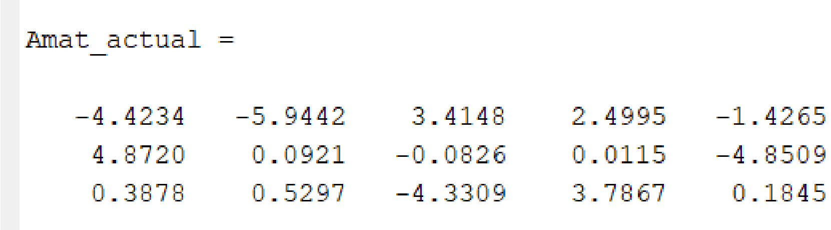
By S.Vishal
CH18B020

CH5440 Assignment-3

Question 1

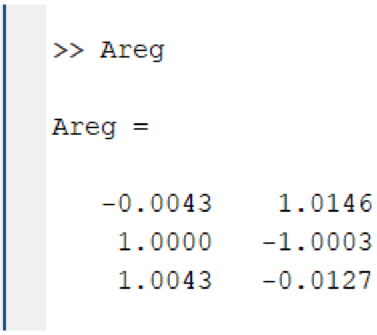
Part a)

Constraint model obtained:



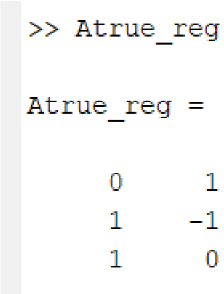
```
Amat_actual =

   -4.4234   -5.9442    3.4148    2.4995   -1.4265
    4.8720    0.0921   -0.0826    0.0115   -4.8509
    0.3878    0.5297   -4.3309    3.7867    0.1845
```

Regression model obtained:



```
>> Areg

Areg =

   -0.0043    1.0146
    1.0000   -1.0003
    1.0043   -0.0127
```

True regression model:

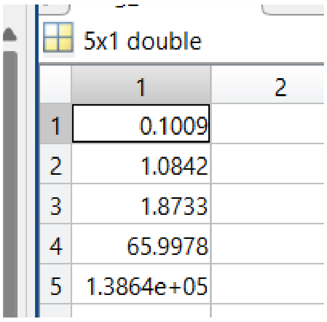


```
>> Atrue_reg

Atrue_reg =

     0     1
     1    -1
     1     0
```

Maxdiff was found to be 0.0146.

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

Eigen values of covariance matrix:



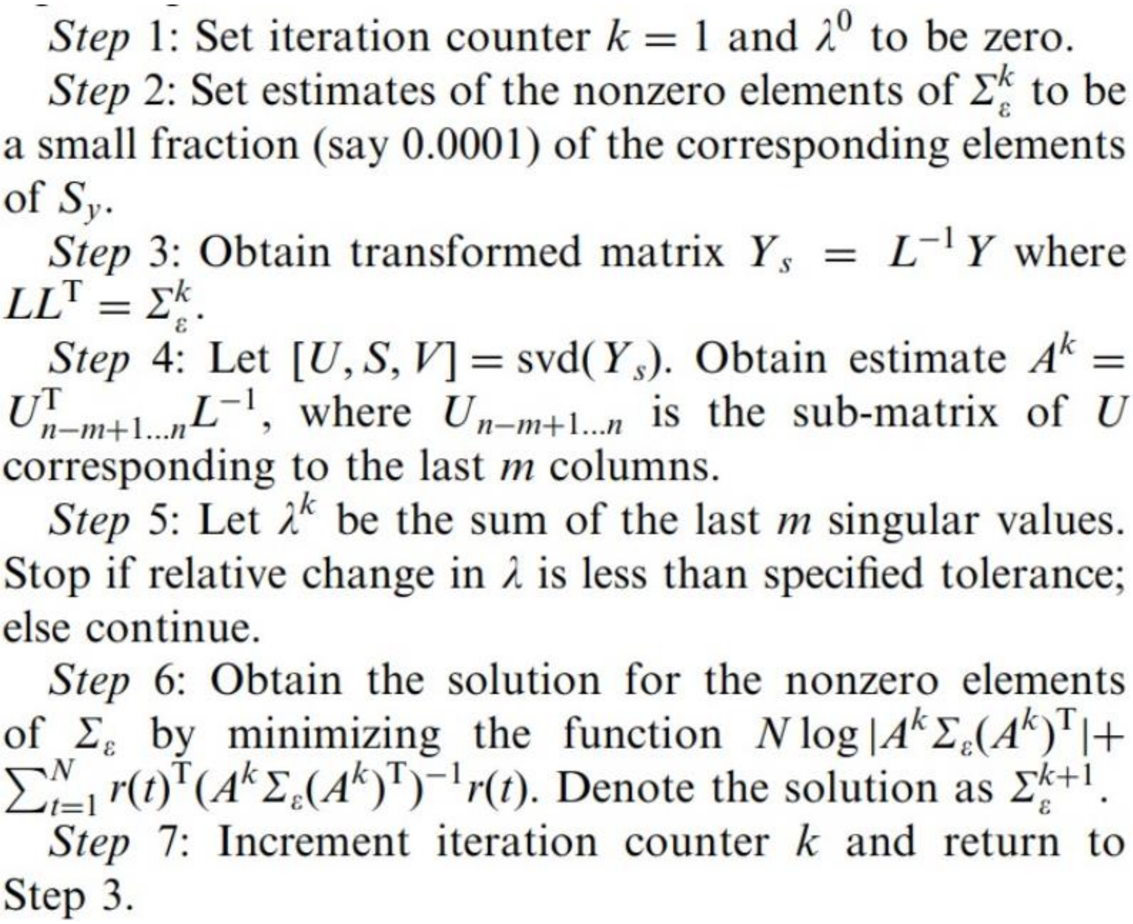
5x1 double

	1	2
1	0.1009	
2	1.0842	
3	1.8733	
4	65.9978	
5	1.3864e+05	

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:



Step 1: Set iteration counter $k = 1$ and λ^0 to be zero.

Step 2: Set estimates of the nonzero elements of Σ_ε^k to be a small fraction (say 0.0001) of the corresponding elements of S_y.

Step 3: Obtain transformed matrix $Y_s = L^{-1}Y$ where $LL^{\mathrm{T}} = \Sigma_\varepsilon^k$.

Step 4: Let $[U, S, V] = \mathrm{svd}(Y_s)$. Obtain estimate $A^k = U^{\mathrm{T}}_{n-m+1\ldots n}L^{-1}$, where $U_{n-m+1\ldots n}$ is the sub-matrix of U corresponding to the last m columns.

Step 5: Let λ^k be the sum of the last m singular values. Stop if relative change in λ is less than specified tolerance; else continue.

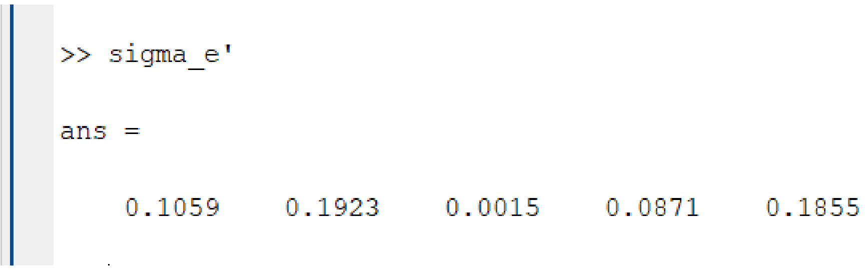
Step 6: Obtain the solution for the nonzero elements of Σ_ε by minimizing the function $N \log|A^k\Sigma_\varepsilon(A^k)^{\mathrm{T}}| + \sum_{t=1}^{N} r(t)^{\mathrm{T}}(A^k\Sigma_\varepsilon(A^k)^{\mathrm{T}})^{-1}r(t)$. Denote the solution as Σ_ε^{k+1}.

