

Polinomillesztés

Feladat: Illesszünk konstans függvényt, első-, valamint másodfokú polinomot a $(2, 1)$, $(4, 3)$, $(6, 2)$ pontokra a legkisebb négyzetek módszerével! (A másodfokú polinom természetesen interpolációs polinom lesz.)

A feladat megoldására a Matlab beépített `polyfit` függvénye használható.

`p = polyfit(x,y,n)` megadja annak az n -edfokú polinomnak az együtthatóit, amely legkisebb négyzetes értelemben a legjobban illeszkedik az x és az y vektor által meghatározott pontokra (az x vektor a pontok x -koordinátáiból áll, y az y -koordinátáiból). A `p` egy vektor lesz, amely a polinom együtthatóit tartalmazza a legnagyobb kitevőjű tagéval kezdve. Ezen polinom értékeit valamely `xx` vektor pontjaiban a `polyval(p,xx)` függvény segítségével számíthatjuk ki.

lnm.m

```
%legkisebb negyzetek modszere
```

```
function lnm(n)
```

```
x = [2 4 6];
```

```
y = [1 3 2];
```

```
%Erre a három pontra illesztünk konstans, linearis, vagy masodfoku
```

```
%fuggvenyt legkisebb negyzetes ertelemben.
```

```
switch n
```

```
    case {0,1,2}
```

```
        p = polyfit(x,y,n);
```

```
    otherwise
```

```
        disp('n = 0, 1 vagy 2 lehet csak!')
```

```
        return
```

```
end
```

```
%Lehetett volna if-fel is...
```

```
xx = linspace(min(x)-1,max(x)+1,10*(max(x)-min(x)+2)+1);
```

```
yy = polyval(p, xx);
```

```
plot(xx,yy,'LineWidth',2)
```

```
plot(x,y,'ro','MarkerEdgeColor','r','MarkerFaceColor','r','MarkerSize',8)
```

Az elágazásban az `otherwise` után beszúrt `return` hatására a Matlab kilép a programból, tehát az ezt követő sorok, amelyeknek ekkor nincs értelmük, nem hajtódnak végre. A `plot` utasításnál a `'ro'` azt jelenti, hogy piros karikák legyenek a pontok helyén (`r` = red, `o`

= karika), utána pedig megadható a karikák körvonalának és belsejének a színe, valamint a karikák mérete. (Lásd még a `plot` help menüjét.)