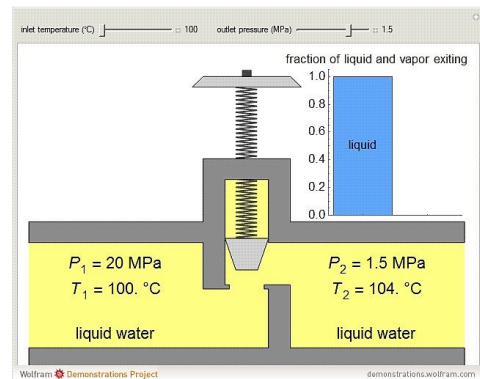
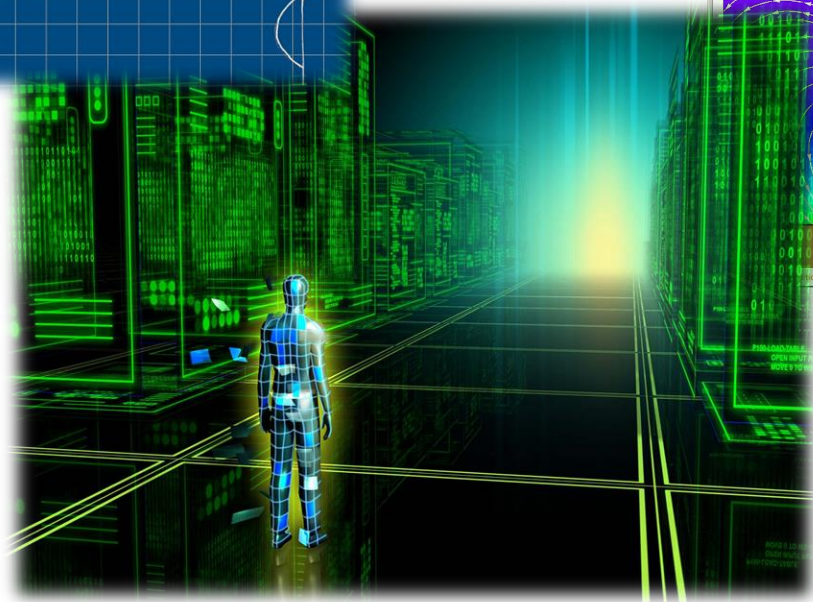
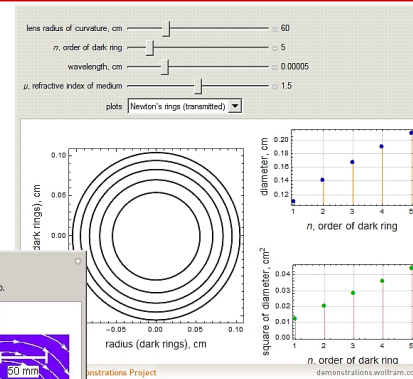
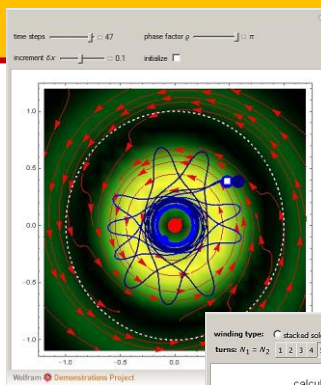
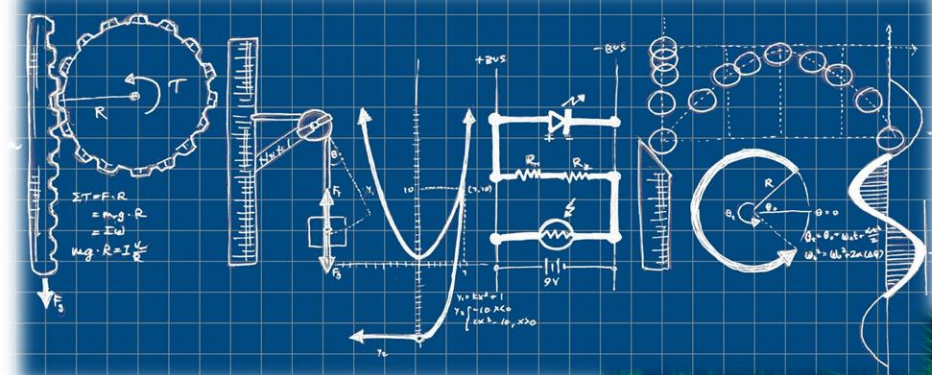
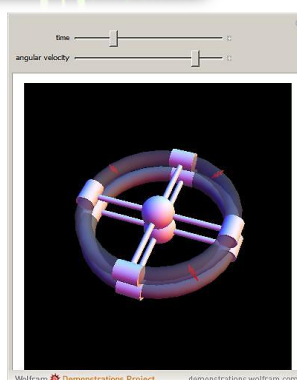
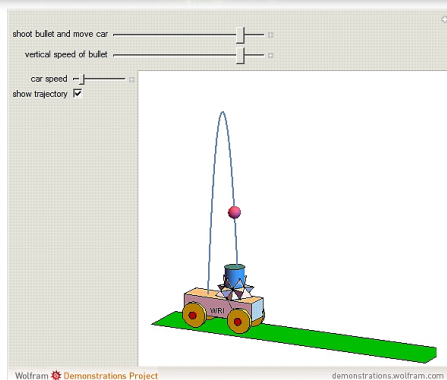
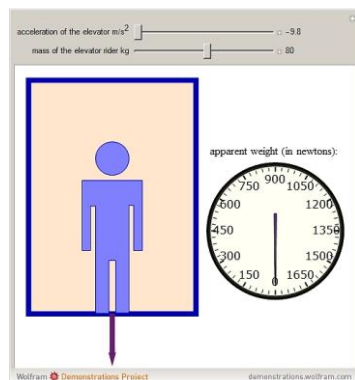




Ομάδα Γ.

Ο υπολογιστής ως επιστημονικό εργαλείο

Εργασία-10: Η mathematica στην υπηρεσία της Φυσικής

Wolfram
Mathematica®



0 | Προγραμματισμός με Mathematica

Μεθοδολογίες Η Mathematica διαθέτει μια υψηλού επιπέδου γλώσσα προγραμματισμού η οποία μας δίνει την δυνατότητα να δημιουργούμε τις δικές μας συναρτήσεις επεκτείνοντας με τον τρόπο αυτό τις δυνατότητες της γλώσσας αυτής. Η γλώσσα προγραμματισμού της Mathematica εκτός από την δυνατότητα να διαχειρίζεται σύμβολα και νούμερα, διαθέτει επίσης και τρεις διαφορετικές μεθοδολογίες προγραμματισμού:

Διαδικασιακός προγραμματισμός (procedural programming):

Στον διαδικασιακό προγραμματισμό πρέπει εμείς να περιγράψουμε, με την μορφή εντολών, τα βήματα που θα πρέπει να ακολουθήσει ο υπολογιστής για να επιλύσει ένα πρόβλημα. Συνήθως διαχωρίζουμε το πρόβλημα μας σε επιμέρους ανεξάρτητα προβλήματα (functions, procedures, subroutines) τα οποία αρχικά επιλύουμε και ελέγχουμε για την ορθότητα τους και στη συνέχεια τα συνθέτουμε ώστε να πάρουμε την επίλυση του αρχικού μας προβλήματος, μια μέθοδος γνωστή ως σχεδίαση από πάνω προς τα κάτω (top-down design).

Συναρτησιακός προγραμματισμός (functional programming):

Το επίκεντρο του συναρτησιακού προγραμματισμού είναι ο υπολογισμός εκφράσεων και όχι η εκτέλεση εντολών (ανάθεση τιμών σε μεταβλητές όπως στον διαδικασιακό προγραμματισμό). Οι εκφράσεις στην γλώσσα του συναρτησιακού προγραμματισμού δημιουργούνται με την χρήση συναρτήσεων οι οποίες δέχονται ως ορίσματα απλές τιμές ή άλλες συναρτήσεις.

Κανονοκεντρικός προγραμματισμός (rule-based programming):

Στον κανονοκεντρικό προγραμματισμό η εκτέλεση των εντολών δεν είναι ακολουθιακή όπως στον διαδικασιακό προγραμματισμό, αλλά βασίζεται στην ενεργοποίηση συγκεκριμένων συνθηκών. Ο όρος κανόνας (rule) στον κανονοκεντρικό προγραμματισμό αναφέρετε στον ορισμό της συνάρτησης. Ένα κανονοκεντρικό πρόγραμμα αποτελείται από ένα σύνολο ορισμών συναρτήσεων τα οποία όμως αφορούν πάντα την ίδια συνάρτηση. Κάθε ορισμός αναφέρεται σε διαφορετική μορφή που μπορούν να έχουν τα ορίσματα της συνάρτησης.

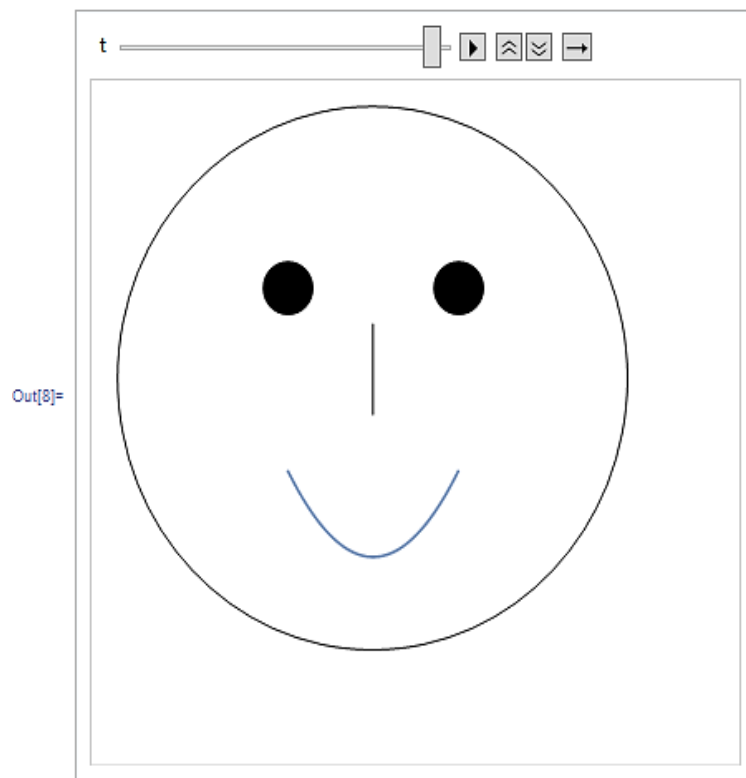


0 | Προγραμματισμός με Mathematica

Διαδραστική Απεικόνιση

Η Mathematica μπορεί να χρησιμοποιηθεί για την επίλυση υπολογιστικών προβλημάτων και την οπτικοποίηση των αποτελεσμάτων τους σε πραγματικό χρόνο, ώστε να γίνει πλήρως κατανοητή η εξέλιξη φαινομένων ή συσχετίσεων μεταξύ δεδομένων.

```
Animate[Plot[t (x^2 - 1) + (1 - t) (-x^2), {x, -1, 1}, Axes -> False, Epilog -> {Circle[{0, 1}, 3], Disk[{1, 2}, .3], Disk[{-1, 2}, .3],  
Line[{{0, 0.6}, {0, 1.6}}]}, PlotRange -> {{-3, 4}, {-3, 4}}, AspectRatio -> 1], {t, 0, 1}, AnimationRunning -> False]
```



Ρυθμίζοντας κατάλληλα τις παραμέτρους υπολογισμού (Evaluation) και αξιοποιώντας τα εργαλεία διαδραστικών οπτικοποιήσεων (Manipulate, Animate), τα αποτελέσματα υπολογίζονται γρήγορα και με ακρίβεια και απεικονίζονται με εποπτικό τρόπο.



0 | Προγραμματισμός με Mathematica

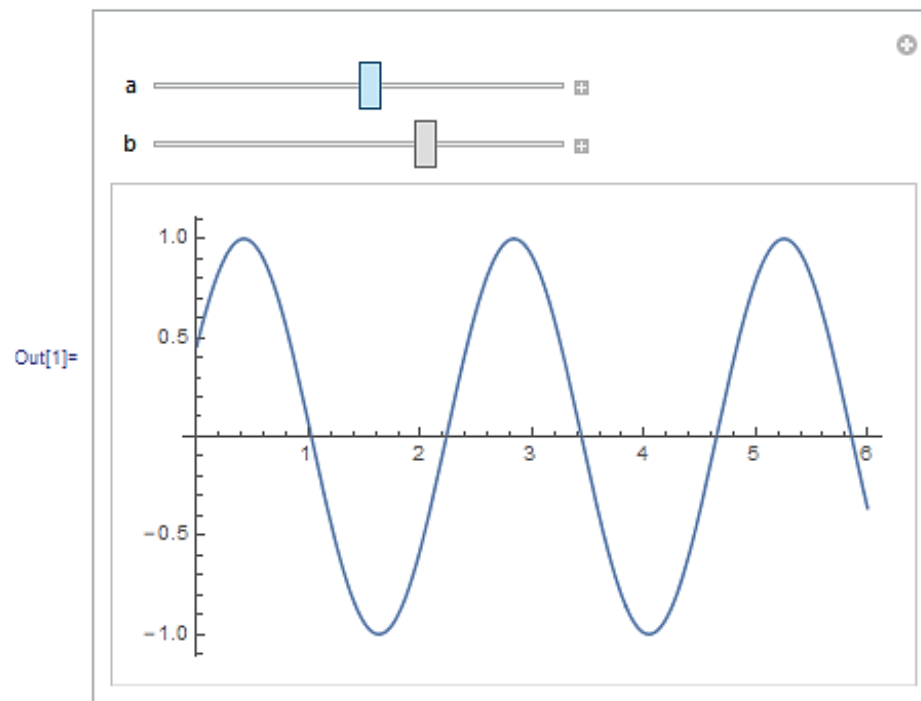
Διαδραστική Απεικόνιση

Η εντολή `Manipulate` μας επιτρέπει να δημιουργήσουμε ένα εκπληκτικό εύρος διαδραστικών γραφικών με λίγες μόνο γραμμές εισόδου. Η εντολή `Manipulate` έχει σχεδιαστεί, ώστε να είναι εύχρηστη και με ανάλογο συντακτικό, όπως οι βασικές εντολές π.χ. `Table` ή `Plot` και δεν απαιτεί ιδιαίτερες γνώσεις προγραμματισμού.

Η έξοδος που λαμβάνουμε από μια εντολή `Manipulate` είναι ένα διαδραστικό αντικείμενο που περιέχει ένα ή περισσότερα στοιχεία ελέγχου που μπορούμε να χρησιμοποιήσουμε για να μεταβάλετε την τιμή μιας ή περισσότερων παραμέτρων.