Step 7: Increment iteration counter k and return to Step 3.

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.

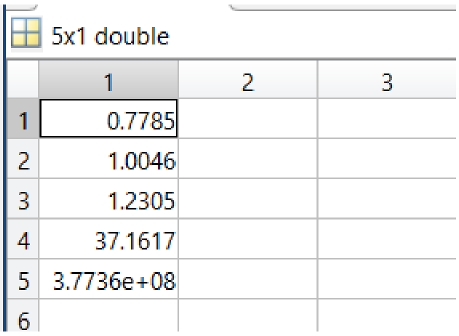
The estimated variances are as follows:



```
>> sigma_e'

ans =

    0.1059    0.1923    0.0015    0.0871    0.1855
```

Maxdiff value = 0.0530

Eigen values of covariance matrix:



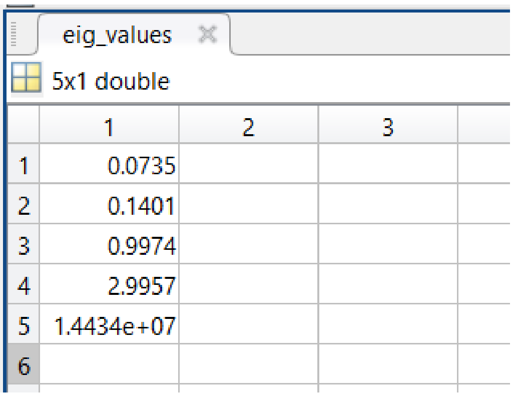
5x1 double

	1	2	3
1	0.7785		
2	1.0046		
3	1.2305		
4	37.1617		
5	3.7736e+08		
6			

Notice how smallest 3 values are almost equal to one denoting 3 constraints. They are equal to one and not σ^2 because we have scaled the values appropriately.

Part c)

Eigen values:



5x1 double

	1	2	3
1	0.0735		
2	0.1401		
3	0.9974		
4	2.9957		
5	1.4434e+07		
6			

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 = $\frac{\lambda_{largest}}{\lambda_{smallest}}$

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

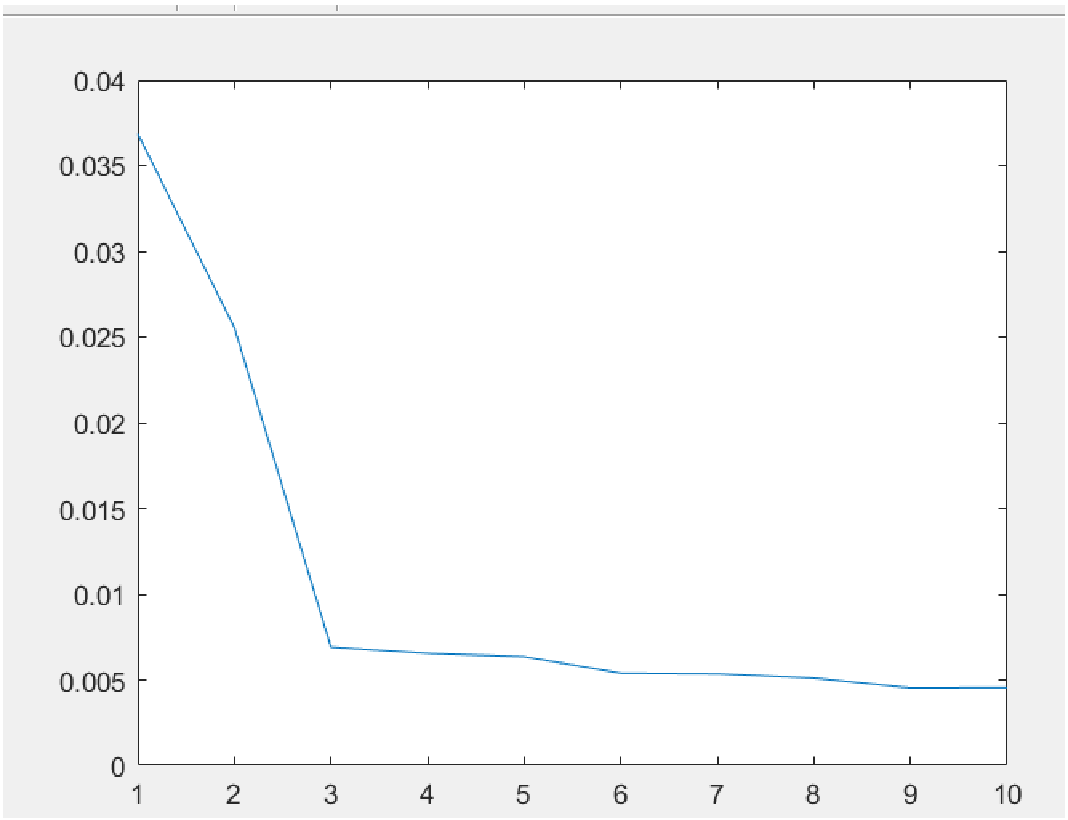
Worst set (highest condition number): F1 and F5

