5ο 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 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

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

Μη γραμμική f ως προς y

Σε αυτή την περίπτωση για να υπολογίσουμε τις προσεγγίσεις χρησιμοποιώντας μια πεπλεγμένη μέθοδο πρέπει να βρούμε τη λύση μιας μη γραμμικής εξίσωσης σε κάθε βήμα. Μια μέθοδος είναι π.χ. η μέθοδος του σταθερού σημείου.

Μέθοδος σταθερού σημείου

Για να βρούμε τη λύση της εξίσωσης $x^\star = g(x^\star)$, για x_0 δοσμένο δημιουργούμε την ακολουθία x_k

$$x_{k+1} = g(x_k), n = 0, \dots$$

Αν x_k συγκλίνει τότε $x_k \to x^\star$. Επειδή δεν μπορούμε να βρούμε ακριβώς το $x^\star$ αλλά μια προσέγγιση, πρέπει ο αλγόριθμος να σταματά μετά από κάποια βήματα. Ένα κριτήριο τερματισμού του αλγορίθμου είναι να θέσουμε ότι η διαφορά δυο συνεχόμενων προσεγγίσεων x_k να γίνει μικρότερη από έναν προκαθορισμένο αριθμό TOL. Ετσι τερματίζουμε την επαναλαληπτική διαδικασια και δεχομαστε ως προσέγγιση της $x^\star$ το x_k όταν $|x_k - x_{k-1}| \leq TOL$. Επειδή μπορεί να δημιουργηθούν και άλλα σφάλματα και το προηγούμενο κριτήριο να μην ικανοποιείτε ποτέ, θέτουμε και για ασφάλεια ένα μέγιστο αριθμό επαναλήψεων $Nmax$. Δηλαδή δεχόμαστε ως $x^\star$ είτε το x_k που πρώτη φορά ικανοποιείται το κριτήριο τερματισμού ή το x_{Nmax}.

Άσκηση 1 (για την εύρεση ενός σταθερού σημείου): Έστω $g(t) = \cos(t)$, στο $[0,1]$. Θέλουμε να βρούμε το σημείο $x^\star \in [0,1]$ τέτοιο ώστε

$$x^\star = \cos(x^\star).$$

Φτιάξτε μια επαναληπτική διαδικασία ώστε να προσεγγίσετε το $x^\star$, ξεκινώντας από $x_0 = 1$. Για κριτήρια τερματισμού, θέσετε μέγιστο αριθμό επαναλήψεων $Nmax = 500$ και $TOL = 10^{-8}$. Ποίο είναι το προσεγγιστικό σταθερό σημείο και πόσα βήματα κάνατε;

In [ ]:

```python
import numpy as np
import matplotlib.pylab as plt

def g(x):
    s=np.cos(x)
    return s

x0=1 #αρχική προσέγγιση - x0 θα ειναι παντα η προηγουμενη προσεγγιση

tol=1.e-8 # Ακριβεια σφάλματος
Nmax=500 #Μεγιστος αριθμός επαναλήψεων
k=0      # Μετρητής επαναλήψεων

err=1. #Θετουμε αρχικά το σφάλμα ίσον με 1 για να ξεκινήσει η διαδικασία

########
#Σχέδιο προτεινόμενου προγράμματος
#Επανάληψη: Όσο (err>tol) and (k<=Nmax) εκτέλεσε τα παρακάτω:
#            1.Υπολογισε την επόμενη προσεγγιση x (x=g(x0))
#            2.Υπολόγισε την απολυτη τιμη της διαφορας 2 συνεχόμενων προσεγγίσεων, x και x0
#            3.Αυξάνουμε το μετρητη k κατά 1.
#.           4.Θέτουμε ως x0 τη τιμή της x,
#               για να υπολογίσουμε την επόμενη προσεγγιση οταν
#               θα ξανατρέξουν τα βηματα της επαναληψης
```

