TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ANNIINA MYÖTYRI

TOISEN KERTALUVUN DIFFERENTIAALIYHTÄLÖT JA SÄHKÖISTEN JÄRJESTELMIEN VERTAILU NIIDEN AVULLA

Kandidaatintyö

Tarkastaja: Terhi Kaarakka
31.07.2018

TIIVISTELMÄ

ANNIINA MYÖTYRI: Toisen kertaluvun differentiaaliyhtälöt ja sähköisten järjestelmien vertailu niiden avulla
Tampereen teknillinen yliopisto
Kandidaatintyö, 36 sivua, 3 liitesivua
Heinäkuu 2018
Teknis-luonnontieteellinen koulutusohjelma
Pääaine: Matematiikka
Tarkastajat: Lehtori Terhi Kaarakka
Avainsanat: Toisen kertaluvun differentiaaliyhtälö, Abitti, MATLABin Live Editor, STACK

Työssä tarkastellaan toisen kertaluvun differentiaaliyhtälöitä ja vertaillaan niihin liittyvien esimerkkitehtävien avulla kolmea sähköistä järjestelmää; Abittia, MATLABin Live Editoria ja STACK-järjestelmää. Abitti on lukion sähköinen kurssikoejärjestelmä, kun taas Live Editor ja STACK-järjestelmä ovat yliopistossa käytettäviä sähköisiä järjestelmiä. Työssä tutkitaan, millaista koodikielten osaamista tehtävänannon tekeminen vaatii sen laatijalta ja ratkaisijalta sekä sitä, millä tavoin tehtävien arviointi onnistuu eri järjestelmissä. Lisäksi pohditaan Abitin soveltuvuutta yliopistomatematiikan laskuharjoitustehtävien ratkaisemiseen.

Työn alussa käydään läpi toisen kertaluvun differentiaaliyhtälöihin liittyvää yleistä teoriaa, jonka jälkeen syvennytään tarkastelemaan yhden epälineaarisen toisen kertaluvun differentiaaliyhtälön, Van der Polin yhtälön, ratkaisemista linearisaation ja keskiarvottamisteorian avulla. Tämän jälkeen esitellään sähköisten järjestelmien vertailuun käytettävät esimerkkitehtävät ja vertaillaan tehtävien tekemistä eri järjestelmissä. Lopuksi kootaan yhteen sähköisten järjestelmien vertailun tulokset.

Vertailun avulla havaittiin, että Abitti ja MATLABin Live Editor ovat hyvin samankaltaisia järjestelmiä. Abitissa ratkaisemiseen käytettiin kahta kaavaeditoria, vaikka Abitissa olisi järkevin ratkaista tehtäviä, joissa käytetään vain vastauslomakkeen kaavaeditoria. Siinä on kuitenkin melko suppeat kaava- ja merkkivalikoimat, joten sen soveltuvuus yliopistomatematiikan laskuharjoitustehtävien ratkaisemiseen on melko heikko. STACK-järjestelmä puolestaan voisi sopia sekä lukioon että yliopistoon, mutta tehtävien laatimisen opettelu vie sillä enemmän aikaa kuin muilla järjestelmillä. Siinä ei myöskään ole WYSIWYG-editoria, kuten muissa järjestelmissä, joten sen käyttämän syntaksin tunteminen on tehtävän ratkaisijalle lähes välttämätöntä. STACK-järjestelmä on myös ainoa järjestelmä, jossa tehtävien automaattitarkastus sekä automaattinen vihjeiden ja palautteen antaminen ovat mahdollisia.

SISÄLLYS

KUVALUETTELO

TAULUKKOLUETTELO

LYHENTEET JA MERKINNÄT

MATLAB	MATrix LABoratory
STACK	a System for Teaching and Assessment using a Computer algebra Kernel

1. JOHDANTO

Nykyään matematiikassa tehdään tehtäviä entistä enemmän erilaisten sähköisten järjestelmien avulla. Ylioppilastutkintolautakunnan viimeisimpänä sähköistymiseen liittyvänä haasteena on sähköisiin ylioppilaskokeisiin siirtyminen. Niiden tekemistä ja niiden tekemiseen valmistautumista varten on luotu sähköinen kurssikoejärjestelmä Abitti. Matematiikan ylioppilaskoe muuttuu sähköiseksi viimeisenä, keväällä 2019. Abitin matematiikkavälineistöön tulee vielä muutoksia ennen matematiikan ensimmäisiä sähköisiä ylioppilaskirjoituksia. Abitin matematiikkavälineistö on kuitenkin nyt jo varsin kattava ja sen avulla onnistuu hyvin erilaisten tehtävien tekeminen. Työssä tutkitaan kahta Abitin tämän hetkisessä versiossa olevaa kaavaeditoria. Toinen niistä on 'libreoffice math'-kaavaeditori ja toinen vastauslomakkeen yhteyteen luotu nopeuteen ja helppouteen pyrkiä kaavaeditori. Vastauslomakkeen yhteydessä oleva kaavaeditori on vielä kehitysversio, joten siihen on tulossa vielä muutoksia. Seuraavat muutokset tulevat toukokuussa 2018, jolloin siihen lisätään ainakin muutamia uusia merkkejä. Yliopistoissa sähköisten laskuharjoitustehtävien ja tenttien käyttö on alkanut aiemmin kuin lukiossa ja siellä onkin käytössä matematiikan tehtävien tekemiseen useampia järjestelmiä. STACK-järjestelmä on useissa yliopistoissa käytössä oleva tehtävien sähköiseen tekemiseen käytettävä järjestelmä, jossa on automaattitarkastusominaisuus. Näin ollen STACK-järjestelmä vähentää huomattavasti myös tehtävän tarkastajan työmäärää. Toinen yliopistoissa usein käytetty järjestelmä on MATLAB, jonka lisäosana tuli vuoden 2016a versiossa myös Live Editor. Se mahdollisti normaalin tekstin, matemaattisen tekstin ja koodikielen kirjoittamisen samaan tiedostoon.

Työssä käsitellään toisen kertaluvun differentiaaliyhtälöitä ja vertaillaan tehtävänantojen tekemistä ja tehtävien ratkaisemista Abitilla, MATLABin Live Editorilla ja STACK-järjestelmällä. Työssä tutustutaan tarkemmin yhteen toisen kertaluvun differentiaaliyhtälön erikoistapaukseen, Van der Polin yhtälöön. Sähköisillä järjestelmillä tehtävänantojen tekemistä ja tehtävien ratkaisemista tarkastellaan esimerkkitehtävillä, joissa ratkaistaan Van der Polin systeemin ominaisarvot. Työssä tarkastellaan, millaista koodikielten osaamista tehtävänannon tekeminen vaatii sen laatijalta ja ratkaisijalta sekä sitä, millä tavoin tehtävien arviointi onnistuu eri järjestelmissä. Lisäksi pohditaan Abitin soveltuvuutta yliopistotason matematiikan laskuharjoitustehtävien tai tenttien ratkaisemiseen.

Työn alussa käydään läpi toisen kertaluvun differentiaaliyhtälöihin liittyvää yleistä teoriaa, jonka jälkeen syvennytään tarkastelemaan tarkemmin Van der Polin yhtälöä. Tämän jälkeen esitellään sähköisten järjestelmien vertailuun käytettävät esimerkkitehtävät malliratkaisuineen ja vertaillaan sähköisiä järjestelmiä esimerkkitehtävien avulla. Jokaista sähköistä järjestelmää käsitellään omassa alaluvussaan ja lopuksi kootaan yhteen vertailun tulokset.

2. TOISEN KERTALUVUN DIFFERENTIAALIYHTÄLÖT

Määritelmä 2.1 *Oletetaan, että y on muuttujan x reaaliarvoinen funktio. Tällöin funktiot $y', y'', ..., y^{(n)}$ ovat funktion y derivaattoja.* Differentiaaliyhtälö *voidaan kirjoittaa* yleisessä muodossa

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0$$

Funktion y korkeimman derivaatan kertaluku on differentiaaliyhtälön kertaluku.*[1, s. 231]*

Toisen kertaluvun differentiaaliyhtälöiden derivaatan kertaluku on siis kaksi.

Määritelmä 2.2 *Toisen kertaluvun differentiaaliyhtälö on* lineaarinen*, jos se on muotoa*

$$a_2(x)y''(x) + a_1(x)y'(x) + a_0(x)y(x) = f(x), \tag{2.1}$$

missä $a_2(x), a_1(x), a_0(x)$ ja $f(x)$ ovat muuttujan x funktioita ja y'' ja y' ovat funktion y derivaattoja. Jos $f(x) = 0$, niin yhtälö on homogeeninen. *Jos toisen kertaluvun differentiaaliyhtälöä ei voi esittää Yhtälön 2.1 mukaisessa lineaarisessa muodossa, niin se on* epälineaarinen.*[1, s. 477—478]*

Esimerkki 2.1 *Van der Polin yhtälö [2, s. 198]*

$$\frac{d^2x}{dt^2} - \alpha(1 - x^2)\frac{dx}{dt} + x = 0 \tag{2.2}$$

on epälineaarinen toisen kertaluvun differentiaaliyhtälö.

Työssä käytetään sähköisten järjestelmien tutkimiseen Van der Polin yhtälöön liittyviä esimerkkitehtäviä, jotka esitellään tarkemmin luvussa 4. Niissä ratkaistaan Van der Polin yhtälöstä muodotetun matriisin ominaisarvot, joten määritellään vielä ominaisarvo.

Määritelmä 2.3 *Olkoon $\boldsymbol{A}$ neliömatriisi. Tällöin*

$$det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0$$

on karakteristinen yhtälö*, missä*

$$det(\boldsymbol{A} - \lambda \boldsymbol{I})$$

on matriisin $\boldsymbol{A}$ karakteristinen polynomi *muuttujan λ suhteen. Karakteristisen yhtälön juuret ovat matriisin $\boldsymbol{A}$* ominaisarvoja*. Jos matriisin $\boldsymbol{A}$ ominaisarvo on λ, niin on olemassa sitä vastaava vektori $\boldsymbol{x}$, joka ei ole nollavektori ja joka toteuttaa yhtälön $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. Kaikki vektorit $\boldsymbol{x}$, joilla $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, ovat ominaisarvoa λ vastaavia* ominaisvektoreita*. [1, s. 130]*

3. VAN DER POLIN YHTÄLÖ

Toisen kertaluvun epälineaaristen differentiaaliyhtälöiden ratkaiseminen on huomattavasti monimutkaisempaa kuin lineaaristen toisen kertaluvun differentiaaliyhtälöiden ratkaiseminen. Epälineaariset differentiaaliyhtälöt ratkaistaan usein numeerisesti, eikä niille ole aina löydettävissä eksakteja ratkaisuja. [2, s. 457] Tämän vuoksi tarkastellaan esimerkkinä vain yhden epälineaarisen differentiaaliyhtälön, Van der Polin yhtälön, ratkaisemista kahdella eri tavalla.

3.1 Van der Polin yhtälön ratkaiseminen linearisaation avulla

Linearisaation avulla voidaan muuttaa toisen kertaluvun epälineaarinen differentiaaliyhtälö lineaariseksi, jolloin sen ratkaisemiseen voidaan käyttää samoja keinoja kuin lineaarisen differentiaaliyhtälön ratkaisemiseen. Linearisaatiota voidaan hyödyntää tasapainopisteen (engl. equilibrium point) lähellä.[2, s. 457]

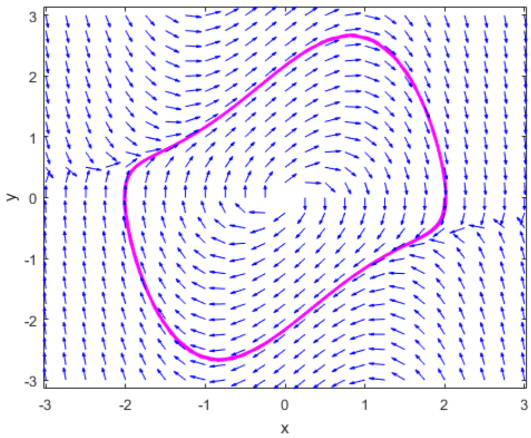
Esimerkki 3.1 *Tarkastellaan Esimerkin 2.1 Van der Polin yhtälöä. Olkoon* $a = 1$*, jolloin yhtälö on muotoa*

$$\frac{d^2x}{dt^2} - (1 - x^2)\frac{dx}{dt} + x = 0.$$

Merkitään $y = \frac{dx}{dt}$*. Tällöin Van der Polin yhtälö voidaan kirjoittaa yhtälöryhmänä [2, s. 458—459]*

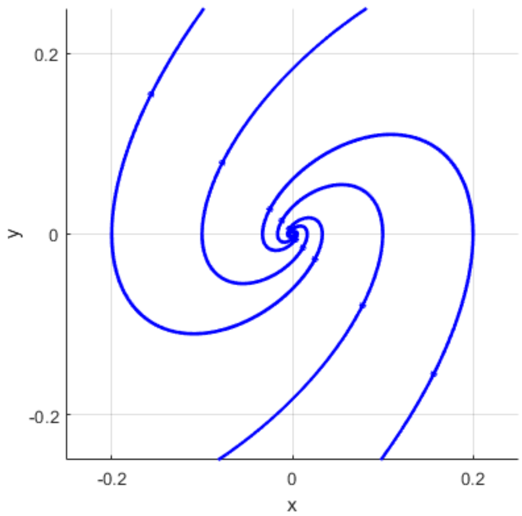
$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -x + (1 - x^2)y. \end{cases} \tag{3.1}$$

Yhtälöryhmää kutsutaan Van der Polin systeemiksi. Systeemin tasapainopiste on origossa, joten tutkitaan, miten ratkaisut käyttäytyvät sen läheisyydessä. Suuntakenttä (engl. direction field) on graafinen tapa approksimoida ensimmäisen kertaluvun differentiaaliyhtälöiden ratkaisuja yhtälöä ratkaisematta. Se muodostuu, kun eri kohtiin piirrettyjen tangentin suuntavektorien joukkoon piirretään ratkaisukäyrän approksimaatio, jonka suuntavektorit osoittavat. Van der Polin systeemille muodostetussa *suuntakentässä ratkaisut muodostavat kehän origon ympärille, kuten nähdään myös seuraavasta kuvasta.*



Kuva 3.1 *Suuntakenttä ja ratkaisujen muodostama käyrä*

Kuvaan 3.1 on piirretty vektorien suuntakenttä ja ne ratkaisut, jotka muodostavat kehän origon ympärille. Jos systeemin lähellä origoa olevia numeerisia approksimaatioita sijoitetaan koordinaatistoon, ne muodostavat kierteisen kuvion origon ympärille. Tätä on havainnollistettu Kuvassa 3.2.



