Parametric and Polar Equations with a Figure Skater

Chapter 13 Technology Application Project

1 Introduction

OBJECTIVE: Represent curves and analyze motion in parametric and polar form.

Parametric equations are very powerful, and the purpose of this module is to help you get used to the idea of expressing curves using parametric equations and analyzing motion in the plane using these parametric equations. Polar plots can also be expressed parametrically, and that can be translated easily into Cartesian coordinates.

1.1 Technology Guidelines

NOTE: If you have just finished a document, restart Maxima before executing a new document. This can be done by choosing "Restart" from the Maxima menu.

TO OPEN OR CLOSE CELLS
Click on the arrow at the top of the cell bracket.

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

ORDER OF EXECUTION
Execute commands in the order given. Do not skip any Maxima Input lines within a given document.

Alternatively, you can execute the entire worksheet by selecting the "Evaluate All Cells" command from the "Cell" drop down menu or simply press Ctrl-r.

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 Maxima and start it up again.

2 Part I: Parametric Equations of a Curve in Two-Dimentional Space

DEFINING THE FUNCTION

First, we define the x and y coordinates parametrically. Suppose that time, t, is the independent variable. Once x and y are defined, we can write the position vector r(t).

```
(%i1) reset()$
      kill(all)$
      load(draw)$
      ratprint:false$
```

```
(%i3) x(t) := cos(t)^3$
      y(t) := 1-exp(sin(t))$
      r(t) := [x(t), y(t)]$
```

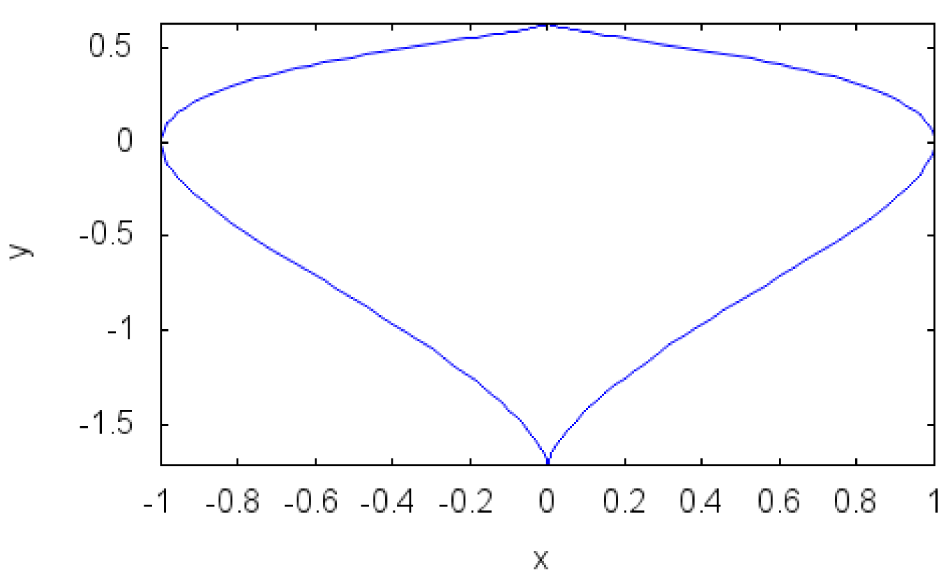
Now we plot the resulting curve in blue. In what direction are you moving on the curve as t increases?

```
(%i6) print("")$
      print("The position vector is: ")$
      print(r(t))$
      wxdraw2d(
         nticks=200,
         color=blue,
         parametric(x(t), y(t), t, 0, 2*%pi),
         xlabel="x",
         ylabel="y");
```

The position vector is:

$[\cos(t)^3, 1 - \%e^{\sin(t)}]$

(%t9)



(%o9)

TAKING THE VELOCITY AND ACCELERATION INTO ACCOUNT

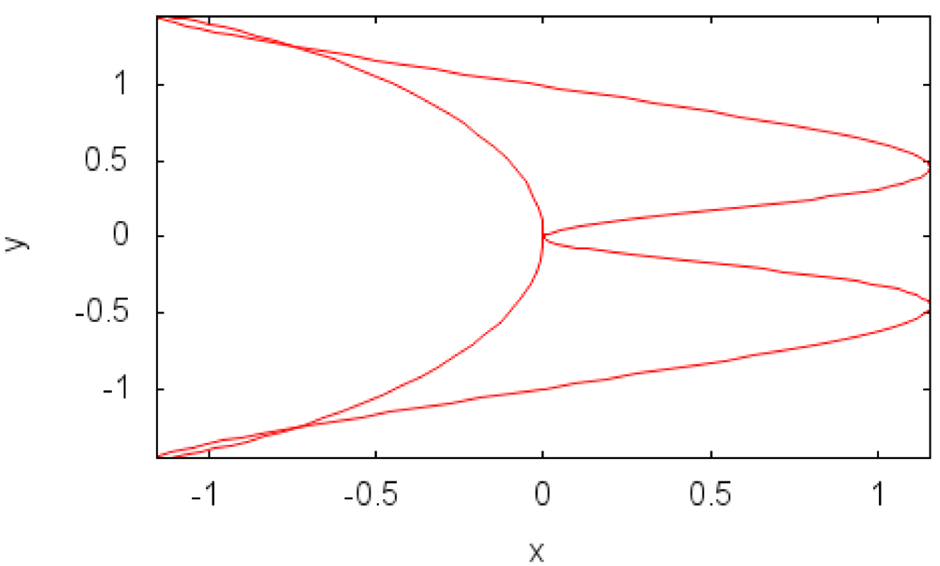
If you consider the parametric equation as a vector equation for the motion of a particle, the velocity vector is found by differentiating each component of the position vector. Similarly, the acceleration vector is found by differentiating the components of the position vector twice. We do this with Maxima and plot the velocity in red and the acceleration in green.

(%i10)
```
xd1: diff(x(t),t,1)$
yd1: diff(y(t),t,1)$
vel: parametric(xd1,yd1,t,0,2*%pi)$
print("")$
print("The velocity function is: ")$
print([xd1, yd1])$
wxdraw2d(
   nticks=200,
   color=red,
   vel,
   xlabel="x",
   ylabel="y");
```

The velocity function is:

$[-3\cos(t)^2\sin(t), -\%e^{\sin(t)}\cos(t)]$

(%t16)



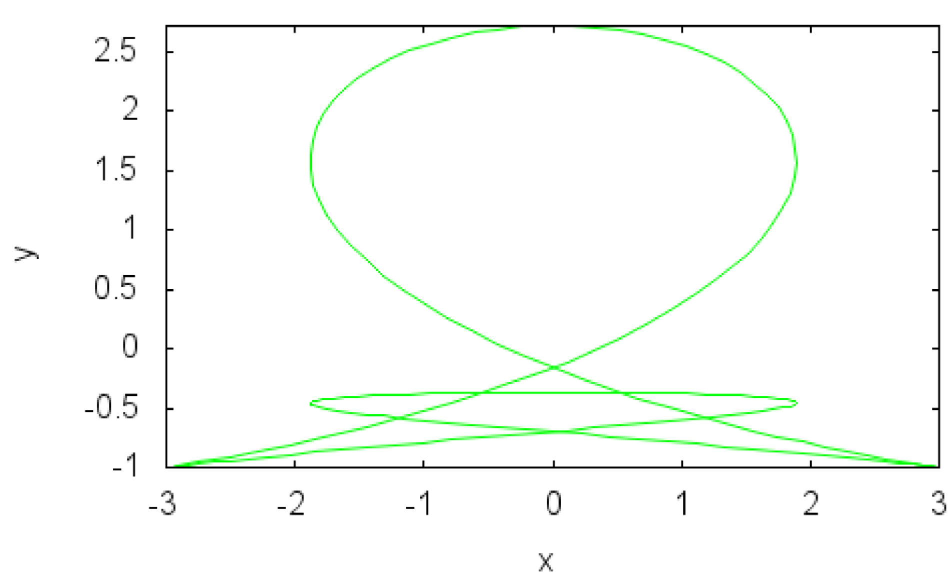
(%o16)

