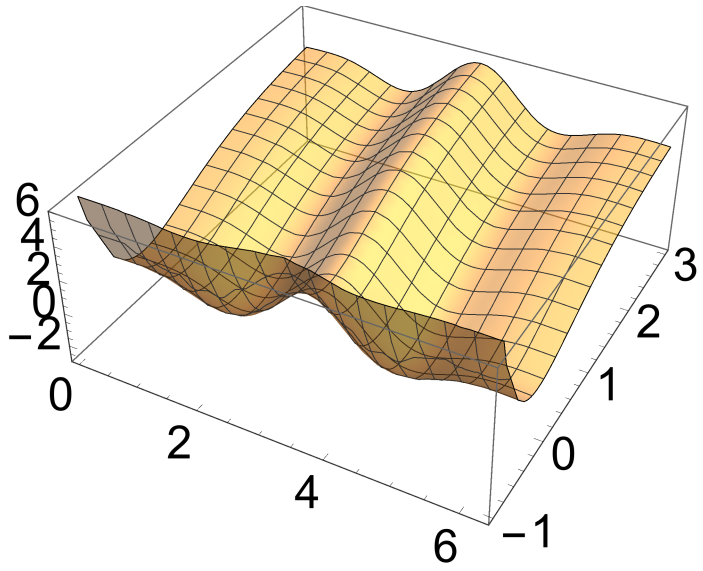
In[1]:=
```
(*pot[x_,y_]:=(x-1)^2*(x+1)^2+.5x+Sin[4y]+1.4y^2
```

In[1]:=
```
pot[x_, y_] := -2 Cos[2 x - π] * E^-((x - π)^2 / 3) + 4 (1 - E^-y)^2
```

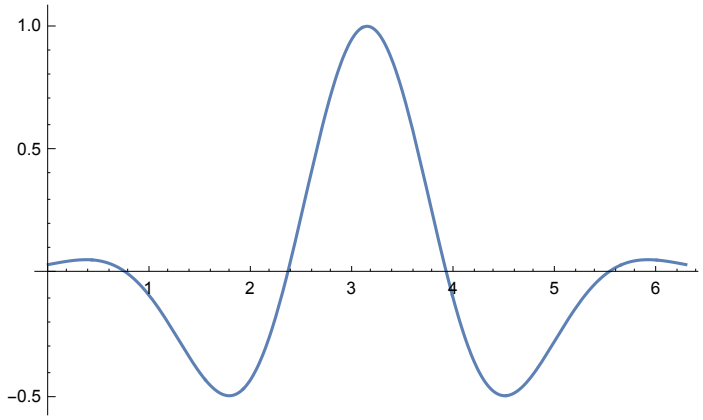
In[2]:=
```
Plot3D[pot[x, y], {x, 0, 2 π}, {y, -1, 3}, PlotRange → {-3, 6},
 PlotStyle → Opacity[.5], LabelStyle → Large, ClippingStyle → None]
```

Out[2]=



In[3]:=
```
Plot[-Cos[2 x - π] * E^-((x - π)^2 / 3), {x, 0, 2 π}]
```

Out[3]=



In[4]:=
```
Plot[2 (1 - E^-y)^2, {y, -11, 11}, PlotRange → {0, 11}];
```

In[5]:=

linFit[] is a function that returns a linear interpolation approximation for Energy at point {xi,yi}, given two endpoints and the values for

{xi,yi}

midPtTest[] is a function that returns the difference in energy between the linear interpolation approximation and the energy value from the cheap potential at the point halfway between {x1, y1} and {x2, y2}. I included cutoffs as well to prevent division by zero errors.

In[6]:=
```
linFit[x1_, x2_, y1_, y2_, z1_, z2_, xi_, yi_] := Module[{dx, dy, dz, mx, my, b},
  dx = x2 - x1;
  If[dx ≥ 0 && dx < .000001, dx = .000001];
  If[dx ≤ 0 && dx > -.000001, dx = -.000001];
  dy = y2 - y1;
  If[dy ≥ 0 && dy < .000001, dy = .000001];
  If[dy ≤ 0 && dy > -.000001, dy = -.000001];
  dz = z2 - z1;
  mx = dz / dx;
  my = dz / dy;
  b = z1 - mx * x1 - my * y1;
  mx * xi + my * yi + b
 ]
midPtTest[x1_, x2_, y1_, y2_] := Module[{linE, deltaE, xTest, yTest, z1, z2},
  xTest = (x1 + x2) / 2;
  yTest = (y1 + y2) / 2;
  z1 = pot[x1, y1];
  z2 = pot[x2, y2];
  linE = linFit[x1, x2, y1, y2, z1, z2, xTest, yTest];
  deltaE = Abs[linE - pot[xTest, yTest]];
  {{xTest, yTest}, deltaE}
 ]
```

In[8]:=
```
Manipulate[
  ListPlot[{
    {{starPoints[[i, 1]], starPoints[[i, 3]]},
     {starPoints[[i, 2]], starPoints[[i, 4]]}},
    {midPtTest[starPoints[[i, 1]], starPoints[[i, 2]],
       starPoints[[i, 3]], starPoints[[i, 4]]][[1]]}
   }, PlotMarkers → {Automatic, Large}], {i, 1, Length[starPoints], 1}];
```

In[9]:=
```
(*gridSearch[steps_,lbound_,ubound_,fixed_]:=Module[
  {dx=.01,xL,xR,xLOld,xROld,i,pts={},cutOff=.05,
   testList,testOut,newPts,hold, tagged,posListTag,posList,n,grid=.5},
  testList=Table[i,{i,lbound,ubound,.5}];
  pts=testList;
  For[n=0,n<steps,n++,
   testOut=
    Table[midPtTest[testList[[i]],testList[[i+1]]],{i,1,Length[testList]-1,1}];
   newPts=testOut[[All,1]];
   hold=Sort[Join[pts,newPts]];
   pts=hold;
   tagged=Select[testOut,#[[2]]>cutOff&][[All,1]];
   posListTag=Table[Position[pts,tagged[[i]]],{i,1,Length[tagged],1}]//Flatten;
   posList=Flatten[Table[{posListTag[[i]]-1,posListTag[[i]],posListTag[[i]]+1},
      {i,1,Length[posListTag],1}]];
   testList=Table[pts[[posList[[i]]]],{i,1,Length[posList]}];
  ];
  Table[{pts[[i]],pot[pts[[i]],fixed]},{i,1,Length[pts],1}]
 ]
```

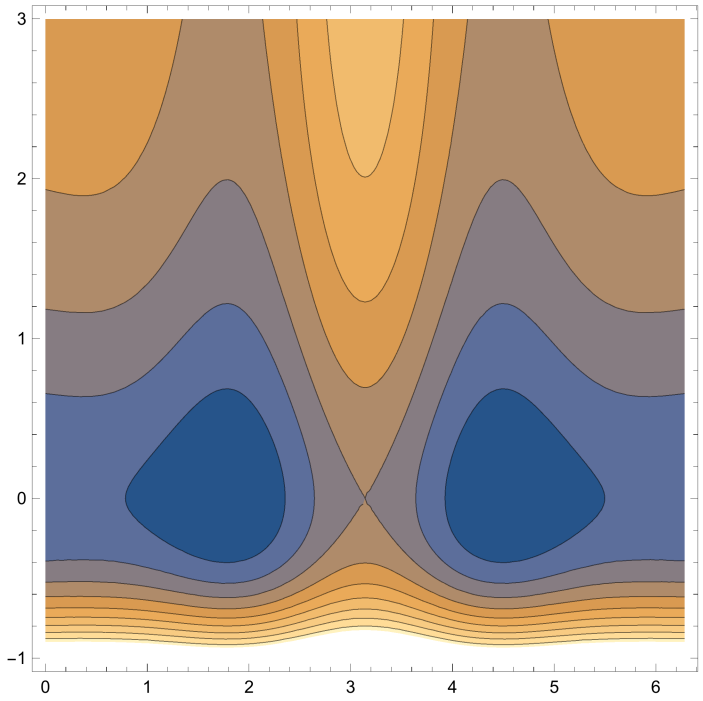
gridSearch[] is the main function creating the Adaptive Grid Seach. It keeps track of a list of test points called testList. testList begins as an evenly spaced grid of points along the line connecting the configuration space points {xLo, yLo} and {xHi, yHi}. The while statement keeps performing the interpolation approximation check (i.e. Is the interpolation energy value within the cutoff of the energy value returned by the cheap potential?) between all adjacent pairs of points in testList until testList is empty.

