Εργαστήριο 2

Παρατηρήσεις για δημιουργία διαμερισμού

Ένας ομοιόμορφος διαμερισμός του $[a,b]$ σε $N+1$ σημεία t_n, $n=0,1,\ldots,N$ μπορεί να γίνει με την εντολή `linspace` της Numpy

In [1]:
```
import numpy as np
t=np.linspace(2,3,5) # Ομοιόμορφος διαμερισμός σε 5 σημεία (N=4)
print(t)
```

```
[2.   2.25 2.5  2.75 3.  ]
```

Το βήμα h θα είναι η διαφορά δύο σημείων του διαμερισμού t_n, π.χ. $h=t_1-t_0$.

In [2]:
```
h=t[1]-t[0]
print(h)
```

```
0.25
```

In [ ]:
```

```

Παρατήρηση:

Πολλές φορές, δεν χρειάζεται να αποθηκεύσουμε όλες τις τιμές των χρονικών βημάτων. Με την εντολή `linspace` δημιουργούμε ένα διάνυσμα-array. Αν θέλουμε το βήμα h να είναι πολύ μικρό, π.χ. $h=10^{-10}$, χρησιμοποιώντας την linspace δημιουργούμε πολύ μεγάλα arrays της Numpy, π.χ. με 10^{10} στοιχεία. Δεσμεύουμε άσκοπα μεγάλη μνήμη στον υπολογιστή και μπορεί να δημιουργηθεί πρόβλημα στη διαχείρηση της από αυτόν. Για να το αποφύγουμε αυτό μπορούμε να θεωρούμε **μόνο** το χρονικό βήμα $t_n=a+nh$ που θέλουμε να χρησιμοποιήσουμε και **όχι όλα** τα χρονικά βήματα του διαμερισμού.

Ένας τρόπος είναι να υπολογίζουμε το επόμενο χρονικό σημείο t_{n+1} αθροίζοντας το βήμα h στο προηγούμενο βήμα που έχουμε ήδη υπολογίζει, t_n.

In [3]:
```
t0=0
t1=t0+h
print(t0,t1)
```

```
0 0.25
```

Παρατήρηση

Αθροιζοντας μια ή μερικές φορές αριθμούς, στον υπολογιστή, δεν δημιουργει κάποιο σημαντικό σφάλμα. Όμως οι πολλές επανάληψεις της πρόσθεσης μπορεί να δημιουργήσουν σημαντικό πρόβλημα. Έτσι, αν θέλουμε να προσθέσουμε το βήμα h, πολλές φορές δεν παίρνουμε το σωστό αποτέλεσμα. Δείτε το ακόλουθο παράδειγμα. Θέλω να δημιουργήσω τα σημεία του διαστήματος [0,100] με βήμα 0.1. Θα χρειαστώ 1000 επαναλήψεις. Όμως το τελικό σημείο που παίρνω δεν είναι το σωστό.

In [4]:
```
h=0.1
t0=0
for i in range(1000): # Κατασκευάζω το σημείο 100, προσθέτοντας διαδοχικά το 0.1 αρχίζοντας από το σημείο 0
     t1=t0+h
     t0=t1
print(t1)
```

```
99.9999999999986
```

Αντί της πρόσθεσης, μπορούμε να επιλέξουμε τον πολλαπλασιασμό, που δημιουργεί μικρότερο σφάλμα από την πρόσθεση.

In [5]:
```
h=0.1
t0=0
t1=1000*h
print(t1)
```

```
100.0
```

Επομένως για να δημιουργήσω το σημείο t_n ενός διαμερισμού του διαστήματος $[a,b]$, χρησιμοποιώντας μια επαναληπτική διαδικασία, θα το ορίζουμε ως `a+h*n`

Μέθοδος Euler

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 τη μέθοδο του Euler. Έστω ένας ομοιόμορφος διαμερισμός του $[a,b]$, στα σημεία $t_n=a+nh$, $n=0,\ldots,N$, με βήμα $h=\dfrac{b-a}{N}$, υπολογίζουμε τις τιμές y^n που αποτελούν προσεγγίσεις στις τιμές $y(t_n)$, $n=0,\ldots,N$, όπου

$$y^{n+1}=y^n+hf(t_n,y^n),\quad n=0,\ldots,N-1.$$

Άσκηση 1: Θέλουμε να δημιουργήσουμε τη γραφική παράσταση μιας συνάρτησης. Έστω $y(t)=e^{-t}+\sin(t)$, στο $[0,10]$. Δημιουργείστε μια διαμέριση του $[0,10]$ με 51 σημεία, t_n, $n=0,1,\ldots,50$, και χρησιμοποιήστε τη βιβλιοθήκη matplotlib για να σχηματίσετε το γράφημα της $y(t)$.

Μπορείτε να θεωρήσετε πολυ περισσοτερα σημεία για να δημιουργήσετε τη γραφική παράσταση. Αν η συνάρτηση είναι "ομαλή" δεν θα "δείτε" διαφορές στη γραφική παράσταση.

