MATH 437 Homework 7 (20 points)

1. Consider the system

$$x_1^2 - x_2^2 = 0,$$
$$x_1^3 + x_2^3 - 16 = 0.$$

   (a) (2 points) Write down Newton's method for this system.

   (b) (2 points) Compute 7 iterations of Newton's method for this system, starting with the initial condition $x_1 = x_2 = 1$. Report the iterations in a table with columns n, $x_1^{(n)}$, $x_2^{(n)}$, where $x_i^{(n)}$ is the value of x_i at the nth iteration and $n = 0, \ldots, 7$.

   (a) *Solution.* Let $F(x) := (x_1^2 - x_2^2,\ x_1^3 + x_2^3 - 16)^T$. Then,

$$\nabla F(x) = \begin{bmatrix} 2x_1 & -2x_2 \\ 3x_1^2 & 3x_2^2 \end{bmatrix} \implies \nabla F(x)^{-1} = \frac{1}{6x_1x_2(x_2+x_1)} \begin{bmatrix} 3x_2^2 & 2x_2 \\ -3x_1^2 & 2x_1 \end{bmatrix}$$
$$\implies \nabla F(x)^{-1}F(x) = \frac{1}{6x_1x_2(x_2+x_1)} \begin{bmatrix} 3x_2^2(x_1^2 - x_2^2) + 2x_2(x_1^3 + x_2^3 - 16) \\ -3x_1^2(x_1^2 - x_2^2) + 2x_1(x_1^3 + x_2^3 - 16) \end{bmatrix}$$

   Newton's method: start with an initial guess $x^{(0)}$ and compute

$$x^{(n+1)} = x^{(n)} - \nabla F(x^{(n)})^{-1} F(x^{(n)})$$
$$\implies$$
$$x_1^{(n+1)} = x_1^{(n)} - \frac{3(x_2^{(n)})^2((x_1^{(n)})^2 - (x_2^{(n)})^2) + 2x_2((x_1^{(n)})^3 + (x_2^{(n)})^3 - 16)}{6x_1^{(n)}x_2^{(n)}(x_2^{(n)} + x_1^{(n)})},$$
$$x_2^{(n+1)} = x_2^{(n)} - \frac{-3(x_1^{(n)})^2((x_1^{(n)})^2 - (x_2^{(n)})^2) + 2x_1((x_1^{(n)})^3 + (x_2^{(n)})^3 - 16)}{6x_1^{(n)}x_2^{(n)}(x_2^{(n)} + x_1^{(n)})}.$$

□

   (b) *Solution.* See `problem_1b.py`.

```
0 [1 1]
1 [3.33333333 3.33333333]
2 [2.46222222 2.46222222]
3 [2.08134125 2.08134125]
4 [2.0031375 2.0031375]
5 [2.00000491 2.00000491]
6 [2. 2.]
7 [2. 2.]
```

□

2. (4 points) Consider the minimization problem

$$\min_{x_1, x_2 \in \mathbb{R}} \left\{ (x_1 + x_2 - 4)^2 + (x_1^2 + x_2^2 - 8)^2 \right\}.$$

Write the gradient descent method for this problem and compute 2 iterations starting with initial guess $x_1 = x_2 = 1$.

Solution. Let $g(x) := (x_1 + x_2 - 4)^2 + (x_1^2 + x_2^2 - 8)^2$. Then

$$\nabla g(x) = \begin{bmatrix} 2(x_1 + x_2 - 4) + 2(x_1^2 + x_2^2 - 8)(2x_1) \\ 2(x_1 + x_2 - 4) + 2(x_1^2 + x_2^2 - 8)(2x_2) \end{bmatrix}.$$

Starting with an initial guess $x^{(0)}$, at the nth stage, we first find the quadratic interpolant $p(\alpha)$ to the function

$$h(\alpha) := g(x^{(n)} - \alpha \nabla g(x^{(n)}) / \|\nabla g(x^{(n)})\|_2)$$

using 3 points $\alpha_1 = 0 < \alpha_2 = \alpha_3/2 < \alpha_3$, where $h(\alpha_3) < g(x^{(n)})$. Then, we find the minimizer α_n of p and set $x^{(n+1)} := x^{(n)} - \alpha_n \nabla g(x^{(n)}) / \|\nabla g(x^{(n)})\|_2$. See `problem_2.py` for an implementation.

```
0 [1 1]
1 [2.85363005 2.85363005]
2 [2.0959484 2.0959484]
```

□

3. (4 points) Find a singular-value decomposition $A = U\Sigma V^T$ for the matrix

$$A := \begin{bmatrix} 2 & 1 \\ 1 & 0 \\ 0 & 2 \end{bmatrix}.$$

Solution. We first find the eigenvalues and eigenvectors for

$$A^T A = \begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix}$$

The eigenvalues are $\lambda = 7,\ 3$, so the singular values are $\sigma_1 = \sqrt{7},\ \sigma_2 = \sqrt{3}$. An orthonormal system of eigenvectors corresponding to these eigenvalues is

$$v_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \qquad v_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}.$$

These form the columns of V. The first 2 columns of U are computed as

$$u_1 = \frac{1}{\sigma_1} A v_1 = \begin{bmatrix} 3/\sqrt{14} \\ 1/\sqrt{14} \\ 2/\sqrt{14} \end{bmatrix}, \qquad u_2 = \frac{1}{\sigma_2} A v_2 = \begin{bmatrix} 1/\sqrt{6} \\ 1/\sqrt{6} \\ -2/\sqrt{6} \end{bmatrix}.$$

By requiring u_1, u_2, u_3 forms an orthonormal system, one arrives at the system

$$\begin{aligned} u_1 \cdot u_3 = 0 &\implies 3u_{3,1} + u_{3,2} + 2u_{3,3} = 0, \\ u_2 \cdot u_3 = 0 &\implies u_{3,1} + u_{3,2} - 2u_{3,3} = 0, \\ u_3 \cdot u_3 = 1 &\implies u_{3,1}^2 + u_{3,2}^2 + u_{3,3}^2 = 1. \end{aligned}$$

Thus,

$$\begin{aligned} u_{3,2} &= -2u_{3,1}, \\ u_{3,3} &= -\frac{1}{2} u_{3,1}, \\ u_{3,1}^2 + u_{3,2}^2 + u_{3,3}^2 &= (1 + 4 + 1/4) u_{3,1}^2 = \frac{21}{4} u_{3,1}^2 = 1 \implies u_{3,1} = \pm \frac{2}{\sqrt{21}}. \end{aligned}$$

Thus, one possible decomposition is

$$U = \begin{bmatrix} 3/\sqrt{14} & 1/\sqrt{6} & -2/\sqrt{21} \\ 1/\sqrt{14} & 1/\sqrt{6} & 4/\sqrt{21} \\ 2/\sqrt{14} & -2/\sqrt{6} & 1/\sqrt{21} \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}.$$

□

4. Let $M \geq N$. Given N vectors $a_1, \dots, a_N \in \mathbb{R}^M$, we form the $M \times N$ matrix $A := [a_1 \cdots a_N]$. Let $\sigma_1 \geq \cdots \geq \sigma_N \geq 0$ denote the singular values of A, and let $u_1, \dots, u_N \in \mathbb{R}^M$ be the columns of U in a singular-value decomposition $A = U\Sigma V^T$, where u_j corresponds to σ_j and σ_j is the ith diagonal entry of Σ. It is known that u_1 is a minimizer for

$$\min_{\substack{u \in \mathbb{R}^M \\ \|u\|_2 = 1}} \sum_{j=1}^{N} \|a_j - \langle a_j, u \rangle_2 u\|_2^2,$$

where

$$\langle v, w \rangle_2 := \sum_{i=1}^{M} v_i w_i$$

and

$$\|w\|_2^2 := \sum_{i=1}^{M} w_i^2.$$

We investigate this via a computational example.

(a) (2 points) Let $M = 80$ and $N = 20$. Construct a $M \times N$ matrix A with random entries. Perform a singular-value decomposition of A to find the unit eigenvector u_1 corresponding to the largest singular value.

(b) (1 point) Compute the projection error

$$\text{Err} := \sum_{j=1}^{N} \|a_j - \langle a_j, u_1 \rangle_2 u_1\|_2^2$$

and the relative projection error

$$E := \frac{\text{Err}}{\sum_{j=1}^{N} \|a_j\|_2^2}.$$

(c) (1 point) Construct a random unit vector $u \in \mathbb{R}^M$ and compute the projection

$$\sum_{j=1}^{N} \|a_j - \langle a_j, u \rangle_2 u\|_2^2.$$

Solution. See `problem_4.py`.

```
4a
s_1 = 20.064012032637624
s_2 = 3.7252185361056673
A = USVt ? True

4b
Err = 124.97881276589577
E = 0.23690717152965735
```

```
4c
projection = 192.97761910087667
projection >= Err ? True
```

□

5. Given $f, g : \mathbb{R}^2 \to \mathbb{R}$, consider the system

$$x = f(x, y),$$
$$y = g(x, y),$$

and the non-standard fixed-point iteration

$$x_{n+1} = f(x_n, y_n),$$
$$y_{n+1} = g(x_{n+1}, y_n).$$

(a) (2 points) Determine the convergence criteria for this method.

