Final Examination Responses

Problem #1: SVD-Based Image Compression

Upon loading the matrix $\mathrm{X} \in \mathbb{R}^{512 \times 512}$ representing the Lena image into Matlab, we compute the singular value decomposition of X:

$$\mathrm{X} = U\Sigma V^{\mathrm{T}}$$

The singular values lie along the main diagonal of the diagonal matrix Σ, so, extracting this diagonal, we obtain the 512 singular values of X:

```
singularValues =

  1.0e+004 *

6.47330825819911
1.05961996225110
0.81755523052069
0.64754134886597
0.58836151256274
0.55525211603603
0.45882437576221
0.40931760421322
0.33750724064540
0.31730579610827
0.26724856425150
0.24484601142455
0.24047312471732
0.23256440070459
0.21813606898978
0.21505935669340
0.20875686840283
0.18541670588498
0.17379513240407
0.17053352709903
0.16372117630356
0.15873537927908
```

```
0.14491152812290
0.13368227356011
0.13229083573501
0.12955690926614
0.12166445781635
0.12114108995456
0.11462913566180
0.11192240493833
0.10622413178413
0.10532762853229
0.10450959011287
0.10201188402096
0.10028099638251
0.09982537696538
0.09406444972126
0.09332145096665
0.09183004098875
0.08914262572861
0.08503520727891
0.08438754892993
0.08274450142526
0.08120658537046
0.07945540253505
0.07706142441812
```

```
0.07458239814630
0.07347897295615
0.07262304408277
0.07030216177910
0.06937707502262
0.06882280955574
0.06544948045212
0.06458945451808
0.06391460701240
0.06309388076284
0.06162118801937
0.06137925895984
0.06068505422924
0.05941807291137
0.05699518172911
0.05605599805989
0.05566175520128
0.05522315967969
0.05375729506860
0.05254202950677
0.05217005097614
0.05203874609744
0.04939252766070
0.04839470182571
```

```
0.04789135788326
0.04772959615833
0.04689180323205
0.04621364375195
0.04587040340122
0.04454325783172
0.04418825364121
0.04397376665254
0.04303339393322
0.04255009407329
0.04181368051836
0.04134734856171
0.04019913816966
0.03983710231835
0.03922451537953
0.03907397022833
0.03838921432917
0.03788175899905
0.03768875512274
0.03755009534634
0.03633786657497
0.03609368966276
0.03567261517049
0.03533847309033
```

0.03519983892244
0.03478119964041
0.03411420873154
0.03384349214755
0.03259766077156
0.03226108457600
0.03162104276373
0.03146444502433
0.03129389752005
0.03069457624026
0.03002012970534
0.02929663256333
0.02917084200909
0.02894637243470
0.02837919384319
0.02816854208754
0.02778543009217
0.02746507769337
0.02713254153186
0.02659688506701
0.02627492272211
0.02572264610455
0.02548626293855
0.02519958691117
0.02504518202795
0.02480829419493
0.02474282928279
0.02425871366356
0.02386002647161
0.02379865112917
0.02319403995815
0.02289073818428
0.02262952121393
0.02245284655677
0.02217714606311
0.02194780230378

0.02176264116825
0.02134725126235
0.02106141683034
0.02094702561805
0.02079241881961
0.02065944002932
0.02033327927870
0.01986094200478
0.01975084036087
0.01938729869360
0.01935220519997
0.01905173571619
0.01898423876157
0.01860457932546
0.01828529947512
0.01815956575842
0.01792393159573
0.01790542930142
0.01756225283259
0.01744374021291
0.01709052227785
0.01694172973893
0.01686376695325
0.01659377718807
0.01639113576845
0.01624249511740
0.01622010126892
0.01590578600953
0.01587925197483
0.01548290746043
0.01537322883796
0.01519869645834
0.01509553936572
0.01503241058277
0.01490866535533
0.01457409808038

0.01453183358558
0.01440121596954
0.01426985427935
0.01421055475677
0.01395881216456
0.01387745337337
0.01372807187905
0.01354118604123
0.01344826453738
0.01330123214328
0.01317544127753
0.01307877169067
0.01294687400545
0.01282495376819
0.01278588520462
0.01275315626898
0.01261250560417
0.01239423538021
0.01231583429259
0.01225254049803
0.01214977108361
0.01202320184053
0.01185731016660
0.01184018888674
0.01177853313851
0.01165269967043
0.01158814092885
0.01143042682669
0.01132360707595
0.01125858568657
0.01110370952913
0.01102243399293
0.01094381054014
0.01089314440180
0.01072321132861
0.01066950243011

0.01059537867668
0.01056995213037
0.01049655271326
0.01040837753675
0.01035426798467
0.01021957812882
0.01012908769305
0.01002686076109
0.00997292871174
0.00990128439272
0.00988555606089
0.00976570542880
0.00967699997947
0.00956190347509
0.00954820199758
0.00948327287450
0.00933964847484
0.00928071175638
0.00914589567352
0.00913936946538
0.00902368337938
0.00898343845818
0.00896091910052
0.00886950610291
0.00881568775949
0.00865869556225
0.00865691197276
0.00855831692230
0.00852541645760
0.00847417361372
0.00844383889879
0.00841502388909
0.00834935156893
0.00826278932005
0.00824629435166
0.00814475508425

