

1 Sissejuhatus

Kursuses käsitletakse klassikalisi numbrilisi meetodeid harilike ja osatuletistega diferentsiaalvõrrandite alg- ja rajatingimustega ülesannete lahendamiseks. Praktikumides ja kodustes ülesannetes tutvustatakse konkreetseid mudeleid reaalelu probleemide lahendamiseks (näiteks ravimite manustamine, orbitaalliikumine, populatsioon, elektroonilised jahutussüsteemid, laine levimine, difusioonivõrrandite rakendused).

Kursuse lõpuhinde määramiseks summeeritakse **kodutöödest/praktikumidest (kuni 50 punkti)**, seminaride aktiivsest osavõtust (kuni 40 punkti) ja loengust (kuni 10 punkti) saadud punktid.

Praktikumi punktid jagunevad järgmiselt.

	Kokku	Mitu	a' punktid
Arvestuslikud ülesanded	42	7	6
Praktikumides osalemine	8	12	0.5-1
Kokku	50		

1.1 Pakett numpy

Arvutipraktikumis kasutame vabavaralist Python tarkvara ja selle pakette (oma kodutöid saab teha ka muus programmis), mis annab samal ajal võimaluse arendada programmeerimisoskusi endale kasulikus suunas.

Programmi Python lihtsamate tõdede kordamiseks on veebis saadaval palju erinevaid materjale, näiteks eesti keeles <http://progeopik.cs.ut.ee/> või inglise keeles <https://docs.python.org/3/tutorial/>.

Meie kursuses kasutame laialdaselt Pythoni paketti **numpy**, selle lühitutvustus on aadressil <https://docs.scipy.org/doc/numpy/user/quickstart.html>.

NB! Selles kursuses me üldiselt ei kasuta Pythoni paketti **math**.

Toome mõned lihtsad näited paketi **numpy** kasutamisest.

```
import numpy as np

def f(x):
    # defineerib ühe muutuja funktsiooni f
    return np.exp(-x) # tagastab e astmes -x
```

```

a = np.linspace(-1, 1, 5)      #loob vektori sõlmedest -1+i*2/4
                                #(jagab lõigu [-1,1] neljaks)
b = np.zeros(len(a))          #loob nullidest koosneva vektori,
                                #mille pikkus on võrdne vektori a pikkusega

a[1:-1] #tagastab vektori a elemendid alates teisest kuni eelviimaseni

a[0] #tagastab vektori a esimese elemendi

f(a)    #rakendab funktsiooni f vektori a kõikidele elementidele
        #(tulemus on vektor)

np.zeros( (3,5) ) #loob nullidest koosneva 3*4 maatriksi

np.ones( (3,5) ) #loob ühtedest koosneva 3*4 maatriksi

A = np.ones( (3,5) )

A*a #maatriksite A ja a korrutis elementide kaupa

u = np.eye(2) #loob teist järku ühikmaatriksi
j = np.array([[0.0 , -1.0], [1.0 , 0.0]])

k = np.dot (j , j) #maatriksite j ja j korrutis

```

1.2 Graafikud Pythonis

Graafikute loomiseks on Pythonis mitmeid erinevaid võimalusi, meie kasutame üldiselt paketti Matplotlib (vt <https://matplotlib.org/users/index.html>).

Ülesanne Joonestada funktsiooni $\cos$ graafiku lõigus $[-4, 4]$.

```

import numpy as np
import matplotlib.pyplot as plt

def h(t):
    return np.cos(t)

t=np.linspace(-4, 4, 41) #loome argumentide vektori

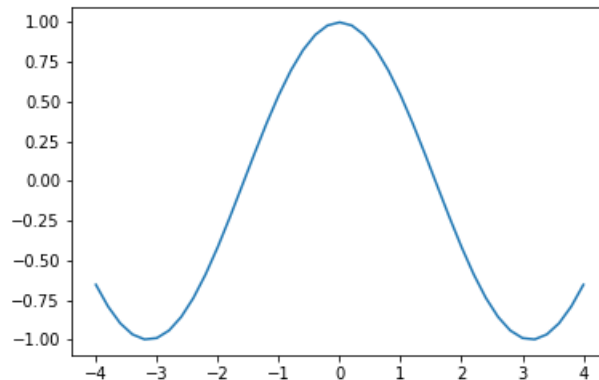
```

```

y=h(t) #rakendame funktsiooni h vektorile t

plt.plot(t,y) #joonistab vektori y väärtused vektori t väärtuste vastu
plt.show() #näitab joonist

```



Ülesanne Kujutada funktsioonid $f_1(t) = t^2e^{-t^2}$ ja $f_2(t) = t^4e^{-t^2}$ lõigus $[0, 3]$ ühel graafikul.

