Tour de TNB

Nicole Dong | Kyle Shi

AKA the dream team

Introduction

The Tour de France is a landmark road race across France that takes place every year, over the course of 3 weeks in July. The 2020 race stage will be an hour-long stretch starting at the Chateau de Cauchy village, and ending at the Chateau de Laplace village.

If changes in elevation are neglected, the proposed route can be (approximately) described by the following position vector (in km):

$$r(t) = \langle\ 2.9 \cos (3.2 \pi t) , (\sin (4 \pi t) + 5t) \rangle$$

One problem is that contestants often crash along dangerous sections of the race, such as tight corners, switchbacks, and traffic circles. To alleviate this problem, organizers plan to place hay bales on the most dangerous curves.

Organizers also plan to include a feed zone, which is a small stretch of roadway where racers are handed food which they consume on the go. Due to safety concerns, the feed zone should be placed in the “straightest” portion of the race.

Neither the hay bales nor the feed zone should be placed within 1 km of either village, since they will interfere with the spectators there.

In this project we will determine the most efficient locations to place the hay bales and the feed zone, so that their placements provide the most safety for the riders and also so that their placements do not interfere with the spectators.

Proposed Race Course

The race starts at the Chateau de Cauchy when t = 0 hours at (2.9,0), and ends at the Chateau de Laplace at t = 1 hours at (-2.346,5) *(see appendix under calculations)*.

The Pont de Pascal is a bridge that passes over a section of the route which the riders cross at times t = .1733184185 and .45168158145 hours; the location of the Pont de Pascal is (-.495,1.688) and the Pont de Pascal precedes the feed zone *(see appendix under calculations for finding the point of intersection)*.

This graph of the proposed route displays the position vector *r(t)* in kilometers plotted against time in hours:

$$r(t) = < 2.9 \cos(3.2\pi\ t)\ ,\ \sin(4\pi\ t) + 5t >$$

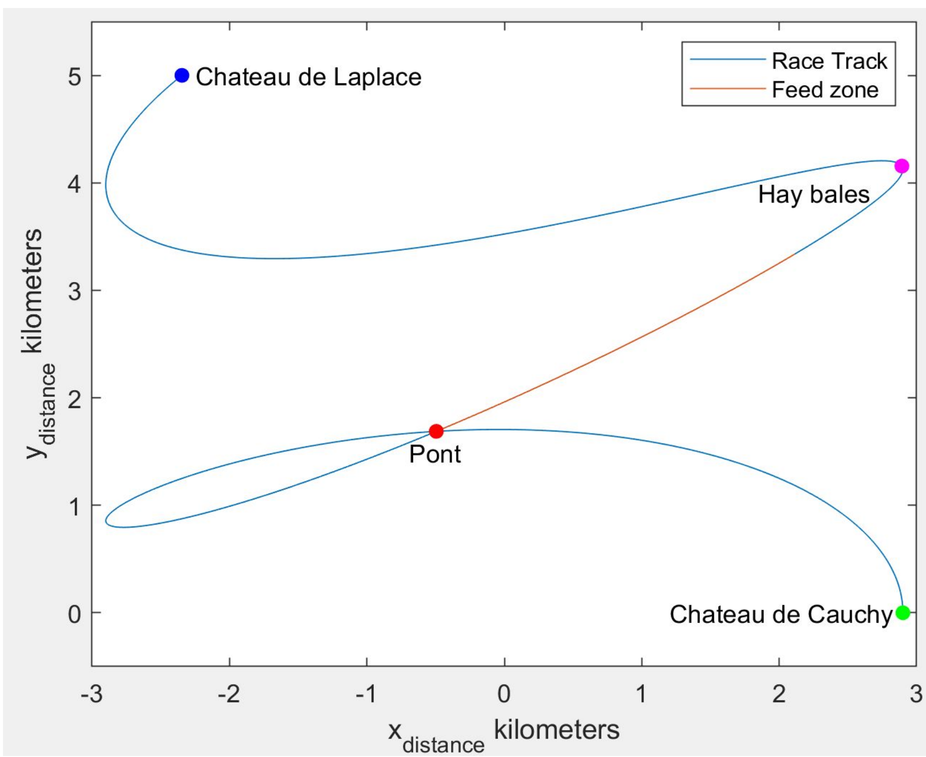

Figure 1: Graph of the proposed route r(t), see appendix

Although this is an acceptable approximation for the actual route and displays a rough map of the course, it is not entirely accurate. In reality, the riders also experience changes in elevation, which will cause the overall speed and distance that the riders travel to be different from the calculated value. This proposed route fails to take into account the change in Z direction that occurs, thus why there is no k component in the position vector r(t).

Without taking this component into account, it is not possible to know the exact value of the track length and the speed of the riders. Furthermore, without an exact description of the race route the calculations for the normal acceleration as well as curvature of the route will vary slightly.

Another important assumption made by the position vector for the course is that the current position vector r(t) assumes that all of the riders are traveling together at the same speed. Different riders racing along the race route will have distinct speeds that may or may not be the same.

