

## **Projeto 2**

|                          |            |
|--------------------------|------------|
| Gabriel Belém Barbosa    | RA: 234672 |
| Diogo Batalha            | RA: 169955 |
| Felipe Kenzo Obata Piton | RA: 171108 |
| Felipe Purini Policastro | RA: 215760 |

22 de Maio de 2021

# Conteúdo

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>                                  | <b>3</b>  |
| <b>2</b> | <b>Desenvolvimento</b>                             | <b>3</b>  |
| 2.1      | Parte I. Comparando diferentes abordagens. . . . . | 3         |
| 2.1.1    | Problema teste . . . . .                           | 3         |
| 2.1.2    | Procedimento . . . . .                             | 5         |
| 2.1.3    | Análise e discussão . . . . .                      | 6         |
| 2.2      | Parte II. Algo mais. . . . .                       | 8         |
| <b>3</b> | <b>Conclusão</b>                                   | <b>10</b> |
| <b>4</b> | <b>Referências</b>                                 | <b>10</b> |
| <b>5</b> | <b>Apêndice</b>                                    | <b>12</b> |

# 1 Introdução

O problema de quadrados mínimos (QM) lineares é de fundamental importância em diversas áreas do conhecimento, como nas engenharias e na física experimental e computacional, entre outras. O problema baseia-se em encontrar, através de equações lineares, uma curva ou superfície em maiores dimensões que melhor se adequa a pontos de interesse. Mais especificamente, busca-se obter

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2$$

em que  $A \in \mathbb{R}^{m \times n}$  e  $b \in \mathbb{R}^m$ . Encontrar  $x$  que satisfaz a equação acima é o mesmo que resolver o sistema  $A^T Ax = A^T b$ <sup>[1]</sup>, denominado sistema normal.

Solucionar tal sistema é um objetivo que aparece na obtenção de alternativas computacionalmente econômicas para conferir valores de uma função de interesse em um dado intervalo, ou para a melhor visualização, entendimento do comportamento ou confirmação da teoria por trás de dados experimentais. Utilizando um sistema algébrico matricial, esse projeto busca analisar métodos numéricos para a simplificação do sistema normal, no qual fundamenta-se a solução de quadrados mínimos.

Mais objetivamente, a Parte I desse projeto visa a análise qualitativa e de custo de formas do tratamento de um problema teste e seu respectivo sistema normal, nominalmente a solução de equações normais via fatoração de Cholesky, solução via decomposição  $QR$  condensada através do processo de Gram-Schmidt modificado (GSM) e fatoração  $QR$  completa com transformações de Householder. Além disso, na Parte II, munido das conclusões obtidas dos resultados anteriores, um novo problema será apresendo e tratado com a escolha justificada de um dos três métodos.

## 2 Desenvolvimento

### 2.1 Parte I. Comparando diferentes abordagens.

#### 2.1.1 Problema teste

O sistema algébrico matricial escolhido foi Octave. Com o intuito de aproximar numericamente a curva de  $f(x) = \exp(\sin(6x))$  em  $x \in [0, 1]$ , usa-se a função `linspace` do Octave para obter-se o espaço de  $x$ , `Xspace`, com 21 elementos igualmente espaçados no intervalo desejado. Após isso é construída a matriz de

Vandermonde desse espaço de  $x$  usando a função vander. Esta é da forma

$$\begin{pmatrix} x_0^{20} & x_0^{19} & \dots & x_0^0 \\ x_1^{20} & x_1^{19} & \dots & x_1^0 \\ \vdots & \vdots & \ddots & \vdots \\ x_{20}^{20} & x_{20}^{19} & \dots & x_{20}^0 \end{pmatrix}$$

Porém, como se deseja um polinômio de grau 10 somente, eliminam-se as primeiras 10 colunas dessa matriz. Também é obtido o espaço associado ao espaço em  $x$ ,  $Yspace$ , aplicando  $f(x)$  ponto a ponto. O sistema linear para o qual deseja-se encontrar a solução de quadrados mínimos fica

$$\underbrace{\begin{pmatrix} x_0^{10} & x_0^9 & \dots & x_0^0 \\ x_1^{10} & x_1^9 & \dots & x_1^0 \\ \vdots & \vdots & \ddots & \vdots \\ x_{20}^{10} & x_{20}^9 & \dots & x_{20}^0 \end{pmatrix}}_A \underbrace{\begin{pmatrix} c_{10} \\ c_9 \\ \vdots \\ c_0 \end{pmatrix}}_c \simeq \underbrace{\begin{pmatrix} y(x_0) \\ y(x_1) \\ \vdots \\ y(x_{20}) \end{pmatrix}}_y$$

Em que  $c$  a matriz dos coeficientes desejada. A matriz  $A$  resultante é uma matriz  $m \times n$  com  $m > n$  e posto completo  $n$ , e o problema recai no sistema normal  $A^T A c = A^T y$ , sendo a aplicação de uma fatoração  $QR$  de  $A$ ,  $Q$  matriz ortogonal e  $R$  triangular superior, empregada para que a solução seja

$$\begin{aligned} c &= (A^T A)^{-1} A^T y \\ &= (R^T Q^T Q R)^{-1} R^T Q^T y \\ &= (R^T R)^{-1} R^T Q^T y \\ &= R^{-1} (R^T)^{-1} R^T Q^T y \\ &= R^{-1} Q^T y \end{aligned} \tag{1}$$

Onde foi usado que  $R$  é invertível e  $Q$  ortogonal por construção. Ao contrário de uma decomposição algébrica de  $A$ , o que é obtido em qualquer método numérico, na realidade, são aproximações  $\tilde{R}$  e  $\tilde{Q}$  tal que  $A = \tilde{Q}\tilde{R} + E$ , sendo  $E$  uma correção para os erros numéricos. De fato, sequer a matriz  $\tilde{Q}$  é ortogonal, sendo, no entanto, possivelmente próxima de o ser numericamente, dependendo da estabilidade do método e do condicionamento da própria matriz  $A$ . Além disso, há a solução do sistema normal em si, que carrega e intensifica erros e gera novos arredondamentos, independe da fatoração, fato esse que será tratado na Análise e Discussão ao final desta parte com a separação do erro em estágios.

Vale notar que as colunas da  $A$  poderiam ser permutadas, sabendo-se que a mesma permutação ocorreria entre os elementos de  $c$ , isto é, ao se permutar as

colunas  $j$  e  $k$  de  $A$ , ao final do processo o coeficiente  $c_j$  e  $c_k$  estariam um no lugar do outro em  $c$ . Tais permutações, entretanto, não afetam o número de condição e a estabilidade do problema, pois, sendo  $P$  uma matriz de permutação, e sendo, portanto  $P^{-1} = P^T$ , tem-se que o número de condição na norma 2,  $\kappa_2$ , de  $PA$  é

$$\kappa_2(PA) = \frac{\sigma_{\max}(PA)}{\sigma_{\min}(PA)} = \sqrt{\frac{\lambda_{\max}(A^T P^T P A)}{\lambda_{\min}(A^T P^T P A)}} = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}} = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} = \kappa_2(A)$$

Com a notação usual para valores essenciais,  $\sigma$ , e autovalores,  $\lambda$ .

### 2.1.2 Procedimento

Para solucionar por transformações de Householder, foi usada a função já presente no Octave, `qr`, cuja rotina é baseada em Householder<sup>[1]</sup>. A função aplicada em  $A$  já retorna as matrizes  $Q_H \in \mathbb{R}^{m \times m}$  e  $R_H \in \mathbb{R}^{m \times n}$  em suas formas completas, e o processo demanda teoricamente  $2mn^2 - \frac{2n^3}{3} - 2mn + \frac{2n}{3}$  *flops*<sup>[3]</sup>. Vale ressaltar que a matriz  $R_H$  desse processo não é necessariamente equivalente às dos demais métodos (mesmo desconsiderando erros numéricos) devido às escolhas estabilizadoras que não garantem positividade estrita da diagonal principal.

