5. Funkcijų aproksimavimo metodai

Funkcijų aproksimavimo uždavinių gausu įvairiose matematikos, fizikos ir technikos srityse. Literatūroje nagrinėjama daugybė funkcijų aproksimavimo uždavinio sprendimo metodų. Tai paaiškinama tuo, kad praktikoje susiduriama su daugeliu skirtingų šio uždavinio formuluočių.

Apibendrintai funkcijų aproksimavimo uždavinį galima suformuluoti taip: turint funkcijos $y = f(x)$ reikšmių lentelę (x_i, y_i) (čia $i = \overline{0, n}$, $y_i = f(x_i)$ ir $x_i \neq x_j$, kai $i \neq j$), reikia rasti funkcijos $f(x)$ reikšmę, kai $x = t$, o $t \in [x_0; x_n]$.

Funkcijos $f(x)$ analizinė išraiška paprastai yra nežinoma arba pernelyg sudėtinga.

Atsižvelgiant į y_i reikšmių tikslumą, taikomi du šio uždavinio sprendimo metodai:

- interpoliavimo metodas,
- suglodinimo metodas.

5.1. Interpoliavimas

Interpoliavimo uždavinio formulavimas. *Duota funkcijos $y = f(x)$ reikšmių lentelė (x_i, y_i);* čia $i = \overline{0, n}$, *y_i reikšmės yra tikslios arba paklaidos tokios mažos, kad praktiškai jų galima nepaisyti. Reikia rasti aproksimuojančiąją funkciją $y = F(x)$, priklausančią funkcijų klasei K ir tenkinančią sąlygas*

$$F(x_i) = y_i, \quad i = \overline{0, n}. \qquad (5.1)$$

Šios sąlygos vadinamos Lagranžo interpoliavimo sąlygomis, o pati funkcija $y = F(x)$ — Lagranžo interpoliacine funkcija, arba tiesiog interpoliacine funkcija.

Jei, be $f(x_i)$ reikšmių, taške x_i yra žinomos funkcijos $f(x)$ išvestinių iki $(m_i - 1)$-osios eilės imtinai reikšmės, tai galima reikalauti, kad aproksimuojančioji funkcija $y = F(x)$ tenkintų sąlygas

$$F^{(l)}(x_i) = f^{(l)}(x_i), \quad i = \overline{0, n}, \quad l = \overline{0,\ m_i - 1}. \qquad (5.2)$$

Simbolis (l) žymi l-tosios eilės išvestinę. Šios sąlygos vadinamos Ermito interpoliavimo sąlygomis, o funkcija $F(x)$ — Ermito interpoliacine funkcija.

Kaip matyti iš interpoliavimo uždavinio formuluotės, keičiant aproksimuojančiųjų funkcijų klasę K bei interpoliavimo sąlygas, galima rasti įvairias interpoliacines funkcijas.

Istoriškai pirmoji aproksimuojančiųjų funkcijų klasė buvo ***n-tosios eilės polinomų*** klasė. Pastaruoju metu plačiai naudojama ***splainų*** klasė.

Splainai tai funkcijos, sudarytos iš polinomų dalių. Jei plokštumoje pažymėsime keletą taškų ir per juos "išraitysime" standžią plieninę liniuotę, tai liniuotė įgaus kubinio splaino formą. Kubinis splainas yra sudarytas iš kubinio polinomo dalių. Liniuotės forma tarp gretimų taškų bus kubinis polinomas. Aišku, kad kubiniai polinomai tarp skirtingų gretimų taškų porų bus skirtingi, tačiau polinomų susidūrimo taškuose tų polinomų ir jų išvestinių iki antros eilės imtinai reikšmės iš kairės ir dešinės bus lygios. Kalbant griežtai, turėsime ***kubinį defekto vienas splainą***. Defektas parodo kiek susidūrimo taškuose yra nelygių aukščiausios eilės išvestinių.

Pradžioje nagrinėsime interpoliacinių polinomų konstravimo bei jų reikšmių apskaičiavimo uždavinius, o po to – analogiškus interpoliacinių splainų uždavinius.

5.1.1. Interpoliacinių polinomų konstravimas ir jų reikšmių apskaičiavimas

Uždavinio formulavimas. *Duota funkcijos $f(x)$ reikšmių lentelė (x_i, y_i); čia $i = \overline{0, n}$. Reikia rasti n-tosios eilės polinomą $y = F(x)$, tenkinantį Lagranžo interpoliavimo sąlygas.*

Interpoliaciniai polinomai paprastai naudojami funkcijos reikšmėms tabuliuoti. Pavyzdžiui, elementariosios funkcijos reikšmės apskaičiuojamos keliuose taškuose, o jos reikšmės kituose taškuose randamos pagal interpoliacinį polinomą. Pastaruoju metu, kai skaičiuojama kompiuteriais, toks elementariųjų funkcijų reikšmių ieškojimo būdas taikomas retai. Todėl gali kilti klausimas, kodėl čia nagrinėjamas interpoliacinių polinomų apskaičiavimas. Atsakymai gali būti keli:

- interpoliaciniai polinomai naudojami ne tik elementariųjų funkcijų reikšmėms apskaičiuoti, bet ir eksperimento duomenims apdoroti;
- interpoliaciniai polinomai naudojami skaitiniam integravimui, diferencijavimui, diferencialinėms lygtims spręsti ir kt.;
- interpoliaciniai polinomai yra puikus glodžiųjų kreivių konstravimo uždavinio sprendimo įvadas.

Pirmasis klausimas, į kurį reikia atsakyti, yra toks: ar interpoliacinio polinomo konstravimo ir jo reikšmės apskaičiavimo uždavinys turi sprendinį ir ar šis sprendinys yra vienintelis?

Tarkime, kad taškai x_i $(i = \overline{0, n})$ yra skirtingi, t. y. $x_i \neq x_j$, kai $i \neq j$. Tada n-tosios eilės interpoliacinis polinomas egzistuoja ir yra vienintelis.

Sakykime,

$$F(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

yra ieškomasis interpoliacinis polinomas. Jis turi tenkinti (5.1) sąlygas, t. y.

$$a_n x_i^n + a_{n-1} x_i^{n-1} + \ldots + a_1 x_i + a_0 = y_i, \quad i = \overline{0, n}. \qquad (5.3)$$

Iš šios tiesinių lygčių sistemos apskaičiuojami interpoliacinio polinomo koeficientai. Kadangi sistemos determinantas yra Vandermondo determinantas, kuris nelygus nuliui, kai $x_i \neq x_j$ $(i \neq j)$, tai sistemos sprendinys egzistuoja ir yra vienintelis.

Nors interpoliacinis polinomas yra vienintelis, tačiau istoriškai susiformavo nemažai jo išraiškų: Lagranžo, Aitkeno, Niutono, Gauso ir kt. Toliau aptarsime Lagranžo, Aitkeno ir Niutono interpoliacines išraiškas, o taip pat interpoliacinių polinomų koeficientų apskaičiavimo uždavinį tiek vienanarių: $1, x, x^2, \cdots, x^n$ bazėje, tiek Čiobyšovo polinomų: $T_0(x), T_1(x), T_2(x), \cdots, T_n(x)$ bazėje.

5.1.1.1. *Lagranžo interpoliacinė išraiška.*

Interpoliacinį polinomą konstruoti remiantis (5.3) lygčių sistema būtų labai neracionalu. Todėl Lagranžas pasiūlė kitokį interpolacinio polinomo apskaičiavimo metodą.

Išnagrinėkime bazinius polinomus

$$F_i(x) = \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}, \; i = \overline{0, n}.$$

Jie yra n-os eilės polinomai ir turi tokią savybę: $F_i(x_j) = \begin{cases} 1, & \text{kai } i = j, \\ 0, & \text{kai } i \neq j; \end{cases}$

čia $j \in \{0, 1, \ldots, n\}$.

Pasinaudodami polinomais $F_i(x)$, galime iš karto parašyti interpoliacinio polinomo $F(x)$ išraišką:

$$F(x) = \sum_{i=0}^{n} F_i(x) \cdot y_i . \qquad (5.4)$$

Šis polinomas interpoliacinis, nes jis yra n-tosios eilės polinomas (n-os eilės polinomų tiesinis darinys yra n-o laipsnio polinomas) ir $F(x_i) = y_i, \, i = \overline{0, n}$. Interpoliacinio polinomo (5.4) išraiška vadinama Lagranžo interpoliacine išraiška, todėl literatūroje simbolis $F(x)$ dažnai keičiamas simboliu $L_n(x)$.

$L_n(x)$ trumpiau galima užrašyti taip: $L_n(x) = \sum\limits_{i=0}^{n} \prod\limits_{\substack{j=0,\\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} \cdot y_i$.

Pavyzdžiui, $L_3(x)$ išraiška yra tokia:

$$L_3(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1 +$$

$$+ \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3 .$$

Žemiau pateikiama funkcija ***lagranz***, kuri įgalina apskaičiuoti interpoliacinio polinomo reikšmes nurodytoms argumento reikšmėms.

```
function fv=lagranz(x,y,t)
% LAGRANZ apskaičiuoja interpoliacinio polinomo,
% nusakyto interpoliavimo taškais (x(i),y(i)),i=1,2,...,n+1),
% reikšmes fv, kai argumento reikšmes apibrėžia masyvo t elementai.
% Polinomo reikšmės skaičiuojamos pagal Lagranžo iterpoliacinę formą.

n=numel(x); m=numel(t); [k,l]=size(x);
if k ~=1
  x=x'; y=y';
end
fv=zeros(1,m);
```

```
for i=1:n
    p=ones(1,m);
    for j=1:n
        if i~=j
            p=p.*(t-x(j))/(x(i)-x(j));
        end;
    end;
    fv=fv+p.*y(i);
end
```

1 Pavyzdys. Pasinaudodami funkcija ***lagranz***, apskaičiuokime funkcijos $y = \sin(x)$, nusakytos reikšmių lentele:

x= 0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0,

y=0; 0.4794255; 0.8414710; 0.99749410; 0.9092974; 0.5984721; 0.1411200,

reikšmes *fv*, kai x = 0.15, 0.75 ir $\frac{\pi}{2}$.

```
>> x=[0 0.5 1.0 1.5 2.0 2.5 3.0];
>> y=sin(x);
>> t=[0.15 0.75 pi/2];
>>  fv=lagranz(x,y,t)
fv =
    0.1495    0.6816    1.0000
```

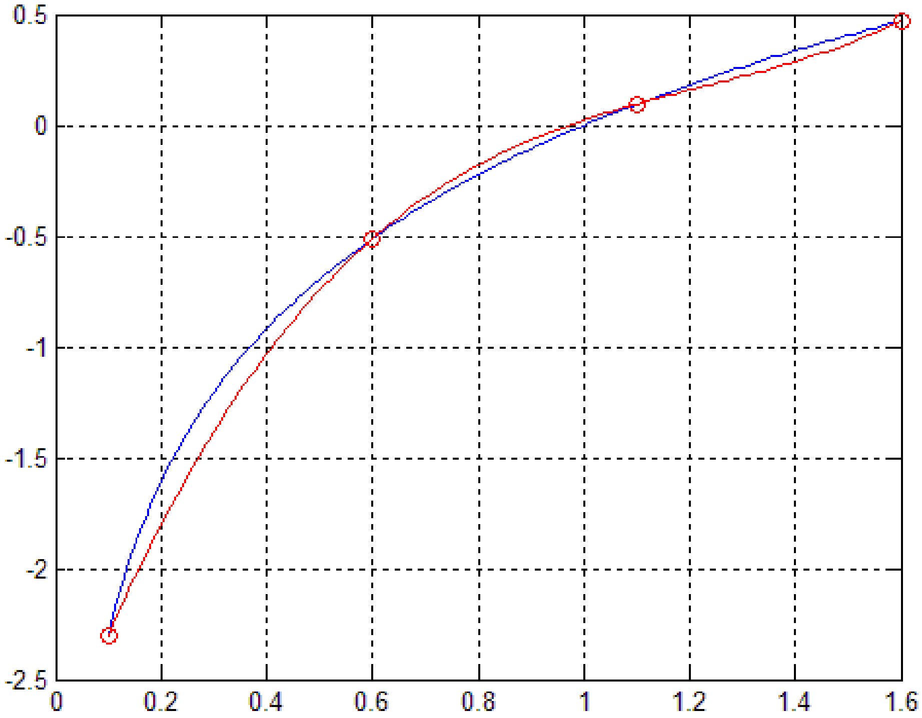
2 Pavyzdys. Funkciją *y=ln* (*x*) aproksimuokime inerpoliaciniu polinomu, kai duoti interpoliavimo mazgai: *x*= 0.1, 0.6, 1.1, 1.6.

Interpoliacinio polinomo ir *y=ln* (*x*) grafikai pavaizduoti 5.1 pav.

5.1 paveikslo grafikus išbrėšime atlikę tokią komandų seką:

```
>> x= [0.1, 0.6, 1.1, 1.6];
>> y=log(x);
>> t=linspace(0.1,1.6,300);
>> yt=log(t);
>> fv=lagranz(x,y,t);
>> plot(t,yt)
>> grid on
>> hold on
>> plot(t,fv,'r')
>> hold on
>> plot(x,y,'ro')
```

Pradžioje apskaičiuojame interpoliavimo taškus *x* ir *y*. Po to apskaičiuojame argumento reikšmių masyvą *t,* o taip pat šioms argumento reikšmėms atitinkančias tikslias natūrinio logaritmo ir interpoliacinio polinomo reikšmes: *yt* ir *fv* . Toliau brėžiami natūrinio logaritmo ir interpoliacinio polinomo grafikai bei pavaizduojamas koordinatinis tinklelis ir interpoliavimo taškai.



5.1 pav. Funkcijos y=ln (x) ir interpoliacinio polinomo grafikai (y=ln (x) grafikas mėlynas, polinomo – raudonas)

5.1.1.2. Aitkeno interpoliacinė išraiška.

Aitkeno interpoliacinei išraiškai būdinga tai, kad interpoliacinis polinomas neužrašomas išreikštiniu pavidalu, o jo reikšmė apskaičiuojama naudojant antrosios eilės determinantus.

Aitkeno išraiškos tiesinė interpoliacija

Duota: taškai $(x_0;y_0)$ ir $(x_1;y_1)$.

Reikia parašyti: per tuos taškus einančios tiesės lygtį.

Tokios tiesės Aitkeno išraiškos lygtis yra

$$P_{01}(x)=\frac{1}{x_1-x_0}\begin{vmatrix} y_0 & x_0-x \\ y_1 & x_1-x \end{vmatrix}.$$

$P_{01}(x)$ yra pirmojo laipsnio polinomas, be to,

$$P_{01}(x_0)=\frac{1}{x_1-x_0}\begin{vmatrix} y_0 & x_0-x_0 \\ y_1 & x_1-x_0 \end{vmatrix}=y_0 \text{ ir } P_{01}(x_1)=\frac{1}{x_1-x_0}\begin{vmatrix} y_0 & x_0-x_1 \\ y_1 & x_1-x_1 \end{vmatrix}=y_1.$$

Aitkeno išraiškos kvadratinė interpoliacija

Duota: taškai $(x_0;\, y_0)$, $(x_1;\, y_1)$ ir $(x_2;\, y_2)$.

Reikia parašyti: per tuos taškus einančios kvadratinės parabolės lygtį.

Tokios parabolės lygtis nustatoma taip:

$$P_{01}(x_1)=\frac{1}{x_1-x_0}\begin{vmatrix} y_0 & x_0-x \\ y_1 & x_1-x \end{vmatrix},\ P_{12}(x)=\frac{1}{x_2-x_1}\begin{vmatrix} y_1 & x_1-x \\ y_2 & x_2-x \end{vmatrix},$$

$$P_{012}(x)=\frac{1}{x_2-x_0}\begin{vmatrix}P_{01}(x) & x_0-x\\ P_{12}(x) & x_2-x\end{vmatrix}.$$ $P_{012}(x)$ — ieškomoji kvadratinė parabolė, nes

$P_{012}(x)$ yra kvadratinis polinomas, be to,

$$P_{012}(x_0)=\frac{1}{x_2-x_0}\begin{vmatrix}P_{01}(x_0) & x_0-x_0\\ P_{12}(x_0) & x_2-x_0\end{vmatrix}=y_0,$$

$$P_{012}(x_1)=\frac{1}{x_2-x_0}\begin{vmatrix}P_{01}(x_1) & x_0-x_1\\ P_{12}(x_1) & x_2-x_1\end{vmatrix}=\frac{1}{x_2-x_0}\begin{vmatrix}y_1 & x_0-x_1\\ y_1 & x_2-x_1\end{vmatrix}=$$

$$=\frac{1}{x_2-x_0}y_1\big(x_2-x_1-(x_0-x_1)\big)=y_1,$$

$$P_{012}(x_2)=\frac{1}{x_2-x_0}\begin{vmatrix}P_{01}(x_2) & x_0-x_2\\ P_{12}(x_2) & x_2-x_2\end{vmatrix}=y_2.$$

Aitkeno interpoliacinė išraiška pagrįsta Aitkeno teorema.

