Fatoração QR utilizando os métodos de Givens e Householder

Iann Takami Singo
Júlio César Fagundes
Luiza Gabriela da Silva
Victor Gabriel Zerger

[1]Departamento de Matemática (DMAT) – Universidade Federal do Paraná (UFPR)

Resumo. *Este trabalho têm como objetivos: fatorar uma matriz A no produto de outras duas, mais especificamente Q e R em Julia; mostrar dois métodos diferentes de obter estas matrizes, chamados Householder e Givens; mostrar vantagens e desvantagens de cada um deles e, explanar nossos acertos e erros desenvolvendo os códigos.*

1. Fatoração QR - o que é

Dada uma matriz $A \in \mathbb{R}^{n\times n}$, se suas colunas são linearmente independentes, podemos escrevê-la como o produto de duas matrizes. Essas matrizes, possuem propriedades e características específicas, as quais nos garantem que A pode ser escrita da seguinte forma: $A = Q \cdot R$, de modo que a matriz Q é ortogonal e R uma matriz triangular superior. Quando isso ocorre, temos o Teorema da *Fatoração QR* [1]. [2]

Essas definições nos dizem algumas coisas, como: Q é invertivel, logo seu determinante é diferente de 0, com isso, temos que sua inversa é igual a sua transposta ($Q^{-1} = Q^T$); em relação à R, todas as entradas abaixo da diagonal principal, são nulas, exemplo:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}.$$

A matriz Q ser ortogonal é importante pois conseguimos fazer algumas operações que nos dão a matriz R. Sabemos que $A = Q \cdot R$, se multiplicarmos pela esquerda ambos os lados da equação por Q^T, temos: $Q^T \cdot A = Q^T \cdot Q \cdot R$ Como $Q^T \cdot Q$ por definição, nos dá a matriz Identidade, ficamos com $Q^T \cdot A = R$. A recíproca também é verdadeira, como mostra o livro de Watkins [3].

Esta fatoração, abre portas para resolver o problema dos mínimos quadrados e serve como base para o algoritmo QR, que calcula autovalores e seus respectivos autovetores. Ela pode ser feita de algumas maneiras, neste trabalho abordaremos duas, o método de Householder e o método de Givens.

---

[1] `https://www.ime.unicamp.br/~marcia/AlgebraLinear/fat_ortogonal.html`

[2] Página 536 ARNOLD, J.T.; JOHNSON, L. W.; RIESS, R. D.. Introduction to Linear Algebra, 2002. Disponível em: `http://math.sjtu.edu.cn/faculty/victorm/[Lee_W._Johnson;_R._Dean_Riess;_Jimmy_T._Arnold]_I(z-lib.org).pdf`

[3] Watkins, D.S. Fundamentals of Matrix Computations, 2002.

2. Método de Householder

O método de fatoração de Householder foi proposto por Alston Scott Householder em 1958 e serve para decompor uma matriz A em outras duas matrizes Q e R, é, a partir de um dado vetor $v \in \mathbb{R}^n$, encontrar uma transformação de modo que neste vetor fique apenas a primeira entrada não nula. Para isso partimos do seguinte princípio, seja u um vetor que possui norma igual a 1 e I uma matriz identidade, o refletor de v pode ser escrito como: $Pv = v - 2 \cdot u \cdot u^T \cdot v$, de uma maneira melhor, P pode ser escrito como $P = I - 2 \cdot u \cdot u^T$. Este vetor ser refletor, nos diz que Pu é igual ao oposto de u, ou seja, ele será contrário a direção de $v \pm ||v|| \cdot e_1$.[4]

Com este vetor refletor conseguimos obter uma matriz refletora, que é dada por $H(v) = I - \dfrac{2 \cdot v^T}{v^T \cdot v}$, ela quem zera as entradas do vetor do jeito que queremos.[5] [6] Mas como aplicar isto? Basicamente multiplicamos muitas vezes as matrizes refletoras pela esquerda de A, com o intuito de ficarmos com uma matriz triangular superior. Como zeramos vetor por vetor, ou seja, coluna por coluna, devemos tomar muito cuidado para não desfazer ou anular o processo anterior, pois queremos que os zeros permaneçam.

