In [1]:

```
%pylab inline
figsize(6,6) #Képméret megváltoztatása
```

```
Populating the interactive namespace from numpy and matplotlib
```

Sympy csomag használata analitikus problémák megoldására

Számos kereskedelmi (Mathematica, Maple) számítógépes algebrai csomag létezik már, melyek bonyolult analitikus formulamanipulációs készséggel rendelkeznek. A sympy csomag egy ingyenes python könyvtár ami lassan használható alternatívát kínál kereskedelmi vetélytársaival szemben. Ebben a notebookban néhány sympy függvénnyel fogunk megismerkedni.

!!! Fontos megjegyezni hogy mivel van pár a sympy függvényeihez hasonló nevű függvény a pylab parancs által importált csomagokban ezért a kellemetlenségek elkerülése végett célszerű mindig külön notebookot indítani a sympy-os problémák megoldásánál!!!

Konkrétabban: Mindkettő csomagban van plot parancs, és a rendszer az utoljára betöltött modul parancsát fogja használni. Azaz átdefiniáljuk a matplotlib csomag plot parancsát sympy plot-tá, ami marginálisan másként működik.

In [2]:

```
from sympy import * # a sympy csomag rutinjainak betöltése
```

Változók, egyenletek és egyenlet rendszerek megoldása

Ahhoz hogy változókat szimbolikusan is manipulálni tudjunk meg kell mondanunk a pythonnak hogy ezentúl tekintsen a változónkra mint valamilyen matematikai formulában előforduló szimbólumra. Ezt a legegyszerűbben így tehetjük meg:

In [3]:

```
x=symbols('x')
```

Ezek után az x változót használhatjuk szimbolikus számításokra. Oldjunk meg például a következő egyszerű egyenletet:

$$3x = 5$$

Ezt a solve függvény segítségével tehetjük meg. A solve függvény első bemente a megoldandó egyenlet 0-ra rendezve a második pedig a keresett változó:

In [4]:

```
solve(3*x-5,x)
```

Out[4]:

```
[5/3]
```

Definiáljunk néhány további változót is!

In [5]:

```
y,z,a,b,c=symbols('y,z,a,b,c') # Így definiálunk egyszerre több változót
```

Oldjuk meg most a

$$ax + b = y$$

egyenletet x -re!

In [8]:

```
solve(a*x+b-y,x)
```

Out[8]:

```
[(-b + y)/a]
```

Figyeljük meg hogy ha több megoldása is van az egyenletnek akkor lehetőség szerint mind a kettőt megtalálja:

In [9]:

```
solve(a*x**2+b*x+c,x)
```

Out[9]:

```
[(-b + sqrt(-4*a*c + b**2))/(2*a), -(b + sqrt(-4*a*c + b**2))/(2*a)]
```

Természetesen a megoldás nem feltétlenül valós szám! A komplex egységgyököt a sympy I-vel jelöli:

In [10]:

```
solve(x**2+1j,x)
```

Out[10]:

```
[-0.707106781186548 + 0.707106781186548*I,
 0.707106781186548 - 0.707106781186548*I]
```

A solve függvényt egyenletrendszerek megoldására is lehet használni. Ilyenkor a nullára rendezett egyenleteket listába foglaljuk. A keresett változókat szintúgy. Oldjuk meg a következő egyenletrendszert az xx és yy változókra:

$$y = x^2 + ax - 4b$$
$$y = x - b$$

Mivel xx -ben másodrendű az első egyenlet ezért két megoldáspárt várunk (A parabola legfeljebb két helyen metszi az egyenest...)

In [11]:

```
solve([x**2 + a*x -4*b - y, x - y - b],[x,y]) # Így kell egyenletrendszert megoldani
```

Out[11]:

```
[(-a/2 - sqrt(a**2 - 2*a + 12*b + 1)/2 + 1/2,
  -a/2 - b - sqrt(a**2 - 2*a + 12*b + 1)/2 + 1/2),
 (-a/2 + sqrt(a**2 - 2*a + 12*b + 1)/2 + 1/2,
  -a/2 - b + sqrt(a**2 - 2*a + 12*b + 1)/2 + 1/2)]
```

A `sympy` egyik legnagyobb előnye hogy a `python` nyelven belül lehetővé teszi egyszerűbb analízis beli feladatok elvégzését. Alább a teljesség igénye nélkül összefoglalunk néhány egyszerű függvényt. Vizsgáljuk meg a következő két határértéket:

$$\lim_{x\to 0}\frac{\sin x}{x}=?$$

illetve

$$\lim_{x\to\infty}\frac{1}{1+\mathrm{e}^{-x}}=?$$

A határértéket a `limit` függvény segítségével tudjuk meghatározni:

In [12]:

```
limit(sin(x)/x,x,0)# sin(x)/x határértéke az x=0 pontban.
```

Out[12]:

```
1
```

Ha a végtelenben vagyunk kíváncsiak a határértékre akkor azt az oo-szimbólummal tudjuk elérni

In [16]:

```
limit(1/(1+exp(-x)),x,oo)
```

Out[16]:

```
1
```

Egy kifejezés deriváltjait a diff függvény segítségével tudjuk meghatározni. Például a sinsin függvény első xx -szerinti deriváltja:

In [17]:

```
diff(sin(x),x)
```

Out[17]:

```
cos(x)
```

A második deriváltat vagy így

In [18]:

```
diff(sin(x),x,x)
```

Out[18]:

```
-sin(x)
```

vagy így, (talán egy kicsit átláthatóban) írjuk.

In [21]:

```
diff(sin(x),x,2)
```

Out[21]:

```
-sin(x)
```

Természetesen parciális deriváltak elvégzésére is van mód:

In [22]:

```
diff(sin(x)*cos(y),x,y)
```

Out[22]:

```
-sin(y)*cos(x)
```

A magasabb rendű parciális deriváltak legyártása az egyszerű deriváltak általánosításán alapszik:

In [23]:

```
diff(sin(x)*cos(y),x,2,y,3)
```

Out[23]:

```
-sin(x)*sin(y)
```

Az integrate függvény segítségével határozott és határozatlan integrálokat tudunk elvégezni. Határozzuk meg először az

$$x^2$$

primitív függvényét:

In [24]:

```
integrate(x**2,x) # x primitív függvénye
```

Out[24]:

```
x**3/3
```