0.00810599344567
0.00802550712092
0.00797443251419
0.00790926445121
0.00785984336607
0.00781484176210
0.00780306815330
0.00775489643508
0.00771794752862
0.00767345335207
0.00758640401648
0.00756463671898
0.00752956531716
0.00746651917545
0.00740677499295
0.00732718215971
0.00726547968741
0.00723499688436
0.00718625546957
0.00713111510212
0.00710497800288
0.00704626453792
0.00702108637841
0.00697632239090
0.00693465037484
0.00688413218874
0.00682251900962
0.00677952291249
0.00669568572112
0.00668597901410
0.00664986838024
0.00661212786974
0.00655998427660
0.00654218618591
0.00652508572461
0.00648725847381

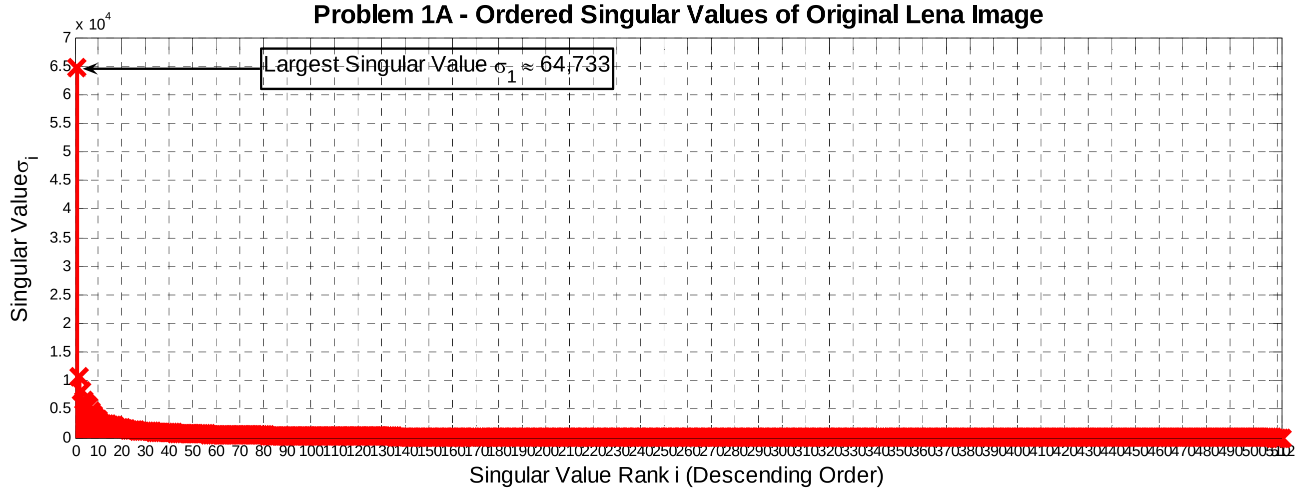
0.00642053722203
0.00640055761436
0.00635960959475
0.00629188339302
0.00624800512113
0.00619986997303
0.00618373898621
0.00617525965893
0.00607611122177
0.00605765802591
0.00599591333082
0.00597171742475
0.00592649494843
0.00587255761013
0.00585025328922
0.00581565686966
0.00578138272617
0.00573682839780
0.00570593376365
0.00567858079965
0.00563815264966
0.00559059658769
0.00553299569950
0.00552756180593
0.00548989387395
0.00542999221540
0.00540824045038
0.00537277901530
0.00534466480566
0.00531965296434
0.00527949445959
0.00524089918188
0.00519553324269
0.00517568874497
0.00514654255598
0.00510938143603

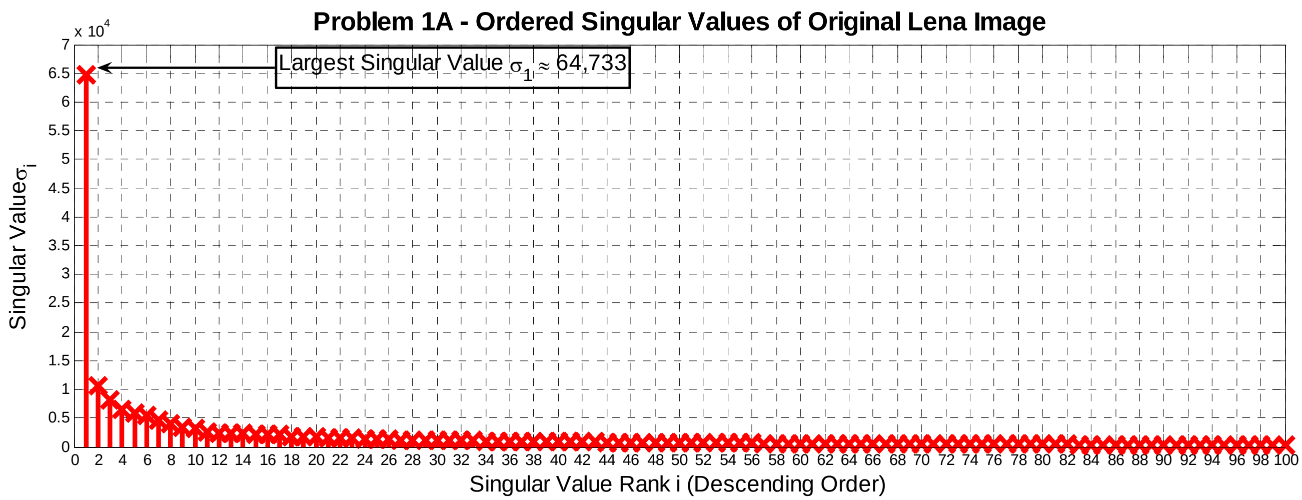
0.00509176879804
0.00507010572151
0.00505213073774
0.00499962997408
0.00497151417525
0.00488617235368
0.00487721010825
0.00484073714970
0.00482230599887
0.00477554003634
0.00473900754853
0.00467877141822
0.00467553168544
0.00464549600262
0.00463165811815
0.00460598272784
0.00458030462138
0.00453275988985
0.00451058011234
0.00447177123911
0.00444668421416
0.00443450482685
0.00439218910538
0.00434878134467
0.00433119103340
0.00432037623236
0.00427434192213
0.00425273516655
0.00423443491249
0.00420481811944
0.00416741797719
0.00410518574852
0.00407743432088
0.00406593334255
0.00402286285979
0.00400549451240

