Homework #6
CS 575: Numerical Linear Algebra
Spring 2023

John Tran

March 30, 2023

Important Notes

- `Python` 3.11 was used to run the notebook (i.e., to use the `match` statement)
- the partially completed notebook was used to complete the coding assignment, so many of the things asked (e.g., verification and testing) was done for us and they were modified for the mat-mat portion
- the `README` was done in `Markdown` but the raw text can still be viewed

Problem 1

Note: the norm will refer to the 2-norm

We are asked to show CGS is the same as MGS in exact arithmetic. The psuedocode/algorithm will be assumed and referenced in this solution (covered in the lecture). The columns of Q of q_i are initially set to columns of A of a_i and all elements of R of r_{ij} are initially set to 0 where $i \in \{1, 2, dots, n\}$ and $j \in \{1, 2, dots, m\}$.

Also, the only difference between CGS and MGS is when solving for $r_{ij} = q_i^T a_j$ for CGS and $r_{ij} = q_i^T q_j$ for MGS, plugging in the appropriate expression for r_{ij} in solving for $q_j = q_j - r_{ij} q_i$ (same for both).

Base Case

$$
\begin{aligned}
q_1^{(CGS)} &= q_1^{(MGS)} \\
r_{11}^{(CGS)} &= r_{11}^{(MGS)}
\end{aligned}
$$

For $i = 1$, we only have 1 choice for q_1 which is in the direction of a_1, which is normalized and r_{11} is just the norm of a_1. This is the same for both CGS and MGS.

Inductive Case

We can got over some examples first for $j = 2, 3$ before moving into the general case.

For $j = 2$,

$$\begin{aligned} r_{12}^{(CGS)} &= q_1^T a_2 \\ q_2^{(CGS)} &= a_2 - r_{12}^{(CGS)} q_1 \end{aligned}$$

then normalize q_2

$$\begin{aligned} r_{12}^{(MGS)} &= q_1^T a_2 \quad q_2 \text{ initially set to } a_2 \\ q_2^{(MGS)} &= a_2 - r_{12}^{(CGS)} q_1 \end{aligned}$$

then normalize q_3

Both cases are the same for r_{12} and r_{22} (diagonal elements are just the norm of q_j at the end of the i loop when changes to q_j are complete, so they are skipped for brevity) and q_2.

Then for $j = 3$,

$$\begin{aligned} r_{13}^{(CGS)} &= q_1^T a_3 \\ r_{23}^{(CGS)} &= q_2^T a_3 \\ q_3^{(CGS)} &= a_3 - r_{13}^{(CGS)} q_1 - r_{23}^{(CGS)} q_2 \end{aligned}$$

then normalize q_3

The steps will be done one step at a time for MGS for clarity

$$\begin{aligned} r_{13}^{(MGS)} &= q_1^T a_3 \\ q_3^{(MGS)} &= a_3 - r_{13}^{(MGS)} q_1 \\ r_{23}^{(MGS)} &= q_2^T q_3 \\ &= q_2^T \left(a_3 - r_{13}^{(MGS)} q_1 \right) \\ &= q_2^T a_3 - q_2^T r_{13}^{(MGS)} q_1 \\ &= q_2^T a_3 - r_{13}^{(MGS)} q_2^T q_1 && r_{13}^{(MGS)} \text{ is a scalar, move to front} \\ &= q_2^T a_3 - 0 && \text{because } q_1, q_2 \text{ are orthogonal after } j = 1, 2 \text{ completed} \\ &= q_2^T a_3 \\ q_3^{(MGS)} &= q_3^{(MGS)} - r_{23}^{(MGS)} q_2 \\ &= a_3 - r_{13}^{(MGS)} q_1 - r_{23}^{(MGS)} q_2 \end{aligned}$$

then normalize q_3

So after doing $j = 2, 3$, we can start seeing the pattern that arises from doing MGS. The computation of the off-diagonal elements r_{ij} with the current value of q_j instead of a_j will produce a term $q_i^T a_j$ followed by a series of subtractions with terms involving dot products $q_i^T q_k$ where $k = 1, 2, \ldots, i-1$ for each ith iteration and that term reduces to 0 because of orthogonality.

The general case for MGS for each i iteration:

$$\begin{aligned}
q_j &= a_j && \text{initially} \\
r_{1j} &= q_1^T a_j \\
q_j &= q_j - r_{ij} q_i \\
q_j^{(1)} &= a_j - r_{1j} q_1 && \text{end of first iteration} \\
r_{2j} &= q_2^T (a_j - r_{1j} q_1) \\
&= q_2^T a_j \\
q_j^{(2)} &= a_j - r_{1j} q_1 - r_{2j} q_2 && \text{end of second iteration} \\
r_{3j} &= q_3^T (a_j - r_{1j} q_1 - r_{2j} q_2) \\
r_{ij} &= q_i^T (a_j - \sum_{k=1}^{i-1} r_{kj} q_k) \quad \text{unrolling recursion} \\
&= q_i^T a_j \\
q_j &= a_j - \sum_{i=1}^{j-1} r_{ij} q_i
\end{aligned}$$

So, the r_{ij} terms are the equal for both CGS and MGS because of reduced terms to 0 due to orthogonality, and the q_j terms are easy to check for equality since the formulation is the same across both (for MGS, accounting for reduced terms to 0 due to orthogonality for plugging in the r_{ij}).