Para a fatoração de Cholesky, usa-se a função `chol`, empregada na matriz  $A^T A \in \mathbb{R}^{n \times n}$  quadrada positiva definida de posto completo para a obtenção de  $R_C \in \mathbb{R}^{n \times n}$  condensada com elementos da diagonal estritamente positivos, esta então sendo usada para a obtenção de  $Q_C \in \mathbb{R}^{m \times n}$ , também em sua forma condensada, através da solução da relação de matrizes  $Q_C R_C = A$ , uma vez que a matriz  $R_C$  obtida é em teoria a mesma encontrada por GSM de  $A$ <sup>[2]</sup>, e portanto possui sua contraparte ortogonal que obedece a relação. O custo operacional de multiplicar  $A^T A$  é  $2mn^2 - n^2$  *flops*<sup>[5]</sup>, enquanto o custo de sua fatoração de Cholesky é  $\frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}$  *flops*<sup>[5]</sup>, e por fim a solução de  $Q_C R_C = A$ , efetuada como  $Q_C = A R_C^{-1}$  demanda  $\frac{n^3}{3} + \frac{2n}{3}$  *flops*<sup>[5]</sup> para a inversão da matriz triangular  $R$  e mais  $n^3$  *flops*<sup>[5]</sup> para multiplicação entre a matriz sólida  $A \in \mathbb{R}^{m \times n}$  e  $R_C^{-1}$ , também triangular. O custo total é, em teoria,  $2mn^2 + \frac{5n^3}{3} - \frac{n^2}{2} + \frac{5n}{6}$  *flops*.

Já para GSM, um simples algoritmo foi desenvolvido com base na teoria desse método<sup>[2]</sup>, começando com as matrizes nulas  $\mathbb{R}^{n \times n}$  e  $\mathbb{R}^{m \times n}$ , que serão preenchidas para se tornarem  $R_{GS}$  e  $Q_{GS}$  respectivamente, ambas na forma condensada, como a seguir.

```
Para j=1,2,...,n faça
  Para i=1,2,...,j-1 faça
    r(i,j)=<qi,aj>
    aj=aj-r(i,j)*qi
```

```

r(j,j)=norma2(aj)
Se r(j,j) é nulo
  Pare
qj=aj/r(j,j)

```

Sendo  $a_p$  e  $q_p$  as  $p$ -ésimas colunas de  $A$  e  $Q_{GS}$ , respectivamente, e o produto interno sendo o usual. Como o posto de  $A$  é completo, o algoritmo não para prematuramente. Vale ressaltar que o algoritmo acima destrói a matriz original  $A$  durante o processo, portanto uma matriz auxiliar que é uma cópia de  $A$  deve ser empregada caso deseja-se trabalhar com esta última após a descomposição. O custo operacional teórico é  $2mn^2 + mn \text{ flops}$ <sup>[4]</sup>.

Ao final das 3 fatorações é efetuado o comando  $c=R \setminus (Q' * Yspace)$  que resolve o sistema normal simplificado, como visto em (1), encontrando a solução dos coeficientes ótimos. Como esse passo é o mesmo para todos os métodos, ele não agrega nada à análise de custo operacional, e será omitido da contagem de *flops*.

### 2.1.3 Análise e discussão

Os coeficientes encontrados para cada método são apresentados a seguir.

Tabela 1: Coeficientes ótimos obtidos por cada método

|          | Householder | Cholesky    | GSM         |
|----------|-------------|-------------|-------------|
| $c_{10}$ | 2718.7511   | 2718.6578   | 2718.7511   |
| $c_9$    | -11252.3527 | -11251.9155 | -11252.3528 |
| $c_8$    | 17030.607   | 17029.7489  | 17030.6072  |
| $c_7$    | -8947.7019  | -8946.7896  | -8947.7022  |
| $c_6$    | -4124.843   | -4125.4097  | -4124.8428  |
| $c_5$    | 7503.2788   | 7503.4845   | 7503.2787   |
| $c_4$    | -3523.8442  | -3523.8846  | -3523.8441  |
| $c_3$    | 622.2207    | 622.224     | 622.2207    |
| $c_2$    | -33.8651    | -33.865     | -33.8651    |
| $c_1$    | 7.5069      | 7.5069      | 7.5069      |
| $c_0$    | 0.99926     | 0.99926     | 0.99926     |

Foi obtido também o maior erro pontual entre a aproximação  $QR$  e  $A$  para cada método, como visto a seguir.

Tabela 2: Máxima diferença absoluta entre  $A$  e  $QR$  de cada método

|                            | Householder | Cholesky   | GSM        |
|----------------------------|-------------|------------|------------|
| $\max(\text{abs}(QR - A))$ | 1.3323e-15  | 2.1538e-10 | 3.3307e-16 |

Com o intuito de separar as influências dos métodos no erro final do polinômio aproximador dos demais processos, foi calculado, para cada método, o erro quadrático médio (MSE-mean squared error) entre a aproximação para  $QR$  e  $A$  além do MSE entre o polinômio encontrado e sua contraparte  $f(x)$  ( $Yspace$ ), como pode ser visto na tabela a seguir. O MSE é calculado como a soma dos quadrados das diferenças ponto a ponto das matrizes ou vetores em questão, dividida pelo número de elementos. Também foram plotadas as diferenças ao quadrado ponto à ponto em função de  $x$  para as três abordagens, que estão em três gráficos no Apêndice.

