

Mathematica kasutamine

1. Sissejuhatus

Programmi *Mathematica* avanemisel pole programmi tuum - *Kernel* - vaikumisi käivitatud. *Kernel* on programmi see osa, mis tegelikult teostab arvutusi. *Mathematica* abil võib mitte ainult arvutada, vaid ka kujundada dokumente. Peale selle, programmi tuum võib töötada hoopis teises arvutis, mis asub samas arvutivõrgus. Seda kõike on võimalik konfigureerida, vaikumisi käivitatakse tuum siiski samas arvutis esimese arvutuse käivitamisel.

Arvutuse teostamiseks tuleb pärast arvutuse sisestamist vajutada *Shift+Enter*. Näiteks:

2+4 Shift+Enter.

Ekraanile ilmub:

In[1]:= 2 + 4

Out[1]= 6

In ja Out näitavad, mitmenda arvutusega on tegemist, antud juhul on esimene sisestus $2 + 4$ ja esimene väljund 6. Edasistes arvutustes saab viidata tagasi eelmistele sisestustele ja väljunditele ning neid arvutustes kasutada. Seda vaatame hiljem. Arvutustes saab kasutada nii tekstipõhist arvutust - kõik tehted trükitakse sisse käsitsi, ei mingit hiirega edasi-tagasi sõelumist. (kes on proovinud hiire abil Wordis valemeid sisestada, teab, kui aega nõudev see on). Mathematica 4 ja 5 versioonides saab kasutada ka graafilist kasutajaliidest. Shift-Enteri asemel võib vajutada ka Enter-klahvi, mis asetseb klaviatuuri paremas servas.

Vihje: tihedalt valemeid täis tekstis kasutatakse TeX-i - tulemus on ilusam, valemite sisestamine

võtab tunduvalt vähem aega. Teisalt - TeX-i õppimine võtab natuke aega, kuid kes teadusega tahab tegelda, selle jaoks on see enam-vähem kohustuslik. Matemaatika ja füüsika ajakirjad võtavad vastu üldjuhul vaid TeX-s või LaTeX-s esitatud tekste), kuigi viimastel aastatel siiski ka

näiteks MS Wordis kirjutatut.

2. Aritmeetika, algebra.

2.1. Aritmeetika. Täpsed ja ligikaudsed tehted.

Tavalistes arvutustes kasutatavad tehted:

$x + y + z$	<i>liitmine</i>
$-x$	<i>lahutamine, miinus, neg. arv</i>
$x \ y \ z$ või $x * y * z$	<i>korrutamine, korrutusmärk või tühik selle asemel</i>
x / y	<i>jagamine</i>
$x \wedge y$ ehk x^y	<i>astendamine</i>

ASCII koodis on astendamismärk 94, selle saab kätte kui vajutada Alt+94 (94 klaviatuuri paremalt poolelt).

Näited:

In[2]:= **2 . 3 + 5 . 6 * 2**

Out[2]= 13 . 5

In[3]:= **2 4 6**

Out[3]= 48

In[4]:= **2 . 4³²**

Out[4]= 1 . 46811 × 10¹²

Viimane avaldis on antud juba nn teaduslikus tähistuses (*scientific notation*). Samamoodi saab arve ka sisestada. Kolm järgmist kirjutist on samaväärsed:

In[5]:= **3 . 5 10³⁴**

Out[5]= 3 . 5 × 10³⁴

In[6]:= **3 . 5 10 ^ 34**

Out[6]= 3 . 5 × 10³⁴

In[7]:= **3 . 5 * ^ 34**

Out[7]= 3 . 5 × 10³⁴

Kuigi Mathematica võib leida väga pikki arvutustulemusi täpselt, on tavaliselt soovitatav need esitada siiski nõ ümmardatud kujul, kümne astmete kaudu. Kümne astmete kaudu (ligikaudsel kujul) saab arve esitada kasutades funktsiooni `N[ ]`:

`In[8]:= 332`

`Out[8]= 1853020188851841`

`In[9]:= 332 // N`

`Out[9]= 1.85302 × 1015`

`In[10]:= N[332]`

`Out[10]= 1.85302 × 1015`

`In[11]:= N[240]`

`Out[11]= 1.09951 × 1012`

Mathematica eristab täisarve ning reaalarve. Kui sisestate arvutuses kõik arvud täisarvudena, siis antakse ka lõpptulemus täisarvude kaudu. Näiteks 34 on antud programmi jaoks täisarv, 34.0 ehk ka 34. reaalarv.

`In[12]:= 1 / 2 + 4 / 7`

`Out[12]=  $\frac{15}{14}$`

`In[13]:= 452 / 62`

`Out[13]=  $\frac{226}{31}$`

`In[14]:= 452.0 / 62`

`Out[14]= 7.29032`

`In[15]:= 1.0 + 452 / 62`

`Out[15]= 8.29032`

Seega, kui arvutuses on antud vähemalt üks reaalarv, antakse vastus reaalarvuna.

Mõned matemaatilised funktsioonid.

<code>Sqrt[x]</code>	ehk $\sqrt{x}$	ruutjuur x-st
<code>Sqrt[x,n]</code>	ehk $\sqrt[n]{x}$	n-s juur x-st
<code>Log[x]</code>		naturaallogaritm - $\ln x$
<code>Log[a,x]</code>		logaritm alusel a - $\log_a x$
<code>Sin[x], Cos[x], Tan[x], Cot[x]</code>		trigoneemetrilised funktsioonid, argument on radiaanides!
<code>ArcSin[x], ArcCos[x], ArcTan[x], ArcCot[x]</code>		trigoneemetrilised pöördfunktsioonid (arkusfunktsioonid), annavad vastuse radiaanides
<code>n!</code>		faktoriaal
<code>Abs[x]</code>		absoluutväärtus
<code>Round[x]</code>		ümmandamine
<code>Mod[n,m]</code>		jääk, mis saadakse jagamisel $\frac{n}{m}$
<code>Quotient[n,m]</code>		jagatise $\frac{n}{m}$ täisosa
<code>Random[]</code>		pseudojuhuslik arv 0 ja 1 vahel
<code>Max[x,y,...], Min[x,y,...]</code>		arvude (avaldiste) x,y,... maksimum, miinimum
<code>FactorInteger[n]</code>		täisarvu n lahutamine algarvude korrutiseks
<code>avaldis // N</code>	ehk <code>N[avaldis]</code>	avaldise ligikaudne väärtus; esitamine kümne astmete kaudu
<code>N[avaldis,n]</code>		avaldise väärtus n numbrikoha (mitte komakoha!) täpsusega

NB! Mathematica's on kõigi funktsioonide argumendid antud alati kandilistes sulgudes. Funktsioonide ja protseduuride nimed algavad alati suure tähega. Seetõttu on mõistlik enda defineeritud suurused ja funktsioonid anda väikeste algustähtedega.

Konstandid.

<code>Pi</code>	või π	$\pi \approx 3,14159$
<code>E</code>	või e	$e \approx 2,71828$
<code>Degree</code>	või $^\circ$	nurgakraadi tähis
<code>I</code>	või i	imaginaarühik, $i = \sqrt{-1}$
<code>Infinity</code>	või ∞	lõpmatus

π ja e võetakse arvuti poolt vaikimisi samuti kui täpsed arvud. Seetõttu ei anna arvuti selliste avaldiste ligikaudseid väärtusi. Mõned näited:

`In[16]:=  $\pi / 4$`

`Out[16]=  $\frac{\pi}{4}$`

`In[17]:= Pi / 4 .`

`Out[17]= 0.785398`

`In[18]:= Sin[20 °] // N`

`Out[18]= 0.34202`

`In[19]:= e20 .`

`Out[19]= 4.85165 × 108`

Ilma ligikaudset vastust nõudva funktsioonita `N` antakse vatuses lihtsalt **Sin[20°]**, mis ongi täpne vastus.

Kompleksarvud.

<code>x + I y</code>	kompleksarv kujul $x + i y$
<code>Re[z]</code>	kompleksarvu reaalosa $\text{Re } z$
<code>Im[z]</code>	kompleksarvu imaginaarosa $\text{Im } z$
<code>Abs[z]</code>	absoluutväärtus $ z $
<code>Arg[z]</code>	kompleksarvu $ z e^{i\varphi}$ argument φ
<code>Conjugate[z]</code>	arvu z kaaskompleks $\bar{z}$ ehk z^*

Näited:

`In[20]:= (3 + 2 I) (7 - I)`

`Out[20]= 23 + 11 i`

`In[21]:= (3 + 2 I) / (7 - I) // N`

`Out[21]= 0.38 + 0.34 i`

`In[22]:= Exp[2.0 - 7 I]`

`Out[22]= 5.57063 - 4.85451 i`

2.2. Algebralised teisendused

Mathematica võimaldab ka tehteid suure arvu muutujate ja arvudega, sisaldab erinevaid teisendusreegleid, võimaldab taandada ja murdudes koondada, tuua mingeid liikmeid sulgude ette jne. Üht omapära tuleb küll enne mainida.

Ülalpool on öeldud, et korrutusmärgina võib kasutada ka tühikut. Kuid näiteks juhul $2(x-3)$ pole tühikut vaja. Samuti $2x$ on sama, mis $2*x$; kuid x^2 on uus muutuja, mida võiks tähistada ka x_2 .

Seega $-$ arvu korrutamisel muutujaga, kui arv asub eespool, pole tühikut või korrutusmärgi vaja.

Kuid muutuja korrutamisel arvuga (muutuja on enne arvu), peab see olema olema $-x^2$. Ka xy tähendab lihtsalt üht uut muutujat, mitte x ja y korrutist.

<code>Expand[avaldis]</code>	korrutab lahti kõik korrutised ja astmed kirjutades saadud tulemuse summana
<code>ExpandAll[avaldis]</code>	rakendab <code>Expand</code> 'i kõikjal (võtab kauem aega)
<code>PowerExpand[avaldis]</code>	kirjutab lahti ka astmete korrutised ja jagatised (näiteks ruutjuur ruudust)
<code>Factor[avaldis]</code>	kirjutab <i>avaldis</i> e vähima arvu tegurite korrutisena
<code>Simplify[avaldis]</code>	üritab leida <i>avaldis</i> e lihtsaimat kuju kasutades erinevaid algebralisi teisendusi
<code>FullSimplify[avaldis]</code>	üritab leida <i>avaldis</i> e lihtsaimat kuju kasutades erinevaid algebralisi teisendusi, võimaldab rohkem aega kulutada,
<code>Together[avaldis]</code>	resultaat võib väga pikkade avaldiste korral olla lühem viib murdude summa kõik liikmed ühisele nimetajale
<code>Apart[avaldis]</code>	kirjutab murru lahti, tulemuseks on murrud, millede nimetajad on võimalikult lihtsamal kujul
<code>Cancel[avaldis]</code>	taandab murru nimetaja ja lugeja ühised tegurid
<code>Collect[avaldis,x]</code>	rühmitab <i>avaldis</i> e liidetavad x astmete järgi
<code>FactorTerms[avaldis,x]</code>	toob sulgude ette liikmed, mis ei sõltu x -st

`In[23]:= $Line = 0;`

Näited:

$$\text{In}[1]:= \mathbf{f} = \frac{(\mathbf{x} - 1)^2 (2 + \mathbf{x})}{(1 + \mathbf{x}) (\mathbf{x} - 3)^2}$$

$$\text{Out}[1]= \frac{(-1 + \mathbf{x})^2 (2 + \mathbf{x})}{(-3 + \mathbf{x})^2 (1 + \mathbf{x})}$$

$$\text{In}[2]:= \mathbf{Expand}[\mathbf{f}]$$

$$\text{Out}[2]= \frac{2}{(-3 + \mathbf{x})^2 (1 + \mathbf{x})} - \frac{3 \mathbf{x}}{(-3 + \mathbf{x})^2 (1 + \mathbf{x})} + \frac{\mathbf{x}^3}{(-3 + \mathbf{x})^2 (1 + \mathbf{x})}$$

$$\text{In}[3]:= \mathbf{s} = \mathbf{ExpandAll}[\mathbf{f}]$$

$$\text{Out}[3]= \frac{2}{9 + 3 \mathbf{x} - 5 \mathbf{x}^2 + \mathbf{x}^3} - \frac{3 \mathbf{x}}{9 + 3 \mathbf{x} - 5 \mathbf{x}^2 + \mathbf{x}^3} + \frac{\mathbf{x}^3}{9 + 3 \mathbf{x} - 5 \mathbf{x}^2 + \mathbf{x}^3}$$

$$\text{In}[4]:= \mathbf{Together}[\mathbf{s}]$$

$$\text{Out}[4]= \frac{2 - 3 \mathbf{x} + \mathbf{x}^3}{9 + 3 \mathbf{x} - 5 \mathbf{x}^2 + \mathbf{x}^3}$$

$$\text{In}[5]:= \mathbf{Apart}[\%]$$

$$\text{Out}[5]= 1 + \frac{5}{(-3 + \mathbf{x})^2} + \frac{19}{4 (-3 + \mathbf{x})} + \frac{1}{4 (1 + \mathbf{x})}$$

$$\text{In}[6]:= \mathbf{Factor}[\%]$$

$$\text{Out}[6]= \frac{(-1 + \mathbf{x})^2 (2 + \mathbf{x})}{(-3 + \mathbf{x})^2 (1 + \mathbf{x})}$$

$$\text{In}[7]:= \mathbf{Simplify}[\%]$$

$$\text{Out}[7]= \frac{(-1 + \mathbf{x})^2 (2 + \mathbf{x})}{(-3 + \mathbf{x})^2 (1 + \mathbf{x})}$$

St, Factor käsuga leitud avaldis ongi juba lihtsaimal kujul

$$\text{In}[8]:= \mathbf{v} = (3 + 2 \mathbf{x})^2 (\mathbf{x} + 2 \mathbf{y})^2$$

$$\text{Out}[8]= (3 + 2 \mathbf{x})^2 (\mathbf{x} + 2 \mathbf{y})^2$$

$$\text{In}[9]:= \mathbf{Expand}[\mathbf{v}]$$

$$\text{Out}[9]= 9 \mathbf{x}^2 + 12 \mathbf{x}^3 + 4 \mathbf{x}^4 + 36 \mathbf{x} \mathbf{y} + 48 \mathbf{x}^2 \mathbf{y} + 16 \mathbf{x}^3 \mathbf{y} + 36 \mathbf{y}^2 + 48 \mathbf{x} \mathbf{y}^2 + 16 \mathbf{x}^2 \mathbf{y}^2$$

In[10]:= **Collect**[v, x]

Out[10]= $4x^4 + 36y^2 + x^3(12 + 16y) + x^2(9 + 48y + 16y^2) + x(36y + 48y^2)$

In[11]:= **FactorTerms**[v, y]

Out[11]= $(9 + 12x + 4x^2)(x^2 + 4xy + 4y^2)$

TrigExpand[avaldis]	korrutab trigonomeetrilised avaldised summana lahti
TrigFactor[avaldis]	kirjutab trigonomeetrilise avaldise lahti tegurite korrutisena
TrigReduce[avaldis]	teisendab, koondab ja taandab avaldise kasutades nurkade trigonomeetriliste funktsioonide vahelisi seoseid
TrigToExp[avaldis]	teisendab trigonomeetrilised funktsioonid eksponentideks
ExpToTrig[avaldis]	teisendab eksponendid trigonomeetrilisteks avaldisteks
FunctionExpand[avaldis]	korrutab (eraldab välja) spetsiaalfunktsioonid
ComplexExpand[avaldis]	korrutab kompleksarvudega avaldised lahti

2.3. Muutujatele, avaldistele väärtuse omistamine, tehted eelnevate arvutustulemustega

$x = \text{väärtus}$	x-le antakse mingi väärtus, see võib olla arvuline väärtus, algebraline avaldis või mingi pikem funktsioon, arvutus, võrrand jne
$x = .$ või Clear [x]	x-i väärtus tühistatakse
$x := \text{väärtus}$	x-le antakse mingi väärtus, sarnane esimese omistamisega
$\text{avaldis} /. x \rightarrow \text{väärtus}$	annab avaldises x-le väärtuse
$\text{avaldis} /. \{x \rightarrow x_{\text{väär}}, y \rightarrow y_{\text{väär}}\}$	avaldises antakse väärtused nii x-le kui ka y-le

Defineerime avaldise s järgmiselt:

In[12]:= **s** = $x^2 + 4.5x - y$

Out[12]= $4.5x + x^2 - y$

Järgnevalt näitame mitmeid erinevaid võimalusi avaldises s sisalduvatele muutujatele väärtuste omistamiseks.

