March 7, 2024

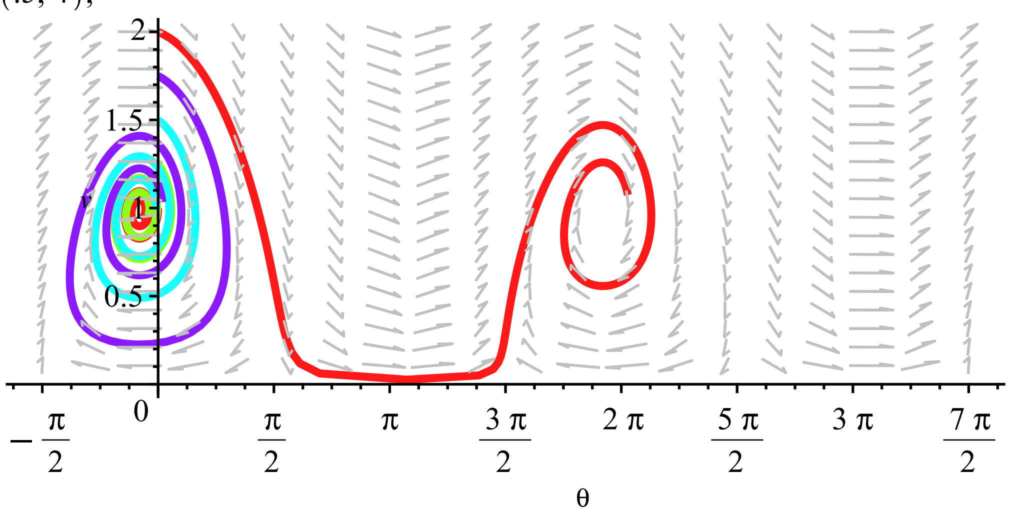
More phugoid

```
> phug := [D(theta)(t) = v(t) - cos(theta(t))/v(t), D(v)(t) = -sin(theta(t)) - R·(v(t))^2]:
> with(DEtools) :
> gliderpic :=proc(R, nsols := 3, step := 0.05)
   local v0, t, h :
   local theta, v, pic;
   local phug;
   phug := [D(theta)(t) = v(t) - cos(θ(t))/v(t), D(v)(t) = -sin(θ(t)) - R(v(t))^2]:
  pic := DEplot(phug, [theta(t), v(t)], t = 0 ..10,
  [seq([theta(0) = 0, v(0) = v0], v0 = 1 ..2, 1.0/nsols)],
  theta = -Pi/2 .. 7·Pi/2, v = 0 ..2, stepsize = step,

  # stepsize controls roughness of solution, smaller is more computation but smoother picture.

  tickmarks = [piticks, default], size = [.75, .5],

  linecolor = [seq(COLOR(HUE, h), h = 0 ..1, 1.0/nsols)], color = gray);
  return(pic);
  end:
> gliderpic(.5, 4);
```



Some more stuff about procedure arguments

```
> Joe :=proc(x, y)
    print(x, y);
  end:
> Joe(2)
```

Error, invalid input: Joe uses a 2nd argument, y, which is missing

```
> Joe(2, rabbit);
```

$$2, rabbit \quad (1)$$

```
> Joe := proc(x, y := "cat")
    print(x, y);
  end:
> Joe(2)
```

$$2, \text{"cat"} \quad (2)$$

```
> Joe(2, 3)
```

$$2, 3 \quad (3)$$

```
> Joe := proc(x, y := "cat", {third := 0})
    print(x, y, third);
  end:
> Joe(1, 2, 3)
```

$$1, 2, 0 \quad (4)$$

```
> Joe(1, 2, third = 7)
```

$$1, 2, 7 \quad (5)$$

```
> Joe(1, third = 5)
```

$$1, \text{"cat"}, 5 \quad (6)$$

parameters /arguments to procs can be positional, with or without default values, or named.
In this example, y is 2nd argument with value "cat" if omitted, and third can come in any order, but we have to give it a value usings its name (eg, third=7).

