6. Aproksimacija trigonometriniais polinomais

Tolydžioji aproksimacija. Trigonometriniai polinomai naudojami periodinėms funkcijoms aproksimuoti. Primename, kad funkcija $y = f(x)$ yra periodinė, jei egzistuoja tokia konstanta *T*, kad $f(x+T) = f(x)$ visoms *x* reikšmėm. Be to, nesiaurindami uždavinio, galime laikyti, kad $T = 2\pi$, o $x \in [-\pi;\pi]$.

Apibrėžimas. *n*-to laipsnio trigonometrinis polinomas yra bazinių funkcijų $\{\varphi_o, \varphi_1, \ldots, \varphi_{2n-1}\}$ tiesinis darinys, čia:

$$\varphi_0(x) = \frac{1}{\sqrt{2\pi}},\ \varphi_k(x) = \frac{1}{\sqrt{\pi}}\cos kx, k = 1,2,\ldots,n,\ \varphi_{n+k}(x) = \frac{1}{\sqrt{\pi}}\sin kx, k = 1,2,\ldots,n-1.$$

Reikia pabrėžti, kad bazinių funkcijų $\{\varphi_o, \varphi_1, \ldots, \varphi_{2n-1}\}$ sistema intervale $[-\pi;\pi]$yra ortonormuota su svoriu $w(x) = 1$, t.y. $\int_{-\pi}^{\pi} \varphi_{n+k}(x)\varphi_j(x)dx = 0$, jei $j \neq 0$ ir $j \neq k$, ir

$\int_{-\pi}^{\pi} \varphi_j^2(x)dx = 1$, $j = \overline{0,2n-1}$.

Tarkime, kad funkcija $y = f(x), x \in [-\pi;\pi]$ yra periodinė su periodu $T = 2\pi$. Šią funkciją aproksimuokime trigonometriniu polinomu $S_n(x) = \sum_{k=0}^{2n-1} a_k \varphi_k(x)$ mažiausių kvadratų metodu. Galima įrodyti, kad tada koeficientai a_k yra apskaičiuojami pagal formulę: $a_k = \int_{-\pi}^{\pi} f(x)\varphi_k(x)dx, k = \overline{0,2n-1}$.

Kai $n \to \infty$, polinomas $S_n(x)$ vadinamas Furje eilute.

Tam, kad koeficientus a_k apskaičiuotume skaitiniu būdu, reikia naudoti specialias kvadratūrines formulės, nes 8-me skyriuje išnagrinėtos Niutono ir Koteso bei Gauso tipo kvadratūrinės formulės šiems integralams apskaičiuoti netinka, kai $k >> 1$. Todėl šiems koeficientams apskaičiuoti taikomi specialūs metodai, skirti greitai osciliuojančioms funkcijoms integruoti. Tai Filono tipo kvadratūrinės formulės, kurios plačiai išnagrinėtos vadovėlyje [6], o taip pat asimptotinės [25,26] bei Olvero [34] kvadratūrinės formulės. Šių formulių čia nenagrinėsime.

Pavyzdys. *Trigonometriniu polinomu aproksimuokime funkciją*

$$f(x) = |x|,\ x \in [-\pi;\pi].$$

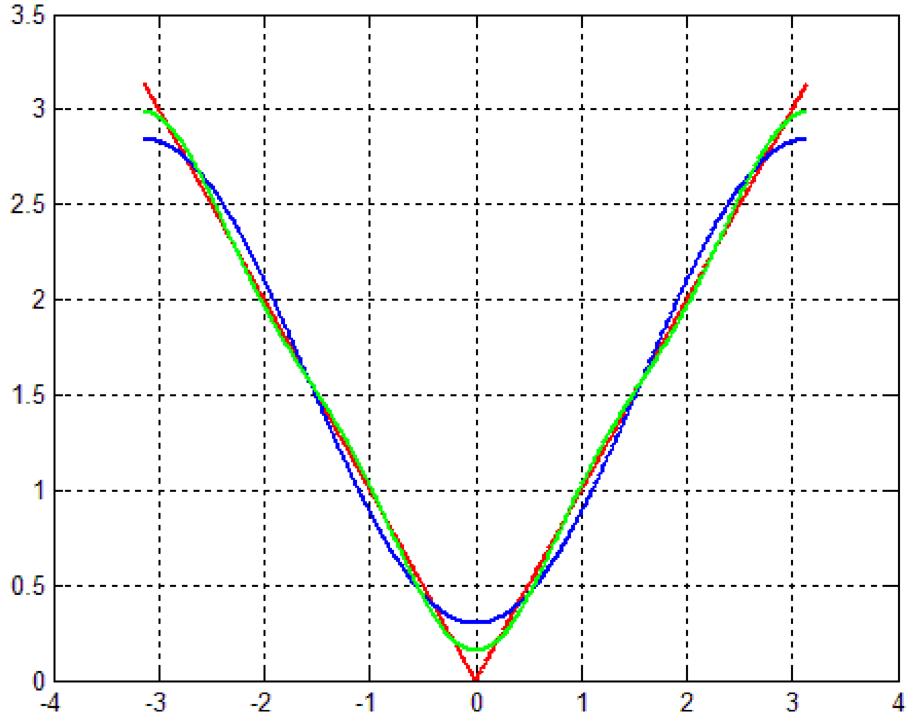
$$a_0 = \int_{-\pi}^{\pi} |x| \frac{1}{\sqrt{2\pi}} dx = \frac{\sqrt{2}\pi^2}{2\sqrt{\pi}},\ a_k = \frac{1}{\sqrt{\pi}} \int_{-\pi}^{\pi} |x| \cos kx dx = \frac{2}{\sqrt{\pi}k^2}\left[(-1)^k - 1\right], k = \overline{1,n}.$$