Speed

The velocity vector *r'(t)* is found by taking the derivative of the position vector *r(t)*, and is as follows:

$$r'(t) = < -9.28\pi \sin(3.2\pi\, t)\ ,\ 4\pi \cos(4\pi\, t) + 5 >$$

The speed $||\, r'(t)\, ||$ is the magnitude of the velocity vector *r'(t)*, and is as follows:

$$||\, r'(t)\, || = \sqrt{[-\ 9.28\pi\, sin(3.2\pi\, t)]^2 + [4\pi\, cos(4\pi\, t)\ +\ 5]^2}$$

The average speed may be found by integrating the speed speed from 0 to 1 and dividing by the total time allotted *(see appendix, under calculations)*.

$$s_{avg} = \frac{\int_0^1 ||r'(t)dt||}{1-0} = 21.0979\ \frac{kilometers}{hour}$$

This graph of the speed along the proposed route displays the speed $||r'(t)||$ in kilometers plotted against time in hours:

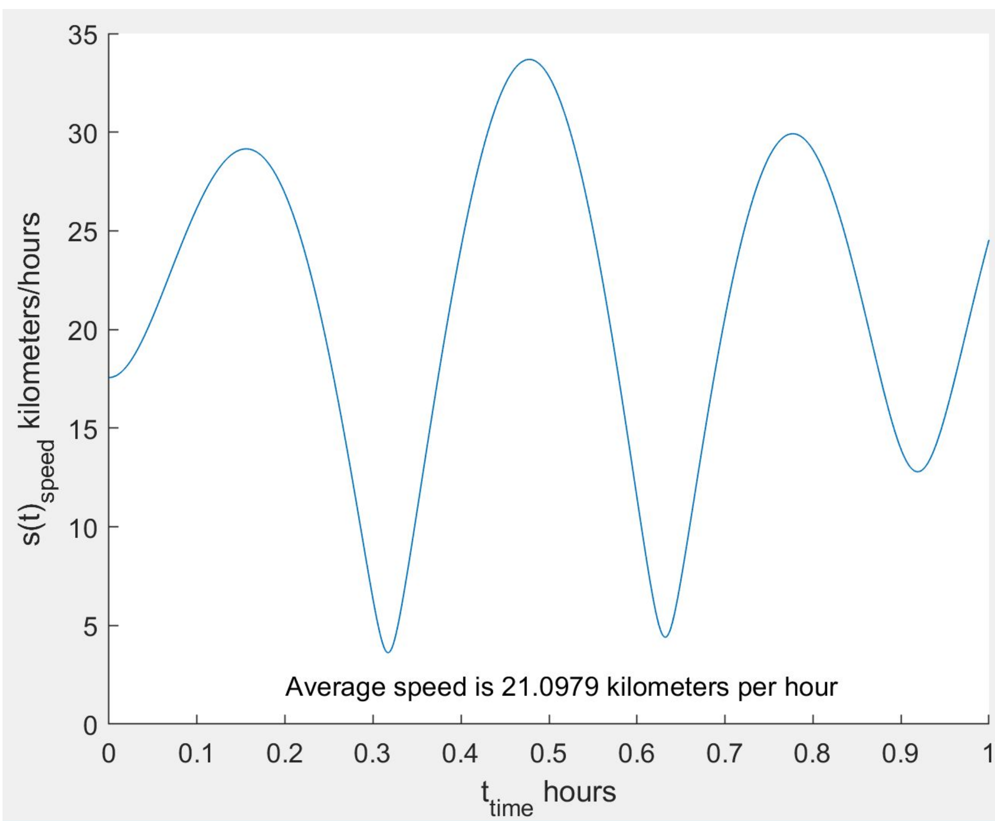

Figure 2: Graph of speed $||\, r'(t)\, ||$ against time t of the proposed route, see appendix

As stated in the graph, the average speed along this stretch of the route, from the Château de Cauchy to the Château de Laplace, is 21.0979 kilometers per hour. From this graph we are also able to see where along the route the riders are going fastest or slowest.

Length of Route

The length of the route that racers take can be calculated using the arc length formula:

$$s = \int_{t=0}^{t=1} || r'(t) || \, dt$$

where s is the length of the route that racers cover from the start of the race until the end, *r'(t)* is the velocity vector, and $|| r'(t) ||$ is the magnitude of the velocity vector, otherwise known as the speed.

Using this formula and Matlab, we determined that the length of the route is approximately 21.0979 kilometers *(see appendix under calculations)*.

Direct Distance

The direct distance between the two villages can be calculated by finding the coordinates of the Château de Cauchy, then finding the coordinates of the Chateau de Laplace, and finding the distance between them using the distance formula.

The coordinates of the Château de Cauchy are *(2.9, 0)* while the coordinates of the Chateau de Laplace are *(2.9cos(3.2π), 5)*. The coordinates of the Chateau de Laplace equate to roughly (-2.346, 5). Using the distance formula, we find that the direct distance between the villages is 7.2472 kilometers *(see appendix, under calculations)*.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Where (x_2, y_2) represents the Chateau de Laplace and (x_1, y_1) represents the Château de Cauchy

$$D = \sqrt{(-2.346 - 2.9)^2 + (5 - 0)^2}$$

D = 7.2472 kilometers

Minimum Distance From Villages

Organizers do not want to place either the set of hay bales or the feed zone too close to the two villages. Thus, we need to determine what times the riders are projected to be 1 kilometer away (along the route) from each of the villages. This is the minimum distance the riders must be from each village for us to place the hay bales or the feed zone.

