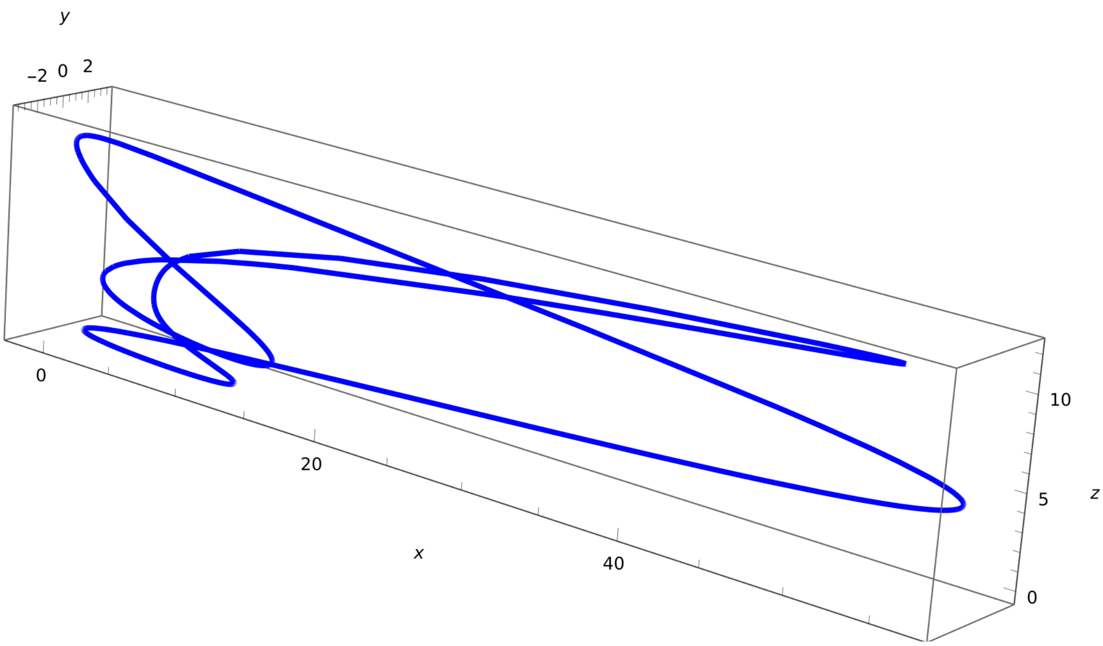
Template for Computer Project 2

1) Plotting Space Curves

We first define the given space curve as a function we can use repeatedly. Note that vectors are typically defined using curly braces in Mathematica.
We use ParametricPlot3D to plot the curve. Note that we also store the plot in a variable so we can use it later as well!

In[1]:=
```
r[t_] := {(3 Cos[3 t] + 5 Sin[t])^2, 3 Cos[3 t] - Cos[t], (Sin[2 t] + 3 Sin[t])^2}
p1 = ParametricPlot3D[r[t], {t, 0, 2 Pi}, AxesLabel → {x, y, z}, PlotStyle → Blue]
```

Out[2]=



2) Arc Length of a Curve

Mathematica has a built-in command for computing arc length.

In[3]:=
```
ArcLength[r[t], {t, 0, 2 Pi}]
```

Out[3]=
```
$Aborted
```

For simpler curves, this would give the exact answer rather quickly. But this curve is complicated enough that Mathematica is having a difficult time producing the exact answer. To remedy this, we pass an argument to ArcLength that tells Mathematica we are happy with an approximation.

In[4]:=
```
ArcLength[r[t], {t, 0, 2 Pi}, WorkingPrecision → 10]
```

Out[4]=
```
278.7806979
```

The value of 10 for WorkingPrecision tells Mathematica we want 10 significant figures. We could have asked for far more, but this is more than precise enough for our purposes!

3) Curvature

Mathematica also has a built-in method for computing curvature.

