Assignment 04: On Non-Linear Ordinary Differential Equations

PH1050 Computational Physics

Pranav S Ramanujam
EP23B038
1^{st} year EP
Department of Physics, IIT Madras

Problem Statement

Part 1: Given the potential energy function of a body of unit mass, we needed to :
1) Plot the function of potential energy as a function of x for small range of x.
2)Find the values of the time period of oscillations corresponding to different values of total energies.

Part2: 1)We needed to find the corresponding Force for the given function of potential energy.
2)Then we had to write the equation of motion of the particle under the influence of an additional damping force and a driving force.
3)Then we had to vary the value of damping constant and plot the graphs of x[t] for different values of damping constant.

Aim

Part1:1) To plot the graph of the given potential as a function of x (for small values of x), to get a feel of the function.
2)To find the corresponding time period for various values of total energy of the particle.

Part2:1)To find the force as a funtion of x.
2)To write the equation of motion along with the damping term and the driving force term.
3)To vary the damping coefficient and plot the graphs of x[t] vs t for various values of the coefficient.
4)To use GraphicsColumn to display all the graphs together.

Introduction

Code Organization

Part1: 1)we made use of Plot function to plot potential.
2)Then wrote the differential equation corresponding to conservation of energy.
3)found dt = f[x]dx
4)Then as we varied the value of energy, we set suitable limits to integrate rhs to get the value of timeperiod.
5)Then plotted the graph of time period vs energy.

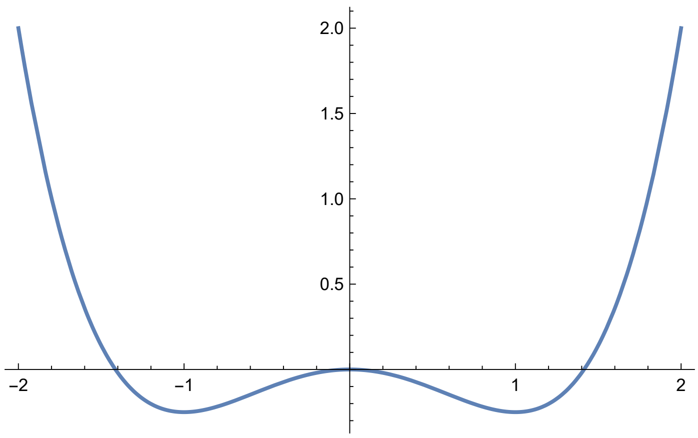
Part2:1)found the corresponding force using differentiation operator.
2)wrote the equation of motion including the damping term and the driving force.
3)Solved it for various values of the damping coefficient to get x[t] as a fucntion of t
4)Plotted all those x[t] vs t.
5) used GraphicsColumn to display all the plots together.

Code for computation

Part-1

In[•]:=
```
vP[x_] = x^4 / 4 - x^2 / 2;  (*Plotting the potential to see its main features*)
Plot[vP[x], {x, -2, 2}]
```

Out[•]=



In[•]:=
```
(*Since this is a conservative system we can say that
  Total Energy=Kinetic Energy + Potential Energy*)
totEnergy[x_] := (1 / 2) * (1) * (dx / dt)^2 + vP[x] == e
totEnergy[x];
dTsol = Solve[totEnergy[x], dt]
dTsol1 = dTsol /. {{{x0_}, {y0_}} → {x0, y0}};
```

Out[•]=

$$\left\{\left\{dt \to -\frac{\sqrt{2}\ dx}{\sqrt{4\ e + 2\ x^2 - x^4}}\right\}, \left\{dt \to \frac{\sqrt{2}\ dx}{\sqrt{4\ e + 2\ x^2 - x^4}}\right\}\right\}$$

In[•]:=
```
dTime = dTsol1[[2]]
```

Out[•]=

$$dt \to \frac{\sqrt{2}\ dx}{\sqrt{4\ e + 2\ x^2 - x^4}}$$

In[ ]:=
```
(*  (1) Considering x=1.21 as one of the turning points*)
dTime = dTsol1[[2]] /. e → vP[1.21]
```

Out[ ]=

$$dt \to \frac{\sqrt{2}\,dx}{\sqrt{-0.784611 + 2\,x^2 - x^4}}$$

In[ ]:=
```
dt /. dTime
```

Out[ ]=

$$\frac{\sqrt{2}\,dx}{\sqrt{-0.784611 + 2\,x^2 - x^4}}$$

In[ ]:=
```
f[x_] := vP[x] == vP[1.21]
soln = NSolve[f[x], x]  (*To get the value of the other extreme point*)
soln
```

Out[ ]=

{{x → -1.21}, {x → -0.732052}, {x → 0.732052}, {x → 1.21}}

Out[ ]=

{{x → -1.21}, {x → -0.732052}, {x → 0.732052}, {x → 1.21}}

In[ ]:=
```
(*here obviously we are supposed to only consider the positive values of
 x as total energy here is negative so the particle cannot cross origin*)
(*We can clearly see that the particle will oscillate between +0.732052 and +1.21*)
solnFinal1 := soln[[4]]
solnFinal1            (*Upper limit of our integral*)
```

Out[ ]=

{x → 1.21}

In[ ]:=
```
solnFinal2 := soln[[3]] (*Lower limit of our Integral*)
solnFinal2
```

Out[ ]=

{x → 0.732052}

In[ ]:=
```
Xextreme1 :=
 x /. solnFinal1 (*Getting the value for one of the extreme point of oscillations*)
```

In[ ]:=
```
Xextreme1
```

Out[ ]=

1.21

In[ ]:=
```
Xextreme2 :=
 x /. solnFinal2 (*Getting the value for one of the extreme point of oscillations*)
```

In[ ]:=
```
Xextreme2
```

Out[ ]=

0.732052

In[ ]:=
```
(*For time period*)
```

$$2\int_{Xextreme2}^{Xextreme1} \frac{\sqrt{2}}{\sqrt{-0.7846111900000001` + 2\,x^2 - x^4}}\,dx$$

In[ ]:=
```
{-0.196, 4.6472}; (*{energy,timeprd}*)
```

In[∘]:= (*This is the value of time period of oscillations for the particle moving between -2.0 and +2.0*)

(*Checking time period for 5 other extreme values of x*)
(* (2) Considering x=1.40 as an extreme point*)

```
dTime = dTsol1[[2]] /. e → vP[1.40]
```

Out[∘]=

$$dt \to \frac{\sqrt{2}\ dx}{\sqrt{-0.0784 + 2\,x^2 - x^4}}$$

In[∘]:= `dt /. dTime`

Out[∘]=

$$\frac{\sqrt{2}\ dx}{\sqrt{-0.0784 + 2\,x^2 - x^4}}$$

In[∘]:=
```
f[x_] := vP[x] == vP[1.40]
soln = NSolve[f[x], x]  (*To get the value of the other extreme point*)
soln
```

Out[∘]=

{{x → -1.4}, {x → -0.2}, {x → 0.2}, {x → 1.4}}

Out[∘]=

{{x → -1.4}, {x → -0.2}, {x → 0.2}, {x → 1.4}}

In[∘]:= (*here obviously we are supposed to only consider the positive values of x as total energy here is negative so the particle cannot cross origin*)

In[∘]:=
```
solnFinal1 := soln[[4]]
solnFinal1            (*Upper limit of our integral*)
```

Out[∘]=

{x → 1.4}

In[∘]:=
```
solnFinal2 := soln[[3]] (*Lower limit of our Integral*)
solnFinal2
```

Out[∘]=

{x → 0.2}

In[∘]:=
```
Xextreme1 := x /. solnFinal1
(*Getting the value for one of the extreme point of oscillations*)
Xextreme1
```

Out[∘]=

1.4

In[∘]:=
```
Xextreme2 :=
 x /. solnFinal2 (*Getting the value for one of the extreme point of oscillations*)
```

In[∘]:= `Xextreme2`

Out[∘]=

0.2

In[∘]:=

$$2 * \int_{\text{Xextreme2}}^{\text{Xextreme1}} \frac{\sqrt{2}}{\sqrt{-0.07840000000000069` + 2\,x^2 - x^4}}\, dx$$

In[ ]:=
```
0.` - 9.48763128310522` ⅈ
(*we got the iota part due to inadequate precision
 we can consider the time period to be just the real part of this*)
{-0.0196, 5.003490235781636} (*energy,time period*)
```

Out[ ]=
```
0. - 9.48763 ⅈ
```

Out[ ]=
```
{-0.0196, 5.00349}
```

In[ ]:=
```
(*(3) Considering x=1.5 as an extreme point*)
dTime = dTsol1[[2]] /. e → vP[1.50]
```

Out[ ]=

$$dt \to \frac{\sqrt{2}\ dx}{\sqrt{0.5625 + 2\,x^2 - x^4}}$$

In[ ]:=
```
dt /. dTime
```

Out[ ]=

$$\frac{\sqrt{2}\ dx}{\sqrt{0.5625 + 2\,x^2 - x^4}}$$

In[ ]:=
```
f[x_] := vP[x] == vP[1.50]
soln = NSolve[f[x], x]  (*To get the value of the other extreme point*)
soln
```

Out[ ]=
```
{{x → -1.5}, {x → 0. - 0.5 ⅈ}, {x → 0. + 0.5 ⅈ}, {x → 1.5}}
```

Out[ ]=
```
{{x → -1.5}, {x → 0. - 0.5 ⅈ}, {x → 0. + 0.5 ⅈ}, {x → 1.5}}
```

In[ ]:=
```
solnFinal1 := soln[[1]]
solnFinal1
```

Out[ ]=
```
{x → -1.5}
```

In[ ]:=
```
solnFinal2 := soln[[4]] (*Lower limit of our Integral*)
solnFinal2
```

Out[ ]=
```
{x → 1.5}
```

In[ ]:=
```
Xextreme1 := x /. solnFinal1
(*Getting the value for one of the extreme point of oscillations*)
Xextreme1
```

Out[ ]=
```
-1.5
```

In[ ]:=
```
Xextreme2 :=
 x /. solnFinal2 (*Getting the value for one of the extreme point of oscillations*)
```

In[ ]:=
```
Xextreme2
```

Out[ ]=
```
1.5
```

In[∘]:= $2 * \int_{\text{Xextreme1}}^{\text{Xextreme2}} \frac{\sqrt{2}}{\sqrt{0.5625` + 2\,x^2 - x^4}}\, dx$

Out[∘]=

9.22366

In[∘]:=
```
{0.1406, 9.22366748324173`}  (*Energy,Timeperiod*)
```

Out[∘]=
```
{0.1406, 9.22367}
```

In[∘]:=
```
(*Considering x=1.70 as an extreme point*)
dTime = dTsol1[[2]] /. e → vP[1.70]
```

Out[∘]=

$dt \to \frac{\sqrt{2}\ dx}{\sqrt{2.5721 + 2\,x^2 - x^4}}$

In[∘]:=
```
dt /. dTime
```

Out[∘]=

$\frac{\sqrt{2}\ dx}{\sqrt{2.5721 + 2\,x^2 - x^4}}$

In[∘]:=
```
f[x_] := vP[x] == vP[1.70]
soln = NSolve[f[x], x]  (*To get the value of the other extreme point*)
soln
```

Out[∘]=
```
{{x → -1.7}, {x → 0. - 0.943398 i}, {x → 0. + 0.943398 i}, {x → 1.7}}
```

Out[∘]=
```
{{x → -1.7}, {x → 0. - 0.943398 i}, {x → 0. + 0.943398 i}, {x → 1.7}}
```

In[∘]:=
```
solnFinal1 := soln[[4]]
solnFinal1
```

Out[∘]=
```
{x → 1.7}
```

In[∘]:=
```
solnFinal2 := soln[[1]] (*Lower limit of our Integral*)
solnFinal2
```

