

CMSE 823 – Numerical Linear Algebra
Homework 5
 Alexander Harnisch

Spring 2020

1.

We re-use the implementations from the previous homework, you can find them again in the file *qr.py*.

2.

For solving the over and well-determined systems we can use Algorithm 11.2. For under-determined systems we simply transpose the system which effectively converts it into an over-determined system. Everything is pretty much the same.

So instead of $Ax = b$ we solve $x^T A^T = b^T$ and perform the reduced QR decomposition on A^T instead. Now we have the solution:

$$x^T = b^T R^{-1} Q^T \quad (1)$$

or equivalently

$$x = QR^{-T}b = Qy \quad (2)$$

where we find y by forward substitution, since

$$R^T y = b \quad (3)$$

with R^T being lower triangular.

I don't know how much of this is going to be discussed in class, I don't have time to wait for it, it doesn't fit my schedule. I also could not find this explicitly in the textbook.

You can find my implementation of the forward and backward substitution in *solve.py*. Note that for small n it is actually faster to calculate the inner product of x and r_j (or l_j in the forward case) in each step when x has been initialized as a zero vector. I showed both ways to implement it by implementing forward and backward subsection in those two different ways.

(a) (b) and (c)

To get the results for the given matrices run *hw05_02.py*. As you can see, the results are correct using all different methods for obtaining the QR decomposition. The output should look as follows:

Part (a)

Solving $ax=b$ with $a =$

$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

$\begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$

$\begin{bmatrix} 7 & 8 & 7 \end{bmatrix}$

```
[4 2 3]
[4 2 2]]
and b =
[ 6. 15. 22.  9.  8.]
```

```
Classical Gram-Schmidt: x =
[1. 1. 1.]
```

```
Modified Gram-Schmidt: x =
[1. 1. 1.]
```

```
Householder Method: x =
[1. 1. 1.]
```

Part (b)

```
Solving ax=b with a =
[[0.7      0.70711]
 [0.70001 0.70711]]
and b =
[1.40711 1.40712]
```

```
Classical Gram-Schmidt: x =
[0.99999352 1.00000641]
```

```
Modified Gram-Schmidt: x =
[0.99999352 1.00000641]
```

```
Householder Method: x =
[1. 1.]
```

Part (c)

```
Solving ax=b with a =
[[1 2 3]
 [4 2 9]]
and b =
[ 6 15]
```

```
Classical Gram-Schmidt: x =
[0.42857143 0.85714286 1.28571429]
```

```
Modified Gram-Schmidt: x =
[0.42857143 0.85714286 1.28571429]
```

```
Householder Method: x =
[0.42857143 0.85714286 1.28571429]
```

3.

(a) and (b)

See the following handwritten page.

It's more convenient to start with 3. (b):

Originally we have the objective function

$$\|b - Ax\|_2^2 = (b - Ax)^* (b - Ax)$$

Which according to Theorem 11.1 is minimized if and only if

$$A^* Ax = A^* b,$$

We now add the L_2 normalization term to get the new objective function:

$$\begin{aligned} & \|b - Ax\|_2^2 + \alpha^2 \|x\|_2^2 \\ &= (b - Ax)^* (b - Ax) + \alpha^2 x^* x \\ &= (\tilde{b} - \tilde{A}x)^* (\tilde{b} - \tilde{A}x) \end{aligned}$$

$$\text{with } \tilde{A} = \begin{pmatrix} A \\ \alpha I \end{pmatrix}, \tilde{b} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

3. (a) Applying Theorem 11.1 to the new objective function immediately yields the new normal equations

$$\tilde{A}^* \tilde{A} x = \tilde{A}^* \tilde{b} = A^* b$$

$$\Rightarrow (A^* A + \alpha^2 I) x = A^* b$$

□

(c)

You can find the code in the file *hilbert.py*.