Student Name: Mina Gerges
ID: 70040441
Computational Physics_ PHYS624
Project 1: Euler Methods {Harmonic Oscillator}

Numerical and Analytical Solutions to the harmonic Oscillator

1. Introduction and Problem Description

This report investigates a physical system where the dynamics of the simple harmonic motion are modeled using both numerical methods (Simple Euler and Modified Euler methods) and an analytical solution. The goal is to compare these numerical approximations with the exact solution to evaluate their accuracy over time.

2. Physical Problem and Initial Conditions

Consider a particle of mass (m) subject to a force field. We aim to model its displacement x(t) over time using Newton's second law {second order differential equation}:

$$m\frac{d^2x}{dt^2} = F(x,t) \quad \text{----------(1)}$$

where $F(x,t)$ represents the net force acting on the particle, which may depend on the position (x) and time (t).

The initial conditions for this problem are as follows:

I. Initial Position (x_0) : [1]
II. Initial Velocity (v_0) : [0]
III. Modified_position(x_0): [1]
IV. Modified _Velocity (v_0) : [0]
V. Time Step (Δt) : **0.1 or 0.01**

3. Numerical Methods

3.1 Simple Euler Method

The Euler method is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. It approximates the next value of x(t) based on the current slope (*Numerical recipes in FORTRAN*, 1992),as it allows us to move to the solution one step at a time (DeVries, 1994). However, this method has a low accuracy as the error is of O(h^2) and is propagated where the error becomes after some steps of O(h). (Pang, 2006)

For our problem:

$$x_{n+1} = x_n + v_n\Delta t \text{ ----------(2)}$$

$$v_{n+1} = v_n + \frac{F(x_n,t_n)}{m}\Delta t \quad \text{---------(3)}$$

Note: This method is straightforward but tends to accumulate significant error over time due to its limited accuracy and stability.

5.2 Modified Euler Method

The Modified Euler method, is a second-order technique that improves upon the basic Euler approach by taking an average of the initial and corrected slopes(Pang, 2006):

$$x_{n+1} = x_n + \frac{\Delta t}{2}(v_n + v_{n+1}) \text{---------- (4)}$$

$$v_{n+1} = v_n + \frac{\Delta t}{2}\left(\mathrm{F}(x_n,t_n) + F(x_{n+1},t_{n+1})\right) \text{----------(5)}$$

This method offers better accuracy than the simple Euler approach, as it reduces truncation error and provides more stability in the solution by calclulating the corrected values of position and velocity from the

Student Name: Mina Gerges
ID: 70040441
Computational Physics_ PHYS624
Project 1: Euler Methods {Harmonic Oscillator}

---

4. Analytical Solution

For comparison, we use the exact analytical solution to the problem, which is derived based on the known force field and initial conditions. The analytical solution provides a benchmark against which the accuracy of the numerical methods can be measured.

The harmonic oscillator is a classical system where a particle or object experiences a restoring force that is proportional to its displacement from an equilibrium position(Dourmashkin, 2020). This is often described by Hooke's Law, where the force acting on the particle is given by:

$$\boldsymbol{F = -kx}$$

where:
(F) is the restoring force
(k) is the spring constant (or force constant),
(x) is the displacement from the equilibrium position.

For a mass (m) on a spring, Newton's second law tells us that:

$$m\frac{d^2x}{dt^2} = -kx$$

This equation can be rewritten as:

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0$$

This is a second-order homogeneous differential equation with constant coefficients and is known as the harmonic oscillator equation.

We are looking for a position function x(t) such that the second time derivative position function is proportional to the negative of the position function

$$x(t) = A\cos(\omega_0 t) + B\sin(\omega_0 t)$$

$$\frac{dx}{dt} = -\omega_0 A\sin(\omega_0 t)$$

$$\frac{d^2x}{dt^2} = -\omega_0^2 A\cos(\omega_0 t) = -\omega_0^2 x$$

$$-\omega_0^2 A\,cos(\omega_0 t) = -\frac{k}{m}A\,cos(\omega_0 t)$$

$$\omega 0 = \sqrt{\frac{k}{m}}$$

The period of oscillation is then

$$T = \frac{2\pi}{\omega 0}$$

$$T = 2\pi\sqrt{\frac{m}{k}}$$

One possible solution for the position is

$$\boldsymbol{x(t) = A\cos(\omega_0 t)}$$ **{this is used to calculate the analytic porition}**