Problem 2

(a) We are asked to show that the unique $y \in \mathbb{R}^n$ that minimizes $\|Qy - b\|_2$ is equal to $Q^T b$. Since we are given a minimization constraint, we can create the corresponding Normal Equations that would minimize y (Q and b are given):

$$Q^T Q y = Q^T b$$

Even though Q is not orthogonal, we have shown in the lecture that $Q^T Q = I$ where $Q \in \mathbb{R}^{m \times n}$ and $m > n$. This is simple to show: Q is reduced along the columns (i.e., some columns have been removed from the full QR decomposition) but the columns still maintain their respective dot products where $q_i \cdot q_i = 1$ as normalized columns vectors for $i = \{1, 2, \ldots, n\}$. So, $Q^T Q$ is just the dot product of the (remaining) orthonormal columnvectors of Q to produce 1s along the diagonal and 0s elsewhere, resulting in the identity matrix I of size $n \times n$.

Also, we know that $\|Qy - b\|_2 \geq 0$ from the property of norms, and a simple case to minimize this is when the equality case holds and we have indeed found above that $Q^T Q y = I y = y = Q^T b$.

(b) We want to show that the least-squares solution of x of $Ax = b$ is the solution of

$$Rx = Q^T b$$

Again, we can use the fact that saying x is a least squares solution is the same as x is a solution to the corresponding normal equations for $Ax = b$, minimizing for x:

$$A^T A x = A^T b$$

If we plug in $A = QR$,

$$\begin{aligned}
(QR)^T QRx &= (QR)^T b \\
R^T Q^T QRx &= R^T Q^T b \\
R^T IRx &= R^T Q^T b \quad \text{from part (a)} \\
R^T Rx &= R^T Q^T b \\
Rx &= Q^T b
\end{aligned}$$

where the last step holds if we let R^T, and thus R, be invertible. This is satisfied because $A \in \mathbb{R}^{m \times n}$ and A has rank n, which means A is full rank and by definition, A is thus invertible. We have also shown in the lecture if A is invertible, then R is also invertible.

Alternate Solution

We could use our result from part (a) instead where we found $y = Q^T b$ and looking at $Ax = QRx = b$, we have the original minimizer $\|QRx - b\|_2$. We also have the minimizer $\|Qy - b\|_2 \geq \|QRx - b\|_2$, but for y to also minimize the constraint, which we have shown above in part (a), the equality case must hold and $y = Rx$. So, we assume R is invertible again (same reasoning as the original solution) and $x = R^{-1}y$ and plug this in for x in $Rx = Q^T b$ since we have shown $\|Qy - b\|_2 = \|QRx - b\|_2 = \|Ax - b\|_2$ where y extends the solution of $Ax = b$:

$$\begin{aligned} Rx &= R(R^{-1}y) \\ &= y \\ &= Q^T b \end{aligned}$$

satisfying $Rx = Q^T b$

Problem 3

Note: the `Jupyter Notebook` template was used for this question and the CGS code was already completed – please see the code for more details (since the pseudocode of all three algorithms were also given, the implementation was rather simple and one-for-one with the pseudocode)

(a) Since the CGS algorithm was given, we assume the implementation was correct and we just need to make sure the two versions of MGS produced the same Q and R matrix results as CGS.

```
CGS
Q= [[ 0.16903085  0.89708523]
 [ 0.50709255  0.27602622]
 [ 0.84515425 -0.34503278]]
R= [[5.91607978 7.43735744]
 [0.         0.82807867]]
MGS-ver1
Q= [[ 0.16903085  0.89708523]
 [ 0.50709255  0.27602622]
 [ 0.84515425 -0.34503278]]
R= [[5.91607978 7.43735744]
 [0.         0.82807867]]
MGS-ver2
Q= [[ 0.16903085  0.89708523]
 [ 0.50709255  0.27602622]
 [ 0.84515425 -0.34503278]]
R= [[5.91607978 7.43735744]
 [0.         0.82807867]]
```

Figure 1: The QR decomposition of the given matrix

As we can see, all three algorithms produced the exact same QR decomposition.

(b) We are asked to compute $\|A - QR\|_2$ and $\left\|Q^T Q - I\right\|_2$ for all three of the algorithms.

```
CGS: norm_of_A_minus_QR =  6.600900888885555e-14 norm_of_QTQ_minus_I =  0.00036884231268251733
MGS-ver1: norm_of_A_minus_QR =  6.429516872878607e-14 norm_of_QTQ_minus_I =  3.4243306240089034e-10
MGS-ver2: norm_of_A_minus_QR =  6.429516872878607e-14 norm_of_QTQ_minus_I =  3.4243306240089034e-10
```

Figure 2: The results of the calculating the norms of the differences for all three algorithms

The results here show us that all three algorithms have approximately the same error in terms of the QR decomposition itself for $\|A - QR\|_2$. However, checking the orthogonality of the resulting Q matrix was quite different between CGS and both versions of MGS. For CGS, the norm of the error between $Q^T Q$ (which should be I in exact arithmetic with orthonormal columns) and I was relatively high compared to the MGS algorithms. This helps to confirm that CGS is not numerically stable in calculating the orthonormal vectors of Q. However, the modifications made in both MGS versions produced the exact same results with relatively low errors in finding the orthonormal vectors – equivalent in terms of floating point operations as we have discussed in the lecture.