(%i17)
```
xd2: diff(x(t),t,2)$
yd2: diff(y(t),t,2)$
acc: parametric(xd2,yd2,t,0,2*%pi)$
print("")$
print("The acceleration function is: ")$
print( [xd2, yd2])$
wxdraw2d(
  nticks=200,
  color=green,
  acc,
  xlabel="x",
  ylabel="y");
```

The acceleration function is:

$$[6\cos(t)\sin(t)^2 - 3\cos(t)^3, \%e^{\sin(t)}\sin(t) - \%e^{\sin(t)}\cos(t)^2]$$

(%t23)



(%o23)

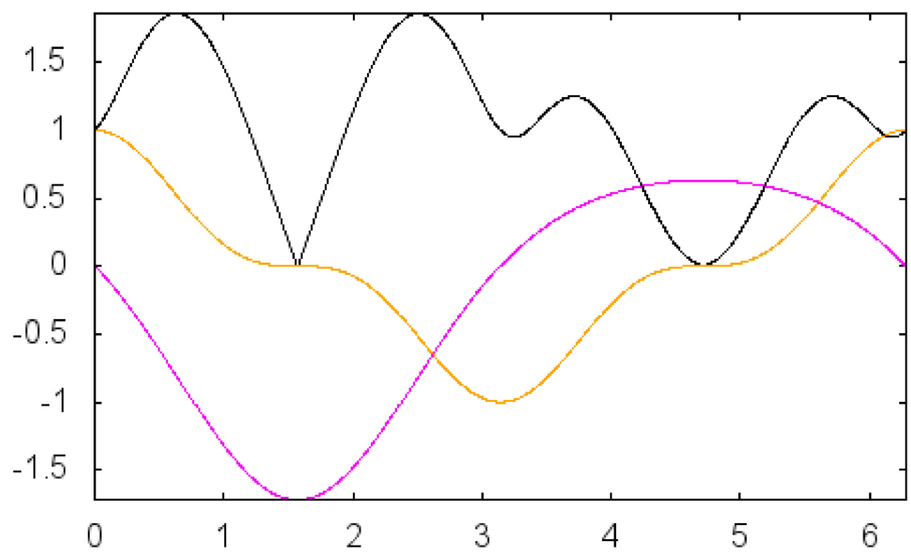
```
Note that the components of the velocity and acceleration
functions are more complicated than the components of the
postion function.

Let's look at the speed function and see what it tells us.
The following plot shows the speed in black, the x-coordinate
in orange, and the y-coordinate in magenta. Contrasting that to
your parametric plot, identify the places where the speed
function is 0.
```

(%i24)
```
speed(t) := sqrt((diff(r(t),t,1).diff(r(t),t)))$
print("The speed is: ")$
print(speed(t))$
wxdraw2d(
  nticks=200,
  color=black,
  explicit(speed(t), t, 0, 2*%pi),
  color=orange,
  explicit(x(t), t, 0, 2*%pi),
  color=magenta,
  explicit(y(t), t, 0, 2*%pi));
```

The speed is:

$$\sqrt{9\cos(t)^4\sin(t)^2+\%e^{2\sin(t)}\cos(t)^2}$$

(%t27)



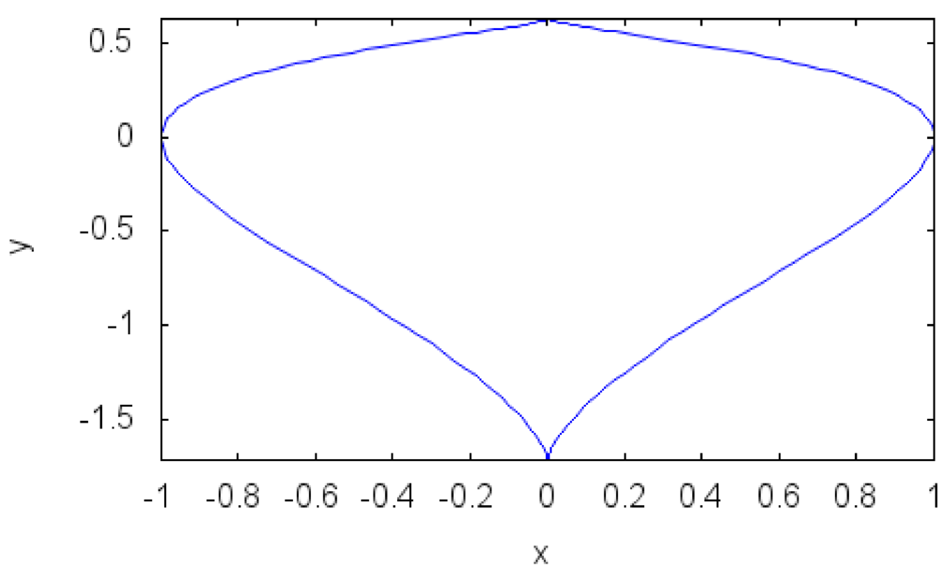
(%o27)

```
(%i28) print("")$
       print("The position vector is: ")$
       print(r(t))$
       wxdraw2d(
         nticks=200,
         color=blue,
         parametric(x(t), y(t), t, 0, 2*%pi),
         xlabel="x",
         ylabel="y");
```

The position vector is:

$[\cos(t)^3, 1-\%e^{\sin(t)}]$

(%t31)



(%o31)

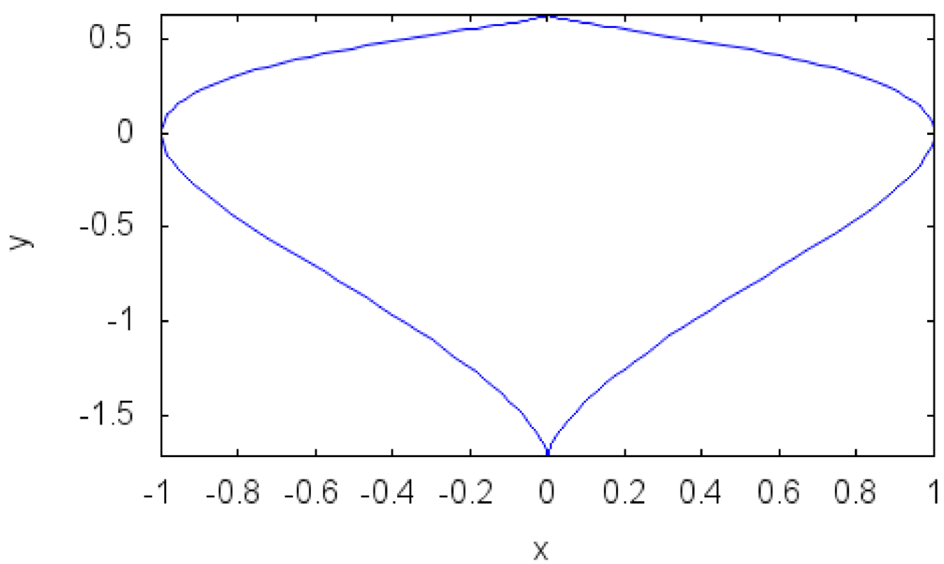
Identify the places on your path where the speed is 0 and the speed is a maximum.

