Will Firmin

Math 151AH: HW2

Problem 1)

In [ ]:

```
import numpy as np

a = np.array([[0.4218,0.6557,0.6787,0.6555],
              [0.9157,0.0357,0.7577,0.1712],
              [0.7922,0.8491,0.7431,0.7060],
              [0.9595,0.9340,0.3922,0.0318]])
c = np.array([0.2769,0.0462,0.0971,0.8235])

def gaussian_elimination(A,b):
    A = A.copy()
    b = b.copy()
    for i in range(A.shape[1]):
        b[i] = b[i] / A[i,i]
        A[i,:] =  A[i,:]/A[i,i]
        if i < A.shape[1]-1:
            for j in range(i+1,A.shape[1]):
                b[j] = b[j] - A[j,i] * b[i]
                A[j,:] -= A[j,i] * A[i,:]
    for i in range(A.shape[1]-2,-1,-1):
        for j in range(i+1,A.shape[1]):
            b[i] = b[i] - A[i,j]*b[j]
            A[i,:] = A[i,:] - A[i,j] * A[j,:]
    return b


x=gaussian_elimination(a,c)
print(f"x = {x}")
```

```
x = [-1.26040472  1.40084283  2.03133593 -2.27103223]
```

Problem 2)

Part a)

WTS $Av = \lambda v \Rightarrow \rho(A)v = \rho(\lambda)v$

Suppose $\rho(A) = \sum_{i=0}^{m} c_i A^i$ for coefficients c_i. Then $\rho(A)v = \sum_{i=0}^{m} c_i A^i v$

Note that $A^i v = \lambda A^{i-1} v = \ldots = \lambda^{i-1} A v = \lambda^i v$

Thus $\rho(A)v = \sum_{i=0}^{m} c_i \lambda^i v = v \sum_{i=0}^{m} c_i \lambda^i = \rho(\lambda)v$

Part b)

$$
\begin{aligned}
\rho(A) &= \rho(P\Lambda P^{-1}) \\
&= \sum_{i=0}^{m} c_i (P\Lambda P^{-1})^i \\
&= \sum_{i=0}^{m} c_i P\Lambda^i P^{-1} \\
&= P(\sum_{i=0}^{m} c_i \Lambda^i) P^{-1} \\
&= P\rho(\Lambda)P^{-1}
\end{aligned}
$$

Since Λ is diagonal, $(\Lambda^k)_{i,j} = (\Lambda_{i,j})^k$ and its diagonal entries of Λ are the eigenvalues of A, therefore $\rho(\Lambda) = \mathrm{diag}(\rho(\lambda_1), \ldots, \rho(\lambda_n))$.

Part c)

$$
\begin{aligned}
e^A &= e^{P\Lambda P^{-1}} \\
&= \sum_{k=1}^{\infty} \frac{1}{k!} (P\Lambda P^{-1})^k \\
&= \sum_{k=1}^{\infty} \frac{1}{k!} P\Lambda^k P^{-1} \\
&= P(\sum_{k=1}^{\infty} \frac{1}{k!} \Lambda^k) P^{-1} \\
&= Pe^{\Lambda}P^{-1}
\end{aligned}
$$

Since Λ is diagonal, $e^{\Lambda} = \mathrm{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n})$ as before in part b.

Problem 3)

Part a)

$$
\begin{aligned}
\|x+y\|_1 &= \sum_{i=1}^{n} |x_i + y_i| \\
&\leq \sum_{i=1}^{n} (|x_i| + |y_i|) \\
&= \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |y_i| \\
&= \|x\|_1 + \|y\|_1
\end{aligned}
$$

Part b)

$$
\begin{aligned}
\|x+y\|_\infty &= \max_i |x_i + y_i| \\
&= |x_I + y_I| && \text{for some } I \leq n \\
&\leq |x_I| + |y_I| \\
&\leq \max_i |x_i| + \max_i |y_i| \\
&= \|x\|_\infty + \|y\|_\infty
\end{aligned}
$$

