



## DVIMAČIŲ IR TRIMAČIŲ GEOMETRINIŲ OBJEKTŲ MAŽESNĖS DIMENSIJOS REPREZENTACIJŲ KONSTRAVIMAS

Lina Laukaitytė

magistrantė

Vilniaus Gedimino technikos universitetas

el. paštas: l.laukaityte@vnmpt.lt

**Anotacija.** Nagrinėjamas dvimačių ir trimačių objektų mažesnės dimensijos reprezentacijų konstravimo uždavinys. Trumpai aptariami naujausi pasiekimai taikymo srityse, tokiose kaip medicina, kompiuterinė grafika ir kt. Supažindinama su svarbiausiomis sąvokomis ir apibrėžimais, naudojamais literatūroje. Aprašomi skeletų tipai ir svarbiausios jų savybės, Voronojaus ir Euklido atstumų skeletavimo metodai, skirti skeleto gavimui. Pateikiami ir aptariami rezultatai, gauti dvimačiams objektams pritaikius kelis skirtingus skeletavimo algoritmus. Analizuojamas matematinis modelis, skeleto viršūnių atstumų apskaičiavimui.

**Reikšminiai žodžiai:** skeletavimas, centrinė daugdara, Voronojaus diagramos, ploninimas, kreivinis /geometrinis skeletas, mažesnės dimensijos reprezentacija, genėjimo algoritmas.

### Įvadas

Objekto mažesnės dimensijos reprezentacija – viena iš daugelio sąvokų, naudojamų literatūroje, įvardijančių skeletą. Pirmą kartą skeleto sąvoka, kaip ploninimo proceso rezultatą pristatė H. Blumas, 1964 metais. Literatūroje taip pat naudojami terminai kaip centrinė ašis (angl. *Medial axis*) ar centrinė daugdara.

Skaitmeninių objektų skeletavimas ir jo istorija yra tokia pat sena kaip ir skaitmeninių vaizdų analizė. Skeletavimo tikslas yra sumažinti dvimačių pavienių objekto vaizdą iki vienmačio linijinio skeleto, išsaugojančio objekto topologiją ir geometrinę informaciją (Borgefors 1999).

Skeleto išskyrimo uždavinių pritaikymas tapo labai svarbus daugelyje mokslo sričių.

Medicinoje naudojamas biomedicinos vaizdų analizėje, kompiuterinėje echoskopijoje ir kt. Prie šios srities inovatyvumo prisideda ir Vilniaus Gedimino technikos universiteto elektronikos sistemų katedros ir matematinio modeliavimo katedros darbuotojai, bendradarbiaujantys su medicinos institucijomis, tarptautinėje programoje „Eureka“ įgyvendino projektą „Žmogaus galvos kraujagyslių aneurizmos srities analizė angiogramose“. Projekte gauti virtualių kraujagyslių skeletai, parinkta metodika virtualių aneurizmų lokalizacijos radimui, pasirinktas būdas aneurizmos srities paviršiaus aprašymui bei aneurizmos srities geometrinių matmenų apskaičiavimui. Taip pat rengiamas projektas „Dešiniojo širdies skilvelio rekonstrukcija iš dvimačių echokardiografijos vaizdų“, kurio tarpinis tikslas – centrinė skilvelio daugdara.

Kompiuterinėje grafikoje taikomas trimačių animacinių personažų judesių kūrimui, rašmenų atpažinimui, kompiuterinėje navigacijoje.

Dažnai skeletas naudojamas tiesiog kaip kompaktiška informacijos saugojimo forma.

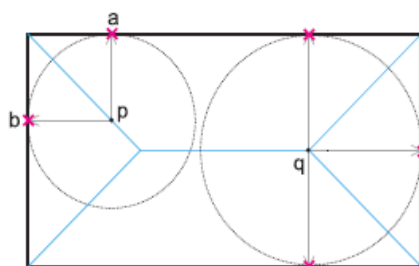
### Skeletų tipai ir jų savybės

Skeletai pagal ypatybes skirstomi į du tipus – geometrinius ir kreivinius.

*Geometrinis skeletas* kitaip dar vadinamas Blumo skeletu tikrąją  $n$ -matę objekto formą sumažina tik viena dimensija, o skeletas apibrėžiamas kaip taškai  $p$ , kurie turi mažiausiai du kraštinius taškus  $a$  ir  $b$ , minimaliu atstumu nutolusius nuo  $p$  (1 pav.).

Taškai  $a$  ir  $b$  yra vadinami ypatingaisiais  $p$  taškais. Taip pat gali būti vadinami kontaktiniais taškais.

Atstumas tarp skeleto taško ir jo ypatingųjų taškų gali būti saugomas kartu su skeletu, nes jis leidžia tikrosios formos rekonstrukciją iš skeleto. Tokia struktūra yra vadinama skeletu arba vidurinėsios ašies transformacija. Svarbi skeleto transformacijos savybė yra jo vienatis: kiekviena forma turi vienintelę skeleto transformaciją.



1.pav. Geometrinis skeletas

*Kreivinis skeletas* yra jungi kreivių aibė ir gali būti matomas kaip formos vaizdavimas šakomis (angl. *stickfiger*). Kreiviniai skeletai neturi plačiai naudojamo apibrėžimo, bet juos galima aprašyti kaip apjungtų centrinių linijų (vienmatė linija, atitinkanti objekto atšakos centrą ir išsaugojanti jos topologiją) aibę.

Kreiviniai skeletai dažnai turi daug geometrinio skeleto savybių, tačiau pagrindinė jų ypatybė yra ta, kad kreivinis skeletas susideda tik iš vienmačių kreivių, o ne dvimačių paviršiaus komponentių (Rosenfeld 2005).



2. pav. Objekto geometrinis (a) ir kreivinis (b) skeletai

Trys svarbiausios savybės, kuriomis turi pasižymėti skeletas:

*Centriškumas* – objekte skeletas yra lokaliai centruotas. Centravimas yra reikalingas tiksliai objekto rekonstrukcijai. Kartais centriškumas apibrėžiamas taip, kad objekto skeletas privalo būti centrinės ašies poaibių. Žinoma, vien tokios sąlygos nepakanka centriškumui apibūdinti. Skeleto centriškumą galima išivaizduoti taip: jei statmename skelete kryptį objekto pjūvyje, nubrėztume daug spindulių jungiančių skeleto tašką su objekto kontūru vienoje tiesėje esančių spindulių ilgai privalėtų sutapti (Rokicki 2010).

*Homotopiškumas* – objekto topologijos išsaugojimas. Jeigu figūrą sudaro tik viena jungi komponentė, tai homotopija reiškia, kad ir skeletas sudarytas iš vienos jungios komponentės (t.y. objektas ir jo skeletas turi vienodą atšakų, tunelių ir kiaurymių kiekį).

*Plonumas* – kreiviniams skeletams galime reikalauti, kad skeletas būtų vienmatis, išskyrus išsišakojimo vietas. Geometriniais skeletams - skeletas yra plonas autentiškos formos atvaizdavimas, tai yra, jis turi viena dimensija mažiau nei tikrasis objektas. Trimatėms formoms ir taškams, kuriuose susitinka šakos ir juostos (angl. *sheets*) apibrėžti plonumą nėra triviale. Dažnai yra sakoma, kad plonumo savybė prieštarauja rekonstruojamumui. Vietoj to, kad reikalautume plonumo visur, galime reikalauti, kad skeletas būtų 1-vokselio storumo, tik kai tai neprieštarauja rekonstruojamumo savybei (Rosenfeld 2005).

Skeletų tipų papildomų savybių priklausomybė pavaizduota 1 lentelėje.

1. Lentelė. Papildomos objektų skeletų savybės

Papildomos objektų skeletų savybės	Geometrinis skeletas	Kreivinis skeletas
<i>Daugiaskalis atvaizdavimas</i> (skirtingų struktūrų vaizdavimas skirtingose skalėse)	⊕	□
<i>Efektyvus skaičiavimas</i> (efektyvus laiko ir atminties (kompiuterio) prasmė)	⊕	□
<i>Glotnumas</i> (antrosios išvestinės tolydumas kiekviename skeleto taške)	□	⊕
<i>Lengvas įgyvendinimas</i> (paprastas skeletavimo metodo įgyvendinimas)	⊕	□
<i>Matomumas</i> (kiekvienas krašto taškas yra matomas iš skeleto taško)	□	⊕
<i>Mazgų aptikimas</i>	□	⊕
<i>Robustiškumas</i> (angl. <i>Robustness</i> ) (Atsparumas „triukšmui“)	⊕	□

- ⊕ - Pasižymi papildomomis savybėmis;
- - nepasižymi papildomomis savybėmis;

## Skeletavimo algoritmai

Yra sukurta daug skeleto išskyrimo algoritmų. Algoritmus, įgyvendinančius skeletavimą, galima suskirstyti į tokias grupes (žinoma tai nėra vienintelis klasifikavimo būdas (Žitkevičius 2007)):

*Iteratyvus ploninimo.* Ploninimo proceso metu skaitmeninės formos suploninamos, išlaikant jų topologiją. Ploninimo metodai egzistuoja tiek kreiviniams tiek geometriniams skeletams. Pagrindinis šios grupės algoritmų principas yra paprastųjų taškų indentifikavimas ir naikinimas. Paprastojo taško panaikinimas neįtakoja objekto topologijos. Ploninimo iteracijos atliekamos tol, kol objekte nelieka paprastųjų taškų. Pagrindinis šių algoritmų privalumas yra greitis, nes paprastojo taško nustatymui pakanka išnagrinėti jo 3x3 kaimynystę (3.pav.). Pagrindinė šių algoritmų grupės problema yra galūnių nuėsdinimas (todėl įvedamos papildomos sąlygos joms išsaugoti).

$a_3$	$a_2$	$a_1$
$a_4$	$(i, j)$	$a_0$
$a_5$	$a_6$	$a_7$

3. pav. Paprastojo taško 3x3 kaimynystė

*Geometriniai metodai* dažniausiai naudojami tokiais atvejais kai objekto paviršius pateikiamas naudojant išsklaidytą taškų aibę arba paviršiaus tinklą. Šie taškai panaudojami Voronojaus diagramos konstravimui, t.y. Voronojaus ląstelių sukūrimui. *Voronojaus diagrama* - pavadinta Georgo Voronoi garbei - ypatinga metrinės erdvės, apibrėžtos atstumais iki nurodytos diskrečios objektų erdvėje aibės, skaidymo rūšis. Kiekviena Voronojaus ląstelės siena yra nutolusi vienodu atstumu nuo ją konstruojančių tinklelio viršūnių ar taškų. Vidinės Voronojaus briaunos gali būti naudojamos geometrinio skeleto aproksimavimui. Norint išskirti vienmatį skeletą yra būtini papildomi ēsdinimo žingsniai (Ogniewicz 1992).

*Atstumų lauko metodai.* Sudaromi kai kiekvienam  $O$  objekto  $P$  taškui yra priskiriama atstumo reikšmė iki artimiausio  $B(O)$  paviršiaus  $Q$  taško:

$$d(P)_{P \in O} = \min_{Q \in B(O)} (d(P, Q)) \quad (1)$$

čia  $d(P, Q)$  yra pasirinkta metrika.

Apskaičiavus atstumų lauką, bandoma nutiesti kelią per taškus, kuriuose atstumas iki objekto ribų yra maksimalus. Būtent šie taškai ir atitinka objekto skeletą. Taškų apjungimas vykdomas taikant gradientinę paiešką arba grafų teoriją (ieškant trumpiausio kelio tarp pasirinktų maksimumų). Dažnai papildomai naudojami ploninimo metodai. Kitas sprendimo būdas yra palikti tik nedidele tarpusavyje neapjungtu maksimaliai nutolusių nuo paviršiaus taškų aibę ir po to juos iš naujo apjunginėti (Rokicki 2010). Pagrindinis šių algoritmų privalumas yra tai, kad atstumu laukas yra greitai apskaičiuojamas. Vienas iš šių metodų – Euklido atstumų diagramos. Šiame metode skaičiuojamas kiekvieno pikselio atstumas iki krašto ir bandoma išgauti skeletą, ieškant pikselių objekto centre; Šie pikseliai yra toliausiai nutolę nuo centro. Atstumo kodavimas pagrįstas Euklido atstumu. Deja, nėra garantijos, kad rezultatas išsaugos jungumą. (Daya 2008)

## Rezultatai

Pagal anksčiau straipsnyje aptartus skeletavimo algoritmus ir skeletų savybes įvertiname algoritmų kokybę ir efektyvumą.

Vertinant kokybę pasirinkome standartines geometrines figūras, nes intuityviai galime nuspėti teisingą galinį rezultatą, t.y. skeletą.

Iš gautų rezultatų (1. Lentelė) matome, kad gauti skeletai skiriasi:

- Ploninimo algoritmų gauti skeletai pasižymi galūnių nuėsdinimu;
- Voronojaus diagramų skeletai - pertekline informacija;
- Voronojaus diagramų su genėjimo algoritmų skeletai vaizduojami su svertinėmis atšakomis;
- Euklido atstumų diagramos – nejungumų.

### 1. Lentelė. Standartinių objektų skeletai

Binarinis vaizdas Skeletavimo metodai				
Ploninimas				
Voronojaus diagramos				
Voronojaus diagramos+ Genėjimo algoritmas				
Euklido atstumų diagramos				

Algoritmų efektyvumą vertinsime pagal programų greitį (sekundėmis) sprendžiant skeleto išgavimo uždavinį. Pasirinkome trijų skirtingų dydžių paveikslėlius (2. Lentelė).

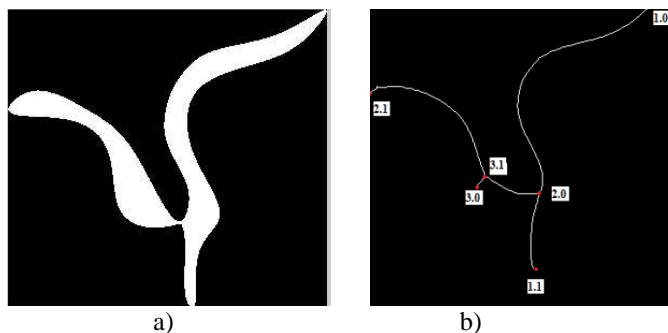
### 2. Lentelė. Skeletavimo algoritmo realizacijos laikas

	Žvaigždė 420x429(pikseliai)	Žirafa 554x565(pikseliai)	Indėnas 625x638(pikseliai)
Ploninimo algoritmas	3,4530	5,3250	8,4320
Voronojaus diagramos	2,1410	4,3590	3,8750
Voronojaus diagramos + Genėjimo algoritmas	10,8440	22,3440	46,3280
Euklido atstumų diagramos	4,4220	2,5310	12,7500

Pagal gautus rezultatus matome, kad jų užduoties įvykdymo laikas mažai skiriasi. Galima išskirti tik Voronojaus diagramų su genėjimo algoritmu, tačiau tai pateisinama – dviejų algoritmų pritaikymu. Šį metodą galima patobulinti, efektyvumo naudai.

Gavę skaitmeninio objekto skeletą, galima jam sukonstruoti matematinį modeliavimo uždavinį. Pateikiame pavyzdį – dvimačio objekto skeleto atstumų tarp taškų skaičiavimas.

Sukonstruotam dvimačiam objektui (4 pav. a)) pritaikę ploninimo algoritmą gavome vienmatį, išskyrus atšakų susijungimo taškus, skeletą (4 pav. b)). Šiam skeletui sukonstravome šakų ilgių atstumų skaičiavimo uždavinį. Kadangi nekėlėme reikalavimų judėjimo skeleto šakomis, krypčiai, gauname simetrinę atstumų matricą kaip rezultatą (4. Lentelė). Taigi nesunkiai sukonstravome uždavinį, kurį galime pritaikyti kompiuterinei navigacijai.



3. pav. a) Dvimatis sukonstruotas objektas b) Dvimačio objekto skeletas

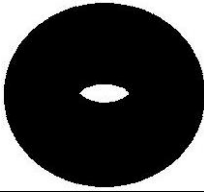


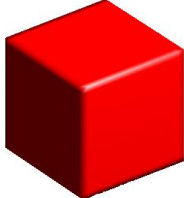
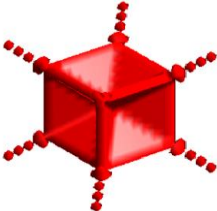
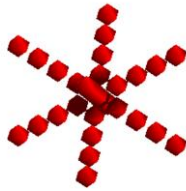
4. Lentelė. Skeleto šakų atstumų rezultatai (pikseliais)

		Šakos pradžios koordinatės					
		Mazgas (1.0)	Mazgas (1.1)	Mazgas (2.0)	Mazgas (2.1)	Mazgas (3.0)	Mazgas (3.1)
Šakos pabaigos koordinatės	Mazgas (1.0)	-	336	258	480	324	313
	Mazgas (1.1)	336	-	78	300	144	133
	Mazgas (2.0)	258	78	-	222	66	55
	Mazgas (2.1)	480	300	222	-	178	167
	Mazgas (3.0)	324	144	66	178	-	11
	Mazgas (3.1)	313	133	55	167	11	-

Pateiksime trimačių objektų skeletavimo pavyzdžių.

Trimačiams objektams – kubui ir torui - pritaikytas Euklido atstumų diagramų skeletavimo algoritmas. Programa realizuota programine įranga Matlab. Gautus rezultatus pavaizduojame 5. Lentelėje.

5. Lentelė. Trimačių objektų skeletavimas

Trimatis objektas	Objektas ploninimo procese	Objekto skeletas
Trimatis toras		
		
Trimatis kubas		
		

Skeleto išskyrimo problema iš trimačių objektų dar nėra iki galo išspręsta. Tai aktualus šių dienų uždavinys.

## Išvados

Sukurta labai daug metodų ir algoritmų išgauti skaitmeninio objekto skeletui. Daugelis konstruojamų skeletų neatitinka visų jiems būdingų savybių, tačiau ir ne visuomet tai yra reikalaujama, priklauso nuo uždaviniui keliamų tikslų.

Straipsnyje aptartų skeletavimo algoritmų realizavimo laikas yra pakankamai vienodas ir trumpas, todėl galime teigti, kad šie algoritmai yra efektyvūs, o pasirinkti geriausią algoritmą rekomenduojama, atsižvelgiant į uždaviniui keliamus reikalavimus:

1. Iteratyvaus ploninimo algoritmai tinka uždaviniams, kurių tikslas gauti greitai skeletą, išlaikantį objekto topologiją.
2. Geometrinio metodų algoritmai efektyviausi kai skaitmeninių objektų paviršiai pateikiami naudojant išsklaidytą tašku aibę arba paviršiaus tinklą.
3. Atstumų lauko metodų algoritmai tinkamiausi, greitam, bet nebūtinai jungiam, skeletų gavimui.

## Naudota literatūra:

- Borgefors, G. 1999. Computing skeletons in three dimensions, *Journal of the Pattern Recognition* 32: 1225.
- Cornea, N. D.; Min, P. 2007. Curve-skeleton properties, applications, and algorithms, *IEEE Transactions on Visualization and Computer Graphics* 13 (3): 530-548.
- Daya, B. 2008. Parallelization of Two-Dimensional Skeletonization Algorithms, *Journal of Undergraduate Research* 9 (4): 2-3.
- Kruszynski, K. J.; van Liere, R.; Kaandorp, J. 2005. Quantifying differences in skeletonization algorithms: A case study, in Proc. *Of Visualization, Imaging, and Image Processing (VIIP 2005)*, ACTA PRESS.
- Ogniewicz, R. L. 1995. Automatic Medial Axis Pruning by Mapping Characteristics of Boundaries Evolving under the Euclidean Geometric Heat Flow onto Voronoi Skeletons. *Technical report*.
- Rokicki, J. 2010. Skeletavimo metodų apžvalga, iš konferencijos „Mokslas-Lietuvos ateitis“ pranešimų medžiaga, 19-22.
- Rosenfeld, A. 2005. Digital Picture Processing, *Academic Press, New York, Usa: 11-28*.
- Žitkevičius, E.; Grigaitis, D.; Navakas, D. 2007. On skeletonization of blood vessels in angiographic MRI images of human brain, *Information Technology and Control* 36(4): 372.

## CONSTRUCTION OF LOW DIMENSION REPRESENTATION OF 2-D AND 3-D GEOMETRICAL OBJECTS

L. Laukaitytė

### Summary

The problem of construction of the lower dimensions representations' extracted from the two-dimensional and three-dimensional objects is presented. Recent developments in fields of application, such as medicine, computer graphics, and others are briefly discussed. The main concepts and definitions are introduced. The types and main features of skeletons are described. Skeletonization types, like Voronoi and Euclidean distances map, used for the extraction of the skeleton, are discussed widely. Results, obtained by applying few different skeletonization algorithms for the two-dimensional objects, are presented and discussed. The mathematical model developed for calculation of the distance of the skeleton vertexes is analyzed.

**Keywords:** skeletonization, medial axis, Voronoi-diagram, thinning, curve/geometry skeleton, pruning method.

# MATEMATINIS MODELIAVIMAS

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“,  
įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

## TRANSPORTO UŽDAVINIO SU FIKSUOTOMIS IŠLAIDOMIS SPRENDIMAS

**Aida Vaičiulytė**

*Vilniaus Gedimino Technikos Universitetas*

[aida.vaiciulyte@gmail.com](mailto:aida.vaiciulyte@gmail.com)

**Anotacija.** Straipsnyje nagrinėjamas euristinis transporto uždavinio su fiksuotomis išlaidomis sprendimo metodas, kuris susideda iš Lagranžo relaksacijos, Šakų ir rėžių metodo ir euristikų taikymo. Metodas pagrįstas ekvivalentaus pagrindiniam uždaviniui, kuriame yra tik dalis visų kintamųjų aibės, uždavinio sprendimu. Algoritmas susideda iš trijų dalių: pirmoje dalyje taikant Lagranžo relaksaciją arba Lagranžo dekompoziciją, randamas apatinis rėžis ir sumažinamas išlaidų kintamųjų skaičius; antroje - panaudojant pirmame žingsnyje gautus rezultatus, sudaromas vienas arba keletas uždavinių su mažesniu skaičiumi kintamųjų; trečioje dalyje taikant standartinį šakų ir rėžių metodą pagerinamas antroje dalyje rastas geriausias sprendinys ir gaunamas galutinis sprendinys.

### **Įvadas**

Literatūroje daug dėmesio skiriama standartinio transporto uždavinio sprendimo nagrinėjimui, tačiau to negalima pasakyti apie didelės apimties transporto uždavinį su fiksuotomis išlaidomis (toliau – TUFİ), kurio tikslas – rasti maršrutų kombinacijas, kurios sumažina bendrą kintamųjų skaičių ir fiksuotas išlaidas, kol yra tenkinamas kiekvieno tiekėjo ir gavėjo pasiūlos ir paklausos poreikis.

Nepaisant uždavinio panašumo į standartinį transporto uždavinį, TUFİ išspręsti yra sunkiau dėl tikslo funkcijos netolydumo (įtraukiamos netolydžios fiksuotos išlaidos). Literatūroje nagrinėjami algoritmai, kurie sprendžia TUFİ tiksliai arba apytiksliai. Hirsh ir Dantzig (1968 m.) pirmieji pradėjo nagrinėti TUFİ ir įrodė, kad šio

uždavinio optimalus sprendinys egzistuoja leistinų sprendinių srityje, ir todėl optimalaus sprendinio paieška apsiriboja šios srities ekstremumais. Dalis algoritmų tiek tikslių, tiek euristinių yra paremti šia savybe (pvz. lokaliuos paieškos metodai, Steinberg (1970 m.) ir Walker (1976 m.), absoliučių taškų (angl. *absolute points*) paieškos metodas, Adlakha ir Kowalski (1998), „Daugiau-už-mažiau“ (angl. *More-for-less*) metodas, Adlakha, Kowalski, Vemuganti, Lev (2004)). Kita dalis algoritmų yra paremti Lagranžo euristikų metodu, Wright (2003), paieškos su draudimais procedūra (angl. *Tabu search heuristic procedure*), Sun (1998). Tiksliųjų algoritmų atžvilgiu TUFİ sprendimui labiausiai pritaikytas šakų ir rėžių metodas. Tačiau šie algoritmai nėra dažnai naudojami, išskyrus mažos apimties uždavinius.

Straipsnyje nagrinėjamas euristinis TUFİ uždavinio sprendimo pritaikymas, kuris sujungia keletą metodų ir padeda efektyviau rasti optimalų TUFİ sprendinį. Taigi sprendžiamam uždaviniui bus pritaikyta Lagranžo relaksacijos (Lagranžo dekompozicija), Šakų ir rėžių metodas bei euristikos. Metodas yra pagrįstas ekvivalentaus pagrindiniam uždaviniui, kuriame yra tik dalis visų kintamųjų aibės, sprendimu. Sprendimo algoritmas susideda iš trijų dalių:

Pirmoje dalyje, taikant Lagranžo relaksaciją arba Lagranžo dekompoziciją, randamas apatinis rėžis bei sumažinamas išlaidų kintamųjų skaičius. Šioje dalyje neieškomas optimalus sprendinys, nes uždavinio forma yra per daug sudėtinga.

Antroje dalyje iš prieš tai rastų išlaidų kintamųjų bus sudaromas vienas arba keletas uždavinių su tokia pačia struktūra kaip pirminio uždavinio, bet su mažesniu skaičiumi kintamųjų. Šiems uždaviniams dar kartą bus pritaikoma Lagranžo relaksacija arba Lagranžo dekompozicija ir geriausias euristinis sprendinys išsaugomas. Šioje dalyje sprendinys randamas tik taikant Lagranžo metodus.

Trečioje dalyje, taikant standartinį šakų ir rėžių metodą su ribotu laiku antro žingsnio uždaviniui, pagerinamas jau rastas geriausias sprendinys, ir gaunamas galutinis sprendinys.



## Uždavinio matematinis modelis

Transporto uždavinio su fiksuotomis išlaidomis matematinis modelis:

$$\min f(x) = \left\{ \sum_{i=1}^m \sum_{j=1}^n (c_{ij}x_{ij} + f_{ij}y_{ij}) \right\}, \quad (1)$$

kai

$$\sum_{j=1}^n x_{ij} = s_i \quad (2)$$

$$\sum_{i=1}^m x_{ij} = d_j \quad (3)$$

$$u_{ij}y_{ij} - x_{ij} \geq 0 \quad (4)$$

$$x_{ij} \geq 0 \quad (5)$$

$$y_{ij} = 0 \text{ arba } y_{ij} = 1 \quad (6)$$

ir  $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ .

Čia  $u_{ij} = \min \{s_i, d_j\}$ , kai  $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ ;

$i = 1, 2, \dots, m$  yra gamybos punktai;

$j = 1, 2, \dots, n$  produkto vartotojai;

$x_{ij}$  - produkto kiekis, transportuojamas iš  $i$ -tosios įmonės  $j$ -tajam vartotojui;

$c_{ij}$  - kintamosios išlaidos produkto vienetui, transportuojamo iš  $i$ -tosios įmonės  $j$ -tajam vartotoju;

$s_i$  - visas produkto kiekis esantis  $i$ -tajame gamybos punkte;

$d_j$  - visas produkto kiekis, reikalingas  $j$ -tajam vartotojui;

$f_{ij}$  - pastovios išlaidos, susijusios su įmonėje  $i$  gaminamų produktų transportavimu. Tai gali būti mokesčiai už transporto priemonių nuomą, kelio mokesčiai.

## Lagranžo relaksacija

Šioje dalyje TUFİ pritaikoma standartinė sveikojo programavimo procedūra – Lagranžo relaksacija. Metodo principas – supaprastinti uždavinio sprendimą, pašalinus vieną iš apribojimų. Relaksuoto uždavinio sprendinių aibė yra didesnė už pradinio uždavinio sprendinių aibę, t.y. pradinio uždavinio sprendinių aibė yra relaksuoto uždavinio sprendinių aibės poaibis. Pagrindinis uždavinio relaksacijos tikslas – rasti tikslo funkcijos reikšmės apatinį rėžį. Viršutiniam tikslo funkcijos reikšmės rėžiui rasti sprendžiamas dualus uždavinys (Guignard 2003).

## Lagranžo dekompozicija

Lagranžo dekompozicija yra atskiras Lagranžo relaksacijos atvejis, kuris naudojamas tuomet, kai uždavinį galima skaidyti į keletą uždavinių su bendrais kintamaisiais, bet kuriuos lengviau spręsti atskirai.

## Antrojo etapo uždavinys

Pirmame etape išsprendus dualų uždavinį Lagranžo relaksacijos atveju gautas optimalus daugiklis  $\overline{\lambda_{ij}}$ , Lagranžo dekompozicijos atveju -  $(\overline{\lambda_{ij}}, \mu_{ij})$ . Remiantis šiais daugikliais sukonstruojamas mažesnės apimties uždavinys, kuriame apibrėžiami galimi optimalaus sprendinio kintamieji.

Išlaidos porai  $(i, j)$  apibrėžiamos:

$$\pi_{ij} = f_{ij} - u_{ij} \overline{\lambda_{ij}} + F(c_{ij} + \overline{\lambda_{ij}}),$$

kur  $F$  - daugiklis skirtas kintamųjų ir fiksuotų išlaidų kompensavimui.  $\pi$  reikšmė yra esminė, kad pagrindinis uždavinys būtų efektyvus ir turi būti parinkta taip, kad uždavinys būtų išsprendžiamas ir neturėtų per daug kintamųjų.

Nepaisant pasiekto sumažinimo tokios formos uždavinys tiksliais algoritmais neišsprendžiamas, todėl naudojami euristiniai algo-

ritmai. Dar kartą taikoma Lagranžo relaksacija arba Lagranžo dekompozicija antrojo etapo uždaviniui tam, kad būtų gautas šio etapo geriausias euristinis sprendinys. Dėl aibių retumo, Lagranžo relaksacijos arba Lagranžo dekompozicijos sprendimas yra daug greitesnis nei viso uždavinio.

## Paprastosios euristikos

Pirmajame etape pradinis uždavinys buvo skaidomas taikant Lagranžo relaksacijos arba Lagranžo dekompozicijos metodą, tačiau geriausio sprendinio nebuvo ieškoma. Šiuo metodu buvo stengiamasi tik sumažinti išlaidų kintamųjų skaičių. Gerų sprendinių ieškoma antrame etape, kada kiekvienam sumažintam uždaviniui yra taikomas tas pats Lagranžo relaksacijos arba Lagranžo dekompozicijos metodas. Kiekvienam šiam uždaviniui yra gaunami geri, bet ne optimalūs sprendiniai. Iš šių sprendinių jau yra paprasta surasti bendrą pradinio uždavinį sprendinį. Bendras sprendinys, iš galimų sprendinių, randamas pritaikius lokalsiosios paieškos euristiką.

## Bendra procedūra

Taigi bendra uždavinio sprendimo procedūra:

**Ivestis:**  $K, N, F_1, \dots, F_N, T\_BB$

**Išvestis:**  $(x^*, y^*), Z^*$

**Pirmas etapas:** Lagranžo relaksacija

1.1. Taikant aprašytą euristiką, sprendžiamas tiesinis transporto uždavinys ir inicializuojami  $(x^*, y^*)$  ir  $Z^*$ .

1.2. Norint gauti optimalius daugiklius  $\bar{\lambda}_{ij}$ , į kuriuos įeina apatinis režis LB, Lagranžo dekompozicija ar Lagranžo relaksacija pradiniam uždaviniui pradedama taikyti nuo viršutinio režio  $UB = Z^*$ .

**Antras etapas:** Pagrindinis uždavinys

2.1. For  $k = 1, \dots, N$  do:

2.1.1. Iš daugiklio  $F_k$  apibrėžiame išlaidų porą  $(i, j)$  kaip  $\pi^k_{ij} = f_{ij} - u_{ij} \overline{\lambda_{ij}} + F_k(c_{ij} + \overline{\lambda_{ij}})$

2.1.2. Pritaikomas dvejetainės paieškos algoritmas režio  $\pi^k$  radimui.

2.1.3. Pagrindinis uždavinys apibrėžiamas susiejant jį su porų aibe  $S_k = \{(i, j) : \pi^k_{ij} < \pi^k\}$

2.1.4. Pagrindiniam uždaviniui taikoma Lagranžo relaksacija arba Lagranžo dekompozicija ir gaunamas euristinis sprendinys  $(x^k, y^k)$  ir tikslo funkcijos reikšmė  $z^k$ .

2.1.5. Pritaikoma lokališios paieškos euristika dabartiniam sprendiniui ir jis pervadinamas  $(x^k, y^k)$  ir  $z^k$ .

2.1.6. Jei  $z^k < z^*$  tada realizuojama  $z^* = z^k$  ir  $(x^*, y^*) = (x^k, y^k)$ .

2.2. Išsaugomas tas uždavinys, kuris davė geriausią sprendinį.

**Trečiasis etapas:** Šakų ir režių metodas

3.1. Antrajame etape išsaugotam uždaviniui, davusiam geriausią sprendinį, pritaikomas Šakų ir režių algoritmas pradedant nuo sprendinio  $(x^*, y^*)$  ir su ribotu laiku. Dar kartą pervadinam galutinį rastą sprendinį kaip  $(x^*, y^*)$  ir tikslo funkcijos reikšmę  $z^*$

## Literatūra

- Adlakha, V.; Kowalski, K. 2003. A simple heuristic for solving small fixed-charge transportation problems, *Omega* 31: 205-211.
- Adlakha, V.; Kowalski, K., Vemugantia, R.R., Lev, B. 2007. More-for-less algorithm for fixed-charge transportation problems, *Omega* 35: 116-127.
- Aguado, J. S. 2008. Fixed Charge Transportation Problem, Springer Science+Business Media.
- Guignard, M. 2003. Lagrangean relaxation, TOP 11: 151-228.
- Martello, S.; Toth, P. 1990. Knapsack problems. New York: Wiley.

# **FIXED CHARGE TRANSPORTATION PROBLEM**

**A. Vaičiulytė**

## **Summary**

In this paper is presented a heuristic approach, which composed of three phases: Lagrangean relaxation, the core problem and branch and bound algorithm. The method is based on an equivalent solution of the problem, where is only a part set of all variables. In the context of Lagrangean relaxation, different relaxations are considered, which leads two types of problem: continuous transportation problem and binary knapsack problems.

# MATEMATINIS MODELIAVIMAS

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“,  
įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

## PAPRASTO SKYLĖTO DAUGIAKAMPIO SKAIDYMO ALGORITMAI

**Danas Motiejauskas**

*Vilniaus Gedimino technikos universitetas*

*El. paštas [danas.motiejaukas@gmail.com](mailto:danas.motiejaukas@gmail.com)*

**Anotacija.** Nagrinėjama paprasto skylėto daugiakampio skaidymo į neviršijančius nustatyto dydžio komponentus problema. Apibrėžiamas uždavinys ir jo svarba. Apžvelgiami egzistuojantys skaidymo algoritmai, padedantys išspręsti uždavinį, bei jų realizacijos. Pateikiamos trianguliacijos ir padalinimo į apytiksliai iškilius daugiakampius algoritmų modifikacijos, jų privalumai ir trūkumai. Palyginami modifikuoto trianguliacijos algoritmo ir FME PĮ pakete realizuoto “Chopper” algoritmo rezultatai ir vykdymo trukmės.

### **Įvadas**

Šiuolaikinės geografinės informacijos sistemos (GIS) leidžia kaupti, redaguoti ir platinti labai didelio tikslumo geografinius duomenis. Didelė dalis objektų yra saugoma daugiakampių pavidalu (pvz. įvairių teritorijų ribos, miškai, ežerai, ir pan.). Kadangi duomenys yra didelio tikslumo, daugiakampį sudarančių viršūnių kiekis gali būti labai didelis ir siekti net kelis milijonus. Tačiau kai kurie duomenų saugojimo formatai ir programos, skirtos darbui su geografiniais duomenimis, turi tam tikrų apribojimų – tai nesugebėjimas nuskaityti ir interpretuoti skyles daugiakampyje bei vieno objekto viršūnių kiekio apribojimai. Skylė daugiakampyje aprašoma kaip papildomas vienas ar keletas kontūrų. Jei programa ar duomenų saugojimo formatas neturi galimybės nuskaityti arba įrašyti skylių kontūrus – prarandami duomenys arba objektas yra neteisingai atvaiz-

duojamas. Tas pats atsitinka kai ribojamas objekto viršūnių kiekis. Pavyzdžiui, bandome išsaugoti daugiakampį iš 1000 viršūnių formatu, kuris palaiko tik 256 viršūnes, arba tą patį daugiakampį atvaizduojame programa, kuri iš duomenų rinkmenos nuskaito daugiausiai 500 viršūnių. Abiem atvejais prarandame informaciją.

Šių problemų sprendimas – daugiakampio skaidymas į mažesnes dalis. Toks suskaidymas taip pat gali supaprastinti sudėtingas duomenų apdorojimo operacijas. Viena iš dažniausiai pasitaikančių problemų – vykdant skaičiavimus pilnai užpildoma kompiuterio operatyvinė atmintinė ir operacinė sistema dalį duomenų iš jos nukelia į kietąjį diską, o tai gali ilgai užtrukti. Taip atsitinka vykdant sudėtingus skaičiavimus ir kai duomenys yra vienas, bet labai didelis daugiakampis. Suskaidžius jį į mažesnes dalis kiekviena dalis analizuojama atskirai, dėl to yra mažesnės atminties sanaudos, ir operatyvinė atmintinė jau nebeužpildoma iki tos ribos, kai ją reikia atlaisvinti nukėlus duomenis į kietąjį diską.

## **Uždavinio formulavimas**

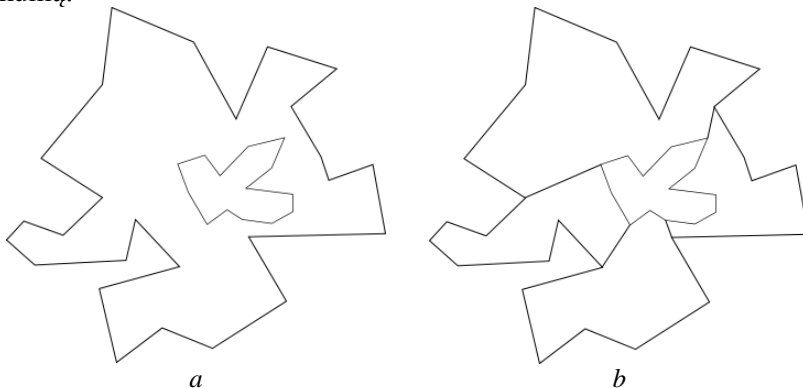
Turime paprastą daugiakampį (1 pav. a). Jį reikia sudalinti į paprastus daugiakampius taip (1 pav. b), kad:

- juose nebūtų skylių,
- juos sudarančių taškų kiekis neviršytų tam tikro nustatyto skaičiaus  $V_{\max}$ ,
- dalių būtų kuo mažiau.

Paprastas daugiakampis yra kontūrų aibė, kur  $\partial P_0$  yra išorinis kontūras, o likę – skylių kontūrai. Daugiakampis yra paprastas, jei jokios nesusijungiančios briaunos nesikerta. Daugiakampius, esančius skylėse, traktuosime kaip atskirus daugiakampius, ir juos nagrinsime atskirai.

Daugiakampis nebūtinai yra iškilus, dėl to iškyla sunkumų. Dalinti bet kaip į dalis, kurias sudarančių taškų kiekis yra lygus nustatytam, negalime, nes galime padalinti į daugiau nei dvi dalis. Be to,

skylių kontūras irgi gali būti sudėtingas, kas taip pat sukelia sunkumų.



1 pav. *a* – paprastas daugiakampis su skylėmis, *b* – galimas padalinimas

## Sprendimo algoritmai

Kadangi daugiakampis nebūtinai yra iškilus, galime pasinaudoti algoritmais, kurie suskaldo daugiakampį į iškilus daugiakampius. Iškilą daugiakampį padalinti į mažesnius atsižvelgiant į anksčiau aptartas sąlygas yra labai paprasta. Tačiau skaidymo į iškilus daugiakampius algoritmai yra ganėtinai sudėtingi –  $O(n^3)$  sudėtingumo. [4]. Be to, gauti iškilieji daugiakampiai yra pakankamai maži ir jų yra labai daug. Dėl to gali iškilti nemalonios situacijos. Tarkime, turime daugiakampį, kurį padalinus į dvi dalis, uždavinys bus išspręstas. Sudalinus jį į iškiluosius daugiakampius uždavinys taip pat gali būti išspręstas, tačiau daugiakampių bus labai daug ir jie bus labai maži. Dar vienas sunkumas – visi tikslaus dalinimo į iškilus daugiakampius algoritmai atsiradus skylėms tampa nepolinominio sudėtingumo. Akivaizdu, kad spręsti nepolinominio sudėtingumo uždavinį tam, kad palengvinti sau darbą, yra neprotinga.

Be tikslaus skaidymo algoritmų yra ir apyiklslio padalinimo į iškiluosius daugiakampius algoritmas (Approximate Convex Decomposition, ACD [1]). Algoritmas gali būti sustabdomas tada, kai įdu-



bimų kiekis ir gylis sumažėja iki nustatyto parametro, pvz.: kai pašalinami ypatingai gilūs įdubimai. Be to, algoritmas apdoroja ir skylės, dėl to jo sudėtingumas beveik nepadidėja.

Uždavinį galime performuluoti taip – turimą daugiakampį padalinti į  $n$ -kampius. Jei  $n$  lygus 3, gauname trianguliacijo uždavinį. Kadangi daugiakampį visada galime trianguliuoti, ir, be to, galime tai padaryti labai greitai (paprasti algoritmai yra  $O(n \log n)$  sudėtingumo, bet yra ir  $O(n)$  sudėtingumo metodų), patogu naudoti šį metodą suformuluotam uždaviniui spręsti.

### **Daugiakampio skaidymo į apytiksliai iškilus daugiakampius algoritmas**

Šio algoritmo kūrėjai pastebėjo, kad uždaviniai, kuriuos yra paprasčiau spręsti, kai daugiakampiai yra iškilūs, panašiai supaprastėja, kai daugiakampiai yra "beveik" iškilūs. Tai yra, naudojant skaidymo į apytiksliai iškilus daugiakampius algoritmą, kitus uždavinius spręsti yra beveik taip lengva, kaip ir prieš tai, tačiau skaidymas žymiai supaprastėja.

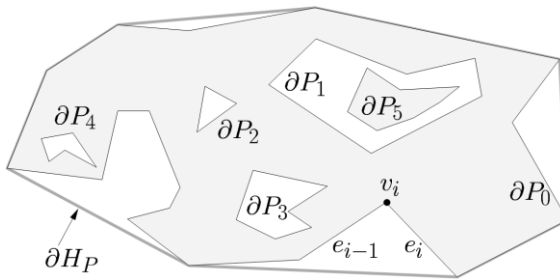
Metodo bendros idėjos:

- apskaičiuoti kiekvienos daugiakampio viršūnės įdubimo laipsnį.
- panaikinti skylės, t.y. "perkirpti" daugiakampį taip, kad skylės kontūras taptų išoriniu kontūru. Skylių kirpimas įvyksta pirmas dėl to, kad skylės įdubimo laipsnis yra nustatomas kitaip, nei išorinio kontūro viršūnių, dėl to yra visada didesnis, nei bet kurios viršūnės ant išorinio kontūro.
- perkirpti daugiakampį nuo viršūnės su didžiausiu įdubimo laipsniu iki patogiausios. Patogiausia viršūnė randama indeksuojant viršūnes pagal jos atstumą iki pradinės viršūnės ir pagal tai, kiek sumažės įdubimo laipsnis.

Toliau pateiksime platesnį algoritmo aprašymą bei galimybes jį pritaikyti mūsų suformuluotam uždaviniui.

Turime paprastą daugiakampį. Paprastas daugiakampis yra kun-  
tūrų aibė, kur  $\delta P_0$  yra išorinis kontūras, o likę – skylių kontūrai. Dau-  
giakampis yra paprastas, jei jokos nesusijungiančios briaunos nesi-  
kerta.

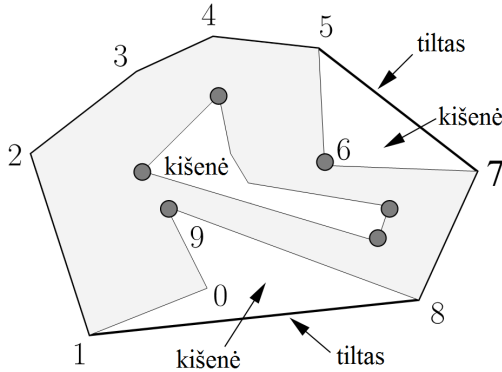
Daugiakampio  $P$  iškilusis lukštas  $H_P$  – tai mažiausias iškilus  
daugiakampis, į kurį telpa  $P$  (2 pav.). Jei  $P = H_P$ , sakome, kad  $P$  yra  
iškilus.  $P$  viršūnės, kurios nėra  $H_P$  viršūnės vadinamos įdubomis,  
vidinis kampas tarp tokios viršūnės briaunų yra didesnės už  $180^0$ .



2 pav. Daugiakampio kontūrai ir iškilus lukštas (pagal [1])

Įdubimo laipsnio matavimui šiame algoritme naudojamos *tilto* ir  
*kišenės* sąvokos. *Tiltas* yra iškilusis lukšto briaunos, sujungiančios  
nesančias greta daugiakampio kontūro  $\delta P_0$  viršūnes, t. y.  $TILTAI(P)$   
 $= \delta H_P \setminus \delta P$ . *Kišenė* yra didžiausia grandinė briaunų, nesančių iškilusio  
lukšto kontūre, t. y.  $KIŠENĖS(P) = \delta P \setminus \delta H_P$ . Tokiu būdu įdubimai  
gali būti tik kišenėse. Kiekvienas tiltas turi atitinkamą kišenę, t. y.  
 $\delta P_0$  briaunų grandinę, kurios pradžia ir pabaiga sutampa su tilto pra-  
džia ir pabaiga. Skylės irgi yra kišenės, tačiau jos neturi atitinkamų  
tiltų (3 pav.).

Algoritmas vadovaujasi *Skaldyk ir valdyk* principu – daugia-  
kampis dalinamas į 2 dalis, toliau yra dalinama kiekviena iš dalių, ir  
pan. Algoritmas rekursiškai dalina daugiakampį tol, kol didžiausias  
įdubimo laipsnis yra didesnis už nustatomą parametą  $\tau$ .  $\tau$  vadinamas  
neiškilumo paklaida. Kuo mažes  $\tau$ , tuo smulkiau bus dalinamas dau-  
giakampis, kad jį sudarančių daugiakampių įdubimo laipsnis



3 pav. Briaunos (5,7) ir (8,1) yra tiltai, o jų atitinkamos kišenės: (5;6), (6;7) ir (8;9), (9;0), (0;1) (pagal [1])

neviršytų  $\tau$ . Akivaizdu, kad kai  $\tau = 0$ , daugiakampis bus padalinamas į visiškai iškilus gabalus.

Pirmas algoritmo žingsnis – paskaičiuoti kiekvienos viršūnės įdubimo laipsnį. Jei viršūnė priklauso išoriniam kontūrai – įdubimo laipsnis yra trumpiausiai atstumas nuo jos iki atitinkamo tilto, nekerantis daugiakampio kontūro (4 pav.). Jei viršūnė priklauso skylės kontūrai – elgiamaisi kitaip. Randamos 2 labiausiai nutolusios viena nuo kitos skylės kontūro viršūnės  $p_i$  ir  $cw(p_i)$ . Apskaiciuojamas atstumas nuo šių dviejų viršūnių iki išorinio kontūro. Imamas mažesnis atstumas, prie jo pridedamas atstumas nuo  $p_i$  ir  $cw(p_i)$  ir maksimalus išorinio kontūro viršūnių įdubimo laipsnis. Gauname visos skylės įdubimo laipsnį, kuris yra didesnis, nei bet kurios išorinio kontūro viršūnės įdubimo laipsnis. Tokiu būdu yra pirma atsikratoma skylių.



4 pav. Trumpiausi keliai nuo tilto iki kiekvienos įdubusios viršūnės (pagal [1])

Tolimesnis algoritmo žingsnis yra surasti viršūnę su maksimaliu įdubimo laipsniu. Ji bus kirpimo linijos pradžia. Šios linijos pabaiga randama suteikus viršūnės indeksus: 0, jei per ją kerpant daugiakampį padalinsime į daugiau nei 2 gabalus, kitais atvejais – skaičių pagal formulę:

$$ind_i = \frac{1 + \alpha \cdot \text{concave}(i)}{\beta \cdot \text{dist}(i, r)}, \quad (1)$$

čia  $\text{concave}(i)$  – viršūnės  $i$  įdubimo laipsnis,  $\text{dist}(i, r)$  – atstumas nuo viršūnės  $i$  iki kirpimo linijos pradžios  $r$ ,  $\alpha$  ir  $\beta$  – pasirenkami parametrai. Kirpimo linijos pabaigos viršūnė bus viršūnė su didžiausiu indeksu. Atlikus kirpimą gauname 2 daugiakampius. Tada kiekvienam iš jų taikome tokį pat algoritmą.

Nors aprašytas algoritmas sprendžia kitokį uždavinį, nei mūsų suformuluotas, bet jį galima pritaikyti. Vienas iš būdų yra toks:

- Atsikratyti skylių pagal ACD algoritmą;
- Skaidyti daugiakampį pagal ACD kol visi komponentų kontūrai bus neilgesni, nei nustatyta;
- Jei ACD algoritmo vykdymas sustojo, o komponentų kontūrai vis dar ilgesni, nei nustatyta – daliname komponentus taip, kaip mums yra patogiu. Jokių problemų neturėtų iškilti, nes komponentai yra iškilūs arba beveik iškilūs.

Metodo privalumai:

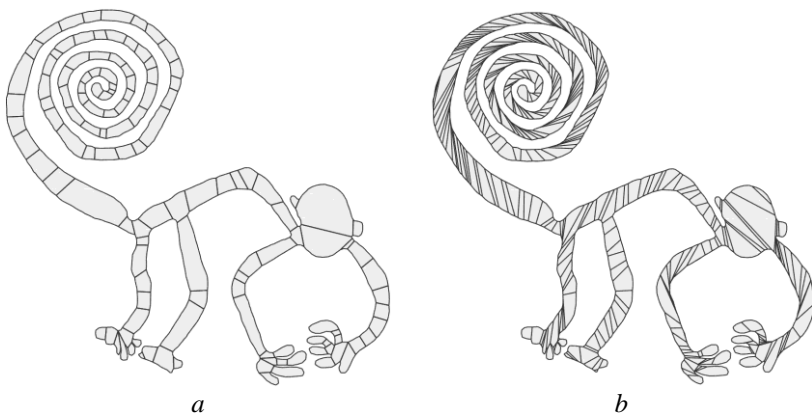
- Apdorojamos skylės.
- Greitas  $O(nr)$  sudėtingumo, čia  $n$  – viršūnių kiekis,  $r$  – įdubimų, gilesnių nei  $\tau$ , kiekis. Mūsų atveju gali neprireikti naikinti visų įdubimų, todėl vykdymas bus greitesnis.
- Gaunami komponentai žmogaus supratimui yra pakankamai prasmingi ir „gražūs“ (5 pav. a), t. y. nėra daugybės ilgų ir siaurų daugiakampių, kaip tikslaus skaidymo algoritmuose (5 pav. b).

- Skaidymo linijos pasirinkimo kriterijus galima nustatyti pačiam. Tokiu būdu galime gauti vienos ar kitos norimos formos daugiakampius.

Metodo trūkumai:

- Tarkime, turime daugiakampį, kurį padalinus į dvi dalis, uždavinys bus išspręstas. Tuo tarpu ACD algoritmus kirps daugiakampį į didelį ir labai mažą. Po keleto žingsnių turėsime vieną didelį daugiakampį ir daug mažų. Atlikus kažkiek skaičiavimų mūsų uždavinys liko neišspręstas, tik iš dalies supaprastintas.
- Bendru atveju komponentų kiekis bus didesnis, nei įmanomas mažiausias.

Skaidymo į apytiksliai iškilus daugiakampius metodą galima šiek tiek modifikuoti. Kai ieškoma, į kurią viršūnę iš labiausiai įdubusios reikia kirpti, galime viršūnėms suteikti indeksą ne tik atsižvelgiant atstumą ir jos įdubimo laipsnį, bet ir į mūsų suformuluoto uždavinio specifiką. Visas daugiakampio viršūnes pernumeruojame



5 pav. a) ACD algoritmo skaičiavimų rezultatas, b) Tikslus padalinimas į iškilus daugiakampius (pagal [1])

taip, kad kirpimo pradžios viršūnė būtų pirma. Toliau viršūnėms pri skiriame indeksą – skaičių, kuris lygus viršūnės numerio dalybos iš maksimalaus viršūnių kiekio daugiakampyje liekanai:

$$ind_i = \text{mod}(i, V_{\max}). \quad (2)$$

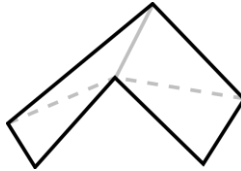
Iš visų matomų viršūnių išsirenkame tokią, kurios indeksas yra mažiausias. Tokiu būdu kai  $ind_i = 0$ , kirpimo linija atskirs daugiakampį, kurio viršūnių kiekis yra lygus arba kartotinas  $V_{\max}$ . Jei viršūnių kiekis lygus  $V_{\max}$  – šio daugiakampio nebenagrinėjame. Jei jų kiekis yra kartotinas  $V_{\max}$  – toliau taikome tokį modifikuotą ACD algoritmą. ACD metodo privalumas – kirpimo pradžia yra viršūnė su didžiausiu įdubimo laipsniu. Tokiu būdu mes panaikiname didžiausią įdubimą, dėl to gauti daugiakampiai yra labiau iškilūs, nei pradinis. Tokiuose daugiakampiuose iš kurios nors viršūnės bus matoma daugiau viršūnių, todėl bus didesnė tikimybė, kad matysime viršūnę su indeksu, lygiu nuliui.

### **Daugiakampio trianguliacijos algoritmo modifikacija**

Kaip jau buvo paminėta, uždavinį galime performuluoti kaip daugiakampio dalinimą į  $n$ -kampius. Tai yra panašu į daugiakampio trianguliacijos uždavinį. Galima įrodyti, kad daugiakampio trianguliacija visada egzistuoja ir gaunamų trikampių skaičius yra  $V-2$ , čia  $V$  yra daugiakampio viršūnių skaičius [3]. Taip pat paprasta pastebėti, kad jei turime daugiakampio trianguliacijos linijas, pašalinus vieną liniją pridėdama viena viršūnė (6 pav.). Taigi, galime trianguliuoti daugiakampį vienu iš žinomų metodų, ir tada naikinti kirpimo linijas taip, kad gautume reikiamo dydžio daugiakampius.

Trianguliacijos algoritmai yra greiti palyginus su daugiakampio tikslaus padalinimo į iškilus daugiakampius. Vieno iš paprastesnių algoritmų (Garey algoritmas), naudojančio daugiakampio monotoniškumo sąlyga, sudėtingumo įvertis yra  $O(n \log n)$ , [3]. 1991 metais B. Chazelle pasiūlė tiesinio sudėtingumo atgoritmą, tačiau jis yra labai sudėtingas [5]. Nereikalingų kirpimo linijų šalinimas irgi yra

tiesinio sudėtingumo operacija, kai trianguliacija atliekama Garey algoritmu. Taigi, bendras kirpimo linijų suradimo sudėtingumas yra  $O(n \log n)$ .



6 pav. Pašalinus vieną liniją pridėdama viena viršūnė. Apatiniame kairiajame kampe pradėdame nuo trikampio. Pašaliname vieną liniją, gauname keturkampį. Dešiniajame kampe pradėjus nuo trikampio ir pašalinus vieną liniją taip pat gauname keturkampį. Taip pašalinus 2 trianguliacijos apskaičiuotas linijas šešiakampį suskaidome į 2 keturkampius.

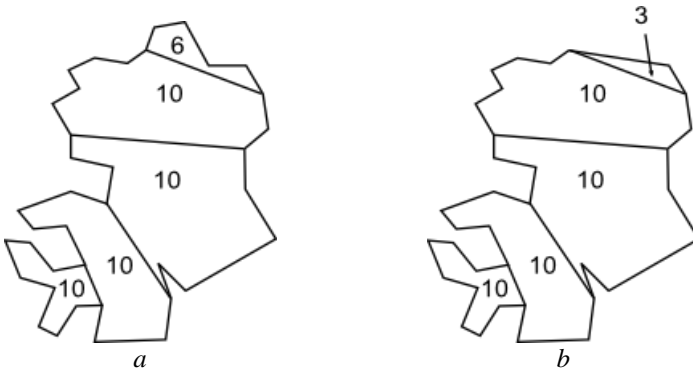
Metodo privalumai:

- Algoritmas yra paprastesnis nei daugiakampio skaidymo į apytiksliai iškilią dalis algoritmas
- Daugumos gautų dalių viršūnių skaičius yra lygus  $V_{max}$ . Tik vienos dalies viršūnių skaičius yra mažesnis. Pavyzdžiui, jei dalinamas daugiakampis iš 38 viršūnių į dalis iki 10 viršūnių, gauname 4 dalis ir 10 viršūnių ir vieną dalį su 6 viršūnėmis (7 pav. a).

Metodo trūkumai:

- Kadangi daugumos gautų dalių viršūnių skaičius yra lygus  $V_{max}$ , esant tam tikram viršūnių skaičiui paskutinės dalies viršūnių skaičius gali būti labai mažas, palyginus su kitomis dalimis. Pavyzdžiui, jei dalinamas daugiakampis ir 35 viršūnių į dalis iki 10 viršūnių, gauname 4 dalis iš 10 viršūnių ir viena dalį tik su 3 viršūnėmis (7 pav. b).
- Algoritmas neveikia, jei daugiakampyje yra skylių, t.y. skylių neturi būti, jis jų neapdoroja. Dėl to skylės reikia

kirpti kažkokiu kitu algoritmu, pavyzdžiui ACD algoritmo modifikacija.



7 pav. *a* – apylygis daugiakampio padalinimas, *b* – netolygus daugiakampio padalinimas, visos dalys yra 10 viršūnių dydžio, išskyrus viršutinę – trikampį.

## Rezultatai

Realizuotas modifikuotas trianguliacijos algoritmas Python programavimo kalba ir integruotas į Safe FME programinės įrangos (PI) paketą. Atlikti skaitiniai eksperimentai. Taip pat atlikti skaitiniai eksperimentai naudojant FME PI pakete esantį Chopper algoritmą. Šio algoritmo veikimo principas nėra atskleistas gamintojų, todėl palyginti galima tik rezultatus ir vykdymo laikus.

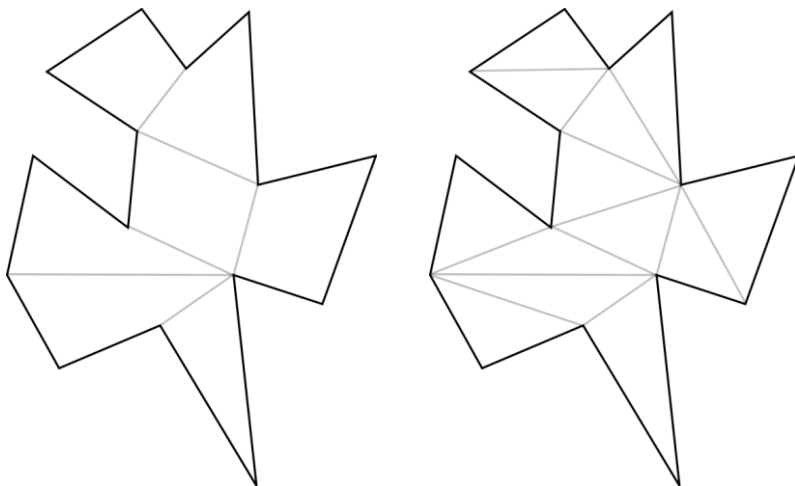
8, 9 ir 10 pav. pateikti skaičiavimo eksperimentų rezultatai. Reikėtų pastebėti, kad Chopper algoritmo rezultatai yra truputį keisti. Jei nurodome, kad dalių dydis neviršytų  $V_{max}$ , tai didžiausios dalies dydis yra  $V_{max} - 1$ . Tai nėra neteisinga, nes  $V_{max} - 1$  neviršija  $V_{max}$ , bet yra truputį keista, nes dažniausiai galima suskaidyti daugiakampį taip, kad būtų bent viena  $V_{max}$  dydžio dalis.

Vykdymo laikus nustatyti yra gana problematiška dėl kelių priežasčių. Visų pirma, FME PI paketas rodo tik bendrą duomenų apdorojimo proceso veikimo laiką. Procesas ne tik skaičiuoja daugiakam-



pio suskaidymą, jis taip pat turi nuskaityti duomenis iš kietojo disko, konvertuoti į savo vidinį formatą, tik tada atlikti reikiamas transformacijas, ir galų gale įrašyti duomenis. Antra, laikas yra apvalinamas iki sekundės dešimtųjų dalių. Trečia, skaičiavimo laikas priklauso ir nuo sistemos būsenos, pvz. nuo kietojo disko, operatyvinės atmintinės ir operacinės sistemos būsenų. Tai buvo nustatyta keletą kartų paleidus skaičiavimus ir gale pastebėjus, kad sisteminė skaičiavimo laiko sudedamoji nuolat kinta nuo 0,1 iki 1,5 s. Bendras skaičiavimo laikas būna iki 4 s. Taigi, sisteminė laiko sudedamoji duoda gana didelę paklaidą.

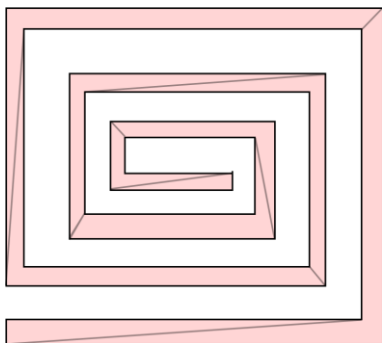
Ir trianguliacijos, ir Chopper algoritmo vykdymo laikai buvo gana panašūs – apie 6000 viršūnių daugiakampis suskaidomas maždaug per 4 s. Kiek tiksliai buvo vykdomas pats algoritmas – nėra aišku dėl prieš pateiktų priežasčių. Toliau tyrimo eigoje planuojama kuo tiksliau išmatuoti algoritmų vykdymo laikus.



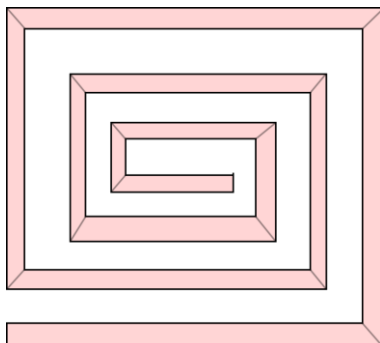
Trianguliacija  
6 keturkampiai ir 1 trikampis,  
viso 7 dalys

FME Chopper  
13 trikampių

8 pav. 15-kampio suskaidymas,  $V_{max} = 4$

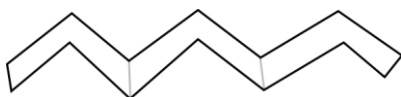


Trianguliacija  
8 penkiakampiai ir 2 trikampiai,  
viso 10 dalių



FME Chopper  
13 keturkampių

9 pav. 28-kampio suskaidymas,  $V_{max} = 5$



Trianguliacija  
3 šešiakampiai



FME Chopper  
Po vieną trikampį, keturkampį, penkia-  
kampį ir šešiakampį, viso 4 dalys

10 pav. 14-kampio suskaidymas,  $V_{max} = 6$

## Išvados

1. Aprašyti du algoritmai – trianguliacijos modifikacijos ir padalinimo į apytiksliai iškilus daugiakampius, kurie yra gana neblogos euristicos suformuluotam uždaviniui spręsti. Tačiau tik realizavus padalinimo į apytiksliai iškilus daugiakampius algoritmą bus galima pateikti rekomendacijas, kada naudoti vieną ar kitą algoritmą.

2. FME PĮ pakete realizuoto Chopper algoritmo rezultatai yra šiek tiek prastesni už trianguliacijos modifikacijos rezultatus, nes dalys yra mažesnės ir jų yra daugiau.

3. Palyginti skaičiavimo laikus yra sudėtinga, nes yra laiko sa-  
naudų, priklausančių ne tik nuo duomenų apimties, bet ir nuo siste-  
mos būsenos.

## Literatūra

- [1] Lien, J.-M.; Amato, N. M. 2004. Approximate convex decomposition of polygons, in *Proc. 20th Annual ACM Symposium on Computational Geometry (SoCG)*. New York, 17–26.
- [2] Chazelle, B.; Dobkin D. P. 1985. Optimal Convex Decomposition, in *G.T. Toussaint (Ed.), Computational Geometry*. Amsterdam, 63–133.
- [3] de Berg, M.; Cheong, O.; van Kreveld, M.; Overmars, M. 2008. *Computational geometry: Algorithms and applications*. Berlin Heidelberg: Springer-Verlag.
- [4] Sack, J.-R.; Urrutia, J. 2000. *Handbook on computational geometry*. Amsterdam: Elsevier.
- [5] Chazelle, B. 1991. Triangulating a simple polygon in linear time, in *Discrete & Computational Geometry*: vol. 6, issue 5. New York: Springer-Verlag, 485–524.

## ALGORITHMS FOR DECOMPOSITION OF SIMPLE POLYGON WITH HOLES

### D. Motiejauskas

#### Summary

This study deals with decomposition of simple polygon with holes into components so that every piece does not exceed some defined number of vertices. We define the problem and its appliances. Existing studies and algorithms for polygon decomposition are covered. We propose modifications of polygon triangulation and approximate convex decomposition algorithms. Also we compare modified triangulation algorithm results with Chopper algorithm (from Safe FME ETL software suite) results.

## VIRINTINIO DVITĖJO SIJOS SKERSPJŪVIO OPTIMIZAVIMAS

**Darius Ciūnys**

*Vilniaus Gedimino technikos universitetas*

[dcunys@gmail.com](mailto:dcunys@gmail.com), [dzukelis@gmail.com](mailto:dzukelis@gmail.com)

Vilniaus Gedimino technikos universitetas

**Santrauka.** Nagrinėjama optimalaus dvitėjinio virintinio sijos skerspjūvio paieškos uždavinys. Realizuoti ir trumpai straipsnyje apibūdinti algoritmai: modifikuotas variantų perrinkimo, atsitiktinės paieškos, kartotinės paieškos ir genetinis algoritmai. Smulkiau paaiškinta, kaip pritaikytas genetinis algoritmas skerspjūvių radimui. Atlikta realizuotų algoritmų rezultatų apžvalga ir palyginimas.

**Reikšminiai žodžiai:** plieninės konstrukcijos, sijinės konstrukcijos, dvitėjiniai plieno skerspjūviai, statybos techninio reglamento (STR) reikalavimai, konstrukciniai reikalavimai, skerspjūvių ploto optimizavimas, optimizavimo algoritmai.

### **Įvadas**

Sprendžiame sijinių konstrukcijų palengvinimo ir atpigavimo klausimus, ypač aktualius šių laikų statybose. Reikia rasti kuo mažesnių plotų sijų skerspjūvius. Kadangi pati optimaliausia skerspjūvių forma yra dvitėjinio tipo profilio, nagrinėsime tik dvitėjus skerspjūvius. Inžinieriai ir projektuotojai fiziškai negali patikrinti daugybės sijos skerspjūvių variantų ir rasti optimalų skerspjūvį, tenkinantį užduotas apkrovas, todėl darbe nagrinėjama kompiuterizuoti sijos optimalių skerspjūvių paieškos būdai, palengvinantis jų darbą ir leidžiantis išvengti žmogiškojo faktoriaus klaidų, skaičiuojant ir tikrinant statybos techninio reglamento (STR) sąlygas.

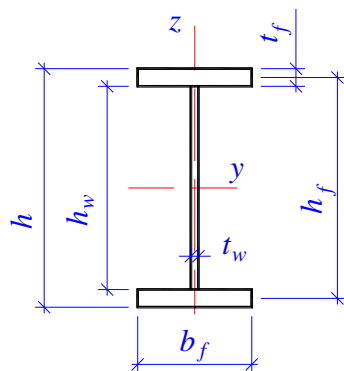
Konstrukcijų projektavimo įrankiuose tokiuose kaip ANSYS, Nastran ir kt. yra pritaikyta įvairių optimizavimo metodų. Optimizavimo algoritmai daug paprasčiau realizuojami, nei deterministiniai, jiems nebūtinai apribojimų tolydumo ir iškilumo sąlygos, nes inžinerijoje jos dažniausiai netenkinamos.

Optimizuojant skerspjūvius, papildomai reikia atsižvelgti į metalo pramonės keliamas sąlygas, kurios apriboja skerspjūvių matmenų pasirinkimo aibes (pavyzdžiui, nenusipirksite 271 mm pločio metalo profilio, nes bus tik 250 ir 300 mm). Egzistuoja dar keletas konstrukcinių sąlygų, kurios taip pat sumažina paieškos sritį (pavyzdžiui, plotis neturi būti mažesnis už storį), vėliau dėl šių sąlygų neįtraukimo būtų netenkinami STR matmenų santykių reikalavimai.

Darbo tikslas būtų, pritaikant žinomas inžinerijos ir optimizavimo metodų idėjas, sukurti originalius virintinių sijų skerspjūvių optimizavimo algoritmus, kurie tenkintų STR sąlygas.

### **Optimalaus virintinės sijos skerspjūvio ploto nustatymo uždavinys**

Optimaliausias sijos skerspjūvis yra dvitėjinio profilio skerspjūvis 1 pav. Dvitėjo skerspjūvio parametrų aibė susideda iš  $h_w$  — sienelės aukščio,  $t_w$  — sienelės pločio,  $b_f$  — juostos pločio ir  $t_f$  — juostos storio. Uždavinio tikslas — rasti kuo mažesnę skerspjūvio plotą, t. y. minimizuoti tikslo funkciją  $\min(h_w \cdot t_w + 2 \cdot b_f \cdot t_f)$ . Uždavinys nėra iškilas, todėl optimizavimo algoritmais negalėsime garantuoti optimalaus sprendinio.



1 pav. Dvitėjo skerspjūvio schema

Fig. 1. Steel W type cross-section schema

$$\min ( 2 \cdot b_f \cdot t_f + h_w \cdot t_w ),$$

$$\sigma_{\max} = \frac{M_{E,d}}{W_d} \leq f_{y,d} \cdot \gamma_c, \quad (1)$$

$$\tau_{\max} = \frac{3}{2} \cdot \frac{V_{E,d}}{t_w \cdot h_w} \leq f_{s,d}, \quad (2)$$

$$h = \left( \frac{1}{8} \dots \frac{1}{20} \right) l, \quad (3)$$

$$W_d \geq \frac{M_{Ed}}{f_{yd} \cdot \gamma_c}, \quad (4)$$

$$b_f = \left( \frac{1}{3} \dots \frac{1}{5} \right) h, \quad (5)$$

$$t_w \leq t_f \leq 3 \cdot t_w, \quad (6)$$

$$\frac{b_{eff}}{t_f} \leq 0,5 \sqrt{\frac{E}{f_{y,d}}}, \quad (7)$$

$$\bar{\lambda}_w = \frac{h_{eff}}{t_w} \sqrt{\frac{f_{y,d}}{E}} \leq \bar{\lambda}_{w,adm}. \quad (8)$$

$$b_{eff} = \frac{b_f - t_w}{2}, \quad (9)$$

$$h_{eff} = h_w.$$

Iš stiprumo sąlygos (1) nustatome reikalingą sijo skerspjūvio atsparumo momentą, pagal kurį vėlesniuose skaičiavimuose nustatome skerspjūvio charakteristikas. Kitos tarpinių skaičiavimų formulės pateiktos magistriniame darbe.

### Skerspjūvio optimizavimo algoritmai

#### Modifikuotas skerspjūvio parametrų variantų perrinkimo algoritmas

Modifikuotas variantų perrinkimo algoritmas remiasi inžinerine virintinio skerspjūvio matmenų nustatymo schema, kuri pateikta magistriniame darbe, o jos pagrindinės formulės pateiktos uždavinio formulavime. Algoritmo veikimo esmė: perrinkti variantus pagal du parametrus, t. y. sienelės aukštį  $h$  ir juostos plotį  $b_f$ , o likusius parametrus

pririšti priklausomybėmis (10), (11), (12) prie jau parinktų parametru. Taip išrinkti mažiausią skerspjuvį tenkinantį STR sąlygas (1), (2).

$$t_w = 7 + \frac{3 \cdot h}{1000}, \quad (10)$$

$$h_w = 0,95 \cdot 1,15 \sqrt{\frac{W_d}{t_w}}, \quad (11)$$

$$t_f = \frac{2 \left( W_d \cdot \frac{h}{2} - \frac{t_w \cdot h_w^3}{12} \right)}{b_f \cdot h_f^2}, \quad (12)$$

Deja, taikant inžinerinę matmenų nustatymo schemą, tikrinami ne visi reikalavimai, todėl po dvigubo ciklo tikrinami likę reikalavimai (7), (8). Ir jei jie netenkinami, tada algoritmas veikia inžineriškai, t. y. paprasčiausiai padidina tam tikras skerspjuvio charakteristikas, kad tenkintų visas sąlygas, o tai įneša didelę tikimybę, kad grįžus į dvigubo ciklo tam tikras vietas, būtų galima rasti šiek tiek kitokias charakteristikas, kurių pabaigoje nereikėtų didinti ir skerspjuvio plotas būtų mažesnis už padidintą, mažiausią rastą, plotą.

#### **Atsitiktinė paieška**

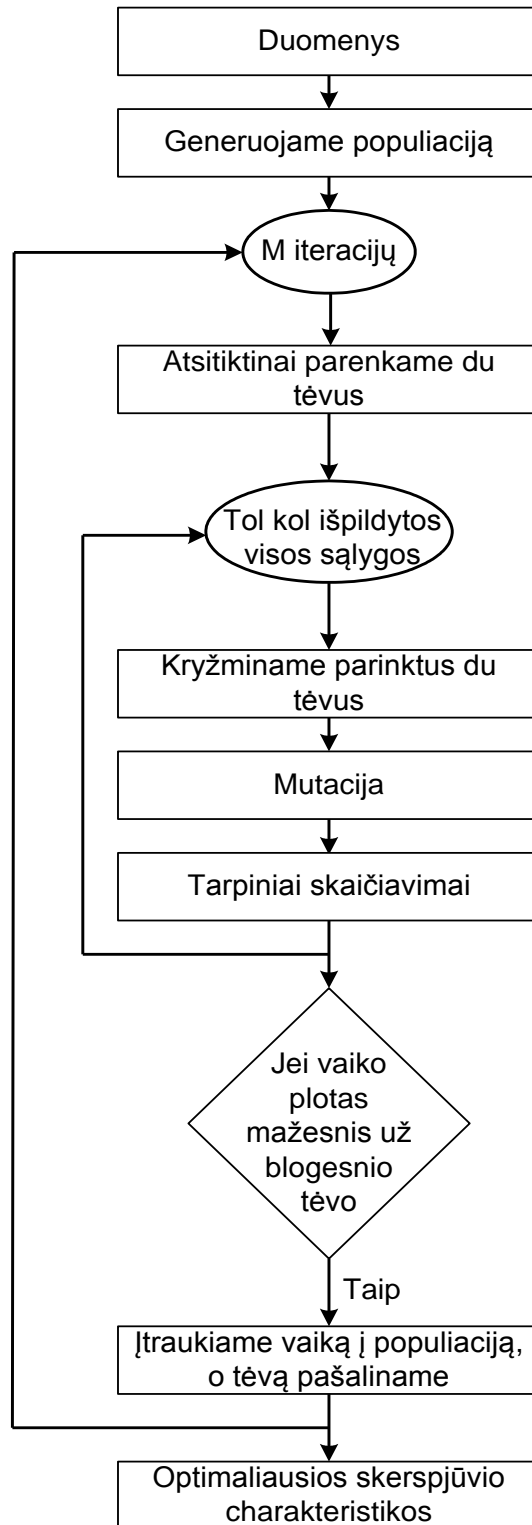
Algoritmo veikimo esmė: atsitiktinai generuoti visas keturias skerspjuvio geometrines charakteristikas, apribotas loginiais, inžineriniais arba konstrukciniais reikalavimais (3), (5), (6) ir  $8 \leq t_w \leq 20$  mm, tol, kol bus tenkinamos reikalaujamos STR sąlygos (1), (2), (7), (8). Taip gauname jėgas ir lenkimo momentus atlaikančią siją, tačiau labai retai artimą optimaliam sijos skerspjuviui. Panaši idėja naudojama inžinierių, pasirenkant pradinius skerspjuvius.

#### **Kartotinė paieška**

Kadangi atsitiktinės paieškos algoritmu visada randame, STR reikalavimus tenkinantį, sprendinį, todėl modifikuojame atsitiktinės paieškos algoritmą taip, kad atlikę pasirinktą kiekį iteracijų, surastume mažiausią skerspjuvio plotą iš visų anksčiau iteraciniame procese apskaičiuotų plotų.

#### **Genetinis algoritmas**

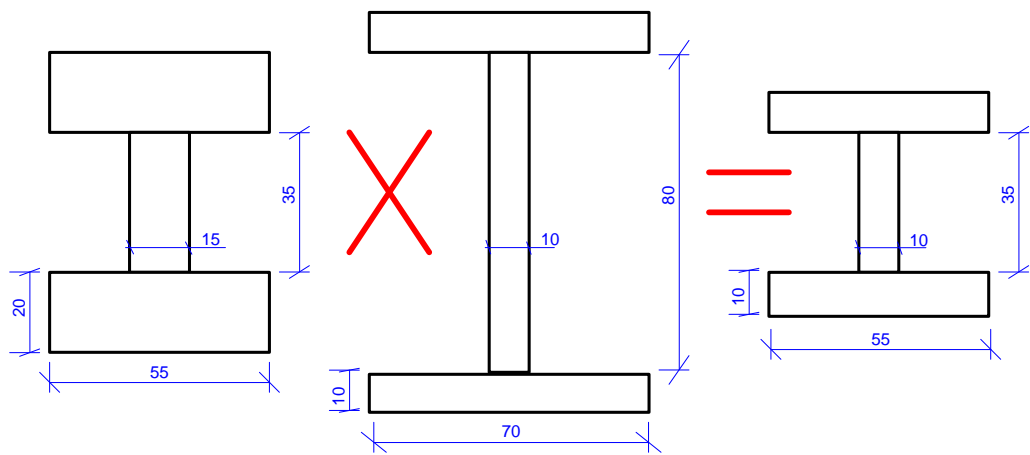
Populiacija, šiuo atveju, yra dvitėjų virintinių skerspjuvių šeima, kurią galime parinkti įvairiais būdais. Vienas iš būdų sudaryti pradinę populiaciją, tai sugeneruoti skerspjuvių parametru rinkinius atsitiktinės arba kartotinės paieškos algoritmais. Iš pradinės populiacijos atsitiktinai išrenkame du tėvus, kuriuos kryžminame ir gauname vaikus, t. y. naujus skerspjuvius gautus atsitiktinai iš vieno arba kito tėvo paėmus atitinkamus parametrus. Pavyzdžiui: I-ojo tėvo parametru rinkinys  $\{h_{w1}, b_{f1}, t_{w1}, t_{f1}\}$ , o II-ojo —  $\{h_{w2}, b_{f2}, t_{w2}, t_{f2}\}$ , taigi jų vaikas galėtų būti toks  $\{h_{w1}, b_{f1}, t_{w2}, t_{f2}\}$  arba  $\{h_{w2}, b_{f1}, t_{w1}, t_{f1}\}$  ir t. t. (kryžminimo pavyzdį žiūrėti 3 pav.). Populiacijos vystymuisi taikysime gamtos dėsnį „išlieka stipriausi“, t. y. egzistuoja baigtinis skaičius individų ir daugiau negali padidėti populiacija nei vienu individų, pavyzdžiui, saloje daugiau nėra maisto dar vienam, todėl atsiradę vaikai arba žūsta patys, jei tėvai stipresni arba atsiradę stipresni vaikai išstumia silpnesnius tėvus. Skerspjuvių atveju, charakteristikų kryžminimais sieksime rasti optimalius skerspjuvius. Skerspjuvių paieškai pritaikyto genetinio algoritmo chema pateikta 2 pav. Jei gautas naujas skerspjuvis netenkina STR ir kitų sąlygų (1) — (8), tai jis iškarto atmetamas kaip netinkamas variantas.



2 pav. Optimalių skerspjūvių paieškai pritaikyto genetinio algoritmo schema

Fig. 2. Genetic algorithm schema for optimal cross-section search

Vykdoma M dviejų atsitiktinių skerspjūvių parinkimų iš pradinės skerspjūvių aibės, tada jų charakteristikas atsitiktinai kryžminame tol, kol gauname visas keliamas sąlygas tenkinantį skerspjūvį, o tada lyginame tėvinių skerspjūvių ir jų vaikų plotus. Jei vaiko plotas mažesnis už blogesnio tėvo, įtraukiame jį į populiaciją.

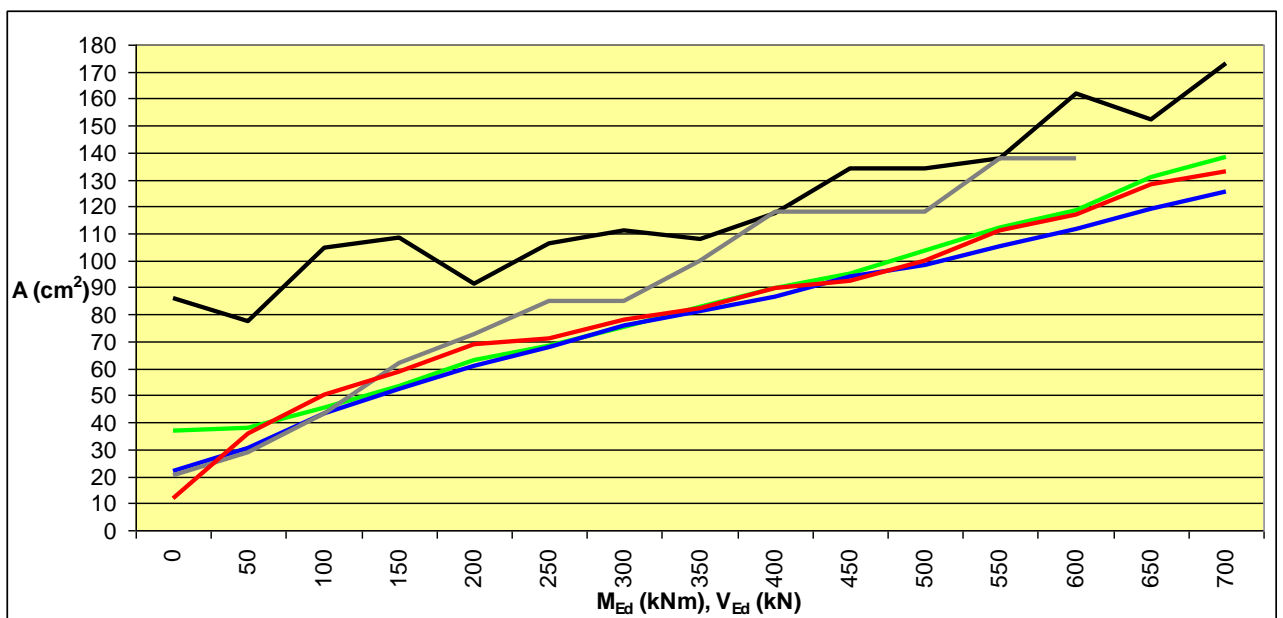


3 pav. Dviejų skerspjūvių kryžminimo pavyzdys

Fig. 3. Two cross-sectional sample of cross

### Rezultatai

Rezultatai pavaizduoti 4 pav. Raudona linija vaizduoja virintinių skerspjūvių priklausomybę, apskaičiuotą taikant modifikuotą perrinkimo algoritmą, žalia — taikant kartotinės 500 kartų paieškos algoritmą, mėlyna — taikant genetinį algoritmą, juoda — taikant atsitiktinę paiešką, pilka — palyginimui įdėta valcuotųjų skerspjūvių priklausomybė. Gamyklose gaminamų dvitėjų valcuotųjų profilių naudojimas konstrukcijose yra labai optimalus, esant mažoms apkrovoms, o esant labai didelėms apkrovoms, tenkinančio stiprumo reikalavimus skerspjūvio jau nebegalima įsigyti, nes jie negaminami, todėl esant didesnėms apkrovoms, reikia naudoti virintinį skerspjūvį. Kaip matome iš paveikslėlio, atsitiktinė paieška (juoda linija) randa sąlygas tenkinantį skerspjūvį, tačiau sprendiniai retai kada artimi optimaliems. Kitų realizuotų algoritmų rezultatai labai panašūs.



4 pav. Virintinių skerspjūvių plotų priklausomybių palyginimas nuo vienodo didumo įrašų

Fig. 4. Welded cross-sectional areas of the dependency comparison of flat-size effort

### Išvados

- Darbe realizuotas algoritmas, taikantis inžinierinę matmenų nustatymo schemą (Modifikuotas variantų perrinkimas). Visada gauna vienareikšmį sprendinį, skaičiuoja labai greitai, tačiau nebegrįžta į ciklus, o jei



netenkinamos sąlygos, užduotos visam skerspjūviui, paprasčiausiai padidinamos tam tikros charakteristikos ir padinama tikimybė, kad nerisime optimalaus sprendinio.

- Realizuota keletas optimizavimo algoritmų virintinių skerspjūvių radimui. Visų šių algoritmų trūkumas, kad jais gauti rezultatai yra nevienareikšmiški, o tai irgi negarantuoja optimalaus sprendinio. Genetinis algoritmas ilgiausiai trunkantis algoritmas, skaičiavimo prasme, tačiau rezultatai geriausi, aiški rezultatus stipriai įtakoja pradinių skerspjūvių parinkimas, o tai dar vienas trūkumas.
- Taikant šiuos algoritmus, būtų palengvintas inžinierių ir projektuotojų darbas, būtų atpigintos pastatų sijinės konstrukcijos, nes jų gamybai būtų sunaudojama mažiau medžiagų.

## Literatūra

1. Statybos techninis reglamentas STR 2.05.08:2005 Plieninių konstrukcijų projektavimas. Pagrindinės nuostatos
2. Statybos techninis reglamentas STR 2.05.04:2003 Poveikiai ir apkrovos
3. STR 2.05.08:2005 2 priedas Sijų klupumo koeficientai
4. C++ builder programavimo pavyzdžiai, J. Blonskis ir kt., Smaltijos leidykla, Kaunas, 2003
5. <http://www.cplusplus.com/doc/tutorial/> (dažnai)
6. <http://www.yevol.com/en/bcb/index.htm> (Borland C++ builder practical learning lesson, dažnai)
7. [http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2004~D\\_20040530\\_221515-42762/DS.005.0.01.ETD](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2004~D_20040530_221515-42762/DS.005.0.01.ETD)
8. [http://lt.wikipedia.org/wiki/Genetiniai\\_algoritmai](http://lt.wikipedia.org/wiki/Genetiniai_algoritmai)
9. [ik.su.lt/.../Genetiniai%20algoritmai/LT%20-%20Genetiniai%20algoritmai.doc](http://ik.su.lt/.../Genetiniai%20algoritmai/LT%20-%20Genetiniai%20algoritmai.doc)
10. <http://www.mokslai.lt/referatai/referatas/17416.html>
11. <http://www.pfi.lt/archyvas/pub/AnRep/2001/PDF/Dargys.pdf>
12. [http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2005~D\\_20051102\\_105606-30576/DS.005.0.01.ETD](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2005~D_20051102_105606-30576/DS.005.0.01.ETD)
13. [http://en.wikipedia.org/wiki/Structural\\_steel](http://en.wikipedia.org/wiki/Structural_steel)
14. <http://www.efunda.com/math/areas/IBeamIndex.cfm>

## WELDED I-BEAM CROSS-SECTION OPTIMIZATION

### D. Ciūnys

#### Abstract

Examining the optimal steel W type (Fig. 1) welded I-beam cross-sectional search task. And realize a short article to describe the algorithms: modified versions of variants, a random search, multiple search and genetic algorithms. For details explaining, how the genetic algorithm is applied to cross-sections of finding. Done realized overview of the results and comparison of algorithms.

**Keywords:** structures of steel, I-beam structures, cross-section of steel, demands of construction of technical regulation, demands of structures, cross-sectional area of optimization, optimization algorithms.



## DARBU (DARBOUX) FORMULĖS TAIKYMAS INTEGRALŲ APROKSIMACIJOMS KONSTRUOTI

**Irena Kulbytė**

*magistrantė*

*Vilniaus Gedimino technikos universitetas*

*el. paštas: Irena.Kulbyte@fmst.vgtu.lt*

**Santrauka.** Nagrinėjami integralai išreikšti specialiosiomis funkcijomis, kurie nereiškiami elementariosiomis funkcijomis ir pateikiami jų pavyzdžiai, funkcijų aproksimacijoms konstruoti taikoma Darbu (pranc. *Darboux*) formulė. Pateikiamos sukonstruotos specialiųjų funkcijų aproksimacijos ir aptariami atlikti aproksimacinių formulių tyrimai, nustatyti formulių tikslumo įverčiai.

**Reikšminiai žodžiai:** specialioji funkcija, aproksimavimas, asimptotiniai skleidiniai, Darbu formulė.

### Įvadas

Istoriškai susiklostė, kad specialiosios funkcijos ne taip gerai žinomos, kaip elementariosios. Tačiau jei pavyksta tokią funkciją iširti ir sužinoti jos savybes, tai ši funkcija mums teikia visą informaciją apie nagrinėjamą matematinį modelį. (Mariūnas et al. 1995)

Specialiosios funkcijos gali būti išreikštos (Olver 1974):

- Teiloro eilute. Jei funkcija  $f(x)$  be galo daug kartų diferencijuojama.
- Gama funkcija  $\Gamma(x)$ . Ši funkcija išreiškiama integralu:

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt$$

- Beselio funkcija  $J_n(x)$ . Sveikiems skaičiams  $n$  ir realiems arba kompleksiniams skaičiams  $x$  funkcija  $J_n(x)$  užrašoma Beselio integralu:

$$J_n(x) = \frac{1}{\pi} \int_0^{\pi} \cos(n\theta - x \sin \theta) d\theta, \quad n = 0, \pm 1, \pm 2, \dots$$

- ir t.t.

Kai nepavyksta specialiųjų funkcijų išreikšti elementariosiomis, tuomet konstruojamos funkcijų aproksimacijos. Funkcijų aproksimavimo uždavinių gausu įvairiose matematikos, fizikos ir technikos srityse. Aproksimacija (priartinimas) tai pateiktos funkcijos pakeitimas jai artima funkcija, taip kad artutinės funkcijos nuokrypis, nagrinėjamame intervale, būtų kuo mažesnis. (Listopadskis 2004)

Kadangi mūsų nagrinėjami integralai yra specialiosios funkcijos, jų aproksimacijoms konstruoti taikysime vieną iš skleidinių konstravimo metodų, leidžianti neintegruojamus integralus išskleisti jiems artimais skleidiniais, tai Darbu formulė.

### Darbu formulės bendroji išraiška

Darbu formulė integralui su  $n$ -tojo laipsnio polinomais:

$$f(x) - f(a) = \frac{1}{P_n^{(n)}(0)} \left( \sum_{m=1}^n (-1)^{m-1} (x-a)^m \left( P_n^{(n-m)}(1) f^m(x) - P_n^{(n-m)}(0) f^m(a) \right) + (-1)^n (x-a)^{n+1} I_n(x, a) \right) \quad (1)$$

$$I_n(x, a) = \int_0^1 P_n(t) f^{(n+1)}(a + t(x-a)) dt. \quad (2)$$

Čia  $P_n(t)$  – n-tojo laipsnio polinomas;  $I_n(x, a)$  – liekana.

### Darbu formulės bendroji išraiška, integralą nagrinėjant su pirmos eilės polinomu

Užrašome Darbu formulę integralui. (1) ir (2) formules nagrinėsime tik su pirmos eilės polinomu  $P_1(t)$ .

$$f(b) - f(a) = \int_a^b f'(s) ds; \quad f'(s) = g(s);$$

$$G(b) - G(a) = \int_a^b g(s) ds = \frac{1}{P_1'(0)} \cdot \left[ (-1)^0 (b-a)^1 (P_1(1)f'(b) - P_1(0)f'(a)) + (-1)^1 (b-a)^2 \cdot I_1(b, a) \right] \quad (3)$$

$$I_1(b, a) = \int_0^1 P_1(t) f''(a + t(b-a)) dt \quad (4)$$

Apsirašome pirmos eilės polinomo reikšmę ir jo išvestines:

$$\begin{array}{lll} P_1(t) = At + B & P_1(1) = A + B & P_1'(1) = A \\ P_1'(t) = A & P_1(0) = B & P_1'(0) = A \end{array}$$

Sustatome į (3) ir (4) išraiškas, pirmos eilės polinomo ir jo išvestinių reikšmes:

$$G_1 = \frac{1}{A} \cdot \left[ (b-a) \cdot ((A+B) \cdot f'(b) - B \cdot f'(a)) - (b-a)^2 \cdot I_1(b, a) \right] \quad (5)$$

$$I_1(b, a) = \int_0^1 (At + B) \cdot f''(a + t(b-a)) dt \quad (6)$$

Pertvarkome naujai gautas (5) ir (6) lygtis:

$$G_1 = (b-a) \cdot \left( \left(1 + \frac{B}{A}\right) \cdot f'(b) - \frac{B}{A} \cdot f'(a) \right) - (b-a)^2 \cdot I_1(b, a) \quad (7)$$

$$I_1(b, a) = \int_0^1 \left( t + \frac{B}{A} \right) \cdot f''(a + t(b-a)) dt \quad (8)$$

Įvedame keitinį  $\beta = \frac{B}{A}$ , kuriuo aprašome polinomo koeficientus A ir B. Perrašome (7) bei (8) lygtis su įvestu keitiniu:

$$G_1 = (b-a) \cdot \left( (1+\beta) \cdot f'(b) - \beta \cdot f'(a) \right) - (b-a)^2 \cdot I_1(b, a) \quad (9)$$

$$I_1(b, a) = \int_0^1 (t + \beta) \cdot f''(a + t(b-a)) dt \quad (10)$$

Gauname bendrąją liekamosios dalies  $I_1(b, a)$  išraišką, kuri priklauso nuo dviejų kintamųjų bei nuo parametro  $\beta$ .

$$I_1(b, a) = \int_0^1 (t + \beta) \cdot f''(a + t(b-a)) dt \quad (11)$$

Norėdami minimizuoti paklaidą (11) formulę prilyginę 0, išsivedame  $\beta(b, a)$ .

$$\beta(b, a) = -\frac{\int_0^1 t \cdot f''(a + t(b-a)) dt}{\int_0^1 f''(a + t(b-a)) dt} \quad (12)$$

Išvedame bendrąją aproksimacinę formulę.

$$G_2 = (b-a) \cdot \left( (1 + \beta(b, a)) \cdot f'(b) - \beta(b, a) \cdot f'(a) \right) \quad (13)$$

## Skaičiavimo rezultatai

Nagrinėkime neintegruojamą integralą, kuris elementariosiomis funkcijomis nereiškiamas.

$$F(v) = \int_0^v e^{-t^2} dt \quad (14)$$

$$g(s) = e^{-t^2}$$

Aprašome nagrinėjamo integralo išvestines (jų prireiks perrašant Darbu formulę pritaikyta savajam integralui):

$$f'(t) = e^{-t^2}; \quad f''(t) = -2 \cdot t \cdot e^{-t^2};$$

Galime užrašyti jau išvestą bendrąją aproksimacinę formulę (13) mūsų nagrinėjamam integralui (14). Vietoje  $b=v$  ir  $a=0$ . Taip pat užrašome ir integralų santykį  $\beta(b,a)$  iš (11) formulės.

$$G_2 = v \cdot e^{-v^2} + v \cdot e^{-v^2} \cdot \beta(v) - v \cdot \beta(v) \quad (15)$$

$$\beta(v) = -\frac{\int_0^1 t^2 \cdot e^{-t^2 v^2} dt}{\int_0^1 t \cdot e^{-t^2 v^2} dt} \quad (16)$$

Sukonstruosime interpoliacinį polinomą, kai integralų santykis (16) apibrėžtas intervale  $v \in (0; 3)$ . Lagranžo interpoliacinis polinomas  $L_n(x)$  yra mažiausio laipsnio polinomas, kuris taškuose  $x_0, x_1, \dots, x_n$  turi sutapti su funkcijos  $f(x)$  reikšmėmis tuose taškuose, t.y. (Listopadskis 2004)

$$L_n(x_i) = f(x_i), \quad i \in \{0, 1, \dots, m\}.$$

Pasirenkame keturis taškus, panaudoję (16) išraišką nustatome funkcijos  $\beta(v_n)$  reikšmes ir paskaičiuojame Lagranžo interpoliacinį polinomą.

$v_0 = 0;$	$\beta(v_0) = -0,6667;$
$v_1 = 1;$	$\beta(v_1) = -0,5995;$
$v_2 = 2;$	$\beta(v_2) = -0,4306;$
$v_3 = 3;$	$\beta(v_3) = -0,2953;$

Gautas reikšmes sustatome į interpoliacinio polinomo išraišką ir paskaičiuojame  $\beta(v)$ .

$$\beta(v) = \beta_1 \cdot v^3 + \beta_2 \cdot v^2 - \beta_3 \cdot v - \beta_4 \quad (17)$$

čia

$\beta_1 = -0,02270 ;$	$\beta_3 = -0,03010 ;$
$\beta_2 = 0,1194 ;$	$\beta_4 = -0,6666 .$

Gautą (17) lygtį įstatome į (15) išraišką

$$G_2(v) = v \cdot e^{-v^2} + (\beta_1 v^3 + \beta_2 v^2 + \beta_3 v + \beta_4) \cdot v \cdot e^{-v^2} - (\beta_1 v^3 + \beta_2 v^2 + \beta_3 v + \beta_4) \cdot v \quad (18)$$

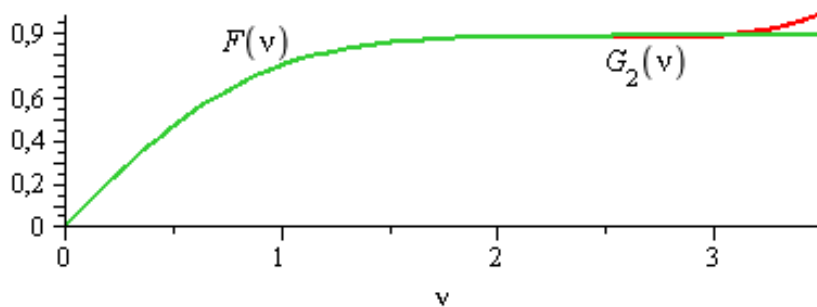
Apskaičiuojame funkciją vieneto aplinkoje, t.y. paskaičiuojame (18) lygtį kai  $v = 1$ , bei tiklią nagrinėjamo integralo reikšmę:

$$G_2(v) = 1 \cdot e^{-1^2} + (\beta_1 1^3 + \beta_2 1^2 + \beta_3 1 + \beta_4) \cdot 1 \cdot e^{-1^2} - (\beta_1 1^3 + \beta_2 1^2 + \beta_3 1 + \beta_4) \cdot 1 = 0,7472 \quad (19)$$

$$F(v) = \int_0^1 e^{-t^2} dt = 0,7471 \quad (20)$$

Gauname, kad paklaida yra lygi:  $|0,7472 - 0,7471| = 1 \cdot 10^{-4}$

Nubraižome grafiką, kuris parodo paklaidą tarp integralo  $F(v)$  tikslios ir aproksimacinės formulių, kai  $v \in (0; 3)$  (1. Pav.).



1. Pav. Funkcijos  $F(n)$  ir jos artutinės formulės  $G_2(n)$  grafikai

Pasinaudoję (14) ir (15) formulėmis, mes paskičiuojame santykinę paklaidą, kuri parodo gautos aproksimacinės formulės tikslumą. Skaičiavimo rezultatai pateikiami 1 lentelėje. Skaičiavimus atliekame matematinio paketo „Maple 10“ pagalba.

1 lentelė. Formulės  $F(n)$ , aproksimacijos tikslumo tyrimas

Žingsnis	Maksimali paklaida 3 taškai	Maksimali paklaida 4 taškai	Maksimali santykinė paklaida
0,1	$6,622890 \cdot 10^{-2}$	$9,16829759 \cdot 10^{-3}$	1,03467059
0,01	$6,622890 \cdot 10^{-2}$	$9,17075497 \cdot 10^{-3}$	1,03494016
0,001	$6,622898 \cdot 10^{-2}$	$9,17152975 \cdot 10^{-3}$	1,03502987
0,0001	$6,622899 \cdot 10^{-2}$	$9,17153524 \cdot 10^{-3}$	1,03503072
0,00001	$6,622899 \cdot 10^{-2}$	$9,17153528 \cdot 10^{-3}$	1,03503073
0,000001	<b><math>6,622899 \cdot 10^{-2}</math></b>	<b><math>9,17153528 \cdot 10^{-3}</math></b>	<b>1,03503073</b>

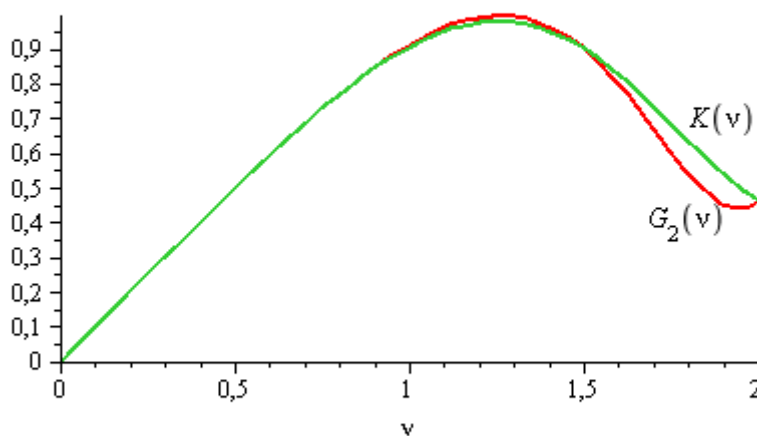
Taip pat nagrinėjome neintegruojamą integralą  $T(n)$ :

$$K(v) = \int_0^v \cos(t^2) dt \quad (21)$$

Pritaikę išvestą bendrąją aproksimacinės formulės išraišką integralui (13) ir atlikę visus skaičiavimus gavome integralo  $T(n)$  aproksimacinę formulę:

$$G_2 = v \cdot \cos(v^2) + v \cdot \cos(v^2) \cdot \beta(v) - v \cdot \beta(v) \quad (22)$$

Nubraižome grafiką, kuris parodo paklaidą tarp integralo  $K(v)$  tikslios ir aproksimacinės formulių, kai  $v \in (0; 2)$  (2. Pav.).



2. Pav. Funkcijos  $K(v)$  ir jos artutinės formulės  $G_2(v)$  grafikai

Pasinaudoję (21) ir (22) formulėmis, mes paskičiuojame santykinę paklaidą, kuri parodo gautos aproksimacinės formulės tikslumą. Skaičiavimo rezultatai pateikiami 2 lentelėje.

2 lentelė. Formulės  $K(n)$ , aproksimacijos tikslumo tyrimas

Žingsnis	Maksimali paklaida 4 taškai	Santykinė maksimali paklaida
0,1	$9,53561 \cdot 10^{-2}$	1,64722
0,01	$9,95186 \cdot 10^{-2}$	1,71397
0,001	$9,95283 \cdot 10^{-2}$	1,71405
0,0001	$9,95283 \cdot 10^{-2}$	1,71406
0,00001	$9,95283 \cdot 10^{-2}$	1,71406
0,000001	<b><math>9,95283 \cdot 10^{-2}</math></b>	<b>1,71406</b>

## Išvados

1. Išnagrinėti integralai, kurie išreikšti specialiosiomis funkcijomis. Jiems pritaikius Darbu formulę, sukonstruotos specialųjų funkcijų aproksimacijos.

2. Atlikti gautų aproksimacinių formulių skaičiavimo eksperimentai, kurie rodo, kad gautų formulių tikslumo įverčiai neviršija 2 % .

## Literatūra

Abramowitz, M.; Stegun, I. A. Handbook of mathematical functions with formulas, graphs and mathematical tables. National bureau of standards, 1964. 830 p.

Kleiza, J. Matematinis paketas maple: mokomoji knyga. Vilnius: Technika, 2003. 70 p.

Listopadskis, N. Taikomoji matematika. Kaunas: Kauno technologijos universitetas, 2004. 161 p.

Mariūnas, M.; Mariūnas, D. Specialiosios funkcijos ir jų transformacijos. Vilnius: Technika, 1995. 27 p.

Olver, F.W.J. Asymptotics and Special Functions. New York: Academic Press, 1974. 572 p.

Weisstein, E. W. Darboux's Formula. 1999 – 2010 m. [žiūrėta 2010-02-15]. Prieiga per internetą: <http://mathworld.wolfram.com/DarbouxFormula.html>

Бочканов, С.; Быстрицкий, В. Специальные функции. 1999 - 2000 m. [žiūrėta 2010-01-25]. Prieiga per internetą: <http://alglib.sources.ru/specialfunctions/>

## APPLICATION OF DARBOUX FORMULA FOR CONSTRUCTION OF APPROXIMATIONS OF INTEGRALS

### I. Kulbytė

#### Summary

Examined of integrals to express the special functions, who is not expressed elementary functions and presented their examples. Functions approximations construct apply Darboux formula. Served the construction of special functions approximations and discuss the carry out investigations formulas approximations, set formulas accuracy estimates.

**Keywords:** Special function, approximation, asymptotic expansions, Darboux formula.

## MATEMATINIS MODELIAVIMAS

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“, įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

# MEDINIO RĖMO SKAIČIUOJAMOSIOS SCHEMOS PARAMETRŲ TYRIMAS

**Irena Naus**

*Vilniaus Gedimino Technikos Universitetas*

*El. paštas: irena.naus@gmail.com*

**Anotacija.** Nagrinėjama medinio rėmo skaičiuojamosios schemos parametrų įtaką rėmo medžiagos kiekiui. Apžvelgti du matematiniai modeliai. Pirmuoju atveju nebuvo atsižvelgta į klupdymo sąlygą, o antruoju atveju modelis papildytas lygtimis, kuriomis yra atsižvelgiama į gegnės stabilumą. Išnagrinėti ir palyginti du netiesinio programavimo metodai. Pateikti išspręstų modelių rezultatai, kurie leidžia pasirinkti optimalias parametrų  $h$  (rėmo elemento stačiakampio aukštis) ir  $H$  (rėmo aukštis) reikšmes.

**Reikšminiai žodžiai:** medinis rėmas, konstrukcija, netiesinis optimizavimas, netiesinis programavimas, trijų šarnyrų rėmas, gegnė, klupdymas, kirpimas, gniuždymas.

### Tiriamos problemos aktualumas

Stabilių, saugių ir optimalių įvairių konstrukcijų projektavimas yra aktuali problema visais laikais. Egzistuoja optimizavimo metodai, kurie realizuoja projektuotojų siekius. Optimizavimo uždaviniai – pirmas konstrukcijų optimalaus projektavimo etapas, pagrįstas matematinio modeliavimo metodais. Pagrindinis siekis – suprojektuoti optimalią konstrukciją, kuri tenkintų visus eksploatacinius reikalavimus. Norint skaičiavimą pritaikyti realioms konstrukcijoms, būtina optimizavimo uždavinių matematinuose modeliuose kuo tiksliau įvertinti konstrukcijos medžiagos savybes, išorinius poveikius ir kitus veiksnius. Iš dalies tai pasiekama, įvertinus medžiagos savybes, kurių įvairovė pasižymi statybinės konstrukcijos. Matematinio programavimo metodus taikant patogu ir



efektyvu modeliuoti konstrukcijos projektavimą atsižvelgiant į normatyvinius konstrukcijos darbo reikalavimus. Statyboje labai svarbu naudoti lengvas, didelės laikomosios galios konstrukcijas. Medinės konstrukcijos yra vienos iš jų. Mediena – seniausia natūrali statybinė medžiaga, plačiai naudojama ir šiandien. Mediena yra labai stipri ir gerai priešinasi smūgiams.

Mūsų nagrinėjamą modelį galima pritaikyti realioms konstrukcijoms. Darbe nagrinėjami matematiniai modeliai, bei gauti rezultatai, padeda sumažinti statybinių medžiagų sąnaudas. Parinktas matematinis optimizacijos modelis leidžia sukurti optimalias konstrukcijas, naudojant mažiau statybinių medžiagų, bet išlaikant statinio stabilumą.

## **Literatūros analizė**

Iš literatūros apžvalgos aišku, kad medinių rėmo optimizavimo modeliai yra brandi sritis.

Akademikas A. Čyras sukūrė mechanikos mokyklą Lietuvoje, kurioje išskirtinis dėmesys buvo skiriamas matematinio programavimo metodų taikymui konstrukcijoms optimizuoti. Šis tyrimo metodas buvo novatoriškas ne tik Rytų Europoje, bet ir tuometinėje Sovietų Sąjungoje. Jo sukurta mokykla sėkmingai tęsia tyrimus ne tik Lietuvoje, bet ir užsienyje. Tarp jų atstovų paminėtini Vilniaus Gedimino technikos (VGTU) mokslininkai A. Čižas, R. Kačianauskas, R. Baušys, taip pat A. Borkowski, dirbantis Lenkijos mokslų akademijoje (Norkus, 2006: 2).

Nuo pirmojo Heyman (1958) darbo, kuriame minimizuotas rėminių konstrukcijų svoris pagal klasikinius optimalumo kriterijus, paskelbti tik pavieniai darbai ir atlikti pavieniai tyrimai. Heyman iteracinė skaičiavimo metodika vėliau buvo pritaikyta kelių kitų autorių (pvz., Cohn ir Parimi 1973; König 1975; Borkauskas ir Atkočiūnas 1975; Save ir Prager 1985; Nguyen ir Morelle 1990). Skaitinės programos, pagrįstos optimalumo kriterijumi (Rozvany 1989), buvo išplėtos ir taikomos tam tikriems konstrukcijų tipams (Gianbanco ir kt. 1994 a,b; Gianbanco ir Palizzolo 1995; Tin-Loi 2000; Kaliszky ir Lógó 2002) (Mirkevičiūtė, 2005: 21).

Optimizavimo klausimai gana išsamiai išnagrinėti V.A. Permyakov ir V.A. Glitin straipsnyje „Trijų šarnyrų rėmo

optimizacijos taikymas su kintamais standumo elementais“. Straipsnyje pateikiamas rekomendacijas dėl įvairios plieninių karkasų paskirties pastatų, kuriuos sudaro trys šarnyrų sujungtos dalys, optimizavimo metodų.

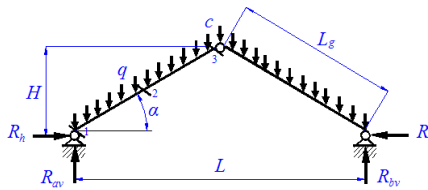
V.P. Jarcev, O.A. Kiseleva straipsnyje „Medinių konstrukcijų projektavimas ir testavimas“ aprašyti skaičiavimo metodai ir atraminių struktūrų bandymai. Darbe nagrinėjamos medinės sijos, dviejų ir trijų šarnyrų rėmai, klijuotinės medienos arkos.

Amerikos miško ir popieriaus asociacija, „Informacija apie tradicines medinio rėmo konstrukcijas“. Leidinyje pateikiama pagrindiniai reikalavimai medinio rėmo statybai ir informacija, kuri padeda statant ir tikrinant medinio rėmo ilgaamžiškumą ir efektyvumą.

## Darbo tikslas

Išnagrinėti medinio rėmo skaičiuojamosios schemas parametru įtaką rėmo medžiagos kiekiui, pasiūlyti racionaliausius schemas parametrus.

Tiriama medinių rėmų parametro–aukščio  $H$  įtaka medžiagos, reikalingos rėmui pagaminti, kiekiui (medienos tūriui). Rėmo tipas parodytas 1 paveiksle.



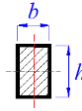
**1 pav.** Nagrinėjamas rėmo tipas

**Fig. 1.** The present frame type

Apibrėžkime pradines prielaidas. Rėmo elementų skerspjūvis stačiakampis (2 pav.). Skerspjūvio plotis  $b$  pasirenkamas ir gali būti 6,0; 10,0; 12,0; 14,0; 16,0; 18,0 cm. Rėmo anga  $L$  pasirenkama ir gali būti 6,0; 9,0; 12,0; 15,0; 18,0; 21,0 m.

Apkrova  $q$  apskaičiuojama pagal stogo dangos konstrukciją ir sniego apkrovą.

Skaičiuojant atsižvelgiama į lenkimo momento, ašinės ir skersinės jėgos poveikį.



**2 pav.** Rėmo elementų skerspjūvis stačiakampis  
**Fig. 2.** Frame elements cross-section rectangle

Skaičiavimai atliekami pagal STR 2.05.07:2005 Medinių konstrukcijų projektavimas reikalavimus.

### Trijų šarnyro rėmo – gegnių geometrijos tyrimas

Darbe nagrinėjama medžiagos tūrio minimizavimo uždaviniai:

$$V = b \cdot h \cdot L_g \cdot 2 = 2 \cdot b \cdot h \cdot \sqrt{H^2 + \left(\frac{L}{2}\right)^2} \quad (1)$$

Su stiprumo sąlygomis:

- antrojo pjūvio lenkimas su gniuždymu;

$$\sigma = \frac{|N_2|}{A} + \frac{|M_2|}{W_w} \leq f; \text{ (į klupdymą neatsižvelgiama)} \quad (2)$$

$$\sigma = \frac{|N_2|}{A} + \frac{|M_{mod}|}{W_w} \leq f; \text{ (atsižvelgiama į klupdymą)}^1 \quad (3)$$

$$\varphi = \begin{cases} 1 - 0,8 \left( \frac{\lambda}{100} \right)^2, & \text{kai } \dots \lambda \leq 70^2, \\ \frac{3000}{\lambda^2}, & \text{kai } \dots \lambda > 70 \end{cases} \quad (4)$$

- pirmojo pjūvio kirpimas

$$\tau = 1,5 \frac{|Q_1|}{A} \leq f_s; \quad (5)$$

- pirmojo pjūvio gniuždymas

$$\sigma = \frac{|N_1|}{A} \leq f; \quad (6)$$

<sup>1</sup> Stabilumo netikimas, strypo išlinkimas, jam netenkant tiesios pusiausvyros formos vadinamas klupimu.

<sup>2</sup> Klupumo koeficientas  $\varphi$  rodo, kiek kartų reikia sumažinti skaičiuojamąjį stiprį, kad užtikrintume saugų gniuždomojo strypo darbą.

čia  $L_g$  – gegnės ilgis,  $N$  – ašinės jėgos,  $M$  – lenkimo momentai,  $Q$  – skersinės jėgos,  $A$  – rėmo elementų stačiakampio skerspjūvio plotis,  $W_x$  – atsparumo momentas,  $\varphi$  – klupumo koeficientas,  $\lambda$  – gegnės liaunis,  $f_s$  – kirpimo stipris.

Suformuluotas uždavinys virsta netiesiniu optimizacijos uždaviniu, kuris išsprendžiamas dviems būdais. Uždavinio kintamieji yra  $h$  ir  $H$ , kiti dydžiai žinomi. Taigi, mūsų netiesinio optimizavimo uždavinys atrodo taip:

$$2 \cdot b \cdot h \cdot \sqrt{H^2 + \left(\frac{L}{2}\right)^2} \rightarrow \min \quad (7)$$

Apribojimai:

$$\frac{\left| -q \left( \frac{L^2}{8H} + \frac{H}{2} \right) \right|}{b \cdot h} + \frac{\left| \frac{3}{8} \cdot q \cdot L \cdot \sqrt{H^2 + \left(\frac{L}{2}\right)^2} \right|}{b \cdot h^2} \leq f \quad (\text{be klupdymo}); \quad (8)$$

$$\frac{|0,375 \cdot q \cdot L|}{b \cdot h} \leq f_s; \quad (9)$$

$$\frac{\left| -q \left( \frac{L^2}{8H} + H \right) \right|}{b \cdot h} \leq f. \quad (10)$$

Antras patikslintas uždavinys skiriasi nuo pirmo, kad pirmą sąlygą (antrojo pjūvio lenkimas su gniuždimu) pakeičiame kita, kur atsižvelgiame į klupdymą:

$$\frac{\left| -q \left( \frac{L^2}{8H} + \frac{H}{2} \right) \right|}{b \cdot h} + \frac{\left| \frac{3}{8} \cdot q \cdot L \cdot \sqrt{H^2 + \left(\frac{L}{2}\right)^2} \right|}{b \cdot h^2 \left( 1 - \frac{q \left( \frac{1}{8} \frac{L^2}{H} + \frac{1}{2} H \right)}{b \cdot h \cdot f \cdot \varphi} \right)} \leq f \quad (11)$$

Netiesiniams matematinio programavimo uždaviniams spręsti yra daug algoritmų (Himmelblau 1975; Vanderplaats 1984; Schittkowski 1994 ir kt.). Kiekvienas metodas turi savų privalumų ir trūkumų, bet nėra tokio, kuris tiktų absoliučiai visiems atvejams. Sprendinio algoritmas priklauso nuo uždavinio formuluotes ir vartotojo (D. Mirkevičiūtė, 2005: 22).

Šie optimizavimo uždaviniai sprendžiami dviems būdais:

- tiesinio aproksimavimo metodu;
- gradiento projektavimo metodu (Rozeno metodas).

## Tiesinis aproksimavimo metodas

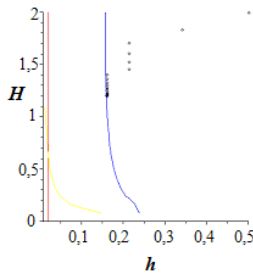
Tikslo funkciją ir kiekvieną sąlygą aproksimuojamatesiškai, taikant Teiloro eilutę taško  $x^{(k)}$  aplinkoje.

Pritaikius algoritmą uždaviniui gauname:

$$x = (h, H);$$

$$\begin{aligned} & F(x^{(0)}) + \nabla^T F(x^{(0)})(x - x^{(0)}); \\ & g_1(x^{(0)}) + \nabla^T g_1(x^{(0)})(x - x^{(0)}) \leq 0; \\ & g_2(x^{(0)}) + \nabla^T g_2(x^{(0)})(x - x^{(0)}) \leq 0; \\ & g_3(x^{(0)}) + \nabla^T g_3(x^{(0)})(x - x^{(0)}) \leq 0. \end{aligned} \quad (12)$$

čia  $F$  – tikslo funkcija, o  $g_1, g_2, g_3$  – netiesinės sąlygos: antrojo pjūvio lenkimas su gniuždymu, pirmojo pjūvio kirpimas, pirmojo pjūvio gniuždymas atitinkamai. Kadangi  $F(x^{(k)})$ ,  $\nabla^T F(x^{(k)})$ ,  $g_i(x^{(k)})$ ,  $\nabla^T g_i(x^{(k)})$  pastovūs vektoriai arba konstantos, kurių reikšmių galime surasti taške  $x^{(k)}$ , tai ši užduotis tampa tiesinio programavimo uždaviniu. Išsprendus uždavinį nustatoma, kad funkcija  $F$  įgyja minimumą taške  $x^{(1)}$ . Toliau pakeičiama  $x^{(1)}$  ir vėl sprendžiamas uždavinys. Taip gaunama seka sprendinių, kuri konverguoja į pirmo uždavinio sprendinį. Gautas rezultatas taškas  $x^{(k)}$  yra apytikslis pradinio optimizavimo uždavinio sprendinys. Matematinio Maple paketo skaičiavimo pavyzdys parodytas 3 pav.



3 pav. Skaičiavimo pavyzdys<sup>3</sup>

Fig. 3. Calculation Example

- <sup>3</sup> mėlyna spalva pavaizduota pirmą sąlygą, t.y. antrojo pjūvio lenkimas su gniuždymu;
- raudona spalva pavaizduota antra sąlyga, t.y. pirmojo pjūvio kirpimas;
- geltona spalva pavaizduota trečia sąlyga, t.y. pirmojo pjūvio gniuždymas.
- taškai rodo sprendinio artėjimą prie minimumo.

## Gradiento projektavimo metodas (Rozeno metodas)

Gradiento metodu buvo sprendžiamas uždavinys atsižvelgiant į gegnės klaidymą. Rozeno projektuojamųjų gradientų metodą galima taikyti, kai tikslo funkcija, apribojimai yra tiek tiesiniai, tiek netiesiniai. Šio minimizavimo metodo privalumas yra greito nusileidimo metodas.

Pasirenkamas bet koks taškas  $x^{(k)}$  iš galimos srities. Tada nustatoma kryptis

$$\hat{s}^{(k)} = \frac{P_l \nabla f(x^{(k)})}{\|P_l \nabla f(x^{(k)})\|},$$

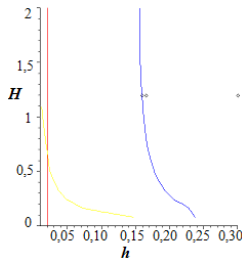
čia  $P_l = I - A_l^T (A_l A_l^T)^{-1} A_l$ .

Matrica  $P_l$  vadinama projekcinė,  $I$  – vienetinė matrica, matrica  $A_l$  – aktyvūs apribojimai).

Surandamas maksimalus  $\lambda$

( $\lambda^* = \max \{ \lambda | x^{(k)} + \lambda \hat{s}^{(k)} \in R \}$ ), čia  $R = \{x | g_i(x) \geq 0, i=1..p\}$ ), kuris neišeitų iš srities ir skaičiuojamas su kurio  $\lambda^{(k)}$  funkcija įgyja minimalią reikšmę, t.y. minimumą. Tokiu būdu skaičiuojamas kitas  $\tilde{x}$  ( $x^{(k+1)} = x^{(k)} + \lambda^{(k)} \hat{s}^{(k)}$ ). Ir vėl kartojama procedūra, kol neatsiduriama prie apribojimų.

Matematinio Maple paketo skaičiavimo pavyzdys parodytas 4 pav.



**4 pav.** Skaičiavimo pavyzdys  
**Fig. 4.** Calculation Example

## Gauti rezultatai

➤ Analizė atliekama pagal tiesinio aproksimavimo metodu gautus rezultatus atsižvelgiant į klupdymą ir be klupdymo.

Pasirinkama apkrova  $q=2,5 \text{ kN/m}^2$ ,  $L=9 \text{ m}$ ,  $b=0,18 \text{ m}$ ,  $f=12 \text{ MPa}$ ,  $f_s=2,5 \text{ MPa}$ , iteracijų skaičių  $k=25$ ,  $x^{(0)}=(0,5;2)$  ir  $\delta = 0,2$ .

Vienu atveju (kai į klupdymą nebuvo atsižvelgta) gaunama, kad:

- $h=0,1408$  ir  $H=1,203$ . Funkcijos reikšmė  $f(h, H) = 0,2361 \text{ m}^3$ ;

kai atsižvelgiama į klupdymą, tai:

- $h=0,1588$  ir  $H=1,203$ . Funkcijos reikšmė  $f(h, H) = 0,2663 \text{ m}^3$ .

Rėmo elemento stačiakampio aukštis su klupdymo sąlyga padidėja  $0,0179 \text{ m}$ , kai stogo aukštis liko toks pats, o tūris padidėja  $0,0302 \text{ m}^3$ .

Fiksavus gautas  $h$  reikšmes ir didinant aukštį  $H$  gaunama, kai neatsižvelgius į klupdymą, tūris po 10 iteracijų padidėja  $1,7 \text{ dm}^3$ , o atsižvelgiant į klupdymą, tūris padidėja  $2 \text{ dm}^3$ .

➤ Tiesinio aproksimavimo metodo ir gradientų metodo palyginimas.

Pasirenkama  $q=2,5 \text{ kN/m}^2$ ,  $L=9 \text{ m}$ ,  $b=0,18 \text{ m}$ ,  $f=12 \text{ MPa}$ ,  $f_s=2,5 \text{ MPa}$ .

Nustatyta, kad minimumas tiesinio aproksimavimo metodu pasiekiamas atlikus 37 iteracijų skaičių, o gradientų metodu prirėikė tik dviejų iteracijų. Gautas gradientų metodų minimumas yra mažesnis, nei tiesinio aproksimavimo metodu gautas rezultatas (1 lent.).

**1 lentelė.** Metodų palyginimas

Metodai	k	$h(\text{m})$	$H(\text{m})$	Tūris ( $\text{m}^3$ )	Tūrio skirtumai
Tiesinio aproksimavimo metodas	37	0,165	0,800	0,27183	0,00525
Gradientų metodas	2	0,159	1,198	0,26658	

- Tūrio ir metodų palyginimas prie skirtingų rėmo angos  $L$  pavaizduotas 2 lentelėje.

## 2 lentelė. Tūrio ir metodų palyginimas

L (m)	$X^{(0)}$	b (m)	Tiesinio aproksimavimo metodas			Gradientų metodas			Tūrio skirtum ai
			$h$ (m)	$H$ (m)	Tūris (m <sup>3</sup> )	$h$ (m)	$H$ (m)	Tūris (m <sup>3</sup> )	
6	(0,3;1)	0,06	0,186	0,604	0,068	0,179	0,996	0,067	0,00049
	(0,3;1)	0,1	0,143	0,616	0,087	0,138	0,995	0,087	0,00005
	(0,3;0,8)	0,12	0,135	0,470	0,098	0,128	0,796	0,095	0,00301
	(0,3;1)	0,14	0,122	0,602	0,104	0,118	0,995	0,104	0,00009
	(0,3;0,8)	0,16	0,115	0,583	0,112	0,112	0,795	0,111	0,00121
	(0,3;0,8)	0,18	0,115	0,402	0,124	0,106	0,795	0,118	0,00634
9	(0,3;1,2)	0,1	0,217	0,802	0,198	0,209	1,198	0,194	0,00388
	(0,3;1,2)	0,12	0,199	0,809	0,218	0,192	1,198	0,214	0,00379
	(0,3;1,2)	0,14	0,185	0,802	0,237	0,179	1,198	0,233	0,0039
	(0,3;1,2)	0,16	0,175	0,802	0,255	0,168	1,198	0,250	0,00507
	(0,3;1,2)	0,18	0,165	0,802	0,271	0,159	1,198	0,266	0,00539
12	(0,3;1,6)	0,12	0,259	1,363	0,382	0,256	1,599	0,381	0,00099
	(0,3;1,6)	0,14	0,244	1,202	0,418	0,238	1,599	0,413	0,00429
	(0,3;1,6)	0,16	0,229	1,211	0,448	0,224	1,599	0,445	0,00369
	(0,3;1,6)	0,18	0,217	1,204	0,478	0,212	1,599	0,473	0,00455

- Maksimalus stogo aukštis, kai pasirinkta tam tikra rėmo anga pavaizduotas 3 lentelėje.

## 3 lentelė. Maksimalus stogo aukštis

Metodai	L (m)	$h$ (m)	$H$ (m)	Tūris (m <sup>3</sup> )	$H$ $max$ (m)
Tiesinio aproksimavimo metodas (be klaidų sąlygos)	9	0,142	0,81	0,233	1,83
Tiesinio aproksimavimo metodas (su klaidų sąlyga)		0,165	0,802	0,272	3,52



Metodai	L (m)	h (m)	H (m)	Tūris (m <sup>3</sup> )	H <sub>max</sub> (m)
<b>Gradientų metodas</b>		0,159	1,198	0,267	2,65
<b>Tiesinio aproksimavimo metodas (be klaidymo sąlygos)</b>	<b>12</b>	0,189	1,121	0,272	2,46
<b>Tiesinio aproksimavimo metodas (su klaidymo sąlyga)</b>		0,217	1,204	0,478	4,3
<b>Gradientų metodas</b>		0,212	1,599	0,474	3,53
<b>Tiesinio aproksimavimo metodas (be klaidymo sąlygos)</b>	<b>15</b>	0,235	1,395	0,646	2,87
<b>Tiesinio aproksimavimo metodas (su klaidymo sąlyga)</b>		0,279	1,204	0,763	6,27
<b>Gradientų metodas</b>		0,27	1,6	0,745	5,18

## Išvados

1. Sprendžiant uždavinį nustatyta, kad svarbiausia yra pirmoji sąlyga (antrojo pjūvio lenkimas su gniuždymu). Tuomet kitos dvi sąlygos (pirmojo pjūvio kirpimas ir pirmojo pjūvio gniuždymas) yra automatiškai išpildamos.

2. Rėmo elemento stačiakampio aukštis atsižvelgiant į įklaidymą padidėja beveik 2 cm, kai stogo aukštis liko toks pats, tūris padidėja 0,0302 m<sup>3</sup>.

3. Prie tam tikrų pasirinktų parametų tiesinio aproksimavimo metodų minimumas pasiekiamas atlikus 37 iteracijų skaičių, o gradientų metodu pririekia tik dviejų iteracijų. Gradientų metodų minimumas yra 0,00525 m<sup>3</sup> mažesnis, nei tiesinio aproksimavimo.

4. Darbe išnagrinėti du uždaviniai leidžia pasirinkti optimalius parametrus h (rėmo elementų stačiakampio aukštis) ir H (rėmo aukštis) reikšmės.

## Literatūra

- Valentinavičius, A.; Valiūnas, B. 2000. *Medinės konstrukcijos*. Vilnius: Enciklopedija. 222 p.
- Lietuvos Respublikos aplinkos ministro įsakymas Nr. 233. 2003. *Statybos techninis reglamentas STR 2.05.04:2003 Poveikiai ir apkrovos*. Vilnius. 50 p.
- Lietuvos Respublikos aplinkos ministro įsakymas Nr. D1-79. 2005. *Statybos techninis reglamentas STR 2.05.07:2005 Medinių konstrukcijų projektavimas*. Vilnius. 44 p.
- Norkus, A. 2006. *Netampriųjų konstrukcijų analizės ir optimizavimo metodai ir konstrukcijų projektavimas* [interaktyvus]. Vilnius. 4 p. Prieiga per internetą:  
<<http://neris.mii.lt/mt/straipsniai/200605/netamp.doc>>.
- Mirkevičiūtė, D. 2005. *Tamprių-plastinių sistemų optimizacija su standumo ir stabilumo sąlygomis*. Vilniaus Gedimino Technikos Universitetas. Vilnius: Technika. 131 p.
- Химмельблау, Д. 1975. *Прикладное нелинейное программирование*. Москва: Мир. 532 с.
- Ярцев, В.; Киселева, О. 2005. *Проектирование и испытание деревянных конструкций*. Тамбов: ТГТУ. 81 с.
- Пермяков, В.; Глітін, О. 2008. *Застосування оптимізації при проектуванні тришарнірних рам каркасів з елементами змінної жорсткості*. Київ. 11 с.
- American Forest & Paper Association. 2001. *Details for conventional wood frame construction*. Washington. 51 p.

## THE RESEARCH OF THE PARAMETERS OF CALCULATING SCHEME FOR WOODEN FRAME

### I. Naus

#### Summary

A computational scheme for a wooden frame and the frame parameters on quantities of material are presenting in the article. We review two mathematical models. The first case was not taken into buckling condition and the second case, equations were added to the model, which takes into account the stability of the rafters. We examined and compared two non-

linear programming techniques. We present the solutions of the models, which allows us to select the optimal parameters  $h$  (the height of the rectangular frame elements) and  $H$  (height of frame) values.

**Keywords:** wood frame, construction, non-linear optimization, non-linear programming, a three-hinge frame, truss, buckling, shearing, compression.

# MATEMATIKA

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“,  
įvykusios Vilniuje 2010 m. 04 mėn. 09 d., medžiaga

## MATEMATIKOS MOKSLINIŲ STRAIPSNIŲ KLASIFIKAVIMO MODELIAVIMAS

**Julija Šliachtovič, Andrius Šimonėlis, Jelena Lavcel**

*Vilniaus Gedimino technikos universitetas, j.sliachtovic@gmail.com*

*Vilniaus Gedimino technikos universitetas, sim.andrius@gmail.com*

*Vilniaus Gedimino technikos universitetas, j.lavcel@gmail.com*

**Anotacija.** Nagrinėjame matematikos mokslinius straipsnius ir bandomė nustatyti publikacijos sritį. Sudarėme raktinių žodžių žodyną ir atlikome mokslinių straipsnių informacinę analizę: nustatėme matematinių terminų, priskirtų moksliniuose straipsniuose, skaičiaus statistiką bei dažnio sumą, raktinių žodžių entropiją ir apskaičiavome straipsnių „atstumus“ viena į kito. Atlikome, skaičiavimui naudojamos, programos testavimą.

### Įvadas

Per visą matematikos mokslo istoriją išleistų mokslinių darbų apimtys skaičiuojamos dešimtimis milijonų puslapių. Šiuose darbuose sukauptos žinios, kurių efektyvus valdymas yra ne šiaip svarbus uždavinys, bet viena pagrindinių mokslo kaip prasmingos veiklos sąlygų. Žinių valdymas suprantamas kaip rinkinys veiklų, kurios leidžia atpažinti, tvarkyti, kurti bei platinti žinias tam, kad jas būtų galima atkartoti, mokyti ir įsisavinti. Šios veiklos forma visada egzistavo mokslo pasaulyje, tačiau paskutinių dešimtmečių pokyčiai kompiuterijos bei komunikacijų srityse atvėrė naujas galimybes, o tai sąlygojo naujų reikalavimų bei tikslų išskėlimą.

Mokslo žinių valdymo problemų rate automatinis tekstų, ypač – mokslinių, klasifikavimas yra vienas aktualiausių uždavinių. Dabar tai, ką paprastai atlikdavo pats teksto autorius, pvz., nurodydamas

raktinius žodžius ar išvardindamas mokslo sritį apibūdinančius standartinius klasifikatorių kodus, gali atlikti arba bent jau padėti atlikti automatinė sistema. Rezultatai neprivalo būti apriboti ir fiksuoti – jie gali kisti, priklausomai nuo konkrečių sistemos poreikių.

Šiame straipsnyje nustatomas matematinių terminų, priskirtų moksliniuose straipsniuose, skaičiaus statistika bei dažnio suma, raktinių žodžių entropija, straipsnių „atstumas“ vieno iki kito ir atliekamas programos testavimas.

Antrajame skyriuje apžvelgiami susiję darbai mokslo publikacijų automatino klasifikavimo uždaviniams spręsti. Trečiajame skyriuje pristatoma programa ir jos testavimo rezultatai. Ketvirtajame skyriuje aprašomas atliktas eksperimentas, iliustruojantis rezultatus.

## **Susijusių darbų apžvalga**

Publikacijų klasifikavimas yra viena iš svarbių mokslo tekstų tvarkymo veiklų, sudarančių galimybes kaupti, ir, kas svarbiausia, esant poreikiui surasti bei panaudoti mokslinės informacijos ir žinių fragmentus. Rankinis šio darbo atlikimas ne tik neefektyvus ir nesudėtingas, bet ir neprasmingas dabartinių techninių galimybių kontekste [1]. Todėl disertacijoje nagrinėjama specifinio mokslinio turinio automatinio klasifikavimo problema.

[1] darbe yra pasiūlyti ir ištirti matematiniai klasifikavimo metodai, paremti statistine mokslo terminijos pasiskirstymo tekstuose analize, kuriuos būtų galima taikyti taikomajam mokslo publikacijų klasifikavimo uždaviniui spręsti.

Atliktas eksperimentinis tyrimas parodė, kad teksto elementų (žodžių, frazių), kuriuos atitinkantys skaitiniai požymiai reprezentuoja dokumentus, aibės parinkimas stipriai įtakoja algoritmų rezultatus. Naudojant iš tyrimams naudotos publikacijų duomenų bazės sudarytą mokslo terminijos (pavienių žodžių ir frazių) žodyną gautas apie 5-10 % didesnis tirtų klasifikavimo algoritmų tikslumas, nei naudojant įprastą visų kalbos žodžių sutinkamų straipsnių tekstuose, žodyną [1].

Disertacijoje prieita išvados, kad moksliniams tekstams klasifikuoti skirtų specialių algoritmų nėra, paprastai tyrimuose ir praktiniuose taikymuose naudojami įprasti bendrinės kalbos tekstų klasifikavimo metodai.

Panagrinėkime kitą straipsnį. [2] darbe buvo atlikta matematikos mokslinių straipsnių informacinė analizė pagal temas ir išmatuotas panašumo lygis. [2] straipsnio autorius šiam uždaviniui spręsti pasiūlė automatinį indeksavimą, kai kompiuteris ieško tekste raktinių žodžių. Autorė sukūrė pakankamai didelį raktinių žodžių žodyną ir skaičiavo šių terminų ir jų variacijų dažnius. Tam panaudojo specialiai sukurtą programą, kuri leido pabandyti įvertinti straipsnių „artumo“ laipsnį. Autorė patikrino tris metrikos aksiomas:

1. simetriškumą
2. trikampio nelygybę
3. maksimumas lygus 2.

Atlikus tyrimus, gautos išvados, kad reikia atsižvelgti į žodžių kombinaciją, praplėsti žodyną, atlikti eksperimentus su kitomis metrikomis. Norėdami tobulinti, reikia daugiau dėmesio skirti žodyno sudarymui [2].

## **Tyrimas**

Tyrimui panaudojome programą, kurią parašė Vilniaus Gedimino technikos universiteto studentas Andrius Šimonėlis. Ši programa leidžia lyginti straipsnių informatyvumą.

Programa parašyta C# programavimo kalba. Iš panašių programų (MS Word, Excel ir kt.) išsiskiria patogumu, lankstumu, bei suderinamumu su kitomis programomis (galima paruošti tekstą kitoms programoms apdorojimui). Be to visi kada nors gauti rezultatai numatomi saugoti duomenų bazėje.

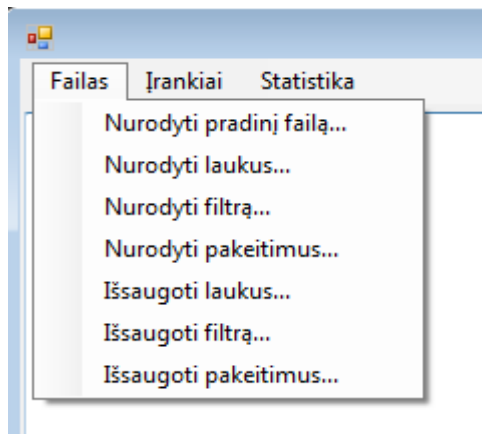
Programos pagalba galima nufiltruoti įvestus simbolius, ar frazes, importuotas iš išorinio failo. Taip pat programos pagalba galima įvesti laukus, kuriems reikia priskirti žodį ar frazę iš teksto. Būtina nurodyti simbolius, kaip nurodyta frazė bus išskirta tekste iš priekio

ir iš galo. Tokių laukų kiekis yra ribotas iki 1000. Apdorojant tekstą sukuriamas naujas failas, kuriame visos rastos frazės, nepriklausomai kartojasi jos ar ne, yra išskirtos. Ši programos savybė sudaro sąlygas, teksto apdorojimui panaudoti ir kitas programas.

Kolkas viena svarbiausių programos galimybių yra žodžių dažnių sudarymas, be to žodžių sąrašas yra importuojamas iš išorinio failo, kas yra išties patogu.

Gauti rezultatai yra atvaizduojami kompiuterio ekrane, taip pat juos galima įrašyti į išorinį failą.

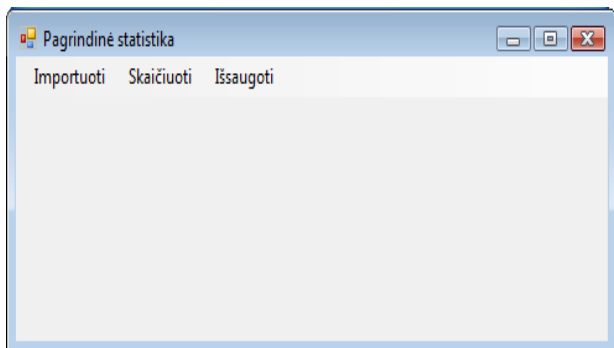
Įsijungus programą būtina nurodyti mokslinį straipsnį TXT formato, tai galima padaryti Meniu pasirinkus Failas->Nurodyti pradinį failą (1 pav.)



**1 pav.** Failas

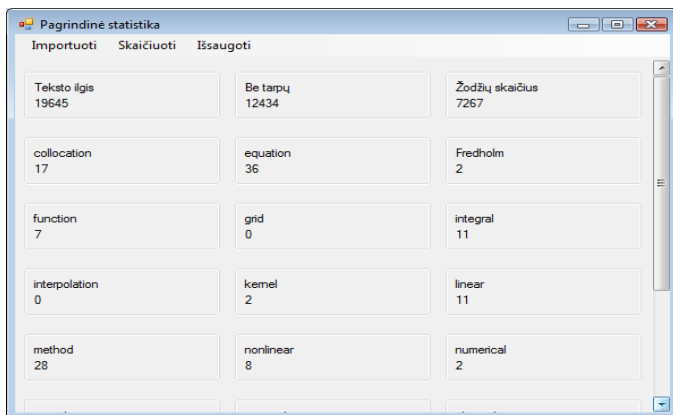
Pasirinkus iš meniu Statistika -> Pagrindinė statistika, atsiradusiame lange (2 pav.), kur galime iškarto rinktis meniu punktą skaičiuoti, tačiau tuomet rezultatų lange matysime tik kelis laukus pagrindinės teksto statistikos (teksto ilgį, žodžių skaičių, ir t.t.).

Vienas svarbesnių dalykų čia yra Importuoti, nes čia mes galime importuoti papildomus žodžius ar frazes iš TXT failo. TXT faile kiekvienas naujas žodis turi prasidėti iš naujos eilutės, o visos jo variacijos turi būti toje pačioje eilutėje atskirtos kableliais.



**2 pav.** Pagrindinė statistika

Pasirinkus analizuojamą tekstą, kuris turi būti TXT formato ir importavus raktinių žodžių failą skaičiuojame kiekvieno žodžio skaičių tekste (3 pav.).



**3 pav.** Rezultatų vaizdavimas

Prieš pateikiant programą vartotojui testavimas yra būtinas. Pasirenkame palaipsnių testavimą. Testavime naudojami atsitiktiniai duomenys t.y. iš 21 išrinkti 7 straipsniai ir 9 žodžiai, kurie yra priskirti bent vienam straipsniui. Straipsnių apimtis siekia iki 300 žodžių. Programa patikrinama apskaičiuojant žodžių pasikartojimą straipsnyje rankiniu būdu.



1 lentelė. Dažnio suma apskaičiuota su progra-

	1	2	6	7	10	12	14
equation	9	6	2	5	6	3	2
grid	1	2	1	4			
integral	3	9		3		3	2
kernel	1	3		3	1	4	
method	4	12	1		6	1	4
numerical	1		1	2			1
polynomial	1	1	1	3	4		5
singular	1			5	1		
solution			1	5	2		4

ma

2 lentelė. Dažnio suma apskaičiuota rankiniu bū-

	1	2	6	7	10	12	14
equation	9	7	2	5	6	3	2
grid	1	2	1	4			
integral	3	9		3		3	2
kernel	1	3		3	1	4	
method	4	12	1		6	1	4
numerical	1		1	2			1
polynomial	1	1	1	3	4		5
singular	1			5	2		
solution			1	5	2		4

du

Tyrimo metu sukurta programa negali atlikti funkcijos pagal iškeltus reikalavimus:

- 2-ame straipsnyje neranda visų žodžių „*equation*“ ir jo variacijų,
- 10-ame straipsnyje neranda žodžių „*singular*“ ir jo variacijų „*singularity*“.

## Eksperimentas

Sukuriame pakankamai didelį raktinių žodžių žodyną, kuris turi būti sudaromas iš terminų ir jame neturėtų būti bendrinės kalbos žodžių. Žodyną sudaro 295 žodžiai. Paimame kiekvieną straipsnį atskirai ir skaičiuojame terminų ir jų variacijų dažnius. Iš šitų dažnių sudarome kiekvienam straipsniui vektorių  $\vec{x} = (x_1, x_2, \dots, x_n)$ . Gauna-

me diskreutinę informacijos šaltinį ir apskaičiuojame entropiją. Tam gautus duomenis normuojame, tai yra sudarytą vektorių padaliname iš rastų žodžių sumos  $\frac{\bar{x}}{n}$  ir apskaičiuojame kiekvieno straipsnio entropiją.

Tam apskaičiuojame santykinius dažnius

$$p_k = \frac{d_k}{d}, \quad (1)$$

čia:  $d_k$  - žodžių dažniai;  $d$  - dažnių suma, kurią skaičiuojama pritaikius formulę

$$d = \sum_{j=1}^n d_j.$$

Toliau apskaičiuojame entropiją pagal formulę

$$E = - \sum_{k=1}^n p_k \cdot \log_2 p_k, \quad (2)$$

čia:  $p_k$  - santykinis dažnis;  $E$  - entropija.

Jei žodžių beveik nesutikome, gausiname mažą entropiją. Entropijos maksimumas galimas, kai turime  $\frac{1}{n}$ . Jei imame kitus straipsnius, kurių vektorių toks pat, tai jie yra identiški, tai yra gauname tiek pat informacijos. Mūsų atveju entropija yra rodiklis, kai gauname informaciją apie straipsnį

Kitas tyrimo žingsnis yra atstumo skaičiavimas. Skaičiuojame straipsnių „atstumus“ vieną iki kito pasinaudojus Hemingo atstumų formulę

$$\|\vec{x} - \vec{y}\| = \sum_{k=1}^n |x_k - y_k|, \quad (3)$$

čia:  $\vec{x} = (x_1, x_2, \dots, x_n)$  - vieno straipsnio dažnių vektorius;  $\vec{y} = (y_1, y_2, \dots, y_n)$  - kito straipsnio dažnių vektorius.

Eksperimentą atlikome su 43 straipsniais. Pasinaudoję programa apskaičiavome kiek kartų žodis pasikartoja kiekviename iš straipsnių ir kiekvieno straipsnio dažnio sumą. Duomenys patalpino į Excel failą (3 lentelė), apskaičiavome santykinį dažnį pagal (1) formulę, sudarėme kiekvienam straipsniui vektorius, sudarytus iš dažnių, normavome juos ir apskaičiavome kiekvieno straipsnio entropiją, pasinaudojus (2) formule.

3 lentelė. Žodžių skaičius ir dažnio suma straipsniuose

	0201-002	003-007	002-001	012-001	002-007	007-007	009-002	004-004	007-004	009-004	004-004	007-004	009-004	009-004	009-004	009-004	
tend	2																1
term	5		4		2	2				1	2						
theorem	17	1	27	1		3	6			5	1						
thin										7							
time	7	4	3		4	5	10	3		11	10	3	35	1			
tool					1		1										1
torsional																	
trajectories,		2															
transformation		2	1		4	10		2		1		2					
translation																	
travel										1							
traveling				1						3							
truncation,					1	1											
uniform		7				3		2				1					
unique,	4	1	9	1		8	11			1	1						1
unstable,										3	1						
variable	1	3			3	1				2							17
vector, vectorial,	2											2	4				1
velocities				5						1							
Whitham's										2							
zero	1	2	2		1	1		1	1	1	2	2	1				2
	$d =$	312	273	320	215	145	526	189	238	355	233	210	399	130	204	161	
entropija	(straipsnio) =	3,69	3,77	3,85	3,93	3,62	3,77	3,67	3,85	4,29	3,55	3,91	3,96	3,76	3	3,25	
		65%	66%	68%	67%	64%	66%	65%	68%	75%	62%	69%	70%	66%	53%	57%	

Kitas tyrimo žingsnis normuoti vektorių ilgius ir apskaičiuoti straipsnių „atstumus“ vieną nuo kito, pasinaudojus (3) formule. Gauname 4 lentelę

**4 lentelė.** Atstumų tarp straipsnių lentelė

	000	0005	0007	0009	0012	0102	0103	0108	0110	0201	geoc	ekon	vady
0003-001	0,7	1,6	1,5	1,1	1,8	1,5	1,6	1,7	1,4	1,8	1,7	1,7	
0005-001		1,5	1,4	1,1	1,8	1,5	1,6	1,7	1,4	1,7	1,6	1,6	
0007-007			1,3	1,5	1,7	1,2	1,7	1,8	1,3	1,6	1,6	1,7	
0009-019				1,3	1,7	1,4	1,4	1,6	1,3	1,7	1,6	1,7	
0012-003					1,6	1,3	1,6	1,7	1,3	1,8	1,7	1,8	
0102-012						1,7	1,5	1,8	1,6	1,8	1,8	1,9	
0103-009							1,7	1,8	1,4	1,7	1,7	1,7	
0108-031								1,5	1,3	1,8	1,7	1,8	
0110-001									1,5	1,8	1,8	1,9	
0201-001										1,8	1,5	1,6	
geodezija													1,7
ekonomika													
vadyba													1,3

Palyginimui paėmėme 3 straipsnius: vadybos, ekonomikos ir geodezijos. Atlikome tuos pačius tyrimus ir pastebėjome tokius rezultatus (3 lentelė). Iš lentelės matome, kad vadybos ir ekonomikos straipsnių entropijos sumažėjo (3 lentelė), o atstumai iki kitų straipsnių padidėjo (4 lentelė). Išimtis yra geodezijos straipsnis, kurio entropija yra ≈66%, bet atstumai iki kitų straipsnių lieka dideli (4 lentelė).

Matome, kad mažiausias atstumas yra lygus 0,7 tarp 0003-001 ir 005-001 straipsnių. Pastebime, kad abu straipsniai yra to paties autoriaus. Taip pat pastebime, kad pats didžiausias atstumas yra lygus 1,9.

Pastebime, kad žodyne yra tokių žodžių, kurie matematikoje yra terminai, o kitur gali būti naudojami, kaip bendrinės kalbos žodžiai. Priklausomai nuo mokslinių straipsnių specifikos tie patys žodžiai gali būti tiek terminai, tiek ir specifinės kalbos bendriniai žodžiai. Pavyzdžiui, žodis „numerical“ gali būti tiek matematikos terminas, tiek bendrinės kalbos žodis, priklausomai nuo konteksto.

## **Apibendrinimas**

1. Reikia tobulinti programą.
2. Žodynas dar nebaigtas. Jį reikėtų analizuoti. Šiame etape jį reikia trumpinti.

## **Literatūra**

- Balys, V. *Mokslinės terminijos matematiniai modeliai ir jų taikymas leidinių klasifikavime*: daktaro disertacija, fiziniai mokslai: matematika (01P). Vilnius: Technika, 2009. 101p
- Lavcel, J. Matematikos mokslinių straipsnių informacinė analizė: 12-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas- Lietuvos ateitis“, Vilnius, 2009 m. balandžio 9 d.
- Сэлтон, Г. 1973. Автоматическая обработка, хранение и поиск информации. Москва. [Salton, G. 1968. Automatic information organization and retrieval. London.]

## **MODELLING OF CLASSIFICATIONS OF MATHEMATICS SCIENTIFIC PAPERS**

**J. Šliachtovič, A. Šimonėlis, J. Lavcel**

### **Summary**

We analyse scientific mathematical articles and try to find the range of publication. Made key words dictionary and did scientific articles information analysis: ascertained mathematical terms which were subsumed in scientific articles, a number statistics and amount rate, key words entropy and we calculated “distances” of articles from one to another. We did a calculation in the used program testing.

## GENETINIŲ ALGORITMŲ PANAUDOJIMAS POLINIŲ PAMATŲ OPTIMIZAVIMUI

Juozas Kaunas

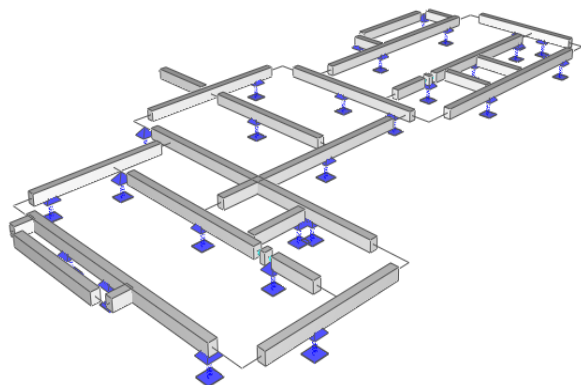
Vilniaus Gedimino technikos universitetas  
el. p. juozas.kaunas@gmail.com

**Anotacija.** Straipsnyje pristatomas tyrimas, kurio metu nagrinėjamas vienas iš globalaus optimizavimo uždavinių – polių padėčių optimizavimas rostverkuose. Tikslų funkcija imama didžiausia atraminė reakcija, kuri gali kilti bet kuriame iš polių. Projektavimo kintamieji – polių koordinatės. Paprastumo dėlei laikoma, kad visi nagrinėjamo rostverko poliai turi vienodas charakteristikas. Sprendimo metu tikslo funkcija minimizuojama. Idealią tikslo funkcijos reikšmę galima nustatyti teoriškai, bet nėra žinomos tą reikšmę atitinkančios polių koordinatės. Optimizavimui parenkamas genetinis algoritmas. Atliekamas tyrimas, kaip optimizavimo rezultatas priklauso nuo genetinio algoritmo parametrų. Gauti rezultatai papildomai palyginami su rezultatais gautais naudojant atsiktinę paiešką.

**Reikšminiai žodžiai:** poliniai pamatai, globalus optimizavimas, genetiniai algoritmai.

### Įvadas

Poliniai pamatai (dar vadinami rostverkais arba sijynais) (1 pav.) yra labai paplitę šiuolaikinėje statyboje. Statant tokius pamatus galima sutaupyti betono, sumažinti darbų apimtį.



1 pav. Polinių pamatų schemos pavyzdys

Polinius pamatus sudaro poliai ir jungiančiosios sijos. Optimizuojant polinius pamatus, galima siekti optimizuoti polių skaičių, betono tūrį arba bendrus statybų kaštus. Šiame straipsnyje aprašomas pirmas iš aukščiau išvardintų optimizavimo variantų – polių skaičiaus optimizavimas. Šis uždavinys yra ypač aktualus, kai statyba vyksta ant silpnų gruntų. Šiuo atveju reikia naudoti labai ilgus polius, kurių gamyba yra brangi. Tokiu būdu, jeigu optimizavimas leidžia sutaupyti net ir kelis polius, tai jau gali duoti svarią ekonominę naudą.

Sprendžiant uždavinį laikoma, kad visos pamatų apkrovos yra žinomos iš anksto. Taip pat priimame, kad visi poliai turi vienodas charakteristikas. Šiuo atveju

polių pasiskirstymas bus idealus, jeigu jie bus apkrauti vienodai. Tokiu būdu, padalinus visų pamatus veikiančių apkrovų sumą  $R_{sum}$  iš leistinosios poliaus reakcijos  $R_{allowed}$  ir paėmus nuo santykio sveikąją dalį iš viršaus, gausime optimalų polių skaičių  $N_{ideal}$ :

$$N_{ideal} = \left\lceil \frac{R_{sum}}{R_{allowed}} \right\rceil, \quad (1)$$

Sprendžiamas optimizavimo uždavinys gali būti suformuluotas taip:

$$\min_{x \in D} R_{max}(x), \quad (2)$$

čia  $R_{max}$  – maksimali reakcijos jėga, galinti kilti bet kuriame iš polių,  $x$  – projektavimo kintamieji (polių koordinatės),  $D$  – galima pamatų konfiguracija.

### Tikslo funkcijos skaičiavimas

Tikslo funkcijos skaičiavimui naudojama FORTRAN programa, sukurta Vilniaus Gedimino technikos universiteto teorinės mechanikos katedroje (programos autorius – prof. habil. dr. Rimantas Belevičius). Programoje realizuotas baigtinių elementų metodas. Jungiančios sijos interpretuojamos kaip lenkiamų strypų elementai, o poliai – kaip atramos.

Lenkiamo strypo elemento standumo matrica apskaičiuojama pagal formulę:

$$[K] = \begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{12}]^T & [K_{22}] \end{bmatrix}, \quad (3)$$

Submatricių  $[K_{11}]$ ,  $[K_{12}]$  ir  $[K_{22}]$  išraiškos gali būti lengvai surastos atitinkamoje literatūroje, pavyzdžiui, (Barauskas ir kiti 2004, Spyraokos and Raftoyiannis 1997).

Toliau pateikiama submatricos  $[K_{11}]$  formulė:

$$\begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & & & & \frac{6EI_z}{L^2} \\ 0 & & \frac{12EI_y}{L^3} & & & -\frac{6EI_y}{L^2} \\ 0 & & & \frac{GJ}{L} & & \\ 0 & & -\frac{6EI_y}{L^2} & & \frac{4EI_y}{L} & \\ 0 & \frac{6EI_z}{L^2} & & & & \frac{4EI_z}{L} \end{bmatrix}, \quad (4)$$

čia  $E$  yra tamprumo modulis,  $A$  – sijos skerspjūvio plotas,  $J$ ,  $I_z$ ,  $I_y$  yra skerspjūvio inercijos momentai.

Pagrindinė statikos lygtis užrašoma pagal formulę:

$$[K]^a \{u\}^a = \{F\}^a, \quad (5)$$

čia indeksas  $a$  žymi baigtinių elementų ansamblį,  $\{u\}$  yra mazgų poslinkiai,  $\{F\}$  – aktyviosios jėgos.

Polių reakcijos jėgos apskaičiuojamos pagal formulę:

$$R_i = \sum_j [K_{ij}] u_j, \quad (6)$$

### Optimizavimo algoritmas

Atsižvelgiant į sprendžiamo uždavinio sudėtingumą ir įvertinus kitų autorių patirtį (Belevičius ir Šešok 2008), optimizavimui buvo naudojamas genetinis algoritmas (GA), kurio schema pavaizduota (2 pav.) (Goldberg 1989).

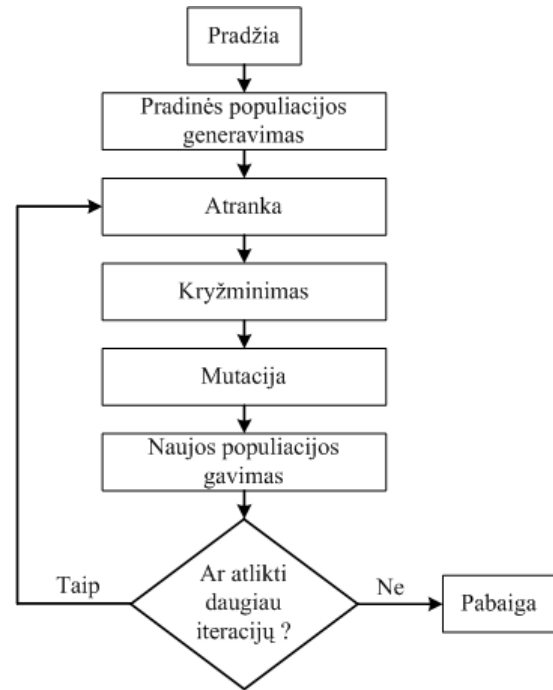
Genetinių algoritmų teorija remiasi Darvino evoliucijos teorijos principais: jeigu egzistuoja kažkokių gyvūnų ar augalų populiacija, tai didesnę šansą išgyventi ir palikti palikuonius (tuo perduodant savo genus) turi tie individai, kurių genai leido geriau prisitaikyti prie aplinkos poveikio.

Naudojant GA optimizavimui, individu laikoma projektavimų kintamųjų visuma užkoduota bitais (bendru atveju gali būti naudojami ir realūs skaičiai arba sudėtingesnės struktūros). Ar vienas individas yra „geresnis“ už kitą, parodo individą atitinkanti tikslo funkcijos reikšmė. Pavyzdžiui, jeigu sprendžiamame minimizavimo uždavinyje, tai „geresni“ individai bus tie, kuriuos atitiks mažesnės tikslo funkcijos reikšmės.

Kaip matome iš GA schemos (2 pav.), pagrindinės GA stadijos yra atranka, kryžminimas ir mutacija. Apraškyme kiekvieną iš šių stadijų.

**Atranka** stadijos metu iš esamos populiacijos atrankami individai naujai populiacijai. Galimi įvairūs at-

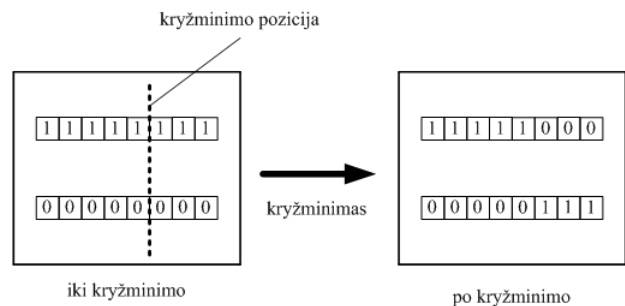
rankos mechanizmai. Vienas iš populiariausių yra ruletės principas: pradžioje, priklausomai nuo tikslo funkcijos reikšmės, kiekvienam individui priskiriama atrankos tikimybė. Kuo „geresnė“ individo tikslo funkcijos reikšmė, tuo didesnė yra jo atrankos tikimybė. Toliau metami burtai, kokį individą atrinkti. Natūralu, kad „geresni“ individai bus atrinkti dažniau, bet ir „blogesni“ individai, nors ir su maža tikimybe, bet irgi gali būti atrinkti į kitą populiaciją.



Pav. 2 Genetinio algoritmo schema

Naujai populiacijai atrinkti individai suskirstomi poromis (atsitiktine tvarka arba pagal kokią nors kitą taisyklę) ir kiekvienai porai taikomas kryžminimo operatorius.

**Kryžminimo** (pav. 3) metu du individai apsikeičia genais.

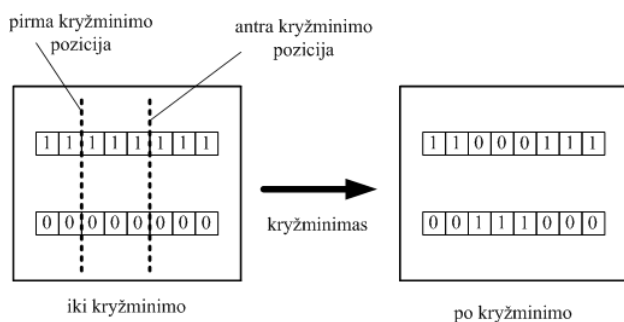


3 pav. Paprastas kryžminimo operatorius



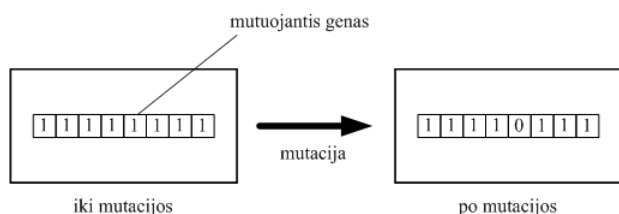
Kryžminimas vykdomas tokiu būdu: iš pradžių atsitiktinai generuojama kryžminimo pozicija. Po to imama pirmo individo dalis iki kryžminimo pozicijos ir antro individo dalis po kryžminimo pozicijos. Taip gaunamas pirmas naujas individas. Antras naujas individas gaunamas paėmus antro individo pozicija iki kryžminimo ir pirmo individo pozicija po kryžminimo.

Galima konstruoti ir sudėtingesnius kryžminimo operatorius. Pavyzdžiui, (pav. 4) parodyta kryžminimo operatoriaus schema, kai naudojamos dvi kryžminimo pozicijos.



4 pav. Sudėtingas kryžminimo operatorius

**Mutacijos** (5 pav.) operacijos metu bito reikšmė pakeičiama į priešinga. T.y. jei iki mutacijos bito reikšmė buvo „0“, tai po mutacijos ji tampa „1“, o jei iki mutacijos „1“ – tai po mutacijos „0“.



5 pav. Mutacijos operatorius

Mutacijos paskirtis – neleisti algoritmui greitai sukonverguoti į lokalų sprendinį.

Labai svarbus klausimas naudojant GA yra parametrų parinkimas. GA parametrai yra šie:

- Generacijų skaičius,
- Populiacijos dydis,
- Kryžminimo tikimybė,
- Mutacijos tikimybė.

Pirmas iš aukščiau išvardintų parametrų – generacijų skaičius – reiškia, kiek GA populiacijų bus sugeneruota sprendimo eigoje. Natūralu, kad kuo daugiau generacijų, tuo ilgiau vyksta paieška ir didėja tikimybė surasti geresnį sprendinį. Bet iš kitos pusės, didelis gene-

racijų skaičius gali reikšti per ilgą, inžinieriui nepriimtina skaičiavimų laiką. Todėl dažnai šis parametras naudojamas, kaip GA veikimo stabdymo kriterijus, t.y. sugeneravus tam tikrą populiacijų skaičių GA veikimas nutraukiamas ir pateikiamas geriausias per tą laiką surastas sprendinys.

Kiti parametrai (populiacijos dydis, kryžminimo bei mutacijos tikimybės) įtakoje GA veikimą. Dažniausiai šie parametrai parenkami eksperimentiniu būdu. Toliau straipsnyje bus pateikti optimizavimo rezultatai naudojant GA su skirtingais parametrų rinkiniais.

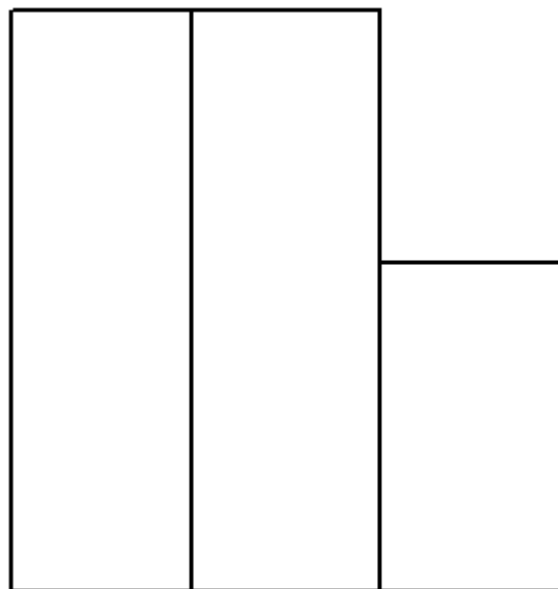
### Programinė realizacija

Uždaviniui spręsti buvo sukurta programa į kurią įeina:

- Valdančioji programa, realizuota C++ kalba.
- Tikslų funkcijos skaičiavimo modulis, realizuotas FORTRAN kalba. Modulis suderintas su valdančiąja programa per DLL bibliotekas.
- Genetinių algoritmų biblioteka, realizuota C++ kalboje. Biblioteka pasiekama Internetiniu adresu: <http://lancet.mit.edu/ga/>. [Žiūrėta 2010.04.15]

### Skaitinis pavyzdys

Optimizavimui buvo pasirinktas rostverkas, kurio bendras sijų ilgis yra 52,9 m., o minimalus atramų kiekis – 18. Schematiškai šis rostverkas pavaizduotas paveiksle žemiau.



6 pav. Optimizuojamo rostverko topologija

Kaip jau buvo minėta, uždavinys buvo sprendžiamas, naudojant GA su skirtingais parametrais. Parametrų reikšmės pateiktos lentelėje apačioje.

**1 lentelė.** GA parametrai

	Populiacijos dydis	Kryžminimo tikimybė, %	Mutacijos tikimybė, %
GA_1	25	50	5
GA_2	10	80	15
GA_3	40	90	5

Su kiekvienu GA parametrų rinkiniu buvo atliekama 30 nepriklausomų atsitiktinių bandymų. Tiek eksperimentų reikėjo atlikti, nes GA yra stochastinis algoritmas. Todėl siekiant objektyviai ištirti konkrečią GA realizaciją (pašalinti statistinį triukšmą), reikia būtinai atlikti keliasdešimt bandymų. Tuomet bus galima teigti, kad kitas tyrėjas, naudojant tą patį algoritmą, atlikus tiek pat bandymų turės gauti panašų rezultatą.

Visais atvejais tikslo funkcijos reikšmė buvo skaičiuojama 5000 kartų. Kadangi visų nagrinėjamų GA parametrų rinkinių populiacijos dydis yra skirtingas, GA\_1 atveju buvo atliekama 200 generacijų, GA\_2 atveju – 500, GA\_3 atveju – 125 ( $200 \cdot 25 = 500 \cdot 10 = 125 \cdot 40 = 5000$ ).

Uždavinys taip pat buvo išspręstas naudojant atsitiktinę paiešką (AP). Kaip ir GA atveju buvo atliekama 30 nepriklausomų atsitiktinių eksperimentų kiekviename eksperimente skaičiuojant tikslo funkcijos reikšmę po 5000 kartų.

Sprendimo rezultatai pateikti lentelėje 2.

**Lentelė 2.** Sprendimo rezultatai

Eksp. numeris	AP	GA_1	GA_2	GA_3
1	153.283	137.37	156.37	140.168
2	162.12	138.463	144.724	148.584
3	130.339	135.678	159.337	159.158
4	140.62	145.158	137.011	154.826
5	162.062	140.056	140.287	144.592
6	148.765	141.299	154.167	122.244
7	162.313	138.775	152.543	147.639
8	159.435	142.718	156.74	145.385
9	159.438	135.287	158.429	149.007
10	163.662	142.759	160.976	144.552
11	165.414	140.62	142.415	144.181
12	147.115	150.016	145.869	154.206
13	150.383	142.112	139.43	139.756
14	162.975	129.355	148.64	142.92
15	160.15	134.312	172.626	149.032
16	164.054	143.76	136.53	164.102
17	140.623	134.918	141.058	159.212
18	137.78	147.09	140.162	141.955
19	146.513	134.894	149.649	135.269
20	155.135	147.463	137.003	151.611
21	165.078	129.731	157.116	139.528
22	159.355	139.457	161.021	152.358

23	151.48	144.593	140.172	140.86
24	139.017	141.057	150.496	138.506
25	156.606	128.276	153.703	145.214
26	144.654	152.689	151.026	133.774
27	140.785	146.376	161.912	148.682
28	158.535	144.292	160.545	139.267
29	161.604	144.859	156.983	141.727
30	153.283	151.906	140.697	141.308

Lentelėje 3 pateiktos vidutinės ir geriausios kiekvienu algoritmu surastos reikšmės.

**Lentelė 3.** Rezultatų statistika

	Vidurkis	Minimumas	Idealus sprendinys
AP	153,63	130,34	104,12
GA_1	140,84	128,28	
GA_2	150,25	136,53	
GA_3	145,32	122,24	

Kaip matome, GA rezultatai pakankamai stipriai priklauso nuo GA parametrų. Blogiausiu atveju surastas sprendinys (136,53) skiriasi nuo idealaus sprendinio (104,12) 31 %, o geriausias surastas sprendinys (122,24) – 17,4%. Taip pat galime pastebėti, kad GA\_2 atveju surastas minimalus sprendinys (136,53) ir blogesnis net už atsitiktinę paiešką surastą sprendinį (130,34), kas byloja apie tai, kad GA\_2 atveju parametrai parinkti netinkamai.

Turime konstatuoti, kad naudojamas standartinis GA neleido išspręsti uždavinio iki galo, nes 17,4% skirtumas nuo idealaus sprendinio yra per didelis. Norint gauti geresnius rezultatus, reikia daugiau laiko skirti GA parametrų parinkimui. Taip pat reikia stengtis modifikuoti GA, kad jame būtų įvertinta sprendžiamo uždavinio specifika, kas leistų gauti geresnį rezultatą per trumpesnę laiką. Dar vienas rezultatų pagerinimo būdas gali būti didelio našumo skaičiavimų panaudojimas.

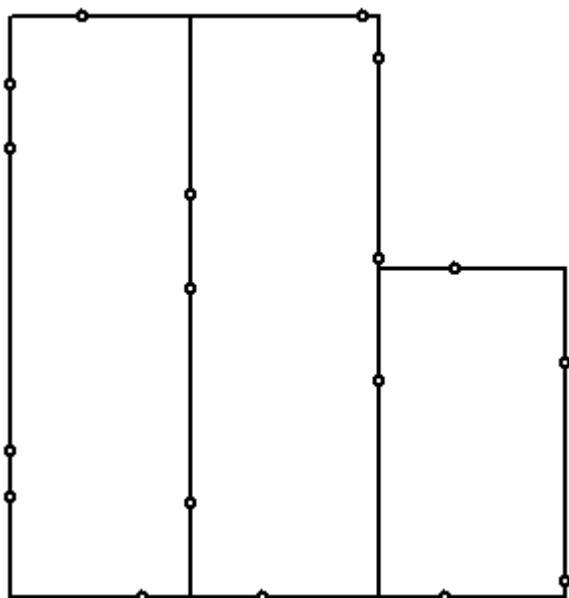
Geriausią surastą sprendinį atitinkančios koordinatės pateiktos lentelėje 4.

**Lentelė 4.** Geriausią surastą sprendinį atitinkančios koordinatės

Poliaus numeris	Koordinatė X	Koordinatė Y
1	0.000	-1.115
2	0.000	-2.171
3	0.000	-7.219
4	0.000	-7.981
5	3.025	-2.937
6	3.025	-4.517
7	3.025	-8.096
8	6.150	-0.696
9	6.150	-4.024
10	6.150	-6.035
11	9.250	-5.748
12	9.250	-9.382
13	1.211	0.000
14	5.880	0.000

15	7.420	-4.200
16	2.228	-9.650
17	4.207	-9.650
18	7.240	-9.650

Grafiškai, geriausias surastas sprendinys pavaizduotas paveiksle 7.



Pav. 7 Geriausias surastas sprendinys

## OPTIMIZATION OF GRILLAGES USING GENETIC ALGORITHMS

J. Kaunas

### Summary

In this article the study of one global optimization problem, optimization of pile positions in grillages, is presented. As an objective function the maximum support reaction that can arise in any of the piles is chosen. Design variables are coordinates of the piles. For the sake of simplicity of the task the assumption that all the piles of grillage have the same characteristics is taken. The objective function should be minimized. An ideal value of the objective function can be calculated theoretically, but the positions of the piles corresponding to this function value are not known. The genetic algorithm is employed for the optimization. The influence of genetic parameters of the algorithm on the results is investigated and the suitable values for parameters are determined. Obtained results are compared with the corresponding results of random search.

### Išvados

1. GA veikimo rezultatas priklauso nuo GA parametrų.
2. Standartinis GA ne visiškai tinka nagrinėjamam uždaviniui – norint gauti geresnius rezultatus, reikia bandyti modifikuoti GA, įvertinant uždavinio specifiką.

### Padėkos

Dėkoju dr. Dmitrij Šešok už pagalbą rengiant straipsnį.

### Literatūra

- Barauskas, R.; Belevičius, R.; Kačianauskas, R. 2004. Baigtinių elementų metodo pagrindai. Vilnius. Technika. 612 p.
- Belevičius, R.; Šešok, D. 2008. Global optimization of grillages using genetic algorithms. *Mechanika*. 6(74): 38-44.
- Goldberg, D. E. 1989. *Genetic algorithms in search, optimization and machine learning*. New-York. Addison-Wesley. p. 59-70.
- Spyrakos, C.; Raftoyiannis, J. 1997. *Linear and nonlinear finite element analysis in engineering practice*. Pittsburgh. Algor Publishing Division. 498 p.

## IŠLIKIMO ANALIZĖS METODŲ TAIKYMAS DRAUDIMO SRITYJE

**Kristina Jucikaitė**

*Vilniaus Gedimino technikos universitetas*

*el. p. KJucikaite@gmail.com*

**Anotacija.** Draudimo kompanijų veiklos principas - prisiimti apdraustųjų rizikas. Tuo atveju, kai apdraustųjų rizikos yra skirtingos, tikslinga, atsižvelgiant į rizikos laipsnį, pritaikyti atitinkamas draudimo įmokas, todėl labai svarbu nustatyti ir įvertinti požymių, įtakančių draudiminio įvykio atsiradimą, svarbą. Taikydami išlikimo analizės metodus siekėme nustatyti, kokie požymiai (objekto buvimo vieta, objekto tipas, konstrukcija, draudimo suma) lemia draudiminio įvykio atsiradimą ir kas įtakoja išlikimą. Nustačius pagrindinius riziką lemiančius požymius buvo įvertinti objektų grupių rizikos laipsniai. Gauti rezultatai leidžia vertinti draudimo įmonės klientus, prognozuoti ir modeliuoti draudiminiuosius įvykius.

**Reikšminiai žodžiai:** Išlikimo analizė, rizika, požymiai, draudimas, draudiminiai įvykiai.

### Įvadas

Būtų sunku rasti suaugusį žmogų, kuris bent kartą savo gyvenime nebūtų pasinaudojęs draudimo paslaugomis. Vairuodami automobilį draudžiamės privalomu civilinės atsakomybės draudimu, vykdami į kelionę – sveikatos draudimu, o norėdami apsaugoti savo turtą – būsto ir namų turto draudimu. Draudėjui rūpi kuo pigiau apsidrausti, o draudikui – įvertinti draudžiamąją rizikos laipsnį ir pritaikyti atitinkamą draudimo įmoką. Vienas iš būdų, leidžiantis vertinti klientus ir prognozuoti įvykius, yra išlikimo analizė (*survival analysis*).

Išlikimo analizė – statistinis metodas, leidžiantis analizuoti įvykius ir įvykių laiką. Ji plačiai naudojama tiek socialiniuose, tiek gamtos moksluose. Kadangi šis metodas taikomas skirtingose mokslo srityse, todėl turi keletą pavadinimų: įvykio istorijos analizė (*event history analysis*) – sociologijoje, patikimumo analizė (*reliability analysis*) – inžinerijoje, trukmės analizė (*duration analysis*) – ekonomikoje ir kitus. Tuo tarpu išlikimo analizė yra geriausiai žinomas ir plačiausiai naudojamas pavadinimas (Allison 1995).

Kitaip nei logistinė regresija, kuri tiria tik bendras įvykio tikimybes nepaisant įvykio kitimo laike, diskretaus laiko išlikimo analizė gali būti lengvai pritaikyta laike kintantiems kintamiesiems. Diskretaus laiko išlikimo analizės modelis leidžia lengvai atlikti, tiek nuoseklius, tiek nenuoseklius, intensyvumo funkcijos skaičiavimus kiekviename diskretaus laiko taške (Muthen *et al.* 2005).

Šiame darbe išlikimo analizės metodai buvo pritaikyti draudimo įmonės veiklos analizei. Tyrimui naudoti ne gyvybės draudimo, bet nekilnojamojo turto draudimo sutarčių duomenys. Remiantis išlikimo analizės metodais siekėme nustatyti rizikos funkcijų dėsningumus ir rizikingiausias draudiminio laikotarpio mėnesius bei ištirti kokie požymiai įtakoja draudiminio įvykio atsiradimą ir nustatyti rizikos laipsnį. Gauti rezultatai padės draudimo kompanijai vertinti klientus ir prognozuoti įvykius.

Šio darbo tikslas - nustatyti ir įvertinti požymių, įtakančių draudiminio įvykio atsiradimą, svarbą.

Tyrimo objektas – vienos Lietuvos draudimo įmonės 2007-2008 metais pasirašytų sutarčių būsto ir namų turto draudimo duomenys: draudimo objektų, draudiminių įvykių ir draudėjų tipai ir požymiai.

Tyrimo metodai – taikytas išlikimo analizės modelis, skaičiavimai atlikti naudojantis SAS programa.

### Išlikimo analizės metodai

Taikant išlikimo analizę stebimieji apibūdinami dviem dydžiais: būkle ir išlikimo trukme. Būklė gali įgyti dvi reikšmes: 1 – laiko momentu  $T$  stebimasis neišliko ir 0 – laiko momentu  $T$  apie stebimąjį turime nepilną informaciją. Atsitiktinis dydis  $T$  žymi išgyventą laiką (Leonavičienė 2009). Šiame darbe laikas bus

matuojamas mėnesiais (laikoma, kad vienas mėnuo lygus 30 dienų). Visi stebėjimai išlikimo analizėje skirstomi į dvi grupes: pilnus (*completed*) ir nepilnus (*censored*).

Draudimo sutartis su draudiku paprastai sudaroma vieneriems metams. Jei stebėjimai pasibaigė (užregistruoti draudiminiai įvykiai), t. y. stebimieji „neišgyveno“ viso draudimo laikotarpio – tai stebėjimai vadinami pilnais. Jei stebėjimai teikia tik dalinę informaciją (išgyventas visas draudiminis laikotarpis be įvykių, nutraukta draudimo sutartis anksčiau laiko), t. y. stebimu laiko momentu stebimieji „išliko“, tačiau apie tolimesnę jų būklę mes informacijos neturime – tai tokie stebėjimai vadinami nepilnais.

Nepilni stebėjimai savo ruožtu skirstomi į kairinius stebėjimus (*left censoring*) ir dešiniuosius stebėjimus (*right censoring*). Stebėjimas vadinamas dešiniu jei jo išlikimo trukmė netrumpesnė nei nustatytas laikas, t. y. jei jis išgyveno iki tam tikro laiko, o stebėjimas vadinamas kairiniu, jei jo išlikimo trukmė trumpesnė nei nustatytas laikas, t. y. jei jis neišgyveno iki tam tikro laiko (Allison 1995).

Išlikimo analizėje naudojamos trys svarbiausios funkcijos: išlikimo (*survival*) funkcija, intensyvumo (*hazard*) funkcija ir rizikos (*cumulative hazard*) funkcija.

*Išlikimo funkcija* parodo tikimybę, kad stebimasis išgyvens ilgesnį laiko tarpą negu  $t$  ir yra priešinga pasiskirstymo funkcijai:

$$S(t) = 1 - F(t) = P(T \geq t), \quad (1)$$

čia:  $F(t)$  – pasiskirstymo funkcija;  $T$  – išgyventas laikas;  $t$  – tam tikras laikas (Allison 1995).

*Intensyvumo funkcija* parodo tikimybę neišlikti laiko momentu  $t$ , kai žinome, kad iki to laiko momento stebimasis išgyveno ir apibrėžiama lygybe:

$$h(t) = \frac{p(t)}{S(t)} = P(T = t | T \geq t), \quad (2)$$

čia  $p(t)$  – tikimybių tankio funkcija (Allison 1995).

Diskrečiuoju atveju (Cox *et al.* 1984, Leonavičienė 2009):

$$h(t) = 1 - \frac{S(t+1)}{S(t)}, \quad (3)$$

Ryšys tarp intensyvumo ir išlikimo funkcijų diskrečiuoju atveju (Cox *et al.* 1984, Leonavičienė 2009):

$$S(t) = \sum_{i=0}^{t-1} (1 - h(i)), \quad (4)$$

*Rizikos funkcija* parodo stebimojo rizikos laipsnį tam tikru laiko momentu  $t$  ir apibrėžiama lygybe (Fox 2006):

$$H(t) = -\int_0^t h(x) dx = -\ln S(t), \quad (5)$$

Ryšys tarp išlikimo ir rizikos funkcijų diskrečiuoju atveju (Cox *et al.* 1984, Leonavičienė 2009):

$$H(t) = -\ln S(t) = -\sum_{i=0}^{t-1} \ln(1 - h(i)), \quad (6)$$

Išlikimo analizės metodą galima suskirstyti į tris pagrindinius etapus: išlikimo funkcijos apskaičiavimas ir įvertinimas, išlikimo funkcijų grupėse palyginimas, rizikos faktorių nustatymas.

*Pirmajame etape* naudojami du metodai: Kaplano Majerio (*Kaplan – Maier*) arba/ir išlikimo lentelės (*Life tables*). Greta šių metodų taip pat naudojame grafinius rezultatų atvaizdavimus. Naudojant pirmą metodą, nėra galimybės pasirinkti laiko intervalo, kuriame norime skaičiuoti t. y. rezultatai gaunami kiekviename laiko  $t$  žingsnyje. Kadangi šiame darbe pasirinkome laiką matuoti mėnesiais (nes kitaip gaunami neinformatyvūs rezultatai), išlikimo funkcijoms apskaičiuoti ir įvertinti naudojamas metodas - išlikimo lentelės. Šio metodo principas – tyrimo laiką suskirstyti į uždarus intervalus (šiuo atveju tai 30 dienų) ir juose apskaičiuoti pagrindines išlikimo analizės funkcijas, kurios padėtų nustatyti tolimesnę tyrimo eigą.

Antrajame etape naudojami du testai: Vilkokso (Gehano) (*Wilcoxon (Gehan)*) ir logaritminis – ranginis (*Log – rank*). Abu testai skirti patikrinti nulinę hipotezę – kad išlikimo funkcijos dvejose grupėse yra vienodos, t.y.  $S_1(t) = S_2(t)$  visiems  $t$ . Skaičiuojamas mažiausias reikšmingumo lygmuo su kuriuo turimiems duomenims gali būti atmesta teisinga hipotezė. Šis reikšmingumo lygmuo vadinamas  $p$  reikšme. Jei  $p < \alpha$ , tai hipotezė atmetama, priešingu atveju hipotezė priimama.  $\alpha$ - pasirinktas reikšmingumo lygmuo (Fischer 2008).

Trečiajame etape rizikos faktorius nustatome analizuodami Kokso regresijos (*Cox regression*) modelius. Kokso modelio sudarymo eigoje naudojame šiuos testus: Valdo (*Wald*) testą – naudojamas modelio koeficientų reikšmių nustatymui, ir LML (*Log-minus log*) grafikas – naudojamas modelio tinkamumo nustatymui.

Kokso regresija - tai statistinis metodas, kuriuo siekiama atskleisti ne išlikimo funkcijas, o intensyvumo funkcijos priklausomybę nuo tam tikrų požymių. Kokso regresijos dėka galima vertinti rizikingiausius požymius ir nustatyti rizikos laipsnį (Leonavičienė 2009).

Kokso regresija nereikalauja, kad būtų pasirinktas tam tikras tikimybinis skirstinys, norint pavaizduoti išlikimo laiką. Dėl šios savybės Kokso regresija vadinama pusiau parametrine (*semiparametric*). Šio tipo regresija leidžia nagrinėti ir nuo laiko priklausančius požymius, taigi požymiai gali kisti viso stebėjimo metu (Allison 1995).

Intensyvumo funkcija modelyje priklauso nuo dviejų narių: pradinės intensyvumo funkcijos  $h_0(t)$  ir nuo rizikos veiksnių  $X = (X_1, X_2, X_3, \dots, X_m)$ :

$$h(t) = h_0(t) e^{\beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots + \beta_m X_m}, \quad (7)$$

čia  $\beta = (\beta_1, \beta_2, \beta_3, \dots, \beta_m)$  - nežinomi regresijos modelio koeficientai. Šie koeficientai nustatomi remiantis dalinio tikėtumo metodu (*partial likelihood*), nes dažniau skaičiuojamos tikimybės tik neišlikusiųjų, o ne visų stebimųjų. Kadangi intensyvumo funkcija yra netiesinė, tai modelio koeficientų reikšmės randamos iteraciniais metodais. Pradinė intensyvumo funkcija – tai intensyvumo funkcija, kai visų kintamųjų reikšmės yra lygios nuliui, t.y.  $X = (0, 0, 0, \dots, 0)$  (Allison 1995).

Valdo testas - tai statistinis metodas, kuris skirtas patikrinti ar visi intensyvumo funkcijos koeficientai, Kokso proporcingųjų intensyvumų modelyje, yra lygūs nuliui, t.y. ar nagrinėjamas požymis neturi įtakos rizikos funkcijai. Kuo didesnė Valdo statistikos reikšmė, tuo reikšmingesnis nagrinėjamas požymis (Allison 1995).

LML grafikas – tai išlikimo funkcijų grafikas, kuris padeda nustatyti ar Kokso proporcingosios rizikos modelis yra tinkamas. Šiame grafike pateikiamos  $\ln(-\ln S(t))$  reikšmės. Jeigu išlikimo kreivės nesusikerta, o geriausia kai jos yra lygiagrečios, ir atstumai tarp kreivių bet kuriuo laiko momentu yra pastovūs, tai proporcingosios rizikos modelis yra tinkamas (Allison 1995).

### **Išlikimo analizės metodų taikymo rezultatai**

Nagrinėjami 8608 draudimo polisai (sutartys), kuriuose apdrausta 17420 objektų (įvykių). Pirmoji draudimo sutartis sudaryta 2007 m. gegužės 23 d., o paskutinioji – 2008 m. gruodžio 31 d. Kadangi draudimo polisai (sutartys) sudaryti tiek 2007, tiek 2008 metais, išlikimo analizės metodas taikomas atskirai šioms dviem grupėms - 2007 metais apdraustiems 5918 objektų (įvykių) ir 2008 metais apdraustiems 11502 objektams (įvykiams).

Mus domina tikimybė, kad apdraustasis išgyvens visą draudimo laikotarpį be draudiminių įvykių, atsižvelgiant į tam tikrus požymius (objekto buvimo vieta, objekto tipas, konstrukcija, draudimo suma).

Objekto buvimo vietą nusako Lietuvos miestai ir rajonai. Kadangi nagrinėjami daugiau negu 150 skirtingi miestai ir rajonai, nuspręsta juos suskirstyti į dešimt Lietuvos apskričių: Vilniaus, Panevėžio, Kauno, Tauragės, Šiaulių, Alytaus, Klaipėdos, Utenos, Marijampolės ir Telšių. Objekto tipą apibūdina keturi faktoriai: pastatas, patalpa, namų turtas ir vertybės, civilinė atsakomybė. Konstrukciją nusako tokie trys faktoriai: karkasinis/rąstinis (medinis), medinis apmūrytas, mūrinis. Draudžiami objektai pagal draudimo sumą

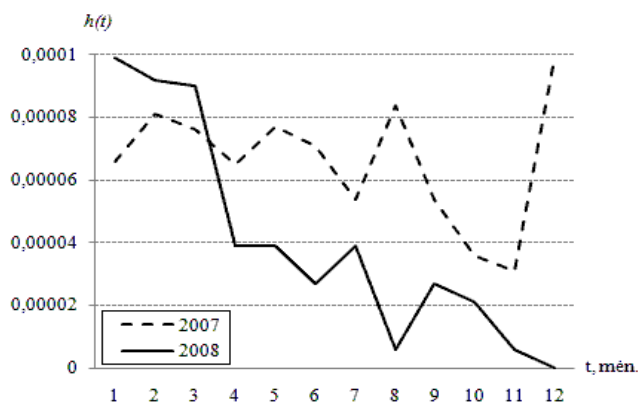
suskirstyti į tris grupes: maža draudimo suma 0-250.000 Lt, vidutinė draudimo suma – 250.000-500.000 Lt, didelė draudimo suma – daugiau nei 500.000 Lt.

*Pirmojo etapo rezultatai.* Sudarius išlikimo lenteles pagal draudimo polisų (sutarčių) sudarymo metus pastebime, kad 2007 metais sudariusiems sutartis didžiausia rizika yra antrame, aštuntame ir dvyliktame draudiminio laikotarpio mėnesyje, kai tuo tarpu 2008 metais sudariusiems sutartis didžiausia rizika yra pirmame, antrame ir trečiame mėnesyje (paryškintos intensyvumo funkcijų reikšmės 1 lentelėje).

**1 lentelė.** Apdraustųjų, sudariusių sutartis 2007-2008 m., išlikimo lentelė

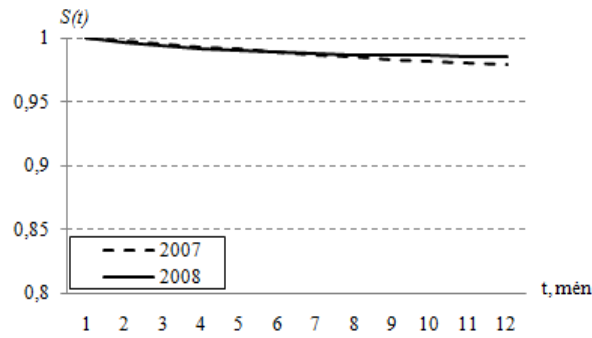
Intervalai [pradžia, pabaiga)		Išlikimo funkcija 2007	Intensyvumo funkcija 2007	Išlikimo funkcija 2008	Intensyvumo funkcija 2008
0	31	1,0000	0,000066	1,0000	<b>0,000099</b>
31	61	0,9980	<b>0,000081</b>	0,9969	<b>0,000092</b>
61	91	0,9955	0,000076	0,9942	<b>0,000090</b>
91	121	0,9933	0,000065	0,9915	0,000039
121	151	0,9914	0,000077	0,9904	0,000039
151	181	0,9891	0,000071	0,9892	0,000027
181	211	0,9870	0,000054	0,9884	0,000039
211	241	0,9854	<b>0,000084</b>	0,9872	0,000006
241	271	0,9829	0,000054	0,9871	0,000027
271	301	0,9813	0,000036	0,9862	0,000021
301	331	0,9802	0,000031	0,9856	0,000006
331	361	0,9793	<b>0,000099</b>	0,9854	0,000000
361	.	0,9764	.	0,9854	.

Taigi, intensyvumo funkcijų dėsningumo, tarp sudariusių sutartis 2007 ir 2008 metais, nepastebime (1 pav.).



**1 pav.** Apdraustųjų, sudariusių sutartis 2007-2008 m., intensyvumo funkcijos

Išlikimo funkcijos, tiek sudariusiems sutartis 2007 tiek 2008 metais, visais draudiminio laikotarpio mėnesiais artimos vienetui, o tai rodo didelę išlikimo tikimybę (2 pav.). 2007 metais buvo sudaryta 5918 sutarčių; jų tarpe neišlikusių – 135. Taigi, išlikimo procentas – 97,72 %. 2008 metais buvo sudaryta 11502 sutartys, o neišlikusių objektų – 164. Šiuo atveju išlikimo procentas – 98,57 %.



**2 pav.** Apdraustųjų, sudariusių sutartis 2007-2008 m., išlikimo funkcijos

Analogiškos išlikimo lentelės ir išlikimo bei intensyvumo funkcijų grafikai buvo sudaryti atskirai kiekvienam požymiui: miestui, objektui, konstrukcijai ir draudimo sumai.

*Antrojo etapo rezultatai.* Atlikus Vilkokso (Gehano) ir logaritminį – ranginį testus keturiose grupėse, sudarytose pagal draudžiamo objekto buvimo vietą, objekto tipą, objekto konstrukciją ir objekto draudimo sumą, pastebime, kad bėgant laikui išlikimo funkcijos statistiškai reikšmingai skiriasi visose grupėse tiek 2007 tiek 2008 metais (2 – 5 lentelės). Skirtumai matyti ir nubraižius LML grafikus (jie nepateikti, kadangi šiuo atveju yra nenaudingi) – grafikai kertasi ir nėra lygiagretūs.

**2 lentelė.** Apdraustųjų, sudariusių sutartis 2007-2008 m., išlikimo funkcijų palyginimas (miestai)

Testas 2007	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė	Testas 2008	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė
Log-Rank	44,395	10	<0,001	Log-Rank	51,3781	10	<0,001
Wilcoxon	44,080	10	<0,001	Wilcoxon	51,3105	10	<0,001

**3 lentelė.** Apdraustųjų, sudariusių sutartis 2007-2008 m., išlikimo funkcijų palyginimas (objektai)

Testas 2007	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė	Testas 2008	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė
Log-Rank	65,1151	3	<0,001	Log-Rank	50,717	3	<0,001
Wilcoxon	65,0999	3	<0,001	Wilcoxon	50,894	3	<0,001

**4 lentelė.** Apdraustųjų, sudariusių sutartis 2007-2008 m., išlikimo funkcijų palyginimas (konstrukcijos)

Testas 2007	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė	Testas 2008	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė
Log-Rank	33,9016	3	<0,001	Log-Rank	29,5480	3	<0,001
Wilcoxon	33,7203	3	<0,001	Wilcoxon	29,5764	3	<0,001

**5 lentelė.** Apdraustųjų, sudariusių sutartis 2007-2008 m., išlikimo funkcijų palyginimas (draudimo sumos)

Testas 2007	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė	Testas 2008	Statistikos reikšmė	Laisvės laipsnių sk.	p reikšmė
Log-Rank	8,3801	2	0,0151	Log-Rank	13,7768	2	0,001
Wilcoxon	8,2171	2	0,0164	Wilcoxon	13,8338	2	0,001



Kadangi visais atvejais  $p < \alpha = 0,05$ , tai išlikimo funkcijos tarp nagrinėjamų požymių grupių statistiškai reikšmingai skiriasi.

*Trečiojo etapo rezultatai.* Be testų, skirtų išlikimo funkcijų grupėse palyginimui, galima patikrinti ar nagrinėjama grupė yra susijusi su išlikimo laiku, t. y. įvertinti ar ji turi įtakos išlikimo laikui. Tam tikslui skaičiuojama statistikos reikšmė kiekvienai grupei, ignoruojant kitas likusias. Šis testas yra naudingas, nes jį atlikus galima „atmesti“ netinkamas grupes, t. y. tokias, kurios neturi įtakos išlikimo laikui. Atlikus šiuos testus (pateikti tik Viloksono, nes kaip matėme jie skiriasi nežymiai), 2007 metais sudariusiems sutartis, pastebime, kad požymių grupės konstrukcija ir draudimo suma, neturi įtakos išlikimo laikui ( $p = 0,2008 > 0,05$  ir  $p = 0,4405 > 0,05$ ) kai tuo tarpu didžiausią įtaką turi draudžiamojo objekto buvimo vieta ( $p < 0,001 < 0,05$ ) ir objekto tipas ( $p = 0,0223 < 0,05$ ) (6 lentelė).

**6 lentelė.** Apdraustųjų, sudariusių sutartis 2007m., požymių efekto tyrimas

<i>Požymis</i>	<i>Laisvės laipsnių sk.</i>	<i>Statistikos reikšmė</i>	<i>p reikšmė</i>
MIESTAS	1	34,2391	<0,0001
OBJEKTAS	2	5,2254	0,0223
KONSTRUKCIJA	3	1,6367	0,2008
DRAUD_SUMA	4	0,595	0,4405

2008 metais sudarytoms sutartims pastebime, kad požymių grupė konstrukcija, neturi įtakos išlikimo laikui ( $p = 0,1787 > 0,05$ ), o didžiausią įtaką turi draudžiamojo objekto buvimo vieta ( $p < 0,001 < 0,05$ ), draudimo suma ( $p = 0,0045 < 0,05$ ) ir objekto tipas ( $p = 0,0071 < 0,05$ ) (7 lentelė).

**7 lentelė.** Apdraustųjų, sudariusių sutartis 2008m., požymių efekto tyrimas

<i>Požymis</i>	<i>Laisvės laipsnių sk.</i>	<i>Statistikos reikšmė</i>	<i>p reikšmė</i>
MIESTAS	1	38,1185	<0,0001
DRAUD_SUMA	2	8,0822	0,0045
OBJEKTAS	3	7,2562	0,0071
KONSTRUKCIJA	4	1,8087	0,1787

Kai jau esame nustatę rizikingiausius požymius, galime įvertinti ir rizikos laipsnį, t. y., kaip ir kuriais laiko tarpais labiausiai rizikos funkciją įtakoja nustatyti rizikingiausi požymiai. Tai galima nustatyti sudarius Kokso proporcingosios rizikos modelius.

Kadangi prieš tai atlikę Viloksono (Gehano) ir logaritminį – ranginį testus bei nubraižę LML grafikus nustatėme, kad išlikimo funkcijos apsibrėžtose grupėse statistiškai reikšmingai skiriasi, atsižvelgę į stebimų ir neišlikusiųjų skaičių sudarome naujas grupes.

Sutartims, pasirašytoms 2007 metais, buvo sudarytos dvi naujos grupės: draudžiamo objekto buvimo vieta apibūdinama dviem faktoriais – Viniiaus miestu ir kitais miestais (Panevėžys, Kaunas ir t.t.), o objekto tipą apibūdina tokie du faktoriai: patalpa ir kiti tipai (pastatas, namų turtas ir vertybės, civilinė atsakomybė).

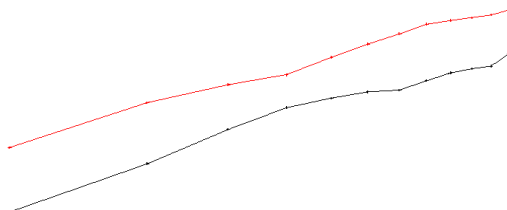
2008 metais pasirašytoms draudimo sutartims buvo sudarytos trys naujos grupės: draudžiamo objekto buvimo vieta apibūdinama dviem faktoriais – Viniiaus miestu ir kitais miestais (Panevėžys, Kaunas ir t.t.), objekto tipą apibūdina du faktoriai: patalpa ir kiti tipai (pastatas, namų turtas ir vertybės, civilinė atsakomybė), o objekto draudimo sumą apibūdina tokie faktoriai: maža draudimo suma 0-250.000 Lt ir kitos draudimo sumos (vidutinė draudimo suma – 250.000-500.000 Lt ir didelė draudimo suma – daugiau nei 500.000 Lt).

Pirmiausia pagal 2007 metais sudarytų sutarčių duomenis ištirsime kaip išlikimą įtakoja draudžiamo objekto buvimo vieta (8 lentelė). 8 lentelėje nustatomi modelio koeficientai ir pateikiama Valdo statistikos reikšmė. Lentelėje esanti reikšmė  $e^{\beta}$  rodo, kad kitų miestų rizika yra 2 kartus mažesnė negu Vilniaus.

**8 lentelė.** Apdraustųjų 2007 m. modelio testas 1

<i>Požymis</i>	$\beta$	<i>Valdo statistika</i>	<i>p reikšmė</i>	$e^\beta$
MIESTAS2 Vilnius - 1 Kiti - 0	0,759	18,8783	<0,001	2,136

Nubraižius LML grafiką (3 pav.) įsitikiname proporcingosios rizikos modelio tinkamumu: išlikimo kreivės nesusikerta ir atstumai tarp kreivių bet kuriuo laiko momentu yra pastovūs. Tai proporcingosios rizikos modelis yra tinkamas.

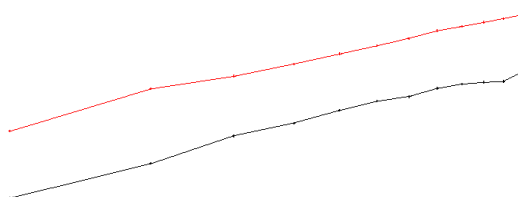


**3 pav.** Apdraustųjų 2007 m. LML grafikas modelio testui 1

Analogiškai gauname, kad 2007 metais sudarytoms sutartims patalpų rizikos laipsnis yra 3,3 karto didesnis negu kitų objektų (9 lentelė).

**9 lentelė.** Apdraustųjų 2007 m. modelio testas 2

<i>Požymis</i>	$\beta$	<i>Valdo statistika</i>	<i>p reikšmė</i>	$e^\beta$
OBJEKTAS2 Patalpa - 1 Kiti - 0	1,18671	47,0784	<0,001	3,276

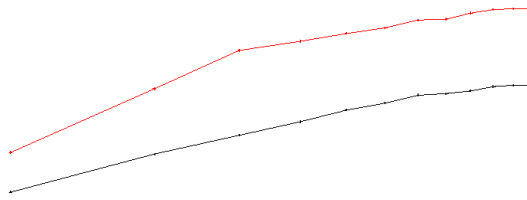


**4 pav.** Apdraustųjų 2007 m. LML grafikas modelio testui 2

Dabar proporcingosios rizikos Kokso modelį pritaikysime sutartims, sudarytoms 2008 metais. Šiuo atveju Vilniaus miesto rizikos laipsnis yra 2,5 karto didesnis negu kitų miestų (10 lentelė), o patalpų - 2,5 karto didesnis negu kitų objektų (11 lentelė).

**10 lentelė.** Apdraustųjų 2008 m. modelio testas 1

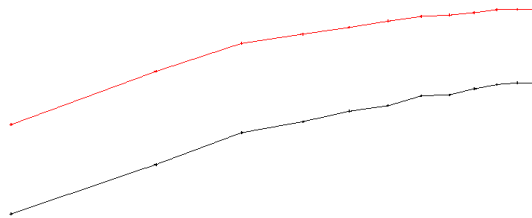
<i>Požymis</i>	$\beta$	<i>Valdo statistika</i>	<i>p reikšmė</i>	$e^\beta$
MIESTAS2 Vilnius - 1 Kiti - 0	0,93308	33,5735	<0,001	2,542



**5 pav.** Apdraustųjų 2008 m. LML grafikas modelio testui 1

**11 lentelė.** Apdraustųjų 2008 m. modelio testas 2

<i>Požymis</i>	$\beta$	<i>Valdo statistika</i>	<i>p reikšmė</i>	$e^\beta$
OBJEKTAS2 Patalpa - 1 Kiti - 0	0,92436	32,9481	<0,001	2,5

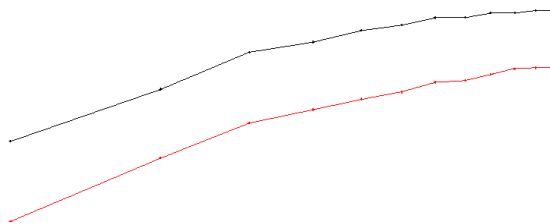


**6 pav.** Apdraustųjų 2008 m. LML grafikas modelio testui 2

12 lentelėje matome, kad mažų draudimo sumų rizika yra 2 kartus mažesnė negu visų kitų.

**12 lentelė.** Apdraustųjų 2008 m. modelio testas 3

<i>Požymis</i>	$\beta$	<i>Valdo statistika</i>	<i>p reikšmė</i>	$e^\beta$
DRAUD_SUMA2 Maža - 1 Kitos - 0	-0,61169	11,8705	0,0006	0,542



**7 pav.** Apdraustųjų 2008 m. LML grafikas modelio testui 3

## Išvados

1. Išlikimo funkcijos grupėse, sudarytose pagal *objekto buvimo vietą, objekto tipą, konstrukciją ir draudimo sumą*, skiriasi.

2. Nustatyti rizikingiausi požymiai, įtakoiantys draudiminio įvykio atsiradimą, ir įvertinti jų rizikos laipsniai:

✓ *objekto buvimo vieta;*

*Vilniaus miesto* rizikos laipsnis vidutiniškai 2 kartus didesnis negu *visų kitų miestų*.

✓ *objekto tipas;*

Objekto *patalpa* rizikos laipsnis vidutiniškai 3 kartus didesnis negu *visų kitų objektų*.

✓ *objekto draudimo suma ;*

2008 metais sudarytoms sutartims *mažų draudimo sumų* rizika yra 2 kartus mažesnė negu *visų kitų*.

2007 metais sudarytoms sutartims draudimo suma įtakos išlikimui neturi.

## Literatūra

Allison, Paul D. 1995. *Survival Analysis Using SAS: A Practical Guide*. Cary, NC: SAS Institute Inc.

Cox, D. R.; Oakes, D. 1984. *Analysis of Survival Data*. Chapman and Hall, London (in Russian).

Fischer, I. 2008. *Survival Analysis* [interaktyvus]. Lecture Notes [žiūrėta 2010 m. kovo 28 d.]. Prieiga per internetą: <[http://www.stat.wisc.edu/~ifischer/bmi511/Lecture\\_Notes/8\\_-\\_Survival\\_Analysis/](http://www.stat.wisc.edu/~ifischer/bmi511/Lecture_Notes/8_-_Survival_Analysis/)>.

Fox, J. 2006. *Introduction to Survival Analysis* [interaktyvus]. Lecture Notes [žiūrėta 2010 m. kovo 28 d.]. Prieiga per internetą: <<http://socserv.mcmaster.ca/jfox/Courses/soc761/survival-analysis.pdf>>

Leonavičienė, T. 2009. *Studijų proceso analizė išlikimo analizės metodais*. Pedagogika, 93, 98-110.

Muthen, B.; Masyn, K. 2005. Discrete-Time Survival Mixture Analysis. *Journal of Educational and Behavioral Statistics*. 30(1), 27-58.

## SURVIVAL ANALYSIS IN INSURANCE

### K. Jucikaitė

#### Summary

Principle of insurance companies is to take risk of insured. If risks of insured are different it is advisable to adjust insurance premium considering into risk rate so it is very important to estimate and evaluate factors from which depends if insured will fail. Purpose is to estimate which factors (object location, object type, structure and insurance premium) have main impact on failing and what factors influence on surviving using survival analysis. After estimating the most risky factors it is required to measure risk rate which helps to estimate customers and forecast occurrence.

**Keywords:** survival analysis, risk, factors, insurance, occur.

## MATEMATINIS MODELIAVIMAS

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“,  
įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

# H. MARKOWITZ INVESTICINIO PORTFELIO KONSTRAVIMAS KRIZĖS SĄLYGOMIS

**Lina Kubiliūte**

*Vilniaus Gedimino Technikos Universitetas*

[kubiliute.lina@gmail.com](mailto:kubiliute.lina@gmail.com)

**Anotacija.** Šiame straipsnyje konstruojami optimalus investiciniai portfeliai remiantis šiuolaikine H. M. Markowitz investicinio portfelio konstravimo teorija. Portfeliai modeliuojami imant statistinius duomenis iš Lietuvos akcijų biržos 2009 metu antrojo pusmečio ir prognozuojama į 2010 metų pirmąjį mėnesį. Pasirenkamos dešimtys Lietuvos firmų akcijos. Portfelijų svoriai randami sprendžiant netiesinį programavimo uždavinį *SAS OR* programinės įrangos pagalba, statistiniais duomenys apdorojai *MS Office* pagalba.

### **Įvadas**

Pastaruosius du metus akcijų kainos pastoviai krito, pelno normos minusinės, taigi, kodėl kas nors turėtų rekomenduoti pirkti akcijas tokiu metu kaip šis – krizės metu? Nepaisant daugelio socialinių, politinių ir ekonominių veiksnių per paskutinius du amžius akcijų rinkos grąža yra nuspėjama ilgam terminui. Sunku pasirinkti investuoti, kai visi akcijas pardavinėja. Reikia turėti daug

drąsos pasirinkti priešingai. Bet remiantis žymaus ekonomisto žodžiais „investuotojas turi būti godus, kai kiti atsargūs, ir turi būti atsargus, kai kiti godūs“.

Akcijų ar kitų vertybinių popierių rinkinukas, kurį planuojama įsigyti ar jau yra įgytas, vadinamas vertybinių popierių portfeliu. Taigi, vertybinių popierių portfelyje gali būti vienos bendrovės vertybinis popierius arba kelių bendrovių vertybiniai popieriai. Aišku, kuo mažiau vertybinių popierių yra portfelyje, tuo jis rizikingesnis. Jei portfelyje yra daugiau vertybinių popierių, tai didesnė tikimybė, kad atpigusių akcijų vertės sumažėjimą kompensuos pabrangusios akcijos, t. y. bendra portfelio rinkos vertė labai nesumažės.

## **H. M. Markowitz modelis ir komponentų parinkimas**

H. Markowitz modelis yra klasikinis investicinio portfelio radimo modelis. Šį modelį šeštajame dešimtmetyje pasiūlė H. Markowitz. Jis remiasi matematiniais skaičiavimais ir juo pagrįsta šiuolaikinė investicinio portfelio teorija. Anksčiau investuotojai suvokė rizikos koncepciją, tačiau jos nematavo. H. Markowitz pasiūlė efektyvaus portfelio terminą. Efektyvus portfelis apibūdinamas kaip portfelis, kuris turi mažiausią riziką duotam pelningumui arba didžiausią pelningumą duotam rizikos lygiui.

Skaičiavimams atlikti naudojama 10-ties Lietuvos įmonių akcijų kainos, kurios pasirenkamos remiantis AB „Finasta“ rekomendacijomis.

## Matematinis modelis stochastinio programavimo atveju

Portfeliai konstruojami generuojant pasirinktų įmonių akcijų kainas, iš kurių skaičiuojama sukonstruotų portfelių grąža. Galima daryti prielaidą, kad grąžos yra atsitiktiniai dydžiai, pasiskirstę pagal konkretų pastovių charakteristikų tikimybinį skirstinį. Tuomet jų tikėtinos grąžos ir standartiniai nuokrypiai gali būti apskaičiuojami iš istorinių akcijų kainų duomenų.

Tegul  $c_{ij}$  yra  $j$  – tosios firmos akcijų kaina  $i$ -ąją dieną.

Pelno normas skaičiuosime pagal formulę:

$$a_{ij} = \left( \frac{c_{ij}}{c_{i-kj}} - 1 \right) \cdot 100 \% \quad (1)$$

(mano atveju  $k = 1$ ).

Pasirinktą portfeliui

$$P = (x_1, x_2, \dots, x_n) \quad (2)$$

sumas

$$a_i = \sum_{j=1}^n a_{ij} x_j \quad (3)$$

vadinsime portfelio grąžos reikšmėmis (čia  $a_{ij}$  yra pelno normas laikotarpio, kurio duomenis naudojame konstruodami Portfelį). Šio portfelio grąžos realizacijomis sekančiame laikotarpyje vadinsime sumas

$$b_i = \sum_{j=1}^n b_{ij} x_j \quad (4)$$

čia  $b_{ij}$  yra  $j$  – tosios firmos akcijų pelno norma  $i$  – ają sekančio laikotarpio dieną.

Portfelio  $P$  vidutinė grąža  $V(P)$  pusės metų laikotarpyje skaičiuojamas pagal formulę

$$V(P) = \sum_{j=1}^n \overline{a_j} x_j \quad (5)$$

čia  $\overline{a_j} = \sum_{i=1}^{m_1} a_{ij}$  yra  $j$  – tosios firmos akcijų pelno normos vidurkis

stebėtam  $m_1$  dienos ilgio laikotarpiui.

Jei pelno normos yra priklausomi atsitiktiniai dydžiai, Portfelio grąžos dispersija yra

$$D(P) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij}^2 x_i x_j \quad (6)$$

čia  $\sigma_{ij}$  yra  $i$  – tosios ir  $j$  – tosios firmų pelno normų kovariacija.

Atsitiktinę akcijų pelno normą  $j$  – tajai firmai pažymėkime  $n_j$ .

Tarkime, kad atsitiktinis dydis (portfelio grąža)

$$N(P) = \sum_{j=1}^n n_j x_j \quad (7)$$

yra pasiskirstęs pagal normalųjį dėsnį su vidurkiu (5) ir dispersija (6).

Išsprendžiamame stochastinio programavimo uždavini

$max W$

$$P\left(\sum_{j=1}^n n_j x_j \geq W\right) \geq 1 - \alpha,$$



$$\sum_{j=1}^n x_j = 1,$$

$$x_j \geq 0, j=1,2,\dots,n. \quad (8)$$

surandame portfelį, kurio grąža, su tikimybe  $1-\alpha$ , turėtų būti nemažesnė už skaičių  $W$ . Remdamiesi prielaida apie normalumą gauname separabelinio programavimo uždavinį :

$max W$

$$\sum_{j=1}^n a_j x_j - u_\alpha \sqrt{\sum_{i=1}^n \sum_{j=1}^n \sigma_{ij}^2 x_i x_j} \geq W,$$

$$\sum_{j=1}^n x_j = 1,$$

$$\geq 0, j=1,2,\dots,n. \quad (1)$$

Čia  $u_\alpha$  yra normaliojo skirstinio kritinė reikšmė eilės  $\alpha$ .  $P_{1-\alpha}$  pažymėsime portfelį, kurio komponentės yra uždavinio sprendinys.  $u_\alpha$  reikmės - skirtingoms normaliojo skirstinio kritinėms reikšmėms  $\alpha$ :

Lentelė nr.1: Normaliojo skirstinio kritinės reikšmės.

P	0,51	0,55	0,6	0,65	0,7	0,75
U $\alpha$	0,0251	0,1257	0,2533	0,38532	0,5244	0,6745

P	0,8	0,85	0,9	0,95	0,999
U $\alpha$	0,8416	1,0364	1,2816	1,6449	3,0902

### Praktinė dalis – portfelių konstravimas:

Visi matematinio modelio veiksmi buvo realizuoti naudojant SAS OR programą, o gautų duomenų apipavidalinimui panaudotas MS Office Excel paketas. Portfeliai konstruojami naudojant statistinius duomenis apie Lietuvos biržos akcijų kainas(LTL), naudojant 2009 antrojo pusmečio statistinius duomenis. Pagal anksčiau aptartą investavimo strategiją pasirenkame iš Lietuvos akcijų biržos 10 Lietuvos įmonių.

Atlikus skaičiavimus su pateiktais duomenimis ir skirtingais pasiklovimo lygmenimis, gauti tokie rezultatai:

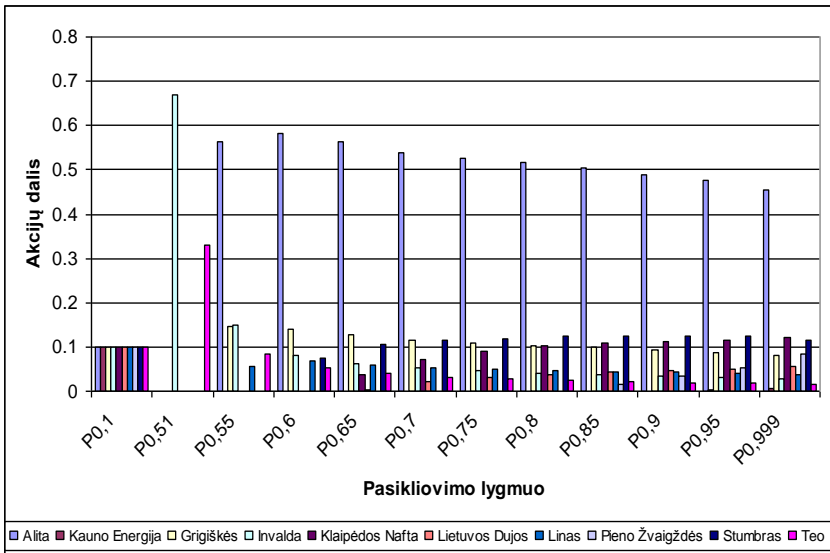
Lentelė nr.3: Sukonstruoti portfeliai ir jų vidurkiai bei standartiniai nuokrypiai.

	Alita	Kauno Energija	Grigiškės	Invalda	Klaipėdos Nafta	Lietuvos Dujos	Linas	Pieno Žvaigždės	Stumbras	Teo
P0,1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
P0,51	0	0	0	0.66	0	0	0	0	0	0.33
P0,55	0.56	0	0.14	0.14	0	0	0.05	0	0	0.08
P0,6	0.58	0	0.14	0.08	0	0	0.06	0	0.07	0.05
P0,65	0.56	0	0.12	0.06	0.03	0.03	0.06	0	0.10	0.04
P0,7	0.54	0	0.11	0.05	0.07	0.02	0.05	0	0.11	0.03
P0,75	0.52	0	0.10	0.04	0.09	0.03	0.05	0	0.1	0.02
P0,8	0.51	0	0.10	0.04	0.10	0.03	0.04	0	0.12	0.02
P0,85	0.50	0	0.09	0.03	0.11	0.04	0.04	0.02	0.12	0.02

	Alita	Kauno Energija	Grigiškės	Invalda	Klaipėdos Nafta	Lietuvos Dujos	Linias	Pieno Žvaigždės	Stumbras	Teo
P0,9	0.48	0.01	0.09	0.03	0.11	0.04	0.04	0.04	0.12	0.02
P0,95	0.47	0.03	0.08	0.03	0.11	0.05	0.04	0.05	0.12	0.01
P0,999	0.45	0.05	0.08	0.02	0.12	0.05	0.03	0.08	0.11	0.01

	6 mėnesiai	
P0,1	Vidurkis	Std
P0,51	0.072408	1.76327
P0,55	0.442265	4.30832
P0,6	0.256458	1.55194
P0,65	0.222201	1.35674
P0,7	0.198887	1.2838
P0,75	0.177954	1.2363
P0,8	0.165887	1.21588
P0,85	0.158148	1.20556
P0,9	0.140609	1.18679
P0,95	0.122429	1.17102
P0,999	0.105695	1.1593
	0.077677	1.14677

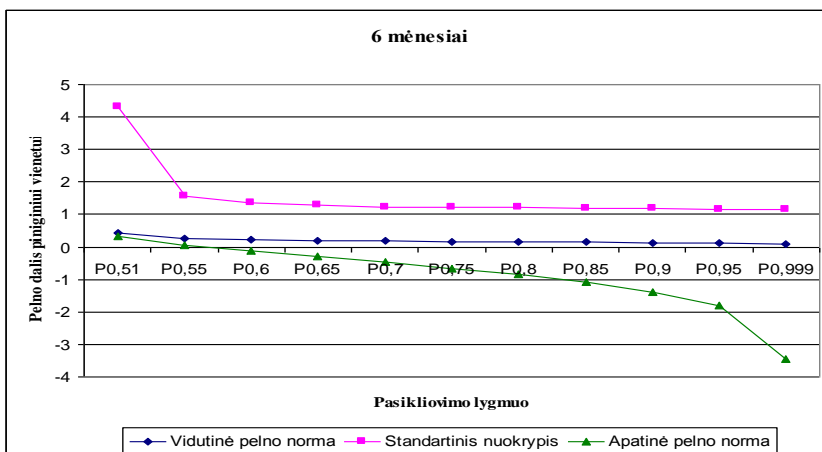
## Grafinis šių rezultatų atvaizdavimas:



1 Paveikslėlis: Sukonstruoti portfeliai.

Kaip galime pastebėti, didėjant pasiklovimo lygmeniui, akcijų portfelis tampa įvairesnis t.y. perkama įvairesnių firmų akcijos (Pvz. : kai  $P = 0,51$  perkama tik 2 iš 10 firmų akcijos, o kai  $P = 0,75$  jau perkamos net 9 firmų akcijos).  $R_{0,1}$  yra bazinis rinkos portfelis (kai visų akcijų perkama po lygiai). Taip pat galime iškirti, kad AB „Teo“ ir AB „Invalda“ akcijų siūloma rinktis visuose sukonstruotuose portfeliuose. AB „Alita“ akcijos sudaro didžiausią dalį devyniuose iš vienuolikos sukonstruotų portfelių.

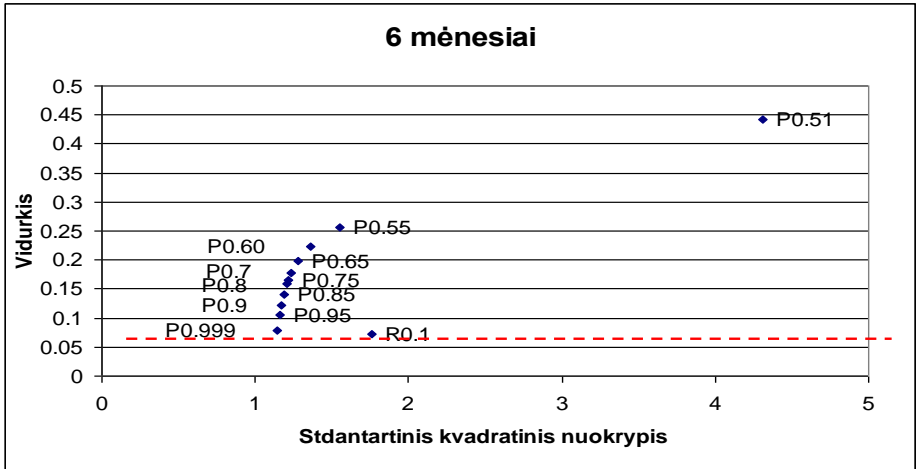
Optimalų portfelį gauname tada, kai jo vidutinė pelno norma bus didžiausia, o dispersija – mažiausia.



2 Paveikslėlis: Portfeliai pagal pasiklovimo lygmenį ir grąža.

Kaip matome iš pateiktos lentelės ir jos grafinio vaizdo, portfelio P0,6 rizika yra mažiausia. Grafiškai atvaizduota standartinio nuokrypio, vidutinės pelno normos ir apatinės pelno normos ribos priklausomybės nuo pasiklovimo lygmens per tiriamus šešis mėnesius.

Portfelių palyginimui naudojama vidurkio priklausomybė nuo standartinio nuokrypio. Kaip matome iš žemiau esančio grafiko „geriausių“ portfelių galime laikyti P0,51, nes jo vidutinė grąža yra pati didžiausia. Kita vertus, šis portfelis nėra labai patikimas, nes rizika taip pat didžiausia. „Blogiausių“ portfelių galime laikyti P0,9. Pateiktame grafike taip pat yra pavaizduotas bazinis rinkos portfelis R0,1, kuris išskiriamas punktyrine linija.



3 paveikslėlis: Portfelijų priklausomybė

## Išvados

Imant didesnę pasiklovimo lygmenį mes mažiname riziką, tuo pačiu sukonstruotas portfelis tampa įvairesnis.

Portfelijų realizacijos: P0,51 - manomas geriausias portfelis, P0,6 - subalansuotas portfelis, R0,1 -bazinis, P0,9 – manomas blogiausias. Taip pat pastebėtina, kad krintančių akcijų kainų situacija rinkoje nėra „siaubinga“, nes žvelgiant iš investuotojo pusės jis gali įsigyti akcija palankiausiomis kainomis su perspektyva jas parduoti brangiau sekančiame laikotarpyje.

## Literatūros sąrašas

Kancerevičius, G. Finansai ir investicijos. 2006

Markowitz, H. M. 1952. Portfolio selection. The journal of finance, 7, p. 77-91

Vakrinienė, S.; Misevičius, G. 2007. tiesinio ir netiesinio optimizavimo modeliai investiciniam portfeliui pasirinkti.

Vakrinienė, Sigutė. Operacijų tyrimas programine įranga SAS/OR. Vilnius, Technika, 2003.

AB "Finasta", Investavimo strategija, Vilnius, 2009m. Spalis – 2010m. Spalis.

Liekmanienė, A. Lietuvos finansų rinkos investicinio portfelio gražos ir rizikos analizė. Kaunas, 2005.

Žilinskij, G. H. Markowitz teorijos plėtra siekiant adekvatesnio portfelio sudarymo ir valdymo.

Kraujalis, Š. Lietuvos bankų investavimo kriterijai. KTU.

## **H. Markowitz investment portfolio construction during crisis period**

The Master's thesis practices on stochastic programming model, in order to select the investment portfolio from the efficient portfolios set. Inefficient portfolios are modeled while searching mixed strategy in the matrix game, the elements of which are stochastic values. Models are tested following the statistical data of stock values.

## SKAITINIS DIFUZIJOS PROCESŲ MODELIAVIMAS HETEROGENINĖSE DVIMATĖSE IR TRIMATĖSE TERPĖSE

Mantas Tarvydas

magistrantas

Vilniaus Gedimino technikos universitetas

El. paštas: mtarvydas@gmail.com

**Anotacija.** Modeliuojami difuzijos procesai heterogeninėje terpėje. Nagrinėjama terpė susideda iš dviejų skirtingų medžiagų, kurių šilumos laidumo savybės yra skirtingos. Apžvelgta teorinė dalis ir gauti eksperimentiniai rezultatai. Remiantis universaliu paketu „ANSYS“, atlikti modeliavimo ir vizualizacijos darbai. Gauti rezultatai leidžia aprašyti, kaip šiluma pasiskirsto, nagrinėjamoje terpėje. Remiantis baigtinių tūrių metodu, sukurta ir parašyta originali programa, kurios pagalba galima modeliuoti difuzijos procesą dvimatėje heterogeninėje terpėje. Gautų tiesinių algebrinių lygčių sistemai spręsti naudojamas matricinis perkelties algoritmas.

**Reikšminiai žodžiai:** difuzija, skaitinis modeliavimas, heterogeninė terpė, šilumos laidumas, baigtinių tūrių metodas, matricinis perkelties algoritmas.

### Įvadas

Daugelis svarbių gamtos ar technikos reiškinių turi savo matematinius modelius, kurie gali būti užrašomi diferencialinėmis lygtimis. Vienas iš tokių reiškinių yra difuzijos procesas, kuris apibūdina dalelių judėjimą iš didesnės koncentracijos vietos link mažesnės koncentracijos. Dar šį procesą galime vadinti šilumos pernešimu ar laidumu.

Šiluminis laidumas – medžiagos savybė praleisti šilumą, t. y. termoizoliacinė medžiagos savybė. Šią savybę apibūdina šilumos laidumo koeficientas  $k$ . Kuo mažesnė  $k$  vertė, tuo geresnėmis izoliacinėmis savybėmis pasižymi medžiaga. Šiluminis laidumas yra savaiminis šiluminės energijos atidavimo procesas per medžiagą iš aukštesnės temperatūros regiono į žemesnės. Procesas stengiasi išlyginti temperatūrinį skirtumą.

Skaitinis šilumos laidumo proceso matematinis modeliavimas yra sudėtingas procesas. Reikia sudaryti diferencialinę lygtį, turėti kraštines sąlygas ir žinoti šilumos laidumo koeficientus. Be to, reikia pasirinkti skaitinius metodus, kurių pagalbą galėsime analizuoti pasirinktą matematinį modelį.

Samarskii (2001) teigia, kad skaitiniai eksperimentai padeda geriau suvokti realiame pasaulyje vykstančius fizikinius procesus.

Tokiems procesams modeliuoti yra sukurta nemažai universalių programų, kurios remiasi baigtinių skirtumų arba baigtinių elementų metodais. Šiame darbe naudojamas vienas iš tokių universalių programinių paketų

„ANSYS“, kurio pagalba, galima modeliuoti ir nagrinėti šiluminius uždavinius.

Šio darbo tikslas – naudojantis skaitiniais metodais, parašyti programą, kuri modeliuotų šilumos laidumo procesą dvimatėje heterogeninėje terpėje. Gautus eksperimentinius rezultatus palyginti su programinio paketo „ANSYS“ gautais rezultatais.

### Uždavinio formulavimas

Dvimačio stacionaraus šilumos laidumo uždavinio matematinis modelis yra diferencialinis uždavinys, kurį sudaro diferencialinė lygtis (1).

$$-\frac{\partial}{\partial x_1}(k(x_1, x_2) \frac{\partial U}{\partial x_1}) - \frac{\partial}{\partial x_2}(k(x_1, x_2) \frac{\partial U}{\partial x_2}) = f(x_1, x_2), \quad (1)$$

čia:  $U = U(x_1, x_2)$  – temperatūra (ieškoma funkcija),  $(x_1, x_2) \in \Omega$  sričiai, kurioje sprendžiamas uždavinys;  $k$  – šilumos laidumo koeficientas;  $f$  – išorinis šilumos šaltinis (Čiegis 2003).

Šilumos laidumo koeficientas  $k(x_1, x_2)$  (2) priklauso nuo dviejų medžiagų iš kurių yra sudaryta nagrinėjama terpė. Tai yra iš I medžiagos ir II medžiagos, todėl užrašome taip:

$$k(x_1, x_2) = \begin{cases} k_1, & (x_1, x_2) \in I \text{ medžiagos sričiai} \\ k_2, & (x_1, x_2) \in II \text{ medžiagos sričiai} \end{cases} \quad (2)$$



Uždavinio matematinėje formuluotėje kraštinės sąlygos yra reikalingos sprendinio egzistavimo ir vienaties užtikrinimui. Nagrinėjant skaitinius metodus kraštinės sąlygos yra būtinos ir labai svarbios, nes be jų nebūtų galima vienareikšmiškai išspręsti nagrinėjamo uždavinio. Modeliuodami naudosime pirmojo tipo kraštinės sąlygas, kai žinome temperatūrą visame srities kontūre  $U|_{\Gamma} = U_a$ .

### Uždavinio skaičiavimo metodas

Norit skaitiniai metodais spręsti nagrinėjamą uždavinį, turime atlikti šiuos tris žingsnius:

- Nagrinėjamoje srityje įvesti diskretųjį tinklą;
- Išvestines pakeisti baigtiniais skirtumais, o diferencialinę lygį - (tiesine) algebrinių lygčių sistema;
- Išspręsti gautą tiesinių algebrinių lygčių sistemą;

[Samarskii (1997)]

Baigtinių tūrių metodas (angl. *Finite volume method*) universalus baigtinių skirtumų schemų sudarymo metodas. Jis taikomas tada, kai diferencialinio uždavinio trūkio koeficientai turi trūkio taškų Čiegis (2003).

Nagrinėjame bendrą elipsinį uždavinį:

$$\begin{cases} -\frac{\partial}{\partial x_1}(k(x_1, x_2)\frac{\partial U}{\partial x_1}) - \frac{\partial}{\partial x_2}(k(x_1, x_2)\frac{\partial U}{\partial x_2}) = f(x_1, x_2), & U \in \Omega, \\ U|_{\Gamma} = U_a, & U \in \Omega. \end{cases} \quad (3)$$

Tarsime, kad yra išpildytos eliptiškumo sąlygos  $k(x, y) \geq k_0 > 0$ .

Tuomet uždavinį (3) galime aproksimuoti tokia schema [Čiegis (2003)]:

$$\begin{cases} -(a_1 y_{\bar{x}_1})_{x_1} - (a_2 y_{\bar{x}_2})_{x_2} = \varphi_{ij}, & x_1, x_2 \in \omega_h, \\ y(x_1, x_2) = \mu(x_1, x_2), & x_1, x_2 \in \partial\omega_h. \end{cases} \quad (4)$$

Koeficientus  $a_{1,2}$  ir  $\varphi$  apskaičiuojame baigtinių tūrių metodu:

$$a_1 = k(x_{i-0.5}, x_{2j}) = \frac{2k(X_{i\pm 1, j})k(X_{ij})}{k(X_{i\pm 1, j}) + k(X_{ij})},$$

$$a_2 = k(x_i, x_{2j-0.5}) = \frac{2k(X_{i, j\pm 1})k(X_{ij})}{k(X_{i, j\pm 1}) + k(X_{ij})},$$

$$\varphi_{ij} = f(x_i, x_{2j}).$$

Nagrinėjamu dvimačių atveju pasirinkta schema apibrėžia tiesinių algebrinių lygčių sistemą, kuria galima užrašyti tokiu pavidalu:

$$-c_{1,ij}y_{i-1,j} - c_{2,ij}y_{i+1,j} - c_{3,ij}y_{i,j-1} - c_{4,ij}y_{i,j+1} + c_{5,ij}y_{ij} = \varphi_{ij}, \quad (5)$$

$$1 \leq i \leq N_1, 1 \leq j \leq N_2,$$

koeficientus  $c_1, c_2, c_3, c_4$  ir  $c_5$  apskaičiuojame pagal šias formules:

$$c_{1,ij} = \frac{k_{i-0.5,j}}{h^2}, \quad c_{2,ij} = \frac{k_{i+0.5,j}}{h^2},$$

$$c_{3,ij} = \frac{k_{i,j-0.5}}{h^2}, \quad c_{4,ij} = \frac{k_{i,j+0.5}}{h^2},$$

$$c_{5,ij} = c_{1,ij} + c_{2,ij} + c_{3,ij} + c_{4,ij}.$$

### Lygčių sistemos sprendimo metodas

Tokioms lygčių sistemoms spręsti yra sukurta daugybė algoritmų. Šiuos algoritmus galime skirstyti į dvi grupes:

- Tiesioginiai metodai.
- Iteraciniai metodai.

Nagrinėjamos sistemos atveju (5), pasirinktas tiesioginis metodas – matricinės perkelties metodas.

Tai yra Gauso (angl. *Gauss*) metodo modifikacija, kuri atsižvelgia į sprendžiamos tiesinių lygčių sistemos matricos struktūrą: kiekvienoje lygčių sistemos lygtyje yra ne daugiau negu penki koeficientai kurie nelygūs nuliui. Ši metodo modifikacija leidžia spręsti uždavinį žymiai efektyviau, nebelieka trivialių aritmetinių operacijų: daugybos iš nulio ar dviejų skaičių, lygių nuliui, sumavimo. Naudojantis šiuo metodu sumažėja aritmetinių veiksmų skaičius, taip pat ir saugomos informacijos apimtis.

Nežinomuosius  $Y$  numeruodami pagal stulpelius

$$Y = (Y_1, Y_2, \dots, Y_{N_1-1}),$$

$$Y_k = \begin{pmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{kN_2-1} \end{pmatrix}, \quad k = 1, 2, \dots, N_1 - 1,$$

gauname tiesinių algebrinių (vektorinių) lygčių sistemą

$$\begin{cases} C_1 Y_1 - B_1 Y_2 = F_1, \\ -A_i Y_{i-1} + C_i Y_i - B_i Y_{i+1} = F_i, & i = 2, 3, \dots, N, \\ -A_{N-1} Y_{N-2} + C_{N-1} Y_{N-1} = F_{N-1}. \end{cases} \quad (6)$$

Matricos  $A_i$  ir  $B_i$  yra blokais trijstrižainės, o matrica  $C_i$  yra trijstrižainė matrica

$$A_i = \begin{pmatrix} c_{1,i1} & 0 & 0 & \dots & 0 \\ 0 & c_{1,i2} & 0 & \dots & 0 \\ 0 & 0 & c_{1,i3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{1,iN_2-1} \end{pmatrix}, \quad B_i = \begin{pmatrix} c_{2,i1} & 0 & 0 & \dots & 0 \\ 0 & c_{2,i2} & 0 & \dots & 0 \\ 0 & 0 & c_{2,i3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{2,iN_2-1} \end{pmatrix},$$

$$C_i = \begin{pmatrix} c_{5,i1} & -c_{4,i1} & 0 & \dots & 0 & 0 \\ -c_{3,i2} & c_{5,i2} & -c_{4,i2} & \dots & 0 & 0 \\ 0 & -c_{3,i3} & c_{5,i3} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -c_{3,iN_2-1} & c_{5,iN_2-1} \end{pmatrix},$$

Lygčių sistemos sprendinį išreiškiame tokiu pavidalu

$$Y_i = \alpha_i Y_{i+1} + \beta_i, \quad i = 1 \dots N-2, \quad (7)$$

Čia  $\alpha_i$  -  $M \times M$  dimensijos matrica,  $\beta_i$  -  $M$  dimensijos vektorius. Matricos  $\alpha_i$  ir vektoriaus  $\beta_i$  apskaičiuojamos pagal šias formules

$$\alpha_i = C_i^{-1} B_i, \quad \beta_i = C_i^{-1} F_i.$$

Koeficientai  $\alpha_i$  ir  $\beta_i$  apskaičiuojami pagal formules

$$\alpha_i = (C_i - A_i \alpha_{i-1})^{-1} B_i,$$

$$\beta_i = (C_i - A_i \alpha_{i-1})^{-1} (F_i + A_i \beta_{i-1}).$$

Iš paskutiniosios tiesinių lygčių sistemos lygties apskaičiuojame sprendinį  $Y_{N-1}$ :

$$Y_{N-1} = (C_{N-1} - A_{N-1} \alpha_{N-2})^{-1} (F_{N-1} + A_{N-1} \beta_{N-2}).$$

Pasinaudojame formule (7) ir atgaline tvarka randame sprendinį  $Y$ .

Skaičiuojant matricinės perkelties koeficientus  $\alpha$  ir  $\beta$  atsiranda sunkumas, kad reikia skaičiuoti atvirkštines matricas  $C_i^{-1}$ . Ši procedūra gali būti sudėtinga, kai turime didelį lygčių skaičių  $N$ , todėl vietoj to, kad skaičiuoti  $X = C^{-1}B$ , galime spręsti  $CX = B$ . Šioje vietoje matrica  $C$  turi būti išskaidoma į  $L$  apatinę ir  $U$  viršutinę matricas  $(LU)x = b$ , kiekvienam stulpeliui  $X$  ir  $B$ . Šis metodas vadinasi LU sklaidos metodas.

$$[C] = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \dots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & \dots & 0 \\ \ell_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \ell_{n1} & \ell_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

Suskaidžius matricą  $C$  į  $L$  ir  $U$  matricas, pirmiausiai tiesiogine eiga sprendžiame tokią sistemą

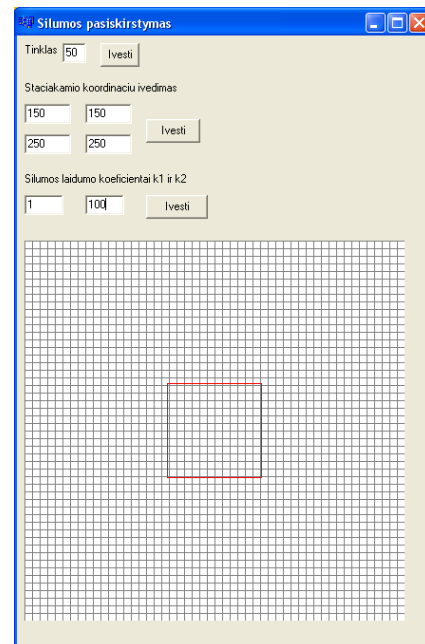
$$Ly = b,$$

antru žingsniu atbuline eiga randame nežinomuosius

$$Ux = y.$$

### Atlikti darbai ir gauti rezultatai

Naudojantis programavimo kalba C++ buvo parašyta originali programa, kurios pagalba galima modeliuoti stacionarius šilumos laidumo procesus heterogeninėje dvimatėje terpėje. Pagrindinis programos langas 1 pav., kuriame stačiakampių pagalba reikia apibrėžti, atvaizduoti nagrinėjamą sritį. Taip pat reikia nurodyti skirtingas šilumos laidumo koeficientų reikšmes, kurios priklauso dviem skirtingoms medžiagoms.



