamento dos modelos decorrente da variação das constantes associadas a cada equação.

2.2 Modelo discreto no espaço de estados

Uma representação que pode ser usada para modelar relações entre variáveis internas ao sistema é a representação em espaço de estados. Esse tipo de representação descreve o sistema no domínio de tempo e é mais conveniente para representar sistemas multivariáveis, o que é o caso neste trabalho.

Um modelo de sistema dinâmico discreto não linear sem entradas pode ser representado matematicamente no espaço de estados pelas relações abaixo:

$$x(k+1) = f(x(k)), \quad (4)$$

$$y(k) = h(x(k)) \quad (5)$$

e com $k \in \mathbb{Z}_+$. Sendo $f(x(k))$ a função de transição de estado, $h(x(k))$ a função que relaciona o estado e a saída em um mesmo instante de tempo, $y(k)$ é o vetor de saída e $x(k)$ é o vetor de estado. Neste caso, tem-se que as matrizes do espaço de estados do sistema (3), têm a forma

$$x(k) = (S(k) \ I(k) \ R(k) \ N(k) \ P(k))^T, \quad (6)$$

$$f(x(k)) = \begin{pmatrix} -\beta_H h P(k)S(k) - (\mu_H - \lambda_H)hS(k) + S(k) + \lambda_H h(R(k) + I(k)) \\ \beta_H h P(k)S(k) + I(k) - (\mu_H + \alpha_H + \delta_H)hI(k) \\ \delta_H h I(k) + R(k) - \mu_H h R(k) \\ N(k) - \beta_I h I(k)N(k) + (\lambda_I - \mu_I - \mu_{(k)N})hN(k) \\ P(k) + \beta_I h I(k)N(k) + (\lambda_I - \mu_I - \mu_N)hP(k) \end{pmatrix} \text{ e } (7)$$

$$h(x(k)) = \begin{pmatrix} S(k) \\ I(k) \\ R(k) \end{pmatrix}. \quad (8)$$

3. ESTIMANDO AS POPULAÇÕES DE MOSQUITOS

Para realizar este processo, faz-se uso do Filtro de Kalman Estendido (FKE) proposto por Smith et al. (1962), amplamente utilizado para estimativas em modelos não lineares. O procedimento consiste em linearizar analiticamente o sistema em torno do estado atual e aplicar as

equações do Filtro de Kalman, para isso fazendo-se necessário o cálculo da matriz Jacobiana. Detalhadamente, temos

$$J(k) = \left. \frac{\partial f(x(k))}{\partial x^T(k)} \right|_{x(k)=\hat{x}(k|k)} \quad \text{e} \quad H(k) = \left. \frac{\partial h(x(k))}{\partial x^T(k)} \right|_{x(k)=\hat{x}(k+1|k)}.$$

Sendo que para o modelo SIR apresentado nas equações (4) e (5) as jacobianas da função de transição de estado $f(x(k))$ e da função de saída $h(x(k))$ são representadas abaixo como,

$$J(k) = \begin{pmatrix} \lambda_H h - \mu_H h - \beta_H h P(k) + 1 & \lambda_H h & \lambda_H h & 0 & -\beta_H h S(k) \\ \beta_H h P(k) & 1 - (\delta_H h + \mu_H h + \alpha_H h) & 0 & 0 & \beta_H h S(k) \\ 0 & \delta_H h & (1 - \mu_H h) & 0 & 0 \\ 0 & -\beta_I h N(k) & 0 & -\beta_I h I(k) + \lambda_I h - \mu_I h - \mu_N h + 1 & 0 \\ 0 & \beta_I h N(k) & 0 & \beta_I h I(k) & \lambda_I h - \mu_I h - \mu_N h + 1 \end{pmatrix} \quad \text{e}$$

$$H(k) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \quad (9)$$

Sendo as matrizes de ruído do processo e de medição respectivamente,

$$Q = \begin{pmatrix} 15 & 0 & 0 & 0 & 0 \\ 0 & 33 & 0 & 0 & 0 \\ 0 & 0 & 45 & 0 & 0 \\ 0 & 0 & 0 & 35 & 0 \\ 0 & 0 & 0 & 0 & 20 \end{pmatrix} \quad \text{e} \quad R = \begin{pmatrix} 0.9 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.6 \end{pmatrix}.$$

O algoritmo do FKE do processo a ser realizado para obter as estimativas das populações, segue abaixo.

Etapa de predição

$$\hat{x}(k+1|k) = f(\hat{x}(k|k), u(k)), \quad (10)$$

$$J(k) = \left. \frac{\partial f(x(k))}{\partial x^T(k)} \right|_{x(k)=\hat{x}(k|k)}, \quad (11)$$

$$\hat{P}(k+1|k) = J(k) \hat{P}(k|k) J(k)^T + Q, \quad (12)$$

$$H(k) = \left. \frac{\partial h(x(k))}{\partial x^T(k)} \right|_{x(k)=\hat{x}(k+1|k)}, \quad (13)$$

Etapa de correção/atualização

$$K(k) = \hat{P}(k+1|k) H(k)^T (H(k) \hat{P}(k+1|k) H(k)^T + R)^{-1}, \quad (14)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)(\hat{y}(k+1) - h(\hat{x}(k+1|k))) \quad \text{e} \quad (15)$$

$$\hat{P}(k+1|k+1) = (I - K(k+1) H(k)) \hat{P}(k+1|k). \quad (16)$$

4. RESULTADO DA SIMULAÇÃO

A partir do modelo discretizado em espaço de estados, tem-se como objetivo realizar as estimativas das populações usando técnicas recursivas de estimação de estado, através de dados

reais da quantidade de indivíduos de cada população, como consta na Fig. 1, ao longo do tempo, com os seguintes parâmetros.

Na Tabela 2, são designados os valores iniciais das populações de mosquitos não portadores da doença $N(0)$ e os portadores $P(0)$, bem como os valores dos parâmetros do modelo, todos estes sendo supostos por um período de um ano. Os únicos dados disponíveis ao longo dos últimos 38 anos são as populações de suscetíveis, infectados e recuperados, que podem ser visualizadas na Fig. 1. A partir desses dados, que foram supostos por um período de um ano, temos os seguintes resultados.

Tabela 2- Descrição dos parâmetros do modelo.

Parâmetro	Valor	Referência
$S(0)$	118.562.500,00	IBGE
$I(0)$	25×10^6	MS
$R(0)$	3×10^6	MS
$N(0)$	44×10^7	Suposto
$P(0)$	22×10^7	Suposto
λ_H	6×10^{-3}	Suposto
β_H	9×10^{-4}	Suposto
μ_H	6×10^{-3}	Suposto
α_H	33×10^{-2}	Suposto
δ_H	67×10^{-2}	Suposto
λ_I	37×10^{-2}	Suposto
β_I	1.5×10^{-7}	Suposto
μ_I	35×10^{-2}	Suposto
μ_N	4×10^{-2}	Suposto
h	1×10^{-9}	Suposto

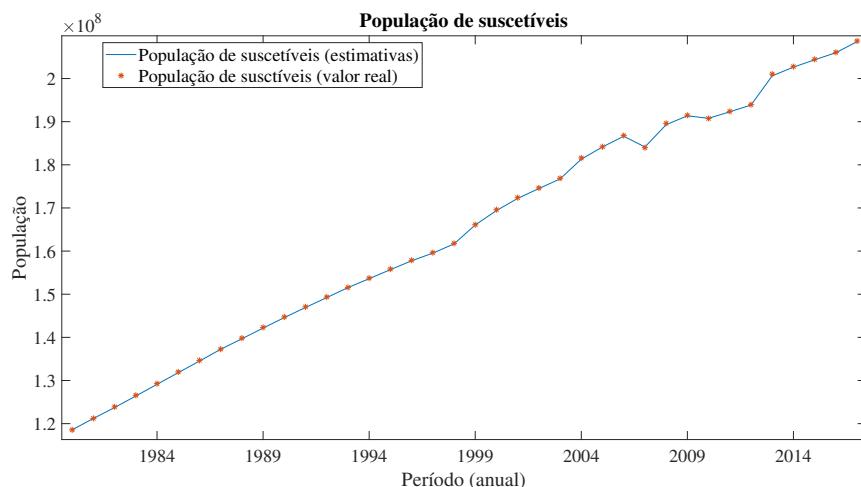


Figura 2- Estimativas das populações de humanos suscetíveis a doença no período de 1980 a 2017.

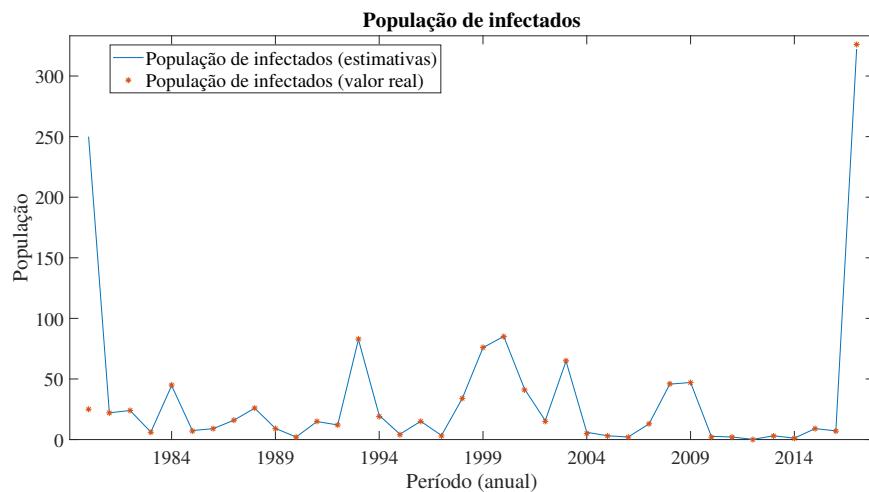


Figura 3- Estimativas das populações de humanos que foram infectadas pela doença no período de 1980 a 2017.

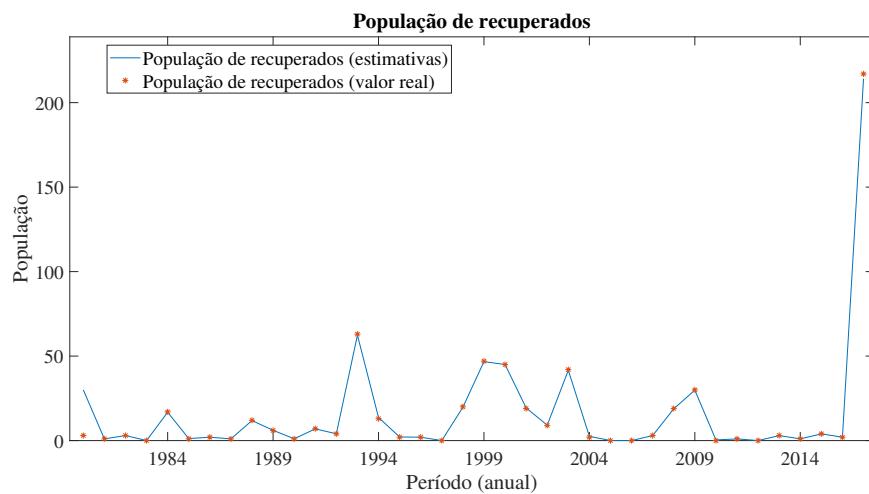


Figura 4- Estimativas das populações de humanos que se recuperaram da doença no período de 1980 a 2017.

Nas Fig. 2, Fig. 3 e Fig. 4, são mostradas as estimativas anuais das populações de suscetíveis, infectados e recuperados respectivamente no período de 1980 a 2017, obtidas com o algoritmo proposto e os valores medidas apresentados na Fig. 1. Os resultados mostram boas estimativas em relação ao dados reais. Também foram realizadas as estimativas das populações de mosquitos, como pode ser observado nas Fig. 5 e 6.

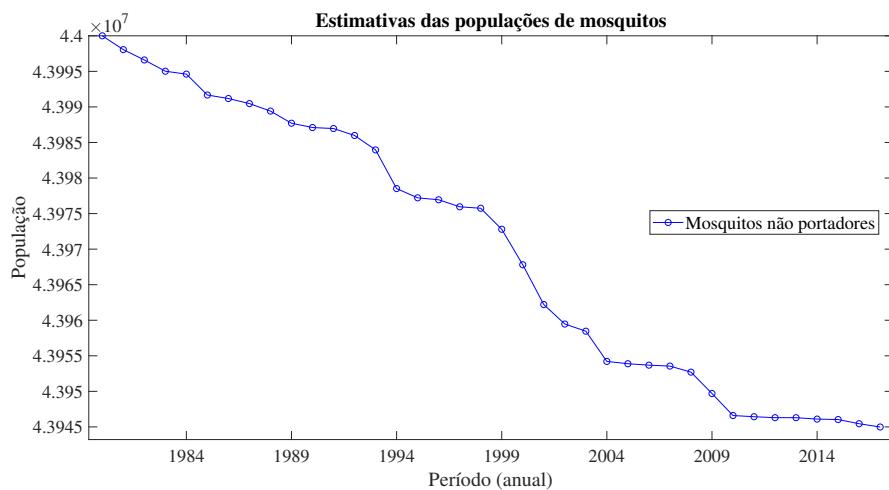


Figura 5- Estimativas das populações de mosquitos não portadores da doença no período de 1980 a 2017.

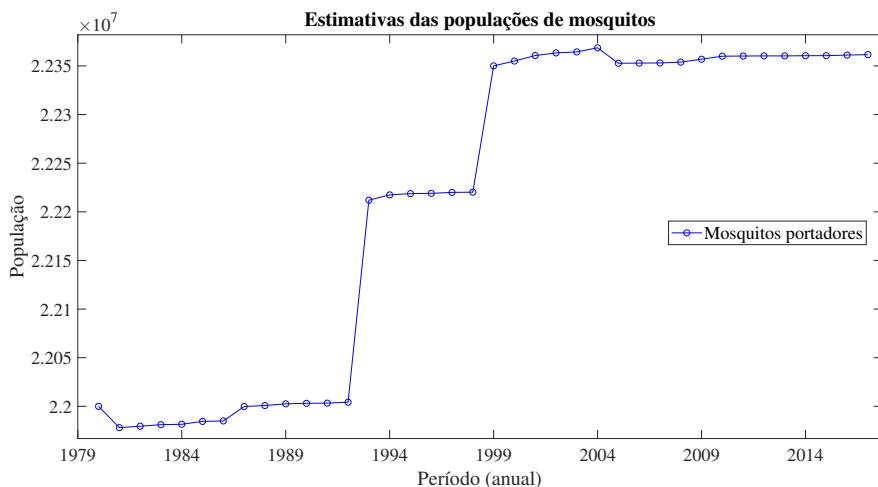


Figura 6- Estimativas das populações de mosquitos portadores da doença no período de 1980 a 2017.

No ano de 1986 foram 9 casos confirmados, porém houve um leve crescimento na quantidade de mosquitos portadores neste período. Talvez por isso, nos anos de 1987 e 1988 ocorreu novamente um aumento no número de casos. No ano de 1993, aconteceu novamente o crescimento de casos, com uma diferença grande entre os anos de 1992 com 12 casos para 1993 com 83 casos, sendo nesse período um dos maiores saltos na quantidade de mosquitos contaminados em relação a anos anteriores, o que é aceitável, pois as contaminações foram relativamente baixas em anos anteriores. Em 1998, foram 34 casos confirmados, e no mesmo período aconteceu um aumento na quantidade de mosquitos portadores, tendo continuidade nos anos seguintes até o ano de 2004, onde obteve-se somente 5 casos, do contrário do ano anterior que foram 65, o que pode ser notado também com a diminuição na quantidade de mosquitos infectados na mesma data. Por fim, entre os anos de 2007 e 2009, nota-se maiores quantidades de infectados, chegando aos seus 47 no último ano citado, em relação ao ano de 2006 que foram somente 2, e durante esse período a população de mosquitos não sofreu grandes variações.

Na Fig. 4 no gráfico das estimativas das pessoas que se recuperaram da doença é proporcional ao de infectados, isto é, quando mais infectados durante determinado período, maior foram a quantidade de recuperados. A estimativa dos mosquitos não portadores equivale as taxas impostas na Tabela 2, sendo que a taxa de mortalidade era maior que a de natalidade, e a população dos mosquitos portadores está condizente com os períodos de infecção da doença.

5. CONCLUSÃO

Os resultados obtidos foram satisfatórios do ponto de vista biológico, sendo empregado uma técnica diferente para realizar as estimativas das populações de mosquitos, esta provando ser capaz de executar estimativas coerentes. Pois em nenhum momento ocorreu uma diminuição na população de mosquitos portadores quando ocorreu um aumento no número de infectados e o contrário também não ocorreu nos períodos de diminuição dos casos confirmados, havendo no máximo uma estabilização na quantidade de mosquitos ou uma leve alteração na mesma durante estes períodos, mas sempre sendo coerente com o comportamento da população de pessoas infectadas. O filtro estimou um aumento significativo no número de infectados e mosquitos portadores nos anos de 1993 e 1999, mesmo que não tenham mantido a mesma proporção (por porcentagem).

O comportamento dinâmico estimado para os compartimentos da população de mosquitos foi conforme esperado, comprovando que os observadores de estado podem ser usados para estimar esse tipo de variável. A estimativa permite que se criem políticas públicas efetivas de controle.

6. AGRADECIMENTOS

Este trabalho recebeu o apoio financeiro parcial da CAPES - Brasil.

