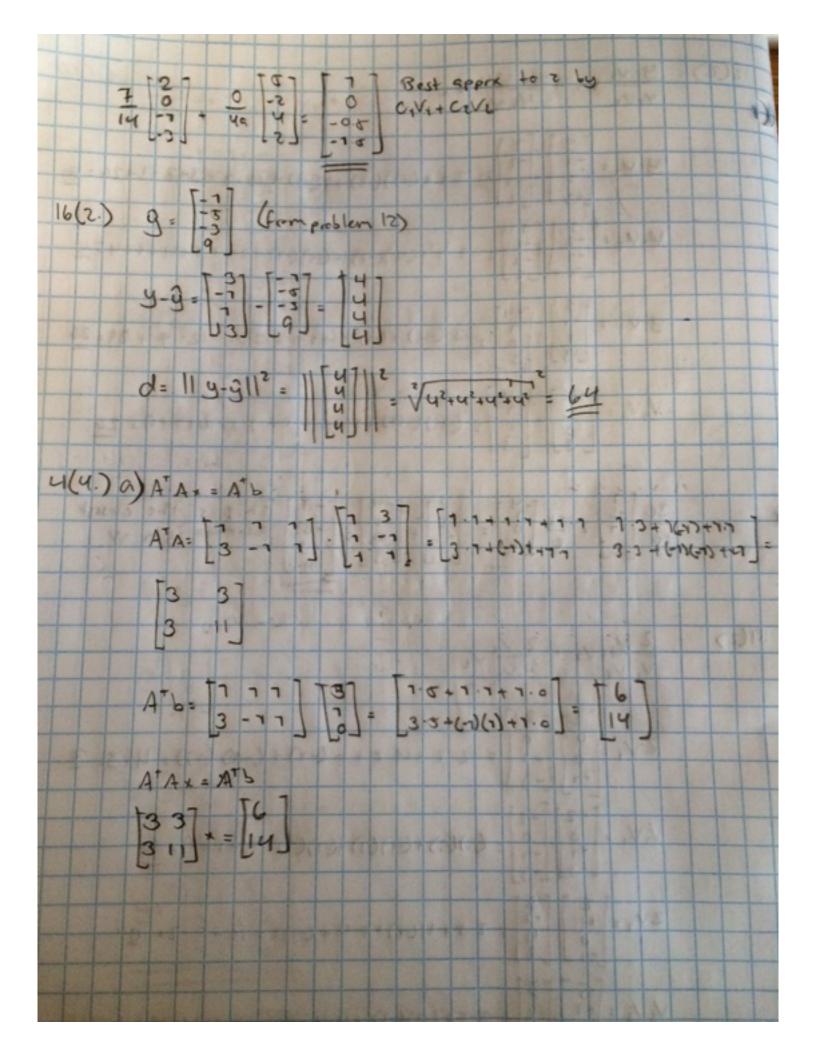