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).

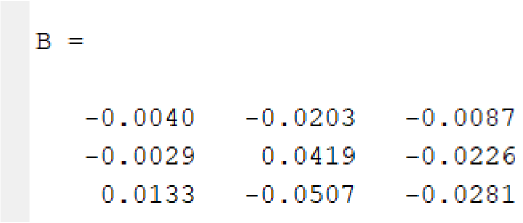
Question 2

Part a)

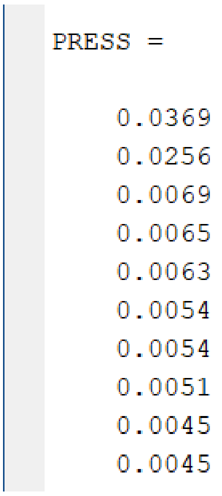


We can see the PRESS plot flattens after p = 3, so we pick 3 components for the final model.

Calibration model:



```
B =

   -0.0040   -0.0203   -0.0087
   -0.0029    0.0419   -0.0226
    0.0133   -0.0507   -0.0281
```



```
PRESS =

    0.0369
    0.0256
    0.0069
    0.0065
    0.0063
    0.0054
    0.0054
    0.0051
    0.0045
    0.0045
```

Part b)

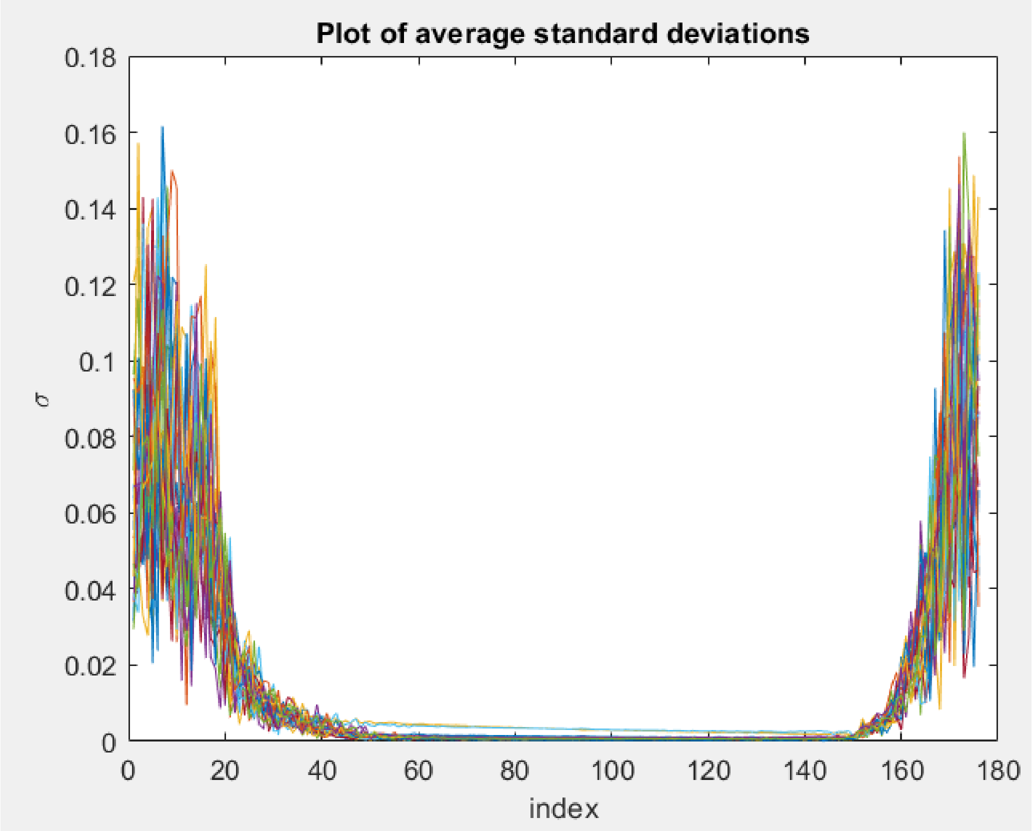
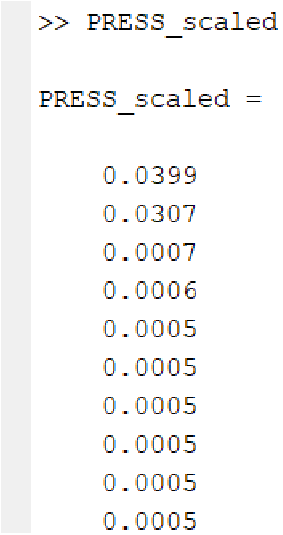
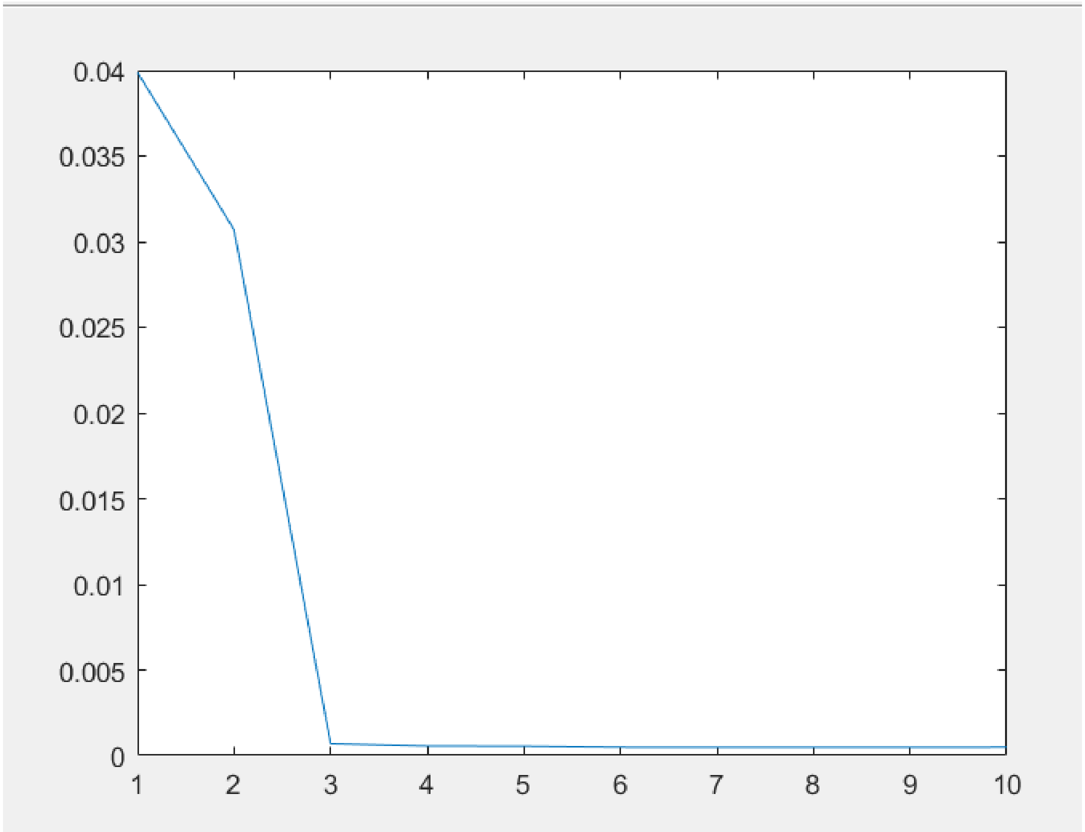