Problem 4)

Part a)

For $u(t) = \frac{t^p}{p} + \frac{1}{q} - t$,

$$
\begin{aligned}
\frac{du}{dt} &= t^{p-1} - 1 = 0 \Rightarrow t = 1 \\
\frac{d^2u}{d^t} &= (p-1)t^{p-2} = p - 1 > 0 \text{ at } t = 1 \\
u(1) &= 0, u(0) = \frac{1}{q} > 0 \\
\Rightarrow 1 &= \textcolor{red}{\backslash\text{argmin}}_t u(t) \\
\text{For } t^* = ab^{-\frac{q}{p}}, \\
u(1) &\leq u(t^*) \\
\Rightarrow 0 &\leq \frac{a^p b^{-q}}{p} + \frac{1}{q} - ab^{-\frac{q}{p}} \\
\Rightarrow 0 &\leq \frac{a^p}{p} + \frac{b^q}{q} - ab^{q-\frac{q}{p}} \\
&= \frac{a^p}{p} + \frac{b^q}{q} - ab^{q(1-\frac{1}{p})} \\
&= \frac{a^p}{p} + \frac{b^q}{q} - ab^{q\frac{1}{q}} \\
&= \frac{a^p}{p} + \frac{b^q}{q} - ab \\
\Rightarrow ab &\leq \frac{a^p}{p} + \frac{b^q}{q}
\end{aligned}
$$

Part b)

$$
\begin{aligned}
\frac{1}{\|x\|_p\|y\|_q}\sum_{k=1}^{n}|x_ky_k| &= \sum_{k=1}^{n}\frac{|x_k|}{\|x\|_p}\frac{|y_k|}{\|y\|_q} \\
&\leq \sum_{k=1}^{n}\left(\frac{1}{p}\frac{|x_k|^p}{\|x\|_p^p}+\frac{1}{q}\frac{|y_k|^q}{\|y\|_q^q}\right) \\
&= \frac{1}{p}\|x\|_p^{-p}\sum_{k=1}^{n}|x_k|^p+\frac{1}{q}\|y\|_q^{-q}\sum_{k=1}^{n}|y_k|^q \\
&= \frac{1}{p}\|x\|_p^{-p}\|x\|_p^p+\frac{1}{q}\|y\|_q^{-q}\|y\|_q^q \\
&= \frac{1}{p}+\frac{1}{q} \\
&= 1 \\
\Rightarrow \sum_{k=1}^{n}|x_ky_k| &\leq \|x\|_p\|y\|_q
\end{aligned}
$$

Part c)

$$
\begin{aligned}
\|x+y\|_p^p &= \sum_{i=1}^{n}|x_i+y_i|^{p-1}|x_i+y_i| \\
&\leq \sum_{i=1}^{n}|x_i||x_i+y_i|^{p-1}+\sum_{i=1}^{n}|y_i||x_i+y_i|^{p-1} \\
&\leq \|x\|_p\left(\sum_{i=1}^{n}(x_i+y_i)^{q(p-1)}\right)^{\frac{1}{q}}+\|y\|_p\left(\sum_{i=1}^{n}(x_i+y_i)^{q(p-1)}\right)^{\frac{1}{q}} \\
&= (\|x\|_p+\|y\|_p)\left(\sum_{i=1}^{n}(x_i+y_i)^p\right)^{\frac{1}{q}} \\
&= (\|x\|_p+\|y\|_p)\left(\sum_{i=1}^{n}(x_i+y_i)^p\right)^{1-\frac{1}{p}} \\
\Rightarrow \|x+y\|_p &\leq (\|x\|_p+\|y\|_p)\frac{\sum_{i=1}^{n}(x_i+y_i)^p}{\|x+y\|_p^p} \\
&= (\|x\|_p+\|y\|_p)\left(\sum_{i=1}^{n}(x_i+y_i)^p\right)^{1-1} \\
&= \|x\|_p+\|y\|_p
\end{aligned}
$$