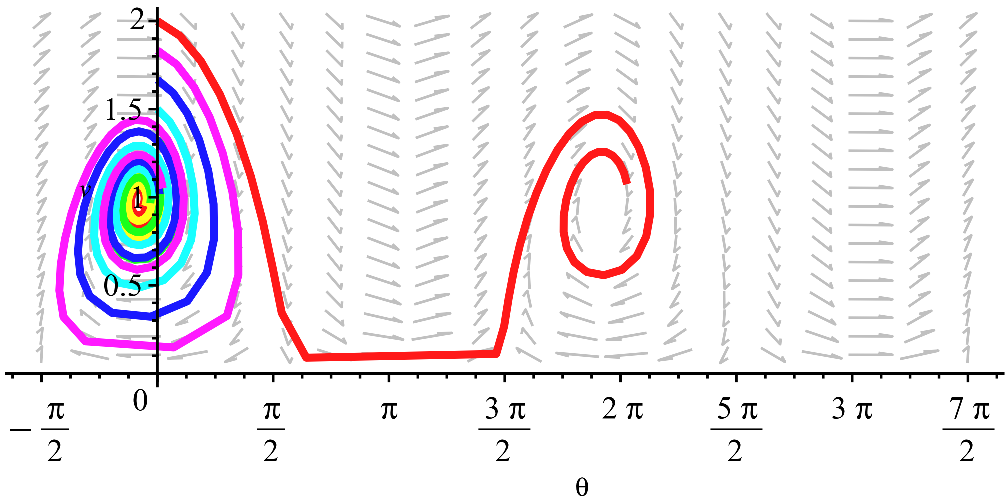
```
> gliderpic := proc(R := 0, {nsols := 3}, {step := 0.05})
   local v0, t, h :
   local theta, v, pic;
   local phug;
   phug := [D(theta)(t) = v(t) - cos(θ(t))/v(t), D(v)(t) = -sin(θ(t)) - R(v(t))^2]:
  pic := DEplot(phug, [theta(t), v(t)], t = 0..10,
  [seq([theta(0) = 0, v(0) = v0], v0 = 1..2, 1.0/nsols)],
  theta = -Pi/2 .. 7·Pi/2, v = 0..2, stepsize = step,

  # stepsize controls roughness of solution, smaller is more computation but smoother picture.

  tickmarks = [piticks, default], size = [.75, .5],

  linecolor = [seq(COLOR(HUE, h), h = 0..1, 1.0/nsols)], color = gray);
  return(pic);
  end:

> gliderpic(.5, step = 1, nsols = 6)
```



A diversion about stepsize....

