



# EAMC2017

X Encontro Acadêmico de Modelagem Computacional

Laboratório Nacional de Computação Científica - LNCC/MCTIC

01 a 03 de Fevereiro de 2017.

## Livro de resumos



Laboratório  
Nacional de  
Computação  
Científica

# **X Encontro Acadêmico de Modelagem Computacional**

Laboratório Nacional de Computação Científica - LNCC/MCTIC

01 a 03 de Fevereiro de 2017. Petrópolis - RJ.

## **Comitê Organizador**

Fortià Vila Verges

Joventino Oliveira Campos

Léo Rodrigues Biscassi

Lucas dos Anjos

Natanael Júnior Soares Bento

Natasha do Nascimento Paiva

Viviane Tavares Maia

Weslley da Silva Pereira

## **Comitê Científico**

Antônio André Novotny - LNCC

Antônio Tadeu Azevedo Gomes - LNCC

Camila Silva de Magalhães - UFRJ

Kary Ann del Carmen Ocaña Gautherot - LNCC

Gustavo Mender Platt - UERJ

Heder Soares Bernardino - UFJF

Leonardo Goliatt da Fonseca - UFJF

Luciane Prioli Ciapina - LNCC

Marcos Garcia Todorov - LNCC

Michel Iskin da Silveira Costa - LNCC

Pablo Javier Blanco - LNCC

Rafael Alves Bonfim - UFJF

Renato Portugal - LNCC

Yoisell Rodríguez Núñez - UFF

# Trabalhos completos

	Page
<i>Modelagem da sinalização intracelular</i> Anna Claudia M. Resende, Heber L. Rocha, Regina C. Almeida, Ernesto A.B.F. Lima	3
<i>Modelagem e análise da malha aérea doméstica brasileira com grafos multiaspectos (MAGs)</i> Bernardo Costa, João Victor Bechara, Klaus Wehmuth, Artur Ziviani	11
<i>A derivada topológica aplicada em problemas inversos e de otimização topológica</i> Cinthia Gomes Lopes, Thiago José Machado	18
<i>Desenvolvimento de uma ferramenta para geração automática de um banco de dados com base em sistema mecânico simulado para treinamento de uma deep neural network-dnn</i> Daniel de M.L. dos Santos, Francisco S.M. Sinfrônio, André B. Cavalcante, Allan K. Barros	29
<i>Um modelo para avaliação de desempenho de plataformas de processamento de grafos</i> Daniel N.R. da Silva, Klaus Wehmuth, Carla Oshtoff, Ana Paula Appel, Artur Ziviani	39
<i>Avaliação de funções empíricas de scoring em experimentos de triagem virtual em larga escala</i> Felipe S.S. Pereira, Isabella A. Guedes, Laurent E. Dardenne	46
<i>Prognóstico de dano estrutural utilizando ondas guiadas ultrassônicas</i> Jorge Franklin Mansur Rodrigues Filho, Gláucio Soares da Fonseca	56
<i>Um estudo comparativo de simulação de hipertermia com nanopartículas magnéticas</i> Gustavo Resende Fatigate, Ruy Freitas Reis, Rafael Felipe Coelho Neves	62

---

<i>Determinação de parâmetros com distribuição espacial não uniforme num modelo de neurociência via o método de Ladweber não linear</i> Jemy A. Mandujano V., Alexandre L. Madureira	70
<i>Simulação orientada a objetos da nucleação e crescimento via autômato celular</i> Jonathas Luis G. Ferreira, Wesley Luiz S. Assis, Tiago A. Neves, Guilherme C. de Oliveira	77
<i>Análise do efeito da precisão finita no algoritmo adaptativo sigmoidal</i> José de R.S. Fonseca, Ewaldo E.C. Santana, André C.B. Cavalcante, Allan K. Barros, Raimundo C.S. Freire	85
<i>Contact assisted protein structure prediction using a genetic algorithm with phenotypic crowding</i> Karina B. Santos, Gregório K. Rocha, Fábio L. Custódio, Laurent E. Dardenne	93
<i>Biological control of a pest in a reaction-diffusion predator-prey system with allee effect</i> Lucas dos Anjos, Regina C. Almeida, Michel Iskin	103
<i>Métodos de elementos finitos mistos para o problema de elasticidade linear em malhas de quadriláteros</i> Thiago O. Quinelato, Abimael F.D. Loula, Maicon R. Correa, Todd Arbogast	110
<i>Ranking texture features through AdaBoost.M2 linear ensembles for granite tiles classification</i> Tiene Andre Filisbino, Gilson Antonio Giraldi, Carlos Eduardo Thomaz, Bruno Montandon Noronha Barros, Millena Basílio da Silva	120
<i>Simulação numérica do tratamento térmico de austêmpera no ferro fundido nodular</i> Vinícius José de Carvalho Menezes, Gláucio Soares da Fonseca	132

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### Modelagem da Sinalização Intracelular

**Anna Claudia M. Resende**

**Heber L. Rocha**

**Regina C. Almeida**

annacmr@lncc.br

heberlr@lncc.br

rcca@lncc.br

Laboratório Nacional de Computação Científica (LNCC/MCTI)

Av. Getúlio Vargas 333, 25651-075, Petrópolis, Rio de Janeiro, Brazil

**Ernesto A. B. F. Lima**

lima@ices.utexas.edu

Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin

201 East 24th St., 78712-1229, Austin, Texas, USA

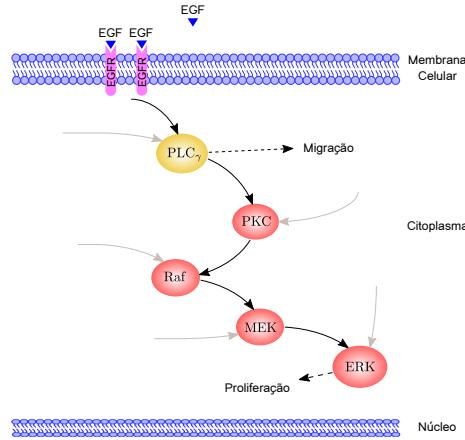
**Resumo.** Os processos celulares são regulados por redes integradas de comunicação intra e inter-células, constituídas por um número grande de proteínas e moléculas, que interagem bioquimicamente de modo a produzir respostas aos estímulos do microambiente. Tais respostas são locais, individuais de cada célula, dependentes da concentração intracelular de determinadas proteínas e/ou moléculas e de estímulos extracelulares. O entendimento dos processos subcelulares que regulam individualmente as transições de fenótipos, por exemplo, é fundamental para a representação mais realista da dinâmica celular e, em última instância, da progressão de várias patologias, como o câncer. Neste trabalho estudamos a rede de sinalização do fator de crescimento epidérmico (EGF), cuja alteração está associada à diversos tipos de câncer. A partir de um modelo relativamente simples, objetivamos identificar suas principais propriedades, funções e parâmetros importantes. Avaliamos possíveis simplificações e sofisticações através da análise de sensibilidade, utilizando o método dos efeitos elementares.

**Palavras-chave:** Modelagem Matemática, Sinalização Celular, Análise de Sensibilidade.

# 1 INTRODUÇÃO

O desenvolvimento crescente das pesquisas moleculares em câncer têm apontado para a importância de redes de sinalização bioquímicas em vários processos celulares. Estímulos extracelulares são internalizados através de receptores específicos da membrana celular, desencadeando cascatas de reações bioquímicas entre moléculas, denominadas redes de sinalização, que culminam na produção de uma (ou mais) resposta regulatória (Hanahan & Weinberg, 2011 e Kholodenko *et al.*, 2010). A complexidade destes mecanismos tem desafiado o desenvolvimento de alvos quimioterápicos, cujo sucesso depende fortemente do entendimento de como a expressão gênica é regulada através das sinalizações, de modo a obter terapias eficientes e menos agressivas. Em termos de modelagem, redes lógicas e Bayesianas, modelos estequiométricos, etc. são técnicas bastante utilizadas, embora a abordagem mais comum seja fundamentada na perspectiva do *continuum* via equações diferenciais ordinárias (EDOs) (Klipp *et al.*, 2016). Neste caso, cada rede é representada por um sistema acoplado e não-linear de EDOs, construído assumindo que as moléculas estão bem misturadas, de modo que a hipótese de homogeneidade espacial é razoável. Dois grandes desafios desta abordagem consistem na definição da granularidade do modelo, que corresponde ao número de equações do sistema, e na aquisição/estimação de parâmetros para a modelagem matemática e computacional.

Particularmente interessados na modelagem da sinalização que controla a proliferação celular tumoral, neste trabalho investigamos a rede de sinalização do fator de crescimento epidérmico (EGF). EGF é um mitógeno, uma substância que estimula a proliferação celular. Via um mecanismo autócrino, é liberado e usado pelas células, e passivamente transportado no microambiente tumoral, atingindo assim também células vizinhas. A sinalização do EGF é hiperativada em cerca de 30% de todos os cânceres, incluindo os de cólon, pulmão, pâncreas, dentre outros, sendo uma das principais redes de sinalizações envolvidas na progressão de tumores. Neste trabalho, utilizamos o modelo apresentado por Wang *et al.* (2007), definido por um sistema de vinte EDOs. A solução deste sistema é obtida usando o método de Runge-Kutta de quarta ordem e a identificação das moléculas-chave é obtida através da análise de sensibilidade (AS), utilizando o método dos efeitos elementares (EE) (Morris, 1991). Como a vantagem proliferativa é definida pela taxa de variação de duas moléculas específicas ( $PLC_{\gamma}$  e ERK), estas foram as quantidades de interesse (QoIs) utilizadas na AS. Nas seções seguintes apresentamos, de forma sucinta, o modelo utilizado, o método EE e sua utilização para a identificação dos parâmetros mais influentes nas QoIs. Finalmente, observações finais são apresentadas na seção 5.



**Figura 1:** Representação esquemática do modelo apresentado em Wang *et al.* (2007). Os triângulos azuis indicam os fatores de crescimento (EGF), em rosa estão indicados os receptores do ligante (EGFR), em amarelo a enzima  $PLC_{\gamma}$  e as proteínas quinase PKC, Raf, MEK e ERK em vermelho. As setas indicam os trajetos de sinalização (em preto), as confluências com outras vias de sinais biológicos (em cinza) e vantagens regulatórias (tracejadas).

## 2 MODELO CINÉTICO

A sinalização de EGF tem papel crucial na divisão e diferenciação celular. A sinalização anormal ocorre quando há hiperexpressão de receptores, desencadeando a amplificação de moléculas-chave implicadas na divisão celular e também em outros processos regulatórios, como a migração e a apoptose celular. O modelo apresentado em Wang *et al.* (2007) é um modelo cinético formado por um sistema de EDOs que descreve a sinalização entre 20 moléculas/complexos distintos. Além do ligante EGF e do receptor de membrana EGFR, 18 outras variáveis estão interligadas na cascata de sinalização, incluídas a enzima PLC $\gamma$  e as proteínas Raf, MEK e ERK. A descrição de cada molécula e associada reação cinética, que representa o balanço entre a produção e o consumo das espécies moleculares, são apresentadas nas Tabelas 1 e 2. A Fig. 1 apresenta um esquema simplificado desta cascata, estando também indicadas as confluências com outras trajetórias de sinalização (*cross-talks*). Conforme Wang *et al.* (2007), as ativações da PLC $\gamma$  e da ERK estão associadas à respostas celulares específicas de migração e proliferação celular (ver Fig. 1). Assim, as taxas das variáveis PLC $\gamma$ -P ( $X_9$ ) e ERK-PP ( $X_{20}$ ) são as quantidades de interesse da análise de sensibilidade apresentada a seguir.

## 3 ANÁLISE DE SENSIBILIDADE

A análise de sensibilidade é uma ferramenta importante para determinar como pequenas mudanças nos parâmetros de entrada de um modelo influenciam na variação de uma determinada quantidade de interesse (QoI) (Saltelli *et al.*, 2008). Neste trabalho, utilizando o método EE, avaliamos as taxas de variações de ERK-PP ( $X_{20}$ ) e PLC $\gamma$ -P ( $X_9$ ) por terem um papel crucial na mudança de fenótipos celulares. Tais QoIs são representadas por uma função  $Y(\theta)$ , em que  $\theta = (\theta_1, \dots, \theta_d)$  é um vetor composto por  $d$  parâmetros de entrada independentes.

Considerando que todas as variáveis do espaço paramétrico são transformadas em variáveis adimensionais em um hipercubo unitário ( $\theta \in [0, 1]^d$ ), o EE do  $i$ -ésimo parâmetro de entrada é definido como:

$$EE_i = \frac{Y(\theta_1, \dots, \theta_i + \Delta, \dots, \theta_d) - Y(\theta)}{\Delta} = \frac{Y(\theta + e_i \Delta) - Y(\theta)}{\Delta}, \quad (1)$$

em que  $\Delta \in \{1/(p-1), \dots, 1-1/(p-1)\}$  e  $p$  representa o número de níveis de discretização do hipercubo. Sendo  $e_i$  um vetor de zeros com uma unidade em sua  $i$ -ésima componente, observa-se que  $(\theta + e_i \Delta)$  difere de  $\theta$  apenas em sua  $i$ -ésima componente, que foi incrementada em

**Tabela 1: Equações cinéticas e concentrações iniciais [nM] para cada variável molecular  $X_i$ . Para complementar, veja a Tabela 2.**

Reagente	Variável Molecular	Concentração Inicial	EDO
$X_1$	EGF	$\theta_{39}$	$d(X_1)/dt = -v_1$
$X_2$	EGFR	$\theta_{40}$	$d(X_2)/dt = -v_1$
$X_3$	EGF-EGFR	-	$d(X_3)/dt = v_1 - 2v_2$
$X_4$	(EGF-EGFR)2	-	$d(X_4)/dt = v_2 + v_4 - v_3$
$X_5$	EGF-EGFR-P	-	$d(X_5)/dt = v_3 + v_7 - v_4 - v_5$
$X_6$	PLC $\gamma$	$\theta_{41}$	$d(X_6)/dt = v_8 - v_5$
$X_7$	EGF-EGFR-PLC $\gamma$	-	$d(X_7)/dt = v_5 - v_6$
$X_8$	EGF-EGFR-PLC $\gamma$ -P	-	$d(X_8)/dt = v_6 - v_7$
$X_9$	PLC $\gamma$ -P	-	$d(X_9)/dt = v_7 - v_8 - v_9 - v_{10}$
$X_{10}$	PLC $\gamma$ -P-I	-	$d(X_{10})/dt = v_9$
$X_{11}$	PKC	$\theta_{42}$	$d(X_{11})/dt = -v_{10}$
$X_{12}$	PKC*	-	$d(X_{12})/dt = v_{10} - v_{11}$
$X_{13}$	Raf	$\theta_{43}$	$d(X_{13})/dt = -v_{11}$
$X_{14}$	Raf*	-	$d(X_{14})/dt = v_{11} - v_{12} - v_{14}$
$X_{15}$	MEK	$\theta_{44}$	$d(X_{15})/dt = v_{13} - v_{12}$
$X_{16}$	MEK-P	-	$d(X_{16})/dt = v_{12} + v_{15} - v_{13} - v_{14}$
$X_{17}$	MEK-PP	-	$d(X_{17})/dt = v_{14} - v_{15} - v_{16} - v_{18}$
$X_{18}$	ERK	$\theta_{45}$	$d(X_{18})/dt = v_{17} - v_{16}$
$X_{19}$	ERK-P	-	$d(X_{19})/dt = v_{16} + v_{19} - v_{17} - v_{18}$
$X_{20}$	ERK-PP	-	$d(X_{20})/dt = v_{18} - v_{19}$

$\Delta$  (Saltelli *et al.*, 2008). Medidas de sensibilidade globais são calculadas a partir dos valores obtidos para cada EE, utilizando como estimativas a média absoluta  $\mu_i^*$  e o desvio padrão  $\sigma_i$  definidas como:

$$\mu_i^* = \frac{1}{r} \sum_{j=1}^r |EE_i^j|; \quad \sigma_i^2 = \frac{1}{r-1} \sum_{j=1}^r (EE_i^j - \mu_i)^2, \quad (2)$$

em que  $r$  representa o número de trajetórias utilizadas para “percorrer” o espaço paramétrico. A média  $\mu_i^*$  avalia o impacto do  $i$ -ésimo parâmetro sobre a QoI analisada e o desvio padrão  $\sigma_i$  determina a interação linear/não-linear existente entre os parâmetros e a QoI. Consideramos um total de  $d = 45$  parâmetros, identificados nas Tabelas 1 e 2, cada um tendo uma distribuição uniforme contínua  $\mathcal{U}(0.8\bar{\theta}_i; 1.2\bar{\theta}_i)$ , sendo  $\bar{\theta}_i$  o valor médio listado nas Tabelas 1 e 2. Maiores detalhes de implementação podem ser obtidos em Resende (2016).

## 4 EXPERIMENTOS

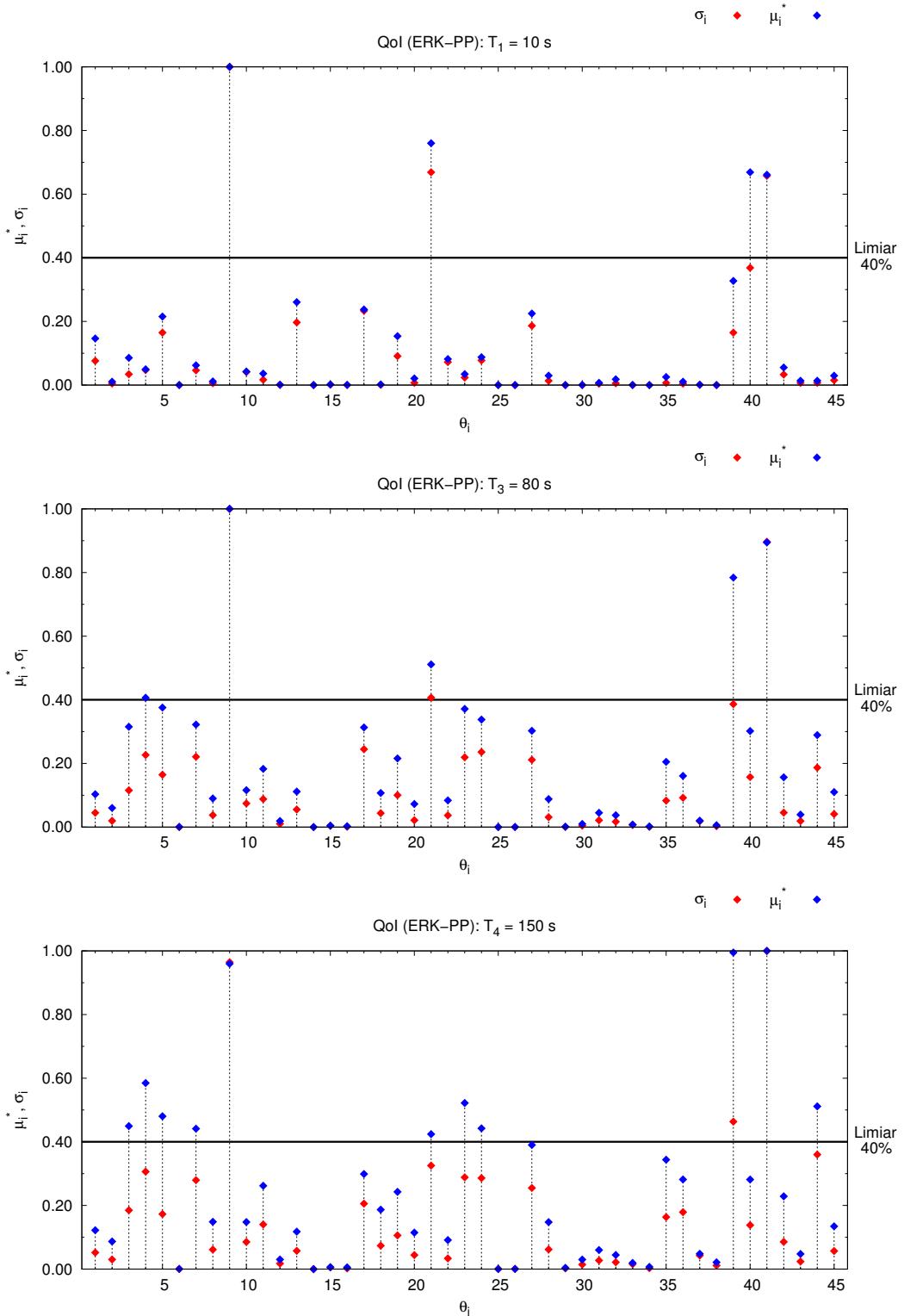
Na ausência de dados experimentais, assumimos que os  $d = 45$  parâmetros  $\theta_i$  (Tabelas 1 e 2) são independentes e estão distribuídos uniformemente segundo  $\mathcal{U}(0.8\bar{\theta}_i; 1.2\bar{\theta}_i)$ . O hipercubo paramétrico foi dividido em  $p = 4$  níveis, com  $\Delta = 1/3$ , e utilizamos  $r = 20$  trajetórias para o cálculo dos EEs.

A AS é apresentada nas Figs. 2 e 3 (referentes às taxas de variação do ERK-PP e do  $PLC_\gamma$ -P, respectivamente) para três diferentes tempos de simulação:  $T_1 = 10\text{ s}$ ,  $T_2 = 80\text{ s}$  e  $T_3 = 150\text{ s}$ . Para compatibilizar a comparação entre os estimadores, todos os valores obtidos para  $\mu_i^*$  e  $\sigma_i$  foram norma-

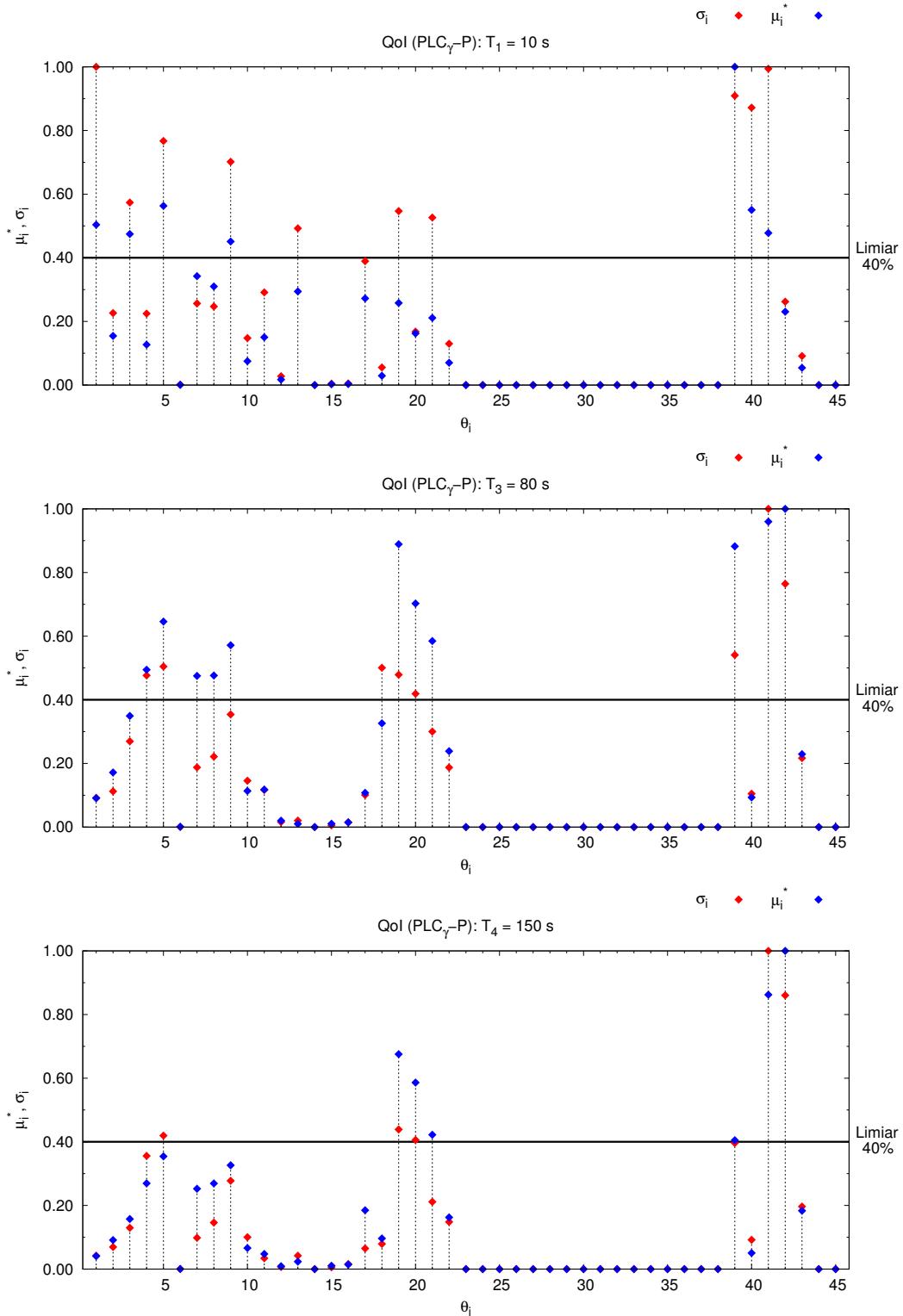
**Tabela 2: Reações (R) e parâmetros cinéticos.** As concentrações e as constantes de Michaelis-Menten ( $M$ ) são dadas em [nM]. Taxas constantes de primeira e segunda ordem são dadas em [ $s^{-1}$ ] e [ $nM^{-1}s^{-1}$ ], respectivamente. As velocidades  $V$  são expressas em [ $nMs^{-1}$ ] (ver Wang *et al.* (2007)).

R.	Equação	Parâmetros Cinéticos			
		$\theta_1$	$k_1 = 3.0 \times 10^{-3}$	$\theta_2$	$k_{r1} = 0.06$
$v_1$	$k_1 \cdot X_1 \cdot X_2 - k_{r1} \cdot X_3$	$\theta_3$	$k_2 = 0.01$	$\theta_4$	$k_{r2} = 0.1$
$v_2$	$k_2 \cdot X_3 \cdot X_4 - k_{r2} \cdot X_4$	$\theta_5$	$k_3 = 1.0$	$\theta_6$	$k_{r3} = 0.01$
$v_3$	$k_3 \cdot X_4 - k_{r3} \cdot X_5$	$\theta_7$	$V_4 = 450.0$	$\theta_8$	$M_4 = 50.0$
$v_4$	$V_4 \cdot X_5 / (M_4 + X_5)$	$\theta_9$	$k_5 = 0.06$	$\theta_{10}$	$k_{r5} = 0.2$
$v_5$	$k_5 \cdot X_5 \cdot X_6 - k_{r5} \cdot X_7$	$\theta_{11}$	$k_6 = 1.0$	$\theta_{12}$	$k_{r6} = 0.05$
$v_6$	$k_6 \cdot X_7 - k_{r6} \cdot X_8$	$\theta_{13}$	$k_7 = 0.3$	$\theta_{14}$	$k_{r7} = 6.0 \times 10^{-3}$
$v_7$	$k_7 \cdot X_8 - k_{r7} \cdot X_5 \cdot X_9$	$\theta_{15}$	$V_8 = 1.0$	$\theta_{16}$	$M_8 = 100.0$
$v_8$	$V_8 \cdot X_9 / (M_8 + X_9)$	$\theta_{17}$	$k_9 = 1.0$	$\theta_{18}$	$k_{r9} = 0.03$
$v_9$	$k_9 \cdot X_9 - k_{r9} \cdot X_{10}$	$\theta_{19}$	$k_{10} = 0.214$	$\theta_{20}$	$k_{r10} = 5.25$
$v_{10}$	$k_{10} \cdot X_9 \cdot X_{11} - k_{r10} X_{12}$	$\theta_{21}$	$V_{11} = 4.0$	$\theta_{22}$	$M_{11} = 64.0$
$v_{11}$	$V_{11} \cdot X_{12} \cdot X_{13} / (M_{11} + X_{13})$	$\theta_{23}$	$V_{12} = 3.5$	$\theta_{24}$	$M_{12} = 317.0$
$v_{12}$	$V_{12} \cdot X_{14} \cdot X_{15} / [M_{12}(1 + X_{16}/M_{14}) + X_{15}]$	$\theta_{25}$	$V_{13} = 0.058$	$\theta_{26}$	$M_{13} = 2.2 \times 10^3$
$v_{13}$	$V_{13} \cdot X_{16} / [M_{13}(1 + X_{17}/M_{15}) + X_{16}]$	$\theta_{27}$	$V_{14} = 2.9$	$\theta_{28}$	$M_{14} = 317.0$
$v_{14}$	$V_{14} \cdot X_{14} \cdot X_{16} / [M_{14}(1 + X_{15}/M_{12}) + X_{16}]$	$\theta_{29}$	$V_{15} = 0.058$	$\theta_{30}$	$M_{15} = 60.0$
$v_{15}$	$V_{15} \cdot X_{17} / [M_{15}(1 + X_{16}/M_{13}) + X_{17}]$	$\theta_{31}$	$V_{16} = 9.5$	$\theta_{32}$	$M_{16} = 1.46 \times 10^5$
$v_{16}$	$V_{16} \cdot X_{17} \cdot X_{18} / [M_{16}(1 + X_{19}/M_{18}) + X_{18}]$	$\theta_{33}$	$V_{17} = 0.3$	$\theta_{34}$	$M_{17} = 160.0$
$v_{17}$	$V_{17} \cdot X_{19} / [M_{17}(1 + X_{20}/M_{19}) + X_{19}]$	$\theta_{35}$	$V_{18} = 16.0$	$\theta_{36}$	$M_{18} = 1.46 \times 10^5$
$v_{18}$	$V_{18} \cdot X_{17} \cdot X_{19} / [M_{18}(1 + X_{18}/M_{16}) + X_{19}]$	$\theta_{37}$	$V_{19} = 0.27$	$\theta_{38}$	$M_{19} = 60.0$
$v_{19}$	$V_{19} \cdot X_{20} / [M_{19}(1 + X_{19}/M_{17}) + X_{20}]$				

lizados a partir dos valores máximos de cada experimento. Estipulamos um limiar de 40% (para exemplificar) acima do qual identificamos os parâmetros  $\theta_i$  que impactam a dinâmica das QoIs. Como esperado, a quantidade do estímulo extracelular (EGF) e de receptores de membrana (EGFR) são importantes na dinâmica inicial e na manutenção da resposta celular. A Fig. 2 indica que, para os tempos  $T_1$  e  $T_2$ , os parâmetros que mais influenciam a dinâmica da taxa de variação do ERK-P ( $X_{20}$ ), em ordem decrescente de importância, são  $\{\theta_9, \theta_{21}, \theta_{40}, \theta_{41}\}$  e  $\{\theta_9, \theta_{41}, \theta_{39}, \theta_{21}\}$ , respectivamente. Note a importância da comunicação transversa introduzida pela condição inicial de  $PLC_\gamma$ , que, em conjunto com as reações  $v_5$  e  $v_{11}$ , atuam sobre a dinâmica das moléculas-chave  $PLC_\gamma$  e Raf.



**Figura 2:** AS dos parâmetros com relação à taxa de variação de ERK-PP em três diferentes tempos de simulação:  $T_1 = 10\text{ s}$ ,  $T_2 = 80\text{ s}$  e  $T_3 = 150\text{ s}$ . Os parâmetros  $\theta_i$  ( $i = 1, \dots, 38$ ) referem-se aos parâmetros cinéticos listados na Tabela 2 e os parâmetros restantes às condições iniciais listadas na Tabela 1. As retas traçadas em  $\mu_i^* = 0.4$  representam o limiar de importância estipulado para nossos experimentos (40%).



**Figura 3: AS dos parâmetros com relação à taxa de variação de PLC $\gamma$ -P em três diferentes tempos de simulação:  $T_1 = 10$  s,  $T_2 = 80$  s e  $T_3 = 150$  s. Os parâmetros  $\theta_i$  ( $i = 1, \dots, 38$ ) referem-se aos parâmetros cinéticos listados na Tabela 2 e os parâmetros restantes às condições iniciais listadas na Tabela 1. As retas traçadas em  $\mu_i^* = 0.4$  representam o limiar de importância estipulado nos experimentos (40%).**

Para tempos maiores, mais moléculas passam a influenciar na sustentação do sinal regulatório. Parâmetros relacionados às reações  $v_2$ ,  $v_3$ ,  $v_5$ ,  $v_{11}$  e  $v_{12}$  tornam-se importantes, assim como a condição inicial especificada para a variável molecular MEK. Com relação à taxa de variação de PLC $_{\gamma}$ -P ( $X_9$ ), observamos que inicialmente as medidas de  $\mu_i^*$  indicam que os parâmetros  $\{\theta_{39}, \theta_5, \theta_{40}, \theta_1, \theta_3, \theta_{41}, \theta_9\}$  são os mais influentes, destacando-se a intensidade de seus efeitos não-lineares sobre a QoI. Além disso, os parâmetros  $\{\theta_{19}, \theta_{21}, \theta_{13}\}$  apresentam efeito não-linear muito mais intenso do que o linear. Este cenário altera-se bastante ao longo do tempo, embora as reações envolvendo  $\{\theta_{23} - \theta_{39}\}$ , que estão a jusante de PLC $_{\gamma}$ -P na cascata bioquímica, mantenham-se não influentes.

## 5 CONCLUSÕES

De modo geral a análise de sensibilidade realizada no modelo apresentado em Wang *et al.* (2007) confirmou a importância de alguns *cross-talks* na cascata de sinalização e o papel das denominadas moléculas-chave. Simplificações e extensões deste modelo podem ser realizadas com o intuito de auxiliar a busca de modelos de baixa granularidade para representação efetiva dos processos intracelulares de regulação do ciclo celular. Um primeiro passo para reduzir a granularidade consiste em agrupar reações que estão envolvidas num mesmo processo bioquímico e então modelar o processo agregando novas hipóteses simplificadoras. Neste passo, o modelo de Michaelis-Menten é frequentemente utilizado (ver Aldridge *et al.*, 2006 e Hornberg *et al.*, 2005). Os resultados obtidos também permitem avaliar e comparar outros modelos de vias de sinalização de EGF disponíveis na literatura (Juty *et al.*, 2015).

## REFERÊNCIAS

- Aldridge, B. B., Burke, J. M., Lauffenburger, D. A., & Sorger, P. K. 2006. Physicochemical modelling of cell signalling pathways. *Nature Cell Biology*, **8(11)**, 1195.
- Hanahan, D., & Weinberg, R. A. 2011. Hallmarks of Cancer: The Next Generation. *Cell*, **144(5)**, 646–674.
- Hornberg, J. J., Bruggeman, F. J., Binder, B., Geest, C. R., de Vaate, A. J., Lankelma, J., Heinrich, Reinhart, & Westerhoff, H. V. 2005. Principles behind the multifarious control of signal transduction. ERK phosphorylation and kinase/phosphatase control. *The FEBS Journal*, **272(1)**, 244–258.
- Juty, N., Ali, R., Glont, M., Keating, S., Rodriguez, N.s, Swat, M. J., Wimalaratne, S. M., Hermjakob, H., Le Novère, N., Laibe, C., & Chelliah, V. 2015. BioModels: Content, Features, Functionality and Use. *CPT: Pharmacometrics & Systems Pharmacology*.
- Kholodenko, B. N., Hancock, J. F., & Kolch, W. 2010. Signalling ballet in space and time. *Nature Reviews: Molecular Cell Biology*, **6(11)**, 414–426.
- Klipp, E., Liebermeister, W., Wierling, C., & Kowald, A. 2016. *Systems Biology*. Wiley-VCH.
- Morris, M. D. 1991. Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics*, **33(2)**, 161–174.
- Resende, A. C. M. 2016. *Sensitivity Analysis as a Tool for Tumor Growth Modeling*. Dissertação de Mestrado, Laboratório Nacional de Computação Científica (LNCC/MCTI), Petrópolis/RJ, Brazil.
- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., & Tarantola, S. 2008. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons.
- Wang, Z., Zhang, L., Sagotsky, J., & Deisboeck, T. S. 2007. Simulating non-small cell lung cancer with a multiscale agent-based model. *Theoretical Biology & Medical Modelling*, **50(4)**.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### Modelagem e Análise da Malha Aérea Doméstica Brasileira com Grafos MultiAspectos (MAGs)

**Bernardo Costa** ([bantunes@lncc.br](mailto:bantunes@lncc.br))

**João Victor Bechara** ([joaoavmb@lncc.br](mailto:joaoavmb@lncc.br))

**Klaus Wehmuth** ([klaus@lncc.br](mailto:klaus@lncc.br))

**Artur Ziviani** ([ziviani@lncc.br](mailto:ziviani@lncc.br))

Laboratório Nacional de Computação Científica (LNCC)

Av. Getúlio Vargas, 333, Quitandinha

CEP 25651-075 – Petrópolis, RJ – Brasil

**Resumo.** *Uma mudança na rede de transporte aérea gera um grande impacto, já que abala a mobilidade da população e a economia no país. Neste artigo foram analisados os impactos que a recente crise econômica provocou na malha aérea brasileira. Essa análise foi efetuada utilizando informações disponibilizadas pela Agência Nacional de Aviação Civil (ANAC) em um momento anterior e um momento posterior ao estabelecimento do cenário de crise econômica atual. Para efetuar a análise dos dados obtidos, a malha aérea doméstica de transporte de passageiros foi modelada antes e depois do estabelecimento da crise, de forma a permitir a comparação entre os dois instantes de tempo. A malha aérea foi modelada por meio de um único Grafo MultiAspecto (MAG), abstração que permite a construção de modelos para redes variantes no tempo e multicamada, permitindo ainda a utilização de múltiplas escalas de tempo. Os resultados apresentados neste trabalho foram obtidos diretamente do modelo construído, através do uso de algoritmos conhecidos para grafos orientados, que podem ser diretamente aplicados a modelos construídos com MAGs.*

**Palavras-chave:** Redes multicamada, Redes variantes no tempo, K-core, Modelagem computacional, Malha aérea brasileira.

## 1 Introdução

Os sistemas de transporte estão muito presentes no dia-a-dia das pessoas, sendo componente fundamental da sociedade moderna. Por isso, torna interessante analisá-los para entender como funciona o deslocamento da população e como as empresas que providenciam esse serviço se preparam para suprir a demanda que vem crescendo bem como se modificando ao longo dos anos. Nesse contexto, a malha aérea doméstica brasileira pode ser entendida naturalmente como uma rede não trivial, tendo em vista que existem diversas companhias que atuam em diferentes regiões do país, oferecendo serviços para diferentes tipos de pessoas com um número diferentes de aviões.

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

Análises recentes da estrutura da malha aérea brasileira [1], bem como da malha aérea global [3, 6] foram elaboradas utilizando grafos agregados da malha aérea, contendo apenas informações sobre aeroportos (localidades) e rotas (ligações entre aeroportos). Esse tipo de análise desconsidera a informação temporal de escalonamento dos voos, fazendo com que caminhos compostos por várias rotas não sejam confiáveis, já que não existe informação que garanta que o encadeamento de voos sugerido por esses caminhos vistos na rede agregada seja mesmo realizável. Ou seja, nesses modelos com rede agregada não é possível garantir a sequência temporal das rotas, uma vez que não há informação temporal no grafo de rotas.

A análise proposta neste trabalho é elaborada de tal forma que todas as informações temporais de agendamento dos voos sejam preservada no modelo de malha aéreas considerado. Assim é sempre possível garantir que essas informações de agendamento dos voos sejam automaticamente utilizadas na construção de caminhos no modelo.

Todos os resultados apresentados nesse trabalho são obtidos através de um modelo de malha aérea composto por um único objeto matemático, capaz de representar a malha aérea em estudo com informação temporal em duas escalas distintas. Dessa forma, é possível obter a informação completa do agendamento de voos de todas as empresas aéreas consideradas no estudo em diferentes instantes de tempo (pré e pós-crise financeira), de forma a permitir a comparação da estrutura da malha nesses diferentes instantes de tempo.

## 2 Metodologia

Nesta seção, serão descritos os métodos e ferramentas utilizados na construção de modelo e obtenção dos resultados apresentados neste artigo.

### 2.1 MAG

Um Grafo MultiAspecto (MAG) [5] é uma generalização de grafo capaz de representar redes complexas dinâmicas com múltiplas camadas, variantes no tempo e multi-escala. Nesse contexto, um aspecto é uma característica independente da rede a ser modelada, como por exemplo localidades, camadas e instantes de tempo. Uma das principais características de um MAG é a de ser isomorfo a um grafo direcionado. Com isso, torna-se possível aplicar o conhecimento já obtido em teoria de grafos diretamente ao ambiente de MAGs.

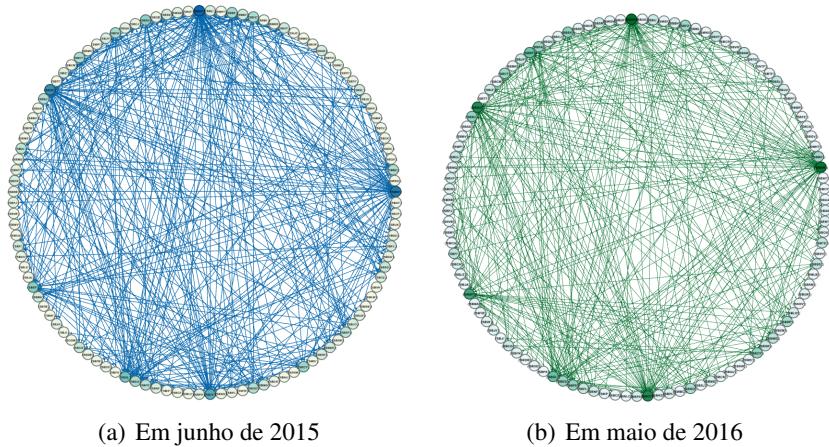
Em particular, algoritmos para aplicação em MAGs podem ser desenvolvidos tomando como base algoritmos já conhecidos para grafos orientados [4], facilitando a análise de redes complexas modeladas por MAGs. Além dos algoritmos e operações normalmente utilizados com grafos direcionados, um modelo baseado em MAGs permite de se faça agregações sobre aspectos, denominadas sub-determinações. As sub-determinações permitem a obtenção de resultados expressos em função de um subconjunto dos aspectos presentes no MAG, porém sempre utilizando todas as informações presentes no modelo para a obtenção do resultado.

Neste trabalho, a malha aérea brasileira é modelada como uma rede composta por localidades (aeroportos), camadas (empresas aéreas) e instantes de tempo. Porém, os resultados obtidos são expressos apenas em termos de localidades (aeroportos) e suas interligações (rotas), implicando o uso de algoritmos sub-determinados. O uso desses algoritmos garante que os resultados obtidos levem em conta a estrutura temporal e de camadas do modelo, de forma a respeitar a sequência temporal dos voos disponíveis, bem como as fronteiras de operação entre empresas aéreas distintas.

## 2.2 Construção do modelo

Para a representação da malha aérea doméstica brasileira foram usadas duas tabelas de voos vigentes nos dias 3 de junho de 2015 e 13 de maio de 2016, publicadas pela Agência Nacional de Aviação Civil (ANAC) em seu sítio eletrônico. Essas tabelas contém dados de voos nacionais, internacionais, postais e de carga no período de uma semana, onde foram extraídas as informações referentes aos voos domésticos comerciais de passageiros que foram o alvo deste trabalho.

O modelo baseado em MAG proposto neste trabalho possui 4 aspectos (aeroportos, empresas aéreas, instantes de tempo e época da base de dados). No primeiro aspecto, trataremos de todos os aeroportos referentes a época da base de dados que estamos utilizando. Para a base de dados de 2015 foram analisados 110 aeroportos e para a base de 2016, 109 aeroportos. No segundo aspecto, são representadas as camadas referentes às companhias aéreas. Para a base de dados de 2015 foram analisadas 7 companhias aéreas (Azul, Avianca, Gol, MAP, Passaredo, TAM, Sete), já para a base de 2016 foram 8 companhias aéreas (Azul, Avianca, Gol, Passaredo, TAM, MAP, TRIP e FlyWays). O terceiro aspecto registra todos os instantes de tempo, equivalentes aos momentos de ocorrência de cada um dos eventos durante uma semana e o quarto aspecto contém uma identificação da tabela referente ao dado analisado (neste trabalho foram utilizados duas tabelas). A Figura 1 mostra as malhas aéreas das duas bases de dados de 2015 e 2016, onde os aeroportos são representados pelos vértices e os voos pelas arestas.



**Figura 1: Malha aérea doméstica brasileira.**

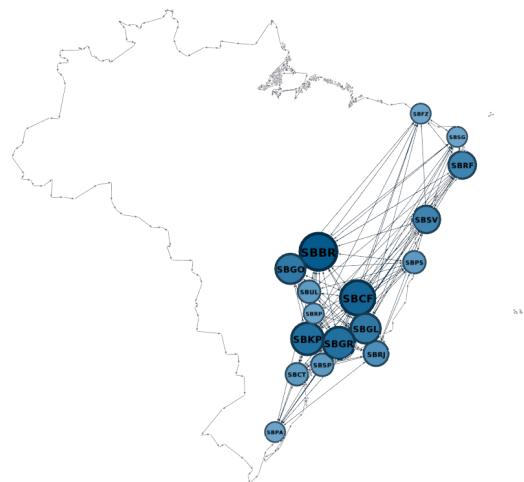
## 3 Método de identificação do núcleo central da rede

Neste estudo de caso, para identificação do núcleo central da rede foi utilizado o algoritmo K-Core [2]. O núcleo correspondente ao K-Core é obtido através de um processo de decomposição progressiva da rede, no qual remove-se progressivamente todos os vértices com conectividade inferior ao valor crescente de  $k$  e suas respectivas conexões. Por exemplo, o 2-Core de uma rede é formado pelo núcleo resultante da remoção de todos os vértices de grau 1, ou seja, o núcleo contendo somente vértices com no mínimo grau 2. O objetivo final do algoritmo é de que o último valor de  $k$  nos forneça apenas o núcleo central da rede, sem que haja a possibilidade de remoção de outros vértices.

A adoção do K-Core disponibiliza maior ganho nas análises, tendo em vista que retorna os nós com maior núcleo da rede, nesse caso, aeroportos com maior incidência de voos. Porém, é possível aplicar qualquer outra métrica na base de dados que seja conveniente para a análise.

Com essa ferramenta de análise é possível encontrar as principais propriedades da malha aérea brasileira das duas bases de dados. Os núcleos dos grafos resultantes de cada empresa aérea informa o quanto cada aeroporto é importante para cada companhia. Outras informações relevantes que são obtidas com o K-Core são: localização da concentração das atividades de cada empresa aérea, comparação entre as estratégias das empresas, identificação dos hubs das empresas e assim por diante. Esse método pode ser aplicado no grafos resultante das operações que serão descritas nas próximas subseções.

A Figura 2 representa o núcleo geo-referenciado da malha aérea referente a primeira base de dados analisada. É possível ver que as regiões com maior participação econômica do país também são as regiões com maior índice de voos.



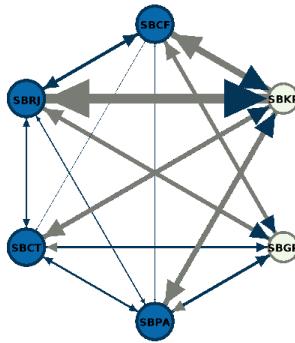
**Figura 2:** Núcleo geo-referenciado com todas as empresas em junho de 2015.

4 Operações MAG

Com a operação de sub-MAG, com um único MAG, é possível separar todas as companhias aéreas e analisar suas características individualmente, gerando um multi-dígrafo para cada companhia em cada período. Essa operação resulta em 14 objetos (para a base de dados de 2015) e 16 objetos (base de dados de 2016), onde se tem a informação da quantidade de voos entre os aeroportos. Para gerar os objetos em dígrafo, é possível realizar a operação de sub-determinação em relação ao aspecto de tempo, resultando nas arestas do MAG com três aspectos: aeroporto, companhia aérea e época. Para analisar somente as rotas de uma empresa, pode-se realizar a operação de subdeterminação nos sub-MAG's em relação ao aspecto de companhia aérea, deixando assim, as arestas com os aspectos de aeroporto, de tempo e de época. Dessa maneira, o resultado será objetos no formato multi-dígrafo. Para analisar apenas as rotas sem a informação da quantidade de voos, pode-se executar a subdeterminação em relação ao aspecto de

tempo, assim ficando com objetos no formato dígrafo, restando nas arestas apenas os aspectos de aeroportos e de época. Da mesma forma, com a operação de subdeterminação, é possível separar apenas umas das duas épocas para analisá-las separadamente. Esse tipo de flexibilidade na análise é alcançada pelo uso do MAG para modelagem.

A Figura 3 representa o núcleo da empresa aérea Azul. Para gerar esse tipo sub-dígrafo é necessário realizar primeiro a operação de sub-MAG, que retorna somente as arestas referentes as empresa aérea Azul. Em seguida, foi realizada a operação de subdeterminação em relação a empresa aérea, que retorna o dígrafo da Azul. Por ultimo, basta aplicar o algoritmo do K-Core no dígrafo resultante.



**Figura 3: Núcleo da empresa aérea Azul em junho de 2015.**

## 5 Comparativo da malha aérea brasileira: 2015 x 2016

A grave crise financeira enfrentada pelo país impacta diretamente diversos setores da economia, levando estes a se ajustarem face à nova realidade. Nesse contexto, o setor de aviação civil não é exceção e houve por parte das companhias aéreas um ajuste na sua disponibilidade de voos, tendo por consequência uma recente modificação na malha aérea doméstica brasileira.

Durante a análise comparativa das malhas aéreas em 2015 e 2016, foram encontradas diversas mudanças estruturais. Em primeiro, o número total de aeroportos foi reduzido de 110 para 109, mas a interseção da primeira base de dados (2015) e da segunda (2016) são apenas 101. Assim, 9 aeroportos não funcionam mais e 8 novos aeroportos começaram a operar. Outra mudança estrutural refere-se às companhias aéreas. A base de dados de 2015 possui 7 companhias aéreas, já a base de dados de 2016 possui 8 companhias aéreas. Porém, a interseção entre o conjunto das companhias das bases de dados resultam em 6 companhias aéreas. A companhia aérea Sete parou de operar da primeira (2015) para a segunda base de dados (2016). Enquanto isso, duas companhias aéreas começaram a operar voos comerciais nesse intervalo de tempo, as companhias TRIP e a FlyWays.

O número de componentes fortemente conexas na primeira base de dados foi de 7, uma por empresa aérea. Agora, este número aumentou para 10. Fato curioso é que a companhia aérea Trip possui 3 componentes fortemente conexos, mostrando como a crise econômica afetou a demanda dos voos e assim, as companhias aéreas tentaram se adaptar.

Uma das principais formas que as empresas aéreas encontraram para otimizar a utilização dos aviões, foi através do *codeshare*, ou seja, as companhias estão fazendo mais parcerias entre

si para compartilhar mesmos voos. Entre as bases de dados analisadas, o número de *codeshare* passou de 923 para 1234 voos compartilhados por semana, um aumento de cerca de 33% no número de voos compartilhados por semana entre junho de 2015 e maio de 2016.

A Tabela 1 mostra a diferença do número de voos entre junho de 2015 e maio de 2016. É possível notar que as três maiores companhias aéreas do país (Gol, Azul e TAM) e uma das empresas pequenas (Passaredo) tiveram reduções significativas em seus voos, especialmente a Gol com 37% de redução no período. Já as empresas médias e pequenas (Avianca e Map) tiveram aumento no seu número de voos (13% e 46% de aumento respectivamente). O crescimento em média das empresas menores em tempos de crise pode ser visto como estratégia de preenchimento das lacunas dos voos deixadas pelas empresas maiores.

**Tabela 1: Número de voos.**

Empresa aérea	Voos antes da crise	Voos depois da crise	Diferença (%)
Gol	6644	4188	-37
Azul	5839	5075	-13
TAM	4814	4158	-14
Avianca	1386	1561	13
Passaredo	661	612	-7
Map	98	143	46

## 6 Conclusão

Neste trabalho foi possível analisar as principais características da malha aérea brasileira nas duas bases de dados (junho de 2015 e maio de 2016). Uma análise comparativa traz resultados preliminares sobre como a crise econômica impactou na malha aérea brasileira e nas estratégias adotadas pelas companhias aéreas para se adaptar a essa crise.

Como resultado deste artigo, foi possível verificar os métodos adotados pelas companhias em resposta ao contexto de crise econômica. Verificou-se que, em média, as principais companhias aéreas tiveram um decréscimo considerável em sua quantidade de voos disponibilizados. Fato esse que levou a quantidade de voos cair cerca de 22% de uma base de dados para a outra. Enquanto isso, as pequenas e médias empresas viram como uma oportunidade de crescimento e por isso expandiram sua quantidade de voos e o número de aeroportos que operam. Foi visto que um dos principais métodos utilizados pelas grandes empresas foi a utilização do *codeshare*, havendo um aumento de 33% da quantidade de voos compartilhados de uma base de dados para a outra.

## Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq, FAPERJ e Ministério de Ciência, Tecnologia, Inovações e Comunicações (MCTIC).

# Referências Bibliográficas

- [1] G. S. COUTO, A. P. C. D. SILVA, L. B. RUIZ, AND F. BENEVENUTO, *Structural Properties of the Brazilian Air Transportation Network*, Anais da Academia Brasileira de Ciências, 87 (2015), pp. 1653 – 1674.
- [2] S. B. SEIDMAN, *Network structure and minimum degree*, Social Networks, 5 (1983), pp. 269 – 287.
- [3] T. VERMA, N. A. M. ARAÚJO, AND H. J. HERRMANN, *Revealing the structure of the world airline network*, Scientific Reports, 4 (2014), pp. 1–6.
- [4] K. WEHMUTH, É. FLEURY, AND A. ZIVIANI, *MultiAspect Graphs: Algebraic representation and algorithms*, arXiv: 1504.07893, (2015), pp. 1–61.
- [5] ——, *On MultiAspect graphs*, Theoretical Computer Science, 651 (2016), pp. 50–61.
- [6] P. WEI, L. CHEN, AND D. SUN, *Algebraic connectivity maximization of an air transportation network: The flight routes' addition/deletion problem*, Transportation Research Part E: Logistics and Transportation Review, 61 (2014), pp. 13–27.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### A DERIVADA TOPOLOGICA APLICADA EM PROBLEMAS INVERSOS E DE OTIMIZAÇÃO TOPOLOGICA

**Cinthia Gomes Lopes**

**Thiago José Machado**

cinthia@lncc.br

thiagojm@lncc.br

Laboratório Nacional de Computação Científica

Av. Getúlio Vargas 333, CEP 25651-075, Petrópolis, Rio de Janeiro, Brasil

**Resumo.** A derivada topológica foi rigorosamente definida por Sokołowski e Żochowski em 1999 e, desde então, tem se mostrado uma importante ferramenta para o tratamento de diferentes problemas, dentre os quais destacam-se: otimização topológica, problemas inversos, processamento de imagens, modelagem constitutiva multiescala, análise de sensibilidade à fratura mecânica, modelagem de evolução de dano e análise de sensibilidade em problemas de contato. A derivada topológica é obtida a partir da análise assintótica de soluções clássicas para problemas de valores de contorno em domínios singularmente perturbados, combinada com a análise assintótica de funcionais de forma com relação ao parâmetro que governa o tamanho da perturbação. Neste trabalho, são apresentados alguns resultados obtidos nos contextos de otimização topológica com restrição em tensão e problemas inversos de reconstrução de fonte.

**Palavras-chave:** Derivada topológica, Análise de sensibilidade, Otimização topológica, Problemas inversos.

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUÇÃO

A derivada topológica de um funcional de forma pode ser caracterizada como o termo de correção de primeira ordem proveniente de uma expansão assintótica deste funcional. Ao realizar perturbações em um domínio, tais como a inserção de furos ou termos fonte, a derivada topológica mensura a variação do funcional. Logo, em problemas onde se deseja minimizar um funcional, utiliza-se a derivada topológica para indicar uma direção de descida.

Desde que foi definida em Sokołowski & Żochowski (1999), a derivada topológica tem se mostrado uma importante ferramenta para vários problemas, tais como: otimização topológica, problemas inversos, processamento de imagens, mecânica da fratura, modelagem de evolução do dano e modelagem multi-escala. Para mais detalhes o leitor pode consultar Novotny & Sokołowski (2013). Em particular, este trabalho traz alguns resultados recentes de aplicações da derivada topológica na obtenção de estruturas e mecanismos flexíveis através de otimização topológica Amstutz & Novotny (2010); Amstutz *et al.* (2012); Lopes & Novotny (2016); Lopes *et al.* (2015) e na resolução de problemas inversos de reconstrução de fontes Canelas *et al.* (2014, 2015); Machado *et al.* (2016).

## 2 A DEFINIÇÃO DA DERIVADA TOPOLÓGICA

Considere um domínio  $\Omega \subset \mathbb{R}^d$ , com  $d \geq 2$ . Associado ao domínio  $\Omega$ , introduz-se uma função característica  $\chi : \mathbb{R}^d \rightarrow \{0, 1\}$  tal que  $\chi = 1_\Omega$ . Suponha que  $\Omega$  seja sujeito a uma perturbação infinitesimal confinada em uma pequena região  $\omega_\varepsilon(\hat{x}) = \hat{x} + \varepsilon\omega$ , onde  $\hat{x}$  é um ponto arbitrário de  $\Omega$ ,  $\varepsilon > 0$  e  $\omega \subset \Omega$  é um conjunto conexo. Esta perturbação gera uma nova função  $\chi_\varepsilon : \Omega \times \mathbb{R}^d \rightarrow \{0, 1, \gamma\}$  tal que

$$\chi_\varepsilon(\hat{x}, x) = \begin{cases} \chi(x), & \text{se } x \in \mathbb{R}^d \setminus \omega_\varepsilon(\hat{x}), \\ \gamma, & \text{se } x \in \omega_\varepsilon(\hat{x}), \end{cases} \quad (1)$$

onde  $\gamma$  é uma constante real não negativa. Se  $\gamma = 0$ , a perturbação é dita singular e, portanto,  $\Omega_\varepsilon = \Omega \setminus \omega_\varepsilon(\hat{x})$  é um domínio topologicamente diferente do domínio inicial. Por outro lado, se  $\gamma > 0$  a perturbação é dita regular e não ocorre perturbação na topologia do domínio inicial. Em seguida, assume-se que um dado funcional de forma  $\psi(\chi_\varepsilon(\hat{x}))$  admite a seguinte expansão assintótica:

$$\psi(\chi_\varepsilon(\hat{x})) = \psi(\chi) + f(\varepsilon)D_T\psi(\hat{x}) + \mathcal{R}(\varepsilon), \quad (2)$$

onde  $f(\varepsilon)$  é uma função positiva que tende a zero quando o tamanho da perturbação tende a zero e  $\mathcal{R}(\varepsilon) = o(f(\varepsilon))$ . Por fim,  $D_T\psi(\hat{x})$  é denominada a derivada topológica do funcional  $\psi$  avaliada no ponto  $\hat{x}$ . Além disso, reescrevendo (2), obtém-se a seguinte definição clássica de derivada topológica Sokołowski & Żochowski (1999):

$$D_T\psi(\hat{x}) = \lim_{\varepsilon \rightarrow 0} \frac{\psi(\chi_\varepsilon(\hat{x})) - \psi(\chi)}{f(\varepsilon)}.$$

Para mais detalhes o leitor pode consultar Novotny & Sokołowski (2013). Agora, com o intuito de fixar as ideias acima, apresenta-se o seguinte exemplo.

**Exemplo 1.** Considere o seguinte funcional associado ao domínio não perturbado:

$$\psi(\chi) := \int_{\Omega} g(x) dx.$$

Suponha que  $\Omega$  seja sujeito a uma perturbação singular, ou seja,  $\gamma = 0$  em (1), e que  $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$  é Lipschitz contínua em  $\omega_\varepsilon(\hat{x})$ . Desta forma, o funcional associado ao domínio topologicamente perturbado é escrito como:

$$\psi(\chi_\varepsilon(\hat{x})) := \int_{\Omega_\varepsilon} g(x) dx. \quad (3)$$

Deseja-se obter uma expansão do tipo (2) para o funcional (3). Neste sentido, pode-se escrever o funcional acima da seguinte maneira:

$$\psi(\chi_\varepsilon(\hat{x})) = \int_{\Omega_\varepsilon} g(x) dx \pm \int_{\omega_\varepsilon} g(\hat{x}) dx = \psi(\chi) - |\omega_\varepsilon|g(\hat{x}) + \int_{\omega_\varepsilon} [g(\hat{x}) - g(x)] dx. \quad (4)$$

Comparando a expressão (4) com a expansão (2), é conveniente escolher  $f(\varepsilon) = |\omega_\varepsilon|$ . Logo, a derivada topológica é  $D_T\psi(\chi) = -g(\hat{x})$ . Basta então, mostrar que a última integral do lado direito de (4) é um resíduo. De fato, usando a desigualdade triangular e a hipótese sobre a função  $g$ , tem-se:

$$\left| \int_{\omega_\varepsilon} [g(\hat{x}) - g(x)] dx \right| \leq \int_{\omega_\varepsilon} c|\hat{x} - x| dx \leq c\varepsilon|\omega_\varepsilon| = o(|\omega_\varepsilon|).$$

### 3 OTIMIZAÇÃO TOPOLOGICA COM RESTRIÇÃO EM TENSÃO

A otimização topológica estrutural é um campo de pesquisa que vem sendo aplicado com sucesso em diferentes problemas da engenharia (o leitor interessado pode consultar Eschenauer & Olhoff (2001) para uma visão geral dos métodos de otimização topológica). Em geral, na otimização topológica, uma determinada quantidade de material é distribuída em um domínio inicial, sujeito a certas restrições e condições de contorno e, através de um processo numérico iterativo, minimiza-se (ou maximiza-se) uma função custo preestabelecida. Uma vez que a derivada topológica fornece uma direção de descida no processo de otimização, para um dado funcional de forma, neste trabalho utiliza-se tal ferramenta para obter a topologia ótima de estruturas elásticas e mecanismos flexíveis sujeitos a uma restrição na tensão de von Mises. Cabe ressaltar que, em contraste com os métodos de otimização topológica encontrados na literatura, a derivada topológica não requer um modelo material baseado em densidades intermediárias. Esta característica é crucial em problemas com restrição em tensão uma vez que esquemas de interpolação são desnecessários. Desta forma, as dificuldades decorrentes dos procedimentos baseados em densidades intermediárias são naturalmente evitados. Além disso, a derivada topológica tem a vantagem de fornecer uma fórmula analítica para a sensibilidade topológica que permite obter a topologia ótima em poucas iterações. Como podemos escrever o problema de otimização de estruturas como sendo um caso particular do de otimização de mecanismos flexíveis, começa-se por apresentar a formulação do problema para este último.

#### 3.1 Formulação do problema

Considere um domínio  $\mathcal{D} \subset \mathbb{R}^2$  aberto e limitado com fronteira Lipschitz  $\Gamma$  e um subdomínio  $\Omega \subset \mathcal{D}$ . A fronteira  $\Gamma$  é dividida em dois subconjuntos disjuntos  $\Gamma = \Gamma_D \cup \Gamma_N$ , onde  $\Gamma_N = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_0$  sendo  $\Gamma_{in}$ ,  $\Gamma_{out}$  e  $\Gamma_0$  mutuamente disjuntos. As trações de entrada, saída e nulas são prescritas sobre  $\Gamma_{in}$ ,  $\Gamma_{out}$  e  $\Gamma_0$ , respectivamente, enquanto os deslocamentos são prescritos sobre  $\Gamma_D$ . Então, dado o domínio  $\mathcal{D}$ , o problema de otimização consiste em encontrar um subdomínio  $\Omega \subset \mathcal{D}$  que satisfaça o seguinte problema de minimização:

$$\begin{cases} \underset{\Omega \subset \mathcal{D}}{\text{Minimizar}} \quad \mathcal{F}_\Omega(u) := \mathcal{J}(u) + \beta|\Omega| \\ \text{sujeito a} \quad \sigma_M(u) \leq \bar{\sigma}, \quad \text{q.t.p.} \quad \Omega^* \subset \Omega, \end{cases} \quad (5)$$

onde  $\Omega = \Omega^* \cup \omega \subset \mathcal{D}$ , com  $\omega$  usado para denotar a região de  $\Omega$  onde não é submetida à restrição em tensão. Em (5),  $\beta > 0$  é um parâmetro de penalização do volume,  $\mathcal{J}(u)$  é dado por

$$\mathcal{J}(u) := \int_{\Gamma_{in}} q_{in} \cdot u \, dx + \kappa \int_{\Gamma_{out}} q_{out} \cdot u \, dx, \quad (6)$$

onde as trações  $q_{in}$  e  $q_{out}$  e o escalar  $\kappa > 0$  são conhecidos. Além disso,  $\bar{\sigma}$  denota a tensão limite do material,  $\sigma_M$  representa a tensão efetiva de von Mises e  $u$  é solução do seguinte problema variacional: Encontrar  $u \in \mathcal{U} := \{\varphi \in H^1(\mathcal{D}; \mathbb{R}^2) : \varphi|_{\Gamma_D} = 0\}$  tal que

$$\int_{\mathcal{D}} \sigma(u) \cdot \nabla^s \eta = \int_{\Gamma_{in}} q_{in} \cdot \eta + \int_{\Gamma_{out}} q_{out} \cdot \eta \quad \forall \eta \in \mathcal{U}, \quad (7)$$

onde  $\sigma(u)$  é o tensor tensão de Cauchy e  $\nabla^s(\cdot)$  denota a parte simétrica do  $\nabla(\cdot)$ . A presença de uma restrição local em tensão dificulta tratar o problema de otimização diretamente. Aqui, será utilizada a abordagem proposta por Amstutz & Novotny (2010) que consiste em aproximar o problema de otimização original por meio de uma função de penalização regularizada para a restrição em tensão. Desta forma, o problema de otimização com restrição (5) pode ser aproximado pelo seguinte problema de otimização penalizado:

$$\underset{\Omega \subset \mathcal{D}}{\text{Minimize}} \mathcal{F}_{\Omega}^{\alpha}(u) := \mathcal{F}_{\Omega}(u) + \alpha \mathcal{G}(u), \quad (8)$$

em que  $\alpha > 0$  denota um dado coeficiente de penalização e  $\mathcal{G}(u)$  é escrito como Amstutz & Novotny (2010)

$$\mathcal{G}(u) := \int_{\Omega^*} \Phi(S_M^2(u)) \quad \text{sendo} \quad \Phi(t) \equiv \Phi_p(t) = [1 + (t)^p]^{1/p} - 1, \quad (9)$$

onde  $p \geq 1$  é um parâmetro real dado e  $S_M(u) = \sigma_M(u)/\bar{\sigma}$ .

### 3.2 Resultado da derivada topológica

A derivada topológica associada ao problema (8) com respeito à nucleação de uma pequena inclusão circular com propriedade material diferente do meio, é dada por Lopes & Novotny (2016):

$$D_T \mathcal{F}_{\Omega}^{\alpha}(x) = \beta D_T |\Omega|(x) + D_T \mathcal{J}(x) + \alpha D_T \mathcal{G}(x) \quad \forall x \in \mathcal{D},$$

em que a derivada topológica do volume e do funcional  $\mathcal{J}(u)$  escrevem-se como:

$$D_T |\Omega|(x) = \begin{cases} -1, & \text{se } x \in \Omega, \\ 1, & \text{se } x \in \mathcal{D} \setminus \overline{\Omega} \end{cases} \quad \text{e} \quad D_T \mathcal{J}(x) = -\mathbb{P}_{\gamma} \sigma(u(x)) \cdot \nabla^s v(x).$$

O contraste  $\gamma$  é tal que:  $\gamma = \rho_0$  em  $\Omega$  e  $\gamma = \rho_0^{-1}$  em  $\mathcal{D} \setminus \overline{\Omega}$ . O tensor de polarização  $\mathbb{P}_{\gamma}$  é dado por

$$\mathbb{P}_{\gamma} = \frac{1 - \gamma}{1 + \gamma a_2} \left( (1 + a_2) \mathbb{I} + \frac{1}{2} (a_1 - a_2) \frac{1 - \gamma}{1 + \gamma a_1} \mathbf{I} \otimes \mathbf{I} \right); \quad a_1 = \frac{\lambda + \mu}{\mu}, \quad a_2 = \frac{\lambda + 3\mu}{\lambda + \mu},$$

em que  $\mu$  e  $\lambda$  denotam os coeficientes de Lamé. E a derivada topológica do funcional de penalização  $\mathcal{G}(u)$  é dada por

$$\begin{aligned} D_T \mathcal{G}(x) = & - [\mathbb{P}_\gamma \sigma(u) \cdot \nabla^s v + (\gamma - 1) \mathbb{1}_{\Omega^*} \Phi'(S_M^2(u)) \mathbb{T}_\gamma \mathbb{B} S(u) \cdot S(u) + \rho_0 \mathbb{1}_{\Omega^*} \Upsilon(S(u)) \\ & + \mathbb{1}_{\Omega^*} \Phi'(S_M^2(u)) \frac{1}{4} \left( \frac{1-\gamma}{1+\gamma a_2} \right)^2 \left( 5(2S(u) \cdot S(u) - \text{tr}^2 S(u)) + 3 \left( \frac{1+\gamma a_2}{1+\gamma a_1} \right)^2 \text{tr}^2 S(u) \right) \\ & + \mathbb{1}_{\Omega^*} \Psi(S(u)) + (\gamma - 1) \mathbb{1}_{\Omega^*} \Phi(S_M^2(u))] (x), \end{aligned}$$

sendo  $S(u) = \sigma(u)/\bar{\sigma}$  e  $\mathbb{T}_\gamma := (\mathbb{P}_\gamma - \mathbb{I})/(1-\gamma)$ . Para os demais termos, o leitor pode consultar o trabalho Lopes & Novotny (2015).

**Observação 1.** No caso particular em que considera-se  $\kappa = 1$  em (6), o funcional  $\mathcal{J}(u)$  resulta no seguinte funcional de complacência (Amstutz, 2006):  $\mathcal{K}(u) := \langle q, u \rangle_{\Gamma_N}$ . Neste caso, a derivada topológica associada ao problema é escrita como Lopes et al. (2015)

$$D_T \mathcal{K}(x) = \mathbb{P}_\gamma \sigma(u(x)) \cdot \nabla^s u(x) \quad \forall x \in \mathcal{D}. \quad (10)$$

### 3.3 Exemplos numéricos

Em ambos os exemplos a seguir, toda a implementação computacional foi realizada no software MatLab2010 onde utilizou-se o Método de Elementos Finitos, em particular, Galerkin com discretização via elementos triangulares lineares.

Primeiramente, considera-se a otimização topológica de estruturas elásticas sujeitas a restrição em tensão (veja Observação 1). Em particular, deseja-se obter a topologia ótima para uma torre de alta tensão. Desta forma, o domínio inicial é dado conforme a Fig. 1(a) cujas dimensões estão em milímetros. Esta estrutura é modelada utilizando condição de simetria vertical e está submetida a um carregamento com distribuição linear cuja resultante é igual a  $-17000 \text{ N}$ .

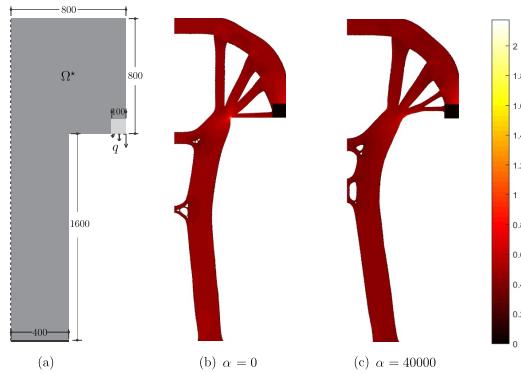
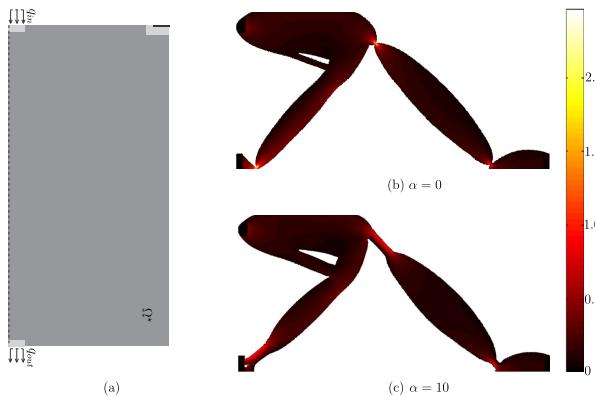


Figura 1: Domínio inicial e condições de contorno (a) e Distribuição de tensão ( $\sqrt{S_M}$ ) para os casos sem (b) e com (c) restrição em tensão.

Para o caso sem restrição em tensão ( $\alpha = 0$ ), Fig. 1(b), a tensão máxima ultrapassa a tensão limite do material devido à presença de singularidade no domínio de projeto. Por sua vez, o resultado para o caso com restrição em tensão ( $\alpha = 40000$ ) é apresentado na Fig. 1(c) com tensão máxima  $\max_{\Omega} (S_M) = 1.0952$ . Neste caso, observa-se que a tensão máxima obtida não

ultrapassa a tensão admissível uma vez que o canto presente no domínio inicial é arredondado eliminando a singularidade e suavizando a tensão local (veja Fig. 1(c)).



**Figura 2:** Domínio inicial e condições de contorno (a) e distribuição de tensão para os casos sem (b) e com (c) restrição em tensão.

Agora, lida-se com a otimização de mecanismos flexíveis com restrição em tensão. Objetiva-se aqui desenhar um inversor. O domínio inicial é representado por um retângulo  $100 \times 50 \text{ mm}^2$  preso no canto esquerdo de comprimento 5 mm, veja a Fig. 2. O inversor é modelado usando condição de simetria horizontal e está submetido a dois carregamentos uniformemente distribuídos:  $q_{in} = 10 \text{ N/mm}^2$  e  $q_{out} = 5 \text{ N/mm}^2$ . Finalmente, escolhe-se  $\beta = 0.02$  e  $\kappa = 10.0$ . Obseva-se que a tensão máxima associada ao caso sem restrição excede a tensão limite devido à presença de juntas móveis, como mostra a Fig. 2(b). Por outro lado, no caso com restrição ( $\alpha = 10.0$ ) a tensão máxima não ultrapassa a tensão limite e a topologia final não apresenta juntas, como pode ser observado na Fig. 2(c).

## 4 O PROBLEMA DE RECONSTRUÇÃO DE FONTES

Seja  $\Omega \subset \mathbb{R}^2$  um conjunto aberto e limitado com fronteira Lipschitz, denotada por  $\partial\Omega$ . Considere o subconjunto  $\Gamma_m \subseteq \partial\Omega$ , onde  $\Gamma_m$  tem medida de Lebesgue não-nula e  $\omega$  é aberto. Considere o seguinte problema de valor de contorno:

$$\left\{ \begin{array}{l} \mathcal{A}u = b^* \quad \text{em } \Omega, \\ u = u^* \quad \text{sobre } \Gamma_m, \\ -\partial_n u = q^* \quad \text{sobre } \Gamma_m. \end{array} \right. \quad (11)$$

O problema de reconstrução de fontes consiste em determinar a fonte  $b^*$  a partir de medições da função  $u$  tomadas sobre  $\Gamma_m$  ou dentro de um subconjunto de  $\Omega$ . Em outras palavras, a fim de determinar a fonte  $b^*$ , assume-se que as funções  $u^*$  e  $q^*$  são conhecidas sobre  $\Gamma_m$ . Dependendo da natureza do problema inverso, pode-se também assumir que apenas um dos dados de Cauchy é conhecido sobre  $\Gamma_m$  e que a função  $u$  seja conhecida dentro de um subconjunto aberto  $\omega \subseteq \Omega$ . Na equação (11),  $\mathcal{A}$  representa um operador linear genérico de segunda ordem. Para maiores detalhes sobre o estado da arte deste problema, veja Machado (2016).

## 4.1 Problema de otimização associado

Da maneira como está apresentado, o problema inverso (11) não possui solução única, ou seja, é possível determinar pelo menos duas fontes  $b_1^*$  e  $b_2^*$  que produzem os mesmos dados de Cauchy  $u^*$  e  $q^*$ . Logo, necessita-se restringir o espaço de busca da fonte  $b^*$  a fim de que o problema inverso possua solução única. Mesmo após recuperar a unicidade de solução, se o operador  $\mathcal{A}$  for compacto, então a solução do problema inverso não é estável, ou seja, não possui dependência contínua com relação aos dados de Cauchy.

A metodologia proposta para resolver o problema inverso (11) consiste em definir um problema de otimização equivalente e utilizar derivada topológica para minimizar o funcional associado. Logo, em vez de lidar com o problema inverso, que é mal posto, a estratégia proposta consiste em tratar um problema de otimização, que é direto e bem posto. Mais ainda, para evitar métodos clássicos de regularização, a ideia utilizada consiste em definir um conjunto de fontes admissíveis  $C(\Omega) \ni b^*$ , de modo que a fonte dada no problema de otimização também pertença a este conjunto. A definição do conjunto de fontes admissíveis depende das hipóteses consideradas acerca da fonte  $b^*$ . De maneira geral, este conjunto pode ser dado por:

$$C(\Omega) = \left\{ \varphi \in L^d(\Omega); \varphi(x) = \sum_{i=1}^M g_i(x) \right\}. \quad (12)$$

No **Caso 1**, considera-se que  $b^*$  é uma fonte constante por partes cujo suporte é a união de conjuntos estrelados  $\omega_i$ , o que implica em  $d = 2$  e  $g_i(x) = \gamma_i \mathbf{1}_{\omega_i}(x)$  (Canelas *et al.*, 2014, 2015). No **Caso 2**, assume-se que  $b^*$  é uma fonte concentrada o que acarreta em  $g_i(x) = \alpha_i \delta(x - x_i)$  e  $d = 1$  (Machado *et al.*, 2016). Para o caso particular do problema inverso do potencial, onde busca-se reconstruir a fonte a partir apenas dos dados de Cauchy, tanto para o Caso 1 quanto para o Caso 2, o funcional de forma do problema de otimização é definido da seguinte forma:

$$\mathcal{J}(u^D, u^N) = \|u^D - u^N\|_{L^2(\Omega)}. \quad (13)$$

A definição deste funcional baseia-se no critério introduzido em Kohn & Vogelius (1984). As funções  $u^D$  e  $u^N$  são soluções dos seguintes problemas auxiliares:

$$\begin{cases} \mathcal{A}u^D = b & \text{em } \Omega, \\ u^D = u^* & \text{sobre } \Gamma_m \end{cases} \quad \text{e} \quad \begin{cases} \mathcal{A}u^N = b + c & \text{em } \Omega, \\ -\partial_n u^N = q^* & \text{sobre } \Gamma_m, \\ \int_{\Omega} u^N dx = \int_{\Omega} u^D dx \end{cases} \quad (14)$$

Em ambos os problemas auxiliares, a fonte  $b \in C(\Omega)$  é conhecida. Tanto para o Caso 1 quanto para o Caso 2, é possível verificar que os problemas auxiliares são bem postos e que tanto  $u^D$  quanto  $u^N$  são quadrado integráveis em  $\Omega$  (ver Clason & Kunisch (2011) e Pieper & Vexler (2013)), o que garante que o funcional (13) está bem definido em ambos os casos.

## 4.2 Problemas perturbados

O próximo passo consiste em perturbar o termo fonte  $b$  dos problemas auxiliares e mensurar a perturbação gerada sobre o funcional (13). No Caso 1, a perturbação da fonte consiste em adicionar  $m$  funções constantes por partes com suportes circulares de raio  $\varepsilon_i$ . No Caso 2, a

perturbação ocorre adicionando  $m$  cargas puntuais à fonte original. Em ambos os casos, a fonte resultante de tais perturbações será denotada por  $b_\varepsilon$ , sendo denominada fonte perturbada. Observe que, em cada caso, a fonte perturbada continua pertencendo ao conjunto  $C(\Omega)$ , definido em (12). Os problemas provenientes da perturbação na fonte são definidos a seguir. O funcional toma a seguinte forma:

$$\mathcal{J}(u_\varepsilon^D, u_\varepsilon^N) = \|u_\varepsilon^D - u_\varepsilon^N\|_{L^2(\Omega)}. \quad (15)$$

As funções  $u_\varepsilon^D$  e  $u_\varepsilon^N$  são soluções dos seguintes problemas auxiliares:

$$\left\{ \begin{array}{l} \mathcal{A}u_\varepsilon^D = b_\varepsilon \text{ em } \Omega, \\ u_\varepsilon^D = u^* \text{ sobre } \Gamma_m \end{array} \right. \text{ e } \left\{ \begin{array}{l} \mathcal{A}u_\varepsilon^N = b_\varepsilon + c_\varepsilon \text{ em } \Omega, \\ -\partial_n u_\varepsilon^N = q^* \text{ sobre } \Gamma_m, \\ \int_\Omega u_\varepsilon^N dx = \int_\Omega u_\varepsilon^D dx \end{array} \right. \quad (16)$$

A partir da perturbação da fonte  $b$ , procura-se uma forma de relacionar as soluções dos problemas perturbados (16) às soluções dos problemas auxiliares (14). Levando em consideração a linearidade do operador  $\mathcal{A}$ , toma-se as seguintes expansões:

$$u_\varepsilon^D = u^D + \sum_{i=1}^m \alpha_i v_i \quad \text{e} \quad u_\varepsilon^N = u^N + \sum_{i=1}^m \alpha_i (v_i + h_i), \quad (17)$$

onde as funções  $v_i$  são soluções de problemas cujas fontes envolvem diretamente as perturbações, com condição de contorno Dirichlet homogênea e  $h_i$  são soluções de problemas regulares, com condição de contorno dependente de  $v_i$ . Deseja-se fazer a análise de sensibilidade do funcional  $\mathcal{J}(\cdot, \cdot)$  com relação aos parâmetros que definem os elementos de  $C(\Omega)$ . Avaliando a variação deste funcional, obtém-se:

$$\mathcal{J}(u_\delta^D, u_\delta^N) - \mathcal{J}(u^D, u^N) = 2 \int_\Omega \sum_{i=1}^m \rho_i h_i (u^D - u^N) - \int_\Omega \left( \sum_{i=1}^m \rho_i h_i \right)^2, \quad (18)$$

onde  $\rho_i = \pi \varepsilon_i^2$  no Caso 1 e  $\rho_i = \alpha_i$  no Caso 2. Observe que o lado direito da equação (18) é uma forma quadrática com relação a  $\rho_i$ . Logo, ao minimizar a variação do funcional, obtém-se o seguinte sistema:

$$H_{ij} \rho_i = d_j, \quad (19)$$

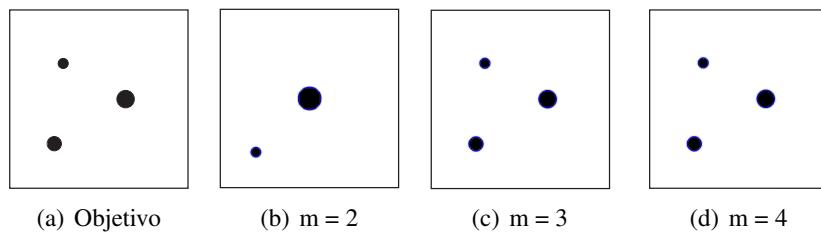
onde  $H_{ij} = (h_i, h_j)_{L^2(\Omega)}$  e  $d_j = (h_j, u^D - u^N)_{L^2(\Omega)}$ . Após calcular a solução de (19), realiza-se uma busca exaustiva e combinatória pelo menor valor da variação  $\mathcal{J}(u_\delta^D, u_\delta^N) - \mathcal{J}(u^D, u^N)$ . Esta busca é realizada sobre um conjunto  $X \subset \Omega$ , que representa uma discretização arbitrária de  $\Omega$ . O conjunto de pontos onde a variação atinge valor mínimo fornece automaticamente as localizações ótimas  $\xi^* \in X^m$  e as intensidades ótimas  $\alpha^* \in \mathbb{R}^m$ .

Cabe ressaltar que o problema de otimização não possui solução única. Para cada valor de  $m$ , é possível encontrar um par  $(\xi^*, \alpha^*)$ . Outra observação importante diz respeito ao método de obtenção dos parâmetros ótimos. Como se trata de um problema combinatório, a busca exaustiva torna-se rapidamente inviável quando  $n \gg m$  quando  $m$  aumenta de valor. Por exemplo, em testes experimentais usando o método proposto, a solução ótima não pôde ser eficientemente obtida considerando  $n = 400$  e  $4 < m \ll n$ . Desde que este é claramente um obstáculo do método, meios alternativos podem ser utilizados para sobrepor tal dificuldade. Em Machado *et al.* (2016), foi utilizada uma abordagem metaheurística e um procedimento de busca seletiva, considerando  $m$  com valor igual ou muito próximo ao valor de  $n$ .

## 4.3 RESULTADOS NUMÉRICOS

Nesta seção, serão apresentados alguns resultados numéricos referentes à resolução do sistema (19), ou seja, obtidos apenas para o caso em que o domínio de definição do problema inverso é limitado. Em todos os exemplos, o domínio  $\Omega$  é dado por um quadrado de lado  $L = 1$ , e utilizou-se elementos finitos triangulares lineares na resolução numérica. Nas figuras subsequentes, os círculos representam as cargas que constituem cada fonte, de modo que o centro indica o ponto onde a carga é aplicada e o raio é proporcional à intensidade.

O primeiro exemplo trata da reconstrução de uma fonte contendo três cargas puntuais com intensidades  $\alpha_1^* = 1.0$ ,  $\alpha_2^* = 2.0$  e  $\alpha_3^* = 3.0$ . Esta fonte está representada pela Fig. 3(a). Este exemplo serve para explicar a metodologia utilizada para determinar a quantidade de cargas. Já foi mencionado que para cada valor de  $m$ , o problema de otimização possui uma solução diferente. Desta forma, toma-se  $m$  de forma crescente até o momento em que se observa que uma das cargas possui intensidade desprezível.



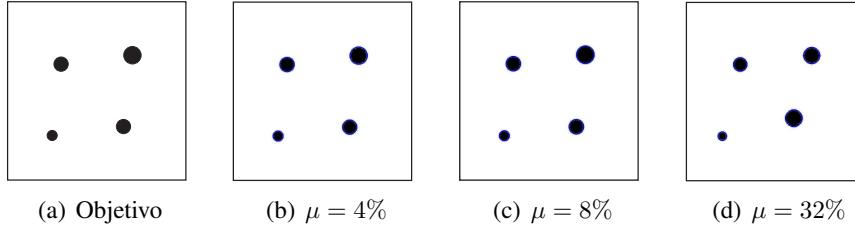
**Figura 3: Exemplo 1: Determinando a quantidade exata de fontes puntuais.**

As Figuras 3(b), 3(c) e 3(d) representam as soluções encontradas para  $m$  variando entre 2 e 4. Quando se toma  $m = 4$ , o vetor de intensidades ótimas é dado por  $\alpha^* = (2.0, 1.0, 3.0, 1.2342 \times 10^{-12})$ , não havendo sobreposição entre as localizações (o algoritmo não permite que isto ocorra). Pode-se concluir então que a fonte a ser reconstruída possui exatamente três cargas puntuais e que a solução correta é aquela apresentada na Fig. 3(c). Note ainda que a reconstrução é exata, ou seja, quando o valor de  $m$  coincide com o número de cargas procuradas, o problema de otimização possui a mesma solução que o problema inverso.

Visando verificar a robustez do método proposto, o próximo exemplo consiste na introdução de ruído no dado medido  $q^*$ . Isto é feito porque esta leitura foi obtida de maneira puramente sintética, o que pode mascarar eventuais instabilidades do método proposto. O ruído introduzido é do tipo Gaussiano branco, com média nula e diferentes valores de desvio-padrão. Para este exemplo, a fonte que se deseja reconstruir possui quatro cargas puntuais, localizadas nos pontos  $x_1^* = (0.20, 0.20)$ ,  $x_2^* = (-0.25, 0.15)$ ,  $x_3^* = (0.15, -0.25)$  e  $x_4^* = (-0.30, -0.25)$ , com intensidades  $\alpha_1^* = 3.0$ ,  $\alpha_2^* = \alpha_3^* = 2.0$  e  $\alpha_4^* = 1.0$ . A Figura 4(a) traz uma ilustração desta fonte. Os resultados obtidos são mostrados nas Figuras 4(b), 4(c) e 4(d), correspondentes a desvios-padrão de níveis 4%, 8% e 32%. O dado poluído com ruído, denotado por  $q_\mu^*$ , pode ser descrito pela seguinte fórmula:

$$q_\mu^* = q^*(1 + \mu\eta), \quad (20)$$

onde  $\eta$  é uma variável aleatória cuja função de densidade de probabilidade é dada por uma distribuição gaussiana e  $\mu \in \{4\%, 8\%, 32\%\}$  representa os possíveis desvios-padrão.



**Figura 4: Exemplo 2: Aumentando o nível de ruído sobre o dado lido.**

A Tabela 1 traz os resultados quantitativos da reconstrução ilustrada nas figuras acima. Note que, mesmo para um nível de ruído de 32%, as localizações são obtidas de maneira quase exata. Por outro lado, as intensidades são bem mais sensíveis e sofrem uma perturbação notável. Cabe destacar ainda que o ruído foi introduzido de maneira aleatória, sendo necessário fazer uma análise estatística para poder obter uma conclusão mais formal sobre a robustez do método.

**Tabela 1: Exemplo 2. Intensidades e localizações ótimas para diferentes níveis de ruído.**

	$\mu = 4\%$	$\mu = 8\%$	$\mu = 32\%$
$\xi_1^*$	(0.20, 0.20)	(0.20, 0.20)	(0.20, 0.20)
$\alpha_1^*$	3.0167 ( <b>+0.56%</b> )	3.0334 ( <b>+1.11%</b> )	2.9540 ( <b>-1.53%</b> )
$\xi_2^*$	(-0.25, 0.15)	(-0.25, 0.15)	(-0.25, 0.15)
$\alpha_2^*$	1.9771 ( <b>-1.15%</b> )	1.9543 ( <b>-2.29%</b> )	1.9306 ( <b>-3.47%</b> )
$\xi_3^*$	(0.15, -0.25)	(0.15, -0.25)	(0.10, -0.20)
$\alpha_3^*$	2.0036 ( <b>+0.18%</b> )	2.0072 ( <b>+0.36%</b> )	3.0452 ( <b>+52.26%</b> )
$\xi_4^*$	(-0.30, -0.25)	(-0.30, -0.25)	(-0.35, -0.25)
$\alpha_4^*$	0.9955 ( <b>-0.45%</b> )	0.9909 ( <b>-0.91%</b> )	0.6412 ( <b>-35.88%</b> )

## 5 CONCLUSÃO

Neste trabalho, a derivada topológica foi utilizada para tratar problemas inversos e de otimização topológica. Em particular, os problemas de reconstrução de fontes e de otimização de estruturas e mecanismos flexíveis foram abordados. Através da análise de sensibilidade topológica, foram obtidas formas analíticas para a derivada topológica, o que conduziu a resultados satisfatórios para ambos os problemas. No contexto de otimização topológica, a fórmula analítica para a sensibilidade topológica do funcional permitiu obter estruturas e mecanismos ótimos em poucas iterações e que atendem à restrição em tensão imposta. Com relação ao problema de reconstrução de fontes, a abordagem baseada na derivada topológica forneceu um método não iterativo de segunda ordem sem resíduo que, através de um processo de otimização combinatória, permitiu resolver o problema sem a necessidade de introduzir regularização do funcional de forma.

## 6 REFERÊNCIAS

- Amstutz, S. 2006. Sensitivity analysis with respect to a local perturbation of the material property. *Asymptotic Analysis*, **49**(1-2), 87–108.
- Amstutz, S., & Novotny, A. A. 2010. Topological optimization of structures subject to von Mises stress constraints. *Structural and Multidisciplinary Optimization*, **41**(3), 407–420.
- Amstutz, S., Novotny, A. A., & de Souza Neto, E. A. 2012. Topological derivative-based topology optimization of structures subject to Drucker-Prager stress constraints. *Computer Methods in Applied Mechanics and Engineering*, **233–236**, 123–136.
- Canelas, A., Laurain, A., & Novotny, A. A. 2014. A New Reconstruction Method for the Inverse Potential Problem. *Journal of Computational Physics*, **268**, 417–431.
- Canelas, A., Laurain, A., & Novotny, A. A. 2015. A new reconstruction method for the inverse source problem from partial boundary measurements. *Inverse Problems*, **31**(7), 075009.
- Clason, C., & Kunisch, K. 2011. A duality-based approach to elliptic control problems in non-reflexive Banach spaces. *ESAIM: Control, Optimisation and Calculus of Variations*, **17**(1), 243–266.
- Eschenauer, H. A., & Olhoff, N. 2001. Topology optimization of continuum structures: a review. *Applied Mechanics Reviews*, **54**(4), 331–390.
- Kohn, R., & Vogelius, M. 1984. Determining conductivity by boundary measurements. *Comm. Pure Appl. Math.*, **37**(3), 289–298.
- Lopes, C. G., & Novotny, A. A. 2015. Síntese de mecanismos flexíveis com restrição em tensão utilizando o conceito de derivada topológica. In: XXXV CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering.
- Lopes, C. G., & Novotny, A. A. 2016. Topology design of compliant mechanisms with stress constraints based on the topological derivative concept. *Structural and Multidisciplinary Optimization (to appear)*.
- Lopes, C. G., dos Santos, R. B., & Novotny, A. A. 2015. Topological derivative-based topology optimization of structures subject to multiple load-cases. *Latin American Journal of Solids and Structures*, **12**, 834–860.
- Machado, T. J. 2016. *Um Novo Método para Reconstrução de Fontes Concentradas*. Ph.D. Thesis, Laboratório Nacional de Computação Científica, Petrópolis, Brazil.
- Machado, T. J., Angelo, J. S., & Novotny, A. A. 2016. A new one-shot pointwise source reconstruction method. *Mathematical Methods in the Applied Sciences*, doi: **10.1002/mma.4059**.
- Novotny, A. A., & Sokołowski, J. 2013. *Topological derivatives in shape optimization*. Interaction of Mechanics and Mathematics. Springer-Verlag, Berlin, Heidelberg.
- Pieper, K., & Vexler, B. 2013. A priori error analysis for discretization of sparse elliptic optimal control problems in measure space. *SIAM Journal on Control and Optimization*, **51**(4), 2788–2808.
- Sokołowski, J., & Żochowski, A. 1999. On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, **37**(4), 1251–1272.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### DESENVOLVIMENTO DE UMA FERRAMENTA PARA GERAÇÃO AUTOMÁTICA DE UM BANCO DE DADOS COM BASE EM SISTEMA MECÂNICO SIMULADO PARA TREINAMENTO DE UMA DEEP NEURAL NETWORK-DNN

**Daniel de M. L. dos Santos**

**Francisco S. M. Sinfrônio**

**André B. Cavalcante**

**Allan K. Barros**

dantmec@gmail.com

kjvida@mac.com

abcborges@gmail.com

akduailibe@gmail.com

UFMA - Universidade federal do Maranhão

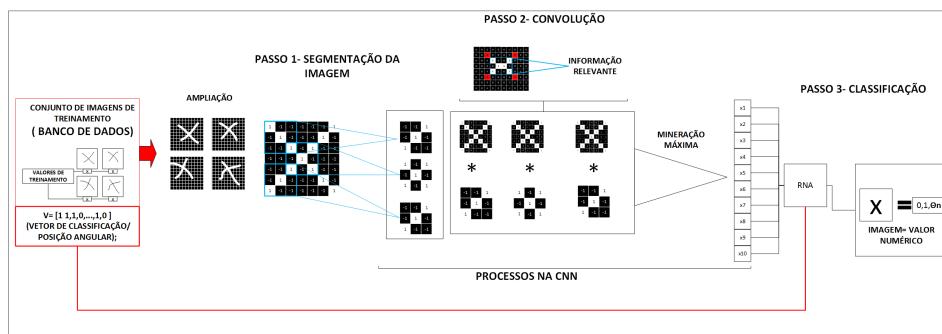
Av. dos Portugueses, 1966 - Bacanga, 65085-580, MA, São Luís, Brasil

**Resumo.** As Redes Neurais Profundas ou DNN (do inglês Deep Neural Networks) tem sido utilizadas como instrumento de extração de padrões, extração de características de imagens e classificação sem a necessidade de processamento prévio. Para o caso de Redes Neurais Convolucionais com aprendizado supervisionado, para finalidade de classificação, são necessários portanto a geração, tanto de um banco de dados prévio com milhares de imagens para o aprendizado da rede, como um vetor de valores desejados (valores de ângulo no presente trabalho), requerendo implementação de sensores e adaptações em sistemas mecânicos reais. Objetivo do presente estudo é a criação automática de um banco de dados de treinamento com base em uma maquete virtual, onde as variáveis angulares e variáveis de posicionamento são geradas com base em um modelo matemático, previamente desenvolvido. Por fim é realizada uma técnica conhecida como Chroma-Key, onde, para cada imagem, ocorre uma mudança de ambiente ou cenário onde estará inserido o sistema mecânico virtual.

**Palavras-chave:** Redes Neurais Profundas, Banco de Dados, Automatização, Modelo Virtual, Imagem.

# 1 INTRODUÇÃO

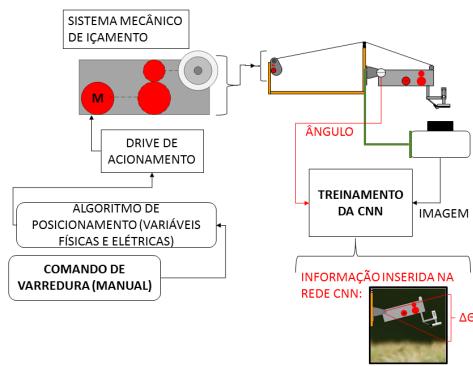
A extração de características de alta ordem para classificação de imagens, por redes neurais, sem a necessidade de processamento prévio, tem sido desenvolvida com a finalidade de aplicar uma ferramenta menos limitada à problemas considerados complexos (problemas onde ocorrem translações ou rotações do objeto a ser classificado na imagem, por exemplo). Redes Neurais Convolucionais (classe específica de Redes Neurais Profundas-DNN, voltada a processamento de imagens) são utilizadas com sucesso para classificação supervisionada e não-supervisionada de imagens como no trabalho de Xie et al. (2016). As redes Neurais Convolucionais foram concebidas posteriormente à análise dos resultados de Hubel. (1981) sobre a forma como o córtex visual de animais respondia a estímulos com orientação espacial variável. O processo de extração de característica(s) de um conjunto de imagens por uma Rede Neural Convolucional, pode ser visualizado na figura 1:



**Figura 1: Modelo de Treinamento de uma Rede Neural Convolucional.** Fonte: Autor.

A Rede Neural Convolucional segmenta a imagem (em escala de cinza ou colorida) através da convolução de uma pequena parte da matriz de pixels sobre toda a imagem, formando matrizes de menor ordem, convertendo-as em submatrizes (através da média dos valores convoluídos, por exemplo, dependendo da função de agrupamento). A subamostragem da imagem da camada anterior gera um mapa de características redundantes. Após esta etapa, características de alta ordem são agrupadas em um vetor de valores numéricos, os quais, serão tratados por uma estrutura usual de combinador linear com uma respectiva função de ativação, robusta ao problema de classificação (para o problema em estudo, a função softmax é utilizada como uma probabilidade normalizada).

Há necessidade portanto, de um vasto banco de dados para o treinamento das Redes Neurais Convolucionais (CNN), no caso em estudo, formado por aproximadamente 10.000 imagens associadas a valores de ângulo em diferentes ambientes. A complexidade da imagem ou o tamanho da mesma está diretamente relacionada ao número de camadas convolucionais necessárias, assim como ao número de dados de imagem para o treinamento (Matthew et al., 2013). Na figura 2 é observado o processo de aprendizado de posicionamento pela CNN no sistema mecânico em estudo:



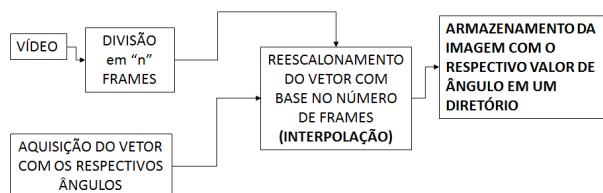
**Figura 2:** Modo de Treinamento da Rede Neural Convolucional, a ser implementado no sistema de posicionamento. **Fonte:** Autor.

A representação da posição angular (imagem vista no canto direito inferior da figura 2) e seu respectivo  $\Delta\theta$ , necessário ao posicionamento desejado, serão inseridos na entrada da CNN onde deverão ser extraídas características de inclinação e posteriormente comparação ao rótulo ( $\Delta\theta$ ) da respectiva imagem. O cálculo do ( $\Delta\theta$ ) é realizado através da medição feita pelo sensor de angulo visto na figura 2, além de parâmetros anteriormente inicializados (ex: considera-se o menor valor de ângulo medido pelo sensor no processo de treinamento como o ângulo desejado para o qual a sonda deverá ser posicionada).

## 2 DESCRIÇÃO DA FERRAMENTA-MÉTODO

O modelo da ferramenta tem como base de funcionamento a extração inicial de quadros ou imagens do vídeo (sensor de imagem com tecnologia *Complementary Metal Oxide System-CMOS*), com resolução aproximada de dois Mega Pixels- 2MP), juntamente com aquisição dos valores de ângulo, medidos por um textithardware de aquisição de dados baseado na plataforma Arduíno. Estes valores são posteriormente transferidos pela porta serial à um servidor (taxa de transferência de dados de cento e quinze mil e duzentos bits por segundo- 115.200 baud), digitalizados por um conversor analógico- digital com resolução de dez bits. A medida em que os valores são adquiridos ocorre sua concatenação e posterior reescalonamento com base no número de quadros (imagens).

A estrutura do algoritmo para geração automática do banco de dados, será mostrada na figura 3:



**Figura 3:** Ferramenta proposta para automatização do banco de dados de treinamento. **Fonte:** Autor.

A ferramenta (observada no fluxograma da figura 3) necessita de um vídeo em formato *Audio Video Interleave*- AVI do sistema de posicionamento em funcionamento (no presente trabalho foi gerado um vídeo com a operação de içamento), associado com os valores adquiridos pelo sensor de ângulo implementado. O modelo proposto acima tem como finalidade reduzir o tempo de implementação de uma rede neural convolucional em sistemas físicos reais, haja vista que o banco de dados será criado automaticamente. O reescalonamento do vetor é necessário para que haja bijeção entre o quadro (imagem) e o vetor de ângulo (mecanismo utilizado-interpolação).

## 2.1 Aquisição de sinais no modelo real

A aquisição do sinal de ângulo foi realizada através de um sensor potenciométrico com curva linear de resistência e um sensor de imagem CMOS, já citado anteriormente, cujos parâmetros estão organizados na Tabela 1:

**Tabela 1: Características principais do sensor potenciométrico utilizado na medição de ângulo no hardware de aquisição de sinais**

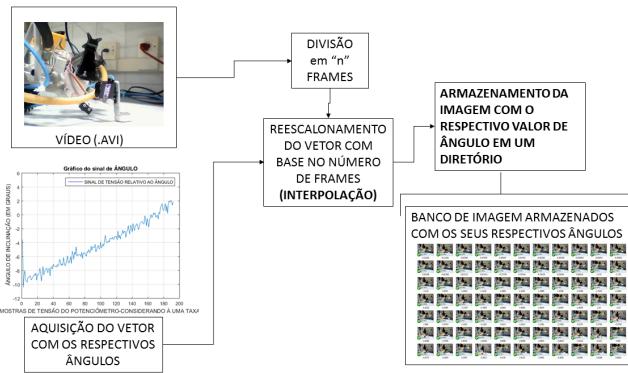
Característica	Unidade	Valor
Constante Angular	$K (Ohm/rad)$	2.3873e+03
Valor de Resistência	R (Ohms)	10e+03
Delta de Varredura	$\theta_{max}$ (rad)	4.1888
Material	Resistor de trilha de carbono	

O sensor de imagem, implementado na parte lateral direita da plataforma robótica (orientação da câmera voltada para o sistema de sondagem, como visto na figura 2), possui as características citadas na Tabela 2:

**Tabela 2: Características principais do sensor de imagem utilizado**

Característica	Unidade	Valor
Taxa de Quadros (utilizada no projeto)	(FPS)	25
Resolução	1600x1200 (pixels)	1920000.00

O vídeo adquirido pelo sensor CMOS, assim como o sinal do sensor potenciométrico descrito na tabela acima, foram extraídos sem nenhuma filtragem prévia, tentando manter o sinal mais próximo possível dos valores adquiridos em cada frame. Aplicados à ferramenta proposta, geraram um banco de dados com frames associados a valores de ângulo de inclinação. Esse processo de aquisição e geração automática do banco de dados no modelo real será apresentada na figura 4:



**Figura 4: Geração do banco de dados do sistema real de sondagem pela ferramenta proposta. Fonte: Autor.**

A aquisição do valor de inclinação no sistema de posicionamento real, absorve as imperfeições de construção e instalação do sensor potenciométrico utilizado (sensor de uso comercial), tornando ruidoso o sinal de ângulo de inclinação adquirido (sinal ruidoso visto graficamente no canto inferior esquerdo da figura 4). Houve então, a necessidade da montagem de um modelo animado de maior simplicidade quanto à forma e cenário de operação, com movimentos correspondentes a um modelo matemático previamente desenvolvido.