Figure: Plot of average standard deviations for each wavelength

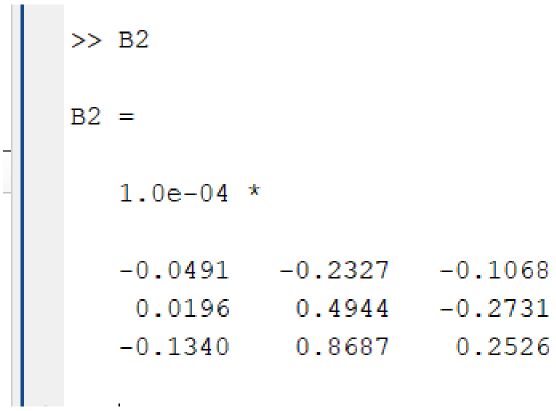


```
>> PRESS_scaled

PRESS_scaled =

    0.0399
    0.0307
    0.0007
    0.0006
    0.0005
    0.0005
    0.0005
    0.0005
    0.0005
    0.0005
```



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

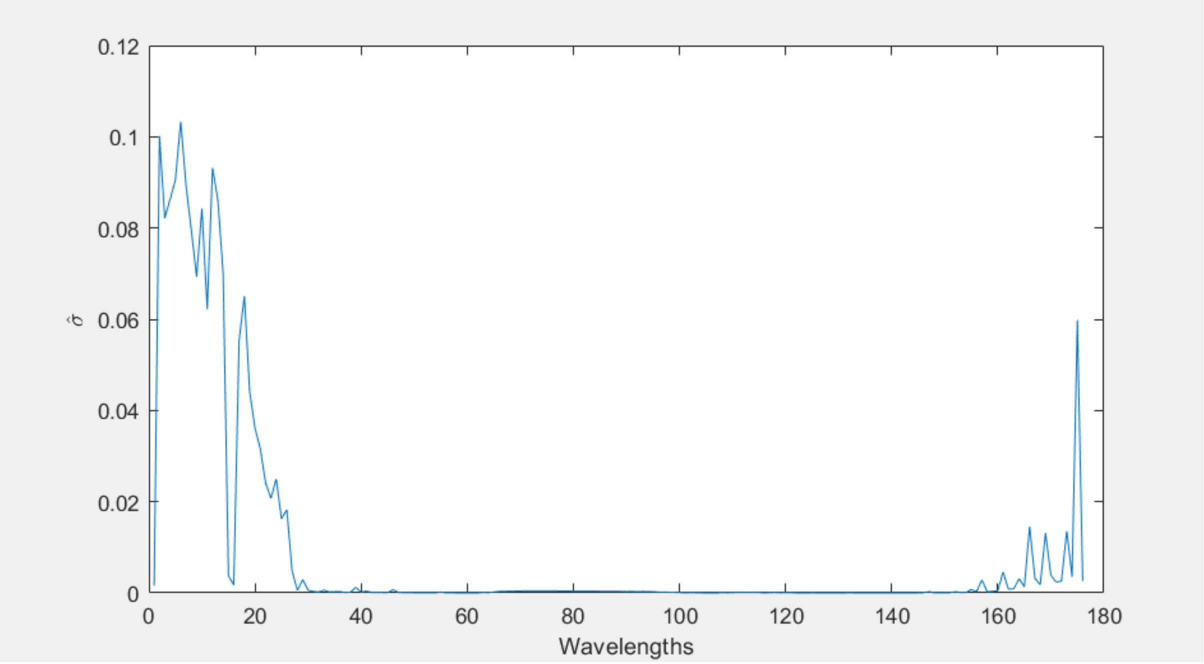
Calibration model:



```
>> B2

B2 =

   1.0e-04 *

  -0.0491   -0.2327   -0.1068
   0.0196    0.4944   -0.2731
  -0.1340    0.8687    0.2526
```