Tabela 3: MSE's de cada método

|                   | Householder | Cholesky   | GSM        |
|-------------------|-------------|------------|------------|
| $MSE(QR, A)$      | 5.2588e-32  | 3.9864e-22 | 3.9640e-33 |
| $MSE(Ac, f(x_i))$ | 1.8062e-05  | 1.8062e-05 | 1.8062e-05 |

A tabela a seguir contém o número teórico de *flops* encontrados anteriormente.

Tabela 4: Número teórico de *flops* por método

|              | Householder                                   | Cholesky  | GSM          |
|--------------|---|---|--------------|
| <i>flops</i> | $2mn^2 - \frac{2n^3}{3} - 2mn + \frac{2n}{3}$ | $2mn^2 + \frac{5n^3}{3} - \frac{n^2}{2} + \frac{5n}{6}$ | $2mn^2 + mn$ |

Usando a função `cond` do Octave, cuja rotina envolve a decomposição em valores essenciais<sup>[6]</sup>, foi calculado o número de condição na norma 2,  $\kappa_2$ , das matrizes triangulares obtidas com cada método, como visto a seguir.

Tabela 5: Número de condição de  $R$  por método

|               | Householder | Cholesky   | GSM        |
|---------------|-------------|------------|------------|
| $\kappa_2(R)$ | 2.3175e+07  | 2.3126e+07 | 2.3175e+07 |

E também foi obtido o número de condição  $\kappa_2(A^T A) = 5.3699e+14$ . Apesar do grande número de condição de  $A^T A$ , a precisão computacional e estabilidade dos três métodos é suficiente para gerar uma boa aproximação de  $A$ , como visto na Tabela 3, sendo o maior MSE dentre os métodos o da fatoração de Cholesky, na ordem de  $10^{-22}$ , e o pior erro pontual na Tabela 2, também proveniente desse método, é na ordem de  $10^{-10}$ . Isso implica que, apesar de erros numéricos, e sendo os  $\kappa_2$  na Tabela 5 na ordem de  $10^7$ , os erros para a aproximação de  $A$  são irrisórios comparados com o que seria necessário para diferenças significativas dado o número de condição do sistema normal  $Rc = Q^T y$ , mesmo considerando o elemento de maior imprecisão de qualquer método (é um sistema relativamente

bem condicionado dada a precisão de máquina e estabilidade dos métodos). Para o problema em questão, portanto, a Tabela 4 é a mais relevante quando se trata de decidir qual dos métodos é o mais adequado, sendo, dentre as opções, a fatoração de Householder a mais econômica e que melhor escala com a ordem do problema.

## 2.2 Parte II. Algo mais.

O problema de quadrados mínimos aparece comumente em aproximação de dados experimentais. Nesse contexto, muitas vezes as funções aproximadoras escolhidas são diversas e vão além de simples polinômios como aquele usado no problema teste, dependendo da natureza dos dados em questão. Para essa parte será apresentado um problema de modelagem de dados de poluição atmosférica na forma da concentração de  $NO$  em uma rua movimentada durante um período de 24 horas (veja o exemplo 2 e 3 em [8]). A aproximação será através de funções trigonométricas e uma constante. Os dados experimentais são

Tabela 6: Concentração de  $NO$   $y_i$  ( $\mu g/m^3$ ) em função do tempo  $t_i$  (Horas)

| $t_i$ | $y_i$  | $t_i$ | $y_i$  | $t_i$ | $y_i$  | $t_i$ | $y_i$  | $t_i$ | $y_i$  |
|-------|--------|-------|--------|-------|--------|-------|--------|-------|--------|
| 0     | 110.49 | 5     | 29.37  | 10    | 294.75 | 15    | 245.04 | 20    | 216.73 |
| 1     | 73.72  | 6     | 74.74  | 11    | 253.78 | 16    | 286.74 | 21    | 185.78 |
| 2     | 23.39  | 7     | 117.02 | 12    | 250.48 | 17    | 304.78 | 22    | 171.19 |
| 3     | 17.11  | 8     | 298.04 | 13    | 239.48 | 18    | 288.76 | 23    | 171.73 |
| 4     | 20.31  | 9     | 348.13 | 14    | 236.52 | 19    | 247.11 | 24    | 164.05 |

