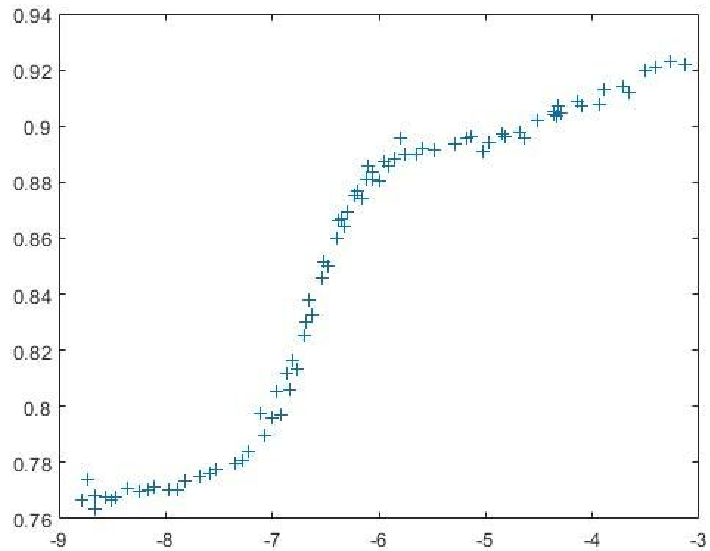
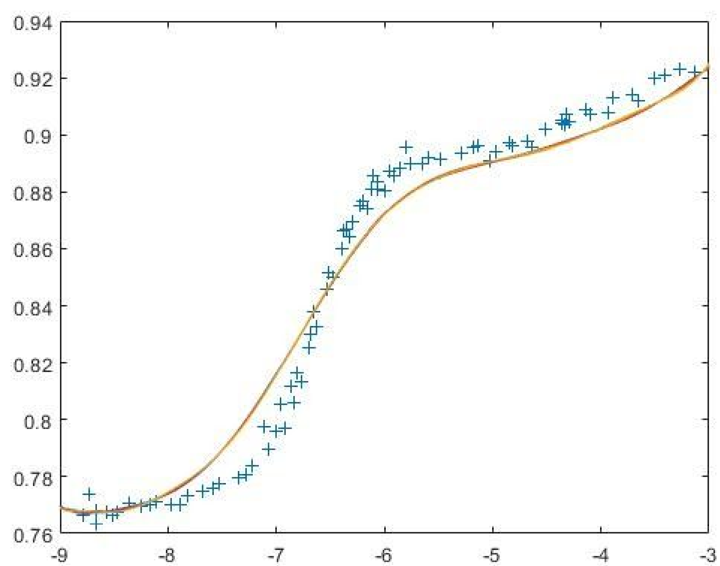
Jakob Horvath
u1092049



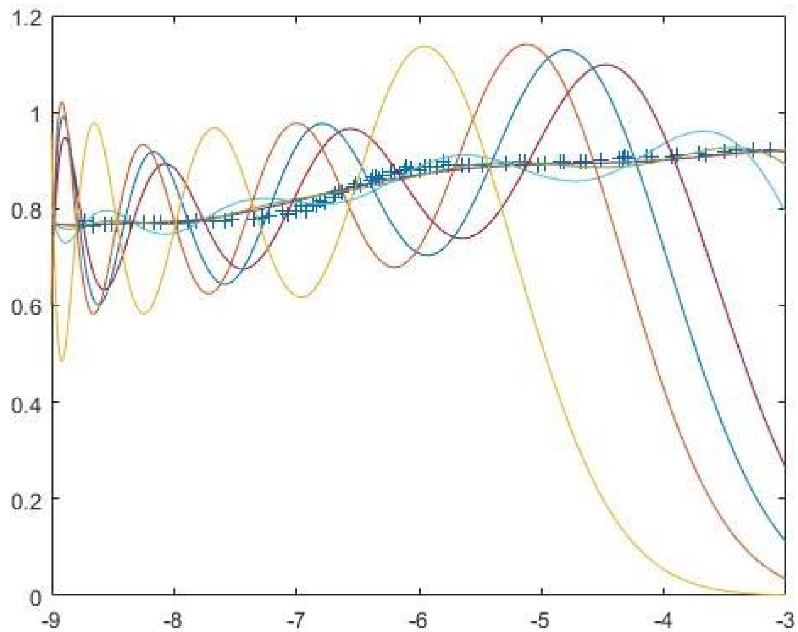
Above are the original x and y values read from the data.txt file.

Normal Equations:



```
imeth= 0 n=82 m=9 l(errlsq) = 9.173504e-02
imeth= 0 n=82 m=11 l(errlsq) = 9.105292e-02
imeth= 0 n=82 m=13 l(errlsq) = 9.118527e-02
imeth= 0 n=82 m=15 l(errlsq) = 9.003898e-02
imeth= 0 n=82 m=17 l(errlsq) = 8.964765e-02
imeth= 0 n=82 m=19 l(errlsq) = 8.917324e-02
imeth= 0 n=82 m=21 l(errlsq) = 8.943386e-02
imeth= 0 n=82 m=23 l(errlsq) = 8.925408e-02
imeth= 0 n=82 m=25 l(errlsq) = 9.012503e-02
```

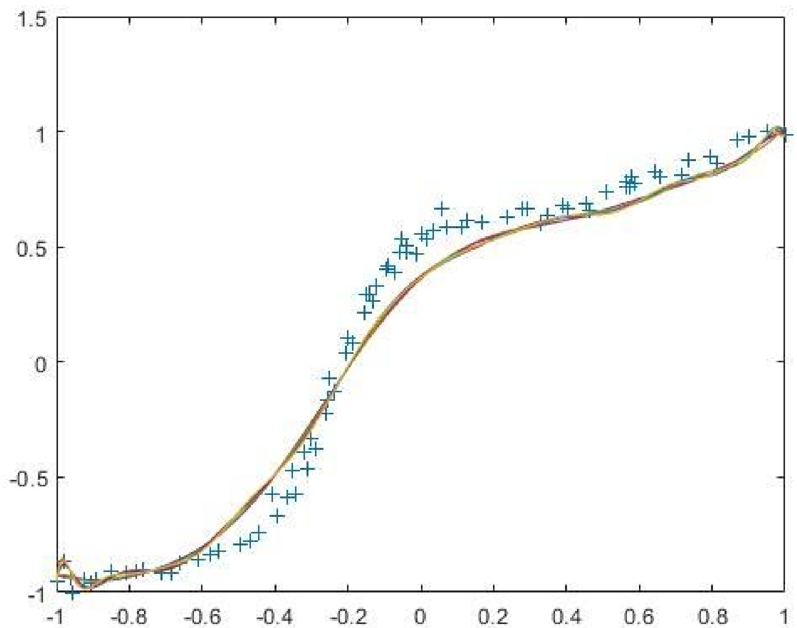
QR:




```
imeth= 1 n=82 m=9 l(errlsq) = 9.148820e-02
imeth= 1 n=82 m=11 l(errlsq) = 9.113293e-02
imeth= 1 n=82 m=13 l(errlsq) = 9.580128e-02
imeth= 1 n=82 m=15 l(errlsq) = 1.196585e-01
imeth= 1 n=82 m=17 l(errlsq) = 2.190206e-01
imeth= 1 n=82 m=19 l(errlsq) = 1.311671e+00
imeth= 1 n=82 m=21 l(errlsq) = 1.896318e+00
imeth= 1 n=82 m=23 l(errlsq) = 2.469235e+00
imeth= 1 n=82 m=25 l(errlsq) = 3.791213e+00
```

The previous charts show the approximations generated by both a normal equations method as well as a QR method. Each line depicts an estimate when using a polynomial degree that varied from 9 to 25, increasing by 2 with each iteration. As the corresponding least squares error shows, the normal equations were much more successful at creating a reasonable approximation. QR decomposition resulted in unusable outcomes when the polynomial exceeded 15 degrees.

QR - After Scaling:

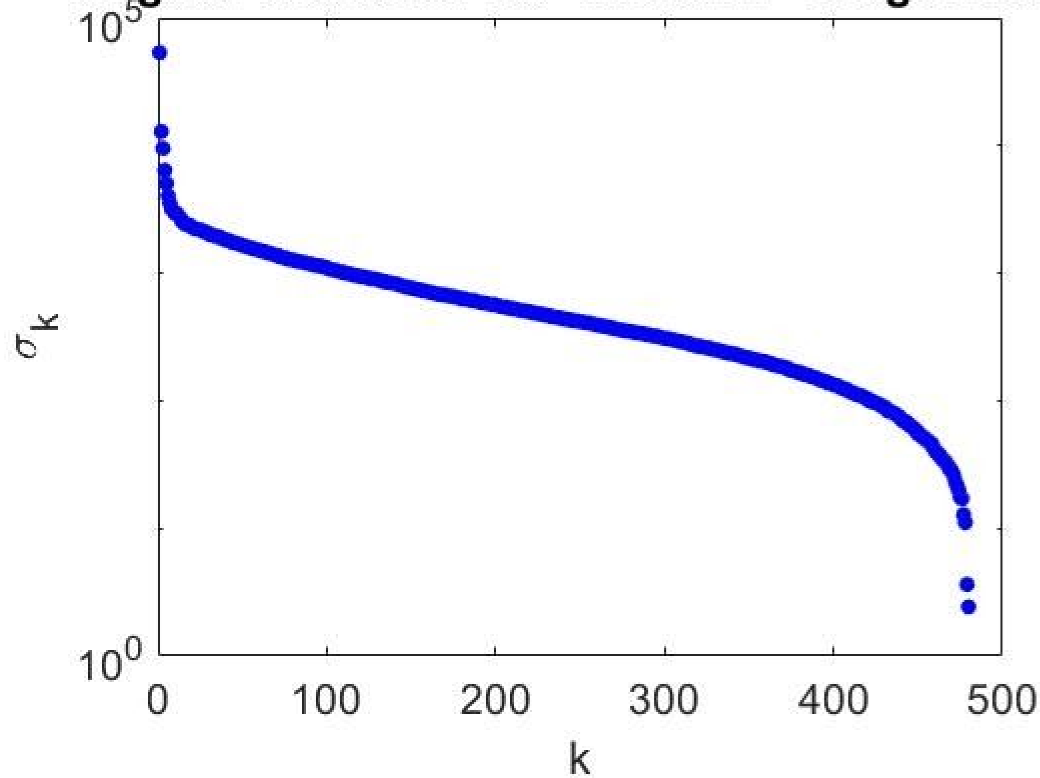



```
imeth= 1 n=82 m=9 l(errlsq) = 1.039621e+00
imeth= 1 n=82 m=11 l(errlsq) = 1.011977e+00
imeth= 1 n=82 m=13 l(errlsq) = 1.012682e+00
imeth= 1 n=82 m=15 l(errlsq) = 1.002277e+00
imeth= 1 n=82 m=17 l(errlsq) = 9.820582e-01
imeth= 1 n=82 m=19 l(errlsq) = 9.837116e-01
imeth= 1 n=82 m=21 l(errlsq) = 9.829238e-01
imeth= 1 n=82 m=23 l(errlsq) = 9.827156e-01
imeth= 1 n=82 m=25 l(errlsq) = 9.839298e-01
```

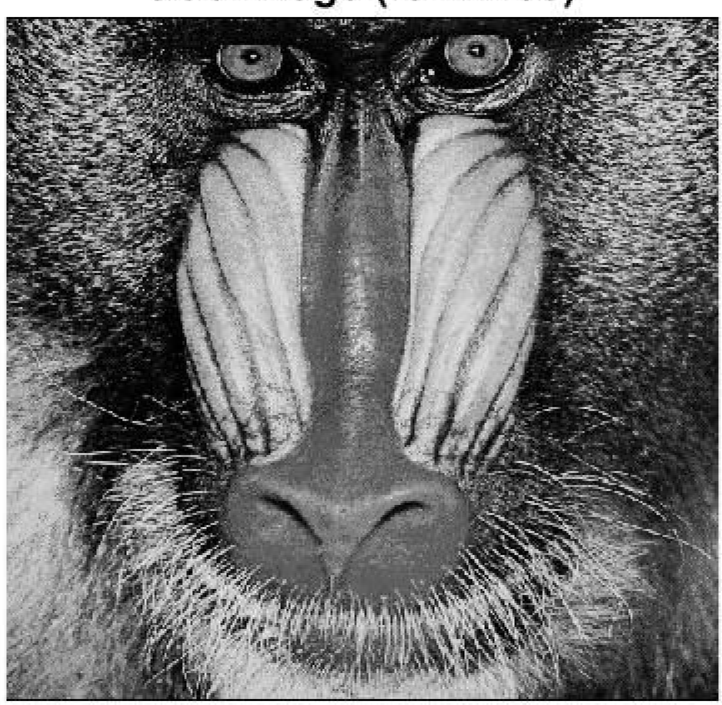
After making use of linear scaling to place all data points within a range from -1 to 1, QR decomposition was much improved. The least squares error reduced with increasing polynomial degrees, and all estimates were reasonable.

