Curvature and the Osculating Circle

by Susan Wildstrom and Peggy Craft

```
> restart;
```

In single variable calculus, students learn that the first and second derivatives of a function help them to see the behavior of the graph of the function (increasing, decreasing, concavity, local maximum and minimum points as well as points of inflection). In multivariable calculus, even for functions whose graphs are just planar curves, the notion of the curvature emerges as a way of measuring the rate at which a curve "bends".

A Review of Parametric notation for specifying lines

Students learn to use $f(x) = \mathit{expression}$ **in** x notation in their early mathematics classes. Students sometimes learn to write parametric equations in upper level courses. Here we will present some methods for writing parametric equations for lines. Suppose that we are given two points (-5,2) and (3,6). To write parametric equations for the line through these two points we need to make a couple of decisions. Which of the two points will we "start" at, and which direction (toward or away from the other point) will we consider to be the positive direction of the line. As a general policy, we will usually consider the direction from our starting point toward the second point to be the positive direction. We will also want to decide how we wish to "calibrate" movement along the line. It is often convenient (if not computationally simplest) to calibrate in such a way that the value of the parameter, d, will represent an explicit distance to be traveled along the line. To that end we need to determine how much of each unit travelled along the line represents motion in a horizontal direction and how much represents motion in a vertical direction. Knowing the actual distance between the two given points and computing the horizontal and vertical changes as we move from the starting point to the second point enables us to determine values (called the directional cosines) that represent the horizontal and vertical components of change per single unit of distance travelled. The following Maple code will produce the parametric equations for the line through (-5,2) and (3,6) using (-5,2) as the starting point and positively directing the line toward (3,6).

```
> x0 := -5; y0 := 2; x1 := 3; y1 := 6; c1 := x1-x0; c2 := y1-
  y0; lambda := sqrt(c1^2+c2^2); cos1 := c1/lambda; cos2 :=
  c2/lambda; x := x0+cos1*d; y := y0+cos2*d;
```

