

## ▼ 7. Interpoláció elmélet: Lagrange módszer, spline interpoláció

Ebben a fejezetben célunk annak megismerése, hogyan lehet egy ismeretlen függvény alakjának egy  $f(x)$  függvénnyel való minél pontosabb megközelítése úgy, hogy a közelítő függvény áthaladjon az ismert, táblázatba szedett pontokon. Az ilyen típusú feladatok fontos szerepet játszanak a számítógépes grafikában, az építészetben, valamint a numerikus integrálásnál és differenciálegyenletek numerikus megoldásánál is.

Ehhez először az alapdefiníciókat fogalmazzuk meg.

### ▼ 7. 1. Definíciók

**Approximáció:** közelítés; ismeretlen mennyiségnek közelítő pontossággal történő meghatározására szolgáló eljárás. □

**Interpoláció:** egy függvény nem ismert értékeinek ismert értékek alapján történő becslése. □

**Extrapoláció:** a függvény értékeinek közelítő meghatározása egy adott pontban a megelőző szakasz értékei alapján. □

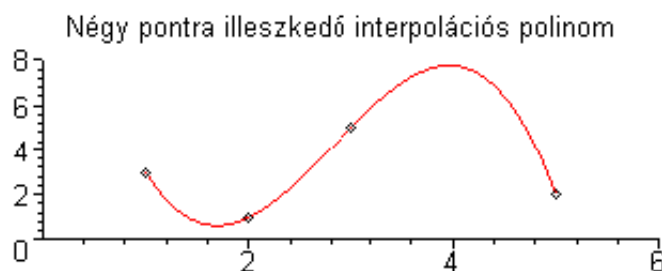
Tekintsük a mérési adatok összetartozó  $(x,y)$  értékpárjainak következő táblázatát

|     |       |       |  |       |
|-----|-------|-------|--|-------|
| $x$ | $x_0$ | $x_1$ |  | $x_n$ |
| $y$ | $y_0$ | $y_1$ |  | $y_n$ |

ahol feltesszük, hogy minden  $x_i$  érték különböző, valamint összesen  $n+1$  mérést végeztünk. Keressünk olyan egyszerűen és gyorsan kiszámítható  $y = f(x)$  függvény kapcsolatot az  $x$ - $y$  között, amely az adott  $(x_i, y_i)$  pontok mindegyikére illeszkedik, azaz

$$y_i = f(x_i) \text{ minden } i = 0, 1, 2, 3, \dots, n \text{ esetén.}$$

Rajzoljuk fel az  $(x_i, y_i)$  pontokat a Descartes-féle derékszögű koordináarendszerben és rajzoljunk a pontokra illeszkedő függvényt (az ábra  $n = 4$  alappontot mutat)!



Mivel egy

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0$$

polinom értéke adott  $x$  esetén könnyen kiszámítható, ezért célszerű a feladat megoldását polinomok között keresni. A fenti  $n$ -edfokú polinomot egyértelműen megadhatjuk az  $a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0$   $n+1$  darab együttható segítségével. Az  $n+1$  ismeretlen együttható egyértelmű meghatározásához  $n+1$

feltétel szükséges. Ezért a feladat megoldását  $n+1$  pont esetén az  $n$ -edfokú polinomok között kell keresnünk.

## ▼ 7. 2. Interpoláció kettő illetve három pontra

Használt Maple parancsok: CurveFitting csomag, PolynomialInterpolation, op, nops, product

Tekintsük a legegyszerűbb  $n+1=3$  esetet, valamint tételezzük fel, hogy a  $P_0(x_0; y_0)$ ,  $P_1(x_1; y_1)$ ,  $P_2(x_2; y_2)$  pontok nincsenek egy egyenesen. Ekkor a keresett polinom másodfokú,  $y = a \cdot x^2 + b \cdot x + c$  alakú lesz, a megoldandó egyenletrendszer

$$\begin{aligned} &> e1 := \text{subs}(\{x=x_0, y=y_0\}, y=a x^2 + b x + c); \\ &e2 := \text{subs}(\{x=x_1, y=y_1\}, y=a x^2 + b x + c); e3 := \text{subs}(\{x=x_2, y=y_2\}, y=a x^2 + b x \\ &+ c) \end{aligned}$$

$$e1 := y_0 = a x_0^2 + b x_0 + c$$

$$e2 := y_1 = a x_1^2 + b x_1 + c$$

$$e3 := y_2 = a x_2^2 + b x_2 + c$$

$$> \text{solve}([e1, e2, e3], \{a, b, c\})$$

$$\begin{aligned} \left\{ \begin{aligned} a &= - \frac{-x_0 y_2 + x_2 y_0 + x_1 y_2 + x_0 y_1 - x_1 y_0 - x_2 y_1}{x_0 x_2^2 - x_1 x_2^2 + x_1 x_0^2 - x_2 x_0^2 + x_2 x_1^2 - x_0 x_1^2}, b \\ &= \frac{-x_0^2 y_2 + x_0^2 y_1 + y_0 x_2^2 - y_1 x_2^2 + x_1^2 y_2 - y_0 x_1^2}{(x_0 - x_1) (-x_0 x_2 + x_0 x_1 + x_2^2 - x_1 x_2)}, c \\ &= \frac{x_0^2 x_1 y_2 - x_0^2 x_2 y_1 - x_1^2 x_0 y_2 + y_1 x_0 x_2^2 + x_1^2 x_2 y_0 - y_0 x_1 x_2^2}{(x_0 - x_1) (-x_0 x_2 + x_0 x_1 + x_2^2 - x_1 x_2)} \end{aligned} \right\} \end{aligned}$$

Ha  $n$  ponton átmenő polinomot keresünk, egy  $n \times n$ -es szabályos lineáris egyenletrendszert kell megoldanunk. Ennek elvi akadályja nincs, de célszerű az interpolációs polinom együtthatóit valamilyen jól áttekinthető formában keresni.

## ▼ 7. 3. Általános eset Lagrange-módszerrel

Keressük azt a legfeljebb  $(n-1)$ -ed fokú  $p(x)$  polinomot, amelyre a  $p(x_1) = y_1, p(x_2) = y_2, \dots, p(x_n) = y_n$  egyenlőségek teljesülnek, ahol feltehetjük, hogy az  $x_1 < x_2 < \dots < x_{n-1} < x_n$  alappontok növekvően rendezettek. (A könnyebb kezelhetőség miatt az első pont koordinátái  $1$ -es, az  $n$ -edik pont koordinátái az  $n$  indexet kapják.)

Tekintsük az olyan  $l_k(n-1)$ -edfokú polinomokat  $k=1, 2, 3, \dots, n$  esetén, amelyek  $x=x_k$ -ra  $1$ -et,  $x \neq x_k$  0-át adnak eredményül ( $\text{product}(\text{lista}, i=\text{kezdőérték}..\text{végérték})$ ).

$$> l[k] := x \rightarrow \prod_{i=1}^{k-1} \left( \frac{x - x_i}{x_k - x_i} \right) \prod_{i=k+1}^n \left( \frac{x - x_i}{x_k - x_i} \right);$$

Vizsgáljuk ezt a szorzatot  $n=3, k=2$  esetre (3 ponton illeszkedő eset, a 2. ponthoz tartozó  $l_2$  polinom). Helyettesítsünk  $x$  helyére

$x_2, x_1, x_3$  értékeket.

>  $l2 := \text{factor}(\text{expand}(\text{subs}(\{k=2, n=3\}, l[k](x))))$

$$l2 := - \frac{(x - x_3)(x - x_1)}{(-x_2 + x_1)(-x_3 + x_2)}$$

>  $\text{simplify}(\text{subs}(x=x_2, l2))$

1

>  $\text{simplify}(\text{subs}(x=x_1, l2)); \text{simplify}(\text{subs}(x=x_3, l2))$

0

0

Általánosan is igazolható, hogy teljesülnek a polinommal szemben támasztott követelmények.

Az  $l_k(x)$  bázispolinomok segítségével előállítjuk az interpolációs polinomot, amely megfelel a  $p(x)$  polinommal szemben támasztott feltételeknek.

>  $p := x \rightarrow \sum_{k=1}^n y_k l[k](x) :$

$l_k$  előzőekben belátott tulajdonságai miatt  $p(x_k) = y_k$  minden  $k=1, 2, 3, \dots, n$  esetén, valamint fokszáma pontosan  $(n-1)$ .

### ▼ 7. 3. 1. Definíció

A  $p := x \rightarrow \sum_{k=1}^n y_k l_k(x)$  polinomot Lagrange-féle interpolációs polinomnak nevezzük, ahol

$$l_k(x) = \prod_{i=1}^{k-1} \left( \frac{x - x_i}{x_k - x_i} \right) \prod_{i=k+1}^n \left( \frac{x - x_i}{x_k - x_i} \right) \square$$

## ▼ 7. 4. Az interpolációs polinom Newton-féle alakja

A következő megfontolásnál feltételezzük, hogy az ismert pontok azonos lépésközzel követik egymást. Most nem egyszerre vizsgáljuk az adott pontokat, hanem először az első két pontot vizsgáljuk, majd ehhez vesszük hozzá egyesével a többi.

Induljunk ki a  $P_1(x_1; y_1)$  és  $P_2(x_2; y_2)$  pontokon átmenő egyenes egyenletéből:

>  $\text{restart}; P1 := [x_1, y_1] : P2 := [x_2, y_2] : p[1] := x \rightarrow y_1 + \frac{(y_2 - y_1)(x - x_1)}{x_2 - x_1} :$

Vezessük be a következő jelölést:

>  $h = x_2 - x_1 : \Delta(y) = y_2 - y_1 :$

Ekkor  $p_1(x)$  a következő alakú:

>  $p[1] := x \rightarrow y_1 + \frac{\Delta(y)(x - x_1)}{h} :$

Vegyünk  $P_1(x_1; y_1)$  és  $P_2(x_2; y_2)$  alappontok mellé még egy 3.  $P_3(x_3; y_3)$  pontot is úgy hogy  $x_3 - x_2 = h$  legyen (egyenletes osztáspontok, **véges differenciák** módszere).

