Assignment4-On ordinary non linear differential equation

Sripathi ravi
ph1050
department of engineering physics

aim

To represent and solve a differential equation using mathematica and plot its graph and analyse it.

problem statement:

part1:
1) we are required to write potential energy expression and plot potential energy vs x
2)find time period for six different values of total energy
part 2:
1)find force equation
2)write the equation of motion with damping constant n.
3)solve the differential equation and plot x[t] vs t for n=0,0.6,1.2,1.8 and comment about motion

code organisation:

part 1:
1)writing potential equation and plotting potential vs x between x=[-2,2]
2)finding equation for time period and solving it to get values of time period.
3)plotting time period vs total energy
part 2:
1) finding force
2)writing equations of motion with n
3)solving differential equation
4)plotting x[t] vs t for n=0,0.6,1.2,1.8

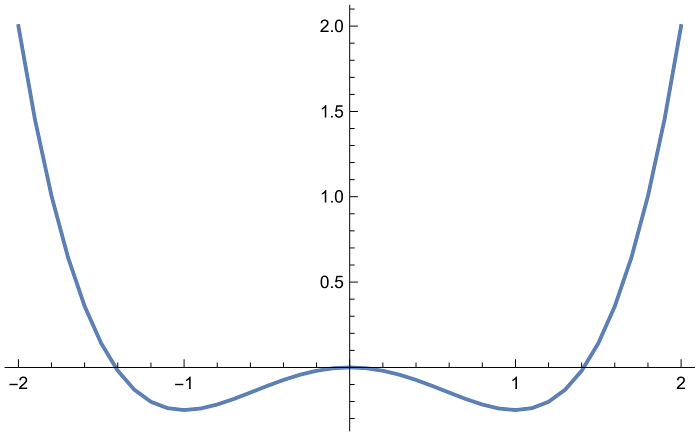
computational code

In[∘]:=
```
vP[x_] = x^4/4 - x^2/2
data = Table[{x, vP[x]}, {x, -2, 2, 0.1}];
ListLinePlot[data]

f[e_] = (2 * (e - vP[x])) ^ (1 / 2)
```

Out[∘]=

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



Out[∘]=

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

In[∘]:=
```
a = Integrate[1 / f[0.067], {x, -1.45, 1.45}]
```
Out[∘]=

5.18009

In[∘]:=
```
b = Integrate[1 / f[0.3], {x, -1.57, 1.57}]
```
Out[∘]=

3.71935

In[∘]:=
```
c = Integrate[1 / f[0.623], {x, -1.69, 1.69}]
```
Out[∘]=

3.10593

In[∘]:=
```
d = Integrate[1 / f[0.182], {x, -1.52, 1.52}]
```
Out[∘]=

4.27683

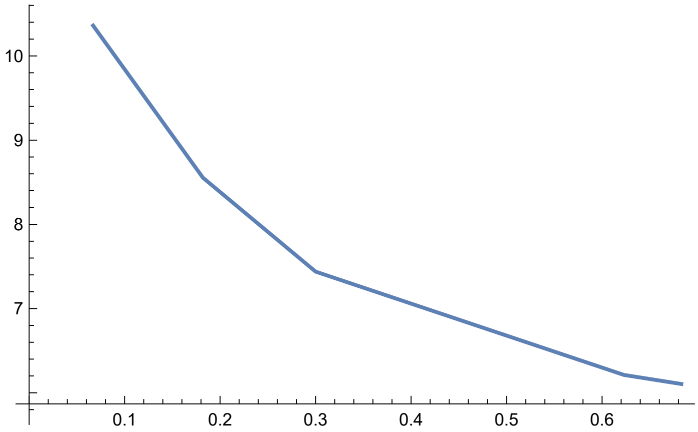
In[∘]:=
```
e = Integrate[1 / f[0.683], {x, -1.71, 1.71}]
```
Out[∘]=

3.05231

In[∘]:=
```
data = {{0.067, 2 a}, {0.182, 2 d}, {0.3, 2 b}, {0.623, 2 c}, {0.683, 2 e}}
```
Out[∘]=

