

Lagrange Goes Skateboarding: How High Does He Go?

Introduction

OBJECTIVE: Learn how to maximize or minimize a function subject to a set of constraints using *Mathematica*.

How do you maximize or minimize a function subject to constraints? The skateboarder from the directional derivative project returns, and this time it's Lagrange himself. He will use his multipliers to determine precisely where along the figure-8 he reaches the high and low points of the surface. You will also investigate the role of the directional derivative.

■ Technology Guidelines

NOTE: If you have just finished a module, restart *Mathematica* or close the *Kernel* before executing a new module.

TO OPEN CELLS, put your cursor on the right cell bracket and double click.

TO STOP AN EXECUTION

Select the *Kernel* pull-down menu and click on *Abort Evaluation*.

ORDER OF EXECUTION

Execute cells in the order given. Do not skip any Input cells within a given notebook.

SAVING NOTEBOOKS

You can save anytime to any directory you choose, and it is wise to save often.

However, before you do your final save, it is a good idea to delete all your output by selecting the

Delete All Output selection under the *Kernel* pull-down menu.

EXPERIENCING MAJOR PROBLEMS

Save if appropriate, then shut down *Mathematica* and start it up again.

Part I: Revisiting the Skateboarder Problem

■ The Figure-8 Path

In the project on the directional derivative, you looked at the rate of change of the height of a skateboarder who was tracing out a figure-8 on different surfaces. For this project, you will use the method of Lagrange multipliers to determine the points on the figure-8 path where the skateboarder's height reaches a maximum or a minimum. Then, you will revisit the graph of the directional derivative to help you interpret the results.

This function is best analyzed in rectangular coordinates; therefore, we will define position, velocity, and unit tangent vectors parametrically in x and y . To show the direction of movement along the figure-8, we place arrows on the graph, which requires reading in a special graphics arrow package. Remember that you **must** read in the package before you execute a command within the package. The package should be read in only once during a worksession.

In[1]:=

```
<< Graphics`Arrow`
```

In[2]:=

```
Off[General::spell]
```

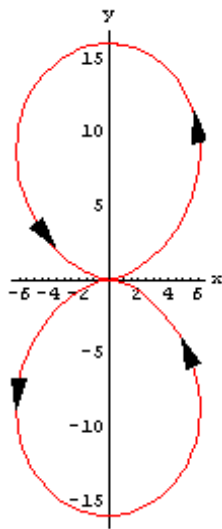
```
Off[General::spell1]
```

```
Clear[x, y, t, f]
```

```
x1[t_] = 16 Sin[t]^2 Cos[t];
```

```
y1[t_] = 16 Sin[t]^3;
```

```
pp8 = ParametricPlot[{x1[t], y1[t]}, {t, 0, 2  $\pi$ },
  AspectRatio -> Automatic, PlotStyle -> RGBColor,
  AxesLabel -> {"x", "y"},
  Epilog -> {Arrow[{x1[1], y1[1]}, {x1[1.1], y1[1.1]}],
    HeadScaling -> Absolute],
  Arrow[{x1[2.5], y1[2.5]}, {x1[2.6], y1[2.6]}],
    HeadScaling -> Absolute],
  Arrow[{x1[4], y1[4]}, {x1[4.1], y1[4.1]}],
    HeadScaling -> Absolute],
  Arrow[{x1[5.5], y1[5.5]}, {x1[5.6], y1[5.6]}],
    HeadScaling -> Absolute]};
```



■ The Ramp

First, we look at the three-dimensional plot and contour plot of the ramp, both showing the figure-8. Can you estimate the points of maximum and minimum height, even before you solve the problem analytically?

In[8]:=

```
Clear[x, y, t]
```

```
ramp[x_, y_] = .1 x + .2 y;
```

```

p8ramp = ParametricPlot3D[
  {x1[t], y1[t], ramp[x1[t], y1[t]], RGBColor
  {t, 0, 2  $\pi$ }, AxesLabel -> {x, y, z},
  DisplayFunction -> Identity];

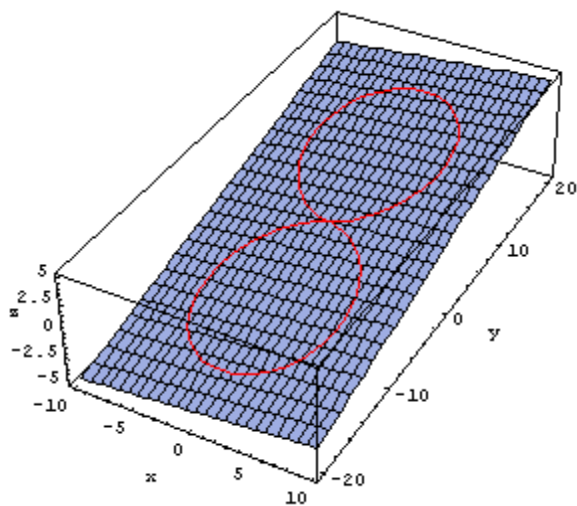
pramp = Plot3D[ramp[x, y], {x, -10, 10}, {y, -2
  DisplayFunction -> Identity];

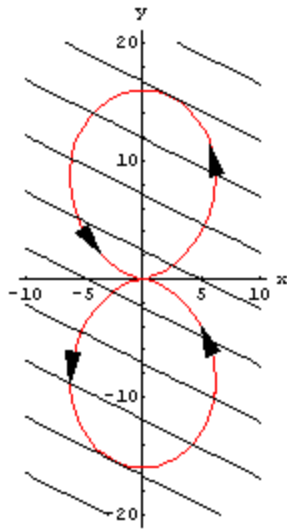
Show[p8ramp, pramp, DisplayFunction -> $Displ

ctramp = ContourPlot[ramp[x, y], {x, -10, 10},
  ContourShading -> False, DisplayFunction ->

Show[pp8, ctramp, DisplayFunction -> $Display

```





You Try It: Part I

Recall the bowl surface we explored in the directional derivative module. Now we will look at it again to estimate the locations where the skateboarder reaches the high and the low points.

In[15]:=

```
bowl[x_, y_] = .02 x2 + .05 y2;

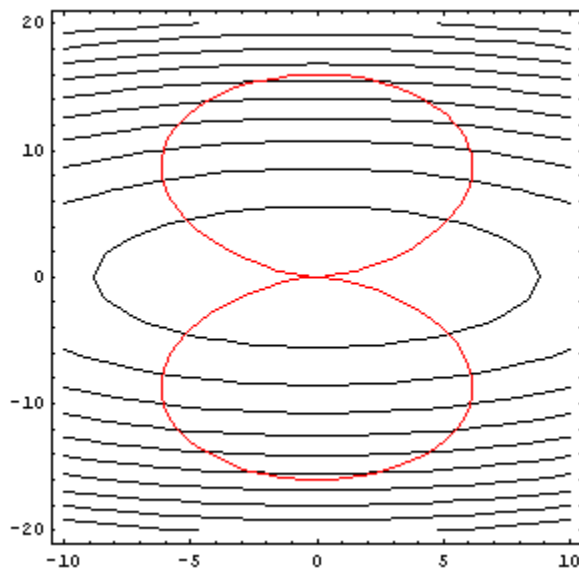
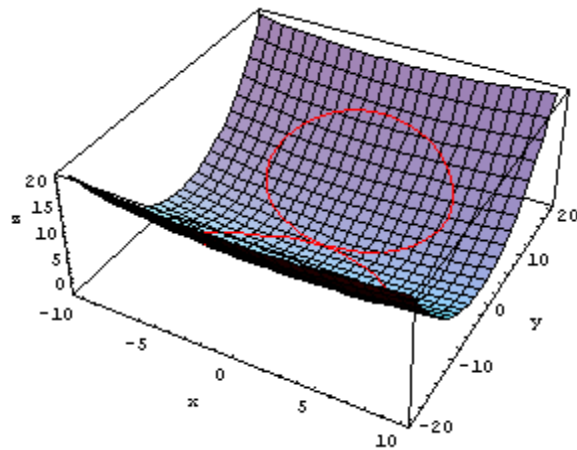
p8bowl = ParametricPlot3D[
  {x1[t], y1[t], bowl[x1[t], y1[t]], {RGBColor
  {t, 0, 2 π}, AxesLabel -> {x, y, z},
  DisplayFunction -> Identity];

pbowl = Plot3D[bowl[x, y] - .1, {x, -10, 10}, {}
  AxesLabel -> {x, y, z}, DisplayFunction -> Id

Show[pbowl, p8bowl, DisplayFunction -> $Displ

ctbowl = ContourPlot[bowl[x, y], {x, -10, 10},
  ContourShading -> False, DisplayFunction ->

Show[ctbowl, pp8, DisplayFunction -> $Display
```



As we noted in the directional derivative module, the surface is plotted just under where it should be so that the figure-8 shows up on top of it.

From the geometry of the problem, indicate where you will reach the high and low points along the figure-8 inside the bowl.

Part II: Applying the Method of Lagrange Multipliers

The figure-8 will be our constraint function, we will need to write it in non-parametric form. This is not a trivial step. You cannot eliminate the parameter t in many parametric expressions. For this figure-8, however, we can eliminate t , and the Cartesian form of the path can be written as $(x^2 + y^2)^3 - (4y)^4 = 0$.

The surfaces we consider are the ramp and the elliptical paraboloid (bowl). We will define the gradient in the usual fashion here.

In[21]:=

```
eight[x_, y_] = (x2 + y2)3 - (4 y)4;

gradient[f_] := {D[f, x], D[f, y]}
```

Next, we set the gradient of the surface function equal to a product of a parameter, λ , times the gradient of the constraint function, and we also require that the constraint be satisfied.

In[23]:=

```
Clear[x, y, λ]

solutionramp =
Solve[{gradient[ramp[x, y]] == λ gradient[eight[x, y]],
eight[x, y] == 0},
{x, y, λ}]
```

Out[24]=

```
{(λ → -0.00265464, x → -1.38986,
y → 0.440065), (λ → -1.15043 × 10-7,
x → -2.44067, y → -15.4168),
(λ → 1.15043 × 10-7, x → 2.44067,
y → 15.4168), (λ → 0.00265464,
x → 1.38986, y → -0.440065)}
```

The solutions you just found are summarized in the table below where we indicate the x , y , and z coordinates of the critical points.

In[25]:=

```
results =
Table[{solutionramp[[i, 2, 2]], solutionramp
ramp[solutionramp[[i, 2, 2]], solutionramp
{i, 1, Length[solutionramp]]};

display = PrependTo[results, {"x", "y", "z"}] .
```

Out[26]//TableForm=

x	y	z
-1.389863028620648`	0.44006509005973377`	-0.050973284850118025`
-2.440670677143082`	-15.416800904335444`	-3.3274272485813974`

2.440670677143082` 15.416800904335444` 3.3274272485813974`
 1.389863028620648` -0.44006509005973377` 0.050973284850118025`

Contrast the values you find here to the values for x and y we found in the directional derivative module that occur when the directional derivative is 0.

x	y	z
-1.38986303224366981`	0.440065091871244629`	-0.0509732848501180413`
-2.44066186681993224`	-15.416805309487529`	-3.32742724857949889`
2.44066983108720947`	15.4168013273632897`	3.32742724858137872`
1.38986303045863547`	-0.440065090978727546`	0.0509732848501180235`

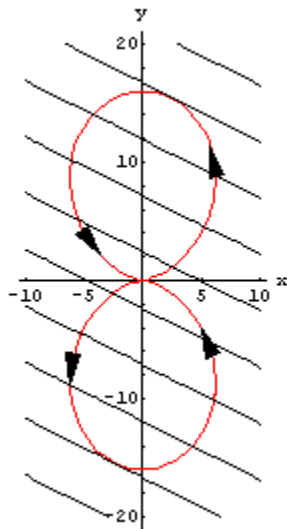
What is the connection?

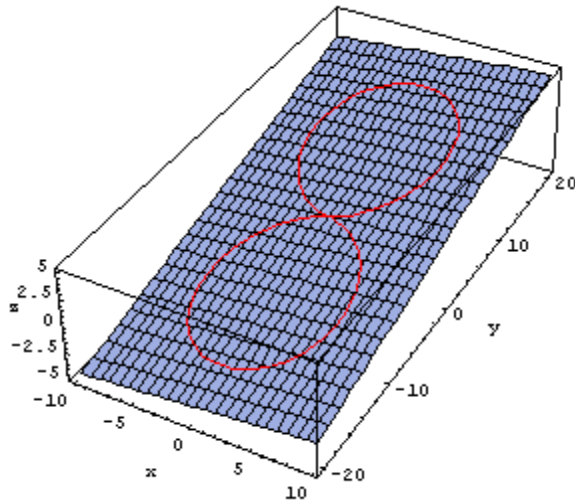
Using the results above, identify the points on the contour and surface plots below that represent a maximum or minimum.

In[27]:=

Show[pp8, ctramp, DisplayFunction -> \$Display

Show[p8ramp, pramp, DisplayFunction -> \$Displ





What do you notice about the slope of the contours compared to the slope of the figure-8 curve at the critical points? What do these slopes have to do with the gradients?

Sketch the gradient of the ramp and of the figure-8 at both the critical points and a few other points. What do you notice?

You Try It: Part II

Now, follow the same steps for the bowl-shaped surface. Find all the values of $\{x, y\}$ that make the gradient of the bowl equal to a multiple (λ) of the gradient of the figure-8 constraint. When you execute this command, you may get a warning that the **Solve** command may not have given all solutions, however, it gives all the real solutions that you need here.

In[29]:=

```
Clear[x, y, λ]

solutionbowl =
Solve[{gradient[bowl[x, y]] == λ gradient[figure8[x, y]],
      figure8[x, y] == 0}, {x, y, λ}]

Solve::svars :
Equations may not give solutions for
all "solve" variables. More...
```

Out[30]=

```
{x → 0., y → 0.},
{λ → 7.62939 × 10-7, x → 0., y → -16.},
{λ → 7.62939 × 10-7, x → 0., y → 16.},
{λ → 2.60711 × 10-6 + 0. i,
 x → 0., y → 0. - 4.74074 i},
{λ → 2.60711 × 10-6 + 0. i,
 x → 0., y → 0. + 4.74074 i}}
```

You can extract the solutions above and determine the z -coordinate on the bowl at each of these points.

In[31]:=

```
resultsbowl =
Table[{solutionbowl[[i, 2, 2]], solutionbowl
      bowl[solutionbowl[[i, 2, 2]], solutionbowl
      {i, 2, Length[solutionbowl]}];

PrependTo[resultsbowl, {0, 0, 0}];

displaybowl = PrependTo[resultsbowl, {"x", "y"}
TableForm
```

Out[33]//TableForm=

x	y	z
0	0	0
0.` -16.`		12.8`
0.` 16.`		12.8`
0.` 0.` -4.740740740740741` i		-1.1237311385459536` + 0.` i
0.` 0.` +4.740740740740741` i		-1.1237311385459536` + 0.` i

Again, let's contrast this to the results we found with the directional derivative.

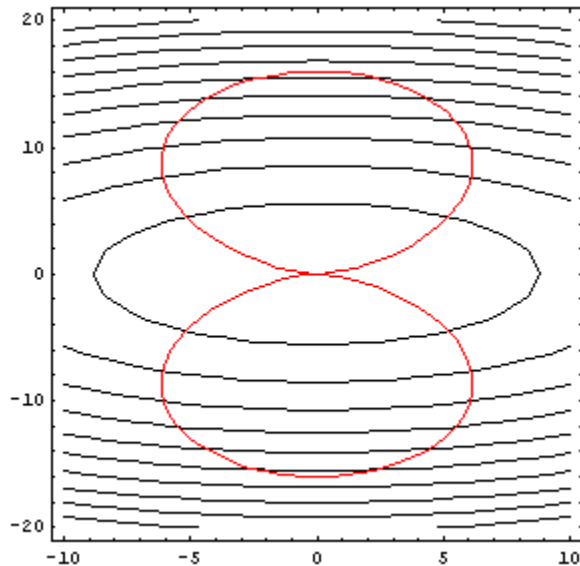
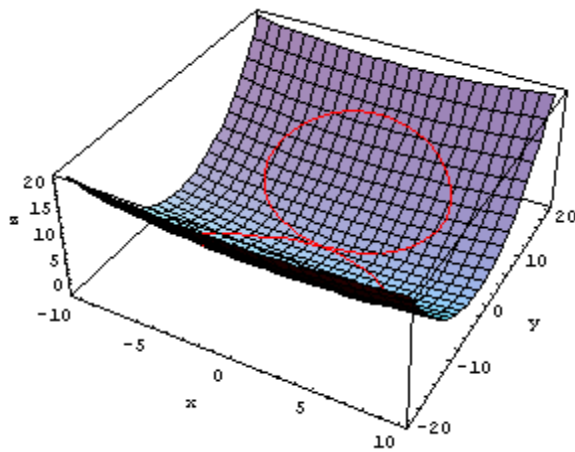
x	y	z
-0.0000104272856113163592`	8.41777135112082852`* ⁻⁹	2.17456924734303047`* ⁻¹²
-2.93905524917370541`* ⁻¹⁵	-16.00000000000000008`	12.80000000000000024`
1.57619399585851667`* ⁻⁶	15.9999999999997655`	12.799999999999676`

Identify the points on the contour plot and surface plot that resulted in a maximum or minimum.

