MATLAB 2017
5. gyakorlat

Polinomok, deriválás, integrálás, anonim függvények

Polinomok

- sok függvény és valós folyamat leírható magasabb rendű polinomokkal
- MATLAB-ban a polinomokat az együtthatóvektorukkal reprezentáljuk:
  - $P_1(x) = x^2 + 2x + 3$ → `P1 = [1 2 3];`
  - $P_2(x) = 10x^3 + 4x^2 + 5x - 7$ → `P2 = [10 4 5 -7];`
  - $P_3(x) = 3x^4 + 5x^2 - 12$ → `P3 = [3 0 5 0 -12];`

`P1`, `P2`, `P3` itt egy egyszerű sorvektor!

Polinomok

Az általunk megadott vektorokat a MATLAB megfelelő beépített függvényei fogják polinomként értelmezni:

- ✘ gyökök kiszámítása: `roots(P);`
- ✘ kiértékelés adott pontban: `y0 = polyval(P, x0);`
- ✘ kiértékelés sok pontban:
```
x = -1:0.01:1;
y = polyval(P, x);
```
- ✘ polinom létrehozása a gyökeiből:
```
r = [-1 1];
P = poly(r);
```
- ✘ polinom illesztése adatsorra:
```
P = polyfit(x_t, y_t, fokszam);
```

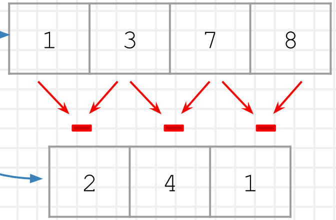
Numerikus deriválás

- **diszkrét értékeken** történik, ezért **fontos a jelek felbontása** (alacsony felbontással mintavételezve egy folytonos függvényt nem kapunk “kellőképpen” folytonos értékeket);
- a differenciálhatóság feltétele általában a függvény folytonossága (vannak persze kivételek);
- MATLAB-ban teljes folytonosságról nem beszélhetünk (minimális lépésköz `eps`), de megfelelő felbontást választva jó közelítéssel **numerikus**an is kiszámítható a **derivált**;
- **MATLAB-ban a derivált mindig differencia-hányados.**

Numerikus deriválás – differencia hányados számítása

- ✘ diff `(vektor);` % előállítja az elemenként vett különbségeket

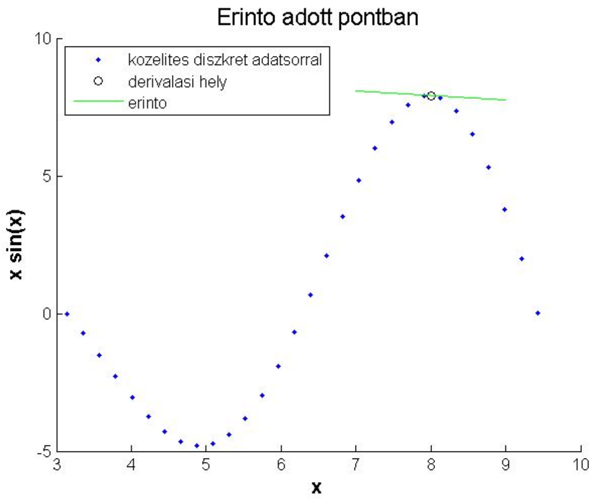
```
alma = [1 3 7 15];
diff(alma)
```



- ✘ állítsuk elő az értékkészlet és az értelmezési tartomány adatsorainak különbségét
  `dy = diff(y); dt = diff(t);`
- ✘ és ezeknek vegyük a hányadosát:
  `differenciahanyados = dy ./ dt;`
- ✘ megjegyzés: vegyük észre, hogy a különbségvektoroknak eggyel kevesebb eleme van: `length(t)` vs. `length(dt)` ezért ne is próbáljuk az értékkülönbségeket az eredeti ÉT-tartományhoz hozzárendelni pl kirajzoláskor: ~~`plot(t, dy, 'r.')`~~ helyett
  `plot(t(1:end-1), dy, 'r.')`

Deriválás adott pontban

- ✘ ha a vizsgált függvény képlettel felírható (pl. polinom, szögfüggvény, stb.), akkor adott hosszúságú és felbontású mintavétel nélkül is megadható egy adott pontban a derivált;
- ✘ ehhez a vizsgált pont tetszőlegesen kicsi környezetét megvizsgáljuk



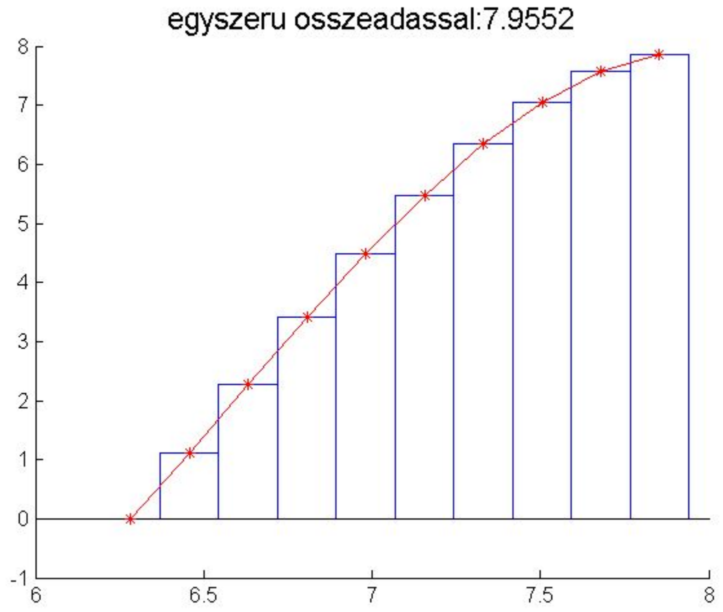

```
x = linspace(pi, 3*pi, 30); y = x .* sin(x);
x_p = 8; y_p = x_p*sin(x_p);
x_t = [8-1e-6, 8+1e-6]; y_t = x_t .* sin(x_t);
dx = diff(x_t); dy = diff(y_t);
differenciahanyados = dy/dx;

Figure; hold on;
plot(x, y, 'b.'); plot(x_p, y_p, 'ko');
plot([x_p-1, x_p+1], [y_p-differenciahanyados, ...
 y_p+differenciahanyados], 'g');
legend('kozelites diszkret adatsorral', 'derivalasi hely', ...
 'Erinto', 'Location', 'NorthWest');
xlabel('x', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('x sin(x)', 'FontSize', 12, 'FontWeight', 'bold');
title('Erinto adott pontban', 'FontSize', 14);
```

Numerikus integrálás

- ✘ a deriváláshoz hasonlóan lehet vektorértékek és megadott függvény alapján is integrálni *(integrál := a függvényértékek és az x-tengely közötti területrészek előjeles összege)*;
- ✘ lehetőségek:
  a. egyszerű összeadással,
  b. trapézszabály alkalmazásával,
  c. megadott függvény alapján

Numerikus integrálás --- a) egyszerű összeadással

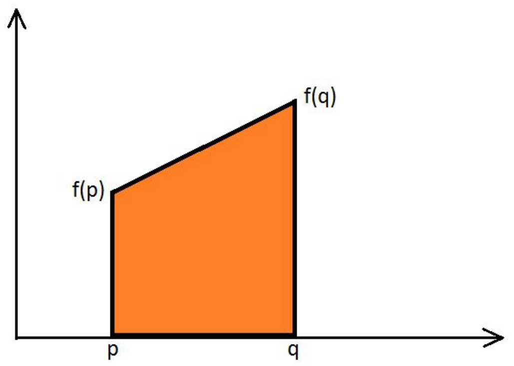
```
x = linspace(2*pi, 2.5*pi, 10);
y = x .* sin(x);
% egyszeru osszeadassal:
szelesseg = x(2)-x(1);
osszeg_1 = szelesseg*sum(y);

figure;
hold on;
bar(x, y, 1, 'w', 'EdgeColor', 'b', …
     'LineStyle', '-');
plot(x, y, 'r*-');
title(strcat('egyszeru osszeadassal: ', …
     num2str(osszeg_1, 5)), 'FontSize', 14);
```



Numerikus integrálás --- b) trapézszabállyal

egy trapéz területe:

$$T_{trapez} = (q-p)\frac{f(p)+f(q)}{2}$$



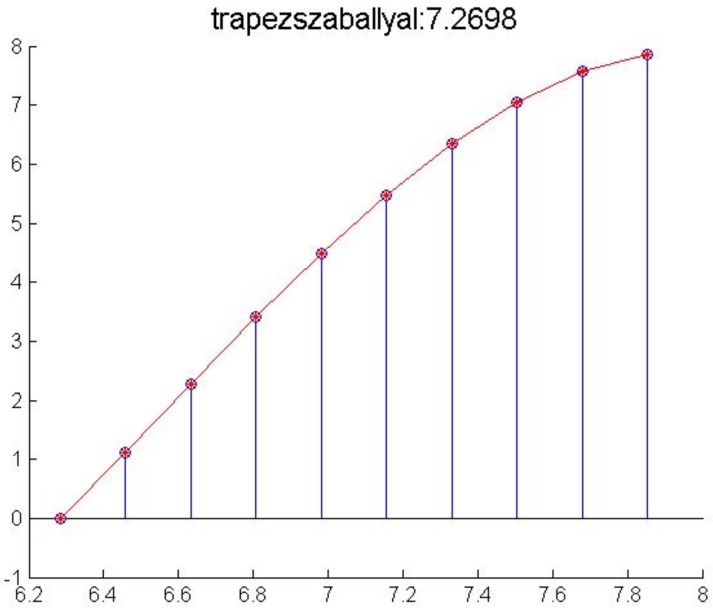
trapézszabály N+1 ekvidisztáns pont esetén:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2N}\sum_{n=1}^{N}\left(f(x_n)+f(x_{n+1})\right)$$

$$= \frac{b-a}{2N}\left[f(x_1)+2f(x_2)+\ldots+2f(x_N)+f(x_{N+1})\right]$$

Numerikus integrálás --- b) trapézszabállyal

```
x = linspace(2*pi, 2.5*pi, 10);
y = x .* sin(x);
% trapezszaballyal:
osszeg_2 = trapz(x, y);

figure;
hold on;
stem(x, y);
plot(x, y, 'r*-');
title(strcat('trapezszaballyal: ', …
    num2str(osszeg_2, 5)), 'FontSize', 14);
```



Anonim függvények

- a MATLAB lehetőséget ad arra, hogy függvényeket “tároljunk” változókban, (ha azok kellőképpen egyszerűek);
- a konstrukció:

```
fv = @(x) x+3;
```

függvénynév - változónév

bemenő pereméter(ek)től függő kifejezés, a függvény törzse

bemenő paraméter(ek)

```
>> fv = @(x) sin(x)-2*x^2+3*x
fv =
     @(x)sin(x)-2*x^2+3*x

>> fv(3)
ans =
  -8.8589
```

```
>> P = [2 0 3];
>> fv2 = @(x) polyval(P, x)
fv2 =
     @(x)polyval(P,x)

>> fv2(10)
ans =
   203
```

Numerikus integrálás --- c) függvénnyel

```
x = linspace(2*pi, 2.5*pi, 10);
fv = @(t) t .* sin(t);
% fuggvennyel:
osszeg_3 = integral(fv, x(1), x(end));

fprintf(['Elojeles osszegek:\n\tegyszeru osszeadassal: %6.4f\n\ttrapezszaballyal: ' ...
'%6.4f\n\tfuggvennyel: %6.4f\n'], osszeg_1, osszeg_2, osszeg_3);
```

```
Elojeles osszegek:
	egyszeru osszeadassal: 7.9552
	trapezszaballyal: 7.2698
	fuggvennyel: 7.2832
```

megjegyzés:
régebbi MATLAB-ban és/vagy linux-os kiadás alatt: **`integral`** *helyett* **`quad`**

Numerikus integrálás --- összegzés

- ✘ integrálásnál a **sima összeadást** lehetőleg **ne használjuk**;
- ✘ tetszőleges **vektoros adatsorokhoz: trapézszabály**;
- ✘ anonim **függvényekkel felírható görbékhez:** **`integral`** függvény.

Feladatok

- ✘ a feladatgyűjtemény **5.1 - 5.7** feladatai, melyeknek neve ez legyen, rendre:
  `gyak5_f51_[NEPTUN].m` …
  `gyak5_f57_[NEPTUN].m`
  (természetesen szögletes zárójelek nélkül).
- ✘ a diasorban ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (március 19.) éjfél.

Feltöltés: users.itk.ppke.hu/~zseta/matlab2017/HF05