In[81]:=

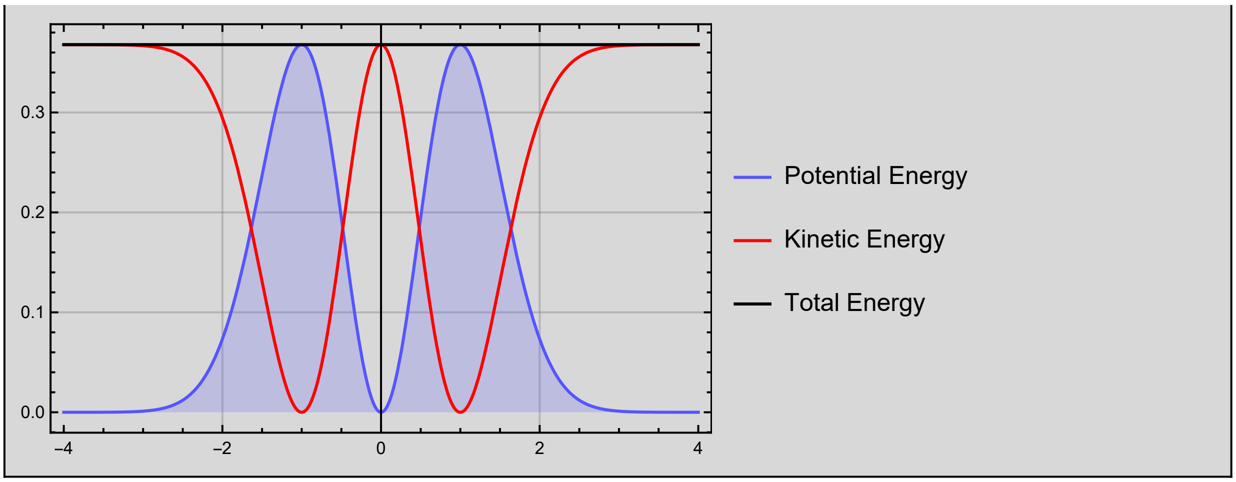
```
(* Assignment 2 *)      Due : 11.00 pm, 24.12 × .22

(*

Using the potential energy function u(x) =  x^2 Exp(-x^2)
write a code that will perform the following:

( Realise that the system admits bound and unbound motion.)

1. Plot the potential energy and kinetic energy
 (assume suitable total energy). You may want to choose a suitable domain
 and range so that the main features of the potential energy are captured.

2. Write down the Hamiltonian (here, the total energy)
 and Taylor expand the same near the critical points.

3. Using the above informaiton draw phase portraits near the critical points.

4. Draw the general phase portrait, without making any assumption.

5. Find the time period of small oscillations and also
 the large oscillations when the mass is executing bound motion.
  (Recall that the function "Integrate" performs symbolic integration,
   while "NIntegrate" performs numerical integration.)

6. Plot the time period of oscillations (for bound motion) as
 a function of amplitude, for at least five different amplitudes.

*)
```

In[81]:=

```
Print["Part 1:- "]
u[x_] := x^2 e^(-x^2)
peplot = Plot[u[x], {x, -4, 4}, PlotStyle → {Lighter[Blue]}, Frame → True,
   PlotLegends → {"Potential Energy"}, GridLines → Automatic, Filling → Bottom];
teplot = Plot[1/e, {x, -4, 4}, PlotStyle → {Black}, PlotLegends → {"Total Energy"}];
ke[x_] = e^-1 - u[x];
keplot = Plot[ke[x], {x, -4, 4}, PlotStyle → {Red},
   Frame → True, PlotLegends → {"Kinetic Energy"}, GridLines → Automatic];
Show[peplot, keplot, teplot]
Print["Without an appropriate domain or range, Mathematica
   does not accurately plot the main features of the curve."]
Print["We clearly see that the critical points are 0, 1, -1"]
```

Part 1:-

Out[87]=



Without an appropriate domain or range,
  Mathematica does not accurately plot the main features of the curve.

We clearly see that the critical points are 0, 1, -1

In[90]:=

```
Print["Part 2 and 3:- "]
ham[p_, x_] = p^2/(2 m) + x^2 e^(-x^2);
```

Part 2 and 3:-

In[92]:=

```
Print["Taylor Expansion about x=0:- "]
tayExp1[p_, x_] := Series[ham[p, x], {x, 0, 3}]
hamNr1[p_, x_] := Normal[tayExp1[p, x]]
traj1[p_, x_] = hamNr1[p, x] /. {m → 1}
```

Taylor Expansion about x=0:-

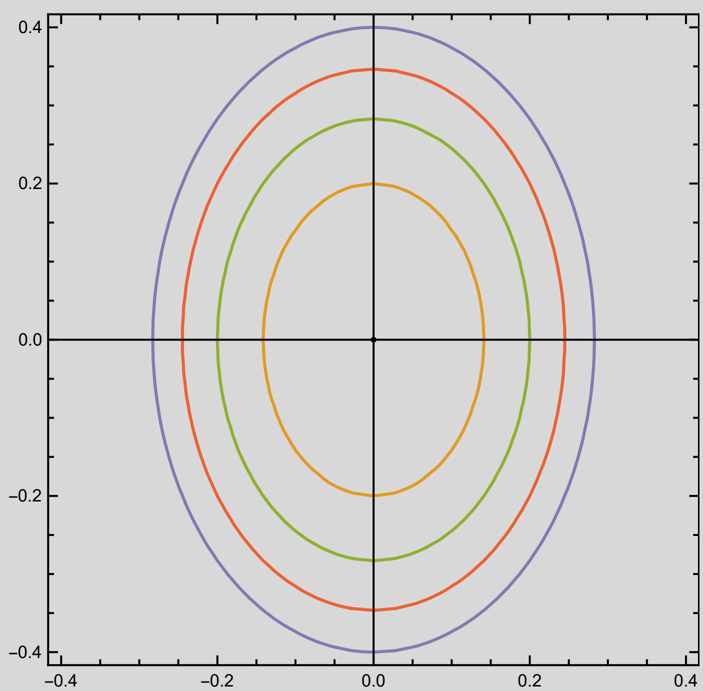
Out[95]=

$$\frac{p^2}{2} + x^2$$

In[96]:=

```
dat1 = Table[traj1[p, x] == j, {j, 0.00001, 0.1, 0.02}];
cplot1 = ContourPlot[Evaluate[dat1], {x, -0.4, 0.4}, {p, -0.4, 0.4}];
axes1 = {Line[{{0, -0.5}, {0, 0.5}}], Line[{{-0.5, 0}, {0.5, 0}}]};
axNr1 = Graphics[{Thin, Black, axes1}];
pointNr1 = Graphics[{Point[{0, 0}]}];
Show[cplot1, axNr1, pointNr1]
```

Out[101]=



In[102]:=

```
Print["Taylor Expansion about x=1:- "]
tayExp2[p_, x_] := Series[ham[p, x], {x, 1, 3}]
hamNr2[p_, x_] := Normal[tayExp2[p, x]]
traj2[p_, x_] = hamNr2[p, x] /. {m → 1}
```