Η έξοδος μοιάζει πολύ με μικροεφαρμογή και έχει το βασικό χαρακτηριστικό ότι δεν είναι ένα στατικό αποτέλεσμα, αλλά μια δυναμική απεικόνιση, με την οποία μπορείτε να αλληλεπιδράσετε και να δείτε τα αποτελέσματά της σε πραγματικό χρόνο.

```
In[1]:= Manipulate[Plot[Sin[a x + b], {x, 0, 6}], {a, 1, 4}, {b, 0, 10}]
```





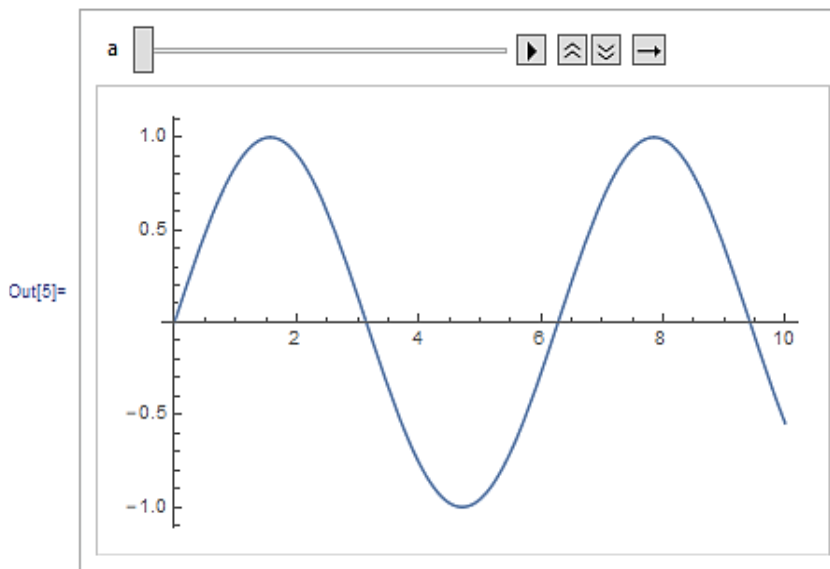
0 | Προγραμματισμός με Mathematica

Διαδραστική Απεικόνιση

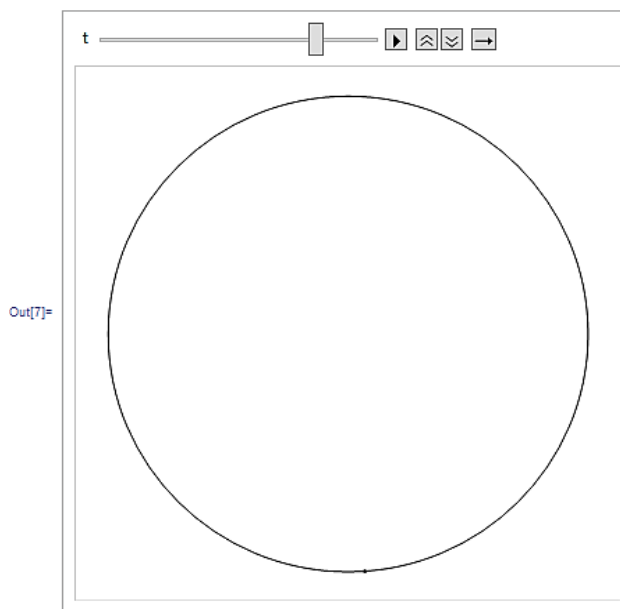
Η εντολή `Animate` αποτελεί ένα χρήσιμο εργαλείο που δημιουργεί ένα κινούμενο σχέδιο με βάση κάποιες εντολές γραφικών απεικονίσεων, δίνοντας ένα πιο ολοκληρωμένο αποτέλεσμα. Παρέχει τα κλασικά εργαλεία προβολής μιας ταινίας (`Play`, `Stop`, `Faster`, `Slower` και `Stop Carre`).

Η βασική της διαφορά από τη `Manipulate` είναι η δυνατότητα της `Manipulate` να έχει εργαλεία χειρισμού που να δείχνουν επίδραση παραμέτρων σε διαδοχικά στατικά στιγμιότυπα, ενώ η `Animate` παρέχει μια κινούμενη απεικόνιση μιας προεπιλεγμένης διαδικασίας.

```
In[5]:= Animate[Plot[Sin[x + a], {x, 0, 10}], {a, 0, 5}, AnimationRunning -> False]
```



```
In[7]:= Animate[Graphics[{Circle[], Point[{Cos[t], Sin[t]}]}], {t, 0, 6}, AnimationRunning -> False]
```





0 | Προγραμματισμός με Mathematica

Αλγόριθμος

Συνήθως ένας αλγόριθμος που αποτελεί το υπολογιστικό κομμάτι ενός προγράμματος περιέχει μια σειρά από εντολές, στην προκειμένη περίπτωση της Mathematica. Σύμφωνα με τον αντικειμενοστραφή προγραμματισμό, το πρόβλημα που έχουμε αντιμετωπίσουμε επιμερίζεται σε μικρότερα προβλήματα που το καθένα τους παίρνει δεδομένα και παράγει αποτελέσματα. Κάθε τέτοιο αντικείμενο μπορεί να είναι ένα αυτοτελές πρόγραμμα, μια ανεξάρτητη διαδικασία με την ίδια δομή που απλά να καλείται από το πρόγραμμα που γράφουμε.

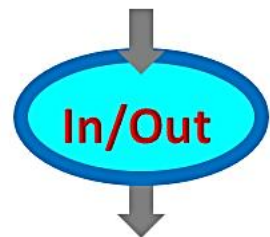
Κάτι ανάλογο δεν κάνουμε εξάλλου κάθε φορά που εκτελούμε μια εντολή της Mathematica;

Καλούμε τη συγκεκριμένη εντολή δίνοντας της τα ορίσματα και τις παραμέτρους που χρειάζεται για να εκτελεστεί σωστά, δηλ. τις παρέχουμε τα δεδομένα εισόδου που χρειάζεται για να λειτουργήσει για να μας δώσει το τελικό της αποτέλεσμα, είτε αυτός είναι ένας υπολογισμός ή μια γραφική απεικόνιση.

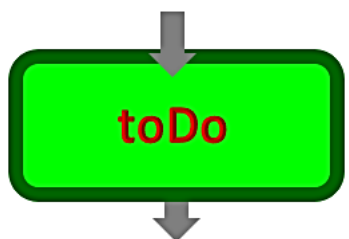


0 | Προγραμματισμός με Mathematica

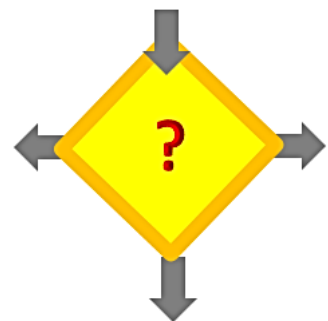
Αλγόριθμος Ένας αλγόριθμος/πρόγραμμα περιλαμβάνει αντικείμενα τριών διαφορετικών τύπων ανάλογα με τις ανάγκες που εξυπηρετούν.



την είσοδο ή έξοδο δεδομένων που γίνεται συνήθως με εντολές όπως η **Input**, **Import**, **Print**, **Plot** και αποτελεί την επικοινωνία του αλγόριθμου με τον «έξω» κόσμος.



τις διαδικασίες υπολογισμού που μπορεί να είναι μικρά αυτόνομα τμήματα προγράμματος ή έτοιμες συναρτήσεις που διεξάγουν υπολογισμούς για μία ή περισσότερες φορές. Κάθε τέτοιο υποπρόγραμμα έχει ένα μοναδικό όνομα με το οποίο καλείται από κάποιο σημείο ενός βασικού προγράμματος ή από άλλο υποπρόγραμμα και εκτελεί μια συγκεκριμένη λειτουργία, ενώ στο τέλος του, υπάρχει η εντολή **Return** που δηλώνει ρητή επιστροφή στην αμέσως επόμενη εντολή του προγράμματος που το κάλεσε. Εντός των διαδικασιών υπολογισμού συνήθως περιλαμβάνονται τις αναδρομικές εντολές **Do**, **For**, **While** που ανάλογα με ένα μετρητή ή μια συνθήκη επαναλαμβάνουν περισσότερες φορές μια σειρά εντολών.



τον έλεγχο ροής όπου εντολές τύπου **If** ανάλογα με την συνθήκη παρεμβαίνουν στη ροή του προγράμματος και το οδηγούν σε διαφορετικό σημείο. Η εντολή **If[condition,T, F]** οδηγεί σε διαφορετικό αποτέλεσμα ανάλογα με την συνθήκη που εισάγεται και αν αυτή καταλήγει σε αληθή (**True**) ή ψευδή συνθήκη (**False**). Όπως φαίνεται στα παραδείγματα που ακολουθούν, περισσότερες εντολές **If** μπορούν να συνδυαστούν ώστε να προβλέπουν διάφορες υποπεριπτώσεις ή να χρησιμοποιηθούν και άλλες εντολές όπως η **Which** ή η **Piecewise**.



0 | Προγραμματισμός με Mathematica

Αναδρομική Σχέση

- ❑ Στα μαθηματικά, μια αναδρομική σχέση είναι μια εξίσωση που εκφράζει τον n° όρο μιας ακολουθίας ως συνάρτηση των k προηγούμενων όρων, για κάποιο σταθερό k (ανεξάρτητο από το n), το οποίο ονομάζεται τάξη της σχέσης. Μόλις δοθούν k αρχικοί όροι μιας ακολουθίας, η σχέση επανάληψης επιτρέπει τον αναδρομικό υπολογισμό όλων των όρων της ακολουθίας.
- ❑ Αριθμητική πρόοδος είναι η ακολουθία, στην οποία για δύο διαδοχικούς όρους της a_n, a_{n+1} ισχύει ότι $a_{n+1} = a_n + \omega$, όπου ω μία σταθερή ποσότητα. Η ποσότητα ω ονομάζεται διαφορά της αριθμητικής πρόοδου. Αν $\omega > 0$, η αριθμητική πρόοδος είναι γνησίως αύξουσα. Αν $\omega < 0$, η αριθμητική πρόοδος είναι γνησίως φθίνουσα. Αν $\omega = 0$, η αριθμητική πρόοδος είναι σταθερή.
- ❑ Η αριθμητική πρόοδος ορίζεται με δύο ισοδύναμους τρόπους:
 - ❑ Γενικός τύπος: $a_n = a_1 + (n-1)\omega$, όπου ορίζεται ο νιοστός όρος συναρτήσει του πρώτου όρου και της διαφοράς.
 - ❑ Αναδρομικός τύπος: $a_n = a_{n-1} + \omega$, όπου ορίζεται ο νιοστός όρος συναρτήσει του προηγούμενου όρου και της διαφοράς.



0 | Προγραμματισμός με Mathematica

Στιγμιαίες μεταβολές φυσικών μεγεθών

❑ Στα φυσική, ως στιγμιαία ταχύτητα $\vec{v} = \frac{d\vec{r}}{dt}$ ενός σώματος ορίζεται ο ρυθμός μεταβολής της θέσης του $\vec{r} = x\hat{i} + y\hat{j} + z\hat{k}$ ως προς το χρόνο, όπως αυτή μετράται σε ένα δεδομένο σύστημα συντεταγμένων Οxyz. Είναι μέγεθος διανυσματικό, δηλαδή χαρακτηρίζεται τόσο από το μέτρο (μέγεθος) της, όσο και από τη φορά (κατεύθυνση) της.

❑ Αν θέλουμε να υπολογίσουμε το μέτρο της ταχύτητας v κάθε χρονική στιγμή t θα πρέπει να βρούμε τα μέτρα των συνιστωσών της (v_x, v_y, v_z) από σχέσεις του τύπου:

$$v_x = \frac{x[t+\Delta t]-x[t]}{\Delta t} \quad v_y = \frac{y[t+\Delta t]-y[t]}{\Delta t} \quad v_z = \frac{z[t+\Delta t]-z[t]}{\Delta t}$$

επιλέγοντας κατάλληλα το $\Delta t \rightarrow 0$ ώστε να είναι αξιόπιστος ο υπολογισμός

και στη συνέχεια το μέτρο της με τη βοήθεια της σχέσης $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$

❑ Αντίστοιχα και για την επιτάχυνση σε περιπτώσεις που δεν είναι σταθερή $a_i = \frac{v_i[t+\Delta t]-v_i[t]}{\Delta t}$, όπου $i=x, y, z$



0 | Προγραμματισμός με Mathematica

Επαναλαμβάνοντας διαδικασίες

- ❑ Στις περιπτώσεις αναδρομικών σχέσεων και στιγμιαίων τιμών μεγεθών, χρειάζεται να υπολογίσουμε ένα μεγάλο αριθμό τιμών ακολουθώντας μια συγκεκριμένη διαδικασία δηλ. τον ίδιο μαθηματικό τύπο, όπου αλλάζουν οι τιμές κάποιων παραμέτρων.
- ❑ Στις γλώσσες προγραμματισμού, αυτή η διαδικασία καλείται βρόχος επανάληψης και υλοποιείται εύκολα, δηλώνοντας μία φορά τον τύπο και μεταβάλλοντας τις κατάλληλες παραμέτρους, υπολογίζουμε n φορές το αποτέλεσμα.
- ❑ Στο δομημένο προγραμματισμό, χρησιμοποιούνται επαναληπτικές δομές, όταν χρειάζεται σειρές εντολών να εκτελούνται πάνω από μία φορά π.χ. ώστε μια αναδρομική σχέση να υπολογίσει τον $n^{\text{οστο}}$ της όρο.
- ❑ Η εντολή `For[start, test, incr]` επαναλαμβάνει μια σειρά εντολών για όσες φορές ικανοποιείται η συνθήκη (test) και την αρχική τιμή (start) και το βήμα (incr).
- ❑ Η διακοπή ενός βρόχου επανάληψης γίνεται είτε με την ολοκλήρωση του ή με μια εντολή `Break[]` ενώ η επιστροφή στην επαναληπτική διαδικασία γίνεται με την εντολή `Continue[]`.

```
In[1]:= For[i = 0, i < 4, i++, Print[i]]
          |βρόχος For          |εκτύπωση
          0
          1
          2
          3
```



0 | Προγραμματισμός με Mathematica

Επαναλαμβάνοντας διαδικασίες

- ❑ Η πιο κλασσική εντολή για δομή επανάληψης είναι η εντολή **Do** με τη βοήθεια της οποίας μπορούμε να επαναλαμβάνουμε, για συγκεκριμένο αριθμό επαναλήψεων, υπολογισμούς τιμών συναρτήσεων ή διαδικασίες προκειμένου π.χ. να συμπληρωθεί ένας πίνακας τιμών για τις συντεταγμένες της θέσης ενός κινητού.
- ❑ Η εντολή **Do** χρειάζεται μία ή περισσότερες εντολές και ένα μετρητή που να καθορίζει το πλήθος των επαναλήψεων που θα εκτελεί.
- ❑ Το παράδειγμα δείχνει την εμφάνιση στην οθόνη (Print) ακεραίων αριθμών που ξεκινούν από το -3, σταματούν στο 5 και έχουν βήμα 2. Δηλ. -3, -1, 1, 3, 5.

```
In[1]:= Do[Print[n], {n, -3, 5, 2}]
```

-3

-1

1

3

5



0 | Προγραμματισμός με Mathematica

Επαναλαμβάνοντας διαδικασίες ☐ Η Do συντάσσεται ως εξής:

Do (βρόχος Do)

Do[Εντολή 1; Εντολή 2; Εντολή 3, {μεταβλητή, αρχική τιμή, τελική τιμή, βήμα}]

Do[*expr*, *n*]
evaluates *expr* *n* times.

Do[*expr*, {*i*, *i_{max}*}]
evaluates *expr* with the variable *i* successively taking on the values 1 through *i_{max}* (in steps of 1).

Do[*expr*, {*i*, *i_{min}*, *i_{max}*}]
starts with *i* = *i_{min}*.

Do[*expr*, {*i*, *i_{min}*, *i_{max}*, *di*}]
uses steps *di*.