In[13]:= **s** /. **x** → **4**

Out[13]= $34. - y$

In[14]:= **s** /. {**x** → 2.3, **y** → 3}

Out[14]= 12.64

In[15]:= **s** /. {**x** → **y**, **y** → **t**}

Out[15]= $-t + 4.5 y + y^2$

In[16]:= **s** /. {**x** → **x**², **y** → **y**³}

Out[16]= $4.5 x^2 + x^4 - y^3$

Viimane on nn rekursiivne väärtustamine *Mathematica*'s ja peaks väga tuttav olema neile, kes on programmeerimisega kokku puutunud. /. abil väärtuste omistamine ja arvutus toimib järgmiselt: esmalt teostatakse arvutus avaldises olevate muutujatega, seejärel omistatakse muutujatele uus väärtus ning antakse lõplik arvutustulemus. Seda demonstreerib ka järgmine näide.

In[17]:= **s**² + 3 /. **x** → 2

Out[17]= $3 + (13. - y)^2$

In[18]:= **Expand**[%]

Out[18]= $172. - 26. y + y^2$

Viimases arvutuses rakendati protseduuri Expand eelmisele (eelviimasele).

Teine võimalus avaldises või arvutuses olevatele muutujatele väärtuse omistamine on toodud järgnevas näites.

In[19]:= **a** = 3. ;

b = 2.5 ;

c = **x** - **y** ;

$a^2 + b^3 - \frac{c}{2.}$

Out[22]= $24.625 - 0.5 (x - y)$

Viimases näites oleme enne arvutust omistanud väärtused muutujatele a, b, c. Semikoolon rea lõpus eristab erinevaid sisestusi; semikooloniga lõppeva rea arvutuse tulemust ei kuvata ekraanile (ka lihtne väärtuse omistamine a=3. on siin mõistetud arvutusena), samas on iga sellise arvutuse tulemus ikkagi hiljem kasutatav. Semikoolonit on vahel mõttekas kasutada ka siis, kui on tegemist pikkade arvutustega ning mingi vahetulemus on väga pikk, sel juhul võib kasutada pidevalt viimast arvutust (vahepeal lihtsustades ning taandades/koondades erinevaid liikmeid) iga kord semikooloniga sisestuse lõpus ning alles viimase sisestus on ilma semikoolonita. Semikooloni kasutamine on mõistlik ka

arvutimälu säästmise seisukohalt.

Veel üks omapära on toodud järgmises näites:

```
In[23]:= 23 x + 14 y - 23 x * y
          2 y^2 - 11 x
```

```
Out[23]= 23 x + 14 y - 23 x y
```

```
Out[24]= -11 x + 2 y^2
```

Esimeses sisestuses on teisele reale mindud üle reavahetuse klahvi abil. Nagu näha on arvuti eeldanud, et tegemist on kahe erineva sisestusega, mitte ühe ja sama avaldisega. Juhul, kui avaldis ei mahu ühte ritta ära ja peaks ilmtingimata just korrutusmärgi koha pealt avaldist tüheldama, peaks esimese rea lõppu kirjutama korrutusmärgi:

```
In[25]:= 23 x + 14 y - 23 x * y *
          2 y^2 - 11 x
```

```
Out[25]= 12 x + 14 y - 46 x y^3
```

Mathematica vanemad versioonid eeldavad vaikimisi siiski, et tegemist oli korrutamisega.

Nagu mainitud, on antud programmis mitmeid võimalusi eelmistele väljunditele.

%	eelmise arvutuse väljund (resultaat)
%%	üle-eelmise arvutuse väljund
%n	n-nda arvutuse väljund
Out[n]	n-nda arvutuse väljund

Veel mõned näited. Pidage silmas, et kui teie sisestate avaldised täpselt nii nagu alltoodud näites, võite saada täiesti erineva tulemuse, kui teie arvutuste numeratsioon erineb antud näidete omast:

```
In[26]:= %14^2
```

```
Out[26]= 159.77
```

```
In[27]:= Expand[%14 - y^2]
```

```
Out[27]= 12.64 - y^2
```

```
In[28]:= Out[16] - Out[17]
```

```
Out[28]= -3 + 4.5 x^2 + x^4 - (13. - y)^2 - y^3
```

In[29]:= **Expand**[%]

Out[29]= $-172. + 4.5 x^2 + x^4 + 26. y - y^2 - y^3$

In[30]:= **Simplify**[%]

Out[30]= $4.5 x^2 + x^4 - 1. (7.51074 + y) (22.9005 - 6.51074 y + y^2)$

2.4. Avaldiste lühendamine, avaldise osade eraldamine, avaldise lihtsustamine eeldustel

Esmalt vaatame mõningaid lihtsustamisi lisaeeldustel. Näiteks

In[31]:= **\$Line = 0;**
Clear[s, a, b, c];

In[2]:= **Simplify**[$\sqrt{x^2}$]

Out[2]= $\sqrt{x^2}$

Selline on vastus tänu sellele, et $\sqrt{x^2} = x$ vaid siis, kui $x > 0$. Kui eeldame, et see tingimus on täidetud, peaksime kirjutama:

In[3]:= **Simplify**[$\sqrt{x^2}$, $x > 0$]

Out[3]= x

St lihtsustamine toimub eeldusel, mis on avaldisest eraldatud komaga.

Veel üks näide sellisest eeldusest on juhul, kui eeldame, et mingi suurus kuulub kindlasse hulka.

Eeldame, et $n \in \mathbb{N}$, st n on täisarv.

In[4]:= **Simplify**[**Sin**[$x + 2 \pi n$], **Element**[n , **Integers**]]

Out[4]= **Sin**[x]

Simplify [<i>avaldis</i> , <i>eeldus</i>]	lihtsustab avaldist antud eeldusel
Simplify [<i>avaldis</i> , { <i>eeldus</i> ₁ , <i>eeldus</i> ₂ }]	lihtsustab avaldist antud eeldustel (mis ei tohi olla
	üksteisega vastuolus
Element [<i>x</i> , <i>hulk</i>]	defineerib <i>x</i> kui <i>hulga</i> liikme
Element [{ <i>x</i> ₁ , <i>x</i> ₂ , <i>x</i> ₃ ...}, <i>hulk</i>]	defineerib kõik <i>x_i</i> <i>hulga</i> liikmetena
Reals	reaalarvud

Integers	täisarvud
Complexes	kompleksarvud
Primes	algarvud

Hulga võib ka ise defineerida ja siis lihtsustada tingimusel, et mingi suurus kuulub defineeritud hulka

Algebraalsetest avaldistest võib kätte saada ka vaid nende osasid, vastavad käsud on järgmised:

Coefficient[avaldis, vorm]	annab avaldise vorm'i kordaja
Exponent[avaldis, vorm]	annab vormi maksimaalse astme
Part[avaldis] või avaldis[[n]]	annab avaldise n-nda liikme
Numerator[avaldis]	annab murdavaldise lugeja
Denominator[avaldis]	annab murdavaldise nimetaja

3. Sümbolarvutus. Tuletised, integraalid. Funktsioonid.

3.1. Tuletis, diferentsiaal, integraal, piirväärtus, summa.

```
In[5]:= Clear[f]; $Line = 0;
```

D[f,x] või $\partial_x f$	(osa)tuletis funktsioonist f x järgi: $\frac{df}{dx}$ ehk $\frac{\partial f}{\partial x}$
D[f,x,y,...] või $\partial_{x,y,z,...} f$	mitmekordne (osa)tuletis funktsioonist või avaldisest f:
$\frac{\partial^3 f}{\partial x \partial y \partial z}$	
D[f,{x,n}]	n-kordne tuletis avaldisest f x järgi: $\frac{d^n f}{dx^n}$
Limit[f, x -> x ₀]	piirväärtus $\lim_{x \rightarrow x_0} f$
Dt[f]	täisdiferentsiaal
Dt[f,x]	täistuletisavaldisest f muutuja x järgi
$\int f dx$ või Integrate[f,x]	määramata integraal
$\int_a^b f(x) dx$ või Integrate[f,{x,a,b}]	määratud integraal
$\int \int f(x, y) dx dy$ või Integrate[f,x,y]	mitmekordne määramata integraal muutujate x ja y järgi
$\int_a^b \int_c^d f(x, y) dx dy$ või	mitmekordne määratud integraal
Integrate[f,{x,a,b},{y,c,d}]	
Series[f, {x, x ₀ , järk}]	Taylori astmerida funktsioonist (avaldisest) f punkti
x = x ₀ juures	
Normal[rida]	juhul, kui rida on Taylori astemrida, siis see protseduur
annab	
	tulemuseks Taylori rea ilma viimase liikmeta

$\sum_{i=i_{\min}}^{i_{\max}} f$ või $\text{Sum}[f, \{i, i_{\min}, i_{\max}\}]$ summa
 $\text{Sum}[f, \{i, i_{\min}, i_{\max}\}, \{j, j_{\min}, j_{\max}\}]$ mitmekordne summa
 $\prod_{i=i_{\min}}^{i_{\max}} f$ ehk $\text{Product}[f, \{i, i_{\min}, i_{\max}\}]$ korrutis

Tuletised, diferentsiaalid

$$\text{In}[1]:= \partial_{\mathbf{x}} \mathbf{x}^3$$

$$\text{Out}[1]= 3 \mathbf{x}^2$$

$$\text{In}[2]:= \mathbf{D}[\mathbf{x}^n, \mathbf{x}]$$

$$\text{Out}[2]= n \mathbf{x}^{-1+n}$$

$$\text{In}[3]:= \mathbf{D}[\mathbf{x}^n, n]$$

$$\text{Out}[3]= \mathbf{x}^n \text{Log}[\mathbf{x}]$$

$$\text{In}[4]:= \partial_{\mathbf{x}} \mathbf{x}^2 \text{Log}[\mathbf{x} + \mathbf{a}]$$

$$\text{Out}[4]= 2 \mathbf{x} \text{Log}[\mathbf{a} + \mathbf{x}]$$

Esimene avaldis on tavaline tuletis. Teine näitab, et kui tuletist võetakse x järgi, on n lihtsalt konstant (tegemist on osatuletisega x järgi). Kolmandas arvutuses oleme leidnud osatuletise muutuja n järgi, Ka viimases arvutuses leitakse osatuletis x järgi võttes a konstandiks.

$$\text{In}[5]:= \partial_{\mathbf{x}, \mathbf{y}} (3 \mathbf{x}^2 \mathbf{y}^5)$$

$$\text{Out}[5]= 30 \mathbf{x} \mathbf{y}^4$$

$$\text{In}[6]:= \partial_{\mathbf{x}, \mathbf{y}} 3 \mathbf{x}^2 \mathbf{y}^5$$

$$\text{Out}[6]= 0$$

Eelviimane avaldis näitab, kuidas käib mitmekordse osatuletise leidmine. Sellise kirjutusviisi korral peab avaldis paiknema kindlasti ümarsulgudes. Vastasel juhul leitakse tuletis vaid avaldise esimesest tegurist, mida näitab viimane näide, kus tuletis on leitud arvust 3 (mis on loomulikult 0). Samast avaldisest võime leida ka kahekordse tuletise muutuja x järgi:

$$\text{In}[7]:= \partial_{\mathbf{x}, \mathbf{x}} (3 \mathbf{x}^2 \mathbf{y}^5)$$

$$\text{Out}[7]= 6 \mathbf{y}^5$$

In[8]:= **D**[**3 x² y⁵**, {**x**, 2}]

Out[8]= $6 y^5$

Tuletise võime leida ka tundmatul kujul antud funktsioonist, kus on näidatud, et see funktsioon sõltub muutujast x:

In[9]:= **∂_x** (**3 f[x]**)

Out[9]= $3 f'[x]$

In[10]:= **D**[**Sin[x] f[3 x]**, **x**]

Out[10]= $\cos[x] f[3x] + 3 \sin[x] f'[3x]$

Niinimetatud täistuletise Dt[f,x] korral eeldatakse, et kõik muutujad (st mitte-arvud) sõltuvad muutujast x.

In[11]:= **Dt**[**n Sin[x]**, **x**]

Out[11]= $n \cos[x] + \text{Dt}[n, x] \sin[x]$

Seega, viimases avaldises tähendab Dt[n,x] tuletist n-st muutuja x järgi.

Integreerimine.

In[12]:= $\int \frac{3 y}{x^2 + y^2} dx$

Out[12]= $3 \text{ArcTan}\left[\frac{x}{y}\right]$

Antud integraali leidmisel arvestati loomulikult, et y ei sõltu x-st, y on konstant. Koolimatemaatikast on meil teada, et määramata integraali leidmisel liidetakse alati otsa konstant. Antud programm jätab selle kasutajate hooleks.

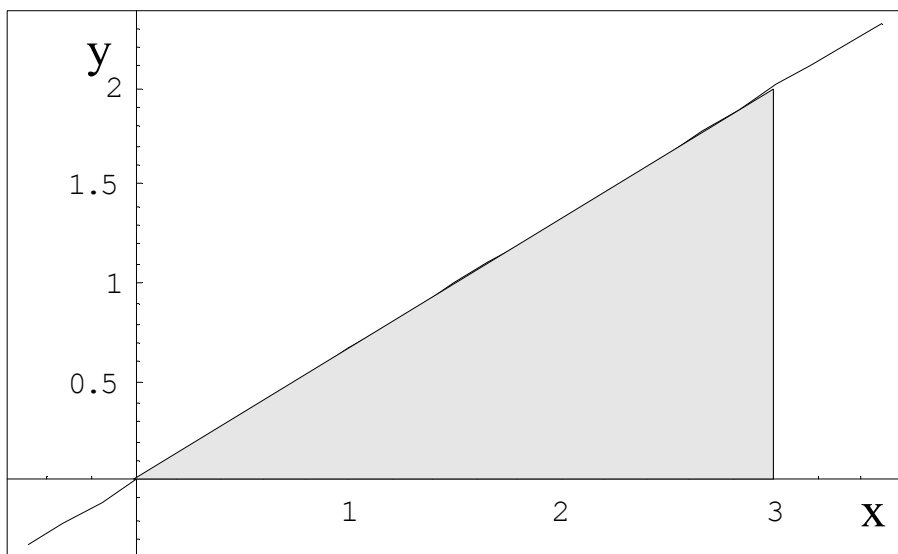
In[13]:= $\int_0^5 2 x^3 dx$

Out[13]= $\frac{625}{2}$

In[14]:= **Integrate**[$\frac{2 y}{x^2 + y^2}$, **x**, **y**]

Out[14]= $-2x + 2y \text{ArcTan}\left[\frac{x}{y}\right] + x \text{Log}[x^2 + y^2]$

Järgmisena vaatame kahekordset määratud integraali, kus integreerimispiirkond on näidatud allpool toodud joonisel. See tähendab, et võime integreerida funktsiooni $f(x,y)$ esmalt y järgi 0-st kuni $\frac{2x}{3}$ -ni ja siis x järgi 0-st 3-ni:



$$\text{In}[15]:= \int_0^3 \int_0^{\frac{2x}{3}} (x^2 + y^2) \, dy \, dx$$

$$\text{Out}[15]= \frac{31}{2}$$

Sama integraal teiste tähistustega:

$$\text{In}[16]:= \text{Integrate}[x^2 + y^2, \{x, 0, 3\}, \{y, 0, \frac{2x}{3}\}]$$

$$\text{Out}[16]= \frac{31}{2}$$

Viimasest arvutusest on näha, et sellisel juhul integreeritakse esimesena viimasena antud muutuja järgi erinevalt tavapärasest esitusviisist, kus esmalt leitakse sisemine integraal.

