4ο Εργαστήριο

Ευστάθεια

Για την αριθμητική επίλυση ενός προβλήματος αρχικών τιμών (Π.Α.Τ.)

$$y'(t) = f(t, y(t)), \quad t \in [a, b], \quad y(0) = y_0$$

θεωρήσαμε άμεσες μεθόδους όπως του Euler αλλά και πεπλεγμένες μεθόδους όπως του Euler και του τραπεζίου. Έστω ένας ομοιόμορφος διαμερισμός του $[a, b]$, στα σημεία $t_n = a + nh$, $n = 0, \dots, N$, με βήμα $h = \dfrac{b-a}{N}$, υπολογίζουμε τις τιμές y_n που αποτελούν προσεγγίσεις στις τιμές $y(t_n)$, $n = 0, \dots, N$.

Άμεση Euler

$$y_{n+1} = y_n + hf(t_n, y_n), \quad n = 0, \dots, N-1.$$

Πεπλεγμένη Euler

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}), \quad n = 0, \dots, N-1.$$

Μέθοδος του τραπεζίου

$$y_{n+1} = y_n + \frac{h}{2}(f(t_{n+1}, y_{n+1}) + f(t_n, y_n)), \quad n = 0, \dots, N-1.$$

Για να μπορέσουμε να βρούμε την προσέγγιση αν θεωρήσουμε μια πεπλεγμένη μέθοδο θα χρειαστεί να βρούμε τη ρίζα μιας μη-γραμμικής συνάρτησης σε κάθε βήμα. Αν όμως η συνάρτηση f είναι γραμμική ως προς y, π.χ. $f(t, y) = g(t)y$, τότε οι παραπάνω πεπλεγμένες μέθοδοι υλοποιούνται εύκολα, π.χ.
Πεπλεγμένη Euler

$$y_{n+1} = \frac{1}{1 - hg(t_{n+1})} y_n, \quad n = 0, \dots, N-1.$$

η **Μέθοδος του τραπεζίου**

$$y_{n+1} = \frac{1 + h\dfrac{g(t_n)}{2}}{1 - h\dfrac{g(t_{n+1})}{2}} y_n, \quad n = 0, \dots, N-1.$$

Άσκηση 1: Έστω $y(t) = \frac{50}{2501}(\sin(t) + 50\cos(t)) - \frac{2500}{2501}e^{-50t}$, στο $[0, 1]$ η οποία είναι λύση στο

$$y'(t) = -50(y(t) - \cos(t)), \quad t \in [0, 1], \quad y(0) = 0.$$

Θεωρείστε την άμεση μέθοδο του Euler, την πεπλεγμένη μέθοδο του Euler και τη μέθοδο τραπεζίου. Θεωρήστε ένα διαμερισμό του $[0, 1]$ σε $N + 1$ σημεία. Για $N = 20$, κατασκευάστε τις προσεγγίσεις που δίνουν οι τρεις μέθοδοι, δημιουργείστε τις γραφικές παράστασεις της προσεγγιστικής λύσης και της ακριβούς στο διάστημα $[0, 1]$. Στη συνέχεια βρείτε το σφάλμα $\max_{0 \le n \le N} |y_n - y(t_n)|$ και δημιουργείστε τις γραφικές παράστασεις του σφάλματος $e_n = |y_n - y(t_n)|$. Αυξήστε το $N = 30, 40, 50, 60$, και παρατηρήστε τη συμπεριφορά των τριών μεθόδων.

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt


l=-50
def f(t,y):
    s=l*(y-np.cos(t))
    return s

def y_exact(t): #ακριβής λύση
    s=(-l/(l**2+1))*(np.sin(t)-l*np.cos(t))-(l**2/(l**2+1))*np.exp(l*t)
    return s

def g(t):  #βοηθητική συνάρτηση (μέρος της αρχικής f) για να χρησιμοποιησουμε σε πεπλεγμέν
ες μεθόδους
    s=-l*np.cos(t)
    return s


####Euler
N=30
t=np.linspace(0,1,N+1)
h=t[1]-t[0]

y=np.zeros(N+1)
y[0]=0

for i in range(N):
    y[i+1]=y[i]+h*f(t[i],y[i])

plt.plot(t,y_exact(t),t,y)
plt.show()
plt.plot(t,abs(y_exact(t)-y))
plt.show()


#Πεπλεγμένη Euler
ybe=np.zeros(N+1)
ybe[0]=0

for i in range(N):
    ybe[i+1]=(ybe[i]+h*g(t[i+1]))/(1-l*h)

plt.plot(t,y_exact(t),t,ybe)
plt.show()
plt.plot(t,abs(y_exact(t)-ybe))
plt.show()

## Τραπεζιου
ytr=np.zeros(N+1)
ytr[0]=0

for i in range(N):
    ytr[i+1]=(ytr[i]+(h/2)*f(t[i],ytr[i])+(h/2)*g(t[i+1]))/(1-l*h/2)