Tulemus salvestada faili **Joonis1.png**.

```

import numpy as np
import matplotlib.pyplot as plt

def f1(t): #defineerib funktsiooni f1
    return t**2*np.exp(-t**2)

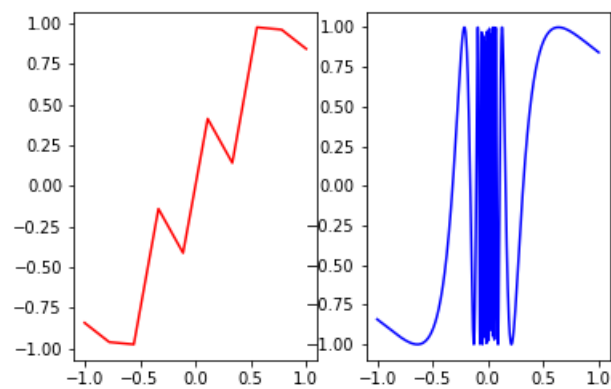
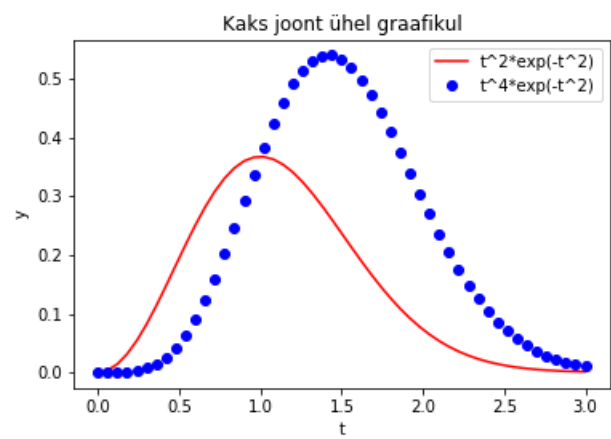
def f2(t): #defineerib funktsiooni f2
    return t**2*f1(t)

t = np.linspace(0, 3, 51) #loob sõlmed
y1 = f1(t) #funktsiooni f1 väärtused sõlmedes
y2 = f2(t) #funktsiooni f2 väärtused sõlmedes

plt.plot(t, y1, 'r-') #f1 graafik, joonistatakse punane (r) joon (-)
plt.plot(t, y2, 'bo') #f2 graafik, joonistatakse sinised (b) ringid (o)
plt.xlabel('t') #x-telg
plt.ylabel('y') #y-telg
plt.legend(['t^2*exp(-t^2)', 't^4*exp(-t^2)']) #joonte tähendused
plt.title('Kaks_joont_ühel_graafikul') #graafiku nimi
plt.savefig('Joonis1.png') #salvestab graafiku faili Joonis1.png

```

```
plt.show()
```



2 Newton'i jahtumisseadus

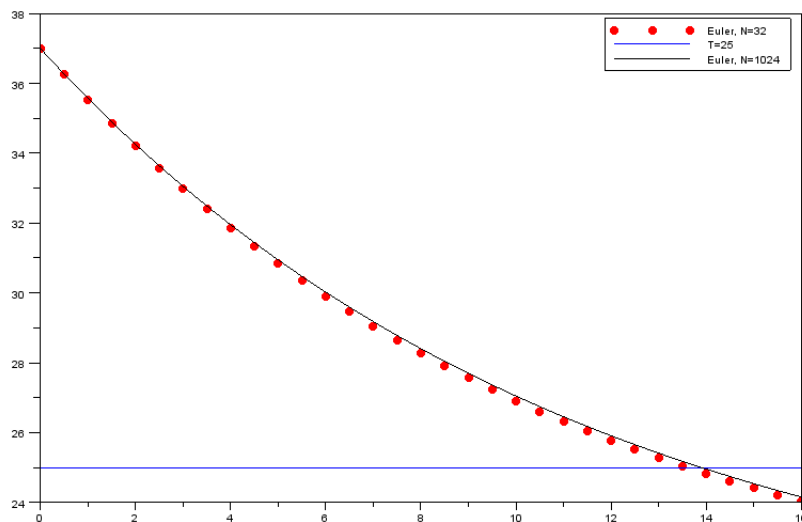
Esimene seadus kehade jahtumise kohta konvektsiooni teel avaldati Isaac Newton'i poolt 28. mail 1701, mis lihtsustatult kõlas

„keha soojushulga muutumise kiirus keha pinnaühiku kohta on võrdeline keha pinna ja ümbritseva keskkonna temperatuuride vahega.“

Newton'i sõnastatud seadusest saab tuletada algtingimusega ülesande

$$\begin{cases} T'(t) = -\beta \cdot (T(t) - T_*) \\ T(0) = T_0 \end{cases}, \quad t \geq 0. \quad (2.1)$$

- otsitav $T(t)$ on keha temperatuur ajahetkel t , $T(0)$ on algtemperatuur.
- T_* on ajas muutumatu väliskeskkonna temperatuur.
- $\beta > 0$ on süsteemi iseloomustav konstant (ühikuks on $1/s$).



