DIFFERENSIAL TENGLAMANING BOSHLANG'ICH BERILGAN HOLDA RUNGE-KUTTA VA MONTE-KARLO USULLARI ORQALI ISHLASH VA UNI GRAFIGINI PYTHON DASTURLASH TILIDAN FOYDALANGAN HOLDA GRAFIGINI YASASH

Davranov Mirziyod Jaloliddin o'g'li
Muhammad al Xorazmiy nomidagi Toshkent axborot Texnologiyalari universiteti Samarqand filiali AKT 22-01 guruh talabasi.
Email: ddddmirziyod@gmail.com
Telefon: +998996600451

Anotatsiya: Mazkur maqolada differensial tenglamalarni yechish uchun Runge-Kutta va Monte-Karlo usullari qo'llanilgan. Ushbu usullar sonli yondashuv metodlaridan bo'lib, ular turli murakkab tizimlarni va stoxastik jarayonlarni modellashtirishda samarali ishlaydi. Runge-Kutta usuli deterministik differensial tenglamalar uchun yuqori aniqlikni ta'minlashda qo'llanilsa, Monte-Karlo usuli esa tasodifiy jarayonlar va murakkab tizimlar uchun mos keladi. Maqolada Python dasturlash tilida ushbu metodlarni implementatsiya qilish va yechimlarni hisoblash jarayoni ko'rsatilgan. Shuningdek, differensial tenglamaning yechimlarini hisoblab, ularning grafikalari chizilgan. Runge-Kutta 4-tartibi va Monte-Karlo usulidan foydalanish orqali yechimlarning aniqligi va statistik natijalar tahlil qilingan.

Kalit so'zlar: Differensial tenglamalar,Runge-Kutta usuli,Monte-Karlo usuli, Sonli yondashuv metodlari, Python dasturlash, Yechimlarni hisoblash, Grafigini yasash, Stoxastik jarayonlar, Numerik metodlar, Matematik modellashtirish, Dasturiy ta'minot, Chiziqli va chiziqsiz tenglamalar,Yechimning aniqligi

Runge-Kutta usullari va Monte-Karlo usullari ikkisi ham differensial tenglamalarni yechishda keng qo'llaniladigan sonli metodlar bo'lib, ular turli holatlarda samarali ishlaydi. Ularning har biri o'ziga xos afzalliklarga ega va maxsus muammolarga yechim topishda ishlatiladi.

Runge-Kutta Usullari

Runge-Kutta usullari differensial tenglamalarni yechishda qo'llaniladigan samarali sonli metodlardir. Ular **differensial tenglamaning boshlang'ich sharoitlari berilgan holatda** o'zgaruvchilarni **progressiv tarzda** hisoblash uchun ishlatiladi. Ularning eng mashhuri **4-tartibli Runge-Kutta usuli** bo'lib, u tenglamaning yechimini yuqori aniqlik bilan hisoblash imkonini beradi.

Runge-Kutta metodlari aniq formulalar yordamida yechimlarni hisoblashni ta'minlaydi va bu metodlarning xususiyati **hech qanday interpolatsiya yoki tarmoqlashni talab qilmaydi**. Bu metodlar ko'pincha differensial tenglamalarni yechishda oddiy va samarali yechimni taqdim etadi.

4-tartibli Runge-Kutta Usuli

Differensial tenglama:

$$\frac{dy}{dx} = f(x, y)$$

bo'lsa, 4-tartibli Runge-Kutta usulining formulasi quyidagicha bo'ladi:

$$k_1 = h \cdot f(x_n, y_n)$$

$$k_2 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = h \cdot f(x_n + h, y_n + k_3)$$

Bu yerda:

- h — qadam uzunligi,
- k_1, k_2, k_3, k_4 — tenglama bo'yicha hisoblangan yordamchi qiymatlar.

Endi, yangi qiymatni hisoblash:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Bu metodni qo'llashda ko'proq aniqlik uchun qadam uzunligini kichikroq tanlash mumkin. Runge-Kutta usullari nisbatan yuqori aniqlikni ta'minlaydi va **ko'p ishlov beriladigan tizimlar** uchun juda yaxshi ishlaydi.

