Problem 3

In [45]:
```
import numpy as np
import time

A = np.random.normal(0, 1, (5000, 5000))
b = np.random.normal(0, 1, (5000, 1))
```

In [46]:
```
def sol1(A, b):
    s = time.time()
    x = np.linalg.solve(A, b)
    e = time.time()
    return e-s

sol1(A, b)
```

Out[46]:
```
0.5387320518493652
```

In [47]:
```
def sol2(A, b):
    s = time.time()
    A_inv = np.linalg.inv(A)
    x = np.matmul(A_inv, b)
    e = time.time()
    return e-s
```

In [48]:
```
sol2(A, b)
```

Out[48]:
```
1.589203119277954
```

using `np.solve` is roughly 2 times faster in terms of running time.

Problem 4

In [49]:
```
import numpy as np

A = np.random.normal(0, 1, (100, 100))
u, s, vh = np.linalg.svd(A, full_matrices=True)
s_dagger = 1/s
```

In [50]:
```
np.allclose(np.transpose(vh).dot(np.diag(s_dagger, k=0)).dot(np.transpose(u)), np.linalg.pinv(A))
```

Out[50]:
```
True
```

Problem 7

(a)

In [51]:
```
a = np.random.normal(0, 1, 500)
b = np.random.normal(0, 1, 500)
```

In [52]:
```
def l2(v):
    return np.sqrt(np.sum(np.square(v)))
```

In [53]:
```
np.allclose(np.linalg.norm(a), l2(a))
```

Out[53]:
```
True
```

(b)

In [54]:
```
1/500*np.linalg.norm(a-b)
```

Out[54]:
```
0.06047165654676527
```

Problem 10

In [55]:
```
# (a)
import numpy as np

N = [10, 100, 1000]

for n in N:
    print(n)
    A = np.random.normal(0, 1, (n,n))
    print(1/n * A.dot(np.transpose(A)))
```

```
10
[[ 1.40650755 -0.32274061  0.52457356  0.36761455  0.07367104  0.2346426
   0.12443221 -0.38801899  0.90601489  0.01211126]
 [-0.32274061  0.54862164 -0.17571886 -0.20636088 -0.17055541 -0.15061271
   0.25774497 -0.1829213  -0.49848589 -0.24021468]
 [ 0.52457356 -0.17571886  1.05226245  0.25528738 -0.08936804 -0.09218026
  -0.1767688   0.16410137  0.32869132  0.31013734]
 [ 0.36761455 -0.20636088  0.25528738  0.54853218  0.07374337  0.26453728
  -0.09220435 -0.00577407  0.36565624  0.20839377]
 [ 0.07367104 -0.17055541 -0.08936804  0.07374337  0.3965839   0.05428062
   0.04308337  0.10949603  0.2780887   0.02752397]
 [ 0.2346426  -0.15061271 -0.09218026  0.26453728  0.05428062  0.69757313
  -0.04608612  0.12270763 -0.20864225 -0.14288277]
 [ 0.12443221  0.25774497 -0.1767688  -0.09220435  0.04308337 -0.04608612
   0.55255577 -0.59540847 -0.08720362 -0.40453337]
 [-0.38801899 -0.1829213   0.16410137 -0.00577407  0.10949603  0.12270763
  -0.59540847  1.23444086  0.29054501  0.4873365 ]
 [ 0.90601489 -0.49848589  0.32869132  0.36565624  0.2780887  -0.20864225
  -0.08720362  0.29054501  2.04630441  0.59302521]
 [ 0.01211126 -0.24021468  0.31013734  0.20839377  0.02752397 -0.14288277
  -0.40453337  0.4873365   0.59302521  1.09335066]]
100
[[ 1.05868267  0.02992563 -0.01602412 ... -0.12793841  0.05678259
   0.02039327]
 [ 0.02992563  1.37169139  0.00861191 ... -0.12554726  0.00263162
   0.09672779]
 [-0.01602412  0.00861191  0.87299955 ...  0.16661814 -0.07445466
   0.01336527]
 ...
 [-0.12793841 -0.12554726  0.16661814 ...  1.36203239  0.02671174
   0.01083847]
 [ 0.05678259  0.00263162 -0.07445466 ...  0.02671174  1.03152965
   0.06183014]
 [ 0.02039327  0.09672779  0.01336527 ...  0.01083847  0.06183014
   1.14249676]]
1000
[[ 9.85688820e-01  5.56999893e-02 -3.73694946e-04 ...  5.24746065e-03
  -7.76753981e-03 -4.79992316e-03]
 [ 5.56999893e-02  1.06670962e+00 -3.05592028e-02 ...  2.23388485e-02
   8.94120282e-03  3.88993613e-02]
 [-3.73694946e-04 -3.05592028e-02  1.02078546e+00 ...  1.46442405e-02
  -8.93814954e-03 -1.82923061e-02]
 ...
 [ 5.24746065e-03  2.23388485e-02  1.46442405e-02 ...  8.99768391e-01
   4.90532195e-02 -3.98556404e-03]
 [-7.76753981e-03  8.94120282e-03 -8.93814954e-03 ...  4.90532195e-02
   1.05831676e+00  1.67648872e-02]
 [-4.79992316e-03  3.88993613e-02 -1.82923061e-02 ... -3.98556404e-03
   1.67648872e-02  1.03133845e+00]]
```

It gets close to identity matrix I as n goes to large

In [56]:
```
# (b)
n=10
M = [10, 100, 1000]

for m in M:
    print(m)
    samples = []
    for _ in range(m):
        samples.append(np.random.normal(0, 1, (n,n)))
    print(1/m*sum([A.dot(A) for A in samples]))
```

```
10
[[ 1.03638217 -0.93599024  0.67747892  1.92132169 -0.54683851  1.88093769
   0.99370384 -0.82473686 -1.22684642 -0.04248232]
 [ 1.74855647  1.97846049  0.12503271  0.26606407  0.54477877  0.48129611
  -0.5992128  -1.33844511  0.64288999  1.10491068]
 [-0.05529785 -2.02894777  1.32230082  0.35026318  1.15957027  1.79485933
  -1.04772295  0.6087262   0.66990206  0.24895119]
 [-0.50961171  0.0687055  -1.2363624   0.83843219 -0.49446009 -0.0588407
  -1.45141956 -1.01432874 -1.35947185 -0.58797739]
 [-0.46184004 -0.40162953 -0.4143332   0.50325931  0.78080594 -0.23331782
   0.63220501 -0.223553   -0.48455207  0.20616234]
 [-0.0896985  -0.76342759 -0.42750407  0.65208608 -0.09193262  0.29487608
   0.34934174  0.5552761  -0.01189022 -1.24220082]
 [ 0.66445897  0.6876502  -0.54899823  0.8855195  -1.22463109 -0.68277302
   1.08984342  1.05200418 -0.80294455  1.367485  ]
 [ 1.31059491  1.03102087  1.74612498 -1.47379568  0.03262548 -0.73607751
   1.97273786  0.374869   -1.51204853 -0.37927116]
 [ 0.50412952 -0.15447891  0.41218498  0.35746725 -0.91567378  0.46156713
  -1.49298676 -0.94988796  1.50311231 -0.59347381]
 [ 0.02495456  2.67284614 -0.05716188 -1.28355411 -1.19424485 -0.43114366
  -0.00453246  0.42502705  0.729496    2.78971646]]
100
[[ 0.94355011  0.12254689 -0.27187739  0.50821612  0.06944769 -0.07569204
   0.08840615  0.04098062 -0.03409669 -0.47849497]
 [ 0.4236466   0.91935204  0.15143813  0.16048001 -0.17022485 -0.17851235
   0.11278749 -0.18012009  0.32548839 -0.30372522]
 [-0.05317777 -0.04267354  0.48624434 -0.45158029 -0.19906039  0.00587724
  -0.17746267 -0.39180712  0.03030927  0.00353438]
 [-0.03126784  0.50067691 -0.0371507   0.37950924  0.10808528 -0.31320014
   0.04977299  0.22138221  0.78799294  0.27245482]
 [-0.64003223  0.19023125  0.48398286 -0.16712261  0.67938387  0.13053045
   0.4498163  -0.25512271 -0.17137297  0.53326061]
 [ 0.28858228  0.01360262 -0.62274922  0.54290715 -0.39533914  0.72815454
  -0.11188991  0.03000587  0.05684441 -0.81641523]