## 2.2 Método de interpolação

O reescalonamento do vetor com os valores de ângulo exigiu uma matriz quadrada, cuja primeira linha fosse formada por todo o vetor de ângulos medidos (tanto no modelo real, como no modelo virtual animado). Considerando  $n_\theta$  como sendo o número de elementos do vetor  $\vec{v}_\theta$  (vetor de posição angular que formará a primeira linha da matriz  $M_\theta$ ), é necessária uma matriz para o reescalonamento, inicialmente formada em sua primeira linha pelo vetor  $\vec{v}_\theta$ , tendo as outras  $n_\theta - 1$  linhas preenchidas por elementos nulos e,  $n_\theta$  colunas. A solução inicialmente adotada no projeto para a matriz quadrada  $M_\theta$  de ordem  $n_\theta$  está apresentada na "Eq. (1)":

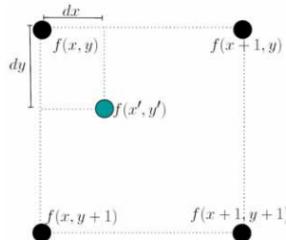
$$M_\theta = \begin{pmatrix} \theta_1 & \theta_2 & \dots & \theta_n \\ 0 & 0 & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (1)$$

Matriz  $M_\theta$

A partir do número de quadros extraídos do vídeo  $n_{frames}$  e da ordem da matriz quadrada  $M_\theta$ , o fator de escala  $R_e$  para nova matriz está inferido na "Eq. (2)":

$$R_e = \frac{n_{frames}}{n_\theta} \quad (2)$$

O método de interpolação utilizado no presente estudo é do tipo bilinear. Em geral esse método é utilizado em aplicações voltadas à reamostragem de imagens, com finalidade de minimização do efeito de serrilhamento ocasionado pelo método de interpolação do vizinho mais próximo ou "Nearest Neighbor"(Dourado.,2014). O método adotado realiza interpolações lineares horizontais e verticais entre os elementos da matriz de reescalonamento. Considerando uma imagem como uma  $f(x, y)$  na figura 5, será representada gráficamente a interpolação bilinear em uma matriz:



**Figura 5:** Esquema da interpolação bilinear aplicado à uma função  $f(x,y)$ . Fonte:(Dourado.,2014).

A equação que demonstra a interpolação bilinear:

$$f(x', y') = (1-d_x)(1-d_y)f(x, y) + dx(1-d_y)f(x+1, y) + (1-d_x)d_yf(x, y+1) + d_xd_yf(x+1, y+1) \quad (3)$$

Pode ser observado na "Equação (3)" um novo ponto  $(x', y')$  modelado pelos valores das linhas e colunas adjacentes, ponderado pelas distâncias ao redor do valor desconhecido. O método possui como característica a redução dos valores interpolados, tanto entre os valores das colunas (devido à interpolação com os elementos nulos da matriz  $M_\theta$ ), como entre os valores das linhas (a interpolação horizontal também é ponderada pelas distâncias verticais).

### 2.3 Chroma-Key

O Chroma-Key é uma técnica de edição de imagem desenvolvida com a finalidade de modificar o cenário ou ambiente virtual em que um objeto qualquer possa estar inserido, de acordo com critérios de cor previamente estabelecidos (Aprile et al., 2007). O sistema de cor das imagens inseridas na ferramenta é o sistema RGB (*Red, Green, Blue*- fazendo referência as matrizes ou planos de cor, contidos em cada imagem ou quadro adquirido). Para realizar a mudança de cenário, ou mudança de fundo, foram detectados os limiares de cor em cada plano. No caso em estudo, foi utilizado um fundo branco em um modelo virtual desenvolvido, desta forma, a parte do algoritmo que compõe a inicialização do limiar (responsável pela detecção de fundo) é um vetor  $\vec{v}_{limiar}$ , cuja forma é demonstrada na "Eq. (4)":

$$\vec{v}_{limiar} = [limiar_{red}, limiar_{green}, limiar_{blue}] \quad (4)$$

Os valores de limiar dos pixels  $[limiar_{red}, , ,]$ ,  $[ , limiar_{green}, ,]$  e  $[ , , limiar_{blue}]$  são inicializados com os respectivos valores de  $[200, 200, 200]$ , tendo como padrão para intensidade de cor o valor máximo de 255 (a cor "branca" ou de intercessão de espectro em uma imagem é

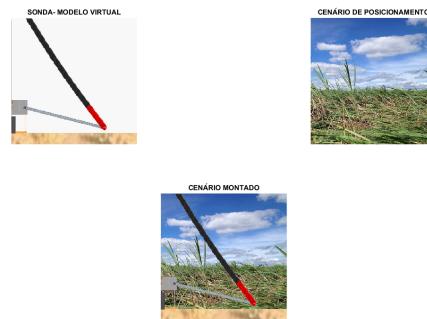
formada por um vetor  $\vec{v}$  com intensidades [255, 255, 255]). A forma de extração do objeto no fundo branco é mostrada na matriz formada pela seguinte função lógica:

$$M_{(red/green/blue)} = \begin{cases} 1 & \text{if } (\vec{\text{pixel}}[\mathbf{R}, \mathbf{G}, \mathbf{B}] - \vec{v}_{\text{limiar}}) \leq [55, 55, 55] \\ 0 & \text{if } (\vec{\text{pixel}}[\mathbf{R}, \mathbf{G}, \mathbf{B}] - \vec{v}_{\text{limiar}}) > [55, 55, 55] \end{cases} \quad (5)$$

A "Equação (5)" extrai a máscara onde será adicionado o cenário, sendo  $\vec{\text{pixel}}[\mathbf{R}, \mathbf{G}, \mathbf{B}]$  o valor do respectivo pixel (considerando um vetor) da imagem original em fundo branco varrido pela função lógica. Desta forma, para montagem do cenário ao fundo do objeto, recorre-se a equação "Eq. (6)":

$$I_{final(i,j)} = pixel_{red(i,j)}M_{red(i,j)} + pixel_{green(i,j)}M_{green(i,j)} + pixel_{blue(i,j)}M_{blue(i,j)} \quad (6)$$

Sendo  $M_{red(i,j)}$ ,  $M_{green(i,j)}$  e  $M_{blue(i,j)}$  os respectivos elementos da matrizes  $M_{red}$ ,  $M_{green}$  e  $M_{blue}$  resultantes da matrizes geradas pela função lógica  $M_{(red/green/blue)}$  referente à "Eq. (5)" e,  $pixel_{red(i,j)}$ ,  $pixel_{green(i,j)}$ ,  $pixel_{blue(i,j)}$  os respectivos pixels do cenário de fundo inseridos em cada plano de cor. Desta forma a imagem ficará conforme apresentado na figura 6:



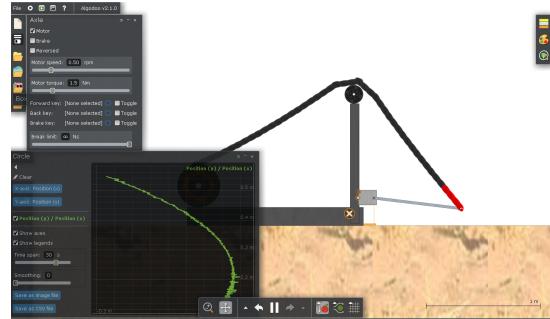
**Figura 6:** Demonstração das etapas do kroma-key aplicado a cada quadro. Fonte: Autor.

Pode ser visto no canto superior esquerdo da figura 6, o modelo virtual (cenário de fundo em cor "branca"), o cenário de fundo onde o objeto ou modelo será inserido (canto superior direito) e posteriormente a imagem ou quadro da sonda montada no cenário desejado (centro).

### 3 MODELO VIRTUAL PROPOSTO PARA COMPARAÇÃO

O modelo virtual simplificado foi desenvolvido no software *Algodoō®* (*software* livre de engenharia física, criado por Emil Ernerfeldt, nomeado inicialmente como "Phun"), liberado em 17 de Dezembro de 2007, sofrendo atualizações desde então. O modelo criado neste software leva em consideração valores reais da plataforma como massa, gravidade, resistência do ar, atrito, proximidade com os pontos de mudança de velocidade do mecanismo (caracterizados

tecnicamente como "Pontos Mortos" do mecanismo), tornando-o o mais próximo da realidade. As posições da extremidade da sonda são armazenadas em um arquivo ".csv", onde posteriormente serão extraídas pela ferramenta proposta e transformadas em valores angulares. O modelo virtual em simulação no software será apresentado na figura 7:



**Figura 7: Modelo virtual animado em plataforma virtual de desenvolvimento.** Fonte: Autor.

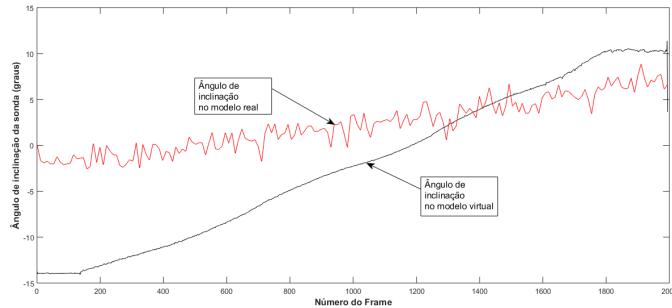
A equação que transforma os valores de posição do ponto de extremidade (circunferência vermelha na extremidade da sonda, vista na figura 7) é representada por:

$$\theta_{cc} = \sin^{-1}\left(\frac{y_{cc}}{\sqrt{x_{cc}^2 + y_{cc}^2}}\right) \quad (7)$$

Sendo  $\theta_{cc}$  o ângulo resultante calculado através das posições corrigidas  $x_{cc}, y_{cc}$  em relação aos pontos centrais da circunferência (deslocada no ponto de rótula), formada pelo sistema de posicionamento.

## 4 RESULTADOS ENCONTRADOS

Foram realizadas aquisições finais quanto aos valores de ângulo, em uma operação de içamento, no sistema de posicionamento real, assim como no modelo virtual. O gráfico dos vetores reescalados  $\vec{v}$  são apresentados na figura 8:

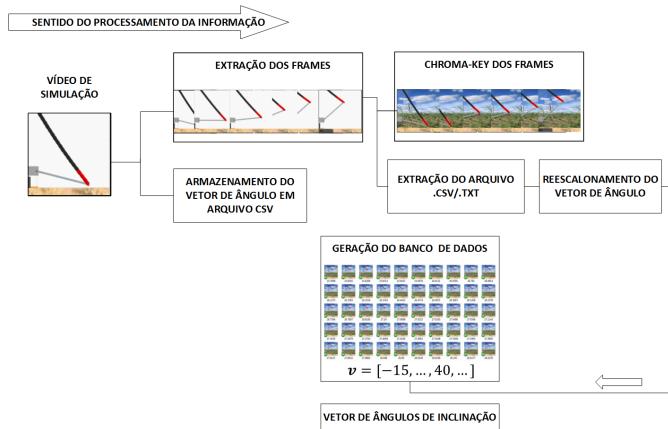


**Figura 8: Gráfico de comparação entre o vetor de aprendizado gerado no teste real e o vetor de aprendizado gerado no modelo virtual.** Fonte: Autor.

Pode ser observado no vetor de posição angular (adquirido pelo sensor no modelo real), as imperfeições construtivas e interferências na aquisição gerando um sinal ruidoso (linha vermelha no gráfico da figura 8), em comparação ao sinal adquirido no modelo virtual (linha preta no gráfico). Há também um aumento da faixa de deslocamento angular que, no modelo virtual, consegue alcançar um  $\Delta_\theta$  de aproximadamente  $25^\circ$ , em comparação à um  $\Delta_\theta$  de aproximadamente  $7^\circ$  no modelo real.

## 4.1 Contribuição Científica da Ferramenta

A ferramenta desenvolvida torna o uso de Redes Neurais Convolucionais mais robusto à problemas de engenharia e visão computacional. Isso ocorre através da geração automática do banco de dados de treinamento, onde seria necessária uma exaustiva aquisição de dados particular a cada problema. Na figura 9 a ferramenta está sendo aplicada em um modelo virtual:



**Figura 9: Geração do banco de dados do modelo virtual em cenário desejado.** Fonte: Autor.

Pode ser visto na figura 9 a geração automática do banco de dados no referido modelo computacional, podendo ser replicado em vários outros cenários (minimizando a possibilidade de overfitting na rede neural), reduzindo assim custos de implementação, ou melhorando o aprendizado de máquina em ambientes totalmente simulados.

## 4.2 Permissão

O autor Daniel de Matos Luna dos Santos é responsável pelas informações publicadas neste artigo.

## AGRADECIMENTOS

A Deus, ao Programa de Pós - Graduação em Engenharia Elétrica e ao Laboratório de Processamento da informação Biológica (PiB), que forneceram subsídios físicos e conhecimento na realização deste trabalho.

## REFERÊNCIAS

- Xie, J., L., Yang, Z.C.Song., & Wying, Hu., 2016. A Theory of Generative ConvNet- *Department of Statistics, University of California, Los Angeles, CA, USA.*
- Hubel, H.D., 1981. EVOLUTION OF IDEAS ON THE PRIMARY VISUAL CORTEX. *Harvard Medical School, Department of Neurobiology, Boston, Massachusetts, U.S.A.-Nobel lecture, 8 December 1981.*
- Matthew,D.,Zeiler,& Fergus, R., 2013. Visualizing and Understanding Convolutional Networks- *New York University.*
- Dourado,W.B., 2014. Avaliação de Técnicas de Interpolação de Imagens Digitais. *Universidade Estadual Paulista. Faculdade de Ciências e Tecnologia de Presidente Prudente Programa de Pós-Graduação em Matemática Aplicada e Computacional*
- Aprile,M., A. Colombari, A. Fusiello, V. Murino., 2007. SEGMENTATION AND TRACKING OF MULTIPLE OBJECTS IN VIDEO SEQUENCES. *Dipartimento di Informatica, University of Verona.*

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### Um Modelo para Avaliação de Desempenho de Plataformas de Processamento de Grafos

**Daniel N. R. da Silva**<sup>1</sup> (dramos@lncc.br)

**Klaus Wehmuth**<sup>1</sup> (klaus@lncc.br)

**Carla Osthoff**<sup>1</sup> (osthoff@lncc.br)

**Ana Paula Appel**<sup>2</sup> (apappel@br.ibm.com)

**Artur Ziviani**<sup>1</sup> (ziviani@lncc.br)

<sup>1</sup> Laboratório Nacional de Computação Científica (LNCC)  
Av. Getúlio Vargas, 333, Quitandinha  
CEP 25651-075 – Petrópolis, RJ – Brasil

<sup>2</sup> IBM Research Brazil  
R. Tutóia, 1157, Vila Mariana  
CEP 04007-900 – São Paulo, SP – Brasil

**Resumo.** *Redes complexas, i.e. grafos cujas características topológicas não são nem puramente aleatórias nem puramente regulares, estão presentes em diversas áreas do conhecimento, p. ex., biologia, economia, química, sociologia e tecnologia. Devido à relevância das análises desses grafos, muitas plataformas destinadas ao seu processamento têm surgido nos últimos anos. Entretanto, a grande quantidade dessas plataformas torna difícil a tarefa dos interessados em escolher aquela mais apropriada ao seu tipo de análise e rede. Soma-se ainda a essa dificuldade a dimensão que alguns desses grafos pode alcançar, o que pode tornar imprescindível o uso de processamento tanto paralelo quanto distribuído. Desse modo, compreender as implicações no desempenho computacional ao serem utilizadas diferentes abordagens no processamento de grafos é um tema de pesquisa pertinente. Neste trabalho, propõe-se um modelo de avaliação de desempenho de plataformas para processamento de grafos que visa atender quatro aspectos chave na compreensão do problema: algoritmos, arquiteturas, plataformas e redes.*

**Palavras-chave:** Avaliação de desempenho, processamento de grafos, redes complexas.

## 1 Introdução

Muitas estruturas e fenômenos podem ser descritos através do conceito de redes: entidades que estão ou não conectadas. A Teoria dos Grafos utiliza o conceito de grafo  $G$  — um par ordenado  $(V, E)$  no qual  $V$  é o conjunto de vértices (entidades) e  $E$  o de arestas (relacionamentos) — para modelar essas estruturas. Nesse cenário, redes cujos padrões de conectividade não são nem puramente regulares, tampouco exclusivamente aleatórios — ditas complexas — apresentam-se

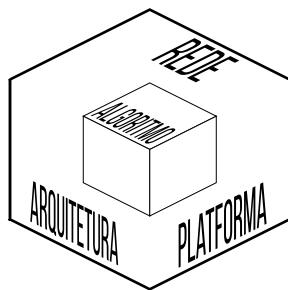
---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

em diversas áreas do conhecimento, p. ex., biologia, química, sociologia e tecnologia (Newman, 2010). A complexidade nos padrões de ligação e dificuldades relacionadas à escala que essas redes atualmente podem alcançar (até bilhões de componentes) impõem desafios ao seu processamento computacional eficiente (Lovász et al., 2009). Assim, estudos que avaliem o processamento de grafos de larga escala possuem relevância para academia e indústria.

O desafio de compreender o processamento de redes complexas de larga escala envolve o entendimento das relações e implicações de 4 aspectos: (i) **arquitetura**: quais implicações no processamento provém das tecnologias de *hardware* de determinado sistema computacional, p. ex., distribuição de memória, tecnologia de armazenamento e rede. (ii) **algoritmo**: que interferências no processamento provém dos tipos de análises e algoritmos. (iii) **plataforma**: em que influí as características de determinada plataforma de *software*, p. ex., estrutura de dados, modelo de programação e otimizações. (iv) **rede**: em que as características das redes, p. ex., grau de agrupamento, distribuição dos vértices e diâmetro, afetam substancialmente o desempenho do processamento. Pode-se perceber que as dimensões são inter-relacionadas, sendo não somente o desempenho final do processo dependente delas, mas elas próprias interdependentes (Fig. 1).



**Figura 1: 4 aspectos chave no processamento de grafos.**

A contribuição deste artigo é um modelo de avaliação de desempenho do processamento de grafos que ataca o desafio relatado nos 4 aspectos citados. É apresentada a metodologia de escolha de algoritmos, arquitetura, plataformas e redes, estabelecendo diretrizes para experimentos futuros.

## 2 Trabalhos Relacionados

Uma grande quantidade de plataformas para o processamento de grafos têm surgido no últimos anos. Em um trabalho recente, são revisadas mais de oitenta delas (Doekemeijer e Varbanescu, 2014). Devido à grande quantidade dessas plataformas, observa-se uma crescente na publicação de estudos de avaliação de desempenho de plataformas para o processamento de grafos (Fig. 2).

Dentre os estudos de avaliação de desempenho, foi proposta uma metodologia de avaliação que contempla os 4 aspectos citados na Seção 1 que culminou no *benchmark Graphalytics* (Iosup et al., 2016). Entretanto, no panorama de plataformas para grafos, grande parte dos estudos de avaliação de desempenho não contempla de modo diversificado os 4 aspectos do problema (Silva et al., 2016). Assim, é substancial a abordagem do domínio do processamento de grafos com maior diversidade a fim de que maior contribuição possa ser gerada aos interessados.

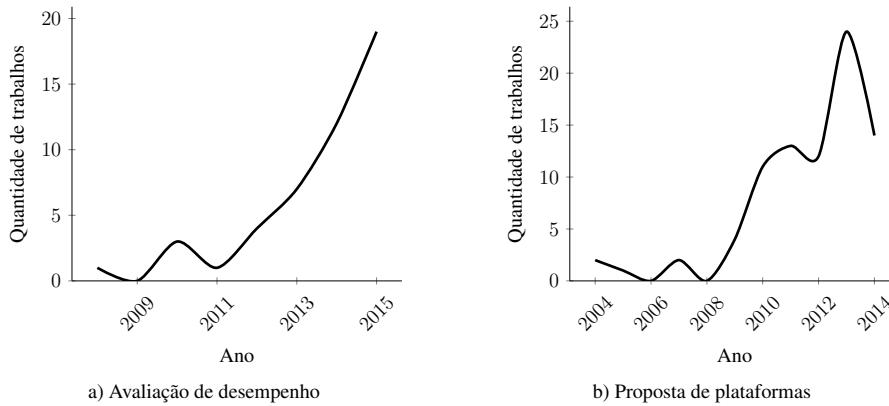


Figura 2: Crescimento na publicação de estudos de a) *avaliação de desempenho* e b) *proposta de plataformas* para processamento de grafos.

### 3 Proposta de Modelo

O objeto de estudo deste trabalho é o comportamento computacional da tarefa de processamento de grafos. Portanto, em uma visão simplificada, define-se tal objeto como um espaço 4-dimensional: **algoritmo, arquitetura, plataforma e rede**. Ao avaliar tal objeto de estudo é essencial que sejam escolhidos pontos que representem de maneira adequada cada aspecto, no intuito de que a análise não se dê de forma tendenciosa. Para esse fim, criteriosamente escolheu-se representantes de cada aspecto, conforme descrito posteriormente nesta seção, visando-se a diversidade de características, assim como, equilíbrio entre abstração e realidade.

Além disso, o presente trabalho lança mão à técnica de medição de desempenho em contraste à modelagem analítica e por simulação. Essa técnica foi escolhida uma vez que geralmente provê os melhores resultados (Lilja, 2005). Desse modo, dados os conjuntos de  $A_l$  (algoritmos),  $A_r$  (arquiteturas computacionais),  $P$  (plataformas) e  $R$  (redes), para cada execução de determinado algoritmo em uma plataforma em um ambiente para certa rede  $e_i \in E = A_l \times A_r \times P \times R$  avaliar-se-á cada  $m_j \in M$  (métricas). A técnica *Multi Way Analysis of Variance* é a ferramenta de avaliação estatística do conjunto de execuções. Os conjuntos serão descritos a seguir.

#### 3.1 Algoritmos

Um grande número de algoritmos para grafos têm sido implementados para as mais diversas análises, os quais segundo (Guo et al., 2014) podem ser divididos nas classes: *componentes conexos, detecção de comunidades, estatística, evolução de grafos, travessia e outros*. Além disso, esses algoritmos usualmente são utilizados em análises maiores que podem ser classificadas nas categorias: *caminho, centralidade, comunidade e conectividade*.

Nesse cenário, em (Kalavri, Vlassov e Haridi, 2016) são revisados estudos de modelos de programação e avaliação de desempenho de processamento de grafos, no quais os três algoritmos mais avaliados são *Page Rank*, Caminhos mínimos, Componentes conexos e variações desses três algoritmos. Portanto, baseando-se no princípio da diversidade de classes de algoritmos, em trabalhos relacionados de avaliação e na disponibilidade dos mesmos nas plataformas escolhidas para realização de experimentos em trabalhos futuros, foram escolhidos os algoritmos de Busca em Largura, Componentes Conexos, Contagem de Triângulos e *Page Rank* (Tab. 1).

**Tabela 1: Algoritmos escolhidos.**

Algoritmo	Tipo de análise	Categoria	CPU	Memória	Rede
Busca em Largura	Caminho	Travessia	Baixo	Baixo	Baixo
Componentes Conexos	Conectividade	Componentes conexos	Baixo	Médio	Médio
Contagem de triângulos	Comunidade	Estatística geral	Alto	Alto	Médio
Page Rank	Centralidade	Outros	Médio	Médio	Alto

**Busca em Largura:** É um algoritmo fundamental para construção de algoritmos mais complexos na análise de redes. Dado um vértice origem  $v \in V$ , esse algoritmo computa a distância de  $v$  aos demais vértices  $u \in V$ .

**Componentes Conexos:** É um algoritmo utilizado na análise de conectividade das redes. Tem por intuito encontrar os componentes conexos de um grafo  $G$ , isto é, subgrafos  $G_i \in G$ , de modo que exista um caminho  $p$  entre cada par de vértice  $u, v \in V(G_i)$ .

**Contagem de Triângulos:** Define-se um triângulo em um grafo  $G$  três vértices  $u \neq v \neq w \in V | (u, v) \in E \wedge (u, w) \in E \wedge (v, w) \in E$ . A quantidade de triângulos em  $G$  é uma estatística concernente ao grau de agrupamento do mesmo.

**Page Rank:** É um algoritmo utilizado em análises de centralidade que visar medir a importância de um nó baseado na importância de seus vizinhos. Esse algoritmo ganhou notoriedade tanto na academia quanto na indústria uma vez que o motor de busca do Google baseia-se nele.

### 3.2 Arquitetura

Segundo a disponibilidade de recursos dos autores, foi definida a realização dos experimentos em um *cluster* de dois nós computacionais, cada um com 2 Intel Xeon E5-2620V3 2.4GHz 6-core, 64GB DDR4 2133MHz, 3 HDD SAS 6Gb/s 4TB 7.2Krpm e Gigabit on-board. Uma vez que o ambiente é modesto, o critério de diversidade adotado se encontra na variação de núcleos de processamento (1, 6 e 12) e nós utilizados (1 e 2) para fins de experimentos de escalabilidade.

### 3.3 Métricas de Desempenho

As escolhas das métricas de avaliação de desempenho levaram em consideração os princípios de linearidade, confiabilidade, repetibilidade, facilidade de medição, consistência e independência (Lilja, 2005). Desse modo, foram selecionadas 17 métricas de desempenho divididas nas seguintes categorias: a) tempo de resposta, b) utilização, c) eficiência, e d) vazão (Tab. 2).

### 3.4 Plataformas

Um dos grandes desafios na avaliação de desempenho de plataformas destinadas ao processamento de grandes grafos é a escolha de um conjunto representativo das mesmas. Levando em consideração trabalhos na literatura que analisam o estado da arte desse domínio (Doekemeijer e Varbanescu, 2014; Elshawi et al., 2015; Pokorny, 2015; Yan et al., 2016), naqueles que propõem sistemas para o processamento de grafos, assim como nos que avaliam o desempenho dessas sistemas, foi escolhido um conjunto diversificado de plataformas (Tab. 3).

**Tabela 2: Métricas de Desempenho escolhidas.**

	Métrica	Descrição
<i>Tempo de Resposta</i>	Tempo de parede ( $TT$ )	Tempo gasto pela plataforma na visão do usuário.
	Tempo de algoritmo ( $TC$ )	Tempo gasto pela plataforma na execução do algoritmo objetivo.
	Tempo de carga ( $TR$ )	Tempo gasto pela plataforma na leitura do grafo do armazenamento secundário.
	Tempo de escrita ( $TW$ )	Tempo gasto pela plataforma na escrita da análise em armazenamento secundário.
	Tempo de sobrecusto ( $TO$ )	$TT - (TC + TR + TW)$
<i>Utilização</i>	Uso de CPU ( $RC$ )	Distribuição de percentual de CPU utilizada por segundo.
	Uso de memória ( $RM$ )	Distribuição de percentual de memória primária utilizada por segundo.
	Uso de rede ( $RN$ )	Distribuição de percentual de rede utilizada por segundo.
	Uso de disco ( $RD$ )	Distribuição de percentual de memória secundária utilizada por segundo.
	Eficiência energética ( $RE$ )	Mede a relação entre $RC, RM, RN, RD$ e $TT$ .
<i>Eficiência</i>	Escalonamento forte ( $SS$ )	Desempenho ao se manter tamanho de entrada constante enquanto aumenta-se o quantidade de processadores.
	Escalonamento fraco ( $WS$ )	Desempenho ao se manter tamanho de entrada por processador constante enquanto aumenta-se a quantidade de processadores.
	Escalonamento horizontal ( $HS$ )	Razão entre o número de nós computacionais e o tempo gasto pela plataforma.
	Escalonamento vertical ( $VS$ )	Razão entre o número de processadores e o tempo gasto pela plataforma.
	<i>Speedup</i> ( $S$ )	Variação na execução da plataforma ao se manter a entrada constante enquanto aumenta-se a quantidade de processadores.
	Eficiência paralela ( $E$ )	Razão entre o <i>Speedup</i> e quantidade de processadores ao se manter o tamanho de entrada constante.
<i>Vazão</i>	Arestas por segundo ( $EPS$ )	$ E /TT$ .
	Vértices por segundo ( $VPS$ )	$ V /TT$ .

**Tabela 3: Plataformas escolhidas.**

Plataforma	Breve descrição e principais características
Apache Giraph	<i>Framework</i> para o processamento de grafos baseado no paradigma centrado em vértice construído em cima do Apache Hadoop. Paradigma de programação centrado em vértice, implementado em Java, suporta computação em memória externa, sistema para memória distribuída, código livre.
Chaos	Sistema de processamento de grafos projetado para análise de grandes grafos utilizando <i>clusters</i> pequenos. Paradigma de programação centrado em aresta, implementado em C++, suporta computação em memória externa, sistema para memória distribuída, código livre.
GraphLab Create 2.0	<i>Framework</i> extensível para análises que utilizam aprendizagem de máquina (possui um <i>toolkit</i> para análise de grafos) que evoluiu a partir de PowerGraph. Paradigma de programação GAS ( <i>gather, apply and scatter</i> ), implementado em C++ com ligações para Python, sistema para memória distribuída, proprietário.
GraphMat	<i>Framework</i> para o processamento de grafos baseado na tradução do paradigma orientado a vértice em multiplicação generalizada entre matriz esparsa e vetor denso. Implementado em C++, sistema para memória compartilhada e distribuída, código livre.
GraphX	Uma API para processamento de grafos do <i>framework</i> Apache Spark. Paradigma de programação centrado em vértice, sistema para memória distribuída, código livre.
GridGraph	Sistema para o processamento de grandes grafos em apenas uma máquina baseado na nova representação de grafos em grade. Modelo de processamento <i>Streaming - Apply</i> , suporta processamento em disco, implementado em C++, código livre.
NetworKit	Biblioteca de alto desempenho destinada à análise de redes complexas de larga escala. Implementada em C++ com <i>bindings</i> para Python, sistema para memória compartilhada, código livre.

### 3.5 Redes

Definiu-se um conjunto de dados de 28 redes, sendo 14 traços de redes reais e 14 redes sintéticas. Ao mesclar dados reais e sintéticos, o poder de análise cresce já que se tem maior equilíbrio entre controle de parâmetros (dados sintéticos) e representatividade (dados reais).

**Tabela 4: Mapeamento do conjunto de dados em classes com a escala**  $s(V, E) = \log(|V| + |E|)$ .

Escala	< 7	[7, 7.5)	[7.5, 8)	[8, 8.5)	[8.5, 9)	[9, 9.5)	$\geq 9.5$
Classe	2XS	XS	S	M	L	XL	2XL

**Dimensão das redes:** Em (Iosup et al., 2016) define-se a escala de um grafo  $G$  como  $s(V, E) = \log(|V| + |E|)$ , mapeando-se  $s$  em 7 categorias (Tab. 4). Para cada categoria foram escolhidos 2 traços de redes reais e gerados 2 grafos sintéticos segundo os critérios a seguir.

**Traços de redes reais:** Coletoou-se um grande conjunto de redes dos repositórios<sup>1</sup> efetuando-se pré processamento através da: remoção de redes duplicadas, mantida apenas a topologia da rede, converteu-se redes direcionados em não direcionados, redes com  $s < 6$  e  $s > 10$  foram desconsideradas, e efetuada computação de estatísticas básicas das redes: a) número de vértices e arestas b) vértices de grau mais alto e mais baixo, c) graus menor, maior e médio, d) contagem de padrões de garra e cunha, e) exponente de lei de potência e f) Vetor de características obtido através da aplicação do método *Degree Distribution Quantification and Comparison* (Aliakbary, Habibi e Movaghbar, 2015). Para cada classe em  $s$  foi criado um espaço de características com as estatísticas sendo escolhidas 2 redes mais distantes entre si pela distância de Manhattan.

**Redes sintéticas:** Para geração das redes sintéticas, adotou-se um modelo generativo de rede pertencente à família RHG (*Random Hyperbolic Graphs*) (Krioukov et al., 2010), o TRHG (*Threshold Random Hyperbolic Graphs*) (Looz et al., 2016). A escolha desse modelo baseou-se no fato de que a família de Grafos Aleatórios Hiperbólicos captura características presentes em redes reais, tais como, distribuição de grau seguindo uma lei de potência e forte efeito de comunidade. Esse modelo permite a manipulação de três parâmetros  $n$  (número de vértices),  $\gamma$  (exponente de lei de potência relacionado a distribuição do grau dos vértices) e  $k$  (grau médio esperado). Para  $n$  optou-se pelos valores  $10 \times 10^4, 10 \times 10^5, 20 \times 10^5, 10 \times 10^6, 20 \times 10^6, 10 \times 10^7, 20 \times 10^7$ . Além disso, para todas as redes utilizou-se o grau médio  $k = 25$ . Em relação a  $\gamma$  adotou-se 2 valores 2.2 e  $\infty$ . Sendo que para  $\infty$ , como TRHG degenera para o modelo de grafo aleatório Erdos Renyii (ER) (Erdos e Renyi, 1959) segundo (Aldecoa, Orsini e Krioukov, 2015), foi adotado o modelo ER com o parâmetro  $p$  ajustado para se ter  $k = 25$ .

## 4 Considerações finais

Este artigo define um modelo de avaliação de desempenho para plataformas de processamento de grafos considerando quatro aspectos chave: algoritmos, arquiteturas, plataformas e redes. Ao abordar esse problema atentando para os aspectos citados, assim como a escolha criteriosa dos representantes de cada um deles, leva-se em conta múltiplas perspectivas na análise em contraste com as alternativas da literatura que tipicamente focam em uma análise sob uma única perspectiva. Como trabalho futuro, pretendemos realizar uma série de experimentos computacionais com base no modelo proposto, culminando em uma análise multi-perspectiva do desempenho de plataformas de processamento de grafo, delineando assim diretrizes para a seleção da plataforma mais adequada conforme a análise pretendida e o conjunto de redes complexas a ser analisado.

**Agradecimentos:** Este trabalho foi parcialmente financiado pelo CNPq.

<sup>1</sup>9th DIMACS, Laboratory for Web Algorithms, Microsoft Academic Graph, Network Data Repository, Stanford Large Network Dataset Collection, The Koblenz Network Collection, The University of Florida Sparse Matrix Collection

## REFERÊNCIAS

- Aldecoa, R., C. Orsini e D. Krioukov (2015). “Hyperbolic graph generator”. Em: *Computer Physics Communications* 196, pp. 492–496.
- Aliakbary, Sadegh, Jafar Habibi e Ali Movaghar (2015). “Feature extraction from degree distribution for comparison and analysis of complex networks”. Em: *The Computer Journal*, bxv007.
- Doekemeijer, N. e A.L. Varbanescu (2014). *A survey of parallel graph processing frameworks*. Rel. téc. Delft University of Technology.
- Elshawi, Radwa et al. (2015). “Big Graph Processing Systems: State-of-the-Art and Open Challenges”. Em: *Proc. of the IEEE Int. Conf. on Big Data Computing Service and Applications (BigDataService)*, pp. 24–33.
- Erdos, P e A Renyi (1959). “On random graphs I”. Em: *Publ. Math. Debrecen* 6, pp. 290–297.
- Guo, Y. et al. (2014). “How Well Do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis”. Em: *Proc. of the IEEE Int. Parallel and Distributed Processing Symp. (IPDS)*, pp. 395–404.
- Iosup, A. et al. (2016). “LDBC Graphalytics: A Benchmark for Large-scale Graph Analysis on Parallel and Distributed Platforms”. Em: *Proc. of the VLDB Endow.* 9.13, pp. 1317–1328.
- Kalavri, V., V. Vlassov e S. Haridi (2016). “High-Level Programming Abstractions for Distributed Graph Processing”. Em: *arXiv preprint arXiv:1607.02646*.
- Krioukov, D. et al. (2010). “Hyperbolic geometry of complex networks”. Em: *Phys. Rev. E* 82 (3), p. 036106.
- Lilja, D. J. (2005). *Measuring computer performance: a practitioner’s guide*. Cambridge University Press.
- Looz, M. von et al. (2016). “Generating massive complex networks with hyperbolic geometry faster in practice”. Em: *arXiv preprint arXiv:1606.09481*.
- Lovász, L. et al. (2009). “Very large graphs”. Em: *Current Developments in Mathematics* 2008, pp. 67–128.
- Newman, M. (2010). *Networks: An introduction*. Oxford University Press.
- Pokorny, J. (2015). “Graph Databases: Their Power and Limitations”. Em: *IFIP Int. Conf. on Computer Information Systems and Industrial Management*. Springer, pp. 58–69.
- Silva, D. N. R. da et al. (2016). “Análise de Desempenho de Plataformas de Processamento de Grafos”. Em: *Simpósio Brasileiro de Bancos de Dados*.
- Yan, D. et al. (2016). “Big Graph Analytics Systems”. Em: *Proc. of the Int. Conf. on Management of Data (ACM SIGMOD)*. New York, NY, USA: ACM, pp. 2241–2243.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### AVALIAÇÃO DE FUNÇÕES EMPÍRICAS DE SCORING EM EXPERIMENTOS DE TRIAGEM VIRTUAL EM LARGA ESCALA

**Felipe S. S. Pereira**

**Isabella A. Guedes**

**Laurent E. Dardenne**

[felipesp@lncc.br](mailto:felipesp@lncc.br)

[isabella@lncc.br](mailto:isabella@lncc.br)

[dardenne@lncc.br](mailto:dardenne@lncc.br)

Grupo de Modelagem Molecular de Sistemas Biológicos, Laboratório Nacional de Computação Científica, LNCC/MCTIC

Av. Getúlio Vargas, 333, Quitandinha, 25651-075, Rio de Janeiro, Petrópolis, Brasil

**Resumo.** Considerando os promissores resultados na predição de modos de ligação proteína-ligante previamente obtidos pelo programa DockThor, objetiva-se no presente momento o desenvolvimento de novas funções empíricas de scoring para predição da afinidade de ligação proteína-ligante para estudos de triagem virtual em larga escala. Objetivamos neste trabalho, avaliar a preditividade e a classificação de ligantes das funções de scoring recentemente desenvolvidas em nosso grupo de pesquisa. Foram analisadas três proteínas-alvo selecionadas a partir do DUD-E, pertencentes a distintas classes proteicas, associadas a conjuntos de ligantes ativos e decoys (i.e. ligantes de menor afinidade que ativos). Buscamos analisar os resultados das predições utilizando diferentes funções empíricas: (1) genérica (i.e., sem especificidade quanto à classe de proteína ou resultados de docking); (2) classe-proteína específica e (3) derivada de resultados de docking obtidos com o programa DockThor. Os valores das áreas sob as curvas ROC obtidos nas análises (maiores que 0,6) demonstram o potencial das funções e nos guiam para o aprimoramento das mesmas, objetivando a obtenção de um classificador confiável para experimentos de triagem virtual em busca da correta distinção de ligantes a partir das afinidades de ligação preditas.

**Palavras-chave:** Triagem virtual, Funções empíricas de afinidade receptor-ligante, docking

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUÇÃO

A metodologia de *docking* molecular consiste basicamente de: (1) algoritmo de busca, que tem como finalidade a exploração da superfície de energia livre de um complexo receptor-ligante almejando encontrar os modos de ligação mais favoráveis, e (2) funções de *scoring* que objetivam realizar a predição da afinidade de ligação, tanto para ranquear os ligantes de acordo com a afinidade predita quanto classificá-los como ativos ou inativos (Guedes et al., 2014). Dado um conjunto de ligantes, a correta distinção entre ativos e inativos é questão crucial na busca por compostos químicos de interesse farmacológico.

O programa DockThor mostra-se competitivo em estudos comparativos com distintos programas de atracamento molecular, obtendo resultados promissores no que se refere à predição de modos de ligação (de Magalhães et al., 2014). Recentemente, foram desenvolvidas em nosso grupo de pesquisa novas funções empíricas de *scoring* para predição da afinidade de ligação proteína-ligante, baseadas na função de *scoring* original do DockThor (i.e. campo de força MMFF94S) (Guedes, 2016). Essas funções empíricas buscam predizer a energia livre de ligação a partir de uma soma ponderada de termos baseados em propriedades físicas, onde os coeficientes associados aos termos são obtidos através de regressão linear, usando como referência dados de afinidade previamente elucidados experimentalmente (Liu & Wang, 2015).

Experimentos computacionais que abordam um grande número de ligantes (e.g. bancos de dados de compostos químicos) com o objetivo de selecionar ligantes potencialmente ativos no âmbito farmacológico, são denominados experimentos de triagem virtual (do inglês, *virtual screening*), complementares aos estudos experimentais de triagem de novos fármacos. Para validação de ferramentas que atacam o problema supracitado, faz-se necessária a consideração do cenário real de estudos de triagem virtual no planejamento racional de fármacos, onde nos deparamos com uma enorme quantidade de compostos químicos, sendo estes em sua maioria indesejados para um dado estudo (i.e. apresentam baixa afinidade de ligação, logo denominados inativos). Para essa validação são disponibilizados conjuntos de dados que buscam reproduzir esse cenário real, i.e. bancos de dados constituídos por um grande número de ligantes associados a uma lista de proteínas-alvo com importância farmacológica. Neste sentido, é de suma importância avaliar a capacidade de nossas funções empíricas com relação a sua confiabilidade na classificação dos compostos químicos (i.e. ligantes) em ativos ou não.

## 2 OBJETIVOS

O presente trabalho objetivou realizar a avaliação de funções empíricas de *scoring* receptor-ligante em estudos de triagem virtual de ligantes em larga escala. O objetivo principal é avaliar a distinção entre ativos e *decoys* dada pelo ranqueamento dos ligantes através das funções de *scoring*, desde que a correta classificação dos ligantes é necessária para o êxito em experimentos de triagem virtual de fármacos. Desde que funções de *scoring* podem apresentar comportamento variável dada a classe da proteína em estudo, buscamos analisar os resultados utilizando diferentes funções empíricas: (1) genérica (i.e., sem especificidade quanto à classe de proteína ou programa de *docking*); (2) classe-proteína específica e (3) específica para resultados de *docking* obtidos com o programa DockThor. A avaliação das funções empíricas se dá em um contexto onde se considera grandes conjuntos de ligantes, e a consequente classificação dos mesmos em ativos ou *decoys*.

### 3 METODOLOGIA

Foi selecionado o DUD-E (*Directory of Useful Decoys: Enhanced*) como conjunto de dados para o *benchmarking* dos experimentos de triagem virtual. O conjunto supracitado constitui-se de uma versão reestruturada do DUD (*Directory of Useful Decoys*), publicada no ano de 2006 (Huang et al., 2006). Nesta reestruturação foi desenvolvida uma base de dados com novas funcionalidades e disponibilizada como ferramenta para avaliação e otimização de programas de *docking* molecular, bem como de funções de *scoring*. O DUD-E é constituído por ligantes que possuem dados de afinidade experimentais previamente recuperados a partir da literatura e *decoys*. A classificação adotada para os ligantes, compreende a denominação de ligantes ativos aqueles que possuem valor de afinidade de ligação obtida experimentalmente abaixo de  $1\mu\text{M}$ , ligantes inativos aqueles que apresentam afinidade de ligação experimental acima de  $30\mu\text{M}$ , ligantes marginais aqueles que apresentam afinidade de ligação experimental entre  $1\mu\text{M}$  e  $30\mu\text{M}$  e *decoys*, ligantes que possuem propriedades físico-químicas similares aos ligantes ativos, porém são topologicamente dissimilares. A adoção da estratégia citada para a adição de *decoys* não experimentais (i.e. sem dados de inibição conhecidos) no conjunto, ocorre devido ao pequeno número de ligantes inativos reportados (i.e. *decoys* experimentais) (Mysinger et al., 2012).

Selecionamos proteínas-alvo do DUD-E representantes de três classes proteicas distintas descritas na literatura com potencial farmacológico : (1) Quinase Dependente de Ciclina 2 (CDK2) pertencente a classe das quinases; (2) Receptor Beta-1 de Hormônio Tireoidiano (THB) pertencente a classe de receptores nucleares e (3) Tripsina (TRY1) pertencente a classe das proteases. Foram utilizadas as estruturas dos respectivos alvos descritas no DUD-E, obtidas a partir do *Protein Data Bank*. Encontra-se na Tabela 1 os identificadores das estruturas e os números correspondentes aos ligantes e *decoys* utilizados em cada caso. Devido à pequena quantidade de ligantes inativos (i.e. ligantes com dados de afinidade experimentais reportados na literatura) relacionados aos alvos selecionados para o presente trabalho, apenas o conjunto de *decoys* foi utilizado nos experimentos de triagem virtual. A pequena quantidade de ligantes inativos disponíveis corrobora que dados de afinidade negativos raramente são reportados, como descrito pelos autores do DUD-E (Mysinger et al., 2012).

**Tabela 1: Estruturas proteicas e as respectivas quantidades de ligantes**

Proteína-Alvo	Código PDB	Ativos	Decoys
<b>CDK2 - Quinase Dependente de Ciclina 2</b>	1H00	443	27.830
<b>THB - Receptor Beta-1 de Hormônio Tireoidiano</b>	1Q4X	40	7.652
<b>TRY1 - Tripsina</b>	2AYW	449	25.367

Os experimentos de *docking* para predição do modo de ligação dos ligantes selecionados foram obtidos através do programa DockThor, que utiliza a abordagem de *docking* ligante flexível/receptor rígido, onde todos os graus de liberdade do ligante (translacionais, rotacionais e conformacionais) são explorados e a proteína é considerada rígida. Para encontrar a solução ótima dentre as possíveis conformações de um ligante junto ao sítio ativo de uma proteína-alvo, é utilizado um algoritmo genético (AG) de múltiplos mínimos como método de busca. A função de avaliação de energia é baseada no campo de força clássico MMFF94 (Halgren, 1999).

O DockThor utiliza a metodologia baseada em grade, onde o cálculo da função de energia é realizado a partir de uma malha construída de acordo com o centro do sítio ativo da proteína-alvo de interesse (de Magalhães et al., 2014). Dada a característica do AG do DockThor gerar múltiplas soluções para os modos de ligação dos ligantes, é realizada um agrupamento e ranqueamento para a seleção do modo de ligação de melhor energia observado para cada ligante. Os parâmetros adotados para a execução do programa DockThor consistiram em 24 rodadas do AG, tamanho da população inicial equivalente a 1.000, número de avaliações do AG equivalente a 1.000.000. As dimensões da grade foram definidas em 20 Å para cada coordenada (x, y, z), com discretização de 0,25 Å. Com exceção do número de rodadas do AG, todos os demais parâmetros supracitados são os valores padrões atuais do programa. Experimentos de *redocking* foram realizados a fim de se validar o protocolo de *docking* adotado para o estudo, em cada proteína-alvo. Avaliou-se a partir de um alinhamento estrutural entre a conformação experimental do ligante e a pose predita pelo *docking*, a similaridade entre os modos de ligação utilizando a métrica do RMSD (*Root Mean Square Deviation*), adotando um limiar de 2,0 Å.

Os cálculos do DockThor foram executados no supercomputador Santos Dumont, sistema de computação petaflópica localizado no Laboratório Nacional de Computação Científica (LNCC), nó central (Tier-0) do Sistema Nacional de Processamento de Alto Desempenho (SINAPAD). No presente trabalho, foram realizadas 61.781 execuções do algoritmo para o cálculo dos *dockings* moleculares. Cada execução do algoritmo da versão atual do DockThor leva em média 5 minutos, obtemos 308.405 minutos de cpu/core, equivalente a aproximadamente 214 dias de cpu/core. Estes valores justificam a necessidade de um ambiente de computação de alto desempenho em estudos de triagem virtual de ligantes em larga escala.

A predição das afinidades de ligação foram realizadas utilizando três funções empíricas de *scoring* desenvolvidas em trabalho prévio. As funções utilizadas neste trabalho são classificadas em (1) genérica, (2) classe-específica e (3) genérica derivada de resultados de *docking*. A função de *scoring* genérica foi desenvolvida a partir de treinamento utilizando o conjunto de dados PDBbind v2013, um conjunto de dados composto por estruturas de complexos proteína-ligante de alta qualidade com dados de afinidade de ligação experimental associados, disponível para desenvolvimento e validação de metodologias aplicadas ao *drug design* (Li, 2014). As funções classe-específicas foram obtidas a partir de sub-conjuntos do PDBbind. O desenvolvimento das funções específicas foi realizado a partir do treinamento considerando conjuntos de complexos proteicos que compartilham propriedades em comum. As funções derivadas de *docking* foram desenvolvidas a partir de treinamento utilizando resultados de *redocking* dos complexos do PDBbind, obtidos pelo programa DockThor, visando considerar as conformações proteína-ligante resultantes de *docking* ao invés das conformações experimentais (Guedes, 2016). Os valores de afinidade proteína-ligante foram obtidos utilizando as funções genéricas, genérica derivada de *docking* e classe-proteína específicas, quando da predição dos valores de afinidade do alvo CDK2 (utilizando função empírica quinase-específica) e do alvo TRY (utilizando função empírica protease-específica).

Todos os ligantes ativos e *decoys* atracados às respectivas proteínas-alvo correspondentes, foram ranqueados de acordo com as funções de *scoring* e o desempenho das funções foi mensurado através de curvas *Receiver Operating Characteristic* (ROC), que considera a fração de verdadeiros positivos no eixo Y (Sensibilidade, Eq. (1)) contra a fração de falsos positivos no eixo X (Especificidade, Eq. (2)) de acordo com os *scores* preditos. Realizamos a análise comparativa da *area under the curve* (AUC), métrica comumente aplicada na avaliação de modelos

de classificação binária, sendo neste caso a classificação entre ativos e *decoys* baseada em valores de *score*. No cenário ideal visualiza-se um valor AUC-ROC igual a 1,0 i.e. o máximo valor para AUC, onde obtemos uma perfeita separação de ligantes ativos e *decoys*, considerando que, o conjunto de *decoys* é composto por ligantes estruturalmente dissimilares dos ligantes ativos, ou seja, ligantes inativos sem dados de afinidade experimentais relacionados e desfavoráveis a ligação. Valor AUC-ROC equivalente a 0,5 conceitua uma discriminação aleatória dos ligantes e valores abaixo de 0,5 implica em uma metodologia que fornece um ranqueamento melhor para *decoys* do que para ativos, um cenário indesejado (Hawkins, 2008; Kirchmair, 2008).

$$Se = \frac{N \text{ ativos selecionados}}{N \text{ total de ativos}} = \frac{TP}{TP + FN} \quad (1)$$

$$Sp = \frac{N \text{ inativos descartados}}{N \text{ total de inativos}} = \frac{TN}{TN + FP} \quad (2)$$

Os programas DockThor, Dtstatistic e as Funções Empíricas de *Scoring* foram desenvolvidos pelo GMMSB/LNCC (de Magalhães et al., 2014; Almeida, 2011; Guedes, 2016).

## 4 RESULTADOS E DISCUSSÃO

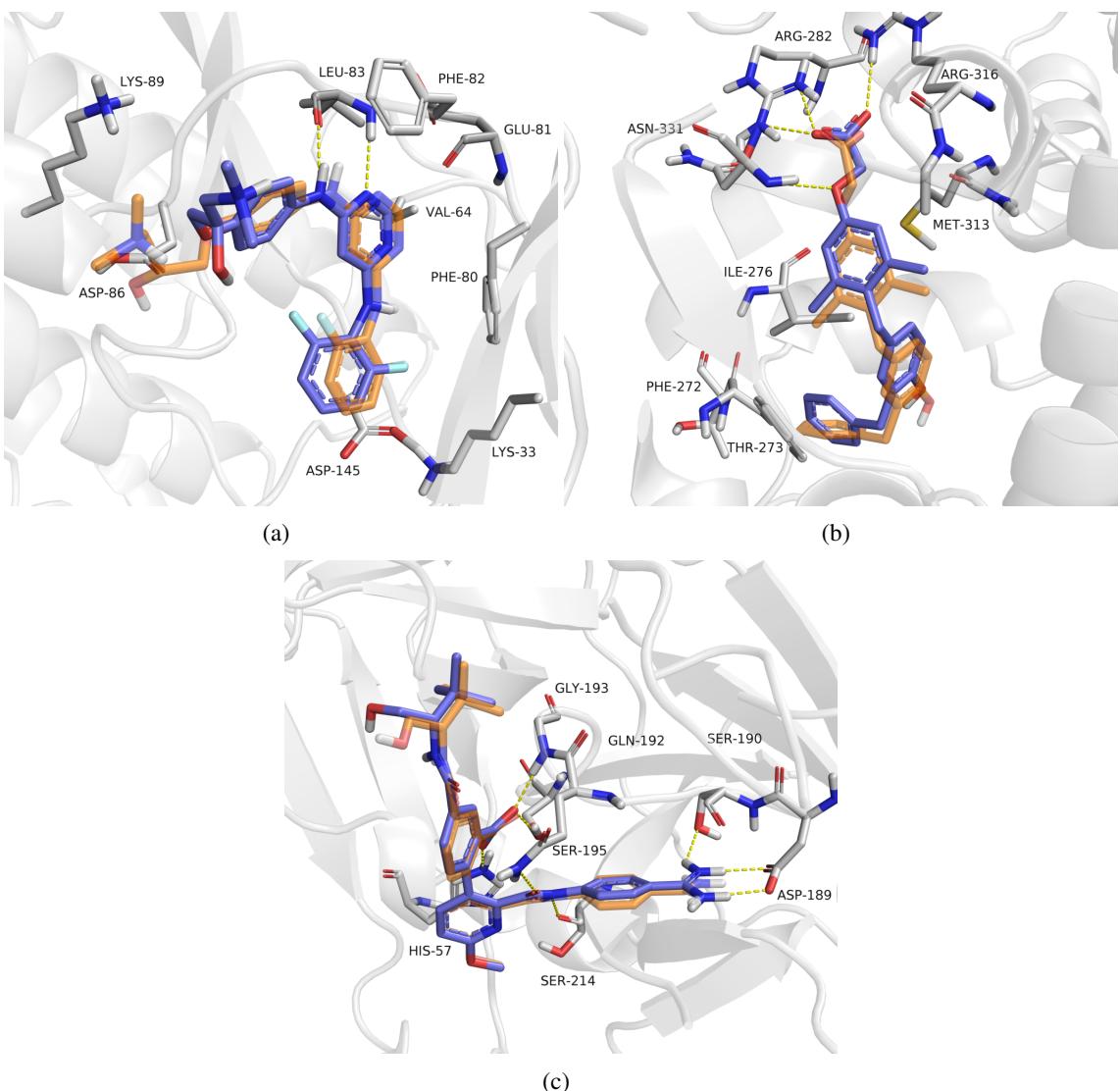
Os resultados dos estudos de *redocking* validaram a metodologia de *docking*, exibindo resultados satisfatórios que demonstram a correta reprodução dos modos de ligação encontrados pelo programa DockThor (Tabela 2). Foram encontrados, para os três sistemas estudados, valores de RMSD abaixo de 2,0 Å quando comparados os modos de ligação preditos e experimentais (Figura 1). Os experimentos de *redocking* são aplicados aos complexos experimentais das proteínas-alvo obtidas do PDB, de forma que validamos a preparação das três estruturas representativas de cada alvo para a triagem virtual em larga escala.