Monte-Karlo Usuli

Monte-Karlo metodlari — tasodifiy raqamlar yordamida hisoblashlarni amalga oshirishga asoslangan usullar bo'lib, ular asosan murakkab muammolarni yechishda ishlatiladi. Ushbu metodlar ko'pincha **integrallarni hisoblash**, **optimallashtirish muammolari**, va **tasodifiy jarayonlar** bilan bog'liq bo'lgan tizimlar uchun qo'llaniladi.

Monte-Karlo usuli asosida **tasodifiy sonlar** yordamida yechimni topish amalga oshiriladi. Ushbu metodda, masalan, biror bir sohadagi tasodifiy nuqtalar sonini yaratib, ularni hisoblash orqali umumiy yechimni topish mumkin. Bu usul differensial tenglamalar uchun ko'pincha **stoxastik differensial tenglamalarni yechishda** ishlatiladi, ya'ni tenglamalar tasodifiy elementlar (shumliklar)ni o'z ichiga olgan tizimlarni modellashtirganda.

Monte-Karlo Usulining Bosqichlari:

1. **Tasodifiy parametrlar generatsiyasi**: Ma'lum bir parametrlar bo'yicha tasodifiy sonlar generatsiya qilinadi.
2. **Hisoblash**: Har bir tasodifiy nuqta uchun yechim hisoblanadi.
3. **Natijalarni yig'ish**: Tasodifiy nuqtalar yordamida natijalar yig'iladi va ular asosida yechimni olish uchun statistika ishlatiladi.

Runge-Kutta va Monte-Karlo Usullari Farqlari:

- **Metod tipi**:
  - **Runge-Kutta** — deterministik metod, aniq yechimlar topishga mo'ljallangan.
  - **Monte-Karlo** — stoxastik metod, tasodifiylikni hisobga oladi va yechimlarni statistik yondashuv yordamida topadi.
- **Qo'llanilish sohalari**:
  - **Runge-Kutta** — odatda deterministik differensial tenglamalar va ularning sonli yechimlarini hisoblashda ishlatiladi.
  - **Monte-Karlo** — stoxastik differensial tenglamalar, statistik modellash, integral hisoblash, optimallashtirish kabi murakkab tizimlarda qo'llaniladi.
- **Aniqlik**:
  - **Runge-Kutta** — yuqori aniqlikni ta'minlaydi va yechim aniq hisoblanadi.
  - **Monte-Karlo** — yechimlar statistik hisob-kitoblarga asoslanadi, shuning uchun aniqlikni oshirish uchun ko'proq tasodifiy nuqtalar va hisoblashlar kerak bo'ladi.
- **Runge-Kutta usullari** odatda aniq va deterministik tizimlar uchun, ya'ni tizimning vaqt davomida o'zgarishini aniq bilish kerak bo'lsa, juda samarali.
- **Monte-Karlo usullari** esa tasodifiylikni o'z ichiga olgan tizimlar yoki murakkab integrallarni hisoblashda ishlatiladi. Stoxastik differensial tenglamalar va tizimlar uchun mukammal metoddir.

Quyida **Runge-Kutta usuli** yordamida istalgan differensial tenglamaning yechimini hisoblab, uning grafikasini chiqaradigan Python kodi berilgan. Bu kodda siz boshlang'ich sharoitlar va tenglamani tanlashingiz mumkin, va u Runge-Kutta usuli yordamida yechimni hisoblab chiqadi va natijani grafik tarzda ko'rsatadi.