Fonte: Mahooti, 2019

As funções escolhidas para a aproximação de dados são

$$f_0(t) = 1, f_1(t) = \sin(\omega t), f_2(t) = \cos(\omega t), f_3(t) = \sin(2\omega t), f_4(t) = \cos(2\omega t), \dots$$

Com o total de nove funções, sendo  $w = \frac{\pi}{12}$  o fator de correção que transforma o ciclo de 24 horas em  $2\pi$ , o que é proveitoso para que todas as funções envolvidas, ao término de um dia, se repitam exatamente no próximo, criando um ciclo dário. Haverão 25 equações lineares da forma

$$y_i = c_0 + c_1 \sin(\omega t_i) + c_2 \cos(\omega t_i) + \dots + c_8 \cos(4\omega t_i), i = 0, 1, \dots, 24$$

O sistema cuja melhor aproximação é buscada fica

$$\underbrace{\begin{pmatrix} \cos(4\omega t_0) & \sin(4\omega t_0) & \dots & 1 \\ \cos(4\omega t_1) & \sin(4\omega t_1) & \dots & 1 \\ \vdots & & & \\ \cos(4\omega t_{24}) & \sin(4\omega t_{24}) & \dots & 1 \end{pmatrix}}_A \underbrace{\begin{pmatrix} c_8 \\ c_7 \\ \vdots \\ c_0 \end{pmatrix}}_c \simeq \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{24} \end{pmatrix}}_y$$



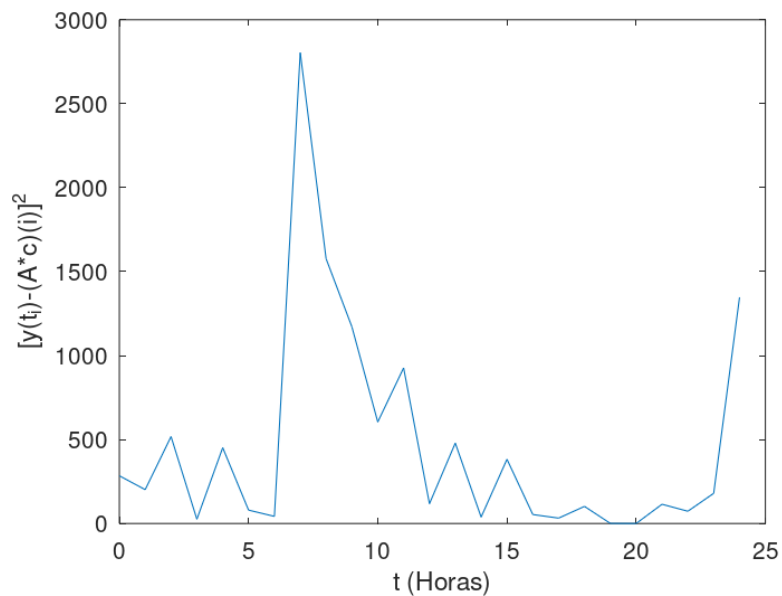
Utilizando a função `linspace`, em seguida foi construído o espaço de  $t$ ,  $Tspace$ , e também montado o espaço  $Yspace$  com os valores de  $y$  da Tabela 6. Através de operações vetoriais do `Octave` sobre  $Tspace$  foi construída a matriz  $A$  como descrita. Aqui foi usada a decomposição de Householder e a solução do sistema normal da mesma maneira do problema teste, usando a relação (1). Este método foi escolhido pelos mesmos motivos citados na Análise e Discussão da Parte I, isto é, o menor custo computacional e equivalente performance dentro das margens de precisão desejadas. Os coeficientes encontrados foram

Tabela 7: Coeficientes ótimos da modelagem de dados atmosféricos

|       | Valores  |
|-------|----------|
| $c_8$ | -7.5190  |
| $c_7$ | -5.8197  |
| $c_6$ | 37.3292  |
| $c_5$ | 25.4682  |
| $c_4$ | 1.7667   |
| $c_3$ | -58.4353 |
| $c_2$ | -93.4740 |
| $c_1$ | -73.2370 |
| $c_0$ | 189.2455 |

