Assignment 2

Mariam Walaa

March 30, 2024

Question 1(A)

Consider the well-known chaotic model of Henon & Heiles (motivated by the motion of stars in a galaxy).
The model is

$$\frac{dp_1}{dt} = -q_1 - 2q_1q_2, p_1(0) = 0.24$$

$$\frac{dp_2}{dt} = -q_2 - q_1^2 + q_2^2, p_2(0) = 0.24$$

$$\frac{dq_1}{dt} = p_1, q_1(0) = 0.24$$

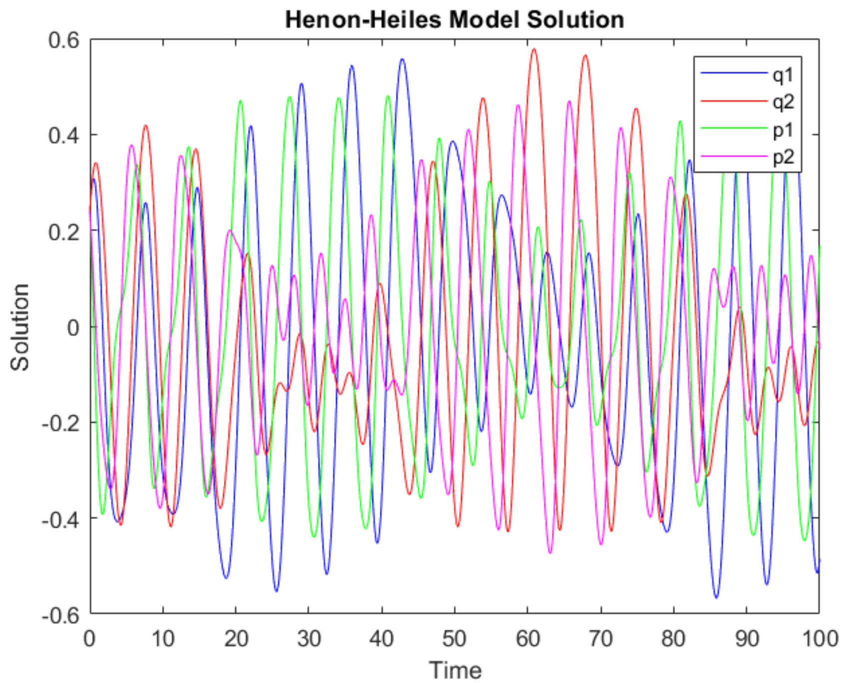
$$\frac{dq_2}{dt} = p_2, q_2(0) = 0.24$$

Integrate on the time interval $[0, 100]$. Plot on the same graph the evolution in time of p_1, p_2, q_1, q_2.

Solution:

```
function A2_Q1()
    % Define the function representing the system of ODEs
    f = @(t, y) HenonHeilesODE(t, y);

    % Set initial conditions
    y0 = [0.24; 0.24; 0.24; 0.24];

    % Set up the tolerances
    options = odeset('RelTol', 100 * eps, 'AbsTol', 1e-14);

    % Time span for integration
    tspan = [0 100]; % Adjust the final time as needed

    % Call ode113 solver
    [t, y] = ode113(f, tspan, y0, options);

    % Plot the results
    figure;
    plot(t, y(:, 1), 'b', t, y(:, 2), 'r', t, y(:, 3), 'g', t, y(:, 4), 'm');
    xlabel('Time');
    ylabel('Solution');
    legend('q1', 'q2', 'p1', 'p2');
    title('Henon-Heiles Model Solution');
end

function dydt = HenonHeilesODE(~, y)
    % Unpack the variables
    q1 = y(1);
    q2 = y(2);
    p1 = y(3);
    p2 = y(4);

    % Define the system of ODEs
    dq1dt = p1;
    dq2dt = p2;
    dp1dt = -q1 - 2 * q1 * q2;
    dp2dt = -q2 - q1^2 + q2^2;
```

```
    % Pack the derivatives into a column vector
    dydt = [dq1dt; dq2dt; dp1dt; dp2dt];
end
```



