

Universidade Estadual Paulista Julio de Mesquita Filho - UNESP  
Instituto de Biociências  
Departamento de Biodiversidade e Bioestatística  
Programa de Pós-Graduação em Biometria  
Métodos Numéricos e Computacionais

## **Trabalho Computacional 3 - Método dos Mínimos Quadrados para Sistemas Lineares**

Aluno(a): Richard Castro Júnior

Professor: Daniela Renata Cantane

Botucatu-SP

2022

# 1 Introdução

A teoria da aproximação envolve dois tipos gerais de problemas: ou desejamos encontrar uma função "mais simples" (um polinômio, por exemplo), dado uma função explicitamente, a fim de aproximar os valores desta função, ou ajustar funções a dados para encontrar a "melhor" função em uma determinada classe para representar os dados (BURDEN; FAIRES, 2010). Podemos supor um conjunto de medidas de uma mesma quantidade física, sujeitas apenas a erros aleatórios; logo, o valor mais provável da quantidade medida é aquele que torna a soma dos quadrados dos erros um mínimo.

Uma solução de mínimos quadrados para um sistema linear  $Ax = b$  é um vetor  $x^* \in \mathbb{R}$  que minimiza a norma quadrada  $\|Ax - b\|^2$ . O conceito de solução de mínimos quadrados é para sistemas não têm solução, ou seja,  $b$  não está na imagem de  $A$ , cuja solução de mínimos quadrados seja única. Isso acontece se, e somente se,  $\ker A = 0$  (as colunas de  $A$  são linearmente independentes; posto  $A = n$ ). Se  $z \in \ker A$ , então  $\tilde{x} = x + z$  também satisfaz (OLVER; SHAKIBAN, 2018)

$$\|A\tilde{x} - b\|^2 = \|A(x + z) - b\|^2 = \|Ax - b\|^2,$$

e, portanto, também é um mínimo. Assim, a unicidade requer  $z = 0$ .

Identificando o subespaço  $W = \text{img} A \in \mathbb{R}^m$  como a imagem ou espaço coluna da matriz  $A$ , se as colunas de  $A$  são linearmente independentes, elas formam uma base para a imagem  $W$ . Como todo elemento da imagem pode ser escrito como  $w = Ax$ , minimizar  $\|Ax - b\|^2$  é o mesmo que minimizar a distância  $\|w - b\|$  entre o ponto  $w$  e o subespaço. A solução  $x^*$  para o problema de minimização quadrática produz o ponto mais próximo  $w^* = Ax^*$  em  $W = \text{img} A$  (OLVER; SHAKIBAN, 2018).

Deste modo, assumindo  $\ker A = 0$ , a solução de mínimos quadrados para o sistema linear  $Ax = b$  sob a norma euclidiana é a única solução  $x^*$  para as equações normais (OLVER; SHAKIBAN, 2018)

$$(A^T A)x = A^T b, \quad \text{ou seja,} \quad x^* = (A^T A)^{-1} A^T b$$

O erro de mínimos quadrados é

$$\|Ax^* - b\|^2 = \|b\|^2 - f^T x^* = \|b\|^2 - b^T A(A^T A)^{-1} A^T b$$

## 1.1 Caso Discreto

Seja  $a_1x_i + a_0$  o valor  $i^{th}$  na linha de aproximação e  $y_i$  o  $i^{th}$  dado valor  $y$ . Assumimos que as variáveis independentes, as  $x_i$ , são exatas, são as variáveis dependentes, as  $y_i$ , que são suspeitas. Esta é uma suposição razoável na maioria das situações experimentais (BURDEN; FAIRES, 2010).

O problema de encontrar a equação da melhor aproximação linear no sentido absoluto requer que os valores de  $a_0$  e  $a_1$  sejam encontrados para minimizar (BURDEN; FAIRES, 2010)

$$E_{\infty}(a_0, a_1) = \max \{|y_i - (a_1x_i + a_0)|\}.$$

Isso é comumente chamado de problema *minimax* e não pode ser tratado por técnicas elementares.

