# Lagrange Goes Skateboarding: How High Does He Go?

*Note: You may notice differences between this Maple worksheet and the equivalent Mathematica notebook. These differences were introduced to preserve the content of these modules and were necessary because of major functional differences between Maple and Mathematica.*

## Introduction

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

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 worksheet, **restart** *Maple* before executing a new worksheet.
TO OPEN SECTIONS,
Click on the **PLUS** sign at the left hand side of the screen *or* select **Expand All Sections** from the **View** drop down menu.

TO STOP AN EXECUTION
Click on **STOP** button from the toolbar.

ORDER OF EXECUTION
Execute commands in the order given. Do not skip any *Maple* Input lines within a given worksheet

Alternatively, you can execute the entire worksheet by selecting the **Execute Worksheet** command from the **Edit** drop down menu.

SAVING WORKSHEETS.

You can save anytime to any directory you choose, and it is wise to save often.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, and then shut down *Maple* 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.

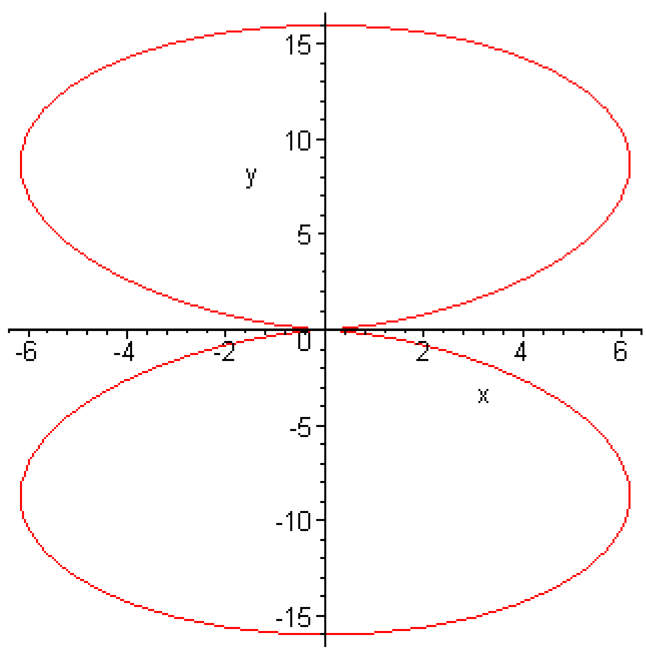
This function is best analyzed in rectangular coordinates; therefore, we will define position, velocity, and unit tangent vectors parametrically *x* and *y*. To show the direction of movement along the figure-8, we place arrows on the graph, which requires reading in a special graphics arrow package.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

```
> restart:
  with(plots):
  with(plottools):
```

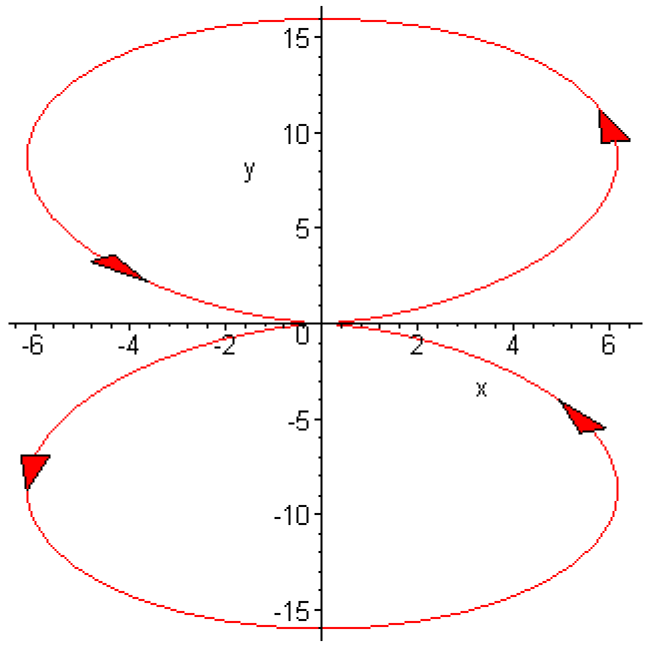
Warning, the name changecoords has been redefined