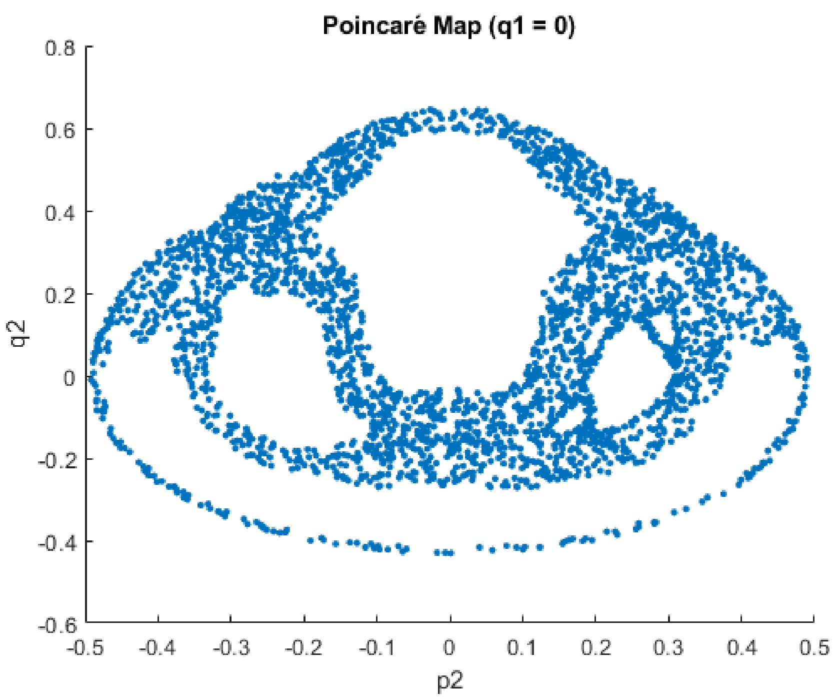
Question 1(B)

Integrate on the time interval $[0, 2 \times 10^4]$. Create the Poincare map of the system by taking $q_1(t) = 0$ and plotting $q_2(t)$ against $p_2(t)$.

Solution:

```
function A2_Q1_b()
    % Define the function representing the system of ODEs
    f = @(t, y) HenonHeilesODE(t, y);

    % Set initial conditions
    y0 = [0.24; 0.24; 0.24; 0.24];

    % Set up the tolerances
    options = odeset('RelTol', 100 * eps, 'AbsTol', 1e-14);

    % Time span for integration
    tspan = [0 2e4];

    % Call ode113 solver
    [~, y] = ode113(f, tspan, y0, options);

    % Find indices where q1 crosses zero and is increasing
    zero_crossings = find(diff(sign(y(:, 1))) > 0);

    % Extract corresponding q2 and p2 values
    q2_at_zero_crossings = y(zero_crossings, 2);
```

```
    p2_at_zero_crossings = y(zero_crossings, 4);

    % Plot the Poincaré map
    figure;
    scatter(p2_at_zero_crossings, q2_at_zero_crossings, 10, 'filled');
    xlabel('p2');
    ylabel('q2');
    title('Poincaré Map (q1 = 0)');
end

function dydt = HenonHeilesODE(~, y)
    % Unpack the variables
    q1 = y(1);
    q2 = y(2);
    p1 = y(3);
    p2 = y(4);

    % Define the system of ODEs
    dq1dt = p1;
    dq2dt = p2;
    dp1dt = -q1 - 2 * q1 * q2;
    dp2dt = -q2 - q1^2 + q2^2;

    % Pack the derivatives into a column vector
    dydt = [dq1dt; dq2dt; dp1dt; dp2dt];
end
```



