

## Mathematica-tehtäviä / vihjeet (versio 1.0, 09.08.2002)

### 1.

Kertomerkki on joko välilyönti tai `*`; numeroiden tapauksessa `*` on selkeämpi (ihmiselle).

### 2.

Kokeile funktioita `Simplify` ja `FullSimplify`.

### 3.

Tarvittavia funktioita: `Sum`, `FactorInteger`, `PrimeQ`. Taulukoita voi tehdä funktiolla `Table`. Liittämällä loppuun määre `//TableForm` taulukko saadaan tulostetuksi havainnollisempaan muotoon.

### 4.

Tarvittava funktio on `FunctionExpand`. Muista iso alkukirjain ja hakasulut: `Sin[Pi/n]`.

### 5.

Taulukoita voi tehdä funktiolla `Table`. Liittämällä loppuun määre `//TableForm` taulukko saadaan tulostetuksi havainnollisempaan muotoon. Trigonometrinen funktioiden argumenttien tulee olla radiaaneissa. Luku  $\pi$  kirjoitetaan `Pi`; se voidaan myös valita valikosta. Muista iso alkukirjain ja hakasulut: `Sin[Pi/4]`, `Cos[0]`.

### 6.

Likiarvot halutulla tarkkuudella saadaan funktiolla `N`. Neperin luku on `E`; se voidaan valita myös valikosta, jolloin symboli näyttää hieman erikoiselta pikku `e`:ltä. Aloita varovaisesti; älä syötä kovin suuria lukuja  $n$ . Tulokset voi kerätä taulukoksi `Table`-funktiolla.

### 7.

Likiarvot halutulla tarkkuudella saadaan funktiolla `N`. Neperin luku on `E`; se voidaan valita myös valikosta, jolloin symboli näyttää hieman erikoiselta pikku `e`:ltä. Kertoma ilmaistaan yksinkertaisesti huutomerkillä. Tulokset voi kerätä taulukoksi `Table`-funktiolla.

### 8.

Kirjoita kaava muuttujien avulla ja sijoita näille arvot korvausoperaattoria `/.` (eli `ReplaceAll`) ja sopivaa korvaussääntöä käyttäen. Arkuskosini on `ArcCos`.

### 9.

Sopivia Mathematican funktioita ovat `Simplify` ja `FullSimplify`.

### 10.

Tarvittavat Mathematican funktiot ovat `Expand` ja `Factor`. Näiden argumenttina oleva lauseke voi olla joko hakasuluissa tai komento voidaan kirjoittaa sen perään: `Expand[lauseke]` tai `lauseke//Expand`.

## 11.

Tarvittavat Mathematican funktiot ovat `Expand` ja `Factor`.

## 12.

Tekijöihin jako: `Factor`. Supistaminen: `Cancel`. Kokeile luvulle  $n$  erikseen eri arvoja. (Miksi yleistä symbolia  $n$  ei voida käyttää?) Kokeilut voidaan laskea myös yhteen taulukkoon käyttämällä komentoa `Table`; tulostuksen saa hieman selkeämpään muotoon kirjoittamalla komennon jälkeen `//TableForm`.

## 13.

Murtolausekkeesta voi poimia osoittajan funktiolla `Numerator` ja nimittäjän lausekkeella `Denominator`. Kokeile myös, mitä tapahtuu, kun murtolausekkeeseen kohdistetaan peräkkäin funktiot `Apart` ja `Simplify`. Mitä nämä periaatteissa tekevät?

## 14.

Sopivia funktioita: `Apart`, `Together`, `Simplify`, `Expand`...

## 15.

Sopivia funktioita: `Apart`, `Numerator`, `Denominator`. Myös hiirtä ja valikkoja voi hyödyntää.

## 16.

Trigonometrinen funktioiden argumentit ilmoitetaan Mathematicassa radiaaneissa. Radiaanien ja asteiden välinen muunnoskerroin on valmiina nimellä `Degree` (katso dokumentaatiota), mutta kertoimen voi tietenkin muodostaa itsekin. Muista: Funktioiden nimet kirjoitetaan isolla alkukirjaimella, argumentit annetaan hakasuluissa.

## 17.

Mathematicalla on useita erilaisia `Expand`- ja `Simplify`-tyyppisiä komentoja. Muista: Funktioiden nimet kirjoitetaan isolla alkukirjaimella, argumentit annetaan hakasuluissa.