Outra abordagem para determinar a melhor aproximação linear envolve encontrar valores de  $a_0$  e  $a_1$  para minimizar (BURDEN; FAIRES, 2010)

$$E_1(a_0, a_1) = \sum_{i=1}^m |y_i - (a_1x_i + a_0)|.$$

Essa quantidade é chamada de desvio absoluto. Para minimizar uma função de duas variáveis, precisamos definir suas derivadas parciais para zero e resolver simultaneamente as equações resultantes. No caso do desvio absoluto, precisamos encontrar  $a_0$  e  $a_1$  com (BURDEN; FAIRES, 2010)

$$0 = \frac{\partial}{\partial a_0} \sum_{i=1}^m |y_i - (a_1x_i + a_0)| \quad \text{e} \quad 0 = \frac{\partial}{\partial a_1} \sum_{i=1}^m |y_i - (a_1x_i + a_0)|.$$

O problema é que a função de valor absoluto não é diferenciável em zero, e talvez não possamos encontrar soluções para esse par de equações.

## 1.2 Caso Contínuo

Seja uma função  $f$  contínua num intervalo  $[a, b]$ . Esta função pode ser aproximada por uma combinação de funções

$$g(x) = \beta_1 g_1(x) + \beta_2 g_2(x) + \cdots + \beta_n g_n(x),$$

em que  $g_1(x), \dots, g_n(x)$  são funções contínuas em  $[a, b]$ .

Neste caso, queremos minimizar a área entre as curvas de  $f$  e  $g$ . Então, objetivando encontrar  $g(x)$  que minimize

$$\phi(\beta_1, \beta_2, \dots, \beta_n) = \int_a^b (f(x) - (\beta_1 g_1(x) + \dots + \beta_n g_n(x)))^2 dx$$

deveremos encontrar  $\beta_1, \beta_2, \dots, \beta_n$ . Isto é feito determinando-se os pontos estacionários de  $\phi$ , ou seja, derivando-a parcialmente com respeito a  $\beta_1, \beta_2, \dots, \beta_n$  e igualando-se a 0.

Assim,

$$\frac{\partial \phi}{\partial \beta_i} = -2 \cdot \int_a^b ((f(x) - (\beta_1 g_1(x) + \dots + \beta_n g_n(x))) g_i(x)) dx = 0,$$

para  $i=1, \dots, n$ .

Agora, seja  $\langle f, g \rangle = \int_a^b (f(x)g(x))dx$ , então

$$\underbrace{\begin{bmatrix} \langle g_1, g_1 \rangle & \langle g_1, g_2 \rangle & \cdots & \langle g_1, g_n \rangle \\ \langle g_2, g_1 \rangle & \langle g_2, g_2 \rangle & \cdots & \langle g_2, g_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle g_n, g_1 \rangle & \langle g_n, g_2 \rangle & \cdots & \langle g_n, g_n \rangle \end{bmatrix}}_A \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \langle g_1, f \rangle \\ \langle g_2, f \rangle \\ \vdots \\ \langle g_n, f \rangle \end{bmatrix}.$$

Se  $\det(A) \neq 0$ , então o sistema possui solução única, ou seja, existe uma única função  $g$  que melhor se aproxima de  $f$ . Além disso, se  $g_i, i = 1, \dots, n$  forem linearmente independentes, então  $\det(A) \neq 0$ .

### 1.2.1 Conexão com mínimos quadrados discretos

As equações normais  $\mathbf{A}^* \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{y}$  formadas por polinômios a aproximação por mínimos quadrados discretos converge exatamente para o mesmo sistema  $(n+1) \times (n+1)$   $\mathbf{H} \mathbf{c} = \mathbf{b}$ , conforme derivamos, de maneira independente à aproximação polinomial por mínimos quadrados contínuos (LECTURE... 2009).