Πεπλεγμένη Euler

Μια πεπλεγμένη μέθοδος είναι η **πεπλεγμένη Euler**

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

Για να υπολογίσουμε την τιμή y_{n+1} χρειάζεται να λύσουμε (στη γενική περίπτωση που η f είναι μη γραμμική ως προς y) μια μη γραμμική εξίσωση. Ένας τρόπος είναι χρησιμοποιήσουμε μια επαναληπτική διαδικασία όπως στον υπολογισμό του σταθερού σημείου. Έτσι, σε κάθε βήμα του αλγορίθμου της πεπλεγμένης Euler, για τον υπολογισμό δηλαδή του y_{n+1} θέτουμε

$$g(s) = y_n + hf(t_{n+1}, s),$$

και αναζητούμε $s^\star$ τέτοιο ώστε

$$s^\star = g(s^\star).$$

Επειδή μπορούμε μόνο να προσεγγίσουμε το $s^\star$, την προσέγγιση s_k που θα υπολογίσουμε χρησιμοποιώντας την επαναληπτική διαδικασία του σταθερού σημείου, θα θέσουμε ως y_{n+1}, δηλαδή θεωρούμε την ακολουθία

$$s_{k+1} = g(s_k), k = 0, \dots,$$

για δοσμένο tol, υπολογίζουμε το s_k τέτοιο ώστε $|s_k - s_{k-1}| \leq tol$ και θέτουμε

$$y_{n+1} = s_k$$

και προχωράμε για την επόμενη προσέγγιση της μεθόδου πεπεπλεγμένης Euler.

Άσκηση 2: θεωρείστε την πεπλεγμένη μέθοδο του Euler για τον υπολογισμό της λύσης του προβλήματος

$$y'(t) = \frac{1}{1+t^2} - 2(y(t))^2, \quad t \in [0,10], \quad y(0) = 0.$$

Η ακριβή λύση του προβλήματος είναι $y(t) = \dfrac{t}{1+t^2}$. Βρείτε το σφάλμα $\max_{0 \leq n \leq N} |y_n - y(t_n)|$. Για $N = 100, 200, 300, 400, 500$ βρείτε τη πειραματική τάξη σύγκλισης για την πεπλεγμένη μεθόδο του Euler. Θεωρείστε για κάθε βήμα, μέγιστο αριθμό επαναλήψεων $Nmax = 500$ και $TOL = 10^{-8}$

In [ ]:

```python
import numpy as np
import matplotlib.pyplot as plt
def y_exact(t):
    s=t/(1+t**2)

    return s

def f(t,y):
    s=1/(1+t**2)-2*y**2
    return s

def g(t,yn,x):
    s=yn+h*f(t,x)
    return s



#### Πεπλεγμένη Euler (Μη γραμμική f ως προς y)
```

Μέθοδος Τραπεζίου

Μια άλλη πεπλεγμένη μέθοδος είναι η **Μέθοδος Τραπεζίου**

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

Για να υπολογίσουμε την τιμή y_{n+1} χρειάζεται να λύσουμε (στη γενική περίπτωση που η f είναι μη γραμμική ως προς y) μια μη γραμμική εξίσωση. Ένας τρόπος είναι χρησιμοποιήσουμε μια επαναληπτική διαδικασία όπως στον υπολογισμό του σταθερού σημείου. Έτσι, σε κάθε βήμα του αλγορίθμου της πεπλεγμένης Euler, για τον υπολογισμό δηλαδή του y_{n+1} θέτουμε

$$g(s) = y_n + (h/2)(f(t_n, y_n) + f(t_{n+1}, s)),$$

και αναζητούμε $s^\star$ τέτοιο ώστε

$$s^\star = g(s^\star).$$

Επειδή μπορούμε μόνο να προσεγγίσουμε το $s^\star$, την προσέγγιση s_k που θα υπολογίσουμε χρησιμοποιώντας την επαναληπτική διαδικασία του σταθερού σημείου, θα θέσουμε ως y_{n+1}, δηλαδή θεωρούμε την ακολουθία

$$s_{k+1} = g(s_k), k = 0, \dots,$$

για δοσμένο tol, υπολογίζουμε το s_k τέτοιο ώστε $|s_k - s_{k-1}| \leq tol$ και θέτουμε

$$y_{n+1} = s_k$$

και προχωράμε για την επόμενη προσέγγιση της μεθόδου πεπεπλεγμένης Euler.

Άσκηση 4: Επαναλάβετε το πρόβλημα για τη μέθοδο του τραπεζίου.

