

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Ignas DAPŠYS

KLASIKINĖS IR TRUPMENINIO LAIPSNIO
DIFUZIJOS-REAKCIJOS MODELIAIS
PAREMTŲ BIOLOGINIŲ JUTIKLIŲ
ATVIRKŠTINIO UŽDAVINIO SPRENDIMO
ALGORITMAI IR JŲ ANALIZĖ

DAKTARO DISERTACIJA

GAMTOS MOKSLAI,
INFORMATIKA (N 009)

Vilnius, 2023

Disertacija rengta 2019–2023 metais Vilniaus Gedimino technikos universitete.

Vadovas

prof. habil. dr. Raimondas ČIEGIS (Vilniaus Gedimino technikos universitetas, informatika – N 009).

Vilniaus Gedimino technikos universiteto Informatikos mokslo krypties disertacijos gynimo taryba:

Pirmininkas

prof. habil. dr. Antanas ČENYS (Vilniaus Gedimino technikos universitetas, informatika – N 009).

Nariai:

prof. dr. Romas BARONAS (Vilniaus universitetas, informatika – N 009),

prof. dr. Francisco Jose GASPAS LORENZ (Saragosos universitetas, Ispanija, matematika – N 001).

prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, informatikos inžinerija – T 007),

prof. habil. dr. Minvydas Kazys RAGULSKIS (Kauno technologijos universitetas, informatika – N 009).

Disertacija bus ginama viešame Informatikos mokslo krypties disertacijos gynimo tarybos posėdyje **2023 m. gruodžio 6 d. 10 val.** Vilniaus Gedimino technikos universiteto SRA-I Posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva.

Tel.: (8 5) 274 4956; faksas (8 5) 270 0112; el. paštas doktor@vilniustech.lt

Pranešimai apie numatomą ginti disertaciją išsiųsti 2023 m. lapkričio 3 d.

Disertaciją galima peržiūrėti Vilniaus Gedimino technikos universiteto talpykloje <http://dspace.vgtu.lt>, Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva), Kauno technologijos universiteto (K. Donelaičio g. 20, LT44239 Kaunas, Lietuva) ir Vytauto Didžiojo universiteto (K. Donelaičio g. 52-215 kab., LT-44244 Kaunas, Lietuva) bibliotekose.

Vilniaus Gedimino technikos universiteto 2023-044-M mokslo literatūros knyga

doi:10.20334/2023-044-M

© Vilnius Gedimino technikos universitetas, 2023

© Ignas Dapšys, 2023

ignas.dapsys@vilniustech.lt

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Ignas DAPŠYS

ANALYSIS OF ALGORITHMS FOR THE
SOLUTION OF THE INVERSE BIOSENSOR
PROBLEM BASED ON CLASSICAL AND
FRACTIONAL POWER
DIFFUSION-REACTION MODELS

DOCTORAL DISSERTATION

NATURAL SCIENCES,
INFORMATICS (N 009)

Vilnius, 2023

The doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2019–2023.

Scientific Supervisor

Prof. Dr Habil. Raimondas ČIEGIS (Vilnius Gediminas Technical University, Informatics – N 009).

The Dissertation Defense Council of the Scientific Field of Informatics of Vilnius Gediminas Technical University:

Chairperson

Prof. Dr Habil. Antanas ČENYS (Vilnius Gediminas Technical University, Informatics – N 009).

Members:

Prof. Dr Romas BARONAS (Vilnius University, Informatics – N 009),

Prof. Dr Francisco Jose GASPAS LORENZ (University of Zaragoza, Spain, Mathematics – N 001),

Prof. Dr Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Informatics Engineering – T 007),

Prof. Dr Habil. Minvydas Kazys RAGULSKIS (Kaunas University of Technology, Informatics – N 009).

The dissertation will be defended at the public meeting of the Dissertation Defence Council of the Scientific Field of Informatics in the SRA-I Meeting Hall of Vilnius Gediminas Technical University at **10 a. m. on 6 December 2023**.

Address: Saulėtekio ave. 11, LT-10223 Vilnius, Lithuania.

Tel.: +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vilniustech.lt

A notification on the intended defence of the dissertation was sent on 3 November 2023.

A copy of the doctoral dissertation is available for review at the Vilnius Gediminas Technical University repository <http://dspace.vgtu.lt>, at the Library of Vilnius Gediminas Technical University (Saulėtekio ave. 14, LT-10223 Vilnius, Lithuania), at the Library of Kaunas University of Technology (K. Donelaičio st. 20, LT-44239 Kaunas, Lithuania) and the library of Vytautas Magnus University (K. Donelaičio st. 52-215 rm., LT-44244 Kaunas, Lithuania).

Reziუმé

Biologiniai jutikliai yra prietaisai, skirti cheminėms medžiagoms aptikti ir analizuoti, remiantis biocheminiais procesais. Siekiant atlikti mėginių analizę praktikoje, reikia spręsti atvirkštinį uždavinį – iš biologinio jutiklio atsako nustatyti tiriamojo mėginio komponentų koncentracijas. Šis uždavinys yra sudėtingas dėl blogo jo sąlygotumo, su kuriuo susiduriama tiriant realų signalą, kurį veikia įvairūs triukšmai, pavyzdžiui, elektrinis triukšmas. Dėl to sumažėja nustatomų koncentracijų tikslumas. Disertacijoje nagrinėjami metodai, skirti šių sprendinių tikslumui gerinti. Kadangi biologiniai jutikliai naudojami įvairiose srityse, tokiose kaip aplinkosauga, maisto analizė ir medicina, kurios turi didelę įtaką žmogaus ir visuomenės gyvenimo kokybei, didesnis biologinių jutiklių tikslumas gali duoti didelę naudą.

Dažniausiai biologiniai jutikliai kuriami naudojant fizinius prototipus – tokių prietaisų gamyba yra brangi ir reikalauja daug laiko, todėl naudojami biologinių jutiklių matematiniai modeliai, kurie leidžia greičiau generuoti virtualius atsakus. Tačiau tokiu atveju svarbu naudoti pakankamai tikslų modelį. Dėl to nagrinėjamas biologinio jutiklio modelis, kuriame įprastinis difuzijos operatorius pakeistas trupmeninio laipsnio elipsiniu operatoriumi (TLEO). Tokiam modeliui reikalingi specializuoti skaitiniai sprendimo metodai. Disertacijoje buvo nagrinėti metodai, paremti funkcijų racionaliąja aproksimacija ir lygiagrečiosios jų versijos.

Disertaciją sudaro įvadas, trys skyriai ir rezultatų apibendrinimas. Pirmajame skyriuje pateikiami disertacijoje naudojami biologinių jutiklių matematiniai modeliai – klasikinis ir su TLEO aprašoma difuzija, apibrėžiami trupmeninio laipsnio elipsiniai operatoriai ir aptiriamas atvirkštinis biologinių jutiklių uždavinys. Antrajame skyriuje aprašomi ir analizuojami sprendimo metodai pirmojo skyriaus biologinių jutiklių modeliams, lygiagrečiosios metodų versijos ir jų efektyvumas, pateikiami eksperimentinių tikslumo ir stabilumo tyrimų rezultatai. Trečiajame skyriuje aptiriamas dirbtinių neuroninių tinklų ir lygiagrečiojo DIRECT globalaus optimizavimo metodų taikymas atvirkštiniam biologinių jutiklių uždaviniui ir eksperimentų, kuriuose taikome atkuriamų substratų leistinų koncentracijų reikšmių srities mažinimo procedūrą, rezultatai.

Disertacijos rezultatai rodo, kad racionaliąja aproksimacija paremti lygčių su TLEO sprendimo algoritmų garantuoja tikslumą, pakankamą praktiniams skaičiavimams. Sudarytos lygiagrečiosios minėtų metodų versijos leidžia efektyviai spręsti didelės apimties nelokaliuosius uždavinius. Biologinio jutiklio modelis su TLEO difuzija suteikia galimybę tiksliau jį pritaikyti realiems duomenims, o neuroninių tinklų tyrimų teoriniai ir praktiniai rezultatai leidžia pateikti rekomendacijas biologinių jutiklių tikslumui gerinti, priklausomai nuo triukšmo tipo ir kraštinių sąlygų įvedimo būdo.

Abstract

Biosensors are devices for the detection and analysis of chemical compounds, based on biochemical processes. To analyze samples in practice, the inverse biosensor problem needs to be solved – to determine component concentrations of the sample from its biosensor signal. The problem is ill-posed for multiple substrates – this property causes the biosensor to become sensitive to noise (e.g. electric noise), which is present in real devices. Due to this reason, the biosensor precision decreases. One of our objectives is to find methods to improve it. Since biosensors are used for important applications, such as environmental protection, medicine and quality control for food production, improving precision can bring clear benefits in these areas and improve the quality of life.

In this dissertation, a virtual biosensor model is used in order to avoid expenses associated with the development of physical prototypes and to obtain biosensor signals faster. This may raise questions about the model's accuracy compared to real-life devices. Therefore, an alternative model has been investigated, where the classical diffusion operator is replaced by a fractional power elliptic operator (FPEO). Solving such equations requires specialized numerical methods – methods based on rational approximations and their parallel versions were developed and analyzed. These methods were applied to a modified biosensor model.

The dissertation consists of an introduction, three main chapters and general conclusions. The first chapter describes the biosensor models used – the classical model and the one with FPEO diffusion, defines fractional power elliptic operators and discusses the inverse biosensor problem. The second chapter gives the description and analysis of solvers for the models in the first chapter, their parallel versions and experimental precision and stability results. The third chapter discusses the application of artificial neural networks and the parallel DIRECT global optimization algorithm for solving the inverse biosensor problem and the experimental results for the effect of noise and the permitted substrate concentration domain shrinkage procedure.

The results of this dissertation show that FPEO equation solvers based on rational approximation have sufficient precision for practical purposes. Parallel versions of these methods scale well for large problems. The biosensor model with FPEO-based diffusion allows for a more precise fit to real data, while the results of neural network experiments lead to recommendations to improve the biosensor precision, based on the type of noise present and the analysis mode.

Žymėjimai

Simboliai

I_{an} ir I_{mn} – adityvaus ir multiplikatyvaus triukšmo paveikti signalai;

ir a – atitinkamai fermentinio ir difuzijos sluoksnių storiai;

I – biologinio jutiklio generuojamas srovės signalas;

u – diferencialinės lygties sprendinys (ne biologinių jutiklių lygtyse);

U – diferencialinio uždavinio sprendinio aproksimacija;

$\bar{\Omega}$ – diskretusis tinklas;

$\bar{\omega}_x$ – diskretusis tinklas koordinatei x (kitoms koordinatėms žymėjimas analogiškas);

X – duomenų matrica;

T_l – duomenų, transformuotų naudojant pagrindinių komponentių analizę, matrica, čia

l – pagrindinių komponentių skaičius;

x, y ir z – erdvės koordinatės konkrečiuose uždaviniuose (bendraja forma užrašytuose uždaviniuose, naudojamas tik x , kuriuo žymimas erdvės koordinačių vektorius);

h – erdvės žingsnis;

H – Hilberto erdvė;

$\|u\| = (u, u)^{1/2}$ – L_2 norma;

τ – laiko žingsnis;

t – laikas;

Δ – Laplaso operatorius;

$E_p^{(par)}$ – lygiagrečiojo algoritmo efektyvumas, kai naudojama p procesorių;

$S_p^{(par)}$ – lygiagrečiojo algoritmo pagreitėjimas, kai naudojama p procesorių;

$T_p^{(par)}$ – lygiagrečiojo algoritmo skaičiavimo laikas, kai naudojama p procesorių;

V – lygties sprendinio aproksimacija, kai operatorius A_h^α aproksimuojamas naudojant racionaliąją funkciją;

f_{act} – neuroninio tinklo aktyvacijos funkcija;

X_{tr} ir Y_{tr} – neuroninio tinklo apmokymo duomenų rinkinio įvestys ir išvestys;

α_{tr} – neuroninio tinklo apmokymo greitis;

L – neuroninio tinklo sluoksnių skaičius;

E_{loss} – neuroninio tinklo tikslo funkcija;

s_l – neuronų skaičius l -tajame sluoksnyje;

λ_j ir ψ_j – operatoriaus A tikrinės reikšmės ir funkcijos;

μ_j ir ψ_j^h – operatoriaus A_h tikrinės reikšmės ir funkcijos;

(u, v) – skaliarinė sandauga, kai $u, v \in H$;

S_i ir P_i – substratų ir produktų koncentracijos biologinių jutiklių modeliuose;

\tilde{S} ir \tilde{P} – substratų ir produktų koncentracijų aproksimacijos biologinių jutiklių lygtyse;

A_h – tolydaus operatoriaus aproksimacija;

A – tolydus operatorius Hilberto erdvėje H ;

σ_n – triukšmo standartinis nuokrypis;

A_h^α – trupmeninio laipsnio elipsinis operatorius, $\alpha \in (0, 1) \subset \mathbb{R}$ yra jo laipsnis.

Santrumpos

1DCNN – vienmatės konvoliucijos neuroninis tinklas (angl. *1-Dimensional Convolutional Neural Network*);

AAA – adaptyvus Antoulo ir Andersono algoritmas (angl. *Adaptive Antoulas-Anderson*);

BA – pastovusis režimas (angl. *Batch Analysis*);

BRASIL – baricentrinė racionalioji aproksimacija taikant selektyvų intervalo ilgio reguliavimą (angl. *Barycentric Rational Approximation by Selective Interval Length adjustment*);

BURA – geriausia tolygioji racionalioji aproksimacija (angl. *Best Uniform Rational Approximation*);

CNN – konvoliucinis neuroninis tinklas (angl. *Convolutional Neural Network*);

CUDA – technologija bei programavimo standartas, skirti lygiagretiesiems skaičiavimams atlikti Nvidia grafiniuose procesoriuose (angl. *Compute Unified Device Architecture*);
cuDNN – dirbtinių neuroninių tinklų CUDA programavimo biblioteka (angl. *CUDA Deep Neural Networks*);
DNR – deoksiribonukleorūgštis;
FC – visiškai jungus neuroninio tinklo sluoksnius (angl. *Fully Connected*);
FIA – srauto injekcijos režimas (angl. *Flow Injection Analysis*);
GPU – vaizdo procesorius (angl. *Graphics Processing Unit*);
NCCL – Nvidia kolektyvinių komunikacijų programavimo biblioteka (angl. *Nvidia Collective Communication Library*);
PCA – pagrindinių komponentių analizė (angl. *Principal Component Analysis*);
PINN – fizika informuoti neuroniniai tinklai (angl. *Physics Informed Neural Networks*);
PK (angl. *PC*) – pagrindinės komponentės (angl. *Fractional Power Elliptic Operator*);
ReLU – dalimis tiesinė tikslo funkcija (angl. *Rectified Linear Unit*);
TLEO (angl. *FPEO*) – trupmeninio laipsnio elipsinis operatorius;
URA – tolygioji racionalioji aproksimacija (angl. *Uniform Rational Approximation*).

Turinys

IVADAS	1
Problemos formulavimas.....	1
Darbo aktualumas.....	2
Tyrimų objektas.....	2
Darbo tikslas.....	3
Darbo uždaviniai	3
Tyrimų metodika	3
Darbo mokslinis naujumas	4
Darbo rezultatų praktinė reikšmė	4
Ginamieji teiginiai	4
Darbo rezultatų aprobavimas.....	5
Disertacijos struktūra.....	5
1. BIOLOGINIŲ JUTIKLIŲ MATEMATINIŲ MODELIŲ IR JŲ SPRENDIMO METODŲ APŽVALGA.....	7
1.1. Biologinių jutiklių matematinis modelis su klasikine difuzija.....	7
1.2. Trupmeninio laipsnio difuzijos operatoriai ir jų naudojimas nестacionariuosiuose reakcijos-difuzijos modeliuose	11
1.3. Biologinių jutiklių matematinis modelis su trupmeninio laipsnio difuzijos operatoriais	15
1.4. Atvirkštinis biologinių jutiklių uždavinys ir jo sprendimo metodai	16
1.5. Pirmojo skyriaus išvados ir disertacijos uždavinių formulavimas.....	18

2. SKAITINIAI BIOLOGINIŲ JUTIKLIŲ MODELIŲ SPRENDIMO ALGORITMAI IR JŲ ANALIZĖ.....	21
2.1. Biologinių jutiklių modelių su klasikine difuzija tiesioginių uždavinių sprendimo algoritmai.....	21
2.1.1. Skaitinės schemas ir jų analizė	21
2.1.2. Lygiagrečiosios algoritmų versijos ir jų analizė	24
2.2. Skaitiniai nestacionariųjų uždavinių su trupmeninio laipsnio operatoriais sprendimo algoritmai.....	25
2.2.1. Skaitinės schemas ir jų analizė	25
2.2.2. Skaidymo metodai	38
2.2.3. Prediktoriaus-korektoriaus metodas.....	41
2.2.4. Biologinės plėvelės formavimosi uždavinys.....	41
2.2.5. Lygiagrečiosios versijos ir jų analizė.....	43
2.3. Biologinių jutiklių modelių su trupmeninio laipsnio difuzija skaitiniai sprendimo algoritmai.....	50
2.4. Antrojo skyriaus išvados	56
3. SKAITINIAI BIOLOGINIŲ JUTIKLIŲ MODELIŲ ATVIRKŠTINIŲ UŽDAVINIŲ SPRENDIMO ALGORITMAI IR JŲ ANALIZĖ	57
3.1. Dirbtinių neuroninių tinklų algoritmai ir jų analizė.....	57
3.1.1. Tiesioginio sklidimo tinklas su pagrindinių komponentų analize	61
3.1.2. Vienmatės konvoliucijos tinklas	67
3.2. Skirtingų kraštinių sąlygų režimų analizė.....	69
3.2.1. Maksimaliosios ir vidutinės paklaidų analizė. Atkuriamųjų parametru srities mažinimas.....	71
3.2.2. Rezultatai su adityviu triukšmu.....	72
3.2.3. Rezultatai su multiplikatyviu triukšmu	75
3.3. DIRECT globaliosios optimizacijos metodo taikymas ir analizė	78
3.3.1. Lygiagretusis DIRECT algoritmas.....	78
3.3.2. DIRECT taikymas biologinio jutiklio atvirkštiniam uždaviniui spręsti	79
3.4. Trečiojo skyriaus išvados	81
BENDROSIOS IŠVADOS	83
LITERATŪRA IR ŠALTINIAI.....	85
AUTORIAUS MOKSLINIŲ PUBLIKACIJŲ DISERTACIJOS TEMA SĄRAŠAS ..	93
SUMMARY IN ENGLISH.....	95

Contents

INTRODUCTION	1
Problem Formulation.....	1
Relevance of the Dissertation	2
Object of the Research.....	2
Aim of the Dissertation	3
Tasks of the Dissertation	3
Research Methodology.....	3
The Scientific Novelty of the Dissertation	4
The Practical Value of the Research Findings.....	4
Defended Statements	4
Approval of the Research Findings	5
Structure of the Dissertation.....	5
1. REVIEW OF MATHEMATICAL BIOSENSOR MODELS AND THEIR SOLUTION METHODS	7
1.1. Mathematical Biosensor Model with Classical Diffusion	7
1.2. Fractional Power Elliptic Operators and Their Use in Non-Stationary Reaction-Diffusion Models	11
1.3. Mathematical Biosensor Model with Fractional Power Elliptic Operators	15
1.4. The Inverse Biosensor Problem and Its Solution Methods.....	16
1.5. Conclusions of the First Chapter and Formulation of the Dissertation Tasks.....	18

2. NUMERICAL METHODS FOR SOLUTION OF BIOSENSOR MODELS AND THEIR ANALYSIS	21
2.1. Algorithms for Solution of Biosensor Models with Classical Diffusion.....	21
2.1.1. Numerical Schemes and Their Analysis	21
2.1.2. Parallel Algorithm Versions and Their Analysis	24
2.2. Numerical Algorithms for Solution of Non-Stationary Problems with Fractional Power Elliptic Operators	25
2.2.1. Numerical Schemes and Their Analysis	25
2.2.2. Splitting Methods.....	38
2.2.3. The Predictor-Corrector Method.....	41
2.2.4. The Problem of Biofilm Formation	41
2.2.5. Parallel Algorithm Versions and Their Analysis	43
2.3. Numerical Algorithms for the Solution of Biosensor Models With Fractional Power Diffusion.....	50
2.4. Conclusions of the Second Chapter.....	56
3. NUMERICAL METHODS FOR THE SOLUTION OF THE INVERSE BIOSENSOR PROBLEM AND THEIR ANALYSIS.....	57
3.1. Artificial Neural Networks and Their Analysis	57
3.1.1. Feedforward Neural Network with Principal Component Analysis.....	61
3.1.2. One-Dimensional Convolutional Network	67
3.2. Analysis of Different Boundary Condition Modes.....	69
3.2.1. Analysis of Maximal and Average Errors. Recovered Parameter Domain Shrinkage.	71
3.2.2. Additive Noise Results	72
3.2.3. Multiplicative Noise Results.....	75
3.3. The DIRECT Global Optimization Method and Its Analysis.....	78
3.3.1. The Parallel DIRECT Algorithm	78
3.3.2. Application of the DIRECT Algorithm for Solving the Inverse Biosensor Problem.....	79
3.4. Conclusions of the Third Chapter.....	81
GENERAL CONCLUSIONS	83
REFERENCES	85
LIST OF SCIENTIFIC PUBLICATIONS BY THE AUTHOR ON THE TOPIC OF THE DISSERTATION	93
SUMMARY IN ENGLISH.....	95

Įvadas

Problemos formulavimas

Biologiniai jutikliai yra prietaisai, skirti cheminėms medžiagoms aptikti ir analizuoti, remiantis biocheminiais procesais (Baronas et al., 2021). Biologinius jutiklius sudaro dvi dalys: bioreceptorius, kuriame vyksta biocheminės reakcijos ir elektrodas, kuris verčia cheminį signalą į elektrinį. Gaunamas signalas priklauso nuo tiriamojo mėginio cheminės sudėties. Disertacijoje nagrinėjami fermentiniai biologiniai jutikliai, o elektrodas matuoja reakcijos metu susidarančio produkto srautą. Šie biologiniai jutikliai dar turi išorinį difuzijos sluoksnį, kuriame vyksta tik substratų difuzija.

Siekiant atlikti mėginių analizę praktikoje, reikia spręsti atvirkštinių biologinio jutiklio uždavinį – iš biologinio jutiklio atsako nustatyti tiriamojo mėginio komponentų koncentracijas. Jei tiriamas mėginys su vienu komponentu, tai padaryti nesunku, žinant stacionariąją biologinio jutiklio signalo srovę. Tačiau jei mėginys yra su keliais komponentais, uždavinys tampa sudėtingas dėl blogo jo sąlygotumo (Žilinskas & Baronas, 2011), nes tada galima aptikti atsakų, kurie tam tikru tikslumu sutampa, nors juos atitinkančios koncentracijos smarkiai skiriasi. Tas pats efektas atsiranda ir tada, kai signalą veikia įvairūs triukšmai. Tai yra svarbu dėl to, kad tokie atsakai dažnai aptinkami praktikoje (Baronas et al., 2014); pavyzdžiui, juos gali veikti elektrinis triukšmas. Dėl šių priežasčių sumažėja atvirkštinio

biologinio jutiklio uždavinio sprendinių tikslumas. Mus domina metodai, skirti šių sprendinių tikslumui gerinti.

Dažniausiai biologiniai jutikliai kuriami naudojant fizinius prototipus. Pagrindinis to trūkumas yra tas, kad tokių prietaisų gamyba yra brangi ir reikalauja daug laiko. Siekiant to išvengti, naudojami biologinių jutiklių matematiniai modeliai – jie leidžia gauti atsakus žymiai greičiau nei naudojantis fiziniais prietaisais. Tačiau tokiu atveju kyla klausimas, ar šie modeliai yra pakankamai tikslūs? Dėl to nagrinėjamas modifikuotas biologinio jutiklio modelis, kuriame difuzijos operatorius pakeistas trupmeninio laipsnio difuzijos operatoriumi. Tokioms lygtims reikalingi specializuoti skaitiniai sprendimo metodai. Disertacijoje buvo nagrinėti tokie metodai, kurie paremti funkcijų racionaliąja aproksimacija ir lygiagrečiosios jų versijos trimačiams uždaviniams spręsti. Šiuos metodus pritaikėme modifikuotam biologinio jutiklio modeliui (Dapšys & Čiegis, 2023).

Darbo aktualumas

Biologiniai jutikliai naudojami įvairiose srityse: iš jų galima paminėti aplinkosaugą, maisto analizę ir mediciną. Aplinkosaugoje biologiniai jutikliai naudojami teršalams – fenoliniams junginiams (Dock et al., 2005), pesticidams (Bachmann et al., 2000; Ferentinos et al., 2013), sunkiesiems metalams aptikti (Huang et al., 2020). Šiems taikymams dažniausiai naudojami fermentiniai biologiniai jutikliai, nes jais naudojantis lengva aptikti teršalus. Analizuojant maistą, biologiniai jutikliai padeda įvertinti jo kokybę (Medina-Plaza et al., 2015), saugumą: aptikti toksiškus teršalus (Mishra et al., 2015; Nesakumar et al., 2015), padirbtą maistą (Bougadi & Kalogianni, 2020). Biomedicinoje, biologiniai jutikliai gali atlikti baltymų, esančių gyvuose organizmuose ir audiniuose, analizę (Miranda et al., 2010), aptikti patogenines bakterijas ir virusus (Barroso et al., 2018), matuoti neuromediatorius (Mazafi et al., 2018). Visi šie taikymai turi didelę įtaką žmogaus ir visuomenės gyvenimo kokybei, todėl didesnis biologinių jutiklių tikslumas gali duoti didelę naudą.

Tyrimų objektas

Disertacijos tyrimų objektas yra virtualieji biologinių jutiklių modeliai, lygiagre tieji algoritmai uždaviniams su nelokaliesiais operatoriais spręsti, atvirkštinių uždavinių sprendimo metodai, tarp jų ir dirbtiniai neuroniniai tinklai.

Darbo tikslas

Šios disertacijos tikslas – sukurti biologinio jutiklio modelį su trupmeninio laipsnio elipsiniais operatoriais (TLEO) paremta difuzija, ištirti lygiagrečiuosius metodus modelio lygtims spręsti ir sukurti neuroniniais tinklais paremtus metodus atvirkštinio biologinių jutiklių uždavinio tikslumui pagerinti.

Darbo uždaviniai

Tikslui pasiekti buvo sprendžiami tokie uždaviniai:

1. Sukurti biologinio jutiklio su TLEO difuzija modelį.
2. Sukurti nestacionariųjų diferencialinių lygčių su TLEO sprendimo algoritmus, lygiagrečiąsias jų versijas, ir išanalizuoti jų konvergavimą ir išplečiamumą.
3. Gauti biologinio jutiklio atsakus atvirkštiniam uždaviniui spręsti, naudojant tiesioginį modelį su standartine difuzija.
4. Išspręsti atvirkštinį biologinių jutiklių uždavinį, kai atsakus veikia įvairaus tipo ir lygio triukšmai.
5. Spręsti atvirkštinį biologinių jutiklių uždavinį lygiagrečiuoju DIRECT globaliosios optimizacijos metodu ir ištirti jo išplečiamumą.
6. Ištirti biologinio jutiklio modelio su TLEO atsakų priklausomybę nuo nelokaliosios difuzijos trupmeninio laipsnio parametro.

Tyrimų metodika

Biologiniai jutikliai modeliuoti naudojant nestacionariusius modelius, grindžiamus netiesinių reakcijų ir diferencialinių difuzijos lygčių sistemomis. Diferencialiniai modeliai aproksimuoti baigtinių tūrių metodu. Lygtims su TLEO spręsti taikyti (Čiegis & Dapšys, 2022; Čiegis et al., 2022; Čiegis et al., 2021) straipsniuose pateikiami metodai. Lygčių sprendikliai realizuoti C++ programavimo kalba; lygiagretieji skaičiavimai atlikti naudojantis MPI biblioteka. Atsako analizei naudoti dirbtiniai neuroniniai tinklai. Duomenų dimensijai mažinti naudota pagrindinių komponentų analizė. Neuroniniai skaičiavimai atlikti naudojantis TensorFlow (Google Brain Team, 2021) ir Keras (Keras Team, 2021) paketais, kurie realizuoti naudojant Python programavimo kalbą. Atvirkštiniam biologinių

jutiklių uždaviniui spręsti dar buvo realizuotas lygiagretusis globaliosios optimizacijos įrankis, paremtas DIRECT (Stripinis et al., 2019; Watson & Baker, 2001) metodu.

Darbo mokslinis naujumas

1. Racionaliosios aproksimacijos metodai pritaikyti nestacionariesiems ir netiesiniams paraboliniams uždaviniams su TLEO spręsti, kai negalime taikyti Furjė metodo.
2. Sukurtos ir pagrįstos lygiagrečiosios lygčių su TLEO sprendimo metodų versijos, efektyviai sprendžiančios didelės apimties uždavinius.
3. Atvirktiniam biologinių jutiklių uždaviniui spręsti panaudoti konvoliuciniai neuroniniai tinklai, parodytas tokio pasirinkimo efektyvumas.
4. Neuroninio tinklo tikslumas gerinamas naudojant atkuriamų substratų leidžiamųjų koncentracijų reikšmių srities mažinimo procedūrą.
5. Atvirktiniam biologinių jutiklių uždaviniui pritaikytas lygiagretusis DIRECT globaliosios optimizacijos metodas.

Darbo rezultatų praktinė reikšmė

Tyrimų rezultatai gali būti naudojami įvairių mokslo ir inžinerijos sričių uždavinių, kuriuose aptinkamos parabolinės ir netiesinės lygtys su TLEO, efektyviam sprendimui, taip pat ir tada, kai jie yra didelės apimties. Naujas biologinių jutiklių modelis su TLEO aprašoma difuzija gali būti naudojamas tikslesniems fiziniams prietaisams kurti. Atvirktinio uždavinio sprendimo metodų tyrimo rezultatai gali būti naudojami triukšmingų biologinių jutiklių atpažinimo tikslumui padidinti.

Ginamieji teiginiai

1. Sudaryti racionaliosios aproksimacijos, išplėtimo ir skaidymo metodai, skirti netiesinėms lygtims su TLEO spręsti; atlikus jų analizę rasta, kad jų tikslumas atitinka šioje srityje keliamus praktinius reikalavimus.

2. Sukurtų lygiagrečiųjų metodų versijų teorinė ir praktinė analizė parodė, kad jie efektyviai panaudoja procesorius sprendžiant didelės apimties uždavinius.
3. Pasiūlyta taikyti skirtingus biologinių jutiklių koncentracijų atpažinimo metodus, orientuotus į skirtingus triukšmų tipus: adityviam triukšmui, daugiasluoksnis perceptronas su PCA transformacija tinkamesnis pastoviam režimui, o konvoliucinis tinklas – srauto injekcijos režimui; multiplikatyviam triukšmui injekcijos režimas yra tikslesnis nei pastovusis.
4. Neuroninio tinklo daromą paklaidą galima sumažinti pritaikius atkuriamų substratų leidžiamųjų koncentracijų reikšmių srities mažinimo procedūrą.
5. Parengta lygiagrečioji DIRECT versija turi gerą išplečiamumą; metodas patvirtina, kad srauto injekcijos režimas tikslesnis už pastovųjį.

Darbo rezultatų apibavimas

Disertacijos tema yra atspausdinti 4 moksliniai straipsniai: visi jie publikuojami mokslo žurnaluose, indeksuojamuose *Clarivate Analytics Web of Science* duomenų bazėje. Trys iš jų patenka į Q1 kvartilį (Čiegis & Dapšys, 2022; Dapšys & Čiegis, 2023; Čiegis et al., 2021), vienas – į Q2 kvartilį (Čiegis et al., 2022). Dar vienas straipsnis (Dapšys et al., 2023) priimtas spausdinti.

Disertacijoje atliktų tyrimų rezultatai buvo paskelbti trijose mokslinėse konferencijose Lietuvoje ir užsienyje:

- MMA 2022: *Mathematical Modelling and Analysis 2022*. 2022 m. gegužės 30 – birželio 2 d., Druskininkai, Lietuva;
- NSFDE&A'22: *Numerical Solution Of Fractional Differential Equations And Applications 2022*. 2022 m. birželio 6 – 9 d., Sozopolis, Bulgarija;
- MMA 2023: *Mathematical Modelling and Analysis 2023*. 2023 m. gegužės 30 – birželio 2 d., Jūrmala, Latvija.

Disertacijos struktūra

Disertaciją sudaro įvadas, trys skyriai ir rezultatų apibendrinimas. Pirmajame skyriuje pateikiami disertacijoje naudojami biologinių jutiklių matematiniai modeliai – klasikinis ir su TLEO aprašoma difuzija, apibrėžiami trupmeninio laipsnio elipsiniai operatoriai ir aptariamas atvirkštinis biologinių jutiklių uždavinys. Antrajame skyriuje aprašomi ir analizuojami sprendimo metodai pirmojo skyriaus

biologinių jutiklių modeliams, lygiagrečiosios metodų versijos ir jų efektyvumas, pateikiami eksperimentinių tikslumo ir stabilumo tyrimų rezultatai. Trečiajame skyriuje aptariamas dirbtinių neuroninių tinklų ir lygiagrečiojo DIRECT globaliosios optimizacijos metodų taikymas atvirkštiniam biologinių jutiklių uždaviniui ir eksperimentų, kuriuose taikoma atkuriamų substratų leidžiamųjų koncentracijų reikšmių srities mažinimo procedūra, rezultatai.

Darbo apimtis yra 85 puslapiai, tekste panaudota 219 numeruotų formulių, 8 paveikslai ir 40 lentelių. Rašant disertaciją buvo panaudoti 83 literatūros šaltiniai.

Biologinių jutiklių matematinų modelių ir jų sprendimo metodų apžvalga

Skyriuje pateikiama svarbiausių biologinių jutiklių matematinų modelių apžvalga. Siekiant tiksliau aprašyti biologiniuose jutikliuose vykstančius procesus, klasikiniai difuzijos modeliai yra apibendrinti ir pasiūlyta naudoti trupmeninio laipsnio difuzijos operatorius. Taip pat apžvelgiami atvirkštinis biologinių jutiklių uždavinys ir jo sprendimo metodai. Trupmeninio laipsnio elipsinių operatorių apibrėžimas pateikiamas Čiegis et al. (2021); Čiegis & Dapšys (2022) straipsniuose. Biologinio jutiklio modelis su TLEO pateikiamas Dapšys & Čiegis (2023) straipsnyje.

1.1. Biologinių jutiklių matematinis modelis su klasikine difuzija

Biologinių jutiklių naudą atspindi literatūros gausa – *Clarivate Analytics Web of Science* duomenų bazėje rasta keli tūkstančiai straipsnių, susijusių su šia tema – tai rodo, kad ši tema yra aktuali (Cartas et al., 2010; Cui et al., 2020; Gonzalez-

Navarro et al., 2016; Zhang et al., 2021b). Dar viena tokio didelio straipsnių skaičiaus priežastis yra didelė biologinių jutiklių rūšių įvairovė – kelios iš jų pateikiamos Martynko & Kirsanov (2020) apžvalgoje, kurioje jos skirstomos pagal bioreceptorių ir detektoriaus tipą. Tai yra fermentiniai biologiniai jutikliai, kurie matuoja fermentinės reakcijos metu susidarančio produkto kiekį arba srautą (Dock et al., 2005); inhibicijos procesu paremti biologiniai jutikliai, kurie skirti fermentų inhibitoriams – medžiagoms, stabdančioms fermentines reakcijas – aptikti, matuojant tų reakcijų lėtėjimo greitį (Bachmann et al., 2000); ląsteliniai, naudojantys gyvas ląsteles (Ferentinos et al., 2013), bakterijas (Feng et al., 2013) arba vienaląsčius grybus (Salamanca-Neto et al., 2020); ir biologiniai jutikliai, naudojantys biologinės kilmės medžiagas, tokias kaip aminorūgštys (Asadollahi-Baboli & Mani-Varnosfaderani, 2014), DNR (Huang et al., 2020), antikūnai (Kim et al., 2013) ir pan. Pateikiami detektorių tipai – amperometriniai (Bachmann et al., 2000; Dock et al., 2005) (matuojamas elektros srovės stipris), potenciometriniai (Ferentinos et al., 2013) (matuojama įtampa), spektrofotometriniai (Huang et al., 2020) (matuojamas šviesos signalas, kurio spektras analizuojamas), elektrocheminės varžos spektroskopijos (Kim et al., 2013) (matuojamas varžos signalo spektras). Nors šioje disertacijoje nagrinėjami fermentiniai amperometriniai biologiniai jutikliai, šios disertacijos autoriaus tiriama metodai skirti duomenų analizei ir tinka bet kokioms biologinių jutiklių rūšims.

Tačiau didžioji dalis straipsnių duomenis gauna naudojant fizinius biologinio jutiklio prototipus – jų konstravimas reikalauja daug laiko ir išteklių. Kaip alternatyva naudojami biologinio jutiklio matematiniai modeliai (Baronas et al., 2021) – juos galime simuliuoti kompiuteriuose ir ištirti jų elgesį prieš fizinio prototipo konstravimą, tai leidžia išvengti papildomų sąnaudų.

Disertacijoje modeliuojama fermentinė reakcija (Baronas et al., 2021):



Substratai jungiasi su fermentu, sudarydami substrato-fermento kompleksą, kuris virsta produktu, o fermentas lieka nepakitęs. Darome prielaidą, kad substratai nereaguoja tarpusavyje, o kelių substratų kompleksai nesiformuoja. Tada kiekvienas substratas yra konkurencinis inhibitorius visiems kitiems substratams.

Modelį supaprastiname, taikydami kvazipastovios būsenos prielaidą. Ji teigia, kad visų substrato-fermento kompleksų koncentracija nesikeičia laike. Tada reakcijos lygtį galime pakeisti tokia:



Nagrinėjamą biologinį jutiklį sudaro trys elementai (Baronas et al., 2014): fermentinis ir išorinis difuzijos sluoksniai bei elektrodas. Fermentiniame sluoks-

nyje vyksta minėta fermentinė reakcija ir substratų bei produktų difuzija. Išoriniame difuzijos sluoksnyje reakcijos nevyksta – tik difuzijos procesas. Elektrodas produktų srautą verčia elektriniu signalu, kuris priklauso nuo tiriamojo mėginio cheminės sudėties. Darydami prielaidą, kad biologinio jutiklio geometrija yra simetriška, o fermentinis sluoksnis yra tolygaus storio ir tankio, naudojame vienmatį biologinio jutiklio modelį.

Jį galima užrašyti kaip netiesinių diferencialinių lygčių dalinėmis išvestinėmis sistema (Baronas et al., 2021). Reakcijos-difuzijos lygtys paremtos Michaelis–Menten cheminių reakcijų kinetikos ir Ficko difuzijos dėsniais. Fermentinio sluoksnio lygtys:

$$\frac{\partial S_{i,e}}{\partial t} = D_{S_{i,e}} \frac{\partial^2 S_{i,e}}{\partial x^2} - \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad (1.3)$$

$$\frac{\partial P_{i,e}}{\partial t} = D_{P_{i,e}} \frac{\partial^2 P_{i,e}}{\partial x^2} + \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad i = \overline{1, m}, \quad x \in (0, d), \quad t > 0, \quad (1.4)$$

čia $S_{i,e}$ ir $P_{i,e}$ yra atitinkamai substrato ir produkto koncentracijos fermentiniame sluoksnyje, $D_{S_{i,e}}$ ir $D_{P_{i,e}}$ – atitinkami difuzijos koeficientai, V_i – didžiausieji fermentinės reakcijos greičiai kiekvienam substratui, K_i – Michaelis–Menten konstantos, d – fermentinio sluoksnio storis.

Difuzijos sluoksnyje vyksta tik difuzijos procesas:

$$\frac{\partial S_{i,b}}{\partial t} = D_{S_{i,b}} \frac{\partial^2 S_{i,b}}{\partial x^2}, \quad (1.5)$$

$$\frac{\partial P_{i,b}}{\partial t} = D_{P_{i,b}} \frac{\partial^2 P_{i,b}}{\partial x^2}, \quad i = \overline{1, m}, \quad x \in (d, d + a), \quad t > 0, \quad (1.6)$$

čia $S_{i,b}$, $P_{i,b}$, $D_{S_{i,b}}$ ir $D_{P_{i,b}}$ yra koncentracijų reikšmės ir difuzijos koeficientai difuzijos membranoje, a – difuzijos membranos storis.