No caso discreto, procuramos minimizar  $\|\mathbf{c} - \mathbf{A} \mathbf{y}\|_2$ , onde (LECTURE... 2009)

$$\mathbf{A} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{bmatrix}$$

Esse problema discreto pode ser resolvido por meio das equações normais (LECTURE... 2009):

$$\mathbf{A}^T \mathbf{A} \mathbf{c} = \mathbf{A}^T \mathbf{y}.$$

sendo

$$\mathbf{A}^T \mathbf{y} = \begin{bmatrix} \sum_{k=0}^n f(x_k) \\ \sum_{k=0}^n x_k f(x_k) \\ \sum_{k=0}^n x_k^2 f(x_k) \\ \vdots \\ \sum_{k=0}^n x_k^n f(x_k) \end{bmatrix} \in \mathbb{C}^{n+1}$$

Observe que se os pontos de aproximação  $m + 1$  estiverem espaçados uniformemente sobre  $[a, b]$ ,  $x_k = a + kh_m$  para  $h_m = (b - a)/m$ , temos

$$\lim_{m \rightarrow \infty} h_m \mathbf{A}^T \mathbf{y} = \begin{bmatrix} \int_a^b f(x) dx \\ \int_a^b x f(x) dx \\ \int_a^b x^2 f(x) dx \\ \vdots \\ \int_a^b x^n f(x) dx \end{bmatrix} = \begin{bmatrix} \langle f, 1 \rangle \\ \langle f, x \rangle \\ \langle f, x^2 \rangle \\ \vdots \\ \langle f, x^n \rangle \end{bmatrix}$$

que é precisamente o vetor do lado direito  $\mathbf{b} \in \mathbb{C}^{n+1}$  obtido para o problema de mínimos quadrados contínuos (LECTURE... 2009).

Da mesma forma, a entrada  $(j+1, k+1)$  da matriz  $\mathbf{A}^T \mathbf{A} \in \mathbb{C}^{(n+1) \times (n+1)}$  para o problema discreto pode ser formado como (LECTURE... 2009)

$$(\mathbf{A}^T \mathbf{A})_{j+1, k+1} = \sum_{\ell=0}^n x_{\ell}^j x_{\ell}^k = \sum_{\ell=0}^n x_{\ell}^{j+k}.$$

Assim, temos no limite (LECTURE... 2009):

$$\lim_{m \rightarrow \infty} h_m (\mathbf{A}^T \mathbf{A})_{j+1, k+1} = \int_a^b x^{j+k} dx = \langle x^j, x^k \rangle.$$

E no agregado:

$$\lim_{m \rightarrow \infty} h_m \mathbf{A}^T \mathbf{A} = \mathbf{H},$$

onde  $\mathbf{H}$  é a matriz no problema de mínimos quadrados contínuos (LECTURE... 2009).

### 1.3 Caso Não Linear

Em alguns casos, o método dos quadrados mínimos linear pode ser usado para ajustar uma função não linear nos coeficientes. Neste caso, o ajuste dos dados como combinação linear dos parâmetros não é adequado.

Para se aplicar o método dos quadrados mínimos, é necessário que se efetue uma linearização do problema através de alguma transformação conveniente. Sendo o método dos quadrados mínimos é aplicado na resolução do problema linearizado, obtém-se os parâmetros do problema em questão a fim de usá-los para calcular os parâmetros originais. Todavia, os parâmetros obtidos não são ótimos dentro do critério dos quadrados mínimos, já está sendo ajustando o problema linearizado por quadrados mínimos e não o problema original (AJUSTE... 2022).

Podemos usar, por exemplo, o ajuste hiperbólico. A representação dos dados possui um comportamento do tipo

$$g(x) = \frac{1}{\beta_1 x + \beta_2}.$$

O sistema de equações não lineares é resolvido fazendo uma transformação. Se  $g(x) = \frac{1}{\beta_1 x + \beta_2}$  aproxima  $f(x)$ , consideramos  $h(x) = \frac{1}{g(x)} = \beta_1 x + \beta_2$  que aproxima  $\frac{1}{f(x)}$  e temos agora o caso linear. Assim,  $g(x) \approx f(x) \Rightarrow \frac{1}{g(x)} \approx \frac{1}{f(x)}$ .

Resolvemos o problema do seguinte modo:

$$\text{minimizar } \sum_{i=1}^n \left( \frac{1}{f(x_i)} - h(x_i) \right)^2$$

e obtemos uma solução aproximada para o problema original.