0.00397406489044
0.00395647349083
0.00390042860965
0.00388795581849
0.00385012596384
0.00383316294029
0.00381330055415
0.00378163608868
0.00375925260822
0.00373425064878
0.00368433362748
0.00366290563341
0.00362922568000
0.00361028810823
0.00359568941848
0.00355351113978
0.00351287567563
0.00347329284630
0.00345540507529
0.00343575719612
0.00341644438270
0.00339858419831
0.00334091722144
0.00332948694782
0.00330116775345
0.00327669503834
0.00325931044924
0.00323107288132
0.00317094828428
0.00316644300094
0.00314672635276
0.00311287002425
0.00305956937837
0.00304634350977
0.00304223549284
0.00299565907706

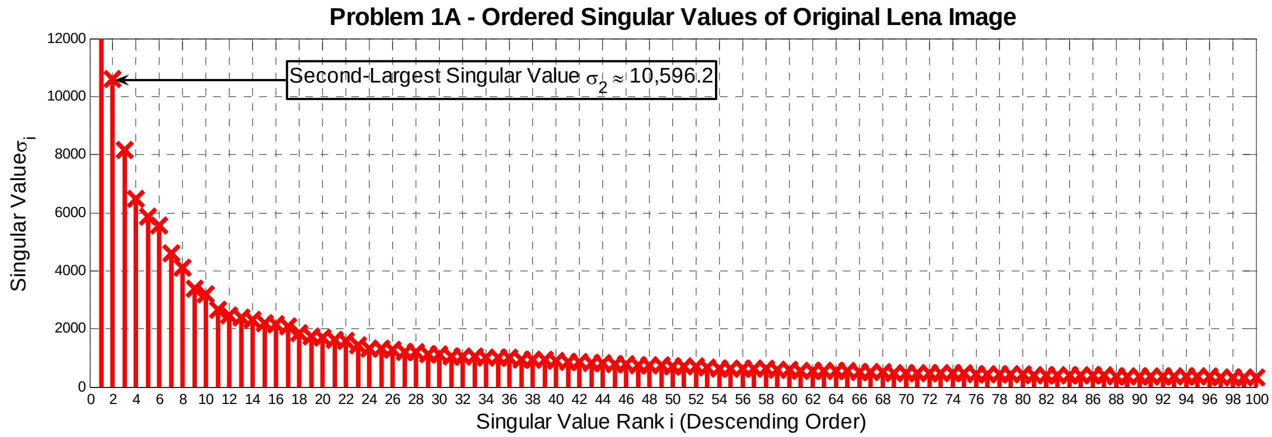
0.00298062043968	0.00207868649634	0.00126217995723	0.00041296786688
0.00297430686626	0.00205962510243	0.00124373230357	0.00039640982620
0.00295338852713	0.00205318288050	0.00121643286909	0.00037247007845
0.00289444816521	0.00201690824392	0.00119418769604	0.00034596755949
0.00284937808869	0.00198863759995	0.00118478798433	0.00033485108630
0.00283757919459	0.00196403964717	0.00114258353104	0.00031333447683
0.00280322027543	0.00195441580532	0.00112098031928	0.00030040622747
0.00278075341203	0.00192653043810	0.00111645326686	0.00026406714791
0.00276037214485	0.00190236925037	0.00109416750472	0.00022560803956
0.00274223107405	0.00186524161635	0.00103959771528	0.00020045003020
0.00270824409071	0.00185937005005	0.00103023898705	0.00017998821195
0.00269762189619	0.00181658119773	0.00101619766241	0.00015384456449
0.00266894640646	0.00180706188179	0.00098607147722	0.00012992207089
0.00265150899101	0.00179960651164	0.00093558987687	0.00008149165915
0.00262161436119	0.00176389280329	0.00091350471384	0.00007544409351
0.00260024261571	0.00175363479176	0.00089984087922	0.00006883536044
0.00257715466362	0.00171441380063	0.00087796216703	0.00004255222893
0.00253871504057	0.00168338677924	0.00087258342528	0.00000000000000
0.00252295363043	0.00166231586638	0.00082985144729	0.00000000000000
0.00249453086587	0.00164708537643	0.00081652783286	0.00000000000000
0.00248000351151	0.00163005205788	0.00078235925140	0.00000000000000
0.00246589282688	0.00160319387990	0.00073863728863	0.00000000000000
0.00244367022191	0.00159162654532	0.00072975014756	
0.00243086650701	0.00157566470087	0.00069902672520	
0.00239788348929	0.00156528142324	0.00068268322352	
0.00234862707594	0.00152397373176	0.00065423648318	
0.00233551594415	0.00151950435955	0.00063508088711	
0.00231577959772	0.00147961119837	0.00059833634203	
0.00230136450998	0.00145268560967	0.00059445893524	
0.00226383015142	0.00142486881132	0.00057948002925	
0.00223020582080	0.00140881004387	0.00056614125427	
0.00221048776606	0.00139016253009	0.00055436956805	
0.00219184753850	0.00137353509782	0.00053131753020	
0.00217734911256	0.00134614010135	0.00046897386720	
0.00212510776143	0.00131395539861	0.00044411684313	
0.00210376989448	0.00127220201109	0.00043844418765	

