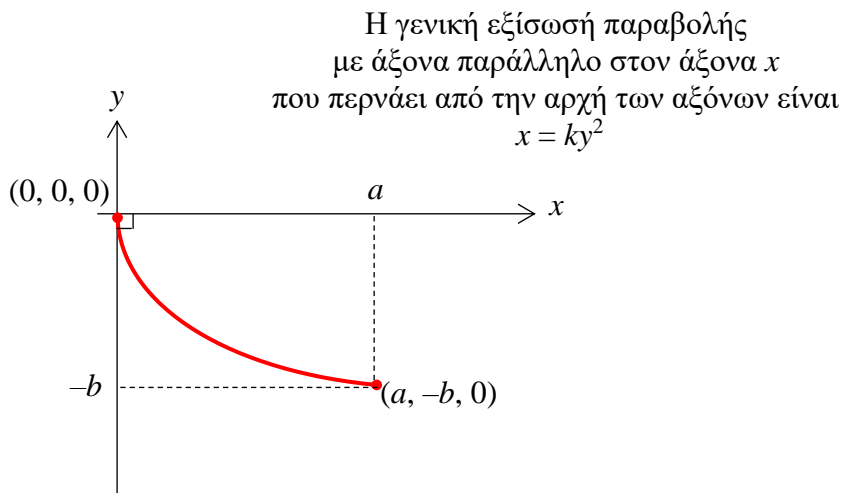


Πως σχεδιάζουμε μια καμπύλη στη VPython

Φτιάχνουμε μια λίστα με n σημεία από τα οποία περνάει η καμπύλη ($[\vec{r}_0, \vec{r}_1, \dots, \vec{r}_{n-1}]$) και ενώνουμε τα σημεία με ευθύγραμμα τμήματα θέτοντας τη λίστα ως τη θέση (pos) μιας καμπύλης (curve). Όσο πιο κοντά είναι τα σημεία τόσο πιο ομαλή θα είναι η καμπύλη.

Π.χ. έστω μια παραβολή στο επίπεδο x - y με την κορυφή της στην αρχή των αξόνων $(0, 0, 0)$, όπου έχει κατακόρυφη κλίση, η οποία περνάει και από το σημείο $(a, -b, 0)$.



Βρίσκουμε το k :

$$a = k(-b)^2 \Rightarrow k = \frac{a}{b^2}$$

και την εξίσωση του τμήματος της παραβολής που έχει αρνητικά y :

$$x = \frac{a}{b^2} y^2 \Rightarrow y = -b\sqrt{\frac{x}{a}}$$

Στην Glowscript (glowscript.org)

```

Web VPython 3.2
scene.background = color.white # κάνω το χρώμα της οθόνης άσπρο (αλλιώς είναι μαύρο)
scene.center = vector(pi/4,-0.5,0) # μετακινώ το κέντρο της οθόνης
a = 0.5*pi # Πρέπει να δώσουμε συγκεκριμένες
b = 1 # αριθμητικές τιμές στα a και b
parabola_list = [] # Φτιάχνουμε μια άδεια λίστα
n=100 # Αποφασίζουμε πόσα σημεία θα έχει η καμπύλη
dy = b/n # Βρίσκουμε πόσο θα απέχουν τα σημεία κατακόρυφα
y=0 # Η x συνιστώσα του πρώτου σημείου
# Φτιάχνουμε ένα βρόχο που υπολογίζει τα σημεία (x, y, 0) και τα προσθέτει (append) στη λίστα
while y<=b : # για όσο το y παραμένει μικρότερο ή το πολύ γίνει ίσο με το b
    parabola_list.append( vector(a*y**2/b, -y, 0) ) # πρόσθετε σημεία στη λίστα που να
    # ικανοποιούν την εξίσωση της παραβολής
    y=y+dy # πήγαινε στο επόμενο y
parabola = curve(pos=parabola_list, color=color.red) # φτιάξε την καμπύλη και δώσε της
# κόκκινο χρώμα
print( parabola.point(0).pos, parabola.point(99).pos ) # Τυπώνω το αρχικό και τελικό σημείο
# της καμπύλης
print(parabola_list[0], parabola_list[99]) # Ισοδύναμα μπορούσα να τυπώσω το αρχικό και τελικό

```

```

# στοιχείο της λίστας
print( vector(0,0,0), vector(a,-b,0)) # Τυπώνω την αρχή των αξονων (0,0,0) και το σημείο (a,-b,0)
# για να συγκρίνω