Student Name: Mina Gerges
ID: 70040441
Computational Physics_ PHYS624
Project 1: Euler Methods {Harmonic Oscillator}

Energy of the Harmonic Oscillator

The total mechanical energy (E) of the harmonic oscillator is constant and given by the sum of kinetic and potential energies:

1. Kinetic Energy: $T = \frac{1}{2}mv^2 = \frac{1}{2}m\left(\frac{dx}{dt}\right)^2$
2. Potential Energy: $U = \frac{1}{2}kx^2$

$$E = T + U$$

$E = \frac{1}{2}kx^2 + \frac{1}{2}m\left(\frac{dx}{dt}\right)^2$ for our problem we consider $m \,\&\, k = 1$

$$\boldsymbol{E = \frac{1}{2}(x^2 + v^2)}$$

{ used to calculate the total energy via Euler simple values and Modified corrected values

This energy remains constant throughout the motion due to the conservation of energy in an ideal harmonic oscillator.

5. C++ Script

```
#include <iostream>
#include <vector>
#include <cmath>
#include <fstream>

using namespace std;

double euler2d(double(*d1x)(double, double, double),
               double(*d2x)(double, double, double),
               double ti, double xi, double vi, double tf,
               double& xf, double& vf, bool modified) {
    double dt = tf - ti;
    if (modified) {
        // Modified Euler method
        xf = xi + d1x(ti, xi, vi) * dt; //predicted
        vf = vi + d2x(ti, xi, vi) * dt; //predicted
        xf = xi + (d1x(ti, xi, vi) + d1x(ti, xf, vf)) * 0.5 * dt; // corrected
        vf = vi + (d2x(ti, xi, vi) + d2x(ti, xf, vf)) * 0.5 * dt; //corrected
    } else {
        // Simple Euler method
        xf = xi + d1x(ti, xi, vi) * dt;
        vf = vi + d2x(ti, xi, vi) * dt;
    }
    return 0.0;
}

double f1(double t, double x, double v) {
    return v; // dx/dt = v
}

double f2(double t, double x, double v) {
    return -x; // dv/dt = -x (harmonic oscillator)
}
```

Student Name: Mina Gerges
ID: 70040441
Computational Physics_ PHYS624
Project 1: Euler Methods {Harmonic Oscillator}

```
int main() {
    // Initial conditions
    double A = 1.0; // amplitude
    double dt = 0.01; // time step
    double tf = 100.0; // final time
    int n_steps = static_cast<int>(tf / dt);

    // Vectors to store results
    vector<double> time(n_steps), simple_x(n_steps), simple_v(n_steps);
    vector<double> modified_x(n_steps), modified_v(n_steps), analytical_x(n_steps);
    vector<double> simple_energy(n_steps), modified_energy(n_steps);

    // Initial conditions
    simple_x[0] = A;
    simple_v[0] = 0.0;
    modified_x[0] = A;
    modified_v[0] = 0.0;

    // Initial energy calculation
    simple_energy[0] = 0.5 * simple_v[0] * simple_v[0] + 0.5 * simple_x[0] * simple_x[0];
    modified_energy[0] = 0.5 * modified_v[0] * modified_v[0] + 0.5 * modified_x[0] *
modified_x[0];

    // Time integration using Simple Euler
    for (int i = 1; i < n_steps; i++) {
        time[i] = i * dt;
        euler2d(f1, f2, time[i - 1], simple_x[i - 1], simple_v[i - 1], time[i],
simple_x[i], simple_v[i], false);

        // Calculate energy for Simple Euler
        simple_energy[i] = 0.5 * simple_v[i] * simple_v[i] + 0.5 * simple_x[i] *
simple_x[i];
    }

    // Time integration using Modified Euler
    for (int i = 1; i < n_steps; i++) {
        euler2d(f1, f2, time[i - 1], modified_x[i - 1], modified_v[i - 1], time[i],
modified_x[i], modified_v[i], true);

        // Calculate energy for Modified Euler
        modified_energy[i] = 0.5 * modified_v[i] * modified_v[i] + 0.5 * modified_x[i] *
modified_x[i];
    }

    // Analytical solution
    for (int i = 0; i < n_steps; i++) {
        analytical_x[i] = A * cos(time[i]);
    }

    // Write results to files for plotting
    ofstream file("results.txt");
    file <<
"Time\tSimple_Euler_X\tModified_Euler_X\tAnalytical_X\tSimple_Energy\tModified_Energy\n";
```