Plotting these singular values in descending order, from largest to smallest, we observe the trend:

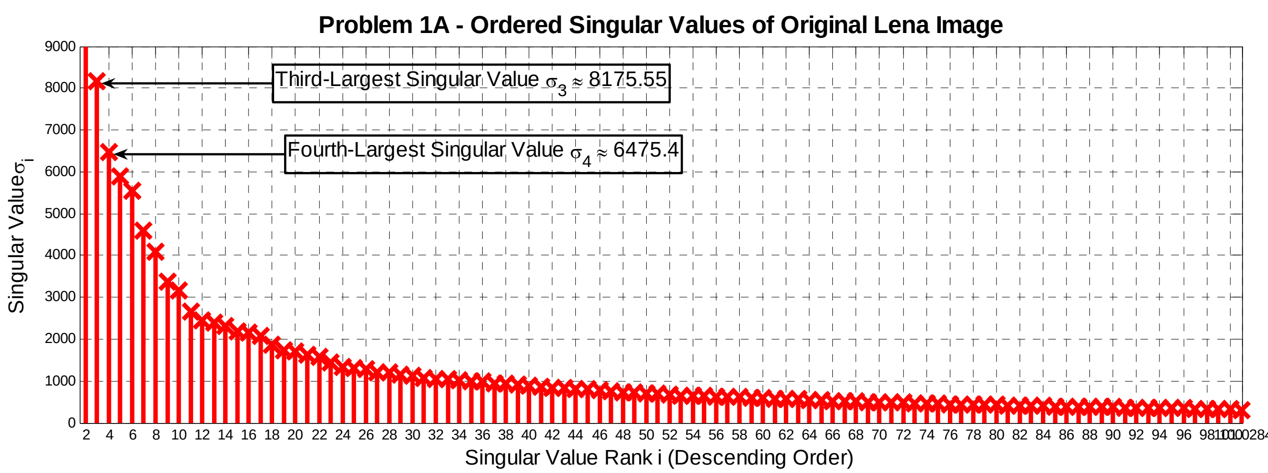




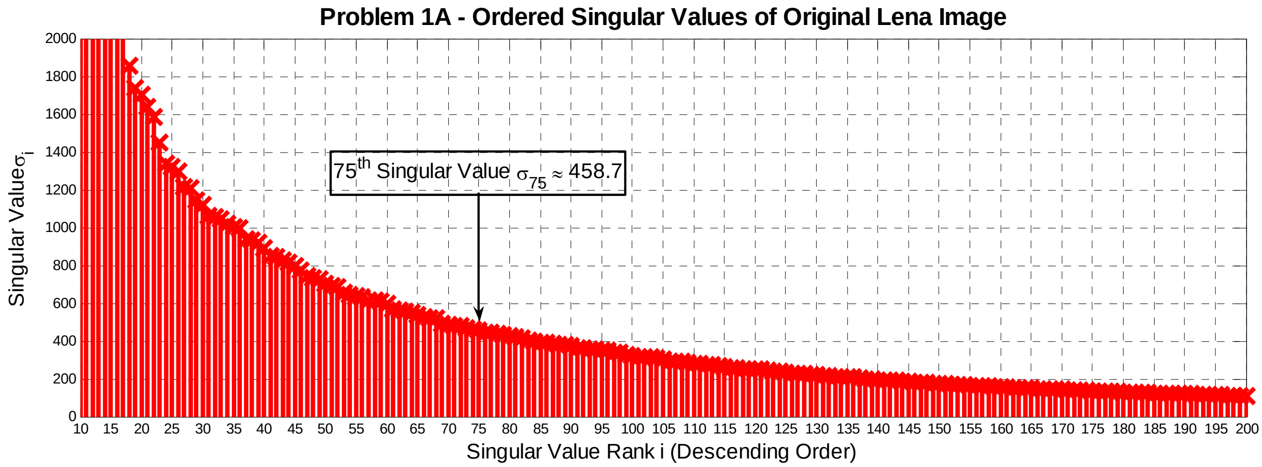
The first singular value $\sigma_1 \approx 64733$ dwarves all the other singular values, suggesting that the majority of the image information in Lena derives from the first singular vectors. The difference between the first and second singular values is approximately $\sigma_1 - \sigma_2 \approx 51437$, meaning that the next singular value is about one sixth in size; the first singular vectors of X truly pack a significant portion of the Lena image! If we focus on the first hundred singular values, we see that the significance decreases almost inversely proportionally after the first, to the next level of insignificance after fifteen values. Removing the first singular value from our viewing plane to magnify the scale,



