In [1]:

```
# an ellipse given in parametric form
# that the ellipse is titled and shifted
# with centre at (x=c1=2, y=c2=3)
# and major radius r1=4
# and minor radius r2=1
# and titled by angle alpha=pi/3
c1=2
c2=3
r1=4
r2=1
alpha=pi/3
x(t)=r1*cos(alpha)*cos(t)-r2*sin(alpha)*sin(t)+c1
y(t)=r1*sin(alpha)*cos(t)+r2*cos(alpha)*sin(t)+c2
```

In [2]:

```
# define a fixed point P at (ptx=2, pty=6)
#
ptx=2
pty=6
```

In [3]:

```
# define the square distance DS(t) between the point
# and the ellipse with parameter t
#
DS(t)=(x(t)-ptx)^2+(y(t)-pty)^2
```

In [4]:

```
# to find the closest point on the ellipse to point P,
# we find t minimizing DS
# thus, we differentiate DS w.r.t. t,
# let it equals zero, and solve for t
#
show(solve(diff(DS(t),t)==0,t))
```

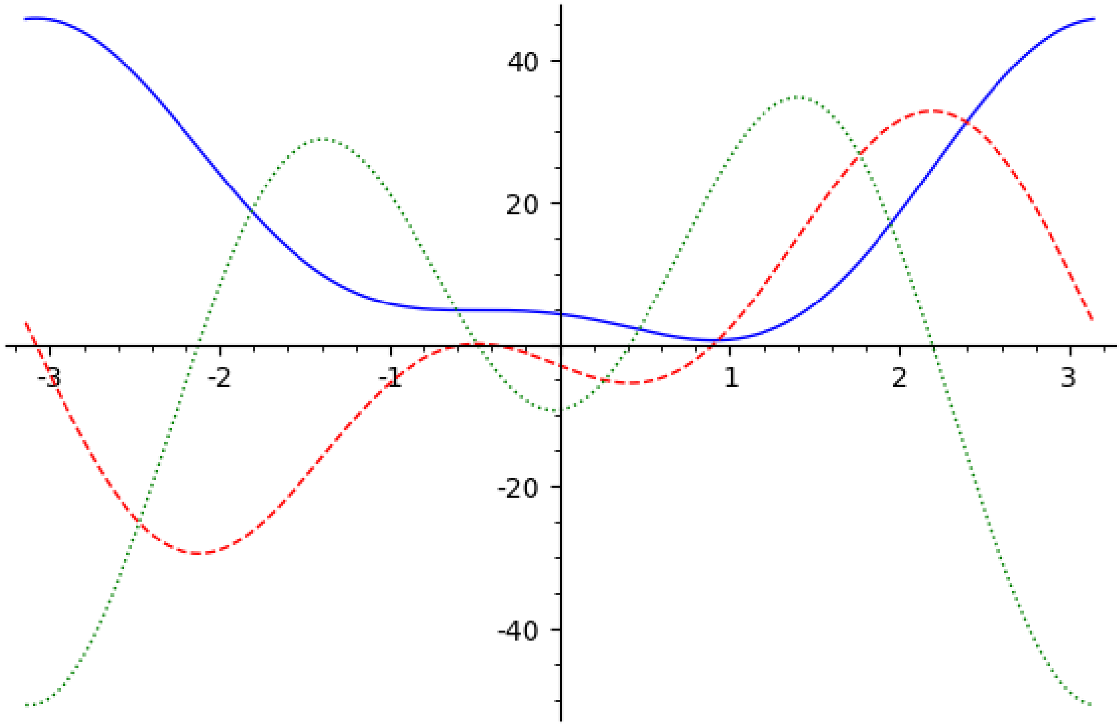
Out[4]:

$$\left[\sin(t) = \frac{\cos(t)}{2\left(2\sqrt{3} - 5\cos(t)\right)}\right]$$

In [5]:

```
p0=plot(DS(t),t,-pi,pi)
p1=plot(diff(DS(t),t),t,-pi,pi,rgbcolor='red',linestyle = "dashed")
p2=plot(diff(DS(t),t,2),t,-pi,pi, rgbcolor='green', linestyle = "dotted" )
p0+p1+p2
```

Out[5]:



In [6]:

```
# 3 real roots for zero derivative, with one multiplicity 2
# since CoCalc cannot show analytic exact answer for t
# we need to find t numerically
#
t0=(diff(DS(t),t)==0).find_root(0,2,t)
show(t0)
```