Piirväärtused. Taylori rida.

Taylori rida on astmerida kujul:

$$f(x) = f(x_0) + f'(x)|_{x=x_0} (x - x_0) + \frac{1}{2} f''(x)|_{x=x_0} (x - x_0)^2 + \dots \\ + \frac{1}{n!} f^{(n)}(x)|_{x=x_0} (x - x_0)^n + \dots$$

Järgmises näites arendame Taylori ritta funktsiooni $f(x) = e^x(1+x)$ punkti $x_0 = 0$ juures.

In[17]:= **Series**[$e^x (1 + x)$, {**x**, 0, 4}]

Out[17]= $1 + 2x + \frac{3x^2}{2} + \frac{2x^3}{3} + \frac{5x^4}{24} + O[x]^5$

Viimane liige - $O[x]^5$ võtab kokku kõik liikmed, milles x aste on 5 või sellest suurem. Et sellest vabaneda, tuleks kasutada käsku **Normal**:

In[18]:= **Normal**[%]

Out[18]= $1 + 2x + \frac{3x^2}{2} + \frac{2x^3}{3} + \frac{5x^4}{24}$

Järgmised paar näidet on piirväärtuste kohta. Esmalt defineerime avaldise, siis leiame piirväärtuse ning vaatama ka seda, kuidas *Mathematica* otsese arvutusega üritab avaldise väärtust leida.

In[19]:= **s** = $\frac{\text{Sin}[x]}{x}$

Out[19]= $\frac{\text{Sin}[x]}{x}$

In[20]:= **Limit**[**s**, **x** → 0]

Out[20]= 1

In[21]:= **s** /. **x** → 0

- Power::infy : Infinite expression $\frac{1}{0}$ encountered.

- ∞::indet : Indeterminate expression 0 ComplexInfinity encountered.

Out[21]= Indeterminate

Seega, avaldisel leidus lõplik piirväärtus - 1, kuid otsesel arvutusel oleks tulnud jagada nulliga ning tulemus jäänud leidmata.

In[22]:= **t** = $\frac{(9 - x^2) 2x^3}{(3x - 9)}$

Out[22]= $\frac{2x^3 (9 - x^2)}{-9 + 3x}$

In[23]:= **Limit**[**t**, **x** → 3]

Out[23]= -108

In[24]:= **t /. x -> 3**

– *Power::infy : Infinite expression $\frac{1}{0}$ encountered.*

– *$\infty::indet$: Indeterminate expression 0 ComplexInfinity encountered.*

Out[24]= Indeterminate

Veel mõned näited:

In[25]:= **Limit[e^{-x} - 1, x -> ∞]**

Out[25]= -1

In[26]:= **Limit[Tan[x], x -> $\frac{\pi}{2}$]**

Out[26]= $-\infty$

Summad ja korrutised.

In[27]:=
$$\sum_{i=0}^5 \frac{2^i}{i!}$$

Out[27]= $\frac{109}{15}$

In[28]:= **N[%]**

Out[28]= 7.26667

In[29]:=
$$\sum_{i=1}^{\infty} \frac{1}{i^2}$$

Out[29]= $\frac{\pi^2}{6}$

In[30]:=
$$\sum_{i=1}^{\infty} \frac{1}{i! + (2i)!}$$

Out[30]=
$$\sum_{i=1}^{\infty} \frac{1}{i! + (2i)!}$$

In[31]:= **N[%]**

Out[31]= 0.373197

In[32]:=
$$\sum_{i=1}^3 \sum_{j=1}^i \mathbf{x}^i \mathbf{y}^j$$

Out[32]= $x y + x^2 y + x^3 y + x^2 y^2 + x^3 y^2 + x^3 y^3$

In[33]:= **Product[x + i, {i, 1, 5}]**

Out[33]= $(1 + x) (2 + x) (3 + x) (4 + x) (5 + x)$

3.2. Funktsioonide defineerimine.

In[34]:= **Clear[s, f];**
\$Line = 0;

Lisaks *Mathematica*'s defineeritud funktsioonidele ja protseduuridele võib neid ka ise teha. Defineerime järgmise ruutfunktsiooni:

In[1]:= **f[x_] := 3 x² + 2 x + 1**

Siin alakriips ('blank') tähendab, et see funktsioon ehk reegel antud funktsiooni leidmiseks kehtib ka siis, kui me edaspidi asendama muutuja x muutujaga y:

In[2]:= **f[y]**

Out[2]= $1 + 2 y + 3 y^2$

In[3]:= **f[2.5]**

Out[3]= 24.75

Võrdluseks mõned vastupidised näited:

In[4]:= **g[x] := 3 x² + 2 x + 1**

In[5]:= **g[y]**

Out[5]= g[y]

```
In[6]:= g[3]
```

```
Out[6]= g[3]
```

St, antud reegel ($g[x]$) ei ütle midagi selle kohta, kui x asemel on y . Me saame küll leida mõningate avaldiste väärtusi, kuid enamus võimalusi jääb kasutamata.

```
In[7]:= g[x] /. x -> y
```

```
Out[7]= 1 + 2 y + 3 y^2
```

```
In[8]:= g[x] /. x -> 3
```

```
Out[8]= 34
```

Seega, sellisel, ilma alakriipusta definitsioonil 'kui funktsioon x -st' puudub mõte. Selliselt võime määrata ka lihtsa avaldisena (vähem vaeva edasistes arvutustes). Esmalt tühistame $g[x]$ definitsiooni:

```
In[9]:= Clear[g]
```

```
In[10]:= g := 3 x^2 + 2 x + 1
```

```
In[11]:= g /. x -> 4
```

```
Out[11]= 57
```

Analoogiliselt võib defineerida ka mitme muutuja funktsioone. Allpool toome näitena füüsikast tuntud avaldise teepikkuse määramiseks mingil ajahetkel kui funktsiooni ajast t , algkiirusest v_0 ja kiirendusest a .

Märkus: alaindeksite kasutamine arvutustes ei ole alati väga hea idee!

```
In[12]:= Clear[s]
```

```
In[13]:= s[t_, v0_, a_] := v0 * t + 1/2 a * t^2
```

```
In[14]:= s[4, 0.5, 1]
```

```
Out[14]= 10.
```

```
In[15]:= s[4, 1, 0.5]
```

```
Out[15]= 8.
```

Funktsioone saab defineerida ka kui mingit funktsioonide jada - protseduuri:

```
In[16]:= avalid[n_] := Expand[ $\prod_{i=1}^n (x + i)$ ]
```

```
In[17]:= avalid[4]
```

```
Out[17]= 24 + 50 x + 35 x2 + 10 x3 + x4
```

Siin defineerisime funktsiooni $(x+1)(x+2)(x+3)\dots(x+n)$, mis korrutab selle avaldise ka lahti (vaikimisi *Mathematica* seda ei tee).

```
In[18]:= ava[x_, n_] := Expand[ $\prod_{i=1}^n (x + i^2)$ ]
```

```
In[19]:= ava[3, 4]
```

```
Out[19]= 6384
```

Mis juhtub, kui n asemele sisetada negatiivne arv või mitte-täisarv?

$f[x_] := x^3$	funktsiooni $f(x)$ defineerimine
$f[x_, y_] := x^3 + \text{Sin}[2 x + 3]$	mitme muutuja funktsiooni defineerimine
<code>Clear[f]</code>	funktsiooni f definitsiooni tühistamine
<code>?f</code>	funktsiooni f kuju näitamine (kehtib ka arvuti poolt defineeritud funktsioonide ja protseduuride kohta)
<code>??f</code>	kogu funktsiooni või protseduuri f kohta käiva info näitamine
$f[x] = x^3$	avaldis (ja mitte funktsiooni) f defineerimine

3.3. Võrrandid.

Alapunktis 2.3 näitasime, kuidas omistada väärtusi muutujatele. Näiteks $x = y$ on lihtsalt muutujale x muutuja y väärtuse omistamine. Selline omistamine deklareerib mõlemad suurused võrdseks, kuid ühe täpsustusega. Kui anname y -le mingi väärtuse, saab ka muutuja x automaatselt sama väärtuse, kuid muutujale x väärtuse omistamisega, jääb y väärtus muutumatuks ning sellest hetkest alates ei olen enam x y -ga võrdne. $x == y$ aga kontrollib, kas x on y -ga võrdne. Kahte võrdusmärki kasutatakse ka võrrandite lahendamisel (võrrandite defineerimisel samuti).

```
In[20]:= x = y
```

```
Out[20]= y
```

```
In[21]:= x == y
```

```
Out[21]= True
```

```
In[22]:= 2 + 4 == 6
```

```
Out[22]= True
```

```
In[23]:= 2 * 5 == 11
```

```
Out[23]= False
```

```
In[24]:= x = .
```

Viimase võrdusega tühistasime x väärtuse.

```
In[25]:= x == 5
```

```
Out[25]= x == 5
```

Kuivõrd x -l väärtus puudub, ei oska ka programm öelda, kas võrdus on tõene või väär.

Avaldis $x==5$ on tegelikult juba võrrand (mille lahendiks on ilmselt $x=5$), sest x väärtus on defineerimata.

Programmis *Mathematica* (ja ka näiteks Maple's) on tihti tavaks anda mingile avaldisele väärtuseks võrrand ise.

```
In[26]:= eq = x2 - 4 x - 7 == 0
```

```
Out[26]= - 7 - 4 x + x2 == 0
```

$x = y$	omistab x -le väärtuse y
$x == y$	kontrollib, kas x ja y on võrdsed
$x != y$	kontrollib, kas x ei võrdu y -ga (kui x ja y ei ole võrdsed, on vastus True)
$x > y$	kontrollib, kas x on suurem, kui y
$x < y$	kontrollib, kas x on väiksem kui y
$x <= y$	väiksem või võrdne
$x >= y$	suurem või võrdne
$x == y == z$	kas kõik on omavahel võrdsed
$x != y != z$	kas kõik on omavahel mittevõrdsed

Ülaltoodud avaldisi saab eriti hästi kasutada keerulisemata ülesannet lahendamisel, näiteks programmeerimisel *Mathematicas*, kasutades laused While-, Do, For, If.

Võrrandite lahendamine.

<code>Solve[vasak pool == parem pool, x]</code>	võrrandi lahendamine x suhtes, tulemuseks on reeglite hulk (list)
$x /. \text{lahend}$	reeglite listi kasutatakse x väärtuste saamiseks
avaldis /. lahend	reeglite listi kasutatakse avaldisele väärtuse andmiseks

Solve abil saab lahendada algebralisi võrrandeid, milles muutuja aste on maksimaalselt 4. Kui kõrgeim aste on üle nelja, ei pruugi täpset lahendit antud protseduur leida.

`In[27]:= Solve[eq, x]`

`Out[27]= {{x -> 2 - Sqrt[11]}, {x -> 2 + Sqrt[11]}}`

Kuivõrd võrrandis olid kõik arvud täisarvud, siis antakse tulemus täpselt. Ligikaudse lahendi saamiseks oleks kordajatest vähemalt üks tulnud ette anda reaalarvuna (nt 7.0) või teha nii nagu järgnevas näites:

`In[28]:= N[%]`

`Out[28]= {{x -> -1.31662}, {x -> 5.31662}}`

Viimane avaldis on reeglite hulk (list). Ainult arvude hulga saame järgmiselt:

`In[29]:= lahendid = x /. %28`

`Out[29]= {-1.31662, 5.31662}`

Kui nüüd soovime omistada esimese väärtuse x_0 -le, teise x_1 -le, tuleks teha järgmiselt:

```
In[30]:= x0 = lahendid[1]
```

```
Out[30]= -1.31662
```

```
In[31]:= x1 = lahendid[2]
```

```
Out[31]= 5.31662
```

```
In[32]:=  $\sqrt{x}$  /. %28
```

```
Out[32]= {0. + 1.14744 i, 2.30578}
```

Kuivõrd Out[9] oli meil list, milles x-l oli kaks väärtust, siis ka $\sqrt{x}$ protseduuri rakendamine andis tulemuseks listi, milles on kaks liiget. Ja kuivõrd esimene väärtus oli negatiivne, siis sellest ruutjuure võtmisel saame tulemuseks loomulikult kompleksarvu.

```
In[33]:= Solve[x5 - 4 x4 + 9 == 0, x]
```

```
Out[33]= {{x -> Root[9 - 4 #14 + #15 &, 1]},
          {x -> Root[9 - 4 #14 + #15 &, 2]}, {x -> Root[9 - 4 #14 + #15 &, 3]},
          {x -> Root[9 - 4 #14 + #15 &, 4]}, {x -> Root[9 - 4 #14 + #15 &, 5]}}
```

Nagu nähe, ei suutnud antud juhul Solve-protseduur anda täpset lahendit. Küll saame ligikaudse:

```
In[34]:= Solve[x5 - 4 x4 + 9.0 == 0, x]
```

```
Out[34]= {{x -> -1.14978}, {x -> -0.0861964 - 1.20257 i},
          {x -> -0.0861964 + 1.20257 i}, {x -> 1.35864}, {x -> 3.96353}}
```

```
In[35]:= Solve[Sin[x] == a, x]
```

```
- Solve::ifun :
  Inverse functions are being used by Solve, so some solutions may not be found.
```

```
Out[35]= {{x -> ArcSin[a]}}
```

Antud juhul kasutatakse siinuse pöördfunktsiooni arcsin, seetõttu on siin ka hoiatus, et mõned lahendid võivad jääda leidmata. Antud võrrandil on tegelikult lõpmata hulk lahendeid, mis erinevad üksteisest $2\pi n$ võrra, n on siin täisarv.

```
In[36]:= Solve[Sin[x] == x, x]
```

```
- Solve::tdep : The equations appear to involve the
  variables to be solved for in an essentially non-algebraic way.
```

```
Out[36]= Solve[Sin[x] == x, x]
```

See tähendab, et ülaltoodud võrrand sisaldab transendentseid funktsioone ning algebraliselt on seda võrrandit võimatu lahendada. Selliste transendentsete võrrandeid lahendatakse numbriliselt ning nende lahendamiseks on funktsioon FindRoot, mida käsitleme hiljem.