Taylor Expansion about x=1:-

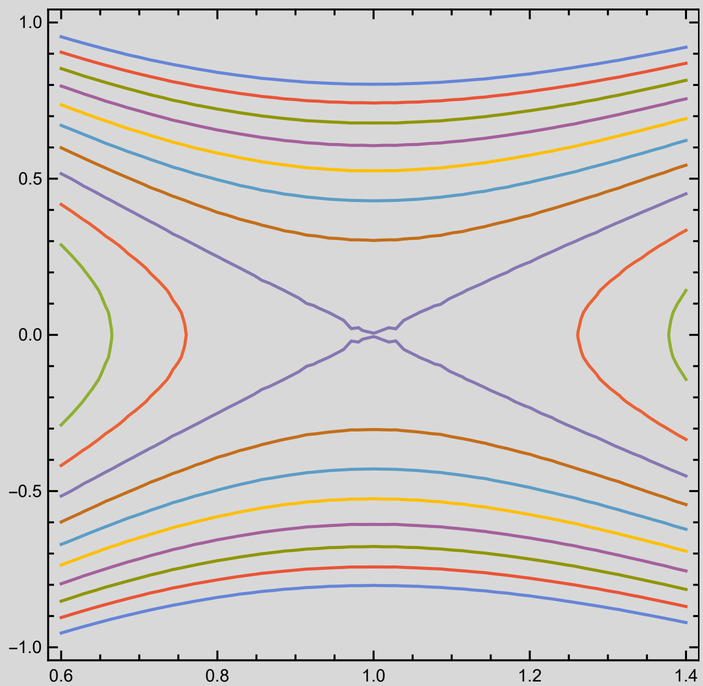
Out[105]=

$$\frac{1}{e} + \frac{p^2}{2} - \frac{2(-1+x)^2}{e} + \frac{2(-1+x)^3}{3e}$$

In[106]:=

```
dat2 = Table[traj2[p, x] == j, {j, 1/(2 e) + 0.0001, 2/e, 1/(8 e)}];
cplot2 = ContourPlot[Evaluate[dat2], {x, 0.6, 1.4}, {p, -1, 1}]
```

Out[107]=



In[108]:=

```
Print["Taylor Expansion about x=-1:- "]
tayExp3[p_, x_] := Series[ham[p, x], {x, -1, 3}]
hamNr3[p_, x_] := Normal[tayExp3[p, x]]
traj3[p_, x_] = hamNr3[p, x] /. {m → 1}
```

Taylor Expansion about x=-1:-

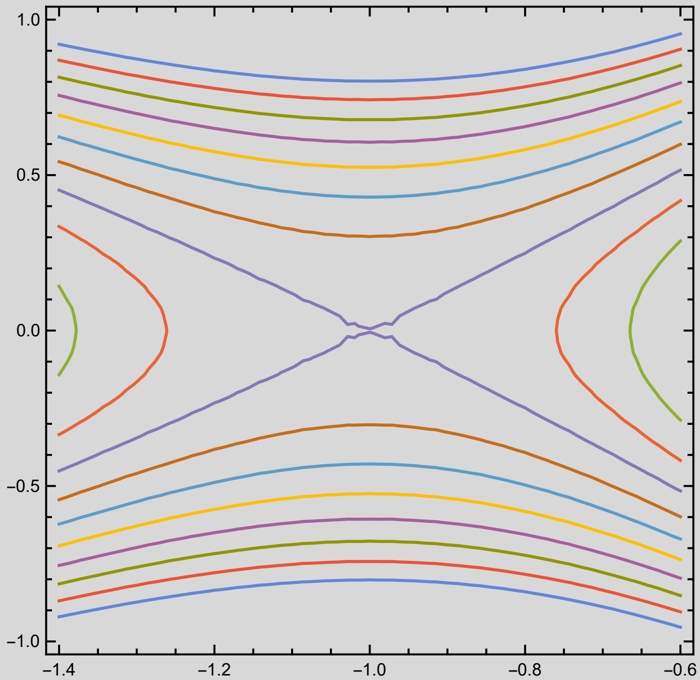
Out[111]=

$$\frac{1}{e} + \frac{p^2}{2} - \frac{2(1+x)^2}{e} - \frac{2(1+x)^3}{3e}$$

In[112]:=

```
dat3 = Table[traj3[p, x] == j, {j, 1/(2 ⅇ) + 0.0001, 2/ⅇ, 1/(8 ⅇ)}];
cplot3 = ContourPlot[Evaluate[dat3], {x, -1.4, -0.6}, {p, -1, 1}]
```