Out[∘]=
```
{x → -1.7}
```

In[∘]:=
```
Xextreme1 := x /. solnFinal1
(*Getting the value for one of the extreme point of oscillations*)
Xextreme1
```

Out[∘]=
```
1.7
```

In[∘]:=
```
Xextreme2 :=
 x /. solnFinal2 (*Getting the value for one of the extreme point of oscillations*)
```

In[∘]:=
```
Xextreme2
```

Out[∘]=
```
-1.7
```

In[ ]:=
```
2 * ∫_{Xextreme2}^{Xextreme1} √2 / √(2.572099999999999` + 2 x^2 - x^4) dx
```
$$2 * \int_{\text{Xextreme2}}^{\text{Xextreme1}} \frac{\sqrt{2}}{\sqrt{2.572099999999999` + 2\,x^2 - x^4}}\, dx$$

Out[ ]=
```
6.35285
```

In[ ]:=
```
{0.643, 6.352848044787067`}; (*energy,timeperiod*)
```

In[ ]:=
```
(*Now taking one extreme as x=2.0*)
dTime = dTsol1[[2]] /. e → vP[2.0]
```

Out[ ]=

$$dt \to \frac{\sqrt{2}\ dx}{\sqrt{8. + 2\,x^2 - x^4}}$$

In[ ]:=
```
dt /. dTime
```

Out[ ]=

$$\frac{\sqrt{2}\ dx}{\sqrt{8. + 2\,x^2 - x^4}}$$

In[ ]:=
```
f[x_] := vP[x] == vP[2.0]
soln = NSolve[f[x], x]  (*To get the value of the other extreme point*)
soln
```

Out[ ]=
```
{{x → -2.}, {x → 0. - 1.41421 i}, {x → 0. + 1.41421 i}, {x → 2.}}
```

Out[ ]=
```
{{x → -2.}, {x → 0. - 1.41421 i}, {x → 0. + 1.41421 i}, {x → 2.}}
```

In[ ]:=
```
solnFinal1 := soln[[4]]
solnFinal1
```

Out[ ]=
```
{x → 2.}
```

In[ ]:=
```
solnFinal2 := soln[[1]] (*Lower limit of our Integral*)
solnFinal2
```

Out[ ]=
```
{x → -2.}
```

In[ ]:=
```
Xextreme1 := x /. solnFinal1
(*Getting the value for one of the extreme point of oscillations*)
Xextreme1
```

Out[ ]=
```
2.
```

In[ ]:=
```
Xextreme2 := x /. solnFinal2
(*Getting the value for one of the extreme point of oscillations*)
Xextreme2
```

Out[ ]=
```
-2.
```

In[•]:= $2*\int_{\text{Xextreme2}}^{\text{Xextreme1}} \frac{\sqrt{2}}{\sqrt{8.\grave{} + 2x^2 - x^4}}\, dx$

Out[•]=

4.68465

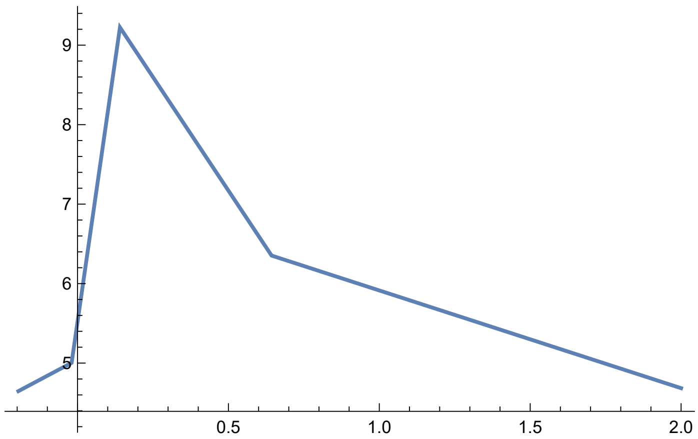
In[•]:=
```
{2, 4.684647540748971`}  (*energy,time period*)
```

