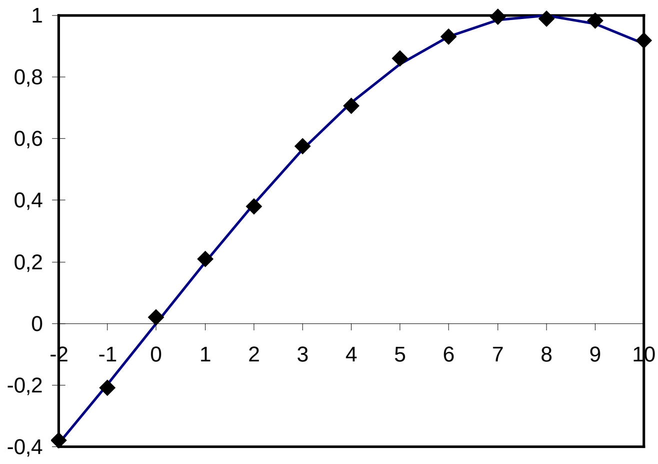
APROKSYMACJA I INTERPOLACJA

Przybliżenie funkcji f($\underline{x}$) przez inną funkcję g($\underline{x}$)

- **funkcja f jest zbyt skomplikowana;**
  użycie f w dalszej analizie problemu jest trudne

- **funkcja f jest znana tylko tabelarycznie;**
  wymagana jest znajomość f dla argumentów poza siatką



najczęściej

$$g(x) = \sum_{k=0}^{n} a_k g_k(x)$$

***$g_k(x)$* - znane funkcje prostsze**

aproksymacja

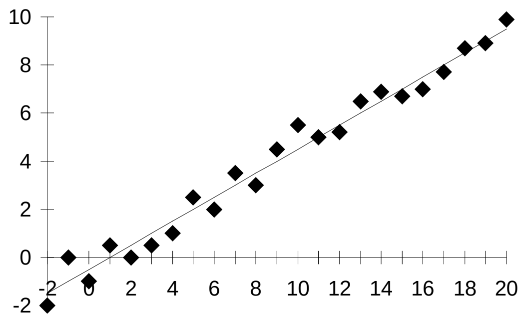
- **wielomianowa**
- **funkcjami trygonometrycznymi**
- **funkcjami wymiernymi**

realizacja np.: wybieramy takie g, że

$$\left\{ g(x_i) \right\} = \left\{ f(x_i) \right\}_{i=1}^{M}$$

$g_k(x_i)$ **- jednomiany ----> aproksymacja wielomianowa**

<u>UWAGA !</u> **jeśli f(x) jest wynikiem doświadczeń nie można stosować interpolacji,**



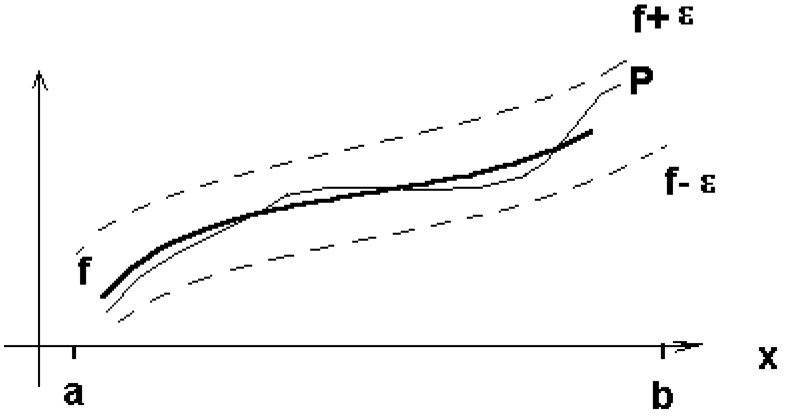
tu stosujemy np. metodę najmniejszych kwadratów (postulujemy kształt g, z parametrami)

$$\min\left(\sum_i \left(g(x_i; a_1, \ldots, a_p) - f(x_i) \right)^2 \right)$$

<u>Tw. Weierstrassa</u>

Jeżeli f(x) jest ciągła w [a,b] ,

$$\underset{\varepsilon>0}{\forall} \quad \underset{n, P_n(x)}{\exists} \quad |f(x) - P_n(x)| < \varepsilon$$



Tw. Weierstrassa (dla funkcji okresowych)

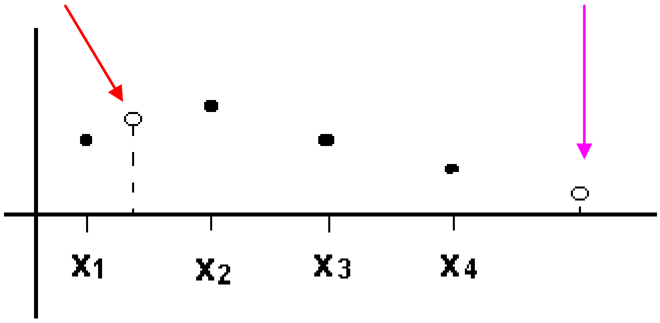
jeśli *F(x)* **jest ciągła i okresowa (2π), to dla każdego ε>0 istnieje wielomian trygonometryczny stopnia n**

$$S_n(x) = a_o + \sum_{k=1}^{n} (a_k \cos(kx) + b_k \sin(kx))$$

że dla każdego *x*

$$|F(x) - S_n(x)| < \varepsilon$$

interpolacja **ekstrapolacja**



Interpolacja wielomianowa

znalezienie, dla układu *n+1* różnych węzłów { x_i }, i=0,1,..n i wartości { f_i }={ $f(x_i)$ }, wielomianu stopnia co najwyżej n, $P_n(x)$ takiego, że

$$P_n(x_i) = f_i$$

TW.

Istnieje dokładnie jeden taki wielomian.

Sposoby konstrukcji wielomianu interpolacyjnego

Dlaczego konstrukcje?
Dlaczego nie tak:

$P_n(x) = a_0x^0 + a_1x^1 + ... + a_nx^n$ - n+1 współczynników,

$\{ P_n(x_i) = f_i \}_{i=0,n+1}$ - n+1 równań

a) możliwa utrata dokładności przy rozwiązywaniu dużej liczby równań
b) chcemy tylko „móc” obliczyć wartość P dla x spoza siatki
= chcemy mieć prosty przepis (wzór) z małą liczbą operacji

gdyby nawet to:

- rozwiązanie układu (n+1) równań algebraicznych wymaga (jak zobaczymy) $\sim n^3$ mnożeń

1) wielomian interpolacyjny <u>Lagrange'a</u>

a) uogólnienie interpolacji liniowej

(x_0, y_0) i (x_1, y_1)

(***) $$P(x) = \frac{(x - x_1)}{(x_0 - x_1)} y_0 + \frac{(x - x_0)}{(x_1 - x_0)} y_1$$

a w ogólności:

$$P(x) = \sum_{k=0}^{n} f_k L_{n,k}(x) \quad , \ f_k = y_k$$

gdzie

wielomiany $L_{n,k} = L_k$ **(ilorazy) stopnia n**

$$L_{n,k}(x_i) = \delta_{ik}$$

są postaci

$$L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)} = \frac{(x - x_0)(x - x_1)..(x - x_{k-1})(x - x_{k+1})..(x - x_n)}{(x_k - x_0)(x_k - x_1)..(x_k - x_{k-1})(x_k - x_{k+1})..(x_k - x_n)}$$

idealne do zaprogramowania.... (pętle)

n **mnożeń prostych ilorazów;**

każdy iloraz to jedno dzielenie, razem to 2n mnożeń,

obl. wartości wielomianu -> n *razy* powyższe mnożenia ...