Warning, the assigned name arrow now has a global binding

```
> x1:=t->16*sin(t)^2*cos(t):
  y1:=t->16*sin(t)^3:
  pp8:=plot([x1(t),y1(t),t=0..2*Pi], labels=["x","y"]):
  print(pp8);
```



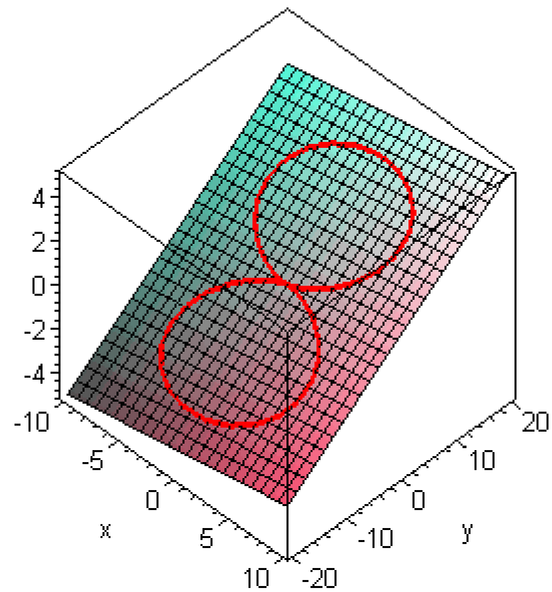

```
> A1:=plottools[arrow]([x1(1), y1(1)], [x1(1.1), y1(1.1)], .3,.6,1,color=red):
  A2:=plottools[arrow]([x1(2.5), y1(2.5)], [x1(2.6), y1(2.6)], .3,.6,1,color=red):
  A3:=plottools[arrow]([x1(4), y1(4)], [x1(4.1), y1(4.1)], .3,.6,1,color=red):
  A4:=plottools[arrow]([x1(5.5), y1(5.5)], [x1(5.6), y1(5.6)], .3,.6,1,color=red):
  plots[display]({pp8,A1,A2,A3,A4});
```

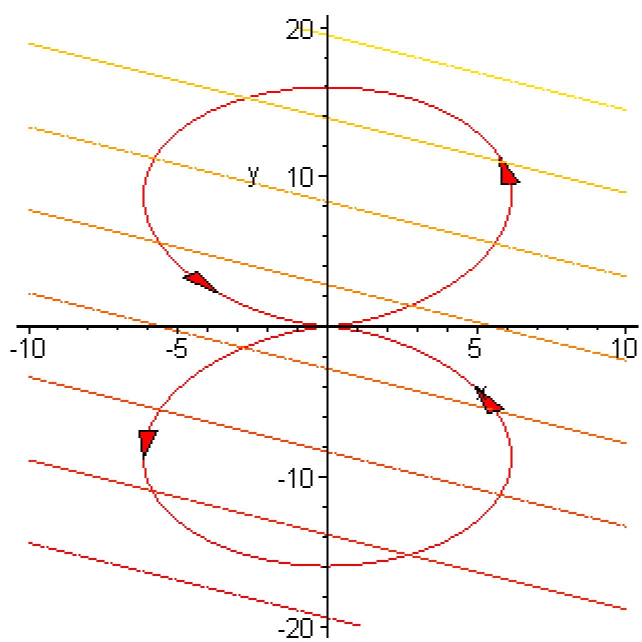



# The Ramp

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?

```
> unassign('x','y','t');
ramp:=(x,y)-> .1*x+.2*y:
p8ramp:=spacecurve([x1(t), y1(t), ramp(x1(t),y1(t)), t=0..2*Pi], color=red, thickness=
pramp:=plot3d(ramp(x,y), x=-10..10, y=-20..20, axes=BOXED, orientation=[-45,45]):
print(display({p8ramp,pramp}));
ctramp:=contourplot(ramp(x,y), x=-10..10, y=-20..20):
print(display({ctramp,pp8,A1,A2,A3,A4}));
```



