ΦΥΣ 347 – Υπολογιστικές Μέθοδοι στη Φυσική ΙΙ

3η Εργασία **Επιστροφή: 08/10/18**

Υπενθύμιση: Οι εργασίες πρέπει να επιστρέφονται με e-mail που θα στέλνετε από το πανεπιστημιακό σας λογαριασμό το αργότερο μέχρι την ημερομηνία που αναγράφεται.
Σα θέμα (subject) του e-mail θα πρέπει να αναγράφεται την εργασία (Phy347_Hm03).
Κάθε αρχείο που επισυνάπτετε (attach) στο e-mail σας θα πρέπει να έχει το όνομα στη μορφή username_hmX.tgz όπου username είναι το username του e-mail σας και X ο αριθμός της εργασίας. Επίσης σα πρώτο σχόλιο μέσα σε κάθε file που περιέχει το πρόγραμμά σας θα πρέπει να αναφέρεται το ονοματεπώνυμό σας. Οι εργασίες είναι ατομικές και πανομοιότυπες εργασίες δε θα βαθμολογούνται.

1. Στο πρόβλημα αυτό θα χρησιμοποιήσετε το λογισμικό ROOT ώστε να κάνετε τις γραφικές παραστάσεις που ζητούνται για την αριθμητική λύση ενός προβλήματος διαφορικής εξίσωσης. Το πρόγραμμά σας μπορείτε να το γράψετε είτε σε μορφή συνάρτησης την οποία θα τρέξετε μέσα στο περιβάλον του ROOT ή μπορείτε αν έχετε κατεβάσει στον προσωπικό σας υπολογιστή το λογισμικό ROOT να γράψετε ένα πρόγραμμα το οποίο θα πρέπει να κάνετε link με τη βιβλιοθήκη του ROOT δίνοντας την ακόλουθη εντολή:

```
g++ -I `root-config --incdir` <όνομα προγράμματός> `root-config --cflags --libs` \
    -L $ROOTSYS/lib –lHtml  -lMinuit  -lMathCore  -o <όνομα του executable>
```

Θα μπορούσατε για να αποφύγετε να γράφετε κάθε φορά την εντολή αυτή να δημιουργήσετε ένα *script* (πρόγραμμα για εντολές φλοιού) και να τρέχετε κάνοντας το compilation. Έστω δημιουργείτε με τον *emacs* ένα αρχείο με όνομα *CompileWithRoot.sh*
Προσέξτε ότι το ακρόνυμο του αρχείου είναι *sh* για να δείχνει ότι είναι αρχείο με εντολές φλοιού. Στο αρχείο αυτό προσθέστε τις δύο παραπάνω γραμμές που γράψαμε για το compilation του προγράμματός σας. Επειδή ωστόσο δεν θέλουμε να κάνουμε *edit* το αρχείο και να αλλάζουμε το όνομα του προγράμματός μας και του παραγόμενου executable θα γράψουμε την εντολή ως εξής:

```
g++ -I `root-config --incdir`  $1  `root-config --cflags --libs` \
    -L $ROOTSYS/lib –lHtml  -lMinuit  -lMathCore  -o $2
```

Παρατηρήστε το $1 και $2. Αυτό σημαίνει ότι θα περάσουμε δύο παραμέτρους εκτός τους προγράμματος. Η $1 θα είναι το όνομα του προγράμματος και η $2 το όνομα του executable. Σώστε το file *CompileWithRoot.sh.* Για να μπορέσετε να τρέξετε το *script* αυτό σαν μία εντολή στο terminal, θα πρέπει να του επιτρέψετε να μπορεί να εκτελέσει την εντολή. Αυτό μπορείτε να το κάνετε με την εντολή: *chmod +x CompileWithRoot.sh*
Η εντολή *chmod* (*change mode*) δίνει *+x* (το + προσφέρει, το – αφαιρεί) εκτελεστικό δικαίωμα στην εντολή που ακολουθεί. Αυτό χρειάζεται να το κάνετε μόνο μία φορά για κάποιο file.
Μπορείτε τώρα να τρέξετε την εντολή CompileWithRoot.sh για το πρόγραμμά σας π.χ.
./CompileWithRoot.sh test.CC test.x

