Ασκήσεων επίλυσης απλών προβλημάτων με Κώδικα Python κεφαλαίου 5.2, 5.3

Βοήθημα, Online βιβλίο Μαθηματικών Γ Λυκείου θετικής κατεύθυνσης:

https://ebooks.edu.gr/ebooks/v/html/8547/2732/Mathimatika_Teuxos-B_G-Lykeiou-ThSp_html-apli/

Κεφάλαιο 5.2, 5.3 :http://repfiles.kallipos.gr/html_books/9863/Ch5.html

Περιεχόμενα

Άσκηση 1: Για μαλθουσιανό μοντέλο $N(t)=100e^{0.2t}$ Να βρεθεί ο ρυθμός μεταβολής στο t=5.2

Άσκηση 2. Κατά κεφαλή ρυθμός ανάπτυξης για $N(t)=50e^{0.4t}$ υπολογίστε r=$\mathbf{1}\frac{dN}{Ndt}$ 2

Άσκηση 3 Σημεία ισορροπίας λογιστικού μοντέλου $\mathbf{F(N) = 0.5N1 - N100}$Να βρεθούν τα σημεία ισορροπίας. 3

Άσκηση 4 3

Άσκηση 5 Μέγιστος ρυθμός ανάπτυξης.....3

Άσκηση 6. Μέγιστη τιμή ανάπτυξης.....4

Άσκηση 7 Gompertz.....4

Άσκηση 8 4

Άσκηση 9 Διακριτό λογιστικό μοντέλο 5

Άσκηση 10 Μέγιστη βιολογική παραγωγικότητα 5

Απλή γραφική παράσταση Συνάρτησης :

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0,40,500)          # ορισμός των σημείων στον x άξονα
y = 10*x/(1+x)                     # ορισμός συνάρτησης
plt.plot(x,y)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

Άσκηση 1: Για μαλθουσιανό μοντέλο $N(t)=100e^{0.2t}$ Να βρεθεί ο ρυθμός μεταβολής στο t=5.

```
#NumPy
Python
import numpy as np
t = 5
dNdt = 100*0.2*np.exp(0.2*t)
print(dNdt)

#SymPy
Python
from sympy import *
t = symbols('t')
N = 100*exp(0.2*t)
dNdt = diff(N,t)
print(simplify(dNdt))
print(N.diff(t).subs(t,5))
```

Άσκηση 2. Κατά κεφαλή ρυθμός ανάπτυξης για $N(t)=50e^{0.4t}$ υπολογίστε $r=\frac{1}{N}\frac{dN}{dt}$.

```
#NumPy
Python
import numpy as np
t = 3
N = 50*np.exp(0.4*t)
dNdt = 0.4*N
r = dNdt/N
print(r)

#SymPy
Python
from sympy import *
t = symbols('t')
N = 50*exp(0.4*t)
r = simplify(diff(N,t)/N)
print(r)
```

Απάντηση:
r=0.4

Άσκηση 3 Σημεία ισορροπίας λογιστικού μοντέλου $\mathbf{F(N) = 0.5N\left(1 - \frac{N}{100}\right)}$ Να βρεθούν τα σημεία ισορροπίας.

```
#NumPy
Python
import numpy as np
 coef = [-0.005,0.5,0]
 print(np.roots(coef))

#SymPy
Python
from sympy import *
 N = symbols('N')
 F = 0.5*N*(1-N/100)
 print(solve(F,N))
```

Απάντηση:

$N*=[0, 100]$

Άσκηση 4

Να εξεταστεί η ευστάθεια στο λογιστικό μοντέλο $F(N)=rN(1-N/200)$ με $r=1.2$

```
#SymPy
Python
from sympy import *
N = symbols('N')
F = 1.2*N*(1-N/200)
Fp = diff(F,N)
print(Fp)
print(Fp.subs(N,0))
print(Fp.subs(N,200))
```

Άσκηση 5 Μέγιστος ρυθμός ανάπτυξης

Να βρεθεί ο πληθυσμός όπου η ανάπτυξη είναι μέγιστη για το μοντέλο $F(N)=N(1-N/100)$.

```
#NumPy Python
import numpy as np
N = np.linspace(0,100,10000)
F = N*(1-N/100)
print(N[np.argmax(F)])
#SymPy Python
from sympy import *
 N = symbols('N')
```

```
F = N*(1-N/100)
print(solve(diff(F,N),N))
```

Απάντηση:
N=50

Άσκηση 6. Μέγιστη τιμή ανάπτυξης

Για την προηγούμενη άσκηση υπολογίστε το στο μέγιστο της τιμής ανάπτυξης F(50).

```
#SymPy Python
from sympy import *
 N = symbols('N')
 F = N*(1-N/100)
 print(simplify(F.subs(N,50)))
```

Απάντηση:
25

Άσκηση 7 Gompertz

Να υπολογιστεί η παράγωγος για το μοντέλο $N(t)=1000e^{-2(e^{0.1t}-1)}$.

```
# NumPy Python
import numpy as np
 def dNdt(t):
  return ( -200*np.exp(0.1*t)*np.exp(-2*(np.exp(0.1*t)-1)))

#SymPy Python
from sympy import *
 t = symbols('t')
 N = 1000*exp(-2*(exp(0.1*t)-1))
 print(diff(N,t))
```

Άσκηση 8

Διακριτό Μαλθουσιανό Μοντέλο $P_{t+1}=1.3P_t$ με $P_0=50$, Να βρεθεί το P_{10}

```
# NumPy Python
P = 50
for i in range(10):
P = 1.3*P
print(P)
```

```
# SymPy Python
from sympy import *
n = symbols('n', integer=True)
P0 = 50
Pn = P0*(1.3)**n
print(Pn.subs(n,10))
```

Άσκηση 9 Διακριτό λογιστικό μοντέλο

$P_{t+1}=P_t(1+2(1-(P_t/100)))$

Για $P_0=10$, να βρεθεί το P_5.

```
#NumPy  Python
P = 10
for i in range(5):
        P = P*(1 + 2*(1-P/100))
print(P)
#SymPy Python
from sympy import *
P = Rational(10)
for i in range(5):
P = simplify(P*(1+2*(1-P/100)))
print(P)
```

Άσκηση 10 Μέγιστη βιολογική παραγωγικότητα .

Στο λογιστικό μοντέλο :

$$\frac{dN}{dt} = 0.8\left(1 - \frac{N}{500}\right)$$

να βρεθεί ο πληθυσμός που δίνει τη μέγιστη ετήσια αύξηση.

```
# NumPy Python
import numpy as np
N = np.linspace(0,500,100000)
F = 0.8*N*(1-N/500)
print(N[np.argmax(F)])

# SymPy Python
from sympy import *
N = symbols('N')
F = 0.8*N*(1-N/500)
critical = solve(diff(F,N),N)
print(critical)
```

Απάντηση: $N=K/2=250$