Koeficientai a_{n+k} Furje eilutėje žymimi b_k, t.y. $b_k \equiv a_{n+k}, k = 1,2,\ldots,n-1$. Mūsų pavyzdyje $b_k = \frac{1}{\sqrt{\pi}} \int_{-\pi}^{\pi} |x| \sin kx dx = 0, k = \overline{1,n-1}$.

Vadinasi, trigonometrinis polinomas, aproksimuojantis funkciją $f(x) = |x|$ yra

$$S_n(x) = \frac{\pi}{2} + \frac{2}{\pi} \sum_{k=0}^{n} \frac{(-1)^k - 1}{k^2} \cos kx.$$

6.1 paveiksle parodyta funkcijos $f(x)=|x|$ aproksimacija polinomais $S_1(x)=\frac{\pi}{2}-\frac{4}{\pi}\cos x$ ir $S_3(x)=\frac{\pi}{2}-\frac{4}{\pi}\cos x-\frac{4}{9\pi}\cos 3x$.



6.1 pav. Funkcijos $f(x)=|x|$ (raudonas grafikas) aproksimacija polinomais $S_1(x)$ (mėlynas grafikas) ir $S_3(x)$ (žalias grafikas).

Furje eilutė funkcijai $f(x)=|x|$ yra $S(x)=\lim\limits_{n\to\infty} S_n(x)=\frac{\pi}{2}+\frac{2}{\pi}\sum\limits_{k=1}^{\infty}\frac{(-1)^k-1}{k^2}\cos kx$, kuri konverguoja visoms realioms x reikšmėms.

Diskrečioji aproksimacija. Praktikoje žymiai dažniau naudojamas diskretusis Furje eilutės analogas. Tarkime, kad žinomas didelis skaičius aproksimuojamos funkcijos reikšmių, apskaičiuotų lygiai vienas nuo kito nutolusiuose taškuose, ir šią funkciją norime aproksimuoti trigonometriniu polinomu mažiausių kvadratų metodu.

Nesiaurindami klausimo nagrinėsime intervalą $[-\pi;\pi]$, t.y. kai funkcijos diskretizavimo mazgai x_j nusakomi formule

$$x_j=-\pi+\left(\frac{j}{m}\right)\pi,\ j=0,1,\ldots,2m-1.$$

Pastaba. Bet kokį intervalą [*a*;*b*] galima transformuoti į intervalą $[-\pi;\pi]$ pagal formulę $t=\frac{2x-(b+a)}{b-a}\pi$.

Diskrečiajai aproksimacijai patogiau naudoti truputį kitokią nei anksčiau apibrėžtą bazinių funkcijų sistemą, t.y. fiksuotam $n < m$ apibrėžkime bazinių funkcijų sistemą $\phi_0(x) = \frac{1}{2}, \phi_k(x) = \cos kx, k = 1,2,\dots,n$ ir $\phi_{n+k}(x) = \sin kx, k = 1,2,\dots,n-1$.

Jei diskretizavimo mazgai $x_j, j = \overline{0,2m-1}$ yra tolygiai išsidėstę intervale $[-\pi;\pi]$, tai ši bazinių funkcijų sistema yra ortogonali sumavimo atžvilgiu, t.y.

$$\sum_{j=0}^{2m-1} \phi_k(x_j)\phi_l(x_j) = 0, k \neq l .$$

Pastaba. Bazinių funkcijų ortogonalumas pagrįstas tuo, kad intervale $[-\pi;\pi]$ tolygiai išdėstytiems diskretizavimo mazgams teisingos lygybės $\sum_{j=0}^{2m-1} \cos kx_j = 0$, $\sum_{j=0}^{2m-1} \sin kx_j = 0$, jei $k < 2m$.

Uždavinio formulavimas. Turime funkcijos $y = f(x)$ reikšmių lentelę $(x_j, y_j), j = \overline{0,2m-1}$.

Raskime tokį bazinių funkcijų tiesinį darinį

$$S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kx + a_{n+k} \sin kx),$$

kad

$$z = \sum_{j=0}^{2m-1} (y_j - S_n(x_j))^2 = \sum_{j=0}^{2m-1} \left(y_j - \left(\frac{a_0}{2} + a_n \cos nx_j + \sum_{k=1}^{n-1} (a_k \cos kx_j + a_{n+k} \sin kx_j) \right) \right)^2$$

įgautų pačią mažiausią reikšmę.

