



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
MULTIMEDIJOS INŽINERIJOS KATEDRA

Dmitrij Kultašev

**Ultragarsinės vizualizacijos sistemos modelis ir jo
greitaveikos optimizavimas**

Magistro darbas

Darbo vadovas

doc. L. Mažeika

Kaunas, 2008



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
MULTIMEDIJOS INŽINERIJOS KATEDRA

Dmitrij Kultašev

Ultragarsinės vizualizacijos sistemos modelis ir jo greitaveikos
optimizavimas.

Magistro darbas

Recenzentas

doc. L. Mažeika

2008-05-

Vadovas

doc. L. Mažeika

2008-05-

Atliko

IFN 6/1 gr. stud.

Dmitrij Kultašev

2008-05-26

Kaunas, 2008

Turinys

Įvadas	8
1. Ultragarsinė vizualizacija ir jos modeliavimo būdai	10
1.1. Ultragarsinė vizualizacija	11
1.2. Hiugenso principas	14
2. Naudojamo modelio struktūra ir algoritmas	16
3. Modelio naudojamų kompiuterinių resursų tyrimas	21
3.1. Užimtos atminties įvertinimas	21
3.2. Programos modelio greičio tyrimas	23
4. Modelio programos optimizavimas	29
4.1. Atminties sutvarkymas	29
4.2. Modeliavimo ribų praplėtimas	31
5. Atspindžio funkcijos tikslumo priklausomybės nuo diskretizavimo žingsnio tyrimas	38
6. Išvados	46
7. Literatūra	47

Įvadas

Naujuose atominiuose MYRRHA tipo reaktoriuose branduolys yra aušinamas skystuoju metalu – švino/bismuto lydiniu [1]. Dėl reaktoriaus saugumo jis turi būti reguliariai tikrinamas. Tačiau skysto metalo terpė yra neskaidri, elektriškai laidi, chemiškai agresyvi ir nėra kitų fizikinių priemonių išskurus ultragarsą, kurios įgalintų „apžvelgti“ reaktoriaus vidų. Šiems tikslams naudojama sistema turi veikti labai sudėtingose sąlygose: aukšta metalo temperatūra (160 – 400 °C), stiprus γ ir neutronų radiacinis spinduliavimas (iki 30kGy/h), cheminis agresyvumas. Šitos sąlygos ženkliai apriboja galimas ultragarsinės vizualizacijos sistemos architektūras bei naudojamas medžiagas. Iš kitos pusės, sistema turi būti pakankamai „paprasčia“, kad galėtų būti įgyvendinta aukščiau aprašytose sunkiose aplinkose.

Projektuojant tokias sistemas reikia patikrinti labai daug įvairių realizavimo galimybių ir parametrų. Atlikti tai eksperimentiniu būdu yra sudėtinga ir brangu. Vienas iš šios prieštaraujančios problemos sprendimo būdų yra sukurti kompiuterinį modelį, kuris imituotų ultragarso sklidimą skystajame metalo ir apskaičiuoti atspindėtą nuo komplikuočių reaktoriaus komponentų paviršiaus signalą.

Belgijos branduolinių tyrimų centro SCK/CEK ir Ultragarso instituto bendradarbiavimo sukurtas MYRRHA tipo branduolinio reaktoriaus akustinis-kompiuterinis modelis, leidžiantis modeliuoti ultragarso bangų sklidimą bei atspindžius reaktoriaus viduje. Be to, modelis leidžia imituoti įvairius ultragarsinius vizualizavimo būdus ir įvertinti jų tinkamumą reaktoriaus kontrolei.

Kadangi vidinė reaktoriaus geometrinė forma yra sudėtinga, visi paviršiai aprašomi elementarių trikampių aibe. Atspindžiams nuo elementarių trikampių skaičiuoti sukurti skaitmeniniai metodai, pagrįsti Hiuigenso principu. Dėl sudėtingų matematinių skaičiavimų bei labai didelio duomenų kiekio, šis modelis veikia sąlyginai lėtai ir prie didelės detalizacijos, skaičiavimai gali užtrukti net kelias dienas.

Šio darbo tikslas yra ištirti veikiančio modelio programą, nustatyti reikalaujamų kompiuterinių resursų priklausomybę nuo įvairių parametrų ir pasiūlyti algoritminius ar programinius pakeitimus kurie optimizuoto modelio veikimą kompiuterinių resursų požiūriu.

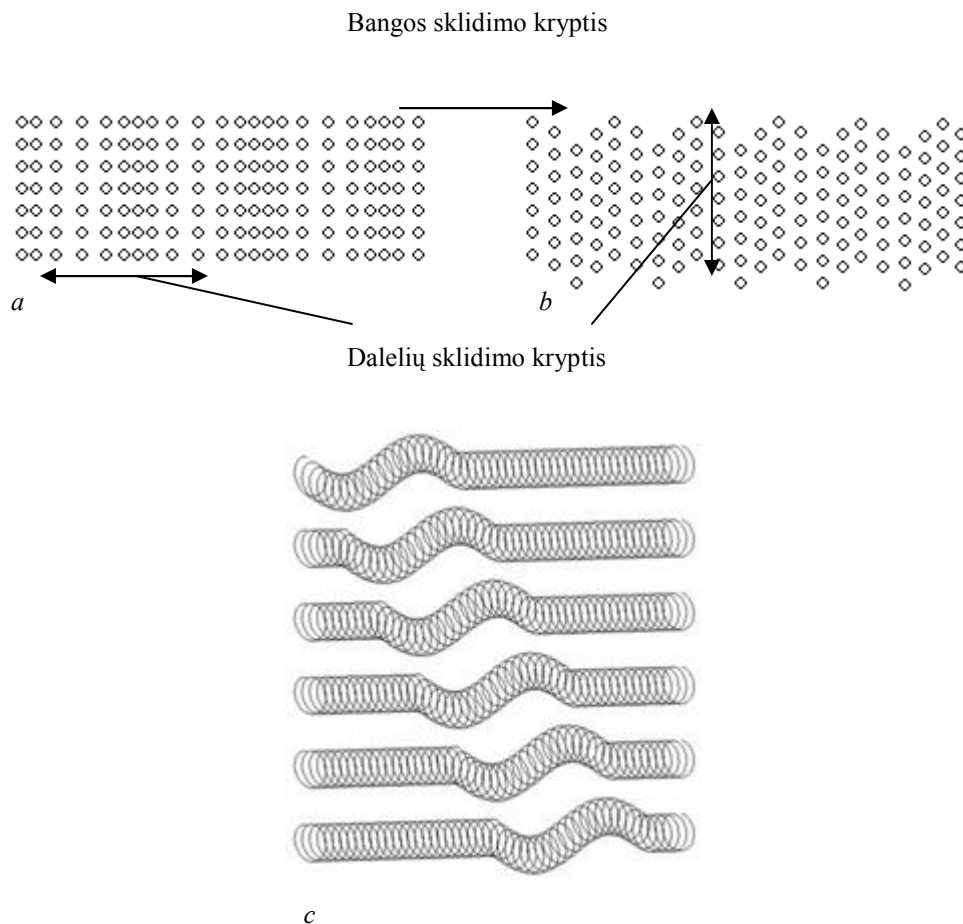
Šio tikslo pasiekimui reik spręsti tokius uždavinius:

- Ištirti naudojamų resursų priklausomybę nuo įvairių parametrų ir nustatyti neoptimalias vietas;

- Pasiūlyti programinius budus, kurie leistų tam pačiam algoritmui funkcionuoti greičiau;
- Nustatyti principines priklausomybes tarp skaidymo žingsnio ir modeliavimo tikslumo.

1. Ultragarsinė vizualizacija ir jos modeliavimo būdai

Norint susipažinti su ultragarsine vizualizacija ir jos modeliavimo būdais, būtina turėti supratimą kas gi yra ultragarsas. Ultragarsas – garso slėgio bangos su dažniu, aukštesniu negu viršutinė žmogaus klausos riba. Kiekvieno žmogaus klausa yra skirtinga, tačiau viršutinė jauno ir sveiko žmogaus jos riba yra apie 16-20kHz [2]. Skystyje ar dujose sklindančios ultragarso bangos vadinamos slėgio bangomis. Kietomis medžiagomis gali sklisti keletą tipų ultragarso bangos. Pagrindinės yra skersinės ir išilginės bangos. Išilginės bangos – tai tokios bangos, kurioms sklindant medžiagos dalelės juda išilgai bangos sklidimo krypties (1 pav., a). Skersinėse bangose medžiagos dalelės juda statmenai bangos sklidimo kryptčiai (1 pav., b). Teoriškai šios bangos sklinda tik ribų neturinčiose aplinkose. Kietos medžiagos riboje (jos paviršiumi) sklinda vadinamosios paviršinės bangos. Jų yra keletą tipų. Plačiausiai žinomos yra Reilėjaus paviršinės bangos [3]. Jose dalelės juda elipsine trajektorija. Šių trijų tipų bangos plačiausiai naudojamos ultragarsiniuose neardomuosiuose bandymuose.

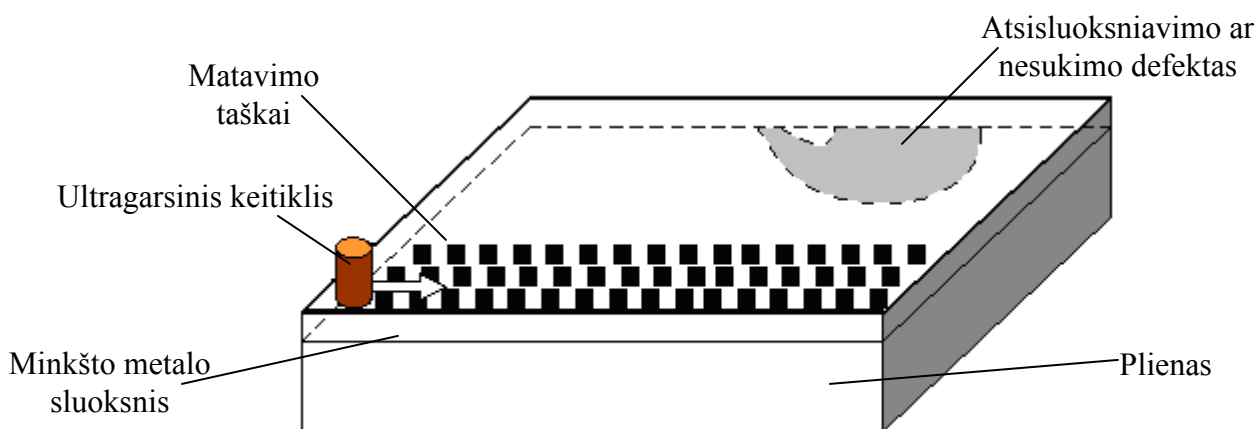


1 pav. Dalelių judėjimas išilginėse (a), skersinėse (b) ir Reilėjaus (c) bangose

1.1. Ultragarsinė vizualizacija

Ultragarsinė vizualizacija, kaip sudėtingesnis neardomųjų bandymų metodas, įgalina ne tik stebėti pavienius ultragarso signalus, bet ir sukurti tam tikrą ultragarsinį objekto vidinės struktūros vaizdą. Kaip matysime vėliau, tai nėra įprastas optinis vaizdas, bet tam tikras vaizdas, suformuotas iš ultragarso signalų, todėl jį analizuoti ir suvokti nėra paprasta.

Ultragarsinę vizualizaciją paprasčiausia yra paaiškinti konkretaus objekto pavyzdžiu. Tarkime, yra tiriama tam tikra plokštė, susidedanti iš dviejų metalo sluoksnių, kurios viduje yra atsisluoksniavimo defektas (2 pav.). Tokio objekto pavyzdys gali būti slydimo guolis. Šiuo atveju pagrindinis metalas yra plienas, o paviršinis – minkšto lydinio (pvz., babilto) sluoksnis.

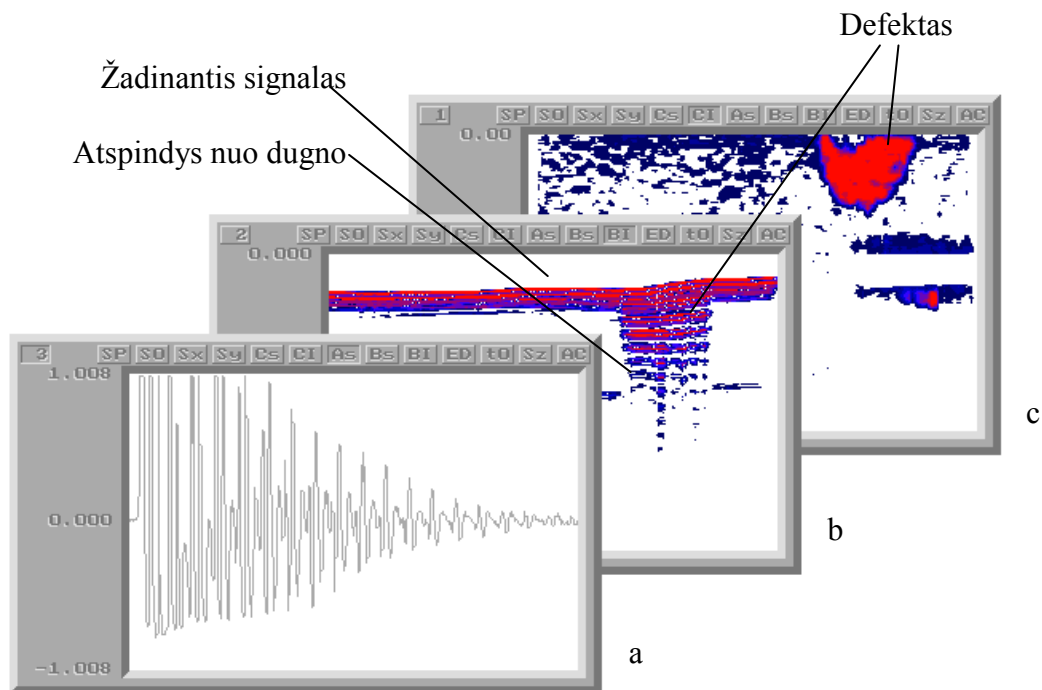


2 pav. Analizuojamo objekto skenavimas vizualizacijos duomenų gavimui.

Objekto skenavimo metu, keitiklis yra fiziškai stumdomas ir yra matuojamas signalas kiekviename taške. Jei metalų sukibimas yra geras, pagrindinio ir minkšto metalų akustinių impedansų skirtumas nėra didelis ir atspindžio koeficientas yra mažas. Šiuo atveju gaunamas tik tai signalas, atsispindėjęs nuo dugno. Tačiau, jei atsiranda nesukibimas, tai toje vietoje gaunamas labai stiprus atspindys nuo dviejų metalų ribos. Kadangi metalai signalą slopina mažai, jis „skambės“ pirmame sluoksnyje, atspindėdamas nuo viršutinio paviršiaus ir metalų nesukibimo vietos. Paprastas signalo vaizdas ultragarsinėje vizualizacijoje vadinamas *A* tipo vaizdu (angl. *A scan image*) (3 pav., a) [4].

Vadinamasis *B* tipo vaizdas suformuojamas, jei vienos kurios nors tiesės kryptimi signalai ir jų amplitudės koduojami spalvomis (3 pav., b). Tokio vaizdo horizontalioji ašis atitinka matavimo poziciją ilgio vienetais, o vertikalioji – signalo laiko ašį (arba gylį į objektą). Fiziškai jį galima būtų vadinti objekto pjūvio vaizdu. Jei surandama maksimali kiekvieno išmatuoto signalo amplitudė ir tik

ji koduojama spalvomis, suformuojamas *C* tipo vaizdas (3 pav., c). *C* tipo vaizdas tai lyg ir objekto vaizdas iš viršaus, parodantis jo struktūros nevientisumą. Kadangi atsispindėjusio signalo amplitudė priklauso nuo defekto, *C* tipo vaizde galima ne tik pastebėti defektą, bet ir nustatyti jo dydį bei formą.

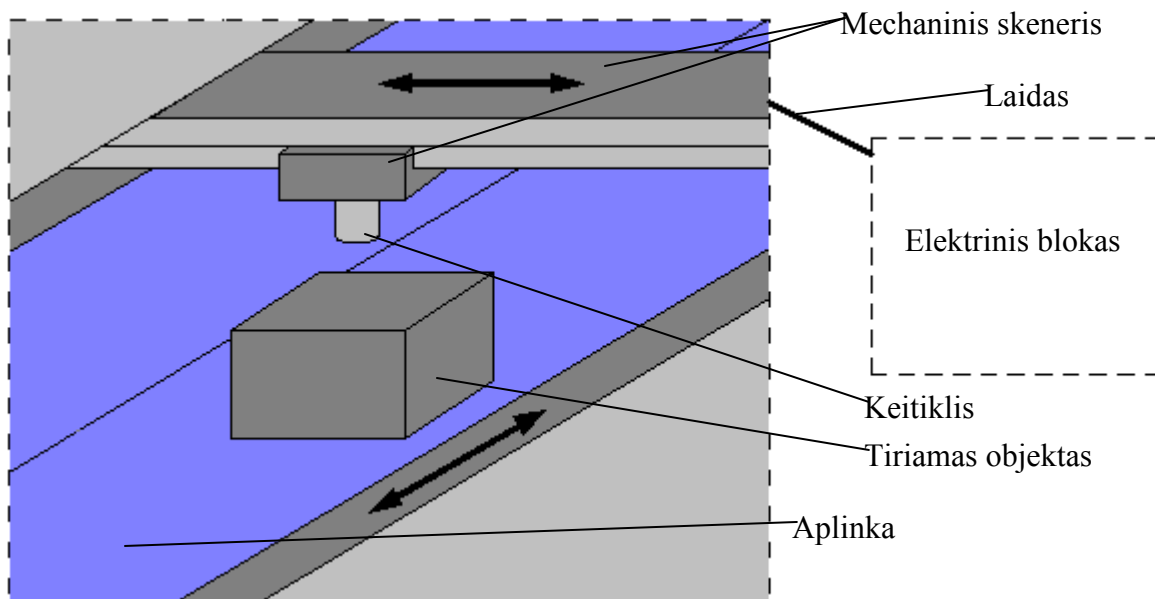


3 pav. Ultragarsinių vaizdų pavyzdžiai: *a* – *A* tipo vaizdas; *b* – *B* tipo vaizdas; *c* – *C* tipo vaizdas

Čia pateiktas labai supaprastintas ultragarsinės vizualizacijos atvejis. Realių objektų vaizdai gaunami gerokai sudėtingesni ir jų interpretacija ne visada būna vienaraikšmiška. Tačiau daugeliu atveju tik ji leidžia aptikti defektus, nustatyti jų vietą ir dydžius. Kadangi ultragarsinei vizualizacijai reikia sudėtingesnės įrangos, kvalifikuotesnio personalo, ji, be abejo, yra brangesnė ir naudojama tik atsakingų objektų neardomiesiems tyrimams. Plačiausiai ji taikoma atominėje energetikoje ir šiuolaikinėje aviacijoje.