> $f := x \rightarrow \sin(x^2);$

$$f := x \mapsto \sin(x^2) \quad (7)$$

> $int(f(x), x = 0 ..Pi);$

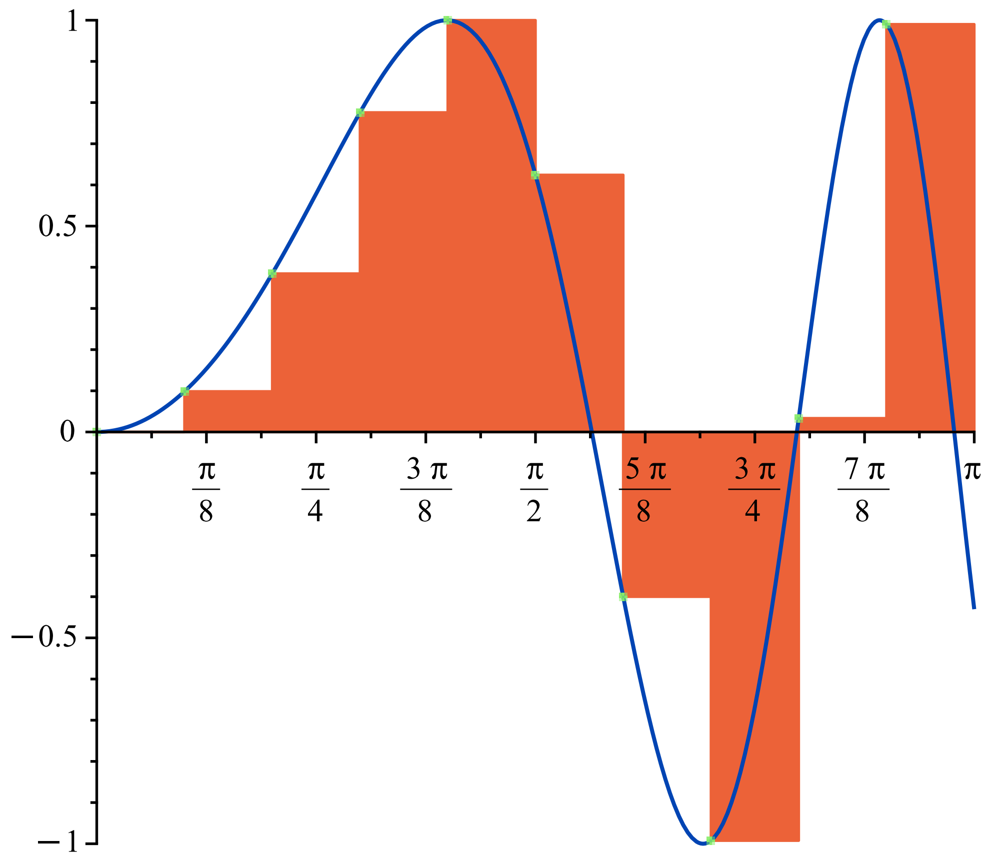
$$\frac{\text{FresnelS}\left(\sqrt{2}\,\sqrt{\pi}\right)\sqrt{2}\,\sqrt{\pi}}{2} \quad (8)$$

> $evalf(\%);$

$$0.7726517130 \quad (9)$$

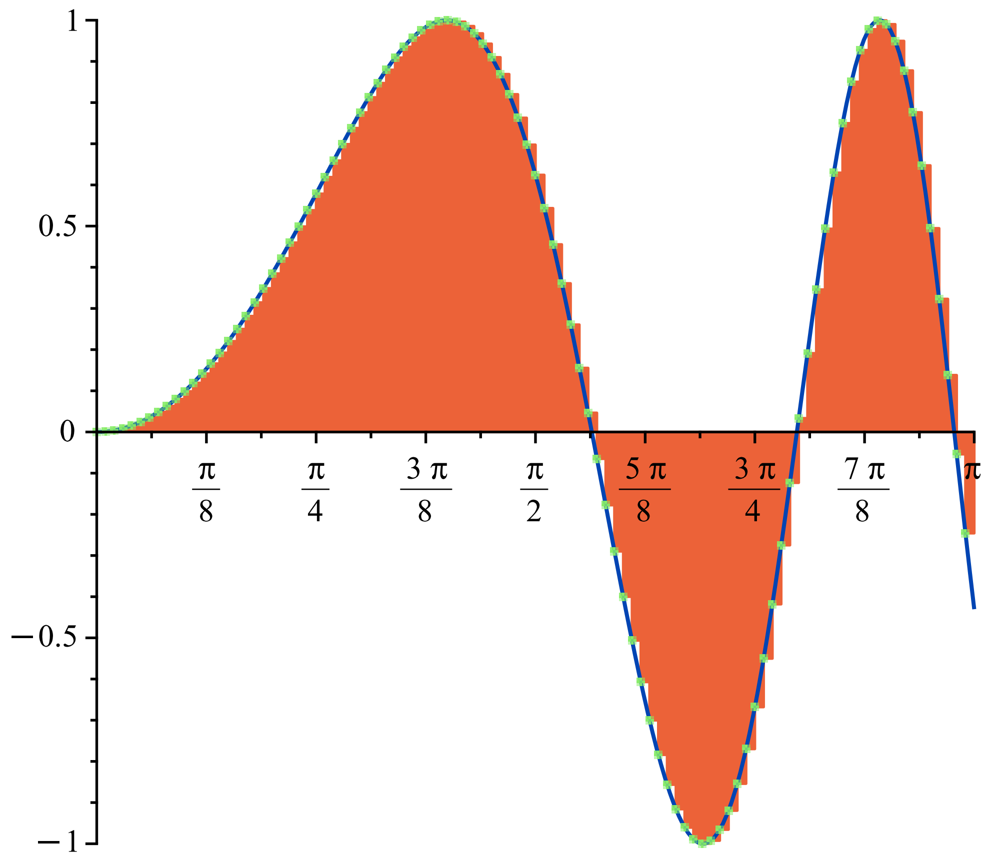
> *with*(*Student*[*Calculus1*]) :

