



Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás

Kiegészítő számítási feladatok nyílt forráskódú eszközökkel

GIS Open konferencia 2026

Nagy Gábor

Óbudai Egyetem, Alba Regia Kar, Geoinformatikai Intézet

2026. április 24.



Kiegyenlítő számítási feladatok nyílt forráskódú eszközökkel

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel
Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítési
feladatok
NumPy-al

Összefoglalás

Milyen lehetőségeink vannak földmérőként, ha a munkánk során felmerülő kiegyenlítési feladatokra nyílt forráskódú eszközöket szeretnénk alkalmazni?

- a GNU Gama használata
 - eszköz a tipikus geodéziai kiegyenlítési feladatokra
 - többféle típusú hálózat számítása
 - többféle bemeneti és kimeneti formátum
- saját alkalmazás fejlesztése
 - szükség lesz valamilyen könyvtárra (modulra), amivel a numerikus számítások (mátrix és vektor műveletek) hatékonyan és egyszerűen megvalósíthatóak
 - Python programok esetében egy ilyen modul a NumPy



A GNU Gama bemutatása

Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás

- nyílt forráskódú:
 - ingyenesen használható
 - a forráskódja (C++) is elérhető
- fő fejlesztője Aleš Čepek
- letölthető a <https://www.gnu.org/software/gama/> oldalról



A GNU Gama használata

Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás

- parancssorból (gama-local), többféle bemeneti formátummal
 - XML
 - YAML
 - SQLite adatbázis
- Qgama, QT alapú grafikus felület
- a kimenet is többféle lehet (mindkét esetben):
 - XML
 - jegyzőkönyv többféle nyelven, akár magyarul is



A GNU Gama előnyei

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítési
feladatok
NumPy-al

Összefoglalás

- nyílt forráskódú
- geodéziai hálózatok kiegyenlítése (vízszintes, szintezési, trigmag)
- szabad hálózatok kezelése
- jól paraméterezhető pl. a mértékgységek tekintetében
- többféle be- illetve kimeneti formátum
- a kiegyenlítő számítás mindenféle paramétere elérhetővé válik (pl. hibaellipszisek adatai)



Mi a NumPy?

Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás



- egy Python modul
- külön telepítendő (nem built-in) modul, de az egyik leginkább elterjedt ilyen kiegészítő a Python nyelvhez
- numerikus számítások egyszerű és hatékony elvégzésére hozták létre
- nagyrészt C-ben készült, gépi kódra lefordított elemekből áll, ezért a számítások hatékonyan működnek (memóriaigény, futásidő)



A NumPy alkalmazása

Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás

- a modul elemeihez így az `np.elem` alakban tudunk hozzáférni
- A NumPy tetszőleges dimenziószámú rendezett adathalmazokat kezel,
- amelyek elemei egyforma típusúak
- a tömbben tárolt adatok változhatnak a program futása során
- A NumPy-ban kezelhető adatok:
 - vektor (1D)
 - mátrix (2D)
 - ndarray (nD, akár 1D vagy 2D is)



Numpy objektum létrehozása különféle módon

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel
Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítő
feladatok
NumPy-al

Összefoglalás

- meghatározott tartalommal:
`np.array([[10, 21, 12], [32, 13, 17]])`
- számtani sorozatot tartalmazó vektor:
`np.arange(0.0, 1.1, 0.1)`
- csupa nullát tartalmazó mátrix:
`np.zeros((6, 10))`
- csupa egyest tartalmazó mátrix:
`np.ones((5, 8))`
- véletlen számokat tartalmazó mátrix:
`np.random.rand(4, 5)`
- egységmátrix:
`np.identity(4)`



Tömbök indexelése

Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás

- a listákhoz hasonlóan:

$v[3]$

$v[3:5]$

$v[2:9:3]$

$v[:4]$

$v[::2]$

item többdimenziós tömbök indexelése:

$m[1, 2]$

- indexelés logikai értékek tömbjével:

$v[v < 0]$

- indexelés indexek felsorolásával:

$v[(1, 2, 4),]$



Elemenként végzett műveletek

Kiegészítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegészítési
feladatok
NumPy-al

