

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Skaičiavimų tinklų naudojimas biokatalizės procesams
tirti**

Using grid computing to investigate biocatalytic processes

Magistro baigiamasis darbas

Atliko: Vytautas Ašeris
Darbo vadovas: Prof. Romas Baronas
Recenzentas: Evelina Gaidamauskaitė

Vilnius – 2009

Santrauka

Šio darbo tikslas buvo sukurti metodiką, kuri užtikrintų efektyvų skaičiavimų tinklų panaudojimą kompiuteriniams modeliams, nagrinėjantiems biokatalizės procesus. Išnagrinėjus skaičiavimų tinklų panaudojimo galimybes biokatalizės procesams tirti, buvo pateikta skaičiavimų tinklų panaudojimo metodika, kurią galima įgyvendinti nepriklausomai nuo konkretaus skaičiavimų tinklo. Metodika buvo įgyvendinta skaičiavimų tinkle *BalticGrid* ir taikant šią metodiką atliktas biojutikliuose vykstančio biokatalizės proceso tyrimas.

Biojutiklio veikimas aprašomas matematiniais modeliais, kuris bendruoju atveju yra nagrinėjamas naudojant kompiuterinį modelį. Biojutiklių veikimo kompiuteriniai modeliai dažnai yra parametrizuojami ir daugeliu atvejų yra sprendžiamas tas pats uždavinys, tik su skirtingais parametru rinkiniais. Šiam modeliavimo procesui atlikti tikslinga naudoti skaičiavimų tinklus. Darbe pateikiama metodika buvo pritaikyta atliekant chemiškai modifikuoto biojutiklio kvazistacionariojo ir detaliojo modelių palyginimas.

Sukurtos metodikos tyrimo metu nustatyta, kad ši metodika gali būti efektyviai taikoma nagrinėjant biokatalizės procesus. Nustatytas metodikos taikymo efektyvumas konkrečiame skaičiavimų tinkle bei pateiktos rekomendacijos kaip efektyviai taikyti šią metodiką.

Summary

The purpose of this master thesis was to develop a methodology, which would assure an efficient way to use computing grids for computer models modeling biocatalytic processes. After researching possibilities to use computing grids for modeling of such processes, a methodology was presented, which can be used independently of grid middleware. Methodology was implemented on *BalticGrid*. By applying this methodology biocatalytic processes in biosensors were investigated.

Operation of biosensor is described with mathematical model, which in general case is solved with computer model. Computer models of biosensors are commonly parameterized, and in most cases same problem is being solved only with different set of parameters. Such processes can be solved by using computational grids. Methodology presented in this paper was used to compare quasi steady state and detailed models of chemically modified biosensors.

After examining methodology presented in this paper it was proved that this methodology can be used efficiently to research biocatalytic processes. Effectiveness of applying this methodology in concrete computational grid was estimated and suggestions were presented how to apply this methodology in an efficient way.

Turinys

Santrauka	2
Summary.....	3
1. Įvadas.....	5
1.1. Tiriamoji sritis	5
1.2. Temos aktualumas ir naujumas	5
1.3. Darbo tikslas ir uždaviniai	6
2. Skaičiavimo tinklai ir jų naudojimas kompiuteriniame modeliavime.....	8
2.1. Skaičiavimų tinklai	8
2.2. Skaičiavimo tinklų panaudojimas kompiuteriniam modeliavimui	14
2.3. Apibendrinimas	16
3. Skaičiavimo tinklų panaudojimo biokatalizės procesams tirti metodika	18
3.1. Reikalavimai kompiuteriniams modeliams	18
3.2. Skaičiavimo tinklų panaudojimo schema	20
3.3. Metodikos santrauka.....	21
3.4. Metodikos įgyvendinimas	23
3.5. Metodikos tyrimas	24
3.5.1. Minimalios efektyviai sprendžiamos skaičiavimų tinkle užduoties apimtis	25
3.5.2. Efektyvaus skaičiavimų tinklų panaudojimo rekomendacijų santrauka.....	30
4. Metodikos taikymas tiriant biokatalizės procesą.....	32
4.1. Chemiškai modifikuoto biojutiklio modeliavimas	32
4.1.1. Detalusis modelis.....	32
4.1.2. Nedimensinis detalusis modelis.....	34
4.1.3. Kvazistacionarusis modelis	36
4.1.4. Nedimensinis kvazistacionarusis modelis	37
4.2. Matematinų modelių sprendimas kompiuteriu	38
4.2.1. Detaliojo modelio atveju	38
4.2.2. Kvazistacionariojo modelio atveju	40
4.3. Kompiuterinių modelių patikimumas.....	40
4.4. Detaliojo ir kvazistacionariojo biojutiklių modelių palyginimas	41
Rezultatai	47
Išvados	48
Šaltinių sąrašas	49
Priedai	53
1 Priedas: Naudotos Q reikšmės.....	53

1. Įvadas

1.1. Tiriamoji sritis

Darbo tiriamoji sritis yra biokatalizės procesų kompiuterinis modeliavimas naudojantis skaičiavimų tinklais.

Nagrinėjami biokatalizės procesai yra modeliuojami skaitmeniškai, naudojant kompiuterinį modelį, kuris pagrįstas matematinio proceso modeliu. Nagrinėjant biokatalizės procesus yra aktualu iširti jų būsenos priklausomybę nuo įvairių fizinių parametrų, todėl nagrinėjamas toks kompiuterinis modelis, kurio pagalba būtų galima atlikti skaičiavimus skirtingoms parametrų reikšmėms. Dažnai yra tikslinga tirti nagrinėjamo biokatalizės proceso elgseną didelėje fiziškai prasmingų parametrų aibėje, todėl modeliuojant proceso veikimą kompiuteriu iškyla itin aktuali resursų problema – skaičiavimai užtrunka nemažai laiko, todėl kyla poreikis kaip įmanoma paspartinti biokatalizės proceso kompiuterinį modeliavimą. Šiuo metu vienas iš būdų spartinti kompiuterinių modelių veikimą yra naudojant skaičiavimų tinklus [Ber02]. Būtent skaičiavimų tinklų panaudojamumas ir efektyvumas minėtam modeliavimo uždaviniui spręsti ir yra tiriamoji darbo sritis.

1.2. Temos aktualumas ir naujumas

Kompiuteriniai biokatalizės procesų veiksmo modeliai darosi patrauklesni įvairiose mokslinėse tyrimų srityse. Tyrinėjant tokius procesus, itin aktualu pakeisti fizinius eksperimentus kompiuteriniais, nes tokie eksperimentai yra greitesni ir pigesni.

Tyrinėjant procesų modelius, kuriuos pilnai išspręsti galima tik skaitiniais metodais, nes analitiškai sprendžiamos tik atskiros modelio dalys, itin aktualu sukurti gerą algoritmą ir jį realizuojančią programinę įrangą, sprendžiančius šiuos modelius. Tokios priemonės ženkliai paspartina įvairius tyrimus ir eksperimentus, tuo pačiu visą tyrimo procesą padarydamos pigesniu ir patogesniu. Skaičiavimai, atliekami naudojantis kompiuteriniais modeliais, užtrunka ilgai, šiuos skaičiavimus galima paspartinti kiek įmanoma tobulinant nuoseklųjį algoritmą arba panaudojant išskirstytus skaičiavimus kompiuterių tinkluose. Nuosekliojo algoritmo tobulinimas yra nagrinėtas ne tik autoriaus ankstesniame darbe [Aše07], bet ir išsamiai iširtas kituose panašaus pobūdžio darbuose [Gai06]. Be to, nuoseklieji skaičiavimai turi kitą problemą – dėl technologinių kliūčių kompiuterių spartinimas vykdomas ne procesorių greitinimo, o jų skaičiaus didinimo keliu. Taigi neišvengiamai reikia nagrinėti biokatalizės proceso modeliavimo išlygiagretinimo galimybę.

Skaičiavimų tinklai yra vis dar besivystanti technologija. Kompiuterių telkinys (angl. *cluster*) yra grupė plačiai sujungtų, tačiau labai glaudžiai dirbančių kompiuterių, kuri daugeliu atžvilgių gali būti laikoma kaip vienas kompiuteris. Nepriklausomus klasterius (angl. *cluster*) jungiant į grupes, gaunami paskirstytų skaičiavimų tinklai, vadinami gridais (angl. *grid*). Lietuvoje vykdomi *LitGrid* ir *BalticGrid* projektai, kurių tikslas yra pateikti tokią moksliniams skaičiavimams skirtą aplinką, kuri užtikrintų efektyvius mokslinius tyrimus.

Nagrinėjant biokatalizės procesą kompiuterinio modelio pagalba, aktualu ištirti konkrečių parametru įtaką proceso būsenai. Modeliuojant yra tiriama proceso elgsenos priklausomybė nuo įvairių parametru rinkinių – daugeliu atvejų gali būti sprendžiamas tas pats uždavinys, tik su skirtingomis parametru reikšmėmis. Tokius uždavinius galima spręsti lygiagrečiai, todėl skaičiavimo tinklų panaudojimas biokatalizės proceso tyrime yra itin aktualus, nes tai yra perspektyvus būdas spartinti šių procesų kompiuterinį modeliavimą, sprendžiant minėtą uždavinį.

Parametrizuojamiems kompiuterinio modeliavimo uždaviniams spęsti yra sukurti įvairių programinių įrangų [*APST*, *YMB+02*, *AGK00*]. Šiame darbe pateikiama nauja abstrakti metodika, skirta pasinaudoti skaičiavimų tinklais parametrizuojamiems modeliavimo uždaviniams spęsti. Metodika ypatinga tuo, kad skirtingai nuo konkrečių skaičiavimo tinklų panaudojimo atvejų, ji nėra priklausoma nuo skaičiavimų tinkluose naudojamos tarpinės programinės įrangos, ir gali būti įgyvendinta bet kokiais skaičiavimų tinklo architektūrai.

Darbe aprašoma metodika buvo įgyvendinta praktiškai, sukuriant programinę įrangą, skirtą pasinaudoti skaičiavimų tinklo *BalticGrid* resursais. Metodika pritaikyta palyginant chemiškai modifikuoto biojutiklio kvazistacionariųjų ir detalųjų veikimo modelius. Aktuali tema yra nustatyti, kokiais atvejais galima laikytis kvazistacionarumo prielaidos nagrinėjant fermentinę reakciją [*SM03*,*LSL08*]. Todėl tikslinga ištirti, kokiomis sąlygomis galima laikytis kvazistacionarumo sąlygos chemiškai modifikuotuose biojutikliuose.

1.3. Darbo tikslas ir uždaviniai

Šio darbo tikslas yra, ištyrus skaičiavimų tinklų panaudojimo galimybes biokatalizės procesams modeliuoti, sukurti metodiką, kuri užtikrintų efektyvų skaičiavimų tinklų panaudojimą kompiuteriniams modeliams, tiriantiems biokatalizės procesą.

Siekiant šio tikslo, buvo sprendžiami tokie uždaviniai:

1. Atlikus skaičiavimų tinklų analizę, nustatyti, kokius reikalavimus turi atitikti kompiuteriniai modeliai, siekiant kompiuterinio modeliavimo procesui efektyviai panaudoti skaičiavimų tinklų resursus.

2. Atsižvelgiant į nustatytus reikalavimus, įvertinti skaičiavimų tinklų panaudojimo galimybę biokatalizės procesams modeliuoti.

3. Sukurti metodiką, užtikrinančią efektyvų išskirstytų skaičiavimų tinklų panaudojimą biokatalizės proceso parametrų įtakai tirti kompiuteriniu modeliu.

4. Sukurtą metodiką pagrįsti eksperimentiškai – sukurti kompiuterinį konkretaus biojutiklio veikimo modelį, ir, naudojantis sudaryta metodika, kompiuterinį modeliavimą atlikti pasinaudojant skaičiavimų tinklų resursais.

5. Iširti parametrų įtaką konkretaus nagrinėjamo biojutiklio elgsenai.

6. Eksperimento metu atlikti skaičiavimų tinklų panaudojimo efektyvumo tyrimą, lyginant su nuosekliais skaičiavimais, ir pateikti rekomendacijas, kurių reikėtų laikytis siekiant kuo didesnio efektyvumo.

2. Skaičiavimo tinklai ir jų naudojimas kompiuteriniame modeliavime

2.1. Skaičiavimų tinklai

Darbe nagrinėjamų skaičiavimo tinklų (angl. *computing GRID*) sąvoka atsirado sąlyginai neseniai, 1998 metais ji buvo apibrėžta taip [FK99a]: skaičiavimo tinklas yra programinės ir aparatūrinės įrangos infrastruktūra, kuri pateikia patikimą, pastovų, plačiai paplitusį ir nebrangų priėjimą prie aukštos kokybės skaičiavimo galimybių. 2002 metais buvo pateiktas kitas skaičiavimo tinklų apibrėžimas, kuris naudojamas iki šiol [Fos02a]: skaičiavimų tinklas yra sistema, kuri tenkina šiuos tris reikalavimus:

1. Koordinuoja išteklius, kurie nėra centralizuotai valdomi – skaičiavimų tinklas apjungia resursus ir vartotojus, kurie gali būti skirtinguose administraciniuose domenuose. Išteklių gali būti skirtingose kompanijose, miestuose ar net valstybėse.
2. Naudoja standartinius, atvirus, bendros paskirties protokolus ir sąsajas – skaičiavimų tinklas pagrįstas kelių lygių protokolais ir sąsajomis, kurie naudojami autentifikacijai ir autorizacijai, taip pat resursų radimui ir panaudojimui. Todėl būtina, kad šie protokolai ir sąsajos būtų standartiniai ir atviri, nes priešingu atveju tai būtų specifinė sistema (angl. *applicationspecific system*) [Ber02].
3. Pateikia aukštos kokybės paslaugas – skaičiavimų tinklas turi leisti savo resursus panaudoti kokybiškai koordinuotai, taip užtikrinant, kad naudojantis skaičiavimų tinklais užduotims resursai būtų išskirti per tinkamą laiko tarpą, užduočių sprendimas nebūtų sutrukdomas tinklo ar kitų techninių problemų kliūčių, taip pat įvertinant ir reikiamus saugumo reikalavimus.

Apibendrintai skaičiavimų tinklus galima vadinti rinkiniu lokaliame ir globaliame tinkle paplitusių skaičiavimo resursų, kurie vartotojui ar aplikacijai atrodo kaip viena didelė skaičiavimų sistema [Kou06]. Skaičiavimų tinklai yra paskirstytųjų skaičiavimų (angl. *distributed computing*) plėtinys, kuris pasižymi tuo, kad apjungia ne tik skirtingas vietas, bet ir skirtingas organizacijas, architektūras, programinę ir techninę įrangą. Skaičiavimų tinklai leidžia saugiai atlikti daugybę užduočių plačiai paskirstytame tinkle.

Kalbant apie skaičiavimų tinklus, galima išskirti tris pagrindinius skaičiavimo lygius [CM05]:

- Skaičiavimo mazgas (angl. *node*) – asmeninis kompiuteris arba serveris, turintis vieną ar daugiau procesorių.

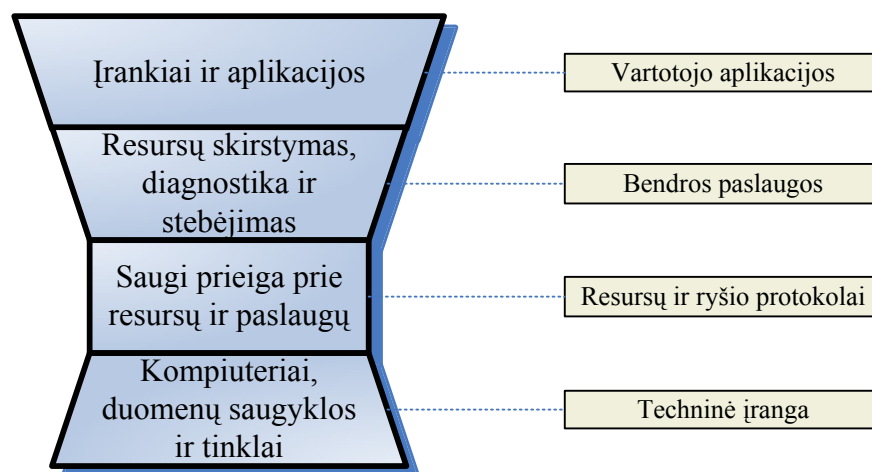
- Telkinys (angl. *cluster*) – rinkinys tarpusavyje susietų skaičiavimo mazgų.
- Skaičiavimų tinklas – klasterių rinkinys.

Skaičiavimo mazgai klasteryje yra sujungti sparčiuoju ryšiu. Kol nebuvo sukurtos *InfiniBand* ir *PCI Express* technologijos, telkiniuose atliekamų užduočių greitis buvo smarkiai apribotas dėl tuometinės tinklo technologijos. Tuo metu buvo naudojama *Ethernet* technologija, kuri buvo silpnoji telkinių vieta skaičiavimo mazgų bendravimo atžvilgiu, nes aplikacijoms, kurioms buvo reikalingas ryšys *skaičiavimo mazgas – skaičiavimo mazgas* veikė lėčiau, nei sukūrus minėtas technologijas.

Kompiuterių telkiniai į skaičiavimų tinklą gali būti jungiami naudojantis vidiniu tinklu (angl. *Local Area Network, LAN*), tačiau tokiu atveju skaičiavimų tinklas būna apribojamas vietovės, kuriose bazuojami kompiuterių telkiniai, ribomis. Kitas sprendimas yra klasterius sujungti į platųjį tinklą (angl. *Wide Area Network, WAN*) – tokiu būdu skaičiavimų tinklo elementai gali būti paplitę po visą pasaulį.

Klasteris gali būti laikytinas skaičiavimų tinklu tik tokiu atveju, jei vartotojui yra pateikiama galimybė naudotis jo skaičiavimo resursais nepriklausomai nuo programinės ar techninės įrangos. Tačiau net jei klasteris būtų sudarytas ir iš šimtų ar tūkstančių skaičiavimo elementų, bet neturėtų infrastruktūros, kokia yra būdinga skaičiavimų tinklui, tokio klasterio skaičiavimų tinklu nebūtų galima vadinti. Tipiniu atveju, norint prisijungti prie klasterio, reikalinga speciali programinė įranga (tokia kaip *rlogin* arba *telnet* [CM05]), kuri priklauso nuo operacinės sistemos tipo, kas nėra suderinama su skaičiavimų tinklo apibrėžimu.

Skaičiavimų tinklo architektūra gali būti pateikiama kaip grupė skirtingo pločio sluoksnių [Fos02b]: fizinis lygis, resursų ir ryšio protokolai, bendrosios paslaugos ir vartotojo aplikacijos (1 pav.).



1 pav. Skaičiavimų tinklo architektūra sluoksniais.

Žemiausiame lygmenyje yra fizinis sluoksniu, kuris yra sudarytas iš techninių įrenginių ir resursų, kuriais vartotojas nori pasinaudoti. Virš fizinio lygmens esančiame resursų ir ryšio lygmenyje esantys protokolai turi būti įgyvendinami visur, todėl natūralu, jog šių protokolų skaičius turi būti nedidelis. Ryšio lygmenyje pateikiami bendravimo (duomenų apsikeitimo tarp resursų) ir autentifikavimosi (kriptografiškai saugūs mechanizmai, skirti vartotojų ir resursų identifikavimui) protokolai skaičiavimų tinklo specifinėms tinklo transakcijoms. Resursų lygmenyje esantys protokolai nurodo, kaip saugiai inicializuoti, stebėti ir kontroliuoti resursų dalijimo operacijas. Bendrajame paslaugų lygyje yra protokolai ir paslaugos, kurios įgyvendina bendravimą tarp resursų grupių. Kadangi šiame lygyje apjungiami komponentai iš siauresnio resursų ir ryšio lygmens, tai vartotojui pateikiama plati paslaugų aibė: resursų radimas ir priskyrimas, stebėjimų ir diagnostikos paslaugos, duomenų kopijavimo bei narystės nustatymo paslaugos. Viršutiniame lygyje yra vartotojo aplikacijos, kurios gali naudotis visomis žemesniuose lygmenyse esančiomis paslaugomis.

Skaičiavimų tinklas, kuris yra sudarytas tiesiogiai jungiant skaičiavimo mazgus, jų nejungiant į klasterius, vadinamas atliekamų resursų (angl. *cycle-scavenging*) skaičiavimų tinklu. Tipiniai tokio skaičiavimų tinklo pavyzdžiai yra savanoriškai atliekami skaičiavimai, kai bet koks vartotojas gali paskirti laisvus kompiuterio resursus tinklo skaičiavimams atlikti.

Sąvoka „skaičiavimų tinklas“ pirmą kartą pavartota 1997 metais, Ian Foster ir Carl Kesselman Argono Nacionalinėje laboratorijoje, Jungtinėse Amerikos valstijose, pradėjus dirbti projekte pavadinimu „*Building a Computational Grid*“.

Vienas didžiausių mokslinių tyrimų, atliktų naudojantis skaičiavimų tinklais, yra Oksfordo Universiteto Centro atliktas tyrimas, kurio metu buvo pasinaudota atliekamų resursų skaičiavimų tinklu [Ric02]. Projekto metu buvo išanalizuota 3.5 milijardo molekulių, tiriant jų galimybes vėžio gydymo srityje. Projekte dalyvavo daugiau nei 3 milijonai kompiuterių iš daugiau kaip 200 valstybių. Projekto metu buvo tiriamos esamos ir galimos molekulės, kurios galėtų blokuoti reikiamus baltymus sukuriant vėžio stabdymo ir gydymo vaistus.

Skaičiavimų tinklai plačiai naudojami įvairiose mokslų srityse: molekulių modeliavimo naujų vaistų gamybos ir tyrimo procesuose [BBG+02], smegenų aktyvumo tyrimams ir analizei atlikti [BDM+04], aukštos energijos fizikoje. Skaičiavimų tinklai naudojami ir Nacionalinėje aeronautikos ir kosmoso administracijoje, modeliuojant kuriamų kosminių laivų paleidimo mechanizmus, tyrinėjant žmonijos įtaką klimato sąlygoms, kuriant saugius ir efektyvius kosminius laivus [Bin05]. Taip pat išaugo skaičiavimo tinklų panaudojamumas ir verslo srityje [Yag04].

Egzistuoja nemažai skaičiavimo tinklų, vieni didžiausių projektų pasaulyje yra [GCP06]:

- *World Community Grid* – viešai prieinamas, daugiausiai orientuotas į mokslinius tyrimus, remiamas kompanijos *IBM*, įkurtas 2004 metais.
- *OurGrid* – atliekamais savanorių vartotojų resursais paremtas skaičiavimų tinklas, įkurtas 2004 metais.
- *BEinGRID* – skaičiavimų tinklas skirtas verslo procesams tirti ir gerinti, dalinai remiamas Europos Komisijos, įkurtas 2006 metais.
- *Grid.org* – organizacija, kuri kuria ir pati naudoja atvirojo kodo programinę įrangą, skirtą klasterių ir skaičiavimo tinklų sujungimui ir panaudojimui. Tyrimus ir darbus skaičiavimų tinklų srityje pradėjo nuo 2001 metų.
- *TeraGrid* – Čikagos universiteto 2004 metais įkurtas skaičiavimo resursų tinklas,
- *Open Science Grid* – Jungtinių Amerikos Valstijų universitetų ir laboratorijų sąjungos skaičiavimų tinklas, įkurtas 2004 metais, skirtas moksliniams tyrimams
- *EGEE* - svarbiausias projektas Europoje. Finansuojamas Europos komisijos lėšomis siekiant sukurti naują skaičiavimų tinklo technologiją ir paslaugų infrastruktūrą, kuri mokslininkams būtų pasiekama bet kuriuo metu.

Lietuvoje mokslinės organizacijos bendradarbiauja dviejuose skaičiavimų tinklų projektuose:

- *LitGrid* – tai Lietuvos akademinė institucijų lygiagrečių ir paskirstytų skaičiavimų tinklas.
 - Tinklo partneriai yra: Vilniaus, Kauno technologijos, Vilniaus Gedimino technikos, Klaipėdos, Vytauto didžiojo ir Šiaulių universitetai, bei VU Teorinės fizikos ir astronomijos, KMU psichofiziologijos ir reabilitacijos, Matematikos ir informatikos, Fizikos institutai.
 - Pagrindiniai sprendžiami uždaviniai: plonasienių konstrukcijų iš kompozicinių medžiagų tyrimas, elektroninės įrangos testavimo ir modeliavimo uždavinių sprendimas, optimizavimo uždavinių sprendimas.
- *BalticGrid* – projektas, savo gyvavimo pradžioje apėmęs tik penkias Baltijos valstybes, skirtas palaikyti ir skatinti atlikti mokslinius tyrimus naudojantis skaičiavimų tinklais.
 - Tinklo partneriai: šiuo metu (nuo 2008 gegužės 1 dienos) projekte dalyvauja 13 institucijų iš 7 Europos valstybių. Lietuvai šiame projekte atstovauja: Vilniaus Universitetas, Vilniaus Gedimino Technikos Universitetas ir Vilniaus universiteto Teorinės fizikos ir astronomijos institutas.

- Pagrindiniai sprendžiami uždaviniai: Baltijos jūros ekosistemos modeliavimas, žvaigždžių spektro tyrimas, telekomunikacijų modeliavimas.

Naudojantis skaičiavimų tinklą paslaugomis, naudojamos ir specialia tarpine programine įranga (angl. *middleware*). Priklausomai nuo konkretaus skaičiavimo tinklo, ši įranga gali skirtis. Populiariausios tarpinės programinės įrangos, naudojamos skaičiavimų tinkluose yra šios [CPP08]: *gLite*, *Globus Toolkit*, *OMII* ir kitos.

Tiriamajam darbui atlikti buvo pasirinktas *BalticGrid* skaičiavimų tinklas, kuris naudojami *gLite* tarpine programine įranga, todėl užduoties konkretus formulavimas ir užduočiai keliami reikalavimai yra apribojami *BalticGrid* skaičiavimų tinklo ir *gLite* programinės įrangos keliamų reikalavimų:

1. *Programavimo kalbos keliami reikalavimai kompiuteriniam modeliui* – efektyviam skaičiavimo tinklo panaudojimui geriausia naudoti *C*, *C++* arba *Fortran* kalbomis.
2. *Operacinės sistemos keliami reikalavimai* – užduoties dalys, kurios yra vykdomos atskiruose skaičiavimo mazguose, yra vykdomos *Linux* operacinės sistemos aplinkoje, todėl programa turi būti suderinama su šia operacine sistema.
3. *Užduoties formulavimo reikalavimai* – užduotis formuojama *Job Description Language* kalba [PB01, LCG08].

Kartu su užduotimi, naudojantis *Job Description Language*, gali būti nurodomi:

- programos išėjties tekstas – reikalinga tada, kai norima programą kompiliuoti skaičiavimų tinkle – tai aktualu tada, kai nėra tiksliai žinoma, ar programa veiks skaičiavimo mazgo aplinkoje.
- Sukompiliuota programa – nurodoma, kai tiksliai žinoma, kad programa veikia korektiškai bet kokioje aplinkoje.
- Duomenų bylos – viena ar daugiau bylų, kuriomis naudosis nusiųsta į skaičiavimų tinklą programa.
- Rezultatų bylos – bylos, kuriose pateikiami programos vykdymo rezultatai.
 - papildomai galima nurodyti bylos, į kurią norėsime įrašyti į ekraną išvestus rezultatus, vardą.
- Reikalavimai skaičiavimo mazgui – įvairūs reikalavimai, kuriuos turi atitikti kompiuteris, kuriame veiks galutinė programa. Dažniausiai nurodomi parametrai yra šie:
 - Reikalavimai procesoriui – gali būti nurodomi: branduolių skaičius, taktinis dažnis, procesoriaus architektūros tipas.

- Reikalavimai adresuojamai atminčiai (angl. *Random Access Memory, RAM*) – kiek turėtų skaičiavimų mazgas turėti adresuojamos atminties.
- Reikalavimai operacinei sistemai – aktualu tada, kai programa, siunčiama į skaičiavimų tinklą gali veikti tik konkrečioje operacinėje sistemoje, ar operacinių sistemų šeimoje. Reikalavimuose gali būti nurodoma operacinės sistemos šeima, pavadinimas, versija.

Užduotis, pateikiama skaičiavimų tinklui, aprašyta *Job Description Language* kalba, gali būti tokių tipų [LCG08]:

- Paprastoji užduotis – į skaičiavimų tinklą paduodama viena *Job Description Language* aprašyta užduotis.
- Darbų rinkinys – skaičiavimų tinklui paduodama grupė užduočių, kurios gali būti tarpusavyje skirtingos. Užduoties tipas ypatingas tuo, kad tokiu būdu atiduodant užduotį į skaičiavimų tinklą yra sutaupomas laikas, kuris būtų sugaištamas užduotis padavinėjant po vieną.
- Tiesioginių neciklinių grafų (angl. *Direct acyclic graphs*) – į skaičiavimų tinklą paduodama užduočių grupė, kuri tarpusavyje yra susijusi tam tikromis priklausomybėmis.
 - Laikoma, kad užduotis A priklauso nuo užduoties B, tada, kai A gali pradėti darbą tik tada, kai B užduotis pabaigė savąjį.
 - Toks užduoties tipas naudingas tada, kai dalis skaičiavimų gali būti atliekami tik turint tarpinius rezultatus.
- Parametrinė – tai darbų rinkinio porūšis, kai užduočių rinkinyje kiekviena užduotis yra identiška, tačiau turi skirtingus parametrus.
- Interaktyvi – toks užduoties tipas, kai darbai turi aktyvų ryšį su nuotoliniu būdu prisijungusiu vartotoju. Bendravimas vyksta nukreipiant standartinius duomenų srautus (*stdin, stdout, stderr*) į vartotojo kompiuterį.
- MPICH – žinučių perdavimo sąsaja (angl. *Message Passing Interface*) paremtos užduotys. Šios užduoties tipas kol kas tėra eksperimentinis, toliau atliekant tyrimus *MPI* panaudojimo galimybėms skaičiavimų tinkluose [PB01, LCG08].

Pagrindinės užduočių, pateikiamų į skaičiavimų tinklą naudojantis *gLite* programine įranga, valdymo komandos yra šios [LCG08]:

- *glite-job-submit* – vykdančią komandą, kaip parametras nurodomas užduoties bylos vardas lokaliame kompiuteryje. Komandos rezultatas – identifikatorius, nurodantis vartotojui unikalų užduoties numerį skaičiavimų tinkle.
- *glite-job-status* – vykdoma nurodant unikalų užduoties numerį, norint sužinoti užduoties statusą skaičiavimų tinkle. Komandos rezultatas – vienas iš šių statusų: *submitted*, *waiting*, *ready*, *scheduled*, *running*, *done*, *cleared*.
- *glite-job-cancel* – užduoties vykdymo nutraukimo komanda. Vykdoma nurodant unikalų užduoties numerį.
- *glite-job-output* – komanda, skirta įvykdytos užduoties rezultatų parsisiuntimui iš skaičiavimų tinklo.

Skaičiavimų tinklų nauda pasireiškia, kai reikia atlikti didelės apimties skaičiavimus, ir paprasto kompiuterio ar mažos kompiuterių grupės tam neužtenka. Nauda pasireiškia, nes skaičiavimai atliekami išskirstytai skaičiavimų tinkle, tačiau atliekant užduoties suskaidymą į mažesnes užduotis, perduodant konkrečią užduotį į jai priskirtą skaičiavimo mazgą (angl. *node*), tikrinant užduočių statusus, susigražinant ir apjungiant rezultatus yra jaučiamas tam tikras uždelsimas, kuris yra dažniausiai įtakojamas parinkto tinklo struktūros [Shi04].

2.2. Skaičiavimo tinklų panaudojimas kompiuteriniam modeliavimui

Kompiuterinis modeliavimas – tai metodas, kai sukuriamas realaus arba teorinio fizinio proceso sistemos modelis, ir tas modelis yra vykdomas naudojantis kompiuterine ir programine įranga siekiant išanalizuoti vykdymo rezultatus [Fis96].

Iš dalies galima teigti, kad pirmą kartą kompiuterinis modeliavimas praktiškai panaudotas *Maheteno projekte* 1944 metais, kuriant atominę bombą *Los Alamos nacionalinėje laboratorijoje*, Jungtinėse Amerikos valstijose [Koc05]. Naudojantis *Monte Carlo* metodu, buvo tiriamos neutronų trajektorijos ir skaičiuojamos tikimybės tolimesnėms reakcijoms – ar neutronas bus absorbuotas, ar pabėgs iš bombos, ar dalyvaus pradedant naują skilimo reakciją. Tokį tyrimą kompiuteriniu modeliavimu galima vadinti tik iš dalies todėl, kad skaičiavimams atlikti vis dar reikėjo itin daug žmogiškųjų pastangų. 1947 metais, John von Neumann ir Nicholas Metropolis atliko neutronų pernešimo tyrimus, kurių metu buvo įrodytas *Monte Carlo* metodų tinkamumas vandenilinės bombos gamybos procese, tuo pačiu panaudojant kompiuterinius modelius itin reikšmingoje srityje, kadangi branduolinė ir termobranduolinė fizika tuo metu nebuvo pajėgi tyrinėti šių procesų praktiškai [LANL07].

Nuo to laiko kompiuteriniai modeliai pradėti naudoti įvairiose mokslinių tyrimų srityse [Har05]: gamtos mokslų tyrėjai modeliuoja žvaigždžių ir galaktikų susikūrimą ir formavimąsi, aukštos energijos branduolinių reakcijų dinamiką, taip pat ir sudėtingus gyvybės vystymosi procesus, socialinių mokslų tyrėjai naudoja kompiuterinį modeliavimą tirdami ekonomikos progresą, sprendimų priėmimo procedūras versle ir panašius dalykus.

Operaciniuose tyrimuose (angl. *operational research*) kompiuterinis modeliavimas vykdomas tokiais etapais [Per08]:

- problemos formulavimas,
- modelio konstravimas,
- modelio validavimas,
- naudojantis modeliu nagrinėjamos galimos alternatyvos,
- Sprendimo įgyvendinimas ir palaikymas.

Bendrai, kompiuterinius modelius galima klasifikuoti į tokias grupes: portretinis, analoginis, ir simbolinis [Per08].

Portretinis modelis yra tiksli savybių kopija realaus pasaulio modelio. Pavyzdžiai tokių modelių yra modeliuojami lėktuvai, žemėlapiai ir pan. Analoginis modelis naudoja rinkinį savybių iš vienos sistemos, kad modeliuotų kitą, pavyzdžiui hidraulinė sistema gali būti panaudota kaip analogas modeliuojant eismo ir tam tikras ekonomines sistemas. Simboliniai modeliai realios sistemos savybes atvaizduoja per simbolius, pavyzdžiui per matematinės formules. Toliau darbe remiamasi būtent tokiu kompiuteriniu modeliu.

Kompiuteriniai modeliai yra ypatingi tuo, kad modeliuoja realaus pasaulio sistemas, modeliavimui naudodami parametrų rinkinius, kurie nusako realaus objekto ar proceso savybes [Mul05]. Nagrinėjant biokatalizės procesus, tyrimuose dažniausiai pasitaikanti užduotis yra ištirti vieno ar kito parametro įtaką proceso eigai ar rezultatui, tyrimus atliekant visoje fiziškai prasmingų nagrinėjamų parametrų aibėje [Aše07, BIK03a, BIK03b]. Tokiu būdu, realus procesas, naudojantis kompiuteriniu modeliu, yra modeliuojamas tūkstančius [Aše07], dešimtis tūkstančių [Gai06] ir šimtus tūkstančių kartų – natūralu, kad naudojantis vienu skaičiavimo mazgu, modeliavimo rezultatų gavimas yra ilgas procesas, tuo pačiu uždelsiantis viso proceso tyrimą.

Pasaulyje skaičiavimų tinklai matematinio ir kompiuterinio modeliavimo srityje yra naudojami įvairiose srityse: modeliuojant skysčių ir temperatūros pasiskirstymą medžiagose [Hon04], kosminių laivų gamyboje ir jų veikimo simuliacijose [Bin05], tyrinėjant klimato pokyčius

[CHJ+03], nagrinėjant įvairias ligas ir ieškant naujų vaistų [BSG05] ir įvairiose kitose mokslo srityse. Lietuvoje skaičiavimų tinklai šioms modeliavimo uždaviniams spręsti [Iva04]:

- Deformuojamų kūnų skverbimosi procesui į tekstilines architektūras modeliuoti.
- Skaičiavimo metodai ir aplikacijos balistikos analizei.
- Skaitiniai algoritmai biomedicinoje.
- Savaimė besiformuojančių objektų modeliavimas mikro ir nano skalėse.
- Standžiųjų junginių judėjimas biologinėse aplinkose.
- Antros ir trečios eilių netiesinių optinių efektų modeliavimas heterogeninėse aplinkose.
- Lazerių, paremtų daugiasluoksnių puslaidininkių perjungėjų schemomis, generavimo dinamikos modeliavimas.

Parametrizuojamų programų, tame tarpe ir kompiuterinių modelių, automatizuotam vykdymui skaičiavimų tinkle yra sukurtos specialios programų sistemos, viena iš jų *APST* [CB02]. Ši programų sistema yra skirta darbui su parametru eilėmis, ir buvo sukurta siekiant dviejų tikslų: iširti ir patikrinti adaptyvų darbų paskirstymą bei remiantis gautais rezultatais pateikti šios programų sistemos naudotojams patogią vartotojo sąsają. *APST* yra pritaikyta darbui su parametru rinkiniais, o vartotojai užduotis skaičiavimų tinklui perduoda suformuluodami *XML* užduočių bylas, kuriose pateikia programos vardą bei jos parametrus. Vartotojai suformuotas užduotis perduoda per vieną aplikaciją – klientą, o užduočių būsenas stebi tarpinė aplikacija, kuri naudojami skaičiavimų tinklo programine įranga. Didelis dėmesys *APST* programų sistemoje yra skiriamas darbų paskirstymo efektyvumui, kuris yra gan glaudžiai susijęs su konkrečiu skaičiavimų tinklo tarpine programine įranga (angl. *middleware*). Nors *APST* palaiko ne vieną skaičiavimų tinklą tarpinę programinę įrangą, tačiau dėl tiesioginės sąsajos su šia PĮ, *APST* negali būti taikoma plačiai, nes ji yra susieta su konkrečiomis architektūromis.

Kitos programų sistemos, tokios kaip *ILAB* [YMB+02] ir *Nimrod/G* [AGK00], taip pat leidžiančios automatizuotai pasinaudoti skaičiavimo tinklų resursais pasižymi panašiomis savybėmis. *Nimrod/G* naudojami *Globus* tarpine programine įranga, todėl taip pat negali būti taikytina visuose skaičiavimų tinkluose. *ILAB* programinė įranga yra sukurta NASA specialiai NASA skaičiavimų tinklui, todėl taip pat yra apriboto panaudojimo programinė įranga.

2.3. Apibendrinimas

Kompiuteriniai modeliai pasižymi tuo, kad jie yra parametrizuojami, o tokioms užduotims spręsti skaičiavimo tinkluose įdiegtos tarpinės programinės įrangos turi specialų užduočių tipą, kuris leidžia efektyviai atlikti kompiuterinio modeliavimo procesą. Automatizuotai atlikti kompiuterinio

modeliavimo procesą galima naudojantis jau egzistuojančiomis programų sistemomis. Tačiau jos yra glaudžiai susijusios su skaičiavimo tinkluose įdiegta tarpine programine įranga. Šiame darbe pateikiama metodika, kuri yra nepriklausoma nuo skaičiavimų tinkluose įdiegtos tarpinės įrangos, bei šios metodikos įgyvendinimas, skirtas biokatalizės proceso analizei.

3. Skaičiavimo tinklų panaudojimo biokatalizės procesams tirti metodika

3.1. Reikalavimai kompiuteriniams modeliams

Siekiant, kad programinė įranga efektyviai pasinaudotų skaičiavimų tinklais, tikslinga apibrėžti kriterijus, pagal kuriuos yra vertinamas PĮ tinkamumas. Ar kompiuterinis modelis yra pritaikytas naudotis skaičiavimų tinklais, buvo vertinama pagal šiuos kriterijus:

1. *Programavimo kalba.* Tikslinga nustatyti, kokią programavimo kalbą naudingiausia pasirinkti, bei kokių reikalavimų reikia laikytis kuriant išeities tekstus. Tinkamas programavimo kalbos pasirinkimas užtikrina sklandų programinės įrangos, besinaudojančios skaičiavimų tinklais, veikimą. Tuo tarpu netinkamai pasirinkta programavimo kalba, ar netinkamas išeities tekstų sukūrimas gali reikšti ne tik neprognozuojamą programinės įrangos elgseną, bet ir visišką jos neveikimą skaičiavimų tinklo aplinkoje.

2. *Programinės įrangos sąsaja su operacine sistema.* Skaičiavimų tinklo mazguose (kompiuteriuose, kurie atlieka jiems paskirtą dalį užduoties) gali būti įvairios operacinės sistemos, todėl reikia įvertinti, kokius reikalavimus turi atitikti PĮ, kad skaičiavimai būtų atliekami be trikdžių, kylančių dėl nesuderinamumo su operacine sistema.

3. *Duomenų įvestis ir išvestis.* Kompiuterinį modelį realizuojanti programinė įranga beveik visuomet turi tam tikrus jai perduodamus pradinis duomenis, ir visuomet turi išvestį – modeliavimo rezultatą. Siekiant, kad kompiuterinis modelis galėtų naudotis skaičiavimų tinklais, reikia nustatyti, kokius reikalavimus, keliamus įvesčiai ir išvesčiai, turi tenkinti programinė įranga.

4. *Programinės įrangos veikimas.* Aktualu apibrėžti, kaip turi būti paleidžiama programinė įranga ir kaip turi būti elgiamasi su vykdymo metu iškylančiomis klaidomis, siekiant užtikrinti kuo efektyvesnį skaičiavimo tinklų panaudojimą.

Pagal šiuos kriterijus, buvo nustatyti reikalavimai, kuriuos turi atitikti kompiuteriniai modeliai, siekiant efektyviai naudotis skaičiavimų tinklais:

1. Programavimo kalbai keliami reikalavimai:

- Modelius tikslinga rašyti *C*, *C++* arba *Fortran* programavimo kalbomis – išeities tekstus, parašytus šiomis kalbomis, galima sukompiliuoti, o taip ir vykdyti nepriklausomai nuo operacinės sistemos. Šios programavimo kalbos palaikomos daugelyje pasaulio skaičiavimo tinklų.

- Naudoti tik standartines, veikiančias visose platformose, programavimo kalbos komandas ir bibliotekas. Būtina išvengti specifinių programavimo kalbos elementų panaudojimo, kurių veikimas yra apribojamas operacinės sistemos. To reikia siekiant išvengti programos nekorektiško veikimo aplinkoje, kurioje programa buvo netestuota.
 - Programinę įrangą kurti taip, kad ji būtų nepriklausoma nuo kompiliatoriaus versijos. Realizuojant kompiuterinį modelį reikia vengti programinės įrangos susiejimo su konkrečia kompiliatoriaus versija, nes kompiliatoriaus versijos skirtinguose skaičiavimų mazguose gali skirtis.
2. Programinės įrangos sąsajai su operacine sistema keliami reikalavimai:
- Programos kompiliavimo ir vykdymo rezultatas turi nepriklausyti nuo operacinės sistemos. Reikia sukurti universalų kompiuterinį modelį – tokį, kuriuo galima efektyviai naudotis nepriklausomai nuo skaičiavimų tinklo programinės ir operacinės įrangos.
 - Kompiuterinis modelis turi visiškai nepriklausyti nuo operacinės sistemos nustatymų – atskiros užduotys atliekamos skirtinguose skaičiavimų tinklo mazguose, kuriuose operacinės sistemos nustatymai gali būti skirtingi. To reikia siekiant išvengti netikėto programos veikimo esant, pavyzdžiui, skirtingiems regioniniams nustatymams.
3. Duomenų įvesties ir išvesties reikalavimai:
- Kompiuterinį modelį orientuoti taip, kad kompiuterinio modelio parametrus būtų galima keisti nemodifikuojant paties kompiuterinio modelio. Taip patartina daryti todėl, kad taip orientuoti modelio duomenys gali būti sukuriami dinamiškai.
 - Įvesties ir išvesties procesas turi būti realizuotas taip, jog nuskaitymas nepriklausytų nuo operacinės sistemos tipo.
 - Kompiuteriniam modeliui perduodant parametrus per bylą, reikia atsižvelgti į tai, kad skaičiavimų tinklui atiduodamos užduoties suminis dydis yra ribojamas. Esant poreikiui, duomenų bylą galima archyvuoti prieš perduodant į skaičiavimų tinklą, o po to išsiarchyvuoti reikiamas jos dalis konkrečiuose skaičiavimo mazguose.
4. Veikimo reikalavimai:
- Programinę įrangą reikia kurti taip, kad ją būtų galima vykdyti naudojantis vien tik komandine eilute, nes programos yra vykdomos automatizuotai.
 - Programa turi nagrinėti vykdymo metu kylančias klaidas ir išvesti jas į ekraną – visą informaciją, kuri vykdymo metu buvo išvesta į ekraną, galima peržiūrėti programai

baigus darbą skaičiavimo mazge nenaudojant papildomų priemonių. Naudojant bet koki kitą klaidų apdorojimo mechanizmą, reikalingos papildomos pastangos ir priemonės, kurios apsunkina efektyvų skaičiavimo tinklų panaudojimą.

3.2. Skaičiavimo tinklų panaudojimo schema

Kompiuteriniai modeliai pasižymi tuo, kad yra parametrizuojami – modelio parametrus atitinka konkrečios realaus arba teorinio fizinio proceso sistemos savybės. Tokių parametrinių skaičiavimų esmė yra priklausomybės tarp pradinių duomenų ir kompiuterinio modeliavimo metu gautų rezultatų tyrimas ir analizavimas [APL06]. Tiriant parametrų įtaką vienai iš modeliujamo proceso charakteristikų, galime tai aprašyti taip:

$$y = f(p_1, p_2, \dots, p_n), \quad (3.1)$$

čia y – tiriama modeliujamo proceso savybė, p_1, p_2, \dots, p_n – parametrai, kurių įtaką reikia tirti, o f žymi funkciją, atliekančią modeliavimo procesą.

Atliekant išsamią proceso analizę, naudojamosi parametrų eilėmis (angl. *parameter sweeps*) – tokiu atveju skaičiavimai atliekami pakartotinai naudojant kompiuterinį modelį, tik kas kartą su skirtingomis parametrų reikšmėmis [Abr05]. Tokiu atveju, yra sukuriami parametrų rinkiniai. Parametrų rinkinių reikšmės yra arba išvardinamos, arba generuojamos pasinaudojant parametrų rinkinių generavimo įrankiu. Rinkinių generavimo atveju aprašant kiekvieną parametą yra nurodoma [SBA+04, APL06]:

- parametro galimų reikšmių intervalas:

$$p_A \leq p_i \leq p_B, \quad (3.2)$$

čia p_A – tiriamų parametro reikšmių pradinis intervalo rėžis, p_B – galinis intervalo rėžis, p_i – parametras iš parametrų rinkinio, $i = 1, 2, \dots, n$.

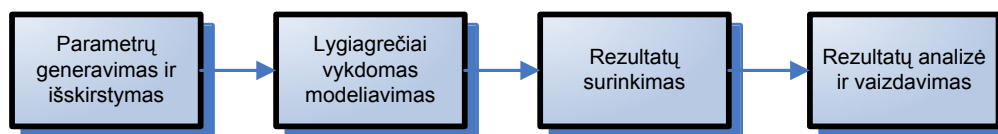
- Kokia progresija (aritmetinė ar geometrinė) yra naudojama generuojant konkretaus parametro reikšmes nagrinėjamame intervale.
- Aritmetinės progresijos skirtumas arba geometrinės progresijos vardiklis.

Kadangi biokatalizės procesą modeliujančių kompiuterinių modelių užduotys taip pat yra parametrizuojamos, naudojantis jau apžvelgtomis užduočių formavimo galimybėmis skaičiavimų tinklui, galima teigti, kad siekiant ištirti skaičiavimo tinklų panaudojamumą kompiuteriniam biokatalizės proceso modeliavimui, efektyviausiai galima pasinaudoti parametrinių darbų užduoties tipu [PB01, LCG08], į skaičiavimų tinklą paduodant grupę užduočių:

$$U = \{prog, param_eil\}, \quad (3.3)$$

čia U – užduočių aibė, $prog$ – kompiuterinį modelį realizuojanti programa, $param_eil$ – parametų rinkinių eilė, kurių kiekviena yra sudaryta iš $\{p_1, p_2, \dots, p_n\}$ parametų rinkinio.

Apibendrintai, kompiuterinį modeliavimą, atliekamą skaičiavimų tinklų pagalba, galima pavaizduoti schematiškai (2 pav.): pirmiausiai yra sugeneruojama visa modelio parametų aibė atsižvelgiant į nurodytus kiekvieno parametro intervalus (3.2). Parametų rinkiniai yra suskaidomi į parametų eiles, ir yra vykdomas lygiagretus kompiuterinio modeliavimo procesas, skaičiavimų tinkle vykdant atskiras užduotis (3.3). Pasibaigus modeliavimo procesui rezultatai yra surenkami, atliekama jų analizė ir atvaizdavimas.



2 pav. Kompiuterinio modeliavimo veiksmų seka naudojantis skaičiavimų tinklais.

3.3. Metodikos santrauka

Skaičiavimo tinklų panaudojimo metodika, sprendžiant parametro įtakos procesui uždavinius, yra aprašoma nuosekliai atliekamais veiksmais:

1. Paruošti kompiuterinį modelį, atitinkantį 3.1. skyriuje aprašomus reikalavimus. Tik tie kompiuteriniai modeliai, kurie tenkina efektyvaus naudojimosi skaičiavimų tinklais reikalavimus gali tinkamai pasinaudoti skaičiavimų tinklo resursais.

2. Apsibrėžti modelio parametų rinkinių formavimo taisykles:

- 2.1. Jei rinkiniai formuojami reikšmių išvardinimo būdu, tada tiksliai apibrėžti, su kokiomis parametų reikšmėmis norime atlikti tyrimą.

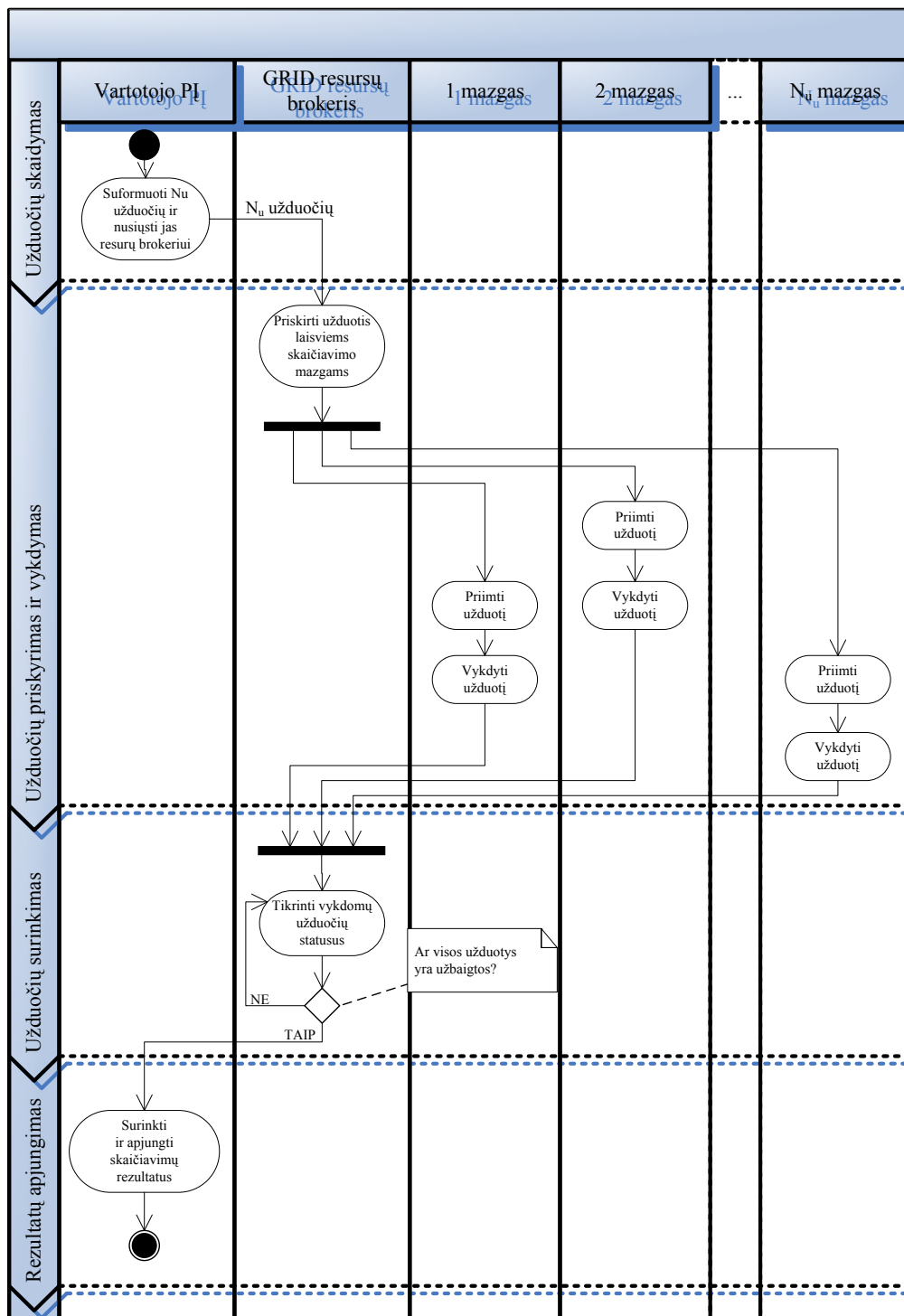
- 2.2. Jei parametų rinkiniai yra generuojami, tada reikia apibrėžti: tiriamų parametro reikšmių pradinį ir galinį režius p_A ir p_B , progresijos, naudojamos parametų reikšmėms generuoti, rūšį ir aritmetinės progresijos skirtumą arba geometrinės progresijos vardiklį.

3. Pagal apibrėžtas taisykles paruošti parametų rinkinius. Pažymėkime, kad iš viso yra N_{rink} parametų rinkinių, sudarytų iš parametų p_1, p_2, \dots, p_n , skaičius.

4. Parametų rinkinius suskaidyti į N_u segmentų. Skaidant siekiama, kad parametų rinkinių eilės, gaunamos kiekvienam segmentui būtų panašios apimties. Kiekvieno segmento parametų rinkinių skaičius $N_{i,rink}$ (čia $i = 1, 2, \dots, N_u$) turi būti kiek galima artimas visų parametų rinkinių segmentų vidutiniam parametų rinkinių skaičiui:

$$\frac{N_{rink}}{N_U} - 1 \leq N_{i,rink} \leq \frac{N_{rink}}{N_U} + 1. \quad (3.4)$$

5. Į skaičiavimų tinklą atiduoti užduočių grupę (3.3), sudarytą iš N_u užduočių.
 6. Skaičiavimų tinkle atliktų skaičiavimų rezultatus surinkti į vieną rezultatų bylą.
- Grafiškai, 4-6 veiksmai pavaizduojami tokia funkcinė diagrama (3 pav.)



3 pav. Funkcinė užduoties atlikimo diagrama.

3.4. Metodikos įgyvendinimas

Metodika buvo įgyvendinta taip:

1. Parametrų reikšmėms generuoti, nurodomi nagrinėjamų parametrų intervalai, progresijos rūšis, bei jos vardiklis.

2. Skaidant parametrų rinkinius į N_u segmentų, suskaičiuojama, kiek vidutiniškai parametrų rinkinių turėtų būti viename segmente:

$$N_{vid} = \frac{N_{rink}}{N_U}. \quad (3.5)$$

Pažymėkime, kad $N_{vid,sv}$ – sveikoji vidurkio N_{vid} dalis. Tada į A skaičių segmentų priskiriamas $N_{vid,sv}$ skaičius parametrų rinkinių, o į B skaičių segmentų – skaičius $N_{vid,sv}+1$. Čia A ir B yra tokie sveikieji skaičiai, kad būtų tenkinama lygybė:

$$N_{rink} = AN_{vid,sv} + B(N_{vid,sv} + 1). \quad (3.6)$$

3. Į skaičiavimų tinklą atiduodama užduočių grupė (3), sudaryta iš N_u užduočių. Kiekviena užduotis susideda iš modeliavimo programinės įrangos ir duomenų bylos, kurioje įrašytas parametrų rinkinių segmentas.

4. Kiekvienos užduoties rezultatai įrašomi į rezultatų bylą.

5. Po kiekvienos užduoties įvykdymo jos rezultatai yra apjungiami su bendrais rezultatais.

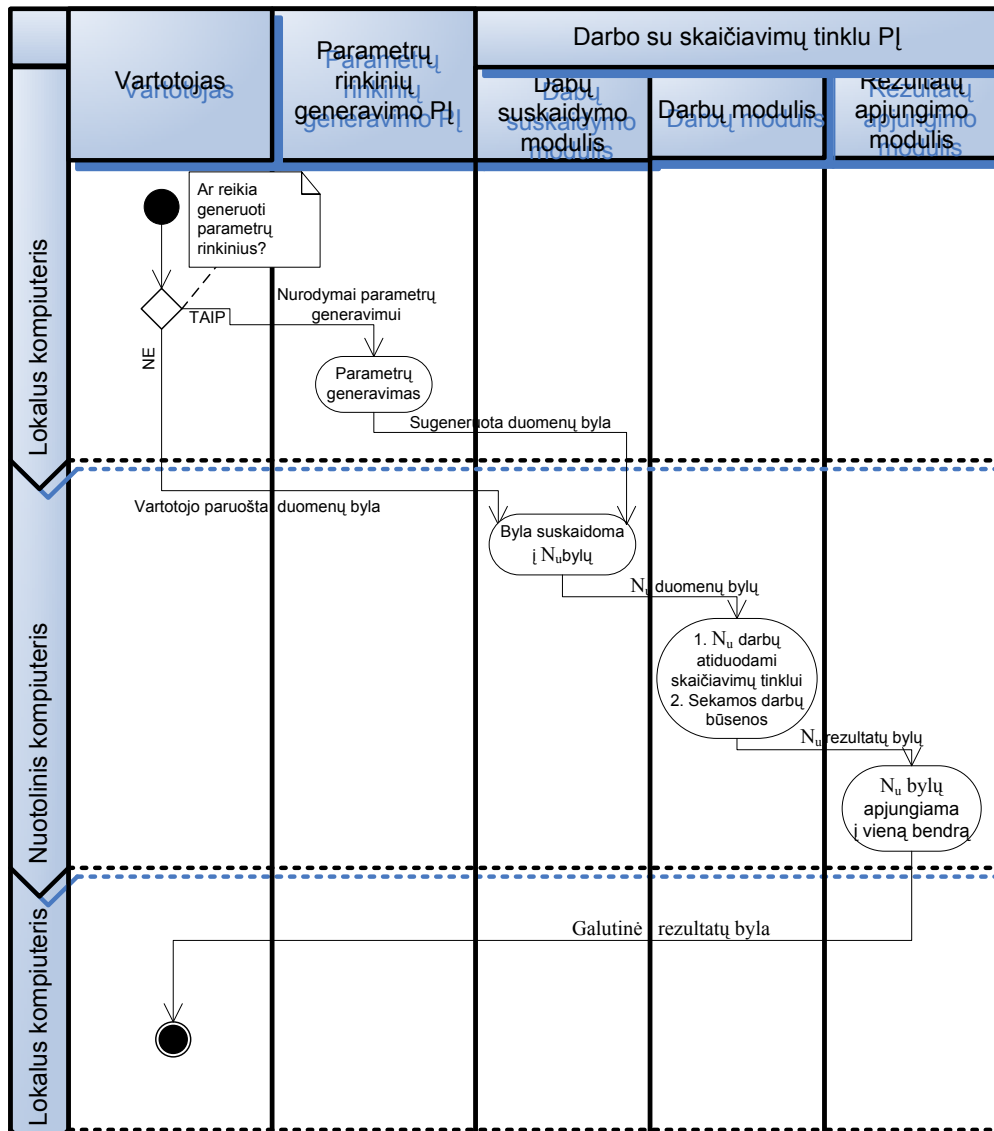
Metodiką užtikrinanti programinė įranga susideda iš šių dalių:

- Parametrų rinkinių generavimo įrankis – naudojantis šiuo įrankiu, pagal parinktą pradinį intervalo rėžį p_A , galinį intervalo rėžį p_B , naudojamą progresiją bei jos vardiklį, sugeneruojami parametrų rinkiniai.

- Darbo su skaičiavimų tinklu programinė įranga, kuri realizuota taip, kad ja galima pasinaudoti nuotoliniu būdu, naudojantis interneto naršykle. Naudojantis šia PI, automatiškai atliekami tokie veiksmai:

- parametrų rinkinys suskaidomas į N_u dalių.
- Į skaičiavimų tinklą atiduodama N_u užduočių grupė (3.3).
- Stebimos ir atnaujinamos užduočių būsenos.
- Į bendrą rezultatų bylą surenkami atskirų užduočių vykdymo rezultatai.

Darbo su skaičiavimų tinklu programinė įranga susideda iš trijų dalių: darbų suskaidymo, darbų vykdymo ir darbų apjungimo modulių. Modulių sąveika ir atliekamos funkcijos pavaizduotos 4 pav.

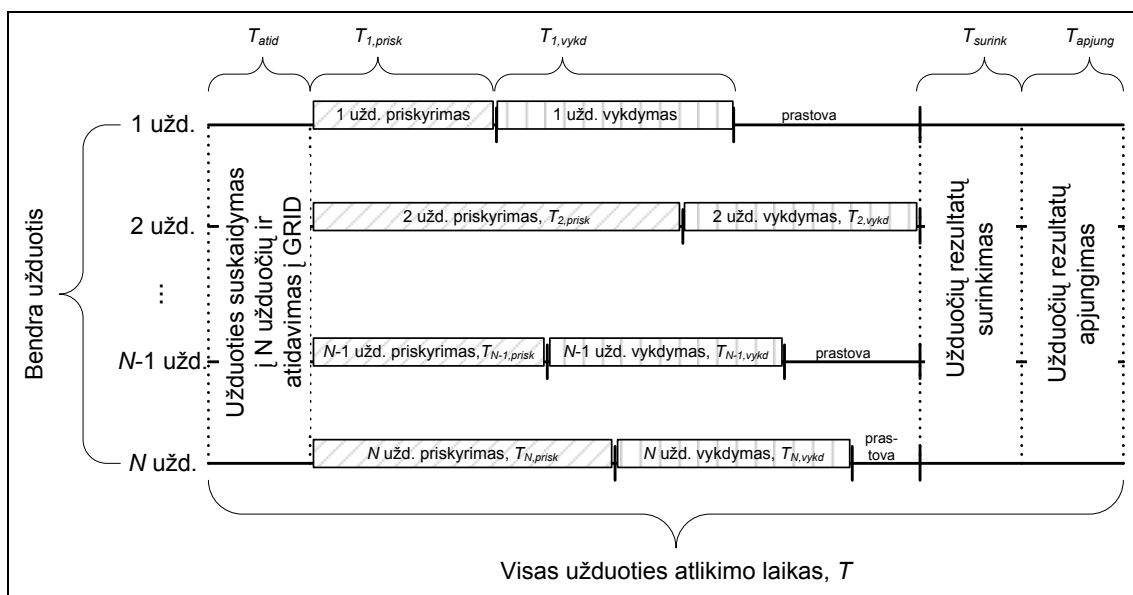


4 pav. Metodiką užtikrinančios PĮ modulių ryšiai.

Kaip matyti iš 4 pav., su skaičiavimų tinklu sąveikauja tik darbų vykdymo modulis. Norint šią programinę įrangą pritaikyti konkrečiai skaičiavimų tinklo architektūrai, reikia modifikuoti tik šį modulį. Modulyje esančios darbų atidavimo į skaičiavimų tinklą, darbų statusų tikrinimo, bei rezultatų surinkimo iš kiekvieno skaičiavimo mazgo komandos yra skirtingos kiekvienai skaičiavimų tinklo programinei įrangai.

3.5. Metodikos tyrimas

Suskaidžius skaičiavimus į N_u užduočių, visas skaičiavimams skirtas laikas gali būti pavaizduojamas taip (5 pav):



5 pav. Užduoties vykdymo trukmė.

Iš 5 pav. matyti, kad užduoties atlikimo laikas T , per kurį įvykdoma visa užduotis yra lygus:

$$T = T_{atid} + \max_{1 \leq i \leq N_U} (T_{i,prisk} + T_{i,vykd}) + T_{surink} + T_{apjung}, \quad (3.7)$$

čia T_{atid} – laikas, per kurį užduotis yra suskaidoma ir nusiunčiama į skaičiavimų tinklą, T_{surink} – laikas, per kurį užduoties rezultatai yra surenkami į vieną skaičiavimų mazgą, T_{apjung} – laikas, per kurį skaičiavimų rezultatai yra apjungiami ir gražinami vartotojui, $T_{i,prisk}$ – laikas, per kurį i -oji užduotis buvo pradėta vykdyti, $T_{i,vykd}$ – laikas, kiek truko užduoties vykdymas skaičiavimų mazge, $i = 1, 2, \dots, N-1, N_U$.

Norint efektyviai pasinaudoti skaičiavimų tinklų resursais, pirmiausia reikia nustatyti, kokia turi būti minimali užduoties atlikimo trukmė lokaliame skaičiavimų kompiuteryje, kad užduotį būtų verta atlikti naudojantis skaičiavimų tinklais. Jei skaičiavimai lokaliame kompiuteryje su parametru rinkiniu p_1, p_2, \dots, p_n atliekami per laiką T_{Irink_lok} , tai reikia nustatyti, su kokia reikšme skaičiavimų trukmė $N_U T_{Irink_lok}$ skaičiavimų tinkle bus mažesnė nei lokaliame.

3.5.1. Minimalios efektyviai sprendžiamos skaičiavimų tinkle užduoties apimtis

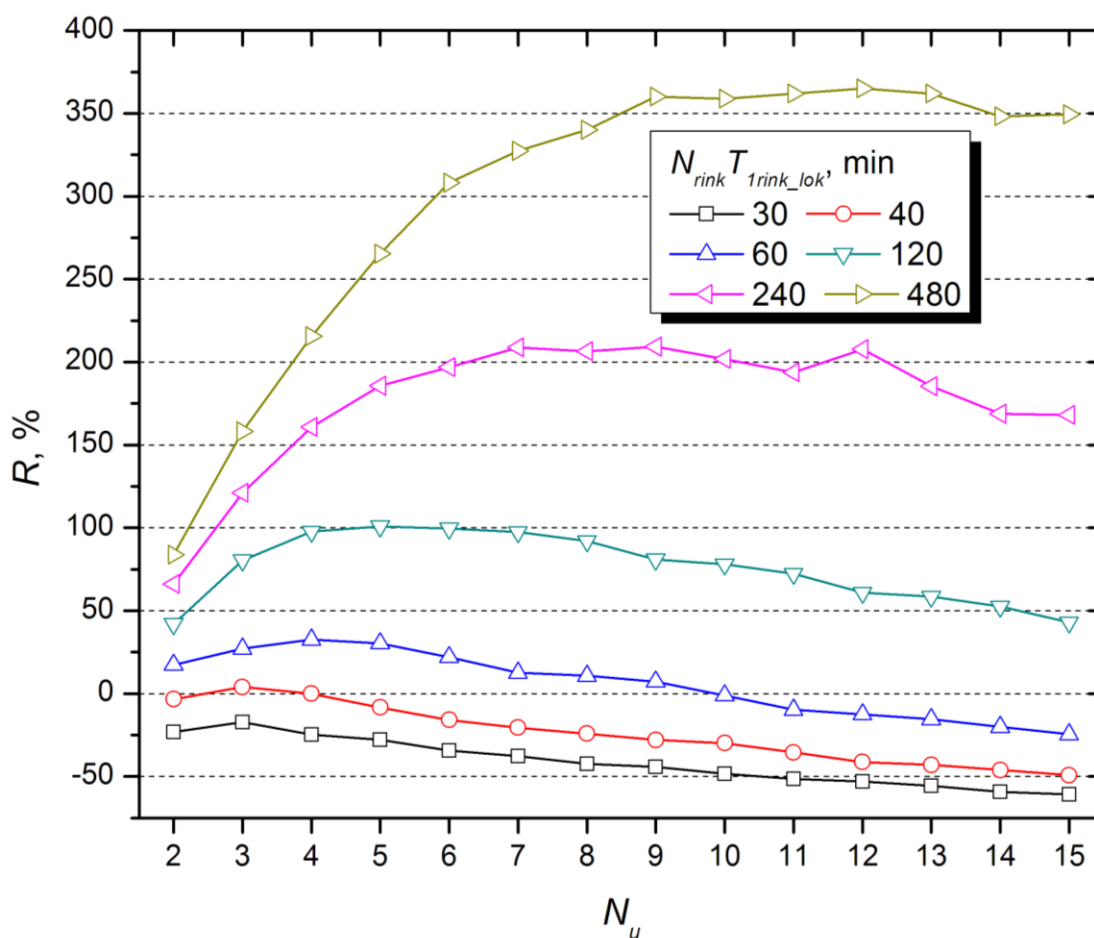
Jei skaičiavimai lokaliame kompiuteryje su parametru rinkiniu p_1, p_2, \dots, p_n atliekami per laiką T_{Irink_lok} tai aktualu žinoti, su kokia T_{Irink_lok} reikšme skaičiavimų trukmė $N_U T_{Irink_lok}$ skaičiavimų tinkle bus mažesnė. Taip pat reikia nustatyti, į kiek užduočių N_U geriausia skaidyti mažos apimties užduotis.

Laikas T_{Irink_lok} buvo matuojamas kompiuteryje, kurio procesoriaus taktinis dažnis yra 2.16 ghz, o RAM atminties kiekis – 1 GB. Atliekant matavimus, buvo vertinama, kiek sumažėja

(kraštiniais atvejais – kiek padidėja) užduoties vykdymo laikas T skaičiavimų tinkle, lyginant su skaičiavimais, kurie atliekami lokaliame kompiuteryje. Skaičiavimų trukmės pokytis išreiškiamas taip:

$$R = \frac{N_{rink} T_{1rink_lok} - T}{T} \cdot 100\% \quad (3.8)$$

Metodika buvo įgyvendinta skaičiavimų tinkle *BalticGrid*. Skaičiavimai buvo atliekami naudojant skaidymo dydžius N_u nuo 2 iki 15, ir tokioms skaičiavimų trukmėms $N_{rink} T_{1rink_lok}$: 30, 40, 60, 120, 240 ir 480 min. Skaičiavimų trukmės priklausomybė nuo šių dviejų dydžių pateikiama 6 ir 7 pav. Metodikos efektyvumo tyrimas atliktas pakartotinai vykdant skaičiavimus, ir apskaičiuojant trukmės vidurkius. Atskirais atvejais užduoties atlikimo trukmė T ženkliai skyrėsi nuo kitų trukmių, todėl 6 pav. pastebimi grafikų netolydumai, pavyzdžiui kai $N_{rink} T_{1rink_lok} = 240$ min, ir $N_u = 11$.



6 pav. Metodikos efektyvumo priklausomybė nuo skaidymo dydžio.

Iš 6 pav. matyti, kad metodikos efektyvumas priklauso nuo užduoties apimties – kuo užduotis didesnė, tuo efektyviau galima pritaikyti darbe aprašomą metodiką, skirtą pasinaudoti skaičiavimų tinklais. Minimalios užduoties trukmė, kuri efektyviai sprendžiama naudojantis skaičiavimų tinklais,

yra apytiksliai 40 minučių. Tokią užduotį suskaidžius į $N_u = 3$ užduočių, metodika taikoma efektyviai, tačiau su nedideliu, $R = 4\%$ pagreitėjimu. Skaidant tokią užduotį į daugiau dalių, metodika tampa nebeefektyvia, ir užduotis skaičiavimų tinkle atliekama lėčiau nei lokaliai, nes

vidutinis užduočių perdavimo konkrečioms skaičiavimų mazgams laikas $\bar{T}_{i,prisk} = \frac{\sum_{i=1}^{N_u} T_{i,prisk}}{N_u}$ išauga

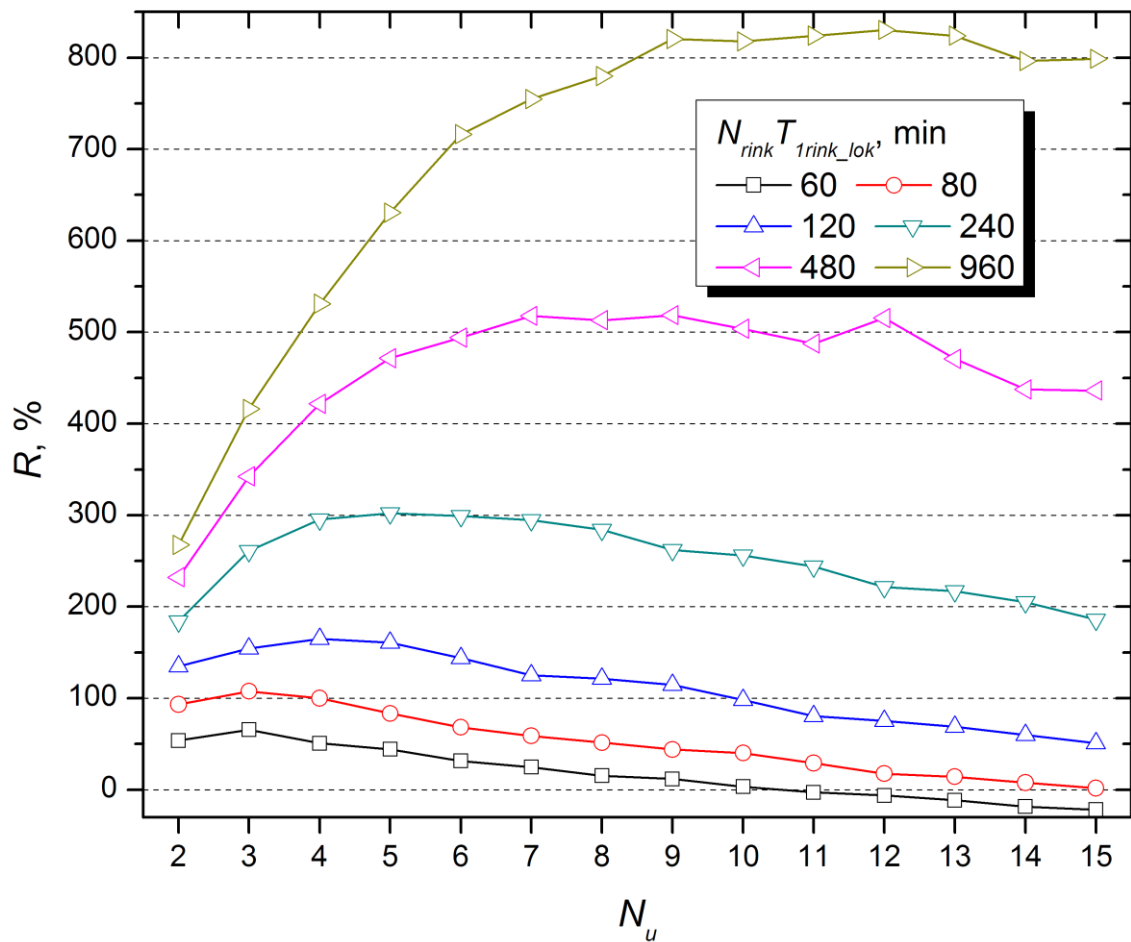
labiau, nei sutrumpėja vidutinis užduočių vykdymo laikas $\bar{T}_{i,vykd} = \frac{\sum_{i=1}^{N_u} T_{i,vykd}}{N_u}$. 120 min. užduotis

efektyviai sprendžiama su visais nagrinėtais N_u , tuo tarpu 60 min. užduotį skaidant į daugiau nei 10 dalių ji sprendžiama nebeefektyviai. Pastebima, kad 8 valandų trukmės užduotis su visais nagrinėtais N_u yra sprendžiama efektyviai, tačiau geriausiai tada, kai $N_u \in [9;13]$. Toks santykinai mažas skaidymo dydis yra paaiškinamas tuo, kad laikas $T_{i,prisk}$, per kurį i -oji užduotis pradeda vykdyti (5 pav.) ir laikas T_{surink} , per kurį yra surenkami užduoties rezultatai, yra priklausomi nuo dviejų dydžių:

1. Tarpinės programinės įrangos skaičiavimų tinkle uždelsimo. Tai laikas, kurį tarpinė programinė įranga užtrunka, kol konkrečiam skaičiavimų tinklo mazgui priskiria užduotį bei įvykdžius užduotį surenka rezultatus. Šis laikas visų eksperimentų metu yra daugmaž toks pats ir yra priklausomas nuo skaičiavimų tinklo konfigūracijos ir naudojamos tarpinės programinės įrangos.

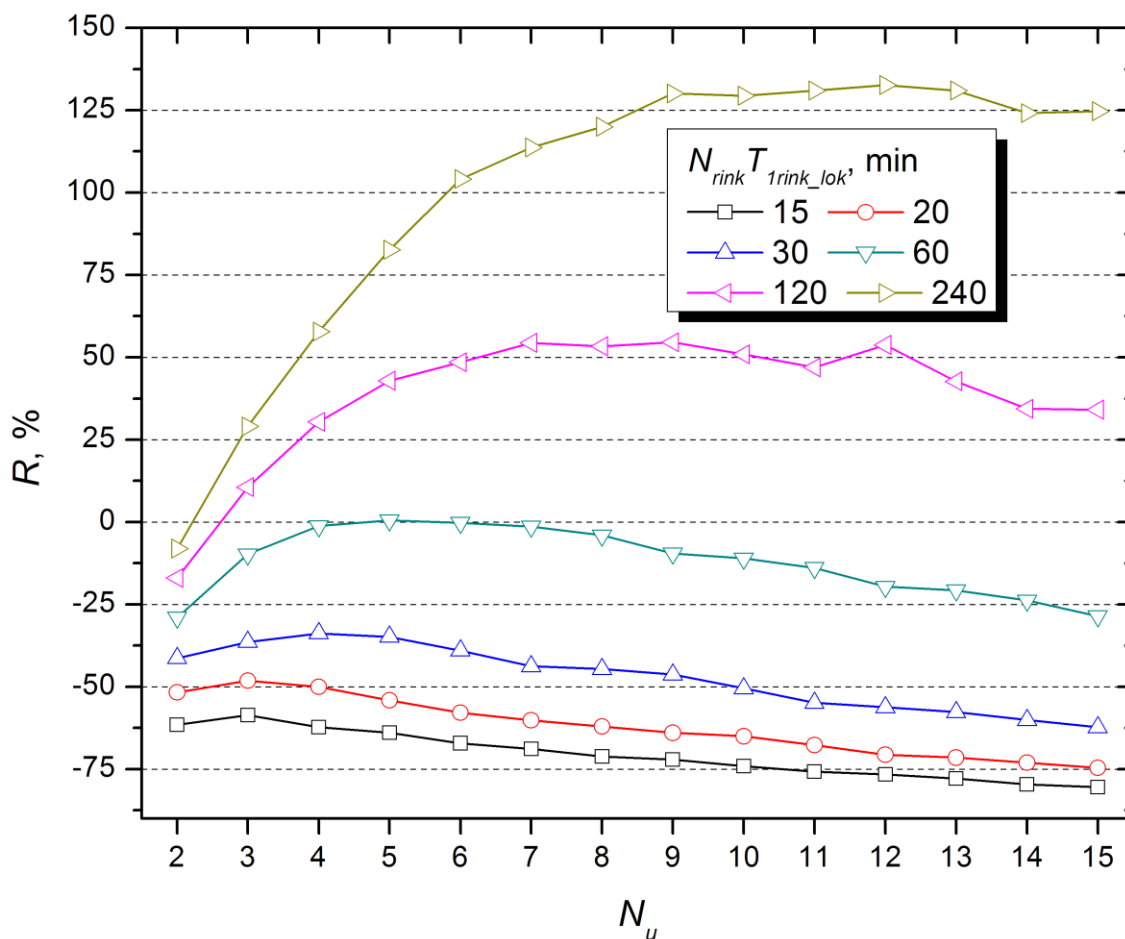
2. Atsitiktinio uždelsimo, kuris atsiranda, kai viena ar daugiau užduočių baigiamos vykdyti vėliau nei kitos. Bendra užduoties atlikimo trukmė padidėja, jei bent vienas iš užduočių atlikti priskirtų skaičiavimo mazgų tuo metu atlieka kitus skaičiavimus. Tokiu atveju priskirta užduotis pradeda vykdyti tik pasibaigus ankstesniajai užduočiai. Taip pat šis uždelsimas gali būti nulemtas ir žymiai lėtesnio skaičiavimų mazgo. Šis atsitiktinis uždelsimas yra priklausomas nuo skaičiavimų tinklo užimtumo bei naudojamos aparatūrinės įrangos.

Skaičiavimų trukmės pokytis buvo perskaičiuotas ir dvigubai lėtesniam lokaliai kompiuteriui, rezultatai pateikiami 7 pav.



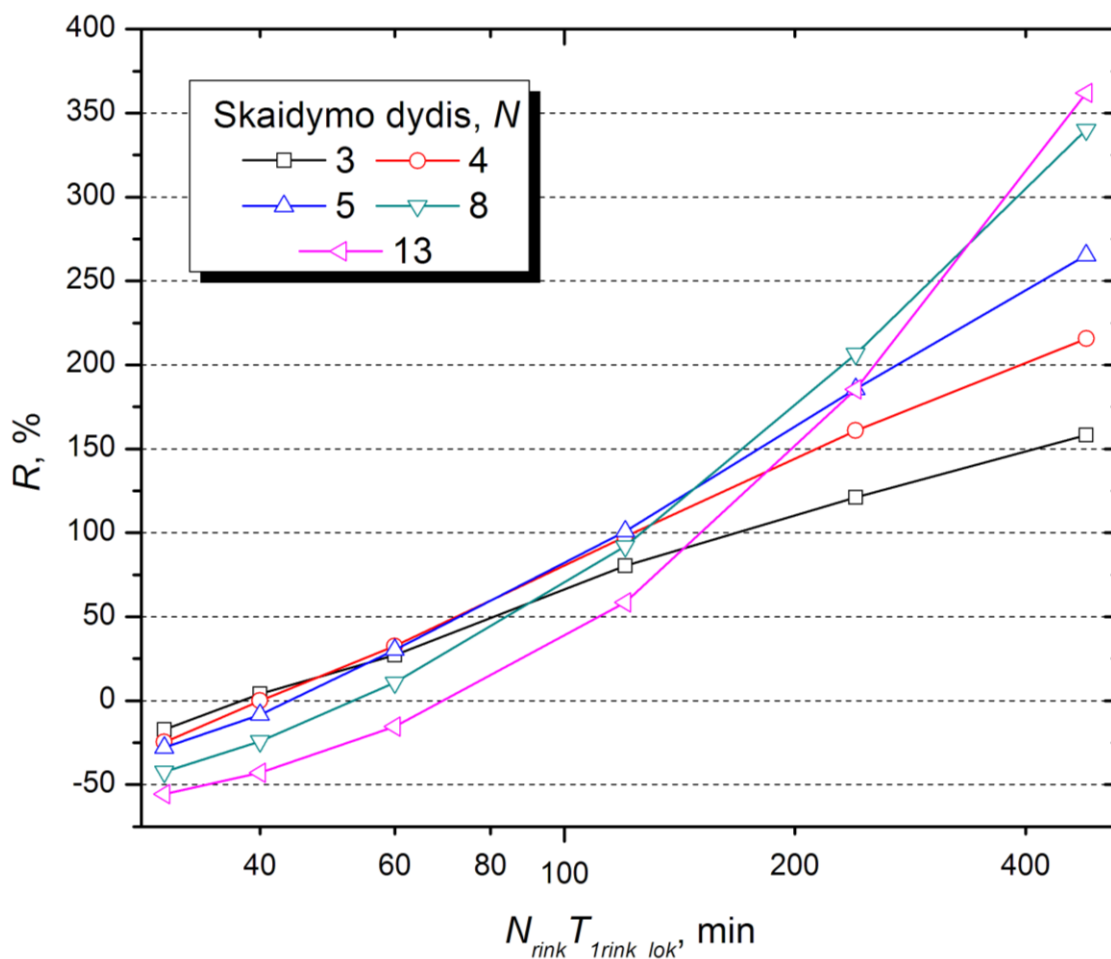
7 pav. Metodikos efektyvumo priklausomybė nuo skaidymo dydžio, perskaičiuota dvigubai lėtesniam kompiuteriui.

7 pav. lyginant su 6 pav. pastebima, kad nors grafikai vizualiai nepakito, tačiau matoma, kad visos užduotys yra sprendžiamos efektyviau. Pavyzdžiui 120 min. trukmės užduotis pirmuoju atveju (6 pav.) greičiausiai sprendžiama su $R = 100\%$ pagreitejimu skaidant užduotį į 5 dalis, tuo tarpu lyginant su dvigubai lėtesniu lokaliu kompiuteriu didžiausias pagreitejimas tai pačiai užduočiai yra $R = 160\%$, pasiekiamas skaidant užduotį į 4 dalis. Taip pat pastebima, kad užduotys efektyviai sprendžiamos platesniuose skaidymo reikšmių intervaluose, pavyzdžiui 60 min. trukmės lokali užduotis pirmuoju (6 pav.) atveju efektyviai sprendžiama ją skaidant į 2-9 dalis, kai antruoju atveju efektyviai sprendžiama ją suskaidant ir į 10 dalių.



8 pav. Metodikos efektyvumo priklausomybė nuo skaidymo dydžio, perskaičiuota dvigubai greitesniam kompiuteriui.

Skaičiavimų trukmės pokytį perskaičius dvigubai greitesniam kompiuteriui (8 pav.), pastebimas pakitęs minimalios efektyviai sprendžiamos užduoties laikas. Su neženkliai $R = 2\%$ pagreitejimu yra sprendžiama tik 60 min. trukmės užduotis, suskaidant ją į 5 dalis, o visos užduotys, trumpesnės nei 60 min. yra sprendžiamos neefektyviai. Skirtingai nei pirmais dviem atvejais, visoms trukmėms $N_{rink} T_{rink_lok}$ skaičiavimai atliekami neefektyviai, jei užduotis suskaidoma tik į dvi dalis. 120 min. užduotis, pradiniu (6 pav.), dvigubai lėtesnio (7 pav.) ir dvigubai greitesnio (8 pav.) kompiuterio atvejais yra geriausiai sprendžiama su tokiais atitinkamais pagreitejimais: 100%, 160% ir 60%. Atitinkamai $N_{rink} T_{rink_lok} = 240$ min. trukmės užduotis skaičiavimų tinkle sprendžiama su tokiais pagreitejimais R : 220%, 300% ir 140%, užduotis suskaidant atitinkamai į 7-9, 4-6 ir 9-13 dalių.



9 pav. Metodikos efektyvumo priklausomybė nuo lokaliai atliekamų skaičiavimų trukmės.

Iš 9 pav. pastebima, kad kuo didesnė užduoties apimtis, tuo didesnis yra optimalus skaidymo dydis N_u , su kuriuo metodika yra efektyviausia: 60 min. trukmės užduotis greičiausiai sprendžiama kai N_u yra 3, 4, 5, tuo tarpu 480 min. trukmės užduotis efektyviausiai sprendžiama kai $N_u = 13$. Metodikos taikymo efektyvumas tik neženkliai skiriasi užduotims, kurių trukmė yra tarp 1 ir 4 valandų, kai $N_u \in [3;8]$.

3.5.2. Efektyvaus skaičiavimų tinklų panaudojimo rekomendacijų santrauka

Praktiškai pritaikius metodiką skaičiavimų tinkle *BalticGrid* ir nustatius minimalios efektyviai sprendžiamos užduoties trukmę, rekomenduojama:

1. Norint pasinaudoti skaičiavimų tinklų resursais, pirmiausiai reikia įvertinti skaičiavimų trukmę T_{1rink_lok} su vienu parametru rinkiniu p_1, p_2, \dots, p_n lokaliame kompiuteryje. Rasti skaičiavimų, atliekamų su visais parametru rinkiniais (rinkinių skaičius žymimas N_{rink}), trukmę lokaliame kompiuteryje, $N_{rink}T_{1rink_lok}$. Skaičiavimų tinklais efektyviai pasinaudoti galima tik tada, kai apskaičiuotas dydis $N_{rink}T_{1rink_lok}$ yra ne mažesnis nei 40 minučių.

2. Priklausomai nuo visos užduoties apimties, atitinkamai reikėtų parinkti ir skaičių N_u , nurodantį, į kiek užduočių reikia skaidyti visą užduotį. Kuo didesnė visos užduoties apimtis, tuo didesnis yra optimalus N_u . Konkrečiu metodikos takymo atveju, kai metodika buvo taikyta skaičiavimų tinkle *BalticGrid* ir atliktas palyginimas su lokaliu kompiuteriu, pastebėta, kad kai $N_{rink}T_{Irink_lok} = 40$ min., tai optimalus N_u yra 3, tuo tarpu kai $N_{rink}T_{Irink_lok} = 480$ min., tai optimalus N_u yra tarp 9 ir 13.

3. Naudojantis *BalticGrid* resursais pastebėta, kad pasitaiko atveju, kai viena ar kelios užduotys ilgai būna nepriskirtos konkrečiam skaičiavimo mazgui. Todėl rekomenduotina tą pačią užduotį skaičiavimų tinklui atiduoti kelis kartus - tokiu atveju, dėka likusių užduočių rezultatus būtų galima gauti anksčiau. Rekomenduotina skaičiavimų tinklui paduoti bent tris tokias pačias užduotis, varijuojant skirtingais užduoties suskaidymo dydžiais N_u .

4. Metodikos taikymas tiriant biokatalizės procesą

Sukurta metodika taikyta tyrinėjant chemiškai modifikuoto biojutiklio matematinį modelį. Biojutiklis – tai matavimo įrenginys, skirtas tirpalo cheminėms savybėms tirti, dažniausiai konkrečios medžiagos koncentracijai nustatyti [CB80,CB90]. Biojutikliai plačiai taikomi labai tiksliais biologiniams junginiams aptikti ir jų koncentracijoms nustatyti [WLS97].

4.1. Chemiškai modifikuoto biojutiklio modeliavimas

Nagrinėjamas toks biojutiklio modelis, kurio biokatalizės procesuose dalyvauja fermentas (E), tiriamoji medžiaga – substratas (S) ir mediatorius (M) – medžiaga, kuria yra chemiškai modifikuojamas elektrodas. Fermentinė reakcija, kurios metu substratas virsta produktu, aprašoma taip:



čia E_{ox} yra oksiduotas fermentas, E_{red} – redukuotas fermentas, E_S – fermento-substrato junginys, P ir P_1 – reakcijos produktai.

4.1.1. Detalusis modelis

Nagrinėjamas vienasluoksnis biojutiklio modelis – tokiaime modelyje substrato, mediatoriaus ir reakcijos produkto koncentracijų kaita nagrinėjama tik fermente, kurio storis yra d . Biojutiklio veikimas matematiškai aprašomas reakcijos-difuzijos lygtimis [Ari75,Bri05], apibūdinančiomis biojutikliuose reakcijos metu vykstančius procesus:

$$\frac{\partial S}{\partial t} = k_{-1}E_S - k_1E_{ox}S + D_S \frac{\partial^2 S}{\partial x^2}, \quad (4.2a)$$

$$\frac{\partial M}{\partial t} = -k_3E_{red}M + D_M \frac{\partial^2 M}{\partial x^2}, \quad (4.2b)$$

$$\frac{\partial P}{\partial t} = k_3E_{red}M + D_P \frac{\partial^2 P}{\partial x^2}, \quad (4.2c)$$

$$\frac{\partial E_S}{\partial t} = k_1E_{ox}S - k_{-1}E_S - k_2E_S, \quad (4.2d)$$

$$\frac{\partial E_{red}}{\partial t} = k_2E_S - k_3E_{red}M, \quad (4.2e)$$

$$\frac{\partial E_{ox}}{\partial t} = k_3 E_{red} M + k_{-1} E_s - k_1 E_{ox} S, \quad 0 < x < d, t > 0. \quad (4.2f)$$

čia x – erdvės koordinatė, elektrodo paviršių atitinka $x = 0$, o išorinį fermento paviršių – $x = d$, t – laikas, $S(x,t)$, $M(x,t)$, $P(x,t)$ – substrato, mediatoriaus ir produkto koncentracijos, D_s , D_m ir D_p – substrato, mediatoriaus ir produkto difuzijos koeficientai.

Tarpinio reakcijos produkto P_I reakcijos-difuzijos lygtis yra nenagrinėjama, nes šis produktas neturi įtakos biojutiklio atsakui. Fermento-substrato junginio E_s , redukuoto fermento E_{red} ir oksiduoto fermento E_{ox} reakcijos aprašomos be difuzijos nario, kadangi difuzijos įtaka šiems procesams yra nedidelė.

Biojutiklis pradeda veikti, kai jis yra panardinamas į tiriamąjį tirpalą. Tokia biojutiklio būseną yra aprašoma pradinėmis sąlygomis (laiko momentu $t = 0$):

$$S(x,0) = \begin{cases} 0, & 0 \leq x < d, \\ S_0, & x = d, \end{cases} \quad (4.3)$$

$$P(x,0) = 0, \quad 0 \leq x \leq d, \quad (4.4)$$

$$M(x,0) = \begin{cases} M_0, & x = 0, \\ 0, & 0 < x \leq d, \end{cases} \quad (4.5)$$

$$E_s(x,0) = 0, \quad 0 \leq x \leq d, \quad (4.6)$$

$$E_{red}(x,0) = 0, \quad 0 \leq x \leq d, \quad (4.7)$$

$$E_{ox}(x,0) = E_0, \quad 0 \leq x \leq d. \quad (4.8)$$

Matematiniai reiškiniai (4.3)-(4.8) yra pagrindžiami fizikinėmis ir cheminėmis sąlygomis – pradinio laiko momentu ($t = 0$), substrato yra tik fermento paviršiuje ($x = d$), tačiau visame fermente substrato koncentracija yra lygi nuliui (4.3). Kadangi reakcija pradinio laiko momentu dar nevyksta, tai visame fermento sluoksnyje nėra susidariusio produkto (4.4), mediatoriaus yra tik ant elektrodo paviršiaus (4.5), o visas fermentas dar nesureagavęs (4.6)-(4.8).

Kraštinės sąlygos aprašo biojutiklio būseną jam pradėjus veikti (kai $t > 0$). Elektrodo paviršiuje ($x = 0$) dėl jo poliarizacijos visas reakcijos produktas yra išsekvojamas srovei generuoti, t.y. reakcijos produkto koncentracija elektrodo paviršiuje pastoviai mažinama iki nulio. Substrato koncentracija elektrodo paviršiuje yra aprašoma nepalaidumo sąlyga (4.9), o mediatoriaus, dengiančio elektrodą, koncentracija yra nuolatos palaikoma pastovia:

$$\left. \frac{\partial S}{\partial x} \right|_{x=0} = 0, \quad (4.9)$$

$$P(0,t) = 0, \quad (4.10)$$

$$M(0,t) = M_0. \quad (4.11)$$

Fermento išorėje ($x = d$) medžiagų koncentracijos išlieka pastovios:

$$S(d,t) = S_0, \quad (4.12)$$

$$P(d,t) = 0, \quad (4.13)$$

$$M(d,t) = 0. \quad (4.14)$$

Reakcijos produktas elektrodo paviršiuje sukelia anodinę srovę, kurios stiprumas yra matuojamas ir vadinamas biojutiklio atsaku. Amperometrinio biojutiklio srovės stiprumo tankis $I(t)$ yra tiesiogiai proporcingas produkto koncentracijos gradientui elektrodo paviršiuje [CB80, Sch90]:

$$I(t) = n_e F D_P \left. \frac{\partial P}{\partial x} \right|_{x=0}, \quad (4.15)$$

čia n_e yra elektronų, kurie perneša krūvį elektrodo paviršiuje, skaičius, F – Faradėjaus konstanta ($F = 96485$ C/mol). Pradžioje biojutiklio srovės stiprumas sparčiai didėja, o vėliau pasiekia savo pastovią būseną:

$$I_\infty(t) = \lim_{t \rightarrow \infty} I(t). \quad (4.16)$$

4.1.2. Nedimensinis detalusis modelis

Palyginant detalųjį ir kvazistacionarųjį modelius buvo nagrinėjama parametrų įtaka biojutiklio atsakui. Tiek detalusis, tiek kvazistacionarusis chemiškai modifikuoto biojutiklio modelis turi didelį rinkinį parametrų, kurie apsunkina palyginimo uždavinį. Todėl tikslinga aprašyti nedimensinį biojutiklio veikimo modelį, kuris palengvintų visapusiško modelių palyginimo uždavinio sprendimą.

Aprašant nedimensinį chemiškai modifikuoto biojutiklio veikimo modelį, buvo įvesti šie nedimensiniai parametrai:

$$\bar{x} = \frac{x}{d}, \quad \bar{t} = \frac{t D_S}{d^2}, \quad \bar{S} = \frac{k_1}{k_{-1} + k_2} S, \quad \bar{S}_0 = \frac{k_1}{k_{-1} + k_2} S_0, \quad (4.17)$$

$$\bar{M} = \frac{k_3}{k_2} M, \quad \bar{M}_0 = \frac{k_3}{k_2} M_0, \quad \bar{P} = \frac{k_3}{k_2} P, \quad (4.18)$$

$$\bar{E}_S = \frac{E_S}{E_0}, \quad \bar{E}_{red} = \frac{E_{red}}{E_0}, \quad \bar{E}_{ox} = \frac{E_{ox}}{E_0}. \quad (4.19)$$

Biojutiklyje vykstančius procesus aprašančios (4.2a)-(4.2f) lygtys tada keičiamos tokiomis:

$$\frac{\partial \bar{S}}{\partial \bar{t}} = Q \frac{k_{-1}k_1}{k_{-1} + k_2} \bar{E}_S - Qk_1 \bar{E}_{ox} \bar{S} + \frac{\partial^2 \bar{S}}{\partial \bar{x}^2}, \quad (4.20a)$$

$$\frac{\partial \bar{M}}{\partial \bar{t}} = -Qk_3 \bar{E}_{red} \bar{M} + \frac{D_M}{D_S} \frac{\partial \bar{M}}{\partial \bar{x}}, \quad (4.20b)$$

$$\frac{\partial \bar{P}}{\partial \bar{t}} = Qk_3 \bar{E}_{red} \bar{M} + \frac{D_P}{D_S} \frac{\partial \bar{P}}{\partial \bar{x}}, \quad (4.20c)$$

$$\frac{\partial \bar{E}_S}{\partial \bar{t}} = \frac{Q}{E_0} (k_{-1} \bar{E}_{ox} \bar{S} + k_2 \bar{E}_{ox} \bar{S} - k_{-1} \bar{E}_S - k_2 \bar{E}_S), \quad (4.20d)$$

$$\frac{\partial E_{red}}{\partial t} = Q \frac{k_2}{E_0} (\bar{E}_S - \bar{E}_{red} \bar{M}), \quad (4.20e)$$

$$\frac{\partial \bar{E}_{ox}}{\partial \bar{t}^*} = \frac{Q}{E_0} (k_2 \bar{E}_{red} \bar{M} + k_{-1} \bar{E}_S - k_{-1} \bar{E}_{ox} \bar{S} - k_2 \bar{E}_{ox} \bar{S}), \quad 0 < \bar{x} < 1, \bar{t} > 0, \quad (4.20f)$$

čia $Q = \frac{E_0 d^2}{D_S}$ yra difuzijos modulio $\sigma^2 = kQ$ dalis, apibūdinanti difuzijos greitį (D_S/d^2) ir pradinę

fermento koncentraciją E_0 .

Pradinės sąlygos (4.3)-(4.8) keičiamos tokiomis:

$$\bar{S}(\bar{x}, 0) = \begin{cases} 0, & 0 \leq \bar{x} < 1, \\ \bar{S}_0, & \bar{x} = 1, \end{cases} \quad (4.21)$$

$$\bar{P}(\bar{x}, 0) = 0, \quad 0 \leq \bar{x} \leq 1, \quad (4.22)$$

$$\bar{M}(\bar{x}, 0) = \begin{cases} \bar{M}_0, & \bar{x} = 0, \\ 0, & 0 < \bar{x} \leq 1, \end{cases} \quad (4.23)$$

$$\bar{E}_S(\bar{x}, 0) = 0, \quad 0 \leq \bar{x} \leq 1, \quad (4.24)$$

$$\bar{E}_{red}(\bar{x}, 0) = 0, \quad 0 \leq \bar{x} \leq 1, \quad (4.25)$$

$$\bar{E}_{ox}(\bar{x}, 0) = E_0, \quad 0 \leq \bar{x} \leq 1. \quad (4.26)$$

Kraštinės sąlygos (4.9)-(4.14) keičiamos tokiomis:

$$\left. \frac{\partial \bar{S}}{\partial \bar{x}} \right|_{\bar{x}=0} = 0, \quad (4.27)$$

$$\bar{P}(0, \bar{t}) = 0, \quad (4.28)$$

$$\bar{M}(0, \bar{t}) = \bar{M}_0, \quad (4.29)$$

$$\bar{S}(1, \bar{t}) = \bar{S}_0, \quad (4.30)$$

$$\bar{P}(1, \bar{t}) = 0, \quad (4.31)$$

$$\bar{M}(1, \bar{t}) = 0. \quad (4.32)$$

Nedimensinė biojutiklyje tekanti srovė aprašoma lygtimi:

$$\bar{I}(\bar{t}) = \left. \frac{\partial \bar{P}}{\partial \bar{x}} \right|_{\bar{x}=0} = I(t) \frac{k_3 d}{k_2 n_e F D_P}. \quad (4.33)$$

Pastovioji srovė aprašoma taip:

$$\bar{I}_\infty(\bar{t}) = \lim_{\bar{t} \rightarrow \infty} \bar{I}(\bar{t}). \quad (4.34)$$

4.1.3. Kvazistacionarusis modelis

Modelį, aprašomą lygčių sistema (4.2) yra prasminga suprastinti, kai galima laikytis kvazistacionarumo prielaidos. Tokiu atveju tarpinio komplekso E_S koncentracija nesikeičia [TKW87, SS92], ir biojutiklio veikimas gali būti modeliuojamas atsižvelgiant į šią sąlyga [BK08]. Tada lygčių sistemą (4.2) galima suprastinti taip:

$$\frac{\partial S}{\partial t} = D_S \frac{\partial^2 S}{\partial x^2} - v(M, S), \quad (4.35a)$$

$$\frac{\partial M}{\partial t} = D_M \frac{\partial^2 M}{\partial x^2} - v(M, S), \quad (4.35b)$$

$$\frac{\partial P}{\partial t} = D_P \frac{\partial^2 P}{\partial x^2} + v(M, S), \quad 0 < x < d, t > 0. \quad (4.35c)$$

Kvazistacionariojo modelio atveju laikoma, kad fermentų reakcijų greičiai E_S , E_{red} ir E_{ox} yra žymiai mažesni už substrato, produkto ir mediatoriaus reakcijų greičius, todėl galima parašyti, kad:

$$\frac{\partial E_S}{\partial t} = \frac{\partial E_{red}}{\partial t} = \frac{\partial E_{ox}}{\partial t} = 0. \quad (4.36)$$

Bendras fermento kiekis yra lygus E_0 ir yra aprašomas kaip tokia suma:

$$E_0 = E_{ox} + E_{red} + E_S, \quad (4.37)$$

iš čia išreiškiame E_S :

$$E_S = E_0 - E_{ox} - E_{red}, \quad (4.38)$$

įrašome (4.38) į (4.2e) ir išreiškiame E_{ox} :

$$E_{ox} = \frac{k_2 E_0 - k_2 E_{red} - k_3 E_{red} M}{k_2}, \quad (4.39)$$

ir (4.38) įrašome į (4.2f) ir išreiškiame E_{ox} :

$$E_{ox} = \frac{k_3 E_{red} M + k_{-1} E_0 - k_{-1} E_{red}}{k_{-1} + k_1 S}, \quad (4.40)$$

sulyginus (4.39) ir (4.40) lygybių dešiniąsias puses išreiškiame E_{red} :

$$E_{red} = \frac{k_1 k_2 S E_0}{k_1 k_3 S M + (k_{-1} k_3 + k_2 k_3) M + k_1 k_2 S}, \quad (4.41)$$

sulyginus (4.2c) ir (4.35c) dešiniąsias puses ir įrašius E_{red} reiškinį (4.41), gauname:

$$v(M, S) = \frac{E_0 k_1 k_2 k_3 S M}{k_1 k_3 S M + (k_{-1} k_3 + k_2 k_3) M + k_1 k_2 S}. \quad (4.42)$$

Pradinės sąlygos (kai $t = 0$) aprašomos lygtimis (4.3)-(4.5).

Kraštinės sąlygos ($t > 0$) elektrodo paviršiuje ($x = 0$) aprašomos lygtimis (4.9)-(4.11), o fermento paviršiuje ($x = d$) - (4.12)-(4.14) lygtimis.

Biojutiklyje tekančios srovės priklausomybė nuo laiko aprašoma (4.15) lygtimi, o pastovioji srovė – (4.16) lygtimi.

4.1.4. Nedimensinis kvazistacionarusis modelis

Aprašant nedimensinį chemiškai modifikuoto biojutiklio veikimo kvazistacionarųjį modelį buvo naudoti detaliojo nedimensinio modelio nedimensiniai parametrai (4.17), (4.18).

Biojutiklyje vykstančius procesus aprašanti lygčių sistema (4.35) tada keičiama tokia:

$$\frac{\partial \bar{S}}{\partial \bar{t}} = \frac{\partial^2 \bar{S}}{\partial \bar{x}^2} - \sigma_{red}^2 \frac{\bar{M} \cdot \bar{S}}{\bar{M} \cdot \bar{S} + \bar{M} + \bar{S}}, \quad (4.43a)$$

$$\frac{\partial \bar{M}}{\partial \bar{t}} = \frac{D_M}{D_S} \frac{\partial^2 \bar{M}}{\partial \bar{x}^2} - \sigma_{ox}^2 \frac{\bar{M} \cdot \bar{S}}{\bar{M} \cdot \bar{S} + \bar{M} + \bar{S}}, \quad (4.43b)$$

$$\frac{\partial \bar{P}}{\partial \bar{t}} = \frac{D_P}{D_S} \frac{\partial^2 \bar{P}}{\partial \bar{x}^2} + \sigma_{ox}^2 \frac{\bar{M} \cdot \bar{S}}{\bar{M} \cdot \bar{S} + \bar{M} + \bar{S}}, \quad 0 < \bar{x} < 1, \bar{t} > 0, \quad (4.43c)$$

čia $\sigma_{red}^2 = k_{red} Q = \frac{k_1 k_2}{k_{-1} + k_2} \frac{E_0 d^2}{D_S}$ yra nedimensinis difuzijos modulis, kuris yra fermentinės reakcijos

ir difuzijos santykis, analogiškai $\sigma_{ox}^2 = k_3 Q = k_3 \frac{E_0 d^2}{D_S}$.

Pradinės sąlygos keičiamos sąlygomis, kurios aprašomos lygtimis (4.21)-(4.23), o kraštinės – lygtimis (4.27)-(4.32).

Biojutiklyje tekančios srovės priklausomybė nuo laiko aprašoma (4.33) lygtimi, o pastovioji srovė – (4.34) lygtimi.

4.2. Matematinų modelių sprendimas kompiuteriu

4.2.1. Detaliojo modelio atveju

Aprašyto matematinio modelio analizinis sprendinys egzistuoja tik prie tam tikrų reakcijų greičių, todėl bendrai taikomi skaitiniai uždavinio sprendimo metodai. Uždaviniui, aprašomam lygčių sistema (4.2) spręsti, taikomas baigtinių skirtumų metodas [KS74, Sam01].

Argumentų x ir t tolydaus kitimo sritis yra keičiama diskrečia taškų aibe [Sam01]:

$$x_i = ih, \quad i = 0, 1, \dots, N, \quad Nh = d, \quad (4.44)$$

$$t_j = j\tau, \quad j = 0, 1, \dots, M, \quad M\tau = T. \quad (4.45)$$

Pažymėkime, kad $S_i^j = S(x_i, t_j)$, $P_i^j = P(x_i, t_j)$, $M_i^j = M(x_i, t_j)$, $E_{S,i}^j = E_S(x_i, t_j)$, $E_{red,i}^j = E_{red}(x_i, t_j)$ ir $E_{ox,i}^j = E_{ox}(x_i, t_j)$.

Sistemos (4.2) diferencialines lygtys dalinėmis išvestinėmis keičiamos skirtuminėmis lygtimis:

$$\frac{S_i^{j+1} - S_i^j}{\tau} = k_{-1}E_{S,i}^j - k_{-1}E_{ox,i}^j S_i^j + D_S \frac{S_{i-1}^j - 2S_i^j + S_{i+1}^j}{h^2}, \quad (4.46a)$$

$$\frac{M_i^{j+1} - M_i^j}{\tau} = -k_3 E_{red,i}^j M_i^j + D_M \frac{M_{i-1}^j - 2M_i^j + M_{i+1}^j}{h^2}, \quad (4.46b)$$

$$\frac{P_i^{j+1} - P_i^j}{\tau} = k_3 E_{red,i}^j M_i^j + D_P \frac{P_{i-1}^j - 2P_i^j + P_{i+1}^j}{h^2}, \quad (4.46c)$$

$$\frac{E_{S,i}^{j+1} - E_{S,i}^j}{\tau} = k_1 E_{ox,i}^j S_i^j - k_{-1} E_{S,i}^j - k_2 E_{S,i}^j, \quad (4.46d)$$

$$\frac{E_{red,i}^{j+1} - E_{red,i}^j}{\tau} = k_2 E_{S,i}^j - k_3 E_{red,i}^j M_i^j, \quad (4.46e)$$

$$\frac{E_{ox,i}^{j+1} - E_{ox,i}^j}{\tau} = k_3 E_{red,i}^j M_i^j + k_{-1} E_{S,i}^j - k_1 E_{ox,i}^j S_i^j, \quad 0 < i < N, j \geq 0. \quad (4.46f)$$

Atliekant skaičiavimus, nagrinėjamoje lygčių sistemoje (4.2), kad nebūtų pažeistos stabilumo sąlygos, diskrečių taškų aibės žingsnius h ir τ turi sieti ryšys [Sam01]:

$$\tau < \frac{h^2}{2D_S}. \quad (4.47)$$

Skaičiavimams atlikti taikoma tokia žingsnio τ reikšmė:

$$\tau = \frac{h^2}{4D_s}. \quad (4.48)$$

Pradinės sąlygos (4.3)-(4.8) keičiamos tokiomis lygtimis:

$$S_i^0 = \begin{cases} 0, & 0 \leq i < N, \\ S_0, & i = N, \end{cases} \quad (4.49)$$

$$P_i^0 = 0, \quad 0 \leq i \leq N, \quad (4.50)$$

$$M_i^0 = \begin{cases} M_0, & i = 0, \\ 0, & 0 < i \leq N, \end{cases} \quad (4.51)$$

$$E_{S,i}^0 = 0, \quad 0 \leq i \leq N, \quad (4.52)$$

$$E_{red,i}^0 = 0, \quad 0 \leq i \leq N, \quad (4.53)$$

$$E_{ox,i}^0 = E_0, \quad 0 \leq i \leq N. \quad (4.54)$$

Kraštinės sąlygos (4.9)-(4.11), aprašančios medžiagų koncentracijas elektrodo paviršiuje ($x = 0$) keičiamos tokiomis lygtimis:

$$S_0^j = S_1^j, \quad (4.55)$$

$$P_0^j = 0, \quad (4.56)$$

$$M_0^j = M_0. \quad (4.57)$$

Fermento išorėje ($x = d$) medžiagų koncentracijos aprašančios lygtys (4.12)-(4.14) keičiamos tokiomis:

$$S_N^j = S_0, \quad (4.58)$$

$$P_N^j = 0, \quad (4.59)$$

$$M_N^j = 0. \quad (4.60)$$

Biojutiklio srovės tankį I_j ($I_j = I(P_0^j, P_1^j)$) momentu t_j , $j = 1, 2, \dots, M$, galime apskaičiuoti taikydami reiškinių (4.15) skaitinę aproksimaciją:

$$I_j \approx I(t_j) = n_e F D_p \frac{(P_1^j - P_0^j)}{h}. \quad (4.61)$$

Srovės reikšmės skaičiuojamos tol, kol dviejų gretimų reikšmių santykinė paklaida nesiskiria daugiau nei pasirinktos paklaidos vertė ε :

$$T_A \approx \min_{j>1, I_j>0} \left\{ t_j : \frac{1}{I_j} \cdot \frac{I_j - I_{j-1}}{\tau} \cdot t_j < \varepsilon \right\}, \quad I_\infty \approx I(T_A),$$

į šį reiškinių įrašius $t_j = \tau \cdot j$ ir suprastinus, gauname:

$$T_A \approx \min_{j>1, I_j>0} \left\{ t_j : \frac{I_j - I_{j-1}}{I_j} \cdot j < \varepsilon \right\}, \quad I_\infty \approx I(T_A). \quad (4.62)$$

4.2.2. Kvazistacionariojo modelio atveju

Diferencialines lygtys (50)-(52) dalinėmis išvestinėmis keičiamos skirtuminėmis lygtimis:

$$\frac{S_i^{j+1} - S_i^j}{\tau} = D_S \frac{S_{i-1}^j - 2S_i^j + S_{i+1}^j}{h^2} - \frac{E_0 k_1 k_2 k_3 S_i^j M_i^j}{k_1 k_3 S_i^j M_i^j + (k_{-1} k_3 + k_2 k_3) M_i^j + k_1 k_2 S_i^j}, \quad (4.63a)$$

$$\frac{M_i^{j+1} - M_i^j}{\tau} = D_M \frac{M_{i-1}^j - 2M_i^j + M_{i+1}^j}{h^2} - \frac{E_0 k_1 k_2 k_3 S_i^j M_i^j}{k_1 k_3 S_i^j M_i^j + (k_{-1} k_3 + k_2 k_3) M_i^j + k_1 k_2 S_i^j}, \quad (4.63a)$$

$$\frac{P_i^{j+1} - P_i^j}{\tau} = D_P \frac{P_{i-1}^j - 2P_i^j + P_{i+1}^j}{h^2} + \frac{E_0 k_1 k_2 k_3 S_i^j M_i^j}{k_1 k_3 S_i^j M_i^j + (k_{-1} k_3 + k_2 k_3) M_i^j + k_1 k_2 S_i^j}, \quad 0 < i < N, j \geq 0. \quad (4.63a)$$

Skaičiavimai atliekami naudojantis (4.48) laiko žingsnio reikšme, pradinės sąlygos aprašomos lygtimis (4.49)-(4.51), o kraštinės sąlygos aprašomos lygtimis (4.55)-(4.60).

Biojutiklio srovės tankis apskaičiuojamas naudojant (4.61) skaitinę aproksimaciją, srovės reikšmių skaičiavimus atliekant remiantis (4.62) sąlyga.

4.3. Kompiuterinių modelių patikimumas

Kvazistacionariojo modelio patikimumas buvo įvertintas palyginant skaitines biojutiklio srovės tankio reikšmes su žinomomis analitinėmis reikšmėmis. Žinoma, kad kai $S_0 \ll (k_{-1} + k_2) / k_1$, tada analitinė kvazistacionarioji srovės reikšmė yra tokia [BIK03a]:

$$I_1 = \frac{n_e F D_S S_0}{d} \left(1 - \frac{1}{\cosh(\sigma_{red})} \right) \quad (4.64)$$

čia σ_{red} yra nedimensinis difuzijos modulis, $\sigma_{red}^2 = \frac{k_1 k_2}{k_{-1} + k_2} \frac{E_0 d^2}{D_S}$.

Skaičiavimai atlikti keičiant pradines mediatoriaus ir substrato koncentracijas, tuo tarpo šių parametrų reikšmės buvo pastovios:

$$\begin{aligned} k_{-1} &= 10^4 \text{ s}^{-1}, k_1 = 1.1 \cdot 10^5 \text{ M}^{-1} \text{ s}^{-1}, k_2 = 10^3 \text{ s}^{-1}, \\ k_3 &= 10^7 \text{ M}^{-1} \text{ s}^{-1}, D_M = D_P = D_S = 3 \cdot 10^{-10} \frac{\text{m}^2}{\text{s}}, n_e = 1. \end{aligned} \quad (4.65)$$

Palyginus skaitiniu būdu rastą srovės tankio vertę (4.61) su analizinės išraiškos verte koncentracijoms $M_0 = 100\mu\text{M}$, $S_0 = 10\mu\text{M}$, $E_0 = 3\mu\text{M}$, gauta santykinė skaitinio modelio paklaida buvo $\sim 0,8\%$.

Tuo atveju, kai mediatoriaus koncentracija yra santykinai maža, t.y. $M_0 \ll k_2/k_3$, analizinė kvazistacionarioji srovė apskaičiuojama taip [BK08]:

$$I_2 = \frac{n_e F D_P m_0}{d} (\sigma_{ox} \coth(\sigma_{ox}) - 1), \quad (4.66)$$

čia $\sigma_{ox}^2 = k_3 \frac{E_0 d^2}{D_S}$.

Atlikus skaičiavimus su koncentracijomis $M_0 = 1\mu\text{M}$, $S_0 = 1\text{M}$, $E_0 = 3\mu\text{M}$, gauta skaitinio sprendinio santykinė paklaida yra $\sim 1,5\%$.

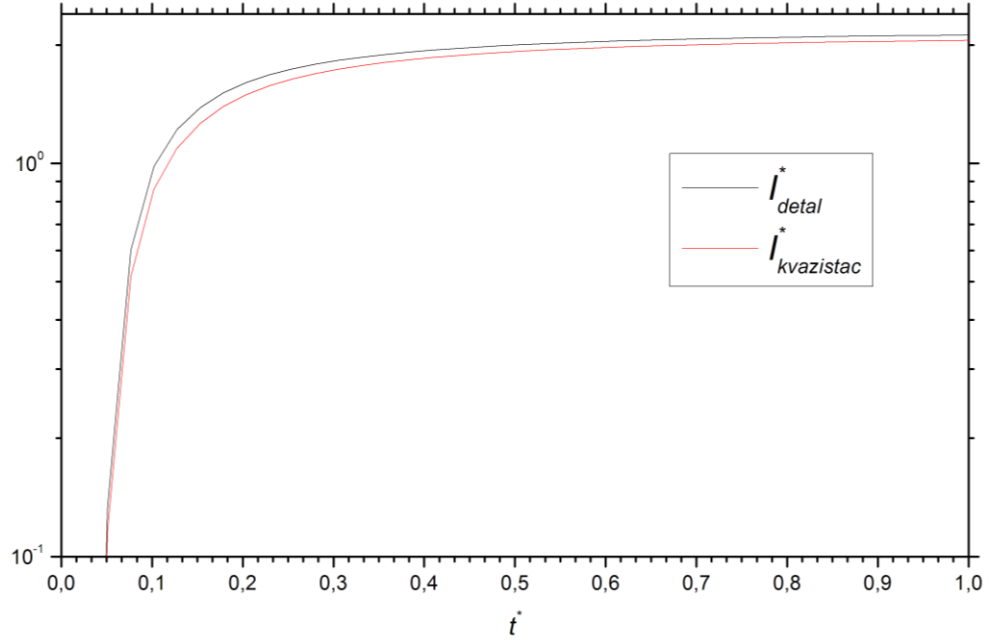
Detaliojo modelio patikimumas pagrįstas tuo, jog atliekant modeliavimo procesą buvo įsitikinta, jog bet kuriame erdvės taške, ir bet kuriuo laiko momentu oksiduoto fermento, redukuoto fermento ir fermento-substrato junginio koncentracijų suma yra lygi pradinei fermento koncentracijai:

$$E_{red,i}^j + E_{ox,i}^j + E_{S,i}^j = E_0, \quad (4.67)$$

čia $i = 0, 1, \dots, N$ (4.44) ir $j = 0, 1, \dots, M$ (4.45).

4.4. Detaliojo ir kvazistacionariojo biojutiklių modelių palyginimas

Metodika buvo pritaikyta atliekant kvazistacionariojo ir detaliojo biojutiklių modelių palyginimus. Buvo lyginamos nedimensinės srovės $\bar{I}_{detal}(\bar{t})$ ir $\bar{I}_{kvazistac}(\bar{t})$. Čia $\bar{I}_{detal}(\bar{t})$ - detaliojo modelio biojutiklio atsako srovė, o $\bar{I}_{kvazistac}(\bar{t})$ - kvazistacionariojo modelio srovė (10 pav.).



10 pav. Biojutiklio atsakas detaliojo ir kvazistacionariojo modelio atvejais. Skaičiavimams naudotos reikšmės:
 $k_{-1} = 10^4 \text{ s}^{-1}$, $k_1 = 1.1 \cdot 10^5 \text{ M}^{-1} \text{ s}^{-1}$, $k_2 = 10^3 \text{ s}^{-1}$, $k_3 = 10^7 \text{ M}^{-1} \text{ s}^{-1}$, $d = 10^{-4} \text{ m}$, $D_M = D_P = D_S = 3 \cdot 10^{-10} \text{ m}^2 / \text{ s}$,
 $M_0 = 10^{-5} \text{ M}$, $S_0 = 10^{-2} \text{ M}$, $E_0 = 3 \cdot 10^{-6} \text{ M}$, $n_e = 1$.

Lyginant aktualu apskaičiuoti kvazistacionariojo modelio biojutiklio tekančios srovės santykinę paklaidą. Analitiškai, bendruoju atveju, ši paklaida aprašoma taip:

$$\varepsilon = \left| \frac{\int_0^{\infty} \bar{I}_{detal}(\bar{t}) d\bar{t} - \int_0^{\infty} \bar{I}_{kvazistac}(\bar{t}) d\bar{t}}{\int_0^{\infty} \bar{I}_{detal}(\bar{t}) d\bar{t}} \right| \cdot 100\% \quad (4.68)$$

Modeliuojant detaliojo modelio biojutiklio veikimą skaitiniais metodais, srovės skaitinės vertės yra skaičiuojamos iki tokio laiko T_A , kol srovė tampa pusiasvyraja (4.62). Norint atlikti tikslų srovių palyginimą, srovių vertės buvo skaičiuojamos iki to paties laiko T_A , kai ir detaliojo ir kvazistacionariojo modelio srovės pasiekė igijo pusiausvyrosios srovės vertę. Lygtis (4.68) gali būti perrašyta taip:

$$\varepsilon = \left| \frac{\int_0^{T_A} \bar{I}_{detal}(\bar{t}) d\bar{t} - \int_0^{T_A} \bar{I}_{kvazistac}(\bar{t}) d\bar{t}}{\int_0^{T_A} \bar{I}_{detal}(\bar{t}) d\bar{t}} \right| \cdot 100\% \quad (4.69)$$

Šios lygties skaitinė aproksimacija, remiantis (4.33) ir yra tokia:

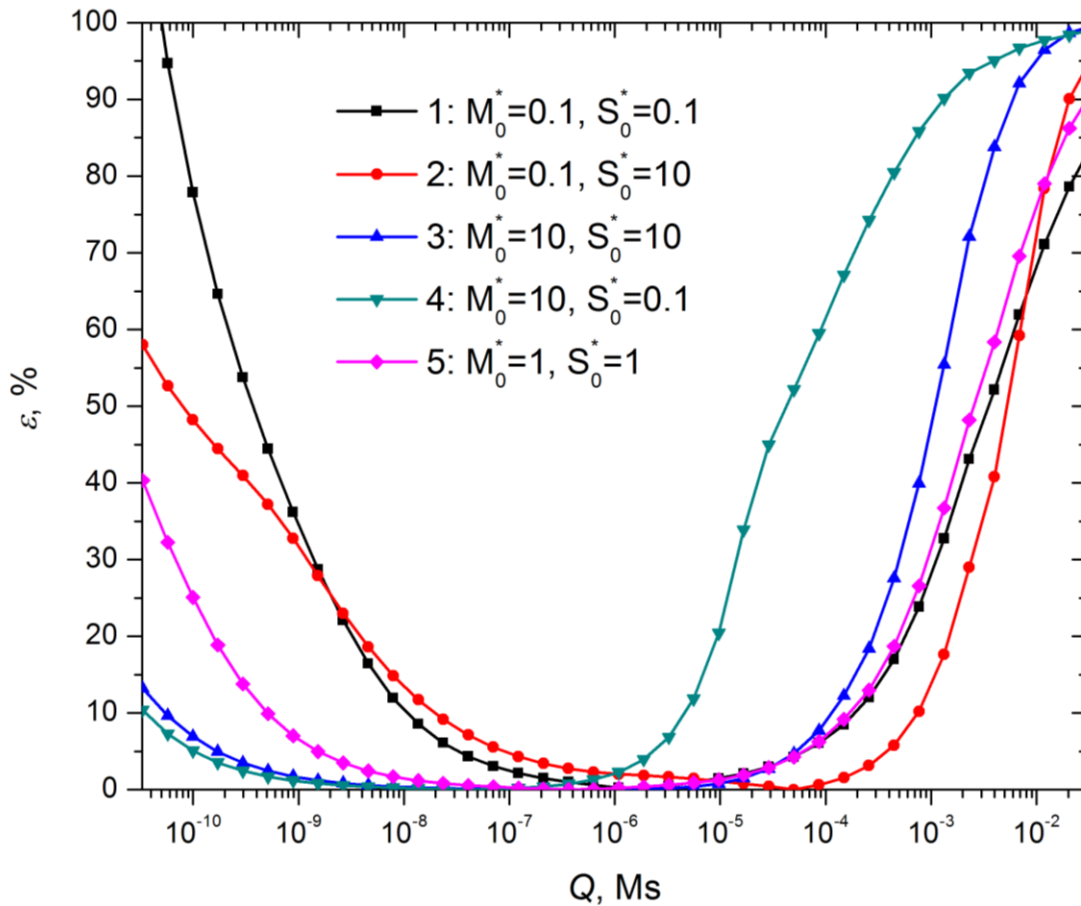
$$\varepsilon = \left| \frac{\sum_{j=1}^{\tau} I_{detal,j} \frac{k_3 d}{k_2 n_e FD_P} - \sum_{j=1}^{\tau} I_{kvazistac,j} \frac{k_3 d}{k_2 n_e FD_P}}{\sum_{j=1}^{\tau} I_{detal,j} \frac{k_3 d}{k_2 n_e FD_P}} \right| \cdot 100\%. \quad (4.70)$$

Į šį reiškinį įrašius (4.61) reiškinį, ir suprastinus gauname, kad kvazistacionariojo modelio atveju biojutiklio tekančios srovės paklaida yra apskaičiuojama taip:

$$\varepsilon = \left| \frac{\sum_{j=1}^{\tau} p_{detal,1}^j - p_{detal,0}^j - p_{kvazistac,1}^j + p_{kvazistac,0}^j}{\sum_{j=1}^{\tau} p_{detal,1}^j - p_{detal,0}^j} \right| \cdot 100\%. \quad (4.71)$$

Detaliojo ir kvazistacionariojo modelių palyginimas buvo atliktas $Q = \frac{E_0 d^2}{D_S}$ atžvilgiu.

Prenkant reikšmes Q buvo keičiami du parametrai: biojutiklio fermento storis d ir fermento pradinė koncentracija E_0 (1 priedas). Tyrimas atliktas su skirtingomis pradinėmis nedimensinės mediatoriaus koncentracijos M_0^* ir nedimensinės substrato koncentracijos S_0^* reikšmėms, nagrinėjant keturis atvejus: $M_0^* = 0.1$ ir $S_0^* = 0.1$, $M_0^* = 0.1$ ir $S_0^* = 10$, $M_0^* = 10$ ir $S_0^* = 10$, $M_0^* = 10$ ir $S_0^* = 0.1$. Skaičiuojant paklaidos vertes, kitos parametrų reikšmės buvo pastovios (4.65). Skaičiavimo rezultatai pateikiami 11 pav.

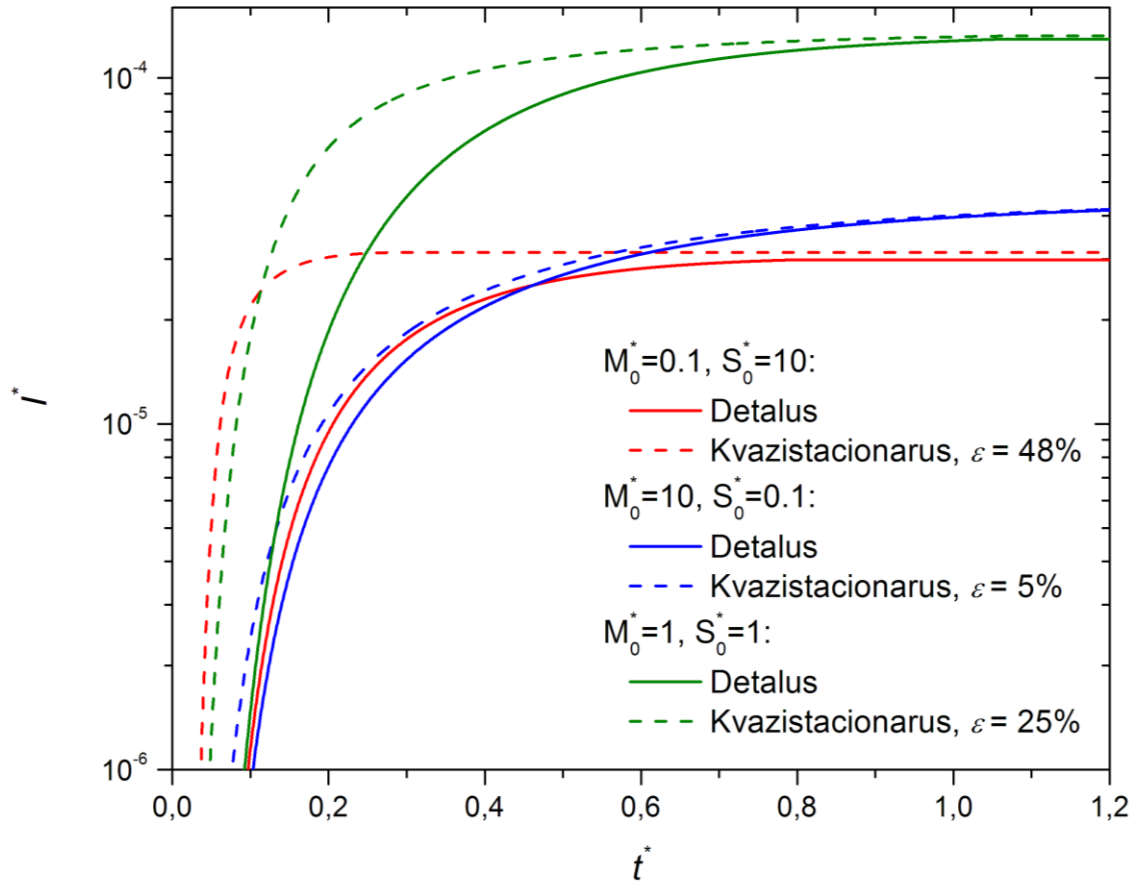


11 pav. Kvizistacionariojo modelio skaičiavimo paklaidos priklausomybė nuo Q .

Iš rezultatų matyti, kad kvazistacionariojo modelio skaičiavimo paklaidos ε vertė (4.6) priklauso nuo Q : prie itin didelių Q reikšmių ($Q > 10^{-2}$ Ms), paklaida visuomet didesnė nei 70%, nepriklausomi nuo to, kokios yra mediatoriaus ar substrato koncentracijos M_0^* ir S_0^* . Išskirtinis atvejis, kai šios koncentracijos yra $M_0^* = 10$ ir $S_0^* = 0.1$ - tada pastebima, kad kvazistacionariojo modelio skaičiavimų paklaida pasidaro didesnė už 20% prie $Q = 10^{-5}$ Ms (ir didesnių), kai visais kitais atvejais (1, 2, 3 ir 5) skaičiavimų paklaida šioms koncentracijoms yra mažesnė nei 5%. 4 atvejo skaičiavimų paklaida yra nulemiama skirtingų pusiausvyrųjų srovių verčių detaliojo ir kvazistacionariojo modelio atvejais – tai pastebima 13 pav.

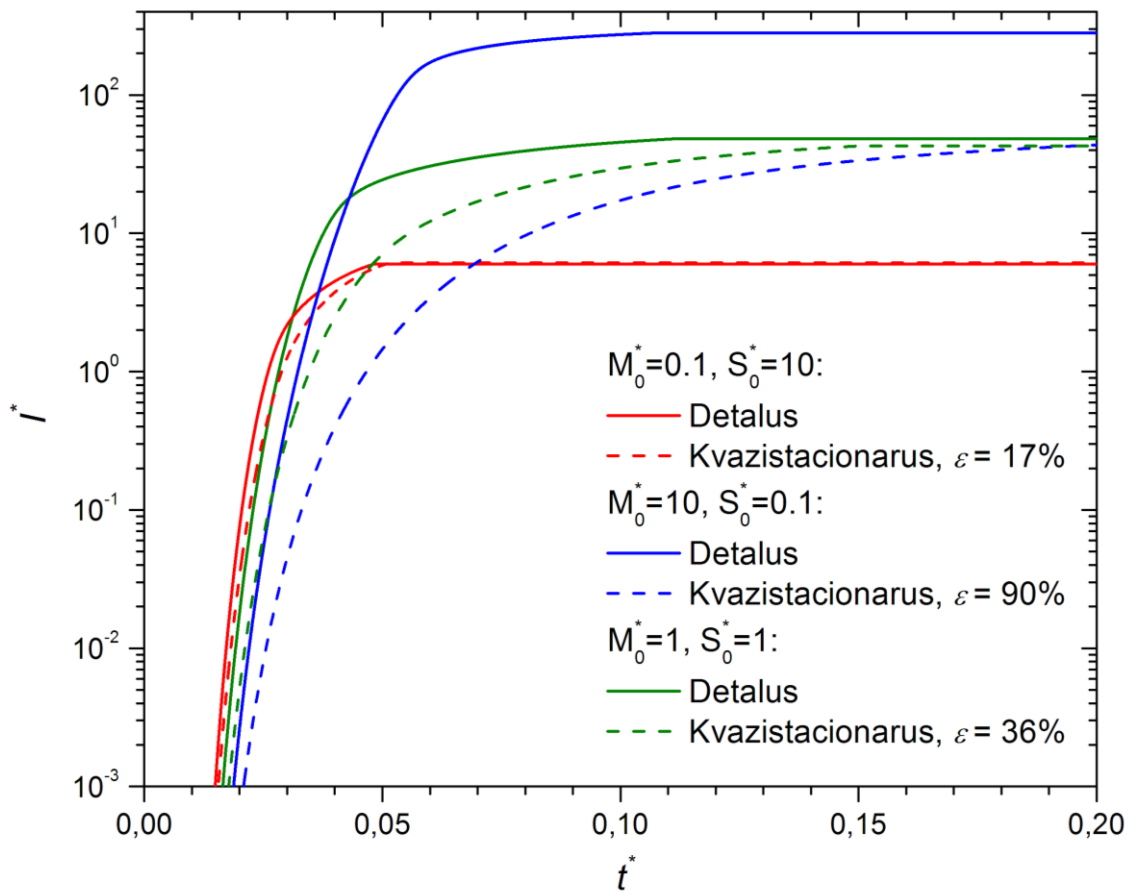
Kai Q reikšmės yra mažos ($Q < 10^{-6}$ Ms), skaičiavimų paklaida (priešingai nei didelių Q atveju) pastebimai priklauso nuo mediatoriaus ir substrato koncentracijų visais (1-5) atvejais: pavyzdžiui, kai $M_0^* \gg S_0^*$ (4 atvejis) kvazistacionarųjį modelį, su 10% paklaida galima naudoti plačiame ($10^{-11} - 10^{-6}$ Ms) Q intervale, tuo tarpu kitais atvejais paklaida šiame Q intervale yra didesnė. Prie mažų Q paklaidos ε vertės yra itin priklausomos vien tik nuo mediatoriaus koncentracijos: kai $M_0^* = 10$ (3 ir 4 atvejai), tai paklaidas su 5% paklaida galima skaičiuoti beveik

visame ($10^{-10} - 10^{-6}$ Ms) mažų Q reikšmių intervale. Tuo tarpu kai $M_0^* \leq 1$ (1, 2 ir 5 atvejai), taikyti kvazistacionarųjį modelį su 5% tikslumu galima žymiai siauresniuose intervaluose.



