Decomposição QR para resolver problemas de quadrados mínimos

Dyckson Ternoski
Marina Sayuri Vieira
Monique Baptista Fragozo
Otávio Dittrich Moreira

UFPR - Cálculo Numérico - 3º Semestre

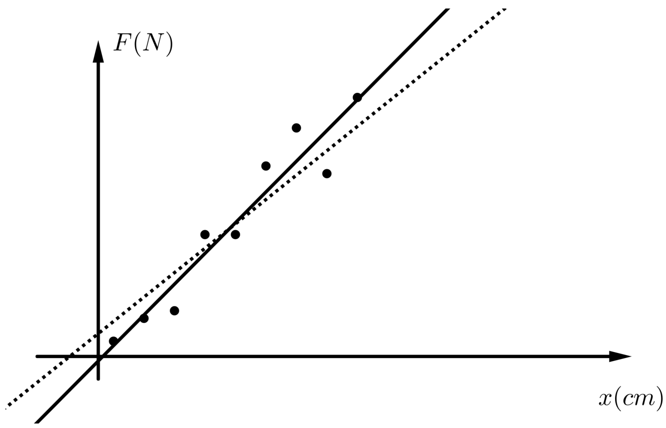
1. Introdução

1.1. Problema de Quadrados Mínimos

Na matemática, é muito importante o estudo de sistemas de equações lineares, pois modelam vários fenômenos em diversos casos. Sabemos, porém, que nem sempre conseguimos obter uma solução exata. Por exemplo, em uma indústria de molas, é necessário especificar a *constante da mola* proveniente da Lei de Hooke, em homenagem ao físico inglês Robert Hooke (1635 - 1703):

$$F = -kx$$

Onde F é a força aplicada à mola e x o deslocamento obtido. Para obte-lá, realizam diversas medidas variando a força e medindo o deslocamento. Porém, toda medição sempre é acometida por um erro, o que torna quase impossível que os valores gerem um sistema consistente, caindo em uma situação como a da figura abaixo.



Como decidir qual é a melhor reta? Melhor em que sentido? Sigamos com a formulação do problema.
Nesse exemplo, as medidas formam o seguinte sistema de equações:

$$\begin{aligned} kx_1 + p &= F_1 \\ kx_2 + p &= F_2 \\ &\vdots \\ kx_m + p &= F_m \end{aligned}$$

onde F_i representa a força aplicada e x_i o deslocamento observado, para $i = 1, 2, \ldots, m$, sendo m o número de medições. As incógnitas do problema são k e p.

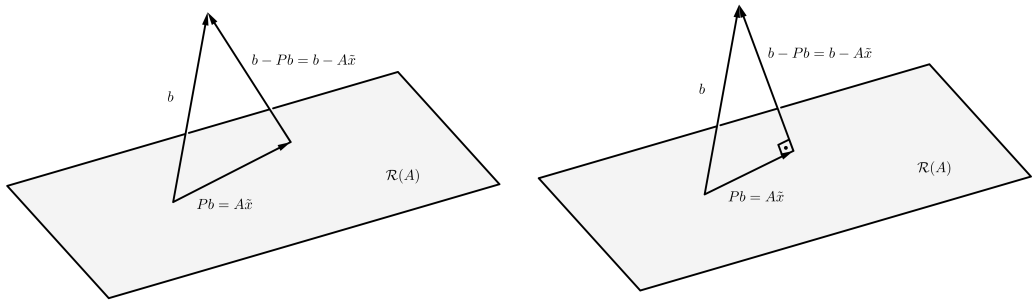
Como resolver esse tipo de problema?

Em um sistema $Ax = b$ inconsistente, sabemos que b não pertence ao espaço coluna de A. Então para obter uma solução razóavel devemos trabalhar com um vetor relativamente próximo de b que pertença a $\mathcal{R}(A)$. Sabemos da Álgebra Linear que o melhor que podemos conseguir é a projeção ortogonal de b sobre $\mathcal{R}(A)$, obtendo o seguinte problema:

$$A\tilde{x} = \text{proj}_{\mathcal{R}(A)} b = Pb \tag{1}$$

para algum projetor P. Assim temos um sistema consistente. O vetor $b - Pb$ será o vetor erro. Como queremos que o erro seja o menor possível, estaremos interessados em encontrar $\tilde{x}$ tal que $b - Pb$ seja o menor possível, ou seja, que minimize a norma $\|b - Pb\|$, ou analogamente, $\|b - Pb\|^2$, donde deriva o nome *quadrados mínimos*.

Da figura da esquerda fica claro que, para que o vetor $A\tilde{x} - b$ for ortogonal ao espaço-coluna de A. O projetor p deve ser, portanto, ortogonal, como na figura da direita.