Nors ultragarso taikymo galimybės yra labai plačios, kiekvienam jų reikalinga skirtingas konfigūravimas, reikalaujantys aukšto ultragarso savybių išmanymą bei aplinkai, kurioje randasi analizuojamas objektas, pritaikyta technika.

Pagrindinė akustinio modelio struktūra, naudojama skirtingiems vizualizacijos principams realizuoti yra parodyta 4 pav .

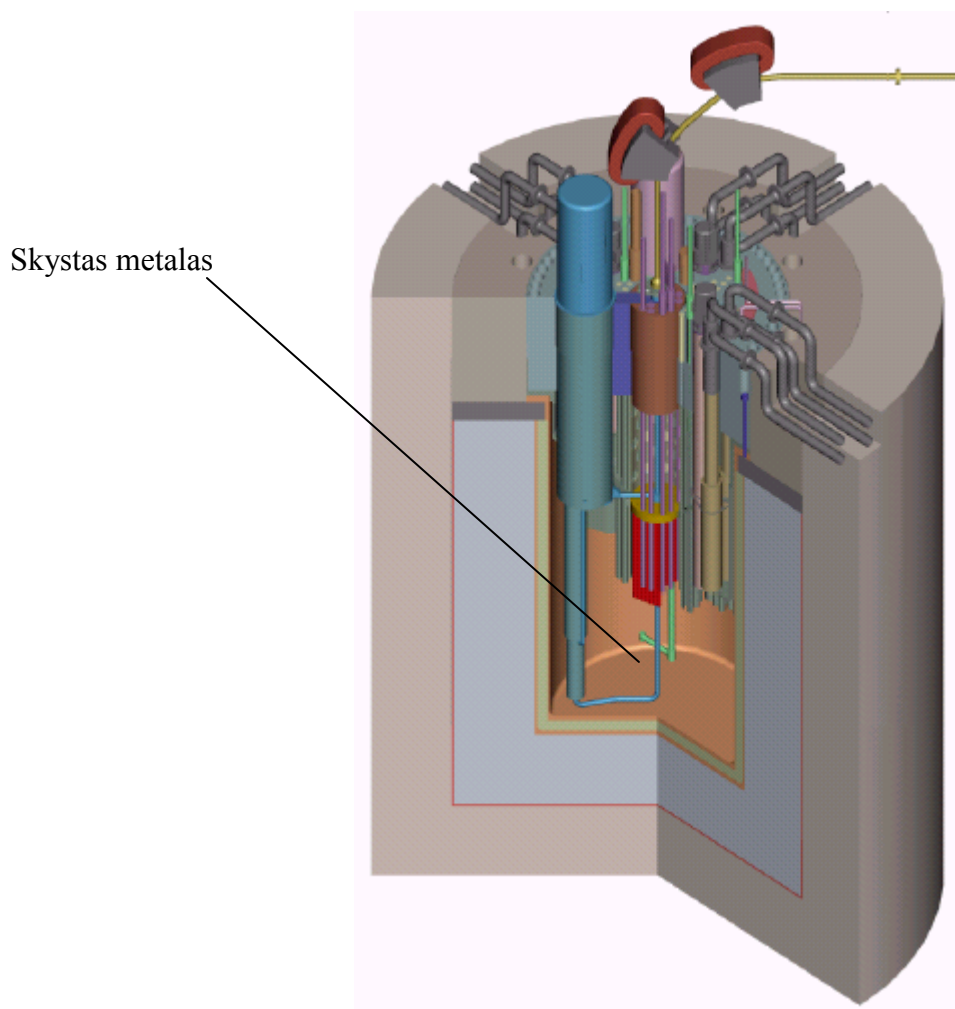


4 pav. Supaprastinta ultragarsinių matavimų modelių schema

Mechaninis skeneris stumdo keitiklį, kuris siunčia signalus ir gauna atspindžius. Keitiklis yra pajungtas prie elektrinio valdymo bloko (dažniausiai kompiuteris). Analizuojamas objektas randasi kažkokioje tai aplinkoje (oras, vanduo, skystas kristalas ir t.t.) [5].

Visa ultragarsinių matavimų problema yra nustatyti kaip sklinda ultragarsinės bangos aplinkoje, kaip jos atspindi nuo objekto ir koki signalą priima. Sudėtingų objektų atvejuose gali prireikti didesnio kiekio keitiklių, analizuojamas objektas gali būti sunkiai prieinamoje vietoje ir/ar sudėtingoje aplinkoje. Kaip gyvą pavyzdį galima būtų pateikti su darbu susijusį atvejį. Belgijos branduolinių tyrimų centro SCK/CEK (angl. *Advanced Nuclear Systems Institute of SCK•CEN*) buvo sukurtas MYRRHA tipo branduolinis reaktorius [6]. Reaktoriaus vidinis interjeras yra labai sudėtingas ir jo viduje vyksta sudėtingos atominės reakcijos. Be to jis dar yra šaldomas skystuoju metalu (žr. 5 pav.) [7]

Įvertinus visus šiuos faktorius, jų sudėtingumą bei brangumą, Belgijos branduolinių tyrimų centro SCK/CEK ir KTU Ultragarso instituto bendradarbiavimo sukurtas MYRRHA tipo branduolinio reaktoriaus akustinis-kompiuterinis modelis, leidžiantis modeliuoti ultragarso bangų sklidimą bei atspindžius reaktoriaus viduje. Be to, modelis leidžia imituoti įvairius ultragarsinius vizualizavimo būdus ir įvertinti jų tinkamumą reaktoriaus kontrolei [8].



5 pav. Vidinis atominio MYRRHA tipo reaktoriaus interjeras.

1.2. Hiugenso principas

Bet kokio ultragarsinio sklidimo modeliavimo pagrindas yra Hiugenso (*Huygens*) principas, kuris teigia, kad kiekvienas bangos fronto taškas yra antrinių sferinių bangų šaltinis, o bangos frontas bet kuriuo kitu laiko momentu yra šių elementariųjų bangų gaubtinė [9].

Pagal tokią Hiugenso principo formuluotę galima spręsti užduotis, susijusias su bangos fronto sklidimo kryptimi, tačiau ji nenusako įvairiomis kryptimis sklindančių bangų intensyvumo. Be to, Hiugenso principas taikomas bet kurio ilgio bangoms.

O.Frenelis (*A.Fresnel*) papildė Hiugenso principą teiginiu, kad antrinės bangos yra koherentinės ir interferuoja. Formaliai nusakyta antrinių bangų gaubtinė įgauna fizikinę prasmę – tai paviršius, kuriame dėl antrinių bangų tarpusavio interferencijos atstojamosios bangos intensyvumas yra didžiausias [10].

Papildytas Hiuigenso principas vadinamas Hiuigenso ir Frenelio principu. Jis yra pagrindinis banginės optikos principas, kuris padeda nusakyti atstojamosios bangos intensyvumą įvairiomis kryptimis, t. y. spręsti šviesos difrakcijos užduotis.

Apibendrinant šį principą gauname, kad norint apskaičiuoti priimtą signalą, mums reikia gauti 3 dalykus:

1. Kas buvo išspinduliuota;
2. Kas buvo atspindėta;
3. Kas buvo priimta.

Praktiškai, reikia visą paviršių suskaidyti į daug elementarių taškiukų ir bendras išspinduliuotas laukas yra šių visų taškinių šaltinių suma, kurių kiekvienas taškas iš tikrųjų yra sferinių bangų šaltinis. Atspindėto signalo atveju, paviršius lygiai taip pat yra skaidomas į mažus taškus. Taip pat yra skaidomas ir keitiklio paviršius [11].

Problema yra tame, kad pagal Hiuigenso principą, norint analizuoti diskretinius signalus, jie turi būti skaidomi priklausomai nuo bangos ilgio. Principinė teorinė riba yra 2 taškai per bangos ilgį. Viso šito pasiekoje skaidymo smulkumas ir taškų skaičius priklauso nuo tyrimo bangos dažnio (arba bangos ilgio).

$$c = f * \lambda ; (1)$$

f – dažnis, λ - bangos ilgis.

Jei tyrimai yra atliekami žemo dažnio srityje, gali ir nekilt didesnių problemų, tačiau jei atliekami tyrimai aukštesnio dažnio srityse, pvz. 3-5Mhz, kas yra labai būdinga neardantiesiems bandymams, bangos ilgis daugelyje skystų aplinkų yra 0.3-0.5 mm.

Praktikoje yra naudojama visokių supaprastintų variantų, pvz. keitiklio laukas skaičiuojamas naudojamas panaudojant skirtingas analitines išraiškas. Vienas iš tokių variantų ir yra būtent yra naudojamas naudojamame modelyje.

Pagrindiniai šio modelio ypatumai yra tokie:

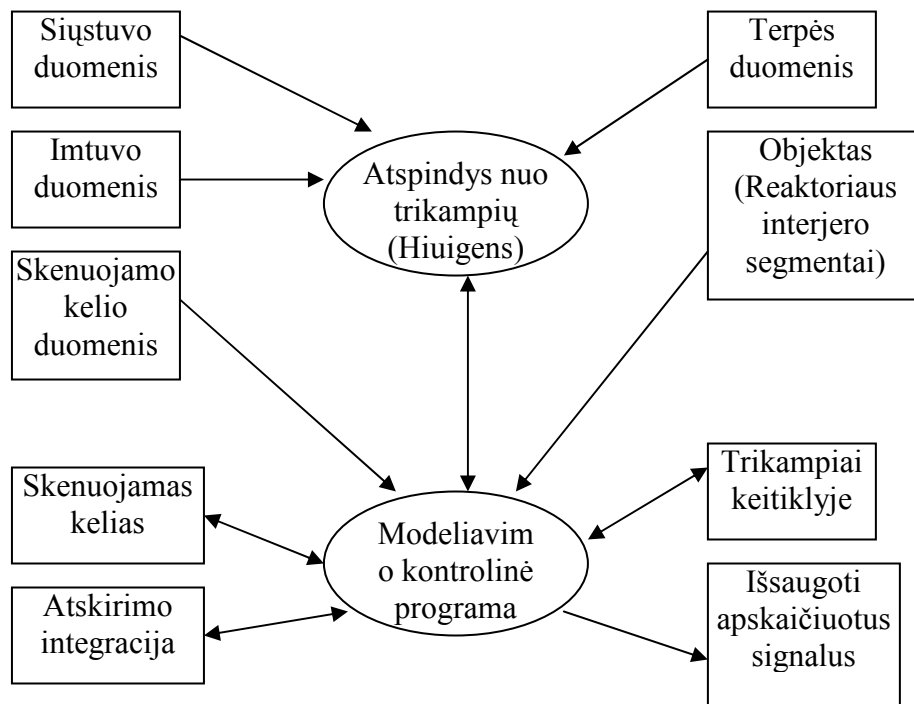
- 1) Keitiklio spinduliuojamo lauko modelis yra supaprastintas;
- 2) Skaičiavimas vyksta taikant Hiuigenso principą atspindėtam nuo sudėtingos geometrinės formos objekto, signalui [12].

Šis modelis yra universalus, tačiau veikia lėtai. Realizuotas jis *MATLAB* aplinkoje.

2. Naudojamo modelio struktūra ir algoritmas

Kaip buvo minėta anksčiau, kad yra naudojamas supaprastintas keitiklio modelis, pagal kurį išspinduliuotas laukas yra aprašomas tikrai kryptingumo diagrama, iš kitos pusės atspindintis paviršius yra skaidomas į daug trikampių. Taip yra pasirinkta dėl to, kad sudėtingos formos geometriniai objektai, tokie kaip pvz. reaktoriaus vidus, dažniausiai yra projektuojami panaudojant automatizuotas projektavimo sistemas, tokias kaip pvz. *Autodesk AutoCAD*. Viena iš *AutoCAD*'o saugojimo formų yra taip vadinami .stl bylos. Formuojant šią bylą, objekto paviršiai yra skaidomi trikampaiais ir šių trikampių duomenis – viršunes (paviršiaus orientavimo normales) yra surašomi į tekstinę bylą. Modelis ir naudoja tokių būdų gautus trikampus. Kiekvienas iš šių trikampių yra skaidomas į daug elementarių taškiukų ir pagal Hiugenso principą yra skaičiuojami atspindžiai.

Pagrindinė akustinio modelio struktūra, naudojama skirtingiems vizualizacijos principams realizuoti yra parodyta 6 pav. Ji susideda iš ultragarsinių keitiklių duomenų bazės, terpės parametrų, objekto duomenų ir skenuojamo kelio parametrai, reikalingų elementarių segmentų surinkimo programa, atspindžio nuo trikampio atspindžio integracijos programa, programa, kuri saugo gautus signalus į duomenų bylą ir simuliacijos kontrolės programą [13].



6 pav. Pagrindinė akustinio modelio struktūra.

Bendroju atveju vykdymo algoritmo modeliavimas gali būti išreikštas šiais žingsniais [14]:

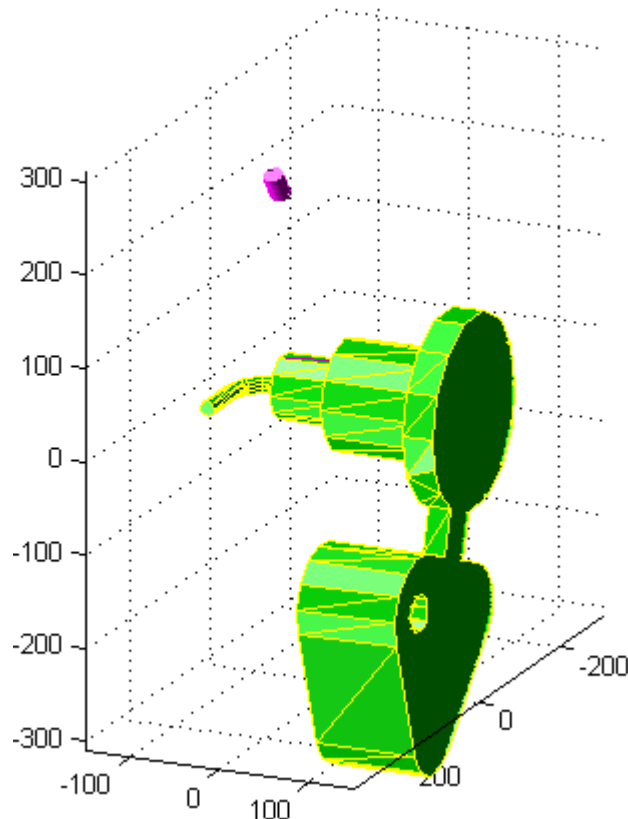
1. Nustatomi modelio pirminiai parametrai.

2. Einamoji skenerio pozicija yra parinkta ir atitinkamos siųstuvo ir imtuvo pozicijos/orientacijos yra nustatytos.
3. Pasirenkami trikampiai, kurie patenka į siųstuvo ir imtuvo matomumo zoną;
4. Apskaičiuojamas kiekvieno pasirinkto trikampio atspindėto signalo banga;
5. Apskaičiuota banga yra integruojama (pridedama viena prie kitos), bendros atspindėto signalo bangos gavimui;
6. Sudėtingų ultragarsinių sistemų atveju (daugiau imtuvų ir siųstuvų) 3-5 žingsniai yra pakartojami kiekvienai imtuvo ir siųstuvo porai;
7. Sekanti skenerio pozicija yra pasirinkta ir 3-6 žingsniai yra pakartojami.

Tuo būdu modeliavimo rezultatas yra signalų, panašių į tuos, kurie yra gaunami pasinaudojant fiziniaus eksperimentais, rinkinys (signalai skirtingose pozicijose). Šie signalai gali būti panaudojami kuriant tipinius A, B, C – vaizdus pasinaudojant skirtingais algoritmais.

Atspindėtas nuo trikampio signalas yra apskaičiuotas pasinaudojant šiais žingsniais [15]:

1. Trikampio, kuris yra pilnai siųstuvo ir imtuvo susikirtimo matomumo zonoje, zona yra rasta ((x_k, y_k, z_k) , $k = 1 \div K$, K – bendras segmentų skaičius trikampyje) (žr. 7 pav.)



7 pav. Objekto trikampiai, keitiklio matomumo zonoje.