> *RiemannSum*(*f*(*x*), *x* = 0 ..Pi, *method* = *left*, *partition* = 10, *output* = *plot*);
 # this is like Euler's method



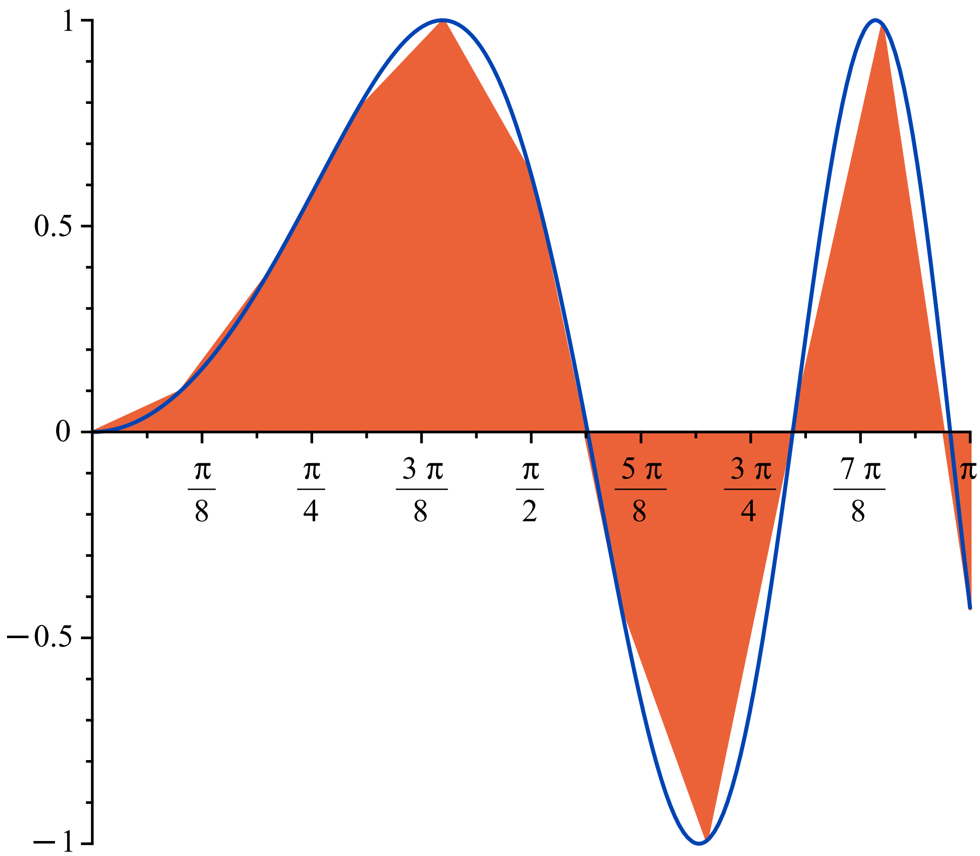
A left Riemann sum approximation of $\int_0^{\pi} f(x)\, \mathrm{d}x$, where $f(x) = \sin(x^2)$ and tl

> $RiemannSum(f(x), x = 0..\mathrm{Pi}, method = left, partition = 100, output = plot);$