Question 2

Consider the initial value problem

$$\frac{dy}{dt} = 4t(\sqrt{y^2+1})/y, y(0) = 1$$

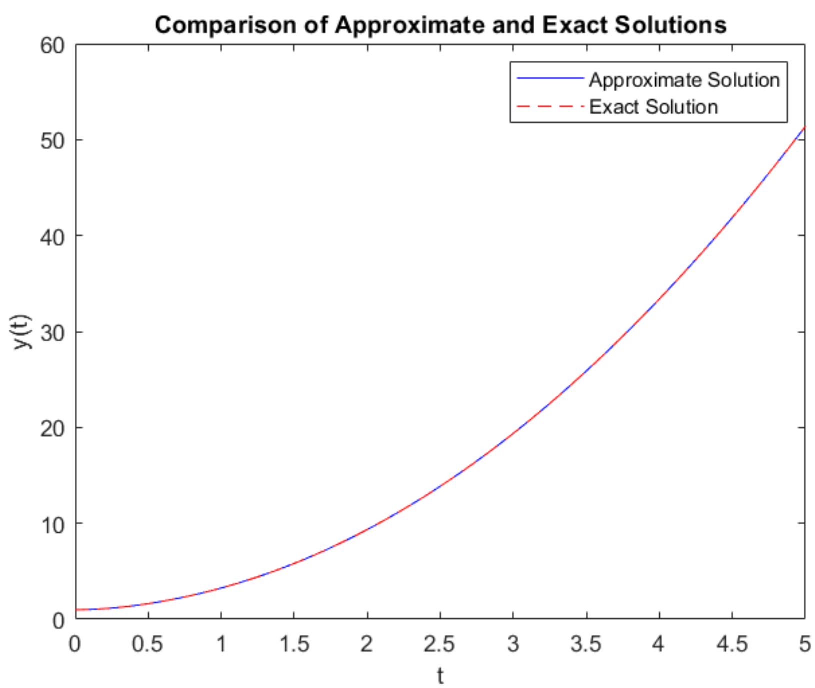
The exact solution of this problem is $y(t) = \sqrt{(2t^2+\sqrt{2})^2 - 1}$.

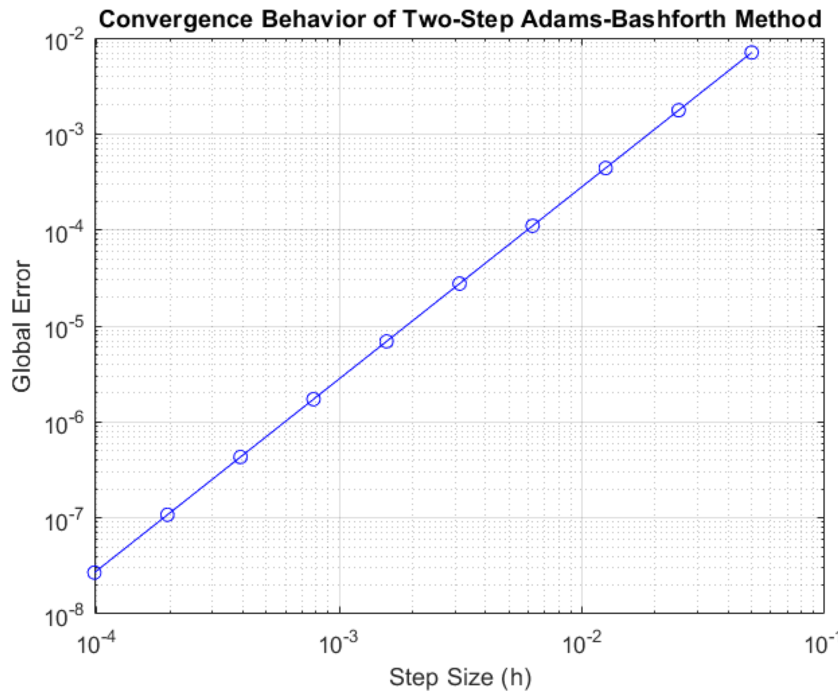
- Write a Matlab program using 2-step Adams-Bashforth method to approximate the solution of this problem.
- Numerically demonstrate that the global error of the 2-step Adams-Bashforth method is $O(h^2)$.

