Unconstrained minimization problems

Using the package provided in the textbook

You can copy the directory "OptimizationToolbox" in the CD into you *Mathematica* installation directory, under "AddOns -> ExtraPackages". In some operating system (for example, linux), the directory "OptimizationToolbox" is shown as "optimizationtoolbox" (some characters in lower case instead of upper case). So are the files under this directory. you will need to manually rename them to the upper case name, so that *Mathematica* can read it correctly.
After copied the directory, you will be able to use the package. It is possible that there will be some warnings sinc ethe package was written for an older version of *Mathematica*.

- **Example 1**

Use the steepest descent method to find (x, y) that minimizes f(x,y) = $(x+y)^2 + \left(2\left(x^2+y^2-1\right)-1/3\right)^2$. The starting point is (-1.25, 0.25).

In[1]:=
```
Needs["OptimizationToolbox`Unconstrained`"];
```

```
General::obspkg :
 LinearAlgebra`MatrixManipulation` is now obsolete. The legacy version being
   loaded may conflict with current Mathematica functionality.
   See the Compatibility Guide for updating information. >>

General::newpkg :
 NumericalMath`NLimit` is now available as the Numerical Calculus Package.
   See the Compatibility Guide for updating information. >>

SetDelayed::write : Tag Norm in Norm[v_] is Protected. >>

SetDelayed::write : Tag Hessian in Hessian[f_List, vars_] is Protected. >>

SetDelayed::write : Tag Hessian in Hessian[f_, vars_] is Protected. >>

General::stop :
 Further output of SetDelayed::write will be suppressed during this calculation. >>
```

In[2]:=
```
?SteepestDescent
```

```
SteepestDescent[f, vars, x0, opts]. Computes a minimum of f(vars) starting from x0 using
  the steepest descent method. The step length is computed using analytical line
  search. See Options[SteepestDescent] to see a list of options for the function.
  The function returns {x, hist}. x is either the optimum point or the next point
  after MaxIterations. hist contains history of values tried at different iterations.
```

In[3]:=
```
?PlotSearchPath
```

```
PlotSearchPath[f, {x1, x1min, x1max}, {x2, x2min, x2max}, hist, opts]
  shows complete search path superimposed on a contour plot of the function
  f over the specified range. hist is assumed to be of the form {pt1, pt2,
  ...}, where pt1 = {x1,x2} is the first point, etc. The function accepts all
  relevant options of the standard ContourPlot and the Graphics functions.
```

Defin the function, variables, and the starting point:

In[4]:=
```
f = (x + y) ^2 + (2 * (x^2 + y^2 - 1) - 1 / 3) ^2;
vars = {x, y};   x0 = {-1.25, 0.25};
```

Use SteepestDescent to find the local minimum:

In[6]:=
```
{opt, hist} = SteepestDescent[f, vars, x0];
```

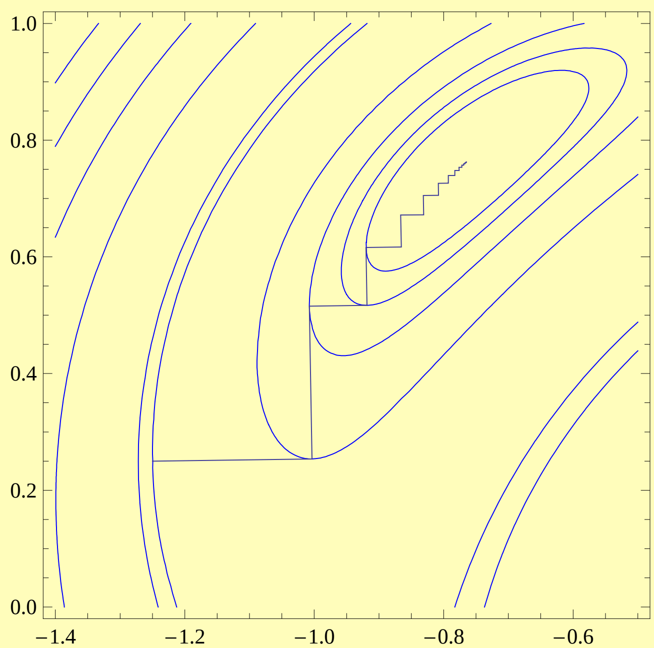
$$f \to (x + y)^2 + \left(-\frac{1}{3} + 2\left(-1 + x^2 + y^2\right)\right)^2$$

$$\nabla f \to \begin{matrix} -\frac{50\,x}{3} + 16\,x^3 + 2\,y + 16\,x\,y^2 \\ 2\,x - \frac{50\,y}{3} + 16\,x^2\,y + 16\,y^3 \end{matrix}$$

```
Optimum: {-0.764703, 0.762588} after 27 iterations
```

