ΦΥΣ 140 – Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών

4η Εργασία **Επιστροφή: 03/11/2023**

__**Υπενθύμιση:**__ Οι εργασίες πρέπει να επιστρέφονται με e-mail στο fotis@ucy.ac.cy που θα στέλνετε από το πανεπιστημιακό σας λογαριασμό το αργότερο μέχρι την ημερομηνία που αναγράφεται.
Ως subject του e-mail θα πρέπει να αναγράφεται την εργασία (username_phy140_hmX όπου X ο αριθμός της εργασίας)
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 σαν 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. Για να κάνετε ένα tgz file (ουσιαστικά tar zipped file) θα πρέπει να δώσετε στο terminal την εντολή tar -czvf username_hmX.tgz *.py όπου py είναι όλα τα py files των προγραμμάτων σας.

1. Μια φοιτήτρια χρησιμοποιεί τη διάταξη ενός μαθηματικού εκκρεμούς για να επαληθεύσει την εξάρτηση της περιόδου από την τετραγωνική ρίζα του μήκους του εκκρεμούς και κατόπιν να μετρήσει την επιτάχυνση της βαρύτητας στην τοποθεσία πραγματοποίησης του πειράματός της. Οι μετρήσεις της περιέχονται σε ένα αρχείο με όνομα *pendulum.dat.* Οι τιμές στο αρχείο είναι αυτές που φαίνονται στον παρακάτω πίνακα. (Θα πρέπει να αντιγράψετε τις τιμές αυτές στο αρχείο *pendulum.dat* ακριβώς όπως δίνονται στον πίνακα). Η πρώτη στήλη περιέχει τις τιμές του μήκους του εκκρεμούς ενώ η 2η στήλη περιέχει τις τιμές της περιόδου.

Για να εξακριβώσει τη συναρτησιακή εξάρτηση των δύο μεγεθών ($T = f(L)$) αποφασίζει να λογαριθμήσει τις μετρήσεις της και κατόπιν να προσαρμόσει τις τιμές αυτές με την καλύτερη ευθεία που περνά από τα λογαριθμισμένα σημεία των μετρήσεων. Η καλύτερη ευθεία μπορεί να επιτευχθεί με την βοήθεια της μεθόδου των ελαχίστων τετραγώνων. Σύμφωνα με τη μέθοδο αυτή, οι τύποι που δίνουν την κλίση *B* της ευθείας και την τεταγμένη, *A,* της ευθείας με τον *y*-άξονα είναι:

$$A = \frac{\sum x^2 \sum y - \sum x \sum (x\cdot y)}{\Delta},\ B = \frac{N\sum(x\cdot y) - \sum x \sum y}{\Delta} \quad \text{και} \quad \Delta = N\sum x^2 - \left(\sum x\right)^2.$$

όπου οι μεταβλητές *x* και *y* καλύπτουν όλες τις μετρήσεις *N* και τα αθροίσματα είναι πάνω σε όλες τις μετρήσεις *N.*

Για τον υπολογισμό του σφάλματος των τιμών *A* και *B*, υποθέτει ότι όλες οι μετρήσεις της περιόδου ακολουθούν Gaussian κατανομή γύρω από την "αληθινή" τιμή $A + Bm_i$ με εύρος σ_l. Επομένως οι αποκλίσεις $l_i - A - Bm_i$ είναι όλες Gaussian κατανεμημένες με την κεντρική τιμή ίση με μηδέν και το ίδιο εύρος σ_l. Μια καλή εκτίμηση της σ_y δίνεται από την σχέση $\sigma_y = \sqrt{\frac{1}{N-2}\sum(y_i - A - Bx_i)^2}$ όπου το άθροισμα είναι ως προς όλες τις μετρήσεις *N.* Με βάση τη σχέση αυτή, οι αβεβαιότητες των *A* και *B* είναι: $\sigma_A = \sigma_y\sqrt{\frac{\sum x^2}{\Delta}}$ και $\sigma_B = \sigma_y\sqrt{\frac{N}{\Delta}}$.