```

Επισημαίνουμε ότι τα στοιχεία της λίστας και της καμπύλης (που είναι 100) αριθμούνται ξεκινώντας από το μηδέν: [0], [1], [2], ... [99]. Άρα το 1^ο στοιχείο της λίστας είναι το `parabola_list[0]` και πρέπει να είναι το διάνυσμα `vector(0,0,0)`, ενώ αντίστοιχα το 1ο σημείο της καμπύλης αναφέρεται ως `parabola.point(0)` και πρέπει να έχει θέση `parabola.point(0).pos` πάλι το διάνυσμα `vector(0,0,0)`. Το τελευταίο στοιχείο της λίστας είναι το `parabola_list[99]` και το τελευταίο σημείο της καμπύλης είναι στη θέση `parabola.point(99).pos`. Και τα δύο πρέπει να είναι ίσα με `vector(a, -b, 0)`. Το πλήθος στοιχείων της λίστας (ή το μήκος της, `length`) αναφέρεται ως `len(parabola_list)` ενώ ο αριθμός σημείων της καμπύλης αναφέρεται ως `parabola.npoints`. Στην περίπτωσή μας `len(parabola_list) = 100` και `parabola.npoints = 100`.

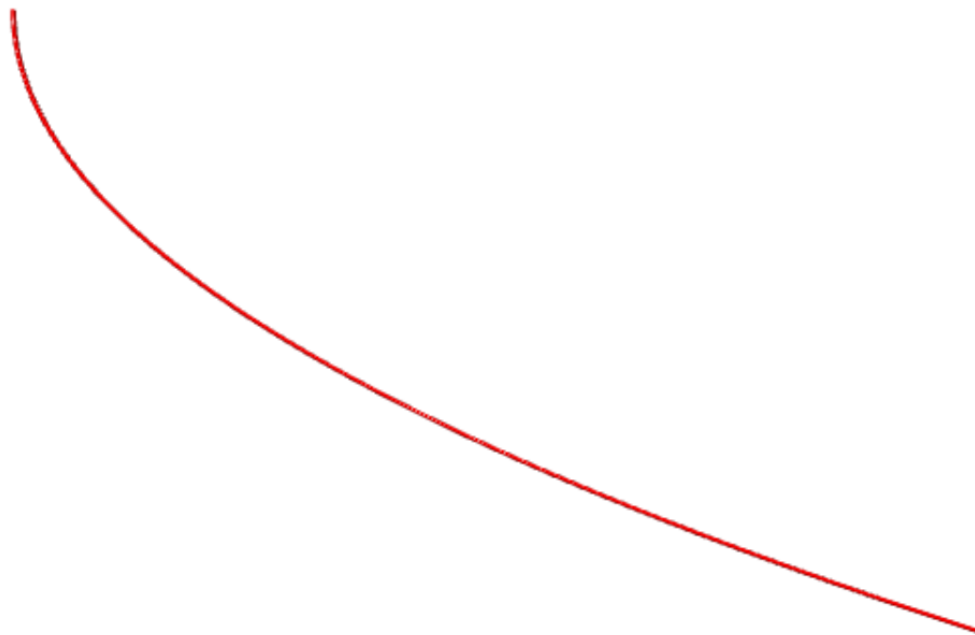
Οπότε για να τυπώσουμε το πρώτο και τελευταίο σημείο θα μπορούσαμε να γράψουμε `print(parabola.point(0).pos, parabola.point(parabola.npoint-1).pos)`

ή `print(parabola_list[0], parabola_list[len(parabola_list)-1])`

Χωρίς να χρειάζεται να βάζουμε με το χέρι το 99. Έτσι αν αποφασίσουμε να χρησιμοποιήσουμε 200 σημεία για να φτιάξουμε την καμπύλη, δεν θα χρειάζεται να αλλάζουμε με το χέρι το 99 σε 199 μέσα στα `print`.