Assim, se $A\tilde{x} - b$ for ortogonal ao espaço-coluna de A, $b - A\tilde{x}$ temos pelo Teorema Fundamental da Álgebra Linear que:

$$A^t(b - A\tilde{x}) = 0$$

isto é,

$$A^tA\tilde{x} = A^tb \tag{2}$$

Para que o sistema (2) possua uma única solução, é necessário que a matriz quadrada A^tA seja invertível e, se for, precisaremos calcular essa inversa. Para sistemas grandes essa tarefa será computacionalmente custosa, mas como evitá-la? [1]. Nesse trabalho focaremos em uma das formas de resolver isso: *A decomposição QR*.

2. Fundamentação teórica

2.1. Matrizes ortogonais

Generalizaremos aqui o conceito de ortogonalidade entre vetores, definiremos a matriz ortogonal e listaremos suas propriedades, que é o grande foco da decomposição QR. As demonstrações dos teoremas e proposições podem ser encontradas em [2].

Definição 1 *Os vetores $q_1, q_2, \ldots, q_n$ são ditos ortogonais quando o produto interno $\langle q_i, q_j \rangle$ é zero sempre que $i \neq j$. Geometricamente, vetores ortogonais são perpendiculares dois a dois.*

Teorema 1 *Um conjunto de vetores ortogonais não nulos são linearmente independentes.*

Definição 2 *Os vetores $v_1, \ldots, v_n$ são ditos normais quando tem norma unitária. E são ditos ortonormais se são ortogonais e normais, ou seja:*

$$\langle q_i, q_j \rangle = \begin{cases} 1 & \text{se} \quad i = j \\ 0 & \text{se} \quad i \neq j \end{cases}$$

Uma matriz com os vetores coluna ortonormais é dita ortogonal e denotaremos-a por Q.

Definição 3 *A decomposição QR de uma matriz A é o par de matrizes Q e R, ortogonal e triangular superior, respectivamente, satisfazendo A = QR.*

Proposição 1 *A matriz Q satisfaz* $Q^tQ = I$.

Observação: Quando Q é uma matriz quadrada, Q é invertível se, e somente se $Q^{-1} = Q^t$.

Exemplo 1: (Rotações) Toda matriz de rotação é ortogonal. A matriz Q_θ que rotaciona um vetor no plano por um ângulo θ é:

$$Q_\theta = \begin{pmatrix} cos\theta & -sen\theta \\ sen\theta & cos\theta \end{pmatrix}$$

Logo, os vetores coluna são:

$$q_1 = \begin{pmatrix} cos\theta \\ sen\theta \end{pmatrix} q_2 = \begin{pmatrix} -sen\theta \\ cos\theta \end{pmatrix}$$

Assim, temos:

$$\begin{aligned} \langle q_1, q_1\rangle &= cos(\theta)^2 + sen(\theta)^2 = 1 \\ \langle q_2, q_2\rangle &= (-sen(\theta))^2 + cos(\theta)^2 = 1 \\ \langle q_1, q_2\rangle &= cos(\theta)(-sen(\theta)) + sen(\theta)cos(\theta) = 0 \\ \langle q_2, q_1\rangle &= (-sen(\theta))cos(\theta) + cos(\theta)sen(\theta) = 0 \end{aligned}$$

Portanto Q_θ é ortogonal.

Proposição 2 *Se Q é uma matriz ortogonal, então Q preserva a norma.*

Demonstração: De fato, se x é um vetor, pela proposição 1:

$$\|Qx\|^2 = \langle Qx, Qx\rangle = (Qx)^t(Qx) = x^tQ^tQx = x^tx = \langle x, x\rangle = \|x\|^2$$

Como $\|Qx\|, \|x\| \geqslant 0, então \|Qx\| = \|x\|$.

2.2. Fatoração QR por Gram-Schmidt

Podemos realizar a fatoração QR pelo algoritmo de Gram-Schmidt. O método consiste em utilizar projeções para ortogonalizar os vetores-coluna.

Seja $A \in \mathbb{R}^{n\times n}$ uma matriz com colunas linearmente independentes, isto é, os vetores coluna $a_1, a_2, \dots, a_n$ são linearmente independentes. Então, podemos utilizar o processo de ortogonalização de Gram-Schmidt no conjunto $B = \{a_1, \dots, a_n\}$, obtendo elementos não nulos e ortogonais $\hat{q}_k$, da forma:

$$\hat{q}_k = a_k - \sum_{j=1}^{k-1} r_{jk}q_j, \tag{3}$$

onde $r_{ik} = \langle v_k, q_i\rangle$, para i = 1, ..., k-1.
Além disso, obtemos vetores ortonormais q_k, normalizando os vetores $\hat{q}_k$, da seguinte forma:

$$q_k = \frac{\hat{q}_k}{r_{kk}}, \quad k = 1, \dots, n \tag{4}$$

onde $r_{kk} = \|\hat{q}_k\|_2 > 0$, pois os a_k são L.I.
Juntando as equações (3) e (4), obtemos:

$$a_k = \sum_{j=1}^{k-1} r_{jk} q_j + r_{kk} q_k, \qquad k = 1, \ldots, n$$

Ou então:

$$\begin{aligned} a_1 &= q_1 r_{11} \\ a_2 &= q_1 r_{12} + q_2 r_{22} \\ &\vdots \\ a_n &= q_1 r_{1n} + q_2 r_{2n} + \cdots + q_n r_{nn} \end{aligned}$$

Estas equações podem ser escritas como um produto matricial:

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix} \Leftrightarrow A = QR$$

Note que a matriz Q é ortogonal, pois suas colunas são vetores ortonormais e R é uma matriz triangular superior com diagonal positiva. Assim, obtemos a fatoração ortogonal $A = QR$ da matriz A, como no algoritmo abaixo.

Algoritmo 1: QR POR GRAM-SCHMIDT

Entrada: $A \in \mathbb{R}^{m \times n}$ $A = (a_1, a_2, \ldots, a_n); \mathtt{ntol} = 1e-12$
Saída: Q,R `Decomposição QR de A`

início
  $q_1 \leftarrow \dfrac{a_1}{\|a_1\|}$
  para *i de 2 à m* **faça**
    $w_i \leftarrow a_i - \sum_{j=1}^{i-1} \langle q_j, a_i \rangle q_j$
    se $\|w_i\| \leqq \mathtt{ntol}$ **então**
      Erro: Norma de w_i muito próxima de 0
    fim
    $q_i \leftarrow \dfrac{w_i}{\|w_i\|}$
  fim
  $Q \leftarrow (\ q_1, q_2, \ldots, q_n\ )$
  $R = Q^t A$
  retorna Q, R
fim

No exemplo abaixo, ilustraremos o algoritmo para uma matriz $A \in R^{3 \times 3}$

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 0 & -3 \\ 0 & -2 & 3 \end{pmatrix}$$

Assim, temos:

$$a_1 = (\ 1 \quad -1 \quad 0\ )\ a_2 = (\ 2 \quad 0 \quad -2\ )\ a_3 = (\ 3 \quad -3 \quad 3\ )$$

Logo:

$$
\begin{aligned}
q_1 &= \frac{a_1}{\|a_1\|} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 & 0 \end{pmatrix} \\
\Rightarrow w_2 &= a_2 - \langle q_1, a_2 \rangle q_1 = \begin{pmatrix} 1 & 1 & 2 \end{pmatrix} \\
\Rightarrow q_2 &= \frac{w_2}{\|w_2\|} = \frac{1}{\sqrt{6}}\begin{pmatrix} 1 & 1 & 2 \end{pmatrix} \\
\Rightarrow w_3 &= a_3 - \langle q_1, a_3 \rangle q_1 - \langle q_2, a_3 \rangle q_2 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\
\Rightarrow q_3 &= \frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}
\end{aligned}
$$

Portanto:

$$
Q = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{6} & 1/\sqrt{3} \\ -1/\sqrt{2} & 1/\sqrt{6} & 1/\sqrt{3} \\ 0 & -2/\sqrt{6} & 1/\sqrt{3} \end{pmatrix} \quad e \quad R = Q^t A = \begin{pmatrix} \sqrt{2} & \sqrt{2} & \sqrt{18} \\ 0 & \sqrt{6} & -\sqrt{6} \\ 0 & 0 & \sqrt{3} \end{pmatrix}
$$

2.3. Fatoração QR por Rotações de Givens

A fatoração QR também pode ser feita a partir de rotações de um ângulo θ no plano. Esse método consiste em usar os elementos da diagonal principal para zerar os elementos abaixo deles usando matrizes de rotação. Logo abaixo temos o algoritmo da fatoração QR por rotação de Givens:

Algoritmo 2: QR POR ROTAÇÕES DE GIVENS

Entrada: $A \in \mathbb{R}^{m \times n}$ $A = (a_1, a_2, \ldots, a_n)^t$
Saída: Q,R `Decomposição QR de A`

início
  $Q = (q_1, q_2, \ldots, q_n)^t =$ Identidade $n \times n$
  para *j de 1 à n* **faça**
    para *i de j+1 à m* **faça**
      se $a_{ij} \neq 0$ **então**
        $c_{ij} \leftarrow a_{jj} / \sqrt{a_{jj}^2 + a_{ij}^2}$ `Supondo` $a_{ij}, a_{jj} \neq 0$
        $s_{ij} \leftarrow a_{ij} / \sqrt{a_{jj}^2 + a_{ij}^2}$
        $a_j,\ a_i \leftarrow a_j c_{ij} + a_i s_{ij},\ \ a_j(-s_{ij}) + a_i c_{ij}$
        $q_j,\ q_i \leftarrow q_j c_{ij} + q_i s_{ij},\ \ q_j(-s_{ij}) + q_i c_{ij}$
      fim
    fim
  fim
  $Q \leftarrow Q^t$
  $R \leftarrow A$
  retorna *Q,R*
fim

Vejamos um exemplo para a matriz $A \in \mathbb{R}^{3\times 3}$

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 0 & 3 \\ 0 & -2 & 3 \end{pmatrix}$$

Temos:

$$a_1 = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} a_2 = \begin{pmatrix} -1 & 0 & -3 \end{pmatrix} a_3 = \begin{pmatrix} 0 & -2 & 3 \end{pmatrix}$$

$$c_{12} = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{12}^2}} = \frac{1}{\sqrt{2}}$$

$$s_{12} = \frac{a_{12}}{\sqrt{a_{11}^2 + a_{12}^2}} = \frac{-1}{\sqrt{2}}$$

$$\Rightarrow a_1 \leftarrow a_1 c_{12} + a_2 s_{12} = \frac{1}{\sqrt{2}}\begin{pmatrix} 2 & 2 & 6 \end{pmatrix}$$

$$\Rightarrow a_2 \leftarrow a_1(-s_{12}) + a_2 c_{12} = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 2 & 0 \end{pmatrix}$$

$$\Rightarrow q_1 \leftarrow q_1 c_{12} + q_2 s_{12} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 & 0 \end{pmatrix}$$

$$\Rightarrow q_2 \leftarrow q_1(-s_{12}) + q_2 c_{12} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 \end{pmatrix}$$

Como $a_{13} = 0$, seguimos para o próximo passo.

$$c_{23} = \frac{a_{22}}{\sqrt{a_{22}^2 + a_{23}^2}} = \frac{\sqrt{2}}{\sqrt{6}}$$

$$s_{23} = \frac{a_{23}}{\sqrt{a_{22}^2 + a_{23}^2}} = \frac{-2}{\sqrt{6}}$$

$$\Rightarrow a_2 \leftarrow a_2 c_{23} + a_3 s_{23} = \frac{1}{\sqrt{6}}\begin{pmatrix} 0 & 6 & -6 \end{pmatrix}$$

$$\Rightarrow a_3 \leftarrow a_2(-s_{23}) + a_3 c_{23} = \frac{1}{\sqrt{3}}\begin{pmatrix} 0 & 0 & 3 \end{pmatrix}$$

$$\Rightarrow q_2 \leftarrow q_2 c_{12} + q_3 s_{12} = \frac{1}{\sqrt{6}}\begin{pmatrix} 1 & 1 & -2 \end{pmatrix}$$

$$\Rightarrow q_3 \leftarrow q_2(-s_{12}) + q_3 c_{12} = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

Logo, como esperado:

$$Q = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{6} & 1/\sqrt{3} \\ -1/\sqrt{2} & 1/\sqrt{6} & 1/\sqrt{3} \\ 0 & -2/\sqrt{6} & 1/\sqrt{3} \end{pmatrix} \quad e \quad R = \begin{pmatrix} \sqrt{2} & \sqrt{2} & \sqrt{18} \\ 0 & \sqrt{6} & -\sqrt{6} \\ 0 & 0 & \sqrt{3} \end{pmatrix}$$

2.4. Problema de Quadrados Mínimos Lineares

Definição 4 *Seja* $A \in \mathbb{R}^{m\times n} (m > n)$ *e* $b \in \mathbb{R}^m$. *Considere* $r = b - Ax$. *O problema de quadrados mínimos de* (A, b) *é achar* $x \in \mathbb{R}^n$ *tal que* $\|r\|^2$ *seja mínima, ou seja,* $\min\limits_{x\in R^n} \|b - Ax\|_2^2$.