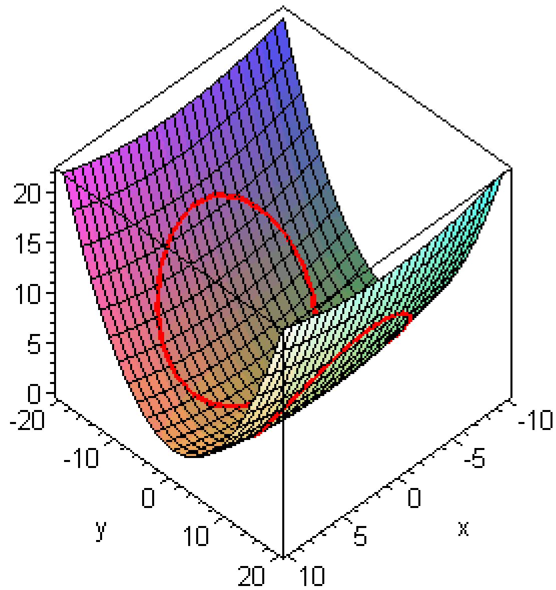



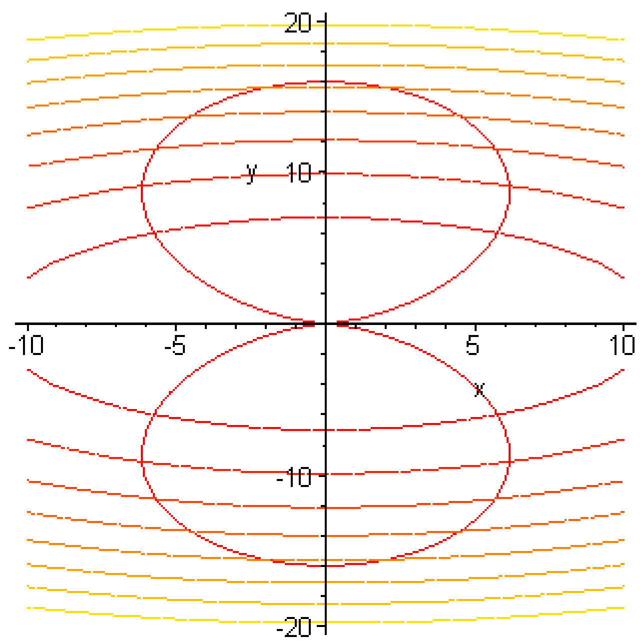

# You Try It: Part I

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.

```
> bowl:=(x,y)->.02*x^2+0.05*y^2;
  p8bowl:=spacecurve([x1(t), y1(t), bowl(x1(t),y1(t)), t=0..2*Pi], color=red, thickness=3):
  pbowl:=plot3d(bowl(x,y), x=-10..10, y=-20..20, axes=BOXED):
  print(display({p8bowl,pbowl}));
  ctbowl:=contourplot(bowl(x,y), x=-10..10, y=-20..20):
  print(display({ctbowl,pp8}));
```

$$bowl := (x, y) \rightarrow 0.02\,x^2 + 0.05\,y^2$$




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 which we will need to write in nonparametric 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.

```
> unassign('lambda');
  eight:=(x,y)->(x^2+y^2)^3-(4*y)^4:
  gradient1:=[diff(eight(x,y),x),diff(eight(x,y),y)];
```

$$gradient1 := \left[6\,(x^2+y^2)^2\,x,\,6\,(x^2+y^2)^2\,y-1024\,y^3\right]$$

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

```
> unassign('x','y','lambda');
  gradient2:=[diff(ramp(x,y),x),diff(ramp(x,y),y)]:
  sys:={gradient2[1]=lambda*gradient1[1], gradient2[2]=lambda*gradient1[2], eight(x,y)=0
  solutionramp:=solve(sys, {x,y,lambda});
```

$$solutionramp := \{x=-2.440670677, y=-15.41680090, \lambda=-0.1150433586\;10^{-6}\},$$
$$\{\lambda=0.002654636060, y=-0.4400650901, x=1.389863029\},$$
$$\{\lambda=-0.002654636060, y=0.4400650901, x=-1.389863029\},$$
$$\{y=15.41680090, x=2.440670677, \lambda=0.1150433586\;10^{-6}\}$$

```
> for i from 1 to 4 do
    assign(solutionramp[i]):
    sol[i]:=[x,y,lambda];
    unassign('x','y','lambda');
  od:
```

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

```
> unassign('z');
  m1:=[[`x`,`y`,`z`],seq([sol[i][1], sol[i][2], ramp(sol[i][1],sol[i][2])], i=1..4)]:
  matrix(m1);
```

| $x$ | $y$ | $z$ |
|---|---|---|
| -2.440670677 | -15.41680090 | -3.327427248 |
| 1.389863029 | -0.4400650901 | 0.05097328488 |
| -1.389863029 | 0.4400650901 | -0.05097328488 |
| 2.440670677 | 15.41680090 | 3.327427248 |

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.

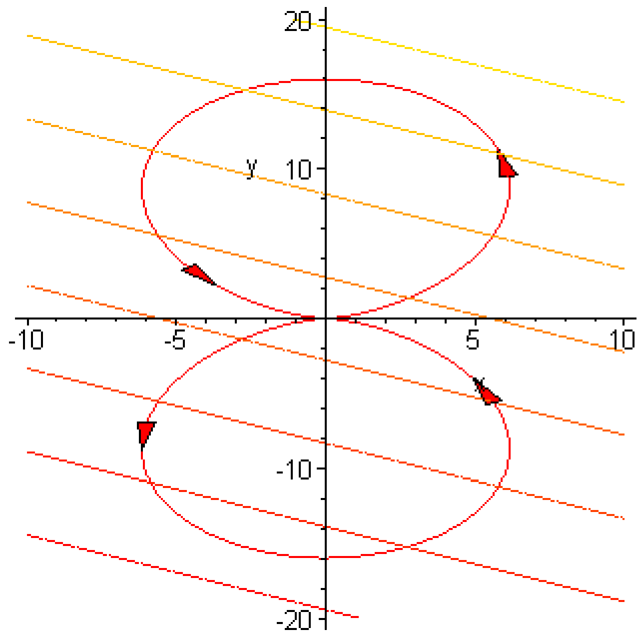
```
> matrix([[`x`,`y`,`z`],[2.44067,15.4168,3.32743],[-1.38986,0.440065,-0.0509733],[-2.44066,-15.4168,-3.32743],[1.38986,-.440065,0.0509733]]);
```

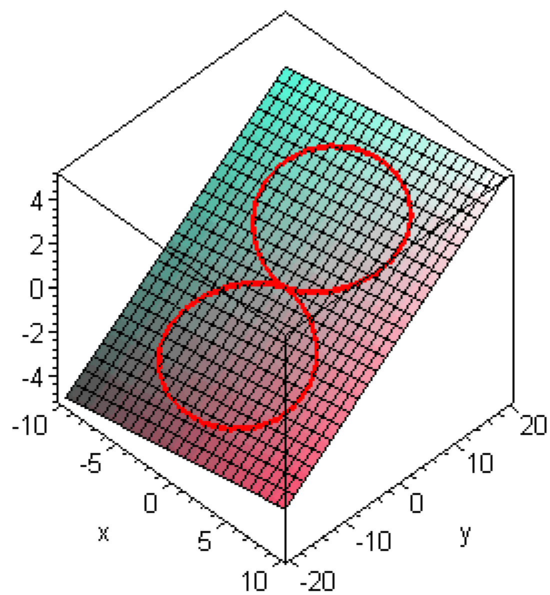
| $x$ | $y$ | $z$ |
|---|---|---|
| 2.44067 | 15.4168 | 3.32743 |
| -1.38986 | 0.440065 | -0.0509733 |
| -2.44066 | -15.4168 | -3.32743 |
| 1.38986 | -0.440065 | 0.0509733 |

What is the connection?

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

```
> print(display({pp8, ctramp, A1,A2,A3,A4}));
  print(display({p8ramp, pramp}));
```




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 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 ( $\lambda$ ) of the gradient of the figure-8 constraint.

```
> gradient3:=[diff(bowl(x,y),x),diff(bowl(x,y),y)]:
  sys:={gradient3[1]=lambda*gradient1[1], gradient3[2]=lambda*gradient1[2], eight(x,y)=0}
  solutionbowl:=[solve(sys, {x,y,lambda})];
```

$solutionbowl := [\{\lambda = \lambda, y = 0., x = 0.\}, \{x = 0., y = 16., \lambda = 0.7629394531\ 10^{-6}\},$
$\{x = 0., \lambda = 0.7629394531\ 10^{-6}, y = -16.\}, \{\lambda = 0.2607107162\ 10^{-5}, y = -4.740740741\ I, x = -8.$
$\{\lambda = 0.2607107162\ 10^{-5}, y = -4.740740741\ I, x = 8.546491912\},$
$\{\lambda = 0.2607107162\ 10^{-5}, y = 4.740740741\ I, x = -8.546491912\},$
$\{\lambda = 0.2607107162\ 10^{-5}, x = 8.546491912, y = 4.740740741\ I\}]$