Problem 5)

Part a)

$v^Tw = \sum_{k=1}^{n} v_kw_k \leq \sum_{k=1}^{n}|v_kw_k| \leq \|v\|_p\|w\|_q \leq \|v\|_p$

Part b)

We are looking for w^* s.t. the above inequality holds as an equality. The inequality comes from two sources: the first is $v_k w_k \leq |v_k w_k|$, which holds equal when sign $(v_k) =$ sign (w_k). The second is Hölder's inequality, which holds as an equality when Young's inequality in turn holds as an equality. This occurs when $ab^{-\frac{q}{p}} = 1$. Translating to the current problem, this condition becomes

$$\frac{|v_k|}{\|v\|_p}(\frac{|w_k|}{\|w\|_q})^{-\frac{q}{p}} = 1$$

Assuming for now that $\|w\|_q = 1$, this gives that

$$|w_k| = (\frac{|v_k|}{\|v\|_p})^{\frac{p}{q}}$$

The first condition dictates that sign $(w_k) =$ sign (v_k).

Now I confirm that $\|w\|_q = 1$ holds: $$\begin{aligned} |w|*q&=(\sum*{i=1}^n|w*k|^q)^\frac1q\ &=(\sum*{i=1}^n(\frac{|v_k|}{|v|_p})^p)^\frac1q\ &=|v|*p^{-\frac pq}(\sum*{i=0}^n|v_k|^p)^\frac1q\ &=|v|_p^{-\frac pq}(|v|_p^p)^\frac1q\ &=1

\end{aligned}$$

Part c)

Part a shows that for the region that we are maximizing $v^T w$ over, the $v^T w \leq \|v\|_p$. Therefore the maximized value must also be less than or equal to $\|v\|_p$. Part b shows that there exists w such that the value $\|v\|_p$ is attained. Since all greater values have been shown to be unattainable, we can conclude that $\|v\|_p$ is the maximum.

Problem 6)

Let $p(x) = \sum_{i=0}^n c_i x^i \in P_n$ for $c_i \in \mathbb{R}$ be represented by coordinates $c = (c_0, \dots c_n) \in \mathbb{R}^n$ with respect to the given basis.

Then $\frac{d}{dx}p(x) = \sum_{i=0}^n i c_i x^{i-1} = \sum_{i=0}^{n-1}(i+1)c_{i+1}x^i$

can be represented by $\hat{c} = (\hat{c}_0, \dots \hat{c}_n)$ where $\hat{c}_i = (i+1)c_{i+1}$ for $i < n$ and $\hat{c}_n = 0$.

The derivative operator is linear: let $p, q \in P_n$. Then for $\alpha, \beta \in \mathbb{R}$,

$$\begin{aligned}\frac{d}{dx}(\alpha p(x)+\beta q(x)) &= \frac{d}{dx}(\alpha\sum_{i=0}^{n}p_i x^i+\beta\sum_{i=0}^{n}q_i x^i)\\ &= \alpha\sum_{i=0}^{n}ip_i x^{i-1}+\beta\sum_{i=0}^{n}iq_i x^{i-1}\\ &= \sum_{i=0}^{n}(\alpha p_i+\beta q_i)ix^{i-1}\\ &= \sum_{i=0}^{n-1}(\alpha p_{i+1}+\beta q_{i+1})(i+1)x^i\end{aligned}$$

has coordinates with respect to $\{x^i\}_{0\leq i\leq n}$ and is thus in P_n. Furthermore, the derivative operator can be represented by matrix $A\in\mathbb{R}^{n\times n}$ where $A_{i,i+1}=i$ for $i<n+1$ and all other entries are 0. A has no eigenvalues since every nonzero polynomial loses a degree under differentiation: the "highest" nonzero coordinate becomes zero.