2. Ši trikampio zona yra padalinta į elementarius segmentus;

3. Apskaičiuojamas atstumas $\{L_k\}$ tarp šių elementarių segmentų ir keitiklio centro;
4. Apskaičiuojama kiekvieno elementaraus segmento atspindėto signalo amplitudė $\{a_k\}$ ir vėlavimas $\{t_k\}$. Skaičiuojant yra įvertinama keitiklio kryptingumo diagrama ir ultragarso greitis aplinkoje c_m ;
5. Naudojant šiuos parametrus, suskaičiuojama objekto atspindžio funkcija
$$h_0(t) = F_{\Sigma}(\{a_k\}, \{t_k\}); \quad (2)$$
6. Atspindėtas signalas gaunamas naudojant atspindžio funkcijos ir atraminio signalo konvoliucija
$$U_m(t) = h_0(t) \otimes u_A(t); \quad (3)$$

Čia pateiktas labai supaprastintas signalo, atspindėto, nuo trikampio apskaičiavimo algoritmas. Iš tikro vieno trikampio uždengimo kitu ar dvigubo atspindžio (žr. 8 pav.) signalų apskaičiavimo algoritmas yra faktiškai realizuojamas 3 stambiais žingsniais:

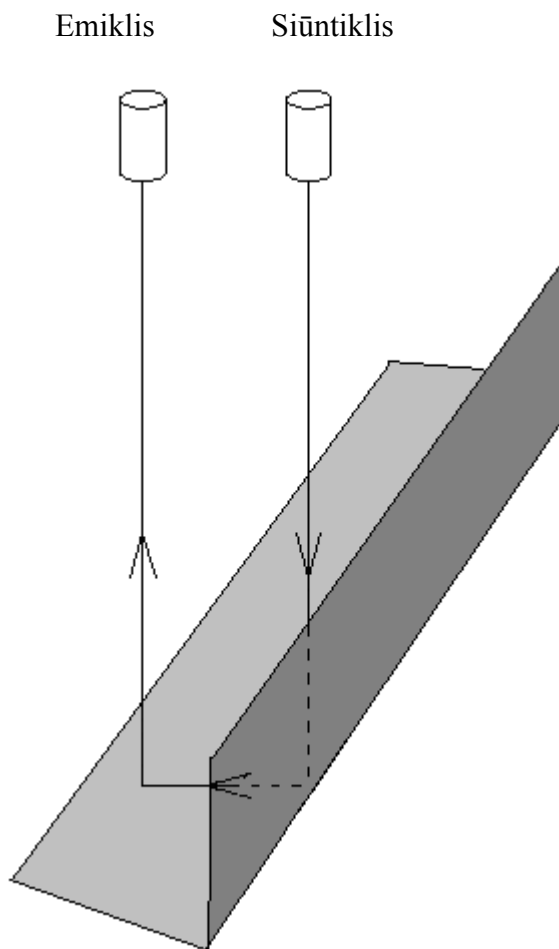
Pirmas žingsnis apskaičiuoja objekto apspinduliuojančio lauko parametrus ir susideda iš sekančių etapų:

- Visas objektas suskaidytas į trikampius. Išrenka visus trikampius, kurie yra kryptingumo diagramoje;
- Surūšiuoja juos taip, kad jie būtų surikiuoti nuo toliausio, artėjant link kryptimi;
- Atliekamas visų trikampių projektavimas į virtualią plokštumą, kuri yra statmena spinduliui ir randasi tolimiausio trikampio atstume. Laikiname masyve, atitinkančiame virtualia plokštuma, yra suskaičiuojama visiems tiems taškams apspinduliuavimo amplitudės ir laikai nuo keitiklio iki to taško trikampyje. Kadangi trikampiai yra nagrinėjami nuo toliausio artėjimo kryptimi, tai aukštesnių trikampių duomenis visada perdengs žemesnius esančius.

Antras žingsnis skaičiuoja priimamo lauko parametrus ir susideda iš tokių žingsnių:

- Apskaičiuojamas atstumas iki emiklio;
- Suskaičiuojama amplitudės priėmimo režime;
- Patikrina ar elementarus taškai vieni neuždengia kitus (paliekami tikrai arčiausi).

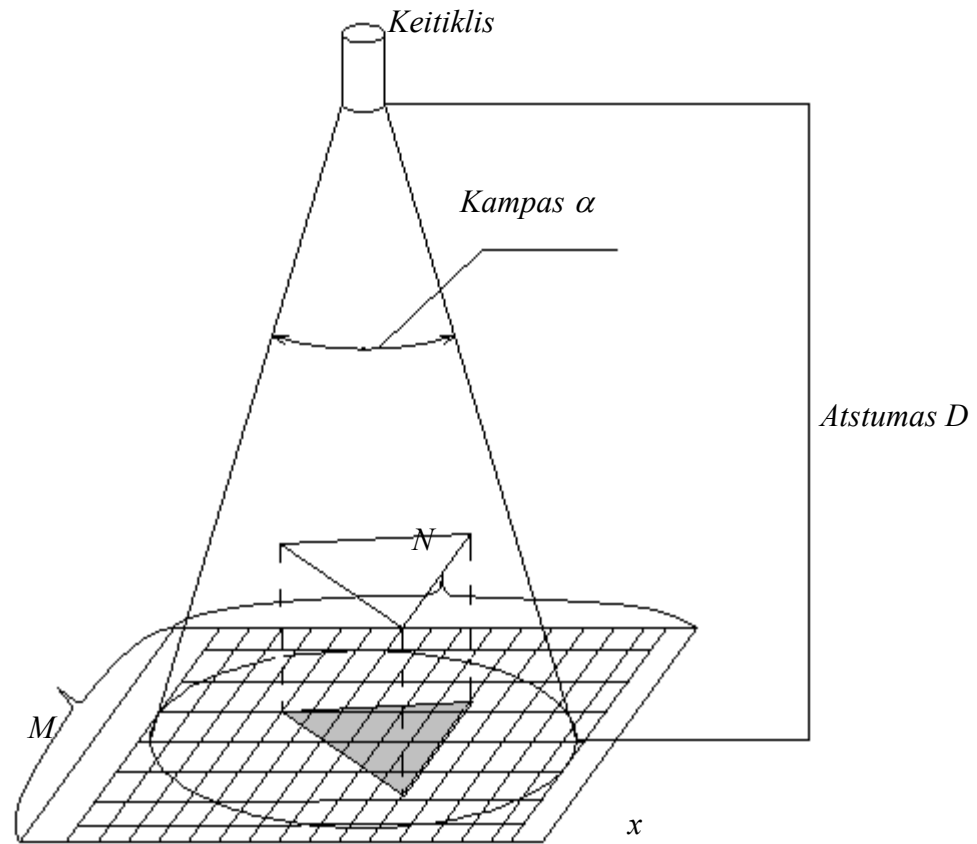
Trečias žingsnis yra skirtas išimtinai dvigubo atspindžio skaičiavimui (žr. 8 pav.).



8 pav. Kampinių atspindžių paaiškinimas

Šio atspindžio apskaičiavimui, atspindinčio objekto paviršius yra skaidomas į didelį skaičių elementarių atspindinančių plotelių ar taškų. Nuo šių taškų skaičiaus faktiškai ir priklauso naudojamos atminties dydis ir skaičiavimo greitis. Šių taškų skaičius priklauso nuo diskretizavimo erdvėje koeficiento, t.y. kiek yra imama taškų į bangos ilgį. Pagal teoriją K_{atk} turėtų būti ne mažesnis už 2 [16], tačiau kaip rodo praktika, šio koeficiento tiksliais rezultatams gauti neužtenka. Realiai K_{atk} turi būti lygus 15, 30 kartais net 40. Kadangi atspindis yra skaičiuojamas nuo paviršiaus, padidinus šį koeficientą skaičiavimo taškų skaičius didėja kvadratine progresija. To pasiekoje taip pat padidėja ir visų masyų ar matricių, naudojamų tarpiniams ir galutiniams skaičiavimams, dydžiai. Naudojamų resursų apimtis taip pat priklauso ir nuo analizuojamų trikampių skaičiaus bei figūrų sudėtingumo.

Ši algoritmą paprasčiau suprasti pažvelgus į 9 pav.



9 pav. Keitiklio pozicijos grafinis vaizdavimas

α - kampas, kuris apibrėžia keitiklio matomumo zoną. Kuo tas kampas yra didesnis, tuo keitiklio zona – platesnė. Ta zona taip pat priklauso ir nuo atstumo tarp keitiklio ir duotojo objekto. x – zonos plotelis, kurio ilgis ir plotis yra lygus bangos ilgiui. Kaip buvo minėta, kad kiekvienas plotelis (bangos ilgis) turi būti dalinamas na mažiau kaip į 2 dalis, $M \times N$ – bendras šių taškų skaičius.

Sekančio skyriaus paskirtis yra nustatyti šių priklausomybių dėsningumus.

Išvados

Siekiant optimizuoti šio modelio veikimą, reikia atlikti sekančius uždavinius:

- Ištirti naudojamų resursų priklausomybę nuo įvairių parametrų ir nustatyti neoptimalias vietas;
- Pasiūlyti programinius būdus, kurie leistų tam pačiam algoritmui funkcionuoti greičiau;
- Nustatyti principines priklausomybes tarp skaidymo žingsnio ir modeliavimo tikslumo.

3. Modelio naudojamų kompiuterinių resursų tyrimas.

Šio skyriaus tikslas yra ištirti kaip priklauso modelio programos naudojami resursai (atmintis, skaičiavimo greitis) nuo pagrindinio modelio parametru. Resursų analizė bus atliekama dviem etapais. Pirmiausia bus nustatyta kiek ir kokie masyvai užima atminties, kaip jų dydis priklauso nuo diskretizavimo žingsnio ir kaip užimamas bendras atminties skaičius kinta modeliavimo metu. Antro etapo metu bus atliekama skaičiavimo laiko analizė nustatant jo priklausomybę nuo diskretizavimo taškų skaičiaus ir kitų parametru.

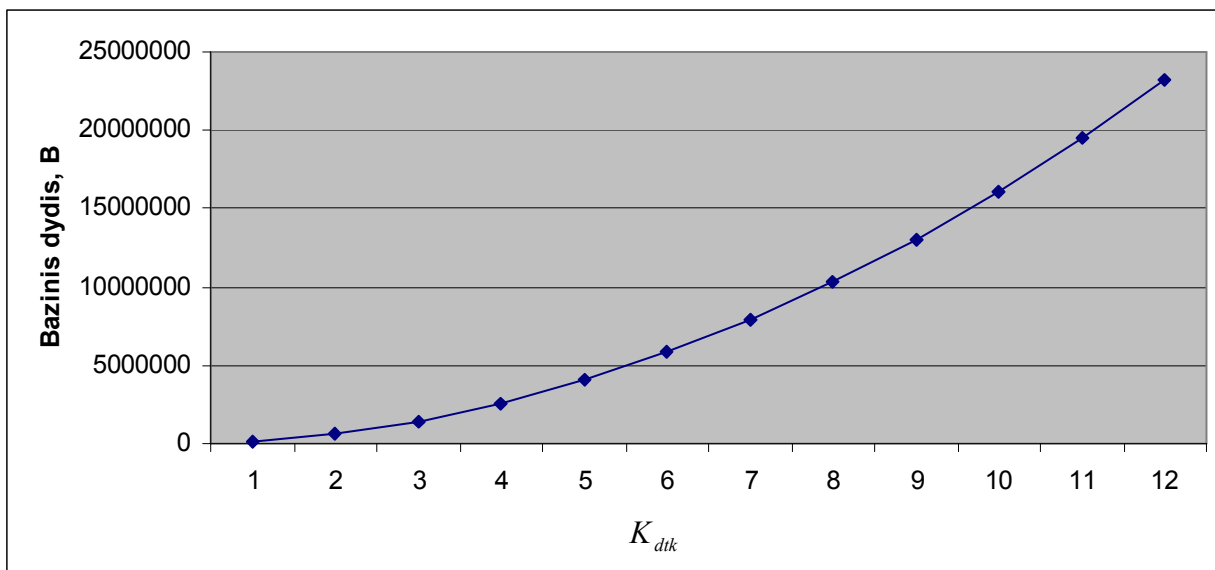
Kadangi operatyviosios atminties atžvilgiu masyvai ir matricos yra faktiškai tas pats (10x10 elementų matrica užima tiek pat atminties kaip ir 100 elementų masyvas), tai tolimesnėje eigoje jie bus vadinami bendrai masyvais.

3.1. Užimtos atminties įvertinimas

Bendrai ultragarsinės vizualizacijos sistemos moduliai panaudoja daug įvairiausių dydžių masyvų, tačiau dalis jų yra nedidelio dydžio ir atminties jie praktiškai nenaudoja. Pagrindinę atmintį išnaudoja 51 baziniai *double* tipo masyvai, kurių kiekvienas elementas užima 8 baitus [17] operatyviosios atminties. Bazinių masyvų bei jų naudojamos atminties dydžiai yra parodyti 1 lentelėje bei jų priklausomybė nuo diskretizavimo koeficiento K_{dtk} parodyta 10 pav.

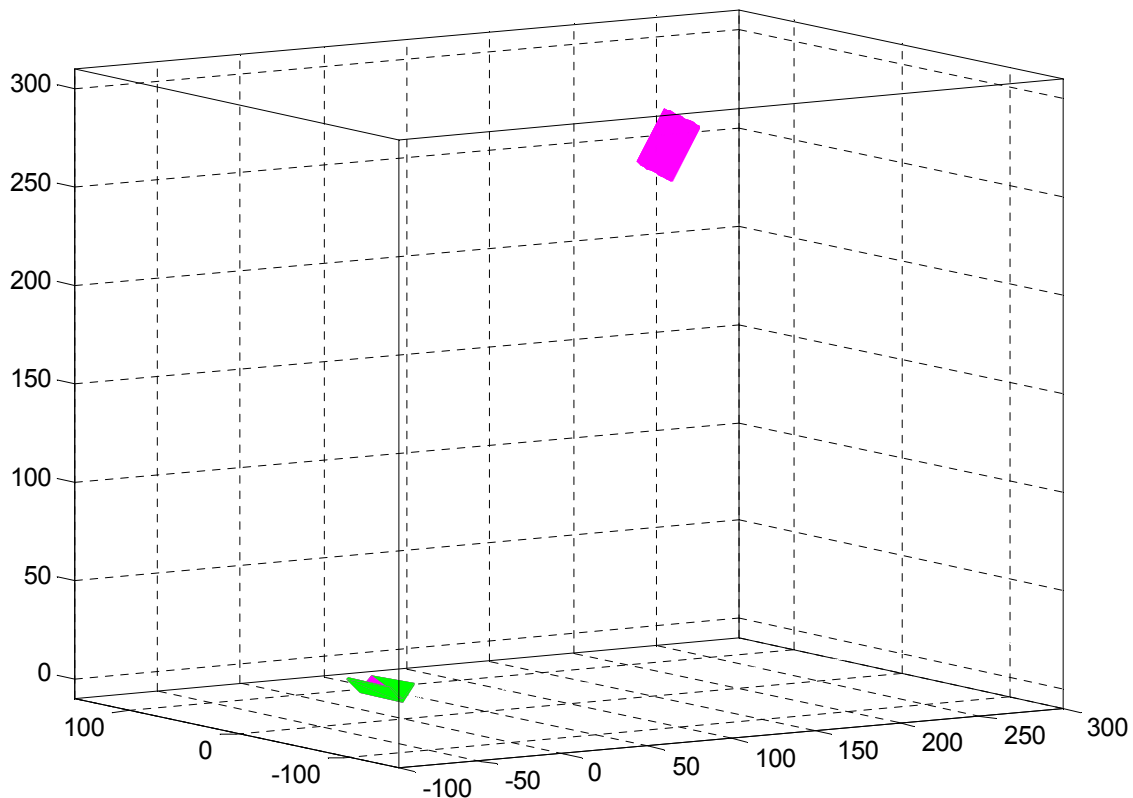
1 lentelė. Bazinių masyvų dydžiai prie skirtingų K_{dtk} .

K_{dtk}	Masyvo elementų skaičius	Masyvo dydis, b
1	20449	163592
2	81225	649800
3	182329	1458632
4	322624	2580992
5	504100	4032800
6	725904	5807232
7	988036	7904288
8	1288225	10305800
9	1630729	13045832
10	2013561	16108488
11	2433600	19468800
12	2896804	23174432



10 pav. Bazinių masių dydžių priklausomybė nuo diskretizavimo koeficiento K_{dtk}

Dažniausiai masyvai yra būtent tokių nurodytų dydžių, tačiau tas dydis gali šiek tiek kisti priklausomai nuo objekto sudėtingumo, bangos ilgio ir kitų parametru. Šie tyrimai buvo atliekami skaičiuojant atspindėtą signalą nuo objekto (žr. 11 pav.), kuris buvo atitolęs nuo keitiklio 400 cm, pasuktas 60° kampu (12 pav.).



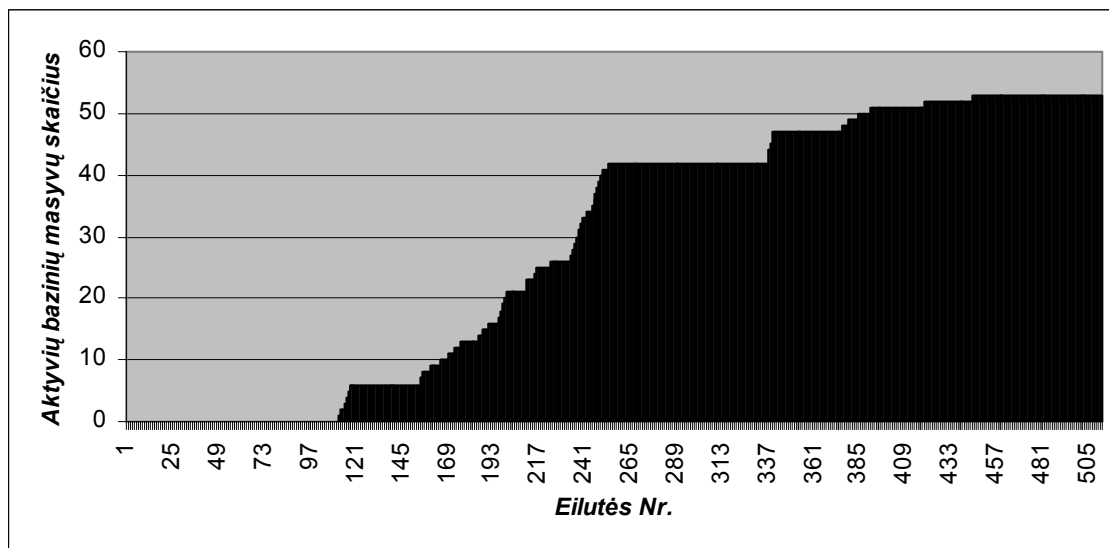
11 pav. Keitiklis ir skenuojamas objektas



12 pav. Tyrimų metu analizuojamas objektas.

Nesunku pastebėti, kad ši priklausomybė auga kvadratine progresija.

Be abejo, masyvo dydį negalima sumažinti. Klausimas ar jie negali būti atlaisvinti? Šio dalyko nustatymui buvo atlikta programos užimamos atminties stebėjimas po kiekvienos programos eilutės ir gauti rezultatai yra pateikti 13 pav.



13 pav. Modelio veikimo bazinių masyvų naudojimas skirtingose programos veikimo vietose

Matosi pastovus atminties augimas, nes atmintis niekada nebuvo atlaisvinama. Šis faktas, be abejo, yra taisytinas.

