Untitled11

June 14, 2025

1 VERIFICATION DU CALCUL SUR LES EQUATIONS DIFFERENTIELLES

[98]:

```
a = var('a')
b = var('b')
c = var('c')
t = var('t')
u = var('u')
delta = var('delta')
alpha = var('alpha')
beta = var('beta')
y_0 = var('y_0')
v_0 = var('v_0')
```

[99]:

```
delta = b**2 - 4*a*c
alpha = (-b - sqrt(delta))/(2*a)
beta = (-b + sqrt(delta))/(2*a)
```

[75]:

```
#from sage.symbolic.function import BuiltinFunction
#class F(BuiltinFunction):
#    def __init__(self):
#        BuiltinFunction.__init__(self, 'sin', nargs=0)
#    def _eval_(self, *args):
#        pass
```

[72]:

```
#f = function("sin", nargs=1)
```

[76]:

```
#f = F()
#f(t)
```

[76]:

```
sin(t)
```

[101]:

```
F(a,b,c,y_0,v_0) = (exp(alpha*t)*integral(-sin(u)*exp(-alpha*u),u,0,t) +␣
 ↪exp(beta*t)*integral(sin(u)*exp(-beta*u),u,0,t) + beta*y_0*exp(alpha*t) -␣
 ↪alpha*y_0*exp(beta*t) - v_0*exp(alpha*t) + v_0*exp(beta*t))/(beta-alpha)
```

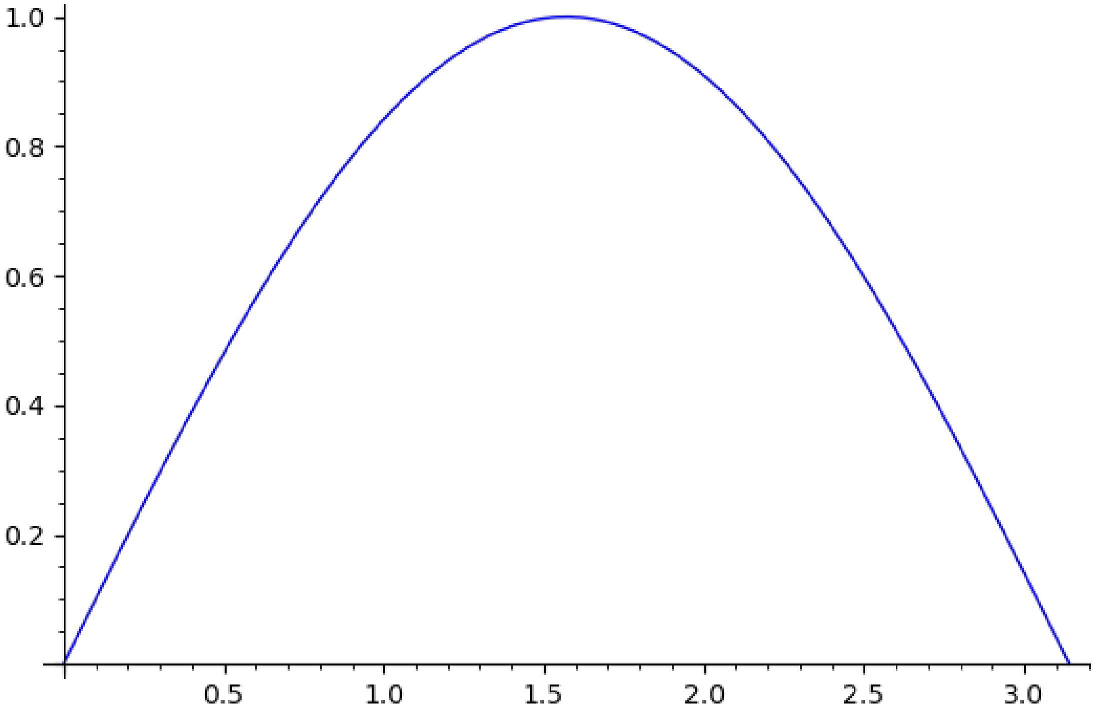
[110]:

```
P(t) = F(1,0,-1,1,1)
```

[111]:
```
S(t,a,b,c) = a*(P.derivative(t)).derivative(t) + b*P.derivative(t) + c*P
```

[115]:
```
T(t) = S(t,1,0,-1)
plot(T,t,0,pi)
```

[115]:



[116]:
```
T(pi/2)
```

[116]:
```
1
```

[117]:
```
T
```

[117]:
```
t |--> sin(t)
```

[ ]:
```

```