



E-kursuse "**Numbrilised meetodid**" (MTMM.00.005)  
materjalid

Aine maht 6 EAP

**Peep Miidla (Tartu Ülikool), 2011**

## Teemad ja üldised õpijuhised

### → **Tarkvarapaketi MATLAB tutvustus.**

Üldine tutvumine programmpaketiga **MATLAB**. Käivitage pakett ja seadistage keskkond ning töölaud (*Desktop*). Seadistage töökaust (*Current Directory*). Tutvuge keskkonnas Moodle (<https://moodle.ut.ee/>) olevate materjalidega. Avage abiaken (*Help*) ja leidke viited ning juhiseid etteantud märksõnade ja sisseehitatud funktsioonide kasutamise kohta. Tehke kaasa harjutusülesanded. Otsige abimaterjale veebist. Paketi veebileht <http://www.mathworks.com>.

### → **Harilik iteratsioonimeetod võrrandi lahendamiseks.**

Võrrandite lahendamisel on täpsete meetodite kõrval tähtsaimad iteratsioonimeetodid ehk järjestikuste lähendite meetodid. Harilik iteratsioonimeetod on neist lihtsaima arvutusskeemiga. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded.

### → **Newtoni meetod võrrandi lahendamiseks.**

Newtoni iteratsioonimeetod on hästi koonduv iteratsioonimeetod võrrandite lahendamiseks. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja esitage lahendused Moodle keskkonnas. Kohustusliku ülesande raames tuleb võrrelda iteratsioonimeetodite tulemusi ja paketi **MATLAB** sisseehitatud vahendite poolt leitud lahendeid.

### → **Võrrandisüsteemide lahendamine iteratsioonimeetoditega.**

Vaadeldakse harilik iteratsioonimeetodit ja Newtoni iteratsioonimeetodit üldiste mittelineaarsete võrrandisüsteemide lahendamiseks. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja esitage lahendused Moodle keskkonnas. Õppige ära võrrandisüsteemide lahendamine paketi **MATLAB** vahenditega.

### → **Lineaarsed võrrandisüsteemid.**

Vaadeldakse Gaussi elimineerimismeetodit ja selle modifikatsioone, samuti harilikku ja Seideli iteratsioonimeetodeid lineaarsete süsteemide lahendamiseks. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja

esitage lahendused Moodle keskkonnas. Õppige ära lineaarsete süsteemide lahendamine paketi **MATLAB** vahenditega.

→ **Funktsioonide interpoleerimine.**

Vaadeldakse funktsioonide interpoleerimise ülesannet ja Lagrange'i ning Newtoni interpolatsioonivalemeid selle ülesande lahendamiseks. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja esitage lahendused Moodle keskkonnas. Õppige ära funktsioonide interpoleerimise ülesande lahendamine paketi **MATLAB** vahenditega.

→ **Numbriline diferentseerimine.**

Vaadeldakse funktsioonide numbrilise diferentseerimise meetodeid selle ülesande lahendamiseks. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja esitage lahendused Moodle keskkonnas. Õppige ära funktsioonide numbrilise diferentseerimise ülesande lahendamine paketi **MATLAB** vahenditega.

→ **Vähimruutude meetod.**

Vaadeldakse funktsioonide lähendamise ülesannet ja vähimruutude meetodit selle ülesande lahendamiseks. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja esitage lahendused Moodle keskkonnas. Õppige ära funktsioonide lähendamise ülesande lahendamine paketi **MATLAB** vahenditega.

→ **Kvadratuurvalemid ja Runge meetodi kasutamine.**

Vaadeldakse funktsioonide numbrilise integreerimise ülesannet ja kvadratuurvalemeid selle ülesande lahendamiseks. Käsitletakse trapetsvalemit, Simpsoni valemit, ja ristkülikvalemit. Vaadeldakse kvadratuurvalemite sammu valimise meetodit – Runge võtet. Tutvuge õppematerjalidega ja tehke kaasa harjutusülesanded. Lahendage kohustuslikud ülesanded ja esitage lahendused Moodle keskkonnas. Õppige ära funktsioonide numbrilise integreerimise ülesande lahendamine paketi **MATLAB** vahenditega.

Õppematerjale.

→ E. Tamme, L. Võhandu, L. Luht, Arvutusmeetodid I. Tln, 1986.

→ R. Plato, Concise Numerical Mathematics. 2003.

→ Otsige veebist GOOGLE'i abil: „Numerical Methods“.

## Näpunäiteid slaidiesitluseks

Allikas: Internet.

### Ettekandeks valmistumine

- Leia aega erinevate materjalidega tutvumiseks. Kontrolli, et info oleks värske.
- Varu aega ettevalmistamiseks! 10 minutilise ettekande tegemiseks arvesta 10 tundi.
- Koosta ettekande kava ( võtmepunktid)
- Koosta enne kõne ja siis esitlus selle näitlikustamiseks. Kasuta lihtsaid lauseid.
- Ettekandes peab olema sissejuhatus, põhiosa ja kokkuvõte.
- Rõhutada tuleb võtmesõnu ja kandvat ideed.
- Ettekanne peab olema asjakohane ja lihtsasti jälgitav.
- Raba teadmistega, uue infoga, kasuta oma isikuomadusi.
- Kuulaja mäletab alati esinemist kauem kui kõne üksikasju.
- Arvesta publikuga (kuulajaskonna suurus, vanus, teemaga kursis olek, mis on kuulajaskonnal ühist omavahel, kultuuritaust, võimalikud eelarvamused, isiklikest tuttavatest kuulajad).
- Arvesta sellega, milline on ruum, kus ettekanne toimub ja seal kasutatavate lisavõimalustega (ruumi suurus, valgustatus, tehniliste vahendite kasutamise võimalus, kasutatav tarkvara jne).
- Otsusta, milliseid audio-visuaalseid vahendeid kasutad.
- Enne esinemist pea vähemalt kolm korda sama ettekanne endale. Vajadusel vii sisse parandused.
- Õpi kõne pähe, kuid märkmed olgu alati käepärast. Diagrammide kasutamisel on vajalik, et oleks kaasas algandmed, et vajadusel täpsustada andmeid.
- Tee varukoopiad esitlusest. Ava enne esinemist slaidiseeria ja veendu, et kõik slaidid on korrektsed. Mõtle välja varuvariant juhuks, kui millegipärast tehnika alt veab.
- Mõtle läbi küsimustele vastamine, eeldatavad küsimused ja vastused neile, samuti aeg, mida võid selleks lubada. Vasta korraga ainult ühele küsimusele. Vajadusel vasta küsijale hiljem individuaalselt. Kriitika ei ole üldjuhul suunatud konkreetselt sulle isiklikult. Jää alati rahulikuks, ära lasku vaidlusesse, vaid püüa alati kuulajale vastata lugupidavalt.

### Erinevate vahendite ja materjalide kasutamine

- Trükitud materjali jaga siis, kui ettekande käigus jagad selgitusi selle kohta. Mõttekas on materjal jagada enne ettekannet või saata e-mailiga varem kuulajatele.
- Paber- ja kriiditahvli kasuta peaausjalikult väikesele kuulajaskonnale esinedes. Kasutamisel pea silmas, et ka viimases reas istuja peab nägema sinna kirjutatut.
- Grafoprojektorit kasuta näiteks diagrammide või tabelite demonstreerimiseks. Ka siin on oluline, et kõik kuulajad näeksid sealt andmeid lugeda. Slaididele viitamiseks on vajalik pliiats või kaardikepp.
- Mikrofon ja võimendid on hädavajalikud suures saalis esinedes.

- Arvutigraafika kasutamine on võimalik nii väikses kui ka suures saalis. Suures saalis jälgi, et projektorit oleks võimalik ekraanist piisavalt kaugelt paigutada.
- Kui pole võimalust kasutada tehnilist tuge, siis tee kindlaks, kas kõik vajalikud juhtmed on kaasas, tavaliselt on vaja ka pikendusjuhet. Varu vähemalt 15 minutit kogu tehnika ülespanekuks ja katsetamiseks.
- Slaidiesitluse eelised: ettekandes muutuste tegemine on võimalik ka viimasel minutil; slaidivahetused ja animatsioonid aitavad tempot kontrollida; graafikud ja pildid aitavad illustreerida juttu ja muuta see kuulajatele kergemini mõistetavaks; võimalus on kasutada videot ja heli.
- Slaidide sisu peab olema kvaliteetne, ilma ühegi kirjaveata.
- Ära kuhja slaidile teksti üle 8 rea.
- Ühes reas ei tohiks olla rohkem kui 6 sõna.
- Slaidi pinnast kasuta ära umbes 60 %.
- Kiri slaidil peaks olema suurusega 24 punkti.
- Slaidide kujundus võiks üldjuhul olla ühtne kogu esitluse osas.
- Ettekannet illustreeri näiteülesannete lahendustega paketi SCILAB või MATLAB abil; akende vahetuseks kasuta klahvikombinatsiooni Alt+Tab.

### **Ettekande pidamine**

- Ära alusta enne, kui kogu auditoorium on vaikne ja valmis sind kuulama. Vajadusel ütle sissejuhatavad sõnad veidi kõvema häälega.
- Kuulaja suudab ekraanil olevat teksti lugeda oluliselt kiiremini, kui sina rääkida. Ära korda kirjapandut, vaid kommenteeri slaidi sisu.
- Kasuta erinevat hääletooni, miimikat.
- Vaata auditooriumi poole. Jälgi kuulajaid ja püüa reageerida auditooriumile.
- Pea kinni ajast.

## **Juhend kohustuslike tööde vormistamiseks**

### **1. Üldised tingimused.**

*Arvestuse saamiseks* tuleb lahendada kõik **kohustuslikud ülesanded**. Viited **kohustuslike ülesannete** tekstidele on lisatud nende teemade akendesse, millede puhul need on ette nähtud ning lahendused tuleb esitada (keskkonnas MOODLE üles laadida) sama teema **ülesannete** tegevuse kaudu. Ülesannete tekste tuleb tähelepanelikult lugeda ning täita seal toodud nõuded. Nimelt tuleb mõnikord lahendada kõik alamülesanded, mõnel juhul võib teha valiku etteantute seast või ise ülesande leida. Kõikide teemade juures **kohustuslikke ülesandeid** ei ole.

**Kohustuslikke ülesandeid** hinnatakse arvestatud-mittearvestatud skaalal, iga üliõpilane peab kõikide **kohustuslike ülesannete** lahenduste eest saama hinde „arvestatud“. On lubatud lahenduste korduvesitamine. Kõikide **kohustuslike ülesannete** lahendused annavad kokku 80% lõpliku arvestuse saamiseks vajalikust, ülejäänud saadakse testist teoreetilise osa kohta.

### **2. Ülesannete lahenduste vormistamine ja esitamine.**

*Kohustuslikud ülesanded* tuleb lahendada ja vormistada programmpaketi **MATLAB** abil. Lahendused tuleb vormistada m-failina paketi **MATLAB** vastava toimetite abil. Faili nimeks tuleb panna üliõpilase perekonnanimi, seejärel teema number.

Näiteks:

Tamm2.m % See on lahendaja Tamm teema nr. 2 kohustusliku ülesande lahendamise kohta käiv m-fail.

Failid peavad olema vormistatud nii, et neid saab pärast allalaadimist paketi **MATLAB** käsuaknast käivitada. Kohe faili algul tuleb kommentaaridena kirjutada faili käsitsemisjuhise, autori andmed ja kirjeldada täpselt ülesannet, millist lahendatakse. Kommentaarid tipitakse paketi **MATLAB** märgi „%“ järel. Esitatavasse faili tuleb kommentaaridena lisada ka kõik vajalikud seletused ülesande lahenduskäigu kohta. Kui seletus lisatakse mõnes teises formaadis vormindatud failina, tuleb see ka põhifailis märkida.

Lühidalt ja veelkord: **kohustuslike ülesannete** lahendused tuleb varustada täielike kommentaaridega.

Kui ühe teema kohta esitatakse mitu faili, tuleb perenime ja teema numbri järel paigutada allkriips „\_“ ja selle järel alamfaili number. See on vajalik siis, kui kasutatakse m-failina vormistatud funktsiooni.

**Näiteks:**

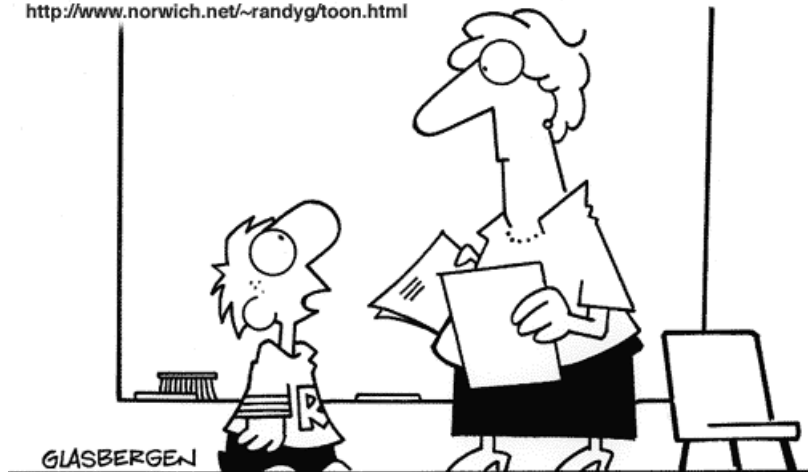
Tamm2\_3.m % See on lahendaja Tamm teema nr. 2 kohta käiv 3. osafail.

Teema **kohustuslike ülesannete** kohta käivad failid tuleb üles laadida vastava teema ülesannete tegevuse raames või foorumis. Kui esitatakse teistsuguses formaadis (mitte m-failidena) materjale, tuleb nende nimetamisel kinni pidada ülaltoodud reeglitest – seda tuleb teha perekonnanime alusel.

## Mängureeglid

- Tuleb lahendada kõik kohustuslikud ülesanded paketi MATLAB abil. Programmid vormistada ja nimetada vastavalt juhendile. Kõik esitatavad failid peavad kommentaaridena sisaldama kuupäeva, millal töö on tehtud.
- Kui on nõutud, tuleb alla laadida MATLAB programmid, need vastavalt kirjeldustele modifitseerida ja samuti esitada. Failinimed alaku autori perekonnanimega, sisus ikka kuupäev ja vajalikud kommentaarid.
- Nõutud tööde esitamise kuupäevadest tuleb kinni pidada.

© 1997 by Randy Glasbergen. E-mail: randyg@norwich.net  
<http://www.norwich.net/~randyg/toon.html>



**“I couldn’t do my homework because my  
computer has a virus and so do all  
my pencils and pens.”**

## Eesmärk ja õpiväljundid

Üliõpilastele esialgsete baasteadmiste andmine tänapäevaste arvutusmeetodite tundmaõppimiseks ja teistes ainetes omandatud oskuste treenimine.

inglise keeles Giving basic knowledge to the students in studying modern numerical methods and training of skills got in other disciplines.

Peale kursuse läbimist üliõpilane

1. tunneb elementaarset vigade teooriat
  2. oskab lihtsamatel juhtudel hinnata arvutamisel tekkivaid vigu
  3. oskab polünoomidega interpoleerida
  4. oskab lahendada mittelineaarseid võrrandeid
  5. oskab lahendada mittelineaarseid võrrandisüsteeme
  6. tunneb lineaarsete süsteemide tegeliku lahendamise meetodeid
  7. oskab numbriliselt diferentseerida
- tunneb lihtsamaid kvadratuurvalemeid ja oskab neid kasutada integraalide ligikaudsel arvutamisel

After passing the course students

1. knows elementary theory of errors
2. can estimate errors occurring in calculation in simplest cases
3. can interpolate with polynomials
4. can solve nonlinear equations
5. can solve nonlinear systems of equations
6. know real methods of solving linear systems
7. can numerically differentiate
8. know simple quadrature formulae and can use them in calculation of integrals

## Sissejuhatus ainesse „Numbrilised meetodid“ Moodle keskkonnas.

Aine MTMM.00.005 „Numbrilised meetodid“ läbimise eest saab üliõpilane 6 ainepunkti (EAP). Oluline rõhk on praktilisel osal, see baseerub rakenduspaketil **MATLAB**. Selle osa omandamiseks vajalik materjal on täielikult kättesaadav Moodle keskkonnas. Kursuse eduka omandamise ja läbimise eelduseks on siiski aktiivne iseseisev töö paketi **MATLAB** sisseehitatud vahendite, abimaterjalide ja demode tundmaõppimisel – ühtki arvutitarkvara ei saa omandada pelgalt õppejõu jutu põhjal. Vastavalt ÕIS-is sätestatule on üliõpilaste iseseisva töö mahuks semestri jooksul 92 tundi. Õpetamine toimub segaõppe (*Blended Learning*) vormis, kus kirjalikud materjalid seletatakse auditooriumis üksikasjalikult lahti.

Kohustuslike ülesannete lahendused tuleb vormistada paketi **MATLAB** failidena, nn m-failidena, mis on vastavalt vormistusjuhendile varustatud piisavate kommentaaridega. Lahendusi kontrollitakse nii, et kõigepealt avatakse fail toimetiaknas, seejärel, kui olulisi süntaksivigu ei avastata, käivitatakse käsuaknast. Lahenduste üleslaadimise tähtajad on näha Moodle keskkonna vastavate vahendite juures. Kohustuslike ülesandeid on kokku 7.

Kursuse teemad:

- **Tarkvarapaketi **MATLAB** tutvustus.**
- **Harilik iteratsioonimeetod võrrandi lahendamiseks.**
- **Newtoni meetod võrrandi lahendamiseks.**  
Kohustuslik ülesanne.
- **Võrrandisüsteemide lahendamine iteratsioonimeetoditega.**  
Kohustuslik ülesanne.
- **Lineaarsed võrrandisüsteemid.**  
Kohustuslik ülesanne.
- **Funktsioonide interpoleerimine.**  
Kohustuslik ülesanne.
- **Numbriline diferentseerimine.**  
Kohustuslik ülesanne.
- **Vähimruutude meetod.**  
Kohustuslik ülesanne.
- **Kvadratuurvalemid ja Runge meetodi kasutamine.**  
Kohustuslik ülesanne.

Uno Hämarik, Liivi 2-417, tel 7375499.

Peep Miidla, Liivi 2-419, tel 7375492.

## Andmeklassid pakettis **MATLAB**

Pakettis **MATLAB** saab töötada 15 fundamentaalse andmetüübiga. Kõikidest võib moodustada massiive või matrikseid.

- Boole'i muutujad – loogilised muutujad väärtustega *true* või *false*.
- Ujukoma arvud ühekordse täpsusega (4 baiti): *single*.
- Ujukoma arvud kahekordse täpsusega (8 baiti, vähim väärtus  $2^{-1022} \approx 2 \cdot 10^{-308}$ , suurim väärtus  $2^{1024} \approx 2 \cdot 10^{308}$ , täpsus  $2^{-52} \approx 10^{-16}$ ): *double*, vaikumisi kasutusel.
- Märgiga täisarvud, 8-bitilised, *int8*.
- Märgiga täisarvud, 16-bitilised, *int16*.
- Märgiga täisarvud, 32-bitilised, *int32*.
- Märgiga täisarvud, 64-bitilised, *int64*.
- Märgita täisarvud, 8-bitilised, *uint8*.
- Märgita täisarvud, 16-bitilised, *uint16*.
- Märgita täisarvud, 32-bitilised, *uint32*.
- Märgita täisarvud, 64-bitilised, *uint64*.
- Funktsioonide viidad, *function\_handle*.
- Tekst, *char*.
- Heterogeensed andmed, nimepõhine, struktuur, *struct*.
- Heterogeensed andmed, indeksipõhine, *cell*.

## Pakett **MATLAB**, harjutusülesandeid

1. Sisestada vabalt valitud elementidega 4x4 maatriks A ning genereerida käsuga **rand** 4x4 maatriks B.

(i) Leida  $A*B$ ,  $A/B$ ,  $A\setminus B$ ,  $A ./B$ ,  $A ./B$ ,  $A \setminus B$ ,  $A^{-1}$ .

(ii) Uurida saadud maatrikseid käskudega **flipud**, **fliplr**, **rot90**, **flipdim**.

(iii) Eraldada leitud maatriksite 2. veerud ning moodustada neist käsuga **cat** uus maatriks.

(iv) Saadud maatriks transponeerida ja jätta sellest välja kaks vähima reasummaga rida.

2. Sisestada kaks nullidest ja ühtedest koosnevat vektorit, mõlemad 10-elementilised. Milliseid võimalusi nende konstrueerimiseks võib leida? Olgu nendeks näiteks u ja v.

(i) Leida **and**(u,v), **or**(u,v), **xor**(u,v), **not**(u), **not**(v). Selgitada tulemusi.

(ii) Leida  $u > v$ ,  $u < v$ ,  $u == v$ ,  $(u > v) | (v < u)$ ,  $(u > v) \& (v < u)$ . Selgitada tulemusi.

(iii) Leida  $u > 3$ ,  $v < 3$ ,  $u((u < 2) | (v \geq 1))$ ,  $v((u < 2) | (u \geq 0))$ . Selgitada tulemusi.

3. Sisestada vabalt valitud vektor pikkusega 15.

(i) Liita vektori igale elemendile 12.

(ii) Liita paarituurvulise indeksiga elementidele 6.

(iii) Leida ruutjuur igast elemendist ja elementide ruutjuurte summa.

(iv) Leida iga elemendi ruut ja nende ruutude keskmine.

4. Moodustada vektor elementidega

$$x_n = (-1)^{n+1}/(2n-1).$$

Alustades vektorist  $n=3$  lisada sellele elemente, kuni vektori pikkuseks on 10.


5. Palun iseseisvalt otsida veebist täiendavaid ülesandeid ja neid lahendada.

## Pakett **MATLAB**, harjutusülesandeid, graafika

1. Genereerida 10-elementiline positiivsete elementidega vektor (näiteks  $u$ ) ja kujutada see joonisel.

2. Leida  $3*u$ ,  $u.^2$ , **sqrt**( $u$ ), **sin**( $u$ ), ... Kujutada need ühel ja samal joonisel.

3. Eelmised joonised (4-6 joonist) paigutada eraldi väljadele samas aknas (**subplot**).

4. Kasutada käskke **xlabel** , **ylabel** , **title**, **clf**, ... Teha samu tegevusi (kirjete paigutamine joonisele) joonise redigeerimisvahendite abil (nupp ).

5. Joonistada parameetriliste funktsioonide pere  $y^2 = x^3 - 3x + c$  kuus esindajat ühele graafikule (2- ja 3- dimensionaalsetele);  $c = 2$ ;  $c = -2$  ;  $c$  väärtus pisut suurem kui 2;  $c$  väärtus pisut väiksem kui 2;  $c$  väärtus pisut suurem kui -2;  $c$  väärtus pisut väiksem kui -2. (Näpunäide: kasutada käskke **mesh** ja **contour**).

6. Joonistada funktsioonide pere  $y^2 = x^2 - x^4$  piirkonnas  $-1.2 < x < 1.2$ ,  $-1 < y < 1$ , kahel viisil: võrguga 51 korda 51; võrguga 100 korda 100. (Näpunäide: kasutada käsku **ezplot** või **contour**).

7. Kasutada ja uurida funktsiooni **cylinder**.

8. Kasutades funktsiooni **subplot** kujutada funktsioone **sin**( $x$ ), **cos**( $x$ ), jt. (omal valikul).

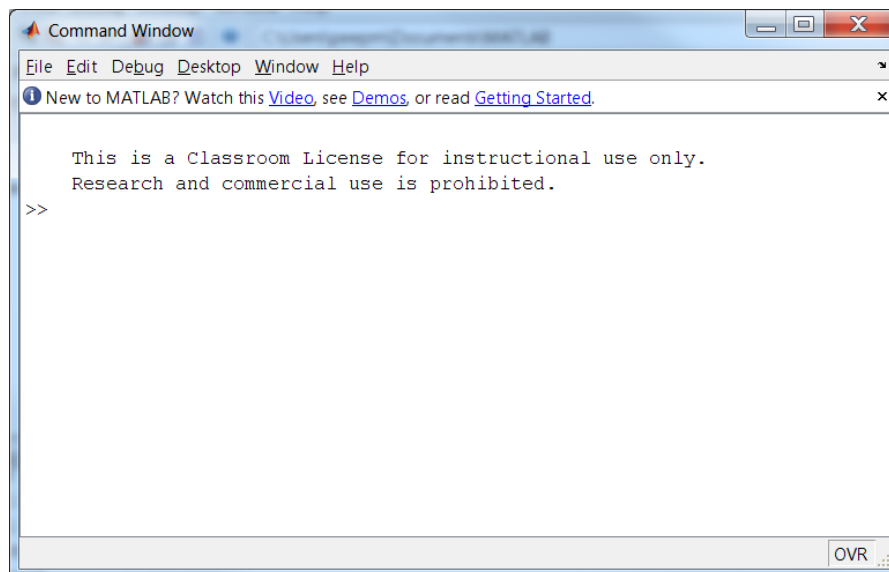
9. Olgu antud kõver parameetrilisel kujul  $(y,z) = (g(t),h(t))$ ,  $a < t < b$  . Pöördpind, mis tekib selle kõvera pöörlemisel ümber  $z$ -telje, on esitatav võrrandiga  $(x,y,z) = (g(t)\cos(\theta), g(t)\sin(\theta), h(t))$ . Uurida graafiliselt järgmistele kõveratele vastavaid pöördpindu. (Kasutada näiteks funktsiooni **plot3**).

(i)  $(y,z) = (t^3, t^2)$ ,  $0 < t < 1$ .

(ii)  $(y,z) = (\cosh(t), \sinh(t))$  ,  $-1.5 < t < 1.5$  . See on osa hüperboolist  $z^2 = y^2 - 1$  . Meeldetuletuseks:  $\cosh(t) = (e^t + e^{-t})/2$  ja  $\sinh(t) = (e^t - e^{-t})/2$  . Kujutada ka hüperboloid  $z^2 = x^2 + y^2 - 1$  .

## Paketi **MATLAB** käsuaken

**Käsuaken**, mille viibalt sisestatakse tegevused (korraldused):



Muutujate nimed ehk identifikaatorid peavad algama tähega, ei tohi sisaldada tühikuid ja täpitähti, samuti mõningaid teisi sümboleid. Parem on nimedes kasutada ainult tähti ja numbreid. Muutujate nimedes on tähed tõstutustundlikud, st. suured ja väikesed tähed eristatakse. Tagasiankurdusnool on kuju muutnud ja asub menüüriba parempoolses otsas.

Käsureale sisestatakse korraldused, sellelt käivitatakse programme ja funktsioone. Valemid tipitakse ühele reale, rea jätkamine toimub kolme punkti „...“ abil. Kommentaarid nii käsureal kui ka programmide real algavad protsendi märgiga „%“.

### **Arvud:**

Täisarvud:        3                    -99

Ujukoma arvud:    9.6397238        1.60210e-20        6.02252e23

Kompleksarvud:  $2+1i$   $-3.14159j$   $3e5i$

Arvutustes 16 kümnendkohta (**double**), tõkked reaalarvudele:  $10^{-308}$  kuni  $10^{+308}$ .

### **Tehted:**

+ liitmine; - lahutamine; \* korrutamine; / jagamine;

\jagamine vasakult (maatriksite korral, vt. edaspidi); ^ astendamine;

' kaaskompleksarv (transponeerib ka maatriksi); () tehete järjekord.

### **Konstandid:**

**pi** = 3.14159265... **i, j** imaginaarühikud, **sqrt(-1)** ;

**eps** - ujukoma arvude suhteline täpsus,  $\epsilon = 2^{-52}$  ;

**realmin** - vähim ujukoma arv,  $2^{-1022}$  ;

**realmax** - suurim ujukoma arv,  $(2 - \epsilon) * 2^{1023}$  ;

**Inf** - lõpmatus; **NaN** - ei-ole-number.

### **Avaldised:**

```
>>roo = (1+sqrt(5))/2
```

```
roo =
```

```
1.6180
```

```
>>a = abs(3+4i)
```

```
a =
```

```
5
```

```
>>z = sqrt(besselk(4/3,rho-i)) % Besseli funktsiooni väärtus.
```

```
z =
```

0.3730+ 0.3214i

```
>> suur = exp(log(realmax))
```

suur =

1.7977e+308

```
>> liigasuur = pi*suur
```

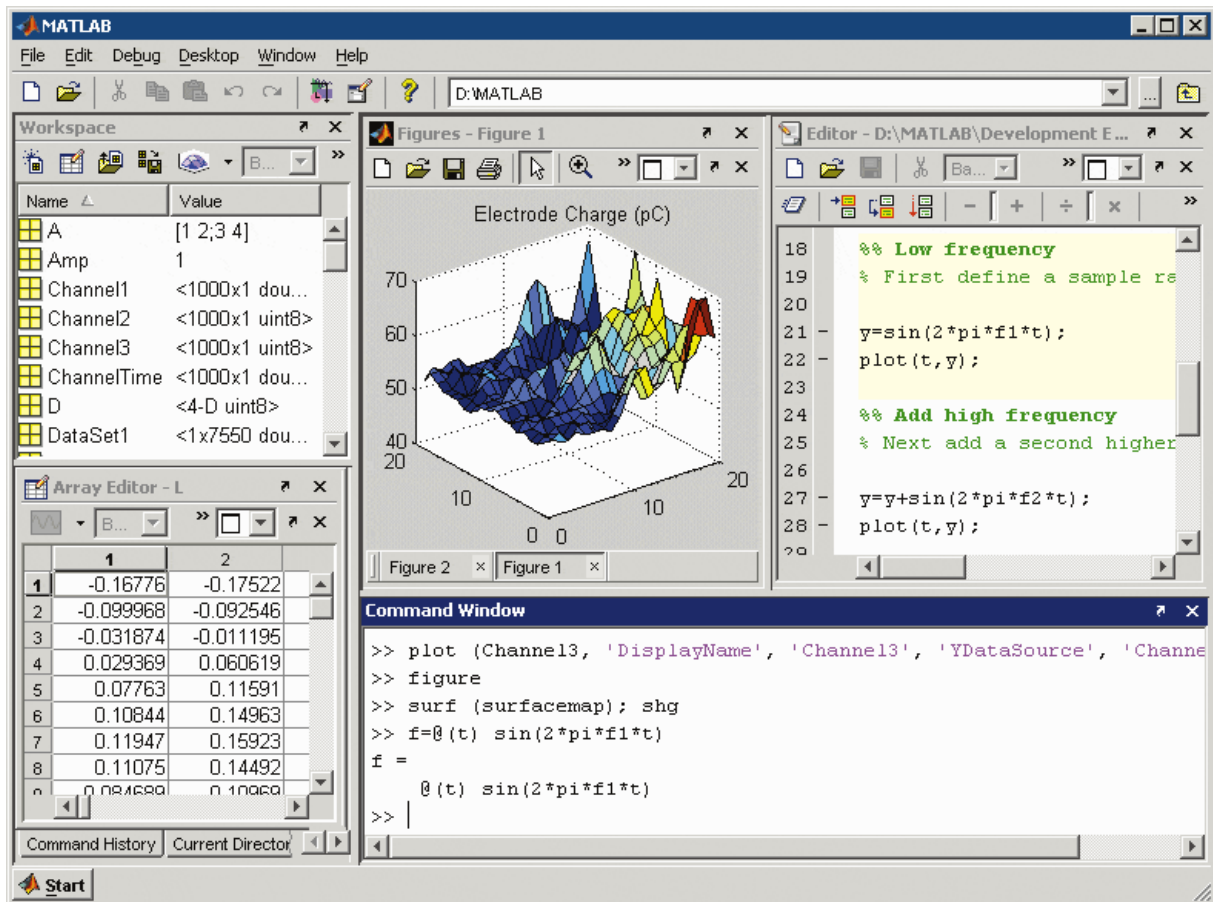
liigasuur =

**Inf**

```
>> b=4; % Kui käsk lõpeb semikooloniga, siis tulemust ei näidata.
```

Paketi MATLAB tähtsad komponendid.

- **Desktop**. Kõik aknad on eraldatavad nupu  abil ja taastatavad  abil.



- **Start**. Paketi MATLAB erinevate komponentide käivitamiseks.

- **Command Window**, käsuaken. Siit käivitatakse paketi MATLAB konstruktsioone (käsud, programmid, funktsioonid).


- **Command History**, käskude ajalugu Käsuaknasse sisestatud korralduste logi.

- **Current Folder**. Näitab töökausta, kuhu salvestatakse ka kasutaja failid ning programmid.

- **Workspace**. Näitab hetkel kasutusel olevaid muutujaid arvuti mälus.

- **Editor**, toimetati, avaneb ka **New M-file** nupuga . Programmide loomiseks ja redigeerimiseks.

- **Help**, abiaken, F1. Asendamatu abimees täpse süntaksi alal. Neli alamakent: **Contents**, **Index**, **Search Results**, **Demos**. Demovahenditega tutvumine on väga oluline.

- **Array Editor**. Muutujate (massiivide) väärtuste kontrollimiseks ja redigeerimiseks.
- **Figure**, jooniseaken. Redigeerimisvahendid avanevad nupu  alt.
- **Toolbox**, spetsialiseeritud lisaprogrammide komplektid vastavate ülesannete lahendamiseks.

[peep.midla@ut.ee](mailto:peep.midla@ut.ee)

## Maatriksid Programmipaketis **MATLAB**

Tähtsaim paketi **MATLAB** poolt toetatav andmetüüp on maatriks.

- [1 2 3] või [1, 2, 3] on reavektor;
- [1; 2; 3] on veeruvektor;
- [1 2; 3 4; 5 6] on  $3 \times 2$  maatriks;
- [1:4] on sama mis [1 2 3 4]; koolon tähendab siin „ühest neljani sammuga 1“;
- [1:0.2:2] on sama mis [1 1.2 1.4 1.6 1.8 2]; koolon tähendab siin „ühest kaheni sammuga 0.2“;
- A' annab transponeeritud maatriksi;
- A(2, 3) on maatriksi A teise rea kolmanda veeru element;
- A(1, :) on maatriksi A esimene rida vektorina;
- A(2, [1 3]) on vektor [ A(2,1), A(2,3) ];
- A([1 2], [3 4]) on maatriksi A alammaatriks [A(1,3) A(1,4);A(2,3) A(2,4)]
- ones(2,3) on ühtedest koosnev  $2 \times 3$  maatriks;
- zeros(2,3) on nullidest koosnev  $2 \times 3$  maatriks;
- eye(2,3) on maatriks mille peadiagonaalil on ühed ja mujal nullid;
- ones(A) on ühtedest koosnev maatriks mille dimensioonid võrduvad maatriksi A omadega;
- A\*B on maatriksite A ja B maatrikskorrutis;
- A.\*B on maatriksite A ja B korrutis vastavate elementide kaupa;
- A+B on maatriksite A ja B summa;
- A-B on maatriksite A ja B vahe;
- 2\*A on maatriks A, mille elementideks on maatriksi A kahekordsed;
- A + 3 liidab maatriksi A igale elemendile arvu 3.
- sum(A) annab maatriksi A elementide summa;
- max(A) annab maatriksi A maksimaalse elemendi;
- sin(A) leiab maatriksi A iga elemendi siinuse;
- inv(A) on maatriksi A pöördmaatriks;
- A/B on maatriks  $A*B^{-1}$  (kui pöördmaatriks  $B^{-1}$  eksisteerib);
- B \A on maatriks  $B^{-1} * A$  (kui pöördmaatriks  $B^{-1}$  eksisteerib);
- A./B on maatriks, mille elementideks on vastavate elementide jagatised;
- B.\A on maatriks, elementideks vastavate elementide pöördjagatised.

Maatriksi elemendid sisestatakse ridade kaupa nurksulgudesse. Reaelemendid eraldatakse üksteisest komade või tühikutega ning read semikoolonitega.

```
-->A=[1 2,3;4,5 6;]
```

```
A =
```

```
1. 2. 3.  
4. 5. 6.
```

Maatrikseid saab kasutada teiste maatriksite elementidena. Sel juhul peavad alammaatriksite ridade arvud ühtima:

```
->C=[A,B]
```

```
C =
```

```
1. 2. 3. 6. 6.  
4. 5. 6. 15. 15.
```

Juhuslike arvudega maatriksi genereerimiseks on pakettis **MATLAB** käsk `rand(m,n)`, kus m on maatriksi ridade ja n veergude arv:

```
-->rand(5,2)
```

```
ans =
```

```
0.5608486 0.2320748  
0.6623569 0.2312237  
0.7263507 0.2164633  
0.1985144 0.8833888  
0.5442573 0.6525135
```

## Programmeerimine pakettis **MATLAB**: Õpijuhised

**Hädavajalikud teadmised.** Peab teadma, kuidas

- luua ja muuta m-faile (skripte ja funktsioone);
- selgitada erinevusi skripti ja funktsiooni vahel;
- kommenteerida programme;
- kasutada käsku disp, et väljastada m-failist andmeid;
- kasutada võrdlusoperaatoreid (<, <=, >, >=, ==, ~=);
- kasutada loogilisi operaatoreid (&, |, ~);
- kirjutada **for** ja **while** tsükleid;
- kirjutada tingimuslikke konstruktsioone: **if...end**, **if...elseif...end**, **if...elseif...else...end**, **switch...case...end**;
- kasutada käsked **break** ja **return**;
- asendada tavalised tsüklid võimalusel vektoriseeritutege;
- enne tsükleid reserveerida mälu massiividele, mis tsüklis kasvavad.
- kirjutada programmi heas stiilis – kommentaarid, tühjad read, taane;
- kavandada programmi ja nende hierarhiat blokk skeemi(de) abil;
- siluda programme (**error**, **warning**, **pause**, **keyboard**);
- kasutada interaktiivseid silumisvahendeid.

## Programmeerimine pakettides **MATLAB** ja **SCILAB**

1. osa

Programmeerimise põhialused pakettides **MATLAB** ja **SCILAB**. Programmid salvestatakse korralduste jadadena, eraldi failidena töökaustades.

*Current Directory* paketis **MATLAB**; seda saab muuta/vahetada aadressirealt.

*Current Directory* paketis **SCILAB**; seda saab muuta/vahetada menüüribalt, **File** -> **Change...**

Programme koostatakse eraldi toimetites (*Editor*).

<b>MATLAB</b> lihtprogramm	<b>SCILAB</b> lihtprogramm
<p>Programmifailid salvestatakse töökaustas (<i>Current Directory</i>) failidena, mille laiendiks on „.m” ja käivitatakse käsuaknast programmi nimega. Kõik muutujad lihtprogrammis on globaalsed, st. kättesaadavad tööpiirkonnas (<i>Workspace</i>).</p> <p>Näiteks, kui töökaustas on programm nimega „programm.m”, siis selle käivitamine toimub käsuaknast nii:</p> <pre>&gt;&gt;programm</pre>	<p>Programmifailid salvestatakse töökaustas (<i>Current Directory</i>) failidena, mille laiendiks on „.sci” või „.sce”; olgu selleks näiteks fail „programm.sci”. See käivitatakse käsuaknast nii:</p> <pre>--&gt;exec("programm.sci")</pre> <p>Lihtprogrammide puhul laienditel vahet pole, see ilmneb alles funktsioonide puhul, mida käsitletakse loengutes.</p>
<p>Kommentaarisid, väga vajalikud märkmed programmides ja ka ülesannete lahendustes, algavad sümboliga „%”:</p> <pre>&gt;&gt; % Järgnevalt käivitame „programm.m”. &gt;&gt;programm % Käivitame „programm.m”. &gt;&gt; %</pre>	<p>Kommentaarisid, väga vajalikud märkmed programmides ja ka ülesannete lahendustes, algavad sümbolitega „/”:</p> <pre>--&gt; // Järgnevalt käivitame "programm.sci". --&gt;exec("programm.sci") // Käivitasime. --&gt; //</pre>

Programmeerimisel on tihti vaja muutujate väärtusi võrrelda. Vastavad võrdlusmärgid on pakettides **MATLAB** ja **SCILAB** ühesugused (tühikud märkide vahel on siin parema loetavuse huvides):

$a = b$  on tõene siis, kui  $a$  ja  $b$  on võrdsed;  
 $a \sim b$  või  $a <> b$  on tõene siis, kui  $a$  ja  $b$  ei ole võrdsed;  
 $a < b$  on tõene siis, kui  $a$  on väiksem kui  $b$ ;  
 $a \leq b$  on tõene siis, kui  $a$  on väiksem muutujast  $b$  või võrdne muutujaga  $b$ ;  
 $a > b$  on tõene siis, kui  $a$  on suurem kui  $b$ ;  
 $a \geq b$  on tõene siis, kui  $a$  on suurem muutujast  $b$  või võrdne muutujaga  $b$ ;

Võrdlustehteid tehakse ka massiivide vahel.

## Programmeerimine pakettis **MATLAB**

2. osa

Programmide puhul on olulisel kohal tsükli organiseerimine arvutustes. Tsüklioperaatoreid on kaks : **for** ja **while**. Pakettis **MATLAB** on käsu **end** kasutamine vajalik tsükli lõpetamiseks. Toome näiteid.

```
for k = 1:4 % See on for - tsükkel.  
disp(k);  
end
```

```
k=1;  
while k<=4 % See on while – tsükkel pakettis MATLAB.  
disp(k)  
k=k+1;  
end
```

Tsükli puhul kasutatakse veel järgmisi juhtimiskorraldusi.

**break** – katkestab tsükli (nii **for** – kui **while** – tsükli) täitmise.

**pause** – katkestab programmi (seejuures ka nii **for** – kui **while** – tsükli) täitmise. Programmi töö jätkub sisestusklahvile vajutamise järel. Kasutatakse programmide silumisel.

Tingimuslikuks direktiiviks on pakettides **if**, selle mõjupiirkonna sees võib kasutada veel käske **else** ja **elseif**. Näide – arvu absoluutväärtuse leidmine (**MATLAB** aknas):

```
x=7;  
if x>0  
    y=x;  
else  
    y=-x;  
end
```

Pakettis **MATLAB** saab tingimusi realiseerida veel käsuga **switch**. (Vt. süntaksit *Help* – aknas).

Programmifailides on rangelt soovitatav kasutada kommentaare, st. juhiseid, mis algavad sümboliga „%” pakettis **MATLAB**, et autoril mõni aeg hiljem, aga ka teistel kasutajatel, oleks võimalik programmi loogikast kiiresti aru saada. Kommentaarid on kohustuslikud kohustuslike ülesannete lahendustes.

## Tuleb ära kasutada paketi **MATLAB** võimalused.

**MATLAB** on sisuliselt kõrgtaseme programmeerimiskeel paljude sisse loodud funktsioonide ja võimalustega. Nagu kõikide selliste keskkondade puhul, nii on ka paketi **MATLAB** võimalik kõrgtasemel konstruktsioone asendada madalatasemelistega. Näide selle kohta, vektori elementide ruutu tõstmise.

Madalatasemeline, jõumeetod:

```
>>x = [ 1 2 3 4 5 ]; % Defineeritakse vektor.  
>>y = zeros(size(x)); % Luuakse uus vektor.  
>>for i = 1 : numel(x) % Algab tsükkel: iga indeksi korral...  
>>y(i) = x(i)^2; % Arvutatakse vektori x elementide ruudud.  
>>end % Tsükli lõpp.
```

Sama asi kõrgtasemelises realisatsioonis, vektoriseeritud kujul:

```
>>x = [ 1 2 3 4 5 ]; % Defineeritakse vektor.  
>>y = x.^2; % Tõstetakse kõik elemendid ruutu.
```

Siiski tuleb iga kord analüüsida, kas kasutada kõrg- või madala tasemega realisatsiooni. Madalatasemelist varianti kasutavad kõik või vähemalt valdav enamus programmeerimiskeeltest ning seetõttu on nende „transportimine“ lihtne. Samuti on lihtsam programmeerijatel ümber kvalifitseeruda.

Kõrgtaseme funktsioonide peamine eesmärk on teha rohkem kui madalatasemelised ekvivalendid. Vastav kood on kompaktsem, lühem, vajab vähem programmeerijatööd ning reeglina on lühema programmi koostamisel ka väiksem võimalus vigu teha. Lühemast programmist saab ka kergemini ülevaate seda uurides. Paketi **MATLAB** töötavad kõrgtaseme programmid ka kiiremini kui madala tasemega koodid. Tuleb veel silmas pidada, et kõrgtasemel koodid on mõnikord nii lühikesed, et muutuvad arusaamatuteks, obskuurseteks. See eeldab nende programmide kommenteerimist ja dokumenteerimist. Kõrgtaseme koodid nõuavad mõneti teistsuguseid mõtteviise, need arenevad praktiseerides ja ainult nii. Üldine nõuanne kõrg- ja madalatasemelise realisatsioonide vahel valimiseks on: pole mõtet kulutada kolme tundi selleks, et optimiseerida koodi eesmärgiga programmi tööajas minutiline võit saavutada, enne kui pole selge, et olemasoleva programmi jooksutamiseks selle kolme tunni vältel vajalikku tulemust ei saavutata.

## Harilik iteratsioonimeetodid võrrandite lahendamiseks

Iteratsioonimeetodiks nimetatakse teatud võtet võrrandite, võrrandisüsteemide, ekstreemumülesannete jms. ligikaudseks lahendamiseks. Iteratsioonimeetodite idee seisneb järgnevas: ülesande täpsele lahendile  $\mathbf{x}^*$  leitakse algühend  $\mathbf{x}_0$ , millest lähtuvalt moodustatakse lähendite jada

$$\mathbf{x}_1; \mathbf{x}_2; \mathbf{x}_3; \dots; \mathbf{x}_n; \dots$$

Teatud tingimustel koondub see jada ülesande täpseks lahendiks  $\mathbf{x}^*$ . Loendaja  $n$  järkjärguline suurendamine tähendab iteratsioonisammude tegemist. See lõpetatakse küllalt suure  $n$  korral ja viimane lähend  $\mathbf{x}_n$  loetakse ülesande lahendiks. Selle „küllalt suure  $n$ “ väljavalimine toimub iteratsioonimeetodi peatumiskriteeriumi alusel.

Harilik iteratsioonimeetod on rakendatav võrrandile kujul

$$\mathbf{x} = \mathbf{g}(\mathbf{x}). \quad (1)$$

Kui võrrand ei ole esialgselt sellisel kujul, tuleb see teisendada selliseks. Siin  $\mathbf{g}$  tähistab etteantud ühemuutuja funktsiooni.

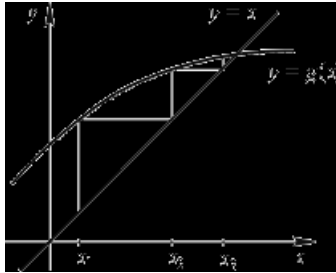
**Hariliku iteratsioonimeetodi** rakendamiseks vaatleme võrrandit kujul (1). Pärast algühendi  $\mathbf{x}_0$  fikseerimist moodustatakse lähendite jada järgmise eeskirjaga abil:

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k), \mathbf{k} = 1, 2, \dots \quad (2)$$

Jada  $\mathbf{x}_1; \mathbf{x}_2; \mathbf{x}_3; \dots; \mathbf{x}_n; \dots$  koondub võrrandi (1) täpseks lahendiks  $\mathbf{x}^*$ , kui mingis jada elemente ja võrrandi täpset lahendit sisaldavas vahemikus on täidetud tingimus

$$|\mathbf{g}'(\mathbf{x})| \leq \mathbf{q} < 1. \quad (3)$$

Kõrvalasuval joonisel on graafiliselt kujutatud hariliku iteratsioonimeetodi koondumisprotsess.



**Näide.** Lahendada võrrand  $e^{x/4} + x - 8 = 0$  (teha läbi iseseisvalt paketi **MATLAB** abil).

Hariliku iteratsioonimeetodi rakendamiseks tuleb teisendada võrrand kujule (1). Seda on võimalik teha mitmel eri viisil, kusjuures sellest, kuidas on valitud funktsioon  $\mathbf{g}$ , sõltub lahendite jada koondumise kiirus.

Üks võimalus oleks näiteks

$$\mathbf{x} = 8 - e^{x/4} .$$

Seega nüüd  $\mathbf{g}(\mathbf{x}) = 8 - e^{x/4}$ .

Ülesande iseseisval lahendamisel kontrollida koondumistingimuse (3) täidetust, valida alglähend ja viia läbi iteratsiooniprotsess (2).

## Võrrandite lahendamine paketi **MATLAB** abi.

Olgu vaja leida järgmise võrrandi lahend (lahendid):

$$f(x) = 0. \quad (1)$$

Siin  $f$  on etteantud ühemuutuja funktsioon. Toodud ülesannet nimetatakse veel funktsiooni  $f$  nullkohtade (ka juurte) leidmise ülesandeks. Tähistame sümboliga  $x^*$  võrrandi (1) täpse lahendi.

Selle lahendamiseks pole üheselt määratud üldist võtet. Enne võrrandi (1) numbrilise lahendamise juurde asumist tuleks leida vastus järgmistele üldistele küsimustele:

- Kas funktsiooni  $f$  väärtused on kergesti leitavad?
- Millised on nõuded lähislahendi täpsusele?
- Kas on ette teadaolevaid nõudeid lahendusalgoritmile (kiirus, stabiilsus jne.)?
- Kas  $f$  on polünoom?
- Kas funktsioonil  $f$  on iseärasusi?
- Millised on veel teada olevad funktsiooni  $f$  kvalitatiivsed omadused?

Võrrandi (1) lahendamise üldine strateegia:

- Joonistada funktsiooni  $f$  graafik. Selle alusel saab lokaliseerida funktsiooni  $f$  nullkoha ja sageli valida iteratsioonimeetodeile vajaliku alglahendi.
- Valida või täpsustada alglahend.
- Täpsustada valitud iteratsioonimeetodiga alglahendit vajaliku täpsuse saavutamiseni.

Alglahendi täpsustamine lõigu jagamise meetodil.

Esmalt tuleb määrata (näiteks graafiku alusel, seda analüütiliselt kontrollides) lõik  $[a,b]$ , mille otstes funktsioon  $f$  on erinevate märkidega, st.  $f(a)*f(b) < 0$ . Kui  $f$  on pidev funktsioon, sisaldab lõik  $[a,b]$  vähemalt ühe funktsiooni  $f$  nullkoha (milline matemaatilise analüüsi tulemus selle garanteerib?). Järgnevalt lõigu jagamise meetodi pseudokood.

Antud:  $f$ ,  $a$ ,  $b$ ,  $n$ , funktsiooni  $f$  väärtustamise algoritm (m-fail, otsedefiniitsioon).

$dx = (b - a)/n$

$x_{\text{vasak}} = a$

**for**  $i = 1:n$

$x_{\text{parem}} = x_{\text{vasak}} + dx$

**if**  $\{f(x)$  muudab märki lõigus  $[x_{\text{vasak}}, x_{\text{parem}}]\}$  % Erinevad kontrollivõimalused.

{salvestada  $[x_{\text{vasak}}, x_{\text{parem}}]$  edasiseks lahendamiseks}

**end**

$x_{\text{vasak}} = x_{\text{parem}}$

**end**

Funktsiooni  $f$  märgi muutumise kontrollimiseks kaks põhilist võimalust:

- **if**  $f(x_{\text{vasak}})*f(x_{\text{parem}}) < 0$

- **if**  $\text{sign}(f(x_{\text{vasak}})) \sim \text{sign}(f(x_{\text{parem}}))$  % Loogiline eitus:  $\sim = .$

Järgnevalt tuleb valida iteratsioonimeetod, kas harilik või Newtoni iteratsioonimeetod ja koostada m-fail, mis lahendab ülesande, st. leiab lähendite jada  $\{x(0), x(1), \dots, x(m)\}$ , mis koonduks võrrandi (1) täpseks lahendiks  $x^*$  protsessis  $m \rightarrow \infty$ . Tulemust kontrollida paketi MATLAB sisseehitatud vahenditega. Järgnevalt ülevaade nendest.

**fzero** leiab ühemuutuva funktsiooni nullkoha.

**Näide.**

$\gg g = \text{inline}('t^2-4')$

$g =$

Inline function:

$g(t) = t^2-4$

$\gg \text{fzero}(g,1)$

**ans** =

2

**roots** leiab polünoomi nullkohad (juured).

**Näide:**

```
>>p = [3 2 0 1]; % Sisestatakse polünoomi  $p(x) = 3x^3 + 2x^2 - 1$  kordajad.  
>> r = roots(p) % Polünoomi nullkohad, mille kordajad on massiivis p.  
  
r =  
  
-1.0000  
  
0.1667 + 0.5528i  
  
0.1667 - 0.5528i
```

**abs** reaalarvu absoluutväärtuse või kompleksarvu mooduli leidmine.

**inline** luuakse objekt käsurealt (*inline object*). Näide ülalpool.

**feval** leiab stringina defineeritud funktsiooni väärtuse.

**Näide:**

```
>> funktsioon=@(t)t^2-4  
  
funktsioon =  
  
@(t)t^2-4  
  
>> feval(funktsioon, 3)  
  
ans =  
  
5
```

**poly** leiab juurte alusel polünoomi kordajad.

**Näide:**

```
>> poly([1,2])  
  
ans =  
  
1 -3 2
```

**polyval** leiab polünoomi väärtuse.

**Näide:**

```
>> polyval(p,6) % Vektor p on defineeritud ülalpool.  
  
ans =
```

Tegelikult piirprotsessi  $m \rightarrow \infty$  muidugi ei realiseerita, vaid peatutakse lõpliku  $m$  korral ning loetakse  $x(m) \approx x^*$ . Iteratsioonimeetodi peatumistingimuse pidev kontroll, st. arvu  $m$  avastamine garanteerib, et ei sooritata üleliigseid arvutusi vajaliku täpsuse saavutamiseks ja selle võib formuleerida kahel viisil:

- Pärast lähendi  $x(k)$  arvutamist kontrollida, kas kaks viimast lähendit on üksteisele juba nii lähedal, et neid võib etteantud täpsuse  $\delta$  piires samastada, st. kontrollida, kas

$$|x(k) - x(k-1)| < \delta .$$

Kui see võrratus on rahuldatud, võib võtta  $x(m) = x(k)$ ,  $x(m) \approx x^*$ .

- Pärast lähendi  $x(k)$  arvutamist kontrollida, kas funktsiooni  $f$  väärtus  $f(x(k))$  on nullile piisavalt lähedal, et lähendi  $x(k)$  võiks lähislahendiks võtta:

$$|f(x(k))| < \varepsilon ?$$

Kui see võrratus on rahuldatud, võib võtta  $x(m) = x(k)$ ,  $x(m) \approx x^*$ . Viimast peatumise kontrolli meetodit nimetatakse ka peatumiseks hälbe järgi. Mõlemad esitatud võtted on kasutatavad nii hariliku kui ka Newtoni iteratsioonimeetodi korral.

## Teemad 2 ja 3

### Kohustuslik ülesanne

Valida võrrandite loetelust üks – selline, mis rühmakaaslaste poolt valimata. Teine võimalus: leida ise üks huvitav võrrand.

Koostada ja esitada m-fail(id), kus valitud võrrand lahendatakse nii hariliku kui ka Newtoni iteratsioonimeetodiga. Võrrelda koondumise kiirusi koonduvuskriteeriumite alusel (neid on kaks). Varieerida lähislahendi täpsusi ja vormistada vastav väljund. Modifitseerida lisatud m-faile vastavalt oma ülesandele ja esitada ka need.

NB! Kõikides esitatavates failides märkida ära autori andmed ja töö teostamise kuupäev(ad). Failinimed peavad algama autori perenimega, nagu Juhendis kirjeldatud.

**Valik võrrandeid lahendamiseks iteratsioonimeetodeil**  
*Paksus kirjas muutujad ja konstandid*

1. **Iks** pluss **üks** sulgudes ja kuubis pluss naturaalloogarithm **iksist** võrdub **nulliga**.
2. **Iks** korrutatud **kaks** astmes **iksiga** võrdub **ühega**.
3. Ruutjuur **iks** pluss **ühest** võrdub **üks** jagatud **iksiga**.
4. **Iks** miinus koosinus **iksist** võrdub **nulliga**.
5. **Kolm** **iksi** pluss koosinus **iksist** pluss **üks** võrdub **nulliga**.
6. **Iks** pluss loogarithm **iksist** võrdub **ühe kahendikuga**.
7. **Kaks** miinus **iks** võrdub naturaalloogarithmiga **iksist**.
8. **Iks** miinus **üks** sulgudes ja ruudus võrdub **e** astmes **iks** jagatud **kahega**.
9. **Iks** korrutatud **e** astmes **iksiga** miinus **kaks** võrdub **nulliga**.
10. **Kaks** koma **kaks** korrutatud **iksiga** miinus **kaks** astmes **iks** võrdub **nulliga**.
11. **Iks** ruudus pluss **neli** korda siinus **iks** võrdub **nulliga**.
12. **Kaks** **iksi** miinus loogarithm **iksist** võrdub **seitsmega**.
13. **Viis** **iksi** miinus **kaheksa** korda naturaalloogarithm **iksist** võrdub **kaheksaga**.

14. **Kolm iksi** miinus **e** astmes **iks** võrdub **nulliga**.
15. **Iks** pluss **üks** sulgudes ja ruudus korrutatud **iksiga** võrdub **ühega**.
16. **Iks** pluss **üks** sulgudes ja kuubis võrdub **iksiga**.
17. **Iks** ruudus võrdub siinus **iksiga**.
18. **Iks** kuubis võrdub siinus **iksiga**.
19. Ruutjuur logaritmi **iks** pluss **kahest** võrdub **iksiga**.
20. Naturaallogaritmi **iks** pluss **ühest** võrdub **iks** ruuduga.
21. **Kaks iksi** pluss logaritmi **iksist** võrdub miinus **ühe kahendikuga**.
22. **Kaks iksi** pluss koosinus **iksist** võrdub **poolega**.
23. **Üks** pluss siinus **iks** kahendikust võrdub **iks** ruuduga.
24. **Iks** kahendikku pluss logaritmi **iks** miinus **ühest** võrdub **ühe kahendikuga**.
25. Siinus **0.5** pluss **iksist** võrdub **kaks iksi** miinus **poolega**.
26. **Kaks iksi** pluss logaritmi **iks** pluss **kahest** võrdub **kolmega**.
27. Logaritmi **kaks iksi** pluss **ühest** võrdub **kaks** miinus **iksiga**.
28. **Kaks** siinus **iks** miinus **null koma kuuest** võrdub **1.5** miinus **iksiga**.
29. **Iks** pluss logaritmi **üks** pluss **iksist** võrdub **ühe koma viiega**.
30. **Iks** pluss koosinus **iksist** võrdub **ühega**.

Näide: Newtoni iteratsioonimeetod ei anna tulemust

Newtoni iteratsioonimeetodi rakendamiseks peab võrrand olema kujul

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \quad (1)$$

Siin  $\mathbf{f}$  tähistab etteantud ühemuutuja funktsiooni.

**Newtoni iteratsioonimeetodi** arvutuseeskiri võrrandi (1) jaoks on järgmine. Valime alg lähendi  $\mathbf{x}_0$  ja koostame lähendite jada  $\{\mathbf{x}_1; \mathbf{x}_2; \mathbf{x}_3; \dots; \mathbf{x}_n; \dots\}$  sellise eeskirja abil:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{f}(\mathbf{x}_k)/\mathbf{f}'(\mathbf{x}_k), \quad \mathbf{k} = 1, 2, \dots \quad (2)$$

Iteratsiooniprotsessi geomeetriline tõlgendus: lähendite jada järgmise liikme  $\mathbf{x}_{k+1}$  saame, kui leiame punktis  $\mathbf{x}_k$  funktsiooni  $\mathbf{f}$  graafikule tõmmatud puutuja lõikepunkti  $\mathbf{x}$ -teljega.

Püüame leida sellise funktsiooni  $\mathbf{f}$ , et iteratsiooniprotsess (2) hakkaks „pendeldama“ kahe lähendi vahel punkti  $\mathbf{a}$  ümber, s.t. et

$$\mathbf{x}_{k+1} - \mathbf{a} = -(\mathbf{x}_k - \mathbf{a}).$$

See juhtub siis, kui  $\mathbf{f}$  rahuldab seost (kontrollida!):

$$\mathbf{x} - \mathbf{a} - \mathbf{f}(\mathbf{x})/\mathbf{f}'(\mathbf{x}) = -(\mathbf{x} - \mathbf{a}).$$

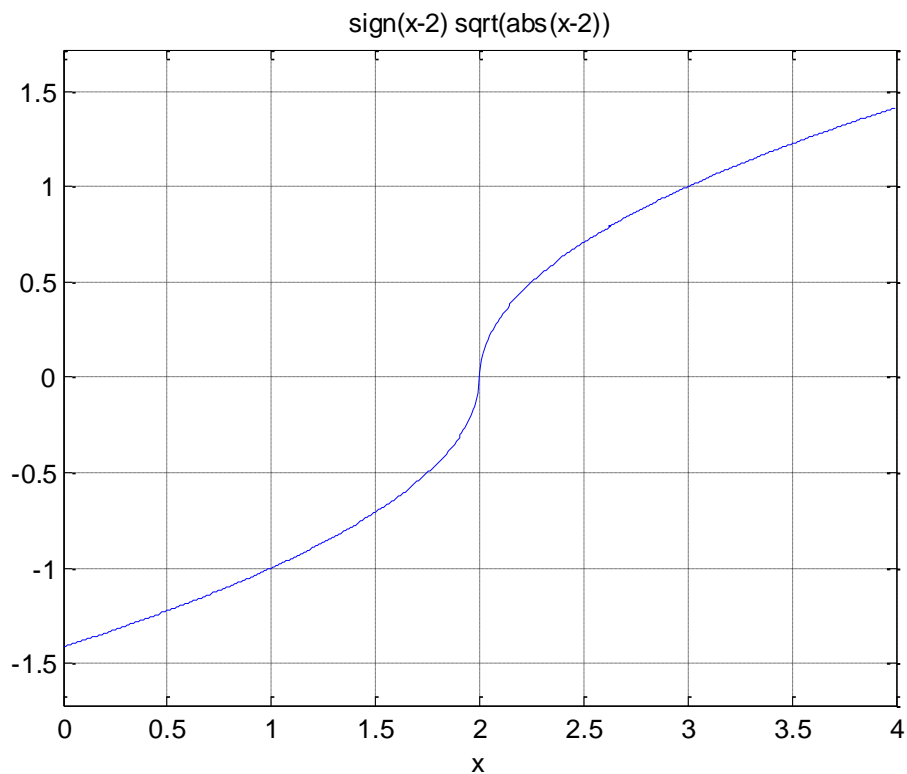
Funktsiooni  $\mathbf{f}$  leidmiseks saame siit eralduvate muutujatega hariliku diferentsiaalvõrrandi:

$$df/f = 1/(2(\mathbf{x} - \mathbf{a})).$$

Selle lahend:

$$\mathbf{f}(\mathbf{x}) = \text{sign}(\mathbf{x} - \mathbf{a}) * \text{sqrt}(|\mathbf{x} - \mathbf{a}|). \quad (3)$$

Funktsiooni  $f$  nullkohaks ja võrrandi (1) lahendiks on  $x^* = a$ . Joonis, kui  $a = 2$ :



See joonis saadakse käsuga:

```
>> ezplot('sign(x-2)*sqrt(abs(x-2))',0,4)
```

Iteratsiooniprotsess (2) viib lõpmatusse tsüklisse iga algühendi puhul: suvalises punktis funktsiooni  $f$  graafikule tõmmatud puutuja lõikab  $x$  – telge sümmeetriliselt teisel pool punkti  $a$ . Teoreetilised koonduvustingimused pole samuti täidetud, sest  $f'(x) \rightarrow \infty$ , kui  $x \rightarrow a$ .

## Newtoni iteratsioonimeetodid võrrandite lahendamiseks

Iteratsioonimeetodiks nimetatakse teatud võtet võrrandite, võrrandisüsteemide, ekstreemumülesannete jms. ligikaudseks lahendamiseks. Iteratsioonimeetodite idee seisneb järgnevas: ülesande täpsele lahendile  $\mathbf{x}^*$  leitakse alglahend  $\mathbf{x}_0$ , millest lähtuvalt moodustatakse lähendite jada

$$\mathbf{x}_1; \mathbf{x}_2; \mathbf{x}_3; \dots; \mathbf{x}_n; \dots$$

Teatud tingimustel koondub see jada ülesande täpseks lahendiks  $\mathbf{x}^*$ . Loendaja  $n$  järkjärguline suurendamine tähendab iteratsioonisammude tegemist. See lõpetatakse küllalt suure  $n$  korral ja viimane lähend  $\mathbf{x}_n$  loetakse ülesande lahendiks. Selle „küllalt suure  $n$ “ väljavalimine toimub iteratsioonimeetodi peatumiskriteeriumi alusel.

Newtoni iteratsioonimeetodi rakendamiseks peab võrrand olema kujul

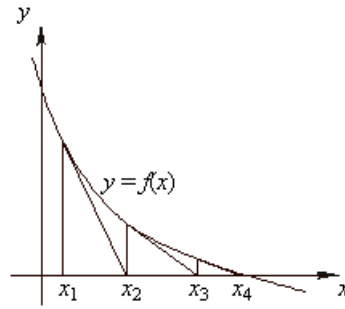
$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \quad (1)$$

Siin  $\mathbf{f}$  tähistab etteantud ühemuutuja funktsiooni.

**Newtoni iteratsioonimeetodi** vaatleme võrrandit kujul (1). Valime alglahendi  $\mathbf{x}_0$  ja koostame lähendite jada järgmise eeskirja abil:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{f}(\mathbf{x}_k)/\mathbf{f}'(\mathbf{x}_k), \quad \mathbf{k} = 1, 2, \dots$$

Newtoni meetodi korral on iteratsiooniprotsessi geomeetriline tõlgendus selline: lähendite jada järgmise liikme  $\mathbf{x}_{k+1}$  saame, kui leiame punktis  $\mathbf{x}_k$  funktsiooni  $\mathbf{f}$  graafikule tõmmatud puutuja lõikepunkti  $\mathbf{x}$ -teljega. Seda mõttekäiku selgitab allolev joonis.



**Näide.** Lahendada võrrand  $3^{x-4} - 2^x + 0,4 = 0$  (teha läbi iseseisvalt paketi MATLAB abil).

Newtoni meetodi korral on järgmise lähendi  $x_{k+1}$  arvutamiseks vaja ka funktsiooni  $f(x)$  tuletise väärtust  $f'(x_k)$  kohal  $x_k$ .

## Võrrandite lahendamine paketiga **MATLAB**: õpijuhhis

**Hädavajalikud teadmised.** Peab teadma, kuidas

- vormistada võrrand  $f(x) = 0$  paketis **MATLAB**;
- joonistada funktsioonide graafikuid;
- demonstreerida graafiliselt funktsiooni  $f$  nullkohta  $x$ -teljel;
- kirjutada  $m$ -fail, mis leiab funktsiooni  $f$  väärtuse;
- lokaliseerida funktsiooni nullkohta lõigu poolitamise meetodil ja tuua vastav näide;
- kirjeldada olukorda, kus lõigu poolitamise meetod annab vale tulemuse;
- demonstreerida graafiliselt hariliku iteratsioonimeetodiga saadavat lahendit;
- kirjeldada olukorda, kus harilik iteratsioonimeetod ei koonu;
- kirjutada  $m$ -fail, mis leiab funktsiooni  $f$  tuletise  $f'(x)$  väärtuse;
- demonstreerida graafiliselt Newtoni iteratsioonimeetodiga saadavat lahendit;
- demonstreerida graafiliselt modifitseeritud Newtoni iteratsioonimeetodiga saadavat lahendit;
- kirjeldada olukorda, kus Newtoni iteratsioonimeetod ei anna tulemust;
- kirjeldada ja kontrollida iteratsiooniprotsessi lõpetamise kriteeriume;
- kasutada sisseehitatud funktsiooni **fzero** ;
- kirjeldada algoritme, mida kasutab **fzero** ;
- põhjendada, miks tavalisi nullkohtade leidmise meetodeid ei soovitata polünoomide nullkohtade leidmiseks;
- leida polünoomide nullkohti (k.a. meetodite kirjeldus);
- kasutada sisseehitatud funktsiooni **roots** ;
- võrrelda hariliku iteratsioonimeetodi ja Newtoni iteratsioonimeetodi koonduvuskiirusi;
- täpsustada iteratsioonimeetodite peatumiskriteeriume etteantud funktsiooni korral, et vältida üleliigseid iteratsioonisamme.

## Mittelineaarsete võrrandsüsteemide lahendamine paketiga **MATLAB**: õpijuhised

**Hädavajalikud teadmised.** Peab teadma, kuidas

- vormistada võrrandsüsteemi  $\mathbf{f}(\mathbf{x}) \equiv [f_1(x_1, \dots, x_n); \dots; f_n(x_1, \dots, x_n)] = 0$  paketi **MATLAB**;
- joonistada funktsioonide 2D ja 3D graafikuid;
- kirjutada m-fail, mis leiab funktsiooni  $f$  väärtuse;
- kirjeldada hariliku iteratsioonimeetodi arvutusskeemi;
- demonstreerida graafiliselt hariliku iteratsioonimeetodiga saadavat lahendit;
- kirjutada m-fail, mis leiab vektorfunktsiooni  $\mathbf{f}$  Frechet' tuletise  $\mathbf{f}'(\mathbf{x})$  e. Jakobiaani;
- kirjeldada Newtoni iteratsioonimeetodi arvutusskeemi;
- kirjeldada modifitseeritud Newtoni iteratsioonimeetodi arvutusskeemi;
- kirjeldada ja kontrollida iteratsiooniprotsessi lõpetamise kriteeriumi;
- kasutada sisseehitatud funktsiooni **fsolve** ;
- kirjeldada algoritme, mida kasutab **fsolve** ;
- võrrelda hariliku iteratsioonimeetodi ja Newtoni iteratsioonimeetodi koonduvuskiirusi;

[peep.miidla@ut.ee](mailto:peep.miidla@ut.ee)

## Teema 4: kohustuslik ülesanne

Kirjutada ja varustada vajalike kommantaaridega m-fail, mis lahendab mittelineaarse võrrandisüsteemi hariliku (Seideli), Newtoni ja Newtoni modifitseeritud iteratsioonimeetoditega. Ülesande võib valida allolevast loetelust või iseseisvalt. Kontrollida tulemusi **MATLAB** abil, funktsioon **fsolve**.

(i) (Siin  $a$  on vabalt valitav konstant.)

$$\begin{aligned}(x - a)^3 + (y - a)^3 &= a^3, \\ e^{a-x} + e^{a-y} &= a+1.\end{aligned}$$

(ii)

$$\begin{aligned}x + y - x*y + 2 &= 0, \\ x*e^{-y} - 1 &= 0.\end{aligned}$$

(iii)

$$\begin{aligned}2*x - y &= e^{-x}, \\ -x + 2*y &= e^{-y}.\end{aligned}$$

(iv)

$$\begin{aligned}x^3 + y &= 1, \\ y^3 - x &= -1.\end{aligned}$$

Muuta lahendi täpsust ja võrrelda meetodite koondumiskiirusi. Väljastada lähendite jadad ja iteratsioonide arvud.

Alglähend leida graafiliselt.

Selgitada ja kommenteerida tulemusi.

*MTMM.00.005 Numbrilised meetodid*

**Võrrandisüsteemide lahendamine,  
mittelineaarsed süsteemid**

# Mittelineaarsete võrrandisüsteemide lahendamine

Vaatluse all on üldine mittelineaarne võrrandisüsteem

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} ,$$

kus  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n)$  on tundmatute vektor ja  $\mathbf{f} = (f_1; f_2; f_3; \dots; f_n)$  on vektorfunktsioon.

Lahendamise põhivõteteks on iteratsioonimeetodid.  
Täpselt õnnestub lahend leida harva.

# Iteratsioonimeetodid võrrandisüsteemide lahendamiseks

Iteratsioonimeetodite idee seisneb järgnevas: ülesande täpsele lahendile  $\mathbf{x}^*$  leitakse mingi alglähend  $\mathbf{x}_0$ , millest lähtuvalt moodustatakse lähendite jada

$$\mathbf{x}_1 ; \mathbf{x}_2 ; \mathbf{x}_3 ; \dots ; \mathbf{x}_n ; \dots .$$

Jada püütakse konstrueerida nii, et see koonduks esialgse ülesande täpseks lahendiks  $\mathbf{x}^*$ . Loendaja  $n$  järkjärguline suurendamine tähendab iteratsioonisammude tegemist. See lõpetatakse küllalt suure  $n$  korral, peatumistingimuse täidetuse korral ja viimane lähend  $\mathbf{x}_n$  loetakse ülesande lahendiks.

# Harilik iteratsioonimeetodid võrrandisüsteemide lahendamiseks

Harilik iteratsioonimeetod on rakendatav võrrandisüsteemile kujul

$$\mathbf{x} = \mathbf{g}(\mathbf{x}).$$

Kui võrrand ei ole esialgselt sellisel kujul, tuleb see teisendada selliseks. Siin  $\mathbf{g}$  tähistab vektorfunktsiooni.

Pärast alglähendi  $\mathbf{x}_0$  fikseerimist moodustatakse lähendite jada järgmise eeskirjaga abil:

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k), \mathbf{k} = 1, 2, \dots$$

Seideli meetodi korral kasutatakse uute lähendvektorite leidmisel juba leitud lähendi komponente.

# Iteratsioonimeetodid võrrandisüsteemide lahendamiseks

Levinud meetod mittelineaarsete algebraliste võrrandisüsteemide lahendamiseks on Newtoni iteratsioonimeetod, samuti selle modifikatsioonid.

Olgu meil antud võrrand kujul  $f(x) = 0$ . Valime alglähendi  $x_0$  ja koostame lähendite jada järgmise eeskirja abil:

$$x_{k+1} = x_k - [f'(x_k)]^{-1} * f(x_k), \quad k = 0, 1, \dots$$

# Iteratsioonimeetodid võrrandisüsteemide lahendamiseks

Newtoni iteratsioonimeetodi modifitseeringutest on tuntuim selline:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{f}'(\mathbf{x}_k)]^{-1} * \mathbf{f}(\mathbf{x}_k), \mathbf{k} = 0, 1, \dots$$

Valemities  $[\mathbf{f}'(\mathbf{x}_k)]^{-1}$  või vastavalt  $[\mathbf{f}'(\mathbf{x}_0)]^{-1}$  tähistavad funkstiooni  $f$  Frechet' tuletist (Jakobiaani) kohal  $\mathbf{x}_k$  või kohal  $\mathbf{x}_0$ .

# Alglähend

Nagu võrranditegi puhul, on ka võrrandisüsteemide korral Newtoni iteratsioonimeetodi heaks koonduvuseks vaja leida hea alglähend  $x_0$  .

Näiteid lineaarsete võrrandisüsteemide lahendamisest.

$$\text{N. } \begin{cases} 2x + 3y - z = 9, \\ 2y - z = 2, \\ 3z = 12. \end{cases} \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}.$$

$$\text{N. } \begin{cases} 3x - 2y = 2, \\ x + y = 9. \end{cases} \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}.$$

N. Gaussi elimineerimismeetod, kolmnurksele kujule viimine.

$$\begin{cases} 2x - 3y + z = -5, \\ 3x + 2y - z = 7, \\ x + 4y - 5z = 3. \end{cases} \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}.$$

$$\left( \begin{array}{ccc|c} 2 & -3 & 1 & -5 \\ 3 & 2 & -1 & 7 \\ 1 & 4 & -5 & 3 \end{array} \right) \rightarrow \text{R2}-3*\text{R3}, 2*\text{R3}-\text{R1} \rightarrow \left( \begin{array}{ccc|c} 2 & -3 & 1 & -5 \\ 0 & -10 & 14 & -2 \\ 0 & 11 & -11 & 11 \end{array} \right) \rightarrow$$

$$\text{R3}/11, 10*\text{R3}+\text{R2} \rightarrow \left( \begin{array}{ccc|c} 2 & -3 & 1 & -5 \\ 0 & -10 & 14 & -2 \\ 0 & 0 & 4 & 8 \end{array} \right) \rightarrow \text{Süsteem on kolmnurksel kujul, tehakse tagasisamm.}$$

N. Gauss-Jordani elimineerimismeetod. Süsteemi maatriksi viiakse diagonaalkujule.

$$\begin{cases} 3x + 4y + z = 6, \\ 2x - y + 2z = -5, \\ x + 3y - z = 9. \end{cases} \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ -4 \end{pmatrix}.$$

$$\left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 2 & -1 & 2 & -5 \\ 1 & 3 & -1 & 9 \end{array} \right) \rightarrow \text{R2}-2*\text{R3}, 3*\text{R3}-\text{R1} \rightarrow \left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 0 & -7 & 4 & -23 \\ 0 & 5 & -4 & 21 \end{array} \right) \rightarrow 5*\text{R2}+7*\text{R3}, \text{R2} \leftrightarrow \text{R3}$$

$$\rightarrow \left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 0 & 5 & -4 & 21 \\ 0 & 0 & -8 & 32 \end{array} \right) \rightarrow -\text{R3}/8, \text{R2}+4*\text{R3} \rightarrow \left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 0 & 5 & 0 & 5 \\ 0 & 0 & 1 & -4 \end{array} \right) \rightarrow \text{R2}/5, \text{R1}-4*\text{R2}$$

$$\rightarrow \left( \begin{array}{ccc|c} 3 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -4 \end{array} \right) \rightarrow \text{R1}-\text{R3} \rightarrow \left( \begin{array}{ccc|c} 3 & 0 & 0 & 6 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -4 \end{array} \right) \rightarrow \text{R1}/3 \rightarrow \left( \begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -4 \end{array} \right).$$

N. „Halb“ süsteem.

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{Maatriksi } LU \text{ faktoriseering: } \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Teeme läbi Gaussi elimineerimismeetodi peaelementi välja eraldamata.

Eialgu  $\varepsilon$  väike arv.

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ saame... } y_1 = 1; y_2 = 1 - \frac{1}{\varepsilon}$$

$$\begin{bmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \text{ ...saame... } x_1 = 0, x_2 = 1$$

Kui võtta  $\varepsilon$  väga väikeseks,  $\varepsilon < \varepsilon_{\text{mach}} \approx 2^{-53}$ , siis saame lahendiks  $x_1 = x_2 = 1$ .

Lineaarsete võrrandisüsteemide  
lahendamine, III osa  
iteratsioonimeetodid

***MTMM.00.005 Numbrilised meetodid***

# Lineaarne võrrandisüsteem (maatriks peab olema ruutmaatriks)

Üldkuju:

$$Ax = b$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{bmatrix}, b_i \in \mathbb{R}$$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{bmatrix}, a_{i,j} \in \mathbb{R} \quad \det(A) = \begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{vmatrix} \neq 0$$

## Iteratsioonimeetodid

$$x_{i+1} = Bx_i + c, i = 0, 1, \dots$$

Richardsoni iteratsioonimeetod:

$$x_{i+1} = x_i + (b - Ax_i) = (E - A)x_i + b \quad \text{st.} \quad B = E - A.$$

Kui tähistada  $e_i = x - x_i$  kirjutada skeem ümber nii:

$$x - x_{i+1} = x - x_i - (Ax - Ax_i) \quad \text{ehk} \quad e_{i+1} = e_i - Ae_i = (E - A)e_i$$

$$\|e_{i+1}\| \leq \|(E - A)e_i\| \leq \|E - A\| \|e_i\| \leq \|E - A\|^2 \|e_{i-1}\| \leq \dots \leq \|E - A\|^i \|e_0\|.$$

Koondumiseks on tarvis, et  $(e_i \rightarrow 0)$  ehk et  $\|E - A\| < 1$ .

# Iteratsioonimeetodid

Jacobi iteratsioonimeetod

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{D}^{-1} (\mathbf{b} - \mathbf{A}\mathbf{x}_i) = (\mathbf{E} - \mathbf{D}^{-1} \mathbf{A})\mathbf{x}_i + \mathbf{D}^{-1} \mathbf{b} ,$$

kus  $\mathbf{D}$  peadiagonaalil on matriksi  $\mathbf{A}$  diagonaal (nullist erinev eelduse kohaselt), mujal nullid;  $\mathbf{B} = \mathbf{E} - \mathbf{D}^{-1} \mathbf{A}$  .

Tähistame  $\mathbf{e}_i = \mathbf{x} - \mathbf{x}_i$  ja kirjutame skeemi ümber kujul:

$$\mathbf{x} - \mathbf{x}_{i+1} = \mathbf{x} - \mathbf{x}_i - \mathbf{D}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}_i) ;$$

$$\mathbf{e}_{i+1} = \mathbf{e}_i - \mathbf{D}^{-1} \mathbf{A}\mathbf{e}_i = (\mathbf{E} - \mathbf{D}^{-1} \mathbf{A})\mathbf{e}_i$$

$$\begin{aligned} \|\mathbf{e}_{i+1}\| &\leq \|(\mathbf{E} - \mathbf{D}^{-1} \mathbf{A})\mathbf{e}_i\| \leq \|\mathbf{E} - \mathbf{D}^{-1} \mathbf{A}\| \|\mathbf{e}_i\| \leq \|\mathbf{E} - \mathbf{D}^{-1} \mathbf{A}\|^2 \|\mathbf{e}_{i-1}\| \leq \\ &\dots \leq \|\mathbf{E} - \mathbf{D}^{-1} \mathbf{A}\|^i \|\mathbf{e}_0\| . \end{aligned}$$

Koondumiseks on vaja, et  $(\mathbf{e}_i \rightarrow 0)$  ehk  $\|\mathbf{E} - \mathbf{D}^{-1} \mathbf{A}\| < 1$  .

# Iteratsioonimeetodid

Gauss-Seideli iteratsioonimeetod:

$$\mathbf{x}_{i+1} = (\mathbf{D} - \mathbf{L})^{-1} \mathbf{x}_{i+1} + (\mathbf{U}\mathbf{x}_i + \mathbf{b}),$$

Kus  $-\mathbf{L}$  on matriksi  $\mathbf{A}$  alumine ja  $-\mathbf{U}$  ülemine kolmnurkne osa;  
 $\mathbf{D}$  on diagonaal.

Täpsemalt välja kirjutades:

$$\left\{ \begin{array}{l} x_1^{(i+1)} = (b_1 - a_{1,2}x_2^{(i)} - a_{1,3}x_3^{(i)} - a_{1,4}x_4^{(i)} - \dots - a_{1,n}x_n^{(i)}) / a_{1,1}, \\ x_2^{(i+1)} = (b_2 - a_{2,1}x_1^{(i+1)} - a_{2,3}x_3^{(i)} - a_{2,4}x_4^{(i)} - \dots - a_{2,n}x_n^{(i)}) / a_{2,2}, \\ x_3^{(i+1)} = (b_3 - a_{3,1}x_1^{(i+1)} - a_{3,2}x_2^{(i+1)} - a_{3,4}x_4^{(i)} - \dots - a_{3,n}x_n^{(i)}) / a_{3,3}, \\ \dots \\ x_n^{(i+1)} = (b_n - a_{n,1}x_1^{(i+1)} - a_{n,2}x_2^{(i+1)} - a_{n,3}x_3^{(i+1)} - \dots - a_{n,n-1}x_{n-1}^{(i+1)}) / a_{n,n}. \end{array} \right.$$

# Lineaarsete võrrandisüsteemide lahendamine, I osa

***MTMM.00.005 Numbrilised meetodid***

# Lineaarne võrrandisüsteem (maatriks peab olema ruutmaatriks)

Üldkuju:

$$Ax = b$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{bmatrix}, b_i \in \mathbb{R}$$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{bmatrix}, a_{i,j} \in \mathbb{R} \quad \det(A) = \begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{vmatrix} \neq 0$$

# Maatriksi faktoriseerimine

$$Ax = b \dots \rightarrow \dots PLUx = b$$

$$L = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{2,1} & 1 & 0 & \dots & 0 \\ l_{3,1} & l_{3,2} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n,1} & l_{n,2} & l_{n,3} & \dots & 1 \end{bmatrix}, \det L = 1; \quad U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \dots & u_{2,n} \\ 0 & 0 & u_{3,3} & \dots & u_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & u_{n,n} \end{bmatrix}, \det U \neq 0.$$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{bmatrix}, a_{i,j} \in \mathbb{R} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{bmatrix}, b_i \in \mathbb{R}$$

# Permutatsioonimaatriks

- **Def.** Permutatsioonimaatriksiks nimetatakse maatriksit  $P$ , mis saadakse ühikmaatriksist ridade vahetamise (permuteerimise) tagajärjel.

**Lemma** Olgu  $P_1$ ,  $P_2$  ja  $P$  *permutatsioonimaatriksid* (dimensiooniga  $n \times n$ ).

Siis

- $P^*X$  on maatriks, mis saadakse maatriksist  $X$  selle ridade vahetamise teel ja  $X^*P$  on sama veergude vahetamise järel.
- $P^{-1} = P^T$  .
- $\det(P) = \pm 1$  .
- $P_1^* P_2$  on samuti permutatsioonimaatriks.

# Gaussi elimineerimismeetod

Algne süsteem  $\mathbf{A}^*\mathbf{x} = \mathbf{b}$  viiakse ekvivalentsele kujule:

$$\mathbf{P}^*\mathbf{L}^*\mathbf{U}^*\mathbf{x} = \mathbf{b}$$

ja seda lahendatakse sellises järjekorras:

$$\mathbf{L}^*\mathbf{U}^*\mathbf{x} = \mathbf{P}^{-1}^*\mathbf{b} = \mathbf{P}^T^*\mathbf{b} \quad (\text{vahetatakse } \mathbf{b} \text{ komponendid});$$

$$\mathbf{U}^*\mathbf{x} = \mathbf{L}^{-1}^*(\mathbf{P}^T^*\mathbf{b}) \quad (\text{forward substitution});$$

$$\mathbf{x} = \mathbf{U}^{-1}^*(\mathbf{L}^{-1}^*\mathbf{P}^T^*\mathbf{b}) \quad (\text{back substitution}).$$

**TEOREEM.** Kui  $n^*n$  maatriks  $\mathbf{A}$  ei ole singulaarne, siis leiduvad permutatsioonimaatriks  $\mathbf{P}$  (ühikmaatriks, mille read on vahetatud), mittesingulaarne alumine kolmnurkmaatriks  $\mathbf{L}$  ja mittesingulaarne ülemine kolmnurkmaatriks  $\mathbf{U}$  sellised, et  $\mathbf{A}=\mathbf{P}^*\mathbf{L}^*\mathbf{U}$ .

# Peaelementide väljaeraldamine (*pivoting*)

- Osaline peaelementide kasutamine (*partial pivoting*).  
Kasutatakse suurimat elementi igas veerus;  $L*U = P*A$ .  
[Kasutavad **SCILAB** ja **MATLAB**]
- Täielik peaelementide kasutamine (*complete pivoting*).  
Leitakse igal elimineerimissammul suurim element kogu maatriksis;  $L*U = P*A*Q$ .
- Peaelemente ei eraldata välja; erijuhtudel.

**Tehete arv Gaussi  
elimineerimismeetodi korral:**

$$n^3 / 3 * (n^2 + 3*n - 1)$$

# Erijuhud

Juhud, mil ei ole vaja juhtelemente välja eraldada:

- Kui maatriks  $\mathbf{A}$  on domineeriva peadiagonaaliga:

$$|a_{j,j}| \geq \sum_{i \neq j} a_{i,j}$$

- Kui maatriks  $\mathbf{A}$  on positiivselt määratud:

$$\mathbf{A}^T = \mathbf{A} ; \mathbf{x}^* \mathbf{A} \mathbf{x}^T > 0 \text{ iga } \mathbf{x} \text{ korral}$$

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

# Näide

Maatriksi **LU** faktoriseering:

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Teeme läbi Gaussi  
elimineerimismeetodi  
peaelementi välja  
eraldamata.

Esiialgu  $\varepsilon$  väike arv.

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ saame... } y_1 = 1; y_2 = 1 - \frac{1}{\varepsilon}$$

$$\begin{bmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \dots \text{saame... } x_1 = 0, x_2 = 1$$

Kui võtta  $\varepsilon$  väga väikeseks,  $\varepsilon < \varepsilon_{\text{mach}} \approx 2^{-53}$ , siis saame lahendiks  $x_1 = x_2 = 1$ .

# Gaussi elimineerimismeetod, kolmnurksele kujule viimine

$$\begin{cases} 2x - 3y + z = -5, \\ 3x + 2y - z = 7, \\ x + 4y - 5z = 3. \end{cases}$$

Lahend:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}$$

$$\left( \begin{array}{ccc|c} 2 & -3 & 1 & -5 \\ 3 & 2 & -1 & 7 \\ 1 & 4 & -5 & 3 \end{array} \right) \rightarrow \text{Rida2}-3*\text{Rida3}, 2*\text{Rida3}-\text{Rida1} \rightarrow \left( \begin{array}{ccc|c} 2 & -3 & 1 & -5 \\ 0 & -10 & 14 & -2 \\ 0 & 11 & -11 & 11 \end{array} \right) \rightarrow$$

Rida3/11, 10\*Rida3+Rida2

$$\rightarrow \left( \begin{array}{ccc|c} 2 & -3 & 1 & -5 \\ 0 & -10 & 14 & -2 \\ 0 & 0 & 4 & 8 \end{array} \right) \rightarrow \text{Süsteem on kolmnurksel kujul, tehakse tagasisamm.}$$

# Gauss-Jordani elimineerimismeetod

$$\begin{cases} 3x + 4y + z = 6, \\ 2x - y + 2z = -5, \\ x + 3y - z = 9. \end{cases}$$

Lahend:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ -4 \end{pmatrix}$$

$$\left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 2 & -1 & 2 & -5 \\ 1 & 3 & -1 & 9 \end{array} \right) \rightarrow \text{R2}-2*\text{R3}, \text{3}* \text{R3}-\text{R1} \rightarrow \left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 0 & -7 & 4 & -23 \\ 0 & 5 & -4 & 21 \end{array} \right) \rightarrow$$

$$5*\text{R2}+7*\text{R3}, \text{R2} \leftrightarrow \text{R3} \rightarrow \left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 0 & 5 & -4 & 21 \\ 0 & 0 & -8 & 32 \end{array} \right) \rightarrow -\text{R3}/8, \text{R2}+4*\text{R3} \rightarrow \left( \begin{array}{ccc|c} 3 & 4 & 1 & 6 \\ 0 & 5 & 0 & 5 \\ 0 & 0 & 1 & -4 \end{array} \right) \rightarrow$$

$$\text{R2}/5, \text{R1}-4*\text{R2} \rightarrow \left( \begin{array}{ccc|c} 3 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -4 \end{array} \right) \rightarrow \text{R1}-\text{R3} \rightarrow \left( \begin{array}{ccc|c} 3 & 0 & 0 & 6 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -4 \end{array} \right) \rightarrow \text{R1}/3$$

$$\rightarrow \left( \begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -4 \end{array} \right)$$

Süsteemi maatriks on viidud diagonaalkujule

Lineaarsete võrrandisüsteemide  
lahendamine, II osa  
Ristkülikukujuline maatriks

***MTMM.00.005 Numbrilised meetodid***

# Lineaarne võrrandisüsteem (maatriks ei pea olema ruutmaatriks)

Üldkuju:

$$Ax = b$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_m \end{bmatrix}, b_i \in \mathbb{R}$$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & a_{m,3} & \dots & a_{m,n} \end{bmatrix}, a_{i,j} \in \mathbb{R} \quad \det(A) = \begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & a_{m,3} & \dots & a_{m,n} \end{vmatrix} \neq 0$$

Maatriksi  $\mathbf{A}$  veerud on lineaarselt sõltumatud parajasti siis, kui süsteemil  $\mathbf{A}\mathbf{x} = \mathbf{0}$  on vaid triviaalne lahend  $\mathbf{x} = \mathbf{0}$ . Sel juhul  $\mathbf{m} \geq \mathbf{n}$  ja  $\text{rank}(\mathbf{A}) = \mathbf{n}$ . Lineaarselt sõltuvate veergude puhul on iga veerg esitatav teiste lineaarkombinatsioonina.

$\mathbf{B}$  on maatriksi  $\mathbf{A}$  vasakpoolne (parempoolne) pöördmaatriks, kui  $\mathbf{BA} = \mathbf{E}$  ( $\mathbf{AB} = \mathbf{E}$ ).

$\mathbf{A}^\#$  on maatriksi  $\mathbf{A}$  üldistatud pöördmaatriks, kui  $\mathbf{AA}^\#\mathbf{A} = \mathbf{A}$ .