Put the two real solutions from this part and the (0, 0) solution found earlier in a list, and then determine the *z*-coordinate on the bowl at each of those points.

```
> for i from 1 to 3 do
    assign(solutionbowl[i]):
    sol[i]:=[x,y,lambda];
    unassign('x','y','lambda');
  od;
```

$$sol_1 := [0., 0., \lambda]$$

$$sol_2 := [0., 16., 0.7629394531\ 10^{-6}]$$

$$sol_3 := [0., -16., 0.7629394531\ 10^{-6}]$$

```
> unassign('z');
  m2:=[[`x`,`y`,`z`],seq([sol[i][1], sol[i][2], bowl(sol[i][1],sol[i][2])], i=1..3)]:
  matrix(m2);
```

| | | |
|---|---|---|
| x | y | z |
| 0. | 0. | 0. |
| 0. | 16. | 12.80 |
| 0. | -16. | 12.80 |

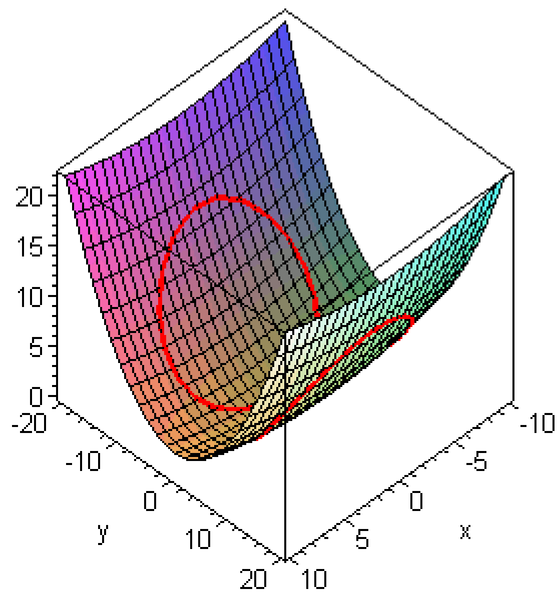
Again, let's contrast this to the results we found with the directional derivative.

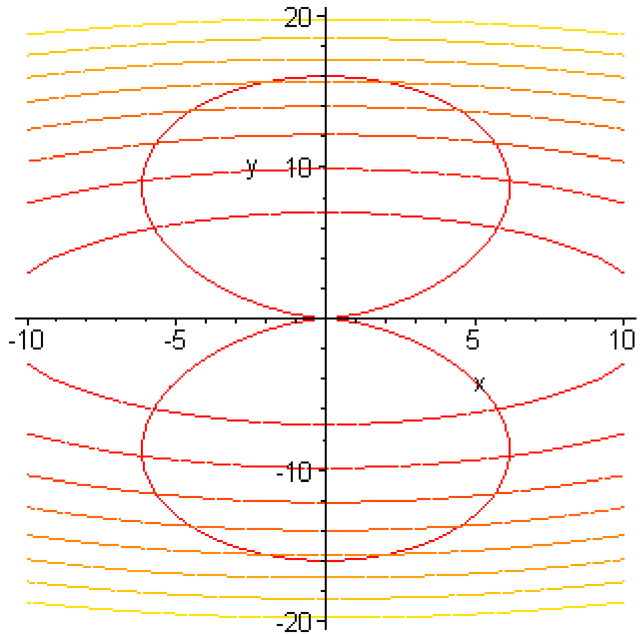
```
> matrix([[`x`,`y`,`z`],[1.57619*10^(-6),16,12.8],[-0.00000104273,8.4177*10^(-9), 2.17457*10^(-
  12)],[-2.93906*10^(-15), -16,12.8]]);
```

$$\begin{bmatrix} x & y & z \\ 0.1576190000\ 10^{-5} & 16 & 12.8 \\ -0.104273\ 10^{-5} & 0.8417700000\ 10^{-8} & 0.2174570000\ 10^{-11} \\ -0.2939060000\ 10^{-14} & -16 & 12.8 \end{bmatrix}$$

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

```
> print(display({pbowl,p8bowl}));
  print(display({ctbowl, pp8}));
```





