

Optimalizálás

Oliver Kiss
Central European University

August 13, 2021

1 Optimalizálás

```
[1]: # Pythonban a scipy csomag kínál optimalizálási megoldásokat
```

```
[2]: # Csomagot importálni az import paranccsal lehet
```

```
[3]: # Példa: a math csomag  
import math
```

```
[4]: math.pi
```

```
[4]: 3.141592653589793
```

```
[5]: # Ha csak a csomag egy elemére van szükség, importálhatjuk azt külön
```

```
[6]: from math import pi
```

```
[7]: pi
```

```
[7]: 3.141592653589793
```

```
[8]: # Optimalizációhoz az fsolve függvényre lesz szükségünk  
from scipy.optimize import fsolve
```

```
[9]: # Az fsolve megkeresi azt az input értéket, amelyre a függvény eredménye nulla.␣  
    ↪Példa:  
def fun1(x):  
    return x+2  
  
# Az fsolve iteratív, szüksége van egy induló értékre ahonnan kezdve keres, ez␣  
    ↪a 2. argumentuma  
fsolve(fun1, 1)
```

```
[9]: array([-2.])
```

```
[10]: # Az fsolve valójában több eredményt is visszaadhat:  
x, infodict, flag, msg = fsolve(fun1, 1, full_output=True)
```

```
[11]: infodict
```

```
[11]: {'nfev': 4,  
      'fjac': array([-1.]),  
      'r': array([-1.]),  
      'qtf': array([6.54365451e-12]),  
      'fvec': array([0.])}
```

```
[12]: flag
```

```
[12]: 1
```

```
[13]: msg
```

```
[13]: 'The solution converged.'
```

```
[14]: # Ez akkor fontos, ha nincs megoldás. Pl:  
def fun2(x):  
    return x**2+1
```

```
[15]: fsolve(fun2, 1)
```

```
/home/kisso/.local/lib/python3.7/site-packages/scipy/optimize/minpack.py:175:  
RuntimeWarning: The iteration is not making good progress, as measured by the  
improvement from the last ten iterations.  
warnings.warn(msg, RuntimeWarning)
```

```
[15]: array([-2.18114415e-12])
```

```
[16]: fsolve(fun2, 1, full_output=True)
```

```
[16]: (array([-2.18114415e-12]),  
      {'nfev': 14,  
      'fjac': array([1.]),  
      'r': array([-0.0078125]),  
      'qtf': array([1.]),  
      'fvec': array([1.])},  
      5,  
      'The iteration is not making good progress, as measured by the \n improvement  
from the last ten iterations.')
```

```
[17]: # Kezdőérték számít. A háttérben egy gradiens alapú algoritmus fut, így  
      ↪ konvergál 0-hoz. Több megoldás esetén a kezdőértéken múlhat a kapott  
      ↪ megoldás.
```

```
[18]: def fun3(x):  
      return x**2-4
```

```
[19]: fsolve(fun3,-5)
```

```
[19]: array([-2.])
```

```
[20]: fsolve(fun3,5)
```

```
[20]: array([2.])
```

```
[21]: # Ha 0 gradiensű pontból indulunk, az gondot jelenthet  
      fsolve(fun3,0, full_output=True)
```

```
[21]: (array([0.]),  
      {'nfev': 13,  
       'fjac': array([[ -1.]]),  
       'r': array([6.50031613e-20]),  
       'qtf': array([4.]),  
       'fvec': array([-4.])},  
      5,  
      'The iteration is not making good progress, as measured by the \n improvement  
from the last ten iterations.')
```

```
[22]: # Lokális minimumok bajokat okozhatnak:  $x^3-3x^2-9x+40$   
      def fun4(x):  
          return x**3-3*x**2-9*x+40
```

```
[23]: fsolve(fun4, 5, full_output=True)
```

```
[23]: (array([2.99577456]),  
      {'nfev': 15,  
       'fjac': array([[ -1.]]),  
       'r': array([-0.00791365]),  
       'qtf': array([-13.00010705]),  
       'fvec': array([13.00010705])},  
      5,  
      'The iteration is not making good progress, as measured by the \n improvement  
from the last ten iterations.')
```

```
[24]: # Olyan kezdőértéket kell adni, ami várhatóan közel van a tényleges megoldáshoz.  
      fsolve(fun4, -5, full_output=True)
```

```
[24]: (array([-3.32495856]),  
      {'nfev': 9,  
       'fjac': array([[ -1.]]),  
       'r': array([-44.11586575]),
```

```
'qtf': array([5.8498685e-08]),  
'fvec': array([-9.9475983e-14])},  
1,  
'The solution converged.')
```

```
[25]: # Konvex problémák gradiense monoton, így jellemzően nincs probléma a  
      ↪ szélsőértékhelyek megtalálásával.
```

2 Egyenletrendszerek

```
[26]: # Az fsolve több input és output változót is kezel. Pl:  $x+y=5$ ,  $x^2+y^2=17$ 
```

```
[27]: def fun4(inputlist):  
      x = inputlist[0]  
      y = inputlist[1]  
      a = x+y-5  
      b = x**2+y**2-17  
      return [a, b]
```

```
[28]: fsolve(fun4, [3, 3])
```

```
[28]: array([1., 4.])
```

```
[29]: # Kezdőérték itt is számít:  
      fsolve(fun4, [0, 5])
```

```
[29]: array([1., 4.])
```

```
[30]: fsolve(fun4, [5, 0])
```

```
[30]: array([4., 1.])
```