Do[*expr*, {*i*, {*i₁*, *i₂*, ...}}]
uses the successive values *i₁*, *i₂*,

Do[*expr*, {*i*, *i_{min}*, *i_{max}*, {*j*, *j_{min}*, *j_{max}*, ...}]
evaluates *expr* looping over different values of *j* etc. for each *i*.

εκτελεί μία ή περισσότερες εντολές σύμφωνα με τις τιμές της μεταβλητής και ολοκληρώνεται όταν η μεταβλητή φθάσει την τελική της τιμή

Ένα κλασσικό πρόβλημα είναι οι ατέρμονες βρόχοι, που σημαίνει ότι δεν ολοκληρώνεται η Do γιατί δεν λαμβάνει η μεταβλητή την τελική τιμή. Στην περίπτωση αυτή επιλέγουμε *Evaluation-Quit Kernel* και ελέγχουμε το λάθος στις μεταβολές της μεταβλητής.

Το παράδειγμα δείχνει τον υπολογισμό των στιγμιαίων ταχυτήτων (V_x , V_y) και θέσεων ενός κινητού (x, y) για κίνηση σε δύο διαστάσεις (x, y) για διαφορετικές τιμές του χρόνου t , από την τιμή t_i μέχρι την τιμή t_f με βήμα Δt

Do[Vx[t + Δt] = Vx[t];

|βρόχος Do

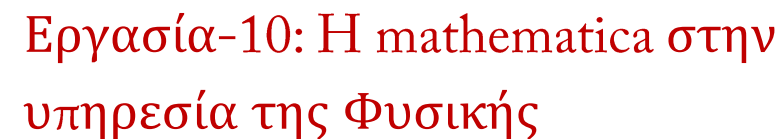
x[t + Δt] = x[t] + Vx[t + Δt] * Δt;

Vy[t + Δt] = Vy[t] + g * Δt;

y[t + Δt] = y[t] + Vy[t + Δt] * Δt, {t, ti, tf, Δt}]

Οι τιμές των t_i , t_f , Δt έχουν οριστεί πριν την εντολή Do

Πειράματα Επίδειξης

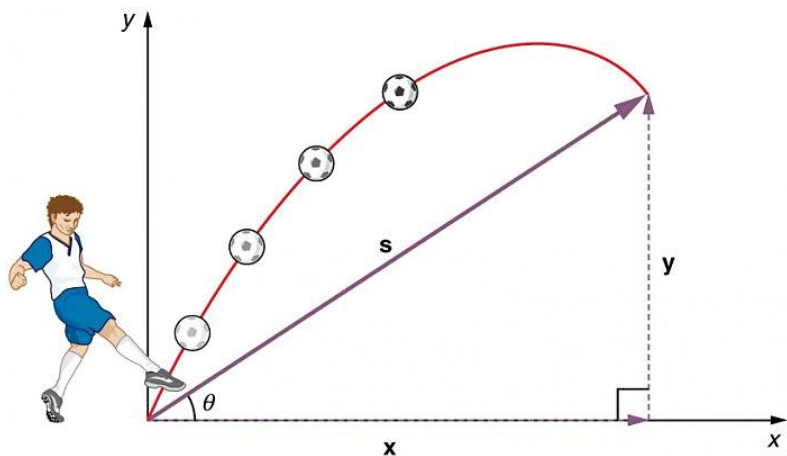


Wolfram
Mathematica® 10



1 | Πρόβλημα 10.1: Βολές

Το πρόβλημα



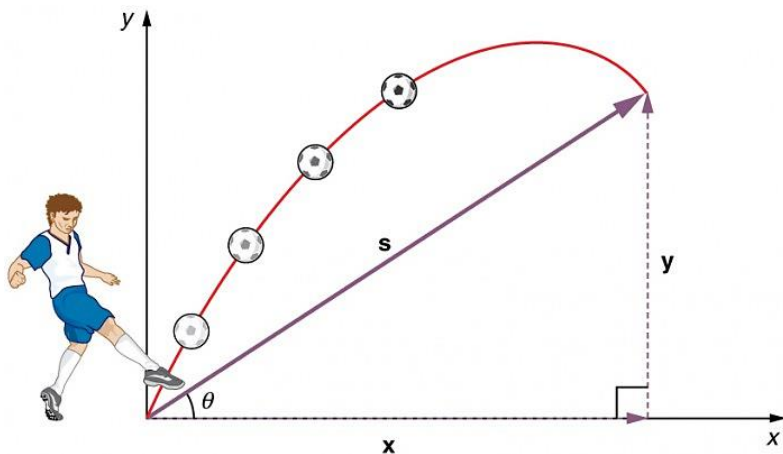
□ Η μελέτη των βολών συνήθως αναφέρεται σε σφαιρικά αντικείμενα που εκτελούν μια σύνθετη κίνηση αποτελούμενη από δύο συνιστώσες: μια οριζόντια κίνηση παράλληλη στην επιφάνεια της γης και μια κατακόρυφη κίνηση.

□ Τα προβλήματα των βολών έχουν να κάνουν συνήθως με την απεικόνιση της τροχιάς για συγκεκριμένο χρονικό διάστημα και τον υπολογισμό φυσικών μεγεθών για διάφορες χρονικές στιγμές



1 | Πρόβλημα 10.1: Βολές

Η φυσική του προβλήματος



- Το σύστημα αναφοράς για τη μελέτη των βολών είναι ένα σύστημα συντεταγμένων x-y.
- Σύμφωνα με την αρχή της ανεξαρτησίας των κινήσεων η κάθε κίνηση είναι ανεξάρτητη της άλλης.

- Η θέση και η **στιγμιαία** ταχύτητα του κινητού προκύπτουν από το διανυσματικό άθροισμα των συνιστωσών τους
- που υπολογίζονται από **αναδρομικές σχέσεις** από την αμέσως προηγούμενη κατάσταση.

Άξονας x:

$$a_x = 0$$

$$v_x[t + \Delta t] = v_x[t] + a_x \Delta t$$

$$x[t + \Delta t] = x[t] + v_x \Delta t$$

Άξονας y:

$$a_y = -g$$

$$v_y[t + \Delta t] = v_y[t] + a_y \Delta t$$

$$y[t + \Delta t] = y[t] + v_y \Delta t$$

$$v = \sqrt{v_x^2 + v_y^2}$$

$$\theta = \tan^{-1} \frac{v_y}{v_x}$$

$$r = \sqrt{x^2 + y^2}$$



1 | Πρόβλημα 10.1: Βολές

Συνθήκες-Σταθερές-Παράμετροι

- ❑ **Κίνηση στον άξονα των x :** Η ταχύτητα v_x είναι σταθερή αν δεν υπάρχει η αντίσταση του αέρα, οπότε $a_x=0$.
- ❑ **Κίνηση στον άξονα των y :** Η επιτάχυνση είναι η επιτάχυνση της βαρύτητας $g=-9.8 \text{ m/s}^2$ αν θεωρήσουμε θετική την κατεύθυνση του άξονα y προς τα πάνω.
- ❑ Σε χρονικό διάστημα Δt , ο χρόνος έχει μεταβληθεί από t σε $t+\Delta t$
- ❑ Άξονας x : $a_x=0$, $v_x[t+\Delta t]=v_x[t]+a_x\Delta t$, $x[t+\Delta t]=x[t]+v_x\Delta t$
- ❑ Άξονας y : $a_y=-g$, $v_y[t+\Delta t]=v_y[t]+a_y\Delta t$, $y[t+\Delta t]=y[t]+v_y\Delta t$
- ❑ Οι στιγμιαίες τιμές της τιμές της ταχύτητας, του διαστήματος αποθηκεύονται σε κελιά πίνακα όπου ο δείκτης θέσης του κελιού δεν είναι κάποιος ακέραιος αριθμός αλλά ο χρόνος t τη συγκεκριμένη χρονική στιγμή Π.χ. $v_x[3.2]=4.5$ σημαίνει ότι η στιγμιαία ταχύτητα στον άξονα x τη χρονική στιγμή 3.2 s είναι 4.5 m/s



1 | Πρόβλημα 10.1: Βολές-Οριζόντια Βολή

- ❑ Ας θεωρήσουμε την απλή περίπτωση όπου ένα σώμα εκτελεί οριζόντια βολή με ταχύτητα $u_x=10 \text{ m/sec}$ και $u_y=0$ και το κινητό ξεκινά από την αρχή των αξόνων ($x=0, y=0$).
- ❑ Για το πρόγραμμα αυτό θα χρειαστεί να υπολογίσουμε τα χρονικώς μεταβαλλόμενα μεγέθη x, y, u_x, u_y (στην γενική περίπτωση: στιγμιαίες θέσεις, ταχύτητες, επιταχύνσεις).
- ❑ Η εντολή που θα χρησιμοποιήσουμε είναι η Do που επαναλαμβάνει μια σειρά εντολών σύμφωνα με την τιμή μιας παραμέτρου (t στη συγκεκριμένη περίπτωση

```
Clear[x, Vx, y, Vy];
```

```
ti=0; tf=2; Δt=1/10; ax=0; ay=-9.8;
```

```
x[ti]=0; y[ti]=0; Vx[ti]=10; Vy[ti]=0;
```

```
Do[Vx[t+Δt]=Vx[t]+ax*Δt;
```

```
  x[t+Δt] = x[t] + Vx[t+Δt]*Δt;
```

```
  Vy[t+Δt] = Vy[t] + ay*Δt;
```

```
  y[t+Δt] = y[t] + Vy[t+Δt]*Δt, {t,ti,tf,Δt}]
```

Καθαρισμός τιμών μεταβλητών

Ορισμός τιμών παραμέτρων

Χρόνος: Αρχικός $t_i=0 \text{ s}$, Τελικός $t_f=2 \text{ s}$, Βήμα $\Delta t=1/10 \text{ s}$,Αρχική Θέση: $x[t_i]=0 \text{ m}$, $y[t_i]=0 \text{ m}$, Αρχική ταχύτητα: $Vx[t_i]=10 \text{ m/s}$, $Vy[t_i]=0$ Επιτάχυνση: $ax=0 \text{ m/s}^2$, $ay=g=-9.8 \text{ m/s}^2$

Το πρόγραμμα εκτελείται για χρονικό διάστημα 2 s , κάθε φορά που εκτελείται η εντολή Do το πρόγραμμα υπολογίζει τις νέες (στιγμιαίες) συντεταγμένες του κινητού (x, y) και τις αντίστοιχες στιγμιαίες ταχύτητες u_x, u_y

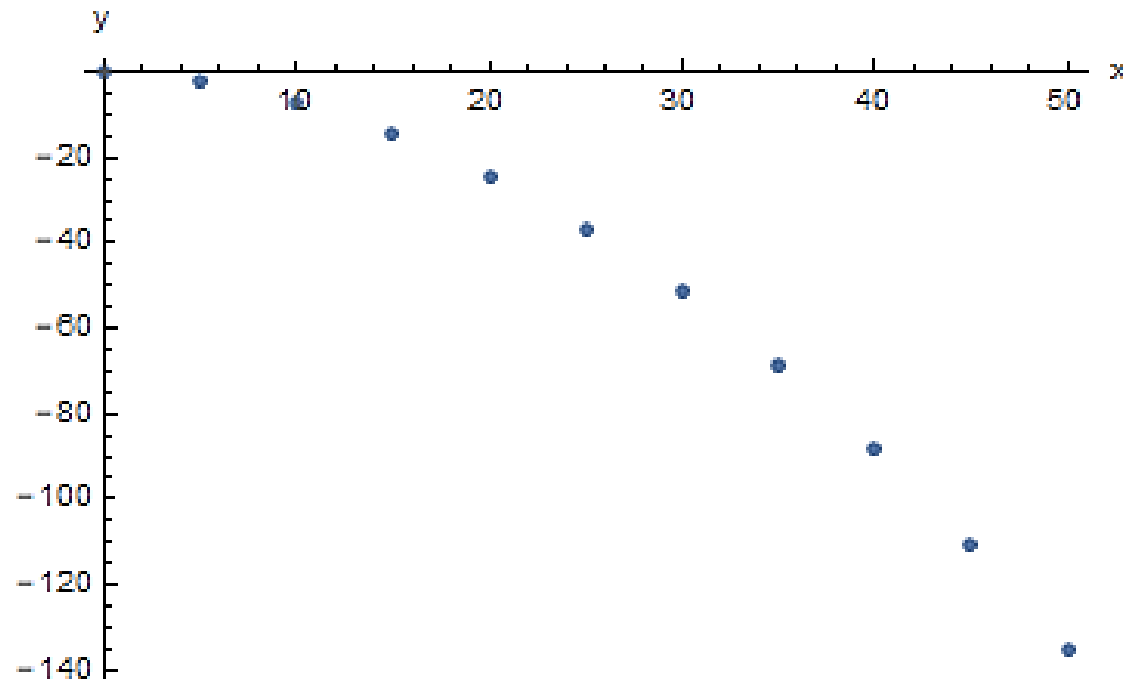


1 | Πρόβλημα 10.1: Βολές-Οριζόντια Βολή-Γραφική Παράσταση

Τα δεδομένα $x[t]$ και $y[t]$ για να γίνουν γραφική παράσταση πρώτα χρησιμοποιείται η εντολή **Table** (function) για να δημιουργηθεί μια λίστα δεδομένων και στη συνέχεια η εντολή **ListPlot** για να εμφανιστεί το γράφημα. που είναι πρακτικά η εξίσωση της τροχιάς.

```
data=Table[{x[t], y[t]}, {t, ti, tf, Δt}];
```

```
ListPlot[data, AxesLabel→{"x","y"}, PlotStyle->PointSize[0.015]]
```





1 | **Πρόβλημα 10.1: Βολές-Πλάγια Βολή** Το πρόβλημα της πλάγιας βολής έχεις ως δεδομένα εισόδου την αρχική ταχύτητα u_0 και τη αρχική γωνία της βολής θ . Στη συνέχεια ακολουθείται η ίδια διαδικασία υπολογισμού της οριζόντιας βολής με την εντολή Do που υπολογίζει τις στιγμιαίες τιμές των $x(t)$, $y(t)$ και $u_x(t)$, $u_y(t)$

```
Clear[x, Vx, y, Vy];
```

```
ti = 0; tf = 0.5; Δt = 1/70;
```

```
V = 18.0; θ = 10;
```

```
x[ti] = 0.; Vx[ti] = V * Cos[θ * π/180];
```

```
y[ti] = 0.; Vy[ti] = V * Sin[θ * π/180]; g = -9.8;
```

Επειδή η Mathematica υπολογίζει τις συναρτήσεις ArcSin, ArcCos, ArcTan σε ακτίνια πρέπει να πολλαπλασιάσετε με $(180/\pi)$ για να βγει το αποτέλεσμα σε μοίρες.

$$\tan \theta = \frac{V_y}{V_x} \quad \theta = \text{ArcTan}\left[\frac{V_y}{V_x}\right] \quad 56.3099$$

$$\theta = \text{N}[180. / \pi] * \text{ArcTan}[15. / 10.]$$

Για κάθε χρονική στιγμή t μπορεί να υπολογιστεί η τιμή της ταχύτητας u_t και η γωνία που σχηματίζει θ_t με τον οριζόντιο άξονα, εφόσον είναι γνωστές οι συνιστώσες της ταχύτητας τη συγκεκριμένη χρονική στιγμή Έστω ότι μια χρονική στιγμή οι συνιστώσες της ταχύτητας είναι

$$V_x = 10 \text{ m/sec και } V_y = 15 \text{ m/sec}$$

$$V^2 = V_x^2 + V_y^2 \quad 18.0278$$

$$V = \text{N}[\text{Sqrt}[10^2 + 15^2]]$$



1 | Πρόβλημα 10.1: Βολές-Πλάγια Βολή-Πρόγραμμα Υπολογισμού θέσης & ταχύτητας

Το πρόγραμμα αυτό υπολογίζει τις διαδοχικές θέσεις $(x[t], y[t])$ ενός κινητού που εκτελεί μια βολή με βάση τις αρχικές συνθήκες: $x_0=y_0=0$, $v=18 \text{ m/s}$, $\theta=10^\circ$, $a_y=g=-9.8 \text{ m/s}^2$ και σχεδιάζει την τροχιά της (y - x).

```
Clear[x, Vx, y, Vy];
```

```
ti = 0; tf = 0.5; Δt =  $\frac{1}{70}$ ;
```

```
V = 18.0; θ = 10;
```

```
x[ti] = 0.; Vx[ti] = V * Cos[θ *  $\frac{\pi}{180}$ ];
```

```
y[ti] = 0.; Vy[ti] = V * Sin[θ *  $\frac{\pi}{180}$ ]; g = -9.8;
```

```
Do[Vx[t + Δt] = Vx[t];
```

```
  x[t + Δt] = x[t] + Vx[t + Δt] * Δt;
```

```
  Vy[t + Δt] = Vy[t] + g * Δt;
```

```
  y[t + Δt] = y[t] + Vy[t + Δt] * Δt, {t, ti, tf, Δt}]
```

```
data=Table[{x[t], y[t]}, {t, ti, tf, Δt}];
```

```
ListPlot[data, AxesLabel→{"x", "y"}, PlotStyle→PointSize[0.015]]
```



1 | **Πρόβλημα 10.1: Βολές-Πλάγια Βολή** Μπορείτε εύκολα να δημιουργήσετε μια **ρεαλιστική απεικόνιση** της βολής, αν είχατε διαδοχικά φωτογραφικά στιγμιότυπα της βολής.