Teorema 2 *(Solução Única do Problema de Quadrados Mínimos) Seja A* $\in \mathbb{R}^{m\times n}$ *e b* $\in \mathbb{R}^m$, $m > n$ *tal que posto(A) = n (posto completo). Então o problema de quadrados mínimos tem solução única e pode ser encontrada resolvendo o sistema não singular* $\hat{R}x = \hat{c}$*, onde* $\begin{pmatrix} \hat{c} \\ d \end{pmatrix} = c = Q^T b$, $\hat{R}$ $\in \mathbb{R}^{n\times n}$ *é triangular superior e Q* $\in \mathbb{R}^{m\times n}$ *é ortogonal.*

De fato, se Q, R é a decomposição QR de A, pelas proposições 1 e 2:

$$\|Ax - b\|_2^2 = \|QRx - b\|_2^2 = \|Q^t(QRx - b)\|_2^2 = \|Rx - Q^t b\|_2^2$$

Considere $\hat{R} \in \mathbb{R}^{n\times n}$, $\hat{c} \in \mathbb{R}^{n\times n}$ e $d \in \mathbb{R}^{m-n\times n}$ tal que $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$ $Q^t b = \begin{pmatrix} \hat{c} \\ d \end{pmatrix}$. Temos:

$$\left\| \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} x - \begin{pmatrix} \hat{c} \\ d \end{pmatrix} \right\|_2^2 = \|\hat{R}x - \hat{c}\|_2^2 + \|d\|_2^2$$

Portanto, x é solução do problema de quadrados mínimos se,e somente se $\|\hat{R}x - \hat{c}\|_2^2 = 0$ com $\hat{R}$ triangular superior.

A resolução de quadrados mínimos é dada pelo algoritmo abaixo:

Algoritmo 3: RESOLUÇÃO DE QUADRADOS MÍNIMOS

Entrada: $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^{m\times 1}$
Saída: $\tilde{x} \in \mathbb{R}^{m\times 1}$ solução de $\min\limits_{x\in R^m} \|Ax–b\|_2^2$

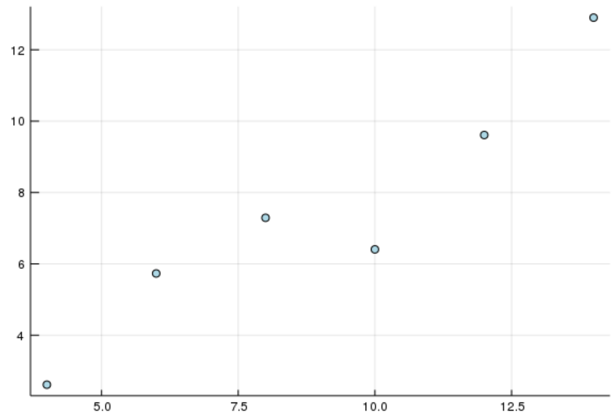
início
  $Q, R \leftarrow$ Decomposição QR de A
  $R_1 \leftarrow R[1 : m, :]$
  $b \leftarrow Q^t b$
  $b_1 \leftarrow b[1 : m]$
  $\tilde{x} \leftarrow R_1^{-1} b_1$
  retorna $\tilde{x}$
fim

Na prática, não é necessário calcular R_1^{-1}, pois é triangular superior. Basta encontrar x por retrosubstituição.

Voltemos ao exemplo da mola. Suponha que as medições realizadas foram:

$$\begin{aligned} 4.0k + p &= 2.614 \\ 6.0k + p &= 5.7309 \\ 8.0k + p &= 7.2909 \\ 10.0k + p &= 6.4058 \\ 12.0k + p &= 9.6112 \\ 14.0k + p &= 12.9009 \end{aligned}$$

Representando gráficamente temos:



Podemos escrever esse sistema em notação matricial como:

$$Ax = b$$

Onde,

$$A = \begin{pmatrix} 4.0 & 1.0 \\ 6.0 & 1.0 \\ 8.0 & 1.0 \\ 10.0 & 1.0 \\ 12.0 & 1.0 \\ 14.0 & 1.0 \end{pmatrix}; \quad x = \begin{pmatrix} k \\ p \end{pmatrix} \text{ e } b = \begin{pmatrix} 2.614 \\ 5.7309 \\ 7.2909 \\ 6.4058 \\ 9.6112 \\ 12.9009 \end{pmatrix}$$

Assim, basta resolver o problema de quadrados mínimos de (A,b). Aplicando o algoritmo de decomposição de decomposição QR por rotacões em A, obtemos:

$$Q = \begin{pmatrix} 0.169638 & 0.703585 & -0.408248 & 0.365148 & 0.316228 & 0.276026 \\ 0.254457 & 0.480094 & 0.816497 & -0.182574 & 0.000000 & 0.069007 \\ 0.339276 & 0.256602 & -0.408248 & -0.730297 & -0.316228 & -0.138013 \\ 0.424094 & 0.033110 & 0.000000 & 0.547723 & -0.632456 & -0.345033 \\ 0.508913 & -0.190382 & 0.000000 & 0.000000 & 0.632456 & -0.552052 \\ 0.593732 & -0.413874 & 0.000000 & 0.000000 & 0.000000 & 0.690066 \end{pmatrix}$$

$$R = \begin{pmatrix} -23.5797 & -2.2901 \\ 0.0000 & -0.8691 \\ 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{pmatrix}$$

$$\Rightarrow R_1 \leftarrow \begin{pmatrix} -23.5797 & -2.2901 \\ 0.0000 & -0.8691 \end{pmatrix}$$