$$x0 := -5$$

$$y0 := 2$$

$$x1 := 3$$

$$y1 := 6$$

$$c1 := 8$$

$$c2 := 4$$

$$\lambda := 4\sqrt{5}$$

$$cos1 := \frac{2}{5}\sqrt{5}$$

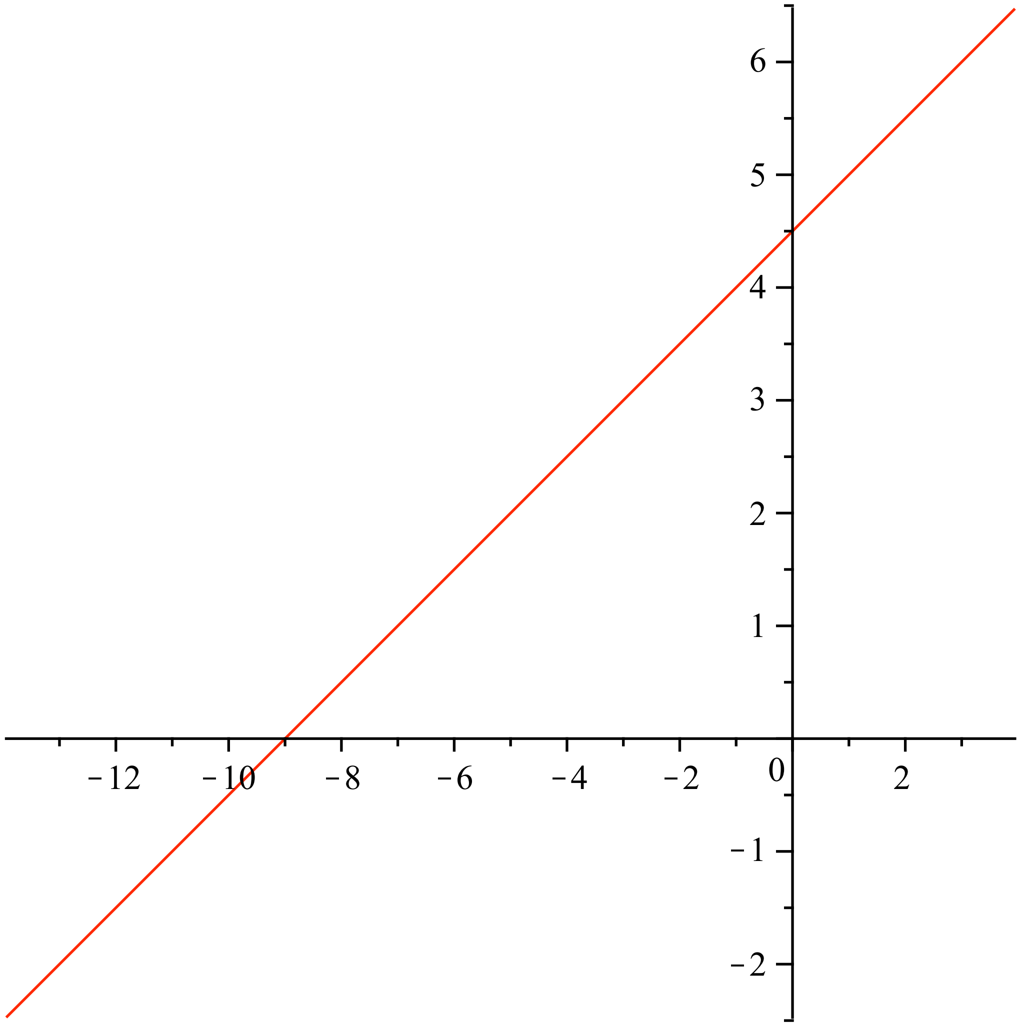
$$cos2 := \frac{1}{5}\sqrt{5}$$

$$x := -5 + \frac{2}{5}\sqrt{5}\,d$$

$$y := 2 + \frac{1}{5}\sqrt{5}\,d \qquad \textbf{(1.1)}$$

To see what the graph of this line looks like, we need to specify not only the function to be plotted but also the range of values we want the parameter to include. Starting our range of d values with 0 will cause the line to begin at (-5,2) and move toward (3,6). The length of the line we draw will be determined by the difference between the starting and ending values of d. Using negative values of d will draw a graph of the line that will include points "before" the starting point of (-5,2). Substituting explicit values of d into the parametric equations we have written will calculate coordinate pairs for points on the line at that particular distance from the starting point (-5,2).

```
> with(plots) : plot([x, y, d=-10..10]);
```



```
>
```

Exercise:

1) Using as the starting point, (number of the month of your birthday, units digit of the date of your birthday) and as the second point (first digit of your telephone number, second digit of your phone

number), write (first by hand if you are able, and then using Maple to check if you are correct) the parametric equations for the line through your pair of points. Graph your line as well.

```
>
```

Perpendicular lines and more

We recall that lines that are perpendicular to one another have slopes that are negative reciprocals of one another. Looking at the parametric equations for a line, is there a way to know the slope of the line (without going back to the slope formula from rectangular form)? If you do not immediately see where the parametric equations contain the slope, compute the slope from the rectangular formula and investigate further.

```
> slope := (y1-y0)/(x1-x0); cos2/cos1;
```

$$slope := \frac{1}{2}$$

$$\frac{1}{2} \tag{2.1}$$

So a quick way to determine the slope of a line perpendicular to a given line is to write the negative reciprocal of the slope of the original line. Since the components of the slope are contained in the directional cosines of the parametric equations, a line perpendicular to the given line will need to have "new" directional cosines that represent the negative reciprocal. How can we write new parametric equations for the line through the same starting point but perpendicular to the original line? Do you need to find an actual point on the perpendicular line or is there a way to do this process without having another point?

```
> perpslope:=-cos1/cos2;
```