Με βάση τους παραπάνω τύπους να γράψετε ένα πρόγραμμα το οποίο διαβάζει τις τιμές του πίνακα των μετρήσεών της που δίνονται από το αρχείο *pendulum.dat* και τις αποθηκεύει σε κατάλληλες λίστες. Κατόπιν το πρόγραμμα χρησιμοποιεί τις παραπάνω σχέσεις και υπολογίζει την ευθεία των ελαχίστων τετραγώνων και επομένως τις σταθερές *A*, *B* και τα αντίστοιχα σφάλματα. Θα πρέπει να επιστρέψετε το πρόγραμμά σας μαζί με τα αποτελέσματα για τις τιμές των *A, B* και τα αντίστοιχα σφάλματά τους ως σχόλια στο τέλος του προγράμματός σας.

(α) Κάντε το γράφημα του μήκους *L* και της περιόδου *T* χρησιμοποιώντας αρχικά γραμμικούς άξονες. Κάντε κατόπιν το ίδιο γράφημα αλλά χρησιμοποιώντας λογαριθμικούς άξονες. Τα δύο γραφήματα να πραγματοποιηθούν στον ίδιο canva ο οποίος να είναι χωρισμένος σε 2 μέρη στην οριζόντιο διεύθυνση.

(β) Υπολογίστε την κλίση της ευθείας και την τεταγμένη σύμφωνα με την μέθοδο των ελαχίστων τετραγώνων. Υπολογίστε επίσης το σφάλμα στις τιμές της περιόδου.

(γ) Χρησιμοποιώντας τις τιμές για την κλίση και την τεταγμένη κατασκευάστε την καλύτερη ευθεία και δείξτε την στο γράφημα log(T) vs log(L) μαζί με τα πειραματικά σημεία.

(δ) Τοποθετήστε την ευθεία ελαχίστων τετραγώνων στο γράφημα όπου οι άξονες είναι σε λογαριθμική κλίμακα. Στο γράφημα αυτό δείξτε γραφικά την τεταγμένη της ευθείας και την τιμή που αντιστοιχεί σε αυτή.

Τα γραφήματα των ερωτημάτων (γ) και (δ) να γίνουν στον ίδιο canva το ένα δίπλα στο άλλο στην *x*-διεύθυνση.

Με βάση τις μετρήσεις και τα αποτελέσματά σας και θεωρώντας ότι η σχέση που δίνει την περίοδο του εκκρεμούς συναρτήσει του μήκους είναι $T = 2\pi\sqrt{l/g}$ υπολογίστε την επιτάχυνση της βαρύτητας που προκύπτει από τις μετρήσεις αυτές. Θα πρέπει να αναγράψετε την τιμή που βρίσκετε στο γράφημα με μορφή text.

Χρησιμοποιήστε τα ακόλουθα δεδομένα:

Μέτρηση	Μήκος Εκκρεμούς, L (m) - x_i	Περίοδος, T_i (sec) - y_i
1	0.1	1.304
2	0.2	2.087
3	0.3	2.354
4	0.4	2.910
5	0.5	3.162
6	0.6	3.341
7	0.7	3.647
8	0.8	4.024
9	0.9	4.153
10	1.0	4.483

2. Να γράψετε μια συνάρτηση που να επιστρέφει κάθε μία από τις τρεις πρώτες σφαιρικές συναρτήσεις Bessel $j_n(x)$:

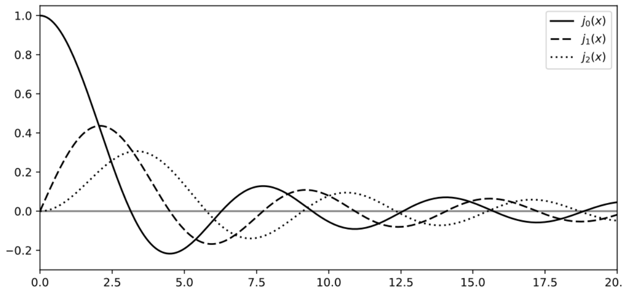
$$j_0(x) = \frac{sinx}{x}$$

$$j_1(x) = \frac{sinx}{x^2} - \frac{cosx}{x}$$

$$j_2(x) = \left(\frac{3}{x^2} - 1\right)\frac{sinx}{x} - \frac{3cosx}{x^2}$$

Η συνάρτησή σας θα πρέπει να δέχεται ως όρισμα έναν *numpy array x,* και την τάξη *n*, και θα πρέπει να επιστρέφει έναν πίνακα της προσδιοριζόμενης τάξης (0,1,2,…) των συναρτήσεων

