

# A Visit és a VTK fájlformátum egy lehetséges alkalmazása

PTE TTK  
Vizualizáció  
2024/25/1  
Király Márk



# A prezentáció tartalma

A témában a keresés alapvetően vizuálisabbnak indult, de rájöttem, hogy Python-nal egyszerűbben megoldhatóak a dolgok, így a VISIT csak szemléltetésként tűnik fel. Azonban a VISIT által is használt VTK fájl kezelése pótolja az élményt.

**Megjegyzés: A Pythonban használt könyvtárak C++ nyelven íródtak (futásidő miatt fontos)!**

- A kiindulási probléma matematikai bevezetése, leírása
  - A probléma paraméteres görbékről szól.
- VTK fájl létrehozása:
  - Pythonnal fogjuk létrehozni a görbe adott mintavételezéssel készült VTK fájlját.
- VTK fájl megjelenítése VISIT-ben
  - A szemléltetés kedvéért megjelenítjük a görbét VISIT-tel.
- Példa számítás végzése a VTK fájlon
  - Pythonnal egyszerűen elvégezzünk egy numerikus számítást a VTK fájlon, majd rávilágítunk, hogy milyen nehezen lehetne szimbolikusan megtenni ugyanezt.

# A kiindulási probléma

## A probléma a következő:

Szeretnénk meghatározni paraméteres alakban megadott (zárt) görbék ívhosszát. A probléma megoldása szimbolikus matematikai módszerekkel nehézkes, mivel gyök alatti (ún. irracionális) kifejezéseket kell integrálnunk. Ráadásuk vannak olyan esetek is, amikor ezeket szimbolikus matematikai módszerekkel egyáltalán nem is lehet kiszámolni. Itt jön a képbe a numerikus számítás és a VISIT.

## A konkrét példa, amiről a prezentáció a továbbiakban szólni fog:

Az alábbi (zárt) görbe ívhosszát fogjuk numerikusan kiszámolni VTK fájlkezelés segítségével:

$$x(t) = t \cdot (t - 1)$$

$$y(t) = t^2 \cdot (1 - t)^3, \text{ ahol: } 0 \leq t \leq 1$$

# Görbéket tartalmazó VTK fájlok létrehozása Python-nal

Az alábbiakban a görbe képlete alapján generálunk például 100 darab koordinátát, majd ezeket a koordinátákat sorozatban összekapcsoljuk egymással, így létrehozva a görbe ívét (=élét, vonalát). Az így kapott objektumot VTK fájlként elmentjük a lemezre, ezt VISIT-tel megjeleníthetjük.

```
import numpy as np
import pyvista as pv

# Paraméterek
num_points = 100 # Mintavételi pontok száma
# t paraméter a [0, 1] tartományban
t = np.linspace(0, 1, num_points)

# Képlet szerinti koordináták generálása
x = t * (1 - t) #  $x(t) = t(1-t)$ 
y = t**2 * (1 - t) ** 3 #  $y(t) = t^2(1-t)^3$ 
z = np.zeros_like(x) # 2D síkban:  $z = 0$ 

# Pontok definiálása
points = np.column_stack((x, y, z))

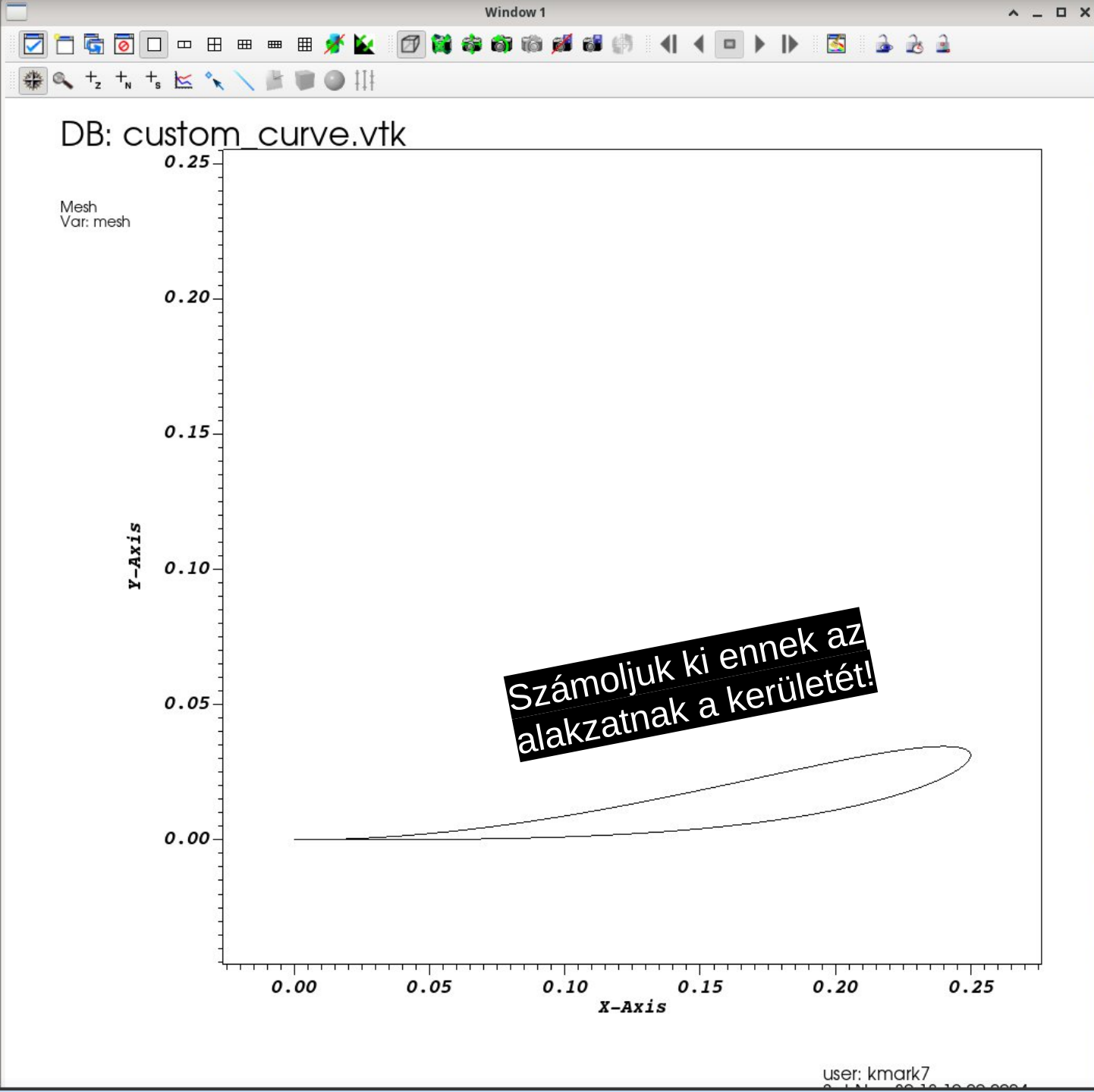
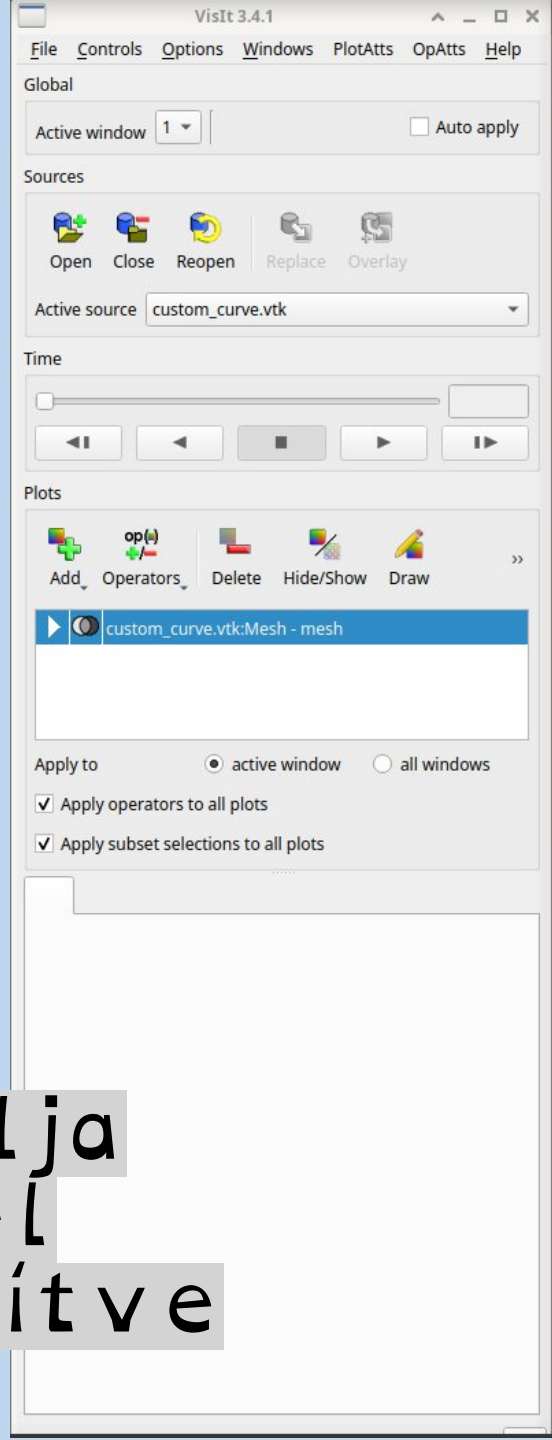
# PyVista PolyData objektum
line = pv.PolyData()
line.points = points

# Csatlakozó élek létrehozása
# Az éleknek 3 értéket kell tartalmaznia: 2 (értékek száma), majd az index
lines = np.zeros((num_points, 3), dtype=int)
for i in range(num_points):
    lines[i, 0] = 2 # Két pontot összekapcsol
    lines[i, 1] = i # Az i-edik pont
    # A következő pont (körbefordulva az utolsó pont az elsővel összekapcsolódik)
    lines[i, 2] = ((i + 1) % num_points)

# Létrehozott élek hozzárendelése
line.lines = lines

# VTK fájl mentése
line.save("custom_curve.vtk")

print("A görbe VTK fájlba mentve: custom_curve.vtk")
```