This magnification only ascertains what the previous graph insinuated: the singular values decrease almost linearly after the first and second, reaching depths of 2000 at about the eighteenth singular value; from that point onwards, singular values all maintain less than 3% of the size of the largest singular value. Cropping even the second singular value $(\sigma_2 \approx 10596)$ from the picture,



The rate by which the singular values decrement also decreases, as decay slows to a near halt after approximately the first seventy-five values, justifying our decision to use a rank-75 approximation to the Lena image. The curve grows more and more similar to a decaying exponential, with the trend revealing that singular values grow closer and closer and lower and lower the deeper we probe into the diagonal matrix Σ. While truncations around the eighteenth or twenty-fifth singular values might also make sense, decrement still occurs after those values, so we opt to wait until the curve truly begins to stabilize into a plateau, as it displays during the seventh-fifth value:



Of course, all stabilization is relative. At the previous viewing scale, it seemed as if the singular values had reached their minimum by the seventh-fifth entry, but in this downscaled plot, it appears that decreasing occurs until the two-hundredth singular value. In fact, decrease actually occurs until the final (512^{th}) singular value, but the numbers grow so low relative to the first that we just ignore them after we have reached a desired error tolerance. By approximating the Lena image with the first seventy-five singular values, our error should match the first omitted singular value, the seventh-sixth one: $\sigma_{76} \approx 445.432578$. We will soon verify this fact. For the time being, we satisfy ourselves with less than 0.7% error, since the first omitted singular value $\sigma_{76} \approx 445.432578$ amounts to less than 0.7% of the maximum singular value $\sigma_1 \approx 64733$, which we call the spectral norm $\|X\|$.

We can save image storage space and improve image-processing efficiency by approximating our full Lena image with a rank-75 approximation:

$$\tilde{X} = \sum_{i=1}^{75} \sigma_i u_i v_i^T$$

Our image still comprises 512×512 doubles, but we need to store only the first seventy-five singular values and left and right singular vectors, since the simple summation above yields the rank-75 approximation. As a result, we need to store only

$$\left(\frac{1 \text{ singular value } \sigma_i + 512 \text{ entries in } u_i + 512 \text{ entries in } v_i^T}{\text{summation term dyad}} \right) \times 75 \text{ dyads} = \boxed{76{,}875 \text{ doubles}}$$

Compared to the original storage requirement of 512 × 512 = 262,144 doubles, we have effectively compressed the image down to

$$\frac{76{,}875 \text{ doubles}}{262{,}144 \text{ doubles}} \approx \boxed{0.293255} \approx 29.3255\%$$

of its original size. In other words, we have reduced the number of bytes required to store information in the Lena image by 70.6745%, a marked improvement. Finally, recalling that our approximation error amounts to the first omitted singular value, we calculate the relative error ratio to be approximately

$$\frac{\|X - \tilde{X}\|}{\|X\|_F} = \frac{\sigma_{76}}{\sqrt{\mathrm{Tr}\,(X^T X)}} \approx \frac{445.432578}{68{,}072.3261} \approx \boxed{0.00654352} \approx 0.654352\%$$

Considering that we tapped only 75 of the 512 singular values, this low-rank approximation feigns the full Lena image quite well, as the following juxtaposition corroborates:

Problem 1 - Original Lena Image

Rank-75 Approximation of Lena

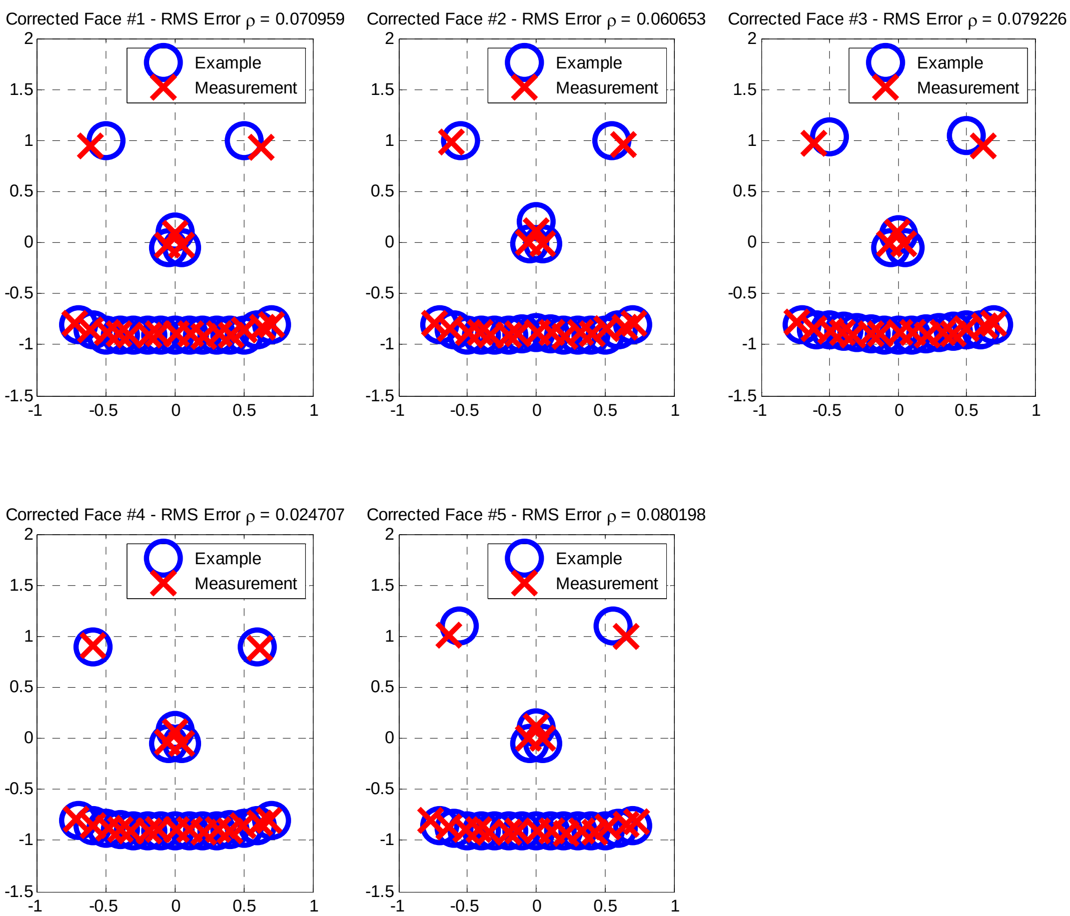
Only some residual artifacts are visible, with the folds in Lena's hat and the smoothness of her skin slightly compromised in the approximation. The compressed image closely parallels the original image in all but the smallest flaws, despite consuming only 29% of the storage space of the original image.