Student Name: Mina Gerges
ID: 70040441
Computational Physics_ PHYS624
Project 1: Euler Methods {Harmonic Oscillator}

```
    for (int i = 0; i < n_steps; i++) {
        file << time[i] << "\t" << simple_x[i] << "\t" << modified_x[i] << "\t" <<
analytical_x[i] << "\t"
             << simple_energy[i] << "\t" << modified_energy[i] << "\n";
    }
    file.close();

    cout << "Results written to results.txt" << endl;
    return 0;
}
```

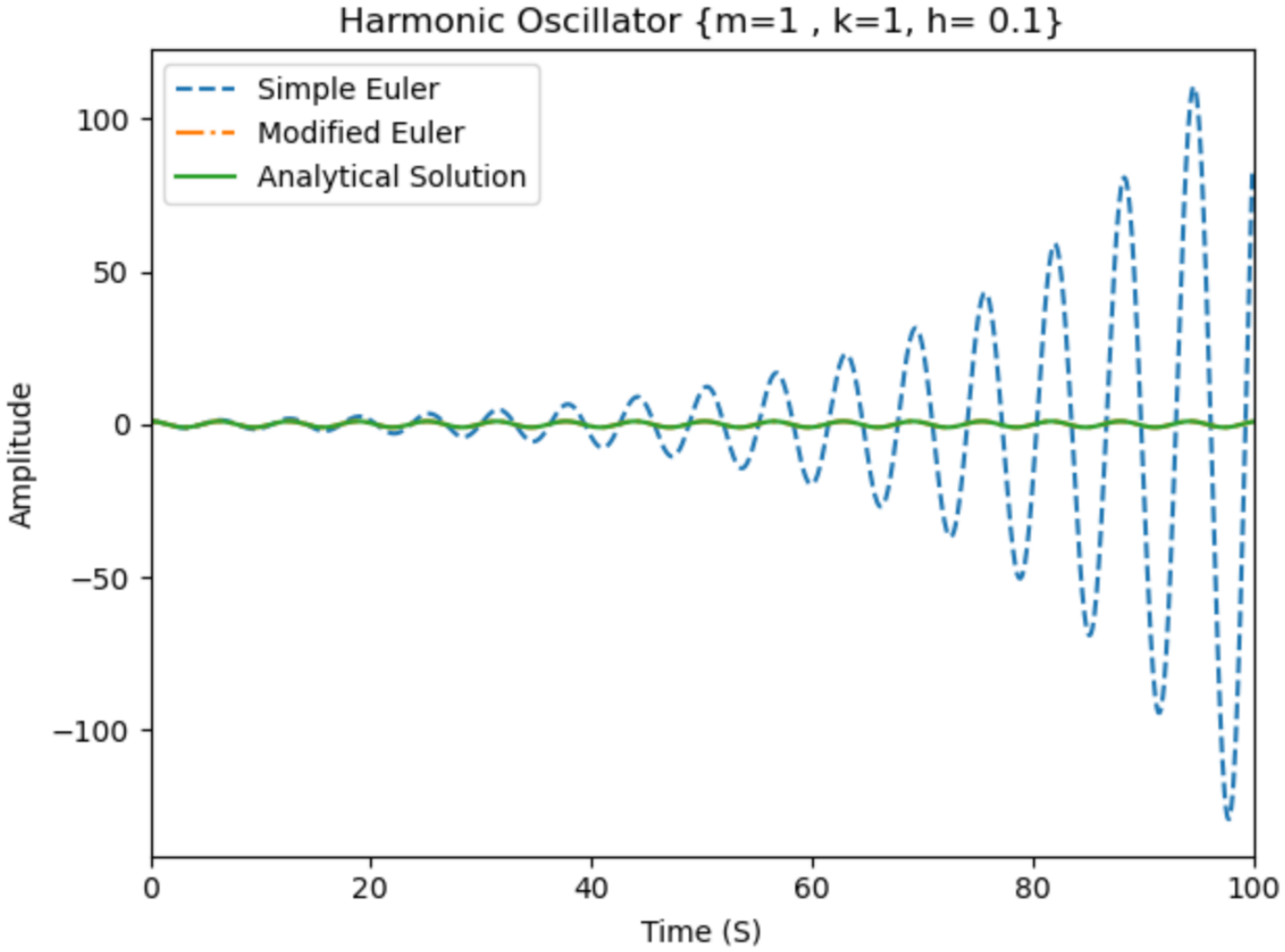
The sscript follow these steps:

1. defining a function for Euler first order and second order of differential equation with bool modified to shift the calculation of modified when required. **bool** modified
2. if – else condition {modified – simple}
3. define a function for the velocity.
4. Define a function for harmonic oscillator.
5. Initializing th emain program with the initial conditions
   Initial Position (x_0) : [1]
   Initial Velocity (v_0) : [0]
   Modified_position(x_0): [1]
   Modified _Velocity (v_0) : [0]
   Time Step (Δt) : **0.1 or 0.01** as time goes form 0 to 100
6. Calculating the energy with the two methods.
7. Calculate the analytic solution of th position(amplitude)
8. Saving results to output file for plotting.

6. Results:
Trial 1:
I ran the script with the initial conditions mentioned above with step size of h=0.1 without calculating the ener. In table 1, part of the calculated values is presented
File{ results(1).txt}

Time	Simple_Euler_X	Modified_Euler_X	Analytical_X
0	1	1	1
0.1	1	0.995	0.995004
0.2	0.99	0.98005	0.980067
0.3	0.97	0.955299	0.955336
0.4	0.9401	0.920996	0.921061
0.5	0.9005	0.877483	0.877583
0.6	0.851499	0.825194	0.825336
0.7	0.793493	0.764654	0.764842
0.8	0.726972	0.696467	0.696707
0.9	0.652516	0.621316	0.62161

Plot1: presents the amplitude as a function of time for the simple harmonic oscillator with m & k = 1 at h= 0.1



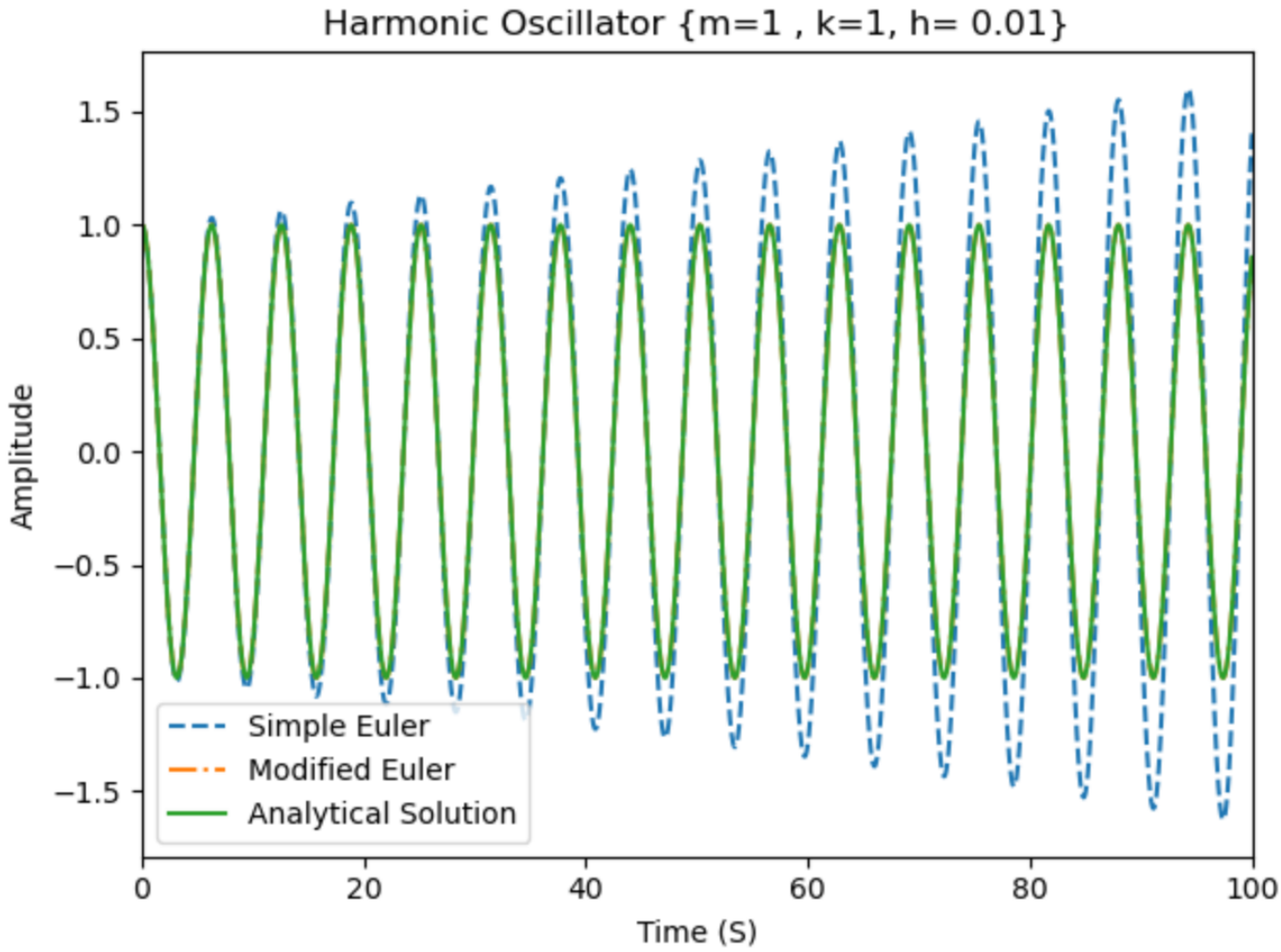
Trial 2:

I ran the script with the initial conditions mentioned above with step size of h=0.01 without calculating the energy. In table 2, part of the calculated values is presented:

File{ results(2).txt}

Time	Simple_Euler_X	Modified_Euler_X	Analytical_X
0	1	1	1
0.01	1	0.99995	0.99995
0.02	0.9999	0.9998	0.9998
0.03	0.9997	0.99955	0.99955
0.04	0.9994	0.9992	0.9992
0.05	0.999	0.99875	0.99875
0.06	0.9985	0.998201	0.998201
0.07	0.9979	0.997551	0.997551

Plot2: Presents the amplitude as a function of time for the simple harmonic oscillator with m & k = 1 at h= 0.01



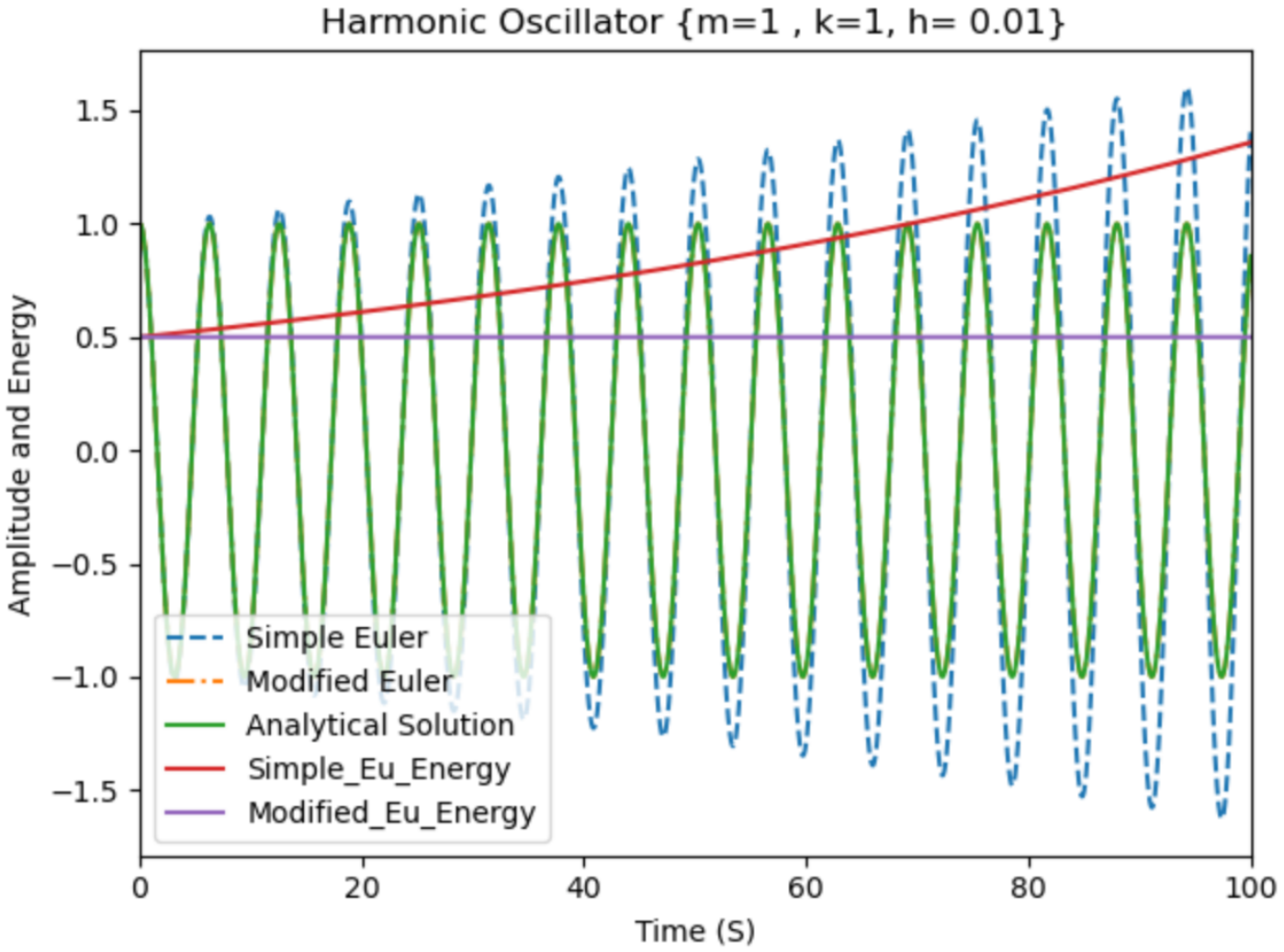
Trial 3:

I ran the script with the initial conditions mentioned above with step size of h=0.01 with calculating the energy by the two methods. In table 3, part of the calculated values is presented:

File{ Amp_Energy1.txt}

Time	Simple_Euler_X	Modified_Euler_X	Analytical_X	Simple_Energy	Modified_Energy
0	1	1	1	0.5	0.5
0.01	1	0.99995	0.99995	0.50005	0.5
0.02	0.9999	0.9998	0.9998	0.5001	0.5
0.03	0.9997	0.99955	0.99955	0.50015	0.5
0.04	0.9994	0.9992	0.9992	0.5002	0.5

Plot3: Presents the amplitude and the energy of the system as a function of time for the harmonic oscillator with m & k = 1 at h= 0.01



7. Discussion

The results of the Simple Euler, Modified Euler, and Analytical solutions are plotted over time to visualize their performance. And here are some key observations from plots and results.

Accuracy: The Modified Euler method demonstrates greater accuracy, particularly over longer time intervals, due to its higher-order nature, while the simple Euler method shows a siginficant rise in the amplitude and energy which is neither physical nor logical since we are not adding energy to the system.

Energy Conservation: Energy plots for the Simple Euler and Modified Euler methods reveal differences in energy conservation. **The Simple Euler method shows drift over time, indicating energy loss or gain due to numerical errors which after some steps of N becomes of O(h).**

Stability: The Modified Euler approach remains more stable across a wider range of time steps compared to the Simple Euler, which is prone to instability in scenarios requiring finer precision.

Step_size: Choosing small step size in simple Euler method reduces the deviation(rise) of calculated position to some extent, however, it stays ineffecient to be implemented due to the accumulated error as discussed previously.

---

8. Conclusion

In this study, we observed that the Modified Euler method provides a significantly more accurate approximation to the analytical solution than the Simple Euler method. While both methods are useful, the choice of method depends on the required accuracy and stability constraints. The Modified Euler method, although computationally more intensive, offers better stability and energy conservation, making it a preferable choice for simulations requiring precision over extended periods.

9. References:

DeVries, P.L., 1994. A first course in computational physics. J. Wiley & sons, New York Chichester Brisbane [etc.].

Dourmashkin, P., 2020. Classical Mechanics,23.2: Simple Harmonic Motion- Analytic [WWW Document]. Physics LibreTexts. URL https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Classical_Mechanics_(Dourmashkin)/23%3A_Simple_Harmonic_Motion/23.02%3A_Simple_Harmonic_Motion-_Analytic (accessed 10.31.24).

Numerical recipes in FORTRAN: the art of scientific computing, 2nd ed. ed, 1992. . Cambridge university press, Cambridge.

Pang, T., 2006. An introduction to computational physics, 2nd ed. ed. Cambridge University Press, Cambridge ; New York.