Essas sucessivas multiplicações da matriz H(v) pela esquerda de A, ao final do processo nos dá a matriz Q^T e temos a seguinte equação: $Q^T \cdot A = R$. Para encontrar Q fica fácil, pois esta matriz tem a característica que sua inversa é igual a sua transposta, logo

$$Q^T \cdot A = R \implies Q \cdot Q^T \cdot A = Q \cdot R.$$

Para nosso código, utilizamos a demonstração do Teorema do livro Introduction to Linear Algebra de Lee W. Johnson, que está na página 520. Este Teorema nos mostra as propriedades da matriz Q, as quais já foram comentadas, e na sua prova temos duas equações: $Q = I - b \cdot u \cdot u^T$, e $b = \dfrac{2}{(u^T \cdot u)}$, elas são exatamente nossas matrizes refletoras H(v), porém escritas de uma maneira diferente.

Implementando este método em Julia:

```
using LinearAlgebra
function FatoraçãoHouseholder(A)
A = Float64.(A) #mudamos todas as entradas para Float
u = []
while k < n #passar de coluna em coluna  zerando as entradas
    ...
    b = 2/(u'*u) #u' significa vetor u transposto
    Q = I - b*u*u'
    R = Q'*A
    println("\nA matriz Q$(k-1)=$Q")
    ...
end
```

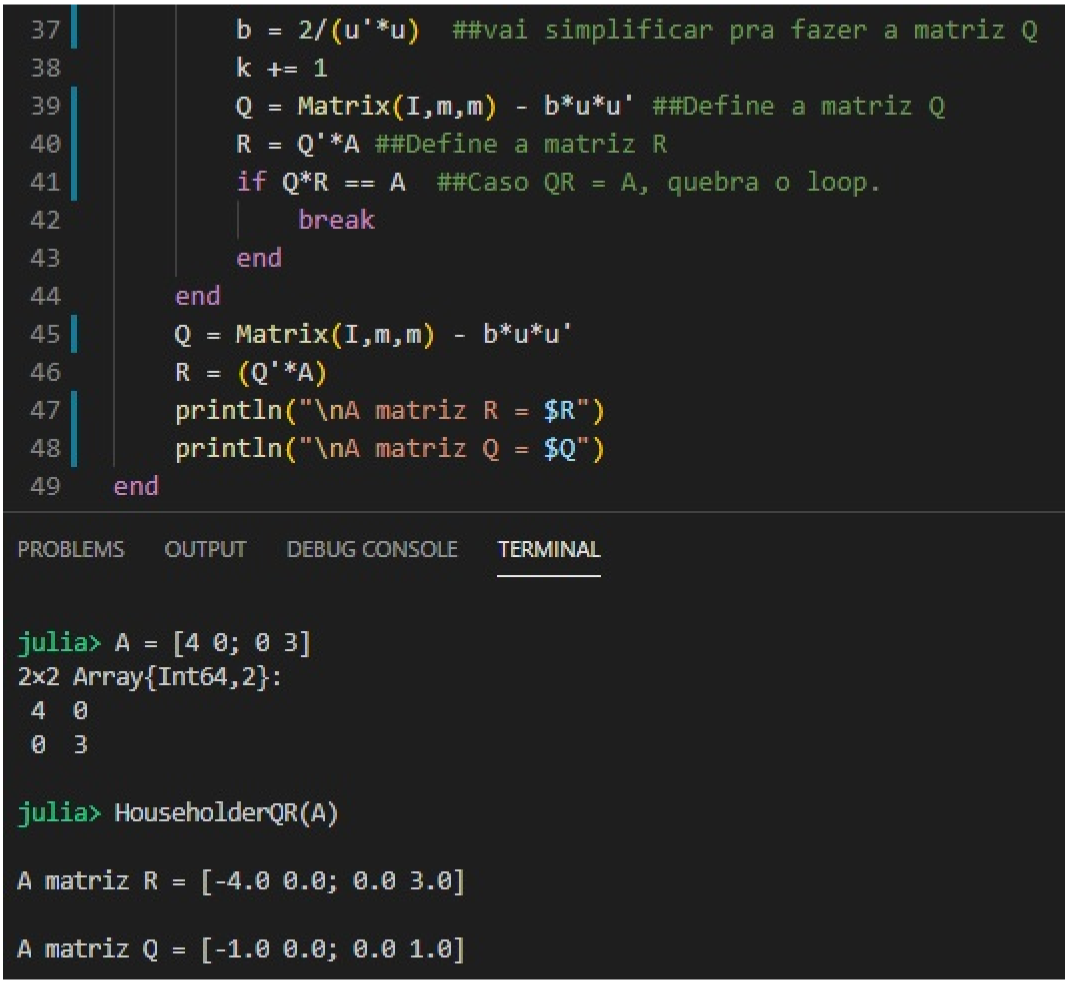
[4] https://www.ime.usp.br/~bruna/dump/labnum/fatoracaoQR.html