(b) (2 points) Let $f(x, y) = 2y$ and $g(x, y) = (x - y)/2$. Find the matrices T_1 and T_2 such that the non-standard fixed point iteration in matrix form is

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = T_1 \begin{bmatrix} x_n \\ y_n \end{bmatrix},$$

and the standard fixed-point iteration in matrix form is

$$\begin{aligned} x_{n+1} &= f(x_n, y_n), \\ y_{n+1} &= g(x_n, y_n). \end{aligned} \implies \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = T_2 \begin{bmatrix} x_n \\ y_n \end{bmatrix}.$$

Find the spectral radius of the matrices T_1 and T_2. Based on this, which method, if any, converges?

(a) *Solution.* Assuming f and g are smooth, there are $\xi, \eta \in \mathbb{R}$ such that

$$\begin{aligned} x_{n+1} - x &= f(x_n, y_n) - f(x, y) \\ &= \partial_x f(\xi, \eta)(x_n - x) + \partial_y f(\xi, \eta)(y_n - y), \\ y_{n+1} - y &= g(x_{n+1}, y_n) - g(x, y) \\ &= \partial_x g(\xi, \eta)(x_{n+1} - x) + \partial_y g(\xi, \eta)(y_n - y) \\ &= \partial_x g(\xi, \eta)(\partial_x f(\xi, \eta)(x_n - x) + \partial_y f(\xi, \eta)(y_n - y)) + \partial_y g(\xi, \eta)(y_n - y) \\ &= \partial_x g(\xi, \eta)\partial_x f(\xi, \eta)(x_n - x) + (\partial_x g(\xi, \eta)\partial_y f(\xi, \eta) + \partial_y g(\xi, \eta))(y_n - y). \end{aligned}$$

In matrix form:

$$\begin{bmatrix} x_{n+1} - x \\ y_{n+1} - y \end{bmatrix} = \begin{bmatrix} \partial_x f(\xi, \eta) & \partial_y f(\xi, \eta) \\ \partial_x g(\xi, \eta)\partial_x f(\xi, \eta) & \partial_x g(\xi, \eta)\partial_y f(\xi, \eta) + \partial_y g(\xi, \eta) \end{bmatrix} \begin{bmatrix} x_n - x \\ y_n - y \end{bmatrix}.$$

Call this matrix $T_1(\xi, \eta)$. Then, the iterative method converges if $\sup_{\xi,\eta} \|T_1(\xi, \eta)\| < 1$ for some matrix norm. More precisely, the iterative method converges if and only if $\sup_{\xi,\eta} \rho(T_1(\xi, \eta)) < 1$, where ρ is the spectral radius. □

(b) *Solution.* One could substitute for f and g and solve the linear systems to find T_1 and T_2. Alternatively, since f and g are linear in x and y, the matrix T_1 is exactly what we computed in the previous part using the partial derivatives of f and g, and, from the class notes, the matrix T_2 is

$$T_2 = \begin{bmatrix} \partial_x f & \partial_y f \\ \partial_x g & \partial_y g \end{bmatrix}.$$

Substituting for f and g:

$$T_1 = \begin{bmatrix} 0 & 2 \\ 0 & 1/2 \end{bmatrix},$$

$$T_2 = \begin{bmatrix} 0 & 2 \\ 1/2 & -1/2 \end{bmatrix}.$$

The eigenvalues of T_1 are

$$\lambda = 0,\ 1/2,$$

so $\rho(T_1) = 1/2$, and thus the non-standard iterative method converges. On the other hand, the eigenvalues of T_2 are

$$\lambda = \frac{-1/2 \pm \sqrt{1/4 + 4}}{2} = \frac{-1 \pm \sqrt{17}}{4}.$$

One of these roots is larger than 1 in absolute value, so $\rho(T_2) > 1$, and thus the standard iterative method does not converge. $\square$