Solution:

```
% Define the function dy/dt
f = @(t, y) 4 * t * sqrt(y^2 + 1) / y;

% Define the exact solution
exact_solution = @(t) sqrt((2*t.^2 + sqrt(2)).^2 - 1);

% Initial values
t0 = 0;
y0 = 1;
T = 5;

% Number of step sizes to test
num_stepsizes = 10;
h_values = zeros(1, num_stepsizes);
global_errors = zeros(1, num_stepsizes);

for i = 1:num_stepsizes
    % Step size
    N = 100 * 2^(i-1); % number of steps, increasing powers of 2
    h = (T - t0) / N;

    % Initialize arrays to store results
    t_values = zeros(1, N+1);
    y_values = zeros(1, N+1);

    % Initial values
    t_values(1) = t0;
    y_values(1) = y0;

    % Two-step Adams-Bashforth method
    for j = 1:N
        if j == 1
            % Use the forward Euler method for the first step
            y_values(2) = y_values(1) + h * f(t_values(1), y_values(1));
        else
            % Predictor step
            y_values(j+1) = y_values(j) + (h / 2) * (3 * f(t_values(j), y_values(j)) - f(t_values(j-1), y_v

            % Corrector step
            %y_values(j+1) = y_values(j) + (h / 2) * (f(t_values(j+1), y_pred) + f(t_values(j), y_values(j)
        end

        % Update time
```

```
        t_values(j+1) = t_values(j) + h;
    end

    % Compute the exact solution values for comparison
    exact_values = exact_solution(t_values);

    % Calculate the global error
    error = abs(exact_values - y_values);
    global_error = max(error);

    % Store the results
    h_values(i) = h;
    global_errors(i) = global_error;
end

% Plot the convergence behavior
figure;
loglog(h_values, global_errors, 'bo-');
xlabel('Step Size (h)');
ylabel('Global Error');
title('Convergence Behavior of Two-Step Adams-Bashforth Method');
grid on;

% Perform linear regression on log-log plot
coefficients = polyfit(log(h_values), log(global_errors), 1);

% Extract slope from the coefficients
slope = coefficients(1);

% Display the slope (convergence rate)
fprintf('Slope (Convergence Rate): %.2f\n', slope);

Slope (Convergence Rate): 2.00
```




Convergence Behavior of Two-Step Adams-Bashforth Method
Global Error
Step Size (h)
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-4}
10^{-3}
10^{-2}
10^{-1}

Question 3

Write a Matlab code to approximate numerically the solution of the following boundary value problem using the finite difference method

$$e^x \frac{d}{dx}(e^x \frac{dy}{dx}) = -1, y(0) = 0, y(1) = 0$$

The exact solution is

$$y(x) = \frac{1}{2}(-e^{-2x} + (1 + e^{-1})e^{-x} - e^{-1})$$

- On the same graph, plot both the approximate and exact solution as functions of x.
- Provide also a graph of the absolute error vs x.

Solution:

We have

$$e^x \frac{d}{dx}(e^x \frac{dy}{dx}) = -1$$

$$e^x (\frac{d}{dx} e^x \frac{dy}{dx} + e^x \frac{d^2y}{(dx)^2}) = -1$$

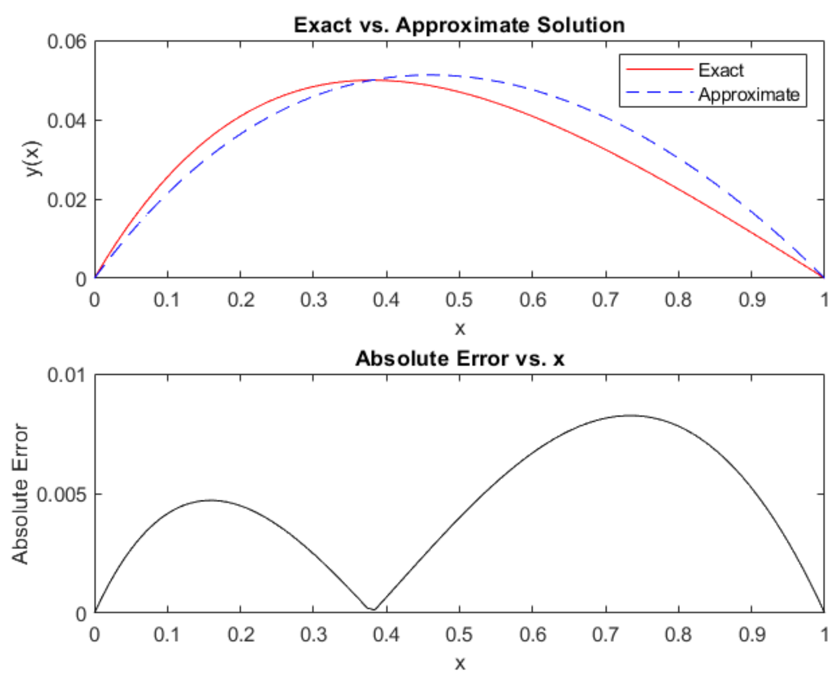
$$e^{2x} \frac{dy}{dx} + e^{2x} \frac{d^2y}{(dx)^2} = -1$$

$$\frac{d^2y}{(dx)^2} = -e^{-2x} - \frac{dy}{dx}$$

This gives $r(x) = -e^{-2x}$, $q(x) = -1$, $p(x) = 0$.

```
% Parameters
N = 100; % Number of grid points
a = 0;
b = 1;   % Length of the domain
h = (b-a) / (N - 1); % Grid spacing
x = linspace(0, (b-a), N); % Grid points

% Define coefficient function and right-hand side function
f = @(x) -1 * h.^2 * -1 * exp(-2 * x);
alpha = 0;
beta = 0;

% Construct coefficient matrix and right-hand side vector
A = zeros(N, N);
b = zeros(N, 1);
A(1, 1) = 1;
A(N, N) = 1;
b(1) = 0;
b(N) = 0;
for i = 2:N-1
    % l_i
    A(i, i-1) = -1 - h/2;
    % d_i
    A(i, i) = 2;
    % u_i
    A(i, i+1) = -1 + h/2;
```

```
    b(i) = f(x(i));
end

% Solve the linear system
y_approx = A \ b;

% Exact solution
y_exact = 0.5 * (-exp(-2*x) + (1 + exp(-1)) * exp(-x) - exp(-1));

% Calculate absolute error
error = abs(y_exact.' - y_approx);

% Plot results
figure;
subplot(2,1,1);
plot(x, y_exact, 'r-', x, y_approx, 'b--');
xlabel('x');
ylabel('y(x)');
title('Exact vs. Approximate Solution');
legend('Exact', 'Approximate');

subplot(2,1,2);
plot(x, error, 'k-');
xlabel('x');
ylabel('Absolute Error');
title('Absolute Error vs. x');
```



Question 4

Determine the order of convergence of the following linear multistep method

$$w_{i+1} - \frac{4}{3}w_i + \frac{1}{3}w_{i-1} = \frac{2}{3}hf(t_{i+1}, w_{i+1})$$

Solution:

We have $m = 2, a_1 = \frac{4}{3}, a_2 = \frac{1}{3}, b_0 = \frac{2}{3}$.

For $k = 1$, we have

$$2^1 = m = \sum_{j=1}^{2} a_j(2-j) + \sum_{j=0}^{2} b_j(2-j)^0 = a_1 + b_0 = \frac{4}{3} + \frac{2}{3} = \frac{6}{3} = 2$$

For $k = 2$, we have

$$2^2 = m^2 = \sum_{j=1}^{2} a_j(2-j)^2 + \sum_{j=0}^{2} b_j(2-j)^1 = a_1 + 4b_0 = \frac{4}{3} + 4\frac{2}{3} = \frac{12}{3} = 4$$

For $k = 3$, we have

$$2^3 = m^3 = \sum_{j=1}^{2} a_j(2-j)^3 + \sum_{j=0}^{2} b_j(2-j)^2 = a_1 + 12b_0 = \frac{4}{3} + 12\frac{2}{3} = \frac{12}{3} = 4 \neq 8$$

Therefore, the order of convergence is $O(h^2)$.

