Moving in Three Dimensions

Chapter 13 Technology Application Project

1 Introduction

OBJECTIVE: Learn to use Maxima to perform calculations to analyze motion in the equations that are given parameterically.

Moving in three dimensions is something with which we are familiar, but visualizing equations of motion and computations involved in analyzing that motion can be cumbersome. This module gives a broad overview on the analysis of motion in the equations that are given parametrically.

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 Three-Dimentions

First, we load the plots and plottools packages and the define x, t, y, and z coordinates for motion parametrically.

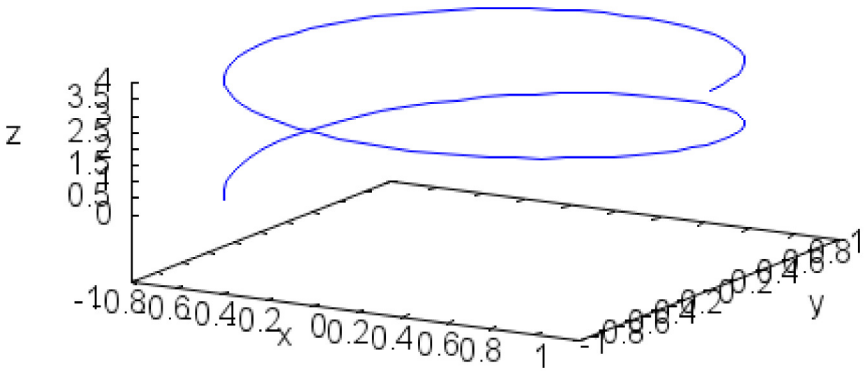
```
(%i1) reset()$
      kill(all)$
      load(draw)$
      ratprint:false$
```

```
(%i3) x(t) := cos(t)$
      y(t) := sin(t)$
      z(t) := 4-t^2/25$
```

Next, we plot the resulting curve in blue. Can you tell in which direction you are moving on the curve as t increases?

```
(%i6) p1: parametric(x(t),y(t),z(t),t,0,10)$
      wxdraw3d(
        nticks=100,
        color=blue,
        p1,
        xlabel="x",
        ylabel="y",
        zlabel="z")$
```

(%t7)



If you consider the parametric equation as a vector equation for the motion of a particle, the derivative of that vector is the velocity vector that we form by differentiating each component. The commands below plot both the first and second derivative of the vector position function.

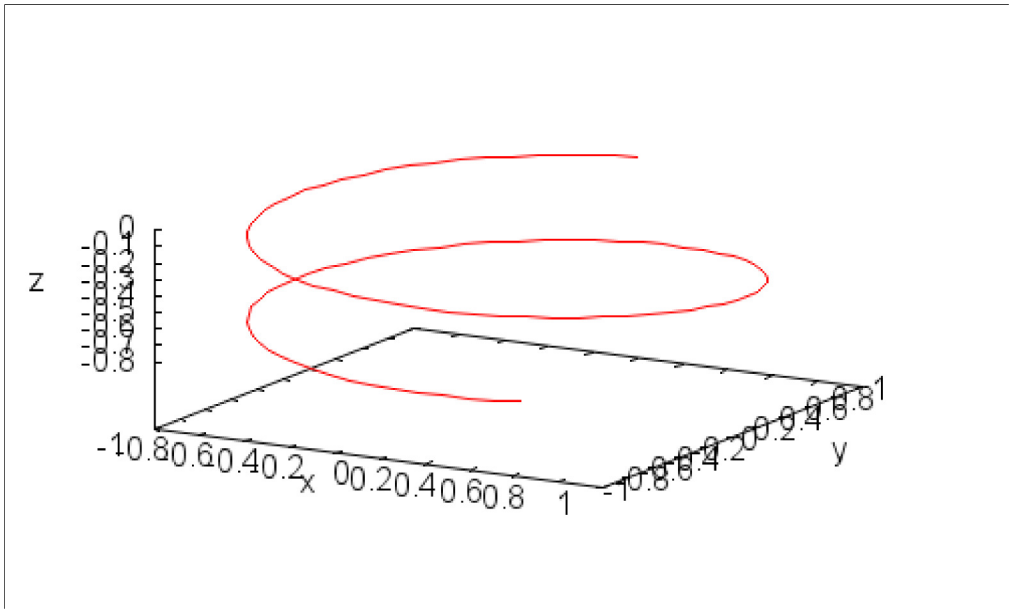
```
(%i8) dx: diff(x(t),t,1)$
      dy: diff(y(t),t,1)$
      dz: diff(z(t),t,1)$
      dvector: [dx,dy,dz]$
      d2x: diff(x(t),t,2)$
      d2y: diff(y(t),t,2)$
      d2z: diff(z(t),t,2)$
      d2vector: [d2x,d2y,d2z]$
      p2: parametric(dx,dy,dz,t,0,10)$
      p3: parametric(d2x,d2y,d2z,t,0,10)$
```

```
(%i18) print("")$
       print("velocity vector: ")$
       print(dvector)$
       wxdraw3d(
         nticks=100,
         color=red,
         p2,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

velocity vector:

$$[-\sin(t),\cos(t),-\frac{2\,t}{25}]$$

(%t21)

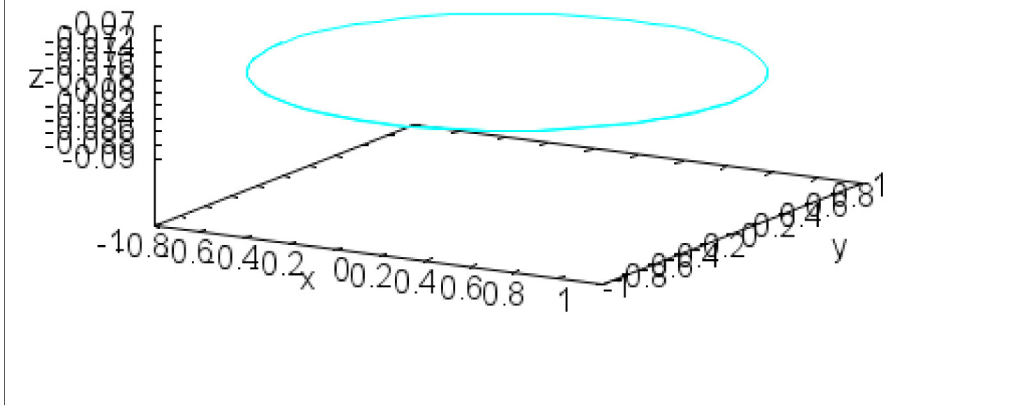


```
(%i22) print("")$
       print("acceleration vector: ")$
       print(d2vector)$
       wxdraw3d(
         nticks=100,
         color=cyan,
         p3,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

acceleration vector:

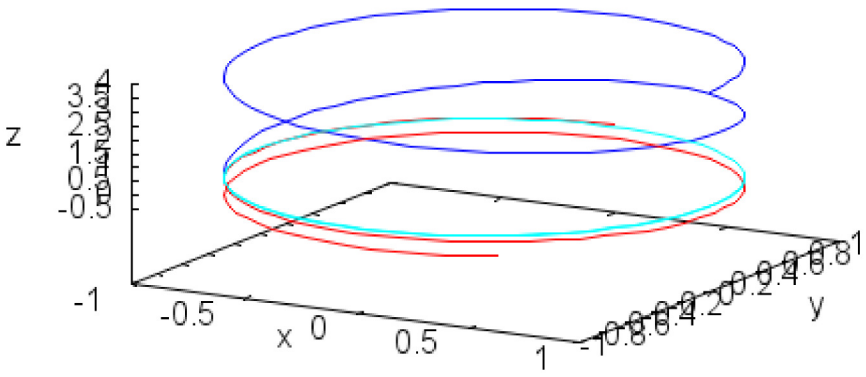
$$[-\cos(t),-\sin(t),-\frac{2}{25}]$$

(%t25)



```
(%i26) wxdraw3d(
         nticks=100,
         color=blue,
         p1,
         color=red,
         p2,
         color=cyan,
         p3,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

(%t26)



2.1 You Try It: Part I

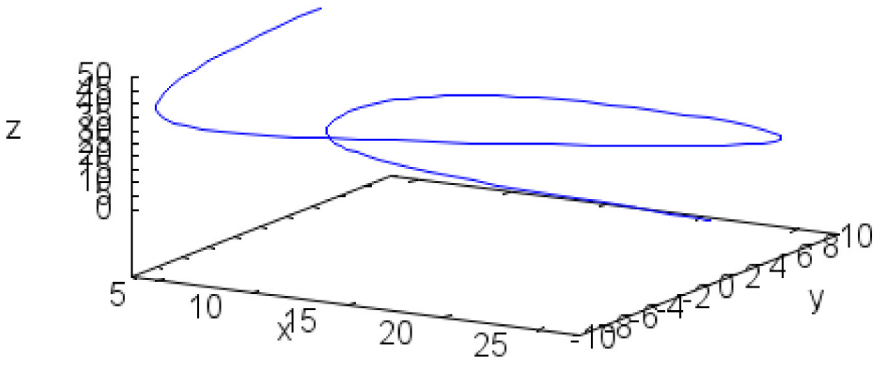
Define your own x, y, and z coordinates for motion by changing the entries for x, y, z. You may or may not want to change the time interval (timeinterval) over which you plot your functions.

