

Estimativa de Parâmetros e Solução de um Problema de Transferência de Calor Utilizando Métodos de Otimização Modificados

A. T. SILVA^{1*}, W. R. TELLES² e G. S. SEMAAN³

Recebido em 26 de junho de 2020 / Aceito em 8 de agosto de 2021

RESUMO. Neste artigo é realizada a estimativa de parâmetros e solução de um problema de transferência de calor, onde os resultados numéricos são comparados com dados experimentais obtidos na literatura. O modelo matemático é resolvido numericamente utilizando o Método das Diferenças Finitas (MDF) com formulação implícita. Já a estimativa dos parâmetros é feita utilizando os métodos de otimização estocástica Luus-Jaakola (LJ) e Algoritmo de Colisão de Partículas (do inglês, *Particle Collision Algorithm* - PCA), bem como modificações propostas nos referidos métodos. Os resultados obtidos foram satisfatórios tendo as modificações apresentado uma redução do número de avaliações da função objetivo (NAF) necessários para encontrar os parâmetros de interesse, bem como possibilitou um bom ajuste entre as temperaturas obtidas experimentalmente e resultados numéricos, motivando novas aplicações em problemas de mesma natureza.

Palavras-chave: transferência de calor, Método das Diferenças Finitas, Luus-Jaakola, algoritmo de colisão de partículas.

1 INTRODUÇÃO

De acordo com Sonntag e Borgnakke [16], problemas que envolvem transferência de calor são de grande relevância para as engenharias da computação, mecânica, química, nuclear, metalúrgica e outras. Além disso, segundo Jen et al. [5], o fenômeno de transferência de calor está presente em diversos ramos do conhecimento humano. Pode-se apontar várias situações em que o assunto deve ser considerado, tais como: microprocessadores de computadores que não podem esquentar demais para que seu funcionamento não seja afetado, geração de energia através do aquecimento

*Corresponding author: Alex Tavares Silva – E-mail: altasilva@gmail.com

¹Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Br 356, Km 3, 28300-000, Itaperuna, RJ, Brasil – E-mail: atsilva@iff.edu.br <https://orcid.org/0000-0001-7829-4203>

²Universidade Federal Fluminense, Instituto do Noroeste Fluminense de Educação Superior, Departamento de Ciências Exatas Biológicas e da Terra, Av. João Jasbick, s/n, 28470-000, Santo Antônio de Pádua, RJ, Brasil – E-mail: wtelles@id.uff.br <https://orcid.org/0000-0002-6032-3405>

³Universidade Federal Fluminense, Instituto do Noroeste Fluminense de Educação Superior, Departamento de Ciências Exatas Biológicas e da Terra, Av. João Jasbick, s/n, 28470-000, Santo Antônio de Pádua, RJ, Brasil – E-mail: gustavosemaan@id.uff.br <https://orcid.org/0000-0003-2873-2628>

de placas solares, manutenção de determinadas condições climáticas com o isolamento de tetos e paredes de casas e edifícios.

Nesse contexto, no presente trabalho atuou-se em um problema de transferência de calor por meio do uso de duas abordagens: direta e inversa. Na abordagem direta, os parâmetros de entrada e o modelo matemático são conhecidos, e o que se deseja obter é o efeito, ou seja, a temperatura espacial. Ainda em se tratando do problema direto, foi utilizado o Método das Diferenças Finitas (MDF) que, segundo Cuminato e Junior [4], é a discretização do domínio e a substituição das derivadas presentes na equação diferencial, por aproximações envolvendo somente valores numéricos da função. Já na abordagem inversa, a temperatura ao longo do tempo e espaço é conhecida, no entanto, os parâmetros de entrada ou modelo matemático não estão definidos.

No processo de modelagem do problema abordado neste artigo, uma placa de aço inox é aquecida e busca-se aferir a temperatura ao longo do tempo através de um termopar. Destaca-se que os parâmetros necessários à modelagem do experimento não estão disponíveis ou especificados de modo a ser necessário estimá-los por meio de uma abordagem baseada em problemas inversos utilizando métodos de otimização.

Em um problema real, seja de transferência de calor ou não, requer o conhecimento e informações de alguns parâmetros que nem sempre são de fácil obtenção através de experimentação. Desta forma, é utilizada como alternativa, uma abordagem inversa para o problema em questão, baseada em técnicas de otimização.

Silva Neto, Becceneri e Campos Velho [11] resolveram um problema de transferência radiativa utilizando métodos de otimização, incluindo os métodos Luus-Jaakola (LJ) e Algoritmo de Colisão de Partículas (do inglês, *Particle Collision Algorithm* - PCA).

Além disso, no que se refere ao método Luus-Jaakola, Telles et al. [18] realizaram uma modificação no mesmo com posterior aplicação na estimativa dos coeficientes de dispersão da Equação de Transporte Bidimensional, onde os resultados apresentaram melhorias em relação à versão clássica, sendo possível estimar os parâmetros de interesse com menos avaliações da função objetivo. Jezowski [6] et al. abordam um método de otimização de busca aleatória para problemas não lineares com variáveis contínuas baseado no método Luus-Jaakola. Já Luz et al. [10] apresentam um novo conceito e os resultados iniciais de uma nova metaheurística baseada na implementação de um algoritmo de otimização de funções que se baseia na colisão de múltiplas partículas.

Assim, este trabalho tem como objetivo principal, modificar e hibridizar os métodos de otimização Luus-Jaakola e o Algoritmo de Colisão de Partículas, analisando e comparando as modificações e hibridizações com as versão clássicas desses métodos, aplicadas em um problema de transferência de calor, com o intuito de estimar dois parâmetros de interesse.

O presente trabalho está dividido em seis seções, incluindo a breve introdução relatada na Seção 1. Na Seção 2 são apresentadas a descrição e o modelo matemático do problema de transferência de calor proposto. Já na Seção 3 o Método das Diferenças Finitas é introduzido e aplicado na

solução do modelo matemático. Na Seção 4, o problema é formulado através de uma abordagem inversa, em que são descritos os métodos Luus-Jaakola e Algoritmo de Colisão de Párticulas e as modificações realizadas, enquanto na Seção 5 são apresentados os resultados obtidos. Por fim, na Seção 6 é apresentada as conclusões, trabalhos futuros e destaca-se que as pesquisas relatadas apresentam-se como promissoras em problemas de transferência de calor de mesma natureza.

2 ABORDAGEM DIRETA NO PROBLEMA DE TRANSFERÊNCIA DE CALOR

No que se refere à modelagem de problemas de transferência de calor, as equações diferenciais são comumente especializadas com base nas restrições, conhecidas como problemas de valor inicial e de contorno [4]. Nesse sentido, é natural que equações diferenciais específicas sejam consideradas com base nos problemas submetidos à resolução. Por exemplo, conforme a propagação do calor em diferentes materiais, podem ser adotados modelos unidimensionais, bidimensionais ou tridimensionais.

Em particular, no problema proposto neste trabalho, o qual tem como base o experimento realizado por Carollo [3], uma amostra de aço inox modelo AISI 304, de comprimento L , foi alocada entre um aquecedor (localizado em sua parte superior) e um isolante térmico, cujo objetivo é obter a temperatura ao longo do tempo por meio de um termopar (localizado em sua parte inferior), conforme ilustra Fig. 1. Para isso, foi aplicado um fluxo de calor $\phi(t)$, na parte superior da placa, ou seja, em $x = 0$, durante 160 s, o qual foi dividido em três etapas: na primeira etapa, aplicou-se um fluxo de 2640 W/m^2 no intervalo de 0 a 20 s; na segunda etapa, no intervalo de 20 a 140 s, a intensidade do fluxo aplicada foi de 660 W/m^2 ; por fim, na terceira etapa, últimos 20 s, a fonte térmica foi desligada, resultando num fluxo de calor nulo. Devido ao isolamento no contorno da placa, considerou-se que não houve variação da temperatura na longitudinal e transversal, apenas a variação vertical, ou seja, a variação referente à sua espessura.

