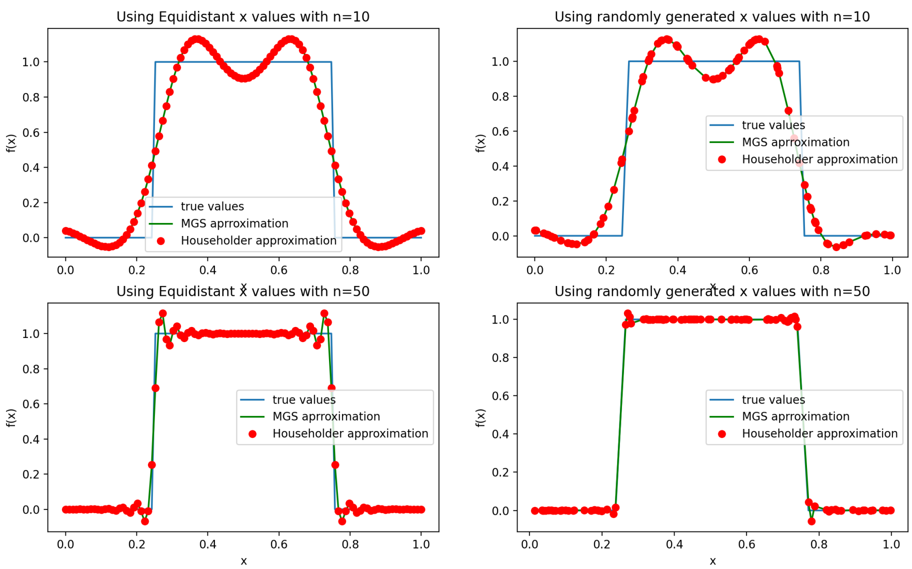
Computational Linear Algebra CW1

CID 01495449

1 Answer to Question 1b



From the Figures above, we can see that both the Gram-Schmidt method and HouseHolder Method give approximations that are very close to each other in all cases.

50 basis functions produce better approximations than 10 basis functions. This is because increasing the number of basis increases the degree of freedom for the approximations to vary, hence higher number of basis functions can approximate a jump discontinuity in a tighter range of points.

The approximations are better for randomly generated points than for the equidistant points as the earlier generates fewer points closer to the jump in the true function.

2 Answer to Question 2a

QR decomposing a m×n matrix A_1, with m n, gives the product of an m×m unitary matrix Q and an m×n upper triangular matrix R. As the bottom (mn) rows of an m×n upper triangular matrix consist entirely of zeroes, so we can partition both R and Q as :

$$A_1 = QR = [Q1, Q2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1$$

Here R_1 is an n×n upper triangular matrix, 0 is an (m n)×n zero matrix, Q_1 is m×n, Q_2 is m×(m n), and Q1 and Q_2 both have orthogonal columns.

Q_1 has full column rank due to their orthogonality hence $rank(A) = rank(Q_1 R_1) = rank(R_1)$

3 Answer to Question 2e

Using the function "reduced_tol_calc_rank" to calculate the rank of the matrix for different tolerances. We get the below results.

For randomly generated matrix with dimensions 40 x 10

Tolerance	Rank Approximation	Rank approximation correct?
100	0	False
10	2	False
1	4	True
1e-1	4	True
1e-2	4	True
1e-4	4	True
1e-6	4	True
1e-8	4	True

For matrix generated matrix with dimensions 50 x 50

Tolerance	Rank Approximation	Rank approximation correct?
100	0	False
10	1	False
1	4	False
1e-1	4	False
1e-2	4	False
1e-4	4	False
1e-6	6	True
1e-8	6	True

For randomly generated matrix with dimensions 60 x 60

Tolerance	Rank Approximation	Rank approximation correct?
100	0	False
10	48	False
1	57	True
1e-1	57	True
1e-2	57	True
1e-4	57	True
1e-6	57	True
1e-8	57	True

From the charts above we can see that as the tolerance decreases, the rank approximation function becomes more accurate.