```
(%i27) kill(all)$
       load(draw)$
       ratprint:false$

       x(t) := 10*exp(cos(t))$
       y(t) := 10*sin(t^2/15)$
       z(t) := t^2/2$
       timeinterval: 10$

       p1: parametric(x(t),y(t),z(t),t,0,timeinterval)$
       wxdraw3d(
         nticks=100,
         color=blue,
         p1,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

(%t8)



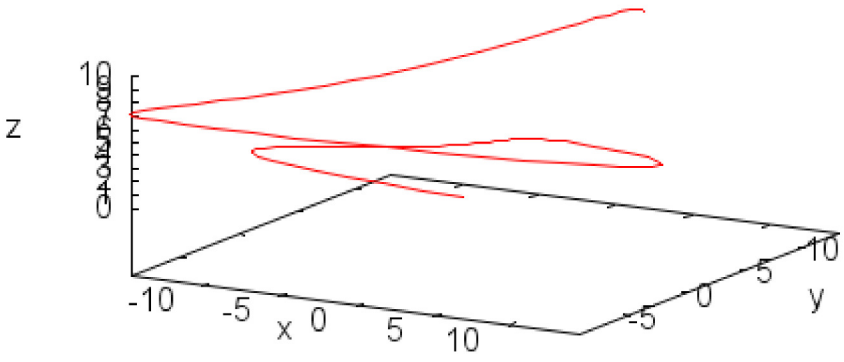

```
(%i9) dx: diff(x(t),t,1)$
      dy: diff(y(t),t,1)$
      dz: diff(z(t),t,1)$
      dvector: [dx,dy,dz]$
      d2x: diff(x(t),t,2)$
      d2y: diff(y(t),t,2)$
      d2z: diff(z(t),t,2)$
      d2vector: [d2x,d2y,d2z]$
      p2: parametric(dx,dy,dz,t,0,timeinterval)$
      p3: parametric(d2x,d2y,d2z,t,0,timeinterval)$
```

```
(%i19) print("")$
       print("velocity vector: ")$
       print(dvector)$
       wxdraw3d(
         nticks=100,
         color=red,
         p2,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

velocity vector:

$$[-10\ \%e^{\cos(t)}\ \sin(t), \frac{4\ t\ \cos\left(\frac{t^2}{15}\right)}{3}, t]$$

(%t22)

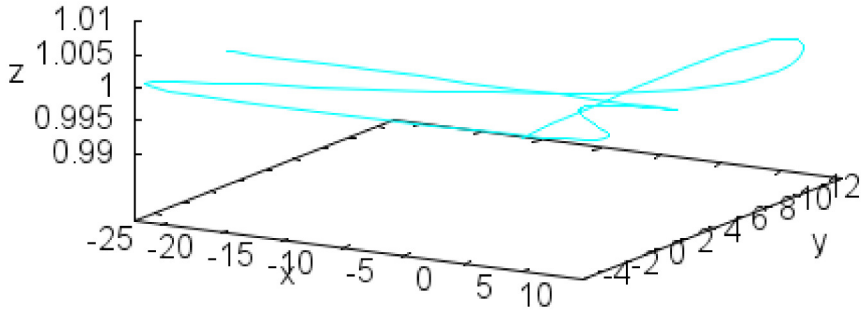


```
(%i23) print("")$
       print("acceleration vector: ")$
       print(d2vector)$
       wxdraw3d(
         nticks=100,
         color=cyan,
         p3,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

acceleration vector:

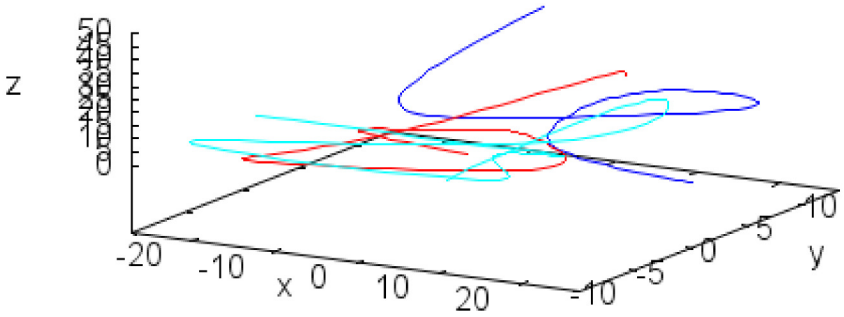
$$[10\ \%e^{\cos(t)}\ \sin(t)^2 - 10\ \%e^{\cos(t)}\ \cos(t), \frac{4\cos\left(\frac{t^2}{15}\right)}{3} - \frac{8\ t^2\ \sin\left(\frac{t^2}{15}\right)}{45}, 1]$$

(%t26)



```
(%i27) wxdraw3d(
         nticks=100,
         color=blue,
         p1,
         color=red,
         p2,
         color=cyan,
         p3,
         xlabel="x",
         ylabel="y",
         zlabel="z")$
```

(%t27)



The plot of all three together may or may not be instructive.

Compare your results to those of your classmates.

3 Part II: Equations of Motion
Velocity -> Position
Velocity -> Acceleration

If you are given the velocity, you can differentiate to find the acceleration and integrate to find the displacement. Since we do three integrations, we need three arbitrary constants to get the general solution.

```
(%i28) kill(all)$
       load(draw)$
       ratprint:false$

       velocity: [3*sqrt(t+1)/2,%e^(-t),1/(t+1)]$
       acceleration: diff(velocity,t,1)$
       position: integrate(velocity, t)$
       rgeneral: position+[a,b,c]$

       print("")$
       print("velocity vector:")$
       print(velocity)$
       print("")$
       print("acceleration vector:")$
       print(acceleration)$
       print("")$
       print("general r (position) vector:")$
       print(rgeneral)$
```

velocity vector:

$$\left[\frac{3\sqrt{t+1}}{2},\%e^{-t},\frac{1}{t+1}\right]$$

acceleration vector:

$$\left[\frac{3}{4\sqrt{t+1}},-\%e^{-t},-\frac{1}{(t+1)^2}\right]$$

general r (position) vector:

$$\left[(t+1)^{3/2}+a,b-\%e^{-t},\log(t+1)+c\right]$$

```
(%i16) s: subst(0,t,rgeneral);
```

$$[a+1,b-1,c]$$

```
(%i17) a: rhs((solve(s[1]=1,a))[1])$
       b: rhs((solve(s[2]=1,b))[1])$
       c: rhs((solve(s[3]=0,c))[1])$
```

```
(%i20) r: position+[a,b,c]$
       print("")$
       print("r (position) vector:")$
       print(r)$
```

r (position) vector:

$$\left[(t+1)^{3/2},2-\%e^{-t},\log(t+1)\right]$$

3.1 You Try It: Part II

Change the entries in your velocity vector and your initial position (velocity and initial), and re-execute the section.

```
(%i24) kill(all)$
       load(draw)$
       ratprint:false$

       velocity: [3*sqrt(t+1)/2,%e^(-t),1/(t+1)]$
       acceleration: diff(velocity,t,1)$
       position: integrate(velocity, t)$
       rgeneral: position+[a,b,c]$

       print("")$
       print("velocity vector:")$
       print(velocity)$
       print("")$
       print("acceleration vector:")$
       print(acceleration)$
       print("")$
       print("general r (position) vector:")$
       print(rgeneral)$
```

velocity vector:

$$\left[\frac{3\sqrt{t+1}}{2},\%e^{-t},\frac{1}{t+1}\right]$$

acceleration vector:

$$\left[\frac{3}{4\sqrt{t+1}},-\%e^{-t},-\frac{1}{(t+1)^2}\right]$$

general r (position) vector:

$$\left[(t+1)^{3/2}+a,b-\%e^{-t},\log(t+1)+c\right]$$

```
(%i16) s: subst(0,t,rgeneral)$

       a: rhs((solve(s[1]=1,a))[1])$
       b: rhs((solve(s[2]=1,b))[1])$
       c: rhs((solve(s[3]=0,c))[1])$

       r: position+[a,b,c]$
       print("")$
       print("r (position) vector:")$
       print(r)$
```

r (position) vector:

$$[(t+1)^{3/2},2-\%e^{-t},\log(t+1)]$$

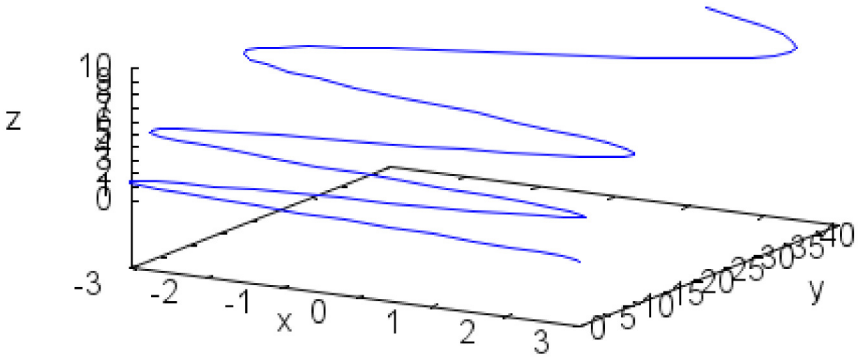
4 Part III: Computing the Distance Traveled on a Curved Path