Aitkeno teorema . *Tarkime* $(x_i, y_i), i=\overline{0,n}$ *yra interpoliavimo taškai. Jeigu*

1. $P_{n-1}^{i}(x)$ *yra* $(n-1)-$*os eilės interpoliacinis polinomas, kurį apibrėžia interpoliavimo mazgai* $\{x_0, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n\}$, *o*
2. $P_{n-1}^{j}(x)$ *yra* $(n-1)-$*os eilės interpoliacinis polinomas, kurį apibrėžia interpoliavimo mazgai* $\{x_0, x_1, \cdots, x_{j-1}, x_{j+1}, \cdots, x_n\}$,
3. $i\neq j$,

tai $P_n(x)$ *- n-to laipsnio interpoliacinis polinomas, kurį nusako visi interpoliavimo taškai* $(x_i, y_i), i=\overline{0,n}$, *apskaičiuojamas pagal formulę*

$$P_n(x)=\frac{x_j-x}{x_i-x_j}P_{n-1}^{i}(x)-\frac{x_i-x}{x_i-x_j}P_{n-1}^{j}(x)\ \ arba\ \ P_n(x)=\frac{1}{x_i-x_j}\begin{vmatrix}P_{n-1}^{i}(x) & x_i-x\\ P_{n-1}^{j}(x) & x_j-x\end{vmatrix}.$$

Aitkeno išraiškos n-tosios eilės interpoliacija

Duota: n — interpoliacinio polinomo eilė,

$(x_i;\, y_i)$ — interpoliavimo taškai $\left(i=\overline{0,n}\right)$;

a — argumento reikšmė; $a\in[x_0;\, x_n]$.

Reikia rasti: n-tosios eilės interpoliacinio polinomo reikšmę, kai argumento reikšmė lygi a, t. y. $P=P_n(a)$.

Tada, remiantis Aitkeno teorema, ieškomoji reikšmė apskaičiuojama pagal tokias Nevilio (E. H. Nevilie) formules:

$P_i=y_i, \quad i=\overline{0,n}$,

$$P_i=\frac{1}{x_{i+j}-x_i}\begin{vmatrix}P_i & x_i-a\\ P_{i+1} & x_{i+j}-a\end{vmatrix},\quad \begin{array}{l} j=\overline{1,\, n},\\ i=\overline{0,\, n-j}.\end{array}$$

$P_n(a)=P_0$.

Pavyzdžiui, kai $n=2$, skaičiavimas pagal šias formules pateiktas 5.1 lentelėje.

5.1 lentelė. Aitkeno išraiškos kvadratinio interpoliacinio polinomo reikšmės apskaičiavimas

i	P_i	$j=1$	$j=2$
0	$P_0=y_0$	$P_0=\dfrac{1}{x_1-x_0}\begin{vmatrix} y_0 & x_0-a \\ y_1 & x_1-a \end{vmatrix}$	$P_0=\dfrac{1}{x_2-x_0}\begin{vmatrix} P_0 & x_0-a \\ P_1 & x_2-a \end{vmatrix}$
1	$P_1=y_1$	$P_1=\dfrac{1}{x_2-x_1}\begin{vmatrix} y_1 & x_1-a \\ y_2 & x_2-a \end{vmatrix}$	
2	$P_2=y_2$		

Žemiau pateikta Nevilio metodą realizuojanti funkcija ***aitken***.

```
   function fv=aitken(x,y,t);
% AITKEN apskaičiuoja interpoliacinio polinomo,
% nusakyto interpoliavimo taškais (x(i),y(i)),i=1,2,...,n+1),
% reikšmes fv, kai argumento reikšmes apibrėžia masyvo t elementai.
% Polinomo reikšmės skaičiuojamos pagal Aitkeno iterpoliacinę formą.
% Įėjimo parametrai
% (x,y)      - interpoliavimo taškai,
%  t         - argumento reikšmių masyvas.
% Išėjimo parametrai
% fv       - interpoliacinio polinomo reikšmės.

   n=numel(x)-1; m=numel(t); [k,l]=size(x);
   if k == 1
       x=x'; y=y';
   end
   xx=repmat(x,1,m); p=repmat(y,1,m);

   for j=1:n
       for i=1:n+1-j
           p(i,:)=(p(i,:).*(xx(i+j,:)-t)-p(i+1,:).*...
                  (xx(i,:)-t))./(xx(i+j,:)-xx(i,:));
       end;
   end;
   fv=p(1,:);
```

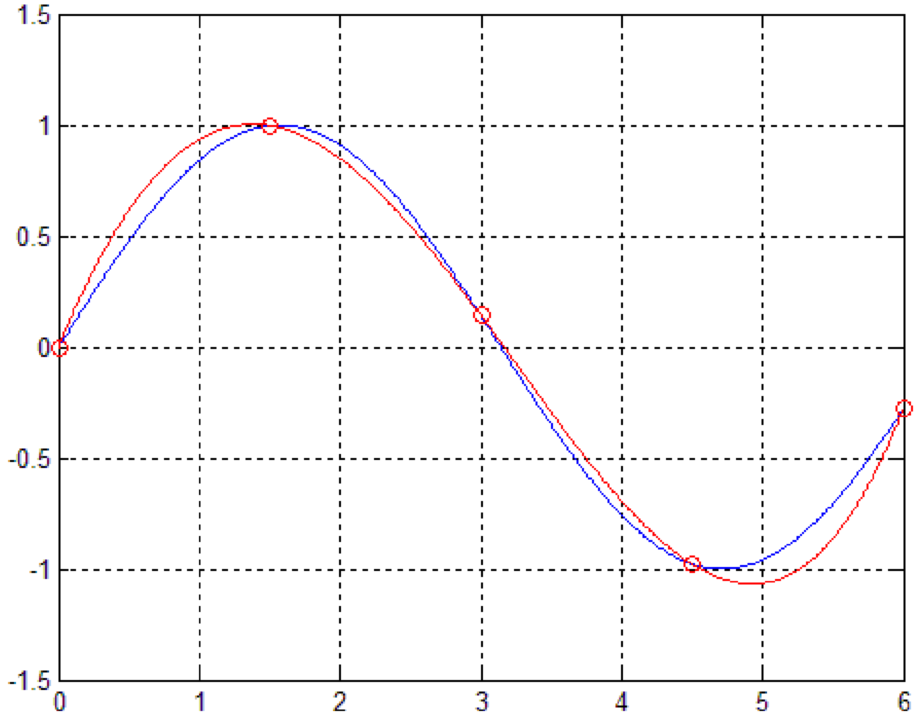
1. ***Pavyzdys.*** Funkciją $y=sin\,(x)$ aproksimuokime inerpoliaciniu polinomu, kai duoti interpoliavimo mazgai: *x=0, 1.5, 3.0, 4.5, 6.0.*

Interpoliacinio polinomo ir $y=sin(x)$ grafikai pavaizduoti 5.2 pav.

5.2 paveikslo grafikus išbrėšime atlikę tokią komandų seką:

```
>> x= [0 1.5 3.0 4.5 6.0];
>> y=sin(x);
>> t=linspace(0,6,300);
>> yt=sin(t);
>> fv=aitken(x,y,t);
>> plot(t,yt)
>> grid on
>> hold on
>> plot(t,fv,'r')
```

```
>> hold on
>> plot(x,y,'ro')
```



5.2 pav. Funkcijos y=sin (x) ir interpoliacinio polinomo grafikai (y=sin (x) grafikas mėlynas, polinomo – raudonas)

5.1.1.3. Interpoliacinis polinomas vienanarių bazėje

Matematikoje standartinė polinomų išraiška yra:

$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$.

Nors ši išraiška taikoma daugiau teorijoje, tačiau dažnai ji būna naudinga ir praktikoje. Todėl čia panagrinėsime, kaip apskaičiuoti interpoliacinio polinomo išraišką vienanarių bazėje.

Tarkime, kad interpoliacinį polinomą $P_n(x)$ *apibrėžia interpoliavimo taškai* $(x_i, y_i), i = \overline{0,n}$. *Reikia apskaičiuoti šio polinomo koeficientus* $a_i, i = \overline{0,n}$.

Koeficientus apskaičiuosime remdamiesi Aitkeno teorema ir Nevilio formulėmis.

Iš Aitkeno teoremos matyti, kad *n-to* laipsnio interpoliacinis polinomas yra dviejų (*n-1*)-jo laipsnio interpoliacinių polinomų tiesinis darinys, kai darinio koeficientai yra pirmo laipsnio polinomai. Vadinasi, *n-to* laipsnio interpoliacinio polinomo koeficientai yra apskaičiuojami remiantis formulę:

$$P_n(x) = \left(\left(P_{n-1}^j(x) - P_{n-1}^i(x)\right)x + P_{n-1}^i(x)x_j - P_{n-1}^j(x)x_i\right) / \left(x_i - x_j\right).$$

Kaip matyti iš formulės, *n-to* laipsnio interpoliacinio polinomo koeficientai yra apskaičiuojami taip:

- Iš polinomo $P_{n-1}^j(x)$ koeficientų, išdėstytų pradedant vyriausiuoju koeficientu, masyvo panariui atimame analogišką polinomo $P_{n-1}^i(x)$ koeficientų masyvą.

- Gautąjį koeficientų skirtumo masyvą papildome lygiu nuliui elementu, kuris talpinamas masyvo gale. Šį masyvą pažymėkime $m1$.
- Polinomo $P_{n-1}^{i}(x)$ koeficientus padauginame iš daugiklio x_j ir iš jų panariui atimame polinomo $P_{n-1}^{j}(x)$ koeficientus, padaugintus iš daugiklio x_i.
- Šį koeficientų skirtumo masyvą papildome lygiu nuliui elementu, kuris talpinamas masyvo pradžioje. Šį masyvą pažymėkime $m2$.
- Tada polinomo $P_n(x)$ koeficientų masyvas bus lygus ($(m1-m2)/(x_i-x_j)$.

Žemiau pateikta funkcija ***koef***, kuri apskaičiuoja interpoliacinio polinomo koeficientus.

```
function kfn=koef(x,y);
% KOEF apskaičiuoja interpoliacinio polinomo, kuri apibrėžia
% interpoliavimo taškai (x,y),koeficientus.
% Įėjimo parametrai
% (x,y)      - interpoliavimo taškai.
% Išėjimo parametrai
% kfn        - interpoliacinio polinomo koeficientai,
%              pradedant vyriausiuoju.

n=numel(x)-1; [k,l]=size(x);
if k == 1
    x=x'; y=y';
end
p=zeros(n+1);
p(:,1)=y;
for k=1:n
    for i=1:n+1-k
        p(i,1:k+1)=([p(i+1,1:k)-p(i,1:k), 0]+...
                    [0, p(i,1:k)*x(i+k)-p(i+1,1:k)*x(i)])/...
                    (x(i+k)-x(i));
    end;
end;
kfn=p(1,:);
```

Pavyzdys. Panagrinėkime uždavinį, kurio sprendinys žinomas. Interpoliuokime funkciją $y = x^5 - 4x^4 + 7x^3 + 2x - 5$, imdami interpoliavimo mazgus $x = -2,-1,01,2,3$. Aišku, interpoliacinis polinomas sutaps su funkcija *y*.

```
>> x=[-2 -1 0 1 2 3];
>> y=x.^5-4*x.^4+7*x.^3+2*x-5;
>> kfn=koef(x,y)
kfn =
     1    -4     7     0     2    -5
```

Kaip matome, funkcija ***koef*** teisingai apskaičiavo interpoliacinio polinomo koeficientus.

Norėdami apskaičiuoti polinomo $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ reikšmę, taikysime 6.1 paragrafe išnagrinėtą Hornerio schemą.

5.1.1.4. Interpoliaciniai polinomai Čiobyšovo polinomų bazėje

Apskaičiuojant interpoliacinio polinomo koeficientus, paklaidos bus žymiai mažesnės (uždavinys bus geriau sąlygotas), jei interpoliaciniams polinomams užrašyti naudosime Čiobyšovo ortogonaliųjų polinomų bazę.

Čiobyšovo polinomai. n-tojo laipsnio Čiobyšovo polinomas apibrėžiamas taip: $T_n(x) = \cos(n \cdot \arccos x)$; čia $n = 0, 1, 2, \ldots$. Vadinasi, $-1 \le x \le 1$ ir $-1 \le T_n(x) \le 1$.

$T_n(x)$ tikrai yra n-tojo laipsnio polinomas. Iš tikrųjų

$T_0(x) = 1;$

$T_1(x) = \cos(\arccos x) = x;$

$T_2(x) = \cos(2\arccos x) = \cos^2(\arccos x) - \sin^2(\arccos x) = 2x^2 - 1$.

n-to laipsnio Čiobyšovo polinomai intervale* [*-1; 1*] *turi n realiųjų šaknų: $x_i = \cos\left(\frac{2i-1}{2n}\pi\right), i = \overline{1,n}$***, o taškuose*** $x_k = \cos\left(\frac{k}{n}\pi\right), k = 0,1,2,\cdots.n$**.**

Daugelis Čiobyšovo polinomų savybių išvedamos iš trigonometrinių funkcijų tapatybių. Pavyzdžiui, Čiobyšovo polinomai tenkina rekurenčiąją lygtį

$$T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x),$$

kuri gaunama iš trigonometrinės tapatybės

$$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos\theta\cos n\theta, \text{ kai } \theta = \arccos x.$$

Ši rekurenčioji lygtis naudojama norimos eilės Čiobyšovo polinomams generuoti. Šiam tikslui ją patogiau užrašyti taip:

$$T_0(x) = 1; \quad T_1(x) = x;$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \ge 1. \qquad (5.5)$$

Remdamiesi (5.5) lygybe, galime įrodyti, kad vyriausiasis polinomo $T_n(x)$ narys yra $2^{n-1}x^n$ ir $T_n(x)$ turi tik lyginio arba nelyginio laipsnio narių, priklausančių nuo to, koks yra n: lyginis ar nelyginis.

Čiobyšovo polinomai yra ortogonalūs su svoriu $\frac{1}{\sqrt{1-x^2}}$, t. y.

$$\int_{-1}^{1} \frac{T_n(x)\cdot T_m(x)}{\sqrt{1-x^2}}dx = \begin{cases} 0, & \text{kai} \quad n \ne m, \\ \frac{\pi}{2}, & \text{kai} \quad n = m \ne 0, \\ \pi, & \text{kai} \quad n = m = 0. \end{cases}$$

Eilutė $f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$ vadinama Čiobyšovo polinomų eilute.

Remiantis ortogonalumo savybe, nesunku įrodyti, kad

$$a_0 = \frac{1}{\pi}\int_{-1}^{1} \frac{f(x)\,dx}{\sqrt{1-x^2}}, \quad a_k = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)\cdot T_k(x)}{\sqrt{1-x^2}}dx, \quad k \ge 1.$$

Aišku, kad x laipsnius $x^k\,(k = 0,1,2,\cdots)$ galima pakeisdami Čiobyšovo polinomų išraiškomis:

$1 = T_0,$ $\quad$ $x^4 = 1/8(3T_0 + 4T_2 + T_4),$

$x = T_1,$ $\quad$ $x^5 = 1/16(10T_1 + 5T_3 + T_5),$

$x^2 = 1/2(T_0 + T_1),$ $\quad$ ....

$x^3 = 1/4(3T_1 + T_3),$

Vadinasi, bet koks polinomas gali būti užrašytas tiesiniu Čiobyšovo polinomų dariniu:

$$f(x) = a_0 + a_1T_1(x) + a_2T_2(x) + \cdots + a_nT_n(x)\,.$$

Tarkime, kad interpoliacinį polinomą $P_n(x)$ *apibrėžia interpoliavimo taškai* $(x_i, y_i), i = \overline{0, n}$. *Reikia apskaičiuoti šio polinomo koeficientus* $a_i, i = \overline{0, n}$ *Čiobyšovo polinomų bazėje.*

Tokio polinomo koeficientus apskaičiuosime remdamiesi Aitkeno teorema ir rekrenčiąja formule $T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x)$.

Aitkeno teorema yra invariantiška polinomų bazės atžvilgiu. Šią teoremą taikant, atsiranda $xT_n(x)$ tipo sandaugos, kurias, remdamiesi rekrurenčiąja formule $T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x)$, keičiame išraiška: $xT_n(x) = (T_{n+1}(x) + T_{n-1}(x))/2$.