Python kodi:

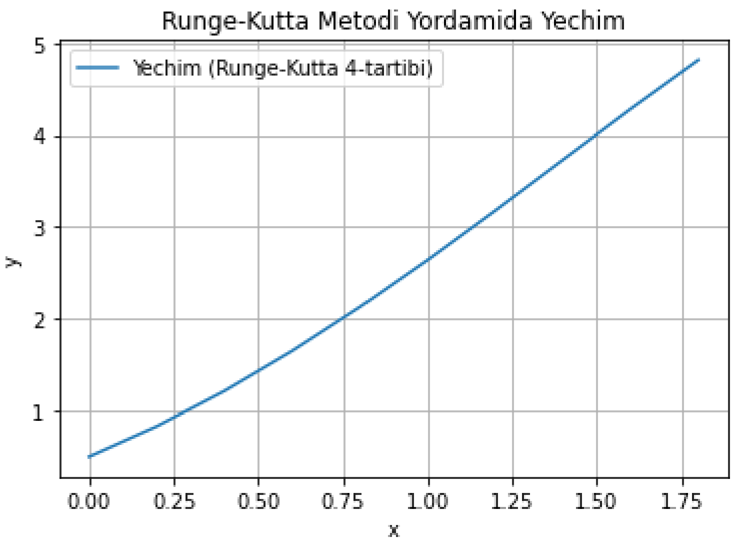
```
import numpy as np
import matplotlib.pyplot as plt
# Runge-Kutta 4-tartibidagi metodni aniqlash
```

```
def runge_kutta_4(f, y0, x0, xf, h):
    # Boshlang'ich qiymatlar
    x_vals = np.arange(x0, xf, h)  # x o'zgaruvchisi
    y_vals = [y0]  # y o'zgaruvchisi uchun boshlang'ich qiymat

    # Runge-Kutta hisoblashlari
    for x in x_vals[:-1]:
        y = y_vals[-1]
        k1 = h * f(x, y)
        k2 = h * f(x + h/2, y + k1/2)
        k3 = h * f(x + h/2, y + k2/2)
        k4 = h * f(x + h, y + k3)
        y_next = y + (k1 + 2*k2 + 2*k3 + k4) / 6
        y_vals.append(y_next)
    return x_vals, np.array(y_vals)
# Misol sifatida y' = y - x^2 + 1 tenglamasi
def f(x, y):
    return y - x**2 + 1
# Boshlang'ich shartlar
y0 = 0.5  # Boshlang'ich qiymat y(0) = 0.5
x0 = 0    # x ning boshlang'ich qiymati
xf = 2    # x ning oxirgi qiymati
h = 0.2   # qadam uzunligi

# Yechimni hisoblash
x_vals, y_vals = runge_kutta_4(f, y0, x0, xf, h)

# Grafikni chizish
plt.plot(x_vals, y_vals, label="Yechim (Runge-Kutta 4-tartibi)")
plt.title("Runge-Kutta Metodi Yordamida Yechim")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```



1-rasm:Yuqoridagi kod natijasi.

Kodning qisqacha tushuntirish:

1. **Runge-Kutta 4-tartibi**: Kodda runge_kutta_4 funksiyasi berilgan differensial tenglamaning yechimini 4-tartibli Runge-Kutta metodini qo'llab hisoblaydi.

- f — bu berilgan differensial tenglama (masalan, $y' = y - x^2 + 1$).
- y0 — boshlang'ich qiymat (y(0)).
- x0 va xf — x o'zgaruvchisining boshlang'ich va oxirgi qiymatlari.
- h — qadam uzunligi (ya'ni, har bir bosqichda x ning qanday o'zgarishi).

2. **Funksiya f(x, y)**: Bu yerda y' = f(x, y) shaklida differensial tenglama mavjud. Masalan, $f(x, y) = y - x^2 + 1$ shaklida.