$$perpslope := -2 \tag{2.2}$$

```
> xperp1:=x0+cos2*d; yperp1:=y0-cos1*d;
```

$$xperp1 := -5 + \frac{1}{5}\sqrt{5}\, d$$

$$yperp1 := 2 - \frac{2}{5}\sqrt{5}\, d \tag{2.3}$$

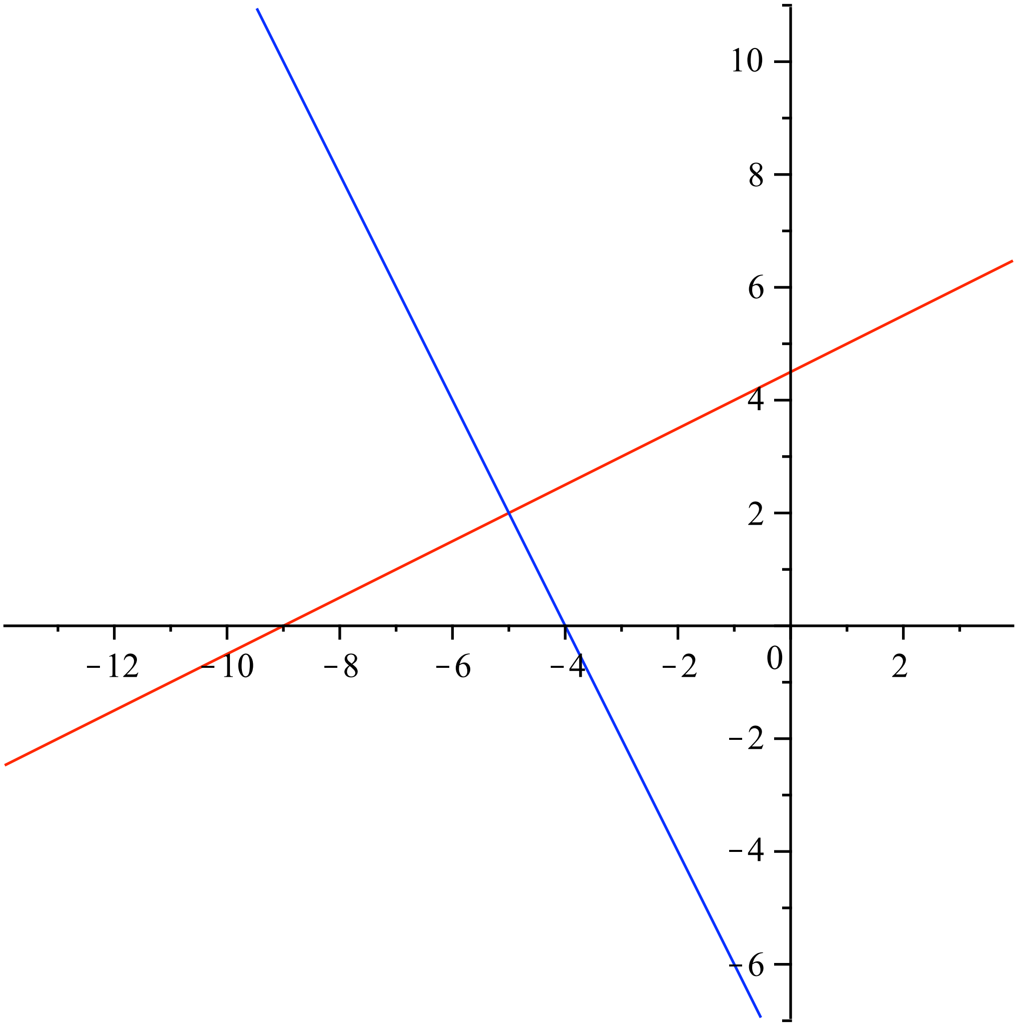
```
> xperp2:=x0-cos2*d; yperp2:=y0+cos1*d;
```

$$xperp2 := -5 - \frac{1}{5}\sqrt{5}\, d$$

$$yperp2 := 2 + \frac{2}{5}\sqrt{5}\, d \tag{2.4}$$

What is the difference between the two sets of equations for the line perpendicular to the given line?

```
> plot({[x, y, d = -10 .. 10],[xperp1,yperp1,d=-10..10]},
  color=[red,blue]);
```



Suppose you wanted to write parametric equations for a line perpendicular to the given line but through a point not on the original line at all? Suppose you wanted to write parametric equations for a line parallel to the original line and through another given point? Experiment with the code above to discover techniques you could use for parallel and perpendicular lines.

Exercise:

2) Write parametric equations for the line through (7, -2) and parallel to the first line we examined.

```
>
```

3) Write parametric equations for the line through (2, -5) and perpendicular to the first line.

```
>
```

4) Graph the two lines you just created. Make the first line show up in blue and the second one in red.

```
>
>
```

Developing Curvature

Consider an angle θ formed between $T(s)$ and i (the unit vector along the positive x-axis). Note then that $\theta = \cos^{-1}(T(s) \cdot \mathrm{i})$. Realize that since the magnitudes of both $T(s)$ and i are 1, there are no denominators to worry about). And now θ will be a function of s (arclength), ie θ is a function of $T(s)$ so this is a composite function but in s. So, $\theta = f(T(s))$

Let $\frac{d\theta}{ds}$ be our measure of how much the curve, C, bends at points along it. We define the curvature, K, of C at $P(x, y)$ as $K = \left|\frac{d\theta}{ds}\right|$

We now can harness the power of Maple to derive the formulas for curvature. First let's assume we have a function in the form $y = f(x)$. Recall that the slope of the tangent line $y' = \tan(\theta)$, so $\theta = \tan^{-1} y'$.