Kuva 3.2 *Vaihekuva (engl. phase potrait) lähellä origoa*

Approksimaation avulla on helpompi analysoida systeemiä kuin ilman approksimaatiota. Yhtälöpari 3.1 ei ole lineaarinen, mutta siinä on vain yksi epälineaarinen termi x^2y*. Muuttujien* x *ja* y *ollessa pieniä termi* x^2y *on pienempi kuin mikään muu yhtälöparin termi. Esimerkiksi, jos* $x = 0,1$ *ja* $y = 0,1$*, niin* $x^2y = 0,001$*, joka on huo-*

mattavasti muuttujaa x ja y pienempi. Jos $x = 0,01$ ja $y = 0,01$, niin $x^2y = 10^{-6}$, joka on edelleen huomattavasti muuttujaa x ja y pienempi. Tämän vuoksi voidaan useissa tapauksissa approksimoida Van der Polin systeemiä jättämällä termi x^2y kokonaan pois. [2, s. 458—459] Tällöin yhtälöpari saadaan muotoon

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -x + y. \end{cases}$$

Saadusta muodosta voidaan helposti ratkaista esimerkiksi Van der Pol yhtälön ominaisarvot. Niiden ratkaisemiseen liittyvien tehtävien avulla vertaillaan työssä sähköisiä järjestelmiä.

3.2 Van der Polin yhtälön ratkaiseminen keskiarvottamisteorian avulla

Van der Polin yhtälön ratkaisemiseen voidaan käyttää myös keskiarvottamisteoriaa. Luvussa käsiteltävä Van der Polin yhtälön ratkaiseminen keskiarvottamisteorian avulla perustuu Väitöskirjaan [3, s. 11—15].

Määritelmä 3.1 *Tarkastellaan häiriöongelmaa (engl. perturbation problem) standardiyhtälön*

$$\frac{dx}{dt} = \varepsilon f(t,x) + \varepsilon^2 g(t,x,\varepsilon)$$

avulla, missä $y(t_0) = x_0$. Oletetaan, että funktion f jakson pituus on T, jolloin nähdään, että funktio f on keskimäärin muuttujan t yläpuolella muuttujan x ollessa vakio. Tällöin keskimääräinen yhtälö *on*

$$\frac{dy}{dt} = \varepsilon f^0(y), \tag{3.2}$$

missä $y(t_0) = x_0$ ja

$$f^0(y) = \frac{1}{T}\int_0^T f(t,x)\,dt. \tag{3.3}$$

Myös Van der Polin yhtälön huomataan olevan samanmuotoinen kuin edellä olevassa määritelmässä esiintyvä standardiyhtälö.

Määritelmä 3.2 *Olkoon $f(t,x)$ vektorikenttä, kun $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. Tällöin vektorikenttä on* Lipschitz-jatkuva *muuttujan x suhteen, kun $D \subset \mathbb{R}^n$ ja $t \geq 0$. Lisäksi vektorikenttä f on jatkuva muuttujien t ja x suhteen, kun $\mathbb{R}^+ \times D$. Jos keskiarvo*

$$f^0(x) = \lim_{t\to\infty} \frac{1}{T}\int_0^T f(t,x)\,dt, \tag{3.4}$$

on olemassa f, jota kutsutaan KBM-vektorikentäksi.

Lause 3.1 *(Yleinen keskiarvottaminen) Tarkastellaan alkuarvo-ongelmia*

$$\frac{dx}{dt} = \varepsilon f(t,x), \qquad x(0) = x_0,$$

missä $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ ja

$$\frac{dy}{dt} = \varepsilon f^0(y), \qquad y(0) = x_0,$$

missä $x, y, y_0 \in D \subset \mathbb{R}^n$, $t \in [0,\infty)$ ja $\varepsilon \in (0, \varepsilon_0]$. Oletetaan, että

a) *vektorikenttä f on KBM-vektorikenttä, jonka keskiarvo on f^0.*

b) *funktio $y(t)$ kuuluu joukon D sisäpisteisiin, kun aikajakson pituus on $\frac{1}{\varepsilon}$.*

Tällöin $x(t) - y(t) = O(\delta^{\frac{1}{2}}(\varepsilon))$, kun $\varepsilon \to 0$ aikajaksolla $\frac{1}{\varepsilon}$ ja

$$\delta(\varepsilon) = \sup_{x\in D} \sup_{t\in[0,\frac{L}{\varepsilon})} \left| \int_0^t [f(\tau,x) - f^0(x)]\, d\tau \right|,$$

missä L on positiivinen vakio ja keskiarvo

$$f^0(x) = \lim_{t\to\infty} \frac{1}{T} \int_0^T f(t,x)\, dt.$$

Lause 3.2 *Van der Polin yhtälön vaimennusfunktio $f(t) = (1 - x^2)\frac{dx}{dt}$ on pariton funktio.*

Todistus. Sijoitetaan vaimennusfunktioon $f(t) = (1 - x^2)\frac{dx}{dt}$ yhtälöparin

$$\begin{cases} x = r\cos(t+\psi) \\ y = -r\sin(t+\psi), \end{cases} \tag{3.5}$$

arvot. Tällöin funktio saadaan muotoon

$$f(t) = -(1 - r^2\cos^2(t+\psi))r\sin(t+\psi). \tag{3.6}$$

Koska funktiossa on parillisen ja parittoman funktion tulo, on saatu tulos pariton.

Esimerkki 3.2 *Tarkastellaan riippumatonta (engl. autonomous) Van der Polin yhtälöä*

$$\frac{d^2x}{dt^2} = -x + \alpha(1 - x^2)\frac{dx}{dt} + b\cos(\omega t). \tag{3.7}$$

Jos bcos(ωt) = 0, saadaan yhtälö Esimerkin 2.1 mukaiseen muotoon. Se voidaan edelleen kirjoittaa

$$\frac{d^2x}{dt^2} = -x + af(x, \frac{dx}{dt}),$$

missä f on mielivaltainen, riittävän sileä (engl. smooth) funktio joukossa D, kun $D \subset R^2$. Toisaalta alkuperäinen Yhtälö 3.7 voidaan kirjoittaa muuttujien x ja y derivaattojen avulla yhtälöryhmänä

$$\begin{cases} \frac{dy}{dt} = -x + b\cos(\omega t), \ y(0) = y_0 \\ \\ \frac{dx}{dt} = y - \alpha(\frac{x^3}{3} - x), \ x(0) = x_0. \end{cases} \tag{3.8}$$

Kun $b = 0$, yhtälöryhmä on muotoa

$$\begin{cases} \frac{dy}{dt} = -x \\ \frac{dx}{dt} = y - \alpha(\frac{x^3}{3} - x). \end{cases}$$

Kun muuttujat x ja y kirjoitetaan Lausekkeen 3.5 mukaisesti, niin häiriöyhtälöt (engl. perturbation equation) saadaan muotoon

$$\begin{cases} \frac{dr}{dt} = -\alpha\sin(t + \psi)f(r\cos(t + \psi), -r\sin(t + \psi)), \ r(0) = r_0 \\ \frac{d\psi}{dt} = -\frac{\alpha}{r}\cos(t + \psi)f(r\cos(t + \psi), -r\sin(t + \psi)), \ \psi(0) = \psi_0. \end{cases} \tag{3.9}$$

Vektorikenttä on 2π-jaksoinen muuttujan t suhteen. Merkitään $\tau = t + \psi$ ja määritellään Määritelmän 3.1 keskimääräiselle Yhtälölle 3.3 kaksi funktiota

$$\begin{cases} f_1(r) = \frac{1}{2\pi}\int_0^{2\pi}\sin(\tau)f(r\cos(\tau), -r\sin(\tau))\,d\tau \\ f_2(r) = \frac{1}{2\pi}\int_0^{2\pi}\cos(\tau)f(r\cos(\tau), -r\sin(\tau))\,d\tau. \end{cases} \tag{3.10}$$

Asymptoottinen approksimaatio saadaan ratkaisemalla yhtälöryhmä

$$\begin{cases} \frac{d\tilde{r}}{dt} = -\alpha f_1(\tilde{r}) \\ \frac{d\tilde{\psi}}{dt} = -\alpha\frac{f_2(\tilde{r})}{\tilde{r}} \end{cases} \tag{3.11}$$

käyttämällä alkuarvoina $r(0) = r_0$ ja $\psi(0) = \psi_0$. Muuttujat $\tilde{r}$ ja $\tilde{\psi}$ kuvaavat keskimääräisiä arvoja. Approksimaation avulla yhtälöryhmä saadaan pelkistettyä ensimmäisen kertaluvun riippumattomaksi systeemiksi.

Määritellään asymptoottinen approksimaatio Lauseen 3.2 vaimennusfunktiolle $f = (1-x^2)\frac{dx}{dt}$. *Kirjoitetaan ensin Yhtälöryhmä 3.9 matriisimuodossa*

$$\begin{pmatrix} \frac{dr}{dt} \\ \frac{d\psi}{dt} \end{pmatrix} = \begin{pmatrix} -\alpha \sin(t+\psi) f \\ \frac{\alpha}{r}\cos(t+\psi) f \end{pmatrix}.$$

Kosinitermin keskiarvo jaksossa on nolla, joten oikea puoli menee nollaan.

Merkitään $\tau = t + \psi$. *Tällöin Yhtälö 3.6 saadaan muotoon*

$$f(\tau) = -(1 - r^2\cos^2(\tau)) r \sin(\tau).$$

Sijoitetaan saatu funktio $f(\tau)$ *Yhtälöparin 3.10 funktioon* $f_1(r)$ *ja sievennetään lauseketta*

$$\begin{aligned}
f_1(r) &= -\frac{1}{2\pi}\int_0^{2\pi} r\sin^2(\tau)(1 - r^2\cos^2(\tau))\,d\tau \\
&= -\frac{1}{2\pi}\int_0^{2\pi} (r\sin^2(\tau) - r^3\cos^2(\tau)\sin^2(\tau))\,d\tau \\
&= -\frac{r}{2\pi}\int_0^{2\pi} \sin^2(\tau)\,d\tau + \frac{r^3}{2\pi}\int_0^{2\pi} (\sin^2(\tau)\cos^2(\tau)\,d\tau \\
&= -\frac{r}{2\pi}\int_0^{2\pi} \frac{1}{2}(1-\cos(2\tau))\,d\tau + \frac{r^3}{2\pi}\int_0^{2\pi} (1-\cos^2(\tau))\cos^2(\tau)\,d\tau \\
&= -\frac{r}{4\pi}\int_0^{2\pi} d\tau + \frac{r}{2\pi}\int_0^{2\pi}\cos(2\tau)\,d\tau + \frac{r^3}{2\pi}\int_0^{2\pi}(\cos^2(\tau)\,d\tau - \frac{r^3}{2\pi}\int_0^{2\pi}\cos^4(\tau)\,d\tau \\
&= -\frac{r}{4\pi}\int_0^{2\pi} d\tau + \frac{r}{2\pi}\int_0^{2\pi}\cos(2\tau)\,d\tau + \frac{r^3}{2\pi}\int_0^{2\pi}(\cos(2\tau)+1)\,d\tau \\
&\quad - \frac{r^3}{2\pi}\int_0^{2\pi}\frac{1}{8}(4\cos(2\tau)+\cos(4\tau)+3)\,d\tau \\
&= -\frac{r}{4\pi}\Big/_{0}^{2\pi}\tau + \frac{r}{2\pi}\Big/_{0}^{2\pi}\frac{1}{2}\sin(2\tau) + \frac{r^3}{4\pi}\Big/_{0}^{2\pi}\left(\frac{1}{2}\sin(2\tau)+\tau\right) \\
&\quad - \frac{r^3}{2\pi}\Big/_{0}^{2\pi}\frac{1}{8}\left(2\sin(2\tau)+\frac{1}{4}\sin(4\tau)+3\tau\right) \\
&= -\frac{r}{2} + \frac{r^3}{2} - \frac{3r^3}{8} \\
&= \frac{1}{8}r^3 - \frac{1}{2}r.
\end{aligned}$$

Sijoittamalla tulos Yhtälöryhmään 3.11 saadaan

$$\begin{cases} \frac{d\tilde{r}}{dt} = -\alpha f_1(\tilde{r}) = \alpha(\frac{1}{2}\tilde{r} - \frac{1}{8}\tilde{r}^3) = \frac{\alpha\tilde{r}}{2}(1 - \frac{1}{4}\tilde{r}^2) \\ \frac{d\tilde{\psi}}{dt} = 0. \end{cases} \tag{3.12}$$

Jos alkuarvo r_0 on 0 tai 2, niin amplitudi $\tilde{r}$ on vakio. Tarkastellaan ratkaisun $r = 0$ vakautta häiritsemällä alkuarvoa $r_0 = 0$ vastaavaa tasapainopistettä hieman muuttujan ξ avulla. Merkitään $r = 0 + \xi$, tällöin Yhtälöryhmästä 3.12 saadaan

$$\frac{d\xi}{dt} = \frac{\alpha}{2}\xi + O(\xi).$$

Kun ei huomioida toisen ja korkeamman kertaluvun termejä, $\xi = \xi_0 e^{\frac{\alpha t}{2}}$. Termi ξ lähestyy ääretöntä, kun aika t kasvaa, jolloin tasapainopisteestä tulee epävakaa.