$\mathbf{A}^+$  on maatriksi  $\mathbf{A}$  pseudopöördmaatriks (Moore – Penrose pöördmaatriks), kui see on

- 1) üldistatud pöördmaatriks maatriksile  $\mathbf{A}$ , s.t  $\mathbf{AA}^+\mathbf{A} = \mathbf{A}$ ;
- 2)  $\mathbf{A}^+\mathbf{AA}^+ = \mathbf{A}^+$ , s.t.  $\mathbf{A}$  on maatriksi  $\mathbf{A}^+$  üldistatud pöördmaatriks;
- 3)  $(\mathbf{AA}^+)^{\mathbf{H}} = \mathbf{AA}^+$ ;      4)  $(\mathbf{A}^+\mathbf{A})^{\mathbf{H}} = \mathbf{A}^+\mathbf{A}$ . Siin  $\mathbf{A}^{\mathbf{H}}$  on konjugeeritud maatriks;  $\mathbf{A}^{\mathbf{H}} = (\text{kaaskompleks}(a_{j,i}))$ .

Kui  $\mathbf{A}$  on sümmeetriline ja positiivselt määratud, kasutatakse Cholesky faktoriseeringut (decomposition):  $\mathbf{A} = \mathbf{LL}^{\mathbf{H}}$ , kus  $\mathbf{L}$  on (regulaarne) alumine kolmnurkmaatriks.

N1. Kui  $\mathbf{x}$  on arv:  $\mathbf{x}^+ = 0$ , kui  $\mathbf{x} = 0$ ; vastasel korral  $\mathbf{x}^+ = \mathbf{x}^{-1}$

N2. Kui  $\mathbf{x}$  on vektor:  $\mathbf{x}^+ = \mathbf{0}$ , kui  $\mathbf{x} = \mathbf{0}$ ; vastasel korral  $\mathbf{x}^+ = \mathbf{x}^H/(\mathbf{x}^H\mathbf{x})$

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}; \quad A^+ = \begin{pmatrix} \frac{1}{5} & \frac{2}{5} \end{pmatrix}$$

Samas teised vasakpoolsed (üldistatud) pöördmaatriksid ei rahulda kõiki pseudopöördmaatriksi tingimusi.

$$A^{-1} = \begin{pmatrix} 3 & -1 \end{pmatrix}; \quad A^{-1}A = \begin{pmatrix} 1 \end{pmatrix} \quad \text{Kontrollime tingimust 3:}$$

$$(AA^{-1})^H = \begin{pmatrix} 3 & -1 \\ 6 & -2 \end{pmatrix}^T = \begin{pmatrix} 3 & 6 \\ -1 & -2 \end{pmatrix} \neq AA^{-1}$$

Moore – Penrose pöördmaatriksit kasutab **MATLAB**

Käsk: **pinv**

## Veel näiteid

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}^+ = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}^+ = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

Kui süsteemil  $Ax = b$  puudub lahend,  $m > n$ , siis leiame selle

vähimruutude meetodil:  $\min \|Ax - b\| = (Ax - b)^H (Ax - b)$

Selle ülesande lahendiks on suvaline  $x = A^\#b$ ;

Lahendi normi minimiseerib  $x = A^+b$ .

**Vabad muutujad:  $n - \text{rank}(A)$**

## Lineaarsete võrrandisüsteemide lahendamine paketiga **MATLAB**: õpijuh

**Hädavajalikud teadmised.** Peab teadma, kuidas

- sisestada või luua vektoreid ja matrikseid;
- ja milliseid tehteid saab teha vektorite ja matriksitega;
- defineerida ja leida matriksi determinanti (**det**) ja astakut (**rank**);
- leida transponeeritud matriksit ja pöördmatriksit (**inv**);
- leida vektorite skalaarkorrutist;
- kasutada koolonit matriksite ja vektoritega tegelemisel;
- hinnata ja selgitada  $n \times n$  matriksi  $A$  regulaarsust ja singulaarsust;
- hinnata ja selgitada süsteemi  $Ax = b$  konsistentsust e. kooskõlastatust;
- kirjutada välja süsteemi  $Ax = b$  formaalne lahend (Crameri peajuht);
- selgitada, et Crameri valemite kasutamine süsteemi  $Ax = b$  lahendamiseks ei ole otstarbekohane;
- kirjeldada algoritme süsteemi  $Lx = b$  või süsteemi  $Ux = b$  lahendamiseks, kus  $L$  on alumine kolmnurkmatriks ja  $U$  ülemine kolmnurkmatriks;
- kirjeldada enamkasutatavaid meetodeid süsteemi  $Ax = b$  lahendamiseks;
- defineeritakse matriksi  $A$  konditsiooniary;
- kirjutada välja hälbevektor ja selgitada selle tähendust hästi või halvasti konditsioneeritud matriksiga süsteemide puhul;
- põhjendada peaelementide väljaeraldamist Gaussi meetodi puhul;
- selgitada matriksi LU- ja Cholesky faktoriseeringuid (**lu**, **chol**);
- kirjutada m-fail, mis realiseerib LU – faktoriseeringu;
- kirjutada m-fail, mis lahendab süsteemi  $Ax = b$  Gaussi meetodil ning võrrelda selle tulemust paketti sisseehitatud vahendite abil;
- rakendada iteratsioonimeetodeid süsteemi  $Ax = b$  lahendamiseks;
- talitada siis, kui  $A$  ei ole ruutmatriks ja seda selgitada seda (**pinv**);
- kirjutada m-faili, mis realiseerib Cholesky faktoriseeringu;
- talitada siis, kui on vaja lahendada mitu süsteemi sama matriksi ja erinevate vabaliikmete vektoritega

## **Teema 5 ülesanne**

Valida lineaarne võrrandisüsteem vähemalt  $10 \times 10$  maatriksiga. Koostada m-fail (paketi **MATLAB** programm), mis realiseerib tavalise ja Seideli iteratsioonimeetodi. Uurida viimaste erinevust (iteratsioonide arvu) erinevate (vähemalt 5) täpsusastmete puhul ( $10^{-2} \dots 10^{-10}$ ). Programmis kasutada võimaluse korral paketi **MATLAB** maatrikstehteid, püüda konstrueerida lühim kood (arvestamata kohustuslikke kommentaare selles).

Tulemused salvestada ja koos algandmete ja kommentaaridega esitada. Formaat: doc.

## Funktsioonide interpoleerimine

### 1. Ülesande püstitus.

Olgu antud arvud  $n, x_1, x_2, \dots, x_n, f_1, f_2, \dots, f_n$ . Olgu fikseeritud teatud funktsioonide, **lähendfunktsioonide klass**  $M$ ; sellest täpsemalt allpool. Olgu lõpuks teada, et  $f_i = f(x_i)$ ,  $i = 1, 2, \dots, n$ , kus  $f$  on mingi ühe muutuja funktsioon, **interpoleeriv funktsioon**, mille kohta peale ülaltoodud  $n$  väärtuse ( $n$  etteantud punktis  $x_1, x_2, \dots, x_n$ ) teame veel, et see on määratud, pidev ja kuni teatud järguni (milliseni täpselt, sellest loengus) pidevalt diferentseeruv vähemalt lõigul  $[x_1, x_n]$ .

Olgu tarvis leida funktsioon  $g = g(x)$ , interpolant klassist  $M$ , mis rahuldaks **interpolatsioonitingimusi**

$$(1) \quad g(x_i) = f_i, \quad i = 1, 2, \dots, n.$$

See ongi **interpolatsiooniülesanne**.

Lisame mõned kommentaarid.

*Esiteks*, interpoleeritava funktsiooni siledus (s.t. diferentseeruvus) on tähtis eelkõige teoorias kvalitatiivsete tulemuste saamiseks; lähendfunktsiooni  $g$  määramiseks tingimuste (1) alusel need tähtsust ei oma.

*Teiseks*, algandmed tuuakse sageli sisse järgmises sõnastuses: olgu antud (nii- või teistsuguste omadustega) funktsiooni  $f$  väärtuste tabel

$$(2) \quad \begin{array}{c|cccc} x & x_1 & x_2 & \dots & x_n \\ \hline f(x) & f_1 & f_2 & \dots & f_n \end{array}.$$

*Kolmandaks*, interpolatsiooniülesande võib püstitada sõltumatult sellest, kas funktsiooni  $f$  analüütiline kuju on teada või mitte – tähtis on vaid väärtuste tabeli teadmine. Tüüpiline on siiski juht, kus analüütiline kuju pole teada; näiteks vajadus katsetulemuste asendamiseks funktsiooni analüütilise lähendiga jne.

*Neljandaks* mõned terminid: punkte  $x_1, x_2, \dots, x_n$  nimetatakse **interpolatsioonisõlmedeks**, nende hulka  $\{x_1, x_2, \dots, x_n\}$  – **võrguks**. Kui  $x_{i+1} - x_i = h > 0$ ,  $i = 1, 2, \dots, n-1$ , s.t. interpolatsioonisõlmed paiknevad järjestatult võrdsete vahemike tagant, on meil tegemist **ühtlase võrguga**.

### 2. Lähendfunktsioonide klass

Lähendfunktsioonide klass  $M$  tuleb fikseerida enne kui saab asuda interpolatsiooniülesande lahendamisele. Kuidas seda teha? Märgime kõigepealt, et põhimõtteliselt võib hulga  $M$  elementideks valida suvalised funktsioonid; sisukas teooria ja koos sellega ka sisukad tulemused saadakse siiski ainult juhul, kui  $M$  rahuldab teatud tingimusi, mida me siinkohal ära ei too. Loetleme vaid üles enamkasutatavad võimalused.

a) Hulgaks  $M$  on (ülimalt)  $n-1$ -järku polünoomide klass, s.t. lähendfunktsiooni e. interpolanti  $g$  otsitakse kujul

$$(3) \quad g(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}.$$

Saadavat ülesannet nimetatakse ka **interpolatsioonipolünoomi** leidmise ülesandeks ja lähendfunktsiooni (3) vastavalt **interpolatsioonipolünoomiks**. Selgub, et punktis 1. püstitatud ülesande korral on interpolatsioonipolünoom üheselt määratud, s.t. tingimused (1) määravad üheselt kordajate  $c_0, c_1, \dots, c_{n-1}$  väärtused kujus (3). Üldjuhul võimaldavad interpolatsioonipolünoomi välja kirjutada ka **Lagrange'i** ja **Newtoni interpolatsioonivalemid**, ühtlase võrgu korral on mugav kasutada **Newtoni valemeid** (või **Gaussi** või **Besseli valemeid**).

**Lagrange'i** ja **Newtoni interpolatsioonivalemite** tundmaõppimine on ette nähtud aine **MTMM.00.005 Numbrilised meetodid** programmis.

b) Hulgaks  $M$  on trigonomeetriliste polünoomide klass. Lähendfunktsiooni kuju:

$$(4) \quad g(x) = c_0 + \sum_{k=1}^s (c_k \cos kx + d_k \sin kx).$$

Määrata tuleb kordajad  $c_0, c_1, \dots, c_s, d_1, \dots, d_s$ , parameeter  $s$  sõltub interpolatsioonisõlmede arvust  $n$ . Trigonomeetrist interpolatsioonipolünoomi on õige kasutada perioodiliste funktsioonide interpoleerimiseks. Vastavat ülesannet nimetatakse **trigonomeetrilise interpolatsioonipolünoomi** leidmise ülesandeks ning funktsiooni (4) **trigonomeetriliseks interpolatsioonipolünoomiks**.

c) Hulgaks  $M$  on sõlmedele  $x_1, x_2, \dots, x_n$  vastavate esimest (teist, kolmandat) järku **splainide** klass. See on "moodne", üha enam kasutatav võimalus. Tegemist on siin **linearse (ruut-, kuup-) interpolatsioonisplaini** leidmise ülesandega.

Klassi  $M$  määramise sisukad võimalused pole nende kolme näitega muidugi ammendatud.

**3. Millele pööratakse tähelepanu arvutusmeetodite (AM) teoorias interpolatsiooniülesannete puhul?**

Konstruksiooni põhjal lõikuvad või puutuvad interpoleeritava funktsiooni  $f$  ja interpoleeriva funktsiooni  $g$  graafikud punktides  $(x_1, f_1), (x_2, f_2)$ ,

$\dots, (x_n, f_n)$ . Funktsiooni  $f$  "asendamisel" interpolandiga  $g$  võib olla kaks (teineteist mittevälisavat) eesmärki:

- on vaja leida funktsiooni  $f$  väärtus mingis punktis sõlmede vahel (või nendest "vasakul" või "paremal" – nn. **ekstrapolatsiooniülesanne**); selle lähisväärtuseks loetakse funktsiooni  $g$  väärtus antud punktis;
- funktsiooniga  $f$  on vaja edaspidi teha analüütilisi tehteid – teisendusi; funktsiooni  $f$  analüütilise kuju asemel kasutatakse funktsiooni  $g$  oma.

Nii ühe kui teise eesmärgi realiseerimiseks on vaja garanteerida funktsioonide  $f$  ja  $g$  väärtuste (graafikute) lähedus ka väljaspool sõlmi. Teooria annabki hinnangu suurusele

$$\max_{x_1 \leq x \leq x_n} |f(x) - g(x)|,$$

samuti tingimused, millistel see hinnang kehtib. Uuritakse, kas ja kunas see suurus koondub nulliks protsessis  $n \rightarrow \infty$  (kuid olukorras, kus äärmiste sõlmede vahe  $x_n - x_1$  jääb muutumatuks). Vastustest nendele küsimustele selgubki, miks üks lähendfunktsioonide klass sobib, teine mitte; miks ja mida on vaja eeldada funktsiooni  $f$  kohta jne.

#### 4. Interpolatsiooniülesande üldistusi.

##### a) Interpoleerimine kordsete sõlmede korral.

Niisugust nime kandev ülesanne tekib, kui peale funktsiooni  $f$  väärtuste  $f_1, f_2, \dots, f_n$  on mingis (või mõnes või koguni kõigis) sõlmedes teada ka funktsiooni  $f$  tuletise (või tuletiste kuni teatava järguni) väärtuse(ed). Kui näiteks sõlmes  $x_i$  on teada väärtused  $f_i = f(x_i)$ ,  $f'_i = f'(x_i)$ ,  $\dots$ ,  $f_i^{(k-1)} = f^{(k-1)}(x_i)$ , siis nimetatakse sõlme  $x_i$  **k-kordseks interpolatsioonisõlmeks**. Punktis  $I$  püstitatud ülesande korral on kõik sõlmed ühekordsed.

Interpolatsioonipolünoomi kordsete sõlmede korral võimaldavad välja kirjutada **Hermite'i** ja **Newtoni interpolatsioonivalemid**.

b) Saab püstitada ja lahendada ka **mitme muutuja funktsioonide interpoleerimise ülesande**.

#### 5. Märkus funktsioonide lähendamise üldiste ülesannete kohta.

Interpolatsiooniülesannet võib vaadelda erijuhuna ühest üldisemast ülesannete klassist. Kõiki selle klassi ülesandeid nimetatakse **funktsioonide lähendamise ülesanneteks** ning neid kõiki võib sõnastada järgmise skeemi kohaselt:

olgu antud teatud info funktsiooni  $f$  kohta; olgu fikseeritud mingi lähendfunktsioonide klass  $M$ . On vaja leida funktsioon klassist  $M$ , mis teatavas mõttes oleks võimalikult lähedal (kui võimalik, siis lähim hulga  $M$  elementide seast) funktsioonile  $f$ .

Täpsustame. "Teatud info ...  $f$  kohta" tähendab eelkõige funktsiooni  $f$  kuulumist kindlasse (kas abstraktsesse või konkreetse) funktsioonide ruumi,

s.t. kvalitatiivset teavet ühelt poolt; teisalt peab praktilisteks arvutusteks olema ette antud ka rida kvantitatiivseid andmeid (väärtuste tabel jne.)

Lähendfunktsioonide klassi  $M$  kohta on põhiline öeldud punktis 2.

Mitmeid võimalusi pakub funktsiooni  $f$  ja lähendfunktsiooni läheduse kriteeriumi fikseerimine. See kriteerium peab olema kindlas kooskõlas nii funktsiooni  $f$  kui ka klassi  $M$  omadustega. Näiteks interpolatsioonitingimused (1) on ka üheks selliseks kriteeriumiks; siin ei saa lähendfunktsiooni otsida näiteks ülimalt  $n - 2$  - järku polünoomide hulgast - üldjuhul poleks see ülesanne siis lahenduv. Peale interpolatsioonitingimuste mainime veel kaht tähtsamat kriteeriumi.

- **Parima ühtlase lähendamise kriteerium** nõuab sellise funktsiooni  $g$  leidmist klassist  $M$ , mis minimiseeriks suuruse

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

- **Parima ruutkeskmise lähendamise kriteerium** nõuab sellise  $g \in M$  leidmist, mis minimiseeriks suuruse

$$\int_a^b [f(x) - g(x)]^2 dx.$$

Sellistel kujudel antud kriteeriumite tähtsus on eelkõige teoreetiline. Praktikas esineb sageli ruutkeskmise kriteeriumi diskreetne analoog, nn. **Vähimruutude kriteerium**. Selle kohaselt tuleb leida  $g \in M$ , mis minimiseeriks summa

$$\sum_{i=1}^n (f(x_i) - g(x_i))^2;$$

siin  $x_1, x_2, \dots, x_n$  on etteantud sõlmed ja muidugi peavad olema teada ka funktsiooni  $f$  väärtused nendes.

[peep.miidla@ut.ee](mailto:peep.miidla@ut.ee)

## Interpolatsioonipolünoom

Interpolatsioonipolünoomi Lagrange'i kuju on selline:

$$P_n(x) = \sum_{i=0}^n l_i(x) f(x_i),$$

Kordajad  $l_i$  arvutatakse järgmiselt:

$$l_i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}, \quad i = 0, \dots, n$$

Jääkliige:

$$R_n(x) = f(x) - P_n(x)$$

Jääkliikme hinnang:

$$|R_n(x)| \leq \frac{|\omega(x)|}{(n+1)!} \max_{x \in [a,b]} |f^{(n+1)}(x)|,$$

kus  $\omega(x) = (x-x_0)(x-x_1)\dots(x-x_n)$ .

## Interpolatsioonipolünoomi Newtoni kuju

Tähistame diferentssuhted, neid arvutatakse rekursiivselt.