Solve[{vp ₁ = pp ₁ , vp ₂ = pp ₂ , ...}, {x ₁ , x ₂ , ...}]	võrrandisüsteemi lahendamine x ₁ , x ₂ , ..suhtes
NSolve[{vp ₁ = pp ₁ , vp ₂ = pp ₂ , ...}, {x ₁ , x ₂ , ...}]	võrrandisüsteemi numbriline lahendamine x ₁ , x ₂ ..suhtes
Eliminate[{vp ₁ = pp ₁ , vp ₂ = pp ₂ , ...}, {x ₁ , x ₂ , ...}]	võrrandisüsteemist muutujate x ₁ , x ₂ elimineerimine
Reduce[{vp ₁ = pp ₁ , vp ₂ = pp ₂ , ...}, {x ₁ , x ₂ , ...}]	annab lihtsustatud võrrandite süsteemi koos võimalike lahendustega
FindRoot[vp = pp, {x, x ₀ }]	leiab numbriliselt lahendi võrrandile, alustades x ₀ -st

```
In[37]:= Solve[{2 x - 4 y + 3 z == 0, x - 3 y - z == 0, x - 3 z == -9}, {x, y, z}]
```

```
Out[37]= {{x -> -117/19, y -> -45/19, z -> 18/19}}
```

```
In[38]:= NSolve[{2 x - 4 y + 3 z == 0, x - 3 y - z == 0, x - 3 z == -9}, {x, y, z}]
```

```
Out[38]= {{x -> -6.15789, y -> -2.36842, z -> 0.947368}}
```

```
In[39]:= Eliminate[{2 x + 4 y + 3 z == 0, x - 3 y == z - 2}, {z}]
```

```
Out[39]= -6 + 5 y == 5 x
```

```
In[40]:= Reduce[{2 x + 4 y + 3 z == 0, x - 3 y == z - 2}, {x, y}]
```

```
Out[40]= x == 1/10 (-8 - 5 z) && y == 1/10 (4 - 5 z)
```

Eliminate käsuga saime kahest kolme muutujaga võrrandist ühe kahe muutujaga võrrandi. Reduce abiga antud juhul saime kaks võimalikku uut võrrandit - esimesel juhul elimineerisime y, teisel juhul x.

Kuigi NSolve leiab võrrandite lahendeid numbriliselt, sobib see protseduur siiski vaid võrrandite jaoks, kus muutujad on mingites astmetes (polünoomvõrrandid). Vastasel juhul tuleb kasutada ikkagi

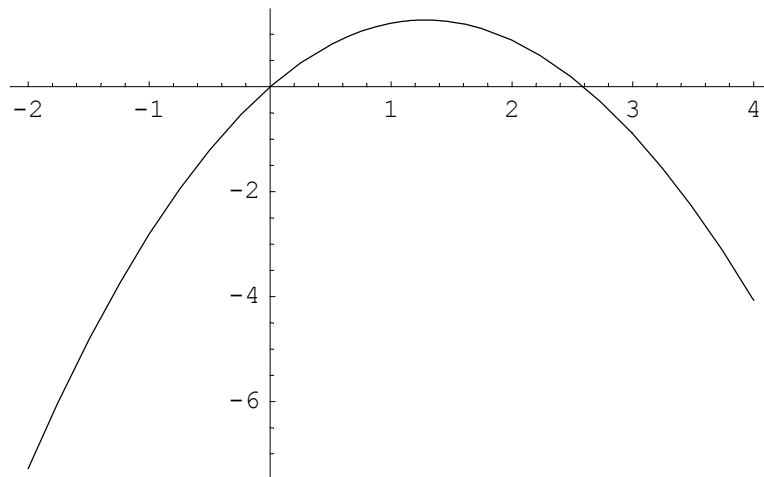
protseduuri FindRoot, mis leiab nullkohad Newtoni iteratsioonimeetodil, kuid selleks tuleb anda piisavalt hea lähtepunkt, st ligikaudne nullkoht. Viimast on üsna hea leida joonise abil.

Järgnevas näites defineerime uue funktsiooni, teeme selle kohta joonise ja siis leiame nullkohad.

```
In[41]:= Clear[f, g];
```

```
In[42]:= g[x_] := 2 x * E0.1 x - x2
```

```
In[43]:= Plot[g[x], {x, -2, 4}]
```



```
Out[43]= - Graphics -
```

Jooniselt on näha, et üks nullkoht on umbes kohal 0, teine kohal 2,5. Tuleb kasutada kaks korda protseduuri FindRoot

```
In[44]:= FindRoot[g[x] == 0, {x, 0}]
```

```
Out[44]= {x -> 0.}
```

```
In[45]:= FindRoot[g[x] == 0, {x, 2.5}]
```

```
Out[45]= {x -> 2.59171}
```

Seega on kaks nullkohta: $x_1 = 0$, $x_2 \approx 2,59$.

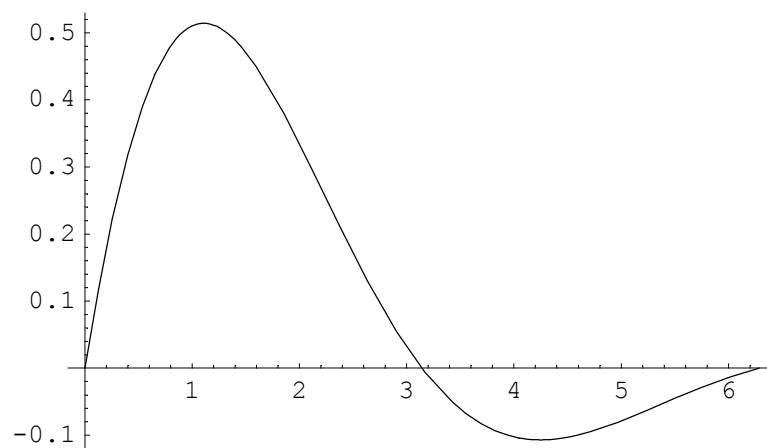
4. Joonised

4.1. Joonised tasandil

<code>Plot[f, {x, x_{min}, x_{max}}]</code>	joonistab funktsiooni f graafiku $x_{\min}$ -st kuni $x_{\max}$ -ni
<code>Plot[{f, g, ...}, {x, x_{min}, x_{max}}]</code>	joonistab mitme funktsiooni graafikud
<code>Show[joonis1, joonis2]</code>	joonistab erinevad joonised ühele kokku

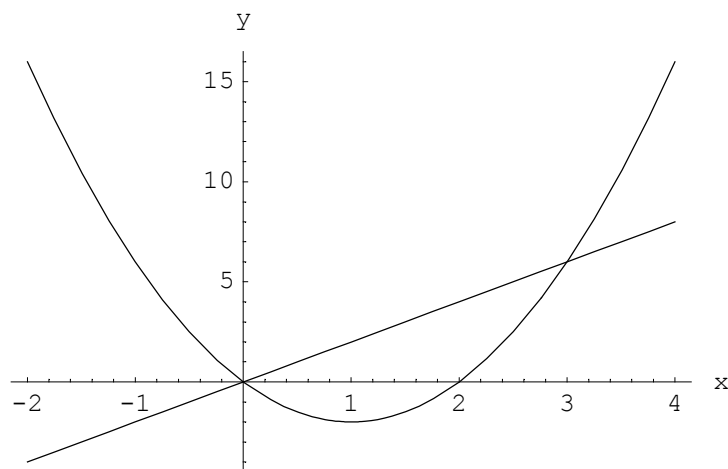
`In[46]:= $Line = 0;`

`In[1]:= Plot[E-0.5 x Sin[x], {x, 0, 2 π}]`



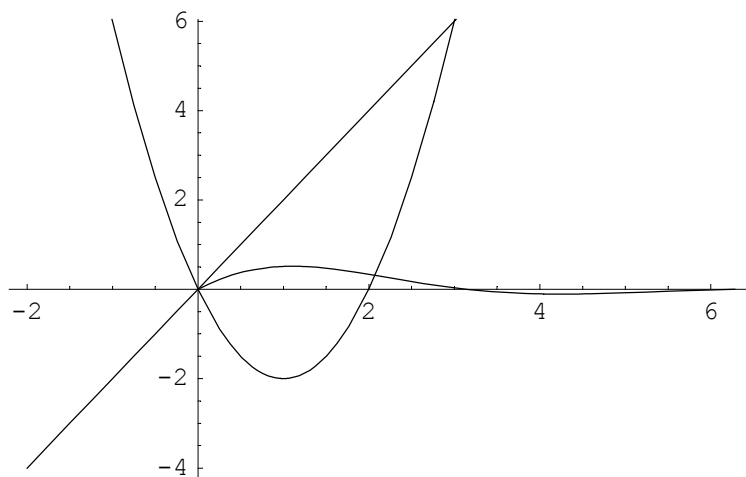
`Out[1]=` ▀ Graphics ▀

```
In[2]:= Plot[{2 x^2 - 4 x, 2 x}, {x, -2, 4}, AxesLabel -> {x, y}]
```



Out[2]= ▀ Graphics ▀

```
In[3]:= Show[%1, %2]
```



Out[3]= ▀ Graphics ▀

Nagu teisest arvutusest näha, võib Plot-protseduurile anda erinevaid lisavõimalusi. Täieliku info võimalustest saab Help-st või kasutades Help-käsku (kehtib kõigi *Mathematica* funktsioonide ja protseduuride kohta).

Mõned näited erinevate võimaluste kasutamise kohta.

```
In[4]:= Plot[{E^x, x, Log[x]}, {x, -3, 3}, PlotStyle→
  {RGBColor[1, 0, 0], RGBColor[0.5, 0.5, 0], RGBColor[0, 0.5, 1]},
  AxesLabel → {"x", "y"}, PlotRange → {{-3, 3}, {-3, 4}},
  AspectRatio → Automatic]
```

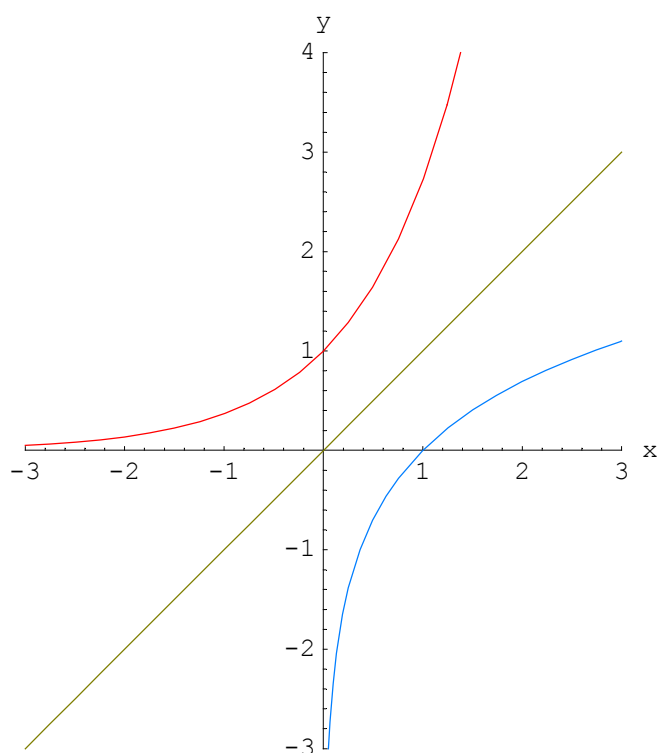
– Plot::plnr : Log[x] is not a machine-size real number at x = -3..

– Plot::plnr : Log[x] is not a machine-size real number at x = -2.7566.

– Plot::plnr : Log[x] is not a machine-size real number at x = -2.49115.

– General::stop :

Further output of Plot::plnr will be suppressed during this calculation.

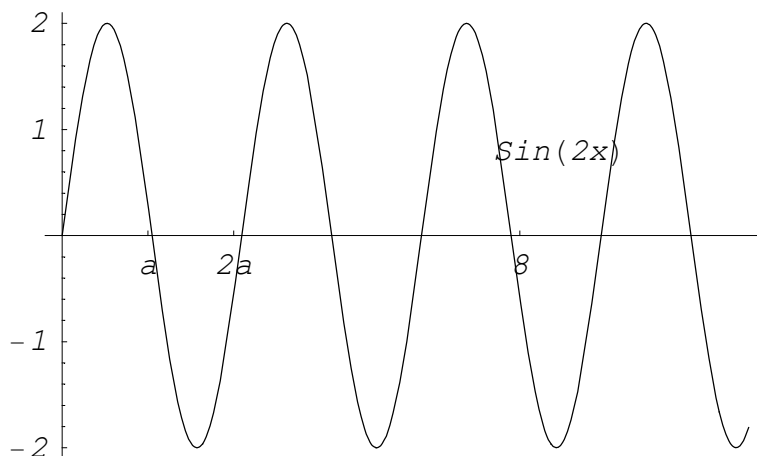


Out[4]= ▀ Graphics ▀

Nagu näha, andis programm ka teada, et funktsiooni $\ln(x)$ ($\text{Log}[x]$) ei saa negatiivsete väärtuste ja nulli kohal leida. Kuivõrd tegemist on kolme funktsiooni graafikuga, siis on ka PlotStyle-s iga funktsiooni jaoks värv ära määratud, st PlotStyle's on tegemist listiga.

```
In[5]:= funkt[x_] := 2 * Sin[2 x]
```

```
In[6]:= Plot[funkt[x], {x, 0, 12},
  TextStyle -> {FontSlant -> "Italic", FontSize -> 12},
  Ticks -> {{1.5, "a"}, {3, "2a"}, 8}, Automatic},
  Epilog -> {Text["Sin(2x)", {8.7, 0.8}, {0, 0}]}]
```



Out[6]= - Graphics -

<i>valik</i>	<i>vaikväärtus</i>	
Axes	Automatic	kas teljed lisada, ntx Axes->No (Yes)
AxesLabel	None	millised tähised telgede juurde panna, nt AxesLabel->{"a","b/s"}
AxesOrigin	Automatic	millises punktis asub telgede ristumiskoht (nullpunkt) nt AxesOrigin->{2,0}
Frame	False	kas tõmmata kast ümber joonise True või False nt Frame->True
GridLines	None	abijooned. Automatic lisab iga olulisema väärtuse kohale joone
PlotRange	Automatic	milline vahemik joonisele pannakse. Tihti joonistab programm "kõige huvitavama" piirkonna, mitte kõike, vastupidine oleks PlotRange->All

Kõikvõimalikke valikuid saab leida nn 'help-käsu abil:

In[7]:= ?? Plot

Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. **More...**

Attributes[Plot] = {HoldAll, Protected}

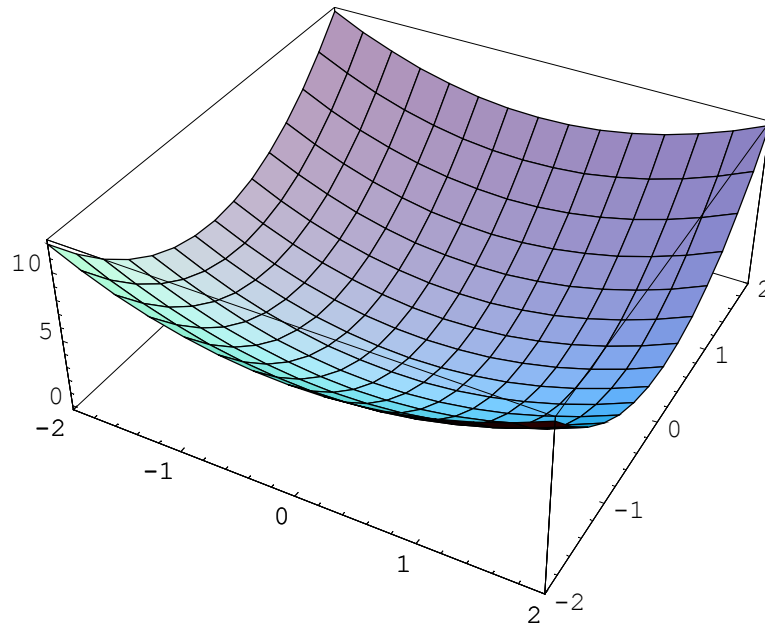
Options[Plot] = {AspectRatio $\rightarrow \frac{1}{\text{GoldenRatio}}$, Axes $\rightarrow$ Automatic, AxesLabel $\rightarrow$ None, AxesOrigin $\rightarrow$ Automatic, AxesStyle $\rightarrow$ Automatic, Background $\rightarrow$ Automatic, ColorOutput $\rightarrow$ Automatic, Compiled $\rightarrow$ True, DefaultColor $\rightarrow$ Automatic, Epilog $\rightarrow$ {}, Frame $\rightarrow$ False, FrameLabel $\rightarrow$ None, FrameStyle $\rightarrow$ Automatic, FrameTicks $\rightarrow$ Automatic, GridLines $\rightarrow$ None, ImageSize $\rightarrow$ Automatic, MaxBend $\rightarrow$ 10., PlotDivision $\rightarrow$ 30., PlotLabel $\rightarrow$ None, PlotPoints $\rightarrow$ 25, PlotRange $\rightarrow$ Automatic, PlotRegion $\rightarrow$ Automatic, PlotStyle $\rightarrow$ Automatic, Prolog $\rightarrow$ {}, RotateLabel $\rightarrow$ True, Ticks $\rightarrow$ Automatic, DefaultFont $\rightarrow$ \$DefaultFont, DisplayFunction $\rightarrow$ \$DisplayFunction, FormatType $\rightarrow$ \$FormatType, TextStyle $\rightarrow$ \$TextStyle}

Jooniste kujundamisel on abiks veel lisapaketid (vt Help-menüüst Add-ons, Standard Packages, Graphics).