Tarkastellaan vastaavasti toisen alkuarvoa $r_0 = 2$ vastaavan tasapainopisteen vakautta. Merkitään $r_0 = 2 + \xi$. Tällöin Yhtälöryhmästä 3.12 saadaan

$$\frac{d\xi}{dt} = \alpha(-\xi + O(\xi^2] + O'(\xi^3)).$$

Tätä voidaan edelleen muokata, kun ei huomioida toisen tai korkeamman kertaluvun termejä. Tällöin saadaan

$$\frac{d\xi}{dt} = -\alpha\xi,$$

ja puolittain integroimalla

$$\xi = \xi_0 e^{-\alpha t}.$$

Termi ξ lähestyy nollaa, kun aika t kasvaa, joten alkuarvoa $r_0 = 2$ vastaava tasapainopiste on vakaa.

Ratkaistaan Differentiaaliyhtälö 3.12. Muokataan se aluksi muotoon

$$-\frac{8}{\alpha}\left(\frac{1}{\tilde{r}^3 - 4\tilde{r}}\right) d\tilde{r} = 1dt, \tag{3.13}$$

missä

$$\frac{1}{\tilde{r}^3 - 4\tilde{r}} = \frac{1}{-\tilde{r}(\tilde{r}+2)(\tilde{r}-2)}. \tag{3.14}$$

Tehdään termille 3.14 osamurtohajotelma

$$\tfrac{1}{-\tilde{r}(\tilde{r}+2)(\tilde{r}-2)} = \tfrac{A}{-\tilde{r}} + \tfrac{B}{\tilde{r}+2} + \tfrac{C}{\tilde{r}-2}$$

ja kerrotaan sen molemmat puolet termillä $-\tilde{r}(\tilde{r}+2)(\tilde{r}-2)$

$$1 = A(\tilde{r}+2)(\tilde{r}-2) - B\tilde{r}(\tilde{r}-2) - C\tilde{r}(\tilde{r}+2.) \tag{3.15}$$

Ratkaistaan siitä kertoimet A, B *ja* C *sijoittamalla reaalijuuret* -2, 0 *ja* 2 *muuttujan* $\tilde{r}$ *paikalle. Kun*

$$\begin{cases} \tilde{r} = -2 \\ \tilde{r} = 0 \\ \tilde{r} = 2 \end{cases} \Leftrightarrow \begin{cases} A = -\frac{1}{4} \\ B = -\frac{1}{8} \\ C = -\frac{1}{8}. \end{cases}$$

Joten

$$\frac{1}{\tilde{r}^3 - 4\tilde{r}} = \frac{1}{4\tilde{r}} - \frac{1}{8(\tilde{r}+2)} - \frac{1}{8(\tilde{r}-2)}.$$

Tämä voidaan sijoittaa Yhtälöön 3.13. Tällöin

$$-\frac{8}{\alpha}\left(\frac{1}{4\tilde{r}} - \frac{1}{8(\tilde{r}+2)} - \frac{1}{8(\tilde{r}-2)}\right) d\tilde{r} = 1dt.$$

Integroidaan yhtälö puolittain, jolloin

$$\frac{8}{\alpha}(\frac{1}{4}\ln|\tilde{r}| - \frac{1}{8}\ln|\tilde{r}+2| - \frac{1}{8}\ln|\tilde{r}-2|) + C_1 = t + C_2.$$

Sievennetään vielä yhtälöä ja yhdistetään integroimisvakiot, jolloin

$$-\frac{1}{\alpha}(2\ln|\tilde{r}| - \ln|\tilde{r}+2| - \ln|\tilde{r}-2|) = t + D.$$

Nyt voidaan ratkaista $\tilde{r}$, *jolloin eräs yhtälön ratkaisu*

$$\tilde{r}(t) = \frac{2e^{\frac{1}{2}(\alpha D + \alpha t)}}{\sqrt{e^{(\alpha D + \alpha t)} - 1}}.$$

Osoitetaan, että myös Väitöskirjassa [3, s. 11—15] oleva ratkaisu

$$\tilde{r}(t) = \frac{2e^{\frac{1}{2}\alpha t}}{\sqrt{e^{\alpha t} + \frac{1}{4r_0} - 1}} \tag{3.16}$$

toteuttaa Differentiaaliyhtälön 3.12 tarkastelemalla differentiaaliyhtälöä

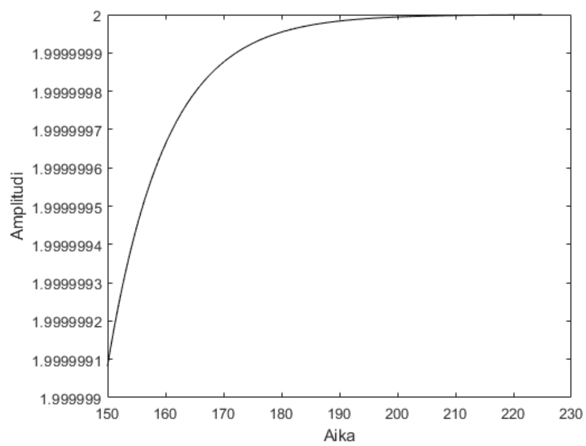
$$\frac{d\tilde{r}}{dt} - \frac{\alpha\tilde{r}}{2}(1 - \frac{1}{4}\tilde{r}^2) = 0. \tag{3.17}$$

Huomataan, että Määritelmän 2.2 mukaan Differentiaaliyhtälö 3.17 on homogeeninen. Näytetään, että Differentiaaliyhtälön 3.17 homogeenisuus toteutuu.

$$\begin{aligned}
&\frac{d\tilde{r}}{dt} - \frac{\alpha\tilde{r}}{2}(1 - \frac{1}{4}\tilde{r}^2) \\
= \;& \frac{d}{dt}\frac{2e^{\frac{1}{2}\alpha t}}{\sqrt{e^{\alpha t} + \frac{1}{4r_0} - 1}} - \frac{\alpha}{2}\frac{2e^{\frac{1}{2}\alpha t}}{\sqrt{e^{\alpha t} + \frac{1}{4r_0} - 1}}\left(1 - \frac{1}{4}\left(\frac{2e^{\frac{1}{2}\alpha t}}{\sqrt{e^{\alpha t} + \frac{1}{4r_0} - 1}}\right)^2\right) \\
= \;& \frac{\alpha e^{\frac{\alpha t}{2}}}{\sqrt{e^{\alpha t} + \frac{1}{4r_0} - 1}} - \frac{\alpha e^{\frac{3\alpha t}{2}}}{(e^{\alpha t} + \frac{1}{4r_0} - 1)^{\frac{3}{2}}} - \frac{\alpha e^{\frac{\alpha t}{2}}}{\sqrt{e^{\alpha t} + \frac{1}{4r_0} - 1}} - \frac{\alpha e^{\frac{3\alpha t}{2}}}{(e^{\alpha t} + \frac{1}{4r_0} - 1)^{\frac{3}{2}}} \\
= \;& 0.
\end{aligned}$$

Homogeenisuus toteutuu eli myös Väitöskirjassa [3, s. 11—15] oleva Ratkaisu 3.16 toteuttaa Differentiaaliyhtälön 3.12.

Amplitudi $\tilde{r}$ pyrkii saavuttamaan vakio arvon 2, kun aika t kasvaa. Tämä on nähtävillä seuraavassa kuvassa kaikille alkuehtojen r_0 arvoille.



Kuva 3.3 *Keskimääräinen amplitudi $\tilde{r}$ ajan t funktiona*

Kuva 3.3 on piirretty Funktion 3.16 avulla. Alkuarvoina käytettiin $\alpha = 0,1$ ja $r_0 = 1$. Tähän asti on käytetty muuttujan r ilmaisemiseen keskiarvottamisen teoriaa, joten häiriötön muuttuja r on saman suuruinen kuin $\tilde{r} + O(\alpha)$. Tällöin Lauseke 3.5 on muotoa

$$x(t) = r_0\cos(t + \psi) + O(\alpha). \tag{3.18}$$

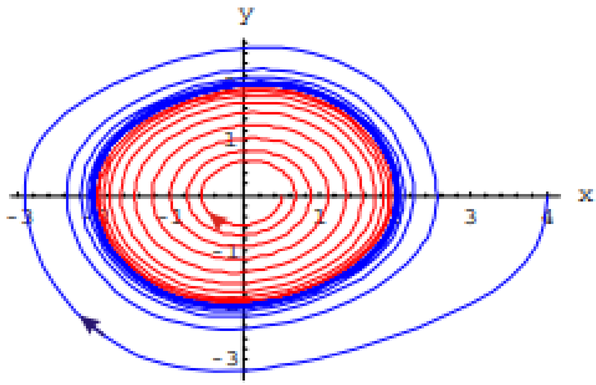
Kun aikajakson pituus on $\frac{1}{\alpha}$ ja alkuarvo $r_0 = 2$, niin

$$x(t) = 2\cos(t + \psi) + O(\alpha).$$

Lisäksi muuttujan x yleiseksi muodoksi saadaan

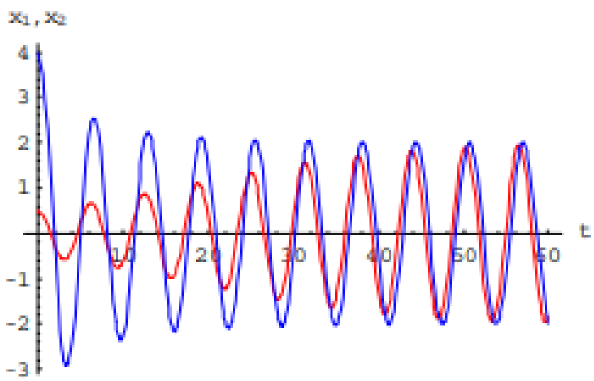
$$x(t) = \frac{r_0 e^{\frac{\alpha t}{2}}}{\sqrt{1 + \frac{1}{4} r_0^2 (e^{\alpha t} - 1)}} \cos(t + \psi_0) + O(\alpha).$$

Arvoille $r_0 = 0$ ja $r_0 = 2$ vakaa ja ajan suhteen jaksollinen ratkaisu on Yhtälö 3.18. Muilla muuttujan r arvoilla ratkaisut pyrkivät kohti jaksollista Ratkaisua 3.18. Ratkaisun vaiheita kutsutaan rajajakson kiertämiseksi. Kuvassa 3.4 on kuvattuna rajajakso.



Kuva 3.4 *Rajajakso[3, s. 11—15]*

Kuvassa 3.5 kahden kiertoradan kehittyminen ajan suhteen eri alkuarvoilla.



Kuva 3.5 *Kahden eri alkuarvot omaavan kiertoradan kehittyminen ajan funktiona[3, s. 11—15]*

Ensimmäinen kiertoratojen kehittymistä kuvaava käyrä alkaa rajajakson sisältä ja *toinen sen ulkopuolelta.*

4. SÄHKÖISET JÄRJESTELMÄT TEHTÄVIEN TEON APUVÄLINEENÄ

Työssä käsitellään matematiikan tehtävien tehtävänantojen laatimista ja tehtävien ratkaisemista kolmella sähköisellä järjestelmällä eli Abitilla, MATLABin Live Editorilla ja STACK-järjestelmällä. Abitti on lukion sähköinen kurssikoejärjestelmä[4]. Se on optimoitu käytettäväksi nimen omaan lukiossa. MATLABin Live Editor ja STACK-järjestelmä ovat puolestaan yliopistojen käytössä olevia järjestelmiä, joita käytetään yliopistossa tehtävien teon apuvälineinä. Työssä tutkitaan kahden esimerkkitehtävän avulla kunkin järjestelmän ominaisuuksia. Erityisesti kiinnitetään huomiota siihen, mitä järjestelmien käyttäjien täytyy osata, miten tehtävien arvioiminen onnistuu ja millaisten tehtävien ratkaiseminen on mielekkäintä. Lisäksi testataan Abitin sopivuutta yliopistomatematiikan ratkaisemiseen.

4.1 Sähköisten järjestelmien testaamiseen käytettävät tehtävät

Sähköisten tehtävien ratkaiseminen vaatii erilaista osaamista kuin perinteinen paperilla ja kynällä tehtävien ratkaiseminen. Jos sähköiset järjestelmät eivät ole entuudestaan tuttuja, vie niiden käytön opettelu jonkun verran aikaa ja aluksi uuden järjestelmän käyttäminen voi olla hidasta. Tämän vuoksi eri järjestelmiä testataan kahden eri tehtävän avulla. Esimerkin 4.1 tehtävänanto vaatii ratkaisijalta vähemmän sähköisesti tapahtuvaa kirjoittamista kuin Esimerkin 4.2 tehtävänanto.

Esimerkki 4.1 *Van der Polin yhtälösysteemin*

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -x + (1 - x^2)y \end{cases}$$

termi x^2y voidaan approksimoida nollaksi. Approksimaatiota $x^2y = 0$ käyttäen saadaan ratkaistua ominaisarvot yhtälösysteemille. Ratkaise
a) matriisi $\boldsymbol{A}$, jota tarvitset ominaisarvojen laskemiseen.
b) toisen asteen yhtälö, josta saadaan ratkaistua ominaisarvot.

c) Van der Polin systeemin ominaisarvot
Kirjoita kaikkiin kohtiin vain lopullinen vastaus. Välivaiheet voi tarvittaessa kirjoittaa omalle suttupaperille.

Esimerkin 4.1 tehtävänannossa on a-, b- ja c-kohdat. Jokaiseen kohtaan pyydetään vain lopulliset vastaukset ilman välivaiheita, joten sähköisesti kirjoitettavien asioiden määrä on melko vähäinen. Lisäksi tehtävänannossa oletetaan, että ratkaisijalla on käytössä suttupaperi, joka mahdollisesti nopeuttaa tehtävän hahmottamista.

Esimerkki 4.2 *Ratkaise Van der Polin yhtälösysteemin*

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -x + (1 - x^2)y \end{cases}$$

ominaisarvot käyttäen approksimaatiota $x^2y = 0$.

Esimerkissä 4.2 jaottelu a-, b- ja c-kohtiin puuttuu, joten selkeän ja ymmärrettävän vastauksen kirjoittaminen vaatii paljon enemmän sähköisellä järjestelmällä kirjoittamista. Seuraavana on esitetty Esimerkin 4.1 tehtävänannon malliratkaisu.

Esimerkki 4.3 *a)*

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix}$$

b) $\lambda^2 - \lambda + 1 = 0$

c) $\lambda_1 = \frac{1+i\sqrt{3}}{2}$ *ja* $\lambda_2 = \frac{1-i\sqrt{3}}{2}$

Esimerkin 4.2 malliratkaisu on puolestaan esitetty seuraavassa esimerkissä.

Esimerkki 4.4

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix}$$

Ratkaistaan ominaisarvot λ *yhtälöstä*

$$\begin{aligned}
&det(\boldsymbol{A} - \boldsymbol{I}\lambda) = 0\\
&\Leftrightarrow \begin{vmatrix} 0-\lambda & 1 \\ -1 & 1-\lambda \end{vmatrix} = 0\\
&\Leftrightarrow \begin{vmatrix} -\lambda & 1 \\ -1 & 1-\lambda \end{vmatrix} = 0\\
&\Leftrightarrow -\lambda(1-\lambda) + 1 = 0\\
&\Leftrightarrow \lambda^2 - \lambda + 1 = 0\\
&\Leftrightarrow \lambda = \frac{1 \pm \sqrt{1^2 - 4 \cdot 1 \cdot 1}}{2 \cdot 1}\\
&\Leftrightarrow \lambda = \frac{1 \pm \sqrt{-3}}{2}\\
&\Leftrightarrow \lambda = \frac{1 \pm i\sqrt{3}}{2}
\end{aligned}$$

Eli ominaisarvot ovat

$\lambda_1 = \frac{1+i\sqrt{3}}{2}$ *ja* $\lambda_2 = \frac{1-i\sqrt{3}}{2}$

Tarkastellaan seuraavissa kappaleissa Esimerkkien 4.1 ja 4.2 tehtävänantojen ja Esimerkkien 4.3 ja 4.4 malliratkaisujen tekemistä Abitilla, MATLABin Live Editorilla ja STACK-järjestelmällä.

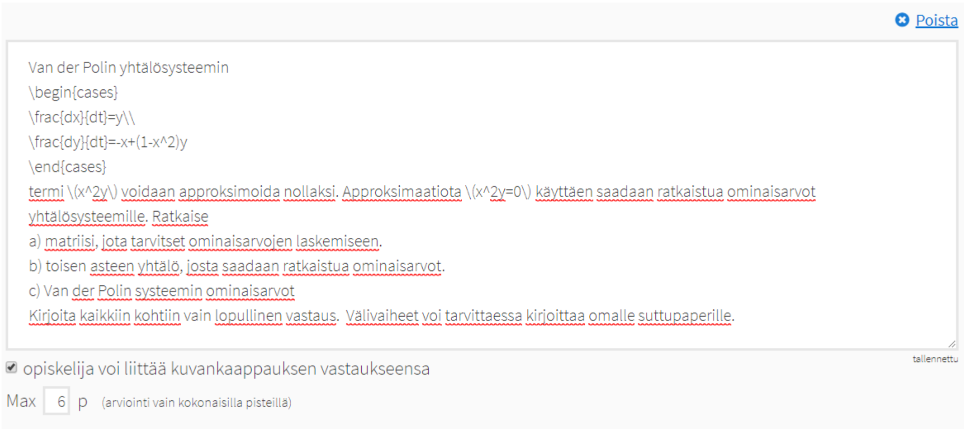
4.2 Abitti-järjestelmä

Abitti on sähköisiä ylioppilaskokeita varten kehitetty Ylioppilastutkintolautakunnan kurssikoejärjestelmä, joka julkaistiin tammikuussa 2015[4]. Sähköisten ylioppilaskokeiden tekemistä varten tarvittiin jokin sähköinen järjestelmä, joten luotiin Abitti. Sen avulla pyritään tutustuttamaan ylioppilastutkintoon osallistuvat kokelaat ja kokeita järjestävät tahot ylioppilastutkinnon sähköisissä kokeissa käytettävään järjestelmään[4]. Abittiin voi rekisteröityä kuka tahansa Ylioppilastutkintolautakunnan 'https://www.abitti.fi/fi/ohjeet/rekisteroityminen/' -sivun kautta. Kirjautumalla Abittiin voi luoda omille sivuille omia kokeita.[5]

Matemaattisen tekstin kirjoittamista voi harjoitella Abittikokeen vastauslomakkeen demosivulla osoitteessa 'https://math-demo.abitti.fi' olevalla kaavaeditorin kehitysversiolla. Editoria voi käyttää hyödyntämällä LaTeX-koodikieltä tai käyttämällä sitä WYSIWYG-editorina (What You See Is What You Get). [6] Tämän vuoksi ei siis ole välttämätöntä osata LaTeX-koodikieltä. Koska Abitti on suunnattu lukiolaisille, on varmasti yleisempää, etenkin lukion juuri alottaneiden keskuudessa, että järjestelmää käytetään ilman LaTeX-koodikielen osaamista. Tarvittavat merkinnät valitaan niitä vastaavista kuvakkeista kaavaeditorista. Vaikka matemaattisen tekstin kirjoitusharjoitukset onnistuvat demosivulla, ei kysymyksiin tai koetehtäviin voi vastata muuta kuin koejärjestelmän kautta, sillä demosivulle ei voi syöttää tehtävänantoa, eikä sitä kautta voi lähettää kirjoittamaansa vastausta arvioitavaksi. Koejärjestelmä koostuu kokelaiden koneista eli päätelaitteista, kahdesta palvelimesta ja paikallisverkosta[6]. Koetilanteessa ja tehtäviä testatessa käynnistetään muistitikulta Ylioppilastutkintolautakunnan toimittama käyttöjärjestelmä toiselle koneista; koetilanteessa oppilaan koneelle. Koetilassa olevista palvelimista toisen kautta kokelaat tai kokeen testaaja saavat koetehtävänsä ja oheisaineistot. Sen lisäksi saman palvelimen kautta saadaan vastaanotettua koesuoritukset. Toinen palvelin eli varapalvelin puolestaan kopio kaikki tiedot. Tämän lisäksi tarvitaan paikallisverkko ja etenkin testauksessa ja tulevissa ylioppilaskirjoituksissa verkko on langallinen. Kokeen tai testitehtävien teon jälkeen suoritukset siirretään verkkopalveluun, jonka kautta tarkastaja pääsee tarkastamaan tehtävät.[6] Kokeen testaamiseen tarvitaan siis kaksi tietokonetta, verkkokaapeli ja kolme muistitikkua. Tämän vuoksi kokeen testaaminen on hankalaa.

4.3 Tehtävänantojen tekeminen Abitilla

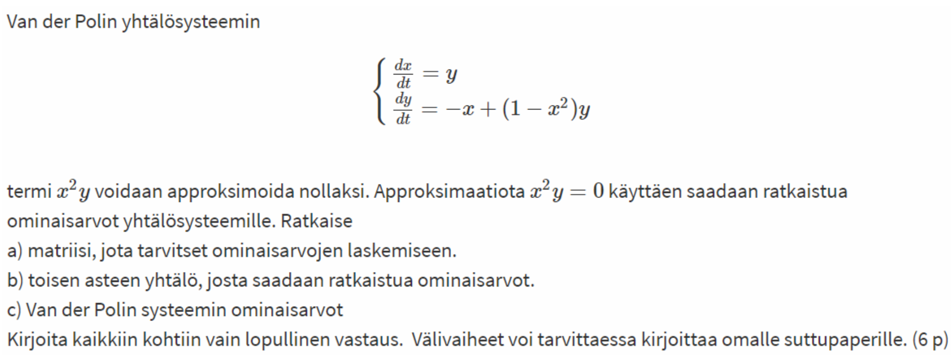
Tehtävänantojen luominen onnistuu Abitin omien sivujen kautta. Aluksi painetaan 'Luo uusi koe' -painiketta, jonka jälkeen luotavalle kokeelle annetaan nimi. Yksi koe on käytettävissä ainoastaan kerran, jos koetta ei monista. Yleisiä lisäohjeita kokelaalle voi antaa sille varattuun kenttään. Ensimmäisen kysymyksen luominen aloitetaan valitsemalla kysymyksen tehtävätyyppi painamalla halutun tehtävätyypin painikkeesta. Valittavissa on kolme erilaista tehtävätyyppiä: tekstikenttä, monivalinta ja aukkotehtävä. [7] Koska Abitin toimintaa testataan työssä Esimerkkien 4.1 ja 4.1 avulla, käytetään tehtävänantojen luomiseen tekstikenttä-tehtävätyyppiä. Työssä tehtävät kirjoitetaan LaTeX-koodikielellä, vaikka tehtävien kirjoittaminen onnistuisi myös AsciiMath-koodikielellä[5]. Ennen koodia kirjoitetaan tehtävänannon kirjoituskenttään kenoviiva ja sulku auki mainitussa järjestyksessä ja koodin jälkeen kenoviiva ja sulku kiinni jälleen edellä mainitussa järjestyksessä[5]. Tehtävänannon kirjoittaminen on siis melko helppoa, jos osaa LaTeX- tai AsciiMath-koodikieltä. Internetistä löytyy myös paljon ohjeita LaTeX-koodikielen käyttöön Abitissa, joten myös LaTeX-koodikieltä osaamattoman on helppo oppia tekemään tehtäviä Abitilla. Seuraavana on esitetty kuvakaappaus tekstikenttä-tehtävätyypillä luodusta tehtävästä.



Kuva 4.1 *Tehtävänanto kirjoitettuna sille tarkoitettuun tekstikenttään*

Kuvan 4.1 tekstikentässä näkyy Esimerkin 4.1 tehtävänanto kirjoitettuna LaTeX-koodikiellä. Lisäksi tekstikentän alapuolella näkyy, saako tehtävän ratkaisija liittää vastaukseensa kuvakaappauksen, ja tehtävästä saatava maksimipistemäärä. Kuten kuvasta huomataan, työssä saa liittää tehtävän ratkaisun yhteyteen kuvakaappauksen. Jos kuvakaappausta ei saisi liittää, vastausta ei pystyisi kirjoittamaan kaavaeditoreja käyttäen, eikä se ole työssä tarkoituksenmukaista, koska työssä on tarkoitus testata Abitin 'libreoffice math' -kaavaeditoria ja vastauslomakkeen kaavaeditoria. Muut Abitissa olevat ohjelmat jätetään työssä huomiotta, koska testitehtävien ratkaisemisen onnistuu ilman niitä.

Luotua tehtävää voi esikatsella kokeen luomisen yhteydessä painamalla 'Esikatsele koetta' -painiketta. Esikatselu on nopea tapa varmistua siitä, ettei koodiin ole jäänyt virheitä. Kuvassa 4.2 on kuvakaappaus Kuvan 4.1 tekstikentässä olevan tehtävän esikatselusta.

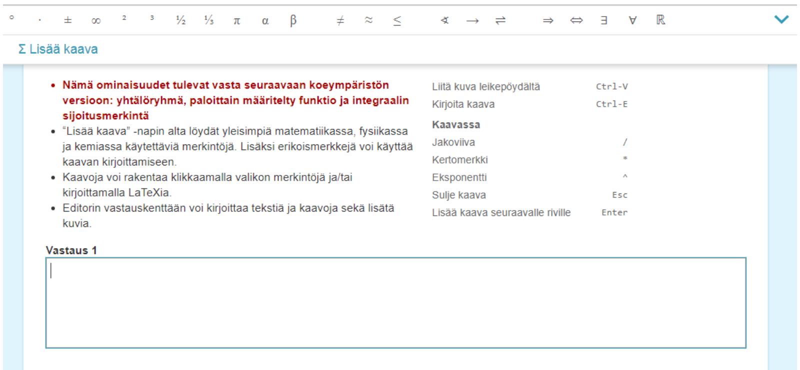


Kuva 4.2 *Tehtävänannon esikatselu*

Tehtävänannon esikatseleminen on nopeaa tehtävänannon teon yhteydessä, mutta tehtävän ratkaisemisen testaaminen on monimutkaisempaa.