As for the ds computation we recall that $s = \int_a^x \sqrt{1 + (y')^2}\, \mathrm{d}x$

It is important to realize that both θ and s are functions of x, and so we will be using the chain rule to compute our formula for curvature

$\frac{d\theta}{ds} = \frac{\frac{d\theta}{dx}}{\frac{ds}{dx}}$ so computing those two parts separately and then combining and simplifying the results should get us a usable formula.

```
> #Differentiate theta:
  restart;
  y:=f(x);
  theta:=arctan(Diff(y,x)):
  diff(theta,x);
```

$$y := f(x)$$

$$\frac{\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}}{\mathrm{d}x} f(x)\right)}{1 + \left(\frac{\mathrm{d}}{\mathrm{d}x} f(x)\right)^2} \qquad (3.1)$$

```
> #Differentiate s:
  s:=int(sqrt(1+(diff(y,x))^2),x=a..x);
  diff(s,x);
```

$$s := \int_a^x \sqrt{1 + \left(\frac{\mathrm{d}}{\mathrm{d}x} f(x)\right)^2}\, \mathrm{d}x$$

$$\sqrt{1 + \left(\frac{\mathrm{d}}{\mathrm{d}x} f(x)\right)^2} \qquad (3.2)$$

```
> #Put them together:
  K:=abs(diff(theta,x)/diff(s,x));
```

(3.3)

$$K := \left| \frac{\frac{d}{dx}\left(\frac{d}{dx} f(x)\right)}{\left(1+\left(\frac{d}{dx} f(x)\right)^2\right)\sqrt{1+\left(\frac{d}{dx} f(x)\right)^2}} \right| \tag{3.3}$$

So in simpler terms we see that the curvature for a function expressed in standard function notation is $K = \left| \frac{y''}{\left(1+(y')^2\right)^{\frac{3}{2}}} \right|$

So, we shall now tackle the case of a function defined parametrically.

$x = f(t)$ and $y = g(t)$. Again we will start with the fact that $\theta = \tan^{-1}\left(\frac{g}{f}\right)'$ and

$s = \int_a^x \sqrt{(f')^2 + (g')^2}$ dx and we will again use the chain rule as before $\frac{d\theta}{ds} = \frac{\frac{d\theta}{dt}}{\frac{ds}{dt}}$

```
> #Differentiate theta:
  x:=f(t); y:=g(t);
  c:=(diff(y,t)/diff(x,t));

  theta:=arctan(c):
  diff(theta,t);
```

$$x := f(t)$$

$$y := g(t)$$

$$c := \frac{\frac{d}{dt} g(t)}{\frac{d}{dt} f(t)}$$

$$\frac{\frac{\frac{d^2}{dt^2} g(t)}{\frac{d}{dt} f(t)} - \frac{\left(\frac{d}{dt} g(t)\right)\left(\frac{d^2}{dt^2} f(t)\right)}{\left(\frac{d}{dt} f(t)\right)^2}}{1 + \frac{\left(\frac{d}{dt} g(t)\right)^2}{\left(\frac{d}{dt} f(t)\right)^2}} \tag{3.4}$$

```
> #Differentiate s:
  s:=int(sqrt((diff(x,t))^2+(diff(y,t))^2),t=a..t);
  diff(s,t);
```

$$s := \int_a^t \sqrt{\left(\frac{d}{dt} f(t)\right)^2 + \left(\frac{d}{dt} g(t)\right)^2}\, dt$$

$$\sqrt{\left(\frac{d}{dt} f(t)\right)^2 + \left(\frac{d}{dt} g(t)\right)^2} \tag{3.5}$$

```
> #Put them together:
  K:=abs(diff(theta,t)/diff(s,t));
```