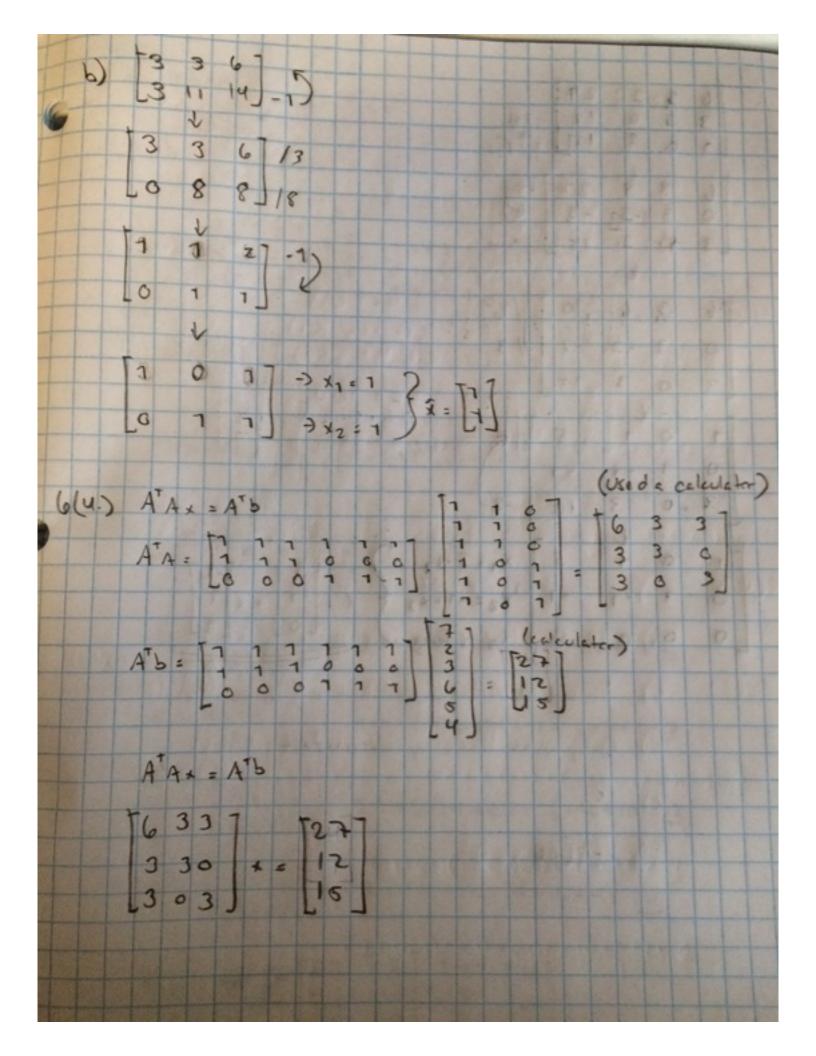
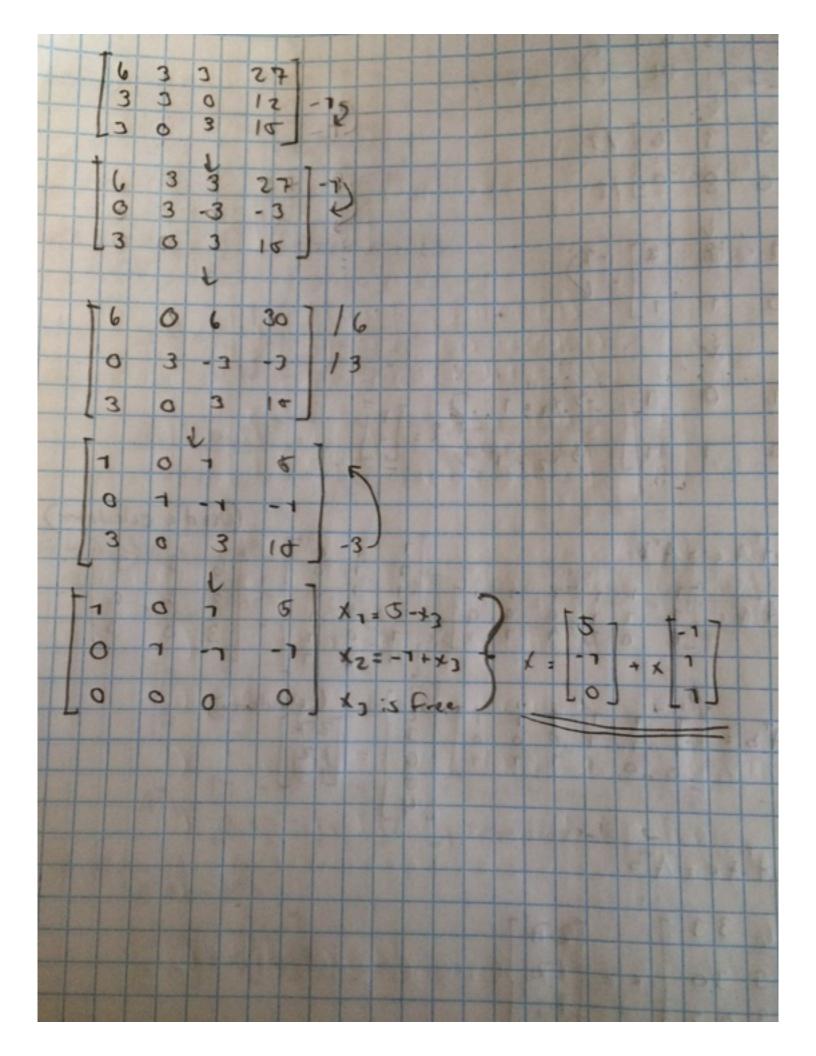