4.4 Tehtävien ratkaiseminen Abitilla

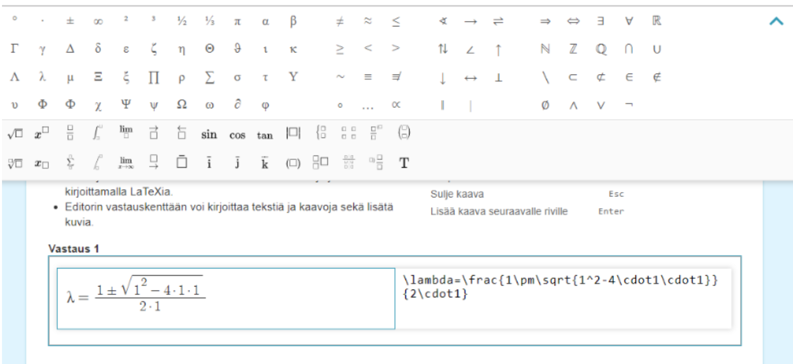
Vastauslomakkeen kaavaeditorista on pyritty luomaan mahdollisimman nopea ja helppokäyttöinen[8]. Sen kehitysversiota voi käyttää aiemmin manitulla demosivulla. Kehitysversiosta on kuvakaappaus seuraavassa kuvassa.



Kuva 4.3 *Vastauslomakkeen kaavaeditorin kehitysversio*

Koetilan palvelimen vastauslomakkeen kaavaeditorissa vastaus kirjoitetaan vastaavanlaiseen tekstikenttään kuin Kuvan 4.3 kehitysversiossa.

Kuvassa 4.3 nähdään osa käytettävissä olevista merkeistä. Kaikki valittavat merkit saadaan sekä koetilan palvelimen vastauslomakkeen yhteydessä että kehitysversion yhteydessä näkyviin, kun laajennetaan merkkivalikko Kuvan 4.3 oikeassa yläkulmassa olevasta nuoli-painikkeesta. Lisäksi kaavaeditorissa käytettävissä olevat kaavat saadaan näkyviin painamalla '$\sum$ Lisää kaava' -painikkeesta tai käyttämällä pikanäppäin yhdistelmää 'Ctrl-E'. Seuraavassa kuvassa on esitetty kaikki kehitysversiossa ja tämän hetkisessä vastauslomakkeen kaavaeditorissa käytössä olevat merkit ja kaavat.

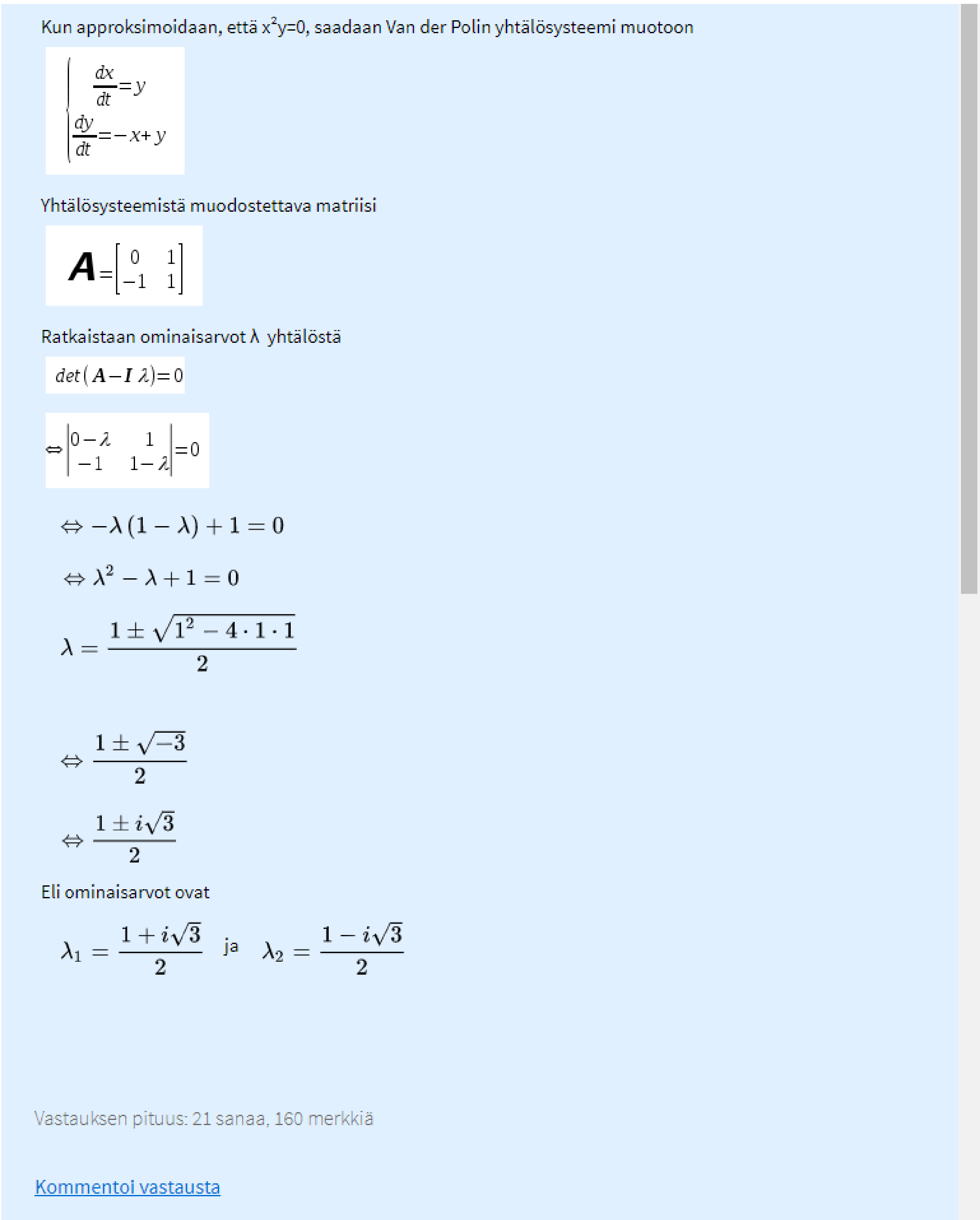


Kuva 4.4 *Vastauslomakkeen kaavaeditorin kehitysversiossa olevat merkit ja kaavat sekä esimerkki kaavan kirjoittamisesta tekstikenttään*

Merkkien ja kaavojen lisäksi Kuvassa 4.4 näkyy tekstikenttä, johon on kirjoitettu Esimerkin 4.4 malliratkaisussa esiintyvä toisen asteen yhtälön ratkaisukaava. Sen avulla havainnollistetaan tekstikentän kaavankirjoittamisnäkymää. Kuvan 4.4 tekstikentässä on näkyvillä kaavankirjoittamisnäkymä, kun se suljetaan, jää jäljelle pelkkä toisen asteen yhtälön ratkaisukaava ilman vieressä näkyvää koodikieltä. Tämän jälkeen kirjoittamista voi myös jatkaa normaalisti tekstikenttään kirjoitetun kaavan ympärille. Myös koetilan palvelimen kaavaeditori toimii vastaavasti.

Esimerkin 4.1 ja 4.2 tehtävät ratkaistiin Abitilla käyttäen vastauslomakkeen kaavaeditoria ja 'libreoffice math' -kaavaeditoria. Vastauksista pyrittiin luomaan vastaavat kuin Esimerkkien 4.3 ja 4.4 malliratkaisuista. Vastauslomakkeen kaavaeditorista puuttui vielä tietyt ominaisuudet, kuten yhtälöryhmän kirjoittaminen, matriisi ja determinantti, joten koko tehtävää ei pystynyt kirjoittamaan käyttämällä vastauslomakkeen kaavaeditoria. 'Libreoffice math' -kaavaeditorissa oli käytössä enemmän erilaisia merkkejä ja kaavoja, joten sen avulla sai tehtyä tehtävien ratkaisuihin ne kohdat, joita ei vastauslomakkeen kaavaeditorilla pystynyt tekemään. 'Libreoffice math' -kaavaeditorin valikoista oikeiden merkkien etsiminen oli aluksi hidasta, mutta nopeutui melko nopeasti editorin tullessa tutuksi. Valikoista merkkien etsimisen sijasta oli kaavojen kirjoittaminen mahdollista myös käyttämällä suoraan 'libreoffice math' -kaavaeditorin käyttämää syntaksia. Osaltaan tehtävien ratkaisemista hidasti välilehdestä toiseen liikkuminen. 'Libreoffice math' -kaavaeditori avautui uuteen välilehteen, joten saman aikaisesti ei voinut nähdä tehtävänantoa ja tehtävän vastaukseen kirjoitettavaa kaava. Editorissa ei voinut myöskään kirjoittaa tekstiä kaavojen väliin, koska se tuli kursivoituna ja kaikki sanat yhteen kirjoitettuna. Tämän vuoksi jokainen tehtävän ratkaisemiseksi tarvittava kaava oli liitettävä aina kuvakaappauksella vastauslomakkeeseen ja kirjoitettava siellä kaavojen väliin tekstiä.

Abitin uusi versio tulee ladattavaksi muistitikulle 2.5.2018, jolloin vastauslomakkeen kaavaeditoriin tulee joitakin uusia ominaisuuksia, kuten yhtälöryhmä, paloittain määritelty funktio ja integraalin sijoitusmerkintä[9][10]. Kaavaeditorin kehitysversioon korjataan siis vähitellen havaittuja puutteita, jotta se saadaan palvelemaan lukiolaisia entistä paremmin. Matriisin puuttuminen on myös havaittu, mutta matriisilaskennan osaaminen ei kuulu lukion opetussuunnitelmaan[10][11, s. 131—139]. Tämän vuoksi sen lisäämisellä vastauslomakkeen kaavaeditoriin ei ole kiire. Koska Abitin ominaisuuksia tutkittiin Esimerkkien 4.1 ja 4.2 avulla, huomattiin nopeasti, että Abitin kohderyhmänä ovat lukiolaiset, eikä sen tämän hetkinen kehitysversio vielä sovellu sujuvasti yliopistomatematiikan perustehtävien ratkaisemiseen. Vastauslomakkeen kaavaeditorin käyttö oli nopeaa LaTeX-koodikielellä, mutta kuvakkeita painelemalla se on hidasta ennen kunnollisen rutiinin saavuttamista. Tällä hetkellä oletetaan, että useimmat käyttäjät kirjoittavat matemaattista tekstiä ensin käyttäen kuvakkeita, joissa näkyy haluttu merkki, mutta vähitellen osan uskotaan siirtyvän käyttämään LaTeX-koodikieltä, sillä se on nopeampaa.[8] . Seuraavassa kuvassa on esitetty Esimerkin 4.4 mukainen malliratkaisu kirjoitettuna Abittia käyttäen.



Kuva 4.5 *Tehtävän ratkaisu Abitilla tehtynä*

Kuvasta 4.5 huomataan, että vastauksesta saa luotua melko siistin, mutta tietyt kohdat on liitetty vastaukseen 'libreoffice math' -kaavaeditorista otettujen kuvakaappausten avulla, joten ne näkyvät tehtävässä eri värisellä taustalla. Koska Abitin vastauslomakkeen kaavaeditorilla pyritään nopeuteen ja helppoutteen, on Abitilla järkevin ratkaista sellaisia kaavaeditorilla ratkaistavia tehtäviä, jotka pystyy tekemään kokonaan vastauslomakkeen kaavaeditorilla. Näin saadaan käytettyä hyödyksi vastauslomakkeen kaavaeditorin nopeus ja helppous. Kuvan 4.5 alareunassa näkyy myös 'Kommentoi vastausta' -kohta. Siitä painamalla avautuu ratkaisijan vastauksen alapuolelle tekstikenttä, johon tarkastaja voi kirjoittaa tehtäväkohtaisen palautteen. Myös tehtävästä saatavat pisteet merkataan omaan tekstikenttäänsä tehtävän ratkaisun arvioinnin yhteydessä.

4.5 MATLABin live-editori

MATLAB (MATrix LABoratory) on numeerinen laskentaohjelmisto, jonka on kehittänyt MathWorks. MATLABia käytetään monessa maassa tutkijoiden ja insinöörien keskuudessa.[12, s. 9][13, s. 13] Sen avulla voi suorittaa symbolista laskentaa, signaalinkäsittelyä, visualisointia ja simuloimista monipuolisten työkalujen avulla [12, s. 9]. MATLABin Live Editor on MATLABin yhteyteen luotu lisäominaisuus, joka on ollut käytössä 2016a -versiosta lähtien. Siinä on mahdollista kirjoittaa LaTeX-koodikielen lisäksi myös normaalia tekstiä ja matemaattisia kaavoja. Versiosta 2016b lähtien Live Editoria on pystynyt käyttämään myös WYSIWYG-editorina, jonka ansiosta sillä voi kirjoittaa matemaattista tekstiä ilman LaTeX-koodikielen osaamista.[13, s. 13]

MATLABin Live Editorista löytyy hyvin vastaavia ominaisuuksia kuin Abitista. Esimerkin 4.2 tehtävänanto ja sitä vastaava Esimerkissä 4.4 esitetty malliratkaisu on esitetty MATLABin Live Editorilla kirjoitettuna seuraavassa kuvassa.