Τρέχοντας το παραπάνω πρόγραμμα παίρνουμε την παρακάτω εικόνα

[Edit this program](#) [Screenshot](#)

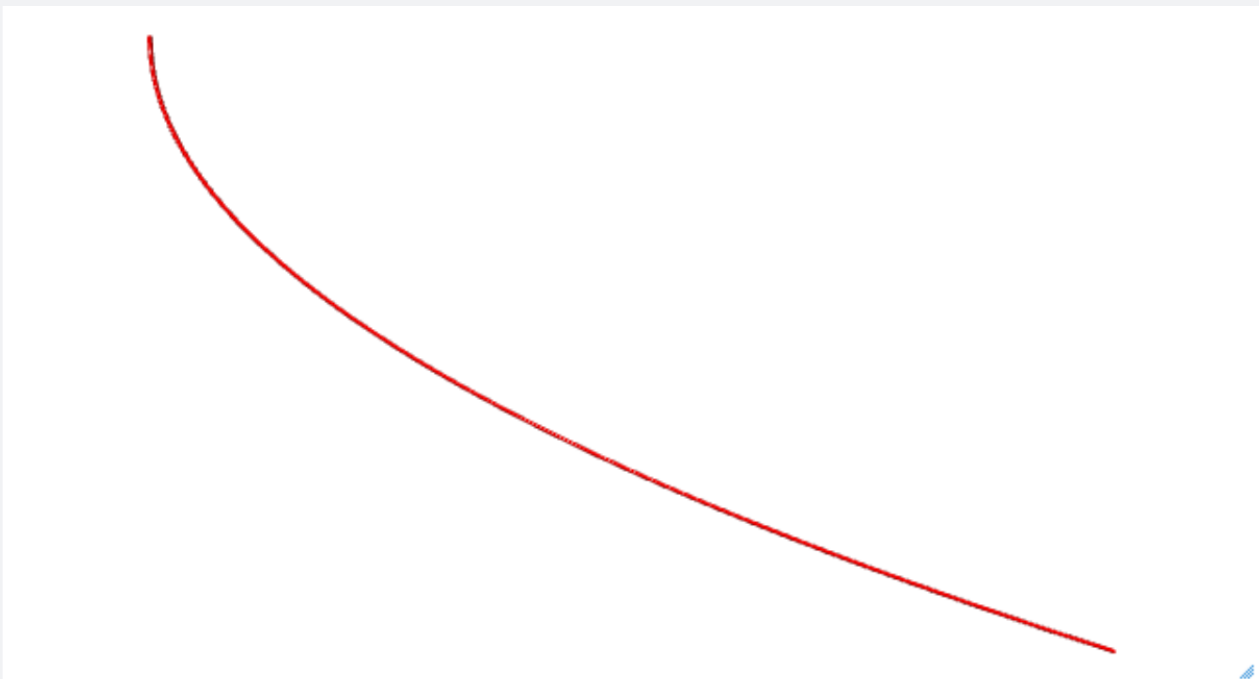


από καμπύλη: $\langle 0, 0, 0 \rangle \langle 1.53954, -0.99, 0 \rangle$
από λίστα : $\langle 0, 0, 0 \rangle \langle 1.53954, -0.99, 0 \rangle$
εξ' ορισμού : $\langle 0, 0, 0 \rangle \langle 1.5708, -1, 0 \rangle$

Τα σημεία της λίστας και της καμπύλης ταυτίζονται και είναι πολύ κοντά στις επιθυμητές τιμές που είχαμε θέσει $a = \pi/2 = 1.5708$ και $b = 1$. Αυτό σημαίνει ότι την τελευταία φορά που εκτελέστηκε ο βρόχος το b είχε την τιμή 0,99 και πήρε την τιμή $0.99 + dy$ στο τέλος του. Το $0.99+dy$ θα ήταν μεγαλύτερο από το 1 και άρα ο βρόχος δεν εκτελέστηκε ξανά και σταμάτησαν οι υπολογισμοί σημείων. Αν θέλουμε να πλησιάσουμε πιο κοντά στο $(1.5708, -1, 0)$ μεγαλώνουμε τον αριθμό σημείων στο οποίο χωρίζουμε το διάστημα 0 ως b από $n=100$ σε $n=1000$ και άρα μικραίνουμε το dy .

Π.χ. για $n=1000$ παίρνουμε

[Edit this program](#) [Screenshot](#)