In [ ]:

```python
#### Τραπεζιου (Μη γραμμική f ως προς y)
# Η αλλαγή σε σχεση με την προηγούμενη άσκηση είναι ο ορισμός της συνάρτησης που δίνει τη μέθοδο.
# Ο τύπος της μεθόδου του τραπεζίου είναι
#y_{n+1}=y_n+ (h/2)(f(t_{n+1},y_{n+1})+f(t_{n},y_{n})),


def g(t,yn,x):
    s=yn+(h/2)*(f(t-h,yn)+f(t,x))
    return s
```

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 $x(t) = \cos(t)$, $y(t) = \sin(t)$.

Το παραπάνω πρόβλημα μπορεί να γραφεί ως σύστημα Δ.Ε.

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}' = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix},\ t \in [0, 2\pi],\ \mu\varepsilon \begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Λόγω της μορφής του συστήματος διαφορικών εξίσωσεων, μπορούμε να γράψουμε τη πεπλεγμένη μέθοδο του Euler ως εξής:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + h \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix}, \quad n = 0, \dots, N-1, \quad \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Μπορούμε να θεωρήσουμε ως A τον πίνακα

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Οπότε μπορούμε να γράψουμε τη μέθοδο ισοδύναμα ως εξής: Ορίζουμε $Y_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$, $n = 0, \dots, N$, οπότε

$$(I - hA)Y_{n+1} = Y_n, \quad n = 0, \dots, N-1, \quad Y_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Για να βρούμε τη προσέγγιση αρκεί να λύνουμε κάθε φορά ένα γραμμικό σύστημα με τον ίδιο πίνακα $B = (I + hA)$ και με διαφορετικό δεξιό μέλος.

Υπενθύμιση: Για την επίλυση γραμμικών συστημάτων χρησιμοποιούμε τη συνάρτηση `solve` που βρίσκεται στη βιβλιοθήκη `linalg` της `numpy` Ας υποθέσουμε ότι A και b είναι ο πίνακας και το διάνυσμα που δίνονται από

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 9 \\ 8 \end{pmatrix}$$

Η λύση x του γραμμικού συστήματος $Ax = b$, προκύπτει ως εξής:

In [ ]:

```python
import numpy as np
A=np.array([[3,1],[1,2]])
b=np.array([9,8])
# Επίλυση
x=np.linalg.solve(A,b)
print('x=',x)
print('Επαλήθευση')
print('b=',np.dot(A,x))
```

Άσκηση 5: Θεωρείστε μια διαμέριση του $[0, 2\pi]$, με $N = 100$ και εφαρμόστε τη πεπλεγμένη μέθοδο του Euler και τη μέθοδο τραπεζίου για συστήματα για να βρείτε τις προσεγγίσεις (x_n, y_n) των $(x(t_n), y(t_n))$, $n = 0, \dots, N$.

1. Δημιουργείστε τη γραφική παράσταση των λύσεων ως προς το χρόνο t.
2. Βρείτε το $x_n^2 + y_n^2$. 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;
3. Δοκιμάστε να βρείτε τις προσεγγίσεις (x_n, y_n) των $(x(t_n), y(t_n))$, $n = 0, \dots, N$, με $N = 200, 400, 800$. Στη συνέχεια δημιουργείστε τη γραφική παράσταση των 2 προσεγγίσεων (x_n, y_n) στο πεδίο xy. Τι παρατηρείται; Επίσης επιβεβαιώστε την τάξη σύγκλισης των μεθόδων

In [ ]:

```python
def x_exact(t):
    s=np.cos(t)
    return s

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



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

Y=np.zeros((N+1,2)) # Δημιουργούμε με μηδεν το διάνυσμα της λύσης
Y[0,0]=1 # Αρχική συνθήκη (1-συνιστώσα)
Y[0,1]=0 # Αρχική συνθήκη (2-συνιστώσα)
A=np.array([[0,-1],[1,0]])
B=np.eye(2)-h*A # Πίνακας λυσης γραμμικού συστηματος BY_{n+1}=Y_n




plt.plot(Y[:,0],Y[:,1]) #Ζωγραφίζω στο xy-επιπεδο
plt.show()

err_Y=np.zeros(5)  #array για τα σφάλματα
N=[100,200,300,400,500]  #διαμερίσεις για να βρω ταξη σύγκλισης
```

Runge - Kutta

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

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

μπορούμε να θεωρήσουμε και τις μεθόδους Runge - Kutta. Αυτές οι μέθοδοι είναι μονοβηματικές περιγράφονται από ένα μητρώο Butcher q σταδίων, που δίνεται ως

$$\begin{array}{ccc|c} a_{11} & \dots & a_{1q} & \tau_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{q1} & \dots & a_{qq} & \tau_q \\ \hline b_1 & \dots & b_q & \end{array}$$