Apibrėžiame pradinės ir kraštinės modelio sąlygas (Baronas et al., 2014). Modelio pradinės sąlygos – pradžioje difuzijos ir fermentiniame sluoksniuose nėra nei substratų, nei produktų:

$$S_{i,e}(x, 0) = 0, \quad P_{i,e}(x, 0) = 0, \quad x \in [0, d], \quad (1.7)$$

$$S_{i,b}(x, 0) = 0, \quad P_{i,b}(x, 0) = 0, \quad x \in [d, d + a], \quad i = \overline{1, m}. \quad (1.8)$$

Biologinio jutiklio išorėje nurodomos pradinės substratų koncentracijos $S_{i,0}$; produktų išorėje nėra:

$$S_{i,b}(d + a, 0) = S_{i,0}, \quad P_{i,b}(d + a, 0) = 0, \quad i = \overline{1, m}. \quad (1.9)$$

Kraštinės modelio sąlygos – elektrodo paviršiuje ($x = 0$) nėra reakcijos produktų dėl elektrodo poliarizacijos, o substratai pro elektrodą neprateka:

$$P_{i,e}(0, t) = 0, D_{S_{i,e}} \frac{\partial S_{i,e}}{\partial x} \Big|_{x=0} = 0. \quad (1.10)$$

Substratų koncentracijos biologinio jutiklio išorėje priklauso nuo analizės režimo. Jei turime pastovų režimą (angl. *batch analysis*, BA), substratų koncentracija nesikeičia laike viso eksperimento metu:

$$S_{i,b}(d + a, t) = S_{i,0}. \quad (1.11)$$

Jei turime srauto injekcijos režimą (angl. *flow injection analysis* – FIA), substratai kontaktuoja su biologiniu jutikliu tik tam tikrą laiko tarpą t_f , vadinamą injekcijos laiku. Likusią eksperimento laiko dalį substratų koncentracija lygi 0:

$$S_{i,b}(d + a, t) = \begin{cases} S_{i,0}, & t \leq t_f \\ 0, & t > t_f \end{cases}. \quad (1.12)$$

Biologinio jutiklio išorėje produkto nėra:

$$P_{i,b}(d + a, t) = 0. \quad (1.13)$$

Difuzijos ir fermentinio sluoksnio sandūroje produktų ir substratų koncentracijos ir srautai sutampa:

$$D_{S_{i,e}} \frac{\partial S_{i,e}}{\partial x} \Big|_{x=d} = D_{S_{i,b}} \frac{\partial S_{i,b}}{\partial x} \Big|_{x=d}, \quad S_{i,e}(d, t) = S_{i,b}(d, t), \quad (1.14)$$

$$D_{P_{i,e}} \frac{\partial P_{i,e}}{\partial x} \Big|_{x=d} = D_{P_{i,b}} \frac{\partial P_{i,b}}{\partial x} \Big|_{x=d}, \quad P_{i,e}(d, t) = P_{i,b}(d, t), \quad i = \overline{1, m}. \quad (1.15)$$

Naudojame Nernsto išorinės difuzijos sluoksnio modelį, pagal kurį sluoksnio storis pastovus laike. Siekdami supaprastinti skaičiavimus, substratų ir produktų koncentracijas normuojame: padaliname iš atitinkamų Michaelis–Menten konstantų $\bar{S}_i = S_i/K_i, \bar{P}_i = P_i/K_i, i = \overline{1, m}$. Šios lygybės galioja visuose sluoksniuose. Tada galime įvesti pradinių koncentracijų vektorių $C \in \mathbb{R}^m$: $C_k = S_{k,0}/K_k, k = \overline{1, m}$. Atlikus normavimą, pasikeičia tik fermentinio sluoksnio lygtys:

$$\frac{\partial \bar{S}_{i,e}}{\partial t} = D_{S_{i,e}} \frac{\partial^2 \bar{S}_{i,e}}{\partial x^2} - \frac{(V_i/K_i) \bar{S}_{i,e}}{1 + \sum_{j=1}^m \bar{S}_{j,e}}, \quad (1.16)$$

$$\frac{\partial \bar{P}_{i,e}}{\partial t} = D_{P_{i,e}} \frac{\partial^2 \bar{P}_{i,e}}{\partial x^2} + \frac{(V_i/K_i) \bar{S}_{i,e}}{1 + \sum_{j=1}^m \bar{S}_{j,e}}, \quad i = \overline{1, m}, \quad x \in (0, d), \quad t > 0. \quad (1.17)$$

Biologinio jutiklio atsakas yra elektros srovė, kurios stipris priklauso nuo visų fermentinės reakcijos produktų srauto virš elektrodo paviršiaus (Baronas et al., 2014). Atsako lygtis:

$$I(t) = \sum_{i=1}^m n_i F^{(el)} D_{P_{i,e}} \left. \frac{\partial P_{i,e}}{\partial x} \right|_{x=0} = \sum_{i=1}^m K_i n_i F^{(el)} D_{P_{i,e}} \left. \frac{\partial \bar{P}_{i,e}}{\partial x} \right|_{x=0}, \quad (1.18)$$

čia n_i – elektronų, dalyvaujančių perduodant krūvį elektrodo, skaičiai kiekvienam produktui, $F^{(el)} = 96485,33 \text{ C/mol}$ – Faradėjaus konstanta. Srovės stipris, normalizuotas elektrodo plotu.

1.2. Trupmeninio laipsnio difuzijos operatoriai ir jų naudojimas nestacionariuosiuose reakcijos-difuzijos modeliuose

Naudojant biologinio jutiklio matematinius modelius, visada kyla klausimas, ar jie yra pakankamai tikslūs? Norėdami tai išsiaiškinti, turime imti daug skirtingų modelių ir jų rezultatus lyginti su tais, kuriuos duoda fiziniai prototipai. Kadangi modelių įvairovė didelė, patikrinti daugumą jų užtruktų daug laiko, todėl tikslinga pradėti nuo paprasčiausio atvejo, kai pakeičiame standartinį Ficko difuzijos operatorių. Taip yra daroma, kai susiduriame su anomalia difuzija, kuriai Ficko dėsnis nebegalioja.

Anomalios difuzijos pavyzdžių literatūroje nemažai. Baeumer et al. (2001) straipsnyje nagrinėjami teršalų difuzijos-pernešimo nehomogenišrame požeminio vandens telkinyje fizinių tyrimų duomenys. Jie rodo, kad difuzija vyksta ne pagal Ficko dėsnius: įprastai koncentracijų profiliai simetriniai, medžiagos nuo pradinės injekcijos vietos išsisklaido greičiu, proporcingu $t^{0.5}$, kraštuose nyksta eksponentiškai, tačiau tirtame vandens telkinyje pastebimas laipsninis nykimas krašte, esančiame pasroviui, nors priešingame jis išlieka eksponentiškas, todėl koncentracijų profilis nesimetrinis, išsisklaidymo greitis tiesinis pagal laiką. Šiems duomenims paaiškinti autoriai sėkmingai pritaiko modelį, kuriame difuzijos-pernešimo operatorius yra trupmeninio laipsnio – modelis atitinka eksperimentinius rezultatus su gana maža paklaida praktiniams taikymams aplinkosaugoje. Remiantis straipsnio rezultatais, laikoma, kad anomalios difuzijos procesą lygtyse be pernešimo irgi galima gerai aprašyti trupmeninio laipsnio elipsiniais operatoriais (TLEO) (Čiegis et al., 2021).

Dar viena sritis, kurioje gali būti aptinkama anomali difuzija, yra ląstelių biologija. Medžiagų difuzija ląstelėse dažniausiai tiriama fluorescencijos atsistatymo

po balinimo šviesa (angl. *fluorescence recovery after photobleaching*, FRAP) metodu. Jo esmė – į ląstelę įdedama medžiaga, pažymėta švytalu ir, naudojantis šviesa, reikiamoje ląstelės dalyje švytalas panaikinamas. Tada stebima, kaip vyksta pažymėtos medžiagos difuzija į išbalintą dalį – matuojamas fluorescuojančios šviesos stipris ir gaunamas jos profilis laike. Saxton (2001) straipsnyje tiriamas uždavinys – iš eksperimentinio fluorescencijos profilio nustatyti, ar difuzija anomali, difuzijos koeficientą ir trupmeninį difuzijos laipsnį. Straipsnyje siūlomas uždavinio sprendimo metodas, kai eksperimentinį profilį aproksimuojame teoriniais profiliais klasikinei ir anomaliai difuzijai, pastarasis gaunamas naudojantis difuzijos koeficiento aproksimacija $D = D_0 t^{\alpha-1}$, kai t pakankamai didelis. Tada nustatoma, kuris iš dviejų teorinių profilių geriausiai atitinka eksperimentinį, naudojantis statistiniu χ^2 testu. Sprendimo metodas tikrinamas naudojantis simuliuotais duomenimis, gautais Monte Karlo metodu, kai imamas didelis skaičius dalelių, judančių pagal atsitiktinį procesą, ir matuojama jų difuzija. Pagrindinis straipsnio rezultatas – daugeliu atvejų difuzijos koeficiento aproksimacija pakankamai gerai įvertina anomalous difuzijos įtaką ir parametrus.

Kituose straipsniuose nagrinėjama difuzija su kliūtimis, kurios turi sudėtingą struktūrą keleto dydžių skalėse, todėl tokiai difuzijai aprašyti taikoma difuzijos fraktaluose teorija. Matematiškai tokia difuzija aprašoma Carpinteri & Satora (2010) straipsnyje, kurio autoriai naudoja trupmeninio laipsnio išvestines difuzijos lygtys. Trupmeninis laipsnis yra lygus fraktalo, kuriame vyksta difuzija, dimensijai. Analizinių metodų tokioms lygtims spręsti yra įvairių: tarp jų galima paminėti sklindančių bangų transformaciją (Yang et al., 2018), Adomiano išskaidymą (Yang et al., 2013), pusiau skaitinį dvimatės diferencialinės transformacijos metodą (Yang et al., 2016). Lygtys su TLEO tinka ir kitokiems procesams, kuriuose atsiranda nelokalūs efektai, vykstantys ilgais nuotoliais (Pozrikidis, 2018). Todėl jos plačiai aptinkamos fizikoje (Metzler & Klafter, 2004), ekonomikoje (Sabatelli et al., 2002), medicinoje (Bueno-Orovio et al., 2014) ir kitose srityse (Lee, 2018).

Skyrelyje bus pateiktas parabolinių lygčių su TLEO apibrėžimas, diskrečioji operatoriaus aproksimacija ir suformuluota pusiau diskreti sprendimo schema (Čiegis & Dašys, 2022). Trupmeninio laipsnio elipsiniai operatoriai pasižymi tuo, kad juos apibrėžti galime keliais neekvivalenčiais būdais. Šioje disertacijoje bus taikomas spektrinis metodas (Bonito & Pasciak, 2015; Hofreither, 2020). Tarkime, H yra Hilberto erdvė su skaliarine sandauga (u, v) , kai $u, v \in H$, ir L_2 norma $\|u\| = (u, u)^{1/2}$. Tarkime, A yra šios erdvės operatorius, kuris yra simetris ir teigiamai apibrėžtas:

$$A: H \rightarrow H, \quad A = A^*, \quad A \geq cI, \quad c > 0,$$

čia I yra vienetinis operatorius. Ieškome operatoriaus A tikrinių funkcijų ψ_j ir jas atitinkančių tikrinių reikšmių λ_j :

$$A\psi_j = \lambda_j\psi_j, \quad j = 1, 2, \dots$$

Visos tikrinės reikšmės yra realios ir teigiamos:

$$0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m \leq \dots,$$

o tikrinės funkcijos sudaro ortonormuotą funkcijų bazę $\{\psi_j\}$ erdvėje H .

Tokius operatorius aptinkame klasikiniuose biologinių jutiklių modeliuose – čia difuzijos operatoriai tenkina mums reikalingas savybes, todėl galime nagrinėti apibendrintus nelokaliuosius difuzijos operatorius. Be to, biologiniuose jutikliuose taikomos realios funkcijos. Kad operatoriaus taikymo rezultatas taip pat būtų reali funkcija, šiuos nelokaliuosius operatorius galima skaičiuoti tik tada, kai matricos, kurią gauname diskretizavę diferencialinį difuzijos operatorių, visos tikrinės reikšmės yra realios ir teigiamos. Jei difuzijos operatoriai nebūtinai simetriiniai arba teigiamai apibrėžti, tokiu atveju tikrinės reikšmės gali būti kompleksiniai skaičiai. Turėdami tikrinį operatoriaus išskaidymą, bet kokią funkciją $u \in H$ galime užrašyti taip:

$$u = \sum_{j=1}^{\infty} (u, \psi_j) \psi_j.$$

Trupmeninio laipsnio elipsinį operatorių A^α su parametru $0 < \alpha < 1$ apibūdiname taip:

$$A^\alpha u = \sum_{j=1}^{\infty} \lambda_j^\alpha (u, \psi_j) \psi_j.$$

Skyrelyje sprendžiama nestacionarioji lygtis su pradinėmis sąlygomis:

$$\frac{\partial u}{\partial t} + A^\alpha u = F, \quad 0 < t \leq T, \tag{1.19}$$

$$u(0) = u_0, \quad u_0 \in H. \tag{1.20}$$

Taikant Furjė metodą, lygties sprendinį galima užrašyti $u(t) = \sum_{j=1}^{\infty} u_j(t) \psi_j$, čia

$$u_j(t) = u_{0j} e^{-\lambda_j^\alpha t} + e^{-\lambda_j^\alpha t} \int_0^t e^{\lambda_j^\alpha s} f_j(s) ds,$$

$$F(t) = \sum_{j=1}^{\infty} f_j(t) \psi_j, \quad u_0 = \sum_{j=1}^{\infty} u_{0j} \psi_j.$$

Toliau yra konstruojama pusiau diskreti schema. Erdvė H pakeičiama baigtinės dimensijos erdve H_h , o operatorius A aproksimuojamas baigtinių skirtumų

arba baigtinių tūrių metodais. Turi būti užtikrinta, kad operatoriaus aproksimacija A_h būtų simetrinis ir teigiamai apibrėžtas diskretusis operatorius:

$$A_h: H_h \rightarrow H_h, \quad A_h = A_h^*, \quad A_h \geq cI_h, \quad c > 0,$$

čia I_h yra diskretus vienetinis operatorius. Toliau sprendžiamas tikrinių reikšmių uždavinys:

$$A_h \psi_j^h = \mu_j \psi_j^h, \quad j = 1, \dots, J. \quad (1.21)$$

Visos jo tikrinės reikšmės yra teigiamos:

$$0 < \mu_1 \leq \mu_2 \leq \dots \leq \mu_J,$$

tikrinės funkcijos sudaro ortonormuotą erdvės H_h bazę ir funkcijas $U \in H_h$ galima užrašyti taip:

$$U = \sum_{j=1}^J (U, \psi_j^h) \psi_j^h.$$

Tada diskretusis trupmeninio laipsnio $0 < \alpha < 1$ operatorius A_h^α apibrėžiamas taip:

$$A_h^\alpha U = \sum_{j=1}^J \mu_j^\alpha (U, \psi_j^h) \psi_j^h.$$

Pusiau diskretus Koši uždavinys formuluojamas taip:

$$\frac{\partial U}{\partial t} + A_h^\alpha U = F, \quad 0 < t \leq T, \quad (1.22)$$

$$U(0) = U_0, \quad U_0 \in H_h. \quad (1.23)$$

Jo sprendinį galima gauti Furjė metodu $U(t) = \sum_{j=1}^J U_j(t) \psi_j^h$, čia

$$U_j(t) = U_{0j} e^{-\mu_j^\alpha t} + e^{-\mu_j^\alpha t} \int_0^t e^{\mu_j^\alpha s} F_j(s) ds,$$

$$F(t) = \sum_{j=1}^J F_j(t) \psi_j^h, \quad U_0 = \sum_{j=1}^J U_{0j} \psi_j^h.$$

Šis metodas yra tinkamas praktiniam taikymui, tačiau jis turi trūkumų: jis tinka tik stačiakampėms sritims; reikia žinoti visas operatoriaus tikrines reikšmes ir funkcijas; turi būti galioti greitosios Furjė transformacijos taikymo sąlygos.

1.3. Biologinių jutiklių matematinis modelis su trupmeninio laipsnio difuzijos operatoriais

Šiame skyrelyje apibrėžiamas biologinio jutiklio matematinis modelis su trupmeninio laipsnio elipsiniais operatoriais aprašoma difuzija. (1.3)–(1.6) modelio lygtys užrašomos taip:

$$\frac{\partial S_{i,e}}{\partial t} = -D_{S_{i,e}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha S_{i,e} - \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad (1.24)$$

$$\frac{\partial P_{i,e}}{\partial t} = -D_{P_{i,e}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha P_{i,e} + \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad x \in (0, d), \quad (1.25)$$

$$\frac{\partial S_{i,b}}{\partial t} = -D_{S_{i,b}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha S_{i,b}, \quad (1.26)$$

$$\frac{\partial P_{i,b}}{\partial t} = -D_{P_{i,b}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha P_{i,b}, \quad x \in (d, d+a), \quad i = \overline{1, m}, \quad t > 0. \quad (1.27)$$

Modelio parametrai tokie patys, kaip ir 1.1 skyrelyje, tik papildomai turime dar vieną parametą – trupmeninį laipsnį α . Jis yra svarbus dėl to, kad nuo jo priklauso gaunamas biologinio jutiklio signalas, o tai yra svarbu sprendžiant atvirkštinių biojutiklių uždavinį, nes nuo signalo priklauso atsako atpažinimo tikslumas.

Disertacijoje nagrinėjamas ir alternatyvus modelis, kai difuzijos koeficientai įtraukiami į nelokalųjį operatorių:

$$\frac{\partial S_{i,e}}{\partial t} = - \left(-D_{S_{i,e}} \frac{\partial^2}{\partial x^2} \right)^\alpha S_{i,e} - \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad (1.28)$$

$$\frac{\partial P_{i,e}}{\partial t} = - \left(-D_{P_{i,e}} \frac{\partial^2}{\partial x^2} \right)^\alpha P_{i,e} + \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad x \in (0, d), \quad (1.29)$$

$$\frac{\partial S_{i,b}}{\partial t} = - \left(-D_{S_{i,b}} \frac{\partial^2}{\partial x^2} \right)^\alpha S_{i,b}, \quad (1.30)$$

$$\frac{\partial P_{i,b}}{\partial t} = - \left(-D_{P_{i,b}} \frac{\partial^2}{\partial x^2} \right)^\alpha P_{i,b}, \quad x \in (d, d+a), \quad i = \overline{1, m}, \quad t > 0. \quad (1.31)$$

Modelis papildomas pradinėmis ir kraštinėmis sąlygomis:

$$S_{i,e}(x, 0) = 0, \quad P_{i,e}(x, 0) = 0, \quad x \in [0, d], \quad (1.32)$$

$$S_{i,b}(x, 0) = 0, \quad P_{i,b}(x, 0) = 0, \quad x \in [d, d+a), \quad i = \overline{1, m}. \quad (1.33)$$

$$(1.34)$$

$$S_{i,b}(d+a, 0) = S_{i,0}, P_{i,b}(d+a, 0) = 0, i = \overline{1, m}. \quad (1.35)$$

$$P_{i,e}(0, t) = 0, D_{S_{i,e}} \frac{\partial S_{i,e}}{\partial x} \Big|_{x=0} = 0. \quad (1.36)$$

$$\begin{aligned} S_{i,b}(d+a, t) &= S_{i,0}. \\ P_{i,b}(d+a, t) &= 0. \end{aligned} \quad (1.37)$$

Difuzijos ir fermentinio sluoksnio sandūroje naudojamos klasikinės jungimo sąlygos:

$$D_{S_{i,e}} \frac{\partial S_{i,e}}{\partial x} \Big|_{x=d} = D_{S_{i,b}} \frac{\partial S_{i,b}}{\partial x} \Big|_{x=d}, \quad S_{i,e}(d, t) = S_{i,b}(d, t), \quad (1.38)$$

$$D_{P_{i,e}} \frac{\partial P_{i,e}}{\partial x} \Big|_{x=d} = D_{P_{i,b}} \frac{\partial P_{i,b}}{\partial x} \Big|_{x=d}, \quad P_{i,e}(d, t) = P_{i,b}(d, t), i = \overline{1, m}. \quad (1.39)$$

Biologinio jutiklio atsakas apibrėžiamas taip pat, kaip ir (1.18) lygtyje.

1.4. Atvirkštinis biologinių jutiklių uždavinys ir jo sprendimo metodai

Turint pakankamai tikslų matematinį modelį, galima spręsti tiesioginį modelavimo uždavinį – užduodamos pradinės tirpalo komponentų koncentracijos, išsprendžiamas modelio lygtys ir gaunamas simuliuotas biologinio jutiklio atsakas. Tokie tiesioginiai uždaviniai yra išsamiai ištyrinėti ir yra sukurta įvairių efektyvių skaitinių metodų jiems spręsti. Tačiau praktikoje atvirkščiai žinomas biologinio jutiklio atsakas, iš kurio reikia nustatyti tiriamųjų medžiagų cheminę sudėtį. Kitaip tariant, iš simuliuoto atsako reikia nustatyti kraštines modelio sąlygas. Šis uždavinys priklauso atvirkštinių uždavinių klasei. Apibrėžkime atvirkštinį biologinių jutiklių uždavinį naudodamiesi biologinio jutiklio modeliu: iš matuoto biologinio jutiklio atsako $I_m(t)$, kurio koncentracijos nežinomos, nustatyti kraštines substratų reikšmes $C_0 = (\bar{S}_{1,0}, \dots, \bar{S}_{m,0})$.

Trupmeninio laipsnio atveju dar kyla klausimas ir dėl tinkamos trupmeninio laipsnio parametro reikšmės parinkimo. Tai galima padaryti sprendžiant papildomą atvirkštinį uždavinį, kai iš matuojamo atsako kartu su koncentracijomis nustatomas ir parametras α . Manome, kad šis uždavinys irgi yra svarbus ir įdomus. Tačiau tokiam uždaviniui spręsti reikia realių biologinių jutiklių duomenų, gaunamų atliekant fizinius eksperimentus, ir naujų sprendimo technikų, todėl šioje disertacijoje jis nenagrinėjamas. Vietoje to laikoma, kad α žinomas – tyrimuose apsiribojama keliomis fiksuotomis parametro reikšmėmis.

Standartinis būdas atvirkštiniais uždaviniais spręsti yra mažiausiųjų kvadratų metodas, kai minimizuojama norma tarp realaus matuoto ir simuliuoto atsakų, keičiant kraštines biologinio jutiklio tirpalo substratų koncentracijas (Žilinskas & Baronas, 2011):

$$C_0 = \arg \min_c \sum_{i=1}^n (I(t_i, C) - I_m(t_i))^2, \quad (1.40)$$

čia $I(t_i, C)$ yra simuliuotas biologinio jutiklio signalas, kurio kraštines substratų koncentracijos yra C . Šį uždavinį galima spręsti įvairiais optimizavimo metodais, tokiais kaip perrinkimas, gradientinis nusileidimas (Dzemyda et al., 2007), DIRECT (Jones et al., 1993; Stripinis et al., 2019) ir pan.

Šis atvirkštinis uždavinys nagrinėjamas Žilinskas & Baronas (2011) straipsnyje, kuriame naudojamas biologinio jutiklio be substratų konkurencijos modelis. Kadangi atsako funkcija $I(t_i, C)$ kiekvienai koncentracijos reikšmei yra diferencialinių lygčių sprendinys, kuris gaunamas kiekviename laiko sluoksnyje išsprendus didelės dimensijos tiesinę lygtį, tai sukelia problemų atvirkštinio uždavinio sprendimui. Tikslų funkcija (1.40) nediferencijuojama ir sunku matematiškai įrodyti jos iškilumą, todėl daroma prielaida, kad tikslo funkcija neiškili ir gali būti multimodali – turėti daugiau nei vieną minimumą. Dėl to lokaliojo optimizavimo naudojimo galimybės yra ribotos ir straipsnio autoriai siūlo uždaviniui naudoti globaliąją optimizaciją. Taip pat tikslo funkcijos reikšmių skaičiavimas reikalauja daug laiko.

Tačiau daugiausiai sunkumų kyla dėl to, kad šis uždavinys yra blogai sąlygotas. Naudojantis Hadamard'o apibrėžimu, tai reiškia, kad (1) vienai uždavinio įvesčiai yra du arba daugiau sprendinių ir (2) mažas įvesties duomenų pokytis sukelia didelį sprendinio pokytį. Biologinio jutiklio uždaviniui pirmoji savybė galioja, to požymis – sutampantys atsakai. Jei dviejų ar daugiau skirtingų atsakų skirtumų maksimumo norma mažesnė už duotą tolerancijos lygį, tada laikome, kad skirtingas koncentracijas atitinka tas pats atsakas. O antroji savybė svarbi dėl to, kad paklaidų atsiranda natūraliai – modeliuodami fizikinius procesus, negalime to padaryti tiksliai. Be to, fizinius signalus beveik visada veikia įvairūs triukšmai, todėl būtina nagrinėti triukšmo įtaką uždavinio sąlygotumui. Tai atliekama tolesniuose darbuose, naudojant dviejų konkuruojančių substratų modelį (Baronas et al., 2013) ir tą patį modelį su išoriniu difuzijos sluoksniu (Baronas et al., 2014); šiuose straipsniuose atsakų atpažinimo paklaidos papildomai tikrinamos ir kai atsakus veikia nukrypimai dėl temperatūros poveikio.

Apibrėžkime blogo sąlygotumo savybę naudojantis biologinio jutiklio modeliu: jei du biologinio jutiklio signalai I_1, I_2 yra arti vienas kito tam tikru atstumu arba tenkina $\|I_1 - I_2\|_\infty < \varepsilon$ nelygybę, tada atstumas tarp signalus atitinkančių koncentracijų $\|C_0^{(1)} - C_0^{(2)}\|_2$ gali būti didelis. Kaip minėta anksčiau, taip gali

būti kai signalai yra veikiami triukšmo. Jeigu dėl to biologinio jutiklio atsakas nežymiai iškraipomas, jį atitinkančios koncentracijos gali smarkiai pasikeisti ir gaunamos didelės atsakų atpažinimo paklaidos. Siekiant įvertinti triukšmo įtaką atsakų atpažinimo tikslumui, buvo naudotas Gauso pasiskirstymas su nuliniu vidurkiu $\mu = 0$. Naudoti du triukšmo tipai: adityvus ir multiplikatyvus; šioje disertacijoje adityvus triukšmas nepriklauso nuo signalo. Adityvaus triukšmo paveikti atsakai skaičiuoti pagal formulę:

$$I_{an}(t_i, C) = I(t_i, C) + \sigma_n X, \quad (1.41)$$

o multiplikatyvaus – pagal formulę:

$$I_{mn}(t_i, C) = I(t_i, C)(1 + \sigma_n X), \quad (1.42)$$

čia $I(t_i, C)$ – biologinio jutiklio atsako reikšmė laiko momentu t_i , kai kraštinių koncentracijų vektorius yra C . I_{an}, I_{mn} – triukšmo paveikti atsakai. σ_n – standartinis triukšmo nuokrypis, X – atsitiktinis dydis, pasiskirstęs pagal Gauso skirstinį: $X \sim N(0,1)$.

Blogo sąlygotumo įtakos mažinimui naudojama reguliarizaciją (Aster et al., 2018). Geriausiai žinomas metodas yra Tichonovo reguliarizacija (Tikhonov, 1963; Tikhonov et al., 1995), kai prie (1.40) tikslo funkcijos pridedamas papildomas narys F_{reg} , vadinamas reguliarizavimo funkcionalu – jo panaudojimas leidžia kontroliuoti sprendinių variacijas:

$$C_0 = \arg \min_C \sum_{i=1}^n (I(t_i, C) - I_m(t_i))^2 + p_{reg} F_{reg}(C), \quad (1.43)$$

čia p_{reg} – reguliarizavimo parametras, kurį reguliuojant galima pagerinti gaunamo sprendinio glodumą ir sumažinti jo jautrumą paklaidoms taip, kad originalus ir reguliarizuoto uždavinio sprendinio savybės būtų panašios. Geriausiai žinomi reguliarizavimo funkcionalo F_{reg} pavyzdžiai – L_1 ir L_2 normos: $F_{reg}(x) = \|x\|_i$, $i = 1, 2$.

1.5. Pirmojo skyriaus išvados ir disertacijos uždavinių formulavimas

Šiame skyriuje buvo apibrėžti amperometrinių biologinių jutiklių matematiniai modeliai su klasikine difuzija. Kadangi kyla klausimų dėl šio modelio tikslumo, buvo sukurtas alternatyvus modelis su trupmeninio laipsnio difuzija. TLEO galima apibrėžti keliais būdais, disertacijoje buvo naudota spektrinė formuluotė. Tokiam modeliui spręsti reikalingi specializuoti skaitiniai metodai. Norint taikyti

biologinius jutiklius praktikoje, reikia spręsti atvirkštinių biologinių jutiklių uždavinį. Šis uždavinys yra blogai sąlygotas, todėl reikia išsiaiškinti triukšmo įtaką atsakų atpažinimo kokybei. Norint spręsti atvirkštinių uždavinį su trupmeninio laipsnio difuzija, reikia papildomai ištirti atsakų priklausomybę nuo trupmeninio laipsnio parametro. Nuo jo priklauso biologinio jutiklio tikslumas.

Pagal išvadas formuluojami šie disertacijos uždaviniai:

1. Sukurti ir išanalizuoti nestacionariųjų diferencialinių lygčių su TLEO sprendimo algoritmus.
2. Gauti biologinio jutiklio atsakus atvirkštiniam uždaviniui spręsti, naudojant tiesioginį modelį su standartine difuzija.
3. Išspręsti atvirkštinių biologinių jutiklių uždavinį, kai atsakus veikia įvairaus tipo ir lygio triukšmai.
4. Ištirti biologinio jutiklio modelio su TLEO atsakų priklausomybę nuo nelokaliosios difuzijos trupmeninio laipsnio parametro.

Skaitiniai biologinių jutiklių modelių sprendimo algoritmai ir jų analizė

Antrajame skyriuje pateikiami skaitiniai biologinių jutiklių modelių sprendimo algoritmai ir jų analizė. Ji atlikta tiek teoriškai, tiek atliekant skaičiavimo eksperimentus. Taip pat aptariamos lygiagrečiosios metodų versijos, pateikiami bei analizuojami eksperimentų rezultatai. Skyriaus medžiaga publikuota Čiegis & Dapšys (2022); Čiegis et al. (2022); Dapšys & Čiegis (2023); Čiegis et al. (2021) straipsniuose ir pristatyta NSFDE&A'22 konferencijoje.

2.1. Biologinių jutiklių modelių su klasikine difuzija tiesioginių uždavinių sprendimo algoritmai

Šiame skyrelyje aprašomas skaitinis klasikinio biologinių jutiklių modelio sprendimas ir lygiagrečiųjų algoritmų taikymas.

2.1.1. Skaitinės schemas ir jų analizė

Biologinio jutiklio lygtys (1.3)–(1.6) spęstos taikant baigtinių tūrių metodą, netiesinės lygtys linearizuojamos pritaikius prediktoriaus-korektoriaus metodą ir

kiekviename laiko sluoksnyje sprendžiamos tiesinių lygčių sistemos faktorizacijos būdu. Iš pradžių apibrėžiami trys diskretieji tinklai: laiko tinklas:

$$\bar{\Omega}_t = t^n: t^0 = 0, t^n = t^{n-1} + \tau, n = 1, \dots, T/\tau,$$

erdvės tinklas fermentiniam sluoksniui:

$$\bar{\Omega}_x^e = \{x_j: x_0 = 0, x_j = x_{j-1} + h_e, j = 1, \dots, N_e\}, \quad (2.1)$$

ir erdvės tinklas difuzijos sluoksniui:

$$\bar{\Omega}_x^b = \{x_j: x_{N_e} = d, x_j = x_{j-1} + h_b, j = N_e + 1, \dots, N\}. \quad (2.2)$$

Erdvės intervalas suskaidytas į N taškų, kurie priskirti biologinio jutiklio sluoksniams proporcingai jų storiams:

$$N_e = \left\lfloor \frac{Nd}{d+a} \right\rfloor, N_b = N - N_e, \quad (2.3)$$

čia N_e yra fermentinio sluoksniu taškų skaičius, N_b – difuzijos sluoksniu taškų skaičius. Erdvės tinklų žingsniai parinkti taip: fermentiniam sluoksniui $h_e = d/N_e$, o difuzijos – $h_b = a/N_b$. Bendruoju atveju tinklas gali būti ir netolygus, tada jo žingsnius žymėsime taip:

$$h_{j-0,5} = x_j - x_{j-1}, \quad h_j = 0,5(h_{j-0,5} + h_{j+0,5}), \quad j = 1, \dots, N - 1.$$

Apibrėžiami skirtuminiai operatoriai – vienas pagal laiką ir trys pagal erdvę:

$$\begin{aligned} \partial_t u_j^n &= \frac{u_j^n - u_j^{n-1}}{\tau}, \quad \partial_x u_j^n = \frac{u_{j+1}^n - u_j^n}{h_{j+0,5}}, \\ \partial_{\bar{x}} u_j^n &= \frac{u_j^n - u_{j-1}^n}{h_{j-0,5}}, \quad \partial_x^2 u_j^n = \frac{\partial_x u_j^n - \partial_{\bar{x}} u_j^n}{h_j}, \end{aligned}$$

čia u_j^n – funkcijos u reikšmė taške $(t^n, x_j) \in \bar{\Omega}_t \times (\bar{\Omega}_x^e \cup \bar{\Omega}_x^b)$. Apibrėžiamas dar vienas operatorius – vidurkinimas pagal laiką:

$$u_j^{n-0,5} = \frac{u_j^n + u_j^{n-1}}{2}.$$

Pažymėkime sprendinių aproksimacijas: $\tilde{S}_{j,i}^n$ – i -tojo substrato aproksimacija n -tajame laiko sluoksnyje ir j -tajame erdvės tinklo taške. $\tilde{P}_{j,i}^n$ – analogiška i -tojo produkto aproksimacija. Diferencialinės modelio lygtys aproksimuojamos baigtinių tūrių metodu. Pasirenkamas baigtinis tūris $[x_{j-0,5}, x_{j+0,5}]$ ir šioje srityje integruojamos diferencialinės lygtys. Tada integralai aproksimuojami taikant vidurinių

reikšmių skaitinio integravimo formulę, funkcijų srantai srities kraštuose aproksimuojami baigtiniais skirtumais, išvestinė pagal laiką aproksimuojama centrinių skirtumų formule, gaunama tokia skirtumų schema:

$$\partial_t \tilde{S}_{j,i}^n = D_{S_{i,e}} \partial_x^2 \tilde{S}_{j,i}^{n-0,5} - \frac{(V_i/K_i) \tilde{S}_{j,i}^{n-0,5}}{1 + \sum_{i=1}^m \tilde{S}_{j,i}^{n-0,5}}, \quad (2.4)$$

$$\partial_t \tilde{P}_{j,i}^n = D_{P_{i,e}} \partial_x^2 \tilde{P}_{j,i}^{n-0,5} + \frac{(V_i/K_i) \tilde{S}_{j,i}^{n-0,5}}{1 + \sum_{i=1}^m \tilde{S}_{j,i}^{n-0,5}}, \quad i = \overline{1, m}, \quad 0 < j < N_e, \quad (2.5)$$

$$\partial_t \tilde{S}_{j,i}^n = D_{S_{i,b}} \partial_x^2 \tilde{S}_{j,i}^{n-0,5}, \quad (2.6)$$

$$\partial_t \tilde{P}_{j,i}^n = D_{P_{i,b}} \partial_x^2 \tilde{P}_{j,i}^{n-0,5}, \quad i = \overline{1, m}, \quad N_e < j < N. \quad (2.7)$$

Kadangi substratų ir produktų koncentracijos normuotos, netiesinio nario var-diklyje nelieka K_i . Apibrėžiamos pradinės ir kraštinės sąlygos:

$$\tilde{S}_{j,i}^0 = 0, \quad \tilde{P}_{j,i}^0 = 0, \quad \forall i, j, \quad (2.8)$$

$$\tilde{S}_{N,i}^n = C_i, \quad \tilde{P}_{N,i}^n = 0, \quad \tilde{P}_{0,i}^n = 0, \quad \forall i, n. \quad (2.9)$$

Kraštinės sąlygos su išvestinėmis ir jungimo sąlygos irgi aproksimuojamos baigtinių tūrių metodu, pasirenkant atitinkamus baigtinius tūrius. Prie krašto $x = 0$ imamas elementas $[0, x_{0,5}]$, o dviejų paviršių sandūroje elementas $[x_{N_e-0,5}, x_{N_e+0,5}]$:

$$D_{S_{i,e}} \partial_x \tilde{S}_{0,i}^{n-0,5} = \frac{h_e}{2} \left(\partial_t \tilde{S}_{0,i}^n + \frac{(V_i/K_i) \tilde{S}_{0,i}^{n-0,5}}{1 + \sum_{i=1}^m \tilde{S}_{0,i}^{n-0,5}} \right), \quad (2.10)$$

$$D_{S_{i,b}} \partial_x \tilde{S}_{N_e,i}^{n-0,5} - D_{S_{i,e}} \partial_x \tilde{S}_{N_e,i}^{n-0,5} = h_{N_e} \partial_t \tilde{S}_{N_e,i}^n + \frac{h_e}{2} \frac{(V_i/K_i) \tilde{S}_{N_e,i}^{n-0,5}}{1 + \sum_{i=1}^m \tilde{S}_{N_e,i}^{n-0,5}}, \quad (2.11)$$

$$D_{P_{i,b}} \partial_x \tilde{P}_{N_e,i}^{n-0,5} - D_{P_{i,e}} \partial_x \tilde{P}_{N_e,i}^{n-0,5} = h_{N_e} \partial_t \tilde{P}_{N_e,i}^n - \frac{h_e}{2} \frac{(V_i/K_i) \tilde{S}_{N_e,i}^{n-0,5}}{1 + \sum_{i=1}^m \tilde{S}_{N_e,i}^{n-0,5}}. \quad (2.12)$$

Diskrečioji schema aproksimuoja diferencialinį uždavinį tikslumu $O(\tau^2 + h^2)$.

Lytys substratams yra netiesinės, jos sprendžiamos iteraciniu algoritmu, nežinomieji yra $\tilde{S}_{j,i}^{n-0,5,k}$:

$$\frac{\tilde{S}_{j,i}^{n-0,5,k} - \tilde{S}_{j,i}^{n-1}}{0,5\tau} = D_{S_{i,e}} \partial_x^2 \tilde{S}_{j,i}^{n-0,5,k} - \frac{(V_i/K_i) \tilde{S}_{j,i}^{n-0,5,k}}{1 + \sum_{i=1}^m \tilde{S}_{j,i}^{n-0,5,k-1}},$$

$$\tilde{S}_{j,i}^{n-0,5,0} = \tilde{S}_{j,i}^{n-1}, \quad k = 1, \dots, k_{max}.$$

Kai $k_{max} = 2$, gaunamas prediktoriaus-korektoriaus algoritmas, kuris jau garantuoja $O(\tau^2)$ tikslumą.

Substratų ir produktų koncentracijų aproksimacijos apskaičiuojamos perkelties metodu sprendžiant tiesinių lygčių sistemas, kurių matricos yra trijstrižinės. Biologinio jutiklio atsakas gaunamas iš produktų sprendinių pagal (2.13) formulę, parametrai tokie patys, kaip ir (1.18) lygtyje:

$$I(t_n, C) = \sum_{i=1}^m K_i n_i F^{(el)} D_{P_{i,e}} \partial_x \tilde{P}_{0,i}^n. \quad (2.13)$$

2.1.2. Lygiagrečiosios algoritmų versijos ir jų analizė

Disertacijoje plačiai naudojami lygiagretieji algoritmai, nes tai yra bene svarbiausias metodas didelės apimties uždaviniams spręsti. Šio metodo privalumas yra tas, kad jį taikydami pagreitiname skaičiavimus ir efektyviau išnaudojame kompiuterinius išteklius. Mums svarbu, kaip įvertinti algoritmo išplečiamumą – gebėjimą spręsti augančio dydžio uždavinius. Tam matuojamas algoritmo vykdymo laikas $T_p^{(par)}$ su p procesorių ir apskaičiuojami pagreitėjimo bei efektyvumo parametrai (Čiegis, 2005). Pirmasis parametras žymimas $S_p^{(par)} = T_1^{(par)} / T_p^{(par)}$ – jis parodo, kiek kartų greičiau atliekami skaičiavimai, lyginant su nuosekliuoju algoritmu. Antrasis parametras žymimas $E_p^{(par)} = S_p^{(par)} / p$ ir jis įvertina, kiek iki galo išnaudojami užimti procesoriai. Jei pagreitėjimas yra tiesinis p atžvilgiu, arba efektyvumas artimas 1, toks algoritmas laikomas gerai išplečiamu.

Norint išspręsti atvirkštinių biologinių jutiklių uždavinį, reikia rasti funkciją, kuri atvaizduoja tiriamus signalus į juos atitinkančias koncentracijas. Disertacijoje neuroninių tinklų apmokymui sudaromi atsakų signalų ir atitinkamų koncentracijų duomenų rinkiniai. Dideliam skaičiui kraštinių koncentracijų kombinacijų sprendžiami tiesiogiai uždaviniai ir apskaičiuojami juos atitinkančių atsakų signalai. Reikia spręsti netiesinių lygčių sistemas duotajame laiko intervale, taigi skaičiavimų apimtis yra labai didelė. Tačiau visi šie uždaviniai yra nepriklausomi ir gali būti sprendžiami lygiagrečiai.

Uždavinys turi duomenų lygiagretumo savybę, nes kiekvienas procesorius gali spręsti sistemą atskirai, tik su jam skirtomis kraštinėmis sąlygomis ir jiems nereikės komunikuoti tarpusavyje. Kadangi kiekvienam procesoriui skirtų užduočių apimtis yra maždaug vienoda, jas paskirstyti galima pagal statinį tvarkaraštį. Iš pradžių sudaromas visų kraštinių koncentracijų kombinacijų sąrašas ir jas visiems procesoriams paskirstome po lygiai, tada procesoriai skaičiavimus su jomis atlieka tuo pačiu metu. Toks uždavinio sprendimo būdas efektyvus, nes komunikacijos tarp procesorių kaštai yra nedideli.

2.2. Skaitiniai nestacionariųjų uždavinių su trupmeninio laipsnio operatoriais sprendimo algoritmai

Skyrelyje aprašomi nestacionariųjų parabolinių uždavinių su trupmeninio laipsnio elipsiniais operatoriais skaitiniai sprendimo metodai, lygiagrečiosios jų versijos ir praktiniai metodų taikymo pavyzdžiai.

2.2.1. Skaitinės schemos ir jų analizė

Skaitinių schemų analizė pradėdama nuo pusiau diskretaus uždavinio (1.22) apksimavimo paklaidos įverčio skaičiavimo. Daroma prielaida, kad sprendinys yra pakankamai glodus. Paklaida užrašoma standartiniu būdu (Hundsorfer & Verwer, 2003; Samarskii, 2001):

$$\Psi_h(t) := \frac{\partial u_h}{\partial t} + A_h^\alpha u_h - F = A_h^\alpha u_h - (A^\alpha u)_h, \quad (2.14)$$

čia u_h yra diferencialinio uždavinio sprendinio projekcija į H_h . Kadangi operatorius yra trupmeninio laipsnio, naudotis Teiloro eilutėmis negalima. Tada paklaida perrašoma naudojant operatoriaus spektrine forma:

$$\Psi_h = \sum_{j=1}^J (u_{hj} \mu_j^\alpha \psi_j^h - u_j \lambda_j^\alpha \psi_j) - \sum_{j=J+1}^{\infty} \lambda_j^\alpha u_j \psi_j.$$

Antrasis narys yra $O(h^k)$ eilės priklausomai nuo sprendinio glodumo. Jei norime, kad paklaida būtų $O(h^2)$ eilės, turi galioti nelygybės:

1. $|(u_h, \psi_j^h) - (u, \psi_j)| \leq Ch^2,$
2. $\|\psi_j^h - \psi_{jh}\| \leq Ch^2,$ 3. $|\mu_j^\alpha - \lambda_j^\alpha| \leq Ch^2, \quad j = 1, \dots, J.$

Toliau pateikiama visiškai diskretaus uždavinio sprendimo schema. Apibrėžiamas diskretusis laiko tinklas:

$$\bar{\Omega}_t = \{t^n: t^n = t^{n-1} + \tau_{n-1}, \quad n = 1, \dots, N, \quad t^0 = 0, \quad t^N = T\},$$

ir aproksimuojama išvestinė pagal laiką:

$$\frac{U^{n+1} - U^n}{\tau_n} + A_h^\alpha U^{n+\sigma} = F(t^{n+\sigma}), \quad n = 0, \dots, N-1, \quad (2.15)$$

$$U^0 = U_0, \quad U_0 \in V_h,$$

čia $U^{n+\sigma}$ yra vidurkinimo operatorius,

$$U^{n+\sigma} = \sigma U^{n+1} + (1 - \sigma)U^n.$$

Kai $\sigma = 1$, gaunamas neišreikštinis Eulerio metodas, o kai $\sigma = 0,5$, gaunamas Kranko ir Nikolsono metodas. Pirmojo metodo aproksimacijos eilė pagal laiko žingsnį yra $O(\tau)$, o antrojo – $O(\tau^2)$. Jei $\sigma < 0,5$, gaunama schema yra tik sąlygiškai stabili, o jei $0,5 < \sigma < 1$, schema bus tik pirmos eilės.

Pastarąją diskrečiąją schema jau galima spręsti praktinius uždavinius, kai TLEO aproksimuojamas taikant Furjė metodą, tačiau jo taikymo galimybės ribotos: jis tinka tik tuo atveju, kai turime stačiakampę sritį ir lygties koeficientai yra pastovūs. Todėl disertacijoje siūloma aproksimuoti nelokaliosios operatorius A_h^α lokaliaisiais operatoriais, taikant racionaliosios aproksimacijos metodus.

Diskrečioji schema perrašoma tokiu pavidalu:

$$\begin{aligned} (I + \sigma\tau_n A_h^\alpha)U^{n+\sigma} &= U^n + \sigma\tau_n F(t^{n+\sigma}), \\ U^{n+\sigma} &= (I + \sigma\tau_n A_h^\alpha)^{-1}(U^n + \sigma\tau_n F(t^{n+\sigma})). \end{aligned}$$

Disertacijoje šie metodai taikomi nauju būdu, kai nelokalūs operatorius $(I + \sigma\tau_n A_h^\alpha)^{-1}$ aproksimuojamas lokaliuoju racionaliuoju operatoriumi $r_{m_a}(A_h)$, čia m_a – aproksimacijos narių skaičius. Funkcija $r_{m_a}(z)$ apibrėžiama taip:

$$r_{m_a}(z) = \frac{p_{m_a}(z)}{q_{m_a}(z)},$$

joje funkcijos p_{m_a} ir q_{m_a} yra laipsnio m_a polinomai.

Galutinė skaitinio sprendimo schema užrašoma taip:

$$V^{n+1} = r_{m_a}(A_h)(V^n + \sigma\tau F^n), n = 1, 2, \dots, N, \quad (2.16)$$

$$V^0 = U_0. \quad (2.17)$$

Aproksimacijai $r_{m_a}(z)$ gauti taikomi BURA (angl. *best uniform rational approximation*) metodai, kai maksimumo normoje aproksimuojama funkcija,

$$f(z) = \frac{1}{1 + \sigma\tau z^\alpha},$$

intervale $z \in [\mu_1, \mu_j]$. Disertacijoje nagrinėjami trys metodai. Pirmasis yra AAA (Nakatsukasa et al., 2018), kuriuo aproksimacija gaunama taip: paimama realiųjų skaičių aibė $Z = \{z_1, \dots, z_M\}$ ir r_{m_a} užrašoma baricentrine forma:

$$r_{m_a-1}(z) = \frac{\sum_{j=1}^{m_a} w_j f_j}{\sum_{j=1}^{m_a} \frac{w_j}{z - z_j}},$$

čia $f_j = f(z_j), j = 1, \dots, M$. Aproximacijos koeficientų skaičiavimas yra jautrus apvalinimo paklaidoms, todėl to daryti tiesiogiai (pvz. mažiausiųjų kvadratų metodu) negalima. Šios funkcijos koeficientų ieškoma, atliekant žingsnius $m = 1, 2, \dots$:

1. Parenkamas atramos taškas $z_m \in Z$, naudojant godžiąją paiešką. Taip aproximacijos koeficientai randami greičiau nei taikant tradicinius metodus. Kadangi uždavinys yra NP sudėtingųjų klasės, optimalus sprendinys sunkiai randamas.
2. Aproximavimo paklaida minimizuojama likusiųjų atramos taškų aibėje $Z^{(m)} = Z \setminus \{z_1, \dots, z_m\}$, keičiant svorius w_1, \dots, w_m :

$$\min \|fq_{m_a-1} - p_{m_a-1}\|_{Z^{(m_a)}}, \quad \|w\|_{m_a} = 1.$$

3. Parenkamas kitas atramos taškas $z_{m_a} \in Z^{(m_a-1)}$ taip, kad netiesinė liekana $|f(z) - r_{m_a-1}(z)|$ būtų didžiausia.
4. Ciklas sustabdomas, jei netiesinė liekana pakankamai maža.

Naudodamiesi racionaliosios aproximacijos forma dalinėmis trupmenomis,

$$\bar{r}_{m_a}(z) = \bar{c}_0 + \sum_{j=1}^{m_a} \frac{\bar{c}_j}{z - \bar{d}_j}, \tag{2.18}$$

nelokaliojo operatoriaus A_h^α aproximaciją užrašome taip:

$$A_h^\alpha \approx \bar{c}_0 I + \sum_{j=1}^{m_a} \bar{c}_j (A_h - \bar{d}_j I)^{-1}. \tag{2.19}$$

Aproximacijos paklaidai galioja įvertis (Harizanov et al., 2018):

$$\left\| A_h^\alpha - \bar{c}_0 I + \sum_{j=1}^{m_a} \bar{c}_j (A_h - \bar{d}_j I)^{-1} \right\| \leq O(e^{2\pi\sqrt{m_a}}). \tag{2.20}$$

Kadangi z yra teigiamas, o tyrimų metu nustatyta, kad $\bar{d}_j \leq 0, j = \overline{1, m_a}$, ši aproximacijos forma išvengia reikšmių skaičiavimo poliaus taško aplinkoje (tiesinių lygčių sistemų sprendimo atveju tai reiškia, kad matricos determinantas nėra nulinis ar jam artimas). Tada pirminį uždavinį galime išskaidyti į m_a mažesnių uždavinių; jie yra nepriklausomi, todėl juos galime spręsti lygiagrečiai.

AAA algoritmas buvo išbandytas sprendžiant vienmatį testinį uždavinį su difuzijos operatoriumi A_h :

$$A_h U = -\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2}, \quad x_j = jh, \quad j = 1, \dots, J-1, \quad x_J = 1, \quad (2.21)$$

ir šaltinio funkcija $F(t)$,

$$F_j(t) = \begin{cases} -1 & 0 \leq x_j \leq 0,5 \\ 1 & 0.5 < x_j \leq 1 \end{cases} \quad (2.22)$$

Parinkti uždavinio parametrai: tinkelio dydis $J = 256$, laiko intervalo galas $T = 0,6$, kai trupmeninis laipsnis $\alpha = 0,5$, ir $T = 0,8$, kai $\alpha = 0,25$. Nors testinis uždavinys yra vienmatis, jo rezultatai reprezentatyvūs ir dviejų bei trijų dimensijų uždaviniams (Hofreither, 2020), nes racionaliosios aproksimacijos tikslumas priklauso tik nuo operatoriaus A_h spektro, o šis priklauso nuo erdvės žingsnio h , bet ne nuo dimensijos. Sprendinys, gautas AAA algoritmu, \tilde{U}_j^n , lyginamas su spektriniu metodu gautu sprendiniu $U(x_j, t^n)$. Paklaida matuojama naudojant maksimumo normą:

$$e_c(N, J) = \max_{(x_j, t^n) \in \bar{\omega}_h \times \omega_t} |U(x_j, t^n) - \tilde{U}_j^n|.$$

Kadangi abu sprendiniai gauti diskretizavus erdvės sritį, turėsime erdvinę paklaidą, tačiau mūsų tikslas yra įvertinti racionaliosios aproksimacijos įtaką, todėl tikslaus analizinio sprendinio naudoti nereikia.

2.1 lentelėje pateikiami AAA algoritmo rezultatai. Racionaliosios aproksimacijos parametrai tokie: funkcijos $(1 + 0,5\tau z^\alpha)^{-1}$ mėginių skaičius $M = 25000$, kai atitinkamos z reikšmės yra iš intervalo $[\lambda_{hmin}, \lambda_{hmax}]$. Lentelėje matome, kad norint pasiekti pakankamai mažą paklaidą, reikia nedaug aproksimacijos narių: $m_\alpha = 6$, kai $\tau = 0,01; 0,005$, ir $m_\alpha = 10$, kai $\tau = 0,0025$, be to, m_α nepriklauso nuo α .

2.1 lentelė. Diskrečiosios schemos sprendinio paklaidos $e_c(N, J)$ AAA algoritmui
Table 2.1. The errors $e_c(N, J)$ of the discrete solution for the AAA algorithm

τ	$m_\alpha = 4$	$m_\alpha = 5$	$m_\alpha = 6$	$m_\alpha = 10$
		$\alpha = 0,5$	$T = 0,6$	
0,01	$2,840 \cdot 10^{-3}$	$1,004 \cdot 10^{-4}$	$7,207 \cdot 10^{-4}$	$7,208 \cdot 10^{-4}$
0,005	$4,980 \cdot 10^{-3}$	$2,641 \cdot 10^{-4}$	$2,642 \cdot 10^{-4}$	$2,642 \cdot 10^{-4}$
0,0025	$6,863 \cdot 10^{-3}$	$1,109 \cdot 10^{-4}$	$1,740 \cdot 10^{-4}$	$6,755 \cdot 10^{-5}$
		$\alpha = 0,25$	$T = 0,8$	
0,01	$1,201 \cdot 10^{-3}$	$9,488 \cdot 10^{-5}$	$5,105 \cdot 10^{-5}$	$5,151 \cdot 10^{-5}$
0,005	$1,374 \cdot 10^{-3}$	$1,399 \cdot 10^{-4}$	$1,440 \cdot 10^{-5}$	$1,284 \cdot 10^{-5}$
0,0025	$1,421 \cdot 10^{-3}$	$1,577 \cdot 10^{-4}$	$1,556 \cdot 10^{-5}$	$3,210 \cdot 10^{-6}$

Antras metodas yra BRASIL (angl. *best rational approximation by successive interval length adjustment*) (Hofreither, 2021), kurį taikant, kaip ir AAA atveju, naudojama baricentrinė racionaliosios funkcijos aproksimacija. Remiantis šiuo metodu, konstruojama BURA schema (2.16)–(2.17), kuri nagrinėjama toliau. Dabar analizuojame šios schemos stabilumą. Pradedame nuo atvejo, kai $\sigma = 1$. Pasitebėkime, kad, esant pakankamai mažoms reikšmėms $\tau\mu_1^\alpha < 1$, galioja tokia nelygybė,

$$\|(I_h + \tau A_h^\alpha)^{-1}\| \leq \frac{1}{1 + \tau\mu_1^\alpha} \leq 1 - 0,5\tau\mu_1^\alpha.$$

Tarkime, m_a yra pakankamai didelis, kad galiotų įvertis.

$$|(1 + \tau z^\alpha)^{-1} - r_{m_a}(z)| \leq 0,5\tau\mu_1^\alpha, \quad \forall z \in [\mu_1, \mu_J].$$

Iš nelygybių gaunamas reikalingas stabilumo įvertis.

$$\|r_{m_a}(A_h)\| \leq \|r_{m_a}(A_h) - (1 + \tau A_h^\alpha)^{-1}\| + \|(1 + \tau A_h^\alpha)^{-1}\| \leq 1.$$

Toliau stabilumą įvertinsime eksperimentiškai, spręsdami trimatį uždavinį (dvimatis uždavinio variantas pateikiamas Lee (2018))

$$\frac{\partial u}{\partial t} + (-\Delta)^\alpha u = F(x, y, z, t, u), \quad (x, y, z) \in \Omega, \tag{2.23}$$

$$\Omega = [0,1] \times [0,1] \times [0,1],$$

$$F(x, y, z, t, u) = t^{2\alpha} \sum_{i=1}^8 \beta_i \lambda_i^\alpha v_i + (2\alpha t^{2\alpha-1} + t^{2\alpha}) \sin^3(\pi x) \sin^3(\pi y) \sin^3(\pi z) - u,$$

$$u(x, y, z, t) = 0, \quad (x, y, z) \in \partial\Omega.$$

Jo parametrai pateikiami 2.2 lentelėje.

2.2 lentelė. Lygties šaltinio parametrai
Table 2.2. The parameters of the source function

i	λ_i	β_i	v_i
1	$3\pi^2$	$27/64$	$\sin(\pi x) \sin(\pi y) \sin(\pi z)$
2	$11\pi^2$	$-9/64$	$\sin(\pi x) \sin(\pi y) \sin(3\pi z)$
3	$11\pi^2$	$-9/64$	$\sin(\pi x) \sin(3\pi y) \sin(\pi z)$
4	$19\pi^2$	$3/64$	$\sin(\pi x) \sin(3\pi y) \sin(3\pi z)$
5	$11\pi^2$	$-9/64$	$\sin(3\pi x) \sin(\pi y) \sin(\pi z)$
6	$19\pi^2$	$3/64$	$\sin(3\pi x) \sin(\pi y) \sin(3\pi z)$
7	$19\pi^2$	$3/64$	$\sin(3\pi x) \sin(3\pi y) \sin(\pi z)$
8	$27\pi^2$	$-1/64$	$\sin(3\pi x) \sin(3\pi y) \sin(3\pi z)$

Šis uždavinys turi tikslų analizinį sprendinį:

$$u(x, y, t) = t^{2\alpha} \sin^3(\pi x) \sin^3(\pi y) \sin^3(\pi z).$$

Erdvės tinklą $\bar{\Omega}_h = \bar{\omega}_x \times \bar{\omega}_y \times \bar{\omega}_z$ apibrėžiame taip:

$$\begin{aligned} \bar{\omega}_x &= \{x_i: x_i = ih_x, i = 0, \dots, J_x, h_x = 1/J_x\}, \\ \bar{\omega}_y &= \{y_j: y_j = jh_y, j = 0, \dots, J_y, h_y = 1/J_y\}, \\ \bar{\omega}_z &= \{z_k: z_k = kh_z, k = 0, \dots, J_z, h_z = 1/J_z\}. \end{aligned}$$

Naudojame standartinį trimatį difuzijos operatorių:

$$\begin{aligned} A_h U &= - \left(\frac{U_{i+1,j,k} - 2U_{ijk} + U_{i-1,j,k}}{h_x^2} + \frac{U_{i,j+1,k} - 2U_{ijk} + U_{i,j-1,k}}{h_y^2} \right. \\ &\quad \left. + \frac{U_{i,j,k+1} - 2U_{ijk} + U_{i,j,k-1}}{h_z^2} \right) \end{aligned} \quad (2.24)$$

Operatorius A_h turi tokias tikrines reikšmes ir funkcijas (Samarskii, 2001) (jos pateikiamos, kai $(h_x = h_y = h_z = h)$):

$$\begin{aligned} \psi_{h,lmr}^h(x_i, y_j, z_k) &= 2\sqrt{2} \sin(\pi l x_i) \sin(\pi m y_j) \sin(\pi r z_k), \quad 0 < l, m, r < J_x, \\ \mu_{lmr} &= \frac{4}{h^2} \left(\sin^2\left(\frac{\pi}{2} lh\right) + \sin^2\left(\frac{\pi}{2} mh\right) + \sin^2\left(\frac{\pi}{2} rh\right) \right). \end{aligned}$$

Stabilumą įvertiname apskaičiuodami reiškinį:

$$R = \max_{0 \leq j \leq K} |r_{m_a}(z_j)|, \quad z_j = \mu_1 + \frac{j}{K}(\mu_J - \mu_1). \quad (2.25)$$

Schema bus stabili, jei $R \leq 1$. Atlikus skaitinius eksperimentus (2.23) uždaviniui ir parametrų reikšmėms $\alpha = 0,25; 0,5; 0,75$; $\tau = 10^{-k}$, $k = 1, \dots, 4$ ir erdvės tinklams $J = (J_x + 1)^3$, $J_x = 200; 400$, nustatyta, kad stabilumo sąlyga tenkinama. Racionalioji funkcija r_{m_a} skaičiuota BURA-BRASIL algoritmu, kai $m_a = 5; 7; 10; 12$.

Toliau nagrinėjamas schemas (2.16)–(2.17) tikslumas. Lyginame aproksimuotos schemas sprendinį V^n su atbulinės Eulerio schemas sprendiniu U^n . Abu užrašome spektrine forma:

$$U^n = \sum_{j=1}^J U_j^n \psi_j^h, \quad V^n = \sum_{j=1}^J V_j^n \psi_j^h,$$

čia

$$\frac{U_j^n - U_j^{n-1}}{\tau} + \mu_j^\alpha U_j^n = F_j^n, \quad j = 1, \dots, J,$$

$$V_j^n = r_{m_a}(\mu_j)(V_j^{n-1} + \tau F_j^n).$$

Antrą lygtį galima užrašyti taip:

$$\frac{V_j^n - V_j^{n-1}}{\tau} + \frac{r_{m_a}^{-1}(\mu_j) - 1}{\tau} V_j^n = F_j^n.$$

Schemos aproksimavimo paklaida apibrėžiama taip:

$$\begin{aligned} \tilde{\Psi}_\tau^n(\mu_j) &= \frac{U_j^n - U_j^{n-1}}{\tau} + \frac{r_{m_a}^{-1}(\mu_j) - 1}{\tau} U_j^n - F_j^n \\ &= \left(\frac{r_{m_a}^{-1}(\mu_j) - 1}{\tau} - \mu_j^\alpha \right) U_j^n = e_{mj} U_j^n. \end{aligned} \quad (2.26)$$

Pažymėkime paklaidos funkciją $Z_j^n = U_j^n - V_j^n$, jai galioja lygybė:

$$Z_j^n = r_{m_a}(\mu_j) \left(Z_j^{n-1} + \tau \tilde{\Psi}_\tau^n(\mu_j) \right).$$

Pritaikę ją iteratyviai, gauname tikslumo įvertį:

$$\|Z^n\| \leq t^n \|\tilde{\Psi}_\tau^n\|.$$

Tolesnė tikslumo analizė atlikta eksperimentiškai. Aproksimavimo paklaidos $\tilde{\Psi}_\tau^n(\xi) = e_{m_a}(\xi)/\xi$ maksimumo normoje pateikiamos 2.3 lentelėje. Naudotas tolygus erdvės tinklas $\bar{\Omega}_h$, kai $J_x = 200$ ir 400 ; trupmeninis laipsnis $\alpha = 0,75$; lygtis integruojama iki laiko $T = 1$.

2.3 lentelė. BURA-BRASIL schemos (2.16) aproksimavimo paklaidos $\|\tilde{\Psi}_\tau^n(\xi)\|$
Table 2.3. The approximation errors $\|\tilde{\Psi}_\tau^n(\xi)\|$ for the BURA-BRASIL scheme (2.16)

τ	$m_a = 5$	$m_a = 7$	$m_a = 10$
$J_x = 200, 29,6 \leq \xi \leq 479970$			
0,1	$5,534 \cdot 10^{-4}$	$2,317 \cdot 10^{-5}$	$1,977 \cdot 10^{-7}$
0,05	$4,632 \cdot 10^{-4}$	$1,936 \cdot 10^{-5}$	$1,652 \cdot 10^{-7}$
0,025	$3,559 \cdot 10^{-4}$	$1,484 \cdot 10^{-5}$	$1,266 \cdot 10^{-7}$
0,0125	$2,481 \cdot 10^{-4}$	$1,033 \cdot 10^{-5}$	$8,813 \cdot 10^{-8}$
$J_x = 400, 29,6 \leq \xi \leq 1919970$			
0,1	$1,552 \cdot 10^{-3}$	$9,257 \cdot 10^{-5}$	$1,274 \cdot 10^{-6}$
0,05	$1,330 \cdot 10^{-3}$	$7,876 \cdot 10^{-5}$	$1,083 \cdot 10^{-6}$
0,025	$1,056 \cdot 10^{-3}$	$6,196 \cdot 10^{-5}$	$8,513 \cdot 10^{-7}$
0,0125	$7,673 \cdot 10^{-4}$	$4,462 \cdot 10^{-5}$	$6,128 \cdot 10^{-7}$

Atlikus išsamesnę paklaidos $\tilde{\Psi}_\tau^n(\xi)$ analizę buvo aptikta, kad didžiausios paklaidos gaunamos didelėms spektrinėms modoms. Tačiau praktikoje lygčių sprendiniai priklauso tik nuo keletu vidutinio dydžio modų. Todėl skaitiniai eksperimentai buvo pakartoti, naudojant siauresnį spektrinį intervalą $[\mu_1, 20\mu_1]$. Šių eksperimentų rezultatai pateikiami 2.4 lentelėje.

2.4 lentelė. BURA-BRASIL schemos (2.16) aproksimavimo paklaidos $\|\tilde{\Psi}_\tau^n(\xi)\|$ spektriniame intervale $[\mu_1, 20\mu_1]$

Table 2.4. The approximation errors $\|\tilde{\Psi}_\tau^n(\xi)\|$ for the BURA-BRASIL scheme (2.16) in the spectral interval $[\mu_1, 20\mu_1]$

τ	$m_a = 5$	$m_a = 7$	$m_a = 10$
$J_x = 200, 29,6 \leq \xi \leq 592,2$			
0,1	$2,262 \cdot 10^{-5}$	$8,415 \cdot 10^{-7}$	$8,208 \cdot 10^{-9}$
0,05	$2,438 \cdot 10^{-5}$	$9,940 \cdot 10^{-7}$	$8,759 \cdot 10^{-9}$
0,025	$4,842 \cdot 10^{-5}$	$2,000 \cdot 10^{-6}$	$2,747 \cdot 10^{-8}$
0,0125	$1,037 \cdot 10^{-4}$	$4,299 \cdot 10^{-6}$	$3,661 \cdot 10^{-8}$
$J_x = 400, 29,6 \leq \xi \leq 592,2$			
0,1	$3,214 \cdot 10^{-5}$	$1,829 \cdot 10^{-6}$	$6,555 \cdot 10^{-7}$
0,05	$3,644 \cdot 10^{-5}$	$2,427 \cdot 10^{-6}$	$1,629 \cdot 10^{-7}$
0,025	$7,427 \cdot 10^{-5}$	$4,246 \cdot 10^{-6}$	$5,791 \cdot 10^{-8}$
0,0125	$1,645 \cdot 10^{-4}$	$9,335 \cdot 10^{-6}$	$1,247 \cdot 10^{-7}$

Matome, kad šiuo atveju tikslumas pagerėja. Papildomai buvo atlikti tikslumo eksperimentai (2.23) uždaviniui; matavome schemos aproksimavimo paklaidą maksimumo normoje:

$$E_{m_a}^{(bura)}(\tau) = \max_{(x_i, y_j, z_k) \in \Omega_h} |V_{ijk}^N - u(x_i, y_j, z_k, T)|.$$

Rezultatai pateikiami 2.5 lentelėje. Trupmeninis parametras ir erdvės tinklas tokie patys, kaip ir anksčiau.

2.5 lentelė. BURA-BRASIL schemos (2.16) aproksimavimo paklaidos $E_{m_a}^{(bura)}(\tau)$ (2.23) uždaviniui

Table 2.5. The approximation errors $E_{m_a}^{(bura)}(\tau)$ of the BURA-BRASIL scheme (2.16) for the (2.23) problem

τ	$E_5^{(bura)}(\tau)$	$E_7^{(bura)}(\tau)$	$E_{10}^{(bura)}(\tau)$
$J_x = 200$			
0,1	$2,012 \cdot 10^{-3}$	$2,011 \cdot 10^{-3}$	$2,012 \cdot 10^{-3}$
0,05	$1,028 \cdot 10^{-3}$	$1,030 \cdot 10^{-3}$	$1,031 \cdot 10^{-3}$
0,025	$5,271 \cdot 10^{-4}$	$5,472 \cdot 10^{-4}$	$5,491 \cdot 10^{-4}$
0,0125	$2,232 \cdot 10^{-4}$	$3,082 \cdot 10^{-4}$	$3,100 \cdot 10^{-4}$

2.5 lentelės pabaiga

τ	$E_5^{(bura)}(\tau)$	$E_7^{(bura)}(\tau)$	$E_{10}^{(bura)}(\tau)$
$J_x = 400$			
0,1	$1,960 \cdot 10^{-3}$	$1,956 \cdot 10^{-3}$	$1,959 \cdot 10^{-3}$
0,05	$9,711 \cdot 10^{-4}$	$9,742 \cdot 10^{-4}$	$9,771 \cdot 10^{-4}$
0,025	$4,512 \cdot 10^{-4}$	$4,923 \cdot 10^{-4}$	$4,940 \cdot 10^{-4}$
0,0125	$9,011 \cdot 10^{-5}$	$2,513 \cdot 10^{-4}$	$2,561 \cdot 10^{-4}$

Išnagrinėję globaliąsias uždavinio paklaidas matome, kad, didėjant aproksimacijos narių skaičiumi m_a , paklaidos keičiasi mažai, todėl užtenka imti nedaug m_a , šiuo atveju apie 5. Tai paaiškinti galima taip: žinome, kad globalioji paklaida susideda iš trijų – erdvinės, laikinės ir racionaliosios aproksimacijos. Kadangi racionaliosios trupmenos konverguoja labai greitai – ši paklaida mažėja eksponentiškai, pradeda dominuoti laikinės ir erdvinės paklaidos, o globalioji paklaida įsotina prie mažų m_a . Padidinus erdvės taškų skaičių, paklaidos irgi keičiasi nedaug. Sumažinus laiko žingsnį du kartus, paklaidos sumažėja irgi du kartus – taip patvirtinamas pirmos eilės konvergavimas pagal laiko žingsnį.

Kai kuriais atvejais matome, kad, m_a didinant, globalioji paklaida didėja, pavyzdžiui, kai $J_x = 400$, $\tau = 0,0125$, o m_a didėja nuo 5 iki 7. Galima tokio reiškinio priežastis – globaliosios paklaidos dalys gali būti teigiamos ir neigiamos, kai susideda dvi didelės paklaidos, bet jos yra priešingų ženklų, jos kompensuoja viena kitą ir sumoje gauname mažas paklaidas. Jei globalioji paklaida didėja, tai reiškia, kad viena jos dalis, kuri anksčiau kompensuodavo kitas, sumažėjo ir toliau nebekompensuoja.

Trečias metodas vadinamas URA (angl. *uniform rational approximation*) (Harizanov et al., 2020). Jį taikydami aproksimuojame funkciją $f(z)$, prieš tai pakeitę kintamąjį $\xi = \mu_1/z$:

$$f(z) = \frac{1}{1 + \sigma\tau z^\alpha} = \frac{\xi^\alpha}{\xi^\alpha + \sigma\tau\mu_1^\alpha}, \quad \xi \in [\mu_1/\mu_J, 1].$$

Tegul $\tilde{r}_{m_a}(\xi)$ yra funkcijos ξ^α BURA aproksimacija intervale $[0,1]$. Tada URA aproksimacija apibrėžiama taip:

$$\bar{r}_{m_a}(\xi) = \frac{\tilde{r}_{m_a}(\xi)}{\tilde{r}_{m_a}(\xi) + \sigma\tau\mu_1^\alpha}. \tag{2.27}$$

Dabar galime sudaryti diskrečiąją schemą:

$$\begin{aligned} V^n &= \bar{r}_{m_a}(\mu_1 A_h^{-1})(V^{n-1} + \sigma\tau F^n), \quad n = 1, 2, \dots, N, \\ V^0 &= U_0. \end{aligned} \tag{2.28}$$

Naudodamiesi racionaliosios funkcijos forma dalinėmis trupmenomis:

$$\bar{r}_{m_a}(\xi) = \bar{c}_0 + \sum_{j=1}^{m_a} \frac{\bar{c}_j}{\xi - \bar{d}_j}, \quad (2.29)$$

gauname m_a uždavinių, kurie yra nepriklausomi vienas nuo kito ir juos galima spręsti lygiagrečiai:

$$V^n = \bar{c}_0(V^{n-1} + \sigma\tau F^n) + \sum_{j=1}^{m_a} \bar{c}_j G_j^n,$$

čia funkcijos G_j^n yra tokių uždavinių sprendiniai,

$$(\mu_1 I_h - \bar{d}_j A_h) G_j^n = A_h(V^{n-1} + \sigma\tau F^n), \quad j = 1, \dots, m_a.$$

URA tipo schemos turi privalumą, kad funkcijos ξ^α BURA skleidinį reikia apskaičiuoti tik vieną kartą – ji tinka įvairiems difuzijos operatoriams, laiko žingsniams ir išvestinių pagal laiką aproksimavimo metodams. Tačiau šios savybės nauda mažėja atrandant naujus BURA tipo aproksimacijos algoritmus.

Išvedame URA schemos (2.27) stabilumo ir tikslumo įterčius, kai $\sigma = 1$. Schema bus stabili, jei galios nelygybė:

$$\|\bar{r}_{m_a}(\mu_1 A_h^{-1})\| \leq 1.$$

Toliau apsiribojama skaitinio kriterijaus skaičiavimais (2.23) uždaviniui:

$$R = \max_{0 \leq j \leq K} |\bar{r}_{m_a}(\mu_1 z_j^{-1})|, \quad z_j = \mu_1 + \frac{j}{K}(\mu_j - \mu_1).$$

Stabilumo sąlyga $R \leq 1$ tenkinama parametrų reikšmėms $\alpha = 0,25; 0,5; 0,75; \tau = 10^{-k}, k = 1, \dots, 4$ ir erdvės tinklams $J = (J_x + 1)^3$, $J_x = 200; 400$. Racionalioji aproksimacija skaičiuota BURA-BRASIL algoritmu, kai $m_a = 5; 7; 10; 12$ intervale $\xi \in [\mu_1/\mu_{J_x}, 1]$.

Nagrinėjame schemos tikslumą. Aproksimacijos paklaida apibrėžiama taip:

$$\begin{aligned} \bar{\Psi}^n(\mu_j) &= \frac{U_j^n - U_j^{n-1}}{\tau} + \frac{\bar{r}_{m_a}^{-1}(\mu_1/\mu_j) - 1}{\tau} U_j^n - F_j^n \\ &= \left(\frac{\bar{r}_{m_a}^{-1}(\mu_1/\mu_j) - 1}{\tau} - \mu_j^\alpha \right) U_j^n = \bar{e}_{m_j} U_j^n. \end{aligned} \quad (2.30)$$

Įrašę (2.27) į šią formulę, gauname:

$$\bar{e}_{mj} = \frac{\mu_1^\alpha}{\tilde{r}_{m_a}(\mu_1/\mu_j)} - \mu_j^\alpha.$$

Matome, kad paklaida nepriklauso nuo laiko žingsnio – ji priklauso tik nuo aproksimacijos eilės m_a .

2.6 lentelėje pateikiamos URA-BRASIL schemos aproksimavimo paklaidos maksimumo normoje:

$$\bar{E}_{m_a}(\mu) = \bar{e}_{m_a}(\mu)/\mu.$$

Eksperimentams naudoti erdvės tinklas $\bar{\Omega}_h$, kai $J_x = 200$ ir 400 , ir trupmeninis laipsnis $\alpha = 0,75$. Racionalioji funkcijos ξ^α aproksimacija skaičiuota BURA-BRASIL algoritmu intervale $\xi \in [\mu_1/\mu_{J_x}, 1]$. Matome, kad URA-BRASIL schemos tikslumas panašus į BURA-BRASIL schemos, todėl abi yra rekomenduotinos praktikoje.

2.6 lentelė. URA-BRASIL schemos (2.28) aproksimavimo paklaidos $\|\bar{E}_{m_a}\|$
Table 2.6. The approximation errors $\|\bar{E}_{m_a}\|$ of the URA-BRASIL scheme (2.28)

$m_a = 5$	$m_a = 7$	$m_a = 10$
$J_x = 200, 29,6 \leq \mu \leq 479970$		
$7,050 \cdot 10^{-4}$	$2,961 \cdot 10^{-5}$	$2,524 \cdot 10^{-7}$
$J_x = 400, 29,6 \leq \mu \leq 1919970$		
$1,914 \cdot 10^{-3}$	$1,155 \cdot 10^{-4}$	$1,587 \cdot 10^{-6}$

Toliau atliekama stabilumo ir tikslumo analizė Kranko ir Nikolsono BURA schemai (Hundsdoerfer & Verwer, 2003), kurią gauname į schemą (2.16)–(2.17) įrašę $\sigma = 0,5$.

$$V^n = r_{m_a}(A_h) \left(V^{n-1} + 0,5\tau F^{n-\frac{1}{2}} \right), n = 1, 2, \dots, N. \tag{2.31}$$

Ją perrašome patogesniu pavidalu,

$$\begin{aligned} \bar{V}^{n-\frac{1}{2}} &= r_{m_a}(A_h) \left(V^{n-1} + 0,5\tau F^{n-\frac{1}{2}} \right), \\ V^n &= 2\bar{V}^{n-\frac{1}{2}} - V^{n-1}. \end{aligned} \tag{2.32}$$

Tokia schemos forma leidžia lengviau atlikti stabilumo analizę, dėl sudėtingos originaliosios schemos stabilumo faktoriaus priklausomybės nuo didelių tikrinių reikšmių. Pagal procedūrą, turime įvertinti normą $\|2r_{m_a}(A_h) - I_h\|$. Tai vėl padarome naudodamiesi skaitiniu kriterijumi (2.23) uždaviniui:

$$R = \max_{0 \leq j \leq K} |2r_{m_a}(z_j) - 1|, \quad z_j = \mu_1 + \frac{j}{K} (\mu_J - \mu_1).$$

Stabilumo sąlyga $R \leq 1$ galioja visoms tirtoms parametų reikšmėms $\alpha = 0,25; 0,5; 0,75; \tau = 10^{-k}, k = 1, \dots, 6$ ir erdvės tinklams $J = (J_x + 1)^3$, $J_x = 200; 400$. Racionalioji aproksimacija gauta BURA-BRASIL algoritmu, kai $m_a = 3; 5; 7; 10; 12$.

Atlikti tyrimai leidžia pateikti rekomendacijas metodams parinkti. Palyginus rezultatus AAA, BURA-BRASIL ir URA-BRASIL schemoms matyti, kad jos visos turi panašų tikslumą. Kaip jau minėta, URA schema pranašesnė tuo, kad galima sutaupyti laiko, skaičiuojant jai reikalingą BURA skleidinį tik vieną kartą, tačiau ši savybė tampa vis mažiau naudinga, nes kuriami vis efektyvesni algoritmai. Praktiniams taikymams šios disertacijos autorius rekomenduoatų BRASIL metodą, nes jis yra naujesnis už AAA, todėl turėtų būti greitesnis ir tikslesnis.

Alternatyva BURA metodams yra nelokaliosios lygties Dirichlet–Neumann atvaizdavimu į singuliarią, lokaliąją elipsinę lygtį paremti metodai. Vienas iš jų yra išplėtimo metodas (Nochetto et al., 2016), kai pradinis uždavinys praplečiamas, įvedant papildomą dimensiją. Tada sritis Ω pakeičiama begaliniu cilindru $\Omega \times (0, \infty)$, kuriame sprendžiama lygtis

$$-\frac{\partial}{\partial y} \left(y^s \frac{\partial \widehat{U}^{n+\sigma}}{\partial y} \right) + y^s A_h \widehat{U}^{n+\sigma} = 0, \quad x \in \Omega, \quad y \in (0, Y), \quad (2.33)$$

$$-\lim_{y \rightarrow 0} y^s \frac{\partial \widehat{U}^{n+\sigma}}{\partial y} = d_\alpha \left(F(t^{n+\sigma}) - \frac{1}{\sigma\tau} (\widehat{U}^{n+\sigma}|_{y=0} - U^n) \right), \quad x \in \Omega,$$

$$\widehat{U}^{n+\sigma} = 0, \quad (x, y) \in (\partial\Omega \times (0, Y)) \cup (\Omega \times \{y = Y\}),$$

čia $s = 1 - 2\alpha$ ir

$$d_\alpha = 2^{1-2\alpha} \frac{\Gamma(1-\alpha)}{\Gamma(\alpha)}.$$

Kadangi naujojo uždavinio sprendinys pagal y koordinatę gęsta greitai, galima imti baigtinį intervalą. Originalaus uždavinio sprendinį gauname, kai $y = 0$:

$$\widetilde{U}^{n+\sigma}(x) = \widehat{U}^{n+\sigma}(x, 0), \quad x \in \Omega.$$

Papildomai dimensijai naudojamas netolygus tinklas su parametru $\gamma \geq 1$ – tinklas tankesnis prie srities krašto $y = 0$, tada taškų reikia mažiau (Hofreither, 2020; Nochetto et al., 2016):

$$y_j = Y \left(\frac{j}{m_a} \right)^Y, \quad j = 0, \dots, m_a. \quad (2.34)$$

Tada, pritaikę baigtinių tūrių metodą, sudarome sprendimo schemą:

$$\begin{aligned} J_{j+1/2}(\widehat{U}^{n+\sigma}) - J_{j-1/2}(\widehat{U}^{n+\sigma}) + \tilde{y}_j^s (A_h \widehat{U}^{n+\sigma})_j &= 0, \quad x \in \Omega, \\ j &= \overline{1, m_a - 1}, \\ J_{1/2}(\widehat{U}^{n+\sigma}) &= d_\alpha (F(t^{n+\sigma}) - \frac{1}{\sigma\tau} (\widehat{U}_0^{n+\sigma} - \tilde{U}^n)), \quad x \in \Omega, \\ \widehat{U}_{m_a}^{n+\sigma} &= 0, \quad x \in \Omega, \end{aligned} \quad (2.35)$$

čia srautas ir \tilde{y}^s apibrėžiami taip:

$$J_{j+1/2}(\widehat{U}^{n+\sigma}) = -y_{j+1/2}^s \frac{\widehat{U}_{j+1}^{n+\sigma} - \widehat{U}_j^{n+\sigma}}{y_{j+1} - y_j}, \quad \tilde{y}_j^s = \frac{y_{j+1/2}^{s+1} - y_{j-1/2}^{s+1}}{s+1}, \quad y_{-1/2}^{s+1} = 0.$$

Ją sprendžiame tikrinio išskaidymo metodu:

$$\begin{aligned} J_{j+1/2}(\Phi_l) - J_{j-1/2}(\Phi_l) &= \tilde{y}_j^s \eta_l \Phi_{lj}, \quad j = 1, \dots, m_a - 1, \\ J_{1/2}(\Phi_l) + d_\alpha \frac{1}{\sigma\tau} \Phi_{l0} &= \tilde{y}_0^s \eta_l \Phi_{l0}, \\ \Phi_{lm} &= 0, \end{aligned} \quad (2.36)$$

čia $\{\eta_0, \dots, \eta_{m_a}\}$ yra tikrinės reikšmės, o $\{\Phi_0, \dots, \Phi_{m_a}\}$ – jas atitinkantys tikriniai vektoriai. Lygties sprendinys užrašomas tokia forma:

$$\widehat{U}^{n+\sigma} = \sum_{l=0}^{m_a-1} W_l^{n+\sigma}(x) \Phi_l(y_j), \quad j = 0, \dots, m_a. \quad (2.37)$$

Jo koeficientai $W_l^{n+\sigma}(x)$ gaunami išsprendus m_a diskrečiųjų uždavinių:

$$(\eta_l I + A_h) W_l^{n+\sigma} = \Phi_{l0} d_\alpha \left(F(t^{n+\sigma}) + \frac{1}{\sigma\tau_n} \tilde{U}^n \right), \quad l = 0, \dots, m_a - 1,$$

o originalaus uždavinio sprendinys bus,

$$\tilde{U}^{n+\sigma} = \sum_{l=0}^{m_a-1} \Phi_{l0}^2 d_\alpha (\eta_l I + A_h)^{-1} \left(F(t^{n+\sigma}) + \frac{1}{\sigma\tau_n} \tilde{U}^n \right). \quad (2.38)$$

Jo pavidalas panašus į AAA arba BRASIL atvejus su m_a nepriklausomų uždavinių. Taigi, nors uždavinys pasidaro sudėtingesnis, jis gerai tinka lygiagretinimui.

Šio metodo tikslumą tikrinome spęsdami (2.21) uždavinį – 2.7 lentelėje matome rezultatus. Naudojome tokius parametrus (Hofreither, 2020): praplėtimo intervalo galą $Y = 3$ ir netolygaus tinklo parametą $\gamma = 12$, kai $\alpha = 0,25$, ir $\gamma = 5$, kai $\alpha = 0,5$. Šiam metodui gaunamos paklaidos panašios į AAA algoritmo paklaidas, kai $\alpha = 0,5$, tačiau $\alpha = 0,25$ atveju aproksimacijai reikia daugiau narių.

2.7 lentelė. Diskrečiosios schemos sprendinio paklaidos $e_C(N, J)$ išplėtimo metodu
Table 2.7. The errors $e_C(N, J)$ of the discrete solution for the extension method

τ	$m_a = 5$	$m_a = 10$	$m_a = 15$	$m_a = 20$
	$\alpha = 0,5$	$T = 0,6$	$\gamma = 3$	
0,01	$2,468 \cdot 10^{-3}$	$6,945 \cdot 10^{-4}$	$7,210 \cdot 10^{-4}$	$7,208 \cdot 10^{-4}$
0,005	$2,464 \cdot 10^{-3}$	$2,482 \cdot 10^{-4}$	$2,643 \cdot 10^{-4}$	$2,642 \cdot 10^{-4}$
0,0025	$2,463 \cdot 10^{-3}$	$5,991 \cdot 10^{-5}$	$6,760 \cdot 10^{-5}$	$6,756 \cdot 10^{-5}$
	$m_a = 20$	$m_a = 40$	$m_a = 80$	$m_a = 160$
	$\alpha = 0,25$	$T = 0,8$	$\gamma = 12$	
0,01	$2,999 \cdot 10^{-3}$	$7,230 \cdot 10^{-4}$	$1,846 \cdot 10^{-4}$	$5,954 \cdot 10^{-5}$
0,005	$2,894 \cdot 10^{-3}$	$7,190 \cdot 10^{-4}$	$1,805 \cdot 10^{-4}$	$4,522 \cdot 10^{-5}$
0,0025	$2,894 \cdot 10^{-3}$	$7,179 \cdot 10^{-4}$	$1,795 \cdot 10^{-4}$	$3,976 \cdot 10^{-6}$

2.2.2. Skaidymo metodai

BURA metodais paremta dar viena svarbi metodų klasė – skaidymo metodai (Harizanov et al., 2018). Iš pradžių aptarsime standartinį adityvų skaidymo metodą, paskui aprašysime jo modifikacijas. Taikydami adityvų skaidymo metodą, išskaidome operatorių A_h^α į dvi dalis: klasikinį elipsinį ir neigiamo trupmeninio laipsnio

$$\frac{U^{n+1} - U^n}{\tau_n} + A_h^{-\beta} A_h U^{n+\sigma} = F(t^{n+\sigma}), \quad (2.39)$$

čia $\beta = 1 - \alpha$. Pastaroji savybė svarbi, nes tokie operatoriai yra apręžti ir juos aproksimuojant gaunama mažesnė paklaida. Tam naudojamas AAA algoritmas, nors galima rekomenduoti ir BRASIL metodą. Operatorius $A_h^{-\beta}$ aproksimuojamas $R_{m_a}(\beta)$:

$$R_{m_a}(\beta) = c_0 I + \sum_{j=1}^{m_a} c_j (A_h - d_j I)^{-1}.$$

Tada lygtis užrašoma taip:

$$\frac{U^{n+1} - U^n}{\tau_n} + \sum_{j=1}^{m_a} B_{h,j}^{-1} A_h U^{n+\sigma} = F(t^{n+\sigma}).$$

Kadangi visi operatoriai lygtyje komutuoja, ji sprendžiama klasikiniu skaidymo algoritmu:

$$\frac{\tilde{U}^{n,j} - \tilde{U}^{n,j-1}}{\tau_n} + B_{h,j}^{-1} A_h (\sigma \tilde{U}^{n,j} + (1 - \sigma) \tilde{U}^{n,j-1}) = \frac{1}{m_a + 1} F(t^{n+\sigma}),$$

čia $j = 0, 1, \dots, m_a$. Sprendinys gaunamas atliekant žingsnius:

1. $\tilde{U}^{n,-1} = U^n$
2. $(D_j + \sigma \tau_n c_j A_h) \tilde{U}^{n,j} = (D_j - \sigma \tau_n c_j A_h) \tilde{U}^{n,j-1} + \frac{\sigma \tau_n}{m_a + 1} D_j F,$ (2.40)
 $j = \overline{0, m_a}.$
3. $\tilde{U}^{n+1} = \tilde{U}^{n, m_a},$

čia

$$D_0 = I, \quad D_j = A_h - d_j I, \quad j = 1, \dots, m_a.$$

Šio metodo lygiagrelinimo strategiją tenka keisti, nes visi skaidymo žingsniai yra atliekami nuosekliai, vienas po kito. Tada lygiagrečiai skaičiuojame vieno laiko sluoksnio realizavimo algoritmą (t. y. tiesinių lygčių sistemų sprendimo algoritmus). Skaidymo metodas turi įvairių modifikacijų, dvi naujos pateikiamos šioje disertacijoje. Pirmoji modifikacija leidžia apskaičiuoti šaltinio įtaką atskirai. Lygtis išskaidoma į stacionariąją ir homogeninę dalis:

$$A_h^\alpha W^{n+\sigma} = F(t^{n+\sigma}), \tag{2.41}$$

$$V^n = U^n - W^{n+\sigma},$$

$$\frac{V^{n+\sigma} - V^n}{\sigma \tau_n} + A_h^\alpha V^{n+\sigma} = 0, \tag{2.42}$$

$$U^{n+\sigma} = V^{n+\sigma} + W^{n+\sigma}.$$

Naudodami tokią schemą, gauname didesnę sprendinio tikslumą, tačiau uždavinio apimtis padidėja du kartus, nes sprendžiami du nelokalieji uždaviniai. Jei šaltinis stacionarusis, tai šį nelokalųjį uždavinį sprendžiame tik vieną kartą.

Skaidymo metodo ir (2.41)–(2.42) modifikacijos įtaką sprendinio tikslumui buvo nagrinėjama eksperimentiškai, sprendžiant (2.21) uždavinį. 2.8 lentelėje pateikiami rezultatai. Racionaliajai aproksimacijai naudotas AAA algoritmas. Matome, kad tikslumas yra didesnis, tai galima paaiškinti tuo, kad aproksimuojame

aprėžtą operatorių. Lentelėje dar pateikiamos modifikacijos paklaidos, kurios parodo, kad ši modifikacija padidina sprendinio tikslumą. Kadangi šaltinis stacionarus, sprendimo iteracijų skaičius išlieka toks pats.

Antroji nauja skaidymo metodo modifikacija – lygtis skaidyti pagal kitus fizikinius procesus, pavyzdžiui, taip:

$$\frac{d\tilde{U}}{dt} = F\left(t^{n+\frac{1}{2}}\right), \quad \tilde{U}(t^n) = U^n, \quad t^n < t \leq t^{n+\frac{1}{2}}, \quad (2.43)$$

$$U^{n+\frac{1}{3}} = \tilde{U}\left(t^{n+\frac{1}{2}}\right),$$

$$\frac{U^{n+\frac{2}{3}} - U^{n+\frac{1}{3}}}{\sigma\tau_n} + A_h^\alpha \frac{U^{n+\frac{2}{3}} + U^{n+\frac{1}{3}}}{2} = 0, \quad (2.44)$$

$$\frac{d\tilde{U}}{dt} = F\left(t^{n+\frac{1}{2}}\right), \quad \tilde{U}\left(t^{n+\frac{1}{2}}\right) = U^{n+\frac{2}{3}}, \quad t^{n+\frac{1}{2}} < t \leq t^{n+1},$$

$$U^{n+1} = \tilde{U}(t^{n+1}). \quad (2.45)$$

Šis metodas turi du privalumus: (1) užtenka spręsti tik vieną nelokalųjį uždavinį ir (2) netiesinius uždavinius, kuriuos galime spręsti efektyviai, pasitelkdami paprastųjų diferencialinių lygčių sprendimo metodus, pritaikytus konkrečiam netiesiškumo tipui.

2.8 lentelė. Diskrečiosios schemos sprendinio paklaidos $e_c(N, J)$ modifikuotam skaidymo metodui

Table 2.8. The errors $e_c(N, J)$ of the discrete solution for the modified splitting method

τ	$m_a = 6$	$m_a = 10$	$m_a = 15$
	$\alpha = 0,5$	$T = 0,6$	
0,01	$3,746 \cdot 10^{-5}$	$9,103 \cdot 10^{-6}$	$4,760 \cdot 10^{-6}$
0,005	$1,993 \cdot 10^{-5}$	$7,559 \cdot 10^{-6}$	$4,079 \cdot 10^{-6}$
0,0025	$5,870 \cdot 10^{-6}$	$2,660 \cdot 10^{-6}$	$1,465 \cdot 10^{-6}$
Stat.	$4,586 \cdot 10^{-6}$	$6,187 \cdot 10^{-8}$	$1,149 \cdot 10^{-11}$
	$m_a = 6$	$m_a = 10$	$m_a = 15$
	$\alpha = 0,25$	$T = 0,8$	
0,01	$2,637 \cdot 10^{-5}$	$1,103 \cdot 10^{-6}$	$8,124 \cdot 10^{-7}$
0,005	$2,637 \cdot 10^{-5}$	$2,762 \cdot 10^{-7}$	$2,072 \cdot 10^{-7}$
0,0025	$2,517 \cdot 10^{-5}$	$8,281 \cdot 10^{-8}$	$5,599 \cdot 10^{-8}$
Stat.	$2,428 \cdot 10^{-5}$	$2,520 \cdot 10^{-8}$	$8,445 \cdot 10^{-9}$

2.2.3. Prediktorius-korektorius metodas

Nagrinėkime lygtį, kurios šaltinis priklauso nuo sprendinio ir erdvės taško:

$$\begin{aligned} \frac{dU}{dt} + A_h^\alpha U &= F(x, t, U), \quad 0 < t \leq T, \quad x \in \Omega, \\ U(0) &= U_0, \quad U_0 \in V_h, \end{aligned} \quad (2.46)$$

čia F yra reakcijos narys. Ši lygtis bendruoju atveju gali būti netiesinė. Lygties sprendimo schema:

$$\frac{U^{n+\sigma} - U^n}{\sigma\tau_n} + A_h^\alpha U^{n+\sigma} = F(x, t^{n+\sigma}, U^{n+\sigma}). \quad (2.47)$$

Visi anksčiau aptarti metodai tinka ir tokio tipo lygtims – tam reikia lygtis linearizuoti. Disertacijoje tai atliekama naudojant prediktorius-korektorius algoritmą:

$$\frac{U^{n+\sigma,k} - U^n}{\sigma\tau_n} + A_h^\alpha U^{n+\sigma,k} = F(x, t^{n+\sigma}, U^{n+\sigma,k-1}), \quad k = 1, \dots, K, \quad (2.48)$$

čia $U^{n+\sigma,k} = \sigma U^{n+1,k} + (1 - \sigma)U^n$. Jei taikome šį algoritmą, uždavinio apimtis padidėja K kartų. Kai $K = 2$ ir $\sigma = 0,5$, metodas garantuoja antros eilės tikslumą pagal laiką. Jei K didintume toliau, nors netiesinio šaltinio skaičiavimo tikslumas didėtų kartu su uždavinio apimtimi, visos schemos tikslumo eilė išliktų antra, todėl K toliau didinti neverta.

2.2.4. Biologinės plėvelės formavimosi uždavinys

Bendros analizės procedūros iš Čiegis & Dapšys (2022) straipsnio nauda demonstruojama sprendžiant praktinį uždavinį – dvimatį biologinės plėvelės formavimosi modelį, paremtą Allen–Cahn lygtimi (Jornet, 2021; Tijani & Appadu, 2022):

$$\begin{aligned} \frac{\partial u}{\partial t} + D(-\Delta)^\alpha &= ru(1-u)(u-\gamma), \\ (x, y) \in \Omega &= (0,1) \times (0,1), \quad t \in (0, T], \end{aligned} \quad (2.49)$$

čia D – difuzijos koeficientas, r – populiacijos augimo koeficientas, γ – populiacijos išnykimo slenkstis. Pradinės sąlygos apibrėžiamos taip:

$$u(x, y, 0) = u_0(x, y), \quad (x, y) \in [0,1] \times [0,1].$$

Naudojamos homogeninės Dirichlet kraštinės sąlygos sritys Ω krašte:

$$u(x, y, t) = 0, \quad (x, y) \in \partial\Omega, \quad t \in (0, T].$$

Disertacijoje siekiama rasti antros eilės tikslumo sprendimo schemą, kuria galima spręsti bendrą tokio tipo uždavinį su nelokalija difuzija. Lokaliojo difuzijos operatoriaus diskretizacija vykdoma taip pat, kaip ir anksčiau – baigtinių tūrių metodu, paėmus diskretųjį erdvės tinklą $\omega_x \times \omega_y$.

Sudaroma simetrinė skaidymo schema:

$$\frac{d\tilde{U}}{dt} = F\left(X, t^{n+\frac{1}{2}}, \tilde{U}\right), \quad \tilde{U}(t^n) = U^n, \quad t^n < t \leq t^{n+\frac{1}{2}}, \quad (2.50)$$

$$U^{n+\frac{1}{3}} = \tilde{U}\left(t^{n+\frac{1}{2}}\right),$$

$$\frac{U^{n+\frac{2}{3}} - U^{n+\frac{1}{3}}}{\tau} + A_h^\alpha \frac{U^{n+\frac{2}{3}} + U^{n+\frac{1}{3}}}{2} = 0, \quad (2.51)$$

$$\frac{d\tilde{U}}{dt} = F\left(X, t^{n+\frac{1}{2}}, \tilde{U}\right), \quad \tilde{U}\left(t^{n+\frac{1}{2}}\right) = U^{n+\frac{2}{3}}, \quad t^{n+\frac{1}{2}} < t \leq t^{n+1},$$

$$U^{n+1} = \tilde{U}(t^{n+1}). \quad (2.52)$$

Ji turi du privalumus; pirmas – yra efektyvių diskretizacijos laike metodų, pritaikytų konkrečioms netiesinėms funkcijoms. Disertacijoje taikomas prediktoriaus-korektoriaus metodas:

$$U^{n+\frac{1}{3},0} = U^n,$$

$$U^{n+\frac{1}{3},s+1} = U^n + \frac{\tau}{4} r U^{n+\frac{1}{3},s} \left(1 - U^{n+\frac{1}{3},s}\right) \left(U^{n+\frac{1}{3},s} - \gamma\right), \quad s = 0,1,$$

$$U^{n+\frac{1}{3}} = 2U^{n+\frac{1}{3},2} - U^n. \quad (2.53)$$

Antras privalumas – reikia spręsti tik vieną nelokalųjį uždavinį kiekvienu laiko momentu. Šie uždaviniai sprendžiami Kranko ir Nikolsono metodu su BURA-BRASIL. Naudojantis analizės procedūra, nebereikia analizuoti stabilumo ir tikslumo – tą šiai schemai atlikome anksčiau. Schema turi antros eilės tikslumą pagal laiką ir pagal erdvę.

Schemas efektyvumas iliustruojamas praktiniu pavyzdžiu, kai sprendžiamas biologinės plėvelės uždavinys (Chen & Kojouharov, 1999; Tijani & Appadu, 2022) esant tokiai pradinei sąlygai:

$$u(x, y, 0) = \sum_{p=1}^2 C_p e^{-w_p \|X - X_p\|^2},$$

ir parametrams $X = (x, y)$, $C_1 = 0,68$, $C_2 = 0,65$, $w_1 = 60$, $w_2 = 100$, $X_1 = (0,35; 0,60)$, $X_2 = (0,48; 0,45)$. Lygties koeficientai $r = 200$, $\gamma = 0,05$ ir

$D = 0,0002$. 2.9 lentelėje skaidymo schemas (SS) konvergavimo greitis lyginamas su kitų dviejų schemų iš (Tijani & Appadu, 2022) straipsnio – nestandartinės baigtinių skirtumų (NSFD) ir išreikštinės Eulerio (EE) schemų greičiais. Lygtys spęstos iki laiko $t = 0,01$, laipsnis $\alpha = 1$, erdvės žingsnis $h = 0,02$. Matome, kad skaidymo schema turi antros eilės tikslumą ir yra tikslesnė už kitus taikytus metodus. Be to, jei trupmeninis laipsnis $0,2 \leq \alpha \leq 0,75$, užtenka imti $m_a \leq 5$ aproksimacijos narių BURA-BRASIL algoritmui, kad gautume tokį patį tikslumą, kaip ir visiškai diskretaus nelokaliojo uždavinio.

2.9 lentelė. Skaidymo schemas (2.50)–(2.52), NSFD ir išreikštinio Eulerio metodu aproksimavimo paklaidos $E(\tau)$ bei konvergavimo greičiai $\rho(\tau)$

Table 2.9. The approximation errors $E(\tau)$ and convergence rates $\rho(\tau)$ for the (2.50)–(2.52) splitting scheme, NSFD and explicit Euler methods

τ	$E_{SS}(\tau)$	$\rho(\tau)$	$E_{NSFD}(\tau)$	$\rho(\tau)$	$E_{EE}(\tau)$	$\rho(\tau)$
$2,50 \cdot 10^{-3}$	$1,858 \cdot 10^{-4}$	1,926	$3,403 \cdot 10^{-2}$	0,290	$1,443 \cdot 10^{-2}$	0,786
$1,25 \cdot 10^{-3}$	$4,763 \cdot 10^{-5}$	1,964	$2,347 \cdot 10^{-2}$	0,536	$7,875 \cdot 10^{-3}$	0,874
$6,25 \cdot 10^{-4}$	$1,201 \cdot 10^{-5}$	1,988	$1,423 \cdot 10^{-2}$	0,722	$4,134 \cdot 10^{-3}$	0,930
$3,13 \cdot 10^{-4}$	$2,964 \cdot 10^{-6}$	2,018	$7,913 \cdot 10^{-3}$	0,846	$2,121 \cdot 10^{-3}$	0,963

2.2.5. Lygiagrečiosios versijos ir jų analizė

Kadangi lygčių su TLEO uždaviniai sunkesni dėl elipsinio operatoriaus nelokalumo, o praktikoje dažnai reikia spręsti lygtis su daugiau matmenų nei vienu ir tai padaryti kuo didesniu tikslumu, naudinga sprendimui naudoti lygiagrečiuosius algoritmus. Didelis dimensijų skaičius ir didesni tikslumo reikalavimai yra svarbiausi veiksniai, kurie lemia didesnes laiko sąnaudas. Disertacijoje siekiama šį laiką išnaudoti kuo efektyviau, todėl naujieji skaitiniai sprendimo algoritmai kuriami taip, kad jie turėtų geras lygiagretumo savybes. 2.2.1 skyrelyje minėta, kad BURA ir išplėtimo algoritmų uždaviniai lengvai lygiagretinami, todėl jiems tinka standartiniai lygiagretinimo metodai (Kriauzienė et al., 2020). Tačiau skaidymo algoritmų uždaviniai priklausomi vienas nuo kito ir jų negalima spręsti lygiagrečiai, todėl lygiagretinamas tik gaunamų tiesinių sistemų sprendimas. Tam galima naudoti jau esamas bibliotekas tiesinėms sistemoms spręsti – disertacijoje pasirinktas AGMG daugiatainklis sprendiklis (Notay, 2012). To privalumai – biblioteka optimizuota naujausiai kompiuterinei įrangai ir joje yra pakankamai parametrų, norint pritaikyti ją specifinėms sprendžiamo uždavinio reikmėms. Kita vertus, skaidymo metodai turi didesnę tikslumą, todėl neišvengiamai kyla klausimas, koks algoritmas geriausiai tinka konkrečiam uždaviniui.

Skaitiniuose eksperimentuose naudoti keturi algoritmai, iš kurių trys paskutiniai yra nauji: pirmasis yra AAA algoritmas (2.19). Antrasis yra modifikuotas

AAA, kai (2.19) metodu sprendžiamos lygtys, išskaidytos pagal (2.41)–(2.42) formules. Tam reikia aproksimuoti kitą funkciją:

$$f_2(z) = z^{-\alpha}, \quad z > 0.$$

Jos AAA aproksimacija bus tokia:

$$\tilde{r}_{m_{a2}}(z) = \frac{N_{m_{a2}}(z)}{D_{m_{a2}}(z)} = \tilde{c}_0 + \sum_{j=1}^{m_{a2}} \frac{\tilde{c}_j}{z - \tilde{d}_j}, \quad (2.54)$$

o sprendimo schema užrašoma taip:

$$\begin{aligned} \tilde{U}^{n+1/2,0} &= \tilde{U}^n, \\ \tilde{W}^{n+1/2,k} &= \left(\tilde{c}_0 I + \sum_{j=1}^{m_{a2}} \tilde{c}_j (A_h - \tilde{d}_j I)^{-1} \right) F(X, t^{n+1/2}, \tilde{U}^{n+1/2,k-1}), \end{aligned} \quad (2.55)$$

$$\begin{aligned} \tilde{V}^n &= \tilde{U}^n - \tilde{W}^{n+1/2,k}, \\ \tilde{V}^{n+1/2,k} &= \left(c_0 I + \sum_{j=1}^{m_a} c_j (A_h - d_j I)^{-1} \right) \tilde{V}^n, \end{aligned} \quad (2.56)$$

$$\begin{aligned} \tilde{U}^{n+1/2,k} &= \tilde{V}^{n+1/2,k} + \tilde{W}^{n+1/2,k}, \quad k = 1, \dots, K, \\ \tilde{U}^{n+1} &= 2 \tilde{U}^{n+1/2,K} - \tilde{U}^n. \end{aligned} \quad (2.57)$$

Trečiasis algoritmas yra modifikuota adityvi skaidymo schema, paremta (2.40) ir (2.41)–(2.42) metodais. Schemos lygtys:

$$\begin{aligned} \tilde{U}^{n+\sigma,0} &= \tilde{U}^n, \\ \tilde{W}^{n+\sigma,k} &= \left(\tilde{c}_0 I + \sum_{j=1}^{m_{a2}} \tilde{c}_j (A_h - \tilde{d}_j I)^{-1} \right) F(X, t^{n+\sigma}, \tilde{U}^{n+\sigma,k-1}), \end{aligned} \quad (2.58)$$

$$\begin{aligned} V^{n-1/m_a,k} &= \tilde{U}^n - \tilde{W}^{n+\sigma,k}, \\ D_{h,j} \frac{\tilde{V}^{n+j/m_a,k} - V^{n+(j-1)/m_a,k}}{\sigma \tau_n} + c_j A_h \tilde{V}^{n+j/m_a,k} &= 0, \quad j = 0, \dots, m_a, \end{aligned} \quad (2.59)$$

$$\begin{aligned} V^{n+j/m_a,k} &= 2 \tilde{V}^{n+j/m_a,k} - V^{n+(j-1)/m_a,k}, \quad j = 0, \dots, m_a, \\ \tilde{U}^{n+\sigma,k} &= \tilde{V}^{n+1,k} + \tilde{W}^{n+\sigma,k}, \quad k = 1, \dots, K, \\ \tilde{U}^{n+1} &= V^{n+1,K} + \tilde{W}^{n+\sigma,K}. \end{aligned}$$

Ketvirtasis algoritmas yra simetrinė skaidymo schema (2.43)–(2.44), kurioje TLEO aproksimuojamas naudojantis Furjė metodu su FFTW biblioteka (Johnson & Frigo, 2007). Tokie metodai parinkti dėl to, kad jie yra paremti lengvai lygiagretinamais AAA ir FFTW algoritmais.

Pabrėžtinai vienas svarbus taikomų metodų aspektas, kad sprendžiamų uždavinių skaičius priklauso nuo taikomo metodo: modifikuotam AAA ir adityviam skaidymo metodams reikia spręsti du nelokaliuosius uždavinius, o visiems kitiems metodams užtenka tik vieno. Be to, jei turime netiesinį lygties šaltinį, atliekamos prediktoriaus-korektorius iteracijos ir uždavinio skaičiavimo apimtis padidėja dvigubai. Dėl to lygiagrečiųjų algoritmų svarba uždaviniams su TLEO išauga dar labiau.

Šiame skyrelyje testuojamas lygiagrečiųjų TLEO lygčių sprendimo metodų tikslumas ir išplečiamumas. Algoritmai testuojami su trimis uždaviniais: pirmi du yra trupmeninio laipsnio difuzijos uždaviniai, o trečias – Gray–Scott lygčių sistema.

Pirmasis uždavinys aprašytas (1.19) lygtyje ir skirtas algoritmų tikslumo analizei. Iš pradžių gauti kontroliniai rezultatai spektriniam metodui, kuris paremtas FFTW biblioteka (Johnson & Frigo, 2007). Šaltinis buvo apskaičiuojamas taikant prediktoriaus-korektorius metodą. Pradžioje buvo fiksuojami parametrai $\alpha = 0,5$ ir $J_x = J_y = J_z = 1024$, ir lygtis integruojama su skirtingais laiko žingsniais iki laiko momento $T = 1$. Maksimalios paklaidos pagal laiko žingsnį τ yra:

$$e(0,1) = 2,1089 \cdot 10^{-4}, e(0,05) = 5,8807 \cdot 10^{-5}, e(0,025) = 1,4636 \cdot 10^{-5}.$$

Paskui buvo fiksuojamas $\tau = 0,002$ ir lygtis sprendžiama skirtingiems J_x ; paklaidos pagal taškų skaičių yra:

$$e(16) = 6,6590 \cdot 10^{-3}, e(32) = 1,6582 \cdot 10^{-3},$$

$$e(64) = 4,1406 \cdot 10^{-4}, e(128) = 1,0341 \cdot 10^{-4}.$$

Iš pateiktų duomenų matome, kad šis metodas turi antros eilės tikslumą pagal laiką ir erdvę. Toliau buvo nagrinėjamas AAA algoritmo (2.19) ir jo modifikacijos (2.55)–(2.57) tikslumas. Paklaida skaičiuota pagal maksimumo normą. Funkcijai aproksimuoti buvo parinkti $M = 50\,000$ taškų iš intervalo $[\lambda_{hmin}, \lambda_{hmax}]$. 2.10–2.11 lentelėse pateikiami metodų rezultatai.

2.10 lentelė. AAA algoritmo (2.19) paklaidos, kai $\alpha = 0,75; 0,5; 0,25$ ir keičiamos parametro m_a reikšmės. FFT – spektrinio metodo paklaidos

Table 2.10. The errors for the AAA algorithm (2.19) for different fixed values of α and changing values of m_a . FFT – the errors for the spectral method

α	$m_a = 5$	$m_a = 7$	$m_a = 10$	$m_a = 12$	FFT
0,75	$6,623 \cdot 10^{-4}$	$1,603 \cdot 10^{-5}$	$4,199 \cdot 10^{-5}$	$4,127 \cdot 10^{-5}$	$4,154 \cdot 10^{-5}$
0,50	$6,920 \cdot 10^{-4}$	$8,848 \cdot 10^{-5}$	$2,577 \cdot 10^{-5}$	$2,393 \cdot 10^{-5}$	$2,545 \cdot 10^{-5}$
0,25	$1,267 \cdot 10^{-3}$	$5,087 \cdot 10^{-4}$	$4,142 \cdot 10^{-4}$	$4,171 \cdot 10^{-4}$	$4,149 \cdot 10^{-4}$

Matome, kad norint pasiekti optimalias paklaidas, reikia imti nedaug, apie 10, aproksimacijos narių. Tas pats galioja ir modifikuotam algoritmui, tik jo tikslumas šiek tiek didesnis. Toliau pateikiami adityvaus skaidymo metodo (2.58)–(2.59) rezultatai (2.12 lentelė). Matome, kad šiuo atveju gaunamas didesnis tikslumas, kai $\alpha = 0,5$ ir $\alpha = 0,75$.

2.11 lentelė. Modifikuoto AAA algoritmo (2.55)–(2.57) paklaidos, kai $\alpha = 0,75; 0,5; 0,25$

Table 2.11. The errors for the modified AAA algorithm (2.55)–(2.57) for different values of α

α	$m_a = 5$	$m_a = 7$	$m_a = 10$	$m_a = 12$	$m_a = 15$
0,75	$9,960 \cdot 10^{-5}$	$4,430 \cdot 10^{-5}$	$4,119 \cdot 10^{-5}$	$4,175 \cdot 10^{-5}$	$4,102 \cdot 10^{-5}$
0,50	$1,982 \cdot 10^{-4}$	$2,536 \cdot 10^{-5}$	$2,543 \cdot 10^{-5}$	$2,582 \cdot 10^{-5}$	$2,421 \cdot 10^{-5}$
0,25	$2,351 \cdot 10^{-4}$	$3,970 \cdot 10^{-4}$	$4,150 \cdot 10^{-4}$	$4,140 \cdot 10^{-4}$	$4,189 \cdot 10^{-4}$

2.12 lentelė. Adityvaus skaidymo metodo (2.58)–(2.59) paklaidos, kai $\alpha = 0,75; 0,5; 0,25$ ir keičiamos parametro m_a reikšmės

Table 2.12. The errors for the additive splitting method (2.58)–(2.59) for different fixed values of α and changing values of m_a

α	$m_a = 5$	$m_a = 7$	$m_a = 10$	$m_a = 12$	$m_a = 15$
0,75	$3,144 \cdot 10^{-5}$	$7,124 \cdot 10^{-6}$	$8,155 \cdot 10^{-6}$	$7,866 \cdot 10^{-6}$	$8,189 \cdot 10^{-6}$
0,50	$1,338 \cdot 10^{-4}$	$2,299 \cdot 10^{-5}$	$1,569 \cdot 10^{-5}$	$1,561 \cdot 10^{-5}$	$1,553 \cdot 10^{-5}$
0,25	$2,930 \cdot 10^{-4}$	$4,153 \cdot 10^{-4}$	$4,163 \cdot 10^{-4}$	$4,163 \cdot 10^{-4}$	$4,164 \cdot 10^{-4}$

Antrasis uždavinys skirtas algoritmų išplečiamumo testams:

$$\begin{aligned} \frac{\partial u}{\partial t} + (-\Delta)^\alpha u &= F(x, y, z), \quad (x, y, z) \in \Omega = (0,1) \times (0,1) \times (0,1), \\ u(x, y, z, 0) &= 0, \quad (x, y, z) \in \Omega, \\ u(x, y, z, t) &= 0, \quad (x, y, z) \in \partial\Omega, \quad t > 0, \end{aligned} \quad (2.60)$$

čia šaltinio funkcija yra:

$$F(x, y, z) = \begin{cases} 1, & \text{jei } (x - 0,5)(y - 0,5)(z - 0,5) > 0; \\ 0, & \text{kitur.} \end{cases} \quad (2.61)$$

Tyrimui atlikti buvo parinktos tokios parametų reikšmės: taškų skaičius tinklo kraštinėje $J_x = J_y = J_z = 201$, laiko žingsnis $\tau = 0,5$. Skaičiavimams atlikti naudoti du Intel® Xeon® E5-2630V4 procesoriai (po 10 branduolių, taktinis dažnis 2.20 GHz) (Intel, 2022) VILNIUS TECH klasteryje „Vanagas“ su 32 GB

DDR4 RAM. Lygiagrešinimui naudojome MPI biblioteką. Skaitiniuose eksperimentuose matavome algoritmų pagreitėjimą ir efektyvumą keliems procesoriaus branduoliams. Iš pradžių buvo gauti kontroliniai duomenys, taikant klasikinę Kranko ir Nikolsono schemą, jie pateikti 2.13 lentelėje. Duomenys AAA algoritmui pateikti 2.14 lentelėje.

2.13 lentelė. Pagreitėjimas ir efektyvumas klasikinei Kranko ir Nikolsono schemei
Table 2.13. Speedup and efficiency for the classical Crank–Nicolson scheme

	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
$T_p^{(par)}$	344,9	193,5	91,71	63,68	56,65
$S_p^{(par)}$	1,0	1,782	3,761	5,416	6,088
$E_p^{(par)}$	–	0,891	0,940	0,678	0,381

2.14 lentelė. Pagreitėjimas ir efektyvumas klasikiniam AAA algoritmui, kai $m_a = 10$
Table 2.14. Speedup and efficiency for the classical AAA algorithm when $m_a = 10$

	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
			$\alpha = 0,5$		
$T_p^{(par)}$	267,9	149,1	72,04	48,49	44,24
$S_p^{(par)}$	1,0	1,797	3,719	5,525	6,056
$E_p^{(par)}$	–	0,899	0,930	0,690	0,378
			$\alpha = 0,25$		
$T_p^{(par)}$	267,6	149,2	71,63	48,65	44,05
$S_p^{(par)}$	1,0	1,797	3,736	5,501	6,075
$E_p^{(par)}$	–	0,899	0,934	0,688	0,380

Šie duomenys rodo, kad AAA algoritmo pagreitėjimas yra didesnis nei klasikinės Kranko ir Nikolsono schemos, o efektyvumas nežymiai didesnis. Toliau buvo nagrinėjama aproksimacijos narių skaičiaus m_a įtaka išplečiamumui, kai parametras $\alpha = 0,5$, o $m_a = 5$. Rezultatai pateikti 2.15 lentelėje.

Šiuo atveju algoritmo vykdymo laikas sutrumpėja, tačiau efektyvumas beveik nepasikeičia. Pabaigoje buvo patikrinta laiko žingsnio τ įtaka rezultatams, kurie pateikti 2.16 lentelėje. Parametrai buvo parinkti taip: $m_a = 10$, $\alpha = 0,5$, laiko žingsnis sumažintas iki $\tau = 0,025$. Dabar vykdymo laikas pailgėja, tačiau efektyvumas pasikeičia nežymiai.

2.15 lentelė. Pagreitėjimas ir efektyvumas AAA algoritmui, kai $m_a = 5$

Table 2.15. Speedup and efficiency for the AAA algorithm when $m_a = 5$

	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
$T_p^{(par)}$	107,1	61,27	29,76	20,15	18,14
$S_p^{(par)}$	1,0	1,748	3,599	5,315	5,904
$E_p^{(par)}$	–	0,874	0,900	0,652	0,369

2.16 lentelė. Pagreitėjimas ir efektyvumas AAA algoritmui, kai $m_a = 10$ ir $\tau = 0,025$

Table 2.16. Speedup and efficiency for the AAA algorithm when $m_a = 10$ and $\tau = 0.025$

	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
$T_p^{(par)}$	293,9	163,96	79,34	54,20	49,61
$S_p^{(par)}$	1,0	1,793	3,704	5,422	5,924
$E_p^{(par)}$	–	0,896	0,926	0,678	0,370

Trečiajame uždavinyje buvo testuojamas simetrinės skaidymo schemos išplečiamumas, sprendžiant trimatę Gray–Scott lygčių sistemą (dvimatis atvejis pateiktas Zhang et al. (2020)):

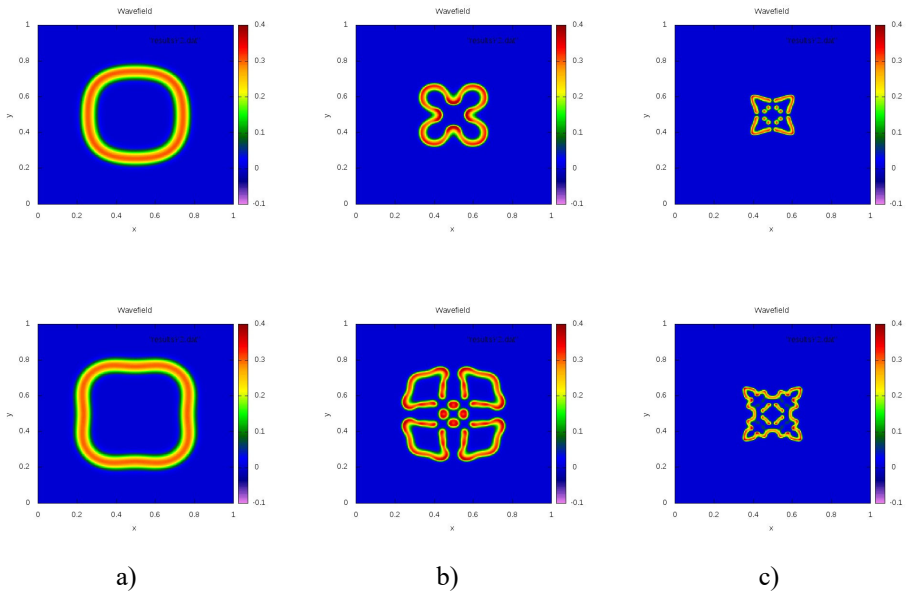
$$\begin{aligned} \frac{\partial u}{\partial t} + K_u(-\Delta)^\alpha u &= (1 - u)v^2 - Fu, \\ \frac{\partial v}{\partial t} + K_v(-\Delta)^\alpha v &= (1 - u)v^2 - (F + \lambda)v, \\ (x, y, z) \in \Omega &= (0,1) \times (0,1) \times (0,1), \end{aligned} \quad (2.62)$$

su parametų reikšmėmis: difuzijos greičiai $K_u = 2 \cdot 10^{-5}$, $K_v = 1 \cdot 10^{-5}$, maitinimo greitis $F = 0,03$, reagento v koncentracijos nykimo konstanta $\lambda = 0,061$. Ant srities krašto $\partial\Omega$ apibrėžtos homogeninės Dirichlet sąlygos, todėl sistemos sprendinys $u = 1 - \tilde{u}$. Pradinės sąlygos:

$$\begin{aligned} u(x, y, z, 0) &= 0,5, \quad v(x, y, z, 0) = 0,25, \quad (x, y, z) \in \Omega_1, \\ u(x, y, z, 0) &= 0, \quad v(x, y, z, 0) = 0, \quad (x, y, z) \in \Omega \setminus \Omega_1, \end{aligned} \quad (2.63)$$

čia $\Omega_1 = \{(x, y, z): (x - 0,5)^2 + (y - 0,5)^2 + (z - 0,5)^2 \leq 0,04^2\}$.

Yra gerai žinoma, kad sprendinys yra jautrus erdvės tinklo parinkimui, todėl visiems testams buvo užfiksuotas jo dydis $J_x = J_y = J_z = 256$. Tyrimo tikslai yra du: išsiaiškinti sprendinio kontūro priklausomybę nuo trupmeninio laipsnio α ir patikrinti simetrinės skaidymo schemos (2.43)–(2.44) pagreitėjimą ir efektyvumą, kai TLEO aproksimuojamas Furjė metodu.



2.1 pav. Gray–Scott lygties sprendinio v kontūrai, esant skirtingoms α reikšmėms: (a) $\alpha = 1$, (b) $\alpha = 0,85$, (c) $\alpha = 0,75$. Viršuje laikas yra $t = 2000$, apačioje – $t = 3000$. z reikšmė $z = 0,5$

Fig. 2.1. Contours of the solution v of the Gray–Scott equation at $z = 0.5$. Top – $t = 2000$; bottom – $t = 3000$

2.17 lentelė. Pagreitėjimas ir efektyvumas spektriniam simetriniam skaidymo algoritmui, kai $\alpha = 0,85$

Table 2.17. Speedup and efficiency for the spectral symmetric splitting method when $\alpha = 0.85$

	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
$T_p^{(par)}$	173,1	90,12	46,32	22,15	12,84
$S_p^{(par)}$	1,0	1,921	3,737	7,815	13,481
$E_p^{(par)}$	–	0,960	0,934	0,977	0,843

Gauti sprendinio v kontūrai pavaizduoti 2.1 paveiksle. Matome, kad, keisdami trupmeninį laipsnį, gauname visiškai skirtingus sprendinius. Tai reiškia, kad sprendinio kontūras jautrus laipsnio pokyčiams. Simetrinės skaidymo schemas išplečiamumo rezultatai pateikti 2.17 lentelėje. Jos duomenys rodo, kad spektrinio metodo efektyvumas yra didesnis (žr. 2.13 lentelę). Tačiau skaidymo schemą ga-

lima naudoti ir tada, kai spektrinio metodo taikyti negalima. Šio uždavinio skaičiavimo apimtis didelė – naudojant 20 procesoriaus branduolių, sprendinio kontūrų skaičiavimai užtrunka apie 3 valandas. Tuo galima paaiškinti, kodėl metodo efektyvumas mažiau krenta didinant branduolių skaičių – šis uždavinys yra netiesinis ir dėl linearizavimo jo apimtis padidėja, todėl procesoriai apkraunami labiau.

2.3. Biologinių jutiklių modelių su trupmeninio laipsnio difuzija skaitiniai sprendimo algoritmai

Šiame skyrelyje aptariamos skaitinės schemos biologinių jutiklių modeliams su nelokaliąja difuzija spręsti. Pradedame nuo modelių su klasikine difuzija aproksimacijos ir realizavimo algoritmų. Apibrėžiame diskrečius laiko ir erdvės tinklus:

$$\begin{aligned}\overline{\omega}_h &= \{x_j: x_j = jh, j = 0, \dots, J\}, & x_{J_d} &= d, & x_j &= d + a, \\ \overline{\omega}_t &= \{t^n: t^n = n\tau, n = 0, \dots, N\}, & t^N &= T.\end{aligned}$$

Pažymėkime $\tilde{S}_{k,j}^n, \tilde{P}_{k,j}^n$ skaitines uždavinio (1.3)–(1.6) sprendinių $S_k, P_k, k = 1, \dots, m$ aproksimacijas tinklo taške (x_j, t^n) . Nagrinėsime du modelio variantus: pirmasis variantas gaunamas, kai į difuzijos operatorių įtraukiami difuzijos koeficientai:

$$A_{hS}\tilde{S} = \begin{cases} -D_{se} \frac{2\tilde{S}_1 - \tilde{S}_0}{h}, & j = 0, \\ -\frac{1}{h} \left(D_{s,j+\frac{1}{2}} \frac{\tilde{S}_{j+1} - \tilde{S}_j}{h} - D_{s,j-\frac{1}{2}} \frac{\tilde{S}_j - \tilde{S}_{j-1}}{h} \right), & j = 1, \dots, J-2, \\ -\frac{1}{h} \left(D_{sb} \frac{-\tilde{S}_{J-1}}{h} - D_{sb} \frac{\tilde{S}_{J-1} - \tilde{S}_{J-2}}{h} \right), & j = J-1, \end{cases}$$

ir

$$A_{hP}\tilde{P} = \begin{cases} -\frac{1}{h} \left(D_{pe} \frac{\tilde{P}_2 - \tilde{P}_1}{h} - D_{pe} \frac{\tilde{P}_1}{h} \right), & j = 1, \\ -\frac{1}{h} \left(D_{p,j+\frac{1}{2}} \frac{\tilde{P}_{j+1} - \tilde{P}_j}{h} - D_{p,j-\frac{1}{2}} \frac{\tilde{P}_j - \tilde{P}_{j-1}}{h} \right), & j = 2, \dots, J-2, \\ -\frac{1}{h} \left(D_{pb} \frac{-\tilde{P}_{J-1}}{h} - D_{pb} \frac{\tilde{P}_{J-1} - \tilde{P}_{J-2}}{h} \right), & j = J-1. \end{cases}$$

Čia $D_{S,j-\frac{1}{2}} = D_{Se}, D_{P,j-\frac{1}{2}} = D_{Pe}$, kai $j = 1, \dots, J_d$ ir $D_{S,j+\frac{1}{2}} = D_{Sb}$, $D_{P,j+\frac{1}{2}} = D_{Pb}$, kai $j = J_d, \dots, J-1$. Šie operatoriai apibrėžiami, kai kraštinės sąlygos yra homogeninės ($\tilde{S}_j = 0$). Taip daroma, kad būtų galima taikyti spektrinį metodą trupmeninio laipsnio operatoriui apibrėžti. Tokiu atveju nehomogeninės biologinio jutiklio kraštinės sąlygos $\tilde{S}_l(d+a, t) = s_l$ turi būti įtrauktos į lygties šaltinį.

Biologinio jutiklio lygtis sprendžiamos simetriniu skaidymo algoritmu, kuris paremtas Kranko ir Nikolsono ir prediktoriaus-korektoriaus metodais. Pirmasis metodas turi antros eilės tikslumą ir yra nesąlygiškai stabilus, o antrąjį taikome netiesinėms lygtims linearizuoti. Pradžioje sprendžiamos lygtys substratams:

$$\frac{\tilde{S}_l^{n-\frac{1}{2}i} - \tilde{S}_l^{n-1}}{0.5\tau} + A_{hs}\tilde{S}_l^{n-\frac{1}{2}i} = -\frac{V_{max}\tilde{S}_l^{n-\frac{1}{2}i-1}}{K_M + \sum_{k=1}^m \tilde{S}_k^{n-\frac{1}{2}i-1}} + \delta_{j,J-1} \frac{D_{Sb}}{h^2} \tilde{S}_{l,j}^{n-\frac{1}{2}},$$

$$l = 1, \dots, m, \quad i = 1, 2, \tag{2.64}$$

čia $\tilde{S}_l^{n-\frac{1}{2}} = \frac{1}{2}(\tilde{S}_l^n + \tilde{S}_l^{n-1})$ ir $V_{max} = 0$, $j = J_d + 1, \dots, J$. Iteracijos apibrėžiamos taip:

$$\tilde{S}_l^{n-\frac{1}{2}0} = \tilde{S}_l^{n-1}, \quad \tilde{S}_l^{n-\frac{1}{2}} = \tilde{S}_l^{n-\frac{1}{2}2}.$$

Gaunamos tiesinės lygčių sistemos sprendžiamos standartiniu faktorizacijos algoritmu. Tada sprendžiamos lygtys produktams:

$$\frac{\tilde{P}_l^{n-\frac{1}{2}} - \tilde{P}_l^{n-1}}{0.5\tau} + A_{hp}\tilde{P}_l^{n-\frac{1}{2}} = \frac{V_{max}\tilde{S}_l^{n-\frac{1}{2}}}{K_M + \sum_{k=1}^m \tilde{S}_k^{n-\frac{1}{2}}}, \quad l = 1, \dots, m, \tag{2.65}$$

čia $\tilde{P}_l^{n-\frac{1}{2}} = \frac{1}{2}(\tilde{P}_l^n + \tilde{P}_l^{n-1})$. Taikydami minėtą metodą, užrašome skaidymo schemas lygtis. Skaidymą atliekame pagal netiesinį reakcijos ir tiesinį difuzijos procesus:

$$\frac{\tilde{S}_{l,j}^{n-\frac{2}{3}i} - \tilde{S}_{l,j}^{n-1}}{\frac{1}{2}\tau} = -\frac{V_{max} \frac{1}{2} \left(\tilde{S}_{l,j}^{n-\frac{2}{3}i-1} + \tilde{S}_{l,j}^{n-1} \right)}{K_M + \sum_{k=1}^m \frac{1}{2} \left(\tilde{S}_{k,j}^{n-\frac{2}{3}i-1} + \tilde{S}_{k,j}^{n-1} \right)},$$

$$l = 1, \dots, m, \quad i = 1, 2, \quad \tilde{S}_{l,j}^{n-\frac{2}{3}0} = \tilde{S}_{l,j}^{n-1}, \quad j = 0, \dots, J_d, \tag{2.66}$$

$$\tilde{S}_{l,j}^{n-\frac{2}{3}} = \tilde{S}_{l,j}^{n-\frac{2}{3}2},$$

$$\frac{\tilde{P}_{l,j}^{n-\frac{2}{3}} - \tilde{P}_{l,j}^{n-1}}{\frac{1}{2}\tau} = \frac{V_{max} \frac{1}{2} \left(\tilde{S}_{l,j}^{n-\frac{2}{3}} + \tilde{S}_{l,j}^{n-1} \right)}{K_M + \sum_{k=1}^m \frac{1}{2} \left(\tilde{S}_{k,j}^{n-\frac{2}{3}} + \tilde{S}_{k,j}^{n-1} \right)},$$

$$l = 1, \dots, m, \quad j = 0, \dots, J_d, \quad (2.67)$$

$$\frac{\tilde{S}_l^{n-\frac{1}{3}} - \tilde{S}_l^{n-\frac{2}{3}}}{\tau} + A_{hs} \frac{\tilde{S}_l^{n-\frac{1}{3}} + \tilde{S}_l^{n-\frac{2}{3}}}{2} = \delta_{j,J-1} \frac{D_{Sb} \tilde{S}_{l,J}^{n-\frac{1}{3}} + \tilde{S}_{l,J}^{n-\frac{2}{3}}}{h^2},$$

$$l = 1, \dots, m, \quad (2.68)$$

$$\frac{\tilde{P}_l^{n-\frac{1}{3}} - \tilde{P}_l^{n-\frac{2}{3}}}{\tau} + A_{hP} \frac{\tilde{P}_l^{n-\frac{1}{3}} + \tilde{P}_l^{n-\frac{2}{3}}}{2} = 0, \quad l = 1, \dots, m, \quad (2.69)$$

$$\frac{\tilde{S}_{l,j}^{n,i} - \tilde{S}_{l,j}^{n-\frac{1}{3}}}{\frac{1}{2}\tau} = - \frac{V_{max} \frac{1}{2} \left(\tilde{S}_{l,j}^{n,i-1} + \tilde{S}_{l,j}^{n-\frac{1}{3}} \right)}{K_M + \sum_{k=1}^m \frac{1}{2} \left(\tilde{S}_{k,j}^{n,i-1} + \tilde{S}_{k,j}^{n-\frac{1}{3}} \right)},$$

$$l = 1, \dots, m, \quad i = 1, 2, \quad \tilde{S}_{l,j}^{n,0} = \tilde{S}_{l,j}^{n-\frac{1}{3}}, \quad j = 0, \dots, J_d, \quad (2.70)$$

$$\frac{\tilde{P}_{l,j}^n - \tilde{P}_{l,j}^{n-\frac{1}{3}}}{\frac{1}{2}\tau} = \frac{V_{max} \frac{1}{2} \left(\tilde{S}_{l,j}^n + \tilde{S}_{l,j}^{n-\frac{1}{3}} \right)}{K_M + \sum_{k=1}^m \frac{1}{2} \left(\tilde{S}_{k,j}^n + \tilde{S}_{k,j}^{n-\frac{1}{3}} \right)},$$

$$l = 1, \dots, m, \quad j = 0, \dots, J_d. \quad (2.71)$$

Tokio skaidymo privalumas yra tas, kad tada turėsime po vieną lygtį su TLEO visiems substratams ir produktams ir taip skaičiavimus atliksime efektyviau.

Antras modelio variantas gaunamas, kai difuzijos koeficientai nėra įtraukiami į difuzijos operatorių. Tarkime, difuzijos koeficientai yra lygūs ir difuzijos, ir reakcijos sluoksniuose:

$$D_{Se} = D_{Sb} = D_S, \quad D_{Pe} = D_{Pb} = D_P.$$

Tada apibrėžiami operatoriai:

$$\mathcal{A}_{hS}\tilde{S} = \begin{cases} -\frac{2}{h}\frac{\tilde{S}_1 - \tilde{S}_0}{h}, & j = 0, \\ -\frac{1}{h}\left(\frac{\tilde{S}_{j+1} - \tilde{S}_j}{h} - \frac{\tilde{S}_j - \tilde{S}_{j-1}}{h}\right), & j = 1, \dots, J-2, \\ -\frac{1}{h}\left(\frac{-\tilde{S}_{J-1}}{h} - \frac{\tilde{S}_{J-1} - \tilde{S}_{J-2}}{h}\right), & j = J-1 \end{cases}$$

ir

$$\mathcal{A}_{hP}\tilde{P} = \begin{cases} -\frac{1}{h}\left(\frac{\tilde{P}_2 - \tilde{P}_1}{h} - \frac{\tilde{P}_1}{h}\right), & j = 1, \\ -\frac{1}{h}\left(\frac{\tilde{P}_{j+1} - \tilde{P}_j}{h} - \frac{\tilde{P}_j - \tilde{P}_{j-1}}{h}\right), & j = 2, \dots, J-2, \\ -\frac{1}{h}\left(\frac{-\tilde{P}_{J-1}}{h} - \frac{\tilde{P}_{J-1} - \tilde{P}_{J-2}}{h}\right), & j = J-1. \end{cases}$$

Dabar apibrėžiame trupmeninio laipsnio difuzijos operatorius. Pirma imame skaliarines sandaugas funkcijoms \tilde{S} ir \tilde{P} , kuriose įtraukiamos reikiamos nehomogeninės kraštinės sąlygos:

$$(\tilde{S}_l, \tilde{S}_k)_S = \tilde{S}_{l,0}\tilde{S}_{k,0}\frac{h}{2} + \sum_{j=1}^{J-1} \tilde{S}_{l,j}\tilde{S}_{k,j}h, \quad (\tilde{P}_l, \tilde{P}_k)_P = \sum_{j=1}^{J-1} \tilde{P}_{l,j}\tilde{P}_{k,j}h.$$

Difuzijos operatoriai \mathcal{A}_{hS} ir \mathcal{A}_{hP} yra simetriniai ir teigiamai apibrėžti, todėl jiems galime rasti tikrines reikšmes $\lambda_{Sj} > 0, \lambda_{Pj} > 0$ ir atitinkamas tikrines funkcijas Φ_{Sj}, Φ_{Pj} :

$$\begin{aligned} \mathcal{A}_{hS}\Phi_{Sj} &= \lambda_{Sj}\Phi_{Sj}, & j &= 0, \dots, J-1, \\ \mathcal{A}_{hP}\Phi_{Pj} &= \lambda_{Pj}\Phi_{Pj}, & j &= 1, \dots, J-1. \end{aligned} \quad (2.72)$$

Tada trupmeniniai difuzijos operatoriai \mathcal{A}_{hS}^α ir \mathcal{A}_{hP}^α apibrėžiami taip:

$$\mathcal{A}_{hS}^\alpha\tilde{S} = \sum_{j=0}^{J-1} \lambda_{Sj}^\alpha(\tilde{S}, \Phi_{Sj})_S \Phi_{Sj}, \quad \mathcal{A}_{hP}^\alpha\tilde{P} = \sum_{j=1}^{J-1} \lambda_{Pj}^\alpha(\tilde{P}, \Phi_{Pj})_P \Phi_{Pj}, \quad (2.73)$$

čia trupmeninis laipsnis $0 < \alpha < 1$. Analogiškai apibrėžiami TLEO pirmajam modelio variantui.

Biologinio jutiklio modeliai su trupmeninio laipsnio difuzija gaunami pakeitus difuzijos sluoksnio lygtis (2.68)–(2.69) pirmajam:

$$\frac{\tilde{S}_l^{n-\frac{1}{3}} - \tilde{S}_l^{n-\frac{2}{3}}}{\tau} + D_S \mathcal{A}_{hS}^\alpha \frac{\tilde{S}_l^{n-\frac{1}{3}} + \tilde{S}_l^{n-\frac{2}{3}}}{2} = \frac{D_S}{h^2} \mathcal{A}_{hS}^{(\alpha-1)} \delta_{j,J-1} \frac{\tilde{S}_{l,J}^{n-\frac{1}{3}} + \tilde{S}_{l,J}^{n-\frac{2}{3}}}{2},$$

$$l = 1, \dots, m, \quad (2.74)$$

$$\frac{\tilde{P}_l^{n-\frac{1}{3}} - \tilde{P}_l^{n-\frac{2}{3}}}{\tau} + D_P \mathcal{A}_{hP}^\alpha \frac{\tilde{P}_l^{n-\frac{1}{3}} + \tilde{P}_l^{n-\frac{2}{3}}}{2} = 0, \quad l = 1, \dots, m \quad (2.75)$$

ir antrajam variantui:

$$\frac{\tilde{S}_l^{n-\frac{1}{3}} - \tilde{S}_l^{n-\frac{2}{3}}}{\tau} + A_{hS}^\alpha \frac{\tilde{S}_l^{n-\frac{1}{3}} + \tilde{S}_l^{n-\frac{2}{3}}}{2} = A_{hS}^{(\alpha-1)} \frac{D_{Sb}}{h^2} \delta_{j,J-1} \frac{\tilde{S}_{l,J}^{n-\frac{1}{3}} + \tilde{S}_{l,J}^{n-\frac{2}{3}}}{2},$$

$$l = 1, \dots, m, \quad (2.76)$$

$$\frac{\tilde{P}_l^{n-\frac{1}{3}} - \tilde{P}_l^{n-\frac{2}{3}}}{\tau} + A_{hP}^\alpha \frac{\tilde{P}_l^{n-\frac{1}{3}} + \tilde{P}_l^{n-\frac{2}{3}}}{2} = 0, \quad l = 1, \dots, m. \quad (2.77)$$

Substratų lygtyse kraštines sąlygas turime įtraukti į lygčių šaltinius. Tai padarome užrašydami nelokalųjį operatorių taip (Ilic et al., 2006):

$$\begin{aligned} A_h^\alpha \frac{\tilde{S}_l^{n-\frac{1}{3}} + \tilde{S}_l^{n-\frac{2}{3}}}{2} &= A_h^{-\beta} A_h \frac{\tilde{S}_l^{n-\frac{1}{3}} + \tilde{S}_l^{n-\frac{2}{3}}}{2} - A_h^{-\beta} \frac{1}{h^2} \delta_{j,J-1} \frac{\tilde{S}_{l,J}^{n-\frac{1}{3}} + \tilde{S}_{l,J}^{n-\frac{2}{3}}}{2} \\ &= A_h^\alpha \tilde{S}_l^{n-\frac{1}{2}i} - A_h^{-\beta} \frac{1}{h^2} \delta_{j,J-1} \frac{\tilde{S}_{l,J}^{n-\frac{1}{3}} + \tilde{S}_{l,J}^{n-\frac{2}{3}}}{2}. \end{aligned}$$

Matome, kad lygčių šaltiniai bus nebelokalūs, nes kraštinėms sąlygoms pritaikomas nelokalasis operatorius.

Gautas difuzijos lygtis galima efektyviai išspręsti taikant FFT metodą, nes nagrinėjamo biologinio jutiklio modelio erdvės sritis yra vienmatis intervalas ir žinomos tikrinės reikšmės ir funkcijos. Jos pateikiamos antrajam variantui, tačiau pirmajam variantui tikrinės reikšmės λ_{Sj} ir λ_{Pj} yra panašios, užtenka tik trivialių skaičiavimų joms gauti, o tikrinės funkcijos Φ_{Sj} ir Φ_{Pj} tos pačios:

$$\Phi_{Sk,j} = \cos\left(\pi \frac{(j-0.5)k}{J}\right), \quad \lambda_{Sj} = \frac{4}{h^2} \sin^2\left(\frac{\pi(j-0.5)}{J}\right), \quad k, j = 1, \dots, J,$$

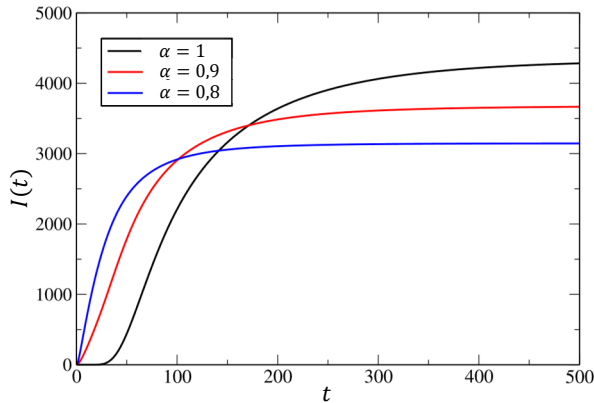
$$\Phi_{Pk,j} = \sin\left(\pi \frac{kj}{J}\right), \quad \lambda_{Pj} = \frac{4}{h^2} \sin^2\left(\frac{\pi j}{2J}\right), \quad k, j = 1, \dots, J-1.$$

Toliau pateikiami biologinio jutiklio atsako (1.18) priklausomybės nuo trupmeninio laipsnio tyrimų rezultatai abiem modelio variantams, kai substratų skaičius yra $m = 3$. Skaičiavimams naudoti tokie parametrai (Baronas et al., 2021):

$$\begin{aligned} V_{max,l} &= 5 \times 10^{-(9-l)}, & K_M &= 1 \times 10^{-4}, & l &= 1,2,3, \\ D_{Se} &= D_{Sb} = 4,5 \times 10^{-6}, & D_{Pe} &= D_{Pb} = 4,5 \times 10^{-6}, \\ s_1 &= 3,88574, & s_2 &= 11,65712, & s_3 &= 5,25716, \\ d &= 0,02\text{cm}, & a &= 0,04\text{cm}, & T &= 400\text{s}. \end{aligned}$$

Erdvės tinklas turi $N = 999$ taškų. Biologinio jutiklio atsako srovės reikšmė $I(t_k)$ išvedama kas sekundę $t_k = k$.

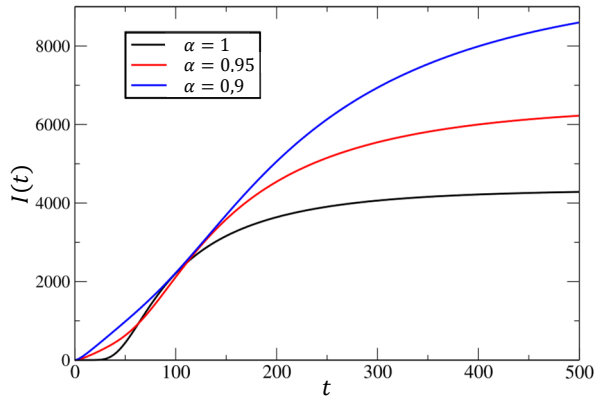
Nustatyta, kad biologinio jutiklio atsakas smarkiai priklauso nuo trupmeninio difuzijos laipsnio α . Pirmojo modelio varianto atsakai (2.2 pav.) pasiekia stacionariąją reikšmę anksčiau, o jos dydis mažėja, mažėjant trupmeniniam laipsniui. Taip pat itin keičiasi signalo augimo greitis pradžioje. Matome, kad signalas yra jautrus α pokyčiams: keičiant α reikšmę po 0,1, įsisotinimo srovė keičiasi po maždaug 500, o įsisotinimo laikas – po maždaug 150 s.



2.2 pav. Pirmojo biojutiklio modelio varianto elektros srovės funkcijos $I(t)$ dinamika trimis trupmeninio laipsnio reikšmėms: a) $\alpha = 1$, b) $\alpha = 0,9$, c) $\alpha = 0,8$

Fig. 2.2. The dynamics of the electric current function $I(t)$ for the first biosensor model variant

Antrojo varianto atsakai (2.3 pav.) elgiasi atvirkščiai – įsisotinimo reikšmė didesnė ir pasiekiamą vėliau, kai mažiname trupmeninį laipsnį. Šį kartą signalo jautrumas α pokyčiams didesnis: jei parametro reikšmė keičiasi po 0,05, signalo srovė keičiasi po maždaug 2000, kai $t = 500$ s.



2.3 pav. Antrojo biojutiklio modelio varianto elektros srovės funkcijos $I(t)$ dinamika trimis trupmeninio laipsnio reikšmėms: a) $\alpha = 1$, b) $\alpha = 0,95$, c) $\alpha = 0,9$

Fig. 2.3. The dynamics of the electric current function $I(t)$ for the second biosensor model variant

2.4. Antrojo skyriaus išvados

1. Skyriuje išanalizuoti skaitiniai metodai biologinių jutiklių modeliams spręsti. Klasikiniam modeliui taikomi baigtinių tūrių ir prediktoriaus-korektoriaus metodai.
2. Taip pat išanalizuoti nestacionariųjų diferencialinių lygčių su TLEO sprendimo algoritmai, kai TLEO aproksimuojame racionaliąja funkcija. Metodų stabilumas ir tikslumas pagrįstas teoriškai, taikant standartinius metodus. Visi taikyti metodai turi tikslumą, pakankamą praktiniams taikymams.
3. Išbandytos lygiagrečiosios sprendimo algoritmų versijos. Nauji disertacijos rezultatai rodo, kad visi lygiagretieji metodai yra efektyviai išplečiami dideliems uždaviniams.
4. Taikant TLEO metodus sukurta schema biologinio jutiklio modeliui su trupmenine difuzija spręsti. Gauti atsakai skirtingoms trupmeninio laipsnio reikšmėms, jie yra jautrūs laipsnio parinkimui: esant mažiems laipsnio pokyčiams, atsakai pastebimai keičiasi.

Skaitiniai biologinių jutiklių modelių atvirkštinių uždavinių sprendimo algoritmai ir jų analizė

Skyriuje pateikiama neuroninių tinklų apžvalga ir parenkamos tinklų architektūros eksperimentams. Atliekami tikslumo tyrimai dviems kraštinių sąlygų režimams, dviems triukšmo tipams, taip pat tikrinama leidžiamųjų koncentracijų srities mažinimo procedūra, pateikiami bei analizuojami atitinkamų skaičiavimo eksperimentų rezultatai. Pateikiama lygiagretaus DIRECT globaliosios optimizacijos metodo apžvalga, eksperimentai kraštinių sąlygų režimams, algoritmo išplečiamumui ir jų rezultatai. Skyriaus medžiaga publikuota Dapšys et al. (2023) straipsnyje ir pristatyta MMA2022 ir MMA2023 konferencijose.

3.1. Dirbtinių neuroninių tinklų algoritmai ir jų analizė

Vieną perspektyviausių metodų biologinio jutiklio uždaviniui spręsti laikytini neuroniniai tinklai (Baronas et al., 2004). Tai yra mašininio mokymo modelis, paremtas žmogaus smegenų funkcijomis (Marsland, 2015; Trask, 2019; Zhang et al., 2021a). Jie taikomi įvairiems aproksimavimo ir klasifikavimo uždaviniams spręsti. Šis modelis priklauso parametrinių mokymo su mokytoju (angl. *supervised learning*) metodų klasei, kai modelis apmokomas naudojant įvesties duomenis

su žinomomis išvestimis ir pagal juos nustatomi modelio parametrai. Tai yra pagrindinis disertacijoje taikomas metodas atvirkštiniam biologinių jutiklių uždaviniui spręsti.

Neuroniniai tinklai pasižymi didele architektūrų įvairove: pavyzdžiui, yra paprasti tinklai, kai neuronai juose suskirstyti sluoksniais; konvoliuciniai tinklai, paremti sąsūkos operacija; autoenkoderiai, skirti duomenims suspausti. Šis metodas yra orientuotas į įvairių atpažinimo ir klasifikavimo uždavinių sprendimą, todėl yra tinkamas ir kitiems nekorektiškiems uždaviniams spręsti. Kaip pavyzdys paminėtina, kad dirbtiniai neuroniniai tinklai jau yra pritaikyti atvirkštiniam šilumos laidumo uždaviniams spręsti, kai iš temperatūros pasiskirstymo reikia nustatyti pradinę uždavinio sąlygą (Shiguemori et al., 2004), kraštinę uždavinio sąlygą (Deng & Hwang, 2007; Kumar et al., 2020), arba šilumos šaltinį (Ghosh et al., 2011). Neuroniniai tinklai taip pat pritaikyti ir biologiniams jutikliams, kai juos modeliuojame be konkurencijos tarp substratų (Baronas et al., 2004), ir su ja (Litvinas & Baronas, 2014). Pastarajame straipsnyje į biologinio jutiklio modelį dar įtraukiamas ir išorinis difuzijos sluoksnis. Visuose minėtuose straipsniuose naudojamas visas biologinio jutiklio signalas, o ne pastovioji srovė. Signalui pritaikoma pagrindinių komponenčių analizė, siekiant sumažinti atsakų dimensiją. Taisant šį metodą, labai svarbu parinkti tinkamą pagrindinių komponenčių skaičių – dažniausiai parenkama tiek komponenčių, kad būtų išsaugoma kuo daugiau variacijos. Šis kriterijus nevisada tinkamas neuroniniams tinklams, nes tada jų apmokymo paklaida gali nesumažėti iki priimtino lygio (Cartas et al., 2010). Be to, nors straipsnių autoriai šį kriterijų pamini, jie konkrečiai neįvardija, kaip jie parenka reikiamą PK skaičių. Dar vienas klausimas kyla, kaip parenkamas paslėptojo sluoksnio neuronų skaičius. Baronas et al. (2004) straipsnyje minima euristika, susijusi su neuroninio tinklo laisvės laipsnių skaičiumi, tačiau kuo ji paremta, neaišku, nes ji necituojama ir plačiau nenagrinėjama. O Litvinas & Baronas (2014) straipsnyje paslėptųjų neuronų skaičiaus parinkimas niekaip neaiškinamas.

Anksčiau minėti straipsniai naudoja neuroninius tinklus, neatsižvelgdami į tai, kad duomenys gauti sprendžiant diferencialines lygtis. Tokiems uždaviniams sukurtas alternatyvus neuroninių tinklų tipas – PINN (angl. *Physics Informed Neural Networks*), kuriuose į neuroninį tinklą įvedama informacija apie lygtį, iš kurios gauti duomenys. Lu et al. (2021) straipsnyje neuroninis tinklas aproksimuoja sprendinį, kurį galima diferencijuoti taikant automatinio diferencijavimo metodą – tai yra sudėtinių funkcijų išvestinės taisyklės taikymas neuroniniams tinklams, panašus į atgalinį sužadinimą, tik šiuo atveju išvestinės skaičiuojamos ne pagal tinklo parametrus, o pagal įvestį. Diferencialinė lygtis, pradinės ir kraštinės sąlygos įvedamos į neuroninio tinklo tikslo funkciją kaip jų paklaidų L_2 normoje suma. Apmokymui naudojami srities, kurioje galioja lygtis, taškai ir taškai ant srities krašto. Atvirkštiniai uždaviniai sprendžiami taip pat, tik tada tinklui reikia pateikti papildomus duomenis apie sprendinį – jo reikšmes keliuose lygties

srities taškuose, tikslo funkcijoje atsiranda paklaida tarp papildomų duomenų ir sprendinio aproksimacijos, o nežinomi uždavinio parametrai apmokomi kartu su neuroninio tinklo parametrais. Pagrindiniai privalumai yra netiesinė sprendinio aproksimacija, kuriai nereikia erdvės ir laiko tinklelių, todėl lygtis galime efektyviai spręsti kai turime ir daugiau erdvės matmenų. Be to, toks metodas pritaikytas įvairiems diferencialiniams uždaviniams: integro-diferencialinėms, stochastinėms ir trupmeninio laipsnio išvestinių lygtims. Trukumai yra ilgesnis skaičiavimo laikas tiesioginiams uždaviniams, palyginti su baigtinių elementų metodu ir tai, kad kyla tinkamos neuroninio tinklo architektūros parinkimo klausimas. Straipsnyje testuojami du atvirkštiniai uždaviniai: Lorenzo sistemos ir reakcijos-difuzijos lygties parametrų nustatymas. Nors straipsnyje pateikiamas metodas duoda gerus rezultatus abiem atvejais, Lorenzo sistema netikrinta reikšmėms, kurioms esant ji tampa jautri pradinėms sąlygoms (tampa chaotiška), o reakcijos-difuzijos lygtyje parametrai atkuriami iš substrato koncentracijų, bet ne iš produktų generuojamos srovės. Šie atvejai turėtų testinius uždavinius apsunkinti.

PINN metodas neuroniniais tinklais aproksimuoja sprendinį, o lygtis į juos įtraukiama kaip tikslo funkcijos dalis. Viena iš alternatyvų yra lygtį spręsti skaitiškai, baigtinių skirtumų ar tūrių metodu, kai sprendikliai įtraukiami į neuroninio tinklo architektūrą kaip atskiras sluoksnis. Taip daroma Pakravan et al. (2021) straipsnyje, kuriame atvirkštinis uždavinys sprendžiamas pasitelkiant autoenkoderio architektūrą, kai enkoderis yra paprastas arba gilusis neuroninis tinklas, skirtas nežinomų parametrų atkūrimui, o dekodeeris yra minėtas lygties sprendiklio sluoksnis. Skirtingai nei taikant PINN metodą, neuroninio tinklo įvestis yra patys sprendinio duomenys. Ieškomi lygties parametrai gaunami iš enkoderio ir dekodeerio sandūros po apmokymo. Jo metu gaunamas enkoderis aproksimuoja atvirkštinį atvaizdį, kuris turimus duomenis apie sprendinį atvaizduoja į atitinkamus lygties parametrus. Šį atvaizdį galima naudoti parametrams atkurti realiuoju laiku – tai yra pagrindinis metodo privalumas. Kiti privalumai yra galimybė taikyti įvairius lygties sprendimo metodus: išreikštinius ir neišreikštinius, taip pat tuos, kuriems nereikia erdvės tinklelio; į tinklą galima nesunkiai įvesti specifines žinias praktinių taikymų atvejais, pavyzdžiui, skleidinius tam tikrose funkcijų bazėse, tvermės dėsnius ir panašiai. Metodą autoriai testavo spręsdami dvimatį atvirkštinį Puasono uždavinį: iš sprendinio nustatyti difuzijos koeficientą, kuris yra nehomogeninis, t. y. funkcija. Enkoderiui buvo naudojama konvoliucinio neuroninio tinklo architektūra. Nors straipsnio autoriai teigia, kad atvirkštiniam uždaviniui užtenka paprasto nuoseklaus neuroninio tinklo, pasirinktas konvoliucinis, nes tai leidžia aproksimuoti atvirkštinį atvaizdį keliems skirtingiems difuzijos koeficientams. Ieškomas koeficientas aproksimuojamas skleidiniu bazinėmis funkcijomis ir tada ieškomi skleidinio koeficientai. Šiuo atveju pasitelkiami ortogonalūs Zernikės polinomialai, kurie plačiai naudojami vaizdams rekonstruoti. Jie duoda didesnę

tikslumą vaizdams su nereguliariais raštais ir yra mažiau jautrūs apvalinimo paklaidoms. Tinklui apmokyti naudojamas vienas sprendinys visuose erdvės tinklelio taškuose, tačiau difuzijos koeficientas gali būti rekonstruojamas ir iš vieno taško: autoriai ima daugiau taškų, kad gautų didesnę tikslumą. Naudojamas autoenkoderis sėkmingai atkuria nežinomą parametą su 2% maksimalia santykinė paklaida; sprendinio paklaida – 1%. Be to, parametro atkūrimas yra atsparus sprendinį veikiančiam triukšmui: pridėjus Gauso triukšmą su standartiniu nuokrypiu 0.025, gaunamas panašus tikslumas.

Nors anksčiau aptarti metodai turi didelį tikslumą, tinkamos neuroninio tinklo architektūros parinkimo klausimas išlieka, o PINN atveju atvirkštinio uždavinio sprendimas panašus į klasikinį metodą, tik vietoje sprendinio naudojamas neuroninis tinklas, o optimizavimas atliekamas tinklo apmokymo metodais. Aplamai, abiejų metodų architektūrose daugiau dėmesio skiriama apmokymo būdų modifikacijoms, o pačių apmokomų tinklų architektūra yra paprasta – tiesioginio sklidimo arba konvoliuciniai tinklai. Todėl, disertacijos autoriaus nuomone, aptarti metodai didelės įtakos atvirkštinio biologinių jutiklių uždavinio rezultatams neturės. Atlikus lyginamąją kelių architektūrų analizę, nuspręsta naudoti būtent tiesioginio sklidimo ir vienmatės konvoliucijos architektūras.

Neuroninius tinklus implementuojame naudodamiesi *TensorFlow* ir *Keras* bibliotekomis, skirtomis *Python* programavimo kalbai. *TensorFlow* biblioteka skirta skaičiavimams su duomenų masyvais (kitai vadinami tenzoriais), o *Keras* biblioteka palengvina darbą su *TensorFlow*, suteikdama įrankius neuroninių tinklų modeliams sudaryti ir apmokyti. *TensorFlow* iš neuroninio tinklo konfigūracijos sukuria kryptinį skaičiavimų grafą, kurio lankai yra duomenų masyvai, o viršūnės – operacijos su jais (Pang et al., 2020). Šis grafas yra kaip tvarkaraštis, kuris nurodo, kokius veiksmus ir kada reikia atlikti prognozavimo metu ir palengvina parametrų gradiento skaičiavimą apmokant tinklą. Biblioteka suteikia galimybę neuroninio tinklo skaičiavimus atlikti lygiagrečiai, naudojantis grafikos procesoriais (angl. *graphics processing unit* – GPU). Šie prietaisai, lyginant su procesoriais, turi labai daug branduolių (apie 1000 eilės), kurie optimizuoti tiesinės algebros operacijoms atlikti. Todėl neuroninių tinklų apmokymas su GPU yra žymiai greitesnis (Coates et al., 2009). Tiesinės algebros operacijoms atlikti *TensorFlow* naudoja CUDA paremta *Nvidia* biblioteką cuDNN.

Neuroninius tinklus galima apmokyti ir naudojant kelis GPU, remiantis duomenų lygiagretumu. Neuroninio tinklo modelis ir jo parametrai išdalijami daugiui GPU, kiekvienas taip pat gauna savo apmokymo duomenų dalį. Tada kiekvienas GPU skaičiuoja savo duomenų dalies įnašą į gradientą. Paskui atliekama sinchronizuota „all-reduce“ operacija, kai visų procesorių kontribucijos į gradientą sudedamos, gaunama galutinė gradiento reikšmė ir ji dalijama visiems procesoriams. Lygiagretiesiems skaičiavimams su keliais GPU naudojama NCCL (angl. *Nvidia Collective Communication Library*) biblioteka.

3.1.1. Tiesioginio sklidimo tinklas su pagrindinių komponentių analize

Iš pradžių aptariama tiesioginio sklidimo tinklo architektūra. Pradedama nuo bendros neuroninių tinklų sandaros. Šie tinklai sudaryti iš neuronų – skaičiavimo vienetų, kurie priima signalą, jį apdoroja ir išveda išvesties signalą. Tarkime, signalą sudaro n kanalų. Pažymėkime $x_{i,sig}$, $i = \overline{0, n-1}$, vieno kanalo skaitinę reikšmę; w_i , $i = \overline{0, n-1}$, – kanalo svorį, o b_{sig} – foninę reikšmę. Neuronų išvesties signalo q_{sig} formulė yra:

$$q_{sig} = f_{act}(w_0x_{0,sig} + w_1x_{1,sig} + \dots + w_{n-1}x_{n-1,sig} + b_{sig}). \quad (3.1)$$

Tai reiškia, kad neuronas apskaičiuoja savo įvesčių svertinę sumą, prideda foninę reikšmę, gaunamai reikšmei pritaiko aktyvavimo funkciją $f_{act}(x)$ – rezultatas ir yra išvesties signalas. Svoriai nusako, kiek svarbi kiekviena neurono įvestis galutiniam signalui. Foninė reikšmė nurodo, kokią reikšmę turi viršyti neurono įvestis, kad neuronas duotų teigiamą signalą. Aktyvavimo funkcijų būna įvairių, jomis gali būti sigmoidinės, dalimis tiesinės, eksponentinės ir kitokios funkcijos: disertacijoje naudojamos trys. Pirmoji yra ReLU (angl. *Rectified Linear Unit*), kuri duoda teigiamą signalą, kai įvestis teigiama. Jei ji neigiama, funkcija nurodo neuronui nepridėti savo signalo prie visų:

$$f_{act}(x) = \max(0, x). \quad (3.2)$$

Antroji yra hiperbolinis tangentas, kuris gali įgauti ir neigiamas reikšmes. Tai galima interpretuoti taip: funkcija gali nurodyti neuronui slopinti bendrą visų neuronų signalą:

$$f_{act}(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \quad (3.3)$$

Paskutinė funkcija yra tiesinė $f_{act}(x) = x$. Ji naudojama tik išvesties sluoksnyje.

Tinklo neuronai grupuojami sluoksniais. Tarkime, neuroninį tinklą sudaro L sluoksnių. Kiekviename sluoksnyje yra s_l , $l = \overline{0, L}$ neuronų. Tiesioginio sklidimo architektūroje sluoksniai jungiami taip, kad kiekvieno sluoksnyje esančio neurono įvestis būtų visi prieš tai buvusio sluoksnio neuronai. Tokį sluoksnį vadiname visiškai jungiu. Tinklo sluoksnius patogiau aprašyti naudojantis vektoriais ir matricomis. Tegul $a^{(l)}$ yra l -tojo sluoksnio neuronų išvesčių vektorius, kuriame yra s_l elementų – jis yra vadinamas sluoksnio aktyvacija.

Kiekvienas sluoksnis turės savo aktyvavimo funkciją $f_{l,act}$ ir foninį vektorių $b^{(l)}$. Kadangi neuronai skaičiuoja svertines sumas, svorius tarp sluoksnių galime

užrašyti matricomis, pažymėkime jas $W^{(l)}$. l -tojo sluoksnio aktyvacijos formulė yra:

$$a^{(l)} = f_{l,act}(W^{(l)}a^{(l-1)} + b^{(l)}), \quad l = \overline{1, L}. \quad (3.4)$$

Čia svorių matricos $W^{(l)}$ dydis yra $s_l \times s_{l-1}$. Visi vektoriai yra stulpeliai, o funkcijos $f_{l,act}(x)$ taikomos kiekvienam vektoriaus elementui atskirai.

Sluoksniai $a^{(0)}$ ir $a^{(L)}$ yra ypatingi – jie atitinkamai vadinami įvesties ir išvesties sluoksniais. Paskesnysis yra skirtas duomenims įvesti į tinklą ir neturi aktyvavimo funkcijos, o pastarasis sluoksnis išveda galutinį tinklo rezultatą. Visi sluoksniai, esantys tarp jų, vadinami paslėptaisiais. Jie reikalingi, kai turime duomenis, kurie nėra tiesiškai koreliuoti, nes jei bandytume su tokiais duomenimis apmokyti vienasluoksnį tinklą (tik su įvesties ir išvesties sluoksniais), jis nesugebėtų išmokti dėsningumą, kurie sieja įvestis su išvestimis (Trask, 2019). Paslėptieji sluoksniai leidžia neuroniniam tinklui pačiam sukurti naują duomenų rinkinį, kuris būtų tiesiškai koreliuotas su duotomis išvestimis. Dėl to paslėptųjų sluoksnio aktyvavimo funkcijos turi būti netiesinės, nes priešingu atveju, gaunamas daugiasluoksnis tinklas, kuris ekvivalentus vienasluoksniam tinklui.

Kaip minėta anksčiau, kadangi neuroniniai tinklai yra mašininio mokymo metodas, juos reikia apmokyti, o tam reikia turėti duomenų rinkinį, vadinamą apmokymo rinkiniu (Zhang et al., 2021a). Pažymėkime X_{tr} apmokymo įvesčių matricą, kurioje yra N eilučių – apmokymo mėginių, ir p stulpelių – požymių. Taip pat pažymėkime $Y_{tr} - N \times q$ matricą, kurios stulpeliai yra išvesties požymiai, o eilučių tiek pat, kiek ir X_{tr} . Porą (X_{tr}, Y_{tr}) laikysime apmokymo rinkiniu. Tada įvesties ir išvesties sluoksniai atitinkamai turės $s_0 = p$ ir $s_L = q$ neuronų.

Apmokymo proceso metu minimizuojama paklaida tarp neuroninio tinklo prognozių apmokymo įvestims ir žinomų apmokymo išvesčių. Paklaidą apibrėžia neuroninio tinklo tikslo funkcija E_{loss} , jų būna įvairių – parinkimas priklauso nuo tinklo taikymų. Paprastumo dėlei, nagrinėjamas atvejis kai tikslo funkcija yra kvadratų sumos paklaida:

$$E_{loss}(x, y) = \sum_{i=1}^q \left(a_i^{(L)}(x) - y_i \right)^2, \quad (3.5)$$

čia x – apmokymo mėginys iš matricos X_{tr} , y – jį atitinkanti išvestis iš Y_{tr} . Tikslo funkcija optimizuojama pagal parametrus $W^{(l)}, b^{(l)}, l = \overline{1, L}$.

Paprasčiausias tikslo funkcijai optimizuoti yra gradientinis nusileidimas, kuriam reikia žinoti funkcijos gradientą. Teoriškai tikslo funkcijos gradientą galima skaičiuoti rankiniu būdu – naudojantis formulėmis, prieš tai į funkciją įrašius (3.4) lygtis kiekvienam l . Tačiau taip daryti sudėtinga, nes

sudėtingesniems tinklams tai užimtu per daug laiko. Efektyvesnis būdas – apskaičiuoti gradientą rekurentiškai. Šis metodas vadinamas atgaliniu sužadinimu (angl. *backpropagation*) – kai paklaidos reikšmė perduodama sluoksniams nuo paskutinio iki pirmojo. Paprasčiausias atvejis implementuoja stochastinį gradientinį nusileidimą, kai neuroninis tinklas apmokomas po vieną apmokymo mėginį kiekviename žingsnyje.

Taikant atgalinį sužadinimą, tikslo funkcijos gradientas skaičiuojamas pagal išvestinių grandinės taisyklę sudėtinėms funkcijoms. Pažymėkime $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$, $l = \overline{1, L}$. Tada:

$$a^{(l)} = f_{l,act}(z^{(l)}), \quad l = \overline{1, L}. \quad (3.6)$$

Paklaidos funkcijai pritaikoma grandinės taisyklė (Zhang et al., 2021a):

$$\frac{\partial E_{loss}}{\partial z^{(l)}} = \text{prod} \left(\frac{\partial E_{loss}}{\partial a^{(l)}}, \frac{\partial a^{(l)}}{\partial z^{(l)}} \right) = \frac{\partial E_{loss}}{\partial a^{(l)}} * f'_L(z^{(L)}), \quad (3.7)$$

čia prod operatorius reprezentuoja daugybą, kai užtikrinamas reikiamas rezultato matricos dydis transponuojant daugiklius ir keičiant jų daugybos tvarką. Lygtyje naudojama elementinė sandauga, nes jei funkcija skaičiuojama kiekvienam elementui atskirai, tai ir jos išvestinė skaičiuojama taip pat. Kadangi šiuo atveju daugikliai vienodų matmenų, išvestinės pagal paskutinį sluoksnį ir aktyvavimo funkcijos išvestinės elementai dauginami atskirai.

Paklaidos funkcijos pokytį pagal aktyvavimo funkcijos įvestį galima sužinoti, įvertinus aktyvavimo funkcijos poveikį sluoksnio išvesties paklaidai, pasinaudojus funkcijos išvestine. Turėdami $\frac{\partial E_{loss}}{\partial z^{(l)}}$, pagal $z^{(l)}$ apibrėžimą, galima apskaičiuoti šias išvestines:

$$\frac{\partial E_{loss}}{\partial W^{(L)}} = \text{prod} \left(\frac{\partial E_{loss}}{\partial z^{(L)}}, \frac{\partial z^{(L)}}{\partial W^{(L)}} \right) = \frac{\partial E_{loss}}{\partial z^{(L)}} a^{(L-1)T}, \quad (3.8)$$

$$\frac{\partial E_{loss}}{\partial b^{(L)}} = \text{prod} \left(\frac{\partial E_{loss}}{\partial z^{(L)}}, \frac{\partial z^{(L)}}{\partial b^{(L)}} \right) = \frac{\partial E_{loss}}{\partial z^{(L)}}, \quad (3.9)$$

$$\frac{\partial E_{loss}}{\partial a^{(L-1)}} = \text{prod} \left(\frac{\partial E_{loss}}{\partial z^{(L)}}, \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \right) = W^{(L)T} \frac{\partial E_{loss}}{\partial z^{(L)}}. \quad (3.10)$$

Pirmoji lygtis teigia, kad kiekvieno sluoksnio neurono paklaida perduodama paskesnio sluoksnio neuronams, o jo aktyvacijos reikšmės dauginamos iš gautųjų paklaidų, kad jos turėtų tokį patį lygmenį. Sviurių tarp minėtų sluoksnių korekcijos randamos išdėsčius sudaugintas paklaidas į matricą. Antroji lygtis nurodo, kad foninis sluoksnio vektorius ir sluoksnio paklaida prieš pat pritaikant aktyvacijos funkciją keičiasi vienodai greitai. Paskutinės lygties prasmė tokia: kiekvieno

sluoksnio neurono paklaida yra svorinė kito sluoksnio neuronų paklaidų suma, kur kiekvienas svoris nusako neurono svarbą. Transponuojant svorių matricą, paklaidoms naudojami tie patys svoriai kaip ir priekinio sužadavimo metu. To užtenka, kad būtų galima efektyviai mažinti neuroninio tinklo apmokymo paklaidą.

Kadangi L parinktas laisvai, ankstesnės formulės galioja ir visiems $l = \overline{1, L}$. Pažymėkime $\delta^{(l)} = \frac{\partial E_{\text{loss}}}{\partial a^{(l)}}$, $\Delta^{(l)} = \frac{\partial E_{\text{loss}}}{\partial W^{(l)}}$ ir $B^{(l)} = \frac{\partial E_{\text{loss}}}{\partial b^{(l)}}$. Atlikus pertvarkymus gauname:

$$\delta^{(l-1)} = W^{(l)T} [\delta^{(l)} * f'_{l,act}(z^{(l)})], \quad (3.11)$$

$$\Delta^{(l)} = [\delta^{(l)} * f'_{l,act}(z^{(l)})] a^{(l-1)T}, \quad (3.12)$$

$$B^{(l)} = [\delta^{(l)} * f'_{l,act}(z^{(l)})]. \quad (3.13)$$

Neuroninio tinklo išvesties sluoksnyje paklaida yra tikslo funkcijos E_{loss} išvestinė pagal sluoksnio aktyvacijos vektorių:

$$\delta^{(L)} = \frac{\partial E_{\text{loss}}}{\partial a^{(L)}} = 2 \left(a_i^{(L)}(x) - y_i \right). \quad (3.14)$$

Parametrams naujos reikšmės priskiriamos taip:

$$W^{(l)} := W^{(l)} - \alpha_{lr} \Delta^{(l)}, \quad (3.15)$$

$$b^{(l)} := b^{(l)} - \alpha_{lr} B^{(l)}, \quad (3.16)$$

čia α_{lr} – parametras, vadinamas apmokymo greičiu, kuris nusako, koku greičiu artėjame prie lokalojo paklaidos minimumo. Nuo apmokymo greičio esmingai priklauso algoritmo konvergavimas. Jei šio parametro reikšmė per didelė, neuroninio tinklo parametų reikšmės gali pradėti svyruoti aplink lokalojo minimumo tašką jo nepasiekdamos, arba gali net tolti nuo jo. Todėl tinkamai parinkti apmokymo greitį yra labai svarbu.

Stochastinis gradientinis nusileidimas turi vieną trūkumą: jei imtumėme po vieną apmokymo rinkinio elementą, ir kiekvieno atžvilgiu vykdomas parametų atnaujinimas, gaunamas chaotiškas apmokymo procesas – žingsnių kryptys labai varijuoja. To išvengti padeda apmokymas partijomis (angl. *batch learning*) (Trask, 2019), kai apmokymo rinkinys suskaidomas į poaibių, kurių kiekviename yra ne daugiau nei K elementų. Kiekvienam partijos elementui apskaičiuojami parametų gradientai, jie suvidurkinami ir parametrams priskiriamos naujos reikšmės. Tarkime $\Delta^{(l)}(x^{(k)}, y^{(k)})$ ir $B^{(l)}(x^{(k)}, y^{(k)})$ yra atitinkamai svorių ir foninių vektorių gradientai, kai jiems skaičiuoti naudojamas k -tasis apmokymo mėginys iš partijos. Tada neuroninio tinklo parametrams naujos reikšmės priskiriamos pagal formules:

$$W^{(l)} := W^{(l)} - \frac{\alpha_{lr}}{K} \sum_{k=0}^{K-1} \Delta^{(l)}(x^{(k)}, y^{(k)}), \quad (3.17)$$

$$b^{(l)} := b^{(l)} - \frac{\alpha_{lr}}{K} \sum_{k=0}^{K-1} B^{(l)}(x^{(k)}, y^{(k)}). \quad (3.18)$$

Taikant apmokymą partijomis, gaunamas glodesnis ir greitesnis apmokymo procesas ir optimizavimo metu parametų reikšmės mažiau svyruoja. Kai apmokymo metu neuroniniam tinklui pateikiami visi apmokymo rinkinio bandiniai, sakoma, kad užbaigta viena optimizavimo epocha. Dažniausiai optimizavimo procesas stabdomas atlikus tam tikrą epochų skaičių.

Tačiau maža apmokymo paklaida dar nereiškia, kad neuroninio tinklo tikslumas bus didelis. Kuo ilgiau trunka apmokymo procesas, tuo smulkesnius duomenų požymius tinklas įsimena. Vadinas, jei tinklas mokomas per ilgai, jis įsimena visus apmokymo mėginius, bet neaptinka bendrų požymių (Trask, 2019). Todėl, nors neuroninis tinklas tiksliai atpažįsta koncentracijų reikšmes iš apmokymo rinkinio, jo tikslumas naujiems duomenims yra prastas. Šis reiškinys vadinamas persimokymu (angl. *overfitting*) (Petkevičius, 2023; Shalev-Shwartz & Ben-David, 2014). Tai gali nutikti ir tada, kai neuroninis tinklas yra pernelyg sudėtingas, lyginant su duomenų kiekiu – modelis pakankamai galingas, kad duomenis įsimintų.

Siekiant to išvengti, pritaikome regularizaciją – metodus, kurie skatina duomenų generalizavimą apsunkindami smulkių požymių apmokymą. Tai galima padaryti Tichonovo metodu, tačiau disertacijoje taikomas kitas metodas – atsitiktinio išmetimo (angl. *random dropout*), kai kiekviename mokymo žingsnyje atsitiktinai parinktos tinklo neuronų dalies išvestys prilyginamos nuliui ir šie neuronai mokymo žingsnyje nedalyvauja. Toks veiksmas sumažina efektyvų tinklo dydį, o mažesni tinklai nėra linkę persimokyti. Atliekant pakankamai daug žingsnių, apmokomi visi tinklo neuronai. Dar vienas būdas yra sumažinti tinklo sudėtingumą – tą galima pasiekti sumažinant turimų duomenų kiekį, nes tada įvesties sluoksnis bus paprastesnis ir jame bus mažiau parametų.

Duomenų dimensijai mažinti taikomas pagrindinių komponentų analizės (angl. *principal component analysis* – PCA) metodas (Marsland, 2015). Jį taikant, turimi didelės dimensijos duomenys yra projektuojami į mažesnės dimensijos hiperplokštumą, naudojant tiesinę transformaciją, kuri išsaugo kiek įmanoma daugiau originalaus duomenų rinkinio variacijos, ir pašalina perteklinius duomenų požymius, kurie gali būti tiesiškai priklausomi, arba nesvarbūs atliekamai užduočiai. Projektuoti duomenys vadinami pagrindinėmis komponentėmis (PK). Tai yra naudinga biologinio jutiklio atsakams, kurie turi daug informacijos – kiekvienu laiko momentu turime po skaitinę reikšmę, todėl kiekvienas laiko momentas gali

būti laikomas požymiu; dažniausiai jų būna keli šimtai. Taikant PCA, biologinio jutiklio atsakų požymių skaičių galima sumažinti iki keliasdešimties.

Prieš atliekant PCA, pradiniai duomenys normalizuojami. Tarkime, X yra $n \times p$ duomenų matrica, kurioje n – atsakų skaičius, o p – laiko momentų skaičius. Normalizavimas atliekamas pritaikius formulę:

$$X_{norm} = \frac{X - J_{n,1}\mu}{J_{n,1}s}, \quad (3.19)$$

čia X_{norm} – normalizuotų atsakų matrica. μ ir s – atitinkamai vidurkių ir standartinų nuokrypių pagal atsakus matricos, abiejų dydis – $1 \times p$. $J_{n,1}$ – $n \times 1$ matrica, kurios visi elementai lygūs 1. Dalybos veiksmas vykdomas kiekvienam matricos elementui atskirai.

Tada skaičiuojama duomenų kovariacijų matrica:

$$C = \frac{1}{n} X_{norm}^T X_{norm}. \quad (3.20)$$

Projekcijos transformacija gaunama atlikus tikrinę kovariacijų matricos išskaidymą – apskaičiuojamos jos tikrinės reikšmės ir tikriniai vektoriai; gaunamas išskaidymas:

$$C = W^T \Lambda W. \quad (3.21)$$

Matrica W ir yra ieškomoji projekcijos transformacija, jos stulpeliai yra kovariacijų matricos tikriniai vektoriai. Matricos Λ įstrižainėje yra tikrinės reikšmės α_j , $1 \leq j \leq p$ – jos nurodo, kiek variacijos išsaugo duota komponentė j . Ji išsaugo tokią variacijos dalį:

$$\frac{\alpha_j}{\sum \alpha_j}. \quad (3.22)$$

Siekiant sumažinti duomenų dimensiją iki l , $l \leq p$, vektoriai surikiuojami pagal išsaugomą variacijos dalį mažėjimo tvarka. Tada iš tikrinių vektorių matricos paimami tik pirmieji l vektoriai:

$$T_l = X_{norm} W_l, \quad (3.23)$$

čia T_l – transformuoti duomenys, W_l – matrica, sudaryta iš pirmųjų l matricos W stulpelių.

Alternatyvus metodas projekcijos transformacijai gauti yra singularusis išskaidymas (angl. *singular value decomposition*, SVD). Jo privalumas tas, kad nereikia skaičiuoti duomenų kovariacijos matricos ir galime atlikti viena matricę daugyba mažiau. Išskaidome matricę C taip:

$$C = U \Sigma V, \quad (3.24)$$

čia $U (n \times n)$ ir $V (p \times p)$ yra atitinkamai kairiųjų ir dešiniųjų vektorių matricos, $\Sigma (n \times p)$ – matrica, kurios įstrižainėje yra singuliariosios reikšmės σ_j . Jos rodo, kokią dalį variacijos išsaugo komponentė j pagal formulę:

$$\frac{\sigma_j^2}{\sum \sigma_j^2}. \quad (3.25)$$

Duomenų dimensiją sumažinama, paliekant l didžiausių singuliarinių reikšmių ir jas atitinkančių matricos U stulpelių:

$$T_l = U_l \Sigma_l. \quad (3.26)$$

3.1.2. Vienmatės konvoliucijos tinklas

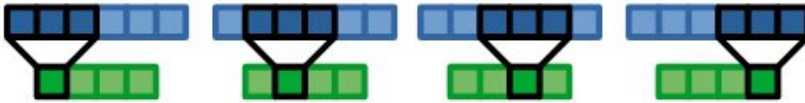
3.1 skyrelyje aptarta architektūra turi trūkumų. Pirmas yra tai, kad tiesioginio sklidimo neuroniniai tinklai daro prognozes pagal reikšmes visuose atsako taškuose, imdami jas atskirai ir nekreipdami dėmesio į lokalią informaciją, kuri yra gretimuose atsako taškuose – reikšmės juose dažniausiai yra susijusios viena su kita, t. y. sudaro erdvinę struktūrą. Antras trūkumas pasireiškia kai naudojama PCA – tai yra tiesinė duomenų transformacija, kuri gali nepakankamai tiksliai aptikti svarbiausius duomenų požymius. Trečias trūkumas yra tai, kad dalis taškų, naudojamų atsakui atpažinti, yra nereikalingi, nes paprastas neuroninis tinklas turės skirtingus svorius kiekvienam erdvės taškui, todėl skirtingose atsako vietose esantis panašus raštas irgi bus aptinkamas naudojant skirtingus svorius, nors panašioms raštams galima naudoti tuos pačius svorius.

Siekiant išnaudoti erdvinę atsako struktūrą, tiksliau atlikti duomenų glaudinimą ir sumažinti tinklo parametrų skaičių, atsako atpažinimui galima naudoti vienmačius konvoliucinius neuroninius tinklus (angl. *1D convolutional neural networks* – 1DCNN) (Acquarelli et al., 2017; Mozaffari & Tay, 2020; Zhang et al., 2021a), kuriuose yra specialūs sluoksniai – konvoliuciniai sluoksniai, kurie atlieka sąsūkos operaciją ir sutelkimo sluoksniai, apibendrinantys gretimų atsako regionų informaciją. Dėl mažesnio tinklo parametrų skaičiaus, ir dėl to, kad sąsūkos operacija nesunkiai lygiagretinama naudojantis grafikos procesoriais (GPU), konvoliuciniai tinklai yra efektyvesni skaičiavimų apimties prasme. Šiuos tinklus galima apibendrinti ir didesniai erdvės dimensijų skaičiui – tokie tinklai naudingi sprendžiant dalinių išvestinių diferencialinių lygčių atvirkštinius uždavinius, kai sprendiniai yra daugiamatiai.

Sąsūkos operacija naudojama dėl dviejų privalumų: sąsūka gali aptikti raštus bet kokioje apdorojamo atsako vietoje ir lokaliai, nekreipiant dėmesio į tolimesnius regionus. Tarkime, X yra atsakas, o H – jo lokalioji reprezentacija konvoliuciniame sluoksnyje. Sąsūkos operacija atliekama pagal formulę:

$$H_i = \sum_{n=-v}^v V_n X_{i+n}. \quad (3.27)$$

Čia V yra sąsūkos branduolys su $2v + 1$ elementu, dar vadinamas filtru. Jo elementų skaičius vadinamas filtro dydžiu. Konvoliuciniuose sluoksniuose dažniausiai būna keli branduoliai, jie randami tinklo apmokymo metu. Kalbant paprastai, sąsūkos operacija atliekama taip (3.1 pav.): imamas sąsūkos branduolys, kuris uždedamas ant atsako kairėje pusėje. Tada branduolys paslenkamas iš kairės pusės į dešinę po vieną tašką. Kiekvieną kartą paslinkus branduolį, vaizdo elementų reikšmės, kurios patenka į sąsūkos branduolį, sudauginamos su branduolio reikšmėmis po vieną elementą atskirai, tada šių sandaugų rezultatai sudedami ir gaunama viena skaitinė reikšmė. Šios reikšmės ir sudaro lokaliąją atsako reprezentaciją, kuri yra netiesinė ir kurią tinklas gali rasti pats.



3.1 pav. Vienmatės sąsūkos operacija. Šviesiai mėlyna – apdorojamas vaizdas; tamsiai mėlyna – sąsūkos branduolys; žalia – lokalioji vaizdo reprezentacija
Fig. 3.1. One-dimensional convolution. Light blue -- the signal being processed; dark blue -- convolution kernel; green -- a local signal representation

Sąsūkos operacija gerai tinka aptikti lokalesiems atsako požymiams, tačiau jos nepakanka, jei siekiama atlikti prognozes visam atsakui. Tai pasiekti galima naudojant sutelkimo sluoksnius (angl. *pooling layers*). Jie neuroniniam tinklui leidžia išmokti globaliąją atsako reprezentaciją, mažinant lokaliųjų reprezentacijų erdvinę raišką ir taip sutelkiant informaciją apie lokaliuosius požymius. Dar vienas sutelkimo sluoksnių privalumas yra tas, kad jie sumažina konvoliucinių sluoksnių jautrumą lokalesiems požymiams. Sutelkimo operacija vykdoma panašiai kaip ir sąsūka, tik vietoje sąsūkos branduolio naudojamas fiksuoto $2v + 1$ dydžio telkinys, kuris slenkamas per visą vaizdo įvestį. Kiekviename žingsnyje skaičiuojamas visų vaizdo elementų, kurie patenka į telkinį, vidurkis:

$$H_i = \frac{1}{(2v + 1)^2} \sum_{n=-v}^v X_{i+n} \quad (3.28)$$

arba maksimumas:

$$H_i = \max_{n \in [-v, v]} X_{i+n}. \quad (3.29)$$

3.2. Skirtingų kraštinių sąlygų režimų analizė

Šiame skyrelyje aprašomi biologinių jutiklių tikslumo bandymai duomenims be triukšmo ir su adityviu bei multiplikatyviu triukšmu, esant skirtingiems kraštinių sąlygų režimams – pastoviam ir srauto injekcijos. Tyrimo tikslas – patikrinti triukšmo ir kraštinių sąlygų režimo įtaką; išbandyti kubo mažinimo procedūrą ir nustatyti, ar ji gali padidinti neuroninio tinklo tikslumą atvirkštinio biologinių jutiklių uždavinio sprendimui.

Modelyje yra 3 substratų koncentracijos – tai mažiausias substratų kiekis, kuriam gaunamos didelės paklaidos triukšmingiems atsakams. Kiekvienos koncentracijos reikšmė yra tarp 3.2 ir 12.8. Apmokymo duomenų koncentracijos parenkamos paėmus apręžtą kubą $C = [3,2; 12,8]^3$ koncentracijų erdvėje, jame generuojant tolygų diskretų trimatį tinklą. Jo kraštai apibrėžiami kaip aibė D (3.30), kurioje yra $N = 22$ diskrečiųjų taškų tarp koncentracijų rėžių:

$$D = \left\{ 3,2 \left(1 - \frac{k}{N-1} \right) + \frac{k}{N-1} 12,8 : k = \overline{0, N-1} \right\}. \quad (3.30)$$

Paskui kiekvienas tinklo taškas naudojamas biologinio jutiklio atsakams generuoti. Testavimo rinkiniui generuoti imamas tam tikras skaičius taškų, atsitiktinai sugeneruotų koncentracijų kube C , ir jie naudojami atsakams generuoti kaip ir anksčiau. Apmokymo rinkinį sudaro 10468 atsakų, o testavimo – 3000.

Tyrimams naudotos tokios (1.3)–(1.5) lygčių sistemos parametrų reikšmės: fermentinio sluoksnio storis $d = 0,02$ ir maksimalūs fermentinės reakcijos greičiai $V_i = 5 \cdot 10^{i-9}$, $i = 1,2,3$. Kiti parametrai pateikiami lentelėje. Numatytasis simuliacijos laikas $T = 400$, laiko žingsnis $\tau = 0,25$, atsako srovės reikšmės išvedamos kas sekundę. Erdvės intervalą sudaro $N = 1000$ taškų, iš jų 333 priklauso fermentiniam sluoksniui. Naudojami pastovus ir srauto injekcijos analizės režimai – pastarajam injekcijos laikas yra $t_f = 3$, o simuliacijos laikas pailgintas iki $T = 1200$. Duomenų generavimo kodas parašytas C++ programavimo kalba, lygiagretiesiems skaičiavimams naudota MPI biblioteka VILNIUS TECH Vilkas klasteryje su 8 procesoriaus branduoliais. Vieno duomenų rinkinio generavimas užtruko apie 30 min. Naudojant VILNIUS TECH Vanagas klasterį, generavimui prireikė 2.5 min. Neuroniniai tinklai realizuoti *Python* programavimo kalba su *TensorFlow* ir *Keras* paketais su *Nvidia Tesla P100* grafiniu procesoriumi.

Tyrimams naudojamo trisluoksnio tinklo konfigūracija tokia: įvesčiai buvo naudotos 4 pagrindinės komponentės, paskui jų skaičius padidintas iki 80. Konfigūracija ir hiperparametrai parinkti koordinatinės paieškos metodu, kol maksimali testavimo paklaida nustojo mažėti. Buvo rinktasi tarp 1 ir 2 paslėptųjų sluoksnių, o neuronų skaičius juose parinktas iš penkių skirtingų reikšmių: 18, 28, 32, 36 ir 46. Testų metu nustatyta, kad neuroninis tinklas su vienu paslėptuoju sluoksniu ir 32 neuronais turėjo geriausią paklaidą. Paslėptojo sluoksnio aktyvavimo funkcija

yra hiperbolinis tangentas, o išvesties – tiesinė. Tinklas buvo apmokytas taikant Adam algoritmą esant apmokymo greičiui $1,25 \cdot 10^{-4}$, partijos dydis 64. Epochų skaičius priklauso nuo triukšmo lygio. Trisluoksniu tinklo apmokymas ilgiausiai užtrunka apie 10 min., esant pastoviajam režimui, ir apie 8 min., esant injekcijos režimui.

3.1 lentelė. Kiti modelio parametrai

Table 3.1. Other model parameters

Parametras	Reikšmė
$D_{S_{i,e}}$	$3 \cdot 10^{-6}$
$D_{P_{i,e}}$	$3 \cdot 10^{-6}$
$D_{S_{i,b}}$	$6 \cdot 10^{-6}$
$D_{P_{i,b}}$	$6 \cdot 10^{-6}$
K_i	10^{-4}
a	0,04
n_i	2

3.2 lentelė. Konvoliucinių neuroninių tinklų konfigūracijos. Stulpelis Sluoksn. nurodo, kuris sluoksnis turi nurodytus parametrus, Telk. dydis – telkinio dydis. Stulpelyje FC pateikti visiškai jungaus sluoksniu neuronų kiekiai

Table 3.2. Convolutional neural network configurations. The columns indicate the network configuration (1), number of kernels (2) in specified layers (3), kernel size (4) in specified layers (5), pool size (6) in specified layers (7). The FC column (8) is the amount of neurons in the fully connected layer

Tinklas	Filtrų skaičius	Sluoksn.	Filtro dydis	Sluoksn.	Telk. dydis	Sluoksn.	FC
1	10	1;2	4	4	2	1;2;3;4	36
	12	3;4	6	3			
			8	2			
			10	1			
2	10	1	10	1;2;3;4	2	1;2;3;4	18
	12	2					
	14	3					
	16	4					
3	10	1;2;3;4	6	1;2;3;4	2	1;2;3;4	36
4	8	3;4	4	4	2	2;3;4	36
	10	1;2	5	1;2;3	3	1	
5	8	2;3;4	4	4	2	2;3;4	36
	10	1	5	3	3	1	
			6	1;2			

Naudojamo konvoliucinio tinklo pagrindą sudaro keturios vienmatės konvoliucijos–sutelkimo sluoksnių poros; po jų eina visiškai jungus sluoksniš. Konfigūracija ir hiperparametrai parinkti koordinatinės paieškos metodu. Buvo naudotos penkios tinklų konfigūracijos, pateikiamos 3.2 lentelėje. Siekiant išvengti persimokymo, tarp visiškai jungaus ir išvesties sluoksnių yra atmetimo sluoksniš; atmetimo tikimybė 0,1. Konvoliuciniuose ir visiškai jungiame sluoksniuose taikyta ReLU aktyvavimo funkcija. Tinklui apmokyti naudotas Adam algoritmas, kai apmokymo greitis $5 \cdot 10^{-4}$ ir partijos dydis 64. Tinklas apmokytas tris kartus ir išrinktas variantas su mažiausia paklaida. Vieno tinklo apmokymas užtrunka apie 42 min., esant pastoviam režimui, ir apie 30 min., esant injekcijos režimui. Skirtumas tarp apmokymo laikų skirtingiems vienodo tipo tinklams esant tam pačiam analizės režimui nedidelis.

3.2.1. Maksimaliosios ir vidutinės paklaidų analizė. Atkuriamųjų parametru srities mažinimas.

Atliktuose tyrimuose svarbiausia, kiek vidutiniškai tiksliai nustatomos koncentracijos iš atsako, todėl neuroninis tinklas laikomas pakankamai tikslu, jei vidutinė paklaida bet kokiems atsakams bus maža. Neuroninio tinklo tikslumas buvo matuotas naudojant vidutinę ir maksimaliąją santykinę procentinę paklaidą, atitinkamai $E_{ds,a}$ ir $E_{ds,m}$:

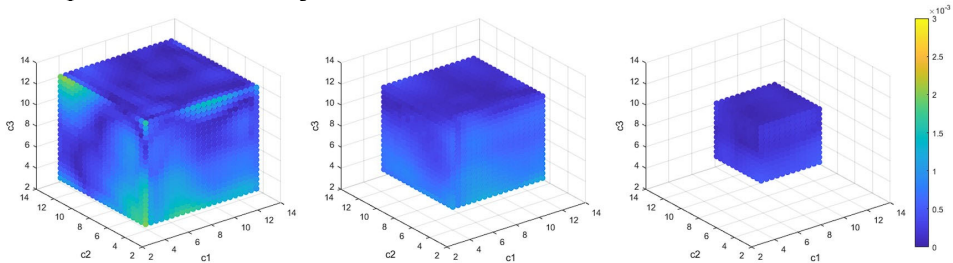
$$E_{ds,a} = \frac{1}{3N_{ds}} \sum_{j=1}^{N_{ds}} \sum_{i=1}^3 \frac{|a_i(I_{ds}^{(j)}) - c_{i,ds}^{(j)}|}{|c_{i,ds}^{(j)}|} \times 100\%, \quad (3.31)$$

$$E_{ds,m} = \max_{\substack{i=1,3; \\ j=1, N_{ds}}} \frac{|a_i(I_{ds}^{(j)}) - c_{i,ds}^{(j)}|}{|c_{i,ds}^{(j)}|} \times 100\%, \quad (3.32)$$

čia indeksas ds nurodo, koks duomenų rinkinys naudojamas, N_{ds} – rinkinyje esančių atsakų skaičius, $a_i(I_{ds}^{(j)})$ yra i -tosios koncentracijos reikšmė j -tajam duomenų rinkinio atsakui $I_{ds}^{(j)}$, prognozuojama neuroninio tinklo, o $c_{i,ds}^{(j)}$ yra tikrosios j -tojo atsako koncentracijų reikšmės.

Nagrinėjant biologinio jutiklio prognozavimo paklaidų apmokymo rinkiniui pasiskirstymą koncentracijų erdvėje, buvo aptiktas įdomus reiškinyš (3.2 pav.). Paveiksle kiekvienas taškas nuspalvintas pagal jį atitinkančio atsako maksimaliąją paklaidą visoms trimis koncentracijoms. Matome, kad didžiausios paklaidos yra ant originalaus apmokymo kubo krašto. Todėl disertacijos autorius sukūrė naują procedūrą triukšmingų atsakų įtakai biologinių jutiklių tikslumui mažinti. Jei pa-

šalinami kubo sluoksniai po 3 iš visų pusių, taip sumažinant matavimo sritį, išvengiama paklaidų ant krašto ir didžiausios paklaidos sumažėja. Ši procedūra leidžia aptikti tas koncentracijų reikšmes, kurioms biologinis jutiklis veikia geriausiai. Ja naudojantis, buvo atlikti bandymai testavimo rinkiniui, kai apribojamos galimos neuroninio tinklo prognozes mažesnėje atkuriamųjų koncentracijų srityje, ir joje generuojamas testavimo rinkinys, taip pat, kaip rašyta anksčiau. Apmokymo, testavimo (tc) ir sumažintos koncentracijų srities (rc) duomenų rinkinių ribos pateiktos 3.3 lentelėje.



3.2 pav. Biologinio jutiklio paklaidų pasiskirstymas. Kairėje – originalus apmokymo kubas, viduryje – pašalinti 3 sluoksniai, dešinėje – pašalinti 6 sluoksniai

Fig. 3.2. Distribution of biosensor prediction errors. The original training cube is on the left, in the middle – 3 layers were removed, and on the right – 6 layers were removed

3.3 lentelė. Biologinio jutiklio duomenų rinkinių ribos ir dydžiai

Table 3.3. Biosensor dataset bounds and sizes

Rinkinys	Pradžia	Pabaiga	Atsakų skaičius
tc	3,2	12,8	10648
rc	5,6	10,4	3000

3.2.2. Rezultatai su adityviu triukšmu

Pradžioje bus pateikti rezultatai, kai signalus veikia adityvus triukšmas. Santykinės prognozavimo paklaidos trisluoksniui tinklui pateiktos 3.4 ir 3.5 lentelėse. Paklaidos testavimo rinkiniui originalioje atkuriamųjų koncentracijų srityje žymimos $E_{ts,a}$ ir $E_{ts,m}$, o sumažintoje srityje – $E_{rc,a}$ ir $E_{rc,m}$. Taip pat pateikti naudotų pagrindinių komponentių ir atliktų epochų skaičiai.

Pastoviojo režimo rezultatai rodo, kad kai neturime triukšmo, maksimaliosios paklaidos yra nedidelės ir jos pagerėja, kai mažinama atkuriamųjų koncentracijų sritis. Keičiant pagrindinių komponentių skaičių gaunami rezultatai panašūs. Triukšmingiems signalams, kai imamos 4 pagrindinės komponentės, paklaidos smarkiai išauga ir yra maždaug pastovios triukšmo lygio atžvilgiu. Maksimalioji

paklaida net po atkuriamų koncentracijų srities sumažinimo yra gana didelė. Todėl siūlomas sprendimas paimti daugiau pagrindinių komponentių – geriausi rezultatai gaunami, kai jų imama 80. Kodėl paklaidos pagerėja paėmus tokį didelį PK skaičių, pasakyti sunku, nes teoriškai, tolimesnių PK reikšmės smarkiai mažėja ir jos turi vis mažesnę įtaką įvesties duomenims. Tačiau neuroninis tinklas sugeba atskirti signalus, panaudodamas minėtas komponentes. Gali būti, kad tinklas apmokymo metu padidina tolimesnių PK svorius taip, kad triukšmingi signalai būtų atskiriami lengviau. Dar pabrėžtina, kad toks PK parinkimas veikia tik tada, kai triukšmas pritaikomas tiek apmokymo, tiek testavimo duomenims, todėl yra galimybė, kad tinklas įsimeina triukšmus.

3.4 lentelė. Prognozavimo paklaidos esant pastoviam režimui

Table 3.4. Prediction errors for the batch mode

σ_n	Epochos	PK	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	3200	4	0,078	1,899	0,038	0,214
0	3200	80	0,050	1,093	0,039	0,269
0,01	2000	4	16,89	154,69	7,77	50,15
0,01	3200	80	1,04	9,88	0,92	6,64
0,02	3200	80	1,87	18,50	1,69	12,16
0,03	3200	80	2,66	26,62	2,38	17,52
0,04	3200	80	3,36	35,72	3,09	22,17
0,05	3200	80	4,08	39,75	3,77	26,50
0,10	3000	4	16,88	156,21	7,82	50,35
0,10	1000	80	8,12	107,02	7,20	52,22
0,20	450	80	12,54	163,91	8,77	74,70
0,40	3000	4	16,95	155,54	7,90	49,74
0,40	477	80	15,24	207,84	9,59	77,95

3.5 lentelė. Prognozavimo paklaidos esant srauto injekcijos režimui

Table 3.5. Prediction errors for the flow injection mode

σ_n	Epochos	PK	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	2200	4	0,030	0,273	0,021	0,068
0	3200	80	0,039	0,443	0,022	0,108
0,01	2200	4	2,52	35,42	2,43	23,97
0,01	1000	80	2,70	38,95	2,59	26,24
0,02	1200	4	3,49	49,56	3,53	34,92
0,03	1500	4	4,27	62,90	4,41	43,62
0,04	2000	4	4,98	76,12	5,15	52,20

3.5 lentelės pabaiga

σ_n	Epochos	PK	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0,05	2000	4	5,61	86,36	5,81	60,08
0,10	2000	4	8,37	139,32	7,38	76,99
0,10	260	80	9,41	148,92	8,66	81,60
0,20	2000	4	13,54	175,08	7,55	75,60
0,40	2000	4	15,03	161,68	7,76	49,77
0,40	150	80	16,64	198,22	10,35	80,29

Kai naudojame 80 PK, maksimaliosios paklaidos triukšmingiems atsakams irgi padidėja smarkiai, bet šį kartą atkuriamų koncentracijų srities mažinimas jas pagerina. Pasiėkus 0,03 triukšmo lygį, nors maksimaliosios paklaidos ir sumažėja, jos lieka gana didelės – šią paklaidą atitinkantiems atsakams biologinis jutiklis nebegali patikimai nustatyti koncentracijų. Jei atkreiptume dėmesį į vidutines paklaidas, matytume, kad kubo mažinimas beveik neturi poveikio iki 0,2 triukšmo lygio, kai paklaidos pastebimai pagerėja.

Naudojantis srauto injekcijos režimu, rezultatai signalams be triukšmo rodo, kad visos palaidos yra mažesnės nei esant pastoviam režimui; kai turime triukšmingus atsakus, paklaidos yra didesnės. Atkuriamų koncentracijų srities mažinimas šiuo atveju veikia panašiai, kaip ir esant pastoviam režimui – nors maksimaliosios paklaidos sumažėja, jos vis tiek yra didelės. Tas galioja visiems triukšmo lygiams, tik esant dideliame triukšmui, pradeda mažėti ir vidutinė paklaida. Kitaip nei esant pastoviam režimui, su 4 PK gauname mažesnes paklaidas nei su 80 PK.

Biologinio jutiklio prognozavimo paklaidos, kai atsako analizei naudojami konvoliuciniai neuroniniai tinklai, pateiktos 3.6 lentelėje. Taip pat, kiekvienam triukšmo lygiui pateikiama ir naudota tinklo konfigūracija.

3.6 lentelė. Biologinio jutiklio prognozavimo paklaidos konvoliuciniams tinklams
Table 3.6. Prediction errors for convolutional networks

Pastovusis režimas						
σ_n	Konfig.	Epochos	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	1	990	0,392	4,420	0,261	1,812
0,01	1	936	2,09	25,75	1,69	13,70
0,05	1	979	5,01	72,97	4,50	38,08
Srauto injekcijos režimas						
σ_n	Konfig.	Epochos	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	3	340	0,095	0,945	0,111	0,470
0,01	5	277	1,25	17,35	1,16	12,19
0,05	5	393	3,25	52,10	3,07	38,97
0,10	7	200	6,03	101,26	5,48	65,81
0,40	7	200	12,74	182,75	8,24	88,29

Lyginant konvoliucinio ir trisluoksnių neuroninio tinklo rezultatus, matome, kad esant pastoviam režimui, nors testavimo paklaidos yra šiek tiek didesnės visiems triukšmo lygiams, praktiškai jos yra panašios, nes manome, jog konvoliucinio tinklo hiperparametrus galima parinkti geriau – taip, kad šio tinklo paklaidos būtų artimesnės trisluoksnių tinklo paklaidoms. Apibendrinant galima pasakyti, kad abiejoms architektūroms triukšmingų atsakų atpažinimas blogiausiai atveju nepatikimas, todėl galima teigti, kad pastovusis režimas jautrus triukšmams ir nėra praktiškas. Toks rezultatas skirtingoms tinklų architektūroms rodo, kad problema yra susijusi su matematinė atvirkštinio uždavinio specifika – blogu sąlygotumu.

Esant srauto injekcijos režimui, konvoliucinio tinklo paklaidos šiek tiek mažesnės nei trisluoksnių tinklui, todėl paskesnioji tinklo architektūra šiam režimui tinka geriau. Nors skirtumas nedidelis, disertacijos autoriaus nuomone, tai yra gerai, nes artėjame prie faktinio uždavinio tikslumo. Nagrinėjant paklaidas sumažintoje atkuriamų koncentracijų srityje, matome, kad jos mažinimas veikia panašiai kaip ir anksčiau, išskyrus tai, kad paklaidos sumažėja labiau, esant pastoviam režimui, nei esant srauto injekcijos režimui. Tai galioja ir dideliems triukšmams – jiems paklaidos irgi sumažėja labiau. Remiantis tuo, rekomenduojama rinktis mažesnę skirtumą tarp originalios ir sumažintos atkuriamų koncentracijų sričių paklaidų, nes tada jos mažesnės. Lyginant analizės režimus tarpusavyje, matome, kad maksimalios paklaidos sumažintoje atkuriamų koncentracijų srityje maždaug vienodos, o tai reiškia, kad jos mažinimui analizės režimas mažiau svarbus. Nepaisant to, kadangi vidutinės paklaidos srauto injekcijos režimui mažesnės, tai nepažeidžia teiginio dėl konvoliucinio tinklo taikymo šiam režimui.

3.2.3. Rezultatai su multiplikatyviu triukšmu

Toliau, 3.7–3.9 lentelėse, pateikiami tyrimų su multiplikatyviu triukšmu rezultatai.

Trisluoksnių neuroniniams tinklui ir pastoviam režimui, esant multiplikatyviu triukšmui, paklaidos yra žymiai didesnės nei esant adityviu triukšmui. Tai dar kartą patvirtina pastoviojo režimo jautrumą triukšmams. Kubo mažinimas veikia taip pat, kaip ir anksčiau: nors maksimaliosios paklaidos sumažėja, jos išlieka didelės ir dideliems triukšmams pradeda mažėti ir vidutinės paklaidos. Imant 80 pagrindinių komponentų gaunami geresni rezultatai negu imant 4 PK iki 0,02 triukšmo lygio, kai vidutinės paklaidos susilygina ir maksimalioji paklaida padidėja.

Esant srauto injekcijos režimui, multiplikatyvus triukšmas turi silpnesnę įtaką paklaidoms, kurios yra daug mažesnės nei esant pastoviam režimui. Atkuriamų koncentracijų srities mažinimas gerai tinka maksimaliosioms paklaidoms mažinti

iki priimtino lygio, o vidutinės paklaidos beveik nesikeičia. Paklaidos sumažėja didesniame triukšmo lygių intervale – iki 0,2.

3.7 lentelė. Prognozavimo paklaidos esant pastoviam režimui ir multiplikatyviam triukšmui, naudojant trisluoksnį neuroninį tinklą

Table 3.7. Batch mode prediction errors for FNN networks and multiplicative noise

σ_n	Epochos	PK	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	3200	4	0,078	1,899	0,038	0,214
0	3200	80	0,050	1,093	0,039	0,269
0,01	2200	4	5,34	82,11	5,17	42,39
0,01	1000	80	5,52	78,93	5,12	41,08
0,02	2200	4	9,64	128,39	8,88	71,27
0,02	700	80	9,61	136,20	8,50	67,72
0,03	2200	4	12,03	141,35	9,41	79,48
0,04	2200	4	13,48	155,49	9,51	86,14
0,05	2200	4	14,52	156,96	9,22	83,86

3.8 lentelė. Srauto injekcijos režimo prognozavimo paklaidos, kai naudojamas trisluoksnis tinklas ir yra multiplikatyvus triukšmas

Table 3.8. Prediction errors for the feedforward architecture and multiplicative noise

σ_n	Epochos	PK	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	2200	4	0,030	0,273	0,021	0,068
0	3200	80	0,039	0,443	0,022	0,108
0,01	2000	4	0,14	1,25	0,11	0,58
0,01	400	80	0,22	2,23	0,16	0,83
0,02	2000	6	0,23	2,11	0,21	1,17
0,03	2000	4	0,36	3,18	0,31	1,78
0,04	2000	4	0,47	4,09	0,41	2,41
0,05	2000	4	0,58	5,02	0,52	3,02
0,10	2000	4	1,11	9,34	1,03	5,88
0,20	2000	4	2,19	19,75	2,06	11,49
0,40	2000	4	4,15	41,12	4,11	22,91
0,80	2000	4	7,67	75,68	7,98	45,23

Kai naudojamas konvoliucinis neuroninis tinklas ir signalus veikia multiplikatyvus triukšmas, testavimo paklaidos pastoviam režimui yra daug didesnės nei naudojant trisluoksnį tinklą. Palyginus rezultatus srauto injekcijos režimui su tais, kai naudojamas trisluoksnis tinklas ir triukšmo lygis nedidelis, paklaidos yra šiek tiek didesnės, bet esant didesniems triukšmams, paklaidos pagerėja. Tai ga-

lioja ir paklaidoms sumažintoje atkuriamų koncentracijų srityje. Nors multiplikatyvus triukšmas labiau iškraipo signalą, rezultatai žymiai geresni. Manome, priežastis gali būti tai, kad triukšmas padeda atskirti atsakus, nes tada atsakai, kurie yra artimi vienas kitam, tampa mažiau panašūs. Išvados, gautos kai konvoliucinis tinklas naudotas duomenims su adityviu triukšmu, galioja ir šiam atvejui: paklaida sumažėja mažiau, jei paklaidos mažesnės, negu tada, kai paklaidos didesnės.

3.9 lentelė. Prognozavimo paklaidos esant multiplikatyviam triukšmui, kai naudojamas konvoliucinis neuroninis tinklas

Table 3.9. Prediction errors for the convolutional architecture and multiplicative noise

Pastovusis režimas						
σ_n	Konfig.	Epochos	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	1	990	0,392	4,420	0,261	1,812
0,01	2	459	6,13	82,67	6,01	45,89
0,05	2	131	14,83	180,28	9,80	94,08
Srauto injekcijos režimas						
σ_n	Konfig.	Epochos	$E_{ts,a}$	$E_{ts,m}$	$E_{rc,a}$	$E_{rc,m}$
0	3	340	0,095	0,945	0,111	0,470
0,01	4	335	0,26	2,25	0,19	1,40
0,05	3	312	0,64	6,81	0,59	3,99
0,10	5	200	1,25	10,00	1,16	7,26
0,40	5	240	4,19	34,23	3,97	24,33
0,80	4	308	6,01	51,47	5,91	37,86

Gauti rezultatai leidžia pateikti rekomendacijas, kurį tinklą naudoti geriau. Trisluoksnis tinklas geresnis už konvoliucinį, kai biologinio jutiklio analizės režimas pastovusis ir nėra triukšmo. Esant tam pačiam režimui, jei turime triukšmingus atsakus, abiem neuroninių tinklų tipams tikslumas panašus. Šis rezultatas nepriklauso nuo triukšmo tipo. Jei naudojame srauto injekcijos režimą ir kai triukšmas adityvus, konvoliucinis tinklas yra šiek tiek geresnis nei trisluoksnis. Kai triukšmas multiplikatyvus, geriausią neuroninio tinklo tipą nulemia triukšmo lygis: esant mažo lygio triukšmams, trisluoksnis tinklas tikslesnis, tačiau triukšmo lygiui augant, paklaidos konvoliuciniam tinklui pasidaro mažesnės nei trisluoksniam. Bendroju atveju, konvoliucinis tinklas turi papildomą pranašumą, kad jam nereikia atlikti rankinio signalo apdorojimo – tai tinklas atlieka pats, apmokymo metu.

3.3. DIRECT globaliosios optimizacijos metodo taikymas ir analizė

Skyrelyje pateikiamas DIRECT globaliosios optimizacijos metodas, jo šeimininko ir darbininko tipo lygiagretinimo schema ir taikymas atvirkštinio biologinio jutiklio uždaviniui spręsti.

3.3.1. Lygiagretusis DIRECT algoritmas

Kadangi atvirkštinio biologinių jutiklių uždavinio tikslo funkcija gali turėti daug ekstremumų, jai optimizuoti geriausiai tinka globaliosios optimizacijos metodai. Iš jų plačiausiai naudojami yra „*branch-and-bound*“ klasės algoritmai. Visų jų esmė – rekursyviai dalinti paieškos sritį į mažesnes dalis, ir jose įvertinti apatinį tikslo funkcijos rėžį. Jei matome, kad kurioje nors dalyje negali būti optimalių sprendinių, toliau jos nebedalijame ir nebenagrinėjame. Tolesniam svarstymui daroma prielaida, kad paieškos sritis yra hiperkubas. Šiuo principu paremti algoritmai garantuotai randa globalųjį optimumą, jei atliekama pakankamai daug iteracijų.

Apatinio rėžio įvertį patogiausia skaičiuoti kai optimizuojama funkcija $f(x)$ tenkina Lipšico sąlygą: bet kokiems $x_1, x_2 \in D$:

$$|f(x_1) - f(x_2)| \leq K \|x_1 - x_2\|, \quad (3.33)$$

čia K – Lipšico konstanta. Ją įvertinti galima tik palyginti paprastoms tikslo funkcijoms, o praktikoje neretai reikia optimizuoti žymiai sudėtingesnes funkcijas, pavyzdžiui, kai tikslo funkcijos reikšmės gaunamos iš diferencialinių lygčių sprendinių. Todėl Jones et al. (1993) pasiūlė DIRECT globaliosios optimizacijos metodą, kuriam nereikia skaičiuoti Lipšico konstantos. DIRECT metodas tikslo funkcijos reikšmes tikrina hiperkubų centro taškuose, nes taip efektyviausiai atliekami skaičiavimai – kiekvienai paieškos srities daliai reikšmės skaičiuojame tik vieną kartą. Tačiau tada svarbu užtikrinti, kad net ir atlikus dalijimo veiksmą, visuose gaunamuose hiperkubuose funkcijos reikšmės būtų taip pat tikrinamos centro taškuose, todėl paieškos sritis dalijama į tris dalis. Yra ir kitų DIRECT variantų: Stripinis et al. (2019) straipsnyje pateikiama algoritmo versija optimizavimui su apribojimais; Deng & Ferris (2007) straipsnyje DIRECT pritaikytas triukšmo veikiamoms tikslo funkcijoms. Daugiau DIRECT variantų galima rasti Jones & Martins (2021) apžvalgoje.

Algoritmas vykdomas taip: hiperkubo centrą žymėsime c_i , $i \in \mathbb{N}$, o $f(c_i)$ – tikslo funkcijos reikšmę jame; dst_i žymėsime c_i atstumą iki hiperkubo krašto. Iš pradžių imame pradinio hiperkubo centrinį tašką c_1 ir jame apskaičiuojame tikslo funkcijos reikšmę $f(c_1)$. Tada atliekame šiuos žingsnius:

1. Nustatome potencialiai optimalius stačiakampius S_{rect} . Tai padarome reprezentuodami juos kaip taškus ($dst_i, f(c_i)$) ir atvaizduodami juos diagramoje. Tada imame tuos stačiakampius, kurių reprezentacijos yra ant iškilijo diagramos apvalkalo apatinės dalies.
2. Visiems potencialiai optimaliems erdvės stačiakampiems vykdome 3–6 žingsnius.
3. Parenkame tas hiperkubo dimensijas I_{dims} , kuriose jo kraštai ilgiausi ir apskaičiuojame trečdalį šio ilgio Λ .
4. Apskaičiuojame $c_i \pm \Lambda e_j$, $j \in I_{dims}$ taškus hiperkube. Čia e_j yra baziniai erdvės vektoriai.
5. Skaičiuojame funkcijos reikšmes $f(c_i \pm \Lambda e_j)$.
6. Hiperkubą padalijame į tris dalis tose dimensijose, kuriose jo kraštai ilgiausi, pradėdami nuo dimensijos, kuriai $f(c_i \pm \Lambda e_j)$ mažiausia, tada dalijame dimensijoje su didžiausia $f(c_i \pm \Lambda e_j)$ reikšme.
7. Randame laikiną minimumo tašką ir laikiną funkcijos minimumą. Vykdomė 2–6 žingsnius tol, kol pasiekiamė maksimalų iteracijų skaičių.

Norėdami rasti tikslo funkcijos minimumą minėtais metodais, turime atlikti daug skaičiavimų, todėl disertacijoje naudojame lygiagrečiąją metodo versiją, aprašytą Watson & Baker (2001) straipsnyje. Algoritmas lygiagretinamas šeimininko ir darbininko principu. Visus algoritmo žingsnius atlieka šeimininkas, išskyrus 5 žingsnį, kai skaičiuojame funkcijų reikšmes taškuose – šį žingsnį atlieka darbininkai, kai šeimininkas jiems nusiunčia 4 žingsnyje gautus taškus. Šeimininkas sudaro uždavinių sąrašą ir jame esančius uždavinius paskirsto darbininkams paeiliui. Darbininkai vykdo darbus, o kai darbas baigiamas, šeimininkui siunčiamas signalas, kurį gavus, skiriamas naujas darbas darbininkui.

3.3.2. DIRECT taikymas biologinio jutiklio atvirkštiniam uždaviniui spręsti

Disertacijoje atliktas naujas atvirkštinio biologinių jutiklių uždavinio sprendimo tyrimas, kai, naudojant lygiagretųjį DIRECT algoritmą, buvo ieškomas vienas atsakas, atitinkantis koncentracijų trejetą (4,5714; 10,0571; 3,2) esant pastoviam ir srauto injekcijos režimams. Tyrimo tikslas – patikrinti, ar esant srauto injekcijos režimui biologinio jutiklio tikslumas didesnis nei esant pastoviam, kai naudojame DIRECT ir įvertinti algoritmo išplečiamumą. Signalai buvo be triukšmo. Siekiant gauti priimtinius rezultatus, reikėjo susiaurinti paieškos sritį iki kubo $R = [3,4; 6] \times [7; 12, 6] \times [3,2; 6]$. Skaičiavimai atlikti VILNIUS TECH (Vilniaus Gedimino technikos universitetas) Vanagas klasteryje. Rezultatai pateikiami 3.10 ir 3.11 lentelėse. Pateikiamos maksimaliosios procentinės paklaidos iš visų trijų koncentracijų.

3.10 lentelė. Prognozavimo paklaidos DIRECT algoritmui
Table 3.10. Prediction errors for the DIRECT algorithm

	Pastovusis režimas		Srauto injekcija	
	CNN	DIRECT	CNN	DIRECT
Paklaida	1,83	7,39	0,229	1,53

3.11 lentelė. Išplečiamumo rezultatai DIRECT algoritmui. p yra procesų darbininkų skaičius (visais atvejais vienas procesorius buvo šeimininkas), $S_p^{(par)}$ yra pagreitinimas ir $E_p^{(par)}$ yra efektyvumas

Table 3.11. Scalability results for the DIRECT algorithm. p is the number of worker processes (all cases also had one master process), $S_p^{(par)}$ is the speedup and $E_p^{(par)}$ is the efficiency

p	$T_p^{(par)}$	$S_p^{(par)}$	$E_p^{(par)}$
1	1109	–	–
9	155,17	7,15	0,79
19	90,26	12,29	0,65

Lentelėse matome, kad DIRECT algoritmas veikia geriau, kai turime srauto injekcijos režimą, kaip ir tikėjomės. Be to, esant srauto injekcijos režimui algoritmas sprendinį rado greičiau: atlikome 50 iteracijų, o esant pastoviam režimui reikėjo atlikti 100 iteracijų. Padidinus iteracijų skaičių iki 200 paklaida beveik nepasikeitė. Taip pat matome, kad konvoliucinio tinklo paklaida mažesnė už DIRECT metodo. Tai galima paaiškinti tuo, kad DIRECT, per tą patį laiką kaip konvoliucinio tinklo apmokymas, nesugeba rasti globaliojo minimumo, nes metodas užima daug laiko tikrindamas potencialiai optimalias sritis kur iš tikrųjų globaliojo minimumo nėra. Taip būna, kai tikslo funkcija turi daug lokaliųjų ekstremumų, kaip ir atvirkštinio biologinių jutiklių uždavinio atveju. Nors DIRECT atlieka globaliąją optimizaciją, jis negarantuoja, kad minimumas bus surastas greitai – tai galima padaryti tik per pakankamai ilgą laikotarpį.

Nagrinėkime išplečiamumo rezultatus: DIRECT algoritmui pagreitinimas mažesnis už atitinkamą procesorių skaičių p ir efektyvumas krenta didėjant p . Tai susiję su tuo, kaip veikia DIRECT: kiekvienoje iteracijoje generuojamas kintamas užduočių skaičius, kuris paskirstomas procesams. Kadangi naudojamas mažas iteracijų skaičius, gaunamas ir mažas užduočių skaičius ir negalime iki galo panaudoti turimų procesų. Dėl to, didinant jų skaičių mažėja efektyvumas. Gerai žinomas rezultatas, kad lygiagretiesiems algoritmams sunku garantuoti stabilų efektyvumą, kai fiksuoto dydžio uždavinį sprendžiame imdami vis daugiau procesorių. Ši problema sprendžiama daug efektyviau, kai fiksuotam procesorių skaičiui sprendžiamas vis didesnis apimtys uždavinys. Kadangi naudojamas mažas

fiksuotas iteracijų skaičius, gaunamas pirmasis atvejis: nepriklausomų uždavinių skaičius irgi yra fiksuotas ir nepavyksta efektyviai išnaudoti visus (vis didesnę jų kiekį) procesorius. Dėl to, didinant jų skaičių mažėja efektyvumas. Nepaisant to, disertacijoje laikoma, kad algoritmo efektyvumas yra aukštas ir jo išplečiamumas geras, nes procesorių skaičius pastovus – tokiems uždaviniams geriau, jei procesorių skaičius reguliuojamas dinamiškai, pagal užduočių skaičių, tada efektyvumas būtų aukštesnis.

3.4. Trečiojo skyriaus išvados

Toliau pateiktos pagrindinės trečiojo skyriaus išvados:

1. Atvirkštiniam uždaviniui spręsti buvo parinkti dviejų skirtingų architektūrų dirbtiniai neuroniniai tinklai: tiesioginio sklidimo su PCA ir konvoliucinį. Atlikus tikslumo tyrimus nustatyta, kad, esant adityviam triukšmui, tiesioginio sklidimo tinklas su PCA transformacija geriau tinka pastoviam režimui, o konvoliucinis tinklas – srauto injekcijos režimui.
2. Kai yra multiplikatyvus triukšmas, paklaidos srauto injekcijos režimui yra daug mažesnės nei pastoviam režimui.
3. Nustatyta, kad atkuriamų koncentracijų srities mažinimo procedūra sumažina maksimaliąsias paklaidas visiems triukšmo lygiams, bet kai triukšmas didelis, maksimaliosios paklaidos nebesumažėja iki priimtino lygio, tačiau tada sumažėja vidutinė paklaida. Tai reiškia, kad didžiausios paklaidos gaunamos koncentracijoms, kurios yra arti apmokymo rinkinio krašto.
4. Tyrimų rezultatai DIRECT metodui rodo, kad jis turi geresnę tikslumą esant srauto injekcijos režimui, nei esant pastoviam, kaip ir neuroninių tinklų atveju. Išplečiamumo rezultatai DIRECT algoritmui rodo, kad jo efektyvumas krenta didėjant p . Tai susiję su tuo, kad kiekvienoje iteracijoje procesams paskirstomas kintamas užduočių skaičius ir kad naudojama mažai iteracijų, todėl gaunama mažai užduočių ir negalima iki galo panaudoti turimų procesų.

Bendrosios išvados

Toliau pateiktos bendrosios disertacijos išvados.

1. Lygčių su TLEO sprendimo algoritmai garantuoja tikslumą, pakankamą praktiniams skaičiavimams. Naudojantis šiais algoritmais, sukurtas alternatyvus biologinio jutiklio modelis su trupmeninio laipsnio difuzija. Gaudami atsakai yra jautrūs laipsnio parinkimui.
2. Sudarytos lygiagrečiosios minėtų metodų versijos leidžia efektyviai spręsti didelės apimties nelokaliuosius uždavinius. Išplečiamumo tyrimai Grėjaus–Skoto lygties sprendimui rodo, kad metodų efektyvumas viršija 80 %, naudojant iki 16 procesorių.
3. Nagrinėtas atvirkštinis biologinių jutiklių uždavinys, kuris yra jautrus triukšmingiems atsakams. Uždavinio sprendimui sėkmingai pritaikyti dirbtiniai neuroniniai tinklai. Jų tyrimų teoriniai ir praktiniai rezultatai leidžia pateikti rekomendacijas biologinių jutiklių tikslumui gerinti.
4. Visais atvejais tikslumui gerinti tinka atkuriamų substratų leidžiamųjų koncentracijų reikšmių srities mažinimo procedūra. Esant žemo lygio triukšmams, ji padeda sumažinti maksimaliąsias atpažinimo paklaidas, o esant didelio lygio triukšmams – sumažina vidutines paklaidas.
5. Uždaviniui spręsti taip pat pritaikytas DIRECT globaliosios optimizacijos metodas – jį naudojant srauto injekcijos režimas tikslesnis nei pastovusis. Jo efektyvumas siekia apie 80 %, naudojant iki 10 procesorių, tačiau dėl algoritmo savybių, maksimalų efektyvumą pasiekti sunkiau.

Literatūra ir šaltiniai

Acquarelli, J., van Laarhoven, T., Gerretzen, J., Tran, T. N., Buydens, L. M., & Marchiori, E. (2017). Convolutional neural networks for vibrational spectroscopic data analysis. *Analytica Chimica Acta*, 954: 22–31. <https://doi.org/10.1016/j.aca.2016.12.010>

Asadollahi-Baboli, M., & Mani-Varnosfaderani, A. (2014). Rapid and simultaneous determination of tetracycline and cefixime antibiotics by mean of gold nanoparticles-screen printed gold electrode and chemometrics tools. *Measurement*, 47: 145–149. <https://doi.org/10.1016/j.measurement.2013.08.029>

Aster, R. C., Borchers, B., & Thurber, C. H. (2018). *Parameter Estimation and Inverse Problems (Third Edition)*. Elsevier. <https://doi.org/10.1016/B978-0-12-804651-7.00015-8>

Bachmann, T. T., Leca, B., Vilatte, F., Marty, J. L., Fournier, D., & Schmid, R. D. (2000). Improved multianalyte detection of organophosphates and carbamates with disposable multielectrode biosensors using recombinant mutants of drosophila acetylcholinesterase and artificial neural networks. *Biosensors and Bioelectronics*, 15: 193–201. [https://doi.org/10.1016/S0956-5663\(00\)00055-5](https://doi.org/10.1016/S0956-5663(00)00055-5)

Baronas, R., Ivanauskas, F., & Kulys, J. (2021). *Mathematical modeling of biosensors*. Springer. <https://doi.org/10.1007/978-3-030-65505-1>

Baronas, R., Ivanauskas, F., Maslovskis, R., & Vaitkus, P. (2004). An analysis of mixtures using amperometric biosensors and artificial neural networks. *Journal of mathematical chemistry*, 36(3): 281–297. <https://doi.org/10.1023/B:JOMC.0000044225.76158.8e>

- Baronas, R., Kulys, J., Žilinskas, A., Lančinskas, A., & Baronas, D. (2013). Optimization of the multianalyte determination with biased biosensor response. *Chemometrics and Intelligent Laboratory Systems*, 126: 108–116. <https://doi.org/10.1016/j.chemolab.2013.05.003>
- Baronas, R., Kulys, J., Lančinskas, A., & Žilinskas, A. (2014). Effect of diffusion limitations on multianalyte determination from biased biosensor response. *Sensors*, 14(3): 4634–4656. <https://doi.org/10.3390/s140304634>
- Barroso, T. G., Martins, R. C., Fernandes, E., Cardoso, S., Rivas, J., & Freitas, P. P. (2018). Detection of bcg bacteria using a magnetoresistive biosensor: A step towards a fully electronic platform for tuberculosis point-of-care detection. *Biosensors and Bioelectronics*, 100: 259–265. <https://doi.org/10.1016/j.bios.2017.09.004>
- Bonito, A., & Pasciak, J. E. (2015). Numerical approximation of fractional powers of elliptic operators. *Mathematics of Computation*, 84: 2083–2110. <https://doi.org/10.1090/S0025-5718-2015-02937-8>
- Bougadi, E. T., & Kalogianni, D. P. (2020). Paper-based dna biosensor for food authenticity testing. *Food Chemistry*, 322: 126758. <https://doi.org/10.1016/j.foodchem.2020.126758>
- Bueno-Orovio, A., Kay, D., Grau, V., Rodriguez, B., & Burrage, K. (2014). Fractional diffusion models of cardiac electrical propagation: role of structural heterogeneity in dispersion of repolarization. *Journal of The Royal Society Interface*, 11, pp. 1–12. <https://doi.org/10.1098/rsif.2014.0352>
- Carpinteri, A., & Sapora, A. (2010). Diffusion problems in fractal media defined on cantor sets. *ZAMM*, 90: 203–210. <https://doi.org/10.1002/zamm.200900376>
- Cartas, R., Mimendia, A., Legin, A., & del Valle, M. (2010). Two analyte calibrations from the transient response of a single potentiometric sensor employed with the SIA technique. *Talanta*, 80: 1428–1435. <https://doi.org/10.1016/j.talanta.2009.09.048>
- Chen, B. M., & Kojouharov, H. V. (1999). Non-standard Numerical Methods Applied to Subsurface Biobarrier Formation Models in Porous Media. *Bulletin of Mathematical Biology*, 61: 779–798. <https://doi.org/10.1006/bulm.1999.0113>
- Coates, A., Baumstarck, P., Le, Q., & Ng, A. Y. (2009). Scalable learning for object detection with GPU hardware. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4287–4293). <https://doi.org/10.1109/IROS.2009.5354084>
- Cui, F., Yue, Y., Zhang, Y., Zhang, Z., & Zhou, H. S. (2020). Advancing biosensors with machine learning. *ACS Sensors*, 5: 3346–3364. <https://doi.org/10.1021/acssensors.0c01424>
- Deng, G., & Ferris, M. C. (2007). Extension of the direct optimization algorithm for noisy functions. In *Proceedings of the 39th Conference on Winter Simulation: 40 Years! The Best is yet to Come, WSC '07* (pp. 497–504). IEEE Press.
- Deng, S., & Hwang, Y. (2007). Solution of inverse heat conduction problems using Kalman filter-enhanced Bayesian back propagation neural network data fusion. *International Journal of Heat and Mass Transfer*, 50(11-12): 2089–2100. <https://doi.org/10.1016/j.ijheatmasstransfer.2006.11.019>

Dock, E., Christensen, J., Olsson, M., Tønning, E., Ruzgas, T., & Emnéus, J. (2005). Multivariate data analysis of dynamic amperometric biosensor responses from binary analyte mixtures-application of sensitivity correction algorithms. *Talanta*, 65: 298–305. <https://doi.org/10.1016/j.talanta.2004.07.002>

Dzemyda, G., Šaltenis, V. ir Tiešis, V. (2007). *Optimizavimo metodai*. Mokslo aidai.

Feng, Y., Kayode, O., & Harper, W. F. (2013). Using microbial fuel cell output metrics and nonlinear modeling techniques for smart biosensing. *Science of The Total Environment*, 449: 223–228. <https://doi.org/10.1016/j.scitotenv.2013.01.004>

Ferentinos, K. P., Yialouris, C. P., Blouchos, P., Moschopoulou, G., & Kintzios, S. (2013). Pesticide residue screening using a novel artificial neural network combined with a bioelectric cellular biosensor. *BioMed Research International*, vol. 2013, pp. 1–8. <https://doi.org/10.1155/2013/813519>

Ghosh, S., Pratihari, D., Maiti, B., & Das, P. (2011). Inverse estimation of location of internal heat source in conduction. *Inverse Problems in Science and Engineering*, 19(3): 337–361. <https://doi.org/10.1080/17415977.2011.551876>

Gonzalez-Navarro, F., Stilianova-Stoytcheva, M., Renteria-Gutierrez, L., Belanche-Muñoz, L., Flores-Rios, B., & Ibarra-Esquer, J. (2016). Glucose oxidase biosensor modeling and predictors optimization by machine learning methods. *Sensors*, 16: 1483. <https://doi.org/10.3390/s16111483>

Google Brain Team (2021). *TensorFlow*. <https://www.tensorflow.org> [žiūrėta 2021-11-22].

Harizanov, S., Lazarov, R., Margenov, S., & Marinov, P. (2020). Numerical solution of fractional diffusion–reaction problems based on BURA. *Computers & Mathematics with Applications*, 80: 316–331. <https://doi.org/10.1016/j.camwa.2019.07.002>

Harizanov, S., Lazarov, R., Margenov, S., Marinov, P., & Vutov, Y. (2018). Optimal solvers for linear systems with fractional powers of sparse SPD matrices. *Numerical Linear Algebra with Applications*, 25: e2167. <https://doi.org/10.1002/nla.2167>

Hofreither, C. (2020). A unified view of some numerical methods for fractional diffusion. *Computers & Mathematics with Applications*, 80: 332–350. <https://doi.org/10.1016/j.camwa.2019.07.025>

Hofreither, C. (2021). An algorithm for best rational approximation based on barycentric rational interpolation. *Numerical Algorithms*, 88: 365–388. <https://doi.org/10.1007/s11075-020-01042-0>

Huang, L., Xiang, L., Zhang, Y., Wang, Y., & Nie, Z. (2020). Simultaneous quantitative analysis of k^+ and tl^+ in serum and drinking water based on uv–vis spectra and chemometrics. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 238: 118392. <https://doi.org/10.1016/j.saa.2020.118392>

Hundsdoerfer, W., & Verwer, J. (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer. <https://doi.org/10.1007/978-3-662-09017-6>

Čiegis, R. (2005). *Lygiagretieji algoritmai ir tinklines technologijos*. Technika.

Ilic, M., Liu, F., Turner, I., & Anh, V. (2006). Numerical Approximation of a Fractional-InSpace Diffusion Equation (II) – with Nonhomogeneous Boundary Conditions. *Fractional Calculus and Applied Analysis*, 9: 333–349.

Intel (2022). *Intel® Xeon® Processor E5-2630 v4 25M Cache 2.20 GHz Product Specifications*. <https://ark.intel.com/content/www/us/en/ark/products/92981/intel-xeon-processor-e52630-v4-25m-cache-2-20-ghz.html> [žiūrėta 2022-12-13].

Johnson, S. G., & Frigo, M. (2007). A modified split-radix FFT with fewer arithmetic operations. *IEEE Transactions on Signal Processing*, 55: 111–119. <https://doi.org/10.1109/TSP.2006.882087>

Jones, D. R., & Martins, J. R. R. A. (2021). The direct algorithm: 25 years later. *Journal of Global Optimization*, 79(3): 521–566. <https://doi.org/10.1007/s10898-020-00952-6>

Jones, D. R., Perttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79: 157–181. <https://doi.org/10.1007/BF00941892>

Jornet, M. (2021). Modeling of Allee effect in biofilm formation via the stochastic bistable Allen–Cahn partial differential equation. *Stochastic Analysis and Applications*, 39: 22–32.

Keras Team (2021). *Keras: the Python deep learning API*. <https://keras.io>. [žiūrėta 2021-11-22]. <https://doi.org/10.1080/07362994.2020.1777163>

Kim, G., Moon, J. H., & Morgan, M. (2013). Multivariate data analysis of impedimetric biosensor responses from salmonella typhimurium. *Analytical Methods*, 5: 4074–4080. <https://doi.org/10.1039/c3ay40256h>

Kriauciene, R., Bugajev, A., & Čiegis, R. (2020). A three-level parallelisation scheme and application to the Nelder–Mead algorithm. *Mathematical Modelling and Analysis*, 25: 584–607. <https://doi.org/10.3846/mma.2020.12139>

Kumar, M. H., Vishweshwara, P., & Gnanasekaran, N. (2020). Evaluation of artificial neural network in data reduction for a natural convection conjugate heat transfer problem in an inverse approach: experiments combined with cfd solutions. *Sadhan a*, 45(1): 1–15. <https://doi.org/10.1007/s12046-020-1303-x>

Lee, H. G. (2018). A second-order operator splitting Fourier spectral method for fractional-in-space reaction–diffusion equations. *Journal of Computational and Applied Mathematics*, 333: 395–403. <https://doi.org/10.1016/j.cam.2017.09.007>

Petkevičius L. (2023). *Dirbtinio intelekto, giliojo/mašininio mokymo sąvokų žodynelis*. <https://klevas.mif.vu.lt/~linp/page/savokos.html> [žiūrėta 2023-04-25].

Litvinas, L. ir Baronas, R. (2014). Medžiagų koncentracijų tirpalų mišiniuose nustatymas, taikant biojutiklius ir dirbtinius neuroninius tinklus. *Lietuvos matematikų draugijos darbai, ser. B*, 55: 78–83.

Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63: 208–228. <https://doi.org/10.1137/19M1274067>

Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC Press. <https://doi.org/10.1201/b17476>

- Martynko, E., & Kirsanov, D. (2020). Application of chemometrics in biosensing: A brief review. *Biosensors*, 10(8): 100. <https://doi.org/10.3390/bios10080100>
- Mazafi, A., Shukla, R. P., Shukla, S. K., Lavon, A., Ben-Yoav, H., Silue, A., McDowell, B., & Peixoto, N. (2018). Intelligent multi-electrode arrays as the next generation of electrochemical biosensors for real-time analysis of neurotransmitters. *MeMeA 2018 - 2018 IEEE International Symposium on Medical Measurements and Applications, Proceedings*, pp. 1-5. <https://doi.org/10.1109/MeMeA.2018.8438720>
- Medina-Plaza, C., García-Hernández, C., de Saja, J. A., Fernández-Escudero, J. A., Barajas, E., Medrano, G., García-Cabezón, C., Martín-Pedrosa, F., & Rodríguez-Mendez, M. L. (2015). The advantages of disposable screen-printed biosensors in a bioelectronic tongue for the analysis of grapes. *LWT - Food Science and Technology*, 62: 940–947. <https://doi.org/10.1016/j.lwt.2015.02.027>
- Metzler, R., & Klafter, J. (2004). The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics. *J. Phys. A: Math. Gen.*, 37: 161–208. <https://doi.org/10.1088/0305-4470/37/31/R01>
- Miranda, O. R., Chen, H. T., You, C. C., Mortenson, D. E., Yang, X. C., Bunz, U. H., & Rotello, V. M. (2010). Enzyme-amplified array sensing of proteins in solution and in biofluids. *Journal of the American Chemical Society*, 132: 5285–5289. <https://doi.org/10.1021/ja1006756>
- Mishra, R. K., Alonso, G. A., Istamboulie, G., Bhand, S., & Marty, J. L. (2015). Automated flow based biosensor for quantification of binary organophosphates mixture in milk using artificial neural network. *Sensors and Actuators B: Chemical*, 208: 228–237. <https://doi.org/10.1016/j.snb.2014.11.011>
- Mozaffari, M. H., & Tay, L.-L. (2020). A review of 1d convolutional neural networks toward unknown substance identification in portable raman spectrometer, *arXiv preprint arXiv: 2006.10575*. <https://doi.org/10.48550/arXiv.2006.10575>
- Nakatsukasa, Y., Sète, O., & Trefethen, L. N. (2018). The AAA Algorithm for Rational Approximation. *SIAM Journal on Scientific Computing*, 40: A1494–A1522. <https://doi.org/10.1137/16M1106122>
- Nesakumar, N., Sethuraman, S., Krishnan, U. M., & Rayappan, J. B. B. (2015). Cyclic voltammetric acetylcholinesterase biosensor for the detection of captan in apple samples with the aid of chemometrics. *Analytical and Bioanalytical Chemistry*, 407: 4863–4868. <https://doi.org/10.1007/s00216-015-8687-1>
- Nochetto, R. H., Otárola, E., & Salgado, A. J. (2016). A PDE Approach to Space-Time Fractional Parabolic Problems. *SIAM Journal on Numerical Analysis*, 54: 848–873. <https://doi.org/10.1137/14096308X>
- Notay, Y. (2012). Aggregation-Based Algebraic Multigrid for Convection-Diffusion Equations. *SIAM Journal on Scientific Computing*, 34: A2288–A2316. <https://doi.org/10.1137/110835347>
- Pakravan, S., Mistani, P. A., Aragon-Calvo, M. A., & Gibou, F. (2021). Solving inverse-PDE problems with physics-aware neural networks. *Journal of Computational Physics*, 440: 110414. <https://doi.org/10.1016/j.jcp.2021.110414>

- Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45: 227–248. <https://doi.org/10.3102/1076998619872761>
- Pozrikidis, C. (2018). *The Fractional Laplacian*. Chapman and Hall/CRC. <https://doi.org/10.1201/9781315367675>
- Sabatelli, L., Keating, S., Dudley, J., & Richmond, P. (2002). Waiting time distributions in financial markets. *The European Physical Journal B - Condensed Matter*, 27: 273–275. <https://doi.org/10.1140/epjb/e20020151>
- Salamanca-Neto, C. A., Marcheafave, G. G., Scremin, J., Barbosa, E. C., Camargo, P. H., Dekker, R. F., Scarminio, I. S., Barbosa-Dekker, A. M., & Sartori, E. R. (2020). Chemometric-assisted construction of a biosensing device to measure chlorogenic acid content in brewed coffee beverages to discriminate quality. *Food Chemistry*, 315: 126306. <https://doi.org/10.1016/j.foodchem.2020.126306>
- Samarskii, A. A. (2001). *The Theory of Difference Schemes (1st ed.)*. CRC Press. <https://doi.org/10.1201/9780203908518>
- Saxton, M. J. (2001). Anomalous subdiffusion in fluorescence photobleaching recovery: A monte carlo study. *Biophysical Journal*, 81: 2226–2240. [https://doi.org/10.1016/S0006-3495\(01\)75870-5](https://doi.org/10.1016/S0006-3495(01)75870-5)
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. <https://doi.org/10.1017/CBO9781107298019>
- Shiguemori, É. H., Da Silva, J. D. S., & de Campos Velho, H. F. (2004). Estimation of initial condition in heat conduction by neural network. *Inverse Problems in Science and Engineering*, 12(3): 317–328. <https://doi.org/10.1080/10682760310001598599>
- Stripinis, L., Paulavičius, R., & Žilinskas, J. (2019). Penalty functions and two-step selection procedure based DIRECT-type algorithm for constrained global optimization. *Structural and Multidisciplinary Optimization*, 59: 2155–2175. <https://doi.org/10.1007/s00158-018-2181-2>
- Tijani, Y. O., & Appadu, A. R. (2022). Unconditionally positive NSFD and classical finite difference schemes for biofilm formation on medical implant using Allen-Cahn equation. *Demonstratio Mathematica*, 55: 40–60. <https://doi.org/10.1515/dema-2022-0006>
- Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. In *Doklady Akademii Nauk*, 151 (pp. 501–504). Russian Academy of Sciences.
- Tikhonov, A. N., Goncharsky, A. V., Stepanov, V. V., & Yagola, A. G. (1995). *Numerical Methods for the Solution of Ill-Posed Problems*. Springer Netherlands, Dordrecht. <https://doi.org/10.1007/978-94-015-8480-7>
- Trask, A. W. (2019). *Grokking deep learning*. Manning Publications.
- Watson, L. T., & Baker, C. A. (2001). A fully-distributed parallel global search algorithm. *Engineering Computations*, 18: 155–169. <https://doi.org/10.1108/02644400110365851>

- Yang, X.-J., Baleanu, D., & Zhong, W.-P. (2013). Approximate solutions for diffusion equations on cantor space-time. *Proceedings of the Romanian Academy - Series A*, 14(2): 127–133.
- Yang, X.-J., Gao, F., & Srivastava, H. (2018). A new computational approach for solving nonlinear local fractional pdes. *Journal of Computational and Applied Mathematics*, 339: 285–296. <https://doi.org/10.1016/j.cam.2017.10.007>
- Yang, X.-J., Machado, J. T., & Srivastava, H. (2016). A new numerical technique for solving the local fractional diffusion equation: Two-dimensional extended differential transform approach. *Applied Mathematics and Computation*, 274: 143–151. <https://doi.org/10.1016/j.amc.2015.10.072>
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021a). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
- Zhang, H., Jiang, X., Zeng, F., & Karniadakis, G. E. (2020). A stabilized semi-implicit Fourier spectral method for nonlinear space-fractional reaction-diffusion equations. *Journal of Computational Physics*, 405: 109141. <https://doi.org/10.1016/j.jcp.2019.109141>
- Zhang, Y., Sun, J., Liu, L., & Qiao, H. (2021b). A review of biosensor technology and algorithms for glucose monitoring. *Journal of Diabetes and its Complications*, 35: 107929. <https://doi.org/10.1016/j.jdiacomp.2021.107929>
- Žilinskas, A., & Baronas, D. (2011). Optimization-based evaluation of concentrations in modeling the biosensor-aided measurement. *Informatika*, 22(4): 589–600. <https://doi.org/10.15388/Informatika.2011.344>

Autoriaus mokslinių publikacijų disertacijos tema sąrašas

Straipsniai recenzuojamuose mokslo žurnaluose

Dapšys, I., & Čiegis, R. (2023). Numerical Simulation of Fractional Power Diffusion Biosensors. *Mathematical Modelling and Analysis*, 28: 180–193. <https://doi.org/10.3846/mma.2023.17583>

Čiegis, R., Dapšys, I., & Čiegis, R. (2022). A Comparison of Parallel Algorithms for Numerical Solution of Parabolic Problems with Fractional Power Elliptic Operators. *Axioms*, 11: 98. <https://doi.org/10.3390/axioms11030098>

Čiegis, R., & Dapšys, I. (2022). On a Framework for the Stability and Convergence Analysis of Discrete Schemes for Nonstationary Nonlocal Problems of Parabolic Type. *Mathematics*, 10: 2155. <https://doi.org/10.3390/math10132155>

Čiegis, R., Čiegis, R., & Dapšys, I. (2021). A comparison of discrete schemes for numerical solution of parabolic problems with fractional power elliptic operators. *Mathematics*, 9. <https://doi.org/10.3390/math9121344>

Dapšys, I., Starikovičius, V. & Čiegis, R. (2023) Applying artificial neural networks to solve the inverse problem of evaluating concentrations in multi-analyte mixtures from biosensor signals, *Nonlinear Analysis: Modelling and Control* (priimtas spausdinti)

Summary in English

Introduction

Problem Formulation

Biosensors are devices that detect and analyse chemical compounds based on biochemical processes (Baronas et al., 2021). A typical biosensor comprises two parts: a bioreceptor, where biochemical reactions occur and the electrode, which converts chemical signals into electric signals. This dissertation is concerned with enzyme biosensors, where the electrode measures the flux of reaction products, and there is an outer diffusion layer where only the diffusion process occurs.

Analysis of samples in practice requires solving the inverse biosensor problem – to determine component concentrations of the sample from its biosensor signal. This problem is ill-posed for multiple substrates (Žilinskas & Baronas, 2011) since some biosensor signals are difficult to discern at a given precision, even though their component concentrations can differ vastly. The other problem is that the biosensor is sensitive to noise (e.g., electric noise) in real devices (Baronas et al., 2014). These reasons decrease the biosensor precision; thus, one of the objectives is to find methods to improve it.

In this dissertation, a virtual biosensor model is used to avoid expenses associated with developing physical prototypes and to obtain biosensor signals faster. This may raise questions about the model's accuracy compared to real-life devices. Therefore, an alternative model was investigated, where the classical diffusion operator is replaced with a fractional power elliptic operator (FPEO). Solving such equations requires specialised numerical methods. To this end, methods based on rational approximations and their parallel

versions were developed and analysed. These methods were applied to a modified biosensor model (Dapšys & Čiegis, 2023).

Relevance of the Dissertation

Biosensors are used for various applications, such as environmental protection and quality control for food production and medicine. In environmental protection, biosensors can detect pollutants – phenolic compounds (Dock et al., 2005), pesticides (Bachmann et al., 2000; Ferentinos et al., 2013), and heavy metals (Huang et al., 2020). In this field, enzyme biosensors are the most common since they are the easiest method for this purpose. In food production, biosensors can be used for safety and quality evaluation (Medina-Plaza et al., 2015), detection of toxic substances (Mishra et al., 2015; Nesakumar et al., 2015) and adulterated foods (Bougadi & Kalogianni, 2020). In medicine, biosensors can analyse proteins in live organisms (Miranda et al., 2010), detect pathogens (Barroso et al., 2018) and measure neuromediators (Mazafi et al., 2018). These applications profoundly influence the quality of life, and improving precision can bring clear benefits.

Object of Research

The research objects are virtual biosensor models, parallel solvers for fractional power elliptic operator problems, and solution methods for inverse problems, including neural networks.

Aim of the Dissertation

The main goal of this dissertation is to develop a biosensor model where the diffusion process is modelled using fractional power elliptic operators (FPEO), develop parallel solvers for such problems and develop neural network-based methods to improve biosensor precision.

Tasks of the Dissertation

1. To develop a biosensor model with FPEO-based diffusion.
2. To develop solvers for non-stationary differential equations with FPEOs and their parallel implementations and to analyse their convergence and scalability.
3. To obtain biosensor responses to solve the inverse problem, using a standard biosensor model.
4. To solve the inverse biosensor problem for signals under the effect of various types and levels of noise.
5. To solve the above inverse problem using the parallel DIRECT method for global optimisation and to analyse its scalability.
6. To investigate the dependence of biosensor signals obtained with the FPEO-based biosensor model on the fractional power parameter.

Research Methodology

Biosensors were modelled using non-stationary models based on systems of nonlinear reaction-diffusion differential equations and solved by the finite volume method. FPEO-based equations were solved by methods given in Čiegis & Dapšys (2022); Čiegis et al.

