**Aplicação de Métodos Para Estimativa de Parâmetros**

**CEFET-RJ - Unidade de Nova Friburgo**

Disciplina: Problemas Inversos em Python

Professora: Josiele da Silva Teixeira

Aluno: Raul Martins Furtado Fernandes (raul.fernandes@aluno.cefet-rj.br)

**RESUMO.** O objetivo deste trabalho foi a implementação de três rotinas computacionais para estimar o calor específico de uma placa de alumínio a partir dos dados experimentais apresentados utilizando três métodos: método da Máxima Verossimilhança, método Maximum a Posteriori e método de Monte Carlo com Cadeias de Markov implementado via algoritmo de Metropolis-Hastings. São apresentados e analisados os resultados obtidos com cada método.

# Introdução

A estimativa de parâmetros é um desafio fundamental em diversas áreas do conhecimento, incluindo estatística, inteligência artificial e aprendizado de máquina. Diferentes métodos podem ser utilizados para inferir parâmetros desconhecidos a partir de dados observados, cada um com suas vantagens e limitações dependendo do contexto e das suposições adotadas.

Neste trabalho, analisamos e comparamos três abordagens amplamente utilizadas para estimativa de parâmetros: o Método da Máxima Verossimilhança (Maximum Likelihood Estimation - MLE), o Método Maximum a Posteriori (MAP) e o Método de Monte Carlo com Cadeias de Markov (MCMC) implementado via o algoritmo de Metropolis-Hastings. O MLE busca os valores dos parâmetros que maximizam a probabilidade dos dados observados, sendo amplamente utilizado devido à sua simplicidade e eficiência em cenários onde os dados são abundantes. O método MAP, por sua vez, incorpora informações prévias na estimativa por meio de uma distribuição de probabilidade a priori, tornando-se útil em situações com dados limitados ou onde conhecimento prévio sobre os parâmetros está disponível. Já o MCMC, especificamente através do algoritmo de Metropolis-Hastings, permite amostrar distribuições complexas e realizar inferências probabilísticas em espaços de alta dimensão, sendo uma ferramenta poderosa para problemas onde as abordagens determinísticas são inviáveis.

O objetivo deste estudo é analisar o desempenho desses métodos em diferentes cenários, avaliando suas vantagens e limitações em termos de precisão, robustez e aplicabilidade. Para isso, implementamos cada técnica e realizamos experimentos comparativos, discutindo os resultados obtidos e suas implicações na escolha do método mais adequado para cada contexto.

# Método da Máxima Verossimilhança

A partir dos dados experimentais fornecidos, foi implementada uma função objetivo que calcula o custo associado ao calor específico da placa de alumínio (*cp*). O custo reflete a discrepância entre os valores de temperatura calculados pela função teórica e os valores observados experimentalmente para diferentes instantes de tempo (*t*). Assim, a função objetivo busca quantificar a precisão do *cp* proposto, permitindo a identificação do valor mais adequado por meio de métodos de otimização.

# Método Maximum a Posteriori

Assim como no Método da Máxima Verossimilhança, foi implementada uma função objetivo a partir dos dados experimentais fornecidos. Essa função objetivo é similar à anterior, porém ela leva em consideração a informação *a priori* para o parâmetro *cp* que foi dada no problema, onde *cp* segue uma distribuição Normal com média *917* *J/(kgK)* e desvio padrão *σcp*, considerando 2 casos: *σcp = 10* e *σcp = 0.5*. Assim, a função objetivo busca quantificar a precisão do *cp* proposto com base em um conhecimento prévio, o que deixa o *cp* estimado mais próximo do valor real.

# Método de Monte Carlo com Cadeias de Markov via algoritmo de Metropolis-Hastings

Ao contrário dos métodos anteriores, o Método de Monte Carlo com Cadeias de Markov não estima somente 1 valor para *cp* para depois ser otimizado por funções de otimização. A ideia central do MCMC implementado via algoritmo de Metropolis-Hastings é construir uma cadeia de Markov cujos elementos, ao longo do tempo, convergem para um valor de *cp* cuja média é o *cp* estimado.

Para montar essa cadeia é utilizada uma função objetivo *pPost* que calcula o custo associado ao *cp* calculado baseado no Método da Máxima Verossimilhança e na informação *a priori* para o parâmetro cp dada no problema.

