Euler - 2o Εργαστήριο

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: Έστω $y(t) = e^{-\lambda t} + \sin(2t)$, στο $[0, 5]$ η οποία είναι λύση στο

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

Θεωρείστε ότι $\lambda = 50$. Για $N = 50$, κατασκευάστε τις προσεγγίσεις που δίνει η μέθοδος του Euler, δημιουργείστε τη γραφική παράσταση της προσεγγιστικής λύσης και της ακριβούς στο διάστημα [0,5]. Στη συνέχεια βρείτε το σφάλμα $\max_{0 \le n \le N} |y_n - y(t_n)|$

Πειραματική εκτίμηση της τάξης σύγκλισης. 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 $p = 1$.

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

Άσκηση 3: Επαναλάβετε την προηγούμενη άσκηση αλλά τώρα θεωρείστε τις διαμερίσεις του $[0, 5]$, με $N = 200, 220, 230, \ldots, 300$. Υπολογίστε τα σφάλματα err_N και βρείτε τους λόγους που χρησιμοποιούμε για την πειραματική τάξη σύγκλισης της μεθόδου του Euler. Είναι $p \approx 1$;

Υπόδειξη:

In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt

def y_exact(t):
    l=50
    s=np.exp(-l*t)+np.sin(2*t)
    return s

def f(t,y):
    l=50
    s=-l*y+2*np.cos(2*t)+l*np.sin(2*t)
    return s



N=list(range(200,301,10))
err=np.zeros(len(N))
for j in range(len(N)):
    t=np.linspace(0,5,N[j]+1)
    h=t[1]-t[0]
    y=np.zeros(N[j]+1)
    y[0]=1

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

    err[j]=max(abs(y_exact(t)-y))

for i in range(len(N)-1):
   print(np.log(err[i+1]/err[i])/np.log(N[i]/N[i+1]))
```

```
1.668768771378355
1.6811547325844762
1.6926699803726768
1.7034005937991619
1.7134220706948788
1.7228008284966017
1.73159547403143
1.7398578780805232
1.7476340857975734
1.7549650894402915
```

Συστήματα Διαφορικών εξισώσεων

Θεωρούμε τώρα το ακόλουθο σύστημα διαφορικών εξισώσεων

$$x'(t) = -y(t), \quad y'(t) = x(t), \; t \in [0, 2\pi], x(0) = 1, \; y(0) = 0$$

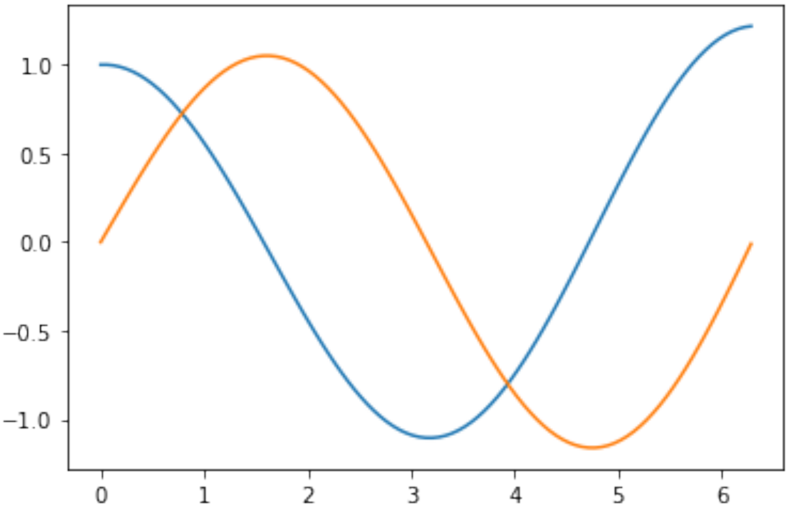
Η ακριβή λύση είναι απλό να δούμε ότι είναι $x(t) = \cos(t)$, $y(t) = \sin(t)$.

Άσκηση 4: Θεωρείστε μια διαμέριση του $[0, 2\pi]$, με $N = 100$ και εφαρμόστε τη μέθοδο του Euler για συστήματα για να βρείτε τις προσεγγίσεις (x_n, y_n) των $(x(t_n), y(t_n))$, $n = 0, \ldots, N$.

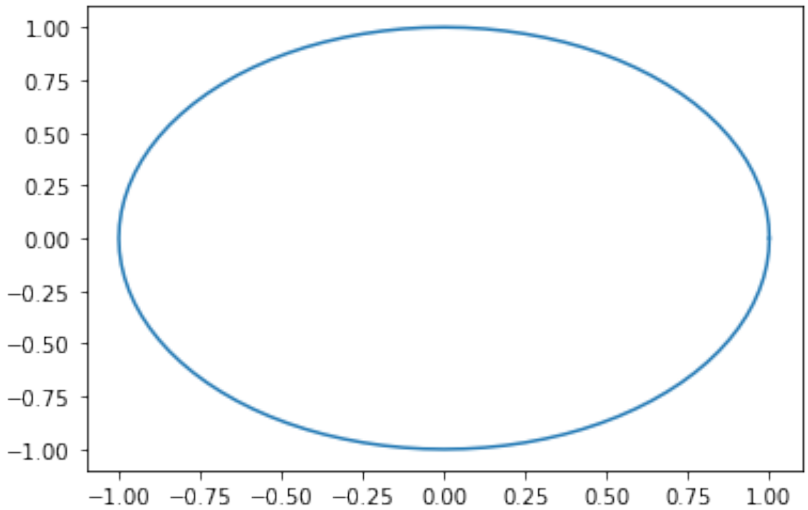
1. Δημιουργείστε τη γραφική παράσταση των λύσεων ως προς το χρόνο t.
2. Επειδή $x(t)^2 + y(t)^2 = 1$, δημιουργείστε τη γραφική παράσταση ανάμεσα στις $(x(t), y(t))$ Παρατηρήστε ότι είναι κύκλος στο πεδίο xy.
3. Βρείτε το $\max_{0 \le n \le N}(x_n^2 + y_n^2)$. Ισχύει $x_n^2 + y_n^2 = 1$; Στη συνέχεια δημιουργείστε τη γραφική παράσταση των 2 προσεγγίσεων (x_n, y_n) στο πεδίο xy. Δημιουργείτε ένας κύκλος;

Υπόδειξη:

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

def f1(t,x,y):
    return -y
def f2(t,x,y):
    return x

#Διαμερισμός
N=100
t=np.linspace(0,2*np.pi,N+1)
h=t[1]-t[0]

# θεσεις για να αποθηκευσω τις προσεγγισεις
x=np.zeros(N+1)
y=np.zeros(N+1)
x[0]=1
y[0]=0

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


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

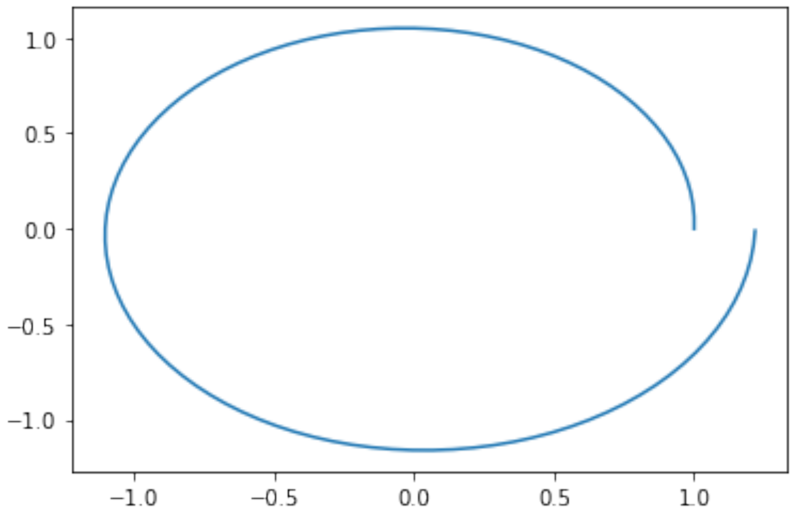


In [3]:
```
import numpy as np
import matplotlib.pyplot as plt