Στο πρόβλημα που έχετε να λύσετε. Θεωρήστε ότι έχετε ένα σώμα μάζας $m = 1kg$ στο άκρο ενός ελατηρίου σταθεράς $k = 1N/m$. Χρησιμοποιήστε την μέθοδο του *Euler* για να βρείτε τη θέση, ταχύτητα και ενέργεια του σώματος συναρτήσει του χρόνου και να τα συγκρίνετε με αυτό που θα αναμένατε λύνοντας αναλυτικά το πρόβλημα. Θα πρέπει να δημιουργήσετε κατάλληλα histograms στα οποία θα αποθηκεύσετε τις μεταβλητές που

μελετάτε και τα histograms αυτά θα τα αποθηκεύσετε σε ένα αρχείο με όνομα *EulerOscillator.root.*
Μπορείτε να δείτε τα histograms ανοίγοντας το root file μέσα στο λογισμικό του ROOT με την εντολή *TFile* file1=new TFile*("*EulerOscillator.root"*);
Αν τα histograms έχουν ονόματα *h1, h2,...etc* μπορείτε να τα ζωγραφίσετε με την εντολή *h*1->*Draw*(); etc. Αν θέλετε να το συγκρίνετε με κάποιο άλλο histogram (θεωρητικές τιμές) μπορείτε να ζητήσετε το 2^{o} histogram να ζωγραφιστεί μαζί με το πρώτο. Έτσι αν έχετε ήδη δώσει την εντολή *h*1->*Draw*(); μπορείτε να δώσετε την *h*2->*Draw*(*"same"*);

2. Στο πρόγραμμα αυτό θα χρησιμοποιήσετε την ιδέα της *structure,* τυχαίους αριθμούς όπως είδατε στο εργαστήριο 4 και την ιδέα του *tree* που υπάρχει στο λογισμικό ROOT.

Θεωρήστε ότι βρίσκεστε σε ένα μονοδιάστατο κόσμο και ότι μπορείτε να κινείστε είτε δεξιά ή αριστερά. Το βήμα που μπορείτε να κάνετε είναι το ίδιο μεγέθους και μπορείτε να το διαλέξετε τυχαία είτε αριστερά ή δεξιά. Επομένως κινήστε τυχαία σε μία διάσταση και αυτό οδηγεί στην έννοια του τυχαίου περιπάτου σε μία διάσταση που αποτελεί και την αρχή μελέτης διαφόρων φαινομένων όπως αυτό της διάχυσης. Η ερώτηση είναι πού θα βρίσκεται ο τυχαίος περιπατητής αφού έχει κάνει Ν βήματα. Έστω x η απόσταση του περιπατητή από το σημείο εκκίνησης, τότε $x = d \times m$ όπου d το μήκος του βήματος και m ο αριθμός των βημάτων. Επειδή τα βήματα λαμβάνονται τυχαία, η πιθανότητερη θέση στην οποία θα βρίσκεται ο περιπατητής θα είναι η θέση x=0.

Για να μπορέσουμε να μελετήσουμε την κίνηση αυτή θα χρειαστεί να προσομοιόσουμε την κίνηση του περιπατητή με N τυχαία βήματα. Ωστόσο αν το κάνουμε αυτό μόνο μία φορά δεν θα μπορούμε να διαπιστώσουμε αν το αποτέλεσμα της συγκεκριμένης δοκιμής πραγματοποιείται πάντοτε αλλά θα πρέπει να ελέγξουμε το αποτέλεσμα πολλών τέτοιων ξεχωριστών διαδρομών όλες με τον ίδιο αριθμό βημάτων. Έτσι το αποτέλεσμά μας αποκτά στατιστική δύναμη. Θα πρέπει δηλαδή να δούμε ποιά είναι η τελική θέση του περιπατητή μετά από N βήματα για πολλές διαφορετικές δοκιμές. Θα πρέπει η μέση τιμή της τελικής θέσης να αντιστοιχεί σε μια διονυμική κατανομή με μέσο το x=0.