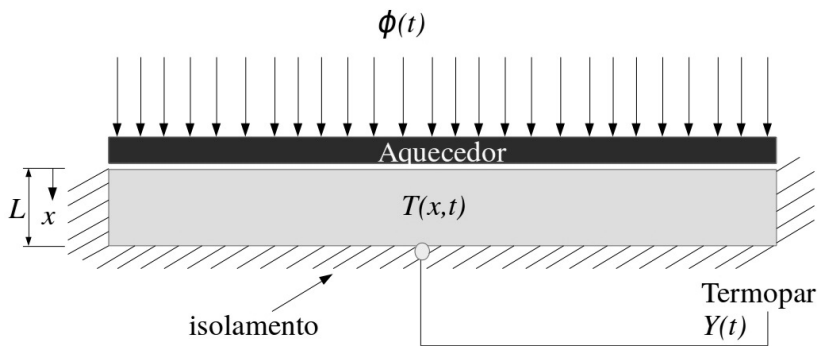


Figura 1: Representação esquemática do problema proposto. Fonte: Adaptado de Carollo (2010).

O modelo matemático unidimensional desse problema é representado pelas Eqs. (2.1) - (2.4).

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{\rho c_p}{\lambda} \frac{\partial T(x,t)}{\partial t}, \text{ em } 0 < x < L \text{ e } t > 0 \tag{2.1}$$

$$-\lambda \frac{\partial T(x,t)}{\partial x} = \phi(t), \text{ em } x = 0 \text{ e } t > 0 \tag{2.2}$$

$$\frac{\partial T(x,t)}{\partial x} = 0, \text{ em } x = L \text{ e } t > 0 \tag{2.3}$$

$$T(x,t) = T_0, \text{ em } 0 \leq x \leq L \text{ e } t = 0, \tag{2.4}$$

onde x é a coordenada cartesiana (m), t é o tempo (s), ρc_p é a capacidade de calor volumétrica (Ws/m^3K), λ é a condutividade térmica (W/mK), $\phi(t)$ é o fluxo de calor (W/m^2) imposto no instante de tempo t e T_0 é a temperatura inicial do corpo ($^{\circ}C$).

A solução desse modelo matemático é dada de forma numérica utilizando o Método das Diferenças Finitas com a formulação implícita.

3 MÉTODO DAS DIFERENÇAS FINITAS

No Método das Diferenças Finitas (MDF), deve-se discretizar o domínio de forma a encontrar soluções aproximadas para um problema. Uma alternativa pra atuar nesse sentido é gerar uma malha de nós (semelhante a uma matriz de dados) no domínio, conforme ilustrado na Fig. 2. Destaca-se que a quantidade de nós está diretamente relacionada tanto à precisão, mais próxima da solução analítica, quanto ao custo computacional (que eventualmente pode tornar inviável a cálculo de soluções).

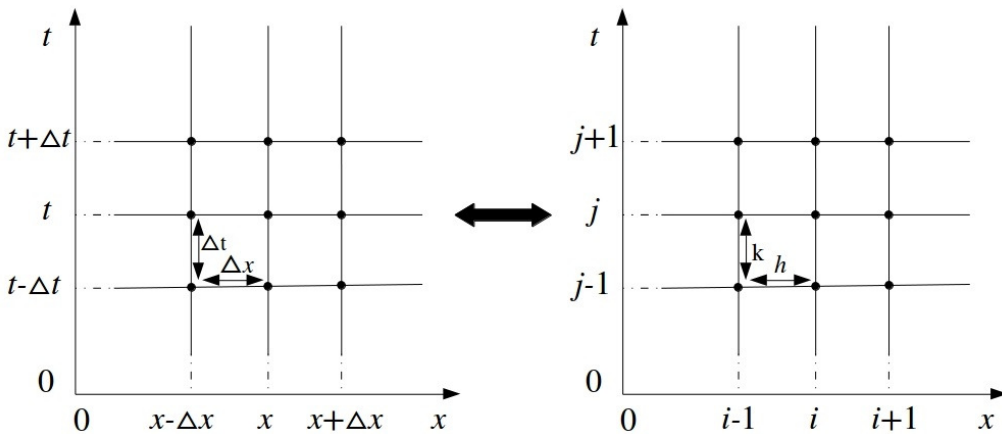


Figura 2: Malha computacional utilizada no Método das Diferenças Finitas para um domínio unidimensional. Fonte: O Autor, 2020.

Uma vez discretizado o domínio, deve-se substituir as derivadas presentes na equação diferencial por aproximações que envolvam apenas valores numéricos da função [4]. Para isso, no Método das Diferenças Finitas, duas principais abordagens podem ser exploradas para solucionar o problema: formulação explícita e formulação implícita. A formulação explícita, embora mais simples de se resolver, possui um critério que está associado à estabilidade e convergência da função, enquanto a implícita exige mais etapas na obtenção da solução do problema, devido a um sistema de equações que deve ser resolvido utilizando algum método numérico. Diante do exposto, neste trabalho é adotada a formulação implícita, a qual é descrita na Seção 3.1.

O leitor interessado em um maior aprofundamento sobre o Método das Diferenças Finitas pode consultar [2, 4, 12].

3.1 Aplicação da Formulação Implícita

Tomando como referência a derivada em relação ao tempo, a qual é de primeira ordem, e utilizando a série de Taylor para realizar uma aproximação por diferenças avançadas, cujo erro é dado por $-\frac{k}{2!}f''(\xi)$, obtém-se a Eq. (3.1).

$$\frac{\partial T(x,t)}{\partial t} \simeq \frac{T(x,t+k) - T(x,t)}{k} = \frac{T_{i,j+1} - T_{i,j}}{k} \tag{3.1}$$

Ao usar a aproximação por diferenças centradas para a derivada espacial, cujo erro é dado por $-\frac{h^2}{3!}f'''(\xi)$, e tomando como referência o instante de tempo $t+k$ (equivalente a $j+1$), obtém-se a Eq. (3.2).

$$\frac{\partial^2 T(x,t)}{\partial x^2} \simeq \frac{T(x+h,t) - 2T(x,t) + T(x-h,t)}{h^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{h^2} \tag{3.2}$$

Tomando $u_{i,j}$ como aproximação de $T_{i,j}$ e o isolando na Eq. (2.1), tem-se a expressão dada pela Eq. (3.3), com base nas Eqs. (3.1) e (3.2), $\forall i = 1, \dots, n-1$ e $j = 0, 1, \dots, m-1$.

$$\frac{k}{\beta h^2} u_{i-1,j+1} - \left(1 + \frac{2k}{\beta h^2}\right) u_{i,j+1} + \frac{k}{\beta h^2} u_{i+1,j+1} = -u_{i,j} \tag{3.3}$$

Para as condições de contorno, em $x = 0$ ($i = 0$) e $x = L$ ($i = n$), obtém-se, respectivamente, as Eqs. (3.4) e (3.5).

$$-\left(1 + \frac{2k}{\beta h^2}\right) u_{0,j+1} + \frac{2k}{\beta h^2} u_{1,j+1} = -u_{0,j} - \frac{2k}{\beta \lambda h} \phi(t) \tag{3.4}$$

$$\frac{2k}{\beta h^2} u_{n-1,j+1} - \left(1 + \frac{2k}{\beta h^2}\right) u_{n,j+1} = -u_{n,j} \tag{3.5}$$

A abordagem implícita possui um grau maior de dificuldade, uma vez que é necessário resolver o sistema que é gerado com a Eq. (3.3), juntamente com as condições de fronteira. Nesse trabalho, foi utilizado o método de Gauss-Seidel com critério de parada dado pela diferença entre duas iterações sucessivas menor ou igual a 10^{-6} para a solução numérica do referido sistema. Para maiores detalhes sobre o método de Gauss-Seidel, consultar [1].

3.2 Análise da Variação das Malhas Espacial e Temporal

Os valores dos parâmetros utilizados no problema direto foram: $\lambda = 14.611 (W/mK)$ e $\rho_{cp} = 3.907 \times 10^6 (Ws/m^3K)$. O experimento foi realizado num tempo total de 160s, aplicando um fluxo de calor $\phi(t)$, na parte superior de uma placa de aço inox AISI 304, sendo que nos primeiros 20s aplicou-se um fluxo de $2640 W/m^2$, entre 20 e 140s a intensidade do fluxo foi de $660W/m^2$ e nos últimos 20s a fonte térmica foi desligada. A temperatura inicial da placa foi de $18,84 C$ e suas dimensões foram de $49,9 \times 49,9 \times 10,9 mm$.

O problema direto foi resolvido dividindo a malha espacial em 5, 10 e 15 nós, ou seja, $\Delta x = 0,00272$, $\Delta x = 0,001209$ e $\Delta x = 0,000777$, respectivamente. Para cada malha espacial, a malha temporal foi dividida em 16, 32, 80, 160, 320, 800 e 1600 nós, obtendo-se os valores para o critério de estabilidade r , conforme Tabela 1.

Tabela 1: Discretizações obtidas tomando como base uma malha espacial contendo 5, 10 e 15 nós espaciais e tempo total de experimento igual a 160 s.

Malha Temporal (-)		$r(-)$		
$\Delta t(s)$	Número de Nós (-)	$\Delta x = 0,00272 m$	$\Delta x = 0,001209 m$	$\Delta x = 0,000777 m$
0,1	1600	0,050544	0,255877	0,619158
0,2	800	0,101087	0,511753	1,238316
0,5	320	0,252718	1,279383	3,095790
1	160	0,505435	2,558765	6,191580
2	80	1,010870	5,117530	12,383160
5	32	2,527175	12,793830	30,957900
10	16	5,054351	25,587650	61,915800

Os resultados numéricos obtidos tendo como base a formulação implícita, ao qual foi implementada na linguagem de programação C, foi satisfatório, conforme mostrado na Fig. 3.

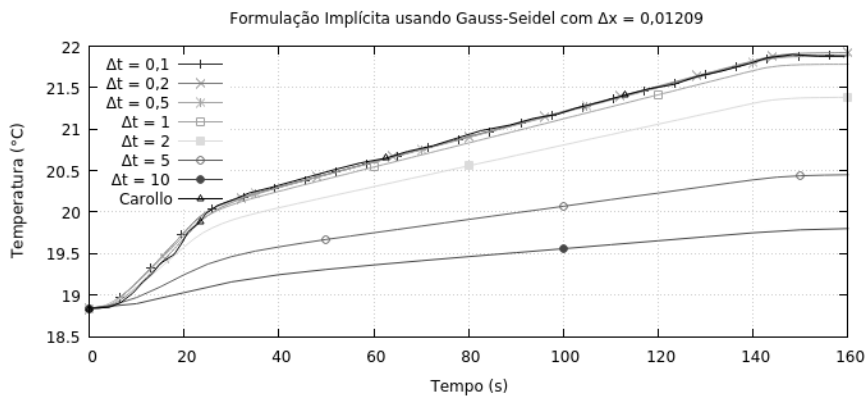


Figura 3: Resultados obtidos com a formulação implícita com $\Delta x = 0,001209m$. Fonte: O Autor, 2020.

Apesar da malha espacial com 10 nós e malha temporal com 1600 nós gerar resultados muito próximos aos dados experimentais, a malha com 800 nós, ou seja, $\Delta x = 0,001209m$ e $\Delta t = 0,2s$, obtém resultados satisfatórios, com menor custo computacional, sendo esta adotada no trabalho. Na Fig. 4 tem-se o erro absoluto entre os dados experimentais e temperaturas numéricas utilizando a referida malha. O leitor interessado em uma discussão detalhada sobre o erro de truncamento na solução do problema utilizando o Método das Diferenças Finitas pode consultar [2, 4, 14].

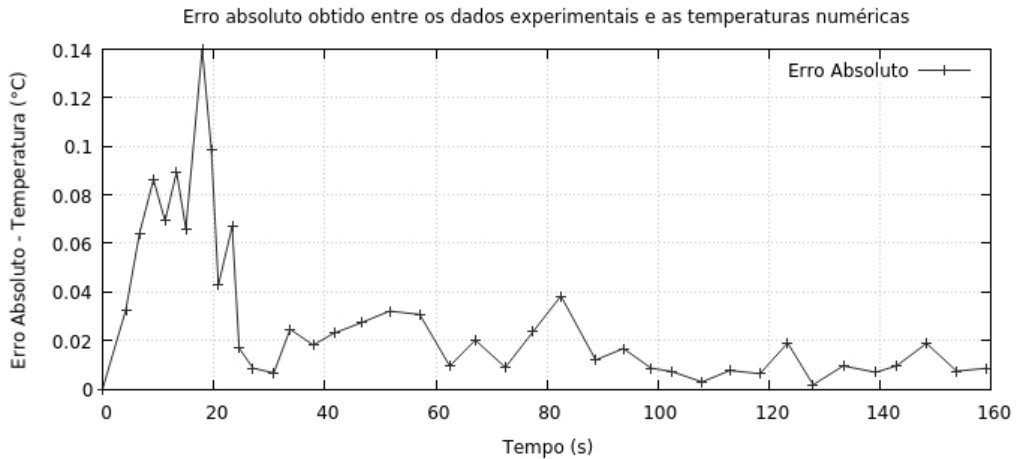


Figura 4: Erro absoluto entre os dados experimentais e as temperaturas numéricas obtidas com a formulação implícita, adotando $\Delta x = 0,01209m$ e $\Delta t = 0,2s$. Fonte: O Autor, 2020.

4 FORMULAÇÃO DO PROBLEMA INVERSO

Segundo Silva Neto, Becceneri e Campus Velho [11], existem diversas classificações para os problemas inversos e os mesmos podem ser divididos em dois grupos: estimativa de parâmetros e estimativa de funções. Quando a função objetivo é conhecida, busca-se estimar os parâmetros iniciais. Por outro lado, se os parâmetros iniciais já são conhecidos, deve-se atuar na busca por uma função adequada. No presente trabalho, a função objetivo já é conhecida, logo deseja-se estimar a condutividade térmica (λ) e a capacidade volumétrica (ρc_p) presentes no modelo matemático descrito pela Eq. (2.1), utilizando dados experimentais de temperatura para essa finalidade.

Assim, o intuito do problema inverso é minimizar a função objetivo dada pela diferença entre as temperaturas experimentais (T_{exp}), obtidas ao longo do tempo por meio de um termopar localizado na superfície inferior da placa de espessura L [3], e numéricas ($T_{num}(\lambda, \rho c_p)$), essa última obtida através da equação do calor, Eq. (2.1), juntamente com suas condições de contorno e ini-

ciais, Eq. (2.2) a (2.4), cuja solução é dada pelo Método das Diferenças Finitas com formulação implícita, conforme Seção 3. A função objetivo é dada pela Eq. (4.1):

$$f(\lambda, \rho c_p) = |\mathbf{T}_{exp} - \mathbf{T}_{num}(\lambda, \rho c_p)| = \sqrt{\sum_{i=1}^{nd} (T_{exp_i} - T_{num_i}(\lambda, \rho c_p))^2}, \quad (4.1)$$

onde T_{exp_i} e T_{num_i} são, respectivamente, as temperaturas experimentais e numéricas, obtidas em cada instante de tempo e nd é o número de dados experimentais disponíveis.

Para a estimativa das componentes do vetor de incógnitas do problema, $\mathbf{Z} = (\lambda, \rho c_p)$, as quais estão presentes na Eq. (2.1), são utilizados métodos de otimização, mais especificamente, os métodos Luus-Jaakola e Algoritmo de Colisão de Partículas, bem como algumas modificações e hibridizações nos mesmos.