plt.plot(t,y_exact(t),t,ytr)
plt.show()
plt.plot(t,abs(y_exact(t)-ytr))
plt.show()
```

Άσκηση 2: Επαναλάβετε την παραπάνω άσκηση για $y(t) = \frac{100}{100^2+1}(\sin(t) + 100\cos(t)) - \frac{100^2}{100^2+1}e^{-100t}$, στο $[0, 1]$ η οποία είναι λύση στο η οποία είναι η λύση για την

$$y'(t) = -100(y(t) - \cos(t)), \quad t \in [0, 1], \quad y(0) = 0,$$

για διαμερισμούς με Αυξήστε το $N = 30, 50, 80, 100$, και παρατηρήστε τη συμπεριφορά των τριών μεθόδων.

Συστήματα Διαφορικών εξισώσεων

Θεωρούμε τώρα το ακόλουθο σύστημα διαφορικών εξισώσεων

$$Y'(t) = F(t, Y(t)),$$

όπου Y είναι μια διανυσματική συνάρτηση, $Y : \mathbb{R} \to \mathbb{R}^m$ και $F : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$. Στην περίπτωση που η F είναι γραμμική ως προς Y, τότε το σύστημα ΔΕ, μπορεί να είναι της μορφής

$$Y'(t) = A(t)Y(t) + G(t),$$

όπου $A(t) \in \mathbb{R}^{m\times m}$ και $G : \mathbb{R} \to \mathbb{R}^m$. Παραδείγματος χάριν

$$x'(t) = -y(t), \quad y'(t) = x(t),$$

μπορούμε να το γράψουμε ως

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}' = A\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \quad \text{με } A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Ένα άλλο παράδειγμα είναι

$$x'(t) = x(t) + 2y(t) + 1, \quad y'(t) = -x(t) + y(t) + t,$$

το οποίο μπορούμε να το γράψουμε ως

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}' = A\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + G(t), \quad \text{με } A = \begin{pmatrix} 1 & 2 \\ -1 & 1 \end{pmatrix}, \quad \text{και } G(t) = \begin{pmatrix} 1 \\ t \end{pmatrix}$$

Στην περίπτωση που η μέθοδος μας είναι πεπλεγμένη (π.χ. Πεπλεγμένη Euler) και η F είναι γραμμική ως προς Y, τότε μπορούμε να υπολογίσουμε ακριβώς τη προσέγγιση στο επόμενο βήμα. Π.χ. για την πεπλεγμένη Euler, αν Y^n είναι η προσέγγιση της $Y(t_n)$ τότε

$$Y^{n+1} = Y^n + h(A(t_{n+1})Y^{n+1} + G(t_{n+1}))$$

δηλαδή

$$(I - hA(t_{n+1}))Y^{n+1} = Y^n + hG(t_{n+1})$$

Οπότε σε κάθε βήμα η Y^{n+1} είναι η λύση ενός γραμμικού συστήματος με πίνακα $I - hA(t_{n+1})$ και δεξί μέλος $Y^n + hG(t_{n+1})$.

Για τη λύση ενός γραμμικού συστήματος μπορούμε να χρησιμοποιήσουμε τη εντολή solve της βιβλιοθήκης numpy.linalg

Άσκηση 3: Θεωρούμε το ΠΑΤ

$$x'(t) = x(t) + 2y(t) + 1, \quad y'(t) = -x(t) + y(t) + t, \ t \in [0,5], x(0) = 2, \ y(0) = -1.$$

Χρησιμοποιήστε την Άμεση Μέθοδο Euler, την Πεπλεγμένη Euler και τη μέθοδο Τραπεζίου για να υπολογίσετε τη προσέγγιση για $t = 5$. Φτιάξτε ένα πίνακα των τιμών προσεγγίσεων για $t = 5$ της παρακάτω μορφής χρησιμοποιήστε ακρίβεια 4 δεκαδικών ψηφίων. Οι ακριβείς τιμές είναι $(x(5), y(5)) = (85.4600, -223.8247)$

N	Άμεση *Euler*	Πεπλεγμένη *Euler*	Τραπεζίου
50			
100			
150			
200			
250			

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
#def f1(t,x,y):
#    return (0.05)*x*(1-0.01*y)
#def f2(t,x,y):
#    return (0.1)*y*(0.005*x-2)
def f1(t,x,y):
    return x+2*y+1
def f2(t,x,y):
    return -x+y+t




N=250
t=np.linspace(0,5,N+1)
h=t[1]-t[0]


#Αμεση Euler
x=np.zeros(N+1)
y=np.zeros(N+1)
#αρχικές τιμες
x[0]=2
y[0]=-1

for i in range(N):
    x[i+1]=x[i]+h*f1(t[i],x[i],y[i])
    y[i+1]=y[i]+h*f2(t[i],x[i],y[i])

plt.plot(x,y)
plt.show()
plt.plot(t,x,t,y)

plt.show()
print(x[N],y[N]) #Τιμές των προσεγγίσεων στο τέλος



### Πεπλεγμένη Euler
xbe=np.zeros(N+1)
ybe=np.zeros(N+1)
#αρχικές τιμες
xbe[0]=2
ybe[0]=-1

B=np.eye(2)-h*np.array([[1,2],[-1,1]])  ### ο πίνακας : I-hA (ειναι ο ίδιος σε καθε β
ημα)

for i in range(N):

    b=np.array([xbe[i],ybe[i]])+h*np.array([1,t[i+1]]) #δεξιό μέλος για να λύσουμε σ
ε κάθε βημα

    Y=np.linalg.solve(B,b) #λύση γραμμικου συστηματος σε καθε βημα

    #### Προσεγγίσεις στο επόμενο σημείο (t_{n+1})
    xbe[i+1]=Y[0]
    ybe[i+1]=Y[1]
```

```python
plt.plot(xbe,ybe)
plt.show()
plt.plot(t,xbe,t,ybe)
plt.show()

print(xbe[N],ybe[N]) #Τιμές των προσεγγίσεων στο τέλος
```

In [ ]:

```python

```