

```

# ===== #
# RESOLUTION DES EQUATIONS DE LOTKA-VOLTERRA (1925). #
# METHODE D'EULER. SCHEMA EXPLICITE. #
# Version utilisant des tableaux numpy. #
# ===== #

# Importation des bibliothèques requises.
# =====
import numpy as np
import matplotlib.pyplot as plt

# Les paramètres du problème
# =====
alpha, beta, gamma, delta = 0.1, 0.2, 0.1, 0.3

# Conditions initiales
# =====
x0, y0 = 1., 1.5

# Nombre de points souhaités
# =====
npoints = int(input('Nombre de points souhaités ? '))

# Paramètres de l'algorithme
# =====
pas = 0.1

# Le coeur de l'algorithme : le schéma numérique proprement dit.
# =====
# Initialisation des tableaux des abscisses et des ordonnées (npoints valeurs).
# Ces tableaux sont initialisés avec des 0.
x, y = np.zeros(npoints), np.zeros(npoints)
x[0], y[0] = x0, y0
# Le schéma numérique.
for i in range(npoints-1):
    x[i+1] = x[i] * (1 + pas * (alpha - beta * y[i]))
    y[i+1] = y[i] * (1 - pas * (gamma - delta * x[i]))

# Affichage graphique
# =====

# On efface le graphique précédent
plt.clf()

# Génération des points du graphique à partir des tableaux x et y.
plt.plot(x,y,'ro',label = str(npoints) + ' points')

# Titre
plt.title('Résolution des équations de Lotka-Volterra (1925)',style='italic')

# Légendes des axes
plt.xlabel('Proies (x)')
plt.ylabel('Prédateurs (y)')

# Positionnement de la légende
plt.legend(loc = 'upper left')

# Affichage graphique
plt.show()

```