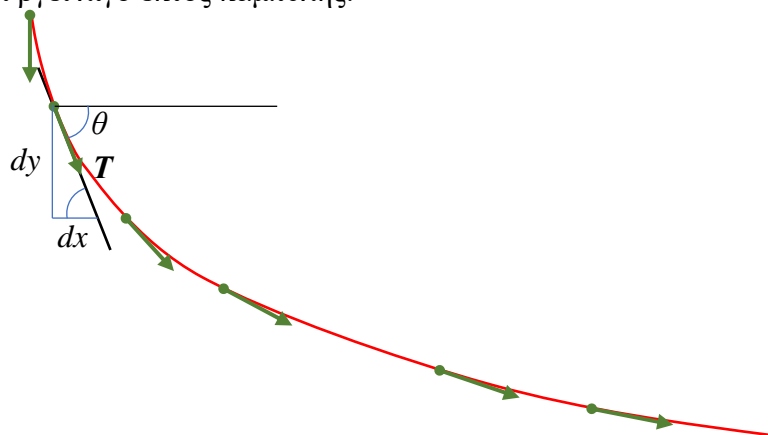


από καμπύλη: $\langle 0, 0, 0 \rangle \langle 1.56766, -0.999, 0 \rangle$
 από λίστα : $\langle 0, 0, 0 \rangle \langle 1.56766, -0.999, 0 \rangle$
 εξ' ορισμού : $\langle 0, 0, 0 \rangle \langle 1.5708, -1, 0 \rangle$

Οπτικά δεν καταλαβαίνουμε διαφορά. Η καμπύλη δεν αλλάζει αισθητά στο μάτι. Όμως το τελευταίο σημείο της είναι πιο κοντά στο σημείο $(1.5708, -1, 0)$. Από το $(1.53954, -0.99, 0)$ πήγε στο $(1.56766, -0.999, 0)$.

Πώς βάζουμε ένα σώμα να κινείται πάνω σε μια καμπύλη

Η ταχύτητα κάθε σώματος είναι στην κατεύθυνση της εφαπτόμενης στην τροχιά του. Έστω ότι το μέτρο v της ταχύτητάς του παραμένει σταθερό. Οπότε βάζουμε το σώμα να προχωρήσει για κάποιο μικρό dt κατά vdt πάνω στην εφαπτομένη της καμπύλης και σε κάθε μικρό βήμα διορθώνουμε τη θέση του που θα έχει βγει λίγο εκτός καμπύλης.



Χρειαζόμαστε το μοναδιαίο εφαπτόμενο διάνυσμα \mathbf{T} σε κάθε σημείο της καμπύλης. Αυτό το βρίσκουμε από την εξίσωση της καμπύλης. Η γωνία θ που σχηματίζει η εφαπτόμενη ευθεία με την καμπύλη είναι η παράγωγός της καμπύλης σε εκείνο το σημείο

$$\frac{dy}{dx} = \tan \theta \Rightarrow \theta = \arctan \left(\frac{dy}{dx} \right), \quad \hat{T} = (\cos \theta, \sin \theta, 0)$$

Στην περίπτωση της προηγούμενης παραβολής η παράγωγος είναι :

$$x = (a/b^2)y^2, \text{ με } y < 0 \Rightarrow dx = (a/b^2)2ydy \Rightarrow \frac{dy}{dx} = \left(\frac{b^2}{2a} \right) \frac{1}{y} < 0$$