Θα πρέπει να γράψετε ένα πρόγραμμα (ή συνάρτηση που τρέχει μέσα στο λογισμικό ROOT) που προσομοιώνει 10000 διαφορετικούς τυχαίους περιπάτους με 100 βήματα ο καθένας. Θα πρέπει τις τιμές της θέσης του περιπατητή *x,* του αριθμού του βήματος *nsteps* για να βρεθεί στη θέση αυτή, του αριθμού των βημάτων που κινήθηκε αριστερά *nLstep* για να βρεθεί στη θέση αυτή, του αριθμού των βημάτων που κινήθηκε δεξιά *nRstep* για να βρεθεί στη θέση αυτή, και του αριθμού της διαδρομής που προσομοιώνεται *jtrial*, να αποτελούν μέλη μιας *structure* με όνομα *walk*. Για κάθε βήμα θα πρέπει η *structure* να αποθηκεύεται μέσα στο *tree* του ROOT. Μετά το τέλος του μέγιστου αριθμού των περιπάτων θα πρέπει να αποθηκεύσετε το *tree* σε ένα αρχείο για περεταίρω επεξεργασία. Θεωρήστε ότι το βήμα *d* και σαν αποτέλεσμα η θέση *x* είναι ακέραιοι.

Για να δημιουργήσετε το *tree* μπορείτε να δώσετε την εντολή:

*TTree *random_walk=new TTree*(*"random_walk", "tree with randoms*");

Το tree που δημιουργούμε έχει το όνομα *random_walk.*

random_walk->Branch("*walk*", &*walk.x*, "*x/I:nsteps/I:nLstep/I:nRstep/I:jtrial/I"*)

Το tree έχει ένα *branch* με όνομα *walk* και στο branch αυτό αντιγράφουμε όλη την *structure walk* ξεκινώντας με την θέση μνήμης του πρώτου στοιχείου της structure (&*walk.x*) που δηλώνεται στο tree με την μεταβλητή *x*, ενώ οι υπόλοιπες μεταβλητές της

structure θα αποθηκευτούν στο tree κάτω από τις μεταβλητές *nsteps, nLstep, nRstep, jtrial* όπου όλες θα είναι τύπου *Integer.*

Μπορείτε να δείτε τι θα αποθηκευτεί στο tree χρησιμοποιώντας τον περιηγητή (browser) του root. Ξεκινήστε το λογισμικό ROOT. Πληκτρολογήστε την εντολή: *TBrowser b;* Θα ανοίξει το παράθυρο του browser. Βρείτε το *root* file που δημιουργήσατε και κάντε double click. Θα δείτε το όνομα του *tree* και κάνετε διπλό click στο όνομα του *tree* οπότε θα δείτε το όνομα του *branch.* Θα δείτε το όνομα των μεταβλητών του branch. Στην προκειμένη περίπτωση θα δείτε *x, nsteps, nLsteps, nRsteps, jtrials.* Οι μεταβλητές έχουν το σχήμα των φύλλων του κλαδιού. Άν κάνετε διπλό click σε μια μεταβλητή (π.χ. τη *x*) θα δείτε την κατανομή της.

Θα μπορούσατε να δώσετε τις εντολές στο ROOT ως εξής:

Root> TFile infile = new TFile*(“*MyRandomWalk.root”*);*
Root> infile->Print();
Root> infile->ls();
*Root> TTree *my_tree = random_walk; // Pointer sto tree random_walk του file.*
Root> my_tree->Draw(“x”,“ntrials==99”); // ζωγραφίζει το x για το τελευταίο βήμα
// όλων των προσπαθειών