If the tolerance is too high, the rank approximation is 0 because the householder function starts setting nonzero norm columns as zero columns because their norms will be below the high tolerance value.

We can also see from the above charts that the rank approximation function is more accurate for bigger matrices. This is because with more entries the norms are more likely to exceed the tolerance hence decreasing the chance of the householder function setting a non-zero column to a zero column.

4 Answer to Question 2f

The more efficient algorithm I propose starts by calculating the norms of the columns of A squared. From this norm vector, we can first select the largest norm and pivot A along with the vector of the norms. Then we do the first Householder iteration. We transform our columns by unitary matrices which preserves the norms of the columns.

For the following iterations, we are only interested in the norms of the columns of the submatrix as we have already processed some rows. We now need to subtract the norms of A squared from the original A squared matrix and choose the new maximum norm value to pivot along.

5 Answer to Question 3a

Let P be equal to

$$\begin{bmatrix} 0 & \cdots & 0 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 1 & 0 & 0 \\ \vdots & ⋰ & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Hence P is a permutation matrix that switches the rows. Let $\hat{A} = PA$. Now we compute the QR decomposition of $\hat{A}^T = \hat{Q}\hat{R}$. In the algorithm, we are given that $Q = P\hat{Q}^T$ and $R = P(P\hat{R}^T)$. Using this we can show $A = RQ$ as below.

$$
\begin{align}
RQ &= P(P\hat{R})^T P\hat{Q}^T, \tag{1}\\
&= P\hat{R})^T P^T P\hat{Q}^T, \tag{2}\\
&= P\hat{R}^T\hat{Q}^T (\text{since } P^T P = I), \tag{3}\\
&= P(\hat{A}^T)^T, \tag{4}\\
&= P\hat{A}, \tag{5}\\
&= P\hat{A}, \tag{6}\\
&= PPA, \tag{7}\\
&= A \tag{8}
\end{align}
$$

Now to show that Q is orthogonal.

$$
\begin{align}
Q^T Q &= (P\hat{Q}^T)^T P\hat{Q}^T, \tag{9}\\
&= \hat{Q}P^T P\hat{Q}^T, \tag{10}\\
&= \hat{Q}\hat{Q}^T (\text{—since } P^T P = I), \tag{11}\\
&= (\hat{Q}^T\hat{Q})^T \text{—}(\hat{Q} \text{ is orthogonal}), \tag{12}\\
&= I \tag{13}
\end{align}
$$

We know $R = P(P\hat{R}^T)$, which is trivially upper triangular.

6 Answer to Question 3c

Given a matrix nxp B, using QR decomposition we can write $B = QS$. Where Q is an orthogonal matrix and S is an upper triangular matrix. Hence $Q^T B = S$. Q is a nxn matrix and S is nxp matrix.

We can now carry out the RQ decomposition of $Q^T B$ such that $Q^T A = RQ_*$. Where R is upper triangular nxn and Q_* is a mxm orthogonal matrix. We can set $U = Q_*^T$ and rearranging we have $Q^T AU = R$ as needed.

Since S is an upper triangular matrix from QR decomposition of B, using the reasoning in answer to question 2a above:

$$S = \begin{bmatrix} S_{11} \\ 0 \end{bmatrix}$$

Where S_{11} is a pxp matrix

We have

$$
\begin{aligned}
R &= P(P\hat{R})^T, && (14)\\
&= P(P\begin{bmatrix}\hat{R}_{11}\\ 0\end{bmatrix})^T, && (15)\\
&= P\begin{bmatrix}0\\ \hat{R}_{11}\end{bmatrix}^T, && (16)\\
&= P[0, \hat{R}_{11}], && (17)\\
&= [0, P\hat{R}_{11}], && (18)\\
&= [0, R_{11}](\text{where we have set } R_{11} = P\hat{R}_{11}]) && (19)
\end{aligned}
$$

Shown as needed.