def x_exact(t):
    return np.cos(t)

def y_exact(t):
    return np.sin(t)

#Διαμερισμός
N=100
t=np.linspace(0,2*np.pi,N+1)

plt.plot(x_exact(t),y_exact(t))
plt.show()
```



In [4]:
```
import numpy as np
import matplotlib.pyplot as plt

def f1(t,x,y):
    return -y
def f2(t,x,y):
    return x

#Διαμερισμός
N=100
t=np.linspace(0,2*np.pi,N+1)
h=t[1]-t[0]

# θεσεις για να αποθηκευσω τις προσεγγισεις
x=np.zeros(N+1)
y=np.zeros(N+1)
x[0]=1
y[0]=0

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

## Στην Numpy η παρακάτω πράξη γίνεται σε κάθε στοιχείο των διανυσμ
άτων x,y
s=max(x**2+y**2)
print(s)
plt.plot(x,y)
plt.show()
```

```
1.4829108522377654
```



Μέθοδος Taylor (2)

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

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

Έστω ένας ομοιόμορφος διαμερισμός του $[a, b]$, στα σημεία $t_n = a + nh$, $n = 0, \ldots, N$, με βήμα $h = \dfrac{b-a}{N}$. θεωρήσαμε τη μέθοδο Taylor (2) όπου υπολογίζουμε τις τιμές y_n που αποτελούν προσεγγίσεις στις τιμές $y(t_n)$, $n = 0, \ldots, N$, σύμφωνα με

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} g(t_n, y_n), \quad n = 0, \ldots, N-1,$$

όπου $g(t, y(t)) = \dfrac{d}{dt} f(t, y(t))$.

Άσκηση 5: Έστω $y(t) = e^{-\lambda t} + \sin(2t)$, στο $[0, 5]$ η οποία είναι λύση στο

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

Θεωρείστε ότι $\lambda = 50$.

1. Για $N = 50$, κατασκευάστε τις προσεγγίσεις που δίνει η μέθοδος του Taylor (2), δημιουργείστε τη γραφική παράσταση της προσεγγιστικής λύσης και της ακριβούς στο διάστημα [0,5]. Στη συνέχεια βρείτε το σφάλμα $\max_{0 \le n \le N} |y_n - y(t_n)|$ και συγκρίνεται με το αντίστοιχο σφάλμα για τη μέθοδο του Euler.
2. Θεωρείστε τις διαμερίσεις του $[0, 5]$, με $N = 10, 20, 30, \ldots, 100$. Υπολογίστε τα σφάλματα err_N και βρείτε τους λόγους που χρησιμοποιούμε για την πειραματική τάξη σύγκλισης της μεθόδου του Taylor (2). Είναι $p \approx 2$; Συγκρίνετε τα σφάλματα με τα αντίστοιχα για τη μέθοδο Euler.