 [ 0.20755797  0.23211207 -0.80362356  0.37351636 -0.13353719  0.10487103
   0.70447003 -0.51409336  0.71065162 -0.2106397 ]
 [ 0.39612922 -0.01247442  0.29534166  0.17992007 -0.1637671  -0.14710787
   0.3021556   0.80821603  0.28950116  0.32077824]
 [ 0.67281376 -0.34201682  0.18979566  0.18695516 -0.05145348  0.11059246
  -0.64044257  0.31579285  0.30777852 -0.54689423]
 [-0.06135255  0.26243655  0.3531865   0.02658097 -0.13410605 -0.28415521
  -0.0971166   0.32055799  0.11246977  0.74861504]]
1000
[[ 1.08855653e+00 -1.09797941e-01 -2.67797316e-02  3.72145884e-02
  -1.02090936e-02 -1.84810940e-01  6.78957153e-03  2.60957504e-03
   2.30671808e-02 -1.31819893e-01]
 [-6.34809908e-02  9.60666089e-01  1.20525441e-01 -1.27914029e-02
  -2.76151965e-02 -1.58002727e-02 -1.43417765e-01 -2.47076488e-02
   4.62577293e-03 -9.81236711e-02]
 [-1.49563026e-01 -9.55173709e-02  1.02111625e+00 -3.22086344e-02
   1.60179027e-01  1.02900986e-04  1.77939725e-01 -2.92069358e-02
  -7.59883983e-02  7.32862735e-02]
 [-5.55983382e-02 -4.33420391e-02  6.26480705e-02  1.06004860e+00
  -7.13405920e-02  1.04241342e-01 -6.03795331e-02  5.53072483e-02
  -4.36525526e-02 -2.74341283e-02]
 [-4.57914538e-02 -1.32770070e-02  1.30533461e-01  6.91511520e-02
   1.16641796e+00 -9.09930439e-02 -1.04654870e-01 -3.72762519e-02
   3.21388673e-02 -4.84812532e-02]
 [-1.39133975e-01 -4.69189482e-02 -2.00560019e-03 -3.05730989e-02
  -5.05633781e-02  1.13940406e+00 -5.07976263e-02 -3.73256434e-02
  -7.18229555e-02  1.61973709e-01]
 [-1.13080374e-01  4.86059595e-02 -9.56500216e-02  5.63264253e-02
   5.47404283e-02  1.83866903e-02  9.45856974e-01  7.27750702e-02
  -4.51833429e-02  1.10102259e-02]
 [ 1.83595985e-01 -8.31421948e-02  9.94459807e-04 -8.33756526e-02
   1.88445949e-02 -7.83048462e-02 -7.59372723e-02  9.04481603e-01
  -7.67604652e-02  2.84401623e-03]
 [ 2.81509105e-02 -8.91575196e-02  6.37659096e-02 -1.86154389e-01
  -2.35106587e-02 -9.42121729e-02  7.49150923e-02  2.58836464e-01
   9.57396984e-01  8.45934068e-02]
 [ 1.70098935e-01  1.58067366e-01 -5.78091586e-02  1.21517751e-01
  -1.18511142e-01 -1.15978082e-01  5.60270920e-02 -8.19186336e-02
  -1.85018813e-02  1.03184616e+00]]
```

It gets close to identity matrix I as n goes to large