Összefoglalás

- a szokásos operátorok mind elemenként működnek
- a `numpy (np)` modulban találhatóak függvények is, amelyeket szintén elemenként tudunk használni
- a tömbök méretének egyeztetése:
 - Az elemenkénti műveleteket akkor lehet elvégezni, a tömbök mérete minden dimenzióban azonos,
 - vagy az egyik tömb esetében a méret 1.
 - Az utóbbi esetben ez az 1 méretű dimenzió felszorzódik a másik tömb megfelelő dimenziójához
 - az egyszerű értékeket minden dimenzióban 1 méretű tömböknek tekinthetjük



Mátrixok szorzása

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítősi
feladatok
NumPy-al

Összefoglalás

- mátrixok szorzása:
 - $A *$ operátor elemenkénti szorzást jelent!
 - A mátrix szorzást a $@$ operátorral lehet elérni.
- transzponált:
 - a `.transpose()` metódussal
 - vagy `.T` attribútumként
- determináns:
`np.linalg.det(x)`
- inverz:
`np.linalg.inv(x)`
- sajátvektor és sajátérték:
`np.linalg.eig(x)`
`np.linalg.eigvals(x)`



Példa: Helmert transzformáció paramétereinek számítása pontpárok alapján

A bemeneti adatok

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítési
feladatok
NumPy-al

Összefoglalás

- A közös pontok koordinátáit a `pt11` és a `pt12` listák (vagy kétdimenziós NumPy tömbök tartalmazzák)

```
import numpy as np
```

```
#Koordináták a kiinduló rendszerben
```

```
pt11=[(0,0),(0,1),(1,1),(1,0)]
```

```
#Koordináták a cél rendszerben
```

```
pt12=[(0.01,0.02),(0.3,0.99),(1.01,1.05),(1.3,0.4)]
```

```
#A pontok száma
```

```
npt=len(pt11)
```



Példa: Helmert transzformáció paramétereinek számítása pontpárok alapján

A kiegyenlítő számítás bemeneti mátrixainak felírása

- A kezdetben üres A és l mátrixok elemeinek feltöltése

#Az A , l és P mátrixok felírása

```
#           Y0  X0   a   b
prex=np.array([0.0, 0.0, 1.0, 0.0])
A=np.zeros([npt*2,4], dtype=np.double)
P=np.identity(npt*2, dtype=np.double)
l=np.zeros([npt*2,1], dtype=np.double)
for i in range(npt):
    A[2*i ][0] = 1.0
    A[2*i ][1] = 0.0
    A[2*i ][2] = ptl1[i][0]
    A[2*i ][3] = ptl1[i][1]
    A[2*i+1][0] = 0.0
    A[2*i+1][1] = 1.0
    A[2*i+1][2] = ptl1[i][1]
    A[2*i+1][3] = -ptl1[i][0]
    l[2*i ][0] = ptl2[i][0]-(prex[0]+prex[2]*ptl1[i][0]+prex[3]*ptl1[i][1])
    l[2*i+1][0] = ptl2[i][1]-(prex[1]-prex[3]*ptl1[i][0]+prex[2]*ptl1[i][1])
```

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatója

Kiegyenlítési
feladatok
NumPy-al

Összefoglalás



Példa: Helmert transzformáció paramétereinek számítása pontpárok alapján

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel
Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítési
feladatok
NumPy-al

Összefoglalás

- A NumPy eszközeivel a szükséges számítások egyszerűen elvégezhetőek

```
#A kiegyenlítő számítás elvégzése  
x=np.linalg.inv(A.T@P@A)@(A.T@P@l)  
v=A@x-l  
m0=np.sqrt((v.T@P@v)[0,0]/(2*npt-4))
```

```
#A Helmert transzformáció paramétereit egy tömbben  
ujx=prex+x.reshape((4,))
```



Összefoglalás

Kiegyenlítő
számítási
feladatok nyílt
forráskódú
eszközökkel

Nagy Gábor

Bevezetés

GNU Gama

A NumPy rövid
bemutatása

Kiegyenlítési
feladatok
NumPy-al

Összefoglalás

- a GNU GAMA többfél geodéziai hálózat kiegyenlítésére is alkalmas
- Ha (saját) Python programokkal akarunk kiegyenlítő számítási feladatokat megoldani, akkor a NumPy modult érdemes használni

Köszönöm a figyelmet!