Despite QR improving, it seems that the normal equations still yield better results with unscaled values. Going off of the least squares errors, a degree of 15 when using the normal equations appears to produce the most accurate estimate of the original data. In addition, the values from this method are all very consistent, meaning that other polynomial degrees could be chosen instead and we would not be losing much accuracy.

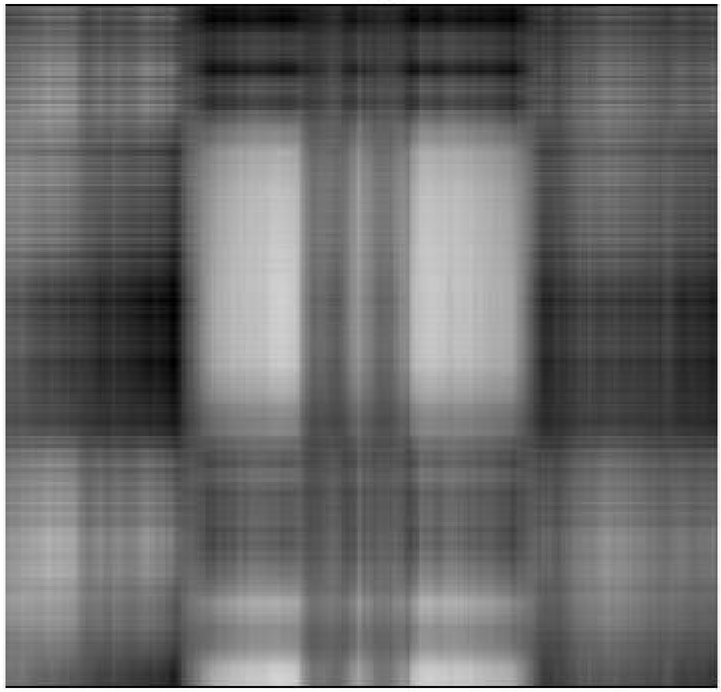
singular values of the "mandrill" image matrix



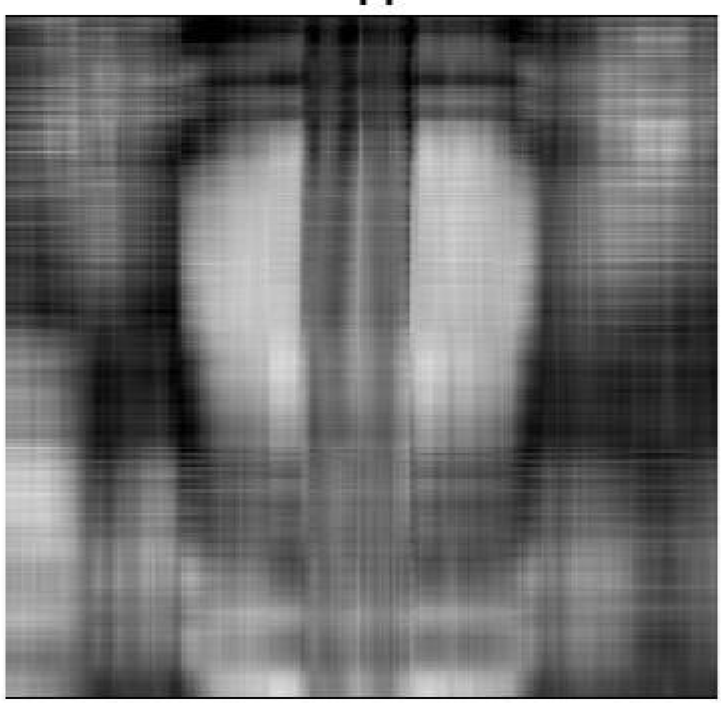
true image (rank 480)