$$K := \left| \frac{\dfrac{\frac{d^2}{dt^2} g(t)}{\frac{d}{dt} f(t)} - \dfrac{\left(\frac{d}{dt} g(t)\right)\left(\frac{d^2}{dt^2} f(t)\right)}{\left(\frac{d}{dt} f(t)\right)^2}}{\left(1 + \dfrac{\left(\frac{d}{dt} g(t)\right)^2}{\left(\frac{d}{dt} f(t)\right)^2}\right)\sqrt{\left(\frac{d}{dt} f(t)\right)^2 + \left(\frac{d}{dt} g(t)\right)^2}} \right| \tag{3.6}$$

```
> simplify( (3.6) );
```

$$\left| \frac{\left(\frac{d^2}{dt^2} g(t)\right)\left(\frac{d}{dt} f(t)\right) - \left(\frac{d}{dt} g(t)\right)\left(\frac{d^2}{dt^2} f(t)\right)}{\left(\left(\frac{d}{dt} f(t)\right)^2 + \left(\frac{d}{dt} g(t)\right)^2\right)^{3/2}} \right| \tag{3.7}$$

We see that the curvature for a function expressed parametrically can be significantly simplified, resolving the complex fraction by multiplying by $\frac{(f'(t))^2}{(f'(t))^2}$ and getting a nicer formula

$$K = \left| \frac{f'g'' - g'f''}{\left((f')^2 + (g')^2\right)^{\frac{3}{2}}} \right|$$

```
>
```

==

Here's a template Peggy developed for viewing curvature. Sorry for the rough seam between Susan and me, but Maple isn't feeling like home yet...

Objective: Seeing curvature in 2D

- User enters explicit 2D equation (twice, once as "f", once as "y"), the range for x's, and the point of interest.
- Template shows graph with point highlighted (sort of)
- then computes and reports curvature at the point, and
- reports the radius of the "osculating circle", or "circle of curvature", and sketches curve along with that circle.

Uglies

- Only works when center of circle is to the left of the graph and first der at point of interest being non-zero. (See newer code at bottom - I was afraid to start cut/pasting - had bad luck with that.)
- I didn't take time to find the plotting program that lets me set axes and contrain scales, so you might want to 1:1 the final graph.
- As the double-entry of the function shows clearly, I don't yet understand functions versus expressions, nor how to evaluate either one elegantly.

Enter explicit function for a curve (twice [sorry]), the x-range and the point of interest:

```
> restart; with(plots) :

  # User input here
   f := x→x^2 ; y := x^2 : X := -1;
  xmin := -4 : xmax := 4 :

  # processing...
 Y := f(X);
 plot( {[x, y, x = xmin..xmax], [X t, Y t, t = 1 ..1.2]}, thickness = 2);

  der1 := diff (y, x) : DER1 := eval(der1, x = X) :
  der2 := diff (y, x, x) : DER2 := eval(der2, x = X) :
```

$$k := \frac{\mathrm{abs}(\mathrm{DER2})}{\left(1 + \mathrm{DER1}^2\right)^{\frac{3}{2}}} : K := \mathrm{eval}(\ k, x = X) :$$

