Numerische Mathematik

8. Übungsserie

Aufgabe 8.1:

(zum selbst Austesten siehe `aufgabe1.py`) Abgerundete Ergebnisse: 862.3, 87.1, 7.2, 2.4, 0 (bzw. $-1.1 \cdot 10^{-13}$)

```python
import numpy as np

def main():
    np.set_printoptions(precision = 53)
    A = np.array([
        [89, -11, 4, 7, -58],
        [-11, 4, -3, -1, -5],
        [4, -3, 5, -2, 1],
        [7, -1, -2, 3, 4],
        [-58, -5, 1, 4, 858]
    ], dtype = np.double)
    epsilon = 10 ** (-9)
    print(jacobi(A, epsilon))

def jacobi(A, epsilon):
    while max(gershgorin(A)) >= epsilon:
        A = jacobi_step(A)
    return A

def jacobi_step(A):
    # get dimensions of A
    if A.shape[0] != A.shape[1]:
        raise ArithmeticError("A is not symmetrical!")
    n = A.shape[0]  # with A as a n x n matrix
    # find the maximum non-diagonal value of A
    max_value = np.abs(A[0,1])
    max_i = 0
    max_j = 1
    for i in range(n):
        for j in range(i+1,n):  # only look at the upper right triangle part since A is symmetrical and we don't want to look at the diagonal
            if max_value < np.abs(A[i,j]):
                max_value = np.abs(A[i,j])
                max_i = i
                max_j = j
    # makes it much more readable
    i = max_i
    j = max_j
    # now actually calculate
    w = np.sqrt((A[i,i] - A[j,j])**2 + 4 * A[i,j]**2)
    tau = (A[i,i] - A[j,j]) / w
    sigma = np.sign(A[i,j])
    c = np.sqrt((1 + tau) / 2)
    s = sigma * np.sqrt((1 - tau) / 2)
    # make G
    G = np.identity(n, dtype = np.double)
    G[i,i] = c
    G[j,j] = c
    G[i,j] = s
    G[j,i] = -s
    # calculate next A (refered to as B)
    B = (G.dot(A)).dot(G.transpose())
    B[i,j] = 0
    B[j,i] = 0
    B[i,i] = (A[i,i] + A[j,j] + w) / 2
    B[j,j] = (A[i,i] + A[j,j] - w) / 2
    return B

def gershgorin(A):
    radii = [np.double(0)]*A.shape[0]
    for i in range(A.shape[0]):
        sum = np.double(0)
        for j in range(A.shape[1]):
            if i != j:
                sum += np.abs(A[i,j])
        radii[i] = sum
    return radii

if __name__ == "__main__":
    main()
```

Aufgabe 8.2:

$$i = k \implies L_i(t_k) = \prod_{j=0, j\neq i}^{n} \frac{t_i - t_j}{t_i - t_j} = 1$$

$$i \neq k \implies L_i(t_k) = \prod_{j=0, j\neq i}^{n} \frac{t_k - t_j}{t_i - t_j} = \frac{t_k - t_k}{t_i - t_j} \prod_{j=0, j\neq i, j\neq k}^{n} \frac{t_k - t_j}{t_i - t_j} = 0$$