12 pav. Nedimensinės srovės priklausomybė nuo nedimensinio laiko, kai $Q=10^{-10}$ Ms. Modeliavimui naudoti kiti parametrai tokie kaip (4.7).

Iš 12 pav. matyti, kad kai $Q=10^{-10}$ Ms, visais atvejais kvazistacionariojo ir detaliojo modelio pusiausvyrosios srovės nusistovi apytiksliai tokios pačios. Kvazistacionariojo modelio skaičiavimų paklaidos vertę ε šiuo atveju nulemia skirtingi srovės kitimų greičiai. Taip pat pastebima, kad iki nusistovint pusiausvyrajai srovei, kvazistacionariojo modelio atveju srovės vertės yra didesnės nei detaliojo modelio. Tai reiškia, kad jei kvazistacionariojo modelio skaičiavimų tikslumui nustatyti naudojama (4.6) formulė būtų taikoma be modulio, tai skaičiavimų tikslumo vertės būtų neigiamos.



13 pav. *Nedimensinės srovės priklausomybė nuo nedimensinio laiko, kai $Q=0.0013Ms$. Modeliavimui naudoti kiti parametrai tokie kaip (4.7).*

Kaip matyti iš 13 pav., kai Q yra didelis ($Q=0.0013Ms$), priešingai nei mažos Q reikšmės atveju, detaliojo modelio atveju gaunamos srovės vertės didėja greičiau nei kvazistacionariojo modelio atveju. Taip pat pastebima, kad ne visais atvejais nusistovi vienodos pusiausvyrosios srovės: kai $M_0^* \gg S_0^*$ (2 atvejis), detaliojo modelio atveju gaunama pusiausvyroji srovė yra ženkliai didesnė nei kvazistacionariojo modelio atveju. Toks skirtumas itin įtakoja ir kvazistacionariojo modelio skaičiavimų paklaidą, kuri šiuo atveju yra $\varepsilon = 90\%$. Būtent dėl skirtingų pusiausvyrųjų srovių 10 pav. (4 atvejis) ir pastebimos aukštos kvazistacionariojo modelio skaičiavimo paklaidos didelių Q intervale (10^{-5} - $10^{-2}Ms$).

Rezultatai

Šiame darbe gauti tokie rezultatai:

1. Pateikti reikalavimai, kuriuos turi atitikti kompiuteriniai modeliai, siekiant efektyviai panaudoti skaičiavimų tinklus.
2. Pateikta skaičiavimo tinklų panaudojimo metodika, skirta biokatalizės proceso parametrų įtakai tirti kompiuteriniu modeliu.
3. Sudaryta metodika pagrįsta sprendžiant parametrų įtakos biokatalizės procesui uždavinį, modeliavimo procesą atliekant naudojantis konkrečiu skaičiavimų tinklu.
4. Metodikos taikymas atliktas modeliuojant konkretaus biojutiklio veikimą.
5. Palyginti chemiškai modifikuoto biojutiklio detalūs ir kvazistacionarusis modeliai.
6. Įvertintas metodikos taikymo efektyvumas skaičiavimų tinkle *BalticGrid* bei pateiktos rekomendacijos metodikos taikymui.

Išvados

1. Darbe pateikiamą skaičiavimo tinklų panaudojimo metodiką galima taikyti atliekant biokatalizės proceso tyrimus. Metodika buvo pagrįsta eksperimentiškai, ją taikant konkretaus biojutiklio veikimo modeliavimui.

2. Metodikos taikymo efektyvumas priklauso nuo užduočių apimties – kuo užduotys didesnės, tuo efektyviau galima taikyti darbe pateikiamą metodiką. Taikant metodiką skaičiavimų tinkle *BalticGrid*, nustatyta, kad jos taikymo efektyvumas priklauso nuo skaidymo dydžio N_u : mažiausią efektyviai skaičiavimų tinkle sprendžiama užduotį, kuri lokaliame kompiuteryje užtrunka apie 40 minučių, rekomenduotina skaičiavimų tinkle suskaidyti į 3 užduotis. Didesnėms užduotims metodika efektyviausiai taikoma jas suskaidant į daugiau dalių: 2 valandas lokaliame kompiuteryje trunkančią užduotį geriausia skaidyti į 3-5 dalis, o užimančią 8 valandas – į 9-13 dalių.

3. Skaičiavimų, atliekamų naudojantis chemiškai modifikuoto biojutiklio kvazistacionariuoju modeliu, paklaida yra priklausoma nuo difuzijos modulio dalies Q . Prie itin mažų ($Q < 10^{-9}$ Ms) ir prie itin didelių ($Q > 10^{-3}$ Ms) Q reikšmių pastebima didelė kvazistacionariojo modelio skaičiavimų paklaida. Taip pat nustatyta, kad, naudojantis kvazistacionariuoju modeliu, skaičiavimų paklaidos nuo Q priklausomybė kinta priklausomai nuo bedimensinių mediatoriaus ir substrato koncentracijų. Šis pokytis itin pastebimas, kai $M_0^* \leq 1$, taip pat ir išskirtiniu atveju – kai $M_0^* \gg S_0^*$ (13 pav).

Šaltinių sąrašas

- [Abr05] D.Abramson. Applications Development for the Computational Grid. Lecture Notes in Computer Science, 3841, p.1 – 12. 2005.
- [AGK00] D.Abramson, J.Giddy, L.Kotler. High Performance Parametric Modelling with Nimrod/G: Killer Application for the Global Grid?. In Proc. of the Int. Parallel and Distributed Processing Symposium. IPDPS. 2000.
- [APL06] D.Abramson, T.Peachey, A.Lewis. Model Optimization and Parameter Estimation with Nimrod/O. The International Conference on Computational Science, Gegužės 28-31, 2006, University of Reading, Didžioji Britanija. 2006.
- [Ari75] R.Aris. The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts. The Theory of the Steady State. Clarendon Press, Oxford, 1975.
- [Aše07] V.Ašeris. Kompiuterinis biojutiklių veiksmo modeliavimas. Bakalauro darbas. 2007.
- [BBG+02] R.Buyya, K.Branson, J.Giddy, D.Abramson. The Virtual Laboratory: a toolset to enable distributed molecular modelling for drug design on the World-Wide Grid. Concurrency and Computation: Practice and Experience, p. 1-21. 2002.
- [BDM+04] R.Buyya, S.Date, Y.M.Matsumoto, S.Venugopa, D.Abramson. Neuroscience Instrumentation and Distributed Analysis of Brain Activity Data: A Case for eScience on Global Grids. 2004.
- [Ber02] V.Berstis. Fundamentals of Grid Computing. IBM Redbooks Paper. 2002.
- [BIK03a] R.Baronas, F.Ivanauskas, J.Kulys. The Influence of the Enzyme Membrane Thickness on the Response of Amperometric Biosensors. Sensors, 3, p. 248-262, 2003.
- [BK103b] R.Baronas, J.Kulys, F.Ivanauskas. Modelling amperometric enzyme electrode substrate cyclic conversion. Biosensors and Bioelectronics, 1106, p. 1-8, 2003.
- [Bin05] J.Binder. Aerospace America. Supercomputers: Where are they now? Aerospace America 2005-06. 2005.
- [BK08] R. Baronas, J.Kulys. Modelling Amperometric Biosensors Based on Chemically Modified Electrodes. Sensors, 8, 2008, p. 4800-4820.
- [Bri05] D.Britz. Digital Simulation in Electrochemistry. Springer-Verlag, Berlin, 3rd edition, 2005.
- [BSG05] K.K.Baldrige, W.Sudholt, J.P.Greenberg, C.Amoreira, Y.Potier, I.Altintas, A.Birnbaum, D.Abramson, C.Enticott, S.Garic. Cluster and Grid Infrastructure for Computational Chemistry and Biochemistry. 2005.

- [CB02] H.Casanova, F.Berman. Parameter Sweeps on the Grid with APST. Chapter 26, Willey Publisher, Inc., 2002.
- [CB80] P.W.Carr, L.D.Bowers. Immobilized Enzymes in Analytical and Clinical Chemistry: Fundamentals and Applications. John Wiley, New York, 1980.
- [CB90] M.Chaplin , C.Bucke. Enzyme Technology. Cambridge University Press, Cambridge, 1990.
- [CHJ+03] J.Chin, J.Harting, S.Jha, P. V.Coveney. Steering in computational science: mesoscale modelling and simulation. Contemporary Physics, 44(5), p 417-434. 2003.
- [CM05] E.Castro-Leon, J.Munter. Grid Computing Looking Forward. Technology@Intel Magazine, p.1-7. 2005.
- [CPP08] UK Computing for Particle Physics. Grid Middleware. 2008.
[žiūrėta 2008-05-21]. Prieiga per Internetą:
< <http://www.gridpp.ac.uk/explain.html>>
- [FK99a] I.Foster, C.Kesselman. The Grid. Blueprint for a new computing infrastructure. Morgan-Kaufman. 1999.
- [Fis96] P.A.Fishwick. Computer Simulation. IEEE Potentials, 15(1), p.24-27. 1996.
- [Fos02a] I.Foster. What is the Grid? A Three Point Checklist. GRIDToday, July 2002. 2002.
- [Fos02b] I.Foster. The Grid: A new infrastructure for 21st century science. Physics Today, February 2002, p. 51-63. 2002.
- [Gai06] E.Gaidamauskaitė. Biojutiklių kompiuterinių modelių sudarymas pagal veiksmo schemas. Magistro baigiamasis darbas. 2006.
- [GCP06] Grid Computing Planet. Grid projects. 2006.
[žiūrėta 2008-04-11]. Prieiga per Internetą:
<<http://www.gridcomputingplanet.com/resources/article.php/944961>>
- [Har05] S.Hartmann. The World as a Process: Simulations in the Natural and Social Sciences. Modeling and Simulation in the Social Sciences from the Philosophy of Science Point of View, p.77-100. 2005
- [Hon04] C.P.Hong. Computer Modelling of Heat and Fluid Flow in Materials Processing. Taylor and Francis. 2004.
- [Iva04] F.Ivanauskas. Lithuanian grids and projects. 2004.
[žiūrėta 2008-06-01]. Prieiga per Internetą:
<http://www.litgrid.lt/informpra/index_html/download>
- [Yag04] T.Yager. Taking a serious look at grids. 2004.
[žiūrėta 2008-06-01]. Prieiga per Internetą:

<http://www.infoworld.com/article/04/01/16/03FEgridsnow_1.html>

- [YMB+02] M.Yarrow, K.McCann, K.Biswas, R.Wijngaart. An advanced User Interface Approach for Complex Parameter Study Process Specification on the Information Power Grid (IPG). In Proc. of the 1st Workshop on Grid Computing (GRID 2002), 2002.
- [Koc05] G.Kochanski. Monte Carlo Simulation. 2005.
[žiūrėta 2008-05-22]. Prieiga per internetą:
<<http://kochanski.org/gpk/teaching/0401Oxford/MonteCarlo.pdf>>
- [Kou06] E.Kourpas. Grid Computing: Past, Present and Future. IBM. 2006.
- [KS74] B.Kvedaras, M.Sapagovas. Skaiciavimo metodai. Mintis, Vilnius, 1974.
- [LANL07] Los Alamos National laboratory. The Development of the Hydrogen Bomb. 2008.
[žiūrėta 2008-04-15]. Prieiga per internetą:
<<http://www.lanl.gov/history/postwar/development.shtml>>
- [LCG08] LHC Computing Grid. gLite 3.1 User Guide. 2008.
- [LSL08] B.Li, Y.Shen, B.Li. Quasi-steady-state laws in enzyme kinetics. The journal of physical chemistry. 2008.
- [Mul05] A.J.Mulholland. Modelling enzyme reaction mechanisms, specificity and catalysis. Drug Discovery Today, 10 (20), p.1393-1402. 2005.
- [PB01] F.Pacini, S.Beco. Job Description Language. 2001.
- [Per08] H.Perros. Computer Simulation Techniques: The definitive introduction. 2008.
- [Ric02] W.G.Richards. Virtual screening using grid computing: the screensaver project. Nature Reviews Drug Discovery, 1, p. 551-555, 2002.
- [SBA+04] W.Sudholt, K.Baldrige, D.Abramson, C.Enticott, S.Garic. Applying Grid Computing to the Parameter Sweep of a Group Difference pseudopotential. Lecture Notes in Computer Science, 3036, p.148-155. 2004.
- [Sam01] A.A.Samarskii. The Theory of Difference Schemes. Marcel Dekker, New York-Basel, 2001.
- [Sch90] T.Schulmeister. Mathematical modeling of the dynamic behaviour of amperometric enzyme electrodes. Selective Electrode Revue, 12, 203-260. 1990.
- [Shi04] S.Shimojo. High Performance and Secure Grid Network Middleware. 2004.
[žiūrėta 2008-01-27]. Prieiga per Internetą:
<http://www.naregi.org/papers/data/04_hpca_06.pdf>
- [SM03] S.Schnell, P.K.Maini. A Century of Enzyme Kinetics: Reliability of the K_M and v_{max} Estimates. Comments on Theoretical Biology, 8, p. 169–187. Taylor & Francis. 2003.

- [SS92] F.Scheller, F.Schubert. Biosensors. Elsevier, Amsterdam, 1992.
- [TKW87] A.P.F.Turner, I.Karube, G.S.Wilson. Biosensors: Fundamentals and Applications. Oxford University Press, Oxford, 1987.
- [WLS97] U. Wollenberger, F. Lisdat, F.W. Scheller. Frontiers in Biosensorics 2, Practical Applications, Birkhauser Verlag, Basel, 1997.

Priedai

1 Priedas: Naudotos Q reikšmės

E_0, M	d, m	Q, Ms
$1.0 \cdot 10^{-08}$	$1.0 \cdot 10^{-06}$	$3.3 \cdot 10^{-11}$
$1.2 \cdot 10^{-08}$	$1.2 \cdot 10^{-06}$	$5.8 \cdot 10^{-11}$
$1.4 \cdot 10^{-08}$	$1.4 \cdot 10^{-06}$	$1.0 \cdot 10^{-10}$
$1.7 \cdot 10^{-08}$	$1.7 \cdot 10^{-06}$	$1.7 \cdot 10^{-10}$
$2.1 \cdot 10^{-08}$	$2.1 \cdot 10^{-06}$	$3.0 \cdot 10^{-10}$
$2.5 \cdot 10^{-08}$	$2.5 \cdot 10^{-06}$	$5.1 \cdot 10^{-10}$
$3.0 \cdot 10^{-08}$	$3.0 \cdot 10^{-06}$	$8.9 \cdot 10^{-10}$
$3.6 \cdot 10^{-08}$	$3.6 \cdot 10^{-06}$	$1.5 \cdot 10^{-09}$
$4.3 \cdot 10^{-08}$	$4.3 \cdot 10^{-06}$	$2.6 \cdot 10^{-09}$
$5.2 \cdot 10^{-08}$	$5.2 \cdot 10^{-06}$	$4.6 \cdot 10^{-09}$
$6.2 \cdot 10^{-08}$	$6.2 \cdot 10^{-06}$	$7.9 \cdot 10^{-09}$
$7.4 \cdot 10^{-08}$	$7.4 \cdot 10^{-06}$	$1.4 \cdot 10^{-08}$
$8.9 \cdot 10^{-08}$	$8.9 \cdot 10^{-06}$	$2.4 \cdot 10^{-08}$
$1.1 \cdot 10^{-07}$	$1.1 \cdot 10^{-05}$	$4.1 \cdot 10^{-08}$
$1.3 \cdot 10^{-07}$	$1.3 \cdot 10^{-05}$	$7.1 \cdot 10^{-08}$
$1.5 \cdot 10^{-07}$	$1.5 \cdot 10^{-05}$	$1.2 \cdot 10^{-07}$
$1.8 \cdot 10^{-07}$	$1.8 \cdot 10^{-05}$	$2.1 \cdot 10^{-07}$
$2.2 \cdot 10^{-07}$	$2.2 \cdot 10^{-05}$	$3.6 \cdot 10^{-07}$
$2.7 \cdot 10^{-07}$	$2.7 \cdot 10^{-05}$	$6.3 \cdot 10^{-07}$
$3.2 \cdot 10^{-07}$	$3.2 \cdot 10^{-05}$	$1.1 \cdot 10^{-06}$
$3.8 \cdot 10^{-07}$	$3.8 \cdot 10^{-05}$	$1.9 \cdot 10^{-06}$
$4.6 \cdot 10^{-07}$	$4.6 \cdot 10^{-05}$	$3.2 \cdot 10^{-06}$
$5.5 \cdot 10^{-07}$	$5.5 \cdot 10^{-05}$	$5.6 \cdot 10^{-06}$
$6.6 \cdot 10^{-07}$	$6.6 \cdot 10^{-05}$	$9.7 \cdot 10^{-06}$
$7.9 \cdot 10^{-07}$	$7.9 \cdot 10^{-05}$	$1.7 \cdot 10^{-05}$
$9.5 \cdot 10^{-07}$	$9.5 \cdot 10^{-05}$	$2.9 \cdot 10^{-05}$
$1.1 \cdot 10^{-06}$	$1.1 \cdot 10^{-04}$	$5.0 \cdot 10^{-05}$
$1.4 \cdot 10^{-06}$	$1.4 \cdot 10^{-04}$	$8.6 \cdot 10^{-05}$
$1.6 \cdot 10^{-06}$	$1.6 \cdot 10^{-04}$	$1.5 \cdot 10^{-04}$
$2.0 \cdot 10^{-06}$	$2.0 \cdot 10^{-04}$	$2.6 \cdot 10^{-04}$
$2.4 \cdot 10^{-06}$	$2.4 \cdot 10^{-04}$	$4.5 \cdot 10^{-04}$
$2.8 \cdot 10^{-06}$	$2.8 \cdot 10^{-04}$	$7.7 \cdot 10^{-04}$
$3.4 \cdot 10^{-06}$	$3.4 \cdot 10^{-04}$	$1.3 \cdot 10^{-03}$
$4.1 \cdot 10^{-06}$	$4.1 \cdot 10^{-04}$	$2.3 \cdot 10^{-03}$
$4.9 \cdot 10^{-06}$	$4.9 \cdot 10^{-04}$	$4.0 \cdot 10^{-03}$
$5.9 \cdot 10^{-06}$	$5.9 \cdot 10^{-04}$	$6.9 \cdot 10^{-03}$
$7.1 \cdot 10^{-06}$	$7.1 \cdot 10^{-04}$	$1.2 \cdot 10^{-02}$
$8.5 \cdot 10^{-06}$	$8.5 \cdot 10^{-04}$	$2.1 \cdot 10^{-02}$
$1.0 \cdot 10^{-05}$	$1.0 \cdot 10^{-03}$	$3.5 \cdot 10^{-02}$