$$r := \frac{1}{(k)} : \quad R := \mathrm{eval}(\ r, x = X) :$$

$$m := -\frac{1}{\mathrm{DER1}} : \quad M := \mathrm{eval}(m, x = X) :$$

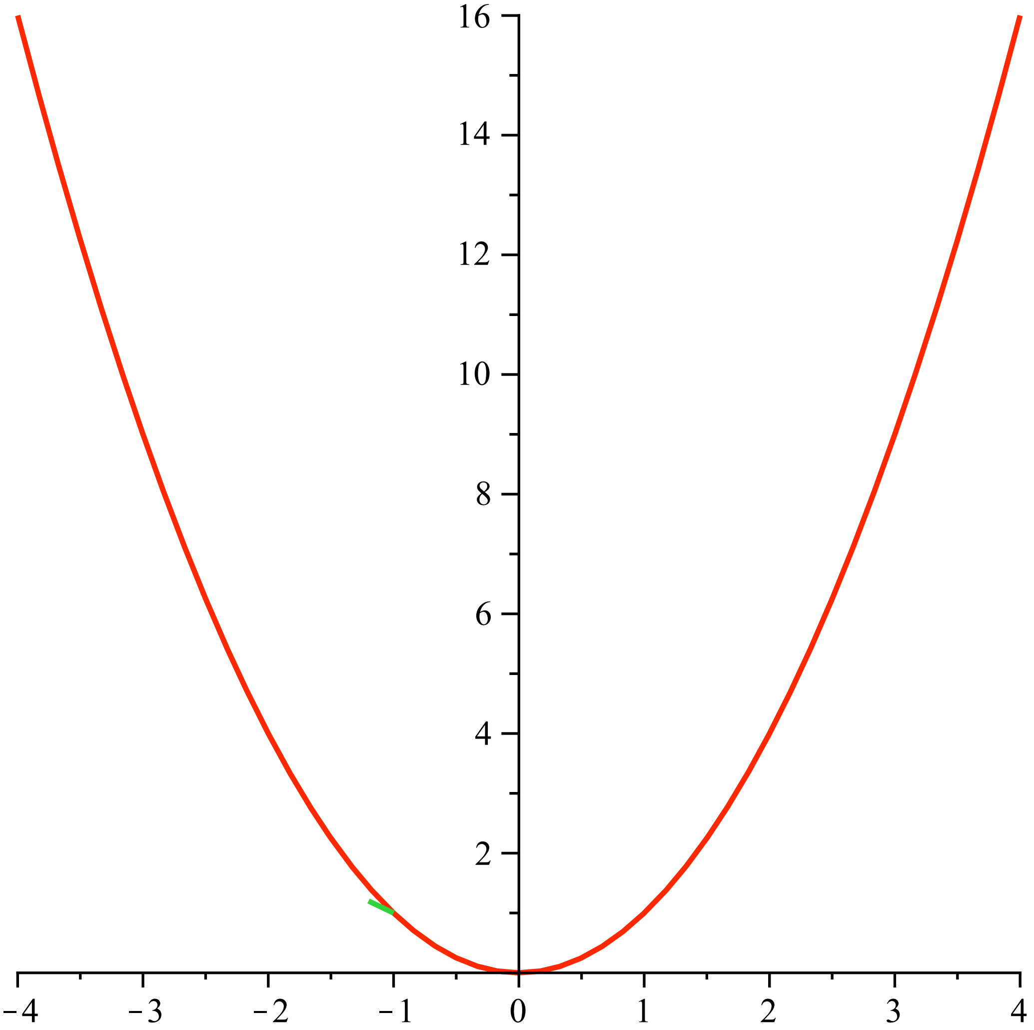
$$xc := X - \frac{r}{\mathrm{sqrt}\left(1 + m^2\right)} : XC := \mathrm{eval}(\ xc, x = X);$$

```
  YC := Y − M·(X−XC) :

 plot( {[x, y, x = xmin..xmax], [R·cos(t) + XC, R·sin(t) + YC, t = 0 ..2·Pi]}) ;
 print("curvature (k) = "); evalf (K, 3);
  print("Radius of oscullating circle = 1/k ...="); evalf (R, 3);

>
```

