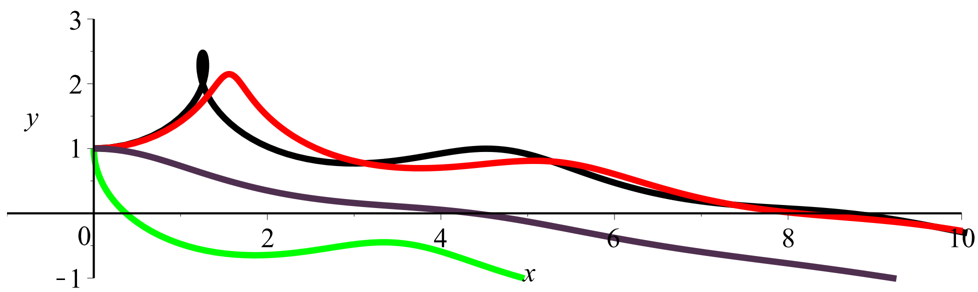
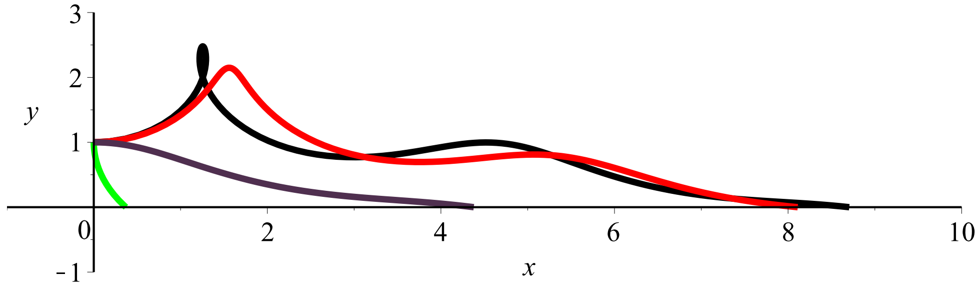
```
>
> with(DEtools) :
> xphug:=R-> [diff(theta(t),t) = (v(t)^2 - cos(theta(t)))/(v(t)),
             diff(v(t),t)      = -sin(theta(t))-R*v(t)^2,
             diff(x(t),t)      = v(t)*cos(theta(t)),
             diff(y(t),t)      = v(t)*sin(theta(t))]:
> DEplot(xphug(0.2), [theta,v,x,y], t=0..25,
      [[v(0)=2.3,theta(0)=0, x(0)=0, y(0)=1],
       [v(0)=2,theta(0)=0, x(0)=0, y(0)=1],
       [v(0)=0.1,theta(0)=-Pi/2, x(0)=0, y(0)=1],
       [v(0)=0.8,theta(0)=0, x(0)=0, y(0)=1]],
      theta=-Pi..3*Pi, v=0..2.2,x=-1..10, y=-1..3,
      linecolor=[black,red, green,violet],
     numpoints=300, obsrange=false,
        scene=[x,y]);
```




```
> FIX := piecewise( y(t)>0, 1, 0);
```

$$FIX := \begin{cases} 1 & 0 < y(t) \\ 0 & otherwise \end{cases} \quad (1)$$

```
> yphug:=R->[diff(theta(t),t)= FIX*(v(t)^2-cos(theta(t)))/(v(t)),
             diff(v(t),t)      = FIX*(-sin(theta(t))-R*v(t)^2),
             diff(x(t),t)      = FIX*(v(t)*cos(theta(t))),
             diff(y(t),t)      = FIX*(v(t)*sin(theta(t)))]:
> DEplot(yphug(0.2), [theta,v,x,y], t=0..25,
      [[v(0)=2.3,theta(0)=0, x(0)=0, y(0)=1],
       [v(0)=2,theta(0)=0, x(0)=0, y(0)=1],
       [v(0)=0.1,theta(0)=-Pi/2, x(0)=0, y(0)=1],
       [v(0)=0.8,theta(0)=0, x(0)=0, y(0)=1]],
      theta=-Pi..3*Pi, v=0..2.2,x=-1..10, y=-1..3,
      linecolor=[black,red, green,violet],
     numpoints=300, obsrange=false,
        scene=[x,y]);
```



Could deal with the "v small causes trouble" issue by using
FIX := piecewise(y(t)>0, arctan(v(t), 0);
which multiplies vectors by "v" for v small, Pi/2 for v large.

New goal:

Instead of looking at how glider flies, just care about how far it goes.

```
> dsolve({D(y)(t)=y(t)^3, y(0)=2});
```

$$y(t) = \frac{2}{\sqrt{-8\,t+1}} \quad \mathbf{(2)}$$

```
> sol:=dsolve({D(y)(t)=y(t)^3, y(0)=2}, numeric);
```

$$sol := \mathbf{proc}(x\_rkf45) \ \ldots \ \mathbf{end\ proc} \quad \mathbf{(3)}$$

```
> sol(0.1);
```

$$[t=0.1, y(t) = 4.47213728529402] \quad \mathbf{(4)}$$

```
> op(yphug(0.2));  # this is my DE without brackets
```

$$\frac{d}{dt}\,\theta(t) = \frac{\left(\begin{cases} 1 & 0 < y(t) \\ 0 & otherwise \end{cases}\right)\left(v(t)^2 - \cos(\theta(t))\right)}{v(t)}, \frac{d}{dt}\,v(t) = \left(\begin{cases} 1 & 0 < y(t) \\ 0 & otherwise \end{cases}\right)\left(-\sin(\theta(t)) - 0.2\,v(t)^2\right), \frac{d}{dt}\,x(t) = \left(\begin{cases} 1 & 0 < y(t) \\ 0 & otherwise \end{cases}\right)v(t)\cos(\theta(t)), \frac{d}{dt}\,y(t) = \left(\begin{cases} 1 & 0 < y(t) \\ 0 & otherwise \end{cases}\right)v(t)\sin(\theta(t)) \quad \mathbf{(5)}$$

```
> sol:=dsolve({op(yphug(0.2)), v(0)=2,theta(0)=0, x(0)=0, y(0)=1},
         numeric);
```

$$sol := \mathbf{proc}(x\_rkf45) \ \ldots \ \mathbf{end\ proc} \quad \mathbf{(6)}$$

```
> sol(1);
```

$$[t=1., \theta(t) = 0.985122195444121, v(t) = 0.996549463923899, x(t) = 1.25746895798693, y(t) = 1.72853892156729] \quad \mathbf{(7)}$$

```
> sol(10);
```

$$[t=10., \theta(t) = -0.129827936655498, v(t) = 1.04316767149762, x(t) = 8.11111836801073, y(t) = -2.78796654869703\ 10^{-7}] \quad \mathbf{(8)}$$

```
> ?dsolve[numeric]
> solp:=dsolve({op(yphug(0.2)), v(0)=2,theta(0)=0, x(0)=0, y(0)=1},

         numeric,output=listprocedure);
```

$$solp := [t = \mathbf{proc}(t) \ \ldots \ \mathbf{end\ proc}, \theta(t) = \mathbf{proc}(t) \ \ldots \ \mathbf{end\ proc}, v(t) = \mathbf{proc}(t) \ \ldots \ \mathbf{end\ proc}, x(t) = \mathbf{proc}(t) \ \ldots \ \mathbf{end\ proc}, y(t) = \mathbf{proc}(t) \ \ldots \ \mathbf{end\ proc}] \quad (9)$$

```
> solp[3](10); solp[4](10);
```

$$v(t)(10) = 1.04316767149762$$
$$x(t)(10) = 8.11111836801073 \quad (10)$$

```
> rhs(solp[4](10));
```

8.11111836801073 **(11)**

Write a maple procedure which takes v_0 as input, and gives x(t) for initial conds
R=0.2, theta(0)=0, x(0)=0, y(0)=1, v(0)=v_0

```
> crashdist:= proc(v0::numeric, {R:=0.2, maxt:=10})
    local solp;
    global yphug;
    solp:=dsolve({op(yphug(R)), v(0)=v0,
                 theta(0)=0, x(0)=0, y(0)=1},
              numeric,output=listprocedure);
    return( rhs(solp[4](maxt)) );
  end:
> crashdist(2.0);
```

8.11111836801073 **(12)**

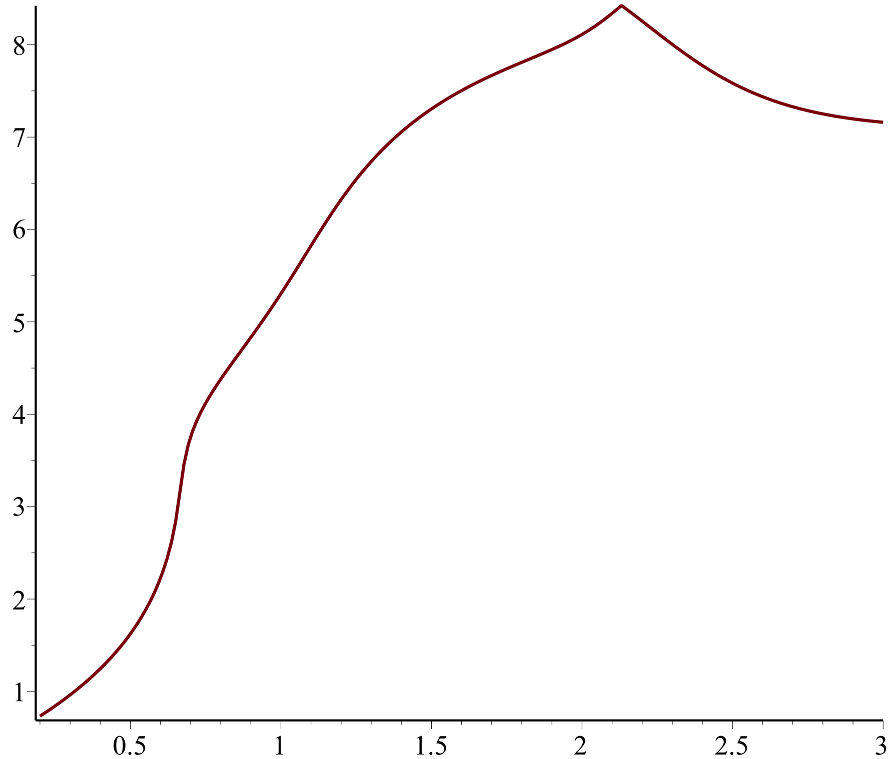
```
> crashdist(2.1);
```

8.33963448195194 **(13)**

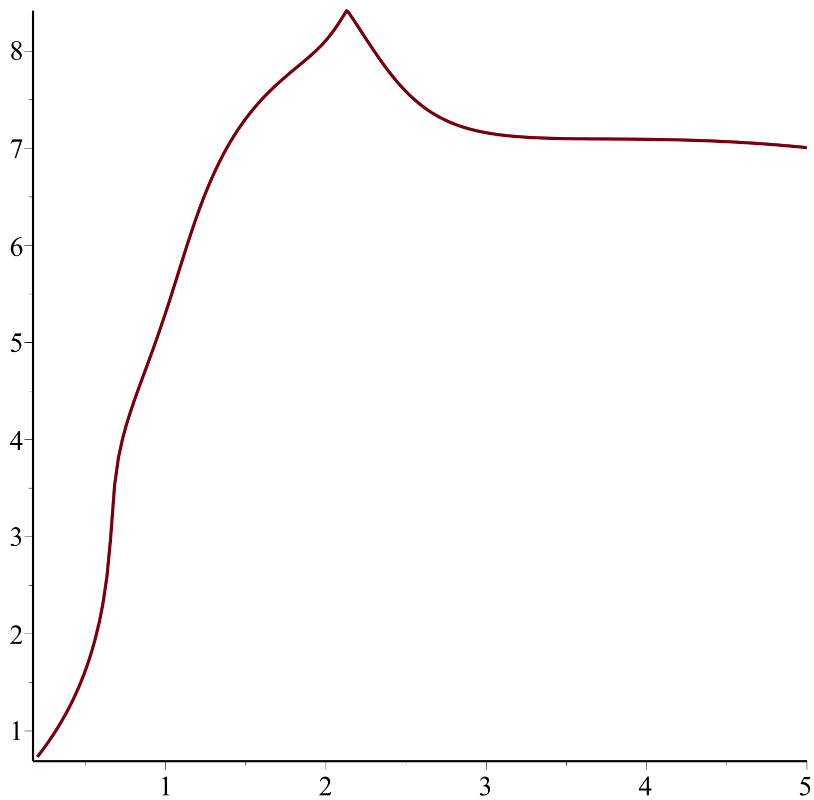
```
> crashdist(2.5);
```

7.58400099982691 **(14)**

```
> plot(crashdist(v), v=0.2..3);
Error, invalid input: crashdist expects its 1st argument, v0, to
be of type numeric, but received v
> plot(crashdist, 0.2..3);
```



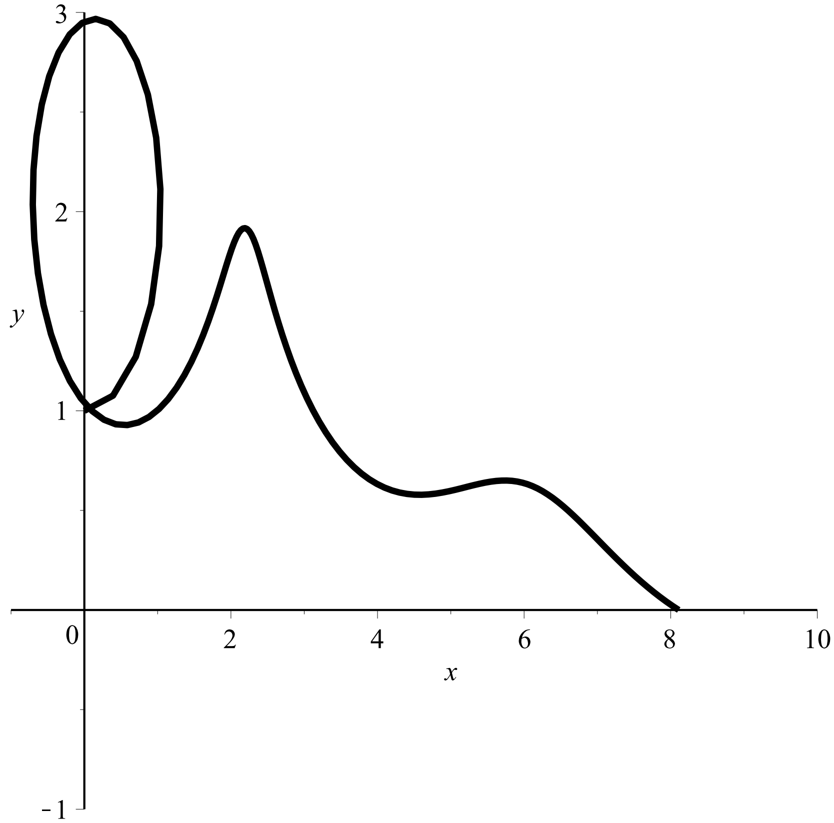

```
> plot(crashdist, 0.2..5);
```



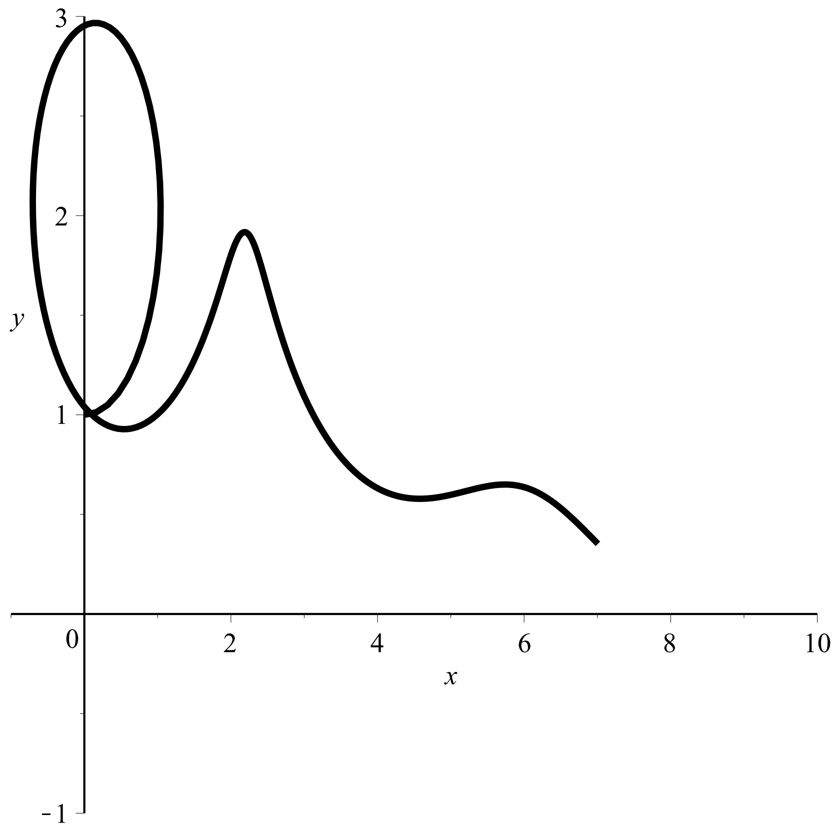
I am suspicious of big values of v0. What is wrong?

$$7.00680266445564 \quad (15)$$

```
> DEplot(yphug(0.2), [theta,v,x,y], t=0..25,
       [[v(0)=5,theta(0)=0, x(0)=0, y(0)=1]
         ],
       theta=-Pi..3*Pi, v=0..2.2,x=-1..10, y=-1..3,
       linecolor=black,
      numpoints=300, obsrange=false,
        scene=[x,y]);
```




```
> DEplot(yphug(0.2), [theta,v,x,y], t=0..10,
      [[v(0)=5,theta(0)=0, x(0)=0, y(0)=1]
        ],
      theta=-Pi..3*Pi, v=0..2.2,x=-1..10, y=-1..3,
      linecolor=black,
     numpoints=300, obsrange=false,
        scene=[x,y]);
```



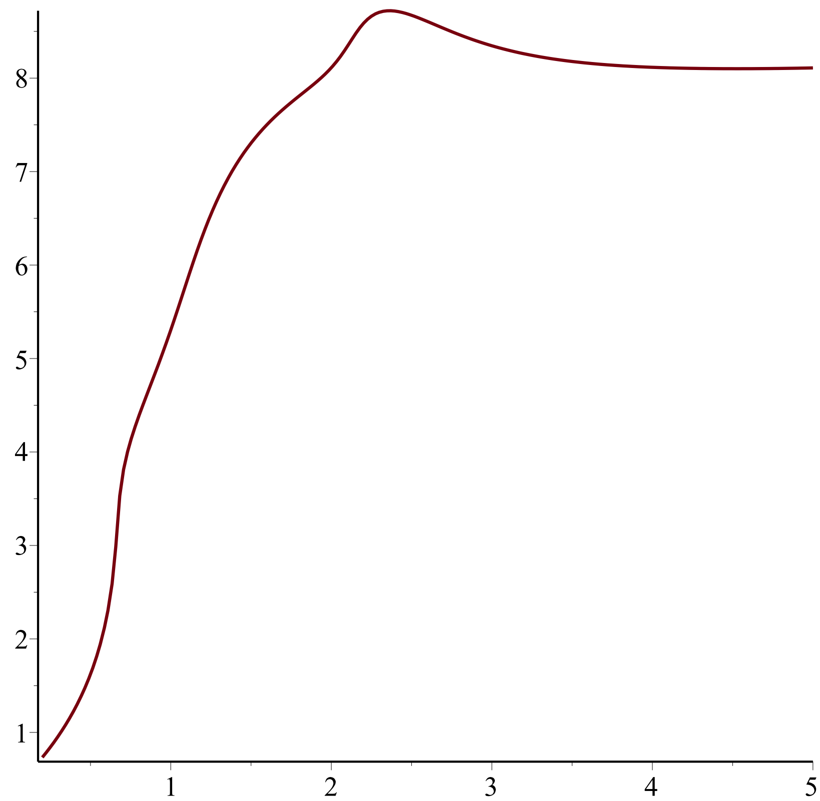
Let's change it so that it keeps flying until it crashes. It runs the DE until t=maxt, checks if y(t)>0, and if so, restarts the DE where it left off and runs it for another maxt time.

```
> crashdist:= proc(v0::numeric, {R:=0.2, maxt:=3})
    local solp, ti, xi, yi, thetai, vi;
   global yphug;
   ti:=0; xi:=0; yi:=1; thetai:=0; vi:=v0;
   while( yi >0) do
     solp:=dsolve({op(yphug(R)), v(0)=vi,
                theta(0)=thetai, x(0)=xi, y(0)=yi},
               numeric,output=listprocedure);
     ti:=ti+maxt;  # ignore solp[1]
     thetai:=rhs(solp[2](maxt));
     vi:=rhs(solp[3](maxt));
     xi:=rhs(solp[4](maxt));
     yi:=rhs(solp[5](maxt));
 #      print("flew for ",ti);
    od;
    return(xi );
 end:
```

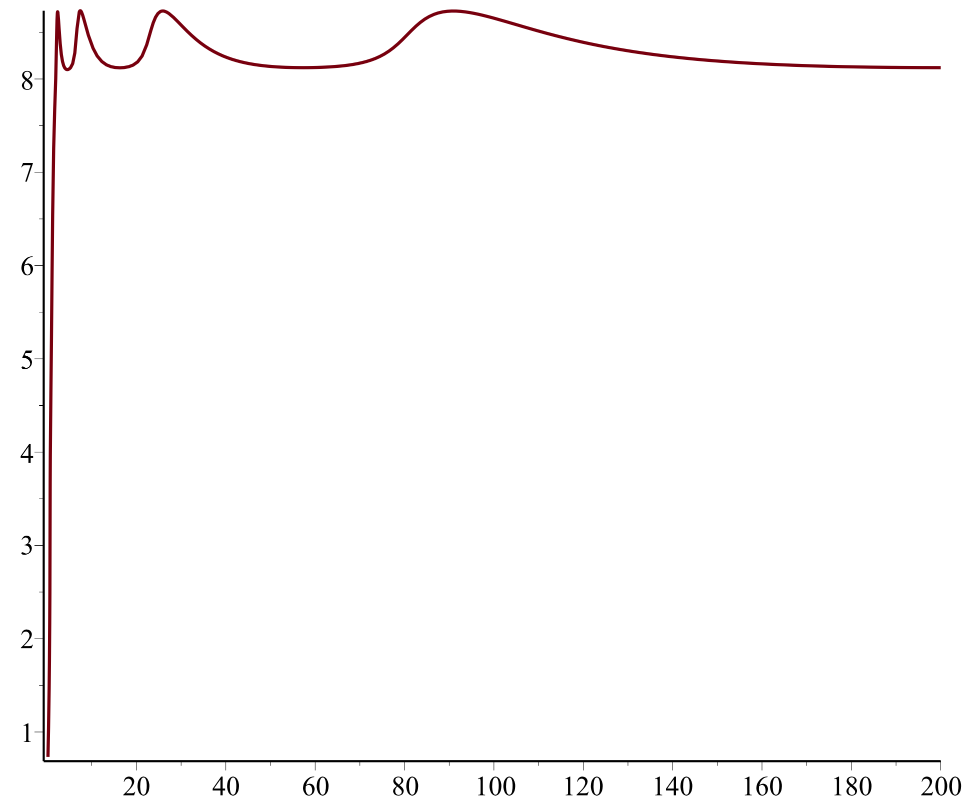
```
> crashdist(5);
```

8.10913672088352 **(16)**

```
> plot(crashdist, 0.2..5);
```




```
> plot(crashdist, 0.2..200);
```


8
7
6
5
4
3
2
1
20
40
60
80
100
120
140
160
180
200