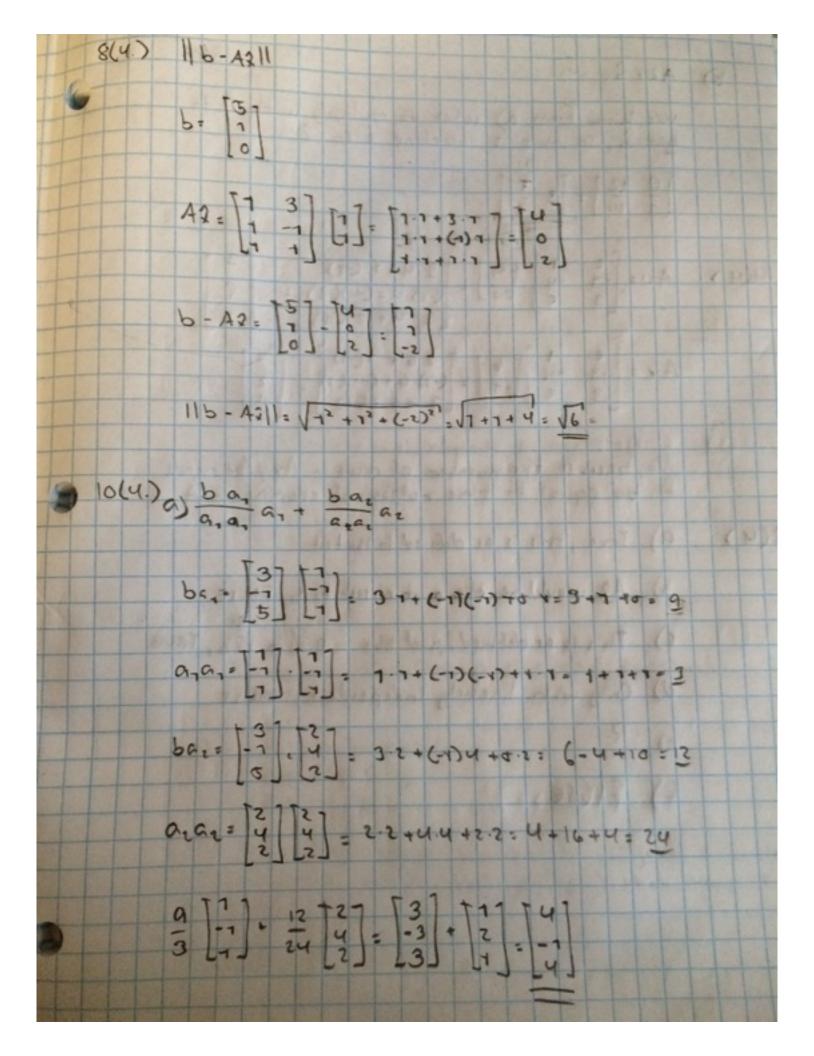