Referências

- [1] Instituto Brasileiro de Geografia e Estatística, <https://www.ibge.gov.br/>. Último acesso em 06-03-2018.
- [2] Leah, E. 2005. Mathematical models in biology. In: *Society for Industrial and Applied Mathematics(SIAM)*, vol. 46.
- [3] Ministério da Saúde, <http://portalmgs.saude.gov.br>. Último acesso em 06-03-2018.
- [4] Massad, E.; Amaku, M.; Coutinho, F. A. B.; Struchiner, C. J.; Lopez, L. F.; Wilder-Smith, A.; Burattini, M. N. 2017. Estimating the size of Aedes aegypti populations from dengue incidence data: Implications for the risk of yellow fever outbreaks. In: *Infectious Disease Modelling*.
- [5] Murray, J. D. 2002. Mathematical Biology I: An Introduction. In: *Interdisciplinary Applied Mathematics*, New York: Springer, vol. 17.
- [6] Smith, G. L.; Schmidt, S. F.; MCGEE, L. A. 1962. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. In: *National Aeronautics and Space Administration*.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

IDENTIFICAÇÃO DE DANOS EM UMA VIGA DE EULER-BERNOULLI UTILIZANDO REDES NEURAIS ARTIFICIAIS: UMA ANÁLISE EM RELAÇÃO À POSIÇÃO DO DANO NA ESTRUTURA NA PRESENÇA DE RUÍDOS

Andressa Alves Machado da Silva¹ - andressa.alves@id.uff.br

Diogo Teixeira dos Santos¹ - diogo_santos@id.uff.br

Rosilene Abreu Portella Corrêa¹ - rosileneportella@id.uff.br

Cleber de Almeida Corrêa Junior¹ - cleberacj@id.uff.br

¹Universidade Federal Fluminense - Instituto do Noroeste Fluminense de Educação Superior

Resumo. O presente trabalho apresenta os resultados obtidos através de uma Rede Neural Artificial (RNA) na localização de danos em uma viga simplesmente apoiada de Euler-Bernoulli. Corrêa (2013) propõe a resolução do problema de identificação de danos, transformando-o em um problema de otimização onde o objetivo é minimizar um funcional definido como a diferença entre a matriz de flexibilidade obtida experimentalmente e a correspondente matriz obtida por um modelo de elementos finitos da estrutura. O autor sugere para estruturas mais complexas a utilização de métodos de localização de danos, visando diminuir o número de variáveis do problema. Com isso, o processo de identificação de danos é dividido em duas etapas: localização e quantificação do dano. Método de inteligência computacional, RNA, apresenta como características mais relevantes à adaptação por experiência, a capacidade de aprendizado e a habilidade de generalização. Assim, acredita-se que a substituição dos métodos de localização de danos utilizados em Corrêa (2013) por Redes Neurais Artificiais (RNA) pode gerar melhores resultados. Portanto, o objetivo deste trabalho é avaliar se a RNA é capaz de localizar danos em regiões da viga apontadas por Corrêa (2013) como sendo mais (ou menos) sensíveis à presença de dano na presença de dados ruidosos.

Palavras-chave: Identificação de danos, Matriz de flexibilidade, Redes Neurais Artificiais

1. INTRODUÇÃO

A identificação de danos tem se tornado tema de relevantes pesquisas, pelo fato da sua importância em diversos ramos da Engenharia. Métodos de otimização estocásticos têm sido bastante utilizados na identificação de danos devido à necessidade de lidar com características como, por exemplo, não linearidade, presença de ruído e medição de um número limitado de Graus de Liberdade (GDL). No entanto, estruturas mais complexas, onde um grande número de variáveis precisam ser atualizadas, podem gerar um grande custo computacional ao se utilizar tais métodos, o que dificulta a sua utilização para um monitoramento contínuo estrutural.

Corrêa (2013) testou diversos métodos disponíveis na literatura para localizar danos estruturais, visando com isso diminuir o número de parâmetros a serem atualizados pelos métodos de otimização. Com isso, pretendia-se fazer a identificação de danos em duas etapas: localização da região afetada pelo dano estrutural e quantificação do dano. Apenas na segunda etapa seriam

utilizados métodos de otimização, para atualizar os parâmetros da região indicada na primeira etapa. Dessa forma pretendia-se diminuir o custo computacional dos métodos de otimização. No entanto, dos sete métodos de localização de danos avaliados, nenhum deles apresentou bons resultados quando um número limitado de Graus de Liberdade (GDL) foram utilizados ou quando considerou-se a presença de ruído de medição. Sendo assim, optou-se por testar o desempenho das Redes Neurais Artificiais (RNAs) para substituir a primeira etapa da identificação de danos citada acima.

Redes Neurais Artificiais têm uma grande vantagem em relação aos métodos de otimização: após seu treinamento ser concluído, dada uma entrada com os dados das matrizes de flexibilidade da estrutura, os resultados sobre a posição e o dano são imediatos. Com isso, o objetivo deste trabalho é avaliar os resultados obtidos através de uma RNA em uma viga simplesmente apoiada de Euler-Bernoulli. As posições de danos utilizadas para testar a rede foram escolhidas de acordo com o trabalho de Corrêa (2013), onde mostra-se que, dependendo da região da viga onde há o dano, pode haver maior dificuldade para a localização e quantificação correta do dano estrutural na presença de ruídos.

2. MODELAGEM DO PROBLEMA DE IDENTIFICAÇÃO DE DANOS

Utilizando a teoria de autovalores e autovetores para obter as frequências naturais e as formas modais não-amortecidas da estrutura, tem-se

$$(K - \omega_i^2 M)\phi_i = 0, \quad (1)$$

onde M é a matriz de massa e K a matriz de rigidez, ambas com dimensão $n \times n$. Os termos ω_i e ϕ_i referem-se, respectivamente, à i -ésima frequência natural e à i -ésima forma modal da estrutura.

De forma geral,

$$K\Phi = M\Phi\Lambda, \quad (2)$$

onde Φ é a matriz modal da estrutura, de dimensão $n \times n$ e Λ é uma matriz diagonal, $n \times n$, formada pelos valores quadráticos das frequências naturais, $\lambda_{ii} = \omega_i^2$.

Para as formas modais da estrutura normalizadas em relação à matriz de massa, tem-se

$$\Phi^T M \Phi = I; \quad (3)$$

$$\Phi^T K \Phi = \Lambda, \quad (4)$$

onde I é a matriz identidade e T representa a transposição de uma matriz.

Considerando a Equação (4) e o fato de a matriz de flexibilidade de uma estrutura ser a inversa da matriz de rigidez, tem-se

$$G = (\Phi \Lambda^{-1} \Phi^T) = \sum_{i=1}^n \frac{1}{\omega_i^2} \phi_i \phi_i^T. \quad (5)$$

Devido às limitações experimentais, na prática tem-se a seguinte aproximação para a matriz de flexibilidade experimental G_{exp} da estrutura,

$$G_{exp} = \sum_{i=1}^{n_{exp}} \frac{1}{\omega_{i,exp}^2} \phi_{i,exp} \otimes \phi_{i,exp}, \quad (6)$$

com $n_{exp} < n$ o número de modos obtidos do ensaio experimental, $\omega_{i,exp}$ e $\phi_{i,exp}$ são, respectivamente, a i -ésima frequência natural não-amortecida e forma modal obtidas experimentalmente.

Uma boa estimativa para a matriz de flexibilidade pode ser obtida experimentalmente a partir de modos de mais baixa frequência da estrutura. Da Equação (6), deve-se notar que a dimensão da matriz G_{exp} depende apenas do número m de graus de liberdade (GDL) medidos no ensaio de vibrações, que é equivalente ao número de componentes dos modos experimentais $\phi_{i,exp}$. Sendo assim, a matriz de flexibilidade possui dimensão $m \times m$. Portanto, para se definir um problema de identificação de danos estruturais baseado na matriz de flexibilidade, torna-se necessária a determinação de uma matriz de flexibilidade analítica reduzida \bar{G} relacionada apenas aos m GDL medidos no ensaio de vibrações que contenha informações a respeito das propriedades de rigidez da estrutura como um todo. Para tal, a matriz de rigidez original deve ser particionada na forma

$$K = \begin{bmatrix} K_{mm} & K_{mo} \\ K_{mo}^T & K_{oo} \end{bmatrix}, \quad (7)$$

onde os índices m e o referem-se, respectivamente, aos GDL medidos e omitidos.

3. REDES NEURAIS ARTIFICIAIS (RNA)

RNAs são modelos matemáticos inspirados em neurônios biológicos e na estrutura de processamento paralelo do cérebro, que tem a capacidade de armazenar conhecimento experimental para utilizar futuramente. Segundo Haykin (2001, p.27) as RNAs se assemelham ao cérebro em dois aspectos: na capacidade de aprendizagem em seu ambiente e nos pesos sinápticos que são utilizados para armazenar conhecimento.

O neurônio artificial mostrado na Figura 1 é a unidade de processamento fundamental para a RNA, uma vez que a combinação de diversos neurônios gera uma RNA.

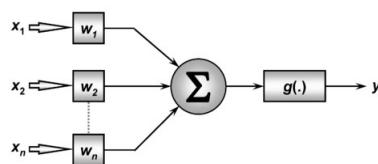


Figura 1- Modelo de neurônio artificial. Fonte: Palmiere, 2016.

Na Figura 1, $x = x_1, x_2, \dots, x_n$ são as entradas da rede, $w = w_1, w_2, \dots, w_n$ são os pesos sinápticos, $\sum = NET = x_1w_1 + x_2w_2 + \dots + x_nw_n$, g é a função de ativação e $y = g(NET)$ é a saída da rede.

A diferença entre uma RNA e outra está na sua arquitetura e no modo como os pesos sinápticos associados às conexões são ajustados durante o treinamento e também na sua topologia (número de neurônios por camadas, funções de ativação, etc.). A RNA Perceptron de Múltiplas Camadas (PMC) utilizada nesse trabalho é dada como na Figura 2. Uma rede pode ter um treinamento supervisionado ou não supervisionado. O treinamento supervisionado é feito informando as entradas e saídas para rede, podendo assim, calcular os erros da RNA durante e após o treinamento.

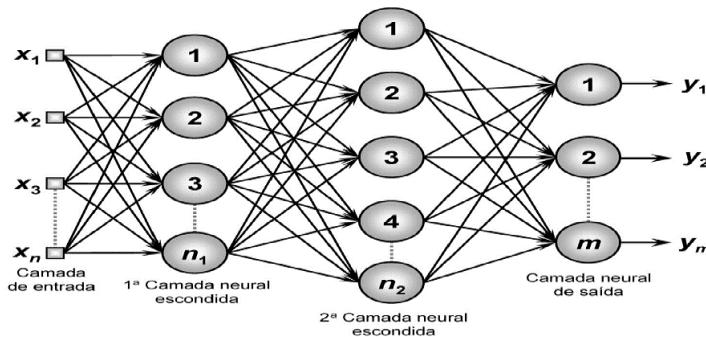


Figura 2- RNA Perceptron de múltiplas camadas. Fonte: Palmiere, 2016.

4. RESULTADOS

No problema de identificação de danos proposto, é considerada uma viga de Euler-Bernoulli simplesmente apoiada. Será considerada uma viga de alumínio simplesmente apoiada com 1,0 m de comprimento, 0,005 m de espessura, 0,05 m de largura, módulo de elasticidade nominal $E_0 = 7,2582 \times 10^{10}$ Pa e momento de inércia de área nominal $I_o = 5,2083 \times 10^{-10} m^4$. A viga em questão foi discretizada pelo Método dos Elementos Finitos em 20 elementos bidimensionais do tipo Euler-Bernoulli. Foram adotados elementos com dois nós, onde cada ponto nodal possui dois GDL de deslocamento, um transversal e outro de rotação, e um parâmetro de coesão. A estrutura possui, portanto, um total de 21 nós, 40 GDL, devido às condições de contorno abordadas, e 21 parâmetros nodais de coesão. No entanto, foram considerados medidos apenas 10 GDL transversais e igualmente espaçados. Admitiram-se os modos de vibração na faixa de frequência de 0 – 450 Hz. A imposição do defeito à viga é realizada através de uma redução na altura relativa da seção transversal $h(x)/h_0$, sendo h_0 a espessura original da viga e $h(x)$ a espessura na posição danificada. Portanto, nos nós defeituosos tem-se $h(x)/h_0 < 1$, e nos nós onde não há danos tem-se $h(x)/h_0 = 1$. Toda a discretização do campo de danos foi realizada via Método dos Elementos Finitos.

A imposição de dano é realizada escolhendo-se uma posição e um valor para o parâmetro de coesão , dado pela Equação (8)

$$\beta = \left(\frac{h(x)}{h_0} \right)^{1/3}, \quad (8)$$

β varia no intervalo [0, 1]. Se $\beta = 1$ a estrutura está sem danos, sendo $\beta = 0$ uma ruptura local que deve ser considerada.

A RNA abordada nessa pesquisa é uma rede *Perceptron de Múltiplas Camadas* (PMC). Utiliza-se no processo de treinamento supervisionado da rede o algoritmo *backpropagation* com *momentum*. Para geração dos dados de entrada para a RNA utiliza-se a Equação (9),

$$E_k = \frac{|G_{mm}(i, j) - G_{exp}(i, j)|}{|G_{mm}(i, j)|} \cdot 100, \quad (9)$$

onde E_k com $k = 1, 2, \dots, m^2$ representa cada elemento do vetor de entrada E , (i, j) representa cada elemento das matrizes G_{mm} e G_{exp} , com $i, j = 1, 2, \dots, m$. Além dos 100 sinais de entrada

gerados pela Equação (9), foram também utilizadas as 6 primeiras frequências naturais da estrutura danificada. Assim, cada entrada possui 106 elementos.

Neste estudo, segue-se a teoria presente em Pereira (2018) na fase de treinamento da rede. A rede foi treinada sem a presença de dados ruidosos. A rede foi testada em todas as posições da viga (*posição 1, ..., posição 21*) com três valores de parâmetro de coesão em cada posição, $\beta = 0,216$, $\beta = 0,422$ e $\beta = 0,729$. Utilizou-se no processo de treinamento da rede: 106 sinais de entrada, três camadas neurais escondidas com 100 neurônios cada uma, uma camada neural de saída com 21 neurônios, função de ativação: *tansig* nas camadas neurais escondidas e *logsig* na camada neural de saída, algoritmo de treinamento *Gradient descent backpropagation* com *momentum* e taxa adaptativa. Foram estabelecidas 800000 iterações, como o máximo de iterações permitidas, erro final de 10^{-7} , taxa de aprendizagem de 0,4, coeficiente de *momentum* 0,6. A rede foi treinada atingindo o erro estipulado, com 604171 iterações.

Após o treinamento da rede neural, segue-se para a etapa de teste da mesma. As posições escolhidas são baseadas no trabalho de Corrêa (2013), onde mostra-se que as regiões próximas aos extremos da viga apresentam menos sensibilidade à presença de dano, sendo assim mais difícil a correta detecção de danos estruturais nestas localizações. A Tabela 1 apresenta as posições e parâmetros utilizados para teste da rede neural após o treinamento.

Tabela 1- Parâmetros da fase de Teste da RNA

Caso	Posição do dano	Nível de ruído(%)	β	$h(x)/h_0$
1	4	1 e 3	0,512	0,8
2	5	1 e 3	0,512	0,8
3	10	1 e 3	0,512	0,8
4	11	1 e 3	0,512	0,8
5	17	1 e 3	0,512	0,8
6	18	1 e 3	0,512	0,8

Com o objetivo de simular de forma mais realística os dados experimentais, a simulação da presença de ruído nas medições foi abordada, adicionando-se às formas modais de vibração uma perturbação aleatória com distribuição uniforme. Os modos de vibração, com imposição de ruído, são então fornecidos através da seguinte formulação,

$$\Phi_r(i, j) = \Phi(i, j)(1 + \frac{p}{100}rand(-1, 1)), \quad (10)$$

onde Φ_r corresponde aos modos de vibração contaminados por ruído e Φ aos modos de vibração originais; p é o nível de ruído acrescentado; e $rand(-1, 1)$ é uma função geradora de números aleatórios no intervalo $(-1, 1)$.

Nas figuras 3 a 14 são mostrados os resultados gráficos para identificação de danos utilizando RNA. Em todos os casos analisados o método conseguiu localizar o dano e também, vale-se destacar, que obteve-se uma ótima estimativa da intensidade do dano estrutural.

Por meio da análise dos resultados obtidos, pode-se afirmar que mesmo na presença de ruídos de medição, a utilização de RNA juntamente com a matriz de flexibilidade estrutural, na maioria dos casos, foi capaz de localizar os danos simulados. Devido às funções de interpolação lineares utilizadas para se descrever o campo de danos, tem-se o perfil de dano triangular que pode ser observado nos resultados.

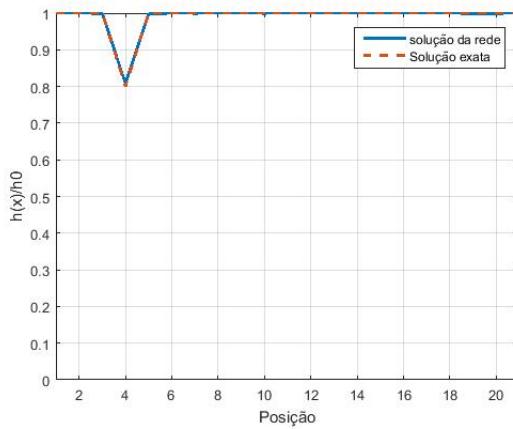


Figura 3- Dano na posição 4 com 1% de ruído.