Out[113]=



In[114]:=

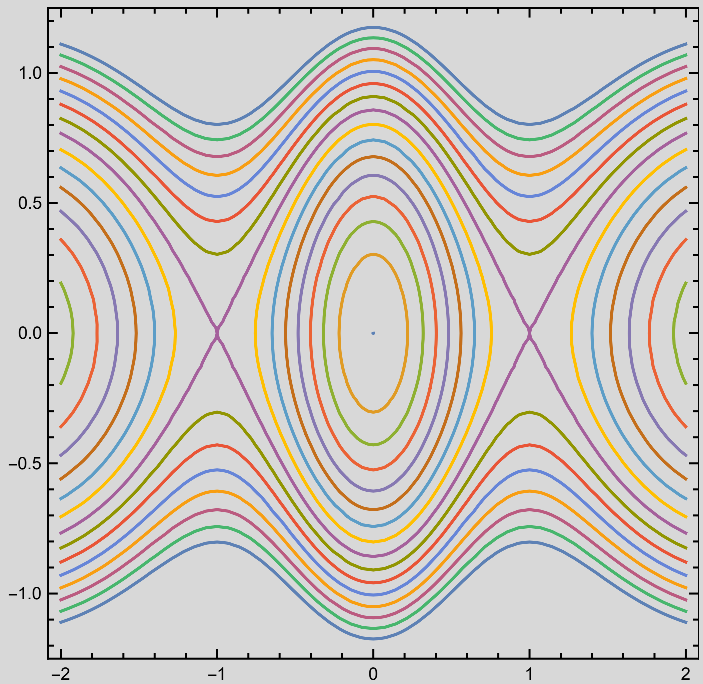
```
Print["Part 4:- "]
(*General phase portrait*)
genTraj[p_, x_] = ham[p, x] /. m → 1
genList = Table[genTraj[p, x] == j, {j, 0.00001, 2/e, 1/(8 e)}];
genPlot = ContourPlot[Evaluate[genList], {x, -2, 2}, {p, -1.2, 1.2}]
```

Part 4:-

Out[115]=

$$\frac{p^2}{2} + e^{-x^2} x^2$$

Out[117]=



In[118]:=

```
Print["Part 5:- "]
Print["To find T for small oscillations, we use Taylor's expansion:- "]
Print[]
uSmall[x_] = Series[u[x], {x, 0, 3}];
uNrSmall[x_] = Normal[uSmall[x]];
forSmall[x_] = -D[uNrSmall[x], x];
(*Note the use of the function for differentiation*)
Print["f=-du/dx"]
Print["Taylor's Expansion of u = ", uNrSmall[x]]
Print["On differentiating the Taylor's
   expansion of u, we get f = -kx = -2x. Therefore, k=2."]
Print["Also, ω=Sqrt[k/m] and ω=Sqrt[2] for m= 1 kg."]
Print["T = 2π/ω = Sqrt[2]π for small oscillations."]
```

Part 5:-

To find T for small oscillations, we use Taylor's expansion:-

$f=-\frac{du}{dx}$

Taylor's Expansion of u = x^2

On differentiating the Taylor's expansion of u, we get f = -kx = -2x. Therefore, k=2.

Also, $\omega=\sqrt{\frac{k}{m}}$ and $\omega=\sqrt{2}$ for m= 1 kg.

$T = \frac{2\pi}{\omega} = \sqrt{2}\pi$ for small oscillations.

In[129]:=

```
Print["To find T for large oscillations:- "]
Print[]
Print["KE = TE - PE"]
Print["1/2 m ẋ^2 = TE - x^2 e^(-x^2)"]
Print["∫_0^A dx/Sqrt[TE - x^2 e^(-x^2)] = ∫_0^(T/4) Sqrt[2/m] dt = ∫_0^(T/4) Sqrt[2] dt = Sqrt[2] T/4 for m=1"]
Print["T= 4/Sqrt[2] ∫_0^A dx/Sqrt[A^2 e^(-A^2) - x^2 e^(-x^2)] for large oscillations in bound motion."]
(*TE = Umax = U when x=A*)
t[A_] := 4/Sqrt[2] NIntegrate[1/Sqrt[A^2 e^(-A^2) - x^2 e^(-x^2)], {x, 0, A}]
  (*Note that we cannot use Integrate instead of NIntegrate,
  because the function "Integrate" performs symbolic integration,
  while "NIntegrate" performs numerical integration.*)
```

To find T for large oscillations:-

KE = TE - PE

$$\frac{1}{2} m \dot{x}^2 = TE - x^2 e^{-x^2}$$

$$\int_0^A \frac{dx}{\sqrt{TE - x^2 e^{-x^2}}} = \int_0^{T/4} \sqrt{\frac{2}{m}}\, dt = \int_0^{T/4} \sqrt{2}\, dt = \frac{\sqrt{2}\, T}{4} \text{ for m=1}$$

$$T= \frac{4}{\sqrt{2}} \int_0^A \frac{dx}{\sqrt{A^2 e^{-A^2} - x^2 e^{-x^2}}} \text{ for large oscillations in bound motion.}$$