In order to calculate what times the riders are projected to be 1 kilometer away, remember that the formula to calculate the distance covered along the route from time t_i to time t_f is as follows, where **s** represents the length of the track:

$$s = \int_{t=t_i}^{t=t_f} \| r'(t) \| \, dt$$

To find the time when riders are exactly 1 kilometer away from the Chateau de Cauchy, we solve for t_f in the following equation:

$$1 = \int_{t=0}^{t=t_f} \| r'(t) \| \, dt$$

$$t_f = .0532 \; hours$$

On the other hand, to find the time when riders are exactly 1 kilometer away from the Chateau de Laplace, we solve for t_i in the following equation:

$$1 = \int_{t=t_i}^{t=1} \| r'(t) \| \, dt$$

$$t_i = .9502 \; hours$$

Using this method, we obtained that riders are 1 kilometer away from the Chateau de Cauchy at time $t = 0.0532$ hours, and 1 kilometer away from the Chateau de Laplace at time $t = 0.9502$ hours. Therefore, the hay bales and feed zone may be placed anywhere within the time interval from .0532 hours to .9502 hours, which represent the points (2.495, .8858) and (-2.876, 4.165) on the race route, respectively *(see appendix, under calculations)*.

Curvature

Curvature is a measure of how fast the direction of a curve changes at a given point. It is calculated using the following equation:

$$k(t) = \frac{||r'(t) \times r''(t)||}{||r'(t)||^3}$$

We find the curvature for the race track to be *(see appendix, under calculations)*:

$$k(t) = \frac{||<0,0,148.48\pi^3 sin(4\ \pi t)sin(3.2\pi t)+118.784\pi^3 cos(3.2\pi t)cos(4\pi t)+148.48\pi^2 cos(3.2\pi t)>||}{[\ \sqrt{[-\ 9.28\pi\ sin(3.2\pi\ t)]^2+[4\pi\ cos(4\pi\ t)+5]^2}\ \ ]^3}$$

We proceed to plot the curvature *k(t)* against time *t*, and obtain the following graph:

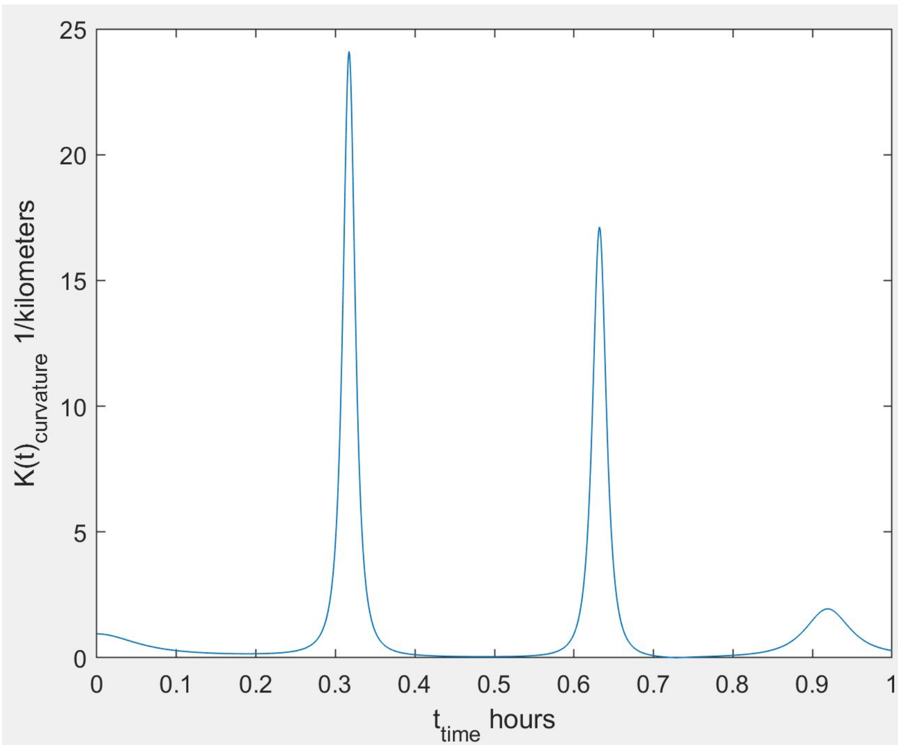

Figure 3: Graph of curvature k(t) against time t, see appendix

The relationship between the graph of curvature against time and the graph of position against time is as follows: When riders hit sharp turns in the position graph, the curvature increases rapidly; when the riders exit the turn, the curvature decreases rapidly; where the race track is straight there is little to no curvature.

This is what we expected; after all, curvature is a measure of how fast a curve changes at a given point.

On the other hand, the minimum points on the curvature graph correspond to the maximum points on the speed graph. This also makes sense. After all, the minimum points on the curvature graph are places during the race where the course is straighter; thus, riders are able to increase their speed without worry of falling. By contrast, riders will likely slow down around sharp turns in order to decrease the chance of injury.

Normal Component of Acceleration

The normal component of acceleration is a measure of the change in direction of velocity. On the other hand, the tangential component of acceleration measures the change in speed. It is calculated using the following equation:

$$N(t) = \frac{\|r'(t) \times r''(t)\|}{\|r'(t)\|}$$

We have already calculated the velocity vector *r'(t)*, the acceleration vector *r''(t),* the magnitude of the cross product of these two vectors, as well as the magnitude of the velocity vector . Thus, we can now write the normal component of acceleration as:

$$N(t) = \frac{\| <0,0,148.48\pi3 sin(4\ \pi t) sin(3.2\pi t)+118.784\pi3 cos(3.2\pi t) cos(4\pi t)+148.48\pi2 cos(3.2\pi t)> \|}{\sqrt{[-\,9.28\pi\ sin(3.2\pi\ t)]^2+[4\pi\ cos(4\pi\ t)+5]^2}}$$

