MAT 167 WQ 2022 CA 02 Report

Randy Li

Sun 3rd Mar, 2024

Contents

1 Introduction and Motivation

For linear algebra calculation, usually, we will input the data into the computer. However, the number of bits in the computer is limited, which causes loss of precision of the result. In this project, we will discuss about least square problem and compare the accuracy of different algorithms including, normal equations, classical Gram-Schmidt algorithm CGS, modified Gram-Schmidt algorithm MGS, and QR factorization. In addition, we will also use the MATLAB operator algorithm to see how matlab decide the algorithm to increase the precision.

2 Problem Description

The least square problem is to find the coefficient of polynomial equation of $p(x) = c_0 + c_1x^1 + c_2\,x^2 + \ \dots\ c_{n-1}\,x^{n-1}$ that best fit the points in R^n.

3 Results (Table)

Put your table here with a short explanation of what the numbers (data) in each column of the table represent. If you have written the polynomial $p(x) = c_0 + c_1x^1 + c_2\,x^2 + \ \dots\ c_{n-1}\,x^{n-1}$ with coefficients $c_0, c_1, c_2, \ \dots\ c_{n-1}$ in the previous section you can simply refer to it. (It is best to set an equation 'label' where you wrote the coefficients out and refer to that label, but you may simply write "... the coefficients ... described in §2."

Table 1: Coefficients of the best 11th degree LS polynomial approximation to $\cos(6\,x)$ computed with six different algorithms

Normal Equations	CGS	MGS	QR	SVD	MATLAB \ operator
1.0000002714951470	1.0002365388729177	1.0000003169738811	1.0000003170422349	1.0000003170422309	1.0000003170422538
-0.0001317530732383	-0.0391066181047989	-0.0001438722076680	-0.0001438869161337	-0.0001438869160993	-0.0001438869182085
-17.9937877214351651	-16.9718745764513663	-17.9933513232051574	-17.9933508474642707	-17.9933508474647184	-17.9933508474090402
-0.1122781423654296	-10.7280507975213446	-0.1185087297139510	-0.1185149454861571	-0.1185149454813931	-0.1185149461304639
55.0468941736848976	111.7153464808626779	55.0933882365252146	55.0934311240109196	55.0934311239819294	55.0934311281521119
-5.7452414920039088	-180.1919609052001476	-5.9506091435795607	-5.9507855970324597	-5.9507855969225663	-5.9507856133029593
-44.9654427902716094	281.3184357131822253	-44.3937470813682040	-44.3932868994403549	-44.3932868997156405	-44.3932868582865510
-44.3161461141112127	-419.4701033225869082	-45.3466232528341280	-45.3474040188019245	-45.3474040183372011	-45.3474040872223156
105.3901004876502867	363.4446262558130343	106.5908342272567921	106.5916939911300432	106.5916939906051937	106.5916940654958154
-55.7320365923660788	-153.3814801391906144	-56.6050997559688440	-56.6056925728579969	-56.6056925724769968	-56.6056926237393156
7.2576169674317876	23.7351409695566353	7.6177635189758703	7.6179961150148827	7.6179961148540229	7.6179961349055638
1.1306233353674253	0.5287448320703061	1.0662674712502709	1.0662278329087460	1.0662278329387838	1.0662278295206449

4 Analysis of the Results

4.1 The Computation of $\hat{\boldsymbol{x}}_{\text{normal}}$

The computation of $\hat{\boldsymbol{x}}_{\text{normal}}$ uses MATLAB's backslash operator to solve the normal equations.

$$\hat{\boldsymbol{x}}_{\text{normal}} = \begin{pmatrix} 1.0000 \\ -0.0001 \\ -17.9938 \\ -0.1123 \\ 55.0469 \\ -5.7452 \\ -44.9654 \\ -44.3161 \\ 105.3901 \\ -55.7320 \\ 7.2576 \\ 1.1306 \end{pmatrix}$$

The accuracy is the second least precise beacuse it requires to calculate $A^T A$.

4.2 The Computation of $\hat{\boldsymbol{x}}_{\text{cgs}}$

It uses a custom function of CGS. $\hat{\boldsymbol{x}}_{\text{cgs}} = \begin{pmatrix} 1.0002 \\ -0.0391 \\ -16.9719 \\ -10.7281 \\ 111.7153 \\ -180.1920 \\ 281.3184 \\ -419.4701 \\ 363.4446 \\ -153.3815 \\ 23.7351 \\ 0.5287 \end{pmatrix}$
The result of this one is most inaccurate because it requres large calculation of matrix multiplication

4.3 The Computation of $\hat{\boldsymbol{x}}_{\text{mgs}}$

(10 points) Give a brief but thoughtful description of what you observe with regards to the accuracy of your results when you used the supplied MATLAB function `CGS` to solve the original Least Squares system (**??**).

Compare these results with the results from the previous sections. Mention your thoughts concerning the (theoretical) reason the results are as they are.

4.4 The Computation of $\hat{\boldsymbol{x}}_{\text{qr}}$

(10 points) Give a brief but thoughtful description of what you observe with regards to the accuracy of your results when you used MATLAB's builtin function `qr` to solve the original Least Squares system (**??**).

Compare these results with the results from the previous sections. Mention your thoughts concerning the (theoretical) reason the results are as they are.

4.5 The Computation of $\hat{\boldsymbol{x}}_{\text{svd}}$

SVD method divide the matrix into U $\sum$ V. The result is= $\begin{pmatrix} 1.0000 \\ -0.0001 \\ -17.9934 \\ -0.1185 \\ 55.0934 \\ -5.9508 \\ -44.3933 \\ -45.3474 \\ 106.5917 \\ -56.6057 \\ 7.6180 \\ 1.0662 \end{pmatrix}$

$$This one has the highest accuracy because it divides matrix.$$

4.6 The Computation of $\hat{\boldsymbol{x}}_{\text{backslash}}$

The backslash algorithm is provided by matlab to choose an algorithm to be used. When calculating in large number, it will choose more accurate method such as QR and SVD.

4.7 Warning Message(s)

That is beacuse the accuracy when calculating will not be accurate. According to the warning message:
Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 3.033068e-17.
Another thing is that when calculating the matrix, it may face some arithmetic error, such as divide by zero, matrix dimension mismatch and so on.

5 Summary and Conclusions

For this report, it can be observed that the most accurate method of calculating least squares problem is to use QR and SVD, while the least one is CGS. The second least one is normal equation. That is because when we calculate normal, we will use large amount of matrix calculation, we will do $A^T A$ which will lose a lot of precision. The amount of calculation of matrix of CGS will be much huger, which will lead to total loss of precision sometimes. Both QR and SVD use the matrix decomposition instead of original matrix multiplication, which are the most accurate two. For matlab operator, because the program choose the best choice of algorithm, the accuracy will be mostly approach to the OR or SVD.