3.2. Programos modelio greičio tyrimas

Programos modelis yra skirtas darbui personaliniame kompiuteryje. Šio atveju atminties dydis faktiškai nepriklauso nuo naudojamo kompiuterio tipo. Visai kitaip yra su skaičiavimo greičiu. Absoliutinis skaičiavimo greitis labai priklauso nuo naudojamo kompiuterio parametrų ir ne tik tiesiogiai nuo procesoriaus greičio bet ir nuo operatyvios atminties dydžio bei spartos, kietojo disko greičio ir kitų parametrų. Todėl norint gauti kiek galima patikimesnius rezultatus, tyrimai buvo

atlikti keičiant to paties kompiuterio operatyvios vienodo greičio atminties dydį. Naudoto kompiuterio parametrai yra išvardinti žemiau:

- Intel® Pentium® Dual Core T2060 procesorius (1.60 GHz, 533 MHz FSB, 1 MB L2 Cache)
- SODIMM DDR2 DDR667 operatyviosios atminties moduliai
- 5400 RPM SATA kietasis diskas

Bandymai buvo atliekami esant skirtingiems operatyviosios atminties dydžiams: 512, 1024 bei 2048 MB, su kiekvienu operatyviosios atminties dydžiu buvo keičiamas K_{atk} iki tol, kol modelis neparodydavo klaidos pranešimo, kad neužtenka atminties. Buvo stebimi 5 daugiausiai laiko reikalaujančios programos metodai, t.y.: *ReflectionCoeff_F01FD5*, *ShowScanningInFig*, *Pasukimas_trans5_LM*, *Keitiklis_pavaizduot_nb* bei *Pasukimas_trans2*.

ReflectionCoeff_F01FD5 yra pagrindinis modeliavimo metodas. Pagrindinė šio metodo funkcija yra apskaičiuoti atspindžio signalą. *ShowScanningInFig* ir *Keitiklis_pavaizduot_nb* funkcijos yra atitinkamai atsakingos už figūros bei keitiklio atvaizdavimą ekrane. *Pasukimas_trans5_LM* bei *Pasukimas_trans2* metodai yra naudojami *ReflectionCoeff_F01FD5* viduje ir atlieka koordinatinių sistemų transformavimus.

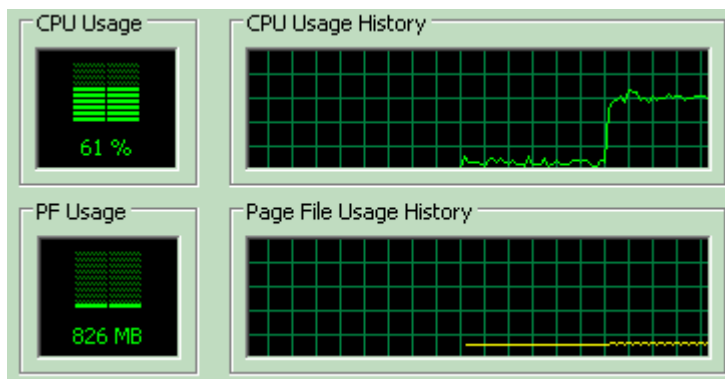
Norint gauti kuo tikslesnį rezultatą, laikas buvo matuojamas *MatLab Profiler* įrankio pagalba. *MatLab Profiler* suskaičiuoja visų naudojamų funkcijų skaičiavimo laiką, rodo kritines eilutes, jų darbo trukmę ir t.t [18]. Tuo pačiu metu buvo stebimas *Windows Task Manager* programos *Performance* skiltis, kuris parodo procesoriaus bei visos naudojamos atminties būsenos, bei jų laikines diagramas.

Rezultatai gauti atliekant bandymus su 512 MB operatyviosios atminties yra pateikti 2 lentelėje.

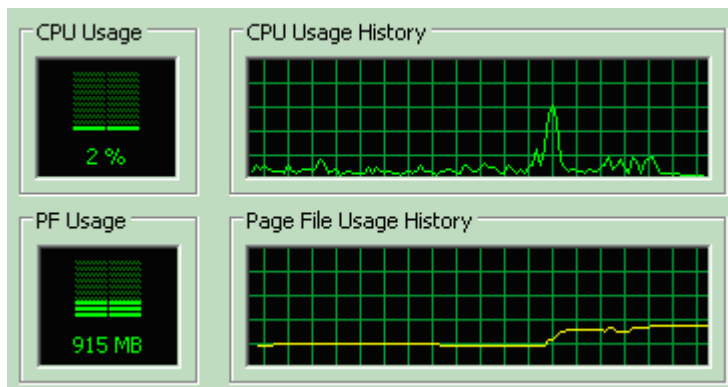
2 lentelė. Pagrindinių programų skaičiavimo laikai sekundėmis su 512MB

K_{atk}	1	2	3	4	5	6	7	8	9
<i>ReflectionCoeff_F01FD5, s</i>	0.567	1.186	2.342	3.763	13.769	25.943	47.005	116.732	283.953
<i>ShowScanningInFig, s</i>	2.68	0.775	0.775	0.725	1.392	3.534	2.456	6.757	12.849
<i>Pasukimas_trans5_LM, s</i>	1.021	0.063	0.14	0.212	0.619	1.342	2.399	6.953	18.149
<i>Keitiklis_pavaizduot_nb, s</i>	1.276	0.411	0.45	0.378	0.712	1.387	1.199	2.938	4.979
<i>Pasukimas_trans2, s</i>	0.315	0.364	0.388	0.317	0.557	0.984	0.971	2.35	3.694

Esant $K_{dtk} = 10$ modelis nustojo funkcionuoti dėl atminties persipildymo. 14 pav. yra pavaizduotas *Windows Task Manager* programos fragmentas, kuriame yra pavaizduoti procesoriaus ir atminties apkrovos laikinės diagramos, modelio veikimo metu prie $K_{dtk} = 4$, o 15 pav. - prie $K_{dtk} = 9$.



14 pav. Atminties bei procesoriaus apkrova prie $K_{dtk} = 4$



15 pav. Atminties bei procesoriaus apkrova prie $K_{dtk} = 9$

Akivaizdžiai matosi, kad prie $K_{dtk} = 4$ procesorius yra apkrautas maždaug 60%, o prie $K_{dtk} = 9$ atminčiai pasiekus tam tikrą ribą, procesoriaus apkrova krenta labai smarkiai, kas ir sąlygoja ilgą veikimo laiką. Tai atsitinka dėl to, kad programos veikimui nebeužtenka operatyvios atminties ir ji pradeda naudoti virtualią atmintį, kuri yra daug lėtesnė už pirmąją ir kietasis diskas nespėja pilnai apkrauti procesoriaus. Šiuolaikinės operatyviosios atminties greitis prasideda nuo 3200MB/sek [19], o kietojo disko (virtualiosios atminties atveju) maždaug nuo 20MB/sek [20].

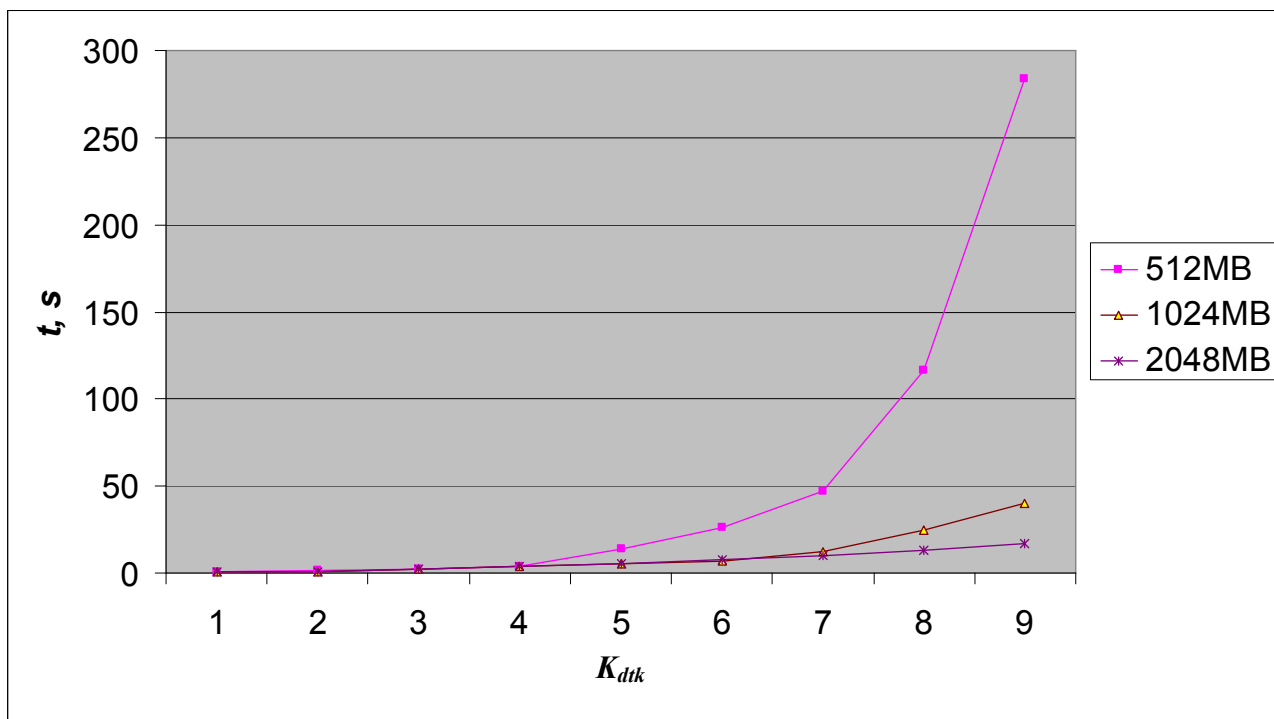
3 ir 4 lentelėse yra pavaizduoti gauti rezultatai esant 1024 ir 2048MB operatyvios atminties, atitinkamai. Visų trijų matavimų grafinis vaizdavimas yra parodytas 16 pav.

3 lentelė. Pagrindinių metodų skaičiavimo laikai su 1024MB

K_{dtk}	1	2	3	4	5	6	7	8	9
<i>ReflectionCoeff_F01FD5</i>	0.536	1.096	2.389	3.72	5.45	7.12	12.21	24.9	40.311
<i>ShowScanningInFig</i>	0.771	0.731	0.72	0.684	0.774	0.71	0.864	1.654	2.334
<i>Pasukimas_trans5_LM</i>	0.034	0.061	0.123	0.187	0.294	0.417	0.695	1.654	2.006
<i>Keitiklis_pavaizduot_nb</i>	0.402	0.411	0.383	0.358	0.403	0.386	0.432	0.809	0.954
<i>Pasukimas_trans2</i>	0.302	0.305	0.306	0.28	0.356	0.324	0.376	0.704	0.756

4 lentelė. Pagrindinių metodų skaičiavimo laikai su 2048MB

K_{dtk}	1	2	3	4	5	6	7	8	9
<i>ReflectionCoeff_F01FD5</i>	0.427	1.048	2.06	3.536	5.281	7.505	10.127	13.3	16.851
<i>ShowScanningInFig</i>	0.647	0.703	0.712	0.71	0.685	0.772	0.703	0.701	0.72
<i>Pasukimas_trans5_LM</i>	0.029	0.03	0.106	0.154	0.28	0.386	0.531	0.67	0.861
<i>Keitiklis_pavaizduot_nb</i>	0.338	0.374	0.363	0.355	0.343	0.386	0.359	0.358	0.376
<i>Pasukimas_trans2</i>	0.294	0.269	0.303	0.293	0.28	0.309	0.281	0.311	0.313



16 pav. Laiko priklausomybė nuo K_{dtk} dydžio su skirtingais operatyviosios atminties dydžiais.

Kaip matome iš grafiko, kad esant $K_{dtk} = 4$, atminties dydis nėra aktualus ir skaičiavimo laikas su visais atminties dydžiais yra praktiškai vienodas. Tačiau toliau didinant K_{dtk} , programos veikimo laikas prie 512MB operatyvios atminties pradeda sparčiai didėti. Esant K_{dtk} iki 7, programos veikimo greitis su 1024 ir 2048MB taip pat kinta nežymiai, skirtumas prasideda tikrai

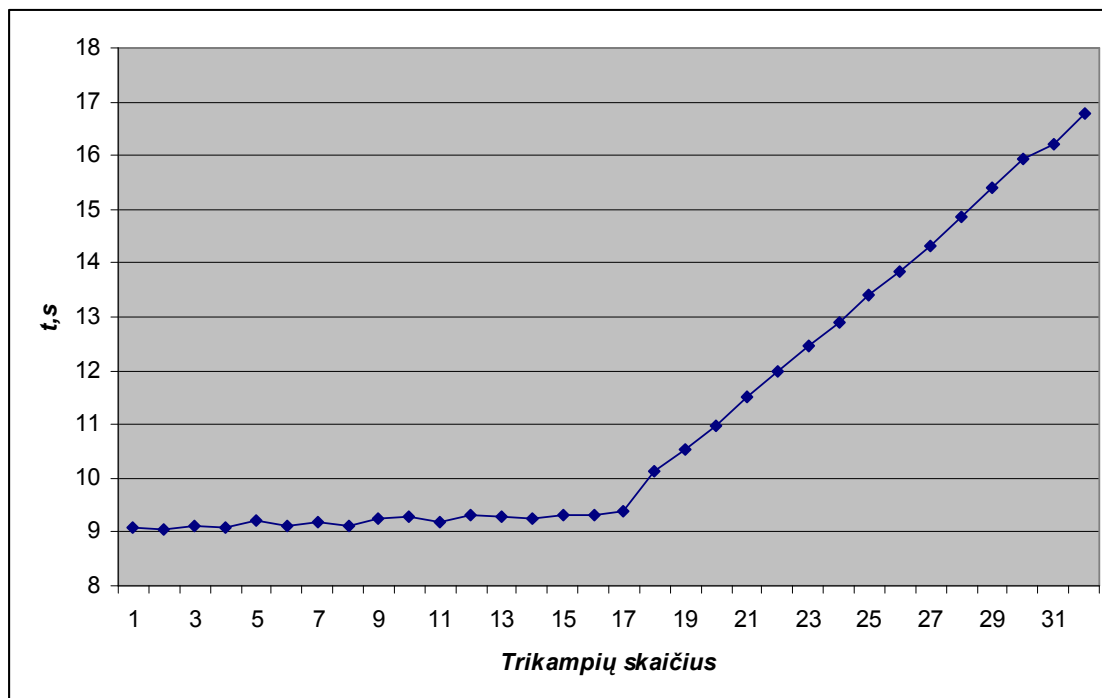
toliau didinant K_{atk} , tačiau esant $K_{atk} = 10$, modelis išsoko klaidingas pranešimas, kad neužtenka atminties tolimesniam programos veikimui prie visų operatyvios atminties dydžių.

Skaiciavimo greitis taip pat priklauso nuo objekto sudėtingumo, o tiksliau nuo trikampių, patenkančių į skaičiavimo zoną, skaičiaus. Sekantys bandymai buvo atliekami su 2GB operatyviosios atminties.

Bandymų metu buvo dirbtinai keičiamas patenkančių į skaičiavimo zoną trikampių skaičius nuo 1 iki 32, t.y. buvo apribuojamas skaičiuojamų trikampių skaičius ir išmatuotas programos veikimo greitis prie $K_{atk} = 5$. Matavimų rezultatai pateikti 5 lentelėje ir pavaizduoti 17 pav.

5 lentelė. ReflectionCoeff_F01FD5 metodo veikimo laikai

<i>Tr. skaičius</i>	<i>t, s</i>	<i>Tr. skaičius</i>	<i>t, s</i>	<i>Tr. skaičius</i>	<i>t, s</i>	<i>Tr. skaičius</i>	<i>t, s</i>
1	9.083	9	9.238	17	9.369	25	13.405
2	9.031	10	9.292	18	10.121	26	13.847
3	9.13	11	9.199	19	10.518	27	14.332
4	9.066	12	9.305	20	10.963	28	14.873
5	9.213	13	9.269	21	11.52	29	15.394
6	9.11	14	9.254	22	11.991	30	15.932
7	9.192	15	9.312	23	12.459	31	16.206
8	9.105	16	9.318	24	12.884	32	16.784



17 pav. Laiko priklausomybė nuo trikampių, patenkančių į skaičiavimo zoną, skaičiaus.

Rezultatai rodo, kad nuo $K_{dtk} = 1$ iki $K_{dtk} = 17$ skaičiavimo laikas kylo nedideliais žingsniais. Kai kuriais momentais skaičiavimo laikas buvo mažesnis, tačiau tai atsitiko dėl to, kad skirtumas tarp dviejų taškų yra labai mažas ir jis visada biški skirsis dėl skirtingų lygiagrečiai paleistų procesų bandymų metu. Išvengti jų yra praktiškai neimanoma, nes tai yra operacinės sistemos tačiau prie 18 trikampių laikas pradeda didėti ženkliai. Tai atsitinka dėl to, kad skaičiavimo laikas priklauso ne tik nuo trikampių skaičiaus, bet ir nuo jų išsidėstymo erdvėje. Pavyzdžiui, jei trikampiai išsidėstė vienoje plokštumoje ir visi atspindi ultragarsines bangas ta pačia kryptimi, programos dalis, atliekanti kampinių atspindžių skaičiavimas faktiškai nedirba (žr. 17 psl.).

Išvados

Atminties analizė parodė, kad:

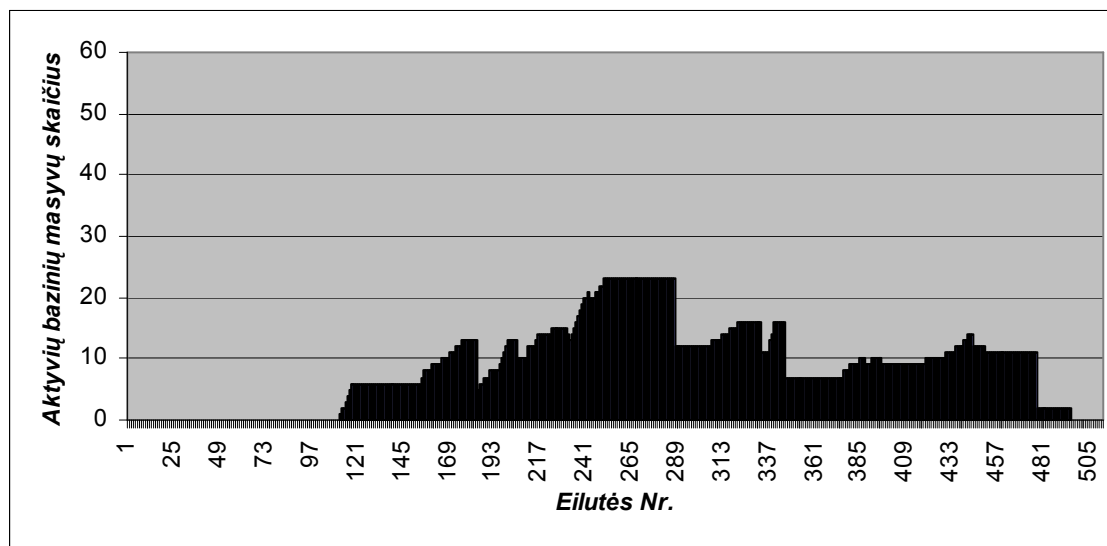
- 1) Programos veikimo metu nėra valomi pagrindiniai laikino saugojimo masyvai, tuo būdu programos veikimo metu gaunamas pastovus atminties augimas, kuo pasėkoje prie $K_{dtk} > 9$ (su ≥ 1024 Mb operatyvios atminties) persipildo atmintis ir programos veikimas nutrukdavo;
- 2) Skaičiavimo laiko analizė parodė, kad skaičiavimo laiko priklausomybė nuo diskretizavimo koeficiento galima suskaidyti į 3 dali:
 - Kai skaičiavimo laikas dėsningai didėja priklausomai nuo diskretizavimo taškų skaičiaus;
 - Jis ženkliai išaugo dėl virtualių masyvų panaudojimo;
 - Toliau dar padidinus diskretizavimo skaičių, modelis visiškai negali funkcionuoti.

