



PÖÖRDÜLESANDED MURRULISTELE ADSORPTSIOONIKINEETIKA VÕRRANDITELE

Bakalaureusetöö

Üliõpilane: Mattias Arakas
Üliõpilaskood: 233508YAFB
Juhendaja: Jaan Janno, professor
Õppekava: Rakendusfüüsika



INVERSE PROBLEMS FOR FRACTIONAL ADSORPTION KINETICS EQUATIONS

Bachelor's thesis

Student: Mattias Arakas
Student code: 233508YAFB
Supervisor: Jaan Janno, professor
Study programme: Applied Physics

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Mattias Arakas

Töö vastab bakalaureusetööle esitatavatele nõuetele.

Juhendaja: Jaan Janno

Sisukord

| | | |
|----------|--|-----------|
| 1 | Sissejuhatus | 6 |
| 2 | Matemaatilised eelteadmised | 7 |
| 2.1 | Päripidi- ja pöördülesanne | 7 |
| 2.2 | Parameetrite hindamine vähimruutude meetodil | 7 |
| 2.3 | Caputo murruline tuletis | 8 |
| 2.4 | Laplace'i teisendus | 9 |
| 2.5 | Mittag–Leffleri funktsioon | 10 |
| 2.6 | Caputo tuletise L1-diskretiseerimine | 11 |
| 2.7 | Sünteesilised andmed ja veamudel | 15 |
| 3 | Fraktaalne adsorptsioonimudel | 17 |
| 3.1 | Mudeli tuletus | 17 |
| 3.2 | Päripidiülesande analüütiline lahend | 18 |
| 3.3 | Märkus mudeli olemuse kohta | 19 |
| 3.4 | Pöördülesande püstitus | 19 |
| 4 | Murruline adsorptsioonimudel | 22 |
| 4.1 | Erijuht $n = 1$ | 22 |
| 4.2 | Üldjuht $n > 1$ | 23 |
| 4.3 | L1-skeemi rakendamine | 24 |
| 4.4 | Pöördülesande püstitus | 26 |
| 4.5 | Numbriliste katsete meetodika | 27 |
| 4.5.1 | Sünteesiliste andmete genereerimine | 28 |
| 4.5.2 | Parameetrite valik | 28 |

| | | |
|----------|--|-----------|
| 4.5.3 | Mudelivalik | 29 |
| 4.5.4 | Pöördülesande lahendamine | 30 |
| 4.5.5 | Koonduvus ja täpsus | 30 |
| 4.5.6 | L1-skeemi numbriline valideerimine | 31 |
| 4.5.7 | Statistiline valideerimine | 31 |
| 5 | Tulemused | 32 |
| 5.1 | Fraktaalne mudel: võrrand (8) | 32 |
| 5.2 | Murruline mudel: võrrand (16) | 33 |
| 5.3 | Mudelite võrdlus ja müratundlikkus | 35 |
| 5.4 | Mudeli järgu n tuvastamine | 37 |
| 6 | Kokkuvõte | 39 |
| | Tänuavaldused | 40 |
| | Annotatsioon | 43 |
| | Abstract | 44 |
| | Lisad | 45 |
| | Lisa 1 – Lihtlitsents | 45 |
| | Lisa 2 – Pythoni skriptid | 46 |

1 Sissejuhatus

Adsorptsioon on protsess, mille käigus lahustunud aine koguneb tahke materjali pinnale. Selliste protsesside kineetika kirjeldamine on oluline keskkonnatehnoloogias ja materjaliteadustes. Klassikalised kineetilised mudelid, nagu pseudo-esimest ja pseudo-teist järku mudelid [10, 12], on laialdaselt kasutatavad, kuid need ei suuda alati kirjeldada keerulisemat, mitte-eksponentsiaalset käitumist, mida eksperimentides täheldatakse.

Viimastel aastakümnetel on murruliste tuletiste [16, 5] tuginev modelleerimine loonud alternatiivse lähenemise, mis võimaldab kirjeldada mäludefektiga ja justkui anomaalse dünaamikaga protsesse [9, 14]. Bakalis ja Zerbetto [2] on esitanud süstemaatilise võrdluse klassikalise, fraktaalset ja murrulise kineetika vahel adsorptsiooniprotsesside kontekstis. Nende töö on käesoleva bakalaureusetöö peamine alusartikkel.

Töö eesmärk on uurida fraktaalset ja murrulise adsorptsioonimudeli pöördülesannete lahendamist sünteetiliste andmete põhjal. Päripidiülesandena mõistetakse mudeli lahendi arvutamist teadaolevate parameetrite korral. Pöördülesandes on eesmärk vastupidine. Nimelt püütakse mõõdetud andmete põhjal taastada mudeli tundmatud parameetrid. Pöördülesanded on tüüpiliselt halvasti püstitatud [7, 8], mistõttu on oluline hinnata parameetrite taastatavuse täpsust ja stabiilsust müra suhtes.

Esmalt käsitletakse fraktaalset pseudo- n -järku mudelit, kus mittestandardne kineetika tekib aja skaleerimisest, ning teisena murrulist pseudo- n -järku mudelit, kus täisarvuline tuletis on asendatud Caputo murrulise tuletisega. Mõlema mudeli korral genereeritakse teadaolevate parameetritega sünteetilised andmed, lisatakse neile müra ning lahendatakse pöördülesanne vähimruutude meetodit kasutades. Fraktaalset mudeli korral viiakse katsed läbi järkudega $n = 1, 2, 3$ ning murrulise mudeli korral käsitletakse juhtu $n = 1$ analüütiliselt ning numbrilised katsed teostatakse järkudega $n = 2, 3$. Müratasemeid testitakse vahemikus $\delta \in \{0,01; 0,05; 0,10\}$. Lisaks uuritakse olukorda, kus ka mudeli järk n on tundmatu ning tuleb andmete põhjal määrata. Järku määramist teostatakse täisarvulise optimeerimisülesandena. Kõik katsed on teostatud sünteetilises seadistuses, kus tasakaaluline adsorptsioonimaht g_e on fikseeritud. See lihtsustab pöördülesannet ning tulemusi tuleb vastavalt tõlgendada.

2 Matemaatilised eelteadmised

Selles peatükis tuuakse sisse vajaminevad matemaatilised definitsioonid, millele toetuvad järgnevad peatükid. Töö keskendub adsorptsioonimudelite pöördülesannetele, mistõttu defineeritakse esmalt päripidi- ja pöördülesande mõisted, vähimruutude meetod ning murrulise analüüsi põhiobjektid. Lisaks kirjeldatakse lühidalt Caputo murrulise tuletise diskretiseerimist L1-skeemi abil.

2.1 Päripidi- ja pöördülesanne

Matemaatilise mudeli kontekstis nimetatakse *päripidiülesandeks* olukorda, kus mudeli kuju ja selle parameetrid on teada ning eesmärk on leida süsteemi vastus. Üldisel kujul võib seda kirjutada kui

$$\mathcal{F}(g, \theta) = 0, \quad (2.1.1)$$

kus g tähistab otsitavat lahendit ja θ mudeli parameetrivektorit. Kui θ on teada, siis päripidiülesanne seisneb lahendi u leidmises.

Pöördülesandes on olukord vastupidine. Olemas on mõõtmisandmed või sünteetilised andmed, mis kirjeldavad süsteemi lahendit, ning eesmärk on nende põhjal taastada mudeliparameetrid. Käesolevas töös tähendab see adsorptsioonimudeli parameetrite leidmist funktsiooni $g(t)$ vaadeldud väärtuste põhjal. Pöördülesanded on sageli halvasti püstitatud selles mõttes, et lahend ei pruugi olla üheselt määratud või võib see olla väga tundlik mõõtevea suhtes [7, 8, 1].

Kui vaatlushetked on t_1, t_2, \dots, t_N ja mõõdetud andmed on y_1, y_2, \dots, y_N , siis saab parameetrite määramise ülesande kirjutada kujul

$$y_j \approx g(t_j; \theta), \quad j = 1, 2, \dots, N, \quad (2.1.2)$$

kus $g(t; \theta)$ on mudeli poolt ennustatud vastus ajahetkel t , kui parameetrivektor on θ .

2.2 Parameetrite hindamine vähimruutude meetodil

Pöördülesannete lahendamine toimub läbi vähimruutude meetodi. Selle idee on valida selline parameetrivektor θ , mille korral mudeli poolt ennustatud väärtused $g(t_j; \theta)$ oleksid võimalikult lähedased vaadeldud andmetele y_j .

Defineerime jäägi ajahetkel t_j valemiga

$$r_j(\theta) = g(t_j; \theta) - y_j. \quad (2.2.1)$$

Siis vähimruutude sihtfunktsioon on

$$J(\theta) = \sum_{j=1}^N r_j(\theta)^2 = \sum_{j=1}^N (g(t_j; \theta) - y_j)^2. \quad (2.2.2)$$

Siin

- N on mõõtmispunktide arv,
- t_j on j -s mõõtmisaeg,
- y_j on mõõdetud või sünteetiliselt genereeritud väärtus hetkel t_j ,
- $g(t_j; \theta)$ on mudeli väärtus samal ajahetkel,
- θ on tundmatute parameetrite vektor.

Vähimruutude meetodi eesmärk on lahendada optimeerimisülesanne

$$\theta^* = \arg \min_{\theta \in \Theta} J(\theta), \quad (2.2.3)$$

kus Θ on lubatud parameetrite hulk. Praktikast valitakse parameetritele sageli kitsendused, näiteks positiivsused

$$0 < a \leq 1, \quad k_n > 0, \quad 0 < \alpha < 1,$$

sest need on mudeli füüsikalise tähenduse seisukohalt loomulikud.

Vähimruutude meetod on mittelineaarsete parameetrite hindamisel üks standardseid lähenemisi. Kui mudel sõltub parameetritest mittelineaarselt, siis sihtfunktsiooni (2.2.2) miinimum leitakse tavaliselt numbriliste optimeerimismeetoditega [15, 1].

2.3 Caputo murruline tuletis

Murruline analüüs üldistab klassikalise integraali ja tuletise mõisted mitte-täisarvulistele väärtustele. Murruliste operaatorite definitsioonides esineb gammafunktsioon

$$\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx, \quad \alpha > 0, \quad (2.3.1)$$

mis üldistab faktoriaali: $\Gamma(n + 1) = n!$ iga $n \in \mathbb{N} \cup \{0\}$ korral.

Käesolevas töös kasutatakse Caputo tüüpi murrulist tuletist, sest see võimaldab algtingimusi sõnastada täisarvuliste tuletiste kaudu, mis on rakenduste seisukohalt mugavam kui Riemann–Liouville’i tuletis.

Olgu $0 < \alpha < 1$. Siis funktsiooni f Caputo murruline tuletis on defineeritud kui

$${}^C D_t^\alpha f(t) = \frac{1}{\Gamma(1 - \alpha)} \int_0^t (t - \tau)^{-\alpha} f'(\tau) d\tau. \quad (2.3.2)$$

Üldisemalt, kui $m - 1 < \alpha < m$, siis

$${}^C D_t^\alpha f(t) = \frac{1}{\Gamma(m - \alpha)} \int_0^t (t - \tau)^{m - \alpha - 1} f^{(m)}(\tau) d\tau. \quad (2.3.3)$$

Caputo tuletise üks tähtsamaid omadusi on see, et konstantse funktsiooni murruline tuletis on null:

$${}^C D_t^\alpha C = 0.$$

See on kooskõlas klassikalise diferentsiaalarvutusega ja lihtsustab algväärtusülesannete käsitlemist [5, 16].

Üks Caputo definitsiooni eelistamise põhjuseid Riemann–Liouville’i definitsiooni ees on see, et Riemann–Liouville’i murruline tuletis konstantsest funktsioonist ei ole null:

$$D_t^\alpha 1 = \frac{t^{-\alpha}}{\Gamma(1 - \alpha)} \neq 0,$$

mis on füüsikaliselt ebaloomulik. Lisaks tuleb Riemann–Liouville’i definitsiooniga diferentsiaalvõrrandite algtingimused anda murdintegraali $I^{m-\alpha} f(0)$ kaudu, mis ei ole otseselt mõõdetav suurus. Caputo definitsiooni korral on algtingimused aga antud täisarvuliste tuletiste kaudu.

2.4 Laplace’i teisendus

Laplace’i teisendus on oluline tööriist diferentsiaalvõrrandite lahendamisel, sest see teisendab diferentsiaalvõrrandi algebraliseks võrrandiks.

Funktsiooni $f(t)$, mis on defineeritud vahemikus $t \geq 0$, Laplace’i teisendus on defineeritud valemiga

$$\mathcal{L}\{f(t)\}(s) = F(s) = \int_0^\infty f(t) e^{-st} dt, \quad (2.4.1)$$

kus s on kompleksne sagedusmuutuja ning integraal eksisteerib piisavalt suurte $\operatorname{Re}(s)$

väärtuste korral [17, 16]. Funktsiooni $F(s)$ nimetatakse funktsiooni $f(t)$ Laplace'i kujutiseks.

Laplace'i teisendus on lineaarne:

$$\mathcal{L}\{c_1 f(t) + c_2 g(t)\}(s) = c_1 F(s) + c_2 G(s). \quad (2.4.2)$$

Tavalise n -ndat järku tuletise Laplace'i teisendus on

$$\mathcal{L}\{f^{(n)}(t)\}(s) = s^n F(s) - \sum_{k=0}^{n-1} s^{n-1-k} f^{(k)}(0), \quad (2.4.3)$$

mis võimaldab diferentsiaalvõrrandid teisendada algebralisteks [17].

Caputo murrulise tuletise Laplace'i teisendus juhul $m - 1 < \alpha < m$ on [16, 5]

$$\mathcal{L}\{{}^C D_t^\alpha f(t)\}(s) = s^\alpha F(s) - \sum_{k=0}^{m-1} s^{\alpha-1-k} f^{(k)}(0). \quad (2.4.4)$$

Kui $0 < \alpha < 1$ (st $m = 1$) ja $f(0) = 0$, siis lihtsustub see kujule

$$\mathcal{L}\{{}^C D_t^\alpha f(t)\}(s) = s^\alpha F(s). \quad (2.4.5)$$

Just seda seost kasutatakse hiljem alampeatükis 4.3 murrulise mudeli analüütilisel lahendamisel.

2.5 Mittag–Leffleri funktsioon

Murruliste diferentsiaalvõrrandite lahendites esineb väga sageli Mittag–Leffleri funktsioon, mis mängib murrulises analüüsis sarnast rolli nagu eksponentfunktsioon tavalistes diferentsiaalvõrrandites.

Üheparameetriline Mittag–Leffleri funktsioon defineeritakse reaga

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha > 0. \quad (2.5.1)$$

See rida koondub kogu kompleksstasandil iga $\alpha > 0$ korral [6]. Kaheparameetriline Mittag–Leffleri funktsioon on

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad \alpha > 0, \beta > 0. \quad (2.5.2)$$

Kui $\alpha = 1$, siis saame tagasi eksponentfunktsiooni:

$$E_1(z) = e^z.$$

Mittag–Leffleri funktsiooni oluline omadus on selle Laplace'i teisenduse paar. Kui $\alpha > 0$ ja $k > 0$, siis kehtib

$$\mathcal{L}\{E_\alpha(-k t^\alpha)\}(s) = \frac{s^{\alpha-1}}{s^\alpha + k}. \quad (2.5.3)$$

Üldisemalt, kaheparameetrilise Mittag–Leffleri funktsiooni korral kehtib

$$\mathcal{L}\{t^{\beta-1} E_{\alpha,\beta}(\lambda t^\alpha)\}(s) = \frac{s^{\alpha-\beta}}{s^\alpha - \lambda}. \quad (2.5.4)$$

Valem (2.5.3) on erijuht $\beta = 1$, $\lambda = -k$ [6, 16].

Mittag–Leffleri funktsioon ilmub loomulikult selliste murdvõrrandite lahendites nagu

$${}^C D_t^\alpha g(t) = \lambda g(t), \quad g(0) = g_0, \quad (2.5.5)$$

mille lahend on

$$g(t) = g_0 E_\alpha(\lambda t^\alpha). \quad (2.5.6)$$

See erijuht on oluline, sest see annab analüütilise kontrollpunkti murrulise mudeli jaoks [6, 16, 5].

2.6 Caputo tuletise L1-diskretiseerimine

Murrulise adsorptsioonimudeli päripidiülesande numbriliseks lahendamiseks on vaja Caputo tuletis diskretiseerida. Selleks kasutatakse L1-skeemi. [13, 19, 5].

Olgu ajavõrk ühtlane:

$$t_n = nh, \quad n = 0, 1, \dots, N, \quad (2.6.1)$$

kus $h > 0$ on ajasamm. Tuletame nüüd L1-skeemi Caputo tuletise definitsioonist. Kui $0 < \alpha < 1$, siis Caputo tuletis on antud kujul

$${}^C D_t^\alpha g(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} g'(\tau) d\tau. \quad (2.6.2)$$

Huvitume selle väärtusest võrgupunktis $t = t_n$. Siis

$${}^C D_t^\alpha g(t_n) = \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} (t_n - \tau)^{-\alpha} g'(\tau) d\tau. \quad (2.6.3)$$

Jagame integraali alamvahemikeks $[t_k, t_{k+1}]$, kus $k = 0, 1, \dots, n-1$:

$${}^C D_t^\alpha g(t_n) = \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} (t_n - \tau)^{-\alpha} g'(\tau) d\tau. \quad (2.6.4)$$

L1-skeemi põhieeldus seisneb selles, et igal lõigul $[t_k, t_{k+1}]$ asendatakse täisarvuline tuletis $g'(\tau)$ selle lõigu lõpp-punktidest saadud jagatisega:

$$g'(\tau) \approx \frac{g(t_{k+1}) - g(t_k)}{h}, \quad \tau \in [t_k, t_{k+1}]. \quad (2.6.5)$$

See tähendab, et funktsiooni g käsitletakse igal alamvahemikul lineaarselt interpoleerituna, mistõttu tuletis on seal ligikaudu konstantne.

Asendades lähenduse (2.6.5) valemisse (2.6.4), saame

$${}^C D_t^\alpha g(t_n) \approx \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \frac{g(t_{k+1}) - g(t_k)}{h} \int_{t_k}^{t_{k+1}} (t_n - \tau)^{-\alpha} d\tau. \quad (2.6.6)$$

Jääb üle integraal

$$\int_{t_k}^{t_{k+1}} (t_n - \tau)^{-\alpha} d\tau.$$

Antud integraali lahendamiseks leiame algfunktsiooni. Kuna $0 < \alpha < 1$, siis

$$\int (t_n - \tau)^{-\alpha} d\tau = -\frac{(t_n - \tau)^{1-\alpha}}{1-\alpha}.$$

Seega

$$\begin{aligned} \int_{t_k}^{t_{k+1}} (t_n - \tau)^{-\alpha} d\tau &= \left[-\frac{(t_n - \tau)^{1-\alpha}}{1-\alpha} \right]_{\tau=t_k}^{\tau=t_{k+1}} \\ &= \frac{(t_n - t_k)^{1-\alpha} - (t_n - t_{k+1})^{1-\alpha}}{1-\alpha}. \end{aligned} \quad (2.6.7)$$

Asendades saadud tulemuse tagasi valemisse (2.6.6), saame

$${}^C D_t^\alpha g(t_n) \approx \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \frac{g(t_{k+1}) - g(t_k)}{h} \cdot \frac{(t_n - t_k)^{1-\alpha} - (t_n - t_{k+1})^{1-\alpha}}{1-\alpha}. \quad (2.6.8)$$

Kasutame nüüd seost

$$(1 - \alpha)\Gamma(1 - \alpha) = \Gamma(2 - \alpha),$$

ning kirjutame

$${}^C D_t^\alpha g(t_n) \approx \frac{1}{\Gamma(2 - \alpha)} \sum_{k=0}^{n-1} \frac{g(t_{k+1}) - g(t_k)}{h} \left((t_n - t_k)^{1-\alpha} - (t_n - t_{k+1})^{1-\alpha} \right). \quad (2.6.9)$$

Kuna võrk on ühtlane ja $t_j = jh$, siis

$$t_n - t_k = (n - k)h, \quad t_n - t_{k+1} = (n - k - 1)h.$$

Seega

$$(t_n - t_k)^{1-\alpha} - (t_n - t_{k+1})^{1-\alpha} = h^{1-\alpha} \left((n - k)^{1-\alpha} - (n - k - 1)^{1-\alpha} \right).$$

Asendades selle valemisse (2.6.9), saame

$$\begin{aligned} {}^C D_t^\alpha g(t_n) &\approx \frac{1}{\Gamma(2 - \alpha)} \sum_{k=0}^{n-1} \frac{g(t_{k+1}) - g(t_k)}{h} \cdot h^{1-\alpha} \left((n - k)^{1-\alpha} - (n - k - 1)^{1-\alpha} \right) \\ &= \frac{1}{\Gamma(2 - \alpha) h^\alpha} \sum_{k=0}^{n-1} \left((n - k)^{1-\alpha} - (n - k - 1)^{1-\alpha} \right) (g(t_{k+1}) - g(t_k)). \end{aligned} \quad (2.6.10)$$

Saadud kuju on korrektne L1-skeem. Sageli teostatakse veel indeksivahetus, et kaalud oleksid kirjutatud kasvava indeksi abil: Olgu

$$j = n - k - 1.$$

Siis, kui $k = 0$, on $j = n - 1$, ja kui $k = n - 1$, on $j = 0$. Summat ümber kirjutades saame

$${}^C D_t^\alpha g(t_n) \approx \frac{1}{\Gamma(2 - \alpha) h^\alpha} \sum_{j=0}^{n-1} \left((j + 1)^{1-\alpha} - j^{1-\alpha} \right) (g(t_{n-j}) - g(t_{n-j-1})). \quad (2.6.11)$$

Saadud tulemus (2.6.11) ongi Caputo tuletise L1-diskretisatsioon ühtlasel ajavõrgul. Piisavalt siledate lahendite korral on L1-skeemi klassikaline veajärk $O(h^{2-\alpha})$ [13, 19]. Murruliste algväärtusülesannete tüüpilised lahendid käituvad aga punkti $t = 0$ läheduses kui $g(t) - g(0) \sim t^\alpha$, mistõttu lahendi esimene tuletis on $t = 0$ läheduses singulaarne. Ühtlasel ajavõrgul langeb sellistel mittesiledatel lahenditel L1-skeemi koonduvusjärk $O(h^\alpha)$ -ni [18, 11]. Klassikalise järgu $O(h^{2-\alpha})$ taastamiseks tuleks

kasutada graderitud ajavõrku, kus sammud on nulli läheduses tihedamalt.

Saadud tulemuses

- h on ajasamm,
- $t_n = nh$ on diskreetne ajahetk,
- $\alpha \in (0, 1)$ on murrulise tuletise järk,
- $g(t_n)$ on lahendi ligikaudne väärtus võrgupunktis t_n ,
- tegurid

$$b_j = (j + 1)^{1-\alpha} - j^{1-\alpha}$$

on L1-skeemi kaalud.

Valemis (2.6.11) sõltub uus väärtus kõigist eelnevatest väärtustest $g(t_0), g(t_1), \dots, g(t_n)$. See peegeldab murrulise tuletise mitte-lokaalset iseloomu ja tähendab, et diskreetsel kujul säilib süsteemi mälu efekt.

Kui vaadeldav diferentsiaalvõrrand on mittelineaarne, siis sisaldab diskreetne samm tavaliselt suurust $g(t_n)$ mittelineaarsel kujul. Sellisel juhul ei saa uut väärtust otse valemist avaldada, vaid tuleb igal ajasammul tuleb lahendada mittelineaarne skalaarne võrrand. Just sellist lähenemist kasutatakse hiljem murrulise adsorptsioonimudeli päripidiülesande lahendamisel.

2.7 Sünteetilised andmed ja veamudel

Arvutuskatsetes kasutatakse sünteetilisi andmeid. Selleks valitakse esmalt parameetrid θ_{true} ning lahendatakse päripidiülesanne. Saadud väärtusi

$$g(t_j; \theta_{\text{true}})$$

käsitletakse ideaalsete andmetena.

Mõõtevea modelleerimiseks lisatakse neile müra:

$$y_j = g(t_j; \theta_{\text{true}}) + \varepsilon_j, \quad (2.7.1)$$

kus ε_j tähistab mürakomponenti. Müra on ühtlaselt jaotunud lõigus $[-\delta, \delta]$:

$$\varepsilon_j \sim \mathcal{U}(-\delta, \delta),$$

kus $\delta > 0$ on mürataseme parameeter. Selline veamudel tagab, et häiritus on tõkestatud ($|\varepsilon_j| \leq \delta$ alati), mis lihtsustab pöördülesande stabiilsuse analüüsi.

Seejärel leitakse vähimruutude meetodil hinnang θ^* ja võrreldakse seda tegelike parameetritega. Hindamise kvaliteeti mõõdetakse parameetri p suhtelise veaga

$$\delta_p = \frac{|p_{\text{true}} - p^*|}{|p_{\text{true}}|} \cdot 100\%, \quad (2.7.2)$$

kus p_{true} on parameetri tegelik väärtus ja p^* selle hinnang. Selline lähenemine võimaldab hinnata, kui tundlik on parameetrite taastamine andmete mürale [1, 8].

3 Fraktaalne adsorptsioonimudel

Mõlema mudeli korral on otsitavaks funktsiooniks $g(t)$, mis kirjeldab adsorbeerunud aine hulka ajahetkel t . Fraktaalse mudeli puhul on päripidiülesande eelis see, et mitmel juhul on olemas analüütiline lahend. Murrulise mudeli puhul tuleb üldjuhul päripidiülesanne lahendada numbriliselt ning alles seejärel saab moodustada vähimruutude sihtfunktsiooni, mis teeb murrulise mudeli pöördülesande arvutuslikult kallimaks.

Käesolevas peatükis vaadeldakse fraktaalset pseudo- n -järku adsorptsioonimudelit, mis on klassikalise kineetilise mudeli üldistus. Fraktaalse mudeli korral muudetakse aja skaleeringut, kuid diferentsiaaloperaator jääb täisarvuliseks tuletiseks. Alusartiklis on vastav mudel esitatud võrrandina (8) [2].

3.1 Mudeli tuletus

Olgu $g(t)$ adsorbeerunud aine hulk ajahetkel t ja g_e tasakaalu adsorptsioonimaht. Klassikaline pseudo- n -järku adsorptsioonivõrrand on kujul

$$\frac{dg(t)}{dt} = k_n(g_e - g(t))^n. \quad (3.1.1)$$

Siin $k_n > 0$ on kineetiline konstant ja n on mudeli järku määrav parameeter [10].

Fraktaalse mudeli korral eeldatakse, et protsess areneb mitte harilikus ajas t , vaid fraktaalses ajas

$$\xi = t^a, \quad 0 < a \leq 1. \quad (3.1.2)$$

Seega kirjutatakse kineetiline seadus kujul

$$\frac{dg(t)}{d(t^a)} = k_n(g_e - g(t))^n. \quad (3.1.3)$$

Rakendades liitfunktsiooni tuletise reeglid,

$$\frac{dg(t)}{dt} = \frac{dg(t)}{d(t^a)} \cdot \frac{d(t^a)}{dt},$$

ja kasutades seost

$$\frac{d(t^a)}{dt} = at^{a-1},$$

saame

$$\frac{dg(t)}{dt} = ak_n t^{a-1} (g_e - g(t))^n. \quad (3.1.4)$$

See ongi fraktaalne pseudo- n -järku adsorptsioonivõrrand. Kui $a = 1$, siis taandub (3.1.4) tagasi klassikaliseks mudeliks (3.1.1).

3.2 Päripidiülesande analüütiline lahend

Eeldame algtingimust

$$g(0) = 0. \quad (3.2.1)$$

Juht $n = 1$

Kui $n = 1$, siis võrrand (3.1.4) on

$$\frac{dg(t)}{dt} = ak_1 t^{a-1} (g_e - g(t)). \quad (3.2.2)$$

See on eraldatavate muutujatega võrrand

$$\frac{dg(t)}{g_e - g(t)} = ak_1 t^{a-1} dt.$$

Integreerides mõlemad pooled, saame

$$-\ln(g_e - g(t)) = k_1 t^a + C.$$

Võttes mõlemad võrrandi poole e astmeks ja kasutades algtingimust (3.2.1), järeldub

$$g(t) = g_e (1 - e^{-k_1 t^a}). \quad (3.2.3)$$

Juht $n \neq 1$

Kui $n \neq 1$, siis (3.1.4) annab

$$\frac{dg(t)}{(g_e - g(t))^n} = ak_n t^{a-1} dt.$$

Integreerides vasakut poolt asendusega $u = g_e - g(t)$ ja paremat poolt otse, saame

$$\frac{(g_e - g(t))^{1-n}}{n-1} = k_n t^a + C.$$

Algtingimusest (3.2.1) järeldub

$$C = \frac{g_e^{1-n}}{n-1}.$$

Seega

$$(g_e - g(t))^{1-n} = (n-1)k_n t^a + g_e^{1-n}.$$

Parempoolsest avaldisest saab välja tuua teguri g_e^{1-n} :

$$(g_e - g(t))^{1-n} = g_e^{1-n} [1 + (n-1)k_n g_e^{n-1} t^a].$$

Avaldades siit $g(t)$, saame

$$g(t) = g_e \left\{ 1 - [1 + (n-1)k_n g_e^{n-1} t^a]^{\frac{1}{1-n}} \right\}. \quad (3.2.4)$$

Tulemused (3.2.3) ja (3.2.4) annavad fraktaalse mudeli päripidiülesande lahendi. Need langevad kokku alusartiklis toodud kujuga [2].

3.3 Märkus mudeli olemuse kohta

Fraktaalse mudeli korral on vasakul pool endiselt täisarvuline tuletis $\frac{dg}{dt}$. Mittestandardne kineetika tekib siin parempoolses liikmes oleva ajasõltuva kordaja at^{a-1} kaudu. Seetõttu jääb mudel lokaalseks ehk muutumise kiirus ajahetkel t sõltub üksnes sama hetke väärtusest $g(t)$ ja ajast t . See eristabki hetkel vaadeltavat fraktaalset mudelit hiljem käsitletavast murrulisest mudelist, kus Caputo tuletise tõttu tekib mitte-lokaalsus ja mälu efekt [2].

Siinkohal on oluline märkida, et võrrand (3.1.4) sisaldab kordajat at^{a-1} , mis on singulaarne punktis $t = 0$, kui $a < 1$, sest $\lim_{t \rightarrow 0^+} t^{a-1} = +\infty$. Sellest hoolimata on analüütilised lahendid (3.2.3) ja (3.2.4) regulaarsed kogu vahemikus $[0, \infty)$ ning rahuldavad algtingimust $g(0) = 0$. Seetõttu tuleb diferentsiaalvõrrandit (3.1.4) mõista poolavatud vahemikus $t > 0$, kusjuures algtingimus kehtib lahendi piirväärtuse kaudu $\lim_{t \rightarrow 0^+} g(t) = 0$.

3.4 Pöördülesande püstitus

Olgu antud mõõtmispunktid

$$t_1, t_2, \dots, t_N$$

ja neile vastavad vaadeldud väärtused

$$y_1, y_2, \dots, y_N,$$

kus

$$y_j \approx g(t_j), \quad j = 1, 2, \dots, N. \quad (3.4.1)$$

Töös käsitletakse olukorda, kus n ja g_e on teada ning hinnatakse parameetreid

$$\theta = (a, k_n). \quad (3.4.2)$$

Siis mudeli väärtus punktis t_j on

$$g(t_j; \theta),$$

kus kasutatakse vastavalt kas valemit (3.2.3) või (3.2.4).

Defineerime jäägid

$$r_j(\theta) = g(t_j; \theta) - y_j, \quad j = 1, 2, \dots, N, \quad (3.4.3)$$

ja vähimruutude funktsionaali

$$J(\theta) = \sum_{j=1}^N r_j(\theta)^2 = \sum_{j=1}^N (g(t_j; \theta) - y_j)^2. \quad (3.4.4)$$

Pöördülesanne seisneb sellise parameetrivektori leidmises, mille korral funktsionaal (3.4.4) on minimaalne:

$$\theta^* = \arg \min_{\theta \in \Theta} J(\theta), \quad (3.4.5)$$

kus loomulikuks lubatud parameetrite hulgaks võib võtta

$$\Theta = \{(a, k_n) : 0 < a \leq 1, k_n > 0\}. \quad (3.4.6)$$

Kui $n = 1$, siis mudeli valem on

$$g(t_j; \theta) = g_e(1 - e^{-k_1 t_j^a}), \quad (3.4.7)$$

ning vähimruutude funktsionaal on

$$J(a, k_1) = \sum_{j=1}^N (g_e(1 - e^{-k_1 t_j^a}) - y_j)^2. \quad (3.4.8)$$

Kui $n \neq 1$, siis kasutatakse mudelit

$$g(t_j; \theta) = g_e \left\{ 1 - [1 + (n-1)k_n g_e^{n-1} t_j^a]^{\frac{1}{1-n}} \right\}, \quad (3.4.9)$$

ning vastav funktsionaal on

$$J(a, k_n) = \sum_{j=1}^N \left[g_e \left(1 - [1 + (n-1)k_n g_e^{n-1} t_j^a]^{\frac{1}{1-n}} \right) - y_j \right]^2. \quad (3.4.10)$$

Fraktaalse mudeli oluline eelis seisneb selles, et päripidiülesande lahendamine taandub valemite (3.2.3) ja (3.2.4) otsesele väärtustamisele: iga ajahetke t_j jaoks saab $g(t_j)$ arvutada ühe valemiga, ilma et oleks vaja lahendada diferentsiaalvõrrandit numbriliselt. Arvutuslik keerukus on seega $O(1)$ punkti kohta ja $O(N)$ kogu ajavõrgu kohta.

4 Murruline adsorptsioonimudel

Klassikaline pseudo- n -järku adsorptsioonimudel on kujul

$$\frac{dg(t)}{dt} = k_n (g_e - g(t))^n. \quad (4.0.1)$$

Selle murruline üldistus saadakse, kui vasakpoolne täisarvuline tuletis asendada Caputo murrulise tuletisega järku $\alpha \in (0, 1)$:

$${}^C D_t^\alpha g(t) = k_n (g_e - g(t))^n, \quad 0 < \alpha < 1. \quad (4.0.2)$$

See ongi alusartikli võrrand (16) ning tähistus a on siin asendatud sümboliga α [2].

Mudelis

- $g(t)$ on adsorbeerunud aine hulk ajahetkel t ,
- g_e on tasakaaluline adsorptsioonimaht,
- $k_n > 0$ on kineetiline konstant,
- n on mudeli järk,
- α on Caputo murrulise tuletise järk.

Eeldame algtingimust

$$g(0) = 0. \quad (4.0.3)$$

4.1 Erijuht $n = 1$

Kui $n = 1$, siis võrrand (4.0.2) muutub lineaarseks:

$${}^C D_t^\alpha g(t) = k_1 (g_e - g(t)). \quad (4.1.1)$$

Kirjutame selle ümber kujule

$${}^C D_t^\alpha g(t) + k_1 g(t) = k_1 g_e. \quad (4.1.2)$$

Rakendame Laplace'i teisendust. Caputo tuletise Laplace'i teisenduse definitsiooni (2.4.4) kohaselt kehtib

$$\mathcal{L}\{{}^C D_t^\alpha g(t)\}(s) = s^\alpha G(s) - s^{\alpha-1} g(0).$$

Kuna $g(0) = 0$, saame võrrandist (4.1.2)

$$s^\alpha G(s) + k_1 G(s) = \frac{k_1 g_e}{s}.$$

Seega

$$G(s) = \frac{k_1 g_e}{s(s^\alpha + k_1)}. \quad (4.1.3)$$

Seda saab kirjutada kujul

$$G(s) = g_e \left(\frac{1}{s} - \frac{s^{\alpha-1}}{s^\alpha + k_1} \right).$$

Kasutades Mittag–Leffleri funktsiooni Laplace'i paari (2.5.3)

$$\mathcal{L}\{E_\alpha(-k_1 t^\alpha)\}(s) = \frac{s^{\alpha-1}}{s^\alpha + k_1},$$

saame pöördteisenduse tulemusena

$$g(t) = g_e(1 - E_\alpha(-k_1 t^\alpha)). \quad (4.1.4)$$

See langeb kokku alusartiklis antud pseudo-esimest järku murrulise kineetika lahendiga [2].

Kui $\alpha = 1$, siis kehtib $E_1(-k_1 t) = e^{-k_1 t}$ ning (4.1.4) taandub klassikaliseks lahendiks

$$g(t) = g_e(1 - e^{-k_1 t}).$$

4.2 Üldjuht $n > 1$

Kui $n > 1$, siis võrrand (4.0.2) on mittelineaarne murd-diferentsiaalvõrrand

$${}^C D_t^\alpha g(t) = k_n (g_e - g(t))^n, \quad g(0) = 0. \quad (4.2.1)$$

Erinevalt fraktaalsetest mudelist ei ole siin üldjuhul võimalik lahendit suletud kujul avaldada. Alusartiklis on näidatud, et juhu $n = 2$ korral saab lahendit arendada astmereaks, kuid üldisema pöördülesande ja numbriliste katsete jaoks on otstarbekam kasutada numbrilist lahendamist [2].

Üldjuhu (4.2.1) lahendamiseks kasutatakse L1-diskretisatsiooni.

4.3 L1-skeemi rakendamine

Olgu ajavõrk ühtlane:

$$t_m = mh, \quad m = 0, 1, \dots, M, \quad (4.3.1)$$

kus $h > 0$ on ajasamm. L1-skeemi järgi ligikaudistatakse Caputo tuletist punktis t_m järgnevalt

$${}^C D_t^\alpha g(t_m) \approx \frac{1}{\Gamma(2-\alpha)h^\alpha} \sum_{j=0}^{m-1} ((j+1)^{1-\alpha} - j^{1-\alpha})(g_{m-j} - g_{m-j-1}), \quad (4.3.2)$$

kus tähistus

$$g_m \approx g(t_m)$$

märgib lahendi ligikaudset väärtust võrgupunktis t_m .

Asendades (4.3.2) võrrandisse (4.2.1), saame

$$\frac{1}{\Gamma(2-\alpha)h^\alpha} \sum_{j=0}^{m-1} ((j+1)^{1-\alpha} - j^{1-\alpha})(g_{m-j} - g_{m-j-1}) = k_n(g_e - g_m)^n. \quad (4.3.3)$$

Tõstame summast välja liikme $j = 0$. Kuna

$$(1)^{1-\alpha} - 0^{1-\alpha} = 1,$$

saame

$$\frac{1}{\Gamma(2-\alpha)h^\alpha} \left[(g_m - g_{m-1}) + \sum_{j=1}^{m-1} ((j+1)^{1-\alpha} - j^{1-\alpha})(g_{m-j} - g_{m-j-1}) \right] = k_n(g_e - g_m)^n. \quad (4.3.4)$$

Korrutades mõlemad pooled teguriga $\Gamma(2-\alpha)h^\alpha$, jõuame kujuni

$$g_m - g_{m-1} + \sum_{j=1}^{m-1} ((j+1)^{1-\alpha} - j^{1-\alpha})(g_{m-j} - g_{m-j-1}) = \Gamma(2-\alpha)h^\alpha k_n(g_e - g_m)^n. \quad (4.3.5)$$

Seega tuleb igal ajasammul lahendada mittelineaarne skalaarne võrrand tundmatu g_m suhtes:

$$g_m - \Gamma(2-\alpha)h^\alpha k_n(g_e - g_m)^n = g_{m-1} - \sum_{j=1}^{m-1} ((j+1)^{1-\alpha} - j^{1-\alpha})(g_{m-j} - g_{m-j-1}). \quad (4.3.6)$$

Just sellist tüüpi sammvõrrandit kasutatakse hiljem koodis päripidiülesande lahendamiseks.

Arvutuslik protseduur

Tähistame mugavuse huvides

$$A = \frac{1}{\Gamma(2 - \alpha) h^\alpha}$$

ning defineerime igal sammul m ajaloo summa

$$H_m = \sum_{j=1}^{m-1} b_j (g_{m-j} - g_{m-j-1}), \quad (4.3.7)$$

kus $b_j = (j + 1)^{1-\alpha} - j^{1-\alpha}$ on L1 kaalud. Summa H_m sisaldab ainult varem arvutatud väärtusi g_0, g_1, \dots, g_{m-1} , seega on see igal sammul teada suurus. Esimesel sammul ($m = 1$) on $H_1 = 0$, sest summa on tühi.

Võrrandi (4.3.6) saab nüüd kirjutada kujul: otsitakse selline $x = g_m$, et

$$F(x) = A(x - g_{m-1} + H_m) - k_n (g_e - x)^n = 0. \quad (4.3.8)$$

Funktsiooni F omadused on järgmised:

- kui $x = 0$, siis $F(0) = A(-g_{m-1} + H_m) - k_n g_e^n$; kõigis käesolevas töös teostatud arvutuskatsetes oli $F(0) < 0$;
- kui $x = g_e$, siis $F(g_e) = A(g_e - g_{m-1} + H_m) > 0$, sest lahend on monotoonselt kasvav ja $A > 0$;
- F on pidev ja rangelt kasvav lõigul $[0, g_e]$, sest selle tuletis on

$$F'(x) = A + n k_n (g_e - x)^{n-1} > 0 \quad \text{iga } x \in [0, g_e], \quad (4.3.9)$$

kuna $A > 0$, $k_n > 0$, $n \geq 1$ ja $g_e - x > 0$.

Kuna F on rangelt kasvav, saab võrrandil $F(x) = 0$ lõigus $[0, g_e]$ olla kõige rohkem üks lahend.

Arvutuskatsetes kontrolliti igal sammul märgimuutust lõigul $[0, g_e]$. Kõigis töös kasutatud parameetrikombinatsioonides kehtis $F(0) < 0$ ja $F(g_e) > 0$, mistõttu oli poolitusmeetod rakendatav ning andis üheselt määratud lahendi füüsikaliselt lubatud vahemikus.

Algoritm (L1-skeem).

1. **Sisend:** parameetrid α, k_n, n, g_e ; ajavõrk t_0, t_1, \dots, t_M sammuga h ; algtingimus $g_0 = 0$.
2. Arvutatakse eeltegur $A = 1/(\Gamma(2 - \alpha) h^\alpha)$ ja L1 kaalud $b_j = (j + 1)^{1-\alpha} - j^{1-\alpha}$ kõigi $j = 0, 1, \dots, M - 1$ jaoks.
3. **Tsükkel** üle ajasammude $m = 1, 2, \dots, M$:
 - (a) arvutatakse ajaloo summa H_m valemiga (4.3.7), kasutades varem leitud väärtusi g_0, \dots, g_{m-1} ;
 - (b) defineeritakse funktsioon $F(x) = A(x - g_{m-1} + H_m) - k_n(g_e - x)^n$;
 - (c) leitakse g_m poolitusmeetodiga lõigul $[0, g_e]$, lahendades võrrandi $F(x) = 0$ etteantud täpsusega;
 - (d) kärbitakse tulemus füüsikalistesse piiridesse: $g_m \leftarrow \max(0, \min(g_m, g_e))$.
4. **Väljund:** lahendi väärtused g_0, g_1, \dots, g_M .

Oluline on rõhutada, et sammu 3(a) arvutuslik keerukus kasvab lineaarselt: sammul m tuleb summeerida $m - 1$ liidetavat. Kogu lahenduse arvutamine nõuab seega $O(M^2)$ operatsiooni. See on oluliselt kallim kui fraktaalse mudeli analüütiline valem, mille väärtustamine ühes punktis on $O(1)$. Siin tähistab $M + 1$ päripidiülesande lahendamiseks kasutatud ajavõrgu punktide arvu. Pöördülesandes kasutatud vaatluspunktide arv N võib olla sama või sellest erinev. Kasutati $M = 3000$ ajasammu (st $M + 1 = 3001$ võrgupunkti), mis tagas piisava täpsuse (vt veajärku $O(h^{2-\alpha})$ alampeatükis 2.6) ning mõistliku arvutusaja.

4.4 Pöördülesande püstitus

Olgu antud mõõtmispunktid

$$t_1, t_2, \dots, t_N$$

ja neile vastavad vaadeldud väärtused

$$y_1, y_2, \dots, y_N,$$

kus

$$y_j \approx g(t_j), \quad j = 1, 2, \dots, N. \quad (4.4.1)$$

Defineerime jäägid

$$r_j(\theta) = g(t_j; \theta) - y_j, \quad j = 1, 2, \dots, N, \quad (4.4.2)$$

kus $g(t_j; \theta)$ saadakse:

- valemist (4.1.4), kui $n = 1$;
- L1-skeemi abil lahendatud diskreetse päripidiülesande kaudu, kui $n > 1$.

Vähimruutude funktsionaal on

$$J(\theta) = \sum_{j=1}^N (g(t_j; \theta) - y_j)^2 = \sum_{j=1}^N r_j(\theta)^2. \quad (4.4.3)$$

Pöördülesanne seisneb sellise parameetrivektori leidmises, mille korral (4.4.3) on minimaalne:

$$\theta^* = \arg \min_{\theta \in \Theta} J(\theta), \quad (4.4.4)$$

kus loomulikuks lubatud parameetrite hulgaks võib võtta

$$\Theta = \{(\alpha, k_n) : 0 < \alpha < 1, k_n > 0\}. \quad (4.4.5)$$

Kui $n = 1$, siis mudeli väärtus on analüütiliselt antud:

$$g(t_j; \alpha, k_1) = g_e(1 - E_\alpha(-k_1 t_j^\alpha)), \quad (4.4.6)$$

ning vähimruutude funktsionaal on

$$J(\alpha, k_1) = \sum_{j=1}^N (g_e(1 - E_\alpha(-k_1 t_j^\alpha)) - y_j)^2. \quad (4.4.7)$$

Kui $n > 1$, siis mudeli väärtus $g(t_j; \alpha, k_n)$ leitakse kõigepealt numbriliselt, lahendades samm-sammult võrrandi (4.3.6) ehk iga optimeerimissamm eelab esmalt numbrilist lahendamist.

4.5 Numbriliste katsete meetodika

Päripidiülesande lahendamiseks vaadeldi ajavahemikku

$$t \in [0, T],$$

kus lõpphetkeks valiti $T = 10$. Ajavahemik diskretiseeriti ühtlase sammuga võrguks

$$t_m = mh, \quad m = 0, 1, \dots, M,$$

kus $h = \frac{T}{M}$ on ajasamm ja M on ajasammude arv. Sünteetilised mõõtmised genereeriti samadel võrgupunktidel, mistõttu on vaatluspunktide arv $N = M + 1 = 3001$. See valik tagab murrulise mudeli L1-skeemi jaoks piisava täpsuse (4.5.6) ning tulemuste otsese võrreldavuse mudelite vahel.

4.5.1 Sünteetiliste andmete genereerimine

Eksperimentides kasutati sünteetilisi andmeid, mis genereeriti mudeli lahendi põhjal. Olgu $g^*(t)$ mudeli lahend tegelike parameetrite θ^* korral. Mõõteandmed konstrueeriti kujul

$$y_i = g^*(t_i) + \varepsilon_i,$$

kus ε_i on juhuslik müra.

Juhuslike arvude genereerimisel kasutati fikseeritud algseadistust, et tagada tulemuste taastatavus.

Oluline piirang on see, et sünteetilised andmed genereeriti sama mudeliklassi põhjal, mille parameetreid hiljem hinnati. Seetõttu kontrollivad katsed eelkõige pöördülesande arvutusliku lahendamise korrektsust antud mudeliraamistiku sees, mitte mudeli võimet kirjeldada sõltumatuid eksperimentaalseid andmeid. Numbrilise pöördülesande kirjanduses nimetatakse sellist ohtu sageli pöördülesande liiga soodsaks testimiseks.

4.5.2 Parameetrite valik

Sünteetiliste andmete genereerimiseks tuleb valida tegelikud parameetrid θ_{true} , mille põhjal arvutatakse mudelilahend. Valik peab rahuldama kahte nõuet: parameetrid peavad kuuluma füüsiliselt lubatud piirkonda ning asuma selle piirkonna siseosas (mitte piiridel).

Tasakaaluline adsorptsioonimaht valiti $g_e = 1,0$. See on sobiv valik, sest mudeli valemites esineb g_e alati kordajana, siis võib üldisust kaotamata fikseerida $g_e = 1$ ja tõlgendada $g(t)$ kui adsorbeerunud aine suhtelist hulka.

Fraktaalse mudeli korral (võrrand (3.1.4)) valiti tegelikeks parameetriteks

$$a_{\text{true}} = 0,7, \quad k_{\text{true}} = 0,9.$$

Parameeter $a = 0,7$ asub lubatud vahemiku $(0, 1]$ siseosas. Äärejuht $a = 1$ taandaks mudeli klassikaliseks. Väga väikesed väärtused ($a \ll 1$) tekitaksid numbriliselt raskusi, kuna kordaja $a t^{a-1}$ muutub singulaarseks punktis $t = 0$. Väärtus $a = 0,7$ väldib mõlemat äärejuhtu. Kineetiline konstant $k = 0,9$ on mõõduka suurusega positiivne arv, mis tagab, et lahendikõver on selgelt eristatav nii nullist kui tasakaaluväärtusest vaadeldavas ajavahemikus $[0, 10]$.

Murrulise mudeli korral (võrrand (4.0.2)) valiti

$$\alpha_{\text{true}} = 0,8, \quad k_{\text{true}} = 1,0.$$

Parameeter $\alpha = 0,8$ asub lubatud vahemiku $(0, 1)$ siseosas. Äärejuht $\alpha \rightarrow 1$ taandaks Caputo tuletise täisarvuliseks tuletiseks ning äärejuht $\alpha \rightarrow 0$ muudaks lahendi käitumise kvalitatiivselt erinevaks ning piirväärtus on matemaatiliselt mittetriviaalne [5]. Väärtus $\alpha = 0,8$ on piisavalt kaugel mõlemast piirist. Kineetiline konstant $k = 1,0$ on valitud sarnastel kaalutlustel nagu fraktaalse mudeli korral.

Kahe mudeli tegelikud parameetrid on teadlikult valitud erinevaks ($a \neq \alpha$, $k_{\text{fraktaalne}} \neq k_{\text{murruline}}$), et tulemused ei sõltuks parameetrite juhuslikust kokkusobivusest.

Mõlema mudeli korral loeti tundmatuteks parameetriteks ainult (a, k_n) (fraktaalne) või (α, k_n) (murruline). Parameetri g_e fikseerimine vähendab parameetritevahelist korrelatsiooni ning parandab pöördülesande stabiilsust. Seetõttu käsitletakse pöördülesande lihtsustatud varianti; kui ka g_e oleks tundmatu, oleks parameetrite eristamine üldjuhul keerulisem. Mudeli järke testiti väärtustega $n = 1, 2, 3$ fraktaalse mudeli korral ning $n = 2, 3$ murrulise mudeli korral.

Optimeerimise algväärtusteks valiti parameetrite tegelikest väärtustest mõnevõrra erinevad väärtused (näiteks $a_0 = 0,8$ ja $k_0 = 1,0$ fraktaalse mudeli korral), et kontrollida, kas optimeerija leiab õiged parameetrid ka siis, kui alglähenend ei ole täpne.

4.5.3 Mudelivalik

Erijuhuna võib pöördülesandes olla tundmatuks ka mudeli struktuuriparameeter, mis määrab mudeli kvalitatiivse käitumise. Selliseks parameetriks on mudeli järk n , mis kirjeldab adsorptsiooni mittelineaarsuse astet. Kui n väärtus ei ole ette teada, muutub ülesanne *mudelivaliku ülesandeks* (ingl *model selection*) [4, 1]. Üks levinud lähenemine on lahendada pöördülesanne eraldi iga $-n$ korral ning valida see n , mille

korral sihtfunktsiooni (2.2.2) optimaalne väärtus on väikseim:

$$n^* = \arg \min_{n \in \mathcal{S}} J(\theta^*(n)), \quad (4.5.1)$$

kus \mathcal{S} on kandidaatväärtuste hulk (näiteks $\mathcal{S} = \{1, 2, 3, 4, 5\}$) ja $\theta^*(n)$ tähistab iga konkreetse n korral leitud optimaalseid parameetreid.

Mudelivaliku katsetes lahendati pöördülesanne eraldi iga kandidaatväärtuse $n \in \{1, 2, 3, 4, 5\}$ korral ning valiti see n , mille korral sihtfunktsiooni väärtus oli väikseim. Iga kandidaadi korral hinnati parameetrid (a, k_n) (fraktaalne) või (α, k_n) (murruline) samamoodi nagu fikseeritud n korral.

4.5.4 Pöördülesande lahendamine

Parameetrite määramiseks kasutati vähimruutude meetodit, mille korral minimeeriti funktsionaali

$$J(\theta) = \sum_{i=1}^N (g(t_i; \theta) - y_i)^2.$$

Optimeerimisülesanne lahendati SciPy [20] funktsiooniga `least_squares`, mis kasutab usalduspiirkonna peegeldavmeetodit (ingl *trust-region-reflective*, TRF) [3, 15]. Iga iteratsiooni korral arvutati päripidiülesande lahend antud parameetrite väärtuste korral.

Parameetritele seati füüsikaliselt põhjendatud kitsendused: $0 < a \leq 1$ (või $0 < \alpha < 1$) ja $k_n > 0$.

4.5.5 Koonduvus ja täpsus

Optimeerimisprotsess lõpetati, kui muutused funktsionaali väärtuses, parameetrites või gradiendis muutusid piisavalt väikesteks. Fraktaalse mudeli korral kasutati vaike-tolerantse. Murrulise mudeli korral seati rangemad tolerantsid $ftol = xtol = gtol = 10^{-10}$, et tagada piisav täpsus ka juhul, kui päripidiülesande numbriline lahendamine sisaldab diskretisatsiooniviga. Saadud parameetrite hinnanguid võrreldi tegelike väärtustega, et hinnata pöördülesande lahendamise täpsust.

4.5.6 L1-skeemi numbriline valideerimine

Kuna murrulise mudeli pöördülesanne kasutab L1-skeemi sisemise lahendajana, on oluline kontrollida, et diskretisatsiooniviga ei moonutaks parameetrite hinnanguid. Selleks kasutati erijuhtu $n = 1$, mille korral on olemas analüütiline lahend (4.1.4). L1-skeemiga arvatud lahendeid parameetritega $\alpha = 0,8$, $k_1 = 1,0$, $g_e = 1,0$, $t \in [0, 10]$ võrreldi analüütilise Mittag–Leffleri lahendiga erinevate ajasammude h korral.

Tulemused näitasid, et maksimaalne viga kahaneb sammuga võrdeliselt $h^{0,8}$, mis vastab järgule $O(h^\alpha)$. See ei vasta siledate lahendite teoreetilisele järgule $O(h^{2-\alpha}) = O(h^{1,2})$, kuid on täielikus kooskõlas tüüpilise FDE-lahendi mittesileduse mõjuga: analüütilise lahendi $g(t) = g_e(1 - E_\alpha(-k_1 t^\alpha))$ Mittag–Leffleri funktsiooni reaarendusest (4.1.4) järel

$$g(t) \sim \frac{g_e k_1}{\Gamma(\alpha + 1)} t^\alpha \quad \text{kui } t \rightarrow 0^+, \quad (4.5.2)$$

mistõttu $g'(t) \sim t^{\alpha-1}$ on punkti $t = 0$ läheduses singulaarne. Stynes et al. tulemuse [18] kohaselt langeb sellistel lahenditel L1-skeemi ühtlasel ajavõrgul koonduvusjärk täpselt $O(h^\alpha)$ -ni, mis on kooskõlas käesoleva eksperimendi tulemustega.

Katsetes kasutatud $N = 3001$ korral ($h \approx 0,0033$) oli maksimaalne diskretisatsiooniviga suurusjärgus 2×10^{-3} , mis on tunduvalt väiksem kõigist testitud müratasemetest.

4.5.7 Statistiline valideerimine

Pöördülesande tulemuste statistilise usaldusväärsuse kontrollimiseks korraldati katseid erinevate juhuslike algseadistustega. Mõlema mudeli korral teostati iga (n, δ) kombinatsiooni jaoks 20 kordust. Mudeli järgu tuvastamise katsetes (jaotis 5.4) kasutati fraktaalse mudeli korral samuti 20 realisatsiooni, kuid murrulise mudeli korral piirduiti arvutusliku keerukuse tõttu 10 realisatsiooniga. Tulemuste esitamisel on toodud suhteliste vigade keskmine ja standardhälve.

5 Tulemused

5.1 Fraktaalne mudel: võrrand (8)

Fraktaalse mudeli pöördülesande jaoks valiti tegelikeks parameetriteks (valikut on põhjendatud alampeatükis 4.5.2)

$$a_{\text{true}} = 0,7, \quad k_{\text{true}} = 0,9, \quad g_e = 1,0.$$

Parameetrit g_e hoiti fikseerituna ning hinnati ainult parameetreid a ja k_n . Kuna fraktaalse mudeli päripidiülesande lahend on analüütiliselt antud valemitega (3.2.3) ja (3.2.4), siis on iga optimeerimissamm arvutuslikult väga odav. Katsetati kolme erinevat järku ($n = 1, 2, 3$) ning kolme mürataset ($\delta = 0,01; 0,05; 0,10$) ehk kokku üheksa kombinatsiooni.

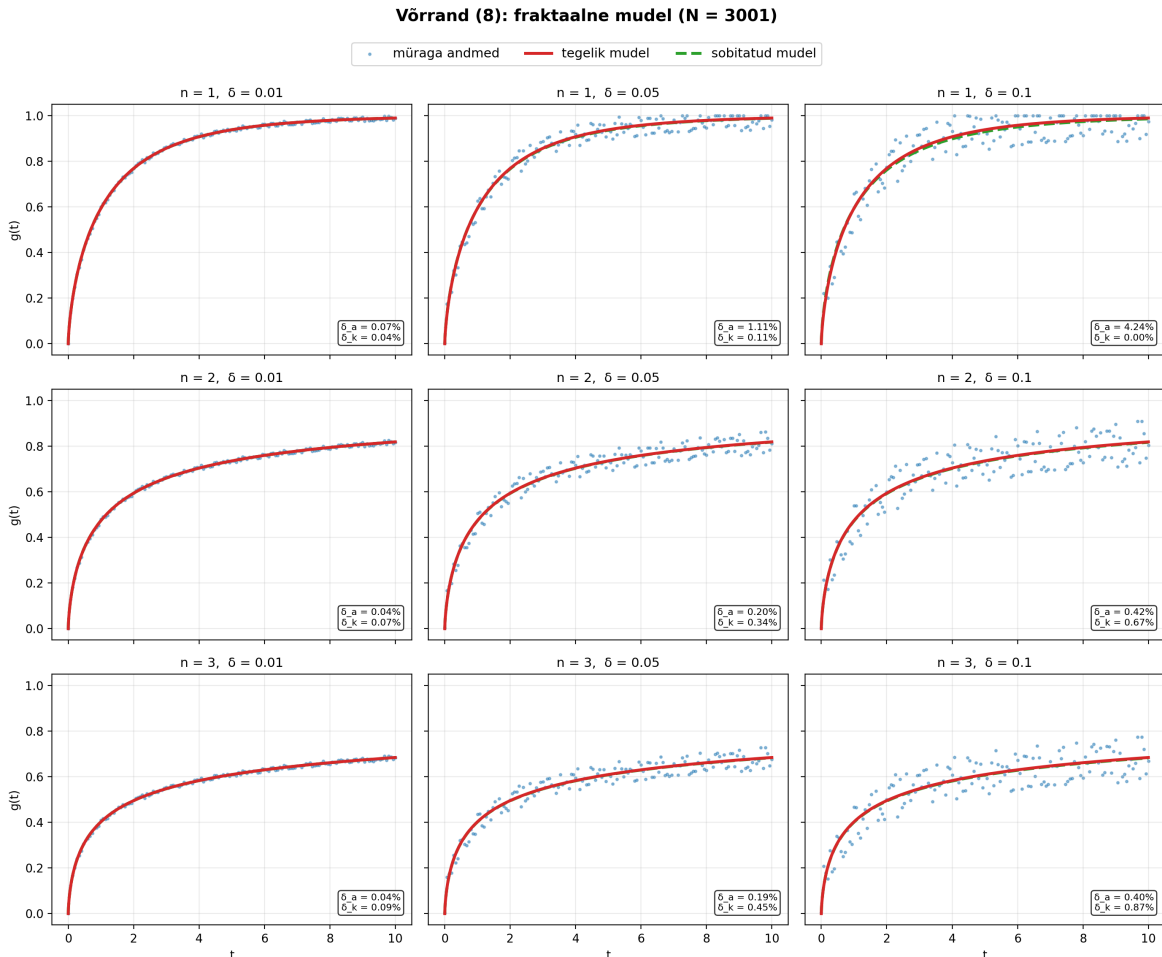
Kõigi üheksa kombinatsiooni korral koondus optimeerimisprotsess edukalt iga realisatsiooni korral. Tabelis 1 on esitatud suhteliste vigade keskmised ja standardhälbed 20 müra realisatsiooni üle. Väikse müra korral ($\delta = 0,01$) jäävad keskmised suhtelised vead alla 0,1%. Müra suurenedes kasvavad vead mõõdukalt: taseme $\delta = 0,10$ korral jäävad suhtelised vead järkude $n = 2$ ja $n = 3$ korral alla 1%. Erandiks on juht $n = 1, \delta = 0,10$, kus parameetri a keskmine viga on 3,7% — see võib olla tingitud juhust $n = 1$ lahendikõvera kujust, mis on parameetri a suhtes vähem tundlik suurte t väärtuste juures. Katsetes ei ilmnenud, et mudeli järk n tekitaks iseenesest ebastabiilsust ja vigade kasv oli eeskätt seotud mürataseme suurenemisega.

Tabel 1: Fraktaalse mudeli, võrrandi (8), pöördülesande tulemused ($N = 3001$). Tegelikud parameetrid: $a_{\text{true}} = 0,7, k_{\text{true}} = 0,9, g_e = 1,0$. Suhtelised vead on esitatud kujul keskmine \pm standardhälve 20 müra realisatsiooni üle.

| n | δ | $\delta_a \pm \sigma$ (%) | $\delta_k \pm \sigma$ (%) | $\bar{J}(\theta^*)$ |
|-----|----------|---------------------------|---------------------------|----------------------|
| 1 | 0,01 | $0,06 \pm 0,04$ | $0,05 \pm 0,03$ | $4,9 \times 10^{-2}$ |
| 1 | 0,05 | $0,77 \pm 0,36$ | $0,26 \pm 0,18$ | $1,1 \times 10^0$ |
| 1 | 0,10 | $3,67 \pm 0,67$ | $0,58 \pm 0,45$ | $4,1 \times 10^0$ |
| 2 | 0,01 | $0,06 \pm 0,04$ | $0,07 \pm 0,05$ | $4,9 \times 10^{-2}$ |
| 2 | 0,05 | $0,28 \pm 0,22$ | $0,37 \pm 0,24$ | $1,3 \times 10^0$ |
| 2 | 0,10 | $0,56 \pm 0,44$ | $0,75 \pm 0,48$ | $5,0 \times 10^0$ |
| 3 | 0,01 | $0,07 \pm 0,05$ | $0,10 \pm 0,06$ | $5,0 \times 10^{-2}$ |
| 3 | 0,05 | $0,34 \pm 0,26$ | $0,48 \pm 0,30$ | $1,2 \times 10^0$ |
| 3 | 0,10 | $0,69 \pm 0,52$ | $0,97 \pm 0,60$ | $4,9 \times 10^0$ |

Neid tulemusi illustreerib joonis 1, kus on kujutatud kõigi üheksa katse müraga sünteetilised andmed, tegelik mudel ja sobitatud mudel. Veergude kaupa on näha,

kuidas suurenev müratase muudab andmeid üha hajusamaks, samas kui sobitatud mudel jääb tegeliku mudeli lähedale. Ridade kaupa on näha, et mudeli järk n muudab lahendikõvera kuju, kuid pöördülesande lahendamine toimib kõigi järkude korral korrektselt.



Joonis 1: Fraktaalse mudeli, võrrandi (8), pöördülesande tulemused kõigi testitud n ja δ kombinatsioonide korral. Sinised punktid — müraga sünteetilised andmed, punane joon — tegelik mudel, roheline kriipsjoon — sobitatud mudel. Igas alamjoonis on toodud suhtelised vead δ_a ja δ_k .

5.2 Murruline mudel: võrrand (16)

Murrulise mudeli pöördülesande jaoks valiti tegelikeks parameetriteks (vt jaotis 4.5.2)

$$\alpha_{\text{true}} = 0,8, \quad k_{\text{true}} = 1,0, \quad n \in \{2, 3\}, \quad g_e = 1,0, \quad g(0) = 0.$$

Päripidiülesanne lahendati L1-skeemiga. Sarnaselt eelmise jaotisega hinnati ainult parameetreid α ja k_n , hoides g_e fikseerituna. Katsetati järke $n = 2$ ja $n = 3$ ning kolme mürataset, kokku kuus kombinatsiooni.

Kõigis kuues kombinatsioonis koondus optimeerimisprotsess edukalt iga realisatsiooni korral. Tabelis 2 esitatud tulemused näitavad, et ka murrulise mudeli korral on väikese müra juures ($\delta = 0,01$) suhtelised vead alla 0,3%. Suurema müra korral ($\delta = 0,10$) ulatuvad parameetri k_n keskmised vead kuni 2,1%, mis on märgatavalt suurem kui fraktaalse mudeli vastavad vead samal müratasemel. See erinevus on osaliselt tingitud sellest, et murrulise mudeli L1-skeem toob sisse täiendava diskretisatsioonivea, mida fraktaalse mudeli analüütiline lahend ei sisalda.

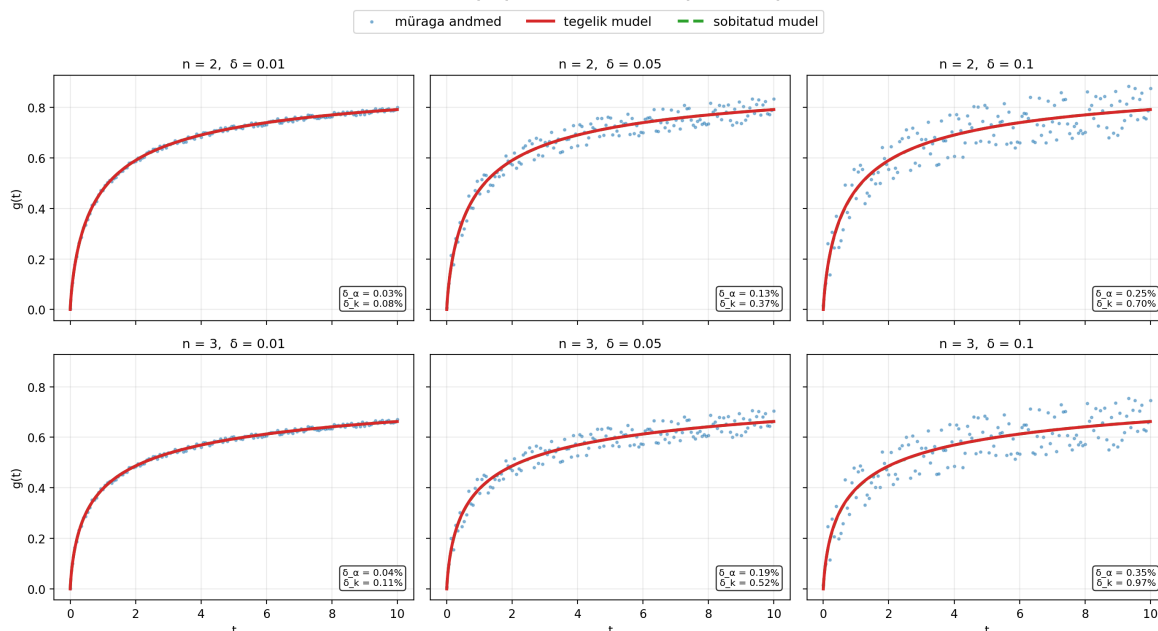
Tabel 2: Murrulise mudeli, võrrandi (16), pöördülesande tulemused ($N = 3001$). Tegelikud parameetrid: $\alpha_{\text{true}} = 0,8$, $k_{\text{true}} = 1,0$, $g_e = 1,0$. Suhtelised vead on esitatud kujul keskmine \pm standardhälve 20 müra realisatsiooni üle.

| n | δ | $\bar{\delta}_\alpha \pm \sigma$ (%) | $\bar{\delta}_k \pm \sigma$ (%) | $J(\theta^*)$ |
|-----|----------|--------------------------------------|---------------------------------|----------------------|
| 2 | 0,01 | $0,06 \pm 0,03$ | $0,16 \pm 0,09$ | $5,0 \times 10^{-2}$ |
| 2 | 0,05 | $0,30 \pm 0,16$ | $0,79 \pm 0,43$ | $1,2 \times 10^0$ |
| 2 | 0,10 | $0,61 \pm 0,32$ | $1,58 \pm 0,88$ | $4,9 \times 10^0$ |
| 3 | 0,01 | $0,08 \pm 0,04$ | $0,21 \pm 0,11$ | $5,0 \times 10^{-2}$ |
| 3 | 0,05 | $0,39 \pm 0,20$ | $1,04 \pm 0,56$ | $1,2 \times 10^0$ |
| 3 | 0,10 | $0,79 \pm 0,42$ | $2,10 \pm 1,12$ | $4,9 \times 10^0$ |

Joonisel 2 on esitatud kõigi kuue katse tulemused. Sarnaselt fraktaalse mudeliga jääb sobitatud mudel kõigil juhtudel tegeliku mudeli lähedale, kuigi tabelite 1 ja 2 kvantitatiivne võrdlus näitab, et murrulise mudeli parameetrite hinnangud on keskmiselt veidi vähem täpsed.

Oluline on märkida, et murrulise mudeli korral on iga sihtfunktsiooni väärtustamine arvutuslikult oluliselt kallim, sest see eeldab L1-skeemiga päripidiülesande lahendamist. Katsetes kasutatud $N = 3001$ võrgupunktiiga lahendamine võttis iga parameetri-kombinatsiooni jaoks suurusjärgus kümme sekundit, samas kui fraktaalse mudeli analüütiline otsene lahend võimaldas kogu optimeerimise läbi viia alla sekundi.

Võrrand (16): murruline mudel (N = 3001)



Joonis 2: Murrulise mudeli, võrrandi (16), pöördülesande tulemused kõigi testitud n ja δ kombinatsioonide korral. Sinised punktid — müraga sünteetilised andmed, punane joon — tegelik mudel, roheline kriipsjoon — sobitatud mudel. Igas alamjoonises on toodud suhtelised vead δ_α ja δ_k .

5.3 Mudelite võrdlus ja müratundlikkus

Kahe mudeli tulemuste võrdlemiseks on tabelis 3 kõrvutatud keskmised suhtelised vead juhtudel, kus mõlema mudeli jaoks on tulemused olemas, st järkude $n = 2$ ja $n = 3$ korral. Mõlema mudeli tulemused põhinevad samal andmepunktide arvul $N = 3001$, mis tagab tulemuste otsese võrreldavuse.

Tabel 3: Fraktaalse ja murrulise mudeli keskmiste suhteliste vigade võrdlus (%). Mõlemad mudelid kasutavad $N = 3001$ võrgupunkti.

| n | δ | Võrrand (8) | | Võrrand (16) | |
|-----|----------|---------------------|----------------|---------------------|----------------|
| | | δ_α (%) | δ_k (%) | δ_α (%) | δ_k (%) |
| 2 | 0,01 | 0,06 | 0,07 | 0,06 | 0,16 |
| 2 | 0,05 | 0,28 | 0,37 | 0,30 | 0,79 |
| 2 | 0,10 | 0,56 | 0,75 | 0,61 | 1,58 |
| 3 | 0,01 | 0,07 | 0,10 | 0,08 | 0,21 |
| 3 | 0,05 | 0,34 | 0,48 | 0,39 | 1,04 |
| 3 | 0,10 | 0,69 | 0,97 | 0,79 | 2,10 |

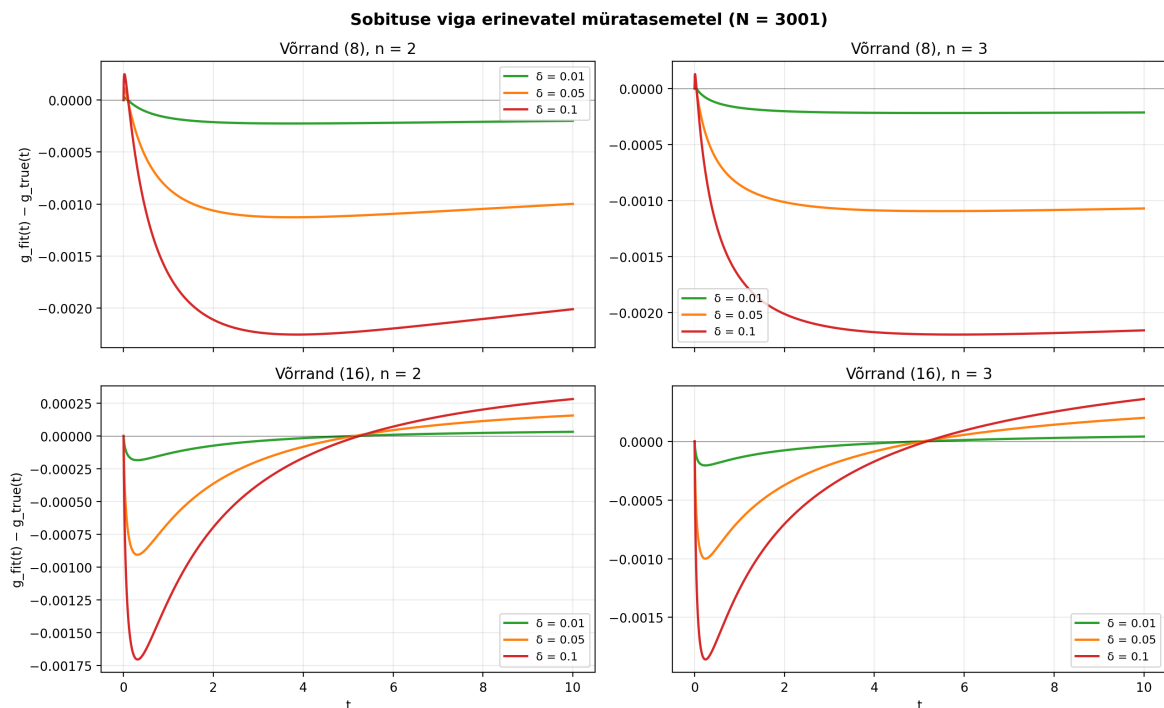
Tabelist 3 järeldub mitu tähelepanuväärset asja. Esiteks, väikse müra korral ($\delta = 0,01$) annavad mõlemad mudelid parameetrite hinnangud, mille keskmine suhteline viga

on alla 0,3%. Seega on mõlema mudeli pöördülesanne hästi lahendatav, kui andmete kvaliteet on kõrge.

Teiseks, mürataseme kasvades ilmneb, et fraktaalse mudeli parameetrite hinnangud on keskmiselt veidi täpsemad kui murrulise mudeli omad. Seda tulemust ei tuleks tõlgendada murrulise mudeli nõrkusena, sest võrdluses erinevad nii päripidiülesande lahendamise viis kui ka valitud parameetritežiimid. Erinevuse peamine põhjus seisneb tõenäoliselt selles, et fraktaalse mudeli päripidiülesanne lahendatakse analüütiliselt (valemid (3.2.3) ja (3.2.4)), mistõttu parameetrite hindamisvead tulenevad ainult andmete müra. Murrulise mudeli korral lisandub andmemürale ka L1-skeemi diskretisatsiooniviga, mis summaarselt suurendab parameetrite hinnangute ebatäpsust.

Kolmandaks, parameetri k_n vead on murrulise mudeli korral süstemaatiliselt suuremad kui parameetri α vead. See viitab sellele, et kineetilise konstandi hindamine on murrulise mudeli puhul tundlikum kui murrulise tuletise järgu hindamine.

Neid erinevusi illustreerib joonis 3, kus on kujutatud sobituse viga $g_{\text{fit}}(t) - g_{\text{true}}(t)$ funktsioonina ajast t mõlema mudeli ja mõlema järgu korral. Tuleb rõhutada, et joonisel ei ole kujutatud mõõteandmete jääke, vaid taastatud mudelikõvera hälvet tegelikust müravabast lahendist.



Joonis 3: Sobituse viga $g_{\text{fit}}(t) - g_{\text{true}}(t)$ erinevatel müratasemetel. Ülemine rida: võrrand (8); alumine rida: võrrand (16). Vasakul: $n = 2$; paremal: $n = 3$.

5.4 Mudeli järgu n tuvastamine

Eelnevates peatükkides eeldati, et mudeli järk n on ette teada. Praktikas ei pruugi see aga nii olla. Adsorptsiooniprotsessi järk tuleb sageli samuti andmete põhjal määrata. Käesolevas alampeatükis uuritakse, kas n on võimalik tuvastada mudelivaliku lähenemisega (4.5.1), kus pöördülesanne lahendatakse eraldi iga kandidaatväärtuse $n \in \{1, 2, 3, 4, 5\}$ korral ning valitakse see n , mille juures sihtfunktsiooni optimaalne väärtus on väikseim ehk lahendatakse täisarvuline optimeerimisülesanne.

Sarnaselt parameetrite hindamisega jaotistes 5.1–5.2 korraliti mudelivalikut mitme müra realisatsiooni üle, et hinnata tuvastamise usaldusväärsust statistiliselt. Fraktaalse mudeli korral kasutati 20 realisatsiooni iga $(n_{\text{true}}, \delta)$ kombinatsiooni kohta; murrulise mudeli korral 10 realisatsiooni.

Fraktaalne mudel

Fraktaalse mudeli korral kasutati samu tegelikke parameetreid nagu jaotises 5.1. Tabelis 4 on esitatud õige järgu tuvastamise edukuse määr 20 müra realisatsiooni üle.

Tabel 4: Fraktaalse mudeli järgu n tuvastamise edukuse määr üle 20 realisatsiooni.

| n_{true} | δ | Edukuse määr |
|-------------------|----------|--------------|
| 1 | 0,01 | 20/20 (100%) |
| 1 | 0,05 | 20/20 (100%) |
| 1 | 0,10 | 20/20 (100%) |
| 2 | 0,01 | 20/20 (100%) |
| 2 | 0,05 | 20/20 (100%) |
| 2 | 0,10 | 20/20 (100%) |
| 3 | 0,01 | 20/20 (100%) |
| 3 | 0,05 | 20/20 (100%) |
| 3 | 0,10 | 20/20 (100%) |

Kõigis üheksas konfiguratsioonis tuvastati õige järk n iga realisatsiooni korral. See näitab, et fraktaalse mudeli korral on mudelivalik väga usaldusväärne isegi suurima testitud müra korral ($\delta = 0,10$).

Murruline mudel

Murrulise mudeli korral kasutati parameetreid $\alpha_{\text{true}} = 0,8$, $k_{\text{true}} = 1,0$, $g_e = 1,0$. Tabelis 5 on esitatud edukuse määr üle 10 realisatsiooni.

Tabel 5: Murrulise mudeli järgu n tuvastamise edukuse määr üle 10 müra realisatsiooni.

| n_{true} | δ | Edukuse määr |
|-------------------|----------|--------------|
| 2 | 0,01 | 10/10 (100%) |
| 2 | 0,05 | 10/10 (100%) |
| 2 | 0,10 | 9/10 (90%) |
| 3 | 0,01 | 10/10 (100%) |
| 3 | 0,05 | 10/10 (100%) |
| 3 | 0,10 | 6/10 (60%) |

Väikese müra korral ($\delta \leq 0,05$) tuvastati õige järk kõigil kordadel. Suurima müra korral ($\delta = 0,10$) langes edukuse määr märgatavalt: juhul $n_{\text{true}} = 2$ ebaõnnestus tuvastamine 10% juhtudest ning juhul $n_{\text{true}} = 3$ koguni 40% juhtudest. Need ebaõnnestumised on tingitud sellest, et suurel müratasel muutub sihtfunktsiooni maastik naaberjärkude vahel nii lamedaks, et juhuslikud müra realisatsioonid võivad kallutada miinimumi vale kandidaadi kasuks. Mõlemal juhul valiti ebaõnnestumise korral naaberjärk: $n^* = 3$ tegeliku $n = 2$ asemel ja $n^* = 2$ tegeliku $n = 3$ asemel.

6 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli uurida fraktaalse ja murrulise adsorptsiooni-mudeli pöördülesannete lahendamist sünteetiliste andmete põhjal. Mõlema mudeli korral genereeriti teadaolevate parameetritega mudelilahend, lisati kontrollitud müra ning taastati parameetrid vähimruutude meetodil. Katsete usaldusväärsuse tagamiseks kasutati mõlema mudeli jaoks sama andmepunktide arvu ($N = 3001$) ning tulemusi valideeriti üle mitme müra realisatsiooni.

Fraktaalse mudeli korral testiti järke $n = 1, 2, 3$ ning kolme mürataset. Kõigi üheksa kombinatsiooni puhul koondus pöördülesanne edukalt; väikse müra korral jäid keskmised suhtelised vead alla 0,1%.

Murrulise mudeli korral testiti järke $n = 2, 3$ samade müratasete juures. Ka siin koondusid kõik kuus kombinatsiooni edukalt iga realisatsiooni korral. Keskmised suhtelised vead olid aga mõnevõrra suuremad kui fraktaalse mudeli korral. See erinevus on peamiselt tingitud L1-skeemi diskretisatsiooniveast, mida fraktaalse mudeli analüütiline lahend ei sisalda.

Lisaks uuriti olukorda, kus ka mudeli järk n on tundmatu. Mudelivaliku lähenemine tuvastas fraktaalse mudeli korral õige n igas realisatsioonis(100%) ning murrulise mudeli korral väikese müra juures samuti 100%-liselt, kuid suurima müra korral langes edukuse määr kuni 60%-ni.

Töö peamiseks järeldusteks on esiteks see, et mõlema vaadeldud mudeli parameetrid on kontrollitud sünteetilistes tingimustes edukalt taastatavad. Teiseks see, et testitud kombinatsioonides ei põhjustanud mudeli järku n suurenemine iseenesest pöördülesande ebastabiilsust. Kolmandaks, et mudeli järku n oli võimalik enamasti tuvastada lihtsa jääksumma põhise optimeerimisülesande lahendamise abil, kuid murrulise mudeli korral muutus see ülesanne suure müratasete juures ebakindlaks.

Tulemusi tuleb siiski käsitleda kontseptuaalse ja arvutusliku valideerimisena, mitte lõpliku järeldusena adsorptsioonikineetika kohta. Peamised piirangud on fikseeritud tasakaalu adsorptsioonimaht g_e , sünteetilised andmed, tõkestatud müra, lokaalne optimeerimine ning asjaolu, et sünteetilised andmed pärinevad samast mudeliklassist, mida hiljem sobitati. Edasistes uuringutes oleks otstarbekas testida meetodit reaalsel adsorptsiooniandmetel, hinnata parameetrit g_e koos teiste parameetritega, kasutada sõltumatut või kõrgema järku päripidiülesande lahendajat ning lisada parameetrite usaldusvahemikud.

Tänuavaldused

Täna juhendajat professor Jaan Jannot teema väljapakkumise ja töö valmimise jooksul antud nõuannete eest.

Viited

- [1] Richard C. Aster, Brian Borchers, and Clifford H. Thurber. *Parameter Estimation and Inverse Problems*. Elsevier, Amsterdam, 3 edition, 2018. ISBN 9780128046517.
- [2] Evangelos Bakalis and Francesco Zerbetto. Adsorption kinetics: Classical, fractal, or fractional? *Langmuir*, 41:19834–19844, 2025. doi: 10.1021/acs.langmuir.5c01726.
- [3] Mary Ann Branch, Thomas F. Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999. doi: 10.1137/S1064827595289108.
- [4] Kenneth P. Burnham and David R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, New York, 2 edition, 2002. doi: 10.1007/b97636.
- [5] Kai Diethelm. *The Analysis of Fractional Differential Equations*, volume 2004 of *Lecture Notes in Mathematics*. Springer, Berlin and Heidelberg, 2010. doi: 10.1007/978-3-642-14574-2.
- [6] Rudolf Gorenflo, Anatoly A. Kilbas, Francesco Mainardi, and Sergei V. Rogosin. *Mittag-Leffler Functions, Related Topics and Applications*. Springer, Berlin and Heidelberg, 2 edition, 2020. doi: 10.1007/978-3-030-57793-1.
- [7] Charles W. Groetsch. *Inverse Problems in the Mathematical Sciences*. Vieweg+Teubner Verlag, Wiesbaden, 1993. doi: 10.1007/978-3-322-99202-4.
- [8] Per Christian Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, Philadelphia, 2010. doi: 10.1137/1.9780898718836.
- [9] Rudolf Hilfer. *Applications of Fractional Calculus in Physics*. World Scientific, Singapore, 2000. doi: 10.1142/3779.
- [10] Yuh-Shan Ho and Gordon McKay. Pseudo-second order model for sorption processes. *Process Biochemistry*, 34(5):451–465, 1999. doi: 10.1016/S0032-9592(98)00112-5.
- [11] Bangti Jin, Raytcho Lazarov, and Zhi Zhou. Two fully discrete schemes for fractional diffusion and diffusion-wave equations with nonsmooth data. *SIAM Journal on Scientific Computing*, 38(1):A146–A170, 2016. doi: 10.1137/140979563.

-
- [12] S. Lagergren. Zur theorie der sogenannten adsorption gelöster stoffe. *Kungliga Svenska Vetenskapsakademiens Handlingar*, 24(4):1–39, 1898.
- [13] Yumin Lin and Chuanju Xu. Finite difference/spectral approximations for the time-fractional diffusion equation. *Journal of Computational Physics*, 225(2): 1533–1552, 2007. doi: 10.1016/j.jcp.2007.02.003.
- [14] Ralf Metzler and Joseph Klafter. The random walk’s guide to anomalous diffusion: a fractional dynamics approach. *Physics Reports*, 339(1):1–77, 2000. doi: 10.1016/S0370-1573(00)00070-3.
- [15] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5.
- [16] Igor Podlubny. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, volume 198 of *Mathematics in Science and Engineering*. Academic Press, San Diego, 1999. ISBN 9780125588409.
- [17] Joel L. Schiff. *The Laplace Transform: Theory and Applications*. Springer, New York, 1999. doi: 10.1007/978-0-387-22757-3.
- [18] Martin Stynes, Eugene O’Riordan, and José Luis Gracia. Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. *SIAM Journal on Numerical Analysis*, 55(2):1057–1079, 2017. doi: 10.1137/16M1082329.
- [19] Zhi-Zhong Sun and Xu-Ning Wu. A fully discrete difference scheme for a diffusion-wave system. *Applied Numerical Mathematics*, 56(2):193–209, 2006. doi: 10.1016/j.apnum.2005.03.003.
- [20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Annotatsioon

Töö eesmärk on uurida fraktaalse ja murrulise adsorptsioonimudeli pöördülesannete lahendamist sünteetiliste andmete põhjal. Fraktaalse mudeli korral kasutatakse analüütilist päripidiülesande lahendit; murrulise mudeli korral lahendatakse päripidiülesanne numbriliselt Caputo tuletise L1-skeemi abil. Pöördülesanne lahendatakse mõlema võrrandi vähimruutude meetodil. Arvutuskatsetes testitakse erinevaid mudelijärke ($n = 1, 2, 3$ fraktaalse ning $n = 2, 3$ murrulise mudeli korral) ja müratasemeid ($\delta = 0,01; 0,05; 0,10$). Tulemused näitavad, et mõlema mudeli parameetrid on edukalt taastatavad: väikese müra korral jäävad suhtelised vead alla 0,3%. Mudelite võrdlus samadel võrgutingimustel viitab sellele, et parameetrite taastamise täpsust mõjutab oluliselt ka päripidiülesande lahendamise viis: fraktaalse mudeli korral kasutatakse analüütilist lahendit, murrulise mudeli korral aga L1-skeemi. Seetõttu ei saa täheldatud erinevusi tõlgendada üksnes mudelite matemaatilise struktuuri erinevusena. Lisaks näitavad katsed, et mudeli järk n on enamikul juhtudel tuvastatav täisarvulise optimeerimisülesandena.

Märksõnad: pöördülesanne, adsorptsioon, murruline tuletis, Caputo tuletis, L1-skeem, vähimruutude meetod.

Abstract

The aim of this bachelor's thesis is to investigate inverse problems for fractal and fractional adsorption models using synthetic data. For the fractal model, the forward problem is solved analytically; for the fractional model, the forward problem is solved numerically using the L1 discretization of the Caputo derivative. In both cases, the inverse problem is solved by the least-squares method. Computational experiments are carried out for different model orders ($n = 1, 2, 3$ for the fractal and $n = 2, 3$ for the fractional model) and noise levels ($\delta = 0.01, 0.05, 0.10$). The results show that model parameters can be successfully recovered for both models: at low noise levels, relative errors remain below 0.3%. A controlled comparison using equal grid sizes suggests that the accuracy of parameter recovery is substantially affected by the forward-solver choice: the fractal model uses an analytic solution, whereas the fractional model relies on the L1 discretization. Therefore, the observed differences cannot be attributed solely to the mathematical structure of the models. The experiments also indicate that the model order n can in most cases be identified successfully by a model-selection approach.

Keywords: inverse problem, adsorption, fractional derivative, Caputo derivative, L1 scheme, least-squares method.

Lisad

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Mattias Arakas,

1. annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Pöördülesanded murrulistele reaktsioonikineetika võrranditele“, mille juhendaja on professor Jaan Janno,
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile;
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

20. mai 2026. a.

Lisa 2 – Pythoni skriptid

Käesolevas lisas on esitatud töös kasutatud Pythoni skriptid. Skriptid eeldavad järgmisi pakette: `numpy`, `scipy` ja `matplotlib`. Failinimed on säilitatud sellistena, nagu nad on `import`-lausetes — `runexperiments.py` ja `poordylvorrand16.py` sõltuvad teistest moodulitest nimepidi.