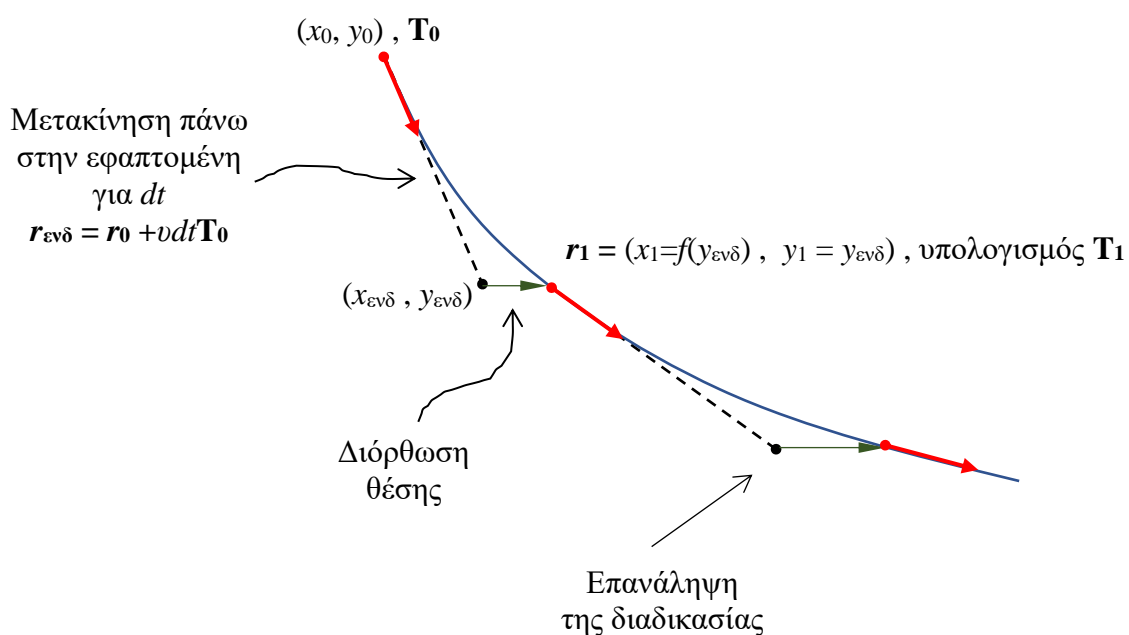
ή

$$y = -b\sqrt{\frac{x}{a}} = -b\left(\frac{x}{a}\right)^{1/2} \Rightarrow \frac{dy}{dx} = -b\left(\frac{1}{2}\right)\left(\frac{x}{a}\right)^{-1/2} \frac{1}{a} \Rightarrow \frac{dy}{dx} = -\frac{b}{2a}\left(-\frac{b}{y}\right) \Rightarrow \frac{dy}{dx} = \left(\frac{b^2}{2a}\right) \frac{1}{y}$$

Η γωνία θ θα είναι λοιπόν : $\theta = \arctan \left(\frac{b^2}{2ay} \right) < 0$ και το μοναδιαίο διάνυσμα θα δείχνει προς τα κάτω και δεξιά.

Η διαδικασία εξελίσσεται ως εξής :

- 1) Είμαστε σε μια αρχική θέση όπου έχουμε υπολογίσει το εφαπτόμενο διάνυσμα: \vec{r}_0, \vec{T}_0 ,
- 2) Προχωράμε σε μια επόμενη ενδιάμεση θέση : $\vec{r}_{ενδ} = \vec{r}_0 + v\vec{T}_0 dt$,
- 3) Κρατάμε την y -συνιστώσα αυτής της θέσης και βρίσκουμε το x_1 που αντιστοιχεί στο σημείο της καμπύλης με $y_1 = y_{ενδ}$ από την εξίσωση της καμπύλης : $x_1 = f(y_{ενδ})$,
- 4) Θέτουμε ως νέα θέση του σώματος την : $\vec{r}_1 = (x_1, y_1, 0)$,
- 5) Υπολογίζουμε τη νέα γωνία της εφαπτομένης και το εφαπτόμενο διάνυσμα \vec{T}_1 στη νέα θέση \vec{r}_1 ,
- 6) Μεταβαίνουμε στην επόμενη χρονική στιγμή: $t = t + dt$,
- 7) Επαναλαμβάνουμε όλα τα βήματα από το 1) θεωρώντας τη θέση που υπολογίσαμε ως νέα αρχική θέση μέχρι το σώμα να φτάσει την τελική θέση που θέλουμε πάνω στην καμπύλη.



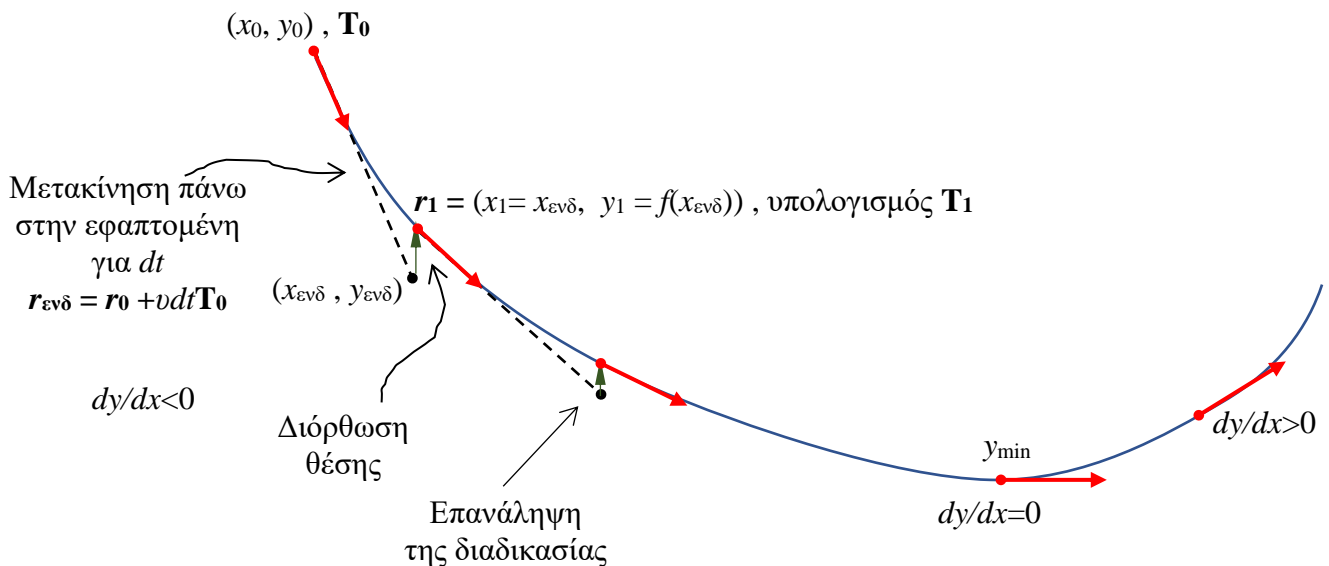
Ισοδύναμα θα μπορούσαμε να κρατήσουμε την x -συνιστώσα της ενδιάμεσης θέσης και από αυτήν να βρούμε την y -συνιστώσα του επόμενου σημείου της καμπύλης.