Pastaba. Kaip ir Furje eilutės atveju, toliau koeficientus a_{n+k} žymėsime b_k, t.y. $a_{n+k} \equiv b_k, k = \overline{1,n-1}$.

Tam, kad apskaičiuotume koeficientus a_k ir b_k, kurie minimizuotų z reikšmę, reikia rasti funkcijos z dalines išvestines pagal a_k bei b_k ir jas prilyginti nuliui. Tada ieškomos a_k ir b_k reikšmės yra tos lygčių sistemos sprendinys. Kadangi bazinių funkcijų sistema yra ortogonali sumavimo atžvilgiu, tai sistemos sprendinys yra

$$a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j, k = 0,1,2,\dots,n \text{ ir } b_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \sin kx_j, k = 1,2,\dots,n-1 .$$

Pavyzdys. *Funkciją $f(x) = x^4 - 3x^3 + 2x^2 - tgx(x-2)$ intervale [0; 2] nusakytą reikšmių lentele, kai diskretizavimo mazgai yra $x_j = j/5, j = \overline{0,9}$, mažiausių kvadratų metodu aproksimuokime trigonometriniu polinomu $S_3(x)$.*

Pirmiausia intervalą [0; 2] transformuokime į intervalą $[-\pi;\pi]$. Pasinaudodami anksčiau pateikta transformavimo formule, turėsime $t_j = (x_j - 1)\pi$.

Tada pradiniai duomenys transformuosis į lentelę $(t_j; y_j)$, $j = \overline{0,9}$.

Ieškomasis trigonometrinis polinomas yra

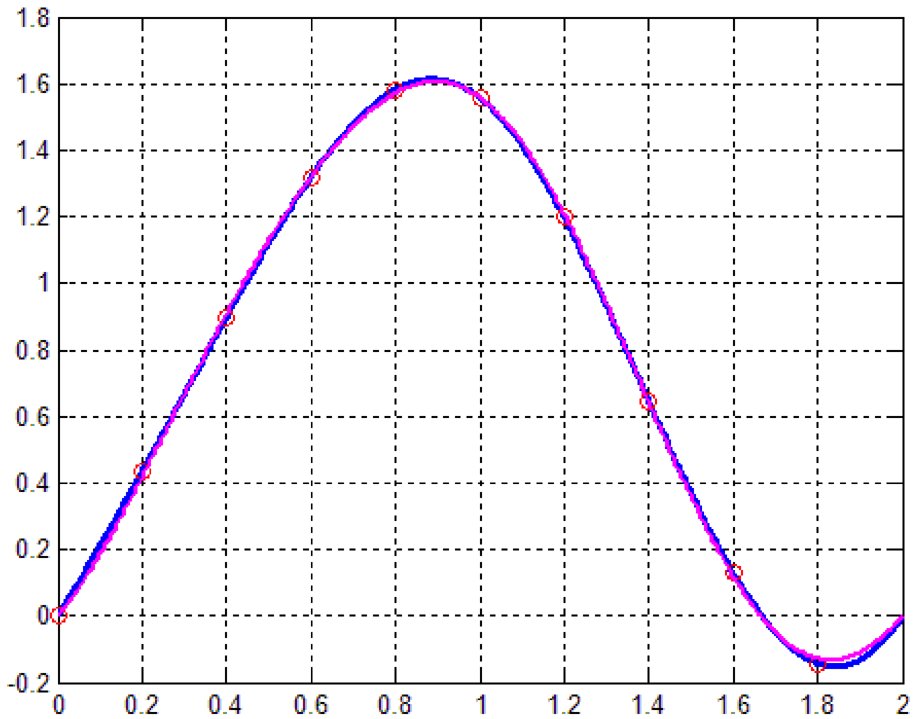
$S_3(t) = \dfrac{a_0}{2} + a_3 \cos 3t + \sum_{k=1}^{2}(a_k \cos kt + b_k \sin kt)$, čia $a_k = \dfrac{1}{5}\sum_{j=0}^{9} y_j \cos kt_j, k = \overline{0,3}$,

$b_k = \dfrac{1}{5}\sum_{j=0}^{9} y_j \sin kt_j, k = \overline{0,2}$.

Atlikę skaičiavimus ir vėl transformavę į intervalą [*0; 2*], turėsime :

$$S_3(x) = 0.76201 + 0.77177\cos\pi(x-1) + 0.017423\cos 2\pi(x-1) + 0.0065673\cos 3\pi(x-1) - \\ - 0.38676\sin\pi(x-1) + 0.047806\sin 2\pi(x-1).$$

6.2 paveikslėlyje grafiškai pavaizduota funkcija $f(x) = x^4 - 3x^3 + 2x^2 - tgx(x-2)$ ir ją aproksimuojantis trigonometrinis polinomas $S_3(x)$.



6.2 pav. Funkcijos $f(x) = x^4 - 3x^3 + 2x^2 - tgx(x-2)$ *grafikas* (*mėlyna spalva*) *ir* $S_3(x)$ *grafikas* (*magnum spalva*).

Greitoji Furje transformacija. Ankstesnėse pastraipose aptarėme periodinių funkcijų aproksimaciją trigonometriniais polinomais, apskaičiuotais mažiausių kvadratų metodu.

Dabar aptarsime šių funkcijų aproksimaciją interpoliaciniais trigonometriniais polinomais.

Uždavinio formulavimas. Tarkime, kad žinomos periodinės funkcijos reikšmės lygiai nutolusiuose vienas nuo kito mazguose (x_j, y_j), $j = \overline{0,2m-1}$, t.y. turime $2m$ funkcijos reikšmių.

Mažiausių kvadratų metodu apskaičiuosime aproksimuojantį trigonometrinį polinomą

$$S_m(x) = \frac{a_0 + a_m \cos mx}{2} + \sum_{k=1}^{m-1} (a_k \cos kx + b_k \sin kx).$$

Pastaba. Polinomo $S_m(x)$ išraiška truputį skiriasi nuo anksčiau nagrinėtos trigonometrinio polinomo išraiškos.

Kadangi polinomą $S_m(x)$ apibrėžia $2m$ nežinomų koeficientų ir taip pat turime $2m$ funkcijos reikšmių, tai mažiausių kvadratų metodu apskaičiuotas polinomas tuo pačiu bus ir interpoliacinis polinomas. Vadinasi, interpoliacinio trigonometrinio polinomo $S_m(x)$ koeficientus galime skaičiuoti pagal anksčiau pateiktas a_k ir b_k formules.

Interpoliaciniai trigonometriniai polinomai, esant dideliam funkcijos reikšmių skaičiui, gana tiksliai aproksimuoja funkciją. Todėl jie naudojami daugelyje fizikos ir technikos sričių (konstruojant skaitmeninius filtrus, kvantinėje mechanikoje, optikoje, signalų analizėje ir pan.). Tačiau tokiems interpoliacinams polinomams apskaičiuoti reikės atlikti daug aritmetinių veiksmų: apytiksliai $(2m)^2$ daugybos ir tiek pat sumavimo veiksmų. Kai funkcijos taškų yra tūkstančiai, tai, tiesiogiai skaičiuojant koeficientus pagal anksčiau pateiktas formules, reikės atlikti milijonus sudėties bei daugybos operacijų. Todėl iki 60-jų metų interpoliacija buvo naudojama retai, nes skaičiavimo laikas buvo per daug ilgas, o apvalinimo paklaidos žymiai įtakojo rezultatų tikslumą. Situacija pasikeitė, kai 1965 m. J.W. Cooley ir J.W. Tukey pasiūlė interpoliacinių trigonometrinių polinomų $S_m(x)$ apskaičiavimo algoritmą, kurio sudėtingumas yra $O(m \log_2 m)$. Šis algoritmas yra žinomas, kaip ***greitosios Furje transformacijos algoritmas.***

Pastaba. Iš tikrųjų šis algoritmas buvo sugalvotas žymiai anksčiau, tačiau daugelio tyrėjų iki tol buvo nepastebėtas.

Aptarkime šį algoritmą.

Primename, kad interpoliavimo mazgus apibrėžia formulė

$$x_j = -\pi + \left(\frac{j}{m}\right)\pi, \; j = 0,1,\ldots,2m-1,$$

o interpoliacinio polinomo $S_m(x)$ koeficientai apskaičiuojami pagal formules

$$a_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \cos kx_j, k = 0,1,2,\ldots,m \text{ ir } b_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \sin kx_j, k = 0,1,2,\ldots,m.$$

Pastaba. Skaičiavimo patogumui į koeficientų rinkinį yra įtraukti polinomo $S_m(x)$ išraiškoje nefigūruojantys koeficientai b_0 ir b_m, kurių reikšmės yra lygios nuliui.

Pagrindinis greitosios Furje transformacijos algoritmo momentas yra tas, kad koeficientai a_k ir b_k pagal pateiktas formules tiesiogiai neskaičiuojami, o vietoj jų yra skaičiuojami išraiškos

$$F(x) = \frac{1}{m}\sum_{k=0}^{2m-1} c_k e^{ikx}$$

kompleksiniai koeficientai $c_k = \sum_{j=0}^{2m-1} y_j e^{\frac{\pi ijk}{m}}, k = \overline{0,2m-1}$.

Žinant koeficientų c_k reikšmes, koeficientai a_k ir b_k yra apskaičiuojami remiantis Oilerio formule $e^{ix} = \cos x + i\sin x$.

Tada

$$\frac{1}{m}c_k e^{-i\pi k} = \frac{1}{m}\sum_{j=0}^{2m-1} y_j e^{\frac{\pi ijk}{m}} e^{-i\pi k} = \frac{1}{m}\sum_{j=0}^{2m-1} y_j e^{ik\left(-\pi + \left(\frac{j}{m}\right)\pi\right)} = \frac{1}{m}\sum_{j=0}^{2m-1} y_j\left(\cos kx_j + i\sin kx_j\right).$$

Vadinasi, $\frac{1}{m}c_k e^{-i\pi k} = a_k + ib_k$.

Skaičiavimo veiksmų skaičiaus sumažėjimas, skaičiuojant koeficientus c_k, greituoju Furje transformacijos algoritmu pagrįstas tuo, kad jame veiksmai yra sugrupuoti.

Panagrinėkime pavyzdį, iliustruojantį greitosios Furje transformacijos algoritmą.

Pavyzdys. Tarkime, kad funkcija nusakyta taškais (x_j, y_j), $j = \overline{0,2m-1}$ ir $m = 4$ (paprastai $m = 2^p$, p - natūralusis skaičius) , t.y. interpoliavimo mazgus apibrėžia formulė

$$x_j = -\pi + j\frac{\pi}{4}, j = \overline{0,7}.$$

Tada

$$S_4(x) = \frac{a_0 + a_4\cos mx}{2} + \sum_{k=1}^{3}\left(a_k\cos kx + b_k\sin kx\right),$$

čia $a_k = \frac{1}{4}\sum_{j=0}^{7} y_j\cos kx_j, k = 0,1,2,3,4$ ir $b_k = \frac{1}{4}\sum_{j=0}^{7} y_j\sin kx_j, k = 1,2,3$.

Apibrėžkime funkciją $F(x) = \frac{1}{4}\sum_{k=0}^{7} c_k e^{ikx}$, kur $c_k = \sum_{j=0}^{7} y_j e^{ijk\pi/4}, k = \overline{0,7}$.

Tiesiogiai pagal anksčiau pateiktą formulę apskaičiavę koeficientus c_k, turėsime:

$c_0 = y_0 + y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7$;

$c_1 = y_0 + \frac{i+1}{\sqrt{2}}y_1 + iy_2 + \frac{i-1}{\sqrt{2}}y_3 - y_4 - \frac{i+1}{\sqrt{2}}y_5 - iy_6 - \frac{i-1}{\sqrt{2}}y_7$;

$c_2 = y_0 + iy_1 - y_2 - iy_3 + y_4 + iy_5 - y_6 - iy_7$;

$c_3 = y_0 + \frac{i-1}{\sqrt{2}}y_1 - iy_2 + \frac{i+1}{\sqrt{2}}y_3 - y_4 - \frac{i-1}{\sqrt{2}}y_5 + iy_6 - \frac{i+1}{\sqrt{2}}y_7$;

$$c_4 = y_0 - y_1 + y_2 - y_3 + y_4 - y_5 + y_6 - y_7;$$
$$c_5 = y_0 - \frac{i+1}{\sqrt{2}} y_1 + iy_2 - \frac{i-1}{\sqrt{2}} y_3 - y_4 + \frac{i+1}{\sqrt{2}} y_5 - iy_6 + \frac{i-1}{\sqrt{2}} y_7;$$
$$c_6 = y_0 - iy_1 - y_2 + iy_3 + y_4 - iy_5 - y_6 + iy_7;$$
$$c_7 = y_0 - \frac{i-1}{\sqrt{2}} y_1 - iy_2 - \frac{i+1}{\sqrt{2}} y_3 - y_4 + \frac{i-1}{\sqrt{2}} y_5 + iy_6 + \frac{i+1}{\sqrt{2}} y_7.$$

Tiesiogiai skaičiuojant koeficientus c_k reikia atlikti 64 daugybos/dalybos ir 56 sudėties/atimties operacijų.

Panagrinėkime greitąjį Furje transformacijos algoritmą. Pirmiausia apibrėžkime konstantas $d_i, i = \overline{0,7}$.

$$d_0 = \frac{1}{2}(c_0 + c_4) = y_0 + y_2 + y_4 + y_6; \quad d_1 = \frac{1}{2}(c_0 - c_4) = y_1 + y_3 + y_5 + y_7;$$
$$d_2 = \frac{1}{2}(c_1 + c_5) = y_0 + iy_2 - y_4 - iy_6; \quad d_3 = \frac{1}{2}(c_1 - c_5) = \frac{i+1}{\sqrt{2}}(y_1 + iy_3 - y_5 - iy_7);$$
$$d_4 = \frac{1}{2}(c_2 + c_6) = y_0 - y_2 + y_4 - y_6; \quad d_5 = \frac{1}{2}(c_2 - c_6) = i(y_1 - y_3 + y_5 - y_7);$$
$$d_6 = \frac{1}{2}(c_3 + c_7) = y_0 - iy_2 - y_4 + iy_6; \quad d_7 = \frac{1}{2}(c_3 - c_7) = \frac{i-1}{\sqrt{2}}(y_1 - iy_3 - y_5 + iy_7);$$

Po to apibrėžkime konstantas $e_i, i = \overline{0,7}$.

$$e_0 = \frac{1}{2}(d_0 + d_4) = y_0 + y_4; \; e_1 = \frac{1}{2}(d_0 - d_4) = y_2 + y_6; \; e_2 = \frac{1}{2}(id_1 + d_5) = i(y_1 + y_5);$$
$$e_3 = \frac{1}{2}(id_1 - d_5) = i(y_3 + y_7); \; e_4 = \frac{1}{2}(d_2 + d_6) = y_0 - y_4; \; e_5 = \frac{1}{2}(d_2 - d_6) = i(y_2 - y_6);$$
$$e_6 = \frac{1}{2}(id_3 + d_7) = \frac{i-1}{\sqrt{2}}(y_1 - y_5); \; e_7 = \frac{1}{2}(id_3 - d_7) = \frac{-i-1}{\sqrt{2}}(y_3 - y_7).$$

Galiausiai apibrėžkime konstantas $f_i, i = \overline{0,7}$.

$$f_0 = \frac{1}{2}(e_0 + e_4) = y_0; \; f_1 = \frac{1}{2}(e_0 - e_4) = y_4; \; f_2 = \frac{1}{2}(ie_1 + e_5) = iy_2;$$
$$f_3 = \frac{1}{2}(ie_1 - e_5) = iy_6; \; f_4 = \frac{1}{2}\left(\frac{i+1}{\sqrt{2}} e_2 + e_6\right) = \frac{i-1}{\sqrt{2}} y_1; \; f_5 = \frac{1}{2}\left(\frac{i+1}{\sqrt{2}} e_2 - e_6\right) = \frac{i-1}{\sqrt{2}} y_5;$$
$$f_6 = \frac{1}{2}\left(\frac{i+1}{\sqrt{2}} e_3 + e_7\right) = -\frac{i+1}{\sqrt{2}} y_3; \; f_7 = \frac{1}{2}\left(\frac{i+1}{\sqrt{2}} e_3 - e_7\right) = -\frac{i+1}{\sqrt{2}} y_7.$$

Konstantos c_i, d_i, e_i ir f_i , $i = \overline{0,7}$ nepriklauso kaip nors ypatingai nuo y reikšmių, o priklauso tik nuo m reikšmės. Nagrinėjamu atveju $m = 4$. Kiekvienai m reikšmei yra vienintelės konstantų c_i, d_i, e_i ir f_i , $i = \overline{0,2m-1}$ aibės. Jos neturi kokios tai taikomosios prasmės, o naudojamos tiktai mažinant skaičiavimų apimtį.

Vadinasi, norėdami apskaičiuoti koeficientus $c_i, i=\overline{0,7}$, elgsimės taip.

1. Apskaičiuosime konstantas $f_i, i=\overline{0,7}$ pagal formules

   $f_0 = y_0$, $f_1 = y_4$; $f_2 = iy_2$; $f_3 = iy_6$; $f_4 = \dfrac{i-1}{\sqrt{2}} y_1$; $f_5 = \dfrac{i-1}{\sqrt{2}} y_5$;

   $f_6 = -\dfrac{i+1}{\sqrt{2}} y_3$; $f_7 = -\dfrac{i+1}{\sqrt{2}} y_7$.

2. Apskaičiuosime konstantas $e_i, i=\overline{0,7}$ pagal formules

   $e_0 = f_0 + f_1$; $e_1 = -i(f_2 + f_3)$; $e_2 = \dfrac{-i+1}{\sqrt{2}}(f_4 + f_5)$; $e_3 = \dfrac{-i+1}{\sqrt{2}}(f_6 + f_7)$;

   $e_4 = f_0 - f_1$; $e_5 = f_2 - f_3$; $e_6 = f_4 - f_5$; $e_7 = f_6 - f_7$.

3. Apskaičiuosime konstantas $d_i, i=\overline{0,7}$ pagal formules

   $d_0 = e_0 + e_1$; $d_1 = -i(e_2 + e_3)$; $d_2 = e_4 + e_5$; $d_3 = -i(e_6 + e_7)$;

   $d_4 = e_0 - e_1$; $d_5 = e_2 - e_3$; $d_6 = e_4 - e_5$; $d_7 = e_6 - e_7$.

4. Apskaičiuosime koeficientus $c_i, i=\overline{0,7}$ pagal formules

   $c_0 = d_0 + d_1$; $c_1 = d_2 + d_3$; $c_2 = d_4 + d_5$; $c_3 = d_6 + d_7$;

   $c_4 = d_0 - d_1$; $c_5 = d_2 - d_3$; $c_6 = d_4 - d_5$; $c_3 = d_6 + d_7$;

Skaičiuojant koeficientus $c_i, i=\overline{0,7}$ pagal pateiktą algoritmą, prireiks 24 daugybos/dalybos veiksmų, o taip pat 24 sudėties/atimties veiksmų. Vadinasi operacijų skaičius yra žymiai mažesnis, negu koeficientus c_k skaičiuojant tiesiogiai, kai reikėjo atlikti 64 daugybos/dalybos ir 56 sudėties/atimties operacijų.

Reikia pabrėžti, kad 4-me žingsnyje koeficientai c_k, nepriklausomai nuo m reikšmės, apskaičiuojami pagal formules :

$c_k = d_{2k} + d_{2k+1}$ ir $c_{m+k} = d_{2k} - d_{2k+1}$, kai $k = 0,1,2,\ldots,m-1$.

