

TARTU ÜLIKOOL
LOODUS- JA TÄPPISTEADUSTE VALDKOND
MATEMAATIKA JA STATISTIKA INSTITUUT

Tobias Toomas Peterson
Interpoleerivad kuupsplainid
Matemaatika
Bakalaureusetöö (9 EAP)

Juhendaja: prof. Arvet Pedas

TARTU 2026

INTERPOLEERIVAD KUUPSPLAINID

Bakalaureusetöö

Tobias Toomas Peterson

Lühikokkuvõte

Bakalaureusetöös käsitletakse antud funktsiooni lähendamist interpoleerivate kuupsplainide abil. Töös tuletatakse interpoleeriva kuupsplaini esitus kolme erinevat tüüpi rajatingimuste komplekti korral. Töös saadud tulemusi illustreeritakse kahe näite abil.

CERCS teaduseriala: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria).

Märksõnad: kuupsplain, interpoleerimine, arvutusmatemaatika.

INTERPOLATING CUBIC SPLINES

Tobias Toomas Peterson

Abstract

This bachelor's thesis examines the approximation of a given function using interpolating cubic splines. The thesis derives the representation of an interpolating cubic spline for three different types of boundary conditions. The obtained results are illustrated with two examples.

CERCS research specialisation: P170 Computer science, numerical analysis, systems, control

Key Words: cubic spline, interpolation, computational mathematics.

Sisukord

Sissejuhatus	3
1 Abitulemused	5
2 Kuupsplaini mõiste	7
3 Interpoleeriva kuupsplaini konstrueerimine	10
4 Pidevustingimused	12
5 Interpoleeriva kuupsplaini olemasolu ja ühesus	16
6 Interpolatsiooniviga ja näited	23
6.1 Näited	23
Kokkuvõte	33
Kasutatud allikad	34
Lisa 1. Kood	35

Sissejuhatus

Funktsioonide lähendamine on arvutusmatemaatikas oluline ülesanne. Paratamatu vajadus funktsioonide lähendamiseks tekib näiteks tabelina antud funktsioonidega opereerimisel. Sel korral arvutatakse funktsiooni väärtused tabelis mitteesinevatel kohtadel teda lähendava funktsiooni väärtustena. Tihhti kasutatakse lõplikul lõigul $[a, b]$ määratud funktsiooni $y = f(x)$ lähendamiseks polünoome, näiteks astmefunktsioone $1, x, x^2, \dots$. Lühikese lõigu korral õnnestub sel teel funktsioonil f leida häid lähendeid juba üsna madala astme polünoomidena, pika lõigu korral ei pruugi sellisel viisil saada rahuldavaid tulemusi [1].

Seetõttu jaotatakse lõik $[a, b]$ sageli väiksemateks osalõikudeks punktide $a = x_0 < x_1 < \dots < x_n = b$ abil ja seejärel lähendatakse funktsiooni f igal osalõigul $[x_{i-1}, x_i]$ ($i = 1, \dots, n$) mingi polünoomiga. Nii moodustub teatav tükiti polünoomiaalne lähend, mis võib üldiselt jaotuspunktides $x_1, \dots, x_{n-1}$ olla katkev. Selliseid lähendeid nimetatakse splainideks [2].

Käesolevas töös käsitletakse mõningaid küsimusi, mis on seotud funktsiooni f lähendamisega kuupsplainiga. Täpsemalt vaadeldakse sellise splaini $S_3(x)$ konstrueerimist, mis igal osalõigul $[x_{i-1}, x_i]$ ($i = 1, \dots, n$) on kuuppolünoom ning mis kogu lõigul $[a, b]$ on kaks korda pidevalt diferentseeruv. Veel nõutakse, et $S_3(x)$ rahuldab interpolatsioonitingimusi $S_3(x_i) = f_i$, $i = 0, \dots, n$, kus $f_0, \dots, f_n$ on etteantud arvud. Selliseid splaine nimetatakse interpoleerivateks kuupsplainideks.

Bakalaureusetöös leitakse interpoleeriva kuupsplaini $S_3(x)$ esitus splaini esimese tuletise väärtuste $m_i := S'_3(x_i)$ ($i = 0, \dots, n$) kaudu ja kolme tüüpi lisatingimuste (mida nimetatakse rajatingimusteks) korral. Töös näidatakse, et suurused $m_0, \dots, m_n$ ja seega ka interpoleeriva splaini $S_3(x)$ avaldis on

kõigi kolme rajatingimuste komplekti juhul üheselt määratud.

Töö viimases osas on toodud raamatus [3] saadud tulemus lõigul $[a, b]$ neli korda pidevalt diferentseeruva funktsiooni f väärtusi $f_i = f(x_i)$ ($i = 0, \dots, n$) interpoleeriva kuupsplaini S_3 vea kohta ning esitatud kaks originaalset näidet (koos kommentaaridega) töös saadud tulemuste illustreerimiseks.

1 Abitulemused

Olgu $\mathbb{R} = (-\infty, \infty)$ reaalarvude hulk ja $\mathbb{N} = \{1, 2, \dots\}$ naturaalarvude hulk.

Definitsioon 1. Olgu $n \in \mathbb{N}$, $n \geq 2$, ja olgu

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = (a_{ij})_{i,j=1}^n,$$

kus $a_{ij} \in \mathbb{R}$. Öeldakse, et maatriksi A peadiagonaal domineerib ridade kaupa, kui

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}|, \quad i = 1, \dots, n.$$

See tähendab, et maatriksi A igas reas on peadiagonaali elemendi absoluutväärtus suurem sama rea ülejäänud elementide absoluutväärtuste summast.

Lemma 1. Olgu $n \in \mathbb{N}$, $n \geq 2$, ja olgu $A = (a_{ij})_{i,j=1}^n$ ruutmaatriks. Kui maatriksi A peadiagonaal domineerib ridade kaupa, siis tema determinant:

$$\det A \neq 0.$$

Tõestus. Oletame vastuväiteliselt, et $\det A = 0$. Vaatleme võrrandisüsteemi

$$Ax = 0,$$

kus otsitavaks on vektor

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = (x_1, \dots, x_n)^T.$$

Et $\det A = 0$, siis leidub nullist erinev vektor $(x_1, \dots, x_n)^T$ nii, et $Ax = 0$.
Valime indeksi k nii, et

$$|x_k| = \max_{1 \leq i \leq n} |x_i| > 0.$$

Kirjutades $Ax = 0$ lahti kui diagonaali ja muude elementide summana, saame

$$a_{kk}x_k + \sum_{\substack{j=1 \\ j \neq k}}^n a_{kj}x_j = 0$$

ehk

$$a_{kk}x_k = - \sum_{\substack{j=1 \\ j \neq k}}^n a_{kj}x_j.$$

Sellest järeldub, et

$$|a_{kk}| |x_k| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| |x_j| \leq \left(\sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| \right) |x_k|.$$

Kuna $|x_k| > 0$, saame

$$|a_{kk}| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}|,$$

mis on vastuolus eeldusega, et peadiagonaal domineerib ridade kaupa. Seega $\det A \neq 0$. □

2 Kuupsplaini mõiste

Olgu $n \geq 3$ fikseeritud täisarv ning olgu lõigul $[a, b]$ antud võrk

$$\Delta_n = \{x_0, x_1, \dots, x_n : a = x_0 < x_1 < \dots < x_n = b\}, \quad (1)$$

mis määrab lõigu $[a, b]$ jaotuse osalõikudes $[x_{i-1}, x_i]$, $i = 1, \dots, n$. Olgu $C^k[a, b]$ lõigus $[a, b]$ k -korda ($k \geq 0$) pidevalt diferentseeruvate funktsioonide hulk. Seejuures $k = 0$ korral kirjutame $C^0[a, b] = C[a, b]$.

Definitsioon 2. Võrgule (1) vastavat funktsiooni $S_3 = S_{3, \Delta_n} : [a, b] \rightarrow \mathbb{R}$ nimetatakse kuupsplainiks (edaspidi ka splainiks), kui

1. igal osalõigul $[x_{i-1}, x_i]$ ($i = 1, \dots, n$) on S_3 ülimalt kuuppolünoom, s.t

$$S_3(x) = c_0^{(i)} + c_1^{(i)}(x - x_{i-1}) + c_2^{(i)}(x - x_{i-1})^2 + c_3^{(i)}(x - x_{i-1})^3 \quad (2)$$

kus $x \in [x_{i-1}, x_i]$ ja $c_0^{(i)}, c_1^{(i)}, c_2^{(i)}, c_3^{(i)} \in \mathbb{R}$;

2. $S_3 \in C^2[a, b]$.

Võrgu Δ_n punkte $x_0, \dots, x_n$ nimetatakse splaini S_3 sõlmedeks. Definitsiooni 2 esimesest tingimusest järeldub, et kuupsplaini määravad kordajad

$$c_j^{(i)}, \quad i = 1, \dots, n, \quad j = 0, 1, 2, 3, \quad (3)$$

kokku $4n$ parameetrit. Kuna igal osalõigul $[x_{i-1}, x_i]$ ($i = 1, \dots, n$) on S_3 kui kuuppolünoom lõpmata palju kordi diferentseeruv, siis tingimus $S_3 \in C^2[a, b]$ tähendab, et funktsioonid S_3 , S'_3 ja S''_3 on lõigul $[a, b]$ pidevad, mis seab igas sisesõlmes $x_1, \dots, x_{n-1}$ splainile 3 tingimust, mis kitsendavad parameetreid (3) valikut:

- $n - 1$ kitsendust splaini S_3 pidevuse jaoks ($S_3 \in C[a, b]$),
- $n - 1$ kitsendust S_3 esimese tuletise pidevuse jaoks ($S'_3 \in C[a, b]$),
- $n - 1$ kitsendust S_3 teise tuletise pidevuse jaoks ($S''_3 \in C[a, b]$).