{{0.067, 10.3602}, {0.182, 8.55367}, {0.3, 7.4387}, {0.623, 6.21185}, {0.683, 6.10462}}

In[◦]:=
```
ListLinePlot[data]
```
Out[◦]=




```
force[x] := D[-vP[x], x]
```
Out[◦]=

$x - x^3$

In[◦]:=
```
force[x]
```
Out[◦]=

$x - x^3$

In[◦]:=
```
eqn1 := {x''[t] - x[t] + (x[t])^3 + n x'[t] == A Sin[2 w t], x[0] == 1.8, x'[0] == 0} /.
   {A → 2, w → 1.5, n → 0}
soln1 = NDSolve[eqn1, x[t], {t, 0, 100}]
solnT1[t_] = x[t] /. Flatten[soln1]
```
Out[◦]=

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

Out[◦]=

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

In[◦]:=
```
eqn2 := {x''[t] - x[t] + (x[t])^3 + n x'[t] == A Sin[2 w t], x[0] == 1.8, x'[0] == 0} /.
   {A → 2, w → 1.5, n → 0.6}
soln2 = NDSolve[eqn2, x[t], {t, 0, 100}]
solnT2[t_] = x[t] /. Flatten[soln2]
```
Out[◦]=

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

Out[◦]=

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

In[ ]:=
```
eqn3 := {x''[t] - x[t] + (x[t])^3 + n x'[t] == A Sin[2 w t], x[0] == 1.8, x'[0] == 0} /.
   {A → 2, w → 1.5, n → 1.2}
soln3 = NDSolve[eqn3, x[t], {t, 0, 100}]
solnT3[t_] = x[t] /. Flatten[soln3]
```

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

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

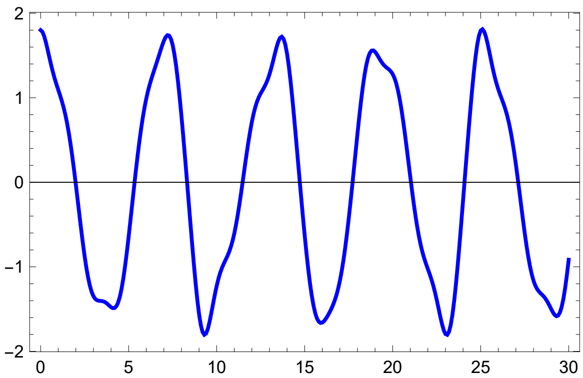
In[ ]:=
```
eqn4 := {x''[t] - x[t] + (x[t])^3 + n x'[t] == A Sin[2 w t], x[0] == 1.8, x'[0] == 0} /.
   {A → 2, w → 1.5, n → 1.8}
soln4 = NDSolve[eqn4, x[t], {t, 0, 100}]
solnT4[t_] = x[t] /. Flatten[soln4]
```

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

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

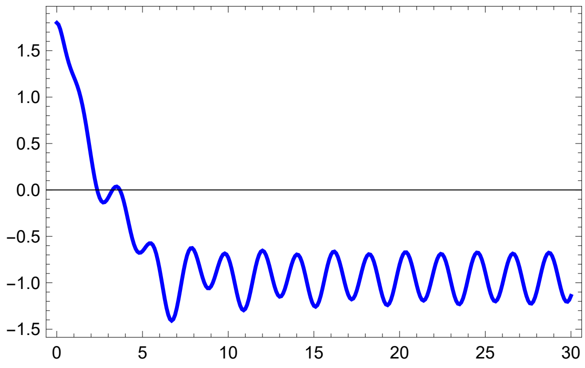
In[ ]:=
```
data1 = Table[{t, solnT1[t]}, {t, 0, 30, 0.1}];
ListLinePlot[data1, PlotRange → All,
 PlotStyle → {Blue, PointSize[0.015]}, Frame → True, ImageSize → {300, 200}]
```