MATLAB'o funkcijos fft ir ifft. Funkcija *fft* realizuoja greitosios Furje transformacijos, o *ifft* – atvirkštinės transformacijos algoritmą. Funkcijos *c* = *fft*(*y*) ir *y* = *ifft*(*c*) realizuoja *n*- komponenčių vektoriaus *y* tiesioginės ir atvirkštinės Furje transformacijos porą, apskaičiuojamą pagal formules:

$c_k = \sum_{j=1}^{n} y_j \omega_n^{(j-1)(k-1)}$, $k = \overline{1,n}$, $y_j = \dfrac{1}{n}\sum_{k=1}^{n} c_k \omega_n^{-(j-1)(k-1)}$, $j = \overline{1,n}$, čia $\omega_n = e^{\frac{-2\pi i}{n}}$ yra *n*-to laipsnio šaknies iš vieneto reikšmės.

Į funkciją *fft* dažniausiai kreipiamasi taip:

c = *fft*(*y*) arba *c* = *fft*(*y,n*),

čia *y* – funkcijos reikšmių vektorius arba matrica, *n* – natūralusis skaičius, o *c* – kompleksinių koeficientų c_k vektorius (jei *y* – vektorius) arba matrica (jei *y*- matrica). Jei *y* yra matrica, tai greitoji Furje transformacija atliekama su kiekvienu stulpeliu atskirai.

Nurodžius parametrą *n*, transformacija atliekama su pirmaisiais *n* vektoriaus *y* (matricos stulpelio) elementais (likę elementai atmetami). Jei *n* yra didesnis už *y* elementų skaičių, tai trūkstami elementai prilyginami nuliui.

Į atvirkštinės transformacijos funkciją kreipiamasi analogiškai. Skirtumas tas, kad vietoj funkcijos reikšmių imamos koeficientų c_k reikšmės, t.y.

y = *ifft*(*c*) arba *y* = *ifft*(*c,n*).

Vadinasi, vektoriui *y* pritaikę tiesioginę Furje transformaciją, o po to koeficientams *c* pritaikę atvirkštinę transformaciją, turėsime pradinį vektorių *y*.

Jei dalį koeficientų *c* atmestume, o po to atliktume atvirkštinę transformaciją, tai apskaičiuotas vektorius *y* skirsis nuo pradinio vektoriaus. Ši savybė naudojama, pavyzdžiui, šalinant signalų triukšmus.

Tarkime, kad turime signalo su triukšmu diskrečiasias reikšmes. Norėdami nufiltruoti triukšmą, elgsimės taip:

- atliksime signalo reikšmių tiesioginę Furje transformaciją ir apskaičiuosime kompleksinius koeficientus c_k;
- koeficientus c_k, kurių moduliai yra mažesni nei pasirinktas slenkstis keisime nuliais (kietojo slenksčio strategija) arba visų koeficientų modulius sumažinsime tuo pačiu dydžiu (slenksčiu) ir koeficientus, kurių moduliai tapo neigiami, pakeisime nuliais (minkštojo slenksčio strategija);
- atliksime atvirkštinę transformaciją su perskaičiuotais koeficientais c_k ir, jei slenkstis parinktas tinkamai, turėsime signalą be triukšmo arba triukšmo lygis bus mažas.

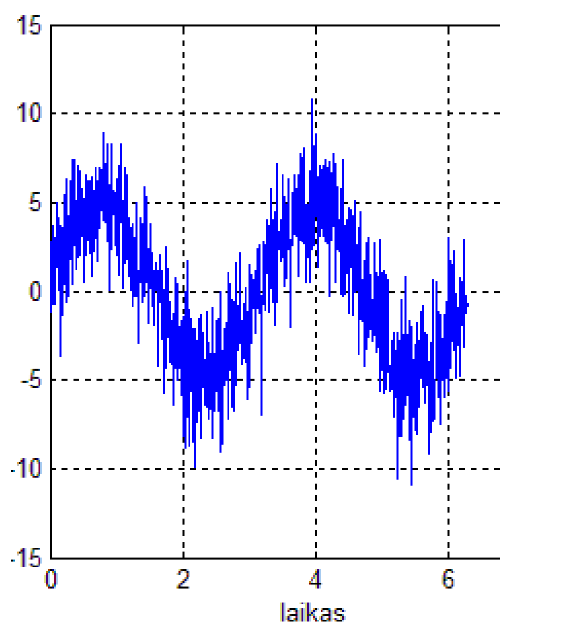
Vienas iš pagrindinių uždavinių yra kaip konkrečiu atveju apskaičiuoti tinkamą slenksčio reikšmę. Paprastai šie klausimai nagrinėjami specialioje literatūroje.

Pavyzdys. *Nufiltruokime užterštą signalą* $y = 5\sin 2t + 2 * randn(size(t))$.

