Matematika pro nematematiky - Úlohy 8

Termín zadání: 16.11.2024

1 Popis pomocí soustav diferenciálních rovnic

Napište soustavy diferenciální rovnice, které popisují následující situace.

1. Na louce roste tráva, trávu žerou krávy. Tráva roste konstantní rychlostí, ale krávy ji žerou rychlostí přímo úměrnou počtu krav. Krávy jednak spontánně umírají, jednak se množí přímo úměrně množství trávy.
2. HIV virus se množí v buňkách imunitního systému, čímž tyto buňky zároveň likviduje. Jinými buňkami imunitního systému, jejichž množství považujme nyní za konstantní, je virus odstraňován. Množí se tedy uměrně počtu buněk. Buňky zanikají úměrně počt virů a množí se rychlostí úměrnou vlastnímu počtu.

2 Metoda součtu řešení

V minulé sérii jste řešili homogenní lineární ODE

1. $x.\frac{dy}{dx} = y$
2. $\frac{dy}{dx} + 3x^2y = 0$

Obecnými řešeními byly funkce:

1. $y = kx$
2. $y = k.e^{-x^3}$

Vyřešte nyní následující nehomogenní rovnice, tak, že využijete skutečnosti, že obecné řešením nehomogenní rovnice O_N je součtem obecného řešení homogenní rovnice O_H a libovolného partikulárního řešení rovnice nehomogenní P_N. Zkuste toto partikulární řešení úhadnou.Nakonec najděte partikulární řešení nehomogenní rovnice pro libovolnou vámi zvolenou počáteční podmínku.

1. $x.\frac{dy}{dx} = y + 1$
2. $\frac{dy}{dx} + 3x^2y = 2$

3 Nehomogenní lineární diferenciální rovnice

Vyřešte stejnou úlohu jako v minuolé sérii, tentokrát však pomocí součtu řešení, nikoli pomocí variace konstant. Partikulární řešení nehomogenní rovnice uhádněte.

Do vany pouštíme vodu konstantní rychlostí. Voda ale otvorem ve dnu vytéká rychlostí přímo úměrnou objemu vody ve vaně. Na počátku je ve vaně 60 litrů a voda vytéká dnem rychlostí 2 litry/minutu. Vodu napouštíme konstantní rychlostí 1 litr/minutu. Při jakém objemu se ustaví rovnováha, tedy hladina už nebude klesat. Za jak dlouho rovnováha nastane? Pozn. Zhruba považujme dosažení rovnováhy po 5 poločasech.

4 Řešení algebraických rovnic numerickými metodami

Níže uvedený program v Pythonu má aplikovovat algoritmus pro řešení algebraické rovnice $\cos^2 x = x$ pomocí metody půlení intervalu. Doplňte vynechaná místa (___) v programu. Zvolte počáteční okraje intervalu, v němž hledáme řešení 0 a 1. Požadovanou přesnost výsledku 0.01. Kolik nejméně iterací musí proběhnout, abychom dosáhli požadované přesnosti.

```
# metoda půlení intervalů

import numpy as ___

def g(x):                   # definice funkce
    return x-___

xL = ___            # počáteční hodnota levého okraje intervalu
xR = ___            # počáteční hodnota pravého okraje intervalu
epsilon = _____     # kriterium ukončení iterací

iterace = 0

while (xR - xL) ___ epsilon:
     xM = (xL + xR)/2
     if g(xR)___g(xM) < 0:
            ___ = xM
     else:
            ___ = xM

iterace = iterace + 1

print(xL, "---", xR)
print(iterace)
```

5 Eulerova metoda pro řešení diferenciální rovnice

Chceme Eulerovou metodou numericky řešit diferenciální rovnici

$$\frac{dc}{dt} = -k.c^2 + v_0$$

Počáteční koncentraci v čase $t = 0$ zvolme $c = 10$, konstanty budou $k = 2$ a $v_0 = 5$. Časový krok zvolme $dt = 0.05$. Níže uvedený program v Pythonu má aplikovovat algoritmus pro Eulerovo řešení této rovnice. Doplňte vynechaná místa (___) v programu. V jakém časovém intervalu bude výpočet probíhat.

```
import _____ as np
import matplotlib.pyplot as ___

k = ___
```

```
v0 = ___

def derivace(c):
        return -k*c**2 ___

N = 150
time = np.zeros(N)
conc = np.zeros(N)

dt = ___
c = ___
t = ___

for i in range(N):
    conc[i] = c
    time[i] = ___
    c = ___ + derivace(c)*dt
    t  = t + ___

fig, axes = plt.subplots()

axes.plot(time, conc, 'b')

axes.set_xlabel('___')
axes.set_ylabel('___')
axes.set_title('vývoj koncentrace v čase')
axes.legend()
```

6 Eulerova metoda pro řešení soustavy diferenciálních rovnic

Chceme Eulerovou metodou numericky řešit soustavu diferenciálních rovnic

$$\frac{dx}{dt} = y$$
$$\frac{dy}{dt} = -x$$

Počáteční koncentraci v čase $t = 0$ zvolme $c = 10$, konstanty budou $k = 2$ a $v_0 = 5$. Časový krok zvolme $dt = 0.01$. Nechme program běžet 1000 kroků. Počáteční hodnoty zvolme $x = 5$ a $y = 5$, $t = 0$. Výsledek vykresleme v grafu, jednak jako funkci času, jednak jako fázový portrét, tedy jako vzájemnou závislost x a y.

Níže uvedený program v Pythonu má aplikovovat algoritmus pro Eulerovo řešení této soustavy. Doplňte vynechaná místa (___) v programu.

```
# Eulerova metoda - soustava diferenciálních rovnic

import ___
import matplotlib.pyplot ___

size = 1000
t = np.linspace(0, 50,size)
```

```
x = np.linspace(0, 50,size)
y = np.linspace(0, 50,size)

def derivace_x_podle_t(x,y,t):
      return ___

def derivace_y_podle_t(x,y,t):
      return ___

dt = ___

i=0
x[i] = ___
y[i] = ___
t[i] = 0

while i < size-1:
    dx = ___ * derivace_x_podle_t(x[i],y[i],t[i])
    dy = ___ * derivace_y_podle_t(x[i],y[i],t[i])
    x[i+1] = x[i] + ___
    y[i+1] = y[i] + ___
    t[i+1] = t[i] + ___
    i += 1

fig, axes = plt.subplots()

axes.plot(t, x, 'b', label = 'x')
axes.plot(t, y, 'r', label = 'y')
axes.set_xlabel('___')
axes.set_ylabel('___')
axes.legend()

# fázový portrét

axes.plot(___, ___, 'b')
axes.set_xlabel('x')
axes.set_ylabel('y')
```