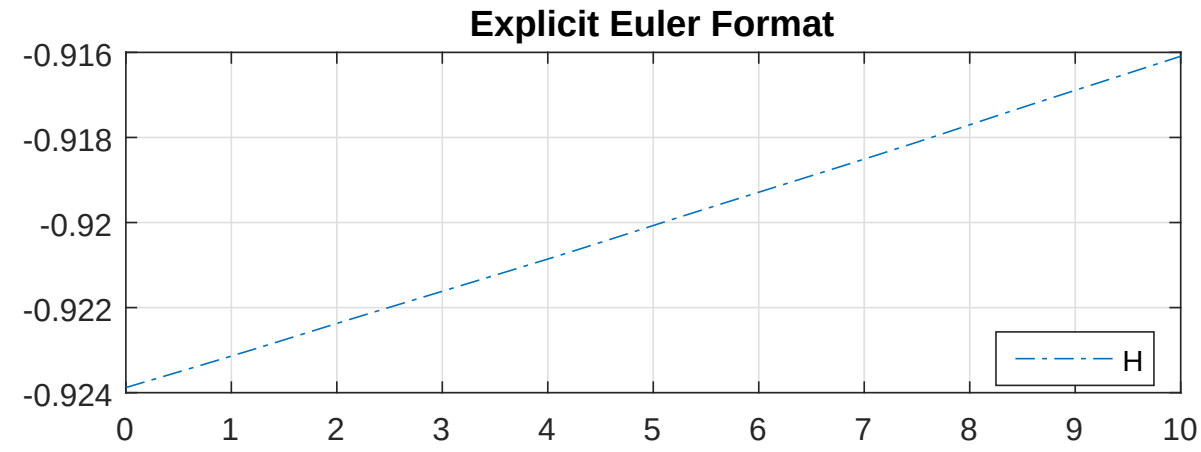
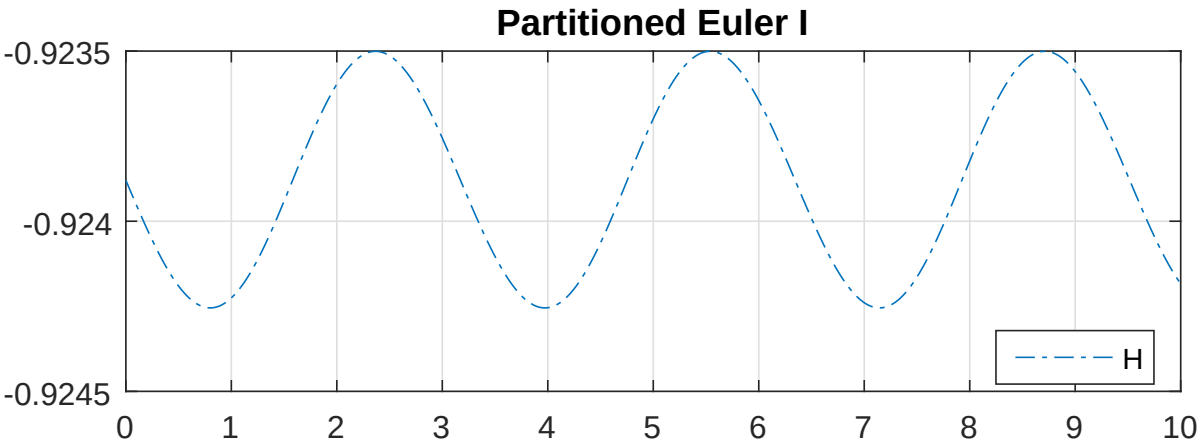
```
function [q, p, t] = Explicit_Euler(q0, p0, t0, h, N)
% Explicit Euler format for the Hamiltonian system;
% (q0, p0, t0): initial conditions;
% h: step length;
% N: number of steps;
    q = zeros(N+1, 1); q(1) = q0;
    p = zeros(N+1, 1); p(1) = p0;
    t = t0 + h.*(0:1:N);
    for i = 1:N
        q(i+1) = q(i)+h*p(i);
        p(i+1) = p(i)-h*sin(q(i));
    end
```

```
function [q, p, t] = Partitioned_Euler_I(q0, p0, t0, h, N)
% Partitioned Euler I format for the Hamiltonian system;
% (q0, p0, t0): initial conditions;
% h: step length;
% N: number of steps;
    q = zeros(N+1, 1); q(1) = q0;
    p = zeros(N+1, 1); p(1) = p0;
    t = t0 + h.*(0:1:N);
    for i = 1:N
        p(i+1) = p(i)-h*sin(q(i));
        q(i+1) = q(i)+h*p(i+1);
    end
```

```
% Initial Conditions
format long;
q0 = pi/8; p0 = 0; t0 = 0;
h = 0.01; N = 1000;

% The Hamiltonian Function
H = @(p, q)((p.^2)./2-cos(q));

[q1, p1, t1] = Explicit_Euler(q0, p0, t0, h, N);
[q2, p2, t2] = Partitioned_Euler_I(q0, p0, t0, h, N);

subplot(211);
plot(t1, H(p1, q1), '-.');
title('Explicit Euler Format');
legend('H', 'Location', 'Best');
grid on;

subplot(212);
plot(t2, H(p2, q2), '-.');
title('Partitioned Euler I');
legend('H', 'Location', 'Best');
grid on;
```


Explicit Euler Format
-0.916
-0.918
-0.92
-0.922
-0.924
0
1
2
3
4
5
6
7
8
9
10
H


Partitioned Euler I
-0.9235
-0.924
-0.9245
0
1
2
3
4
5
6
7
8
9
10
H