In[34]:=

```
Show[pbowl, p8bowl, DisplayFunction -> $Displ
```

```
Show[cthowl, pp8, DisplayFunction -> $Display]
```



What do you notice about the slope of the contours compared to the slope of the figure-8 curve at the critical points? What do these slopes have to do with the gradients? Sketch the gradient of the ramp and of the figure-8 at both the critical points and a few other points. What do you notice?

You Try It: Take a Ride on a Roller Coaster

Select a surface of your own.

Check out the following roller coaster in the form of a cubic.

In[36]:=

```

parx[t_] = 16 Sin[t]^2 Cos[t];

pary[t_] = 16 Sin[t]^3;

path[x_, y_] := (x^2 + y^2)^3 - (4 y)^4

rc[x_, y_] = x^3

p8rc = ParametricPlot3D[
  {parx[t], pary[t], rc[parx[t], pary[t]], Rf
  {t, 0, 2 π}, AxesLabel -> {x, y, z},
  DisplayFunction -> Identity];

prc = Plot3D[rc[x, y] - 10, {x, -10, 10}, {y, -2
  AxesLabel -> {x, y, z}, PlotRange -> All,
  DisplayFunction -> Identity];

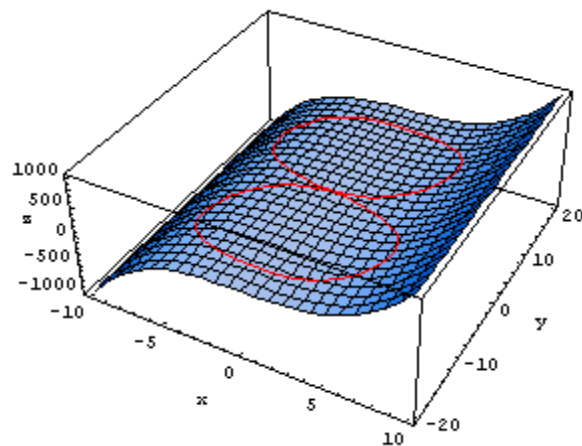
Show[prc, p8rc, DisplayFunction -> $DisplayFu

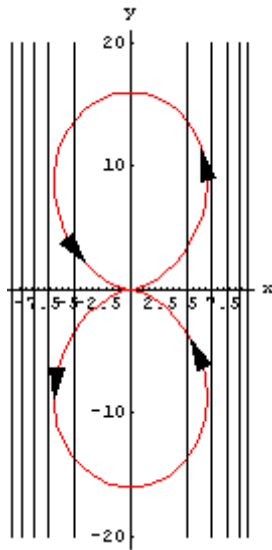
ctrc = ContourPlot[rc[x, y], {x, -10, 10}, {y,
  ContourShading -> False, DisplayFunction ->

Show[pp8, ctrc, DisplayFunction -> $DisplayFu

```

Out[39]=

 x^3 



In[45]:=

```
Clear[x, y]
```

```
solutionrc =
```

```
Solve[{gradient[rc[x, y]] == λ gradient[eight],
      eight[x, y] == 0}, {x, y, λ}] // N
```

```
Solve::svars :
Equations may not give solutions for
all "solve" variables. More...
```

Out[46]=

```
{{x → 0., y → 0.}, {x → 0., y → 0.},
 {x → 0., y → 0.}, {x → 0., y → 0.},
 {x → 0., y → 0.}, {λ → 0., x → 0., y → -16.},
 {λ → 0., x → 0., y → -16.},
 {λ → 0., x → 0., y → 16.},
 {λ → 0., x → 0., y → 16.},
 {λ → -0.000237861, x → -6.1584, y → -8.7093},
 {λ → -0.000237861, x → -6.1584, y → 8.7093},
 {λ → 0.000237861, x → 6.1584, y → -8.7093},
 {λ → 0.000237861, x → 6.1584, y → 8.7093}}
```

The z -coordinates on the roller coaster that correspond to these x - and y -coordinates are determined.

In[47]:=

```
rc[x, y] /. solutionrc
```

Out[47]=

```
{0., 0., 0., 0., 0., 0., 0., 0., 0.,  
-233.563, -233.563, 233.563, 233.563}
```

How many critical points do you have? Locate these points on your contour and surface plots to determine which represent maximums or minimums.

Created by [Mathematica](#) (April 2, 2005)