łącznie ~n² mnożeń

ćwiczenie:

porównaj przybliżenie do wartości funkcji f(x) = 1/x w punkcie x=3 za pomocą 3-wyrazowego rozwinięcia w szereg Taylora w otoczeniu x_0=1 oraz wielomianu Lagrange'a na 3 węzłach w punktach 1.0, 2.0, 4.0

Błąd interpolacji wielomianowej

Twierdzenie

Jeśli $x_0, x_1,..x_n \in [a,b]$ i $f \in C^{n+1}[a,b]$ to

$$\underset{x\in[a,b]}{\forall}\ \underset{\xi(x)\in(a,b)}{\exists}\ f(x)-P(x)=\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\ldots(x-x_n)$$

dowód

zdefiniujmy $L_n(x)=(x-x_0)(x-x_1)\ldots(x-x_n)$

(wielomian *n+1* **stopnia), oraz**

$$\varphi(x)=f(x)-P(x)-K\cdot L_n(x)$$

wybierzmy takie K**, żeby dla dodatkowego** x **z** $[a,b]$

$$\varphi(x)=0$$

Tw. Rolle'a

$\varphi(x)$ **:** **ma** *n+2* **miejsc zerowych**

... **...**

$\varphi^{(n+1)}(x)$ **:** **ma** *1* **miejsce zerowe**

to istnieje takie ξ **w [a,b], że**

$$0 = \varphi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K \cdot (n+1)! \qquad \textbf{(*)}$$

(znika pochodna *P(x),* **pochodna** $L_n(x) = (n+1)!$ **)**

wstawienie *K* **do (*) kończy dowód.**

... program INTERP_G ...

<u>węzły równoodległe</u>

$$x_{i+1} - x_i = h$$

wprowadzając zmienną

$$t = (x - x_0) / h$$

dostajemy

$$\boxed{L_k(x) = \frac{(-1)^{n-k}}{n!}\binom{n}{k}\prod_{\substack{i=0 \\ i \neq k}}^{n}(t-i)}$$

dla

$$P(x) = \sum_{k=0}^{n} f_k L_k(x)$$

w L_k - 2n mnożeń , łączni ~n^2 ale b. prosty algorytm (!)

<u>metoda Neville'a</u> (interpolacja iteracyjna)

jak znaleźć rząd wielomianu, który "dostatecznie" dokładnie przybliża *f* w punkcie *x* bez konieczności generowania "od początku" kolejnych wielomianów

$$x_0, x_1, x_2, \ldots x_n$$

niech

$Q_{i,j-1}$, $i \geq j-1$ **oznacza wielomian stopnia** $j-1$

określony na j węzłach:

$$x_{i-j+1}, x_{i-j+2}, \ldots x_{i-1}, x_i$$

to

wielomian stopnia j **można zbudować jako**

$$Q_{i,j}(x) = \frac{(x - x_{i-j})Q_{i,j-1}(x) - (x - x_i)Q_{i-1,j-1}(x)}{x_i - x_{i-j}}$$

tzn. na takiej samej zasadzie jak konstruowaliśmy kolejne wielomiany Lagrange'a w (***) (np. *i=1, j=1*)

---

x_0 $Q_{0,0}$

x_1 $Q_{1,0}$ $Q_{1,1}$

x_2 $Q_{2,0}$ $Q_{2,1}$ $Q_{2,2}$

x_3 $Q_{3,0}$ $Q_{3,1}$ $Q_{3,2}$ $Q_{3,3}$

... ...

odpowiedni fragment algorytmu

```
* wprowadź wartości  x0, x1, ... xn  na  xt[0..m]
  oraz  f(x0), .. f(xn)  jako  Q0,0, Q1,0, .. Qn,0  na  Q[0..m,0..m]
  i:=0;
  repeat
        i:=i+1;
        for  j:=1   to   i   do
        Q[i,j]:=( (x-xt[i-j])*Q[i,j-1] - (x-xt[i])*Q[i-1,j-1] ) / (x[i]-xt[i-j]) ;
   until   abs( Q[i,i] - Q[i-1,j-1] ) < epsilon;
```