ΣΗΜΑΝΤΙΚΟ: πρέπει να επιλέγουμε να ακολουθήσουμε την συνιστώσα που είναι μονότονη. Αυτή που διαρκώς μεγαλώνει.

Αλλιώς με κάποιο τρόπο πρέπει να εξασφαλίσουμε ότι αν (if), κατά τους αριθμητικούς υπολογισμούς, ξεπεράσουμε το y_{\min} οπότε δεν θα μπορεί να υπολογιστεί κάποιο x να κάνουμε διόρθωση στο y ώστε να επανέλθουμε στην καμπύλη και να έχουμε πλέον θετική παράγωγο.

Π.χ. αν η καμπύλη είναι όπως στο παρακάτω σχήμα, θα πρέπει να επιλέξουμε την x -συνιστώσα ώστε να εξασφαλίσουμε ότι μετά το σημείο ελάχιστου y όπου $\frac{dy}{dx} = 0$ ο υπολογισμός της παραγώγου δίνει

θετικό αποτέλεσμα $\frac{dy}{dx} > 0$



Αν το μέτρο της ταχύτητας δεν παραμένει σταθερό τότε θα πρέπει να ξέρουμε πως αλλάζει και να το υπολογίζουμε κι αυτό σε κάθε σημείο της τροχιάς. Π.χ. αν το σώμα ξεκινάει από την ηρεμία και ολισθαίνει χωρίς τριβή υπό την επίδραση της βαρύτητας, τότε από τη διατήρηση της ενέργειας $v = \sqrt{-2g(y - y_0)}$.

Παράδειγμα κίνησης με σταθερή ταχύτητα μεταξύ των σημείων $(0,0)$ και $(a, -b)$ πάνω σε μια παραβολή και έναν κύκλο ο οποίος παρουσιάζει ελάχιστο

Στην Glowscript

```

Web VPython 3.2
scene.background = color.white
scene.center = vector(pi/4,-0.6,0)
# Η παραβολή
parabola_list = []
R=0.5
a=pi*R
b=2*R
n=1000
dy=b/n
y=0
while y<=b:
    x=a*y**2/b**2
    parabola_list.append(vector(x,-y,0))