(2022); Čiegis et al. (2021). Solver software was written in C++ and parallelized using the MPI library. Neural networks, along with principal component analysis, were employed to solve the inverse problem; they were implemented in Python, using the TensorFlow (Google Brain Team, 2021) and Keras (Keras Team, 2021) packages. An alternative approach was based on the parallel DIRECT method for global optimization (Stripinis et al., 2019; Watson & Baker, 2001).

The Scientific Novelty of the Dissertation

1. Rational approximation methods were applied for non-stationary and non-linear parabolic problems with FPEO when the Fourier method could not be applied.
2. Parallel versions of FPEO-based equation solvers were developed and proven to solve large-scale problems efficiently.
3. Convolutional neural networks were employed to solve the inverse biosensor problem, and their effectiveness was shown.
4. The precision of neural networks was improved by applying the procedure of permitted substrate concentration domain shrinkage.
5. The parallel DIRECT method for global optimisation was applied to the inverse biosensor problem.

The Practical Value of the Research Findings

The research results can be used to solve small- and large-scale applied parabolic, non-linear, FPEO-based problems efficiently in science and engineering. The new FPEO-based biosensor model can be used to develop precise physical devices. The test results of methods for solving the inverse biosensor problem can be used to increase the precision of biosensors for noisy signals.

Defended Statements

1. Solvers for non-linear equations with FPEOs, based on rational approximation, extension and splitting methods, were developed; the method analysis showed that their precision meets the practical requirements for such problems.
2. The analysis of parallel versions of the solvers mentioned above has shown that they efficiently use processors for large-scale problems.
3. Different biosensor concentration prediction methods were proposed depending on the noise type: for additive noise, feedforward networks with PCA are more suitable for batch mode, whereas, for flow injection mode, convolutional networks are recommended. For multiplicative noise, flow injection is more precise than batch mode.
4. The permitted substrate concentration domain shrinkage procedure can reduce the neural network prediction errors.
5. The parallel DIRECT version has good scalability, and the method proves that the flow injection mode is more precise than the batch mode.

Approval of the Research Findings

The dissertation results were published in four scientific publications in referred journals of the *Clarivate Analytics Web of Science* database. Three publications were issued in the Q1 quartile (Čiegis & Dapšys, 2022; Dapšys & Čiegis, 2023; Čiegis et al., 2021) and one in the Q2 quartile (Čiegis et al., 2022). One additional article (Dapšys et al., 2023) has been accepted for publication.

Results were also presented at three scientific conferences in Lithuania and abroad:

1. MMA 2022: *Mathematical Modelling and Analysis 2022*. 30 May – 2 June 2022, Druskininkai, Lithuania;
2. NSFDE&A'22: *Numerical Solution Of Fractional Differential Equations And Applications 2022*. 6–9 June, 2022, Sozopol, Bulgaria;
3. MMA 2023: *Mathematical Modelling and Analysis 2023*. 30 May – 2 June 2023, Jūrmala, Latvia.

Structure of the Dissertation

The dissertation consists of an introduction, three main chapters and general conclusions. The first chapter describes the biosensor models used – the classical model and the model with FPEO diffusion, defines fractional power elliptic operators and discusses the inverse biosensor problem. The second chapter describes and analyses solvers for the models in the first chapter, their parallel versions and experimental precision and stability results. The third chapter discusses the application of artificial neural networks and the parallel DIRECT global optimisation algorithm for solving the inverse biosensor problem and the experimental results for the effect of noise and the permitted substrate concentration domain shrinkage procedure.

The total scope of the dissertation is 85 pages, with 219 equations, eight figures, and 40 tables in the text. The dissertation text uses 83 references.

1. Review of mathematical biosensor models and their solution methods

First, the classical biosensor model (Baronas et al., 2021) is considered, i.e., a system of reaction-diffusion equations. Enzymatic reactions occur in the enzyme layer, while in the outer diffusion layer, only the diffusion process takes place:

$$\frac{\partial S_{i,e}}{\partial t} = D_{S_{i,e}} \frac{\partial^2 S_{i,e}}{\partial x^2} - \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad (S1.1)$$

$$\frac{\partial P_{i,e}}{\partial t} = D_{P_{i,e}} \frac{\partial^2 P_{i,e}}{\partial x^2} + \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad i = \overline{1, m}, \quad x \in (0, d), \quad t > 0, \quad (S1.2)$$

$$\frac{\partial S_{i,b}}{\partial t} = D_{S_{i,b}} \frac{\partial^2 S_{i,b}}{\partial x^2}, \quad (S1.3)$$

$$\frac{\partial P_{i,b}}{\partial t} = D_{P_{i,b}} \frac{\partial^2 P_{i,b}}{\partial x^2}, \quad i = \overline{1, m}, \quad x \in (d, d + a), \quad t > 0, \quad (S1.4)$$

where $S_{i,e}$ and $P_{i,e}$ are the substrate and the product concentrations in the enzyme layer, respectively, $D_{S_{i,e}}$ and $D_{P_{i,e}}$ are the respective diffusion constants, $S_{i,b}$, $P_{i,b}$, $D_{S_{i,b}}$ and $D_{P_{i,b}}$

are the concentration values and the diffusion constants in the outer diffusion layer, V_i are the maximal reaction rates, K_i are the Michaelis-Menten constants, the enzyme layer thickness is d , while the diffusion layer has a thickness a . Such a model is used by Baronas et al. (2014) and Litvinas and Baronas (2015). The initial and boundary conditions are:

$$S_{i,e}(x, 0) = 0, P_{i,e}(x, 0) = 0, x \in [0, d], \quad (S1.5)$$

$$S_{i,b}(x, 0) = 0, P_{i,b}(x, 0) = 0, x \in [d, d + a], i = \overline{1, m}. \quad (S1.6)$$

$$S_{i,b}(d + a, 0) = S_{i,0}, P_{i,b}(d + a, 0) = 0, i = \overline{1, m}. \quad (S1.7)$$

$$P_{i,e}(0, t) = 0, D_{S_{i,e}} \frac{\partial S_{i,e}}{\partial x} \Big|_{x=0} = 0. \quad (S1.8)$$

where $S_{i,0}$ are the initial substrate concentrations. How to include these in the model depends on the chosen analysis mode. Two modes are being tested: the batch mode, when the initial substrate concentrations are constant for the duration of the experiment:

$$S_{i,b}(d + a, t) = S_{i,0}, \quad (S1.9)$$

and the flow injection analysis (FIA) mode when the initial concentrations are constant for a limited amount of time t_f , which is called the injection time:

$$S_{i,b}(d + a, t) = \begin{cases} S_{i,0}, & t \leq t_f \\ 0, & t > t_f \end{cases}. \quad (S1.10)$$

There are no products outside the biosensor:

$$P_{i,b}(d + a, t) = 0. \quad (S1.11)$$

On the boundary between the diffusion and enzyme layers, the product and substrate concentrations and fluxes are equal:

$$D_{S_{i,e}} \frac{\partial S_{i,e}}{\partial x} \Big|_{x=d} = D_{S_{i,b}} \frac{\partial S_{i,b}}{\partial x} \Big|_{x=d}, \quad S_{i,e}(d, t) = S_{i,b}(d, t), \quad (S1.12)$$

$$D_{P_{i,e}} \frac{\partial P_{i,e}}{\partial x} \Big|_{x=d} = D_{P_{i,b}} \frac{\partial P_{i,b}}{\partial x} \Big|_{x=d}, \quad P_{i,e}(d, t) = P_{i,b}(d, t), i = \overline{1, m}. \quad (S1.13)$$

The electrode measures the combined flux of all reaction products and depends on the sample composition – this is the biosensor response, and it is given by:

$$I(t) = \sum_{i=1}^m n_i F^{(el)} D_{P_{i,e}} \frac{\partial P_{i,e}}{\partial x} \Big|_{x=0}, \quad (S1.14)$$

where n_i is the number of electrons involved in charge transfer for each product, $F^{(el)} = 96485.33$ C/mol is the Faraday constant.

An alternative biosensor model with fractional power diffusion is also considered. To this end, a definition of fractional power elliptic operators is needed. Notably, such operators can be used to model other non-local processes (Pozrikidis, 2018) and are widespread

in other areas of science as well (Buono-Orovio et al., 2014; Lee, 2018; Metzler & Klafter, 2004; Sabatelli et al., 2002).

FPEOs can be defined in many different ways. This dissertation employed the spectral method (Bonito & Pasciak, 2015; Hofreither, 2020). Let A_h be a discrete, symmetric and positive definite operator, defined on a Hilbert function space H_h :

$$A_h: H_h \rightarrow H_h, \quad A_h = A_h^*, \quad A_h \geq cI_h, \quad c > 0,$$

where I_h is a discrete identity operator. The fractional power elliptic operator A_h^α with fractional power $0 < \alpha < 1$ is defined as:

$$A_h^\alpha U = \sum_{j=1}^J \mu_j^\alpha (U, \psi_j^h) \psi_j^h.$$

Here, (u, v) is the inner product for $u, v \in H_h$, μ_j and ψ_j^h are the eigenvalues and eigenfunctions of A_h , obtained from the equation:

$$A_h \psi_j^h = \mu_j \psi_j^h, \quad j = 1, \dots, J. \quad (\text{S1.15})$$

All eigenvalues are positive:

$$0 < \mu_1 \leq \mu_2 \leq \dots \leq \mu_J,$$

this requirement ensures that the action of the FPEO produces real functions, as required for biosensor applications. The eigenfunctions form an orthonormal basis, spanning the space H_h , and any function $U \in H_h$ can be written as:

$$U = \sum_{j=1}^J (U, \psi_j^h) \psi_j^h.$$

This dissertation studies the semi-discrete Cauchy problem:

$$\frac{\partial U}{\partial t} + A_h^\alpha U = F, \quad 0 < t \leq T, \quad (\text{S1.16})$$

$$U(0) = U_0, \quad U_0 \in H_h. \quad (\text{S1.17})$$

It can be solved by the Fourier method $U(t) = \sum_{j=1}^J U_j(t) \psi_j^h$, where:

$$U_j(t) = U_{0j} e^{-\mu_j^\alpha t} + e^{-\mu_j^\alpha t} \int_0^t e^{\mu_j^\alpha s} F_j(s) ds,$$

$$F(t) = \sum_{j=1}^J F_j(t) \psi_j^h, \quad U_0 = \sum_{j=1}^J U_{0j} \psi_j^h.$$

Although this method can be applied in practice, it has several drawbacks: it only works if the equation domain is rectangular and the Fourier transform can be applied; also, all the eigenvalues and eigenfunctions of the operator must be known.

To obtain the biosensor model with fractional diffusion, only the space derivative in (S1)–(S1.4) equations needs to be replaced by an FPEO:

$$\frac{\partial S_{i,e}}{\partial t} = -D_{S_{i,e}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha S_{i,e} - \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad (\text{S1.18})$$

$$\frac{\partial P_{i,e}}{\partial t} = -D_{P_{i,e}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha P_{i,e} + \frac{(V_i/K_i)S_{i,e}}{1 + \sum_{j=1}^m S_{j,e}/K_j}, \quad x \in (0, d), \quad (\text{S1.19})$$

$$\frac{\partial S_{i,b}}{\partial t} = -D_{S_{i,b}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha S_{i,b}, \quad (\text{S1.20})$$

$$\frac{\partial P_{i,b}}{\partial t} = -D_{P_{i,b}} \left(-\frac{\partial^2}{\partial x^2} \right)^\alpha P_{i,b}, \quad i = \overline{1, m}, \quad x \in (d, d + a), \quad t > 0. \quad (\text{S1.21})$$

The initial boundary conditions and parameters of the model are the same as before, except for the inclusion of the fractional power parameter α . It is important since the obtained biosensor response, and hence, the biosensor precision depends on it.

In practice, when chemical analysis is performed, the inverse biosensor problem needs to be solved by determining the boundary substrate concentrations $C_0 = (\tilde{S}_{1,0}, \dots, \tilde{S}_{m,0})$ from their measured biosensor response $I_m(t)$. Note that for biosensors with FPEOs, it is assumed that α is known. The standard approach is the least squares method (Žilinskas & Baronas, 2011), when the following function is minimised:

$$C_0 = \arg \min_c \sum_{i=1}^n (I(t_i, C) - I_m(t_i))^2, \quad (\text{S1.22})$$

where $I(t_i, C)$ is the simulated biosensor response, whose boundary concentrations are C . This is done by optimisation methods, such as brute-force search, gradient descent, or global optimisation (Jones et al., 1993; Stripinis et al., 2019).

This is a difficult problem due to its ill-posedness (Žilinskas & Baronas, 2011). If two signals, I_1, I_2 , are close to each other, i.e., satisfy $\|I_1 - I_2\|_\infty < \varepsilon$, the distance between their concentrations $\|C_0^{(1)} - C_0^{(2)}\|_2$ may be large. This can occur if the biosensor signals are under the effect of noise, which is present in real devices (Baronas et al., 2014). This noise is modelled by the formulae:

$$I_{mn}(t) = I(t)(1 + \sigma_n X), \text{ or } I_{an}(t) = I(t) + \sigma_n X,$$

where the first equation is for multiplicative noise, the second is for additive noise, and X is a normally distributed random variable: $X \sim N(0,1)$; σ_n is the standard deviation, also known as the noise level.

2. Numerical methods for solution of biosensor models and their analysis

The fully discrete Cauchy problem is obtained by taking a discrete time mesh:

$$\bar{\Omega}_t = \{t^n: t^n = t^{n-1} + \tau_{n-1}, \quad n = 1, \dots, N, \quad t^0 = 0, \quad t^N = T\},$$

and approximating the time derivative:

$$\frac{U^{n+1} - U^n}{\tau_n} + A_h^\alpha U^{n+\sigma} = F(t^{n+\sigma}), \quad n = 0, \dots, N-1, \quad (\text{S2.1})$$

$$U^0 = U_0, \quad U_0 \in V_h,$$

where $U^{n+\sigma}$ is the averaging operator:

$$U^{n+\sigma} = \sigma U^{n+1} + (1 - \sigma)U^n.$$

If $\sigma = 1$, then the implicit Euler scheme is obtained; if $\sigma = 0.5$, then the Crank-Nicolson method is obtained.

A more universal method – BURA (*Best Uniform Rational Approximation*) – is obtained when the non-local operators A_h^α are approximated by local ones using the method of rational approximation. Rewrite the discrete scheme as:

$$(I + \sigma\tau_n A_h^\alpha)U^{n+\sigma} = U^n + \sigma\tau_n F(t^{n+\sigma}),$$

$$U^{n+\sigma} = (I + \sigma\tau_n A_h^\alpha)^{-1}(U^n + \sigma\tau_n F(t^{n+\sigma})).$$

Then, approximate the non-local operator $(I + \sigma\tau_n A_h^\alpha)^{-1}$ by the local operator $r_{m_a}(A_h)$. To do that, the following function is approximated in the maximum norm:

$$f(z) = \frac{1}{1 + \sigma\tau z^\alpha} \approx r_{m_a}(z),$$

on the interval $z \in [\mu_1, \mu_j]$. One method used in this dissertation is the AAA algorithm (Nakatsukasa et al., 2018), where the approximation is performed by taking a set $Z = \{z_1, \dots, z_M\}$ of real numbers and writing $r_{m_a}(z)$ in the barycentric form. The coefficients are obtained by the greedy search method. Another method to achieve this is the BRASIL method (Hofreither, 2021). Once the approximation is done, the operator A_h^α can be written as:

$$A_h^\alpha \approx \bar{c}_0 I + \sum_{j=1}^{m_a} \bar{c}_j (A_h - \bar{d}_j I)^{-1}. \quad (\text{S2.2})$$

The error estimate comes from Harizanov et al. (2018):

$$\left\| A_h^\alpha - \bar{c}_0 I + \sum_{j=1}^{m_a} \bar{c}_j (A_h - \bar{d}_j I)^{-1} \right\| \leq O\left(e^{2\pi\sqrt{m_a}}\right). \quad (\text{S2.3})$$

Since conducted experiments show that $\bar{d}_j \leq 0$, $j = \overline{1, m_a}$, this form of the operator has no poles for $z > 0$.

Precision tests have been performed for the AAA method on a one-dimensional test problem, with a standard diffusion operator A_h ,

$$A_h U = -\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2}, \quad x_j = jh, \quad j = 1, \dots, J-1, \quad x_j = 1, \quad (\text{S2.4})$$

and a source function $F(t)$,

$$F_j(t) = \begin{cases} -1 & 0 \leq x_j \leq 0.5 \\ 1 & 0.5 < x_j \leq 1 \end{cases}. \quad (\text{S2.5})$$

The chosen parameter values were: mesh size $J = 256$, simulation time $T = 0.6$ when the fractional power is $\alpha = 0.5$ and $T = 0.8$ when $\alpha = 0.25$. Although the test problem is one-dimensional, the precision results are representative of the performance for two- or three-dimensional models since the precision depends on the spectrum of A_h and not on the space dimension. The solutions to the problem \tilde{U}_j^n are compared with the solution $U(x_j, t^n)$ obtained by the spectral method. The error was measured in the maximum norm:

$$e_c(N, J) = \max_{(x_j, t^n) \in \bar{\omega}_h \times \omega_t} |U(x_j, t^n) - \tilde{U}_j^n|.$$

The results of the AAA method are presented in Table 2.1. The function $(1 + 0.5\tau z^\alpha)^{-1}$ was sampled at $M = 25000$ discrete points for $z \in [\lambda_{hmin}, \lambda_{hmax}]$. Errors in the table show that this method needs a small number of terms in the approximation to achieve an acceptable error: $m_a = 6$ when $\tau = 0.01, 0.005$ and $m_a = 10$ when $\tau = 0.0025$. Also, m_a does not depend on α .

The BRASIL algorithm was also tested for the problem given in Eq. (2.23) (for the English version, see Čiegis et al. (2022)) when $\sigma = 1$. The chosen parameters were: the space mesh $\bar{\Omega}_h$ uses $J_x = 200$ ir and 400; the fractional power $\alpha = 0.75$; the final time $T = 1$. The error was measured in the maximum norm:

$$E_{m_a}^{(bura)}(\tau) = \max_{(x_i, y_j, z_k) \in \Omega_h} |V_{ijk}^N - u(x_i, y_j, z_k, T)|.$$

Results are given in Table 2.5. It shows that, as in the AAA case, a small m_a is needed to achieve acceptable errors since the convergence rate for rational approximation is exponential. In some cases, the error increases, e.g., when $J_x = 400$, $\tau = 0.0125$, and m_a goes from 5 to 7. This can be explained by the fact that the global error is a sum of three parts: the time, space and rational approximation errors. Some of these can be negative and can compensate for others. If the global error increases, this means that the negative error components have decreased to a level where the compensation is negligible. Compared to AAA, BRASIL has similar precision, but the latter is recommended as it is newer – it should be faster and more robust.

An alternative method is the additive splitting scheme, where the operator A_h^α is decomposed into a classical elliptic and an FPEO of a negative power:

$$\frac{U^{n+1} - U^n}{\tau_n} + A_h^{-\beta} A_h U^{n+\sigma} = F(t^{n+\sigma}), \quad (\text{S2.6})$$

where $\beta = 1 - \alpha$. Such operators are bounded and have a smaller approximation error. The AAA method is employed:

$$R_{m_a}(\beta) = c_0 I + \sum_{j=1}^{m_a} c_j (A_h - d_j I)^{-1}.$$

Then the equation is written as:

$$\frac{U^{n+1} - U^n}{\tau_n} + \sum_{j=1}^{m_a} B_{h,j}^{-1} A_h U^{n+\sigma} = F(t^{n+\sigma}).$$

Since all the operators in this equation commute, it can be solved by the classical splitting algorithm:

$$\begin{aligned} \frac{\tilde{U}^{n,j} - \tilde{U}^{n,j-1}}{\tau_n} + B_{h,j}^{-1} A_h (\sigma \tilde{U}^{n,j} + (1 - \sigma) \tilde{U}^{n,j-1}) &= \frac{1}{m_a + 1} F(t^{n+\sigma}), \\ j = 0, 1, \dots, m_a \end{aligned}$$

More detailed implementations for all of these methods are presented by Čiegis et al. (2021).

The splitting scheme has several modifications, one of which is when the equation is split along other physical processes, such as in the following:

$$\frac{d\tilde{U}}{dt} = F\left(t^{n+\frac{1}{2}}\right), \quad \tilde{U}(t^n) = U^n, \quad t^n < t \leq t^{n+\frac{1}{2}}, \quad (\text{S2.7})$$

$$U^{n+\frac{1}{3}} = \tilde{U}\left(t^{n+\frac{1}{2}}\right),$$

$$\frac{U^{n+\frac{2}{3}} - U^{n+\frac{1}{3}}}{\sigma \tau_n} + A_h^\alpha \frac{U^{n+\frac{2}{3}} + U^{n+\frac{1}{3}}}{2} = 0, \quad (\text{S2.8})$$

$$\begin{aligned} \frac{d\tilde{U}}{dt} &= F\left(t^{n+\frac{1}{2}}\right), \quad \tilde{U}\left(t^{n+\frac{1}{2}}\right) = U^{n+\frac{2}{3}}, \quad t^{n+\frac{1}{2}} < t \leq t^{n+1}, \\ U^{n+1} &= \tilde{U}(t^{n+1}). \end{aligned} \quad (\text{S2.9})$$

This scheme has two advantages: (1) only one non-local problem needs to be solved, and (2) the non-linear problems can be effectively solved by standard ordinary differential equation solvers made for the specific types of non-linear sources.

If a non-linear equation needs to be solved:

$$\begin{aligned} \frac{dU}{dt} + A_h^\alpha U &= F(x, t, U), \quad 0 < t \leq T, \quad x \in \Omega, \\ U(0) &= U_0, \quad U_0 \in V_h, \end{aligned} \quad (\text{S2.10})$$

the aforementioned methods can be applied here as well if linearisation is performed. One way to do this is by the predictor-corrector method:

$$\frac{U^{n+\sigma,k} - U^n}{\sigma \tau_n} + A_h^\alpha U^{n+\sigma,k} = F(x, t^{n+\sigma}, U^{n+\sigma,k-1}), \quad k = 1, \dots, K, \quad (\text{S2.11})$$

where $U^{n+\sigma,k} = \sigma U^{n+1,k} + (1 - \sigma)U^n$. When this method is employed, the amount of computations is increased by K times. If $K = 2$ and $\sigma = 0.5$, the method has second-order precision.

All of the FPEO equation solvers mentioned above have parallel versions. The BURA method is the most suited for this since it expresses the solution as a sum whose terms can be computed independently by solving classical elliptic problems. For the splitting method, each of its subproblems depends on previous iterations, so parallelisation is more difficult. It can only be done for solving the obtained linear systems – the AGMG multigrid solver is employed. Details for parallel algorithms are given in Čiegis et al. (2022).

The scalability of the parallel symmetric splitting scheme was tested on the three-dimensional Gray-Scott system (for a 2D example, see Zhang et al. (2020)):

$$\begin{aligned} \frac{\partial u}{\partial t} + K_u(-\Delta)^\alpha u &= (1 - u)v^2 - Fu, \\ \frac{\partial v}{\partial t} + K_v(-\Delta)^\alpha v &= (1 - u)v^2 - (F + \lambda)v, \\ (x, y, z) \in \Omega &= (0,1) \times (0,1) \times (0,1), \end{aligned} \quad (\text{S2.12})$$

and the parameters were the diffusion constants $K_u = 2 \cdot 10^{-5}$, $K_v = 1 \cdot 10^{-5}$, feed rate $F = 0.03$, and the decay constant $\lambda = 0.061$ for the reagent v . The Dirichlet-type conditions on the boundary $\partial\Omega$ were homogeneous, therefore the solution is defined as $u = 1 - \tilde{u}$. The initial conditions were:

$$\begin{aligned} u(x, y, z, 0) &= 0.5, \quad v(x, y, z, 0) = 0.25, \quad (x, y, z) \in \Omega_1, \\ u(x, y, z, 0) &= 0, \quad v(x, y, z, 0) = 0, \quad (x, y, z) \in \Omega \setminus \Omega_1, \end{aligned} \quad (\text{S2.13})$$

here, $\Omega_1 = \{(x, y, z): (x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 \leq 0.04^2\}$. The execution time $T_p^{(par)}$ (in seconds) with p cores, the speedup $S_p^{(par)} = T_1^{(par)} / T_p^{(par)}$ compared to the running time on one core and the efficiency $E_p^{(par)} = S_p^{(par)} / p$ (the load on all cores) were measured. The dependence of the solution contour on the fractional power α was investigated. The mesh size was fixed at $J_x = J_y = J_z = 256$ for all tests due to the fact that the system is sensitive to the mesh size. The FPEO was solved by using the Fourier method.

The contours for solution v are shown in Fig. 2.1. As the α value is changed, the contours change significantly, which means that the contours are sensitive to changes in the fractional power. The scalability results for the symmetric splitting scheme are presented in Table 2.17. The data shows that the Fourier method has a higher efficiency (Table 2.13). However, the splitting scheme can be used for cases where the Fourier method is not applicable.

Next, an application of FPEO methods is discussed – biosensor models, where the diffusion operator is defined as an FPEO. Define time $\overline{\omega}_t$ and space $\overline{\omega}_h$ meshes (p. 51). The approximate solutions $S_k, P_k, k = 1, \dots, m$ at a mesh point (x_j, t^n) for the (S1.18)–(S1.21) problem are denoted by $\tilde{S}_{k,j}^n, \tilde{P}_{k,j}^n$. Consider two model variants: the first one is obtained when the diffusion constants are included in the diffusion operator:

$$A_{hS}\tilde{S} = \begin{cases} -D_{Se} \frac{2\tilde{S}_1 - \tilde{S}_0}{h}, & j = 0, \\ -\frac{1}{h} \left(D_{S,j+0.5} \frac{\tilde{S}_{j+1} - \tilde{S}_j}{h} - D_{S,j-0.5} \frac{\tilde{S}_j - \tilde{S}_{j-1}}{h} \right), & j = 1, \dots, J-2, \\ -\frac{1}{h} \left(D_{Sb} \frac{-\tilde{S}_{J-1}}{h} - D_{Sb} \frac{\tilde{S}_{J-1} - \tilde{S}_{J-2}}{h} \right), & j = J-1 \end{cases}$$

and,

$$A_{hP}\tilde{P} = \begin{cases} -\frac{1}{h} \left(D_{Pe} \frac{\tilde{P}_2 - \tilde{P}_1}{h} - D_{Pe} \frac{\tilde{P}_1}{h} \right), & j = 1, \\ -\frac{1}{h} \left(D_{P,j+0.5} \frac{\tilde{P}_{j+1} - \tilde{P}_j}{h} - D_{P,j-0.5} \frac{\tilde{P}_j - \tilde{P}_{j-1}}{h} \right), & j = 2, \dots, J-2, \\ -\frac{1}{h} \left(D_{Pb} \frac{-\tilde{P}_{J-1}}{h} - D_{Pb} \frac{\tilde{P}_{J-1} - \tilde{P}_{J-2}}{h} \right), & j = J-1. \end{cases}$$

Here, $D_{S,j-0.5} = D_{Se}$, $D_{P,j-0.5} = D_{Pe}$, when $j = 1, \dots, J_d$ and $D_{S,j+0.5} = D_{Sb}$, $D_{P,j+0.5} = D_{Pb}$, when $j = J_d, \dots, J-1$. To employ the spectral method to define the FPEOs, these operators are defined for homogeneous boundary conditions ($\tilde{S}_j = 0$). Therefore, non-homogeneous biosensor boundary conditions $\tilde{S}_l(d+a, t) = s_l$ must be included in the source term – they are non-local functions since a non-local operator is applied to them.

Biosensor equations are solved using the symmetric splitting algorithm, which is based on the Crank-Nicolson and the predictor-corrector methods. The former is an unconditionally stable second-order solver, while the latter is used to linearise non-linear equations. The substrate equations (2.64) are solved first. Next, the product equations (2.65) are solved. The equations for the splitting scheme (2.66)–(2.71) are obtained by splitting along the non-linear reaction and linear diffusion processes. The advantage of this approach is that only one non-local equation for every product and substrate needs to be solved.

The second model variant is obtained when the diffusion constants are not included in the diffusion operator. Let the diffusion constants be equal both in the diffusion and the reaction layers. Then, define the operators:

$$\mathcal{A}_{hS}\tilde{S} = \begin{cases} -\frac{2\tilde{S}_1 - \tilde{S}_0}{h}, & j = 0, \\ -\frac{1}{h} \left(\frac{\tilde{S}_{j+1} - \tilde{S}_j}{h} - \frac{\tilde{S}_j - \tilde{S}_{j-1}}{h} \right), & j = 1, \dots, J-2, \\ -\frac{1}{h} \left(\frac{-\tilde{S}_{J-1}}{h} - \frac{\tilde{S}_{J-1} - \tilde{S}_{J-2}}{h} \right), & j = J-1, \end{cases} \quad \mathcal{A}_{hP}\tilde{P} = \begin{cases} -\frac{1}{h} \left(\frac{\tilde{P}_2 - \tilde{P}_1}{h} - \frac{\tilde{P}_1}{h} \right), & j = 1, \\ -\frac{1}{h} \left(\frac{\tilde{P}_{j+1} - \tilde{P}_j}{h} - \frac{\tilde{P}_j - \tilde{P}_{j-1}}{h} \right), & j = 2, \dots, J-2, \\ -\frac{1}{h} \left(\frac{-\tilde{P}_{J-1}}{h} - \frac{\tilde{P}_{J-1} - \tilde{P}_{J-2}}{h} \right), & j = J-1. \end{cases}$$

The fractional powers \mathcal{A}_{hS}^α and \mathcal{A}_{hP}^α of these operators are defined by (2.73), where eigenvalues and eigenfunctions are obtained by solving equations (2.72). These fractional power operators are defined for both variants of the model.

Biosensor models with fractional power diffusion are obtained by changing the diffusion layer equations (2.68)–(2.69) to equations (2.74)–(2.75) for the first model and to equations (2.76)–(2.77) for the second model. These diffusion equations can be efficiently solved by the Fourier method since the system is defined on a 1D interval, and the eigenvalues and eigenfunctions are known. Both model variants have the same eigenfunctions, Φ_{Sj} and Φ_{Pj} :

$$\Phi_{Sk,j} = \cos\left(\pi \frac{(j-0.5)k}{J}\right), \quad \lambda_{Sj} = \frac{4}{h^2} \sin^2\left(\frac{\pi(j-0.5)}{2J}\right), \quad k, j = 1, \dots, J,$$

$$\Phi_{Pk,j} = \sin\left(\pi \frac{kj}{J}\right), \quad \lambda_{Pj} = \frac{4}{h^2} \sin^2\left(\frac{\pi j}{2J}\right), \quad k, j = 1, \dots, J-1.$$

Only the eigenvalues for the second variant are given – eigenvalues for the first variant are similar and can be computed trivially.

The results for the dependence of the biosensor response (S1.14) on the fractional power parameter for both model variants are given below. Three substrates $m = 3$ and these parameter values (Baronas et al., 2021) were used:

$$\begin{aligned} V_{max,l} &= 5 \times 10^{-(9-l)}, \quad K_M = 1 \times 10^{-4}, \quad l = 1, 2, 3, \\ D_{Se} = D_{Sb} &= 4.5 \times 10^{-6}, \quad D_{Pe} = D_{Pb} = 4.5 \times 10^{-6}, \\ s_1 &= 3.88574, \quad s_2 = 11.65712, \quad s_3 = 5.25716, \\ d &= 0.02 \text{ cm}, \quad a = 0.04 \text{ cm}, \quad T = 400 \text{ s}. \end{aligned}$$

The space mesh size is $N = 999$. The biosensor signals $I(t_k)$ are sampled every second.

The results show that the biosensor response has a strong dependence on the fractional power. The signals for the first model variant reach the stationary value faster, and its magnitude gets lower as the fractional power decreases (Fig. 2.2). The signal is sensitive to changes in α : as its value changes by 0.1, the initial rate of growth changes noticeably, the saturation current changes by about 500, and the saturation time – by about 150s.

The responses for the second variant display the opposite behaviour – the saturation current gets larger and is reached later (Fig. S2.3). This time, the sensitivity to α is greater: as its value changes by 0.01, the saturation current changes by about 2000, $t = 500$.

3. Numerical methods for the solution of the inverse biosensor problem and their analysis

The main method for solving the inverse biosensor problem in this dissertation is artificial neural networks (Trask, 2019; Zhang et al., 2021a) – a machine learning method that models the functions of the human brain. Two neural network architectures are used – feedforward and convolutional. Feedforward neural networks are the simplest, but they often require dimensionality reduction of input data to help reduce overfitting and increase training speed; in this case, principal component analysis (PCA) was used. Convolutional

neural networks (CNN) learn a suitable non-linear lower dimensional representation of data by themselves. Since the inputs are one-dimensional, these networks use 1D convolution layers.

The architecture for the feedforward neural network was configured by choosing between one and two hidden layers and between five different amounts of neurons in each: 18, 28, 32, 36 and 46. The most precise configuration found was a single hidden layer of 32 neurons. Initially, four principal components (PCs) were used as input, and the amount was gradually increased to 80. The activation functions were tanh for the hidden layer and linear for the output layer. 1D convolutional neural networks were configured as four pairs of convolution-pooling layers. Five CNN configurations were used in the dissertation. In Table 3.2, numbers of kernels, kernel sizes, and pool sizes are given for the specified convolution-pooling pairs of layers. After the fully connected layer, the dropout procedure was employed – randomly set some of the outputs of the preceding layer to 0. The dropout rate was 0.1. The activation function chosen for the convolution and fully connected layers was ReLU. Both types of neural networks were trained using the Adam optimiser, had a batch size of 64 and a learning rate of $1.25 \cdot 10^{-4}$ for feedforward and $5 \cdot 10^{-4}$ for convolutional networks. The number of training epochs varied for every noise level. The latter network type was trained three times each, and the one with the lowest testing error was picked.

Consider the distribution of errors for biosensor predictions on the concentration cube for the training set (Fig. 3.2). Every point in this figure is coloured according to the largest error for all three concentrations. If cube layers are removed from all sides, it can be seen that the maximal errors are decreasing. The original training cube is on the left; in the middle – three layers were removed, and on the right – six layers were removed. Also, the worst errors are on the edge of the original cube. Therefore, the permitted substrate concentration domain shrinkage procedure was devised, which in this dissertation is called cube shrinkage, when the testing dataset is generated inside a smaller domain whose edges are some distance away from the original domain.

Next, neural network precision tests were performed for the inverse biosensor problem for two architectures: feedforward and convolutional. The responses were obtained for two analysis modes – batch and flow injection analysis and two noise types – additive and multiplicative.

The model used three substrate concentrations, and the value of each was between 3.2 and 12.8, as provided by Baronas et al. (2014) and Litvinas & Baronas (2015). Taking a bounded cube $C = [3.2, 12.8]^3$ in concentration space, the training set was generated inside as a 3D discrete mesh, a single edge D (3.30) for it is comprised of $N = 22$ discrete points between bounds. The total number of points – the size of the training set was 10 648. The testing set was generated as a cloud of randomly generated points inside C . Each point in these sets is used to generate respective responses. The size of the testing set was 3000. Another testing data set “rc” was generated in a reduced cube (Table 3.3). A parallel algorithm based on the master-slave paradigm was used to speed up computations. For the (S1.1)–(S1.4) model, these parameter values were used: enzyme layer thickness $d = 0.02$ and the maximal enzymatic rates $V_i = 5 \cdot 10^{i-9}$, $i = 1, 2, 3$. Other parameters are given in Table 3.1. Simulation time is $T = 400$ for batch mode and $T = 1200$ for flow injection analysis, the time step $\tau = 0.25$, and responses were sampled every second. Injection time was $t_f = 3$.

The performance of the neural network was measured by the average and maximal relative absolute percentage errors, $E_{ds,a}$ (3.31) and $E_{ds,m}$ (3.32), respectively. The relative prediction errors for the most relevant tests are given in Tables 3.4–3.9. Errors for the testing set inside the original cube are denoted by $E_{ts,a}$ and $E_{ts,m}$; errors inside the reduced cube are $E_{rc,a}$ and $E_{rc,m}$. The number of principal components (PC in tables), the neural network configuration (for convolutional networks) and the number of epochs are also given. The overview of these test results is provided after the tables.

The results are summarized as follows:

- When using the feedforward neural network and additive noise, the errors were better for 80 principal components for noisy signals in the batch mode than for four principal components. A possible reason for this might be that the neural network changes the weighting of higher principal components during training so that noise itself is employed to distinguish between different signals. For multiplicative noise, this result holds until the noise level 0.02 is reached when mean errors get similar, and the maximal errors get larger. In the FIA mode, 4 PCs have smaller errors for both noise types.
- It was determined that the simple feedforward neural network architecture has lower errors for batch mode than for flow injection analysis, while the convolutional network is more precise for the latter mode rather than the batch mode when the signals are affected by additive noise (Tables 3.4, 3.5 and 3.6).
- The results for the convolutional neural network are similar to the feedforward neural network for the batch mode. Although the former network gives smaller errors, these may become similar by a better choice of hyperparameters. Both architectures fail to give adequate errors in the worst case; therefore, the batch mode is considered impractical. This suggests that such poor performance is due to the ill-posedness of the inverse problem. For the FIA mode, the convolutional architecture is slightly more precise (Tables 3.4, 3.5 and 3.6).
- When compared to additive noise, multiplicative noise has a larger effect on errors for the batch mode, but for the FIA mode, the errors for this noise are significantly smaller (Tables 3.7 and 3.9). The reason might be that signal noise can improve accuracy because similar signals may become more distinguishable.
- The permitted substrate concentration domain shrinkage procedure can reduce maximal errors to acceptable levels but fails for large noises. However, average errors start decreasing in that case. An example of this can be seen in Table S3.4: maximal errors are reduced to manageable levels until the noise level of 0.03 is reached; average errors remain similar until the noise level of 0.2 when the errors begin to decrease. Cube shrinkage generally works similarly for other neural network architectures and noise types.
- Based on these results, both types of neural networks can be compared. FNN and CNN have similar precision for the batch mode with both types of noise and the FIA mode with additive noise. FNN is slightly more precise than CNN for small levels of multiplicative noise, but for larger noise levels, it becomes better than the FNN network.

Lastly, the parallel master-worker DIRECT global optimisation algorithm (Watson & Baker, 2001) was tested. A description of a serial version of this algorithm and some of its variants is given by Jones & Martins (2021). The goal of this experiment is to solve

the inverse biosensor problem for a single noiseless response corresponding to a concentration triple (4.5714, 10.0571, 3.2) in batch and FIA modes to see which one gives a higher precision. Acceptable results were obtained only by using a smaller search region: $R = [3.4, 6] \times [7, 12.6] \times [3.2, 6]$. The precision results are in Table S3.1 – here, the largest relative error among three concentrations was used. Scalability results are in Table 3.11.

Table S3.1. Prediction errors for the DIRECT algorithm

	Batch		FIA	
	CNN	DIRECT	CNN	DIRECT
Error	1.83	7.39	0.229	1.53

Below is the summary of the test results:

- The DIRECT algorithm performs better for the FIA mode than the batch mode. Also, the errors for the CNN are smaller than for DIRECT. This can be explained by the objective function having many local extrema, and the DIRECT method spends time checking regions containing them. Thus, it fails to find the global minimum within the same time when the CNN training completes, allowing the latter to achieve a smaller error.
- A solution was found faster for the FIA than for the batch mode: 50 iterations for the former and 100 for the latter; doubling to 200 had no benefit.
- Efficiency decreases as p increases because, in every iteration, the master process distributes a certain amount of tasks for the workers. Since there is a small fixed amount of iterations, the amount of tasks is also small and fixed, and the worker processes cannot be fully loaded as their number increases. Efficiency could be improved by dynamically scaling the number of worker processes according to the number of tasks.

General Conclusions

The general conclusions of the dissertation are as follows.

1. The analysis of FPEO equation solvers proves that the solvers have sufficient precision for practical purposes. These solvers were used to create and solve a biosensor model with FPEO-based diffusion. The obtained signals are sensitive to the fractional power parameter.

2. Parallel versions of these methods scale well for large problems with FPEOs. Scalability tests for the Gray-Scott equation show that the efficiency exceeds 80% using up to 16 processors.

3. Experiments were performed on solving the inverse biosensor problem, which is ill-posed and sensitive to noise. Neural networks successfully solved the problem. The results of experiments lead to recommendations for improving biosensor precision.

4. Applied neural networks can benefit from permitted substrate concentration domain shrinkage for error improvement in all cases tested. It can improve maximal biosensor errors for a low noise level, while average errors are reduced for a high noise level.

It is also recommended to use the parallel DIRECT global optimisation method, which is more precise for the flow injection mode than for the batch mode. Its efficiency is 80% when using up to ten CPUs, but the algorithm properties make maximum efficiency harder to achieve.

Ignas DAPŠYS

KLASIKINĖS IR TRUPMENINIO LAIPSNIO
DIFUZIJOS-REAKCIJOS MODELIAIS PAREMTŲ
BIOLOGINIŲ JUTIKLIŲ ATVIRKŠTINIO UŽDAVINIO
SPRENDIMO ALGORITMAI IR JŲ ANALIZĖ

Daktaro disertacija

Gamtos mokslai,
Informatika (N 009)

ANALYSIS OF ALGORITHMS FOR THE SOLUTION
OF THE INVERSE BIOSENSOR PROBLEM
BASED ON CLASSICAL AND FRACTIONAL
POWER DIFFUSION-REACTION MODELS

Doctoral Dissertation

Natural Sciences,
Informatics (N 009)

Anglų kalbos redaktorė Jūratė Griškėnaitė

Lietuvių kalbos redaktorė Dalia Markevičiūtė

2023 11 03. 10,5 sp. l. Tiražas 20 egz.
Leidinio el. versija <https://doi.org/10.20334/2023-044-M>
Vilniaus Gedimino technikos universitetas
Saulėtekio al. 11, 10223 Vilnius
Spausdino UAB „Ciklonas“
Žirmūnų g. 68, 09124 Vilnius