

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΕΡΓΑΣΙΕΣ
ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ: 26/05/2021

ΕΡΓΑΣΙΑ 1

- (α) Να γράψετε συνάρτησεις Python που θα υλοποιούν τους παρακάτω κανόνες ολοκλήρωσης (σύνθετη μορφή τους):
- Αριστερού παραλληλογράμμου
 - Δεξιού παραλληλογράμμου
 - Τραπεζίου
 - Simpson
- (β) Να εφαρμόσετε τους κανόνες ολοκλήρωσης που προγραμματίσατε στο παρακάτω πρόβλημα ολοκλήρωσης για το οποίο δίνεται η ακριβής λύση:

$$\int_0^{\frac{\pi}{4}} e^{3x} \sin(2x) dx = \frac{3}{13} e^{3\pi/4} + \frac{2}{13}$$

Συγκεκριμένα, να γράψετε κώδικα ο οποίος θα παράγει ένα log-log διάγραμμα το οποίο στον οριζόντιο άξονα θα έχει διάφορες τιμές για το βήμα διακριτοποίησης (h) (π.χ. $h = 1, 10^{-2}, 10^{-3}, \dots, 10^{-7}$) ενώ στον κάθετο άξονα θα έχει τις τιμές για το απόλυτο σφάλμα όπως αυτό προκύπτει προσεγγίζοντας το ολοκλήρωμα με καθέναν από τους κανόνες ολοκλήρωσης που προγραμματίσατε.

- (γ) Να σχολιάσετε τα αποτελέσματα της ανάλυσής σας (στο (β)) σε σχέση με τους γνωστούς τύπους σφαλμάτων για τις παραπάνω μεθόδους.

ΕΡΓΑΣΙΑ 2

Σε μια απομακρυσμένη δασική περιοχή ζουν κυρίως λαγοί και αλεπούδες. Έστω $x_0(t)$ ο αριθμός των λαγών (θηραμάτων) και $x_1(t)$ ο αριθμός των αλεπούδων (θηρευτών) την χρονική στιγμή t . Η σχέση μεταξύ των πληθυσμών των δύο αυτών ειδών μπορεί να μοντελοποιηθεί με το παρακάτω σύστημα κανονικών διαφορικών εξισώσεων:

$$x'_0 = 1,1x_0 - 0,4x_0x_1$$

$$x'_1 = 0,4x_0x_1 - 0,1x_1$$

- (α) Να γράψετε συνάρτηση Python που θα υλοποιεί την μέθοδο Euler με χρήση διανυσμάτων και θα επιτρέπει την αριθμητική επίλυση συστημάτων κανονικών διαφορικών εξισώσεων με αρχικές τιμές. Ο παρακάτω μη ολοκληρωμένος κώδικας μπορεί να σας βοηθήσει να ξεκινήσετε:

```
def euler(F,x0,t0,tmax,dt):  
    t = # set up a vector for time  
    # set up x so that it is an array where the columns  
    # are the different dimensions of the problem.  
    # In this problem there will be 2 columns and len(t) rows
```

```

x = np.zeros( (len(t), len(x0)) )
x[0,:] = x0 # insert the initial condition in the first row
for n in range(len(t)-1):
    x[n+1,:] = x[ ??? , ??? ] + dt*F(t[ ??? ], x[ ??? , ??? ])
return t, x

```

- (β) Χρησιμοποιείστε την συνάρτηση που δημιουργήσατε στο (α) για να επιλύσετε το παραπάνω σύστημα κανονικών διαφορικών εξισώσεων με αρχικές συνθήκες $x_0(0) = 20$ λαγούς και $x_1(0) = 1$ αλεπού, χρησιμοποιώντας $t_{max} = 200$ και κατάλληλο μικρό βήμα διακριτοποίησης (dt).
- (γ) Να δημιουργήσετε ένα διάγραμμα που να δείχνει την εξέλιξη του πληθυσμού των λαγών και των αλεπούδων σε σχέση με τον χρόνο και ένα διάγραμμα που να δείχνει την εξέλιξη του πληθυσμού των λαγών σε σχέση με τον πληθυσμό των αλεπούδων. Τι συμπεράσματα μπορείτε να βγάλετε από τα δύο αυτά διαγράμματα;
- (δ) Αν μειώσετε το βήμα διακριτοποίησης στο $\frac{1}{10}$ της αρχικής σας επιλογής, τι παρατηρείτε στα δύο διαγράμματα που δημιουργήσατε στο (γ); Τι παρατηρείτε ότι κάνει η μέθοδος Euler σε αυτή την περίπτωση;

Ο παρακάτω μη ολοκληρωμένος κώδικας μπορεί να σας βοηθήσει να ξεκινήσετε για να απαντήσετε τα ερωτήματα (β), (γ) και (δ):

```

F = lambda t, x: np.array([ ??? , ??? ])
x0 = [ ??? , ??? ] # initial conditions
t0 = 0
tmax = ???
dt = ??? # your choice. pick something small
t, x = euler(F,x0,t0,tmax,dt)
# Plot the solutions against time
??? .plot(t,x[:, ???],b-,t,x[:, ???],r--)
??? .grid()
??? .set_title(Time Evolution of Foxes Population and Rabbits Population)
??? .legend([which legend entry here,which legend entry here])
??? .set_xlabel(time)
??? .set_ylabel(No. of Rabbits and No. of Foxes )
# Plot one solution against the other
# In this plot time is implicit (not one of the axes)
??? .plot(x[:, ???], x[:, ???], k--)
??? .grid()
??? .title(Phase Plot)
??? .set_xlabel(???)
??? .set_ylabel(???)
plt.show()

```

Παραδοτέο στην ιστοσελίδα του μαθήματος στο elearning (Εργασία 2021):

Αρχείο .ipynb που θα περιέχει τον κώδικα και τις απάντησεις στα ανωτέρω ερωτήματα. Εναλλακτικά, μπορείτε να παραδώσετε ένα αρχείο με τον κώδικά σας και ένα αρχείο σε μορφή pdf με τις απαντήσεις σας στα παραπάνω ερωτήματα.