Question 5

Derive the coefficient b_5 of $f(t_{i-4}, w_{i-4})$ in the difference equation for the 5-step Adams-Bashforth method

$$\frac{w_{i+1} - w_i}{h} = b_1 f(t_i, w_i) + b_2 f(t_{i-1}, w_{i-1}) + b_3 f(t_{i-2}, w_{i-2}) + b_4 f(t_{i-3}, w_{i-3}) + b_5 f(t_{i-4}, w_{i-4})$$

Solution:

To derive the coefficient b_5 in the 5-step Adams-Bashforth method, we'll use the Lagrange interpolation method.

We are trying to interpolate (t_{n+1}, f_{n+1}) using (t_n, f_n), (t_{n-1}, f_{n-1}), (t_{n-2}, f_{n-2}), (t_{n-3}, f_{n-3}), and (t_{n-4}, f_{n-4}).

The 5-step Adams-Bashforth method uses the formula:

$$w_{i+1} = w_i + h \sum_{j=0}^{5} b_j f(t_{i-j}, w_{i-j})$$

Let $m = 5$. We can identify b_5 as the coefficient of $f(t_{i-4}, w_{i-4})$. Now, let's construct a Lagrange interpolating polynomial to find b_5. The Lagrange interpolating polynomial is given by:

$$P_m(t) = \sum_{j=0}^{m} L_j(t) \cdot f_j = f_1 \cdot L_1(t) + f_2 \cdot L_2(t) + f_3 \cdot L_3(t) + f_4 \cdot L_4(t) + f_5 \cdot L_5(t)$$

where $L_j(t)$ are the Lagrange basis polynomials:

$$L_j(t) = \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{t - t_m}{t_j - t_m}$$

Let $j = 2 - 4$. We'll use $L_{n-4}(t)$ to represent the interpolating polynomial of degree 4 since we have 5 points.

Now, we need to find the Lagrange basis polynomial $L_{n-4}(t)$.

$$L_{n-4}(t) = \frac{(t - t_n)(t - t_{n-1})(t - t_{n-2})(t - t_{n-3})}{(t_{n-4} - t_n)(t_{n-4} - t_{n-1})(t_{n-4} - t_{n-2})(t_{n-4} - t_{n-3})}$$

So

$$b_5 = \int_{t_n}^{t_{n+1}} \frac{(t - t_n)(t - t_{n-1})(t - t_{n-2})(t - t_{n-3})}{(t_{n-4} - t_n)(t_{n-4} - t_{n-1})(t_{n-4} - t_{n-2})(t_{n-4} - t_{n-3})}$$

Simplifying:

$$\begin{aligned}
b_5 &= \int_{t_n}^{t_{n+1}} \frac{(h)(-h)(-h)(-h)u(u-1)(u+1)(u+2)}{(-4h)(-3h)(-2h)(-h)} \\
&= \int_{t_n}^{t_{n+1}} \frac{h^4 \cdot u(u-1)(u+1)(u+2)}{24 \cdot h^4} \\
&= \int_1^2 \frac{u(u-1)(u+1)(u+2)}{24} \\
&= \int_1^2 \frac{u^4 + 2u^3 - u^2 - 2u}{24} \\
&= \frac{\frac{u^5}{5} + 2\frac{u^4}{4} - \frac{u^3}{3} - u^2}{24} \\
&= \frac{\frac{251}{30}}{24} \\
&= \frac{251}{720}
\end{aligned}$$

So, the coefficient b_5 of $f(t_{i-4}, w_{i-4})$ in the difference equation for the 5-step Adams-Bashforth method using Lagrange interpolation is $\frac{251}{720}$.