Ekkor a pontok egy másodfokú függvény görbéjén helyezkednek el, vagyis

$$> \quad p[2] := y_0 + \frac{\Delta(y) (x - x_0)}{h} + g(x) :$$

ahol  $g(x)$ -nek egy másodfokú függvénynek kell lennie. Mivel  $p_1(x_1) = y_1, p_2(x_1) = y_1$ , valamint  $p_1(x_2) = y_2, p_2(x_2) = y_2$  ezért  $g(x_1) = 0, g(x_2) = 0$  valamint  $g(x_3) = y_3$ . Tehát

$$> \quad g := x \rightarrow C (x - x_1) (x - x_2) :$$

alakú kell legyen, ahol  $C$  állandó. Mivel  $p_2(x_3) = y_3$

$$> \quad e := y_3 = y_1 + \frac{\Delta(y) (x_3 - x_1)}{h} + C (x_3 - x_1) (x_3 - x_2) :$$

Ha egyenletes a beosztás,

$$> \quad x_3 - x_1 = 2 h : x_3 - x_2 = h :$$

$$> \quad \text{subs}(\{\Delta(y) = y_2 - y_1, x_3 - x_1 = 2 h, x_3 - x_2 = h\}, \text{\% \% \%})$$

$$y_3 = -y_1 + 2 y_2 + 2 C h^2$$

$$> \quad C := \text{solve}(\text{\%}, C)$$

$$C := \frac{1}{2} \frac{y_1 - 2 y_2 + y_3}{h^2}$$

ahol a számláló az un. másodrendű osztott differencia, jele  $\Delta^2 y$  így

$$> \quad p[2] := x \rightarrow y_1 + \frac{\Delta(y) (x - x_1)}{h} + \frac{\Delta^2 y (x - x_1) (x - x_2)}{2 h^2} :$$

Általános esetre is igaz, hogy

$$> \quad p[n] := x \rightarrow y_1 + \sum_{k=1}^{n-1} \frac{\Delta^k y \prod_{i=1}^k (x - x_i)}{k! h^k} :$$

A módszer előnye, hogy további pontokat lehet a már adottakhoz hozzávenni.

Ha nem ekvidisztáns (egyenletes) a független változók ( $x$ -ek) választása, a véges differenciák és a  $h$  helyett a kettő hányadosát, az. un. osztott differenciákat kell használni.

## ▼ 7. 5. Interpoláció Maple-lel

A Maple-ben a *CurveFitting* csomagja tartalmazza a pontokra görbét illesztő numerikus parancsokat. A *PolynomialInterpolation* nevű eljárás az interpolációs polinomot adja vissza.

Határozzuk meg az interpolációs polinomot 3 adott pont esetén.

$$> \quad \text{restart}; \text{with}(\text{CurveFitting}) : \text{with}(\text{plots}) : P[1] := [x_1, y_1] : P[2] := [x_2, y_2] :$$

$$P[3] := [x_3, y_3] :$$

$$> \quad X := [op(1, P_1), op(1, P_2), op(1, P_3)]$$

$$> \quad Y := [op(2, P_1), op(2, P_2), op(2, P_3)]$$

$$X := [x_1, x_2, x_3]$$

$$Y := [y_1, y_2, y_3]$$

A *PolynomialInterpolation* parancs három formában, *Lagrange*, *Newton* és *power* (hatvány)

alakban adjuk meg az interpolációs polinomot.

> *polinomLagrange* := *PolynomialInterpolation*(*X*, *Y*, *x*, *form* = *Lagrange*)

$$\text{polinomLagrange} := \frac{y_1 (x - x_2) (x - x_3)}{(x_1 - x_2) (x_1 - x_3)} + \frac{y_2 (x - x_1) (x - x_3)}{(x_2 - x_1) (x_2 - x_3)} + \frac{y_3 (x - x_1) (x - x_2)}{(x_3 - x_1) (x_3 - x_2)}$$

> *polinomNewton* := *PolynomialInterpolation*(*X*, *Y*, *x*, *form* = *Newton*)

$$\text{polinomNewton} := \left( \frac{\left( y_3 - \frac{(y_2 - y_1) (x_3 - x_1)}{x_2 - x_1} - y_1 \right) (x - x_2)}{(x_3 - x_1) (x_3 - x_2)} + \frac{y_2 - y_1}{x_2 - x_1} \right) (x - x_1) + y_1$$

> *polinomhatvany* := *PolynomialInterpolation*(*X*, *Y*, *x*, *form* = *power*)

$$\begin{aligned} \text{polinomhatvany} := & - \frac{(-y_3 x_1 + y_3 x_2 + y_2 x_1 - y_2 x_3 + y_1 x_3 - y_1 x_2) x^2}{(x_1 - x_2) (x_1 - x_3) (x_2 - x_3)} \\ & - \frac{(y_3 x_1^2 - y_2 x_1^2 + y_2 x_3^2 - y_3 x_2^2 + y_1 x_2^2 - y_1 x_3^2) x}{(x_1 - x_2) (x_1 - x_3) (x_2 - x_3)} \\ & - \frac{y_1 x_2 x_3^2 - y_2 x_3^2 x_1 - y_3 x_1^2 x_2 - y_1 x_2^2 x_3 + y_2 x_1^2 x_3 + y_3 x_2^2 x_1}{(x_1 - x_2) (x_1 - x_3) (x_2 - x_3)} \end{aligned}$$

Ellenőrizzük a kapott kifejezések ekvivalenciáját.

> *simplify*(*polinomLagrange* - *polinomNewton*)

0

> *simplify*(*polinomLagrange* - *polinomhatvany*)

0

**1. Példa** Adjuk meg a következő 5 pontra illeszkedő interpolációs polinomot, majd ábrázoljuk a pontokat és a kapott polinomot egy közös koordináta rendszerben.

>  $P[1] := [1, 3] : P[2] := [2, 1] : P[3] := [3, 5] : P[4] := [5, 2] : P[5] := [7, 3] :$

**Megoldás:**

> *with*(*CurveFitting*) : *with*(*plots*) :

>  $X := [op(1, P_1), op(1, P_2), op(1, P_3), op(1, P_4), op(1, P_5)]$

>  $Y := [op(2, P_1), op(2, P_2), op(2, P_3), op(1, P_4), op(1, P_5)]$

$X := [1, 2, 3, 5, 7]$

$Y := [3, 1, 5, 5, 7]$

A későbbiekben hasznos lehet, ha a *nops*(*X*) paranccsal megszámoljuk az *X* halmaz elemeinek számát.

>  $n := nops(X);$

$n := 5$

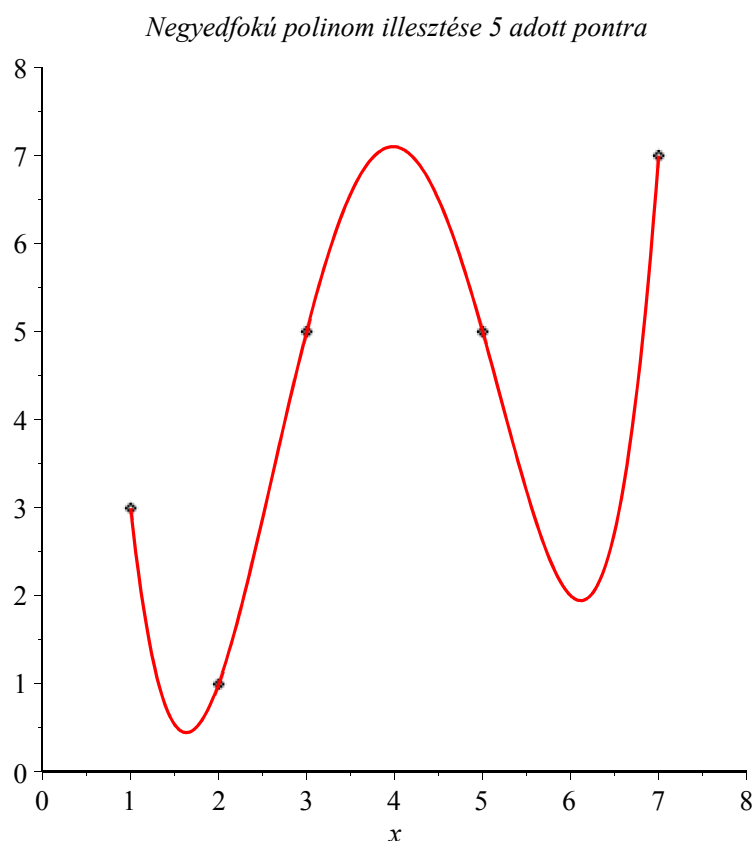
> *polinomLagrange* := *PolynomialInterpolation*(*X*, *Y*, *x*, *form* = *Lagrange*)

$$\begin{aligned} \text{polinomLagrange} := & \frac{1}{16} (x - 2) (x - 3) (x - 5) (x - 7) - \frac{1}{15} (x - 1) (x - 3) (x - 5) (x \\ & - 7) + \frac{5}{16} (x - 1) (x - 2) (x - 5) (x - 7) - \frac{5}{48} (x - 1) (x - 2) (x - 3) (x - 7) \end{aligned}$$

$$+ \frac{7}{240} (x-1)(x-2)(x-3)(x-5)$$

Az ábrázoláskor, a pontok koordinátáinak beolvasásához használjuk a  $\text{seq}([X[i], Y[i]], i=1..n)$  parancsot.

