ΣΥΝΗΘΕΙΣ ΔΙΑΦΟΡΙΚΕΣ ΕΞΙΣΩΣΕΙΣ

Η *sp.dsolve() εντολή Python χρησιμοποιείται για να λύσει Συνήθεις Δ.Ε. - ordinary differential equations (ODEs) χρησιμοποιώντας την sympy library της Python για συμβολικά Μαθηματικά. Η συνάρτηση dsolve() είναι ένα πανίσχυρο εργαλείο για επίλυση ODEs και μπορεί να επιλύσει ένα μεγάλο εύρος Δ.Ε.*.

Όμως:

Η dsolve() χρησιμοποιείται για να παρέχει συμβολικές (γενικές) λύσεις και δεν επιλύει αριθμητικά την Δ.Ε. Για Αριθμητικές επιλύσεις θα πρέπει να χρησιμοποιήσετε την ***scipy.integrate***

Αν η *dsolve() δεν μπορεί να βρει μια λύση κλειστής μορφής θα επιστρέψει ένα αποτέλεσμα χωρίς τιμή και θα πρέπει να χρησιμοποιήσετε αριθμητικές μεθόδους.*

sp.dsolve()

sp.dsolve(eq, func, ics)

- **eq**: Η Δ.Ε. προς επίλυση y(x). Πρέπει να είναι μια έκφραση sympy που αναφέρεται σε μια ODE.
- **func** (optional): Η συνάρτηση της ODE, που αντιστοιχεί στην y(x). Αυτή είναι απαραίτητη αν η ODEείναι συγκεκριμένη συνάρτηση. Αν την παραβλέψω η dsolve() προσπαθεί να ανιχνεύσει την συνάρτηση.
- **ics** (optional): Αρχικές συνθήκες για επίλυση της ODE. Παράδειγμα, {y(0): 1} που δείχνει ότι y(0) = 1. Αν δεν υπάρχουν αρχικές συνθήκες η dsolve() επιστρέφει μια γενική λύση.

Η εντολή eq = sp.Eq(y(x).diff(x), y(x)) χρησιμοποιεί τη βιβλιοθήκη SymPy της Python για να δημιουργήσει μια εξίσωση και την παράγωγό της.

Πχ Λύσε την dy/dx=y

```
import sympy as sp
# Ορισμός συμβόλων
x = sp.symbols('x')
y = sp.Function('y')
# Ορισμός ODE Εξίσωσης
eq = sp.Eq(y(x).diff(x), y(x))
# Λύση ODE
solution = sp.dsolve(eq)
print(solution)
```

Αποτέλεσμα!

Eq(y(x), C1*exp(x)) Γενική λύση $y(x)=C_1 . e^x$ όπου C1 σταθερά.

Ας λύσω την ίδια με αρχική συνθήκη y(0)=1:!!!!!!!!!!!

```
# Λύση ODE
import sympy as sp
# Ορισμός συμβόλων
```

```
x = sp.symbols('x')
y = sp.Function('y')
# Ορισμός ODE Εξίσωσης
eq = sp.Eq(y(x).diff(x), y(x))
# Λύση ODE με Αρχικές συνθήκες όπου η σταθερά C =1
ics={y(0):1}
solution = sp.dsolve(eq, ics=ics)
print(solution)
```

Αποτέλεσμα:

Eq(y(x), exp(x))

$y(x)= e^x$

Επίλυση Διαφορικής Εξίσωσης ΔΕΥΤΕΡΟΥ ΒΑΘΜΟΥ

$d^2y/dx^2+y=0$

Κώδικας Python

```
import sympy as sp
x = sp.symbols('x')
y = sp.Function('y')
# Ορισμός ODE Εξίσωσης
eq2 = sp.Eq(y(x).diff(x, x) + y(x), 0)
# Επίλυση της ODE
```

```
solution2 = sp.dsolve(eq2)

print(solution2)
```

Αποτέλεσμα!

Eq(y(x), C1*cos(x) + C2*sin(x))

Δηλαδή η Γενική επίλυση: y(x)=C1cos(x)+C2sin(x), where C1C_1C1 and C2C_2C2 are constants of integration.

ΜΕΡΟΣ Β.

Y'+3.y=x^2 και y(0)=1

```
import sympy as sp

x = sp.symbols('x')

y = sp.Function('y')(x)

equation = sp.Eq(y.diff(x)+ 3*y, x**2)

solution = sp.dsolve(equation, y, ics={y.subs(x,0):1})

print(solution)

# Βρες την τιμή του y(0.05)

y_at_0_05=solution.rhs.subs(x, 0.05)

print('y(0.05) = ', y_at_0_05)
```

ΣΧΟΛΙΟ!

***`solution.rhs`**: Στην SymPy, μια εξίσωση αναπαρίσταται ως αντικείμενο Ισότητας με δύο συνιστώσες: Το left-hand side (`lhs`) και το right-hand*

side (`rhs`). Κάθε πρόσβαση στο `solution.rhs` μας δίνει το right-hand side της εξίσωσης. Πχ αν η λύση είναι `Eq(f(x), 0)`, τότε το `solution.rhs` θα είναι το `0`.

Η λειτουργία ivp

Μπορεί να λύσει προβλήματα αρχικής τιμής και ΜΕΡΙΚΕΣ ΔΙΑΦΟΡΙΚΕΣ ΕΞΙΣΩΣΕΙΣ

Ένα Πρόβλημα αρχικής τιμής (IVP) Initial Value Problem για μια διαφορική εξίσωση περιλαμβάνει την επίλυση της διαφορικής εξίσωσης με ένα δεδομένο σύνολο αρχικών συνθηκών σε ένα συγκεκριμένο σημείο. Ο στόχος είναι να βρεθεί μια συνάρτηση που να ικανοποιεί τη διαφορική εξίσωση και επίσης να πληροί τις αρχικές συνθήκες

Η **solve_ivp** είναι μια συνάρτηση από το scipy.integrate module της Python, που λύνει προβλήματα αρχικής τιμής ... initial value problem". Είναι ένα ευέλικτο εργαλείο που χρησιμοποιείται για την αριθμητική επίλυση συνηθισμένων διαφορικών εξισώσεων (ODEs), δεδομένων των αρχικών συνθηκών και ενός χρονικού διαστήματος.

ΠΑΡΑΔΕΙΓΜΑ:

Λύσε την 1ης τάξεως ODE με αρχικές Τιμές Initial Values (Initial conditions) που περιγράφουν την κατάσταση σε μια αρχική στιγμή. Αρχική συνθήκη: Αυτή είναι η τιμή της συνάρτησης (ή των παραγώγων της) σε ένα συγκεκριμένο σημείο, συνήθως στο t=0 ή κάποιο άλλο σημείο εκκίνησης.
Θα λύσω την ODE: dy/dt=−2y+1
#με αρχική συνθήκη y(0)=0, στο χρονικό διάστημα t=[0,5].
#Πρόκειται για την εξίσωση: y’ +2y = 1

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
# Ορισμός συνάρτησης ΔΕ  dy/dt = -2y + 1
def model(t, y):
   return -2 * y + 1
# Αρχική συνθήκη
y0 = [0]
# Χρονικό Διάστημα για την λύση (Από  t=0 έως t=5)
t_span = (0, 5)
# Καθορίστε τα σημεία στα οποία θέλουμε να
αξιολογήσουμε τη λύση
t_eval = np.linspace(0, 5, 100)
# Λύση της Διαφορικής Εξίσωσης
sol = solve_ivp(model, t_span, y0, t_eval=t_eval)
# Έλεγχος
if sol.success:
   print("Η επίλυση ήταν επιτυχής \n")
# Εμφάνισε την αριθμητική λύση σε κάθε σημείο t_eval
print("t\t y(t)")
for t_val, y_val in zip(sol.t, sol.y[0]):
   print(f"{t_val:.2f}\t {y_val:.4f}")
# Εμφάνισε τη λύση σε ένα συγκεκριμένο σημείο πχ., t=0.05
t_specific = 0.05
```

```
y_at_0_05 = np.interp(t_specific, sol.t, sol.y[0])
print(f"\ny({t_specific}) = {y_at_0_05:.4f}")



# Γραφική παράσταση
plt.plot(sol.t, sol.y[0], label='y(t)')
plt.xlabel('Time t')
plt.ylabel('y(t)')
plt.title('Solution to dy/dt = -2y + 1')
plt.legend()
plt.grid(True)
plt.show()
```