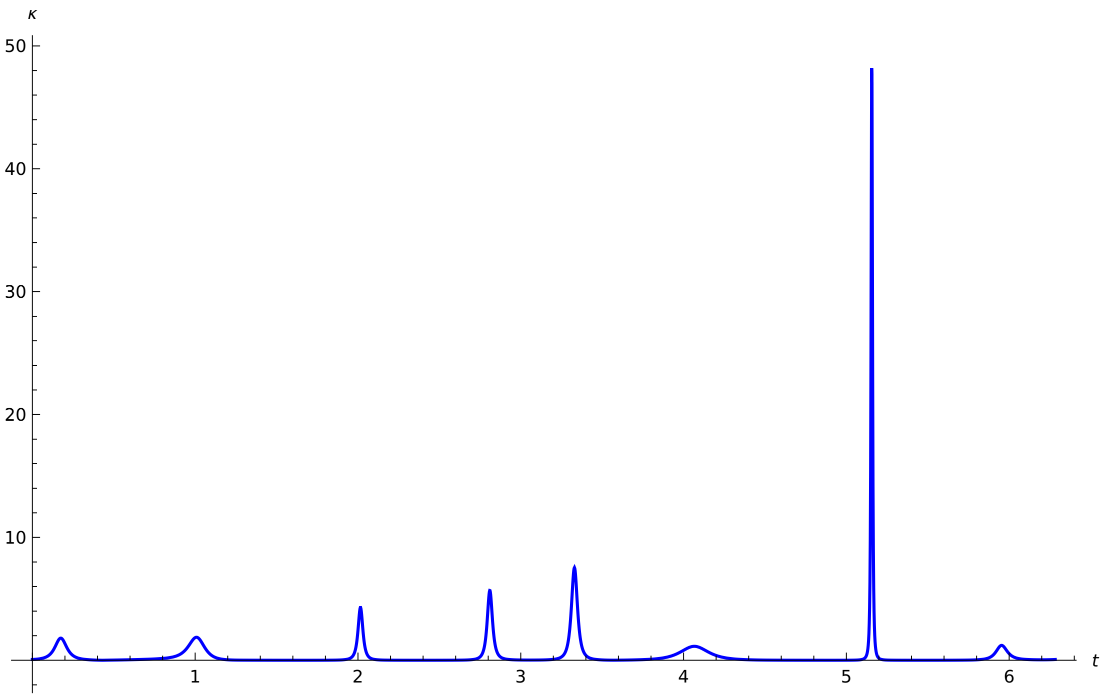
In[5]:=
```
curv = FullSimplify[ArcCurvature[r[t], t]]
```

Out[5]=
```
Sqrt[((20 964 240 Cos[t] + 8 711 694 Cos[2 t] + 24 438 768 Cos[3 t] -
      17 857 332 Cos[4 t] - 30 602 448 Cos[5 t] - 19 034 321 Cos[6 t] - 15 594 408 Cos[7 t] -
      16 829 370 Cos[8 t] + 4 428 552 Cos[9 t] + 4 145 535 Cos[10 t] + 3 239 712 Cos[11 t] +
      12 184 100 Cos[12 t] - 6 (-7 526 179 - 521 316 Cos[13 t] + 188 703 Cos[14 t] -
         78 408 Cos[15 t] + 29 844 Cos[16 t] + 221 616 Cos[17 t] + 212 139 Cos[18 t] +
         26 244 Cos[19 t] + 1944 Cos[20 t] + 509 400 Sin[t] + 9 742 260 Sin[2 t] +
         4 248 000 Sin[3 t] + 3 504 480 Sin[4 t] + 4 864 320 Sin[5 t] + 288 940 Sin[6 t] -
         3 527 640 Sin[7 t] - 1 275 360 Sin[8 t] - 1 312 920 Sin[9 t] -
         3 947 220 Sin[10 t] - 518 400 Sin[11 t] + 506 560 Sin[12 t] - 195 840 Sin[13 t] +
         247 860 Sin[14 t] + 456 840 Sin[15 t] + 305 640 Sin[16 t] + 19 440 Sin[17 t])) /
   (6111 + 144 Cos[t] - 3646 Cos[2 t] + 42 Cos[3 t] - 1084 Cos[4 t] - 150 Cos[5 t] -
      3798 Cos[6 t] - 36 Cos[7 t] + 4946 Cos[8 t] - 729 Cos[12 t] - 6240 Sin[2 t] +
      120 Sin[4 t] + 3000 Sin[6 t] + 1620 Sin[8 t] - 3240 Sin[10 t])^3)]
```

Even after simplification (using the FullSimplify command), the curvature is horrendous! Let's plot it.

In[6]:=
```
Plot[curv, {t, 0, 2 Pi}, PlotRange → Full, AxesLabel → {t, κ}, PlotStyle → Blue]
```

Out[6]=



Clearly, the point of maximum curvature is somewhere in the range $5.1 < t < 5.2$.

4) Identifying the Point of Maximum Curvature

The curvature function is complicated enough that even Mathematica will have trouble identifying the exact maximum. Instead, we rely on a numeric root finding procedure.

In[7]:=
```
maxCurvT = FindRoot[D[curv, t], {t, 5.1, 5.2}, Method → "Brent", WorkingPrecision → 10]
```

Out[7]=
```
{t → 5.156133915}
```

The numeric method we're using is a procedure developed by Richard Brent in 1973. To get the point of maximum on the graph, we can use the replacement rule returned by FindRoot.

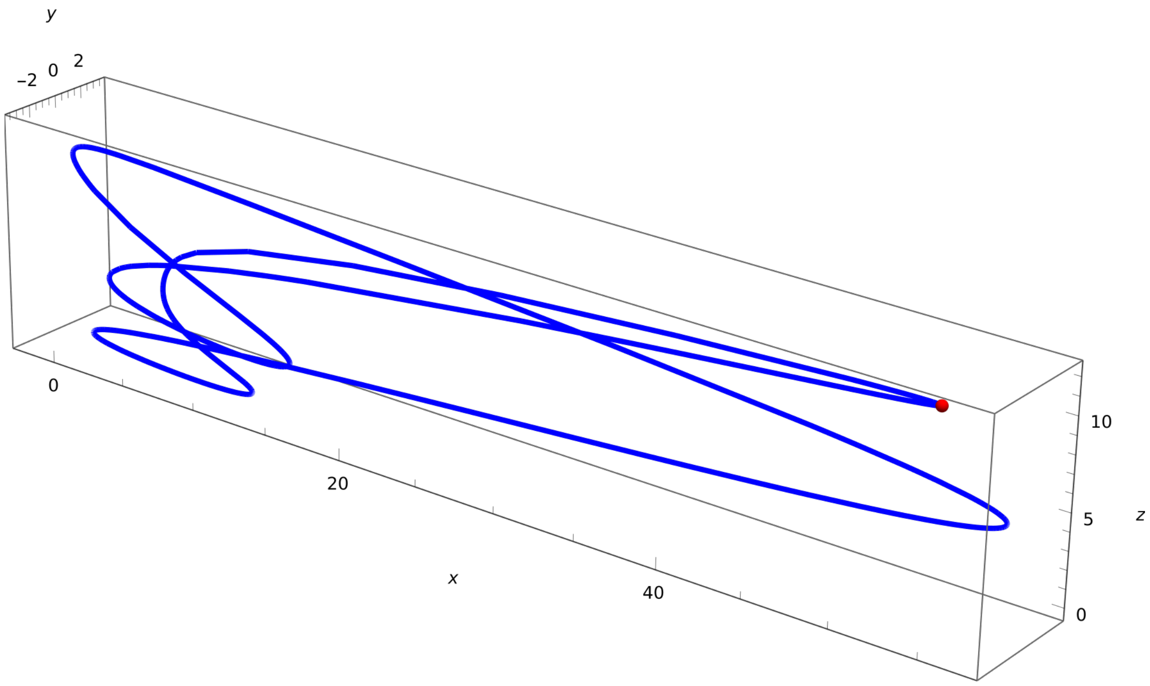
In[8]:=
```
maxCurvPt = r[t] /. maxCurvT
```

Out[8]=
```
{55.2060496, -3.34365103, 12.14480692}
```

We can display this point on the graph by plotting a tiny sphere centered at the point above, and using the Show command to overlay the plot of the space curve with the sphere. Note that the order of plot in Show is important! Show will us the first plot to determine window size, axes labels, etc. There is a way to override this behavior, but in general you should list the main plot that you want to overlay things on first.

In[9]:=
```
p2 = Graphics3D[{Red, Sphere[maxCurvPt, 0.3]}];
Show[p1, p2]
```

Out[10]=



Certainly the point we've marked looks like the point of highest curvature. You should be careful though. If the axes are scaled very differently, the curvature at various points may actually be larger or smaller than what is shown on the graph.