Žemiau pateikta funkcija ***cebkoef***, kuri apskaičiuoja interpoliacinio polinomo koeficientus Čiobyšovo bazėje.

```
function kfn=cebkoef(x,y);
% CEBKOEF apskaičiuoja interpoliacinio polinomo, kuri apibrėžia
% interpoliavimo taškai (x,y),koeficientus Čiobyšovo bazėje.
% Įėjimo parametrai
% (x,y)     - interpoliavimo taškai. x∈[-1;1].
% Išėjimo parametrai
% kfn       - interpoliacinio polinomo koeficientai,
%             pradedant vyriausiuoju.

n=numel(x)-1; [k,l]=size(x);
if k == 1
    x=x'; y=y';
end
p=zeros(n+1);
p(:,1)=y;
for k=1:n
    for i=1:n+1-k
     p(i,1:k+1)=([p(i+1,1:k-1)/2 p(i+1,k) 0]+[0 0 p(i+1,1:k-1)/2]-...
                 [p(i,1:k-1)/2 p(i,k) 0]-[0 0 p(i,1:k-1)/2]+...
                 [0 p(i,1:k)*x(i+k)-p(i+1,1:k)*x(i)])/(x(i+k)-x(i));
    end;
end;
kfn=p(1,:);
```

Norėdami apskaičiuoti polinomo $f(x) = a_0 + a_1T_1(x) + a_2T_2(x) + \cdots + a_nT_n(x)$ reikšmę, taikysime Klenšou metodą.

Klenšou metodas. Reikia rasti sumą $y = \sum_{k=0}^{n} a_k T_k(x)$. Tokios sumos apskaičiavimo metodą pasiūlė Klenšou (*Clenshow C. W).*

Klenšou metodas remiasi iš (5.5) formulės gauta tapatybe

$$T_{n+1} - 2xT_n + T_{n-1} = 0.$$

(Prastindami simboliką, čia ir toliau $T_n(x)$ žymėsime T_n.)

Koeficientus a_k pakeisime taip:

$$a_k = b_k - 2xb_{k+1} + b_{k+2} \quad \left(k = \overline{n, 0}\right),$$

$$b_{n+1} = b_{n+2} = 0.$$

Tada

$$y = \sum_{k=0}^{n} a_k T_k = \sum_{k=0}^{n} \left(b_k - 2xb_{k+1} + b_{k+2}\right)T_k =$$

$$= b_{n+2}T_n - 2xb_{n+1}T_n + b_nT_n + b_{n+1}T_{n-1} - 2xb_nT_{n-1} + b_{n-1}T_{n-1} +$$

$$+ b_nT_{n-2} - 2xb_{n-1}T_{n-2} + b_{n-2}T_{n-2} + \ldots + b_4T_2 - 2xb_3T_2 + b_2T_2 +$$

$$+ b_3T_1 - 2xb_2T_1 + b_1T_1 + b_2T_0 - 2xb_1T_0 + b_0T_0 =$$

$$= \left(T_n - 2xT_{n-1} + T_{n-2}\right)b_n + \left(T_{n-1} - 2xT_{n-2} + T_{n-3}\right)b_{n-1} + \ldots +$$

$$+ \left(T_2 - 2xT_1 + T_0\right)b_2 + b_1T_1 - 2xb_1T_0 + b_0T_0 =$$

$$= b_1x - 2xb_1 + b_0 = b_0 - b_1x.$$

Vadinasi,

$$y = \sum_{k=0}^{n} a_k T_k = b_0 - b_1x;$$

čia $b_k = a_k + 2xb_{k+1} - b_{k+2}\left(k = \overline{n, 0}\right),$ (5.6)

$$b_{n+1} = b_{n+2} = 0.$$

(5.6) formulės ir apibrėžia Klenšou metodą.

Pavyzdys. Raskime sumą $y = 5T_4(x) - 2T_2(x) + 6T_1(x) + 3T_0(x)$, kai $x = 1/2$.

Sumos skaičiavimas Klenšou metodu, parodytas 5.2 lentelėje.

5.2 lentelė. Sumos $y = 5T_4(x) - 2T_2(x) + 6T_1(x) + 3T_0(x)$ apskaičiavimas Klenšou metodu

a			5	0	–2	6	3
b	0	0	5	5	–2	–1	4

Tada $y = b_0 - b_1x = 4 - (-1) \cdot 1/2 = 9/2$.

```
function px=klensou(a,x);
% KLENSOU apskaičiuoja interpoliacinio polinomo,
% užrašyto Čiobyšovo polinomo bazėje, reikšmę.
% Įėjimo parametrai
% a         - polinomo koeficientai,
% x         - argumento reikšmių masyvas.
% Išėjimo parametrai
% px        - polinomo reikšmių masyvas.

n=numel(a);
bk2=0; bk1=0;
for k=1:n
```

```
     bk=a(k)+2*x.*bk1-bk2;
     bk2=bk1; bk1=bk;
     end;
px=bk-x.*bk2;
```

Pavyzdys. Apskaičiuokime funkcijos $y = x^5 - 5x^4 + 2x^3 + x^2 - 6x + 1$ interpoliacinį polinomą Čiobyšovo bazėje, kurį apibrėžia mazgai: $x = -2,-1,0,1,2,3$, o taip pat apskaičiuokime funkcijos ir interpoliacinio polinomo reikšmes taškuose: *xx*= -1.5, -0.5, 0.5, 1.5, 2.5.

Šį pavyzdį išspręsime atlikę žemiau pateiktą komandų seką.

```
>> x=[-2 -1 0 1 2 3]
>> y=x.^5-5*x.^4+2*x.^3+x.^2-6*x+1
>> a=x(1); b=x(end);
>> xt=(2*x-(b+a))/(b-a);% mazgus transformuojame į intervalą [-1;1]
>> kfn=cebkoef(xt,y)
>> xx=(x(1:end-1)+x(2:end))/2
>> t=(2*xx-(b+a))/(b-a);% xx transformuojame į intervalą [-1;1]
>> px=klensou(kfn,t)
>> yxx=xx.^5-5*xx.^4+2*xx.^3+xx.^2-6*xx+1
```

Gauti rezultatai.

```
x = -2    -1     0     1     2     3
y = -111     0     1    -6   -39  -116
kfn =  6.1035  -12.2070    9.0332  -55.8594  -17.6367  -45.4336
xx  = -1.5000   -0.5000    0.5000    1.5000    2.5000
px  = -27.4063    3.6562   -1.7813  -16.7188  -74.1563
yxx = -27.4063    3.6563   -1.7813  -16.7188  -74.1563
```

Kaip ir reikėjo tikėtis, interpoliacinio polinomo ir funkcijos reikšmės mazguose *xx,*- sutapo.

5.1.1.5. Niutono interpoliacinė forma

Prieš pradėdami nagrinėti Niutono interpoliacinę išraišką, aptarkime skirtuminio santykio sąvoką. Sakykime, turime funkcijos $y = f(x)$ reikšmių lentelę $(x_i, f(x_i))$, $i = \overline{0, n}$. Nulinės eilės skirtuminiai santykiai yra $f(x_i)$, $i = \overline{0, n}$. Pirmosios eilės skirtuminiai santykiai apibrėžiami taip: $f(x_i, x_j) = \dfrac{f(x_j) - f(x_i)}{x_j - x_i}$, o antrosios eilės — taip:

$$f(x_i, x_j, x_k) = \frac{f(x_j, x_k) - f(x_i, x_j)}{x_k - x_i}.$$

Apskritai *k*-tosios eilės skirtuminiai santykiai nusakomi formule

$$f(x_1, x_2, \ldots, x_k, x_{k+1}) = \frac{f(x_2, \ldots, x_k, x_{k+1}) - f(x_1, x_2, \ldots, x_k)}{x_{k+1} - x_1}.$$

n-tosios eilės Lagranžo išraiškos interpoliacinį polinomą galime užrašyti taip:

$$L_n(x) = L_0(x) + (L_1(x) - L_0(x)) + (L_2(x) - L_1(x)) + \ldots + (L_n(x) - L_{n-1}(x)).$$

Aišku, kad $L_m(x)-L_{m-1}(x)$ yra m-tosios eilės polinomas, turintis šaknis $x_0, x_1, \ldots, x_{m-1}$, nes $L_m(x_j)=L_{m-1}(x_j)$, kai $j=\overline{0, m-1}$. Vadinasi,

$$L_m(x)-L_{m-1}(x)=a_{m-1}\omega_{n-1}(x)=a_{m-1}(x-x_0)\ldots(x-x_{m-1}).$$

Kadangi $L_m(x_m)=f(x_m)$, tai, remdamiesi (5) lygybe, gauname:

$$L_m(x_m)-L_{m-1}(x_m)=f(x_m)-L_{m-1}(x_m)=f(x_m, x_0, \ldots, x_{m-1})\omega_{m-1}(x_m).$$

Antra vertus,

$L_m(x_m)-L_{m-1}(x_m)=a_{m-1}(x_m-x_0)\ldots(x_m-x_{m-1})=a_{m-1}\omega_{m-1}(x_m)$, todėl

$a_{m-1}\omega_{m-1}(x_m)=f(x_m, x_0, \ldots, x_{m-1})\omega_{m-1}(x_m)$ ir $a_{m-1}=f(x_0, x_1, \ldots, x_{m-1}, x_m)$.

Taigi $L_m(x)-L_{m-1}(x)=f(x_0, x, \ldots, x_m)\omega_{m-1}(x)$. Vadinasi, ***Niutono išraiškos interpoliacinis polinomas yra toks***:

$$F(x)=L_n(x)=f(x_0)+f(x_0, x_1)(x-x_0)+f(x_0, x_1, x_2)(x-x_0)(x-x_1)+$$
$$+\ldots+f(x_0, x_1, \ldots, x_n)(x-x_0)(x-x_1)\ldots(x-x_{n-1}).$$

Praktikoje labai dažnai x_i reikšmės būna išsidėsčiusios tvarkingai, t. y. $x_i=x_0+i\cdot h,\ i=\overline{0, n}$. Šiuo atveju

$f(x_0, x_1, \ldots, x_k)=\dfrac{\Delta^k y_0}{k!h^k}$; čia $\Delta^k y_0$ — k-tosios eilės baigtinis skirtumas.

Baigtiniai skirtumai apibrėžiami taip:

$\Delta y_k=y_{k+1}-y_k,\quad k=\overline{0, n-1}$,

$\Delta^m y_k=\Delta^{m-1}y_{k+1}-\Delta^{m-1}y_k,\quad m=\overline{1, n},\quad k=\overline{0, n-m}$.

Remdamiesi baigtinių skirtumų formule ir pažymėję $\dfrac{x-x_0}{h}=t$, ankstesnę Niutono interpoliacinę išraišką galime užrašyti taip:

$$F(x)=y_0+t\,\Delta y_0+\frac{t(t-1)}{2!}\Delta^2 y_0+\frac{t(t-1)(t-2)}{3!}\Delta^3 y_0+\ldots+\frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0.$$

Žemiau pateikta funkcija ***niuton***, kuri apskaičiuoja interpoliacinio polinomo reikšmes, pagal Niutono interpoliacinę išraišką.

```
function fv=niuton(x,y,t)
% NIUTON apskaičiuoja interpoliacinio polinomo,
% nusakyto interpoliavimo taškais (x(i),y(i)),i=1,2,...,n+1),
% reikšmes fv, kai argumento reikšmes apibrėžia masyvo t elementai.
% Polinomo reikšmės skaičiuojamos pagal Niutono iterpoliacinę formą.
% Įėjimo parametrai
% (x,y)      - interpoliavimo taškai,
%  t         - argumento reikšmių masyvas.
% Išėjimo parametrai
% fv       - interpoliacinio polinomo reikšmės.

n=numel(x)-1; m=numel(t);
[k,l]=size(t);
```

```
if k ==1
  t=t';
end
[k,l]=size(x);
if k ~=1
  x=x'; y=y';
end
d=y;
for k=1:n
    h=x(k+1:end)-x(1:end-k);
    tt=(d(k+1:end)-d(k:end-1))./h;
    d(k+1:end)=tt;
end
xx=repmat(x,m,1); dd=repmat(d,m,1); tt=repmat(t,1,n);
p=tt-xx(:,1:end-1); r=ones(m,1); s=[r cumprod(p,2)];
fv=sum((dd.*s)');
```

Pavyzdys. Apskaičiuokime funkcijos $y = x^5 - 5x^4 + 2x^3 + x^2 - 6x + 1$ ir interpoliacinio polinomo, kurį apibrėžia mazgai: $x = -2,-1,0,1,2,3$, reikšmes taškuose: *xx*= -1.5, -0.5, 0.5, 1.5, 2.5.

Kadangi interpoliuojamoji funkcija yra penkto laipsnio polinomas ir interpoliacinį polinomą nusako šeši mazgai, tai funkcijos ir interpoliacinio polinomo reikšmės privalo sutapti. Atlikę komandas:

```
>> x=linspace(-2,3,6);
>> y=x.^5-5*x.^4+2*x.^3+x.^2-6*x+1;
>> xx=[-1.5 -0.5 0.5 1.5 2.5];
>> fv=niuton(x,y,xx)
>> yxx=xx.^5-5*xx.^4+2*xx.^3+xx.^2-6*xx+1,
```

matysime, kad `fv` ir `yxx` reikšmės sutampa ir yra lygios:

```
fv=yxx=[-27.4063    3.6563   -1.7813  -16.7188  -74.1563].
```

5.1.1.6. Ermito interpoliacinis polinomas

Kaip jau buvo minėta, jei, be $f(x_i)$ reikšmių, mazge x_i žinomos ir $f(x)$ išvestinių iki $(m_i - 1)$-osios eilės imtinai reikšmės, tai galima reikalauti, kad interpoliacinis polinomas $y = F(x)$ tenkintų Ermito interpoliavimo sąlygas

$$F^{(l)}(x_i) = f^{(l)}(x_i), \ i = \overline{0, n}, \ l = \overline{0, m_i - 1};$$

čia simbolis (l) žymi l-tosios eilės išvestinę. Polinomas, tenkinantis šias sąlygas, vadinamas Ermito interpoliaciniu polinomu. Aišku, kad jo laipsnis N turi būti lygus :

$$N = m_0 + m_1 + \ldots + m_n - 1.$$

Toks polinomas egzistuoja ir yra vienintelis, o jo koeficientai apskaičiuojami iš lygčių sistemos, kurią apibrėžia Ermito interpoliavimo sąlygos.

Praktikoje šie polinomai naudojami rečiau, nes, norint juos sukonstruoti, reikia žinoti interpoliuojamosios funkcijos išvestinės reikšmes.

Čia aptarsime Ermito interpoliacinio polinomo išraišką, kurios visi $m_i = 2$ $(i = \overline{0, n})$, taigi sudarysime $(2n + 1)$-osios eilės polinomą $F(x)$, tenkinantį sąlygas

$$F(x_i) = f(x_i) = y_i,$$
$$F'(x_i) = f'(x_i) = y_i', \quad i = \overline{0, n}.$$

Šį polinomą konstruosime panašiai kaip ir Lagranžo polinomą. Iš pradžių rasime ne aukštesnės kaip $(2n+1)$-osios eilės polinomus $H_i(x)$ ir $h_i(x)$, tenkinančius tokias sąlygas: kai $j \neq i$, tai

$H_i(x_j) = 0$, $H_i'(x_j) = 0$, $h_i(x_j) = 0$, $h_i'(x_j) = 0$,

$H_i(x_i) = 1$, $H_i'(x_i) = 0$, $h_i(x_i) = 0$, $h_i'(x_i) = 1$.

Tada ieškomasis Ermito interpoliacinis polinomas bus užrašomas formule

$$F(x) = \sum_{i=0}^{n} \left(H_i(x) y_i + h_i(x) y_i' \right).$$

Pirmiausia sudarysime polinomą $h_i(x)$. Kadangi $h_i(x_j) = 0$ ir $h_i'(x_j) = 0$, tai šis polinomas turi daugiklius $(x - x_j)^2$, $j \neq i$. Jei $h_i(x_i) = 0$, o $h_i'(x_i) \neq 0$, tai jis turi daugiklį $(x - x_i)$. Kad būtų $h_i'(x_i) = 1$, $h_i(x)$ turi įgyti tokią išraišką:

$$h_i(x) = \frac{(x-x_0)^2 (x-x_1)^2 \ldots (x-x_{i-1})^2 (x-x_i)(x-x_{i+1})^2 \ldots (x-x_n)^2}{(x_i-x_0)^2 (x_i-x_1)^2 \ldots (x_i-x_{i-1})^2 (x_i-x_{i+1})^2 \ldots (x_i-x_n)^2}.$$

Lengva įsitikinti, jog $h_i(x)$ tenkina anksčiau išvardytas sąlygas.