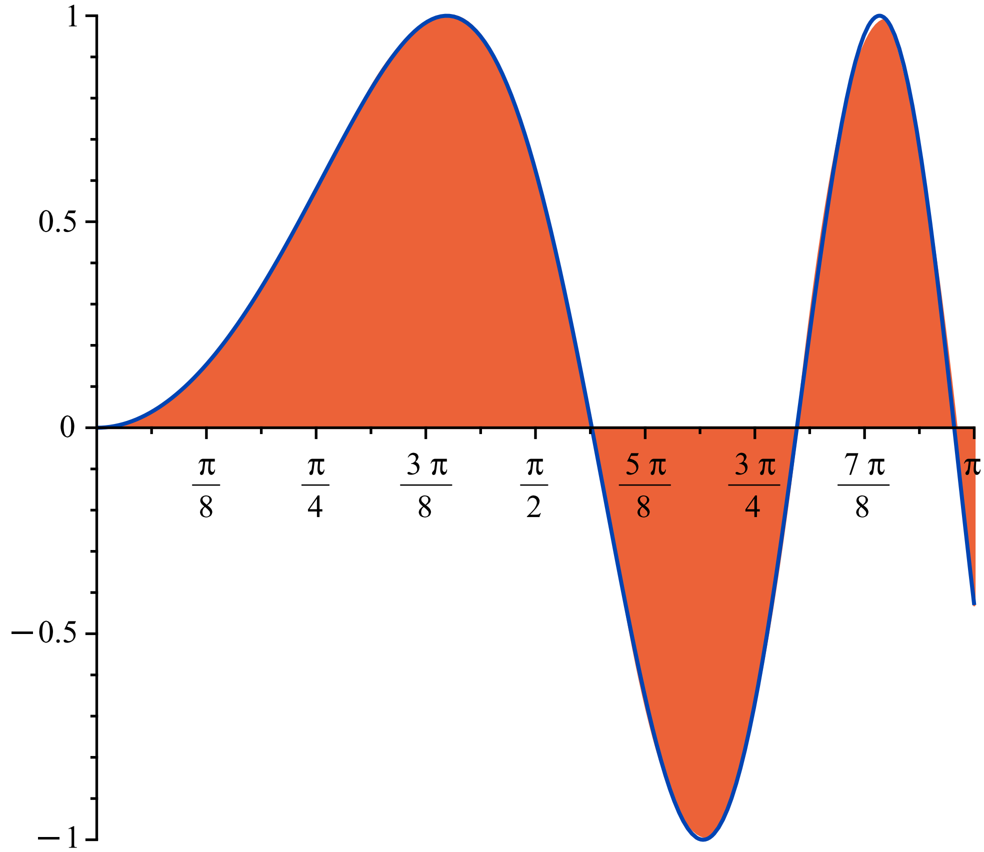
A left Riemann sum approximation of $\int_0^{\pi} f(x)\,dx$, where $f(x) = \sin(x^2)$ and tl

> *ApproximateInt*$(f(x), x = 0..\text{Pi}$, *method* = *trapezoid*, *partition* = 10, *output* = *plot*);



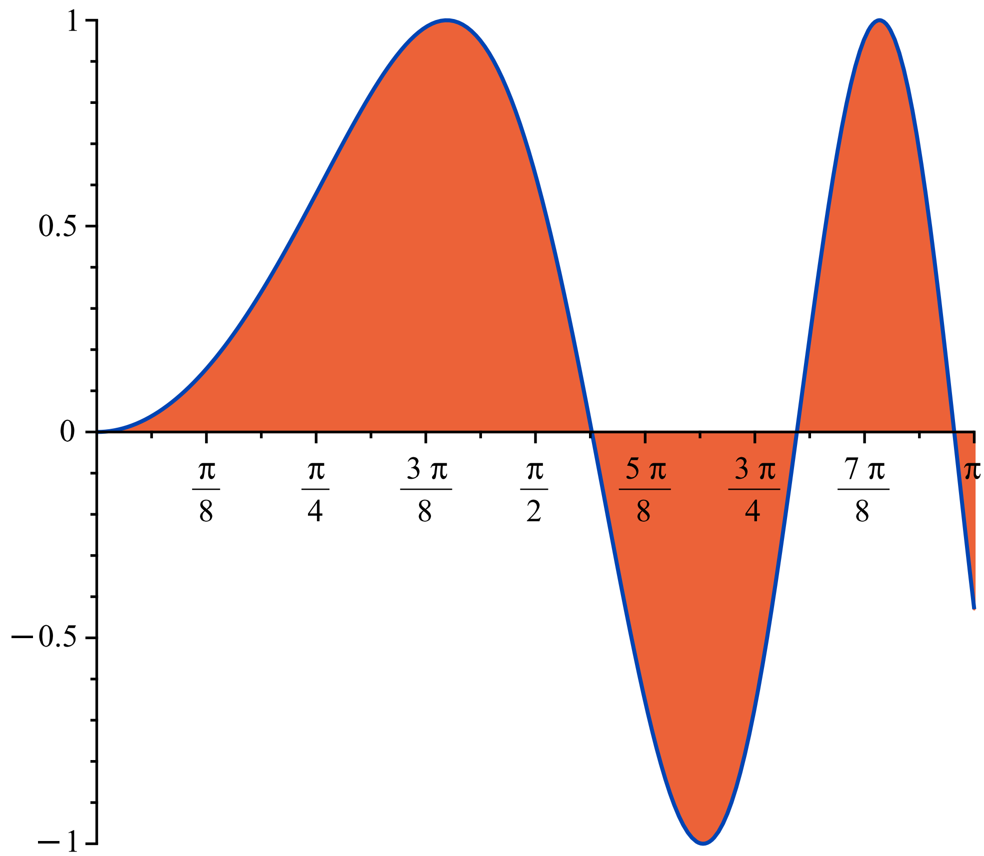
An approximation of $\int_0^{\pi} f(x)\,dx$ using trapezoid rule, where $f(x) = \sin(x^2)$ ar