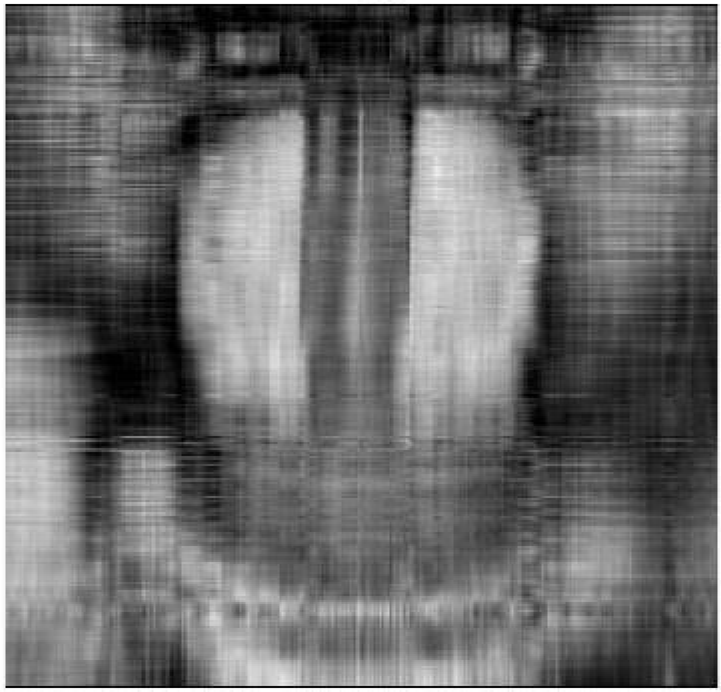
best rank-2 approximation



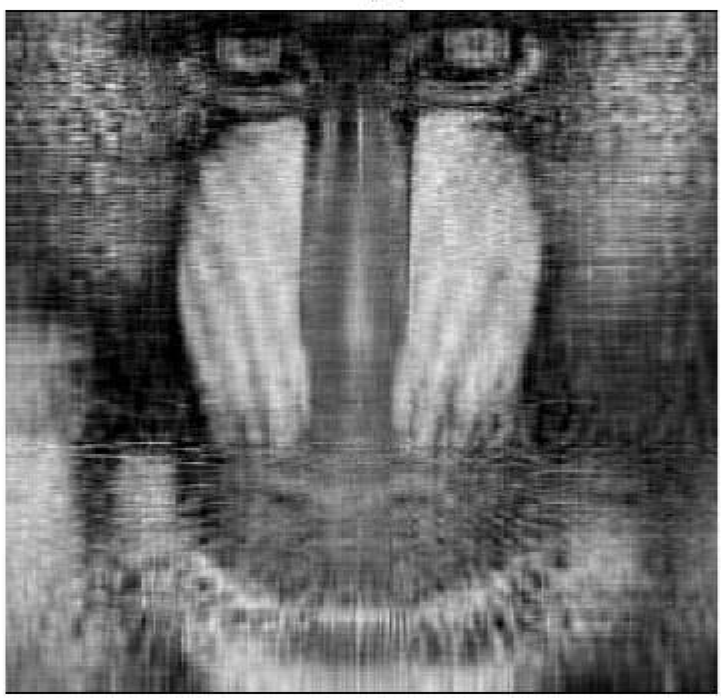
best rank-4 approximation



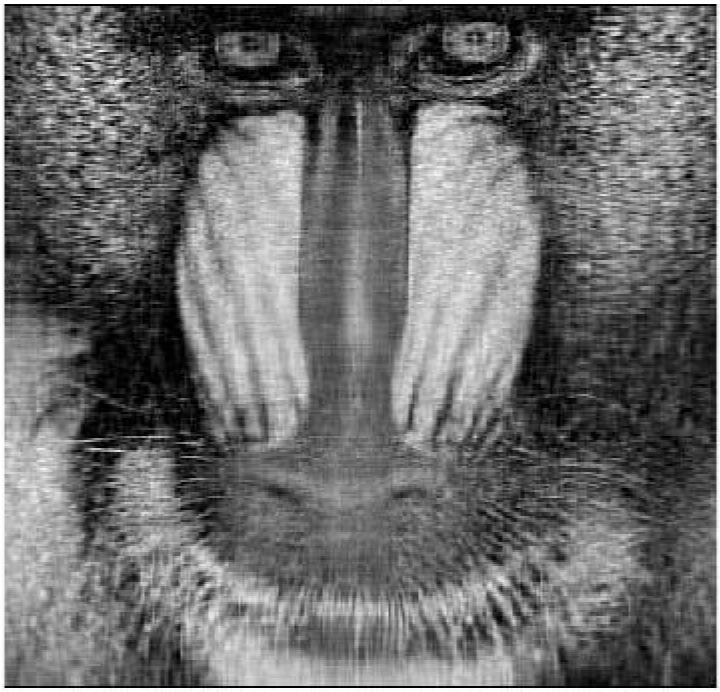
best rank-8 approximation



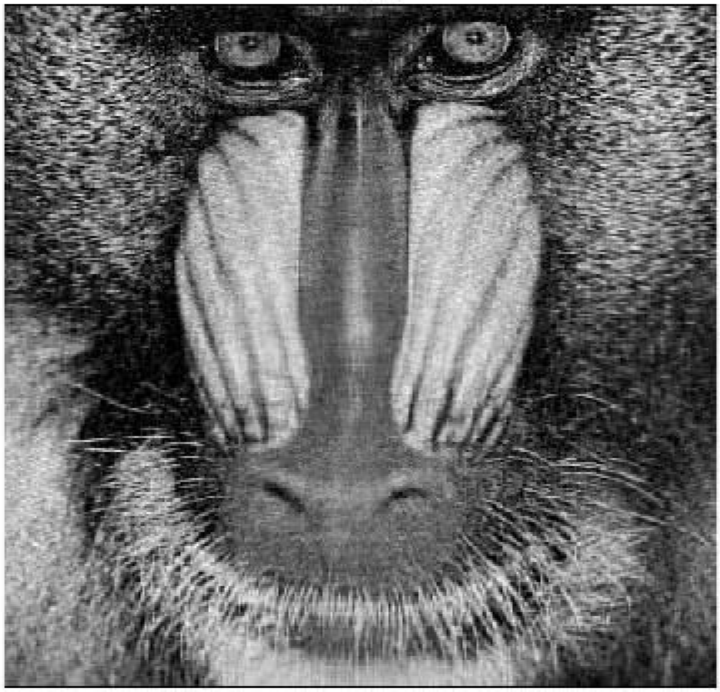
best rank-16 approximation



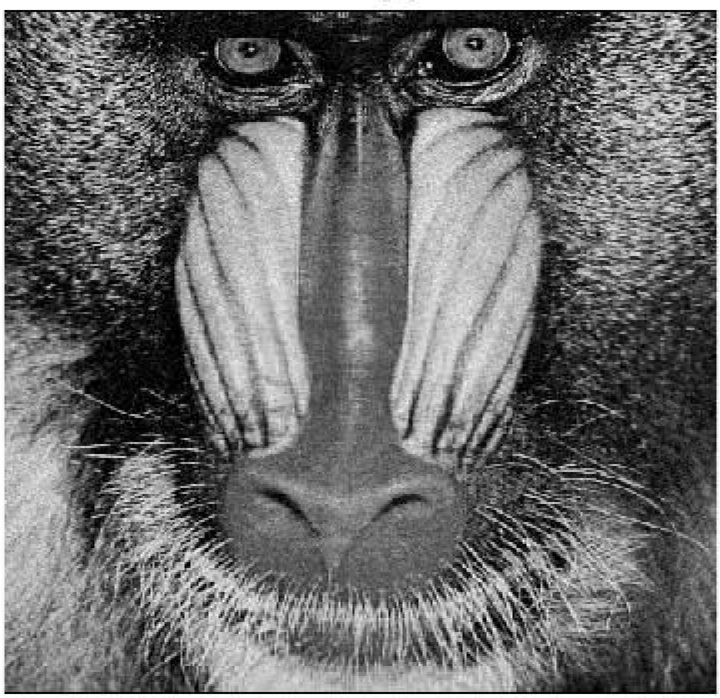
best rank-32 approximation

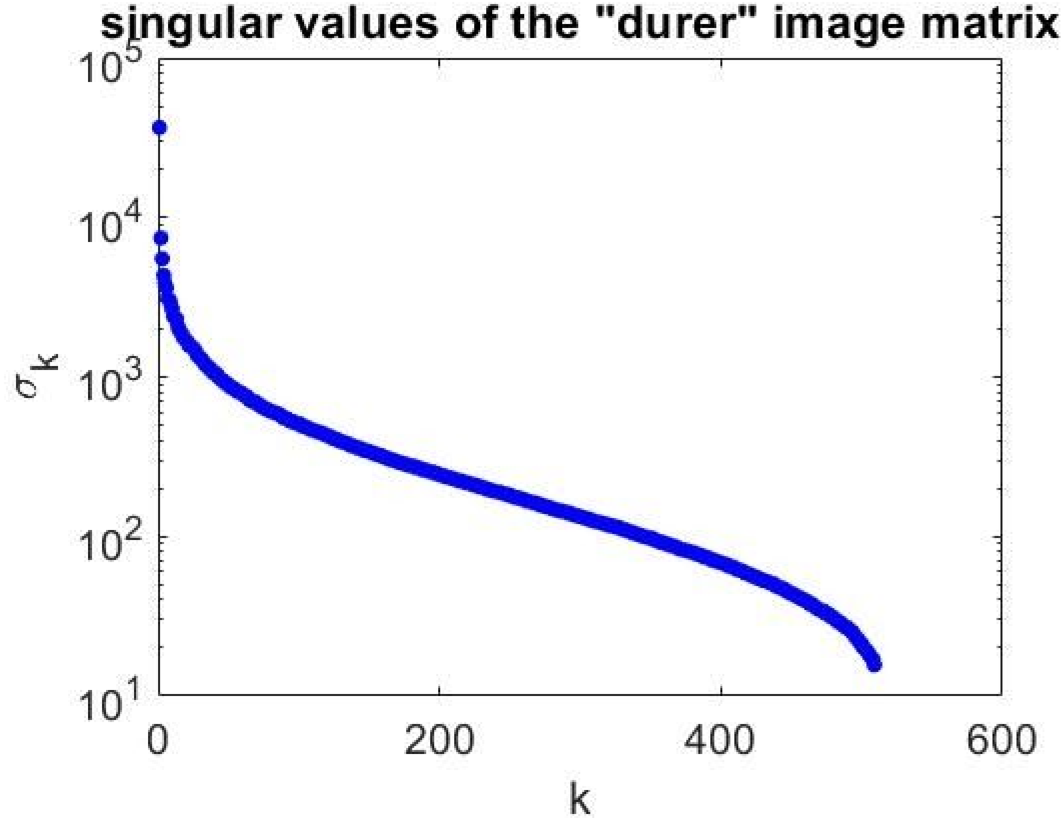


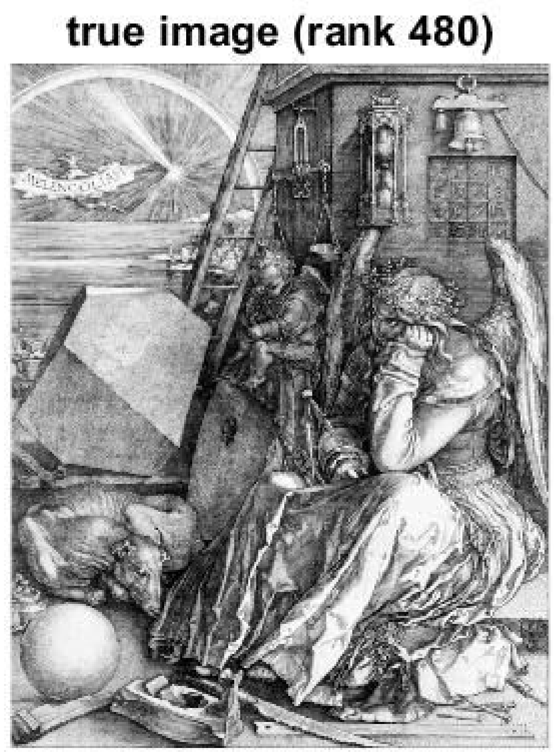
best rank-64 approximation



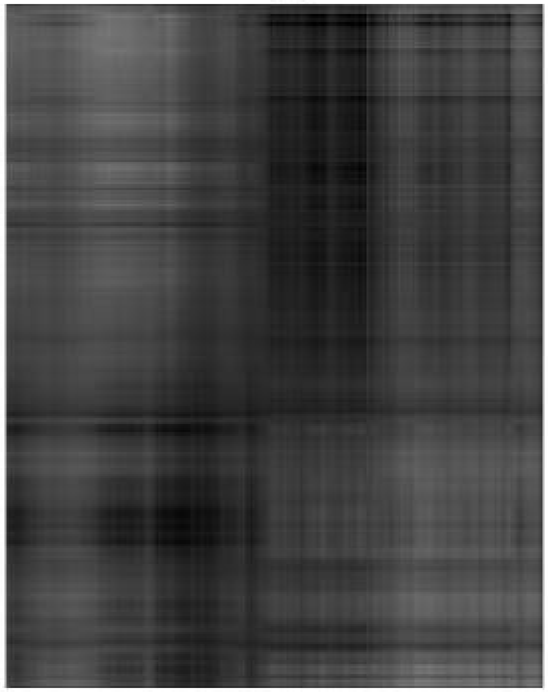