Η αρχική παραμετροποίηση των γραφικών απεικονίσεων περιλαμβάνει δύο γενικές εντολές

Dynamic Updating Enabled –Ενεργοποίηση δυναμικής ενημέρωσης – απαραίτητη για κυκλικούς υπολογισμούς-*Γίνεται μέσα από το μενού Evaluation*

Σε κάθε περίπτωση θα πρέπει στην εντολή

SetOptions[Graphics, AspectRatio->1, Axes-> Automatic, AxesOrigin -> { x_o , y_o }, AxesLabel->{"x (m)", "y(m)"}, GridLines->Automatic, PlotRange ->{{ $x_{\min}$, $x_{\max}$ },{ $y_{\min}$, $y_{\max}$ }}];

να ρυθμίσετε σωστά τις παραμέτρους

AxesOrigin-> { x_o , y_o } που θέλετε να βρίσκεται η αρχή των αξόνων και

PlotRange-> {{ $x_{\min}$, $x_{\max}$ }, { $y_{\min}$, $y_{\max}$ }} και ποιο είναι το εύρος τιμών του κάθε άξονα

Η παράμετρος DisplayFunction → Print χρειάζεται στις εκδόσεις της Mathematica μετά την 6.0

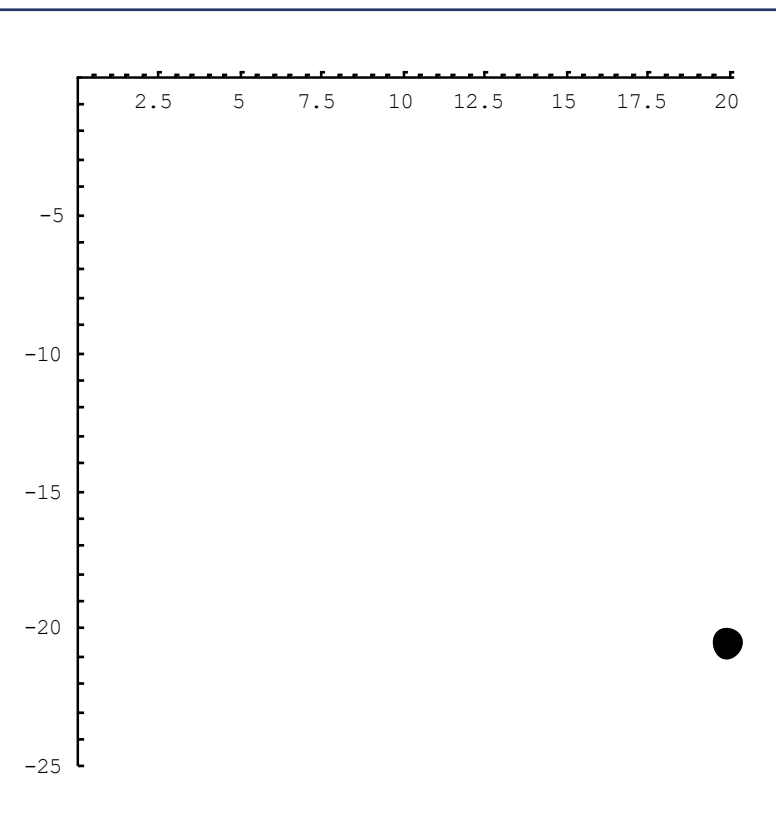


1 | **Πρόβλημα 10.1: Βολές-Πλάγια Βολή** Μπορείτε εύκολα να δημιουργήσετε μια **ρεαλιστική απεικόνιση** της βολής, με **πολλά** διαδοχικά φωτογραφικά στιγμιότυπα της βολής.

Η αρχική παραμετροποίηση των γραφικών απεικονίσεων περιλαμβάνει δύο γενικές εντολές

Dynamic Updating Enabled –Ενεργοποίηση δυναμικής ενημέρωσης – απαραίτητη για κυκλικούς υπολογισμούς-

Γίνεται μέσα από το μενού Evaluation



SetOptions[Graphics, AspectRatio->1, Axes-> Automatic, AxesOrigin -> {x[ti], y[tf]*1.2}, AxesLabel->{"x (m)", "y(m)"}, GridLines->Automatic, PlotRange ->{{0,1.2*x[tf]},{y[ti],1.2*y[tf]}}];

Πολλά διαδοχικά φωτογραφικά στιγμιότυπα:

Do[Show[Graphics[{PointSize[0.04], Point[{x[t], y[t]}]}], DisplayFunction -> Print, PlotLabel -> t], {t, ti, tf, Δt}]

Η παράμετρος DisplayFunction → Print χρειάζεται στις εκδόσεις της Mathematica μετά την 6.0



1 | Πρόβλημα 10.1: Βολές-Ρεαλιστικές Απεικονίσεις Γραφικών

Μπορείτε εύκολα να δημιουργήσετε μια ρεαλιστική απεικόνιση **με ένα κινούμενο στιγμιότυπο** της βολής

Η αρχική παραμετροποίηση των γραφικών απεικονίσεων περιλαμβάνει δύο γενικές εντολές

Dynamic Updating Enabled –Ενεργοποίηση δυναμικής ενημέρωσης – απαραίτητη για κυκλικούς υπολογισμούς-

Γίνεται μέσα από το μενού Evaluation

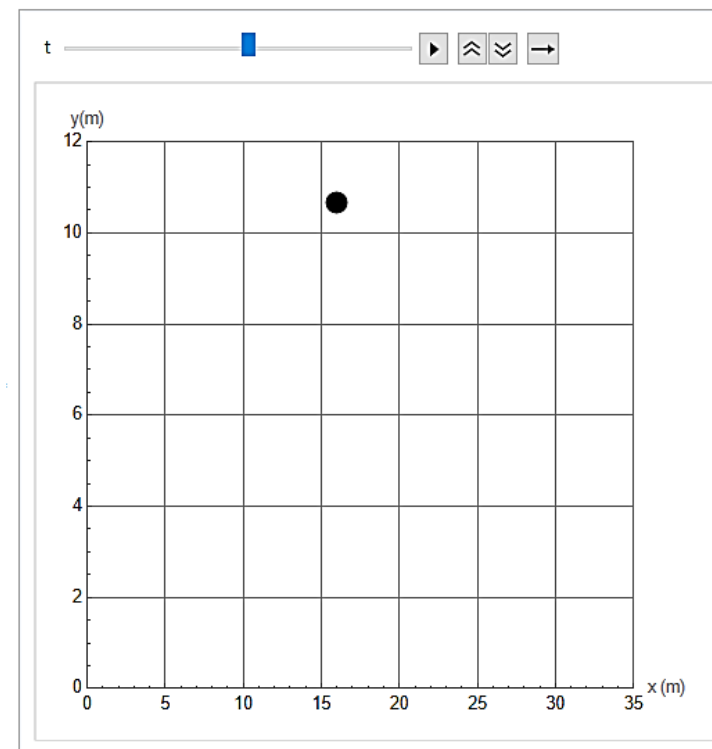
SetOptions[Graphics, AspectRatio->1, Axes-> Automatic, AxesOrigin -> {x[ti], y[tf]*1.2}, AxesLabel->{"x (m)", "y(m)"}, GridLines->Automatic, PlotRange ->{{0,1.2*x[tf]},{y[ti],1.2*y[tf]}}];

Η απεικόνιση με **1 κινούμενο στιγμιότυπο** γίνεται ως εξής:

Animate[Show[Graphics[{PointSize[0.04],Point[{x[t],y[t]}]}]], {t,ti,tf,Δt},

AnimationRunning->False]

Η Animate υποστηρίζεται από τις εκδόσεις της Mathematica μετά την 7





1 | Πρόβλημα 10.1: Βολές



Να δημιουργήσετε για την παρακάτω κίνηση (κατακόρυφη βολή), το πρόγραμμα που υπολογίζει τις διαδοχικές θέσεις $(x[t], y[t])$ του κινητού, το διάγραμμα x - y καθώς και την κινούμενη απεικόνιση:

$$t_i = 0; t_f = 5; \Delta t = \frac{5}{10};$$

$$x[t_i] = 0.; V_x[t_i] = 0.; y[t_i] = 140; V_y[t_i] = 0.; g = -9.8;$$



1 | Πρόβλημα 10.1: Βολές



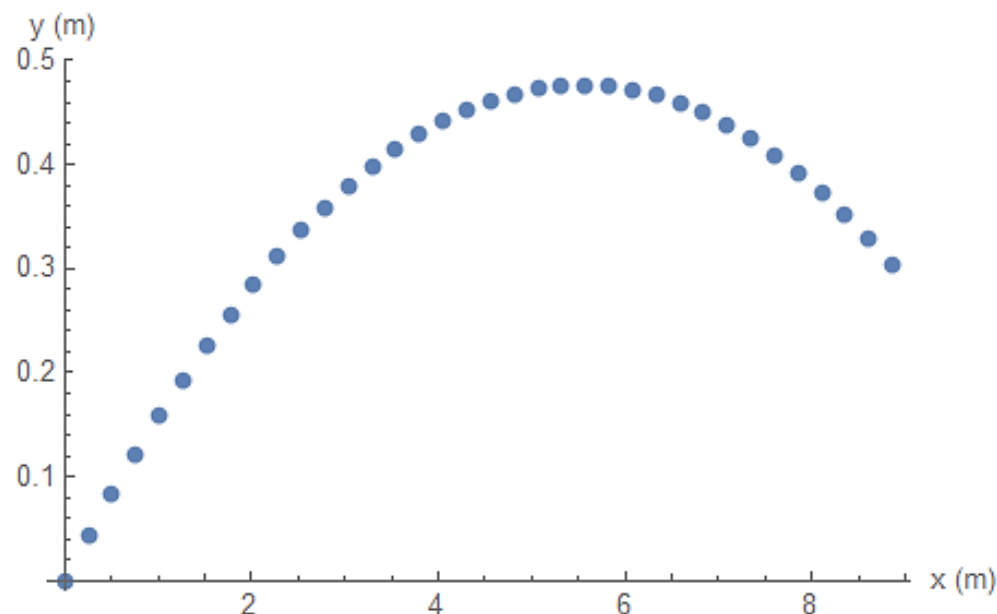
1. Να δημιουργήσετε πρόγραμμα για το διάγραμμα x-y καθώς και την κινούμενη απεικόνιση για την παρακάτω κίνηση με τις εξής παραμέτρους:

Πλάγια βολή $t_i = 0$; $t_f = 0.5$; $\Delta t = \frac{1}{70}$;

$V = 18.0$; $\theta = 10$; $g = -9.8$;

$x[t_i] = 0.$;

$y[t_i] = 0.$;



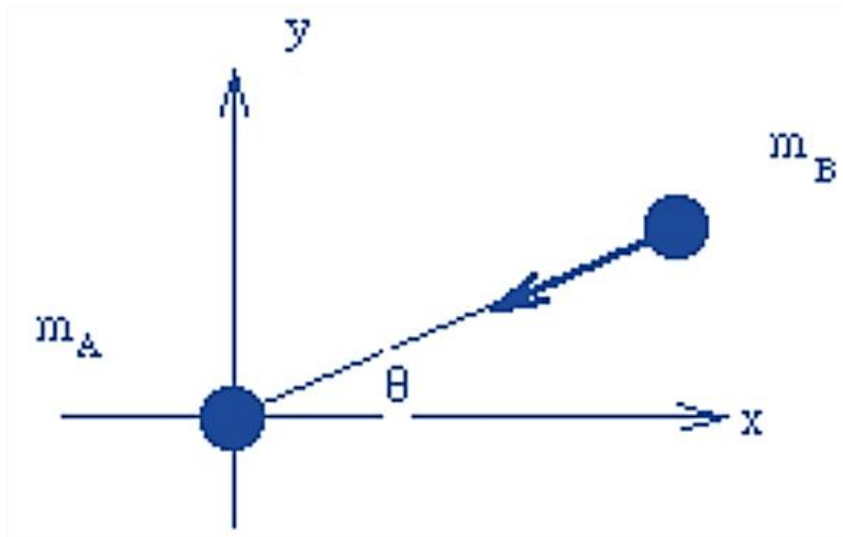
2. Το βεληνεκές μιας πλάγιας βολής $x_{\max}$ είναι η οριζόντια απόσταση που διανύει το κινητό μέχρι να ακουμπήσει το έδαφος ($y=0$, Προσοχή το $y=0$ αντιστοιχεί και στην αρχική θέση της βολής όπου $x=0$). Να υπολογίσετε το χρόνο που κάνει το σώμα να ξαναφτάσει στο έδαφος και πόσο είναι το βεληνεκές του.

3. Στο διάγραμμα τροχιάς το σώμα απεικονίζεται να μην φθάνει στο έδαφος. Ρυθμίζοντας τις αρχικές παραμέτρους χρόνου μπορείτε να δείξετε την πλήρη βολή μέχρι το σώμα να φθάσει στο έδαφος;



1 | Πρόβλημα 10.2: Κινήσεις Πλανητών

Το πρόβλημα



□ Η μελέτη της κίνησης ενός πλανήτη αναφέρεται στην απλή της μορφή σε σφαιρικά αντικείμενα που εκτελούν μια ομαλή κυκλική κίνηση γύρω από ένα βαρυτικό κέντρο (τον ήλιο).