`poordylvorrand8.py` – võrrandi (8) analüütiline päripidilahend ning parameetrite (a, k_n) hindamine vähimruutude meetodil.

```
1 """
2 Inverse fitting for Eq. (8):  dg/dt = a * k_n * t^(a-1) * (g_e - g)^n.
3
4 Forward model uses the analytic solution; the parameters (a, k_n) are
5 estimated from observed data by nonlinear least squares, with g_e
6   fixed.
7 """
8 import numpy as np
9 from dataclasses import dataclass
10 from scipy.optimize import least_squares
11
12 @dataclass
13 class FitResult:
14     a_hat: float
15     k_hat: float
16     success: bool
17     cost: float
18
19
20 def forward_eq8_analytic(t, a, k_n, n, g_e):
21     """Analytic solution of Eq. (8) with g(0) = 0."""
22     t = np.asarray(t, float)
23     if n == 1:
24         return g_e * (1.0 - np.exp(-k_n * np.power(t, a)))
25     inside = 1.0 + (1.0 - n) * (g_e ** (n - 1)) * k_n * np.power(t, a)
26     inside = np.maximum(inside, 1e-15)
27     return g_e * (1.0 - np.power(inside, 1.0 / (1.0 - n)))
28
29
30 def make_synthetic_data_uniform_noise(g_true, delta, *, clip_to=None,
31 seed=0):
32     """Add uniform noise U(-delta, delta); optionally clip to [lo,
33     hi]."""
34     rng = np.random.default_rng(seed)
35     g_noisy = g_true + rng.uniform(-delta, delta, size=g_true.shape)
36     if clip_to is not None:
```

```

35     g_noisy = np.clip(g_noisy, clip_to[0], clip_to[1])
36     return g_noisy
37
38
39 def fit_eq8_inverse(t, g_obs, n, g_e=1.0, a0=0.8, k0=1.0):
40     """Estimate (a, k_n) by nonlinear least squares."""
41     t = np.asarray(t, float)
42     g_obs = np.asarray(g_obs, float)
43
44     def residuals(x):
45         return forward_eq8_analytic(t, a=x[0], k_n=x[1], n=n,
46             g_e=g_e) - g_obs
47
48     res = least_squares(residuals, x0=[a0, k0],
49         bounds=([1e-6, 1e-12], [1.0, 1e6]),
50         method="trf")
51     return FitResult(float(res.x[0]), float(res.x[1]),
52         bool(res.success), float(res.cost))

```

vorrand16lahendamine_NUMBRILINE.py – võrrandi (16) numbriline päripidilähendaja implitsitse L1-skeemi abil.

```

1  """
2  Implicit L1 time-stepping solver for the fractional kinetic equation
3  (Eq. 16):
4       $C D_0^a g(t) = k_n * (g_e - g(t))^n, \quad 0 < a \leq 1, \quad n \geq 1.$ 
5  At each step the scalar nonlinear equation for  $g_{\{m+1\}}$  is solved by
6  bisection
7  within the physical interval  $[0, g_e]$ .
8  """
9  import numpy as np
10 from scipy.special import gamma
11
12 def _check_uniform_grid(t):
13     dt = np.diff(t)
14     h = float(dt[0])
15     if not np.allclose(dt, h, rtol=1e-10, atol=1e-12):
16         raise ValueError("Uniform time grid required.")
17     return h
18
19
20 def _L1_weights(a, m):
21     """ $b_j = (j+1)^{(1-a)} - j^{(1-a)}$  for  $j = 0..m$ ."""
22     j = np.arange(0, m + 1, dtype=float)
23     return np.power(j + 1.0, 1.0 - a) - np.power(j, 1.0 - a)

```

```

24
25
26 def solve_eq16_L1_implicit(t, a, k_n, n, g_e, g0=0.0,
27                             bisection_maxiter=200,
28                             bisection_tol=1e-12):
29
30     """
31     Solve Eq. (16) on a uniform grid using the implicit L1 scheme.
32
33     Step equation at t_{m+1}:
34     A * (x - g_m + H_m) - k_n * (g_e - x)^n = 0,
35     where A = 1 / (Gamma(2-a) * h^a) and H_m is the L1 history term.
36     """
37     t = np.asarray(t, dtype=float)
38     h = _check_uniform_grid(t)
39     if not (0.0 < a <= 1.0):
40         raise ValueError("Require 0 < a <= 1.")
41     if k_n <= 0.0 or g_e <= 0.0:
42         raise ValueError("Require k_n > 0 and g_e > 0.")
43
44     A = 1.0 / (gamma(2.0 - a) * (h ** a))
45     N = t.size - 1
46     g = np.empty(t.size, dtype=float)
47     g[0] = float(g0)
48     b = _L1_weights(a, max(N - 1, 0))
49
50     for m in range(N):
51         # History term H_m = sum_{j=1}^m b_j * (g_{m+1-j} - g_{m-j})
52         H_m = 0.0
53         if m >= 1:
54             j = np.arange(1, m + 1)
55             H_m = float(np.dot(b[1:m + 1], g[(m + 1) - j] - g[m - j]))
56         g_m = float(g[m])
57
58         def F(x):
59             return A * ((x - g_m) + H_m) - k_n * (g_e - x) ** n
60
61         lo, hi = 0.0, float(g_e)
62         f_lo, f_hi = F(lo), F(hi)
63         if f_lo * f_hi > 0.0:
64             g[m + 1] = lo if abs(f_lo) < abs(f_hi) else hi
65             continue
66
67         for _ in range(bisection_maxiter):
68             mid = 0.5 * (lo + hi)
69             f_mid = F(mid)
70             if abs(f_mid) <= bisection_tol or (hi - lo) <=
71                 bisection_tol:
72                 lo = hi = mid

```

```

70         break
71         if f_lo * f_mid <= 0.0:
72             hi, f_hi = mid, f_mid
73         else:
74             lo, f_lo = mid, f_mid
75         g[m + 1] = float(np.clip(0.5 * (lo + hi), 0.0, g_e))
76
77     return g

```

poordylvorrاند16.py – võrrandi (16) parameetrite (α, k_n) hindamine, kasutades L1-lahendajat päripidimudelina.

```

1  """
2  Inverse fitting for Eq. (16):  $C D^a g(t) = k_n * (g_e - g)^n$ .
3
4  Forward model is the implicit L1 scheme (see
5  vorrand16lahendamaine_NUMBRILINE).
6  The parameters (a, k_n) are estimated from observed data by nonlinear
7  least
8  squares, with g_e fixed.
9  """
10 import numpy as np
11 from dataclasses import dataclass
12 from scipy.optimize import least_squares
13
14
15 @dataclass
16 class FitResult:
17     a_hat: float
18     k_hat: float
19     success: bool
20     cost: float
21
22
23 def forward_eq16(t, a, k_n, n, g_e, g0=0.0):
24     """Forward model via the implicit L1 scheme."""
25     return solve_eq16_L1_implicit(t, a=a, k_n=k_n, n=n, g_e=g_e,
26                                   g0=g0)
27
28 def make_synthetic_data_uniform_noise(g_true, delta, *, clip_to=None,
29                                       seed=0):
30     """Add uniform noise U(-delta, delta); optionally clip to [lo,
31     hi]."""
32     rng = np.random.default_rng(seed)

```

```

31     g_noisy = g_true + rng.uniform(-delta, delta, size=g_true.shape)
32     if clip_to is not None:
33         g_noisy = np.clip(g_noisy, clip_to[0], clip_to[1])
34     return g_noisy
35
36
37 def fit_eq16_inverse(t, g_obs, n, g_e=1.0, g0=0.0, a0=0.8, k0=1.0):
38     """Estimate (a, k_n) by nonlinear least squares."""
39     t = np.asarray(t, float)
40     g_obs = np.asarray(g_obs, float)
41     if t.shape != g_obs.shape:
42         raise ValueError("t and g_obs must have the same shape.")
43
44     def residuals(x):
45         return forward_eq16(t, a=x[0], k_n=x[1], n=n, g_e=g_e, g0=g0)
46             - g_obs
47
48     res = least_squares(residuals, x0=[a0, k0],
49                        bounds=([1e-6, 1e-12], [1.0, 1e6]),
50                        method="trf", ftol=1e-10, xtol=1e-10,
51                        gtol=1e-10)
52     return FitResult(float(res.x[0]), float(res.x[1]),
53                    bool(res.success), float(res.cost))

```

runexperiments.py – numbriliste katsete läbiviija. Genereerib mõlema mudeli jaoks päripidilahendi, lisab müra ning sobitab parameetrid tagasi; tulemused (kokkuvõte + joonised) salvestatakse kausta results/.

```

1     """
2     Run forward problem -> add noise -> inverse fit, for both Eq. (8)
3     (analytic
4     forward) and Eq. (16) (L1 implicit forward). For each setting (n,
5     delta) a
6     text summary and a PNG plot are saved into ./results/.
7     """
8
9     import os
10    from dataclasses import asdict
11    from datetime import datetime
12
13    import numpy as np
14    import matplotlib.pyplot as plt
15
16    from poordylvorrand8 import (
17        forward_eq8_analytic, fit_eq8_inverse,
18        make_synthetic_data_uniform_noise as noise_eq8,
19    )
20    from poordylvorrand16 import (

```

```

18     forward_eq16, fit_eq16_inverse,
19     make_synthetic_data_uniform_noise as noise_eq16,
20 )
21
22
23 def ensure_results_dir(path="results"):
24     os.makedirs(path, exist_ok=True)
25     return path
26
27
28 def summarize_fit(title, true_params, fit_obj):
29     lines = [title, "-" * len(title), "True parameters:"]
30     for k, v in true_params.items():
31         lines.append(f" {k} = {v}")
32     lines.append("\nFit result:")
33     for k, v in asdict(fit_obj).items():
34         lines.append(f" {k} = {v}")
35     return "\n".join(lines)
36
37
38 def plot_and_save(out_png, t, g_obs, g_true, g_fit, title):
39     plt.figure(figsize=(9, 5))
40     plt.plot(t, g_obs, label="synthetic noisy data", linewidth=1)
41     plt.plot(t, g_true, label="true model", linewidth=2)
42     plt.plot(t, g_fit, label="fit model", linewidth=2)
43     plt.xlabel("t"); plt.ylabel("g(t)"); plt.title(title)
44     plt.grid(True); plt.legend(); plt.tight_layout()
45     plt.savefig(out_png, dpi=200)
46     plt.close()
47
48
49 def experiment_eq8(results_dir, n, delta, seed):
50     a_true, k_true, g_e = 0.7, 0.9, 1.0
51     t = np.linspace(0.0, 10.0, 400)
52
53     g_true = forward_eq8_analytic(t, a=a_true, k_n=k_true, n=n,
54                                   g_e=g_e)
55     g_obs = noise_eq8(g_true, delta=delta, clip_to=(0.0, g_e),
56                      seed=seed)
57     fit = fit_eq8_inverse(t, g_obs, n=n, g_e=g_e)
58     g_fit = forward_eq8_analytic(t, a=fit.a_hat, k_n=fit.k_hat, n=n,
59                                  g_e=g_e)
60
61     stamp = datetime.now().strftime("%Y%m%d_%H%M%S")
62     base = os.path.join(results_dir,
63                          f"eq8_n{n}_delta{str(delta).replace('.', 'p')}_{{stamp}}")
64
65     with open(base + ".txt", "w", encoding="utf-8") as f:

```

```

63     f.write(summarize_fit(
64         f"Experiment: Eq(8), n={n}, delta={delta}",
65         {"a_true": a_true, "k_true": k_true, "n": n, "g_e": g_e,
66         "delta": delta, "seed": seed},
67         fit))
68     plot_and_save(base + ".png", t, g_obs, g_true, g_fit,
69                 f"Eq(8), n={n}, delta={delta}")
70
71
72     def experiment_eq16(results_dir, n, delta, seed, N=3001):
73         a_true, k_true, g_e, g0 = 0.8, 1.0, 1.0, 0.0
74         t = np.linspace(0.0, 10.0, N)
75
76         g_true = forward_eq16(t, a=a_true, k_n=k_true, n=n, g_e=g_e,
77                               g0=g0)
78         g_obs = noise_eq16(g_true, delta=delta, clip_to=(0.0, g_e),
79                             seed=seed)
80         fit = fit_eq16_inverse(t, g_obs, n=n, g_e=g_e, g0=g0)
81         g_fit = forward_eq16(t, a=fit.a_hat, k_n=fit.k_hat, n=n, g_e=g_e,
82                               g0=g0)
83
84         stamp = datetime.now().strftime("%Y%m%d_%H%M%S")
85         base = os.path.join(results_dir,
86                             f"eq16_n{n}_delta{str(delta).replace('.', 'p')}_
87                             {stamp}")
88
89         with open(base + ".txt", "w", encoding="utf-8") as f:
90             f.write(summarize_fit(
91                 f"Experiment: Eq(16), n={n}, delta={delta}",
92                 {"a_true": a_true, "k_true": k_true, "n": n, "g_e": g_e,
93                 "g0": g0, "delta": delta, "seed": seed, "N_grid": N},
94                 fit))
95             plot_and_save(base + ".png", t, g_obs, g_true, g_fit,
96                             f"Eq(16), n={n}, delta={delta}")
97
98
99     def main():
100         results_dir = ensure_results_dir("results")
101         deltas = [0.01, 0.05, 0.10]
102
103         for i, n in enumerate([1, 2, 3]):
104             for j, delta in enumerate(deltas):
105                 experiment_eq8(results_dir, n=n, delta=delta,
106                               seed=1 + 10 * i + j)
107
108         for i, n in enumerate([2, 3]):
109             for j, delta in enumerate(deltas):
110                 experiment_eq16(results_dir, n=n, delta=delta,
111                                 seed=100 + 10 * i + j)

```

```
108  
109  
110 if __name__ == "__main__":  
111     main()
```