In[7]:=
```
xhist = Drop[Transpose[hist][[1]], 1];
PlotSearchPath[f, {x, -1.4, -0.5}, {y, 0, 1}, xhist]
```

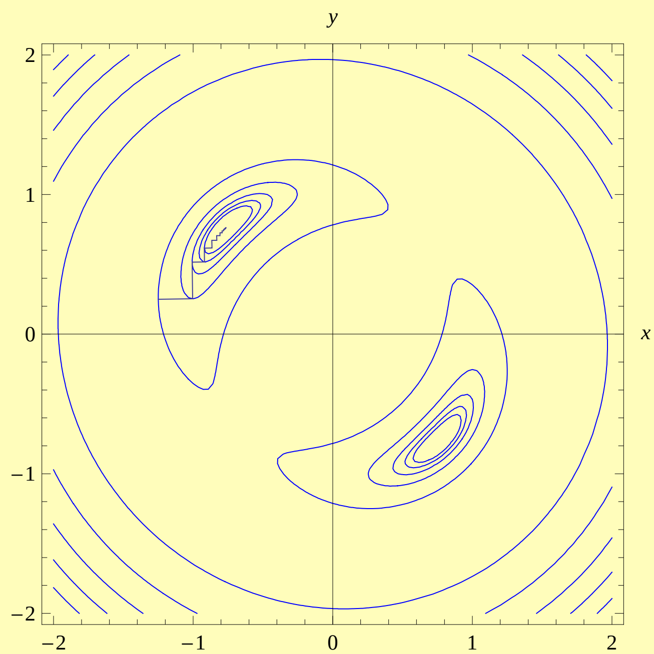
Out[8]=



However, if we plot the graph in a larger region, we can see that this problem may have another local minimum:

In[9]:=
```
PlotSearchPath[f, {x, -2, 2}, {y, -2, 2}, xhist]
```

Out[9]=



If we try a new starting point (1.25, -0.25), we can see that it converges to another local minimum:

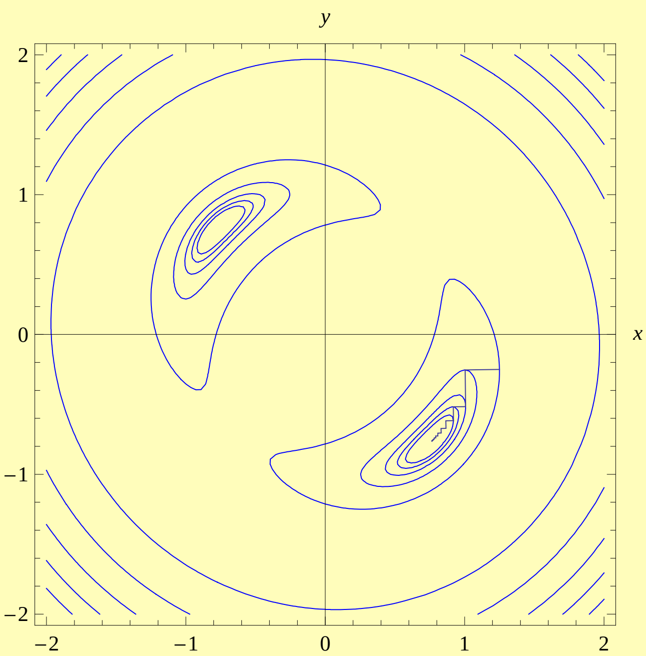
In[10]:=
```
x0 = {1.25, -0.25};
{opt, hist} = SteepestDescent[f, vars, x0];
xhist = Drop[Transpose[hist][[1]], 1];
PlotSearchPath[f, {x, -2, 2}, {y, -2, 2}, xhist]
```

$f \to (x+y)^2 + \left(-\frac{1}{3} + 2\left(-1 + x^2 + y^2\right)\right)^2$

$\nabla f \to \begin{pmatrix} -\frac{50\,x}{3} + 16\,x^3 + 2\,y + 16\,x\,y^2 \\ 2\,x - \frac{50\,y}{3} + 16\,x^2\,y + 16\,y^3 \end{pmatrix}$

```
Optimum: {0.764703, -0.762588} after 27 iterations
```

Out[13]=



■ Example 2

Use the conjugate gradient method to find (x,y) that minimized f(x,y)=$(x+y)^2 + \left(2\left(x^2+y^2-1\right)-1/3\right)^2$. The starting point is (-1.25, 0.25).