Polinomas $H_i(x)$ konstruojamas panašiai. Aišku, kad jis turi daugiklius $(x - x_j)^2$ $(j \neq i)$, bet kartu $H_i(x_i) = 1$ ir $H_i'(x_i) = 0$. $H_i(x_i)$ ieškosime tokios išraiškos:

$$H_i(x) = \frac{(x-x_0)^2 (x-x_1)^2 \ldots (x-x_{i-1})^2 (ax+b)(x-x_{i+1})^2 \ldots (x-x_n)^2}{(x_i-x_0)^2 (x_i-x_1)^2 \ldots (x_i-x_{i-1})^2 (x_i-x_{i+1})^2 \ldots (x_i-x_n)^2}.$$

Sąlyga $H_i(x_i) = 1$ reikalauja, kad būtų $ax_i + b = 1$. Apskaičiuokime $H_i'(x_i)$. Atsižvelgdami į tai, kad $H_i'(x_i) = 0$, turime:

$$H_i'(x_i) = \frac{2}{x_i - x_0} + \frac{2}{x_i - x_1} + \ldots + \frac{2}{x_i - x_{i-1}} + a + \frac{2}{x_i - x_{i+1}} + \ldots + \frac{2}{x_i - x_n} = 0.$$

Vadinasi, $a = -2 \sum_{\substack{j=0, \\ j \neq i}}^{n} \frac{1}{x_i - x_j}$, $b = 1 - ax_i$.

Čia pateiktas Ermito interpoliacinio polinomo sudarymo metodas yra bendras ir gali būti lengvai išplėstas konstruojant Ermito interpoliacinius polinomus, kai $m_i = 3, 4, \ldots$.

Ermito polinomo liekamasis narys lygus

$$\frac{f^{(N+1)}(c)}{(N+1)!} \Omega_n(x); \text{ čia } \Omega_n(x) = (x - x_0)^{m_0} (x - x_1)^{m_1} \ldots (x - x_n)^{m_n}, \text{ o } c \in [x_0, x_n].$$

Nors anksčiau pateikta Ermito interpoliacinio polinomo formulė pilnai apibrėžia Ermito polinomą, tačiau, pagal šią formulę skaičiuojant polinomo reikšmes, reikia atlikti labai daug veiksmų. Todėl praktiškai naudojama racionalesnė Ermito polinomo formulė, kuri, kaip ir Niutono interpoliacinė formulė, pagrįsta skirtuminiais santykiais.

Remdamiesi interpoliavimo mazgais $x_0, x_1, \cdots, x_n$, sudarykime naują mazgų seką: $z_0, z_1, \cdots, z_{2n+1}$, čia $z_{2i} = z_{2i+1} = x_i$, o taip pat tiems mazgams atitinkančią funkcijos reikšmių seką $f(z_0), f(z_1), \cdots, f(z_{2n+1})$, čia $f(z_{2i}) = f(z_{2i+1}) = f(x_i)$ visiems $i = 0,1,\cdots,n$.

Tada ***Ermito interpoliacinio polinomo skirtuminių santykių forma*** yra:

$H_{2n+1} = f(z_0) + \sum_{k=1}^{2n+1} f(z_0, z_1, \cdots, z_k)(x - z_0)(x - z_1)\cdots(x - z_{k-1})$, kur $z_{2k} = z_{2k+1} = x_k$ ir $f(z_{2k}, z_{2k+1}) = f'(x_k)$ visiems $k = 0,1,\cdots,n$.

Žemiau patalpinta Ermito interpoliacinio polinomo reikšmių apskaičiavimo funkcija ***ermito***.

```
function fv=ermito(x,y,yi,t)
% ERMITO apskaičiuoja Ermito interpoliacinio polinomo reikšmes.
% Polinomo reikšmės skaičiuojamos pagal skirtuminių santykių formą.
% Įėjimo parametrai
% (x,y)      - interpoliavimo taškai,
%  yi        - pirmųjų išvestinių reikšmės interpoliavimo taškuose,
%  t         - argumento reikšmių masyvas.
% Išėjimo parametrai
% fv         - Ermito interpoliacinio polinomo reikšmės.

m=numel(t);  n=numel(x);
z=zeros(1,2*n); d=zeros(1,2*n);
[k,l]=size(t);
if k ==1
  t=t';
end
[k,l]=size(x);
if k ~=1
  x=x'; y=y';
end
% Pirmųjų skirtuminių santykių apskaičiavimas
dt=y; h=x(2:end)-x(1:end-1);
tt=(dt(2:end)-dt(1:end-1))./h; dt(2:end)=tt;
% Pirmųjų skirtuminių santykių masyvo ir z masyvo formavimas
z(1:2:end-1)=x; z(2:2:end)=x;
d(1:2:end-1)=dt; d(2:2:end)=yi;
% Aukštesnės eilės skirtuminių santykių apskaičiavimas
for k=2:2*n-1
    h=z(k+1:end)-z(1:end-k);
    tt=(d(k+1:end)-d(k:end-1))./h;
    d(k+1:end)=tt;
end
% Ermito interpoliacinio polinomo reikšmių apskaičiavimas
xx=repmat(z,m,1); dd=repmat(d,m,1); tt=repmat(t,1,2*n-1);
p=tt-xx(:,1:end-1); r=ones(m,1); s=[r cumprod(p,2)];
fv=sum((dd.*s)');
```

Pavyzdys. Turime funkcijos $y = f(x)$ bei jos išvestinės reikšmės mazguose: *x*=1.3, 1.6, 1.9; t.y., $f(1.3) = 0.6200860$, $f(1.6) = 0.4554022$, $f(1.9) = 0.2818186$ ir $f'(1.3) = -0.5220232$, $f'(1.6) = -0.5698959$, $f'(1.9) = -0.5811571$. Pagal Ermito interpoliacinį polinomą apskaičiuokime šios funkcijos reikšmes taškuose *t*=1.5 ir *t*=1.7.

Šias reikšmes apskaičiuosime pasinaudodami funkcija ***hermit***, o taip pat užrašysime mazgų z seką ir skirtuminių santykių, t.y. masyvo d , reikšmes, kaip jos apskaičiuojamos skaičiavimo eigoje.

```
z = 1.3000,  1.3000,  1.6000,  1.6000,  1.9000,  1.9000.
d = 0.6201, -0.5220, -0.5489, -0.5699, -0.5786, -0.5812.
d = 0.6201, -0.5220, -0.0897, -0.0698, -0.0291, -0.0085.
d = 0.6201, -0.5220, -0.0897,  0.0664,  0.0680,  0.0686.
d = 0.6201, -0.5220, -0.0897,  0.0664,  0.0027,  0.0010.
d = 0.6201, -0.5220, -0.0897,  0.0664,  0.0027, -0.0028.
f(1.5)= 0.5118; f(1.7)= 0.3980.
```

5.1.1.7. Interpoliacinių polinomų liekamojo nario apskaičiavimas

Interpoliacinio polinomo ir funkcijos reikšmės sutampa tik interpoliavimo mazguose $x_i, i = 0,1,\cdots,n$. ***Kyla klausimas, kiek skiriasi interpoliacinio polinomo ir funkcijos reikšmės taškuose, kurie nėra interpoliavimo mazgai?***

Aptarsime interpoliacinio polinomo paklaidą.

Tarkime, kad aproksimuojamoji funkcija $y = f(x)$ yra tolydžioji ir turi tolydžiąsias išvestines iki $(n+1)$-osios eilės imtinai. Žinoma, kad $f(x) = F(x) + R(x)$; čia $F(x)$ — n-tosios eilės interpoliacinis polinomas, o

$$R(x) = \frac{f^{(n+1)}(c)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n), (c \in [x_0; x_n]) \text{ —}$$

interpoliacinio polinomo liekamasis narys.

Pastaba. Literatūroje polinomas $(x-x_0)(x-x_1)\cdots(x-x_n)$ dažnai žymimas simboliu $\omega_n(x)$, t.y. $\omega_n(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$.

Jei žinome, kad $\left|f^{(n+1)}(x)\right| \le M$, kai $x \in [x_0; x_n]$, tai $|R(x)| \le \frac{M}{(n+1)!}|\omega_n(x)|$.

Paprastai tokia M reikšmė nėra žinoma ir todėl šis būdas nėra labai patogus.

Praktiškai patogiau naudotis formule

$$R(x) \approx |P_{n+1}(x) - P_n(x)| \qquad (5.7)$$

čia $P_n(x), P_{n+1}(x)$ - n-os ir (n+1)-os eilės interpoliaciniai polinomai, kuriuos atitinkamai apibrėžia mazgai $x_0, x_1, \cdots, x_n$ ir $x_0, x_1, \cdots, x_n, x_{n+1}$.

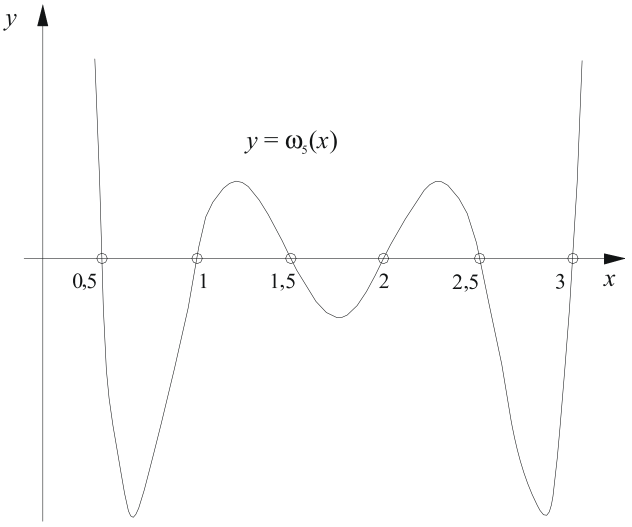
Liekamojo nario reikšmę galima sumažinti tinkamai parinkus interpoliavimo taškus. Jei lentelės žingsnis pastovus, tai $\omega_n(x)$ grafikas yra analogiškas 5.3 paveiksle pavaizduotam grafikui, kai n — nelyginis, arba 5.4 paveiksle pavaizduotam grafikui, kai n — lyginis.

Aišku, kad interpoliavimo paklaida bus mažesnė, jei interpoliavimo taškus parinksime taip, kad x būtų artimas intervalo $[x_0; x_n]$ vidurio taškui.

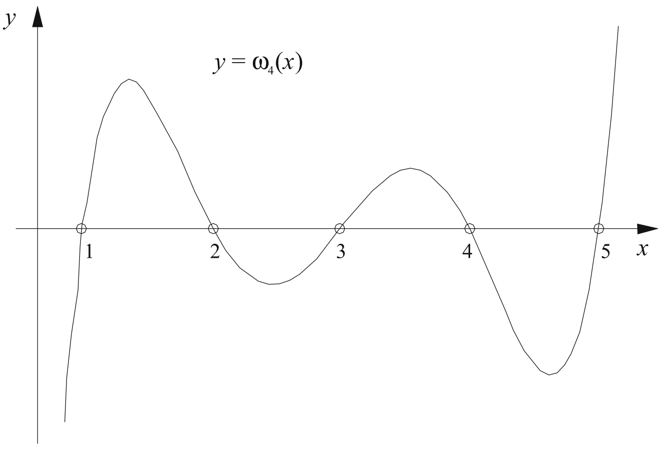
Dažnai įmanoma patiems sudaryti funkcijos $y = f(x)$ reikšmių lentelę. Kyla klausimas: *kaip parinkti taškus x_i $(i = \overline{0, n})$, kad liekamojo nario reikšmė būtų kuo mažesnė?* Atsakymas į šį klausimą yra toks.

Tarkime, kad funkciją $y = f(x)$ intervale [*a*;*b*] *norime aproksimuoti n-os eilės interpoliaciniu polinomu. Tada interpoliavimo mazgus reikia sutapatinti su n-os eilės*

Čiobyšovo polinomo šaknimis, transformuotomis į intervalą $[a;b]$, *t.y. interpoliavimo mazgai turi būti apskaičiuoti pagal formulę* $x_i = \frac{b+a}{2} + \frac{b-a}{2}\cos\left(\frac{\pi(2i+1)}{2n}\right), i = 0,1,\ldots,n-1$.



5.3 pav. $\omega_5(x)$ *grafikas, kai* $x_i = x_0 + ih$, $i = 0, \ldots, 5$, $h = 0{,}5$



5.4pav. $\omega_4(x)$ *grafikas, kai* $x_i = x_0 + ih$, $i = 0, \ldots, 4$, $h = 1$

5.1.1.8. Interpoliacinių polinomų konvergavimas

Skaičiuotojas, naudojantis interpoliacinius polinomus, yra linkęs manyti, kad, didinant interpoliacinio polinomo $F(x)$ eilę, polinomas ir aproksimuojamoji funkcija $f(x)$ vis geriau sutampa. Tačiau taip esti ne visada. Yra žinomos dvi teoremos, kurias sąlygiškai galime pavadinti „optimistine“ ir „pesimistine“.

Vejerštraso teorema („optimistinė“). Kad ir kokia būtų funkcija $y = f(x)$ $\left(x \in [a; b]\right)$, visada galima rasti tokius mazgus $x_i \in [a; b]$, su kuriais

$$\lim_{n \to \infty} \max_{x \in [a; b]} \left| f(x) - F(x) \right| = 0 .$$

Fabero teorema („pesimistinė“). Kad ir kokie būtų mazgai $x_i \in [a; b]$, visada galima rasti tokią funkciją $y = f(x)$, su kuria

$$\lim_{n \to \infty} \max_{x \in [a; b]} \left| f(x) - F(x) \right| \to \infty .$$

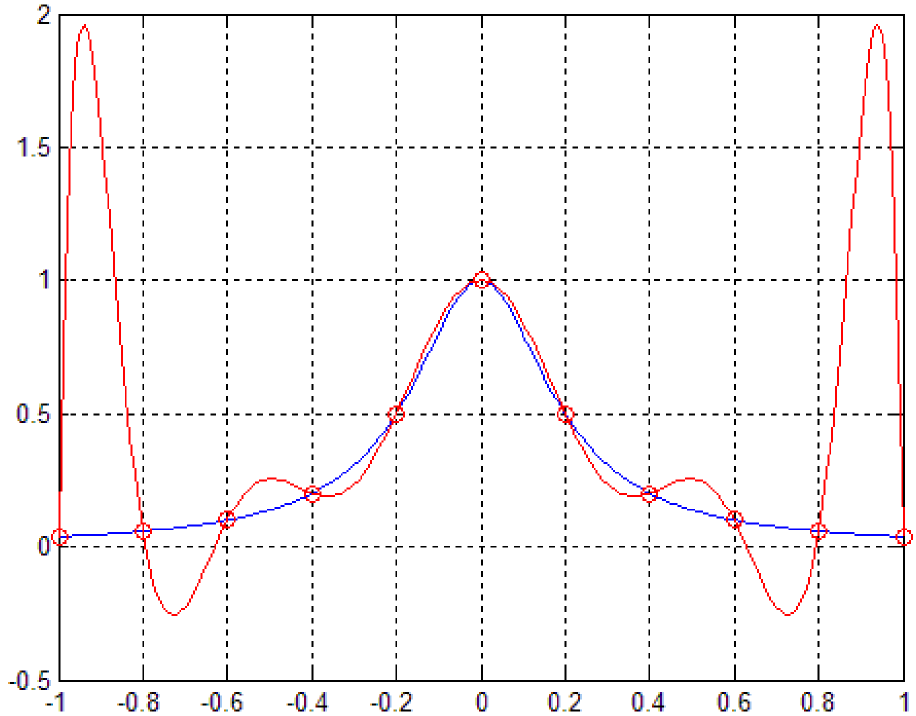
Pirmasis šiuos reiškinius 1901 metais pastebėjo K. Rungė (K. Runge). Jis nagrinėjo funkciją $y = \dfrac{1}{1 + 25x^2}$, kurios $x \in [-1; 1]$. Jei interpoliacinis polinomas bus konstruojamas imant vienodai nutolusius vienas nuo kito taškus, tai, didinant polinomo eilę, centrinėje intervalo dalyje interpoliacinio polinomo ir funkcijos reikšmės praktiškai sutaps, tačiau intervalo galuose jų skirtumo modulis artės prie begalybės.

Tačiau jei taškai bus sutapatinami su Čiobyšovo polinomo šaknimis, tai, didinant interpoliacinio polinomo eilę, funkcija ir polinomas praktiškai sutaps visame intervale. Analogišką rezultatą 1916 m. gavo S. Bernšteinas, nagrinėdamas funkciją $y = |x|$, kurios $x \in [-1; 1]$. Šie rezultatai ir buvo viena iš priežasčių, dėl kurių XX amžiaus šeštajame dešimtmetyje funkcijoms aproksimuoti vietoj polinomų pradėtos naudoti naujos funkcijos — splainai.

