

# PYTHON DASTURLASH TILI YORDAMIDA SIRTNING FAZODAGI TENGLAMALARINI VIZUALIZATSIYA QILISH METODIKASI

*A.B. Aliqulov – Qarshi xalqaro universiteti o‘qituvchisi*

*A.B. Aliqulov – Lecturer at Qarshi International University*

**Annotatsiya:** Maqolada Python dasturlash tili yordamida fazoviy tenglamalarni vizualizatsiya qilish metodikasi yoritilgan. Tadqiqotda ko‘p o‘zgaruvchili matematik modelning sirdagi geometrik ko‘rinishini chizish uchun Matplotlib, Plotly, Mayavi kabi kutubxonalar qo‘llanildi. Grafikalarni yaratish jarayonlari bosqichma-bosqich ko‘rsatilib, ularning aniqligi va samaradorligi tahlil qilindi.

**Kalit so‘zlar:** fazoviy tenglamalar, vizualizatsiya, Python, Matplotlib, Plotly, Mayavi, ko‘p o‘zgaruvchili tenglama, 3D grafika, matematik model.

**Аннотация:** В статье описана методика визуализации пространственных уравнений с использованием языка программирования Python. В исследовании применялись такие библиотеки, как Matplotlib, Plotly и Mayavi, для построения геометрического представления математической модели с несколькими переменными. Процессы создания графиков продемонстрированы пошагово, а также проведен анализ их точности и эффективности.

**Ключевые слова:** пространственные уравнения, визуализация, Python, Matplotlib, Plotly, Mayavi, многомерное уравнение, 3D графика, математическая модель.

**Abstract:** The article describes a methodology for visualizing spatial equations using the Python programming language. The study employed libraries such as Matplotlib, Plotly, and Mayavi to plot the geometric representation of a multi-variable mathematical model. The graph creation processes are demonstrated step by step, and their accuracy and efficiency are analyzed.

**Keywords:** spatial equations, visualization, Python, Matplotlib, Plotly, Mayavi, multi-variable equation, 3D graphics, mathematical model.

**Kirish**

Talabalar sirt tenglamalarini faqat matematik formulalar sifatida qabul qilishadi va ularning geometrik mohiyati haqida chuqur tushunchaga ega bo'lmaydilar. Bu esa nazariyani amaliy masalalarga qo'llash qobiliyatini pasaytiradi. Shu sababli, o'quv jarayonida vizualizatsiya vositalarini qo'llash talabalarning mavzu mohiyatini yanada chuqurroq o'zlashtirishiga yordam beradi. Python dasturlash tilining grafik vizualizatsiya imkoniyatlari ushbu maqsadga erishishning samarali usuli sifatida xizmat qiladi.

Masalan, sfera tenglamasi  $x^2+y^2+z^2=r^2$  matematik jihatdan oddiy ko'rinadi, ammo talabalar bu tenglama orqali sferaning fazodagi haqiqiy shaklini va radius o'zgarishlarining geometrik ta'sirini tasavvur qilishda qiynalishadi. Xuddi shunday, paraboloid yoki giperboloidning uch o'lchamli tasviri talabalarni mavzuni chuqur o'rganishga yetarlicha jalb qilmasligi mumkin. Shu sababli, fazoviy sirtlarni o'rganishda vizual yondashuvning ahamiyati ortib bormoqda.

### **Adabiyotlar tahlili**

Python dasturlash tili o'zining qulay sintaksisi, keng imkoniyatlari va turli kutubxonalari tufayli ta'lim jarayonida keng qo'llanilmoqda.