Problem 7)

Part a)

$$\begin{aligned}\|D\|_\infty &= \sup_{\|v\|_1=1}\|Dv\|_\infty\\ &= \max_v(\max_i\{D_{i,i}v_i\})\text{ s.t. }\sum_i|v_i|=1\\ |v_i|\leq 1 &\Rightarrow D_{i,i}v_i\leq|D_{i,i}v_i|\leq|D_{i,i}|\\ D_{i,i}v_i &= |D_{i,i}|\text{ for }v_i=\text{sign}(D_i),v_j=0\forall j\neq i\\ &\text{so }D_{i,i}v_i\text{ is maximized with }v\text{ s.t. all weight is on }D_{i,i}.\\ &\text{This makes the maximization problem a choice of }i.\\ \Rightarrow\|D\|_\infty &= \max_v\{\max_i\{D_{i,i}v_i\}\}\\ &= \max_i|D_{i,i}|\end{aligned}$$

Part b)

$$\begin{aligned}\|U\|_2 &= \sup_{v\neq 0}\frac{\|Qv\|_2}{\|v\|_2}\\ &= \sup_{v\neq 0}(\frac{<Qv,Qv>}{<v,v>})^{\frac{1}{2}}\\ &= \sup_{v\neq 0}(\frac{<v,Q^*Qv>}{<v,v>})^{\frac{1}{2}}\\ &= \sup_{v\neq 0}(\frac{<v,v>}{<v,v>})^{\frac{1}{2}}\\ &= 1\end{aligned}$$

Part c)

By Schur decomposition there exists a unitary matrix U s.t. $A = U^*TU$ for upper triangular T.

$$\begin{aligned} UAU^* &= T \\ \Rightarrow (UAU^*)^* &= T^* \\ \Rightarrow UA^*U^* &= T^* \\ \Rightarrow UAU^* &= T^* \\ \Rightarrow T &= T^* \end{aligned}$$

Since T is upper triangular, this implies that T must be diagonal.

Problem 8)

In [ ]:

```
def diagonal(A):
    n = len(A)
    D = np.zeros([n,n])
    D[np.diag_indices(n)] = A[np.diag_indices(n)]
    return D

def error(x):
    return ((x - np.array([1,2,3,4]))**2).sum()**0.5

def Jacobi(A, b, N=100):
    n = len(a)
    A2 = diagonal(A)
    A2inv = np.zeros([n,n])
    A2inv[np.diag_indices(n)] = np.reciprocal(A2[np.diag_indices(n)])
    A1 = A - A2
    x = [np.zeros(n)]
    count = 0
    while error(x[-1]) > 10**(-8) and count < N:
        x.append(A2inv.dot(b-A1.dot(x[-1])))
        count += 1
    return x

a = np.array([[2.4117,0.6557,0.6787,0.6555],
              [0.9157,1.8804,0.7577,0.1712],
              [0.7922,0.8491,3.0905,0.7060],
              [0.9595,0.9340,0.3922,2.3175]])
c = np.array([8.3813,7.6345,14.5862,13.2743])

X = Jacobi(a,c, N=200)
eJ = [np.log(error(x)) for x in X]
```

In [ ]:

```
def GaussSeidel(A, b, N=100):
    n = len(a)
    x = [np.zeros(n)]
    count = 0
    while error(x[-1]) > 10**(-8) and count < N:
        xk = np.zeros(n)
        xk1 = x[-1]
        for i in range(n):
            s1 = sum([A[i,j]*xk[j] for j in range(i)])
            s2 = sum([A[i,j]*xk1[j] for j in range(i+1,n)])
            xk[i] = 1/A[i,i]*(b[i] - s1 - s2)
        x.append(xk)
```

```
        count += 1
    return x

X = GaussSeidel(a,c,N=200)
eG = [np.log(error(x)) for x in X]

import matplotlib.pyplot as plt
plt.plot(range(len(eJ)), eJ, label="Jacobi's Method")
plt.plot(range(len(eG)), eG, label="GaussSeidel Method")
plt.legend()
plt.show()
```