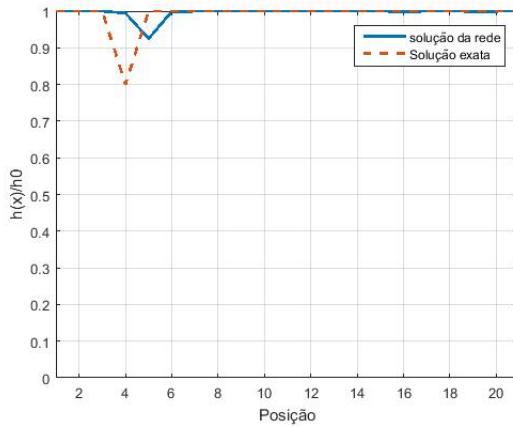


Figura 4- Dano na posição 4 com 3% de ruído.

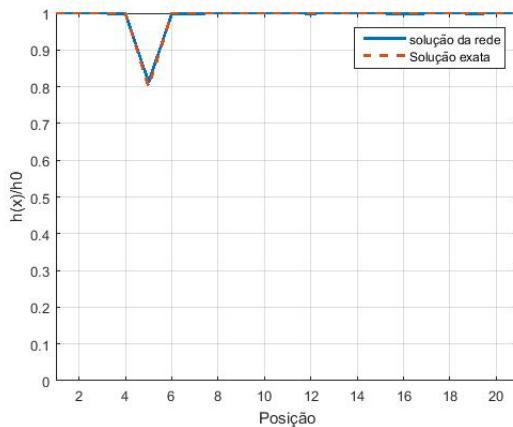


Figura 5- Dano na posição 5 com 1% de ruído.

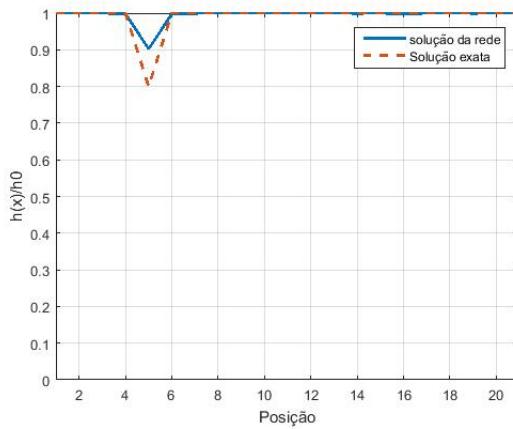


Figura 6- Dano na posição 5 com 3% de ruído.

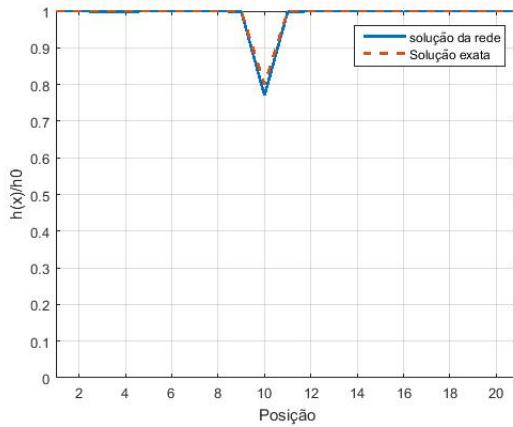


Figura 7- Dano na posição 10 com 1% de ruído.

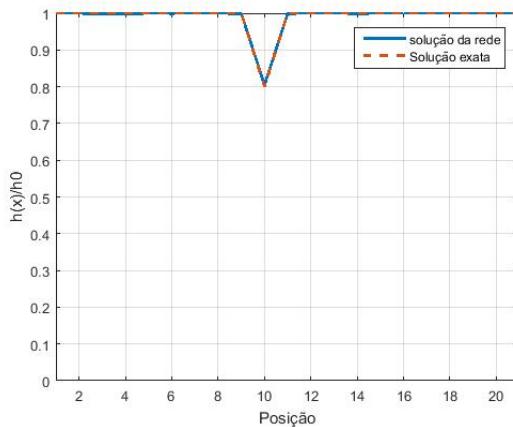


Figura 8- Dano na posição 10 com 3 % de ruído.

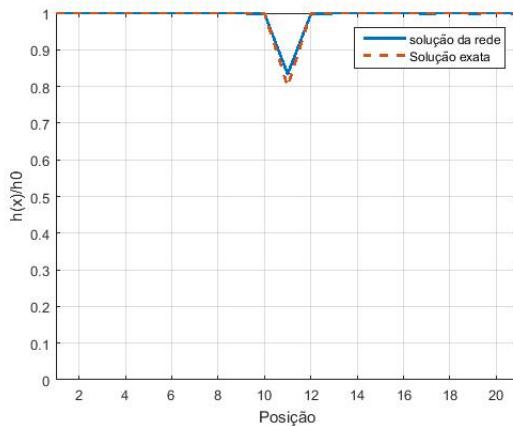


Figura 9- Dano na posição 11 com 1 % de ruído.

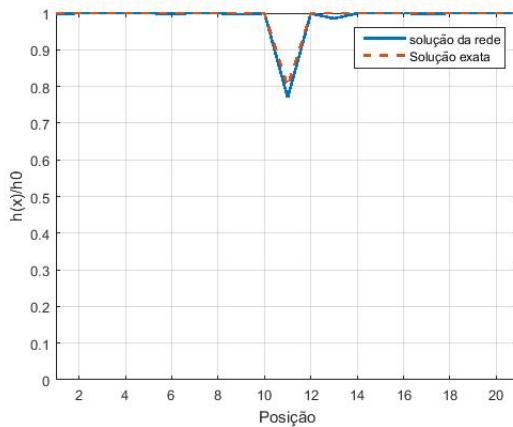


Figura 10- Dano na posição 11 com 3 % de ruído.

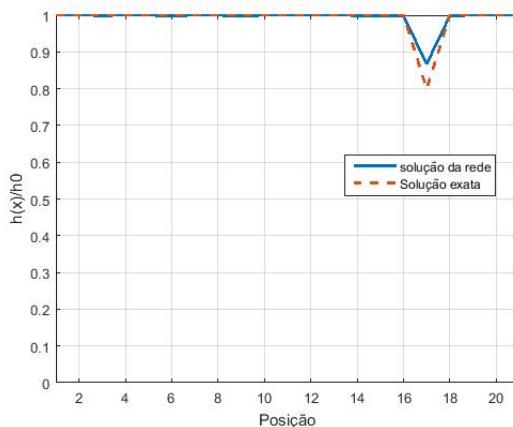


Figura 11- Dano na posição 17 com 1 % de ruído.

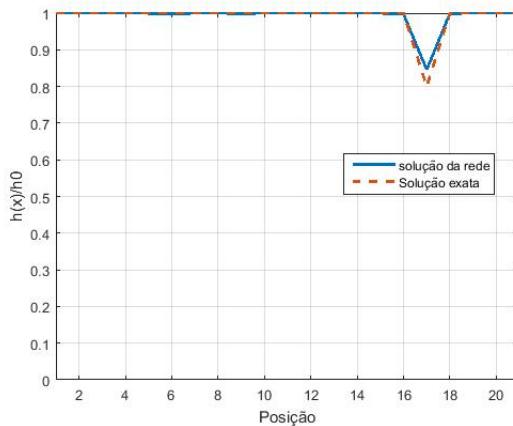


Figura 12- Dano na posição 17 com 3 % de ruído.

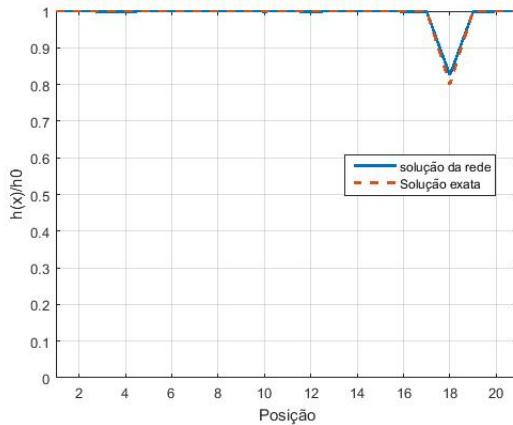


Figura 13- Dano na posição 18 com 1 % de ruído.

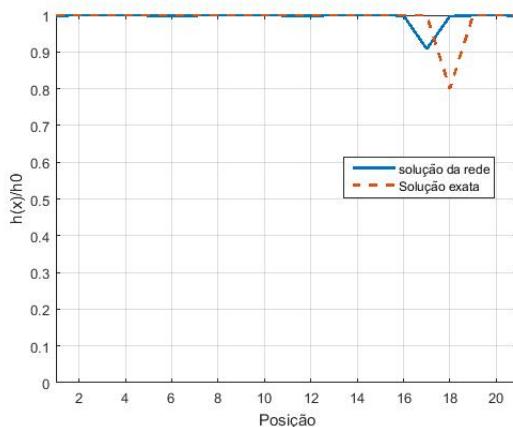


Figura 14- Dano na posição 18 com 3 % de ruído.

5. CONCLUSÕES

A substituição dos métodos de localização de danos apresentados em Corrêa (2013) por RNA apresentou resultados bastante satisfatórios, uma vez que foi possível identificar a região potencialmente danificada na presença de ruídos. Além disso, em alguns casos, obteve-se uma ótima estimativa da intensidade do dano estrutural principalmente na presença de 1% de ruído, o que gera uma grande expectativa da RNA em estruturas mais complexas ser capaz de localizar e quantificar o dano estrutural sem a necessidade de utilização de algum método de otimização. É valido ressaltar que considerou-se um número bastante reduzido de GDL da estrutura, admitindo-se apenas modos de vibração na faixa de 0-450 Hz, o que pode justificar o fato de em alguns casos a rede localizar o dano na posição vizinha a posição correta. Ressalta-se que mais testes devem ser realizados para garantir a eficiência do modelo utilizado, como por exemplo, na simulação de danos múltiplos.

Agradecimentos

Ao apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES). À pós-graduação em Modelagem Computacional em Ciências e Tecnologia (MCCT/UFF). Ao Instituto do Noroeste Fluminense de Educação Superior (INFES/UFF).

Referências

- [1] CORRÊA, R.A.P. *Identificação de danos em estruturas bi-dimensionais via matriz de flexibilidade baseada em um modelo de dano contínuo.* 2013. 131 f. Tese (Doutorado em Modelagem Computacional)– IPRJ/UERJ, Nova Friburgo, 2013.
- [2] GUYAN, R. J. Reduction of stiffness and mass matrices. American Institute of Aeronautics and Astronautics Journal, vol. 3, n. 2, p. 380, 1965. DOI:10.2514/3.2874.
- [3] HAYKIN, S. *Redes neurais: princípios e prática.* 2 ed. Porto Alegre: Bookman, 2001.
- [4] PALMIERE, S. *Arquitetura e topologia de Redes Neurais Artificiais.* 2016. Disponível em: <https://www.embarcados.com.br/redes-neurais-artificiais/>. Acesso em: 10 de out. de 2018.
- [5] PEREIRA, J. L. *Identificação de danos estruturais baseada em um modelo de dano contínuo via Redes Neurais Artificiais.* 2018. 94 f. Dissertação (Mestrado em Modelagem Computacional)–MCCT/UFF, Volta Redonda, 2018.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

ALGORITMOS DE INTELIGÊNCIA DE ENXAME COM BUSCA POR PULL MOVE APLICADOS AO PROBLEMA DE PREDIÇÃO DE ESTRUTURA DE PROTEÍNAS

Christiane Regina Soares Brasil¹ - christiane@ufu.br

Douglas Monteiro Cavalcanti¹ - douglas.cavalcanti@ufu.br

¹Faculdade de Computação - Universidade Federal de Uberlândia

Resumo. O problema da predição da estrutura de uma proteína a partir de sua sequência de aminoácidos é um problema da classe NP-completo, e por isso não há abordagem determinística capaz de tratá-lo. Neste sentido, diversos métodos não determinísticos baseados em meta-heurísticas têm sido desenvolvidos para tratamento do problema. Estes métodos, apesar de não garantirem soluções ótimas, são capazes de gerar soluções boas em tempo de execução viável. Neste artigo, duas abordagens não determinísticas baseadas em Inteligência de Enxame, Otimização por Colônia de Formiga e Otimização por Enxame de Partícula foram aplicadas ao Problema de Predição de Proteínas no modelo simplificado Hidrofóbico-Polar 2D, incorporando uma heurística como método de busca local chamada de busca por Pull Move. Os resultados mostraram que o algoritmo de Otimização por Enxame de Partículas com Pull Move é um poderoso método para predição de estruturas de proteínas, superando o ACO e alcançando a solução ótima para todas as proteínas da base de testes.

Keywords: Otimização por Enxame de Partícula, Otimização por Colônia de Formiga, Predição de Estrutura de Proteínas, Inteligência de Enxame, Modelo 2D HP

1. Introdução

A sofisticação, robustez e a adaptabilidade dos fenômenos de auto-organização encontrados na natureza representam uma motivação poderosa para replicá-los na tentativa de gerar métodos computacionais que consigam lidar com situações onde técnicas convencionais de otimização são limitadas. As técnicas Otimização por Colônia de Formiga (ACO, do inglês *Ant Colony Optimization*) [9] e Otimização por Enxame de Partícula (PSO, do inglês *Particle Swarm Optimization*) [10] são os principais exemplos de técnicas computacionais de Inteligência de Enxame, e em anos recentes, têm sido usadas com sucesso no tratamento de problemas para os quais não se conhece método determinístico capaz de gerar soluções em tempo hábil.

O Problema de Predição de Estrutura de Proteínas (PSP, do inglês *Protein Structure Prediction*), um dos mais desafiadores da área da bioinformática, é um exemplo de problema intratável a partir de abordagens analíticas [1]. O problema consiste em tentar predizer a estrutura que uma proteína adota após o processo de dobramento. Métodos convencionais para determinação das estruturas das proteínas, como difração de raio-x e ressonância nuclear magnética, são limitados, caros e demorados [2]. Além disso, muitos dos fatores envolvidos no processo de dobramento são ainda desconhecidos, motivo pelo qual o problema é normalmente tratado a partir de modelos simplificados que consideram apenas algumas características das proteínas.

No entanto, mesmo a partir de modelos simplificados, predizer a estrutura de uma proteína ainda é um problema da classe NP-completo, não havendo, portanto, abordagem determinística capaz de tratá-lo.

Neste trabalho, as técnicas ACO e PSO foram aplicadas ao problema de predição de estrutura de proteínas no modelo simplificado Hidrofóbico-Polar (HP) 2D [11], incorporando a ambos os métodos uma heurística auxiliar de busca chamada de busca por *pull move* [12]. Diversos trabalhos na literatura já mostraram a eficiência de ambos os métodos para o problema [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. Neste sentido, o objetivo deste artigo foi comparar os dois algoritmos incorporando a heurística de busca por *pull move*.

Os resultados foram promissores, com o ACO alcançando o resultado ótimo para 7 das 8 proteínas da base de testes [14], e o PSO, alcançando a solução ótima para todas as 8 proteínas, sendo que este caracterizou uma nova abordagem com busca por *pull move* para o método. Além disso, estes resultados evidenciaram que, apesar do ACO ser naturalmente voltado para problemas de otimização combinatorial, o PSO, que precisou ser adaptado dada a característica discreta do problema, é mais eficaz para o problema.

2. O Problema da Predição da Estrutura de Proteínas e o Modelo HP

Esta seção começa descrevendo o PSP e as principais abordagens para tratamento do problema. Em seguida, descreve com mais detalhes o Modelo HP e a forma de representação das conformações.

2.1 O Problema da Predição da Estrutura de Proteínas

As proteínas são compostos orgânicos que desempenham funções fundamentais nos organismos vivos, como catálise, transporte, motilidade, armazenamento, regulação e defesa [1]. Em condições adequadas, as proteínas se enovelam (dobram) em uma estrutura tridimensional, assumindo uma conformação específica, apesar das inúmeras possibilidades de dobramento, e alcançando o estado de mais baixa energia do ponto de vista termodinâmico. Esta conformação é chamada de conformação nativa da proteína, e está diretamente relacionada com a sua função biológica. Conhecer a estrutura de uma proteína significa também entender sua atividade funcional, e por isso, o estudo e a catalogação das estruturas das proteínas é de grande importância nas mais diversas áreas da biologia, para fins farmacêuticos e da Medicina, de modo geral.

O PSP consiste em prever a estrutura tridimensional de uma proteína [1]. Neste contexto, há diferentes abordagens para o processo de predição, baseadas em algum conhecimento prévio da proteína alvo ou não. Na abordagem *ab initio*, tratada neste trabalho, a predição é feita conhecendo-se unicamente a estrutura primária¹ da molécula. Para isso, realiza-se uma busca no espaço conformacional a fim de encontrar a estrutura de energia mínima. Há, todavia, uma dificuldade em construir modelos computacionais para o PSP, pois existem limitações no entendimento dos critérios que influenciam o enovelamento proteico. Devido a esta limitação, foram desenvolvidos modelos simplificados que não consideram alguns detalhes e informações do processo de enovelamento. Estes modelos simplificam tanto a estrutura da proteína quanto a forma como a energia é calculada e são conhecidos como modelos baseados em *lattice*.