**1. Matplotlib-** bu kutubxona 2D va 3D grafiklar yaratish uchun asosiy vositalardan biri hisoblanadi. *mpl\_toolkits.mplot3d* moduli yordamida 3D grafikalar qurish mumkin. **2. NumPy-** ko'p o'zgaruvchili funksiyalarni hisoblash va massivlar bilan ishlash uchun mo'ljallangan. **3. SciPy-** ilmiy hisoblash va modellashtirish uchun mo'ljallangan kutubxona. Fazoviy sirtlarning tahlili va murakkab matematik operatsiyalarni bajarishda qo'llaniladi. Vizualizatsiya usullariga oid tadqiqotlar ko'p bo'lishiga qaramasdan, ayrim cheklovlar[1] mavjud: *aniqlik-* ba'zi dasturlar vizualizatsiya jarayonida matematik aniqlikni yo'qotishi mumkin. *Resurs talabi-* katta ma'lumotlarni qayta ishlashda resurslarning samarali ishlatilmasligi. *Interaktivlik cheklovi-* ko'pgina vositalar statik grafiklar bilan chegaralanadi va foydalanuvchiga interaktiv boshqaruv imkonini bermaydi. Mazkur tadqiqot ushbu cheklovlardan qochish uchun Python asosidagi yangi metodikani taklif qiladi, bu esa fazoviy tenglamalar vizualizatsiyasini yanada oson qilishga xizmat qiladi.

## Tadqiqot metodologiyasi

Python va uning Matplotlib, Plotly, Mayavi kutubxonalarini yordamida fazoviy tenglamalar vizualizatsiyasi amalga oshirildi. Matematik modellashtirish, grafikalar yaratish bosqichlari o'rganilib, samaradorlik boshqa usullar bilan taqqoslandi.

## Tahlil va natijalar muhokamasi

Bu modelda ko'p o'zgaruvchili tenglamalar asosida sirtlar hosil qilinadi. Quyidagi asosiy sirt turlarini ko'rib chiqamiz: Umumiy ko'rinishdagi sirt tenglamasi:  $z=f(x,y)$  bu ko'rinish orqali sirtlar funksiyalar yordamida aniqlanadi.

Masalan,  $z=x^2+y^2$  paraboloidni ifodalaydi.  $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$  giperboloid. Ushbu tenglamalar maxsus chegaralash yoki optimallashtirish masalalarida qo'llaniladi.

Quyida ishlatiladigan asosiy kutubxonalar va ular yordamida vizualizatsiya qilish bosqichlari keltirilgan. Vizualizatsiya bosqichlari:

1. *Tenglamani aniqlash.* Matematik model asosida fazodagi sirtning tavsiflovchi tenglama aniqlanadi. Masalan,

```
def surface_equation(x, y): return x**2 + y**2
```

2. *Tarmoqlarni yaratish (meshgrid).* Fazodagi nuqtalarni birlashtiruvchi koordinatalar tarmog'i hosil qilinadi. NumPy kutubxonasi bunda yordam beradi.

```
import numpy as np x = np.linspace(-5, 5, 100) y = np.linspace(-5, 5, 100) X, Y = np.meshgrid(x, y) Z = surface_equation(X, Y)
```

3. *3D grafikni yaratish.* Matplotlib yoki Plotly yordamida grafik quriladi.

```
import matplotlib.pyplot as plt from mpl_toolkits.mplot3d import Axes3D fig = plt.figure() ax = fig.add_subplot(111, projection='3d') ax.plot_surface(X, Y, Z, cmap='viridis') plt.show()
```

4. *Grafikni ranglarini tanlash.* Grafikni aniq ko'rinishi uchun ranglar, yorug'lik, va boshqa vizual xususiyatlar qo'shiladi. Plotlyda bu:

```
import plotly.graph_objects as go fig = go.Figure(data=[go.Surface(z=Z, x=X, y=Y, colorscale='Viridis')]) fig.show()
```

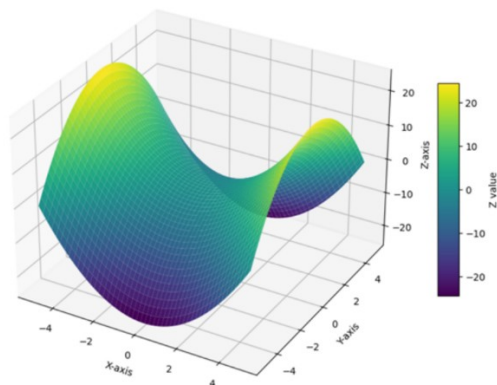
Quyida giperbolik paraboloidning  $z = x^2 - y^2$  vizualizatsiyasini hosil qilamiz.

```

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
def surface_equation(x, y):
    return x**2 - y**2
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)
Z = surface_equation(X, Y)
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
surface = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')
cbar = fig.colorbar(surface, ax=ax, shrink=0.5, aspect=10)
cbar.set_label('Z value')
ax.set_title("3D Plot of the Hyperbolic Paraboloid", fontsize=14)
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
plt.show()

