

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:

2 + 4

6

2. Aritmeetika, algebra.

2.1. Aritmeetika. Täpsed ja ligikaudsed tehted.

Tavalistes arvutustes kasutatavad tehted:

$x + y + z$	liitmine
$-x$	lahutamine, miinus, neg. arv
$x y z$ või $x * y * z$	korrutamine, korrusmärg või tühik selle asemel
x / y	jagamine
x^y ehk x^y	astendamise

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

Näited:

2.3 + 5.6 * 2

13.5

2 4 6

48

2.4³²

1.46811 × 10¹²

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[ ]`:

3³²

1853020188851841

3³² // N

1.85302 × 10¹⁵

N[3³²]

1.85302 × 10¹⁵

N[2⁴⁰]

1.09951 × 10¹²

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>	trigonomeetrilised funktsioonid, argument on radiaanides!
<code>ArcSin[x], ArcCos[x], ArcTan[x], ArcCot[x]</code>	trigonomeetrilised 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 $\frac{n}{m}$

Quotient[n,m]	jagatise $\frac{n}{m}$ täisosa
Random[]	pseudojuhuslik arv 0 ja 1 vahel
Max[x,y,...], Min[x,y,...]	arvude (avaldiste) x,y,... maksimum, miinimum
FactorInteger[n]	täisarvu n lahutamine algarvude korrutiseks
avaldis // N ehk N[avaldis]	avaldisel ligikaudne väärtus; esitamine kümne astmete kaudu
N[avaldis,n]	avaldisel 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.

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

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
avaldis /. x-> väärtus	annab avaldises x-le väärtuse
avaldis /. {x -> x _{väärt} , y -> y _{väärt} }	avaldises antakse väärtused nii x-le kui ka y-le

Defineerime avaldises s järgmiselt:

$$s = x^2 + 4.5 x - y$$

$$4.5 x + x^2 - y$$

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

s /. x → 4

34. - y

s /. {x → 2.3, y → 3}

12.64

s /. {x → y, y → t}

- t + 4.5 y + y²

s /. {x → x², y → y³}

4.5 x² + x⁴ - y³

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.

a = 3. ;

b = 2.5;

c = x - y;

a² + b³ - $\frac{c}{2}$.

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.

%	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:

3. Sümbolarvutus. Tuletised, integraalid. Funktsioonid.

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

Clear[f]; \$Line = 0;

$D[f, x]$ või $\partial_x f$	(osa)tuletis funktsioonist f x järgi: $\frac{df}{dx}$ ehk $\frac{\partial f}{\partial x}$
$D[f, x, y, \dots]$ või $\partial_{x, y, z, \dots} f$	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: $\frac{d^n f}{dx^n}$
$\text{Limit}[f, x \rightarrow x_0]$	piirväärtus $\lim_{x \rightarrow x_0} f$
$Dt[f]$	täisdiferentsiaal
$Dt[f, x]$	täistuletisavaldisest f muutuja x järgi
$\int f dx$ või $\text{Integrate}[f, x]$	määramata integraal
$\int_a^b f(x) dx$ või $\text{Integrate}[f, \{x, a, b\}]$	määratud integraal
$\int \int f(x, y) dx dy$ või $\text{Integrate}[f, x, y]$	mitmekordne määramata integraal muutujate x ja y järgi
$\int_a^b \int_c^d f(x, y) dx dy$ või $\text{Integrate}[f, \{x, a, b\}, \{y, c, d\}]$	mitmekordne määratud integraal
$\text{Series}[f, \{x, x_0, \text{järk}\}]$	Taylori astmerida funktsioonist (avaldisest) f punkti $x = x_0$ juures
$\text{Normal}[\text{rida}]$	juhul, kui <i>rida</i> 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 $\text{Product}[f, \{i, i_{\min}, i_{\max}\}]$	korruis

Tuletised, diferentsiaalid

$$\partial_x x^3$$

$$3x^2$$

$$D[x^n, x]$$

$$n x^{-1+n}$$

$$D[x^n, n]$$

$$x^n \text{Log}[x]$$

$$\partial_x x^2 \text{Log}[x + a]$$

$$2 x \text{Log}[a + x]$$

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.

$$\partial_{x,y} (3 x^2 y^5)$$

$$30 x y^4$$

$$\partial_{x,y} 3 x^2 y^5$$

$$0$$

Integreerimine.

$$\int \frac{3 y}{x^2 + y^2} dx$$

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

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.