We proceed to plot the normal component of acceleration *N(t)* against time *t*, and obtain the following graph:

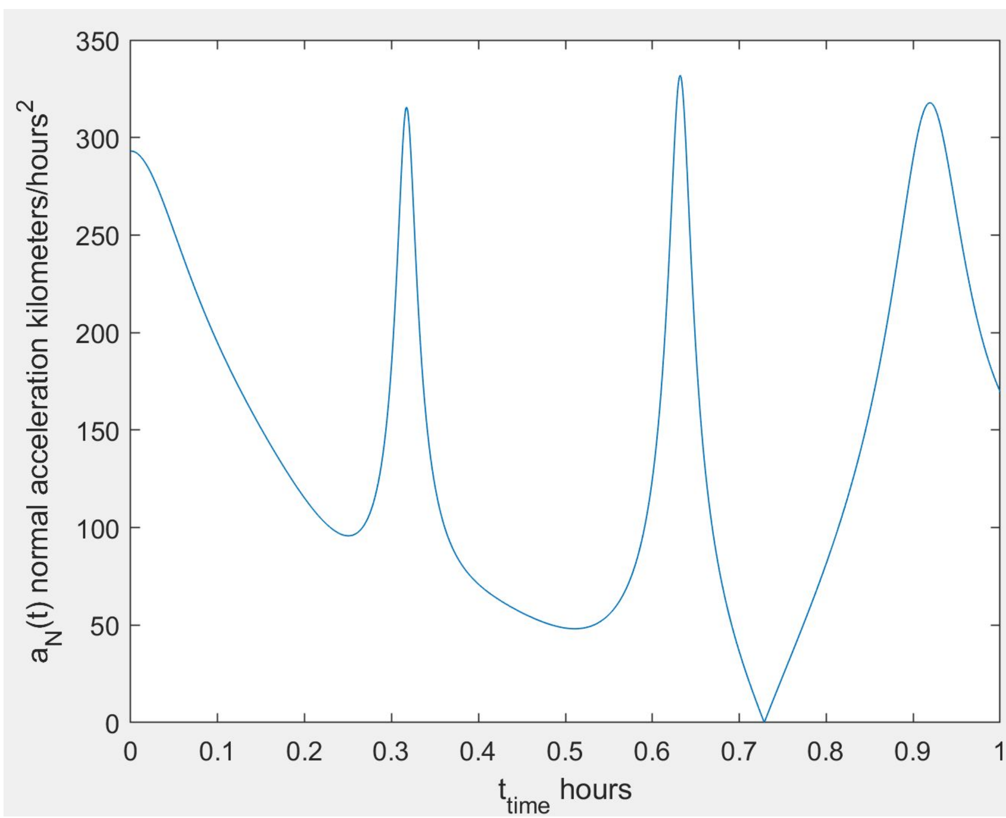

Figure 4: Graph of normal acceleration $a_n(t)$ against time t, see appendix

The maximum points on this graph correspond to the maximum points in the curvature graph. This is to be expected; after all, if a turn has a higher curvature, the direction change along that curve is greater. The highest peaks in the graph of normal acceleration align with the places where the path in the position graph contains the sharpest turns.

By contrast, these maximum points also correspond to the minimum points in the speed graph. This is also what we expected. As already stated, normal acceleration and curvature correspond, and places with highest curvature also have the lowest speed.

From the point of view of the riders, places with a greater normal acceleration will be more dangerous. The riders experience a more significant centripetal force, which increases their chances of falling off their bikes. When taking into account the speed of the riders as well as the high curvature at the point of maximum normal acceleration, we know that this point is where the riders are most likely to fall off their bikes. Thus, it makes the most sense to place the hay bales at the point where the normal acceleration is greatest.

It is important to note that the maximum normal component of acceleration does not occur at the same point on the course as the maximum curvature. This is because the curvature is calculated only based on the geometry of the course, while the normal acceleration also relies on the velocity of the riders. Thus, normal acceleration is a much more accurate representation of when riders will experience the most extreme changes in direction.

Placement of Hay Bales

Knowing the graph of normal acceleration as well as what it represents, we are able to determine an accurate placement for the hay bales.

Due to the fact that normal acceleration is a more accurate representation of when riders experience the most extreme change in direction, we will place hay bales at the point where normal acceleration is at a maximum.

Using the graph, we determined that the maximum normal component occurs at t = 0.632 hours by direct observation of the graph; the position here is at (2.8928, 4.1561) and the normal acceleration at this time is 332 km/hr^2. This is the point of maximum normal acceleration so we will place the set of hay bales here to ensure the safety of the riders. Their placement is plotted on Figure 1.

Placement of Feed Zone

We propose a feed zone on the part of the course from $t = 4.5$ to $t = 5.5$, which is from position (-.5434,1.6622) to P2(2.114,3.3378) *(see appendix, under calculations)*.

This is a good place to put a feed zone because the curvature is at its lowest. When the value of the curvature is low, the race track is straighter. Since there will be minimal bends and turns in the track, the riders can travel in a straight path; thus the riders will be able to safely grab food and focus on eating without having to worry about turning or irregularities in the race track. Placing the feed zone here will eliminate the risk of falling due to sharp turns. Also, the proposed feed zone allows the riders an entire 6 minutes to eat without having to maneuver any sharp turns. Another benefit is that this feed zone is more than 1 kilometer away from both villages, which will allow room for spectators to enjoy watching the race.