In[9]:=
```
gridSearch[cutOff_, xLo_, xHi_, yLo_, yHi_] := Module[
  {dx = .01, xL, xR, xLOld, xROld, i, xRange, yRange, pts = {},
   testList, testOut, newPts, hold, tagged,
   posListTag, posList, n, gridSteps = 10, gridSizeX, gridSizeY},
  xRange = xHi - xLo;
  yRange = yHi - yLo;
  gridSizeX = xRange / gridSteps;
  gridSizeY = yRange / gridSteps;
  testList = Table[{xLo + i * gridSizeX, yLo + i * gridSizeY}, {i, 0, gridSteps, 1}];
  pts = testList;
  (*For[n=0,n<steps,n++, *)
  While[Length[testList] > 0,
   testOut = Table[midPtTest[testList[[i, 1]], testList[[i + 1, 1]],
      testList[[i, 2]], testList[[i + 1, 2]]], {i, 1, Length[testList] - 1, 1}];
   newPts = testOut[[All, 1]];
   hold = Sort[Join[pts, newPts]];
   pts = hold;
   tagged = Select[testOut, #[[2]] > cutOff &][[All, 1]];
   posListTag =
    Table[Position[pts, tagged[[i]]], {i, 1, Length[tagged], 1}] // Flatten;
   posList = Flatten[Table[{posListTag[[i]] - 1, posListTag[[i]],
       posListTag[[i]] + 1}, {i, 1, Length[posListTag], 1}]];
   testList = Table[pts[[posList[[i]]]], {i, 1, Length[posList]}];
  ];
  Table[{pts[[i, 1]], pts[[i, 2]], N[pot[pts[[i, 1]], pts[[i, 2]]]]},
   {i, 1, Length[pts], 1}]
 ]
```

In[10]:=
```
perimPoints = Join[
   Table[{x, -.6}, {x, 0, 2 π, 2 π / 2}],
   Table[{x, 3}, {x, 0, 2 π, 2 π / 2}],
   Table[{0, y}, {y, -.6, 3, 4 / 2}],
   Table[{2 π, y}, {y, -.6, 3, 4 / 2}]];
```

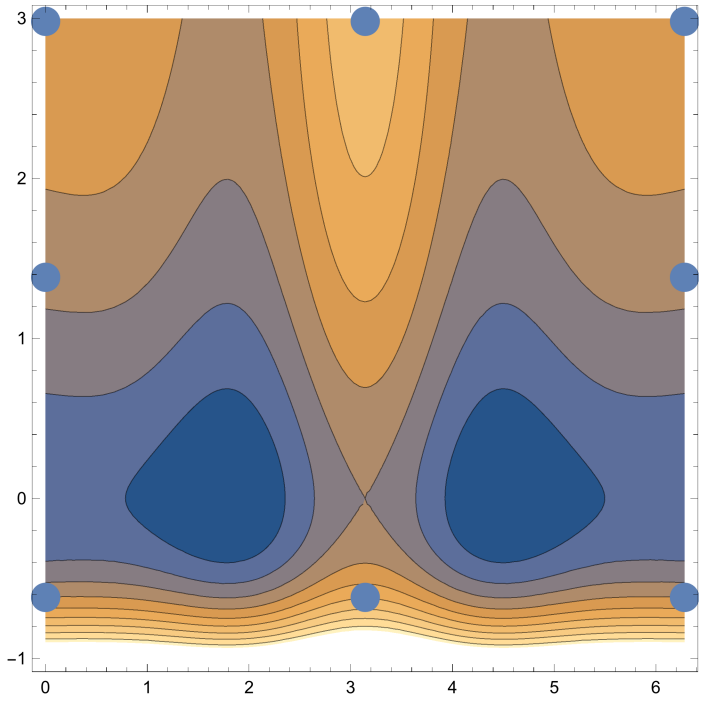
In[11]:=
```
ContourPlot[pot[x, y], {x, 0, 2 π}, {y, -1, 3}]
```

Out[11]=



In[12]:=
```
Show[ContourPlot[pot[x, y], {x, 0, 2 π}, {y, -1, 3}],
 ListPlot[perimPoints, PlotMarkers → {Automatic, Large}]]
```

Out[12]=



In[13]:=
```
starPoints = Table[{1.8, perimPoints[[i, 1]], 0, perimPoints[[i, 2]]},
    {i, 1, Length[perimPoints], 1}] // N
```

Out[13]=
```
{{1.8, 0., 0., -0.6}, {1.8, 3.14159, 0., -0.6}, {1.8, 6.28319, 0., -0.6},
 {1.8, 0., 0., 3.}, {1.8, 3.14159, 0., 3.}, {1.8, 6.28319, 0., 3.}, {1.8, 0., 0., -0.6},
 {1.8, 0., 0., 1.4}, {1.8, 6.28319, 0., -0.6}, {1.8, 6.28319, 0., 1.4}}
```

In[14]:=
```
gridSearch[.1, starPoints[[2, 1]],
  starPoints[[2, 2]], starPoints[[2, 3]], starPoints[[2, 4]]];
```