Bessel. Προσέξτε ώστε η συνάρτησή σας συμπεριφέρεται σωστά για μικρές τιμές του x κοντά στο 0.
Δείξτε τη χρήση της συνάρτησής σας γράφοντας ένα πρόγραμμα το οποίο κάνει το γράφημα των 3 συναρτήσεων Bessel για $0 \leq x \leq 20$. Το γράφημά σας θα πρέπει να είναι όπως αυτό που φαίνεται δίπλα. Κάτι για να σκεφθείτε: Μπορείτε να γράψετε τον $j_1(x)$ όρο συναρτήσει του $j_0(x)$ ενώ ο όρος $j_2(x)$ μπορεί να γραφεί συναρτήσει του $j_0(x)$ και $j_1(x)$. Μπορείτε να εκμεταλλευτείτε την ιδιότητα αυτή ώστε να γράψετε ένα περισσότερο αποδοτικό πρόγραμμα για τον υπολογισμό των $j_1(x)$ και $j_2(x)$. Για $x \ll 10^{-8}$, θα πρέπει να αναπτύξετε τη συνάρτηση κατά Taylor για τιμές κοντά στο x. Θυμηθείτε το ανάπτυγμα *Taylor* μιας συνάρτησης ως προς $x = 0$, δίνεται από την εξίσωση: $f(x) = f(0) + xf'(x)|_{x=0} + \frac{1}{2!}x^2f''(x)|_{x=0} + \frac{1}{3!}x^3f'''(x)|_{x=0} + \cdots$



3. Θεωρήστε την 2η άσκηση του 7ου εργαστηρίου αλλά αυτή τη φορά θα δημιουργήσετε 1000000 τυχαίους αριθμούς κατανεμημένους εκθετικά, $e^{\lambda x}$, χρησιμοποιώντας τη μέθοδο ***expovariate*** αντί της μεθόδου *gauss*. Η συνάρτηση *expovariate* δέχεται ένα όρισμα, το οποίο είναι η τιμή του λ στην προηγούμενη σχέση. Θεωρήστε ότι $\lambda = 0.8$. [Αν η τιμή του λ είναι αρνητική τότε οι τιμές που επιστρέφει η μέθοδος είναι αρνητικές]. Κάντε το ιστόγραμμα της κατανομής των 1000000 τυχαίων αριθμών χρησιμοποιώντας κατάλληλο αριθμό υποδιαστημάτων ώστε το εύρος του κάθε bin να είναι 0.1 και η μέγιστη τιμή του στον x-άξονα να είναι 40. Θεωρήστε ότι η επιλογή του ορίσματος density είναι με την τιμή none. Δημιουργήστε κατόπιν μια νέα list οι τιμές των στοιχείων της οποίας προκύπτουν αν στις τιμές του περιεχομένου του κάθε bin του ιστογράμματος που πήρατε από τη μέθοδο *expovariate* προσθέτετε το άθροισμα του περιεχομένου όλων των προηγούμενων bins. Κάντε το ιστόγραμμα των τιμών των στοιχείων της νέας αυτής λίστας. Κάντε το ιστόγραμμα αυτό δίπλα από το ιστόγραμμα που κατασκευάσατε αρχικά. Ξανακάντε τα δύο ιστογράμματα θέτοντας στην επιλογή *density = True.* Η κατανομή που θα πάρετε στην περίπτωση αυτή ονομάζεται Cumulative Density Function (CDF) και δίνει την πιθανότητα να βρεθεί μια τιμή της μεταβλητής στο x-άξονα, μεταξύ 0 και της εν λόγω τιμής.

4. Θεωρήστε το ανάπτυγμα Taylor της συνάρτησης $f(x) = \cos(x)$ ως προς $x = 0$. Κάντε τη γραφική παράσταση του πολυωνύμου *Taylor* προσθέτοντας επιπλέον όρους του αναπτύγματος κάθε φορά και συγκρίνετε με την συνάρτηση $f(x) = \cos(x)$. Το γράφημά σας θα πρέπει να γίνει για τιμές γωνιών x στο διάστημα $[-\pi, \pi]$. Θα πρέπει να προσθέσετε το ανώτερο μέχρι τον 6ο όρο του πολυωνύμου. Στο γράφημά σας θα πρέπει να φαίνονται 6 καμπύλες, μία για κάθε επιπλέον όρο που προσθέτετε στο πολυώνυμο, καθώς και η καμπύλη της συνάρτησης του $\cos(x)$. Η κάθε καμπύλη θα πρέπει να έχει το κατάλληλο κείμενο που περιγράφει την εξίσωση που εμφανίζεται.

Με την ευκαιρία, το πακέτο sympy περιέχει την μέθοδο *series* που μπορεί να σας δώσει το ανάπτυγμα *Taylor* μιας συνάρτησης. Για παράδειγμα έστω η συνάρτηση $f(x) = (1 + x)^n$. Θα μπορούσατε να γράψετε:

```
import sympy as sp
from sympy.abc import x, n  # Για να μην γράφετε x=Symbols('x')
f=(1+x)**n
x0 =0           # ανάπτυγμα ως προς x0=0
```

Το *removeO*() χρειάζεται ώστε ο τύπος να μην περιέχει την τάξη του όρου στον οποίο γίνεται η αποκοπή του αναπτύγματος. Δοκιμάστε με και χωρίς το *removeO*() το παραπάνω παράδειγμα για να δείτε τη διαφορά στο αποτέλεσμα.

Ακολουθήστε τις οδηγίες στην ιστοσελίδα των ανακοινώσεων του μαθήματος για να εγκαταστήσετε το πακέτο *sympy.*

5. Χρησιμοποιώντας το πακέτο *sympy* της *python* να κάνετε το γράφημα του αναπτύγματος *Taylor* της συνάρτησης $f(x) = \sin(x)/x$ ως προς $x_0 = 0$ για *x* στο διάστημα $[-5.5, 5.5]$ και να συγκρίνετε το αποτέλεσμα με αυτό της συνάρτησης $f(x)$ όπως υπολογίζεται από το πακέτο *sympy.*

<u>*Υπόδειξη:*</u> Για να υπολογίσετε τις τιμές μιας *lambdify* συνάρτησης πρέπει να δώσετε ως όρισμα στην συνάρτηση *lambdify* το πακέτο που θα χρησιμοποιηθεί για τον υπολογισμό της. Για παράδειγμα `st = lambdify(x,x+1,module='numpy')` δηλώνει ότι η *python* συνάρτηση που προκύπτει από την συνάρτηση `x+1` θα υπολογισθεί χρησιμοποιώντας το πακέτο *numpy.* Θα μπορούσαμε να έχουμε *numpy', sympy, math.* Αν η επιλογή είναι *numpy* μπορεί να χρησιμοποιηθούν οι αριθμητικές πράξεις και συναρτήσεις της *numpy.* Αν η επιλογή για το module ήταν *sympy* θα μπορούσαμε να πάρουμε μια νέα συνάρτηση. Για παράδειγμα έστω ότι δώσαμε `evalf=lambdify(x,x+1,'sympy')` Καλώντας `evalf(cos(x))` θα έχει αποτέλεσμα `cos(x)+1`.

Θα μπορούσαμε να υπολογίσουμε το αποτέλεσμα μιας *sympy* συνάρτησης χωρίς να την μετατρέψουμε σε *lambdify* αντικείμενο τύπου συνάρτησης. Μπορούμε να χρησιμοποιήσουμε την μέθοδο *subs*(). Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί είτε για αριθμητικό υπολογισμό ή για αντικατάσταση μιας έκφρασης στην *sympy* συνάρτηση. Για παράδειγμα:

```
>> f=x+1
>> f.subs(x,2)
 3
>> f.subs(x,cos(x))
cos(x)+1
```