¹Recebe o nome de estrutura primária a sequência de aminoácidos que constitui uma proteína.

No modelo baseado em *lattice*, cada aminoácido da proteína é representado como um ponto, as ligações entre os aminoácidos são representadas por arestas e os aminoácidos são distribuídos em uma malha dividida em unidades que podem conter até um aminoácido cada. Os ângulos das ligações podem assumir apenas alguns poucos valores discretos, definidos pela estrutura da malha. Neste artigo foi utilizado um modelo baseado em *lattice*.

2.2 O Modelo HP

O Modelo HP [11] é baseado em *lattice* e considera apenas a propriedade da hidrofobicidade dos aminoácidos das proteínas. Os aminoácidos são classificados em hidrofóbicos (H) e hidrofílicos, ou polares (P). Aminoácidos subsequentes na estrutura primária da proteína ocupam posições adjacentes na malha. Os aminoácidos adjacentes não conectados são chamados de vizinhos. No modelo, as conformações são geradas a fim de maximizar o número de vizinhos hidrofóbicos. A representação da proteína pode ser feita a partir de duas diferentes abordagens, considerando uma malha bidimensional, abordagem chamada de HP 2D, ou considerando uma malha tridimensional, abordagem chamada de HP 3D.

Seja uma proteína no Modelo HP dada pela sequência S definida como $s_1 s_2 \dots s_n$, onde s_i denota o i -ésimo aminoácido da sequência e n o comprimento da proteína. A energia mínima E de uma proteína é dada pela Equação 1, onde E_{ij} assume o valor -1 se s_i e s_j são ambos do tipo H, ou 0 caso contrário, e Δr_{ij} assume o valor 1 se os aminoácidos s_i e s_j são vizinhos, ou 0 caso contrário.

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Delta r_{ij} E_{ij} \quad (1)$$

Neste artigo, a conformação no Modelo HP será representada pela direção relativa das arestas entre os aminoácidos. Nesta forma de representação, uma aresta pode assumir o valor reto (S, do inglês *straight*), direito (R, do inglês *right*) ou esquerdo (L, do inglês *left*).

3. Otimização por Colônia de Formiga

O ACO [9] é uma meta-heurística originalmente introduzida para problemas de otimização combinatorial, inspirada no comportamento das formigas ao trilhar um caminho entre o formigueiro e a fonte de alimento.

Nesta seção será descrito o funcionamento do ACO, seus parâmetros e regras.

3.1 Funcionamento

O funcionamento do ACO é realizado pela construção de soluções candidatas para um problema de otimização combinatorial. Estas soluções são construídas por formigas artificiais que decidem como compôr suas soluções baseadas em uma regra probabilística que considera eventuais informações heurísticas sobre os componentes a serem combinados e informações de uma memória compartilhada, baseada em trilhas de feromônio [9, 29]. Ao concluir o procedimento de construção, cada formiga deposita uma quantidade de feromônio nos componentes da solução construída. Deste modo, componentes mais usados ganham destaque entre os outros, aumentando a probabilidade de que sejam escolhidos novamente.

Para contrabalancear a avaliação positiva dos componentes, há o processo de evaporação do feromônio, que garante que trilhas geradas em um momento mais avançado do algoritmo consigam competir com trilhas mais antigas, já consolidadas pelo feromônio. Por meio da evaporação, o feromônio acumulado nos componentes é reduzido a cada iteração. Desta maneira, as trilhas cuja qualidade já tenha sido superada, são esquecidas conforme é diminuída a frequência de formigas que optam por elas. O critério de parada é dado pelo número máximo de iterações.

Em [29], os autores adaptaram o ACO incorporando ao algoritmo um método auxiliar de busca local. Os resultados obtidos mostraram que o uso do método de busca local aumenta significativamente a performance do ACO.

3.2 Regras e Parâmetros

A regra probabilística para construção das soluções é dada pela Equação 2

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{j=1}^n (\tau_{ij})^\alpha (\eta_{ij})^\beta} \quad (2)$$

onde p_{ij} é a probabilidade de uma formiga incluir o componente j na solução, τ_{ij} é a intensidade do feromônio na aresta que conecta o componente i ao componente j , η_{ij} é a informação heurística associada a esta mesma aresta, n é o número de componentes disponíveis, α é um valor real positivo que regula a influência do feromônio e β é um valor real positivo que regula a influência da informação heurística.

A evaporação do feromônio é realizada seguindo a Equação 3:

$$\tau_{ij} = \rho \tau_{ij} \quad (3)$$

onde ρ é um valor real no intervalo de 0 a 1, que controla a taxa de persistência do feromônio.

O depósito de feromônio é executado por cada formiga seguindo a Equação 4:

$$\tau_{ij} = \tau_{ij} + \frac{1}{l} \quad (4)$$

onde l é o custo da solução (valor a ser minimizado pelo algoritmo).

No início do algoritmo, os componentes são inicializados com um valor arbitrário de feromônio, denotado por τ_0 .

3.3 O Algoritmo Implementado

O algoritmo ACO implementado para este artigo teve como base a versão do ACO implementada em [13], e conta com 3 fases: a fase de construção, a fase de busca local e a fase de atualização do feromônio. Essas fases são executadas em sequência a cada iteração do algoritmo. A melhor conformação encontrada pelo algoritmo é retornada como solução.

Fase de Construção: Nesta fase, cada formiga constrói sua conformação e calcula sua energia. O procedimento de construção é um laço que itera por todas as arestas da proteína sorteando seus valores, sendo que a probabilidade da aresta i assumir o valor d , com $d \in \{S, L, R\}$ é dada pelas Equações 5 e 6:

$$p_{id} = \frac{(\tau_{id})^\alpha (\eta_{id})^\beta}{\sum_{d \in \{S, L, R\}} (\tau_{id})^\alpha (\eta_{id})^\beta} \quad (5)$$

$$\eta_{ij} = \exp(\Delta E, 0.3) \quad (6)$$

onde, no cálculo da informação heurística, ΔE é o número de novos contatos que serão gerados se a aresta i assumir o valor d .

A construção de uma conformação pode ser inviabilizada caso o procedimento leve a um estado em que não é possível posicionar uma aresta sem que isto acarrete na colisão de aminoácidos. A solução adotada para gerenciamento deste tipo de estado é o método de cópia parcial. Quando um estado inviável é atingido, a conformação parcialmente construída é descartada e substituída por uma cópia parcial da melhor conformação da população. Esta cópia tem comprimento equivalente ao comprimento da conformação descartada, de forma que a formiga continua a construção da aresta onde parou.

Fase de Busca Local: Nesta fase, cada formiga ou partícula aplica à sua conformação o método de busca por *pull moves*. *Pull move* é um conjunto de movimentos que pode reduzir a energia de uma conformação, reorganizando alguns de seus aminoácidos. A seguir, define-se a implementação de um *pull move* [12]: Considere o i -ésimo aminoácido de uma proteína, localizado na posição (x_i, y_i) da malha. Seja F uma posição livre adjacente à posição (x_{i+1}, y_{i+1}) , e diagonalmente adjacente à posição (x_i, y_i) . Os aminoácidos das posições (x_i, y_i) , (x_{i+1}, y_{i+1}) e a posição livre F formam os 3 vértices de um quadrado. Seja C o quarto vértice deste quadrado. Um *pull move* pode ser performado se C é igual a (x_{i-1}, y_{i-1}) (Figura 2 (a)) ou se C é uma posição livre da malha (Figura 2 (b)).

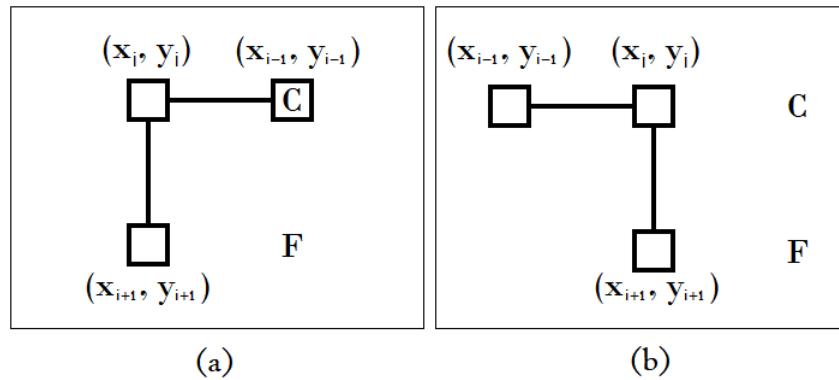


Figura 1- Exemplo de posicionamentos adequados dos aminoácidos para aplicação do *pull move*.

Quando C é igual a (x_{i-1}, y_{i-1}) , a aplicação do *pull move* consiste em somente mover o aminoácido i para a posição F . Quando C é uma posição livre, o aminoácido i é movido para a posição F e o aminoácido $i - 1$ para posição C . Neste último caso, a operação pode resultar no afastamento dos aminoácidos $i - 1$ e $i - 2$, que deixarão de ser adjacentes na malha. Para resolver este problema, os aminoácidos que antecedem o de índice $i - 1$ são movidos um a um até que uma conformação válida seja atingida. Este processo é iniciado movendo o aminoácido $i - 2$ para a posição antes ocupada pelo aminoácido i , a posição $i - 3$ para a posição antes ocupado pelo aminoácido $i - 1$, e assim por diante, até que todos os aminoácidos consecutivos na sequência ocupem posições adjacentes na malha.

Uma mesma conformação pode conter várias possibilidades de *pull move*, e por isso o método de busca lista todos os possíveis *pull moves* e seleciona aquele que mais decrementa a energia da conformação. Após a aplicação do *pull move*, uma nova conformação é gerada e o método de busca é repetido sobre a nova conformação. Este laço se mantém enquanto houver redução na energia.

Fase de Atualização do Feromônio: Nesta fase, cada formiga atualiza os valores de feromônio. A estrutura usada para representar o feromônio é uma matriz que armazena, para cada aresta da proteína, o nível de feromônio (representado por um número real positivo) para cada um dos 3 possíveis valores que a aresta pode assumir: $\{L, R, S\}$. Tem-se, portanto, uma matriz τ_{ij} de ponto flutuante, de dimensão $(n-1)^3$, onde n é o número de aminoácidos da proteína. Ao fim da etapa de construção e da etapa de aplicação da busca por *pull moves*, cada formiga atualiza a matriz de feromônio, seguindo as Equações 7 e 8:

$$\tau_{ij} = \rho\tau_{ij} + \Delta \quad (7)$$

$$\Delta = \frac{E(c)}{E(best)^3} \quad (8)$$

onde, $E(c)$ é a energia da conformação encontrada pela formiga e $E(best)$ é a energia da melhor solução encontrada pelo algoritmo até o momento. Neste ponto, o ACO implementado difere do HACO. No HACO, $E(best)$ não diz respeito a energia da melhor solução encontrada, mas ao valor da energia de uma conformação ótima, informação passada como dado de entrada para o algoritmo.

4. Otimização por Enxame de Partículas

O PSO é uma meta-heurística introduzida em 1995 por Kennedy e Eberhart [10], para otimização de funções contínuas não lineares, inspirada no comportamento social observado em algumas espécies de seres vivos. O conceito do PSO foi fundamentado a partir do estudo e do desenvolvimento de simulações simplificadas de sistemas sociais que buscaram representar a coreografia imprevisível de uma revoada de pássaros.

Nesta seção será descrito o funcionamento do PSO, bem como seus parâmetros e regras.

4.1 Funcionamento

Nesta técnica de otimização, uma possível solução para o problema é representada por uma partícula em um espaço de busca. As coordenadas desta partícula definem a solução representada por ela, e a qualidade desta posição é calculada por meio da função relacionada ao problema alvo, chamada de função *fitness*. Cada partícula possui uma taxa de variação de sua posição, chamada de velocidade, e mantém um registro de sua melhor posição prévia, chamada de *pbest*. A melhor posição dentre todas as posições encontradas pela população de partículas é chamada de *gbest* e é conhecida por todo enxame [10].

O algoritmo é iniciado com a população de partículas distribuída aleatoriamente sobre o espaço de busca, e a cada iteração, para cada partícula são calculados os vetores diferença entre a posição atual da partícula e as posições *pbest* e *gbest*. A velocidade da partícula é ajustada na direção da soma destes vetores, podendo ser limitada a um valor máximo.

4.2 Regras e Parâmetros

A velocidade v de uma partícula e sua posição x na iteração $i + 1$ são calculadas seguindo as Equações 11 e 12 [31]:

$$v_{i+1} = v_i w + r_1 \varphi_1(pbest - x_i) + r_2 \varphi_2(gbest - x_i) \quad (9)$$

$$x_{i+1} = x_i + v_{i+1} \quad (10)$$

A influência da posição $pbest$ sobre o cálculo da velocidade é controlada pelo fator cognitivo, denotado por φ_1 , enquanto que a influência da posição $gbest$ é controlada pelo fator social, denotado por φ_2 . Estes parâmetros podem assumir qualquer valor real positivo. Os fatores r_1 e r_2 são dois valores randômicos dentro do intervalo de 0 a 1, definidos a cada iteração, que multiplicam os fatores cognitivo e social, respectivamente, inserindo um fator aleatório na movimentação das partículas. O controle da influência da velocidade v_i sobre a velocidade v_{i+1} é feita pelo parâmetro w , chamado de peso de inércia, cujo valor pode variar no intervalo de 0 a 1.

4.3 O Algoritmo Implementado

O PSO é originalmente contínuo. Aplicá-lo ao problema 2D HP exige adaptar o algoritmo para um espaço de busca discreto. Tal adaptação foi realizada usando como base o *set-based* PSO [32], que redefine a posição e velocidade de uma partícula, bem como os operadores necessários, para um espaço discreto, usando uma representação baseada em conjuntos. Nesta versão discreta do algoritmo, a velocidade de uma partícula em uma dimensão i é um conjunto de tuplas do tipo $(d, p(d))$, onde $d \in \{S, L, R\}$, e $p(d)$ é a probabilidade associada ao elemento d nesta dimensão. A posição de uma partícula é um conjunto de tuplas do tipo (i, d) , que associa uma direção d a uma aresta (dimensão) i .

A fase de construção é incorporada ao procedimento de atualização da posição e funciona da mesma maneira que a fase de construção do ACO 2D HP, exceto pela regra probabilística. A probabilidade de uma partícula escolher determinada direção d para uma aresta i é definida pela Equação 20:

$$p_{id} = \frac{p(d)(\eta_{id})^\beta}{\sum_{d \in \{S, L, R\}} p(d)(\eta_{id})^\beta} \quad (11)$$

onde $p(d)$ é a probabilidade associada à direção $d \in \{S, L, R\}$ na dimensão i , e β é um valor real positivo que regula a influência da informação heurística.

A fase de busca por *pull moves* é iniciada após todas partículas construírem suas conformações.

5. Análise Experimental

Os algoritmos foram implementados na linguagem C e todos os experimentos foram performedos em um Notebook com processador Intel Core i5-7200U CPU @ 2.50GHz 4, com 8 GB de RAM, em um ambiente Linux Ubuntu 16.04 LTS 64 bit.

O experimento consistiu em aplicar o ACO e o PSO, com e sem o uso do método de busca por *pull move*, às proteínas listadas na Tabela 1, onde E_{best} é a menor energia conhecida para a proteína e T é o tamanho da sequência. Os parâmetros do ACO foram configurados como $\tau_0 = 1/3$, $\alpha = 1$ e $\beta = 2$, enquanto os parâmetros do PSO foram configurados como $\varphi_1 = \varphi_2 = 2, 1$, $w = 0, 5$ e $\beta = 2$. Para ambos os algoritmos, o número de iterações foi

definido como 500 e o tamanho da população foi definido como 200 para instâncias de tamanho menor ou igual a 25 e 1500 para as demais instâncias.

Tabela 1- Tabela de *Benchmark*.

<i>id</i>	Proteína	E_{best}	<i>T</i>
1	$HPHP_2H_2PHP_2HPH_2P_2HPH$	-9	20
2	$H_2P_2(HP_2)_6H_2$	-9	24
3	$P_2HP_2(H_2P_4)_3H_2$	-8	25
4	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$	-14	36
5	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$	-23	48
6	$H_2(PH)_3PH_4P(HP_3)_2HP_4(HP_3)_2HPH_4(PH)_3PH_2$	-21	50
7	$P_2H_3PH_8P_3H_{10}PHP_3H_{12}P_4H_6PH_2PHP$	-36	60
8	$H_{12}(PH)_2((P_2H_2)_2P_2H)_3(PH)_2H_{11}$	-42	64

Ambos os algoritmos foram executadas 20 vezes de maneira independente, para cada uma das proteínas. Ao fim de cada conjunto de 20 execuções foi recuperada a conformação de energia mínima, bem como o tempo médio gasto pelo algoritmo. A Tabela 2 apresenta a comparação entre os resultados alcançados por ambos os algoritmos.

Tabela 2- Comparação ACO e PSO.

