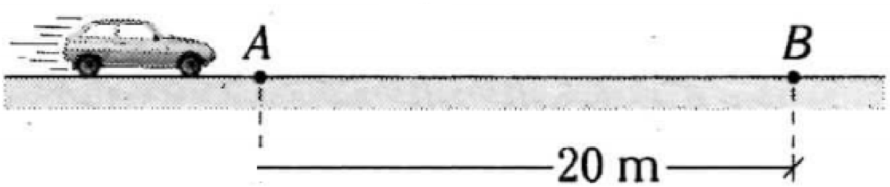
```
from sympy import *
from polyclass import *
from libaldo_math import *
from libaldo_show import *
from physic_lib import *
from IPython.display import display, Math
init_printing()
```

Creating our Physics Object for kinematic

```
whit kinematic problems is important to declare
      x1,x2,y1,y2 pos initial and finall in x and y
      v , we used v when is rectilinius mov and
      v1,v2, when is vert or parabolic, ac=0 make uniform mov
      g gravity =g or 9.8 or 10 ...
      a = angle of fly..   =0 if not fly...
```

Basic kinematic problems for understand how work the library



V= 10 m/s and cte find t

```
t=symbols('t')
P=mparticle(x1=0,x2=20,t=t,v=10,ac=0,a=0) # moss relevant in this case
```

```
P.x_pos(t=0) # pos when t=0
```

0

```
P.x_pos(t=1) # pos when t=1
```

10.0

```
P.x_pos(t=2) # pos when t=1
```

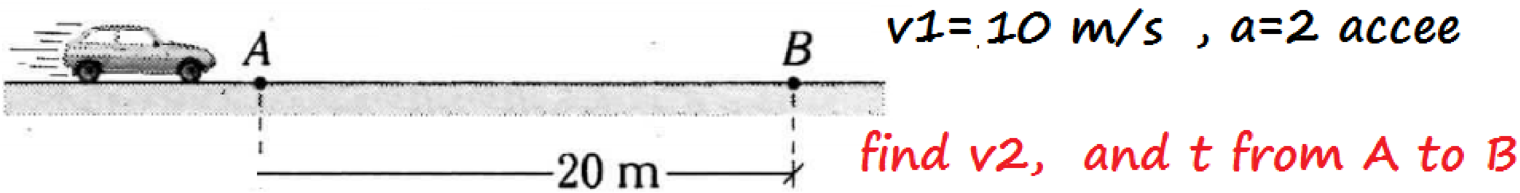
20.0

```
#and x_pos in time =t
P.x_pos() # pos when t=t
```

$10t$

```
csolve(P.x_pos()-20,t,'t')
```

$t = 2$
2




```
t,v2=symbols('t v2',positive=True)  # we use positive = True because pow is necessary
P=mparticle(x1=0,x2=20,y1=0,y2=0,t=t,v=10,v2=v2,ac=2,a=0) # acceleration=2 , angle=0
```

```
P.x_pos(t) # pos when t
```

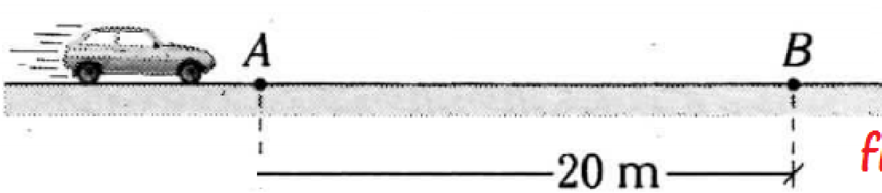
$t^2 + 10t$

```
csolve(P.x_pos(t)-20,t)
```

$-5 + 3\sqrt{5}$

kinematic classic functions

```
# ********     kinematic      ***********

    def x_vel(self,t=t,kope='',keval=True):

    def y_vel(self,t=t,kope='',keval=True):

    def xy_vel(self,tt='',kope='',keval=True):

    def x_pos(self, t=t,kope='',keval=True):

    def y_pos(self, t=t,kope='',keval=True):

    def y_max(self,t=t,keval=True,kope='',krelative=True):

    def x_max(self,kope='',keval=True):

    def t_fly(self,kope='',keval=True):

    def tan_angle_in_t(self,t=t,keval=True,kope='',ktan=True):
```



v1= 10 m/s , a=2 accee

find v2, and t from A to B

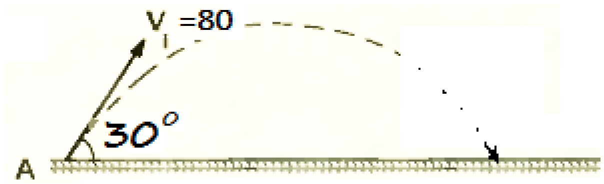
```
 # kope is option to simplify format answer and applicate in all library
csolve(P.x_pos(t)-20,t,kope='v')
```

1.70820393249937

```
P.x_vel(t=1.7) # finding vel final
```

13.4

Basic parabolic




```
P=mparticle(x1=0,y1=0,v=80,a=pi/6,g=10,ac=0) #  angle always in rad ac only for rect
```

```
P.t_fly()
```

$$8$$

```
P.x_max()
```

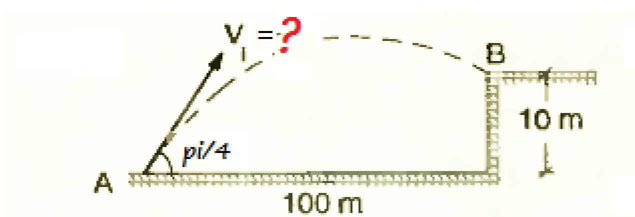
$$320\sqrt{3}$$

```
P.x_max(kope='v')
```

$$554.256258422041$$

```
P.y_max(kope='v')
```

$$80.0$$




```
t,V,x2,y2=symbols('t V x2 y2',positive=True)
P=mparticle(x1=0,y1=0,v=V,a=pi/4,g=10,ac=0,x2=x2,y2=y2) #  angle always in rad ac only for rect
```

```
P.x_pos(t)
```

$$\frac{\sqrt{2}Vt}{2}$$

```
T=csolve(P.x_pos(t)-100,t,'T') # find t like V function
```

$$T = \frac{100\sqrt{2}}{V}$$

```
P.store_val(t,T) #  store t in V form
```

```
P.store_val(t,T) #  store t in V form
```

```
csolve(P.y_pos(T)-10,V) # resolve V but..???
```

$$[]$$

```
csolve(P.y_pos(T)-10,V,'V',unifique=True)  # see option but we need positive then...
```

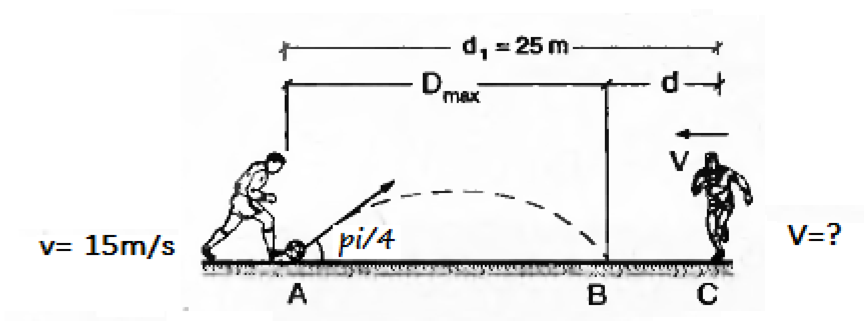
$$V = \left[-\frac{100}{3}, \frac{100}{3}\right]$$

$$\left[-\frac{100}{3}, \frac{100}{3}\right]$$

```
V=csolve(P.y_pos(T)-10,V,'V',unifique=True,kpositive=True) # nice.... f**k*ou symp..lib..
```

$$V = \frac{100}{3}$$

Parabolic whit two Object




```
t,V,x2=symbols('t V x2')
A=mparticle(v=15,x1=0,y1=0,g=10,a=pi/4,x2=x2,y2=0,ac=0)
C=mparticle(v=V,x1=25,y1=0,g=10,a=pi,x2=x2,y2=0,ac=0)
```

```
# total time is A.time fly..Ok then
T=A.t_fly();T
```

$$\frac{3\sqrt{2}}{2}$$

```
# xpos A + xpos C -25 =0 solve
A.x_pos(t)+C.x_pos(t)-25
```

$$-Vt + \frac{15\sqrt{2}t}{2}$$

```
# xpos A + xpos C -25 =0 solve and V
V=csolve(A.x_pos(t)+C.x_pos(t)-25,V,'V')
```

$$V = \frac{15\sqrt{2}}{2}$$

Little advance of real form solve Physic system problem using full atb lib

---

```
t,V,x2=symbols('t V x2')
A=mparticle(v=15,x1=0,y1=0,g=10,a=pi/4,x2=x2,y2=0,ac=0)
C=mparticle(v=V,x1=25,y1=0,g=10,a=pi,x2=x2,y2=0,ac=0)
e1=polyclass(A.x_pos(t)+C.x_pos(t),25)
e2=polyclass(A.t_fly(),t)
vec=[e1,e2]
P=Mpolyclass(vec)
P.csolves([t,V]) #  VVOOOAALLLAAAAAAAAAAA
```

$$\left[\left(\frac{3\sqrt{2}}{2}, \frac{15\sqrt{2}}{2}\right)\right]$$

```
# polyclass class create Eq to input in system Eq to solve see that
e1.Q
```

$$-Vt + \frac{15\sqrt{2}t}{2} + 25 = 25$$

```
e1.Q
```

$$-Vt + \frac{15\sqrt{2}t}{2} + 25 = 25$$

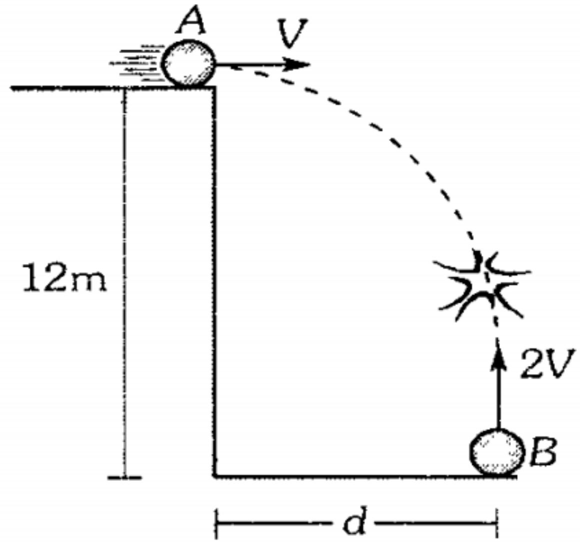
```
# and...
P.s()
```

$$\begin{aligned} -Vt + \frac{15\sqrt{2}t}{2} + 25 &= 25 \\ \frac{3\sqrt{2}}{2} &= t \end{aligned}$$

---

58. Dos esferas A y B son lanzadas simultáneamente tal como muestra la figura. La esfera A es lanzada horizontalmente con rapidez V y la esfera B verticalmente hacia arriba con rapidez $2V$. Determine **d** sabiendo que impactan. $(g=10\ m/s^2)$

A) 4 m B) 5 m C) 6 m
D) 7 m E) 8 m




```
V,d,t,x2,y2=symbols('V d t x2 y2')


A=mparticle(x1=0,x2=d,y1=12,y2=y2,v=V,a=0,g=10,ac=0)
B=mparticle(x1=0,x2=0,y1=0,y2=y2,v=2*V,a=pi/2,g=10,ac=0)


e1=polyclass(A.y_pos(),B.y_pos()) # creating A.y_pos = B.y_pos Equation
e2=polyclass(A.x_pos(),d) # creating A.y_pos = d Equation
vec=[e1,e2] # joined all eq in matrix
P=Mpolyclass(vec) # creating super class to make allllllll
P.s() # show the super Mpolyclass class
```

$$12-5t^2 = 2Vt-5t^2$$
$$Vt = d$$

```
# Two Eq whit two variablw ..then is possible to solve whit simple command like
P.subsolve(V*t,0,kremp=True) # means... find V*t from eq  0 and subs in all ..
```

$$Vt = 6$$
$$6 = d$$

0 s completado a las 21:12