4.1 Luus-Jaakola

O método Luus-Jaakola (LJ) é um procedimento estocástico de busca aleatória, onde, segundo Luus e Jaakola [8], o domínio de busca inicial é o domínio de cada variável do problema e, a cada iteração, o tamanho do espaço de busca é reduzido.

Inicialmente, deve-se determinar o fator de redução do intervalo de busca, dado pelo parâmetro ε ($0 < \varepsilon < 1$), que está relacionado com o percentual em que o intervalo é contraído. Além desse parâmetro, há o número de *loops* externos do algoritmo, n_{out} , o qual determina quantas vezes essa contração ocorrerá, e o número de estimativas geradas a cada *loop*, dado pelo número de *loops* internos, n_{in} .

Posteriormente, deve-se determinar os intervalos de busca para as variáveis que se deseja estimar, os quais dependem do problema a ser tratado, bem como a amplitude dos referidos intervalos, denotada por $\mathbf{r}^{(k)} = (r_1^{(k)}, \dots, r_n^{(k)})$, onde n é o número de variáveis, ou seja, o número da dimensão do problema.

Configurados todos os parâmetros iniciais, executa-se o algoritmo e o mesmo gera uma solução inicial, $\mathbf{x}_0^{(0)}$, tida como melhor solução para o problema até o momento, $\mathbf{x}^* = \mathbf{x}_0^{(0)}$, obtida de forma aleatória. A partir daí, são geradas novas estimativas candidatas à solução ótima do problema, sempre definidas pelo processo iterativo dos *loops* externos e internos (n_{out} e n_{in}), dada pela fórmula de recorrência: $\mathbf{x}_j^{(k)} = \mathbf{x}^* + R\mathbf{r}^{(k-1)}$, $j = 1, \dots, n_{in}$ e $k = 1, \dots, n_{out}$, sendo R é uma matriz diagonal de números aleatórios entre -0,5 e 0,5 gerada toda vez que uma nova estimativa é calculada. Caso o vetor solução $\mathbf{x}_j^{(k)}$ seja melhor que a solução \mathbf{x}^* , a nova estimativa armazenada em \mathbf{x}^* receberá o valor de $\mathbf{x}_j^{(k)}$, ou seja, $\mathbf{x}^* = (\mathbf{x})_j^{(k)}$. Ao final de cada *loop* interno, a amplitude do intervalo de busca é recalculada (reduzida) por $\mathbf{r}^{(k)} = (1 - \varepsilon)\mathbf{r}^{(k-1)}$.

Após a finalização dos *loops* externos, ou seja, quando todas as iterações forem executadas, obtém-se a solução do problema inverso, que resulta no menor valor da função objetivo (caso o problema seja de minimização). No Algoritmo 1 tem-se o pseudocódigo do método Luus-Jaakola.

Algoritmo 1 *Luus-Jaakola (LJ-Clássico)*Defina o número de *loops* externos: n_{out} Defina o número de *loops* internos: n_{in} Defina o fator de contração: ε Defina o intervalo de busca inicial: $r^{(0)}$ Gere uma solução inicial: $x_0^{(0)}$

```

1: for  $k = 1$  to  $n_{out}$  do
2:   for  $j = 1$  to  $n_{in}$  do
3:      $x_j^{(k)} = x^* + R_j^{(k)} r^{(k-1)}$ 
4:     if  $f(x_j^{(k)}) < f(x^*)$  then
5:        $x^* = x_j^{(k)}$ 
6:     end if
7:   end for
8:    $r^{(k)} = (1 - \varepsilon)r^{(k-1)}$ 
9: end for

```

4.2 Algoritmo de Colisão de Partículas

O Algoritmo de Colisão de Partículas associa-se à fenômenos físicos de absorção (*absorption*) e espalhamento (*scattering*) que ocorrem em reatores nucleares, mais precisamente, no processo de colisão das partículas nucleares, onde a partícula é espalhada por um núcleo-alvo e absorvida pelo núcleo de destino [13].

Assim como o Luus-Jaakola, no PCA se faz necessário definir os limites inferior (**min**) e superior (**max**) dos intervalos de busca para as variáveis que se deseja estimar, de acordo com o problema a ser tratado, além do número de vezes (n_{PCA}) em que ocorrerá o processo iterativo, onde novas estimativas são geradas e avaliadas de acordo com a função objetivo [17]. Posteriormente, de acordo com Lobato et al. [7] e Luz et al. [9], o PCA é inicializado com a seleção (randômica ou não) de um projeto inicial denominado **OldConfig**. Essa configuração também é considerada a melhor solução do problema até o momento, ou seja, **BestConfig** = **OldConfig**.

Em seguida, esse projeto inicial é modificado de forma estocástica por uma rotina denominada *Perturbation()* que, por sua vez, permite a obtenção de uma nova solução candidata a ótimo denominada **NewConfig**.

No entanto, se a nova solução gerada pela rotina *Perturbation()* não for melhor que a anterior, ou seja, se $f(\mathbf{NewConfig}) > f(\mathbf{OldConfig})$, então verifica-se a probabilidade da mesma ser aceita levando em consideração sua aptidão. Se a partícula atingir um núcleo com alta aptidão, a mesma é absorvida e explora-se seus limites através da rotina *Exploitation()*. Caso contrário, é espalhada para outra região através da rotina *Scattering()*. Esse processo se repete até que a solução seja

encontrada ou que o número de *loops* externos n_{PCA} seja atingido. No Algoritmo 2 é apresentada a estrutura principal Algoritmo de Colisão de Partículas.

Algoritmo 2 PCA (PCA-Clássico)

Defina o número de vezes que o PCA irá executar o processo: n_{PCA}

Defina o número de explorações ao redor da solução: n_{EXP}

Defina o número de variáveis (espaço): n

Defina os limites dos intervalos de busca: **min** e **max**

Gere um vetor com estimativas iniciais: **NewConfig**

faça

OldConfig=NewConfig

BestConfig=OldConfig

```

1: for  $k = 1$  to  $n_{PCA}$  do
2:   Perturbation()
3:   if ( $f(\text{NewConfig}) < f(\text{OldConfig})$ ) then
4:     OldConfig = NewConfig
5:     if ( $f(\text{NewConfig}) < f(\text{BestConfig})$ ) then
6:       BestConfig = NewConfig
7:     end if
8:     Exploitation()
9:   else
10:    Scattering()
11:   end if
12: end for

```

4.3 Modificações no Método Luus-Jaakola

Como na versão clássica do método Luus-Jaakola, descrita na Seção 4.1, um novo vetor $\mathbf{x}_j^{(k)}$ é gerado e compara-se $f(\mathbf{x}_j^{(k)})$ com $f(\mathbf{x}^*)$, descarta-se todos os elementos de $\mathbf{x}_j^{(k)}$ quando $f(\mathbf{x}_j^{(k)}) > f(\mathbf{x}^*)$. No entanto, pode ocorrer que o vetor $\mathbf{x}_j^{(k)}$ contenha algum componente melhor do que seu correspondente em \mathbf{x}^* . Sendo assim, nessa primeira modificação, denominada LJ-M1, é realizada a combinação entre os elementos de $\mathbf{x}_j^{(k)}$ e \mathbf{x}^* .

A alteração foi realizada após a linha 3 do Algoritmo 1, onde são realizadas 3 avaliações da função objetivo, combinando os elementos dos vetores \mathbf{x}^* e $\mathbf{x}_j^{(k)}$, de maneira a aproveitar os elementos que estejam próximos de seus "valores alvo" em algum dos dois vetores. Em outras palavras, a cada nova estimativa dos parâmetros, é feita uma combinação desta com a melhor estimativa, ou seja, se a melhor for $\mathbf{x}^*=(x_1^*, x_2^*)$ e a nova for $\mathbf{x}_j^{(k)}=(x_1, x_2)$, então, com a combinação tem-se 3 estimativas a serem avaliadas, incluindo a que acabou de ser gerada: (x_1, x_2) , (x_1^*, x_2) e (x_1, x_2^*) . Foram analisadas, ainda, outras alterações no algoritmo LJ no que se refere apenas à

redução do intervalo de busca (linha 8 do Algoritmo 1) ou seja: $\varepsilon = \frac{10^{-6}}{(max_i - min_i)} \frac{1.0}{n_{out}}$ (LJ-M2) e $\varepsilon = \frac{10^{-6}}{(max_i - min_i)} \frac{k}{n_{out}}$ (LJ-M3), onde $i = 1, \dots, n$.

Posteriormente, uma outra alteração foi realizada na estrutura do método Luus-Jaakola, fazendo com que novas estimativas sejam obtidas utilizando a mesma técnica de combinação de estimativas, porém, nessa nova versão, a redução dos intervalos é obtida conforme descrito para os métodos LJ-M2 e LJ-M3, as quais foram denominadas LJ-M4 e LJ-M5, respectivamente.

Em [15], os métodos de otimização foram modificados utilizando uma técnica de combinação de estimativas, além da função objetivo. Com essa técnica, foi possível aproveitar qualquer melhoria nos parâmetros de forma individual. Já no presente trabalho, novas modificações foram realizadas, como alteração na contração do intervalo de busca do método LJ e modificação na função *Perturbation()* do PCA, além de hibridizações entre os métodos.

4.4 Modificação no Algoritmo de Colisão de Partículas

A mesma ideia de combinação implementada no método LJ, foi implementada no PCA, denominada PCA-M1. No entanto, as três avaliações da função objetivo são realizadas durante o processo de exploração, onde, após a geração de uma nova estimativa utilizando a rotina *SmallPerturbation()* por meio da fórmula $\mathbf{NewConfig} = 0,5\{\mathbf{OldConfig} + [\mathbf{Lower} + R(\mathbf{Upper} - \mathbf{Lower})]\}$, é feita a combinação da mesma com a melhor estimativa até o momento armazenada em $\mathbf{BestConfig} = (\mathbf{BestConfig}_1, \mathbf{BestConfig}_2)$, ou seja, além de $\mathbf{NewConfig} = (\mathbf{NewConfig}_1, \mathbf{NewConfig}_2)$, são avaliadas as estimativas de forma combinada $(\mathbf{BestConfig}_1, \mathbf{NewConfig}_2)$ e $(\mathbf{NewConfig}_1, \mathbf{BestConfig}_2)$.

Foi realizado um trabalho inicial com modificações no PCA em [15] e apesar das muitas modificações realizadas, o NAF não diminuiu em relação ao presente trabalho, no entanto, hibridizações entre as modificações realizadas nos métodos, presentes em [15], trouxeram novos resultados, conforme Seção 4.5.

4.5 Hibridizações nos Métodos Luus-Jaakola e Algoritmo de Colisão de Partículas

Além das modificações descritas anteriormente, foram realizadas hibridizações entre os algoritmos LJ e PCA com o objetivo de se obter uma redução do número de avaliações da função objetivo. Tais hibridizações foram realizadas com base nas versões clássicas dos mesmos, ou seja, Algoritmo 1 e Algoritmo 2.

No método LJ foi implementada uma hibridização que se refere à perturbação realizada pelo método PCA, com o objetivo de escapar de mínimos locais com maior facilidade, uma vez que o método verifica outras alternativas para melhorar a solução antes de executar a próxima iteração. Assim, a linha 3 do Algoritmo 1 foi substituída pela função *Perturbation()*, onde, nessa última, são propostos dois processos de geração de novas estimativas, dado por $\mathbf{x}_j^{(k)} = \frac{1}{4}[3\mathbf{x}^* + (\mathbf{x}^* + R\mathbf{r}^{(k)})]$ e $\mathbf{x}_j^{(k)} = \frac{1}{4}[\mathbf{x}^* + 3(\mathbf{x}^* + R\mathbf{r}^{(k)})]$, denominados LJ-PCA-H1 e LJ-PCA-H2, respectivamente.

5 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os resultados obtidos para a estimativa da condutividade térmica, λ , e capacidade volumétrica, ρc_p , presentes na Eq. (2.1), por meio dos métodos de otimização Luus-Jaakola, Algoritmo de Colisão de Partículas e variantes de ambos. Os resultados foram gerados em um computador com processador Intel Core i5-5200U 2.2GHz, 8GB de memória RAM, sistema operacional Ubuntu GNU/Linux, sendo utilizada a linguagem de programação C e o software Gnuplot para geração dos gráficos.

5.1 Resultados dos Métodos Luus-Jaakola e Algoritmo de Colisão de Partículas nas Versões Clássicas

Para estimar os valores de λ e ρc_p do problema de transferência de calor descrito pelas Eq. (2.1) a (2.4) e resolvido numericamente pelo Método das Diferenças Finitas com a formulação implícita, adotando a malha espacial com 10 nós ($\Delta x = 0,001209m$) e malha temporal com 800 nós ($\Delta t = 0,2s$), uma vez que esta configuração apresentou melhor custo-benefício no que se refere ao tempo computacional e erro, conforme descrito na Seção 3.2. Foi utilizada uma semente inicial igual a 31 para a geração dos números aleatórios adotando a função $rand()$, presente na linguagem C, para todos os casos de forma a possibilitar uma maior igualdade na análise. No que se refere ao método LJ, foi adotado $n_{out} = 100$, $n_{in} = 50$ e $\varepsilon = 0,02$, enquanto para o método PCA utilizou-se $n_{PCA} = 100$ e $n_{EXP} = 50$. O intervalo inicial de busca para λ e ρc_p são respectivamente iguais a $[1.0, 100.0]$ (W/mK) e $[10^5, 10^7]$ (Ws/m^3K).

Além disso, os métodos foram configurados para realizarem 100 execuções com o intuito de analisar alguns parâmetros estatísticos e, em cada uma dessas execuções, foi contabilizado o número de avaliações da função objetivo (NAF), necessárias para os algoritmos encontrarem os parâmetros de interesse (λ e ρc_p) de acordo com um critério de parada estabelecido *a priori* ($f(\lambda, \rho c_p) \leq 3 \times 10^{-2}$ ou $NAF > 5000$). O número de avaliações da função objetivo indica a quantidade de novas estimativas que foram geradas ao longo do processo iterativo, dado pela relação entre loops externos e internos dos algoritmos, tendo a qualidade das mesmas (menor erro entre dados experimentais e numéricos de temperatura) verificada através da Eq. 4.1. Logo, os parâmetros estatísticos são calculados levando em consideração esse total de execuções, os quais são: melhor resultado (menor NAF), pior resultado (maior NAF), média, mediana e desvio padrão da condutividade térmica, λ , da capacidade volumétrica, ρc_p , do valor e do número de avaliações da função objetivo e do tempo computacional.

Na Tabela 2 é mostrado os resultados obtidos pelo método Luus-Jaakola (Algoritmo 1), versão clássica, onde pode-se observar que o melhor resultado foi obtido com $NAF = 227$. Cabe ressaltar que o desvio padrão para $f(\lambda, \rho c_p)$ foi de 0,000447, mostrando que os resultados não desviam muito da média. Para esta versão do algoritmo, o melhor (menor) tempo computacional necessário para estimar os parâmetros de interesse foi igual a 4543 ms. Para verificar o quão distante estão a média e o pior resultado em relação ao melhor resultado dos parâmetros λ e ρc_p , tomando como base o menor NAF, foram calculados, ainda, os respectivos erros relativos,

em módulo. Na versão clássica do LJ, os erros relativos da média para λ e ρc_p foram iguais a $0,025173 W/mK$ e $0,000063 \times 10^6 Ws/m^3K$, respectivamente, enquanto para o pior caso foram de $0,004334 W/mK$ e $0,000018 Ws/m^3K$, respectivamente.

