Homework 2

1 Exercises 3.1

1.1 Set up the overdetermined 3 ×2 system of linear equations corresponding to the data collected.

$$\begin{bmatrix} 1 & 10 \\ 1 & 15 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11.6 \\ 11.85 \\ 12.25 \end{bmatrix}$$

1.2 Compute each possible pair of values for (x1; x2) obtained by selecting any two of the equations from the system.

$$\begin{cases} 1x_1 + 10x_2 = 11.6 \\ 1x_1 + 15x_2 = 11.85 \end{cases}$$

$$\begin{cases} 1x_1 + 10x_2 = 11.6 \\ 1x_1 + 20x_2 = 12.25 \end{cases}$$

$$\begin{cases} 1x_1 + 15x_2 = 11.85 \\ 1x_1 + 20x_2 = 12.25 \end{cases}$$

$$\text{For the first set of equations:} \quad x_1 = 11.10, \quad x_2 = 0.05$$

$$\text{For the second set of equations:} \quad x_1 = 10.95, \quad x_2 = 0.065$$

$$\text{For the third set of equations:} \quad x_1 = 10.65, \quad x_2 = 0.08$$

1.3 Set up the system of normal equations and solve it to obtain the least squares solution to the overdetermined system.

1. **Construct the Normal Equations:**

$$A^T A\mathbf{x} = A^T \mathbf{y}$$

$$A^T A = \begin{bmatrix} 1 & 1 & 1 \\ 10 & 15 & 20 \end{bmatrix} \begin{bmatrix} 1 & 10 \\ 1 & 15 \\ 1 & 20 \end{bmatrix} = \begin{bmatrix} 3 & 45 \\ 45 & 725 \end{bmatrix}$$

$$A^T \mathbf{y} = \begin{bmatrix} 1 & 1 & 1 \\ 10 & 15 & 20 \end{bmatrix} \begin{bmatrix} 11.6 \\ 11.85 \\ 12.25 \end{bmatrix} = \begin{bmatrix} 35.7 \\ 573.75 \end{bmatrix}$$

2. **Solve for x:**

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{y}$$

$$\mathbf{x} = \begin{bmatrix} 10.925 \\ 0.065 \end{bmatrix}$$

2 Exercises 3.18

2.1 How many Householder transformations are required?

A is 4×3 size, so 3 Householder transformations are required.

2.2 What does the first column of A become as a result of applying the first Householder transformation?

$$\sqrt{1^2+1^2+1^2+1^2}=2$$

$$v_1 = a_1 - \alpha e_1 = \begin{bmatrix}1\\1\\1\\1\end{bmatrix} - 2\begin{bmatrix}1\\0\\0\\0\end{bmatrix} = \begin{bmatrix}-1\\1\\1\\1\end{bmatrix}$$

$$\text{So first column} = \begin{bmatrix}-2\\0\\0\\0\end{bmatrix}$$

2.3 What does the first column then become as a result of applying the second Householder transformation?

No change, still $\begin{bmatrix}-2\\0\\0\\0\end{bmatrix}$, because H_2 just contributes to second column of A.

3 [Programming] Implement Algorithm 3.2 (Classical Gram-Schmidt Orthogonalization).

```
% Use the QR decomposition to solve the least squares problem Ax = b
% First, define the matrix A and vector b from the problem
A = [1 0 0; 0 1 0; 0 0 1; -1 1 0; -1 0 1; 0 -1 1];
b = [1237; 1941; 2417; 711; 1177; 475];

% Then, perform QR decomposition using the classical Gram-Schmidt algorithm
[Q, R] = classicalGramSchmidt(A);

% Now solve Rx = Q'b for x using back substitution
% MATLAB has a built-in function 'backslash' operator for solving such systems
x = R \ (Q' * b);

% Display the solution
disp('The solution x is:');
disp(x);
```

```
function [Q, R] = classicalGramSchmidt(A)
    [m, n] = size(A);
    Q = zeros(m, n);
    R = zeros(n, n);

    for k = 1:n
        Q(:, k) = A(:, k);
        if k ~= 1
            R(1:k-1, k) = Q(:, 1:k-1)' * A(:, k);
            Q(:, k) = Q(:, k) - Q(:, 1:k-1) * R(1:k-1, k);
        end
```

```
        R(k, k) = norm(Q(:, k));
        Q(:, k) = Q(:, k) / R(k, k);
    end
end
```

```
The solution x is:
   1.0e+03 *

    1.2360
    1.9430
    2.4160
```