First we define the x, y, and z coordinates for motion parametrically.

```
(%i24) kill(all)$
       numer:false$
       load(draw)$
       ratprint:false$
```

```
(%i4) r: [3*cos(2*t),t^(10/3)/50,t]$
      v: diff(r,t,1)$
      speed: sqrt(v.v)$
      distance: quad_qags(speed,t,0,10)[1]$
      p1: parametric(r[1],r[2],r[3],t,0,10)$
      wxdraw3d(
        nticks=100,
        color=blue,
        p1,
        xlabel="x",
        ylabel="y",
        zlabel="z")$
```

(%t9)



```
(%i10) print("")$
       print("The position given in feet is:")$
       print(r)$
       print("")$
       print("The velocity given in feet per minute is:")$
       print(v)$
       print("")$
       print("The speed, ds/dt, given in feet per minute is:")$
       print(speed)$
       print("")$
       print("The distance traveled in feet in 10 minutes is:")$
       print(distance)$
```

The position given in feet is:

$$[3\cos(2\,t),\frac{t^{10/3}}{50},t]$$

The velocity given in feet per minute is:

$$[-6\sin(2\,t),\frac{t^{7/3}}{15},1]$$

The speed, ds/dt, given in feet per minute is:

$$\sqrt{36\sin(2\,t)^2+\frac{t^{14/3}}{225}+1}$$

The distance traveled in feet in 10 minutes is:

66.02703743251648

```
In the above example, if you had tried to integrate
symbolically, instead of numerically, what would have happened?
(The quad_qags( ) function signifies numerical integration.)
Many integrals arising in the computation of arc length are
too complicated to evaluate symbolically.
```

```
(%i22) integrate(speed,t,0,10);
```

$$\int_0^{10}\sqrt{36\sin(2\,t)^2+\frac{t^{14/3}}{225}+1}\,\mathrm{d}t$$

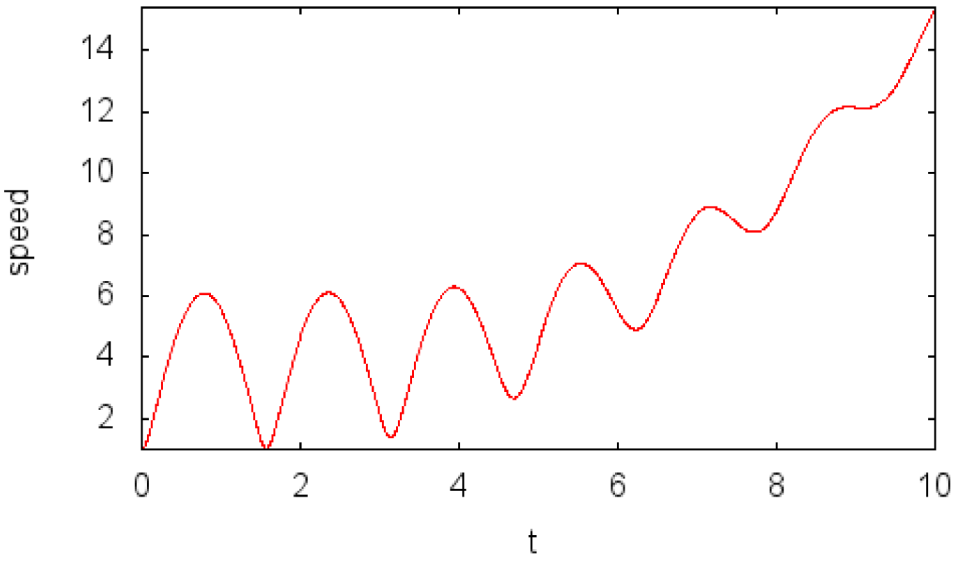
```
(%i23) numer:true$
       print("Speed at 5 minutes in feet per minute is:")$
       print(subst(5,t,speed))$
```

Speed at 5 minutes in feet per minute is:

4.447110790434019

```
(%i26) wxdraw2d(
         nticks=100,
         color=red,
         explicit(speed,t,0,10),
         xlabel="t",
         ylabel="speed");
```

(%t26)



(%o26)

If t represents time in minutes, what does this say about the speed of the hiker? Is it possible?

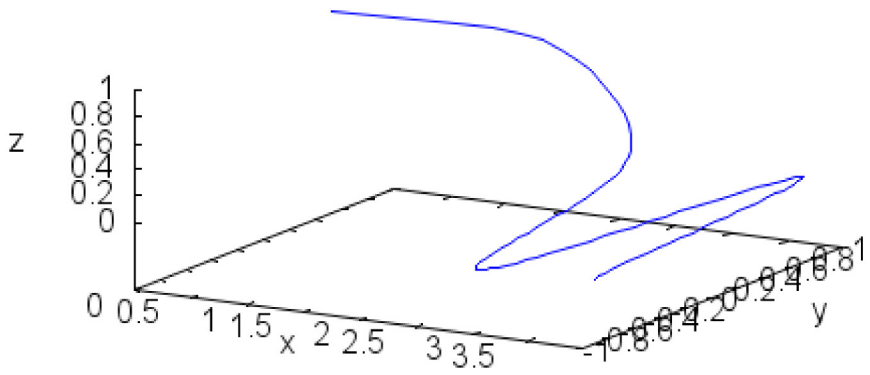
4.1 You Try It: Part III

Choose your own position function (r) and the interval over which you wish to integrate.

```
(%i27) kill(all)$
       numer:false$
       load(draw)$
       ratprint:false$

       r: [2*t^(.3),cos(t-1),exp(-t*t)]$
       v: diff(r,t,1)$
       speed: sqrt(v.v)$
       distance: quad_qags(speed,t,0,10)[1]$
       p1: parametric(r[1],r[2],r[3],t,0,10)$
       wxdraw3d(
          nticks=100,
          color=blue,
          p1,
          xlabel="x",
          ylabel="y",
          zlabel="z")$
```

(%t9)



```
(%i10) print("")$
       print("The position given in feet is:")$
       print(r)$
       print("")$
       print("The velocity given in feet per minute is:")$
       print(v)$
       print("")$
       print("The speed, ds/dt, given in feet per minute is:")$
       print(speed)$
       print("")$
       print("The distance traveled in feet in 10 minutes is:")$
       print(distance)$
```

The position given in feet is:

$$[2\,t^{0.3},\cos(t-1),\%e^{-t^2}]$$

The velocity given in feet per minute is:

$$[\frac{0.6}{t^{0.7}},-\sin(t-1),-2\,t\,\%e^{-t^2}]$$

The speed, ds/dt, given in feet per minute is:

$$\sqrt{\sin(t-1)^2+4\,t^2\,\%e^{-2\,t^2}+\frac{0.36}{t^{1.4}}}$$

The distance traveled in feet in 10 minutes is:

8.862500045739576

5 Part IV: Computing Curvature and Torsion for a Space Curve

```
Maxima simplifies the process of finding curvature and torsion.
The computations below use formulas directly from your text,
and we graphically explore the interpretations of the
curvature and torsion functions.

The vect package define the operator ~ as the cross product.
So v~a finds the cross product of vectors v and a.

Note that the package lapack may take some time to load,
especially if this is the first time loading this package in
Maxima.
```

```
(%i22) kill(all)$
       numer:false$
       load(draw)$
       load(vect)$
       ratprint:false$
```

```
(%i5) mag(v) := (sqrt(v.v))$
      r: [10*sin(t),10*exp(-t),t^(10/3)/100]$
      v: diff(r,t,1)$
      a: diff(v,t,1)$
      speed: mag(v)$
      utan: [v[1]/speed,v[2]/speed,v[3]/speed]$
      crossva: express(v~a)$
      transcrossva: transpose(crossva)$
      curvature: mag(crossva)/speed^3$
      m2: matrix(v,a,diff(a,t,1))$
      det2: determinant(m2)$
      prod2: crossva.transcrossva$
      torsion: det2/prod2$
```

```
(%i18) print("")$
       print("The position vector is:")$
       print(r)$
       print("")$
       print("The velocity vector is:")$
       print(v)$
       print("")$
       print("The acceleration vector is:")$
       print(a)$
       print("")$
       print("The speed is:")$
       print(speed)$
       print("")$
       print("The unit tangent vector is:")$
       print(utan)$
       print("")$
```

The position vector is:

$$[10\sin(t), 10\ \%e^{-t}, \frac{t^{10/3}}{100}]$$

The velocity vector is:

$$[10\cos(t), -10\ \%e^{-t}, \frac{t^{7/3}}{30}]$$

The acceleration vector is:

$$[-10\sin(t), 10\ \%e^{-t}, \frac{7\ t^{4/3}}{90}]$$

The speed is:

$$\sqrt{100\cos(t)^2 + 100\ \%e^{-2\ t} + \frac{t^{14/3}}{900}}$$

The unit tangent vector is:

$$[\frac{10\cos(t)}{\sqrt{100\cos(t)^2 + 100\ \%e^{-2\ t} + \frac{t^{14/3}}{900}}}, -\frac{10\ \%e^{-t}}{\sqrt{100\cos(t)^2 + 100\ \%e^{-2\ t} + \frac{t^{14/3}}{900}}}, \frac{t^{7/3}}{30\sqrt{100\cos(t)^2 + 100\ \%e^{-2\ t} + \frac{t^{14/3}}{900}}}]$$

(%i34)
```
print("The curvature is:")$
print(curvature)$
print("")$
print("The torsion is:")$
print(torsion)$
```

The curvature is:

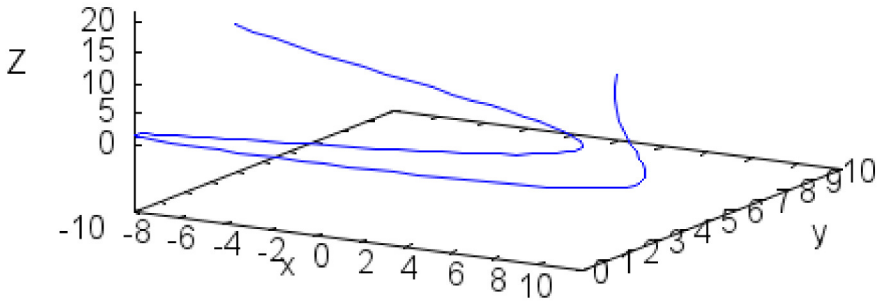
$$\frac{\sqrt{\left(100\ \%e^{-t}\cos(t)-100\ \%e^{-t}\sin(t)\right)^2+\left(-\frac{t^{7/3}\sin(t)}{3}-\frac{7\,t^{4/3}\cos(t)}{9}\right)^2+\left(-\frac{t^{7/3}\,\%e^{-t}}{3}-\frac{7\,t^{4/3}\,\%e^{-t}}{9}\right)^2}}{\left(100\cos(t)^2+100\ \%e^{-2\,t}+\frac{t^{14/3}}{900}\right)^{3/2}}$$

The torsion is:

$$\left(\frac{t^{7/3}\left(100\ \%e^{-t}\sin(t)+100\ \%e^{-t}\cos(t)\right)}{30}+10\ \%e^{-t}\left(\frac{7\,t^{4/3}\cos(t)}{9}-\frac{28\,t^{1/3}\sin(t)}{27}\right)+10\left(\frac{7\,t^{4/3}\,\%e^{-t}}{9}+\frac{28\,t^{1/3}\,\%e^{-t}}{27}\right)\cos(t)\right)\Big/\left(\left(100\ \%e^{-t}\cos(t)-100\ \%e^{-t}\sin(t)\right)^2+\left(-\frac{t^{7/3}\sin(t)}{3}-\frac{7\,t^{4/3}\cos(t)}{9}\right)^2+\left(-\frac{t^{7/3}\,\%e^{-t}}{3}-\frac{7\,t^{4/3}\,\%e^{-t}}{9}\right)^2\right)$$

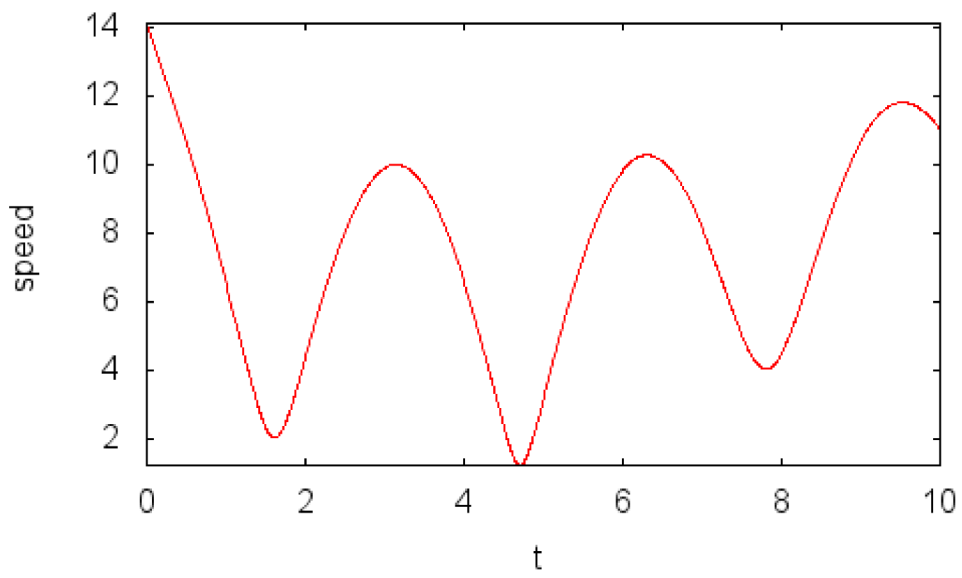
(%i39)
```
p1: parametric(r[1],r[2],r[3],t,0,10)$
wxdraw3d(
  nticks=100,
  color=blue,
  p1,
  xlabel="x",
  ylabel="y",
  zlabel="Z")$
```

(%t40)



```
(%i41) p2: explicit(speed,t,0,10)$
       wxdraw2d(
         nticks=200,
         color=red,
         p2,
         xlabel="t",
         ylabel="speed");
```

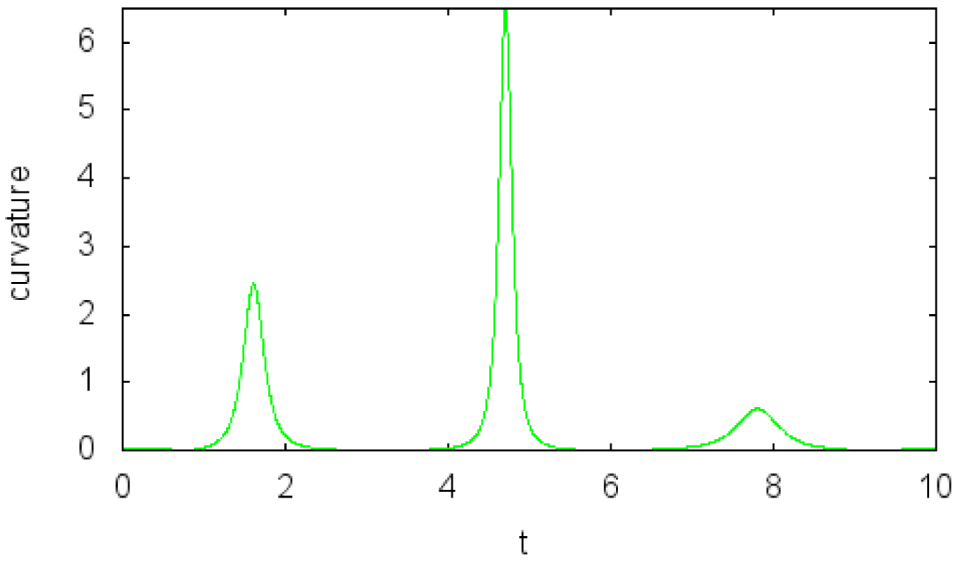
(%t42)



(%o42)

```
(%i43) p3: explicit(curvature,t,0,10)$
       wxdraw2d(
         nticks=200,
         color=green,
         p3,
         xlabel="t",
         ylabel="curvature");
```

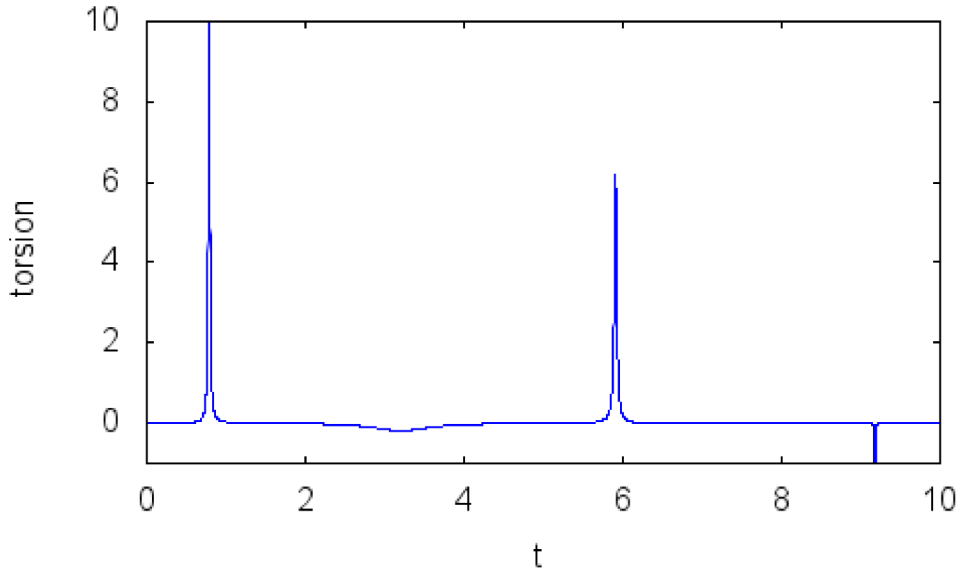
(%t44)



(%o44)

```
(%i45) p4: explicit(torsion,t,0,10)$
       wxdraw2d(
         nticks=200,
         color=blue,
         p4,
         xlabel="t",
         yrange=[-1,10],
         ylabel="torsion");
```

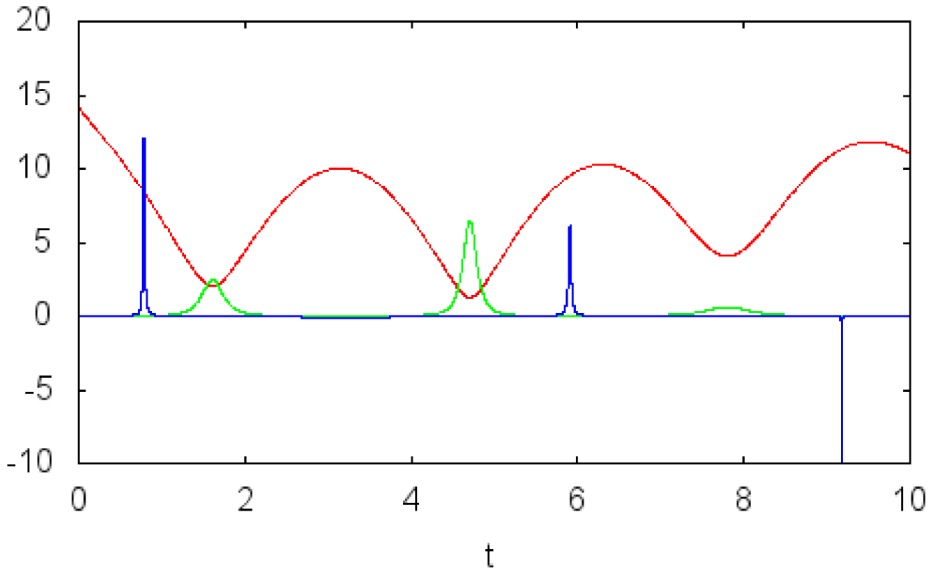
(%t46)



(%o46)

```
(%i47) wxdraw2d(
         nticks=200,
         color=red,
         p2,
         color=green,
         p3,
         color=blue,
         p4,
         yrange=[-10,20],
         xlabel="t");
```

(%t47)



(%o47)

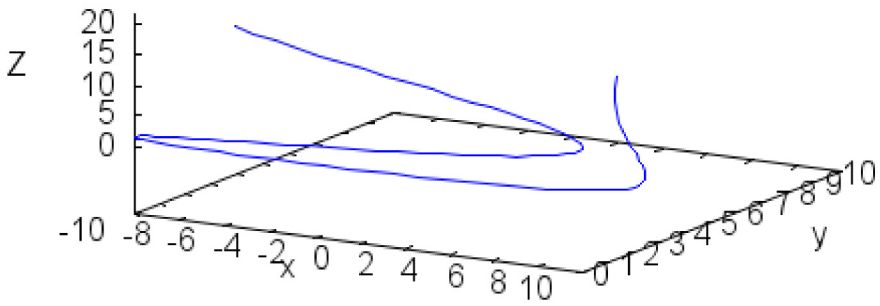
Look at your space curve and see if you can identify the points on your space curve at which the curvature or torsion spike. What has caused these events? Are they related at all to the speed?

Recall from the definition of curvature that relates it to the cross product of the velocity and acceleration that the curvature is smallest when the velocity and acceleration are in the same direction. The curvature is largest when the velocity and acceleration are perpendicular to one another.

Looking at the torsion as the dot product of the derivative of the unit binormal and the normal vectors. Torsion will peak when those two vectors are in the same direction and be close to 0 when those two vectors are perpendicular.

```
(%i48) wxdraw3d(
         nticks=100,
         color=blue,
         p1,
         xlabel="x",
         ylabel="y",
         zlabel="Z")$
```

(%t48)



You can see the TNB frame for a curve in action in Part VI of this module.

5.1 You Try It: Part IV

Redefine your position function (r) and compute. It should be noted that the computations here are not trivial, so more complicated functions could lead to long waits.

```
(%i49) kill(all)$
       numer:false$
       load(draw)$
       load(vect)$
       ratprint:false$
```

```
(%i5) mag(v) := (sqrt(v.v))$
      r: [2*t^.3,cos(t-1),%e^(-t*t)]$
      v: diff(r,t,1)$
      a: diff(v,t,1)$
      speed: trigsimp(mag(v))$
      utan: [v[1]/speed,v[2]/speed,v[3]/speed]$
      crossva: express(v~a)$
      transcrossva: transpose(crossva)$
      curvature: trigsimp(mag(crossva)/speed^3)$
      m2: matrix(v,a,diff(a,t,1))$
      det2: determinant(m2)$
      prod2: crossva.transcrossva$
      torsion: trigsimp(det2/prod2)$
```

```
(%i18) print("")$
       print("The position vector is:")$
       print(r)$
       print("")$
       print("The velocity vector is:")$
       print(v)$
       print("")$
       print("The acceleration vector is:")$
       print(a)$
       print("")$
       print("The speed is:")$
       print(speed)$
       print("")$
```

The position vector is:

$$[2\,t^{0.3},\cos(t-1),\%e^{-t^2}]$$

The velocity vector is:

$$[\frac{0.6}{t^{0.7}},-\sin(t-1),-2\,t\,\%e^{-t^2}]$$

The acceleration vector is:

$$[-\frac{0.42}{t^{1.7}},-\cos(t-1),4\,t^2\,\%e^{-t^2}-2\,\%e^{-t^2}]$$

The speed is:

$$\frac{\%e^{-t^2}\sqrt{-\frac{t^{2/5}\,(25\,t\,\%e^{2\,t^2}\cos(t-1)^2-25\,t\,\%e^{2\,t^2}-100\,t^3)-9\,\%e^{2\,t^2}}{t^{7/5}}}}{5}$$

```
(%i31) print("The unit tangent vector is:")$
       print(utan)$
       print("")$
       print("The curvature is:")$
       print(curvature)$
       print("")$
       print("The torsion is:")$
       print(torsion)$
```

The unit tangent vector is:

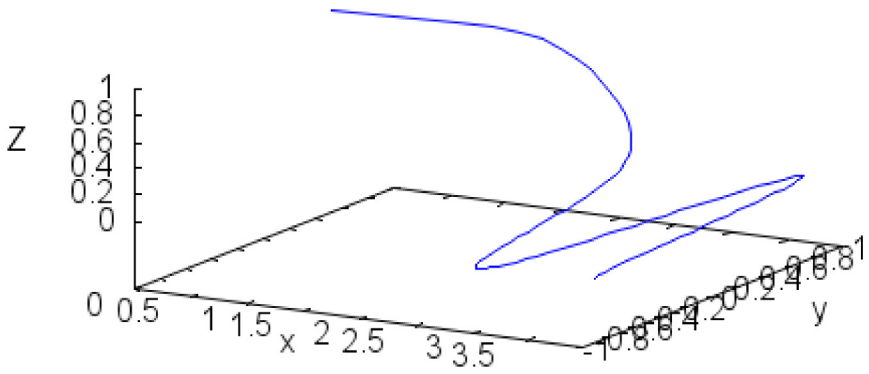
$$\left[\frac{3.0\ \%e^{t^2}}{t^{0.7}\sqrt{-\frac{t^{2/5}\left(25\ t\ \%e^{2\ t^2}\cos(t-1)^2-25\ t\ \%e^{2\ t^2}-100\ t^3\right)-9\ \%e^{2\ t^2}}{t^{7/5}}}},-\frac{5\ \%e^{t^2}\sin(t-1)}{\sqrt{-\frac{t^{2/5}\left(25\ t\ \%e^{2\ t^2}\cos(t-1)^2-25\ t\ \%e^{2\ t^2}-100\ t^3\right)-9\ \%e^{2\ t^2}}{t^{7/5}}}},-\frac{10\ t}{\sqrt{-\frac{t^{2/5}\left(25\ t\ \%e^{2\ t^2}\cos(t-1)^2-25\ t\ \%e^{2\ t^2}-100\ t^3\right)-9\ \%e^{2\ t^2}}{t^{7/5}}}}\right]$$

The curvature is: $\left(5\ \%e^{2\ t^2}\ sqrt\left(t^{3/5}\left(\left(441-900\ t^2\right)\%e^{2\ t^2}\sin(t-1)^2+1260\ t\ \%e^{2\ t^2}\cos(t-1)\sin(t-1)+900\ t^2\ \%e^{2\ t^2}+14400\ t^6-24480\ t^4+10404\ t^2\right)+\left(40000\ t^8-50000\ t^6+10000\ t^4\right)\sin(t-1)^2+\left(40000\ t^7-20000\ t^5\right)\cos(t-1)\sin(t-1)+10000\ t^6\right)\right)/\left(2\ t^2\left(\frac{t^{2/5}\left(25\ t\ \%e^{2\ t^2}\sin(t-1)^2+100\ t^3\right)+9\ \%e^{2\ t^2}}{t^{7/5}}\right)^{3/2}\right)$

The torsion is: $\left(\left(2400\ t^5-14640\ t^3+3570\ t\right)\%e^{t^2}\sin(t-1)+\left(12000\ t^6-18000\ t^4-3570\ t^2\right)\%e^{t^2}\cos(t-1)\right)/\left(t^{3/10}\left(\left(441-900\ t^2\right)\%e^{2\ t^2}\sin(t-1)^2+1260\ t\ \%e^{2\ t^2}\cos(t-1)\sin(t-1)+900\ t^2\ \%e^{2\ t^2}+14400\ t^6-24480\ t^4+10404\ t^2\right)+t^{7/10}\left(\left(40000\ t^7-50000\ t^5+10000\ t^3\right)\sin(t-1)^2+\left(40000\ t^6-20000\ t^4\right)\cos(t-1)\sin(t-1)+10000\ t^5\right)\right)$

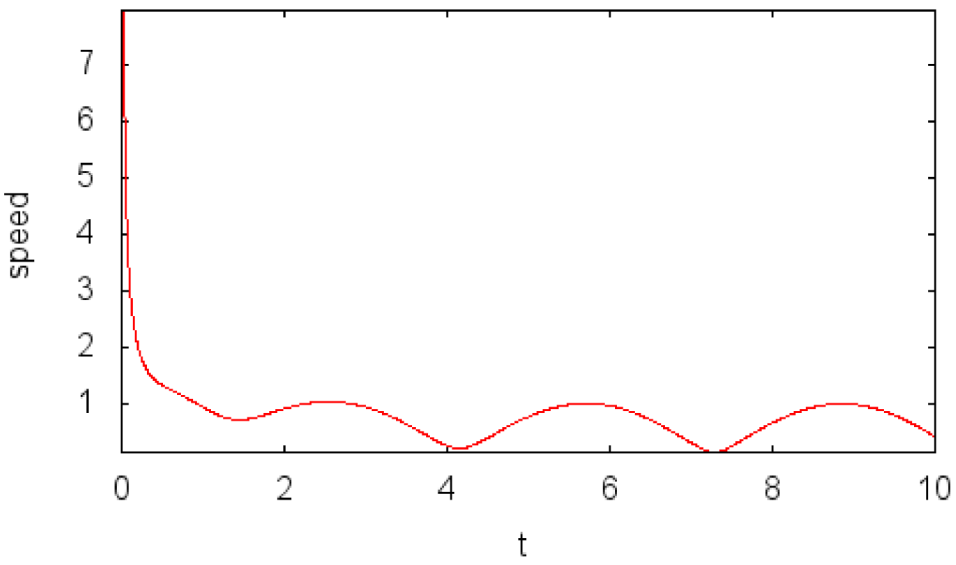
```
(%i39) p1: parametric(r[1],r[2],r[3],t,0,10)$
       wxdraw3d(
         nticks=100,
         color=blue,
         p1,
         xlabel="x",
         ylabel="y",
         zlabel="Z")$
```

(%t40)



```
(%i41) p2: explicit(speed,t,0,10)$
       wxdraw2d(
         nticks=200,
         color=red,
         p2,
         xlabel="t",
         ylabel="speed");
```

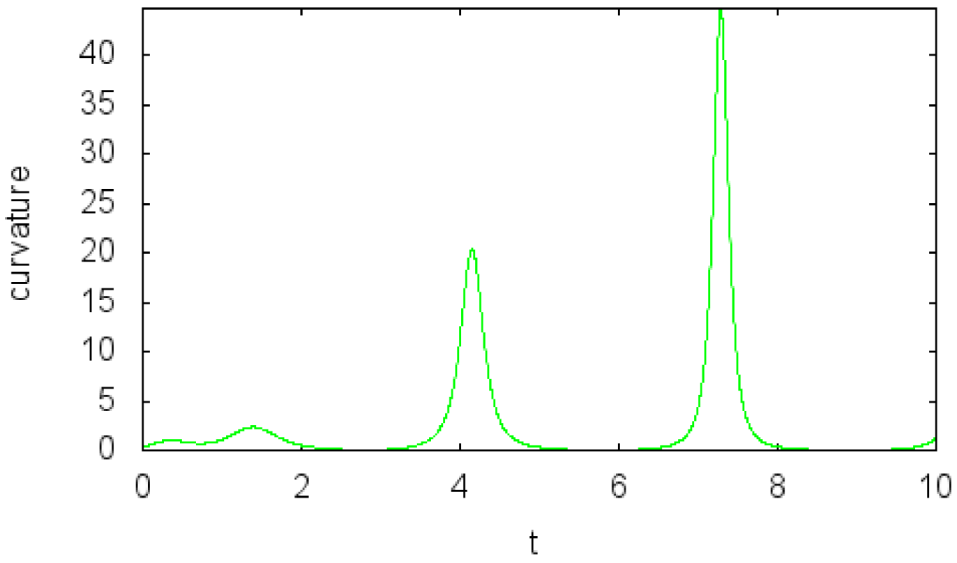
(%t42)



(%o42)

```
(%i43) p3: explicit(curvature,t,0,10)$
       wxdraw2d(
         nticks=200,
         color=green,
         p3,
         xlabel="t",
         ylabel="curvature");
```

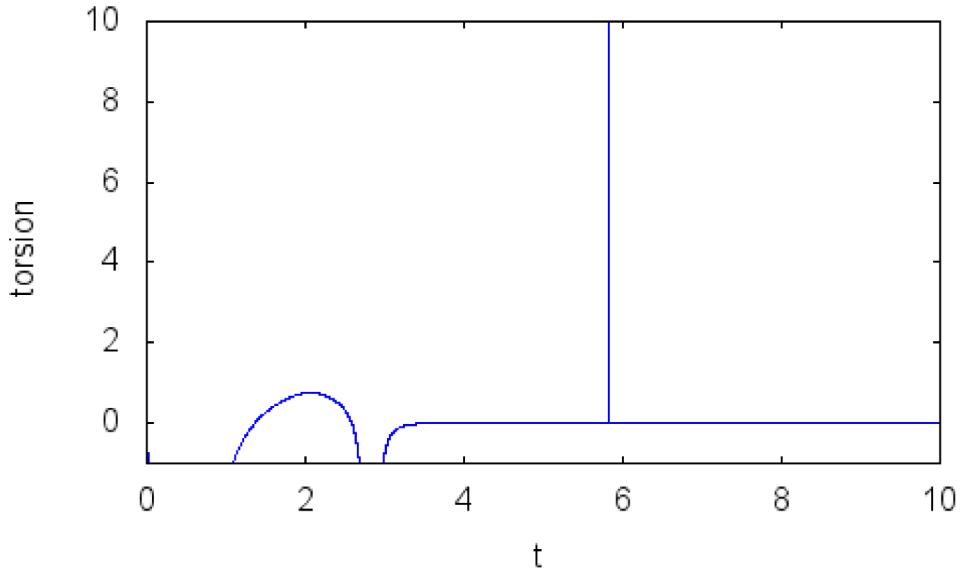
(%t44)



(%o44)

```
(%i45) p4: explicit(torsion,t,0,10)$
       wxdraw2d(
         nticks=200,
         color=blue,
         p4,
         xlabel="t",
         yrange=[-1,10],
         ylabel="torsion");
```