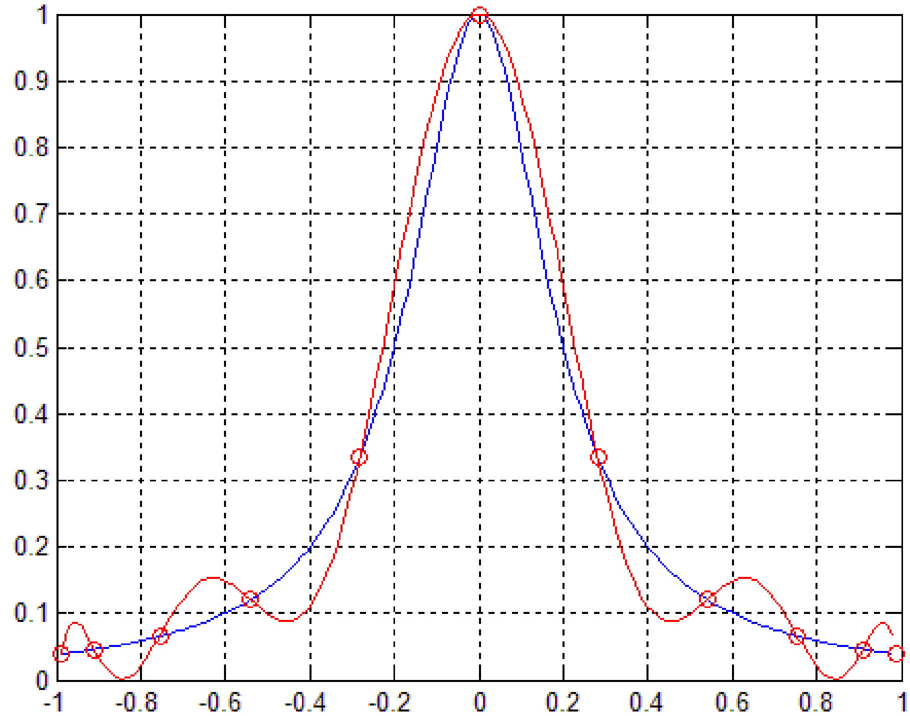
Vejerštraso ir Fabero teoremų teisingumą vaizdžiai iliustruoja 5.5, 5.6, 5.7 ir 5.8 paveikslėliai.

5.5 paveikslėlyje pavaizduotas Rungės funkcijos grafikas (mėlyna spalva) ir interpoliacinio polinomo, kurį nusako tolygiai išsidėstę mazgai, grafikas (raudona spalva). Šis paveikslėlis iliustruoja Fabero teoremos teisingumą.

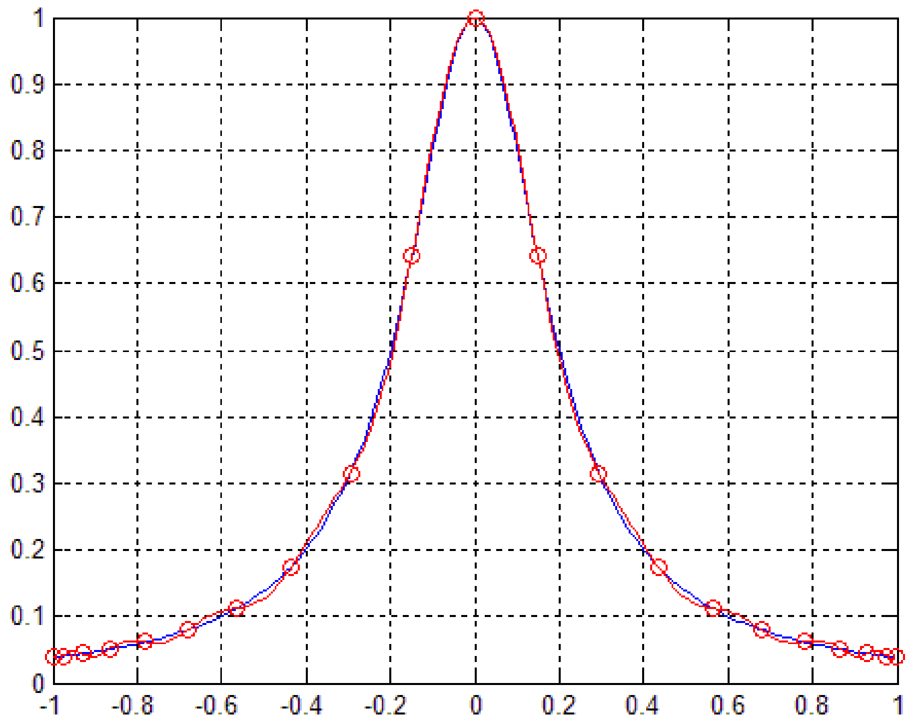
5.6, 5.7 ir 5.8 paveikslėliuose pavaizduotas Rungės funkcijos grafikas (mėlyna spalva) ir interpoliacinio polinomo, kurio mazgus apibrėžia Čiobyšovo polinomo šaknys, grafikai (raudona spalva). 5.6 paveikslėlyje mazgų skaičius yra 11, 5.7 paveikslėlyje mazgų skaičius yra 21, o 5.8 paveikslėlyje mazgų skaičius yra 41. 5.8 paveikslėlyje interpoliacinio polinomo grafikas vizualiai sutampa su tiksliu Rungės funkcijos grafiku, todėl paveikslėlyje matosi tik raudonos spalvos grafikas. Šie paveikslėliai iliustruoja Vejerštraso teoremos teisingumą.



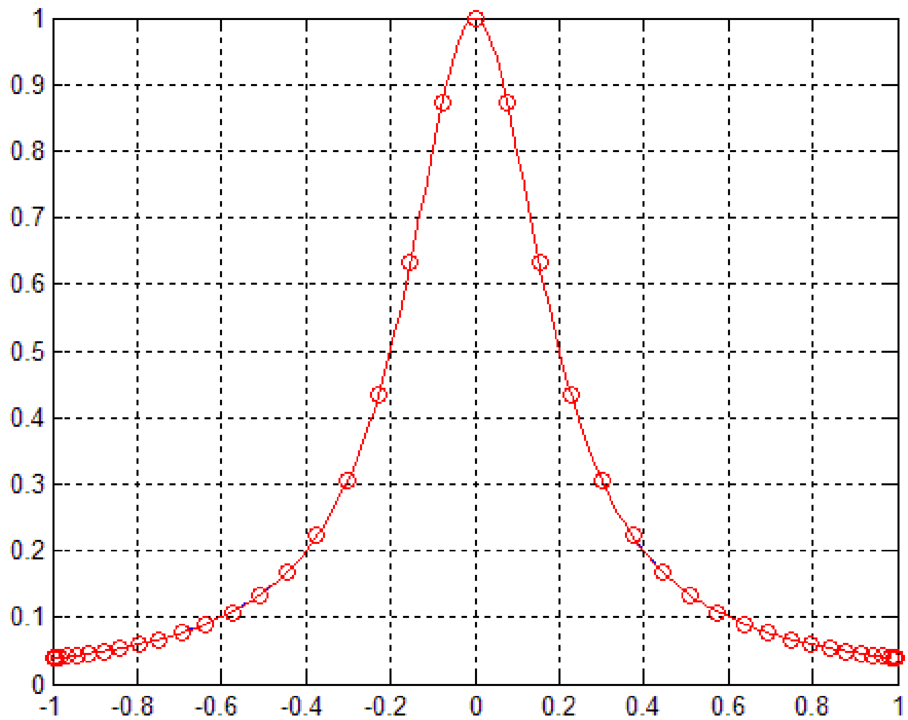
5.5 pav. Rungės funkcijos (mėlyna spalva) ir interpoliacinio polinomo (raudona spalva), kurį apibrėžia 11 tolygiai išsidėsčiusių mazgų, grafikai.



5.6 pav. Rungės funkcijos (mėlyna spalva) ir interpoliacinio polinomo (raudona spalva), kurio mazgai apibrėžiami Čiobyšovo polinomo šaknimis (11 mazgų), grafikai.



5.7 pav. Rungės funkcijos (mėlyna spalva) ir interpoliacinio polinomo (raudona spalva), kurio mazgai apibrėžiami Čiobyšovo polinomo šaknimis (21 mazgas), grafikai.



5.8 pav. Rungės funkcijos (mėlyna spalva) ir interpoliacinio polinomo (raudona spalva), kurio mazgai apibrėžiami Čiobyšovo polinomo šaknimis (41 mazgas), grafikai.

5.1.2. Interpoliacinių splainų konstravimas ir jų reikšmių apskaičiavimas

Splainas — tai tolydžioji iki p-tosios eilės išvestinės imtinai funkcija, sudaryta iš kurios nors funkcijos dalių. Istoriškai splainai pradėti konstruoti iš n-tosios eilės polinomo dalių, todėl čia, kalbėdami apie splainus, laikysime juos funkcijomis, sudarytomis iš polinomo dalių. Pastaruoju metu splainams konstruoti naudojamos kitos funkcijos, pavyzdžiui, trigonometrinės, taigi galime kalbėti apie trigonometrinius splainus.

Be trigonometrinių, dar naudojami splainai, angliškai vadinami gyvatėmis (*snakes*). Jie konstruojami dviem etapais:

1) pasirenkama tikslo funkcija — funkcionalas;

2) taikant variacinio skaičiavimo metodus, apskaičiuojama kreivė, minimizuojanti pasirinktą funkcionalą.

Paprasčiausias ir seniausiai naudojamas splainas yra laužtė. Tai pirmosios eilės splainas, kurio $p = 0$.

Splainas, sudarytas, pavyzdžiui, iš kubinio polinomo dalių, vadinamas kubiniu; jo $0 \le p \le 2$.

1946 m. I. Šionbergas (I. J. Schoenberg) splainų teorijos tėvas pirmasis pavartojo splaino sąvoką ir splainus pradėjo naudoti aproksimuodamas funkcijas. Ypač audringai splainų teorija buvo plėtojama po 1957 metų, kai Dž. K. Holidėjus (J. C. Holliday) atskleidė kubinių interpoliacinių splainų optimizacines savybes, o Dž. H. Albergas (J. H. Ahlberg), E. N. Nilsonas (E. N. Nilson) ir Dž. L. Volšas (J. L. Walsh) 1962 metais jas apibendrino, taikydami visiems nelyginio laipsnio splainams.

Splaino apibrėžimas. *Tarkime, kad intervale* $[a; b]$ *duotas tinklelis* Δ: $a = x_0 < x_1 < \ldots < x_N = b$. *Tada n-tosios eilės defekto k splainas yra funkcija* $y = g(x)$, *tenkinanti šias sąlygas:*

1) kiekviename intervale $[x_{i-1}; x_i]$ $\left(i = \overline{1, N}\right)$ $g(x)$ *yra n-tojo laipsnio polinomas;*

2) kiekviename vidiniame tinklelio taške x_i $\left(i = \overline{1, N-1}\right)$ *galioja lygybė*

$$g^{(l)}(x_i - 0) = g^{(l)}(x_i + 0), \quad l = \overline{0, n-k}.$$

Splaino pavadinimas kilęs iš anglų kalbos žodžio *spline*, reiškiančio standžią liniuotę. Inžinieriai (jau ir Leonardas da Vinčis), norėdami per atokiau išsidėsčiusius taškus nubrėžti tolydžią kreivę, imdavo standžią, pavyzdžiui, plieninę, liniuotę ir išraitydavo ją taip, kad eitų per tuos taškus. Liniuotė įgaudavo kubinio splaino formą. Iš tikrųjų ji stengėsi įgyti tokią formą, kad jos vidinė potencinė energija būtų mažiausia. Iš fizikos žinoma, kad liniuotės vidinė potencinė energija yra

$$\Phi(u) = \int_a^b (u''(x))^2 dx;$$

čia $u(x)$ — liniuotės formą apibūdinanti kreivė. Kaip 1957 m. įrodė Dž. K. Holidėjus, funkcionalas $\Phi(u)$ turi pačią mažiausią reikšmę, kai $u(x)$ yra kubinis splainas $g(x)$ su natūraliomis kraštinėmis sąlygomis, t. y. $g''(a) = g''(b) = 0$.

Praktikoje dažniausiai naudojami kubiniai defekto 1 splainai, kuriuos toliau vadinsime tiesiog kubiniais splainais.

Splaino užrašymo formos. Literatūroje nurodomos keturios splainų užrašymo formos, iš kurių tik dvi naudojamos praktiniame skaičiavime:

1) dalimis polinominė forma, kai kiekviename intervale $[x_{i-1}; x_i]$ nurodoma polinomo išraiška;

2) tiesinio B splainų darinio forma.

5.1.2.1. Interpoliacinio splaino egzistavimo sąlygos

Kubinio interoliacinio splaino konstravimo uždavinys. Duota funkcijos $y = f(x)$ reikšmių lentelė (x_i, y_i), $i = \overline{0, N}$. Reikia rasti kubinį splainą $y = g(x)$, tenkinantį Lagranžo interpoliavimo sąlygą

$$g(x_i) = y_i,\ i = \overline{0, N}.$$

Ar šis uždavinys apibrėžtas?

Kaip matyti iš apibrėžimo, kubinį splainą apibūdina $4N$ koeficientai: kiekviename intervale $[x_{i-1}; x_i]$ kubinį polinomą nusako keturi koeficientai, o intervalų skaičius yra N.

Kiek turime lygčių, siejančių tuos koeficientus?

Turime $N+1$ interpoliavimo sąlygą. Kiekviename vidiniame intervalo taške polinomai turi tenkinti tris sąlygas, vadinasi, turime dar $3(N-1)$ lygtis. Taigi $4N$ kubinio splaino koeficientai privalo tenkinti $4N-2$ lygtis. Kad interpoliacinis kubinis splainas būtų apibrėžtas vienareikšmiškai, reikia pasirinkti dvi papildomas sąlygas, vadinamas kraštinėmis sąlygomis. Dažniausiai pasirenkamos šios kraštinės sąlygos:

1) $g''(x_0) = g''(x_N) = 0$ (jos vadinamos natūraliomis kraštinėmis sąlygomis ir rodo, kad plieninės liniuotės, išraitytos per interpoliavimo taškus, galai paliekami laisvai kaboti);

2) $g''(x_0) = f''(x_0)$, $\quad g''(x_N) = f''(x_N)$;

3) $g'(x_0) = f'(x_0)$, $\quad g'(x_N) = f'(x_N)$;

4) $g'(x_0) = g'(x_N)$, $\quad g''(x_0) = g''(x_N)$ (jos vadinamos periodinėmis kraštinėmis sąlygomis);

5) $g'''(x_1 - 0) = g'''(x_1 + 0)$, $\quad g'''(x_{N-1} - 0) = g'''(x_{N-1} + 0)$ (šios sąlygos reiškia, kad intervaluose $[x_0; x_1]$ ir $[x_1; x_2]$, taip pat $[x_{N-2}; x_{N-1}]$ ir $[x_{N-1}; x_N]$ atitinkamai yra ta pati kubinio polinomo išraiška).

n-os eilės defekto k interpoliacinio splaino egzistavimo sąlygos. Kiek kraštinių sąlygų reikia pasirinkti apskaičiuojant n-tosios eilės defekto k interpoliacinį splainą? Aišku, kad splainą nusako $(n+1)\cdot N$ koeficientų, kurie turi tenkinti $(N+1)+(n+1-k)(N-1)$ lygčių. Vadinasi, kraštinių sąlygų turi būti

$$(n+1)N - (N+1) - (n+1-k)(N-1) = N(k-1) + n - k.$$

Praktiškai daugiausia naudojami defekto 1 splainai, t. y. $k = 1$. Tokių splainų kraštinių sąlygų skaičius yra $n - 1$. Pavyzdžiui, norint sudaryti 5-osios eilės defekto 1 interpoliacinį splainą, reikia pasirinkti keturias kraštines sąlygas.

5.1.2.2. *B splainai*

Kaip jau buvo minėta, viena iš splaino užrašymo formų yra tiesinis *B* splainų darinys. Kas yra tie *B* splainai? Tarkime, kad turime tinklelį Δ: $a = x_0 < x_1 < \ldots < x_N = b$. Išplėskime jį papildomais taškais: $x_{-n} < x_{-(n-1)} < \ldots < x_{-1} < a$, $b < x_{N+1} < x_{N+2} < \ldots < x_{N+n}$.

Galime tarti, pavyzdžiui, kad $x_{-i} = x_0 - i\left(x_1 - x_0\right)$, $x_{N+i} = x_N + i\left(x_N - x_{N-1}\right)$, $i = \overline{1, n}$. Tada n-tosios eilės defekto vienas *B* splainas yra splainas, kuris $(n+1)$ intervale nelygus nuliui, o visuose kituose intervaluose lygus nuliui, t. y.

$$\widetilde{B}_n^i(x) = \begin{cases} \neq 0, & \text{kai } x \in \left(x_i;\, x_{i+n+1}\right), \\ = 0, & \text{kai } x \notin \left(x_i;\, x_{i+n+1}\right). \end{cases}$$

Šį splainą nusako $(n+1)^2$ koeficientų, kurie visuose taškuose x_i turi tenkinti glodumo sąlygas, t. y. kiekviename taške *B* splaino ir jo išvestinių iki $(n-1)$-osios eilės imtinai reikšmės iš kairės ir dešinės turi sutapti. Vadinasi, sąlygų skaičius lygus $(n+2)\cdot n = n^2 + 2n$.