4 A is a square and invertible matrix, show that $\|A^{-1}\|_2 = \frac{1}{\sigma_{\mathbf{min}}}$, where $\sigma_{\mathbf{min}}$ is the minimum singular value of matrix A.

$$\|A^{-1}\|_2 = \sigma_{\max(A^{-1})} = \sqrt{\lambda_{\max}(A^{-1,T}A^{-1})} = \frac{1}{\sqrt{\lambda_{\min}((A^{-1,T}A^{-1})^{-1})}} = \frac{1}{\sqrt{\lambda_{\min}(AA^T)}} = \frac{1}{\sigma_{\min}(A)}$$

5 Consider $f(x) = x^2 - 1$, take $x_0 = 2$ and apply Newton's method with 4 iterations and show the iteration history. Take $x_0 = -2$ and repeat the computation. What do you find?

For $x_0 = 2$:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 2 - \frac{2^2 - 1}{2 \cdot 2} = 2 - \frac{3}{4} = 1.25$$
$$x_2 = x_1 - \frac{x_1^2 - 1}{2 \cdot x_1} = 1.25 - \frac{1.25^2 - 1}{2 \cdot 1.25} = 1.25 - 0.225 = 1.025$$
$$x_3 = x_2 - \frac{x_2^2 - 1}{2 \cdot x_2} = 1.025 - \frac{1.025^2 - 1}{2 \cdot 1.025} \approx 1.0003$$
$$x_4 = x_3 - \frac{x_3^2 - 1}{2 \cdot x_3} = 1.0003 - \frac{1.0003^2 - 1}{2 \cdot 1.0003} \approx 1.00000$$

For $x_0 = -2$:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -2 - \frac{(-2)^2 - 1}{2 \cdot (-2)} = -2 + \frac{3}{4} = -1.25$$
$$x_2 = x_1 - \frac{x_1^2 - 1}{2 \cdot x_1} = -1.25 - \frac{(-1.25)^2 - 1}{2 \cdot (-1.25)} = -1.025$$
$$x_3 = x_2 - \frac{x_2^2 - 1}{2 \cdot x_2} = -1.025 - \frac{(-1.025)^2 - 1}{2 \cdot (-1.025)} \approx -1.0003$$
$$x_4 = x_3 - \frac{x_3^2 - 1}{2 \cdot x_3} = -1.0003 - \frac{(-1.0003)^2 - 1}{2 \cdot (-1.0003)} \approx -1.00000$$

The result is different for the different start point.

6 Consider $f(x) = x^2 - 1$, take $x_0 = 3$, $x_1 = 2$ and apply the secant method with 6 iterations and show the iteration history.

Given $x_0 = 3$ and $x_1 = 2$:

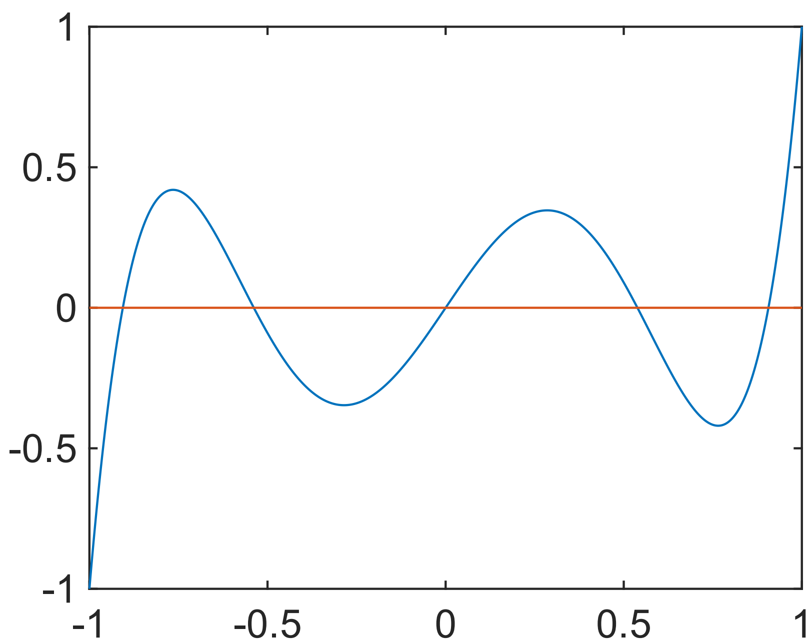
$$x_2 = x_1 - \frac{f(x_1)}{\frac{f(x_1)-f(x_0)}{x_1-x_0}} = 1.4$$
$$x_3 = x_2 - \frac{f(x_2)}{\frac{f(x_2)-f(x_1)}{x_2-x_1}} = 1.1176$$
$$x_4 \approx 1.0187$$
$$x_5 \approx 1.0010$$
$$x_6 \approx 1.0000$$
$$x_7 \approx 1.0000$$

7 [Programming] Find all five zeros of the polynomial $L_5(x) = \frac{63x^5-70x^3+15x}{8}$

```
% Define the Legendre polynomial function L5(x)
L5 = @(x) (63*x.^5 - 70*x.^3 + 15*x) / 8;

% Define the derivative of L5(x) for Newton's method
dL5 = @(x) (315*x.^4 - 210*x.^2 + 15) / 8;

% Plot
x = linspace(-1, 1, 1000);
plot(x, L5(x), x, zeros(size(x)));

% Intervals for bisection, based on the image of function
intervals = [-1, -0.7, -0.2, 0.2, 0.7, 1];

% Use bisection method to find a rough approximation of zeros
approx_zeros = zeros(1, 5);
for i = 1:5
    approx_zeros(i) = bisection(L5, intervals(i), intervals(i+1), 1e-5);
end

% Refine zeros using Newton's method
zeros_L5 = zeros(1, 5);
for i = 1:5
    zeros_L5(i) = newton(L5, dL5, approx_zeros(i), 1e-10);
end

% Display the zeros of the polynomial
disp('The zeros of the polynomial are:');
disp(zeros_L5);
```

```
function root = bisection(f, a, b, tol)
% Bisection method to find initial approximations of zeros
    if f(a) * f(b) > 0
        error('f(a)f(b)<0 not satisfied!') % ensures the root is bracketed
    end

    while (b - a) / 2 > tol
        c = (a + b) / 2;
        if f(c) == 0
```

```
            break;
        end
        if f(a) * f(c) < 0
            b = c;
        else
            a = c;
        end
    end
    root = (a + b) / 2;
end

function root = newton(f, df, x0, tol)
% Newton's method to refine the approximation
    x = x0;
    while abs(f(x)) > tol
        x = x - f(x) / df(x);
    end
    root = x;
end
```




```
The zeros of the polynomial are:
   -0.9062   -0.5385    0.0000    0.5385    0.9062
```

8 Exercises 6.4

8.1 $f(x) = x^3 + 6x^2 - 15x + 2$

Consider the function

$$f(x) = x^3 + 6x^2 - 15x + 2,$$

Its first derivative is

$$f'(x) = 3x^2 + 12x - 15,$$

The second derivative is

$$f''(x) = 6x + 12.$$

To find the critical points, we set the first derivative to zero

$$f'(x) = 0$$

which gives us the critical points

$$x_1 = -5 \quad \text{and} \quad x_2 = 1.$$

Evaluating the second derivative at these points gives

$$f''(x_1) = 6 \times (-5) + 12 = -18 < 0,$$

$$f''(x_2) = 6 \times 1 + 12 = 18 > 0.$$

Thus, $x_1 = -5$ is a maximum since $f''(x_1) < 0$, and $x_2 = 1$ is a minimum since $f''(x_2) > 0$.

Because $f(x)$ is a cubic polynomial, it goes to $\pm\infty$ as x goes to $\pm\infty$. Therefore, there is no global maximum or global minimum.

8.2 $f(x) = x^2 e^x$

Consider the function

$$f(x) = x^2 e^x.$$

Its first derivative is

$$f'(x) = 2xe^x + x^2 e^x.$$

The second derivative is

$$f''(x) = 2e^x + 2xe^x + 2xe^x + x^2 e^x = (x^2 + 4x + 2)e^x.$$

To find the critical points, we set the first derivative to zero:

$$0 = f'(x) = 2xe^x + x^2 e^x.$$

Factoring out e^x, we have

$$0 = e^x(2x + x^2).$$

So the critical points are $x_1 = -2$ and $x_2 = 0$.

Evaluating the second derivative at x_1, we get

$$f''(x_1) = -2e^{-2} < 0.$$

Therefore, $x_1 = -2$ is a maximum.

Evaluating the second derivative at x_2, we get

$$f''(x_2) = 2 > 0.$$

Therefore, $x_2 = 0$ is a minimum.

The function value at the critical points are

$$f(-2) = 4e^{-2} \quad \text{and} \quad f(0) = 0$$

Since $\lim_{x\to\infty} f(x) = \infty$ and $\lim_{x\to-\infty} f(x) = 0$, we conclude that $f(x)$ has a global minimum of 0 at $x = 0$, but does not have a global maximum.

9 [Programming] Implement Algorithm 6.1 (Golden section search).

```
% The function to minimize
f = @(x) 1 - 0.3 * x * exp(-x^2);

% Initial bracket
a = 0;
b = 2;
tol = 1e-5; % Tolerance for the convergence of the algorithm
```

```
% Call the golden_section_search function
x_min = golden_section_search(f, a, b, tol);

% Display the result
disp(['The minimum of the function is at x = ', num2str(x_min)]);
disp(['The minimum of this function is f(x) = ', num2str(f(x_min))]);
```

```
function x_min = golden_section_search(f, a, b, tol)
    % Define the golden ratio
    gr = (sqrt(5) + 1) / 2;

    % Initialize the points
    c = b - (b - a) / gr;
    d = a + (b - a) / gr;

    % Loop until the interval size is acceptable
    while abs(c - d) > tol
        % Evaluate the function at the test points
        fc = f(c);
        fd = f(d);

        % Update the points based on the function evaluations
        if fc < fd
            b = d;
            d = c;
            c = b - (b - a) / gr;
        else
            a = c;
            c = d;
            d = a + (b - a) / gr;
        end
    end

    % Choose the best point as the minimum
    x_min = (b + a) / 2;
end
```

```
The minimum of the function is at x = 0.70711
The minimum of this function is f(x) = 0.87134
```

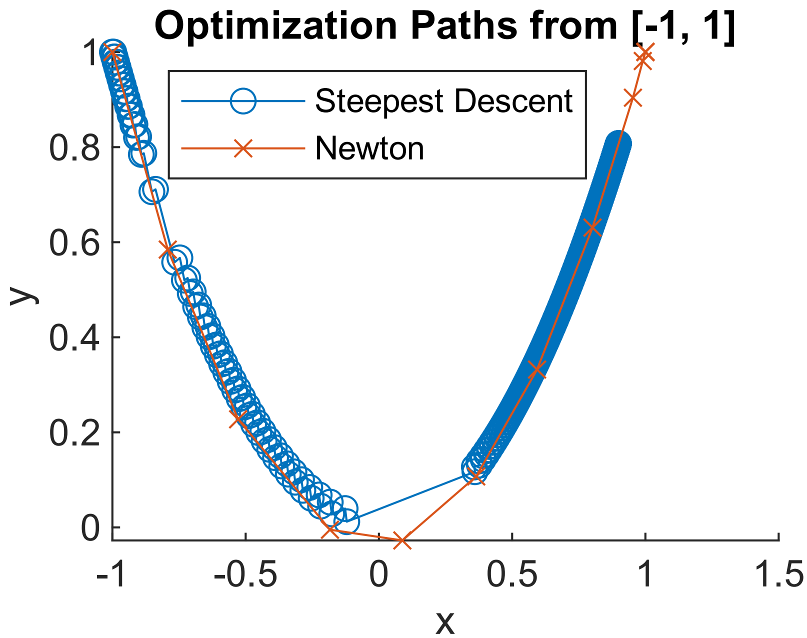
10 **[Programming]** Computer Problems 6.9 (a) and (b).

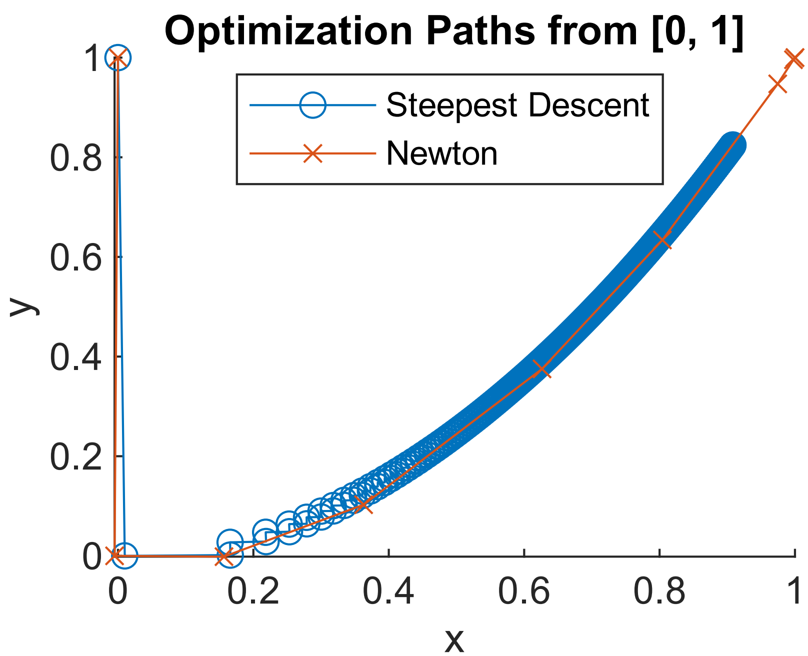
```
% Define starting points
starting_points = [-1 1; 0 1; 2 1];

% Define the Rosenbrock function and its derivatives
f = @(x) 100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;
grad_f = @(x) [-400*x(1)*(x(2) - x(1)^2) - 2*(1 - x(1));
               200*(x(2) - x(1)^2)];
hess_f = @(x) [-400*(x(2) - 3*x(1)^2) + 2, -400*x(1);
               -400*x(1), 200];
```

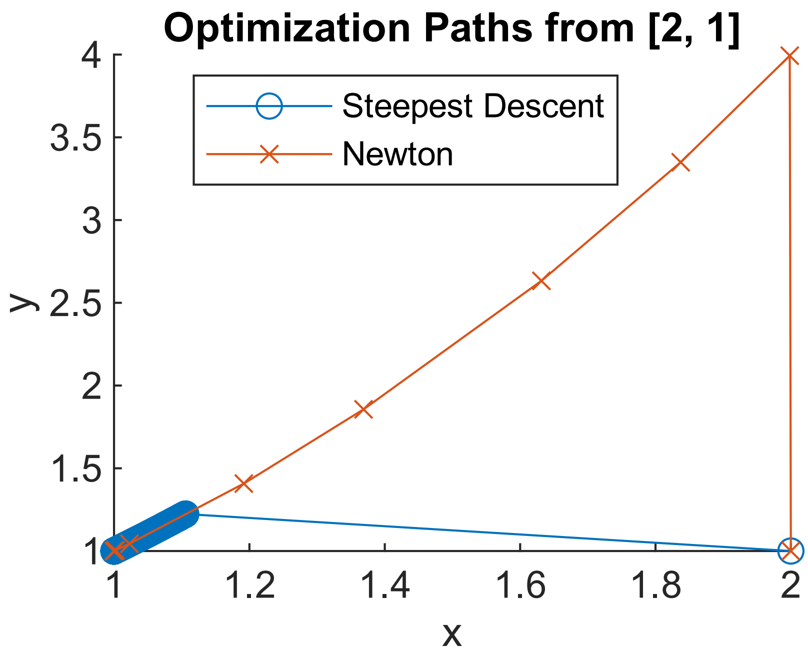
```
% Optimization parameters
max_iter = 1000;
tol = 1e-6;

% Loop over starting points
for i = 1:size(starting_points, 1)
    x0 = starting_points(i, :)';

    % Steepest Descent
    [x_sd, path_sd] = steepest_descent(f, grad_f, x0, max_iter, tol);

    % Newton's Method
    [x_newton, path_newton] = newton_method(f, grad_f, hess_f, x0, max_iter, tol);

    % Plotting
    figure;
    hold on;
    plot(path_sd(1, :), path_sd(2, :), '-o', 'DisplayName', 'Steepest Descent');
    plot(path_newton(1, :), path_newton(2, :), '-x', 'DisplayName', 'Newton');
    legend;
    title(sprintf('Optimization Paths from [%d, %d]', x0(1), x0(2)));
    xlabel('x');
    ylabel('y');
    hold off;
end
```

```
function [x, path] = steepest_descent(f, grad_f, x0, max_iter, tol)
    x = x0;
    path = x;
    for i = 1:max_iter
        grad = grad_f(x);
        if norm(grad) < tol
            break;
        end
        alpha = fminsearch(@(alpha) f(x - alpha*grad), 0);
        x = x - alpha * grad;
        path = [path, x];
    end
end

function [x, path] = newton_method(f, grad_f, hess_f, x0, max_iter, tol)
    x = x0;
    path = x;
    for i = 1:max_iter
        grad = grad_f(x);
        if norm(grad) < tol
            break;
        end
        H = hess_f(x);
        p = -H\grad; % Equivalent to solving Hp = -grad
        alpha = fminsearch(@(alpha) f(x + alpha*p), 0);
        x = x + alpha * p;
        path = [path, x];
    end
```

```
end
```


Optimization Paths from [-1, 1]
Steepest Descent
Newton
y
x
1
0.8
0.6
0.4
0.2
0
-1
-0.5
0
0.5
1
1.5


Optimization Paths from [0, 1]
Steepest Descent
Newton
y
x
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1



When $(x, y) = (1, 1)$ has minimum 0.