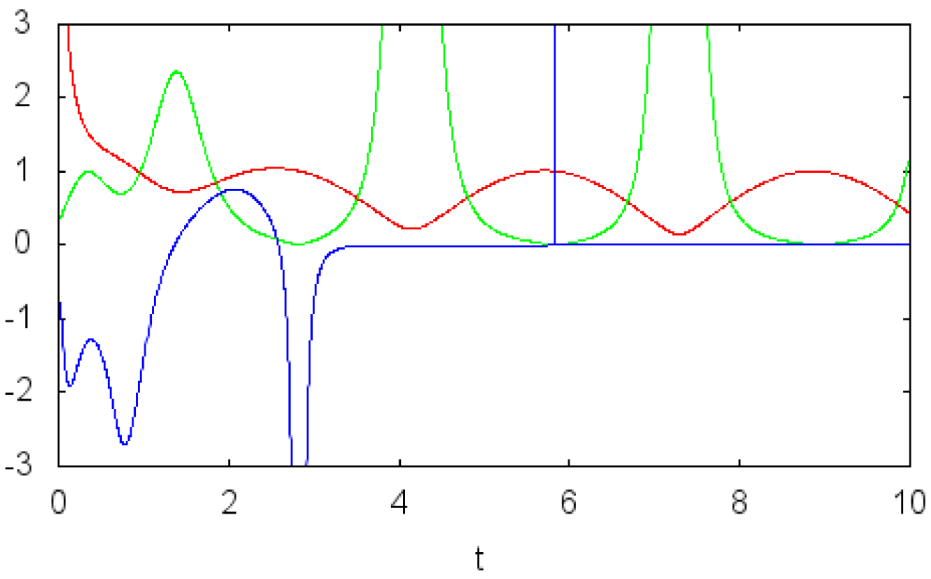
(%t46)



(%o46)

```
(%i47) wxdraw2d(
         nticks=200,
         color=red,
         p2,
         color=green,
         p3,
         color=blue,
         p4,
         yrange=[-3,3],
         xrange=[0,10],
         xlabel="t");
```

(%t47)



(%o47)

6 Part V: TNB Frame - Computation and Visualization

The steps here resemble the ones above, except that we choose a simpler function and extend the formulas to the unit normal and binormal vectors.

```
(%i48) kill(all)$
       numer:false$
       load(draw)$
       load(vect)$
       ratprint:false$
```

```
(%i5) mag(v) := (sqrt(v.v))$
      r: [sin(2*t),cos(2*t),t^2/10]$
      v: diff(r,t,1)$
      a: diff(v,t,1)$
      speed: trigsimp(mag(v))$
      utan: [v[1]/speed,v[2]/speed,v[3]/speed]$
      crossva: express(v~a)$
      transcrossva: transpose(crossva)$
      curvature: trigsimp(mag(crossva)/speed^3)$
      m2: matrix(v,a,diff(a,t,1))$
      det2: determinant(m2)$
      prod2: crossva.transcrossva$
      torsion: trigsimp(det2/prod2)$
      top: trigsimp(diff(utan,t))$
      bottom: trigsimp(radcan(mag(top)))$
      un: trigsimp(radcan(top/bottom))$
      ubn: trigsimp(radcan(express(utan~un)))$
```

```
(%i22) print("")$
       print("The position vector is:")$
       print(r)$
       print("")$
       print("The velocity vector is:")$
       print(v)$
       print("")$
       print("The acceleration vector is:")$
       print(a)$
       print("")$
       print("The speed is:")$
       print(speed)$
       print("")$
```

The position vector is:

$$[\sin(2\ t),\cos(2\ t),\frac{t^2}{10}]$$

The velocity vector is:

$$[2\ \cos(2\ t),-2\ \sin(2\ t),\frac{t}{5}]$$

The acceleration vector is:

$$[-4\ \sin(2\ t),-4\ \cos(2\ t),\frac{1}{5}]$$

The speed is:

$$\frac{\sqrt{t^2+100}}{5}$$

```
(%i35) print("The unit tangent vector is:")$
       print(utan)$
       print("")$
       print("The curvature is:")$
       print(curvature)$
       print("")$
       print("The torsion is:")$
       print(torsion)$
       print("")$
       print("The unit normal is:")$
       print(un)$
       print("")$
       print("The unit binormal is:")$
       print(ubn)$
```

The unit tangent vector is:

$$[\frac{10\cos(2\,t)}{\sqrt{t^2+100}},-\frac{10\sin(2\,t)}{\sqrt{t^2+100}},\frac{t}{\sqrt{t^2+100}}]$$

The curvature is:

$$\frac{25\sqrt{16\,t^2+1604}}{(t^2+100)^{3/2}}$$

The torsion is:

$$-\frac{40\,t}{4\,t^2+401}$$

The unit normal is:

$$[-\frac{(2\,t^2+200)\sin(2\,t)+t\cos(2\,t)}{\sqrt{t^2+100}\sqrt{4\,t^2+401}},\frac{t\sin(2\,t)+(-2\,t^2-200)\cos(2\,t)}{\sqrt{t^2+100}\sqrt{4\,t^2+401}},\frac{10}{\sqrt{t^2+100}\sqrt{4\,t^2+401}}]$$

The unit binormal is:

$$[-\frac{\sin(2\,t)-2\,t\cos(2\,t)}{\sqrt{4\,t^2+401}},-\frac{2\,t\sin(2\,t)+\cos(2\,t)}{\sqrt{4\,t^2+401}},-\frac{20}{\sqrt{4\,t^2+401}}]$$

Let's find the tangential and normal components of the accelerations.

```
(%i49) at: trigsimp(a*utan).[1,1,1]$
       an: ratsimp(trigreduce(ratsimp((a*un)).[1,1,1]))$
       print("")$
       print("The tangential component of acceleration is ", at)$
       print("The normal component of acceleration is ", an)$
```

The tangential component of acceleration is $\frac{t}{5\sqrt{t^2+100}}$

The normal component of acceleration is $\frac{2\sqrt{4\,t^2+401}}{\sqrt{t^2+100}}$

```
You can get a visual perspective of the tangential and normal
components of the acceleration in two dimensions by exploring
the Java applet, "Tangent and Normal Vectors."

If you compare these components found by the dot products to
the formulas given in your book, you will see that they agree,
once the absolute value sign is removed from the curvature
formula.
```

```
(%i54) print("The tangential component of acceleration is ", diff(speed,t,1))$
       print("The normal component of acceleration is ", radcan(curvature*speed^2))$
```

The tangential component of acceleration is $\frac{t}{5\sqrt{t^2+100}}$

The normal component of acceleration is $\frac{2\sqrt{4\,t^2+401}}{\sqrt{t^2+100}}$

```
The equality of the tangential components can be verified
easily if you notice that the first two terms of the result
found previously by dotting the acceleration vector into the
unit tangent vector subtract out with one another. The equality
of the normal components is not as obvious. You can see the
advantage in using the formulas used in the latter part, due to
the complexity of the unit normal vector.

The following set of commands will plot the unit tangent, unit
normal, and unit binormal vectors as you move along the curve.
The resulting animated image will be saved in the wxMaxima
folder with the filename "13_m3_1.gif".
```

```
(%i56) p1: parametric(r[1],r[2],r[3],t,0,2*%pi)$
       numer: true$
       P: []$

       for i: 0 thru 6.4 step .4 do block(
          p2: vector(subst(i,t,r),subst(i,t,utan)),
          p3: vector(subst(i,t,r),subst(i,t,un)),
          p4: vector(subst(i,t,r),subst(i,t,ubn)),
          p5: points([
             subst(i,t,r),
             subst(i,t,r+0.2*ubn),
             subst(i,t,r+0.2*utan+0.2*ubn),
             subst(i,t,r+0.2*utan),
             subst(i,t,r),
             subst(i,t,r+0.2*un),
             subst(i,t,r+0.2*utan+0.2*un),
             subst(i,t,r+0.2*utan),
             subst(i,t,r+0.2*utan+0.2*ubn),
             subst(i,t,r+0.2*utan+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*utan+0.2*un),
             subst(i,t,r+0.2*utan+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*un),
             subst(i,t,r+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*ubn)]),

          P: append(P,[gr3d(
             nticks=100,
             color=black,
             line_width=2,
             p1,
             color=red,
             line_width=3,
             p2,
             color=green,
             p3,
             color=blue,
             p4,
             points_joined=true,
             color=black,
             line_width=1,
             point_type=-1,
             p5,
             xrange=[-1.5,1.5],
             yrange=[-1.5,1.5],
             zrange=[0,4],
             title="Tangent (Red), Unit Normal (Green), Unit Binormal (Blue)")]))$
```

```
(%i60) draw(
         terminal=animated_gif,
         delay=10,
         file_name="13_m3_1",
         P)$
```

6.1 You Try It: Part V

```
The computational code is written so that all you have to do is
change the initial vector ( r ) function and then re-execute the
commands. We suggest that you choose some of the homework
problems from your text for the TNB frame, since the computation
for the unit normal and unit binormal can get you very bogged
down, even in Maple. You may also wish to change the domain over
which you extend your function in the second section.

If you check out the following example, expect to wait a few
minutes for the computations.
```

```
(%i61) kill(all)$
       numer:false$
       load(draw)$
       load(vect)$
       ratprint:false$
```

```
 (%i5) mag(v) := (sqrt(v.v))$
       r: [2*t,cos(t-1),%e^(-t)]$
       v: diff(r,t,1)$
       a: diff(v,t,1)$
       speed: trigsimp(mag(v))$
       utan: [v[1]/speed,v[2]/speed,v[3]/speed]$
       crossva: express(v~a)$
       transcrossva: transpose(crossva)$
       curvature: trigsimp(mag(crossva)/speed^3)$
       m2: matrix(v,a,diff(a,t,1))$
       det2: determinant(m2)$
       prod2: crossva.transcrossva$
       torsion: trigsimp(det2/prod2)$
       top: trigsimp(diff(utan,t))$
       bottom: trigsimp(radcan(mag(top)))$
       un: trigsimp(radcan(top/bottom))$
       ubn: trigsimp(radcan(express(utan~un)))$
```

```
(%i22) print("")$
       print("The position vector is:")$
       print(r)$
       print("")$
       print("The velocity vector is:")$
       print(v)$
       print("")$
       print("The acceleration vector is:")$
       print(a)$
       print("")$
       print("The speed is:")$
       print(speed)$
       print("")$
       print("The unit tangent vector is:")$
       print(utan)$
       print("")$
```