Out[•]=

{2, 4.68465}

In[•]:=
```
(*Plotting energy vs time period*)
ListLinePlot[{{-0.196, 4.6472}, {-0.0196, 5.003490235781636},
  {0.1406, 9.22366748324173`}, {0.643, 6.352848044787067`}, {2, 4.684647540748971`}}]
(*Energy on x axis and time period on y axis*)
```

Out[•]=



Part2: Nature of Motion from the plots of the solution

In[•]:=
```
vP[x_] = x^4 / 4 - x^2 / 2
(*(1) Finding the force*)
F[x_] = -D[vP[x], x]
```

Out[•]=

$-\frac{x^2}{2}+\frac{x^4}{4}$

Out[•]=

$x - x^3$

In[•]:=
```
A = 2;
ω = 1.5;
γ = 0.0;
```

In[ ]:=
```
eqn1 := {x''[t] + γ * x'[t] == x[t] - (x[t])^3 + A Sin[2 * ω * t], x[0] == -1.8, x'[0] == 0.0};
soln = NDSolve[eqn1, x[t], {t, 0, 100}]
solnX[t_] = x[t] /. Flatten[soln]
```

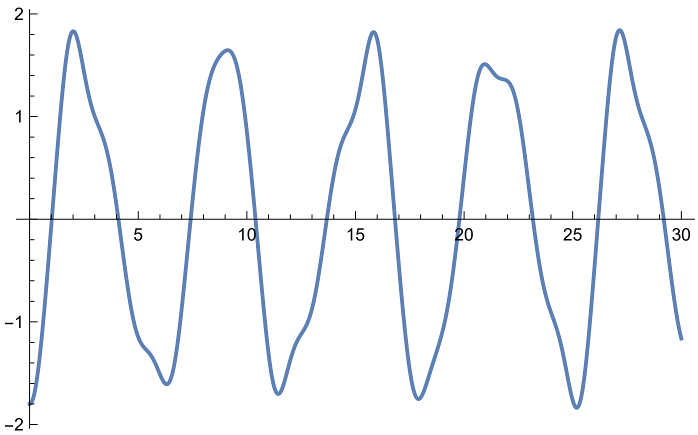
Out[ ]=
```
{{x[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]}}
```

Out[ ]=
```
InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]
```

In[ ]:=
```
data = Table[{t, solnX[t]}, {t, 0, 30, 0.01}];
```

In[ ]:=
```
graphics1 = ListLinePlot[data]
```

Out[ ]=



In[ ]:=
```
(*Changing the values of γ*)
A = 2;
ω = 1.5;
γ = 0.6;
```

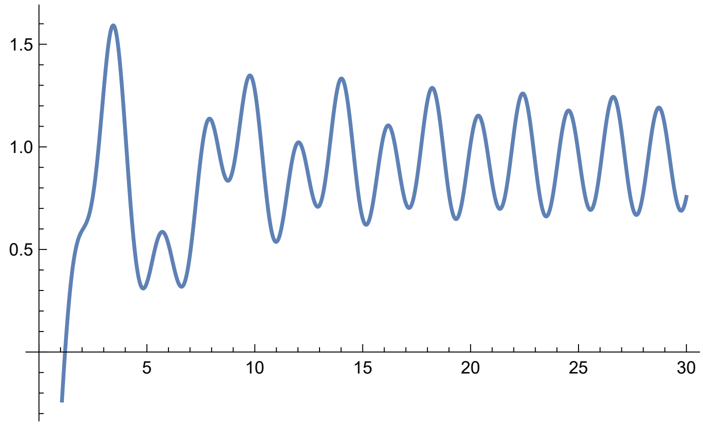
In[ ]:=
```
eqn1 := {x''[t] + γ * x'[t] == x[t] - (x[t])^3 + A Sin[2 * ω * t], x[0] == -1.8, x'[0] == 0.0};
soln = NDSolve[eqn1, x[t], {t, 0, 100}]
solnX[t_] = x[t] /. Flatten[soln]
```