<u>Problem #2</u>: Optimal Correction of Facial Features

Upon applying least squares to minimize the root-mean-square (RMS) error value for each of the five example faces, we obtain sets of least squares parameters for each of the feature sets in $\{F^{(1)}, F^{(2)}, F^{(3)}, F^{(4)}, F^{(5)}\}$. Once we construct the least squares rotation matrices and two-dimensional translation vectors for each of the five example faces, we can invert the transformation:

$$x_{inverted}^{(i)} = R^{-1}\,(y - t)$$

applied to every feature point *y* in the noisy measurement matrix Y. The unrotated versions of all five example faces follow:



The fourth face described by features in $F^{(4)}$ yields the lowest RMS error and most sensible appearance, so we conclude that the measured face in Y must have originally been the **fourth** example face. The transformation parameters for the fourth face are

Dilation by $\boxed{\alpha \approx 1.511813}$

Counterclockwise rotation by $\boxed{\theta \approx 19.9661°}$

Translation by $\boxed{t \approx \begin{bmatrix} 0.2539154 \\ 0.1010018 \end{bmatrix}}$

Scaling Rotation Matrix: $\boxed{R \approx \begin{bmatrix} 1.420945 & -0.51623 \\ 0.51623 & 1.420945 \end{bmatrix}}$

The RMS residual from matching the fourth face yields the lowest RMS residual of any correction:

$$\rho_{Face\ \#1} \approx 0.07095939$$

$$\rho_{Face\ \#2} \approx 0.06065328$$

$$\rho_{Face\ \#3} \approx 0.07922649$$

$$\boxed{\rho_{Face\ \#4} \approx 0.02470671}$$

$$\rho_{Face\ \#5} \approx 0.08019847$$

Both algebraically and geometrically, the fourth face clearly matches our corrected measurement most closely.

Problem #5: Estimating Path Gain Matrix and Self-Noise Power

In our data, we learn that we have $n = 3$ transmitter/receiver pairs and $K = 5$ sets of measurements. Having already estimated the self-noise power to be approximately $\sigma = 0.01$, we proceed to define the coefficient matrices:

$$P_1 = \begin{bmatrix} \frac{p_1^{(1)}}{s_{meas,1}^{(1)}} & -p_2^{(1)} & -p_3^{(1)} \\ \frac{p_1^{(2)}}{s_{meas,1}^{(2)}} & -p_2^{(2)} & -p_3^{(2)} \\ \frac{p_1^{(3)}}{s_{meas,1}^{(3)}} & -p_2^{(3)} & -p_3^{(3)} \\ \frac{p_1^{(4)}}{s_{meas,1}^{(4)}} & -p_2^{(4)} & -p_3^{(4)} \\ \frac{p_1^{(5)}}{s_{meas,1}^{(5)}} & -p_2^{(5)} & -p_3^{(5)} \end{bmatrix}, \quad P_2 = \begin{bmatrix} -p_1^{(1)} & \frac{p_2^{(1)}}{s_{meas,2}^{(1)}} & -p_3^{(1)} \\ -p_1^{(2)} & \frac{p_2^{(2)}}{s_{meas,2}^{(2)}} & -p_3^{(2)} \\ -p_1^{(3)} & \frac{p_2^{(3)}}{s_{meas,2}^{(3)}} & -p_3^{(3)} \\ -p_1^{(4)} & \frac{p_2^{(4)}}{s_{meas,2}^{(4)}} & -p_3^{(4)} \\ -p_1^{(5)} & \frac{p_2^{(5)}}{s_{meas,2}^{(5)}} & -p_3^{(5)} \end{bmatrix}, \quad P_3 = \begin{bmatrix} -p_1^{(1)} & -p_2^{(1)} & \frac{p_3^{(1)}}{s_{meas,3}^{(1)}} \\ -p_1^{(2)} & -p_2^{(2)} & \frac{p_3^{(2)}}{s_{meas,3}^{(2)}} \\ -p_1^{(3)} & -p_2^{(3)} & \frac{p_3^{(3)}}{s_{meas,3}^{(3)}} \\ -p_1^{(4)} & -p_2^{(4)} & \frac{p_3^{(4)}}{s_{meas,3}^{(4)}} \\ -p_1^{(5)} & -p_2^{(5)} & \frac{p_3^{(5)}}{s_{meas,3}^{(5)}} \end{bmatrix}.$$

