Exercise 1

Given the atmospheric pressure function: $p(t) = 1 - \frac{12t}{57+16(t-2)^2}$

a) To determine if $p(t)$ is unimodal, we need to find the critical points by taking the first derivative $p'(t)$ and set it to 0.

Using Wolfram Alpha, we find that: $p'(t) = \frac{864-864t+192t^2}{(57+16(t-2)^2)^2}$

Setting $p'(t) = 0$ and solving for t, we find that there is only one critical point $t = 1$ over the interval $0 \le t \le 3$.

b) To determine if the critical point is a local minimum, we use the second derivative test.

Using Wolfram Alpha, we find that: $p''(t) = \frac{96(47-48t+16t^2)}{(57+16(t-2)^2)^3}$

Evaluating $p''(1)$, we get: $p''(1) = \frac{96(47-48+16)}{(57+16(1-2)^2)^3} = \frac{96 \cdot 15}{57^3} > 0$

Since $p''(1) > 0$, the critical point $t = 1$ is a local minimum.

c) To find the minimum pressure, we evaluate $p(t)$ at the local minimum and the interval boundaries:

$p(0) = 1 - \frac{12 \cdot 0}{57+16(0-2)^2} = \frac{1}{57}$

$p(1) = 1 - \frac{12 \cdot 1}{57+16(1-2)^2} = \frac{-11}{57}$

$p(3) = 1 - \frac{12 \cdot 3}{57+16(3-2)^2} = \frac{-35}{185}$

The minimum pressure is $p(1) = \frac{-11}{57}$.

d) To find the maximum pressure, we compare the pressure values at the interval boundaries:

$p(0) = \frac{1}{57}$

$p(3) = \frac{-35}{185}$

The maximum pressure is $p(0) = \frac{1}{57}$.

e) The tornado is right on top of the weather station when the pressure is minimized. The minimum pressure occurs at $t = 1$ minute.

Exercise 2

Given the function: $f(x) = |x - 1|$ over the interval $-2 \le x \le 6$

a) Three-point equal interval search:

1st iteration: Divide the interval into 3 equal parts: $x_1 = -2$, $x_2 = 0$, $x_3 = 2$, $x_4 = 4$, $x_5 = 6$ Evaluate $f(x)$ at x_2, x_3, and x_4: $f(-2) = |-2-1| = 3$ $f(0) = |0-1| = 1$ $f(2) = |2-1| = 1$ Since $f(0) = f(2) < f(-2)$, we choose the interval $[-2, 2]$.

2nd iteration: Divide the interval into 3 equal parts: $x_1 = -2$, $x_2 = -\frac{2}{3}$, $x_3 = \frac{2}{3}$, $x_4 = 2$ Evaluate $f(x)$ at x_2, x_3, and x_4: $f(-\frac{2}{3}) = |-\frac{2}{3} - 1| = \frac{5}{3}$ $f(\frac{2}{3}) = |\frac{2}{3} - 1| = \frac{1}{3}$ $f(2) = |2-1| = 1$ Since $f(\frac{2}{3}) < f(-\frac{2}{3})$ and $f(\frac{2}{3}) < f(2)$, we choose the interval $[-\frac{2}{3}, 2]$.

3rd iteration: Divide the interval into 3 equal parts: $x_1 = -\frac{2}{3}$, $x_2 = \frac{2}{9}$, $x_3 = \frac{22}{9}$, $x_4 = 2$ Evaluate $f(x)$ at x_2, x_3, and x_4: $f(\frac{2}{9}) = |\frac{2}{9} - 1| = \frac{7}{9}$ $f(\frac{22}{9}) = |\frac{22}{9} - 1| = \frac{13}{9}$ $f(2) = |2-1| = 1$ Since $f(\frac{2}{9}) < f(\frac{22}{9})$ and $f(\frac{2}{9}) < f(2)$, we choose the interval $[-\frac{2}{3}, \frac{22}{9}]$.

