



# Kiegészítő előadás

## Matlab 7.

(Szimbolikus számítások)

Dr. Szörényi Miklós,  
Dr. Kallós Gábor

2013–2014





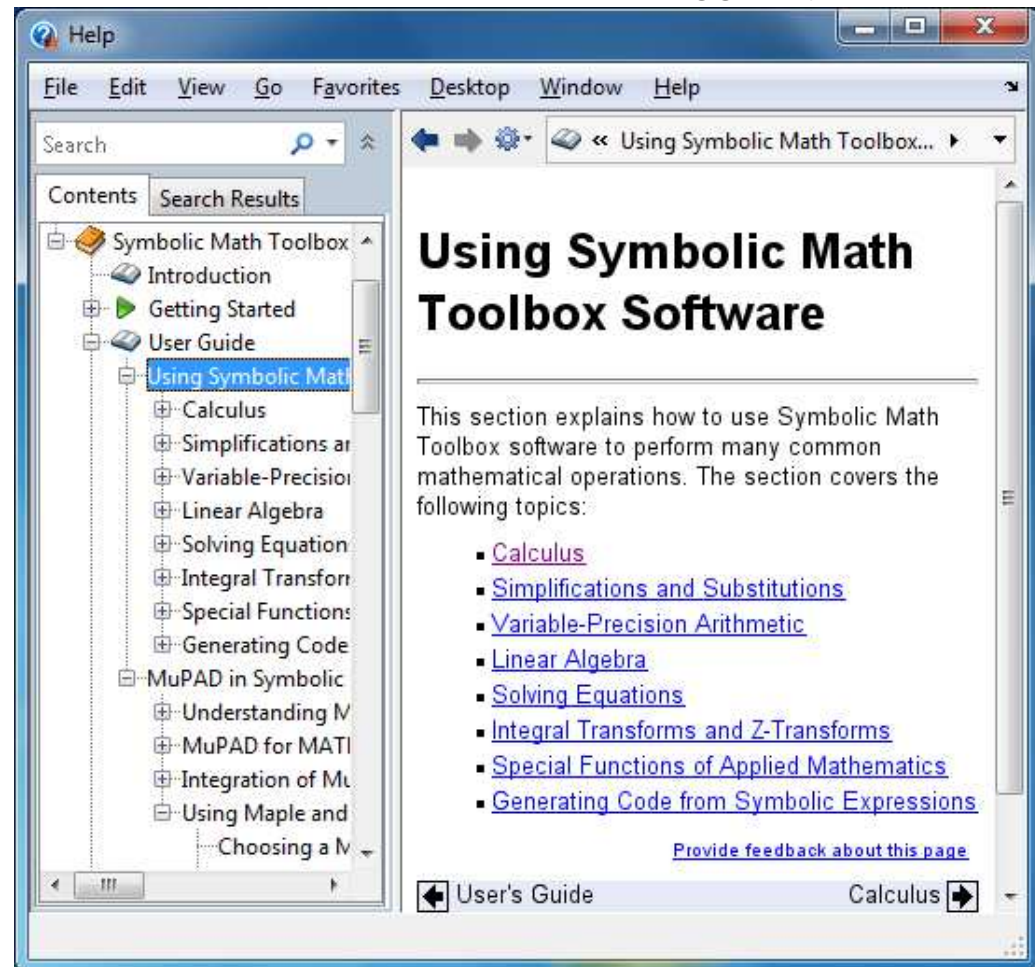
## Tartalom

- Symbolic Math Toolbox – áttekintés
- Szimbolikus változók és konstansok, szimbolikus kifejezések, mátrixok
- Nagy pontosságú aritmetika, egyenletek szimbolikus megoldása
- Szorzat vagy összeg alakú felírás?
- Egyszerűsítések
- Szimbolikus deriválás és integrálás
- Határértékek
- Taylor-közelítés



## Symbolic Math Toolbox – áttekintés

- A Matlab szimbolikus matematikai eszköztára (Symbolic Math Toolbox) több száz szimbolikus Matlab parancsot/függvényt tartalmaz
- A Toolbox fontosabb lehetőségei
  - Szimbolikus matematikai kifejezések kezelése (szimbolikus Matlab függvények pl. a deriválás, határérték számítás, integrálás, egyszerűsítés, egyenletmegoldás, diffegyenletek témakörében)
  - Speciális könyvtárak (parancsgyűjtemények) a fontosabb érintett területeken
  - Pontos aritmetikai (nagyon nagy pontosságú) számítások végrehajtásának lehetősége
  - Külön programozási nyelv (MuPAD) a szimb. objektumok hatékony manipulálására
  - Konverziós függvények a szimbolikus kifejezések átalakítására (Matlab, ill. C, MathML, TeX között)