- Järku null (funktsiooni väärtus sõlmes):

$$f[x_0] = f(x_0)$$

- Esimest järku:

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

- Teist järku:

$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

- Järku m:

$$f[x_m, \dots, x_0] = \frac{f[x_m, \dots, x_1] - f[x_{m-1}, \dots, x_0]}{x_m - x_0}$$

Interpolatsioonipolünoomi Newtoni kuju on selline:

$$f_n(x) = b_0 + b_1(x - x_0) + \dots + b_n(x - x_0)(x - x_1)\dots(x - x_{n-1})$$

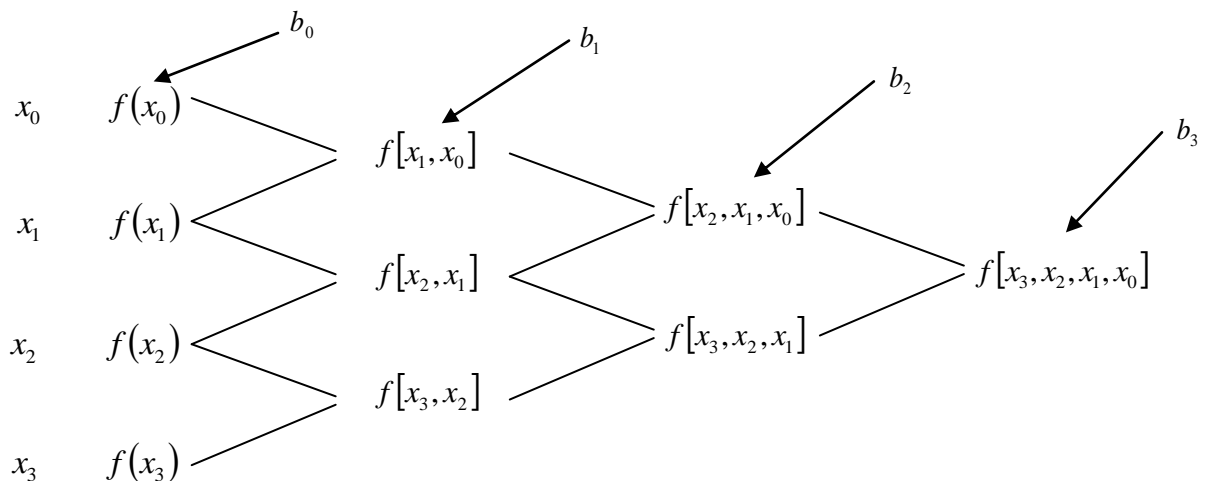
kus

$$b_0 = f[x_0], \quad b_1 = f[x_1, x_0], \quad b_2 = f[x_2, x_1, x_0],$$

$$b_{n-1} = f[x_{n-1}, x_{n-2}, \dots, x_0]$$

$$b_n = f[x_n, x_{n-1}, \dots, x_0]$$

Joonisel: rekurrentne arvutuskeem diferentssuhete leidmiseks kolmandat järku polünoomi jaoks.



Kolmandat järku interpolatsioonipolünoom ise:

$$f_3(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1) + f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)$$

## Interpolatsioonipolünoom: kohustuslik ülesanne.

Leida Lagrange'i interpolatsioonipolünoomi abil tabelina antud funktsiooni ligikaudne väärtus etteantud kohal. Andmed valida ise.

Konstrueerida paketi **MATLAB** vahenditega erinevat järku lähendpolünoome. Joonistada täpsed ja lähendfunktsioonid samasse jooniseaknasse.

NB! Tuleb valida mitteühtlase sammuga tabel, soovitavalt tuntud funktsiooni väärtustega:

$x$	$x_1$	$x_2$	$\dots$	$x_n$
$f(x)$	$f_1$	$f_2$	$\dots$	$f_n$

Funktsiooniks  $f$  võiks olla näiteks trigonomeetriline funktsioon, aga tabeliks võib võtta ka näiteks börsinformatsiooni..

## Interpoleerimisel kasutatavad funktsioonid pakettis **MATLAB**

Polünoom esitatakse pakettis **MATLAB** selle kordajate vektorina:

Näiteks polünoom  $x^3 - 3x^2 + 7x - 3$  esitub vektorina  $p=[1 -3 7 -3]$ .

**abs** absoluutväärtuse leidmine;

**inline** INLINE funktsioon,  $\gg g = inline('t^2');$

**interp1** (ühedimensionaalne) interpolatsioon;

**linspace** ühtlase sammuga vektor,  $\gg y = linspace(1,10)$

**poly** juurte järgi polünoom,  $\gg p = poly(r); r -$  juured;

**polyval** polünoomi väärtuste leidmine,  $\gg y = polyval(p,x); p -$  kordajad;

**ppval** tükiti polünoomi väärtuste leidmine;

**roots** polünoomi juurte leidmine,  $\gg r = roots(p); p -$  kordajad;

**spline** interpolateerimine kuupsplainiga.

# Numbriline diferentseerimine

Esitluse põhi: Kerttu Maidla 2009

[peep.miidla@ut.ee](mailto:peep.miidla@ut.ee)

# Mis on numbriline diferentseerimine?

- Numbrilise diferentseerimise all mõistetakse tabelina esitatud funktsiooni tuletiste väärtuste arvutamist.
- Numbrilise diferentseerimise valemeid kasutatakse ka siis, kui funktsioon on antud keerulise analüütilise avaldisega.
- Valemite tuletamisel lähendatakse antud funktsioon teatava lihtsa analüütiliselt antud funktsiooniga ning arvutatakse viimase tuletised.

# MATLABi funktsioon `diff()`

- Arvutamaks numbrilist tuletist on pakett MATLAB funktsioon `diff(X)`
- See arvutab vektori  $X$  lähedaste väärtuste vahed
- Näiteks kui on  $X = [1\ 2\ 3\ 4\ 5]$ , siis `diff(X) = 1 1 1 1`
- Niisiis on `diff(X)` poolt tagastatav vektor ühe elemendi võrra lühem:  $[X(2)-X(1)\ X(3)-X(2)\ \dots\ X(n)-X(n-1)]$
- Kui  $x$  on maatriks, siis tagastatakse ridade vahede maatriks:  $[X(2:m,:)-X(1:m-1,:)]$ , mis on ühe rea võrra algsest maatriksist lühem.

# diff()

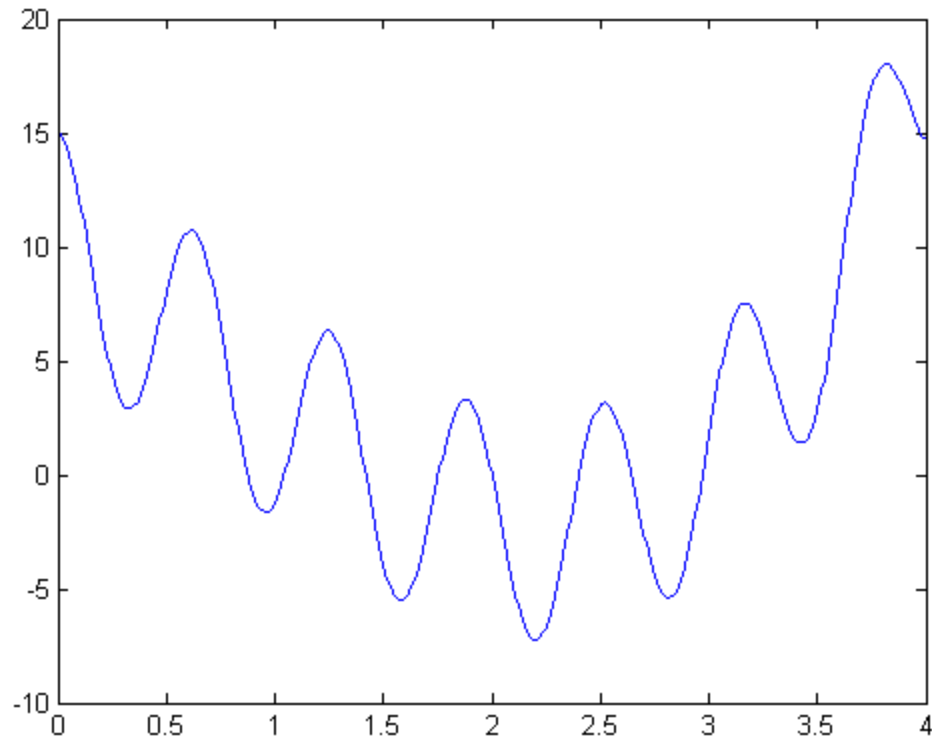
- $\text{diff}(X,n)$  – kordab massiivil  $X$  funtsiooni  $\text{diff}()$  rekursiivselt  $n$  korda, saades tulemuseks  $n$ -inda vahe;  $\text{diff}(X,2) = \text{diff}(\text{diff}(X))$
- Näiteks kui  $X = [1\ 2\ 3\ 4\ 5]$ , saame tulemuseks:  $\text{diff}(X,2) = 0\ 0\ 0$
- Kui funktsiooni väärtused on vektoris  $y$  ja vastavad argumendi väärtused vektoris  $x$ , siis ligikaudne tuletis on:  $\text{deriv}_y = \text{diff}(y) ./ \text{diff}(x)$ ;
- Näiteks:  $X = [1, 2, 3, 4]$  ja  $Y = [4, 8, 3, 9]$
- Siis:  $\text{deriv}_y = \text{diff}(y) ./ \text{diff}(x)$ ;
- Annab tulemuseks:  $\text{deriv}_y = 4\ -5\ 6$

# Näide:

- `x = 0:.01:4;`
- `y = 5*cos(10*x) + x.^3 - 2*x.^2 - 6*x + 10;`
- `plot(x,y)`
- `deriv_y = diff(y)./diff(x);`
- `xd = x(2:length(x)-1);`
- `figure`
- `xd = x(2:length(x));`
- `plot(xd,deriv_y)`

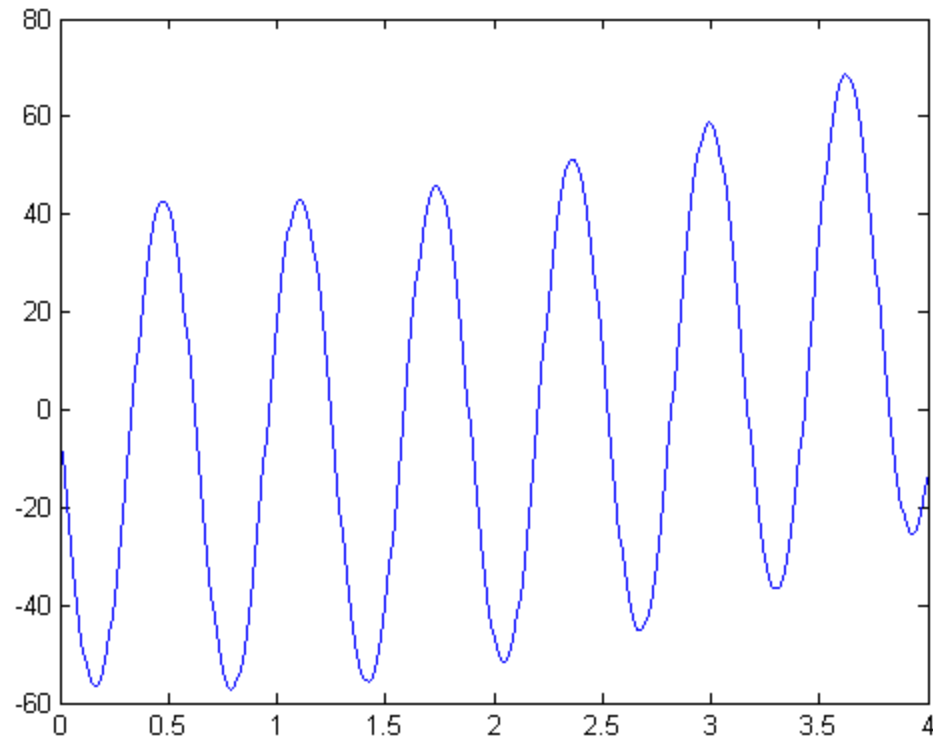
# Polynomial-Trig Function

Polünominaalne - trigonomeetiline funktsioon



# Polynomial-Trig Function Derivative

Polünominaalse - trigonomeetrilise funtsiooni tuletis



## Numbriline diferentseerimine

**Ülesande püstitus** on üsna sarnane interpolatsiooniülesande püstitusele.

Olgu antud ühe muutuja funktsiooni  $f$  väärtuste tabel

$x$	$x_1$	$x_2$	$\dots$	$x_n$
$f(x)$	$f_1$	$f_2$	$\dots$	$f_n$

s.t.  $f_i = f(x_i)$ ,  $i = 1, 2, \dots, n$ . Funktsiooni  $f$  kohta peame eeldama, et see on määratud, pidev, pidevalt diferentseeruv (vähemalt) lõigul  $[x_1, x_n]$ . Teoreetiliste tulemuste saamiseks tuleb eeldada ka mõnede kõrgemat järku tuletiste olemasolu ja pidevust sellel lõigul.

Vaja on leida funktsiooni tuletise väärtus  $f'(x)$  etteantud punktis  $x \in [x_1, x_n]$  (või ka mitmes etteantud punktis).

**Ülesande lahendamise** üldiseks võtteks on funktsioonile  $f$  interpolandi leidmine; interpolandi tuletise väärtus punktis  $x$  loetaksegi arvu  $f'(x)$  lähisväärtuseks. Nimetatud interpolandina interpolatsioonipolünoomi kasutamine on teatud mõttes mittekorrektne. Küllalt edukalt saab kasutada kuupsplaine.

“Hästi lahenduv” ja sagedamini ette tulev (vt. Näiteks diferentsiaalvõrranditega seotud ülesannete numbrilisi lahendusmeetodeid) on nn. **Sõlmedes numbrilise diferentseerimise** ülesanne, s.t. juht, kus funktsiooni  $f$  tuletiste väärtus tuleb sama alginfo põhjal leida ühes või mitmes sõlmes.

Saab konstrueerida valemeid ka kõrgemat järku tuletiste leidmiseks sõlmedes.

## Numbriline diferentseerimine

Kohustuslik ülesanne.

Leida valitud funktsiooni tabeli abil selle funktsiooni esimese ja teise tuletise väärtused tabeli sõlmes. Kasutada diferentseerimise valemeid ja paketi **MATLAB** funktsiooni **diff**.

Võrrelda leitud ligikaudseid väärtusi täpsetega.

## Numbrilise diferentseerimise valemid

Sammuga ette:

$$f'(x_i) \approx (y_{i+1} - y_i)/(x_{i+1} - x_i) .$$

Sammuga taha:

$$f'(x_i) \approx (y_i - y_{i-1})/(x_i - x_{i-1}) .$$

Tsentraalne:

$$f'(x_i) \approx (y_{i+1} - y_{i-1})/(2h), \text{ kus } h = x_i - x_{i-1} .$$

Teine tuletis:

$$f''(x_i) \approx (y_{i+1} - 2y_i + y_{i-1})/h^2 .$$

## Vähimruutude meetod paketiga **MATLAB**

### õpijuhhis

**Hädavajalikud teadmised.** Peab teadma, kuidas

- defineerida lähendfunktsiooni vähimruutude mõttes (mida minimiseeritakse?);
- arvutada lineaarse lähendfunktsiooni parameetreid: tõusu ja vabaliiget;
- lahendada üledetermineeritud (võrrandeid rohkem kui tundmatuid) lineaarset võrrandisüsteemi  $Ax = b$ ; selleks:  $\gg x = A \setminus b$ ; või  $\gg x = \text{pinv}(A) * b$ ;
- kasutada paketi **MATLAB** sisseehitatud funktsiooni polyfit;
- kasutada paketi **MATLAB** sisseehitatud funktsiooni polyval polynoomi väärtuste arvutamiseks etteantud punktis x;
- joonistada graafikut, millel on nii algandmed kui ka lähendpolynoom(id);
- kirjutada välja ülemääratud lineaarne võrrandisüsteem vähimruutude lähendi leidmiseks suvaliste baasfunktsioonide korral;

## Funktsioonide lähendamine **vähimruutude** meetodil Paketi **MATLAB** vahendid

Ülesande püstitus on selline:

Olgu lõigul  $[a, b]$  antud **võrk sõlmedega**  $a = x_0 < x_1 < \dots < x_n = b$  ja olgu sõlmedes teada funktsiooni  $f$  väärtused  $f_i = f(x_i)$ ;  $i = 0, 1, \dots, n$ . Teisiti öeldes: olgu antud funktsiooni väärtuste tabel:

$x$	$x_1$	$x_2$	$\dots$	$x_n$
$f(x)$	$f_1$	$f_2$	$\dots$	$f_n$

Paketis **MATLAB** realiseerib vähimruutude meetodi lähendpolünoomi leidmiseks funktsioon **polyfit**.

Näide.

```
>>x = -1:1:1; % Argumentide vektor.
>>y = exp(x); % Funktsiooni väärtused.
>>plot(x,y,'bo') % Joonis algandmetest
>>xlabel Argument % X-telje nimi.
>>ylabel Funktsioon % Y-telje nimi.
>>grid on % Võrk.
>>title 'Eksponentsiaalsed andmed' % Joonise pealkiri.
>>P1 = polyfit(x,y,1) % Esimest järku lähendpolünoom.
>>P2 = polyfit(x,y,2) % Teist järku lähendpolünoom.
>>ylineaarne = polyval(P1,x); % Lineaarpolünoomi väärtused.
```

```
>>plot(x,y,'bo',x,ylineaarne,'r-') % Joonis.
>>xlabel Argument % X-telje nimi.
>>ylabel Funktsioon % Y-telje nimi.
>>grid on % Võrk.
>>title 'Lineaarne lähendpolünoom' % Joonise pealkiri.
>>yruut = polyval(P2,x); % Ruutpolünoomi väärtused.
>>plot(x,y,'bo',x,yruut,'r-') % Joonis.
>>xlabel Argument % X-telje nimi.
>>ylabel Funktsioon % Y-telje nimi.
>>grid on % Võrk.
>>title 'Teist järku lähendpolünoom' % Joonise pealkiri.
>>vead = y - ylineaarne; % Arvutatakse vead sõlmedes ehk jääkliikmed.
>>plot(x,vead,'bo') % Jääkliikmete (residuals) joonis.
>>xlabel Argument % X-telje nimi.
>>ylabel Jääkliikmed % Y-telje nimi.
>>grid on % Võrk.
>>title 'Lineaarpolünoomi jääkliikmed' % Joonise pealkiri.
```

[peep.miidla@ut.ee](mailto:peep.miidla@ut.ee)

## Funktsioonide lähendamine **vähimruutude** meetodil

Olgu lõigul  $[a, b]$  antud **võrk sõlmedega**  $a = x_0 < x_1 < \dots < x_n = b$  ja olgu sõlmedes teada funktsiooni  $f$  väärtused  $f_i = f(x_i)$ ;  $i = 0, 1, \dots, n$ . Otsime funktsiooni  $g = g(x)$  mingist kindlast funktsioonide klassist  $\mathbf{M}$ , mis **vähimruutude** mõttes lähendaks funktsiooni  $f$ . Kui interpolatsioonitingimused nõuaksid, et funktsioonide  $g$  ja  $f$  väärtused sõlmedes langeksid kokku, siis vähimruutude mõttes lähendab funktsioon  $g$  funktsiooni  $f$  siis, kui summa

$$J(f,g) = \sum_{i=1}^n (f(x_i) - g(x_i))^2 \quad (1)$$

Omandab minimaalse väärtuse üle kõigi funktsioonide klassist  $\mathbf{M}$ . Suurus (1) kannab **vähimruutude funktsionaali** nime, tingimust

$$\min J(f,g), g \in \mathbf{M} \quad (2)$$

aga nimetatakse **vähimruutude kriteeriumiks**.