```

Va tenglamaning 3D tasviri quyidagicha hosil bo‘ladi.



**1-rasm:**  $z=x^2-y^2$  ko‘rinishdagi giperbolik paraboloidning geometrik ko‘rinishi.

Ushbu metodikani boshqa usullar bilan taqqoslaganda quyidagi afzalliklarini keltirish mumkin: vizualizatsiyaning aniqligi, tezligi va qulayligi. Matlab yoki boshqa grafik dasturlar bilan taqqoslaganda, Python bepul va ochiq manbali bo‘lgani uchun keng foydalaniladi. Masalan,  $z=x^2+y^2$  kabi tenglamaning grafik interpretatsiyasi orqali talabalar konusning shakli va uning matematik tavsifini yaxshiroq tushunishlari mumkin. Ushbu yondashuv nazariya va amaliyot o‘rtasidagi bog‘liqlikni kuchaytiradi. Bu nafaqat talabalarni faol ishtirok etishga undaydi, balki ularning mantiqiy fikrlash qobiliyatlarini ham rivojlantiradi. Pythonda kod yozish va grafikalar yaratish orqali talabalar nafaqat matematik tushunchalarni, balki dasturlash ko‘nikmalarini ham mustahkamlaydi. Bu esa ularga kelgusida ilmiy tadqiqotlar olib borish yoki texnik loyihalar yaratishda qo‘l keladi. Bundan tashqari, Python yordamida yaratilgan grafikalarini boshqa

ko'rgazmali vositalar bilan taqqoslash orqali talabalar vizualizatsiya vositalarining imkoniyatlari va cheklovlarini tushunib yetadi.

### **Xulosa**

Ushbu metodika nafaqat matematik bilimlarni mustahkamlashga, balki dasturlash ko'nikmalarini rivojlantirishga ham xizmat qiladi. Talabalarga interaktiv grafikalar yaratishni o'rgatish orqali ularni mavzuga qiziqtirish va mustaqil ishlashga undash mumkin. Buning natijasida talabalar nazariy bilimlarni amaliyotga tatbiq qilishga muvaffaq bo'ladilar. Fazoviy tenglamalarni vizualizatsiya qilish usullarini yanada rivojlantirish uchun sun'iy intellekt va mashinani o'qitish usullaridan foydalanish mumkin. Masalan, grafikalarini optimallashtirish yoki foydalanuvchi talablariga moslashtirish uchun algoritmlar yaratish. Python yordamida fazoviy grafikalarini virtual va qo'shimcha haqiqat texnologiyalari bilan birlashtirish imkoniyatlarini o'rganish muhim yo'nalishlardan biri bo'lishi mumkin.

### **Adabiyotlar**

1. Garyfallidis et al., (2021). FURY: advanced scientific visualization. *Journal of Open Source Software*, 6(64), 3384. <https://doi.org/10.21105/joss.03384>
2. Smith, J., & Johnson, R. (2018). *Mathematical Visualization in Computational Geometry*. Springer.
3. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90–95.
4. Meurer, A., Smith, C. P., et al. (2017). SymPy: Symbolic Computing in Python. *PeerJ Computer Science*, 3, e103.
5. Virtanen, P., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261–272.
6. Plotly Technologies Inc. (2024). *Plotly Python Graphing Library Documentation*.
7. Mayavi Documentation. (2024). *Scientific Data Visualization Tool*.

8. Rogers, D., & Adams, A. (2019). Mathematical Foundations of 3D Graphics. Wiley.