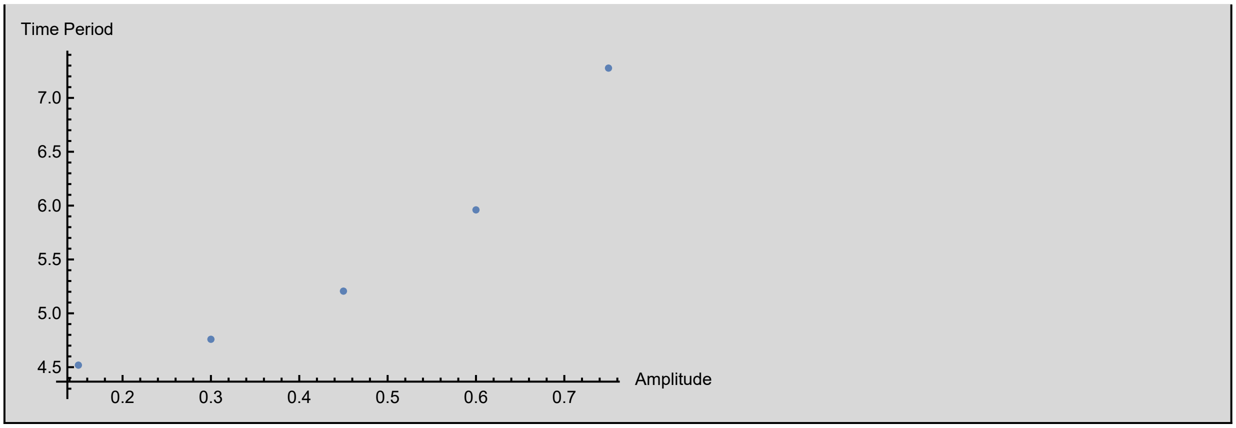
In[136]:=

```
Print["Part 6:- "]
Print["For 5 different amplitudes:- (that is, A = 0.15, 0.30, 0.45, 0.60, 0.75) "]
aLSt = List[{0.15, t[0.15]}, {0.3, t[0.3]},
   {0.45, t[0.45]}, {0.6, t[0.6]}, {0.75, t[0.75]}];
aPLt = ListPlot[aLSt, AxesLabel → {"Amplitude", "Time Period"}]
```

Part 6:-

For 5 different amplitudes:- (that is, A = 0.15, 0.30, 0.45, 0.60, 0.75)

Out[139]=



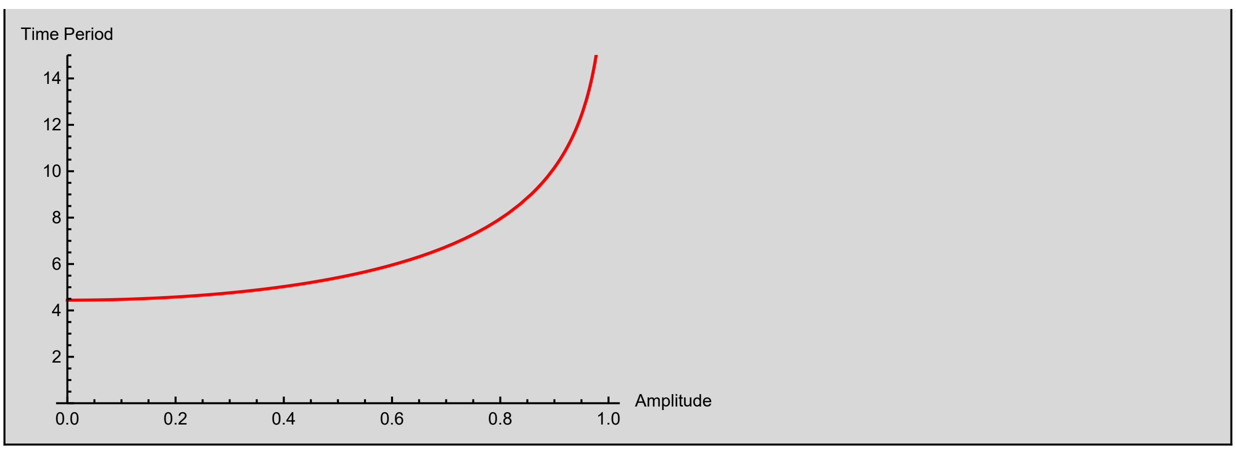
In[140]:=

```
(*Extra points*)
```

In[141]:=

```
(*Plotting time period as a function of amplitude*)
Plot[t[A], {A, 0, 1}, PlotRange → {0, 15},
 PlotStyle → {Red}, AxesLabel → {"Amplitude", "Time Period"}]
Print["Note that T tends to √2 π as
   A tends to 0, as proved above for small oscillations."]
```