Out[ ]=



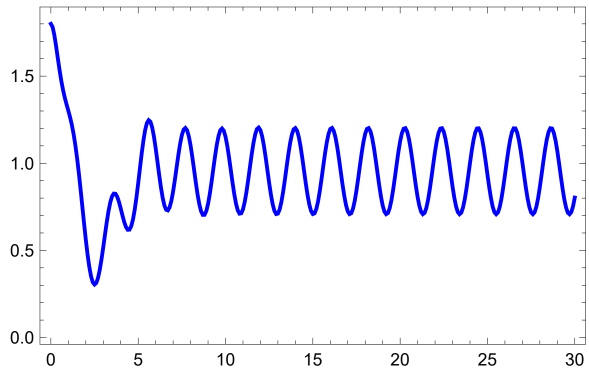
In[•]:=
```
data2 = Table[{t, solnT2[t]}, {t, 0, 30, 0.1}];
ListLinePlot[data2, PlotRange → All,
 PlotStyle → {Blue, PointSize[0.015]}, Frame → True, ImageSize → {300, 200}]
```

Out[•]=



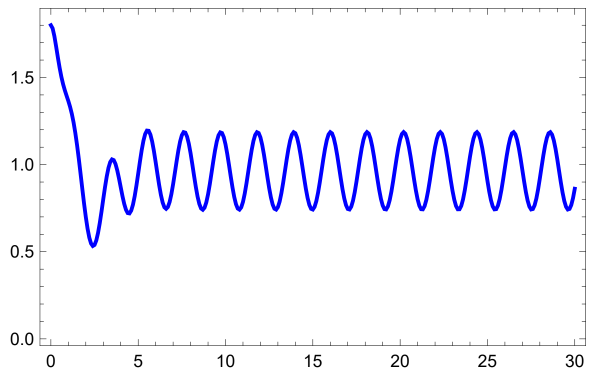
In[•]:=
```
data3 = Table[{t, solnT3[t]}, {t, 0, 30, 0.1}];
ListLinePlot[data3, PlotRange → All,
 PlotStyle → {Blue, PointSize[0.015]}, Frame → True, ImageSize → {300, 200}]
```

Out[•]=



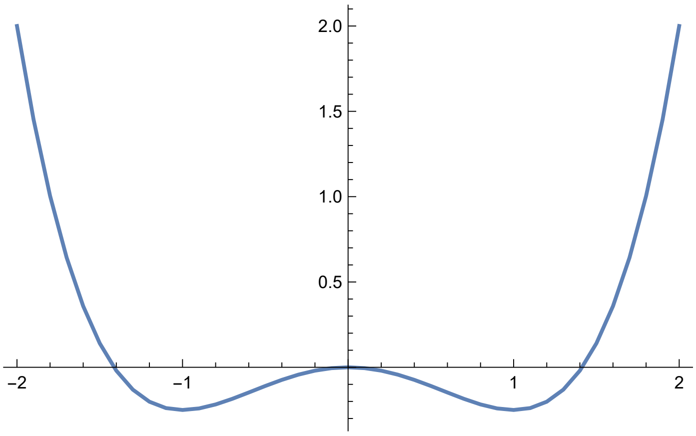
In[•]:=
```
data4 = Table[{t, solnT4[t]}, {t, 0, 30, 0.1}];
ListLinePlot[data4, PlotRange → All,
 PlotStyle → {Blue, PointSize[0.015]}, Frame → True, ImageSize → {300, 200}]
```

Out[•]=

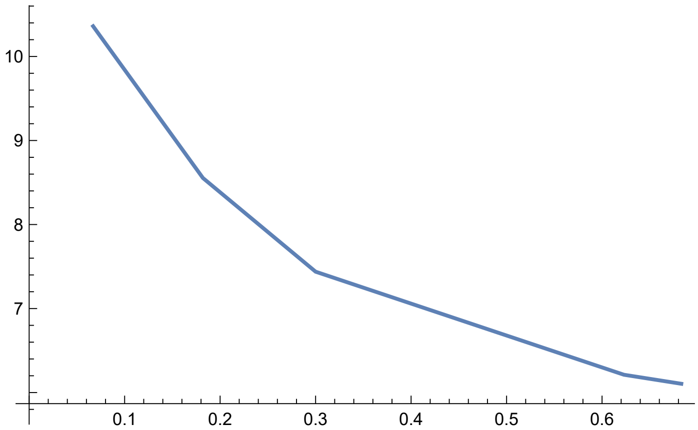


result:

plot of potential vs x:

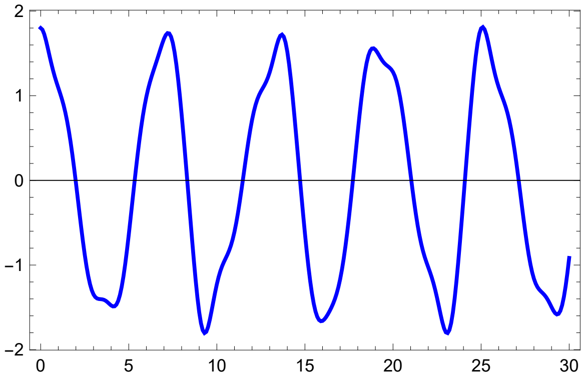


plot of time period vs total energy:

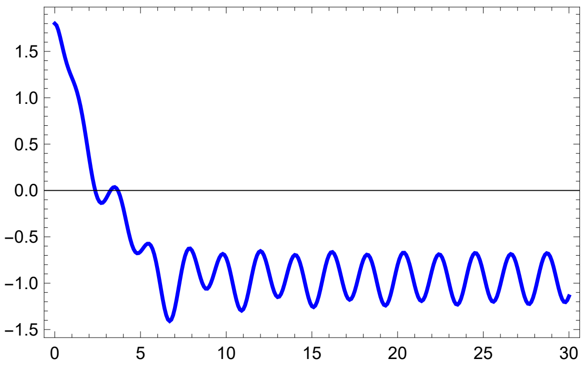


plot of x[t] vs t:

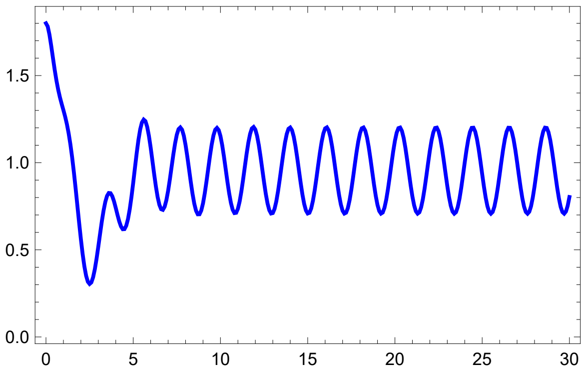
n=0:



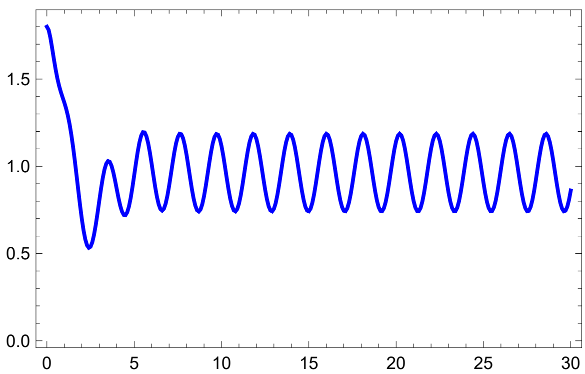
n = 0.6 :



n = 1.2 :



n = 1.8 :



discussion:

1)n=0: in this case there is no damping . hence the particle executes periodic motion about x=0 with an amplitude of 1.8
2)n=0.6:in this case the particle starts at x=1.8 and goes till x=-0.5 . then it takes a turn and goes up a small distance and comes down again. in this case the damping is very less. after this damping becomes negiligible due to low velocity . Hence finally the particle achives steady state where it executes periodic motion about x=-1 with amplitude 0.3.
3)n=1.2: in this case the daming is strong such that the particle released from x=1.8 doesnt even

cross the origin. it turns at =0.5 and then go up and achive steady state where it oscillates about x=1 with amplitude of 0.2
4)n=1.8: this is similar to last case. but damping is much stronger. here in steady state it executes periodic motion about x=1 with amplitude 0.1

comment:

I learnt how to write and solve a differential equation in mathematica. due to time constrains i wasn't able to attend the thid part

references:

1)mathematica documentation
2)mathematica stack exchange.