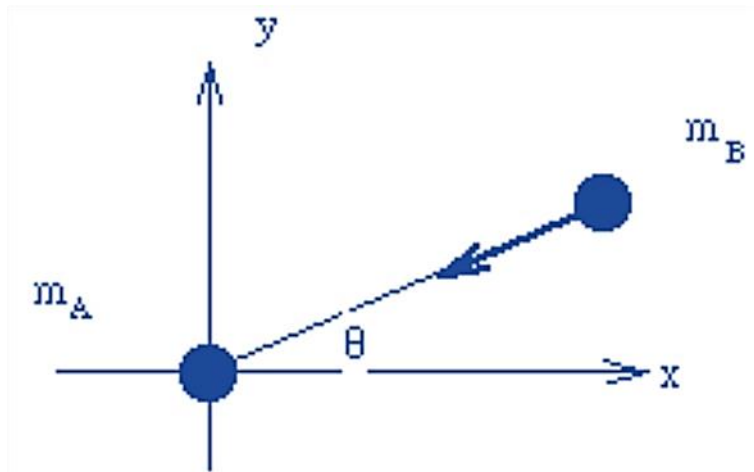
□ Τα προβλήματα κίνησης πλανητών έχουν να κάνουν συνήθως με την απεικόνιση της τροχιάς για συγκεκριμένο χρονικό διάστημα και τον υπολογισμό φυσικών μεγεθών για διάφορες χρονικές στιγμές.



1 | Πρόβλημα 10.2: Κινήσεις Πλανητών

Η φυσική του προβλήματος

- Ο νόμος της βαρύτητας ανακαλύφθηκε από τον Νεύτωνα και εφαρμόζεται σε όλα τα αντικείμενα που έχουν μάζα.
- Η δύναμη αυτή είναι ελκτική και βρίσκεται πάνω στην ευθεία που ενώνει τα κέντρα των δύο μαζών.



$$F = G \frac{m_A m_B}{r^2} \quad G = 6.67 \cdot 10^{-11} \text{ N m}^2/\text{kg}^2$$

- Η περίπτωση $m_A \gg m_B$ είναι πιο απλή καθώς η κίνηση της μάζας m_B έχει μικρή επίδραση στην κίνηση της μάζας m_A .
- Σε αυτή την περίπτωση, είναι πιο εύκολο να χρησιμοποιήσουμε ένα σύστημα συντεταγμένων όπως η μάζα m_A βρίσκεται στο κέντρο του συστήματος.
- Η m_B είναι η μόνη μάζα που κινείται και βρίσκεται στη θέση (x, y) όπως φαίνεται στο σχήμα.



1 | Πρόβλημα 10.2: Κινήσεις Πλανητών

Συνθήκες-Σταθερές-Παράμετροι

Θεωρείστε τις παρακάτω τιμές για τα φυσικά μεγέθη:

Η σταθερά της παγκόσμιας έλξης: $G = 6.67 \times 10^{-11} \text{ N m}^2/\text{kg}^2$, η μάζα του ήλιου $m_A = 2.0 \times 10^{30} \text{ kg}$, η μάζα της γης $m_B = 6.0 \times 10^{24} \text{ kg}$.

Ο χρόνος για μια πλήρη περιφορά της γης γύρω από τον ήλιο t_f : $1 \text{ έτος} = 365 \text{ μέρες} = 3.1 \times 10^7 \text{ sec}$ και το χρονικό βήμα, $t = t_f/600 \text{ s} = 51667 \text{ s}$.

Επίσης η ταχύτητα της γης χρειάζεται σαν αρχική συνθήκη. Θεωρώντας την τροχιά της γης ως

κυκλική με ακτίνα $r = 1.5 \times 10^{11} \text{ m}$ μπορεί να υπολογιστεί από τη σχέση: $v_y = \frac{2\pi r}{t_f}$



1 | Πρόβλημα 10.2: Κινήσεις Πλανητών

Υπολογισμός τροχιάς πλανήτη

Στο συγκεκριμένο πρόβλημα έχουμε τις εξής σχέσεις:

$$F_x = -G \frac{m_A m_B}{r^2} \cos \theta$$

$$F_y = -G \frac{m_A m_B}{r^2} \sin \theta$$

$$\cos \theta = \frac{x}{r}$$

$$r = \sqrt{x^2 + y^2}$$

$$\sin \theta = \frac{y}{r}$$

Και τις εξής αρχικές συνθήκες

```
Clear[x, y, Vx, Vy];
```

365×24×3600 s

```
ti = 0.; tf = 3.1 * 10^7; Steps = 600.; Δt = tf / Steps;
```

```
x[0] = 1.5 * 10^11; Vx[0] = 0.; y[0] = 0.0; Vy[0] = 30400.;
```

```
G = 6.67 * 10^-11; Ma = 2.0 * 10^30;
```

$$v_y = \frac{2\pi r}{t_f}$$

Επομένως ο πλανήτης αρχικά βρίσκεται στη δεξιά πλευρά του άξονα x και κινείται προς τα πάνω.

Με τη βοήθεια της εντολής **Do** θα χρειαστεί να υπολογίσετε τις παρακάτω ποσότητες

$$a_x[t] = \frac{F_x[t]}{m_B} = -\frac{G m_A x[t]}{r^3}$$

$$a_y[t] = \frac{F_y[t]}{m_B} = -\frac{G m_A y[t]}{r^3}$$

$$V_x[t + \Delta t] = V_x[t] + a_x[t] \Delta t$$

$$V_y[t + \Delta t] = V_y[t] + a_y[t] \Delta t$$

$$x[t + \Delta t] = x[t] + V_x[t] \Delta t$$

$$y[t + \Delta t] = y[t] + V_y[t] \Delta t$$

$$a = \sqrt{a_x^2 + a_y^2}$$

$$v = \sqrt{v_x^2 + v_y^2}$$

$$r = \sqrt{x^2 + y^2}$$



1 | Πρόβλημα 10.2: Κινήσεις Πλανητών

Υπολογισμός τροχιάς πλανήτη

1. Αρχικές Παράμετροι

```
Clear[x, y, Vx, Vy];  
ti = 0.; tf = 3.2 * 107; Steps = 600.; Δt = tf / Steps;  
x[0] = 1.5 * 1011; Vx[0] = 0.; y[0] = 0.0; Vy[0] = 30400.;  
G = 6.67 * 10-11; Ma = 2.0 * 1030;
```

```
Do[ r[n] = Sqrt[x[n]^2 + y[n]^2];
```

```
  V[n] = Sqrt[Vx[n]^2 + Vy[n]^2];
```

```
  ax[n] = -G * Ma * x[n] / r[n]^3;
```

```
  ay[n] = -G * Ma * y[n] / r[n]^3;
```

```
  a[n] = Sqrt[ax[n]^2 + ay[n]^2];
```

```
  Vx[n + 1] = Vx[n] + ax[n] * Δt;
```

```
  Vy[n + 1] = Vy[n] + ay[n] * Δt;
```

```
  x[n + 1] = x[n] + Vx[n + 1] * Δt;
```

```
  y[n + 1] = y[n] + Vy[n + 1] * Δt, {n, 0, Steps}]
```

2. Ρουτίνα Υπολογισμού

Αν θέλω η Do να επαναλάβει περισσότερες από μία εντολές

Το n είναι η παράμετρος που παίρνει τιμές από 0 μέχρι 600 (θα υπολογίσω τις διαφορές παραμέτρους 600 φορές). Ο χρόνος ανεβαίνει κάθε φορά κατά $\Delta t = 1/600$ της περιόδου περιστροφής της γης.

3. Σχεδιασμός Τροχιάς

```
datax = Table[{x[n], y[n]}, {n, 0, Steps}];  
ListPlot[data, AxesLabel -> {"x", "y"}, AspectRatio -> 1.0]
```




1

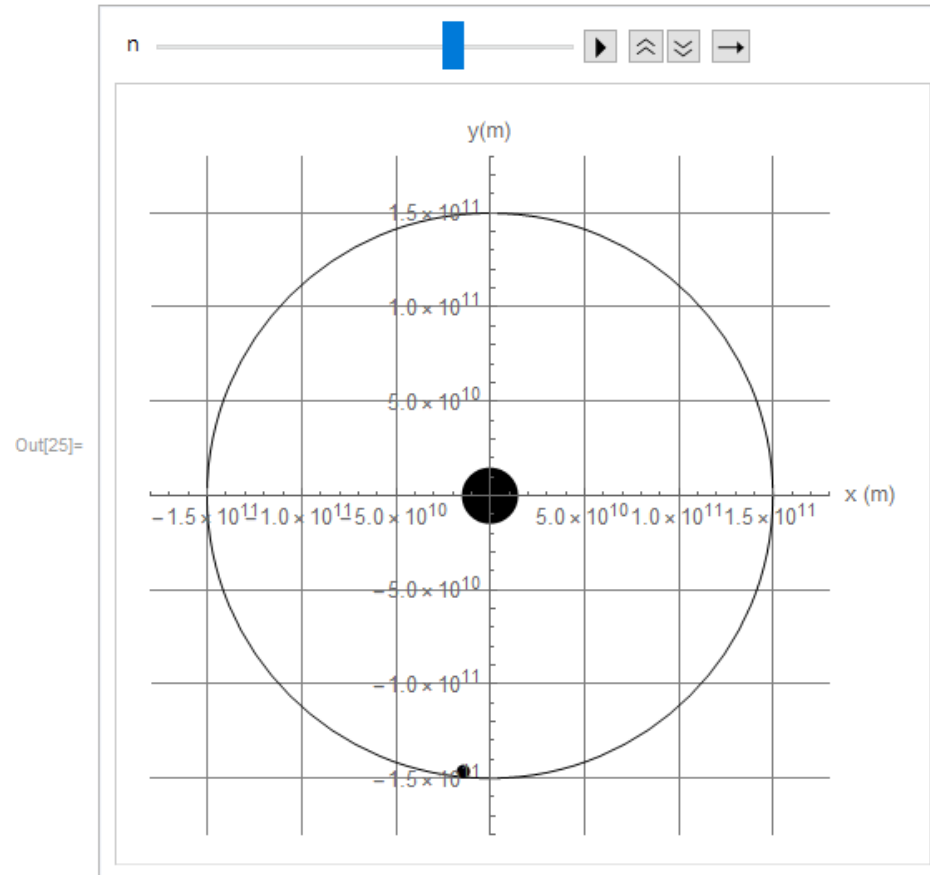
Πρόβλημα 10.2: Κινήσεις Πλανητών

Απεικόνιση τροχιάς πλανήτη

`SetOptions[Graphics, AspectRatio  $\rightarrow$  1, Axes  $\rightarrow$  Automatic, AxesOrigin  $\rightarrow$  {0,0}, AxesLabel`

`$\rightarrow$  {"x (m)", "y(m)"}, GridLines  $\rightarrow$  Automatic, PlotRange  $\rightarrow$  {{-1.2ro, 1.2ro}, {-1.2ro, 1.2ro}}];`

`Animate[Show[Graphics[{Circle[{0,0}, ro], Disk[{0,0}, ro/10], PointSize[0.02], Point[{x[n], y[n]}]}]], {n, 0, Steps
- 30, 2}, AnimationRunning $\rightarrow$ False]`





1 | Πρόβλημα 10.2: Κινήσεις Πλανητών

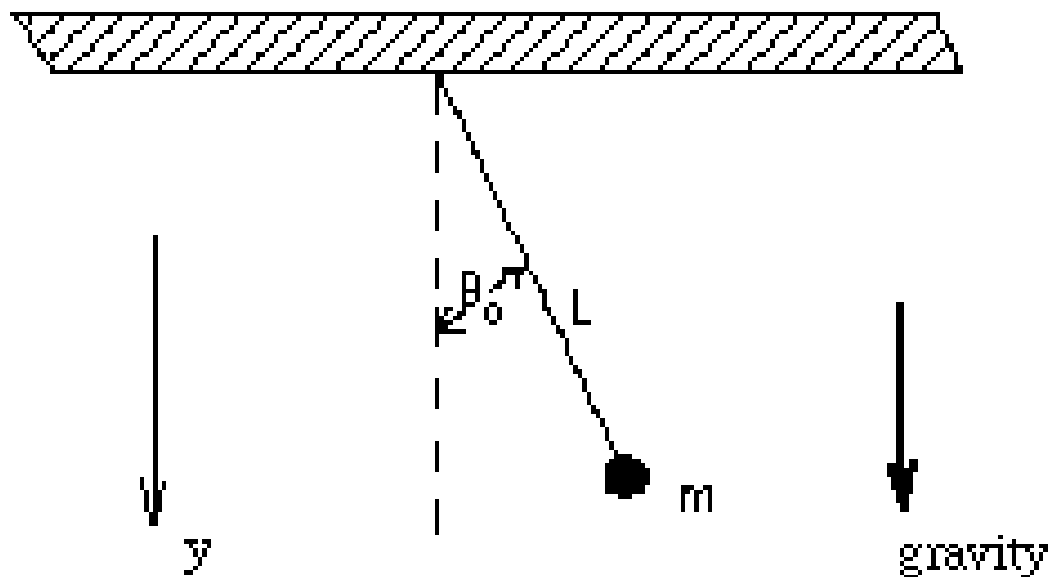


1. Να δημιουργήσετε το πρόγραμμα υπολογισμού της τροχιάς της Γης γύρω από τον ήλιο.
2. Να υπολογίσετε με τη βοήθεια της εντολής `Do` τις παραστάσεις: $\mathbf{u}$ και $\mathbf{a}$ (όπως υπολογίσατε την r). Να κάνετε τις γραφικές παραστάσεις $\mathbf{u}-r$, $\mathbf{a}-r$.
3. Να υπολογίσετε από τις παραπάνω γραφικές παραστάσεις τις μέγιστες τιμές ταχύτητας και επιτάχυνσης καθώς και σε ποιες θέσεις αυτές εμφανίζονται.
4. Ποιες εντολές θα προσθέσετε στο πρόγραμμα αυτό για να δείχνει με κινούμενη απεικόνιση την τροχιά;
5. Τι θα πρέπει να αλλάξετε στο πρόγραμμα για να λειτουργεί και για ένα διαφορετικό πλανήτη.



1 | Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές

Το πρόβλημα



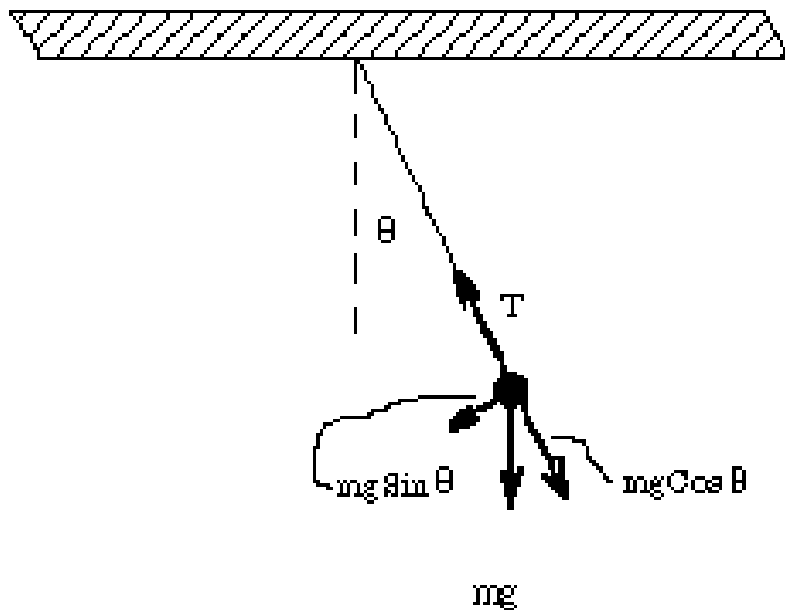
Το απλό μαθηματικό εκκρεμές αποτελείται από μια μάζα που κρέμεται από ένα αβαρές σχοινί και κινείται περιοδικά. Η δύναμη του βάρους της μάζας είναι η αιτία της κίνησης αυτής.