- >  $\text{pontok} := \text{plot}([\text{seq}([X_i, Y_i], i=1..n)], \text{style}=\text{point}, \text{symbol}=\text{diamond}, \text{color}=\text{black})$  :
- $\text{graf} := \text{plot}(\text{PolynomialInterpolation}(X, Y, x), x=X_1..X_n, \text{color}=\text{red})$  :
- >  $\text{plots}[\text{display}](\text{[pontok, graf]}, \text{view}=[0..8, 0..8], \text{scaling}=\text{constrained}, \text{title}=\text{Negyedfokú polinom illesztése 5 adott pontra})$



**2. Példa** A  $\sqrt{100} = 10$ ,  $\sqrt{121} = 11$ ,  $\sqrt{144} = 12$  ismert értékek felhasználásával és az interpolációs polinom alapján számítsuk ki a  $\sqrt{110}$  közelítő értékét!

a; Mekkora hibát követtünk el?

b; A  $[100, 144]$  intervallumon mennyi a maximális hiba a másodfokú interpolációs polinom és a  $\sqrt{x}$  függvény között?

**Megoldás a;**

Mivel 3 alappont van, ezért a közelítés másodfokú polinommal lehetséges.

>  $X := [100, 121, 144]$  :  $Y := [10, 11, 12]$  :

>  $\text{parabola} := \text{PolynomialInterpolation}(X, Y, x, \text{form}=\text{power});$

$$parabola := -\frac{1}{10626} x^2 + \frac{727}{10626} x + \frac{660}{161}$$

A  $\sqrt{110}$  közelítő értékét megkapjuk, ha a polinomba  $x$  helyére 110-et helyettesítünk.

> `evalf(subs(x=110, parabola))`

10.48654244

A pontos érték és az interpolált érték eltérése, vagyis az elkövetett hiba

> `hiba := abs(evalf( $\sqrt{110}$  - subs(x=110, parabola)))`  
`hiba := 0.00154604`

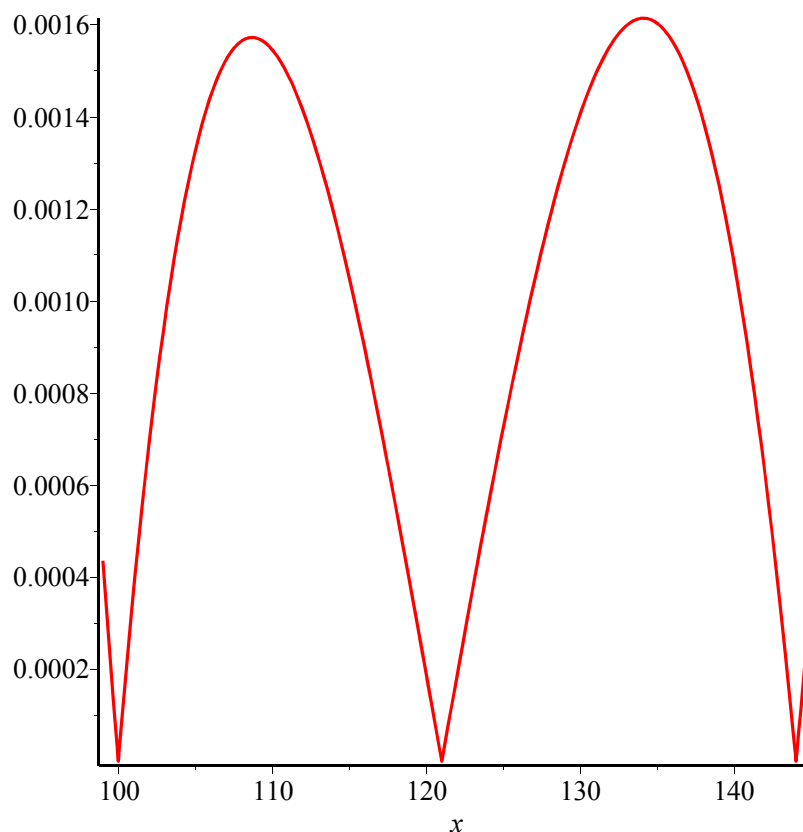
> `hiba_szazalek := evalf( $\frac{hiba \cdot 100}{\sqrt{110}}$ , 4)`  
`hiba_szazalek := 0.01474`

Az elkövetett hiba 0.01%, ami elég pontos eredmény.

**b;**

Képezzük a  $\sqrt{x}$  függvény és a parabola különbségének abszolútértékét, majd rajzoljuk fel a [100, 144] intervallumon!

> `plot(abs(sqrt(x) - parabola), x=99..145, color=red);`



Az eltérések a 100, 121 és 144 pontokban természetes 0 értékűek. Hol lesz a hibafüggvény maximuma és mekkora ez a maximális eltérés? Ezt deriválással dönthetjük el.

$$> \frac{\partial}{\partial x} (\sqrt{x} - parabola) = 0$$

$$\frac{1}{2\sqrt{x}} + \frac{1}{5313}x - \frac{727}{10626} = 0$$

Ez az egyenlet csak numerikusan oldható meg, például a 2. fejezetben megismert *fsolve* utasítással.

```
> x1 := fsolve(% , x = 100 .. 120);
x2 := fsolve('%%', x = 120 .. 141); max_hiba_hely := max(x1, x2); max_hiba
:= evalf(subs(x = max_hiba_hely, abs(sqrt(x) - parabola)))
x1 := 108.6850875
x2 := 134.0926307
max_hiba_hely := 134.0926307
max_hiba := 0.001614746
```

Vizsgáljuk az interpolációs polinom és a valódi függvény közötti eltérést olyan esetben, amikor a valódi függvényt is ismerjük! A leglényegesebb hátránya a Lagrange-interpolációnak az, hogy bizonyos esetekben indokolatlanul erőteljes hullámszások jelennek meg a polinom grafikonján. Sajnos az alappontok számának növelése még "jó" függvény esetében sem vezet mindig tetszőleges pontosításhoz. Ezt mutatja a következő ún. *Runge-függvény* példája.

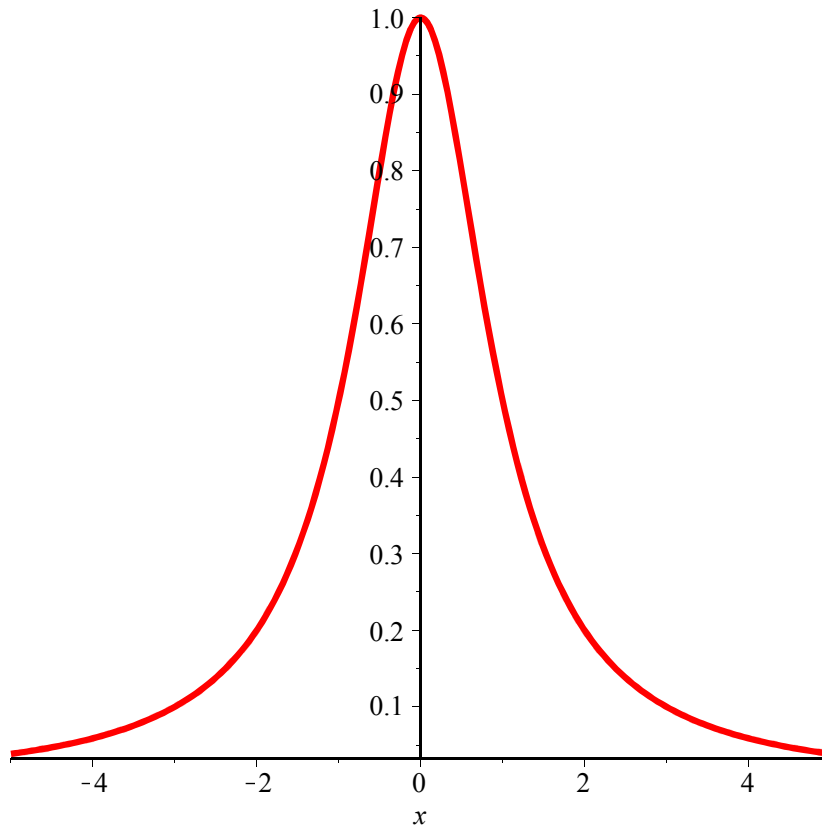
**3. Példa** Tekintsük az  $f(x) = \frac{1}{1+x^2}$  Runge-függvényt a  $[-5, 5]$  intervallumon.

a; Azonos eltérésű osztáspontok esetén növeljük  $n$  értékét és figyeljük meg a közelítés pontatlanságának növekedését.

b; Használjuk a **Csebisev - alappontokat**, ahol  $x_i = \frac{a+b}{2} + \frac{(b-a) \cos\left(\frac{(2i-1)\pi}{2n}\right)}{2}$ ,  $i = 1, 2, \dots, n$ , az  $[a, b]$  intervallumon belül. Nézzük meg hogyan javítja az interpolációs polinom hibáját az alappontok Csebisev-féle választása a fenti Runge-függvényre:

**Megoldás a;**

```
> f := x -> 1/(x^2 + 1) : abra1 := plot(f(x), x = -5 .. 5, color = red, thickness = 3) : abra1
```



Ha az intervallumot egyenlető részekre osztjuk, a  $X$  halmaz elemei egy számtani sorozat tagjai. Az  $x$ -ekhez tartozó  $y$  értékeket egy lépésben meg tudjuk határozni, a *map* utasítás segítségével, így  $X$  minden eleméhez hozzárendelődik a megfelelő függvényérték.

```
> n := 10; X := [seq(-5 + 10*i/n, i = 0..n)]; Y := map(f, X)
```

```
n := 10
```

```
X := [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
```

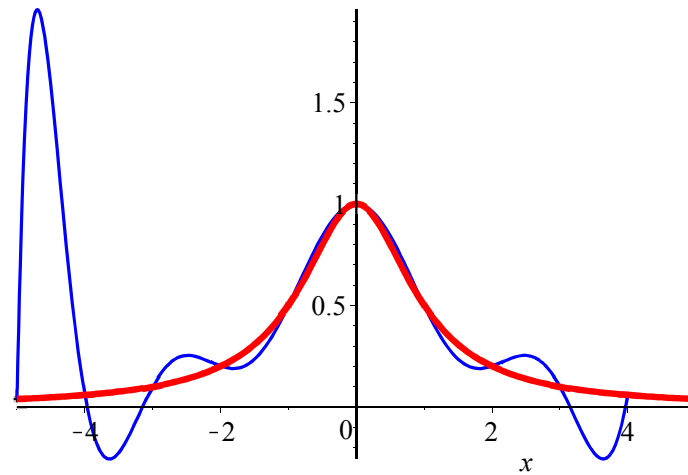
```
Y := [1/26, 1/17, 1/10, 1/5, 1/2, 1, 1/2, 1/5, 1/10, 1/17, 1/26]
```

```
> PolynomialInterpolation(X, Y, x, form = Lagrange) :
```

```
> pontok := plot([seq([X_i, Y_i], i = 1..n)], style = point, symbol = diamond, color = black) :
```

```
graf := plot(PolynomialInterpolation(X, Y, x, form = Lagrange), x = X_1..X_n, color = blue) :
```

```
> plots[display]([pontok, graf, abra1])
```



A következő animáció azt mutatja, hogy az osztópontok számát növelve a kilengések nem csökkennek.

> **for**  $n$  **from** 10 **by** 2 **to** 20 **do**

$X := \left[ \text{seq} \left( -5 + \frac{10i}{n}, i = 0 \dots n \right) \right];$

$Y := \text{map}(f, X);$

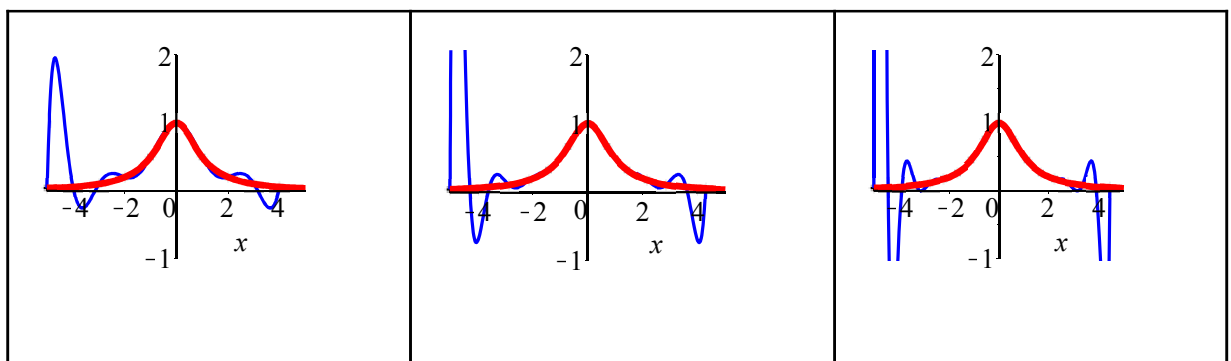
$\text{pontok}[n] := \text{plot} \left( \left[ \text{seq} \left( [X_i, Y_i], i = 1 \dots n \right) \right], \text{style} = \text{point}, \text{symbol} = \text{diamond}, \text{color} = \text{black} \right);$

$\text{graf}[n] := \text{plot} \left( \text{CurveFitting}[\text{PolynomialInterpolation}](X, Y, x, \text{form} = \text{Lagrange}), x = X_1 \dots X_n, \text{color} = \text{blue} \right);$

$\text{kep}[n] := \text{plots}[\text{display}] \left( \left[ \text{pontok}[n], \text{graf}[n], \text{abra1} \right], \text{view} = [-5 \dots 5, -1 \dots 2] \right)$

**end do;**

>  $\text{Matrix}(1, 3, [\text{kep}[10], \text{kep}[14], \text{kep}[18]]) : \text{plots}[\text{display}] (\%)$



**b;**

A Csebisev alappontokat véve

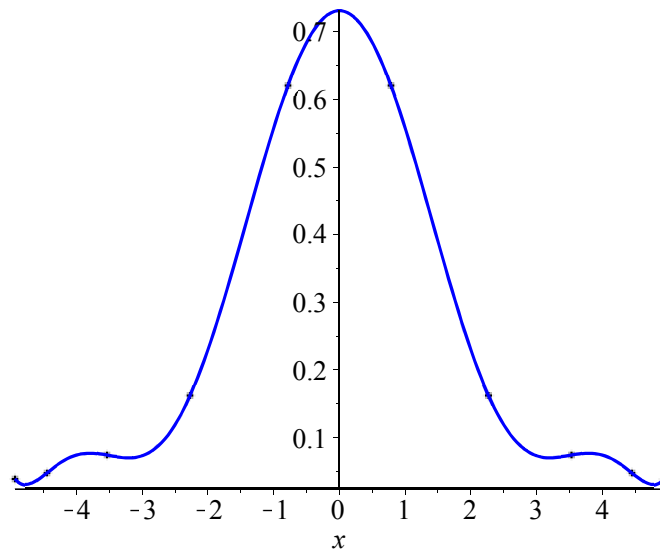
>  $n := 10 : X := \text{evalf} \left( \left[ \text{seq} \left( 5 \cos \left( \frac{(2i-1)\pi}{2n} \right), i = 1 \dots n \right) \right] \right) : Y := \text{map}(f, X) :$

>  $\text{pontok} := \text{plot} \left( \left[ \text{seq} \left( [X_i, Y_i], i = 1 \dots n \right) \right], \text{style} = \text{point}, \text{symbol} = \text{diamond}, \text{color} = \text{black} \right) :$

```

graf := plot( CurveFitting[PolynomialInterpolation](X, Y, x, form = Lagrange), x = X1 .. Xn,
             color = blue ) :
> plots[display]( [pontok, graf] )

```



Ha most az osztópontok számát növeljük, a görbe simábbá válik.

```

> for n from 10 to 20 do

```

```

    X := evalf( [ seq( 5 cos( (2 i - 1) π / (2 n) ), i = 1 .. n ) ] );

```

```

    Y := map(f, X);

```

```

    pontokI[n] := plot( [ seq( [ Xi, Yi ], i = 1 .. n ) ], style = point, symbol = diamond, color
    = black );

```

```

    grafI[n] := plot( CurveFitting[PolynomialInterpolation](X, Y, x, form
    = Lagrange), x = X1 .. Xn, color = blue );

```

```

    kepI[n] := plots[display]( [ [ pontokI || n, grafI || n, abraI ], view = [ -5 .. 5, 0 .. 1 ] ] )

```

```

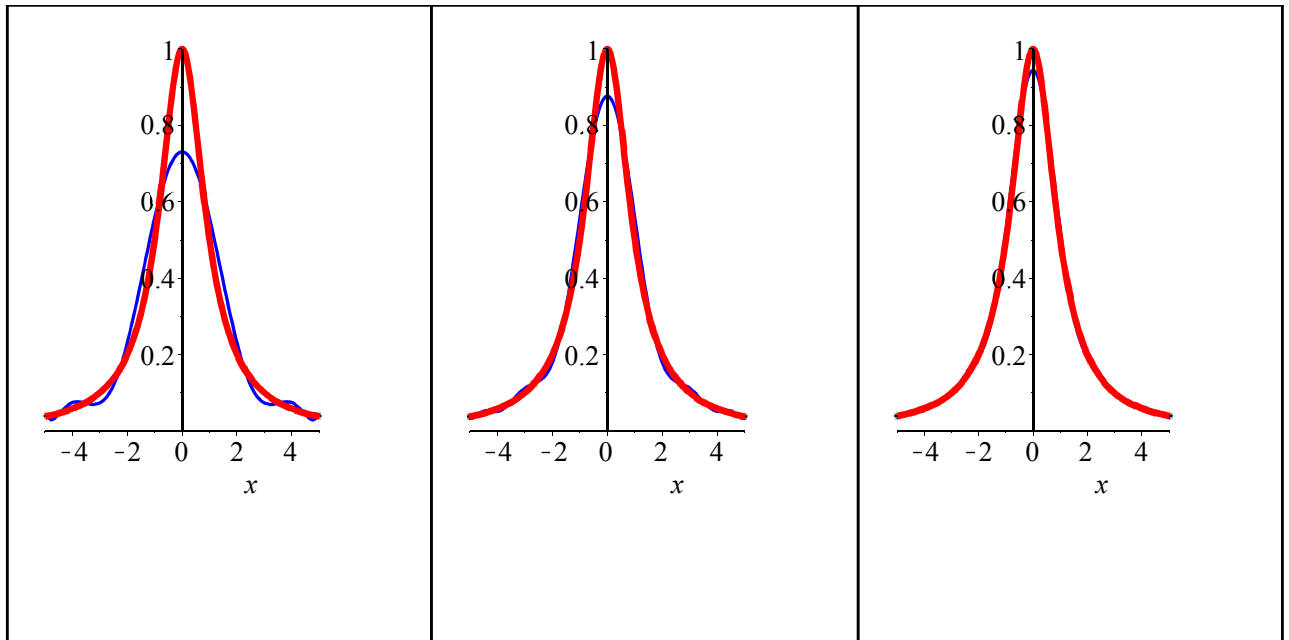
end do:

```

```

> Matrix(1, 3, [ kepI[10], kepI[14], kepI[18] ]) : plots[display](%)

```



## ▼ 7. 6. Spline interpoláció elve, megadása Maple-lel

Használt Maple parancs: [spline](#)

Bonyolult alakú tárgyak megfelelő modellezésére és valósághű képi megjelenítésére a CAD-rendszerek (Computer Aided Design) a spline-technikán alapuló matematikai módszereket használnak.

Az  $x_1 = a, x_2, \dots, x_{n-1}, x_n = b$  pontokban adott  $f(x)$  függvényt most nem egyetlen polinommal közelítjük a teljes  $[a, b]$  intervallumon - mint az interpoláció esetében - hanem az  $[a, x_1], [x_1, x_2], \dots, [x_{n-1}, b]$  részintervallumokon különböző (általában harmadfokú) polinommal közelítjük a függvényt.