As previously determined, the self-noise power vector is

$$\underline{\sigma} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \end{bmatrix}.$$

After ascertaining that the power matrices are *full rank* with $\text{rank}(P_1) = \text{rank}(P_2) = \text{rank}(P_3) = 3$, we proceed with the least-squares approximate solution:

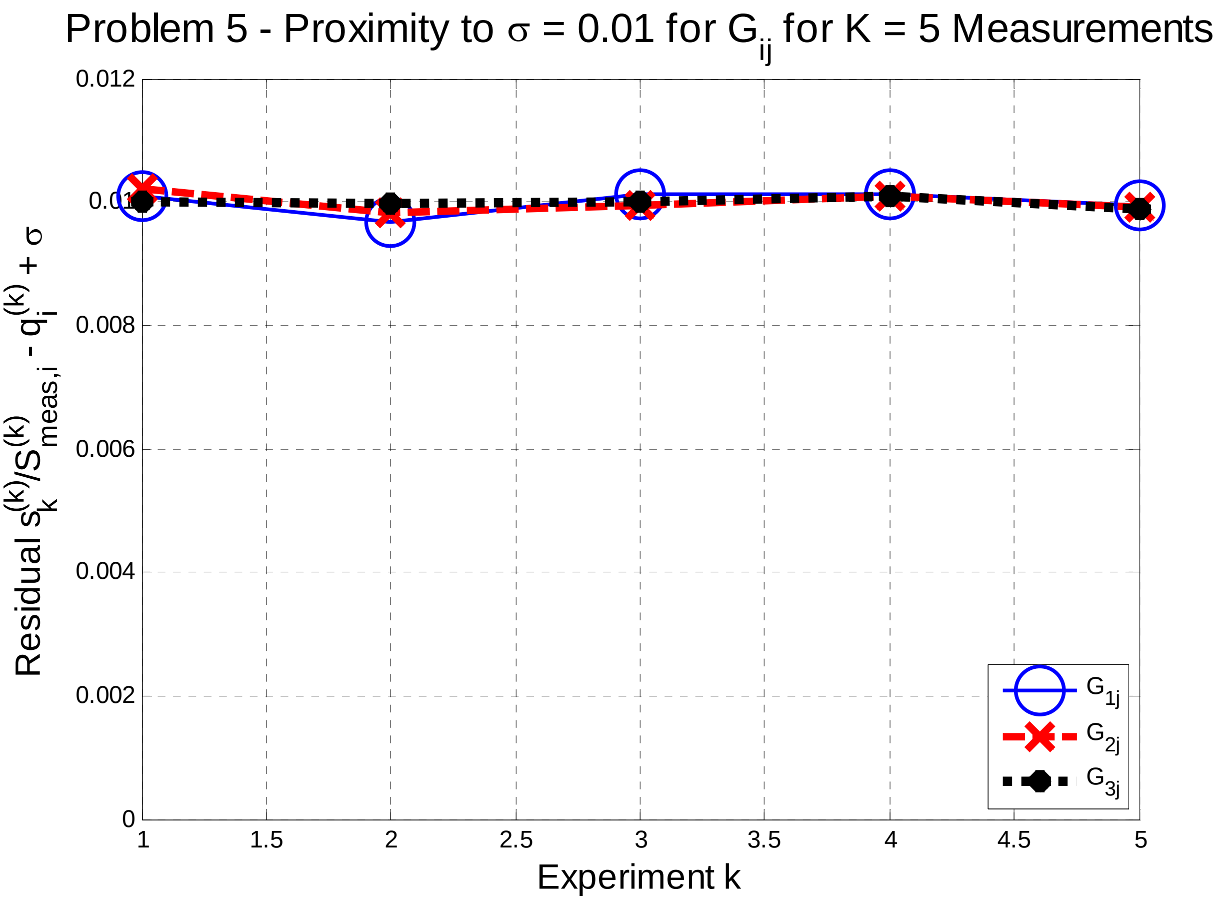
$$G_{least\ squares} = \begin{bmatrix} \leftarrow & \underline{\sigma}^T P_1 (P_1^T P_1)^{-1} & \rightarrow \\ \leftarrow & \underline{\sigma}^T P_2 (P_2^T P_2)^{-1} & \rightarrow \\ \leftarrow & \underline{\sigma}^T P_3 (P_3^T P_3)^{-1} & \rightarrow \end{bmatrix}$$

$$\boxed{G_{least\ squares} \approx \begin{bmatrix} 1.547988 & 0.103438 & 0.206325 \\ 0.108878 & 2.712652 & 0.325769 \\ 0.098612 & 0.098754 & 0.987341 \end{bmatrix}}$$