4.2. Joonised ruumis.

Plot3D[f, {x, x _{min} , x _{max} }, {y, y _{min} , y _{max} }]	teeb kolmemõõtmelise joonise kui funktsiooni kahest muutujast x ja y
---	--

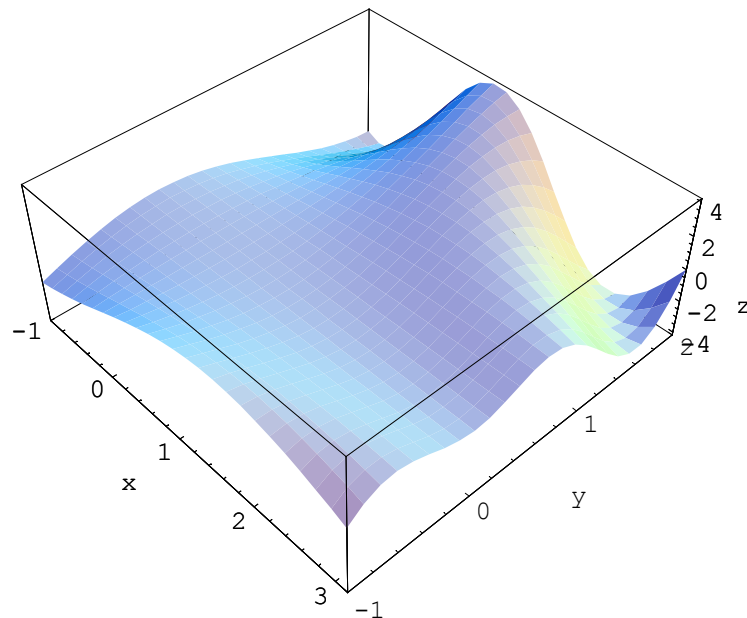
```
In[8]:= Plot3D[x2 + 2 y2, {x, -2, 2}, {y, -2, 2}]
```



```
Out[8]= ▯ SurfaceGraphics ▯
```

```
In[9]:= Plot3D[2 y Sin[x y], {x, -1, π}, {y, -1, 2}, AxesLabel → {x, y, z},  
ViewPoint → {1.6, -1.5, 2}, Mesh → False,  
PlotPoints → 25, PlotLabel → "z(x,y) = 2y sin(x*y)"]
```

$$z(x,y) = 2y \sin(x*y)$$



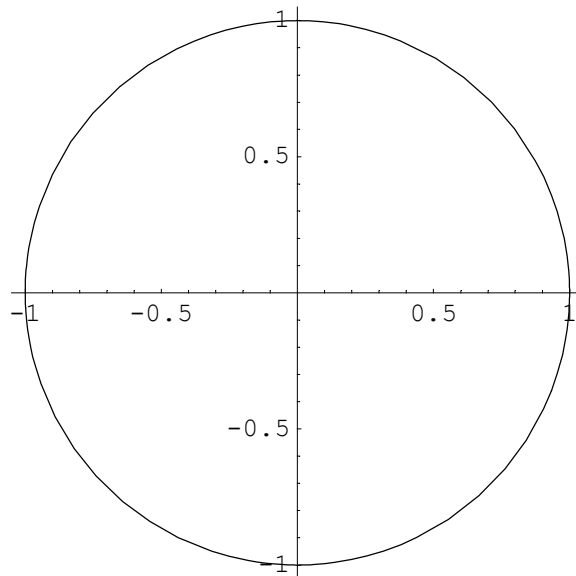
```
Out[9]= ▯ SurfaceGraphics ▯
```

Ka antud leiab kogu võimalike valikute hulga käsu `Plot3D` abil, soovi korral koos näidetega.

4.3. Parameetrilised joonised.

<code>ParametricPlot[{x(t), y(t)}, {t, t₁, t₂}]</code>	parameetrilise	joonise	tegemine
<code>ParametricPlot[{x₁(t), y₁(t)}, {x₂(t), y₂(t)}, ..., {t, t₁, t₂}]</code>	mitme parameetrilise joonise tegemine		
<code>ParametricPlot[{x(t), y(t)}, {t, t₁, t₂}, AspectRatio → Automatic]</code>	parameetriline joonis, selline, et ka joonise õige kuju säiliks		
<code>ParametricPlot3D[{x(t), y(t), z(t)}, {t, t₁, t₂}]</code>	parameetriline joon 3-mõõtmelises ruumis		

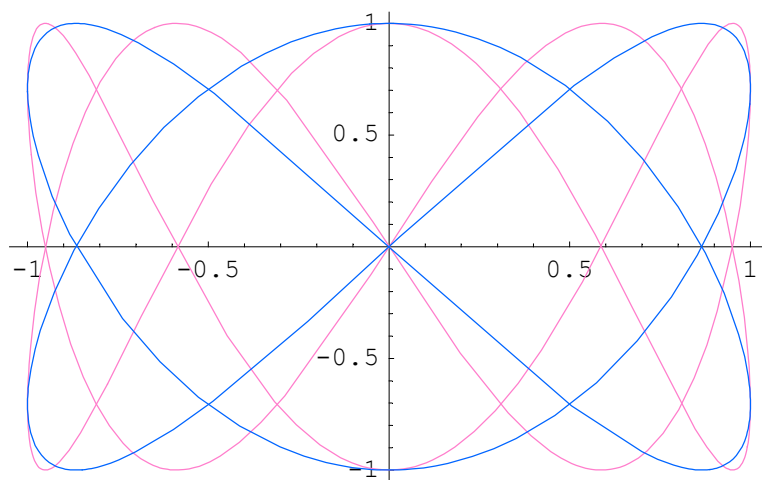
`In[10]:= ParametricPlot[{Sin[t], Cos[t]}, {t, 0, 2 π}, AspectRatio → Automatic]`



`Out[10]=` ▀ Graphics ▀

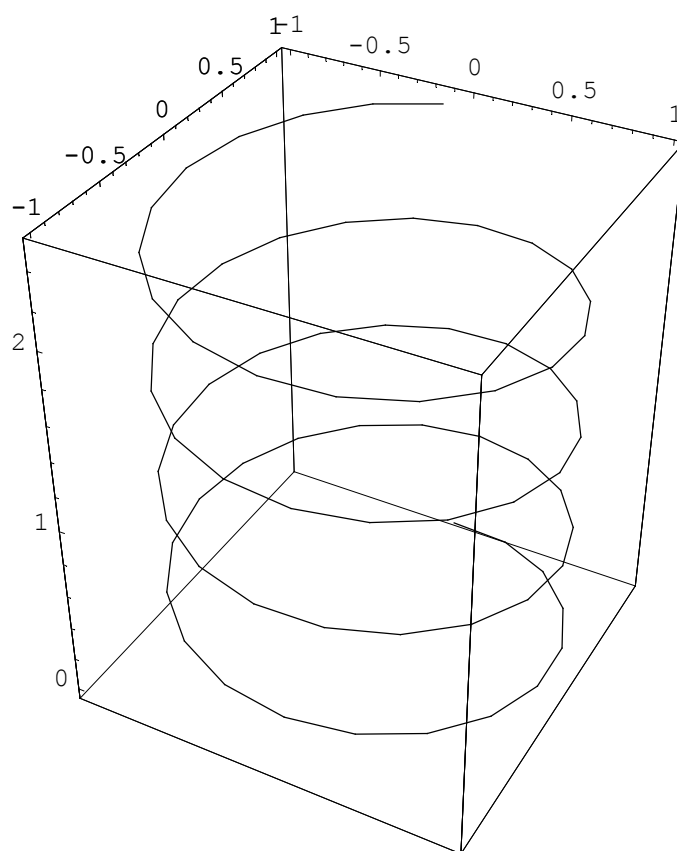
Järgnevalt veel mõned näited nn Lissajoux' kujunditest (eelmine on kõige lihtsam).

```
In[11]:= ParametricPlot[{{Sin[2 t], Sin[5 t]}, {Sin[2 t], Cos[3 t]}}, {t, 0, 2  $\pi$ },  
PlotStyle -> {RGBColor[1, 0.5, 0.8], Hue[0.6]}]
```



Out[11]= ▀ Graphics ▀

```
In[12]:= ParametricPlot3D[{Sin[t], Cos[t], 0.1 t}, {t, 0, 25}]
```



```
Out[12]= ▀ Graphics3D ▀
```

5. Diferentsiaalvõrrandite lahendamine. Integraalide ja diferentsiaalvõrrandite numbriline lahendamine.

<code>DSolve[{vor1_{vas} == vor1_{par}, vor2_{vas} == vor2_{par}, ...}, {f₁[x], f₂[x], ...}, x]</code>	diferentsiaalvõrrandite süsteemi lahendamine
<code>NDSolve[{vor1_{vas} == vor1_{par}, vor2_{vas} == vor2_{par}, ...}, {f₁, f₂, ...}, {x, x₁, x₂}]</code>	diferentsiaalvõrrandite süsteemi numbriline lahendamine lõigus [x ₁ , x ₂]
<code>NIntegrate[f[x], {x, x₁, x₂}]</code>	numbriline integreerimine

Vaatame näitena järgmist diferentsiaalvõrrandit: $\frac{d^2 y(x)}{dx^2} = a y(x) + 1$, mis on teist järku diferentsiaalvõrrand. Selle lahendamine:

`In[13]:= DSolve[ $\partial_x y[x] == a * y[x] + 1$ , y[x], x]`

`Out[13]=` $\left\{ \left\{ y[x] \rightarrow -\frac{1}{a} + e^{a x} C[1] \right\} \right\}$

Selle tähistusviisiga analoogiline on järgmine:

`In[14]:= DSolve[y' [x] == a * y[x] + 1, y[x], x]`

`Out[14]=` $\left\{ \left\{ y[x] \rightarrow -\frac{1}{a} + e^{a x} C[1] \right\} \right\}$

Nagu näha, on siin sees ka kaks konstanti, mis määratakse lisatingimustest.

Anname ette ka lisatingimused ning vaatame, milline näeb välja lahendi graafik.

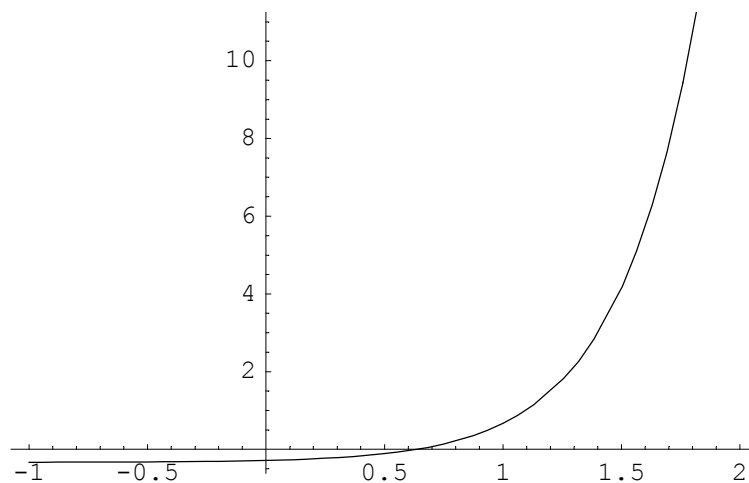
`In[15]:= lahend = DSolve[ $\left\{ \partial_x y[x] == a * y[x] + 1, y[1] == \frac{2}{a} \right\}$ , y[x], x]`

`Out[15]=` $\left\{ \left\{ y[x] \rightarrow \frac{e^{-a} (-e^a + 3 e^{a x})}{a} \right\} \right\}$

`In[16]:= lahend[[1]][[1]][[2]]`

`Out[16]=` $\frac{e^{-a} (-e^a + 3 e^{a x})}{a}$

```
In[17]:= Plot[lahend[[1]][[1]][[2]] /. a -> 3, {x, -1, 2}]
```



```
Out[17]= - Graphics -
```

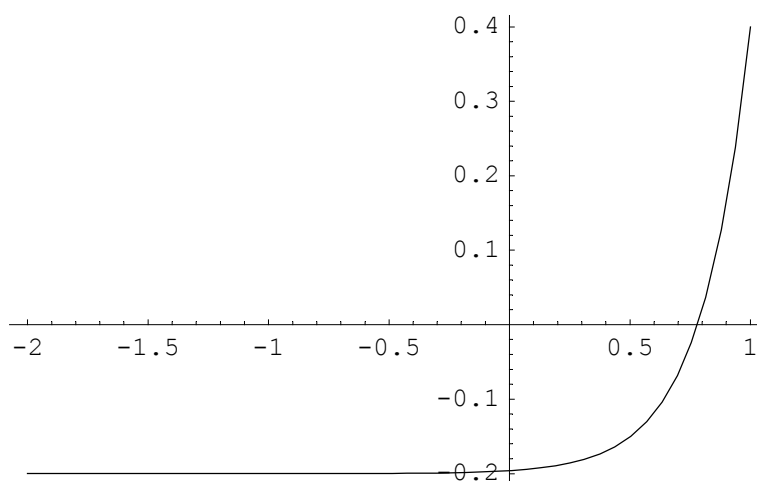
Ülaltoodud tähistus lahend[[1]][[1]][[2]] annab parasjagu kahekordsetes loogelistes sulgudes oleva avaldise teise liikme, st funktsiooni. (Proovige järgi, mis on lahend[[1]], lahend[[1]][[1]] ja lahend[[1]][[1]][[1]]).