WIELOMIAN INTERPOLACYJNY NEWTONA

1. Różnice skończone

różnice progresywne

$$\Delta^0 f(x) = f(x)$$

$$\Delta^k f(x) = \Delta^{k-1} f(x+h) - \Delta^{k-1} f(x)$$

$$\Delta^k f(x) = \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} f(x+ih)$$

różnice wsteczne

$$\nabla^0 f(x) = f(x)$$
$$\nabla^k f(x) = \nabla^{k-1} f(x) - \nabla^{k-1} f(x-h)$$

tablice różnic skończonych

$f_i = f(x_0 + ih)$

....

x_0	f_0	$\Delta^1 f_0$	$\Delta^2 f_0$	$\Delta^3 f_0$		$\Delta^{n-1} f_0$	$\Delta^n f_0$
x_1	f_1	$\Delta^1 f_1$	$\Delta^2 f_1$		...	$\Delta^{n-1} f_1$	
x_2	f_2	$\Delta^1 f_2$	$\Delta^2 f_2$	...			
...	...	...					
x_{n-1}	f_{n-1}	$\Delta^1 f_{n-1}$					
x_n	f_n						

wzór interpolacyjny Newtona (w przód)

$$\boxed{P(x_0 + h \cdot s) = \sum_{i=0}^{n} \binom{s}{i} \Delta^i f_0} \qquad \textbf{(**)}$$

gdzie $\binom{s}{i}$ **należy rozumieć jako**

$$\frac{s(s-1)(s-2)\cdots(s-i+1)}{i!}$$

dla dowolnego s rzeczywistego, także dla s < 0 ,

2. Ilorazy różnicowe

$$\begin{array}{llllll}
x_0 & f[x_0] & & & & \\
 & & f[x_0,x_1] & & & \\
x_1 & f[x_1] & & f[x_0,x_1,x_2] & & \\
 & & f[x_1,x_2] & & & f[x_0,x_1,x_2,x_3,x_4] \\
x_2 & f[x_2] & & f[x_1,x_2,x_3] & & \\
 & & f[x_3,x_2] & & & \\
x_3 & f[x_3] & & & &
\end{array}$$

$$f[x_i, x_{i+1}, .., x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, .., x_{i+k}] - f[x_i, x_{i+1}, .., x_{i+k-1}]}{x_{i+k} - x_i}$$

gdzie $f[x_i]=f_i$, $f[x_i,x_{i+1}]=(f[x_{i+1}]-f[x_i]) / (x_{i+1} - x_i)$
itd.

Dowód wyrażenia (**)

zapiszmy wielomian interpolacyjny jako:

$$P_n(x) = a_o + a_1(x - x_o) + \ldots + a_n(x - x_o)(x - x_1)\ldots(x - x_{n-1})$$

gdzie $a_o, a_1, a_2, \ldots$ **wyznaczone są z wartości** $f(x_i)$

$$a_o = f(x_o) = f[x_o]$$

$$a_1 = \frac{f(x_1) - f(x_o)}{x_1 - x_o} = f[x_o, x_1]$$

$$\ldots$$

$$a_n = f[x_o, x_1, \ldots, x_n]$$

ostatecznie

$$P_n(x) = f[x_o] + \sum_{k=1}^{n} f[x_o, x_1, \ldots, x_k](x - x_o)\ldots(x - x_{k-1})$$

dla równoodległych węzłów $(x_{i+1} - x_i) = h$ **i** $x = x_0 + sh$

mamy (iloczyn (*x-x_k*) da ***h^k $s(s-1)(s-2)\ldots(s-k+1)$*** **)**

$$P_n(x_o + sh) = \sum_{k=0}^{n} \binom{s}{k} k! h^k f[x_o, x_1, \ldots, x_k]$$

ale ponieważ $f[x_o, \ldots, x_k] = \dfrac{1}{k! h^k} \Delta^k f(x_o)$ **, dostajemy (**)**

(gdyż ***$(x_{i+k} - x_i) = kh$*** **)**

$$\boxed{P(x_o + h \cdot s) = \sum_{i=0}^{n} \binom{s}{i} \Delta^i f_0} \qquad \textbf{(**)}$$