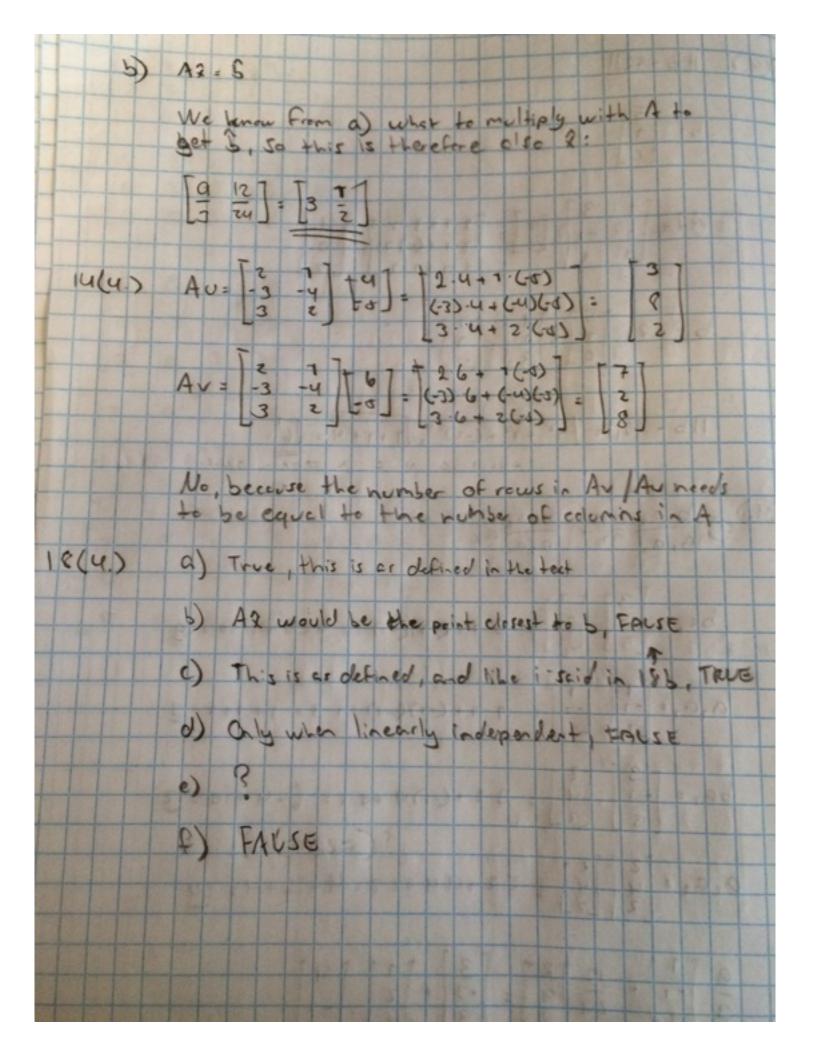
12(2.) 4. V. V. + y. V. V. y v = 3 1 -2 - 3-1+(-1)(-2)+1(-1)+13-2=3+2-1+26=30 y v = = 3 - 1 = 3 - (-4) + (-1) -1 +1 -0 -13 -3 = -12 - + + 39 = 26 30 -2 26 [3] = [3] + -4] This is the closest 14(2) 2 V, V, + 2 V, V2 2-1 = 2 - 2 = 2 - 2 + 4 0 + 0 (-1) + (-1) (-1) = 4+3= 3 V,V, = 0 -2 = (2)(2)+(4)(1)+(1)(1)+(1)(1)=4+1+9=14 V2V1 = 1-2 -2 = 5-5+(-2)(-2) +4-4+2-2 = 25+4+16+4, 49