best rank-128 approximation



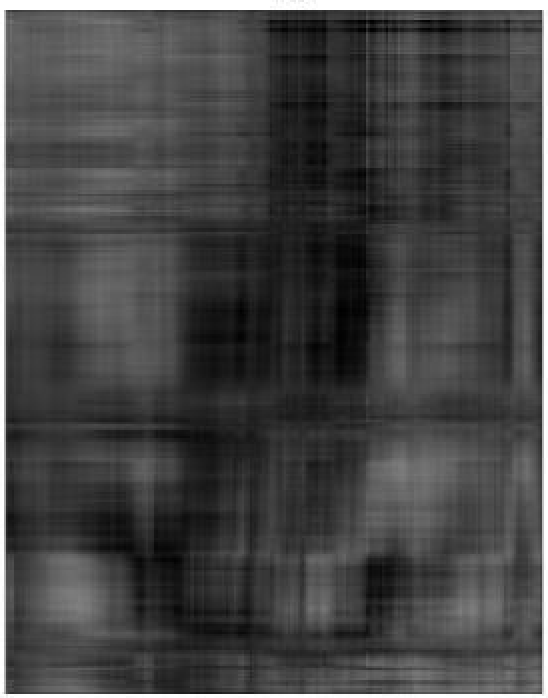

singular values of the "durer" image matrix
10^5
10^4
10^3
10^2
10^1
σ_k
0
200
400
600
k


true image (rank 480)

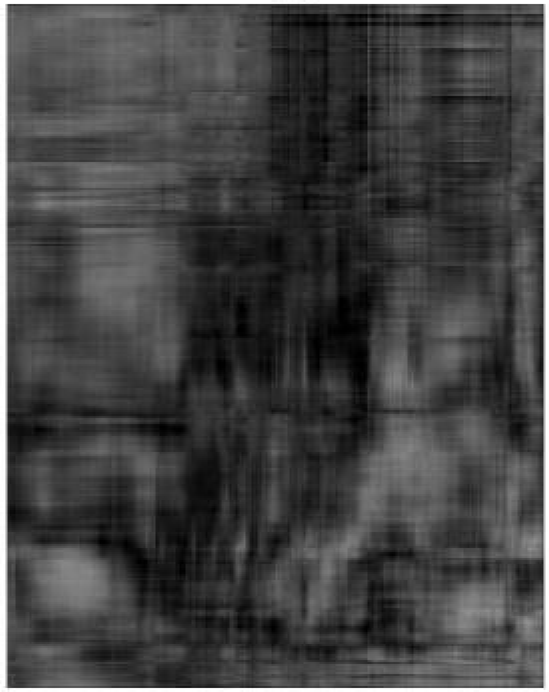
best rank-2 approximation



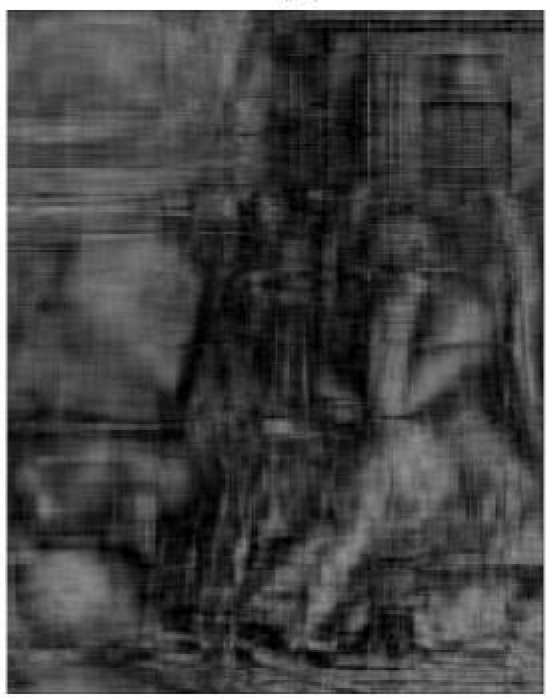
best rank-4 approximation



best rank-8 approximation



best rank-16 approximation



best rank-32 approximation

best rank-64 approximation

best rank-128 approximation

Both images begin to lose their visual coherence at around rank-16. However, based on the graphs of each image's singular values, the durer image performs better. It's average singular value is lower than that of the mandrill image, suggesting that it is less affected by the data compression. In terms of rank, each instance of compression is only using the r-most significant columns from the matrices obtained through SVD. For example, the original durer image is 648x509 pixels, meaning that 329,832 individual pixels need to be represented. But after image compression using a rank of 64 with SVD, we only require the most significant 64 columns from matrices U and V. So instead of 329,832 values needing to be stored, we can reduce that total to 74,048 ($64 * (648 + 509)$), cutting the original file down to around 22% its former size. We could continue to reduce the storage size by further reducing the rank, but obviously the quality of the image will begin to take a noticeable hit.