Συνήθως ζητείται η καταγραφή των θέσεων του εκκρεμούς και η περίοδος του ανάλογα με κάποιες παραμέτρους (γωνία εκτροπής, μήκος εκκρεμούς, πεδίο βαρύτητας)



1 | Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές

Η φυσική του προβλήματος



Εκτός της δύναμης του βάρους στη μάζα ασκείται και μία ακόμα δύναμη, η τάση του νήματος T , η οποία όμως εξουδετερώνεται από την συνιστώσα του βάρους στην κατεύθυνση του νήματος.

Η επιτροχία συνιστώσα του βάρους **$F = -m g \sin \theta$**

είναι τελικά η υπεύθυνη για την παλινδρομική κίνηση του εκκρεμούς. Η συνιστώσα αυτή είναι η δύναμη επαναφοράς και κατευθύνεται αντίθετα προς τις αυξήσεις της γωνίας θ .

γραμμική ταχύτητα: **$v = L\omega$** , γωνιακή ταχύτητα: **$\omega = d\theta/dt$**

2ος νόμος του Νεύτωνα: **$F = ma \Rightarrow F = m dv/dt \Rightarrow F = mL d\omega/dt \Rightarrow d\omega/dt = F/mL$**

Αναδρομικές σχέσεις για τον υπολογισμό της

γωνιακής ταχύτητας ω και της γωνίας θ

$$\omega(t+\Delta t) = \omega(t) + d\omega/dt \Delta t = \omega(t) + F/mL \Delta t$$

$$\theta(t+\Delta t) = \theta(t) + \omega(t+\Delta t) \Delta t$$



1 | Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές

Συνθήκες-Σταθερές-Παράμετροι

```
g=9.8; L=2.0; q[0]=0.05; w[0]=0.0;  
  
Δt=0.005; TotalTime=2.9; Steps=Round[TotalTime/Δt];  
  
Do[w[n+1]=w[n]-(g/L)*Sin[q[n]]*Δt;  
  
q[n+1]=q[n]+w[n+1]*Δt,{n,0, Steps}]
```

Η μάζα του εκκρεμούς δεν εμφανίζεται στους υπολογισμούς καθώς απαλείφεται μετά τις αντικαταστάσεις. Αυτό σημαίνει ότι ένα εκκρεμές μάζας 300 g θα έχει την ίδια περίοδο με ένα εκκρεμές μάζας 6 kg.

- Η αρχική γωνιακή ταχύτητα είναι **w[0]=0** καθώς το εκκρεμές αφήνεται χωρίς αρχική γωνιακή ταχύτητα.
- Η γωνία στο πρόγραμμα συμβολίζεται με το γράμμα **q** και η αρχική της τιμή είναι **q[0]=0.05** ακτίνια (~3°).
- Το χρονικό βήμα **Δt=0.005** είναι ένας αρκετά μικρός αριθμός που όσο μικρότερος είναι τόσο περισσότερα βήματα υπολογισμού απαιτούνται αλλά η περίοδος υπολογίζεται με μεγαλύτερη ακρίβεια.



3 | Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές

Καταγραφή τροχιάς εκκρεμούς

```
g=9.8; L=2.0; q[0]=0.05; w[0]=0.0;
```

```
Δt=0.005; TotalTime=2.9; Steps=Round[TotalTime/Δt];
```

```
Do[w[n+1]=w[n]-(g/L)*Sin[q[n]]*Δt;
```

```
q[n+1]=q[n]+w[n+1]*Δt,{n,0, Steps}]
```

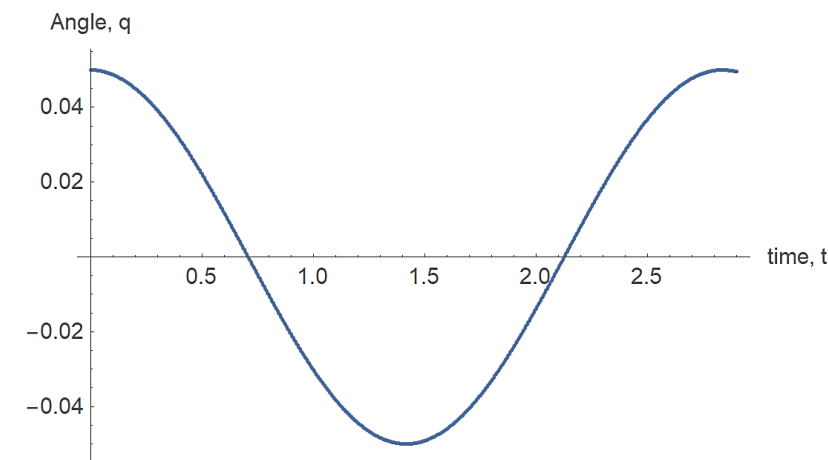
```
qData = Table[{n*Δt,q[n]}, {n,0,TotalTime, Steps}];
```

```
qplot = ListPlot[qData, AxesLabel->{"time, t","Angle, q"}]
```

1. Αρχικές Παράμετροι

2. Ρουτίνα Υπολογισμού
Αν θέλω η Do να επαναλάβει
περισσότερες από μία εντολές

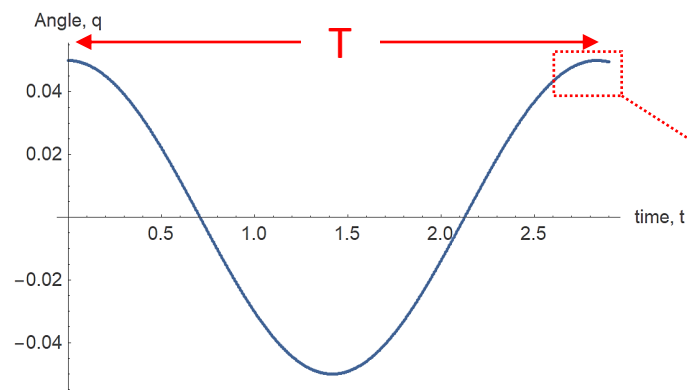
3. Σχεδιασμός Τροχιάς





1 | Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές

Υπολογισμός περιόδου εκκρεμούς

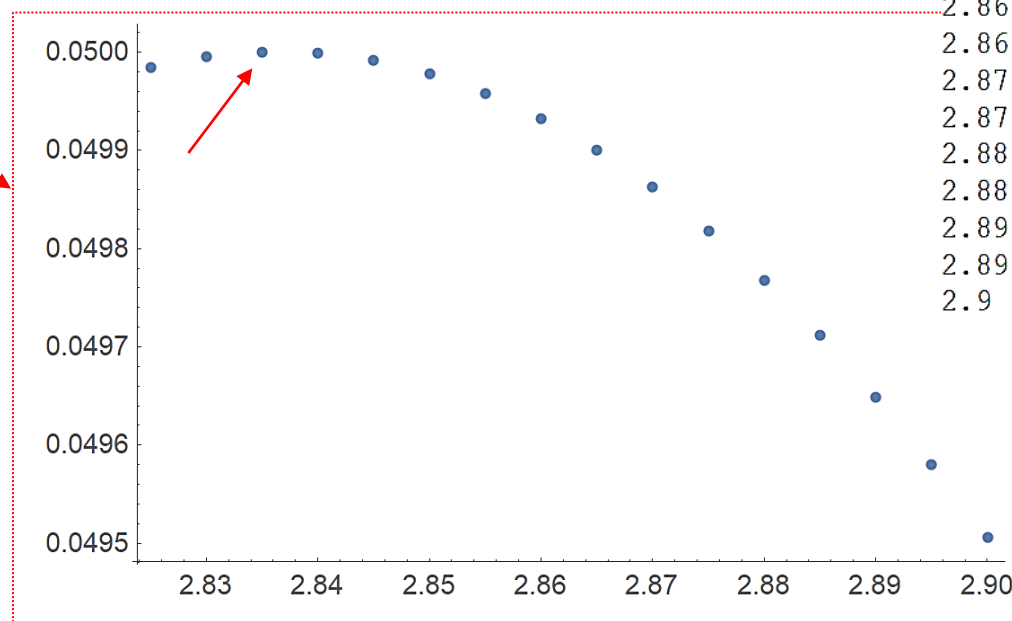


Ας εντοπίσουμε το 1^ο μέγιστο στη γωνία μετά το $t=0$

```
qdata2=Table[{n*Δt,q[n]},{n,Steps-15,Steps}]
```

```
qplot2 = ListPlot[qData2, AxesLabel->{"time, t","Angle, q"}]
```

2.8	0.0498388
2.805	0.0498802
2.81	0.0499156
2.815	0.0499448
2.82	0.0499679
2.825	0.0499849
2.83	0.0499958
2.835	0.0500005
2.84	0.0499992
2.845	0.0499917
2.85	0.0499781
2.855	0.0499583
2.86	0.0499325
2.865	0.0499005
2.87	0.0498624
2.875	0.0498183
2.88	0.049768
2.885	0.0497116
2.89	0.0496492
2.895	0.0495806
2.9	0.049506



Υπάρχει ένα μέγιστο (0.0500005) τη χρονική στιγμή $t = 2.835$.

Άρα η περίοδος του εκκρεμούς είναι 2.835 s.

Καθώς το βήμα των υπολογισμών ορίζεται από το $\Delta t = 0.005$

η τελική απάντηση είναι :

$$T = 2.835 \pm 0.005 \text{ s}$$

Αν θέλουμε μεγαλύτερη ακρίβεια μπορούμε να επιλέξουμε μικρότερο Δt .

Αντίστοιχος θεωρητικός υπολογισμός: $T = 2\pi \sqrt{\frac{L}{g}}$



1

Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές



Διερεύνηση των παραμέτρων του απλού μαθηματικού εκκρεμούς

L (m)	T _{πειρ} (s)	T _{θεωρ} (s)	π (%)
1.0			
2.0	2.835	2.83845	-0.12
3.0			
4.0			
5.0			

1. Μεταβολή του L: Με οδηγό τον πίνακα υπολογίστε την περίοδο του εκκρεμούς (βρίσκοντας ποια χρονική στιγμή η μάζα περνάει από μέγιστο για 2^η φορά). Η παράμετρος **TotalTime** πρέπει να ρυθμιστεί κατάλληλα ώστε το πρόγραμμα να καταγράφει τουλάχιστον 1 πλήρη περίοδο. Διατηρείστε την επιτάχυνση της βαρύτητας $g=9.8 \text{ m/s}^2$ και το αρχικό πλάτος $q[0]=0.05$ ακτίνια.

I. Σχεδιάστε το T^2 σαν συνάρτηση του L. Τι είδους καμπύλη παίρνετε; Πόση είναι η κλίση της;

II. Υπολογίστε τις αντίστοιχες θεωρητικές τιμές από τη σχέση : $T = 2\pi\sqrt{\frac{L}{g}}$ και στη συνέχεια τα αντίστοιχα

σφάλματα $\pi = (T_{\text{πειρ}} - T_{\text{θεωρ}}) / T_{\text{θεωρ}} \%$



1

Πρόβλημα 10.3: Απλό Μαθηματικό Εκκρεμές

Διερεύνηση των παραμέτρων του απλού
μαθηματικού εκκρεμούς

2. Μεταβολή του g : Υπολογίστε την περίοδο του εκκρεμούς σύμφωνα με τους 2 πίνακες σε διαφορετικά ύψη από την επιφάνεια της γης και σε διαφορετικούς πλανήτες. Σε κάθε περίπτωση διατηρήστε το πλάτος 0.1 ακτίνια και το μήκος του εκκρεμούς 2.0 m.

Σε ποιον πλανήτη το εκκρεμές ταλαντώνεται πιο αργά και σε ποιον πιο γρήγορα από τους υπόλοιπους;

Πως μεταβάλλεται η περίοδος με την απόσταση από την επιφάνεια της γης (διάγραμμα-ListPlot);

Πλανήτης	g (m/s ²)
Ερμής	3.61
Αφροδίτη	8.83
Άρης	3.75
Δίας	26.0
Κρόνος	11.2
Ουρανός	10.5
Ποσειδώνας	13.3
Πλούτων	0.61

Απόσταση από		g (m/s ²)
την επιφάνεια της γης ($\times 10^3$ km)	το κέντρο της γης ($\times 10^6$ m)	
0	6.4	9.8
1	7.4	7.33
2	8.4	5.68
3	9.4	4.53
4	10.4	3.70
5	11.4	3.08
6	12.4	2.60
7	13.4	2.23
8	14.4	1.93
9	15.4	1.69
10	16.4	1.49
50	56.4	0.13



1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Το πρόβλημα



Κρούση ονομάζεται το φαινόμενο όπου ένα σώμα συγκρούεται με ένα άλλο. Η περίπτωση της κρούσης όπου η κινητική ενέργεια διατηρείται καλείται ελαστική κρούση. Π.χ. οι μπάλες του μπιλιάρδου κάνουν σχεδόν ελαστικές κρούσεις.

Συνήθως στα προβλήματα κρούσεις ζητούνται οι ταχύτητες των σωμάτων πριν ή μετά την κρούση ανάλογα με τα δεδομένα και οι κατευθύνσεις στις οποίες θα κινηθούν τα σώματα μετά την κρούση.



1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Η φυσική του προβλήματος



* **Υπόθεση:** οι σφαίρες απλώς ολισθαίνουν και δεν περιστρέφονται αλλιώς θα έπρεπε να συμπεριληφθεί και ενεργειακός όρος λόγω περιστροφής.