Az egyes polinomok mindig két egymást követő, megadott pont között értelmezettek, valamint úgy határozzuk meg a polinomok együtthatóit, hogy az illeszkedési pontokban a deriváltak megfelelőképpen viselkedjenek. A **köbös spline** esetén, mint neve is elárulja, harmadfokú polinomokkal kötjük össze az egymást követő pontokat. A gyakorlatban általában ezt használják, ezért vizsgáljuk meg, milyen feltételeknek kell teljesülniük köbös spline esetén.

Keressük az értéktáblázat pontjaira illeszkedő  $S(x)$  köbös spline függvényt, amely

- az  $[x_i, x_{i+1}]$  szakaszon egy  $S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$  harmadfokú polinommal egyenlő minden  $i=1, 2, 3, \dots, (n-1)$ -re. Ez összesen  $(n-1)$  darab egyenlet, amelyben **4**  $(n-1)$  számú ismeretlen van.
- $S(x_i) = y_i$  minden  $i=1, 2, 3, \dots, n$  Ez ad  **$n$**  darab **feltételi egyenletet**.
- $S(x), \frac{d}{dx} S(x)$  és  $\frac{d^2}{dx^2} S(x)$  függvények folytonosak minden  $x_2, x_3, \dots, x_{n-1}$  belső csomópontban. Ez **3**  $(n-2)$  darab feltételi egyenletet ad.

Összesen tehát meg kell határozni  **$4n-4$  együtthatót**, amihez  **$n + 3(n-2) = 4n-6$  egyenlet** van. Tehát az egyenletek száma 2-vel kevesebb, mint az ismeretlenek száma. Így a feladat megoldása nem lenne egyértelmű. A spline meghatározásához vegyük még hozzá a következő 2 feltételt

- $S''(x_1) = S''(x_n) = 0$  **2** darab **feltétel** (az ilyen spline-t természetes vagy **natural spline**-nak

nevezik) .

Ezzel együtt a spline már egyértelműen meghatározható.

**4. Példa** Adjuk meg a következő 9 pontra illeszkedő Lagrange-féle interpolációs polinomot, a különböző fokszámú spline interpolációs polinomokat, majd ábrázoljuk őket közös koordinátarendszerben.

$X := [1, 2, 5, 6, 7, 8, 10, 13, 17]$ ,  $Y := [3, 3.7, 3.9, 4.2, 5.7, 6.6, 7.1, 6.7, 4.5]$

### Megoldás

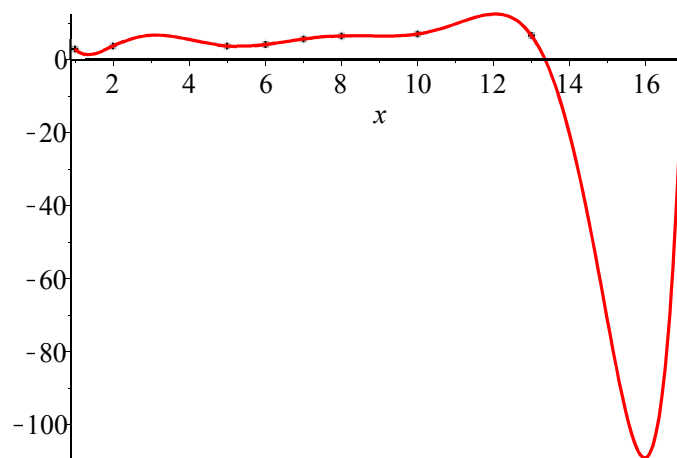
A *CurveFitting* csomag *Spline* parancsa alapesetben a köbös spline interpolációt adja. A *degree=fokszám* opcióval ezt megváltoztathatjuk.

```
> restart : with(CurveFitting) : X := [1, 2, 5, 6, 7, 8, 10, 13, 17] :  
Y := [3, 3.7, 3.9, 4.2, 5.7, 6.6, 7.1, 6.7, 4.5] : N := nops(Y) :
```

A Lagrange interpoláció 8-adfokú polinom. Ábrázoljuk a pontokkal együtt.

```
> simplify(PolynomialInterpolation(X, Y, x, form = power))  
> pontok := plot([seq([Xi, Yi], i = 1 .. N)], style = point, symbol = diamond, color = black) :  
> graf := plot(PolynomialInterpolation(X, Y, x), x = X1 .. XN, color = red) :  
> plots[display]([pontok, graf])
```

$0.00003966124529 x^8 - 0.002505406239 x^7 + 0.06510434690 x^6 - 0.9016264337 x^5$   
 $+ 7.191872218 x^4 - 33.23545879 x^3 + 84.63256867 x^2 - 104.2706592 x + 49.52066494$



Az ábra is mutatja, hogy a kapott görbe nagyon "kileng", nem valószínű, hogy érdemes például a [14,16] intervallumban ez alapján következtetni a függvény viselkedésére. Vizsgáljuk a különböző fokszámú spline-okat.

fokszám=1

fokszám=2

```
> spl := Spline(X, Y, x,  
degree = 1)
```

```
> Digits = 4 : sp2 := Spline(X, Y, x, degree = 2)
```

```

spl := {
  2.3 + 0.7 x
  3.566666667 + 0.06666666667 x
  2.4 + 0.3 x
  -4.8 + 1.5 x
  -0.6 + 0.9 x
  4.600000000 + 0.2500000000 x
  8.433333333 - 0.1333333333 x
  13.85000000 - 0.5500000000 x
}
sp2 := {
  3. + 1.03966512018099 (x - 1)^2
  2.25866948072398 + 0.720669759638011 x - 0.3189953605
  3.80685224494736 + 0.0186295510105273 x + 0.0849819576
  -1.64487458980465 + 0.974147431634109 x + 0.870535922
  -3.65540101429373 + 1.33648585918482 x - 0.5081974954
  1.74450069394413 + 0.606937413256983 x - 0.2213509505
  6.45408437682486 + 0.0645915623175137 x - 0.0498219749
  12.2662902178021 - 0.428176170600161 x - 0.1144339360
  4.500000000000000 + 0.221477978674980 (x - 1)
}

```

```

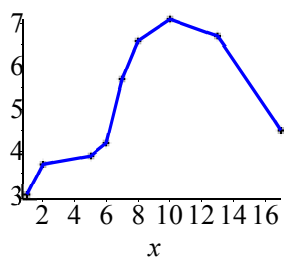
> spl_abra := plot(spl, x
= X1..XN, color
= blue) :

```

```

> plots[display]([pontok,
spl_abra])

```



```

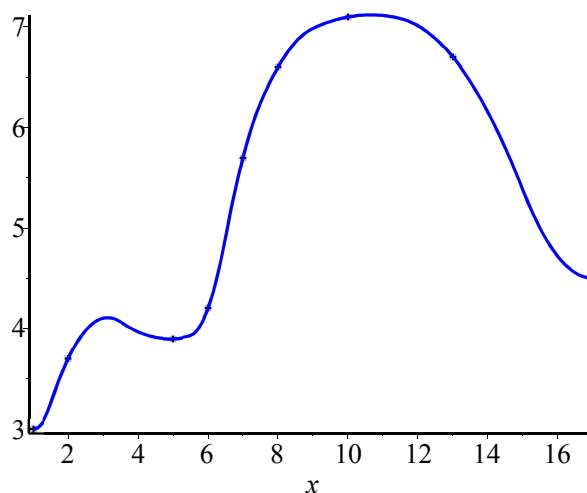
> sp2_abra := plot(sp2, x = X1..XN, color = blue) :

```

```

> plots[display]([pontok, sp2_abra])

```



fokszám=3

```

> sp3 := Spline(X, Y, x, degree = 3)

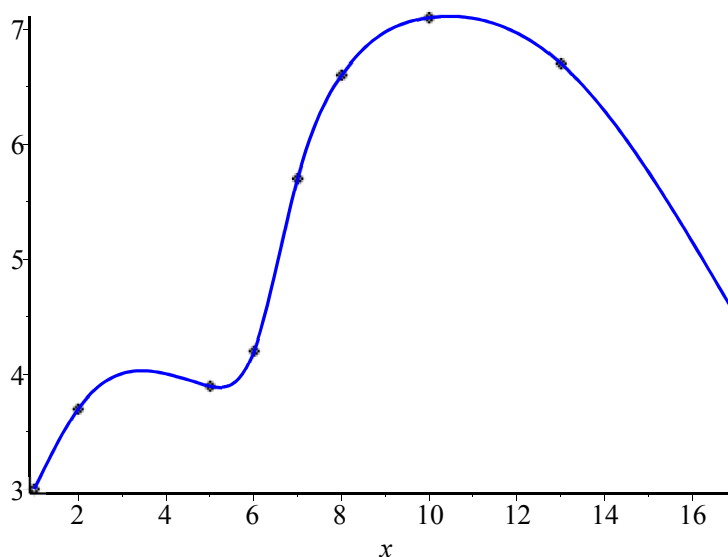
```

$$sp3 := \left\{ \begin{array}{l} 2.21429557735367 + 0.785704422646330 x - 0.0857044226463298 (x - 1)^2 \\ 2.64281769058532 + 0.528591154707340 x - 0.257113267938989 (x - 2)^2 + 0.034379479 \\ 4.32921252798856 - 0.0858425055977122 x + 0.0523020478373052 (x - 5)^2 + 0.33354045 \\ -1.91629778014872 + 1.01938296335812 x + 1.05292342111853 (x - 6)^2 - 0.572306384 \\ -4.15817456515664 + 1.40831065216523 x - 0.663995732311411 (x - 7)^2 + 0.155685080 \\ 2.22100457615247 + 0.547374427980941 x - 0.196940491872884 (x - 8)^2 + 0.024126638 \\ 6.60867872216113 + 0.0491321277838865 x - 0.0521806582256431 (x - 10)^2 - 0.002880387 \\ 11.1423896515810 - 0.341722280890844 x - 0.0781041446659336 (x - 13)^2 + 0.006508678 \end{array} \right.$$

>

>  $sp3\_abra := plot(sp3, x = X_1..X_N, color = blue) :$

>  $plots[display]([pontok, sp3\_abra])$



>

A természeteses köbös spline-ra teljesül a következő tétel:

### ▼ 7. 6. 1. Tétel

Legyen  $S(x)$  olyan természetes spline függvény, amely az  $S(x_i) = y_i$  ( $i = 1, 2, 3, \dots, n$ ) alappontokra illeszkedik. Továbbá az  $y(x)$  legyen tetszőleges kétszer differenciálható olyan függvény, amely szintén illeszkedik az alappontokra, azaz  $y(x_i) = y_i$  ( $i = 1, 2, 3, \dots, n$ ).

Ekkor teljesül az  $\int_a^b \left( \frac{d^2}{dx^2} S(x) \right)^2 dx \leq \int_a^b \left( \frac{d^2}{dx^2} y(x) \right)^2 dx$  egyenlőtlenség.

**5. Példa** Bizonyítsuk be az előző tétel teljesülését az  $f(x) = e^x$  függvény esetén a  $[0,1]$  zárt intervallumon, az intervallum 5 egyenlő részre osztásával.

### Megoldás

Először határozzuk meg a görbe azon pontjait, amelyeken a polinomok majd keresztül haladnak.

```
> restart; f := x -> e^x : a := 0 : b := 1 : N := 5 : X := [seq( (i*(b-a))/N, i=0..N )];
Y := evalf( map(f, X) )
X := [0, 1/5, 2/5, 3/5, 4/5, 1]
Y := [1., 1.221402758, 1.491824698, 1.822118800, 2.225540928, 2.718281828]
```

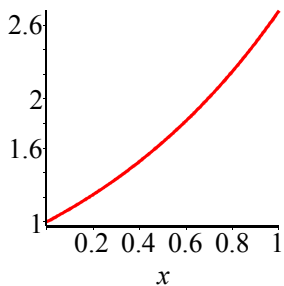
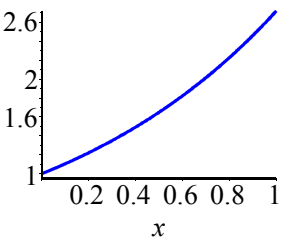
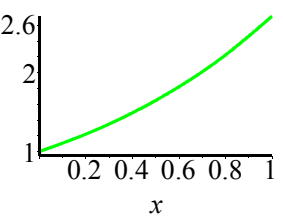
Az 5-ödfokú interpolációs polinom

```
> Lagrange_polinom := CurveFitting[PolynomialInterpolation](X, Y, x)
Lagrange_polinom := 0.01385411458 x^5 + 0.03486687501 x^4 + 0.1704093854 x^3
+ 0.4990689250 x^2 + 1.000082528 x + 1.
```

A harmadfokú spline

```
> spline3 := CurveFitting[Spline](X, Y, x, degree=3, endpoints='natural')
```

```
spline3 := {
    1. + 1.05796629971292 x + 1.22618725717703 x^3
    0.980381003885167 + 1.20510877057416 x + 0.735712354306221 (x - 1/5)^2 - 0.0035385
    0.892237062803828 + 1.49896908799043 x + 0.733589232775119 (x - 2/5)^2 + 0.144589
    0.736265495521531 + 1.80975550746411 x + 0.820342864593301 (x - 3/5)^2 + 1.082163
    0.411339062277512 + 2.26775233215311 x + 1.46964125885168 (x - 4/5)^2 - 2.449402
```

| $f(x) = e^x$                                                                                                                      | $Lagrange\_polinom$                                                                                                                            | $spline3$                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&gt; plot(f(x), x=0..1, color=red)</pre>  | <pre>&gt; plot(Lagrange_polinom, x=0..1, color=blue)</pre>  | <pre>&gt; plot(spline3, x=0..1, color=green)</pre>  |

A három függvény látszólag azonos. Számítsuk ki az  $f(x)$  függvény, az interpolációs polinom és a spline függvény második deriváltja négyzetének integrálját a  $[0, 1]$  intervallumon.

$$> \text{integral\_f} := \int_a^b \left( \frac{d^2}{dx^2} f(x) \right)^2 dx = \text{evalf} \left( \int_a^b \left( \frac{d^2}{dx^2} f(x) \right)^2 dx \right)$$

$$> \text{integral\_Lagrange} := \int_a^b \left( \frac{\partial^2}{\partial x^2} \text{Lagrange\_polinom} \right)^2 dx = \text{evalf} \left( \int_a^b \left( \frac{\partial^2}{\partial x^2} \text{Lagrange\_polinom} \right)^2 dx \right)$$

$$> \text{integral\_spline} := \int_a^b \left( \frac{\partial^2}{\partial x^2} \text{spline3} \right)^2 dx = \text{evalf} \left( \int_a^b \left( \frac{\partial^2}{\partial x^2} \text{spline3} \right)^2 dx \right)$$

$$\text{integral\_f} := \int_0^1 \left( \frac{d^2}{dx^2} e^x \right)^2 dx = 3.194528050$$

$$\text{integral\_Lagrange} := \int_0^1 \left( \frac{d^2}{dx^2} (0.01385411458 x^5 + 0.03486687501 x^4 + 0.1704093854 x^3 + 0.4990689250 x^2 + 1.000082528 x + 1.) \right)^2 dx = 3.193847417$$

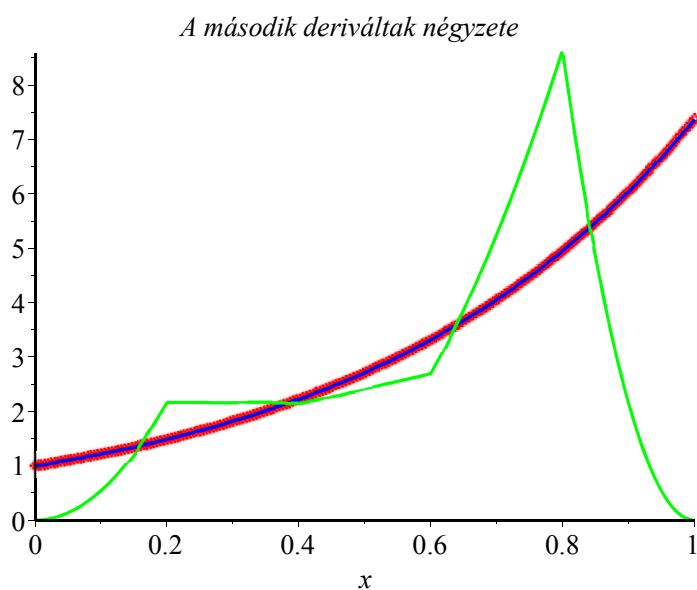
$$\text{integral\_spline} := \int_0^1 \left( \frac{d^2}{dx^2} \left\{ \begin{array}{l} 1. + 1.05796629971292 x + 1.226187257 \\ 0.980381003885167 + 1.20510877057416 x + 0.735712354306221 (x - \\ 0.892237062803828 + 1.49896908799043 x + 0.733589232775119 (x - \\ 0.736265495521531 + 1.80975550746411 x + 0.820342864593301 (x - \\ 0.411339062277512 + 2.26775233215311 x + 1.46964125885168 (x - \end{array} \right. \right)$$

$$\int_0^1 \left( \frac{d^2}{dx^2} f(x) \right)^2 dx = 2.712421981$$

Valóban, spline második deriváltja négyzetének az integrálja a legkisebb.

Rajzoljuk még fel az  $f(x)$ , az interpolációs polinom és a spline függvények második deriváltjainak négyzetét. Az előző észrevétel azt mondja ki, hogy a görbék alatti területek közül a legkisebb a spline görbe alatti terület!

```
> plot( [ ( (d^2/dx^2 f(x))^2, (d^2/dx^2 Lagrange_polinom)^2, (d^2/dx^2 spline3)^2 ], x=0..1, style
= [point, line, line], title = A második deriváltak négyzete, color = [red, blue, green] )
```



## ▼ 7. 7. Felület interpoláció

Használt Maple parancs: [ArrayInterpolation](#), [matrixplot](#), [ArrayTools](#) csomag, [Concatenate](#)

A felületek interpolációja az építészetben, számítógépes grafikában nagy jelentőségű. Bár e jegyzet kereteit meghaladja ennek részletes elméleti tárgyalása, egy példán keresztül megmutatjuk a Maple

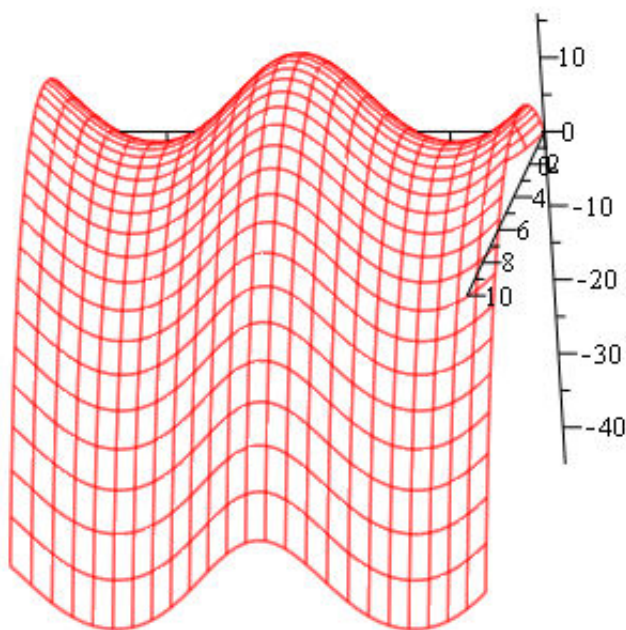
nyújtotta lehetőségeket.