In[14]:=
```
f = (x + y) ^2 + (2 * (x^2 + y^2 - 1) - 1 / 3) ^2;
vars = {x, y};   x0 = {-1.25, 0.25};
{opt, hist} = ConjugateGradient[f, vars, x0];
xhist = Drop[Transpose[hist][[1]], 1];
PlotSearchPath[f, {x, -1.4, -0.5}, {y, 0, 1}, xhist]
```

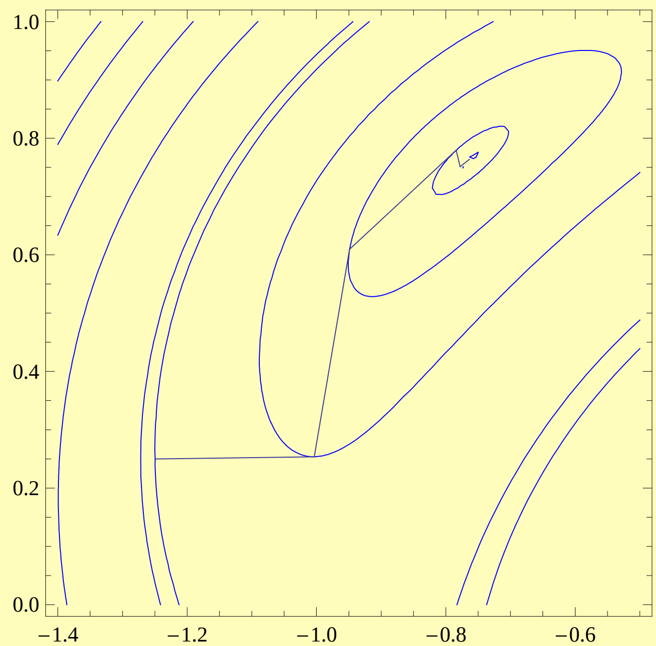
$$f \to (x+y)^2 + \left(-\frac{1}{3} + 2\left(-1 + x^2 + y^2\right)\right)^2$$

$$\nabla f \to \begin{pmatrix} -\frac{50x}{3} + 16x^3 + 2y + 16xy^2 \\ 2x - \frac{50y}{3} + 16x^2 y + 16y^3 \end{pmatrix}$$

```
Using the PolakRibiere method with Analytical line search
```

```
Optimum: {-0.763759, 0.763766} after 6 iterations
```

Out[18]=



Remark: comparing the iterations for convergence, we can see that for the example problem, SteepestDescent converges in 27 steps and ConjugateGradient converges in 6 iterations.

Using the Package provided by Mathematica

Before we start, let’s first clean the entire working space. This will remove the previously loaded package OptimizationToolbox.

In[19]:=

```
<< Utilities`CleanSlate`
CleanSlate[];
```

```
(CleanSlate) Contexts purged: {Global`}
```

```
(CleanSlate) Approximate kernel memory recovered: 472 Kb
```

Mathematica contains an Optimization package which provides solvers for UnconstrainedProblems. To load this package, use the following command:

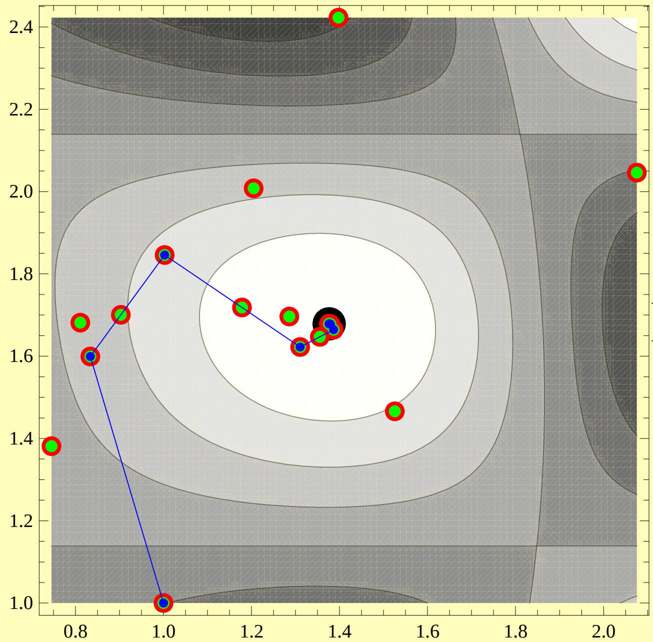
In[1]:=
```
Needs["Optimization`UnconstrainedProblems`"]
```

In[2]:=
```
f = Cos[x^2 - 3 * y] + Sin[x^2 + y^2];
FindMinimum[f, {{x, 1}, {y, 1}}, Method → "ConjugateGradient"]
```

Out[3]=
```
{-2., {x → 1.37638, y → 1.67868}}
```

In[4]:=
```
FindMinimumPlot[f, {{x, 1}, {y, 1}}, Method → "ConjugateGradient"]
```

Out[4]=
```
{{-2., {x → 1.37638, y → 1.67868}}, {Steps → 9, Function → 22, Gradient → 22},
```




```
}
```