Apibendrinus visą išvadą galima teigti, kad esamo modelio versija faktiškai neleidžia skaičiuoti naudojant K_{dtk} net ant galingesniu kompiuteriu dėl atminties stokos.

4. Modelio programos optimizavimas.

4.1. Atminties sutvarkymas

Kaip buvo paminėta ankstesniame skyriuje, pagrindiniai naudojami masyvai nebuvo valomi iš karto po to, kai jie yra toliau nebenaudojami. Šio trukumo pašalinimui *ReflectionCoeff_F01FD5* funkcija buvo pertvarkyta ištrinant masyvus iškart, kai jie nenaudojami. Šio atminties valymo pasiekoje atmintis buvo išnaudota net 70% efektyviau. (žr. 18 pav.)

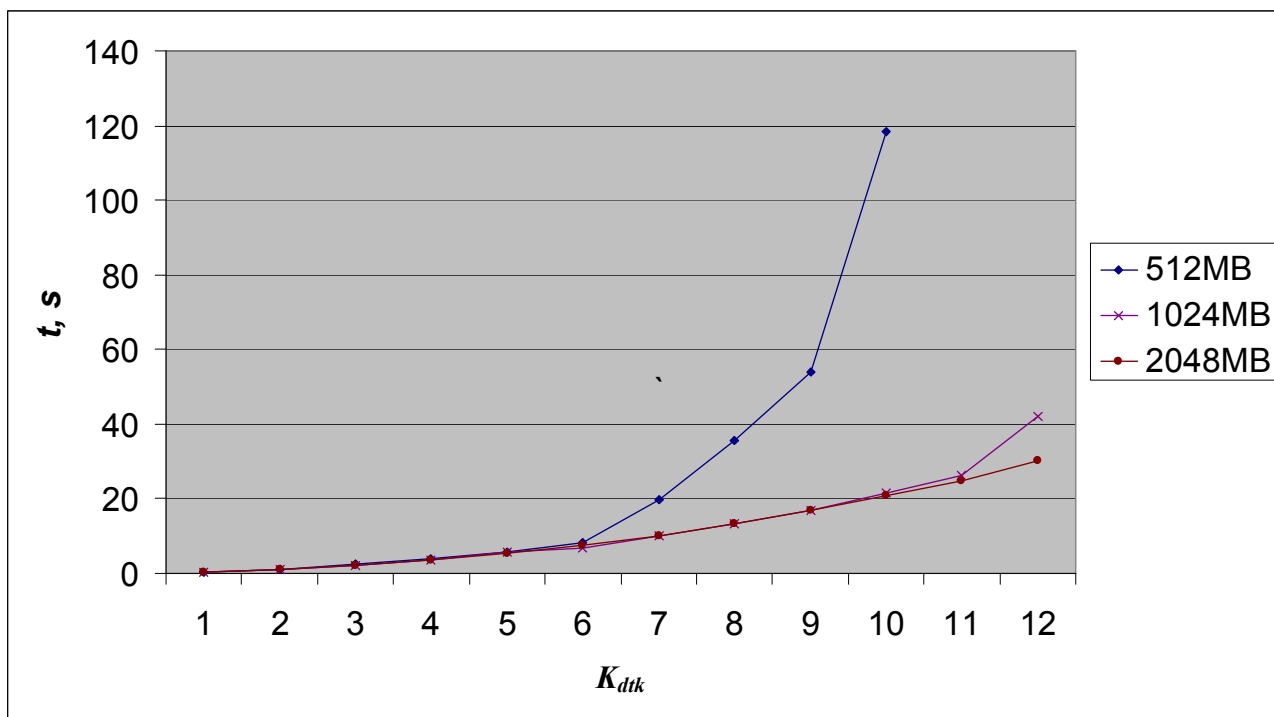


18 pav. Programos modelio atminties išnaudojimas po masyvų valymo.

Siekiant nustatyti, kokia įtaka programos veikimo greičiui turėjo atlikta modifikacija, buvo pakartotinai paimti *ReflectionCoeff_F01FD5* programos veikimo laikai prie $K_{dik} = 1 \div 12$. Gauti rezultatai yra pateikti 6 lentelėje esant 512, 1024 bei 2048MB operatyvios atminties ir atitinkamai bei pavaizduoti 19 pav.

6 lentelė. Pagrindinių metodų skaičiavimo laikai po atminties valymo su 2048MB

K_{dik}	1	2	3	4	5	6
<i>ReflectionCoeff_F01FD5</i> (512Mb), s	0.521	1.184	2.341	3.795	5.886	8.301
<i>ReflectionCoeff_F01FD5</i> (1024Mb), s	0.507	1.178	2.202	3.705	5.603	7.001
<i>ReflectionCoeff_F01FD5</i> (2048Mb), s	0.472	1.134	2.112	3.54	5.406	7.6
K_{dik}	7	8	9	10	11	12
<i>ReflectionCoeff_F01FD5</i> (512Mb), s	19.8	35.6	53.935	118.372	---	---
<i>ReflectionCoeff_F01FD5</i> (1024Mb), s	10.1	13.4	16.76	21.544	26.266	42.136
<i>ReflectionCoeff_F01FD5</i> (2048Mb), s	10.056	13.293	16.903	21.027	24.996	30.126



19 pav. Laiko priklausomybė nuo smulkumo koeficiento dydžio su skirtingais operatyviosios atminties dydžiais po atminties valymo.

Kaip matome, kad skirtumas tarp 13 pav. ir 18 pav. yra akivaizdus. Modelis naudoja žymiai mažiau operatyviosios atminties, tačiau jos dydis vis tiek yra ribotas. Norint gauti didesnę rezultatų kokybę, operatyviosios atminties nebeužtenka. Su atminties išvalymu iki $K_{dtk} = 6$ visos priklausomybės prie skirtingų operatyviosios atminties dydžių yra panašūs. Prie koeficiento = 7, 512MB operatyvios atminties jau nebeužtenka ir programos veikimo laikas pradeda sparčiai didėti. Skirtumas tarp 1024MB ir 2048MB pradėjo reikštis tikrai prie $K_{dtk} = 12$, o dar padidinus K_{dtk} , programą nustojo veikti, nes neužteko atminties ir su 1024Mb ir su 2048Mb.

Programos veikimo ribos padidėjo iki $K_{dtk} = 9$ su 512MB ir iki $K_{dtk} = 12$ su 1024MB bei 2048MB, tačiau norint gauti detalesnę vaizdą, reikėtų K_{dtk} didinti dar.

Ankstesnių ir dabartinių rezultatų palyginimui, patogiausiai būtų padalinti rezultatus į 3 dalis:

- 1) $K_{dtk} = 1 \div 4$ (užsiriboja atmintis prieš valymą su 512MB operatyviosios atminties)
- 2) $K_{dtk} = 5 \div 7$ (užsiriboja atmintis prieš valymą su 1024MB operatyviosios atminties)
- 3) $K_{dtk} = 8 \div 9$ (Programa nustojo veikti)

Šių intervalų suminiai laikai yra pateikti 7 lentelėje.

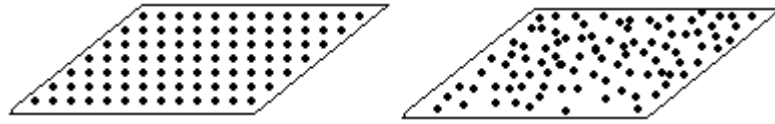
7 lentelė. *ReflectionCoeff_F01FD5* metodo skaičiavimo laikai su skirtingais K_{dtk} , esant 512, 1024 ir 2048MB operatyviosios atminties prieš ir po atminties valymo

K_{dtk} \ t, s	$\Sigma_{t,512MB}$		$\Sigma_{t,1024MB}$		$\Sigma_{t,2048MB}$	
	<i>Prieš</i>	<i>Po</i>	<i>Prieš</i>	<i>Po</i>	<i>Prieš</i>	<i>Po</i>
$1 \div 4$	7.858	7.841	7.741	7.592	7.071	7.258
$5 \div 7$	86.717	33.987	24.78	22.704	22.913	23.062
$8 \div 9$	400.685	89.535	65.211	30.16	30.151	30.196

Nesunku pastebėti, kad tose ribose, kur atmintis nėra užsiribojusi, skaičiavimo laikai yra labai panašūs ir juos galima laikyti vienodais. Intervale $K_{dtk} = 5 \div 7$ skaičiavimo greitis su 512MB operatyviosios atminties sumažėjo $\approx 61\%$, o $7 \div 9$ intervale - $\approx 78\%$. Su 1024MB, intervale $K_{dtk} = 8 \div 9$ programa efektyviau veikė $\approx 54\%$, o su 2048MB skaičiavimo laikas praktiškai nepasikeitė.

4.2. Modeliavimo ribų praplėtimas

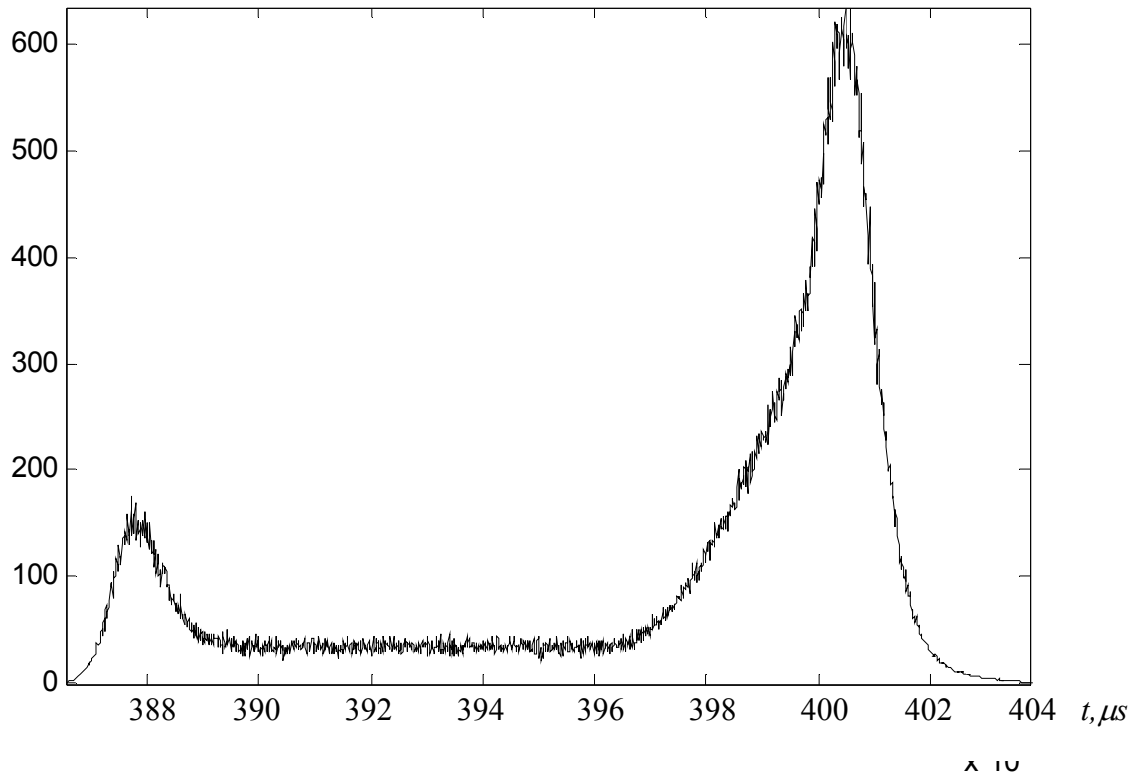
Tyrimai rodo, kad vienas iš esminių modelio trukumų yra tai, kad jis iš principo neleidžia skaičiuoti su $K_{dtk} > 12$. Siekiant panaikinti šį trukumą, buvo detaliau išsiaiškintas naudojamas modelio diskretizavimo metodas. Daugeliui modeliavimo uždavinių erdvinis diskretizavimas, jei toks reikalingas, yra atliekamas naudojant reguliarų erdvinį tinkliuką su tam tikru erdvinio žingsniu (20 pav., a) [21]. Skaičiuojant pakartojantį atspindį, naudojant reguliarų tinkliuką, visada gausi tą patį rezultatą ir didesnis taškų kiekis visada gali būti gautas tikta naudojant smulkesnį diskretizavimo žingsnį. Šiame modelyje erdvinis diskretizavimas yra atliekamas naudojant atsitiktinių skaičių generavimo funkcija ir taškai atsitiktinai išbarstomi tame paviršiuje (20 pav., b). Šiuo atveju, atliekant pakartotinius skaičiavimus, atspindinančiame paviršiuje gaunamas kitas atsitiktinių taškų rinkinys ir atspindėto signalo rezultatas yra panašus, bet kažkiek skirsis ir pakartotiniai skaičiavimo rezultatai faktiškai vienas kitą papildo ir gali būti sudedami. Todėl, norint gauti tikslesnį rezultatą esant smulkesniam diskretizavimui, galima arba sugeneruoti didesnį taškų skaičių atspindinčiame objekte, ką darydavo šita programa iki šiol, arba atlikti pakartotinių skaičiavimų ir juos susumuoti. Rezultatai gaunami yra faktiškai identiški, tačiau antras siūlomas metodas nereikalauja be galo didelių masyvų ir faktiškai įgalina suskaičiuoti su bet koku norimai smulkiu diskretizavimo žingsniu, nors aišku tokie skaičiavimai truks ilgai, tačiau įmanomi.



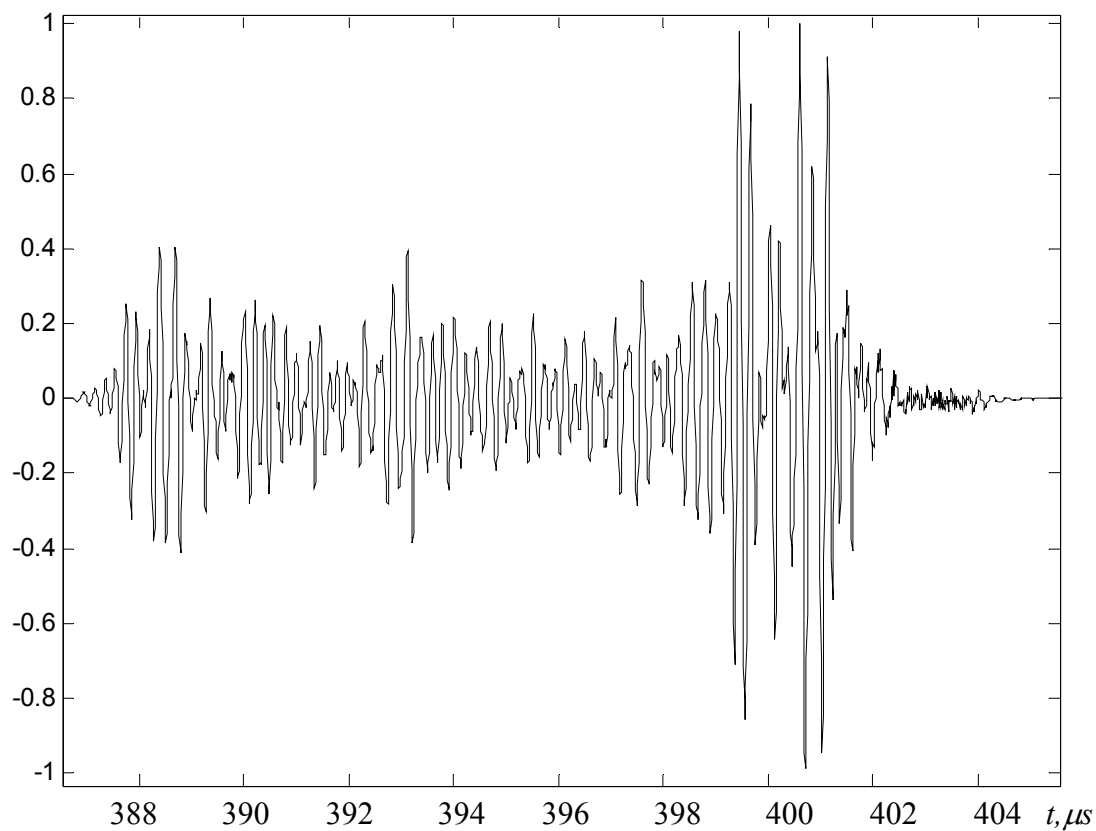
20 pav. Erdvinio diskretizavimo išbarstymo pavyzdžiai reguliariajame tinkliuke: *a* – su nustatytais diskretizavimo taškais, *b* – su atsitiktiniais diskretizavimo taškais

Norint įvertinti kiek tokia modifikacija užima laiko lyginant su ankstesniuoju variantu, buvo atlikti 2 bandymai. Pirmasis modeliavimas buvo atliekamas prie $K_{dtk} = 10$, antrasis prie $K_{dtk} = 1$, tačiau skaičiavimai buvo atliekami $10^2=100$ kartų. Paprastumo dėlei buvo įvestas naujas koeficientas N , kuris nusako kiek kartu buvo vykdomas ciklas. Jei $N = 2$, tai skaičiavimai buvo atliekami 4 kartus, jei $N = 3$ - 9 kartus ir t.t.

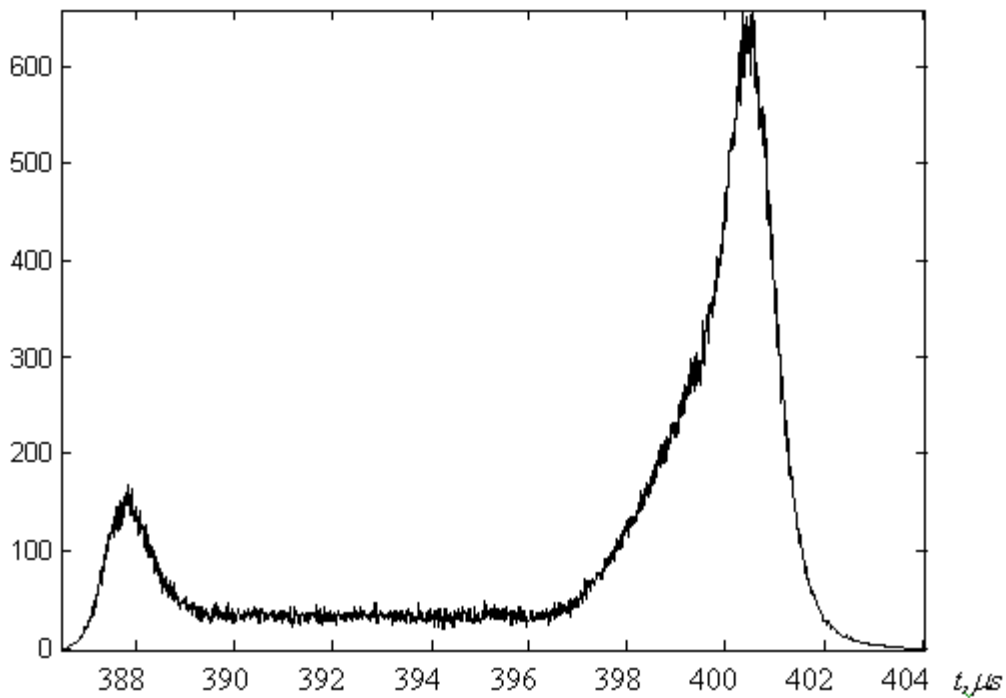
Abiejų bandymų metu buvo saugomas gautos atspindžio funkcijos. Gautų atspindžių funkcijų grafiniai vaizdai yra pateikti 21 ir 22 pav., o signalų, apskaičiuotų iš šių atspindžių funkcijų, vaizdai yra pateikti 23 ir 24 pav.



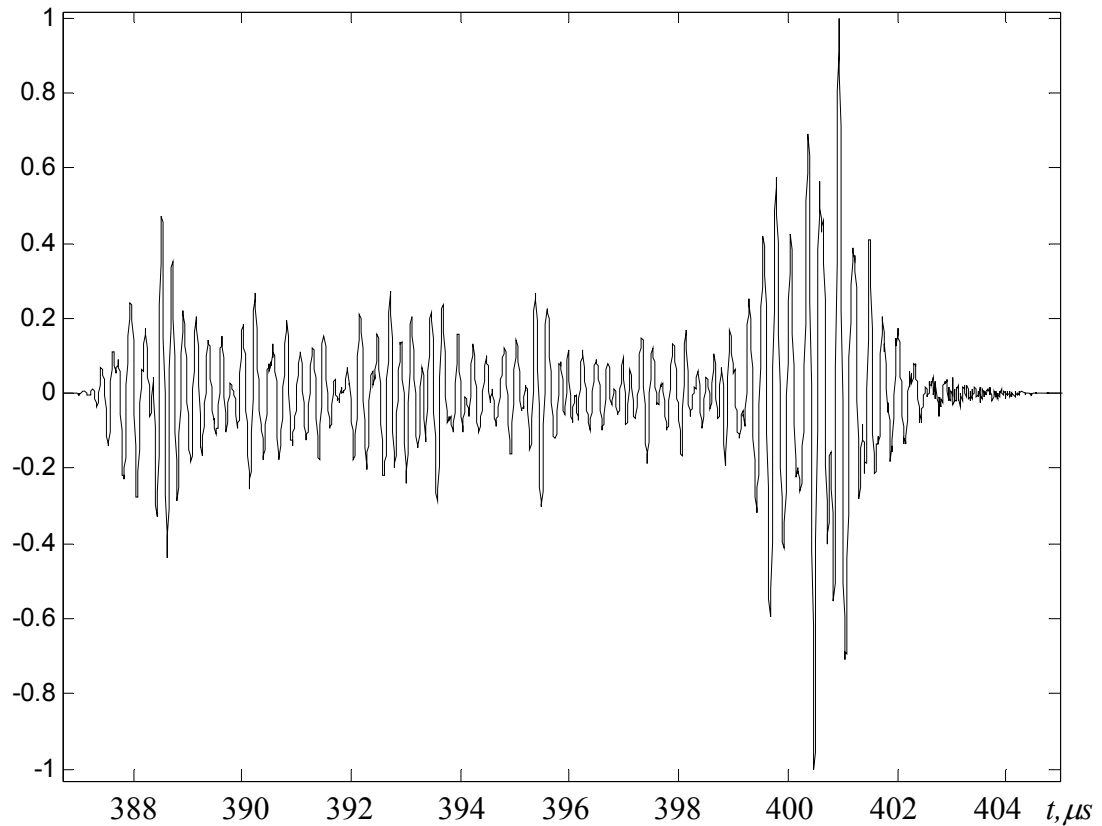
21 pav. Atspindžio funkcijos grafinis vaizdavimas prie $K_{dtk} = 10$.



22 pav. Apskaičiuoto signalo grafinis vaizdavimas prie $K_{atk} = 10$

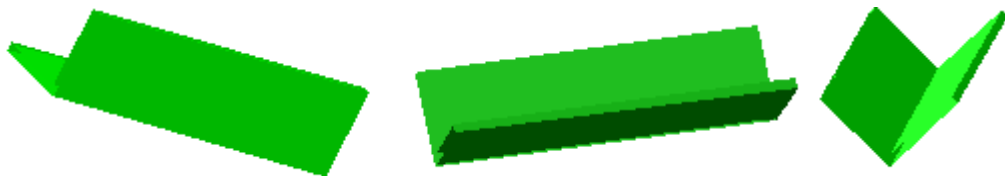


23 pav. Atspindžio funkcijos grafinis vaizdavimas prie $N = 10$



24 pav. Apskaičiuoto signalo grafinis vaizdavimas prie $N = 10$

Kaip matome iš gautų rezultatų, kad jie yra panašūs, tačiau vis tiek skiriasi. Tai atsitinka dėl atsitiktinio triukšmo, kuris iškraipo signalą. Iš tiesų, jei bandymas būtų pakartotas dar kartą, tai gauti signalai prie kintamojo N ir kintamojo K_{dtk} , skirtusi nuo dabartinių. Iš tikrųjų tai kaip skirtingai jie neatrodytų, jie yra labai identiški ir būtent šiai figūrai signalo kokybės pilnai užtenka. Gauto rezultato grafinį modelį iš skirtingų rakursų, galima pamatyti 25 pav.



25 pav. Analizuojamo objekto grafinis modelis iš skirtingų pusių.

Tolimesniu bandymų metu bus matuojami ir lyginami $ReflectionCoeff_F01FD5$ funkcijos darbo laikai prie skirtingų K_{dtk} ir N .

Norint užtikrinti kuo didesnę matavimų tikslumą, matavimai prie skirtingų K_{dtk} buvo pakartoti dar kartą ir rezultatai yra pateikti 8 lentelėje, o modeliavimas prie skirtingų N buvo daromi

iki $N = 16$, kad aiškiau matytųsi darbo veikimo priklausomybė nuo N dydžio. Gauti rezultatai yra pateikti 9 lentelėje.

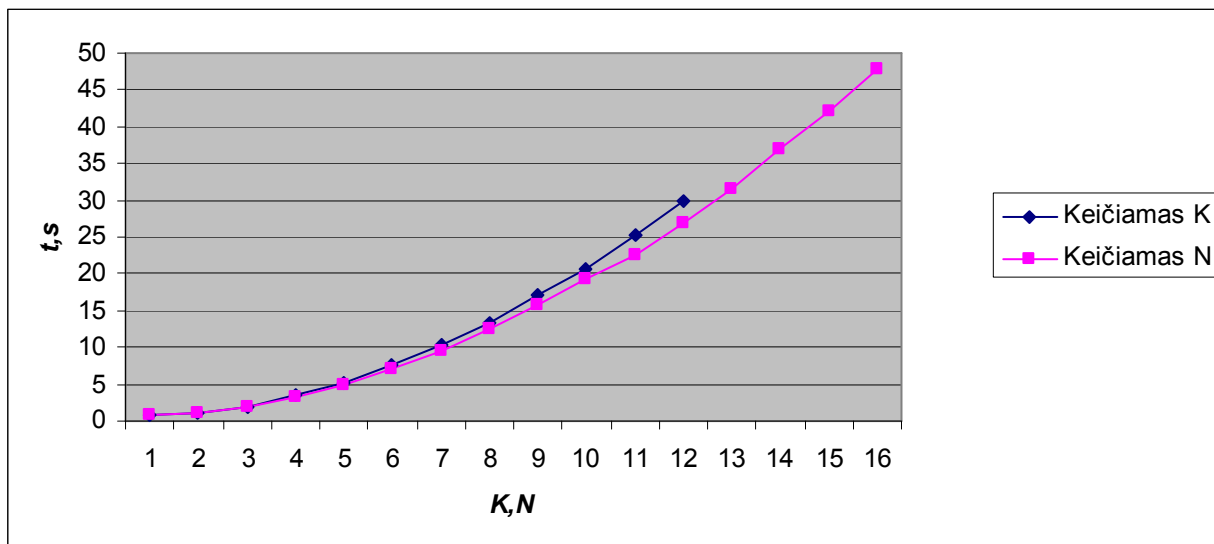
8 lentelė. *ReflectionCoeff_F01FD5* veikimo laikai esant skirtingiems K_{dtk} koeficientams

K_{dtk}	1	2	3	4	5	6
<i>ReflectionCoeff_F01FD5</i> ,s	0.68	1.036	1.997	3.488	5.213	7.526
K_{dtk}	7	8	9	10	11	12
<i>ReflectionCoeff_F01FD5</i> ,s	10.4	13.285	17	20.573	25.156	29.932

9 lentelė. *ReflectionCoeff_F01FD5* veikimo laikai esant skirtingiems N koeficientams

N	1	2	3	4	5	6	7	8
<i>ReflectionCoeff_F01FD5</i> ,s	0.822	1.077	2.025	3.377	5.025	7.15	9.513	12.464
N	9	10	11	12	13	14	15	16
<i>ReflectionCoeff_F01FD5</i> ,s	15.717	19.335	22.43	25.406	31.501	36.981	42.002	47.956

Nesunku pastebėti, kad iki $K_{dtk} = N = 1,2,3$ programa greičiau veikė prie didesnio smulkumo koeficiento, tačiau kuo toliau, tuo efektyviau programa veikė kai buvo dindinamas ciklų skaičius prie pastovaus $K_{dtk} = 1$. Akivaizdžiau tai matosi grafiškai (26 pav).



26 pav. *ReflectionCoeff_F01FD5* veikimo laikai esant skirtingiems K_{dtk} ir N .

Iš šito grafiko jau galima numatyti apytikslį darbo veikimo greitį su didesniu koeficientu N . Vidutinis darbo laikas prie keičiamo N sumažėjo $\approx 8\%$.

Gauti rezultatai parodė, kad programos modifikacija ne tik paspartina programą, bet ir leidžia „praplėsti“ jos veikimo ribas iki begalybės. Atminties atžvilgiu modelis išnaudoja tiek pat

atminties nepriklausomai nuo to, kokio tikslumo reikia, t.y. tiek atminties, kiek jos yra reikalinga prie $K_{dtk} = 1$.

Kitais bandymais buvo bandyta sužinoti kaip keisis programos veikimo greitis jei keisime kartu ir K_{dtk} ir N koeficientus. Kadangi vien keičiant K_{dtk} programos galimybės yra labai ribotos, tai atitinkamai bandymų skaičius buvo nedidelis.

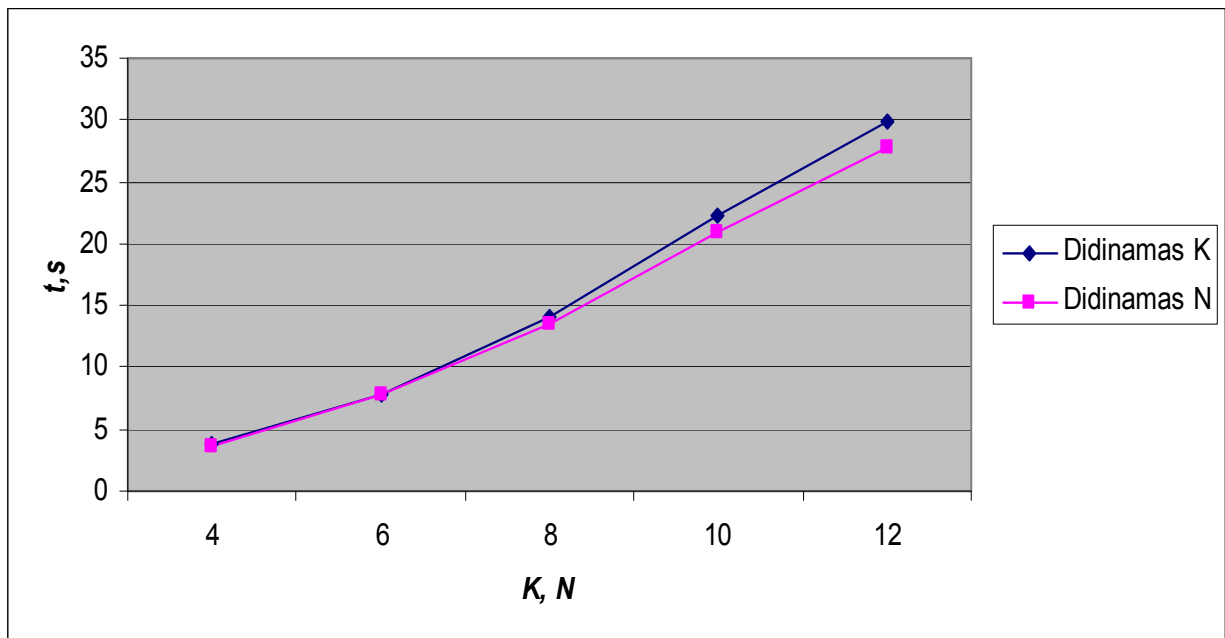
Iš pradžių buvo fiksuojamas $K_{dtk} = 2$ ir keičiamas $N = 2,3,4,5,6$ (10 lentelė), vėliau buvo fiksuojamas $N = 2$ ir keičiamas $K_{dtk} = 2,3,4,5,6$ (11 lentelė). Grafiškai rezultatus galima pamatyti 27 pav.

10 lentelė. *ReflectionCoeff_F01FD5* veikimo laikai esant $K_{dtk} = 2$

N	4	6	8	10	12
<i>ReflectionCoeff_F01FD5</i> (s)	3.565	7.72	13.573	20.947	27.727

11 lentelė. *ReflectionCoeff_F01FD5* veikimo laikai esant $N = 2$

K	4	6	8	10	12
<i>ReflectionCoeff_F01FD5</i> (s)	3.72	7.746	14.016	22.308	29.959



27 pav. Programos veikimo laikinė priklausomybė nuo skirtingų K_{dtk} bei N reikšmių.

Iš grafiko aiškiai matosi, kad ir vėl darbo laiko atžvilgiu programa veikia prie pastovaus K_{dtk} ir kintančio N .

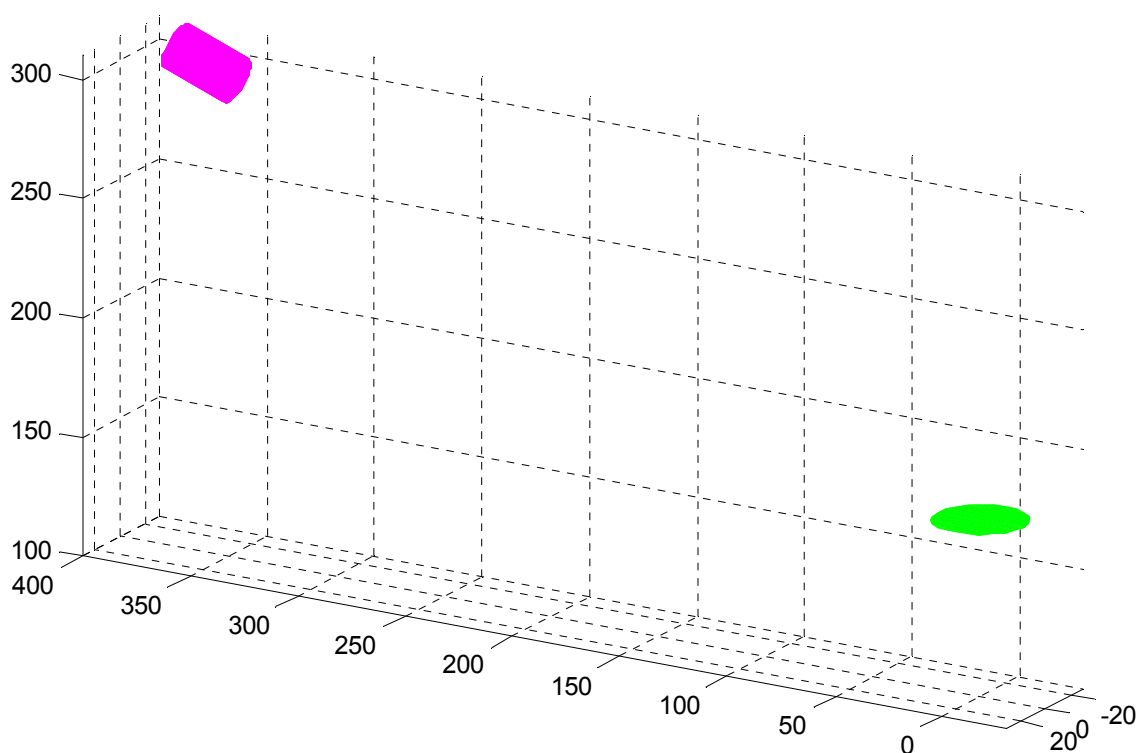
Išvados

Pasiūlytos programos optimizavimas ir modifikacijos įgalina:

- 1) Atminties optimizavimo pasiekoje intervale $K_{atk} = 5 \div 7$ skaičiavimo greitis su 512MB operatyviosios atminties sumažėjo $\approx 61\%$, o $7 \div 9$ intervale - $\approx 78\%$. Su 1024MB, intervale $K_{atk} = 8 \div 9$ programa efektyviau veikė $\approx 54\%$.
- 2) Pasiūlytas modifikavimo algoritmas iš principo panaikino modelio taikymo ribą dėl atminties perpildymo, be to sumažėjo ir skaičiavimo greitis, apytiksliai 8% lyginant su programa po atminties valymo ir 15% lyginant su pradine programos būsena.