Lähendfunktsioonide klass  $\mathbf{M}$  tuleb fikseerida enne kui saab asuda lähendamisülesande lahendamisele. Märgime kõigepealt, et põhimõtteliselt võib hulga  $\mathbf{M}$  elementideks valida suvalised funktsioonid; sisukas teooria ja koos sellega ka sisukad tulemused saadakse siiski ainult juhul, kui  $\mathbf{M}$  rahuldab teatud teoreetilisi tingimusi, mida me siinkohal ära ei too. Loetleme vaid üles enamkasutatavad võimalused lähendfunktsioonide hulga  $\mathbf{M}$  määratlemiseks.

a) Hulgaks  $\mathbf{M}$  on (ülimalt)  $n-1$ -järku **polünoomide** klass, s.t. lähendfunktsiooni  $g$  otsitakse kujul

$$g(x) = c_1 + c_2x + \dots + c_kx^k, \quad k \leq n-1. \quad (3)$$

Märgime, et **interpolatsioonipolünoomi** leidmise ülesandes  $k = n-1$  ja lähendfunktsiooni (3) nimetatakse siis vastavalt **interpolatsioonipolünoomiks**. Vähimruutude mõttes lähendi leidmisel võetakse sageli  $k = 1$  (lineaarse regressiooni ülesanne) või  $k = 2$  (ruutlähendi

leidmise ülesanne. Sel juhul tingimus (2) ei määra reeglina (st. kui  $k < n-1$ ) kordajate  $c_1, c_2, \dots, c_k$  väärtusi üheselt kujus (3).

b) Hulgaks  $\mathbf{M}$  on trigonomeetriliste polünoomide klass. Lähendfunktsiooni kuju:

$$g(x) = c_0 + \sum_{k=1}^s (c_k \cos kx + d_k \sin kx). \quad (4)$$

Määrata tuleb kordajad  $c_0, c_1, \dots, c_s, d_1, \dots, d_s$ , parameeter  $s$  sõltub interpolatsioonisõlmede arvust  $n$ . Trigonomeetrist lähendpolünoomi on õige kasutada perioodiliste funktsioonide aproksimeerimiseks.

c) Üldiselt võib vähimruutude meetodil otsida lähendfunktsiooni  $g$  funktsioonide klassist, mis esituvad kujul

$$g(x) = \sum_{i=1}^m c_i \psi_i(x),$$

kus  $\psi_1, \psi_2, \dots, \psi_m$  on etteantud, nn koordinaatfunktsioonid ja  $c_1, c_2, \dots, c_m$  on rittaarenduse kordajad, aproksimatsiooniülesandes otsitavad.

Funktsiooni  $\mathbf{J}(f, g)$  miinimumi tarvilik tingimus on

$$\frac{\partial J}{\partial c_k} = 0, \quad k = 1, 2, \dots, m.$$

Kordajad  $c_1, c_2, \dots, c_m$  saab määrata lineaarsest võrrandisüsteemist

$$\sum_{k=1}^m \left( \sum_{i=1}^{n-1} \psi_k(x_i) \psi_j(x_i) \right) c_k = \sum_{i=1}^{n-1} f_i \psi_j(x_i), \quad j = 1, 2, \dots, m.$$

Võib vaadelda ka kaalutud vähimruutude meetodit. Sel juhul minimiseeritakse kaalutud summat

$$\sum_{i=1}^n \omega_i (f(x_i) - g(x_i))^2.$$

Vähimruutude meetod, kohustuslik ülesanne.

Valida funktsioon (trigonomeetiline, ...) ja konstrueerida algandmete tabel (vähemalt 20 elemendist).

Selle põhjal konstrueerida paketi **MATLAB** abil erinevat järku lähendpolünoome ( $n=1, 2, 5, \dots$ ) ja kujutada nii algandmed kui ka lähendpolünoomid ühel joonisel.

Leida jääkliikmed ja kujutada need teisel joonisel.

## Numbriline integreerimine

### 1. Ülesande püstitus.

Olgu fikseeritud lõik  $[a, b]$ ,  $a < b$  (üldiselt võib olla ka  $a = -\infty$  ja/või  $b = \infty$ ).

Olgu fikseeritud **kaalufunktsioon**  $p = p(x)$ , mis rahuldab tingimusi:

a)  $p(x) > 0$ ,  $x \in (a, b)$ ;

b) integraalid  $\int_a^b p(x)x^k dx$  eksisteerivad (st. on lõplikud) iga  $k = 0, 1, \dots$

korral.

Vaatleme suvalist ühemuutuja funktsiooni  $f = f(x)$ , mis on määratud, pidev ja teatud arv korda (kui mitu korda – sellest loengus) pidevalt diferentseeruv lõigul  $[a, b]$ .

On vaja leida (arv)

$$(1) \quad J(f) = \int_a^b p(x)f(x)dx.$$

Lõiku  $[a, b]$  (lõpmatute  $a$  ja/või  $b$  korral vastavalt vahemikku) nimetatakse **integreerimislõiguks** (**-vahemikuks**).

Eeldame edaspidi, et  $a$  ja  $b$  on lõplikud.

### 2. Lähislahend

Kui on teada korrutise  $pf$  algfunktsioon, siis saab määratud integraali väärtuse (1) leida **Newton – Leibnizi valemi** abil. Teatavasti pole aga aga kaugetki mitte kõikide integreeruvate funktsioonide algfunktsioonid elementaarfunktsioonide kaudu avalduvad. Seetõttu püstitatakse teoorias ülesanne leida arv  $J(f)$  ligikaudu. Täpsemalt, arvutusmeetodite teoorias konstrueeritakse valemeid, mis võimaldavad leida sellise arvu  $J_n(f)$ , et  $J(f) \approx J_n(f)$ . Võib kirjutada ka

$$(2) \quad J(f) = J_n(f) - R_n(f).$$

Valemid, mida kasutatakse **lähisväärtuse**  $J_n(f)$  leidmiseks, kannavad nimetust **kvadratuurvalemid** ning nende küllaltki üldine kuju on selline:

$$(3) \quad J_n(f) = \sum_{i=1}^n A_i f(x_i).$$

Suurust  $R_n(f)$  seoses (2), mis iseloomustab integraali täpse väärtuse ja ligikaudse väärtuse erinevust, nimetatakse **kvadratuurvalemi jääkliikmeks**.

Suurusi  $A_i$  ( $i = 1, 2, \dots, n$ ) nimetatakse **kvadratuurvalemi kordajateks** ja suurusi  $x_i$  ( $i = 1, 2, \dots, n$ ) – **kvadratuurvalemi sõlmeks**.

### 3. Kvadratuurvalemid.

Seos (3) esitab kvadratuurvalemi üldise kuju. Et tuletada konkreetset kvadratuurvalemit, mille abil saaks reaalselt rehkendada, tuleb fikseerida kaalufunktsioon  $p$ , integreerimisloik  $[a, b]$  ja parameeter  $n$ . Edasi tuleb otsustada sõlmede üle: kas need fikseerida enne valemi tuletamist, või tuletuse käigus leida. Kui see kõik on tehtud, võibki asuda konkreetse kvadratuurvalemi tuletamisele.

Kvadratuurvalem on tuletatud, kui on leitud kordajate ja sõlmede arvulised väärtused valemis (3) ning jääkliikme kuju seoses (2). Kui kvadratuurvalem (3) on juba tuletatud, siis jääkliige  $R_n(f)$  omandab konkreetse kuju, mis sõltub “parameetritest”  $p, n, a, b, x_1, x_2, \dots, x_n$ , samuti aga ka integreeritavast funktsioonist  $f$  (täpsemalt – selle teatud järku tuletisest). Osutub, et kõikide tehtud eelduste täidetuse korral ( $p(x) > 0 \quad \forall x; a < b$ ) leidub veel funktsioone, millede korral  $R_n(f) \neq 0$ ; meenutame veel, et funktsiooni  $f$  kohta nõudsime “vaid” määratud ja teatud järguni pidevat diferentseerumist lõigul  $[a, b]$ .

Kvadratuurvalemi tuletamisel on üheks võimaluseks fikseerida konkreetset funktsioonid (funktsioonide klassid), millede korral jääkliige peab kindlasti võrduma nulliga. Selliste funktsioonide korral, niisiis,

$$J(f) = \sum_{i=1}^n A_i f(x_i).$$

Funktsioonideks, millede korral jääkliige peab võrduma nulliga, valime loengus võimalikult kõrget järku polünoomid. Milline on see “võimalikult kõrge järk”, sõltub eelkõige sellest, kas kvadratuurvalemi sõlmed on ette fikseeritud või ei.

Selgitame pisut järgmises punktis.

### 4. Kvadratuurvalemite tuletamine.

Olgu fikseeritud kaalufunktsioon  $p$  ja parameetrid  $n, a, b$ . Lihtne on valemisse asendamise teel veenduda, et

$$(4) \quad R_n(\alpha f + \beta g) = \alpha R_n(f) - \beta R_n(f),$$

kus  $\alpha$  ja  $\beta$  on suvalised konsatandid,  $f$  ja  $g$  aga funktsioonid. Kvadratuurvalemi tuletamiseks, mis oleks täpne võimalikult kõrget järku polünoomide korral, tuleb nüüd lahendada algebraline võrrandisüsteem

$$(5) \quad R_n(x^k) = 0, \quad k = 0, 1, \dots, m,$$

kus  $m + 1$  on tundmatute parameetrite arv:  $m + 1 = n$ , kui sõlmed on fikseeritud ja  $m + 1 = 2n$  “vabade” sõlmede korral. Seose (4) põhjal  $R_n(\sum_{j=0}^m \alpha_j x^j) = 0$ , s.t.  $m$

ongi see "võimalikult kõrge järk"; siin  $\alpha_j$  ( $j = 0, 1, \dots, m$ ) on  $m$ -järku polünoomi kordajad, suvalised konstandid.

Kirjutame süsteemi (5) täpsemalt välja valemeid (1), (2) ja (3) arvestades:

$$\int_a^b p(x) dx = A_1 + A_2 + \dots + A_n \quad (\Leftrightarrow R_n(x^0 = 1) = 0),$$

$$\int_a^b p(x)x dx = A_1 x_1 + A_2 x_2 + \dots + A_n x_n \quad (\Leftrightarrow R_n(x) = 0),$$

$$\int_a^b p(x)x^2 dx = A_1 x_1^2 + A_2 x_2^2 + \dots + A_n x_n^2 \quad (\Leftrightarrow R_n(x^2) = 0),$$

.....

$$\int_a^b p(x)x^m dx = A_1 x_1^m + A_2 x_2^m + \dots + A_n x_n^m \quad (\Leftrightarrow R_n(x^m) = 0).$$

Näeme, et kui sõlmed  $x_1, x_2, \dots, x_n$  on fikseeritud ( $m+1 = n$ ), siis osutub see süsteemiks kvadratuurvalemi kordajate  $A_1, A_2, \dots, A_n$  suhtes ning seda on lihtne lahendada. Kui aga ka sõlmed tuleb leida, on tegu mittelineaarse algebralise võrrandisüsteemiga, mille lahendamiselka tuleb kasutada spetsiaal-seid võtteid.

Huvi korral palun jääkliikme avaldiste tuletamist uurida iseseisvalt.

### 5. Mõningaid kvadratuurvalemite tüüpe.

a) Kui on fikseeritud kaalufunktsioon  $p$  ja parameetrid  $n, a, b$  ning tuleb leida kvadratuurvalemi sõlmed, kordajad (süsteemist (5)) ja jääkliige, saame **kõrgeima algebralise täpsusastmega e. Gaussi tüüpi kvadratuurvalemi**.

b) Kui eelmises situatsioonis on fikseeritud konkreetselt  $p(x) \equiv 1$ , same ka konkreetselt **Gaussi kvadratuurvalemi** (mis on muidugi samuti kõrgeima algebralise täpsusastmega e. Gaussi tüüpi).

c) Kui on fikseeritud  $p(x) \equiv 1$ , parameetrid  $n, a, b$  ja sõlmed  $x_i = a + (i-1) \frac{b-a}{n-1}$ ,  $i = 1, 2, \dots, n$  ning tuleb leida vaid kvadratuurvalemi kordajad (süsteemist (5)) ja jääkliige, saame **Newton - Cotesi kvadratuurvalemi**. Need on ka kõige sagedamini tarvitusel.

## Numbriline integreerimine

Kohustuslik ülesanne

Leida valitud määratud integraali lähisväärtus trapetsvalemiga

$$\int_a^b f(x)dx \approx h \left( \frac{f_0}{2} + f_1 + \dots + f_{N-1} + \frac{f_N}{2} \right)$$

ja Simpsoni valemiga.

$$\int_a^b f(x)dx \approx \frac{h}{3} \sum_{k=1}^m (f_{2k-1} + 4f_{2k} + f_{2k+1})$$

Valida erinevaid täpsusi ning leida selle saavutamiseks vajalik sammude arv ja sammupikkus. Hindamisel kasutada Runge võtet.

Väljastada täpsuse saavutamiseks vajalike sõlmede arv, integraali lähisväärtused ja veahinnang. Võrrelda tulemusi täpse väärtusega. Leida integraali väärtusi paketi **MATLAB** funktsiooniga **quad**.

## Numbriline integreerimine paketi **MATLAB** abil

**quad** %Simpsoni meetod

```
>>F = @(x)1./(x.^3-2*x-5); % Funktsiooni defineerimine inline formaadis.
```

```
>>Q = quad(F,0,2); % Integreerimine.
```

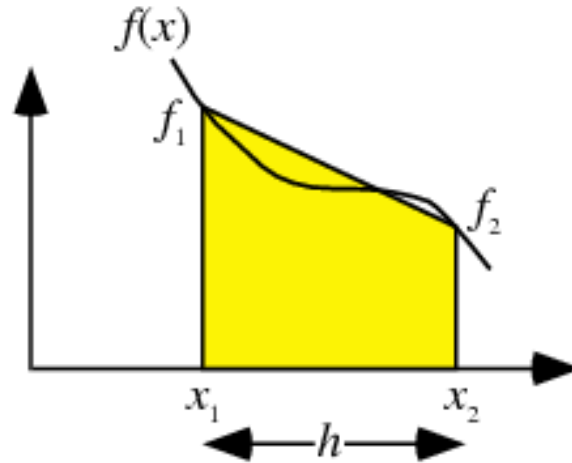
**quadl** % Lobatto kvadratuurvalem.

```
>>q = quadl(F,0,2) % Integreerimine.
```

**inline** % Inline funktsiooni defineerimine.

**dblquad** % Kahekordse integraali arvutamine.

Newton-Cotes'i kvadratuurvalemid



Trapetsvalem: joone  $y = f(x)$  alune pindala asendatakse trapetsitega  $x_1 \rightarrow x_2 \rightarrow f_2 \rightarrow f_1$ .

Üldine lähenemine: funktsioon  $f$  asendatakse interpolatsioonipolünoomiga ja integreeritakse seda.

Trapetsvalem:

$$\int_a^b f(x) dx = h(0.5f_1 + f_2 + f_3 + \dots + f_{n-1} + 0.5f_n) - [(b-a)h^2 f''(\xi)]/12.$$

Simpsoni valem:

$$\int_a^b f(x) dx = \frac{h}{3} \sum_{k=1}^m (f_{2k-1} + 4f_{2k} + f_{2k+1}) - \frac{1}{90} h^5 \sum_{k=1}^m f^{IV}(\xi_k),$$

ehk

$$\int_a^b f(x) dx = (h/3)(f_1 + 4f_2 + 2f_3 + \dots + 4f_{n-1} + f_n) - [(b-a)h^4 f^{IV}(\xi)]/180.$$

Ristkülikvalem:

$$\int_a^b f(x) dx = h(f_{3/2} + f_{5/2} + f_{7/2} + \dots + f_{n-1/2}) - [(b-a)h^2 f''(\xi)]/24.$$

## Numbriline integreerimine paketiga **MATLAB**: õpjuhis

**Hädavajalikud teadmised.** Peab teadma, kuidas

- sisestada või luua vektoreid ja matrikseid ja kuidas nendega tehteid teha;
- leida matriksi determinanti (**det**), astakut (**rank**) ja pöördmatriksit (**inv**);
- defineerida määratud integraal;
- kasutada Newton-Leibnizi valemit;
- defineerida määratud integraali lähisväärtust;
- tuletatakse Newton- Cotesi kvadratuurvalemeid;
- defineeritakse kõrgeima täpsusastmega kvadratuurvalemeid;
- defineerida paketi MATLAB funktsioone (inline, m-file);
- kasutada paketi MATLAB sisseehitatud vahendeid (quad);
- illustreerida numbrilise integreerimise teematikat paketi MATLAB abil;
- realiseerida paketi MATLAB programmiselt tuntud kvadratuurvalemeid;
- realiseerida paketi MATLAB programmiselt Runge veahindamismeetodit.

## Integraali vea ligikaudne hindamine

### Runge võtte

Arvutame määratud integraali

$$J = J(f) = \int_a^b p(x) f(x) dx$$

kvadratuurvalemi abil, näiteks trapets- või Simpsoni valemiga, osalõikude arvuga  $n$ , sammuga  $h$ . Püstitame küsimuse: kui suur on lähisväärtuse täpsus?

Tähistame vastava integraali lähisväärtuse  $J_n$ . Avaldugu kvadratuurvalemi jääkliige kujul

$$J - J_h = \kappa h^q + O(h^{q+1}).$$

Näiteks trapetsvalemi korral:

$$q = 2 \text{ ja } \kappa = -[f''(b) - f''(a)]/12 ;$$

Simpsoni valemi korral aga:

$$q = 4 \text{ ja } \kappa = -[f^{(4)}(b) - f^{(4)}(a)]/180 .$$

Seejärel kahekordistame sammupikkust  $h$  ja rakendame uuesti kvadratuurvalemit. Vastav integraali lähisväärtus olgu  $J_{2h}$ .

Integraali arvutamisel tehtud viga on nüüd vastavalt jääkliikme hinnangule:

$$J - J_{2h} = \kappa(2h)^q + O((2h)^{q+1}).$$

Kahe valemi põhjal saame hinnata integraali  $J$  ligikaudset väärtust selliselt:

$$J = J_h + (J_h - J_{2h})(2^q - 1) + O(h^{q+1}).$$

Kui täpsuses meid ei rahulda, siis kordame protseduuri, jagame esialgse sammu kaheks. Nii jätkame kuni soovitud täpsuse saavutamiseni.

Niisugune lihtne veahinnang töötab praktilistes arvutustes enamasti hästi.