Ratkaise Van der Polin yhtälösysteemin

$$\begin{cases} \dfrac{\mathrm{dx}}{\mathrm{dt}} = y \\ \dfrac{\mathrm{dy}}{\mathrm{dt}} = -x + \left(1 - x^2\right) y \end{cases}$$

ominaisarvot käyttäen approksimaatiota $x^2 y = 0$.

Kun approksimoidaan, että $x^2 y = 0$, saadaan Van der Polin yhtälösysteemi muotoon

$$\begin{cases} \dfrac{\mathrm{dx}}{\mathrm{dt}} = y \\ \dfrac{\mathrm{dy}}{\mathrm{dt}} = -x + y \end{cases}$$

Saadusta yhtälösysteemistä voidaan muodostaa matriisi **A**, jonka avulla saadaan ratkaistua ominaisarvot λ

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}$$

Ratkaistaan ominaisarvot λ yhtälöstä

$$\begin{aligned}
& det\left(\boldsymbol{A} - \boldsymbol{I}\lambda\right) = 0 \\
& \Leftrightarrow \begin{vmatrix} 0-\lambda & 1 \\ -1 & 1-\lambda \end{vmatrix} = 0 \\
& \Leftrightarrow \begin{vmatrix} -\lambda & 1 \\ -1 & 1-\lambda \end{vmatrix} = 0 \\
& \Leftrightarrow -\lambda\left(1 - \lambda\right) + 1 = 0 \\
& \Leftrightarrow \lambda^2 - \lambda + 1 = 0 \\
& \lambda = \frac{1 \pm \sqrt{1^2 - 4 \cdot 1 \cdot 1}}{2 \cdot 1} \\
& \Leftrightarrow \lambda = \frac{1 \pm \sqrt{-3}}{2} \\
& \Leftrightarrow \lambda = \frac{1 \pm i\sqrt{3}}{2}
\end{aligned}$$

Eli ominaisarvot ovat

$\lambda_1 = \dfrac{1 + i\sqrt{3}}{2}$ ja $\lambda_2 = \dfrac{1 - i\sqrt{3}}{2}$

Kuva 4.6 *MATLABin live-editorilla kirjoitettu tehtävänanto ja tehtävän ratkaisu*

Kuvassa 4.6 näkyvät tehtävänanto ja tehtävä tehtiin käyttämällä live-editoria WYSIWYG-editorina. MATLABin Live Editorissa kaavoja ja merkkejä on valittavissa enemmän kuin Abitissa ja niitä on useissa eri valikoissa. Siellä on myös kuvake, josta saa matriisin. Suurempi kaavojen ja merkkien määrä hidastaa kuitenkin oikean merkin löytämistä jonkun verran, joten nopeuteen ja helppouteen pyrkivästä Abitin kaavaeditorista on varmasti tietoisesti tehty suppeampi. Muuten nämä kaksi järjestelmää olivat melko samanlaisia. Abitin käytön oppineiden on varmasti helppoa siirtyä käyttämään MATLABin Live Editoria, sillä yhden editorin käytön oppiminen helpottaa todella paljon seuraavan editorin käytön oppimista[14].

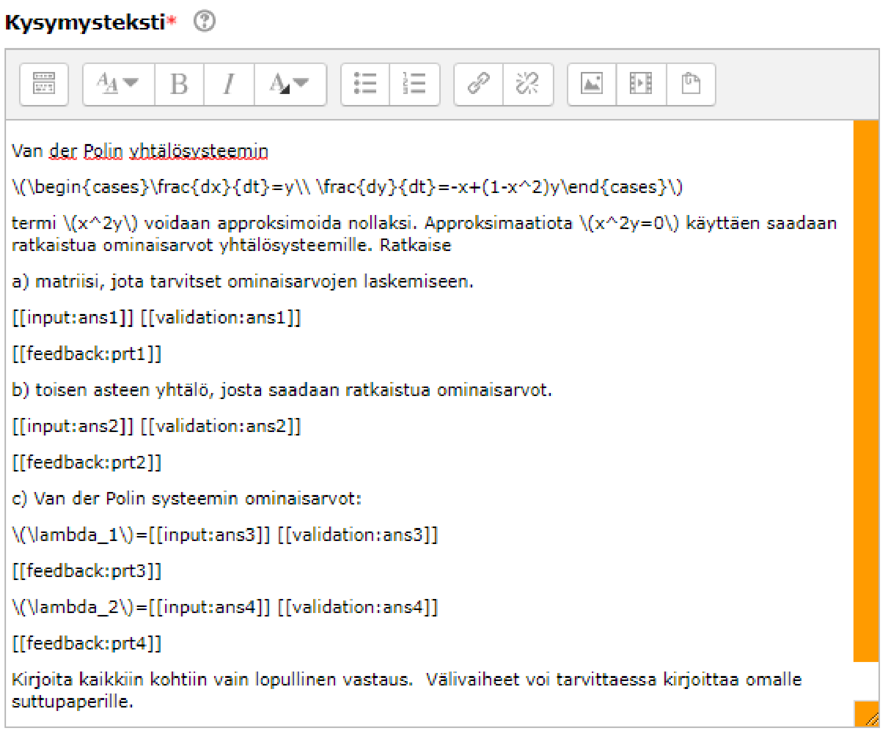
4.6 Stack-järjestelmä

STACK (a System for Teaching and Assessment using a Computer algebra Kernel) on avoimen lähdekoodin ohjelmisto, jonka on kehittänyt Chris Sangwin [15, s. 102—104]. STACK käyttää symbolisten ja numeeristen ilmaisujen käsittelemiseen Maximaa, joka on avoimen lähdekoodin matematiikkaohjelmisto. Matemaattisen tekstin kirjoittaminen onnistuu Maximassa LaTeX-koodikielellä. [16, s. 14]. STACK pyrkii tarjoamaan hyödyllisen ja luotettavan arviointijärjestelmän matematiikan tehtäville. Järjestelmän avulla pystytään arvioimaan ja antamaan palautetta siihen syötettyjen tehtävien ratkaisuista. Lisäksi järjestelmä kerää ylös ratkaisijoiden kaikki ratkaisuyritykset, joten sen avulla pystyy keräämään tietoa tyypillisimmistä virheistä, joita tapahtuu tiettyä tehtävää ratkaistaessa.[15, s. 102—104] STACK-järjestelmän keräämien ratkaisuyritysten avulla pystytään muokkaamaan tehtävänantoja ja antamaan täsmällisempiä vihjeitä, jotta ratkaisija onnistuisi tehtävänratkaisemisessa. STACK-järjestelmän automaattinen arviointi ja palautteen anto helpottavat myös monella tapaa tehtävien tarkastajien töitä.

STACK-järjestelmän avulla voidaan myös luoda satunnaisuutta tehtäviin. Satunnaisuuden ansiosta malliratkaisuja ei voi jakaa tehtävän ratkaisijoiden kesken, vaan jokaisen on itse ratkaistava tehtävä. [15, s. 102—104] Satunnaisuuden luominen onnistuu, kun tehtävän tekijä määrittelee tehtävässä käytettävät muuttujat sellaisiksi, että ne saavat vaihdella satunnaisesti tietyllä välillä. Työssä esimerkkinä käytettyissä tehtävissä satunnaisuuden hyödyntäminen ei onnistu, koska käsitellään tiettyä yhtälöä. STACK-järjestelmään ei myöskään ole tarkoituksenmukaista syöttää pitkiä ratkaisuja, jotka pystyisi ilmaisemaan usealla eri tapaa, koska tällöin automaattitarkastuksen luominen olisi mahdotonta. Sen vuoksi luodaan STACK-järjestelmää käyttäen vai Esimerkin 4.1 mukainen tehtävä. Työssä keskitytään vain tehtävänannon luomiseen ja tehtävän ratkaisemiseen STACK-järjestelmällä. Lisäksi hyödynnetään STACK-järjestelmän automaattitarkastus ominaisuutta.

4.7 Tehtävänannon tekeminen ja automaattitarkastus STACK-järjestelmällä

STACK-järjestelmällä tehtävien luominen vaatii jonkin verran Maximan ja LaTeX-ohjelmiston tuntemista STACK-järjestelmän tuntemisen lisäksi. Monimutkaisten tehtävien teon oppiminen voi viedä aikaa, mutta yksinkertaisten tehtävien tekemisen oppiminen on mahdollista nopeastikin [16, s. 16]. Esimerkin 4.1 tehtävänanto on melko yksinkertainen, joten sen kirjoittaminen STACK-järjestelmällä sujuu kohtalaisen helposti. Seuraavassa kuvassa on esitetty kuvakaappaus STACK-tehtävien luomislomakkeen kohdasta, johon tehtävänanto on kirjoitettu käyttäen koodikieltä.

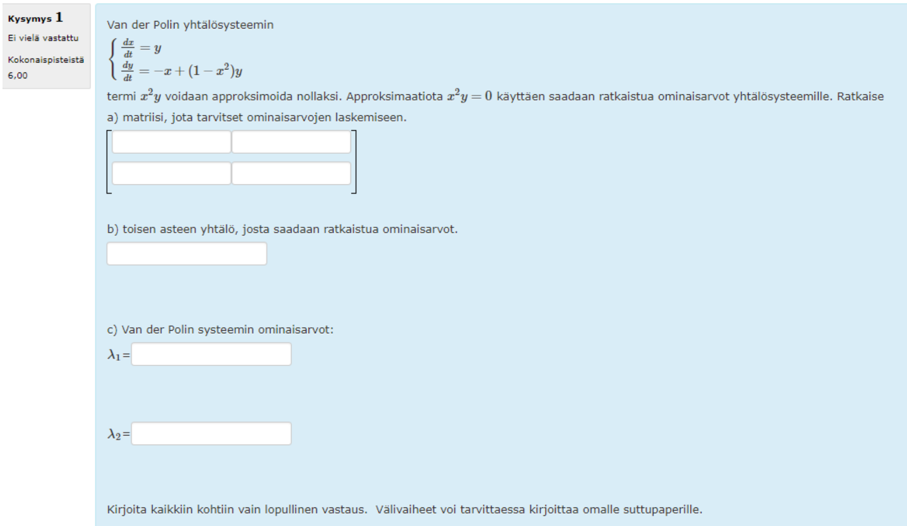


Kuva 4.7 *STACK-tehtävän tehtävänanto koodikielellä*

Kuvan 4.7 tehtävänannossa näkyvien kaksinkertaisten hakasulkeiden sisällä olevat tekstit viittaavat myöhemmin tehtävänantolomakkeessa määritettyihin vastauskohtaisiin lisätietoihin, joissa määritellään muun muassa vastauksen tyyppi ja vastauskentän koko. Lisäksi tehtävään käytettävät muuttujat ja vastaukset on kirjoitettava erilliseen 'Tehtävän muuttujat' -kohtaan, jotta pystytään käyttämään automaattitarkastus ominaisuutta ja satunnaisuutta. Satunnaisuutta ei kuitenkaan esimerkkinä käytetyssä tehtävässä voi hyödyntää, eikä tehtävässä ole määrittelyä kaipaavia yksittäisiä muuttujia, joten 'Tehtävän muuttujat' -kenttään tulee vain kaikkien kohtien vastaukset automaattitarkastusta varten.

Automaattitarkastusta varten luodaan myös vastauspuu, jossa määritellään, milloin vastaus tulee hyväksyä. Halutessaan vastauspuun yhteyteen voi kirjoittaa myös tehtävästä annettavan palautteen. Automaattitarkastusta luodessa on otettava huomioon, kuinka monella eri tapaa vastauksen voi kirjoittaa oikein, jotta automaattitarkastus hyväksyy kaikki vähän eri tavoin, mutta silti oikein, kirjoitetut vastaukset. Lisäksi voi luoda vihjepolun, joka ohjailee ratkaisijaa oikeaan ratkaisuun pääsyssä. Tehtävänannon tekijän täytyy kuitenkin määrittää, missä kohdassa vihjeitä annetaan ja millaiset vihjeet ovat sopivia. Tämä vaatii usein tehtävänannon tekijältä tietoa tai jopa arvauksia tyypillisimmästä virheistä. Jos tehtävänanto tehdään ensimmäistä kertaa, eikä ole saatavilla tietoa tyypillisimmistä virheistä ratkaisussa, ei vihjepolusta välttämättä saada tarpeeksi kattavaa. Usein vihjepolku muotoutuu paremmaksi vasta, kun tehtävän ratkaisemiseen liittyvistä tyypillisimmistä virheistä saadaan tietoa. Luodut STACK-tehtävät voi myös tallentaa myöhempää käyttöä varten, joten hyväksi havaittuja tehtäviä ei tarvitse joka kerta luoda uudestaan. Lisäksi tehtäväntekijä voi helposti täydentää tehtäväänsä havaitsemiaan puutteita.

Seuraavassa kuvassa on puolestaan kuvakaappaus ratkaisijalle näkyvästä tehtävänannosta.

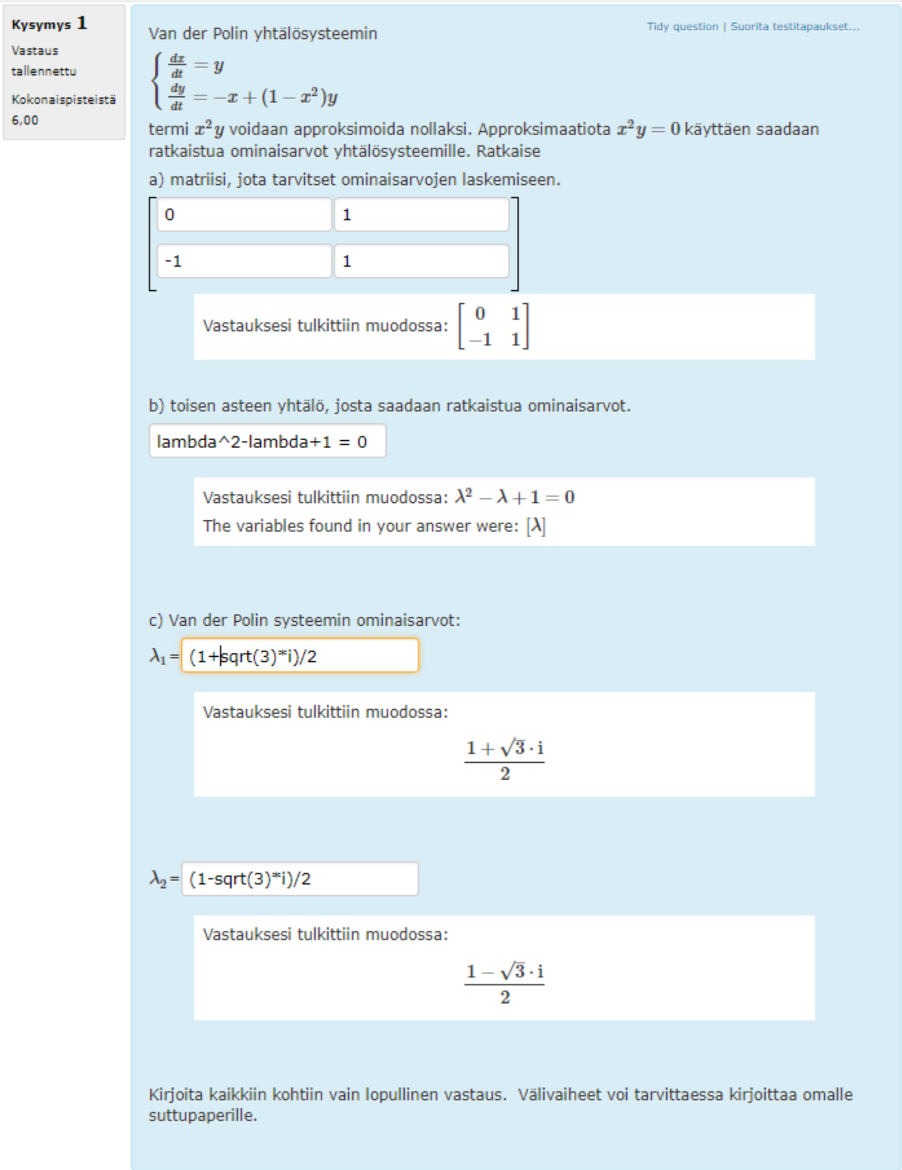


Kuva 4.8 *STACK-tehtävän ratkaisijalle näkyvä versio*

Koska STACK-järjestelmässä tehtävänratkaisu tarkastetaan automaattisesti, on se oltava yksiselitteisesti ratkaistava. Ratkaisijalle näkyvä Kuvassa 4.8 olevan version vastauskentät johdattelevat ainakin matriisin kohdalla hieman oikeaan ratkaisuun pääsyssä.

4.8 STACK-tehtävän ratkaiseminen

Kuvassa 4.8 olevan tehtävän ratkaisemiseksi täytyy osata kirjoittaa tiettyjä merkkejä ja symboleita Maximan syntaksia käyttäen. Ratkaisemiseen tarvittavia merkkejä ei kuitenkaan ole kovin montaa ja suurin osa niistä on useimmille tehtävän ratkaisijoille entuudestaan tuttuja muualta. Seuraavassa kuvassa on esitetty Kuvan 4.8 tehtävän ratkaisu. Siitä nähdään myös, mitä merkkejä täytyy osata kirjoittaa, jotta tehtävän saa ratkaistuksi.



Kuva 4.9 *STACK-tehtävän vastaus*

Jos Kuvan 4.9 mukaisen ratkaisun tekemiseen vaadittavien symbolien ja merkkien osaamista ei vaadita ennakkotietona tehtävän ratkaisijalta, voidaan ne kirjoittaa tehtävänannon yhteyteen.

4.9 Sähköisten järjestelmien vertailu

Kaikilla järjestelmillä onnistui esimerkkinä käytettyjen tehtävänantojen luominen. STACK-järjestelmässä Esimerkin 4.2 tehtävänantoon vastaaminen ei kuitenkaan olisi ollut mielekästä. Koska automaattitarkastus vaati, että tehtävä on ratkaistavissa lähes yksikäsitteisesti tai vain rajallisella määrällä vaihtoehtoja. Esimerkin 4.2 tehtävän olisi voinut ratkaista äärettömän monella eri tavalla, jos oletetaan, että näkyviin halutaan välivaiheet, jotka perustelevat sen, miten tiettyyn ratkaisuun on päädytty. Tämän vuoksi kyseiselle tehtävälle olisi ollut mahdoton luoda automaattitarkastusta. Muissa järjestelmissä molempien esimerkkien tehtävänantoihin vastaaminen onnistui, mutta jos sähköisiä järjestelmiä on käyttänyt vain vähän, on Esimerkin 4.1 tehtävä nopeampi ratkaista, koska sähköisesti kirjoitettavaa tekstiä on vähemmän. Seuraavassa taulukossa on esitetty eri sähköisten järjestelmien vaatimat koodikielet. Lisäksi siinä on eritelty, tarvitseeko kyseistä koodikieltä sekä tehtävänannon tekijä että tehtävän ratkaisija.

Taulukko 4.1 *Eri koodikielten osaamisvaatimukset tehtävänannon tekijältä ja tehtävänratkaisijalta Abitissa, MATLABin Live Editorissa ja STACK-järjestelmässä*

	Abitti	**MATLABin Live Editor**	**STACK**
LaTeX-koodikielen osaamista vaaditaan:			
Tehtävänannon tekijältä	KYLLÄ**	EI	KYLLÄ
Tehtävän ratkaisijalta	EI	EI*	EI
AsciiMath -koodikielen osaamista vaaditaan:			
Tehtävänannon tekijältä	KYLLÄ**	EI	EI
Tehtävän ratkaisijalta	EI	EI	EI
Maksima -koodikielen osaamista vaaditaan:			
Tehtävänannon tekijältä	EI	EI	KYLLÄ
Tehtävän ratkaisijalta	EI	EI	KYLLÄ*

* = Tehtävänannon tekijä voi tehtävänannollaan määrittää, tarvitaanko tehtävän ratkaisemiseen koodikieltä, joten riippuen tehtävästä vastaus voi olla myös päin vastainen.
**= vaihtoehtoiset keskenään, molempien osaamista ei tarvita.

Taulukosta 4.1 huomataan, että MATLABin Live Editor on ainut järjestelmä, jossa ei tarvitse osata mitään koodikieltä. Muissa järjestelmissä LaTeX tai AsciiMath -koodikielen osaaminen on tehtävänannon tekijälle välttämätöntä. STACK-järjestelmässä vaaditaan myös Maxima-ohjelmiston tuntemista. Ainoastaan STACK-järjestelmässä Maksiman syntaksin osaamista tarvitsee myös tehtävänratkaisija. Useimmissa tehtävissä sen syvällistä osaamista ei vaadita. Tietyt merkit on osattava kirjoittaa koodikielellä, koska STACK-järjestelmä on kolmesta tutkitusta järjestelmästä ainoa, jossa ei ole mahdollisuutta kirjoittaa WYSIWYG-editorilla. Seuraavassa taulukossa vertaillaan eri järjestelmien yleisiä ominaisuuksia.

Taulukko 4.2 *Abitin, MATLABin Live Editorin ja STACK-järjestelmän ominaisuuksien vertailu*

	Abitti	**MATLABin Live Editor**	**STACK**
Tehtävänannon kirjoittaminen helppoa	KYLLÄ	KYLLÄ	EI
Tehtävänannon testaaminen nopeaa	EI	KYLLÄ	KYLLÄ
Automaattitarkastus	EI	EI	KYLLÄ
Automaattinen vihjeiden antaminen	EI	EI	KYLLÄ
Automaattinen palautteen antaminen	EI	EI	KYLLÄ

Taulukosta 4.2 huomataan, että Abitin ja MATLABin Live Editorin ominaisuudet ovat todella samankaltaisia. Abitissa tehtävänannon testaaminen on hidasta koetilan palvelimen kautta, mutta tehtävänannon esikatseleminen on kuitenkin nopeaa. STACK-järjestelmässä testaaminen taas on nopeaa, mutta itse tehtävänannon kirjoittaminen on huomattavasti hitaampaa kuin Abitilla ja MATLABin Live Editorilla, koska siinä täytyy kirjoittaa muun muassa tehtävänannon ja vastausten muuttujat sekä määritellä vastaustyypit. STACK-tehtävät ovat ainoita, joissa on ominaisuutena automaattinen tarkastus, vihjeiden antaminen ja palautteen antaminen. Sillä tehtyjen tehtävien avulla pystytään siis vähentämään tarkastajan työmäärää ja antamaan kesken tehtävän ratkaisun vihjeitä tilanteen niin vaatiessa.

Seuraavassa taulukossa vertaillaan eri järjestelmien sopivuutta lukioon ja yliopistoon. Käytetään vertailussa asteikkoa 1-3, missä 1 tarkoittaa sitä, että järjestelmä ei sovellu käytettäväksi lukiossa tai yliopistossa, 2 tarkoittaa sitä, että järjestelmä soveltuisi kohtalaisen hyvin ja 3 tarkoittaa, että järjestelmä soveltuisi erittäin hyvin.

Taulukko 4.3 *Abitin, MATLABin Live Editorin ja STACK-järjestelmän sopivuus lukioon ja yliopistoon*

	Abitti	**MATLABin Live Editor**	**STACK**
Sopii lukioon	3	2	2
Sopii yliopistoon	1	3	2

Taulukosta 4.3 havaitaan, että lukioon sopii kaikki järjestelmät erittäin hyvin tai kohtalaisen hyvin. MATLABin Live Editor soveltuvat kohtalaisen hyvin, koska järjestelmä on melko samanlainen kuin lukiossa käytössä oleva Abitti. Siinä on kuitenkin käytössä huomattavasti lukion tarpeita enemmän erilaisia kaavoja ja merkkejä, jotka saattavat osaltaan hidastaa oikeiden kaavojen ja merkkejen löytämistä, jolloin ajatus nopeasta ja helposta kaavaeditorista ei välttämättä toteutuisi. Myös STACK-järjestelmän soveltuvuus lukioon on kohtalaisen hyvä. Tehtävien ratkaiseminen sen avulla onnistuisi varmasti lukiolaisilta ja STACK-järjestelmän avulla oppilaat voi-

sivat saada ratkaistavakseen sähköisiä harjoitustehtäviä, joiden ratkaiseminen olisi mahdollista juuri oppilaalle itselleen sopivalla hetkellä. STACK-järjestelmällä tehtävänantojen tekeminen on kuitenkin monimutkaisempaa kuin muilla järjestelmillä, joten sen käytön oppiminen vaatii opettajilta perehtymistä. Ainoastaan Abitti siis soveltuu lukioon erittäin hyvin, mutta siitäkin löytyy vielä toistaiseksi omat puutteensa. Kuitenkin uusimman päivityksen tullessa toukokuun alussa, alkaa vastauslomakkeen kaavaeditorissa olla lähes kaikki yleisesti lukiossa tarvittavat merkit ja kaavat, jolloin kyseinen kaavaeditori palvelee entistä paremmin lukion tarpeita. Yliopiston käyttöön Abitti olisi kuitenkin edelleen huono, koska nopeuteen ja helpouteen pyrkivä vastauslomakkeen kaavaeditori ei sovellu yliopistomatematiikan laskemiseen ja olisi turhan monimutkaista, jos joutuisi aina käyttämään myös 'libreoffice math' -editoria. Vastaavat ominaisuudet kuin vastauslomakkeen kaavaeditorista löytyvät myös MATLABin Live Editorista. Niiden lisäksi Live Editorista löytyy helposti myös muut usein yliopistomatematiikassa tarvittavat kaavat ja merkinnät, joten sen soveltuvuus yliopistomatematiikkaan on erittäin hyvä. STACK-järjestelmän ominaisuudet soveltuvat hyvin myös yliopistomatematiikan tehtävien ratkaisemiseen, mutta myös yliopistossa tehtävänantojen tekijöiden on perehdyttävä hieman monimutkaisempaan tehtävien tekojärjestelmään. Tämä vähentää hieman järjestelmän helppoa ja nopeaa käytettävyyttä.

5. YHTEENVETO

Työssä on tarkasteltu toisen kertaluvun differentiaaliyhtälöitä ja vertailtu niihin liittyvien tehtävien avulla kolmea sähköistä järjestelmää. Aluksi on käsitelty toisen kertaluvun differentiaaliyhtälöihin liittyvää yleistä teoriaa, jonka jälkeen on syvennytty tarkastelemaan epälineaaristen toisen kertaluvun differentiaaliyhtälöiden ratkaisemista. Koska niiden ratkaiseminen tapahtuu usein numeerisesti, eikä eksakteja ratkaisuja ole aina löydettävissä, tarkastellaan työssä esimerkkinä vain yhden epälineaarisen differentiaaliyhtälön eli Van der Polin yhtälön ratkaisemista. Van der Polin yhtälön ratkaisemista on käsitelty sekä linearisaation että keskiarvottamisteorian avulla.