και χρησιμοποιούν προσεγγίσεις της ακριβούς λύσης σε σημεία ενδιάμεσα των t_n και t_{n+1}, $t_{n,i} = t_n + \tau_i h$, $i = 1, \dots, q$, ως εξής: Κατασκευάζουμε τις τιμές $y_{n,i}$, $i = 1, \dots q$, σύμφωνα με

$$y_{n,i} = y_n + h \sum_{j=1}^{q} a_{ij} f(t_{n,j}, y_{n,j}), \quad i = 1, \dots, q,$$

και

$$y_{n+1} = y_n + h \sum_{j=1}^{q} b_i f(t_{n,i}, y_{n,i})$$

Αν ισχύει ότι $a_{ij} = 0$, $i \leq j$ τότε έχουμε μια άμεση μέθοδο Runge-Kutta γιατί ο προσδιορισμός των τιμών $y_{n,i}$, $i = 1, \dots q$ γίνεται με άμεσο τρόπο, χωρίς τη λύση κάποιου συστήματος.

Αν η συνάρτηση f είναι μη γραμμική ως προς τη δεύτερη μεταβλητή της, y, τότε για να προσδιορίσουμε τις τιμές $y_{n,i}$, $i = 1, \dots, q$ χρειάζεται να λύσουμε ένα μη γραμμικό σύστημα q εξισώσεων με q αγνώστους. Αν $a_{ij} = 0$, $i < j$ και $a_{ii} \neq 0$, $i = 1, \dots, q$ τότε έχουμε μια ημιπεπλεγμένη μέθοδο Runge-Kutta. Αν $a_{ij} \neq 0$, $i < j\ i = 1, \dots, q$ τότε έχουμε μια πλήρως πεπλεγμένη μέθοδο Runge-Kutta.

Παραδείγματα άμεσων μεθόδων Runge-Kutta

1. Runge-Kutta 1-σταδίου

$$\begin{array}{c|c} 0 & 0 \\ \hline 1 & \end{array}$$

2. Runge-Kutta 2-σταδίων

$$\begin{array}{cc|c} 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \hline \frac{1}{2} & \frac{1}{2} & \end{array} \qquad \begin{array}{cc|c} 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \hline 0 & 1 & \end{array}$$

3. Runge-Kutta 3-σταδίων

$$\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ \dfrac{1}{3} & 0 & 0 & \dfrac{1}{3} \\ 0 & \dfrac{2}{3} & 0 & \dfrac{2}{3} \\ \hline \dfrac{1}{4} & 0 & \dfrac{3}{4} & \end{array}$$

Για να υλοποιήσουμε τις άμεσες Runge-Kutta μπορούμε να ακολουθήσουμε την ακόλουθη διαδικασία: Αν έχουμε μια άμεση Runge-Kutta q-σταδίων θεωρούμε τις βοηθητικές μεταβλητές $k_1, \dots, k_q$ οι οποίες ορίζονται ως εξής

$$k_i = f(t_{n,i}, y_{n,i}), \quad i = 1, \dots, q$$

Οπότε οι προσεγγίσεις $y_{n,i}$ στα ενδιάμεσα σημεία $t_{n,i}$ μπορούν να υπολογιστούν ως

$$y_{n,1} = y_n, \quad k_1 = f(t_{n,1}, y_{n,1}) \tag{1}$$
$$y_{n,2} = y_n + a_{21}hk_1, \quad k_2 = f(t_{n,2}, y_{n,2}) \tag{2}$$
$$y_{n,3} = y_n + a_{31}hk_1 + a_{32}hk_2, \quad k_3 = f(t_{n,2}, y_{n,2}) \tag{3}$$
$$\vdots \tag{4}$$
$$y_{n,q} = y_n + h \sum_{j=1}^{q-1} a_{qj} k_j, \quad k_q = f(t_{n,q}, y_{n,q}) \tag{5}$$
$$y_{n+1} = y_n + h \sum_{i=1}^{q} b_i k_i \tag{6}$$

στα ενδιάμεσα σημεία

$$y_{n,i} = y_n + h \sum_{j=1}^{q} a_{ij} f(t_{n,j}, y_{n,j}), \quad i = 1, \dots, q,$$

Άσκηση 6: θεωρείστε το πρόβλημα αρχικών τιμών

$$y'(t) = \frac{1}{1+t^2} - 2(y(t))^2, \quad t \in [0,10], \quad y(0) = 0.$$

Η ακριβή λύση του προβλήματος είναι $y(t) = \dfrac{t}{1+t^2}$. Θεωρείστε τη μέθοδο Runge-Kutta 2-σταδίων

$$\begin{array}{cc|c} 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \hline \frac{1}{2} & \frac{1}{2} & \end{array}$$

Θεωρήστε ένα διαμερισμό του $[0,10]$ σε $N+1$ σημεία, και υλοποιήστε αυτη τη μεθόδο για N=20. Βρείτε το σφάλμα $\max_{0 \leq n \leq N} |y_n - y(t_n)|$ και συγκρίνετε με το αντίστοιχο σφάλμα αν χρησιμοποιήσατε την αμεση μεθοδο του Euler.

Βρείτε το σφάλμα $\max_{0 \leq n \leq N} |y_n - y(t_n)|$, για $N = 20, 40, 80, 160$ καθώς και την προσεγγιστική τάξη σύγκλισης p.

Παρατήρηση: Η μέθοδος είναι μια άμεση μονοβηματική μέθοδος 2 σταδίων και είναι τάξη ακρίβειας είναι 1. Όσο και η άμεση Euler. </p>

In [ ]:

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

def y_exact(t):
    s=t/(1+t**2)
    return s

def f(t,y):
    s=1/(1+t**2)-2*y**2
    return s

N=20
y=np.zeros(N+1)
y0=0

## αμεση Euler (για συγκριση αποτελεσματων)

t=np.linspace(0,10,N+1)
h=t[1]-t[0]
y[0]=y0

# Υπολογισμός των προσεγγισεων και εκτυπωση αποτελεσμάτων



## Μια αμεση Runge-Kutta 2 σταδιων

N=20
y=np.zeros(N+1)
t=np.linspace(0,10,N+1)
h=t[1]-t[0]
y[0]=y0

a=1./2
t2=a # το βήμα για το 2ο ενδιαμεσο σημειο

for i in range(N):


     ### Υπολογισμός 1ου ενδιαμεσου βηματος και της αντίστοιχης τιμης της f



    ### Υπολογισμός 2ου ενδιαμεσου βηματος και της αντίστοιχης τιμης της f


     #### Υπολογισμος της επομενης προσεγγισης (χρησιμοποιουμε τις ενδιαμεσες προσεγγισεις που εχουμε βρει)


# Εκτυπωση αποτελεσμάτων



####Υπολογισμος ταξης ακριβειας
```

Άσκηση 7: θεωρείστε το πρόβλημα αρχικών τιμών

$$y'(t) = \frac{1}{1+t^2} - 2(y(t))^2, \quad t \in [0,10], \quad y(0) = 0.$$

Η ακριβή λύση του προβλήματος είναι $y(t) = \dfrac{t}{1+t^2}$. Θεωρείστε τη μέθοδο Runge-Kutta 2-σταδίων

$$\begin{array}{cc|c} 0 & 0 & 0 \\ \dfrac{1}{2} & 0 & \dfrac{1}{2} \\ \hline 0 & 1 & \end{array}$$

Θεωρήστε ένα διαμερισμό του $[0,10]$ σε $N+1$ σημεία, και υλοποιήστε αυτη τη μεθόδο για N=20. Βρείτε το σφάλμα $\max_{0 \leq n \leq N} |y_n - y(t_n)|$ και συγκρίνετε με το αντίστοιχο σφάλμα αν χρησιμοποιήσατε την αμεση μεθοδο του Euler και την μεθοδο Runge-Kutta 2-σταδίων της προηγούμενης άσκησης

Βρείτε το σφάλμα $\max_{0 \leq n \leq N} |y_n - y(t_n)|$, για $N = 20, 40, 80, 160$ καθώς και την προσεγγιστική τάξη σύγκλισης p.

Παρατήρηση: Η μέθοδος είναι μια άμεση μονοβηματική μέθοδος 2 σταδίων και είναι τάξη ακρίβειας είναι 2. Η απλή μονοβηματικη μέθοδος Euler έχει τάξη κατά ένα λιγότερο. </p>

In [ ]: