

```
In [1]: import numpy as np
        from scipy.io import loadmat
```

## Question 2

```
In [2]: Xtrue = loadmat("incomplete.mat")["Xtrue"]
        Y1 = loadmat("incomplete.mat")["Y1"] #least amount of missing vals
        Y2 = loadmat("incomplete.mat")["Y2"]
        Y3 = loadmat("incomplete.mat")["Y3"]
```

```
In [3]: Xtrue
```

```
Out[3]: array([[30, 12, 24,  8, 12, 14, 12, 12, 22, 24, 14, 10, 14, 24, 20, 12],
               [30, 21, 15, 11, 21, 26, 12, 21, 10, 15, 17, 16,  8, 15, 23, 12],
               [15,  9,  9,  5,  9, 11,  6,  9,  7,  9,  8,  7,  5,  9, 11,  6],
               [35, 16, 26, 10, 16, 19, 14, 16, 23, 26, 17, 13, 15, 26, 24, 14],
               [15,  9,  9,  5,  9, 11,  6,  9,  7,  9,  8,  7,  5,  9, 11,  6],
               [25, 11, 19,  7, 11, 13, 10, 11, 17, 19, 12,  9, 11, 19, 17, 10],
               [45, 24, 30, 14, 24, 29, 18, 24, 25, 30, 23, 19, 17, 30, 32, 18],
               [30, 15, 21,  9, 15, 18, 12, 15, 18, 21, 15, 12, 12, 21, 21, 12],
               [25, 11, 19,  7, 11, 13, 10, 11, 17, 19, 12,  9, 11, 19, 17, 10],
               [20, 13, 11,  7, 13, 16,  8, 13,  8, 11, 11, 10,  6, 11, 15,  8],
               [45, 24, 30, 14, 24, 29, 18, 24, 25, 30, 23, 19, 17, 30, 32, 18],
               [30, 15, 21,  9, 15, 18, 12, 15, 18, 21, 15, 12, 12, 21, 21, 12],
               [25, 17, 13,  9, 17, 21, 10, 17,  9, 13, 14, 13,  7, 13, 19, 10],
               [40, 20, 28, 12, 20, 24, 16, 20, 24, 28, 20, 16, 16, 28, 28, 16],
               [30, 18, 18, 10, 18, 22, 12, 18, 14, 18, 16, 14, 10, 18, 22, 12],
               [25, 11, 19,  7, 11, 13, 10, 11, 17, 19, 12,  9, 11, 19, 17, 10]],
            dtype=uint8)
```

```
In [4]: def ItSingValThresh(Y, r):
        """
        Iterative Singular Value Thresholding function for Matrix Completion
        """
        tol = 10**(-3) # difference between iterates at termination
        max_its = 100;
        n,p = Y.shape
        X = np.array(Y) #make a copy so operations do not mutate the original
        X[np.isnan(X)] = 0 # Fill in missing entries with zeros

        err = 10**6
        itt = 0

        while err > tol and itt < max_its:
            U,s,VT = np.linalg.svd(X, full_matrices=False)
            V, S = VT.T, np.diag(s)
            Xnew = U[:,r:]@S[:,r:]@VT[:,r:] #first 2 cols of u, first 2 rows of
            for i in range(n):
                for j in range(p):
```

```
        if ~np.isnan(Y[i,j]): #replace Xnew with known entries
            Xnew[i,j] = Y[i,j]
    err = np.linalg.norm(X-Xnew, 'fro')
    X = Xnew
    itt += 1
    return X
```

```
In [5]: #2a)
        Xtrue = ItSingValThresh(Y1, 2)
```

```

Out[5]: array([[ 4.07434916e+01,  1.31354726e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  2.83476513e-01, -9.67220560e+00,  0.00000000e+00,
                  -4.57221767e-01, -4.25823904e-03,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  2.90966065e-01,
                  -1.36581604e-01],
                [-2.78439730e+01,  0.00000000e+00, -1.97657529e+00,
                  5.66419347e-01,  9.19646297e-01,  1.49831794e+00,
                  -3.63347685e-01,  2.57333130e+01, -2.48733069e+00,
                  0.00000000e+00,  0.00000000e+00,  8.12512071e-01,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  -2.21783906e-01],
                [-2.09524075e+00,  5.48841035e-01, -6.69404893e-01,
                  2.46130158e-01,  0.00000000e+00,  6.60280207e-01,
                  -4.34872322e-02,  6.85302142e+00, -7.87424658e-01,
                  0.00000000e+00,  0.00000000e+00,  3.70674441e-01,
                  0.00000000e+00, -1.31010554e-01,  1.71423941e-01,
                  0.00000000e+00],
                [ 3.03231866e+01,  1.10120347e+00, -4.09538806e-01,
                  0.00000000e+00, -1.15272034e-01,  3.43546726e-01,
                  1.39923444e-01, -2.70048713e+00,  0.00000000e+00,
                  -1.69776662e-01,  0.00000000e+00,  2.50633073e-01,
                  -4.91009795e-01,  0.00000000e+00,  0.00000000e+00,
                  -2.10015219e-01],
                [-3.50478129e+00,  0.00000000e+00,  0.00000000e+00,
                  1.92720832e-01,  2.07167992e-01,  4.35777883e-01,
                  -5.74349176e-02,  7.39470942e+00, -5.99112219e-01,
                  0.00000000e+00,  0.00000000e+00,  2.57991860e-01,
                  -2.75887768e-01,  0.00000000e+00,  0.00000000e+00,
                  -7.12227645e-02],
                [ 1.87041436e+01,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00, -8.93994241e-01,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  8.69745050e-01,
                  0.00000000e+00, -3.77535966e-01,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00, -2.07926014e-01,
                  -1.87928027e-01],
                [ 0.00000000e+00, -7.65134387e-01, -2.84872247e-01,
                  -2.33076365e-01, -1.46990686e+00, -1.38852378e+00,
                  0.00000000e+00,  1.46024487e+01,  2.36983990e-01,
                  1.43182482e+00, -8.48552693e-01,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  -5.15085226e-01],
                [ 2.04378372e+01,  1.58067106e+00,  0.00000000e+00,
                  4.48608144e-01,  0.00000000e+00,  0.00000000e+00,
                  1.89845037e-01,  0.00000000e+00,  0.00000000e+00,
                  -4.13261482e-01,  0.00000000e+00,  7.39874148e-01,
                  0.00000000e+00,  0.00000000e+00,  5.36693150e-01,
                  0.00000000e+00],
                [ 1.88726736e+01,  0.00000000e+00,  3.12330148e-01,
                  0.00000000e+00, -9.02881551e-01,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00, -4.01483707e-01,  0.00000000e+00,
                  9.04113679e-02,  0.00000000e+00, -2.43642364e-01,
                  -2.13806110e-01],
                [-1.46823099e+01,  0.00000000e+00, -7.06499761e-01,
                  0.00000000e+00,  7.59373133e-02,  0.00000000e+00,