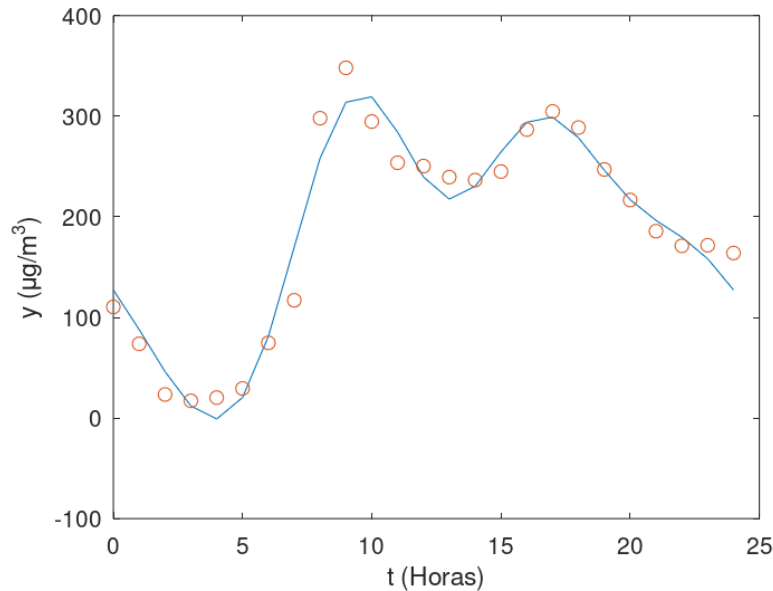
E a seguir o gráfico do erro ponto a ponto ao quadrado.

Figura 1: Erro quadratico ponto a ponto



E o gráfico dos dados e da aproximação sobrepostos.

Figura 2: MMQ vs.  $y(t_i)$



O MSE entre os pontos experimentais e a aproximação foi de 464.18, o que não é de se surpreender para uma modelagem de dados experimentais (que não seguem precisamente uma função devido ao *white noise*<sup>[7]</sup>) dada a magnitude dos valores envolvidos. O pior erro pontual entre a fatoração *QR* numérica por Householder e a matriz *A* original foi de 1.1102e-15, enquanto o MSE entre estes foi de somente 5.0901e-32. O  $\kappa_2$  da matriz *R* encontrada e de  $A^T A$  são 1.4556 e 2.1187, respectivamente, sendo então esse problema, em comparação, muito mais estável que o anterior, portanto as mesmas conclusões anteriores (com respeito à análise dos erros dos métodos em contraste aos números de condição) são aplicáveis ao problema atual.

### 3 Conclusão

### 4 Referências

1. C. B. Moler, *Numerical Computing with MATLAB*, Philadelphia: SIAM, 2004.
2. D. S. Watkins, *Fundamentals of Matrix Computations*, New Jersey: John Wiley & Sons, 2 ed., 2002.

3. R. A. van de Geijn, *Notes on Householder QR Factorization*, Austin: The University of Texas at Austin, 2014. Disponível em: <<https://www.cs.utexas.edu/users/flame/Notes/NotesOnHouseholderQR.pdf>> Acesso em: 20 mai. 2021.
4. R. A. van de Geijn, *Notes on Gram-Schmidt QR Factorization*, Austin: The University of Texas at Austin, 2014. Disponível em: <<https://www.cs.utexas.edu/users/flame/Notes/NotesOnGS.pdf>> Acesso em: 21 mai. 2021.
5. R. Hunger, *Floating Point Operations in Matrix-Vector Calculus*. Munich: mediaTUM – the media and publications repository of the Technical University of Munich, versão 1.3, 2007. Disponível em: <<https://mediatum.ub.tum.de/doc/625604/625604>> Acesso em: 21 mai. 2021.
6. Octave Forge. *Function Reference: cond*. Disponível em: <<https://octave.sourceforge.io/octave/function/cond.html>> Acesso em: 21 mai. 2021.
7. M. Mahooti. *Least Squares Fitting Trigonometric Functions (MATLAB code)*. ResearchGate, 2019. Disponível em: <[https://www.researchgate.net/publication/337653997\\_Least\\_Squares\\_Fitting\\_Trigonometric\\_Functions\\_MATLAB\\_code](https://www.researchgate.net/publication/337653997_Least_Squares_Fitting_Trigonometric_Functions_MATLAB_code)> Acesso em: 28 mai. 2021.

## 5 Apêndice

Figura 3: Erro quadrático ponto a ponto para Householder

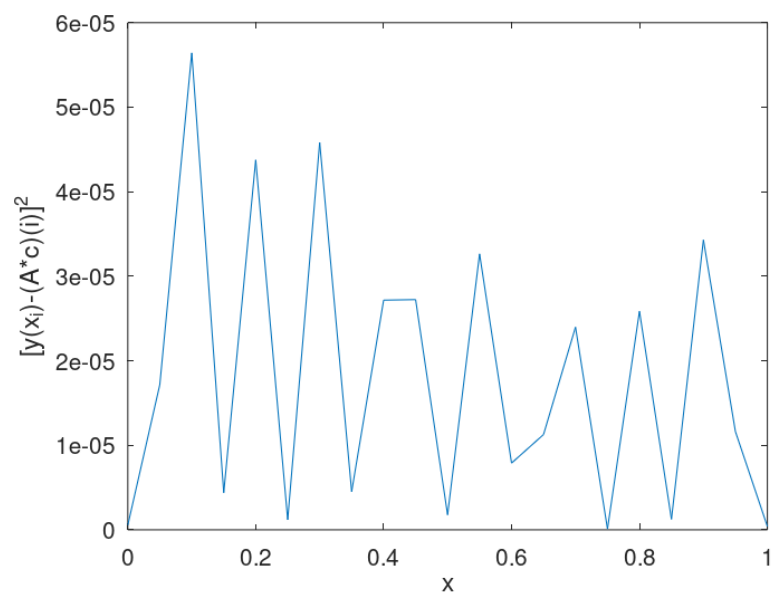


Figura 4: Erro quadrático ponto a ponto para Cholesky

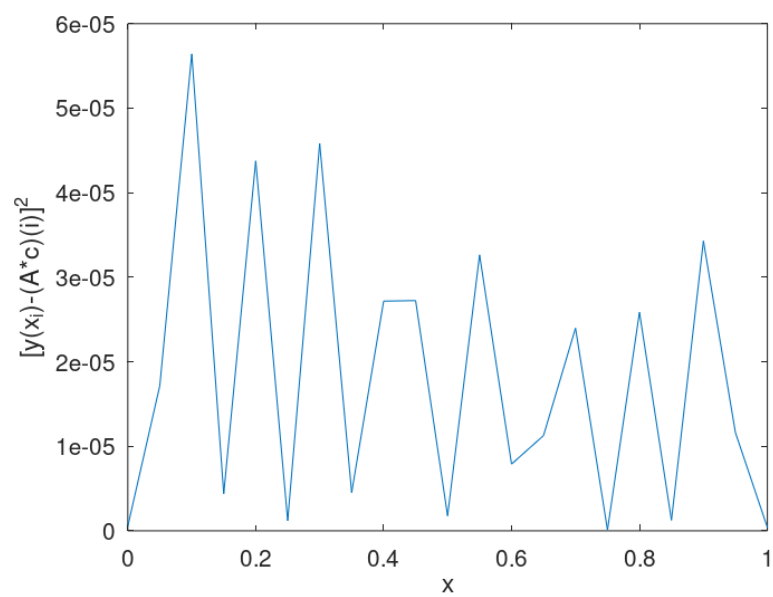
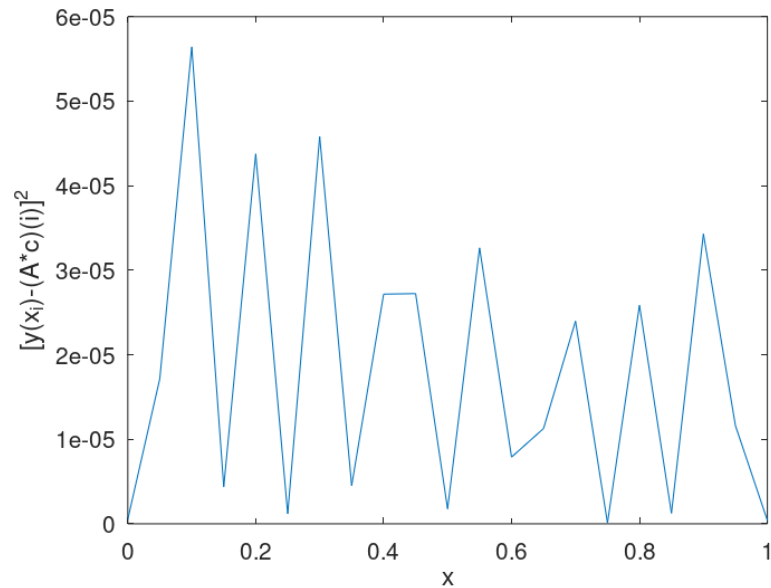