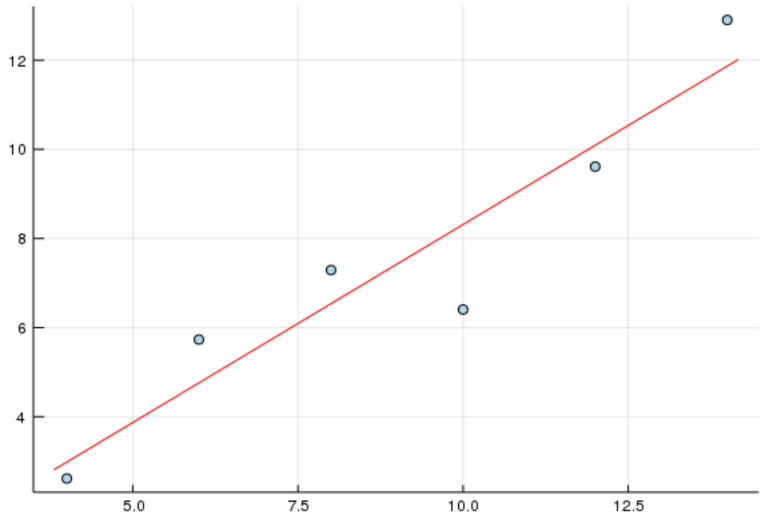
$$\Rightarrow b_1 \leftarrow Q^t b = \begin{pmatrix} 19.6429 \\ -0.4956 \\ 0.6356 \\ -1.9077 \\ 0.5483 \\ 1.4971 \end{pmatrix}$$

$$\Rightarrow \tilde{x} \leftarrow R^{-1} Q^t b = \begin{pmatrix} 0.8884 \\ -0.5703 \end{pmatrix}$$

Ou seja,

$$k = 0.8884 \text{ e } p = -0.5703$$

Graficamente:



Portanto, a melhor aproximação que temos é:

$$F = 0.0884x - 0.5703$$

3. Metodologia e Análise

Como vimos anteriormente, a fatoração QR pode ser feita de duas maneiras: usando o processo de Gram-Schmidt ou usando rotações de Givens. Veremos as vantagens e desvantagens de cada um dos métodos implementados em nosso código nessa seção (o código está disponível na **Seção 4**, nas considerações finais). Além disso, veremos em detalhes o problema de quadrados mínimos implementado pela equipe funcionando a partir da decomposição QR com cada um dos métodos implementados.

3.1. Fatoração QR

3.1.1. QR por Gram-Schmidt

O processo de Gram-Schmidt ortogonaliza os vetores coluna da matriz com projeções, obtendo Q, e a R surge como consequência. É instável numericamente, pois a ortogonalização feita no processo é propensa a erros numéricos. Uma vantagem significativa, no entanto, é a facilidade de implementação.

3.1.2. QR por rotação de Givens

Na fatoração QR por rotação, em cada nova iteração zeramos um elemento da matriz encontrada na iteração anterior; se estivermos na primeira iteração, zeramos um elemento da matriz A. Os elementos são zerados com o intuito de deixar a matriz no formato triangular superior, ou seja, encontramos nossa matriz R. Uma vantagem desse método é que cada novo elemento zero a_{ij} afeta apenas a linha com o elemento a ser zerado (i) e uma linha acima (j), o que torna o algoritmo mais eficiente.

3.2. Quadrados Mínimos

Nesta subseção, observaremos as principais diferenças entre a fatoração QR por Gram-Schmidt e por rotação de Givens na hora de calcularmos uma solução para o problema de quadrados mínimos.

3.2.1. Tempo e Memória

Utilizamos a biblioteca BenchmarkTools para calcular o tempo que o código leva para encontrar a solução de quadrados mínimos e ver a quantidade de memória que foi utilizada enquanto o código rodava. Essa biblioteca nos permite também ver o quanto de memória foi utilizada enquanto o código roda, além de nos mostrar o número de alocações. Observe os resultados obtidos em alguns exemplos numéricos:

Exemplo 1: Considerando as matrizes A e b abaixo:

$$A = \begin{pmatrix} 0.0 & 1.0 \\ 1.0 & 1.0 \\ 2.0 & 1.0 \\ 3.0 & 1.0 \\ 4.0 & 1.0 \end{pmatrix} \quad b = \begin{pmatrix} -5 \\ 0 \\ 5 \\ 12 \\ 24 \end{pmatrix}$$