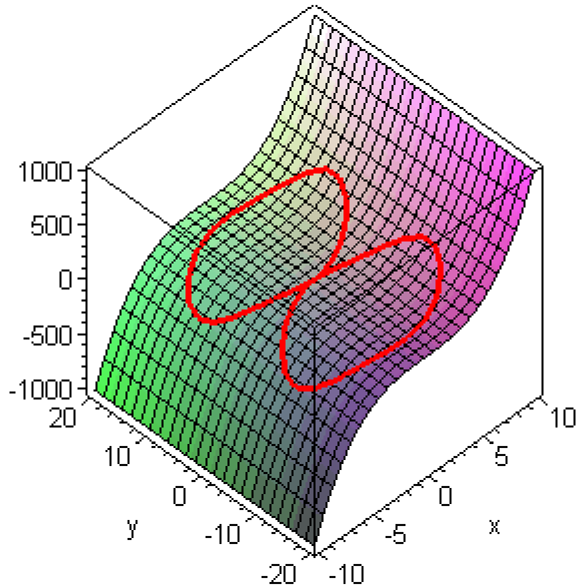

What do you notice about the slope of the contours compared to the slope of the figure-8 curve at the critical points? What do these slopes have to do with the gradients?

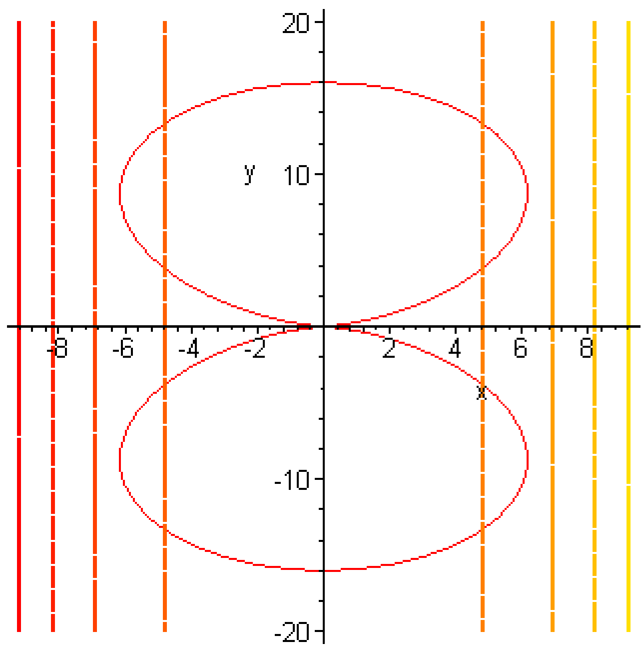
Sketch the gradient of the ramp and the figure-8 at both the critical points and a few other points. What do you notice?

# 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.

```
> 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:=spacecurve([parx(t), pary(t), rc(parx(t), pary(t)) , t=0..2*Pi], labels=["x","y","z"],
  thickness=3, color=red):
  prc:=plot3d(rc(x,y)-10, x=-10..10, y=-20..20, labels=["x","y","z"], orientation=[225,45],
  axes=BOXED):
  display({p8rc, prc});
  ctrc:=contourplot(rc(x,y), x=-10..10, y=-20..20, thickness=2):
  display({pp8, ctrc});
```




```
> unassign('x','y','lambda');
  gradientrc:=[diff(rc(x,y),x),diff(rc(x,y),y)]:
  sys:={gradientrc[1]=lambda*gradient1[1], gradientrc[2]=lambda*gradient1[2], eight(x,y)=
  solutionrc:=[solve(sys, {x,y,lambda})];
```

$$solutionrc := \Big[ \{\lambda = \lambda, y = 0, x = 0\}, \{x = 0, \lambda = 0, y = 16\}, \{x = 0, y = -16, \lambda = 0\}, \{ x = \frac{32}{3}\mathrm{RootOf}(3\_Z^2 - 1, label = \_L33), y = \frac{32}{3}\mathrm{RootOf}(3\_Z^2 - 2, label = \_L39), \lambda = \frac{27}{65536}\mathrm{RootOf}(3\_Z^2 - 1, label = \_L33)\} \Big]$$

Maple found three solutions in simplified form, but the fourth solution is not too readable. Because of this, the results that follow are just as unreadable. To see the result in simplified form, use the 5.1 version of Maple or the *Mathematica* version on the Website.

>