Loomulikult oleksime võinud lihtsalt defineerida uue funktsiooni (cut-paste abil) ning siis selle graafiku üles joonistada juhul, kui $a=5$ näiteks:

```
In[18]:= u[x_] := 
$$\frac{e^{-a} (-e^a + 3 e^{a x})}{a}$$

```

```
In[19]:= Plot[u[x] /. a -> 5, {x, -2, 1}, PlotRange -> All]
```



```
Out[19]= - Graphics -
```

Teise näitena vaatame sumbuvat võnkumist kirjeldavat võrrandit koos algtingimustega. See on teist järku lineaarne konstantsete kordajatega diferentsiaalvõrrand:

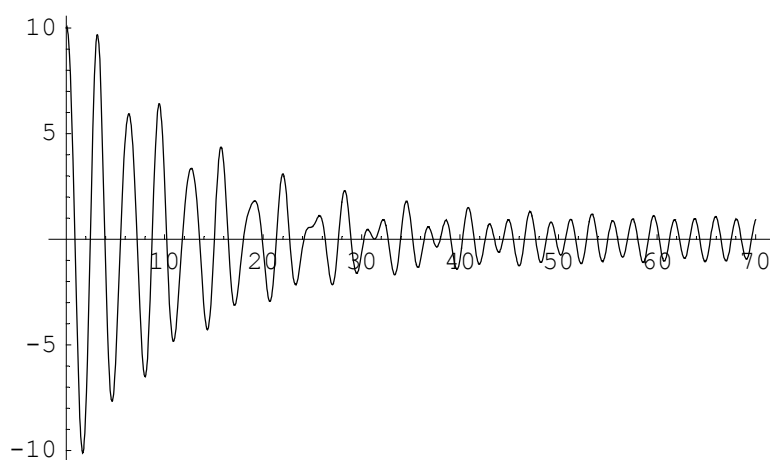
$$y''(x) + 0.01y'(x) + 4y(x) = 0.2 \cos(3x), \quad y'(0) = 2, \quad y(0) = 10$$

```
In[20]:= lahendus = DSolve[{y''[x] + 0.15 y'[x] + 4 y[x] == 5 Cos[3 x],
    y'[0] == 2, y[0] == 10}, y[x], x]
```

```
- Less::nord : Invalid comparison with 0. - 3.99719 i attempted.
```

```
Out[20]= {{y[x] -> (-1.24216 + 0. i) e^(-0.075 x) ((-8.84911 + 0. i) Cos[1.99859 x] +
    (1. + 0. i) e^(0.075 x) Cos[1.00141 x] Cos[1.99859 x] -
    (0.201416 + 0. i) e^(0.075 x) Cos[1.99859 x] Cos[4.99859 x] -
    (0.0748946 + 2.07701 x 10^-19 i) e^(0.075 x) Cos[1.99859 x] Sin[1.00141 x] -
    (1.02981 + 0. i) Sin[1.99859 x] -
    (0.0748946 + 2.07701 x 10^-19 i) e^(0.075 x) Cos[1.00141 x] Sin[1.99859 x] -
    (0.00302209 + 1.00292 x 10^-21 i) e^(0.075 x) Cos[4.99859 x] Sin[1.99859 x] -
    (1. - 6.74197 x 10^-18 i) e^(0.075 x) Sin[1.00141 x] Sin[1.99859 x] +
    (0.00302209 + 1.00292 x 10^-21 i) e^(0.075 x) Cos[1.99859 x] Sin[4.99859 x] -
    (0.201416 + 1.44581 x 10^-19 i) e^(0.075 x) Sin[1.99859 x] Sin[4.99859 x])}}
```

```
In[21]:= Plot[lahendus[[1]][[1]][[2]],
    {x, 0, 70}, PlotPoints -> 100, PlotRange -> All]
```



```
Out[21]= - Graphics -
```

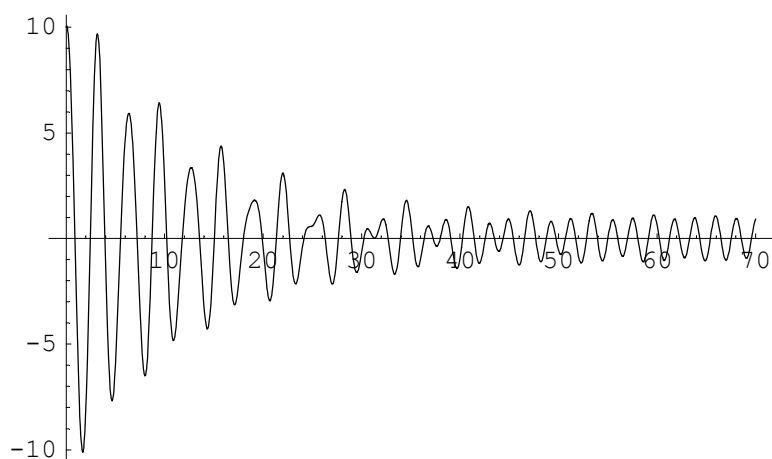
Kuigi lahendis on kompleksarvud sees, on lahend tegelikult reaalne. I ees olevad kordajad on arvutustäpsuse piiiril. Nagu oli näha, arvestas programm seda funktsiooni graafiku joonistamisel. Kuid selleks, et antud juhul neist neist lahti saada, võib kasutada funktsiooni Chop:

```
In[22]:= lahend = Chop[%20]
```

```
Out[22]= {{y[x] → -1.24216 e-0.075 x
(-8.84911 Cos[1.99859 x] + 1. e0.075 x Cos[1.00141 x] Cos[1.99859 x] -
0.201416 e0.075 x Cos[1.99859 x] Cos[4.99859 x] -
0.0748946 e0.075 x Cos[1.99859 x] Sin[1.00141 x] -
1.02981 Sin[1.99859 x] - 0.0748946 e0.075 x Cos[1.00141 x]
Sin[1.99859 x] - 0.00302209 e0.075 x Cos[4.99859 x] Sin[1.99859 x] -
1. e0.075 x Sin[1.00141 x] Sin[1.99859 x] +
0.00302209 e0.075 x Cos[1.99859 x] Sin[4.99859 x] -
0.201416 e0.075 x Sin[1.99859 x] Sin[4.99859 x])}}
```

Chop[avaldis]	viskab ära liikmed, mis on väiksemad kui 10^{-10}
Chop[avaldis, dx]	viskab ära liikmed, mis on väiksemad kui dx

```
In[23]:= Plot[lahend[[1]][[1]][[2]],
{x, 0, 70}, PlotPoints → 100, PlotRange → All]
```



```
Out[23]= - Graphics -
```

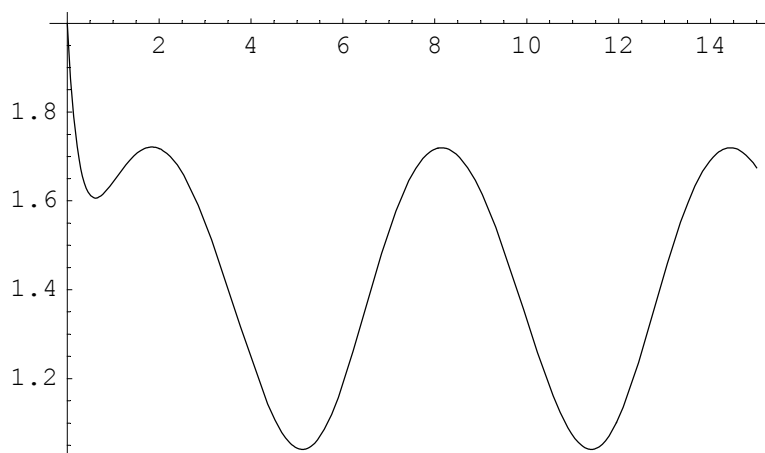
```
In[24]:= Clear[x, u, t, y]
```

Tihti pole võimalik leida diferentsiaalvõrrandi või võrrandite süsteemi lahendit analüütiliselt. Sel juhul tuleb alati ette anda ka algtingimused. Tulemuseks saame funktsiooni, mis on esitatud interpoleeritud funktsioonina. Viimast saame omakorda lähemalt uurida graafiku abil.

```
In[25]:= funkt = NDSolve[{u'[t] == 2 - (u[t])2 + Sin[t], u[0] == 2}, u, {t, 0, 15}]
```

```
Out[25]= {{u → InterpolatingFunction[{{0., 15.}}, <>]}}
```

```
In[26]:= Plot[Evaluate[u[t] /. funkt[[1]][[1]]], {t, 0, 15}]
```



```
Out[26]= - Graphics -
```

Ülaltoodud protseduuris Evaluate[u[t]/.funkt[[1]][[1]]] ongi käsklus, mille kohaselt programm võtab funktsiooni u[t] väärtuse nii nagu diferentsiaalvõrrandi lahendamisel leitud.

```
In[27]:= Evaluate[u[3] /. funkt[[1]][[1]]]
```

```
Out[27]= 1.55248
```

```
In[28]:= 2 * Sin[4] * Evaluate[u[4] /. funkt[[1]][[1]]]
```

```
Out[28]= -1.88302
```

Võime lahendada ka mitmest diferentsiaalvõrrandist koosneva süsteemi.

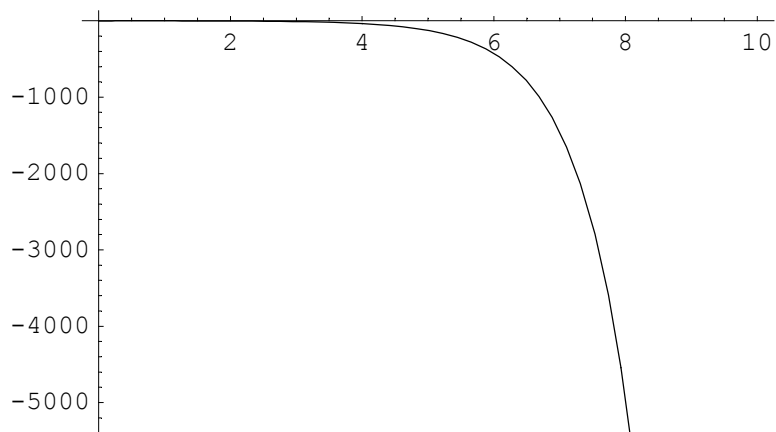
```
In[29]:= sol = NDSolve[{x'[t] == 2 (x[t])^2 - 0.1 (y[t])^2,
  y'[t] == -x[t] + y[t], x[0] == -2, y[0] == 1}, {x, y}, {t, 0, 10}]
```

```
Out[29]= {{x -> InterpolatingFunction[{{0., 10.}}, <>],
  y -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[30]:= sol[[1]][[1]]
```

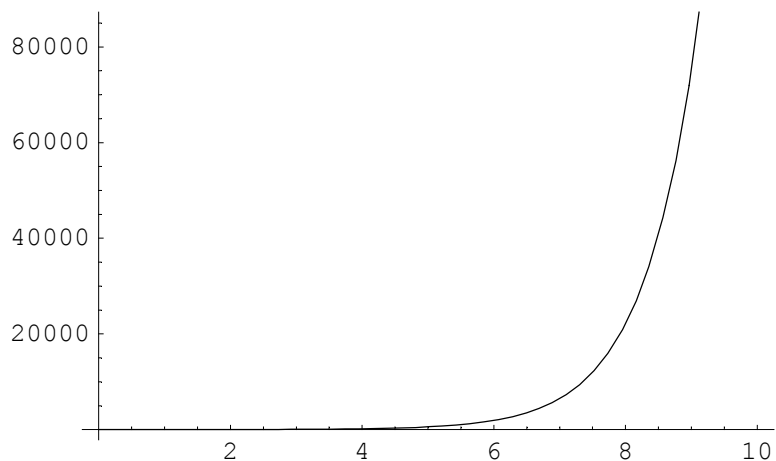
```
Out[30]= x -> InterpolatingFunction[{{0., 10.}}, <>]
```

```
In[31]:= Plot[Evaluate[x[t] /. sol[[1]][[1]]], {t, 0, 10}]
```



Out[31]= - Graphics -

```
In[32]:= Plot[Evaluate[y[t] /. sol[[1]][[2]]], {t, 0, 10}]
```



Out[32]= - Graphics -

Mõned näited integraali numbrilise arvutuse kohta. Esmalt - kuidas oleks tulemus tavakujul ja NIntegrate abil.

```
In[33]:=  $\int_1^3 \frac{e^{x^2}}{x^3} dx$ 
```

```
Out[33]=  $\frac{1}{2} (e - \text{ExpIntegralEi}[1]) + \frac{1}{18} (-e^9 + 9 \text{ExpIntegralEi}[9])$ 
```

```
In[34]:= N[%33]
```

Out[34]= 69.1794

```
In[35]:= NIntegrate[Ex2/x3, {x, 1, 3}]
```

```
Out[35]= 69.1794
```

6. Vektorid, maatriksid, listid.

6.1. Tehteid listidega

```
In[36]:= $Line = 0;
```

Listiks võiks nimetada igasugust tüüpi avaldis, mis avaldatakse loogeliste sulgude vahel ning avaldise liikmed on omavahel eraldatud komadega. Mõned näited:

```
In[1]:= v = {2, 3, 4}
```

```
Out[1]= {2, 3, 4}
```

Toodud üherealist listi võime nimetada ka vektoriks. Allpool vataleme ka pikemalt, kuidas vektoritega tehteid teha. Mõned lihtsamad aga siin.

```
In[2]:= Sqrt[v]
```

```
Out[2]= {√2, √3, 2}
```

```
In[3]:= N[%]
```

```
Out[3]= {1.41421, 1.73205, 2.}
```

```
In[4]:= v + 2
```

```
Out[4]= {4, 5, 6}
```

```
In[5]:= v * a
```

```
Out[5]= {2 a, 3 a, 4 a}
```

```
In[6]:= xv
```

```
Out[6]= {x2, x3, x4}
```

In[7]:= $\mathbf{D}[\mathbf{x}^{\mathbf{v}}, \mathbf{x}]$

Out[7]= $\{2\,x, 3\,x^2, 4\,x^3\}$

In[8]:= $\int_0^3 \mathbf{x}^{\mathbf{v}} \, d\mathbf{x}$

Out[8]= $\{9, \frac{81}{4}, \frac{243}{5}\}$

In[9]:= $\int_0^3 \mathbf{v}^{\mathbf{x}} \, d\mathbf{x}$

Out[9]= $\{\frac{7}{\text{Log}[2]}, \frac{26}{\text{Log}[3]}, \frac{63}{\text{Log}[4]}\}$

In[10]:= $\mathbf{N}[\%]$

Out[10]= $\{10.0989, 23.6662, 45.4449\}$

Seega võib antud juhul vaadata listi kui mingit andmetekogumit. Kasutades listi mingites algebralistes operatsioonides, rakendatakse neid operatsioone kõigile listi liikmetele. Mingi algoritmi järgi saab listi genereerida kasutades käsku Table.