Kokkuvõttes tekib $3(n - 1)$ tingimust, mis kitsendavad parameetrite (3) väärtusi. Nii sisaldab kuupsplain üldiselt

$$4n - 3(n - 1) = n + 3$$

vaba parameetrit. Nende parameetrite määramiseks seatakse sageli tingimused, et splain interpoleeriks sõlmedes $x_0, x_1, \dots, x_n$ etteantud väärtuseid

$$f_0, f_1, \dots, f_n \in \mathbb{R}.$$

Arvud $f_0, f_1, \dots, f_n$ võivad olla näiteks mingi funktsiooni $f = f(x)$ väärtused punktides $x = x_i, i = 0, 1, \dots, n$.

Niisiis splaini S_3 määramiseks nõuamegi, et oleksid täidetud interpolatsioonitingimused

$$S_3(x_i) = f_i, \quad i = 0, 1, \dots, n, \quad (4)$$

kus $f_0, f_1, \dots, f_n$ on etteantud arvud. Interpolatsioonitingimusi (4) rahuldavad splainid S_3 nimetatakse interpoleerivateks kuupsplainideks. Kuna kuupsplaini S_3 määravate parameetrite arv on $n + 3$, tuleb interpoleeriva kuupsplaini S_3 üheseks leidmiseks lisada tingimustele (4) veel kaks tingimust. Tavaliselt on lisatingimused seotud lõigu $[a, b]$ otspunktidega või nendele lähedaste punktidega ja seetõttu nimetatakse neid rajatingimusteks. Näiteks võib rajatingimustena nõuda, et splaini S_3 esimese tuletise väärtused lõigu $[a, b]$ otspunktides a ja b oleksid võrdsed etteantud arvudega α_1, β_1 :

$$S'_3(a) = \alpha_1, \quad S'_3(b) = \beta_1. \quad (5)$$

Analoogiliselt võib rajatingimustes nõuda, et splaini S_3 teise tuletise väärtused lõigu $[a, b]$ otspunktides oleksid võrdsed etteantud arvudega α_2 ja β_2 :

$$S''_3(a) = \alpha_2, \quad S''_3(b) = \beta_2. \quad (6)$$

Kuna võrgu Δ_n sisesõlmedes $x_1, \dots, x_{n-1}$ võivad splaini S_3 kolmandat järku tuletised olla katkevad, siis võib 'rajatingimusena' nõuda S_3 kolmanda tuletise pidevust äärmistes sisesõlmedes x_1 ja x_{n-1} :

$$S'''_3(x_1 - 0) = S'''_3(x_1 + 0), \quad S'''_3(x_{n-1} - 0) = S'''_3(x_{n-1} + 0). \quad (7)$$

Siin ja edaspidi

$$S'''_3(x_i - 0) := \lim_{\substack{x \rightarrow x_i \\ x < x_i}} S'''_3(x), \quad S'''_3(x_i + 0) := \lim_{\substack{x \rightarrow x_i \\ x > x_i}} S'''_3(x), \quad i = 1, \dots, n-1.$$

3 Interpoleeriva kuupsplaini konstrueerimine

Olgu antud võrk Δ_n sõlmedega $a = x_0 < x_1 < \dots < x_n = b$ ja olgu

$$h_i := x_i - x_{i-1}, \quad i = 1, \dots, n. \quad (8)$$

Otsime väärtusi $f_0, f_1, \dots, f_n$ interpoleerivat kuupsplaini $S_3(x)$, mis lõigul $[x_{i-1}, x_i]$ avaldub kujul (2), kus

$c_0^{(i)}, c_1^{(i)}, c_2^{(i)}, c_3^{(i)}$ on tundmatud kordajad. Siis iga $i = 1, \dots, n$ korral

$$S'_3(x) = c_1^{(i)} + 2c_2^{(i)}(x - x_{i-1}) + 3c_3^{(i)}(x - x_{i-1})^2, \quad x \in [x_{i-1}, x_i],$$

$$S''_3(x) = 2c_2^{(i)} + 6c_3^{(i)}(x - x_{i-1}) \quad x \in [x_{i-1}, x_i].$$

Tähistame

$$m_i := S'_3(x_i), \quad i = 0, 1, \dots, n.$$

Siis

$$S_3(x_{i-1}) = f_{i-1}, \quad S_3(x_i) = f_i, \quad S'_3(x_{i-1}) = m_{i-1}, \quad S'_3(x_i) = m_i,$$

millest järeldub, et

$$c_0^{(i)} = f_{i-1}, \quad c_1^{(i)} = m_{i-1}.$$

Lisaks saame

$$\begin{cases} m_{i-1}h_i + c_2^{(i)}h_i^2 + c_3^{(i)}h_i^3 = f_i - f_{i-1}, \\ 2c_2^{(i)}h_i + 3c_3^{(i)}h_i^2 = m_i - m_{i-1}. \end{cases}$$

kus $c_2^{(i)}$ ja $c_3^{(i)}$ on otsitavad. Lahendades viimase võrrandisüsteemi, saame

$$c_2^{(i)} = \frac{1}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right),$$

$$c_3^{(i)} = \frac{1}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right).$$

Seega saame kuupsplaini $S_3(x)$ igal osalõigul $[x_{i-1}, x_i]$ ($i = 1, \dots, n$) esitada kujul

$$\begin{aligned} S_3(x) &= f_{i-1} + m_{i-1}(x - x_{i-1}) \\ &\quad + \frac{1}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) (x - x_{i-1})^2 \\ &\quad + \frac{1}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right) (x - x_{i-1})^3, \quad x \in [x_{i-1}, x_i], \end{aligned} \quad (9)$$

Diferentseerimise abil leiame esitusest (9), et $x \in [x_{i-1}, x_i]$ ($i = 1, \dots, n$) korral

$$\begin{aligned} S_3'(x) &= m_{i-1} + \frac{2}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) (x - x_{i-1}) \\ &\quad + \frac{3}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right) (x - x_{i-1})^2 \end{aligned} \quad (10)$$

ning

$$\begin{aligned} S_3''(x) &= \frac{2}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) \\ &\quad + \frac{6}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right) (x - x_{i-1}). \end{aligned} \quad (11)$$

4 Pidevustingimused

Veendume kõigepealt, et esituse (9) abil defineeritud funktsioon S_3 on pidevalt diferentseeruv lõigus $[a, b]$. Valemist (9) on näha, et funktsioon S_3 on igal osalõigul $[x_{i-1}, x_i]$ ($i = 1, \dots, n$) kuuppolünoom, seega sellel osalõigul ka pidevalt diferentseeruv. Järelikult on vaja kontrollida vaid funktsiooni S_3 ja S'_3 pidevust sisesõlmedes x_i , $i = 1, \dots, n - 1$.

Seosest (9) järeldub iga $i = 1, \dots, n - 1$ korral, et

$$\begin{aligned} S_3(x_i - 0) &= f_{i-1} + m_{i-1}h_i + \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i)\right) \\ &\quad + \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1})\right) \\ &= f_i. \end{aligned}$$

Analoogiliselt saame iga $i = 1, \dots, n - 1$ korral, et

$$\begin{aligned} S_3(x_i + 0) &= f_i + m_i(x_i - x_i) \\ &\quad + \frac{1}{h_{i+1}^2} \left(3(f_{i+1} - f_i) - h_{i+1}(2m_i + m_{i+1})\right) (x_i - x_i)^2 \\ &\quad + \frac{1}{h_{i+1}^3} \left(h_{i+1}(m_i + m_{i+1}) - 2(f_{i+1} - f_i)\right) (x_i - x_i)^3 \\ &= f_i. \end{aligned}$$

Seega

$$S_3(x_i - 0) = f_i = S_3(x_i + 0), \quad i = 1, \dots, n - 1.$$

Järelikult on S_3 pidev lõigul $[a, b]$.

Kontrollime funktsiooni S'_3 pidevust võrgu sisesõlmedes x_i , $i = 1, \dots, n - 1$. Selleks piisab näidata, et iga sisesõlme x_i korral vasak- ja parempoolne piirväärtus langeb kokku.

Seosest (10) järeldub iga $i = 1, \dots, n - 1$ korral, et

$$\begin{aligned}
S'_3(x_i - 0) &= m_{i-1} + \frac{2}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) (x_i - x_{i-1}) \\
&\quad + \frac{3}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right) (x_i - x_{i-1})^2 \\
&= m_{i-1} + \frac{2}{h_i} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) \\
&\quad + \frac{3}{h_i} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right) \\
&= m_i.
\end{aligned} \tag{12}$$

Analoogiliselt saame iga $i = 1, \dots, n - 1$ korral, et

$$\begin{aligned}
S'_3(x_i + 0) &= m_i + \frac{2}{h_{i+1}^2} \left(3(f_{i+1} - f_i) - h_{i+1}(2m_i + m_{i+1}) \right) (x_i - x_i) \\
&\quad + \frac{3}{h_{i+1}^3} \left(h_{i+1}(m_i + m_{i+1}) - 2(f_{i+1} - f_i) \right) (x_i - x_i)^2 \\
&= m_i.
\end{aligned} \tag{13}$$

Seega

$$S'_3(x_i - 0) = S'_3(x_i + 0), \quad i = 1, \dots, n - 1.$$

Järelikult on S'_3 pidev lõigul $[a, b]$.

Niisiis oleme saanud, et valemiga (9) määratud funktsiooni S_3 korral $S_3 \in C^1[a, b]$. Selleks, et kehtiks $S_3 \in C^2[a, b]$, tuleb nõuda S_3 teise tuletise pidevust sisesõlmedes, s.t peab kehtima

$$S''_3(x_i - 0) = S''_3(x_i + 0), \quad i = 1, \dots, n - 1. \tag{14}$$

Kasutades teise tuletise esitust (11), saame

$$\begin{aligned}
S_3''(x_i - 0) &= \frac{2}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) \\
&\quad + \frac{6}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right) (x_i - x_{i-1}) \\
&= \frac{2}{h_i^2} \left(3(f_i - f_{i-1}) - h_i(2m_{i-1} + m_i) \right) \\
&\quad + \frac{6}{h_i} \left((m_{i-1} + m_i) - \frac{2(f_i - f_{i-1})}{h_i} \right) \\
&= -\frac{6(f_i - f_{i-1})}{h_i^2} + \frac{2}{h_i} (m_{i-1} + 2m_i).
\end{aligned} \tag{15}$$

Analoogiliselt leiame

$$\begin{aligned}
S_3''(x_i + 0) &= \frac{2}{h_{i+1}^2} \left(3(f_{i+1} - f_i) - h_{i+1}(2m_i + m_{i+1}) \right) \\
&\quad + \frac{6}{h_{i+1}^3} \left(h_{i+1}(m_i + m_{i+1}) - 2(f_{i+1} - f_i) \right) (x_i - x_i) \\
&= \frac{2}{h_{i+1}^2} \left(3(f_{i+1} - f_i) - h_{i+1}(2m_i + m_{i+1}) \right) \\
&= \frac{6(f_{i+1} - f_i)}{h_{i+1}^2} - \frac{2}{h_{i+1}} (2m_i + m_{i+1}).
\end{aligned} \tag{16}$$

Asendades need avaldised tingimusse (14), saame iga $i = 1, \dots, n - 1$ korral võrduse

$$-\frac{6(f_i - f_{i-1})}{h_i^2} + \frac{2}{h_i} (m_{i-1} + 2m_i) = \frac{6(f_{i+1} - f_i)}{h_{i+1}^2} - \frac{2}{h_{i+1}} (2m_i + m_{i+1})$$

ehk

$$\frac{2}{h_i} (m_{i-1} + 2m_i) + \frac{2}{h_{i+1}} (2m_i + m_{i+1}) = \frac{6(f_i - f_{i-1})}{h_i^2} + \frac{6(f_{i+1} - f_i)}{h_{i+1}^2}.$$

Korrutades viimase saadud võrduse mõlemad pooled läbi teguriga

$$\frac{h_i h_{i+1}}{2(h_i + h_{i+1})},$$

jõuame järgmise lineaarse võrrandisüsteemini suuruste $m_0, \dots, m_n$ suhtes:

$$\frac{h_{i+1}}{h_i + h_{i+1}} m_{i-1} + 2m_i + \frac{h_i}{h_i + h_{i+1}} m_{i+1} = \frac{3h_i h_{i+1}}{h_i + h_{i+1}} \left(\frac{f_i - f_{i-1}}{h_i^2} + \frac{f_{i+1} - f_i}{h_{i+1}^2} \right), \quad (17)$$

kus $i = 1, \dots, n - 1$.

Kuna otsitavaid $m_0, m_1, \dots, m_n$ on kokku $n + 1$, kuid võrrandeid (17) on kokku ainult $n - 1$, siis ei piisa neist veel suuruste $m_0, \dots, m_n$ üheseks määramiseks. Seetõttu tuleb lisada kaks täiendavat võrrandit. Selleks võime kasutada ühte rajatingimustest (5)–(7).

5 Interpoleeriva kuupsplaini olemasolu ja ühesus

Selleks, et kirjutada võrrandid (17) kompaktsel kujul, tähistame

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \quad \mu_i = \frac{h_i}{h_i + h_{i+1}}, \quad i = 1, \dots, n-1, \quad (18)$$

$$d_i = 3\lambda_i \frac{f_i - f_{i-1}}{h_i} + 3\mu_i \frac{f_{i+1} - f_i}{h_{i+1}}, \quad i = 1, \dots, n-1. \quad (19)$$

Siis saame võrrandid (17) esitada kujul

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = d_i, \quad i = 1, \dots, n-1. \quad (20)$$

milles $m_0, \dots, m_n$ on otsitavad. Kuna süsteemi (20) võrrandite arv on $n-1$. Nendest üksi ei piisa tundmatute $m_0, \dots, m_n$ leidmiseks. Seetõttu lisame võrranditele (20) kaks võrrandit, mille saame rajatingimusest (5) või (6) või (7).

Kui kasutada rajatingimusi (5), siis saame võrdused

$$\alpha_1 = m_0 = S'_3(a), \quad \beta_1 = m_n = S'_3(b). \quad (21)$$

Seostest (20) ja (21) järeldub, et rajatingimuste (5) korral saame suuruste $m_0, \dots, m_n$ leidmiseks $n+1$ võrrandist koosneva lineaarse algebralise võrran-

disüsteemi kujul

$$\begin{cases} m_0 = \alpha_1, \\ \lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = d_i, & i = 1, \dots, n-1, \\ m_n = \beta_1, \end{cases} \quad (22)$$

kus λ_i , μ_i ja d_i on antud seostega (18) ja (19) ning $m_0, \dots, m_n$ on otsitavad.

Süsteemi (22) võib esitada ka maatrikskujul

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \lambda_1 & 2 & \mu_1 & \cdots & 0 & 0 \\ 0 & \lambda_2 & 2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mu_{n-2} & 0 \\ 0 & 0 & \cdots & \lambda_{n-1} & 2 & \mu_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \\ m_n \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ d_1 \\ \vdots \\ d_{n-1} \\ \beta_1 \end{pmatrix}. \quad (23)$$

Seosest (18) on näha, et süsteemi (23) maatriks on domineeriva peadiagonaaliga.

Kui kasutada rajatingimusi (6) ja seoseid (15)–(16), saame

$$S_3''(a) = \frac{2}{h_1^2} \left(3(f_1 - f_0) - h_1(2m_0 + m_1) \right) = \alpha_2$$

ehk

$$2m_0 + m_1 = 3 \frac{f_1 - f_0}{h_1} - \frac{\alpha_2 h_1}{2}.$$

Analoogiliselt leiame

$$S_3''(b) = -\frac{6(f_n - f_{n-1})}{h_n^2} + \frac{2}{h_n}(m_{n-1} + 2m_n) = \beta_2$$

ehk

$$m_{n-1} + 2m_n = 3\frac{f_n - f_{n-1}}{h_n} + \frac{\beta_2 h_n}{2}.$$

Koos võrranditega (20) saame süsteemi

$$\begin{cases} 2m_0 + m_1 = 3\frac{f_1 - f_0}{h_1} - \frac{\alpha_2 h_1}{2}, \\ \lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = d_i, & i = 1, \dots, n-1, \\ m_{n-1} + 2m_n = 3\frac{f_n - f_{n-1}}{h_n} + \frac{\beta_2 h_n}{2}, \end{cases} \quad (24)$$

kus λ_i , μ_i ja d_i on antud seostega (18) ja (19) ning $m_0, \dots, m_n$ on otsitavad.

Süsteemi (24) maatrikskuju on järgmine:

$$\begin{pmatrix} 2 & 1 & 0 & 0 & \cdots & 0 \\ \lambda_1 & 2 & \mu_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 2 & \mu_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{n-1} & 2 & \mu_{n-1} \\ 0 & \cdots & 0 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{pmatrix} = \begin{pmatrix} 3\frac{f_1 - f_0}{h_1} - \frac{\alpha_2 h_1}{2} \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ 3\frac{f_n - f_{n-1}}{h_n} + \frac{\beta_2 h_n}{2} \end{pmatrix}. \quad (25)$$

Ka süsteemi (25) maatriks on domineeriva peadiagonaaliga.

Vaatleme nüüd rajatingimusi (7). Valemist (9) saame $x \in [x_{i-1}, x_i]$ korral

$$S_3'''(x) = \frac{6}{h_i^3} \left(h_i(m_{i-1} + m_i) - 2(f_i - f_{i-1}) \right).$$

Seega punkti x_1 korral

$$S_3'''(x_1 - 0) = \frac{6}{h_1^3} \left(h_1(m_0 + m_1) - 2(f_1 - f_0) \right)$$

ning

$$S_3'''(x_1 + 0) = \frac{6}{h_2^3} \left(h_2(m_1 + m_2) - 2(f_2 - f_1) \right).$$

Rajatingimuses (7) esimesest seosest saame seega võrduse

$$\frac{h_1(m_0 + m_1) - 2(f_1 - f_0)}{h_1^3} = \frac{h_2(m_1 + m_2) - 2(f_2 - f_1)}{h_2^3}.$$

Korrutades selle võrduse mõlemad pooled läbi teguriga $h_1^2 h_2^2$, saame

$$h_2^2 m_0 + (h_2^2 - h_1^2) m_1 - h_1^2 m_2 = \alpha_3,$$

kus

$$\alpha_3 = 2 \left(\frac{h_2^2}{h_1} (f_1 - f_0) - \frac{h_1^2}{h_2} (f_2 - f_1) \right).$$

Seega

$$m_0 = \frac{\alpha_3}{h_2^2} - \frac{h_2^2 - h_1^2}{h_2^2} m_1 + \frac{h_1^2}{h_2^2} m_2. \quad (26)$$

Punkti x_{n-1} korral leiame, et

$$S_3'''(x_{n-1} - 0) = \frac{6}{h_{n-1}^3} \left(h_{n-1}(m_{n-2} + m_{n-1}) - 2(f_{n-1} - f_{n-2}) \right)$$

ning

$$S_3'''(x_{n-1} + 0) = \frac{6}{h_n^3} \left(h_n(m_{n-1} + m_n) - 2(f_n - f_{n-1}) \right).$$

Rajatingimuses (7) teine seos annab seega võrduse

$$\frac{h_{n-1}(m_{n-2} + m_{n-1}) - 2(f_{n-1} - f_{n-2})}{h_{n-1}^3} = \frac{h_n(m_{n-1} + m_n) - 2(f_n - f_{n-1})}{h_n^3}.$$

Korrutades selle võrduse mõlemad pooled läbi teguriga $h_{n-1}^2 h_n^2$, saame

$$h_n^2 m_{n-2} + (h_n^2 - h_{n-1}^2) m_{n-1} - h_{n-1}^2 m_n = \beta_3,$$

kus

$$\beta_3 = 2 \left(\frac{h_n^2}{h_{n-1}} (f_{n-1} - f_{n-2}) - \frac{h_{n-1}^2}{h_n} (f_n - f_{n-1}) \right).$$

Seega

$$m_n = \frac{h_n^2}{h_{n-1}^2} m_{n-2} + \frac{h_n^2 - h_{n-1}^2}{h_{n-1}^2} m_{n-1} - \frac{\beta_3}{h_{n-1}^2}. \quad (27)$$

Asendades m_0 avaldise (26) esimesse võrrandisse

$$\lambda_1 m_0 + 2m_1 + \mu_1 m_2 = d_1$$

võrranditest (20), saame

$$\left(1 + \frac{h_1}{h_2} \right) m_1 + \frac{h_1}{h_2} m_2 = \tilde{d}_1,$$

kus

$$\tilde{d}_1 = \frac{h_2}{h_1(h_1 + h_2)} (f_1 - f_0) + \frac{h_1(2h_1 + 3h_2)}{h_2^2(h_1 + h_2)} (f_2 - f_1) = d_1 - \frac{\alpha_3}{h_2(h_1 + h_2)}$$

Asendades m_n avaldise (27) viimasesse sisevõrrandisse

$$\lambda_{n-1} m_{n-2} + 2m_{n-1} + \mu_{n-1} m_n = d_{n-1},$$

saame

$$\frac{h_n}{h_{n-1}} m_{n-2} + \left(1 + \frac{h_n}{h_{n-1}} \right) m_{n-1} = \tilde{d}_{n-1},$$

kus

$$\begin{aligned} \tilde{d}_{n-1} &= \frac{h_n(3h_{n-1} + 2h_n)}{h_{n-1}^2(h_{n-1} + h_n)} (f_{n-1} - f_{n-2}) + \frac{h_{n-1}}{h_n(h_{n-1} + h_n)} (f_n - f_{n-1}) \\ &= d_{n-1} + \frac{\beta_3}{h_{n-1}(h_{n-1} + h_n)}. \end{aligned}$$

Seega jõuame tundmatute $m_1, \dots, m_{n-1}$ suhtes võrrandisüsteemini

$$\begin{cases} \left(1 + \frac{h_1}{h_2}\right) m_1 + \frac{h_1}{h_2} m_2 = \tilde{d}_1, \\ \lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = d_i, & i = 2, \dots, n-2, \\ \frac{h_n}{h_{n-1}} m_{n-2} + \left(1 + \frac{h_n}{h_{n-1}}\right) m_{n-1} = \tilde{d}_{n-1}, \end{cases} \quad (28)$$

mille maatrikskuju on järgmine:

$$\begin{pmatrix} 1 + \frac{h_1}{h_2} & \frac{h_1}{h_2} & 0 & \dots & 0 \\ \lambda_2 & 2 & \mu_2 & \ddots & \vdots \\ 0 & \lambda_3 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \mu_{n-2} \\ 0 & \dots & 0 & \frac{h_n}{h_{n-1}} & 1 + \frac{h_n}{h_{n-1}} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} \tilde{d}_1 \\ d_2 \\ d_3 \\ \vdots \\ \tilde{d}_{n-1} \end{pmatrix}. \quad (29)$$

Süsteemi (29) maatriksi puhul kehtivad järgmised võrratused:

$$1 + \frac{h_1}{h_2} > \frac{h_1}{h_2},$$

$$2 > \lambda_i + \mu_i = 1, \quad i = 2, \dots, n-2,$$

$$1 + \frac{h_n}{h_{n-1}} > \frac{h_n}{h_{n-1}}.$$

Seega on võrrandisüsteem (29) domineeriva peadiagonaaliga.

Kokkuvõttes on süsteemide (23) ja (25) maatriksid domineeriva peadiagonaaliga. Järelikult on need süsteemid üheselt lahenduvad ning suurused $m_0, \dots, m_n$ rajatingimuste (5) ja (6) korral üheselt määratud. Rajatingimuse (7) puhul on süsteemi (29) maatriks domineeriva peadiagonaaliga, mistõttu on suurused $m_1, \dots, m_{n-1}$ üheselt määratud. Seejärel avalduvad m_0 ja m_n valemite

(26) ja (27) kaudu, mistõttu on kõik suurused $m_0, \dots, m_n$ üheselt määratud. Me näeme, et kõigi kolme rajatingimuste juhul (5)–(7) saame suurused $m_0, \dots, m_n$ leida ning asendada valemisse (9). Seega saame valemi (9) abil üheselt määrata kuupsplaiini S_3 , mis rahuldab interpolatsioonitingimusi (4) ja vastavaid rajatingimusi.

6 Interpolatsiooniviga ja näited

Eespool konstrueerisime etteantud funktsioonile f interpoleeriva kuupsplaini S_3 . Järgmine teoreem iseloomustab interpoleerimisviga $S_3(x) - f(x)$ ning näitab, kuidas vea suurus sõltub võrgu sammust h (teoreemi tõestuse võib leida raamatust [3]).

Teoreem 1. *Olgu $S_3(x)$ võrguga Δ_n (vt. (1)) seotud kuupsplain, mis interpoleerib funktsiooni $f \in C^4[a, b]$ ja rahuldab ühte rajatingimuste komplektidest (5), (6) või (7). Siis kehtib hinnang*

$$\max_{x \in [a, b]} |S_3(x) - f(x)| \leq \frac{5}{384} h^4 \max_{x \in [a, b]} |f^{(4)}(x)|, \quad (30)$$

kus

$$h = \frac{b - a}{n}, \quad n \in \mathbb{N}, \quad n \geq 3.$$

Märkus 1. Hinnangus (30) olevat konstanti $\frac{5}{384}$ ei saa vähendada [3].

6.1 Näited

Järgnevalt vaatleme funktsiooni $f \in C^4[a, b]$ interpoleeriva kuupsplaini S_3 käitumist rajatingimuste (5)–(7) korral.

Võtame lõigul $[a, b]$ ühtlase võrgu Δ_n punktidega

$$x_i = a + ih, \quad i = 0, 1, \dots, n, \quad h = \frac{b - a}{n}, \quad n \geq 3. \quad (31)$$

Iga valitud n korral konstrueerime kuupsplaini $S_3 = S_3(x)$, mis interpoleerib funktsiooni f väärtusi võrgu (31) sõlmedes:

$$S_3(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

Splaini S_3 üheseks määramiseks kasutame ühte eespool kirjeldatud rajatingimustest (5)–(7). Rajatingimuse (5) korral määratakse S_3 tingimustel:

$$S'_3(a) = f'(a), \quad S'_3(b) = f'(b). \quad (32)$$

Analoogiliselt saab leida interpoleeriva kuupsplaini S_3 rajatingimuse (6) korral, kus

$$S''_3(a) = f''(a), \quad S''_3(b) = f''(b). \quad (33)$$

Mõnevõrra teistsugune võimalus on kasutada rajatingimust (7), kus nõutakse S_3 kolmanda tuletise pidevust äärmistes sisesõlmedes:

$$S'''_3(x_1 - 0) = S'''_3(x_1 + 0), \quad S'''_3(x_{n-1} - 0) = S'''_3(x_{n-1} + 0). \quad (34)$$

Suuruse

$$\max_{x \in [a, b]} |S_3(x) - f(x)|$$

ligikaudseks leidmiseks arvutame

$$\varepsilon_n = \max_{\substack{j=1, \dots, n \\ k=1, \dots, 10}} |S_3(x_{jk}) - f(x_{jk})|, \quad (35)$$

kus

$$x_{jk} = x_{j-1} + k \frac{h}{10}, \quad j = 1, \dots, n, \quad k = 1, \dots, 10.$$

Suurus ε_n annab ligikaudse interpoleerimisvea $S_3(x) - f(x)$ jaoks lõigul $[a, b]$.

Lisaks arvutame suhted

$$\Theta_n = \frac{\varepsilon_{n/2}}{\varepsilon_n}, \quad n = 16, 32, \dots \quad (36)$$

Vaatleme hinnangut (30). Kui $n/2$ asemel kasutada n , siis võrgu samm väheneb kaks korda. Kui n korral on võrgu samm

$$h = \frac{b-a}{n},$$

siis $n/2$ korral on vastav samm

$$\frac{b-a}{n/2} = 2h.$$

Seega on hinnangu (30) parempoolne avaldis $n/2$ korral

$$\frac{5}{384}(2h)^4 \max_{x \in [a,b]} |f^{(4)}(x)|$$

ning n korral

$$\frac{5}{384}h^4 \max_{x \in [a,b]} |f^{(4)}(x)|.$$

Nende avaldiste suhe on

$$\frac{\frac{5}{384}(2h)^4 \max_{x \in [a,b]} |f^{(4)}(x)|}{\frac{5}{384}h^4 \max_{x \in [a,b]} |f^{(4)}(x)|} = \frac{(2h)^4}{h^4} = 16.$$

Seega peaks Θ_n väärtus n kasvades lähenema arvule 16.

Näide 1. Vaatleme funktsiooni

$$f(x) = \frac{1}{x}, \quad x \in \left[\frac{1}{5}, 3\right].$$

Funktsioon f kuulub ruumi $C^4 \left[\frac{1}{5}, 3\right]$, mistõttu saab selle funktsiooni korral rakendada teoreemis 1 toodud veahinnangut. Selle funktsiooni esimesed neli

tuletist on

$$f'(x) = -\frac{1}{x^2}, \quad f''(x) = \frac{2}{x^3}, \quad f'''(x) = -\frac{6}{x^4}, \quad f^{(4)}(x) = \frac{24}{x^5}.$$

Olgu lõigul $[\frac{1}{5}, 3]$ antud ühtlane võrk

$$x_i = 0,2 + ih, \quad i = 0, 1, \dots, n, \quad h = \frac{3 - 0,2}{n} = \frac{2,8}{n}.$$

Konstrueerime kuupsplaini $S_3 = S_3(x)$, mis interpoleerib funktsiooni $f(x) = 1/x$ väärtusi võrgu sõlmedes. Vaatleme kõiki kolme rajatingimust. Kasutame (32), et määrata funktsiooni esimese tuletise väärtused:

$$S'_3(0,2) = f'(0,2) = -25, \quad S'_3(3) = f'(3) = -\frac{1}{9}. \quad (37)$$

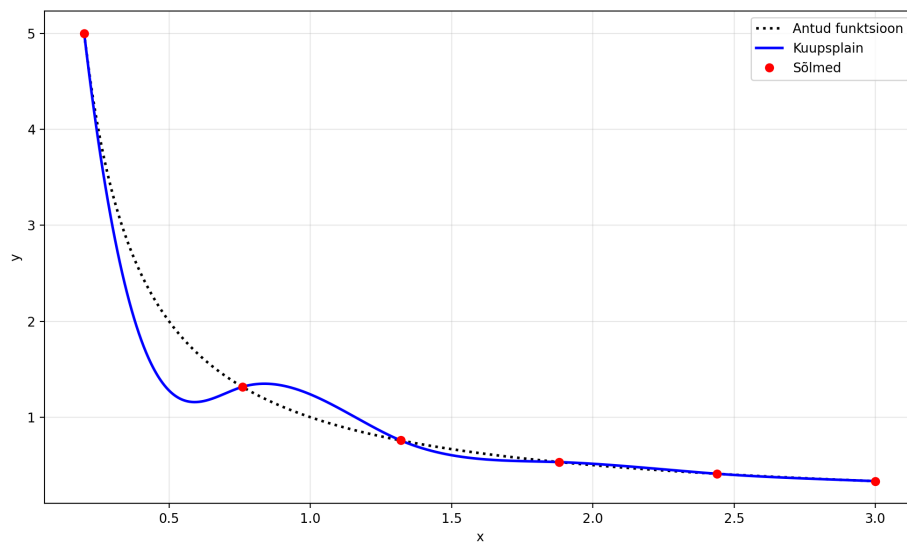
Teiseks kasutame (33), et määrata funktsiooni teise tuletise väärtused:

$$S''_3(0,2) = f''(0,2) = 250, \quad S''_3(3) = f''(3) = \frac{2}{27}. \quad (38)$$

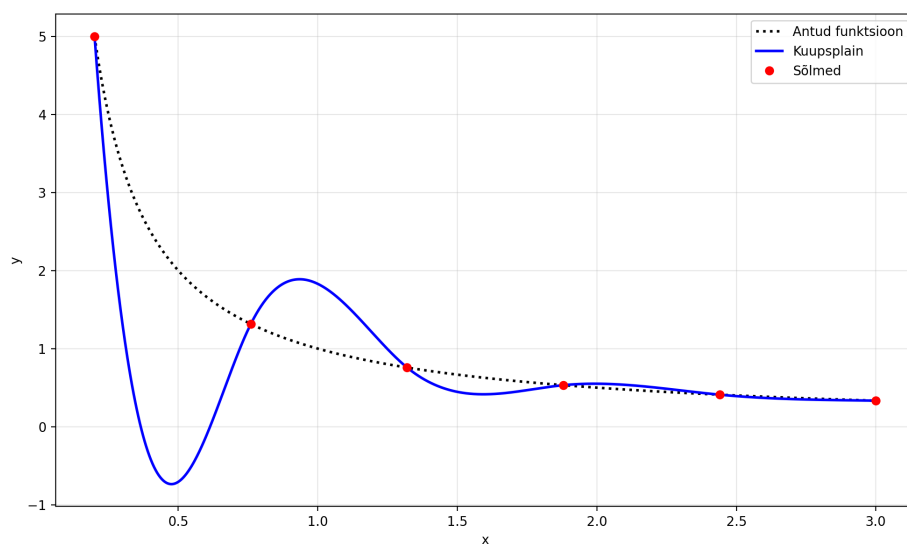
Lõpuks kasutame (34), et nõuda splaini kolmanda tuletise pidevust punktides x_1 ja x_{n-1} . Joonistel 1–3 on kujutatud funktsiooni $f(x) = 1/x$ ja seda interpoleeriva kuupsplaini graafikud kolme erineva rajatingimuse korral.

Vea arvutamiseks kasutame suurust ε_n , mis on defineeritud valemiga (35), ning järjestikuste vigade suhteid Θ_n , mis on defineeritud valemiga (36). Tulemused on esitatud tabelites 1.

Tabelitest 1 on näha, et võrgu tihendamisel viga väheneb kõigi kolme rajatingimuse korral. Suhted Θ_n kasvavad ja lähevad arvule 16 lähemale.



Joonis 1: Funktsioon $f(x) = \frac{1}{x}$ (punktiiriga) ja seda interpoleeriv kuupsplain (tume joon) rajatingimuste (37) korral.

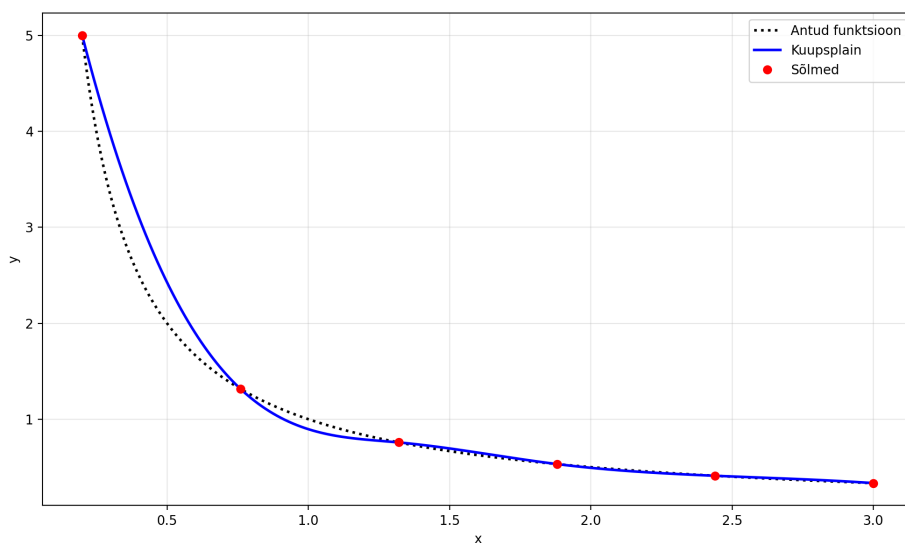


Joonis 2: Funktsioon $f(x) = \frac{1}{x}$ (punktiiriga) ja seda interpoleeriv kuupsplain (tume joon) rajatingimuste (38) korral.

Tabel 1: Funktsiooni $f(x) = 1/x$ interpoleerimisvead erinevate rajatingimuste korral.

Rajatingimused (37)			Rajatingimused (38)		
n	ε_n	Θ_n	n	ε_n	Θ_n
8	$2,7801 \cdot 10^{-1}$	–	8	$9,2841 \cdot 10^{-1}$	–
16	$4,8918 \cdot 10^{-2}$	5,68	16	$1,3693 \cdot 10^{-1}$	6,78
32	$5,9549 \cdot 10^{-3}$	8,21	32	$1,5219 \cdot 10^{-2}$	9,00
64	$5,3396 \cdot 10^{-4}$	11,15	64	$1,3183 \cdot 10^{-3}$	11,54
128	$3,9438 \cdot 10^{-5}$	13,54	128	$9,6793 \cdot 10^{-5}$	13,62
256	$2,6458 \cdot 10^{-6}$	14,91	256	$6,5161 \cdot 10^{-6}$	14,85
512	$1,7039 \cdot 10^{-7}$	15,53	512	$4,2142 \cdot 10^{-7}$	15,46
1024	$1,0790 \cdot 10^{-8}$	15,79	1024	$2,6766 \cdot 10^{-8}$	15,74
2048	$6,7851 \cdot 10^{-10}$	15,90	2048	$1,6860 \cdot 10^{-9}$	15,88
4096	$4,2530 \cdot 10^{-11}$	15,95	4096	$1,0578 \cdot 10^{-10}$	15,94

Rajatingimused (34)		
n	ε_n	Θ_n
8	$3,5498 \cdot 10^{-1}$	–
16	$1,0700 \cdot 10^{-1}$	3,32
32	$2,1264 \cdot 10^{-2}$	5,03
64	$2,8542 \cdot 10^{-3}$	7,45
128	$2,8385 \cdot 10^{-4}$	10,06
256	$2,2917 \cdot 10^{-5}$	12,39
512	$1,6387 \cdot 10^{-6}$	13,99
1024	$1,0973 \cdot 10^{-7}$	14,93
2048	$7,1023 \cdot 10^{-9}$	15,45
4096	$4,5176 \cdot 10^{-10}$	15,72



Joonis 3: Funktsioon $f(x) = \frac{1}{x}$ (punktiiriga) ja seda interpoleeriv kuupsplain (tume joon) rajatingimuste (34) korral.

Näide 2. Vaatleme funktsiooni

$$f(x) = \sin(5x) + \cos(2x), \quad x \in [0, 2\pi].$$

See funktsioon kuulub ruumi $C^4[0, 2\pi]$. Selle esimesed neli tuletist on

$$f'(x) = 5 \cos(5x) - 2 \sin(2x),$$

$$f''(x) = -25 \sin(5x) - 4 \cos(2x),$$

$$f'''(x) = -125 \cos(5x) + 8 \sin(2x),$$

$$f^{(4)}(x) = 625 \sin(5x) + 16 \cos(2x).$$

Olgu lõigul $[0, 2\pi]$ antud ühtlane võrk

$$x_i = ih, \quad i = 0, 1, \dots, n, \quad h = \frac{2\pi}{n}.$$

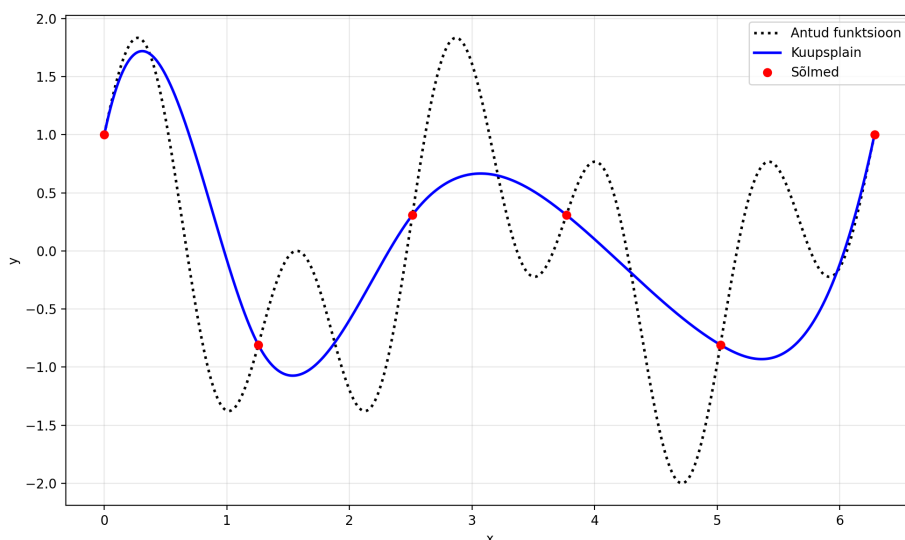
Konstrueerime kuupsplaini $S_3 = S_3(x)$, mis interpoleerib funktsiooni $f(x) = \sin(5x) + \cos(2x)$ väärtusi võrgu sõlmedes. Kasutame (32), et määrata funktsiooni esimese tuletise väärtused:

$$S'_3(0) = f'(0) = 5, \quad S'_3(2\pi) = f'(2\pi) = 5. \quad (39)$$

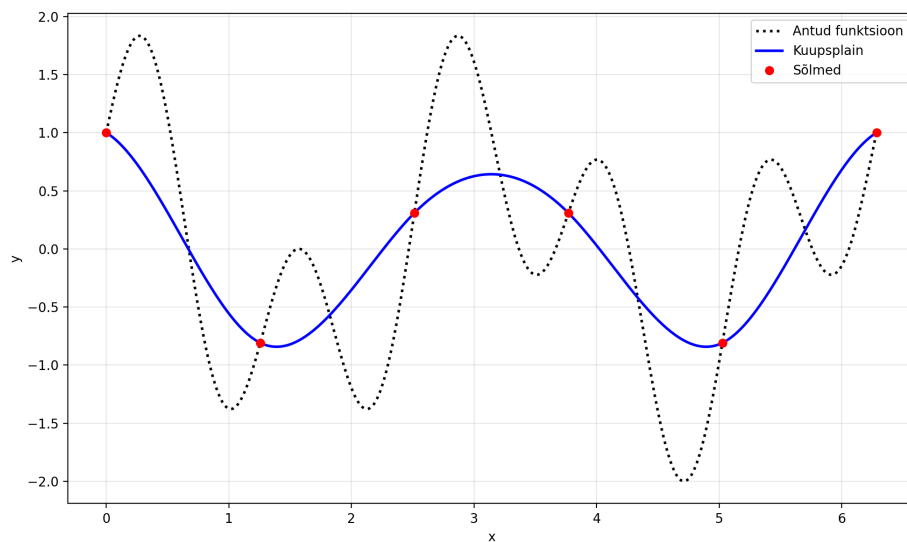
Teiseks kasutame (33), et määrata otspunktides funktsiooni teise tuletise väärtused:

$$S''_3(0) = f''(0) = -4, \quad S''_3(2\pi) = f''(2\pi) = -4. \quad (40)$$

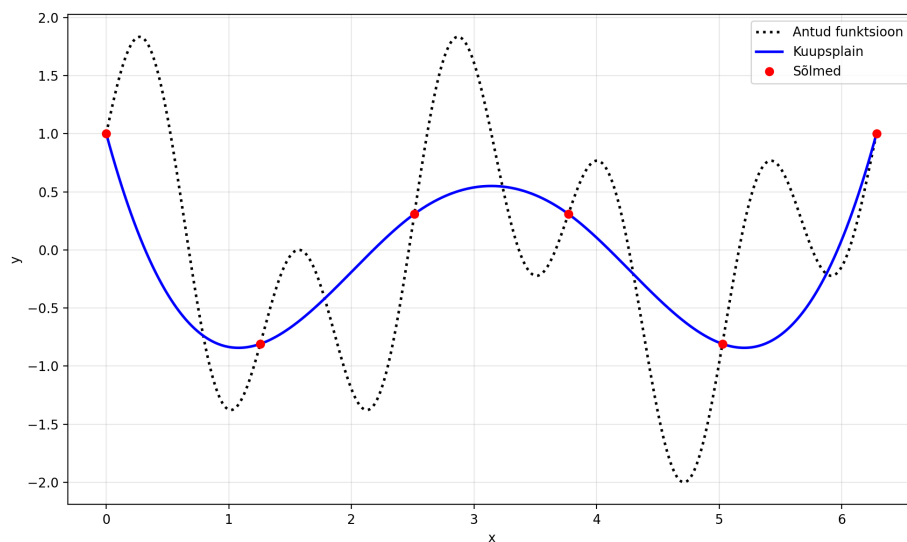
Lõpuks kasutame (34), et nõuda splaini kolmanda tuletise pidevust punktides x_1 ja x_{n-1} . Joonistel 4–6 on kujutatud funktsiooni $f(x) = \sin(5x) + \cos(2x)$ ja vastava interpoleeriva kuupsplaini graafikud kolme erineva rajatingimuse korral.



Joonis 4: Funktsioon $f(x) = \sin(5x) + \cos(2x)$ (punktiiriga) ja seda interpoleeriv kuupsplain (tume joon) rajatingimuste (39) korral.



Joonis 5: Funktsioon $f(x) = \sin(5x) + \cos(2x)$ (punktiiriga) ja seda interpooleeriv kuupsplain (tume joon) rajatingimuste (40) korral.



Joonis 6: Funktsioon $f(x) = \sin(5x) + \cos(2x)$ (punktiiriga) ja seda interpooleeriv kuupsplain (tume joon) rajatingimuste (34) korral.

Vea arvutamiseks kasutame suurust ε_n , mis on defineeritud valemiga (35), ning järjestikuste vigade suhteid Θ_n , mis on defineeritud valemiga (36). Tulemused on esitatud tabelites 2.

Tabel 2: Funktsiooni $f(x) = \sin(5x) + \cos(2x)$ interpoleerimisvead erinevate rajatingimuste korral.

Rajatingimused (39)			Rajatingimused (40)		
n	ε_n	Θ_n	n	ε_n	Θ_n
8	$1,6687 \cdot 10^0$	–	8	$1,6715 \cdot 10^0$	–
16	$8,9790 \cdot 10^{-2}$	18,58	16	$8,6950 \cdot 10^{-2}$	19,22
32	$3,3300 \cdot 10^{-3}$	26,96	32	$3,0888 \cdot 10^{-3}$	28,15
64	$1,6227 \cdot 10^{-4}$	20,52	64	$1,6177 \cdot 10^{-4}$	19,09
128	$9,7848 \cdot 10^{-6}$	16,58	128	$9,7850 \cdot 10^{-6}$	16,53
256	$6,0686 \cdot 10^{-7}$	16,12	256	$6,0686 \cdot 10^{-7}$	16,12
512	$3,7876 \cdot 10^{-8}$	16,02	512	$3,7876 \cdot 10^{-8}$	16,02
1024	$2,3664 \cdot 10^{-9}$	16,01	1024	$2,3664 \cdot 10^{-9}$	16,01

Rajatingimused (34)		
n	ε_n	Θ_n
8	$2,1967 \cdot 10^0$	–
16	$3,3445 \cdot 10^{-1}$	6,57
32	$2,2602 \cdot 10^{-2}$	14,80
64	$8,3280 \cdot 10^{-4}$	27,14
128	$2,8189 \cdot 10^{-5}$	29,54
256	$9,6927 \cdot 10^{-7}$	29,08
512	$3,7876 \cdot 10^{-8}$	25,59
1024	$2,3664 \cdot 10^{-9}$	16,01

Tabelitest 2 on näha, et võrgu tihendamisel viga väheneb kõigi kolme rajatingimuse korral. Kuna funktsioon võngub lõigul $[0, 2\pi]$ kiiresti, sõltuvad suhted Θ_n väikeste n väärtuste korral tugevalt sellest, kui hästi võrgu sõlmed funktsiooni võnkumist tabavad. Tabelist on samuti näha, et Θ_n läheneb arvule 16.

Kokkuvõte

Töös käsitleti interpoleerivaid kuupsplaine ning nende ülesehitust etteantud võrgu ja interpolatsioonitingimuste põhjal.

Esmalt defineeriti kuupsplain kui osalõikudel kuuppolünoom, millel on pidev esimene ja teine tuletis. Seejärel tuletati kuupsplaini esitus osalõikudel ning näidati, kuidas pidevuse tingimustest saadakse lineaarne võrrandisüsteem kuupsplaini parameetrite määramiseks.