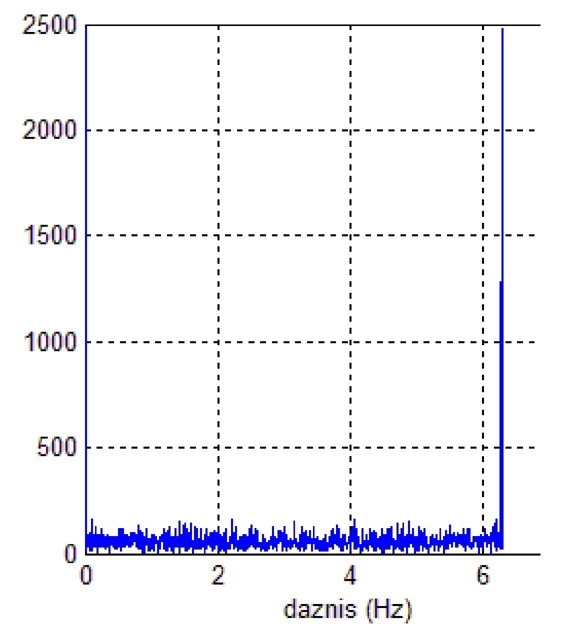
Žemiau pateikta programa, realizuojanti kieto slenksčio filtravimo strategiją.

```
t = linspace(0,2*pi,1024);
x = 5*sin(2*t);
y = x +2*randn(size(t));
plot(t,y)
grid on
title('Signalas su triukšmu')
xlabel('laikas')
Y= fft(y);
Pyy = sqrt(Y.* conj(Y));
figure(2)
plot(t,Pyy)
title('koeficientų moduliai')
xlabel('dažnis (Hz)')
ind=Pyy<180;% tinkamas slenksčio dydis
Pyy(ind)=0;
figure(3)
plot(t,Pyy)
title('nufiltruotų koeficientų moduliai')
xlabel('dažnis (Hz)')
Y(ind)=0;
yt=ifft(Y);
figure(4)
plot(t,yt)
title('Signalas be triukšmo')
xlabel('laikas')
grid on
```

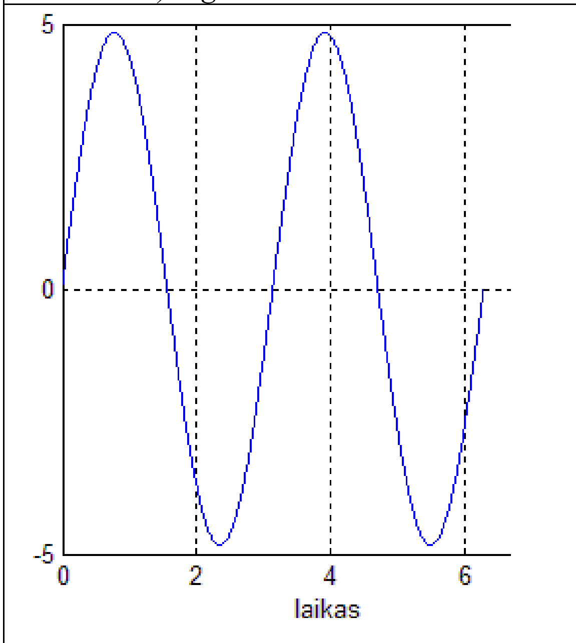
6.3 paveiksle pavaizduoti filtravimo rezultatai: a) signalas su triukšmu; b) signalo su triukšmu Furje koeficientų moduliai; c) nufiltruotas signalas, kai slenksčio dydis parinktas tinkamai; d) nufiltruotas signalas, kai parinktas slenksčio dydis yra per mažas.



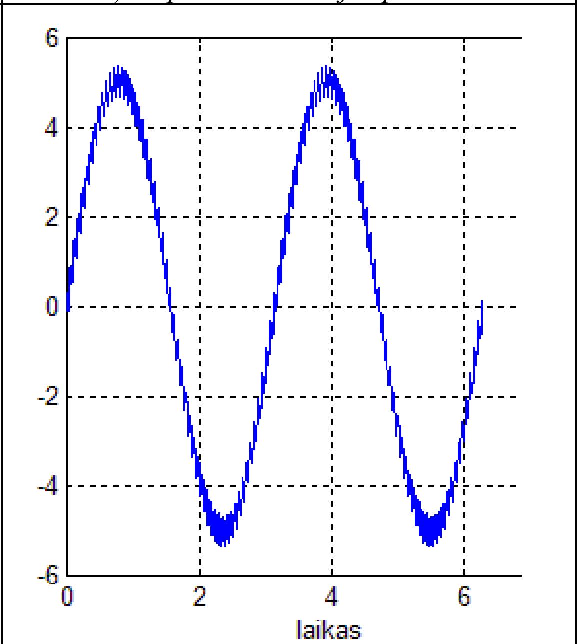
a) Signalas su triukšmu



b)Amplitudinis Furje spektras



c) Nufiltruotas signalas, kai slenkstis parinktas tinkamai



d) Nufiltruotas signalas, kai parinktas slenkstis yra per mažas

6.3 pav. Signalo $y = 5\sin 2t + 2 * randn(size(t))$ *filtracija*