Shown below is the graph of curvature *k(t)* plotted against time, which has been amplified to display the interval of lowest curvature. As can be seen, the time period from $t = 4.5$ to $t = 5.5$ is when the curvature is at a minimum. The feed zone is plotted on Figure 1.

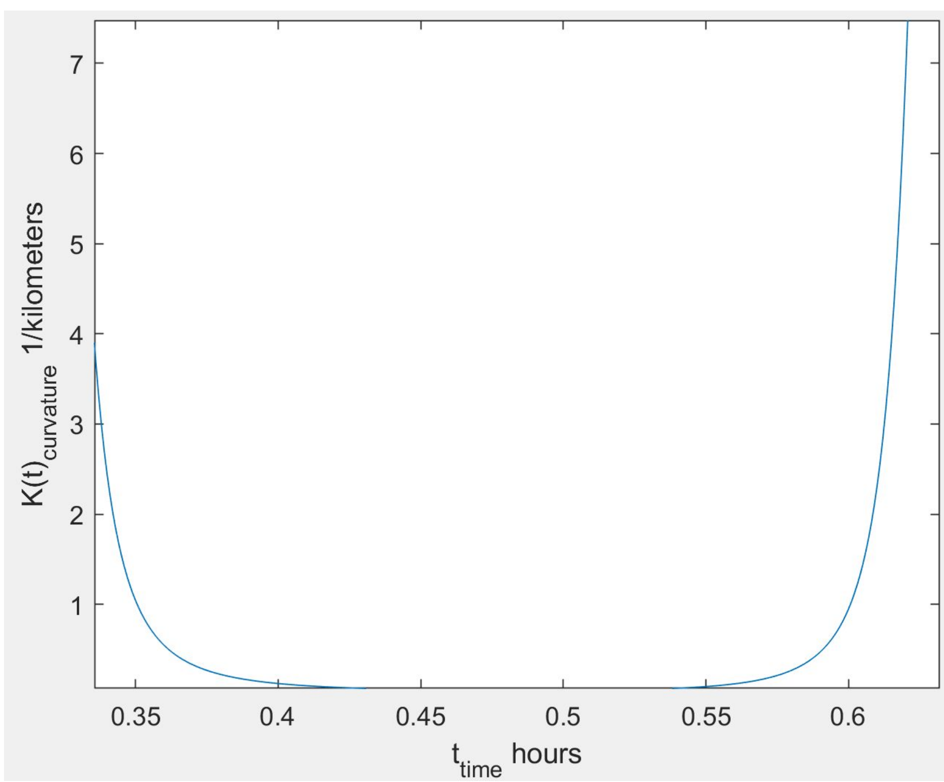

Figure 5: Amplified graph of curvature k(t) plotted against time t, NOTE: Code for figure 5 is the same as for figure 4

Final proposal

We propose that the feed zone be placed from (-.5434,1.6622) to (2.114,3.3378). The coordinates of these two positions on the race track correspond to the times t = .45 hours and t = .55 hours respectively. This area of the race track corresponds to the points of lowest curvature on the plot of curvature. Placing the feed zone in this area allows the riders a full 6 minutes to eat with minimal curvature and no interference with the spectators because the feed zone is more than 1 km away from both of the villages. We propose that the best position to place the hay bales is at t=.632 hours, the position (2.8928, 4.1561) on the race track. The riders reach a maximum normal acceleration of 332 $\frac{kilometers}{hours^2}$ here, meaning that they are most likely to fall here as opposed to other turns on the race track (all of our proposals may be seen on Figure 1).

APPENDIX

FIGURES

Figure 1

```
%this is for the position graph, t is the time and x and y are the i and j components of the position vector r
t = 0:.001:1;
x = 2.9*cos(3.2*pi*t);
y = sin(4*pi*t)+5*t;
figure(1)
plot(x,y);
xlabel('x_{distance} kilometers')
ylabel('y_{distance} kilometers')
ylim([-.5 5.5])
hold
% this is to create the plot of the feed zone (orange portion of the graph)
tz = .45:.001:.55;
xz = 2.9*cos(3.2*pi*tz);
yz = sin(4*pi*tz)+5*tz;
plot(xz,yz);
%This creates the legend for the race track as well as plots location of the villages, the bridge, and hay bales
legend({'Race Track', 'Feed zone'},'AutoUpdate','off');
text(2.9*cos(3.2*pi*(.1733184185))-.2,sin(4*pi*(.1733184185))+5*(.1733184185)-.2, 'Pont')
plot(2.9*cos(3.2*pi*(.1733184185)),sin(4*pi*(.1733184185))+5*(.1733184185),'.r','MarkerSize', 20)
text(1.2,0,'Chateau de Cauchy');
plot(2.9,0,'.g','MarkerSize', 20)
text(2.9*cos(3.2*pi)+.1,5,'Chateau de Laplace');
plot(2.9*cos(3.2*pi),5,'.b','MarkerSize', 20)
text(2.9*cos(3.2*pi*.632) - 1.05,sin(4*pi*(.632))+5*(.632) + -.25,'Hay bales');
plot(2.9*cos(3.2*pi*.632),sin(4*pi*(.632))+5*(.632),'.m','MarkerSize',20);
```