COMPUTING THE DISTANCE TRAVELED ON A CURVED PATH

Suppose that you are walking along the path given above. The distance traveled can be found by integrating the speed function over a particular interval.

```
(%i32) distance: quad_qags(speed(t), t, 0, 2*%pi)[1]$
       wxdraw2d(
         nticks=200,
         color=blue,
         parametric(x(t), y(t), t, 0, 2*%pi),
         xlabel="x",
         ylabel="y");
```

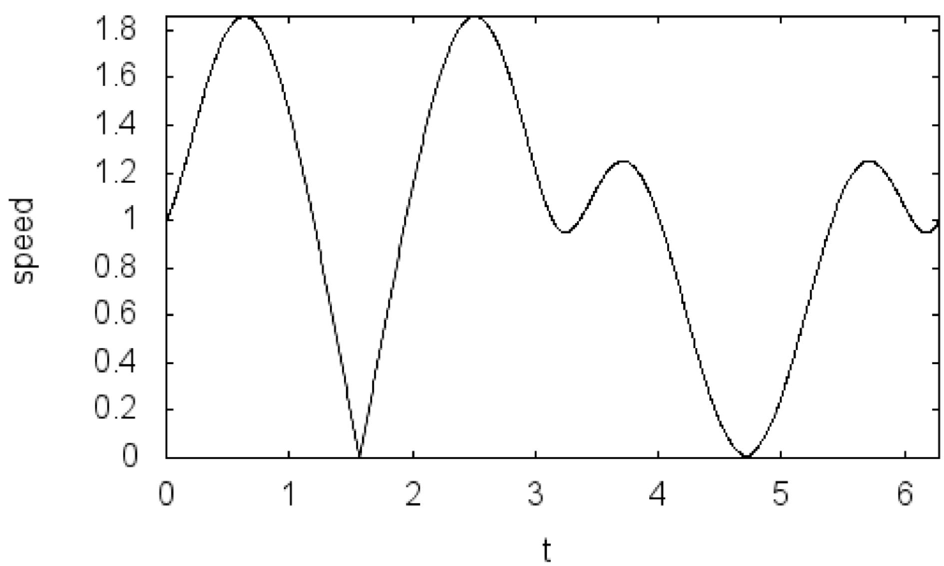
(%t33)



(%o33)

```
(%i34) wxdraw2d(
          nticks=200,
          color=black,
          explicit(speed(t), t, 0, 2*%pi),
          xlabel="t",
          ylabel="speed");
       print("")$
       print("The distance traveled around the closed path is ")$
       print(distance, " units")$
```

(%t34)



(%o34)

The distance traveled around the closed path is
6.530615383716405 *units*

Think of this answer as either the distance around the curve or as the area under the speed function over the interval t from 0 to 2*%pi.

2.1 You Try It: Part I

DEFINING A FUNCTION

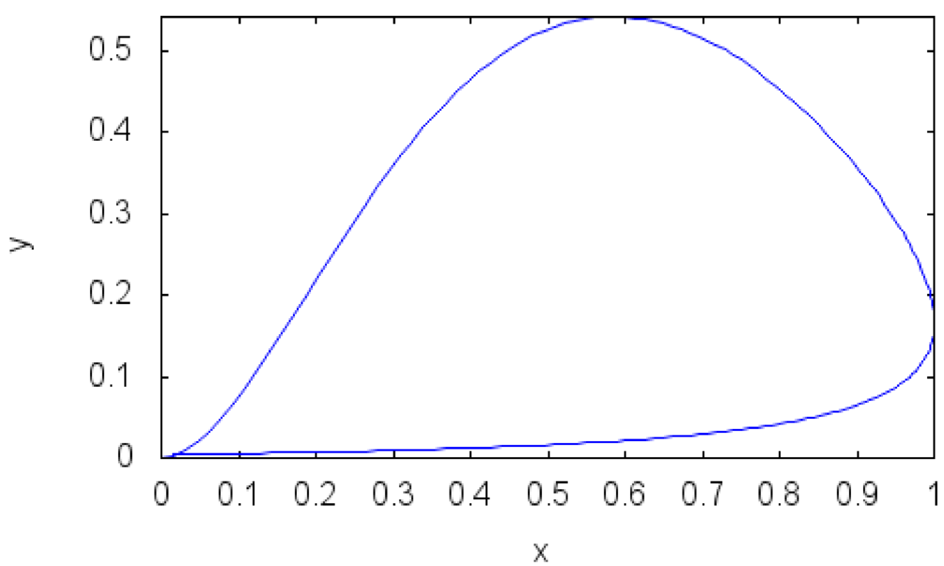
Select your own functions for x(t) and for y(t) in the cell below, and then execute the command below. You need not select a closed path, and you may wish to change the bounds for the parameter to something other than 0 to 10.

```
(%i38) x(t) := sin(t/3.2)$
       y(t) := exp(-t)*t^2$
       r(t) := [x(t),y(t)]$
       print("")$
       print("The position vector is: ")$
       print(r(t))$
       wxdraw2d(
         nticks=200,
         color=blue,
         parametric(x(t), y(t), t, 0, 10),
         xlabel="x",
         ylabel="y")$
```

The position vector is:

$[\sin(0.3125\, t), t^2\ \%e^{-t}]$

(%t44)



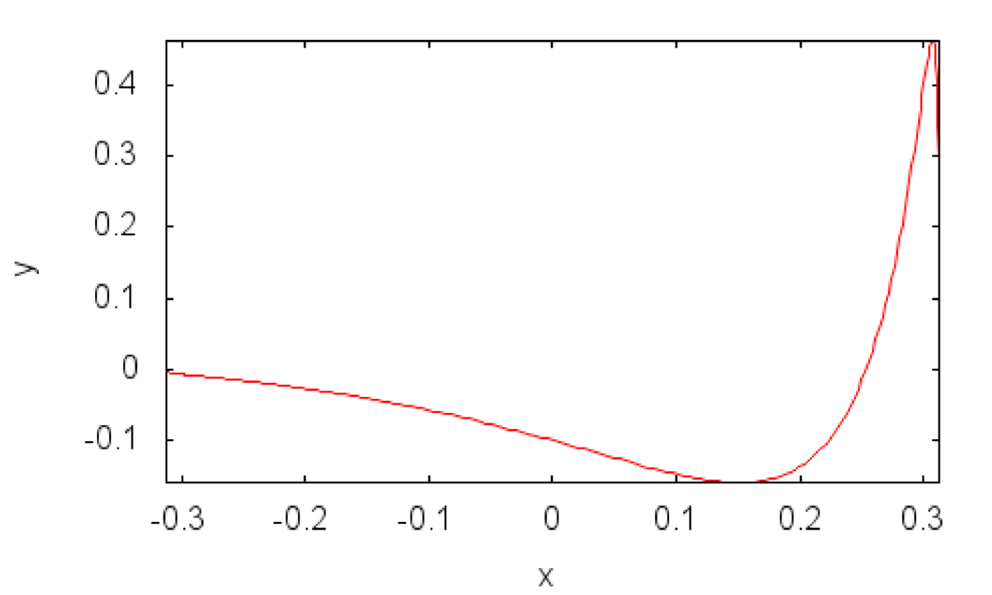
COMPUTING THE VELOCITY AND ACCELERATION VECTORS
AND ANALYZING THE SPEED

```
(%i45) xd1: diff(x(t),t,1)$
       yd1: diff(y(t),t,1)$
       vel: parametric(xd1,yd1,t,0,10)$
       print("")$
       print("The velocity function is: ")$
       print([xd1, yd1])$
       wxdraw2d(
         nticks=200,
         color=red,
         vel,
         xlabel="x",
         ylabel="y")$
```

The velocity function is:

$[0.3125\cos(0.3125\,t), 2\,t\,\%e^{-t} - t^2\,\%e^{-t}]$

(%t51)