## Szimbolikus változók és konstansok

### ■ Szimbolikus változók definiálása

- `x = sym('x')` vagy `syms` *változók felsorolása*
- A **syms** parancsot célszerű használni interaktív környezetben (kényelmesebb), a `sym` parancsot pedig scriptekben (biztonságosabb)

#### ■ Példa

```
>> syms x y % változók definiálása
>> (x-y)*(x-y)^2
>> expand(ans)
>> factor(ans) % polinom faktorizációja
```

### ■ Szimbolikus konstansok megadása

- A **sym** parancsot kell használni aposztrófok közé írással

#### ■ Példa

```
>> a = sym('1/3')
>> b = cos(sym('pi/2')) % pontos érték!
```

### ■ A Matlab a szimbolikus változókkal és konstansokkal pontos aritmetikát végez

```
>> sym('2/5') + sym('1/3')
```

- Az aritmetikai műveletek a szimbolikus objektumokkal is a megszokottak





## Szimbolikus kifejezések, mátrixok

### ■ Szimbolikus kifejezések

- Felépítjük őket az alkotóelemekből

- Példa

```
>> phi = sym('(1+sqrt(5))/2')  
>> f = phi^2 - phi - 1  
>> eval(f)      % vagy: simplify(f)
```

- Vigyázni kell arra, hogy mit tekintünk alkotóelemnek!

```
>> f = sym('a*x^2 + b*x + c')  
% vagy  
>> syms a b c x  
>> f = a*x^2 + b*x + c  
>> diff(f)      % most mindkét esetben működik
```

### ■ Szimbolikus mátrixok

- Szimbolikus elemekből felépíthetők, vagy már létező mátrixból konverzióval is létrehozhatók

- Példa

```
>> H4 = hilb(4)  
>> sym(H4)      % pontos értékek
```





## Nagypontosságú aritmetika

- Mint láttuk, szimbolikus kifejezésekkel pontosan (kerekítési hibák nélkül) lehet számolni

- `>> sin(sym('pi/3'))`

- A szimbolikus megadás lehetőséget teremt nagyon nagy pontosságú – numerikus – kiértékelésre is

- Megadhatjuk, hogy hány jegyre szeretnénk pontos eredményt kapni

- Példák – a **vpa** parancs (variable-precision arithmetic) használata

- `>> vpa('sqrt(2)',50)`

- `>> vpa('pi',1000)`

- A szimbolikus lehetőségeket egyenletek megoldásánál is felhasználhatjuk

- Példa

- `>> solve('x^2-2*x-4 = 0')`

- `% a numerikus megoldás ans-ból kinyerhető, vagy eltárolható:`

- `>> [x y] = solve('x^2-y = 2', 'y-2*x = 5')`

- `% az első példában [x] kellene, majd hiv-ként x(1) és x(2)`

- `>> solve('sin(x) = 2-x')`

- `% a Matlab itt nem tud pontos megoldást adni`

- Működik `fzero('x^2-2*x-4',0)` és `fzero(inline('x^2-2*x-4'),0)` is (inline megadás)





## Szorzat vagy összeg alakú felírás?

- Szimbolikus kifejezéseknél (pl. polinom) nem feltétlenül nyilvánvaló, hogy melyik alak tekinthető a legegyszerűbbnek
  - A feladat jellegétől is függ, hogy milyen felírássra van szükségünk
  - A probléma általánosan sem egyszerű
- Az **expand** parancs segítségével összeg formájú felírást kérhetünk, a **factor** parancssal pedig szorzattá alakítottat

```
>> syms x y
>> factor(x^3-y^3), factor(x^9-y^9)
>> factor(x^10-1)
>> f=(x-1)*(x+1)*(x^4+x^3+x^2+x+1)*(x^4-x^3+x^2-x+1)
>> expand(f)
```

- Numerikus kiértékelésre a Horner-alak a legmegfelelőbb

```
>> h = x^5+x^4+x^3+x^2+x
>> horner(h)
```

- Megj.: a factor parancs egész számokra is alkalmazható

```
>> factor(1111111111)
>> factor(111111111111) % túl nagy szám
>> factor(sym('111111111111')) % így lehet
```





## Egyszerűsítések

- A Matlab sokszor automatikus egyszerűsítés(eke)t végez, de gyakran ezt nekünk kell kérni (kényszerítés)
  - Ha nem kapjuk meg a céljainknak megfelelő alakot
- A **simplify** függvény erős, általános célú egyszerűsítő eszköz, amely számos algebrai azonosságot tud alkalmazni (pl. összegek, hatványok, gyökös és más kifejezések, függvényazonosságok stb.)
- Példák

```
>> f = (1-x^2)/(1-x)
>> simplify(f)
>> g = x^3-6*x^2+11*x-6
>> simplify(g)
>> h = exp(x)*exp(y)
>> simplify(h)
```
- A **simple** függvény többféle Matlabos egyszerűsítő függvényt használ (simplify, factor, collect, ...), és – kérésre – tájékoztatást ad az így elérhető eredményről
  - Végül a legegyszerűbb alakot adja vissza
- Példák

```
>> f = cos(3*acos(x))
>> simple(f)
>> f = simple(f)
```







