



Matlab 5a. előadás

Numerikus deriválás és integrálás

Dr. Szörényi Miklós,
Dr. Kallós Gábor

2017–2018





Tartalom

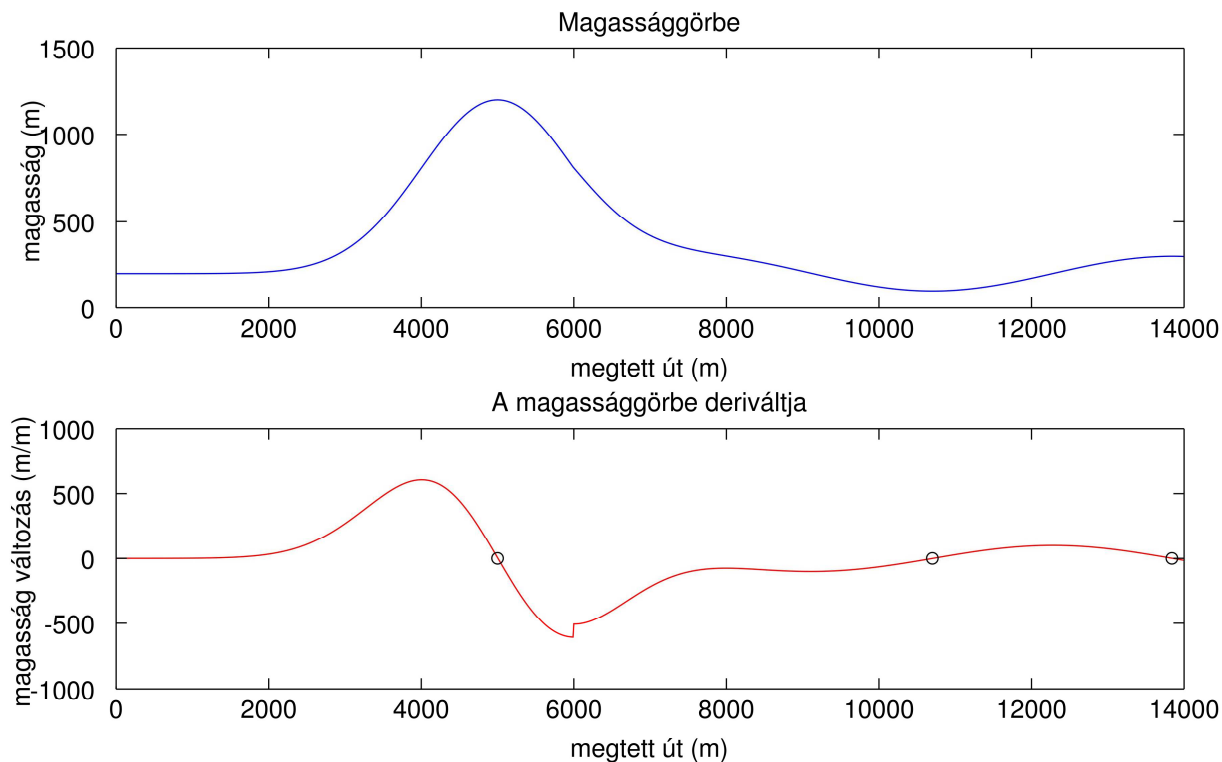
- Motiváció
- Numerikus deriválás
- Numerikus integrálás, függvény alatti terület
 - Lineáris módszerek
 - Magasabb fokú módszerek
 - Metszéspontok keresése
- Integrálfüggvény





Numerikus deriválás és integrálás

- Motiváció
 - Sokszor csak numerikusan tudjuk végrehajtani a deriválás és integrálás műveleteket
 - Ha olyan jellegű a függvény, ill. ha nincs szimbolikus toolboxunk
 - Ez sok fizikai és mérési jelenség vizsgálata/elemzése során szükséges lehet
- A műveletek végrehajtása ekkor diszkrét értékeken történik
 - A jeleket fel kell bontani

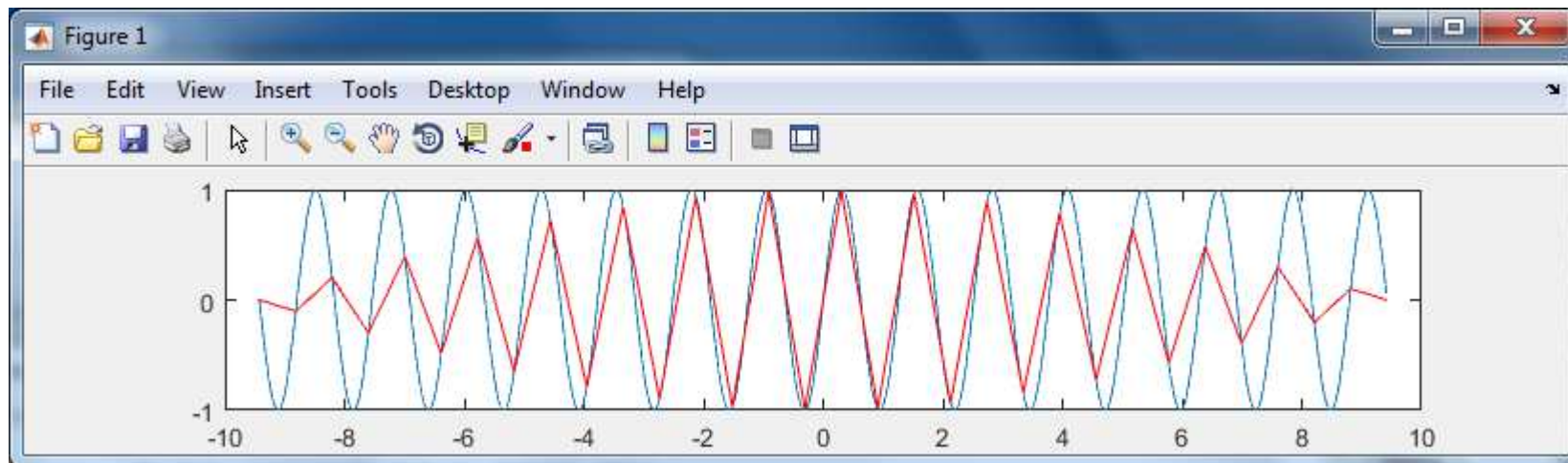




Numerikus deriválás

- Numerikus deriválás (diszkrét értékeken, elméleti összefoglaló)
 - A differenciálhatóság feltétele általában a függvény folytonossága
 - Kivétel pl. az abs fv. – gyors változás a meredekségben! (def.: szorgalmi Hf.)
 - A Matlabban sosincs teljes folytonosság (van egy min. lépésköz, eps), de jó sűrű felbontással legtöbbször numerikusan jó derivált számolható (*differenciahányados*)
 - Hibalehetőség: *alacsony felbontású mintavételezéssel* folytonos függvénynél sem kapunk „kellően” folytonos értékeket
 - Ilyenkor persze a fv. alakjával kapcsolatban is tévedhetünk
 - Példa

```
>> x=-3*pi:0.01:3*pi; x2=linspace(-3*pi,3*pi,32);  
>> plot(x, sin(5*x)), hold on, plot(x2, sin(5*x2), 'r')
```





Numerikus deriválás

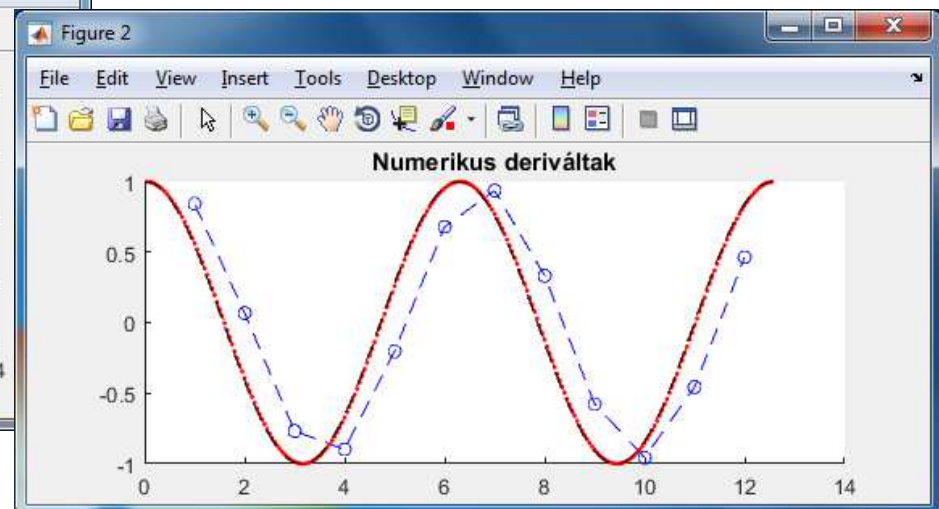
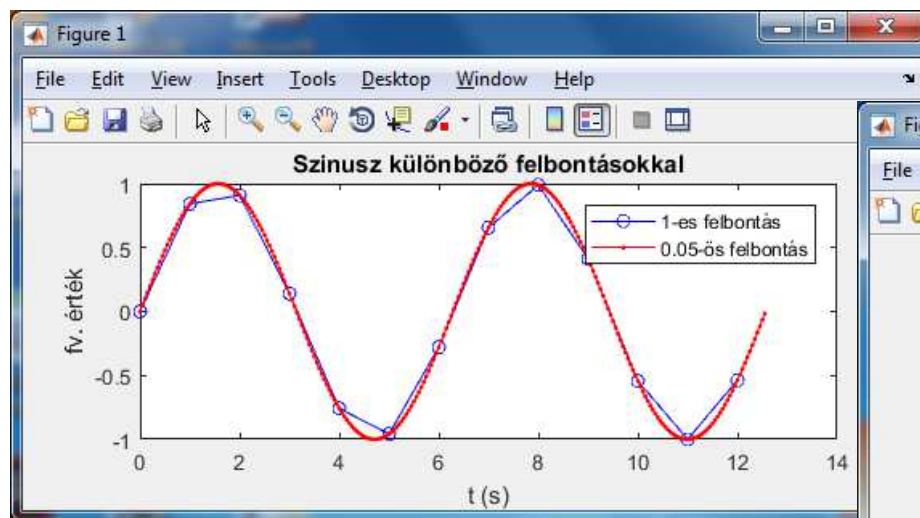
- Feladat (M. D.):
Rajzoljunk szinusz jelet a $[0, 4\pi]$ intervallum felett 1-es, majd 0,05-ös felbontással. Határozzuk meg és ábrázoljuk a görbék numerikus deriváltjait!
- Tudjuk: a derivált ilyenkor a differenciahányados (különbségek hányadosa)
 - Vektorok elemenként vett különbségeinek kiszámításához a *diff* függvény használható
- Megoldás
 - ```
>> t1 = 0:1:4*pi; t2 = 0:0.05:4*pi; s1 = sin(t1);
s2 = sin(t2);
% idővektorok és fv. értékek (alapjelek)
>> figure(1); plot(t1,s1,'bo-'); hold on;
plot(t2,s2,'r.-'); % kirajzolás
>> title('Szinusz különböző felbontásokkal');
xlabel('t (s)'); ylabel('fv. érték');
legend('1-es felbontás', '0.05-ös felbontás');
```



## Numerikus deriválás

- Megoldás (folyt., különbségek)

```
>> dt1 = diff(t1); dt2 = diff(t2);
ds1 = diff(s1); ds2 = diff(s2);
% értelmezési tartomány és értékkészlet
>> ds1_dt1 = ds1./dt1; ds2_dt2 = ds2./dt2;
% differencia hányados
>> figure(2); hold on;
plot(t1(2:end), ds1_dt1, 'bo--');
plot(t2(2:end), ds2_dt2, 'r.--'); % kirajzolás
plot(t2, cos(t2), 'k--'); % ellenőrzéshez, az első hibás!
title('Numerikus deriváltak');
```





## Numerikus integrálás

- A numerikus integrálás alapvető célja, tipikus alkalmazások
  - Függvény(grafikon) alatti terület, függvények közötti terület (különbségfüggvénnyel!) – intervallumon
  - Integrálfüggvény számítása (diszkrét értékekkel)
- A Matlabban a művelet vektorértékek és megadott függvény alapján is elvégezhető
  - Hasonlóan a numerikus deriváláshoz
- Az integrál a függvényértékek és az x-tengely közötti területrészek *előjeles* összege
  - De: a valóságban „negatív terület” nem létezik!
- Emiatt a függvények közötti területet célszerű metszéspontok között vizsgálni
  - Pozitív-negatív váltás kiküszöbölése
- Megj.: az integrálás sok esetben szimbolikusan is elvégezhető, szimbolikus toolboxszal (tudjuk már, ill. lásd szimbolikus óra)
  - Példa

```
>> syms a b c x; f = a*exp(-b*x)*sin(c*x)
>> int(f, x)
```





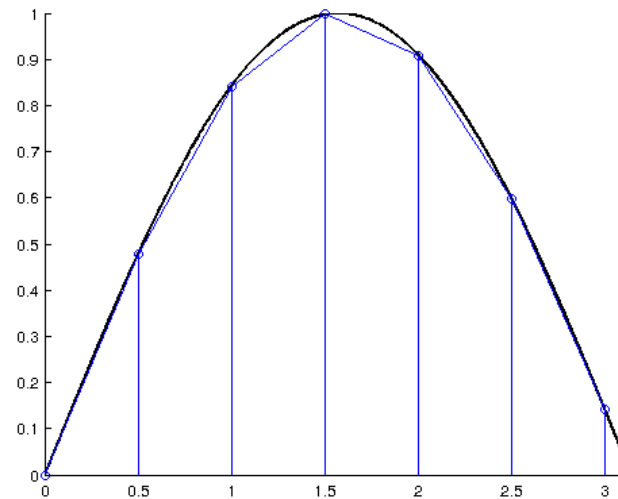
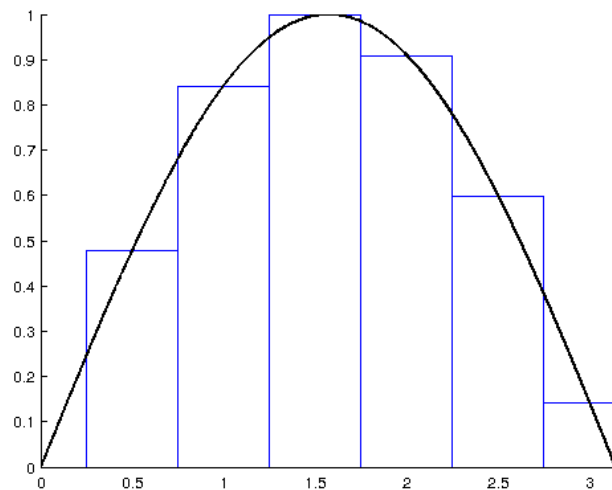
## Numerikus integrálás

- A számolásra használható beépített parancsok
  - Lineáris módszerek: *trapz* (trapézszabály)
    - A pontosságot az alappontok határozzák meg
  - Magasabb fokú módszerek: *quad*, *integral*
    - Alapértelmezett magas pontosság
  - Saját eljárás is írható, gyakorlásként (de a beépítettek jobbak)
- Trapézszabály

- Példa

```
>> res=0.01; x=0:0.01:pi;
>> z1=trapz(x,sin(x))
```

$$\int_a^b f(x) dx \approx \frac{b-a}{2N} \sum_{n=1}^N (f(x_n) + f(x_{n+1}))$$
$$= \frac{b-a}{2N} [f(x_1) + 2f(x_2) + \dots + 2f(x_N) + f(x_{N+1})]$$







## Numerikus integrálás

- *quad* parancs ('quadrature'): univerzális „jó eszköz”
- Határozott integrál számítása az adaptív, rekurzív Simpson szabállyal
  - Részletek: sűgő
- Szintaktika:  
*quad*(függvénymutató, alsó határ, felső határ, tolerancia)
  - A tolerancia (pontosság) paraméter elmaradhat
- A függvénymutató megadásának lehetőségei
  - Elérhető függvénynév  

```
>> quad('sin', 0, pi, eps) % elemi függvény
>> quad('f1', mxx, mnx)
% m-fájllal definiált, pontozott műveletek kellene!
```
  - Kiértékelhető karakterlánc  

```
>> quad('sin(x)+cos(x/2)', 0, pi)
```
  - Anonimus függvény (akár a *quad*-ban is megadható)  

```
>> F = @(x) 1./(x.^3-2*x-5)
>> Q = quad(F, 0, 2)
>> quad(@(x) log10(x)-cos(x), xd(1), xd(2))
>> quad(fd, xd(1), xd(2))
```





## Numerikus integrálás

Függvények közötti terület meghatározása

- Algoritmus (tudjuk már): különbségfüggvénnyel, metszéspontok keresésével
- A metszéspontok keresésére alkalmazható módszerek
  - Egyenlőség vizsgálata (nem jó, mert a véges felbontás miatt *általában* nem lesz tényleges egyenlőség, eml.: *fzero* indokoltsága... )
  - Különbségfv. abszolút értékek minimumhelyei
  - Különbségfv. zérushelyei
    - Már láttuk egy példában
- Illusztráció az első lehetőséghez
  - % sin és cos függvényekre

```
>> format long
>> egyenlohely=fzero(@(x) sin(x)-cos(x), 0.75)
 egyenlohely = 0.785398163397448
>> sin(egyenlohely)==cos(egyenlohely)
% látjuk, hogy nem azonosak!
>> sin(egyenlohely), cos(egyenlohely)
% pici eltérés
```

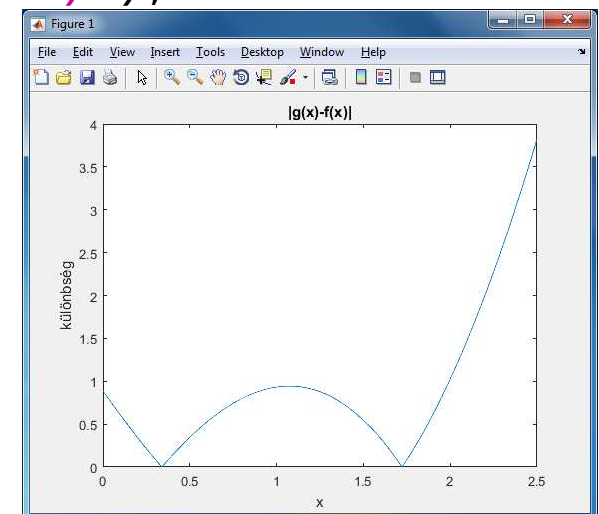
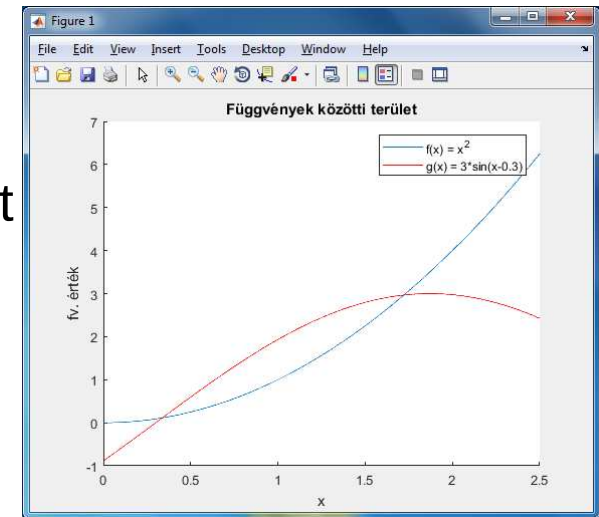


## Numerikus integrálás

### Függvények közötti terület meghatározása

- Metszéspontok keresése a különbségfv. abszolút értékének minimumhelyeivel
- Példánkban (M. D. alapján) 0,001-es lépésközt használunk

```
>> x = 0:0.001:2.5;
fx = x.^2; gx = 3*sin(x-0.3);
figure; hold on; plot(x,fx); plot(x,gx,'r');
>> title('Függvények közötti terület');
xlabel('x'); ylabel('fv. érték');
legend('f(x) = x^2', 'g(x) = 3*sin(x-0.3)');
% abszolút különbség
>> xdiff = abs(gx-fx);
figure; plot(x,xdiff);
title('|g(x)-f(x)|');
xlabel('x'); ylabel('különbség');
```





## Numerikus integrálás

Függvények közötti terület meghatározása (folyt.)

- Különbségfv. abszolút értékének min. helyei (folyt.)
  - % a kül.vektor értékeinek rendezése növekedően  
>> [xdiff\_sorted sorted\_ind] = sort(xdiff, 'ascend');  
% integr.i határ a rendezett indexek vektorának első két eleme  
>> minhelyek0 = sorted\_ind(1:2)  
minhelyek0 = 339 1724  
% integrálási intervallum indexei  
>> tr\_int = minhelyek0(1):minhelyek0(2);  
% vektorértékekből számolt intervallum alapján  
>> T1 = trapz(x(tr\_int), gx(tr\_int) - fx(tr\_int))  
>> T3 = quad(f, x(minhelyek0(1)), x(minhelyek0(2)))
- Különbségfv. zérushelyei
  - % a különbségfv. megadása anonim fv.-ként  
>> f = @(x) 3\*sin(x-0.3) - x.^2;  
% a fv. adott pont környékén található gyökének meghatározása  
>> minhelyek1 = [fzero(f, 0.5) fzero(f, 1.7)];  
>> T2 = quad(f, minhelyek1(1), minhelyek1(2))  
T2 = 0.8639





## Numerikus integrálás

### Integrálfüggvény

- Def.: Legyen  $f(x)$  integrálható fv.  $[a, b]$  felett.  $f(x)$  *integrálfüggvényének* nevezzük és  $F(x)$ -szel jelöljük azt az  $[a, b]$ -n értelmezett függvényt, amelyre

$$F(x) = \int_a^x f(t) dt$$

- Ha  $f(x)$  folytonos is  $[a, b]$ -n, akkor használhatjuk a Newton-Leibniz formulát
- Integrálfüggvény példa (M. D. alapján)

- % intervallum és lépésköz  
>> res = 0.001; x = 0:res:pi/2;  
% kumulált összeg trapézszabállyal, adott lépésközzel  
% az eredményvektor utolsó eleme adja meg az adott  
intervallumra az integrál értékét  
>> z4 = cumtrapz(sin(x))\*res;  
z4(end)  
% x határozza meg a lépésközt, nem kell szorozni  
>> z5 = cumtrapz(x, sin(x));  
z5(end)  
% ellenőrzés, csak végpontra! (Newton-Leibniz f.)  
>> -(cos(pi/2)-cos(0))





## Numerikus integrálás

### Integrálfüggvény

- Példa (M. D.): Adott egy vasúti szerelvény egyenes sínszakaszon mért (lineáris) sebességgörbéje. Adjuk meg a szerelvény (lineáris) pozícióját leíró görbét a teljes mérési szakaszon!
  - (Megj.: az útgörbe deriváltja a sebességgörbe)
  - % a szerelvény sebessége (mérési értékek)

```
>> t = linspace(0,100,1000);
v = 25*sin(2*pi*t*(1/150));
% a szerelvény pillanatnyi pozícióját v integrálfv.e
adja meg
% kumulált összeg trapézszabály alkalmazásával, t
határozza meg a lépésközt
>> Fv = cumtrapz(t,v);
% ellenőrzésként számítsuk ki Fv deriváltját
>> dFvdt = diff(Fv)./diff(t);
% eredmény kirajzoltatás
>> figure;
```



## Numerikus integrálás

### Integrálfüggvény

- Vasúti szerelvény sebességgörbéje (folyt.)

```

■ % első rajz
>> subplot(211);
% mért értékek
>> hold on;
plot(t,v, 'LineWidth',2);
% az integrálfv.-ből
visszaszámolt értékek
>> plot(t(2:end),dFvdt,...
 'r--', 'LineWidth',2);
>> plot([0 t(end)],[0 0], 'k--'); % vonal az x-tengelyre
>> title('Sebesség');
xlabel('t (s)'); ylabel('v (m/s)');
legend('v_{eredeti}', 'v_{számolt}');
subplot(212);
% a kiszámolt pozíció értékek
>> plot(t,Fv); title('Pillanatnyi pozíció');
xlabel('t (s)'); ylabel('s (m)');

```