Neste caso, o sistema é dado por:

$$\begin{cases} \left( \sum_{i=1}^n x_i^2 \right) \beta_1 + \left( \sum_{i=1}^n x_i \right) \beta_2 = \sum_{i=1}^n \frac{x_i}{y_i} \\ \left( \sum_{i=1}^n x_i \right) \beta_1 + n \beta_2 = \sum_{i=1}^n \frac{1}{y_i} \end{cases}$$

Além do ajuste hiperbólico, podemos usar também o ajuste exponencial cuja representação dos dados possui um comportamento do tipo

$$g(x) = a(b)^x,$$

com  $a$  e  $b$  positivos.

Neste caso, usamos a transformação

$$h(x) = \ln(g(x)) = \ln(a(b)^x) = \ln a + x \ln b.$$

Definindo-se  $\beta_1 = \ln(b)$  e  $\beta_2 = \ln(a)$ , então  $a = e^{\beta_2}$  e  $b = e^{\beta_1}$ . Portanto,  $h(x) = \beta_1 g_1(x) + \beta_2 g_2(x)$ , em que  $g_1(x) = x$  e  $g_2(x) = 1$ .

Assim,  $g(x) \approx f(x) \Rightarrow h(x) = \ln(g(x)) \approx \ln(f(x))$  e resolvemos o problema da seguinte forma:

$$\text{minimizar } \sum_{i=1}^n (\ln(f(x_i)) - h(x_i))^2$$

obtendo uma solução aproximada para o problema original.

Neste caso, o sistema é dado por:

$$\begin{cases} \left( \sum_{i=1}^n x_i^2 \right) \beta_1 + \left( \sum_{i=1}^n x_i \right) \beta_2 = \sum_{i=1}^n \ln(y_i) x_i \\ \left( \sum_{i=1}^n x_i \right) \beta_1 + n \beta_2 = \sum_{i=1}^n \ln y_i \end{cases}$$

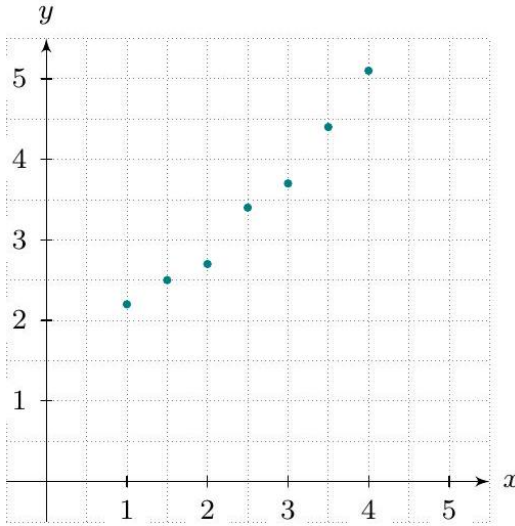


Figura 1: Pontos a serem ajustados

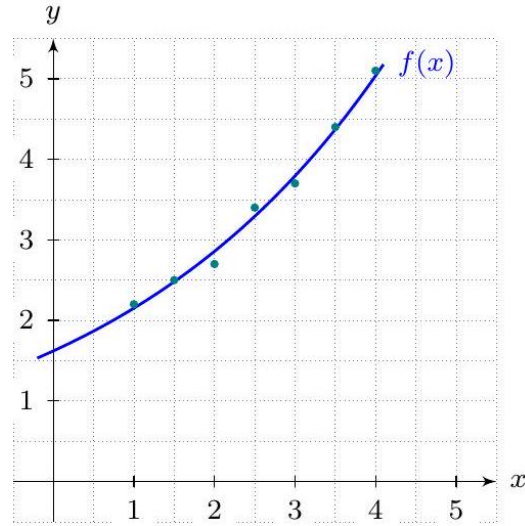


Figura 2: Pontos ajustados pela função

## 2 Procedimento

### 2.1 Mínimos Quadrados

Os métodos de mínimos quadrados (caso discreto, caso contínuo e caso não linear) foram implementados utilizando o MATLAB ®. Trata-se de uma linguagem de programação que expressa diretamente matrizes e arranjos matemáticos, combinando um ambiente de desktop ajustado para análise iterativa e processos de design com uma linguagem de programação.