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)

<u>Implementation</u>

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):

By S.Vishal
CH18B020

$$S^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{(x_{ij} - \hat{x}_{ij})^2}{\sigma_{ij}^{\;2}}$$

Minimize S^2 wrt xhat$_{ij}$

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**.

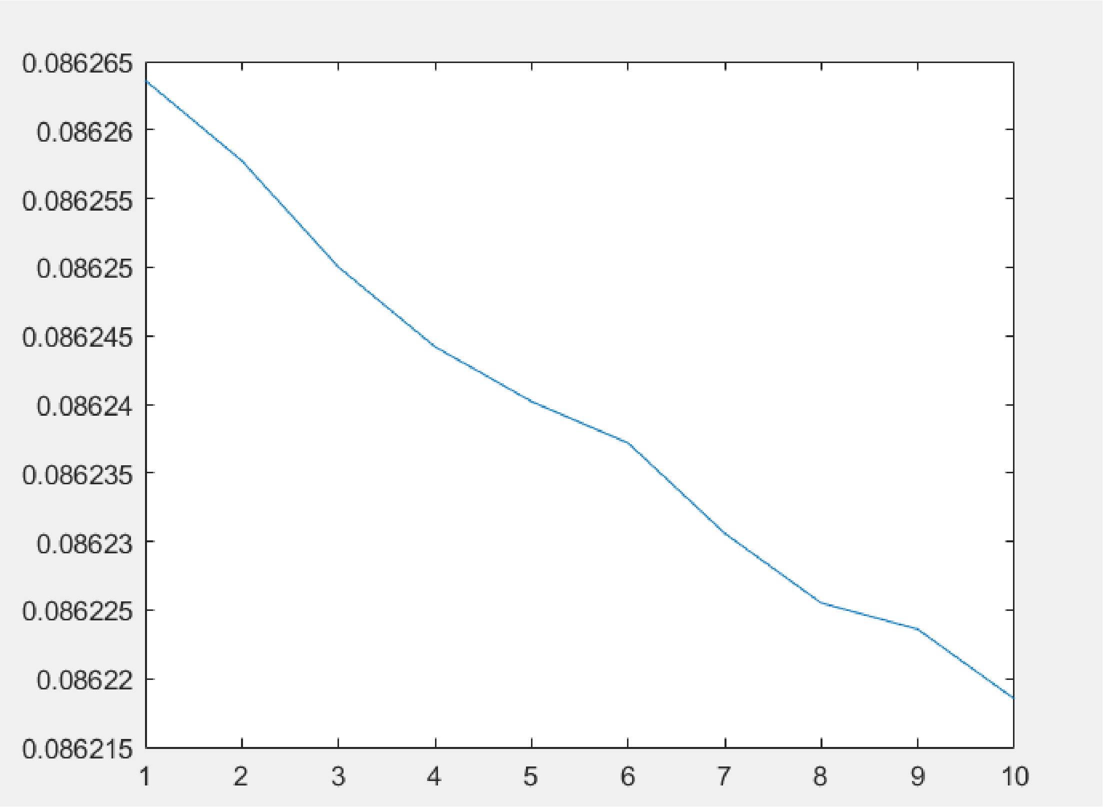
Projection is weighted by the errors in the measurements (equation-12):

$$\hat{\mathbf{x}}_i = \mathbf{x}_i\Sigma_i^{-1}\check{\mathbf{V}}(\check{\mathbf{V}}^{\mathrm{T}}\Sigma_i^{-1}\check{\mathbf{V}})^{-1}$$

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).

<u>Model performance</u>

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.