Figure 2

```
%this is for the speed vs time graph. xs is the magnitude of the velocity vector
xs = sqrt((-9.28*pi*sin(3.2*pi*t)).^2 + (4*pi*cos(4*pi*t)+5).^2);
figure(2)
hold
text(.2,2,'Average speed is 21.0979 kilometers per hour');
t = 0:.001:1;
plot(t,xs);
xs1 = @(t) sqrt((-9.28*pi*sin(3.2*pi*t)).^2 + (4*pi*cos(4*pi*t)+5).^2);
s = (1/(1-0))*integral(xs1,0,1);
xlabel('t_{time} hours')
```

```
ylabel('s(t)_{speed} kilometers/hours')
% the average speed is found to be 21.0979, (s is the average speed)
```

Figure 3

```
%These are the components needed to calculate the cross product (later used to find k(t) and normal acceleration)
%The cross product is r' x r" which is <0,0, x'y" - y'x">, x1 and x2 represent x' and x", y1 and y2 represent y' and y"
syms x1 x2 y1 y2;
x1 = -9.28*pi*sin(3.2*pi*t);
y1 = 4*pi*cos(4*pi*t)+5 ;
x2 = -29.696*pi^2*cos(3.2*pi*t);
y2 = -16*pi^2*sin(4*pi*t);

%plot of the curvature vs time, where K is the curvature figure(3)
%crossProduct represents the cross product of the first and second derivatives of the position vector r
%the crossProduct only has a k component so the vector would look like <0,0,(x1.*y2)- (y1.*x2)>
t = 0:.001:1;
crossProduct = (x1.*y2)- (y1.*x2);
magnitudeCrossProduct = sqrt((crossProduct).^2);
K = magnitudeCrossProduct./(xs.^3);
plot(t,K);
xlabel('t_{time} hours')
ylabel('K(t)_{curvature} 1/kilometers')
```

Figure 4

```
%plot of the normal component of acceleration, where n is the normal component of acceleration
figure(4)
t = 0:.001:1;
n = (magnitudeCrossProduct./xs);
plot(t,n);
xlabel('t_{time} hours')
ylabel('a_{N}(t) normal acceleration kilometers/hours^2')
```

Calculations

Finding points of Chateau de Cauchy and Château de Laplace

The Chateau de Cauchy is at t = 0, r(0) = (2.9 cos (3.2 π *0) , (sin (4 π *0) + 5*0))= *(2.9,0)*

The Chateau de Laplace is at t = 1, r(1) = (2.9 cos (3.2 π *1) , (sin (4 π *1) + 5*1) = (-2.346,5)

Finding the intersection/Pont de Pascal

```
syms t1 T;
[sols_t1, sols_T] = vpasolve([2.9*cos(3.2*pi*t1) == 2.9*cos(3.2*pi*T), sin(4*pi*T)+5*T == sin(4*pi*t1)+5*t1], [t1, T], [1/4, 1/2]);
%t1 = .1733184185 T = .45168158145
```

At time t = .1733184185, the point is (*2.9 cos(3.2π* *.1733184185*)* , *sin(4π* *.1733184185*)* + *5**.1733184185*),* which corresponds to (-.495,1.688).

Average speed

```
xs1 = @(t) sqrt((-9.28*pi*sin(3.2*pi*t)).^2 + (4*pi*cos(4*pi*t)+5).^2);
s = (1/(1-0))*integral(xs1,0,1);
```

Length of route

```
%the length of the course is the integral from 0 to 1 of the speed
l = integral(xs1,0,1); %where l is the length of the course
```

Direct distance between two villages

```
%direct distance between the two villages, found using distance formula
d = sqrt(((2.9*cos(3.2*pi))-2.9)^2 + (5-0)^2 + (0-0)^2);
%the distance is found to be 7.2472 kilometers
```

Times/Points where riders are 1km away from two villages

NOTE: MATLAB was unable to find an indefinite integral for the speed function so we found the times when the riders are 1km away from each of the villages using a computer science oriented approach. T1 = .0532 hours, T2 = .9502

At t = .0532, point is $(2.9\cos(3.2\pi *.0532), \sin(4\pi *.0532) + 5*.0532) =$ (2.495, .8858)

At t = .9502, point is $(2.9\cos(3.2\pi *.9502), \sin(4\pi *.9502) + 5*.9502) = (-2.876, 4.165)$

```
%Here a compsci oriented approach is used to solve for the times that the
%riders are 1 km away from the chateau de cauchy and the chateau de
%laplace
%we weren't able to find the times analytically using vpasolve
%because MATLAB isn't able to find an indefinite integral for the speed
%function


%This segment of code is used to find the time when the riders are 1km from
%chateau de cauchy
syms time1 %the independent variable
% the function we are trying to integrate
f = @(time1) sqrt( (-9.28*pi*sin(3.2*pi*time1)).^2 + (4*pi*cos(4*pi*time1) + 5).^2 );

% initialize variables
error = 0.1;
time = 0;
dist = 0;

% using a for-loop to find the value of the definite integral, for times t = 0 to t = 0.1
for time_temp = 0 : 0.0001 : 0.1

% evaluates the value of the definite integral, i.e. the distance, for the temp_time
dist_temp = integral(f,0.0,time_temp, 'RelTol',0,'AbsTol',1e-12);

   % calculates the error (how far away it is from 1) for the dist we just calculated
   error_temp = abs(dist_temp - 1);

   % if this temp_error is the smallest error we've found so far,
   if error_temp < error
      % then save this temp_error to the variable "error"
      error = error_temp;
      % update the time and dist (to the newer, more accurate values)
      time = time_temp;
      dist = dist_temp;
   end
end

% print the results
disp('time1 = ')
disp(time);
%the time for when the riders are 1km away from the chateau de cauchy is
%found to be .0532 hours
```

```
%this code segment is used to find when the riders are 1km away from
%chateau de laplace
syms time2 %the independent variable
% the function we are trying to integrate
f = @(time2) sqrt( (-9.28*pi*sin(3.2*pi*time2)).^2 + (4*pi*cos(4*pi*time2) + 5).^2 );

%initialize variables
error = 0.1;
time = 0;
dist = 0;

% using a for-loop to find the value of the definite integral, for times t = 0.9 to t = 1
for time_temp = 0.90 : 0.0001 : 1

% evaluates the value of the definite integral, i.e. the distance, for the temp_time
dist_temp = integral(f,time_temp,1, 'RelTol',0,'AbsTol',1e-12);

  % calculates the error (how far away it is from 1) for the dist we just calculated
  error_temp = abs(dist_temp - 1);

  % if this temp_error is the smallest error we've found so far,
  if error_temp < error
    % then save this temp_error to the variable "error"
    error = error_temp;
    % update the time and dist (to the newer, more accurate values)
    time = time_temp;
    dist = dist_temp;
  end
end

% print the results
disp('time2 = ')
disp(time);
%the second time is found to be .9502 hours
```