Foi utilizada uma matriz extraída do site <https://math.nist.gov/MatrixMarket/>, que fornece acesso a um repositório de dados de teste para uso em estudos comparativos de algoritmos para álgebra linear numérica. A matriz escolhida foi bcsstk22.mtx, real simétrica indefinida, de ordem 138 e com 417 entradas (disponível em: <https://math.nist.gov/MatrixMarket/data/Harwell-Boeing/bcsstruc3/bcsstk22.html>).



## 2.2 Matriz de Hilbert

Uma função que receba um inteiro  $n$  e que retorne uma matriz de Hilbert de ordem  $n$  foi implementada e testados para todos os programas implementados utilizando a matriz de Hilbert para resolver o sistema linear com

$$a_{ij} = \frac{1}{i+j-1} \quad b = \frac{1}{3} \sum_{j=1}^n a_{ij}$$

Foram considerados  $n = 5$ ,  $n = 7$ ,  $n = 9$ ,  $n = 10$  e  $n = 12$ , cujos resultados foram analisados e comparados. Estes valores foram escolhidos baseados no Trabalho Computacional 1 - Métodos Diretos para Sistemas Lineares, onde foi verificado que, a partir da ordem 29, a matriz de Hilbert é singular, isto é, não possuem matriz inversa e, pela Decomposição de Cholesky, a partir da ordem 13, a matriz não é definida positiva.

Para a análise do caso discreto, foi necessário aplicar cálculos para matrizes de outras ordens além das estipuladas, a fim de ter resultados mais precisos para a análise.

Implementados os cálculos para as matrizes de Hilbert, os resultados foram organizados de acordo com o método implementados. A partir da escolha da precisão  $\epsilon$ , foram comparados os resultados para o código do *script* e a função `hilb(n)` do MATLAB ®. Foram analisados o tempo computacional, o número de iterações da resolução do sistema, ou seja, quantas iterações foram necessárias para obter a solução aproximada, e a precisão da resolução, a partir do erro relativo da última iteração ( $\epsilon$ ).

## 3 Resultados e discussão

### 3.1 Matriz de Hilbert

Implementados os cálculos para as matrizes de Hilbert, os resultados foram organizados de acordo com o método implementados. Para cada um deles, foram comparados os resultados para o código do *script* e a função `hilb(n)` do MATLAB, analisando o tempo computacional, o número de iterações e o erro relativo final de cada cálculo. Os resultados estão dispostos conforme as tabelas 1, 2, 3 e 4:

Tabela 1: Implementação do método de mínimos quadrados discreto.

n	Código implementado			Função $h_{ilb}(n)$		
	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
5	1.1218	692	2.5347e-23	1.1202	617	2.5347e-23
7	2.6592	1670	4.6651e-19	4.5710	1523	4.6651e-19
9	5.5306	3286	1.4809e-19	4.4535	3043	1.4809e-19
10	6.5109	4385	1.4225e-17	4.6216	4085	1.4225e-17
11	4.6071	5702	3.4324e-17	4.2779	5339	3.4324e-17
12	4.4488	7261	7.9599e-19	3.9928	6829	7.9599e-19
13	4.5422	9080	3.1788e-17	4.1320	8573	3.1788e-17
14	4.2893	11177	1.5346e-18	4.6639	10589	1.5346e-18
15	4.8191	40728	3.4728e-17	5.7082	12901	3.4728e-17

Pela tabela 1, conforme a ordem da matriz de Hilbert aumenta, nota-se que há um padrão para as variáveis averiguadas, isto é, há uma função crescente. Os cálculos são mais rápidos e precisos e o gasto computacional é menor quanto menor for a ordem da matriz de Hilbert. Pode-se ver que, comparando as ordens 5 e 11, o tempo computacional chega a ser mais de três vezes maior, a precisão cai  $1.35 \times 10^6$  vezes e gastos computacional aumenta mais de sete vezes. Os cálculos efetuados com a função  $h_{ilb}(n)$  apresentaram o mesmo padrão, mas os cálculos foram minuciosamente mais rápidos e com gastos computacionais ligeiramente menores; os erros foram os mesmos em relação aos cálculos com o código implementado. Não obstante, para os cálculos feitos pelo método em questão, a partir da ordem 12 o erro relativo final apresenta outro padrão, que vai diminuindo e aumentando conforme a alternância crescente da ordem da matriz.

Buscando a eficácia dos cálculos, foram analisados os cálculos pelo métodos de mínimos quadrados contínuo, alterando o intervalo da integração e fixando uma matriz de Hilbert de ordem 10.

Tabela 2: Implementação do método de mínimos quadrados contínuo para matriz de ordem 10.

Intervalo	Código implementado			Função $h_{ilb}(n)$		
	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
[0,1]	6.1828	4497	5.6557e-18	5.7095	4197	5.6557e-18
[0,3]	4.2704	4497	5.4437e-18	5.6747	4197	5.4437e-18
[0,5]	1.3563	4497	4.9111e-15	1.3995	4197	4.9111e-15
[0,6]	1.5192	4497	5.4437e-18	1.5792	4197	5.4437e-18
[0,7]	1.3381	4497	6.7039e-19	1.4160	4197	6.7039e-19
[0,10]	6.2103	4497	1.4225e-17	5.5181	4197	1.4225e-17
[0,30]	5.6475	4497	2.1486e-17	6.4585	4197	2.1486e-17
[0,50]	6.0618	4497	1.8509e-11	6.8717	4197	1.8509e-11
[0,100]	4.5959	4497	9.6639e-18	3.9999	4197	9.6639e-18

A análise da tabela 2 mostra que tanto para os cálculos feitos a partir do código implementado quanto a partir da função  $hilb(n)$  houve uma redução no número de iterações, ou seja, um gasto computacional menor, mas, apesar da mudança dos intervalos de integração, o número de iterações permaneceu o mesmo. Deste modo, parece interessante buscar outros métodos para obter resultados que tenham menor gasto computacional, uma vez que, para uma mesma matriz, diferentes intervalos de integração não resultaram em gastos computacionais menores. Entretanto, os resultados mostram que cálculos com intervalos de diferença 6 e 7 apresentaram os cálculos mais eficiente, isto é, com menores tempos computacionais e maiores precisões (menores erros). Mesmo assim, não há um padrão relação as estas variáveis que é observável, o que justifica a necessidade da análise de resultados a partir de cálculos implementados com outros métodos.

Foram então aplicados os métodos de mínimos quadrados não lineares com ajustes hiperbólico e exponencial.

Tabela 3: Implementação do método de mínimos quadrados não linear com ajuste hiperbólico.

n	Código implementado			Função $hilb(n)$		
	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
5	0.9568	702	8.1662e-16	1.0506	627	8.1662e-16
7	4.1138	1684	5.5974e-05	3.9283	1537	5.5974e-05
9	3.5240	3304	3.6708e-07	4.5175	3304	3.6708e-07
10	4.5400	4405	1.5752e-07	4.4822	4105	1.5752e-07
12	4.1265	7285	1.1104e-06	4.0639	6853	1.1104e-06

Tabela 4: Implementação do método de mínimos quadrados não linear com ajuste exponencial.

n	Código implementado			Função $hilb(n)$		
	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
5	1.1156	697	3.7352e-17	0.9682	622	3.7352e-17
7	5.0761	1677	1.7305e-06	4.2300	1530	1.7305e-06
9	4.1962	3295	9.3746e-09	4.3324	3052	9.3746e-09
10	4.1806	4395	3.7561e-09	3.9728	4095	3.7561e-09
12	4.0959	7273	3.2221e-08	6.6726	6841	3.2221e-08

A implementação de ambos os ajustes apresentam mudanças significativas nos resultados. Em comparação aos casos discreto e contínuo, o caso não linear tem menores números de iterações, ou seja, a aplicação do método de mínimos quadrados não lineares tem menor gasto computacional, mas este caso não é tão preciso quanto os anteriores, tendo em vista que o erro relativo final para matriz de ordem 10, por exemplo, está na grandeza de ordem -7, enquanto para os casos discreto

e contínuo (com intervalo  $[0,7]$ ) a ordem da grandeza é de -17 e -19, respectivamente.