Proteína	ACO _{padrão}			ACO _{pull move}			PSO _{padrão}			PSO _{pull move}		
	<i>id</i>	E_{best}	E_{min}	<i>t</i> (s)	E_{min}	<i>t</i> (s)	E_{min}	<i>t</i> (s)	E_{min}	<i>t</i> (s)	E_{min}	<i>t</i> (s)
1	-9	-9	0.26	-9	1.00	-9	0.25	-9	0.49			
2	-9	-9	0.30	-9	0.83	-9	0.30	-9	0.69			
3	-8	-7	0.33	-8	1.78	-7	0.31	-8	0.96			
4	-14	-14	3.75	-14	13.21	-14	3.41	-14	7.98			
5	-23	-22	5.23	-23	24.76	-23	4.69	-23	15.80			
6	-21	-21	5.29	-21	23.18	-21	4.74	-21	13.97			
7	-36	-35	7.06	-36	31.61	-35	6.27	-36	18.64			
8	-42	-38	6.97	-39	19.61	-40	6.47	-42	14.67			

6. Conclusão

Por meio dos experimentos descritos, pode-se observar que o PSO superou os resultados obtidos pelo ACO para o problema em questão. Apenas o PSO conseguiu encontrar a conformação de energia mínima para a maior proteína. Além disso, o algoritmo registrou as menores médias de tempo de execução, evidenciando seu bom desempenho computacional. Além disso, os experimentos mostram resultados animadores com a inserção do método de busca *pull move*, assinalando o PSO com *pull move* como uma técnica eficaz para o problema, tanto quanto à qualidade das soluções quanto ao tempo médio de execução.

Referências

- [1] J. A. Doudna, & M. M. Cox, Biologia Molecular-Princípios e Técnicas. 2012.
- [2] D. L. Nelson & M. M. Cox, Lehninger: principios de bioquímica. 2015.
- [3] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni & M. Yannakakis, On the complexity of protein folding, *Journal of computational biology* 5.3, pp. 423-465, 1998.
- [4] A. C. Kaushik & Shakti Sahi, Biological complexity: ant colony meta-heuristic optimization algorithm for protein folding, *Neural Computing and Applications* 28.11, pp. 3385-3391, 2017.
- [5] D. D. Do, A novel and efficient ant colony optimization algorithm for protein 3D structure prediction, VNU-UET Technical Report, 2017.
- [6] T. Hu, M. Hu, L. Lv, C. Zhou, Improved Genetic Algorithm-Particle Swarm Optimization Based on Multiple Populations for 3D Protein Structure Prediction, *Journal of Computational and Theoretical Nanoscience* 12.7, pp. 1414-1419, 2015.
- [7] Y. Guo, Z. Wu, Y. Wang & Y. Wang, Extended particle swarm optimisation method for folding protein on triangular lattice, *IET systems biology* 10.1, pp. 30-33, 2016.
- [8] X. Zheng, L. Wu, S. Ye & R. Chen, Simplified Swarm Optimization-Based Function Module Detection in Protein–Protein Interaction Networks, *Applied Sciences* 7.4, p. 412, 2017.
- [9] M. Dorigo, Optimization, learning and natural algorithms, PhD Thesis, Politecnico di Milano, 1992.
- [10] R. Eberhart, & J. Kennedy, A new optimizer using particle swarm theory, *Micro Machine and Human Science*, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, 1995.
- [11] K. F. Lau, & K. A. Dill, A lattice statistical mechanics model of the conformational and sequence spaces of proteins, *Macromolecules* 22.10, 1989.
- [12] N. Lesh, M. Mitzenmacher, & S. Whitesides, A complete and effective move set for simplified protein folding, *Proceedings of the seventh annual international conference on Research in computational molecular biology*. ACM, 2003.
- [13] Z. Liu, & Z. Yang, Heuristic ant colony optimization algorithm for predicting the structures of 2D HP model proteins, *Biomedical Engineering and Informatics (BMEI)*, 2014 7th International Conference on. IEEE, 2014.
- [14] A. Shmygelska, R. Aguirre-Hernandez, & H. H. Hoos, An ant colony optimization algorithm for the 2D HP protein folding problem, *International Workshop on Ant Algorithms*. Springer, Berlin, Heidelberg, 2002.
- [15] S. Fidanova, & I. Lirkov, Ant colony system approach for protein folding, *Computer Science and Information Technology*, 2008. IMCSIT 2008. International Multiconference on. IEEE, 2008.
- [16] X. Hu, J. Zhang, & Y. Li, Flexible protein folding by ant colony optimization, *Computational Intelligence in Biomedicine and Bioinformatics*. Springer, Berlin, Heidelberg, 2008. 317-336.
- [17] A. G. Citrolo, & G. Mauri, A hybrid Monte Carlo ant colony optimization approach for protein structure prediction in the HP model, *arXiv preprint arXiv:1309.7690*, 2013.
- [18] N. Thilagavathi & T. Amudha, Rank based ant algorithm for 2D-HP protein folding, *Computational Intelligence in Data Mining-Volume 3*. Springer, New Delhi, pp. 441-451, 2015.

- [19] A. Llanes, C. Vélez, A. M. Sánchez, H. Pérez-Sánchez, J. M. Cecilia, Parallel ant colony optimization for the hp protein folding problem, International Conference on Bioinformatics and Biomedical Engineering. Springer, Cham, 2016.
- [20] A. Băutu, & H. Luchian, Protein structure prediction in lattice models with particle swarm optimization, International Conference on Swarm Intelligence. Springer, Berlin, Heidelberg, 2010.
- [21] C. Lin, & S. Su, Protein 3D HP Model Folding Simulation Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization, International Journal of Fuzzy Systems 13.2, 2011.
- [22] N. Mansour, F. Kanj, & H. Khachfe, Particle swarm optimization approach for protein structure prediction in the 3D HP model, Interdisciplinary Sciences: Computational Life Sciences 4.3, pp. 190-200, 2012.
- [23] N. D. Jana, & J. Sil, Particle swarm optimization with backtracking in protein structure prediction problem, Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on. IEEE, 2012.
- [24] J. Xiao, L. Li, & X. Hu, Solving lattice protein folding problems by discrete particle swarm optimization, Journal of Computers 9.8, 2014.
- [25] Y. Cheng-Hong , L. Yu-Shiun , C. Li-Yeh & C. Hsueh-Wei, Hybrid High Exploration Particle Swarm Optimization Algorithm Improves the Prediction of the 2-Dimensional Hydrophobic-Polar Model for Protein Folding, Current Bioinformatics 13.2, pp. 182-192, 2018.
- [26] C. R. S. Brasil & J. Manfrin, Comparando Algoritmos de Otimização Computacional Aplicados ao Problema de Predição de Estruturas Proteicas com Modelo HP-2D. Revista Brasileira de Computação Aplicada Volume 9, pp. 87–99, 2017.
- [27] C. R. S. Brasil & J. Manfrin, Comportamento da Energia no Algoritmo de Colônia de Formiga para Predição de Estruturas de Proteínas. Anais do Computer on the Beach, pp. 831–840, 2018.
- [28] W. Chen, et al, A novel set-based particle swarm optimization method for discrete optimization problems, IEEE Transactions on evolutionary computation 14.2, pp. 278-300, 2010.
- [29] M. Dorigo, V. Maniezzo, & A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26.1, pp. 29-41, 1996.
- [30] G. Yan, & B. Shi, The Application of Ant Colony Optimization Algorithm in DNA Encoding, Journal of Computational Information Systems, Vol.7, pp.3591-3598, 2011.
- [31] W. X. Ran, L. Liu, & G. M. Yang, A hybrid ant colony algorithm for vehicle routing problem with time windows, Information Technology Journal, Vol.12, pp.5701-5706, 2013.
- [32] Y. Shi & R. Eberhart, A modified particle swarm optimizer, Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on. IEEE, 1998.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

APLICAÇÃO DO MÉTODO DAS DIFERENÇAS FINITAS EM UM PROBLEMA DE TRANSFERÊNCIA DE CALOR UTILIZANDO AS FORMULAÇÕES EXPLÍCITA E IMPLÍCITA

Alex Tavares Silva¹ - altasilva@gmail.com

Wagner Rambaldi Telles^{1,2} - wtelles@id.uff.br

Gustavo Silva Semaan^{1,2} - gustavosemaan@gmail.com

¹Universidade Federal Fluminense, MCCT - Volta Redonda, RJ, Brasil

²Universidade Federal Fluminense, INFES - Santo Antônio de Pádua, RJ, Brasil

Resumo. Este trabalho apresenta a resolução de um problema direto de transferência de calor utilizando o Método das Diferenças Finitas com as formulações explícita e implícita. Uma solução numérica é obtida e comparada com os dados experimentais oriundos de um experimento realizado na literatura, onde uma placa é aquecida e isolada termicamente em seu contorno, tendo como objetivo analisar a melhor configuração para as malhas espacial e temporal adotadas no referido método. Para a formulação implícita utilizou-se o método de Gauss-Seidel para resolver o sistema de equações resultante. Os resultados obtidos pelas formulações são compatíveis e satisfatórios em relação aos dados experimentais da literatura, com destaque para a formulação implícita, a qual apresentou as melhores simulações.

Keywords: Transferência de Calor, Diferenças Finitas, Formulações Explícita e Implícita

1. INTRODUÇÃO

Problemas que envolvem transferência de calor são de grande relevância para as engenharias. Pode-se apontar várias situações em que o assunto deve ser considerado, como: microprocessadores de computadores que não podem esquentar demais para que seu funcionamento não seja afetado, geração de energia através do aquecimento de placas solares, manutenção de determinadas condições climáticas com o isolamento de tetos e paredes de casas e edifícios. Além disso, segundo Jen et. al. (2006) o fenômeno da transferência de calor está presente em diversos ramos do conhecimento humano, o que pode ser percebido em equipamentos que contribuem para o nosso bem-estar e qualidade de vida, como aparelhos de ar-condicionado de residências e escritórios, radiadores dos motores dos carros e caldeiras industriais, dentre outros.

Nesse contexto, um método muito utilizado na modelagem de problemas é o Método das Diferenças Finitas, que possui duas formulações, a explícita e a implícita.

2. DESCRIÇÃO E MODELAGEM MATEMÁTICA DO PROBLEMA DIRETO DE TRANSFERÊNCIA DE CALOR

O problema proposto para este trabalho é baseado no experimento realizado por (Carollo, 2010), representado na Fig. 1, o qual trata-se de uma amostra que está compreendida entre um

aquecedor e um isolante t rmico de forma que se deseja obter a temperatura final da mesma, constitu da de um material de a o inox modelo AISI 304, de comprimento L , onde no final se encontra um termopar, com o objetivo de se aferir tal temperatura (Carollo, 2010).

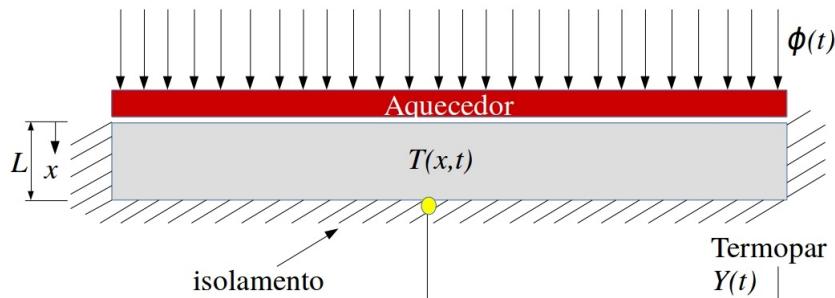


Figura 1- Representa o esquem tica do problema proposto.

Fonte: Adaptado de Carollo (2010).

Devido ao isolamento no contorno da placa, considera-se que n o h a varia o da temperatura na longitudinal e transversal, apenas a varia o vertical, ou seja, a varia o referente 脿 sua espessura. Assim, o modelo matem tico unidimensional desse problema 莩 representado pelas Eqs. (1) - (4).

$$\frac{\partial^2 T(x, t)}{\partial x^2} = \frac{\rho c_p}{\lambda} \frac{\partial T(x, t)}{\partial t}, \text{ em } 0 < x < L \text{ e } t > 0. \quad (1)$$

$$-\lambda \frac{\partial T(x, t)}{\partial x} = \phi(t), \text{ em } x = 0 \text{ e } t > 0. \quad (2)$$

$$\frac{\partial T(x, t)}{\partial x} = 0, \text{ em } x = L \text{ e } t > 0. \quad (3)$$

$$T(x, t) = T_0, \text{ em } 0 \leq x \leq L \text{ e } t = 0. \quad (4)$$

onde x 莩 a coordenada cartesiana (m), t 莩 o tempo (s), $\phi(t)$ 莩 o fluxo de calor (W/m^2) imposto no instante de tempo t e T_0 莩 a temperatura inicial do corpo ($^{\circ}C$).

Segundo Burden (2016), Equa es Diferenciais Parciais (EDP) se classificam de forma parecida com a classifica o de se es c nicas. Considere a seguinte equa o linear de segunda ordem com coeficientes constantes, conforme a Eq. (5).

$$au_{xx} + 2bu_{xy} + cu_{yy} + du_x + eu_y + fu = g, \quad (5)$$

Diz-se que a Equa o Diferencial Parcial 莩 classificada como: el ptica, se $b^2 - ac < 0$; parab lica, se $b^2 - ac = 0$; hiperb lica, se $b^2 - ac > 0$. De acordo com esses crit rios, tem-se que a equa o do problema 莩 parab lica e isso indica uma continuidade na fun o em todo o intervalo. Essa Equa o Diferencial Parcial 莩 resolvida utilizando o M todo das Diferen as Finitas com as formula es expl cita e impl cita.

3. M TODO DAS DIFEREN AS FINITAS

Em m todos num ricos existem v rias t cnicas de solu es num ricas e, dentro desse grande universo, o M todo de Diferen as Finitas   uma t cnica para obten o de solu es aproximadas de equa es diferenciais, sendo que, para se obter tais solu es, deve-se discretizar o dom nio. A ideia fundamental que est  por tr s dos m todos num ricos   a de discretiza o do cont nuo, ou seja, dado um dom nio cont nuo, deve-se discretiz -lo de forma a encontrar solu es aproximadas da solu o anal tica, quando se as t m.

No M todo de Diferen as Finitas, uma vez discretizado o dom nio, deve-se substituir as derivadas presentes na equa o diferencial por aproxima es que envolvam apenas valores num ricos da fun o (Cuminato e J nior, 2013).

Para discretizar um problema,   preciso gerar uma malha com uma quantidade de n s no dom nio. Entretanto,   preciso ter cautela na escolha dessa quantidade de n s, pois quanto mais n s tiver a malha, espera-se que mais pr xima da solu o anal tica a solu o num rica vai estar, por m o custo computacional pode ser muito alto, levando muito tempo ou at  mesmo n o resolvendo o problema.

No que se refere ao M todo das Diferen as Finitas existem duas abordagens que podem ser exploradas para solucionar o problema. Uma delas   a abordagem expl cita e a outra impl cita, sendo essa segunda mais trabalhosa de se desenvolver. A formula o expl cita, embora mais simples de se resolver, possui um crit rio que est  associado   estabilidade e converg ncia da fun o, enquanto a impl cita exige um recurso a mais na obten o da solu o do problema, devido a um sistema de equa es que deve ser resolvido utilizando algum m todo num rico. Apesar disso, a formula o a ser utilizada depende da caracter stica de cada problema.

3.1 Aplica o da Formula o Expl cita ao Problema Proposto

Tomando como refer cia a derivada em rela o ao tempo, a qual   de primeira ordem, e utilizando a s rie de Taylor para realizar uma aproxima o por diferen as avan adas, cujo erro   dado por $-\frac{h}{2!} f''(\xi)$, obt m-se a Eq. (6).

$$\frac{\partial T(x, t)}{\partial t} \simeq \frac{T(x, t+k) - T(x, t)}{k} = \frac{T_{i,j+1} - T_{i,j}}{k} \quad (6)$$

Ao usar a f rmula centrada para o espa o, cujo erro   dado por $-\frac{h^2}{3!} f'''(\xi)$, e tomado como refer cia o instante de tempo $j + 1$, obt m-se a Eq. (7).

$$\frac{\partial^2 T(x, t)}{\partial x^2} \simeq \frac{T(x+h, t) - 2T(x, t) + T(x-h, t)}{h^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{h^2} \quad (7)$$

Considerando $\beta = \frac{\rho c_p}{\lambda}$ e aplicando os resultados obtidos nas Eqs. (6) e (7), tem-se a express o dada pela Eq. (8).

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{h^2} - \beta \frac{T_{i,j+1} - T_{i,j}}{k} = 0 \quad (8)$$

Tomando $u_{i,j}$ como aproxima o de $T_{i,j}$, a Eq. (8) pode ser reescrita conforme a Eq. (9).

$$\frac{k}{\beta h^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) - u_{i,j+1} + u_{i,j} = 0 \quad (9)$$

Ao isolar $u_{i,j+1}$, obt m-se a Eq. (10).

$$u_{i,j+1} = u_{i,j} + \frac{k}{\beta h^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \quad (10)$$

$$u_{i,j+1} = \frac{k}{\beta h^2} u_{i-1,j} + \left(1 - \frac{2k}{\beta h^2}\right) u_{i,j} + \frac{k}{\beta h^2} u_{i+1,j} \quad (10)$$

$\forall i = 1, \dots, n-1$ e $j = 0, 1, \dots, m-1$.

Para as condi es de contorno dadas pelas Eqs. (2) e (3),  utilizada uma f rmula centrada, com erro de truncamento de $-\frac{h^2}{3!} f'''(\xi)$.