5. Atspindžio funkcijos tikslumo priklausomybės nuo diskretizavimo žingsnio tyrimas

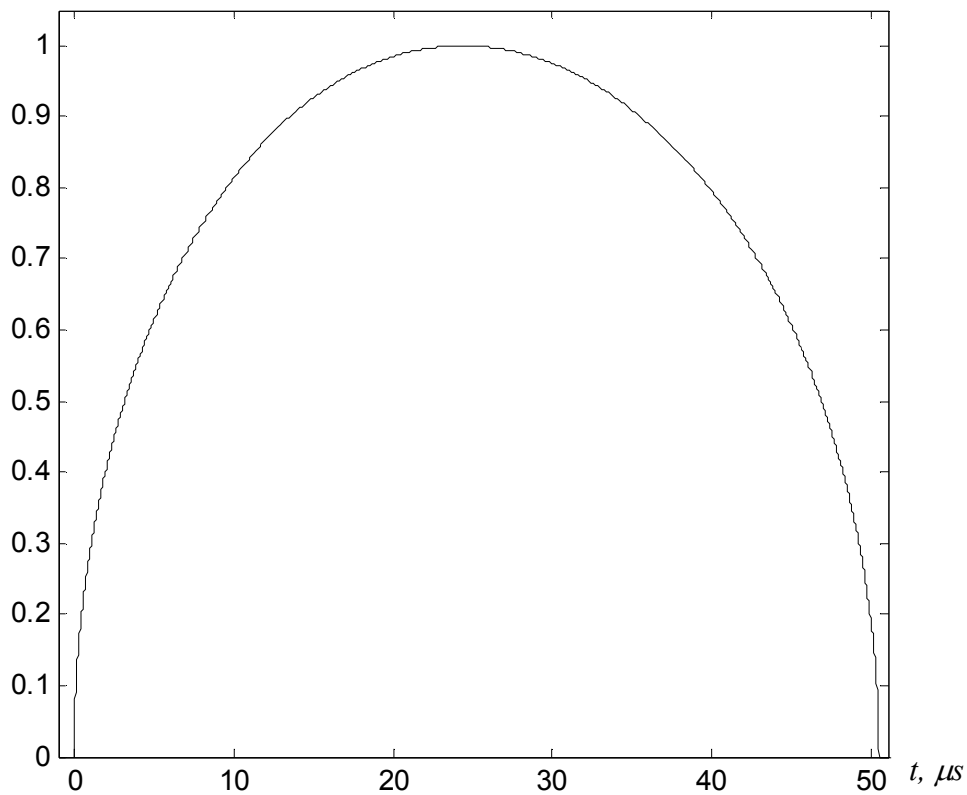
Šio bandymo tikslas buvo išsiaiškinti kaip gautas signalas yra priklausomas nuo smulkumo erdvėje koeficiento N . Bandymai buvo atliekami su refleksiniu disku, kurio skersmuo yra 40mm. Šis diskas buvo paimtas ne atsitiktinai, o dėl to, kad jo atspindžio funkciją ir tuo pačiu signalą galima nesunkiai apskaičiuoti teoriškai ir palyginti su praktiškais rezultatais. Šis diskas buvo atidėtas horizontaliai xOy plokštumai ir pakeltas $z = 174\text{mm}$. Keitiklis buvo įrengtas taške, kurio koordinatės $(346,0,300)$ ir pasuktas 70° kampu z ašies atžvilgiu. Šio bandymo parengimo grafinį vaizdą galima pamatyti 28 pav.



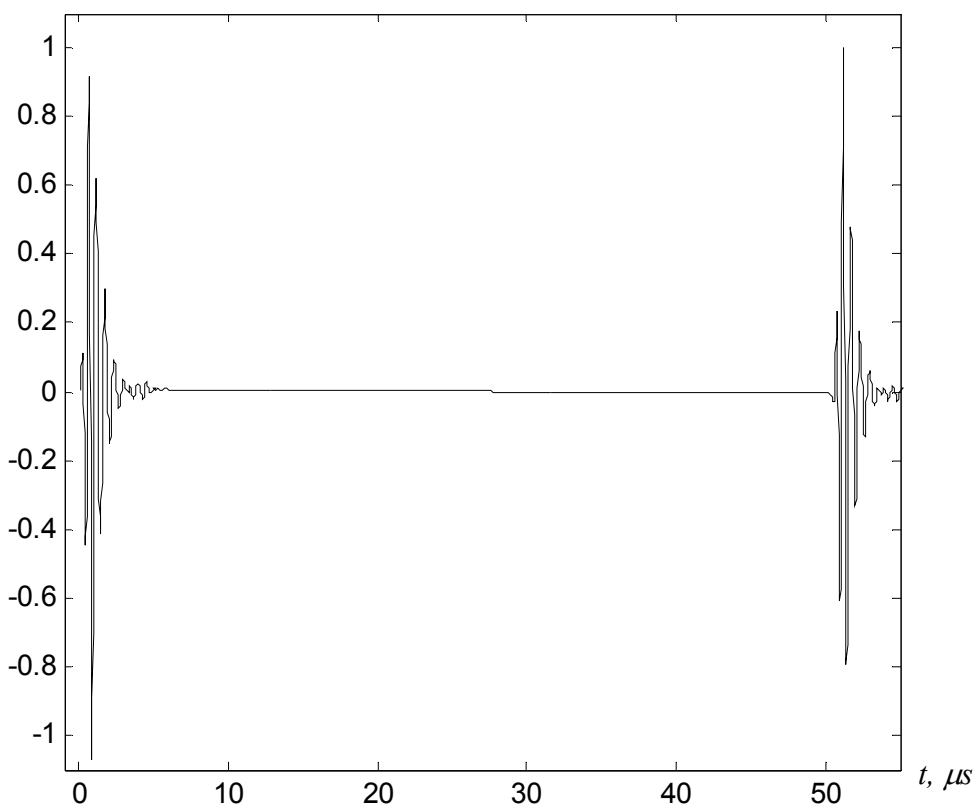
28 pav. Eksperimento paruošimas

Toks kampas buvo parinktas specialiai, kad, keitiklis būtų nukreiptas taip, kad atspindėtų labai silpnus krašto bangų signalus. Tokie signalai daugelyje vizualizavimo sistemoje nėra naudojami, paprastai yra naudojami signalai, atspindėti nuo plokštumų ar kampų, tačiau krašto signalai yra naudojami sąlyginai tiksluose difracciniuose metoduose.

Teoriškai šio objekto signalo atspindžio, bei apskaičiuoto signalo grafinės funkcijos yra pavaizduoti 29 pav ir 30 pav.



29 pav. Teorinė signalo atspindžio funkcija.

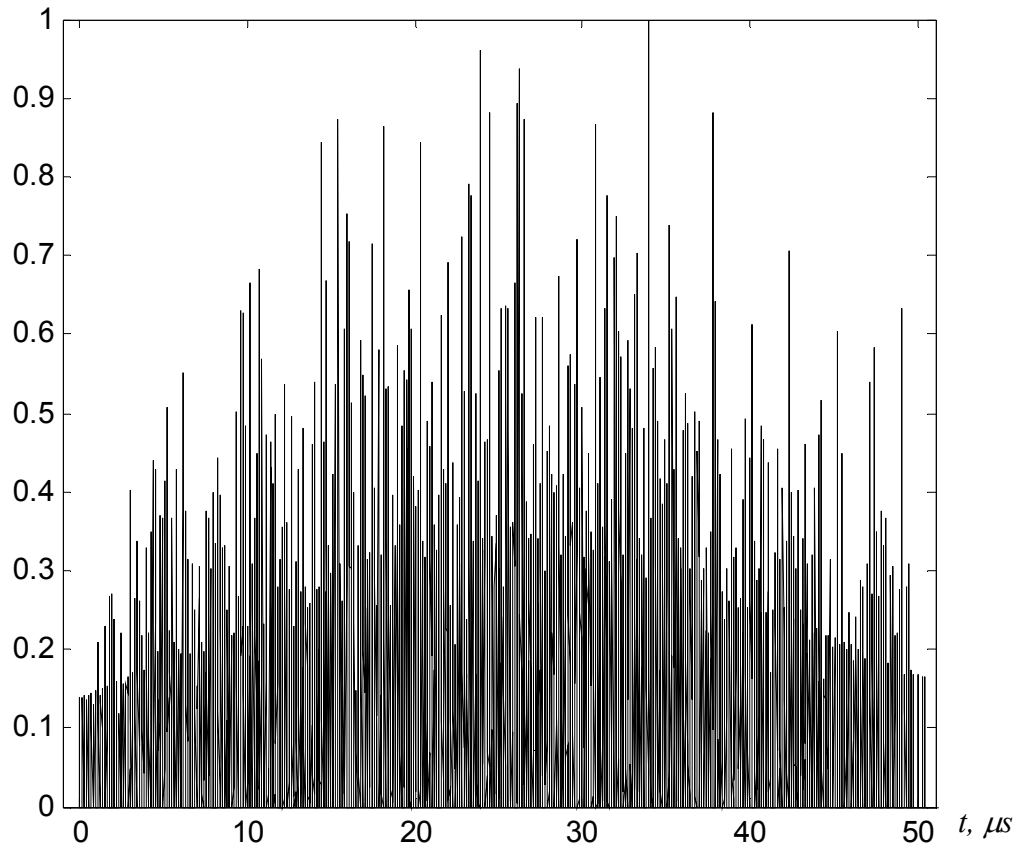


30 pav. Teorinis signalas

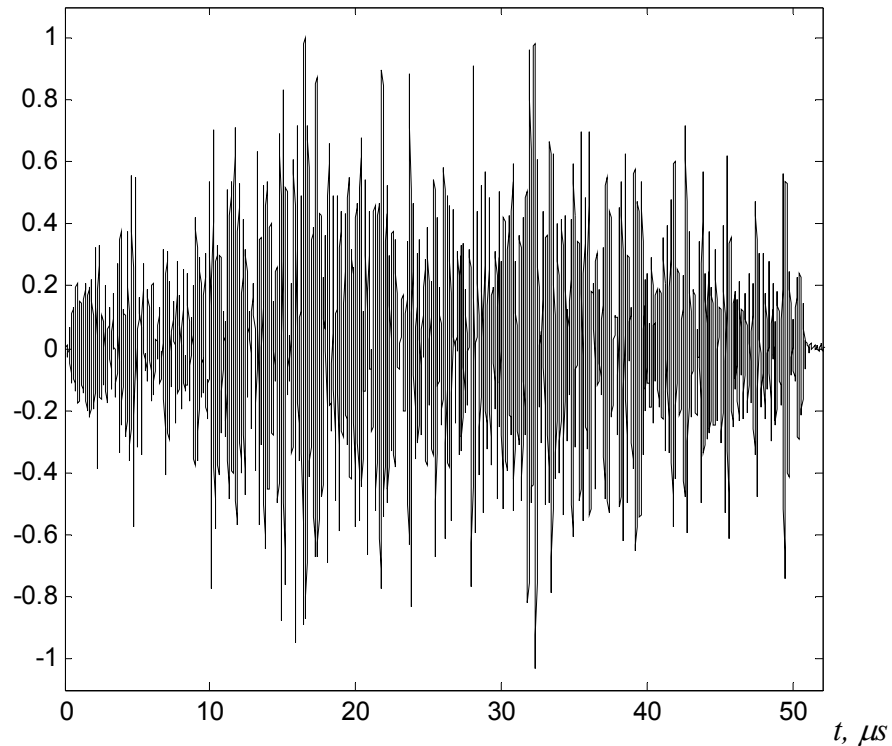
Šis bandymas yra labai svarbus dėl to, kad per didelis N reikalauja labai ilgo skaičiavimo laiko. Šiam eksperimentui keičiamas N (diskretizacijos žingsnis apytiksliai 0,3mm). Kaip buvo minėta anksčiau, kad skaičiavimo laikas auga kvadratine progresija N padidinus vienetui, tai šis koeficientas turi būti tinkamai parinktas, kad galima būtų parinkti optimaliausią laikas/kokybė santykį.

Tyrimų metu buvo atlikta 31 matavimų su $N = 1, 10 \div 300$ koeficientais. Tai yra iš pradžių buvo apskaičiuojami signalai prie $N = 1, N = 10$, o visi sekantys N koeficientai buvo didinami kas 10 iki 300.

Pirmas bandymas buvo atliktas prie $N = 1$, bandymo atspindžio funkcija, bei gautas signalas yra pavaizduoti atitinkamai 31 ir 32 pav.



31 pav. Signalų atspindžio funkcija prie $N = 1$



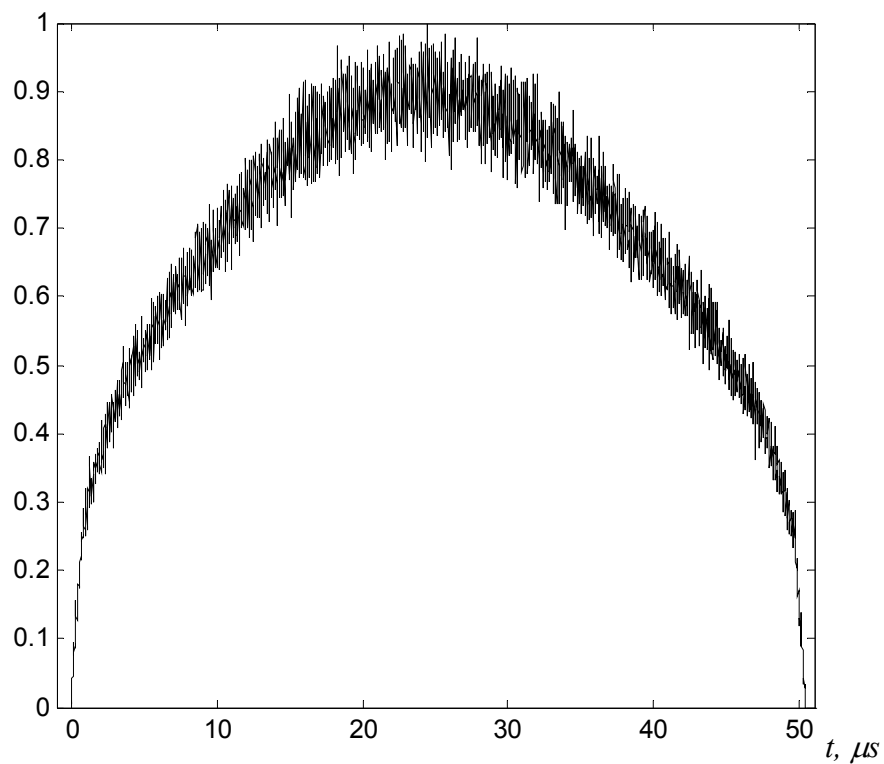
32 pav. Signalas prie $N = 1$

Akivaizdžiai matosi, kad nei atspindžio funkcijos, nei signalo vaizdai nėra net truputi panašūs į teorinius. Teoriškai signalo amplitudė $A_m(5, 48) \mu s$ režiuose turi būti lygi 0 (žr. 30 pav.). Akivaizdu, kad visa kita tuose režiuose yra triukšmas. Norint įvertinti šio signalo kokybę, paprasčiausias būdas yra apskaičiuoti signalo/triukšmo santykį [22]:

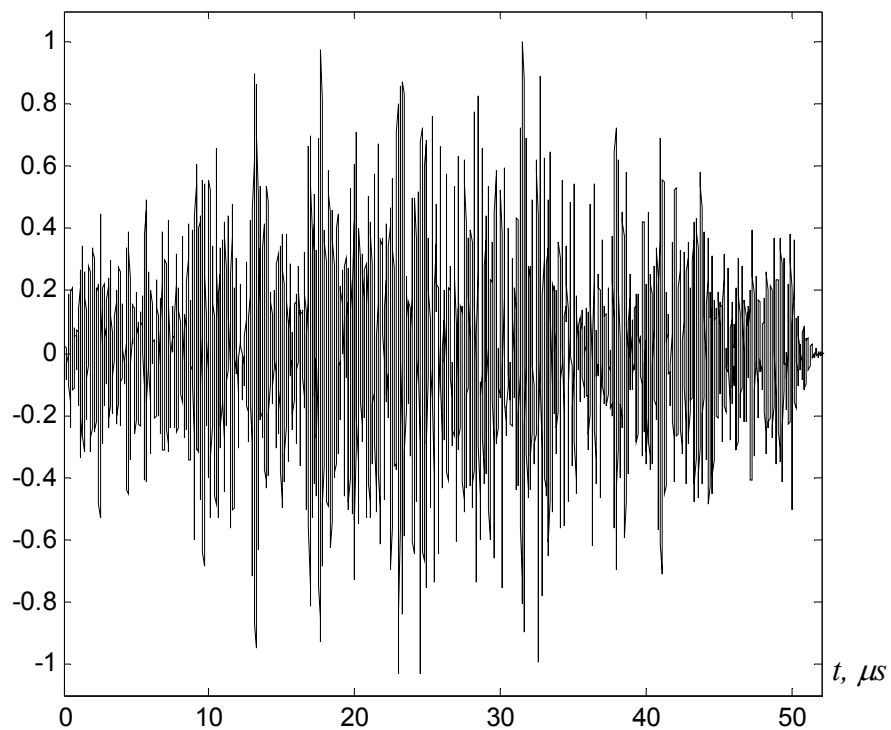
$$S_N / S_T = \frac{\max(A_N)}{\max(A_T)};$$

Kur $\max(A_n)$ – maksimali naudingo signalo amplitudė (0, 3) μs režiuose, o $\max(A_T)$ – maksimali triukšmo amplitudė (5, 48) režiuose. Šiuo atveju $S_N/S_T = 0.01$. Galima teigti, kad tokio silpno signalo su tokiu mažu diskretizavimo žingsniu aptikti neįmanoma dėl per mažo signalas/triukšmas santykiu.

