McGill University

Matrix Computations

COMP 540

Assignment 5

Name: Doug Shi-Dong

Student ID: 260466662

November 21, 2015

Question 1

Part 1

Letting $a_1(t) = 1$ and $a_2(t) = t$, the single precision condition number of A is $\kappa_2(A) =$ 8.6798010E+02. The norm of the residual of the least-square solution by the QR factorization method is $\|b - Ax_{qr}\|_2 =$ 8.0615806E-05.

The condition number of the tranpose of A with itself is $\kappa_2(A^T A) =$ 7.5613331E+05. The norm of the residual of the least-square solution by the normal equation is $\|b - Ax_{ne}\|_2 =$ 1.1265145E-04.

Using double precision for the A and the b matrix allows to compare the previous solutions with the true solution. All the solutions have been compiled in Table 1.

	Single Precision QR	Single Precision NE	Double Precision QR
x_1	1.0167466	1.028004	1.016737195131018
x_2	0.9849972	0.9748591	0.985005503165425

Table 1: Results for $a_1(t) = 1$ and $a_2(t) = t$

The single precision QR factorization is expected to give a better least-square solution than solving the normal equations. The condition number of $A^T A$ is three orders of magnitude bigger than the condition number of A. The QR factorization allows to retrieve up to 4 digits of accuracy, whereas the normal equation only retrieves 2 digits of the real solution.

Part 2

Now, let $a_1(t) = 1$ and $a_2(t) = (t - 1.11) * 100$. The single precision condition number of A is $\kappa_2(A) = 4.5301147$. The norm of the residual of the least-square solution by the QR factorization method is $\|b - Ax_{qr}\|_2 =$ 8.0615806E-05.

The condition number of the tranpose of A with itself is $\kappa_2(A^T A) = 20.5219326$. The norm of the residual of the least-square solution by the normal equation is $\|b - Ax_{ne}\|_2 =$ 8.0527971E-05.

The single precision and double precision results have been compiled in Table 2.

	Single Precision QR	Single Precision NE	Double Precision QR
x_1	2.1100941	2.1100929	2.110093286276619
x_2	0.0098492	0.0098508	0.009850104587809

Table 2: Results for $a_1(t) = 1$ and $a_2(t) = (t - 1.11) * 100$

The change of basis has improved the condition number of both A and $A^T A$ by a few orders of magnitude. Therefore, both x_{qr} and x_{ne} are much closer to the real solution x_{qr}. Note that the resulting x_2 in this new basis is just a multiple of the x_2 from the first basis.

Question 2

Handwritten derivations are attached.

Question 3

Singular Value Decomposition

The singular values and vectors of B are shown below.

$$\lambda_B = \{-0.026463896961605, 2.404909433839755, -35.134394975254480\}$$

$$U_B = \begin{bmatrix} 0.6482 & 0.6737 & 0.3550 & -0.0000 & 0.0000 \\ 0.6237 & -0.3900 & -0.3986 & -0.5140 & 0.1893 \\ -0.3149 & 0.0696 & 0.4427 & -0.5442 & 0.6355 \\ 0.0060 & 0.2508 & -0.4869 & 0.4535 & 0.7031 \\ 0.3028 & -0.5712 & 0.5310 & 0.4838 & 0.2569 \end{bmatrix}$$

$$V_B = \begin{bmatrix} 0.4183 & 0.8854 & 0.2026 \\ -0.8137 & 0.2662 & 0.5167 \\ 0.4036 & -0.3810 & 0.8318 \end{bmatrix}$$

Stability

In order to test for stability, every entry of B has been perturbed by 1E-05. The error of the singular value and vectors are given by $\|S_{pert} - S_B\|_2 = 6.8766\text{E-}06$, $\|U_{pert} - U_B\|_2 = 1.9715\text{E-}05$, and $\|V_{pert} - V_B\|_2 = 1.0605\text{E-}07$. This test does not prove stability of the algorithm since only mathematical derivations can, but it does support that the algorithm is *probably* numerically stable since a small error in the input lead to small error in the solution.

Moore-Penrose Matrix and Rank

The SVD of A computed by MATLAB gives the following three singular values:

$$\lambda_A = \{35.1272233335747, 2.46539669691652, 2.57621344955340\text{E-}16\}$$

This shows that the A matrix numerically has rank 3. However, it is easy to show that the columns of A are not linearly independent. Therefore, matrix A mathematically has a rank of 2.

To test the validity of the Moore-Penrose matrix, the following properties are checked:

$$\|AGA - A\| = 0 \quad \|GAG - G\| = 0 \quad \|(AG)^T - AG\| = 0 \quad \|(GA)^T - GA\| = 0$$

If we decide that the matrix 3 has full column rank, it is obvious that Σ_1^{-1} blows up due to the small singular value. Therefore, none of the above test are satisfied and the resulting Moore-Penrose matrix is invalid.

Instead, if we say that A has rank 2 instead of 3 (by removing a column of the A matrix), the Moore-Penrose matrix given by $G = V_1 \Sigma_1^{-1} U_1^T$ satisfies the above tests. Moreover, the `pinv` MATLAB command confirms the results obtained.

$$G = \begin{bmatrix} 0.2467 & -0.1333 & 0.0200 & 0.0933 & -0.2067 \\ 0.0667 & -0.0333 & -0.0000 & 0.0333 & -0.0667 \\ -0.1133 & 0.0667 & -0.0200 & -0.0267 & 0.0733 \end{bmatrix}$$

The minimum 2-norm LS solution is given by $\hat{x} = Gb$. If rank 2 is chosen, $x_{LS2} = [0.74, 0.20, -0.34]^T$ is the solution to the LS problem. However, if rank 3 is chosen, then the solution blows up to a huge number $x_{LS3} = [-3.64, 7.28, -3.64]^T * 10\text{E}{+}15$ due to the bad Moore-Penrose matrix obtained.

With the MATLAB built-in solvers, it is the equivalent of comparing `pinv(A)*b` and `A b`. `pinv` finds the same Moore-Penrose matrix found earlier by assuming rank 2. The backslash solver however finds the solution $x_{LS2} = [0.84, 0, -0.24]^T$ and notifies the user of rank deficiency. Therefore, we see that the choice of the matrix rank can give different computational results.

Codes

All codes are available on my GitHub:

`https://github.com/dougshidong/comp540/tree/master/a5`

More specifically `q1.m` corresponds to Question 1, `jacrot.m` to Question 3a, `SVDKog.m` to Question 3b and `q3.m` to the rest of everything else in Question 3.