```

```

y=y+dy
parabola = curve(pos=parabola_list, color = color.red )
# Ο κύκλος
circle_list = []
RC = (a**2+b**2)/(2*a)
phi = pi
phifin = 3*pi/2+atan((a-RC)/b)
dphi = (phifin-phi)/n
while phi<=phifin:
    xc = RC+RC*cos(phi)
    yc = RC*sin(phi)
    circle_list.append(vector(xc,yc,0) )
    phi=phi+dphi
circle = curve(pos=circle_list, color = color.green )
# Αρχικές θέσεις, εφαπτόμενα διανύσματα
# Παραβολής
ballp = sphere(pos=vector(0,0,0), radius = 0.02)
thetap = atan(b**2/2/a/ballp.pos.y)
ballp.T = vector(cos(thetap), -sin(thetap), 0)
attach_arrow(ballp, "T",scale=0.2,shaftwidth=0.01)
# Κύκλου
d=0.00001
ballc = sphere(pos=vector(d,-sqrt(RC**2-(RC-d)**2),0), radius = 0.02) # ελάχιστα κουννημένο
αλλιώς δεν θα αλλάξει το x στο loop αφού η ταχύτητα είναι κατακόρυφη
alpha = asin( abs( ballc.pos.y/RC ) ) # Η μικρότερη γωνία της ακτίνας (x,y) με τον άξονα x
ballc.T = vector(sin(alpha), -cos(alpha), 0)
attach_arrow(ballc, "T",scale=0.2,shaftwidth=0.01)
# Μέτρο ταχύτητας
speed= 0.2
print("speed=",speed, " [m/s]")
print("PRESS ANY KEY")
scene.waitfor('keydown')
# Ολίσθηση λόγω βαρύτητας χωρίς τριβή πάνω στην καμπύλη
t=0
dt=0.01
while ballc.pos.x<=a : # θεωρώ ότι η χάντρα στον κύκλο θα πηγαίνει πιο αργά
    rate(100)
    if ballp.pos.x<=a :
        intermediatep = ballp.pos+speed*dt*ballp.T
        xcorrectp =a*intermediatep.y**2/b**2
        ballp.pos = vector(xcorrectp, intermediatep.y,0)
        thetap = atan(b**2/2/a/ballp.pos.y)
        ballp.T = vector(cos(thetap), sin(thetap), 0)
        tp=t
# Στον κύκλο παρακολουθώ το x που πάντα μεγαλώνει και διορθώνω το y
intermediatec = ballc.pos+speed*dt*ballc.T
ycorrectc = -sqrt(RC**2-(RC-intermediatec.x)**2)
ballc.pos = vector(intermediatec.x, ycorrectc, 0)
alpha = asin( abs( ballc.pos.y/RC ) )
if ballc.pos.x < RC:
    ballc.T = vector(sin(alpha), -cos(alpha), 0)
if ballc.pos.x > RC:

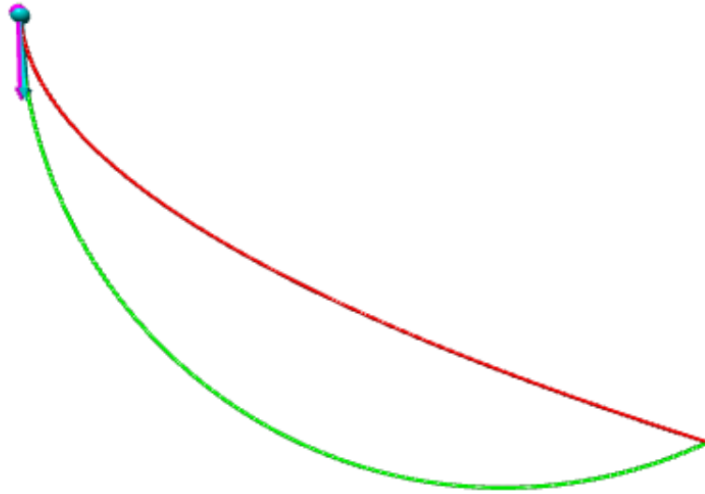
```

```

ballc.T = vector(sin(alpha), cos(alpha), 0)
t=t+dt
print("χρόνος διέλευσης παραβολής=",tp,"[s]","μήκος διαδρομής παραβολής=",speed*tp,"[m]")
print("χρόνος διέλευσης κύκλου = ",t,"[s]","μήκος διαδρομής κύκλου=",speed*t,"[m]")

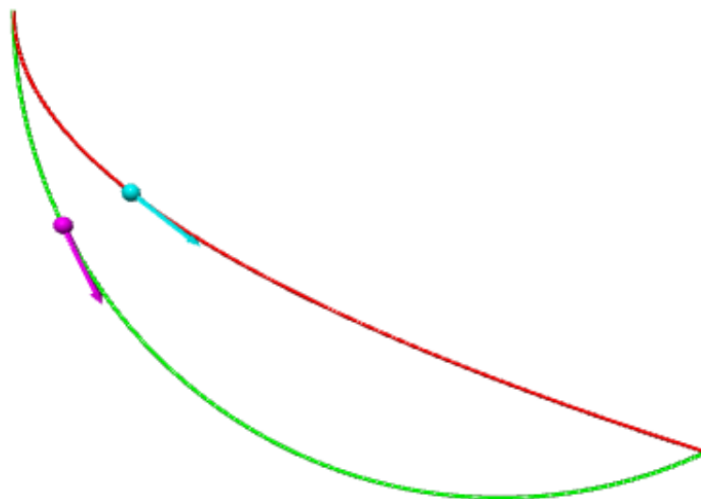
```