A  
görbe  
VTK fájlja  
VISIT-tel  
megjelenítve

# Numerikus számítások a VTK fájlal

Ahogy az előző dián említve volt, most az előbb látott alakzat kerületét (tehát a görbe ívét) határozzuk meg. **(Próbáltam a VISIT GUI-ban erre eszközt keresni, de nem találtam...)** Egyszerűen csak végigmegyünk a betöltött VTK fájl tartalmán: a koordinátákon, és az egymás mellettiek távolságát kiszámítjuk, és összeszummázzuk, így megkapva a teljes (zárt) görbe ívhosszát.

```
import numpy as np
import pyvista as pv

# A görbét tartalmazó VTK fájl betöltése
line = pv.read("custom_curve.vtk")
points = line.points # A görbe pontjainak kinyerése a fájlból

# Az ívhossz kiszámítása: összeadjuk a két egymást követő pont közötti távolságokat
length = 0
for i in range(len(points) - 1): # Végigiterálunk a pontokon, kivéve az utolsó előtti pontot
    p1 = points[i] # Az i-edik pont
    p2 = points[i + 1] # A következő pont
    length += np.linalg.norm(p2 - p1) # A két pont közötti távolság kiszámítása és hozzáadása az összhozhoz

# Az összesített ívhossz kiírása
print("Total curve length:", length)
```

**A PROGRAM KIMENETE, VAGYIS AZ EREDMÉNY:** \$ python python\_teszt\_szamit.py  
Total curve length: 0.5100743859947733

## Megjegyzés:

A prezentáció során megfogalmazott konkrét példa megoldása szimbolikusan így nézne ki:

$$\int_0^1 \sqrt{(1-2t)^2 + (2-5t)^2(-1+t)^4 t^2} dt$$

Az integrál értéke nagyon kicsi hibaküszöbvel az előbb kapott érték: 0.5100743859947733



Aki nem hiszi, számoljon utána!

# Köszönöm a figyelmet!