Tabela 2: Resultados obtidos pelo método LJ na versão clássica.

-	λ (W/mK)	ρc_p (Ws/m ³ K)	$f(\lambda, \rho c_p)$ -	NAF -	Tempo (ms)
Melhor	11,843451	3862946,26667	0,029427	227	4543
Pior	11,792125	3862876,380008	0,029125	1232	24100
Média	11,545312	3862700,318705	0,029220	681	14561
Mediana	11,332384	3856706,222554	0,029200	702	14837
Desvio Padrão	0,246520	7238,788952	0,000447	148	3430

Já na Fig. 5 são apresentados os perfis da temperatura experimental e numérica para o melhor caso, pior caso e média, referentes ao método LJ na versão clássica (LJ-Clássico), bem como o perfil com os mesmos parâmetros utilizados por Carollo [3] (denominado, no gráfico, como Carollo (2010)), onde é possível verificar que os resultados numéricos estão muito próximos aos experimentais [3]. Além disso, na Fig. 6 são apresentados os erros absolutos entre a temperatura experimental e numérica para o melhor caso, pior caso, média e para os parâmetros utilizados por Carollo [3], demonstrando que os referidos erros para esses três casos são próximos entre si, o que representa uma robustez nos resultados.

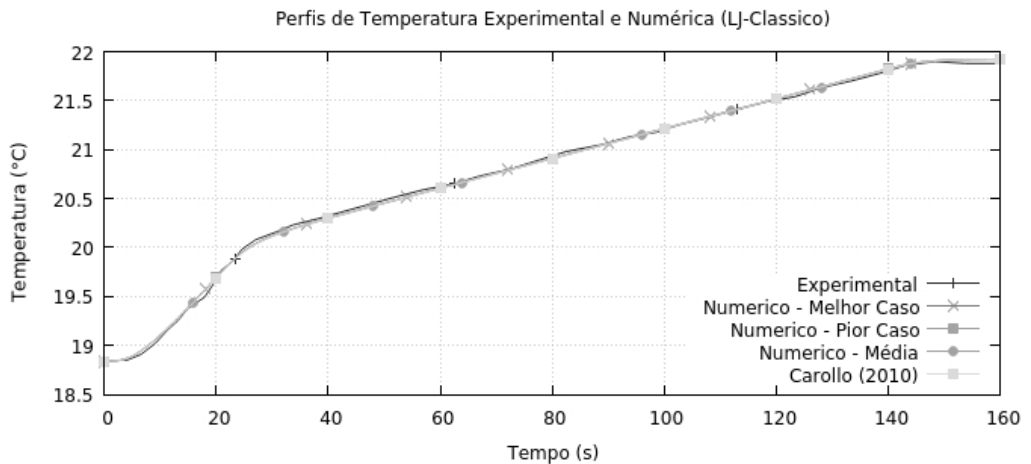


Figura 5: Perfis de temperatura obtidos com os parâmetros estimados pelo método LJ-Clássico, por Carollo [3] e dados experimentais. Fonte: O Autor, 2020.

Por outro lado, na Tabela 3 são mostrados os resultados obtidos pela versão clássica do método PCA (Algoritmo 2), sendo o número de avaliações da função objetivo igual a 85, para o melhor caso. No que se refere ao tempo computacional, a versão clássica do PCA obteve 2787

ms no melhor tempo e o desvio padrão do tempo foi igual a 5753 ms. Nessa versão clássica do PCA, os erros relativos da média para λ e ρc_p foram respectivamente iguais a 0,001988 W/mK e $0,000669 \times 10^6$ Ws/m³K e para o pior caso foram de 0,003686 W/mK e $0,001703 \times 10^6$ Ws/m³K, respectivamente, todos em relação ao melhor resultado obtido.

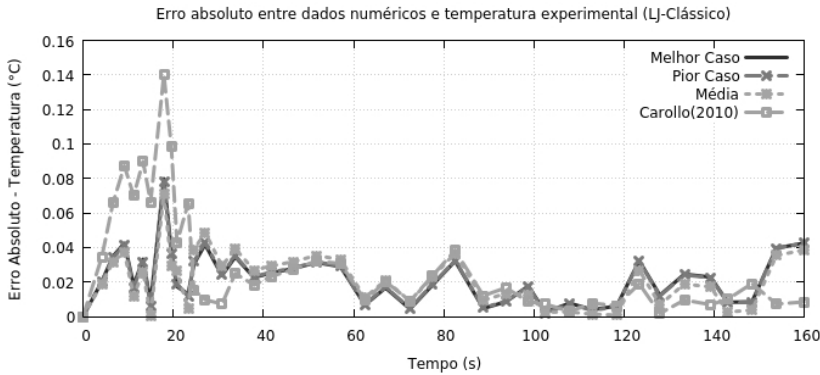


Figura 6: Erros absolutos obtidos com os parâmetros estimados pelo método LJ-Clássico e por Carollo. [3]. Fonte: O Autor, 2020.

Tabela 3: Resultados obtidos pelo método PCA na versão clássica.

-	λ (W/mK)	ρc_p (Ws/m ³ K)	$f(\lambda, \rho c_p)$ -	NAF -	Tempo (ms)
Melhor	11,62529	3867808,499111	0,028564	85	2787
Pior	11,668145	3874394,593906	0,029394	946	30242
Média	11,648396	3865222,851587	0,029230	416	12948
Mediana	11,850866	3862856,357012	0,029547	409	12524
Desvio Padrão	0,220402	6777,322211	0,000423	183	5753

Na sequência, Fig. 7, são apresentados os perfis da temperatura experimental e numérica para o melhor caso, pior caso e média, referentes ao método PCA na versão clássica (PCA-Clássico), bem como o perfil com os mesmos parâmetros utilizados por Carollo [3] (denominado Carollo (2010)), onde é possível perceber que durante o tempo de simulação igual a 160 s, as temperaturas numéricas se aproximam da temperatura experimental. Além disso, na Fig. 8, são apresentados os erros absolutos entre as temperaturas experimental e numérica para o melhor caso, pior caso, média e para os parâmetros utilizados por Carollo [3].

Com o intuito de sintetizar os resultados obtidos, na Fig. 9 é mostrado o *boxplot*, onde verifica-se que o PCA realizou um número de avaliações da função objetivo menor do que o LJ. Observa-se, ainda, que tanto o LJ quanto o PCA possuem *outliers*, que são valores considerados discrepantes em relação ao resultado. Nesse sentido, o LJ obteve o NAF igual a 227 para o melhor caso, sendo que esse valor é um *outlier*, enquanto o PCA obteve um NAF igual a 85, e esse valor não é um *outlier*. Sob esse aspecto, o PCA na versão clássica foi melhor do que o LJ na versão clássica.

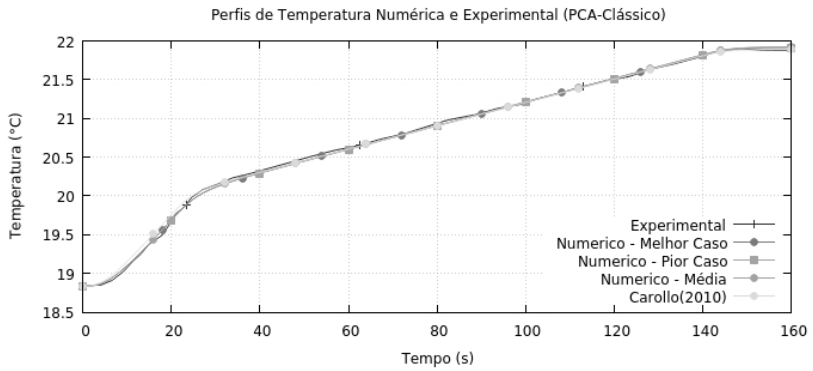


Figura 7: Perfis de temperatura obtidos com os parâmetros estimados pelo método PCA-Clássico, por Carollo [3] e dados experimentais. Fonte: O Autor, 2020.

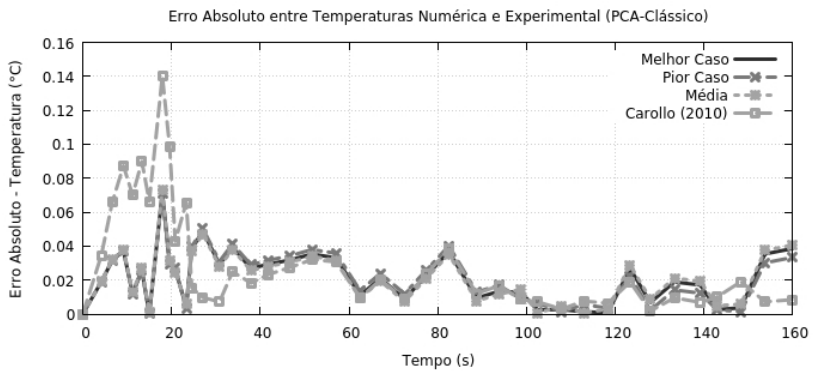


Figura 8: Erros absolutos obtidos com os parâmetros estimados pelo método PCA-Clássico e por Carollo [3]. Fonte: O Autor, 2020.

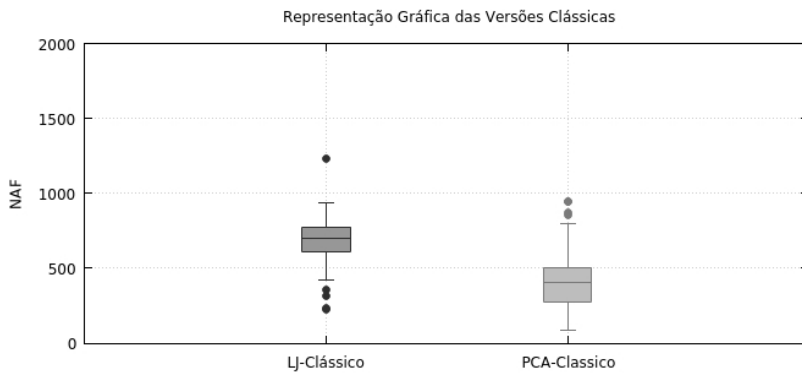


Figura 9: Distribuição de dados dos métodos LJ e PCA nas versões clássicas. Fonte: O Autor, 2020.

5.2 Resultados das Modificações nos Métodos Luus-Jaakola e Algoritmo de Colisão de Partículas

Assim como ocorrido com as versões clássicas, a condutividade térmica (λ) e a capacidade volumétrica (ρc_p) também foram estimadas utilizando as versões modificadas dos métodos LJ e PCA. Ressalta-se que, nessa seção, pelo fato dos algoritmos realizarem combinações entre estimativas, o número de *loops* internos do LJ foi configurado com $\frac{1}{3}$ da configuração original e o número de explorações do PCA também com $\frac{1}{3}$ da configuração original. Além disso, com o intuito de não tornar o texto cansativo, é apresentada apenas uma tabela contendo uma síntese geral para os melhores resultados em cada modificação descrita anteriormente.

Na Tabela 4 são apresentados os melhores resultados obtidos pelas modificações feitas nos métodos LJ e PCA, além dos melhores resultados obtidos pelas versões clássicas, onde é possível perceber que a maioria das modificações melhoraram os parâmetros e diminuíram o NAF. As modificações LJ-M2 e LJ-M3, que não obtiveram um NAF menor do que as versões clássicas, no entanto os parâmetros encontrados pelas mesmas foram muito próximos aos encontrados pelas demais modificações. Uma atenção especial deve ser dada à modificação LJ-M5, que obteve resultados promissores no que se refere à redução do NAF e dos valores dos parâmetros encontrados. Esse resultado se deve ao fato de se utilizar a técnica de combinação de resultados unida a alteração no fator de contração. No caso das modificações do método LJ, o tempo computacional também melhorou, nos casos em que o NAF foi menor do que a versão clássica, no entanto no PCA modificado (PCA-M1), o tempo computacional foi superior em relação a versão clássica. As mesma situação aconteceu para as hibridizações, onde o tempo computacional se tornou superior ao das versões clássicas. Para os casos em que o tempo computacional não foi menor do que os das versões clássicas, vale ressaltar que houve uma melhora nos resultados dos parâmetros.

Para finalizar, apresenta-se os resultados da estimativa dos parâmetros λ e ρc_p utilizando as hibridizações realizadas envolvendo os métodos LJ e PCA, nas quais foram utilizadas as mesmas configurações descritas na Seção 5.1 para as versões clássicas. Na Tabela 5 é mostrado o valor dos parâmetros de interesse, bem como os respectivos valor e número de avaliações da função objetivo e tempo computacional das simulações das versões hibridizadas, bem como das clássicas dos métodos em análise.

Com o intuito de analisar as diversas implementações propostas, na Fig. 10 é apresentado o *boxplot* dos resultados obtidos com as melhores modificações e hibridizações, onde verifica-se que as modificações LJ-M1 a LJ-M5 foram melhores do que a versão clássica do LJ, assim como ocorrido com a modificação PCA-M1, a qual foi melhor que sua versão clássica. Ressalta-se, ainda, que, apesar do número de avaliações da função objetivo nas modificações LJ-M2 e LJ-M3 não serem menores do que a versão clássica, conforme já mencionado, a cauda inferior do diagrama de distribuição da versão clássica está acima das caudas inferiores das respectivas modificações. Já as hibridizações que obtiveram melhores resultados foram as hibridizações LJ-PCA-H1 e LJ-PCA-H2.

Tabela 4: Melhores resultados obtidos com os métodos LJ e PCA modificados.

-	λ (W/mK)	ρ_{c_p} (Ws/m ³ K)	$f(\lambda, \rho_{c_p})$ -	NAF -	Tempo (ms)
LJ-Clássico	11,843451	3862946,26667	0,029427	227	4543
LJ-M1	11,348177	3862323,4761	0,028881	66	1464
LJ-M2	11,233181	3854816,478295	0,029296	344	7142
LJ-M3	11,233181	3854816,478295	0,029296	344	7618
LJ-M4	11,731186	3859365,042997	0,029284	66	1455
LJ-M5	11,340849	3863403,733211	0,028974	14	310
PCA-Clássico	11,62529	3867808,499111	0,028564	85	170
PCA-M1	11,929746	3869631,894343	0,029433	49	2585

Tabela 5: Melhores resultados obtidos com as hibridizações LJ-PCA-H1 e LJ-PCA-H2.

-	λ (W/mK)	ρ_{c_p} (Ws/m ³ K)	$f(\lambda, \rho_{c_p})$ -	NAF -	Tempo (ms)
LJ-Clássico	11,843451	3862946,26667	0,029427	227	4543
PCA-Clássico	11,62529	3867808,499111	0,028564	85	170
LJ-PCA-H1	11,62586	3865031,654401	0,028462	42	840
LJ-PCA-H2	11,278285	3852269,776411	0,029398	221	5199

Representação gráfica das melhores modificações e hibridizações nos métodos LJ e PCA

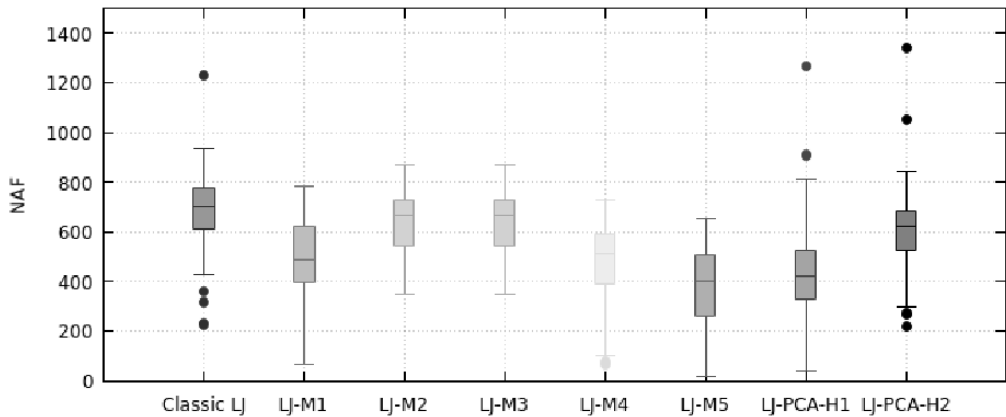


Figura 10: Distribuição de dados dos métodos LJ e PCA. Fonte: O Autor, 2020.

6 CONCLUSÕES

Este trabalho teve como objetivo principal modificar os métodos de otimização LJ e PCA, aplicando tanto as modificações quanto as versões clássicas em um problema de transferência de calor a fim de estimar parâmetros de interesse e comparar os resultados.

No problema direto de transferência de calor, na formulação implícita, obteve-se resultados numéricos compatíveis e satisfatórios, levando em consideração os dados experimentais e o baixo erro absoluto, assim como o tempo computacional quando comparado às malhas mais refinadas. Na formulação implícita foi considerada a discretização de 10 nós para malha espacial e 800 nós para malha temporal.

Já no problema inverso de transferência de calor, onde buscou-se estimar a condutividade térmica e a capacidade volumétrica utilizando os métodos de otimização Luus-Jaakola e Algoritmo de Colisão de Partículas, verificou-se que os referidos métodos, em suas versões clássicas (Algoritmos 1 e 2), obtiveram como melhor resultado os NAFs de 227 e 85, respectivamente. Além disso, algumas modificações e hibridizações foram realizadas nos referidos métodos com objetivo de melhorar tais resultados. Nas modificações do método LJ, onde o NAF reduziu, o tempo computacional também foi reduzido, no entanto, nas modificações do PCA e nas hibridizações dos métodos, o tempo computacional não reduziu, embora os valores encontrados para os parâmetros tenham sido satisfatórios. O melhor resultado, no que se refere ao NAF e aos valores dos parâmetros obtidos, foi a modificação LJ-M5, que obteve um NAF igual a 14 utilizando a técnica de combinação de estimativas com uma alteração no fator de contração e, a modificação PCA-M1 foi a que obteve, em média, menor NAF, o que demonstra uma forte tendência de melhoria no desempenho dos métodos quando se realiza a combinação de alguns métodos estocásticos. Devido a isso, pretende-se aplicar essas modificações em outras classes de problemas de transferência de calor, bem como propor novas alterações envolvendo métodos de otimização com o intuito de torná-los mais robustos.

Agradecimentos

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

ABSTRACT. In this article, the estimation of parameters and the solution of a heat transfer problem is performed, where the numerical results are compared with experimental data obtained in the literature. The mathematical model is solved numerically using the Finite Difference Method (MDF) with implicit formulation. The parameters are estimated using the Luus-Jaakola (LJ) and Particle Collision Algorithm (PCA) stochastic optimization methods, as well as proposed modifications to the referred methods. The results obtained were satisfactory and the modifications presented a reduction in the number of objective function evaluations (NAF) necessary to find the parameters of interest, as well as allowing a good adjustment between the temperatures obtained experimentally and numerical results, motivating new applications in problems of same nature.

Keywords: heat transfer, Finite Difference Methods, Luus-Jaakola, particle collision algorithm.

REFERÊNCIAS

- [1] S. Arenales & A. Darezzo. “Cálculo Numérico: Aprendizagem com o Apoio de Software”. 2 ed. Cengage Learning, Brasil (2015).
- [2] R.L. Burden, J.D. Faires & A.M. Burden. “Análise Numérica”. Trad. 10 ed. Cengage Learning (2016).
- [3] L.F.S. Carollo. “Predição de Pontos Críticos de Misturas Termodinâmicas”. Ph.D. thesis, Universidade Federal de Itajubá, Itajubá, MG (2010).
- [4] J.A. Cuminato & M.M. Júnior. “Discretização de Equações Diferenciais Parciais: Técnicas de Diferenças Finitas”. 1 ed. SBM, Brasil (2013).
- [5] L.C. Jen, G. Rissetti, A.G. Ferreira, H.A. Yamagushi & L.F. Coelho. Aplicativo visual para problemas de transferência de calor. *Directory of Open Access Journals*, **4** (2006), 385–390.
- [6] R.B.J. Jezowski & G. Ziomek. Random search optimization approach for highly multi-modal nonlinear problems. In “Advances Engineering Software”, volume 36. Elsevier (2005), p. 504–517.
- [7] F.S. Lobato, G.M. Alemida & C.F. Fernandes. Algoritmo de Colisão de Partículas Aplicado ao Projeto de Sistema de Engenharia. *Simpósio de Pesquisa Operacional e Logística da Marinha*, (2010).
- [8] R. Luus & T.H.I. Jaakola. Optimization by Direct Search and Systematic Reduction of the Size of Search Region. *American Institute Chemical Engineering*, **19** (1973), 760–766.
- [9] E.F.P. Luz, J.C. Becceneri & H.F.C. Velho. Uma nova metaheurística baseada em algoritmo de colisão de múltiplas partículas. In “IX Simpósio de Mecânica Computacional” (2008).
- [10] E.F.P. Luz, J.C. Becceneri & H.F.C. Velho. Uma nova metaheurística baseada em algoritmo de colisão de múltiplas partículas. In “Simpósio Operacional e Logística da Marinha, SPOLM”. Centro de Análises de Sistemas Navais, Rio de Janeiro, RJ (2008).
- [11] A.J.S. Neto, J.C. Becceneri & H.F.C. Velho. “Inteligência Computacional Aplicada a Problemas Inversos em Transferência Radiativa”. EdUERJ, Brasil (2016).
- [12] M.N. Ozisik, H.R.B. Orlande, M.J. Colaço & R.M. Cotta. “Finite Dierence Methods in Heat Transfer”. CRC Press, 2 ed. (2017).
- [13] W.F. Sacco & C.R.E. Oliveira. A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic. *6th World Congresses of Structural and Multidisciplinary Optimization*, (2005).
- [14] A.T. Silva, W.R. Telles & G.S. Semaan. Aplicação do Método das Diferenças Finitas em um Problema de Transferência de Calor Utilizado as Formulações Explícita e Implícita. *Anais do XII Encontro Acadêmico de Modelagem Computacional, XII EAM, LNCC, Petrópolis, RJ*, (2019).
- [15] A.T. Silva, W.R. Telles & G.S. Semaan. Avaliação dos Métodos de Otimização Luus-Jaakola e Algoritmo de Colisão de Partículas em um Problema de Transferência de Calor. *Anais do XII Encontro Acadêmico de Modelagem Computacional, XII EAM, LNCC, Petrópolis, RJ*, (2019).
- [16] R.E. Sonntag & C. Borgnakke. “Fundamentals of thermodynamics”, volume 8. Wiley New York (2013).

- [17] W.R. Telles. “Previsão do comportamento hidráulico de um rio com base na estimativa de coeficientes que controlam seu escoamento. Estudo de caso: Rio Bengalas”. Ph.D. thesis, Universidade do Estado do Rio de Janeiro, Nova Friburgo, RJ (2014).
- [18] W.R. Telles, G.I. Gonçalves, A.J.S. Neto & P.P.G.W. Rodrigues. Estimativa de parâmetros hídricos usando técnicas de problemas inversos. *Associação Brasileira de Engenharia e Ciências Mecânicas - ABCM, Universidade Estadual de Santa Cruz, BA*, (2013).