Entre os ajustes implementados, o ajuste hiperbólico apresentou os menores tempos computacionais, porém, apesar de sutil, os cálculos com ajuste exponencial apresentou os menores números de iterações e, de maneira mais relevante, teve erros menores. Portanto, dos cálculos feitos a partir dos métodos computacionais não lineares, aqueles operados com ajuste exponencial tiveram os melhores resultados, tendo em vista o custo computacional e a precisão. Entretanto, em vista de uma melhor precisão, os cálculos com métodos de mínimos quadrados contínuos são melhores do que os casos não lineares, que tem melhores resultados de gasto computacional, em contrapartida.

Assim como nos casos discreto e contínuos, os cálculos no caso não linear foram mais eficientes utilizando a função *hilb*(*n*), do MATLAB.

### 3.2 Comparação com métodos iterativos

Os próximos cálculos também foram operados com os métodos anteriormente utilizados. Os cálculos mostram que a matriz utilizada neste estudo é esparsa, pois tem esparsidade igual a 174405663 e grau de esparsidade igual a 0.999996. A tabela 5 mostra os resultados para os cálculos aplicados com métodos de mínimos quadrados e métodos iterativos (foram escolhidos três cálculos com métodos iterativos que tiveram os melhores resultados no Trabalho 2), onde a matriz matriz foi utilizada em todos os cálculos.

Tabela 5: Implementação dos métodos de mínimos quadrados e dos métodos iterativos.

Método	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
Mínimos quadrados discreto	1.8220	26737359	1.953367e-19
Mínimos quadrados contínuo com intervalo $[0,20]$	0.8418	8912453	2.325598e-19
Mínimos quadrados não linear com ajuste hiperbólico	1.4824	8893545	2.306542e-16
Mínimos quadrados não linear com ajuste exponencial	1.9166	8893407	9.586003e-19
Gradiente Conjugado com Pré-Condicionamento	235.2001	250904297	8.4727e-11
Métodos de SOR com $\omega = 1$	15575.19	1918416	5.3764e-07
Gauss-Seidel sem reordenamento	1271.179	4076407	9.4507e-07

A fim de comparar os melhores resultados obtidos, foi necessário fazer previamente alguns cálculos utilizados o método de mínimos quadrados contínuo, pois, como já esperado e visto nos resultados da tabela 6, apesar do número de iterações não mudar com a variação do intervalo de integração, o tempo computacional e o erro relativo final apresentaram mudanças.

Tabela 6: Implementação dos métodos de mínimos quadrados contínuo.

Intervalo	Tempo computacional ( $10^{-3}s$ )	Número de iterações	Erro relativo final
[0,1]	2.9459	8912453	2.165208e-19
[0,3]	3.6547	8912453	1.188792e-19
[0,5]	6.3731	8912453	1.903644e-16
[0,7]	7.0131	8912453	3.799948e-19
[0,9]	0.8513	8912453	6.227557e-11
[0,10]	0.9377	8912453	7.928844e-19
[0,20]	0.8418	8912453	2.325598e-19
[0,30]	6.4032	8912453	3.200448e-19
[0,50]	0.9927	8912453	1.123649e-18

Deste modo, foi escolhido o resultado do cálculo com intervalo  $[0,20]$ , uma vez que tem o menor tempo computacional e o segundo menor erro relativo final (há uma diferença de, aproximadamente, 48% com o menor erro, mas, para o intervalo  $[0,3]$ , o tempo computacional é, aproximadamente, 3.34 vezes maior e, por isso, os valores do intervalo  $[0,20]$  foi escolhido) e, assim, inserido na tabela 5 para a comparação com outros cálculos.