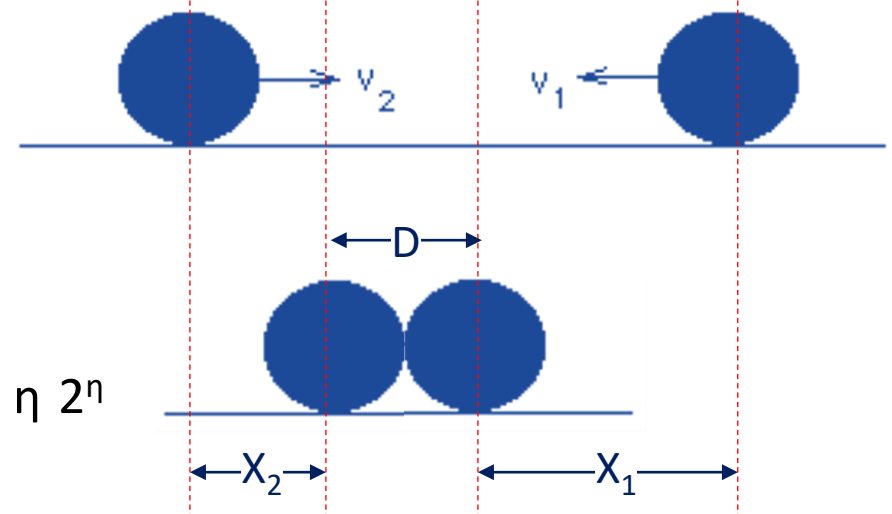
- Για απλότητα θεωρούμε ότι τα δύο αντικείμενα είναι σφαίρες και κινούνται σε μία διάσταση όπως φαίνεται στο σχήμα.
- Αρχικά τα δύο σώματα βρίσκονται στις θέσεις $\mathbf{x}_1$, $\mathbf{x}_2$ ενώ
- μετά από χρόνο Δt στις θέσεις $\mathbf{X}_1 = \mathbf{x}_1 + \mathbf{u}_1 \Delta t$, $\mathbf{X}_2 = \mathbf{x}_2 + \mathbf{u}_2 \Delta t$
- Η σφαίρα 1 κινείται προς τα αριστερά και έχει αρνητική ταχύτητα ενώ η 2^η σφαίρα κινείται προς τα δεξιά και έχει θετική ταχύτητα. Αρχή διατήρησης της
 - ορμής: $\mathbf{m}_1 \mathbf{u}_1 + \mathbf{m}_2 \mathbf{u}_2 = \mathbf{m}_1 \mathbf{u}'_1 + \mathbf{m}_2 \mathbf{u}'_2$
 - κινητικής ενέργειας: $\frac{1}{2} \mathbf{m}_1 \mathbf{u}_1^2 + \frac{1}{2} \mathbf{m}_2 \mathbf{u}_2^2 = \frac{1}{2} \mathbf{m}_1 \mathbf{u}_1'^2 + \frac{1}{2} \mathbf{m}_2 \mathbf{u}_2'^2$



1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Συνθήκες-Σταθερές-Παράμετροι

- Αρχικά τα δύο σώματα βρίσκονται στις θέσεις $\mathbf{x}_1$, $\mathbf{x}_2$ ενώ
- μετά από χρόνο Δt στις θέσεις $\mathbf{X}_1 = \mathbf{x}_1 + \mathbf{u}_1 \Delta t$, $\mathbf{X}_2 = \mathbf{x}_2 + \mathbf{u}_2 \Delta t$
- Η σφαίρα 1 κινείται προς τα αριστερά και έχει αρνητική ταχύτητα ενώ η 2^η σφαίρα κινείται προς τα δεξιά και έχει θετική ταχύτητα.
- Όταν η απόσταση μεταξύ των δύο σφαιρών είναι ίση με τη διάμετρο των σφαιρών τότε συμβαίνει το φαινόμενο της κρούσης $\text{Dia} = \mathbf{X}_2 - \mathbf{X}_1$, $t_{\text{collis}} = (\text{Dia} - (\mathbf{x}_2 - \mathbf{x}_1)) / (\mathbf{u}_2 - \mathbf{u}_1)$



Αρχικές συνθήκες:

η μάζα $\mathbf{m}_1$ ηρεμεί αρχικά, $\mathbf{u}_1 = 0$ και βρίσκεται στην αρχή του άξονα-x, $\mathbf{x1}[0] = 0 \text{ m}$ και

η μάζα $\mathbf{m}_2$ κινείται προς τη $\mathbf{m}_1$ με ταχύτητα $\mathbf{u}_2 = 6 \text{ m/s}$, αρχική θέση $\mathbf{x2}[0] = -5 \text{ m}$.

$$\mathbf{x1}[0] = 0.; \mathbf{v1} = 0.; \mathbf{x2}[0] = -5.; \mathbf{v2} = 6.;$$

$$\mathbf{m1} = 1.; \mathbf{m2} = 1.; \text{Dia} = 1.;$$

$$\mathbf{tcollis} = (\text{Dia} - (\mathbf{x1}[0] - \mathbf{x2}[0])) / (\mathbf{v1} - \mathbf{v2});$$

$$\text{NumFrames} = 10; \Delta t = \mathbf{tcollis} / \text{NumFrames};$$



1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Υπολογισμός θέσεων και ταχυτήτων

Το πρόγραμμα υπολογίζει τη θέση της κάθε μάζας σε **10** στιγμιότυπα πριν και **10** στιγμιότυπα μετά από την κρούση και τη μεταβολή της ταχύτητας κάθε μάζας κατά τη διάρκεια της κρούσης.

(* Κίνηση πριν την κρούση *)

```
Do[x1[n+1]=x1[n]+v1*Δt;
```

```
  x2[n+1]=x2[n]+v2*Δt,
```

```
  {n,0,NumFrames}]
```

(* Μεταβολή ταχύτητας λόγω της κρούσης *)

```
DeltaV2=2*(v2-v1)/(1+(m2/m1));
```

```
DeltaV1=-(m2/m1)*DeltaV2;
```

```
v1=v1-DeltaV1; v2=v2-DeltaV2;
```

(* Κίνηση μετά την κρούση *)

```
Do[x1[n+1]=x1[n]+v1*Δt;
```

```
  x2[n+1]=x2[n]+v2*Δt,
```

```
  {n,NumFrames,2*NumFrames}]
```

Το πρόγραμμα αυτό δεν εμφανίζει απευθείας έξοδο μετά την εκτέλεση του. Υπολογίζει τις θέσεις των δύο μαζών πριν και μετά την κρούση και οι θέσεις αυτές χρησιμοποιούνται στην αναπαράσταση της κρούσης. Για να εμφανιστούν οι ταχύτητες των δύο μαζών μετά την κρούση προσθέτω δύο ακόμη γραμμές στο τέλος του παραπάνω προγράμματος.

```
Print["Velocity of particle 1 (after collision) =", v1]
```

```
Print["Velocity of particle 2 (after collision) =", v2]
```




1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Πρόγραμμα υπολογισμού θέσεων και ταχυτήτων

```
x1[0]=0.;v1=0.;x2[0]=-5.;v2=6.;  
m1=1.; m2=1.; Dia=1.;  
tcollis=(Dia-(x1[0]-x2[0]))/(v1-v2);  
NumFrames=10;Δt=tcollis/NumFrames;
```

1. Αρχικές Παράμετροι

```
(* Κίνηση πριν την κρούση *)
```

```
Do[x1[n+1]=x1[n]+v1*Δt;  
  x2[n+1]=x2[n]+v2*Δt,  
  {n,0,NumFrames}]
```

```
(* Αλλαγή ταχύτητας λόγω της κρούσης*)
```

```
DeltaV2=2*(v2-v1)/(1+(m2/m1));  
DeltaV1=-(m2/m1)*DeltaV2;  
v1=v1-DeltaV1; v2=v2-DeltaV2;
```

2. Ρουτίνα Υπολογισμού

```
(* Κίνηση μετά την κρούση *)
```

```
Do[x1[n+1]=x1[n]+v1*Δt;  
  x2[n+1]=x2[n]+v2*Δt,  
  {n,NumFrames,2*NumFrames}]
```

```
Print["Velocity of particle 1 (after collision) =", v1]
```

```
Print["Velocity of particle 2 (after collision) =", v2]
```

3. Εμφάνιση αποτελεσμάτων



1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Απεικόνιση της κρούσης-Αρχικές ρυθμίσεις

Για να απεικονίσουμε μία (Animate) ή περισσότερες (Do..Show) απεικονίσεις πρέπει πριν τις αντίστοιχες εντολές να ρυθμίσουμε κατάλληλα τις γραφικές επιλογές της Mathematica με την παρακάτω εντολή

```
SetOptions[Graphics,AspectRatio->0.5,Axes->True, Ticks->{{-5.,-2.5,2.5,5.0},None}, PlotRange->{{-6.0,6.0},{-2.0,2.0}}];
```

Ας δούμε τι κάνει η κάθε παράμετρος

Axes->True: Εμφάνιση αξόνων x, y

Ticks->{{-5.,-2.5,2.5,5.0},None}: τοποθετεί ticks στον οριζόντιο άξονα ενώ στον κάθετο όχι

AspectRatio->0.5: η κλίμακα του άξονα y ρυθμίζεται στο $\frac{1}{2}$ εκείνης του x

PlotRange->{{-6.0,6.0},{-2.0,2.0}}: ρυθμίζει τα όρια των αξόνων

Θα δούμε παρακάτω και τις επόμενες παραμέτρους

PointSize[0.09]: το σημείο ορίζεται στο 9% του πλάτους της οθόνης. Το Point1 αντιστοιχεί στη δεξιά σφαίρα με συντεταγμένες $x1[t]$ και $y1[t]=0$. Η αριστερή σφαίρα έχει συντεταγμένες $x2[t]$, $y=0$

Show: εμφανίζει ταυτόχρονα τις δύο σφαίρες στην οθόνη



1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Απεικόνιση της κρούσης

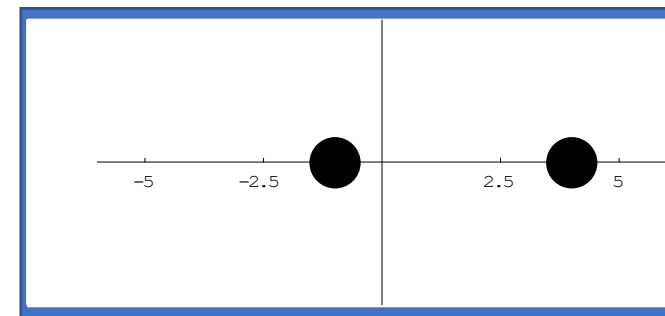
Με δεδομένα τις θέσεις της κάθε μάζας σε **10** στιγμιότυπα πριν και **10** στιγμιότυπα μετά από την κρούση το πρόγραμμα απεικονίζει τα **20** στιγμιότυπα του φαινομένου.

```
SetOptions[Graphics,AspectRatio->0.5,Axes->True, Ticks->{{-5.,-2.5,2.5,5.0},None}, PlotRange->{{-6.0,6.0},{-2.0,2.0}}];
```

```
Do[Point1=Graphics[{PointSize[0.09],Point[{x1[n],0.}]}];
```

```
Point2=Graphics[{PointSize[0.09],Point[{x2[n],0.}]}];
```

```
Show[{Point1,Point2}, DisplayFunction->Print, {n,0,2*NumFrames}]
```

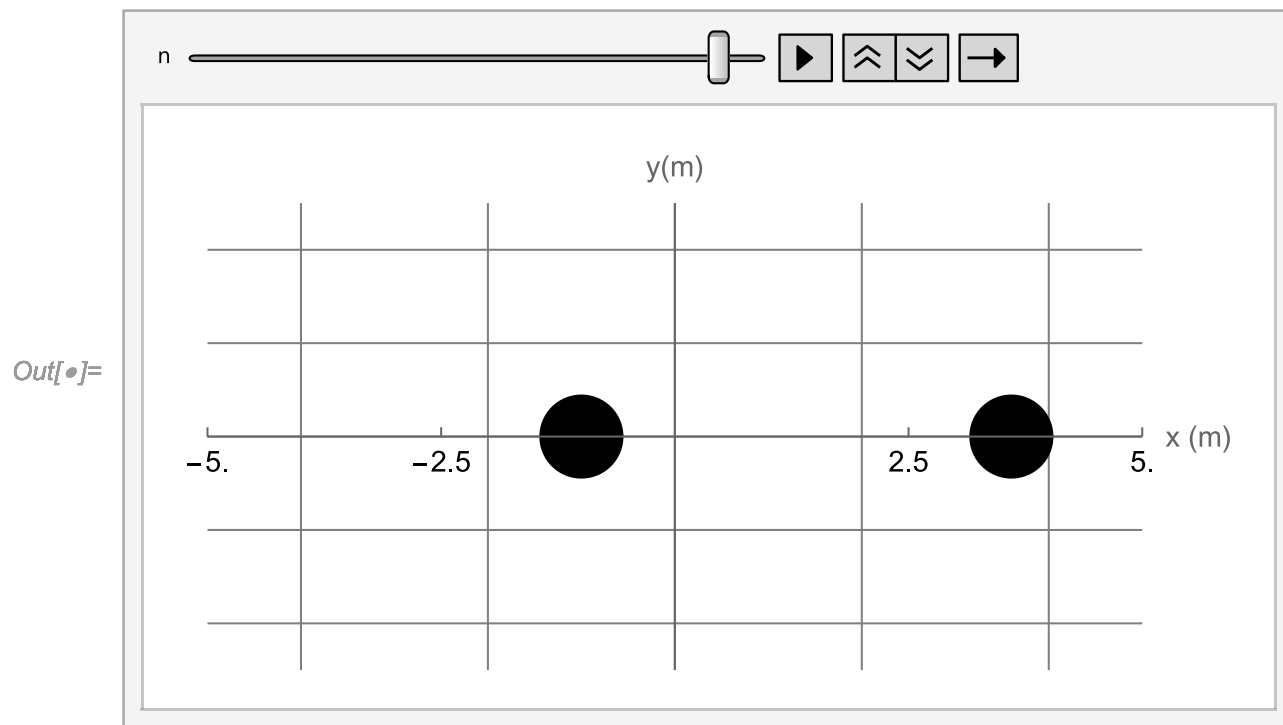




1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις

Απεικόνιση της κρούσης Με δεδομένα τις θέσεις της κάθε μάζας σε **10** στιγμιότυπα πριν και **10** στιγμιότυπα μετά από την κρούση το πρόγραμμα απεικονίζει το φαινόμενο με ένα διαδραστικό παράθυρο μιας εντολής **Animate**

```
Animate[Show[Graphics[ {PointSize[0.09], Point[{x1[n], 0}], Point[{x2[n], 0}]]],  
{n, 0, 2*NumFrames, 1}, AnimationRunning -> False]
```





1 | Πρόβλημα 10.4: Ελαστικές Κρούσεις



1. Επαναλάβετε την παραπάνω διαδικασία θέτοντας τις μάζες (SI) από τους παρακάτω πίνακες διατηρώντας τις υπόλοιπες παραμέτρους ίδιες. Καταγράψτε κάθε φορά την νέα απεικόνιση και καταγράψτε τις τελικές ταχύτητες των μαζών u'_1 , u'_2 μετά την κρούση.