> *ApproximateInt*(*f*(*x*), *x* = 0 ..Pi, *method* = *simpson*, *partition* = 10, *output* = *plot*);



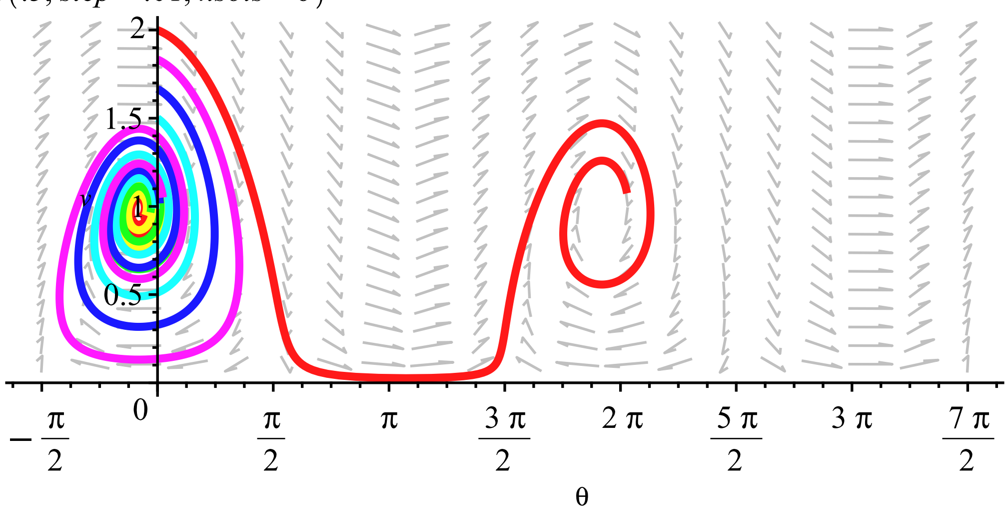
An approximation of $\int_0^{\pi} f(x)\,dx$ using Simpson's rule, where $f(x) = \sin(x^2)$ a

> *ApproximateInt*($f(x)$, $x = 0$ ..Pi, *method* = *simpson*, *partition* = 100, *output* = *plot*);



An approximation of $\int_0^{\pi} f(x)\, dx$ using Simpson's rule, where $f(x) = \sin(x^2)$ a

> $gliderpic(.5, step = .01, nsols = 6)$



How to see glider path?