Em $x = 0$ ($i = 0$), chega-se a Eq. (11).

$$-\lambda \frac{\partial T(x,t)}{\partial x} = \phi(t) \implies \frac{\partial T(x,t)}{\partial x} = -\frac{\phi(t)}{\lambda} \implies \frac{u_{1,j} - u_{-1,j}}{2h} = -\frac{\phi(t)}{\lambda} \quad (11)$$

Logo, conclui-se $u_{-1,j}$ conforme Eq. (12).

$$u_{-1,j} = u_{1,j} + \frac{2h}{\lambda} \phi(t) \quad (12)$$

Substituindo a Eq. (12) na Eq. (10) para $i = 0$, obt m-se a Eq. (13).

$$u_{0,j+1} = \left(1 - \frac{2k}{\beta h^2}\right) u_{0,j} + \frac{2k}{\beta h^2} u_{1,j} + \frac{2k}{\beta h \lambda} \phi(t) \quad (13)$$

Em $x = L$ obt m-se como resultado a Eq. (14).

$$\frac{\partial T(x,t)}{\partial x} = 0 \implies \frac{u_{n+1,j} - u_{n-1,j}}{2h} = 0 \quad (14)$$

Assim, conclui-se $u_{n+1,j}$ conforme a Eq. (15).

$$u_{n+1,j} = u_{n-1,j} \quad (15)$$

Substituindo a Eq. (15) na Eq. (10) para $i = n$, obt m-se a Eq. (16).

$$u_{n,j+1} = \frac{2k}{\beta h^2} u_{n-1,j} + \left(1 - \frac{2k}{\beta h^2}\right) u_{n,j} \quad (16)$$

A formula o expl cita  condicionalmente est vel, o que significa que precisa respeitar a condi o de estabilidade dada por $r = \alpha \frac{k}{h^2} \leq \frac{1}{2}$, onde $\alpha = \frac{\lambda}{\rho c_p}$  a difusividade t rmica (m^2/s).

3.2 Aplicação da Formulação Implícita ao Problema Proposto

Considerando as Eqs. (1) - (4) que modelam o problema de transferência de calor (Carollo, 2010) e considerando ainda $u_{i,j}$ uma aproximação de $T_{i,j}$, aplicando diferenças centradas do lado esquerdo da Eq. (1) e diferenças progressivas do lado direito obtém-se a Eq. (17).

$$\frac{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}}{h^2} = \beta \frac{u_{i,j+1} - u_{i,j}}{k} \quad (17)$$

Realizando algumas operações algébricas, a Eq. (17) é expressa conforme a Eq. (18).

$$\frac{k}{\beta h^2} u_{i-1,j+1} - (1 + \frac{2k}{\beta h^2}) u_{i,j+1} + \frac{k}{\beta h^2} u_{i+1,j+1} = -u_{i,j} \quad (18)$$

Ressalta-se que, na Eq. (18), apenas $u_{i,j}$ é um valor conhecido. Como já foi dito, a abordagem implícita possui um grau maior de dificuldade para resolvê-la. A dificuldade está justamente no fato de resolver o sistema que é gerado com a Eq. (18), juntamente com as condições de fronteira, tomando $i = 1, 2, \dots, n-1$ e $j = 0, 1, \dots, m-1$.

Pelas condições de contorno, em $x = 0$, no método implícito, $i = 0$, obtém-se a Eq. (19).

$$-\lambda \frac{\partial T(x,t)}{\partial x} = \phi(t) \implies \frac{u_{1,j+1} - u_{-1,j+1}}{2h} = -\frac{\phi(t)}{\lambda} \quad (19)$$

Logo, a Eq. (19) pode ser expressada de acordo com a Eq. (20).

$$u_{-1,j+1} = u_{1,j+1} + \frac{2h}{\lambda} \phi(t) \quad (20)$$

Substituindo a Eq. (20) na Eq. (18) para $i = 0$, obtém-se a Eq. (21).

$$-(1 + \frac{2k}{\beta h^2}) u_{0,j+1} + \frac{2k}{\beta h^2} u_{1,j+1} = -u_{0,j} - \frac{2k}{\beta \lambda h} \phi(t) \quad (21)$$

Pelas condições de contorno, em $x = L$, no método implícito, $i = n$, obtém-se a Eq. (22).

$$\frac{\partial T(x,t)}{\partial x} = 0 \implies \frac{u_{n+1,j+1} - u_{n-1,j+1}}{2h} = 0 \quad (22)$$

Assim, obtém-se a Eq. (23).

$$u_{n+1,j+1} = u_{n-1,j+1} \quad (23)$$

Substituindo a Eq. (23) na Eq. (18), para $i = n$, chega-se a Eq. (24).

$$\frac{2k}{\beta h^2} u_{n-1,j+1} - (1 + \frac{2k}{\beta h^2}) u_{n,j+1} = -u_{n,j} \quad (24)$$

4. RESULTADOS DO PROBLEMA DIRETO DE TRANSFER NCIA DE CALOR

Os par metros configurados no modelo matem tico para a aplic o o do m todo das diferen as finitas utilizado na solu o o e obten o o dos resultados foram a condutividade t rmica λ e a capacidade volum trica ρc_p , valendo 14.611 W/(mK) e $3.9073 \times 10^6\text{ Ws/(m}^3\text{K)}$, respectivamente (Carollo, 2010).

4.1 An lise da Varia o das Malhas Espacial e Temporal

O problema direto foi resolvido dividindo a malha espacial em 5, 10 e 15 n os, ou seja, $\Delta x = 0,00272$, $\Delta x = 0,001209$ e $\Delta x = 0,000777$, respectivamente. Para cada malha espacial, a malha temporal foi dividida em 16, 32, 80, 160, 320, 800 e 1600 n os.

Para a malha espacial com 5 n os, $\Delta x = 0,00272$, obteve-se os seguintes valores para a malha temporal (Tabela 1), onde r  e o crit rio de estabilidade para a formula o expl cita. J  para a malha espacial com 10 n os, $\Delta x = 0,001209$, obteve-se os valores para a malha temporal apresentados na Tabela 2. Por fim, para a malha espacial com 15 n os, $\Delta x = 0,000777$, obteve-se os valores para a malha temporal apresentados na Tabela 3.

Tabela 1- Discretiza es obtidas tomando como base $\Delta x = 0,00272$ utilizando a formula o expl cita.

Tempo total do experimento (s)	Δt (s)	N�os (-)	r (-)
160	0,1	1600	0,050544
160	0,2	800	0,101087
160	0,5	320	0,252718
160	1	160	0,505435
160	2	80	1,010870
160	5	32	2,527175
160	10	16	5,054351

Tabela 2- Discretiza es obtidas tomando como base $\Delta x = 0,001209$ utilizando a formula o expl cita.

Tempo total do experimento (s)	Δt (s)	N�os (-)	r (-)
160	0,1	1600	0,255877
160	0,2	800	0,511753
160	0,5	320	1,279383
160	1	160	2,558765
160	2	80	5,117530
160	5	32	12,793830
160	10	16	25,587650

Os resultados num ricos obtidos tendo como base as formula es expl cita e impl cita, as quais foram implementadas na linguagem de program o C, foram satisfat rios, respeitando o crit rio de estabilidade da formula o expl cita, uma vez que   condicionalmente convergente (Lisboa et al., 2017; Cuminato e J nior, 2013).

Na Tabela 1, verifica-se que, de acordo com os valores de r , o m todo apresenta-se est vel para as malhas temporais 1600, 800 e 320, ou seja, $\Delta t = 0,1$, $\Delta t = 0,2$ e $\Delta t = 0,5$ respectivamente (Fig. 2). Tomando como base a Tabela 2,   poss vel verificar que, de acordo com os

Tabela 3- Discretizações obtidas tomando como base $\Delta x = 0,000777$ utilizando a formulação explícita.

Tempo total do experimento (s)	Δt (s)	Nós (-)	r (-)
160	0,1	1600	0,619158
160	0,2	800	1,238316
160	0,5	320	3,095790
160	1	160	6,191580
160	2	80	12,383160
160	5	32	30,957900
160	10	16	61,915800

valores de r , o método só é estável para malha temporal 1600, ou seja, $\Delta t = 0,1$. Nesse caso, o resultado é satisfatório ao se comparar com os dados obtidos experimentalmente por Carollo (2010), conforme mostrado na Fig. 3. Ainda na formulação explícita, para $\Delta x = 0,000777$, na Tabela 3, é mostrado que a formulação não atende ao critério de estabilidade para todas as malhas temporais escolhidas (resultados gráficos não mostrados aqui).

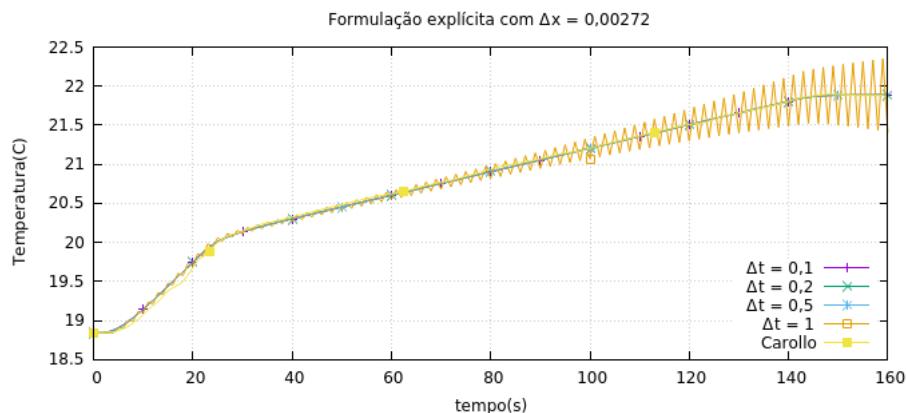


Figura 2- Resultados da formulação explícita com $\Delta x = 0,00272$ e diferentes valores para Δt .
Fonte: O autor, 2019.

Por outro lado, como a formulação implícita é incondicionalmente estável, espera-se que a última coluna das Tabelas 1 - 3, não interfira nos resultados obtidos pelo algoritmo, no que se refere às oscilações numéricas. Na formulação implícita a matriz é resolvida pelo método Gauss-Seidel, cujo critério de parada adotado é a diferença entre as soluções de duas iterações consecutivas menor que 10^{-10} .

Para malha espacial com 5 nós, $\Delta x = 0,00272$, as malhas temporais com 16 e 32 nós, que equivalem a Δt igual 10 e 5, respectivamente, apresentaram comportamento da temperatura distantes dos dados experimentais. Entretanto as outras malhas possibilitaram um resultado compatível com o experimento de Carollo (2010), conforme mostrado na Fig. 4. Para $\Delta x = 0,001209$, o algoritmo encontrou resultados satisfatórios para as malhas temporais 320, 800 e 1600 relacionadas a Δt iguais a 0,5, 0,2 e 0,1, respectivamente, conforme mostrado na Fig. 5. Já para $\Delta x = 0,000777$, apenas a malha temporal com 1600 nós, $\Delta t = 0,1$, obteve resultado próximo ao experimento de Carollo (2010) (Fig. 6). As demais malhas não reproduziram resultados próximos aos dados experimentais.

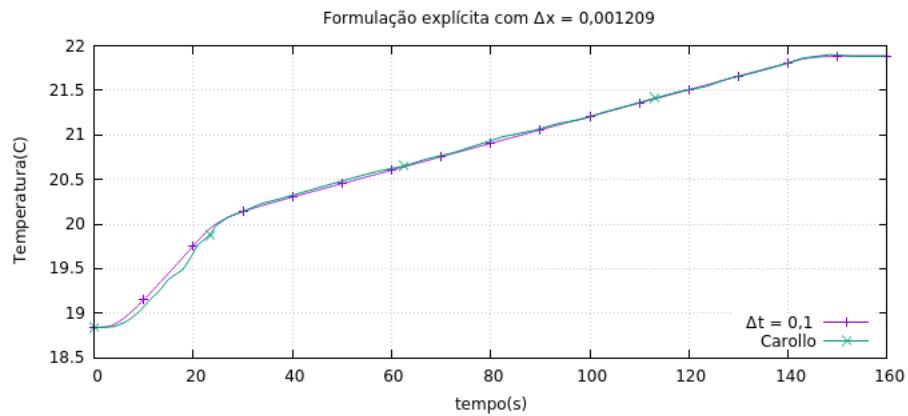


Figura 3- Resultados da formulação explícita com $\Delta x = 0,001209$ e $\Delta t = 0,1$.
Fonte: O autor, 2019.

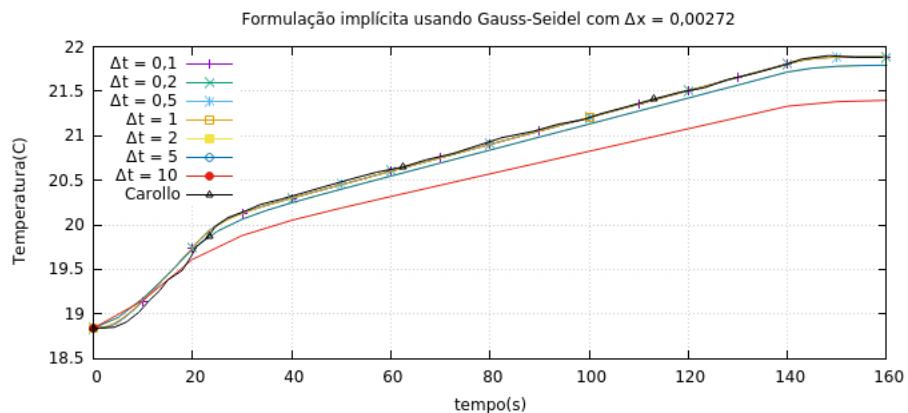


Figura 4- Resultados da formulação implícita com $\Delta x = 0,00272$ e diferentes valores para Δt .
Fonte: O autor, 2019.

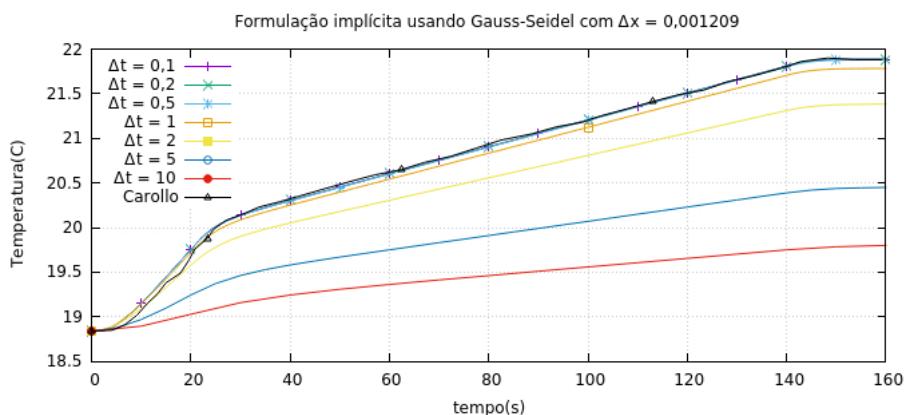


Figura 5- Resultados da formulação implícita com $\Delta x = 0,001209$ e diferentes valores para Δt .
Fonte: O autor, 2019.

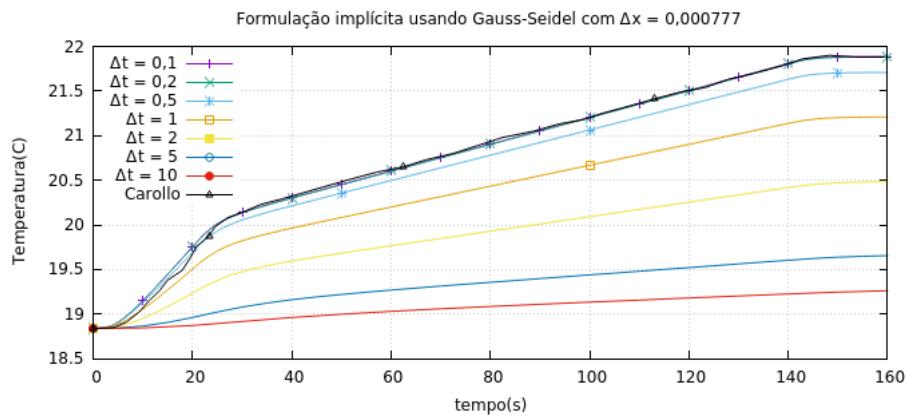


Figura 6- Resultados da formulação implícita com $\Delta x = 0,000777$ e diferentes valores para Δt .
Fonte: O autor, 2019.

Em suma, apesar da malha espacial com 10 nós e malha temporal com 1600 nós gerar resultados muito próximos aos dados experimentais, tanto na formulação explícita como na formulação implícita, a malha com 800 nós na formulação implícita, ou seja, $\Delta x = 0,001209$ e $\Delta t = 0,2$, também obtém resultados muito satisfatórios, com menor custo computacional. A Fig. 7 apresenta o erro absoluto obtido entre os dados experimentais e temperaturas numéricas utilizando, respectivamente, as formulações explícita e implícita para a referida malha.