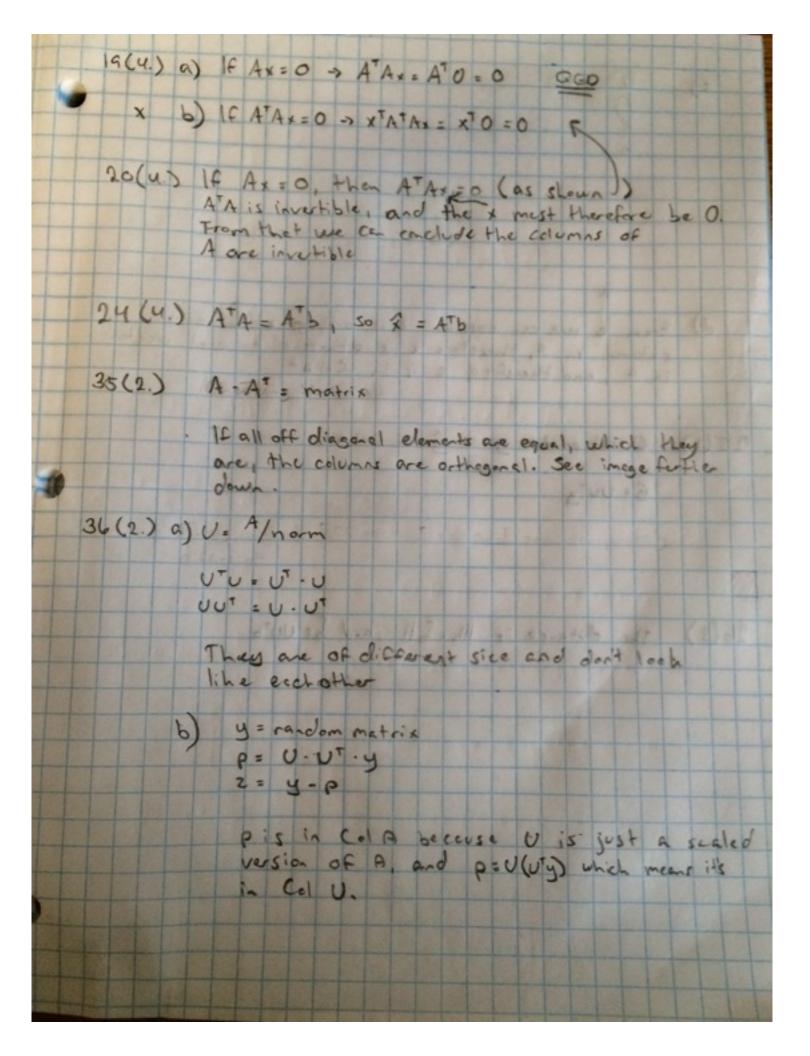


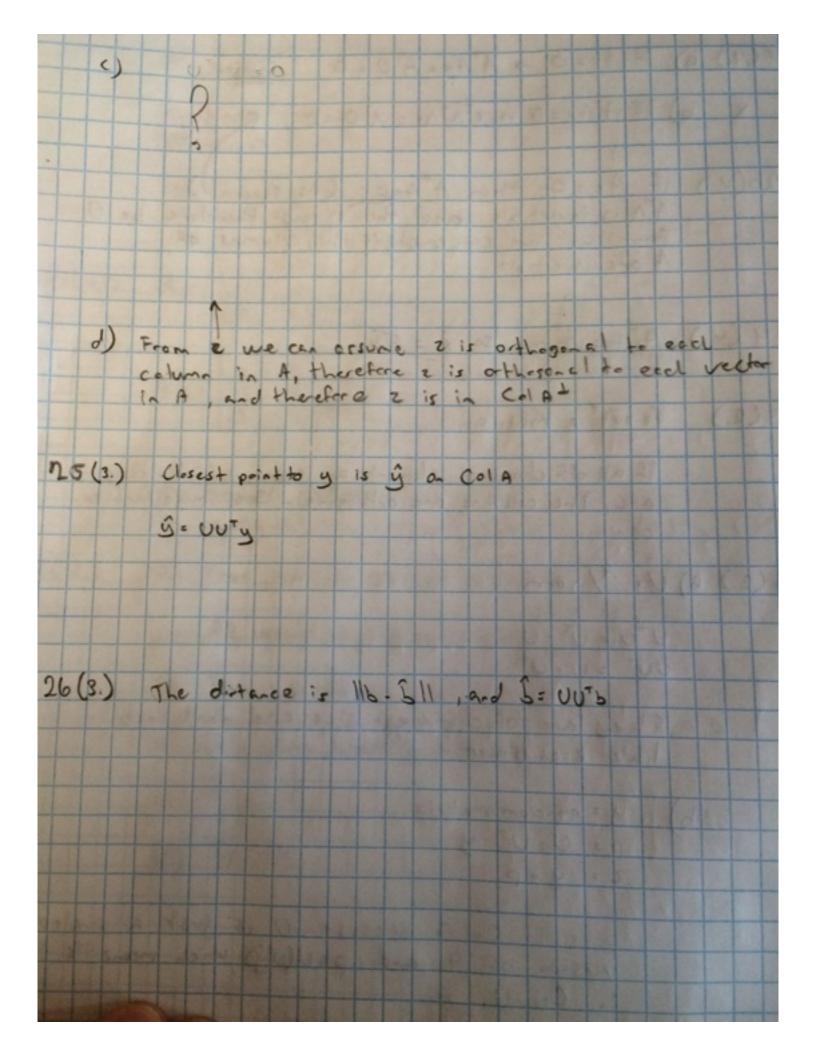












```
def problem35():
    A = np.array([[-6,-3, 6, 1],
                  [-1, 2, 1, -6],
                  [3, 6, 3, -2],
                  [6,-3,6,-1],
                  [2,-1, 2, 3],
                  [-3, 6, 3, 2],
                  [-2,-1, 2,-3].
                  [ 1, 2, 1, 6]])
    transpose = np.transpose(A)
    result = np.dot(A, transpose)
    print "Problem 35:"
    print "The columns are orthogonal if the off diagonal elements are equal"
    print result
def problem36():
   A = np.array([[-6, -3, 6, 1],
                  [-1, 2, 1, -6],
                  [ 3, 6, 3,-2],
                  [6,-3,6,-1],
                  [2,-1, 2, 3],
                  [-3, 6, 3, 2],
                  [-2,-1, 2,-3]
                  [ 1, 2, 1, 6]])
   U = np.divide(A, np.linalg.norm(A))
```

transpose = np.transpose(U)
UTU = np.dot(transpose, U)
UUT = np.dot(U, transpose)

print "Problem 36a:"