**1 pav.** Pagrindinis programos langas, kuriame reikia įsivesti nagrinėjamą sritį ir šilumos laidumo koeficientų reikšmes.  
**Fig. 1.** Main window of created program. It is need to indicate heterogeneous media and thermal conductivity.

Šioje programoje realizuoti prieš tai aprašyti metodai ir algoritmai, t.y. baigtinių tūrių metodas - diskretizuoti diferencialiniai lygčiai, taip pat matricinės perkelties metodas ir LU sklaidos metodas, kurių pagalba yra sprendžiama penkiajstrižainė tiesinių algebrinių lygčių sistema.

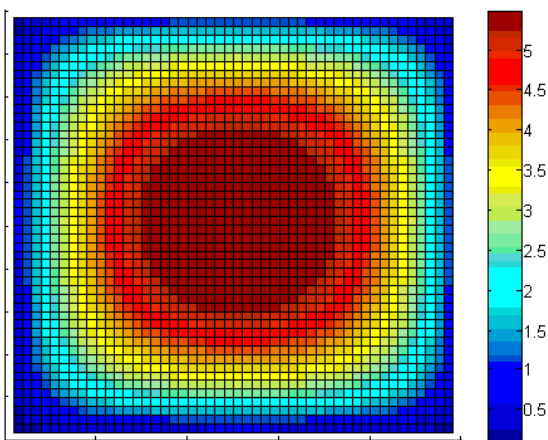
Kaip testinis uždavinys buvo sprendžiamas toks diferencialinis uždavinys (8), nagrinėjama terpė sudaryta iš dviejų skirtingų medžiagų (1 pav.).

$$\begin{cases} -\frac{\partial}{\partial x_1}(k(x_1, x_2)\frac{\partial U}{\partial x_1}) - \frac{\partial}{\partial x_2}(k(x_1, x_2)\frac{\partial U}{\partial x_2}) = 100, & U \in \Omega, \\ U|_{\Gamma} = 0, & U \in \Omega. \end{cases} \quad (8)$$

Atitinkamai paimti skirtingi šilumos laidumo koeficientai  $k_1$  ir  $k_2$ :

$$k(x_1, x_2) = \begin{cases} k_1 = 1, & (x_1, x_2) \in I \text{ medžiagos sričiai} \\ k_2 = 100, & (x_1, x_2) \in II \text{ medžiagos sričiai} \end{cases}$$

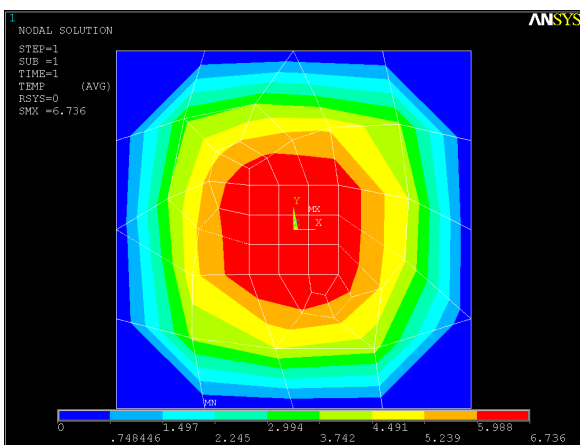
Gauti tokie skaičiavimo rezultatai 2 pav.



2 pav. parašytos programos gauti rezultatai, žingsnis  $h=0.02$

2 fig. Received results form original program, step  $h=0.02$

Naudojantis programiniu paketu ANSYS buvo modeliuojamas toks pat dvimatis stacionarus šilumos laidumo uždavinys heterogeninėje terpėje. Buvo gauti rezultatai ir atlikta vizualizacija 3 pav.



3 pav. Programa ANSYS gauti rezultatai

3 fig. Received results with software ANSYS

## Išvados

1) Peržiūrėti ir nagrinėti metodai ir algoritmai, kurių pagalba būtų galima spręsti nagrinėjamą uždavinį. Pasirinktas baigtinių tūrių metodas ir tiesioginis tiesinių algebrinių lygčių sprendimo metodas.

2) Parašyta originali programa, kurios pagalba galima modeliuoti stacionarų dvimatį šilumos laidumo uždavinį.

3) Naudojantis paketu ANSYS, buvo modeliuojami įvairūs uždaviniai. Buvo rastas temperatūrinis laukas nagrinėjamoje heterogeninėje terpėje.

4) Originaliosios programos gaunami rezultatai, gaunamai neblogai sutampa su programinio paketo ANSYS gautais rezultatais.

## Literatūra

- R. Čiegis. Diferencialinių lygčių skaitiniai sprendimo metodai. Vilnius: Technika 2003. 448 p.
- R. Čiegis, V. Būda. Skaičiuojamoji matematika. Vilnius: TEV 1997. 240 p.
- R. Čiegis, A. Ilgevičius, H. Liess, M. Meilūnas, O. Subač. Mathematical modeling and analysis. Numerical simulation of the heat conduction in electrical cables. Vilnius: Technika 2007. Volume 12, Number 4. 425-439 p.
- A. A. Samarskii. The Theory of Difference Schemes. New York: Marcel Dekker, Inc. 2001. 788 p.
- A. A. Самарский. Теория разностных схем. Москва: Наука, 1983. 616 p.
- A. Richard Barrett, *et al.* Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. 1993. 112 p. ISBN-13: 978-0-898713-28-2 / ISBN-10: 0-89871-328-5
- Jamie Trahan, Computational Time for Finding the Inverse of a Matrix: LU Decomposition vs. Naive Gaussian Elimination. Prieiga per internetą:  
[http://numericalmethods.eng.usf.edu/simulations/nbm/04sle/nbm\\_sle\\_sim\\_inversecomptime.pdf](http://numericalmethods.eng.usf.edu/simulations/nbm/04sle/nbm_sle_sim_inversecomptime.pdf)
- Methods for the Solution of Linear Systems Deriving from Elliptic PDEs. Prieiga per internetą:  
[http://ses.library.usyd.edu.au/bitstream/2123/376/5/adt-NU20010730.12021504\\_chapter\\_3.pdf](http://ses.library.usyd.edu.au/bitstream/2123/376/5/adt-NU20010730.12021504_chapter_3.pdf)
- Randall J. LeVeque. Finite Difference methods for differential Equations. AMath 585, Winter Quarter 2006, University of Washington, Version of January, 2006. 144 p. Prieiga per internetą:  
<http://www.amath.washington.edu/~rjl/pubs/am58X/am585winter06.pdf>
- SHU XU, B.S. PARALLEL SOLUTION OF BLOCK-TRIDIAGONAL SYSTEMS. 1993 m. Prieiga per internetą:  
<http://etd.lib.ttu.edu/theses/available/etd-03272009-31295007717811/unrestricted/31295007717811.pdf>
- A. Самарский, Е.С. Николаев. Методы решения сеточных уравнений. Москва: Наука, 1978. 592 p.
- A. A. Самарский. Введение в теорию разностных схем. Москва: Наука, 1971. 553 p.

# NUMERICAL SIMULATION OF DIFFUSION PROCESSES IN HETEROGENEOUS TWO-DIMENSIONAL AND THREE-DIMENSIONAL MEDIA

**Mantas Tarvydas**

Abstract

Diffusion processes simulation in heterogeneous media. This heterogeneous media consist of two different materials, and their heat transfer coefficient has different values. Gathered and overviewed theoretical part, also some experiments are made. Using universal engineering design and analyses software ANSYS were made modeling and visualization in heterogeneous two dimensional media. From received results it is seen how heat is distributed in analyzed area. Furthermore were written and created programs in C++ to analyze diffusion processes in heterogeneous two dimensional media. This program is using finite volume method and others numerical methods to solve linear equations system.

**Keywords:** diffusion, numerical simulation, heterogeneous media, thermal conductivity, finite volume method, block tridiagonal algorithm.

## MECHATRONIKOS GAMINIŲ GAMYBOS PROCESO ATRANKINĖ IR STATISTINĖ KOKYBĖS KONTROLĖ

**Martynas Karkauskas**

*Vilniaus Gedimino technikos universitetas, shaldytuv@gmail.com*

**Anotacija.** Šiame darbe nagrinėjami mechatronikos gaminių kokybės užtikrinimo metodai, naudojami gamybos proceso duomenimis ir patikimumo charakteristikomis. Metodai realizuojami Maple matematiname pakete. Pateikiama realaus proceso analizė.

**Reikšminiai žodžiai.** Statistinė kokybės kontrolė, atrankinė kontrolė, patikimumas.

### Įvadas

Įvairių produktų gamybai ir jos realizacijai tampant vis globalesniu reiškiniu, konkuruodami tarpusavyje gamintojai stengiasi didinti gaminių kokybę ir patikimumą. Šiuo metu yra pripažįstama, jog efektyviausias būdas tai padaryti yra gamybos proceso stebėjimo ir analizės metodų taikymas.[1, 723psl.]

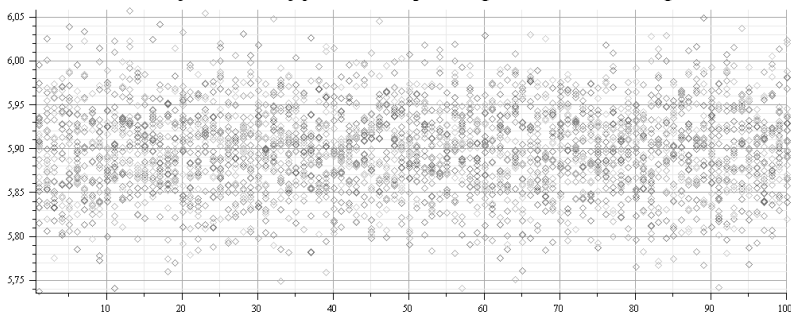
Pagrindinė statistinės kontrolės užduotis yra tampriai susijusi su gaminio patikimumo užtikrinimu. Viskas priklauso nuo konkrečios kontrolės operacijos, jeigu kontrolės metu yra galimybė aptikti netinkamą gaminį, tai statistinė kontrolė tuo pat metu yra ir patikimumo kontrolė.[2, 376psl.]

Buvo tirtas kompiuterinis transporto priemonių, jų medžiagų ir įrenginių kokybės įvertinimo metodas. Statistiniam kokybės ir kontrolės valdymo skaičiavimui ir vizualizavimui pritaikyti kompiuteriniai paketai. Nustatytas sąrašas kokybės procesams įgyvendinti. Išanalizuoti kokybės kontrolės grafikai. Pritaikytas statistinis vidurkio metodas ir atlikti kompiuteriniai skaičiavimai [6, 41psl.].

## Duomenys

Norint atlikti gamybos proceso statistinę kokybės kontrolę, pirmiausia turi būti surinkti duomenys apie gamybos procesą. Šiuo atveju pasirenkama mechatronikos gaminių viena iš charakteristikų, kuri yra nuskaitoma kontrolės metu. Pagal gaminio techninius reikalavimus ši charakteristika neturėtų viršyti 6VA, t.y. jei gaminio naudojama įtampos galia viršija leistiną, tai gamins pripažįstamas netinkamu.

Tyrimui pasirinkti 3000 gaminių, kurie yra suskirstyti į 30 partijų. Charakteristikų reikšmių pasiskirstymas pavaizduotas 1 pav.



1 pav. „Mechatronikos gaminių charakteristikos reikšmių pasiskirstymo grafikas“

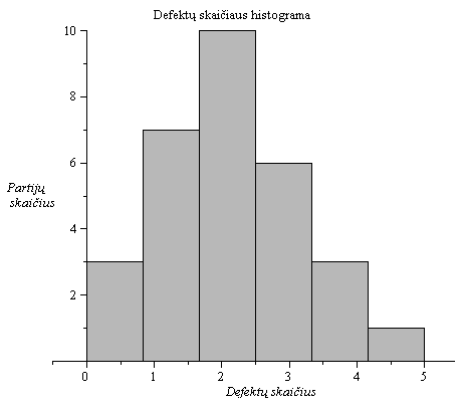
## Atrankinė kontrolė

Kai kuriose įmonėse gamyba baigiama produkcijos išleidžiamąja kontrole. Pagal jos rezultatus sprendžiama apie pagamintos produkcijos kokybės lygį ir jos kitimo dinamiką. Gamybos metu gaminiai komplektuojami vienodo dydžio  $N$  partijomis ir siunčiami atrankinei kontrolei. Į vieną partiją patenka produkcija, pagaminta apytiksliai tuo pačiu metu vienodomis sąlygomis.

Atrankinės kontrolės metu patikrinama dalis patekusios partijos.

Remiantis gautais matavimų rezultatais daroma išvada apie visą partiją: partija gali būti priimta arba atmesta.[3, 16psl.]

Konkrečiu atveju yra iširta 30 partijų po 100 gaminių, todėl galima nustatyti, koks yra netinkamų gaminių skaičiaus partijose pasiskirstymas (tai atvaizduota 2 pav.).



2pav. „Netinkamų gaminių partijose skaičiaus pasiskirstymas“

Iš viso aptikti 62 defektiniai gaminiai, tačiau atliekant atrankinę kontrolę, netikrinami visi partijos gaminiai, todėl būtina nustatyti kokį skaičių gaminių reikia tikrinti ir kada partiją laikyti tinkama, o kada ne. Dėl šių priežasčių yra randamas  $q$  – defektiškumo lygis. Jis padeda apskaičiuoti defektinių gaminių skaičių  $d$ , kuris atrankinės kontrolės dydyje –  $n$ , atitiks nustatytą defektiškumo lygį:

„Antrankinės kontrolės planų tikimybių lentelė“

$d$	$N=100$	$n=20$	$q=0.0207$	$N=100$	$n=40$	$q=0.0207$
	$P_{ND}^{nd}$			$P_{ND}^{nd}$		
0		0.658			0.434	
1		0.278			0.366	
2		0.056			0.151	
3		0.007			0.04	
4		$6 \cdot 10^{-4}$			0.008	
5		$4 \cdot 10^{-5}$			0.001	

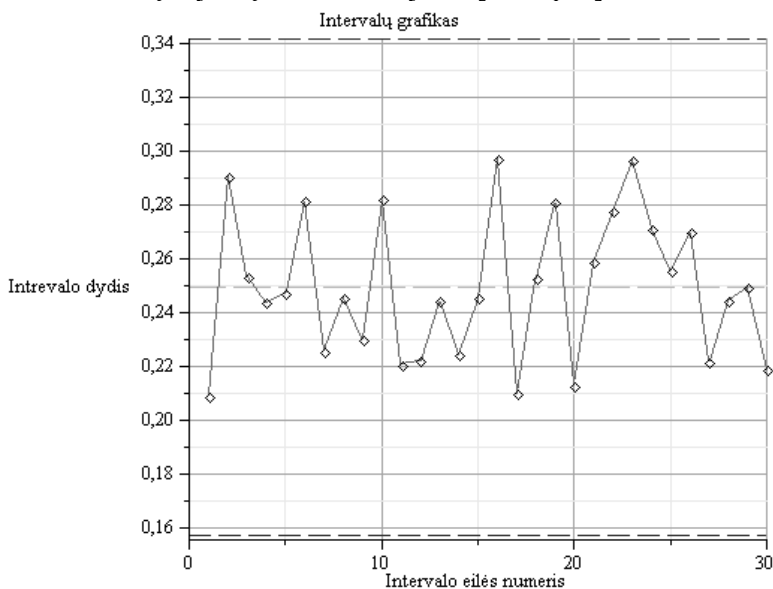
Lentelėje pateikti duomenys rodo, kad vykdant atrankinę kontrolę partija yra pripažįstama netinkama, kai tikrinamų gaminių dydis yra 20 ir 40, o juose aptinkamų defektinių gaminių skaičius viršija 2 gaminius.

### Statistinė kokybės kontrolė

Tai dar vienas statistikos instrumentas, kuris yra dažnai naudojamas gamyboje. Teisingas kokybės kontrolės pasirinkimas, gali padidinti gamybos apimtį, pagerinti produkcijos kokybę ir sumažinti gamybos sąnaudas. Čia pateikiama tik keletas metodų, kurių pagalba galima nustatyti, kurie gamybos proceso periodai yra už kokybės ribų ir reikalauja tolimesnio žmogaus įsikišimo į procesą.[5,462psl.]

Statistinės kokybės kontrolės grafikų sudarymas yra aprašytas 6 šaltinyje, nors jis yra skirtas transporto priemonių kokybės tyrimui, bet statistinės kokybės kontrolės metodo esmė nesikeičia.

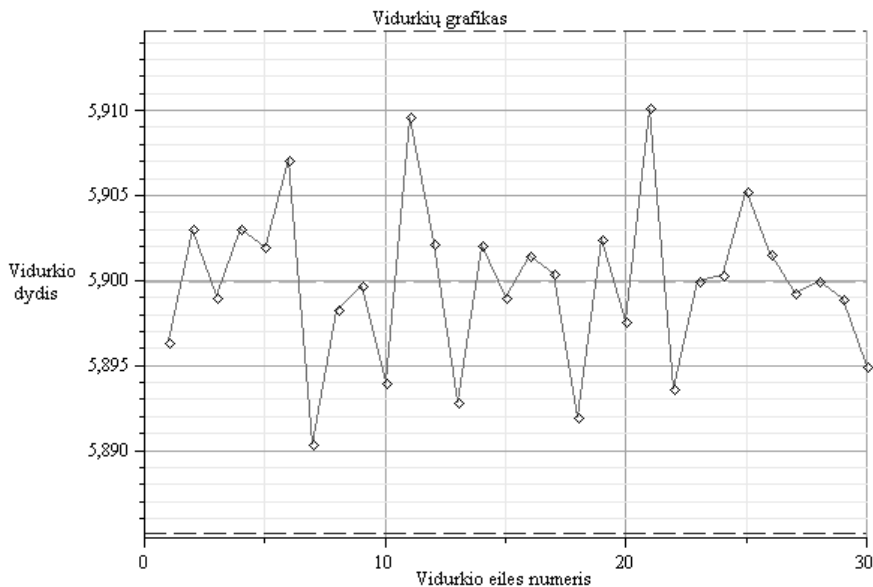
Vienas intervalų grafikas yra retai naudojamas. Jis dažniausiai yra analizuojamas kartu su vidurkių grafiku. Intervalų grafikas naudojamas kai tiriamų objektų imtis neviršija 12.[4,225psl.]



3pav. „Intervalų grafikas“

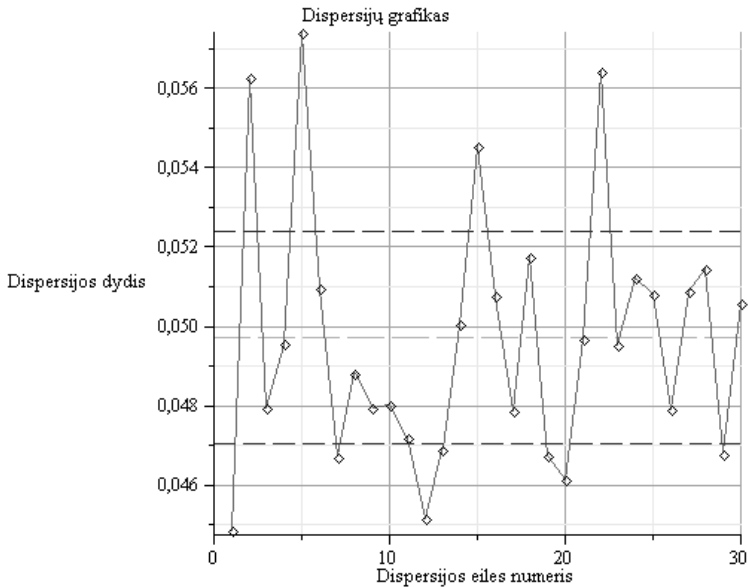


Vidurkių grafikas yra plačiausiai paplitęs dėl kiekybinių požymių. Dažniausiai( tačiau ne visada), jis yra naudojamas kartu su dispersijų arba intervalų grafikais.[4, 221psl.]



**4pav.** „Vidurkių grafikas“

Dispersijų grafikas yra naudojamas, kai tiriamų partijos objektų skaičius viršija 12. Taip yra todėl, kad intervalų grafikas didelėms imtims turi mažesnę efektyvumą.[4, 223psl.]



**5pav.** „Dispersijų grafikas“

Išnagrinėjus 3,4 ir 5 paveikslus nėra aišku iš intervalų ir vidurkių grafikų, kurios iš tiriamų partijų yra už kontroliuojamų ribų ir ar galima būtų teigti, jog šie procesai yra be nuokrypių. Kitaip yra žiūrint į dispersijų grafiką: čia pastebimi ribų kirtimai. Taigi galima būtų teigti, jog 2,5 ir 15 partijose gali būti ir daugiau neatitinkančių kokybės reikalavimų gaminių, negu likusiose. Šiuo atveju intervalų ir vidurkių grafikas gal tiksliai ir neparodo šių nuokrypių, tačiau būtų galima griežtinti jų ribas ir tada turėtų išryškėti atitinkami nuokrypiai.

### **Išvada**

Naudojantis statistinės kontrolės metodais buvo ištirti mechatronikos gaminių kokybė ir patikimumas.

Metodų pagalbą galima nesunkiai aptikti įvairių technologinio proceso nuokrypių, taip pat pasirinkti tokius kokybės kontrolės reikalavimus, jog šie tenkintų kokybės reikalavimus.

## **Literatūra**

1. Navidi, William.: Statistics for engineers and scientists /William Navidi New York (N.Y.) : McGraw-Hill Higher Education, 2006.. xviii, 869 p. : iliustr. + 1 kompakt. diskas..
2. Гнеденко Б.В., Беляев Ю.К, Соловьев А.Д. Математические методы в теории надежности. Москва.1965.
3. J. Kruopis, A. Vaišvyla, R. Kalnius Mechatronikos gaminių kokybė. VU Leidykla, Vilnius, 2005
4. Коуден Д. Статистические методы контроля качества. Москва: Физматлит. 1961
5. Berk, Kenneth N., . Анализ данных с помощью Microsoft Excel : адаптировано для Office XP / Кеннет Н. Берк, Патрик Лэйри. - Москва : Издательский дом "Вильямс", 2005. - 555 p.
6. V. Valavičius, M. Karkauskas. Statistical quality assurance of vehicles. ISSN 1392-2114. Ultragarasas. Nr. 3 (64). Kaunas: Technologija, 2009. P. 38-41.

## **SAMPLING CONTROL AND STATISTICAL QUALITY CONTROL OF MECHATRONIC PRODUCTS**

### **M.Karkauskas**

#### **Summary**

This job is showing how to use sampling control and statistical quality control in the real production. We find out failure and probability of production process to sure our selected quality limit.

## MATEMATINIS MODELIAVIMAS

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“,  
įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

# LYGIAGRETUS PREDIKTORIAUS- KOREKTORIAUS ALGORITMAS PARABOLINĖMS LYGTIMS GRAFUOSE SPRĘSTI

**Natalija Tumanova**

*VGTU, tumnat@gmail.com*

**Anotacija.** Nagrinėjama parabolinė reakcijos-difuzijos lygtis vienmačio grafo briaunose. Konstruojant algoritmą naudojama diskrečioji schema, leidžianti suskaidyti grafa į atskiras briaunas. Pateikiamas lygiagretus algoritmas, grindžiamas duomenų lygiagretumo paradigma. Duomenims tarp procesų paskirstyti naudojamas Metis įrankis. Nagrinėjamos lygiagrečiojo algoritmo savybės, pateikiami skaičiavimo eksperimento rezultatai.

### Įvadas

Daugelis taikomųjų uždavinių gali būti modeliuojami kaip reakcijos-difuzijos lygtys grafuose: neuronų sužadavimo, transportavimo siaurais vamzdžiais, elektros grandinės modeliavimo ir kiti uždaviniai. Realūs uždaviniai paprastai pasižymi didele skaičiavimų apimtimi, apima dažnai netiesines lygčių sistemas, todėl jų sprendimas yra gana sudėtingas.

Skaitiniai algoritmai šiems modeliams spręsti nagrinėjami darbuose (Chegis 1992; Mascagni, Sherman 1989; Mascagni 1990; Mascagni 1991; Rempe, Chopp 2006; Ying, Rose, Henriquez 2008). Naudojant tikslesnes ir stabilesnes pilnai netiesines skirtumines schemas kiekviename laiko žingsnyje reikia spręsti didelę tiesinių lygčių sistemą. Tačiau prediktoriaus-korektorius metodas bei kiti srities skaidymo metodai leidžia suskaidyti grafa į atskiras briaunas ir

spřesti uždavinį kiekvienoje briaunoje atskirai (Rempe, Chopp 2006). Tai sudaro geras sąlygas lygiagretinti skaičiavimus. Šiame straipsnyje nagrinėjamas lygiagretus algoritmas tiesinei parabolinei lygčiai grafuose spřesti, grindžiamas prediktoriaus-korektorius skir-tumine schema.

## Matematinis modelis

Uždavinio formulavimo sritis yra grafas. Grafo viršūnių aibė  $P = \{p_i, i = 1, \dots, I\}$ , briaunų aibė  $E = \{e_k = (p_{k_s}, p_{k_f}), p_{k_s}, p_{k_f} \in P, k = 1, \dots, K\}$ . Kiekvienoje grafo viršūnėje apibrėžiamos įeinančių ir išeinančių briaunų aibės:  $N^+(p_i) = \{e_k : e_k = (p_{i_s}, p_i) \in E, s = 1, \dots, S_i\}$ ,  $N^-(p_i) = \{e_k : e_k = (p_i, p_{i_f}) \in E, f = 1, \dots, F_i\}$ . Taip pat apibrėžiamos galinės viršūnės:  $T = \{p_i : p_i \in P, N^+(p_i) = \emptyset \text{ arba } N^-(p_i) = \emptyset\}$ .

Kiekvienoje grafo briaunoje sprendžiama parabolinė lygtis:

$$\frac{\partial u^k}{\partial t} = \frac{\partial}{\partial x} \left( d^k \frac{\partial u^k}{\partial x} \right) - q^k u^k + f^k, \quad 0 < x < l_k, \quad k = 1, \dots, K, \quad (1)$$

$$0 < d_0 \leq d^k(x, t) \leq d_M, \quad q^k(x, t) \geq 0, \quad b_i = \begin{cases} 0, & \text{jei } e_m = (p_j, p_s), \\ l_m, & \text{jei } e_m = (p_s, p_j). \end{cases}$$

$$u^m(p_i, t) := u^m(b_i, t) = \mu_i(t), \quad \forall p_i \in T,$$

Grafo viršūnių aibę  $P = T \cup P_1$  sudaro galinės ir vidinės viršūnės (pavadinkim jas mazgais). Mazguose galioja tvermės dėsnis:

$$c \frac{\partial u_s}{\partial t} + q_s u_s = \sum_{e_m \in N^-(p_i)} d^m \frac{\partial u^m}{\partial x} \Big|_{x=0} - \sum_{e_k \in N^+(p_i)} d^k \frac{\partial u^k}{\partial x} \Big|_{x=l_k},$$

$$u_s(t) = u^m(p_s, t), \quad \forall p_s \in P_1, \quad e_m \in N^\pm(p_s). \quad (2)$$

## Diskrečioji schema

Uždavinio sritis diskretizuojama erdvėje ir laike. Diskretusis tinklas:  $\varpi_h(k) = \{x_j^k : x_j^k = jh_k, j = 0, \dots, J_k\}$ ,  $x_{J_k}^k = l_k$ ,  $k = 1, \dots, K$ ,  $\varpi_\tau = \{t^n : t^n = n\tau, n = 0, \dots, N, N\tau = T\}$ , jame apibrėžiame funkciją  $U_j^{k,n} = U^k(x_j, t^n)$ ,  $k = 1, \dots, K$ . Naudosime tokius baigtinių skirtumų žymėjimus:

$$U_{\bar{i}}^n = \frac{U^n - U^{n-1}}{\tau}, \partial_x U_j^{k,n} = \frac{U_{j+1}^{k,n} - U_j^{k,n}}{h_k}, \partial_{\bar{x}} U_j^{k,n} = \frac{U_j^{k,n} - U_{j-1}^{k,n}}{h_k}.$$

Visiškai neišreikštinė schema parabolinei lygčiai (1) briaunose spęsti yra tokia:

$$U_{\bar{i}}^{k,n} = \partial_{\bar{x}} (d_{j+\frac{1}{2}}^{k,n} \partial_x U_j^{k,n}) - q^k U_j^{k,n} + f_j^{k,n}, x_j \in \omega_h(k), k = 1, \dots, K, (3)$$

$$U^{m,n}(p_i) = \mu_i(t^n), \quad \forall p_i \in T.$$

Tvermės lygtis grafo mazguose aproksimuojama tokiu būdu:

$$\begin{aligned} c U_{s,\bar{i}}^n + q_s^n U_s^n + \sum_{e_k \in N^+(p_s)} \left[ d_{J_k-\frac{1}{2}}^{k,n} \partial_{\bar{x}} U_{J_k}^{k,n} + \frac{h_k}{2} (U_{J_k,\bar{i}}^{k,n} + q_{J_k}^{k,n} U_{J_k}^{k,n} - f_{J_k}^{k,n}) \right] \\ - \sum_{e_m \in N^-(p_s)} \left[ d_{\frac{1}{2}}^{m,n} \partial_{\bar{x}} U_0^{m,n} - \frac{h_m}{2} (U_{0,\bar{i}}^{m,n} + q_0^{m,n} U_0^{m,n} - f_0^{m,n}) \right] = 0, \\ \forall p_s \in P_1, \end{aligned} (4)$$

$$U_s^n = U^{m,n}(p_s), \quad \forall p_s \in P_1, \quad e_m \in N^\pm(p_s). (5)$$

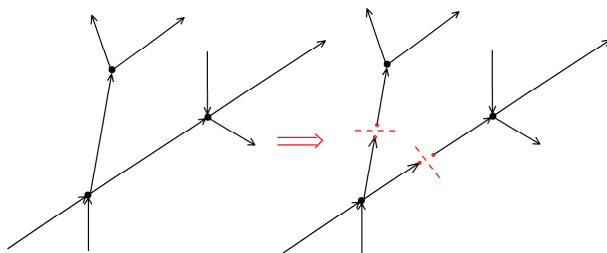
Schemos aproksimacijos tikslumas  $O(\tau + h^2)$  visose taškuose.

Tokiu būdu kiekvienai grafo briaunai gaunamos  $J-1$  lygtys (3). Papildomos 2 lygtys gaunamos iš kraštinių sąlygų, tolydumo sąlygų (5) ir tvermės dėsnio mazguose aproksimacijos (4). Gauta tiesinių lygčių sistema sieja visus vieno laiko sluoksnio diskrečiojo tinklo taškus, todėl jos sprendimas realiam uždaviniui reikalauja didelių kaštų. Prediktoriaus-korektorius schema leidžia apskaičiuoti

sprendinį grafo mazguose, o po to skaičiuoti kiekvienoje briaunoje atskirai, naudojant mazgų reikšmes kaip kraštines sąlygas. Kiekvienoje briaunoje gaunama trijstrižainė tiesinių lygčių sistema, kuri efektyviai sprendžiama perkelties metodu. Detalai nuoseklus algoritmas yra aprašytas (Rempe, Chopp 2006).

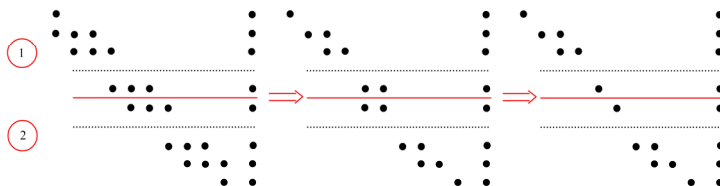
### Lygiagretus prediktoriaus – korektoriaus algoritmas

Lygiagretus prediktoriaus – korektoriaus algoritmas yra grindžiamas duomenų lygiagretumo paradigma. Duomenims tarp procesų paskirstyti buvo naudojamas Metis įrankis (Karypis Lab). Metis paskirsto procesams mazgus. Natūralu, visos briaunos, jungiančios vieno proceso mazgus, irgi priklauso tam procesui. Briaunos, kurių pradžios ir pabaigos mazgai priklauso skirtingiems procesams, yra dalijamos pusiau, prie pjūvio atsiranda specifiniai kraštiniai taškai:



1 pav. Grafo padalijimas procesams

Tokiu būdu skaičiavimo eiga kiekvienam procesui skiriasi tik skaičiuojant bendras (su kitu procesu) briaunas. Jose irgi naudojamas faktorizacijos metodas, kuris schematiškai pavaizduotas 2 pav.



2 pav. Bendros briaunos faktorizacija

Po antro žingsnio, atlikus pusę faktorizacijos procedūros, procesai keičiasi koeficientais prie kraštinių taškų, išsprendžia dviejų lygčių sistemą, po to baigia spręsti lygčių sistemą briaunoje.

Trumpai lygiagretų prediktoriaus-korektorius algoritmą galima aprašyti taip ( $K_{\text{shared}}$  yra skaičius briaunų, kuriomis procesas dalijasi su kitais, o briaunų ir viršūnių skaičiai yra lokalūs):

```

while (t<T){
  for (k=0; k<K; k++){
    Apskaičiuojami briaunos trijstrižainės sistemos koeficientai
    if ( $p_{k_s} \in P_1$ ) Apskaičiuojamos mazgo koeficientų sumos
    if ( $p_{k_f} \in P_1$ ) Apskaičiuojamos mazgo koeficientų sumos
  }
  for (i=0; i<I1; i++) Apskaičiuojamas sprendinys mazge
  for (k=0; k<Kshared; k++){
    Faktorizuojama šaka
    Siunčiami koeficientai kitam procesui
  }
  for (k=Kshared; k<K; k++) Sprendžiama trijstrižainė sistema
  for (k=0; k<Kshared; k++){
    Gaunami koeficientai iš kito proceso
    Apskaičiuojamas sprendinys kraštiniame taške
    Baigiama spręsti trijstrižainė sistema
  }
  for (i=0; i<I1; i++) Patikslinamas sprendinys mazge
}

```

Bendras uždavinio dydis  $W = O(\sum_{k=1}^K J_k)$ , o lygiagrečiojo algoritmo vykdymo laiką galima įvertinti taip:

$$T_p = \frac{W}{p} \theta + \gamma(p)(\alpha + \beta L), \quad (6)$$

kur  $p$  yra procesorių skaičius,  $\theta$  - padalinto grafo subalansavimo koeficientas (rodo kaip skiriasi uždaviniai, atitekę skirtingiems procesams).



sams),  $(\alpha + \beta L) - L$  dydžio pranešimo persiuntimo tarp dviejų procesorių kaštai, o  $\gamma(p)$  parodo kiek daugiausia pranešimų siunčia vienas procesas:  $\gamma(p) = \max_p(K_{share})$ . Reikėtų pabrėžti, kad įvertis

(6) yra pesimistiškas, nes numato tą atvejį, kai procesui, kuriam atiteko didžiausias uždavinys, taip pat tenka siusti daugiausia pranešimų.

Kompiuterių klasteriuose pranešimo parengimo laikas yra žymiai didesnis už vieno skaičiaus siuntimo laiką ( $\alpha \gg \beta$ ), o siunčiami pranešimai yra labai trumpi, todėl (6) galima supaprastinti taip:

$$T_p \approx \frac{W}{p} \theta + \alpha \cdot \gamma(p), \quad (7)$$

Programoje naudojamas asinchroninis duomenų siuntimas, o tai reiškia, kad procesui nereikia laukti, kol pranešimas pasieks adresatą ir tas pranešimą apdoros. Daugelis šiuolaikinių kompiuterių gali tuo pačiu metu atlikti skaičiavimus ir siusti pranešimus, todėl, kai skaičiavimų apimtis su lokaliais duomenimis tarp duomenų siuntimo ir gavimo (šiuo atveju tik „savų“ briaunų skaičiavimas) yra pakankamai didelė, duomenų persiuntimo kaštai mažėja, nes netenka laukti tuščiai duomenų iš kito proceso. Šiuo atveju (7) formulės antras dėmuo dar mažėtų.

Naudojant (7) įvertį gaunamos tokios lygiagrečiojo algoritmo papildomos sąnaudos:

$$K_p(W, p) = W(\theta - 1) + \alpha \cdot p \cdot \gamma(p),$$

o tada lygiagrečiojo algoritmo izoefektyvumo funkcija yra

$$W = \Theta(\gamma(p) \cdot p). \quad (8)$$

## Skaičiavimo eksperimentas

Skaičiavimai buvo atlikti VGTU klasteryje Vilkas su Intel® Core™ i7-860 @ 2.80 GHz procesoriais, turinčiais po 4 branduolius. Testinio uždavinio duomenys:

- Modelio koeficientai  $d^k(x,t) = 1 + x$ ,  $q^k(x,t) = e^{t+1}$ , o funkcija  $f^k(x,t)$  parenkama taip, kad tikslus sprendinys  $u^k(x,t) = x^2(l_k - x)^2 t$ .
- Briaunų ilgiai generuojami atsitiktinai nuo 0.1 iki 5.

Skaičiavimai buvo atlikti su trimis grafais. Du iš jų yra gana išretinti, briaunų yra tik apie du kartus daugiau nei mazgų, o antras grafas yra du kartus didesnis už pirmąjį. Abiejų grafų padalijimo balansavimo koeficientai yra panašūs. Trečias grafas yra labiau prisotintas, jo mazgų skaičius yra  $I_1 \approx \sqrt[3]{K}$ , o balansavimo koeficientai nuo 1.03 iki 1.3 kartų blogesni. Skaičiavimų rezultatai pateikti lentelėse.

Algoritmo efektyvumas, skaičiuojant 1275 briaunų ir 510 mazgų;  $T_0=754$

$p(mxb)$	1x2	2x1	1x4	2x2	4x1	2x4	4x2	8x1	4x4	8x2	8x4
$T_p$	389	383	224	206	201	116	106	102	61	58	34
$S_p$	1.94	1.97	3.37	3.66	3.75	6.50	7.11	7.39	12.36	13.00	22.18
$E_p$	0.97	0.98	0.84	0.92	0.94	0.81	0.89	0.92	0.77	0.81	0.69

Algoritmo efektyvumas, skaičiuojant 2555 briaunų ir 1022 mazgų;  $T_0=1535$

$p(mxb)$	1x2	2x1	1x4	2x2	4x1	2x4	4x2	8x1	4x4	8x2	8x4
$T_p$	794	768	455	407	389	233	210	196	120	110	66
$S_p$	1.93	2.00	3.37	3.77	3.95	6.59	7.31	7.83	12.79	13.95	23.26
$E_p$	0.97	1.00	0.84	0.94	0.97	0.82	0.91	0.98	0.80	0.87	0.73

Algoritmo efektyvumas, skaičiuojant 1275 briaunų ir 110 mazgų;  $T_0=751$

$p(mxb)$	1x2	2x1	1x4	2x2	4x1	2x4	4x2	8x1	4x4	8x2	8x4
$T_p$	393	394	226	214	210	122	110	108	67	61	37
$S_p$	1.91	1.91	3.32	3.51	3.58	6.16	6.83	6.95	11.21	12.31	20.30
$E_p$	0.96	0.95	0.83	0.88	0.89	0.77	0.85	0.87	0.70	0.77	0.63

Skaičiavimo eksperimento metu  $\gamma(p)$  įgijo skirtingas reikšmes, tačiau jos buvo nedidelės ir nebedidėjo, kai  $p > 8$ :  $\gamma(p) \leq \gamma_{\max} \ll K$ . Todėl galime laikyti, kad šiais atvejais izoefektyvumo funkcija (8) yra tiesinė. Skaičiavimų rezultatai tai patvirtina: lyginant pirmos ir antros lentelių duomenis matome, kad apie du

kartus didesniai uždaviniui spręsti panašiu efektyvumu galime panaudoti du kartus daugiau procesorių.

Taip pat matome ryškiają tendenciją rezultatams blogėti, jeigu yra naudojamas daugiau negu vienas branduolys skaičiavimo mazge. Tai galima paaiškinti tuo, kad procesai, vykstantys skirtinguose vieno skaičiavimo mazgo branduoliuose, konkuruoja tarpusavyje dėl duomenų skaitymo/rašymo iš/į atmintį (Levy, Gal-On 2008).

Blogesnius lygiagrečiojo algoritmo rodiklius trečiajame skaičiavime galima paaiškinti blogesniu grafo balansavimu.

Tačiau sunku buvo suderinti skaičiavimų rezultatus su (7) įverčiu. Taip galėjo atsitikti dėl to, kad (6) ir (7) įverčiai yra pesimistiški, rodo maksimalų algoritmo vykdymo laiką. Taip pat heterogeniniuose kompiuteriuose daug priklauso nuo to, su kuriuo procesoriumi buvo paskaičiuotas dydis  $W = T_0$  ir kaip paskirstytos užduotys tarp nevienodų procesorių. Todėl kartais, kai  $T_0$  apskaičiuotas su lėtesniu procesoriumi, ir lėtesniam procesoriui, matyt, atitenka mažesnė užduotis lygiagrečiuose skaičiavimuose, skaičiavimo rezultatai rodo, kad

$$T_p < \frac{W}{p} \theta.$$

## Apibendrinimas

1. Prediktoriaus-korektorius schema leidžia suskaidyti uždavinį grafe į vienmačius uždavinius kiekvienoje grafo briaunoje ir lengvai lygiagretinti skaičiavimus, kas yra svarbu sprendžiant didelės apimties realius uždavinius.

2. Pasiūlytas lygiagretus prediktoriaus-korektorius algoritmas. Papildomos algoritmo sąnaudos labiausiai priklauso nuo grafo padalijimo kokybės.

3. Skaičiavimo eksperimento rezultatai rodo algoritmo efektyvumą. Rezultatai sprendžiant apie du kartus didesnę uždavinį patvirtina, kad izoefektyvumo funkcija yra tiesinė. Skaičiavimai rodo, kad efektyviausia naudoti po vieną procesorių skaičiavimo mazge.

## Literatūra

- Čiegis, R. 2005. Lygiagretieji algoritmai ir tinklinės technologijos. Vilnius: Technika. ISBN 9986-05-910-0.
- Chegis, R. 1992. Investigation of difference schemes for a class of models of excitability, *Comput. Maths. Math. Phys.* 32(6): 757–567.
- Levy, M.; Gal-On, S. 2008. *Get the truth on multicore-processor performance* [žiūrėta 2010 balandžio 15 d.]. Prieiga per internetą: <[http://www.embeddedintel.com/special\\_features.php?article=920](http://www.embeddedintel.com/special_features.php?article=920)>.
- Mascagni, M. V. 1990. The backward Euler method for numerical simulation of the Hodgkin – Huxley equations of nerve conduction, *SIAM J. Numer. Anal.* (27): 941–962.
- Mascagni, M. V. 1991. A parallelizing algorithm for computing solutions to arbitrarily branched cable neuron models, *J. Neurosci. Methods* (36): 105–114.
- Mascagni, M. V.; Sherman, A. S. 1989. Numerical methods for neuronal modeling, *Methods in neuronal modeling: From synapses to networks*: 439–484. ISBN:0-262-11133-0.
- Rempe, M. J.; Chopp, D. L. 2006. A predictor-corrector algorithm for reaction-diffusion equations associated with neural activity on branched structures, *SIAM J. Scientific Computing* (28)6: 2139–2161.
- Ying, W.; Rose D. J.; Henriquez C. S. 2008. Efficient fully implicit time integration methods for modeling cardiac dynamics. *IEEE Trans. Biomed. Eng.*(55)12: 2701-2711.

## PARALLEL PREDICTOR CORRECTOR ALGORITHM FOR PARABOLIC EQUATION ON GRAPHS

**N. Tumanova**

### Summary

We consider reaction-diffusion equation formulated on the edges of the graph. We have proposed the parallel predictor corrector algorithm based on data distribution paradigm. Algorithm is implemented to decouple computations at each edge of the graph. We have discussed the properties of the algorithm and have presented the results of computational experiments.

## SKAITINIS KONVEKCIJOS – DIFUZIJOS LYGČIŲ SPRENDIMAS

**Povilas Germanavičius**

*Vilniaus Gedimino technikos universitetas*

### **Anotacija**

Šiame straipsnyje bus supažindinama su vienmačiu ir dvimačiu difuzijos – konvekcijos tipo uždaviniu ir jo analize. Vienmačiu atveju difuzijos lygtis sprendžiama Furjė ir neišreikštiniu Eulerio metodais. Dvimačiu atveju lygtis sprendžiama išreikštiniu, neišreikštiniu ir simetriniu Eulerio metodais. Pateikti skaičiavimo eksperimentų rezultatai lyginami su teoriniais konvergavimo įverčiais.

### **Įvadas**

Oro taršos problema egzistuoja jau nuo XIX a. prasidėjusios pramoninės revoliucijos. Oro tarša - medžiagų, kenksmingų žmoniems bei kitiems organizmams arba pažeidžiančių aplinką, išmetimas į atmosferą. Norint išsiaiškinti apie taršos sklidimą aplinkoje, kuriami taršos sklidimo matematiniai modeliai. Šių medžiagų pasiskirstymas ore priklauso nuo trijų procesų. Pirma, kenksmingos medžiagos atsiranda kaip gamybinės, technologinės ir kitokios žmonių veiklos šalutinis produktas. Antra, kenksmingas medžiagas perneša vėjas – konvekcinis procesas. Trečia, šios medžiagos sklinda į aplinką dėl difuzijos proceso.

## Matematinis modelis vienmačiu atveju

Pažymėkime  $u(x,t)$  kenksmingų medžiagų koncentraciją nuo laiko  $t$ , taške  $x$ . Tarkime kenksmingos medžiagos sklinda į aplinką tik difuzijos proceso. Šį uždavinį spręsimė Furjė metodu ir skaitiniu metodu – neišreikštiniu Eulerio metodu. Nagrinėsime tokį matematinį modelį:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right), \quad 0 < x < 1, \quad 0 < t < 1. \quad (1)$$

Kraštinės sąlygos:

$$u(0,t) = 0, \quad u(1,t) = 0, \quad t > 0. \quad (2)$$

Pradinės sąlyga:

$$u(x,0) = \phi(x) = -2x^2 + 2x, \quad 0 \leq x \leq 1 \quad (3)$$

## Furjė metodas

Šį uždavinį spręsimė kintamųjų atskiriamuoju arba kitaip Furjė metodu. Ieškosime tokio pavidalo nenolinių šilumos laidumo lygties sprendinių:

$$u(x,t) = \sum_{n=1}^{\infty} T(t) X(x)$$

Tada  $\frac{\partial u}{\partial t} = T'(t) X(x)$ ,  $\frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right) = k T(t) X''(x)$  ir įrašę šiuos reiškinius į lygtį (1) gauname:

$$\frac{T'(t)}{T(t)} = k \frac{X''(t)}{X(t)} \quad (4)$$

Kairėje (4) lygybės pusėje esantis reiškinys yra vien tik kintamojo  $t$  funkcija, o dešinėje - tik  $x$  funkcija, todėl abi lygybės turi būti tapatingai lygios konstantai (sąlyga, kad  $const < 0$  priešingas atvejis neturi fizikinės prasmės):

$$\frac{T'(t)}{T(t)} = k \frac{X''(t)}{X(t)} = -\lambda^2$$

Gauname  $T_n(t)$  ir  $X_n(t)$  bendruosius sprendinius:

$$T_n(t) = C_1 e^{-\lambda^2 kt}, \quad X_n(t) = C_1 \sin(\lambda x) + C_2 \cos(\lambda x), \quad \lambda = \pi n.$$

Iš kraštinių sąlygų (2) gauname:

$$T_n(t) = b_n e^{-\pi^2 n^2 kt}, \quad X_n(t) = b_n \sin(\pi nx)$$

Furjė metodu sprendinys  $u(x,t)$ , kur  $b_n$  sandaugą pažymime  $a_n$ :

$$u(x,t) = \sum_{n=1}^{\infty} a_n e^{-\pi^2 n^2 kt} \sin(\pi nx)$$

Koeficientus  $a_n$  apskaičiuojame skleidami Furjė eilutę uždavinio pradinę sąlygą (3):

$$\sum_{n=1}^{\infty} a_n \sin(\pi nx) = \phi(x) \tag{5}$$

Iš (5) lygybės išreiškiame  $a_n$ :

$$a_n = 2 \left( \int_0^1 \phi(x) \sin(\pi nx) dx \right)$$

Gauname Furjė metodu lygties (1) sprendinį  $u(x,t)$ :

$$u(x, t) = \sum_{n=1}^{\infty} 2 \left( \int_0^1 \phi(x) \sin(\pi n x) dx \right) e^{-\pi^2 n^2 k t} \sin(\pi n x) \quad (6)$$

### Neišreikštinis Eulerio metodas vienmačiu atveju

Lygties dešiniąją pusę  $\frac{\partial u}{\partial t}$  ir kairiąją pusę  $\frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right)$  pakeisime baigtiniais skirtumais:

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\tau} \quad (7a)$$

$$\frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right) \approx k \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2} \quad (7b)$$

$\tau$  – žingsnis pagal laiką  $t$ ,

$h$  – žingsnis pagal  $x$ .

Taikydami neišreikštinį Eulerio metodą [1] diferencialinę lygtį (1) aproksimuojame diskrečia lygtimi:

$$\frac{u_i^{n+1} - u_i^n}{\tau} = k \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2} \quad (8)$$

### Konvergavimo analizė

Panagrinęsime ar neišreikštinio Eulerio metodo (8) gauti sprendiniai konverguoja į diferencialinio uždavinio sprendinį (6). Skaitiškai sprenddami tokius uždavinius, turime įsitikinti, kad diskretusis sprendinys artėja prie diferencialinio uždavinio sprendinio, priešingu atveju laikysime, kad pasirinktas skaitinis metodas skaičiuoja blogai.



Neišreikštino Eulerio metodo aproksimacijos paklaida įvertinama tokia nelygybe:

$$|\varepsilon| \leq O(\tau + h^2)$$

Ši lygybė nusako tai kad jeigu norime gauti keturis kartus tikslesni skaitini sprendinį, laiko žingsnį  $\tau$  pagal laiką  $t$  turime padidinti keturis kartus, žingsnį  $h$  pagal  $x$  turime padidinti du kartus.

Imkime  $h = 0,1$ ,  $\tau = 0,005$  ir nagrinėkime sprendinius laiko momentu  $t = 0,5$ . 1 lentelėje yra pateiktos diferencialinio kraštinio uždavinio (1) sprendinio  $u$  ir baigtinių skirtumų schemos (8) sprendinio  $y$  reikšmės, taip pat diskrečiojo sprendinio globaliosios paklaidos  $z(x,t) = |y(x,t) - u(x,t)|$  reikšmės.

**1 lentelė.** Paklaidos tarp analizinio ir skaitinio sprendinio.

x	u(x,t)	y(x,t)	z(x,t)
0	0,0000	0,0000	0,0000
0,1	0,0135	0,0142	0,0007
0,2	0,0257	0,0270	0,0013
0,3	0,0354	0,0372	0,0018
0,4	0,0416	0,0437	0,0021
0,5	0,0438	0,0460	0,0022
0,6	0,0416	0,0437	0,0021
0,7	0,0354	0,0372	0,0018
0,8	0,0257	0,0270	0,0013
0,9	0,0135	0,0142	0,0007
1	0,0000	0,0000	0,0000

Neišreikštinio Eulerio metodo sprendinio  $y(x,t)$  konvergavimo greitį įvertiname sprenddami uždavinį (8) su vis mažesniais žingsniais pagal erdvinę ir laiko koordinates. 2 lentelėje pateiktos globalios paklaidos reikšmės.

$$E(h, \tau) = \max_{0 \leq i \leq N} |y(x_i, t) - u(x_i, t)|$$

**2 lentelė.** Baigtinių skirtumų schemos maksimalios globalios paklaidos

$\tau$	$h$	$E(h, \tau)$
0,005000	0,1000	0,0022276
0,001250	0,0500	0,0005555
0,000313	0,0250	0,0001388
0,000078	0,0125	0,0000347

Iš pateiktų rezultatų matome, kad gautas skaitinis konvergavimo greitis atitinka teorinį konvergavimo greitį, t.y., laiko žingsnį  $\tau$  pagal laiką  $t$  padidinę keturis kartus, o žingsnį  $h$  pagal  $x$  padidinę du kartus gavome keturis kartus tikslesnį sprendinį.

### Matematinis modelis dvimačiu atveju

Dvimačiu atveju taipogi kaip ir vienmačiu atveju laikysime, kad kenksmingos medžiagos sklinda į aplinką tik difuzijos proceso. Šį uždavinį spęsimė trimis skaitiniais metodais:

- Išreikštinu Eulerio metodu;
- Neišreikštinu Eulerio metodu;
- Simetriniu Eulerio metodu;

Nagrinėsime tokį matematinį modelį:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial X} \left( k \frac{\partial u}{\partial X} \right), \quad X = (x, y), \quad 0 < x, y < 1, \quad 0 < t < 0,1. \quad (9)$$

Kraštinės sąlygos:

$$u(0, y, t) = 0, \quad u(1, y, t) = 0, \quad u(x, 0, t) = 0, \quad u(x, 1, t) = 0. \quad (10)$$

Pradinė sąlyga:

$$u(x, y, 0) = \sin(\pi x) \cdot \sin(\pi y) \quad (11)$$

Lygties (9) su nurodytomis kraštinėmis (10) ir pradine sąlyga (11) tikslus sprendinys:

$$u(x, y, t) = \sin(\pi x) \cdot \sin(\pi y) \cdot e^{-2\pi^2 t}$$

### Konvergavimo analizė

Ištirsime ar taikant *išreikštinį*, *neišreikštinį*, ar *simetrinį* Eulerio metodus, skaitinis sprendinys artėja į tikslų sprendinį, tokiu tikslumu kuris nusakomas pagal tokias aproksimacijos paklaidos nelygybes:

$$\text{Išreikštiniu Eulerio metodu } |\varepsilon| \leq O(\tau + h^2);$$

$$\text{Neišreikštiniu Eulerio metodu } |\varepsilon| \leq O(\tau + h^2);$$

$$\text{Simetriniu Eulerio metodu } |\varepsilon| \leq O(\tau^2 + h^2);$$

Taikant *išreikštinį Eulerio metodą* stabilumo reikalavimas :

$$\text{Vienmačiu atveju: } \tau \leq \frac{h^2}{4}$$

$$\text{Dvimačiu atveju: } \tau \leq \frac{h_1^2 + h_2^2}{4}$$

Naudodami simbolinių skaičiavimų paketą *Matlab*, kur realizuoti visi trys paminėti baigtinių skirtumų metodai randame uždavinio (9) skaitinius sprendinius. 3 lentelėje pateikiame kiekviename tinklo taške paklaidas, kai taikome išreikštinį Eulerio metodą, kur nagrinėjamus dydžius pažymėkime taip:  $u(x, y, t)$  – tikslus sprendinys,  $y(x, y, t)$  – skaitiškai gautas sprendinys,  $z(x, y, t) = |y(x, y, t) - u(x, y, t)|$  - paklaida. Atsižvelgdami į šio metodo stabilumo reikalavimą pasirenkame tinklo dydį  $N=10$ , žingsnių skaičių pagal laiką 5. Sprendinio paklaidas ieškosime laiko momentu  $t=0.025$ . 4 lentelėje pateikiamos maksimalios globalios paklaidos.

**3 lentelė.** Skaitinio ir analizinio sprendinio paklaida kiekviename tinklo taške.

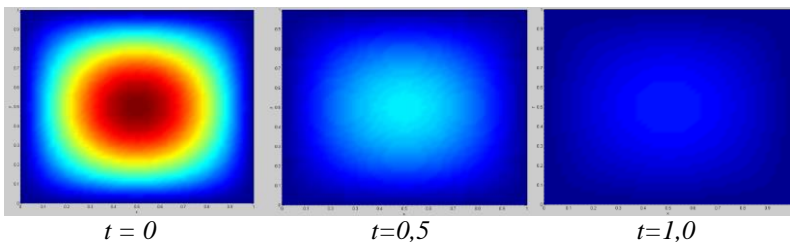
$z(x,y,t) =  y(x,y,t) - u(x,y,t) $										
y <sub>10</sub>	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
y <sub>9</sub>	0,0000	0,0015	0,0027	0,0037	0,0042	0,0042	0,0037	0,0027	0,0015	0,0000
y <sub>8</sub>	0,0000	0,0027	0,0051	0,0069	0,0079	0,0079	0,0069	0,0051	0,0027	0,0000
y <sub>7</sub>	0,0000	0,0037	0,0069	0,0093	0,0106	0,0106	0,0093	0,0069	0,0037	0,0000
y <sub>6</sub>	0,0000	0,0042	0,0079	0,0106	0,0120	0,0120	0,0106	0,0079	0,0042	0,0000
y <sub>5</sub>	0,0000	0,0042	0,0079	0,0106	0,0120	0,0120	0,0106	0,0079	0,0042	0,0000
y <sub>4</sub>	0,0000	0,0037	0,0069	0,0093	0,0106	0,0106	0,0093	0,0069	0,0037	0,0000
y <sub>3</sub>	0,0000	0,0027	0,0051	0,0069	0,0079	0,0079	0,0069	0,0051	0,0027	0,0000
y <sub>2</sub>	0,0000	0,0015	0,0027	0,0037	0,0042	0,0042	0,0037	0,0027	0,0015	0,0000
y <sub>1</sub>	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>	x <sub>9</sub>	x <sub>10</sub>

**4 lentelė.** Baigtinių skirtumų schemų maksimalios globalios paklaidos.

<i>baigtinių skirtumų metodas</i>						
<i>išreikštinis</i>		<i>neišreikštinis</i>		<i>simetrinis</i>		
<i>h</i>	$\tau$	$E(h,\tau)$	$\tau$	$E(h,\tau)$	$\tau$	$E(h,\tau)$
0,1	0,005	0,01204	0,005	0,01645	0,005	0,00273
0,05	0,00125	0,00305	0,00125	0,00431	0,0025	0,00062
0,025	0,00031	-*	0,00031	0,00109	0,00125	0,00015

\*- neatitinka stabilumo reikalavimo

Naudodami simbolinių skaičiavimų paketą *Matlab* vizualizuosime skaitinius sprendinius. Nagrinėsime, kai taršos sklidimas veikiamas difuzijos proceso, kur difuzijos koeficientą pasirinkime  $k = 1$ . Imkime  $N = 50$ , žingsnį  $h$  pagal  $x$  ir  $y$  koordinatę  $h = 0,02$ , žingsnį pagal laiką  $\tau = 0,005$ . Gauti rezultatai pateikti 1 paveikslėlyje.



**1 pav.** Taršos sklidimas veikiamas difuzijos proceso kintant laikui

### Dvimatis difuzijos – konvekcijos uždavinys

Panagrinėsime kaip kenksmingos medžiagos sklinda į aplinką difuzijos ir konvekcijos proceso. Nagrinėsime tokį matematinį modelį.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial X} \left( k \frac{\partial u}{\partial X} \right) - v \frac{\partial u}{\partial X}, \quad X = (x, y), \quad 0 < x, y < 1, \quad 0 < t < 0,1.$$

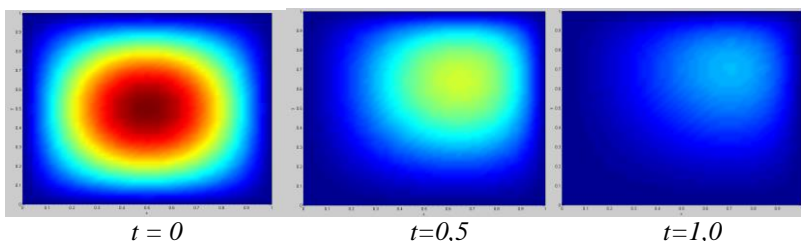
$$\frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right) - \text{difuzija, kur } k \text{ koeficientą pasirinkome } k = \frac{1}{2};$$

$$v \frac{\partial u}{\partial x} - \text{konvekcija, kur } v \text{ koeficientą pasirinkome } v = \frac{\pi}{2};$$

Kraštinės sąlygos:

$$u(0, y, t) = 0, \quad u(1, y, t) = 0, \quad u(x, 0, t) = 0, \quad u(x, 1, t) = 0.$$

Vizualizuosime skaitinius sprendinius, kai taršos sklidimas veikiamas difuzijos – konvekcijos proceso. Imkime  $N = 50$ , žingsnį  $h$  pagal  $x$  ir  $y$  koordinatę  $h = 0.02$ , žingsnį pagal laiką  $\tau = 0,005$ . Gauti rezultatai pateikti 2 paveikslėlyje.



**2 pav.** Taršos sklidimas veikiamas difuzijos – konvekcijos proceso kintant laikui

## Išvados

1. Vienamačiu atveju naudojant neišreikštinį Eulerio metodą, kur skaičiavimai atlikti naudojant du simbolinių skaičiavimų paketus Maple ir Matlab, buvo nustatyta, kad Matlab sprendžia žymiai greičiau negu Maple, todėl sprendžiant sudėtingesnę – dvimatį uždavinį naudosisime Matlab.

2. Nagrinėjant dvimatį difuzijos uždavinį atlikti skaitiniai eksperimentai taikant išreikštinį, neišreikštinį ir simetrinį Eulerio metodus, kur gauti rezultatai atitinka teorinius konvergavimo įverčius.

3. Atlikti skaičiavimo eksperimentai su difuzijos – konvekcijos lygtimi taikant skaitinį neišreikštinį Eulerio metodą.

## Literatūra

R. Čiegis. *Diferencialinių lygčių skaitiniai sprendimo metodai*. Technika, Vilnius, 2003.

*Partial Differential Equations and Fourier Methods*.

[žiūrėta 2010 balandžio 8 d.]

<<http://calclab.math.tamu.edu/~fulling/m401/401part2.pdf>>.

B.Kvedaras, M.Sapagovas. *Skaičiavimo metodai*, Vilnius, "Mintis", 1974, 516 psl.

*Maple mokymo programa* [žiūrėta 2010 balandžio 8 d.].

Prieiga per internetą:

<<http://www.maplesoft.com/support/training/index.aspx>>.

*Matlab mokymo programa* [žiūrėta 2010 balandžio 8 d.].

Prieiga per internetą: <<http://www.cyclismo.org/tutorial/matlab/>>.

## **NUMERICAL SOLUTION OF CONVECTION – DIFFUSION EQUATIONS**

**Povilas Germanavičius**

### Summary

This article there are presented one - dimensional and two - dimensional diffusion - convection type problem and its analysis. One - dimensional diffusion equation is solved using Fourier and implicit Euler methods. Two - dimensional equation is solved using an explicit, implicit and symmetric Euler methods. The calculation of experimental results are compared with theoretical convergence estimates.

## TELEGRAFINIŲ LYGČIŲ SISTEMOS ASIMPTOTINIS TYRIMAS

Rima Kriauzienė

Mykolo Romerio universitetas,  
kriauziene@gmail.com

**Santrauka.** Ieškomas telegrafinių lygčių sistemos su kintamais koeficientais asimptotinis sprendinys, sudarant suvidurkintąją sistemą. Konstruojama integralinių diferencialinių lygčių sistema, kuri aprašo bangų sąveiką. Suvidurkintąją sistemą galima pertvarkyti į paprastųjų diferencialinių lygčių sistemą. Šio darbo tikslas – parodyti, kad nagrinėjamoje sistemoje, esant tam tikroms sąlygoms, atsiranda bangų sąveika.

**Reikšminiai žodžiai:** telegrafinių lygčių sistema, vidurkinimo metodas, asimptotinis integravimas.

## Įvadas

Telegrafinių lygčių sistemos nagrinėjamos literatūroje (Fliess *et al.*; Колесов, А. Ю.; Розов, Н. Х. 2006: 63–96, Романовский, Р. К.; Мендзив, М. В. 2007: 1134–1141, Shaoyong, L. 1997: 657, Kulikov, A. N. 2008: 57–68).

Straipsnyje nagrinėjama telegrafinių lygčių sistema

$$\begin{aligned} \frac{\partial i}{\partial t} + \frac{\partial v}{\partial x} + \varepsilon \alpha(t, x) i &= 0, \\ \frac{\partial v}{\partial t} + \frac{\partial i}{\partial x} + \varepsilon \beta(t, x) v &= 0, \end{aligned} \quad (1)$$

čia  $\varepsilon$  – mažas teigiamas parametras,  $\alpha(t, x) \geq 0$ ,  $\beta(t, x) \geq 0$ ,  $i = i(t, x)$  – stiprumas,  $v = v(t, x)$  įtampa, gali būti pertvarkyta ir į vieną lygtį. Diferencijuojame pirmąją lygtį pagal  $t$ , o antrąją pagal  $x$ , atėmę iš gautos sistemos pirmos lygties antrąją ir pritaikę lygybes  $v_{tx} = v_{xt}$ ,  $i_{tx} = i_{xt}$ :

$$i_{tt} - i_{xx} + \varepsilon(\alpha_t(t, x) i + i_t \alpha(t, x) - \beta_x(t, x) v - v_x \beta(t, x)) = 0. \quad (2)$$

Remiantis (1) sistemos pirmąja lygtimi, t.y. į (2) įsirašome vietoje  $v_x = -i_t - \varepsilon \alpha(t, x) i$ , gauname

$$i_{tt} - i_{xx} + \varepsilon(\alpha_t(t, x) i + i_t \alpha(t, x) - \beta_x(t, x) v + i_t \beta(t, x)) + \varepsilon^2 \alpha(t, x) \beta(t, x) i = 0 \quad (3)$$

Šio darbo tikslas – parodyti, kad nagrinėjamoje sistemoje, esant tam tikroms sąlygoms, atsiranda bangų sąveika.

## Literatūros apžvalga

Nagrinėjama sistema buvo nagrinėta straipsnyje (Романовский, Р. К.; Мендзив, М. В. 2007: 1134–1141). Panašios telegrafinių lygčių sistemos įvairiais aspektais nagrinėjamos daugelyje straipsnių. (Fliess *et al.*; Колесов, А. Ю.; Розов, Н. Х. 2006: 63–96). (Shaoyong, L. 1997: 657) straipsnyje atlikta asimptotinė analizė pusiau tiesinės sutrukdytosios telegrafo lygties.

## Tyrimo eiga

(1) sistemą galima užrašyti Rymano (žr.: Рождественский, Б. Л.; Яненко, Н. Н. 1978) invariantais, atliekant funkcijų pakeitimą:

$$i = \frac{1}{2}(u_1 + u_2), \quad v = \frac{1}{2}(u_1 - u_2). \quad (4)$$

Įrašę (4) ir sugrupavę, (1) sistemą perrašome taip:

$$\begin{aligned} \frac{1}{2}(u_{1t} + u_{2t}) + \frac{1}{2}(u_{1x} - u_{2x}) + \frac{\varepsilon}{2} \alpha(t, x)(u_1 + u_2) &= 0, \\ \frac{1}{2}(u_{1t} - u_{2t}) + \frac{1}{2}(u_{1x} + u_{2x}) + \frac{\varepsilon}{2} \beta(t, x)(u_1 - u_2) &= 0. \end{aligned} \quad (5)$$

Sudėję (5) sistemos lygtis, o kitu atveju, atėmę iš (5) sistemos pirmos lygties antrąją lygtį, tuomet (5) sistema atrodytų:

$$\begin{aligned} u_{1t} + u_{1x} &= -\frac{1}{2} \varepsilon [u_1(\alpha(t, x) + \beta(t, x)) + u_2(\alpha(t, x) - \beta(t, x))], \\ u_{2t} - u_{2x} &= -\frac{1}{2} \varepsilon [u_1(\alpha(t, x) - \beta(t, x)) + u_2(\alpha(t, x) + \beta(t, x))]. \end{aligned} \quad (6)$$



Šiame darbe mes nagrinėsime (6) sistemos atskirą atvejį, kai  $\alpha(t, x) + \beta(t, x) = 0$ , tuomet turime, kad  $\alpha(t, x) = -\beta(t, x)$ .

Pažymėkime

$$\alpha(t, x) = -\beta(t, x) = -\sin(kx - wt). \quad (7)$$

Perrašome (6) sistemą tokiu pavidalu:

$$\begin{aligned} u_{1t} + u_{1x} &= \varepsilon u_2 \sin(kx - wt), \\ u_{2t} - u_{2x} &= \varepsilon u_1 \sin(kx - wt). \end{aligned} \quad (8)$$

Ieškome tolygiai tinkamo, kai  $0 \leq t < \frac{\tau_0}{\varepsilon}$  perio-

dinio pagal  $y_j$  asimptotinio sprendinio

$$u_j(t, x; \varepsilon) \approx U_j(\tau, y_j), \quad j = 1, 2,$$

čia  $\tau = \varepsilon t$ ,  $y_1 = x - t$ ,  $y_2 = x + t$ . Visos funkcijos

nagrinėjamos laiko  $t$  intervale nuo 0 iki  $\frac{\tau_0}{\varepsilon}$  ir, kai  $\varepsilon$  mažas gauname ilgą intervalą. Tai ir yra uždavinio sprendimas dviejų mastelių metodu, kur  $y_j$  - greitieji kintamieji, o  $\tau$  - lėtasis laikas (Найфэ А., 1984). Toliau sudarome suvidurkintą sistemą ir ieškome skleidinio didelėje srityje:

$$\frac{\partial U_j}{\partial \tau} = \langle F_j \rangle_j, \quad j = 1, 2, \quad (9)$$

arba

$$\begin{aligned} \frac{\partial U_1}{\partial \tau} &= \langle U_2 \sin(kx - wt) \rangle_1, \\ \frac{\partial U_2}{\partial \tau} &= \langle U_1 \sin(kx - wt) \rangle_2. \end{aligned} \quad (10)$$

Panašias integralines diferencialines sistemas nagrinėjo (Krylovas, A.; Čiegis, R. 2001: 458–470, Krylovas, A.; Čiegis, R. 2004: 209–222, Krylovas, A.; Miškinis P. 2007: 123–127, Krylovas, A. 2008: 47–54, Lavcel, O.; Krylovas, A. 2009: 41–46). (9) pavidalo sistemose gali atsirasti rezonansai, kurių tyrimas yra pakankamai sudėtingas.

(10) sistemos dešinėje pusėje yra užrašyti tokie vidurkinimo pagal charakteristikas operatoriai:

$$\langle U_2 \sin(kx - wt) \rangle_1 = \frac{1}{2\pi} \int_0^{2\pi} U_2(\tau, x + s) \sin(kx - ws) ds, \quad (11)$$

$$\langle U_1 \sin(kx - wt) \rangle_2 = \frac{1}{2\pi} \int_0^{2\pi} U_1(\tau, x - s) \sin(kx - ws) ds, \quad (12)$$

(11), (12) vidurkinimo operatoriai atvaizduoja funkcijas  $f(\tau, t, x)$  į funkcijas  $\langle f \rangle(\tau, y_j)$ , kai integruojant atitinkamai keičiame kintamuosius  $x = y_1 + s$  ir  $x = y_2 - s$ . Kai funkcijos  $U_j(\tau, y_j)$ , tada gauname

$$\begin{aligned} \langle U_2 \sin(kx - wt) \rangle_1 &= \\ &= \frac{1}{2\pi} \int_0^{2\pi} U_2(\tau, y_1 + s + s) \sin(k(y_1 + s) - ws) ds, \end{aligned} \quad (13)$$

$$\begin{aligned} \langle U_1 \sin(kx - wt) \rangle_2 &= \\ &= \frac{1}{2\pi} \int_0^{2\pi} U_1(\tau, y_2 - s - s) \sin(k(y_2 - s) - ws) ds, \end{aligned} \quad (14)$$

o sutraukę panašius narius, vidurkinimo operatoriai atrodo taip:

$$\begin{aligned} \langle U_2 \sin(kx - wt) \rangle_1 &= \\ &= \frac{1}{2\pi} \int_0^{2\pi} U_2(\tau, y_1 + 2s) \sin(ky_1 + s(k - w)) ds, \end{aligned} \quad (15)$$

$$\begin{aligned} \langle U_1 \sin(kx - wt) \rangle_2 &= \\ &= \frac{1}{2\pi} \int_0^{2\pi} U_1(\tau, y_2 - 2s) \sin(ky_2 - s(k + w)) ds. \end{aligned} \quad (16)$$

Taigi (10) sistemą galime perrašyti tokiu pavidalu:

$$\begin{aligned} \frac{\partial U_1}{\partial \tau} &= \frac{1}{2\pi} \int_0^{2\pi} U_2(\tau, y_1 + 2s) \sin(ky_1 + s(k - w)) ds, \\ \frac{\partial U_2}{\partial \tau} &= \frac{1}{2\pi} \int_0^{2\pi} U_1(\tau, y_2 - 2s) \sin(ky_2 - s(k + w)) ds. \end{aligned} \quad (17)$$

### Suvidurkintos sistemos Furjė analizė

Dabar skleidžiame (17) sistemos nežinomas funkcijas  $U_1(\tau, y_1)$  ir  $U_2(\tau, y_2)$  Furjė eilute:

$$U_1(\tau, y_1) = U_{10}(\tau) + U_{11c}(\tau) \cos y_1 + U_{11s}(\tau) \sin y_1 + U_{12c}(\tau) \cos 2y_1 + U_{12s}(\tau) \sin 2y_1 + \dots, \quad (18)$$

$$U_2(\tau, y_2) = U_{20}(\tau) + U_{21c}(\tau) \cos y_2 + U_{21s}(\tau) \sin y_2 + U_{22c}(\tau) \cos 2y_2 + U_{22s}(\tau) \sin 2y_2 + \dots \quad (19)$$

Tuomet (17) sistemos pirmąją lygtį galime užrašyti taip:

$$\begin{aligned} & \frac{dU_{10}}{d\tau} + \frac{dU_{11c}}{d\tau} \cos y_1 + \frac{dU_{11s}}{d\tau} \sin y_1 + \\ & \frac{dU_{12c}}{d\tau} \cos 2y_1 + \frac{dU_{12s}}{d\tau} \sin 2y_1 + \dots = \\ & = \frac{1}{2\pi} \int_0^{2\pi} U_{20}(\tau) \sin(ky_1 + s(k-w)) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{21c}(\tau) \cos(y_1 + 2s) \sin(ky_1 + s(k-w)) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{21s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + s(k-w)) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{22c}(\tau) \cos(y_1 + 2s) \sin(ky_1 + s(k-w)) ds + \\ & \frac{1}{2\pi} \int_0^{2\pi} U_{22s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + s(k-w)) ds + \dots, \end{aligned} \quad (20)$$

o antrąją lygtį:

$$\begin{aligned} & \frac{dU_{20}}{d\tau} + \frac{dU_{21c}}{d\tau} \cos y_2 + \frac{dU_{21s}}{d\tau} \sin y_2 + \\ & \frac{dU_{22c}}{d\tau} \cos 2y_2 + \frac{dU_{22s}}{d\tau} \sin 2y_2 + \dots = \\ & = \frac{1}{2\pi} \int_0^{2\pi} U_{10}(\tau) \sin(ky_2 - s(k+w)) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{11c}(\tau) \cos(y_2 - 2s) \sin(ky_2 - s(k+w)) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{11s}(\tau) \sin(y_2 - 2s) \sin(ky_2 - s(k+w)) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{12c}(\tau) \cos(y_2 - 2s) \sin(ky_2 - s(k+w)) ds + \end{aligned}$$

$$+ \frac{1}{2\pi} \int_0^{2\pi} U_{12s}(\tau) \sin(y_2 - 2s) \sin(ky_2 - s(k+w)) ds + \dots, \quad (21)$$

Tarkime, kad  $k-w$  yra sveikasis skaičius ir  $k \neq w$ . Tada skaičiuojame (20) lygybės dešiniąją pusę. Pirmasis

integralas  $\left( \frac{1}{2\pi} \int_0^{2\pi} U_{20}(\tau) \sin(ky_1 + s(k-w)) ds \right)$  yra

lygus nuliui, nes kosinusas yra pirmąją sinuso funkcija ir  $k-w$  – sveikasis skaičius, todėl šis integralas lygus nuliui. Pertvarkome antrąjį

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{21c}(\tau) \cos(y_1 + 2s) \sin(ky_1 + s(k-w)) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{21c}(\tau) \sin(y_1(1+k) + s(2+k-w)) ds + \\ & + \frac{1}{4\pi} \int_0^{2\pi} U_{21c}(\tau) \sin(y_1(k-1) + s(k-w-2)) ds \end{aligned}$$

ir ketvirtąjį

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{22c}(\tau) \cos(y_1 + 2s) \sin(ky_1 + s(k-w)) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{22c}(\tau) \sin(y_1(1+k) + s(2+k-w)) ds + \\ & + \frac{1}{4\pi} \int_0^{2\pi} U_{22c}(\tau) \sin(y_1(k-1) + s(k-w-2)) ds \end{aligned}$$

integralus.

Nagrinėsime atvejį, kai  $k-w=2$ , t.y.  $w=k-2$ , o šie integralai tuomet lygūs:

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{21c}(\tau) \cos(y_1 + 2s) \sin(ky_1 + s(k-w)) ds = \\ & = \frac{1}{2} U_{21c}(\tau) \sin y_1 (k-1), \\ & \frac{1}{2\pi} \int_0^{2\pi} U_{22c}(\tau) \cos(y_1 + 2s) \sin(ky_1 + s(k-w)) ds = \\ & = \frac{1}{2} U_{22c}(\tau) \sin y_1 (k-1). \end{aligned}$$

Lieka dar apskaičiuoti:

$$\frac{1}{2\pi} \int_0^{2\pi} U_{21s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + s(k-w)) ds$$

$$\text{ir } \frac{1}{2\pi} \int_0^{2\pi} U_{22s}(\tau) \sin(y_1 + 2s) \sin(ky_2 + s(k-w)) ds. \quad (22)$$

Toliau nagrinėsime atvejį, kai  $k-w=2$ . Tuomet  $w=k-2$ , o (22) galime užrašyti taip:

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{21s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + 2s) ds \\ & \text{ir} \\ & \frac{1}{2\pi} \int_0^{2\pi} U_{22s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + 2s) ds. \end{aligned} \quad (23)$$

Pertvarkome (23) pavidalo integralus:

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{21s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + 2s) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{21s}(\tau) \cos(y_1 + 2s - ky_1 - 2s) ds \\ & \quad - \frac{1}{4\pi} \int_0^{2\pi} U_{21s}(\tau) \cos(y_1 + 2s + ky_1 + 2s) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{21s}(\tau) \cos(y_1 + 2s - ky_1 - 2s) ds \\ & \quad - \frac{1}{4\pi} \int_0^{2\pi} U_{21s}(\tau) \cos(y_1 + 2s + ky_1 + 2s) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{21s}(\tau) (\cos((1-k)y_1) - \cos((1+k)y_1 + 4s)) ds, \\ & \quad \frac{1}{2\pi} \int_0^{2\pi} U_{22s}(\tau) \sin(y_1 + 2s) \sin(ky_1 + 2s) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{22s}(\tau) \cos(y_1 + 2s - ky_1 - 2s) ds \\ & \quad - \frac{1}{4\pi} \int_0^{2\pi} U_{22s}(\tau) \cos(y_1 + 2s + ky_1 + 2s) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{22s}(\tau) \cos(y_1 + 2s - ky_1 - 2s) ds \\ & \quad - \frac{1}{4\pi} \int_0^{2\pi} U_{22s}(\tau) \cos(y_1 + 2s + ky_1 + 2s) ds = \\ & = \frac{1}{4\pi} \int_0^{2\pi} U_{22s}(\tau) (\cos((1-k)y_1) - \cos((1+k)y_1 + 4s)) ds, \end{aligned} \quad (24)$$

Tada (20) lygybė atrodys taip:

$$\begin{aligned} & \frac{dU_{10}}{d\tau} + \frac{dU_{11c}}{d\tau} \cos y_1 + \frac{dU_{11s}}{d\tau} \sin y_1 + \\ & \frac{dU_{12c}}{d\tau} \cos 2y_1 + \frac{dU_{12s}}{d\tau} \sin 2y_1 + \dots = \\ & = \frac{1}{2} \cos(y_1(1-k)) U_{21s}(\tau) + \frac{1}{2} \sin(y_1(k-1)) U_{21c}(\tau) + \\ & + \frac{1}{2} \cos(y_1(1-k)) U_{22s}(\tau) + \frac{1}{2} \sin(y_1(k-1)) U_{22c}(\tau) \dots \end{aligned} \quad (25)$$

Analogiškai skaičiuojame (21) lygybės dešiniąją pusę. Įvedus pažymėjimą  $k-w=2$ , čia  $w=k-2$ , tuomet pirmasis integralas lygus:

$$\frac{1}{2\pi} \int_0^{2\pi} U_{10}(\tau) \sin(ky_2 - 2s(k-1)) ds, \quad (26)$$

o antrasis ir ketvirtasis

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{11c}(\tau) \cos(y_2 - 2s) \sin(ky_2 - 2s(k-1)) ds, \\ & \frac{1}{2\pi} \int_0^{2\pi} U_{12c}(\tau) \cos(y_2 - 2s) \sin(ky_2 - 2s(k-1)) ds. \end{aligned}$$

Pertvarkę šiuos du integralus, gauname

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} U_{11c}(\tau) \cos(y_2 - 2s) \sin(ky_2 - 2s(k-1)) ds = \\ & = \frac{1}{2\pi} \int_0^{2\pi} U_{11c}(\tau) \sin(y_2(1+k) - 2sk) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{11c}(\tau) \cos(y_2(k-1) - 2s(k-2)) ds, \\ & \frac{1}{2\pi} \int_0^{2\pi} U_{12c}(\tau) \cos(y_2 - 2s) \sin(ky_2 - 2s(k-1)) ds = \\ & = \frac{1}{2\pi} \int_0^{2\pi} U_{12c}(\tau) \sin(y_2(1+k) - 2sk) ds + \\ & + \frac{1}{2\pi} \int_0^{2\pi} U_{12c}(\tau) \cos(y_2(k-1) - 2s(k-2)) ds. \end{aligned} \quad (27)$$

Dabar lieka apskaičiuoti:

$$\frac{1}{2\pi} \int_0^{2\pi} U_{11s}(\tau) \sin(y_2 - 2s) \sin(ky_2 - s(k+w)) ds$$

ir

$$\frac{1}{2\pi} \int_0^{2\pi} U_{12s}(\tau) \sin(y_2 - 2s) \sin(ky_2 - s(k+w)) ds. \quad (28)$$

Atlikus pakeitimą  $k - w = 2$ , t.y.  $w = k - 2$ , tada (28) atrodo:

$$\frac{1}{2\pi} \int_0^{2\pi} U_{11s}(\tau) \sin(y_2 - 2s) \sin(ky_2 - s(2k - 2)) ds$$

ir

$$\frac{1}{2\pi} \int_0^{2\pi} U_{12s}(\tau) \sin(y_2 - 2s) \sin(ky_2 - s(2k - 2)) ds, \quad (29)$$

atlikę aritmetinius veiksmus, (29) integralus galime užrašyti taip:

$$\begin{aligned} & \frac{1}{4\pi} \int_0^{2\pi} U_{11s}(\tau) \cos(y_2(1-k) + 2s(k-2)) ds - \\ & - \frac{1}{4\pi} \int_0^{2\pi} U_{11s}(\tau) \cos(y_2(1+k) - 2ks) ds, \\ & \frac{1}{4\pi} \int_0^{2\pi} U_{12s}(\tau) \cos(y_2(1-k) + 2s(k-2)) ds - \\ & - \frac{1}{4\pi} \int_0^{2\pi} U_{12s}(\tau) \cos(y_2(1+k) - 2ks) ds. \end{aligned} \quad (30)$$

Iš (26), (27) ir (30) integralų matyti, kada atsiranda bangų sąveika. Ji atsiranda tada, kai šie integralai nelygūs nuliui. Pastebime, jog (26) integralas nelygus nuliui tada, kai  $k = 1$ , (27) ir (30) reiškiniai nelygūs nuliui, kai  $k = -1, k = 0, k = 1$  arba  $k = 2$ . Parodysime kelis atvejus, kaip atrodo paprastųjų diferencialinių lygčių sistema, esant keliems konkrečioms atvejams.

1 atvejis. Iš (26), (27), (30) pastebime, kai  $k \neq 2, k \neq 1$  ir  $k \neq 0$ , integralai lygūs nuliui. Tuomet remiantis (25), (26), (27) ir (30) lygybėmis, (17) sistemą užrašome tokiu pavidalu:

$$\begin{aligned} \frac{dU_{10}}{d\tau} + \frac{dU_{11c}}{d\tau} \cos y_1 + \frac{dU_{11s}}{d\tau} \sin y_1 + \dots &= \frac{1}{2} \cos(y_1(1-k)) U_{21s}(\tau) + \frac{1}{2} \sin(y_1(k-1)) U_{21c}(\tau) + \dots \\ \frac{dU_{20}}{d\tau} + \frac{dU_{21c}}{d\tau} \cos y_2 + \frac{dU_{21s}}{d\tau} \sin y_2 + \dots &= 0. \end{aligned} \quad (31)$$

Kai  $k = 3$ , tuomet (31) atrodo:

$$\begin{cases} \frac{dU_{10}}{d\tau} = 0, \frac{dU_{11s}}{d\tau} = 0, \frac{dU_{11c}}{d\tau} = 0, \frac{dU_{2js/c}}{d\tau} = 0, \\ \frac{dU_{12c}}{d\tau} = \frac{1}{2}(U_{21s}(\tau) + U_{22s}(\tau)), \\ \frac{dU_{12s}}{d\tau} = \frac{1}{2}(U_{21c}(\tau) + U_{22c}(\tau)). \end{cases} \quad (32)$$

2 atvejis. Kai  $k = 2$ , naudodami (25), (26), (27) ir (30) lygybes, (17) sistemą užrašome taip:

$$\begin{cases} \frac{dU_{10}}{d\tau} + \frac{dU_{11c}}{d\tau} \cos y_1 + \frac{dU_{11s}}{d\tau} \sin y_1 + \dots = \frac{1}{2} \cos y_1 U_{21s}(\tau) + \frac{1}{2} \sin y_1 U_{21c}(\tau) + \dots, \\ \frac{dU_{20}}{d\tau} + \frac{dU_{21c}}{d\tau} \cos y_2 + \frac{dU_{21s}}{d\tau} \sin y_2 + \dots = \frac{1}{2} \cos y_2 U_{11s}(\tau) + \frac{1}{2} \sin y_2 U_{11c}(\tau) + \dots \end{cases} \quad (31)$$

arba gauname paprastąją diferencialinių lygčių sistemą

$$\begin{cases} \frac{dU_{10}}{d\tau} = 0, \frac{dU_{20}}{d\tau} = 0, \frac{dU_{11c}}{d\tau} = \frac{1}{2}(U_{21s}(\tau) + U_{22s}(\tau)), \\ \frac{dU_{11s}}{d\tau} = \frac{1}{2}(U_{21c}(\tau) + U_{22c}(\tau)), \\ \frac{dU_{21c}}{d\tau} = \frac{1}{2}(U_{11s}(\tau) + U_{12s}(\tau)), \\ \frac{dU_{21s}}{d\tau} = \frac{1}{2}(U_{11c}(\tau) + U_{12c}(\tau)). \end{cases} \quad (32)$$

Nagrinėjame Koši uždavinį, kai  $u_1(0, x) = \sin x$ . Tuomet

$$\begin{cases} U_{10}(\tau) = 0, U_{20}(\tau) = 0, \\ U_{11c}(\tau) = 0, \\ U_{11s}(\tau) = \frac{\tau^2}{8}, \\ U_{21c}(\tau) = \frac{1}{2}\tau, \\ U_{21s}(\tau) = 0. \end{cases}$$

3 atvejis. Kai  $k = 0$ , tuomet gauname:

$$\begin{cases} \frac{dU_{10}}{d\tau} = 0, \frac{dU_{20}}{d\tau} = 0, \\ \frac{dU_{11c}}{d\tau} = \frac{1}{2}(U_{21s}(\tau) + U_{22s}(\tau)), \\ \frac{dU_{11s}}{d\tau} = \frac{1}{2}(U_{21c}(\tau) + U_{22c}(\tau)) \\ \frac{dU_{21c}}{d\tau} = -\frac{1}{2}(U_{11s}(\tau) + U_{12s}(\tau)), \\ \frac{dU_{21s}}{d\tau} = -\frac{1}{2}(U_{11c}(\tau) + U_{12c}(\tau)). \end{cases} \quad (33)$$

4 atvejis. Kai  $k = 1$ , tada paprastųjų diferencialinių lygčių sistema atrodo taip:

$$\begin{cases} \frac{dU_{10}}{d\tau} = \frac{1}{2}(U_{21s}(\tau) + U_{22s}(\tau)), \\ \frac{dU_{11c}}{d\tau} = 0, \frac{dU_{11s}}{d\tau} = 0, \\ \frac{dU_{20c}}{d\tau} = 0, \frac{dU_{21c}}{d\tau} = 0, \\ \frac{dU_{21s}}{d\tau} = U_{10}(\tau). \end{cases}$$

Gauname baigtinę paprastųjų diferencialinių lygčių sistemą. Rezultatas būtų kitas, jeigu turėtume netiesines sistemos lygtis. Nors pradiniu momentu turėtume vieną harmoniką, tačiau laikui bėgant atsirastų jų be galo daug. Tokių sistemų tyrimas gali būti ir skaitinis (Krylovas, A.; Čiegis, R. 2004: 209–222, Krylovas, A.; Čiegis, R. 2001: 458–470). Taip pat galima bandyti „nukirpti“ gaunamą begalinę paprastųjų diferencialinių lygčių sistemą ir palikti tik kelias pirmąsias harmonikas (Lavcel, O.; Krylovas, A. 2009: 41–46).

## Išvados

1. Atlikta sistemos asimptotinė analizė
2. Parodyta kaip telegrafinių lygčių sistemai taikyti vidurkinimo pagal charakteristikas techniką.
3. Nustatytos bangų sąveikos atsiradimo sąlygos.
4. Duotoji diferencialinių lygčių sistema pertvarkyta į baigtinę paprastųjų diferencialinių lygčių sistemą.
5. Numatomas tolimesnis tokių sistemų skaitinis ir analizinis tyrimas.

## Padėkos

Dėkoju prof. Aleksandrui Krylovui už pagalbą rengiant straipsnį.

## Literatūra

- Fliess, M.; Martin, P.; Petit, N.; Rouchon, P. Active signal restoration for the telegraph equation.
- Krylovas, A. 2008. Asymptotic Method for Approximation of Resonant Interaction of Nonlinear Multidimensional Hyperbolic Waves. *Mathematical Modelling and Analysis*, 13(1): 47–54.
- Krylovas, A.; Čiegis, R. 2004. Review of Numerical Asymptotic Averaging for Weakly Nonlinear Hyperbolic Waves. *Mathematical Modelling and Analysis*, 9(3): 209–222.
- Krylovas, A.; Čiegis, R. 2001. Asymptotic approximation of hyperbolic weakly nonlinear systems. *Nonlinear Math. Phys.*, 8(4): 458 – 470.

- Krylovas, A.; Miškinis P. 2007. Absoliučiai tamprios nesvarios stygos netiesiniai svyravimai. Asimptotikų konstravimas. *Liet. matem. rink.* 47 (spec nr): 123–127.
- Krylovas, A.; Miškinis P. 2007. Equations of a one-dimensional barotropic gas. (Nauka, Moscow), (Russian).
- Lavcel, O.; Krylovas, A. 2009. Absoliučiai tamprios nesvarios stygos netiesinių svyravimų asimptotikų tyrimas. *Lietuvos matematikos rinkinys.* 50 (spec. nr.): 41–46.
- Shaoyong, L. 1997. The asymptotic theory of semi-linear perturbed telegraph equation and it's application. *Applied Mathematics and Mechanics.* Vol. 18, No. 7: 657–662.
- Колесов, А. Ю.; Мищенко, Е. Ф.; Розов, Н. Х. 2000. Явление буферности в резонансных системах нелинейных гиперболических уравнений. *Успехи математических наук.* Том 55, № 2(332): 97–120.
- Колесов, А. Ю.; Розов, Н. Х. 2006. Явление буферности в одной неклассической гиперболической краевой задаче из радиофизики. *Математический сборник.* Том 197, № 6: 63–96.
- Kulikov, A. N. 2008. The attractors of two boundary value problems for a modified nonlinear telegraph equation. *Rus. J. Nonlin. Dyn.*, V.4, №1: 57–68.
- Найфэ А., 1984. *Введение в методы возмущений.* Москва: МИР. 535с. ББК 22.161.6 Н20, УДК 517.94
- Романовский, Р. К.; Мендзив, М. В. 2007. Устойчивость решений задачи Коши для гиперболической системы на плоскости с периодическими по времени коэффициентами. *Сибирский математический журнал.* Том 48, № 5: 1134–1141.
- Рождественский, Б. Л.; Яненко, Н. Н. 1978. *Системы квазилинейных уравнений и их приложения к газовой динамике.* Москва: Наука. 688с. 517.2 Р 62, УДК 517.944.

## ASYMPTOTICAL INVESTIGATION OF TELEGRAPH EQUATIONS SYSTEM

R. Kriauzienė

Abstract

The system of telegraph equations with variable coefficients are considered in the article. Looking for the asymptotic solution as part of the averaged system of equations. An integrated system of differential equations, which can lead to wave interactions is constructed. The averaged system can be transformed into a finite ordinary differential equations. The aim of the work is to show that the presented system, under certain conditions can describe interactions of the waves.

**Keywords:** telegraph system, averaged method, asymptotic integration, wave interaction.

## SKAITINIS ŠILUMOS LAIDUMO PROCESŲ MODELIAVIMAS KOMPOZITUOSE

Tadas Bulovas

Vilniaus Gedimino technikos universitetas  
El. paštas: [tbulovas@gmail.com](mailto:tbulovas@gmail.com);

**Santrauka.** Šiame darbe nagrinėjamas šilumos laidumo proceso kompozituose uždavinys. Uždavinio sprendimas pagrįstas medžiagos homogenizacijos teorija. Teorinis pagrindimas atliktas vienmačio modelio atveju. Skaitinis modelis realizuotas dvimačiam kompozitinės medžiagos atveju. Kaip pavyzdys sprendžiamas modelinis uždavinys terpėje kuria sudaro dvi medžiagos. Pilkosios medžiagos kurios šilumos laidumo koeficientas yra  $c$  yra 40% ,o baltosios medžiagos sudarančios 60% visos terpes šilumos laidumo koeficientas yra 1. Medžiagos geometrinė struktūra gerai žinoma. Skaičiuojant efektyvųjį nehomogenines kompozitinės medžiagos šilumos laidumo koeficientą sprendžiamas pagalbinis elipsinis diferencialinis šilumos laidumo uždavinys su periodinėmis kraštinėmis sąlygomis. Uždavinio sprendimo realizacija atlikta naudojant C++ programavimo kalbą ir „Matlab“ matematinį paketą.

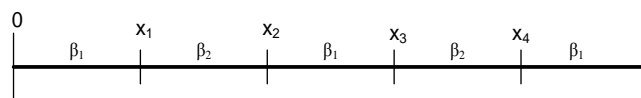
**Reikšminiai žodžiai.** šilumos laidumo uždavinys, efektyvusis šilumos laidumo koeficientas, nehomogeninė terpė, anizotropinė terpė

### Įvadas

Nagrinėsime tiesinį stacionarųjį šilumos laidumo nehomogeninėje terpėje modelį, paremtą kompozitinės medžiagos aproksimacija homogenine anizotropine terpe. Toks modelis plačiai taikomas aprašant šilumos laidumo procesą kompozitinėse medžiagose, bei natūraliai porėtose, pluoštinėse medžiagose, pvz., medienoje. Laikysime, kad šilumos laidumo proceso nehomogeninėje terpėje modelis aprašo toks pat procesas anizotropinėje homogeninėje terpėje, kurios savybės įvairiomis kryptimis skiriasi, tačiau nepriklauso nuo tiriamos srities taško. Izotropinės medžiagos šilumos laidumo koeficientas yra konstanta, tuo tarpu anizotropinės medžiagos turi šilumos laidumo koeficientų matricą. Tokios matricos koeficientų reikšmės modelyje yra atskirų medžiagų, sudarančių kompozitinę medžiagą, šilumos laidumo koeficientų harmoninis vidurkis. Jei dvi skirtingos izotropinės medžiagos, kurių geometrija gerai žinoma, turi bendrą sąlyčio paviršių, jos nagrinėjamos kaip viena kompozitinė medžiaga. Pastarosios šilumos laidumo koeficientas viename paviršiuje, kuris yra kompozitinės medžiagos viduje, greitai pakinta arba įvyksta “šuoliukas”, tai yra šilumos srautas pereina iš vienos homogeninės medžiagos į kitą. Homogenizacijos teorija leidžia apskaičiuoti efektyvųjį visos kompozitinės medžiagos šilumos laidumo koeficientą. Suskaičiavus efektyvųjį šilumos laidumo koeficientą, kompozitinę medžiagą galima nagrinėti kaip homogeninę anizotropinę, - “šuoliukų” nebelieka.

### Šilumos laidumo proceso nehomogeninėje terpėje homogeniniai modeliai vienmačiu atveju

Vienmatę kompozitinę medžiagą galima schematiškai pavaizduoti taip (1 pav.)



Pav. 1 Vienmatė kompozitinė medžiaga

Čia  $\beta_1$  ir  $\beta_2$  - dviejų skirtingų medžiagų šilumos laidumo koeficientai, medžiagų sandūras yra taškuose  $x_i$ . O  $\beta_1$  ir  $\beta_2$  kitimas pagal  $x$ , kai  $x \in [0,1]$  esant 4-iems trūkio taškams užrašomas taip:

$$\beta(x) = \begin{cases} \beta_1, & x \in [0, x_1], \\ \beta_2, & x \in (x_1, x_2], \\ \beta_1, & x \in (x_2, x_3], \\ \beta_2, & x \in (x_3, x_4], \\ \beta_1, & x \in (x_4, 1]. \end{cases}$$

Nusistovėjusį šilumos laidumo procesą aprašysime diferencialine lygtimi. Jei leisime, kad šiluma sklinda tik dėl difuzijos, tai lygties pavidalas bus

$$\frac{d}{dx} \left( \beta(x) \frac{dU}{dx} \right) = f(x). \quad (1)$$

Tarsime, kad temperatūra  $U(x)$  tenkina pirmojo tipo kraštinės sąlygas:

$$\begin{aligned} U(0) &= \mu_0, \\ U(1) &= \mu_1. \end{aligned}$$

## Efektivityo šilumos laidumo proceso koeficiento skaičiavimas vienmačiu atveju

Efektivusis šilumos laidumo proceso koeficientas – tai suvidurkintas visos kompozitinės medžiagos šilumos laidumo koeficientas. Šis koeficientas vienmačių atveju suskaičiuojamas pagal ilgį suintegravus šilumos srautą [A.Wiegmann, A.Zemitis.2006]:

$$\beta^* = \int_0^1 \beta(x) \left( \frac{dU}{dx} + 1 \right) dx; \quad (1)$$

Prieš skaičiuojant  $\beta^*$  yra sprendžiamas pagalbinis uždavinys. Tai diferencialinė lygtis su dešiniąja puse lygia nuliui ir vienetiniu vektoriumi  $\vec{e}_l$ , kuris šiuo atveju lygus 1, nes kryptis  $l$  yra tik viena ir ji sutampa su koordinatinių ašimi  $x$ . Šio uždavinio tikslas – sužinoti temperatūros pasiskirstymo funkcijos reikšmę  $U$ . Vienmačiu atveju sritis, kurioje ieškome sprendinio, yra atkarpa  $[0, 1]$  jei periodas yra 1. Uždavinys yra diferencialinė lygtis

$$\frac{d}{dx} \left( \beta \left( \frac{dU}{dx} + 1 \right) \right) = 0, \text{ kai } x \in (0,1), \quad (2)$$

su periodinėmis kraštinėmis sąlygomis

$$U(0) = U(1),$$

$$\beta(0) \frac{\partial U}{\partial x}(0) = \beta(1) \frac{\partial U}{\partial x}(1) + (\beta(1) - \beta(0)).$$

### Pagalbinio uždavinio sprendimas

Tegul turime vienmatį kompozitinį medžiagą, sudarytas iš dviejų medžiagos, turinčias penkias atkarpas su pastoviu šilumos laidumo koeficientu ir 4 trūkius, modelį. Tokio uždavinio sprendinį, kai  $\beta$  yra funkcija nuo  $x$ , galima užrašyti dalimis tiesinės funkcijos pavidalu:

$$U(x) = \begin{cases} \lambda_0 + \lambda_1 x, & x \in [0, x_1], \\ \lambda_0 + \lambda_1 x_1 + \lambda_2 (x - x_1), & x \in [x_1, x_2], \\ \lambda_0 + \lambda_1 x_1 + \lambda_2 (x_2 - x_1) + \lambda_3 (x - x_2), & x \in [x_2, x_3], \\ \lambda_0 + \lambda_1 x_1 + \lambda_2 (x_2 - x_1) + \lambda_3 (x_3 - x_2) + \lambda_4 (x - x_3), & x \in [x_3, x_4], \\ \lambda_0 + \lambda_5 (x - 1), & x \in [x_4, 1]. \end{cases} \quad (3)$$

Taigi turime 5 nežinomus koeficientus  $\lambda_i$  kai  $i=1, 2, 3, 4, 5$ . Žinoma, kad kiekviename trūkio taške  $x_i$  galioja tokia tolydumo sąlyga:

$$\beta_1 \left( \frac{dU}{dx} + 1 \right) \Big|_{x_i=x_i-0} = \beta_2 \left( \frac{dU}{dx} + 1 \right) \Big|_{x_i=x_i+0}.$$

Tai yra, išvestinės trūkio taške  $x_i$  iš dešinės ir iš kairės turi būti lygios. Panaudojant šią sąlygą galima užrašyti tokią keturių lygčių sistemą:

$$\begin{cases} \beta_1 (\lambda_1 + 1) = \beta_2 (\lambda_2 + 1), \\ \beta_2 (\lambda_2 + 1) = \beta_1 (\lambda_3 + 1), \\ \beta_1 (\lambda_3 + 1) = \beta_2 (\lambda_4 + 1), \\ \beta_2 (\lambda_4 + 1) = \beta_1 (\lambda_5 + 1). \end{cases}$$

Penktą lygtį gauname pasinaudoję temperatūros tolydumu paskutiniame trūkio taške:

$$\begin{cases} \lambda_0 + \lambda_1 x_1 + \lambda_2 (x_2 - x_1) + \lambda_3 (x_3 - x_2) + \lambda_4 (x - x_3), & x \in [x_3, x_4], \\ \lambda_0 + \lambda_5 (x - 1), & x \in [x_4, 1]. \end{cases}$$

Vietoj  $x$  įstatome paskutinio trūkio koordinatę  $x_4$  ir gauname lygybę:

$$\lambda_1 x_1 + \lambda_2 (x_2 - x_1) + \lambda_3 (x_3 - x_2) + \lambda_4 (x_4 - x_3) = \lambda_5 (x_4 - 1).$$

Sprendžiamą lygčių sistemą galima užrašyti matricine forma:

$$A \cdot \Lambda = F.$$

Kai vektorius  $\Lambda$  yra:

$$\Lambda^T = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5),$$

Matrica  $A$ :

$$A = \begin{pmatrix} x_1 & x_2 - x_1 & x_3 - x_2 & x_4 - x_3 & 1 - x_4 \\ \beta_1 & -\beta_2 & 0 & 0 & 0 \\ 0 & \beta_2 & -\beta_1 & 0 & 0 \\ 0 & 0 & \beta_1 & -\beta_2 & 0 \\ 0 & 0 & 0 & \beta_2 & -\beta_1 \end{pmatrix}$$

Vektorius  $F$ :

$$F^T = (0, \beta_2 - \beta_1, \beta_1 - \beta_2, \beta_2 - \beta_1, \beta_1 - \beta_2).$$

Išsprendus lygčių sistemą gautas rezultatas rodo, kad yra dvi skirtingos  $\lambda_1$  ir  $\lambda_2$ , o kitos kartojasi:

$$\lambda_1 = \lambda_3 = \lambda_5 \text{ ir } \lambda_2 = \lambda_4$$

$\lambda_1$  ir  $\lambda_2$  pasikartojimas yra ir tuomet, kai trūkių skaičius yra  $n$ , todėl galima užrašyti bendras  $\lambda_1$  ir  $\lambda_2$  išraiškas ir nebespręsti lygčių sistemos kiekvieną kartą pasikeitus trūkių vietai ar skaičiui. Bendrosios formulės:

$$\lambda_1 = \frac{(\beta_1 - \beta_2) \sum_{i=1}^n (-1)^{i-1} x_i}{\left( \left( \sum_{i=1}^n (-1)^{i-1} x_i \right) + 1 \right) \beta_2 - \beta_1 \sum_{i=1}^n (-1)^{i-1} x_i}, \quad (4)$$

$$\lambda_2 = \frac{\left( \left( \sum_{i=1}^n (-1)^{i-1} x_i \right) + 1 \right) (\beta_1 - \beta_2)}{\left( \left( \sum_{i=1}^n (-1)^{i-1} x_i \right) + 1 \right) \beta_2 - \beta_1 \sum_{i=1}^n (-1)^{i-1} x_i}.$$

Šio uždavinio atveju, kai yra 4 trūkiai,  $\lambda$  išraiškos atrodo taip:

$$\lambda_1 = \frac{(x_1 - x_2 + x_3 - x_4)(\beta_1 - \beta_2)}{(x_1 - x_2 + x_3 - x_4 + 1)\beta_2 - (x_1 - x_2 + x_3 - x_4)\beta_1},$$

$$\lambda_2 = \frac{(x_1 - x_2 + x_3 - x_4 + 1)(\beta_1 - \beta_2)}{(x_1 - x_2 + x_3 - x_4 + 1)\beta_2 - (x_1 - x_2 + x_3 - x_4)\beta_1}.$$

$\beta^*$  formulę taip pat galima užrašyti bendru atveju, kai yra  $n$  trūkių, suintegravus pagalbinio uždavinio sprendinį:

$$\beta^* = \beta_1(\lambda_1 + 1) \left( \left( \sum_{i=1}^n (-1)^{i-1} x_i \right) + 1 \right) + \beta_2(\lambda_2 + 1) \left( \sum_{i=1}^n (-1)^i x_i \right) \quad (4)$$

Efektyvusis šilumos laidumo koeficientas, kai yra keturi trūkiai, kurių koordinatės  $x_1, x_2, x_3, x_4$  skaičiuojamas pagal formulę:

$$\beta^* = \beta_1(\lambda_1 + 1)(x_1 - x_2 + x_3 - x_4 + 1) + \beta_2(\lambda_2 + 1)(-x_1 + x_2 - x_3 + x_4)$$

### Šilumos laidumo proceso nehomogeninėje terpėje skaitinis modeliavimas dvimačių atveju.

Efektyvusis šilumos laidumo koeficientas skaičiuojamas nehomogeninei medžiagai. Anizotropinės medžiagos efektyvieji šilumos laidumo koeficientai dvimačiu atveju užrašomi matrica [A.Wiegmann, A.Zemitis.2006]:

$$\beta^* = \begin{pmatrix} \beta_{11}^* & \beta_{12}^* \\ \beta_{21}^* & \beta_{22}^* \end{pmatrix}.$$

Ši matrica apskaičiuojama naudojant homogenizacijos teoriją. Dvimačiu atveju sprendžiami du skirtingi singularūs periodinių kraštinių sąlygų uždaviniai:

$$\nabla \cdot (\beta(\vec{x})(\nabla U_l + \vec{e}_l)) = 0, \text{ kai } l=1, 2 \quad (5)$$

$$U_l(\vec{x} + id_1\vec{e}_1 + jd_2\vec{e}_2) = U_l(\vec{x}), \text{ kai } i, j \in Z$$

kur  $\vec{e}_l$  yra vienetinis vektorius  $x_l$  kryptimi. O  $d_1$  ir  $d_2$  - tiriamos srities geometriniai parametrai. Singuliarūs uždavinius galim išrašyti taip:

a) kryptimi  $l=1$ :

$$\frac{\partial}{\partial x_1} \beta(x_1, x_2) \left( \frac{\partial U_1}{\partial x_1} + \vec{e}_1 \right) + \frac{\partial}{\partial x_2} \beta(x_1, x_2) \left( \frac{\partial U_1}{\partial x_2} + \vec{e}_1 \right) = 0,$$

$$U_1(\vec{x} + id_1\vec{e}_1 + jd_2\vec{e}_2) = U_1(\vec{x}), \text{ kai } i, j \in Z;$$

b)  $l=2$ :

$$\frac{\partial}{\partial x_1} \beta(x_1, x_2) \left( \frac{\partial U_2}{\partial x_1} + \vec{e}_2 \right) + \frac{\partial}{\partial x_2} \beta(x_1, x_2) \left( \frac{\partial U_2}{\partial x_2} + \vec{e}_2 \right) = 0,$$

$$U_2(\vec{x} + id_1\vec{e}_1 + jd_2\vec{e}_2) = U_2(\vec{x}), \text{ kai } i, j \in Z;$$

Apskaičiuojami tolydūs sprendiniai  $U_1, U_2$ . Šilumos laidumo proceso efektyviųjų koeficientų reikšmių matricos elementų skaičiavimas yra sprendinio išvestinės integravimas skaičiuojamoje srityje dviem kryptimis:  $m=1, 2$  prieš šilumos srauto kryptį  $l=1, 2$

$$\beta_{ml}^* = \frac{1}{d_1 \cdot d_2} \int_{\Omega} \langle \vec{e}_m, \beta(\vec{x})(\nabla U_l + \vec{e}_l) \rangle d\vec{x}, \quad (6)$$

kai  $l=1, 2; m=1, 2$ .

$\langle \vec{e}_m, \beta(\vec{x})(\nabla U_l + \vec{e}_l) \rangle$  yra skaliarinė sandauga.  $\Omega$  - nagrinėjama sritis, dvimačiu atveju tai yra stačiakampis  $d_1 \times d_2$ . Pagrindinė problema – kraštinio uždavinio (5) sprendimas. Šilumos srautas  $\beta \nabla U_l$  yra tolydus tik vienetiniam vektoriui  $\vec{e}_l$  statmena kryptimi ir turi trūkius  $\vec{e}_l$  kryptimi. Tai reiškia, kad kai  $\beta$  yra konstanta, tai  $f=0$ , o kai  $\beta$  trūki, tai  $f$  yra vienareikšmė. Galima užrašyti trūkio sąlygą:

$$[U_l]_{\vec{x}=\vec{x}^*} = 0,$$

$$\left[ \beta \frac{\partial U}{\partial x_i} \right]_{\vec{x}=\vec{x}^*} = -\delta_{il} [\beta]_{\vec{x}=\vec{x}^*} \text{ kai } i = 1, 2.$$

$\partial x_i$  nurodo, kokia koordinatė ieškoma išvestinės,

$\delta_{il}$  - Kronekerio simbolis

$$\delta_{il} = \begin{cases} 1, & \text{kai } i = l, \\ 0, & \text{kai } i \neq l, \end{cases}$$

$[g]_{\vec{x}=\vec{x}^*}$  - trūkis  $\vec{x} = \vec{x}^*$  paviršiuje.

### Dvimačio uždavinio formulavimas

Sprendžiame pagalbinį efektyviojo šilumos laidumo koeficientų radimo uždavinį matriciniu forma iš šaltinio straipsnio [A.Wiegmann, A.Zemitis.2006]:

$$\begin{bmatrix} A & \Psi \\ D & I \end{bmatrix} \begin{bmatrix} U \\ J \end{bmatrix} = \begin{bmatrix} 0 \\ F_2 \end{bmatrix}, \quad (7)$$

Kur  $U$  vektorius temperatūros reikšmių tinklo taškuose,  $J$  yra temperatūros reikšmė kur  $\beta$  yra trūkus.  $J$  vektoriaus ilgis yra  $m = m_1 + m_2$ , kai  $m_i$  yra trūkių skaičius  $i$  kryptimi.  $U$  ilgis yra  $n = n_1 \cdot n_2$ .  $A$  submatrica yra  $n \times n$  dydžio ir ji yra Laplaso operatorius su periodinėm kraštinėm sąlygom. Submatricos  $\Psi$



nenuliniai koeficientai yra  $-\frac{1}{2h}$  ir eilutėse nu rodo ar

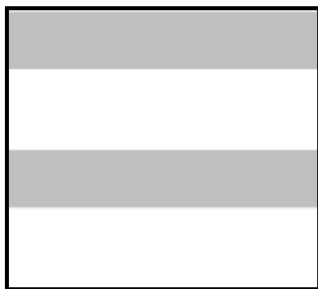
atitinkamos eilutes laštelės centras yra kaimynystėje su šilumos laidumo koeficiento trūkiu, jos dydis  $n \times m$ . Submatrica  $D$  yra šilumos laidumo koeficiento diskretizacija kiekviename tinklo taške su kuriuo jis turi bendrą karštinę. Nenuliniai submatricos  $D$  koeficientai yra medžiagų tarp kuriu vyksta šilumos difuzija šilumos laidumo koeficientų vidurkis, matricos dydis  $m \times n$ . Submatrica  $I$  yra vienetinė matrica dydžio  $m \times m$ .

Pastebėkime, kad suma visu  $F$  yra lygi nuliui  $\sum F = 0$ , taip pat  $\sum DU = 0$ . Tai leidžia numatyti, kad vienintelis sprendinys  $(U, J)$  egzistuoja tik tuomet kai  $J$  atitinka sąlyga  $\sum J = 0$ .

Nagrinėjame periodine dvimate struktūra (2 pav.) kurios šilumos laidumo koeficientų kitimas užsirašo formule:

$$\beta(x, y) = \begin{cases} c: & y \in [5k, 2 + 5k), k \in (-\infty, \infty) \\ 1: & y \in [2 + 5k, 5 + 5k), k \in (-\infty, \infty) \end{cases}$$

Tokios medžiagos periodo ilgis 5. Tokio medžiagoje yra 40% pilkos ( $\beta_1 = c$ ) medžiagos ir 60% baltos ( $\beta_2 = 1$ ).



Pav 2 Dvimatės medžiagos struktūra

Tokia medžiaga neturi trukių x kryptimi, todėl jos šilumos laidumo koeficientu tenzorius:

$$\beta^* = \begin{pmatrix} \beta_{11}^* & 0 \\ 0 & \beta_{22}^* \end{pmatrix}$$

Šiuo atveju reikšmes  $\beta_{11}^*$  ir  $\beta_{22}^*$  galima nesunkiai suskaičiuoti analiziškai.  $\beta_{11}^*$  yra aritmetinis, o  $\beta_{22}^*$  harmonine šilumos koeficientų reikšmės.

$$\beta_{11}^* = \frac{2}{5}c + \frac{3}{5} \cdot 1,$$

$$\beta_{22}^* = \left( \frac{2}{5} \cdot \frac{1}{c} + \frac{3}{5} \cdot \frac{1}{1} \right)^{-1}.$$

Pasirenkame  $c=5$ . Tada  $\beta_{11}^* = 2.6$ , o  $\beta_{22}^* = 1.47$ .

### Realizacija

Nagrinėjama 2D erdvė  $d \times d = 5 \times 5$ . Padaliname visa erdvę į diskretų tinklą pasirinktu žingsniu  $h$ . Eidami kiekvienu

tinklo tašku suskaičiuojame trūkių skaičių kryptimi  $l=1$  ir  $l=2$ . Išskiriame reikalingus duomenų masyvus. Juos užpildome sprendami uždavinį (7).

Pagal nagrinėjama periodine medžiaga aprašome submatrica  $A$  iš matricinės uždavinio formos (7). Submatricos  $A$  pagrindinėje įstrižainėje surašome

$$A_{i,j} = \frac{-4}{h^2}, \text{ kai } i=j.$$

Pagal periodine medžiagos struktūra formuluojama likusi matricos dalis koeficientai:

$$A_{i,j} = \frac{1}{h^2}.$$

Submatricos  $\Psi$  elementus užpildome pagal  $A$  matricos elementų išdėstymą. Visi jos nenuliniai elementai lygus:

$$\Psi_{i,j} = \frac{\pm 1}{2} h.$$

Submatricą  $D$  užpildome pasinaudoję submatrica  $\Psi$ . Eidami matricos  $\Psi$  elementais kiekvienu stulpeliu ieškome nenolinių elementų. Kiekvienam nenuliniam  $\Psi$  elementui submatricoje  $D$  eilutėje kurios skaičius lygus elemento submatricoje  $\Psi$  stulpeliu skaičiui priskiriame koeficientą:

$$\Psi_{i,j} = \pm \frac{2(c-1)}{(c-1)h}.$$

Gauta lygčių sistema sprendžiame Zeidelio metodu. Gautus rezultatus integruojame skaitiškai pasinaudoję stačiakampių formule:

$$\beta_{ml} \approx \frac{1}{d_2 \cdot d_2} \cdot \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \beta_{i,j} \cdot (D_{m;i,j} + \delta_{ml}) \cdot h^2,$$

Čia  $D_{m;i,j}$  yra :

$$D_{1;i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2 \cdot h} - \frac{[\partial x_1 u]_{i+0.5,j} - [\partial x_1 u]_{i-0.5,j}}{4},$$

$$D_{2;i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2 \cdot h} - \frac{[\partial x_1 u]_{i,j+0.5} - [\partial x_1 u]_{i,j-0.5}}{4}.$$

Po to, kai gauti efektyvieji šilumos laidumo koeficientai, sprendžiamas diferencialinis uždavinys :

$$\beta_{11} \frac{\partial^2 u}{\partial x_1^2} + \beta_{22} \frac{\partial^2 u}{\partial x_2^2} = f(x_1, x_2)$$

Šį diferencialinį uždavinį sprendžiame pasinaudoję matematinių paketo „MATLAB“ plėtinių „pdetool“. Kraštinės sąlygos pasirenkamos Dirichle tipo. Išorinio šaltinio funkcija pasirenkame konstantą visoje erdvėje:

$$f(x_1, x_2) = const.$$

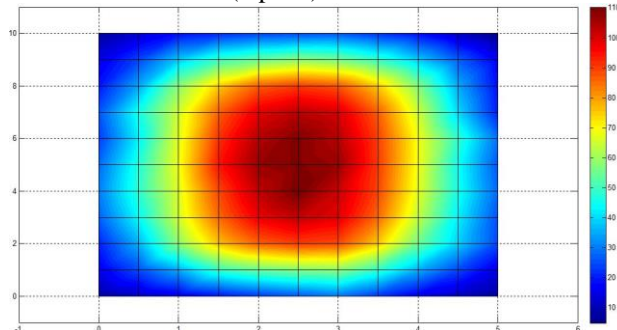
### Modeliavimo pavyzdžiai

#### Pirmas pavyzdys

Kontrastingumas  $c=5$ , išorinio šaltinio funkcija  $f=-5$ . Žingsnis  $h=1$ , Zeidelio metodu [R.Čiegis. 1997]

sprendžiamos lygčių sistemos paklaida  $\varepsilon=0.001$ . Reikalingas tikslumas pasiekiamas per 39 metodo iteracijas. Tikslios efektyviojo šilumos laidumo koeficiento reikšmės  $\beta_{11}^* = 2.6000$ , o  $\beta_{22}^* = 1.4706$ .

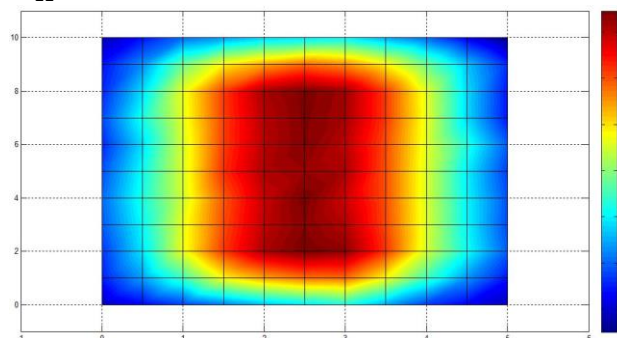
Skaitinės reikšmės  $\beta_{11} = 2.6004$ , o  $\beta_{22} = 1.4703$ . Tokio uždavinio rezultatas (3 pav.).



Pav. 3 Rezultatų pavyzdys 1

### Antras pavyzdys

Kontrastingumas  $c=100$ , išorinio šaltinio funkcija  $f=-100$ . Žingsnis  $h=1$ , Zeidelio metodu sprendžiamos lygčių sistemos paklaida  $\varepsilon=0.001$ . Tikslios efektyviojo šilumos laidumo koeficiento reikšmės  $\beta_{11}^* = 40.6000$ , o  $\beta_{22}^* = 1.6556$ . Skaitinės reikšmės  $\beta_{11} = 40.608$ , o  $\beta_{22} = 1.6553$ . Tokio uždavinio rezultatas (4 pav.).



Pav. 4 Rezultatų pavyzdys 2

Maksimali temperatūra  $u=7.7029$ . Dėl kelis šimtus kartų didesnio efektyviojo šilumos laidumo  $y$  kryptimi temperatūros funkcijos  $u$  reikšmės sumažėja visoje erdvėje, nes didelį poveikį duoda kraštinės sąlygos kuriuose užduotas pastovus 0.

### Išvados

1. Atliktas darbas apie šilumos laidumo proceso modeliavimą kompozitinėse medžiagose metodus parodo, kad skaičiuojant efektyvius šilumos laidumo koeficientus skaitiškai gaunamas rezultatas yra norimo tikslumo, kuris priklauso nuo pasirinkto tiesinių lygčių sprendimo metodo ir medžiagos aproksimacijos tikslumo užduodamo žingsniu  $h$ .

2. Remiantis pateiktais skaitiniais eksperimentais galima nustatyti, kad modelis atitinka fizikinę uždavinio prasmę ir aproksimuoja šilumos laidumo nehomogeninėje terpėje procesą.

### Literatūra

R. Čiegis. Diferencialinių lygčių skaitiniai sprendimo metodai. Vilnius: Technika 2003. 448 p.

R. Čiegis, V. Būda. Skaičiuojamoji matematika. Vilnius: TEV 1997. 240 p.

R. Čiegis, A. Ilgevičius, H. Liess, M. Meilūnas, O. Subač. Mathematical modeling and analysis. Numerical simulation of the heat conduction in electrical cables. Vilnius: Technika 2007. Volume 12, Number 4. 425-439 p

A. Wiegmann, A. Zemitis. EJ-HEAT: A fast explicit jump harmonic averaging solver for the effective heat conductivity of composite materials. Kaiserslautern: Fraunhofer ITWM 2006. Berichte des Fraunhofer ITWM, 94. 36 p.

Alireza Kaboorani. Thermal Properties of Composites Made of Heat-treated Wood and Polypropylene. 2599 originally published online Aug 13, 2009; Journal of Composite Materials 43

David Kahaner. „Numerical Methods and Software“, Prentice-Hall 1989, 540p.

D. Hull, T. W. Clyne. „An Introduction to Composite Materials“. Cambridge University Press 1996. 320p.

K. Kow. „Mechanics of composite materials“. Florida: 1997. 323p.

### Abstract

A model of heat conduction process in heterogeneous materials is analyzed in this work. Model is justified in homogenizing theory. The theoretical justification is made for an one-dimensional case material. Two-dimensional numerical model in composite materials is realized. As an example task in environment which consists of two materials resolved. Grey material with a heat conduction coefficient  $c$  is 40%, while white material represents 60% of all composite and its heat transfer coefficient is 1. The geometric structure of the material is well known. To find effective thermal conductivity coefficient of non-homogeneous composite material the elliptical differential heat conduction problem with periodic boundary conditions is solved. Model is realized done using C++ programming language.

**Keywords:** heat conduction problem, effective thermal conductivity coefficient, heterogeneous material, anisotropic material



# MATEMATINIS MODELIAVIMAS

---

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“,  
įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

## GSM TINKLO DAŽNIŲ PASKIRSTYMO PLANAVIMAS

**Vitalijus Marazas**

*Vilniaus Gedimino Technikos Universitetas,*

[vitalijus.marazas@gmail.com](mailto:vitalijus.marazas@gmail.com);

**Anotacija.** Nagrinėjamas šiuolaikinis uždavinys apie optimalaus dažnių paskirstymą vartotojams mobiliojo GSM ryšio tinkluose. Aptariama: pagrindinės problemos su kuriomis susiduriama paskirstant GSM tinklo dažnius vartotojams, GSM tinklo aprašymas matematiiniu modeliu, matematinio modelio sąryšis su grafų teorija, pačios problemos aprašymas.

### **Įvadas**

GSM – populiariausias standartas mobiliesiems telefonams pasaulyje. Šiandien mobiliaisiais telefonais naudojami daugiau kaip milijardas žmonių 200 pasaulio šalyse. GSM tinklo egzistavimas leidžia naudotis ne tik savo šalies operatoriais, bet ir kitų šalių operatoriais. Pagrindinis skirtumas GSM nuo jo pirmtakų yra, kad visi jo kanalai yra skaitmeniniai, todėl GSM laikoma antros kartos mobiliųjų telefonų sistema. GSM tapus populiariausiu mobiliojo ryšio sistema, fiksuoto telefono sistemos tapo pasenusiomis ir jomis mažiau jau yra naudojamos. Toks GSM populiarumas yra iššūkis tinklo operatoriams, todėl jie yra priversti optimizuoti savo tinklus, kad tinklas veiktų kuo efektyviau, be jokių trukdžių.

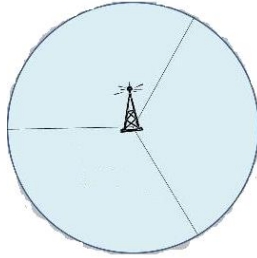
## GSM tinklo struktūra

GSM tinklo struktūra skirstoma į 5 komponentus:

- **Mobiliosios stotys (MS).** Jos susideda iš fizinę įrangos, kurią sudaro radijo siųstuvas ir imtuvai, vaizdo ir skaitmeninių signalų procesoriai bei SIM kortelė. Dažniausiai tai yra mobilusis telefonas.
- **Bazinės stotys.** Jos yra sudarytos iš dviejų dalių: pagrindinio siųstuvo/imtuvo (BTS) ir pagrindinės stoties kontrolierio (BSC). Jie yra sujungti specialia ABIS sąsaja, leidžiančia vykdyti operacijas tarp komponentų, pagamintų skirtingų tiekėjų. Pagrindinis siųstuvas susideda iš radijo siųstuvų, kurie nustato telefono ląstelę ir valdo ryšio protokolus su mobiliuoju telefonu. Pagrindinis stoties kontrolieris valdo radijo resursus vienos ar daugelio BTS stočių.
- **Komunikacijų centras.** Sudėtingiausias tinklo elementas, kuris susijungia su kitais tinklais: nacionaliniais, tarptautiniais mobiliojo ir fiksuoto ryšio, nuolatos tikrina, kurioje tinklo vietoje yra MS, sujungia abonentus ir palaiko sujungimą visoje tinklo teritorijoje, skaičiuoja ryšių trukmes ir formuoja apmokestinimo informaciją.
- **Duomenų bazė (DB).** Visa informacija reikalinga komunikacijų centro darbui, kaupiama ir apdorojama duomenų bazėse. Jose saugoma informacija apie abonentui teikiamas paslaugas, individualūs informacijos kodavimo raktai, MS buvimo vieta.
- **Operacijų aptarnavimo centras (OAC).** OAC pastebi ir pašalina įvairius sutrikimus tinkle.

## Pagrindinio siųstuvo struktūra

Kiekvienas pagrindinis siųstuvas yra sudarytas iš ląstelės ir jose esančių sektorių (1 arba 3) (1 pav.). Kiekviename sektoriuje yra kelias siųstuvų/imtuvų (6-7, arba 8-12), kuriems yra priskiriami kanalai. Taip pat, kiekvienam siųstuvui/imtuvui yra priskiriami ir blokuoti kanalai, kurie yra nenaudojami dėl valdžios, karinių reikalavimų ir panašiai.

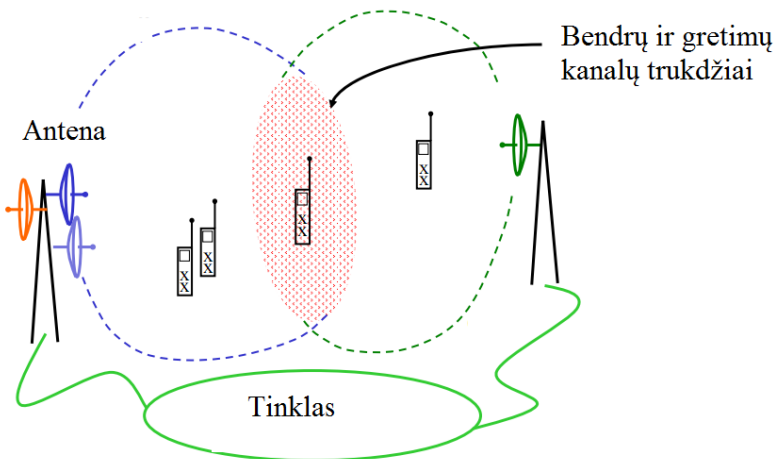


1 pav. Pagrindinio siųstuvo ląstelė ir jos sektoriai

### Dažnių planavimo problemos

Vienas iš svarbiausių GSM tinklo planavimo uždavinių yra išspręsti GSM tinklų trukdžius. Trukdžiai yra dviejų tipų (2 pav.) :

1. **Bendrų kanalų.** Tai tokie trukdžiai, kurie atsiranda, kai MS yra kelių BTS persidengime ir keli BTS siūlo tą patį kanalą.
2. **Gretimų kanalų.** Tai tokie trukdžiai, kurie atsiranda, kai MS yra kelių BTS persidengime ir keli BTS siūlo kanalus, kurie skiriasi vienetu.



2 pav. Bendrų ir gretimų kanalų trukdžiai

## Matematinis modelis

Tarkim turim neorientuotą grafą  $G = (V, E)$ . Grafo viršūnės vadinamos *nešėjais* ir atvaizduoja siųstuvus/imtuvus esančius pagrindinio siųstuvo ląstelėje. Aibė  $C$  yra baigtinis neneigiamų skaičių intervalas, vaizduojantis visus įmanomus kanalus. Kiekvienam *nešėjui*  $v \in V$  priskirta aibė  $B_v \subseteq C$ , kuri žymi blokuotus kanalus. Blokuotų kanalų aibė gali būti ir tuščia. Kanalai  $C \setminus B_v$  vadinami *nešėjo  $v$  laisvaisiais kanalais*. Briaunoms  $E$  yra nurodyti parametrai  $c^{co}$  ir  $c^{ad}$  iš intervalo  $[0;1]$ . Kiekvienai briaunai  $(v,w)$  iš  $E$  suteikiamas atskyrimas tarp nešėjų  $v$  ir  $w$ , kuris yra žymimas  $d(v,w)$ . *Atskyrimu* yra vadinamas atstumas, esantis tarp siųstuvų/imtuvų. Bendri ir gretimi kanalų trukdžiai, kurie gali įvykti tarp  $v$  ir  $w$ , žymimi  $c^{co}(vw)$  ir  $c^{ad}(vw)$ . Pažymima funkcija  $N = (V, E, C, \{B_v\}_{v \in V}, d, c^{co}, c^{ad})$ , kuri yra vadinama *nešėjų tinklu* arba tiesiog *tinklu*. *Dažnių paskirstymas* arba priskyrimas  $N$  yra funkcija  $y: V \rightarrow C$ . Šis priskyrimas yra įmanomas, jeigu kiekvienam *nešėjui*  $v \in V$  priskirtas kanalas ir tenkinamos tokios sąlygos:

$$\begin{aligned} y(v) &\in C \setminus B_v \quad \forall v \in V \\ |y(v) - y(w)| &\geq d(vw) \quad \forall vw \in E \end{aligned} \quad (1)$$

Sprendžiame optimizacijos problemą

$$\min_y \sum_{\substack{vw \in E \\ y(v)=y(w)}} c^{co}(vw) + \sum_{\substack{vw \in E \\ |y(v)-y(w)|=1}} c^{ad}(vw) \quad (2)$$

Tikslas yra priskirti turimus dažnius taip, kad būtų minimizuoti bendrų ir gretimų kanalų trukdžiai.

Toks uždavinys yra susijęs su grafų teorija ir grafų spalvinimo uždaviniu. Grafo  $G$  spalvinimo problema yra vadinama *T-spalvinimu*. Grafo viršūnės reikia taip nuspalvinti, kad gretimos viršūnės gautų skirtingas spalvas.

## **Literatūra**

Borndorfer, R.; Eisenblatter, A.; Grottschel, M.; Martin, A. Frequency Assignment in Cellular Phone Networks. Berlin: 1997, p. 2-5.

Eisenblatter, A. Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds, 2001, p. 5-12, 33-37.

Eisenblatter, A.; Grottschel, M.; Koster, A. M. C. A. Frequency Planning and Ramifications of Coloring. Berlin: 2000, p. 4-7.

Grottschel, M. Frequency Assignment in Mobile Phone Systems. Berlin: 2000, p. 2-8.

Mobilūs ryšiai [interaktyvus].2005 [Žiūrėta 2009m. spalio 23 d.]. Prieiga per internetą:

<[http://www.straipsniai.lt/Mobilus\\_ryusiai/puslapis/9236](http://www.straipsniai.lt/Mobilus_ryusiai/puslapis/9236) >

## **FREQUENCY ASSIGNMENT IN GSM NETWORKS**

### **Vitalijus Marazas**

#### **Summary**

Modern task about optimum frequency allocation to users in cellular GSM networks is analyzed in this article. These parts are considered: Main problems occurring when allocating frequencies of GSM network to users; Mathematical modeling of GSM network; Relationship of mathematical model with graph theory; Description of the problem itself



# MATEMATIKA

13-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“, įvykusios Vilniuje 2010 m. balandžio mėn. 9 d., medžiaga

## PIEŠTINIŲ JAPONŲ KKRYŽIAŽODŽIŲ SPRENDIMAS

Tadas Markūnas

*Vilniaus Gedimino technikos universitetas*

**Anotacija.** Nagrinėjamas pieštinių japonų kryžiažodžių uždavinys, aptariami įvairūs šio uždavinio sprendimo būdai: variantų perrinkimo, euristiniai. Pateikiamos loginio algoritmo realizavimo detalės, tobulinimo planai ir rezultatai, sprendžiant uždavinius šiuo algoritmu.

### Įvadas

Pieštiniai japonų kryžiažodžiai yra loginių galvosūkių tipas, kurių sprendimo rezultate gaunamas piešinys. Šie kryžiažodžiai atsirado Japonijoje 1987 metais. Kryžiažodį sudaro lentelė, sudaryta iš langelių ir skaičiai prie lentelės eilučių ir stulpelių. Skaičiai rodo iš eilės nuspalvintų langelių kiekį eilutėje (stulpelyje). Keli skaičiai (pvz., 5, 2, 1) parodo, kad bus trys nuspalvintos sritys po 5, 2 ir 1 langelį. Sritis turi būti atskirtos bent vienu nenuspalvintu langeliu. Kryžiažodis išsprendžiamas kai nuspalvinami visi reikalingi langeliai.

	1			1					
	1		2		1				
	1	3	2	3	1				
1.1.1									
3									
2.2									
3									
1.1.1									

	1			1					
	1		2		1				
	1	3	2	3	1				
1.1.1		•		•					
3	•							•	
2.2				•					
3	•							•	
1.1.1		•		•					

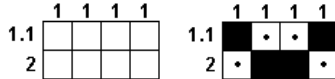
1 pav. Pradinio ir užpildyto kryžiažodžio pavyzdys

## Kryžiažodžio sprendimas

Pieštinių japonų kryžiažodžių sprendimas yra priskiriamas *NP-complete* uždavinių klasei (Ueda, Nagao 1996). Tai reiškia, kad nėra žinomo sprendimo būdo, kuriuo galima išspręsti šį uždavinį per polinominį laiką. Todėl dažniausiai šie kryžiažodžiai sprendžiami panaudojant euristikas. Pilnas variantų perrinkimas, dėl didelių vykdy- mo kaštų, naudojamas tik nesudėtingiems ir nedideliems kryžiažodžiams spręsti.

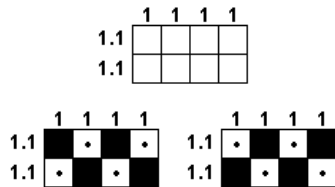
## Uždavinio problematiškumas

Sprendžiant pieštinius japonų kryžiažodžius, kartais susiduriama su problema, kai tolesnis sprendimas reikalauja spėjimo, t.y. kai vienu metu sprendžiama tik viena eilutė (stulpelis) negalima viena- reikšmiškai pasakyti, kuris langelis turi būti nuspalvintas ar nenus- palvintas (2 pav.):



2 pav. Spėjimo problemos pavyzdys

Spėjimas vykdomas paprastai – atsitiktinai parinktą langelį pažymime kaip nuspalvintą (nenuspalvintą) ir jei toliau sprendžiant kryžiažodį pastebimas sąlygų netenkinimas, spėtasis langelis pažy- mimas kaip nenuspalvintas (nuspalvintas). Šiame pavyzdyje yra tik vienas galimas sprendinys, tačiau yra ir tokių situacijų, kai galimi ir keli sprendiniai (3 pav.):



3 pav. Kelių sprendinių problemos pavyzdys

## Loginis algoritmas

Loginis algoritmas grindžiamas logika, kuria naudojasi žmogus sprenddamas pieštinį japonų kryžiažodį. Kiekvienoje eilutėje stengiamasi pažymėti visus įmanomus langelius, kurie turi būti nuspalvinti ir kurie ne.

Pradžia											0
Pabaiga	.	.	.	.	.	.	.	.	.	.	0

4 pav. Sprendimo pavyzdys Nr.1

Jei prie eilutės parašyta „0“ – visus tos eilutės langelius pažymime, kaip langelius kurie nebus nuspalvinti. Taip pat jei nurodytas skaičius lygus eilutės langelių skaičiui – visus langelius nuspalviname.

Pradžia											2.3.3
Pabaiga		.			.						2.3.3

5 pav. Sprendimo pavyzdys Nr.2

Eilutę galima užpildyti jei prie jos parašytų skaičių suma ir minimalūs vieno langelio tarpai tarp nuspalvintų sričių užpildo visą eilutę.

Pradžia											2.4
		.									2.4
					.						2.4
Pabaiga											2.4

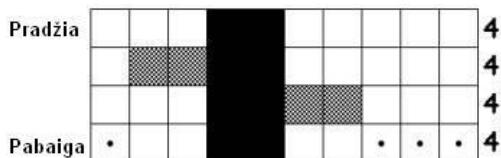
6 pav. Sprendimo pavyzdys Nr.3

Jei sritis yra didesnė už jai leidžiamą didžiausią poslinkį eilutėje, galima nuspalvinti kelis tos srities langelius.

Pradžia				.							5
				.							5
				.							5
Pabaiga	.	.	.	.							5

7 pav. Sprendimo pavyzdys Nr.4

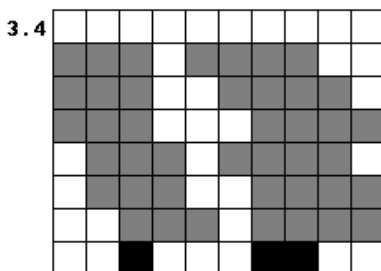
Jei yra langelių, kurie negali būti nuspalvinti, reikia nustatyti kurioje eilutės dalyje sritis telpa ir po to nustatyti kurie langeliai bus būtinai nuspalvinti. Taip pat galima pažymėti ir langelius kurie tikrai nebus nuspalvinti.



8 pav. Sprendimo pavyzdys Nr.5

Jei yra nuspalvintų langelių, kurių dalis tenka tai sričiai, kuri turi būti nuspalvinta toje eilutėje, tačiau negalima nustatyti, kurie langeliai turi būti nuspalvinti papildomai, galima pažymėti tuos langelius, kurie tikrai nebus nuspalvinti.

Loginis algoritmo vykdymo metu kiekvienai eilutei (stulpeliui) perrenkami visi įmanomi išdėstymo būdai atsižvelgiant į sąlygas ir jau užpildytus langelius (jei tokių jau yra). Baigus eilutės perrinkimą tikrinama, kurie langeliai viso perrinkimo metu išliko nepakitę ir atitinkamai užpildomi.



9 pav. Eilutės perrinkimo pavyzdys

Taip cikliškai perrenkamos visos eilutės ir stulpeliai, kol užpildomas visas kryžiažodžio laukas.

## Algoritmo spartinimas

Siekiant padidinti algoritmo efektyvumą, buvo realizuoti keli patobulinimai: eilutės perrinkimo stabdymas ir užpildytų eilučių praleidimas.

### Eilutės perrinkimo stabdymas

Perrenkant eilutę įvedamas tikrinimas, ar yra nors vienas langelis, kuris perrinkimo metu nepakito. Jei tokių langelių nebelieka, perrinkimas stabdomas. Eksperimentai parodė, kad toks apribojimas sutrumpino algoritmo vykdymo laiką iki 5%.

### Užpildytų eilučių praleidimas

Jei kryžiažodžio eilutė yra pilnai užpildyta, tokia eilutė kitose iteracijose praleidžiama. Šis apribojimas sumažino algoritmo kaštus iki 30%.

Planuojama įdiegti dar vieną patobulinimą, kuris paspartintų algoritmą, tai – potencialiai neinformatyvių eilučių praleidimas. Čia būtų iš anksto tikrinama, ar perrinkus eilutę bus užpildytas nors vienas langelis. Tikrinimo sąlyga:

$$B_M > E_d - (B_d + B_s - 1) \quad (1)$$

Čia  $B_M$  – didžiausio bloko ilgis eilutėje,

$E_d$  – eilutės ilgis,

$B_d$  – bendras blokų ilgis eilutėje,

$B_s$  – blokų skaičius eilutėje.

Ši sąlyga galioja tik tuščioms eilutėms. Eilutėms, kuriose jau yra užpildytų langelių reikėtų vykdyti perrinkimą ir jei eilutė papildytą nebūtų, pažymėti ją kaip potencialiai neinformatyvią ir ją kitose iteracijose praleisti, kol joje atsiras naujų užpildytų langelių.

## Ekspirimentų rezultatai

Dimensija	10x7	20x20	20x25	26x30	50x38	50x45
Iteracijų sk.	2	8	11	4	35	27
Laikas (s)	0,003	0,03	0,256	0,05	183,6	1315

## Išvados

1. Sukurta programa realizuojanti loginį algoritmą sprendžia pieštinius japonų kryžiažodžius.

2. Įdiegti patobulinimai leido sumažinti algoritmo vykdymo laiką. Potencialiai neinformatyvių eilučių praleidimas turėtų dar labiau jį sumažinti.

3. Eksperimentų rezultatai rodo, jog kryžiažodžio sprendimo trukmė priklauso ne vien nuo kryžiažodžio dimensijos, bet ir nuo pradinių sąlygų.

## Literatūra

Ueda, N.; Nagao, T. 1996. *NP-completeness results for NONOGRAM via Parsimonious Reductions*. Tokyo. [žiūrėta 2010 m. balandžio 15 d.].

Prieiga per internetą:

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.5277&rep=rep1&type=pdf>>

Griddlers Solver with Animator. [žiūrėta 2010 m. balandžio 15 d.]. Prieiga per internetą: <<http://jsimlo.sk/griddlers/index.php>>

Nonogram Solver. [žiūrėta 2010 m. balandžio 15 d.]. Prieiga per internetą: <<http://www.comp.lancs.ac.uk/~ss/nonogram/>>

## "JAPANESE CROSSWORD" - PICTURE SOLVING

### T. Markūnas

#### Summary

The study observes a problem of solving a "Japanese crosswords". Various problem solving methods, including brute force and heuristics, are discussed. Loginis algorithm implementation description and improvements are listed. Experimental calculations were performed and results were discussed.