In[11]:= $\mathbf{li} = \mathbf{Table}[\mathbf{i}^2, \{\mathbf{i}, 5\}]$

Out[11]= $\{1, 4, 9, 16, 25\}$

In[12]:= $\mathbf{li2} = \mathbf{Table}[\mathbf{i} - 3, \{\mathbf{i}, 5\}]$

Out[12]= $\{-2, -1, 0, 1, 2\}$

In[13]:= $\mathbf{li} + \mathbf{li2}$

Out[13]= $\{-1, 3, 9, 17, 27\}$

In[14]:= $\mathbf{Table}[\mathbf{Sin}[\mathbf{n}/5], \{\mathbf{n}, -2, 4\}]$

Out[14]= $\{-\text{Sin}[\frac{2}{5}], -\text{Sin}[\frac{1}{5}], 0, \text{Sin}[\frac{1}{5}], \text{Sin}[\frac{2}{5}], \text{Sin}[\frac{3}{5}], \text{Sin}[\frac{4}{5}]\}$

In[15]:= $\mathbf{maat} = \mathbf{Table}[\mathbf{x}^{\mathbf{i}} + \mathbf{y}^{\mathbf{j}}, \{\mathbf{i}, 3\}, \{\mathbf{j}, 2\}]$

Out[15]= $\{\{x + y, x + y^2\}, \{x^2 + y, x^2 + y^2\}, \{x^3 + y, x^3 + y^2\}\}$

```
In[16]:= TableForm[%]
```

```
Out[16]//TableForm=
```

$$\begin{array}{cc} x + y & x + y^2 \\ x^2 + y & x^2 + y^2 \\ x^3 + y & x^3 + y^2 \end{array}$$

```
In[17]:= Table[Random[], {4}]
```

```
Out[17]= {0.0136174, 0.992316, 0.577466, 0.724002}
```

Table[f, {imax}]	genereeritakse list, milles on imax liiget, antakse funktsiooni f abil
Table[f, {i, imax}]	genereeritakse list, mille liikmed antakse f abil 1-st kuni imax-ni
Table[f, {i, imin, imax}]	genereeritakse list, mille liikmed antakse f abil imin-st kuni imax-ni
Table[f, {i, imin, imax, di}]	genereeritakse list, mille liikmed antakse f abil 1-st kuni imax-ni sammuga di
Table[f, {i, imin, imax}, {j, jmin, jmax}]	genereeritakse tabel (või maatriks) funktsiooni f abil
TableForm[list]	list kuvatakse tabeli kujul

Listidest ja maatriksitest saab välja eraldada mõningaid liikmeid, teha tehteid nendega ning koostada neist uusi liste.

```
In[18]:= maks = maat[[3]]
```

```
Out[18]= {x^3 + y, x^3 + y^2}
```

```
In[19]:= li3 = li2[[2]]
```

```
Out[19]= -1
```

```
In[20]:= li2[{{3, 4, 5}}]
```

```
Out[20]= {0, 1, 2}
```

```
In[21]:= li3 * maks
```

```
Out[21]= {-x^3 - y, -x^3 - y^2}
```

t[[i]] või Part[t,i]	annab listi t alamlisti i (vt maks) või ühe elemendi (vt li3) alamlistiks võib olla ka vaid üks listi liige
----------------------	--

$t[\{i_1, i_2, \dots, i_n\}]$	genereeritakse uus n liikmeline list vanast listist t
$t[i_j, \dots]$ ehk $\text{Part}[t, i_j, \dots]$	antakse listi t osa i-nda osa j-s osa

In[22]:= $t = \{\{2, 3\}, \{a + z, g\}, \{d, \sqrt{3 + 5}\}\}$

Out[22]= $\{\{2, 3\}, \{a + z, g\}, \{d, 2\sqrt{2}\}\}$

In[23]:= $t[[2]]$

Out[23]= $\{a + z, g\}$

In[24]:= $t[[2, 1]]$

Out[24]= $a + z$

In[25]:= $t[[2]][[1]]$

Out[25]= $a + z$

In[26]:= $t[[2]][[1]][[1]]$

Out[26]= a

In[27]:= $t[[2, 1, 1]]$

Out[27]= a

Nagu ülaltoodud näidetest näha võib sel moel tuua ka pikast avaldisest välja erinevaid osi, antud juhul listi ühe osa üht liidetavatest. Mõned sarnased näited veel, kus saab mingist pikemast avaldisest osa liikmeid välja tuua.

In[28]:= **vorrandidlahend =**

Solve[{x + 2 y - 3.0 z == 2, y - z == 0.5, x + 2 y - 6 z == 4}, {x, y, z}]

Out[28]= $\{\{x \rightarrow 0.333333, y \rightarrow -0.166667, z \rightarrow -0.666667\}\}$

In[29]:= **iks = vorrandidlahend[[1]][[1]]**

Out[29]= $x \rightarrow 0.333333$

In[30]:= **ks = vorrandidlahend[[1]][[1]][[2]]**

Out[30]= 0.333333

Siin kasutasime asjaolu, et võrrandi lahend antakse listina - see on antud loogeliste sulgude vahel. Sel listil on omakorda 3 liiget ($x \rightarrow 0.333333$, $y \rightarrow 0.166667$, $z \rightarrow 0.666667$), mis koosnevad kahest liikmest.

```
In[31]:= ig = vorrandilahend[[1]][[2]][[2]]
```

```
Out[31]= -0.166667
```

```
In[32]:= ze = vorrandilahend[[1]][[3]][[2]]
```

```
Out[32]= -0.666667
```

```
In[33]:= Clear[x, t]
```

```
In[34]:= difvor = DSolve[{2.0 x''[t] + 0.5 x'[t] + 5 x[t] == 2.5 Sin[5 t],  
x'[0] == 2, x[0] == -1}, x[t], t]
```

```
- Less::nord : Invalid comparison with 0. - 3.15238 i attempted.
```

```
Out[34]= {{x[t] -> (-0.498462 - 0.682756 i) e^(-0.125 - 1.57619 i) t  
(-0.304627 - 0.952472 i) + (1. + 1.0842 x 10^-19 i) e^(3.15238 i t) +  
(0.00214621 - 0.00293973 i) e^(0.125 + 1.57619 i) t Cos[5. t] +  
(0.0386318 - 0.0529151 i) e^(0.125 + 1.57619 i) t Sin[5. t])}}
```

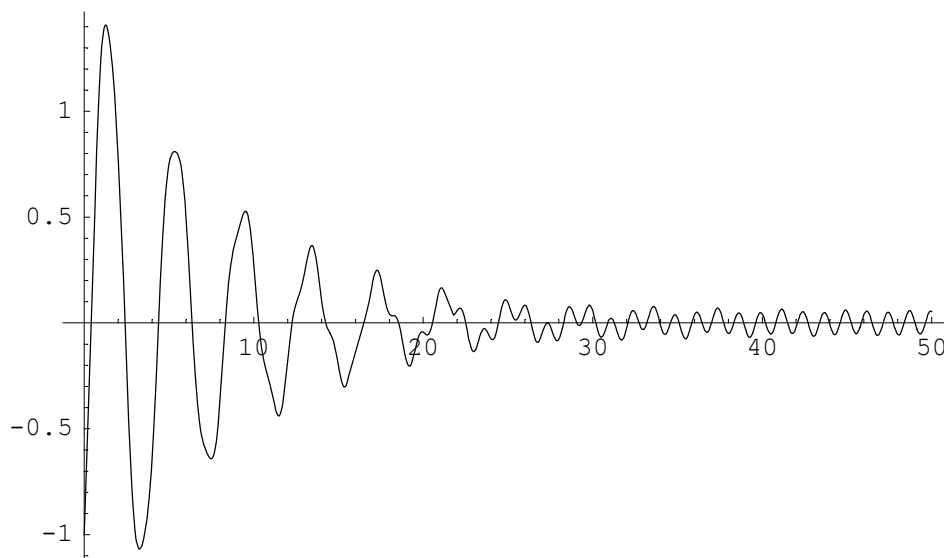
```
In[35]:= Chop[%]
```

```
Out[35]= {{x[t] ->  
(-0.498462 - 0.682756 i) e^(-0.125 - 1.57619 i) t (-0.304627 - 0.952472 i) +  
1. e^(3.15238 i t) + (0.00214621 - 0.00293973 i) e^(0.125 + 1.57619 i) t Cos[5. t] +  
(0.0386318 - 0.0529151 i) e^(0.125 + 1.57619 i) t Sin[5. t])}}
```

Allpool defineerime suuruse fun listi abil.

```
In[36]:= fun := difvor[[1]][[1]][[2]]
```

```
In[37]:= Plot[fun, {t, 0, 50}, PlotRange -> All]
```



```
Out[37]= - Graphics -
```

6.2. Vektorid ja maatriksid.

Vektor on samasugustest elementidest koosnev üherealine list (üherealine maatriks). Maatriksil võib mõõtmeid olla tunduvalt rohkem, kuigi enamkasutatav on kahemõõtmeline maatriks.

$\{a, b, c\}$	vektor (a,b,c) ehk ühemõõtmeline 1x3 maatriks
$\{\{a,b\},\{c,d\}\}$	kahemõõtmeline 2x2 maatriks $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$
$\{\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\}, \{c_1, c_2, c_3\}, \{d_1, d_2, d_3\}\}$	kolmemõõtmeline 2x2x3-maatriks
$c \cdot v$	vektori korrutamine skalaariga
$a.b$	vektorite skalaarkorrutis
$\text{Cross}[a,b]$	vektorite vektorkorrutis

```
In[38]:= a = {3, 1, 2};
          b = {3, -2, 0};
```

```
In[40]:= a.b
```

```
Out[40]= 7
```

```
In[41]:= Cross[a, b]
```

```
Out[41]= {4, 6, -9}
```

Juhul, kui nii a kui ka b on mõlemad samade mõõtmetega vektorid, tähendab $a \times b$ uut vektorit, mille komponentideks on vastavate komponentide korrutised.

```
In[42]:= a.b
```

```
Out[42]= {9, -2, 0}
```

```
In[43]:= Clear[a, b, c, v]
```

```
In[44]:= a = {{2, 3, 4}, {-1, 0, 3}};
          b = {{0, -2}, {3, 2}, {-1, -2}};
```

```
In[46]:= MatrixForm[a]
```

```
Out[46]//MatrixForm=

$$\begin{pmatrix} 2 & 3 & 4 \\ -1 & 0 & 3 \end{pmatrix}$$

```

```
In[47]:= MatrixForm[b]
```

```
Out[47]//MatrixForm=

$$\begin{pmatrix} 0 & -2 \\ 3 & 2 \\ -1 & -2 \end{pmatrix}$$

```

```
In[48]:= a.b
```

```
Out[48]= {{5, -6}, {-3, -4}}
```

```
In[49]:= b.a
```

```
Out[49]= {{2, 0, -6}, {4, 9, 18}, {0, -3, -10}}
```

Tehted vektoritega:

Table[f, {i,n}]	n-mõõtmelise vektori loomine funktsiooni f abil, $i=1,2,\dots,n$
Array[a,n]	n-mõõtmelise vektori loomine kujul {a[1], a[2], ..., a[n]}
Range[n]	listi {1,2,...,n} loomine
Range[n ₁ , n ₂]	listi {n ₁ , n ₁ + 1, ..., n ₂ - 1, n ₂ } loomine
Range[n ₁ , n ₂ , dn]	listi {n ₁ , n ₁ + dn, ..., n ₂ - dn, n ₂ } loomine
list[[i]] ehk Part[list, i]	annab listi i-nda elemendi

Length[list]	annab listis olevate elementide arvu
ColumnForm[list]	näitab vektorit ekraanil ühes veerus

Tehted maatriksitega:

Table[f, {i,n},{j,n}]	$m \times n$ maatriksi loomine funktsiooni f abil
Array[a,{m,n}]	$m \times n$ maatriksi loomine, kus maatriksi i -nda rea j -nda veeru element
	antakse kujul $a[i,j]$
IdentityMatrix[n]	$n \times n$ ühikmaatriksi loomine
DiagonalMatrix[list]	luuakse ruutmaatriks, mille diagonaalil on listi elemendid
list[[i]] ehk Part[list,i]	antakse maatriksi <i>list</i> i -rida
list[[i,j]] ehk Part[list,i,j]	antakse listi i -nda rea j -nda veeru element
Dimensions[list]	annab listi dimensioonid
MatrixForm[list]	maatriks kuvatakse ekraanile maatriksina
c m	maatriksi m korrutamine skalaariga c
a.b	maatriksite korrutamine
Inverse[m]	pöördmaatriksi leidmine
MatrixPower[m,n]	maatriksi m n -s aste
Det[m]	determinant
Tr[m]	maatriksi jälg
Transpose[m]	transponeeritud maatriksi leidmine
Eigenvalues[m]	maatriksi omaväärtuste leidmine
Eigenvectors[	omavektorite leidmine
Eigenvalues[N[m]], Eigenvectors[N[m]]	numbrilised omaväärtused ja omavektorid

In[50]:= **mx = Table[s^{i-j}, {i, 3}, {j, 4}]**

Out[50]= $\left\{ \left\{ 1, \frac{1}{s}, \frac{1}{s^2}, \frac{1}{s^3} \right\}, \left\{ s, 1, \frac{1}{s}, \frac{1}{s^2} \right\}, \left\{ s^2, s, 1, \frac{1}{s} \right\} \right\}$

In[51]:= **MatrixForm[mx]**

Out[51]//MatrixForm=

$$\begin{pmatrix} 1 & \frac{1}{s} & \frac{1}{s^2} & \frac{1}{s^3} \\ s & 1 & \frac{1}{s} & \frac{1}{s^2} \\ s^2 & s & 1 & \frac{1}{s} \end{pmatrix}$$

In[52]:= **sx = Table[i - j, {i, 4}, {j, 3}]**

Out[52]= $\left\{ \{0, -1, -2\}, \{1, 0, -1\}, \{2, 1, 0\}, \{3, 2, 1\} \right\}$

In[53]:= **MatrixForm[sx]**

Out[53]//MatrixForm=

$$\begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}$$

In[54]:= **mx.sx**

$$\begin{aligned} \text{Out[54]} = & \left\{ \left\{ \frac{3}{s^3} + \frac{2}{s^2} + \frac{1}{s}, -1 + \frac{2}{s^3} + \frac{1}{s^2}, -2 + \frac{1}{s^3} - \frac{1}{s} \right\}, \right. \\ & \left\{ 1 + \frac{3}{s^2} + \frac{2}{s}, \frac{2}{s^2} + \frac{1}{s} - s, -1 + \frac{1}{s^2} - 2s \right\}, \\ & \left. \left\{ 2 + \frac{3}{s} + s, 1 + \frac{2}{s} - s^2, \frac{1}{s} - s - 2s^2 \right\} \right\} \end{aligned}$$

In[55]:= **MatrixForm[%]**

Out[55]//MatrixForm=

$$\begin{pmatrix} \frac{3}{s^3} + \frac{2}{s^2} + \frac{1}{s} & -1 + \frac{2}{s^3} + \frac{1}{s^2} & -2 + \frac{1}{s^3} - \frac{1}{s} \\ 1 + \frac{3}{s^2} + \frac{2}{s} & \frac{2}{s^2} + \frac{1}{s} - s & -1 + \frac{1}{s^2} - 2s \\ 2 + \frac{3}{s} + s & 1 + \frac{2}{s} - s^2 & \frac{1}{s} - s - 2s^2 \end{pmatrix}$$

In[56]:= **MatrixForm[2 sx]**

Out[56]//MatrixForm=

$$\begin{pmatrix} 0 & -2 & -4 \\ 2 & 0 & -2 \\ 4 & 2 & 0 \\ 6 & 4 & 2 \end{pmatrix}$$

In[57]:= **Det[mx.sx]**

Out[57]= 0

In[58]:= **Tr[mx.sx]**

$$\text{Out[58]} = \frac{3}{s^3} + \frac{4}{s^2} + \frac{3}{s} - 2s - 2s^2$$

In[59]:= **Eigenvectors[mx.sx]**

$$\text{Out[59]} = \left\{ \left\{ -\frac{1-s^2-2s^3}{3+2s+s^2}, 0, 1 \right\}, \left\{ -\frac{2+s-s^3}{3+2s+s^2}, 1, 0 \right\}, \left\{ \frac{1}{s^2}, \frac{1}{s}, 1 \right\} \right\}$$

In[60]:= **Eigenvalues[mx.sx]**

$$\text{Out[60]} = \left\{ 0, 0, \frac{3+4s+3s^2-2s^4-2s^5}{s^3} \right\}$$

```
In[61]:= ahaa = {{2, 3}, {1, 7}}
```

```
Out[61]= {{2, 3}, {1, 7}}
```

```
In[62]:= Eigenvalues[N[ahaa]]
```

```
Out[62]= {7.54138, 1.45862}
```

```
In[63]:= Eigenvectors[N[ahaa]]
```

```
Out[63]= {{-0.476089, -0.879397}, {-0.984104, 0.177592}}
```

Seega saame maatriksi ahaa avaldada omaväärtuste ja omavektorite abil järgmiselt:

$$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} -0.4761 \\ -0.8794 \end{pmatrix} = 7.54138 \begin{pmatrix} -0.4761 \\ -0.8794 \end{pmatrix} \text{ ning } \\ \begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} -0.9841 \\ -0.1776 \end{pmatrix} = 1.45862 \begin{pmatrix} -0.9841 \\ -0.1776 \end{pmatrix}$$

Ülal on toodud kaks erinevat võimalust maatriksite defineerimiseks - Array ning Table abil.