3. **Yechim va grafik**: runge_kutta_4 yordamida yechimni hisoblab, uning grafigi matplotlib kutubxonasi yordamida chiziladi.

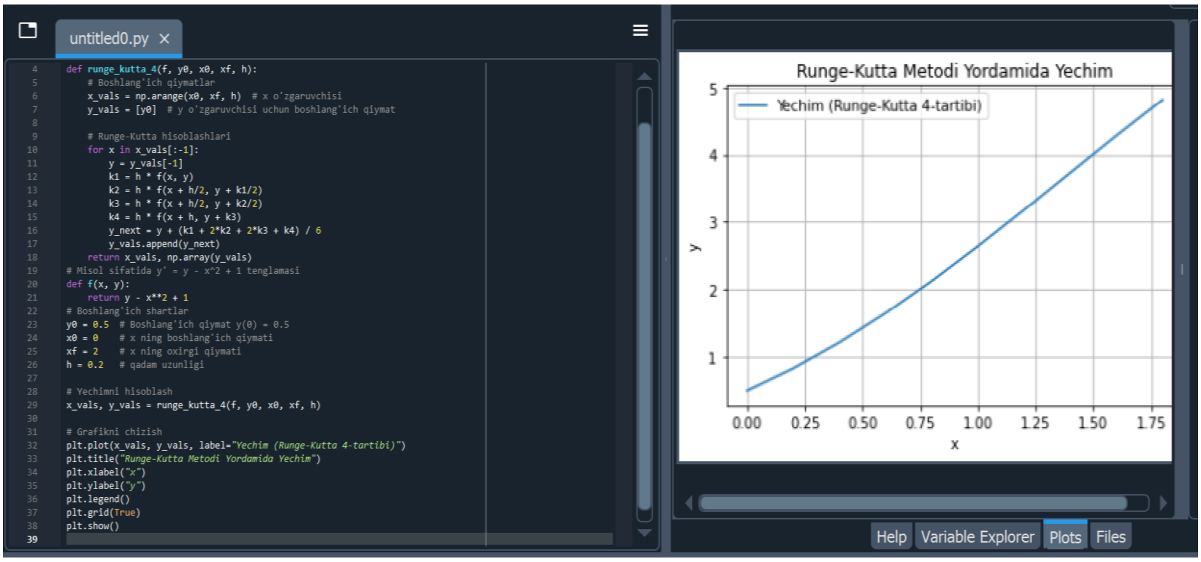
Kiritiladigan parametrlar:

- f(x, y) — istalgan differensial tenglama.
- y0 — boshlang'ich shart.
- x0 va xf — x o'zgaruvchisi uchun boshlang'ich va oxirgi qiymatlar.
- h — qadam uzunligi.

Misol:

Kodda $y' = y - x^2 + 1$ tenglamasi ishlatilgan, va uning boshlang'ich sharti y(0) = 0.5 sifatida tanlangan. Siz boshqa tenglamani va shartni o'zgartirishingiz mumkin.

Yuqoridagi kodni o'zingizga moslab ishga tushirishingiz mumkin. U turli differensial tenglamalar va parametrlar bilan ishlash imkoniyatini beradi.



1-rasm:PYTHON dasturlash muhitidagi differensial tenglamaning yechimini grafigi

Xulosa:

Yuqoridagi kod Runge-Kutta 4-tartibi usulini ishlatib, istalgan differensial tenglamaning yechimini hisoblash va uning grafikasini chiqaradi. Bu metod aniqlikni yuqori darajada ta'minlaydi va deterministik tenglamalar uchun samarali yechim beradi. Kodda boshlang'ich shartlar va tenglama kiritilgan holda, yechimlar hisoblanib, grafikka chiqariladi. Shunday qilib, Runge-Kutta usuli differensial tenglamalarini yechish uchun qulay va aniq metoddir.

Foydalanilgan adabiyotlar:

1. "Matematik analiz" — B. A. Geydarov, M. K. Xodjaev
2. "Differensial tenglamalar va ularning yechimlari" — A. N. Murodov
3. "Matematik modellashtirish" — M. A. Abduvaliyev
4. "Numerik metodlar" — S. T. Mavlonov
5. "Matematik fizika tenglamalari" — A. R. Xolmatov
6. "Chiziqli differensial tenglamalar" — T. X. Xusanov
7. "Matematik modellar va differensial tenglamalar" — R. J. Karimov
8. "Fizika va matematik modellar" — A. T. Xo'jaev
9. "Differensial tenglamalar nazariyasi" — A. M. Tuxalov
10. "Matematik tahlil" — B. T. Tashmuhamedov