Edit this program Screenshot

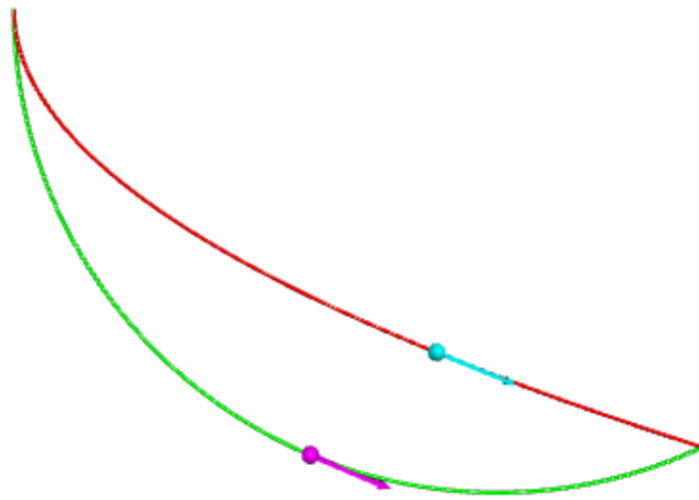


speed= 0.2 [m/s]
PRESS ANY KEY

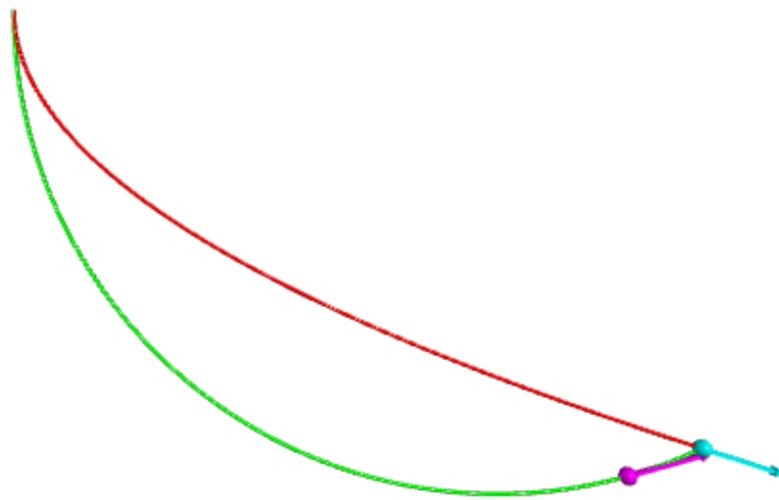
Edit this program Screenshot



speed= 0.2 [m/s]
PRESS ANY KEY

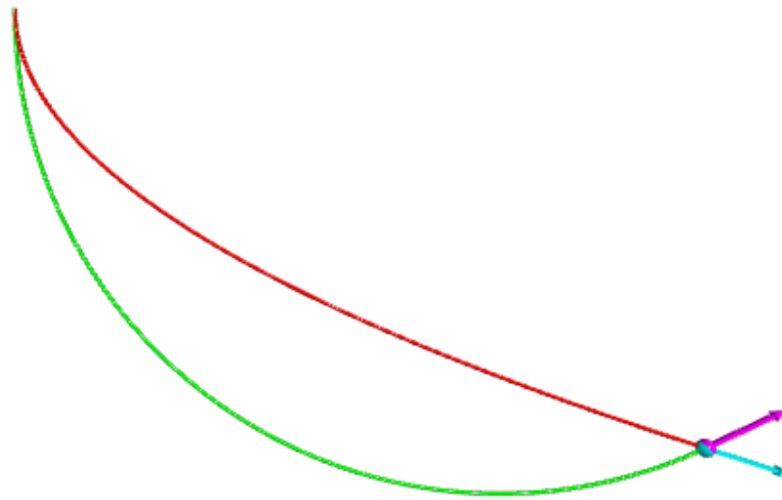
Edit this program Screenshot

speed= 0.2 [m/s]
PRESS ANY KEY

Edit this program Screenshot

speed= 0.2 [m/s]
PRESS ANY KEY

Edit this program Screenshot



speed= 0.2 [m/s]

PRESS ANY KEY

χρόνος διέλευσης παραβολής= 9.71 [s], μήκος διαδρομής παραβολής= 1.942 [m]

χρόνος διέλευσης κύκλου = 11.11 [s], μήκος διαδρομής κύκλου= 2.222 [m]