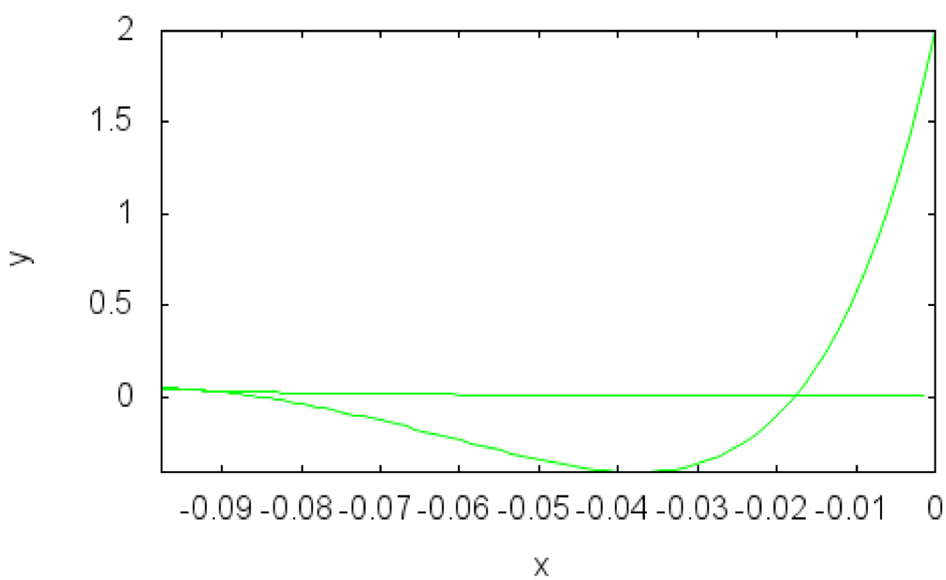


```
(%i52) xd2: diff(x(t),t,2)$
       yd2: diff(y(t),t,2)$
       acc: parametric(xd2,yd2,t,0,10)$
       print("")$
       print("The acceleration function is: ")$
       print( [xd2, yd2])$
       wxdraw2d(
          nticks=200,
          color=green,
          acc,
          xlabel="x",
          ylabel="y")$
```

The acceleration function is:

$[-0.09765625\ \sin(0.3125\ t), t^2\ \%e^{-t} - 4\ t\ \%e^{-t} + 2\ \%e^{-t}]$

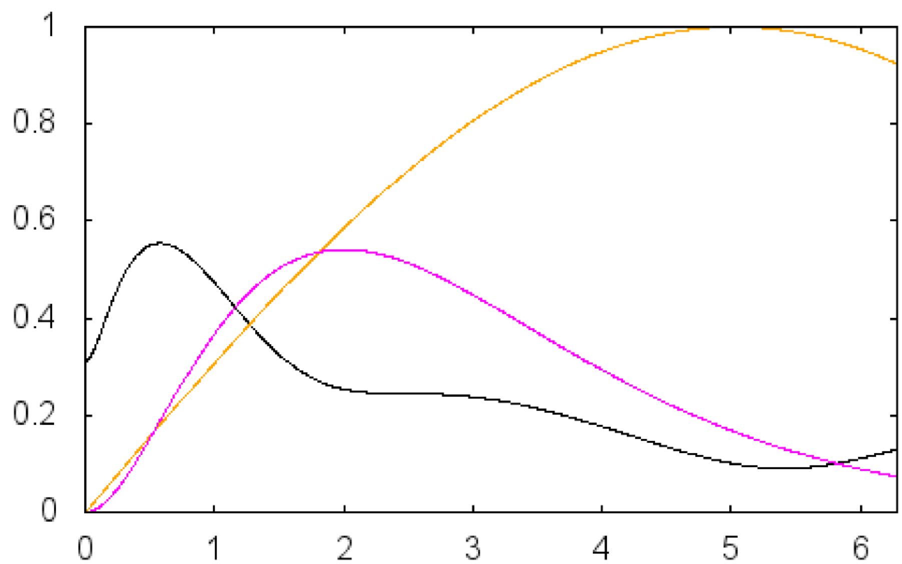
(%t58)



(%i59)
```
speed(t) := sqrt((diff(r(t),t,1).diff(r(t),t)))$
print("The speed is ", speed(t))$
wxdraw2d(
  nticks=200,
  color=black,
  explicit(speed(t), t, 0, 2*%pi),
  color=orange,
  explicit(x(t), t, 0, 2*%pi),
  color=magenta,
  explicit(y(t), t, 0, 2*%pi));
print("speed(t): black, x(t): orange, y(t): magenta")$
```

The speed is $\sqrt{0.09765625\cos(0.3125\,t)^2+(2\,t\,\%e^{-t}-t^2\,\%e^{-t})^2}$

(%t61)



(%o61)

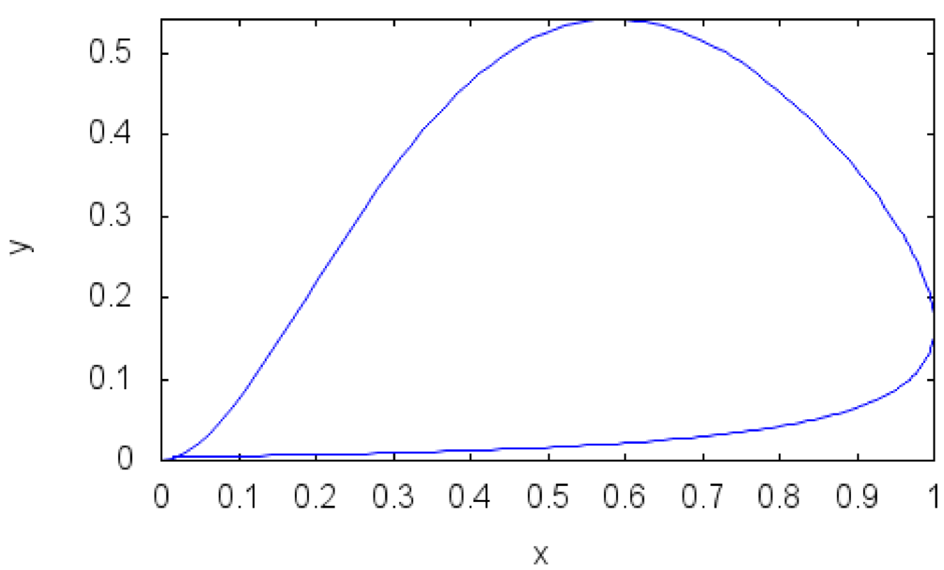
speed(t): black, x(t): orange, y(t): magenta

```
(%i63) print("")$
       print("The position vector is: ")$
       print(r(t))$
       wxdraw2d(
          nticks=200,
          color=blue,
          parametric(x(t), y(t), t, 0, 10),
          xlabel="x",
          ylabel="y")$
```

The position vector is:

$[\sin(0.3125\, t), t^2\, \%e^{-t}]$

(%t66)