5. CONCLUSÕES

Em estudos de equações diferenciais existe o que se chama de solução analítica e solução aproximada. Na solução analítica o domínio do problema é contínuo, enquanto para soluções aproximadas é preciso discretizar o domínio. Essas soluções obtidas através da discretização do domínio geram uma solução aproximada, onde deve-se analisar se a mesma satisfaz ou não o problema. Tanto na formulação explícita quanto na implícita, obteve-se resultados numéricos compatíveis e satisfatórios com os dados experimentais obtidos por Carollo (2010), no entanto, devido ao critério de estabilidade da formulação explícita, a formulação implícita apresentou melhores resultados considerando a discretização de 10 nós para malha espacial e 800 nós para malha temporal.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

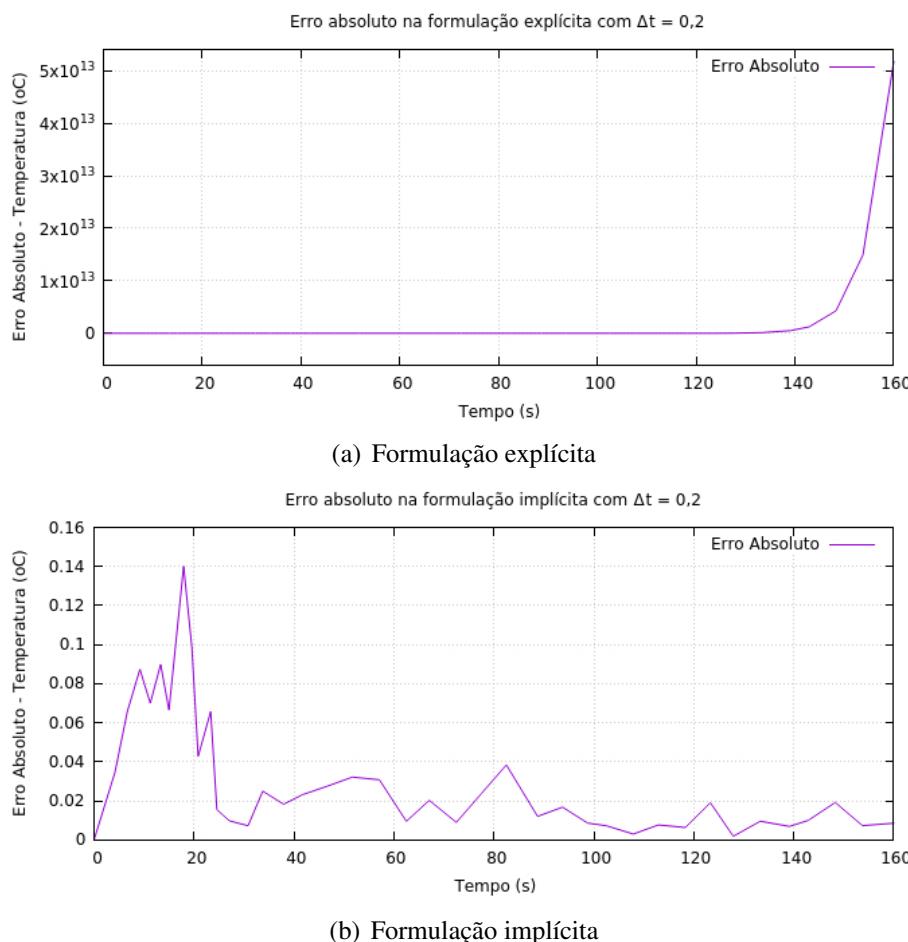


Figura 7- Erro absoluto nas formula es expl cita e impl cita com $\Delta x = 0,001209$ e $\Delta x = 0,2$.

Refer ncias

- [1] Carollo, L. F. S. 2010. *Predic o de Pontos Cr ticos de Misturas Termodin micas*, Dissertaci o de Mestrado, UNIFEI, Itajub .
- [2] Cuminato, J. A. e J nior, M. M., 2013. *Discretiza o de equa es diferenciais parciais: T cnicas de diferen as finitas*, 1^a ed., SBM, Brasil.
- [3] Burden, R. L., Faires, D. J. and Burden, A. M. 2016. *Numerical Analysis*, 10^a ed., CENGAGE LEARNING, Bonston.
- [4] Pereira, A. J., Lisboa, N. H. e Filho, J. H. D. 2017. *An\'alise de estabilidade do m todo expl cito para discretiza o de equa es diferenciais parab licas por meio de diferen as finitas*, ed. Inicia o Cient fica.
- [5] Jen, L. C. and Rissetti, G., Ferreira, A. G., Yamagushi, A. H., Coelho, L. F. 2006. *Aplicativo visual para problemas de transfer ncia de calor*, Exacta, Volume 4, pp. 385-390.

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

AVALIAÇÃO DOS MÉTODOS DE OTIMIZAÇÃO LUUS-JAAKOLA E ALGORITMO DE COLISÃO DE PARTÍCULAS EM UM PROBLEMA DE TRANSFERÊNCIA DE CALOR

Alex Tavares Silva¹ - altasilva@gmail.com

Wagner Rambaldi Telles^{1,2} - wtelles@id.uff.br

Gustavo Silva Semaan^{1,2} - gustavosemaan@id.uff.br

¹Universidade Federal Fluminense, EEIMVR - Volta Redonda, RJ, Brasil

²Universidade Federal Fluminense, INFES - Santo Antônio de Pádua, RJ, Brasil

Resumo. Este trabalho apresenta uma avaliação de modificações nos métodos Luus-Jaakola (LJ) e Algoritmo de Colisão de Partículas (PCA) na resolução de um problema inverso de transferência de calor, cujo objetivo é encontrar a melhor configuração dos parâmetros de interesse, de maneira que os resultados numéricos se aproximem dos dados experimentais obtidos na literatura. O experimento consiste em aquecer na parte superior uma placa de Aço Inox modelo AISI 304 de comprimento L , compreendida entre um aquecedor e um isolante térmico nas laterais e na parte inferior, donde encontra-se localizado um termopar com o intuito de se aferir a temperatura no local. Os resultados obtidos apresentam-se satisfatórios e com uma redução do número de avaliações da função objetivo (NAF) necessários para resolver o problema.

Keywords: Transferência de Calor, Problema Inverso, Otimização, Luus-Jaakola, Algoritmo de Colisão de Partículas

1. INTRODUÇÃO

Métodos de otimização são importantes, entre muitos casos, quando se deseja estimar parâmetros desconhecidos em um determinado problema, ou seja, quando se conhece o efeito mas a causa é desconhecida. Esse tipo de abordagem, que parte da solução final de um problema para se chegar aos parâmetros iniciais, é conhecida como abordagem inversa.

De acordo com Cezaro (2010), problemas que envolvem a determinação da causa a partir de um efeito medido ou observado, possuem uma quantidade grande de aplicações em várias áreas da ciência e, a esse campo de estudos da matemática aplicada, denomina-se *Problemas Inversos*. Além disso, segundo Silva Neto et al. (2016), problemas inversos são utilizados para obtenção de estimativa de parâmetros ou estimativa de funções.

Neste trabalho, é abordado um problema inverso, onde busca-se a estimativa de dois parâmetros em um problema de transferência de calor, a condutividade térmica (λ) e a capacidade de calor volumétrica (ρc_p), utilizando os métodos de otimização estocástica Luus-Jaakola e Algoritmo de Colisão de Partículas. Além de encontrar os parâmetros desejados, busca-se melhorar os resultados, reduzindo a quantidade de consultas à função objetivo e, consequentemente, o custo computacional, através de modificações realizadas nas versões clássicas dos referidos métodos.

2. DESCRIÇÃO DO PROBLEMA DE TRANSFERÊNCIA DE CALOR

O problema proposto para este trabalho é baseado no experimento realizado por Carollo (2010), o qual trata-se de uma placa que está compreendida entre um aquecedor e um isolante térmico de forma que se deseja obter a temperatura final da mesma, conforme ilustra a Fig. 1, a qual é constituída de um material de aço inox modelo AISI 304, de comprimento L , onde no final encontra-se um termopar, com o objetivo de aferir tal temperatura.

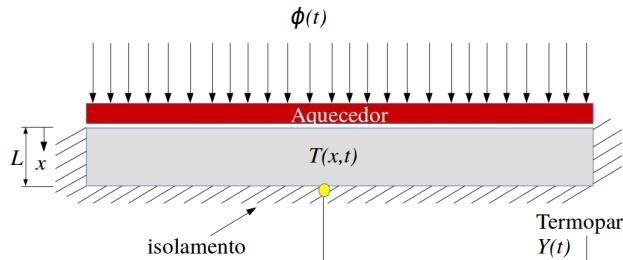


Figura 1- Representação esquemática do problema proposto.

Fonte: Adaptado de Carollo (2010).

Já a modelagem matemática desse problema é representada pela Eq. (1), com condições de contorno dadas pelas Eqs. (2)-(3) e condição inicial dada pela Eq. (4).

$$\frac{\partial^2 T(x, t)}{\partial x^2} = \frac{\rho c_p}{\lambda} \frac{\partial T(x, t)}{\partial t}, \text{ em } 0 < x < L \text{ e } t > 0 \quad (1)$$

$$-\lambda \frac{\partial T(x, t)}{\partial x} = \phi(t), \text{ em } x = 0 \text{ e } t > 0 \quad (2)$$

$$\frac{\partial T(x, t)}{\partial x} = 0, \text{ em } x = L \text{ e } t > 0 \quad (3)$$

$$T(x, t) = T_0, \text{ em } 0 \leq x \leq L \text{ e } t = 0 \quad (4)$$

onde x é a coordenada cartesiana (m), t é o tempo (s), λ é a condutividade térmica (W/mK), ρc_p é a capacidade de calor volumétrica (Ws/m^3K), $\phi(t)$ é o fluxo de calor (W/m^2) imposto no instante de tempo t e T_0 é a temperatura inicial do corpo ($^{\circ}C$).

Problemas envolvendo o aço inox 304 são comumente encontrados na literatura, como em Assael e Gialou (2003) e BlackWell et al. (2000).

Para a solução numérica desse problema é utilizado o Método das Diferenças Finitas com formulação implícita, onde a solução do sistema de equações resultante é dada pelo método de Gauss-Seidel. Já a estimativa dos parâmetros de interesse (λ e ρc_p) é obtida com base nos métodos estocásticos Luus-Jaakola e Algoritmo de Colisão de Partículas.

3. MÉTODO LUUS-JAAKOLA

O método Luus-Jaakola (LJ) é um procedimento estocástico de busca aleatória, onde, segundo Luus e Jaakola (1973), o domínio de busca inicial é o domínio de cada variável do problema e, a cada iteração, o tamanho do espaço de busca é reduzido.

De acordo com Telles (2014), nesse método a principal ideia é considerar uma região de busca ampla englobando possíveis valores das variáveis e gerar soluções aleatórias, enquanto essa região vai contraindo cada vez mais ao longo das iterações.

Inicialmente, deve-se determinar o fator de redução do intervalo de busca, dado pelo parâmetro ϵ ($0 < \epsilon < 1$), que está relacionado com o percentual em que o intervalo é contraído. Além desse parâmetro, há ainda o número de *loops* externos do algoritmo, n_{out} , o qual determina quantas vezes essa contração irá acontecer e o número de estimativas geradas a cada *loop*, dado pelo número de *loops* internos, n_{int} .

Posteriormente, deve-se determinar os intervalos de busca para as variáveis que se deseja estimar, os quais dependem do problema a ser tratado, bem como a amplitude dos referidos intervalos, denotada por $\mathbf{r}^{(k)} = (r_1^{(k)}, \dots, r_n^{(k)})$, onde n é o número de variáveis, ou seja, o número da dimensão do problema.

Configurados todos os parâmetros iniciais, executa-se o algoritmo e o mesmo gera uma solução inicial, $\mathbf{x}_0^{(0)}$, tida como melhor solução para o problema até o momento, $\mathbf{x}^* = \mathbf{x}_0^{(0)}$, obtida de forma aleatória. A partir daí, são geradas novas estimativas candidatas à solução ótima do problema, sempre definidas pelo processo iterativo dos *loops* externos e internos n_{out} e n_{int} , dada pela fórmula de recorrência: $\mathbf{x}_j^{(k)} = \mathbf{x}^* + R\mathbf{r}^{(k-1)}$, $j = 1, \dots, n_{int}$ e $k = 1, \dots, n_{out}$, sendo R é uma matriz diagonal de números aleatórios entre -0,5 e 0,5 gerada toda vez que uma nova estimativa é calculada. Caso o vetor solução $\mathbf{x}_j^{(k)}$ seja melhor que a solução \mathbf{x}^* , a nova estimativa armazenada em \mathbf{x}^* receberá o valor de $\mathbf{x}_j^{(k)}$, ou seja, $\mathbf{x}^* = \mathbf{x}_j^{(k)}$. Ao final de cada *loop* interno, a amplitude do intervalo de busca é recalculada (reduzida) por $\mathbf{r}^{(k)} = (1 - \epsilon)\mathbf{r}^{(k-1)}$.

Após a finalização dos *loops* externos, ou seja, quando todas as iterações forem executadas, obtém-se a solução do problema inverso, que resulta no maior valor da função objetivo (caso o problema seja de maximização) ou no menor valor da função objetivo (caso o problema seja de minimização).

4. ALGORITMO DE COLISÃO DE PARTÍCULAS

O Algoritmo de Colisão de Partículas (do inglês *Particle Collision Algorithm*, PCA) foi desenvolvido em 2005 por Wagner F. Sacco e assemelha-se ao algoritmo de Recozimento Simulado (do inglês *Simulated Annealing*, SA). O PCA se associa com fenômenos físicos de absorção (*absorption*) e espalhamento (*scattering*) que ocorrem em reatores nucleares, mais precisamente, no processo de colisão das partículas nucleares, onde a partícula é espalhada por um núcleo-alvo e absorvida pelo núcleo de destino (Sacco e Oliveira, 2005).

Conforme Sacco e Oliveira (2005), o PCA possui uma quantidade menor de parâmetros a serem configurados, em comparação a algoritmos clássicos como Algoritmos Genéticos, por exemplo, pois a maioria dos parâmetros são definidos por rotinas aleatórias.

Assim como o Luus-Jaakola, no PCA se faz necessário definir os limites inferior (**min**) e superior (**max**) dos intervalos de busca para as variáveis que se deseja estimar, de acordo com o problema a ser tratado, além do número de vezes (n_{PCA}) em que ocorrerá o processo iterativo,

onde novas estimativas são geradas e avaliadas de acordo com a função objetivo (Telles, 2014). Posteriormente, de acordo com Lobato et al. (2010) e Luz et al. (2008), o PCA é inicializado com a seleção (randômica ou não) de um projeto inicial denominado **OldConfig**. Essa configuração também é considerada a melhor solução do problema até o momento, ou seja, **BestConfig** = **OldConfig**.

Em seguida, esse projeto inicial é modificado de forma estocástica por uma rotina denominada *Perturbation()* que, por sua vez, permite a obtenção de uma nova solução candidata a ótimo denominada **NewConfig**, utilizando a fórmula $\text{NewConfig} = \text{OldConfig} + \{R(\max - \text{OldConfig}) + (I-R)(\text{OldConfig} - \min)\}$, onde R é uma matriz diagonal de números randômicos entre 0 e 1 e I é a matriz Identidade. As qualidades das duas configurações são comparadas. Considerando f a função objetivo, n o número de variáveis do problema (dimensão) e **BestConfig** o vetor contendo a melhor solução para um problema de mininização, o algoritmo verifica se $f(\text{NewConfig}) < f(\text{OldConfig})$ e, caso seja verdade, então o vetor **OldConfig** recebe os valores do vetor **NewConfig**; além disso, se $f(\text{NewConfig}) < f(\text{BestConfig})$ então o vetor **BestConfig** também recebe os valores do vetor **NewConfig**. Na sequência, uma exploração é feita na região utilizando a rotina *SmallPerturbation()* por meio da fórmula $\text{NewConfig} = \frac{1}{2}\{\text{OldConfig} + [\text{Lower} + R(\text{Upper} - \text{Lower})]\}$, onde $\text{Upper} = \text{OldConfig} + |(0, 2R \text{OldConfig})|$ e $\text{Lower} = \text{OldConfig} - |(0, 2R \text{OldConfig})|$, apresentando valores aleatórios, sendo o mais próximo da solução do problema selecionado. O número de vezes em que novas soluções serão geradas em torno da melhor solução é determinado pelo parâmetro n_{EXP} .

No entanto, se a nova solução gerada pela rotina *Perturbation()* não for melhor que anterior, ou seja, se $f(\text{NewConfig}) > f(\text{OldConfig})$, então verifica-se a probabilidade da mesma ser aceita levando em consideração sua aptidão. Para isso, gera-se um número aleatório entre 0 e 1, compara-se com o valor de $p_{scattering}$, dado por $p_{scattering} = 1 - f(\text{BestConfig})/f(\text{NewConfig})$ e, sendo esse número maior que $p_{scattering}$, a solução é aceita, processo esse realizado pela rotina denominada *Scattering()*. Caso, ainda assim, a nova estimativa não seja aceita, um espalhamento é realizado e uma solução aleatória em outro local do domínio é selecionada, realizando novamente o processo de perturbação, seguido de exploração ou espalhamento. Esse processo se repete até que a solução seja encontrada ou que o número de *loops* externos n_{PCA} seja alcançado.