## 18.

Käytä funktioita `TrigFactor`, `TrigExpand`, `TrigReduce`, `ExpToTrig`, `TrigToExp`. Muista: Funktioiden nimet kirjoitetaan isolla alkukirjaimella, argumentit annetaan hakasuluissa.

## 19.

Anna symbolille  $n$  numeerisia arvoja. Funktioiden nimet ovat `Cos` ja `ArcCos`.

## 20.

Polynomi voidaan nollakohtien avulla kirjoittaa tulomuotoon

$$(x - x_1)(x - x_2) \dots (x - x_n).$$

Hyödyllisiä funktioita ovat mm. `Product`, `Collect`, `Coefficient`, `CoefficientList`. Jos listassa on pitkiä alkioita, sen voi tulostaa selkeämpään muotoon kirjoittamalla perään `//TableForm`.

## 21.

Talleta ensin yhtälö jollekin nimelle. Yhtälöissä käytetään yhtäläisyysmerkinä `==`. Yhtälön ratkaiseminen `Solve`-funktiolla tuottaa ratkaisut ns. korvaussääntöjen muodossa. Näiden avulla voidaan saadut juuret helposti sijoittaa mihin tahansa lausekkeeseen, esimerkiksi yhtälöön: `yhtalo/.korvaus`. Tässä `/.` on lyhennemerkintä Mathematican funktiolle `ReplaceAll`.

## 22.

Likiarvot voidaan laskea joko tarkkojen arvojen likiarvoina funktiolla `N` tai suoraan käyttämällä funktiota `NSolve`. Juurten sijoittaminen yhtälöön: `yhtalo/.korvaus`, missä `/.` on lyhennemerkintä Mathematican funktiolle `ReplaceAll`.

## 23.

Mathematican versiosta riippuen juurten tarkat arvot saatetaan esittää hieman erikoisessa muodossa `Root`-funktion avulla. Tämä itse asiassa ilmoittaa vain, että kyseessä on juuri tämän polynomin juuri!

## 24.

Suorita laskut siten, että et joudu käsin syöttämään uudelleen jo laskettuja tuloksia. Käytä korvausoperaattoria `ReplaceAll` eli `/.` sopivalla tavalla. Esimerkkiyhtälön juuret ovat kompleksilukuja, mutta juurista muodostettu lauseke on yksinkertainen reaalinen murtoluku.

## 25.

Yleiset ratkaisukaavat saadaan ratkaisemalla `Solve`-funktiolla yhtälöt, joissa kertoimet ovat symboleja.

## 26.

Kolmannen asteen yhtälön yleinen ratkaisukaava on sangen mutkikas eikä kaikissa tapauksissa anna ongelmitta oikeaa ratkaisua. Ratkaistaessa yhtälö suoraan saatetaan tarvita funktiota `ComplexExpand` saatujen juurten sieventämiseen.

## 27.

`Solve`-funktion antamasta korvauslistasta voidaan poimia  $k$ :s korvaussääntö joko hiiren avulla tai asettamalla listan nimen perään indeksi kaksinkertaisiin hakasulkuihin, esimerkiksi `ratk[[k]]`. Tulos voidaan saada myös `Factor`-funktiolla, mutta siinä tarvitaan lisäoptio; ks. Mathematican dokumentaatiota.

## 28.

Sievennä juurten lausekkeet `ComplexExpand`-funktiolla. Kompleksiluvun itseisarvo saadaan funktiolla `Abs`.

## 29.

`Solve`-funktion ensimmäisenä argumenttina voi olla usean yhtälön muodostama lista ja toisena usean tuntemattoman muodostama lista.

## 30.

Ryhmän rakennetta voi myös tutkia tarkemmin eliminoimalla esimerkiksi tuntemattoman  $x$  kahdesta ensimmäisestä ja vastaavasti kahdesta jälkimmäisestä yhtälöstä. Tällöin on apua Mathematican funktiosta `Eliminate`.

### 31.

Ratkaise ensin yhtälöpari ja poimi hiirellä ratkaisusta ehto luvulle  $a$ . Epäyhtälöitä voidaan ratkaista myös funktiolla `InequalitySolve`, joka on ensin ladattava: `Needs["Algebra`InequalitySolve`"]`. Huomaa 'väärinpäin' olevat lainausmerkit!

### 32.

Kumpi on järkevämpää: Hakea leikkauspisteiden koordinaateille tarkat arvot vai likiarvot? Kuvaajat voidaan piirtää funktiolla `ImplicitPlot`. Tämä on ladattava ensin: `Needs["Graphics`ImplicitPlot`"]`. Huomaa 'väärinpäin' olevat lainausmerkit!

### 33.

Funktiota `Solve` tai edes `NSolve` ei voida käyttää, koska kyseessä on transkendenttiyhtälö. Iteratiiviseen numeeriseen ratkaisemiseen (Newtonin menetelmän tapaan) on käytettävissä funktio `FindRoot`.

### 34.

Funktioita `Solve` ja `NSolve` ei voida käyttää, koska ne ratkaisevat vain algebrallisia yhtälöitä. Transkendenttinen yhtälö tai yhtälöryhmä voidaan ratkaista funktiolla `FindRoot`, joka käyttää yksi- tai useampiulotteista Newtonin menetelmää. Sopivien alkuarvojen löytämiseksi käyrät on syytä piirtää. Tällöin voidaan käyttää funktiota `ImplicitPlot`, joka ei kuitenkaan ole Mathematican perusfunktioita ja on sen johdosta erikseen ladattava: `Needs["Graphics`ImplicitPlot`"]`. Huomaa 'väärinpäin' olevat lainausmerkit!

### 35.

Ratkaise kolmen algebrallisen yhtälön ryhmä. Muodosta tätä varten ensin ympyrän yhtälö, jossa säde ja keskipisteen koordinaatit ovat tuntemattomia, ja sijoita siihen annetut arvot.

### 36.

Käytä funktiota `Solve` usean algebrallisen yhtälön muodostaman ryhmän ratkaisemiseen.

### 37.

Muodosta polynomille lauseke tuntemattomin kertoimin ja johda kertoimille yhtälöryhmä ehtojen perusteella.

### 38.

Tarkastele riittävän pitkää väliä. Muista merkinnät: `Sin[8 x]` etc.

### 39.

Funktioiden nimet ovat `Cos` ja `ArcCos`. Yhden kuvaajan piirtäminen tapahtuu komennolla `Plot`. Useita kuvioita voidaan yhdistää samaan kuvaan komennolla `Show`, jonka argumenteiksi kirjoitetaan ne tulosteet, jotka kuvaan halutaan. Esimerkiksi `%k` viittaa tulosteeseen `Out[k]`; tulosteille voidaan antaa myös nimet kirjoittamalla esimerkiksi `kuva1 = Plot ...`.

### 40.

Tarvittava funktio on `Plot3D`. Tutki piirtoalueen vaikutusta kuvaajan muotoon ja miten siihen voidaan vaikuttaa `Plot3D`-funktion optioilla (katso dokumentaatiota tai kirjoita `??Plot3D`).

## 41.

Funktionnimi on `ArcTan`. Tarvittava piirtokomento on `Plot3D`. Piirtotiheyttä voi säätää optiolla `PlotPoints`. Muitakin optioita on; katso dokumentaatiota tai kokeile `??Plot3D`.

## 42.

Mieti ensin, millä arvoilla  $(x, y)$  funktio on määritelty.

## 43.

Säädä piirtotiheys sopivaksi, niin että saat kauniin kuvan: `PlotPoints -> {n1, n2}` asettaa eri muuttujien suunnissa eri määrät jakopisteitä. Onko funktio jatkuva origossa?

## 44.

Piirtokomento on `ParametricPlot`. Piirtotiheyttä voidaan säätää optiolla `PlotPoints`.

## 45.

Piirtokomento on `ParametricPlot3D`. Piirtotiheyttä voidaan säätää optiolla `PlotPoints`.

## 46.

Piirtokomento on `ParametricPlot3D`. Piirtotiheyttä voidaan säätää optiolla `PlotPoints`. Pallopinta: Jos korkeuskoordinaatti on  $t$ , niin mikä on pisteen etäisyys  $z$ -akselista?

## 47.

Valitse parametriksi  $y$ .

## 48.

Piirtokomento on `ParametricPlot3D`. Säädä piirtotiheys sopivaksi, niin että saat kauniin kuvan: `PlotPoints -> {n1, n2}` asettaa eri parametrien suunnissa eri määrät jakopisteitä.

## 49.

Käytä funktiota `ImplicitPlot`. Tämä on erikseen ladattava: `Needs["Graphics`ImplicitPlot`"]`.

## 50.

Logaritmifunktio on `Log`. Käytä funktiota `ImplicitPlot`. Tämä on erikseen ladattava: `Needs["Graphics`ImplicitPlot`"]`. Katso dokumentaatiota. Mieti, mikä on sopiva piirtoalue.

## 51.

Tarvittavia funktioita: `Plot3D`, `ContourPlot`. Jälkimmäiselle voidaan antaa optiona korkeuskäyrien korkeudet muodossa `Contours->{...}`, missä korkeudet luetellaan listassa; katso myös dokumentaatiota.

## 52.

Käytä funktiota `ContourPlot3D`. Tämä on erikseen ladattava: `Needs["Graphics`ContourPlot3D`"]`. Katso dokumentaatiosta käyttöohjeet. Optiolle `PlotPoints` on hyvä antaa hieman oletusta suurempi arvo, esimerkiksi `{5, 5}`; liian suuri hidastaa laskentaa melkoisesti.

### 53.

Funktio voidaan määritellä kahdella tavalla: antamalla muotoa  $f[x_] := \dots$  oleva määrittely tai käyttämällä funktiota `Function`. Kokeile molempia. Taulukko muodostetaan funktiolla `Table`, derivaatta saadaan esimerkiksi kirjoittamalla  $f'[x]$ . Integroinnissa tarvittava äärettömyys on `Infinity`; se voidaan myös valita valikosta.

### 54.

Funktio voidaan määritellä kahdella tavalla: antamalla muotoa  $f[x_] := \dots$  oleva määrittely tai käyttämällä funktiota `Function`. Edellistä tapaa käytettäessä voidaan lisäksi määritellä erikseen arvo yksittäisissä pisteissä:  $f[0] = \dots$ .

### 55.

Huomaa: Hyperbolisen kosinin käänteisfunktio (päähaara) `arcosh` on Mathematicassa (virheellisesti) `ArcCosh`. Funktion paloittaisessa määrittelyssä käytetään symbolia `/;` rajoittamaan määrittelyaluetta, esimerkiksi  $f[x_ / ; x > -1 \ \&\& \ x < 1] := \dots$ .

### 56.

Piirtämisen onnistuminen saattaa riippua Mathematican versiosta. Itseisarvofunktio (`Abs`) integraalin sisällä on ongelmallinen. Turvallisinta on määritellä funktio itseisarvolausekkeen merkkien mukaan kolmessa osassa. Tutki, osaako Mathematica derivoida paloittain määriteltyä funktiota!

### 57.

Ennen Mathematican funktion `a` määrittelyä anna komento `Remove[a]`, jolla poistetaan mahdolliset aiemmat määrittelyt. Katso myös, mitä komento `?a` määrittelyn jälkeen antaa. Kyseessä on rekursiivinen funktiomäärittely. Taulukko voidaan muodostaa funktiolla `Table`. Miksi taulukon laskeminen kestää?

### 58.

Funktion arvot voidaan laskea taulukkoon `Table`-funktiolla. Miten laskisit käsin arvon  $f(1)$ ?

### 59.

Surjektio  $A \rightarrow B$  on funktio, jossa jokainen maalijoukon  $B$  alkio on jonkin alkion kuva. Summa voidaan muodostaa funktiolla `Sum`, binomikerroin on `Binomial`. Taulukon (kaksinkertaisen listan) voi muodostaa `Table`-komennolla ja sen saa näkymään kaksiulotteisena kirjoittamalla perään `//TableForm`.

### 60.

Tutki erikseen, mitä `FactorInteger` antaa. Indeksimerkintä `[[1,1]]` poimii sen tulostuksesta osan.

### 61.

Huomaa merkintä: `Sin[x]^4` etc. Kahden lausekkeen samuutta voi tutkia tarkastelemalla niiden erotusta. Tämän vertaaminen nollaan on yleensä helpompaa kuin kahden monimutkaisen lausekkeen vertaaminen toisiinsa. Toisena vaihtoehtona on tarkastella lausekkeiden kuvaajia.

### 62.

Kokeile: `Integrate[x^n/.n->-1,x] == Integrate[x^n,x]/.n->-1`. Sijoita myös muuttujalle  $n$  jokin muu arvo. Miten selität tulokset?

## 63.

Funktio on derivoituva myös origossa, mutta derivaatta on laskettava erotusosamäärän raja-arvona (miksi?). Mathematicassa on valmiina funktio `Limit`.

## 64.

Poista ensin funktioille mahdollisesti aiemmin tehtyt määrittelyt: `Remove[f, g]`. Määäämätön funktio voidaan ottaa käyttöön yksinkertaisesti kirjoittamalla `f[x]`.

## 65.

Määäämätön funktio voidaan ottaa käyttöön kirjoittamalla `f[x, y]`. Jos olet aiemmin käyttänyt samaa symbolia jossakin muussa merkityksessä, hävitä se ensin: `Remove[f]`. Derivointioperaattori on `D` riippumatta siitä, lasketaanko tavallisia vai osittaisderivaattoja.

## 66.

Funktiolla `Integrate` lasketaan sekä integraalifunktioita että määrättyjä integraaleja. Äärettömyys on `Infinity`; se voidaan myös valita valikosta. Mathematica tuntee monia muitakin funktioita kuin ns. tavalliset alkeisfunktiot. Mitä nämä oikeastaan ovat, ei aina ilmene dokumentaatiosta, koska määritelmät eivät välttämättä ole yksinkertaisia.

## 67.

Symbolisissa ohjelmissa derivointi on yleensä integrointia luotettavampaa. Tarkistus voi siis tapahtua derivoimalla. Logaritmifunktio on `Log`; ks. dokumentaatiota.

## 68.

Symbolinen integrointi tapahtuu funktiolla `Integrate`, numeerinen funktiolla `NIntegrate`. Jälkimmäisessä sovelletaan suoraan jotakin numeerisen integroinnin menetelmää, jonka valintaan myös käyttäjä voi vaikuttaa. Ks. dokumentaatiota, erityisesti `Implementation Notes`.

## 69.

Komennolla `Integrate` lasketaan sekä integraalifunktio että määrätty integraali. Numeeriselle integroinnille (määrätyn integraalin laskemiseen) on komento `NIntegrate`. Korvausoperaattori on `ReplaceAll` eli `/.`.

## 70.

Talleta aluksi funktion lauseke jollakin nimellä, jotta siihen viittaaminen myöhemmin on helppoa. Tarvittavia funktioita: `D`; `Solve`. Arvojen sijoittaminen johonkin lausekkeeseen tapahtuu korvausoperaattorilla `/.` eli `ReplaceAll`. Mikäli saadut numeeriset lausekkeet näyttävät hankalilta, ne voi hahmottaa paremmin laskemalla likiarvot funktiolla `N`.

## 71.

Valitse parametrisoinnissa toiseksi parametriksi  $x$  ja toiseksi pyörähdyskulma. Piirtäminen funktiolla `ParametricPlot3D`. Muista: `Sin[x^2]` jne.

## 72.

Periaatteessa yksinkertainen integrointi, mutta tutki tarkoin, mitä funktiota on integroitava.

### 73.

Valitse kardioidin parametrisoinnissa napakulma  $\varphi$  parametriksi, lausu  $x$  ja  $y$  tämän avulla ja käytä funktiota `ParametricPlot`. Tomaattipinnan parametrisoinnissa ota parametreiksi napakulma  $\varphi$  ja pyörähdyskulma  $x$ -akselin ympäri. Piirtäminen tapahtuu funktiolla `ParametricPlot3D`. Tilavuus saadaan periaatteessa integraalista  $\int y^2 dx$ , johon on tehtävä sellaiset sijoitukset, että muuttujaksi saadaan napakulma  $\varphi$ .

### 74.

Kaarenpituusintegraali:  $\int ds = \int \sqrt{x'(\varphi)^2 + y'(\varphi)^2} d\varphi$ .

### 75.

Käyrä on luontevinta kirjoittaa vektoriksi  $\mathbf{r} = \{\text{Cos}[t]/t, \text{Sin}[t]/t, \text{ArcTan}[t]\}$  ja laskea kaarenpituus integraalista  $\int |\mathbf{r}'(t)| dt$ .

### 76.

Laskusta tulee selkeämpi, jos se lasketaan symboleja käyttäen ja vasta lopuksi asetetaan näille arvot. Arvojen sijoittamiseksi on luontevaa määritellä korvaussääntö  $\{r \rightarrow 6.6, s \rightarrow 11.0, h \rightarrow \text{Sqrt}[11.0^2 - 6.6^2]\}$ , jolloin arvot voidaan helposti sijoittaa mihin tahansa välitulokseenkin ja tämän jälkeen jatkaa laskua symboleilla.

### 77.

Lausu pyramidin tilavuus sopivan muuttujan avulla. Tätä varten tarvitaan sopivien kolmioiden yhdenmuotoisuutta. Hyödynnä Mathematican komentoja, jotta et joudu syöttämään käsin aiempien laskujen tuloksia!

### 78.

Määrittele alueen  $A_n$  pinta-ala funktioksi:  $a[n_] := \dots$  ja sievennä (`Simplify`, `FullSimplify`) peräkkäisten alojen suhteen lauseke. Funktiolla `Sum` voi laskea myös äärettömän monen termin summia.

### 79.

Esimerkiksi: Laske leikkauspiste, kun suoran kulman kärjen  $x$ -koordinaatti on  $t$  ja toisen kateetin kulmakerroin on  $k$ .

### 80.

Ratkaise nestepinnan korkeus nestetilavuuden funktiona ja valitse saaduista ratkaisuista oikea. Lauseke on hankala, mutta Mathematica pystyy kuitenkin käsittelemään sitä.

### 81.

Satunnaislukuja generoidaan funktiolla `Random`, jolle voidaan antaa argumentiksi mm. väli, jolta lukuja halutaan.

### 82.

Joko: Muodosta funktion  $y$  ja erikseen lasketun toisen derivaatan summa ja sievennä tämä. Tai: Määrittele  $y$  Mathematican funktioksi ja sievennä  $y''[x] + y[x]$ . Logaritmifunktio on `Log`. Itseisarvot voidaan jättää huomiotta, sillä Mathematica tuntee logaritmifunktion  $\ln x$  myös negatiivisilla argumenteilla  $x$ , jolloin se eroaa funktiosta  $\ln|x|$  vain kompleksisella vakiolla  $i\pi$ . Kokeile: `FullSimplify[ComplexExpand[Log[-x]], x > 0]`.

### 83.

Käytä komentoa `DSolve`, jonka avulla voidaan löytää sekä yleinen ratkaisu että yksittäisratkaisu.



## 84.

Käytä komentoa `DSolve`. Ratkaisu ei ole lausuttavissa tavallisten alkeisfunktioiden avulla, mutta kylläkin Mathematican tuntemien funktioiden avulla.

## 85.

Käytä funktioita `DSolve` ja `Solve`. Kumpikin antaa ratkaisun korvaussäännön muodossa. Ratkaisu voidaan tallentaa muuttujaksi jatkokäsittelyä varten korvausoperaattorilla, esimerkiksi `lukum = y[t]/.ratkaisu`.

## 86.

Käytä funktioita `DSolve` ja `Solve`. Kumpikin antaa ratkaisun korvaussäännön muodossa. Ratkaisu voidaan tallentaa muuttujaksi jatkokäsittelyä varten korvausoperaattorilla, esimerkiksi `lukum = y[t]/.ratkaisu`.

## 87.

Käytä funktiota `Series`. Sarjan jäännöstermi voidaan pudottaa pois komennolla `Normal`.

## 88.

Sarjan summa lasketaan funktiolla `Sum`. Suppenemisalueen tarkastelussa voidaan käyttää myös funktiota `InequalitySolve`, joka on ensin ladattava: `Needs["Algebra`InequalitySolve`"]`.

## 89.

Esitä vektorit kaksialkioisina listoina. Muodosta näiden avulla vektoriyhtälö ja ratkaise se `Solve`-funktiolla.

## 90.

Syötä aluksi pisteet kahden alkion listoina ja muodosta näiden avulla tarvittavat vektorit. Puolittajavektori saadaan helpoimmin kulman kylvien suuntaisten yksikkövektoreiden summana.

## 91.

Syötä pisteet kolmialkioisina listoina ja laske näillä. Normaali voidaan määrittää joko skalaaritulojen avulla tai vektorituloa käyttäen.

## 92.

Esitä tehtävän vektorit symbolisina kaksi- (tai kolmi-) komponenttisina listoina, so. Mathematican vektoreina. Funktioista `Simplify` ja `FullSimplify` on apua; katso dokumentaatiota.

## 93.

Kertaa käyräteoria jostakin oppikirjasta! Em. laskujen läpivieminen johtaa yleensä vaikeuksiin, koska kaarenpituusintegraalista tulee liian monimutkainen. Tehtävän käyrä on kuitenkin poikkeuksellinen: lasku onnistuu! Syötä käyrä Mathematican vektorina, so. kolmialkioisena listana ja laske vektorialgebraa käyttäen.

## 94.

Lataa lisäpaketti: `Needs["Statistics`NormalDistribution`"]`. Tutki tämän funktioita `NormalDistribution` ja `CDF` (= *cumulative distribution function* = kertymäfunktio), joiden avulla voidaan laskea mihin tahansa normaalijakaumaan liittyvät todennäköisyydet. Ks. ohjeita: `?NormalDistribution`, `?CDF`.