prb2_p

March 14, 2022

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

1 Sheet 2, Exercise 3

[2]:
```
def gauss_siedel(A, b, k):
    n, m = Q.shape
    D = np.reshape([Q[i][j] if i==j else 0 for i in range(n) for j in␣
 ↪range(n)], (n ,n))
    L = np.reshape([Q[i][j] if i>j else 0 for i in range(n) for j in range(n)],␣
 ↪(n ,n))
    U = np.reshape([Q[i][j] if i<j else 0 for i in range(n) for j in range(n)],␣
 ↪(n ,n))

    x = np.random.rand(n)
    for i in range(k):
        x = np.linalg.inv(D)@(b - (L + U)@x)
    return x

def poisson_mat(n, m=None):
    return 2 * np.eye(n, m) + (-1) * np.eye(n, m, k=1) + (-1) * np.eye(n, m,␣
 ↪k=-1)

# test
for n in range(5, 20):
    Q = poisson_mat(n)
    b = np.ones(n)
    x = gauss_siedel(Q, b, k=1000)
    np.testing.assert_allclose(Q@x, b, rtol=1e-5, err_msg=f'GS failed at dim -␣
 ↪{n}')
```

2 Sheet 2, Exercise 5

[6]:

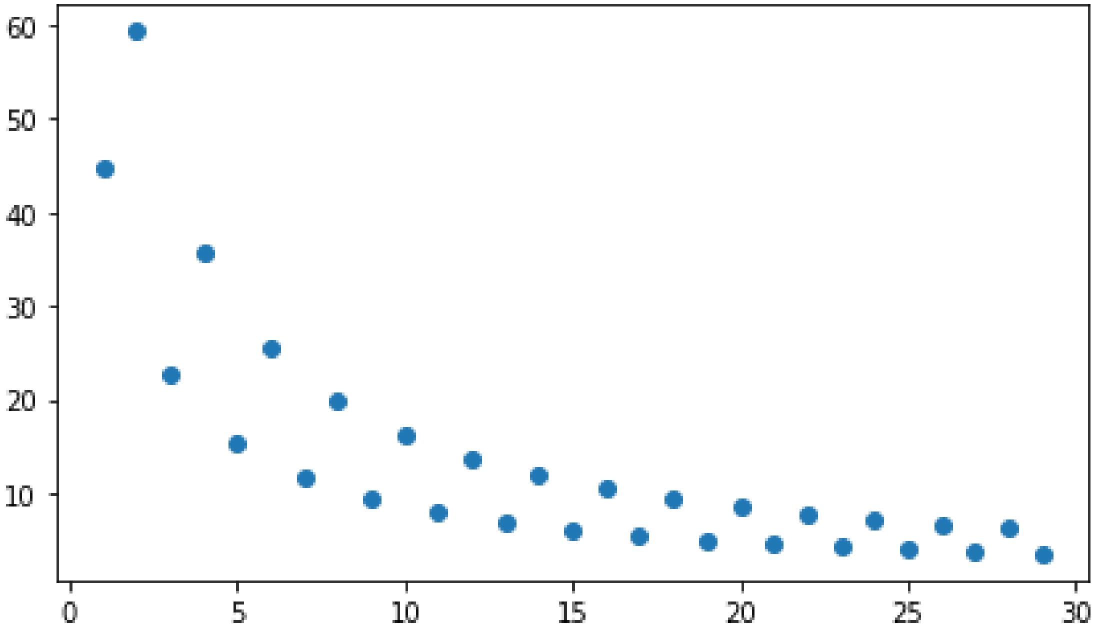
```
def neumann_polynomial_preconditioner(n, p):
    Q = poisson_mat(n)
    D = np.reshape([Q[i][j] if i==j else 0 for i in range(n) for j in␣
  ↪range(n)], (n ,n))
    N = D - Q
    C_p = np.zeros([n, n])
    for k in range(p+1):
        C_p += np.linalg.matrix_power(N @ np.linalg.inv(D), k)
    return np.linalg.inv(D) @ C_p


n = 20
Q = poisson_mat(n)
P = np.arange(1, 30)
cond_2 = []
for p in P:
    C_p = neumann_polynomial_preconditioner(n, p)
    cond_2.append(np.linalg.cond(C_p @ Q, p=2))
    print(p, cond_2[p-1], sep='\t')

plt.figure(figsize=[7, 4])
plt.scatter(P, cond_2)
print("Max. and Min. Singular value are far apart from each other for uneaven␣
  ↪p")
```

```
1       44.76606865271526
2       59.35975010638207
3       22.760834328149066
4       35.62184004487846
5       15.344146462132612
6       25.450588757787237
7       11.636387156050118
8       19.801556558635152
9       9.412478177542988
10      16.208077941288785
11      7.930495393514105
12      13.721435287552058
13      6.872470144914724
14      11.898893608722659
15      6.079418049762978
16      10.506063704904786
17      5.463014409761165
18      9.407246486889383
19      4.970264368579988
20      8.518437685556936
```

```
21      4.567443892886474
22      7.784851843354031
23      4.2320702628688585
24      7.1692347898822
25      3.948578490221603
26      6.645370572800227
27      3.705850699860457
28      6.194275175956779
29      3.495733758960119
Max. and Min. Singular value are far apart from each other for uneaven p
```



[ ]:

[ ]:

[ ]: