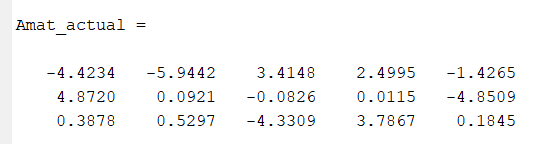
CH5440 Assignment-3

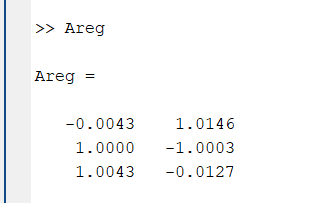
# Question 1

## Part a)

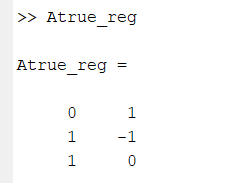
Constraint model obtained:



Regression model obtained:



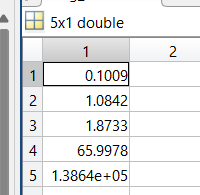
True regression model:



Maxdiff was found to be 0.0146.

Singular values: 372.3466, 8.1239, 1.3687, 1.0412, 0.3177

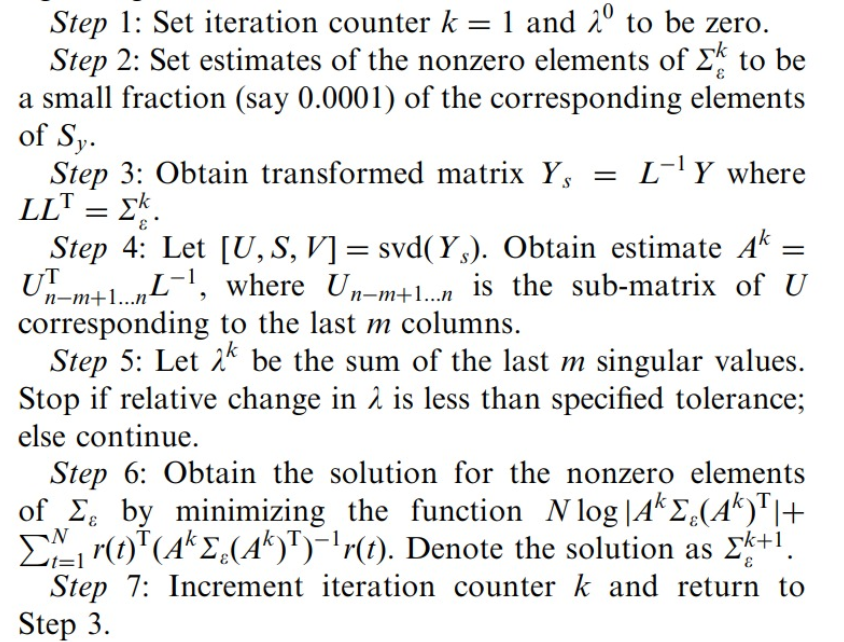
Eigen values of covariance matrix:



Notice that 3 eigen values are much smaller compared to the rest so we can say there are 3 constraints.

## Part b)

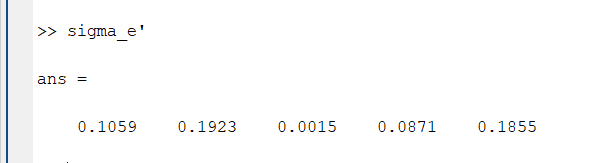
Following algorithm was used:



Source: Shankar Narasimhan, Sirish L. Shah, Model identification and error covariance matrix estimation from noisy data using PCA, Control Engineering Practice, Volume 16, Issue 1, 2008, Pages 146-155, ISSN 0967-0661, <https://doi.org/10.1016/j.conengprac.2007.04.006>

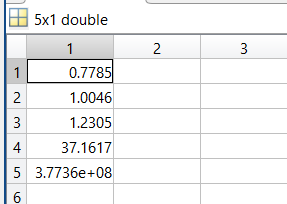
Step-6 was replaced by the linear equation form taught in class – minimizing negative of log likelihood couldn’t achieve fast convergence.

The estimated variances are as follows:



Maxdiff value = 0.0530

Eigen values of covariance matrix:



Notice how smallest 3 values are almost equal to one denoting 3 constraints. They are equal to one and not σ2 because we have scaled the values appropriately.

## Part c)

Eigen values:

## 

We can see that there is an order of magnitude change from fourth to third, and second to first. Since we are using IPCA (where we scale the data), all the eigenvalues corresponding to constraints should be equal to 1 and definitely should not have any change in order of magnitude. So our assumption that there are 4 constraints (=> only one principal component of interest) is wrong.

## Part d)

To determine the set of independent variables, we can check the condition number of the dependent variable matrix and choose the set of independent variables that gives a dependent variable matrix with minimal condition number (since we have to invert that).

Condition number =

|  |  |
| --- | --- |
| Independent Variables | Condition Number |
| [1,2] | 5.93471 |
| [1,3] | 2.934996 |
| [1,4] | 3.276916 |
| [1,5] | 1139.628 |
| [2,3] | 4.280301 |
| [2,4] | 4.940446 |
| [2,5] | 7.468241 |
| [3,4] | 52.69788 |
| [3,5] | 3.867451 |
| [4,5] | 4.245946 |

Best set (lowest condition number): F1 and F3

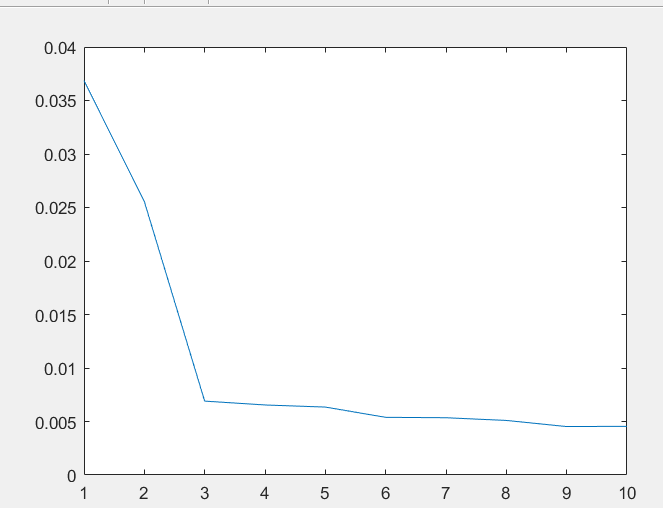
Worst set (highest condition number): F1 and F5

Yes, the inferences can be justified physically:

* F1 and F3 is the best because we can obtain F3 = F4 and F1 = F5, with only F2 requiring both these values. Similar pair – [1,4] also has a close enough condition number.
* F1 and F5 are the worst because they are actually linearly dependent pair. With just F1 and F5 it is impossible to determine the other values (note we don’t get infinity as condition number since there is some noise in the data).

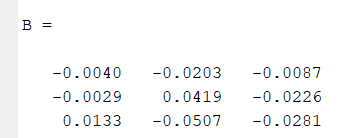
# Question 2

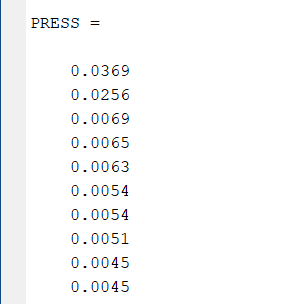
## Part a)



We can see the PRESS plot flattens after p = 3, so we pick 3 components for the final model.

Calibration model:

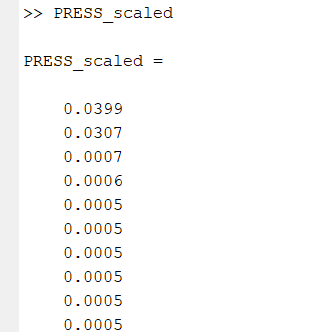


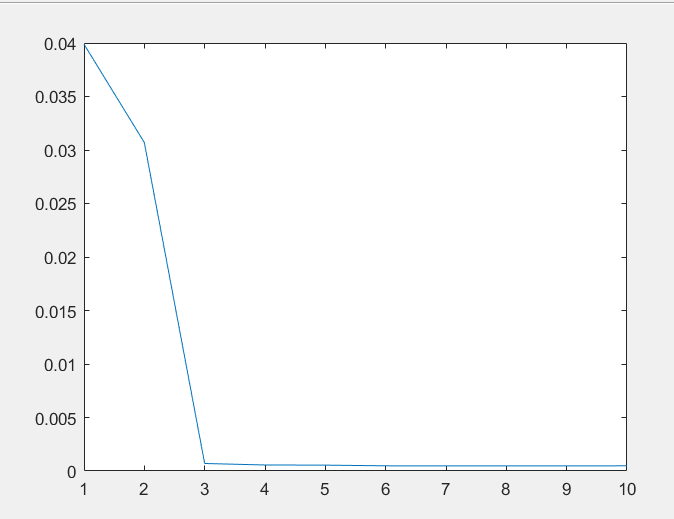


## Part b)

## 

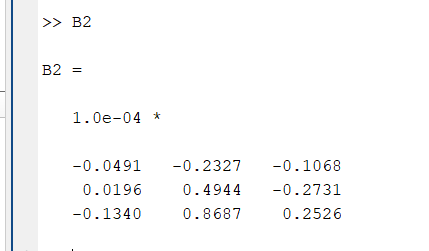
Figure: Plot of average standard deviations for each wavelength





Once more PRESS flattens after 3 components, so we choose 3 components for the final model

Calibration model:



Before this model is used, one should SCALE the input data using the standard deviations used to build this model.

## Part c)

## 

Plot of estimated standard deviations at each wavelength

Since the convergence of the model takes time (~176 parameters), I just computed RMSE for the entire dataset (entire dataset used to train and entire dataset used to test).

RMSE = 6.9188e-05 which is better than that of normal PCR (6.3474e-04) and scaled PCR (7e-05).

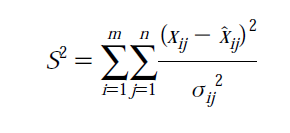
Note that the code takes a lot of time to converge. (~30 minutes)

## Part d)

Implementation

MLPCA was implemented as described in the paper, and then OLS was used to complete the implementation of MLPCR.

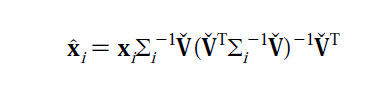
The following was the objective (copy-pasting from the paper; equation-9):



Minimize S2 wrt xhatij

Solving this objective is equivalent to performing PCA for the data matrix where **each data point is scaled according to its average standard deviation measured using replicates**.

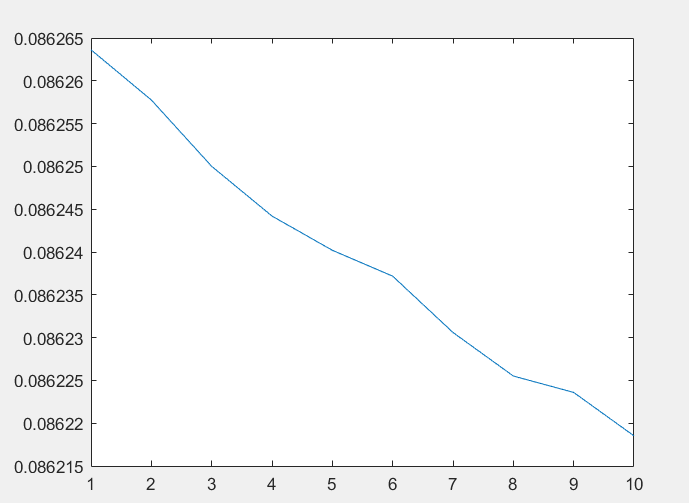
Projection is weighted by the errors in the measurements (equation-12):



**Assuming errors across mixtures and wavelengths are uncorrelated**, we get our covariance matrix for errors in each sample as simply a diagonal matrix of the average standard deviations (176\*1 vector) corresponding to that sample (so matrix size is 176\*176).

Model performance

If we take all 26 samples for training and find the RMSE of the resulting model, is **0.0012**. I also wrote a function LOOCV\_MLPCR which is similar to LOOCV\_PCR in idea except that it finds the leave one out cross validation error using MLPCR.



PRESS plot for MLPCR

As one can see RMSE does NOT vary much as we change the number of components and is constant around **0.0862**. This performance is worse compared to even the normal PCR. I believe these two could be the reasons for the same:

1. Order of standard deviation of measurement errors: some are in the order of 0.0001, probably they are causing some computation issues when we take the square of their inverse for projection.
2. Implementation issues in MLPCR; although I am confident that I translated my understanding to the code properly, my understanding of the expressions could’ve been wrong.