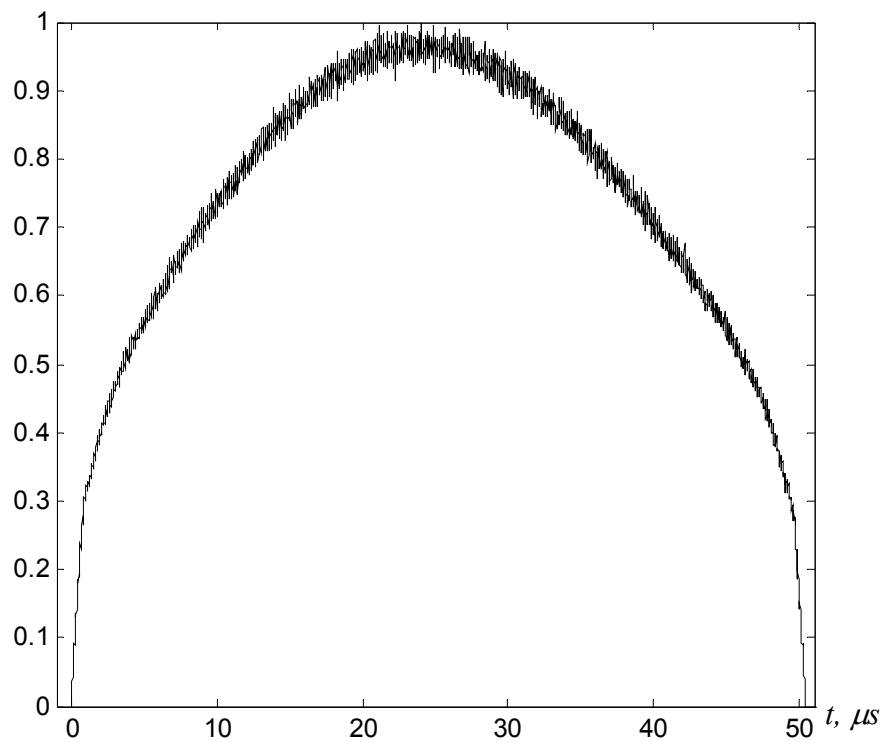
Toliau buvo atlikti visi likusieji bandymai ir jų tarpiniai atspindžio funkcijų bei signalų funkcijos yra pavaizduoti 33 pav. ir 34 pav. prie $N = 30$, 35 pav. ir 36 pav. prie $N = 100$, 37 pav. ir 38 pav. prie $N = 300$, o 39 pav. yra pateikta S_N/S_T priklausomybė nuo N .



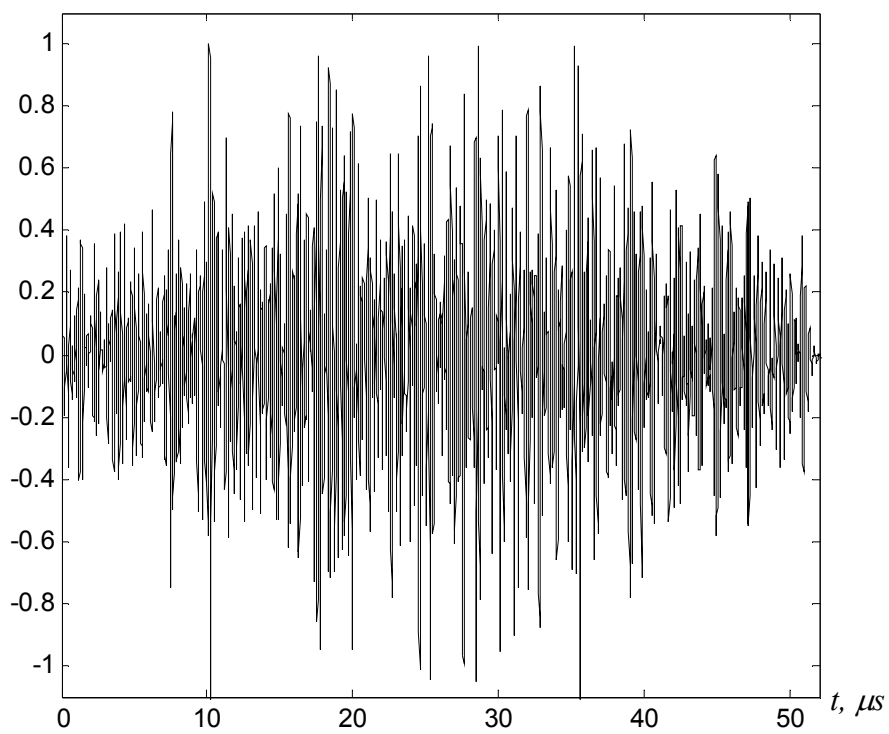
33 pav. Signalo atspindžio funkcija prie $N = 30$



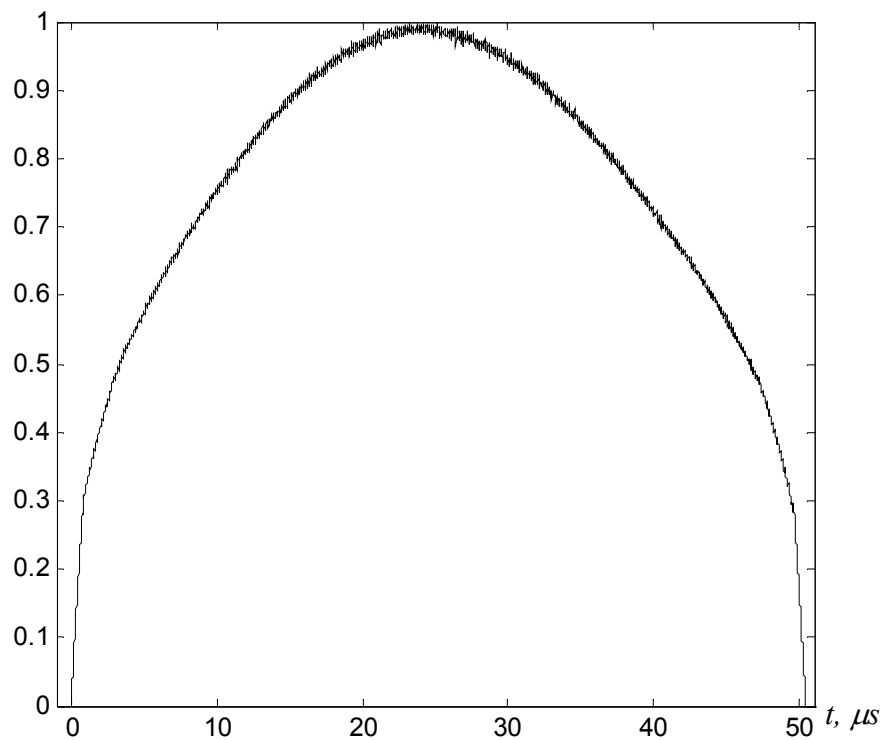
34 pav. Gautas signalas prie $N = 30$. $S_N/S_T = 0,2$



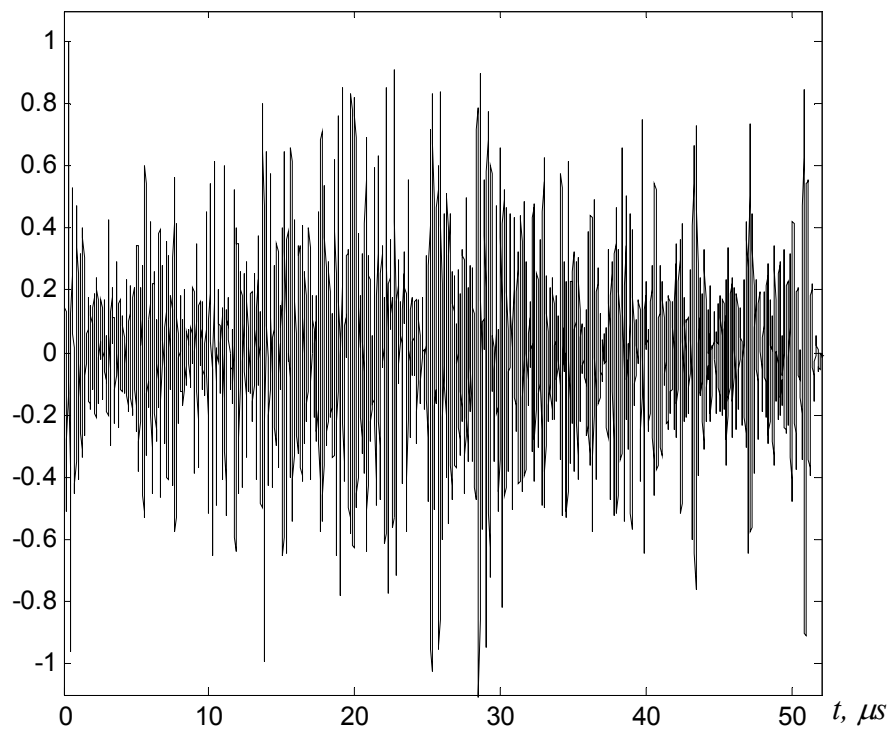
35 pav. Signalo atspindžio funkcija prie $N = 100$



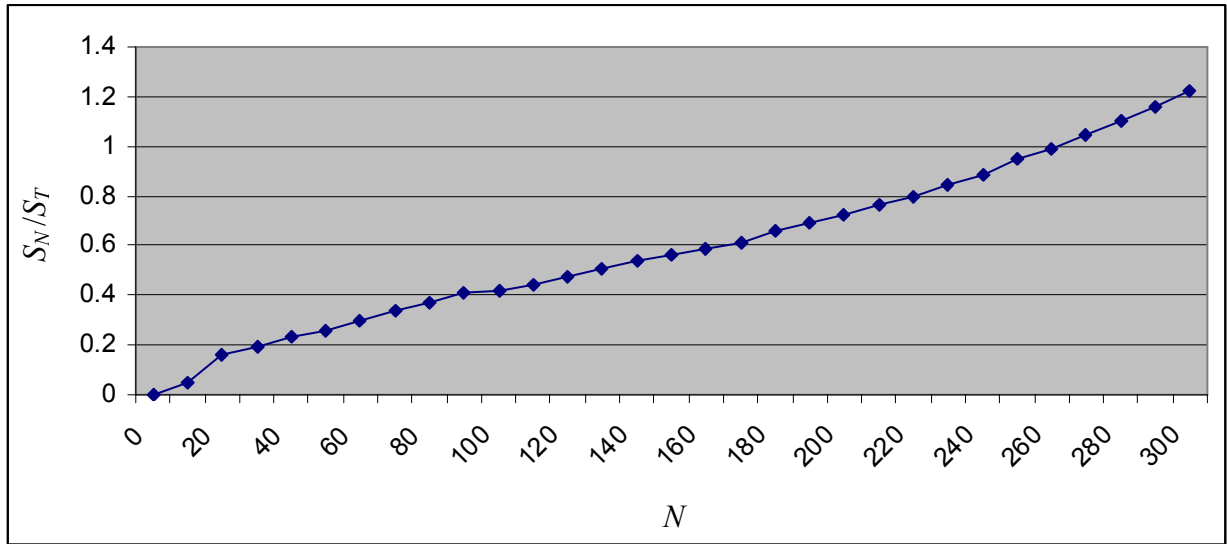
36 pav. Signalo atspindžio funkcija prie $N = 100$ $S_N/S_T = 0.4$



37 pav. Signalo atspindžio funkcija prie $N = 300$



38 pav. Signalo atspindžio funkcija prie $N = 300$ $S_N/S_T = 1.225$



39 pav. Naudingo krašto signalo prie skirtingų N koeficientų ir triukšmo santykis.

Gauti rezultatai parodo, kad net prie $N = 300$ signalas yra sunkiai atskiriamas nuo triukšmo. Akivaizdu, kad N koeficientas turi būti didinamas dar, norint gauti norimą tikslumą, kurį nesunkiai galima numatyti iš aukščiau pateikto grafiko.

Kaip buvo minėta anksčiau (36 psl.), kad šie bandymai buvo atliekami su labai sunkiai apčiuopiamu signalu ir realiai praktikoje jie reikalingi labai retai.

6. Išvados

1. Programos modelio veikimo analizė parodė, kad jo veikimo metu nebuvo valomi pagrindiniai laikino saugojimo masyvai, tuo būdu programos veikimo metu gaunamas pastovus atminties augimas, kuo pasėkoje prie 10 ir daugiau taškų per bangos ilgį programos veikimo nutrukdavo. Ši programos modifikacija turėjo labai žemą matavimų tikslumo ribą;
2. Atminties optimizavimo pasiekoje skaičiavimo greitis tam tikruose skaičiavimo tikslumo intervaluose sumažėjo net 78%, o skaičiavimo ribos prasiplėtė 25%. Pasiūlytas modifikavimo algoritmas iš principo panaikino modelio taikymo ribą dėl atminties perpildymo, be to sumažėjo ir skaičiavimo greitis, apytiksliai 8% lyginant su programa po atminties valymo;
3. Skaidymo žingsnio ir modeliavimo tikslumo priklausomybės tyrimo metu buvo išsiaiškinta, kad labai silpnų signalų atvejuose gali būti nepakankama net imant 300 taškų į bangos ilgį. Buvo nustatyta grafinė priklausomybė, kurios pagalba galima numatyti modelio skaičiavimo laiką norimam tikslumui gauti.

7. Literatūra

1. Kažys, Rymantas Jonas; Voleišis, Algirdas; Mažeika, Liudas; Šitleris, Reimondas; Van Nieuwenhove, R.; Kupshus, Peter; Abderrahim, Hamid Ait. Investigation of ultrasonic properties of a liquid metal used as a coolant in accelerator driven reactors; Elektroninis išteklius // IEEE International Ultrasonic Symposium 2002.
2. ENSMINGER D.; Ultrasonics Fundamentals, Technology, applications. Second Edition, revised and expanded; USA; 580p. ISBN 0-8247-7659-3 ; 1988
3. Liudas Mažeika; Neardomųjų bandymų metodai ir sistemos; Kaunas; 2007
4. Albert S. Birks; Robert E. Green, Jr.; Nondestructive Testing Handbook Second Edition; USA ISBN 0-931403-04-9; 893p; 1991.
5. Albert S. Birks; Robert E. Green, Jr.; Nondestructive Testing Handbook Second Edition; USA ISBN 0-931403-04-9; 893p; 1991.
6. Elektroninis šaltinis; Ultragarsinių metodų MYRRHA tipo reaktoriausvidui vizualizuoti tyrimas, pagrįstas akustiniu kompiuteriniu modeliavimu; http://www.ktu.lt/lt/mokslas/moksliniai_tyrimai/mt2005/bar.pdf [žiūrėta 2008-05-10]
7. Kažys, Rymantas Jonas; Mažeika, Liudas; Jasiūnienė, Elena; Šliteris, Reimondas; Kupshus, P.; Van Nieuwenhove, R.; Abderrahim, H. Ait. Ultrasonic imaging techniques for the visualisation in hot metals; Elektroninis išteklius // Proceedings of the World Congress on Ultrasonics 2003, September 7-10, 2003, Paris, France. Paris, 2003. ISBN 2-9521105-0-6.p. 1391-1394
8. R. Kažys, A. Voleišis, R. Šliteris, L. Mažeika, R. Van Nieuwenhove, P. Kupshus, H. Ait Abderrahim. Ultrasonic transducers for high temperature applications in accelerator driven reactors. In:// Proceedings of the World Congress on Ultrasonics WCU-2003, ISBN 2-9521105-06, Paris, 7-10 September 2003, p. 33-36.
9. MEIRE H.B., FARRANT P.; Basic Ultrasound, John Wiley & Sons, Inc., pp.10-17; 1995
10. Ensminger, D.; M. Dekker; Ultrasonics: Fundamentals, Technology, Applications; New York; 1988.
11. V.Dumbrava, R.Kažys, L.Mažeika, L.Svilainis. "Digital Ultrasonic Imaging System: the Software Concept" Informatica, vol.2, No.4, pp.478-597. 1991.
12. И.Н. Ермолов; Ультразвуковые пьезопреобразователи для неразрушающего контроля; Москва; "Машиностроение"; 1986; 280p

13. Kažys, Rymantas Jonas; Mažeika, Liudas; Šliteris, Reimondas; Jasiūnienė, Elena; Abderrahim, Hamid Ait; Dierckx, Marc. Ultrasonic imaging in the liquid metals // ISEM 2005 : 12ht international symposium on interdisciplinary Electromagnetic, Mechanic and Biomedical Problems, 12-14 Sept. 2005, Bad Gastein, Austria: short paper proceedings / editors: Helmut Pfutzner, Elizabeth Leiss. . Vienna: Vienna Magnetic Group Reports, 2005. ISBN 3-902105-00-1. p. 164-165.
14. Й. Крауткремер; Г. Крауткремер; Ультразвуковой контроль материалов; Москва.; 1991; 751p.
15. PARKER D., LEC R.M., PENDSE H.P., VETELINO J.F.; Ultrasonic Sensor for the Characterisation of Colloidal Slurries; IEEE Ultrasonics Symposium; pp.295-298; 1990
16. von Ramm OT, Smith SW. Real time Volumetric Ultrasound Imaging System; J Digit Imaging; 1990;3:261–66.
17. Elektroninis šaltinis; Floating-Point Numbers :: Data Types (Programming); http://www.system.nada.kth.se/unix/software/matlab/Release_14.1/techdoc/matlab_prog/ch11_st4.html#double_precision_defined [žiūrėta 2008-04-30]
18. Elektroninis šaltinis; MATLAB®; Profiling for Improving Performance; http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab_env/f9-17018.html&http://www.google.com/search?client=safari&rls=en&q=matlab%20profiler%20manual&ie=UTF-8&oe=UTF-8 [žiūrėta 2008-05-01]
19. Elektroninis šaltinis; Wikipedia elektroninė enciklopedija; <http://en.wikipedia.org/wiki/DDR2> [žiūrėta 2008-05-10]
20. Elektroninis šaltinis; PC STATS; <http://www.pcstats.com/articleview.cfm?articleID=1746> [žiūrėta 2008-05-10]
21. R. Kažys, A. Voleišis, R. Šliteris, L. Mažeika, R. Van Nieuwenhove, P. Kupschus, H. Ait Abderrahim. High temperature ultrasonic transducers for imaging and measurements in a liquid Pb/Bi eutectic alloy. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. ISSN 0885-3010, 2005, Vol. 52, No. 4, 525-537.
22. Elektroninis šaltinis; Physics 24/7; <http://www.physics247.com/physics-tutorial/ultrasound-physics.shtml> [žiūrėta 2008-05-01]