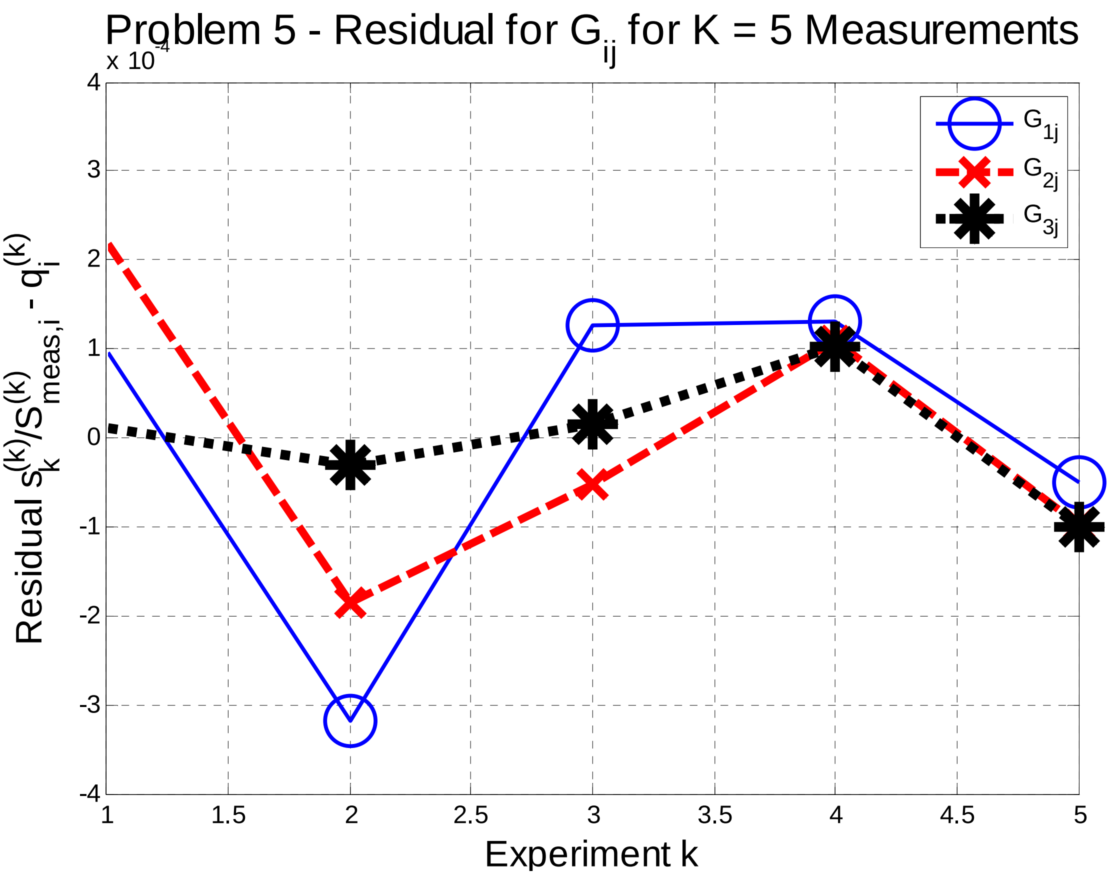
The minimized root-mean-square (RMS) residual error is approximately

$$\rho = \frac{1}{5}\left[\sum_{k=1}^{5}\sum_{i=1}^{3}\left(\frac{p_i^{(k)}}{S_{meas,i}^{(k)}}G_{ii} - \sum_{j\neq i}\left\{p_j^{(k)}G_{ij}\right\} - \sigma\right)^2\right] \approx \boxed{0.0002340469256}$$

In order to ascertain that our path gain matrix model provides the ideal fit closely to the five experimental data, we first compare the quantity $\frac{p_i^{(k)}}{s_{meas,i}^{(k)}} G_{ii} - \sum_{j \neq i} \left\{ p_j^{(k)} G_{ij} \right\}$ to $\sigma = 0.01$ to ensure that the gain values stay close to the data relative to this self-noise value:



Indeed, all values computed using the gain matrix above appear to hover randomly around $\sigma = 0.01$, therefore suggesting that we have found a workable solution; the experimental data, due to noise, temporal decorrelation, and other measurement variants, foil a perfect fit, but the values that we have determined using least squares appear to yield the closest possible fit. We can also plot the residual independent of the self-noise power:



This figure reveals that all residuals fall within ± 0.0004 of zero. Interestingly, near all residuals for a single experiment generally share the same sign, indicating that each experiment seemed to encounter the same measurement flaws in all transmitters and receivers; for example, setting the power level to $p^{(2)}$ seemed to yield a value of $S_{\text{meas}}^{(2)}$ that was consistently higher than anticipated given our gain matrix. However, the beauty of least squares arises from the fact that, after collecting several (K = 5) measurements, we are well-equipped to determine the *best* gain mode that fits all of the experimental data, blending together all of the small measurement differences between experiments.

Finally, the graph also reveals that we can justifiably optimize each set of receiver gains separately, since we can minimize the residual over each set of $G_{i1}, \cdots, G_{i5}$ without consulting the other receiver gains. Minimizing each term in the total error therefore minimizes the total sum of residuals. In other words, we cannot reduce the residuals associated with receiver i by tweaking gain values at receiver $j \neq i$, simply because the two sets $\{G_{i1}, \cdots, G_{i5}\}$ and $\{G_{j1}, \cdots, G_{j5}\}$ do not influence each other.