Table abil on mugavam luua maatriksit mingi algoritmi järgi. Array-ga loodud maatriksis on maatriksi elementide muutmine mõneti lihtsam (sarnane erinevates programmeerimiskeeltega).

Seetõttu on ka Array kasutamine programmeerimisel üsna loomulik.

```
In[64]:= Clear[s, v, a, b]
```

```
In[65]:= s = Table[i * j, {i, 3}, {j, 3}]
```

```
Out[65]= {{1, 2, 3}, {2, 4, 6}, {3, 6, 9}}
```

```
In[66]:= s[[2]]
```

```
Out[66]= {2, 4, 6}
```

```
In[67]:= s[[2]][[3]]
```

```
Out[67]= 6
```

```
In[68]:= s[[2]][[3]] = 2
```

– *Set::setps : s[[2]] in assignment of part is not a symbol.*

```
Out[68]= 2
```

Nagu näha, keelatakse selline omistamine ära.

```
In[69]:= a = Array[v, {3, 3}]
```

```
Out[69]= {{v[1, 1], v[1, 2], v[1, 3]},  
          {v[2, 1], v[2, 2], v[2, 3]}, {v[3, 1], v[3, 2], v[3, 3]}}
```

```
In[70]:= v[1, 1] = 2
```

```
Out[70]= 2
```

```
In[71]:= For[i = 1, i ≤ 3, i = i + 1,  
            For[j = 1, j ≤ 3, j = j + 1,  
              v[i, j] = i * j  
            ]  
          ]
```

```
In[72]:= a
```

```
Out[72]= {{1, 2, 3}, {2, 4, 6}, {3, 6, 9}}
```

```
In[73]:= a[[2]]
```

```
Out[73]= {2, 4, 6}
```

```
In[74]:= v[2, 3]
```

```
Out[74]= 6
```

Selliselt oli ka käsu `Array` korral maatriksile `a` võimalik omistada samasuguseid väärtusi nagu maatriksis `s`. Erinevalt `Table`-käsu kasutamisest, saab maatriksi elementide väärtusi pärast maatriksi loomist muuta.

6.3. Protseduure (ja tehteid) maatriksite ja listide osadega.

First[list]	listi esimene liige
Last[list]	listi viimane liige
Part[list,n] ehk list[[n]]	listi n -s liige
Part[list,-n] ehk list[[-n]]	listi tagantpoolt lugedes n -s liige
Part[list, { $n_1, n_2, \dots$ }]	uus list, mille elementideks on vana listi elemendid positsioonidel $n_1, n_2, \dots$
Take[list,n]	listi n esimest liiget
Take[list,-n]	listi n viimast liiget
Take[list, { m,n }]	listi liikmed alates m -ndast kuni n -ndani (kaasa arvatud)
Take[list, { $m,n,step$ }]	listi liikmed alates m -nast kuni n -ndani sammuga s
Rest[list]	list ilma esimese liikmeta
Drop[list,n]	list ilma esimese n liikmeta
Drop[list,-n]	list ilma viimase n liikmeta
Drop[list, { m,n }]	list, millest elemendid m -ndast kuni n -ndani on ära jäetud
Drop[list, { m,n,s }]	list, millest on välja jäetud elemendid m -ndast kuni n -ndani sammuga s

```
In[75]:= $Line = 0;
```

```
In[1]:= Clear[a, b, c, d, e, f, g, h, t]
```

```
In[2]:= t = {a, b, c, d, e, f, g}
```

```
Out[2]= {a, b, c, d, e, f, g}
```

```
In[3]:= Last[t]
```

```
Out[3]= g
```

Leiame listi t 3.nda elemendi

```
In[4]:= t[[3]]
```

```
Out[4]= c
```

```
In[5]:= t[{{2, 5}}]
```

```
Out[5]= {b, e}
```

```
In[6]:= Take[t, -2]
```

```
Out[6]= {f, g}
```

Teeme uue listi, milles on kõik listi t liikmed, välja arvatud 4.s

```
In[7]:= Drop[t, {4, 4}]
```

```
Out[7]= {a, b, c, e, f, g}
```

Part[list,i,j,...] ehk list[[i]][[j]][[...]] ehk list[[i,j,...]]
 listi i -nda liikme j -nda alamlisti ... (jne)
 Part[list, {i₁, i₂, ...}, {j₁, j₂, ...}, ...] või list[[{i₁, i₂, ...}, {j₁, j₂, ...}, ...]]
 uus list vana listi elemendidest
 Part[list,...,All,...] ehk list[[...,All,...]]
 list elementidest, milles on kõik antud tasandi elemendid sees
 Part[list,All,i] ehk list[[All,i]]
 listi i -s veerg

```
In[8]:= Clear[t]
```

```
In[9]:= t = {{a, b, c}, {d, e, f}}
```

```
Out[9]= {{a, b, c}, {d, e, f}}
```

```
In[10]:= t[[2]]
```

```
Out[10]= {d, e, f}
```

```
In[11]:= t[[2, 3]]
```

```
Out[11]= f
```

```
In[12]:= t[[2]][[3]]
```

```
Out[12]= f
```

```
In[13]:= t[{{2, 2, 1, 1}}]
```

```
Out[13]= {{d, e, f}, {d, e, f}, {a, b, c}, {a, b, c}}
```

See andis uue listi, milles on kaks korda esitatud vana listi teine rida, kaks korda esimene rida.

```
In[14]:= t[{2, 2, 1, 1}, {2, 3}]
```

```
Out[14]= {{e, f}, {e, f}, {b, c}, {b, c}}
```

Eelnevast näitest võeti kõigist ridadest vaid 2. ja 3. element.

```
In[15]:= t[All, 1]
```

```
Out[15]= {a, d}
```

Leidsime listi (maatriksi) esimese veeru.

```
In[16]:= t[1]
```

```
Out[16]= {a, b, c}
```

Selliselt leidi maatriksi esimene rida.

```
In[17]:= si1 = {{a1, a2}, {a3, a4}};
          si2 = {{b1, b2}, {b3, b4}};
          si3 = {{c1, c2}, {c3, c4}};
          si4 = {{d1, d2}, {d3, d4}};
```

```
In[21]:= s = {{si1, si2}, {si3, si4}}
```

```
Out[21]= {{{a1, a2}, {a3, a4}}, {{b1, b2}, {b3, b4}}},
          {{{c1, c2}, {c3, c4}}, {{d1, d2}, {d3, d4}}}}
```

```
In[22]:= MatrixForm[s]
```

```
Out[22]//MatrixForm=
```

$$\left(\begin{array}{cc} \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} & \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \\ \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} & \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix} \end{array} \right)$$

```
In[23]:= Part[s, 2]
```

```
Out[23]= {{{c1, c2}, {c3, c4}}, {{d1, d2}, {d3, d4}}}
```

```
In[24]:= MatrixForm[%]
```

```
Out[24]//MatrixForm=
```

$$\left(\begin{array}{cc} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} & \begin{pmatrix} c_3 \\ c_4 \end{pmatrix} \\ \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} & \begin{pmatrix} d_3 \\ d_4 \end{pmatrix} \end{array} \right)$$

```
In[25]:= Part[s, 2, 1]
```

```
Out[25]= {{c1, c2}, {c3, c4}}
```

```
In[26]:= Part[s, 2, 1, 1]
```

```
Out[26]= {c1, c2}
```

```
In[27]:= Part[s, 2, 2, 1, 1]
```

```
Out[27]= d1
```

```
In[28]:= Part[s, 2, All, 2]
```

```
Out[28]= {{c3, c4}, {d3, d4}}
```

```
In[29]:= Part[s, All, 2]
```

```
Out[29]= {{{b1, b2}, {b3, b4}}, {{d1, d2}, {d3, d4}}}
```

```
In[30]:= MatrixForm[%]
```

```
Out[30]//MatrixForm=
```

$$\begin{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} & \begin{pmatrix} b_3 \\ b_4 \end{pmatrix} \\ \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} & \begin{pmatrix} d_3 \\ d_4 \end{pmatrix} \end{pmatrix}$$

6.4. Listi ja maatriksi elementide otsimine, testimine, asendamine, kustutamine.

Position[list, form]	positsioonid, milles form listis esineb
Count[list, form]	mitu korda form listis esineb
MemberQ[list, form]	testi, kas form on üks listi elementidest
FreeQ[list, form]	testi, kas form ei ole üks listi elementidest (ei leidu mitte kusagil)

Nagu kusagil üleval juba mainitud, kehtivad paljud listide kohta käivad protseduurid ka algebraliste avaldiste kohta või muude *Mathematica* konstruktsioonide kohta.

```
In[31]:= $Line = 0;
```

```
In[1]:= Clear[s, t]
```

In[2]:= **t = {a, b, c, d, a, d, f, e, e, a}**

Out[2]= {a, b, c, d, a, d, f, e, e, a}

In[3]:= **Count[t, a]**

Out[3]= 3

In[4]:= **Position[t, a]**

Out[4]= {{1}, {5}, {10}}

In[5]:= **MemberQ[t, f]**

Out[5]= True

In[6]:= **FreeQ[t, 0]**

Out[6]= True

In[7]:= **u = 2 + b + v + m + j * $\frac{2}{w}$**

Out[7]= $2 + b + m + v + \frac{8}{w}$

In[8]:= **MemberQ[u, v]**

Out[8]= True

In[9]:= **Take[u, {2, 3}]**

Out[9]= b + m

In[10]:= **Take[u, {3, 5}]**

Out[10]= $m + v + \frac{8}{w}$

In[11]:= **u[[5]]**

Out[11]= $\frac{8}{w}$

```
In[12]:= u[[5, 1]]
```

```
Out[12]= 8
```

Ülaltoodud näited algebralistest avaldistest mõnede liikmete eraldamise kohta on üsnagi lihtsad. Kui avaldis (näiteks mõne arvutuse tulemus) on mitmeid lehekülgi pikk, ei või väga kindel olla, et programm eraldab sealt tõesti just sellise osa, mida eeldaks. Peale selle - ülaltoodud u avaldises paistab ka välja, et kusagil on j -le omistatud väärtus (For-lauses vist) - veel üks probleem, mis võib tekkida pika *Mathematica* dokumendi loomisel, arvutuste tegemisel.

Prepend[list,element]	lisab elemendi listi algusesse
Append[list,element]	lisab elemendi listi lõppu
Insert[list, element, i]	lisab elemendi listi i -ndale positsioonile
Insert[list, element, -i]	lisab elemendi listi lõpust lugedes i -ndale positsioonile
Insert[list,element,{i,j,...}]	lisab elemendi listi mitmetesse kohtadesse
Delete[list,i]	kustutab listi i -nda elemendi
Delete[list, {{i ₁ , j ₁ }, {i ₂ , j ₂ },...}]	kustutab mitmeid listi elemente
ReplacePart[list,uus,i]	asendab listis i -ndal kohal oleva elemendi <i>uue</i> -ga
ReplacePart[list,uus,-i]	asendab listis tagantpoolt i -ndal kohal oleva elemendi <i>uue</i> -ga
ReplacePart[list,uus,{i,j,...}]	asendab listis i -ndal, j -ndal ... kohal olevad elemendid <i>uue</i> -ga

```
In[13]:= Prepend[{a, b, c}, x]
```

```
Out[13]= {x, a, b, c}
```

```
In[14]:= Append[%, y]
```

```
Out[14]= {x, a, b, c, y}
```

```
In[15]:= Insert[%, h, 3]
```

```
Out[15]= {x, a, h, b, c, y}
```

```
In[16]:= Insert[%, hg, -3]
```

```
Out[16]= {x, a, h, b, hg, c, y}
```

```
In[17]:= Delete[%30, {{3}, {2}}]
```

```
- Delete::partw :  
Part {3} of {{{b1, b2}, {b3, b4}}, {{d1, d2}, {d3, d4}}} does not exist.
```

```
Out[17]= Delete[{{{b1, b2}, {b3, b4}}, {{d1, d2}, {d3, d4}}}, {{3}, {2}}]
```

```
In[18]:= uus = {x, a, h, b, hg, c, y};
```

```
In[19]:= ReplacePart[uus, 2 x, {{2}, {4}}]
```

```
Out[19]= {x, 2 x, h, 2 x, hg, c, y}
```

```
In[20]:= m = {{1, 0, 0}, {0, 0, 2}, {3, 2, 1}}
```

```
Out[20]= {{1, 0, 0}, {0, 0, 2}, {3, 2, 1}}
```

```
In[21]:= m = ReplacePart[m, {r, q}, {2, 2}]
```

```
Out[21]= {{1, 0, 0}, {0, {r, q}, 2}, {3, 2, 1}}
```

```
In[22]:= MatrixForm[m]
```

```
Out[22]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \{r, q\} & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

Join[list ₁ , list ₂ , ...]	listide üheks liitmine
Union[list ₁ , list ₂ , ...]	listide liitmine, kusjuures korduvad elemendid eemaldatakse ja järgi jäänud eemaldatakse
Sort[list]	listi elementide sorteerimine kasvavas järjekorras
Reverse[list]	listi elementide järjekorra pööramine
RotateLeft[list]	listi liikmete asukoha nihutamine ühe koha võrra vasakule
RotateLeft[list,n]	listi liikmete asukoha nihutamine n kohta vasakule
RotateRight[list]	listi liikmete asukoha nihutamine ühe koha võrra paremale
RotateRight[list,n]	listi liikmete asukoha nihutamine n kohta paremale
Partition[list,n]	listi lõhkumine n-elementilisteks tükkideks
Partition[list, n, d]	listi lõhkumine n-elementilisteks tükkideks sammuga d
Apply[Plus, list]	listi kõigi liikmete summa
Apply[Times, list]	listi kõigi liikmete korrutamine
Flatten[list]	listi silumine üheks 'vektori'
Flatten[list,n]	ülemisel n astmel olevate listi liikmete silumine üheks
FlattenAt[list,i]	listi i-ndal positsioonil oleva alamlisti silumine üheks

```
In[23]:= Join[m, uus]
```

```
Out[23]= {{1, 0, 0}, {0, {r, q}, 2}, {3, 2, 1}, x, a, h, b, hg, c, y}
```

```
In[24]:= Flatten[%]
```

```
Out[24]= {1, 0, 0, 0, r, q, 2, 3, 2, 1, x, a, h, b, hg, c, y}
```