**6. Példa** Az alábbi példa az  $f := (x, y) \rightarrow (3 - \sin(x))^2 - (3 - y)^2$  kétváltozós felület pontjainak közelítését mutatja.

### Megoldás

Az *ArrayInterpolation* parancs használatához először definiálni kell az  $x$  és  $y$  független változókat, valamint az ehhez tartozó  $z$  függvényértékeket  $X$ ,  $Y$ ,  $Z$  halmazokban. Vagyis azokat a pontokat, amelyeket a felületnek tartalmaznia kell.

```
> restart; with(CurveFitting) : with(plots) :  
> f := (x, y) → (3 - sin(x))^2 - (3 - y)^2 :  
> a := plot3d(f, 0 ..10, 0 ..10, axes = normal, style = wireframeopaque, color = red, orientation  
    = [90, 70, -10]) :  
a
```



```
> X := [0, 1.5, 3.5, 5, 6, 8, 9]  
> Y := [0, 3, 5, 6, 6.5, 7.5]  
> Z := Matrix(7, 6, (a, b) → evalf(f(X[a], Y[b])))  
X := [0, 1.5, 3.5, 5, 6, 8, 9]  
Y := [0, 3, 5, 6, 6.5, 7.5]
```

```
Z :=  
[[0., 9., 5., 0., -3.25, -11.25],
```

```

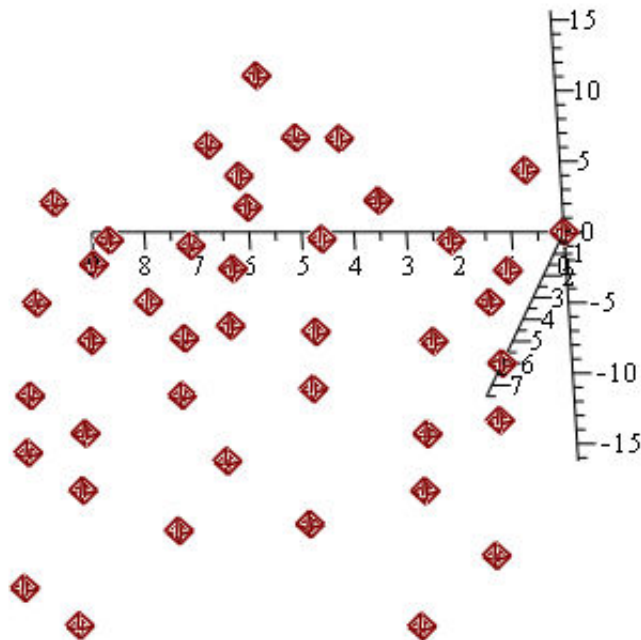
[-4.989973673, 4.010026327, 0.010026327, -4.989973673, -8.239973673, -16.23997367],
[2.22774824, 11.22774824, 7.22774824, 2.22774824, -1.02225176, -9.02225176],
[6.67308142, 15.67308142, 11.67308142, 6.67308142, 3.42308142, -4.57691858],
[1.75456601, 10.75456601, 6.75456601, 1.75456601, -1.49543399, -9.49543399],
[-4.957319741, 4.042680259, 0.042680259, -4.957319741, -8.207319741, -16.20731974],
[-2.302869264, 6.697130736, 2.697130736, -2.302869264, -5.552869264, -13.55286926]]

```

```

> c := plots[pointplot3d]([seq(seq([Xp, Yp, Zi,j], j = 1 .. 6), i = 1 .. 7)], axes = normal, symbol
    = diamond, symbolsize = 30, color = red, symbolsize = 30, orientation = [90, 70, -10]) :
c

```



Most definiáljuk azt a  $A$  mátrixot, amely az interpolációs felület pontjainak  $x$  és  $y$  koordinátáit, a  $B$  mátrix pedig az interpolált felületi pontok koordinátáit tartalmazza..

```

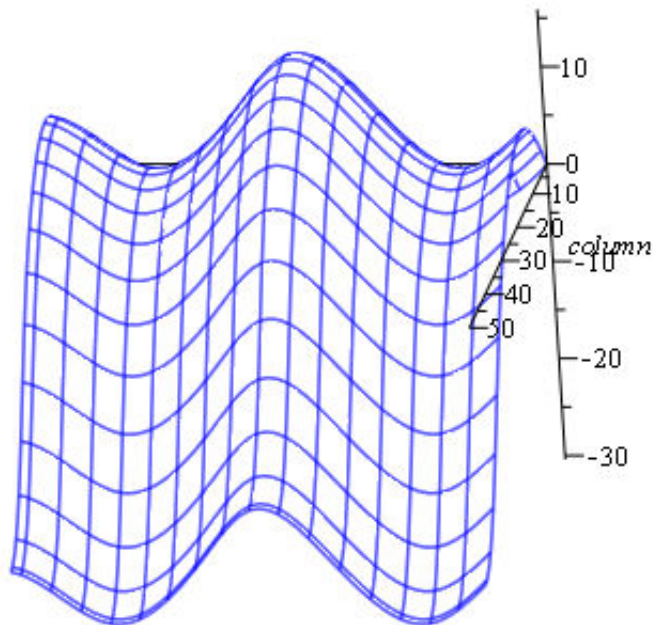
> a1 := Matrix(50, 50, (i, j) →  $\frac{i}{5}$ ) : a2 := Matrix(50, 50, (i, j) →  $\frac{j}{5}$ ) :
A := ArrayTools[Concatenate](3, a1, a2) :
> B := ArrayInterpolation([X, Y], Z, A, method = spline);

```

$$B := \begin{bmatrix} 1..50 \times 1..50 \text{ Array} \\ \text{Data Type: float}_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

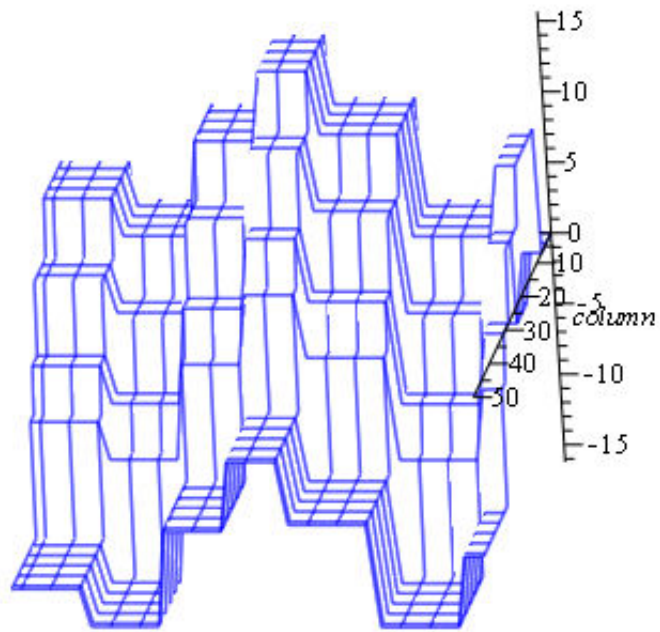
Például  $B[2,3]=0.672604278238919$  azt jelenti, hogy az  $A$  mátrix második sor harmadik oszlopában lévő független változóhoz  $0.672604278238919$  függvényérték tartozik. Ezeket a pontokat a *matrixplot* paranccsal tudjuk ábrázolni.

>  $b := \text{matrixplot}(\text{Matrix}(B), \text{axes} = \text{normal}, \text{color} = \text{blue}, \text{style} = \text{hidden}, \text{orientation} = [90, 70, -10]) :$   
 $b$

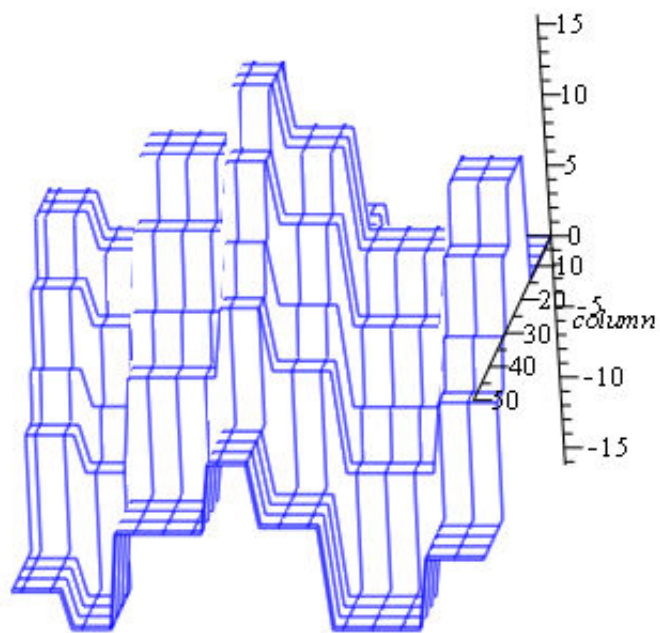


Vizsgáljuk meg, a különböző módszerek milyen közelítő felületeket ad.