Out[141]=

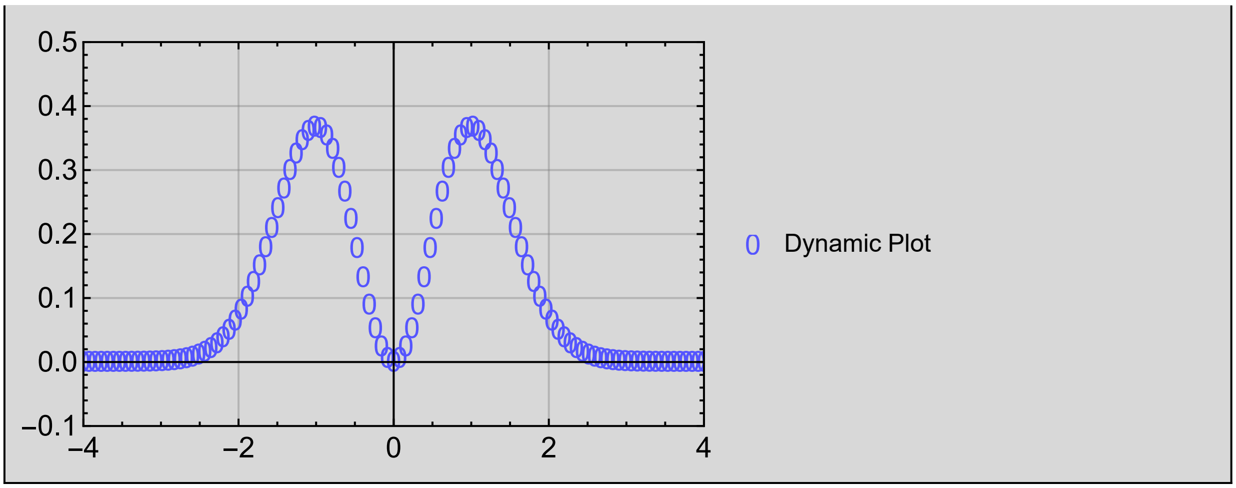


Note that T tends to $\sqrt{2}\pi$ as A tends to 0, as proved above for small oscillations.

In[151]:=

```
(*Animating potential energy using dynamic*)
f[n_] := (Pause[0.01]; n)
Dynamic[ListPlot[datSlow,
    PlotRange → {{-4, 4}, {-0.1, 0.5}}, PlotLegends → {"Dynamic Plot"},
        PlotStyle → Directive[Lighter[Blue], Thick],
  Frame → True, ImageSize → Medium,
        GridLines → Automatic, BaseStyle → {FontSize → 15}, PlotMarkers → "0"]]
n = -3 Pi;
datSlow = {};
While[n ≤ 3 Pi,
    datSlow = AppendTo[datSlow, {N[f[n]], Chop[u[N[f[n]]]]}];
    n = n + Pi / 40;
    ]
```

Out[152]=



In[148]:=

```
(*Another interesting point*)
Print[
 "We notice that the graphs plotted for critical points 1 and -1, using Taylor's
   expansion are not exactly symmetric about 1 and -1 respectively."]
Print["This is due to the (-1+x)^3 and
   (1+x)^3 terms respectively in the final Taylor's expansion."]
Print["We can make it symmetric by using Taylor's
   expansion to an even order, which eliminates the cubic term."]
```

We notice that the graphs plotted for critical points 1 and -1, using Taylor's expansion are not exactly symmetric about 1 and -1 respectively.

This is due to the $(-1+x)^3$ and $(1+x)^3$ terms respectively in the final Taylor's expansion.

We can make it symmetric by using Taylor's expansion to an even order, which eliminates the cubic term.