## Szimbolikus deriválás, integrálás

- Szimbolikus kifejezések deriválása a `diff` paranccsal végezhető el
  - Megadható az a szimbolikus változó, amely szerint deriválni szeretnénk (többváltozós kifejezés esetén)
- Példák

```
>> diff(sin(x^2)), diff(sin(x)^2)
>> f = sin(x)^2+cos(y)^2
>> diff(f)           % ugyanaz, mint diff(f, x)
>> diff(f,y)
>> diff(diff(f,y),y)  % ugyanaz, mint diff(f,y,2)
>> diff(diff(f,y),x)
```
- Az integrálást (primitív függvény meghatározása) az `int` paranccsal tudjuk elvégezni
  - Itt is meg lehet adni a változót (többváltozós eset), ha nem az alapértelmezettet akarjuk használni
  - A legtöbb köznapi esetben elég gyorsan megkapjuk a – barátságos – választ
- Példák

```
>> int(sin(x)^2), int(-2*x/(1 + x^2)^2)
>> int(x^n + y^n)           % x szerint integrál
(lásd súgó: default symbolic variable)
>> int(x^n + y^n, y)        % y szerint
```





## Határozott integrál, határértékek

- A határozott integrál kiszámítása alapvetően nem a szimbolikus számítások témaköréhez tartozik, de van rá lehetőség ily módon is
  - Ekkor a `diff` parancsnak paraméterként a határokat is meg kell adnunk

- Példa

```
>> int(x^n + y^n, 1, 10)
>> int(x*log(1 + x), 0, 1)
```

- \*Megj.: a szimbolikus integrálás nehéz problémakör! Előfordulhat, hogy ...
  - Az integrál nem létezik zárt alakban (elméletileg sem határozható meg)
  - Az integrál létezik, de a Matlab nem tudja meghatározni
  - A Matlab meghatározza ugyan az integrált, de annak alakja számunkra furcsa
- A határérték-számítást a `limit` függvénnyel végezhetjük el
  - Alapértelmezés szerint a Matlab a 0 helyen számol
  - Lehetőség van bal- és jobboldali határérték meghatározására is (jó esetben a két határérték megegyezik)

- Példa

```
>> limit(sin(x)/x, 0)      % a 0 elhagyható
>> limit(1/x, x, 0)        % nem létezik
>> limit(1/x, x, 0, 'right'), limit(1/x, x, 0, 'left')
>> ezplot('1/x')          % ellenőrzés
>> limit((1 + 1/n)^n, Inf)  % nevezetes határérték
>> limit((sin(x + h) - sin(x))/h, h, 0)
% differenciálhányados
```



## Taylor-közelítés

- Ha a függvény (ill. deriváltjának) kiszámítása/kezelése egy adott helyen (ill. környékén) nem valósítható meg egyszerűen, akkor célszerű alkalmazni a Taylor-féle közelítést („csonkolt” hatványsor)
- A Matlab **taylor** parancsánál megadható a közelítés helye és pontossága
- Példa
 

```
>> f = sin(x)
>> taylor(f)
% ötödrendű közelítés
>> taylor(f, 10)
% kilencedrendű közelítés
% az objektumok színét beállítjuk
...
```
- Az ismert módon felrakjuk egy ábrára az eredeti függvényt és a közelítéseket
  - A tengelyek beállítását nem módosítjuk

The following expression present the Taylor series for an analytic function  $f(x)$  about the base point  $x=a$ :

$$f(x) = \sum_{m=0}^{\infty} (x-a)^m \cdot \frac{f^{(m)}(a)}{m!}$$

**Examples**

The following table describes the various uses of the `taylor` command and its relation to Taylor and Maclaurin series. Before using the `taylor` command, define the function you want to expand. For example:

```
syms x
f = exp(x^2);
```

Mathematical Operation	MATLAB Operation
$\sum_{m=0}^5 x^m \cdot \frac{f^{(m)}(0)}{m!}$	<code>taylor(f)</code>
$\sum_{m=0}^{n-1} x^m \cdot \frac{f^{(m)}(0)}{m!}$ $n$ is a positive integer	<code>taylor(f,n)</code> $n$ is a positive integer.
$\sum_{m=0}^5 (x-a)^m \cdot \frac{f^{(m)}(a)}{m!}$ $a$ is a real number	<code>taylor(f,a)</code> $a$ is a real number.



## Taylor-közelítés

- Az elkészült ábra, jelmagyarázattal