Märkus 2.1

Märgime, et kui välistemperatuur $T_* < T_0$, siis $T(t) - T_* > 0$ ja temperatuuri muutumise kiirus $T'(t)$ on negatiivne ning keha jahtub. Kui välistemperatuur $T_* > T_0$, siis $T(t) - T_* < 0$ ja temperatuuri muutumise kiirus $T'(t)$ on positiivne ning keha soojeneb.

Märkus 2.2

Newton'i jahtumisseaduse ja täiustatud mittelineaarsete mudelite kohta võib soovi korral lugeda materjalidest [1]-[6].

2.1 Algtingimusega ülesande lahendamine Euler'i meetodiga

Lahendame algtingimusega ülesannet

$$y' = f(t, y), \quad y(a) = T_0.$$

1. Esiteks on kasulik defineerida kahe muutuja funktsioon f .

```
def f(t, y):  
    return # siia tuleb vastav avaldis t ja y kaudu.
```

2. Valime osalõigu $[a, b]$, milles ülesannet lahendame.
3. Valime osalõikude arvu $N \in \mathbb{N}$, moodustame ühtlaselt paiknevad sõlmed

paberil	arvutis
$a = t_0 < t_1 < \dots < t_N = b$	<code>t = np.linspace(a , b , N+1)</code>

4. Otsime lähislahendit $y_0, \dots, y_N$ vastavalt sõlmedes $t_0, \dots, t_N$. Selleks loome arvutis nullidega täidetud vektori

paberil	arvutis
$y_0 = y_1 = \dots = y_N = 0$ $y_0 = T_0$	<code>y = np.zeros(len(t))</code> <code>y(0)=T0</code>

5. Leiame ülejäänud lähislahendid $y_1, \dots, y_N$ Euler'i meetodiga

$$y_{i+1} = y_i + h \cdot f(t_i, y_i), \quad i = 0, \dots, N-1.$$

6. Graafiku joonestamiseks võime kasutada käsku

```
plot( t , y , "r." ) # vektor vastu vektorit, värv on punane  
                        # ja graafiku tüübiks on punktid
```

Ülesanne 2.1

Spetsialist-kriminalist Sæde leiab Torni pagariäri omaniku mõrvatuna laoruumist kell 16 : 00. Uurimise käigus tekib hüpotees, et omanik mõrvati u. kell 05 : 35 varahommikul. Antud kellaajast järeldab süüdistaja, et peamine kahtlusallane on postiljon Petškin, kes tõi pagariärisse hommikused lehed alati kell 05 : 30 ja keda eelmisel päikselisel hommikul pikasabaline koheva karvaga pagari koer olla pahemast kannast näksanud.

Kohtuvaidlusesse kaasatakse ka teid kui spetsialisti ja palutakse versiooni kontrollida matemaatiliselt. Te saate teada, et laoruumi temperatuuri hoitakse stabiilselt 19 kraadi juures, u. kell 16 : 00 mõõdeti (piisavalt täpselt) ohvri temperatuuriks 26 kraadi, kusjuures pagar olla eelmisel päeval rõõmus ja tervise juures (seega $T_0 \approx 37$ kraadi).

Kommentaarid

1. Parameeter $\beta = 0.089$ (1/tund), selle kohta võib hiljem lugeda abimaterjalidest.
2. Leidke ülesande (2.1) lähislahend Euler'i meetodiga osalõikude arvu $N = 16$ ja $N = 512$ korral.
3. Mõrvaaja määrame selle järgi, millises punktis t temperatuur $T(t)$ on kõige lähemal 26 kraadile (vastus ise peaks olema antud paariminutilise täpsusega). Kõige lihtsam on kasutada graafiku abi.
4. Kui suur on ajavahe lahendite $N = 16$ ja $N = 512$ korral?
5. Viige vajalikud andmed ühele graafikule. Püüdke saada sarnane töölehel olevaga või kui oskate, siis veel paremini vormistatud graafik. piisavalt lähedal ja vastus ei muutu enam oluliselt)?

Viited

- [1] Y. A. Cengel. Introduction to Thermodynamics and Heat Transfer. USA, McGraw-Hill, 2008.
- [2] C. Henssge, B. Madea. Estimation of the time since death in the early post-mortem period. Forensic Science International 144, 167-175, 2004.
- [3] G. S. W. de Saram, G. Webster, N. Kathirgamatamby. Post-mortem temperature and the time of death. 46 J. Crim. L. Criminology & Police Sci. 562, 1955-1956.
- [4] J. Sedlacek, Jiri Dolejsi. A physical description of coffee cooling in a pot.
- [5] C. T. O'Sullivan. Newton's law of cooling - A critical assessment. Am. J. Phys. 58 (10), 1990.
- [6] M. Vollmer. Newton's law of cooling revisited. Eur. J. Phys. 30: 1063-1084, 2009.