The position vector is:

$$[2\,t,\cos(t-1),\%e^{-t}]$$

The velocity vector is:

$$[2,-\sin(t-1),-\%e^{-t}]$$

The acceleration vector is:

$$[0,-\cos(t-1),\%e^{-t}]$$

The speed is:

$$\%e^{-t}\sqrt{-\%e^{2\,t}\cos(t-1)^2+5\,\%e^{2\,t}+1}$$

The unit tangent vector is:

$$\left[\frac{2\,\%e^{t}}{\sqrt{-\%e^{2\,t}\cos(t-1)^2+5\,\%e^{2\,t}+1}},-\frac{\%e^{t}\sin(t-1)}{\sqrt{-\%e^{2\,t}\cos(t-1)^2+5\,\%e^{2\,t}+1}},-\frac{1}{\sqrt{-\%e^{2\,t}\cos(t-1)^2+5\,\%e^{2\,t}+1}}\right]$$

```
(%i38) print("The curvature is:")$
       print(curvature)$
       print("")$
       print("The torsion is:")$
       print(torsion)$
       print("")$
       print("The unit normal is:")$
       print(un)$
       print("")$
```

The curvature is:

$$\frac{\%e^{2\,t}\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2\,t}\cos(t-1)^2+5}}{(\%e^{2\,t}\sin(t-1)^2+4\,\%e^{2\,t}+1)^{3/2}}$$

The torsion is:

$$-\frac{2\,\%e^{t}\sin(t-1)-2\,\%e^{t}\cos(t-1)}{2\cos(t-1)\sin(t-1)+4\,\%e^{2\,t}\cos(t-1)^2+5}$$

The unit normal is:

$$\left[\frac{(2\,\%e^{2\,t}\cos(t-1)\sin(t-1)-2)\sqrt{\%e^{2\,t}\sin(t-1)^2+4\,\%e^{2\,t}+1}}{(\%e^{2\,t}\cos(t-1)^2-5\,\%e^{2\,t}-1)\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2\,t}\cos(t-1)^2+5}},-\right.$$

$$\frac{\sin(t-1)+(4\,\%e^{2\,t}+1)\cos(t-1)}{\sqrt{-\%e^{2\,t}\cos(t-1)^2+5\,\%e^{2\,t}+1}\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2\,t}\cos(t-1)^2+5}},-(\%e^{3\,t}\sin(t-1)^4+\%e^{3\,t}$$

$$\cos(t-1)\sin(t-1)^3+(8\,\%e^{3\,t}+\%e^{t})\sin(t-1)^2+(4\,\%e^{3\,t}+\%e^{t})\cos(t-1)\sin(t-1)+16\,\%e^{3\,t}+4\,\%e^{t})$$

$$/(\sqrt{-\%e^{2\,t}\cos(t-1)^2+5\,\%e^{2\,t}+1}\left(\%e^{2\,t}\cos(t-1)^2-5\,\%e^{2\,t}-1\right)$$

$$\left.\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2\,t}\cos(t-1)^2+5}\,)\right]$$

(%i47)
```
print("The unit binormal is:")$
print(ubn)$
```

The unit binormal is:

$$\left[-\frac{\sin(t-1)+\cos(t-1)}{\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2t}\cos(t-1)^2+5}},-\left(\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2t}\cos(t-1)^2+5}\right.\right.$$
$$\left(\left(2\,\%e^{4t}\cos(t-1)^3+\left(-10\,\%e^{4t}-2\,\%e^{2t}\right)\cos(t-1)\right)\sin(t-1)-2\,\%e^{2t}\cos(t-1)^2+10\,\%e^{2t}+2\right)$$
$$\sqrt{\%e^{2t}\sin(t-1)^2+4\,\%e^{2t}+1}+\sqrt{-\%e^{2t}\cos(t-1)^2+5\,\%e^{2t}+1}$$
$$\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2t}\cos(t-1)^2+5}\,\left(2\,\%e^{4t}\sin(t-1)^4+2\,\%e^{4t}\cos(t-1)\sin(t-1)^3+\right.$$
$$\left.\left.(16\,\%e^{4t}+2\,\%e^{2t})\sin(t-1)^2+(8\,\%e^{4t}+2\,\%e^{2t})\cos(t-1)\sin(t-1)+32\,\%e^{4t}+8\,\%e^{2t}\right)\right)/\left(\right.$$
$$\sqrt{-\%e^{2t}\cos(t-1)^2+5\,\%e^{2t}+1}\,\left(\right.$$
$$\left(2\,\%e^{4t}\cos(t-1)^5+\left(-20\,\%e^{4t}-4\,\%e^{2t}\right)\cos(t-1)^3+\left(50\,\%e^{4t}+20\,\%e^{2t}+2\right)\cos(t-1)\right)\sin(t-1)+4$$
$$\%e^{6t}\cos(t-1)^6+\left(-40\,\%e^{6t}-3\,\%e^{4t}\right)\cos(t-1)^4+\left(100\,\%e^{6t}-10\,\%e^{4t}-6\,\%e^{2t}\right)\cos(t-1)^2+125$$
$$\left.\left.\%e^{4t}+50\,\%e^{2t}+5\right)\right),\left(\sqrt{-\%e^{2t}\cos(t-1)^2+5\,\%e^{2t}+1}\right.$$
$$\left(2\,\%e^{t}\sin(t-1)+(8\,\%e^{3t}+2\,\%e^{t})\cos(t-1)\right)\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2t}\cos(t-1)^2+5}+$$
$$\sqrt{2\cos(t-1)\sin(t-1)+4\,\%e^{2t}\cos(t-1)^2+5}\,\sqrt{\%e^{2t}\sin(t-1)^2+4\,\%e^{2t}+1}$$
$$\left.\left(2\,\%e^{3t}\cos(t-1)\sin(t-1)^2-2\,\%e^{t}\sin(t-1)\right)\right)/\left(\sqrt{-\%e^{2t}\cos(t-1)^2+5\,\%e^{2t}+1}\,\left(\right.\right.$$
$$\left(2\,\%e^{2t}\cos(t-1)^3+\left(-10\,\%e^{2t}-2\right)\cos(t-1)\right)\sin(t-1)+4\,\%e^{4t}\cos(t-1)^4+\left(\%e^{2t}-20\,\%e^{4t}\right)$$
$$\left.\left.\cos(t-1)^2-25\,\%e^{2t}-5\right)\right)\Bigg]$$

```
The following set of commands will plot the unit tangent, unit
normal, and unit binormal vectors as you move along the curve.
The resulting animated image will be saved in the wxMaxima
folder with the filename "13_m3_2.gif".
```

```
(%i49) p1: parametric(r[1],r[2],r[3],t,0,10)$
       numer: true$
       P: []$

       for i: 0 thru 10 step .5 do block(
          p2: vector(subst(i,t,r),subst(i,t,utan)),
          p3: vector(subst(i,t,r),subst(i,t,un)),
          p4: vector(subst(i,t,r),subst(i,t,ubn)),
          p5: points([
             subst(i,t,r),
             subst(i,t,r+0.2*ubn),
             subst(i,t,r+0.2*utan+0.2*ubn),
             subst(i,t,r+0.2*utan),
             subst(i,t,r),
             subst(i,t,r+0.2*un),
             subst(i,t,r+0.2*utan+0.2*un),
             subst(i,t,r+0.2*utan),
             subst(i,t,r+0.2*utan+0.2*ubn),
             subst(i,t,r+0.2*utan+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*utan+0.2*un),
             subst(i,t,r+0.2*utan+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*un),
             subst(i,t,r+0.2*ubn+0.2*un),
             subst(i,t,r+0.2*ubn)]),

          P: append(P,[gr3d(
             nticks=100,
             color=black,
             line_width=2,
             p1,
             color=red,
             line_width=3,
             p2,
             color=green,
             p3,
             color=blue,
             p4,
             points_joined=true,
             color=black,
             line_width=1,
             point_type=-1,
             p5,
             xrange=[-1,21],
             yrange=[-2,2],
             zrange=[-1,2],
             title="Tangent (Red), Unit Normal (Green), Unit Binormal (Blue)")]))$
```

```
(%i53) draw(
          terminal=animated_gif,
          delay=10,
          file_name="13_m3_2",
          P)$
```