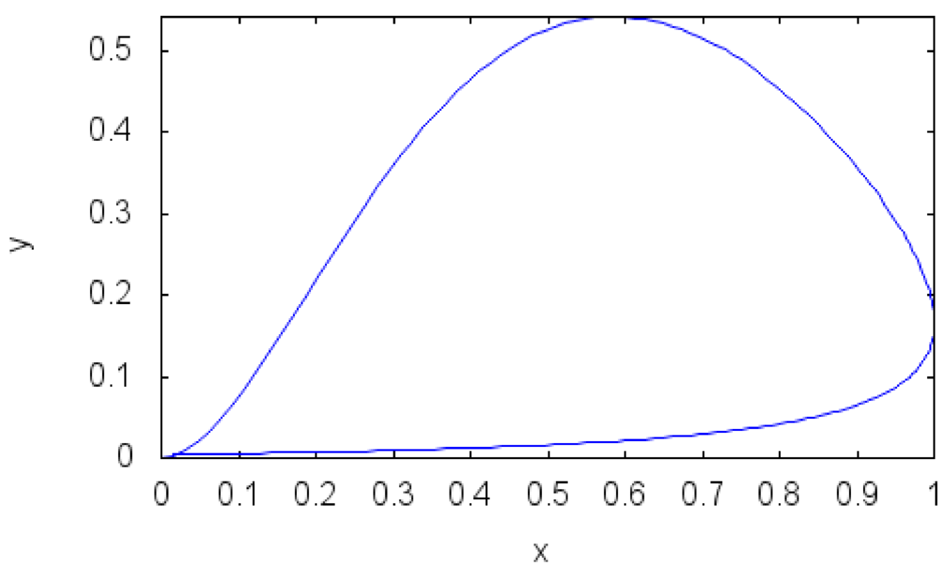



```
Identify the places on your path where the speed is 0 and the
speed is a maximum.
```

```
FIND THE DISTANCE ALONG THE CURVED PATH

Adjust the values of the parameter in the integration you
did so earlier.
```

(%i67)
```
distance: quad_qags(speed(t), t, 0, 10)[1]$
wxdraw2d(
  nticks=200,
  color=blue,
  parametric(x(t), y(t), t, 0, 10),
  xlabel="x",
  ylabel="y");
```

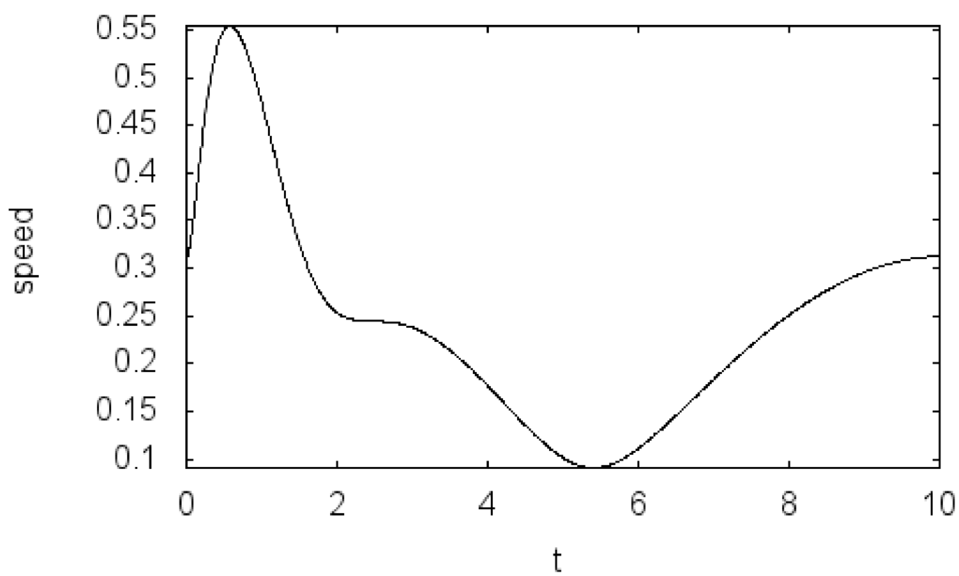
(%t68)



(%o68)

```
(%i69) wxdraw2d(
           nticks=200,
           color=black,
           explicit(speed(t), t, 0, 10),
           xlabel="t",
           ylabel="speed");
       print("")$
       print("The distance traveled around the closed path is ")$
       print(distance, " units")$
```

(%t69)



(%o69)

The distance traveled around the closed path is
2.462231543520764 *units*

3 Part II: A Figure Skater Tracing a Polar Plot

FOUR-PETAL PATTERN

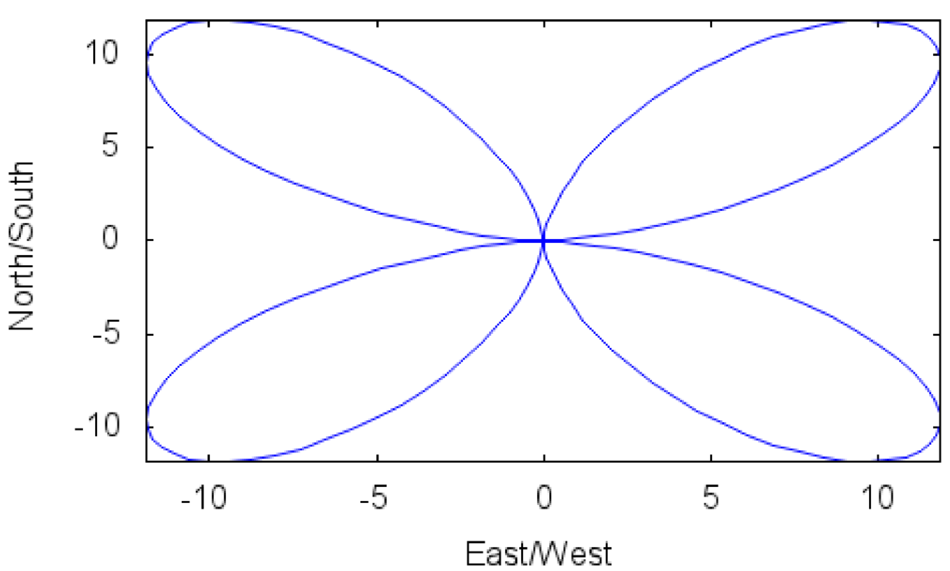
Think of a figure skater who is tracing out a four-petal flower on the ice. The first set of commands gives the parametric equations of the figure skater in terms of a path that would be traced in the x-y plane. It is easiest to start with the equations in polar form.

(%i73)
```
kill(all)$
load(draw)$
ratprint:false$
r(t) := 16*sin(t)^2;
theta(t) := t/2;
```

(%o3) $r(t) := 16\,\sin(t)^2$

(%o4) $\theta(t) := \frac{t}{2}$

(%i5)
```
p1: polar(r(2*theta),theta,0,2*%pi)$
wxdraw2d(
   nticks=200,
   color=blue,
   p1,
   xlabel="East/West",
   ylabel="North/South");
```

(%t6)


(%o6)

(%i7)
```
parx(t) := r(t) * cos(theta(t))$
pary(t) := r(t) * sin(theta(t))$
position: [parx(t),pary(t)]$
print("")$
print("The position vector is: ")$
print(position)$
velocity: diff(position,t,1)$

print("")$
print("The velocity vector is: ")$
print(velocity)$
speed(t) := sqrt(velocity.velocity)$
arr1: vector([parx(1), pary(1)], [parx(1.1)-parx(1),pary(1.1)-pary(1)])$
arr2: vector([parx(2.5), pary(2.5)], [parx(2.6)-parx(2.5),pary(2.6)-pary(2.5)])$
arr3: vector([parx(4), pary(4)], [parx(4.1)-parx(4),pary(4.1)-pary(4)])$
arr4: vector([parx(5.5), pary(5.5)], [parx(5.6)-parx(5.5),pary(5.6)-pary(5.5)])$
arr5: vector([parx(7), pary(7)], [parx(7.1)-parx(7),pary(7.1)-pary(7)])$
arr6: vector([parx(8.5), pary(8.5)], [parx(8.6)-parx(8.5),pary(8.6)-pary(8.5)])$
arr7: vector([parx(10), pary(10)], [parx(10.1)-parx(10),pary(10.1)-pary(10)])$
arr8: vector([parx(11.5), pary(11.5)], [parx(11.6)-parx(11.5),pary(11.6)-pary(11.5)])$
```

The position vector is:

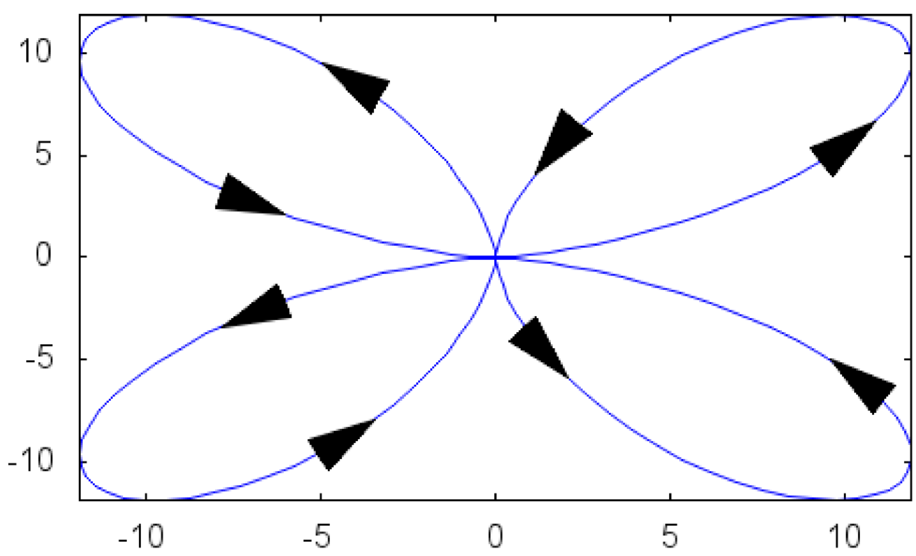
$$[16\cos\left(\frac{t}{2}\right)\sin(t)^2, 16\sin\left(\frac{t}{2}\right)\sin(t)^2]$$

The velocity vector is:

$$[32\cos\left(\frac{t}{2}\right)\cos(t)\sin(t) - 8\sin\left(\frac{t}{2}\right)\sin(t)^2, 8\cos\left(\frac{t}{2}\right)\sin(t)^2 + 32\sin\left(\frac{t}{2}\right)\cos(t)\sin(t)]$$

(%i26)
```
wxdraw2d(
        nticks=200,
        color=blue,
        p1,
        head_angle=15,
        color=black,
        arr1,arr2,arr3,arr4,arr5,arr6,arr7,arr8);
```

(%t26)



(%o26)

Can you tell in what direction the skater is moving right at the origin?

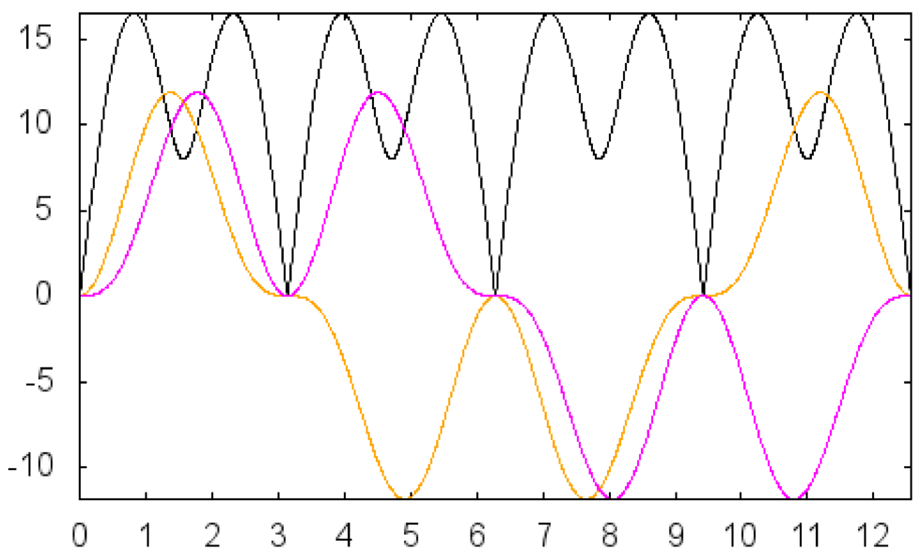
Let's look at the speed function and see what it tells us. The following plot shows the speed in black, the x-coordinate in orange, and the y-coordinate in violet. Contrasting that to your parametric plot, identify the places where the speed function is 0.

```
(%i27) tickst: [0,1,15]$
       ticksf: [-15,5,20]$
       print("")$
       print("The speed is: ")$
       print(speed(t))$
       wxdraw2d(
         nticks=200,
         color=black,
         explicit(speed(t), t, 0, 4*%pi),
         color=orange,
         explicit(parx(t), t, 0, 4*%pi),
         color=magenta,
         explicit(pary(t), t, 0, 4*%pi),
         xtics=tickst,
         ytics=ticksf);
       print("speed(t): black, x(t): orange, y(t): magenta")$
       print("")$
```

The speed is:

$$\sqrt{\left(32\cos\left(\frac{t}{2}\right)\cos(t)\sin(t)-8\sin\left(\frac{t}{2}\right)\sin(t)^2\right)^2+\left(8\cos\left(\frac{t}{2}\right)\sin(t)^2+32\sin\left(\frac{t}{2}\right)\cos(t)\sin(t)\right)^2}$$

(%t32)

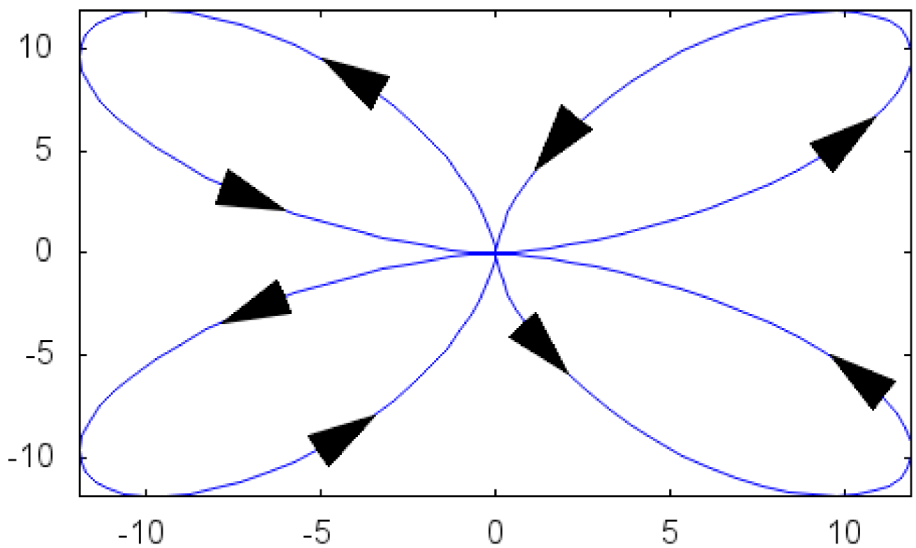


(%o32)

speed(t): black, x(t): orange, y(t): magenta

```
(%i35) wxdraw2d(
        nticks=200,
        color=blue,
        p1,
        head_angle=15,
        color=black,
        arr1,arr2,arr3,arr4,arr5,arr6,arr7,arr8);
```

(%t35)



(%o35)

Where are the places on your four-petal plot that the speed is 0?

VELOCTIY AND ACCELERATION: WHEN ARE THEY PERPENDICULAR?