Este método recebe como parâmetros os limites mínimo e máximo para *cp* (*bounds*), o tamanho da cadeia de Markov gerada (*n*) e o desvio padrão dos valores gerados (*sig*).

# Descrição do Algoritmo

1. Inicialização
   1. A variável *cadeia* é criada como um vetor inicialmente vazio;
   2. Um valor de *cp* é gerado aleatoriamente dentro dos limites (*bounds*) e é adicionado na *cadeia*;
   3. O algoritmo realiza *n* iterações e, em cada uma, acrescenta um elemento na *cadeia*;
2. Iterações
   1. Um novo valor candidato para *cp* é gerado a partir do elemento anterior da cadeia seguindo uma distribuição normal com desvio padrão (*sig*);
   2. A variável *alfa* é calculada a partir da seguinte formula matemática: *exp(* *pPost(i) – pPost(i – 1) )*;
   3. É gerado um valor aleatório *u* entre 0 e 1;
   4. Caso *alfa* seja maior ou igual a *u*, o valor candidato é adicionado na *cadeia*, caso contrário, o valor anterior é repetido;
3. Fim
   1. Ao fim de todas as iterações, a variável *cadeia* irá conter todos os elementos aceitos cujos valores começar a convergir em torno da média com um desvio cada vez menor.

# ANÁLISE DOS RESULTADOS

Com os métodos implementados, foram realizadas análises para identificar como os parâmetros de cada método influenciam a estimativa do calor específico da placa de alumínio.

# Método da Máxima Verossimilhança

Após diversas execuções do método Luus-Jaakola (*nInt=200, nOut=200, coef=0.2*) com a função objetivo da Máxima Verossimilhança, observou-se que o valor estimado de *cp* foi *918,39325 J/(kgK)*, variando somente a partir da sexta casa decimal.

# Método Maximum a Posteriori

Ao contrário do método anterior, este método considera a informação *a priori* para estimar o valor de *cp*. Sendo assim, considerando o desvio padão da informação *a priori* como *10* e *0,5*, os valores estimados para *cp* após diversas execuções do método Luus-Jaakola (*nInt=200, nOut=200, coef=0.2*) com a função objetivo Maximum a Posteriori foram, respectivamente, *919,10137* e *918,79262* *J/(kgK)*. Esse resultado mostra que ao considerar um desvio padrão mais alto, o *cp* estimado está mais distante do valor real.

# Método de Monte Carlo com Cadeias de Markov via algoritmo de Metropolis-Hastings

O método de MCMC foi testado diversas vezes com diferentes valores para *sig*. Para cada valor de *sig* foi gerado um gráfico com 10 execuções do método. Esses gráficos estão representados nas figuras a seguir.

A graph of different colored lines

AI-generated content may be incorrect.

Figura 1 - Resultados MCMC (sig = 0.1%)

A graph of colored lines

AI-generated content may be incorrect.

Figura 2 - Convergência MCMC (sig = 0.1%)

A graph of a line graph

AI-generated content may be incorrect.

Figura 3 - Resultados MCMC (sig = 0.2%)

A graph of colored lines

AI-generated content may be incorrect.

Figura 4 - Convergência MCMC (sig = 0.2%)

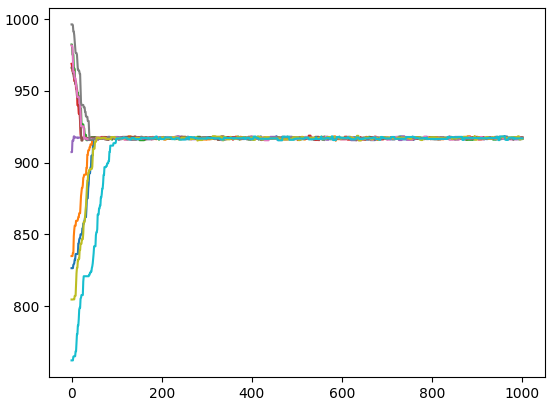


Figura 5 - Resultados MCMC (sig = 0.5%)

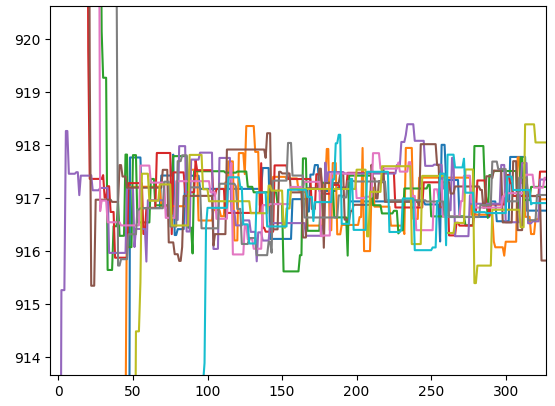


Figura 6 - Convergência MCMC (sig = 0.5%)

A graph of a line

AI-generated content may be incorrect.

Figura 7 - Resultados MCMC (sig = 1%)

A graph of colored lines

AI-generated content may be incorrect.

Figura 8 - Resultados MCMC (sig = 1%)

Ao observar os resultados do MCMC podemos perceber que valores mais baixos para o desvio padrão dos elementos candidatos demoram mais para convergir, enquanto valores mais altos convergem mais rápido. Entretanto, o valor de 1% é o que traz a convergência em menos iterações. Valores mais altos que 1% não trazem resultados melhores do que esse observado.

Já o valor médio de *cp* após a convergência não é afetado pelo valor de *sig*. Em todos os casos a média pós convergência é de 917, o que muda é o tamanho da cadeia necessário para os valores convergirem com o 917.

Todos os gráficos acima foram gerados considerando o desvio padrão da informação *a priori* como 0,5. Pode-se observar que, após a convergência, os valores variam entre 916 e 918. Já para o desvio padrão da informação *a priori* como 10, pode-se observar na Figura 9 que os dados estão mais distantes do 917 após a convergência.

A graph of a graph

AI-generated content may be incorrect.

Figura 9 - Resultados MCMC (sig = 1%, sig a priori = 10)

# CONCLUSÃO

Neste trabalho, foram analisados três métodos estatísticos para a estimativa do calor específico *cp*: Máxima Verossimilhança, Maximum a Posteriori e Monte Carlo com Cadeias de Markov via algoritmo de Metropolis-Hastings. Os resultados mostraram que cada método apresenta características distintas em termos de precisão, robustez e dependência de informações prévias.

O método da Máxima Verossimilhança apresentou estimativas consistentes, com pouca variação nos valores obtidos, reforçando sua eficácia quando não há informações *a priori* disponíveis. No entanto, o método Maximum a Posteriori, ao incorporar informações prévias, mostrou que o desvio padrão dessas informações influencia significativamente a estimativa final. Um desvio padrão maior na distribuição a priori resultou em valores mais afastados do real, evidenciando a sensibilidade do método à qualidade da informação prévia.

Por fim, o método MCMC via Metropolis-Hastings demonstrou que o tempo de convergência depende fortemente da escolha do desvio padrão dos elementos candidatos. Valores mais baixos aumentam o tempo de convergência, enquanto valores adequados, como 1%, proporcionam um equilíbrio entre eficiência e precisão. Apesar das variações no processo de amostragem, o valor médio final de *cp* manteve-se próximo de *917 J/(kgK)*, independentemente do desvio padrão dos elementos candidatos. Já o desvio padrão da informação *a priori* influencia diretamente na assertividade dos elementos da cadeia após a convergência, evidenciando também a sensibilidade do método à qualidade da informação prévia.

Dessa forma, conclui-se que a escolha do método mais adequado depende do contexto do problema e da disponibilidade de informações *a priori*. O método da Máxima Verossimilhança é uma boa escolha quando não há conhecimento prévio confiável, enquanto o MAP permite incorporar informações adicionais, desde que bem fundamentadas. O MCMC, por sua vez, oferece maior flexibilidade para inferências bayesianas.

# Apêndice 1 – Implementação do cálculo da temperatura

A screenshot of a computer program

AI-generated content may be incorrect.

# Apêndice 2 – Implementação da informação *a priori*

A black background with white text

AI-generated content may be incorrect.

# Apêndice 3 – Implementação da função objetivo da Máxima Verossimilhança

A screen shot of a computer code

AI-generated content may be incorrect.

# Apêndice 4 – Implementação da função objetivo da Maximum a Posteriori



# Apêndice 5 – Implementação da função *pPost* para o MCMC



# Apêndice 6 – Implementação do MCMC

A screen shot of a computer program

AI-generated content may be incorrect.

# Apêndice 7 – Implementação do arquivo main.py

A computer screen shot of a program

AI-generated content may be incorrect.

# Apêndice 8 – Dados experimentais

A screen shot of a computer screen

Description automatically generated