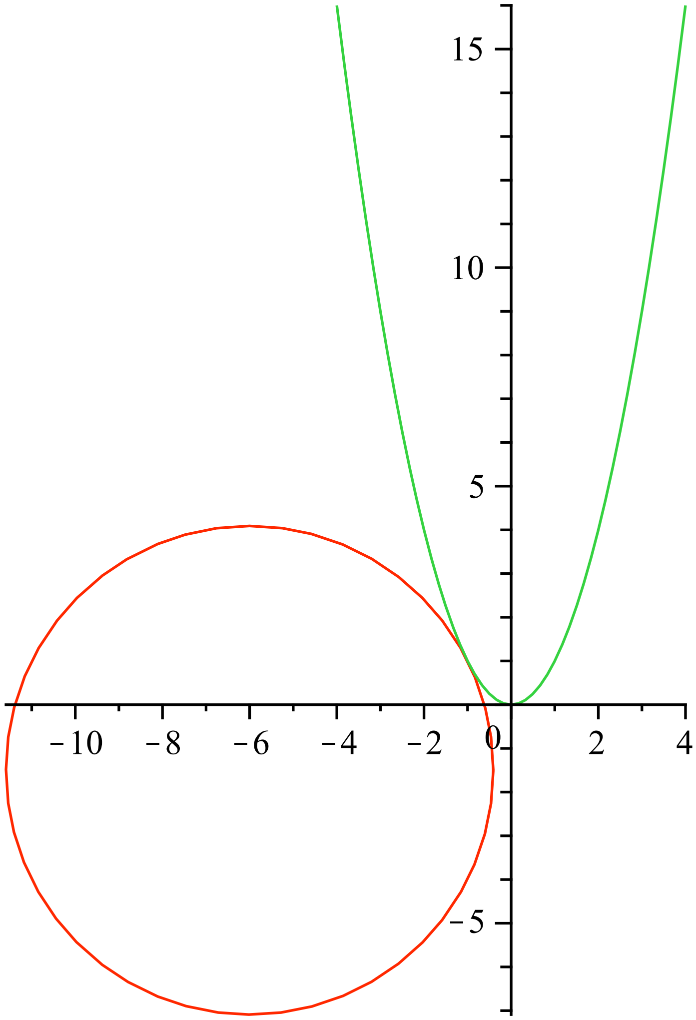
$$f := x \rightarrow x^2$$

$$X := -1$$

$$Y := 1$$



$XC := -6$

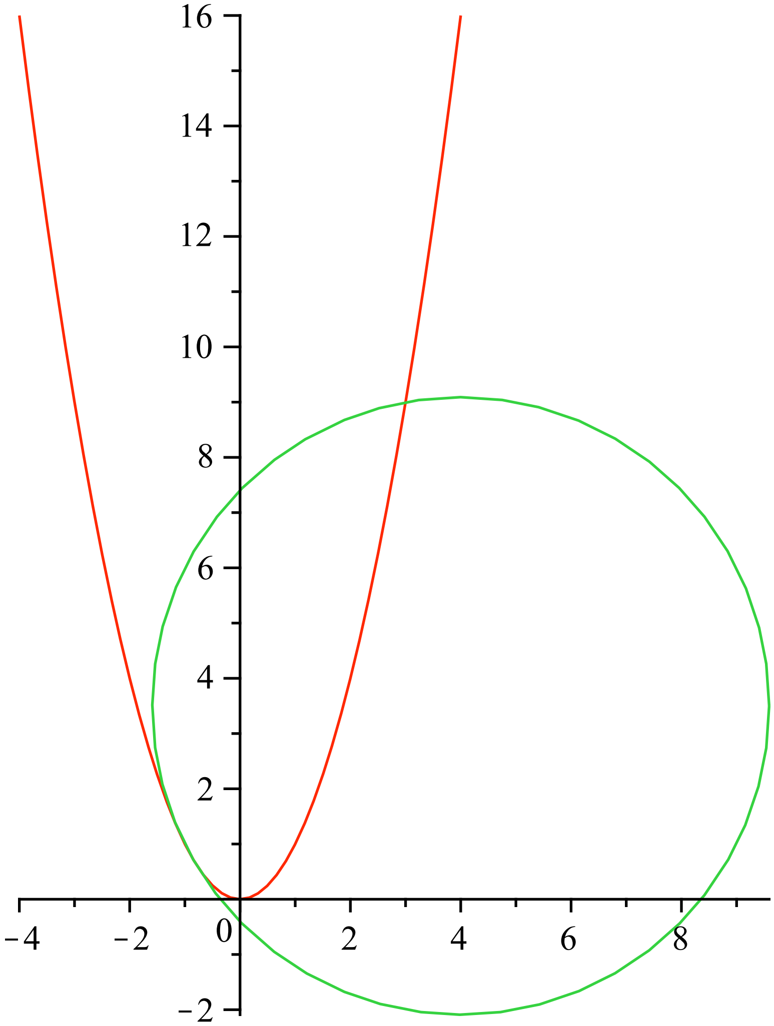


```
"curvature (k) = "
0.179
"Radius of oscullating circle = 1/k ...="
5.60                                                  (1)
```

Brand new code to deal with the center being other than "left and up".
Still need to deal with der at point of interest being zero...

```
> xcSign :=if( ( DER1 > 0 and DER2 < 0 ) or ( DER1 < 0 and DER2 > 0 ),-1, 1 ) :
  ycSign :=if( DER2 < 0,-1, 1 ) :
  xcc := X − r·xcSign / sqrt(1 + m^2) : XCC := eval( xcc, x = X) :
  YCC := Y − M·(X − XCC) :
```

```
> plot(  { [x, y, x = xmin..xmax], [R·cos(t) + XCC, R·sin(t) + YCC, t = 0 ..2·Pi ]})   ;
  print("curvature (k) = "); evalf(K, 3);
   print("Radius of osculating circle = 1/k ...="); evalf(R, 3);
```



"curvature (k) = "

0.179

"Radius of oscullating circle = 1/k ...="

5.60 **(2)**

>