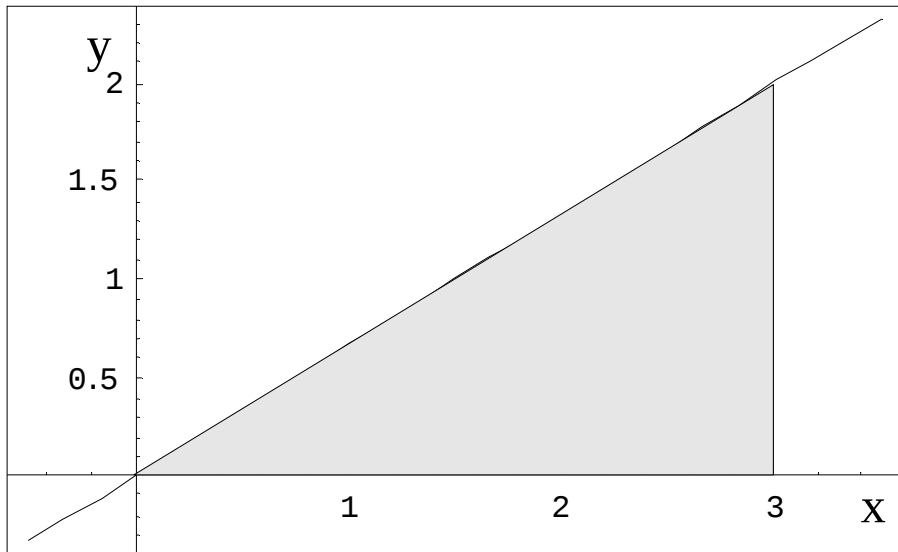
$$\int_0^5 2 x^3 dx$$

$$\frac{625}{2}$$

$$\text{Integrate}\left[\frac{2y}{x^2 + y^2}, x, y\right]$$

$$-2x + 2y \operatorname{ArcTan}\left[\frac{x}{y}\right] + x \operatorname{Log}[x^2 + y^2]$$

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:



$$\int_0^3 \int_0^{\frac{2x}{3}} (x^2 + y^2) \, dy \, dx$$

$$\frac{31}{2}$$

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

$$s = \frac{\sin[x]}{x}$$

$$\frac{\sin[x]}{x}$$

$$\text{Limit}[s, x \rightarrow 0]$$

$$1$$

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

$$t = \frac{(9 - x^2) 2 x^3}{(3 x - 9)}$$

$$\frac{2 x^3 (9 - x^2)}{-9 + 3 x}$$

Limit[t, x → 3]

-108

Limit[e^{-x} - 1, x → ∞]

-1

Limit[Tan[x], x → $\frac{\pi}{2}$]

-∞

3.2. Funktsioonide defineerimine.

Clear[s, f];
\$Line = 0;

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

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:

f[y]

f[2.5]

g[x] /. x → y

1 + 2 y + 3 y²

g[x] /. x → 3

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:

```
Clear[g]
```

```
g := 3 x^2 + 2 x + 1
```

```
g /. x -> 4
```

```
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!

```
Clear[s]
```

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

```
s[4, 0.5, 1]
```

```
s[4, 1, 0.5]
```

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

$x = y$

y

$x == y$

True

$2 + 4 == 6$

True

$2 * 5 == 11$

False

$x = .$

Viimase võrdusega tühistasime x väärtuse.

$x == 5$

$x == 5$

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.

$eq = x^2 - 4 x - 7 == 0$

$-7 - 4 x + x^2 == 0$

$x = y$	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)
<code>x /. lahend</code>	reeglite listi kasutatakse x väärtuste saamiseks
<code>avaldis /. lahend</code>	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.

`Solve[eq, x]`

$\{\{x \rightarrow 2 - \sqrt{11}\}, \{x \rightarrow 2 + \sqrt{11}\}\}$

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:

`N[%]`

$\{\{x \rightarrow -1.31662\}, \{x \rightarrow 5.31662\}\}$

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

`lahendid = x /. %28`

$\{-1.31662, 5.31662\}$

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

```
x0 = lahendid[ [1] ]
```

```
-1.31662
```

```
x1 = lahendid[ [2] ]
```

```
5.31662
```

```
 $\sqrt{x}$  /. %28
```

```
{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.

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

```
{ {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:

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

```
{ {x → -1.14978}, {x → -0.0861964 - 1.20257 i},  
  {x → -0.0861964 + 1.20257 i}, {x → 1.35864}, {x → 3.96353} }
```

```
Solve[Sin[x] == a, x]
```

```
{ {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.

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

```
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.

<code>Solve[{vp₁ == pp₁, vp₂ == pp₂, ...}, {x₁, x₂, ...}]</code>	võrrandisüsteemi lahendamine x ₁ , x ₂ , ..suhtes
<code>NSolve[{vp₁ == pp₁, vp₂ == pp₂, ...}, {x₁, x₂, ...}]</code>	võrrandisüsteemi numbriline lahendamine x ₁ , x ₂ ..suhtes
<code>Eliminate[{vp₁ == pp₁, vp₂ == pp₂, ...}, {x₁, x₂, ...}]</code>	võrrandisüsteemist muutujate x ₁ , x ₂ elimineerimine
<code>Reduce[{vp₁ == pp₁, vp₂ == pp₂, ...}, {x₁, x₂, ...}]</code>	annab lihtsustatud võrrandite süsteemi koos võimalike lahendustega
<code>FindRoot[vp == pp, {x, x₀}]</code>	leiab numbriliselt lahendi võrrandile, alustades x ₀ -st

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

{ {x → - $\frac{117}{19}$, y → - $\frac{45}{19}$, z → $\frac{18}{19}$ } }

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

{ {x → -6.15789, y → -2.36842, z → 0.947368} }

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

-6 + 5 y == 5 x

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

x == $\frac{1}{10}$ (-8 - 5 z) && y == $\frac{1}{10}$ (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.

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

FindRoot[g[x] == 0, {x, 0}]

{x → 0.}

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

```
{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

```
$Line = 0;
```

```
Plot[E-0.5 x Sin[x], {x, 0, 2 π}]
```

```
Plot[{2 x2 - 4 x, 2 x}, {x, -2, 4}, AxesLabel → {x, y}]
```

```
Show[%1, %2]
```

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.

```
Plot[{Ex, 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]
```

Nagu näha, andis programm ka teada, et funktsiooni ln(x) (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.

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

```
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}]}]
```

- Graphics -

valik	vaikväärtus	
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:

4.2. Joonised ruumis.

`Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}]` teeb kolmemõõtmelise joonise kui funktsiooni kahest muutujast x ja y

```
Plot3D[x2 + 2 y2, {x, -2, 2}, {y, -2, 2}]
```

```
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)"]
```

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:

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

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

Selle tähistusviisiga analoogiline on järgmine:

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

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

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

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

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

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

`lahend[[1]][[1]][[2]]`

$\frac{e^{-a} (-e^a + 3 e^{a x})}{a}$

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

- 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:

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

```
Plot[u[x] /. a → 5, {x, -2, 1}, PlotRange → All]
```

- Graphics -

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

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

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

```
Plot[lahendus[[1]][[1]][[2]],
  {x, 0, 70}, PlotPoints → 100, PlotRange → All]
```

- Graphics -

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

```
lahend = Chop[%20]
```

```
{ {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

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

- Graphics -

```
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.

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

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

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

```
- Graphics -
```

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

```
Evaluate[u[3] /. funkt[[1]][[1]]]
```

```
1.55248
```

```
2 * Sin[4] * Evaluate[u[4] /. funkt[[1]][[1]]]
```

```
-1.88302
```

Võime lahendada ka mitmest diferentsiaalvõrrandist koosneva süsteemi.

```
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}]
```

```
{{x → InterpolatingFunction[{{0., 10.}}, <>],  
y → InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
sol[[1]][[1]]
```

```
x → InterpolatingFunction[{{0., 10.}}, <>]
```

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

```
- Graphics -
```

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

- Graphics -

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