Kad *B* splainas būtų apibrėžtas vienareikšmiškai, keliama viena papildoma sąlyga:

$$\int_{-\infty}^{+\infty} \widetilde{B}_n^i(x)\, dx = 1.$$

Praktiškai skaičiuojant patogiau naudoti normalizuotus *B* splainus, apibrėžiamus taip:

$$B_n^i(x) = \frac{x_{i+n+1} - x_i}{n+1}\, \widetilde{B}_n^i(x),$$

taigi $\widetilde{B}_n^i(x)$ dauginami iš vidutinio žingsnio, kuriame $\widetilde{B}_n^i(x) \neq 0$, ilgio. Aišku, kad

$$B_0^i(x) = \begin{cases} 1, & \text{kai } x \in \left[x_i;\, x_{i+1}\right), \\ 0, & \text{kai } x \notin \left[x_i;\, x_{i+1}\right). \end{cases}$$

Toliau naudosime tik normalizuotus *B* splainus, kuriuos vadinsime tiesiog *B* splainais.

Normalizuoti *B* splainai tenkina rekurenčiąją lygtį

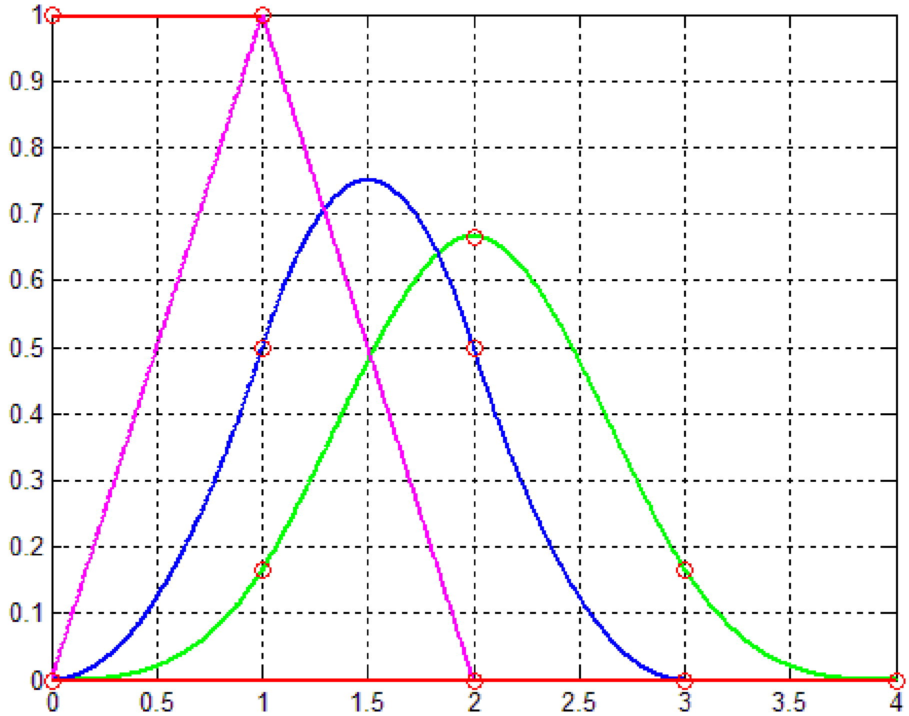
$$B_n^i(x) = \frac{x - x_i}{x_{i+n} - x_i}\, B_{n-1}^i(x) + \frac{x_{i+n+1} - x}{x_{i+n+1} - x_{i+1}}\, B_{n-1}^{i+1}(x), \quad n \geq 1. \quad (5.8)$$

Pavyzdys. Remdamiesi (5.8) formule, užrašykime $B_1^i(x)$ ir $B_2^i(x)$ analizines išraiškas.

Žemiau pateiktos norimos išraiškos, o 5.9 paveiksluose pavaizduoti *B* splainų: $B_0^0(x), B_1^0(x), B_2^0(x), B_3^0(x)$ grafikai, kai tinklelio ilgis lygus vienetui.

$$B_1^i(x) = \begin{cases} \dfrac{x - x_i}{x_{i+1} - x_i}, & \text{kai } x \in \left[x_i;\, x_{i+1}\right], \\ \dfrac{x_{i+2} - x}{x_{i+2} - x_{i+1}}, & \text{kai } x \in \left[x_{i+1};\, x_{i+2}\right], \\ 0, & \text{kai } x \notin \left[x_i;\, x_{i+2}\right]; \end{cases}$$

$$B_2^i(x)=\begin{cases}\dfrac{(x-x_i)^2}{(x_{i+1}-x_i)(x_{i+2}-x_i)}, & \text{kai } x\in[x_i;\,x_{i+1}],\\[2ex] \dfrac{(x-x_i)(x_{i+2}-x)}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})}+ & \\ +\dfrac{(x_{i+3}-x)(x-x_{i+1})}{(x_{i+3}-x_{i+1})(x_{i+2}-x_{i+1})}, & \text{kai } x\in[x_{i+1};\,x_{i+2}],\\[2ex] \dfrac{(x_{i+3}-x)^2}{(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})}, & \text{kai } x\in[x_{i+2};\,x_{i+3}],\\[2ex] 0, & \text{kai } x\notin[x_i;\,x_{i+3}];\end{cases}$$



5.9 pav. $B_0^0(x), B_1^0(x), B_2^0(x), B_3^0(x)$ *splainų grafikai*

Pagrindinė B splainų savybė. *B splainai :* $B_n^i(x)$ $\left(i=\overline{-n,\,N-1}\right)$ *yra tiesiškai nepriklausomi ir sudaro splainų* $S_{n,1}(\Delta)$ *klasės bazę; čia* $S_{n,1}(\Delta)$ *— n-tosios eilės defekto 1 splainai, kuriuos apibūdina tinklelis* Δ.

Vadinasi, bet kuris n-tosios eilės defekto 1 splainas $g(x)$ gali būti užrašytas kaip tiesinis B splainų darinys:

$$g(x)=\sum_{i=-n}^{N-1} b_i B_n^i(x). \qquad (5.9)$$

Toks splaino užrašas yra labai patogus, nes, apskaičiuojant splaino reikšmes, tik $n+1$ (5.9) sumos dėmuo nelygus nuliui.

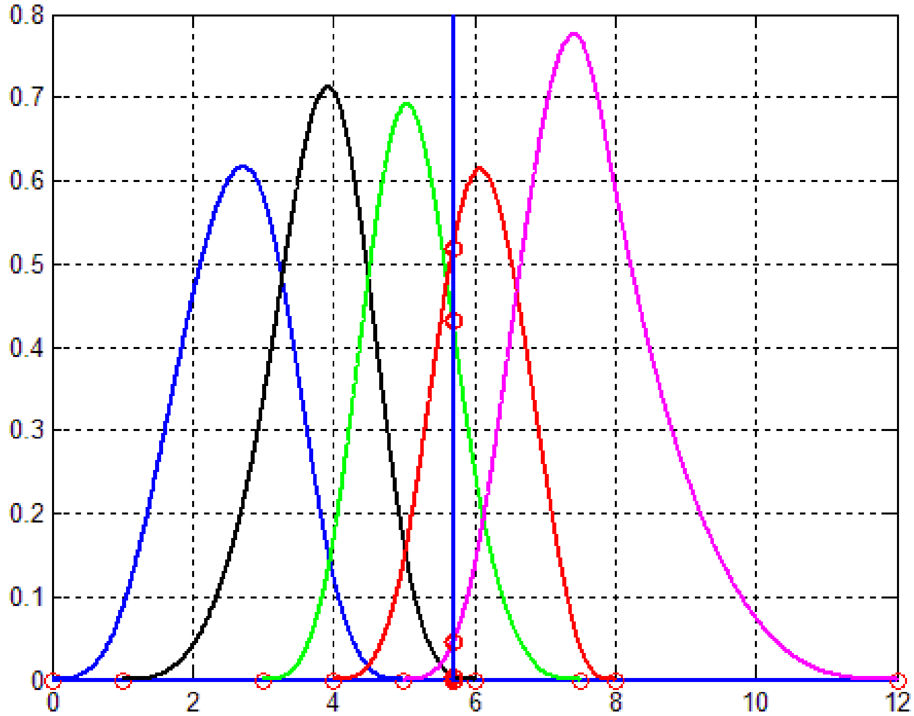
Tarkime, kad norime apskaičiuoti $g(x)$ reikšmę, kai $x \in [x_i; x_{i+1}]$. Aišku, kad

$$g(x) = \sum_{k=i-n}^{i} b_k B_n^k(x). \qquad (5.10)$$

B splainai turi vieneto išskaidymo savybę:

tarkime $x \in [x_i; x_{i+1}]$*; tada* $\sum_{k=i-n}^{i} B_n^k(x) = 1$.

5.10 paveiksle pavaizduotas tinklelis $x_0 = 0, x_1 = 1, x_2 = 3, x_3 = 4, x_4 = 5, x_5 = 6, x_6 = 7.5, x_7 = 8, x_8 = 12$ ir dalis kubinių B splainų bazės, kuri duotam tinkleliui apibrėžia bet kurį kubinį splainą. Be to, paveiksle pavaizduota ir B splainų vieneto išskaidymo savybė: vertikalė *x*=5.7 kerta splainų B_3^1, B_3^3, B_3^4 ir B_3^5 grafikus ir susikirtimo taškų ordinačių suma yra lygi vienetui.



5.10 pav. B splainų bazės poaibis ir vieneto išskaidymo savybė

Norint rasti $g(x)$ išvestinės reikšmę, reikia mokėti apskaičiuoti $B_n^k(x)$ išvestinę. Literatūroje nurodyta, kad B splaino $B_n^k(x)$ išvestinė apskaičiuojama pagal formulę

$$\left(B_n^k(x)\right)' = \frac{n}{x_{k+n} - x_k} B_{n-1}^k(x) - \frac{n}{x_{k+n+1} - x_{k+1}} B_{n-1}^{k+1}(x). \qquad (5.11)$$

Tada splaino $g(x)$ išvestinės reikšmes, kai $x \in [x_i; x_{i+1}]$, bus galima rasti pagal formulę

$$g'(x) = n \sum_{k=i-n+1}^{i} b_k^{(1)} B_{n-1}^k(x); \text{ čia } b_k^{(1)} = \frac{b_k - b_{k-1}}{x_{k+n} - x_k} \qquad (5.12)$$

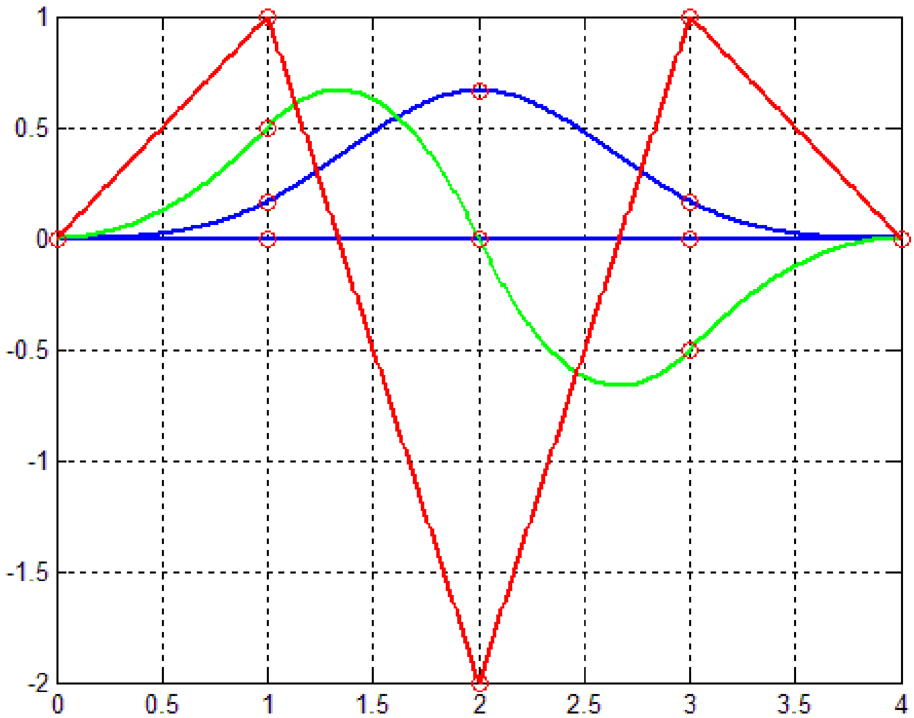
Aišku, kad r-tosios eilės splaino išvestinė, kai $x \in [x_i; x_{i+1}]$, bus apskaičiuojama pagal formulę

$$g^{(r)}(x) = n(n-1)\ldots(n-r+1) \sum_{k=i-n+r}^{i} b_k^{(r)} B_{n-r}^{k}(x); \qquad (5.13)$$

čia $b_k^{(0)} = b_k, \quad k = \overline{i-n, i}$,

$$b_k^{(l)} = \frac{b_k^{(l-1)} - b_{k-1}^{(l-1)}}{x_{k+n+1-l} - x_k}, \quad l = \overline{1, r}, \quad k = \overline{i-n+l, i}. \qquad (5.14)$$

5.11 paveiksle pavaizduoti splaino $B_3^0(x)$ ir jo pirmosios ir antrosios išvestinių, apskaičiuotų pagal (5.12)-(5.14) formules, grafikai.



5.11 pav. Splaino $B_3^0(x)$ ir jo pirmosios bei antrosios išvestinių grafikai

Žemiau pateiktos ***bsplin*** ir ***bg*** funkcijos, iš kurių pirmoji skirta brėžti B splainų grafikus (žr.5.9 ir 5.10 paveikslus), o antroji – apskaičiuoti splaino ir jo išvestinių reikšmes, kai splainas nusakomas tiesiniu B splainų dariniu (žr. 5.11 paveikslą, kuris gaunamas kreipiantis į ***bg*** funkciją, imant mazgus x=[0,1,2,3,4], svorinius koeficientus b=[0 0 0 1 0 0 0] ir r=0, 1,2.).

```
function [n,xx,yy,y]=bsplin(x)
% BSPLIN apskaičiuoja n-os eilės B splaino, kurį
% apibrėžia tinklelis x, grafiką.
Įėjimo parametrai
% x - tinklelio mazgai.
Išėjimo parametrai
% n - B splaino eilė,
% xx - Argumento reikšmės, išdėstytos žingsniu h=0.005,
```

```
% yy - B splaino reikšmės taškuose xx,
% y  - B splaino reikšmės tinklelio mazguose.

nn=numel(x);
n=nn-2; % n - B splaino eile
y=[]; xx=[]; yy=[];
for i=1:n+1  % nagrinėjame i-ąjį intervalą
    u=x(i):0.005:x(i+1);
    xx=[xx u];
    t=u';
    m=numel(t);
    a=zeros(m,i+1); % apibrėžiame nulinės eiles B splainų reikšmes
    a(:,i)=1;
    for el=1:n
        i1=max(1,i-el);
        i2=min(i,n-el+1);
        for k=i1:i2
            if (x(k+el)-x(k)) ~=0
                a(:,k)=(t-x(k)).*a(:,k)/(x(k+el)-x(k));
            end
            if (x(k+el+1)-x(k+1)) ~=0
                a(:,k)=a(:,k)+(x(k+el+1)-t).*a(:,k+1)/(x(k+el+1)-
                       x(k+1));
            end
        end
    end
    yy=[yy (a(:,1))']; y=[y a(1,1) a(m,1)];
end
y(3:2:end-1)=[];

function [grt,a]=bg(n,r,x,b,t)
% bg funkcija apskaičiuoja n-os eilės splaino, nusakyto
% tiesiniu B splainų dariniu,r-os eiles išvestinės reikšmes,kai x=t.
Įėjimo parametrai
% n - splaino eile,
% r - išvestinės eilė ( 0<=r<=n),
% x - tinklelio mazgai,
% b - tiesinio darinio koeficientai,
% t - argumento reikšmių, priklausančių
% intervalui [x(1),x(end)], masyvas,
% a - a(k) - (n-r)-os eilės B splainų, nusakytų
%     taškų x(i-n+r+k), k=1,2, ...,n-r+1, atžvilgiu, reikšmės,
%grt - splaino r-os eiles išvestinės reikšmės, kai x=t.

nn=numel(x);  % tinklelio mazgų skaičius
m=numel(t);   % t reikšmių skaičius
if (t(1) < x(1))|(t(m) > x(nn))
    disp('t už intervalo ribų');
    grt=nan; a=[];
    return
end
if n==0
    grt=ones(1,m); a=[]; return
end
if r < 0
    grt=nan; a=[];
    disp('neigiama išvestinės reikšmė');
```

```
    return
end
if r > n
    grt=zeros(1,m); a=[];
    return
end
tm=zeros(m,nn+2*n); xm=zeros(m,nn+2*n);
a=zeros(m,n-r+2);   xb=zeros(m,2*n-2*r+2);
xd=zeros(m,2*n);
% Tinklelio papildymas
h0=x(2)-x(1); hn=x(nn)-x(nn-1);
xt0=linspace(x(1)-n*h0,x(1)-h0,n);
xtn=linspace(x(nn)+hn,x(nn)+n*hn,n);
xx=[xt0 x xtn];
% Kuriam intervalui priklauso t?
tm=repmat(t',1,nn+2*n); xm=repmat(xx,m,1);
ind=nn+2*n-sum((xm>=tm)');
if ind(1)==n
    ind(1)=n+1;
end;
%Koeficientu b(k) perskaičiavimas
db=zeros(m,n+1);
for j=1:m
    ij=ind(j);  dbt=b(ij-n:ij);
    db(j,:)=dbt;
end
for j=1:m
    ij=ind(j);  xdt=xx(ij-n+1:ij+n);
    xd(j,:)=xdt;
end
for el=1:r
    for k=n:-1:el
        v=xd(:,n+k-el+1)-xd(:,k);
        db(:,k+1)=(db(:,k+1)-db(:,k))./v;
    end
end
% B splainų reikšmių apskaičiavimas
at=zeros(1,n-r+2);
at(n-r+1)=1;
a=repmat(at,m,1);  % nulinės eilės B splainų reikšmės
for j=1:m
    ij=ind(j);
    xbt=xx(ij-n+r:ij+n-r+1);
    xb(j,:)=xbt;
end
for el=1:n-r
    for k=n-r+1-el:n-r+1
        a(:,k)=(t'-xb(:,k)).*a(:,k)./(xb(:,k+el)-xb(:,k))+...
            (xb(:,k+el+1)-t').*a(:,k+1)./(xb(:,k+el+1)-xb(:,k+1));
    end
end
%grt apskaičiavimas
koef=n:-1:n-r+1; grt=prod(koef)*sum((a(:,1:n-r+1).*db(:,r+1:n+1))');
```

5.1.2.3. *Interpoliacinių splainų konstravimas*

Aptarkime interpoliacinio splaino , kai splainas nusakytas tiesiniu B splainų dariniu, apskaičiavimo uždavinį. Pradžioje išnagrinėkime kubinio, o po to, n-os eilės interpoliacinio splaino konstravimą.

Kubinio interpoliacinio splaino konstravimas. Uždavinio formuluotė pateikta 5.1.2.1 paragrafe: duota funkcijos $y = f(x)$ reikšmių lentelė (x_i, y_i), $i = \overline{0, N}$; reikia rasti kubinį interpoliacinį splainą, tenkinantį vieną iš 5.1.2.1 paragrafe išvardytų kraštinių sąlygų. Šiuo atveju splainas užrašomas formule

$$g(x) = \sum_{i=-3}^{N-1} b_i B_3^i(x). \tag{5.15}$$

Kadangi čia kalbėsime apie trečios eilės B splainus, todėl, prastindami jų žymėjimo simboliką, vietoj $B_3^k(x)$ rašysime $B_k(x)$.

Kadangi $B_k(x)$ nelygus nuliui intervale $\left[x_k; x_{k+4}\right]$, tai interpoliavimo sąlygos, atsižvelgiant į kraštines sąlygas, užrašomos lygčių sistema

$$\begin{cases} b_{i-3}B_{i-3}(x_i) + b_{i-2}B_{i-2}(x_i) + b_{i-1}B_{i-1}(x_i) = y_i, i = \overline{0,N}, \\ b_{-3}B'_{-3}(x_0) + b_{-2}B'_{-2}(x_0) + b_{-1}B'_{-1}(x_0) = f'(x_0), \\ b_{N-3}B'_{N-3}(x_N) + b_{N-2}B'_{N-2}(x_N) + b_{N-1}B'_{N-1}(x_N) = f'(x_N). \end{cases} \tag{5.16}$$

Čia pasirinktos 3-iojo tipo kraštinės sąlygos (žr. 5.1.2.1 paragrafą). Vadinasi, (5.15) formulės tiesinio darinio koeficientų aibė yra (5.16) sistemos sprendinys.

(5.16) lygčių sistemos matrica apskaičiuojama funkcijos ***bg*** pagalba.

Tarkime, kad interpoliavimo mazgai yra: x=[-2; -0.5; 0; 1; 3; 4]. Tada komandos `[grt,a]=bg(n,0,x,b,x)` išėjimo parametro a nenuliniai nariai apibrėš sistemos matricos pradedant 1-ja ir baigiant N+1 eilute nenulinius koeficientus.

Norėdami apskaičiuoti $B_k^{(r)}(x_0), k = -3,-2,-1$ ir $B_k^{(r)}(x_N), k = N-3, N-2, N-1$ reikšmes, kai $r = 1$ arba $r = 2$, t. y. norėdami rasti (5.16) sistemos dviejų paskutinių lygčių koeficientus, kreipsimės į funkciją ***bg***, su tinkamai parinktomis koeficientų b reikšmėmis. Koeficientų b reikšmės parenkamos taip, kad splainas $g(x)$, apibrėžtas (5.15) formule, sutaptų su norimu B splainu. Pavyzdžiui, jei paimsime $b_{-3} = 1$ ir $b_l = 0, l = \overline{-2, N-1}$, tai $g(x)$ sutaps su $B_{-3}(x)$. Vadinasi, koeficientų b rinkiniai apibrėžiami pagal formulę: $b_k = 1, b_l = 0, l \neq k$ visiems $k = -3,-2,-1, N-3, N-2, N-1$. Tada apskaičiuotos ***grt*** reikšmės ir bus ieškomi dviejų paskutiniųjų lygčių koeficientai.

Anksčiau nurodytiems interpoliavimo mazgams (5.16) sistemos matrica bus lygi:

```
0.1667    0.6190    0.2143         0         0         0         0         0
     0    0.0357    0.5893    0.3750         0         0         0         0
     0         0    0.2222    0.7302    0.0476         0         0         0
     0         0         0    0.3810    0.5357    0.0833         0         0
     0         0         0         0    0.0833    0.5833    0.3333         0
     0         0         0         0         0    0.1250    0.7083    0.1667
0.4444   -1.0159    0.5714         0         0         0         0         0
     0         0         0         0         0    0.7500   -1.7500    1.0000
```

Išsprendę (5.16) lygčių sistemą, turėsime koeficientų b rinkinį, kuris ir apibrėš kubinį interpoliacinį splainą.

Pavyzdys. Apskaičiuokime kubinį interpoliacinį splainą $y = g(x)$, kuris interpoliuotų polinomą $y(x) = x^3 - 2x^2 + 1$ mazguose *x*=[-2; -0.5; 0; 1; 3; 4] ir kuris tenkintų kraštines sąlygas: $g'(x_0) = -16, g'(x_N) = 20$.

Kadangi (5.16) lygčių sistemos matrica priklauso tik nuo interpoliavimo mazgų ir yra pavaizduota anksčiau, tai belieka apskaičiuoti laisvųjų narių stulpelį: $\left[y(x_0), y(x_1), \cdots, y(x_N), y'(x_0), y'(x_N)\right]^t$ =[-15.0; 0.3750; 1.0; 0; 10.0; 33.0; -16.0; 20.0]^t.

Išsprendę sistemą, turėsime ieškomąjį splainą nusakantį koeficientų *b* rinkinį:

b= [-57.0; -9.0; 0.3333; 1.3333; -1.0; 0.3333; 29.6667; 71.6667].

Kadangi interpoliuojame kubinį polinomą ir kraštines sąlygas parinkome taip, kad $g'(x_0) = y'(x_0), g'(x_N) = y'(x_N)$, tai intervale [-2; 4] apskaičiuotas splainas sutampa su polinomu.

n-os eilės interpoliacinių splainų konstravimas. Splaino užrašas tiesiniu *B* splainų dariniu yra universalus. Todėl, naudojant šią formą, galima sudaryti funkcijas, įgalinančias apskaičiuoti *n*-os eilės interpoliacinį splainą, tenkinantį įvairias kraštines sąlygas. Tokių splainų konstravimas iš esmės niekuo nesiskiria nuo aptarto kubinio interpoliacinio splaino konstravimo: sudaroma ir išsprendžiama (5.16) tipo lygčių sistema, kurios sprendinys (koeficientų *b* masyvas) ir apibrėžia ieškomąjį interpoliacinį splainą.

Lygčių sistemos sudarymo metodika pilnai sutampa su (5.16) sistemos konstravimu kubiniams splainams. Tokių sistemų matricų struktūra pavaizduota 5.13 ir 5.14 paveikslėliuose. 5.13 pav. vaizduoja sistemos matricą, kai kraštinės sąlygos yra neperiodinės, 5.14 pav. vaizduoja sistemos matricą, kai kraštinės sąlygos – periodinės.

```
 ******
  ******
      . . .
        ******
          ******
******
 . . .
******
          ******
           . . .
           ******
```

5.13 pav. Sistemos matricos struktūra, kai kraštinės sąlygos – neperiodinės

```
******
  ******
      . . .
         ******
            ******
******     ******
          . . .
******     ******
```

5.14 pav. Sistemos matricos struktūra, kai kraštinės sąlygos –periodinės

Žemiau pateitos funkcijos ***bnspln*** ir ***pbnspln***, iš kurių pirmoji skirta apskaičiuoti *n*-os eilės interpoliacinius splainus su neperiodinėmis kraštinėmis sąlygomis, o antroji – su periodinėmis kraštinėmis sąlygomis.

```
function b=bnspln(n,e,prs,x,y)
% BSPLN apskaičiuoja n-os eiles  interpoliacinį splainą,
% kai splainas nusakytas tiesiniu B splainų dariniu.
```

```
% Įėjimo parametrai
% n - splaino eile,
% e – kraštines sąlygas nusakančios išvestinių eilės,
% prs – kraštinių sąlygų reikšmės:
% (pirmiausia nurodomos visos kraštinės sąlygos taškui x(0),
% o po to taškui x(N)),
% x - interpoliavimo mazgų abscisės,
% y - interpoliavimo mazgų ordinatės.
% Išėjimo parametrai
% b - tiesinio darinio koeficientai,

N=numel(x)-1; % N - intervalų skaičius
b=zeros(1,N+n); pa=zeros(N+1,n+2); a=zeros(N+n,N+n);
% Matricos a formavimas
[grt,pa]=bg(n,0,x,b,x);
a(1:2,1:n+1)=pa(1:2,1:n+1);
for i=3:N+1
    a(i,i:i+n-1)=pa(i,2:n+1);
end
% Kraštinių sąlygų įvertinimas taške x(1)
if mod(n,2)==0
    k1=floor((n-2)/2)+1;
else
    k1=floor((n-1)/2);
end
for k=1:k1
    b(1)=1; r=e(k);
    if r > n
        disp('per didelė išvestinės eilė'); return
    end
    if r < 0
        disp('išvestinės eilė - neigiama'); return
    end
    for i=1:n
        [grtr,pa]=bg(n,r,x,b,x(1));
        a(N+k+1,i)=grtr;
        b(i)=0; b(i+1)=1;
    end
    b(n+1)=0;
end
% Kraštinių sąlygų įvertinimas taške x(N+1)
 for k=k1+1:n-1
    b(N+1)=1; r=e(k);
    if r > n
        disp('per didelė išvestinės eilė');
        return
    end
    if r < 0
        disp('išvestinės eilė - neigiama');
        return
    end
    for i=1:n
        [grtr,pa]=bg(n,r,x,b,x(N+1));
        a(N+k+1,N+i)=grtr;
        b(N+i)=0; b(N+i+1)=1;
    end
    b(N+n+1)=0;
```

```
end
% Laisvieji nariai
d(1:N+1)=y;
d(N+2:N+n)=prs;
% Tiesinio darinio koeficientų skaičiavimas
b=a\d';

function b=pbnspln(n,x,y)
% PBSPLN apskaičiuoja n-os eilės periodinį interpoliacinį splainą.
% Įėjimo parametrai
% n- splaino eilė,
% x - interpoliavimo mazgų abscises,
% y - interpoliavimo mazgų ordinates.
% Išėjimo parametrai
% b - tiesinio darinio koeficientai,

N=numel(x)-1; % N - intervalų skaičius
b=zeros(1,N+n); pa=zeros(N+1,n+2); a=zeros(N+n,N+n);
% Matricos a formavimas
[grt,pa]=bg(n,0,x,b,x);
a(1:2,1:n+1)=pa(1:2,1:n+1);
for i=3:N+1
    a(i,i:i+n-1)=pa(i,2:n+1);
end
% Kraštinių sąlygų įvertinimas taške x(1)
for r=1:n-1
    b(1)=1;
    for i=1:n
        [grtr,pa]=bg(n,r,x,b,x(1));
        a(N+r+1,i)=a(N+r+1,i)+grtr;
        b(i)=0; b(i+1)=1;
    end
    b(n+1)=0;
end
% Kraštinių sąlygų įvertinimas taške x(N+1)
for r=1:n-1
    b(N+1)=1;
    for i=1:n
        [grtr,pa]=bg(n,r,x,b,x(N+1));
        a(N+r+1,N+i)=a(N+r+1,N+i)-grtr;
        b(N+i)=0; b(N+i+1)=1;
    end
    b(N+n+1)=0;
end
% Sistemos laisvųjų narių apskaičiavimas
d(1:N+1)=y; d(N+2:N+n)=0;
% Tiesinio darinio koeficientų skaičiavimas
b=a\d';
```

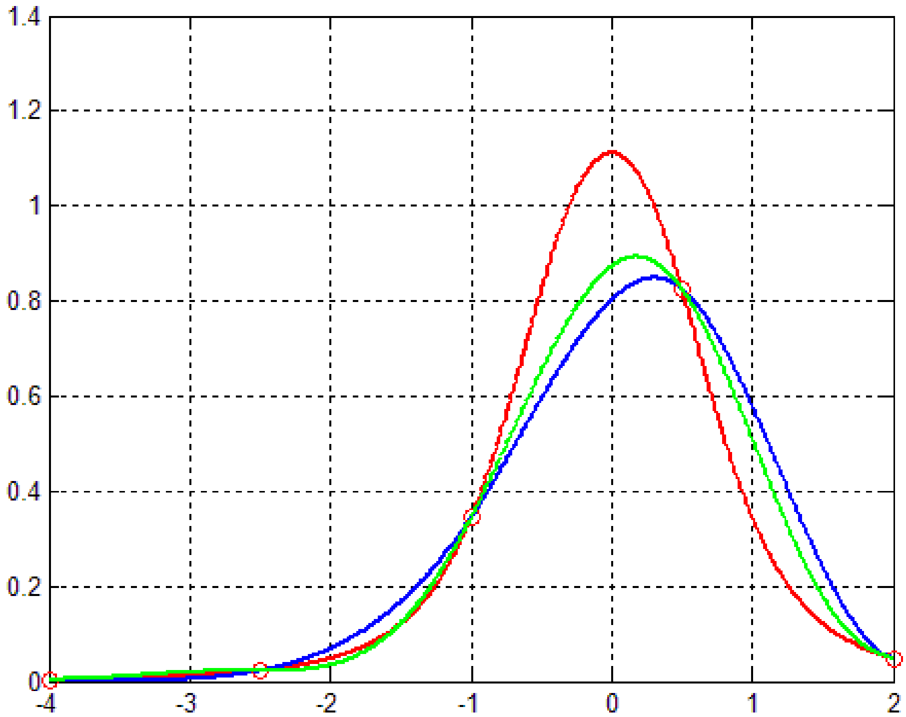
Pavyzdys. Funkciją $y(x)=\dfrac{1}{x^4+x^2+0.9}$ interpoliuokime trečios ir penktos eilės splainais su tokiomis kraštinėmis sąlygomis:

- trečios eilės splainui $g(x)$: $g'(x_0)=y'(x_0), g'(x_N)=y'(x_N)$,
- penktos eilės splainui $g(x)$:

$$g'(x_0)=y'(x_0), g''(x_0)=y''(x_0), g'(x_N)=y'(x_N), g''(x_N)=y''(x_N),$$

kai interpoliavimo mazgai yra: x= {-4; -2.5; -1; 0.5; 2}.

Funkcijos *y(x)* ir interpoliacinių splainų grafikai pavaizduoti 5.12 paveiksle.



5.12 pav. Funkcijos $y(x)=\dfrac{1}{x^4+x^2+0.9}$ *(mėlynas grafikas) interpoliacija trečios (raudonas grafikas) ir penktos (žalias grafikas) eilės splainais.*

5.1.2.4. Kraštinių sąlygų parinkimas ir interpoliacinių splainų konvergavimas.

Praktiškai taikant kubinius splainus, reikia tinkamai parinkti kraštines sąlygas. Tai ypač svarbu, jei splainas turi gerai sutapti su aproksimuojamąja funkcija interpoliavimo intervalo galinėse srityse. Kubinių splainų kraštinės sąlygos didžiausią įtaką splaino formai turi interpoliavimo intervalo galinėse srityse ir labai nedidelę — kitose to intervalo srityse. Vadinasi, kraštines sąlygas galima parinkti remiantis lokaliosiomis aproksimuojamosios funkcijos savybėmis.

Žinant $f'(x_0)$ ir $f'(x_N)$ arba $f''(x_0)$ ir $f''(x_N)$ reikšmes, tikslinga reikalauti, kad splaino $g(x)$ atitinkamos eilės išvestinės taškuose x_0 ir x_N sutaptų su funkcijos $f(x)$ tos pačios eilės išvestinėmis. Jeigu galima rinktis pirmosios arba antrosios eilės išvestines, tai pirmenybė teikiama pirmosios eilės išvestinėms.

Kai aproksimuojamoji funkcija yra periodinė, be abejo, verta rinktis periodines kraštines sąlygas. Natūralios kraštinės sąlygos paprastai blogai aproksimuoja funkciją taškuose, artimuose interpoliavimo intervalo galams. Jei interpoliuojamosios funkcijos išvestinės nežinomos, tai vietoj natūralių kraštinių sąlygų imamos 5-ojo tipo (žr. 5.1.2.1 paragrafą) kraštinės sąlygos.

Išnagrinėkime kubinio interpoliacinio splaino ir jo išvestinių konvergavimą prie aproksimuojamosios funkcijos ir jos išvestinių. Tarkime, kad segmente $[a;b]$ yra duotas pastovaus žingsnio tinklelis, t. y.

$\Delta:\ a = x_0 < x_1 < \ldots < x_{N-1} < x_N = b$ ir $h_i = x_i - x_{i-1} = h$, kai visi $i = \overline{1, N}$.

Apibrėžkime funkcijos tolydumo modulį. Sakykime, segmente $[a;b]$ nusakyta funkcija $\varphi(x)$. Tada dydis

$$\omega(h,\varphi) = \sup_{\substack{x_1, x_2 \in [a;b], \\ |x_1 - x_2| \le h}} |\varphi(x_1) - \varphi(x_2)|$$

vadinamas funkcijos $\varphi(x)$ tolydumo moduliu. Jis rodo didžiausią pagal modulį funkcijos $\varphi(x)$ pokytį ilgio h atkarpoje, kai h priklauso segmentui $[a;b]$.

Tarkime, kad segmente $[a;b]$ nusakyta tolydžioji funkcija $f(x)$, turinti tolydžiąsias pirmosios ir antrosios eilės išvestines. Šią funkciją interpoliuokime kubiniu splainu $g(x)$, kuris nusakomas tinkleliu Δ ir tenkina kraštines sąlygas $g'(x_0) = f'(x_0),\ g'(x_N) = f'(x_N)$. Tada teisingi tokie rezultatai:

$$|f''(x) - g''(x)| \le 5\omega(h, f''),\ |f'(x) - g'(x)| \le 5h\omega(h, f''),\ |f(x) - g(x)| \le \frac{5}{2}h^2\omega(h, f'').$$

Jie teisingi ir esant 2-ojo tipo (žr. 5.1.2.1 paragrafą) bei periodinėms kraštinėms sąlygoms.

Didėjant funkcijos glodumo laipsniui, t. y. funkcijai $f(x)$ turint aukštesnės nei antrosios eilės tolydžiųjų išvestinių, konvergavimo greitis didėja. Tačiau negalima pasiekti didesnio konvergavimo greičio nei $O(h^4)$.

5.1.2.5. Lokalieji splainai.

Iš anksčiau pateikto nagrinėjimo matyti, kad, norint apskaičiuoti interpoliacinį splainą, reikia suformuoti ir išspręsti tiesinių lygčių sistemą. Atsižvelgiant į interpoliavimo sąlygų skaičių, ši sistema gali būti labai didelė. Be to, pakeitus kraštines sąlygas arba bent vieną interpoliavimo tašką, daugiau ar mažiau keičiasi ir visa splaino forma. Tačiau interpoliacinio splaino apskaičiavimo uždavinį galima suformuluoti taip, kad minėtą sistemą būtų galima suskaidyti į mažesnės apimties nepriklausomų posistemių aibę. Tada kiekviename posistemyje splaino parametrams apskaičiuoti bus naudojamos tik kelios interpoliavimo sąlygos. Taip apskaičiuoti splainai vadinami lokaliaisiais. Aišku, kad pats paprasčiausias lokaliojo splaino pavyzdys yra pirmosios eilės interpoliacinis splainas — laužtė, einanti per interpoliavimo taškus. Jos lygtis intervale $[x_{i-1};x_i]$ $(i = \overline{1, N})$ yra tokia:

$$g(x) = \frac{x_i - x}{h_i} y_{i-1} + \frac{x - x_{i-1}}{h_i} y_i;\ \text{čia}\ h_i = x_i - x_{i-1}.$$

Pakeitus interpoliavimo tašką $(x_i; y_i)$, splaino išraiška keičiasi tik intervaluose $[x_{i-1}; x_i]$ ir $[x_i; x_{i+1}]$.

Toliau nagrinėsime aukštesnės eilės lokaliuosius splainus — Ermito kubinius splainus ir Akima splainus.

Ermito kubiniai splainai. Tarkime, kad tinklelio $\Delta: x_0 < x_1 < ... < x_N$ taškuose yra žinomos funkcijos $y = f(x)$ bei jos išvestinės reikšmės, t. y. $f(x_i) = y_i$, $f'(x_i) = y'_i$, $i = \overline{0, N}$.

Reikia apskaičiuoti kubinį defekto 2 splainą $g(x)$, kuris tenkintų šias sąlygas:

1) kiekviename intervale $[x_i; x_{i+1}]$ $(i = \overline{0, N-1})$ $g(x)$ yra kubinis polinomas;

2) $g(x_i) = y_i$, $g'(x_i) = y'_i$, $i = \overline{0, N}$.

Toks splainas vadinamas Ermito kubiniu splainu. Kadangi kiekviename intervale $[x_i; x_{i+1}]$ kubinį splainą nusako keturi koeficientai, kurie turi tenkinti keturias sąlygas

$$g(x_i) = y_i, \quad g'(x_i) = y'_i, \quad g(x_{i+1}) = y_{i+1}, \quad g'(x_{i+1}) = y'_{i+1},$$

tai toks polinomas egzistuoja ir yra vienintelis. Be to, jo išraiška priklauso tik nuo intervalo $[x_i; x_{i+1}]$ sąlygų, todėl splainas $g(x)$ yra lokalusis, taigi, pakeitus y_i arba y'_i, splaino forma keičiasi tik intervaluose $[x_{i-1}; x_i]$ bei $[x_i; x_{i+1}]$ ir nesikeičia kituose intervaluose.

Splaino $g(x)$ išraiška intervale $[x_i; x_{i+1}]$ bus paprastesnė, jeigu jos ieškosime tokio pavidalo:

$$g(x) = a + b(x - x_i) + c\frac{(x - x_i)^2}{h_i} + d\frac{(x - x_i)^3}{h_i^2}; \text{ čia } h_i = x_{i+1} - x_i.$$

Iš sąlygos $g(x_i) = y_i$ apskaičiuojame koeficientą *a*: $a = y_i$. Kadangi

$$g'(x) = b + 2c\frac{x - x_i}{h_i} + 3d\frac{(x - x_i)^2}{h_i^2}, \text{ tai } g'(x_i) = y'_i = b. \text{ Be to,}$$

$$g(x_{i+1}) = g(x_i + h) = a + bh_i + ch_i + dh_i = y_{i+1} \text{ ir } g'(x_{i+1}) = b + 2c + 3d = y'_{i+1}.$$

Taigi koeficientus c ir d apskaičiuojame išsprendę lygčių sistemą

$$\begin{cases} c + d = (y_{i+1} - y_i)/h_i - y'_i, \\ 2c + 3d = y'_{i+1} - y'_i. \end{cases}$$

Iš jos

$$d = y'_i + y'_{i+1} - \frac{2(y_{i+1} - y_i)}{h_i}, \quad c = (y_{i+1} - y_i)/h_i - y'_i - d.$$

Vadinasi, Ermito splaino analizinė išraiška intervale $[x_i; x_{i+1}]$ $(i = \overline{0, N-1})$ yra tokia:

$$g(x) = y_i + (x - x_i)(y'_i + t(c + td)); \text{ čia } t = \frac{x - x_i}{h_i}.$$

Ermito kubinio splaino interpoliavimo paklaidą priklauso nuo interpoliuojamos funkcijos glodumo ir geriausiu atveju skirtumas tarp Ermito kubinio splaino bei funkcijos r-tosios $\left(r=\overline{0,3}\right)$ eilės išvestinių yra $O\left(\|\Delta\|^{4-r}\right)$; čia $\|\Delta\|=\max\limits_{1\le i\le N} h_i$.

Žemiau patalpinta funkcija ***ermitspln***, apskaičiuojanti ermito kubinį splainą.

```
function gt=ermitspln(x,y,yis,t);
% ERMITSPLN apskaičiuoja Ermito kubinio interpoliacinio splaino
% reikšmes, kai x=t.
% Įėjimo parametrai
% x  - tinklelio mazgai,
% y  - funkcijos reikšmės,
% yi - splaino pirmųjų išvestinių reikšmės,
% t  - masyvas argumento reikšmių, kuriose norime apskaičiuoti
% splaino reikšmes.
% Išėjimo parametrai
% gt - splaino reikšmės taškuose x=t.

m=numel(t); n=numel(x);
if (t(1)<x(1)) | (t(m)>x(n))
    disp('argumentas t už intervalo ribų')
    gt=nan; return
end
h=x(2:n)-x(1:n-1); % žingsnių masyvas
% Kuriems intervalams priklauso t reikšmės
tm=repmat(t',1,n); xx=repmat(x,m,1);
ind=n-sum((xx>=tm)');
if ind(1)==0
    ind(1)=1;
end
% gt reikšmių apskaičiavimas
xi1=x(ind); xi=x(ind+1);
yi1=y(ind); yi=y(ind+1);
hi=h(ind);
fmi1=yis(ind); fmi=yis(ind+1);
d=fmi+fmi1-2*(yi-yi1)./hi;
c=(yi-yi1)./hi-fmi1-d;
tt=(t-xi1)./hi;
gt=yi1+(t-xi1).*(fmi1+tt.*(c+tt.*d));
```

Akima kubiniai splainai. Akima kubiniai splainai iš esmės yra Ermito kubiniai splainai. Skirtumas yra tik tas, kad, turint interpoliavimo taškus $(x_i, y_i), i=\overline{1,N}$, funkcijos išvestinės reikšmės juose apskaičiuojamos taip, kad, išbrėžus Ermito kubinį splainą, jis yra panašus į dailininko išpieštą kreivę, t.y. neturi bereikalingų švytavimų. Vadinasi, pagrindinis uždavinys ir yra, kaip, turint interpoliavimo taškus $(x_i, y_i), i=\overline{1,N}$, apskaičiuoti juose išvestinių reikšmes.

Žemiau aptarsime du minėtų išvestinių apskaičiavimo metodus.

Akima formulės . Tam, kad mazge x_i apskaičiuotume pirmosios išvestinės reikšmę, naudosime penkis gretimus taškus:
$(x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2})$.

Pasinaudodami šiais taškais, apskaičiuojame keturis skirtuminius santykius:

$$d_j = \frac{y_{j+1} - y_j}{x_{j+1} - x_j}, j = i-2, i-1, i, i+1.$$

Tada funkcijos išvestinės reikšmės ***vidiniuose mazguose*** $x_i, i = \overline{2, N-1}$ apskaičiuojamos pagal formulę:

$$y_i' = \frac{w_{i-1}d_{i-1} + w_i d_i}{w_{i-1} + w_i}, \text{ čia } w_{i-1} = |d_{i+1} - d_i|, w_i = |d_{i-1} - d_{i-2}|.$$

Ši formulė pagrįsta tokiu euristiniu samprotavimu: $y_i' \approx d_{i-1}$ kuo yra mažesnis skirtumas tarp d_{i-2} ir d_{i-1}; analogiškai, $y_i' \approx d_i$ kuo yra mažesnis skirtumas tarp d_{i+1} ir d_i. Be to, kad išvengtume bereikalingų švytavimų kai ordinates sutampa, t.y. kai $y_{i-2} = y_{i-1} = y_i$ (šiuo atveju $d_{i-2} = d_{i-1} = 0$), išvestinės reikšmė $y_i' = 0$.

Aptarkime y_i' apibrėžimą, kai ne visos d_j reikšmės yra skirtingos.

$y_i' = d_{i-1}$, jei $d_{i-2} = d_{i-1}$ ir $d_i \neq d_{i+1}$;

$y_i' = d_i$, jei $d_i = d_{i+1}$ ir $d_{i-2} \neq d_{i-1}$;

$y_i' = d_{i-1} = d_i$, jei $d_{i-1} = d_i$;

y_i' reikšmė negali būti apskaičiuota pagal anksčiau pateiktą formulę, jei $d_{i-2} = d_{i-1} \neq d_i = d_{i+1}$.

Šiuo atveju y_i' reikšmė apskaičiuojama pagal formulę:

$$y_i' = \frac{d_{i-1} + d_i}{2}.$$

Išvestinės reikšmės ***kraštiniuose mazguose***: x_1 ir x_N. Tam tikslui kiekviename intervalo $[x_1, x_N]$ gale įveskime po du papildomus mazgus: x_{-1}, x_0 kairiajame intervalo gale ir x_{N+1}, x_{N+2} dešiniajame intervalo gale. Mazgus apibrėžkime taip, kad jie tenkintų sąlygą:

tegu $h_1 = x_2 - x_1, h_2 = x_3 - x_2$ ir analogiškai $h_{N-2} = x_{N-1} - x_{N-2}, h_{N-1} = x_N - x_{N-1}$;

tada $x_0 = x_1 - h_2, x_{-1} = x_0 - h_1$ ir $x_{N+1} = x_N + h_{N-2}, x_{N+2} = x_{N+1} + h_{N-1}$, kitaip tariant, $x_3 - x_1 = x_2 - x_0 = x_1 - x_{-1}$ ir $x_N - x_{N-2} = x_{N+1} - x_{N-1} = x_{N+2} - x_N$.

y-kų reikšmes mazgams x_{-1}, x_0 apskaičiuosime iš parabolės $P_2(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)^2$, kuri eina per taškus $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, t.y. $y_0 = P_2(x_0), y_{-1} = P_2(x_{-1})$.

Taip įvedus papildomus taškus, turėsime: $d_{-1} - d_0 = d_0 - d_1 = d_1 - d_2$. Vadinasi,

$$d_0 = 2d_1 - d_2 \text{ ir } d_{-1} = 2d_0 - d_1.$$

Analogiškai, *y*-kų reikšmes mazgams x_{N+1}, x_{N+2} apskaičiuosime iš parabolės $P_2(x) = a_0 + a_1(x - x_N) + a_2(x - x_N)^2$, kuri eina per taškus $(x_{N-2}, y_{N-2}), (x_{N-1}, y_{N-1}), (x_N, y_N)$, t.y. $y_{N+1} = P_2(x_{N+1}), y_{N+2} = P_2(x_{N+2})$.

Ir šiuo atveju galioja lygybės: $d_{N-1} - d_{N-2} = d_N - d_{N-1} = d_{N+1} - d_N$. Vadinasi,

$d_N = 2d_{N-1} - d_{N-2}$ ir $d_{N+1} = 2d_N - d_{N-1}$.

Modifikuotos Akima formulės. Aptarkime modifikuotas Akima formules, kurios įgalina papildomai sumažinti, lyginant su anksčiau išnagrinėtomis formulėmis, Ermito splaino bereikalingus švytavimus. Šias formules naudoja MATLAB‘o funkcija ***pchip***.

Išvestinių reikšmės vidiniuose taškuose $(x_k ; y_k), k = \overline{2.N-1}$. Apskaičiuokime skirtuminius santykius $d_i = \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i}, i = \overline{1, N-1}$. Tegu $h_k = x_{k+1} - x_k$.

Išvestinų yi_k reikšmės vidiniuose taškuose yra;

- d_{k-1} ir d_k svertinis vidurkis, jei d_{k-1} ir d_k ženklai sutampa;
- yi_k reikšmė lygi nuliui, jei d_{k-1} ir d_k yra priešingų ženklų arba vienas iš jų lygus nuliui.

Svertinis vidurkis apskaičiuojamas pagal formules:

tegu $hs = h_k + h_{k+1}$, $w1 = \dfrac{h_k + hs}{3hs}$, $w2 = \dfrac{h_{k+1} + hs}{3hs}$, $d\max = \max(|d_k|, |d_{k+1}|)$ ir $d\min = \min(|d_k|, |d_{k+1}|)$;

tada $$yi_{k+1} = \frac{d\min}{w1d_k / d\max + w2d_{k+1} / d\max}.$$