[5] https://www.ime.unicamp.br/~marcia/AlgebraLinear/fat_QR%28householder%29.html

[6] https://edisciplinas.usp.br/pluginfile.php/3020714/mod_folder/content/0/topico_6.pdf?forcedownload=1

Explicando um pouco o código: utilizamos o pacote LinearAlgebra para nos auxiliar com os comandos, uma vez que só trabalhamos com matrizes e vetores. Criamos uma função que ao final retorna todas as matrizes de Householder, ou seja, todas as ”Q’s”que foram encontradas pelo vetor u com a fórmula $Q = I - b \cdot u \cdot u^T$, onde $b = \dfrac{2}{(u^T \cdot u)}$ e multiplicadas à esquerda de A; a matriz R e o produto $Q \cdot R$. Para retornar de uma maneira mais elegante, fizemos o que está escrito na linha 10 em todos os casos, de forma que o \n pula uma linha. Assim, as matrizes não ficaram ”coladas”umas as outras, e conseguimos vizualizar por exemplo qual matriz é a A e qual é a Q1.

Exemplo da matriz $A = [4.0\ \ 0.0;\ \ 0.0\ \ 3.0]$ no VS Code com a função HouseholderQR(A):



Infelizmente tivemos um problema com sinal e não conseguimos resolver, mas a ideia está implementada.

Um ponto interessante deste método é que ele é muito mais estável por conta das suas reflexões e como suas entradas são zeradas. Sua desvantagem encontra-se na grande mudança que pode acontecer indesejadamente, uma vez que estamos trabalhando no vetor coluna inteiro da matriz, e a cada reflexão que fazemos podemos alterar tanto a matriz R, quanto a matriz Q absurdamente.[7]

[7] https://pt.wikipedia.org/wiki/Transforma%C3%A7%C3%A3o_de_Householder

3. Método de Givens

O método de Givens recebe este nome em homenagem à Wallace Givens, e foi criado em 1950 pelo mesmo autor. Este método é baseado em rotações no plano para anular um elemento de um vetor $v \in \mathbb{R}^n$.[8] Como o intuito do método de Givens também é fatorar a matriz $A \in \mathbb{R}^{n\times n}$ no produto de outras duas $(Q \cdot R)$, ele também possui uma matriz transformação para anular as entradas que queremos, a diferença dos métodos consiste em como fazer isso. Em Givens, mexemos apenas nas entradas, diferentemente de Householder que muda o vetor coluna inteiro.[9]

A matriz deste método chama-se Matriz das Rotações, e é muito parecida com a Matriz Identidade[10]. Seja $i < j$, temos que a Matriz de Rotação P_j^i, de ordem 2x2 pode ser escrita como:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

Onde c e s são dados por:

$$c = \frac{a_{i-1,j}}{\sqrt{a_{i-1,j}^2 + a_{i,j}^2}} \text{ e } s = \frac{a_{i,j}}{\sqrt{a_{i-1,j}^2 + a_{i,j}^2}},$$

No exemplo do vídeo do Canal Poujh, chamado ”Givens rotation for QR factorization (old, see description)”[11], conseguimos visualizar qual será a matriz de rotação. Se quisermos anular o valor da entrada a_{21}, fazemos: $c = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}$ e $s = \frac{a_{21}}{\sqrt{a_{21}^2 + a_{11}^2}}$, e a matriz P1 terá um formato como este:

$$\begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Assim, quando multiplicarmos por A pela esquerda, zeramos a entrada a_{21}.

