
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΦΥΣ 140 Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών
Χειμερινό Εξάμηνο 2023

Φώτης Πτωχός και Αλέξανδρος Αττίκης
Φροντιστήριο 7

23 Οκτωβρίου 2023
15:00 - 17:00



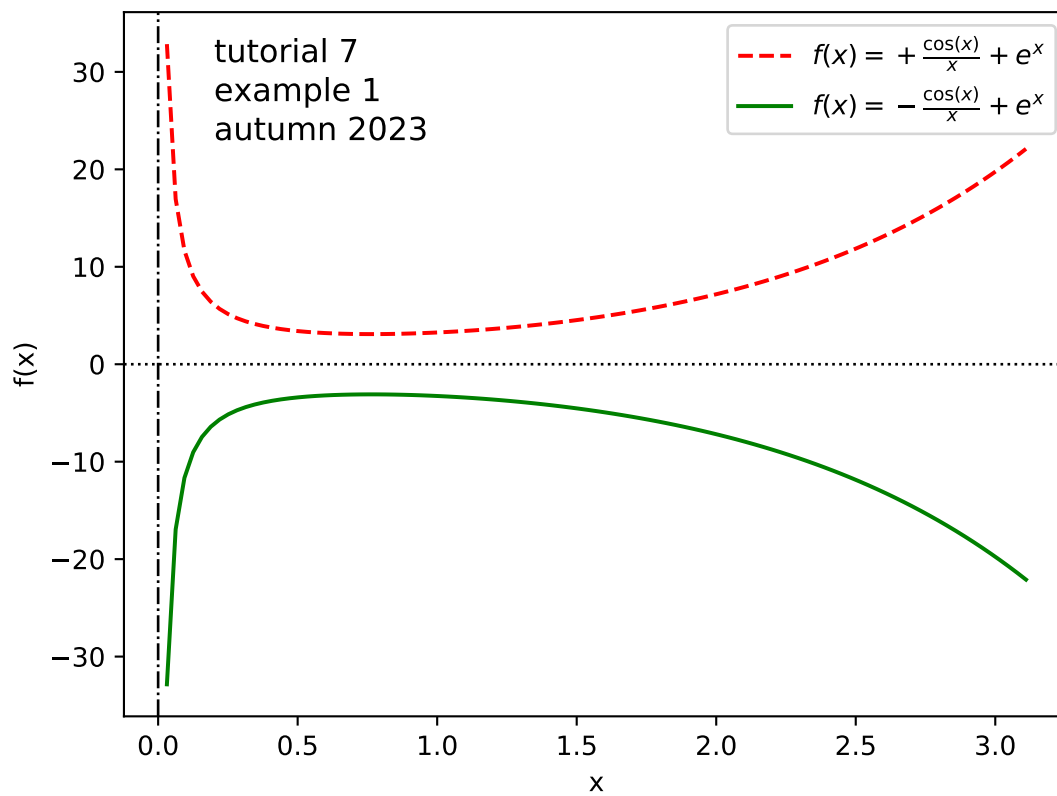
Φροντιστήριο 7

Παράδειγμα 1 Απλό πρόγραμμα για τη χρήση της εντολής *plot* και σχετιζόμενες επιλογές:

tutorial7/ex1.py

```
1  #!/usr/bin/python3
2  '''
3  DESCRIPTION:
4  Example usage of matplotlib pyplot library
5
6  LINKS:
7  https://matplotlib.org/stable/api/pyplot_summary.html
8  '''
9  import numpy as np
10 import matplotlib.pyplot as plt
11 import math
12
13 def myFunc(x):
14     return (np.cos(x)/x) + math.exp(x)
15
16 # Get a range of values over pi: numpy.arange(start, stop, step, dtype=None)
17 x = np.arange(np.pi/100, np.pi, np.pi/100)
18 y = list( map(myFunc, x) )
19
20 # Create the canvas & draw the plot on the canvas
21 plt.figure()
22 plt.plot(x, y, "r--", label=r'$f(x)=\frac{\cos(x)}{x} + e^x$')
23 plt.plot(x, [-a for a in y], "g-", label=r'$f(x)=-\frac{\cos(x)}{x} + e^x$')
24
25 # Add axes labels
26 plt.xlabel("x")
27 plt.ylabel("f(x)")
28
29 # Add horizontal (h) and vertical (v) lines
30 plt.axhline(0, color='black', lw=1, ls=":")
31 plt.axvline(0, color='black', lw=1, ls="-.")
32
33 # Add additional text
34 yText = ["tutorial 7", "example 1", "autumn 2023"]
35 yPos  = [31, 27, 23]
36 for i in range( len(yPos) ):
37     plt.text(+0.2, yPos[i], yText[i], fontsize=12, color="black")
38
39 # Add a legend to display the labels!
40 plt.legend()
41
42 # Save the current plot to a file in a specified format
43 for ext in [".png", ".pdf"]:
44     plt.savefig("ex1" + ext)
45
46 # Display the plot in an interactive window (press 'q' to exit/quit)
47 plt.show()
```

Αποτέλεσμα:

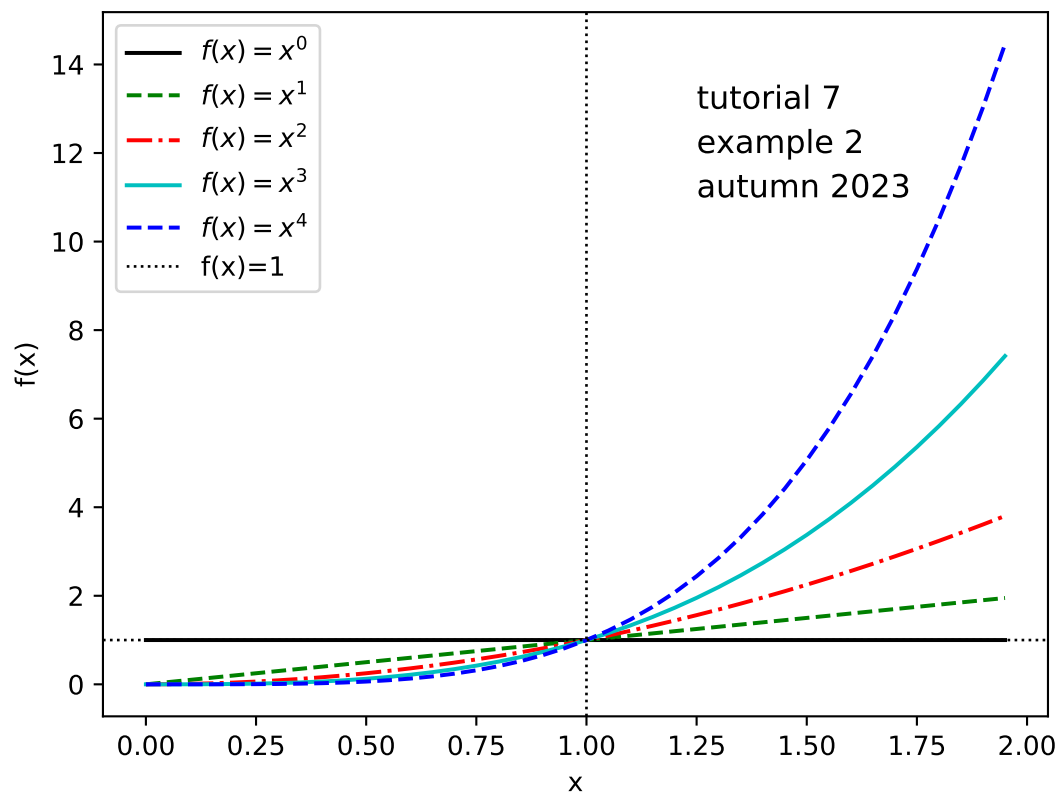


Παράδειγμα 2 Παράδειγμα προγράμματος για χάραξη πολλαπλών καμπυλών στο ίδιο γράφημα:

tutorial7/ex2.py

```
1  #!/usr/bin/python3
2  '''
3  DESCRIPTION:
4  Example usage of matplotlib pyplot library
5
6  LINKS:
7  https://matplotlib.org/stable/api/pyplot_summary.html
8  '''
9  import numpy as np
10 import matplotlib.pyplot as plt
11 import math
12
13 # Get a range of values over pi: numpy.arange(start, stop, step, dtype=None)
14 xList = np.arange(0, 2, 1/20)
15 cList = ["k-", "g--", "r-.", "c-", "b--"]
16 fList = []
17
18 for i in range(0, 5, 1):
19     fList.append( [x**i for x in xList] )
20
21 # Create the canvas & draw the plot on the canvas
22 plt.figure()
23 for i in range( len(fList) ):
24     plt.plot(xList, fList[i], cList[i], label=r'$f(x)=x^{%d}$' % (i))
25
26 # Add axes labels
27 plt.xlabel("x")
28 plt.ylabel("f(x)")
29
30 # Add horizontal (h) and vertical (v) lines
31 plt.axhline(1, color='black', lw=1, ls=":")
32 plt.axvline(1, color='black', lw=1, ls=":", label="f(x)=1")
33
34 # Add additional text
35 yText = ["tutorial 7", "example 2", "autumn 2023"]
36 yPos  = [13, 12, 11]
37 for i in range( len(yPos) ):
38     plt.text(+1.25, yPos[i], yText[i], fontsize=12, color="black")
39
40 # Add a legend to display the labels!
41 plt.legend()
42
43 # Save the current plot to a file in a specified format
44 for ext in [".png", ".pdf"]:
45     plt.savefig("ex2" + ext)
46
47 # Display the plot in an interactive window (press 'q' to exit/quit)
48 plt.show()
```

Αποτέλεσμα:



Παράδειγμα 3 Παράδειγμα προγράμματος για πολλαπλά γραφήματα στον ίδιο κανβά από ανάγνωση δεδομένων από αρχείο “expDecay.dat”, τα περιεχόμενα του οποίου είναι:

"expDecay.dat"

1.000	0.017	1005.930	30.178
1.100	0.017	978.742	29.362
1.200	0.017	952.290	28.569
1.300	0.017	926.552	27.797
1.400	0.017	901.510	27.045
1.500	0.017	877.145	26.314
1.600	0.017	853.439	25.603
1.700	0.017	830.373	24.911
1.800	0.017	807.930	24.238
1.900	0.017	786.094	23.583
2.000	0.017	764.848	22.945
2.100	0.017	744.177	22.325
2.200	0.017	724.064	21.722
2.300	0.017	704.495	21.135
2.400	0.017	685.454	20.564
2.500	0.017	666.928	20.008
2.600	0.017	648.903	19.467
2.700	0.017	631.365	18.941
2.800	0.017	614.302	18.429
2.900	0.017	597.699	17.931
3.000	0.017	581.545	17.446
3.100	0.017	565.827	16.975
3.200	0.017	550.535	16.516
3.300	0.017	535.655	16.070
3.400	0.017	521.178	15.635
3.500	0.017	507.092	15.213
3.600	0.017	493.387	14.802
3.700	0.017	480.052	14.402
3.800	0.017	467.078	14.012
3.900	0.017	454.454	13.634
4.000	0.017	442.172	13.265
4.100	0.017	430.221	12.907
4.200	0.017	418.593	12.558
4.300	0.017	407.280	12.218
4.400	0.017	396.273	11.888
4.500	0.017	385.563	11.567
4.600	0.017	375.142	11.254
4.700	0.017	365.003	10.950
4.800	0.017	355.138	10.654
4.900	0.017	345.540	10.366

Στον κανβά (a) έχουμε ένα απλό γράφημα, στον κανβά (b) βλέπουμε το ίδιο γράφημα σε ημιλογαριθμικό χαρτί, και στον κανβά (c) βλέπουμε εισαγωγή σφαλμάτων και χάραξη της ευθείας που περιγράφεται με την κατάλληλη συνάρτηση:

tutorial7/ex3.py

```
1 #!/usr/bin/python3
2 '''
```

```

3  NOTES:
4      log(y) = log(A) + m*log(x)
5  => log(y) - log(A) = m*log(x) = log(x)^(m)
6  => log(y/A) = log(x^(m))
7  => y/A = x^klisi
8  => y = A * x^klisi
9  '''
10 import math
11 import numpy as np
12 import matplotlib.pyplot as plt
13
14 # Variable definition
15 inputFile= open('expDecay.dat', 'r')
16 X, eX, Y, eY = [], [], [], []
17
18 # Read the values from the file
19 for line in inputFile:
20     try:
21         # split string (file line) into 4 parts and remove empty space
22         x, ex, y, ey = line.strip().split()
23         X +=[float(x)]
24         eX +=[float(ex)]
25         Y +=[float(y)]
26         eY +=[float(ey)]
27     except:
28         print("Problem with line:", line)
29         continue
30 inputFile.close()
31
32 # Variables that will be used for all plot
33 xLabel = "t (s)"
34 yLabel = "Activity (Bq)" # 1 Bq = 1 radioactive decay per second
35 xMin, xMax = 0.0, 5.0
36 yMin, yMax = 5e2, 2e3
37
38 # Create canvas with dimensions 14 x 7 inches
39 plt.figure(figsize=(14,7))
40
41
42 # Create a 1x3 grid of subplots & work on leftmost subplot.
43 plt.subplot(1,3,1)
44 plt.xlabel(xLabel) # etiketa aksona x
45 plt.ylabel(yLabel) # etiketa aksona y
46 plt.plot(X, Y, 'bo')
47 plt.ylim(0.0, yMax) # oria aksona y
48 plt.grid(True)
49 plt.title("(a)", size=12)
50
51
52 # Working on the middle subplot
53 plt.subplot(1,3,2)
54 plt.errorbar(X, Y, yerr=eY, xerr=eX, fmt="bo")
55 plt.semilogy()

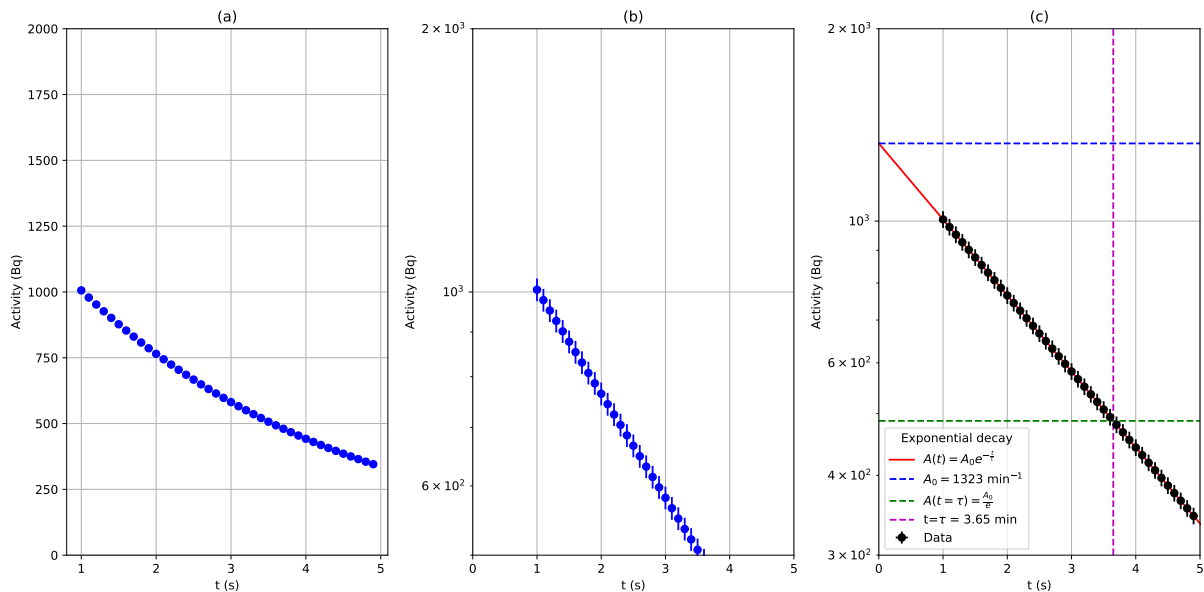
```

```

56 plt.xlabel(xLabel)
57 plt.ylabel(yLabel)
58 plt.xlim(xMin, xMax)
59 plt.ylim(yMin, yMax)
60 plt.grid(True)
61 plt.title("(b)", size=12)
62
63
64 # Working on the rightmost subplot
65 plt.subplot(1,3,3)
66 plt.errorbar(X, Y, yerr=eY, xerr=eX, fmt='ko', label="Data")
67 plt.semilogy()
68 plt.xlabel(xLabel)
69 plt.ylabel(yLabel)
70 plt.xlim(xMin, xMax)
71 plt.ylim(3e2, yMax)
72 plt.grid(True)
73 plt.title("(c)", size=12)
74
75
76 # Determine the parameters (m, c) of the line y = mx + c:"
77 '''
78     A = A0 exp(-t/tau) = A0 * exp(lambda*t)
79     lnA = lnA0 + ln( exp(lambda*t) ) = lnA0 + lambda*t
80 => lnA0 = lnA - lambda*t
81     A0 = exp(lnA - lambda*t)
82
83 Substitute known values for A (i.e. y) and t (i.e. x) to evaluate A0 which is
    the y-intercept
84 '''
85 slope = ( np.log(Y[-1]) - np.log(Y[0]) ) / (X[-1] - X[0]) # activity in number
    of radioisotopes per min
86 tau    = -1/slope # decay constant in min (time taken to reduce activity A(t) to
    A0/e)
87 A0     = np.exp( np.log(Y[-1]) - slope * X[-1])
88 print("== tau = %.2f min , A0 = %.2f / min" % (tau, A0) )
89
90 newX = [0.01*i for i in range(0, int(20)*100, 1)]
91 newY = [A0 * math.exp(-t/tau) for t in newX]
92 plt.plot(newX, newY, 'r-', label=r"$A(t) = A_{0}e^{-\frac{t}{\tau}}$")
93
94 plt.axhline(y=A0, color='b', linestyle='--', label=r"$A_{0} = %0.0f$ min$^{-1}$"
    " % (A0) )
95 plt.axhline(y=A0/np.exp(1.0), color='g', linestyle='--', label=r"$A(t=\tau)=\frac{A_{0}}{e}$")
96 plt.axvline(x=tau, color='m', linestyle='--', label=r"$t=\tau$ = %.2f min" % (
    tau) )
97 plt.tight_layout() # Correct padding around the figures
98 plt.legend(title="Exponential decay")
99 for ext in [".png", ".pdf"]:
100     plt.savefig("ex3" + ext)
101 plt.show()

```


Αποτέλεσμα :



Στη συνάρτηση `plt.errorbar()` της βιβλιοθήκης *Matplotlib*, η παράμετρος *fmt* χρησιμοποιείται για τον καθορισμό της μορφής των σημείων δεδομένων και των γραμμών σφάλματος στο διάγραμμα σας. Αυτή η συμβολοσειρά μορφής χρησιμοποιείται για τον καθορισμό της εμφάνισης των σημείων δεδομένων, συμπεριλαμβανομένου του στυλ δείκτη, του χρώματος και του στυλ γραμμής τόσο για τις γραμμές δεδομένων όσο και για τις γραμμές σφαλμάτων.

Η συμβολοσειρά *fmt* αποτελείται συνήθως από έναν ή περισσότερους χαρακτήρες που καθορίζουν το στυλ των σημείων δεδομένων και των γραμμών σφάλματος. Ακολουθούν ορισμένα κοινά στοιχεία που μπορείτε να χρησιμοποιήσετε στη συμβολοσειρά *fmt*:

Διαθέσιμες επιλογές για στυλ σημείων:

‘o’ Κυκλικοί δείκτες

‘s’ Τετράγωνοι δείκτες

‘D’ Διαμαντένιοι δείκτες

‘^’ δείκτες τριγώνου που δείχνουν προς τα πάνω

‘v’ δείκτες τριγώνου με κατεύθυνση προς τα κάτω

‘p’ δείκτες του Πενταγώνου

‘+’ δείκτες συν

‘x’ Σταυροί δείκτες

Διαθέσιμες επιλογές για στυλ γραμμής:

‘—’ Συμπαγής γραμμή

‘:’ Διακεκομμένη γραμμή με τελείες

‘— —’ Διακεκομμένη γραμμή με γραμμές

‘—.’ Γραμμή με τελεία

Διαθέσιμες επιλογές για χρώματα:

‘b’ Μπλε

‘g’ Πράσινο

‘r’ Κόκκινο

‘c’ Κυανό

‘m’ Ματζέντα

‘y’ Κίτρινο

‘k’ Μαύρο

Μπορείτε να συνδυάσετε αυτά τα στοιχεία μέσα στη συμβολοσειρά *fmt* για να καθορίσετε το στυλ που θέλετε. Για παράδειγμα, το ‘bo—’ καθορίζει μπλε κύκλους με μια συμπαγή γραμμή που τους συνδέει, ενώ το ‘rs:’ καθορίζει τα κόκκινα τετράγωνα με μια διακεκομμένη γραμμή. Η σειρά των στοιχείων μέσα στη συμβολοσειρά *fmt* έχει σημασία, με το στυλ δείκτη πρώτα, ακολουθούμενο από το στυλ γραμμής (αν υπάρχει) και μετά το χρώμα.

Ακολουθούν μερικά παραδείγματα συμβολοσειρών *fmt*:

- ‘ro’ για κόκκινα κυκλικά σημεία (χωρίς γραμμές σύνδεσης).
- ‘g—’ για πράσινη συμπαγή γραμμή χωρίς σημεία.
- ‘bs—’ για μπλε τετράγωνα σημεία με μια διακεκομμένη γραμμή που τα συνδέει.