> $xphug := \left[\mathrm{D}(\mathrm{theta})(t) = v(t) - \frac{\cos(\mathrm{theta}(t))}{v(t)}, \mathrm{D}(v)(t) = -\sin(\mathrm{theta}(t)) - R \cdot (v(t))^2, \right.$

$D(x)(t) = v \cdot \cos(\text{theta}(t)), D(y)(t) = v \cdot \sin(\text{theta}(t)) \Big];$

$$xphug := \left[D(\theta)(t) = v(t) - \frac{\cos(\theta(t))}{v(t)}, D(v)(t) = -\sin(\theta(t)) - R\,v(t)^2, D(x)(t) = v\cos(\theta(t)), D(y)(t) = v\sin(\theta(t)) \right] \quad \textbf{(10)}$$

\> $R := .5;$

$DEplot\Big(xphug, [\text{theta}(t), v(t), x(t), y(t)], t = 0..10,$

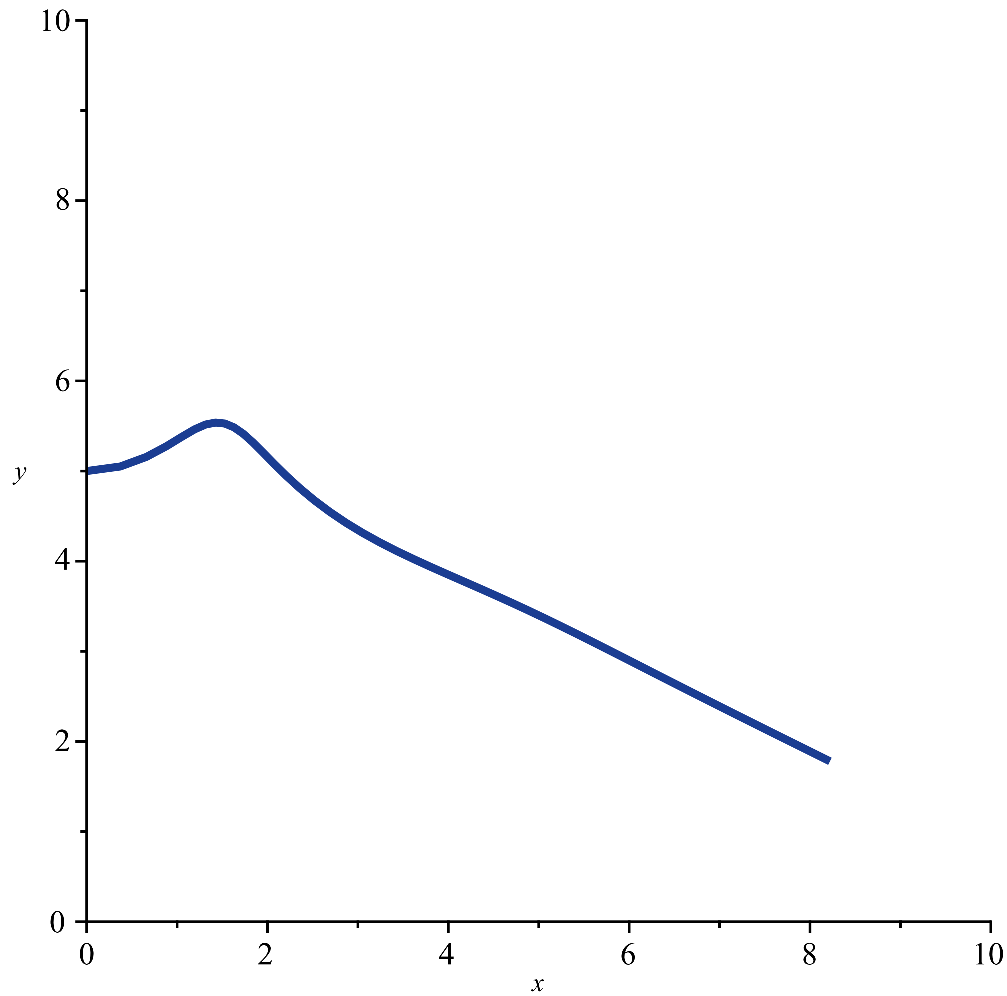
$[[v(0) = 2, \text{theta}(0) = 0, x(0) = 0, y(0) = 5]],$

$\text{theta} = -\text{Pi}..\frac{5\,\text{Pi}}{2}, v = 0..3, x = 0..10, y = 0..10,$

$scene = [x, y]\Big);$

$$R := 0.5$$

Warning, v is present as both a dependent variable and a name. Inconsistent specification of the dependent variable is deprecated, and it is assumed that the name is being used in place of the dependent variable.



> $DEplot\left(xphug, [\theta(t), v(t), x(t), y(t)], t=0..10,\right.$

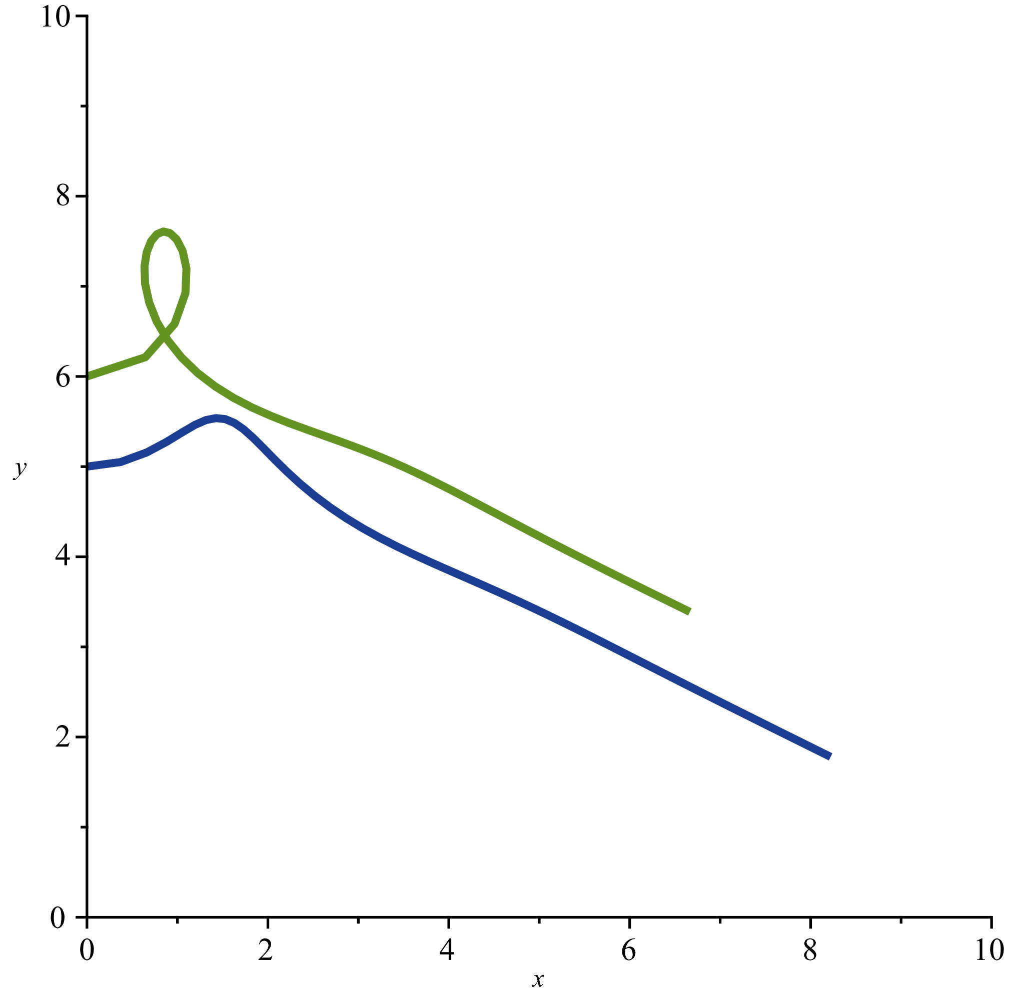
$[[v(0)=2, \theta(0)=0, x(0)=0, y(0)=5],$

$[v(0)=4, \theta(0)=0, x(0)=0, y(0)=6]],$

$\theta=-\pi..\frac{5\pi}{2}, v=0..3, x=0..10, y=0..10,$

$\left.scene=[x, y]\right)$

Warning, v is present as both a dependent variable and a name. Inconsistent specification of the dependent variable is deprecated, and it is assumed that the name is being used in place of the dependent variable.



>