Exemplo: Seja uma matriz A dada por

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 1 & 2 \end{bmatrix}$$

Se quisermos anular a entrada a_{21} precisar calcular c e s, logo:

$$c = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} = \frac{1}{\sqrt{1^2 + 2^2}} = \frac{1}{\sqrt{5}} = 0.447$$

21/01/2021 15:39

[8] `https://youtu.be/0wbvw8pJp7I` 20 de jan de 2021, às 18:40.

[9] `https://pt.wikipedia.org/wiki/Decomposi%C3%A7%C3%A3o_QR` 12 de jan de 2021, às 22:30

[10] `https://www.cos.ufrj.br/uploadfile/1368210590.pdf` pag 31

[11] `https://youtu.be/tFIsWKVS5M0` 20 de jan de 2021 hora: 20:00

e

$$s = \frac{a_{21}}{\sqrt{a_{21}^2 + a_{11}^2}} = \frac{2}{\sqrt{2^2+1^2}} = \frac{2}{\sqrt{5}} = 0.89$$

Com esses valores, basta substituir na matriz de rotação e acrescentar a linha e a coluna que faltam. Sabemos que P_j^i é dado por

$$P_j^i = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Substituindo c e s:

$$\rightarrow P_1^2 = \begin{bmatrix} 0.447 & 0.89 & 0 \\ -0.89 & 0.447 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fazendo a multiplicação desta matriz pela esquerda de A, temos:

$$\begin{pmatrix} 0.447 & 0.89 & 0 \\ -0.89 & 0.447 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 2 & & \\ 2 & 2 & 22 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 2.27 & 2.674 & 2.674 \\ 0.004 & -0.886 & -0.886 \\ 2 & 1 & 2 \end{pmatrix}$$

Podemos perceber que a entrada a_{21} é muito próxima de 0, sabemos que a máquina tem suas limitações e que aproximações já resolveriam nosso problema, por isso consideramos 0.004 como 0.

A Matriz de Rotação P_j^i é ortogonal, e se modificará a cada entrada anulada, com isso todas as "P's"serão ortogonais e a multiplicação de todas elas nos dará a matriz Q^T, como aconteceu no método anterior. Para Givens, foi utilizado o seguinte raciocínio para encontrarmos a matriz Q:

$$Q^T \cdot A = R$$
$$Q^T \cdot A \cdot A^{-1} = R \cdot A^{-1}$$
$$Q^T = R \cdot A^{-1}$$
$$Q = (R \cdot A^{-1})^T$$

Implementando esta ideia em Julia, com o princípio da ordem da Matriz A ser 2:

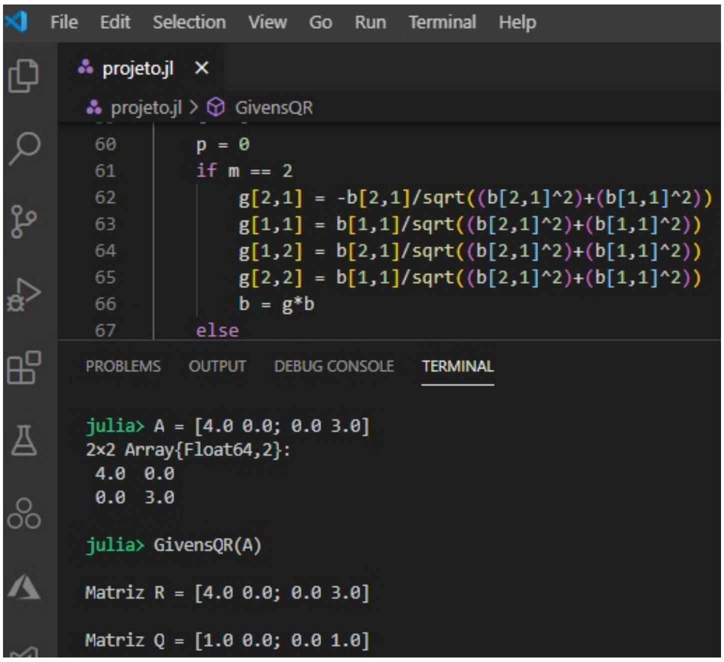
```
using LinearAlgebra
function GivensQR(a::Matrix)
(m,n) = size(a)
if m != n
    error("A matriz deve ser quadrada")
end
g = Matrix{Float64}(I,m,m)
...
if m == 2
    g[2,1] = -b[2,1]/sqrt((b[2,1]^2)+(b[1,1]^2))
    g[1,1] = b[1,1]/sqrt((b[2,1]^2)+(b[1,1]^2))
```

```
    g[1,2] = b[2,1]/sqrt((b[2,1]^2)+(b[1,1]^2))
    g[2,2] = b[1,1]/sqrt((b[2,1]^2)+(b[1,1]^2))
    b = g * b
else
...
```

Começamos com o pacote LinearAlgebra para nos auxiliar nos comandos relacionados à matrizes e em seguida criamos a função GivensQR, a qual deve retornar o valor de Q e R. Preferimos trabalhar somente com matrizes quadradas, por isso se ”m for diferente de n”deve dar erro, caso não seja diferente, mas a ordem seja igual a 2, temos uma mudança para cada entrada da matriz. Depois de fazer esta ordem separada, temos mais uma parte do código que é voltada para matrizes de ordem maior ou igual a três, que não iremos colocar aqui pois o princípio é o mesmo.

Para o código acima vamos dar um exemplo, dada uma matriz $A = [4.0\ \ 0.0;\ \ 0.0\ \ 3.0]$, observamos que é uma matriz quadrada de ordem 2, logo deve entrar no segundo if do nosso código. Fazendo todas as contas, o resultado sai exatamente como o esperado $Q = [1.0\ \ 0.0;\ \ 0.0\ \ 1.0]$ e a matriz R, que é a triangular superior, é igual a matriz A. Este exemplo e sua resposta, retiramos da aba ”Testes”do Exercício 5 - Gram Schmidt da matéria de Cálculo Numérico.

Segue imagem deste exemplo funcionando no VS Code com a linguagem de programação Julia:



O método de Givens é complicado de se entender e implementar, porém se feito corretamente é bem eficaz, uma vez que a cada novo elemento zerado somente duas linhas são afetadas, a que ele está inserido e a anterior.[12]

[12]https://pt.wikipedia.org/wiki/Decomposi%C3%A7%C3%A3o_QR#Vantagens_e_desvantagens_2 dia 21/01/2021 às 15:35

4. Conclusões

Durante todo o processo de criação deste projeto, a parte mais complicada foi conseguir entender os dois métodos, pois este tema é a junção de Álgebra Linear com Geometria Analítica e Transformações. A parte de Geometria Analítica não conseguimos compreender tão bem, pois como os métodos envolvem reflexão e rotação de vetores, acreditamos que a visualização é importante, mas infelizmente não conseguimos compreender a tempo, ficará para uma próxima oportunidade!

Referente a cada método, foi muito intrigante como matemáticos vermos que dois processos completamente diferentes levam a mesma resposta. Claro que cada um possui suas particularidades, o método de Givens por exemplo é complicado de implementar, e acreditamos que as melhores matrizes para ele são as quais não possuem muitas entradas para ser anuladas [13], ou seja, se estivermos trabalhando com uma matriz quadrada de ordem 3, na qual as entradas a_{21} e a_{31} precisem ser anuladas, o método de Givens será bem eficiente. Já em uma matriz quadrada de ordem 10, com muitas entradas a serem anuladas, preferimos o método de Householder.

A matéria de Cálculo Numérico foi uma experiência muito estressante porém que agregou demais a todos deste grupo, gostaríamos de agradecer aos Professores Abel Siqueira e Lucas Pedroso por terem nos ensinado tanto durante este período totalmente atípico que estamos passando. Obrigado!

5. Matrizes teste

Segue nossas matrizes teste para os códigos:

$$A_1 = \begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix};$$

$$A_2 = \begin{bmatrix} 12 & -51 & 4 \\ 6 & 167 & 5 \\ -4 & 24 & -41 \end{bmatrix};$$

$$A_3 = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix};$$

$$A_4 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix};$$

$$A_5 = \begin{bmatrix} 4 & 1 & 2 & 2 \\ 1 & 2 & 1 & 0 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & 2 & 1 \end{bmatrix}.$$

[13] `https://pt.wikipedia.org/wiki/Decomposi%C3%A7%C3%A3o_QR#Vantagens_e_desvantagens_2` dia 21/01/2021 às 15:35

6. Github

Segue nossos links do GitHub que contém este relatório, o slide da apresentação do dia 25/01/2021 e documento com os dois códigos aqui comentados.

Iann Takami Singo:

`https://github.com/iannts/Projeto-2`

Júlio César Fagundes:

`https://github.com/JulioCFagundes/Projeto-2`

Luiza Gabriela da Silva:

`https://github.com/Lu1zaGabriela/Projeto-2`

Victor Gabriel Zerguer:

`https://github.com/Victor-Zerger/Projeto-2`

7. Referências

[1] RUGGIERO, M. A. G.; VITORINO A. (2016). Álgebra Linear e Aplicações. Disponível em: https://www.ime.unicamp.br/~marcia/AlgebraLinear/index.html

[2] ARNOLD, J.T.; JOHNSON, L. W.; RIESS, R. D. (2002). Introduction to Linear Algebra Ed 5. Disponível em: http://math.sjtu.edu.cn/faculty/victorm/[Lee_W._Johnson;_R._Dean_Riess;_Jimmy_T._Arnold]_I(z-lib.org).pdf

[3] WATKINS, D.S. (2002). Fundamentals of Matrix Computations Ed 2. Disponível em: https://davidtabora.files.wordpress.com/2015/01/david_s-_watkins_fundamentals_of_matrix_computat.pdf

[4] THALENBERG, G. Fatoração QR. Disponível em: https://www.ime.usp.br/~bruna/dump/labnum/fatoracaoQR.html

[5] RUGGIERO, M. A. G.; VITORINO A. (2016). Álgebra Linear e Aplicações. Disponível em: https://www.ime.unicamp.br/~marcia/AlgebraLinear/fat_QR%28householder%29.html

[6] FABRIS, A. E. Refletores de Householder e Fatoração QR. Disponível em: https://edisciplinas.usp.br/pluginfile.php/3020714/mod_folder/content/0/topico_6.pdf?forcedownload=1

[7] Wikipedia. Transformação de Householder. Editada pela última vez às 01h39min de 5 de novembro de 2019. Disponível em: https://pt.wikipedia.org/wiki/Transforma%C3%A7%C3%A3o_de_Householder Acesso em: 21/01/2021 15:39

[8] Poujh. Givens Rotation Introduction. Disponível em: https://youtu.be/0wbvw8pJp7I Acesso em: 20 de jan de 2021, às 18:40.

[9] Wikipedia. Decomposição QR. Editada pela última vez às 19h47min de 31 de dezembro de 2020. Disponível em: https://pt.wikipedia.org/wiki/Decomposi%C3%A7%C3%A3o_QR Acesso em: 12 de jan de 2021, às 22:30.

[10] HUAMANI, M. C. N. (1984) Uso Da Fatorização QR na Estimação Dos Multiplicadores de Lagrange em Programação Não Linear Com Restrições Lineares. Disponível em: https://www.cos.ufrj.br/uploadfile/1368210590.pdf

[11] Poujh. Givens rotation for QR factorization (old, see description). Acesso em: 20 de jan de 2021, às 20:00. Disponível em: https://youtu.be/tFIsWKVS5M0

[12] Wikipedia. https://pt.wikipedia.org/wiki/Decomposi%C3%A7%C3%A3o_QR#Vantagens_e_desvantagens_2