Temos:

$$Q = \begin{pmatrix} 0.000000 & 0.774597 & -0.408248 & 0.365148 & -0.316228 \\ 0.182574 & 0.516398 & 0.816497 & -0.182574 & 0.000000 \\ 0.365148 & 0.258199 & -0.408248 & -0.730297 & -0.316228 \\ 0.547723 & 0.000000 & 0.000000 & 0.547723 & -0.632456 \\ 0.730297 & -0.258199 & 0.000000 & 0.000000 & 0.632456 \end{pmatrix} \quad e \quad R = \begin{pmatrix} 5.47723 & 1.82574 \\ 0.00000 & 1.29099 \\ 0.00000 & 0.00000 \\ 0.00000 & 0.00000 \\ 0.00000 & 0.00000 \end{pmatrix}$$

Solução do problema de quadrados mínimos: $x = \begin{pmatrix} 7.0 \\ -6.8 \end{pmatrix}$

Para chegar nas soluções acima, as alocações, memória e tempo foram os seguintes:

	Tempo	Alocações	Memória
Gram-Schmidt	0.0000009651 segundos	24 allocations	2.66 KiB
Rotação	0.0000052995 segundos	151 allocations	16.78 KiB

Neste exemplo, o processo de Gram-Schmidt é muito mais eficiente.
O Exemplo 1 é um exemplo qualquer, a matriz A não tem nada "especial", agora vejamos os exemplos a seguir.

Exemplo 2: (Matrizes esparsas) Utilizamos a biblioteca SparseArrays para criar uma matriz $A \in \mathbb{R}^{10\times 10}$ esparsa arbitrária e criamos um vetor $b \in \mathbb{R}^{10}$, também arbitrário. Obtivemos os seguintes dados:

	Tempo	Alocações	Memória
Gram-Schmidt	0.0000150070 segundos	351 allocations	57.81 KiB
Rotação	0.0000021864 segundos	30 allocations	8.02 KiB

Neste exemplo, ao contrário do exemplo 1, o método das rotações de Givens é mais eficiente.

Exemplo 3: (Matrizes densas) Criando uma matriz densa arbitrária $A \in \mathbb{R}^{10\times 10}$ e $b \in \mathbb{R}^{10}$ um vetor arbitrário , obtivemos:

	Tempo	Alocações	Memória
Gram-Schmidt	0.0000152320 segundos	351 allocations	57.81 KiB
Rotação	0.0000409430 segundos	910 allocations	145.52 KiB

Novamente, nesse exemplo, o método mais vantajoso é o processo de Gram-Schmidt.

3.2.2. Explicação dos resultados e conclusão

Após ver os 3 exemplos e comparar os resultados de tempo e memória utilizada para cada um deles, há uma pergunta relevante a ser respondida: Por que obtivemos esses resultados?

Nos exemplos 2 e 3 podemos ver uma diferença gigantesca entre os valores obtidos, principalmente se tratando de memória utilizada e alocações, sendo que no exemplo 2, a rotação é muito melhor que Gram-Schmidt e no exemplo 3 temos totalmente o contrário.

Isso acontece porque para matrizes esparsas o método da rotação é de fato muito mais eficiente, pois quando o aplicamos, procuramos zerar elementos da matriz para encontrar a matriz R no formato triangular superior. Se a matriz já tem vários zeros (esparsa), então não precisamos zerá-los, portanto, o algoritmo da rotação pulará diversos passos.

Já para matrizes densas, como no exemplo 3, não há zeros na matriz 10×10 (a densidade tem coeficiente 1), e portanto, para o método de rotação é necessário zerar diversas entradas dela, em vez de "pular"algumas, como acontece com matrizes esparsas. Por conta disso, o método de Gram-Schmidt se torna mais vantajoso, já que utiliza menos memória e realiza menos alocações comparado ao método da rotação de Givens.

Como no exemplo 1 pegamos uma matriz $A \in \mathbb{R}^{5\times 2}$ não esparsa e obtivemos um melhor desempenho do processo de Gram-Schmidt, então era de se esperar que para uma matriz densa 10×10 (exemplo 3) acontecesse o mesmo.

Concluímos que não há maneira mais eficiente de se calcular a fatoração QR, já que para o caso de matrizes esparsas, o preferível é utilizar das rotações de Givens e, para matrizes densas, o melhor é o processo de Gram-Schmidt.