Curvature calculations

Curvature is a measure of how fast the direction of a curve changes at a given point. It is calculated using the following equation:

$$k(t) = \frac{||r'(t) \times r''(t)||}{||r'(t)||^3}$$

As we already know, the velocity vector *r'(t)* can be calculated by taking the derivative of the position vector *r(t)*. Furthermore, the acceleration vector *r''(t)* can be calculated by taking the derivative of the velocity vector *r'(t)*. Thus, we now have the following equations:

$$r'(t) = < -9.28\pi \sin(3.2\pi t) , 4\pi \cos(4\pi t) + 5 ,0>$$

$$r''(t) = < -29.696\pi^2 \cos(3.2\pi t) , -16\pi^2 \sin(4\pi t),0 >$$

$$r'(t) \times r''(t) = 0i + 0j + (-9.28\pi\ sin(3.2\pi\ t)(- 16\pi 2\ sin(4\pi\ t)) - (- 29.696\pi 2\ cos(3.2\pi\ t))(4\pi\ cos(4\pi\ t) + 5))k$$

$$r'(t) \times r''(t) = < 0, 0, 148.48\pi^3 sin(4\ \pi t)sin(3.2\pi t) + 118.784\pi^3 cos(3.2\pi t)cos(4\pi t) + 148.48\pi^2 cos(3.2\pi t) >$$

We have already calculated $|| r'(t) ||$, so we can now write the curvature *k(t)* as:

$$k(t) = \frac{|| <0,0,148.48\pi^3 sin(4\ \pi t)sin(3.2\pi t)+118.784\pi^3 cos(3.2\pi t)cos(4\pi t)+148.48\pi^2 cos(3.2\pi t)> ||}{[\ \sqrt{[- 9.28\pi\ sin(3.2\pi\ t)]^2+[4\pi\ cos(4\pi\ t) + 5]^2}\ \ ]^3}$$

Calculations for points of the feed zone and hay bales

Point 1 of feed zone at t = .45 hours

```
>> 2.9*cos(3.2*pi*(.45))

ans =

   -0.5434

>> sin(4*pi*.45)+5*(.45)

ans =

    1.6622
```

Point 2 of feed zone at t = .55 hours

```
>> 2.9*cos(3.2*pi*(.55))

ans =

    2.1140

>> sin(4*pi*.55)+5*(.55)

ans =

    3.3378
```

Point of hay bale placement at t = .632 hours

```
>> 2.9*cos(3.2*pi*(.632))

ans =

    2.8928

>> sin(4*pi*.632)+5*(.632)

ans =

    4.1561
```