Στην Άσκηση 1, θεωρείστε 1000 σημεία για να σχηματίσετε το γράφημα.

Άσκηση 2: Έστω $y(t)=e^{-t}+\sin(t)$, στο $[0,10]$, η οποία είναι λύση στο

$$y'(t)=-y(t)+\cos(t)+\sin(t),\quad t\in[0,10],\quad y(0)=1.$$

Ορίστε στη Python τη συνάρτηση 2 μεταβλητών $f(t,y)=-y+\cos(t)+\sin(t)$ χρησιμοποιώντας τις συναρτήσεις sin και cos της Numpy

In [6]:
```
def f(t,y):
  s=-y+np.cos(t)+np.sin(t)
  return s
```

1. Για βήμα $h=0.5$ και αρχική τιμή $y^0=1$, υπολογίστε με τη μέθοδο του Euler την προσέγγιση y_{10}. Για ποίο σημείο δημιουργήσαμε προσέγγιση; Δείτε ότι το χρονικό σημείο είναι το $t=0+h*10$, δηλαδή το $t=5$. Δηλαδή η προσέγγιση y^{10} με βήμα $h=0.5$ προσεγγίζει την τιμή $y(5)$ και όχι την $y(10)$.
2. Για $N=50$ θα έχουμε $h=(b-a)/N=10/50=0.2$. Τότε για πιά τιμή της συνάρτησης θα προσεγγίζει η y^{10}; Κατασκευάστε τις προσεγγίσεις y_n, $n=0,\ldots,N$ που δίνει η μέθοδος του Euler και δημιουργείστε τη γραφική παράσταση της προσεγγιστικής λύσης.

Άσκηση 3: Για το παραπάνω πρόβλημα αρχικών τιμών, υπολογίστε το σφάλμα ανάμεσα στην ακριβή λύση και την προσεγγιστική στο σημείο $t=10$, $|y_N-y(10)|$, όταν $N=50,100,200,400$. Θεωρείστε ότι σφάλματα, ανάλογα με τη διαμέριση τα ονομάζουμε err_1,err_2,err_3, και err_4.

Στη συνέχεια υπολογίστε τους παρακάτω 3 λόγους που δίνουν αυτές οι τιμές, err_i/err_{i+1} για $i=1,2,3$. Διαπιστώστε αν ο λόγος err_i/err_{i+1} είναι περίπου ο ίδιος για $i=1,2,3$
</p>

Άσκηση 4: Θεωρούμε και πάλι την προηγούμενη Άσκηση 3. Τώρα δίνουμε διαφορετικές τιμές στα err_1,err_2,err_3, και err_4. Δεν υπολογίζουμε το σφάλμα μόνο στο τελικό βήμα, αλλά θεωρούμε το μέγιστο των σφαλμάτων για κάθε χρονικό βήμα. Έτσι για τη διαμέριση με $N=50$, υπολογίζουμε όλα τα σφάλματα $|y^n-y(t_n)|$, $n=0,\ldots,N$ και θέτουμε τώρα

$$err_1=\max_{0\le n\le N}|y_n-y(t_n)|.$$

Στη συνέχεια επαναλαμβάνουμε και για τις διαμερίσεις με $N=100,200,400$ και θέτουμε ως

$$err_i=\max_{0\le n\le N}|y_n-y(t_n)|,\quad N=50,100,200,400.$$

Διαπιστώστε τωρα αν ο λόγος err_i/err_{i+1} είναι περίπου ο ίδιος για $i=1,2,3$

Πειραματική εκτίμηση της τάξης σύγκλισης

Στη θεωρία γνωρίζουμε ότι για τη μέθοδο Euler το σφάλμα της μεθόδου ικανοποιεί

$$\max_{0\le n\le N}|y^n-y(t_n)|\le Ch^p$$

με $p=1$. Στη συνέχεια θέλουμε να επαληθευσουμε ότι το $p=1$ πειραματικά.
Υπολογίζοντας το σφάλμα

$$err_N=\max_{0\le n\le N}|y^n-y(t_n)|$$

για δύο διαφορετικές διαμερίσεις με $N_1<N_2$, η πειραματική τάξη σύγκλισης ορίζεται ως

$$p=\frac{\ln(\frac{err_{N_2}}{err_{N_1}})}{\ln(\frac{N_1}{N_2})}$$

Άσκηση 5: Θεωρείστε τις διαμερίσεις του $[0,5]$, με $N=10,20,30,\ldots,100$. Υπολογίστε τα σφάλματα err_N για το πρόβλημα αρχικών τιμών που θεωρήσαμε παραπάνω και βρείτε τους λόγους που χρησιμοποιούμε για την πειραματική τάξη σύγκλισης της μεθόδου του Euler. Είναι $p\approx 1$;

In [ ]:
```

```