**Tabela 2: Valores de RMSD dos estudos de Redocking**

Complexo Estrutural	RMSD
<b>CDK2 (PDB 1H00)</b>	1,954
<b>THB (PDB 1Q4X)</b>	1,380
<b>TRY1 (PDB 2AYW)</b>	0,912

Os experimentos de *docking* em larga escala executados pelo DockThor resultaram na obtenção da pose (conformação) de menor valor de energia para cada ligante dos conjuntos, i.e. o modo de ligação predito do ligante no sítio ativo da respectiva proteína-alvo que tende a ser o encontrado na natureza. As melhores poses (complexos proteína-ligante) de cada ligante foram utilizadas para a predição dos valores de afinidade de ligação utilizando a função de *scoring* genérica para os três alvos, e utilizando as funções classe-específicas de quinases e proteases, para CDK2 e TRY, respectivamente.

As curvas ROC obtidas nos estudos de triagem virtual do alvo CDK2 podem ser visualizadas na Figura 2. Os valores para AUC obtidos utilizando os três tipos de funções avaliadas



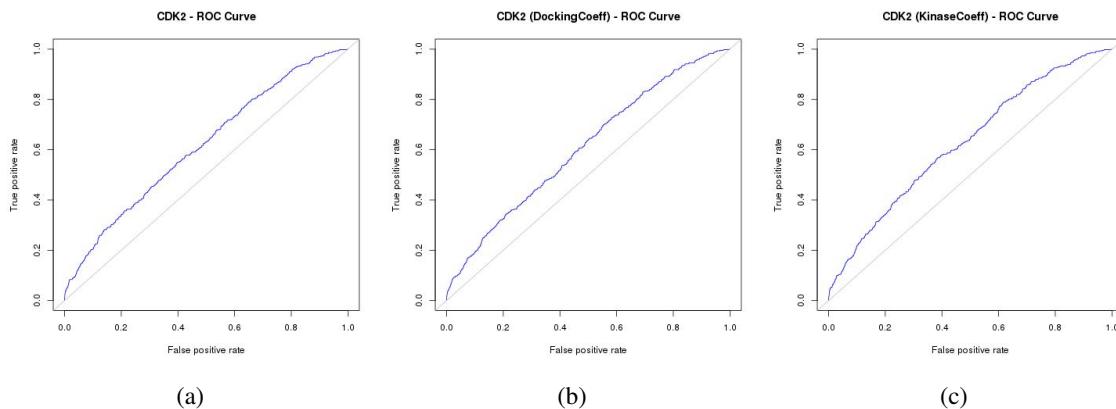
**Figura 1:** Modos de ligação obtidos no estudo de *redocking*. A conformação do *docking* encontra-se colorida em azul e a conformação do ligante do cristal, colorida em laranja. (a) CDK2, PDB 1H00, (b) THB, PDB 1O4X, (c) TRY, PDB 2AYW.

(Tabela 3) são bastante similares e demonstram a dificuldade das funções em discriminar ligantes ativos de decoys. Entretanto, é possível observar um desempenho um pouco melhor para a função proteína-específica. A dificuldade na discriminação dos ligantes deste alvo pode ser atribuída ao alto grau de flexibilidade observado em proteínas pertencentes a classe das quinases. A qualidade da predição da afinidade de ligação a partir das funções de scoring são dependentes da correta predição do modo de ligação e são bem conhecidas as limitações do tratamento da flexibilidade da proteína pelo método *single docking* receptor-rígido (i.e. realizando o atracamento dos ligantes utilizando apenas uma estrutura para representar a proteína-alvo).

Os resultados da triagem virtual dos ligantes do alvo THB (Figura 3) são similares qualitativamente aos obtidos para o alvo CDK2. O sítio ativo das proteínas pertencentes a classe de receptores nucleares, assim como as quinases, também demonstra considerável flexibilidade e

**Tabela 3:** Valores de *area under the curve* (AUC) das curvas ROC nas triagens virtuais obtidos pelas distintas funções

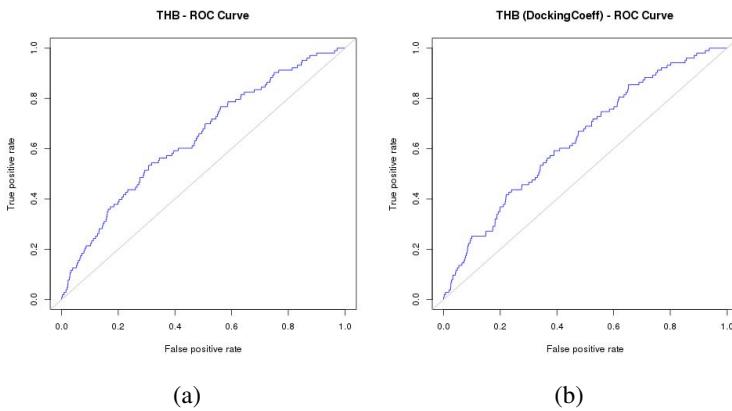
Proteína-alvo	Funções	AUC
<b>CDK2</b>	Genérica	0,614
	<i>Docking</i>	0,607
	Específica	0,627
<b>THB</b>	Genérica	0,642
	<i>Docking</i>	0,637
<b>TRY</b>	Genérica	0,701
	<i>Docking</i>	0,722
	Específica	0,821



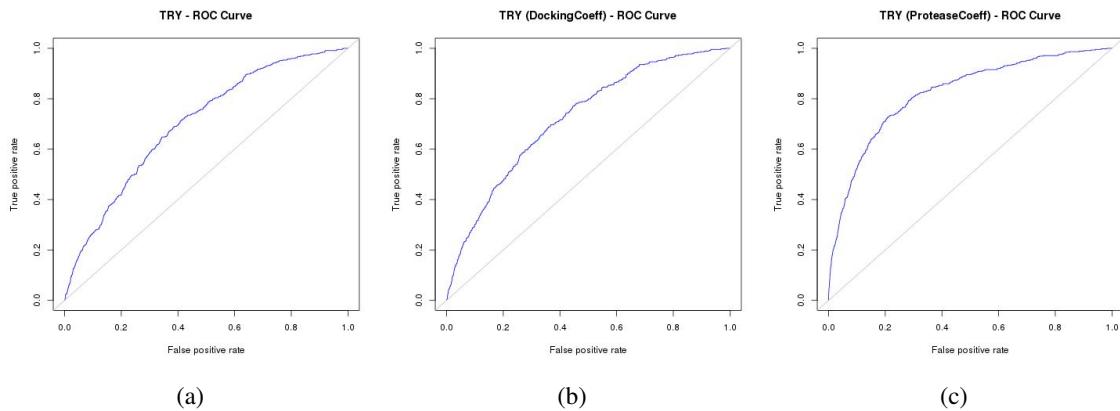
**Figura 2:** Curvas ROC da triagem virtual de ligantes da proteína-alvo Quinase Dependente de Ciclina 2 (CDK2), pertencente a classe das quinases. (a) função genérica, (b) função derivada de *docking* e (c) função quinase-específica.

um comportamento dinâmico, comprovados inclusive pela dificuldade na obtenção de estruturas por cristalografia (Huang et. al., 2010), e devido a isso acreditamos que para a melhoria dos resultados seja necessário tratar a flexibilidade. A classe de proteínas receptores nucleares ainda não dispõe de uma função específica e devido a isso, a análise da triagem virtual do alvo THB foi realizada utilizando apenas as funções genérica e a derivada de *docking*.

Os resultados da triagem da proteína-alvo TRY demonstram bom desempenho no ranqueamento dos ligantes, com valores das AUC-ROC superiores a 0,7 (Tabela 3). As curvas ROC demonstram boa predição das funções (Figura 4), atribuindo um *score* melhor aos ligantes ativos (i.e. verdadeiros positivos) em comparação aos *decoys*. Analisando a classificação realizada ao aplicar as três funções disponíveis, observa-se que a função de *docking* obtém melhores resultados que a genérica e, mais importante, observa-se uma considerável melhora ao utilizarmos a função específica para a classe de proteases.



**Figura 3:** Curvas ROC da triagem virtual de ligantes da proteína-alvo Receptor Beta-1 de Hormônio Tiroidiano (THB), pertencente a classe de receptores nuclear. (a) função genérica e (b) função derivada de docking,



**Figura 4:** Curvas ROC da triagem virtual de ligantes da proteína-alvo Tripsina (TRY), pertencente a classe das proteases. (a) função genérica, (b) função derivada de docking e (c) função protease-específica.

Resultados recuperados da literatura permitem a realização de um comparativo de desempenho junto as funções empíricas de *scoring* desenvolvidas pelo grupo. No trabalho de Zhang, (2014) foi realizada a triagem virtual de alvos selecionados do DUD-E com o objetivo de analisarem o desempenho do programa Vina (Trott, 2010) e da metodologia MM/GBSA. A triagem virtual do alvo CDK2 no estudo citado utilizando o Vina obteve AUC = 0.684 e utilizando o MM/GBSA, AUC = 0.744. Em análise da proteína-alvo TRY, são demonstrados AUC = 0.776 utilizando o programa Vina e AUC = 0.633 utilizando o MM/GBSA (Zhang, 2014). Observamos que nossos resultados encontram-se próximos dos visualizados e demonstram um desempenho melhor quando avaliamos a proteína-alvo Tripsina.

## 5 CONCLUSÕES

As análises demonstram o potencial de nossas funções empíricas de *scoring* aplicadas a estudos de triagem virtual, onde a classificação precisa de ativos e *decoys* consolida-se como característica mandatória. Os resultados observados apontam para o desenvolvimento de funções

classe-específicas, visto que a classificação ativo/inativo utilizando funções empíricas específicas resultam em um melhor ranqueamento dos ligantes, nos proporcionando um classificador mais confiável.

Acreditamos que a flexibilidade observada em determinadas classes de proteínas pode prejudicar a predição do modo de ligação e logo, interferir na predição da afinidade de ligação que implica diretamente na correta classificação dos ligantes. Uma maneira de superar esta limitação é a utilização da estratégia de *ensemble docking*, que em síntese, consiste em realizar o atracamento de um ligante, junto a mais de uma estrutura do receptor, objetivando representar a flexibilidade do receptor por um conjunto de conformações distintas.

Como continuação deste trabalho, almejamos estender a validação das funções utilizando um maior número de alvos do conjunto DUD-E, abordando outras classes de proteínas. É nossa intenção realizar esta validação utilizando a estratégia de *ensemble docking* e procurar implementar termos adicionais nas funções empíricas de afinidade objetivando uma melhoria no processo de classificação ativo/inativo.

### ***Agradecimentos***

Os autores agradecem a estrutura e suporte prestado pela equipe Santos Dumont – SINAPAD, ao fomento concedido pelos órgãos CAPES, CNPq, FAPERJ e a colaboração junto ao INCT-INO FAR.

### **REFERÊNCIAS**

- Almeida, D. M., 2011. *Dockthor: Implementação, Aprimoramento e Validação de um Programa de Docking Receptor-Ligante*. MSc thesis, Laboratório Nacional de Computação Científica / Petrópolis-RJ
- de Magalhães, C. S., Almeida, D. M., Barbosa, H. J. C., & Dardenne, L. E., 2014. A dynamic niching genetic algorithm strategy for docking highly flexible ligands. *Information Sciences*, vol. 289, p. 206-224.
- Guedes, I. A., de Magalhães, C. S., & Dardenne, L. E., 2014. Receptor–ligand molecular docking. *Biophysical Reviews*, vol. 6, n. 1, p. 75-87.
- Guedes, I. A., 2016. *Development of Empirical Scoring Functions for Predicting Protein-Ligand Binding Affinity*. PhD thesis, Laboratório Nacional de Computação Científica / Petrópolis-RJ.
- Halgren, T. A., 1999. MMFF VII. Characterization of MMFF94, MMFF94s, and other widely available force fields for conformational energies and for intermolecular-interaction energies and geometries. *Journal of Computational Chemistry*, vol. 20, n. 7, p. 730-748.
- Hawkins, P. C., et al. 2008. How to do an evaluation: pitfalls and traps. *Journal of computer-aided molecular design*, vol. 22, n. 3-4, p. 179-190.
- Huang, N., Shoichet, B. K., & Irwin, J. J., 2006. Benchmarking sets for molecular docking. *Journal of medicinal chemistry*, vol. 49, n. 23, p. 6789-6801.
- Huang, P., Chandra, V., & Rastinejad, F., 2010. Structural overview of the nuclear receptor

- superfamily: insights into physiology and therapeutics. *Annual review of physiology*, vol. 72, p. 247.
- Kirchmair, J., et al. 2008. Evaluation of the performance of 3D virtual screening protocols: RMSD comparisons, enrichment assessments, and decoy selection—What can we learn from earlier mistakes?. *Journal of computer-aided molecular design*, vol. 22, n. 3-4, p. 213-228.
- Li, Y., et al. 2014. Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the test set. *Journal of chemical information and modeling*, vol. 54, n. 6, p. 1700-1716.
- Liu, J., & Wang, R., 2015. Classification of current scoring functions. *Journal of chemical information and modeling*, vol. 55, n. 3, p. 475-482.
- Mysinger, M. M., Carchia, M., Irwin, J. J., & Shoichet, B. K., 2012. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, vol. 55, n. 14, p. 6582-6594.
- Trott, O. & Olson, A. J., 2010. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading, *Journal of Computational Chemistry*. vol. 31, p. 455-461
- Zhang, et al., 2014. Toward fully automated high performance computing drug discovery: a massively parallel virtual screening pipeline for docking and molecular mechanics/generalized Born surface area rescoring to improve enrichment. *Journal of chemical information and modeling*, vol. 54, n. 1, p. 324-337.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### PROGNÓSTICO DE DANO ESTRUTURAL UTILIZANDO ONDAS GUIADAS ULTRASSÔNICAS

**Jorge Franklin Mansur Rodrigues Filho**

Jorge.franklin.m@hotmail.com

**Gláucio Soares da Fonseca**

glauclio@metal.eeimvr.uff.br

Universidade Federal Fluminense

Av. dos Trabalhadores, 420, Vila Santa Cecília, 27255-125, Rio de Janeiro, Volta Redonda, Brasil

**Resumo.** As ondas guiadas da família cisalhante horizontal, em seu modo fundamental de propagação, são o objeto de estudo do presente trabalho, dada sua característica não dispersiva e melhor definição de falhas na direção de propagação. Modelos em elementos finitos foram desenvolvidos em Abaqus® para reproduzir o comportamento de propagação, reflexão e difração dessas ondas em uma placa de alumínio. A validade desses modelos foi atestada pela similaridade de seus resultados com dados obtidos experimentalmente em trabalhos recentes com mesmo foco. Foi possível visualizar a característica não dispersiva desta família de ondas tanto quanto verificar a capacidade de excitar apenas este modo e de fazê-lo se propagar de forma onidirecional. Além disso, das análises de interação com uma trinca foram extraídos os padrões de reflexão e difração para diferentes ângulos de incidência das ondas guiadas e a comparação entre as amplitudes de incidência e de reflexão. Também foi observada a conversão do modo cisalhante horizontal para o modo simétrico. Desta forma, todos os resultados dos modelos foram importantes para permitir um estudo mais profundo do fenômeno físico.

**Palavras-chave:** Ondas guiadas ultrassônicas, Elementos finitos, Ondas cisalhantes horizontais, SHM.

## 1 INTRODUÇÃO

Sistemas capazes de realizar um prognóstico de dano estrutural em produtos de engenharia em áreas como a mecânica, a civil e a aeroespacial têm ganhado a atenção dos pesquisadores nos últimos anos. Estes sistemas, baseados nas técnicas de ensaios não destrutivos, têm por objetivo informar, em tempo real, dados sobre a integridade de uma

estrutura, apontando a presença de danos e estimando assim a sua vida. Deste modo, as manutenções podem ser baseadas na condição estrutural, reduzindo custos, tempo e trabalho relativos aos clássicos planos de manutenção.

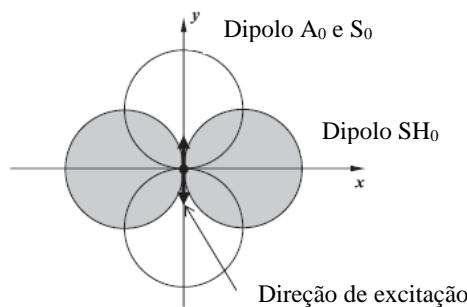
A principal técnica aplicada para realização de prognóstico é o teste por ondas guiadas (GW testing), que oferece um método efetivo de estimar a localização, severidade e tipo de dano, cobrindo uma grande área da estrutura. Outras técnicas de ensaios não destrutivos para avaliação de danos como por ultrassom clássico, raios-x e líquidos penetrantes não são viáveis para esta aplicação, pois não cobrem grandes áreas, apesar de sua boa precisão na identificação de defeitos.

As ondas guiadas podem ser definidas como ondas de tensão que são forçadas a percorrer um caminho definido pelos contornos de uma estrutura. Dentre os modos de propagação, o cisalhante horizontal é o mais indicado para testes não destrutivos em placas. Isto se deve ao fato de que este modo é não dispersivo em todos os valores de frequência e o movimento das partículas ocorre no plano da placa. Sendo assim, essas ondas podem viajar longas distâncias com atenuação desprezível e ainda manter o formato do sinal inalterado. Além disso, em baixas frequências, este é o único modo cisalhante horizontal que pode se propagar na placa, facilitando o processamento dos sinais (Rose, 2014).

O principal objetivo neste trabalho é desenvolver um estudo de caso sobre uma placa de alumínio na presença de um entalhe retangular. A partir da criação de um modelo em elementos finitos espera-se conseguir estudar o comportamento de reflexão e difração de ondas guiadas da família cisalhante horizontal devido à interação das ondas com o defeito inserido na placa. Portanto, buscam-se conclusões a respeito dos fenômenos que ocorrem no momento da interação, como a direção e amplitude do sinal de resposta e a influência do ângulo de incidência.

## 2 ONDA GUIADA CISALHANTE HORIZONTAL

O termo que nomeia este modo de propagação é utilizado em função do movimento vibracional das partículas que ocorre em um plano paralelo às superfícies da placa. Para excitar este movimento uma partícula da superfície deve sofrer um deslocamento axial com baixa frequência ultrassônica. A figura 1 mostra o movimento da partícula e quais dipolos de propagação irão surgir. Neste caso têm-se ambos os modos fundamentais de ondas cisalhantes horizontais ( $SH_0$ ) e de ondas simétricas ( $S_0$ ) e antissimétricas ( $A_0$ ). A excitação da mesma partícula em todas as direções equivale ao movimento torcional, o que causa a interferência destrutiva dos outros modos, mantendo apenas o modo cisalhante horizontal ( $SH_0$ ). Isto é positivo para a geração das ondas que terão apenas um modo se propagando em todas as direções, o que facilita o processamento dos sinais (Boivin. G, 2015).



**Figura 1: Dipolo de propagação de ondas, adaptado de Boivin. G, 2015**

Das equações de movimento é possível traçar um diagrama de velocidade de propagação dos vários modos em função da frequência. As curvas de dispersão para uma placa de Alumínio estão descritas na figura 1. É possível verificar que o modo fundamental SH<sub>0</sub> é não dispersivo ao longo do espectro de frequência, dada à invariância na velocidade de propagação. A faixa de frequência de interesse é abaixo de 1MHz, pois possui apenas os modos fundamentais de propagação.

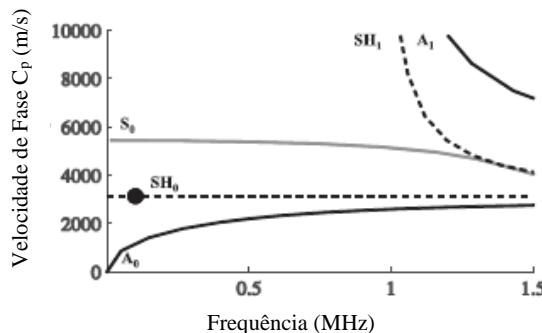


Figura 2: Curvas de dispersão para placa de alumínio, adaptado de Boivin. G, 2015

### 3 METODOLOGIA

O modelo foi desenvolvido utilizando o software Abaqus/CAE®, em seu módulo de análise explícita. As tensões fora do plano podem ser consideradas desprezíveis (Lowe, 2006), logo, para fins de redução do custo computacional, o modelo foi criado com apenas duas dimensões. Consiste de uma placa quadrada com dimensões de 500 mm de lado e espessura de 1.6 mm. A geometria representativa de uma trinca passante é introduzida na placa como um vazio retangular de 20 mm de comprimento e 0.1 mm de espessura.

As propriedades materiais do Alumínio utilizadas foram a densidade de 2700 kg/m<sup>3</sup>, módulo de Young de 70.8 GPa e Poisson de 0.34. O ponto de excitação recebeu um sinal de cinco ciclos em janela Hanning na frequência de 100 kHz para três diferentes posições, mostrados na figura 3, que distam 105 mm do vazio retangular. A posição 1 causa incidência perpendicular na trinca, enquanto a 2 leva as ondas a 45 graus e por ultimo a 3 gera incidência horizontal. Nessas condições, a excitação torcional gera uma onda cisalhante horizontal com comprimento de onda de aproximadamente 31.4 mm. Por isso, a malha foi construída com elementos do tipo CPS4R, quadrados com quatro nós, com dimensão máxima de 1.9625 mm, respeitando a relação de 16 elementos por comprimento de onda.

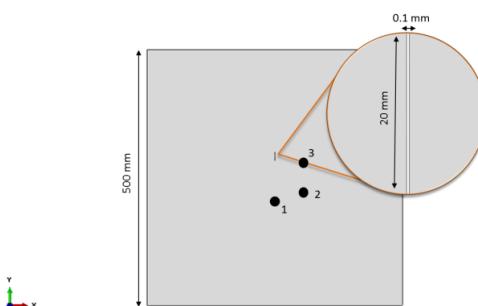
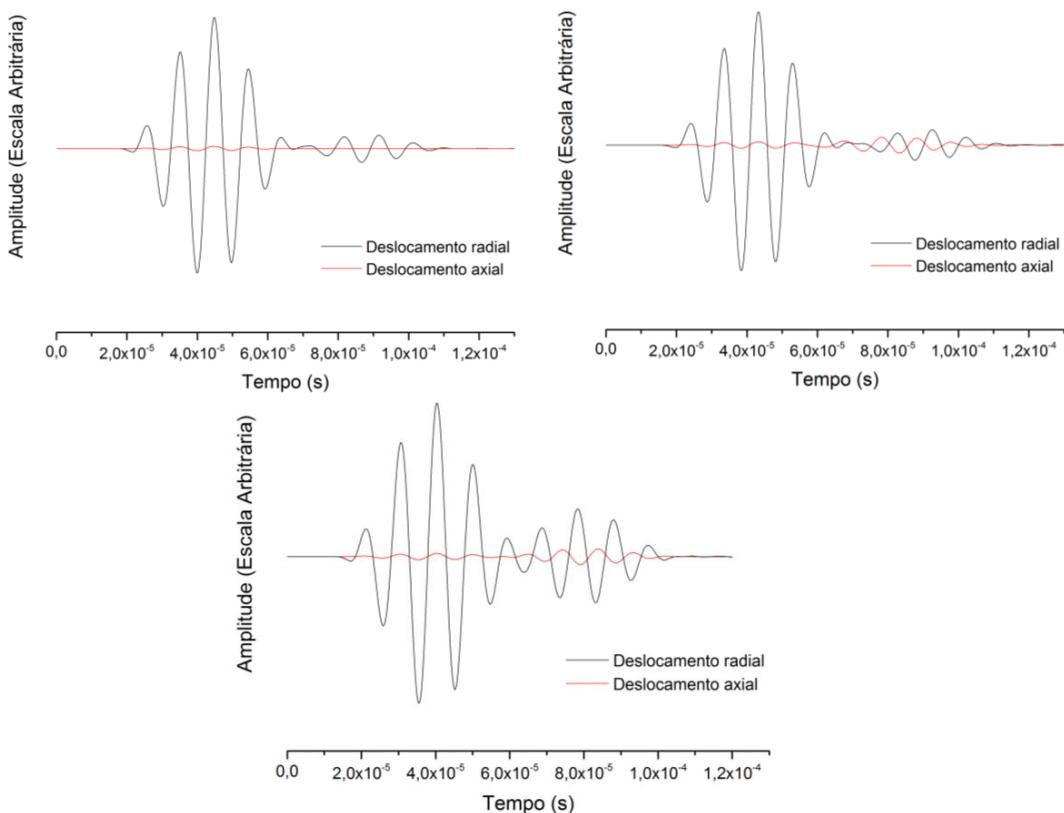


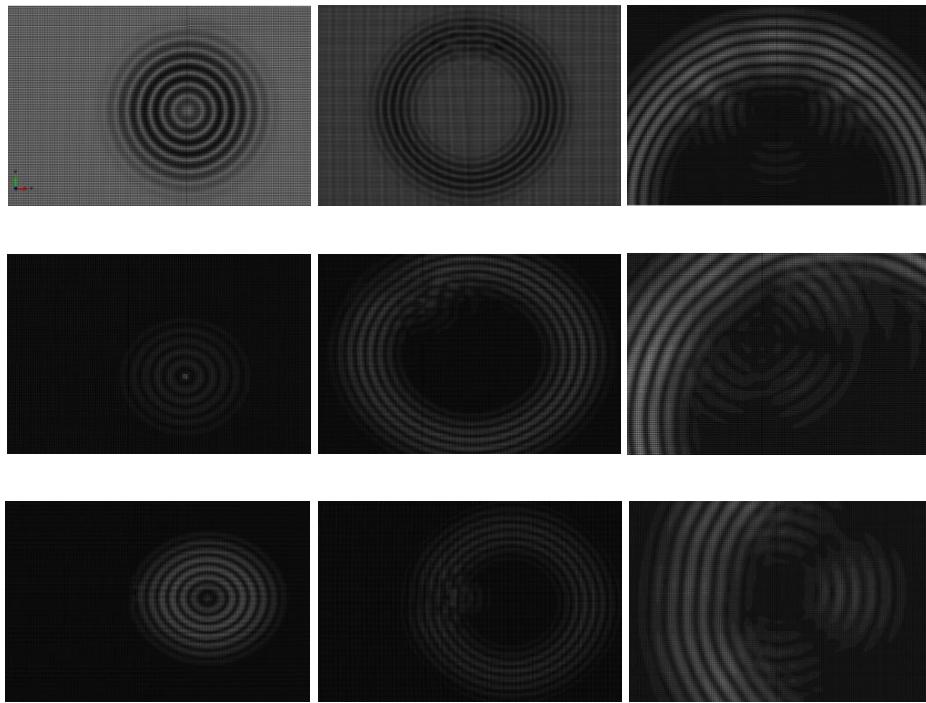
Figura 3: Características geométricas do modelo

## 4 RESULTADOS E DISCUSSÃO

Os três diferentes estudos foram executados com êxito e mostraram os resultados a seguir. A figura 4 contém os resultados de propagação e reverberação monitorados respectivamente nos pontos 1, 2 e 3. As grandes amplitudes observadas no início das curvas referem-se às ondas incidentes, que são geradas pelo sinal que aplica o movimento de torção da superfície. O movimento das partículas é referenciado pelo epicentro das ondas, por isso são descritos como axial e radial. No caso com incidência perpendicular é possível verificar grande movimento radial das partículas e desprezível movimento axial, mostrando forte propagação do modo SH0. No entanto, a onda refletida possui uma amplitude muito pequena quando comparada à direção de propagação horizontal. É o mesmo que pode ser observado no segundo caso, a 45 graus, que também apresenta uma reflexão mais fraca na direção de incidência. Isto é um provável desdobramento a interação da onda guiada com a geometria do defeito, que passa a agir como um gerador de ondas com características específicas definidas por seus contornos.



**Figura 4: Respostas de deslocamento nodal nas direções axial e radial para os três diferentes pontos de aplicação/monitoramento respectivamente.**



**Figura 5: Resultados das simulações mostrando a propagação das ondas através da magnitude do movimento das partículas. Estudo realizado a partir de três diferentes pontos de excitação.**

Estas características podem ser visualizadas através da figura 5 que ilustra a propagação das ondas para os três casos em sequência, na qual se pode ver a propagação através do epicentro, a interação com o defeito e por fim as ondas de resposta. Claramente nota-se uma reflexão mais intensa no último caso, com incidência horizontal. É possível verificar também que a trinca não reflete e difrata o modo cisalhante horizontal em todas as direções, diferente da fonte de propagação. Ocorre um fenômeno de transformação de modo, no qual o modo simétrico fundamental ( $S_0$ ) passa a se propagar a partir do defeito. Esta ocorrência está prevista, pois existem três modos de propagação ocupando esta faixa de baixa frequência.

Contudo, pode-se visualizar que para fins identificação da posição espacial do defeito a direção de excitação não tem influência sobre os resultados. Apesar de uma redução em sua quantidade de energia dependente da direção de propagação, o sinal de retorno possui o mesmo formato e seu tempo de viagem pode ser utilizado para estimar a posição do defeito.

## 5 CONCLUSÃO

Apesar da simplicidade do modelo apresentado, existe uma riqueza de conceitos aplicados à sua execução e de informações para um melhor estudo dos fenômenos de propagação de ondas guiadas.

A análise dos sinais incidentes e resultantes mostrou que, para a geometria de defeito apresentado, o ângulo de incidência terá papel importante tanto na amplitude, quanto na direção de propagação das ondas refletidas e difratadas. Outro ponto importante é que foi possível a verificação do surgimento de diferentes modos de propagação devido à interação com a trinca.

Os resultados demonstraram similaridade aos encontrados em trabalhos recentes, corroborando as considerações utilizadas para realização das análises. Espera-se que em trabalhos futuros estudos mais profundos possam ser extraídos destes modelos, além da

realização do aperfeiçoamento de suas características, como a adição de defeitos com geometrias complexas. Desta forma, têm-se a visão de que este é um trabalho de inicio ao estudo desta área da engenharia.

## **REFERÊNCIAS**

- Rose, J. L. Ultrasonic Guided Waves in Solid Media. Cambridge University Press. Vol.1, 2014.
- Ratassepp, M.; Lowe, M. J. S. E Cawley, P. “Scattering of the fundamental shear horizontal mode in a plate when incident at a through crack aligned in the propagation direction of the mode.” J. Acoust. Soc. Am. \_2006\_.
- Boivin, G; Belangér, P. “Development of a low frequency omnidirectional piezoelectric shear horizontal wave transducer” IOP- Smart Mater \_2016
- R. Rajagopal; M. J. S. Lowe, “Short range scattering of the fundamental shear horizontal guided wave mode normally incident at a through-thickness crack in an isotropic plate,” J. Acoust. Soc. Am. \_2007\_.
- Ratassepp, M.; Lowe, M. J. S. E Cawley, P. “Angular influence on the scattering of fundamental shear horizontal guided waves by a through-thickness crack in an isotropic plate” J. Acoust. Soc. Am. \_2007\_.
- Rajagopal, R.; Lowe, M. J. S. “Scattering of the fundamental shear horizontal guided wave by a part-thickness crack in an isotropic plate” J. Acoust. Soc. Am. \_2008\_.
- Ma, J. E Cawley, P. “Low-frequency pulse echo reflection of the fundamental shear horizontal mode from part-thickness elliptical defects in plates” J. Acoust. Soc. Am. \_2010\_.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### UM ESTUDO COMPARATIVO DE SIMULAÇÃO DE HIPERTERMIA COM NANOPARTÍCULAS MAGNÉTICAS

**Gustavo Resende Fatigate**

gustavoresende2010@gmail.com

Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais

Avenida Dirce Pereira Rosa, 300 - Jd. Esperança , Poços de Caldas - MG, 37713-100 - Brasil

**Ruy Freitas Reis**

ruyfreis@gmail.com

Universidade Federal de Juiz de Fora

Rua José Lourenço Kelmer, S/n - Martelos, Juiz de Fora - MG, 36036-330 - Brasil

**Rafael Felipe Coelho Neves**

rafael.neves@ifsuldeminas.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais

Avenida Dirce Pereira Rosa, 300 - Jd. Esperança , Poços de Caldas - MG, 37713-100 - Brasil

**Resumo.** O câncer é a segunda maior causa de morte no mundo, segundo a OMS. Os métodos de tratamento mais conhecidos são a quimioterapia e a radioterapia. Porém, um tratamento alternativo que vem sendo pesquisado e pode ser utilizado em conjunto a essas técnicas está relacionado com a hipertermia por nanopartículas magnéticas que podem ser injetadas no tumor alvo, e ao serem submetidas a um campo magnético alternado geram calor. Esse calor pode ser utilizado para elevar a temperatura do tecido cancerígeno até um limiar que cause necrose celular. Assim, é possível mensurar as porções de tecido saudável e tumoral que foram destruídas após a aplicação da hipertermia. Esta pesquisa busca simular computacionalmente a hipertermia, implementando dois modelos matemáticos distintos (Pennes e Meios Porosos), a fim de comparar seus resultados. Foi simulado um tecido vivo composto por dois tipos de células: saudáveis e tumorais. Os parâmetros utilizados para descrever ambos os tecidos foram retirados da literatura, para célula saudável utilizou-se os valores referentes a um tecido muscular. Para obter a solução numérica dos modelos implementados foi utilizado o método das diferenças finitas. Os modelos foram discretizados espacialmente utilizando uma diferença centrada e discretizados temporalmente com uma diferença progressiva, o que garante uma convergência  $O(\Delta_x^2)$  no espaço e  $O(\Delta_t)$  na marcha do tempo. A simulação utilizando dois modelos distintos possibilitou realizar a modelagem da hipertermia utilizando diferentes parâmetros.

**Palavras-chave:** Hipertermia, Biotransferência de Calor, Meios Porosos, Pennes, Método das Diferenças Finitas

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 Introdução

Com base em dados da OMS (2016) o câncer é responsável pelo segundo maior índice de mortalidade no mundo sendo responsável por cerca de 8,2 milhões de mortes em 2012. Há inúmeros tipos de câncer que podem atingir praticamente todos os órgãos do corpo. Dentre estes os mais comumente diagnosticados são os de pulmão(12,7% do total), mama (10,9%) e colorretal (9,7%).

Sendo o câncer um problema de saúde mundial, técnicas de combate e controle são alvo de interesse da pesquisa científica. Várias técnicas já foram criadas para auxiliar e tratar os pacientes com câncer. A título de exemplo pode-se citar a quimioterapia e radioterapia, estas sendo as mais usadas em seu tratamento (Salloum *et al.*, 2009). Um novo tratamento vem sendo estudado com o intuito de auxiliar os tratamentos de câncer já conhecidos, a hipertermia. O tratamento consiste no superaquecimento do tecido até a necrose celular. Este possui a vantagem de poder ser utilizado tumores sólidos irregulares e/ou profundos, no qual se injeta as nanopartículas que ao serem expostas a um campo magnético alternado de baixa frequência geram calor.

Um modelo para descrever a biotransferência de calor foi proposto por Pennes (1948). Esta abordagem é mais comumente utilizada devido à proximidade entre os resultados teóricos e experimentais no fenômeno de transferência de calor em tecidos vivos. Por outro lado, a biotransferência de calor pode ser descrita através de uma aproximação em meios porosos (Khaled e Vafai, 2003; Reis *et al.*, 2015). Este modelo leva em consideração alguns fatores a mais que o proposto por Pennes (1948), tais como a velocidade do sangue. Apesar das diferenças na formulação, ambos os modelos possuem resultados finais próximos.

Nesta pesquisa almeja-se simular a utilização de nanopartículas magnéticas no tratamento de tumores através de equações diferenciais parciais (EDPs) que descrevem o fenômeno. Foram implementados dois modelos matemáticos, o de Pennes (1948) e o de meio poroso (Khaled e Vafai, 2003) afim de compará-los. A resolução das EDPs são realizadas através do método das diferenças finitas em meio heterogêneo. Além disso, esta pesquisa deseja realizar os cálculos da biotransferência de calor, considerando as diferenças entre tecidos cancerosos e saudáveis tais como perfusão sanguínea, calor metabólico, entre outros parâmetros.

Esse trabalho será dividido em cinco seções começando pela Introdução, seguida pelo Modelo Matemático, o Método Numérico, Resultados e, por último, a seção de Conclusões e Trabalhos Futuros.

## 2 Modelo Matemático

Os modelos matemáticos utilizados, como dito anteriormente, foram o modelo de Pennes (1948) e o de meio poroso (Khaled e Vafai, 2003). Ambos utilizados para o cálculo de biotransferência de calor em tecidos vivos, podendo ser comparados a fim de estudo. Serão apresentadas as respectivas equações de cada modelo citado. Ambos os modelos apresentam o termo de calor gerado ( $Q_m$ ), o qual modela a geração de calor promovida pelas nanopartículas magnéticas no tecido, quando submetidas a um campo magnético alternado.

### 2.1 Equação de Pennes

O modelo de hipertermia baseado na abordagem feita por Pennes (1948) está de acordo com o utilizado em Reis *et al.* (2016) e é dado pela Eq. (1)

$$\begin{cases} \rho c \frac{\partial T}{\partial t} = \nabla \cdot k \nabla T + \omega_b \rho_b c_b (T_a + T) + Q_m + Q_r & \text{em } \Omega \times I \\ k \nabla T \cdot \vec{n} = f_{tecido} & \text{em } \partial\Omega \times I \\ T(\cdot, 0) = T_0 & \text{em } \Omega, \end{cases} \quad (1)$$

onde  $\Omega \in \mathbb{R}$  é o domínio espacial da equação,  $I \in \mathbb{R}^+$  é o domínio do tempo,  $T : \Omega \times I \rightarrow \mathbb{R}^+$  é o campo de temperaturas do tecido;  $\rho$ ,  $c$  e  $k$  são, respectivamente, densidade, calor específico e condutividade térmica do tecido;  $\rho_b$ ,  $c_b$  e  $\omega_b$  são, respectivamente, densidade do sangue, calor específico do sangue e perfusão sanguínea;  $T_a$  e  $T_0$  são temperatura do sangue arterial e inicial, respectivamente;  $Q_m$  e  $Q_r$  são, respectivamente, quantidade de calor metabólico gerado e quantidade de calor externo adicionado. Finalmente,  $f_{tecido}$  é o valor da condição de contorno do tipo Neumann aplicado ao modelo.

Vale a pena destacar, que a modelagem proposta por Pennes (1948) assume as seguintes simplificações: (1) Equilíbrio local, ou seja a principal troca de calor entre o sangue e o tecido acontece entre os vasos capilares e as arteríolas; (2) O fluxo sanguíneo nos pequenos capilares é assumido como sendo isotrópico; (3) Não é considerado a geometria vascular; (4) Assume-se que o sangue que alcança as arteríolas está na temperatura interna do corpo  $T_a$  (Jiji, 2009).

## 2.2 Modelo do Meio Poroso

Para utilizar a abordagem baseada em meio poroso aplicada à tecidos vivos, utilizou-se o modelo conforme a Eq. (2) (Khaled e Vafai, 2003; Reis *et al.*, 2015).

$$\begin{cases} [\rho c(1 - \varepsilon) + \rho_b c_b \varepsilon] \frac{\partial T}{\partial t} = \nabla \cdot k \nabla T + \beta \frac{\rho_b (u_b)_{avr} \varepsilon c_{p_b}}{\delta} (T_a - T) + (1 - \varepsilon)(Q_m + Q_r) & \text{em } \Omega \times I \\ k \nabla T \cdot \vec{n} = f_{tecido} & \text{em } \partial\Omega \times I \\ T(\cdot, 0) = T_0 & \text{em } \Omega, \end{cases} \quad (2)$$

onde  $\Omega \in \mathbb{R}$  é o domínio espacial da equação,  $I \in \mathbb{R}^+$  é o domínio do tempo,  $T : \Omega \times I \rightarrow \mathbb{R}^+$  é o campo de temperaturas do tecido;  $k = (1 - \varepsilon)k_t + \varepsilon k_b$ ;  $\rho$ ,  $c$  e  $k_t$  são, respectivamente, densidade, coeficiente térmico e condutividade térmica do tecido;  $c_b$ ,  $\rho_b$ ,  $k_b$  e  $(u_b)_{avr}$  são, respectivamente, condutividade térmica do sangue, densidade do sangue, condutividade térmica do sangue e velocidade do sangue;  $\varepsilon$ ,  $\delta$  e  $\beta$  são, respectivamente, porosidade do meio, distância média entre vasos e fator de correção;  $Q_m$ ,  $Q_r$  são, respectivamente, quantidade de calor metabólico gerado e quantidade de calor externo adicionado.  $T_a$  e  $T_0$  são as temperatura do sangue e inicial, respectivamente. Finalmente,  $f_{tecido}$  é o valor da condição de contorno do tipo Neumann aplicado ao modelo.

## 3 Método Numérico

O método numérico utilizado para resolver ambas as equações é o método das diferenças finitas (MDF) com ordem de convergência  $O(\Delta_x^2)$ , além disso a discretização utilizada no domínio do tempo foi Euler explícito com ordem de convergência  $O(\Delta_t)$ . Utilizou-se o critério de convergência demonstrado em Reis *et al.* (2016) e a ordem de convergência não foi verificada, uma vez que o método foi o mesmo utilizado em Reis *et al.* (2014) além de ser um método bem conhecido pela literatura científica.

Os modelos matemáticos foram apresentados em  $\mathbb{R}^3$ , porém seus resultados serão apresentados em um domínio unidimensional, uma vez que facilita as suas respectivas compreensões e visualizações. Para

ambos os modelos foram aplicados aos seus respectivos domínios  $\Omega \times I$  uma discretização uniforme de coordenadas  $x_i = i\Delta x$ , onde  $\Delta x$  é o espaçamento da grade na direção  $x \in \mathbb{R}$ . Análogo à discretização do espaço, se propõe a temporal da seguinte maneira:  $t_n = n\Delta t$ , onde  $\Delta t$  é o espaçamento da discretização no domínio do tempo.

### 3.1 Modelo de Pennes Discreto

O modelo dado pela Eq. (1) discretizado através de diferenças finitas é dado por:

$$T_i^{n+1} = \frac{\Delta t}{\rho c} \left[ \frac{k_{i+1/2}(T_{i+1}^n - T_i^n) - k_{i-1/2}(T_i^n - T_{i-1}^n)}{\Delta x^2} + \rho_b c_b \omega_b (T_a - T_i^n) + Q_m + Q_r \right] + T_i^n \quad (3)$$

### 3.2 Modelo Baseado em Meio Poroso Discreto

O modelo dado pela Eq. (2) discretizado através de diferenças finitas é dado por:

$$\begin{aligned} T_i^{n+1} = & \frac{\Delta t}{\rho c_p(1-\varepsilon) + \rho_b c_p \varepsilon} \left[ \frac{k_{i+1/2}(T_{i+1}^n - T_i^n) - k_{i-1/2}(T_i^n - T_{i-1}^n)}{\Delta x^2} \right. \\ & \left. + \beta \frac{\rho_b v \varepsilon c_p}{\delta} (T_a - T_i^n) + (1-\varepsilon)(Q_m + Q_r) \right] + T_i^n \end{aligned} \quad (4)$$

## 4 Resultados Numéricos

Todas as simulações são executadas e implementadas utilizando a linguagem de programação C/C++ e o compilador GNU-GCC, além da ferramenta de visualização científica GNUPlot.

O domínio simulado foi de  $0,1m$  e divido em 2 tipos de tecidos: para  $x \in [0,045; 0,056]$  foi definido como tumor e caso contrário como tecido saudável. Para cada tipo de tecido são utilizados diferentes valores de parâmetros, que foram retirados da literatura (Salloum *et al.*, 2009, 2008; Reis *et al.*, 2015).

A injeção no ponto é modelada por SAR (taxa de absorção específica), de acordo com Salloum *et al.* (2008). A função  $Q_r$  se calcula a partir de  $\sum_{i=1}^n Ae^{-r(\vec{x})_i^2/r_{0,i}^2}$ , onde  $n$  é o número de injeções;  $r = ((x - x_0)^2)^{0.5}$ ;  $A$  é energia total gerada pela injeção;  $x$  e  $x_0$  são, respectivamente, o local do espaço e a posição da injeção;  $r_0$  é o raio de ação. As condições iniciais e de contorno utilizadas nos modelos estão especificadas em Tab. 1. Para todas as simulações foi utilizada uma única injeção direta no centro do tecido tumoral.

**Tabela 1:** Condições iniciais e de contorno utilizadas nos modelos.

Modelo	Condição Inicial	Condição de Contorno
Modelo de Pennes	$T_0 = 37,0^\circ C \forall x \in \Omega$	$k\nabla T \cdot \vec{n} = 0 \forall x \in \partial\Omega \times I$
Modelo de meio poroso	$T_0 = 37,0^\circ C \forall x \in \Omega$	$k\nabla T \cdot \vec{n} = 0 \forall x \in \partial\Omega \times I$

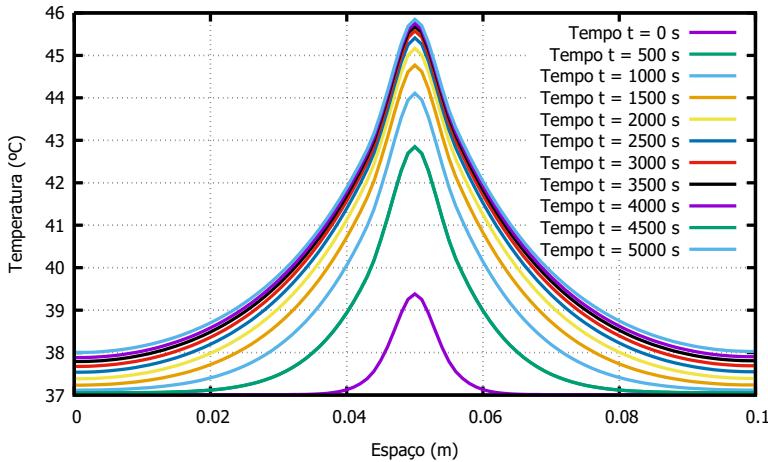
Para o SAR tem-se os valores de  $A = 0,2 \times 10^6 W$ ,  $r_0 = 2,5 \times 10^{-3} m$  e  $x_0 = 0,05 m$ . Todos as simulações foram feitas com o tempo de simulação fixado em  $t = 5000 s$ , uma vez que se almeja uma exposição por cerca de 50 min de hipertermia para garantir a necrose do tecido alvo. Por fim todas as simulações utilizaram uma discretização no espaço com  $\Delta x = 0,001 m$  e no tempo com  $\Delta t = 0,0001 s$ .

## 4.1 Abordagem de Pennes

Todos os parâmetros do modelo estão na Tab. 2 e foram retirados da literatura (Salloum *et al.*, 2009; Reis *et al.*, 2016). A Fig. 1 apresenta o resultado da Eq. (1) discretizada pela Eq. (3). Como pode ser observado a temperatura se elevou acima de  $43^{\circ}C$  na região onde o tumor está situado.

**Tabela 2: Parâmetros do Modelo de Pennes**

Parâmetros	Unidade	Tecido Saudável	Tecido Tumoral
$k$	$W/m^{\circ}C$	0,51	0,64
$\omega_b$	$s^{-1}$	$5,0 \times 10^{-4}$	$1,25 \times 10^{-3}$
$\rho$	$Kg/m^3$	1000,0	1000,0
$\rho_b$	$Kg/m^3$	1000,0	1000,0
$Q_m$	$W/m^3$	420,0	4200,0
$c$	$J/Kg^{\circ}C$	4200,0	4200,0
$c_b$	$J/Kg^{\circ}C$	4200,0	4200,0



**Figura 1: Resultado do experimento *in silico* utilizando a Eq. (1)**

## 4.2 Abordagem de Meio Poroso

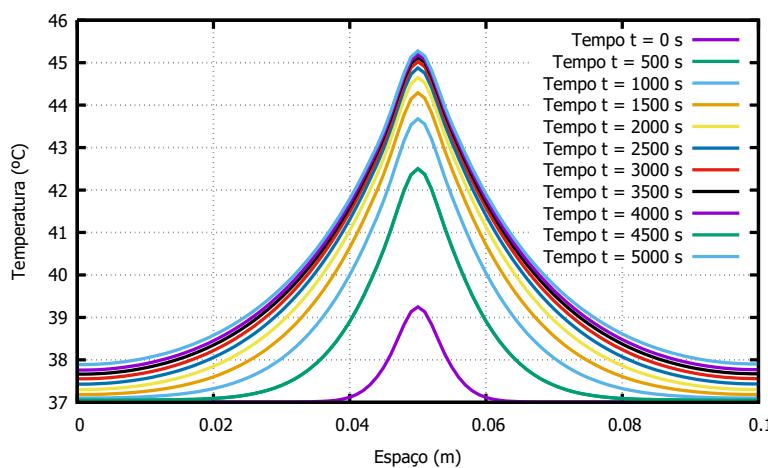
Todos os parâmetros do modelo estão na Tab. 3 e foram retirados da literatura (Khaled e Vafai, 2003; Reis *et al.*, 2015). A Fig. 2 apresenta o resultado da Eq. (2) discretizada pela Eq. (4). Também pode ser observado que a temperatura se elevou acima de  $43^{\circ}C$  na região onde o tumor está situado.

## 4.3 Comparação Entre os Modelos

A Fig. 3 mostra os resultados de ambos os modelos de modo que é possível observar que os dois modelos são capazes de obter resultados qualitativamente bem próximos, principalmente em formato das curvas e valores. Vale destacar que tomou-se um ponto dentro e um fora do tumor em cada uma das simulações.

**Tabela 3: Parâmetros do modelo de meio poroso**

Parâmetro	Unidade	Tecido Saudável	Tecido Tumoral
$k_t$	$W/m^oC$	0,51	0,56
$k_b$	$W/m^oC$	0,64	0,64
$\rho$	$Kg/m^3$	1000,0	1000,0
$\rho_b$	$Kg/m^3$	1000,0	1000,0
$Q_m$	$W/m^3$	420,0	4200,0
$u_b$	$mm/s$	1,5	3,5
$\delta$	$mm$	6,0	4,0
$\varepsilon$	—	0,02	0,02
$\beta$	—	0,085	0,085



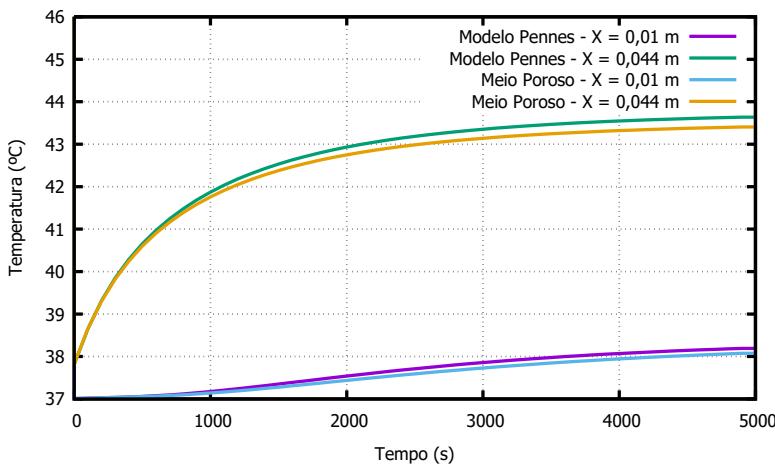
**Figura 2: Resultado do experimento *in silico* utilizando a Eq. (2)**

## 5 Conclusões e Trabalhos Futuros

Uma vez que assumindo as mesmas simplificações que Pennes (1948) fez para obter seu modelo, conforme detalhado em Khaled e Vafai (2003), o modelo baseado em meio poroso utilizando 2 fases (sangue e tecido) recai em um modelo matematicamente parecido com o de Pennes (1948). Maiores detalhes da simplificação do modelo bifásico para chegar ao modelo de Pennes podem ser encontrados em Khaled e Vafai (2003). Utilizando o valor de correção  $\beta$  o modelo baseado em meio poroso produz resultados qualitativamente parecidos ao modelo de Pennes, como se pode observar na figura 3, obtida ao observar os pontos 0,01m e 0,04m em função do tempo para ambos os modelos.