FULL CODE

```
close all

syms t1 T; %this finds where the graph intersects itself, otherwise known as the location of the pont
[sols_t1, sols_T] = vpasolve([2.9*cos(3.2*pi*t1) == 2.9*cos(3.2*pi*T), sin(4*pi*T)+5*T == sin(4*pi*t1)+5*t1], [t1, T], [1/4, 1/2]);
%t1 = .1733184185 T = .45168158145

%this is for the position graph, t is the time and x and y are the
%i and j components of the position vector r respectively
t = 0:.001:1;
x = 2.9*cos(3.2*pi*t);
y = sin(4*pi*t)+5*t;
figure(1)
plot(x,y);
xlabel('x_{distance} kilometers')
ylabel('y_{distance} kilometers')
ylim([-.5 5.5])
hold

% this is to create the plot of the feed zone (orange portion of the graph)
tz = .45:.001:.55;
xz = 2.9*cos(3.2*pi*tz);
yz = sin(4*pi*tz)+5*tz;
plot(xz,yz);

%This creates the legend for the race track as well as plots location of the villages,
%the bridge, and lastly the hay bales on the race track
legend({'Race Track', 'Feed zone'},'AutoUpdate','off');
text(2.9*cos(3.2*pi*(.1733184185))-.2,sin(4*pi*(.1733184185))+5*(.1733184185)-.2, 'Pont')
plot(2.9*cos(3.2*pi*(.1733184185)),sin(4*pi*(.1733184185))+5*(.1733184185),'.r','MarkerSize', 20)
text(1.2,0,'Chateau de Cauchy');
plot(2.9,0,'.g','MarkerSize', 20)
text(2.9*cos(3.2*pi)+.1,5,'Chateau de Laplace');
plot(2.9*cos(3.2*pi),5,'.b','MarkerSize', 20)
text(2.9*cos(3.2*pi*.632) - 1.05,sin(4*pi*(.632))+5*(.632) + -.25,'Hay bales');
plot(2.9*cos(3.2*pi*.632),sin(4*pi*(.632))+5*(.632),'.m','MarkerSize',20);

%this is for the speed vs time graph. xs is the magnitude of the velocity
%vector
xs = sqrt((-9.28*pi*sin(3.2*pi*t)).^2 + (4*pi*cos(4*pi*t)+5).^2);
figure(2)
hold
text(.2,2,'Average speed is 21.0979 kilometers per hour');
t = 0:.001:1;
plot(t,xs);
%next two lines calculate average speed
xs1 = @(t) sqrt((-9.28*pi*sin(3.2*pi*t)).^2 + (4*pi*cos(4*pi*t)+5).^2);
s = (1/(1-0))*integral(xs1,0,1);
xlabel('t_{time} hours')
ylabel('s(t)_{speed} kilometers/hours')
% the average speed is found to be 21.0979, (s is the average speed)

%direct distance between the two villages, found using distance formula
d = sqrt(((2.9*cos(3.2*pi))-2.9)^2 + (5-0)^2 + (0-0)^2);
%the distance is found to be 7.2472 kilometers
%the length of the course is the integral from 0 to 1 of the speed
l = integral(xs1,0,1); %where l is the length of the course


%Here a compsci oriented approach is used to solve for the times that the
%riders are 1 km away from the chateau de cauchy and the chateau de
%laplace
%we weren't able to find the times analytically using vpasolve
%because MATLAB isn't able to find an indefinite integral for the speed
%function
```

```
%This segment of code is used to find the time when the riders are 1km from
%chateau de cauchy
syms time1 %the independent variable
% the function we are trying to integrate
f = @(time1) sqrt( (-9.28*pi*sin(3.2*pi*time1)).^2 + (4*pi*cos(4*pi*time1) + 5).^2 );

% initialize variables
error = 0.1;
time = 0;
dist = 0;

% using a for-loop to find the value of the definite integral, for times t = 0 to t = 0.1
for time_temp = 0 : 0.0001 : 0.1

% evaluates the value of the definite integral, i.e. the distance, for the temp_time
dist_temp = integral(f,0.0,time_temp, 'RelTol',0,'AbsTol',1e-12);

  % calculates the error (how far away it is from 1) for the dist we just calculated
  error_temp = abs(dist_temp - 1);

  % if this temp_error is the smallest error we've found so far,
  if error_temp < error
    % then save this temp_error to the variable "error"
    error = error_temp;
    % update the time and dist (to the newer, more accurate values)
    time = time_temp;
    dist = dist_temp;
  end
end

% print the results
disp('time1 = ')
disp(time);
%the time for when the riders are 1km away from the chateau de cauchy is
%found to be .0532 hours


%this code segment is used to find when the riders are 1km away from
%chateau de laplace
syms time2 %the independent variable
% the function we are trying to integrate
f = @(time2) sqrt( (-9.28*pi*sin(3.2*pi*time2)).^2 + (4*pi*cos(4*pi*time2) + 5).^2 );

%initialize variables
error = 0.1;
time = 0;
dist = 0;

% using a for-loop to find the value of the definite integral, for times t = 0.9 to t = 1
for time_temp = 0.90 : 0.0001 : 1

% evaluates the value of the definite integral, i.e. the distance, for the temp_time
dist_temp = integral(f,time_temp,1, 'RelTol',0,'AbsTol',1e-12);

  % calculates the error (how far away it is from 1) for the dist we just calculated
  error_temp = abs(dist_temp - 1);

  % if this temp_error is the smallest error we've found so far,
  if error_temp < error
    % then save this temp_error to the variable "error"
    error = error_temp;
    % update the time and dist (to the newer, more accurate values)
    time = time_temp;
    dist = dist_temp;
  end
end
```

```
% print the results
disp('time2 = ')
disp(time);
%the second time is found to be .9502 hours


%These are the components needed to calculate the cross product (which is
%later used to find the curvature and the normal acceleration)
%The cross product is r' X r'' which is 0i + 0j + (x'y'' - y'x'')k, x1 and x2 represent x' and
%x'' respectively, y1 and y2 represent y' and y'' respectively
syms x1 x2 y1 y2;
x1 = -9.28*pi*sin(3.2*pi*t);
y1 = 4*pi*cos(4*pi*t)+5 ;
x2 = -29.696*pi^2*cos(3.2*pi*t);
y2 = -16*pi^2*sin(4*pi*t);

%plot of the curvature vs time, where K is the curvature
figure(3)
%crossProduct represents the cross product of the first and second derivatives of
%the position vector r
%the crossProduct only has a k component so the vector would look like
%<0,0,(x1.*y2)- (y1.*x2)>
t = 0:.001:1;
crossProduct = (x1.*y2)- (y1.*x2);
magnitudeCrossProduct = sqrt((crossProduct).^2);
K = magnitudeCrossProduct./(xs.^3);
plot(t,K);
xlabel('t_{time} hours')
ylabel('K(t)_{curvature} 1/kilometers')


%plot of the normal component of acceleration, where n is the normal
%component of acceleration
figure(4)
t = 0:.001:1;
n = (magnitudeCrossProduct./xs);
plot(t,n);
xlabel('t_{time} hours')
ylabel('a_{N}(t) normal acceleration kilometers/hours^2')
```