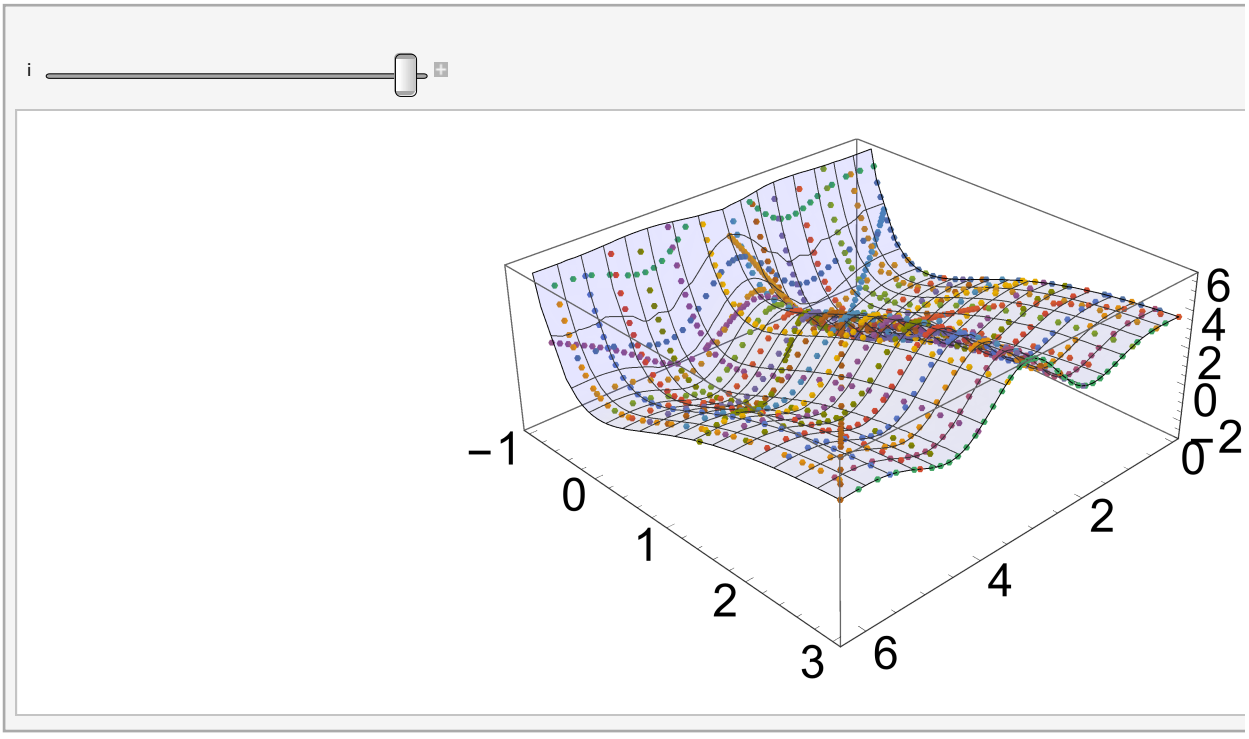
In[15]:=
```
starPoints[[4]]
```

Out[15]=
```
{1.8, 0., 0., 3.}
```

In[16]:=
```
points = {
   Table[gridSearch[.06, 0 + i, 0 + i, -1, 3], {i, 0, 2 π, .5}],
   Table[gridSearch[.1, 0, 6, -1 + i, -1 + i], {i, 0, 4, .25}],
   Table[gridSearch[.3, starPoints[[i, 1]], starPoints[[i, 2]],
     starPoints[[i, 3]], starPoints[[i, 4]]], {i, 1, Length[starPoints], 1}],
   Join[Table[gridSearch[.1, 0 + i, 0 + i, -1, 3], {i, 0, 2 π, .5}],
    Table[gridSearch[.1, 0, 6, -1 + i, -1 + i], {i, 0, 4, .25}],
    Table[gridSearch[.3, starPoints[[i, 1]], starPoints[[i, 2]],
      starPoints[[i, 3]], starPoints[[i, 4]]], {i, 1, Length[starPoints], 1}]]};
```

In[17]:=
```
Manipulate[
 Show[
  Plot3D[pot[x, y], {x, 0, 2 π}, {y, -1, 3}, PlotRange → {-3, 6}, PlotStyle →
    Directive[Blue, Opacity[.1]], LabelStyle → Large, ClippingStyle → None],
  ListPointPlot3D[points[[i]], PlotStyle → PointSize[.01]]
 ], {i, 1, Length[points], 1}]
```

Out[17]=



In[18]:=
```
(*Show[Plot[pot[x],{x,-2,2},PlotRange→{-1,3},PlotStyle→Thickness[.01],
  LabelStyle→Large],ListPlot[asdf,PlotStyle→Red,PlotMarkers→{Automatic,Large}]]
```