Conforme pode ser observado nas Figs. 1 e 2, a simulação de hipertermia por nanopartículas magnéticas tanto pelo modelo de Pennes quanto o de meio poroso monofásico, produzem respostas muito próximas uma da outra, conforme o esperado.

Além disso, nos pontos que representam o tumor elevou-se a temperatura acima de  $43^oC$ , o que leva à necrose do tecido cancerígeno Salloum *et al.* (2009). Vale destacar, que neste caso, se a injeção estiver



**Figura 3: Resultado comparativo dos modelos dados pelas Eqs. (1) e (2)**

localizada no centro do tumor obteve-se a necrose de todo o tecido tumoral, e apenas 1,24% de tecido saudável.

Em trabalhos futuros se planeja fazer uma implementação tridimensional dos modelos, possibilitando executar a simulação em geometrias mais complexas. Além disso, também deseja-se implementar a versão bifásica do modelo de meio poroso.

## Referências Bibliográficas

- Jiji L.M. 2009, *Heat Conduction*. Springer-Verlag Berlin Heidelberg. ISBN 978-3-642-01266-2.
- Khaled A.R. e Vafai K., 2003. The role of porous media in modeling flow and heat transfer in biological tissues. *International Journal of Heat and Mass Transfer*, v. 46, n. 26, pp. 4989 – 5003.
- OMS. Organização Mundial da Saúde. Publicação Eletrônica - <http://www.who.int/>. Último acesso em 11 de Outubro de 2016, 2016.
- Pennes H.H., 1948. Analysis of tissue and arterial blood temperature in the resting human forearm. *Journal of Applied Physiology*, v. 1, pp. 93–122.
- Reis R.F., dos Santos Loureiro F., e Lobosco M., 2014. A parallel 2d numerical simulation of tumor cells necrosis by local hyperthermia. *Journal of Physics: Conference Series*, v. 490, n. 012138.
- Reis R.F., dos Santos Loureiro F., e Lobosco M., 2015. Parameters analysis of a porous medium model for treatment with hyperthermia using openmp. *Journal of Physics: Conference Series*, v. 633, n. 1, pp. 012087.
- Reis R.F., dos Santos Loureiro F., e Lobosco M., 2016. 3d numerical simulations on gpus of hyperthermia with nanoparticles by a nonlinear bioheat model. *Journal of Computational and Applied Mathematics*, v. 295, pp. 35–47.
- Salloum M., Ma R., e Zhu L., 2008. An in-vivo experimental study of temperature elevations in animal tissue during magnetic nanoparticle hyperthermia. *International Journal of Hyperthermia*, v. 24, n. 7, pp. 589–601.

Salloum M., Ma R., e Zhu L., 2009. Enhancement in treatment planning for magnetic nanoparticle hyperthermia: Optimization of the heat absorption pattern. *International Journal of Hyperthermia*, v. 25, n. 4, pp. 309–321.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### DETERMINAÇÃO DE PARÂMETROS COM DISTRIBUIÇÃO ESPACIAL NÃO UNIFORME NUM MODELO DE NEUROCIÊNCIA VIA O MÉTODO DE LANDWEBER NÃO LINEAR

**Jemy A. Mandujano V.**

jhimyunac@gmail.com

**Alexandre L. Madureira**

alm@lncc.br

Laboratório Nacional de Computação Científica

Av. Getúlio Vargas 333 - Quitandinha, 25651-075, Rio de Janeiro, Petrópolis, Brasil

**Resumo.** O objetivo deste trabalho é obter de forma indireta três parâmetros (condutância de sódio, condutância de potássio e condutância de vazamento) de uma equação diferencial parcial, utilizando um método de Regularização Iterativa. O problema é motivado pelo comportamento de canais iônicos da célula neuronal, que é de difícil determinação experimental. Utilizamos um modelo simplificado, no caso a equação do cabo passivo, que é uma equação diferencial parabólica linear, com termos de difusão e reação. Consideramos que os parâmetros desconhecidos tem distribuição espacial não uniforme. Para determinar os parâmetros utiliza-se o método de Landweber não linear, que, a partir de um ponto inicial qualquer (um espaço de Hilbert), busca de forma iterativa aproximações para as funções desconhecidas. Neste trabalho descrevemos a motivação biológica do problema, bem como a base matemática do algoritmo, e por fim mostramos um resultado computacional.

**Palavras-chave:** Neurociência Computacional, Equação do Cabo, Problema inverso, Método de Landweber, Método de Diferenças Finitas.

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUÇÃO

Hodgkin e Huxley (1952) desenvolveram seu famoso modelo , que descreve matematicamente como potenciais de ação nos neurônios são iniciados e propagados. A corrente iônica ( $I_{\text{iônica}}$ ) é dada pela soma das correntes iônicas de íons específicos. Hodgkin e Huxley postularam que a corrente iônica de um íon é independente da corrente iônica dos outros íons e identificaram três correntes iônicas que são as principais responsáveis pela geração do potencial de ação: a corrente de sódio ( $I_{Na}$ ), a corrente de potássio ( $I_k$ ) e a corrente de outros íons ( $I_{\text{vazamento}}$ ).

$$I_{\text{iônica}} = I_{Na} + I_k + I_{\text{vazamento}} \quad (1)$$

O modelo de Hodgkin e Huxley é dado por uma equação diferencial parcial parabólica não linear acoplada. Devido à complexidade do modelo não linear, simplifica-se a uma equação de cabo passivo, que é uma equação diferencial parcial parabólica linear. Mandujano e Madureira (2016) realizaram uma modelagem matemática da equação do cabo passivo, considerando uma só corrente  $I_{\text{iônica}} = I_k = G_k(x)[V(t, x) - E_k]$ ,

$$\frac{1}{R_i + R_e} V_{xx}(t, x) = C_M V_t(t, x) + I_{\text{iônica}}. \quad (2)$$

Neste trabalho é considerado as três correntes (1). A modelagem das demais correntes (Sódio e Vazamento) é similar à corrente de potássio,  $I_{Na} = G_{Na}(x)[V(t, x) - E_{Na}]$  e  $I_{\text{vazamento}} = G_l(x)[V(t, x) - E_l]$ , então da equação (2) temos

$$\begin{aligned} \frac{1}{R_i + R_e} V_{xx}(t, x) &= C_M V_t(t, x) + G_k(x)[V(t, x) + E_k] I_{Na} \\ &\quad + G_{Na}(x)[V(t, x) - E_{Na}] + G_l(x)[V(t, x) - E_l]. \end{aligned} \quad (3)$$

Onde  $R_i$  é a resistência interna do neurônio,  $R_e$  a resistência externa do neurônio,  $E_k$ ,  $E_{Na}$  e  $E_l$  são potenciais de Nernst de potássio, de sódio e de vazamento, respectivamente,  $V$  é o potencial da membrana,  $G_k$ ,  $G_{Na}$  e  $G_l$  são as condutâncias de potássio, de sódio e de vazamento, respectivamente,  $C_M$  é a capacitância. Os dados  $R_i$ ,  $R_e$ ,  $C_M$ ,  $E_k$ ,  $E_{Na}$ ,  $E_l$  são constantes conhecidas.

Existe um forte interesse em estimar parâmetros em modelos baseados no trabalho de Hodgkin e Huxley (1952), começando com o trabalho de Rall (1960). Outros métodos foram posteriormente descritos, entre eles Schierwagen (1990), White, Manis e Young (1992). Em particular, Cox e Li (2000) desenvolveram um método chamado independência de entrada, que busca recuperar os parâmetros do modelo de Hodgkin e Huxley.

Todos os trabalhos mencionados acima são para estimar parâmetros constantes. Sabe-se de poucos trabalhos, experimentais ou teóricos, que obtêm boas estimativas de parâmetros espacialmente distribuídos. Bell e Craciun (2005) desenvolveram um método numérico eficiente para recuperar a condutância não uniforme. Porém aspectos matemáticos de estabilidade, convergência e consistência não foram avaliados neste trabalho. Cox (2006) utilizou o método de multiplicadores de Lagrange num modelo não-linear para recuperar a distribuição não uniforme da condutância. Avdonin e Bell (2013) utilizaram o método de controle de fronteira para resolver o problema inverso de recuperar a condutância. Mandujano e Madureira (2016) utilizaram o

método de Landweber não linear para recuperar uma só condutância, com distribuição espacial não uniforme.

Sejam  $c = C_M(R_i + R_e)$ ,  $f(x) = G_k(x)(R_i + R_e)$ ,  $g(x) = G_{Na}(x)(R_i + R_e)$  e  $h(x) = G_l(x)(R_i + R_e)$ . Suponha que as constantes  $c$ ,  $E_k$ ,  $E_{Na}$  e  $E_l$  sejam conhecidas, e que, as funções  $r$ ,  $p$ , e  $q$  são conhecidas, pode-se medir o valor do potencial da membrana  $V$  na fronteira do eixo espacial. O objetivo do presente trabalho é achar as condutâncias  $f$ ,  $g$  e  $h$  tal que

$$\begin{cases} V_{xx}(t, x) = cV_t(t, x) + f[V(t, x) - E_k] + g[V(t, x) - E_{Na}] + h[V(t, x) - E_l], \\ V(0, x) = r(x) & 0 < x < L, \\ V_x(t, 0) = p(t); \quad V_x(t, L) = q(t) & 0 < t < T. \end{cases} \quad (4)$$

## 2 Sobre problemas inversos

Antes de começar com problemas inversos, primeiro defina-se os operadores que serão utilizados neste trabalho.

**Definição 2.1** Sejam os conjuntos  $\Omega = (0, T) \times (0, L)$ , e definamos  $\partial\Omega = (0, T) \times \{0, L\}$ . O operador não linear  $F$ , com domínio  $D(F) \subset [L^2(0, L)]^3$ , é definido por:

$$\begin{aligned} F : D(F) &\longrightarrow [L^2(0, T)]^2 \\ \vec{\mathcal{G}} &\mapsto F(\vec{\mathcal{G}}) = V|_{\partial\Omega} = (-F_0(\vec{\mathcal{G}}), F_L(\vec{\mathcal{G}})) := (-V(\cdot, 0), V(\cdot, L)) \end{aligned}$$

onde  $\vec{\mathcal{G}} = (f, g, h)$  e  $V$  satisfaz a equação diferencial parcial (4).

O problema inverso consiste em achar  $\vec{\mathcal{G}}$  dado  $V|_{\partial\Omega}$ .

**Definição 2.2** Seja  $F$  o operador da definição 2.1, então a derivada de Gâteaux é definida por:

$$\begin{aligned} F' : [L^2(0, T)]^3 &\longrightarrow [L^2(0, T)]^2 \\ \vec{\mathcal{Q}} &\mapsto F'(\vec{\mathcal{G}})\vec{\mathcal{Q}} = W|_{\partial\Omega} = (-F'_0(\vec{\mathcal{G}}), F'_L(\vec{\mathcal{G}})) := (-W(t, 0), W(\cdot, L)) \end{aligned}$$

onde  $W$  satisfaz a equação diferencial parcial (7). O operador  $F'(\vec{\mathcal{G}})$  é a derivada de Gâteaux na direção de  $\vec{\mathcal{Q}} \in [L^2(0, L)]^3$  no ponto  $\vec{\mathcal{G}} \in D(F)$ . Veja que  $F'(\vec{\mathcal{G}})$  é um operador linear e limitado.

**Definição 2.3** Dado que  $F'(\vec{\mathcal{G}})$  é um operador linear e limitado, pode-se definir o operador adjunto  $F'(\vec{\mathcal{G}})^*$

$$F'(\vec{\mathcal{G}})^* : [L^2(0, T)]^2 \longrightarrow [L^2(0, L)]^3$$

definido por

$$\langle F'(\vec{\mathcal{G}})^*Y, X \rangle_{[L^2(0, L)]^3} = \langle Y, F'(\vec{\mathcal{G}})X \rangle_{[L^2(0, T)]^2}$$

Matematicamente, problemas inversos pertencem à classe de problemas mal-postos. O problema é dito mal-posto se alguma das seguintes condições não é satisfeita: existe solução;

a solução é única; a solução tem uma dependência contínua com os dados de entrada (estabilidade). O trabalho supõe a existência de uma única solução para o problema, mas não sua continuidade (estabilidade). A estabilidade é necessária se desejamos assegurar que pequenas variações nos dados conduzam a pequenas mudanças na solução. Os problemas de instabilidade podem ser controlados pelos métodos de regularização em particular o Método de Landweber não-Linear.

A iteração de Landweber não-linear é definida por

$$\vec{\mathcal{G}}_{n+1}^\delta = \vec{\mathcal{G}}_n^\delta + F'(\vec{\mathcal{G}}_n^\delta)^*(V^\delta|_{\partial\Omega} - F(\vec{\mathcal{G}}_n^\delta)), \quad (5)$$

com  $V^\delta|_{\partial\Omega}$  representando os dados com ruídos, das quais assumimos conhecer o **nível de ruídos**  $\delta$ , satisfazendo  $\|V|_{\partial\Omega} - V^\delta|_{\partial\Omega}\| \leq \delta$ . O operador  $F'(\cdot)^*$  representa a adjunta da derivada de Gâteaux de  $F$ .

A iteração (5) depende fortemente de um critério de parada. Uma alternativa para a escolha do critério de parada é o princípio da discrepância: a iteração é parada no índice  $n_* = n(\delta, V^\delta)$  quando, pela primeira vez,

$$\|V^\delta|_{\partial\Omega} - F(\vec{\mathcal{G}}_{n_*}^\delta)\| \leq \tau\delta < \|V^\delta|_{\partial\Omega} - F(\vec{\mathcal{G}}_n^\delta)\|, \quad 0 \leq n < n_*, \quad \tau > 2\frac{1+\eta}{1-2\eta} > 2. \quad (6)$$

**Teorema 2.1** Suponha que existe uma bola fechada  $B_{2\rho}(\vec{\mathcal{G}}_1) \subset D(F)$  de raio  $2\rho > 0$  e de centro  $\vec{\mathcal{G}}_1 = \vec{\mathcal{G}}_1^\delta$ . Seja  $n_* = n(\delta, V^\delta|_{\partial\Omega})$  escolhido de acordo com o princípio da discrepância (6) e que as hipóteses abaixo sejam satisfeitas:

1. O operador  $F(\cdot)$  é contínuo e o problema possui uma solução  $\vec{\mathcal{G}}_*$  que pertence a  $B_{2\rho}(\vec{\mathcal{G}}_1)$ ,
2. A derivada de Fréchet de  $F(\cdot)$  satisfaz  $\|F'(x)\| \leq 1$ ,  $\vec{\mathcal{G}} \in B_{2\rho}(\vec{\mathcal{G}}_1)$ ,
3.  $\|F(\vec{\mathcal{G}}) - F(\tilde{g}) + F'(\vec{\mathcal{G}})(\vec{\mathcal{G}} - \tilde{g})\| \leq \eta\|F(\vec{\mathcal{G}}) - F(\tilde{g})\|$ ,  $\eta < \frac{1}{2}$ ,  $\vec{\mathcal{G}}, \tilde{g} \in B_{2\rho}(\vec{\mathcal{G}}_1)$ ,

Então a iteração de Landweber  $\vec{\mathcal{G}}_{n_*}^\delta$  converge a uma solução de  $F(\vec{\mathcal{G}}) = V|_{\partial\Omega}$ , quando  $\delta \rightarrow 0$ .

**Prova.** Ver Kaltenbacher, Neubauer e Scherzer (2008), Teorema 2.6.

### 3 Método de Landweber não-Linear de forma algoritmica

Para fins práticos vamos denotar  $\vec{\mathcal{G}}^\delta = \vec{\mathcal{G}}$  e  $V^\delta = V$ . Seja  $\lambda \in \mathbb{R}$ , avaliando os vetores  $\vec{\mathcal{G}}$  e  $\vec{\mathcal{Q}} + \lambda\vec{\mathcal{Q}}$  no operador  $F$ , pela definição da derivada de Gâteux temos

$$F'(\vec{\mathcal{G}}).\vec{\mathcal{Q}} = \lim_{\lambda \rightarrow 0} \frac{F(\vec{\mathcal{G}} + \lambda\vec{\mathcal{Q}}) - F(\vec{\mathcal{G}})}{\lambda} = (-W(t, 0), W(t, L)).$$

Logo, temos

$$\begin{cases} W_{xx}(t, x) - c W_t(t, x) - (f + g + h) W(t, x) = \vec{\mathcal{Q}}.(V - E_k, V - E_{Na}, V - E_l), \\ W(0, x) = 0; & 0 < x < L, \\ W_x(t, 0) = 0; \quad W_x(t, L) = 0 & 0 < t < T. \end{cases} \quad (7)$$

Dado  $\vec{\mathcal{G}}_n$ , defina-se a seguinte equação diferencial parcial

$$\begin{cases} -U_{xx}(t, x) - c U_t(t, x) + (f + g + h) U(t, x) = 0, \\ U(T, x) = 0; & 0 < x < L, \\ U_x(t, 0) = -V(t, 0) + F_0(\vec{\mathcal{G}}_n)) = -\Gamma(t, 0); & 0 < t < T, \\ U_x(t, L) = V(t, L) - F_L(\vec{\mathcal{G}}_n)) = \Gamma(t, L); & 0 < t < T. \end{cases} \quad (8)$$

### 3.1 Solução da adjunta da deriva de Gâteaux

Dado  $\vec{\mathcal{Q}} \in [L^2(0, L)]^3$ , pela iteração de Landweber não linear (5), temos

$$\vec{\mathcal{G}}_{n+1} = \vec{\mathcal{G}}_n + F'(\vec{\mathcal{G}}_n)^*(V|_{\partial\Omega} - F(\vec{\mathcal{G}}_n)),$$

$$\begin{aligned} \langle \vec{\mathcal{G}}_{n+1} - \vec{\mathcal{G}}_n, \vec{\mathcal{Q}} \rangle_{[L^2(0, L)]^3} &= \langle F'(\vec{\mathcal{G}}_n)^*(V|_{\partial\Omega} - F(\vec{\mathcal{G}}_n)), \vec{\mathcal{Q}} \rangle_{[L^2(0, L)]^3} \\ \langle \vec{\mathcal{G}}_{n+1} - \vec{\mathcal{G}}_n, \vec{\mathcal{Q}} \rangle_{[L^2(0, L)]^3} &= \langle F'(\vec{\mathcal{G}}_n)^* \left[ (-V(t, 0), V(t, L)) - (-F_0(\vec{\mathcal{G}}_n), F_L(\vec{\mathcal{G}}_n)) \right], \vec{\mathcal{Q}} \rangle, \\ \langle \vec{\mathcal{G}}_{n+1} - \vec{\mathcal{G}}_n, \vec{\mathcal{Q}} \rangle_{[L^2(0, L)]^3} &= \langle (-V(t, 0), V(t, L)) - (-F_0(\vec{\mathcal{G}}_n), F_L(\vec{\mathcal{G}}_n)), F'(\vec{\mathcal{G}}_n) \cdot \vec{\mathcal{Q}} \rangle_{[L^2(0, T)]^2}, \\ \langle \vec{\mathcal{G}}_{n+1} - \vec{\mathcal{G}}_n, \vec{\mathcal{Q}} \rangle_{[L^2(0, L)]^3} &= \langle (U_x(t, 0), U_x(t, L)), (-W(t, 0), W(t, L)) \rangle_{[L^2(0, T)]^2} \end{aligned}$$

$$\begin{aligned} \int_0^L (\vec{\mathcal{G}}_{n+1} - \vec{\mathcal{G}}_n) \cdot \vec{\mathcal{Q}} dx &= \int_0^T [U_x(t, L)W(t, L) - U_x(t, 0)W(t, 0)] dt \\ &= \int_0^T [U_x(t, x)w(t, x)]_0^L \quad (9) \end{aligned}$$

Multiplicamos a primeira equação da EDP (8) por  $-W$

$$U_{xx}(t, x)W(t, x) + c U_t(t, x)W(t, x) - (f(x) + g(x) + h(x)) U(t, x)W(t, x) = 0$$

Aplicando a integral dupla nos intervalos  $[0, T]$  e  $[0, L]$ , e mais algumas operações algébricas, temos

$$\begin{aligned} \int_0^T [U_x(t, x)w(t, x)]_0^L &= - \int_0^L \int_0^T (W_{xx} - cW_t - (f_n + g_n + h_n)W) U dt dx \\ &= - \int_0^L \int_0^T \vec{\mathcal{Q}} \cdot (V_n - E_k, V_n - E_{Na}, V_n - E_l) U dt dx \\ &= - \int_0^L \left( \int_0^T (V_n - E_k) U, \int_0^T (V_n - E_{Na}) U, \int_0^T (V_n - E_l) U \right) \cdot \vec{\mathcal{Q}} dx \end{aligned}$$

Da última equação e a equação (9), temos a seguinte iteração de Landweber não linear (5)

$$\vec{\mathcal{G}}_{n+1} = \vec{\mathcal{G}}_n + \left( \int_0^T (V_n - E_k) U dt, \int_0^T (V_n - E_{Na}) U dt, \int_0^T (V_n - E_l) U dt \right) \quad (10)$$

### 3.2 Algoritmo de Landweber não linear

A iteração funciona da seguinte maneira: suponha que conhecemos  $\vec{G}_n$ . Então para obter  $\vec{G}_{n+1}$  temos que realizar os seguintes passos:

**Passo 1.** Substituindo  $\vec{G}_n$  na equação diferencial parcial (4), e utilizando o método de diferenças finitas, calculamos  $V_n$ , obtendo portanto  $V_n|_{\partial\Omega} = F(\vec{G}_n) = (F_0(\vec{G}_n), F(\vec{G}_n)_L)$ .

**Passo 2.** Substituindo  $F(\vec{G}_n)$  na equação diferencial parcial (8), e utilizando o método de diferenças finitas obtemos  $U$ .

**Passo 3.** Substituindo  $U$  na equação (10), e utilizando o método de Trapézio calculamos a integral da equação (10), obtendo assim  $F(\vec{G}_{n+1})$ .

Observe que para cada iteração temos que resolver duas equações diferenciais parciais e calcular uma integral.

## 4 Experimentos Computacionais

Nesta seção apresentamos um exemplo computacional, onde “ $nx$ ” é o número de pontos do eixo  $x$ , e utilizando a iteração de Landweber não-Linear (10) obtemos a solução do exemplo. Seja  $I \in \{f, g, h\}$ , definamos o erro para cada  $I$  e o erro total, respectivamente:

$$\begin{aligned} \text{Erro\_}I_n &= \frac{|I - I_n|}{nx} = \frac{|I(x_1) - I_n(x_1)| + |I(x_2) - I_n(x_2)| + \dots + |I(x_{nx}) - I_n(x_{nx})|}{nx}, \\ \text{Erro}_n &= \frac{\text{Erro\_}f_n + \text{Erro\_}g_n + \text{Erro\_}h_n}{3}. \end{aligned}$$

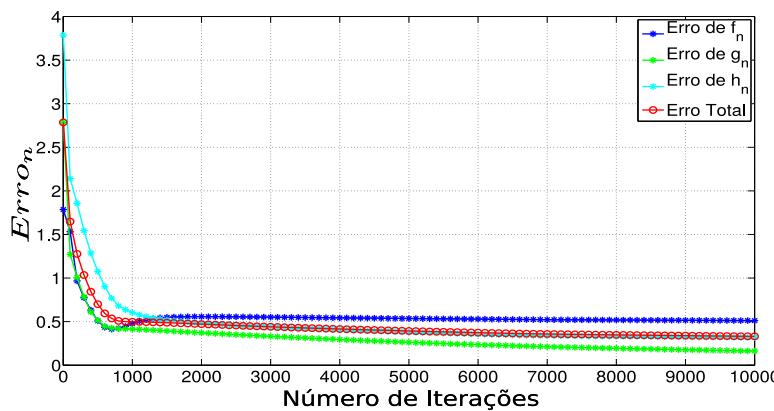
### 4.1 Exemplo

Achar  $\vec{G} = (f, g, h)$  da seguinte equação diferencial parcial, conhecendo  $V|_{\partial\Omega}$ .

$$\left\{ \begin{array}{l} V_{xx}(t, x) = V_t(t, x) + f(x)[V(t, x) - 1] + g(x)[V(t, x) - 2] + h(x)[V(t, x) - 3], \\ V(0, x) = \cos(x + \pi/2) \quad 0 < x < \pi/2, \\ V_x(t, 0) = -\exp(t); \quad V_x(t, L) = 0 \quad 0 < t < 1. \end{array} \right. \quad (11)$$

Para comparar os resultados obtidos, primeiro calcula-se  $V$  utilizando o método de diferenças finitas, para  $\vec{G} = (f(x), g(x), h(x)) = (\sin(x), \cos(x), x^2)$ . Agora considere-se  $\vec{G}$  desconhecido. Utilizando a iteração de Landweber não-Linear (10) obtemos  $\vec{G}$ , dado  $\vec{G}_1 = (x + 1 + \sin(x), x + 2 + \cos(x), x + 3 + x^2)$ .

Na figura 1 vamos mostrar o erro versus o número de iterações. A linha em azul representa o erro cometido na aproximação de  $f$  por  $f_n$ , para a iteração  $n$ . A linha em verde representa o erro cometido na aproximação de  $g$  por  $g_n$ , para a iteração  $n$ . A linha em azul claro representa o erro cometido na aproximação de  $h$  por  $h_n$ , para a iteração  $n$ . A linha em vermelho representa o erro da média de  $f_n, g_n$  e  $h_n$  na aproximação da média de  $f, g$  e  $h$ , para a iteração  $n$ . O algoritmo de Landweber é parado em 10000 iterações, obtendo uma boa aproximação para as funções desconhecidas. Até o momento não encontramos trabalhos, que obtêm os três parâmetros desconhecidos.



**Figura 1:** Erro cometido na iteração n.

## AGRADECIMENTOS

Ao concluir este trabalho, quero agradecer: Aos professores Alexandre L. Madureira e Antonio C. Leitão (UFSC) por sua orientação durante a construção do trabalho, à CAPES pela concessão de bolsa de estudo.

## REFERÊNCIAS

- Avdonin, Sergei, & Bell, Jonathan, 2013. Determining a distributed parameter in a neural cable model via a boundary control method. *Journal of mathematical biology*, vol. 67, n. 1, pp. 123-141.
- Bell, Jonathan, & Craciun, Gheorghe, 2005. A distributed parameter identification problem in neuronal cable theory models. *Mathematical biosciences*, vol. 194, n. 1, pp. 1-19.
- Cox, Steven J., & Ji, Lin, 2000. Identification of the cable parameters in the somatic shunt model. *Biological cybernetics*, vol. 83, n. 2, pp. 151-159.
- Cox, Steven J., 2006. An adjoint method for channel localization. *Mathematical Medicine and Biology*, vol. 23, n. 2, pp. 135-152.
- Hodgkin, Alan L., & Huxley, Andrew F., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, vol. 117, n. 4, pp. 500.
- Kaltenbacher, Barbara, & Neubauer, Andreas, & Scherzer, Otmar, 2008. Iterative regularization methods for nonlinear ill-posed problems. *Walter de Gruyter*, vol. 6.
- Mandujano V., Jemy A., & Madureira , Alexandre L., 2016. Numerical solution of an inverse problem in neurosciense via the nonlinear Landweber method. *XXXVI Congresso Nacional de Matemática Aplicada e Computacional*.
- Rall, Wilfrid, 1960. Membrane potential transients and membrane time constant of motoneurons. *Experimental neurology*, vol. 2, n. 5, pp. 503-532.
- Schierwagen, AK, 1990. Identification problems in distributed parameter neuron models. *Automatica*, vol. 26, n. 4, pp. 739-755.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### SIMULAÇÃO ORIENTADA A OBJETOS DA NUCLEAÇÃO E CRESCIMENTO VIA AUTÔMATO CELULAR

**Jonathas Luis G. Ferreira**

**Wesley Luiz S. Assis**

**Tiago A. Neves**

**Guilherme C. de Oliveira**

[jonathasferreira@id.uff.br](mailto:jonathasferreira@id.uff.br)

[wesleyassis@id.uff.br](mailto:wesleyassis@id.uff.br)

[tneves@id.uff.br](mailto:tneves@id.uff.br)

[guilherme.ori@gmail.com](mailto:guilherme.ori@gmail.com)

Universidade Federal Fluminense - EEIMVR

Av. dos Trabalhadores, No. 420, CEP 27140-360, Vila Santa Cecília, Volta Redonda, RJ, Brasil

**Resumo.** A evolução microestrutural em três dimensões de transformações que envolvem nucleação e crescimento em materiais cristalinos é simulada pelo método do Autômato Celular com vizinhanças de Von Neumann e condições de contorno periódicas. É assumido que toda a nucleação ocorre no começo da transformação (saturação de sítios). O programa usa o conceito de orientação a objetos que torna possível o melhor aproveitamento do código e facilita futuras melhorias. São construídas classes que simulam os processos de transformação e que controlam a entrada e saída de dados. Este trabalho está sendo comparado com as teorias analíticas e resultados da simulação de outros autores. O programa gera dados que podem ser usados para a visualização tridimensional de grãos e, também, para a comparação com equações tradicionais no estudo destas transformações. Os resultados obtidos têm sido coerentes com o atual entendimento destes tipos de fenômenos. É visto como a programação orientada a objetos permite a facilização na implementação das simulações de nucleação e crescimento e na adaptação do código a futuros implementos.

**Palavras-chave:** autômato celular, microestrutura, cinética, orientação a objetos

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUÇÃO

As transformações que envolvem processos de nucleação e crescimento são muito comuns em vários fenômenos da Natureza como transformações de fase e processos de recristalização em materiais cristalinos. Nesses fenômenos, ocorre o surgimento de partículas muito pequenas (núcleos) que crescem até completar a transformação.

Rios e Padilha (2007) observaram que o mecanismo mais comum de surgimento de uma nova fase envolve o rearranjo atômico por meio do qual uma fase mais estável ( $\beta$ ) se forma e se desenvolve na fase matriz ( $\alpha$ ). No entanto, na recristalização não há flutuações atômicas, mas sim uma migração de contorno de grãos ou subgrãos, ou ainda, um coalescimento de subgrãos.

Enquanto a nucleação está relacionada à formação de uma interface entre a matriz e a nova fase, o crescimento diz respeito à migração dessa interface. A migração, por sua vez, depende de uma força motriz e de um mecanismo de migração. Todavia, o estudo termodinâmico desses processos, que busca responder ao porquê ocorre uma transformação, será omitido neste trabalho, sendo apenas considerados os aspectos geométricos e as características cinéticas da transformação.

A teoria clássica de Johnson, Mehl, Avrami e Kolmogorov (JMAK) (citada por Hesselbarth e Göbel, 1991) descreve o crescimento de núcleos. Para o crescimento tridimensional isotrópico de grãos esféricos, a teoria de JMAK leva à seguinte equação para o grau de recristalização (ou fração volumétrica transformada)  $V_V$ :

$$\ln \frac{1}{1 - V_V} = \frac{\pi}{3} \dot{N} G^3 t^4 \quad (1)$$

Nesta equação,  $\dot{N}$  é a taxa de nucleação, considerada constante;  $G$  é a velocidade de crescimento de grãos, também definida como constante; e  $t$  é o tempo.

O tratamento analítico da nucleação e do crescimento de novas fases pode se tornar extremamente complexo devido a várias características que devem ser consideradas e que podem em dúvida algumas das aproximações que geralmente são feitas nesses casos. Sendo assim, a simulação computacional é uma ferramenta de grande valor pois pode reproduzir situações que vão além do tratamento analítico.

Dentre os modelos usados para simulações, o método do autômato celular tem se mostrado adequado para simulação de transformações de fase em geral e de recristalização (Rios et al., 2006). De modo contrário da abordagem de JMAK, o modelo do autômato celular fornece uma descrição desses tipos de transformações que é dependente do tempo e do espaço (Hesselbarth, Göbel, 1991). Tal modelo não está restrito à premissas como distribuição homogênea dos núcleos ou valores estatísticos médios.

Autômatos Celulares têm a vantagem de serem modelos simples para descreverem fenômenos físicos. Processos dependentes do tempo e do espaço são geralmente descritos por equações diferenciais parciais. Todavia, tais equações, na maioria dos casos, precisam ser resolvidas por meio de métodos numéricos. Autômatos Celulares, por sua vez, já são um algoritmo numérico e, portanto, permitem uma análise mais imediata e ampla, além de permitirem a variação de muitas premissas e parâmetros que, muitas vezes, não podem ser trabalhados por meio de equações (Hesselbarth, Göbel, 1991).

Tradicionalmente, modelos de Autômatos Celulares vêm sendo implementados por meio de programação estruturada. Apesar desses programas se mostrarem muito úteis e eficientes para o estudo de transformações de fase, eles possuem as deficiências comuns aos códigos construídos sob o paradigma procedural: a dificuldade de reutilização do código e os problemas de construção e adaptação do código por meio de trabalho em equipe (Deitel, Deitel, 2001).

Um paradigma de programação que tem sido usado para transpor esses problemas é a orientação a objetos. Objetos são partes reutilizáveis de um programa que modelam o mundo real. Tem-se percebido que a utilização de uma abordagem de implementação de códigos orientados a objetos torna os grupos de desenvolvimento mais produtivos do que quando utilizadas somente técnicas de programação estruturada. A orientação a objetos permite a criação de programas mais fáceis de entender, corrigir e modificar (Deitel, Deitel, 2001).

O propósito deste trabalho é apresentar o desenvolvimento de um programa escrito segundo o paradigma de orientação a objetos que utiliza o método do autômato celular para o estudo das transformações de fase. Esse desenvolvimento é motivado pela busca da melhoria no trabalho em conjunto e na produtividade do grupo de estudos do Núcleo de Modelamento Microestrutural da Universidade Federal Fluminense.

A princípio, é assumido que toda a nucleação ocorre no início da transformação e é utilizado o tipo de vizinhanças de Von Neumann. Os dados resultantes das simulações são comparados com as previsões de equações analíticas e com o trabalho de Rios et al. (2006).

## 2 Desenvolvimento do Código

O programa é formado por classes que buscam simular o comportamento de objetos que representam um material que passará por um processo de nucleação e crescimento. As principais classes são: Dados, Matriz, Nucleação, Crescimento, Autômato e Saída.

Foi utilizada a linguagem C++ por suportar a orientação a objetos e ser uma linguagem amplamente utilizada. C++ possui vasta documentação e elevada contribuição de usuários de todo o mundo.

### 2.1 A classe “Dados”

A classe “Dados” é responsável pela coleta dos parâmetros que definirão as transformações durante a execução. Optou-se pela utilização de um arquivo de entrada padronizado que contém os parâmetros necessários ao programa. Realiza-se, então, a criação de um objeto que armazena os dados e os torna acessíveis a qualquer momento sem a necessidade de releitura do arquivo de entrada. Os principais dados armazenados são: tipo e tamanho da malha, tipo de nucleação, número de núcleos iniciais, tipo de vizinhança e critérios de parada do código.

### 2.2 A classe “Matriz”

Esta classe é a responsável pela criação da malha onde serão realizadas as transformações. A princípio, foi implementada uma malha tridimensional do tipo cúbica.

Como forma de melhoria de desempenho, a malha é construída em um vetor unidimensional ao invés de usar uma matriz tridimensional. A dimensão deste vetor é igual ao número

de elementos da matriz tridimensional equivalente. Embora essa escolha implique em algoritmos mais sofisticados para realização de algumas tarefas apresentadas a seguir, os ganhos com desempenho, inclusive relacionados à paralelização, a tornam justificável.

O método do autômato celular trabalha com a ideia de estados transformados na matriz. Sendo assim, dois vetores são construídos: o primeiro para representar os estados da malha antes dos passos de transformação e o segundo para armazenar as transformações durante o passe. No instante inicial, ambos os vetores possuem todos os valores iguais a zero, representando a fase não transformada.

Além disso, a classe “Matriz” contém os métodos relacionados à medições como fração volumétrica transformada,  $V_V$ , e área interfacial entre região transformada e não transformada por unidade de volume,  $S_V$ .

## **2.3 A classe “Nucleação”**

A classe “Nucleação” altera o vetor de estado conforme o tipo de nucleação e o número de núcleos iniciais declarados na entrada do programa. Inicialmente foi implementado um algoritmo para simular a nucleação aleatória, homogeneamente distribuída pela malha e que ocorre totalmente no início da transformação (saturação de sítios).

Para a distribuição dos núcleos, foi usada a geração de números pseudoaleatórios com uma distribuição uniforme disponível na biblioteca “Random” de C++. Conforme as posições no vetor de estado são sorteadas, estas posições recebem um valor sequencial de 1 até  $n$ , onde  $n$  é o número total de núcleos. Se alguma posição onde haja um núcleo for novamente escolhida, o sorteio é repetido.

## **2.4 As classes “Crescimento” e “Autômato”**

A classe “Crescimento” determina o tipo de crescimento que os núcleos terão na malha. Neste trabalho, foi implementada a classe “Autômato” para o crescimento por meio do método do autômato celular.

Um autômato celular tem as seguintes características (Hesselbarth, Göbel, 1991):

1. A geometria das células.
2. Os estados que uma célula pode possuir.
3. A definição da vizinhança da célula.
4. As regras de transição que determinam o estado das células no próximo passo com base em seu estado atual e no estado atual das células vizinhas.

Conforme já mencionado, a geometria das células implementada é a cúbica. Os estados que a célula pode possuir são o estado não transformado, representado pelo valor zero, e o estado transformado, representado por valores de 1 até  $n$ , sendo  $n$  o número total de núcleos. Cada núcleo possui um valor e esse valor é propagado para a vizinhança conforme ocorre o crescimento.

A vizinhança implementada foi a de Von Neumann que é composta pelas células imediatamente próximas nas direções axiais. Para simular uma amostra infinita, são usadas condições

de contorno periódicas, isto é, no caso de uma célula que se encontre em um dos planos limites da malha, é considerada sua vizinha a célula pertencente ao plano oposto na mesma direção.

A regra de transição é uma verificação do estado atual da célula e de suas vizinhas. Se a célula já estiver transformada, a verificação passa para a próxima. Caso contrário, verifica-se se alguma das células vizinhas está transformada e, se estiver, a célula analisada recebe o valor da vizinha transformada.

Como é possível haver mais de uma célula vizinha transformada, a ordem de checagem das vizinhas é aleatória e a primeira vizinha transformada encontrada é a escolhida. Isso evita que haja uma orientação preferencial quando duas regiões transformadas se encontram e suas fronteiras tendem, portanto, a serem próximas a planos.

Em cada intervalo de tempo, toda a malha é atualizada conforme a regra de transição. O vetor de estados que armazenava os valores de antes do passe recebe os valores do vetor que registrou as transformações e o processo se repete. O crescimento para quando algum valor preestabelecido de fração volumétrica transformada ou número de passes é atingido. Durante as iterações, os valores de  $V_V$  e  $S_V$  são calculados e registrados.

## 2.5 A classe “Saída”

Esta última classe é responsável pela saída dos dados gerados pelo programa em forma de arquivos de texto. O programa gera como saída os estados da malha no final da transformação ou em alguma etapa e os valores de  $V_V$  e  $S_V$  ao longo dos passes. Esses dados podem ser utilizados em algum programa gráfico.

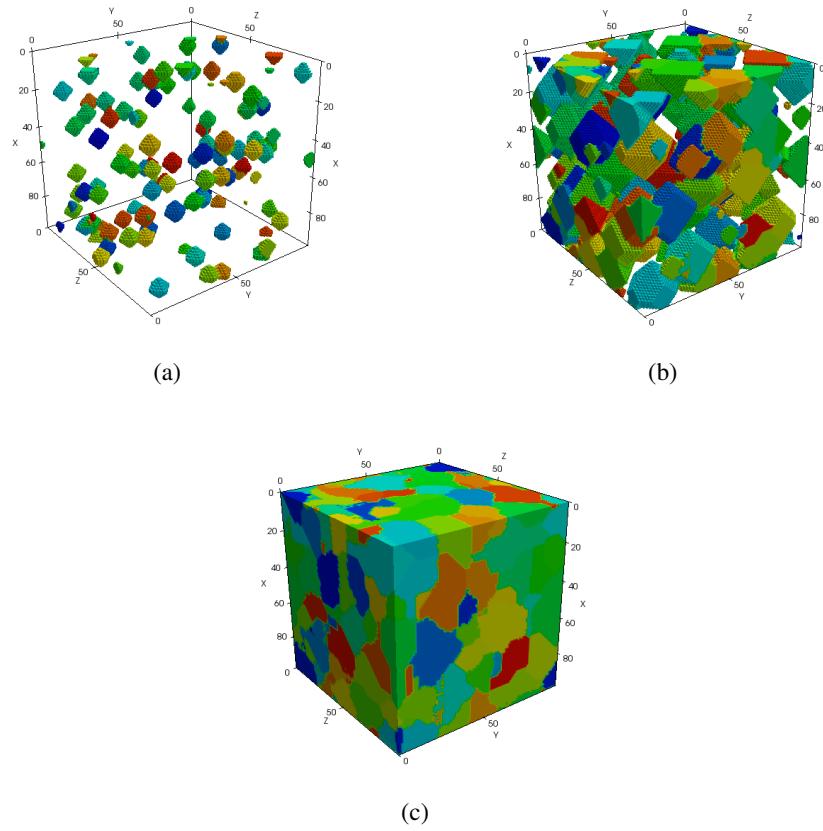
## 3 Resultados e Discussão

O programa gera como saída valores de estado para cada posição na malha. Como exemplo, foi simulada uma transformação em uma malha de dimensão  $100 \times 100 \times 100$  com 100 núcleos iniciais. O código foi executado até que 100% da malha fosse transformada. Foram realizadas 43 iterações para que o processo se concluisse.

A Figura (1) apresenta a construção gráfica desta malha e a evolução dos grãos durante o crescimento. A coloração é aleatória. Pode-se perceber a presença de grãos poligonais e equiaxiais típicos de uma estrutura recristalizada. Os grãos crescem e se chocam gerando contornos de grão que tendem a ser retos ou serrilhados. Esse aspecto deve-se à aleatoriedade na escolha dos vizinhos durante a transformação das células. Além disso, pode-se perceber que os grãos no início da transformação têm a forma octaédrica, característica do método do autômato celular. Também pode-se notar que, durante o crescimento, o uso de condições de contorno periódicas proporciona que grãos que crescem próximo aos limites da malha continuem crescendo na extremidade oposta.

O programa também gera como saída valores de fração volumétrica transformada,  $V_V$ , e área interfacial entre região transformada e não transformada por unidade de volume,  $S_V$ , ao longo das iterações que podem ser comparados com previsões analíticas. Cada iteração é considerada como uma unidade de tempo. Todas as unidades serão omitidas.

Rios et al. (2006) demonstraram que, para a nucleação do tipo de saturação de sítios, a teoria de JMAK leva a uma expressão, dada pela Eq. (2), para o cálculo da fração volumétrica,  $V_V$ , em função do número de núcleos por unidade de volume,  $N_V$ , e do tempo,  $t$ .



**Figura 1:** Estrutura de grãos gerada pela simulação computacional. **(a)** 5%, **(b)** 50% e **(c)** 100% de transformação.

$$V_V = 1 - \exp\left(-\frac{4}{3}N_V t^3\right) \quad (2)$$

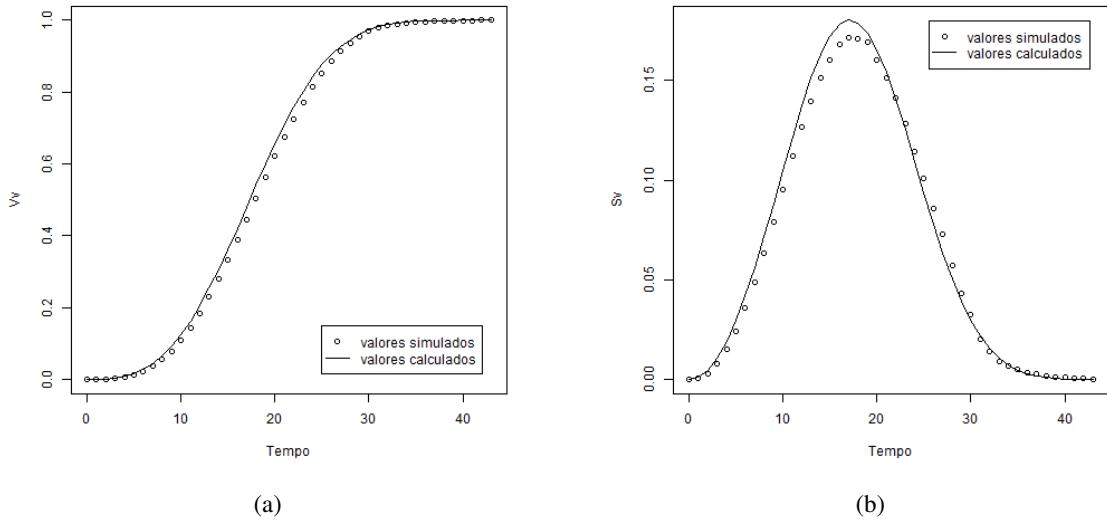
Além disso, Rios et al. (2006) também obtiveram uma expressão, dada pela Eq. (3), para o cálculo da área interfacial entre região transformada e não transformada por unidade de volume,  $S_V$ , em função do número de núcleos por unidade de volume,  $N_V$ , e do tempo,  $t$ .

$$S_V = 12N_V t^2 \exp\left(-\frac{4}{3}N_V t^3\right) \quad (3)$$

Desta forma, a Figura (2) apresenta a comparação entre a  $V_V$  e a  $S_V$  calculadas pela Eq. (2) e pela Eq. (3) e as obtidas pela simulação do autômato celular.

Pode-se perceber que há bom acordo entre os valores da Eq. (2) e da Eq. (3) e os valores da simulação. É visto que o crescimento pode ser dividido em três partes principais: um período inicial de lenta transformação, seguido por um onde a taxa de crescimento é consideravelmente mais alta, terminando em um período onde a taxa de transformação volta a ser lenta até terminar.

O algoritmo utiliza valores pseudoaleatórios na nucleação e na transformação de regiões onde serão os contornos de grão. Uma não uniformidade nesses valores pode ser o fator respon-



**Figura 2: Comparação entre valores calculados analiticamente e obtidos na simulação: a) Fração volumétrica,  $V_V$ , em função do tempo e b) área interfacial entre região transformada e não transformada por unidade de volume,  $S_V$ , em função do tempo.**

sável pela pequena discrepância observada entre os valores calculados e simulados da Figura (2).

Uma outra forma de analisar o crescimento de grãos é chamada de caminho microestrutural e consiste em escrever  $S_V$  em função de  $V_V$ . Rios et al. (2006) utilizaram a Eq. (4) para isso.

$$S_V = 3(36N_V)^{\frac{1}{3}}(1 - V_V) \left[ \ln \left( \frac{1}{1 - V_V} \right) \right]^{\frac{2}{3}} \quad (4)$$

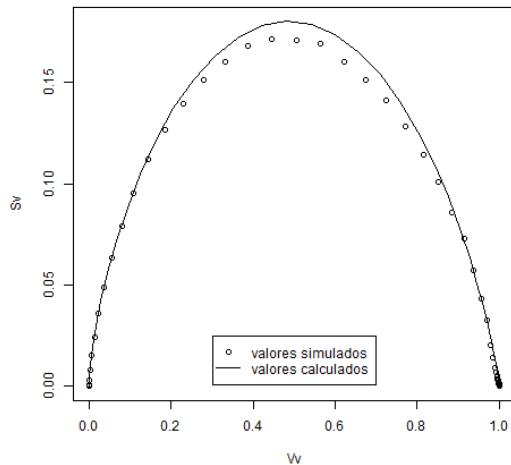
Assim, a Figura (3) apresenta a comparação entre o caminho microestrutural calculado e o simulado.

Há bom acordo entre os valores da Eq. (4) e os valores da simulação. Os valores simulados aparecem um pouco abaixo da curva calculada devido à interação entre os grãos no início do crescimento, diminuindo a quantidade de área interfacial entre as regiões recristalizadas e não recristalizadas ( $S_V$ ) no decorrer da transformação (Oliveira et al., 2006).

Tanto os resultados apresentados na Figura (2) e na Figura (3) estão em conformidade com os obtidos por Rios et al. (2006). Como mencionado por esses autores, o bom acordo entre os valores calculados e simulados mostram que o método do Autômato Celular é confiável para este tipo de transformação.

## 4 Conclusão

A programação orientada a objetos se mostrou bastante adequada na implementação de um algoritmo de autômato celular para simulação de fenômenos de nucleação e crescimento. O programa gerou resultados coerentes com as previsões analíticas e com trabalhos de outros autores.



**Figura 3: Caminho microestrutural: Área interfacial entre regiões transformada e não transformada por unidade de volume,  $S_V$ , contra Fração volumétrica,  $V_V$ .**

O uso do paradigma de orientação a objetos permite que novas funcionalidades sejam incorporadas ao programa de forma bem mais simples que o convencional. A simulação de outros tipos de nucleação, geometrias das malhas e vizinhanças das células é pretendida para trabalhos futuros. A adaptabilidade do programa ao recebimento de novas funções tem sido um dos objetivos principais desde sua criação.

### Agradecimentos

Os autores agradecem o apoio da Pró-Reitoria de Assuntos Estudantis (PROAES) da Universidade Federal Fluminense, do Mestrado em Modelagem Computacional em Ciência e Tecnologia e do Núcleo de Modelamento Microestrutural.

### REFERÊNCIAS

- Deitel, H., & Deitel, P., 2001. *C++ Como Programar*, 3 ed.. Bookman.
- Hesselbarth, H., & Göbel, I., 1991. Simulation of Recrystallization by Cellular Automata. *Acta Metallurgica et Materialia*, vol. 39, n. 9, pp. 2135 à 2143.
- Oliveira, V., Oliveira, L., Rios, P., & Castro, J., 2006. Desenvolvimento de um Código em 3D para Simular a Recristalização pelo Método do Autômato Celular. *Tecnologia em Metalurgia e Materiais*, vol. 2, n. 4, pp. 34 à 39.
- Rios, P., Oliveira, V., Pereira, L., Pereira, M., & Castro, J., 2006. Cellular Automata Simulation of Site-saturated and Constant Nucleation Rate Transformations in Three Dimensions. *Materials Research*, vol. 9, n. 2, pp. 223 à 230.
- Rios, P., & Padilha, A., 2007. *Transformações de Fase*. Artliber.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### ANÁLISE DO EFEITO DA PRECISÃO FINITA NO ALGORITMO ADAPTATIVO SIGMOIDAL

**José de R. S. Fonseca**

**Ewaldo E. C. Santana**

**André C. B. Cavalcante**

**Allan K. Barros**

ribamar.slz@hotmail.com

UFMA - Universidade Federal do Maranhão.

Av. dos Portugueses, 1966, CEP 65080-805, São Luís-MA, Brasil

**Raimundo C. S. Freire**

UFCG - Universidade Federal de Campina Grande

Rua Aprigio Veloso, 882 - B. Univ., CEP 58429-900, Campina Grande-PB, Brasil

**Resumo.** A filtragem adaptativa constitui atualmente uma ferramenta importante no processamento estatístico de sinais, especialmente quando é necessário processar sinais de ambientes com estatísticas desconhecidas que variam com o tempo. O estudo de filtragem adaptativa foi impulsionado com o desenvolvimento do algoritmo Least Mean Square (LMS) em 1960. Novos algoritmos tem surgido, entre eles, o algoritmo Sigmoidal (SA) que se apresentou superior ao LMS, em relação à taxa de convergência e o desajuste em implementação em precisão infinita. Nos dispositivos de hardwares, tais como DSPs, Microcontroladores e FPGAs, os algoritmos adaptativos são implementados em precisão finita, em geral, com aritmética de ponto fixo. Alguns efeitos nessa implementação podem afetar o seu desempenho. Em última análise, levar à divergência devido aos erros de quantização, especificados na aproximação dos valores das variáveis de seus valores originais. Este artigo propõe analisar o desempenho do algoritmo adaptativo Sigmoidal (SA) em precisão finita, quando implementado utilizando aritmética de ponto fixo. Em particular, analisar sua curva de desempenho e o desajuste, comparando-os em diferentes comprimentos de palavras (número de bits). Os resultados apresentados neste artigo propõe uma aproximação em série de Taylor do gradiente da função de custo  $\ln(\cosh e)$  do algoritmo SA para implementação em precisão finita. Analisamos a sua curva de desempenho para diferentes comprimentos de palavras. Mostra-se que o algoritmo apresenta estabilidade em seu desempenho em relação à convergência, para diferentes comprimentos de palavras, e que o aumento no nível do desajuste em estado estacionário é sensível ou influenciado pela quantização dos valores das variáveis envolvidas nos cálculos desse algoritmo.

**Palavras-chave:** Filtros Adaptativos, Aritmética de ponto-fixo, Sigmoidal

---

EAMC 2017

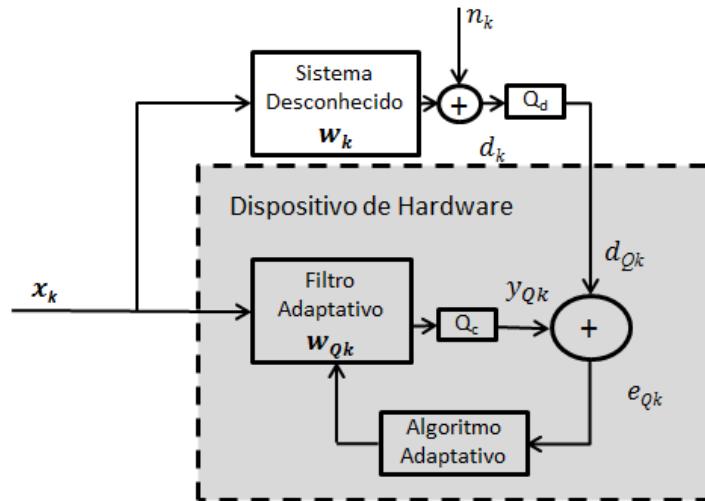
Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUÇÃO

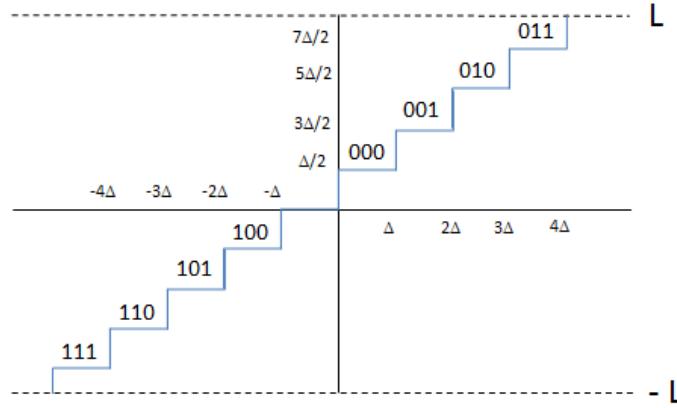
Erros de quantização podem degradar o desempenho de um filtro adaptativo de várias maneiras. Por exemplo, eles podem afetar a estabilidade do filtro e, em última análise conduzir à sua divergência. Também podem degradar o desempenho em estado estacionário do filtro fazendo-o atingir um erro médio quadrático maior do que o esperado a partir de uma análise de precisão infinita (Gupta et al, 2000). A degradação do desempenho tende a ser mais grave para o algoritmo *Recursive Least Square* (RLS) convencional podendo ser mais sensível a erros de precisão finita e divergir como resultado (Sayad, 2013) em oposição ao algoritmo *Least Mean Square* (LMS) (Gupta et al, 2000). Isto porque para filtros do tipo LMS, erros gradiente de ruído são mais significativos do que os erros de precisão finita durante a fase transitória. No entanto, em estado estacionário, um significativo excesso de erro quadrático pode ser formado (Verhaegen, 1989). Trabalhos recentes foram publicados em que analisam filtros adaptativos em precisão finita utilizando aritmética de ponto fixo, como em (Maluenda, 2005) que apresenta nova modelagem para o algoritmo LMS, (Meher, 2014) traz uma arquitetura eficiente para o algoritmo LMS e deriva expressões de erros de estados. Algoritmos de filtragem adaptativa com base em erro não-médio-quadrático tem sido sugeridos por vários autores, quer como uma forma de melhorar o desempenho de adaptação destes algoritmos para ambientes estatísticos particulares, ou como forma de simplificar o cálculo do algoritmo em hardware (Douglas Meng, 1994). Desta forma propomos uma análise de um algoritmo baseados em erro não-médio-quadrático dado em precisão finita, denominado Sigmoidal (SA) (Santana, 2006).

## 2 FILTRAGEM ADAPTATIVA E OS EFEITOS DA PRECISÃO FINITA

Considere o diagrama em blocos mostrado na Figura 1, referente ao problema de identificação de sistemas em filtragem adaptativa.



**Figura 1:** Identificação de sistema implementado em precisão finita. Onde  $x_k$  designa o sinal de entrada. Os blocos  $Q_d$  e  $Q_c$  são quantizadores de  $b$  bits. Assim, o quantizador,  $Q_d$ , usa  $b_d$  bits para os dados de entrada e  $Q_c$  quantiza os coeficientes em  $b_c$  bits. O sinal de saída  $y_{Qk}$  indica a forma quantizada. O sinal desejado definido por  $d_k$  do sistema desconhecido e  $d_{Qk}$  sua forma quantizada. O ruído de medição  $n_k$  e o sinal de erro quantizado  $e_{Qk}$ . Os coeficientes  $w_k$  do sistema desconhecido e  $w_{Qk}$  do filtro adaptativo.(Gupta, 2000)



**Figura 2:** Representação esquemática de níveis de quantização para o caso de  $b = 2$  bits e  $\Delta = L/4$ . Consideramos um valor  $L$  limiar. Usando  $b$  bits, o intervalo de  $[0, L]$  pode ser dividido em  $2^b$  níveis de largura  $\Delta = L/2^b$  cada. Se, além disso, usamos 1 (um) bit de sinal, os resultantes  $(b + 1)$  bits dividem o intervalo  $[-L, L]$  em  $2^{2b}$  níveis. Aqui assumimos operação de arredondamento.(Sayde, 2013)

O filtro adaptativo Sigmoidal (SA) tem como função de custo  $F_k(e) = \cosh(\alpha e)$  aplicada sobre o erro. Essa função é uma não linearidade par, contínua, simétrica e não tem mínimo local, apenas o mínimo global (Santana, 2006).

A partir desta função, pode-se gerar uma família de funções,  $\cosh(\alpha e)$ , multiplicando o argumento  $e$  por um inteiro positivo  $\alpha$ , ou seja, quando aumenta  $\alpha$ , aumenta-se a inclinação da superfície de desempenho, onde as curvas destas funções tem inclinações maiores do que a curva da função quadrática do algoritmo Least Mean Square (LMS) (Santana, 2006). Para desenvolver o algoritmo SA, estimamos o gradiente de  $F_k(e)$ . Onde ficamos com a Eq. (1):

$$\hat{\nabla} F_k(e) = -\alpha \tanh(\alpha e) \mathbf{x}_k \quad (1)$$

Logo o algoritmo adaptativo será dado pela Eq. (2):

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \alpha \tanh(\alpha e) \mathbf{x}_k \quad (2)$$

A Equação (2) caracteriza o algoritmo adaptativo Sigmoidal (SA). Onde o  $\mu$  é uma constante que regula a velocidade e a estabilidade da adaptação (Santana, 2006).

Considerando a Figura 1, e a estrutura do algoritmo SA, as equações que definem a atualização dos coeficientes do filtro adaptativo em precisão finita são:

$$d_{Qk} = (d_k + n_k)_Q \quad (3)$$

$$y_{Qk} = (\mathbf{w}_{Qk} \mathbf{x}_k)_Q \quad (4)$$

$$e_{Qk} = (d_{Qk} - y_{Qk})_Q \quad (5)$$

$$\mathbf{w}_{Q(k+1)} = (\mathbf{w}_{Qk} + \mu \alpha \tanh(\alpha e_{Qk}) \mathbf{x}(k))_Q \quad (6)$$

Note que  $Q$  é a operação de quantização. Todos os escalares e elementos do vetor do algoritmo SA irá desviar os seus valores originais, devido aos efeitos da quantização, Figura

2. Podemos definir os erros ou ruídos de quantização nas variáveis e quantidades relacionadas com algoritmo SA como segue:

$$n_\theta = \theta_k - \theta_{Qk} \quad (7)$$

Onde  $\theta$  representa a variável ou quantidade relacionada no algoritmo e o subscrito  $Q$  indica a sua forma quantizada.

### 3 ANÁLISE DE DESEMPENHO

Invocamos uma série de pressupostos sobre a natureza estatística dos dados de entrada e ruído característico do modelo, comumente utilizados na literatura. Nesta análise, nós linearizamos uma função não linear arbitrária  $f(e_k)$  sobre um ponto de funcionamento adequado (amostra de ruído  $n_k$ ) e, em seguida, prosseguimos com a análise de covariâncias em estado estacionário.

Pressupostos:

A1: Assume-se que  $\mathbf{x}_k$  é um sinal aleatório discreto com  $-1 < \mathbf{x}_k \leq 1$  não sofre quantização e cada  $\mathbf{x}_k$  vetor de dados de entrada é estatisticamente independente de todos os vetores de dados anteriores  $\mathbf{x}_j$ ,  $j < k$ .

A2: A sequência de ruído amostrado por medição ou erros de quantização gerado em qualquer etapa no algoritmo é considerado como uma variável aleatória com média zero, que é estatisticamente independente de quaisquer outros erros e/ou quantidades relacionadas.

A3: O parâmetro de tamanho do passo  $\mu$  deve ser escolhido suficientemente pequeno de tal modo que o excesso de erro médio quadrático na iteração  $k$  seja muito menor do que o mínimo erro médio quadrático perto de convergência.

B1: As variações dos erros dependem do tipo da quantização e aritmética que vai ser utilizada na implementação do algoritmo. Em nosso caso utilizamos aritmética de ponto fixo. Utilizando aritmética de ponto fixo o erro pode ser modelado em processo estocástico com média zero, com variância dada por Eq. (8) (Diniz, 2013):

$$\sigma^2 = C \frac{2^{-2b}}{12} \quad (8)$$

Onde  $C$  é uma constante.

A saída do sistema desconhecido pode ser expressa pela Eq. (9):

$$d_k = \mathbf{w}^{*t} \mathbf{x} + n_k \quad (9)$$

Onde  $\mathbf{w}^*$  é o coeficiente ótimo.. Supõe-se que o sinal de entrada e o sinal desejado não sofre quantização, de modo que apenas a quantização interna de computação são tidas em conta.

Podemos expressar o erro em sua forma quantizada pela Eq. (10)

$$e_{Qk} = n_k - n_e - \mathbf{v}_k^t \mathbf{x}_k \quad (10)$$

Onde definimos  $n_e$  como o ruído devido a quantização e  $\mathbf{v}_k^t$  é o vetor desvio do peso dado por:  $\mathbf{v}_k = \mathbf{w}_k - \mathbf{w}^*$ .

Expressando a não linearidade  $f(e_k)$  em uma expansão em série de taylor sobre o valor do ruído, obtemos: Eq.(11).

$$\begin{aligned} f(e_k) &= \sum_{i=0}^{\infty} \frac{f^i}{i!} (-n_{ek} - \mathbf{v}_k^t \mathbf{x}_k)^i \\ &= f(n_k) - f'(n_k)(n_e + \mathbf{v}_k^t \mathbf{x}_k) + \frac{f''(n_k)}{2}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^2 - \frac{f'''(\delta)}{6}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^3 \end{aligned} \quad (11)$$

Onde  $f^i$  denota a derivada ( $i$ ) da função  $f$ , temos utilizado a forma derivada para representar o termo restante de taylor tal que  $\delta$  encontra-se no intervalo  $[0, n_{ek} + \mathbf{v}_k^t \mathbf{x}_k]$ . Tecnicamente esta representação de série de taylor limita a não linearidade  $f(.)$  para uma classe restrita de funções que são diferenciáveis até a terceira ordem ao longo de toda extensão do erro  $e_k$ .

Agora vamos examinar o comportamento da Eq. (11) no caso em que o vetor de erro peso é pequeno, de tal modo que o termo  $\frac{f'''(\delta)}{6}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^3$  pode ser desprezado em relação à menor ordem do termo. Em seguida a algoritmo se torna: Eq. (12)

$$\mathbf{v}_{Q(k+1)} = \mathbf{v}_k + \mu[f(n_k)\mathbf{x}_k - f'(n_k)(n_e + \mathbf{v}_k^t \mathbf{x}_k)\mathbf{x}_k + \frac{f''(n_k)}{2}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^2 \mathbf{x}_k] - \mathbf{n}_v \quad (12)$$

Em que  $\mathbf{n}_v$  é um vetor de ruído devido à quantização do vetor peso  $\mathbf{v}_k$ .

A fim de determinar equações para desajustes, utilizamos a Eq. (12) e multiplicando ambos os lados por sua transposta. Tomando a expectativas e invocando os pressupostos A1, A2. Assumindo o vetor erro de peso pequeno, podemos ignorar os termos de ordem  $O^3$ .

$$\begin{aligned} E[\mathbf{v}_{Q(k+1)} \mathbf{v}_{Q(k+1)}^t] &= E[\mathbf{v}_{Qk} \mathbf{v}_{Qk}^t] - 2\mu[f'(n_k)]\mathbf{R}E[\mathbf{v}_{Qk} \mathbf{v}_{Qk}^t] + \mu^2 E[f^2(n_k)]\mathbf{R} \\ &\quad + \mu^2 E[f^2(n_k)]\mathbf{R}\sigma_e^2 + \mu^2 E[f'^2(n_k)]E[\mathbf{x}_k \mathbf{x}_k^t \mathbf{v}_k \mathbf{v}_k^t \mathbf{x}_k \mathbf{x}_k^t] \\ &\quad + \mu^2 E[f(n_k)f''(n_k)]E[\mathbf{x}_k \mathbf{x}_k^t \mathbf{v}_k \mathbf{v}_k^t \mathbf{x}_k \mathbf{x}_k^t] + \sigma_v^2 I \end{aligned} \quad (13)$$

Para  $R = E[\mathbf{x}_k \mathbf{x}_k^t]$ ,  $E[\mathbf{n}_v \mathbf{n}_v^t] = \sigma_v^2 I$  e  $E[n_e^2] = \sigma_e^2$ .

Podemos desprezar o quinto e sexto termo na Eq. (13), porque se  $\mu$  e  $\mathbf{v}_{Qk}$  são pequenos, pressuposto A3, esses termos são muito menores do que os outros termos. Podemos utilizar a transformação de similaridade para expressar a Eq.(13) como um conjunto de equações iterativas dissociadas. Definindo a matriz unitária  $\mathbf{L}$  de autovetores e a matriz diagonal  $\Lambda$  de autovalores podemos expressar a matriz de valor real e covariância simétrica como  $\mathbf{R} = \mathbf{L} \Lambda \mathbf{L}^t$ . Assim, definindo o vetor de erro de peso no domínio de coordenadas transformadas como  $\tilde{\mathbf{v}}_{Qk} = \mathbf{L}^t \mathbf{v}_{Qk}$ , e para  $f(n_k) \gg f'(n_k)$ . Obtemos a Eq. (14).

$$E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] = \frac{1}{2E[f'(n_k)]} (\mu E[f^2(n_k)] + \frac{\sigma_v^2}{\mu}) I \quad (14)$$

Que possui a solução de  $E[\tilde{\mathbf{v}}_{Qk}\tilde{\mathbf{v}}_{Qk}^t] = E[\mathbf{v}_{Qk}\mathbf{v}_{Qk}^t]$ . Sabe-se que o excesso de erro médio quadrático na saída do filtro é causado por flutuações aleatórias (*i*) nos coeficientes de peso durante a adaptação. No qual assumimos a independência de  $\mathbf{x}_k$  e  $\mathbf{v}_k$ , bem como a natureza não correlacionada do vetor de erro de peso. Eq. (15) e (16).

$$E[(\mathbf{v}_{Qk}^t \mathbf{x}_{Qk})^2] = E\left[\left(\sum_{i=1}^N v_{i,k} x_{i,k}\right)^2\right] \quad (15)$$

$$E[(\mathbf{v}_{Qk}^t \mathbf{x}_{Qk})^2] = \frac{1}{2E[f'(n_k)]} (\mu E[f^2(n_k)] + \frac{\sigma_v^2}{\mu}) E[x_{i,k}^2] \quad (16)$$

O desajuste é definido como excesso de erro médio quadrático sobre o erro mínimo médio quadrático e dado por: Eq. (17).

$$M = \frac{E[(\mathbf{v}_{Qk}^t \mathbf{x}_{Qk})^2]}{E[n_k^2]} \quad (17)$$

$$M = \frac{tr[\mathbf{R}]}{2E[f'(n_k)]\sigma_v^2} (\mu E[f^2(n_k)] + \frac{\sigma_v^2}{\mu}) \quad (18)$$

Onde na Eq. (18) o  $tr[.]$  denota a operação de traço  $E[x_{i,k}^2] = tr[\mathbf{R}]$  e  $E[n_k^2] = \sigma_v^2$  variância do ruído de medição.

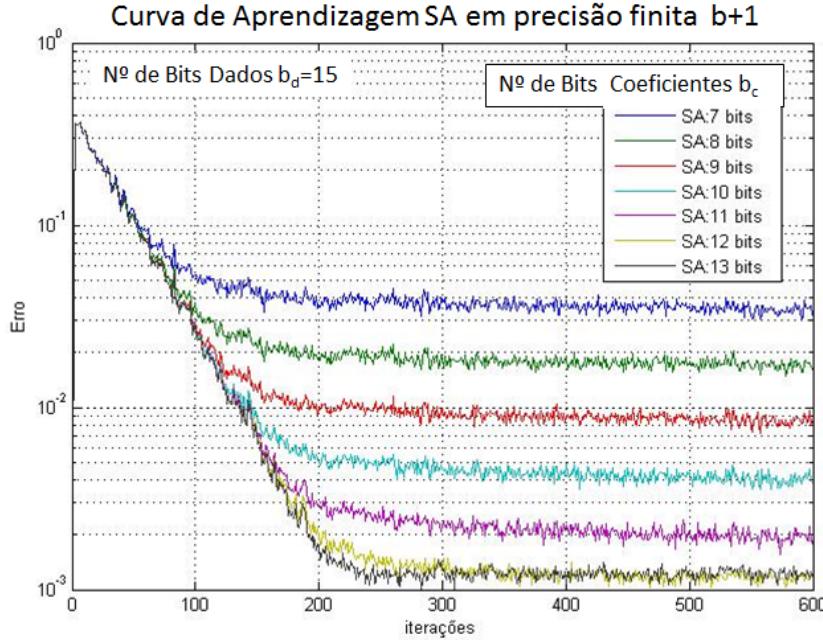
Para  $\frac{\sigma_v^2}{\mu} = 0$  não tem o ruído devido a quantização.

Assim, estas equações podem representar uma generalização de muitos resultados específicos baseados em uma família de algoritmos adaptativos baseados em uma função de custo de erros não-médio-quadrático agindo sobre o erro utilizando a técnica do gradiente estocástico desenvolvida em precisão finita. Em que podemos analisar o desajuste em estado estacionário em sua forma quantizada.

## 4 RESULTADOS E SIMULAÇÕES

Para utilizar a função  $tanh(e)$  em precisão finita utilizando aritmética de ponto fixo, foi necessário realizarmos uma aproximação desta função em série de Taylor até o quinto grau.

Aplicamos o algoritmo adaptativo em um problema de identificação de sistemas desconhecido, Figura (1) onde temos uma planta representada pela função de transferência polinomial  $P(z) = 0.2037z^{-1} + 0.5926z^{-2} + 0.2037z^{-3}$ , cuja saída é corrompida por um ruído,  $n_k$ . Nossa objetivo é encontrar, de modo adaptativo, um modelo de planta,  $\hat{P}(z)$  em precisão finita utilizando aritmética de ponto fixo. Com este objetivo, utilizamos o algoritmo SA em diferentes comprimentos de palavras. O sinal de entrada foi simulado como um sinal aleatório uniformemente distribuído, limitado no intervalo  $[-1, 1]$ . Como ruído utilizou um sinal de ordem  $10^{-3}$  com distribuição uniforme de probabilidades. O sinal desejado foi posto como a soma do sinal de entrada mais ruído. O parâmetro tamanho do passo para o algoritmo SA foi  $\mu_{SA} = 0.01$  e utilizamos para o parâmetro  $\alpha$  com o valor 3. Além do bit das variáveis utilizamos aqui um bit de sinal, resultante  $(b+1)$ . Foram realizadas 100 simulações de Monte Carlo.



**Figura 3: Curva de aprendizagem SA em precisão finita para  $b_d = 15\text{bits}$  e  $C = 0,0590$  em diferentes comprimentos de palavras**

**Tabela 1: Desajuste e MSE do SA em Precisão Finita para  $b_d = 15\text{bits}$  e  $C = 0,0590$**

Nº de Bits $b_c$	Experimental	Teorico	MSE
7	4,4654	1,9999	$4,4654 \cdot 10^6$
8	1,1144	0,5000	$1,1145 \cdot 10^6$
9	0,2570	0,2500	$2,5703 \cdot 10^7$
10	0,0705	0,0625	$7,0529 \cdot 10^8$
11	0,0187	0,0156	$1,8703 \cdot 10^8$
12	0,0064	0,0078	$6,4395 \cdot 10^9$
13	0,0068	0,0078	$6,7681 \cdot 10^9$

## 5 DISCUSSÃO

Este resultados sugerem um aumento do nível de erro ou ruído adicional devido à quantização, para alguns comprimentos de palavras (7, 8, 9, 10 e 11 bits). Que dependem do problema envolvido, tal característica também pode ser encontrada em outros algoritmos adaptativos como o LMS. Em nosso modelo para uma previsão adequada do desajuste, pode ser necessário escolher um comprimento de palavra, de forma que o excesso de erro médio quadrático (MSE) seja muito menor que o erro mínimo médio quadrático, conforme tabela (1). Pode-se ainda especificar uma relação entre o comprimento de palavra mínimo para os coeficientes de modo a garantir a convergência com um ruído de quantização desprezível, no caso (12 e 13 bits), para o algoritmo SA.

## 6 CONCLUSÃO

Analisamos o algoritmo SA em precisão finita utilizando aritmética de ponto fixo. Para isto derivamos expressões analíticas para o desajuste em precisão finita e realizamos uma aproximação em série de Taylor, o que tornou possível realizar essa implementação. Os resultados preliminares encontrados sugerem a possível implementação em dispositivos de hardware desse algoritmo. Isso implica em novas alternativas para resolução de problemas de processamento de sinais encontrados em diversas áreas, como por exemplo, na medicina, militar e outros. Os próximos rumos dessa pesquisa serão continuarmos avaliando estes resultados analíticos para outros algoritmos adaptativos em precisão finita, baseados em funções de custo de erro não-médio-quadrático, para diferentes sinais de entrada e ruídos. Como pesquisa futura sugerimos a implementação desse algoritmo em dispositivo de hardware.

## AGRADECIMENTOS

Ao PPGEE-UFMA (Programa de Pós Graduação em Engenharia Elétrica-Universidade Federal do Maranhão), ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), a FAPEMA (Fundação de Amparo à Pesquisa e Desenvolvimento Científico do Maranhão) e ao PIB (Laboratório da Informação Biológica).

## REFERÊNCIAS

- Bershad, N. J., & Bermudez, J., C., M., 1996. A Nonlinear analytical model for the quantized LMS algorithm - The power-of-two step size case. *IEEE Transactions on Sig. Proc.*, vol. SP-44, no. 11, pp. 2895-2900.
- Diniz, P., 2013. *Adaptive filtering*, Rio de Janeiro.
- Douglas, S., C., & Meng, T., H., 1994. Stochastic gradient adaptation under general error criteria. *IEEE Transactions on Sig. Proc.*, Vol. 42, no. 6.
- Gupta, R., & Hero, A., O., 2000. Transient Behavior of fixed point LMS adaptation,. *University of Michigan, Ann Arbor*, MI 48109-2122.
- Haykin, S., 2014. *Adaptive filter theory and Applications*, USA
- Maluenda, Y., R., M., & et al, 2005. Propriedades do algoritmo LMS operando em precisão finita. *XXII Simpósio Brasileiro de Tele.*, SBrT'05.
- Meher, P., K., & Park, S., Y., 2014. Area-delay-power efficient fixed-point LMS adaptive filter with low adaptation-delay. *IEEE Transactions on Very large scale int.(VLSI) Systems*, Vol. 22, no. 2.
- Oppenheim, A. V., & et al, 1996. *Signals and Systems*, Prentice Hall, New Jersey.
- Sayed, A. H., 2013. *Fundamentals of Adaptive Filtering*, IEEE Press, University of California.
- Santana, E., 2006. Estudo e desenvolvimento de uma família de algoritmos não lineares para filtragem adaptativa. In Olhoff, N. & Rozvany, G. I. N., eds, *UFMA*.
- Widrow, B., & Stearns, S., D., 1985. *Adaptive signal processing*, Prentice Hall, Englewood Cliffs. *Prentice Hall, Englewood Cliffs*.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### CONTACT ASSISTED PROTEIN STRUCTURE PREDICTION USING A GENETIC ALGORITHM WITH PHENOTYPIC CROWDING

**Karina B. Santos**

**Gregório K. Rocha**

**Fábio L. Custódio**

**Laurent E. Dardenne**

karinabs@lncc.br

gregorio@lncc.br

flc@lncc.br

dardenne@lncc.br

National Laboratory for Scientific Computing (LNCC)

Av. Getúlio Vargas, 333 - Quitandinha, 25651-075, RJ, Petrópolis, Brazil

**Abstract.** *The use of protein contact maps in protein structure predictions methodologies has proved promising and was highlighted in the last CASP editions (CASP10 and CASP11). The goal of this work is the investigation and the development of strategies to use the contact map's information to improve the quality of the protein models predicted by the program GAPF (Genetic Algorithm for Protein Folding). GAPF uses a steady-state genetic algorithm and a parental replacement by phenotypic crowding. This strategy provides a better exploration of the energy hypersurface allowing the identification of multiple minima and preserving the population diversity of the generated structures. Concerning the GAPF predictive ability, it's algorithm needs some improvement in prediction of proteins that have complex structures, with several  $\beta$ -strands or with more than 100 residues in the sequence. To overcome these difficulties, we incorporated the protein contact map information into the GAPF program by developing and implementing a new term in the fitness function, associated with the contacts restrictions. The methodology was evaluated using a test set of six proteins belonging to different classes (i.e., preferably  $\alpha$ ,  $\alpha/\beta$  and preferably  $\beta$ ). The results show that using contact map knowledge associated with a new term in GAPF's fitness function increased the capacity of the algorithm to obtain better protein models. For almost all target proteins the best-predicted models had their RMSD below 4.30 Å and GDT-TS above 57%.*

**Keywords:** Protein structure prediction, protein contact map, genetic algorithm

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUCTION

The protein structure prediction (PSP) is still being one of the most important purposes of computational molecular biology research area, and its goal is to discover the native three-dimensional (3D) protein structure using the information stored in the amino acid sequence (Michel et al., 2014). Since 1994, with the CASP event (*The Critical Assessment of Techniques for Protein Structure Prediction*), it has been possible to note a significant advance in computational methods to predict protein structures (Dill and MacCallum, 2012). In the last two CASP editions (CASP10 and CASP11), the use of the knowledge about predicted sets of residue pairs that are closer to each other in the native protein structure, *i.e.* that are in “contact”, combined with *de novo* PSP strategies showed promising results (Taylor et al., 2014; Koscioletk and Jones, 2014; Kinch et al., 2016a). The information about which residues should be in contact, represented as a contact map, is commonly used in the form of a distance restraint term in the fitness function (Joo et al., 2016; Kim et al., 2014; Koscioletk and Jones, 2015; Wang et al., 2016). During CASP10 experiment, models predicted using contact maps information (contact-assisted predictions) had GDT-TS (*Global Distance Test Total Score*) scores 40 points higher than the best-predicted models, for the same proteins, without contact information (contact-unassisted prediction) (Taylor et al., 2014). The contact-assisted structure modeling experiments in CASP11 significantly improved structure predictions, with a correct *de novo* prediction of proteins with much larger domains than it was possible to be modeled on previous CASP editions (Kinch et al., 2016b).

In a contact map, two residues are in “contact” if the Euclidean distance between its C- $\beta$  is less than or equal to a predefined distance threshold, usually, 8 Å (Wang and Xu, 2013). Residue-residue contacts can be predicted from analysis of correlated evolution mutations obtained from multiple sequence alignments of a sufficiently large number of homologous proteins (Michel et al., 2014).

This work proposes the development of a strategy to use the information from protein contact maps to improve the predictive ability of PSP methodologies. We also focuses on the GAPF PSP program, a phenotypic crowding-based steady-state genetic algorithm, *i.e.*, an algorithm that work with a population of candidate solutions whose parental replacement method forces competition between the most similar individuals. GAPF also uses protein fragment libraries and secondary structure predictions (obtained from a PSIPRED horizontal format file, Jones (1999)) to guide the protein structure prediction (Custódio et al., 2010, 2014; Rocha et al., 2015; Santos et al., 2015).

## 2 METHODS

### 2.1 The Fitness Function

A coarse-grained representation is used in GAPF, where the side chain atoms are replaced by a super-atom located at its geometric center and the fitness function is based on the energy from the interaction between the atoms of the protein. This energy is calculated using a molecular force field made up of a dihedral potential (*Dihed*) from GROMOS96 force field (van Gunsteren et al., 1996), four hydrogen bonds terms, which one is fully independent of secondary structure prediction ( $HB_{dist-NHO}$ ) and three dependent on secondary structure prediction ( $HB_{hx}$  - for  $\alpha$ -helix;  $HB_{st}$  - for  $\beta$ -sheet;  $HB_{att-st}$  - for attracting the backbone dipoles of  $\beta$ -strands),

an atomic repulsive term ( $A_{rep}$ ) and a hydrophobic compaction term ( $C_{pk}$ ) (Rocha, 2015). The proposed strategy consists of (I) incorporate into the GAPF fitness function a residue-residue contact term to model the knowledge contained in the protein contact map generated for the protein being folded (II) assess its influence on the final quality of the predicted models.

**Contact Map Term.** Typically, for each amino acid pair described in the contact map, there is a confidence value ( $\gamma$ ) associated with the probability of such residues being in contact. Often this confidence value ranges between 0-1.

The Contact Map Term proposed in this work deals with the residue-residue contacts described in the contact map in the form of distance restraints. Thus, for each protein model generated, the distance of two residues ( $(\sigma(a_i, a_j))$ ) is calculated on C $\beta$  atoms (C $\alpha$  for Glycine). According to the distance value, a contributing value ( $\lambda_{(ai,aj)}$ ) is computed (Eq. (1)). Equation 1 is applied for every residue-residue pair found in the contact map, and each contribution value is assigned to the contact term score ( $E_{rr}$ ) (Eq. (2)).

$$\lambda_{(ai,aj)} = \begin{cases} \gamma * 1000 & if 2.0 \leq \sigma(a_i, a_j) \leq 8 \\ \gamma/2 * 500 & if 8 < \sigma(a_i, a_j) \leq 10 \end{cases} \quad (1)$$

$$E_{rr} = \sum_{rr-pairs} \lambda_{(ai,aj)} \quad (2)$$

$$E_{total} = Dihed + HB_{dist-NHO} + HB_{hx} + HB_{st} + HB_{att-st} + A_{rep} + C_{pk} - E_{rr} \quad (3)$$

Thus, in the Contact Map Term, the more residues are respecting the distance constraints described in the contact map, the greater their contribution to the Fitness Function.

Equation 3 shows the complete fitness function used in GAPF program.

## 2.2 “Experimental” Contact Map

To validate Contact Map Term embedded in the GAPF fitness function, for each target protein, a “experimental” contact map was generated using the PDB file from the respective experimental protein structure. In this case, for each residue pair with at least three residue separation along the protein sequence, the Euclidean distance between its C- $\beta$  (C $\alpha$  for Glycine) was calculated. So, the “experimental” contact map consists of all residue pairs that have distance value less than or equal to 8 Å. The threshold value assigned to each residue pair in the “experimental” contact map is 1.0. Table 1 shows the total number of contact pairs described in the “experimental” contact map for each target protein.

## 2.3 Test Set

To assess the effects of Contact Map Term in the quality of predicted structures, a set of six proteins with 3D experimentally determined structure, with 56-108 amino acid residues in the sequence, was used (Table 1).

**Table 1: Test set and contact map**

ID (PDB/CASP)	Class	Sequence Length*	“Experimental” contact pairs
3FIL	$\alpha/\beta$	56	128
2N2U (CASP ID - T0773-D1)	$\alpha/\beta$	67	189
____** (CASP ID - T0820-D1)	$\alpha$	90	186
1FNA	$\beta$	91	212
2MQ8 (CASP ID - T0769-D1)	$\alpha/\beta$	97	295
4Q53 (CASP ID - T0766-D1)	$\alpha/\beta$	108	317

\*Number of amino acids in sequence.

\*\*Protein without identifier in the Protein Data Bank. Experimentally determined structure obtained directly from CASP11 server (<http://www.predictioncenter.org/casp11/>).

## 2.4 Parameters and Tests Configurations

The parameters for the GAPF program during the tests were: population size of 200 individuals, a maximum number of 2,000,000 fitness function evaluations and 20 independent runs per sequence.

At the end of each run, the quality of the 200 generated structures was evaluated after comparing it to the reference structure using the RMSD (*Root-mean-square deviation of atomic positions*) and the GDT-TS.

The quality of the models predicted using the contact map strategy (Contact Map Protocol) was compared with those predicted by the GAPF running with the original protocol, i.e., without the contact map term.

## 3 RESULTS AND DISCUSSION

### 3.1 Impact of the use of Contact Maps

The use of distance constraints imposed by contact maps allowed the algorithm to generate models with structural conformations closer to that found in the native protein. That's because satisfying these restrictions, increases the probability of correctly reproduce the 3D arrangement of the predicting protein.

Table 2 shows the RMSD and GDT-TS values of the best models and of the best energy models predicted in 20 independent runs of GAPF program. According to Kryshtafovych et al. (2005), are considered good predictive ability RMSD values bellow 4.0 Å. Martí-Renom et al. (2000) believes that RMSD values between 3.0 and 4.0 Å means that the model can correctly

represent the protein folding. RMSD values above 5.0 indicate little information about the three-dimensional protein structure (Unger, 2004). A GDT-TS value equal to 100% means that the distance between the structural alignments is no more than 1.0 Å. GDT-TS values above 50% indicate good predictive ability, values below 50% infer that proteins do not have the same folding, and scores between 10-20% represent only random overlaps between the structures (Xu and Zhang, 2010).

**Table 2: Comparison between the original and the contact map protocols using the best RMSD and GDT-TS values obtained in 20 independent runs of the GAPF program.**

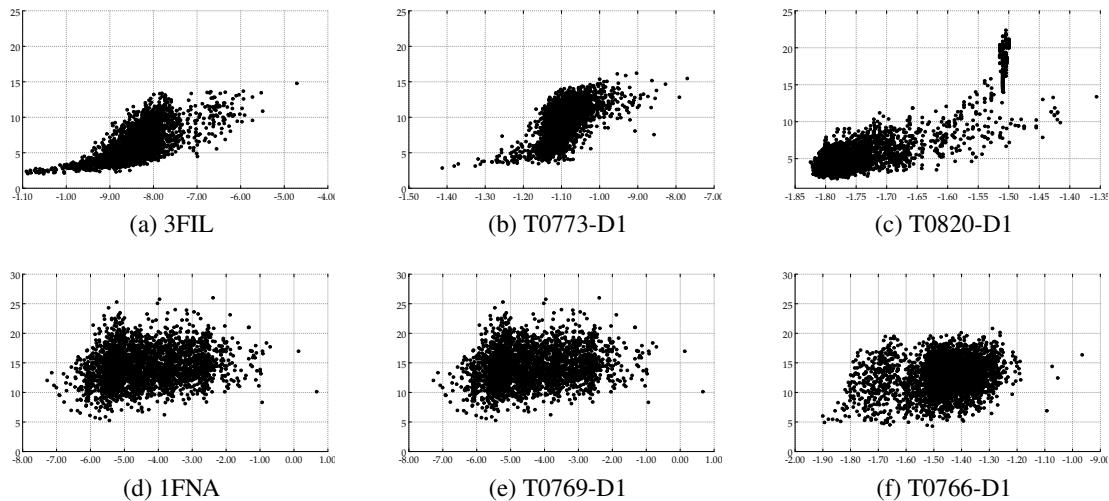
ID (PDB/CASP)	Original Protocol				Contact Map Protocol			
	RMSD (Å)		GDT-TS (%)		RMSD (Å)		GDT-TS (%)	
	Best model	Best energy	Best model	Best energy	Best model	Best energy	Best model	Best energy
3FIL	5.33	11.59	51.82	37.73	2.03	2.38	58.64	51.82
T0773-D1	7.40	11.85	41.79	24.25	2.84	2.84	69.03	69.03
T0820-D1	9.60	13.78	40.28	26.94	2.28	4.53	75.56	57.22
1FNA	8.77	15.52	21.43	12.09	5.26	12.02	17.86	13.74
T0769-D1	7.78	14.61	42.01	20.88	3.91	7.15	60.82	55.41
T0766-D1	10.63	17.45	31.25	21.99	4.29	6.00	57.18	52.55

The relationship between the energy and RMSD values of the models generated using the Contact Map Protocol is summarized in Figure 1. Our data shows that, for most targets, the use of contact term in the fitness function provides best structural quality models (best RMSD value) among those of lower energy (best energy). One common behavior observed from all sequences is that models with same energy values can present very distinct RMSD values, this illustrates the multiple minima characteristic of the GAPF algorithm.

Table 3 indicates the average of the best models in terms of RMSD and GDT-TS values and in terms of energy in each run. In Tables 2 and 3, it can be observed that when executed using the original protocol, the quality of solutions decreases for longer sequences, e.g., proteins with more than 90 residues in the sequence. Furthermore, the use of the “experimental” contact maps improved the results generating models for the same sequences with RMSD below 4.30 Å and GDT-TS value above 57% (Table 2).

Figure 2 shows the best models generated for each target using the original protocol and the contact map protocol. The compelling advantage of the new approach is the improvement of the predictive ability for all secondary structure type, with GAPF being able to generate  $\alpha$ -helix and  $\beta$ -strands in the correct place. Even for 1FNA protein, that had RMSD and GDT-TS values representative of random overlaps between structures in both experiments (Original and Contact Term protocols), its 3D representation shows the formation of the  $\beta$ -barrel (Fig. 3).

For T0769-D1 protein, the shortened  $\beta$ -sheet can be explained by the secondary structure predicted by the PSIPRED program (Jones, 1999) that shows some differences between the



**Figure 1:** Results for 20 independent runs for each sequence. Energies in the  $x$ -axis are in Kcal/mol and are divided by  $10^4$  (lower is better). Structural variations (measured with the RMSD) of the predicted models relative to the native reference structure for each protein in the  $y$ -axis are in Å.

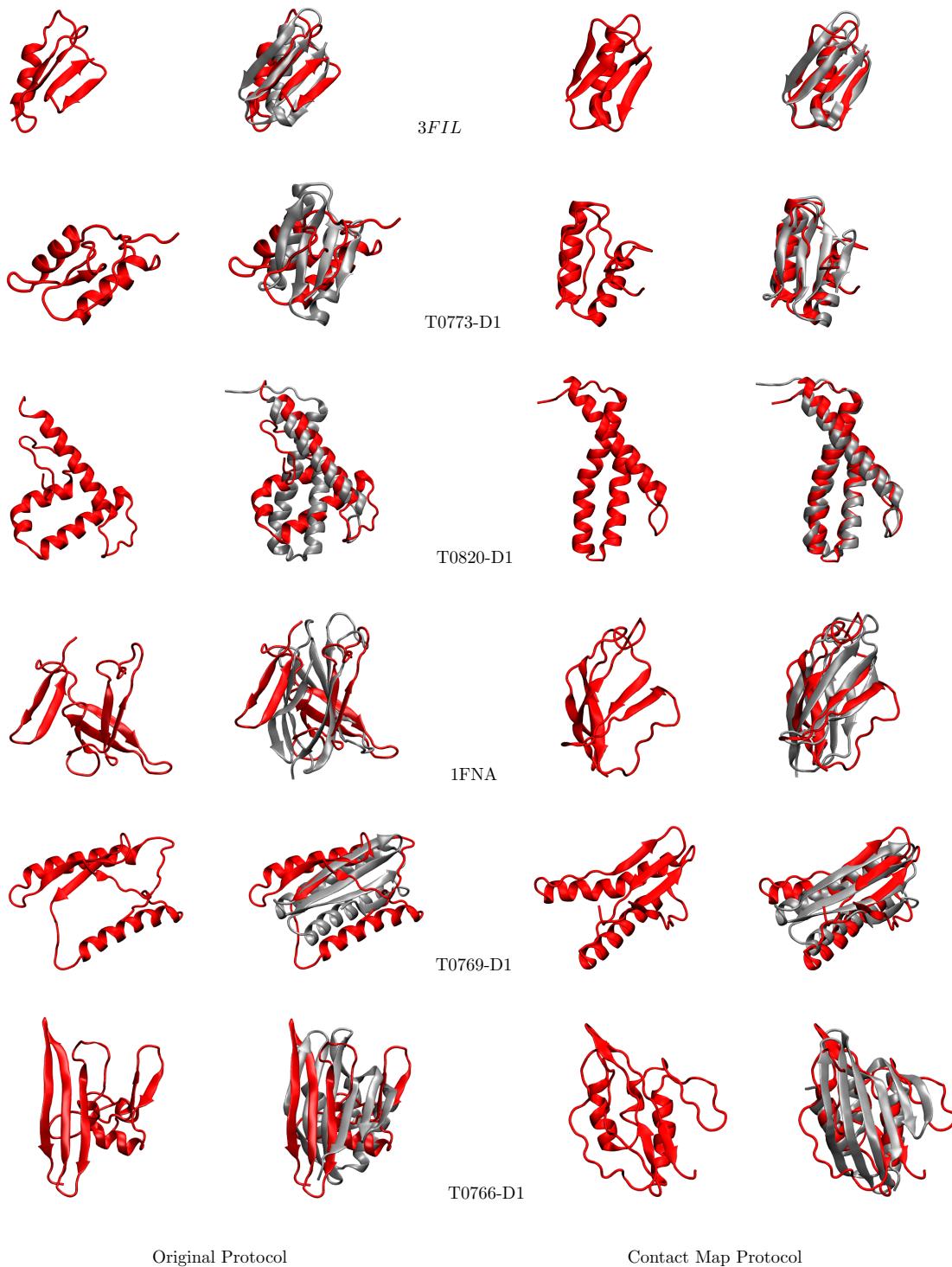
**Table 3:** Comparison between the original and the contact map protocols using the average of the best RMSD and GDT-TS values obtained in 20 independent runs.

ID (PDB/CASP)	Original Protocol				Contact Map Protocol			
	RMSD (Å)		GDT-TS (%)		RMSD (Å)		GDT-TS (%)	
	Best model	Best energy	Best model	Best energy	Best model	Best energy	Best model	Best energy
3FIL	6.41	9.55	43.59	36.61	2.74	3.46	55.97	51.61
T0773-D1	8.68	11.45	38.17	30.91	3.57	4.02	63.26	58.99
T0820-D1	11.44	14.91	34.70	26.38	2.69	3.96	70.50	62.59
1FNA	11.41	14.63	16.77	13.18	7.18	10.46	16.34	13.40
T0769-D1	10.79	14.74	32.40	25.54	4.90	5.60	56.03	52.21
T0766-D1	11.86	16.37	24.52	20.37	5.75	8.43	48.85	41.75

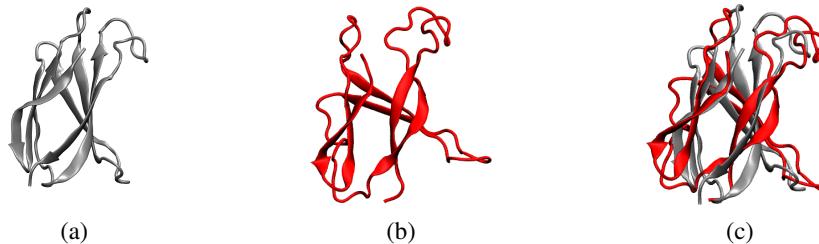
secondary structure assigned by DSSP (Kabsch and Sander, 1983) (Fig. 4), and also shows low confidence values at the end of each predicted secondary structure type which could have hindered a better fragment insertion.

## 4 CONCLUSION

The use of contact maps in the form of distance constraints is a promising strategy for *de novo* PSP methodologies. Using “experimental” contact maps composed by real contacts found in the native structure permitted to assess the potential of this technique, isolating the problem of the correctness of the contacts assignment.



**Figure 2: Comparison between the best models generated with GAPF program (red structures). On the left are the models generated using the original GAPF protocol, on the right created using Contact Map Term. Each model was aligned with the experimentally determined structure that was used as reference for calculating the RMSD and GDT-TS (silver structures).**



**Figure 3:** 1FNA protein seen for another angle that emphasizing the formation of  $\beta$ -barrel.(a) Experimentally determined structure, (b) Model generated using Contact Map Protocol and (c) Alignment between experimental and predicted structure.

**Figure 4: T0769-D1 - Psipred Secondary structure prediction compared with DSSP determined secondary structure. The letters ‘C’, ‘H’ and ‘E’ means Coil, helix- $\alpha$  and  $\beta$ -sheet respectively. In red are highlighted the disagreement secondary structures. ‘Conf’ term refers to confidence in this secondary structure prediction by PSIPRED.**

However, when the protein 3D structure is unknown, it is necessary to use other techniques to predicted residue-residue contacts. Protein contact maps can be predicted using coevolution-based methods. As an example of programs to generate contact maps are METAPSICOV (Jones et al., 2015) and GREMLIN (Ovchinnikov et al., 2014). METAPSICOV and GREMLIN contact maps, usually show the probability of all residues being in contact with all residues. In this case, these maps can contain various incorrect contact predictions. The lack of full confidence in the prediction of such contacts, points as very necessary to develop some strategies to filter and enhance the information of predicted contacts. In this way, the next step towards the improvement of the use of a Contact Map Term should be the development and introduction of a contact map filter that identifies which predicted contacts, from METAPSICOV, GREMLIN or other contact map predictor, are really valuable or not for PSP.

## ACKNOWLEDGMENT

The authors would like to thank the support from CAPES and FAPERJ (grant n°. E26/010.001229/2015).

## REFERENCES

- Custódio, F. L., Barbosa, H. J., and Dardenne, L. E. Full-atom ab initio protein structure prediction with a genetic algorithm using a similarity-based surrogate model. *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

Custódio, F. L., Barbosa, H. J., and Dardenne, L. E. A multiple minima genetic algorithm for protein structure prediction. *Applied Soft Computing*, 15:88–99, 2014.

- Dill, K. A. and MacCallum, J. L. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.
- Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.
- Jones, D. T., Singh, T., Kosciolék, T., and Tetchner, S. Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, 31(7):999–1006, 2015.
- Joo, K., Joung, I., Cheng, Q., Lee, S. J., and Lee, J. Contact-assisted protein structure modeling by global optimization in casp11. *Proteins: Structure, Function, and Bioinformatics*, 2016.
- Kabsch, W. and Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.
- Kim, D. E., DiMaio, F., Yu-Ruei Wang, R., Song, Y., and Baker, D. One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):208–218, 2014.
- Kinch, L. N., Li, W., Monastyrskyy, B., Kryshtafovych, A., and Grishin, N. V. Assessment of casp11 contact-assisted predictions. *Proteins: Structure, Function, and Bioinformatics*, 2016a.
- Kinch, L. N., Li, W., Monastyrskyy, B., Kryshtafovych, A., and Grishin, N. V. Evaluation of free modeling targets in casp11 and roll. *Proteins: Structure, Function, and Bioinformatics*, 2016b.
- Kosciolék, T. and Jones, D. T. De novo structure prediction of globular proteins aided by sequence variation-derived contacts. *PloS one*, 9(3):e92197, 2014.
- Kosciolék, T. and Jones, D. T. Accurate contact predictions using covariation techniques and machine learning. *Proteins: Structure, Function, and Bioinformatics*, 2015.
- Kryshtafovych, A., Milostan, M., Szajkowski, L., Daniluk, P., and Fidelis, K. Casp6 data processing and automatic evaluation at the protein structure prediction center. *Proteins: Structure, Function, and Bioinformatics*, 61(S7):19–23, 2005.
- Martí-Renom, M. A., Stuart, A. C., Fiser, A., Sánchez, R., Melo, F., and Šali, A. Comparative protein structure modeling of genes and genomes. *Annual review of biophysics and biomolecular structure*, 29(1):291–325, 2000.
- Michel, M., Hayat, S., Skwark, M. J., Sander, C., Marks, D. S., and Elofsson, A. Pconsfold: improved contact predictions improve protein models. *Bioinformatics*, 30(17):i482–i488, 2014.
- Ovchinnikov, S., Kamisetty, H., and Baker, D. Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *Elife*, 3:e02030, 2014.
- Rocha, G. K., Custódio, F. L., Barbosa, H. J. C., and Dardenne, L. E. A multiobjective approach for protein structure prediction using a steady-state genetic algorithm with phenotypic crowding. *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on*, pages 1–8, 2015.

- Rocha, G. K. *Desenvolvimento de Metodologias Para Predição de Estruturas de Proteínas Independente de Moldes*. Tese de Doutorado, PhD thesis, Laboratório Nacional de Computação Científica (LNCC), Petrópolis, RJ., 2015.
- Santos, K. B., Custódio, F. L., Barbosa, H. J., and Dardenne, L. E. Genetic operators based on backbone constraint angles for protein structure prediction. *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on*, pages 1–8, 2015.
- Taylor, T. J., Bai, H., Tai, C.-H., and Lee, B. Assessment of casp10 contact-assisted predictions. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):84–97, 2014.
- Unger, R. The building block approach to protein structure prediction. pages 177–188. Springer, 2004.
- van Gunsteren, W. F., Billeter, S., Eising, A., Hünenberger, P. H., Krüger, P., Mark, A. E., Scott, W., and Tironi, I. G. Biomolecular simulation: the {GROMOS96} manual and user guide. 1996.
- Wang, S., Li, W., Zhang, R., Liu, S., and Xu, J. Coinfold: a web server for protein contact prediction and contact-assisted protein folding. *Nucleic acids research*, pages gkw307, 2016.
- Wang, Z. and Xu, J. Predicting protein contact map using evolutionary and physical constraints by integer programming. *Bioinformatics*, 29(13):i266–i273, 2013.
- Xu, J. and Zhang, Y. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### BIOLOGICAL CONTROL OF A PEST IN A REACTION-DIFFUSION PREDATOR-PREY SYSTEM WITH ALLEE EFFECT

**Lucas dos Anjos**

lanjos@lncc.br

**Regina Célia Cerqueira de Almeida**

rcca@lncc.br

**Michel Iskin da Silveira Costa**

michel@lncc.br

Laboratório Nacional de Computação Científica

Av. Getúlio Vargas, 333, Quitandinha Petrópolis - Rio de Janeiro CEP 25651-075 - Brasil

**Abstract.** A common strategy in pest biological control is the release of a natural enemy of the pest. Such introduced species is sometimes named control agent. One of the explanations for biological control failure accounts for the presence of Allee effects in the control agent. An Allee effect can be defined as the positive density dependence between one (or more) fitness item of an individual and the density of its own population. The aim of biological control is to eradicate the pest or at least keep it below some economic threshold that does not affect significantly the rent of the involved activity (pest suppression). In the present work we assess how the intensity of the release of the pest enemy can be detrimental or advantageous to pest management goals. To this end we resort to numerical simulations of a reaction-diffusion prey-predator model in a one-dimensional environment where the native and exotic species interact.