print "UTU:\n" + str(UTU)
print "UUT:\n" + str(UUT)

p = np.dot(U, transpose)

z = np.subtract(y, p)
print "\nProblem 36b:"
print "p:\n" + str(p)
print "z:\n" + str(z)

p = np.dot(p, y)

y = np.random.randint(-10, 10, size=(8, 4))

```
A = np.array([[-6,-3, 6, 1],
                 [-1, 2, 1, -6],
                 [3, 6, 3, -2],
                 [6,-3,6,-1],
                 [ 2,-1, 2, 3],
                 [-3, 6, 3, 2],
                 [-2,-1, 2,-3],
                 [ 1, 2, 1, 6]])
   U = np.divide(A, np.linalg.norm(A))
   transpose = np.transpose(U)
   UUT = np.dot(U, transpose)
   print(UUT)
   y = np.array([1, 1, 1, 1, 1, 1, 1, 1])
   yHat = np.dot(UUT, y)
   print "Problem 25:"
   print "The closest point is:\n" + str(yHat)
In [54]: problem35()
Problem 35:
The columns are orthogonal if the off diagonal elements are equal
       0 -20
                      20 24
[[ 82
               8
                   6
                               0]
[ 0 42 24
                      6
                          20 -32]
               0 -20
[-20 24
          58
              20
                      32
                           0
                   0
                               67
 Г
   8
      0
          20
              82
                  24 -20
                          6
                               0]
 Γ
   6 -20
          0
              24
                  18
                      0
                         -8
                              20]
 Γ 20
          32 -20
                  0
                      58
                          0
       6
                              247
 Γ 24
      20
           0
               6
                  -8
                      0
                         18 -207
   0 -32
           6
               0
                  20
                      24 -20 42]]
In [55]: problem36()
Problem 36a:
UTU:
[[ 2.50000000e-01 -6.93889390e-18 -8.67361738e-18
                                                      0.00000000e+00]
 [ -6.93889390e-18
                                     0.00000000e+00 -6.93889390e-18]
                    2.50000000e-01
 Γ -8.67361738e-18
                   0.00000000e+00
                                     2.50000000e-01
                                                     0.00000000e+00]
 0.0000000e+00 -6.93889390e-18 0.0000000e+00 2.50000000e-01]
UUT:
[[ 0.205 0.
                              0.015 0.05
                -0.05
                       0.02
                                            0.06
                                                   0.
          0.105 0.06
                       0.
                             -0.05
                                     0.015 0.05
                                                 -0.08 7
 Γ0.
 [-0.05
         0.06
                0.145 0.05
                              0.
                                     0.08
                                            0.
                                                   0.015]
 [ 0.02
                0.05
                       0.205 0.06 -0.05
                                            0.015
                                                   0.
         0.
 [ 0.015 -0.05
                0.
                              0.045
                                     0.
                                           -0.02
                                                   0.05 ]
                       0.06
 [ 0.05
         0.015 0.08 -0.05
                              0.
                                     0.145 0.
                                                   0.06 ]
 [ 0.06
         0.05
                0.
                       0.015 -0.02
                                     0.
                                            0.045 -0.05 ]
         -0.08
                0.015 0.
                                     0.06 -0.05
 Γ0.
                              0.05
                                                   0.105]]
```

def problem25():

```
Problem 36b:
p:
[[ 1.455 -0.195 -1.155 1.595]
 [-1.215 -1.165 -0.575 -1.7 ]
 [-1.38 -1.305 -1.35 -0.85]
 [ 1.395 -0.405 0.105 -1.295]
 [ 0.915  0.315 -0.015  0.685]
 [-0.945 -0.795 -2.025 1.6 ]
 [ 0.035 -0.515 -0.335 -0.585]
 [ 0.44  0.465 -0.55  1.95 ]]
z:
[[ 5.545 -0.805 -2.845 6.405]
 [-6.785 -8.835 5.575 -3.3 ]
 [-2.62 -3.695 -6.65 -1.15 ]
 [ 4.605  1.405  0.895 -5.705]
 [ 1.085 -6.315 -0.985 6.315]
 [-2.055 -0.205 -7.975 3.4 ]
 [-3.035 -1.485 -3.665 -5.415]
 [-2.44 -0.465 4.55 4.05]]
In [58]: problem25()
Problem 25:
The closest point is:
[ 0.3 0.1 0.3 0.3 0.1 0.3 0.1 0.1]
```