Out[ ]=
```
{{x[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]}}
```

Out[ ]=
```
InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]
```

In[ ]:=
```
data = Table[{t, solnX[t]}, {t, 0, 30, 0.01}];
```

In[*]:=
```
graphics2 = ListLinePlot[data]
```
Out[*]=



In[*]:=
```
A = 2;
ω = 1.5;
γ = 1.2;
```

In[*]:=
```
eqn1 := {x''[t] + γ * x'[t] == x[t] - (x[t])^3 + A Sin[2 * ω * t], x[0] == -1.8, x'[0] == 0.0};
soln = NDSolve[eqn1, x[t], {t, 0, 100}]
solnX[t_] = x[t] /. Flatten[soln]
```
Out[*]=

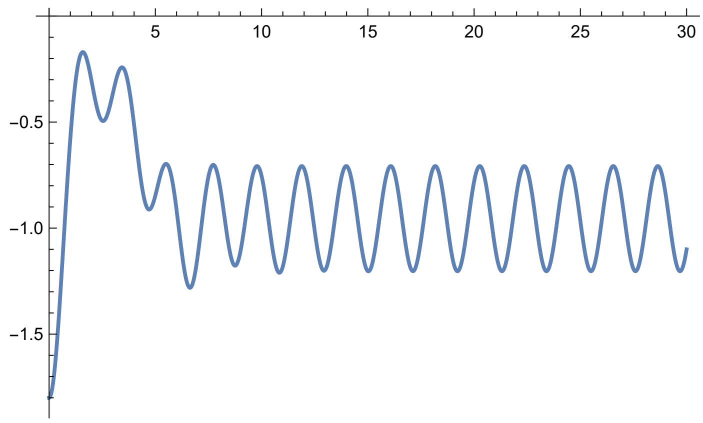
{{x[t] → InterpolatingFunction[Domain: {{0., 100.}} Output: scalar][t]}}

Out[*]=

InterpolatingFunction[Domain: {{0., 100.}} Output: scalar][t]

In[*]:=
```
data = Table[{t, solnX[t]}, {t, 0, 30, 0.01}];
```

In[*]:=
```
graphics3 = ListLinePlot[data]
```
Out[*]=



In[*]:=
```
A = 2;
ω = 1.5;
γ = 1.8;
```

In[ ]:=
```
eqn1 := {x''[t] + γ * x'[t] == x[t] - (x[t])^3 + A Sin[2 * ω * t], x[0] == -1.8, x'[0] == 0.0};
soln = NDSolve[eqn1, x[t], {t, 0, 100}]
solnX[t_] = x[t] /. Flatten[soln]
```

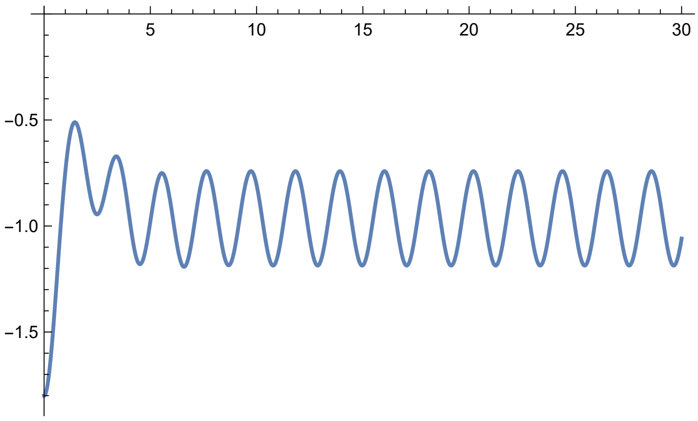
Out[ ]=
```
{{x[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]}}
```

Out[ ]=
```
InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]
```

In[ ]:=
```
data = Table[{t, solnX[t]}, {t, 0, 30, 0.01}];
```

In[ ]:=
```
graphics4 = ListLinePlot[data]
```

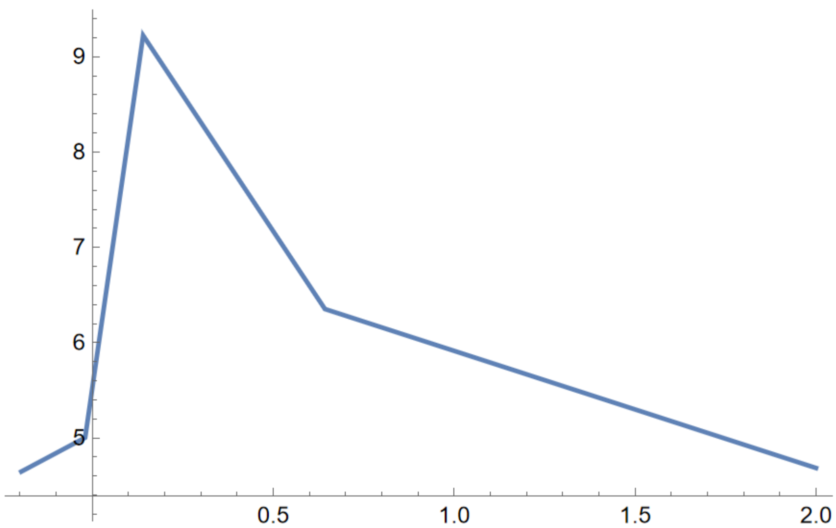
Out[ ]=



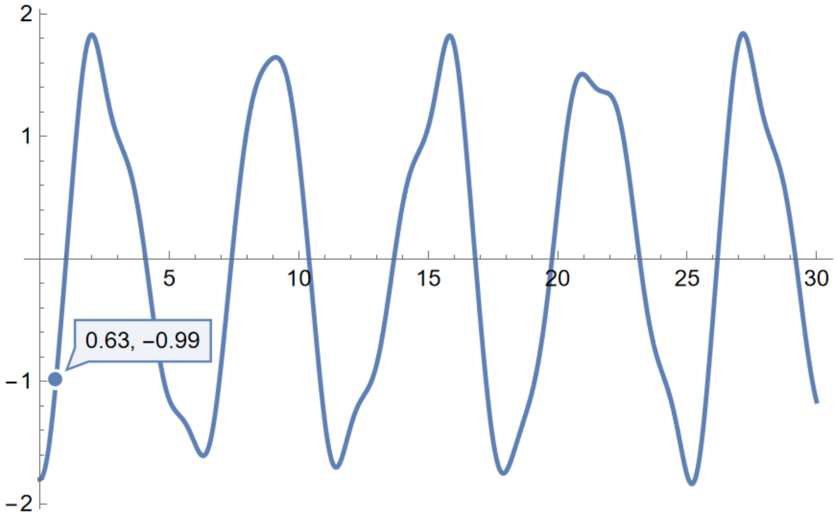
In[1]:=
```
GraphicsColumn[{graphics1, graphics2, graphics3, graphics4}]
```

Results

```
(*time period vs energy*)
```

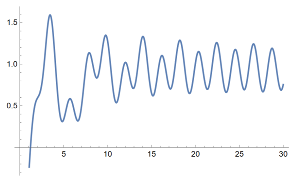


```
(*gamma=0*)
```

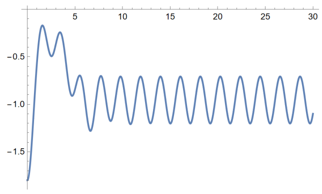



```
(*Changing the values of γ*)
```

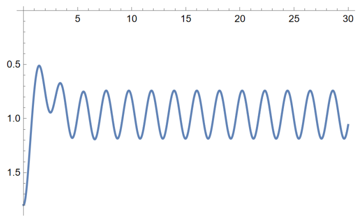
```
(*gamma=0.6*)
```

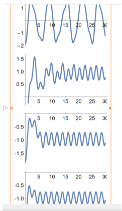



```
(*Gamma=1.2*)
```




```
(*gamma=1.8*)
```



 (*using GraphicsColumn*)

Discussions

This assignment tests knowledge of plotting, integration , solving differential equations and the displaying the graphs together using GraphicsColumn[]

Comments

This assignment felt very long due to the manually changing the values of energy in part 1 and the values of gamma in part 2.

References