Suppose we wish to determine for which values of t certain vectors describing the equations of motion are orthogonal. To do this, we will use the dot product, since perpendicular vectors yield a dot product of 0. Here we will examine when the velocity and acceleration vectors are perpendicular to one another. We begin by computing the dot product of the two vectors and we plot the resulting function of t to get an idea of when the dot product might be 0.

```
(%i36) acceleration: diff(velocity,t,1)$
       vdota: velocity.acceleration$
       print("")$
       print("velocity=")$
       print(velocity)$
       print("")$
       print("acceleration=")$
       print(acceleration)$
       print("")$
       print("velocity dotted into acceleration=")$
       print(vdota)$
       wxdraw2d(
          nticks=200,
          color=black,
          explicit(vdota,t,0,4*%pi),
          xlabel="t",
          ylabel="vdota");
```

velocity=

$$[32\cos\left(\frac{t}{2}\right)\cos(t)\sin(t)-8\sin\left(\frac{t}{2}\right)\sin(t)^2, 8\cos\left(\frac{t}{2}\right)\sin(t)^2+32\sin\left(\frac{t}{2}\right)\cos(t)\sin(t)]$$

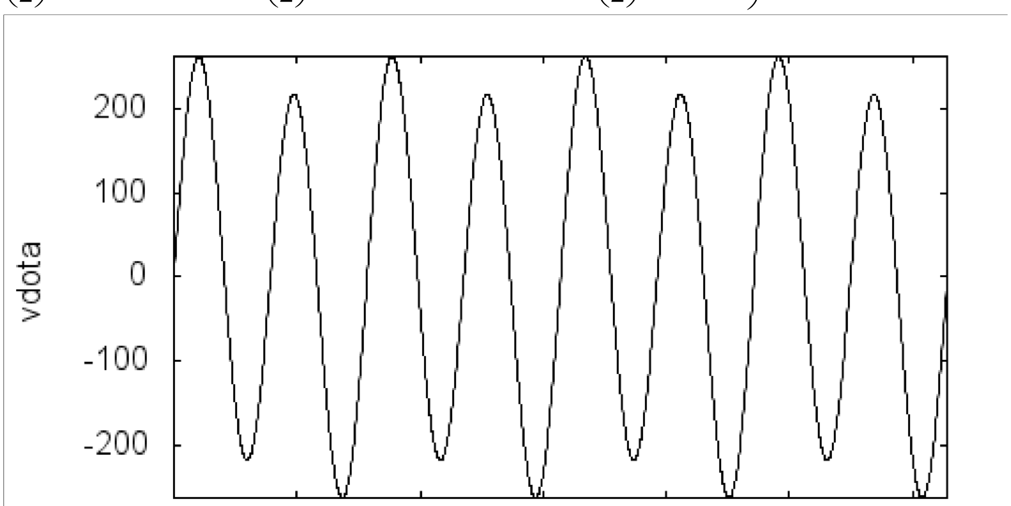
acceleration=

$$[-36\cos\left(\frac{t}{2}\right)\sin(t)^2-32\sin\left(\frac{t}{2}\right)\cos(t)\sin(t)+32\cos\left(\frac{t}{2}\right)\cos(t)^2, -36\sin\left(\frac{t}{2}\right)\sin(t)^2+32\cos\left(\frac{t}{2}\right)\cos(t)\sin(t)+32\sin\left(\frac{t}{2}\right)\cos(t)^2]$$

velocity dotted into acceleration=

$$\left(-36\cos\left(\frac{t}{2}\right)\sin(t)^2-32\sin\left(\frac{t}{2}\right)\cos(t)\sin(t)+32\cos\left(\frac{t}{2}\right)\cos(t)^2\right)\left(32\cos\left(\frac{t}{2}\right)\cos(t)\sin(t)-8\sin\left(\frac{t}{2}\right)\sin(t)^2\right)+\left(8\cos\left(\frac{t}{2}\right)\sin(t)^2+32\sin\left(\frac{t}{2}\right)\cos(t)\sin(t)\right)\left(-36\sin\left(\frac{t}{2}\right)\sin(t)^2+32\cos\left(\frac{t}{2}\right)\cos(t)\sin(t)+32\sin\left(\frac{t}{2}\right)\cos(t)^2\right)$$

(%t47)



As you can see from the graph, there are many times when the velocity and acceleration vectors are perpendicular to each other.

The following commands find some of the places where the velocity and acceleration vectors are perpendicular. Notice that we use seed values in solve that seem close to some of the places where our function crosses the horizontal axis.

```
(%i48) sol1: []$
       for i: 0.785 thru 12 step 0.785 do block(
         temp: find_root(vdota, t, i-0.3925, i+0.3925),
         sol1: append(sol1, [temp]))$
       sol1;
```

(%o50) [0.818756237602561,1.570796326794897,2.322836415987232,3.141592653589793, 3.960348891192354,4.71238898038469,5.464429069577026,6.283185307179586, 7.101941544782147,7.853981633974483,8.606021723166819,9.42477796076938, 10.24353419837194,10.99557428756428,11.74761437675661]

Now we evaluate x and y at the values of t we have found.

```
(%i51) sol2: makelist([parx(sol1[i]),pary(sol1[i])],i,1,length(sol1));
```

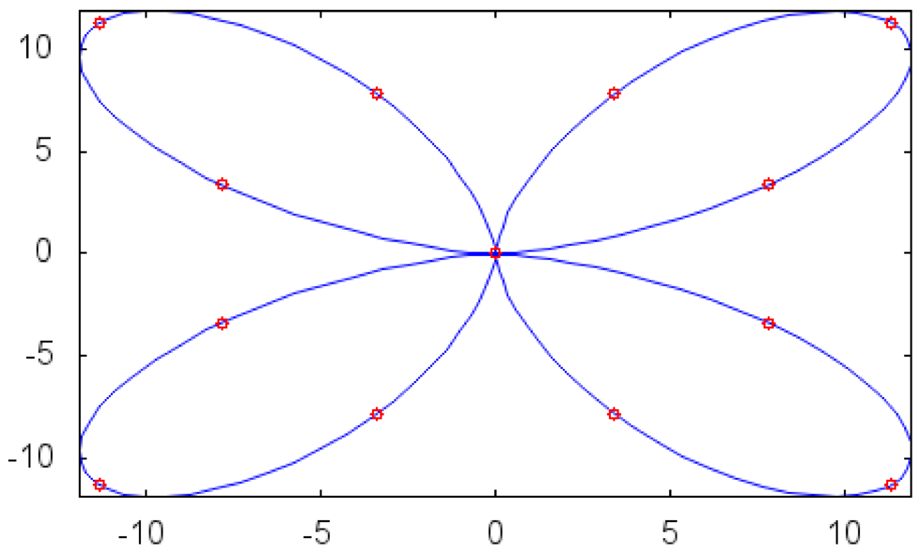
(%o51) [[7.82821148249957,3.396598705034156],[11.31370849898476,11.31370849898476],[3.396598705034154,7.828211482499566],[$1.4691952388169937\ 10^{-47}$,$2.39945715491542\ 10^{-31}$],[-3.396598705034156,7.82821148249957],[-11.31370849898476,11.31370849898476],[- 7.828211482499563,3.39659870503415],[$-9.59782861966168\ 10^{-31}$,$1.175356191053595\ 10^{-46}$] ,[-7.82821148249957,-3.396598705034155],[-11.31370849898476,-11.31370849898476],[- 3.396598705034152,-7.828211482499565],[$-3.966827144805883\ 10^{-46}$,$-2.159511439423878\ 10^{-30}$],[3.396598705034163,-7.828211482499579],[11.31370849898476,-11.31370849898476] ,[7.828211482499566,-3.396598705034153]]

We can now see where those points are relative to our petals.