Figura 5: Erro quadrático ponto a ponto para GSM



Código para Householder:

```
n=11;
m=21;
Xspace=linspace(0,1,m);
A=vander(Xspace);
A=A(:,n:end);
Yspace=transpose(exp(sin(6.*Xspace)));

[Q,R]=qr(A);
c=R\ (Q'*Yspace);

err=(Q*R-A);
max(max(abs(err)))
err=err.^2;
MSE=sum(err(:))/numel(A)

err=(Yspace-A*c).^2;
MSE=sum(err(:))/m

plot(Xspace, err)
xlabel('x')
ylabel('[y(x_i)-(A*c)(i)]^2')
```

```

figure()

plot(Xspace, Yspace, Xspace, A*c)
xlabel('x')
ylabel('y')

cond(R)

Código para Cholesky:

n=11;
m=21;
Xspace=linspace(0,1,m);
A=vander(Xspace);
A=A(:,n:end);
Yspace=transpose(exp(sin(6.*Xspace)));

ATA=A'*A;
R=chol(ATA);
Q=A*inv(R);
c=R\ (Q'*Yspace);

err=(Q*R-A);
max(max(abs(err)))
err=err.^2;
MSE=sum(err(:))/numel(A)

err=(Yspace-A*c).^2;
MSE=sum(err(:))/m

plot(Xspace, err)
xlabel('x')
ylabel(' [y(x_i)-(A*c)(i)]^2 ')
figure()

plot(Xspace, Yspace, Xspace, A*c)
xlabel('x')
ylabel('y')

cond(ATA)
cond(R)

```

Código para GSM:

```
n=11;
m=21;
Xspace=linspace(0,1,m);
A=vander(Xspace);
A=A(:,n:end);
Acopia=A;
Yspace=transpose(exp(sin(6.*Xspace)));
R=zeros(n);
Q=zeros(m,n);

for j=1:n
    for i=1:j-1
        R(i,j)=Q(:,i)'*A(:,j);
        A(:,j)-=R(i,j)*Q(:,i);
    end
    R(j,j)=norm(A(:,j));
    if (R(j,j)==0)
        break
    end
    Q(:,j)=A(:,j)/R(j,j);
end
c=R\ (Q'*Yspace);

err=(Q*R-Acopia);
max(max(abs(err)))
err=err.^2;
MSE=sum(err(:))/numel(A)

err=(Yspace-Acopia*c).^2;
MSE=sum(err(:))/m

plot(Xspace, err)
xlabel('x')
ylabel('[y(x_i)-(A*c)(i)]^2')
figure()

plot(Xspace, Yspace, Xspace, Acopia*c)
xlabel('x')
ylabel('y')
```

```
cond(R)
```

Código para a Parte II:

```
n=9;
m=25;
Tspace=transpose(linspace(0,24,25));
A=[cos((4*pi/12).*Tspace), sin((4*pi/12).*Tspace), cos((3*pi/12).*Tspace),
sin((3*pi/12).*Tspace), cos((2*pi/12).*Tspace), sin((2*pi/12).*Tspace),
cos((pi/12).*Tspace), sin((pi/12).*Tspace), ones(m,1)];
Yspace=[110.49;
73.72;
23.39;
17.11;
20.31;
29.37;
74.74;
117.02;
298.04;
348.13;
294.75;
253.78;
250.48;
239.48;
236.52;
245.04;
286.74;
304.78;
288.76;
247.11;
216.73;
185.78;
171.19;
171.73;
164.05]

[Q,R]=qr(A);
c=R\ (Q'*Yspace);

err=(Q*R-A);
max(max(abs(err)))
```



```

err=err.^2;
MSE=sum(err(:))/numel(A)

err=(Yspace-A*c).^2;
MSE=sum(err(:))/m

plot(Tspace, err)
xlabel('t (Horas)')
ylabel('[y(t_i)-(A*c)(i)]^2')
figure()

plot(Tspace, A*c)
hold();
scatter(Tspace, Yspace)
xlabel('t (Horas)')
ylabel(strcat('y (\mu', 'g/m^3)'))

cond(R)
cond(A'*A)

```