$m_1=1$	$m_1=1$	$m_1=1$	$m_1=1$	$m_1=1$	$m_1=1$
$m_2=1$	$m_2=2$	$m_2=4$	$m_2=8$	$m_2=16$	$m_2=32$
$u'_1=.....$	$u'_1=.....$	$u'_1=.....$	$u'_1=.....$	$u'_1=.....$	$u'_1=.....$
$u'_2=.....$	$u'_2=.....$	$u'_2=.....$	$u'_2=.....$	$u'_2=.....$	$u'_2=.....$

$m_1=1$	$m_1=2$	$m_1=4$	$m_1=8$	$m_1=16$	$m_1=32$
$m_2=1$	$m_2=1$	$m_2=1$	$m_2=1$	$m_2=1$	$m_2=1$
$u'_1=.....$	$u'_1=.....$	$u'_1=.....$	$u'_1=.....$	$u'_1=.....$	$u'_1=.....$
$u'_2=.....$	$u'_2=.....$	$u'_2=.....$	$u'_2=.....$	$u'_2=.....$	$u'_2=.....$

2. Δημιουργήστε τη γραφική παράσταση των u'_1 , u'_2 ως (χωριστές γραφικές παραστάσεις) συνάρτηση του m_2 (ListPlot) και σχολιάστε τα αποτελέσματα. Τι συμβαίνει όταν $m_1 \gg m_2$ ή $m_2 \gg m_1$;

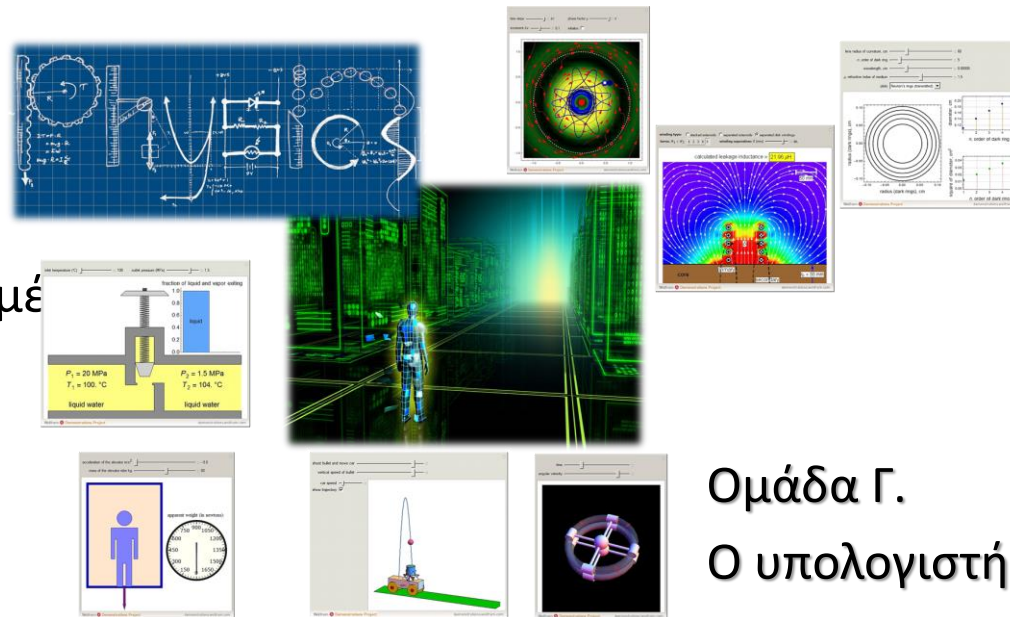
Πρόβλημα 10.1: Βολές

Πρόβλημα 10.2: Κινήσεις Πλανητών

Πρόβλημα 10.3: Απλό μαθηματικό εκκρεμές

Πρόβλημα 10.4: Ελαστικές Κρούσεις

Πειράματα Επίδειξης



Ομάδα Γ.

Ο υπολογιστής ως επιστημονικό εργαλείο

Εργασία-10: Η mathematica στην υπηρεσία της Φυσικής

Wolfram
Mathematica® 10



2 | Προγράμματα Επίδειξης

**Τι είναι τα αρχεία Computable Document Format™ (CDF)?**

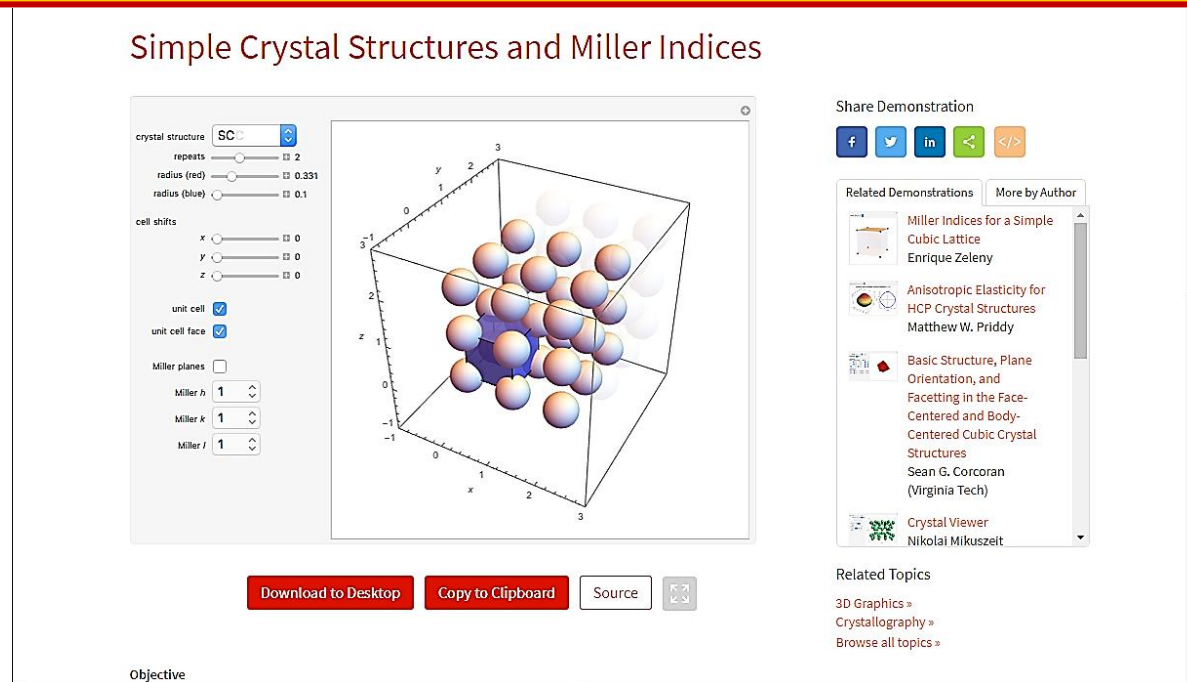
- Τα σημερινά online αρχεία έχουν διαδεχθεί τα έντυπα αρχεία διατηρώντας όμως πολλά από τα μειονεκτήματά τους, όπως είναι η έλλειψη διαδραστικότητας.
- Τα αρχεία CDF είναι αρχεία που έχουν δημιουργηθεί με τη Mathematica και λειτουργούν αυτόνομα χωρίς να χρειάζεται η Mathematica, παρέχουν διαδραστικά εργαλεία προκειμένου να δημιουργούν ρεαλιστικές απεικονίσεις φαινομένων.
- Η Wolfram έχει δημιουργήσει ένα πρόγραμμα (Wolfram CDF player) που παρέχεται δωρεάν και που διαχειρίζεται τέτοια αρχεία ενώ υπάρχει και μια τεράστια βιβλιοθήκη τέτοιων εκτελέσιμων αρχείων (<https://www.wolfram.com/cdf/>) που παρέχουν οπτικοποιημένο υλικό από φαινόμενα διαφόρων επιστημονικών κλάδων και υπολογισμών, ενώ δίνεται και ο αντίστοιχος κώδικας της Mathematica προκειμένου οι χρήστες να εξοικειωθούν με τη δημιουργία σύνθετου κώδικα και οπτικοποιημένων διαδραστικών εργαλείων.



2 Προγράμματα Επίδειξης



Επιλέγοντας το αρχείο **Simple Crystal Structures and Miller Indices** εμφανίζεται το παράθυρο που μου δείχνει το συγκεκριμένο αρχείο.



Όπως φαίνεται από το αναδυόμενο ενημερωτικό παράθυρο που εμφανίζεται (κάτω αριστερά εικόνα) στο παράθυρο της προεπισκόπησης, το συγκεκριμένο αρχείο λειτουργεί με τον καλύτερο τρόπο όταν βρίσκεται τοπικά στον υπολογιστή.

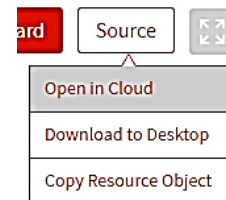
This demonstration is optimized for desktop

Download to desktop

Replay Preview

Copy to clipboard

Μπορώ λοιπόν από το παράθυρο αυτό ή από τα εργαλεία κάτω από την προεπισκόπηση του αρχείου CDF να αποθηκεύσω το αρχείο στον υπολογιστή μου (Download to Desktop), να το αντιγράψω στο πρόχειρο (Copy to Clipboard) ή να αντιγράψω τις αντίστοιχες εντολές κώδικα της Mathematica (Source) όπως φαίνεται στη δεξιά εικόνα.





2 | Προγράμματα Επίδειξης

Simple Crystal Structures and Miller Indices

Στο δεξί μέρος του παραθύρου υπάρχουν επιλογές για **Related Demonstrations** και **Related Topics**, ώστε να οδηγηθώ γρήγορα σε άλλα σχετικά αρχεία **CDF** ή σε άλλα σχετικά θέματα αντίστοιχα.

Στο κάτω αριστερό μέρος του παραθύρου υπάρχουν ορισμένες επιλογές που παρέχουν επιπλέον πληροφορίες για τη συγκεκριμένη εφαρμογή όπως φαίνεται στις εικόνες που ακολουθούν.

Η επιλογή **Objective** στο κάτω αριστερό της προεπισκόπησης εμφανίζει μια σύντομη περιγραφή και πληροφορίες για το στόχο της συγκεκριμένης εφαρμογής και πως λειτουργεί (**Goal/Tasks**), ενώ εμφανίζει και τα ονόματα των δημιουργών της.

Στο κάτω αριστερό μέρος του παραθύρου υπάρχουν ορισμένες επιλογές που παρέχουν επιπλέον πληροφορίες για τη συγκεκριμένη εφαρμογή όπως φαίνεται στις εικόνες που ακολουθούν.

Η επιλογή **Objective** στο κάτω αριστερό της προεπισκόπησης εμφανίζει μια σύντομη περιγραφή και πληροφορίες για το στόχο της συγκεκριμένης εφαρμογής και πως λειτουργεί (**Goal/Tasks**), ενώ εμφανίζει και τα ονόματα των δημιουργών της.

Αντίστοιχα κάτω από την επιλογή **Objective** εμφανίζονται οι επιλογές **SnapShots** και η επιλογή **Permanent Citation** που δείχνουν ορισμένα φωτογραφικά στιγμιότυπα της εκτέλεσης της εφαρμογής αυτής και τα στοιχεία που πρέπει να αναφέρονται εφόσον συμπεριληφθεί σε μια επιστημονική αναφορά, που όπως συζητήσαμε στο 1^ο κεφάλαιο θα πρέπει να συμπεριλαμβάνει τις βιβλιογραφικές της πηγές από όπου έχει αντλήσει πληροφορίες.

Objective

This Demonstration interactively illustrates several crystal structures, showing unit cells and Miller indices.

Goals/Tasks

1. Repetition of unit cell to make crystals
2. Differences in SC, BCC, FCC, HCP and various simple ceramics structures
3. Conventional unit cell (use shift in three directions)
4. Contact condition for atoms (by changing atomic radius)
5. Close-packed directions and close-packed planes
6. Draw planes for Miller indices
7. Explore crystals by rotation and zoom

The last may run slowly, so be patient when there are many 3D spheres and planes to draw!

[\[less\]](#)

Contributed by: [Yunfeng Shi](#) and [Adam Bross](#) (April 2018)
(MSE@RPI)

Open content licensed under [CC BY-NC-SA](#)



2 Προγράμματα Επίδειξης

Physics Demonstrations

The screenshot shows the Wolfram Demonstrations Project website. The browser address bar displays `demonstrations.wolfram.com/topic.html?limit=20&topic=Physics`. The page header includes the Wolfram logo and the text "WOLFRAM Demonstrations Project" with a subtext "12,000+ Interactive Demonstrations Powered by Notebook Technology". Navigation links include SEARCH, EXPLORE, LATEST, ABOUT, PARTICIPATE, and AUTHORIZING AREA. The main content area is titled "PHYSICS DEMONSTRATIONS" and shows a grid of 15 demonstration thumbnails. Each thumbnail includes a small image and a title. The titles are: "Electrostatic Force between Charged Conducting Spheres", "Chemical Potential Dependence on Temperature and Pressure", "Temperature-Entropy Diagram for Water", "Injecting a Liquid into an Evacuated Tank", "Marcus Theory of Electron Transfer 2: Semiclassical Marcus Equation in Three Dimensions", "Two Rotating Waves in de Broglie-Bohm Mechanics", "Heat Flow across Circuit Board", "Emissive Power for a Blackbody over a Range of Wavelengths", "Feedback Control of Cellular Concentration in a Continuous Bioreactor (Turbidostat)", "Marcus Theory of Electron Transfer 1: Classical versus Semiclassical Models", "Michaelis-Menten", "Boltzmann's Analysis", "Motion of Single Ion in", "Torque Using Cross", and "Semiclassical".



XRayReflectionByPlanesOfAtomsInACubicLattice.cdf



WeightOfAPersonRidingInAnElevator.cdf



ElectrostaticForceBetweenChargedConductingSpheres.cdf



PropagationOfReflectedAndRefractedWavesAtAnInterface.cdf



MotionOfADiskHangingFromASpringOnAnInclinedPlane.cdf



MagneticDipoleInAUniformMagneticField.cdf



AlternatingCurrentGenerator.cdf



LightRefractionWithAWaterdrop.cdf



ElectricFieldLinesDueToACollectionOfPointCharges.cdf



ForcesOnARollerCoaster.cdf



SimpleHarmonicMotion.cdf



RotatingSpaceStation.cdf



Manometers.cdf



ElectromagneticWave.cdf



TheSolenoid.cdf



ForceBalanceOnAMovingCart.cdf



Πως μπορεί κανείς να λύσει προβλήματα Φυσικής με τη βοήθεια της Mathematica

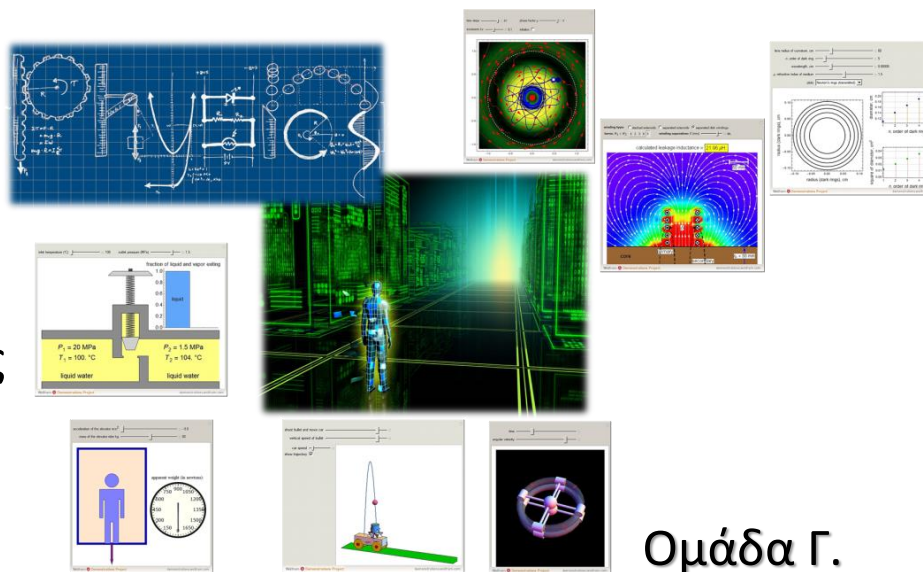
Πρόβλημα 10.1: Βολές

Πρόβλημα 10.2: Κινήσεις Πλανητών

Πρόβλημα 10.3: Απλό μαθηματικό εκκρεμές

Πρόβλημα 10.4: Ελαστικές Κρούσεις

Πειράματα Επίδειξης



Ομάδα Γ.

Ο υπολογιστής ως επιστημονικό εργαλείο

Η εργασία 10 περιλαμβάνει:

1. Επίλυση ενός εκ των τεσσάρων προβλημάτων
2. Επιλογή ενός πειράματος επίδειξης και εξήγηση του προβλήματος που επιλύει και των δυνατοτήτων του.

Εργασία-10: Η mathematica στην υπηρεσία της Φυσικής

Wolfram
Mathematica® 10