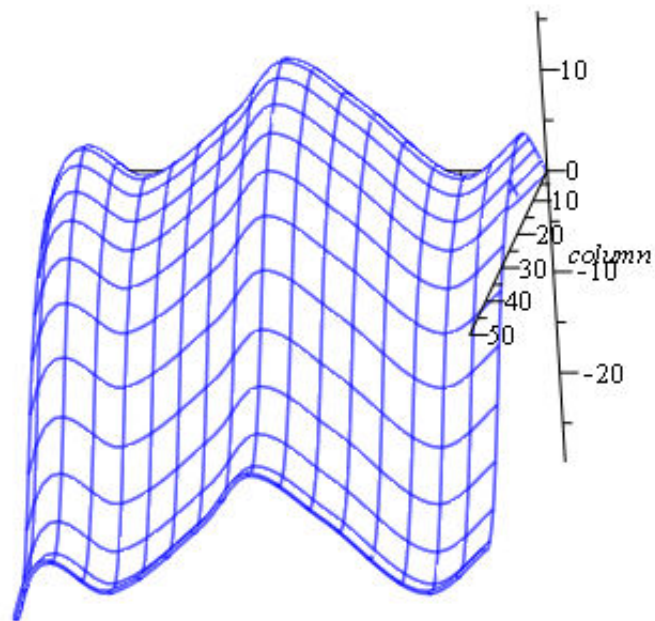
>  $C := \text{ArrayInterpolation}([X, Y], Z, A, \text{method} = \text{nearest}) : \text{matrixplot}(\text{Matrix}(C), \text{axes} = \text{normal}, \text{color} = \text{blue}, \text{style} = \text{hidden}, \text{orientation} = [90, 70, -10])$



>  $E := \text{ArrayInterpolation}([X, Y], Z, A, \text{method} = \text{lowest}) : \text{matrixplot}(\text{Matrix}(E), \text{axes} = \text{normal}, \text{color} = \text{blue}, \text{style} = \text{hidden}, \text{orientation} = [90, 70, -10])$



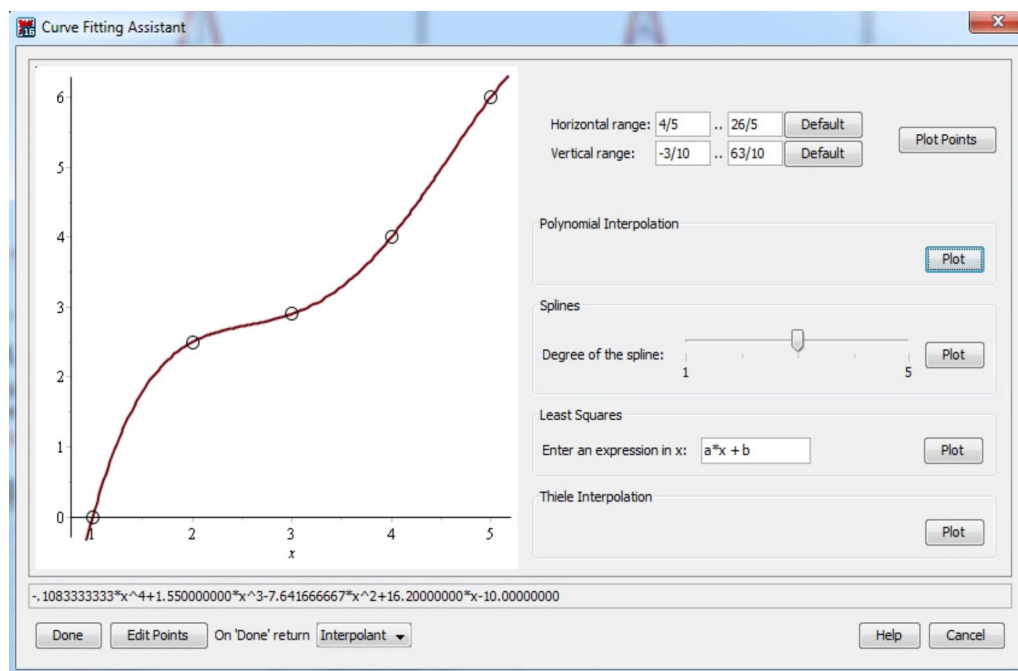
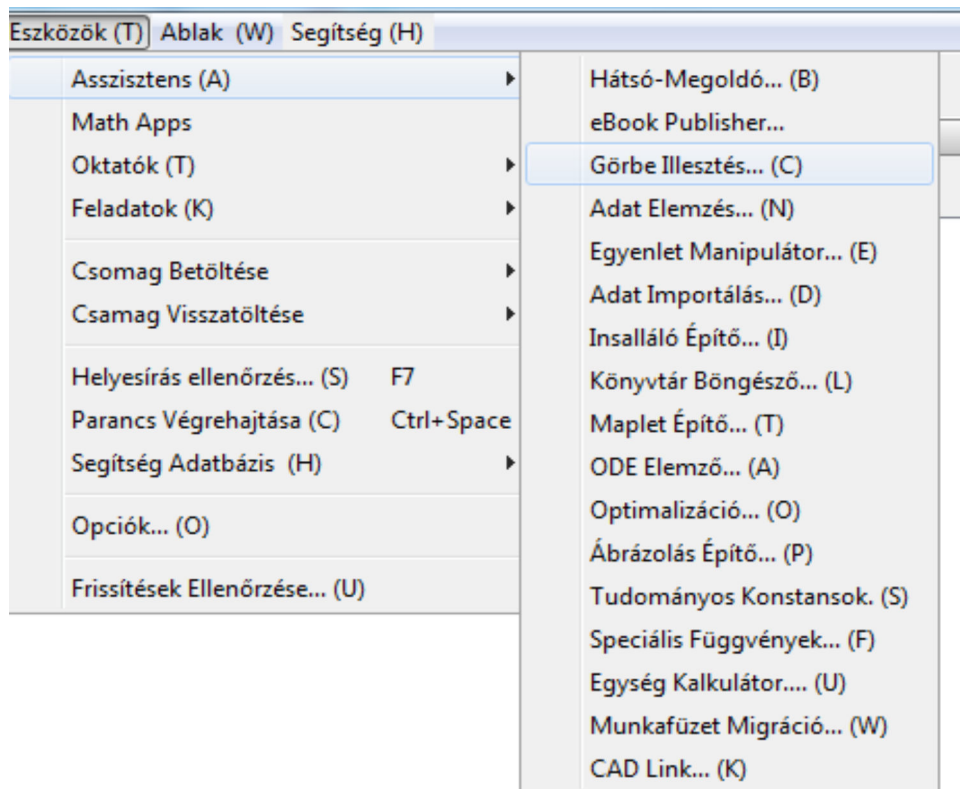
>  $F := \text{ArrayInterpolation}([X, Y], Z, A, \text{method} = \text{cubic}) :$   
 $\text{matrixplot}(\text{Matrix}(F), \text{axes} = \text{normal}, \text{color} = \text{blue}, \text{style} = \text{hidden}, \text{orientation} = [90, 70,$   
 $-10])$



## ▼ 7. 8. Beépített Maple lehetőségek

Eszközök->Asszisztens(A)->Görbeillesztés(C) asszisztens többféle görbeillesztést tesz lehetővé. A pontok megadása után választhatunk Polinomial Interpolation vagy Splines lehetőségek közül.

> **CurveFitting[Interactive]();**



## ▼ 7. 9. Feladatok

**1. feladat** Egy társaság házibulit rendez. A mulatság 4 órakor kezdődik. Ekkor 60 üveg sor van. Egy óra alatt elfogyott 15 üveg. 8 órakor valaki megszámolta a teli üvegeket, és 12-t talált. Hány üveg sör lehetett 7 órakor, és előreláthatólag meddig lesz elég a maradék?

**2. feladat** Egy légy egyik lábával belelépett egy tintapacába, majd sétaútra indult egy koordinátarendszerrel ellátott fehér papíron. Határozzuk meg a légy útvonalát Newton-féle interpolációval, ha tudjuk, hogy áthaladt a következő pontokon A(-1,0), B(1,3), C(3,4), D(4,2.5)

**3. feladat** Egy statisztikai felmérés szerint száz 12 éves gyerekből 5, száz 16 évesből 15, száz 20 évesből 24, száz 24 évesből 36 dohányzik. A Newton-féle interpoláció segítségével meghatározható-e, hogy a 40 évesek hány százaléka dohányzik?

**4. feladat** Határozza meg az adott alappontokra illeszkedő Newton-féle interpolációs polinomot, és határozza meg a polinom  $x$  helyen felvett értékét, ha  $f(0)=6$ ,  $f(1)=f(2)=5$ ,  $f(3)=15$ ,  $f(4)=50$ . és  $x=5$ .

**5. feladat**  $\lg(1)=0$ ,  $\lg(10)=1$ ,  $\lg(100)=2$  felhasználásával, a Lagrange-féle interpolációs polinom segítségével határozzuk meg  $\lg(3.4)$  értékét! Hol lesz az eltérések maximuma a  $[0,100]$  intervallumon?

**6. feladat** A  $\sin(0)=0$ ,  $\sin \frac{\pi}{6}=\frac{1}{2}$ ,  $\sin \frac{\pi}{3}=\frac{\sqrt{3}}{2}$ ,  $\sin \frac{\pi}{2}=1$  felhasználásával, a Lagrange-féle interpolációs polinom segítségével határozza meg  $\sin \frac{\pi}{5}$  értékét! Hol lesz az eltérések maximuma a  $[0, \frac{\pi}{2}]$  intervallumon?

**7. Feladat** Közelítsük az  $y = \ln(1 + x^2)$  függvényt a  $[-1,1]$  intervallumon interpolációs polinommal és harmadfokú spline függvény segítségével is úgy, hogy az intervallumot  $n=10$  egyenlő részre osztjuk fel! Rajzoljuk fel a kapott függvényeket az eredeti függvénnyel azonos koordinátarendszerben! Számoljuk ki a  $[-1,1]$  intervallumon az  $f(x)$  függvény és a közelítő függvények második deriváltja négyzeteinek integrálját! Melyik integrál lesz a legkisebb? Rajzoljuk fel a három darab második derivált függvény grafikonját azonos koordinátarendszerben!

## ▼ 7. 9. Ellenőrző kérdések

1. Írja le a polinom interpoláció alapproblémáját!
2. Másodfokú interpolációs polinom illesztése 3 pontra Lagrange módszerrel.
3. Másodfokú interpolációs polinom illesztése 3 pontra Newton módszerrel.
4. Mi a spline interpoláció elve?