Sähköisten järjestelmien eli Abitin, MATLABin Live Editorin ja STACK-järjestelmän vertailuun on käytetty esimerkkitehtäviä, joissa ratkaistiin Van der Polin yhtälön ominaisarvot. Työssä on vertailtu, millaista koodikielten osaamista tehtävänannon tekeminen vaatii sen laatijalta ja ratkaisijalta sekä sitä, millä tavoin tehtävien arviointi onnistuu eri järjestelmissä. Lisäksi pohdittiin järjestelmien soveltuvuutta yliopisto- ja lukiomatematiikan ratkaisemiseen. Seuraavana on esitelty vertailun tulokset.

Tehtävän laatijalta vaaditaan kaikissa muissa järjestelmissä paitsi MATLABin Live Editorissa joko LaTeXtai AsciiMath -koodikielen osaamista. Lisäksi STACK-järjestelmässä tarvitaan myös Maxima-ohjelmiston tuntemista. Maksiman syntaksin osaamista vaaditaan STACK-järjestelmässä myös tehtävän ratkaisijalta. Muissa järjestelmissä ratkaisu onnistuu useimmiten WYSIWYG-editoria käyttäen. Abitissa ja Live Editorissa ratkaisemisessa voidaan käyttää myös LaTeX-koodikieltä.

Abitin ja MATLABin Live Editorin ominaisuudet ovat hyvin samankaltaisia. Kummassakin järjestelmässä tehtävänannon kirjoittaminen on helppoa. Abitissa tehtävän esikatselu on nopeaa samoin kuin Live Editorissa. Tehtävän ratkaisujen testaaminen on kuitenkin Abitissa työlästä ja Live Editorissa nopeaa. Kummassakaan järjestelmässä ei myöskään ole mahdollista antaa vihjeitä tai palautetta automaattisesti. Tehtävän ratkaisun tarkastuskaan ei näissä järjestelmissä onnistu automaattisesti. STACK-järjestelmässä puolestaan on mahdollista antaa automaattisesti vihjeitä ja palautetta. Siinä myös tehtävän tarkastaminen onnistuu automaattitarkastusominaisuutta käyttäen. STACK-järjestelmässä tehtävänannon esikatselu on helppoa, mutta sen kirjoittaminen monimutkaisempaa.

Järjestelmistä lukioon soveltuu parhaiten Abitti, mutta myös muiden järjestelmien soveltuvuus lukioon on kohtalaisen hyvä. Live Editor on hyvin samankaltainen kuin Abitin vastauslomakkeen kaavaeditori, mutta siitä löytyy turhan paljon kaavoja ja merkkejä, joiden osaamista ei lukiossa tarvita. Tämän takia Abitin vastauslomakkeen kaavaeditori suppeampine kaava- ja merkkivalikoimineen on parempi. Työssä käytettiin myös Abitissa olevaa toista kaavaeditoria, 'libreoffice math' -kaavaeditoria, koska yliopistotason tehtävien ratkaiseminen ei kokonaisuudessaan onnistunut pelkällä vastauslomakkeen kaavaeditorilla. Sen käyttäminen ei kuitenkaan ollut kovin nopeaa, koska siinä on paljon merkkejä ja kaavoja eri valikoissa. Lisäksi sillä tehdyt kaavat täytyy liittää vastauslomakkeeseen kuvankaappauksella, joka osaltaan myös hidastaa ratkaisun tekemistä. Tämän vuoksi Abitilla sopii parhaiten ratkaistavaksi tehtävät, jotka pystyy ratkaisemaan kokonaan vastauslomakkeen kaavaeditorilla. Koska vastauslomakkeen kaavaeditorista ei löydy kaikkia yliopistomatematiikan laskuharjoitustehtävien ratkaisemiseen tarvittavia kaavoja, se ei sovellu kovinkaan hyvin yliopistoon.

Myös STACK-järjestelmä soveltuisi kohtalaisen hyvin lukioon. Sillä voisi antaa helpommin sähköisesti ratkaistavia kotitehtäviä lukiolaisille kuin Abitilla. Lisäksi oppilas voisi ratkaista tehtävän juuri haluamallaan hetkellä ja STACK-tehtäviin luotavan vihjepolun avulla saada ratkaisemiseen myös vinkkejä, jos ratkaisemisessa tulee ongelmakohtia. Oppilas voisi myös heti nähdä, onko tehtävä oikein ja mahdollisesti pystyisi heti korjaamaan tekemänsä virheet. STACK-järjestemällä tehtävänantojen laatiminen ja tehtävien ratkaiseminen on kuitenkin vähän monimutkaisempaa kuin muilla järjestelmillä.

LÄHTEET

[1] C. Clapham, J. Nicholson, *The Concise Oxford Dictionary of Mathematics*, 4nd ed., Oxford University Press, 2009.

[2] P. Blanchard, R.L. Devaney, G. R. Hall, *Differential equations*, 4nd ed., Brooks/Cole Thomson Learning cop., 2012.

[3] Marios Tsatsos, *Theoretical and Numerical Study of the Van der Pol equation*, Aristotle University of Thessaloniki, 2006.[WWW], [viitattu 25.3.2018]

[4] Ylioppilastutkintolautakunta, *Mikä Abitti?*,[WWW], [viitattu 22.3.2018]. Saatavilla:`https://www.abitti.fi/fi/abitti/`

[5] Ylioppilastutkintolautakunta, *Matemaattiset kaavat koetehtävissä*, [WWW], [viitattu: 22.3.2018]. Saatavilla:`https://www.abitti.fi/fi/ohjeet/matemaattiset-kaavat/`

[6] Ylioppilastutkintolautakunta, *Matemaattisen tekstin kirjoittaminen Abitissa*, [WWW], [viitattu: 22.3.2018]. Saatavilla:`https://www.abitti.fi/fi/ohjeet/matemaattisen-tekstin-kirjoittaminen-abitissa/`

[7] Ylioppilastutkintolautakunta, *Kokeen laatiminen*, [WWW], [viitattu: 24.3.2018]. Saatavilla:`https://www.abitti.fi/fi/ohjeet/kokeen-laatiminen/`

[8] Ylioppilastutkintolautakunta, *Blogi, Abittiin lisää MAFYKE-välineitä*, 31.3.2017, [WWW], [viitattu: 24.3.2018]. Saatavilla:`https://www.abitti.fi/blogi/2017/03/abittiin-lisaa-mafyke-valineita/`

[9] Ylioppilastutkintolautakunta, *Kaavaeditorin kehitysversio*, versio 23.3.2018, [WWW], [viitattu: 24.3.2018]. Saatavilla:`https://math-demo.abitti.fi/`

[10] Ylioppilastutkintolautakunta, *Blogi, Uudessa Abitti-versiossa uudistunut vastauseditori ja lisää laitetukea*, 2.5.2017, [WWW], [viitattu: 24.3.2018]. Saatavilla:`https://www.abitti.fi/blogi/2017/05/uudessa-abitti-versiossa-uudistunut-vastauseditori-ja-lisaa-laitetukea/`

[11] Opetushallitus, *Lukion opetussuunnitelmanperusteet 2015*, 2015, [WWW], [viitattu: 24.3.2018]. Saatavilla:`http://www.oph.fi/download/172124_lukion_opetussuunnitelman_perusteet_2015.pdf`

[12] Miika Huhtanen, *Matematiikan oppimisen tukeminen MATLABin ja vuorovaikutteisten opetusohjelmien avulla*, Tampereen teknillinen yliopisto, 10/2017, [WWW], [viitattu: 24.3.2018]. Saatavilla:https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/25218/huhtanen.pdf?sequence=1&isAllowed=y

[13] Jesse Kela, *Ohjelmistot sähköisessä tenttimisessä*, Tampereen teknillinen yliopisto, 15.06.2017, [WWW], [viitattu: 24.3.2018]. Saatavilla:https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/24947/kela.pdf?sequence=1&isAllowed=y

[14] Marjo Mansén, *Helpon ja nopean editorin jäljillä (osa 2/2)*, Ylioppilastutkintolautakunta, 17.03.2017, [WWW], [viitattu: 25.3.2018]. Saatavilla: https://www.ylioppilastutkinto.fi/ajankohtaista/blogitekstit/382-helpon-ja-nopean-editorin-jaljilla-2-2

[15] Chris Sangwin, *Computer Aided Assessment of Mathematics*, OUP Oxford, 2013.

[16] Ari-Mikko Mäkelä, *Verkkotyökalut yliopistomatematiikan peruskursseilla*, Tampereen teknillinen yliopisto, 03/2016, [WWW], [viitattu: 24.3.2018] Saatavilla:https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/23768/Makela.pdf?sequence=1

[17] Ylioppilastutkintolautakunta, *Sähköinen ylioppilastutkinto - matematiikka*, 2016, [WWW], [viitattu: 22.3.2018]. Saatavilla:https://www.ylioppilastutkinto.fi/images/sivuston_tiedostot/Sahkoinen_tutkinto/fi_sahkoinen_matematiikka_28.11.2016.pdf

LIITE A: KUVAN 3.1 PIIRTÄMISEEN KÄYTETTY MATLAB-KOODI

```
%Kuva 3.1 Suuntakenttä ja ratkaisujen muodostama käyrä
%ratkaisujen muodostama käyrä
vdp1 = @(t,z) [z(2), (1 - z(1)^2)*z(2) - z(1)]';%määritellään van Der Polin yhtälö
[T,Z] = ode45(vdp1,[0 20],[2 0]);%määrää, millä arvoilla vpd1 ratkaistaan
figure(1)
plot(Z(:,1),Z(:,2),'-')
set(findobj(gca,'Type','line'),'Color','m','LineWidth', 2);
xlabel('x');
ylabel('y');
hold on

%Suuntakentän suuntavektorien piirtäminen
[x,y] = meshgrid(-3:0.25:3);%piirretään vektorit koordinaatistoon välille x=[-
3,3]ja y=[-3, 3]
U = y; %määritellään Van der Polin yhtälösysteemin ensimmäinen yhtälö
V = - x+(1 - x.^2).*y ; %määritellään Van der Polin yhtälösysteemin toinen yhtälö
L=sqrt(U.^2+V.^2); %vektorin pituus
quiver(x,y,U./L,V./L,0.6,'b');%piirretään vektorit
axis tight;
hold off
```

LIITE B: KUVAN 3.2 PIIRTÄMISEEN KÄYTETTY MATLAB-KOODI

```
%Kuva 3.2 Vaihekuva lähellä origoa
%määritellään x0 ja z0
x0=-0.2:0.1:0.2;
n=numel(x0);
z0=[x0;zeros(size(x0))];

%Määritellään ajanhetket, joilla Van der Polin yhtälö ratkaistaan
aika_taaksepain= 10:-0.05:0;
aika_eteenpain=0:0.05:30;

%Määritellään kuvaan taustaruudukko ja sopivat akselit
figure(2)
hold on;
axis equal;
grid on;
axis([-0.25 0.25 -0.25 0.25]);

for i=1:n
    vdp1 = @(t,z) [z(2), (1 - z(1)^2)*z(2) - z(1)]';
    [T, Z]=ode45(vdp1, aika_taaksepain, z0(:,i)'); %etsitään ratkaisut
    plot(Z(:,1),Z(:,2),'b-','LineWidth',2);%piirretään ratkaisut kuvaan

    %piirretään nuolet ja etsitään niiden paikat suoran y=-x kohdalta
    TF = find(islocalmin(abs(Z(:,1)+Z(:,2)),'MinSeparation',10));
    for j=1:numel(TF)
        v=Z(TF(j)+1,:)-Z(TF(j),:);
        v=-0.002*v/norm(v);
        quiver(Z(TF(j),1),Z(TF(j),2),v(1),v(2),'b','LineWidth',4)
    end

    [T,Z] = ode45(vdp1,aika_eteenpain,z0(:,i)');%etsitään ratkaisut
    plot(Z(:,1),Z(:,2),'b-','LineWidth',2);%piirretään ratkaisut kuvaan

    % piirretään nuolet ja etsitään niiden paikat suoran y=-x kohdalta
    TF = find(islocalmin(abs(Z(:,1)+Z(:,2)),'MinSeparation',10));
    for j=1:numel(TF)
        v=Z(TF(j)+1,:)-Z(TF(j),:);
        v=0.002*v/norm(v);
        quiver(Z(TF(j),1),Z(TF(j),2),v(1),v(2),'b','LineWidth',4)
    end
end

%muotoillaan kuvan akseleita ja nimetään ne
set(gca,'XTick',-0.2:0.2:0.2)
set(gca,'YTick',-0.2:0.2:0.2)
xlabel('x');
ylabel('y');
```

LIITE C: KUVAN 3.3 PIIRTÄMISEEN KÄYTETTY MATLAB-KOODI

```
%Kuva 3.3: Keskimääräinen amplitudi {$\tilde{r}$} ajan t funktiona
r0=1;
a=0.1;
t=(150: 225/1000: 225);
r=2.*exp(a.*t./2)./sqrt(-1+exp(a.*t)+4./r0.^2);
figure(3)
plot(t,r, '-k')
xlabel('Aika');
ylabel('Amplitudi');
```