3.2.3. Retas

Utilizamos a biblioteca Plots para gerar um gráfico com a reta que aproxima pontos dados. No Exemplo 1 da seção anterior, tínhamos as seguintes matrizes:

$$A = \begin{pmatrix} 0.0 & 1.0 \\ 1.0 & 1.0 \\ 2.0 & 1.0 \\ 3.0 & 1.0 \\ 4.0 & 1.0 \end{pmatrix} \quad b = \begin{pmatrix} -5 \\ 0 \\ 5 \\ 12 \\ 24 \end{pmatrix}$$

Sendo z um vetor de incógnitas $\quad z = \begin{pmatrix} c \\ d \end{pmatrix}$

Ao resolver $Az = b$, geramos o seguinte sistema linear:

$$\begin{aligned} 0c + d &= -5 \\ 1c + d &= 0 \\ 2c + d &= 5 \\ 3c + d &= 12 \\ 4c + d &= 24 \end{aligned}$$

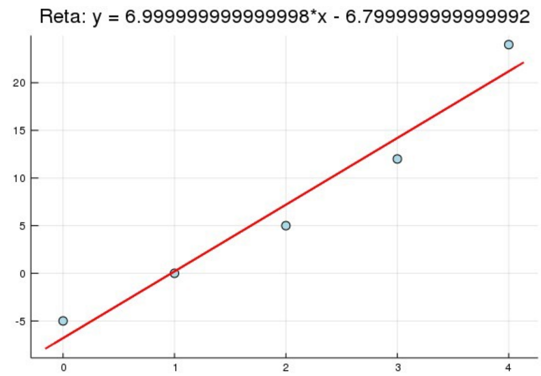
Cada uma dessas equações representa uma função afim. Na primeira equação, por exemplo, temos a função $y = cx + d$, sendo $x = 0$ e $y = -5$.

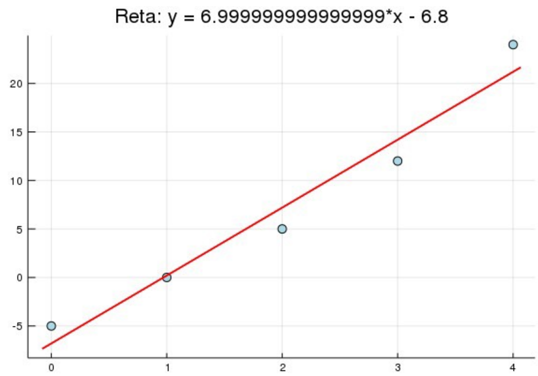
Como os coeficientes c e d são os mesmos para todas as equações, e nosso objetivo é achar o vetor $z = \begin{pmatrix} c \\ d \end{pmatrix}$, então resolver o sistema $Az = b$ é equivalente a achar a função afim de coeficiente angular c e linear d tal que a reta (gráfico dessa função) seja aquela que passa mais perto dos pontos: $(0, -5)$, $(1, 0)$, $(2, 5)$, $(3, 12)$, $(4, 24)$.

Portanto, dados: x um vetor que contém as abscissas dos pontos e y o vetor que contém as ordenadas, encontramos a equação da reta desejada simplesmente por resolver o sistema $Az = b$.

$$x = \begin{pmatrix} 0.0 \\ 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \end{pmatrix} \quad y = \begin{pmatrix} -5 \\ 0 \\ 5 \\ 12 \\ 24 \end{pmatrix} \quad \text{Solução obtida: } z = \begin{pmatrix} 7.0 \\ -6.8 \end{pmatrix}$$

E portanto, obtivemos a seguinte reta aproximada: $y = 7.0x - 6.8$. Abaixo estão os gráficos obtidos ao gerar a reta resolvendo o sistema $Az = b$ por Gram-Schmidt e por rotação de Givens, respectivamente.





Podemos perceber que ao construir a reta, não há diferenças muito significativas entre Gram-Schmidt e rotação de Givens, afinal, apesar de um método ser mais eficiente que outro, o resultado obtido é o mesmo.

4. Considerações finais

Por fim, segue link do diretório do Github contendo o código para avaliação: `https://github.com/dycksont/Fatoracao-QR-e-Quadrados-Minimos`

Referências

[1] ARAUJO, T.P., *Álgebra Linear: Teoria e Aplicações* Barbacena, Textos Universitários. Sociedade Brasileira de Matemática, 2014.

[2] STRANGE, G. *Introduction to Linear ALgebra* Wellesley, Cambridge Press, 2009.

[3] LIMA, E. L. , *Álgebra Linear* IMPA.

[4] LANG, S., *Álgebra Linear*, Springer, 1973

[5] WATKINS,D. S.*Introduction to Matrix Computation*, Wiley-Interscience, 2002.

[6] STEWART,G.W.,*Introduction to Matrix Computation*, Academic Press, 1973.

[7] AXLER, Sheldon, *Linear Algebra Done Right*, Springer, 2004.