Na tabela 5, analisando apenas os cálculos com métodos de mínimos quadrados, os resultados mostram as mesmas comparações obtidas com os resultados dos cálculos a partir da matriz de Hilbert. A implementação do caso contínuo gerou uma redução no número de iterações, assim como no tempo computacional do cálculo (66.66% e 53.80%, respectivamente), entretanto a precisão do resultado do cálculo ficou menor. Além disso, a aplicação do caso contínuo apresentou o menor tempo computacional entre os cálculos com métodos de mínimos quadrados, mas, entre eles, não demonstrou ser tão eficiente em questão da precisão, tendo em vista os resultados do caso discreto e com ajuste hiperbólico, e, em número de iterações, não teve melhores resultados do que os cálculos em casos não lineares (apesar da diferença relativamente pequena: aproximadamente, 0.2%).

Em contrapartida aos resultados da matriz de Hilbert, ambos os cálculos do caso não linear apresentaram o mesmo número de iterações, inclusive, o menor número de iterações entre os cálculos dos métodos de mínimos quadrados. Não obtiveram os melhores tempos computacionais, mas, com ajuste hiperbólico, houve uma melhora mais relevante em relação ao caso discreto e comparado ao resultado do cálculo com ajuste exponencial, como ocorreu para a matriz de Hilbert. No entanto, a precisão do cálculo com ajuste hiperbólico é muito pior quando comparado aos demais resultados para métodos de mínimos quadrados (a precisão cai numa ordem de grandeza de  $10^{-3}$ ).

Quando os resultados dos cálculos dos métodos de mínimos quadrados são comparados aos dos métodos iterativos, vê-se uma expressiva diferença em todas as variáveis calculadas. Os cálculos de mínimos quadrados são abruptamente mais rápidos e extremamente mais preciso, no entanto, os métodos iterativos, especificamente os métodos de SOR com  $\omega = 1$  e Gauss-Seidel sem reordenamento, mostraram menores gastos computacionais, isto é, menores números de iterações (o número de iterações do método de Gauss-Seidel é menos da metade em comparações ao do caso não linear, enquanto, para o métodos de SOR com  $\omega = 1$ , o número é, aproximadamente, sete vezes menor).

## 4 Conclusão

Os métodos de mínimos quadrados foram implementados computacionalmente e foram feitas comparação entre todos os casos em relação ao número de iterações, tempo computacional e erro relativo final para a matriz de Hilbert e para uma matriz específica. Os cálculos que utilizaram os métodos de mínimos quadrados mostraram que cada caso apresenta uma eficiência maior em certo aspecto. O caso discreto é o mais preciso dentre eles, o caso contínuo é o mais rápido e o caso não linear é que possui menor gastos computacional. Todavia, buscando a melhor eficiência dentre eles, o método de mínimos quadrados no caso contínuo parece ser o melhor cálculo, desde uma análise do intervalo de integração seja gerenciado previamente. Além do mais, os resultados mostram que o caso não linear não é recomendado para cálculos que exigem melhores precisões. Em comparação aos métodos iterativos, os cálculos a partir de métodos de mínimos quadrados são expressamente mais rápidos e precisos, mas, em alguns casos, tem maiores gastos computacionais. Ainda sim, os métodos de mínimos quadrados são melhores que os métodos iterativos em geral.

## 5 Referências

AJUSTE DE CURVAS PELO MÉTODO DOS QUADRADOS MÍNIMOS. Disponível em: [http://wwwp.fc.unbal.br/Iniciacao\\_Cientifica/interpolacao/teoria/4\\_Metodo\\_dos quadrados \\_minimos.pdf](http://wwwp.fc.unbal.br/Iniciacao_Cientifica/interpolacao/teoria/4_Metodo_dos%20quadrados_minimos.pdf). Acesso em: 10 nov. 2022.

BURDEN, Richard L.; FAIRES, J. Douglas. Numerical Analysis. 9. ed. Boston: Cengage Learning, 2010.

LECTURE 19: Continuous Least Squares Approximation. 2009. Disponível em: <http://www.sci.utah.edu/~beiwang/teaching/cs6210-fall-2016/lecture19.pdf>  
Acesso em: 16 nov. 2022.

OLVER, Peter J.; SHAKIBAN, Chehrzad. Applied Linear Algebra. 2. ed. San Francisco: Springer, 2018.