Vaadeldi kolme erinevat rajatingimust: esimese tuletise väärtuste etteandmist otspunktides, teise tuletise väärtuste etteandmist ning kolmanda tuletise pidevuse tingimusi äärmistes sisesõlmedes. Iga rajatingimuse korral tuletati vastav lineaarvõrrandisüsteem ja näidati, et süsteemid on üheselt lahenduvad ning vastav interpoleeriv kuupsplain eksisteerib.

Lisaks käsitleti kuupsplaini interpoleerimisvea hinnangut. Arvutuslikes näidetes uuriti, kuidas kuupsplain lähendab erinevaid funktsioone ning esitati kuupsplaini ja funktsiooni erinevus ehk viga.

Töö tulemusena saadi ülevaade sellest, kuidas interpoleerivat kuupsplaini esitada, millistel tingimustel on see üheselt määratud ning millise suurusjärguga on interpoleerimisviga.

Kasutatud allikad

- [1] Enn Tamme, Leo Võhandu ja Lembit Luht. *Arvutusmeetodid*. 2. ümber-
töötatud trükk. Tallinn: Valgus, 1986.
- [2] Gheorghe Micula ja Sanda Micula. *Handbook of Splines*. Dordrecht:
Kluwer Academic Publishers, 1999.
- [3] Ю. С. Завьялов, Б. И. Квасов и В. Л. Мирошниченко. *Методы
сплайн-функций*. Москва: Наука, 1980.

Lisa 1. Kood

Programmi fail `spline.py` on interaktiivne: kasutaja sisestab käsereal funktsiooni, vahemiku, n arvu, rajatingimuse ning veauuringu parameetrid. Programm arvutab interpoleeriva kuupsplaiini, vea ning salvestab tulemused (joonis ja väärtuste tabeli) eraldi kausta.

```
1 #.venv source
2
3 from datetime import datetime
4 from pathlib import Path
5
6 import matplotlib
7 import numpy as np
8
9 matplotlib.use("Agg")
10 import matplotlib.pyplot as plt
11
12 PUNKTE_OSALOIGUS = 10
13 ARVAVALDISE_NIMED = {"pi": np.pi, "e": np.e}
14
15
16 def arvuta_arvavaldis(avaldis):
17     avaldis = avaldis.strip().replace(",", ".")
18     return float(eval(avaldis, {"__builtins__": {}},
19                        ARVAVALDISE_NIMED))
20
21 def loe_ujuk(tekst):
22     while True:
23         try:
24             return arvuta_arvavaldis(input(tekst).strip())
25         except (
```

```

26         NameError,
27         SyntaxError,
28         TypeError,
29         ValueError,
30         ZeroDivisionError,
31         OverflowError,
32     ):
33         print("Sisesta arv voi avaldis, naiteks 0.2, pi
           voi 2*pi.")
34
35
36 def loe_taisarv_min(tekst, miinum):
37     while True:
38         try:
39             vaartus = int(input(tekst).strip())
40         except ValueError:
41             print("Sisesta taisarv.")
42             continue
43
44         if vaartus >= miinum:
45             return vaartus
46
47         print(f"Sisesta arv vahemalt {miinum}.")
48
49
50 def loe_tingimus():
51     print()
52     print("Vali rajatingimus:")
53     print("1 - esimese tuletise tingimus")
54     print("2 - teise tuletise tingimus")
55     print("3 - kolmanda tuletise pidevus punktides x1 ja x_(n
           -1)")
56

```

```

57     valik = input("Sinu valik (1/2/3): ").strip()
58     if valik == "1":
59         return {
60             "tyyp": 1,
61             "alpha": loe_ujuk("Sisesta alpha = S'(a): "),
62             "beta": loe_ujuk("Sisesta beta = S'(b): "),
63         }
64     if valik == "2":
65         return {
66             "tyyp": 2,
67             "alpha": loe_ujuk("Sisesta alpha = S''(a): "),
68             "beta": loe_ujuk("Sisesta beta = S''(b): "),
69         }
70     return {"tyyp": 3, "alpha": None, "beta": None}
71
72
73 def arvuta_funktsioon(avaldis, x):
74     x = np.asarray(x, dtype=float)
75     lubatud = {
76         "np": np,
77         "x": x,
78         "sin": np.sin,
79         "cos": np.cos,
80         "tan": np.tan,
81         "exp": np.exp,
82         "log": np.log,
83         "sqrt": np.sqrt,
84         "abs": np.abs,
85         "pi": np.pi,
86         "e": np.e,
87     }
88
89     with np.errstate(all="ignore"):

```

```

90     y = eval(avaldis, {"__builtins__": {}}, lubatud)
91 y = np.asarray(y, dtype=float)
92
93 if y.shape == ():
94     return np.full_like(x, float(y), dtype=float)
95
96 return np.broadcast_to(y, x.shape).astype(float)
97
98
99 def lahenda_m(f, h, lam, muu, d, tingimus):
100     n = len(h)
101
102     if tingimus["tyyp"] == 1:
103         A = np.zeros((n + 1, n + 1), dtype=float)
104         b = np.zeros(n + 1, dtype=float)
105         A[0, 0] = 1.0
106         A[-1, -1] = 1.0
107         b[0] = tingimus["alpha"]
108         b[-1] = tingimus["beta"]
109
110         for i in range(1, n):
111             A[i, i - 1] = lam[i - 1]
112             A[i, i] = 2.0
113             A[i, i + 1] = muu[i - 1]
114             b[i] = d[i - 1]
115
116         return np.linalg.solve(A, b), {}
117
118     if tingimus["tyyp"] == 2:
119         A = np.zeros((n + 1, n + 1), dtype=float)
120         b = np.zeros(n + 1, dtype=float)
121         A[0, 0] = 2.0
122         A[0, 1] = 1.0

```

```

123     A[-1, -2] = 1.0
124     A[-1, -1] = 2.0
125
126     b[0] = 3.0 * (f[1] - f[0]) / h[0] - tingimus["alpha"]
           * h[0] / 2.0
127     b[-1] = 3.0 * (f[-1] - f[-2]) / h[-1] + tingimus["
           beta"] * h[-1] / 2.0
128
129     for i in range(1, n):
130         A[i, i - 1] = lam[i - 1]
131         A[i, i] = 2.0
132         A[i, i + 1] = muu[i - 1]
133         b[i] = d[i - 1]
134
135     return np.linalg.solve(A, b), {}
136
137     h1 = h[0]
138     h2 = h[1]
139     hn1 = h[-2]
140     hn = h[-1]
141
142     alpha3 = 2.0 * ((h2**2 / h1) * (f[1] - f[0]) - (h1**2 /
           h2) * (f[2] - f[1]))
143     beta3 = 2.0 * ((hn**2 / hn1) * (f[-2] - f[-3]) - (hn1**2
           / hn) * (f[-1] - f[-2]))
144
145     d1_tilde = d[0] - alpha3 / (h2 * (h1 + h2))
146     dn_tilde = d[-1] + beta3 / (hn1 * (hn1 + hn))
147
148     B = np.zeros((n - 1, n - 1), dtype=float)
149     c = np.zeros(n - 1, dtype=float)
150     B[0, 0] = 1.0 + h1 / h2
151     B[0, 1] = h1 / h2

```

```

152     c[0] = d1_tilde
153
154     for i in range(2, n - 1):
155         rida = i - 1
156         B[rida, rida - 1] = lam[i - 1]
157         B[rida, rida] = 2.0
158         B[rida, rida + 1] = muu[i - 1]
159         c[rida] = d[i - 1]
160
161     B[-1, -2] = hn / hn1
162     B[-1, -1] = 1.0 + hn / hn1
163     c[-1] = dn_tilde
164
165     sisemised = np.linalg.solve(B, c)
166     m = np.zeros(n + 1, dtype=float)
167     m[1:n] = sisemised
168     m[0] = alpha3 / h2**2 - ((h2**2 - h1**2) / h2**2) * m[1]
169         + (h1**2 / h2**2) * m[2]
170
171     m[-1] = (hn**2 / hn1**2) * m[-3] + ((hn**2 - hn1**2) /
172         hn1**2) * m[-2] - beta3 / hn1**2
173
174     lisa = {
175         "alpha3": alpha3,
176         "beta3": beta3,
177         "d1_tilde": d1_tilde,
178         "dn_tilde": dn_tilde,
179     }
180     return m, lisa
181
182 def arvuta_splain_solmedest(x, f, tingimus):
183     n = len(x) - 1
184     h = np.diff(x)

```

```

183     lam = h[1:] / (h[:-1] + h[1:])
184     muu = h[:-1] / (h[:-1] + h[1:])
185     d = 3.0 * lam * (f[1:-1] - f[:-2]) / h[:-1] + 3.0 * muu *
        (f[2:] - f[1:-1]) / h[1:]
186
187     m, lisa = lahenda_m(f, h, lam, muu, d, tingimus)
188
189     koef = np.zeros((n, 4), dtype=float)
190     for i in range(n):
191         delta = f[i + 1] - f[i]
192         koef[i, 0] = f[i]
193         koef[i, 1] = m[i]
194         koef[i, 2] = (3.0 * delta - h[i] * (2.0 * m[i] + m[i
            + 1])) / h[i] ** 2
195         koef[i, 3] = (h[i] * (m[i] + m[i + 1]) - 2.0 * delta)
            / h[i] ** 3
196
197     return {
198         "a": float(x[0]),
199         "b": float(x[-1]),
200         "n": n,
201         "x": x,
202         "f": f,
203         "h": h,
204         "lam": lam,
205         "muu": muu,
206         "d": d,
207         "m": m,
208         "koef": koef,
209         "tingimus": tingimus,
210         "lisa": lisa,
211     }
212