```

```

-1.97540158e-01, 1.46736643e+01, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 1.90852566e-01,
-2.64762390e-01, 3.35433758e-01, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 1.51743122e+01, 0.00000000e+00,
2.26348241e+00, 0.00000000e+00, -2.30860517e-01,
4.56200686e-01, 1.98782076e+00, 0.00000000e+00,
0.00000000e+00],
[ 2.00033761e+01, 0.00000000e+00, -1.07481101e+00,
3.24429391e-01, 4.50678758e-01, 1.01712406e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
-8.36031441e-01, 0.00000000e+00, 0.00000000e+00,
-2.20372197e-01],
[ 0.00000000e+00, 3.80728221e+00, 0.00000000e+00,
1.56400935e+00, 0.00000000e+00, 4.85472992e+00,
0.00000000e+00, 1.34949840e+01, -3.25258663e+00,
0.00000000e+00, 1.61197729e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[-3.10701953e+00, -2.10322082e+00, 2.12071287e+00,
0.00000000e+00, -2.62092779e+00, -3.06955564e+00,
0.00000000e+00, 1.34451243e+01, 3.04495462e+00,
3.76427463e+00, -8.98890629e-01, -1.40531015e+00,
0.00000000e+00, 3.48784537e+00, 0.00000000e+00,
0.00000000e+00],
[-9.66289537e+00, 4.70594616e-01, 0.00000000e+00,
0.00000000e+00, 2.54871344e-01, 0.00000000e+00,
0.00000000e+00, 1.60094797e+01, -6.12220700e-01,
9.88114516e-01, 0.00000000e+00, 4.24849102e-01,
0.00000000e+00, 7.13834084e-01, 3.20934966e-01,
-8.17338983e-03],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
-6.11808933e-01, 0.00000000e+00, -3.23966792e+00,
1.65864446e-01, 5.91139468e+00, 3.72618684e+00,
3.68871548e+00, -7.57337003e-01, 0.00000000e+00,
0.00000000e+00, 3.54320083e+00, 0.00000000e+00,
9.08837492e-02]])

```

In [6]: Xtrue - ItSingValThresh(Y2, 2)

```

Out[6]: array([[ 0.00000000e+00,  6.01059580e-04,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  6.35687498e-04,  0.00000000e+00,
                -6.29990630e-04,  1.53545490e-04,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 9.52641391e-05],
               [ 0.00000000e+00,  0.00000000e+00, -1.02258736e-03,
                 0.00000000e+00,  0.00000000e+00,  2.31537157e-03,
                 0.00000000e+00,  0.00000000e+00, -2.35779034e-03,
                 0.00000000e+00,  0.00000000e+00,  5.54179835e-04,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                -1.41286522e-04],
               [ 1.56719927e-04,  0.00000000e+00,  5.59416675e-05,
                 5.58659350e-06,  0.00000000e+00,  4.19575929e-04,
                 5.70732747e-05, -1.14403411e-04,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  2.65245723e-04,  0.00000000e+00,
                 0.00000000e+00],
               [ 0.00000000e+00,  3.74389587e-04, -1.40124641e-04,
                 0.00000000e+00,  1.93825861e-04, -4.46475074e-05,
                -1.76496530e-06,  3.57848293e-04,  0.00000000e+00,
                -4.42816567e-04,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 4.77021861e-05],
               [ 1.20507247e-04,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  1.78850754e-04,  6.90971806e-04,
                 4.25886799e-05,  8.73948667e-05, -5.51683631e-04,
                 0.00000000e+00,  0.00000000e+00,  2.00295289e-04,
                -4.55975597e-05,  0.00000000e+00,  0.00000000e+00,
                 1.88970888e-05],
               [-5.24973123e-06,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  5.16749190e-05,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 4.16415716e-05],
               [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                -1.27712673e-04, -2.82744810e-04,  0.00000000e+00,
                 0.00000000e+00, -3.30702145e-04,  0.00000000e+00,
                 0.00000000e+00, -1.40293981e-04,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                -4.67652686e-05],
               [ 1.35171684e-05,  5.52730800e-05,  0.00000000e+00,
                -2.75827396e-05,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00],
               [ 0.00000000e+00,  0.00000000e+00,  1.61988111e-05,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                -4.22118679e-05,  0.00000000e+00,  9.27627657e-06,
                 3.98372035e-05],
               [-4.92993668e-06,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00])

```

```

0.00000000e+00, -4.87393996e-04, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, -5.37096006e-05,
1.65112706e-04, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 4.12953783e-04, 0.00000000e+00,
0.00000000e+00],
[ 2.15867236e-06, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, -5.85142289e-05, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
2.58546015e-05, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 5.66405884e-05, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 1.56545993e-03,
0.00000000e+00, 0.00000000e+00, -1.46097036e-03,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 6.47959652e-05,
0.00000000e+00, 0.00000000e+00, -8.11331915e-05,
-1.36730328e-04, -2.14664138e-05, 0.00000000e+00,
0.00000000e+00, 1.62275021e-04, 0.00000000e+00,
0.00000000e+00],
[ 1.80078127e-04, -4.66267985e-04, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, -6.02467530e-04, 0.00000000e+00,
6.58701178e-04, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 7.44098262e-04, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, -3.95911768e-04,
0.00000000e+00, 1.01036096e-04, 3.19093029e-04,
-2.11233746e-04, -2.46612581e-05, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
2.46178809e-05]])

```

In [7]: `Xtrue - ItSingValThresh(Y3, 2)`

```

Out[7]: array([[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00, -2.06749275e-05,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  1.09677346e-04,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 2.51033270e-05,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 0.00000000e+00,  0.00000000e+00,  9.41627285e-05,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  1.77539919e-05,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  5.14112472e-06,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  2.52004533e-05,
                  1.33700956e-05,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [-2.01394180e-05,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 0.00000000e+00, -6.02194882e-05,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00],
               [ 4.89656768e-05,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                  0.00000000e+00])

```

```

0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, -3.05136634e-04,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 4.21924336e-04,
0.00000000e+00, 0.00000000e+00, -4.51478963e-04,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, -8.47534783e-05, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00],
[ 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
-1.41677235e-06]])

```

In [8]: #2b)

*#below is the frobenious norm difference between the true and the ISV new ma  
#of the matrix with rank 3 and 2.*

```

f_e3 = np.linalg.norm(Xtrue-ItSingValThresh(Y1, 3),'fro')
f_e2 = np.linalg.norm(Xtrue-ItSingValThresh(Y1, 2),'fro')
print(f_e3, f_e2)

```

```

f_e3 = np.linalg.norm(Xtrue-ItSingValThresh(Y2, 3),'fro')
f_e2 = np.linalg.norm(Xtrue-ItSingValThresh(Y2, 2),'fro')
print(f_e3, f_e2)

```

```

f_e3 = np.linalg.norm(Xtrue-ItSingValThresh(Y3, 3),'fro')
f_e2 = np.linalg.norm(Xtrue-ItSingValThresh(Y3, 32),'fro')
print(f_e3, f_e2)

```



128.77804846771974 87.24667705099637  
48.97940976510738 0.004735599527401415  
20.785069891601946 67.57218362610462