(%i52) psol: points(sol2)$

```
wxdraw2d(
  nticks=200,
  color=blue,
  p1,
  color=red,
  point_type=circle,
  psol);
```

(%t53)



(%o53)

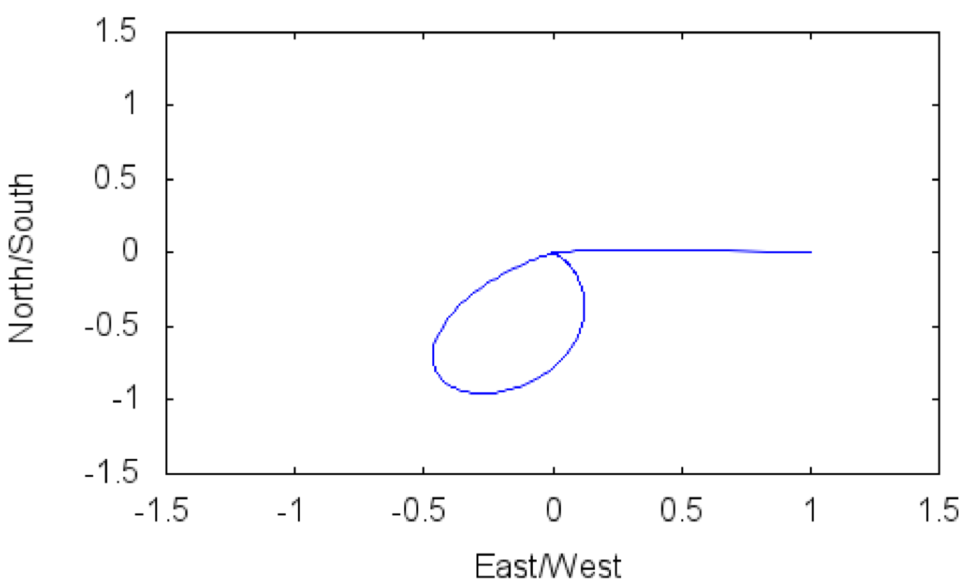
The dots show where the velocity and acceleration vectors are perpendicular to each other. Can you describe what is happening to the figure skater at those points?

3.1 You Try It: Part II

Try your own functions for r(t) and theta(t). Remember to solve for t as a function of theta before you attempt to do a polar plot in the form of r as a function of theta. Replace the terms in r(t) and theta(t).

(%i54)
```
kill(t,x,y)$
r(t) := cos(t)^3$
theta(t) := t^2/8$
pp1: polar(cos(sqrt(8*theta))^3,theta,0,3)$
wxdraw2d(
   nticks=200,
   color=blue,
   pp1,
   xlabel="East/West",
   ylabel="North/South",
   xrange=[-1.5,1.5],
   yrange=[-1.5,1.5]);
```

(%t58)

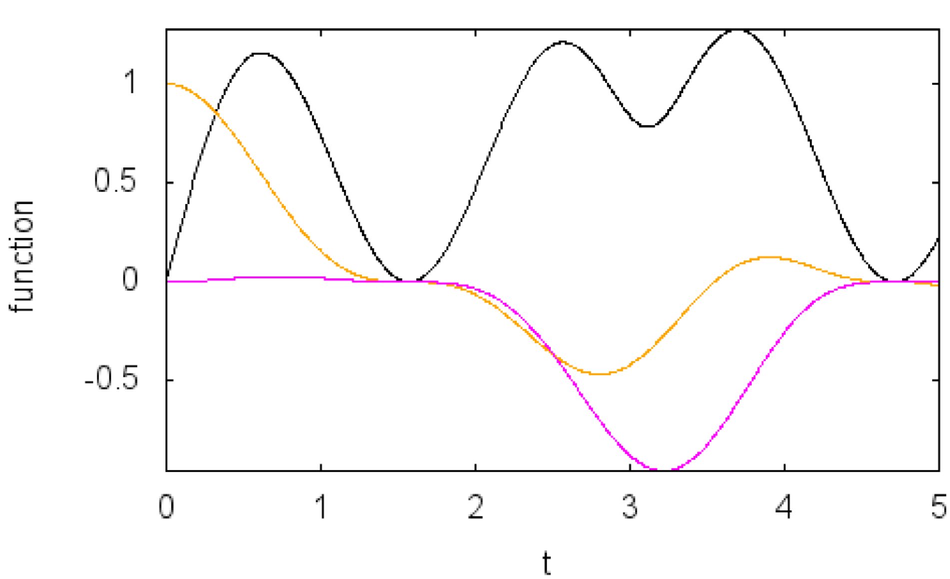


(%o58)

```
(%i59) parx(t) := r(t)*cos(theta(t))$
       pary(t) := r(t)*sin(theta(t))$
       position: [parx(t), pary(t)]$
       velocity: diff(position, t, 1)$
       speed: sqrt(velocity. velocity)$
       print("Position Vector: ", position)$
       print("Speed: ", speed)$
       wxdraw2d(
          nticks=200,
          color=black,
          explicit(speed,t,0,5),
          color=orange,
          explicit(parx(t),t,0,5),
          color=magenta,
          explicit(pary(t),t,0,5),
          xlabel="t",
          ylabel="function")$
       print("speed: black")$
       print("x-coordinate of the path of motion: orange")$
       print("y-coordinate of the path of motion: magenta")$
       print("")$
```

Position Vector: $[\cos(t)^3 \cos\left(\frac{t^2}{8}\right), \cos(t)^3 \sin\left(\frac{t^2}{8}\right)]$

Speed: $\sqrt{\left(\frac{t \cos(t)^3 \cos\left(\frac{t^2}{8}\right)}{4} - 3\cos(t)^2 \sin(t) \sin\left(\frac{t^2}{8}\right)\right)^2 + \left(-\frac{t \cos(t)^3 \sin\left(\frac{t^2}{8}\right)}{4} - 3\cos(t)^2 \sin(t) \cos\left(\frac{t^2}{8}\right)\right)^2}$

(%t66) 

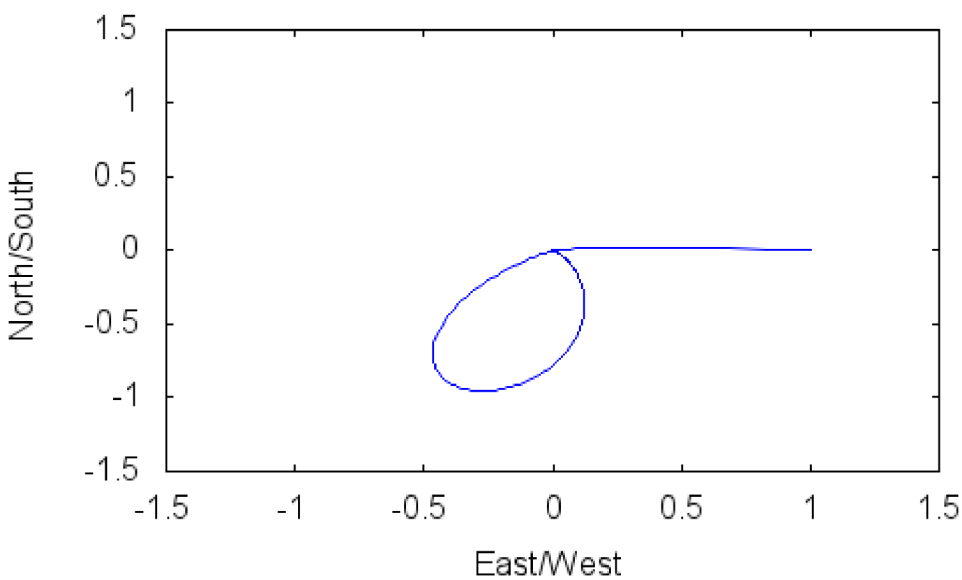
speed: black

x-coordinate of the path of motion: orange

y-coordinate of the path of motion: magenta

```
(%i71) wxdraw2d(
        nticks=200,
        color=blue,
        pp1,
        xlabel="East/West",
        ylabel="North/South",
        xrange=[-1.5,1.5],
        yrange=[-1.5,1.5])$
```

(%t71)



What is happening to your speed as t increases?