```

```

213
214 def arvuta_splain_funktsioonist(avaldis, a, b, n, tingimus):
215     x = np.linspace(a, b, n + 1)
216     f = arvuta_funktsioon(avaldis, x)
217     andmed = arvuta_splain_solmedest(x, f, tingimus)
218     andmed["avaldis"] = avaldis
219     return andmed
220
221
222 def splain_vaartus(xv, x_solmed, koef):
223     indeks = np.searchsorted(x_solmed, xv, side="right") - 1
224     indeks = np.clip(indeks, 0, len(x_solmed) - 2)
225     dx = xv - x_solmed[indeks]
226     return (
227         koef[indeks, 0]
228         + koef[indeks, 1] * dx
229         + koef[indeks, 2] * dx**2
230         + koef[indeks, 3] * dx**3
231     )
232
233
234 def koosta_veapunktid(x_solmed, punkte_osaloigus=
PUNKTE_OSALOIGUS):
235     osaloigud = []
236     for algus, lopp in zip(x_solmed[:-1], x_solmed[1:]):
237         samm = (lopp - algus) / punkte_osaloigus
238         osaloigud.append(algus + samm * np.arange(1,
punkte_osaloigus + 1, dtype=float))
239     return np.concatenate(osaloigud)
240
241
242 def arvuta_diskreetne_viga(andmed, avaldis, punkte_osaloigus=
PUNKTE_OSALOIGUS):

```

```

243     veapunktid = koosta_veapunktid(andmed["x"],
244                                     punkte_osaloigus)
245     splain = splain_vaartus(veapunktid, andmed["x"], andmed["
246                             koef"])
247     tapsed = arvuta_funktsioon(avaldis, veapunktid)
248     return float(np.max(np.abs(splain - tapsed)))
249
250 def arvuta_veauuring(avaldis, a, b, tingimus, n_vaartused):
251     tulemused = []
252     eelmine_epsilon = None
253
254     for n in n_vaartused:
255         andmed = arvuta_splain_funktsioonist(avaldis, a, b, n
256                                             , tingimus)
257         epsilon_n = arvuta_diskreetne_viga(andmed, avaldis)
258         theta_n = None
259         if eelmine_epsilon is not None and epsilon_n != 0.0:
260             theta_n = eelmine_epsilon / epsilon_n
261
262         eelmine_epsilon = epsilon_n
263         tulemused.append(
264             {
265                 "n": n,
266                 "h": float(andmed["h"][0]) if np.allclose(
267                     andmed["h"], andmed["h"][0]) else None,
268                 "epsilon_n": epsilon_n,
269                 "theta_n": theta_n,
270             }
271         )
272
273     return tulemused

```

```

272
273 def vorminda(x):
274     return f"{float(x):.12g}"
275
276
277 def salvesta_tulemused(andmed):
278     kaust = Path(__file__).resolve().parent / "valjund" /
        datetime.now().strftime("%Y%m%d_%H%M%S")
279     kaust.mkdir(parents=True, exist_ok=True)
280
281     txt_fail = kaust / "tulemused.txt"
282     png_fail = kaust / "graafik.png"
283
284     read = []
285     read.append("Kuupsplaini arvutus")
286     read.append("=====")
287     read.append(f"Funktsioon: {andmed['avaldis']}")
288     read.append(f"Vahemik: [{vorminda(andmed['a'])}, {
        vorminda(andmed['b'])}]")
289     read.append(f"Alamvahemike arv n: {andmed['n']}")
290     if np.allclose(andmed["h"], andmed["h"][0]):
291         read.append(f"Vorgu samm h = {vorminda(andmed['h
        '][0])}")
292     else:
293         read.append("Sammud h_i:")
294         for i in range(andmed["n"]):
295             read.append(f"h_{i+1} = {vorminda(andmed['h'][i])
                }")
296     read.append("")
297     read.append("Solmed ja vaartused")
298     read.append("-----")
299     for i in range(andmed["n"] + 1):
300         read.append(

```

```

301         f"x_{i} = {vorminda(andmed['x'][i])}, "
302         f"f_{i} = {vorminda(andmed['f'][i])}, "
303         f"m_{i} = {vorminda(andmed['m'][i])}"
304     )
305
306 read.append("")
307 read.append("Sisevaartused lambda_i, mu_i, d_i")
308 read.append("-----")
309 for i in range(andmed["n"] - 1):
310     read.append(
311         f"i = {i+1}: "
312         f"lambda_{i+1} = {vorminda(andmed['lam'][i])}, "
313         f"mu_{i+1} = {vorminda(andmed['muu'][i])}, "
314         f"d_{i+1} = {vorminda(andmed['d'][i])}"
315     )
316
317 read.append("")
318 read.append("Kuuppolunoomide kordajad")
319 read.append("-----")
320 for i in range(andmed["n"]):
321     c0, c1, c2, c3 = andmed["koef"][i]
322     read.append(
323         f"[x_{i}, x_{i+1}] : "
324         f"c0 = {vorminda(c0)}, "
325         f"c1 = {vorminda(c1)}, "
326         f"c2 = {vorminda(c2)}, "
327         f"c3 = {vorminda(c3)}"
328     )
329
330 if andmed["tingimus"]["tyyp"] == 3:
331     read.append("")
332     read.append("Kolmanda tuuletise tingimuse  
lisavaartused")

```

```

333     read.append("-----")
334     read.append(f"alpha_3 = {vorminda(andmed['lisa']['alpha3'])}")
335     read.append(f"beta_3 = {vorminda(andmed['lisa']['beta3'])}")
336     read.append(f"d_tilde_1 = {vorminda(andmed['lisa']['d1_tilde'])}")
337     read.append(f"d_tilde_(n-1) = {vorminda(andmed['lisa']['dn_tilde'])}")
338
339     read.append("")
340     n_loend = ", ".join(str(n) for n in andmed["veauuringu_n_vaartused"])
341     read.append(f"Veauuring fikseeritud vörkudel: {n_loend}")
342     read.append("-" * len(read[-1]))
343     read.append("n | h | epsilon_n | Theta_n")
344     for rida in andmed["veauuring"]:
345         theta = "--" if rida["theta_n"] is None else vorminda(rida["theta_n"])
346         h_vaartus = "--" if rida["h"] is None else vorminda(rida["h"])
347         read.append(
348             f"{rida['n']} | {h_vaartus} | {vorminda(rida['epsilon_n'])} | {theta}"
349         )
350
351     txt_fail.write_text("\n".join(read), encoding="utf-8")
352
353     xp = np.linspace(andmed["a"], andmed["b"], 1000)
354     sp = splain_vaartus(xp, andmed["x"], andmed["koef"])
355
356     plt.figure(figsize=(10, 6))

```

```

357     yp = arvuta_funktsioon(andmed["avaldis"], xp)
358     plt.plot(xp, yp, "k:", linewidth=2, label="Antud
        funktsioon")
359     plt.plot(xp, sp, "b-", linewidth=2, label="Kuupsplain")
360     plt.plot(andmed["x"], andmed["f"], "ro", label="Sõlmed")
361     plt.grid(True, alpha=0.3)
362     plt.xlabel("x")
363     plt.ylabel("y")
364     plt.legend()
365     plt.tight_layout()
366     plt.savefig(png_fail, dpi=200)
367     plt.close()
368
369     return kaust, txt_fail, png_fail
370
371
372 def prindi_luhidalt(andmed, kaust, txt_fail, png_fail):
373     print()
374     print("Arvutus valmis.")
375     print(f"Valjundkaust: {kaust}")
376     print()
377     print("Peamised vaartused:")
378     for i in range(andmed["n"] + 1):
379         print(
380             f"x_{i} = {vorminda(andmed['x'][i])}, "
381             f"f_{i} = {vorminda(andmed['f'][i])}, "
382             f"m_{i} = {vorminda(andmed['m'][i])}"
383         )
384     print()
385     print(f"epsilon_n = {vorminda(andmed['epsilon_n'])}")
386     print("Theta_n uuring on salvestatud tekstifaili.")
387     print()
388     print(f"Tekstifail: {txt_fail.name}")

```

```

389     print(f"Graafik: {png_fail.name}")
390
391
392 def main():
393     print("programm")
394     print()
395
396     avaldis = input("Sisesta funktsioon f(x), näiteks sin(x)
397                   + x**2: ").strip()
398     a = loe_ujuk("Sisesta vahemiku algus a: ")
399     b = loe_ujuk("Sisesta vahemiku lopp b: ")
400     n = loe_taisarv_min("Sisesta alamvahemike arv n (n >= 3):
401                   ", 3)
402     tingimus = loe_tingimus()
403     algne_n = loe_taisarv_min("Sisesta veauuringu algne n: ",
404                   3)
405     veauuringu_n_arv = loe_taisarv_min("Mitu n väärtust
406                   veauuringusse arvutada? ", 1)
407     veauuringu_n_vaartused = tuple(algne_n * 2**i for i in
408                   range(veauuringu_n_arv))
409
410
411     andmed = arvuta_splain_funktsioonist(avaldis, a, b, n,
412                   tingimus)
413     andmed["epsilon_n"] = arvuta_diskreetne_viga(andmed,
414                   andmed["avaldis"])
415     andmed["veauuring"] = arvuta_veauuring(
416                   andmed["avaldis"],
417                   andmed["a"],
418                   andmed["b"],
419                   andmed["tingimus"],
420                   veauuringu_n_vaartused,
421                   )
422     andmed["veauuringu_n_vaartused"] = veauuringu_n_vaartused

```

```
415
416     kaust, txt_fail, png_fail = salvesta_tulemused(andmed)
417
418     prindi_luhidalt(andmed, kaust, txt_fail, png_fail)
419     return 0
420
421
422 if __name__ == "__main__":
423     main()
```

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Tobias Toomas Peterson,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Interpoleerivad kuupsplainid, mille juhendaja on Arvet Pedas, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Tobias Toomas Peterson

17.05.2026