5. MODIFICAÇÃO NOS MÉTODOS LUUS-JAAKOLA E ALGORITMO DE COLISÃO DE PARTÍCULAS

Nessa seção é apresentada uma proposta de modificação para os métodos LJ e PCA. Como na versão clássica do método Luus-Jaakola, descrita na Seção 3., um novo vetor $x_j^{(k)}$ é gerado e compara-se $f(x_j^{(k)})$ com $f(x^*)$, muitas vezes descarta-se todos os elementos de $x_j^{(k)}$ quando $f(x_j^{(k)}) > f(x^*)$. No entanto, pode-se ocorrer que o vetor $x_j^{(k)}$ contenha algum componente melhor do que seu correspondente em x^* . Como no problema de transferência de calor abordado nesse trabalho busca-se estimar dois parâmetros (λ e ρc_p), no desenvolvimento a seguir, são considerados os vetores contendo apenas duas componentes. Sendo assim, nessa primeira modificação, denominada LJ-M1, é proposta a combinação entre os elementos $x_j^{(k)}$ e x^* .

A cada nova estimativa dos parâmetros, é feita uma combinação com a melhor estimativa, ou seja, se a melhor estimativa for $x^* = (x_1^*, x_2^*)$ e a nova for $x_j^{(k)} = (x_1, x_2)$, então, com a combinação tem-se 3 estimativas a serem avaliadas, incluindo a que acabou de ser gerada:

(x_1, x_2) , (x_1^*, x_2) e (x_1, x_2^*) . Consequentemente, o número de *loops* internos corresponde a um terço da versão original do algoritmo Luus-Jaakola.

Como exemplo, considere o problema de minimização da função $f(x_1, x_2) = x_1^2 + x_2^2$, cuja solução é $(0, 0)$. O método se inicia calculando o vetor de estimativa inicial, $x_0^{(0)} = (1, 11)$. Logo a solução ótima, x^* , nesse momento, é igual a $(1, 11)$, pois é a única solução conhecida. No processo iterativo, para $k = 1$ e $j = 1$, uma nova estimativa é gerada e, suponha que seja $x_1^{(1)} = (0, 12)$. Observe que apesar do primeiro parâmetro ser 0, que é a solução do problema para o respectivo parâmetro, $x_1^{(1)} = (0, 12)$ não é melhor do que $x^* = (1, 11)$, pois $f(1, 11) < f(0, 12)$. As versões clássicas nesse momento descartariam a estimativa $x_1^{(1)}$, porém, com a técnica de combinação de estimativas, mais dois vetores, $(1, 12)$ e $(0, 11)$, são analisados. Embora $(1, 12)$ não seja melhor do que $x^* = (1, 11)$, tem-se que $(0, 11)$ é, logo a nova solução ótima x^* é igual a $(0, 11)$. Nesse momento, um dos parâmetros já foi encontrado, faltando apenas encontrar o segundo parâmetro.

Na sequência, considere agora que a nova estimativa, na iteração $k = 1$ e $j = 2$, seja $x_2^{(1)} = (5, 0)$. Novamente, tem-se a solução para um dos parâmetros, nesse caso o segundo. Fazendo a combinação entre os elementos das estimativas $(5, 0)$ e $(0, 11)$, novas combinações são comparadas, $(5, 11)$ e $(0, 0)$, onde esta última é a nova solução ótima, além de ser a solução do problema. Observe que na iteração $k = 1$ e $j = 2$, nas versões clássicas, a solução ótima seria $x^* = (5, 0)$, pois a estimativa $(0, 11)$, seria descartada por não ser melhor do que a estimativa gerada $(5, 0)$, ou seja, $f(0, 11) > f(5, 0)$, mesmo contendo a solução do primeiro parâmetro e, assim, o método na versão clássica continuaria o processo de busca para a solução do problema, pois o primeiro parâmetro ainda não foi encontrado. Esse processo de combinação faz com que o método encontre a solução do problema com menos iterações.

A mesma idéia de combinação implementada no método LJ, foi implementada no PCA, denominada PCA-M1. No entanto, as três avaliações da função objetivo são realizadas durante o processo de exploração, onde, após a geração de uma nova estimativa utilizando a rotina *SmallPerturbation()* por meio da fórmula $\text{NewConfig} = \frac{1}{2}\{\text{OldConfig} + [\text{Lower} + R(\text{Upper} - \text{Lower})]\}$, é feita a combinação da mesma com a melhor estimativa até o momento armazenada em $\text{BestConfig} = (\text{BestConfig}_1, \text{BestConfig}_2)$, ou seja, além de $\text{NewConfig} = (\text{NewConfig}_1, \text{NewConfig}_2)$, são avaliadas as estimativas de forma combinada $(\text{BestConfig}_1, \text{NewConfig}_2)$ e $(\text{NewConfig}_1, \text{BestConfig}_2)$. Com isso, o processo de exploração é redimensionado a um terço do valor original do parâmetro n_{EXP} .

6. RESULTADOS DOS MÉTODOS LUUS-JAAKOLA E ALGORITMO DE COLISÃO DE PARTÍCULAS E SUAS RESPECTIVAS MODIFICAÇÕES

O problema inverso a ser resolvido neste trabalho consiste em estimar os parâmetros λ e ρc_p do problema de transferência de calor, presentes na Eq. (1), utilizando os métodos de otimização LJ e PCA, assim como suas modificações. Para isso, busca-se minimizar a função objetivo dada pela soma dos resíduos ao quadrado entre os dados experimentais e numéricos, definida pela Eq. (5).

$$f(\lambda, \rho c_p) = \sqrt{\sum_{i=1}^{nd} (T_{exp_i} - T_{num_i}(\lambda, \rho c_p))^2} \quad (5)$$

onde nd é o número de dados experimentais, T_{exp_i} são as temperaturas experimentais obtidas no trabalho de Carollo (2010) e T_{num_i} são as temperaturas obtidas através da solução numérica da Eq. (1), com condições de contorno e inicial dadas pelas Eqs. (2)-(4), utilizando o Método das Diferenças Finitas com formulação implícita, adotando a malha espacial com 10 nós ($\Delta x = 0,001209$) e malha temporal com 800 nós ($\Delta t = 0,2$).

Os experimentos computacionais foram realizados em um computador com processador Intel Core i5-5200U 2.2GHz, 8GB de memória RAM, sistema operacional Ubuntu GNU/Linux, sendo utilizada a linguagem de programação C e Gnuplot para geração de gráficos.

Para a obtenção dos resultados, os métodos foram configurados para realizar 100 execuções e, em cada uma dessas execuções, foi contabilizado o número de avaliações da função objetivo (NAF), dada pela Eq. (5), necessários para que os algoritmos encontrem os parâmetros de interesse (λ e ρc_p) de acordo com um critério de parada estabelecido *a priori*. Logo, os dados estatísticos são calculados levando em consideração esse total de execuções, os quais são: melhor resultado (menor número de avaliações da função objetivo), pior resultado (maior número de avaliações da função objetivo), média, mediana e desvio padrão do número de avaliações da função objetivo, da condutividade térmica, λ , e da capacidade volumétrica, ρc_p , conforme pode ser observado ao longo dessa Seção. Foram utilizadas as seguintes configurações para os métodos LJ e PCA:

Luus-Jaakola (LJ):

- número de iterações externas: $n_{out} = 100$;
- número de iterações internas: $n_{int} = 50$;
- fator de contração do intervalo de busca: $\epsilon = 0,2$;
- critério de parada estabelecido: $f(\lambda, \rho c_p) < 3 \times 10^{-2}$ ou NAF > 20000.

Algoritmo de Colisão de Partículas (PCA):

- número de vezes que o processo será executado: $n_{pca} = 100$;
- número de explorações em torno de uma solução: $n_{exp} = 50$;
- critério de parada estabelecido: $f(\lambda, \rho c_p) < 3 \times 10^{-2}$ ou NAF > 20000.

Na Tabela 1 são mostrados os resultados obtidos pelo método Luus-Jaakola, versão clássica, onde pode-se observar que, no pior resultado, o NAF foi de 1232 e, o melhor resultado foi obtido com NAF igual a 227. Isso significa que a tolerância pré-fixada no algoritmo fez o mesmo parar sua execução quando o valor da função objetivo se tornou menor do que 3×10^{-2} . A média do NAF foi igual a 681, enquanto a mediana foi de 702.

Quando alguns resultados variam muito, a média pode não ser um parâmetro tão interessante para se avaliar, por isso calcula-se a mediana dos resultados. Por outro lado, o desvio padrão é uma medida de dispersão que mensura o quanto um conjunto de dados se desvia da média, obtendo um NAF igual a 148. Cabe ressaltar que o desvio padrão para $f(\lambda, \rho c_p) = 0,000447$, mostra que os resultados não desviam muito da média. Para esta versão do algoritmo, o melhor (menor) tempo computacional necessário para estimar os parâmetros de interesse foi igual a 4543 ms e o pior (maior) tempo foi igual a 24100 ms. A média e a mediana foram, respectivamente, iguais a 14561 ms e 14837 ms, enquanto o desvio padrão foi igual a 3430 ms.

Já na Tabela 2 são mostrados os resultados obtidos pela versão clássica do método PCA, sendo o número de avaliações da função objetivo igual 85 para o melhor caso. O desvio padrão

Tabela 1- Resultados obtidos pelo método Luus-Jaakola na versão clássica para o problema de transferência de calor.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,843451	3862946,26667	0,029427	227	4543
Pior	11,792125	3862876,380008	0,029125	1232	24100
Média	11,545312	3862700,318705	0,029220	681	14561
Mediana	11,332384	3856706,222554	0,029200	702	14837
Desvio Padrão	0,246520	7238,788952	0,000447	148	3430

para $f(\lambda, \rho c_p)$ foi igual a 0,000423 e, assim como na versão clássica do método Luus-Jaakola, os resultados apresentaram um desvio padrão baixo. A versão clássica do PCA obteve melhores resultados para o problema abordado, em relação à versão clássica do Luus-Jaakola.

Tabela 2- Resultados obtidos pelo método PCA na versão clássica para o problema de transferência de calor.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,62529	3867808,499111	0,028564	85	2787
Pior	11,668145	3874394,593906	0,029394	946	30242
Média	11,648396	3865222,851587	0,029230	416	12948
Mediana	11,850866	3862856,357012	0,029547	409	12524
Desvio Padrão	0,220402	6777,322211	0,000423	183	5.753

Na sequência, na Tabela 3 são apresentados os resultados do método LJ, com alteração na combinação de estimativas, onde o NAF foi de 66 no melhor resultado.

Tabela 3- Resultados obtidos pelo método LJ (LJ-M1), com combinação de estimativas.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,348177	3862323,4761	0,028881	66	1464
Pior	11,729473	3867708,714182	0,028658	780	16150
Média	11,558500	3863989,728552	0,029271	495	9962
Mediana	11,814359	3872059,083399	0,029466	487	9803
Desvio Padrão	0,252759	7642,957763	0,000462	152	3149

Por fim, no método PCA modificado, Tabela 4, observa-se NAF igual a 49, o que é uma solução satisfatória. A média possui NAF igual a 332 e, NAF igual a 354 para mediana. O desvio padrão obteve um NAF igual a 112. É possível observar que a modificação realizada no método PCA melhorou o resultado, tanto em relação ao NAF quanto em relação à distribuição dos parâmetros de interesse.

Também foi analisada como ocorreram as avaliações da função objetivo ao longo das 100 execuções de cada método. Na Fig. 2, é apresentada a distribuição dos dados, conforme resultados dos métodos LJ e PCA, nas versões clássicas e suas respectivas modificações, onde é possível perceber que a versão clássica do PCA obteve os resultados com um número menor de avaliações da função objetivo, além dos dados resultantes estarem mais concentrados em torno da média quando comparado aos outros métodos. Por outro lado, a versão clássica do

Tabela 4- Resultados obtidos pelo método PCA (PCA-M1), na versão clássica, com combinação de estimativas.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,929746	3869631,894343	0,029433	49	2585
Pior	11,499198	3854252,397865	0,029204	562	21518
Média	11,657043	3865764,479263	0,029286	332	11561
Mediana	11,535788	3870681158195	0,029096	354	11574
Desvio Padrão	0,224947	8135,798915	0,000461	112	4210

Luus-Jaakola possui *outliers*, inclusive abaixo a cauda inferior, logo, o melhor caso com NAF igual a 227 trata-se de uma situação atípica. Isso quer dizer que, dos 100 resultados gerados, após calcular os quartis, a mediana e as caudas (limites inferiores e superiores), qualquer valor que se encontre acima da cauda superior ou abaixo da cauda inferior, são exceções, distantes da maioria dos resultados encontrados. O PCA na versão clássica também possui *outliers*, mas na parte superior da cauda e, por isso, o melhor caso, com NAF igual a 85, implica em um resultado mais robusto. Por fim, também é possível perceber que além de reduzir o NAF no melhor caso em relação à versão clássica, obtendo um NAF igual a 66, os resultados obtidos pela versão modificada do método LJ estão concentrados em torno da média, cujo NAF foi igual a 495.

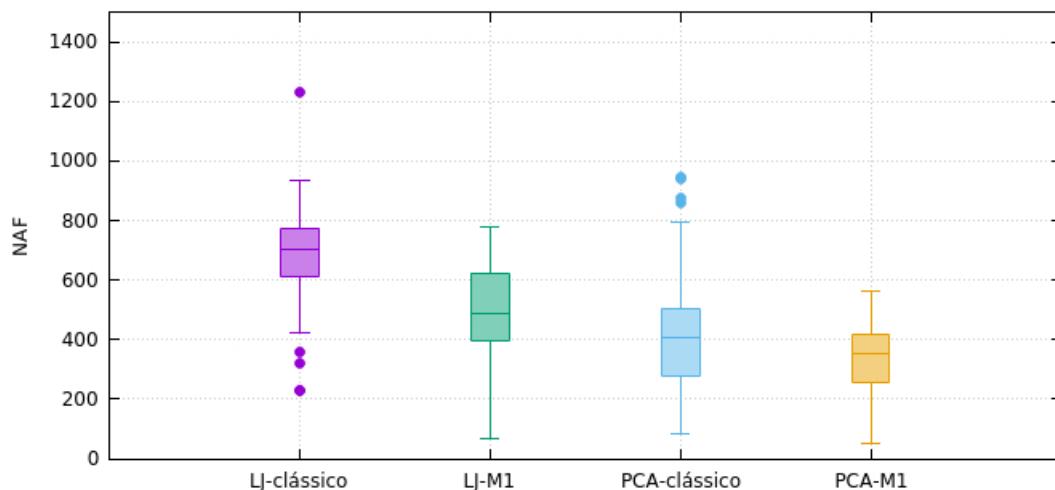


Figura 2- Distribuição dos dados para os métodos LJ e PCA nas versões clássicas e modificadas.

Fonte: O Autor, 2019.

7. CONCLUSÃO

Ao analisar os resultados obtidos, verifica-se que a versão clássica do PCA foi melhor do que a versão clássica do LJ quando aplicados ao problema de transferência de calor abordado nesse trabalho.

No que se refere aos métodos de otimização LJ e PCA modificados, conclui-se que os mesmos obtiveram melhores resultados, tanto em relação ao NAF, quanto em relação à distribuição dos dados, quando comparados às versões clássicas. As modificações apresentaram resultados mais robustos, o que sinaliza que a técnica de combinação de estimativas se mostraram ideais para a resolução desse problema.

Por fim, conclui-se ainda que, ao se comparar as 4 versões apresentadas nesse trabalho, o PCA na versão modificada foi a que melhor atendeu ao problema, obtendo melhores resultados em relação ao NAF e na concentração dos dados em torno da média. Sendo assim, o método PCA-M1, a princípio, é o mais indicado para resolver o problema de transferência de calor aqui apresentado, uma vez que a modificação apresentou-se mais robusta em detrimento às demais abordadas aqui.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- [1] Assael, M. J., Gialou, K. 2003. *Measurement of the Thermal Conductivity of Stainless Steel AISI 304L up to 550 K*. International Journal of Thermophysics, v 24, pp 1145-1153.
- [2] Blackwell, B. F., Gill, W., Dowding, K., Voth, T. (2000). *Determination of Thermal Conductivity of 304 Stainless Steel Using Parameter Estimation Techniques*. The 34 th National Heat Transfer Conference – NHTC 2000, Pittsburgh, Pennsylvania, United States.
- [3] Cezaro, A. *Problemas Inversos: Uma Introdução*. Colóquio de Matemática da Região Sul, Santa Maria, 2010, pp. 01-163.
- [4] Luus, R, Jaakola, T. H. I. 1973. *Optimization by Direct Search and Systematic Reduction of the Size of Search Region*. American Institute Chemical Engineering, vol. 19, pp. 760-766.
- [5] Sacco, W. F., Oliveira, C. R. E. 2005. *A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic*. 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro.
- [6] Silva Neto, A. J., Becceneri, J. C., Campos Velho, H. F. 2016. *Inteligência Computacional Aplicada a Problemas Inversos em Transferência Radiativa*. Rio de Janeiro: EdUERJ.
- [7] Telles, W. R. 2014. *Previsão do comportamento hidráulico de um rio com base na estimativa de coeficientes que controlam seu escoamento. Estudo de caso: Rio Bengalas, Nova Friburgo-RJ*. Rio de Janeiro.
- [8] Lobato, F. S., Alemida, G. M., Fernandes, C. F. 2010. *Algoritmo de Colisão de Partículas Aplicado ao Projeto de Sistema de Engenharia*. Nono Simpósio de Mecânica Computacional, UFSJ.
- [9] Luz, E. F. P., Becceneri, J. C. Becceneri , Velho H. F. C. 2008. *Uma nova metaheurística baseada em algoritmo de colisão de múltiplas partículas*. SPOLM: Rio de Janeiro, ISSN 2175-6295.