Out[6]:

$$0.9085927836295521$$

In [7]:

```
t1=(diff(DS(t),t)==0).find_root(-1,0,t)
```

In [8]:

```
t2=(diff(DS(t),t)==0).find_root(-4,-3,t)
```

In [9]:

```
show(RR(x(t0)))
```

Out[9]:

$$2.54673063263921$$

In [10]:

```
show(RR(y(t0)))
```

Out[10]:

$$5.52424380418267$$

In [11]:

```
show(RR(sqrt(DS(t0))))
```

Out[11]:

$$0.724747088662418$$

In [12]:

```
# find the slope of tangent line m
# m(t) = dy/dx = (dy/dt)/(dx/dt)
```

```
#
var('yy xx')
xdash(t)=diff(x(t),t)
ydash(t)=diff(y(t),t)
m(t)=ydash(t)/xdash(t)
```

In [13]:

```
# tangent line at t=t0
#
tangent(xx)=solve((yy-y(t0))/(xx-x(t0))==m(t0),yy)[0].rhs()
```

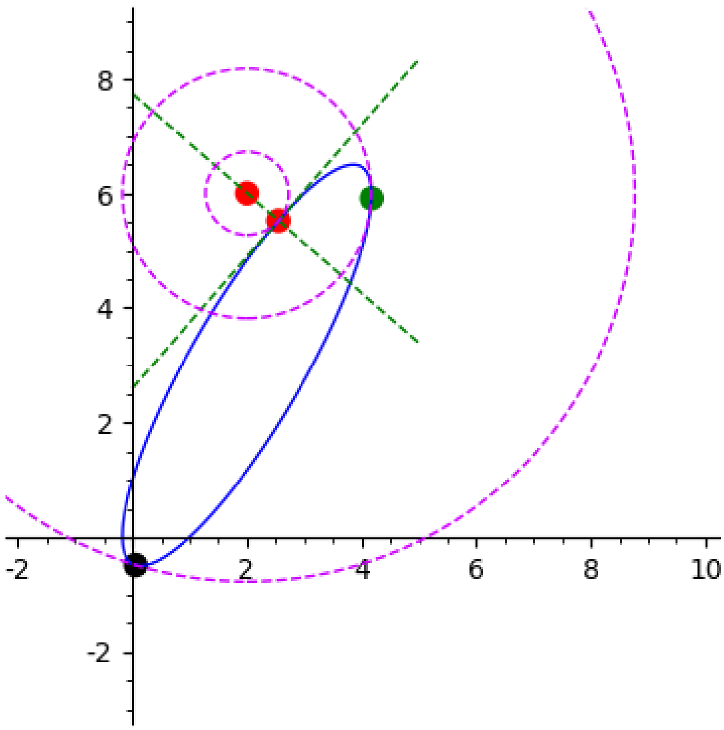
In [14]:

```
# normal line at t=t0
#
normal(xx)=solve((yy-y(t0))/(xx-x(t0))==(-1/m(t0)),yy)[0].rhs()
```

In [15]:

```
p1=parametric_plot( (x(t), y(t)), (t, -pi, pi) )
p2=plot(tangent(xx),xx,0,5, rgbcolor='green', linestyle = "dashed")
p3=plot(normal(xx),xx,0,5, rgbcolor='green', linestyle = "dashed")
cplt0=circle((ptx,pty),sqrt(DS(t0)),color=hue(0.8), linestyle = "dashed")
cplt1=circle((ptx,pty),sqrt(DS(t1)),color=hue(0.8), linestyle = "dashed")
cplt2=circle((ptx,pty),sqrt(DS(t2)),color=hue(0.8), linestyle = "dashed")
pt0 = point((x(t0),y(t0)), rgbcolor='red', pointsize=80)
pt00 = point((ptx,pty), rgbcolor='red', pointsize=80)
pt1 = point((x(t1),y(t1)), rgbcolor='green', pointsize=80)
pt2 = point((x(t2),y(t2)), rgbcolor='black', pointsize=80)
plotall=p1+p2+p3+pt00+pt0+pt1+pt2+cplt0+cplt1+cplt2
(plotall).show(xmin=-2, xmax=10, ymin=-3, ymax=9,aspect_ratio=1)
#
# Note that the green point is neither a maximum nor a minimum
# thus, still 2 extrema in total
#
```

Out[15]:



In [0]: