f-24-jupyter-potensmetoden

April 29, 2021

[1]:
```
import numpy as np
```

[2]:
```
a = np.array([[2., 1., 1.],
              [1., 3., 1.],
              [1., 1., 4.]])
a.T == a
```

[2]:
```
array([[ True,  True,  True],
       [ True,  True,  True],
       [ True,  True,  True]])
```

[6]:
```
rng = np.random.default_rng()

w = rng.standard_normal((a.shape[0], 1))
w /= np.linalg.norm(w)
n = 25
lambda_out = np.empty(n)

for i in range(n):
    v = a @ w
    w = v / np.linalg.norm(v)
    lambda_out[i] = w.T @ (a @ w)

print(lambda_out)
```

```
[4.90895603 5.18752758 5.21134835 5.21385361 5.21422858 5.21430026
 5.21431546 5.21431879 5.21431953 5.2143197  5.21431973 5.21431974
 5.21431974 5.21431974 5.21431974 5.21431974 5.21431974 5.21431974
 5.21431974 5.21431974 5.21431974 5.21431974 5.21431974 5.21431974
 5.21431974]
```

[7]:
```
w
```

[7]:
```
array([[-0.39711255],
       [-0.52065737],
       [-0.75578934]])
```

[8]:
```
a @ w - lambda_out[-1] * w
```

[8]:
```
array([[ 1.07773612e-09],
       [ 3.41649775e-09],
       [-2.91987057e-09]])
```

[9]:
```
np.linalg.eig(a)
```

[9]:
```
(array([5.21431974, 1.32486913, 2.46081113]),
 array([[ 0.39711255,  0.88765034, -0.23319198],
        [ 0.52065737, -0.42713229, -0.73923874],
        [ 0.75578934, -0.17214786,  0.63178128]]))
```

[ ]:
```

```