After three iterations, the interval $[-\frac{2}{3}, \frac{22}{9}]$ is guaranteed to contain the value of x that corresponds to the minimum of f.

b) Successive parabolic interpolation

Successive parabolic interpolation:

Initial samples: $x_1 = 0$, $x_2 = 2$, $x_3 = 4$ Evaluate $f(x)$ at x_1, x_2, and x_3: $f(0) = |0-1| = 1$ $f(2) = |2-1| = 1$ $f(4) = |4-1| = 3$

Let's perform one iteration of successive parabolic interpolation:

$$x_{min} = x_2 - \frac{1}{2}\frac{(x_2 - x_1)^2(f(x_2) - f(x_3)) - (x_2 - x_3)^2(f(x_2) - f(x_1))}{(x_2 - x_1)(f(x_2) - f(x_3)) - (x_2 - x_3)(f(x_2) - f(x_1))}$$

Substitute the values and calculate x_{min}:

$$x_{min} = 2 - \frac{1}{2}\frac{(2)^2(1-3) - (2)^2(1-1)}{(2)(1-3) - (2)(1-1)} = \frac{5}{3}$$

Based on this one iteration, the minimum value of $f(x)$ occurs at $x = \frac{5}{3}$.

Exercise 3

a) Gradient descent with $a = 1$, $x_0 = 1$, and $\gamma = 1/4$:

$f(x) = ax^2 = x^2$

$f'(x) = 2ax = 2x$

Iteration 1: $x_1 = x_0 - \gamma f'(x_0) = 1 - \frac{1}{4}(2) = \frac{1}{2}$

Iteration 2: $x_2 = x_1 - \gamma f'(x_1) = \frac{1}{2} - \frac{1}{4}(2 \cdot \frac{1}{2}) = \frac{1}{4}$

Iteration 3: $x_3 = x_2 - \gamma f'(x_2) = \frac{1}{4} - \frac{1}{4}(2 \cdot \frac{1}{4}) = \frac{1}{8}$

Gradient descent appears to be converging.

b) Gradient descent with $a = 1$, $x_0 = 1$, and $\gamma = 1$:

Iteration 1: $x_1 = x_0 - \gamma f'(x_0) = 1 - 1(2) = -1$

Iteration 2: $x_2 = x_1 - \gamma f'(x_1) = -1 - 1(-2) = 1$

Iteration 3: $x_3 = x_2 - \gamma f'(x_2) = 1 - 1(2) = -1$

Gradient descent does not appear to be converging.

c) The second derivative of $f(x)$ is:

$f''(x) = 2a = 2$

The learning rate γ must be strictly less than $\frac{1}{L} = \frac{1}{2}$ for gradient descent to converge. In part (a), $\gamma = \frac{1}{4}$, which is less than $\frac{1}{2}$, so gradient descent converges. In part (b), $\gamma = 1$, which is not less than $\frac{1}{2}$, so gradient descent does not converge.

d) Given $f(x) = x^2$, $x_0 = 1$, $x^* = 0$, and $\gamma = \frac{1}{4}$, find k such that $|f(x_k) - f(x^*)| < 10^{-15}$:

$$\frac{(x_0 - x^*)^2}{2\gamma k} < 10^{-15}$$

$$\frac{(1-0)^2}{2 \cdot \frac{1}{4} k} < 10^{-15}$$

$$\frac{1}{2k} < 10^{-15}$$

Multiply both sides by $2k$:

$$1 < 2k \cdot 10^{-15}$$

Divide both sides by $2 \cdot 10^{-15}$:

$$k > \frac{1}{2 \cdot 10^{-15}}$$

$$k > 5 \cdot 10^{14}$$

k must be greater than $5 \cdot 10^{14}$ for the error $|f(x_k) - f(x^*)|$ to be less than 10^{-15}.

```
% Define the given function x(t)
x = @(t) 10/3 .* (exp(-t./24) - exp(-t./2));

% Define y(t) as the negation of x(t)
y = @(t) -x(t);

% Call the modified threePtSearch function to compute the minimum of y(t)
% over the interval 0 <= t <= 12, with error threshold epsilon = 1e-3
interval = [0, 12];
epsilon = 1e-3;
[A1, A3] = threePtSearchModified(y, interval, epsilon);

% Compute the midpoint of A1
A2 = (A1(1) + A1(2)) / 2;

% Display the results
disp(['A1: ', num2str(A1)]);
disp(['A2: ', num2str(A2)]);
disp(['A3: ', num2str(A3)]);


% Set the initial sample points for the successive parabolic interpolation
t1 = 3;
t2 = 6;
t3 = 9;
initial_samp = [t1, t2, t3];

% Call the modified succParInt function to compute the minimum of y(t)
% using successive parabolic interpolation, with error threshold epsilon =
 1e-3
[A4, A6] = succParIntModified(y, initial_samp, epsilon);

% Compute the midpoint of A4
A5 = (A4(1) + A4(2)) / 2;

% Display the results
disp(['A4: ', num2str(A4)]);
disp(['A5: ', num2str(A5)]);
disp(['A6: ', num2str(A6)]);


% Correctly calculate the derivative of y(t)
dy_dt = @(t) (10/3) * (exp(-t/24) / 24 - exp(-t/2) / 2);

% Set the initial guess, learning rate, and Cauchy error threshold
t0 = 1;
gamma = 20; % Adjust the learning rate to a smaller value
threshold = 1e-3;

% Perform gradient descent on y(t)
A7 = gradDescent(dy_dt, t0, gamma, threshold);
```

```
% Display the result
disp(['A7: ', num2str(A7)]);
% Function threePtSearch is modified to return both the interval and the
 number of iterations
function [int, N] = threePtSearchModified(f, int, epsilon)
    a = int(1);
    b = int(2);

    x = linspace(a, b, 5);
    L = b - a;
    N = 0; % Initialize the number of iterations to zero

    while L > epsilon
        N = N + 1; % Increment the number of iterations
        y = f(x);
        min_y = min(y);
        min_ind_y = find(y == min_y);

        if length(min_ind_y) == 1
            if min_ind_y(1) == 1
                a = x(1);
                b = x(2);
            elseif min_ind_y(1) == 5
                a = x(4);
                b = x(5);
            else
                j = min_ind_y(1);
                a = x(j - 1);
                b = x(j + 1);
            end
        else
            if min(min_ind_y) == 1
                a = x(1);
                b = x(2);
            elseif max(min_ind_y) == 5
                a = x(4);
                b = x(5);
            else
                j = min(min_ind_y);
                a = x(j);
                b = x(j + 1);
            end
        end

        x = linspace(a, b, 5);
        L = b - a;
    end

    int = [a b];
end

% Function succParInt is modified to return both the interval and the number
 of iterations
function [int, N] = succParIntModified(f, samp, epsilon)
```

```
    L = samp(3) - samp(1);
    N = 0; % Initialize the number of iterations to zero

    while L > epsilon
        N = N + 1; % Increment the number of iterations
        f_samp = f(samp);
        pcoeff = polyfit(samp, f_samp, 2);
        x4 = -pcoeff(2) / (2 * pcoeff(1));

        x_new = [samp, x4];
        f_new = [f_samp, f(x4)];

        x_new(min(find(f_new == max(f_new)))) = [];
        samp = sort(x_new);
        L = samp(3) - samp(1);
    end

    int = [samp(1) samp(3)];
end

% Function gradDescent is copied from the lecture notes
function [x1] = gradDescent(df, x0, gamma, threshold)
    x1 = x0 - gamma * df(x0);
    epsilon = max(abs(x1 - x0));

    while epsilon >= threshold
        x0 = x1;
        x1 = x0 - gamma * df(x0);
        epsilon = max(abs(x1 - x0));
    end
end
```

```
A1: 5.4214      5.4221
A2: 5.4218
A3: 14
A4: 5.4215      5.4216
A5: 5.4216
A6: 8
A7: 5.4216
```

Published with MATLAB® R2022b