**Keywords:** Allee effect, reaction-diffusion, invasive species, biological control, predator-prey system.

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUCTION

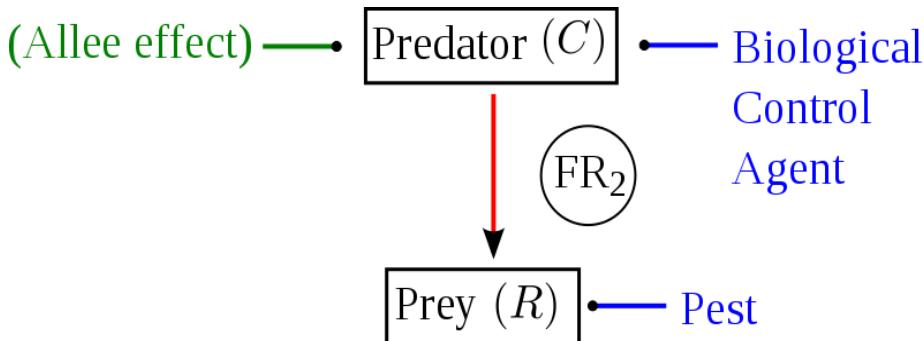
Invasive species increasingly threaten ecosystems worldwide. A commonly used biocontrol strategy to counteract these invasions consists of the deliberate introduction of natural enemies of an invasive pest species (Blackwood et al., 2012; Suckling et al., 2012). A particular case is the introduction of a biological control agent when a pest invades a plant cultivation in a greenhouse (Van Lenteren, 2000).

In this work, in order to assess the effectiveness of this biocontrol strategy, we develop a predator-prey dynamical model where the prey species is the pest (invasive species) and the consumer is the deliberately introduced biological control agent. That is to say, in our proposed model, both pest and control agent are supposed to be exotic species. Based on the empirical evidences (Bompard et al., 2013), it is supposed that the consumer (the biological control agent) has an Allee effect in its numerical response (Zhou et al., 2005; Verdy 2010). We first analyze a non-spatial predator-prey model based on Verdy (2010) composed of a system of two coupled nonlinear differential equations. Secondly, we analyze the corresponding spatial model adding diffusion processes to both prey and predator, converting thus the model to a system of coupled nonlinear partial differential equations. By means of numerical simulations it is shown that the amount of released enemies is crucial to the success or failure of the biocontrol strategy in both modeling contexts.

The results found in this work should not be seen as general results. Instead, they are related to a specific set of parameter values. The analyzed population models are of strategic type (May, 2001) and therefore they do not usually describe the dynamics of a specific real biological system. In this view, the analysis is based on a hypothetical set of parameters. Since the studied models have a large number of parameters, the intention is to show some possible outcomes instead of an exhaustive study for all possible outcomes (Abrams and Roth, 2004).

## 2 METHODS

The population framework to be analyzed is schematically displayed in figure 1 (hereinafter  $FR_2$  denotes functional response type 2).



**Fig. 1:** An introduced control agent ( $C$ ) preys upon the pest ( $R$ ) with functional response type 2. Allee effect acts on the agent  $C$ . Arrow denotes consumption.

The trophic scheme of figure 1 consists basically of a ditrophic food chain: prey (pest,  $R$ ); predator (an introduced biological control agent,  $C$ ). The dynamics of figure 1 in an homogeneous environment can be modeled by (Verdy, 2010) :

$$\frac{dR}{dt} = rR \left(1 - \frac{R}{K}\right) - \frac{a_{CR}R}{1 + a_{CR}T_{hCR}R}C; \quad (1)$$

$$\frac{dC}{dt} = ef_{RC} \left(\frac{C}{\theta_C + C}\right) \left(\frac{a_{CR}R}{1 + a_{CR}T_{hCR}R}C\right) - m_C C,$$

where  $R$  is the density of pest individuals and  $C$  is the density of biological control agent individuals;  $r$  is the maximum per capita rate of pest growth and  $K$  is its carrying capacity;  $a_{CR}$  represents the attack coefficient of the introduced biological control agent  $C$  upon the pest  $R$ ;  $T_{hCR}$  is the manipulation time of  $R$  by  $C$ , while  $ef_{RC}$  is the biological control agent's food-to-offspring ( $R$  to  $C$ ) conversion efficiency coefficient. Finally,  $\left(\frac{C}{\theta_C + C}\right)$  describes the Allee effect on  $C$ , where  $\theta_C$  denotes its intensity, and  $m_C$  is the density independent per capita mortality rate of  $C$ .

A spatio-temporal counterpart to model (1) can be given by the following model:

$$\begin{aligned} \frac{\partial R}{\partial t} &= D_R \frac{\partial^2 R}{\partial x^2} + rR \left(1 - \frac{R}{K}\right) - \frac{a_{CR}R}{1 + a_{CR}T_{hCR}R}C; \\ \frac{\partial C}{\partial t} &= D_C \frac{\partial^2 C}{\partial x^2} + ef_{RC} \left(\frac{C}{\theta_C + C}\right) \left(\frac{a_{CR}R}{1 + a_{CR}T_{hCR}R}C\right) - m_C C, \end{aligned} \quad (2)$$

where  $D_R$  and  $D_C$  are the diffusivity coefficients of prey and predator, respectively. The spatial domain  $x$  has the length from 0 to  $L = 10$ .

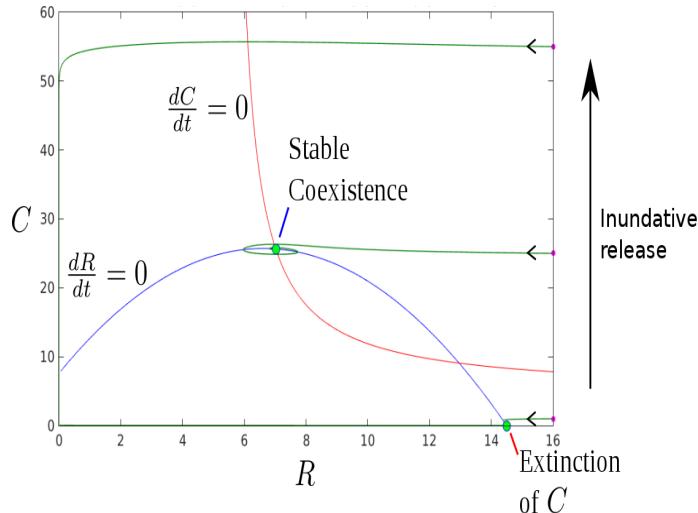
We assume that the initial spatial distributions of pest and agent are uniform:  $R(x, 0) = \alpha > 0$ ;  $C(x, 0) = \beta > 0$ , except on the boundaries. This assumption may describe the uniform spatial distribution of pest among plants and a concomitant uniform spatial spreading of the agent as a measure of pest biological control. Moreover, we apply homogeneous Dirichlet boundary conditions, that is,  $R(0, t) = R(L, t) = C(0, t) = C(L, t) = 0$ . These boundary conditions can well represent plant cultivation in a greenhouse where pest and agent individuals cannot survive on its borders.

Model (1) is solved using a 4th Runge-Kutta method. In the following experiments we set  $h = 0.05$  and  $\Delta t = 0.05$ , which yield convergent discrete approximations. Remark that the numerical scheme is stable independently of the parameters which are small enough to guarantee the solution accuracy. Model (2) is solved numerically by using the finite differences method. To this end both the spatial and time domains are uniformly discretized into  $N$  and  $M$  parts so that  $h = \frac{L}{N}$  and  $\Delta t = \frac{T}{M}$ , where  $T$  is the total simulation time. Backward Euler method is used in time and the second order operator is approximated by central differences. The resulting system of algebraic equations is linearized using the Picard method and solved by uncoupling the equations with a Gauss-Seidel structure. Finally, the Gaussian elimination is used to the two linear systems.

Our intent is to assess the effects of the predator ( $C$ ) initial condition on model (1) and the predator initial distribution on model (2) regarding the coexistence and/or extinction of the involved species for both models.

### 3 RESULTS

First, we choose a specific set of parameter values in model (1) such that species stable coexistence occurs as shown in figure 2.

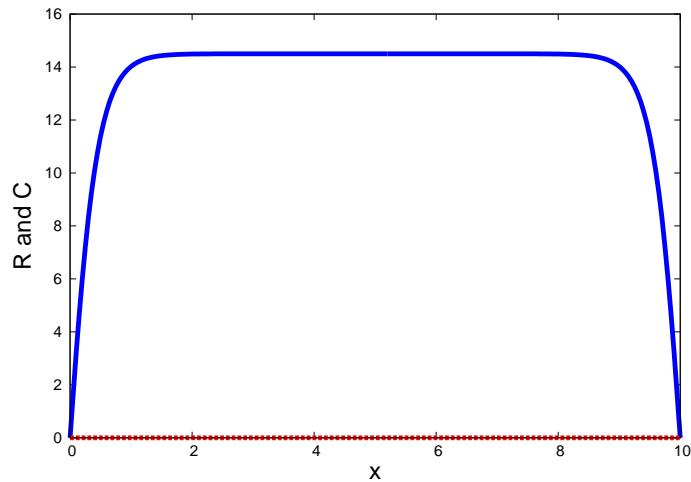
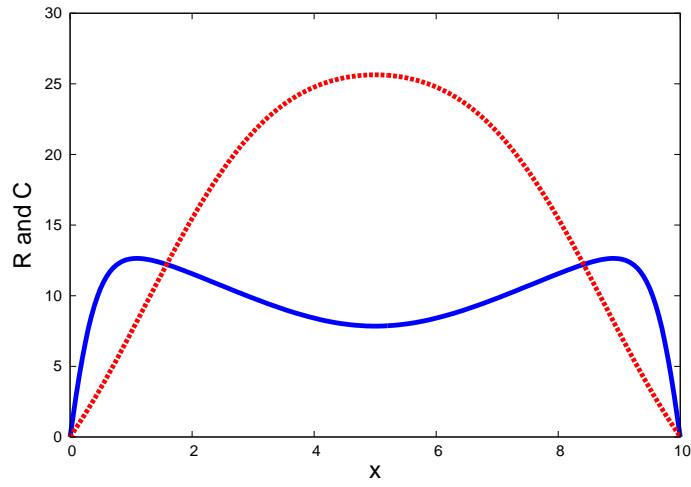
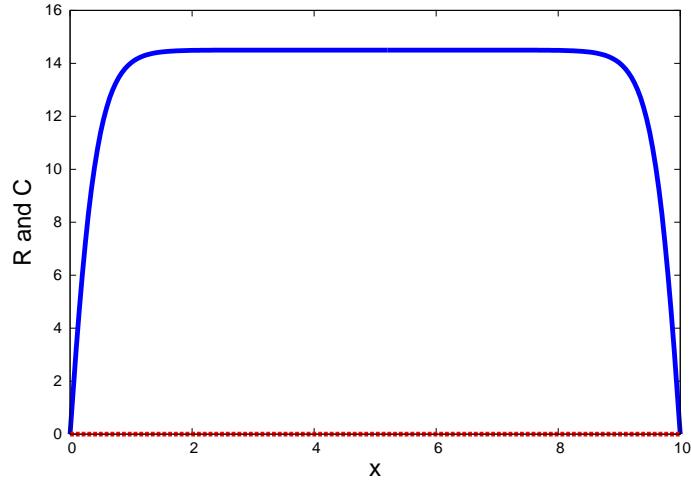


**Fig. 2: One phase plane of model (1).** Parameter values:  $r = 6$ ;  $K = 14.5$ ;  $a_{CR} = 0.7903$ ;  $T_{hCR} = 1$ ;  $ef_{RC} = 0.8$ ;  $\theta_C = 1.1$ ;  $m_C = 0.65$ . '●' - initial conditions: (16, 1), (16, 25) and (16, 55); '●' - final value for  $t = 500$ .

Figure 2 shows that the dependence on the initial population of agent plays a crucial role on the ultimate dynamics of model (1). Actually, this phase plane represents an inundative release of pest natural enemy (i.e., increasing agent initial population while maintaining pest population constant). Note that agent extinction occurs for inoculative as well as inundative releases. However, for intermediary values of enemy release pest and agent, stable coexistence can occur, which may portray a possible success of pest control depending on whether the stable pest level is below economic thresholds (Boukal and Berec, 2009).

These results are partially in contrast with the dynamical results pertaining to model (1) without the Allee effect in the agent (i.e.,  $\theta_C = 0$ ). In this case model (1) reduces to a Rosenzweig-McArthur model (Turchin, 2003) where according to parameter values the dynamics consist of either a globally stable limit cycle or a globally stable equilibrium point. In the former the introduction of an agent would be advantageous to pest biological control if the magnitude of the sustained oscillation is relatively small; otherwise it might imply pest outbreaks. In the latter it would depend on whether the pest stabilization level is below economic thresholds (Boukal and Berec, 2009). In terms of the numerical solution of model (2), setting  $\theta_C = 0$  reduces the nonlinearity, which eventually improves convergence of the iterative process.

Figure 3 shows a simulation of model (2) with the same parameter values of model (1) with diffusion coefficients of pest and agent respectively given by  $D_R = 0.4$ ;  $D_C = 0.17$ . In order to simulate an inundative release of pest enemy in a spatial context, the initial distributions of pest and agent are uniform and with the same values as the initial conditions of figure 2.

(a) Uniform initial conditions:  $R = 16$  and  $C = 1$ .(b) Uniform initial conditions:  $R = 16$  and  $C = 25$ .(c) Uniform initial conditions:  $R = 16$  and  $C = 55$ .

**Fig. 3:** Spatial distribution of pest and agent along space  $x$ ,  $L = 10$ , at  $t = 100$  in model (2). Diffusion coefficients of pest and agent are respectively: ( $D_R = 0.4$ ;  $D_C = 0.17$ ); initial distributions: (a) (16, 1); (b) (16, 25); (c) (16, 55). Solid blue line: pest  $R$ ; dashed red line: agent  $C$ . Parameter values are the same as those of figure 2.

Similarly as in figure 2, figure 3 displays three behaviors as  $C(x, 0)$  increases:  $C$  extinction, stable coexistence of  $C$  and  $R$ , and  $C$  extinction again. In terms of pest control this sequence of results points to the fact that failure of pest biocontrol can be interspersed with pest and agent coexistence (which can portray a possible success of pest control depending on the stable pest spatial distribution) as more agents are introduced in the system. This non-monotonic behavior is at variance with the view that agent inundative releases are proper to pest eradication or control. We also performed simulations of models (1) and (2) under the same structure of augmenting the pest enemy releases for  $\theta_C = 1$  (in which case sustained oscillations are generated in model (1)). The results from pest biocontrol point of view were the same as the ones presented in this section. It is important to note, however, that the degree of generality of the above results awaits further analyses, such as the influence on the dynamics of model (2) generated by the relation between the magnitude of the diffusion coefficients  $D_R$  and  $D_C$ , and non-uniform initial conditions, to name a few. In this way in table 1 we present the results for some other values of  $D_R$  and  $D_C$ .

**Table 1: Possible outcomes of model (2) for some values of  $D_R$  and  $D_C$ ; a: intermediary initial conditions generate coexistence, b: agent extinction for all simulated initial conditions. Parameter values are the same as those of figure 2.**

$D_R$	$D_C$	Outcome
0.17	0.1	a
0.17	0.17	a
0.17	0.4	b
0.4	0.17	a
0.4	0.4	b
0.4	0.74	b
0.74	0.17	a
0.74	0.74	b
0.74	0.9	b

## 4 DISCUSSION

In order to investigate the influence of the intensity of pest enemy releases on pest dynamics, a non-spatial and spatial predator-prey model were devised where the basal species portrayed the pest and the introduced biological control agent was cast as the predator. The main finding of this work is that the intensity of pest enemy releases is relevant to the success

or failure of the proposed biocontrol strategy. However, contrary to the common belief in pest control procedures, an excessive amount of pest enemy individuals (i.e., inundative releases) may be detrimental to the effectiveness of biocontrol (Costa and dos Anjos, 2015). These results are valid for both spatial and non-spatial analyzed predator-prey models for the considered set of parameters.

---

## ACKNOWLEDGEMENTS

The authors acknowledge the helpful comments and suggestions made by the referees on an earlier version of this work.

## REFERENCES

- Abrams, P.A. & Roth, J., 1994. The responses of unstable food chains to enrichment. *Evolutionary Ecology*, vol. 8, n. 2, pp. 150-171.
- Blackwood, J.C., Berec, L., Yamanaka, T., Epanchin-Niell, R.S., Hastings, A. & Liebold, A.M., 2012. Bioeconomic synergy between tactics for insect eradication in the presence of Allee effects. *Proceedings of the Royal Society B: Biological Sciences*, vol. 279, n. 1739, pp. 2807-2815.
- Bompard, A., Amat, I., Fauvergue, X. & Spataro, T., 2013. Host-parasitoid dynamics and the success of biological control when parasitoids are prone to Allee effects. *PLoS One*, vol. 8, n. 10, pp. e76768.
- Boukal, D.S. & Berec, L., 2009. Modelling mate-finding Allee effects and populations dynamics, with applications in pest control. *Population Ecology*, vol. 51, n. 3, pp. 445-458.
- Costa, M.I.S. & dos Anjos, L., 2015. Integrated pest management in a predator-prey system with Allee effects. *Neotropical Entomology*, vol. 44, n. 4, pp. 385-391.
- Garvie, M.R., 2007. Finite-difference schemes for reaction–diffusion equations modeling predator-prey interactions in MATLAB. *Bulletin of Mathematical Biology*, vol. 69, n. 3, pp. 931-956.
- May, R.M., 2001. *Stability and complexity in model ecosystems*. Princeton University Press.
- Suckling, D.M., Tobin, P.C., McCullough, D.G. & Herms, D.A., 2012. Combining tactics to exploit Allee effects for eradication of alien insect populations. *Journal of Economic Entomology*, vol. 105, n. 1, pp. 1-13.
- Turchin, P., 2003. Complex population dynamics: a theoretical/empirical synthesis, vol. 35. Princeton University Press.
- Van Lenteren, J.C., 2000. Success in biological control of arthropods by augmentation of natural enemies. In Gurr, G. & Wratten, S., eds., *Biological control: measures of success*, pp. 77-103. Springer Netherlands.
- Verdy, A., 2010. Modulation of predator-prey interactions by the Allee effect. *Ecological Modelling*, vol. 221, n. 8, pp. 1098-1107.
- Zhou, S.R., Liu, Y.F. & Wang, G., 2005. The stability of predator-prey systems subject to the Allee effects. *Theoretical Population Biology*, vol. 67, n. 1, pp. 23-31.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### MÉTODOS DE ELEMENTOS FINITOS MISTOS PARA O PROBLEMA DA ELASTICIDADE LINEAR EM MALHAS DE QUADRILÁTEROS

**Thiago O. Quinelato**

**Abimael F. D. Loula**

thiagoq@lncc.br

aloc@lncc.br

Laboratório Nacional de Computação Científica

Av. Getúlio Vargas, 333, Quitandinha, 25651-075, Rio de Janeiro, Petrópolis, Brasil

**Maicon R. Correa**

maicon@ime.unicamp.br

Universidade Estadual de Campinas

Rua Sérgio Buarque de Holanda, 651, Barão Geraldo, 13083-859, São Paulo, Campinas, Brasil

**Todd Arbogast**

arbogast@ices.utexas.edu

The University of Texas at Austin

1 University Station C0200, 78712, Texas, Austin, Estados Unidos da América

**Resumo.** *O problema da elasticidade descreve as deformações de um corpo e as tensões a que ele está submetido, relacionando-as ao conjunto de forças atuantes sobre ele. Sua aproximação deve preservar a continuidade da tração, implicando na conservação local do momento linear, o que não acontece quando métodos de elementos finitos baseados em deslocamento são usados. Uma forma de contornar essa deficiência é a adoção de formulações mistas, que exigem a escolha de espaços de aproximação compatíveis. Outra característica desejável é a simetria do tensor de tensões, que implica na conservação do momento angular. A condição de simetria é exigida de forma variacional, ideia que tem sido amplamente utilizada em malhas formadas por triângulos, fornecendo aproximações de qualidade aceitável tanto para o deslocamento quanto para a tensão. Neste trabalho, é destacado que os espaços de aproximação comumente adotados não são aplicáveis a malhas formadas por quadriláteros convexos arbitrários, já que nesse caso a solução aproximada para o tensor de tensões (a principal variável de interesse) sofre significativa deterioração. Uma formulação híbrida é adotada, o que reduz o custo computacional e melhora a qualidade da matriz resultante. Por fim, as imprecisões observadas são explicadas e alternativas viáveis às estratégias usuais de aproximação são sugeridas.*

**Palavras-chave:** Elasticidade linear, Método dos elementos finitos, Malhas quadrilaterais

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

## 1 INTRODUÇÃO

O problema da elasticidade linear descreve as deformações de um corpo e as tensões a que ele está submetido, relacionado-as ao conjunto de forças atuantes sobre ele. Há diversos desafios relacionados a este problema, que justificam o desenvolvimento de modelos matemáticos e computacionais para obtenção de aproximações numéricas estáveis e precisas. Em particular, neste trabalho o comportamento de um modelo proposto por (Arnold et al., 2015) para o problema da elasticidade linear em malhas de quadriláteros é comparado com dois outros métodos, que são aqui introduzidos. O objetivo é verificar como a distorção da malha de elementos finitos afeta a qualidade da aproximação obtida com os métodos numéricos estudados.

Este trabalho está inserido no contexto de desenvolvimento de métodos de elementos finitos mistos baseados na formulação de Hellinger-Reissner para o problema da elasticidade linear. Ou seja, métodos em que as variáveis de interesse (tensão e deslocamento) são aproximadas simultaneamente em subespaços de dimensão finita  $\mathcal{S}_h$  e  $\mathcal{U}_h$ . O interesse por esse tipo de método (chamado de método misto) se dá pelo fato de que as aproximações para o campo de tensões são melhores que aquelas obtidas em formulações baseadas apenas em deslocamento. Uma questão importante a ser estudada quando formulações mistas são utilizadas é a estabilidade do método numérico, que é em geral condicionada a uma relação de compatibilidade entre os espaços adotados para os campos de tensão e deslocamento.

A principal dificuldade na construção de aproximações para a formulação de Hellinger-Reissner é a conservação do momento angular. A dificuldade reside em atender a condição de compatibilidade entre os espaços de aproximação preservando a simetria do campo de tensões. Este trabalho segue as ideias de Fraeijs de Veubeke (Arnold et al., 1984, 2006) e utiliza um multiplicador de Lagrange associado à rotação para impor a condição de simetria do tensor de tensões de forma variacional. Formulações mistas híbridas baseadas nessa ideia foram originalmente apresentadas por Amara e Thomas (1979). No problema discreto, a rotação é aproximada por um polinômio independente em cada elemento e seus graus de liberdade podem ser condensados graças ao processo de hibridização. Dessa maneira, a introdução de uma variável para reforçar a condição de simetria das tensões (e, consequentemente, a conservação do momento angular) não causa aumento do sistema global de equações resultante da discretização do problema. Além disso, a utilização de métodos híbridos facilita a construção de bases locais para os espaços de aproximação.

Espaços de dimensão finita baseados no trabalho de Raviart e Thomas (1977) foram apresentados por Arnold et al. (2015). Nesse trabalho os autores observam que essa estratégia pode ser aplicada a malhas de paralelogramos, apresentando taxas ótimas de convergência em  $H(\text{div})$  para o campo de tensões e em  $L^2$  para os campos de deslocamentos e rotações. Entretanto, para malhas com quadriláteros arbitrários há certa dificuldade de convergência para a tensão na seminorma de  $H(\text{div})$ .

A proposta é estudar a convergência de estratégias de aproximação da formulação de Hellinger-Reissner usando alternativas aos espaços adotados em Arnold et al. (2015). A primeira escolha de espaços é baseada nos espaços apresentados por Arnold et al. (2005) para o problema de Darcy e fornece boas aproximações para as variáveis de interesse. Entretanto, o número de graus de liberdade nos problemas locais é significativamente maior quando comparamos com os espaços de Arnold et al. (2015). Uma combinação de espaços baseada no trabalho de Arbogast e Correa (2016) é então apresentada e a convergência desta aproximação

é estudada.

Este trabalho está apresentado da seguinte forma: primeiro são introduzidas as notações que serão utilizadas no texto; a seguir, o problema da elasticidade linear é apresentado em dimensões infinita e finita; logo após, os resultados de experimentos numéricos em malhas de quadrados e trapézios são relatados e discutidos.

## 2 NOTAÇÕES

$\mathbb{M}$  é usado para denotar o conjunto dos tensores de ordem 2 e  $\mathbb{S}$  para denotar o subespaço de  $\mathbb{M}$  formado por tensores simétricos. Seja  $\Omega \subset \mathbb{R}^2$  um aberto limitado com contorno  $\Gamma = \partial\Omega$ . O espaço das funções  $f : \omega \rightarrow \mathbb{F}$  (com  $\mathbb{F} = \mathbb{R}, \mathbb{R}^2, \mathbb{M}$  ou  $\mathbb{S}$ ) cujo quadrado é integrável é denotado por  $L^2(\omega, \mathbb{F})$ , com produto interno  $(u, v)_\omega$  e norma  $\|v\|_\omega^2 = (v, v)_\omega$  definidos da forma usual. O índice  $\omega$  será omitido quando  $\omega = \Omega$ . Considere a seguinte definição para o espaço  $H(\text{div}, \Omega, \mathbb{M})$ :

$$H(\text{div}, \Omega, \mathbb{M}) := \{\boldsymbol{\tau} \in L^2(\Omega, \mathbb{M}); \text{div } \boldsymbol{\tau} \in L^2(\Omega, \mathbb{R}^2)\}$$

e o espaço  $H(\text{div}, \Omega, \mathbb{S})$  definido de forma similar.

A discretização por elementos finitos do domínio  $\Omega$  é feita da forma usual: uma malha  $\mathcal{T}_h$  de elementos quadriláterais convexos  $K$  é definida no domínio  $\Omega_h$ . O conjunto de todas as arestas em  $\mathcal{T}_h$  é denotado por  $\mathcal{E}_h$ . Arestas interiores, isto é, aquelas compartilhadas por dois elementos, formam o conjunto  $\mathcal{E}_h^0$ , enquanto as arestas do contorno formam o conjunto  $\mathcal{E}_h^\partial = \mathcal{E}_h \setminus \mathcal{E}_h^0$ .

O conjunto dos polinômios definidos no domínio  $e \subset \mathbb{R}$  de grau máximo  $k$  é indicado por  $\mathbb{P}_k(e)$ . Essa notação é estendida para um conjunto de arestas  $\mathcal{E}$  como segue:

$$\mathbb{P}_k(\mathcal{E}) = \{v \in \mathbb{P}_k(e) \forall e \in \mathcal{E}\}.$$

$\tilde{\mathbb{P}}_k(K)$  são os polinômios de grau  $k$  definidos no domínio  $K \subset \mathbb{R}^2$  e  $\mathbb{P}_k(K)$  é o conjunto dos polinômios em  $K$  de grau máximo  $k$ . Além disso, a notação  $\mathbb{P}_{r,s}(K)$  é utilizada para indicar o conjunto de polinômios em  $K$  de graus máximos  $r$  em  $x_1$  e  $s$  em  $x_2$ .

Seja  $\mathbf{F}_K : \hat{K} \rightarrow K$  um difeomorfismo do elemento de referência  $\hat{K} = (0, 1)^2 \subset \mathbb{R}^2$  para o quadrilátero convexo  $K$ . Uma função escalar (ou vetorial)  $\hat{\varphi}$  definida em  $\hat{K}$  é transformada para uma função  $\varphi = P_K^0 \hat{\varphi}$  em  $K$  por composição:

$$\varphi(\mathbf{x}) = (P_K^0 \hat{\varphi})(\mathbf{x}) = \hat{\varphi}(\hat{\mathbf{x}}),$$

onde  $\mathbf{x} = \mathbf{F}_K \hat{\mathbf{x}}$ . Funções vetoriais  $\hat{\varphi}$  em  $\hat{K}$  também podem ser transformadas para funções  $\varphi = P_K^1 \hat{\varphi}$  em  $K$  pela transformação de Piola:

$$\hat{\varphi}(\mathbf{x}) = (P_K^1 \hat{\varphi})(\mathbf{x}) = \frac{1}{J_K(\hat{\mathbf{x}})} [\mathbf{D}\mathbf{F}_K(\hat{\mathbf{x}})] \hat{\varphi}(\hat{\mathbf{x}}),$$

onde  $J_K = \det \mathbf{D}\mathbf{F}_K$ . A transformação de Piola conserva a continuidade da componente normal das funções transformadas e por isso é a maneira adequada para transformar funções vetoriais em  $H(\text{div})$ . Ainda, a transformação de Piola também pode ser usada para transformar uma função tensorial  $\varphi(\hat{K})$  para uma função  $\varphi(K)$ . Neste caso a transformação é aplicada a cada linha da representação matricial de  $\varphi$ . Também será usado  $P_K^1$  para denotar essa operação.

## 3 O PROBLEMA DA ELASTICIDADE LINEAR BIDIMENSIONAL

Por simplicidade, o problema da elasticidade linear será formulado como segue:

**Problema da Elasticidade Linear:** Dados  $\tilde{\mathbf{u}} : \Gamma \rightarrow \mathbb{R}^2$  e  $\mathbf{g} : \Omega \rightarrow \mathbb{R}^2$ , encontrar o campo de deslocamentos  $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$  e o campo de tensões efetivas  $\boldsymbol{\sigma} : \Omega \rightarrow \mathbb{S}$  tais que

$$\begin{aligned}\mathbf{A}\boldsymbol{\sigma} &= \boldsymbol{\varepsilon}(\mathbf{u}) \quad \text{em } \Omega, \\ \operatorname{div} \boldsymbol{\sigma} &= \mathbf{g} \quad \text{em } \Omega, \\ \mathbf{u} &= \tilde{\mathbf{u}} \quad \text{sobre } \Gamma,\end{aligned}$$

onde  $\mathbf{A} = \mathbf{A}(\mathbf{x}) : \mathbb{M} \rightarrow \mathbb{M}$  é o tensor de complacência uniformemente limitado em  $\Omega$  e positivo-definido, que no caso isotrópico é dado por

$$\mathbf{A}\boldsymbol{\sigma} = \frac{1}{2\mu} \left( \boldsymbol{\sigma} - \frac{\lambda}{2(\lambda + \mu)} \operatorname{tr} \boldsymbol{\sigma} \mathbf{I} \right),$$

$\lambda \geq 0$  e  $\mu > 0$  são as constantes de Lamé e  $\boldsymbol{\varepsilon}(\mathbf{u}) = \nabla^s \mathbf{u} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$  é a parte simétrica do gradiente de  $\mathbf{u}$ .

A definição para a rotação  $r(\mathbf{u}) = \operatorname{as}(\nabla \mathbf{u})/2$  (onde  $\operatorname{as} \tau = \tau_{12} - \tau_{21}$  mede a assimetria de uma matriz  $\tau \in \mathbb{M}$ ) é usada para reescrever a equação constitutiva da elasticidade:

$$\mathbf{A}\boldsymbol{\sigma} = \nabla \mathbf{u} - \mathbf{R},$$

com

$$\mathbf{R} = \begin{bmatrix} 0 & r \\ -r & 0 \end{bmatrix}.$$

Sejam os espaços  $\mathcal{S} = H(\operatorname{div}, \Omega, \mathbb{M})$ ,  $\mathcal{U} = L^2(\Omega, \mathbb{R}^2)$  e  $\mathcal{R} = L^2(\Omega, \mathbb{R})$ . Assim, a formulação variacional do Problema da Elasticidade Linear será

**Problema Variacional:** Dados  $\tilde{\mathbf{u}} \in H^{1/2}(\Gamma, \mathbb{R}^2)$  e  $\mathbf{g} \in L^2(\Omega, \mathbb{R}^2)$ , encontrar  $(\boldsymbol{\sigma}, \mathbf{u}, r) \in \mathcal{S} \times \mathcal{U} \times \mathcal{R}$  tais que

$$\begin{aligned}(\mathbf{A}\boldsymbol{\sigma}, \boldsymbol{\tau}) + (\mathbf{u}, \operatorname{div} \boldsymbol{\tau}) + (r, \operatorname{as} \boldsymbol{\tau}) &= (\tilde{\mathbf{u}}, \boldsymbol{\tau} \boldsymbol{\nu})_\Gamma, \quad \forall \boldsymbol{\tau} \in \mathcal{S}, \\ (\operatorname{div} \boldsymbol{\sigma}, \boldsymbol{\eta}) &= (\mathbf{g}, \boldsymbol{\eta}), \quad \forall \boldsymbol{\eta} \in \mathcal{U}, \\ (\operatorname{as} \boldsymbol{\sigma}, s) &= 0, \quad \forall s \in \mathcal{R},\end{aligned}$$

onde  $\boldsymbol{\nu}$  é o vetor normal exterior a  $\Gamma$ .

Note que a condição de simetria do tensor de tensão deixou de ser uma restrição às funções do espaço  $\mathcal{S}$ , sendo nesta formulação imposta pela formulação variacional.

## 4 APROXIMAÇÕES POR ELEMENTOS FINITOS

O espaço de aproximação para a tensão é baseado em um produto de espaços RT (Raviart e Thomas, 1977), ABF (Arnold et al., 2005) ou AC (Arbogast e Correa, 2016) em cada elemento:

$$\mathcal{S}_h := \{\boldsymbol{\sigma}_h \in H(\operatorname{div}, \Omega_h, \mathbb{M}); \boldsymbol{\sigma}_h|_K \in \mathcal{V}^2(K) \ \forall K \in \mathcal{T}_h\},$$

**Tabela 1: Comparação entre o número de graus de liberdade locais dos espaços de aproximação.**

	$\mathcal{S}_h$	$\mathcal{U}_h$	$\mathcal{R}_h$	Total
RT <sub>1</sub>	24	8	3	35
ABF <sub>1</sub>	32	16	3	51
AC <sub>1</sub>	20	6	3	29

onde

$$\begin{aligned}\mathcal{V}(K) &= \mathcal{V}^{RT}(K) := P_K^1(\mathbb{P}_{k+1,k}(\hat{K}), \mathbb{P}_{k,k+1}(\hat{K})), \\ \mathcal{V}(K) &= \mathcal{V}^{ABF}(K) := P_K^1(\mathbb{P}_{k+2,k}(\hat{K}), \mathbb{P}_{k,k+2}(\hat{K})) \quad \text{ou} \\ \mathcal{V}(K) &= \mathcal{V}^{AC}(K) := \mathbb{P}_k^2(K) \oplus \mathbf{x}\tilde{\mathbb{P}}_k(K) \oplus \text{span}\{\sigma_1, \sigma_2\},\end{aligned}$$

com  $\sigma_1 = P_K^1(\text{curl}(\hat{x}_1^2 \hat{x}_2))$  e  $\sigma_2 = P_K^1(\text{curl}(\hat{x}_1 \hat{x}_2^2))$ .

Já o espaço para os deslocamentos é polinomial por partes, transformado por composição a partir do elemento de referência (para RT e ABF):

$$\begin{aligned}\mathcal{U}_h^{RT} &:= \{\mathbf{u}_h \in L^2(\Omega_h, \mathbb{R}^2); \mathbf{u}_h|_K \in P_K^0(\mathbb{P}_{k,k}^2(\hat{K})) \forall K \in \mathcal{T}_h\}, \\ \mathcal{U}_h^{ABF} &:= \{\mathbf{u}_h \in L^2(\Omega_h, \mathbb{R}^2); \mathbf{u}_h|_K \in P_K^0(\mathcal{R}_k^2(\hat{K})) \forall K \in \mathcal{T}_h\},\end{aligned}$$

com  $\mathcal{R}_k(\hat{K}) = \mathbb{P}_{k+1,k+1}(\hat{K}) \setminus \text{span}\{\hat{x}_1^{k+1} \hat{x}_2^{k+1}\}$ , ou definido nas coordenadas do elemento (para AC):

$$\mathcal{U}_h^{AC} := \{\mathbf{u}_h \in L^2(\Omega_h, \mathbb{R}^2); \mathbf{u}_h|_K \in \mathbb{P}_k^2(K) \forall K \in \mathcal{T}_h\}.$$

O espaço para as rotações é polinomial por partes, definido nas coordenadas do elemento:

$$\mathcal{R}_h := \{r_h \in L^2(\Omega_h, \mathbb{R}); r_h|_K \in \mathbb{P}_k(K) \forall K \in \mathcal{T}_h\}.$$

Na Tabela 1 é comparado o número de graus de liberdade locais dos espaços definidos acima, para o caso particular  $k = 1$ .

Usando essas definições para os espaços de dimensão finita, o problema discreto é apresentado em uma formulação mista:

**Problema Discreto:** Dados  $\tilde{\mathbf{u}} \in (\mathbb{P}_k^2(\mathcal{E}_h^\partial))'$  e  $\mathbf{g} \in \mathcal{U}'_h$ , encontrar  $(\boldsymbol{\sigma}_h, \mathbf{u}_h, r_h) \in \mathcal{S}_h \times \mathcal{U}_h \times \mathcal{R}_h$  tais que

$$\begin{aligned}(\mathbf{A}\boldsymbol{\sigma}_h, \boldsymbol{\tau}) + (\mathbf{u}_h, \text{div } \boldsymbol{\tau}) + (r_h, \text{as } \boldsymbol{\tau}) &= (\tilde{\mathbf{u}}, \boldsymbol{\tau}\boldsymbol{\nu})_\Gamma, \quad \forall \boldsymbol{\tau} \in \mathcal{S}_h, \\ (\text{div } \boldsymbol{\sigma}_h, \boldsymbol{\eta}) &= (\mathbf{g}, \boldsymbol{\eta}), \quad \forall \boldsymbol{\eta} \in \mathcal{U}_h, \\ (\text{as } \boldsymbol{\sigma}_h, s) &= 0, \quad \forall s \in \mathcal{R}_h.\end{aligned}$$

Na formulação híbrida, os espaços para aproximação dos deslocamentos e das rotações são os mesmos da formulação mista, enquanto a continuidade da tração é imposta pela formulação variacional através de multiplicadores de Lagrange. Assim, as tensões são procuradas no espaço

$$\mathcal{S}_h := \{\boldsymbol{\sigma}_h \in L^2(\Omega_h, \mathbb{M}); \boldsymbol{\sigma}_h|_K \in \mathcal{V}^2(K) \forall K \in \mathcal{T}_h\},$$

com  $\mathcal{V}(K)$  como definido anteriormente. Introduzindo o espaço

$$\mathcal{L}_h^\varphi := \{\hat{\mathbf{u}} \in \mathbb{P}_k^2(e) \forall e \in \mathcal{E}_h; \hat{\mathbf{u}}|_e = \varphi|_e \forall e \in \mathcal{E}_h^\partial\}$$

para o multiplicador de Lagrange, a seguinte formulação hibridizada para o problema da elasticidade pode ser apresentada:

**Problema na Forma Híbrida:** Dados  $\tilde{\mathbf{u}} \in (\mathbb{P}_k^2(\mathcal{E}_h^\partial))'$  e  $\mathbf{g} \in \mathcal{U}'_h$ , encontrar  $(\boldsymbol{\sigma}_h, \mathbf{u}_h, r_h, \hat{\mathbf{u}}_h) \in \mathcal{S}_h \times \mathcal{U}_h \times \mathcal{R}_h \times \mathcal{L}_h^{\tilde{\mathbf{u}}}$  tais que

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \left[ (\mathbf{A}\boldsymbol{\sigma}_h, \boldsymbol{\tau})_K + (\mathbf{u}_h, \operatorname{div} \boldsymbol{\tau})_K + (r_h, \operatorname{as} \boldsymbol{\tau})_K - (\hat{\mathbf{u}}_h, \boldsymbol{\tau} \boldsymbol{\nu})_{\partial K \cap \mathcal{E}_h^0} \right] &= 0, & \forall \boldsymbol{\tau} \in \mathcal{S}_h, \\ \sum_{K \in \mathcal{T}_h} (\operatorname{div} \boldsymbol{\sigma}_h, \boldsymbol{\eta})_K &= \sum_{K \in \mathcal{T}_h} (\mathbf{g}, \boldsymbol{\eta})_K, & \forall \boldsymbol{\eta} \in \mathcal{U}_h, \\ \sum_{K \in \mathcal{T}_h} (\operatorname{as} \boldsymbol{\sigma}_h, s)_K &= 0, & \forall s \in \mathcal{R}_h, \\ \sum_{K \in \mathcal{T}_h} (\boldsymbol{\sigma}_h \boldsymbol{\nu}, \hat{\boldsymbol{\eta}})_{\partial K \cap \mathcal{E}_h^0} &= 0, & \forall \hat{\boldsymbol{\eta}} \in \mathcal{L}_h^0, \end{aligned}$$

onde a última equação impõe fracamente a continuidade da tração nas interfaces dos elementos.

## 5 RESULTADOS NUMÉRICOS

Nesta seção são relatados resultados numéricos correspondentes a estudos de convergência das aproximações propostas para o problema da elasticidade linear em sua formulação híbrida. O domínio é o quadrado  $(-1, 1) \times (-1, 1)$  e são consideradas duas sequências de malhas: uma construída pela subdivisão regular em quadrados e outra com malhas em que cada elemento é congruente a um trapezoide (Fig. 1). Condições de contorno de Dirichlet são impostas em  $\hat{\mathbf{u}}$  sobre o contorno  $\Gamma$  e a solução exata

$$\mathbf{u}(x, y) = \begin{bmatrix} \cos(\pi x) \sin(\pi y) \\ \sin(\pi x) \cos(\pi y) \end{bmatrix}$$

é usada, juntamente aos parâmetros  $\lambda = 123$  e  $\mu = 79,3$  para determinar a força de corpo  $\mathbf{g}$ .

A convergência de aproximações utilizando os espaços de dimensão finita apresentados na seção anterior é estudada em malhas de  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$  e  $512 \times 512$  elementos quadrados e trapezoidais. Os resultados são exibidos na Tabela 2 para as malhas de quadrados e na Tabela 3 para as malhas de trapézios.

## 6 DISCUSSÕES E CONSIDERAÇÕES FINAIS

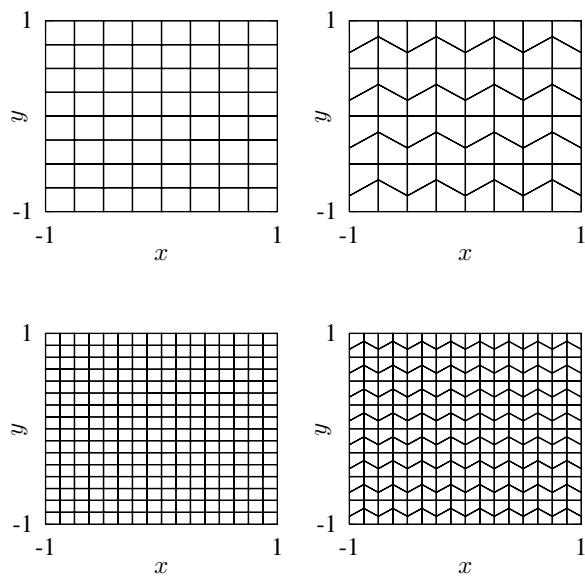
Conforme já observado por Arnold et al. (2015), a escolha dos espaços apresentados por Raviart e Thomas (1977) para aproximação do problema da elasticidade linear mostra-se adequada apenas em malhas com quadriláteros que são paralelogramos. Uma alternativa viável é buscar aproximações em um espaço formado por um produto dos espaços sugeridos por Arnold

**Tabela 2: Resultados de convergência em malhas de quadrados.**

n	$\ \boldsymbol{\sigma} - \boldsymbol{\sigma}_h\ $		$\ \operatorname{div}(\boldsymbol{\sigma} - \boldsymbol{\sigma}_h)\ $		$\ \mathbf{u} - \mathbf{u}_h\ $		$\ r - r_h\ $	
	erro	ordem	erro	ordem	erro	ordem	erro	ordem
<b>RT<sub>1</sub></b>								
8	$6,271 \cdot 10^1$	1,96	$2,590 \cdot 10^2$	1,95	$6,593 \cdot 10^{-2}$	1,96	$2,044 \cdot 10^{-2}$	2,88
16	$1,580 \cdot 10^1$	1,99	$6,528 \cdot 10^1$	1,99	$1,661 \cdot 10^{-2}$	1,99	$2,555 \cdot 10^{-3}$	3,00
32	$3,957 \cdot 10^0$	2,00	$1,635 \cdot 10^1$	2,00	$4,160 \cdot 10^{-3}$	2,00	$3,182 \cdot 10^{-4}$	3,01
64	$9,897 \cdot 10^{-1}$	2,00	$4,090 \cdot 10^0$	2,00	$1,041 \cdot 10^{-3}$	2,00	$3,972 \cdot 10^{-5}$	3,00
128	$2,474 \cdot 10^{-1}$	2,00	$1,023 \cdot 10^0$	2,00	$2,602 \cdot 10^{-4}$	2,00	$4,964 \cdot 10^{-6}$	3,00
256	$6,186 \cdot 10^{-2}$	2,00	$2,557 \cdot 10^{-1}$	2,00	$6,505 \cdot 10^{-5}$	2,00	$6,204 \cdot 10^{-7}$	3,00
512	$1,547 \cdot 10^{-2}$	2,00	$6,392 \cdot 10^{-2}$	2,00	$1,626 \cdot 10^{-5}$	2,00	$7,755 \cdot 10^{-8}$	3,00
<b>ABF<sub>1</sub></b>								
8	$6,310 \cdot 10^1$	1,98	$2,202 \cdot 10^1$	3,12	$7,240 \cdot 10^{-3}$	3,11	$1,974 \cdot 10^{-2}$	2,72
16	$1,582 \cdot 10^1$	2,00	$2,672 \cdot 10^0$	3,04	$8,837 \cdot 10^{-4}$	3,03	$2,536 \cdot 10^{-3}$	2,96
32	$3,958 \cdot 10^0$	2,00	$3,313 \cdot 10^{-1}$	3,01	$1,098 \cdot 10^{-4}$	3,01	$3,177 \cdot 10^{-4}$	3,00
64	$9,898 \cdot 10^{-1}$	2,00	$4,133 \cdot 10^{-2}$	3,00	$1,371 \cdot 10^{-5}$	3,00	$3,971 \cdot 10^{-5}$	3,00
128	$2,475 \cdot 10^{-1}$	2,00	$5,164 \cdot 10^{-3}$	3,00	$1,713 \cdot 10^{-6}$	3,00	$4,963 \cdot 10^{-6}$	3,00
256	$6,186 \cdot 10^{-2}$	2,00	$6,454 \cdot 10^{-4}$	3,00	$2,141 \cdot 10^{-7}$	3,00	$6,204 \cdot 10^{-7}$	3,00
512	$1,547 \cdot 10^{-2}$	2,00	$8,067 \cdot 10^{-5}$	3,00	$2,676 \cdot 10^{-8}$	3,00	$7,755 \cdot 10^{-8}$	3,00
<b>AC<sub>1</sub></b>								
8	$6,430 \cdot 10^1$	2,09	$4,230 \cdot 10^2$	1,86	$1,073 \cdot 10^{-1}$	1,87	$1,242 \cdot 10^{-2}$	3,17
16	$1,589 \cdot 10^1$	2,02	$1,083 \cdot 10^2$	1,97	$2,752 \cdot 10^{-2}$	1,96	$2,258 \cdot 10^{-3}$	2,46
32	$3,962 \cdot 10^0$	2,00	$2,723 \cdot 10^1$	1,99	$6,925 \cdot 10^{-3}$	1,99	$3,085 \cdot 10^{-4}$	2,87
64	$9,900 \cdot 10^{-1}$	2,00	$6,816 \cdot 10^0$	2,00	$1,734 \cdot 10^{-3}$	2,00	$3,942 \cdot 10^{-5}$	2,97
128	$2,475 \cdot 10^{-1}$	2,00	$1,705 \cdot 10^0$	2,00	$4,337 \cdot 10^{-4}$	2,00	$4,954 \cdot 10^{-6}$	2,99
256	$6,187 \cdot 10^{-2}$	2,00	$4,262 \cdot 10^{-1}$	2,00	$1,084 \cdot 10^{-4}$	2,00	$6,201 \cdot 10^{-7}$	3,00
512	$1,547 \cdot 10^{-2}$	2,00	$1,066 \cdot 10^{-1}$	2,00	$2,711 \cdot 10^{-5}$	2,00	$7,754 \cdot 10^{-8}$	3,00

**Tabela 3: Resultados de convergência em malhas de trapézios.**

n	$\ \boldsymbol{\sigma} - \boldsymbol{\sigma}_h\ $		$\ \operatorname{div}(\boldsymbol{\sigma} - \boldsymbol{\sigma}_h)\ $		$\ \mathbf{u} - \mathbf{u}_h\ $		$\ r - r_h\ $	
	erro	ordem	erro	ordem	erro	ordem	erro	ordem
<b>RT<sub>1</sub></b>								
8	$7,080 \cdot 10^1$	1,98	$3,899 \cdot 10^2$	1,72	$7,751 \cdot 10^{-2}$	1,93	$4,141 \cdot 10^{-2}$	2,51
16	$1,785 \cdot 10^1$	1,99	$1,458 \cdot 10^2$	1,42	$1,964 \cdot 10^{-2}$	1,98	$7,705 \cdot 10^{-3}$	2,43
32	$4,471 \cdot 10^0$	2,00	$6,508 \cdot 10^1$	1,16	$4,927 \cdot 10^{-3}$	1,99	$1,666 \cdot 10^{-3}$	2,21
64	$1,118 \cdot 10^0$	2,00	$3,147 \cdot 10^1$	1,05	$1,233 \cdot 10^{-3}$	2,00	$3,972 \cdot 10^{-4}$	2,07
128	$2,796 \cdot 10^{-1}$	2,00	$1,560 \cdot 10^1$	1,01	$3,083 \cdot 10^{-4}$	2,00	$9,802 \cdot 10^{-5}$	2,02
256	$6,990 \cdot 10^{-2}$	2,00	$7,783 \cdot 10^0$	1,00	$7,707 \cdot 10^{-5}$	2,00	$2,442 \cdot 10^{-5}$	2,00
512	$1,748 \cdot 10^{-2}$	2,00	$3,889 \cdot 10^0$	1,00	$1,927 \cdot 10^{-5}$	2,00	$6,100 \cdot 10^{-6}$	2,00
<b>ABF<sub>1</sub></b>								
8	$7,107 \cdot 10^1$	1,99	$5,291 \cdot 10^1$	2,66	$1,149 \cdot 10^{-2}$	3,01	$3,360 \cdot 10^{-2}$	2,20
16	$1,790 \cdot 10^1$	1,99	$1,069 \cdot 10^1$	2,31	$1,467 \cdot 10^{-3}$	2,97	$8,067 \cdot 10^{-3}$	2,06
32	$4,482 \cdot 10^0$	2,00	$2,489 \cdot 10^0$	2,10	$1,955 \cdot 10^{-4}$	2,91	$2,012 \cdot 10^{-3}$	2,00
64	$1,121 \cdot 10^0$	2,00	$6,104 \cdot 10^{-1}$	2,03	$2,961 \cdot 10^{-5}$	2,72	$5,051 \cdot 10^{-4}$	1,99
128	$2,803 \cdot 10^{-1}$	2,00	$1,519 \cdot 10^{-1}$	2,01	$5,580 \cdot 10^{-6}$	2,41	$1,267 \cdot 10^{-4}$	1,99
256	$7,007 \cdot 10^{-2}$	2,00	$3,792 \cdot 10^{-2}$	2,00	$1,255 \cdot 10^{-6}$	2,15	$3,174 \cdot 10^{-5}$	2,00
512	$1,752 \cdot 10^{-2}$	2,00	$9,476 \cdot 10^{-3}$	2,00	$3,045 \cdot 10^{-7}$	2,04	$7,945 \cdot 10^{-6}$	2,00
<b>AC<sub>1</sub></b>								
8	$8,346 \cdot 10^1$	1,95	$4,457 \cdot 10^2$	1,86	$1,136 \cdot 10^{-1}$	1,85	$2,422 \cdot 10^{-1}$	0,93
16	$2,361 \cdot 10^1$	1,82	$1,144 \cdot 10^2$	1,96	$2,923 \cdot 10^{-2}$	1,96	$2,027 \cdot 10^{-1}$	0,26
32	$6,836 \cdot 10^0$	1,79	$2,880 \cdot 10^1$	1,99	$7,350 \cdot 10^{-3}$	1,99	$1,113 \cdot 10^{-1}$	0,86
64	$1,849 \cdot 10^0$	1,89	$7,213 \cdot 10^0$	2,00	$1,837 \cdot 10^{-3}$	2,00	$3,757 \cdot 10^{-2}$	1,57
128	$4,741 \cdot 10^{-1}$	1,96	$1,804 \cdot 10^0$	2,00	$4,591 \cdot 10^{-4}$	2,00	$1,021 \cdot 10^{-2}$	1,88
256	$1,194 \cdot 10^{-1}$	1,99	$4,510 \cdot 10^{-1}$	2,00	$1,148 \cdot 10^{-4}$	2,00	$2,607 \cdot 10^{-3}$	1,97
512	$2,990 \cdot 10^{-2}$	2,00	$1,128 \cdot 10^{-1}$	2,00	$2,869 \cdot 10^{-5}$	2,00	$6,553 \cdot 10^{-4}$	1,99



**Figura 1: Malhas de quadrados e trapézios.**

et al. (2005). Entretanto essa estratégia resulta na adição de um significativo número de graus de liberdade nos problemas locais. A terceira escolha possível (utilizar produtos dos espaços de Arbogast e Correa (2016)) resulta em aproximações que apresentam boas taxas de convergência para os deslocamentos em  $L^2(\Omega, \mathbb{R}^2)$  e para as tensões na seminorma de  $H(\text{div}, \Omega, \mathbb{M})$ . No entanto há alguma deficiência na convergência de tensões na norma  $L^2(\Omega, \mathbb{M})$  e de rotações na norma  $L^2(\Omega, \mathbb{R})$ , sobretudo nas malhas menos refinadas.

A relevância da utilização de uma formulação híbrida deve ser ressaltada, já que com essa técnica os graus de liberdade do sistema global resumem-se àqueles relacionados ao multiplicador de Lagrange nas arestas. Com isso, o custo computacional da resolução do sistema linear global é o mesmo em todos os casos aqui considerados. Ainda, os espaços de aproximação para as tensões, os deslocamentos e as rotações podem ser construídos de forma local, elemento a elemento.

## AGRADECIMENTOS

Thiago Quinelato recebeu auxílio financeiro do CNPq (Processo 141009/2013-6) e da Capes (Processo BEX 6993/15-0) durante a realização deste trabalho.

## REFERÊNCIAS

- M. Amara e J. M. Thomas, 1979. Equilibrium finite elements for the linear elastic problem. *Numerische Mathematik*, vol. 33, n. 4, pp. 367–383.
- T. Arbogast e M. R. Correa, 2016. Two families of  $H(\text{div})$  mixed finite elements on quadrilaterals of minimal dimension. *SIAM Journal on Numerical Analysis*, vol. 54, n. 6, pp. 3332–3356. doi: 10.1137/15M1013705.

- D. N. Arnold, F. Brezzi, e J. Douglas Jr, 1984. Peers: A new mixed finite element for plane elasticity. *Japan Journal of Applied Mathematics*, vol. 1, n. 2, pp. 347–367.
- D. N. Arnold, D. Boffi, e R. S. Falk, 2005. Quadrilateral  $H(\text{div})$  finite elements. *SIAM J. Numer. Anal.*, vol. 42, n. 6, pp. 2429–2451.
- D. N. Arnold, R. S. Falk, e R. Winther, 2006. Differential complexes and stability of finite element methods ii: The elasticity complex. In D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, e M. Shashkov, eds, *Compatible Spatial Discretizations*, volume 142 de *The IMA Volumes in Mathematics and its Applications*, pp. 47–67. Springer New York.
- D. N. Arnold, G. Awanou, e W. Qiu, 2015. Mixed finite elements for elasticity on quadrilateral meshes. *Advances in Computational Mathematics*, vol. 41, n. 3, pp. 553–572. ISSN 1019–7168.
- P. A. Raviart e J. M. Thomas, 1977. A mixed finite element method for 2-nd order elliptic problems. In Ilio Galligani e Enrico Magenes, eds, *Mathematical Aspects of Finite Element Methods*, volume 606 de *Lecture Notes in Mathematics*, pp. 292–315. Springer Berlin / Heidelberg.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### Ranking Texture Features Through AdaBoost.M2 Linear Ensembles for Granite Tiles Classification

**Tiene Andre Filisbino**

**Gilson Antonio Giraldi**

tiene@lncc.br

gilson@lncc.br

Laboratório Nacional de Computação Científica

Getúlio Vargas Av., 333, Quitandinha, CEP 25651-075, RJ, Petrópolis, Brasil

**Carlos Eduardo Thomaz**

cet@fei.edu.br

Department of Electrical Engineering, FEI

São Bernardo do Campo, 09850-901, SP, Brasil

**Bruno Montandon Noronha Barros**

bbarros@cetem.gov.br

CETEM/MCTIC

Cidade Universitária, 21941-908, RJ, Rio de Janeiro, Brasil

**Millena Basílio da Silva**

mbasilio@cetem.gov.br

CETEM-ES/MCTIC

Morro Grande, 29.300-970, Cachoeiro de Itapemirim, ES, Brasil

---

EAMC 2017

Anais de congresso do X Encontro Acadêmico de Modelagem Computacional  
Petrópolis, RJ, Brasil, 01 a 03 de Fevereiro, 2017

**Abstract.** *Texture analysis is a field of image processing that has been extensively applied for object recognition. In the specific case of automatic classification of granite tiles, the image acquisition can generate large resolution data. So, we need to construct efficient feature spaces in order to reduce the computational cost of further steps. The co-occurrence matrix and Haralick features are techniques to compute the feature vectors to represent the image samples. These methodologies are classical in pattern recognition and texture image analysis. In this paper, we apply the Multi-Class.M2 DPCA methodology for ranking texture features for granite tiles recognition. Hence, given a multi-class granite image database, we build multiple linear learners that are combined through an ensemble technique, the AdaBoost.M2, in order to determine the discriminant contribution of each feature. We implement the linear learners using the support vector machine (SVM). The strong learner built by the ensemble technique is processed following a strategy to get the global discriminant vector to sort texture features according to their relevance for classification tasks. In the computational experiments we analyse the obtained approach using a five-class granite image database. Our experimental results have shown that the features selected by the proposed technique allow competitive recognition rates when compared with related methods.*

**Palavras-chave:** *Texture, Feature Space, Ranking Texture features, Multi-class Recognition, AdaBoost.*

## 1 Introduction

Image analysis is one of the main tasks involved for pattern recognition in image databases. The main goal is to get enough information to separate different image regions or to distinguish sample groups in classification tasks [1]. The acquisition of information for different patterns can be obtained through the computation of texture features (TF) and subsequent analysis [2]. Texture analysis [2] is a field of image processing that has been extensively applied for object recognition. A particular case is automatic classification of granite tiles, where the image acquisition can generate large resolution data. So, we need efficient methods to construct feature spaces in order to reduce the computational cost of further steps. The co-occurrence matrix and Haralick's texture features, or simply Haralick's descriptors, have been applied to compute the feature vectors to represent the sample images in such application [3]. These methodologies are classical in the pattern recognition and texture image analysis [4].

A co-occurrence matrix for a given image is generated by computing the distribution of co-occurring pixel values (gray scale values, or colors) at a given offset. Hence, we can describe texture through a set of features considering multiple directions defined by a set of offsets. To summarize the information contained in the co-occurrence matrix we follow [5] and compute the Haralick's texture features [6], in order to generate the feature space. In this paper, we apply the Multi-Class.M2 DPCA methodology described in [7, 8], for discriminant analysis (ranking texture features) on a multiclass problems in granite tiles recognition.

Given an  $N$ -class granite tiles database, the Multi-Class DPCA.M2 algorithm (MDPCA.M2) [7] builds a linear support vector machine (SVM) ensemble, composed of  $N$  SVM machines, to get the discriminant weights that are combined through the AdaBoost technique in order to determine the discriminant contribution of each texture features. The Multi-Class DPCA.M2 methodology applies the Multi-Class.M2 algorithm described in [7, 8] and combines the separating SVM hyperplanes through a simple strategy to compute the global discriminant weights for ranking texture features given by the group-differences. Also, we follow [9] and apply the Multi-Class.M2 methodology to compute discriminant weights through the linear discriminant analysis (LDA) [1]. We have focused here on the SVM and LDA methods but any other separating hyperplane could be used. Besides, we verify the consequences of ranking techniques in textures feature through Fisher criterion and the Multi-Class DPCA technique [10, 11]. The computational experiments are performed using only 5 of RGB granite images with resolution  $4488 \times 4488$ , obtained through a scanner from multifunctional HP Deskjet 2050. Before computation, we convert each image to gray scale and perform a partition of the images into blocks in order to better explore the high resolution data for recognition tasks. The results show that the features selected by the proposed technique allow competitive recognition rates.

The paper is organized as follows. In section 2 we presented the techniques to feature extraction. Next, section 3 describes the Multi-Class.M2 DPCA for texture analysis. The computational experiments are presented in section 4. Finally, in section 5, we conclude the paper, summarizing its main contributions and describing further developments.

## 2 Feature Extraction

In this paper, the process of image feature extraction consists in the computation of Haralick's texture features through the gray level co-occurrence matrix (GLCM), computed considering multiple offsets, to represent the visual information [5]. So, given an  $M \times N$  image

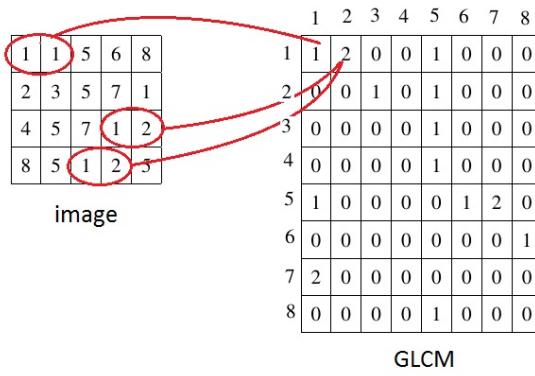
$I$  coded using  $L$  grey levels in the set  $\{0, 1, 2, \dots, L - 1\}$  and an offset  $(\Delta x, \Delta y)$ , the GLCM matrix is a  $L \times L$  two-dimensional array define by:

$$GLCM(i, j, \Delta x, \Delta y, I) = \sum_{x=1}^M \sum_{y=1}^N \delta(I(x, y) - i, I(x + \Delta x, y + \Delta y) - j), 0 \leq i, j \leq (L - 1) \quad (1)$$

where the function  $\delta$  is defined as:  $\delta(m, n) = 1$ , if  $m = n = 0$ , and,  $\delta(m, n) = 0$ , otherwise.

Therefore, the GLCM characterizes the texture of an image by calculating how often pairs of pixels  $((x, y), (x + \Delta x, y + \Delta y))$  with the gray level intensities  $(i, j)$  occur in a specific offset  $(\Delta x, \Delta y)$  [5]. We shall emphasize that we are working in the digital context and, consequently, expression (1) only makes sense if  $\Delta x, \Delta y \in \mathbb{N}$ .

Figure 1 illustrates the process to build the GLCM by only walking in horizontal direction,  $(\Delta x, \Delta y) = (1, 0)$ .



**Figure 1: Process to create GLCM for image  $I$ , with  $L = 8$  in expression (1).**

In the Figure 1,  $GLCM(1, 1, 1, 0, I) = 1$  because there is only one instance in the input image  $I$  where two horizontally adjacent pixels  $I(x, y)$  and  $I(x + 1, y)$  have values  $I(x, y) = 1$  and  $I(x + 1, y) = 1$ . Analogously, the  $GLCM(1, 2, 1, 0, I) = 2$  because there are two pairs of pixels such that  $I(x, y) = 1$  and  $I(x + 1, y) = 2$ , and so on.

In this paper, the process of image feature extraction consists in calculating the Haralick's texture features through the GLCM matrix computed considering multiple directions to represent the visual information [12, 5].

Haralick extracted 14 descriptors from the co-occurrence matrix [6], but only five texture features are frequently used due to correlations between the descriptors [3]. So, in this work, we use only the five texture features presented in Table (1). According to [3], these descriptors are adequate to give good results in classification task.

In this work, for each Haralick's Descriptors presented in Table (1), we consider the offsets  $(\Delta x, \Delta y) \in \{(1, 0), (1, 1), (0, 1), (-1, -1)\}$ . Besides,  $GLCM(i, j, \Delta x, \Delta y, I)$  that is defined by expression (1), is normalized such that the sum of its elements is equal to one [6]. In the above table,  $\mu_x$  and  $\mu_y$  are the horizontal and vertical mean while  $\sigma_x$  and  $\sigma_y$  denote the horizon-

Haralick's Descriptors	
Energy	$\sum_i \sum_j (GLCM(i, j, \Delta x, \Delta y, I))^2$
Contrast	$\sum_i \sum_j  i - j ^2 GLCM(i, j, \Delta x, \Delta y, I)$
Correlation	$\frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) GLCM(i, j, \Delta x, \Delta y, I)}{\sigma_i \sigma_j}$
Homogeneity	$\frac{\sum_i \sum_j GLCM(i, j, \Delta x, \Delta y, I)}{1 +  i - j }$
Entropy	$\sum_i \sum_j GLCM(i, j, \Delta x, \Delta y, I) \log(GLCM(i, j, \Delta x, \Delta y, I))$

**Table 1: Five texture features**

tal and vertical standard deviations, defined by expressions (2)-(3):

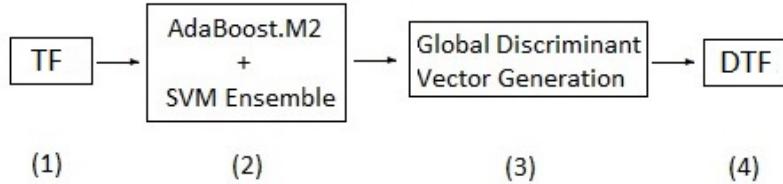
$$\mu_x = \sum_i \sum_j i GLCM(i, j, \Delta x, \Delta y, I) \quad \text{and} \quad \mu_y = \sum_j \sum_i j GLCM(i, j, \Delta x, \Delta y, I) \quad (2)$$

$$\sigma_x = \sqrt{\sum_i \sum_j (i - \mu_x)^2 GLCM(i, j, \Delta x, \Delta y, I)} \quad \text{and} \quad \sigma_y = \sqrt{\sum_i \sum_j (j - \mu_y)^2 GLCM(i, j, \Delta x, \Delta y, I)} \quad (3)$$

### 3 Multi-Class Discriminant Analysis

The whole texture feature analysis is summarized in the pipeline of Figure 2 and the Multi-Class.M2 DPCA methodology is presented in the Algorithm 1. We follow [11, 7] and apply the technique presented in Section 2 for build texture features in the step (1) of the pipeline. Then, in step (2) of Figure 2, we compute a set of linear SVM hyperplanes, based on the “one-against-all” SVM multi-class approach. We also apply an ensemble technique, the AdaBoost.M2 algorithm, to combine the linear classifiers in order to compute the global discriminant vector. The key idea of this step is based on the fact that AdaBoost.M2 linearly combines weak classifiers to get the strong hypothesis. So, it is straightforward to obtain the global discriminant weights from the expression that defines the strong classifier by using a simple scheme, that corresponds to step (3) of Figure 2. This strategy can be also used to combine discriminant directions computed by other multi-class approaches, like linear discriminant analysis (LDA).

However, it is known that a strong learner like SVM does not work well as the base component for AdaBoost [13]. Therefore, we follow [13] and implement a strategy to compute a weakened version of SVM that is useful as an AdaBoost.M2 feature [9].

**Figure 2: Flowchart with main steps of the proposed technique.**

Finally, in the stage (4), we sort discriminant texture features (DTF) in the decreasing order of the global discriminant weights. The method is not restricted to any particular probability density function of the sample groups because it can be based on either a parametric or non-parametric separating hyperplane approaches.

In the Algorithm 1, at the input of the procedure, the training instances in the database  $X \subset \Re^n$  are supposed independently and identically distributed from an uniform distribution  $D$ . In line 9 of MDPCA.M2 algorithm, each weak learner generates an hypotheses, which has the form  $h : X \times Y \rightarrow [0, 1]$ , and can be interpreted as the probability that  $y$  is the correct label associated with instance  $\mathbf{x}$ . So, given a sample  $\mathbf{x}_i$ , the probability of choosing an incorrect label  $y$  is [9]:  $Pr = \frac{1}{2}(1 - h(\mathbf{x}_i, y_i) + h(\mathbf{x}_i, y))$ .

However, we have  $|Y| - 1$  possibilities to obtain the incorrect answer. So, we can define the loss of the hypothesis through a weighted average according to some  $q_{i,y}$ , called the label weighting function, that assigns to each example  $i$  in the training set a load, with  $\sum_{y \neq y_i} q_{i,y} = 1$ . The resulting formula is called the pseudo-loss of  $h$  on training instance  $i$  with respect to  $q$  [9]:

$$ploss_q(h, i) = \frac{1}{2} \left( 1 - h(\mathbf{x}_i, y_i) + \sum_{y \neq y_i} q_{i,y} h(\mathbf{x}_i, y) \right). \quad (4)$$

So, following the AdaBoost.M2 strategy [9], in each iteration  $t$  of the Algorithm 1, the weak learner's goal is to minimize the expected pseudo-loss, computed in line 10 of the Algorithm 1, for a distribution  $D^t$  and weighting function  $q^t$ .

Next, the MDPCA.M2 computes a set of SVM hyperplanes, based on the one-against-all SVM multi-class approach presented in [14]. Hence, as we have  $N$  classes, the internal loop in the Algorithm 1 (line 6 to 9) constructs  $N$  weakened SVMs, in the texture features, using the Algorithm 2. In line 7 of Algorithm 1 we buid the  $\Theta^y$  set by taking all  $k_y$  projected samples from class  $y$  and label them as 1. Then, using random sampling we choose  $(2k_y)/(N - 1)$  projected samples from classes other than  $y$  and label them as -1. The obtained set of feature vectors  $\mathbf{x}_m^y \in \Re^{m'}$  and corresponding labels  $y_m \in \{-1, 1\}$ :

$$\Theta^y = \left\{ (\mathbf{x}_1^y, l_1), (\mathbf{x}_2^y, l_2), \dots, (\mathbf{x}_{3k_y}^y, l_{3k_y}) \right\}, \quad (5)$$

are the input to call the Algorithm 2 which construct the weak SVM (WSVM) model  $y$ , represented by a hyperplane direction  $(\phi_y^t)$  and a linear coefficient  $(b_y^t)$ .

The lines 16-18 of the Algorithm 1 are based on the AdaBoost.M2 idea of deriving a strong learner  $h_f$  by using the linear combination of weak (WSVM, in our case) learners  $h^1, h^2, \dots, h^T$ :

$$h_f(\mathbf{x}) = \arg \max_{y \in Y} \sum_{t=1}^T \tilde{\alpha}^t h^t(\mathbf{x}, y), \quad (7)$$

where  $\tilde{\alpha}^t$  is computed in line 16. This expression offers the possibility of extending the DPCA methodology to multi-class problems using the Adaboost.M2 result.

The output of the MDPCA.M2 procedure is the discriminant texture components  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{m'}$  where  $\mathbf{q}_i$  is a texture feature component selected according to its discriminant weight  $v(i)$ .

On the other hand, if we compute the LDA in the texture feature space, we get  $N - 1$  hyperplane directions  $\phi_{lda}^i \in \Re^{m'}, i = 1, 2, \dots, (N - 1)$ . Consequently, we obtain in this case a LDA weight matrix  $\phi_{lda}^{i,j}$ , which can be processed according to lines 19-20 of Algorithm 1, by just replacing  $\Phi_{i,y}$  by  $\phi_{lda}^{i,j}$ . The obtained global discriminant weights are named Multi-Class LDA-DPCA in the following sections.

---

**Algorithm 1:** Multi-Class.M2 DPCA Procedure

---

**Input:** Samples:  $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_M, y_M)\}$ ; where  $y_i \in Y$  and  $Y = \{1, 2, 3, \dots, N\}$ ; Distribution  $D$  over the  $M$  examples; Percentage  $\mu$ ;

- 1 Initialize the weight vector:  $w_{i,y}^1 = \frac{D(i)}{|Y|-1}$ , for  $i = 1, \dots, M; y \in Y - \{y_i\}$
- 2 Calculate texture features ( $TF$ ) through of GLCM.
- 3 Build the labeled TF data set  $\Theta = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_M, y_M)\}$
- 4 **for**  $t = 1, \dots$  **to**  $T$  **do**
- 5     **for**  $y \neq y_i$ :  $q_{i,y}^t = \frac{w_{i,y}^t}{W_i^t}$ ; and set  $D^t(i) = \frac{W_i^t}{\sum_{i=1}^N W_i^t}$
- 6     **for**  $y = 1, \dots$  **to**  $N$  **do**
- 7         Build the subset  $\Theta^y$ , given by expression (5);
- 8          $(\phi_y^t, b_y^t) = WSVM(\Theta^y, \mathcal{Y}, D^t, \mu)$  where  $\mathcal{Y} = \{-1, 1\}$ ;
- 9         Get hypothesis  $h^t : X \times Y \rightarrow [0, 1]$ , given by  $h^t(\mathbf{x}, y) = f(\langle \mathbf{x}, \phi_y^t \rangle + b_y^t)$
- 10         Compute:  

$$e^t = \frac{1}{2} \sum_{i=1}^N D^t(i) \left( 1 - h^t(\mathbf{x}_i, y_i) + \sum_{y \neq y_i} q_{i,y}^t h^t(\mathbf{x}_i, y) \right)$$
- 11         **if**  $e^t > 0.5$  **then**
- 12             **break**;
- 13         Calculate AdaBoost.M2 weights:  $\alpha^t = \frac{1}{2} \ln \left( \frac{1-e^t}{e^t} \right)$ ;
- 14         **for**  $i = 1, \dots, N$  and  $y \in Y - \{y_i\}$  **do**
- 15             **Update**:  $w_{i,y}^{t+1} = w_{i,y}^t \exp(-\alpha^t(1 - h^t(x_i, y_i) + h^t(x_i, y)))$ ;
- 16     Normalize  $\tilde{\alpha}^t = \alpha^t / \sum_{j=1}^T \alpha^j$ ,  $t = 1, 2, \dots, T$
- 17     **for**  $i = 1, \dots$  **to**  $m'$  **do**
- 18          $|\Phi_{i,y}| = \left| \sum_{t=1}^T \tilde{\alpha}^t \frac{\phi_{i,y}^t}{z_{max,y}^t - z_{min,y}^t} \right|, y \in Y$  (6)
- 19     Compute  $v(i) = \max_{y \in Y} \{|\Phi_{i,y}|\}$ ,  $i = 1, 2, \dots, m'$
- 20     Sort discriminant weights:  $v(1) \geq v(2) \geq \dots v(m')$
- 21     Select the texture feature following  $v(i)$

**Output:** discriminant texture components:  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{m'}$

---

**Algorithm 2:** WSVM Procedure: Build a Weakened version of SVM.

---

**Input:** Labeled samples:  $X = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, n'\}$  where  $y_i \in Y$  is the label of the sample  $\mathbf{x}_i$ ;  
Samples probability distribution  $D(\mathbf{x}_i)$ ;  
Percentage  $\mu$ ;  
Select  $\mathcal{J}$  so that,  $\sum_{j \in \mathcal{J}} D(\mathbf{x}_j) \leq (1 - \mu)$ ;  
Select  $(\mathbf{x}_i, y_i); i \in \mathcal{J}$ , and define  $D^* = D_{\mathcal{J}}$ ;  
Compute the weighted data  $X^* = \{(D_i^* \cdot \mathbf{x}_i, y_i), i \in \mathcal{J}\}$   
Compute the (weak) SVM hyperplane  $\phi_{svm}$  using  $X^*$ ;  
**Output:** WSVM hyperplane  $\phi_{svm}$ ,  $b$ .

---

## 4 Computational Experiments

In this section we perform granite images experiments maintained by CETEM [15]. We take five different granite tiles that are showed in Figure 3. In order to save memory allocation along the algorithms execution, we convert each granite to gray scale and resize it to  $4480 \times 4480$  before computation. The high spatial resolution of each image allows to subdivide it into smaller blocks that keep the texture patterns. So, each image is partitioned into blocks using crop windows with size  $280 \times 280$  and  $320 \times 320$ , generation new databases that we call  $DB280$  and  $DB320$ , with 256 and 196 images, respectively. Next we take 128 and 98 images for training and 128 and 98 for test set obtain 640 and 490 images for train and 640 and 490 test respectively.



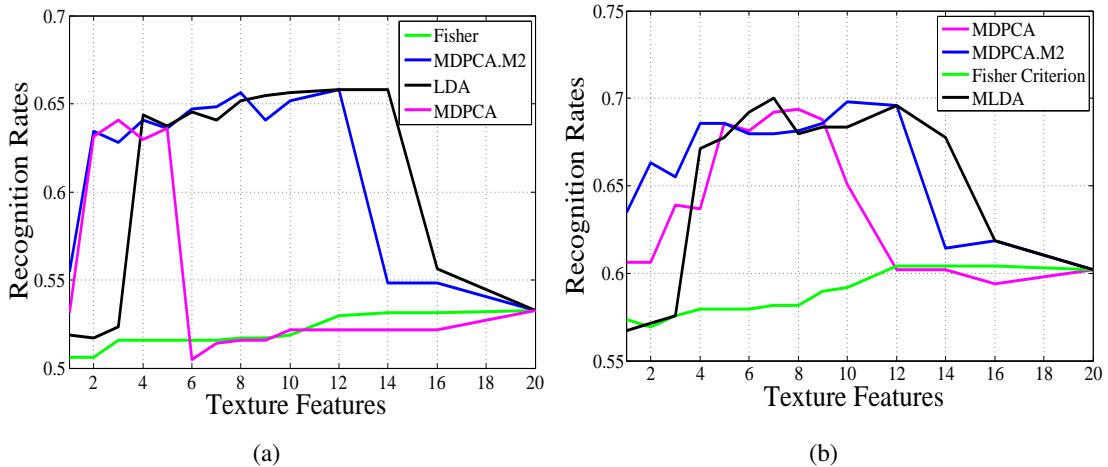
**Figure 3: Granite Image database from CETEM**

In the following, we consider the Multi-Class DPCA (MDPCA) and Fisher Criterion discriminant technique, presented in [11, 7] and the discriminant approaches explained in the section 3: Multi-Class.M2 DPCA (MDPCA.M2) and the Multi-Class LDA-DPCA (MLDA).

In the texture space, the mean of each class  $i$  has been calculated from the corresponding training images and the Euclidean distance from each class mean  $\hat{\mathbf{x}}_i$  has been used to assign a test observation  $\mathbf{x}_r$  to either the different granite tiles. That is, we have assigned  $\mathbf{x}_r$  to class  $i$  that minimizes:

$$d_i(\mathbf{x}_r) = \sqrt{\sum_{j=1}^k (x_{rj} - \hat{x}_{ij})^2}, \quad (8)$$

where  $x_{rj}$  and  $\hat{x}_{ij}$  are the sample  $\mathbf{x}_r$  and the mean  $\hat{\mathbf{x}}_i$ , respectively, in the  $j$ th texture feature considered.



**Figure 4: Recognition rates to texture features considering 32 gray level**

The Figure 4 shows the average recognition rates of the 10-fold cross validation experiments of texture features using the discriminant techniques for the five-class classification problems above mentioned. The recognition rates presented in Figures 4.(a)-(b) were computed using the databases *DB280* and *DB320*, respectively, both with 32 gray level to build texture features. We notice that in Figure 4 the MDPCA.M2 achieve highest recognition rates or perform closer to the best one. For instance, in the Figure 4.(b), the MDPCA.M2 is better than the other ones in the intervals for  $1 \leq k \leq 5$  and  $9 \leq k \leq 12$ . Also, it is competitive with the other discriminant techniques for other intervals. Besides, the recognition rates of discriminant techniques showed in Figure 4.(b) are better than the recognition rates presented in Figure 4.(a). The Table 2 synthesizes some information presented in the Figure 4 related to maximum and minimum recognition rates.

Correlation of cropping in 32 gray level				
	cropping: 280		cropping: 320	
	Minimum	Maximum	Minimum	Maximum
MDPCA.M2	$k = 20$ (53%)	$k = 12$ (66%)	$k = 20$ (60%)	$k = 10$ (69%)
MDPCA	$k = 6$ (51%)	$k = 20$ (64%)	$d k = 16$ (58%)	$k = 8$ (68%)
MLDA	$k = 2$ (52%)	$k = 14$ (51%)	$k = 1$ (56%)	$k = 7$ (70%)
Fisher Criterion	$k = 1$ (51%)	$k = 20$ (53%)	$k = 2$ (56%)	$k = 16$ (61%)

**Table 2: Table to show the effect of cropping in recognition rates**

So, from Figure 4 and Table 2 we notice that the recognition rates for *DB320* are higher than the ones obtained when using the *DB280* database. Thus, the size of the crop window is an important parameter for the classification performance.

## 5 Conclusion and Future Works

This paper applies the Multi-Class.M2 DPCA algorithm for ranking texture features computed from multi-class granite image databases. The basic methodology has a computational complexity dominated by the AdaBoost.M2 algorithm plus texture feature computed through co-occurrence matrix and Haralick's descriptors. The granite image experiments show that, in general, the texture features selected by Multi-Class.M2 DPCA algorithm allow higher recognition rates using less features than Multi-Class LDA-DPCA, Multi-Class DPCA and Fisher Criterion. Besides, the recognition rates are very sensitive respect to the crop window size. This is an important parameter for our methodology and we must seek for nearly optimal values in further works as well as to explore deep learning methods based on convolutional neural networks (CNNs) in order to compute feature spaces components that will be processed by the discriminant techniques [16].



# Bibliography

- [1] T. Hastie, R. Tibshirani, and J. Friedman, “The elements of statistical learning,” *Springer*, 2001.
- [2] M. Petrou and P. García-Sevilla, *Image processing - dealing with texture.* Wiley, 2006.
- [3] V. Arvis, C. Debain, M. Berducat, and A. Benassi, “Generalization of the cooccurrence matrix for colour images: application to colour texture classification,” *Image Analysis & Stereology*, vol. 23, no. 1, pp. 63–72, 2011.
- [4] F. Bianconi, E. González, A. Fernández, and S. A. Saetta, “Automatic classification of granite tiles through colour and texture features,” *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11 212–11 218, Sep. 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.eswa.2012.03.052>
- [5] N. Zayed and H. A. Elnemr, “Statistical analysis of haralick texture features to discriminate lung abnormalities,” *Journal of Biomedical Imaging*, vol. 2015, pp. 12:12–12:12, Jan. 2015.
- [6] R. M. Haralick, “Statistical and structural approaches to texture,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [7] T. Filisbino, G. Giraldi, and C. Thomaz, “Ranking principal components in face spaces through adaboost.m2 linear ensemble,” in *Graphics, Patterns and Images (SIBGRAPI), 2016 26th SIBGRAPI - Conference on*, São José dos Campos, SP, Brazil, Octuber 2016.
- [8] T. A. Filisbino, G. A. Giraldi, and C. E. Thomaz, “Multi-class discriminant analysis based on linear classifiers and adaboost.m2 for ranking principal components in face spaces,” Tech. Rep., 2016. [Online]. Available: [http://www.lncc.br/departamentos/producaocientificageral.php?vMenu=2&vTipo=13&vCabecalho=pesq&vTitulo=lncc&vDepto=&idt\\_responsavel=&vAno=2016&ano=2016&anof=2016&idt\\_linha\\_pesquisa=](http://www.lncc.br/departamentos/producaocientificageral.php?vMenu=2&vTipo=13&vCabecalho=pesq&vTitulo=lncc&vDepto=&idt_responsavel=&vAno=2016&ano=2016&anof=2016&idt_linha_pesquisa=)
- [9] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [10] H. Lu, K. Plataniotis, and A. Venetsanopoulos, “MPCA: Multilinear principal component analysis of tensor objects,” *Neural Networks, IEEE Transactions on*, vol. 19, no. 1, pp. 18–39, Jan 2008.
- [11] T. Filisbino, D. Leite, G. Giraldi, and C. Thomaz, “Multi-class discriminant analysis based on svm ensembles for ranking principal components,” in *36th Ibero-Latin Am. Cong. on Comp. Meth. in Eng. (CILAMCE)*, Nov 2015.
- [12] A. Felix, M. Oliveira, A. Machado, and J. Raniery, “Using 3d texture and margin sharpness features on classification of small pulmonary nodules,” in *Graphics, Patterns and Images (SIBGRAPI), 2016 26th SIBGRAPI - Conference on*, São José dos Campos, SP, Brazil, Octuber 2016.

- [13] E. Garcia and F. Lozano, "Boosting Support Vector Machines," in *Proceedings of International Conference of Machine Learning and Data Mining (MLDM'2007)*. Leipzig, Germany: IBal publishing, Jul. 2007, pp. 153–167.
- [14] E. Yildizer, A. M. Balci, M. Hassan, and R. Alhajj, "Efficient content-based image retrieval using multiple support vector machines ensemble," *Expert Syst. Appl.*, vol. 39, pp. 2385–2396, Feb. 2012.
- [15] M. B. da Silva, "CETEM/NRES Centro de Tecnologia Mineral, Núcleo Regional do Espírito Santo."
- [16] D. Mo, "A survey on deep learning: one small step toward ai," *Dept. Computer Science, Univ. of New Mexico, USA*, 2012.

# X ENCONTRO ACADÊMICO

## MODELAGEM COMPUTACIONAL

### SIMULAÇÃO NUMÉRICA DO TRATAMENTO TÉRMICO DE AUSTÊMPERA NO FERRO FUNDIDO NODULAR

**Vinícius José de Carvalho Menezes**

**Gláucio Soares da Fonseca**

viniciusjcm@hotmail.com

glaucio@metal.eeimvr.uff.br

Universidade Federal Fluminense - EEIMVR

Av. dos Trabalhadores, 420, Vila Santa Cecília, 27255-125, Rio de Janeiro, Volta Redonda, Brasil

**Resumo.** Os ferros fundidos são ligas amplamente utilizadas na engenharia dado seu baixo custo de produção e propriedades mecânicas adequadas para diversas aplicações. Dentre os principais tipos de ferro fundido, tem-se o nodular, que apresenta grafita esferoidal gerada por tratamentos ainda em fase líquida, sendo conhecido pela excelente ductilidade e tenacidade em relação às demais ligas. Através do tratamento térmico de austêmpera, no qual se aquece o material até a austenitização com posterior resfriamento controlado, propriedades do ferro fundido nodular como resistência mecânica e ductilidade podem ser aprimoradas. Contudo, a realização do processo de austêmpera tem como consequência secundária variações dimensionais indesejáveis na peça, que se tornam um problema sempre que pequenas tolerâncias devam ser satisfeitas. Prever essas variações é, portanto, fundamental para evitar gastos desnecessários com usinagens após a realização do tratamento. Esse trabalho apresenta a reprodução e análise de um modelo de variação dimensional pelo método dos elementos finitos desenvolvido em Abaqus®, a fim de quantificar e prever as influências dos parâmetros de tratamento térmico nas dimensões finais de peças tratadas. Os resultados são posteriormente comparados e validados através de dados experimentais obtidos de amostras de ferro fundido nodular.

**Palavras-chave:** Ferro Fundido Nodular, ADI, Austêmpera, Elementos Finitos, Modelo de Variação Dimensional

## 1 INTRODUÇÃO

Os ferros fundidos, antes considerados materiais com pouca aplicabilidade na engenharia por sua fragilidade e baixas propriedades mecânicas, atualmente possuem vasto campo de utilização nos mais variados segmentos e se tornaram indispensáveis no âmbito industrial. Esse novo olhar em relação aos ferros fundidos ocorreu devido aos avanços em tecnologia e pesquisa que proporcionaram melhoria nas técnicas e métodos de obtenção e alteração da estrutura desses materiais.

Dentre os principais tipos de ferros fundidos, destaca-se o nodular, cuja relevância industrial vem de suas propriedades mecânicas vantajosas aliadas a excelente ductilidade e de sua facilidade de fundição que o tornam uma alternativa atrativa na substituição de aços forjados tradicionais. Não bastante todas as vantagens já citadas, sua resistência mecânica pode ainda ser maximizada por meio da realização do tratamento térmico conhecido como austêmpera.

Apesar dos benefícios gerados pelo tratamento térmico no ferro fundido nodular para ajuste de microestrutura final e propriedades mecânicas, a austêmpera provoca indesejável variação dimensional na peça submetida ao tratamento. A variação de dimensões lineares específicas deve ser levada em conta para garantir que a peça esteja dentro das tolerâncias requeridas em projeto. Prever a variação dimensional que a peça tratada apresentará é, portanto, fundamental para garantir a realização de usinagens antes da peça ser tratada sob melhores condições de usinabilidade, evitando assim gastos desnecessários após o tratamento. Alguns modelos foram propostos exatamente para tentar realizar essa previsão de forma qualitativa e quantitativa.

Dentre os modelos mais conhecidos, destaca-se o de Dai Pra *et. al.* (2005) que apresenta uma abordagem por redes neurais para treinamento de dados obtidos experimentalmente e correlação desses com as variações dimensionais medidas. Outro modelo mais recente que também se mostra bastante apropriado para realizar as estimativas é o de Boccardo *et. al.* (2014), que realiza o modelamento em *software* para análise pelo método dos elementos finitos baseado em problemas térmicos, mecânicos e metalúrgicos acoplados.

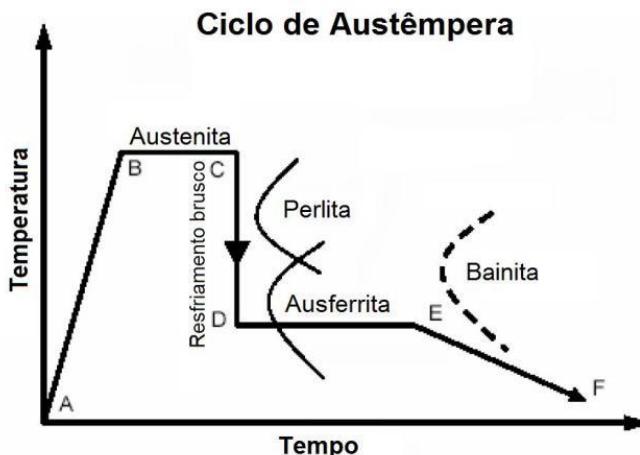
Baseado nesses estudos, este artigo apresenta um modelo termomecânico desenvolvido em *Abaqus®* para previsão das variações dimensionais longitudinais de uma amostra de ferro fundido nodular, ocasionadas por um possível tratamento térmico de austêmpera e posterior comparação dos resultados obtidos via simulação com os encontrados de forma experimental, por meio de aparelho dilatômetro e amostras ainda em estado bruto de fusão.

## 2 MATERIAIS E MÉTODOS

No estado fundido, o ferro nodular apresenta estrutura tipicamente composta de matriz perlítica e grafita esferoidal, mas também pode conter ferrita e cementita em estado livre (Chiaverini, 1990). Existem diversas aplicações para o ferro nessa condição puramente fundida, sejam elas na produção de peças ou componentes mecânicos. Contudo, nesse estado bruto, a liga pode não apresentar propriedades mecânicas satisfatórias para muitas outras aplicações nas quais deseja-se uma maior resistência ou ductilidade. Dessa forma, faz-se necessário um tratamento térmico posterior à fundição para garantir tais propriedades no material.

No tratamento térmico conhecido como austêmpera, ocorre a transferência de carbono para a matriz austenitizada pelo aquecimento entre 850 e 930 °C. Por se tratar de um tratamento isotérmico, requer um estrito e preciso controle de tempo e temperatura. O tempo de austenitização varia de duas a quatro horas, de acordo com a secção a ser tratada, para a

máxima solubilização do carbono. Já o resfriamento é interrompido para a geração de uma fase isotérmica prolongada. A Figura 1 apresenta um diagrama esquemático da austêmpera com o resfriamento e passo isotérmico definidos para o tratamento.



**Figura 1: Ciclo do tratamento térmico de austêmpera. Adaptado de Hayrynen, 2002**

Os parâmetros de tratamento térmico de austêmpera utilizados tanto na parte experimental quanto na simulação computacional do presente artigo foram os de Grau IV segundo a norma ASTM A297 e são apresentados na Tabela 1 a seguir.

**Tabela 1: Ciclo de tratamento térmico**

Parâmetros de Tratamento	Valor
Temperatura de Austenização (°C)	930
Tempo de Austenização (s)	7200
Temperatura de Austêmpera (°C)	280
Tempo de Austêmpera (s)	2400

## 2.1 Procedimento Experimental

A amostra cilíndrica de estudo do presente trabalho foi previamente cortada e usinada a partir de uma barra de ferro fundido nodular em estado bruto de fusão. As dimensões finais da amostra obtida foram de 24,64 mm de comprimento por 8,01 mm de diâmetro, medidas por micrômetro externo analógico ZAAS com resolução de 0,01 mm e exatidão de 0,003 mm.

Para a obtenção das deformações longitudinais causadas ao longo do tempo pelo tratamento térmico de austêmpera, optou-se pela simulação deste procedimento através do uso de um aparelho dilatômetro de ensaio, que aqueceria a amostra de maneira semelhante ao tratamento térmico com a vantagem de garantir o preciso controle de temperatura por termopares e a resposta no tempo de todas as deformações sofridas pela amostra.

O modelo de dilatômetro utilizado no ensaio foi o DIL 402 C, fabricado pela NETZSCH e a taxa de aquecimento foi mantida em 15 °C/min, para que não ocorresse variação suficiente de temperatura entre o termopar da amostra e o termopar do forno do equipamento, capaz de interromper o procedimento. Os demais parâmetros como temperatura e tempo de austenização foram os mesmos apresentados na Tabela 1 e a temperatura inicial escolhida de 30 °C.

Uma vez que o modelo de dilatômetro utilizado não possui sistema de resfriamento próprio, houve a impossibilidade da realização de resfriamento brusco durante o experimento e a curva de tratamento térmica foi replicada apenas até o fim da completa austenitização, não contemplando o restante do processo. Essa limitação experimental, porém, não interferiu na qualidade dos resultados já que os dados obtidos foram mais que suficientes para a posterior validação do modelo simulado.

O tempo de ensaio até a temperatura de austenitização foi de 1 hora e o tempo de ensaio total durou cerca de 3 horas.

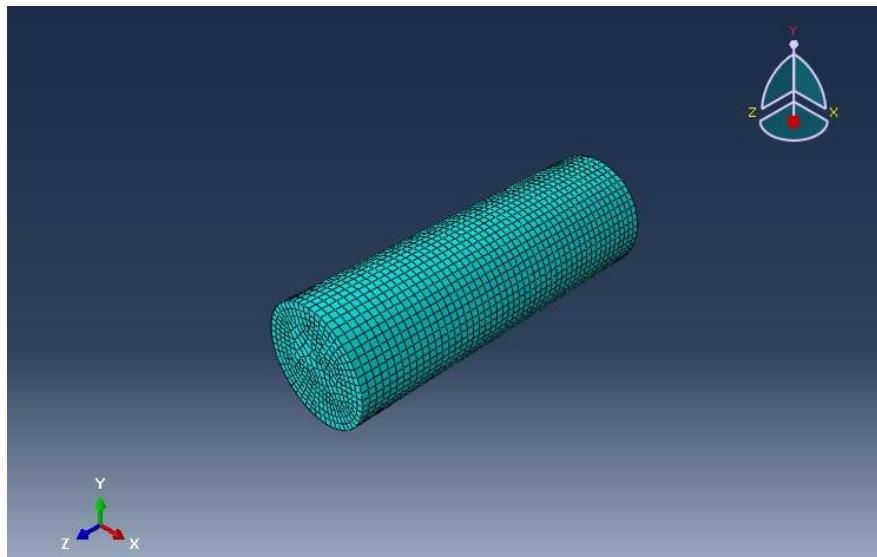
## 2.2 Modelo e Simulação

O modelamento e o desenho 3D do corpo de prova foram baseados nas mesmas dimensões e propriedades da amostra utilizada na parte experimental do trabalho, a fim de reproduzir as mesmas condições do ensaio para a correta comparação dos dados finais. Tanto as dimensões quanto as propriedades foram inseridas com suas unidades no SI, e o módulo de elasticidade, calor específico e condutividade térmica tiveram suas entradas dependentes da temperatura obtidas a partir do trabalho numérico e experimental de Dardati *et. al.* (2009) com ferro fundido nodular.

Os processos de transferência de calor por convecção e radiação foram implementados por meio de curvas de temperatura ao longo do tempo replicando as alterações de temperatura sofridas durante o tratamento térmico de austêmpera pela amostra. Estas curvas entraram como amplitudes durante a criação das interações. O passo único adotado na simulação apresentou a mesma duração do ensaio e incremento de 300 segundos.

Em relação a malha que pode ser observada na Figura 2, optou-se por geometria de malha do tipo hexagonal, interpolação linear, elemento de temperatura e deslocamentos acoplados, com controle de curvatura e máximo fator de desvio de 0,1 nas bordas e aproximadamente 0,5 mm de tamanho global, gerando um total de 15190 elementos.

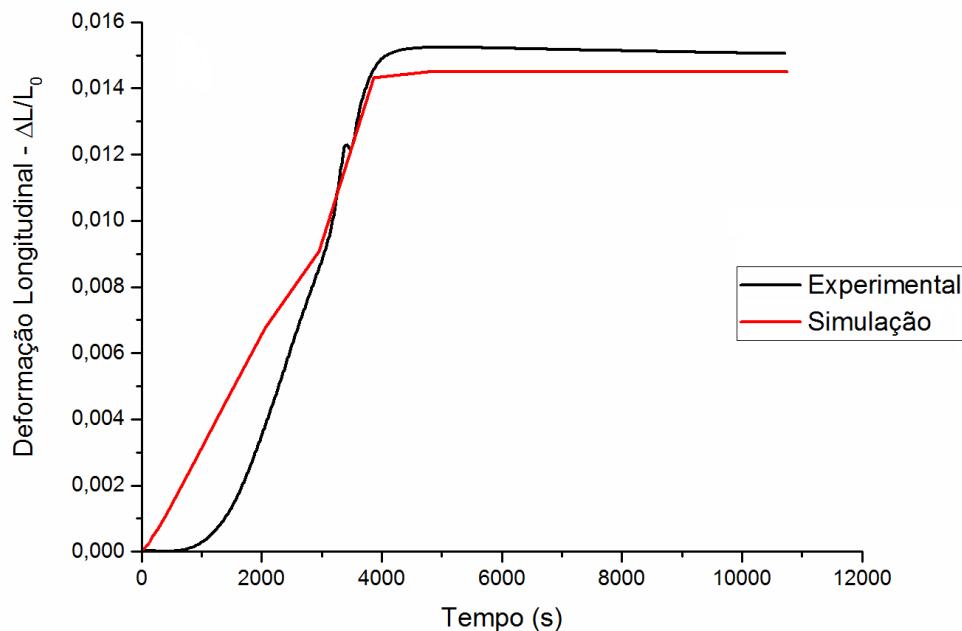
Terminada a elaboração da malha, deu-se inicio a execução da simulação com 8 núcleos de processamento simultâneos, com tempo total de 13 minutos. Todo o processo de modelamento e simulação mencionado acima foi realizado no software *Abaqus/CAE* versão 6.13-1, no *solver standard*.



**Figura 2:** Malha utilizada no modelo simulado

### 3 RESULTADOS E DISCUSSÃO

Os dados de deformação ao longo do tempo obtidos tanto na parte experimental quanto na simulação foram previamente tratados no software gráfico *OriginPro* para análise de dados e estatísticas desenvolvido pela OriginLab. O gráfico obtido após o tratamento de dados é apresentado a seguir na Figura 3, no qual a curva em preto representa os dados experimentais enquanto a resposta da simulação é dada pela curva em vermelho.



**Figura 3:** Gráfico de deformação longitudinal em função do tempo para o ferro fundido nodular obtido de forma experimental e através de modelo simulado

Analizando os resultados mostrados na Figura 3 é possível perceber um comportamento semelhante das curvas em questão. Ambas apresentam uma crescente inicial de deformação linear com a mesma praticamente se estagnando durante a parte isotérmica do processo, ao chegar aos 930 °C por volta de 3600 segundos de ensaio. O início da curva experimental apresenta um patamar horizontal não encontrado na simulação, provavelmente gerado por folgas oriundas do dilatômetro utilizado e de imperfeições na superfície da amostra.

Além disso, foi realizado o ajuste linear dos dados observados na primeira porção da curva experimental da Figura 3 utilizando o método dos mínimos quadrados. Com o auxílio da Eq. (1), obteve-se então o coeficiente de dilatação térmica linear da amostra que é apresentado juntamente com o da simulação, na Tabela 2. Os valores máximos de deformação encontrados para os dois casos também são comparados na mesma tabela.

$$\Delta L = L_i \alpha \Delta T \quad (1)$$

Onde  $\Delta L$  é a variação de comprimento,  $L_i$  é o comprimento inicial da amostra,  $\Delta T$  é a variação de temperatura e  $\alpha$  o coeficiente de dilatação térmica linear.

**Tabela 2: Comparaçao entre dados experimentais e simulados**

	<b>Experimental</b>	<b>Simulação</b>
<b>Coeficiente de dilatação térmica linear, <math>\alpha</math> (<math>^{\circ}\text{C}^{-1}</math>)</b>	$1,27 \times 10^{-5} \pm 0,01$	$1,30 \times 10^{-5}$
<b>Deformação máxima</b>	$1,52 \times 10^{-2} \pm 0,01$	$1,48 \times 10^{-2}$

Para complementar a análise dos dados, foram realizados dois cálculos de erro percentual entre as curvas tanto na porção inicial, baseado na maior diferença entre valores, quanto na porção final, que é a região de interesse para o estudo da deformação. Os resultados obtidos são apresentados na Tabela 3.

**Tabela 3: Valores de erro para deformação entre dados experimentais e simulados**

	<b>Erro (%)</b>
<b>Porção inicial das curvas</b>	26,86
<b>Porção final das curvas</b>	2,63

O erro maior obtido entre as curvas na porção inicial já era esperado devido às diferenças na superfície da amostra lixada que alteram o resultado até o completo assentamento no ensaio. Por mais que seja considerado um erro elevado, porém, pode ser desconsiderado uma vez que a parte de interesse é a final isotérmica. Essa, por sua vez, apresentou erro de apenas 2,63 % entre os valores de ensaio e os simulados.

Conforme observado na Tabela 2, os dados experimentais de dilatação térmica e deformação máxima se apresentaram próximos aos obtidos na simulação realizada, o que, juntamente com a comparação das curvas através do cálculo de erro percentual disponível na Tabela 3, possibilita validar de forma satisfatória o modelo proposto no presente trabalho.

## 4 CONCLUSÃO

O uso do dilatômetro como equipamento para procedimento experimental mostrou-se como uma alternativa eficaz em relação ao forno industrial de tratamento térmico para a aquisição dos dados de deformação, garantindo um número significativo de dados para a posterior comparação com o modelo simulado.

Em relação às variáveis de ensaio, a taxa de aquecimento escolhida foi suficiente para garantir a mínima diferença entre os termopares, possibilitando a execução completa do ensaio e a falta de um sistema de resfriamento não apresentou influência nos resultados finais.

O modelo desenvolvido representou de forma satisfatória o tratamento térmico em questão, considerando os processos de troca de calor por convecção e radiação entre o forno e a amostra, além de contar com uma malha refinada e adequada ao estudo e tempo de execução de simulação otimizado, de cerca de apenas 13 minutos.

Por fim, os dados simulados foram comparados aos experimentais e mostraram consistência, coesão e semelhança com a realidade. Não apresentaram nenhuma discrepância

evidente, o que pode garantir, de fato, a validação do modelo proposto para avaliação de deformações em peças de ferro fundido nodular austemperadas.

### **Agradecimentos**

Ao professor Gláucio Soares da Fonseca pela orientação e participação no processo de confecção do experimento. Ao empenho e dedicação dos técnicos Célio e Bonifácio, pela usinagem das amostras e ensaio no dilatômetro, respectivamente. Ao apoio financeiro e concessão de bolsa pela CAPES e à Universidade Federal Fluminense - EEIMVR pela estrutura e equipamentos que possibilitaram a realização desse artigo.

### **REFERÊNCIAS**

- Boccardo, A. D.; Sosa, A. D.; Echeverría, M. D.; Dardati, P. M.; Celentano, D. J.; Godoy, L. A., 2014. *Numerical simulation of dimensional change during austempering heat treatment in ductile cast iron. 10th International Symposium on the Science and Processing of Cast Iron - SPCI10*.
- Chiaverini, V., 1990. Aços e ferros fundidos. Sexta Edição. São Paulo: Associação Brasileira de Metalurgia e Materiais.
- Dai Pra, A. L.; Echeverría, M. D.; Scandurra, A.; Moncada, O.; Sikora, J., 2005. *Dimensional change prediction in austempered ductile iron parts using fuzzy modelling. Latin American Applied Research*. Vol. 35, p. 197-204.
- Dardati, P.M.; Celentano, D. J.; Godoy, L. A.; Chiarella, A. A. and Schulz, B. J., 2009. *Analysis of ductile cast iron solidification: numerical simulation and experimental validation*, Int. J. Cast Met. Res., 22, (5), 390–400.
- Guesser, W. L., 2009. Propriedades mecânicas dos ferros fundidos. Primeira Edição. São Paulo: Editora Blucher.
- Hayrynen, K. L., 2002. *The Production of Austempered Ductile Iron (ADI). World Conference on ADI*.
- Silva, C., 2005. Influência do número de nódulos de grafita nas propriedades mecânicas do ferro fundido nodular austemperado. Dissertação de Mestrado. Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil.