Tampereen yliopisto

Pekka Mattila

TYYPILLISTEN VIRHEELLISTEN RATKAISUJEN TUNNISTAMINEN MATEMATIIKAN SÄHKÖISISSÄ TEHTÄVISSÄ

Informaatioteknologian ja viestinnän tiedekunta
Diplomityö
Toukokuu 2019

TIIVISTELMÄ

Pekka Mattila: Tyypillisten virheellisten ratkaisujen tunnistaminen matematiikan sähköisissä tehtävissä
Diplomityö
Tampereen yliopisto
Sähkötekniikan DI-tutkinto-ohjelma
Ohjaajat: Lehtori Terhi Kaarakka, Yliopistonlehtori Simo Ali-Löytty, Tohtorikoulutettava Elina Viro.
Toukokuu 2019

Opiskelumaailman yleisesti muuttuessa yhä enemmän verkkopohjaiseksi ja opintojen siirtyessä sähköisille alustoille myös matematiikan sähköisiä opetus- ja opiskelumenetelmiä kehitetään nopeasti.

Opiskelutehtävien tekeminen verkossa on useimmille opiskelijoille ainakin jossain määrin tuttua. Sähköiset tehtävät koetaan yleisesti mielekkääksi tavaksi suorittaa matematiikankin harjoitustehtäviä. Erityisesti mahdollisuus tehdä harjoituksia sitoutumatta tiettyyn harjoituspaikkaan ja aikaan koetaan mielekkääksi. Sähköisiä harjoituksia on tuotu perinteisten harjoitusmenetelmien rinnalla ja korvaamaan erityisesti perustehtävien harjoittelua, jolloin harjoitustilaisuuksissa voidaan keskittyä enemmän ohjausta vaativiin tehtäviin.

Harjoitustehtävien sähköistyminen avaa mahdollisuuden myös automaattiselle arvioinnille. Automaattisen arvioinnin edut ovat kiistattomat. Kun harjoitus- ja muiden tehtävien vastauksia voidaan arvioida automaattisesti, opetushenkilökunta voi suunnata enemmän resursseja varsinaiseen opetustyöhön.

Tässä työssä tarkastellaan uudenlaisella, automaattista yksilöllistä ohjausta ja matematiikan tehtävien välivaiheiden kirjoitusmahdollisuutta tarjoavalla MathCheck-kaavatarkastimella tehtyä matematiikan perustaitojen kertauskokonaisuutta. Aineistona käytettiin Tampereen teknillisen yliopiston perustaitotestin yhteydessä syksyllä 2017 sähköisesti tehdyistä kertaustehtävistä kerättyjä opiskelijoiden tekemiä ratkaisuja. Tarkoituksena on löytää yleisiä, toistuvia virheellisiä ratkaisuja, jotka voitaisiin arvostella yhdellä kertaa automaattisella palautteella.

Tutkimusaineistoa lajiteltiin ohjelmallisesti. Ohjelmallisen lajittelun tuloksia tarkasteltiin myös laskemalla virheitä käsin tulosten luotettavuuden varmistamiseksi ja sellaisten virheiden löytämiseksi, joita ohjelmallisesti ei voinut huomata.

Sähköinen matematiikan opiskelu on kehittymisvaiheessa, ja uudenlaisia järjestelmiä ja opiskelualustoja kehitetään jatkuvasti. Tässä tutkimuksessa käytetty ohjelma oli useimmille uusille opiskelijoille vielä vieras. Uusien ohjelmien käyttöönottoon liittyy aina epävarmuustekijöitä, ja tutkimuksen toisena mielenkiinnon kohteena on millä tavalla uudet opiskelijat onnistuivat pelkästään sähköisessä kertauksessa. Teknisistä näkökulmasta MathCheckin uusien ominaisuuksien sopivuus laajempaan käyttöön kiinnostaa sähköisten opiskelualustojen kehittäjiä.

Ohjelmallinen lajittelu todettiin tämän tutkimuksen tarpeisiin riittävän tarkaksi. Virheiden käsinlaskenta täydensi lopputulosta ja tarjosi lisänäkemyksen kertaustehtävien kehitystyöhön. Suurin osa opiskelijoista selviytyi sähköisestä kertauksesta hyvin. Uudenlaisesta ohjelmasta aiheutui tyypillisiä, esimerkiksi syntaksista johtuvia virheitä, mutta harjoiteltuaan useimmat saivat tehtävät lopulta oikein. Kertauksessa käytetty MathCheck-kaavatarkastin todettiin soveltuvaksi ainakin matematiikan perustehtävien harjoitteluun. Opiskelijat suhtautuivat ohjelman uusiin ominaisuuksiin pääosin positiivisesti. Positiivinen palaute kannustaa jatkamaan kertausohjelman kehitystyötä.

Avainsanat: Sähköinen opiskelu, automaattinen arviointi, MathCheck-kaavatarkastin, matematiikka,perustaitotesti

ABSTRACT

Pekka Mattila: Recognition of typical erroneous solutions in electronic exercises in mathematics.
Master's thesis
Tampere University
Master of Science in Electrical Engineering
Examiners: Lecturer Terhi Kaarakka, University Lecturer Simo Ali-Löytty, PhD student Elina Viro.
May 2019

As the academic world becomes more and more web-based incorporating electronic platforms, electronic learning methods in mathematics are also undergoing rapid development.

Performing learning tasks online is familiar to most students, and such tasks are generally regarded as a meaningful way of performing math exercises. In particular, the possibility of doing weekly exercises that are not limited to a specific place and time is considered meaningful. Electronic exercises have been introduced alongside the more traditional training methods, in particular to replace the performance of basic tasks, which would otherwise require more group work and individual guidance. Online tasks have also opened up the possibility for automatic assessment. The benefits of automatic evaluation are undeniable. When the answers to basic exercises and other tasks can be evaluated automatically, teaching staff can devote more of their resources to their real job of teaching.

This work examines revising basic sets of mathematical skills with a novel MathCheck equation chain checker that offers automatic individual guidance, and the possibility of writing intermediate steps and comments into electronic mathematical tasks. The material used in this study consists of electronically collected solutions made by students at Tampere University of Technology during the tests of basic mathematical skills held in autumn, 2017. The purpose was to find common, repetitive, erroneous solutions which could be assessed with automatic feedback in one go. The results of the software sorting were also backed up by counting the errors manually to ensure the reliability of the software sorting results, and to find errors that might be ignored by the software sorting.

The use of electronic systems to study mathematics is still under development, and new types of systems and study platforms are continuously being developed. The introduction of new software is always accompanied with uncertainty, and the program used in this study was unfamiliar to the majority of the new students. The second point of interest in this study is how well the new students succeeded in carrying out electronic assignments and revision. The suitability of MathCheck's new features is also of interest to technical developers of e-learning platforms for more widespread use.

The software sorting was found to be accurate enough for the purposes of this study, and the manual assessment complemented the end results and gave an insight into the development work needed for revision tasks. Most students did well in the electronic assignments. Although this novel program caused typical task-specific errors, such as syntax errors, after practice most of the tasks were finally carried out correctly. Thus, the MathCheck equation chain checker used in the revision tasks was found to be suitable for this kind on use. The students were mainly positive about the new features of the program, and such positive feedback indicates that the development of this revision program should be continued.

Keywords: e-learning, automatic assessment, MathCheck equation chain checker, equation chain checking, mathematics, mathematics test of basic skills

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä diplomityö on tehty Tampereen yliopiston tekniikan alan koulutuksen Matematiikan laboratoriolle. Työ liittyy Tampereen teknillisessä yliopistossa kesällä 2017 tehtyyn Matematiikkajumppa -projektiin, jossa tuotettiin sähköisesti toteutettu matematiikan perustaitojen kertausohjelma aiemman kertausohjelman rinnalle ja korvaajaksi.

Työ on ollut mielenkiintoinen, haastava ja antoisa. Työ on myös tekijäänsä opettanut. Haluan kiittää Matematiikan laboratoriota mahdollisuudesta työskennellä tämän projektin parissa. Erityiset kiitokset haluan osoittaa työni ohjaajille ja tarkastajille Lehtori Terhi Kaarakalle, Yliopistonlehtori Simo Ali-Löytylle ja Tohtorikoulutettava Elina Virolle.

Kiitokset kuuluvat eittämättä myös kotirintamalle Terhille, erityisesti työn viimeisen vaiheen aikana osoitetusta pitkämielisyydestä.

Tampereella, 23. toukokuuta 2019

Pekka Mattila

SISÄLLYSLUETTELO

KUVALUETTELO

TAULUKKOLUETTELO

OHJELMA- JA ALGORITMILUETTELO

LYHENTEET JA MERKINNÄT

A	Matriisi A
a_i	Yhtälön muuttujan kerroin
$\mathbf{a}$	Vektori
AMK	Ammattikorkeakoulu (engl. University of Applied Sciences)
AsciiMath	Matematiikan merkintäkieli
$\mathbb{C}$	Kompleksilukujen joukko
IMA	Insinöörimatematiikka, lyhenne insinöörimatematiikan opintojaksoille
IMA-123	Insinöörimatematiikka 123, opintojakso
ISO	Kansainvälinen standardointiorganisaatio (engl. International Organization for Standardization)
JYU	Jyväskylän yliopisto (engl. University of Jyväskylä)
K	Itseisarvoltaan suurin kokonaislukukerroin
KOM	Competencies and the Learning of Mathematics
MAT-1	Matematiikka 1, opintojakso
MathCheck	A tool for checking math solutions in detail
MathJax	Matematiikan ladonnan standardi
MATLAB	Matrix Laboratory (Matriisilaskentaan perustuva numeerinen laskentaohjelmisto)
MWG	Mathematics Working Group
n	Kokonaislukukerroin
P	Polynomi P(x)
$\mathbb{R}$	Reaalilukujen joukko
rref	Reduced Row Echelon Form (Redusoitu vaakariviporrasmuoto).
SEFI	European Society of Engineering Education
STACK	System for Teaching and Assesment using Computer algebra Kernel
TAU	Tampereen yliopisto (engl. Tampere University)
TIM	The Interactive Material (Dokumenttipohjainen pilvipalvelu interaktiivisten materiaalien tuottamiseksi).
TTY	Tampereen teknillinen yliopisto (Tampere University of Technology)
TUNI	Tampereen korkeakouluyhteisö (engl. Tampere Universities)
x_i	Reaalimuuttuja
$\mathbb{Z}$	Kokonaislukujen joukko
z_i	Kompleksimuuttuja

1 JOHDANTO

Yhteiskunnan teknologisoituessa nopeasti myös opetusteknologian kehittämiseen kiinnitetään kasvavaa huomiota. Suuri yksittäinen teknologinen muutosvaihe opetusmaailmassa saadaan päätökseen, kun ylioppilaskokeet muuttuvat kokonaan sähköisiksi vuoden 2019 kuluessa [100]. Samaan aikaan viimeisimmässä perusopetuksen opetussuunnitelmauudistuksessa [55] korostetaan tietoteknisen osaamisen merkitystä. Koko koulujärjestelmän uudistuessa etenkin teknillisten alojen yliopisto- ja ammattikorkeakouluopiskelija voi olettaa opiskelulta ainakin peruskurssien tasolla edistyksellisyyttä.

Matematiikan perusopetuksen kehittämisen merkittävä tutkimus- ja kehitysalue on ollut tietotekniikan hyödyntäminen opetuksessa [63]. Yksi esimerkki tehdystä kehitystyöstä on sähköiset laskuharjoitukset, joita on hyödynnetty jo vuosia ainakin Tampereen teknillisen yliopiston (TTY) ja Aalto-yliopiston matematiikan peruskurssien yliopisto-opetuksessa.

Sähköisillä harjoituksilla on monia etuja. Opiskelijat eivät ole enää sidottuja tiettyyn harjoitusaikaan tai paikkaan, vaan he voivat tehdä opintojakson kulloisenkin viikon tehtävät verkkoyhteydessä paikasta ja kellonajasta riippumatta. Opiskeluohjelma voi parhaassa tapauksessa antaa lisäksi vihjeitä ja kannustaa opiskelijaa opiskelijan tekemän ratkaisun perusteella. Ohjelmien uusilla ominaisuuksilla tehtäviä voidaan muokata yhä monipuolisemmiksi. Opettajan kannalta on suuri etu, että järjestelmä voi tarkastaa ja arvioida automaattisesti suuren osan peruskurssien rutiininomaisista tehtävistä. Lisäksi on havaittu, että sähköinen harjoitusjärjestelmä aktivoi myös heikoimpia opiskelijoita harjoitusten tekoon [63, 70, 71].

Matematiikan perustaitojen ja ylipäätään matemaattisen osaamisen heikentyminen on Euroopan insinöörikoulutusyhteisö SEFI:n raportin mukaan Euroopan- ja maailmanlaajuinen ilmiö. Ilmiöön havahduttiin 2000-luvun alussa, ja se on edelleen olemassa. Yliopistot euroopassa ja muualla maailmassa ovat reagoineet tilanteen korjaamiseksi esimerkiksi mukauttamalla kurssien vaatimuksia opiskelijoiden taitotasoon ja lisäämällä tukitoimia. Toteutetut toimet eivät kuitenkaan ratkaise perustavanlaatuisista matematiikan perustaitojen puutteista johtuvaa ongelmaa. [20, 80]

Insinööritieteiden ja matematiikan suhdetta kuvataan hyvästä syystä symbioosilla. Lähes kaikki tekniikan alat ovat jollain tavalla riippuvaisia matemaattisesta osaamisesta, ja toisaalta matematiikka on usein uudistunut uusien teknologisten sovellusten aiheuttamien ongelmien kautta.

Matematiikan perustaitojen opettelemiseen ja kertaamiseen onkin kiinnitettävä huomiota jo heti opintojen alkuvaiheessa. Opintojen sujumisen takaamiseksi matematiikan perustaitotesti on ollut TTY:llä pakollinen osa ensimmäisen matematiikan kurssin suoritusta [80]. Perustaitotesti tulee nähtävästi olemaan jossain muodossa osa matematiikan perusopintojen suoritusta myös nykyisessä Tampereen yliopiston tekniikan alan koulutuksessa.

Testissä heikosti menestyneet ohjataan Matematiikkajumppa -tukiopetusohjelmaan, jossa opiskelijat kertaavat ja täydentävät matematiikan perustaitojaan, kunnes ovat saavat suoritettua perustaitotestin hyväksytysti. Jumpassa opiskelijat harjoittelevat insinööreille tärkeää proseduraalista sujuvuutta samankaltaisilla tehtävillä kuin perustaitotestissä. [80]

Tässä työssä tarkastellaan TTY:n matematiikan perustaitotestin kertauksen yhteydessä syksyllä 2017 rajatulta opiskelijajoukolta kerättyjä sähköisesti tuotettuja ratkaisuja. Perustaitotestin sähköinen kertaaminen toteutettiin uudenlaisella MathCheck-kaavatarkastimella ja siihen perustuvalla kertausohjelmalla.

Antti Valmari aloitti Tampereen teknillisessä yliopistossa (TTY) MathCheck-kaavatarkastimen kehitystyön 2010-luvun alussa. MathCheck otettiin TTY:ssä kokeilukäyttöön syksyllä 2015. Ohjelma todettiin sopivaksi opetuskäyttöön, ja sen käyttö ja kehitystyö jatkuu. Tekninen kehitystyö siirtyi vuonna 2018 Jyväskylän yliopistoon.

Merkittävä uudistus verrattuna aiemmin käytössä olleisiin matematiikan sähköisiin opiskeluohjelmiin on tehtävän tarjoama välitön palaute ja mahdollisuus kirjoittaa tehtävään välivaiheita ja muistiinpanoja. Nyt tehtäviä voidaan tehdä sähköisesti ilman apuvälineitä ja -ohjelmia. Ohjelma tarkastaa vastauksen koko päättelyketjun oikeellisuuden. Jos tehtävä on oikein, ohjelma hyväksyy vastauksen. Jos tehtävässä on virhe, MathCheck tarjoaa opiskelijalle välittömän, ohjaavan palautteen [**valmari2016mathcheck**]. Opiskelija voi heti korjata virheen ja jatkaa tehtävän loppuun.

Työn aihealue on yliopistomatematiikan perustaitotason tehtävien automaattisen arvioinnin kehittäminen ja työssä tarkastellaan myös uusien opiskelijoiden valmiutta sähköisesti toteutettuun matematiikan opiskeluun.

Työssä tutkitaan,

- kuinka sähköisesti annetusta vastausjoukosta pystytään ohjelmallisesti lajittelemalla löytämään luotettavasti keskenään samaa tarkoittavat virheelliset vastaukset,
- minkälaisia opiskelijoiden tekemät virheelliset vastaukset ovat,
- kuinka valmiita uudet opiskelijat ovat MathCheck-kaavatarkastimella toteutettuun matematiikan opiskeluun,
- kuinka opiskelijat onnistuivat sähköisessä kertauksessa.

Työn luvussa 2 tarkastellaan työn pedagogista ja matemaattista taustaa ja luvussa 3 MathCheck-kaavatarkastinta ja uuteen sähköiseen opiskelukokonaisuuteen liittyen käytettävyyden käsitettä. Luvussa neljä täydennetään työn matemaattista osaa esittelemällä työssä käsiteltävien tehtävien kertoimien satunnaistamista. Luvussa 5 esitellään työssä tehtävän tutkimuksen käytännön toteutus. Työn tulokset esitellään luvussa 6. Lopuksi luvussa 7 tehdään yhteenveto työn tuloksista ja pohditaan tutkimuksen aikana tehtyjä havaintoja ja esitetään kehitysehdotuksia.

Tutkimuksessa kerätään matematiikan kertaustehtävistä opiskelijoiden tekemiä ratkaisuja ja lajitellaan niistä ohjelmallisesti keskenään ekvivalentit virheelliset toistuvat ratkaisut. Tarvittaessa opiskelijoiden tekemiä vastauksia tarkastellaan myös laskemalla virheitä käsin.

Virhelaskentojen perusteella tehdään päätelmät eri laskentamenetelmien soveltuvuudesta työn tarkoituksiin. Sekä virhelaskentojen että kyselytutkimuksen perusteella tehdään johtopäätökset opiskelijoiden onnistumisesta sähköisesti toteutetuissa tehtävissä.

2 TYÖN TEOREETTINEN TAUSTA

Mitä matematiikka on? Käsitys matematiikan olemuksesta riippuu yleensä vastaajan näkökulmasta. Esimerkiksi didaktikko ja matemaatikko tarkastelevat matematiikkaa eri tavalla, samoin matematiikan opettajalla ja opiskelijalla on usein erilainen käsitys matematiikasta ja sen merkityksestä.

Matematiikan ja didaktiikan tutkija Pehkonen [59] jakaa matematiikan käsitteen karkeasti *laskutaitoon* ja *ymmärtämiseen*. Ymmärtäminen lieneekin matematiikan opetuksen keskeinen tavoite. Esimerkiksi lukion opetussuunnitelman perusteissa "kannustetaan kehittämään luovia ratkaisuja" [54]. Kysymykseen mitä matematiikka on, Pehkonen esittää kolme keskeistä näkökulmaa matematiikkaan. Pehkosen mukaan matematiikka on sekä *työkalupakki*, *systeemi* että *prosessi*. [59]

Työkalupakki sisältää laskusäännöt ja kulloinkin tarvittavat toimenpiteet. Systeemikäsitys tarkoittaa muodollisia toimintasääntöjä, esimerkiksi alkeismerkkien käytön. Prosessinäkökulmasta matematiikka muovautuu toiminnassa, kukin luo oman matematiikkansa omien tarpeidensa ja mahdollisuuksiensa mukaisesti. [59]

2.1 Matemaattinen tieto ja ajattelu

Yrjönsuuren [101] mukaan matemaattisen tiedon muodollinen rakenne poikkeaa muista tiedon alueista. Myöhemmin virhetarkastelun yhteydessä matematiikkaa verrataan oppilaalle vieraaseen kieleen [67], jolla on oma semantiikkansa. Matematiikalla onkin sille ominainen rakenne ja oma, riippumaton symboliikka. Matematiikan käsitteet ja määritelmät ilmaistaan muodollisten rakenteiden ja symbolien avulla. Tarkasta rakenteesta johtuen matematiikan kieli on täsmällistä, luotettavaa, teoreettista ja abstraktia [101].

Symboleilla on keskeinen asema matematiikassa, ja niiden voikin ajatella vastaavan sosiaalisten kielten aakkosia. Symboleihin ja niistä muodostuviin merkkijonoihin voi tallentaa informaatiota tiiviissä ja täsmällisessä muodossa, niiden avulla voidaan ilmaista matemaattisia rakenteita, käsitteitä ja luoda uusia käsitteitä. [101]

Matematiikan symbolien, rakenteiden ja ilmaisujen tuntemista pidetään usein matemaattisena tietona. Määrittelyjen osaaminen yhdessä muiden tietojen kanssa johtaa matematiikan osaamiseen. Matemaattisena tietona pidetään myös erilaisia oppimista tukevia toimia, kuten lausekkeen muodostaminen tai yhtälön rakenteen ja ominaisuuksien tunnistaminen. [101]

Matemaattinen ajattelu ja tieto usein samaistetaan, vaikka niiden merkitys poikkeaa toisistaan. Matemaattista ajattelua opitaan matemaattisen tiedon avulla. Matemaattisen tiedon rakenne ilmenee matemaattisissa sisällöissä, kuten esimerkiksi yhtälöinä, geometrisinä ominaisuuksina tai matemaattisina malleina. Esimerkiksi rationaaliluvun rakennetta pidetään tietona, jonka oppimista uutena asiana voidaan pitää matemaattisena ajatteluna. [101]

Joutsenlahden [28] mukaan tiedon käsite jaetaan usein karkeasti kahteen tyyppiin, taidon oppimiseen ja tiedon ymmärtämiseen [28]. Tiedon tarkempaan määrittelemiseen käytetään usein konseptuaalisen ja proseduraalisen tiedon käsitteitä [21, 24, 28]. Tunnetut tiedon luonteen tutkijat Hiebert & Lefevre [24] ovat määritelleet konseptuaalisen tiedon tiedoksi riippuvuuksista. Proseduraalinen tieto (matematiikassa) on ensinnäkin yksittäisten symbolien ja niiden syntaktisen käytön hallintaa ja toisaalta matemaattisten ongelmien ratkaisemiseksi tarvittavien proseduurien hallintaa [24].

Joutsenlahti toteaa jaon olevan kuitenkin epätäydellinen [28]. Kaikkea tietoa ei voi luokitella kahteen luokkaan ja osa tiedosta voi kuulua useampaan luokkaan [28]. Seuraavassa luvussa tarkastellaan kahden suomalaisen tutkijan, Haapasalon [21] ja Ruohotien [75] määritelmiä konseptuaaliselle ja proseduraaliselle tiedolle.

2.1.1 Konseptuaalinen tieto

Haapasalo [21] on täydentänyt ja muokannut modernimmaksi aiempien tutkijoiden konseptuaalisen tiedon mallia. Haapasalon mukaan konseptuaalinen tieto on merkityksistä koostuva verkko, jonka solmuja ja yhteyksiä loogisesti toimiva yksilö pystyy tulkitsemaan ja muokkaamaan. Solmujen ja yhteyksien ei tarvitse olla objektiivisia, vaan ne voivat muotoutua yksilön subjektiivisten ajatteluprosessien, konstruktioiden, kautta. Ruohotie on aiemmin määritellyt konseptuaalisen tiedon samaan tapaan kuin Haapasalo käsitteiden ja asiatietojen linkkien muodostamaksi tietoverkoksi [75].

2.1.2 Proseduraalinen tieto

Proseduraalinen tieto koostuu Haapasalon mukaan säännöistä ja menetelmistä, kuten esimerkiksi algoritmien suorittamisesta tiettyjen *proseduurien* mukaan. Tehtävän tai algoritmin suorittamisen vaatima proseduuri voi olla sisäistynyt, jolloin eri esitystapoja ei tarvitse tietoisesti ajatella. [21]

Ruohotien [75] proseduraalisen tiedon määritelmä pitää sisällään samankaltaisia elementtejä. Hänen mukaansa proseduraalinen tieto yhdistää tiedot ja taidot, ja muistuttaa sääntöä tai reseptiä, joka on helppo palauttaa mieleen ja soveltaa tilanteen mukaan.

Käytännöllisesti ja tiiviisti ilmaistuna proseduraalinen tieto sisältää matemaattiset operaatiot ja algoritmit, joita tarvitaan ongelmien ja laskutoimitusten ratkaisuun. Proseduraalista tietoa voi harjoittaa kehittämällä laskurutiinia. Konseptuaalinen tieto puolestaan tarkoittaa käsitteiden ja periaatteiden ja niiden välisten suhteiden ymmärtämistä ja taitoa soveltaa niitä. Konseptuaalista tietoa voi kehittää vain käsitteiden merkitysten sisäistämisellä omien ajattelu- ja päättelyprosessien kautta. [85]

2.2 Työssä käsiteltäviä oppimiskäsityksiä

Tiedon käsittelyn ja oppimisen prosessit ovat monimutkaisia, ja niitä tarkastellaan yleensä erilaisten oppimiskäsitysten kautta. Pedagoginen tutkimus jakaa perinteisesti oppimiskäsitykset neljään pääluokkaan, behavioristiseen, kognitiiviseen, konstruktivistiseen ja kontekstuaaliseen.

Oppimiskäsitykset kuvaavat kulloinkin vallitsevia käsityksiä tiedon ja henkisten prosessien luonteesta ja ne kuvaavat myös yleisiä yhteiskunnan koulutukselle ja opetustyölle asettamia odotuksia. Oppimiskäsityksiin vaikuttavat myös pedagogisen tutkimuksen tulkintaperinne ja teoriat [73]. Esimerkiksi konstruktivistinen oppimiskäsitys edustaa Rauste-von Wrightin ja von Wrightin mukaan nykyisiä oppimiskäsityksiä ja pedagogisia uudistuksia [73].

2.2.1 Behaviorismi

Behavioristinen oppimiskäsitys on kehittynyt E.L Thorndiken (1874-1949) kehittämän yleisen oppimisteorian pohjalta. Teorian mukaan oppiminen tapahtuu tekojen seurasten kautta ja palkitsemiseen johtavat yritykset opitaan. 1900-luvun alun pedagogiikassa Thorndiken teoriaa sovellettiin yksilöiden välisten erojen mittaamiseen. Yksilöiden välisiä eroja pidettiin tuolloin pysyvinä ja mitattavina ominaisuuksina. Yksilöiden välisten erojen, esimerkiksi koulusuoritusten, mittaaminen oli helpointa tehdä tehtävillä, joiden tulokset voidaan ilmaista objektiivisesti numeroina. [73]

Behavioristinen oppimiskäsitys omaksui ja systematisoi Thorndiken kehittämää teoriaa noustessaan pedagogiikan valtasuuntaukseksi 1920-luvulla. behaviorismi katsoi oppimisen tapahtuvan sekä eläin- että ihmiskunnassa samalla tavalla: oppiminen ja käyttäytyminen voidaan palauttaa yksinkertaisiin osiin ja osista rakentuviin kokonaisuuksiin. [73]

Behavioristisessa pedagogiikassa opettajan rooli on auktoriteetin asemassa aktiivisesti valmistella ja jakaa tietoa. Oppiminen on passiivista tiedon vastaanottamista. Behavioristinen oppiminen on pohjimmiltaan reagointia ulkoiseen ärsykkeeseen ja *ärsyke-reaktio* -ketjujen muodostumista [65]. Opiskelijan motivoijana on esimerkiksi oikein suoritetusta tehtävästä seuraava palkkio tai muu positiivinen palaute. Opiskeltavan asian suuremmat kokonaisuudet on jaettu osatavoitteisiin ja edelleen pilkottu yleensä oppitunnin mittaisiin palasiin, jotka voidaan opettaa kenelle tahansa [65, 73].

Puhtaasti behavioristisesta näkökulmasta oppimistapahtuma nähdään ulkoisena prosessina, johon ei liity oppijan sisäisten henkisten prosessien tarkastelua [92]. Behavioristisella lähestymistavalla on kuitenkin opiskelumaailmassa puolustajansa, ja sitä pidetään tehokkaana lähestymistapana esimerkiksi pidemmälle edenneissä teknis-luonnontieteellisissä opinnoissa, jos peruskäsitteet osataan jo hyvin [40].

2.2.2 Konstruktivismi

Kaikkea oppimista ei kuitenkaan voitu selittää behaviorismin avulla. 1950-luvulla vastakohtana behaviorismille kehittynyt konstruktivistinen näkökulma korostaa informaation prosessointinäkökulmaa [41] ja antaa opiskelijalle vapautta päättää ja vastata omista opinnoistaan. Vastuusta seuraavan hallinnan ja itsenäisyyden kokemuksen katsotaan lisäävän opiskelumotivaatiota [65, 73, 92].

Konstruktivismin tausta on antiikin kreikassa ja niin sanotussa muistitaidon perinteessä, jossa oppimista ja muistamista tehostetaan systemaattisesti organisoimalla muistettava aines mielessä oppimisvaiheessa. Kun opittava aines kytketään mielikuvien verkkoon, sen muistaminen helpottuu. [73].

Konstruktivistisen näkemyksen mukaan oppiminen on, vastakohtana behaviorismin passiivisuudelle vastaanottamiselle, aktiivista henkistä työskentelyä ja käsitteiden rakentamista. Opiskelu nähdään prosessina, jossa opiskelija itse on avainasemassa. Konstruktivistiseen oppimiskäsitykseen liittyy myös merkittävinä tekijöinä oppimaan oppiminen ja uusiutuminen koko elämän ajan.

Taulukko 2.1. *Behavioristisen ja konstruktivistisen oppimiskäsityksen vertailu [37, 46].*

Oppimiskäsitys	**Behaviorismi**	**Konstruktivismi**
Opetuksen tavoite	Tiedon jakaminen	Ymmärtämisen tukeminen
Opiskelijan rooli	Passiivinen vastaanottaja	Aktiivinen toimija
Opettajan rooli	Tiedon valmistelija ja jakaja	Oppimiseen auttaja ja ohjaaja
Opetuksen luonne	Tiedon välittäminen	Oppimisprosessin tukeminen
Opetuksen sisältö	Ennalta annettu tarkka opetussuunnitelma	Määrittyy osin oppimisprosessissa
Tiedon rakentuminen	Osista rakentuu kokonaisuus	Opiskelija rakentaa itse oman tietonsa

Taulukossa 2.1 on koottuna behavioristisen ja konstruktivistisen opetus- ja oppimiskäsityksen merkittävimpiä ominaisuuksia ja eroja [29]. Behavioristisen mallin mukaan oppiminen on kokonaisuuksien rakentumista pienistä yksityiskohdista. Konstruktivistinen oppiminen rakentuu kokonaisuuksien hahmottamisen kautta yksityiskohtiin. Kun behavioris-

tinen opetus pyrkii paloittelemaan opiskeltavan aiheen pieniksi aihealueiksi, joista rakentuu esimerkiksi matematiikan tietoteorioita, konstruktivistinen opetus puolestaan rakentaa tietoteorioita pienistä osista [73] luoden linkkejä osien välille henkisten prosessien avulla.

Käsitys matematiikasta pelkkänä laskutaitona ja mekaanisten sääntöjen ja kaavojen soveltamisena on Haapasalon [22] mukaan nykyään riittämätön. Konstruktivistinen oppimiskäsitys on vaikuttanut vahvasti lukion opetussuunnitelmiin jo kymmeniä vuosia. Vuoden 2015 lukion opetussuunnitelman perusteissa [54] konstruktivistinen ote näkyy esimerkiksi lukiokohtaisen opetussuunnitelman laatimista ohjaavassa kappaleessa, jossa opetussuunnitelma laaditaan yhteistyössä koko henkilökunnan, opiskelijoiden ja heidän vanhempiensa kanssa [53, 54].

2.3 Matemaattisen osaamisen mallit

Matemaattista osaamista on kutsuttu eri aikakausilla hieman erilaisilla nimityksillä. Aiempi matematiikan didaktisen tutkimuksen keskeinen käsite oli ymmärtäminen [60], ja 2000-luvulla siirryttiin tutkimaan oppijoiden matemaattista kompetenssia ja taitoa [39]. Matemaattinen kompetenssi tarkoittaa hyvin lyhyesti ilmaistuna matemaattista taitoa tai osaamista. Käsite esitellään laajemmin kappaleessa 2.4.

Matemaattinen osaaminen on laaja käsite. Usein matemaattisen osaamisen ajatellaan olevan ajattelun ja tiedon yhdistävä prosessi. Seuraavaksi tarkastellaan matemaattista osaamista kahden tunnetun mallin, tehtävien luokitteluun tarkoitetun Wilsonin taksonomian [97] ja matemaattisen taitavuuden kuvailemiseen sopivan Kilpatrickin, Swaffordin ja Findellin mallin [38] kautta.

2.3.1 Kilpatrick, Swafford ja Findellin malli

Matemaattinen osaaminen on kykyä käyttää matematiikkaa monipuolisesti matemaattisen ongelmanratkaisun alueilla. Kilpatrick et al. [38] mukaan matemaattisen osaamisen malli koostuu viidestä matemaattisen osaamisen ominaisuudesta. Joutsenlahti [28] on esittänyt ominaisuuksien suomenkieliset nimet.

1. **Käsitteellinen ymmärtäminen** tarkoittaa matematiikan peruskäsitteiden, laskentaoperaatioiden ja matemaattisten relaatioiden ymmärtämistä.
2. **Proseduraalinen sujuvuus** liittyy käsitteelliseen ymmärtämiseen, ja voidaan ymmärtää taidoksi käyttää laskentaproseduureja sujuvasti.
3. **Strateginen kompetenssi** käsitteenä on lähellä matemaattisten ongelmien ratkaisukykyä ja ongelman ratkaisuun liittyvää ongelmanmäärittelyä ja oikean ratkaisustrategian löytämistä.
4. **Mukautuva päättely** on kykyä loogiseen ajatteluun ja yleisemminkin ongelmien selittämiseen ja ratkaisumenetelmien perusteluun.

5. **Yritteliäisyys** sisältää kokemuksen matematiikan ja matemaattisten taitojen merkityksestä ja voi toimia motivoivana tekijänä matemaattisten taitojen kehittämisen suhteen.

Kilpatrick et. al. [38] mukaan mallin viisi osatekijää muodostavat matemaattisen osaamisen perustan, ja usein mallin osa-alueita kuvataan kirjallisuudessa puuta pystyssä pitävänä juuristona tai viidestä säikeestä muodostuvana köytenä.

2.3.2 Wilsonin taksonomia

Wilson esitteli Wilsonin taksonomiana tunnetun, Bloomin [7] taksonomiaan perustuvan, pedagogisten tavoitteiden luokittelujärjestelmän. Wilsonin taksonomia luokitteli erityisesti matemaattisen osaamisen alueita. Wilsonin taksonomian jaottelu perustuu neljään perustaitotasoon. [97]

1. **Laskutaito** tarkoittaa operaatioiden muistamista ja laskentaproseduurien hallintaa.
2. **Ymmärrys** tarkoittaa laskentaperiaatteiden, matemaattisen rakenteen ja ratkaisuperiaatteiden yleistettävyyden hallintaa.
3. **Soveltaminen** edellyttää oikeiden tietojen muistamista ja sopivien laskuoperaatioiden käyttämistä.
4. **Analyysi** on luovan ja keksivän ongelmanratkaisun taso, joka matematiikassa voi edistää asioiden välisten yhteyksien ja yleistysten löytämistä.

Taksonomia on alun perin eliökunnan luokittelujärjestelmä [88], ja biologisesta lähtökohdastaan se on yleistynyt tarkoittamaan käytännön luokittelua yleisesti. Benjamin Bloomin kehittämä Bloomin taksonomia [7] on tunnettu luokitus oppimisen kannalta tärkeiden tieteellisten tavoitteiden, esimerkiksi osaamistavoitteiden määrittämiseksi [56]. Wilsonin taksonomiassa [98] arvioidaan opiskelijan ajatteluprosesseja matematiikan osa-alueilla hänen suoritustensa pohjalta [28].

2.4 Insinöörikoulutuksen matemaattisen osaamisen yhteiseurooppalainen viitekehys

Euroopan insinöörikoulutusyhteisö SEFI:n matematiikan työryhmä Mathematics Working Group (MWG) on työskennellyt vuodesta 1982 [15] luodakseen yhteiseurooppalaisen insinöörikoulutuksen opetussuunnitelman suuntaviivat. Ensimmäinen koottu opetussuunnitelmaa käsitellyt julkaisu tehtiin vuonna 1992 [5]. Suunnitelmaa päivitettiin vuonna 2002 [20], ja uusin suunnitelma on julkaistu 2013 [44]. Julkaisuissa määritellään matemaattisen kompetenssin käsite, keskeiset insinöörimatemaattisen osaamisen osa-alueet, yleiset matematiikan opetukseen sopivat menetelmät ja yleisiä arvioinnin suuntaviivoja.

Matemaattinen kompetenssi määritellään tanskalaisen Competencies and the Learning of Mathematics (KOM) -projektin [51] tavoin "kyvyksi ymmärtää, tehdä päätelmiä ja käyttää matematiikkaa eri konteksteissa ja tilanteissa, joissa matematiikkaa voidaan soveltaa". KOM-projekti määritteli tarkemmin kahdeksan osakompetenssia, jotka yhdessä muodostavat Matemaattisen kompetenssin käsitteen. SEFI [44] on osin muotoillut ja tarkentanut KOM -projektin määritelmiä, alkuperäiset määritelmät ovat lähteissä [51, 52].

- **Matemaattinen ajattelu** kuvaa kykyä ymmärtää ja arvioida, milloin matemaattinen lähestymistapa on hyödyllinen.
- **Matemaattinen päättely** on kykyä ymmärtää argumentointia ja itse tuottaa matemaattisia päättelyketjuja.
- **Matemaattinen ongelmanratkaisukyky** on kykyä muotoilla kysymys matemaattisena ongelmana ja käyttää matemaattisia työkaluja ongelman ratkaisemiseen.
- **Matemaattinen mallinnustaito** tarkoittaa kykyä työskennellä valmiilla malleilla ja tuottaa itse malleja.
- **Matemaattisten kokonaisuuksien hahmottamiskyky** on kyky valita sopivat matemaattiset esitystavat ja tarvittaessa vaihtaa toiseen, tilanteeseen paremmin sopivaan esitystapaan.
- **Matematiikan muotokielen ymmärtäminen** tarkoittaa kykyä ymmärtää ja tuottaa ilmaisuja matematiikan symbolien ja muodollisten ilmausten avulla.
- **Matemaattinen kommunikointikyky** on sekä suullinen että kirjallinen kyky ilmaista itseään matematiikan ilmaisukeinoilla.
- **Kyky käyttää matematiikan apuvälineitä**, esimerkiksi ohjelmistoja ja muita työkaluja.

Insinöörimatemaattisen osaamisen kannalta keskeiset osa-alueet on jaettu neljälle tasolle. Tasot ovat core zero, core level 1, core level 2 ja core level 3. Core zero sisältää olennaisimmat esitiedot, joita tarvitaan yliopistotason insinöörimatematiikan opinnoissa. Suomalaisessa koulujärjestelmässä core zero vastaa pääosin lukion pitkän matematiikan oppisisältöä. Tasolle core level 1 kuuluvat tiedot vastaavat Suomessa tekniikan alan yliopistokoulutuksessa matematiikan perusopintoja sovellettuna koulutusaloittain. Core level 2 -tason matematiikan sisällöt ovat syvällisempiä ja sisältävät koulutusalakohtaiset oleelliset aineopintotasoiset tiedot. Core level 3 opinnot ovat teoreettisempia, syventäviä opintoja, ja niitä opiskellaan yleensä vain sovelluskohtaisesti.

Edeltävissä luvuissa on tarkasteltu matematiikkaa pedagogisesta näkökulmasta. Kuten alussa todettiin, käsitykset matematiikasta riippuvat vastaajan näkökulmasta. Vastaavasti matematiikan opettaminen heijastelee sitä kontekstia, missä sitä opetetaan.

Pohjolainen et al. [63] korostavat proseduraalisten taitojen merkitystä insinöörikoulutuksessa. Proseduraalinen tieto ja siihen liittyvät matemaattiset perustaidot taidot liittyvät ongelmien ja laskutoimitusten ratkaisussa tarvittavien operaatioiden ja algoritmien hallintaan, laskurutiinin kehittämiseen.

Toinen työssä korostuva pedagoginen osa-alue on konstruktivistinen oppimiskäsitys. Konstruktivistisen näkemyksen mukaan oppiminen on opiskelijan aktiivista työskentelyä ja käsitteiden rakentamista. Työssä käytetyn kertausohjelman, MathCheck-kaavatarkastimen [4], perusfilosofia on konstruktivistinen, ja sen kehittäjän alkuperäinen ajatus [94] oli opiskelijoiden matemaattisen ajattelun kehittäminen sähköisen opetusohjelman avulla.

Työssä käsitellyt matemaattisen osaamisen mallit ja viitekehykset ovat opettajille työkalu, jota vasten matematiikan oppimista, opetusta ja opetuksen suunnittelua voidaan tarkastella.

2.5 Virheet matemaattisten tehtävien ratkaisuissa

Termillä *virhe* on useita erilaisia merkityksiä asiayhteydestä riippuen. Yleensä virhe tarkoittaa epäonnistumista jonkin asian suhteen tai virheellistä toimintaa. Tässä työssä virhe tarkoittaa lähinnä proseduraalisia virheitä matematiikan tehtävissä.

Virheiden etsimistä ja luokittelua niiden laadun perusteella kutsutaan *virhetarkasteluksi*. Virhetarkastelussa tarkastellaan kohderyhmän tekemiä virheellisiä ratkaisuja ja pyritään löytämään yleisiä virheellisiä käsityksiä ja etsimään syitä virheellisten toimintatapojen taustalla. Virheet pyritään tunnistamaan tutkimusaineistosta ja tutkimuksen tarkoituksesta riippuen luokittelemaan sopivalla tavalla.

Matematiikan opetuksen tutkimuksessa virhetarkastelulla on pitkä historia. Tutkijat aluksi Yhdysvalloissa ja Saksassa ovat etsineet matematiikan tehtävissä tehtyjä virheitä ja analysoineet virheiden syitä 1920-luvulta alkaen [10, 11, 67]. Matematiikan tehtävissä tehtyjä virheitä analysoivat tutkimukset keskittyivät etenkin alkuaikoina alempien kouluasteiden aritmetiikkaan. Vaikka virhetarkastelua matematiikassa tehdään nykyään laajalti ja useilla eri matematiikan osa-alueilla, tutkimuksen painopiste on edelleen ala- ja yläkoulun matematiikassa.

Suomalainen matematiikan virhetarkastelun tutkimusperinne on vielä melko nuorta. Suomessa virhetarkastelua on tehty ala- ja yläkoulumatematiikan lisäksi esimerkiksi neuropsykologian [68], lääkelaskennan [25, 90], ja viime aikoina myös yliopistomatematiikan opinnoissa [77, 89].

Yleinen tapa luokitella virheitä on jakaa ne systemaattisiin virheisiin ja satunnaisiin lipsahduksiin [9, 19, 93]. Lähihoitajien matematiikkaa tutkinut Huhtala [25] kuvaa opiskelijoiden tekemiä virheitä tahallisiksi ja tahattomiksi. Tahallinen virhe, tai "väärin osaaminen", tapahtuu kun oppilas/opiskelija uskoo toimivansa oikein, mutta lopputulos on väärä. Virhe on tahaton, kun lopputulos ei ole sitä mitä tarkoitettiin.

Virheitä on perinteisesti kerätty paperilla ja lajiteltu käsin. Käsin paperille kirjoitetun aineiston avulla on päästy analysoimaan sekä yksittäisiä virheitä että virheellisiä päättelyketjuja. Haittapuolena aineiston käsin kirjaaminen ja käsittely on varsin työlästä, ja tutkimusotos rajautuu käytännön syistä melko pieneksi.

Uudet sähköiset opiskeluympäristöt tarjoavat tutkijoille mahdollisuuden saada vastauksia melko helposti sähköisessä muodossa. Sähköisen aineistonkeruun etuja ovat mahdollisuus suurempaan aineistoon, käsittelyn nopeus ja suuri vastausmäärä, jolloin tilastollinen analyysi ja virheiden yleisten piirteiden löytäminen on mahdollista.

Ehkä tunnetuin sähköisissä opiskeluympäristöissä tehtyjen virheiden ja automaattisen arvioinnin tutkija on Sangwin tutkimuskumppaneineen [77, 78, 79]. Suomessa sähköisissä järjestelmissä tehdyissä testeissä löytyneitä virheitä ja automaattista arviointia on tutkittu Teknillisessä korkeakoulussa [70], Aalto-yliopistossa [71, 72, 89, 91] ja Helsingin yliopistossa [90].

2.5.1 Virheiden yleisiä piirteitä

Oppilaiden ajatteluprosessit ja virheellisten päätelmien syyt ovat yleensä näkymättömiä. Siksi matematiikassa opiskelijan ajatusprosesseja tutkitaan usein epäsuorasti analysoimalla opiskelijan laskutehtävissä ja kokeissa annettujen vastausten virheitä. [43]

Virhetarkastelussa pyritään yleensä löytämään virheiden yleisiä piirteitä. Useat tutkijat ovat etsineet ja tunnistaneet oppilaiden virheiden yleisiä syitä [14, 16, 43]. Greer ja Mulhern [43] ovat artikkeliinsa "Between the Ears" etsineet ja kirjanneet kirjallisuudesta löydettyjä virheiden yleisiä piirteitä:

- Virheet voivat olla yllättäviä, jos opettaja ei havaitse niitä heti, vaan vasta jonkin ajan kuluttua.
- Virheelliset toimintatavat eivät yleensä korjaannu itsestään..
- Virheet ovat systemaattisia tai satunnaisia. Satunnaiset virheet kertovat yleensä huolimattomuudesta tai lipsahduksista, ja niiden merkitys tutkimukselle on pieni. Eri syistä johtuvat virheet pitäisi pystyä tunnistamaan ja erottelemaan toisistaan.
- Usein virheet ovat järjettömiä, eli annettu vastaus ei mitenkään voi olla mahdollinen vastaus.

Brousseau [8] on tehnyt neljä edellistä listaa täydentävää havaintoa:

- Virheet johtuvat usein perustavanlaatuisesta väärinymmärryksestä.
- Virheitä syntyy, kun oppilas käyttää virheellisiä proseduureja.
- Opiskelijalla on vaikeasti havaittavia virheellisiä käsityksiä.
- Oppilaat muodostavat usein omia, epämuodollisia tapoja ratkaista ongelmia.

Karppinen [34] kutsuu viimeistä, epämuodollisia ratkaisutapoja koskevaa havaintoa "vääräksi yleistykseksi" ja Huhtala [25] miniteoriaksi. Miniteorialla Huhtala tarkoittaa laskusääntöä, jonka opiskelijat itse keksivät [25].

2.5.2 Näkökulmia virheiden tutkimiseen

Opiskelijan tekemä opiskelutyö konkretisoituu esimerkiksi kokeessa tai muussa arvioinnissa. Opiskelijan kannalta virheet eivät ole toivottuja, koska virheettömyys kuvaa kulloinkin tehdyn opiskelutyön onnistumista. Opettajan kannalta opiskelijan tekemät virheet kuvaavat opiskelijan ajatusmaailmaa [43, 89]. Virhetarkastelun avulla opettaja oppii opiskelijan virheistä ja saa työkaluja pedagogisesti mielekkään palautteen antamiseen opiskelijoille. Palautteen perusteella opiskelija voi parantaa suoritustaan.

Opiskelijan sisäinen ajatteluprosessi rajautuu tämän työn aihealueen ulkopuolelle. Työssä keskitytään tarkastelemaan opiskelijoille yhteisiä virheellisiä ratkaisuja, vaikka virheiden käsinkirjauksessa tarkastellaan myös yksittäisen opiskelijan tekemiä virheitä. Perusidea on kuitenkin molemmissa tapauksissa samanlainen.

Mulhern ja Greer [43] ovat virheiden yleisten piirteiden lisäksi koonneet kirjallisuudesta näkökulmia virheiden tutkimiseen:

- Suoraviivainen erityyppisissä laskutehtävissä tehtyjen virheiden etsiminen ja kirjaaminen. Tätä yksinkertaista lähestymistapaa on käytetty laajalti virhetutkimuksessa.
- Erityyppisten virheiden analysointi, luokittelu, oikeaan vastaukseen vertaaminen ja virheeseen johtaneiden syiden etsiminen.
- Virhemallin analysointi. Etsitään systemaattisia, satunnaisia ja/tai tehtävätyypille tyypillisiä virheitä.
- Edellisten vaiheiden perusteella tehtäviä voidaan rakentaa niin, että opiskelijoita johdatellaan tekemään virheitä. Yksilöllisiä virhemalleja tutkitaan, ja yritetään löytää virheiden taustalla olevat syyt. Edelleen laaditaan ongelmia, joiden ajatellaan tuottavan samankaltaisia virheitä.

Luettelin viimeisessä kohdassa virheisiin johdattaminen voi tarkoittaa käytännössä sitä, että kokenut opettaja tietää, missä kohdassa opiskelijat kussakin tehtävätyypissä yleensä tekevät virheitä. Harjoitustehtävät voidaan laatia niin, että opiskelija johdatellaan virheeseen ja kohdistetaan opiskelijan huomio virheeseen. Virheen syitä voidaan käsitellä ja yrittää muuttaa virheellistä ajatusmallia tai tuottaa kokonaan uusi. [43]

Virheitä voi syntyä monista syistä. Edellä olleissa tarkasteluissa virheiden syiden ajatellaan olevan opiskelijan ajatteluprosessissa tai epäonnistuneessa suorituksessa. Radatz [66] näkee asian yksilöllisiä virheitä laajemmasta näkökulmasta ja esittää virheluokittelun lähtökohdaksi edellisten lisäksi opetusprosessiin liittyvät tekijät (opettaja, opetussuunnitelma, ympäristö) sekä informaation käsittelyn ongelmat.

Informaation käsittelyyn liittyvä ongelma tarkoittaa tässä "kielellisiä" ongelmia, joissa matematiikka nähdään vieraana kielenä [1, 33, 62], hahmotusongelmia [99], puutteellisista esitiedoista aiheutuvia ongelmia, kaavamaisesta ajattelusta tai asioiden väärästä yhdistelystä aiheutuvia ongelmia [61] ja väärien teorioiden käyttöä [18]. Väärä teoria tarkoittaa samaa kuin Huhtalan [25] käyttämä miniteoria.

Kielellisiin ongelmiin viittaaminen tarkoittaa matematiikan kieltä, merkintätapoja, symboleja ja matematiikan erityissanastoa. Hahmotusongelmilla viitataan oppimateriaalin visuaalisuuteen, selkeyteen ja esitystapaan. Useissa tutkimuksissa onkin todettu esitystavan aiheuttavan suurempia vaikeuksia kuin itse matemaattinen ongelma. [66]

Opetusprosessiin liittyvät ongelmat voivat johtua yksittäisistä tekijöistä tai niiden kombinaatioista tai koko opetusprosessin epäonnistumisesta [66]. Myös sähköiset oppimisympäristöt voivat huonosti toteutettuna aiheuttaa ongelmia oppimistuloksissa. Silloin on kyse käytettävyydestä, jonka käsitteisiin tutustutaan sähköisten oppimisympäristöjen yhteydessä kappaleessa 3.1.

Edellä on eritelty virheiden yleisiä piirteitä, eri näkökulmia virheiden tutkimiseen ja lopuksi virheiden syitä. Löydetyistä virheistä ei ole kuitenkaan pedagogista hyötyä, ellei niistä pystytä rajaamaan hyödyllisiä, opiskelijan ajattelua kuvaavia virheitä mukaan tarkasteluun ja luokittelemaan niitä luotettavasti.

2.5.3 Virheiden analysointi ja luokittelu

Virhetarkastelun keskeinen ongelma on, miten virheitä voidaan kategorisoida tai luokitella luotettavasti. Useimmat virhetarkastelua hyödyntävät tutkimukset ovat käyttäneet erilaista näkökulmaa ja erilaista luokittelua [90]. Luokittelun luotettavuuden arviointia vaikeuttaa lisäksi se, että opiskelijan tekemä yksittäinen virhe voi johtua useammasta virhetekijästä [58, 66]. Virhetekijällä tarkoitetaan tässä virheen aiheuttavaa yksittäistä tekijää tai virheeseen johtavaa syytä.

Tella ja Lavonen [87] määrittelevät tutkimuksen luotettavuuden *validiuden*, *reliaabeliuden* ja *stabiiliuden* käsitteillä. Samoja kriteerejä voidaan käyttää myös virheluokittelun yhteydessä. Virheluokittelun tulee olla validi. Validius tarkoittaa tässä, että mittari mittaa oikeita asioita. Virheluokittelun tulee olla myös reliaabeli. Reliaabeli tulos on pysyvä, mittausmenetelmä ei anna satunnaisia tuloksia. Eri tutkijoiden tulee päätyä samanlaiseen luokitteluun mallin avulla. Luokittelun tulee olla myös kestää satunnaisvaihtelua, eli sen tulee olla stabiili [87].

Radatz [66] on etsinyt kirjallisuudesta virheiden luokittelutapoja ja löytänyt niistä yhteisiä piirteitä. Historiallisessa perspektiivissä tutkimuksen mielenkiinto on kohdistunut viiteen tavoitteeseen [66].

1. Etsitään ja listataan kaikki mahdolliset virheet.
2. Tutkitaan eri virheiden ilmenemisen tiheysjakauma eri ikäryhmissä.
3. Pyritään tunnistamaan erityisvaikeudet.
4. Yksittäisten virhetekniikoiden tunnistaminen ja niiden pysyvyys.
5. Tunnistetaan ja luokitellaan virheet.

On olemassa yleisiä luokittelumalleja, mutta yleensä luokittelu on aina tapauskohtaista.

Esimerkiksi murtolukujen ja lääkelaskennan virheiden luokittelussa on yhteisiä piirteitä, mutta samaa luokittelumallia ei yleensä voi suoraan soveltaa erityyppiseen ongelmaan esimerkiksi virheen merkityksellisyyden takia.

Tietokoneavusteisen arvioinnin yhteydessä on kehitetty sähköisen arvioinnin tarpeisiin luokittelumalleja. Gill ja Greenhow [17] ovat kehittäneet virhetarkastelun pohjalta yleisen virhetyyppien taksonomian. Taksonomia on esitetty Taulukossa 2.2. Taksonomia on kehitetty mekaniikan monivalintatehtävien perusteella, mutta se on tehty mahdollisimman yleiseksi ja monikäyttöiseksi. Luokittelun käyttöä voidaan taksonomian yleisyyden takia soveltaa tapauskohtaisesti ja tarpeen mukaan jättää pois tai lisätä luokkia [17].

Taulukko 2.2. *Gill ja Greenhow:n yleinen virheiden luokittelumalli [17].*

Virhetyyppi	**Luokittelu**
Oletus	On oletettu jotain, joka ei pidä paikkaansa.
Laskuvirhe	Ratkaisutapa on oikeanlainen, mutta ratkaisussa on tehty laskuvirhe.
Kopiointivirhe	Tehtävän arvot on kopioitu väärin.
Määritelmä	Jonkin termin oikeaa määritelmää ei tunnettu.
Kaava	On käytetty virheellistä kaavaa.
Väärät arvot	On käytetty vääriä arvoja esimerkiksi sijoitettaessa arvoja kaavaan.
Tieto	Tarvittavat tiedot ovat puutteelliset.
Metodi	On yritetty ratkaista käyttäen väärää taktiikkaa.
Mallintaminen	Tilannetta ei ole pystytty mallintamaan oikein.
Prosessi	On käytetty oikeaa metodia, mutta sen kaikkia vaiheita ei ole osattu.
Tulkinta	Kysymys on tulkittu väärin.
Trigonometrinen	On käytetty virheellisiä sinin, kosinin ja tangentin perusmääritelmiä.

2.6 Työn matemaattinen tausta

Kappaleessa esitellään työssä tehtävään virhetarkasteluun valitut tehtävätyypit ja niiden matemaattinen tausta. Virhetarkasteluun valittiin itseisarvotehtävä, ensimmäisen asteen yhtälötehtävä, toisen asteen yhtälötehtäviä ja trigonometristen yhtälöiden tehtävä.

Työssä käsitellyt tehtävät olivat reaalikertoimisia ja tehtävien ratkaisut kuuluvat reaalilukujen joukkoon.

2.6.1 Itseisarvotehtävät

Kertaustehtävien itseisarvotehtävät olivat helppoja perustehtäviä, joiden ratkaisemiseen riitti itseisarvon määritelmän hallinta. Itseisarvon määritelmät pohjautuvat lähteeseen [64].

Määritelmä 2.1. Reaaliluvun a *itseisarvo* on reaaliakselilla luvun a ja origon ei-negatiivinen etäisyys. Merkitään $|a|$,

$$|a| = \begin{cases} a, & \text{jos } a \geq 0 \\ -a, & \text{jos } a < 0. \end{cases}$$

Positiivisen reaaliluvun ja nollan itseisarvo on luku itse. Negatiivisen reaaliluvun itseisarvo on luvun vastaluku eli luku kerrottuna luvulla -1.

Määritelmä 2.2. Kompleksiluvun $z = x + iy$ *itseisarvo* määritellään

$$|z| = |x + iy| = \sqrt{x^2 + y^2}$$

Kompleksiluvun itseisarvoa kutsutaan myös *moduliksi*. Se tarkoittaa pisteen etäisyyttä nollasta.

2.6.2 Tehtävät polynomiyhtälöistä

Kappaleessa esitetyt määritelmät pohjautuvat lähteisiin [13, 35] ellei määritelmän yhteydessä muuta mainita. Yhtälö on kahden ilmaisun välinen yhtäsuuruus, joka sisältää vakioita ja yhden tai useampia muuttujia. Yksinkertaisimmat yhtälöt koostuvat vain tekijöidensä yhteen- ja kertolaskuista.

Tämän työn tutkimusosuudessa käsitellään yksinkertaisia ensimmäisen ja toisen asteen yhden muuttujan reaalisia polynomiyhtälöitä. Polynomiyhtälö koostuu summasta, jossa jokainen termi on vakion ja muuttujan ei-negatiivisen kokonaislukupotenssin tulo. Muuttujien arvo pyritään määräämään niin, että yhtälö toteutuu.

Määritelmä 2.3. Astelukua n oleva *reaalimuuttujan polynomi* on funktio
$P : \mathbb{R} \to \mathbb{R}$ muotoa

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0,$$

jossa kertoimet $a_0, a_1, a_2, \ldots a_n$ ovat reaalisia vakioita ja $a_n \neq 0$. Kompleksimuuttujan polynomi määritellään kuten reaalimuuttujan polynomi, mutta lähtöjoukko ja maalijoukko ovat kompleksilukujen joukko: $P : \mathbb{C} \to \mathbb{C}$ ja kertoimet $a_0, a_1, a_2, \ldots, a_n$ ovat kompleksisia vakioita.

Määritelmä 2.4. *Kompleksilukujen joukko* $\mathbb{C}$ koostuu järjestetyistä reaalilukupareista

$$\mathbb{C} = \{(a, b)|a \in \mathbb{R} \text{ ja } b \in \mathbb{R}\}$$

Lause 2.1. *Imaginaariyksikölle i on voimassa $i^2 = i \cdot i = -1$.*

Todistus. Todistuksessa käytetään kompleksilukujen kertolaskun ominaisuutta $z_1 z_2 = (a + bi)(c + di) = (ac + bd) + (ad + bc)i$ kun $i^2 = -1$. Tuloksena saadun kompleksiluvun reaaliosa voidaan kirjoittaa muotoon $a = c = 0$ ja imaginaariosa $b = d = 1$, jolloin $i^2 = i \cdot i = (0 + i)(0 + +i) = (0 \cdot 0 + 0 \cdot i) + (0 \cdot i + i \cdot i = 0 + (-1) = -1$. □

Kompleksiluvun reaaliosa $Re(z)$ sijoittuu kompleksitason vaakasuuntaiselle reaaliakselille kuten reaaliluvut, ja imaginaarinen osa $Im(z)$ pystysuuntaiselle imaginaariakselille. Kompleksilukua $(a + bi)$ kuvataan kompleksitasoon piirretyllä lukuparilla (a, b).
Lause 2.2. Algebran peruslause. *Kaikilla $n \geq 1$ astetta olevilla kompleksimuuttujan polynomeilla on, useampikertaiset nollakohdat huomioiden, n nollakohtaa, jotka voivat olla kompleksisia tai reaalisia.*

Todistus. Algebran peruslauseen todistus lukiolaiselle esitetään lähteessä [26]. □

Algebran peruslauseesta seuraa, että polynomi voidaan jakaa tekijöihin nollakohtiensa avulla.
Lause 2.3. *Polynomi P, joka on astetta $n \geq 1$, on jaollinen polynomilla $(z - z_1)$*

$$P(z) = (z - z_1)Q_{n-1}(z),$$

jossa z_1 *on polynomin P nollakohta ja Q on $n - 1$ asteen polynomi.*

Todistus. Lähteessä [35, 81] □

Polynomin jaollisuuden ja nollakohtien välillä on yhteys.
Lause 2.4. *n. asteen polynomi voidaan esittää nollakohtiensa avulla muodossa*

$$P(z) = a_n(z - z_1)(z - z_2) \dots (z - z_n),$$

jossa luvut $z_1, z_2, \dots, z_n$, ovat polynomin nollakohtia.

Todistus. Lauseen 2.3 mukaan jakolaskulla $P(z)/(z - z_1)$ saadaan polynomi $P(x) = (z - z_1)Q_{n-1}(z)$. Algebran peruslauseen mukaan Q_{n-1}:llä on nollakohta z_2. Edelleen jakamalla $Q_{n-1}/(z - z_2)$ päädytään yhtälöön $P(z) = (z - z_1)(z - z_2)Q_{n-2}(z)$. Näin jatkamalla päädytään yhtälöön $P(z) = (z - z_1)(z - z_2)...(z - z_n)Q_0(z)$, jossa polynomin Q_0 asteluku on nolla ja Q_0 on potenssin z^n kerroin $Q_0(z) = a_n$. □

Ensimmäisen asteen reaalikertoiminen yhden muuttujan yhtälö

Ensimmäisen asteen polynomi on affiini kuvaus $f(x) = a_1x + a_0$. Muuttujan x reaalikertoiminen ensimmäisen asteen polynomiyhtälö voidaan aina sieventää muotoon $a_1x + a_0 = 0$, missä vakio $a_1 \neq 0$, ja vakiot a_1 ja a_0 kuuluvat reaalilukujen joukkoon.

Lause 2.5. *Ensimmäisen asteen yhtälöllä on ratkaisu* $x = -\frac{a_0}{a_1}$.

Todistus. Määritelmän 2.3 mukaan ensimmäisen asteen polynomin $a_1x + a_0$ kerroin $a_1 \neq 0$, jolloin ensimmäisen asteen yhtälö $a_1x + a_0 = 0$ voidaan aina sieventää muotoon $x = -\frac{a_0}{a_1}$. □

Koulutehtävissä yhtälöt ovat harvoin sievennetyssä muodossa. Usein yhtälöt esitetään murtolukumuodossa, tai niissä verrataan kahta polynomia keskenään, esimerkiksi $a_1x + a_3 = a_2x + a_4$.

Toisen asteen reaalikertoiminen yhden muuttujan polynomiyhtälö

Määritelmä 2.5. Toisen asteen muuttujan x yhtälö on, tai se voidaan sieventää muotoon

$$a_2x^2 + a_1x + a_0 = 0,\ a_2 \neq 0.$$

Muotoa kutsutaan yhtälön *normaalimuodoksi*.

Lause 2.6. *Toisen asteen yhtälön ratkaisut saadaan ratkaisukaavalla*

$$x = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2a_0}}{2a_2}. \tag{2.1}$$

Todistus. Johdetaan toisen asteen yhtälön ratkaisukaava neliöksi täydentämällä □

$$\begin{aligned}
& a_2x^2 + a_1x + a_0 = 0, \\
\Leftrightarrow\ & a_2x^2 + a_1x && = -a_0 \\
\Leftrightarrow\ & 4a_2^2x^2 + 4a_2a_1x + a_1^2 && = a_1^2 - 4a_2a_0 \\
\Leftrightarrow\ & (2a_2x + a_1)^2 && = a_1^2 - 4a_2a_0 \\
\Leftrightarrow\ & 2a_2x + a_1 && = \pm\sqrt{a_1^2 - 4a_2a_0} \\
\Leftrightarrow\ & 2a_2x && = -a_1 \pm \sqrt{a_1^2 - 4a_2a_0}.
\end{aligned}$$

Ratkaisemalla tuntemattoman x suhteen saadaan ratkaisut

$$x_{1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2a_0}}{2a_2}. \tag{2.2}$$

Lause 2.7. *Toisen asteen yhtälön ratkaisukaavassa juurrettava eli diskriminantti* $D = a_2^2 - 4a_1a_0$ *määrää yhtälön ratkaisujen lukumäärän ja luonteen,*

Jos $D > 0,$ *niin yhtälöllä on kaksi erisuurta ratkaisua* x_1 *ja* x_2.

Jos $D = 0,$ *niin yhtälöllä on kaksinkertainen ratkaisu* $x_1 = x_2$.

Jos $D < 0,$ *niin yhtälön ratkaisut ovat kompleksisia.*

Todistus. Osoitetaan, että kun $D < 0$, yhtälön ratkaisut ovat kompleksisia

$$\begin{aligned}
&a_2x^2 + a_1x + a_0 = 0 \\
&\Leftrightarrow x^2 + \frac{a_1}{a_2}x = \frac{a_0}{a_2} \\
&\Leftrightarrow x^2 + 2\frac{a_1}{2a_2}x + \left(\frac{a_1}{2a_2}\right)^2 = \left(\frac{a_1}{2a_2}\right)^2 - \frac{a_0}{a_2} \\
&\Leftrightarrow x^2 + 2\frac{a_1}{2a_2}x + \left(\frac{a_1}{2a_2}\right)^2 = \left(\frac{a_1}{2a_2}\right)^2 - \frac{a_0}{a_2} \\
&\Leftrightarrow \left(x + \frac{a_1}{2a_2}\right)^2 = -\frac{4a_2a_0 - a_1^2}{4a_2^2}.
\end{aligned}$$

Merkitsemällä $\frac{4a_2a_0 - a_1^2}{4a_2^2} > 0$ ja soveltamalla yhteyttä $i^2 = -1$, saadaan

$$x + \frac{a_1}{2a_2} = \pm\sqrt{i^2\,\frac{4a_2a_0 - a_1^2}{4a_2^2}} = \pm i\sqrt{\frac{4a_2a_0 - a_1^2}{4a_2^2}} = \pm i\frac{\sqrt{4a_2a_0 - a_1^2}}{|2a_2|}.$$

Ratkaisemalla x saadaan

$$x = \frac{-a_1 \pm i\sqrt{4a_2a_0 - a_1^2}}{2a_2}.$$

□

Lause 2.8. *Toisen asteen reaaliselle polynomille* $P(x) = a_2x^2 + a_1x + a_0,\ a_1 \neq 0,$ *jonka juuret ovat* x_1 *ja* x_2*, on voimassa*

$$x_1 + x_2 = -\frac{a_1}{a_2}, \tag{2.3}$$

$$x_1x_2 = \frac{a_0}{a_2}. \tag{2.4}$$

Todistus. Olkoon x_1 ja x_2 toisen asteen yhtälön $P(x) = 0$ ratkaisut. Kaavojen (2.2), (2.3) ja (2.4) mukaan

$$x_1 = \frac{-a_1 + \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \text{ ja } x_2 = \frac{-a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2},\ a_2 \neq 0$$

$$x_1 + x_2 = \frac{-a_1 - a_1}{2a_2} = -\frac{a_1}{a_2},$$

$$\begin{aligned}
x_1 \cdot x_2 &= \frac{\left(-a_1 + \sqrt{a_1^2 - 4a_2a_0}\right)\left(-a_1 - \sqrt{a_1^2 - 4a_2a_0}\right)}{4a_2^2} \\
&= \frac{a_1^2 - \left(\sqrt{a_1^2 - 4a_2a_0}\right)^2}{4a^2} = \frac{a_1^2 - \left(a_1^2 - 4a_2a_0\right)}{4a_2^2} \\
&= \frac{a_1^2 - a_1^2 + 4a_2a_0}{4a_2^2} = \frac{4a_2a_0}{4a_2^2} = \frac{a_0}{a_2}.
\end{aligned}$$

□

2.6.3 Trigonometriset polynomiyhtälöt

Trigonometristen yhtälöiden yhteydessä käytetään reaalifunktioiden jaksollisuuden ja parillisuuden määritelmiä sekä trigonometristen funktioiden muunnosten ominaisuuksia. Kappaleen määritelmät pohjautuvat lähteeseen [13], ellei määritelmän yhteydessä muuta ole mainittu.

Määritelmä 2.6. *Reaaliarvoinen funktio* f on reaalilukujen joukossa määritelty funktio, joka yhdistää jokaisen lukujoukon x alkion täsmälleen yhteen reaalilukuun $f(x)$.

Määritelmä 2.7. Funktio on jaksollinen, jos on olemassa sellainen positiivinen vakio p, että $f(x) = f\left(x + n \cdot p\right)$, kaikille kokonaisluvuille $n = 1, 2, 3, \ldots$. [96]

Lause 2.9. *Trigonometriset funktiot $y = \sin(x)$ ja $y = \cos(x)$ ovat jaksollisia funktioita ja niiden jakso on 2π. Jaksollisuus voidaan ilmaista myös seuraavasti*

$$\begin{aligned}
\sin(x + n \cdot 2\pi) &= \sin(x), \\
\cos(x + n \cdot 2\pi) &= \cos(x),
\end{aligned}$$

kaikille kokonaisluvuille n.

Määritelmä 2.8. Funktiota f kutsutaan *parilliseksi*, jos kaikilla määrittelyjoukon arvoilla x on voimassa $f(-x) = f(x)$, ja *parittomaksi*, jos $f(-x) = -f(x)$.

Lauseen 2.9 perusteella sinifunktio on parillinen ja kosinifunktio pariton.

Lause 2.10. *Jaksollisuuden ja parillisuuden perusteella sini- ja kosinifunktiolle on voimassa on voimassa yhtälöt*

$$\cos\left(\frac{\pi}{2} - x\right) = \cos\left(x - \frac{\pi}{2}\right) = \sin\left(x\right), \tag{2.5}$$

$$\sin\left(\frac{\pi}{2} - x\right) = \sin\left(\frac{\pi}{2} + x\right) = \cos\left(x\right). \tag{2.6}$$

Todistus. Kosinin vähennyslaskukaavasta [96]

$$\cos\left(x+(-y)\right)=\cos(x)\cos(-y)+\sin(x)\sin(-y)$$

saadaan, että

$$\begin{aligned}\cos\left(\frac{\pi}{2}-x\right)&=\cos\left(\frac{\pi}{2}\right)\cos(-x)+\sin\left(\frac{\pi}{2}\right)\sin(-x)\\&=0\cdot\cos(x)+1\cdot\sin(x)\\&=\sin(x),\end{aligned}$$

joka todistaa lauseen 2.5.

Sijoittamalla yhtälöön 2.5 muuttujan x paikalle lauseke $\left(\frac{\pi}{2}-y\right)$ saadaan identiteetti

$$\sin\left(\frac{\pi}{2}-y\right)=\cos(y). \tag{2.7}$$

$\square$

3 MATEMATIIKAN SÄHKÖISET OPPIMISYMPÄRISTÖT

Sähköiset oppimisympäristöt ovat oleellinen osa nykyaikaista opiskelua oppilaitoksista ja opiskelun tasosta riippumatta. Esimerkiksi ylioppilaskokeet [100] ovat muuttuneet kokonaan sähköisiksi. Myös monet oppilaitosten pääsykokeet ja esivalinnat järjestetään vain sähköisesti, eikä esimerkiksi monimuoto-opiskelu olisi mahdollista perinteisin opetusmenetelmin [6]. Sähköisten oppimisympäristöjen etuna pidetään erityisesti joustavuutta ajan ja paikan suhteen, arvostelun ainakin osittaista automatisointia ja mahdollisuutta järjestää verkko-opintojaksoja [36].

Nykyisin yliopistomatematiikan perusopinnoissa on yleisesti käytössä STACK-järjestelmä. STACK perustuu alun perin Chris Sangwinin Birminghamin yliopistossa kehittämään laskuharjoitusten tarkastusjärjestelmään [70]. Uudempana tulokkaana sähköisten oppimisympäristöjen joukossa on TTY:llä kehitetty ja käyttöönotettu MathCheck-kaavatarkastin [**valmari2016mathcheck**, 4].

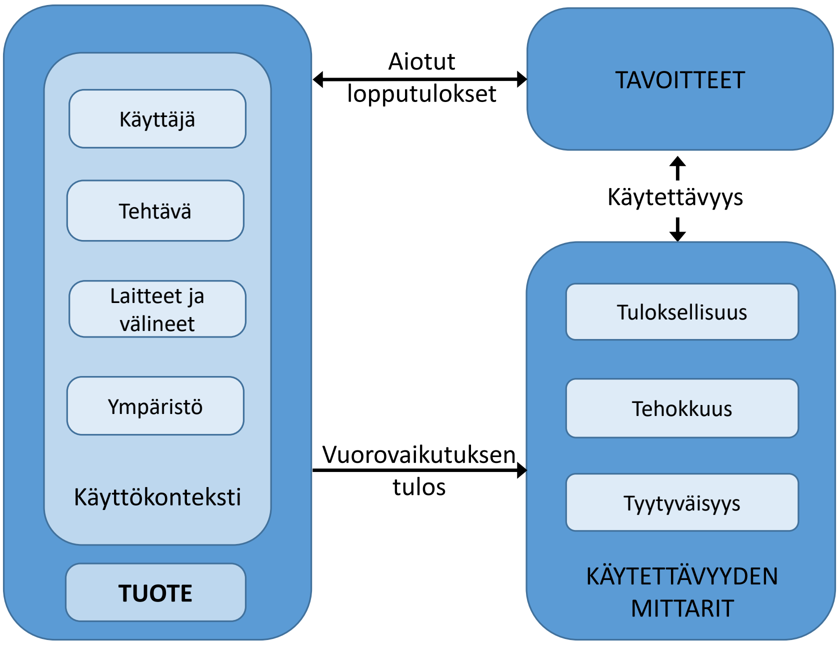
3.1 Oppimisympäristön käytettävyys

Tutkimuksessa käytettiin uudenlaista sähköisen oppimisen ympäristöä, joten on perusteltua tarkastella lyhyesti käytettävyyden käsitettä. Käytettävyyden yhteydessä mainitaan yleensä kaksi yleistä mallia: ISO 9241-11 [27] ja Nielsenin [48] malli. Vaikka käytettävyyskäsitteet voi yleistää koskemaan suunnittelutyötä yleisestikin, tässä työssä käytettävyyttä tarkastellaan sähköisten oppimisympäristöjen näkökulmasta.

ISO 9241-11 -standardin mukaan käytettävyyden kriteereinä on *vaikuttavuus*, *tehokkuus* ja käyttäjän tyytyväisyys [27]. Vaikuttavuus tarkoittaa sitä, miten hyvin käyttäjä saavuttaa tavoitteensa, esimerkiksi onko ohjelman käyttö edistänyt opiskelijan oppimista. Tehokkuus tarkoittaa ohjelman parissa käytetyn ajan ja koetun hyödyn suhdetta, eli oppiiko opiskelija ohjelman avulla paremmin verrattuna johonkin muuhun menetelmään. *Tyytyväisyys* vastaa käyttäjän kokemusta ohjelman käytöstä yleisesti. ISO 9241-11 mallin periaatekaavio kuvassa 3.1.

Kuvassa 3.1 mallin nuolet kuvaavat vaikuttavuutta ja tehokkuutta. Käyttäjätyytyväisyys kuvaa onnistumista järjestelmän eri osa-alueiden suunnittelussa.

Useimmissa käytettävyystutkimuksissa viitataan ISO 9241-11 [27]-mallin lisäksi Nielse-



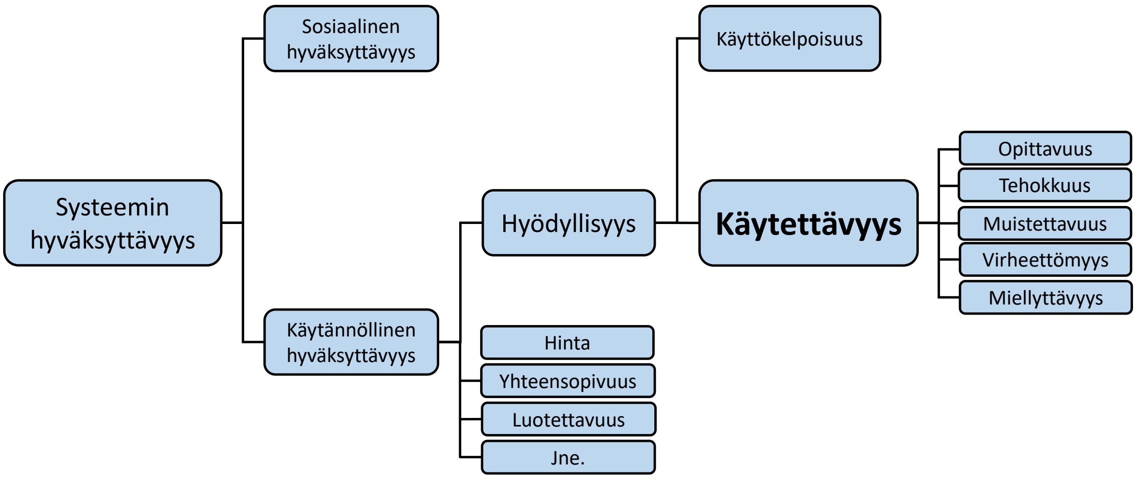
Kuva 3.1. *ISO 9241-11 käytettävyysmalli [27]*

nin [47] alun perin tietokoneistumisen alkuaikana julkaistuun käytettävyysteoriaan. Joskus julkaisuissa "Yleiseksi käytettävyysteoriaksi" tai "Nielsenin säännöiksi" nimettyä teoriaa voidaan soveltaa tietotekniikan ohjelmiin ja käyttöliittymiin yleisesti. Nielsenin [47] mallin periaatekaavio on kuvassa 3.2.

Käytettävyystekijät vaikuttavat käyttäjän kokemukseen ohjelmasta. Positiiviset kokemukset vaikuttavat käyttäjien haluun käyttää palvelua tai oppimisympäristöä. Jos käyttäjä kokee palvelun helppokäyttöiseksi, hän todennäköisesti käyttää palvelua myöhemminkin [49], ja jos palvelusta etsitty tieto ei ole saatavilla helposti, käyttäjä siirtyy toiseen järjestelmään [23]. Rice on tutkinut Moodle-ympäristöä ja todennut hyvän käytettävyyden ensivaikutelman vaikuttavan käyttäjän oppimiskokemukseen ja motivoivan opiskelijaa, kun taas huono kokemus turhauttaa, jättää negatiivisen vaikutelman ja oppimisympäristön hyväksi koetut ominaisuudet voivat jäädä löytämättä. [74].

Kaaviosta 1 nähdään, että Nielsenin malli on useampitasoinen. Nielsen [47] määritteli käytettävyyden ISO 9241-11 [27] -standardia laajemmin tuotteen *hyväksyttävyytenä*, jonka osatekijä käytettävyys on. Nielsenin mallin ensimmäinen taso sisältää *sosiaalisen* ja *käytännöllisen* hyväksyttävyyden. Sosiaalinen hyväksyttävyys viittaa esimerkiksi järjestelmän käyttäjistä keräämiin tietoihin [47].

Käytännöllinen hyväksyttävyys sisältää mallin toisen tason käsitteet *hyödyllisyyden* ja käytännön suunnitteluun liittyviä käsitteitä. *Käytettävyys* ja *käyttökelpoisuus* sisältyvät mallin toisen tason käsitteeseen hyödyllisyys. Nielsen erottaa käyttökelpoisuuden käytettävyydestä, koska käyttökelpoisuus selviää usein vasta järjestelmää käytettäessä [47]. Käytettävyyden osatekijöitä Nielsenin [47] mukaan ovat



Kuva 3.2. *Nielsenin käytettävyysmalli [47]*

- **Opittavuus** tarkoittaa, että järjestelmän käyttö on helppo oppia.
- **Tehokkuus** viittaa työskentelytehokkuuteen käytön oppimisen jälkeen.
- **Muistettavuus** ilmaisee, miten käyttäjä muistaa tuotteen käytön tauon jälkeen.
- **Virheettömyys** havaitaan käyttäjän virheiden määränä ja vakavuutena. Myös käyttäjän virheistä palautuminen kuuluu virheettömyyteen.
- **Tyytyväisyys** kertoo tuotteen käytön miellyttävyydestä ja siitä, miten helppo tuotetta on lähestyä.

Nielsen kokosi teoksessaan [47] hyväksyttävyyden ja käytettävyyden käsitteiden lisäksi käytettävyyden kymmenen periaatetta, joita voidaan soveltaa järjestelmän käytettävyyttä suunniteltaessa. Käytettävyyden suunnitteluperiaatteet esitetään tässä periaatetasolla, laajempi kuvaus Nielsenin teoksissa [47, 48].

1. **Yksinkertaisuus**: ylimääräinen informaatio voi häiritä kokenuttakin käyttäjää.
2. **Luonnollinen kieli**: käytetyn kielen pitää olla ymmärrettävää.
3. **Muistettavan minimointi**: tulisi käyttää tiedon valintaa tai muokkausta uuden tiedon sijaan.
4. **Yhdenmukaisuus**: kannattaa hyödyntää käyttäjälle tuttuja toiminta- tai merkintätapoja.
5. **Palaute**: palautteen nopeus on merkittävä tekijä.
6. **Kontrollin tunteen tukeminen**: käyttäjän kannalta paras ratkaisu on, jos jokaisen toiminnon pystyy tarvittaessa perumaan, ja käyttäjän tekemien virheiden korjaamisen olisi oltava mahdollisimman helppoa.
7. **Virheilmoitusten** tulisi olla mahdollisimman selkeitä: käyttäjää pitäisi ohjeistaa virheilmoitusten tulkinnassa.
8. **Virheiden ennaltaehkäisy**: matematiikkaohjelmissa aiheutuu usein virheitä siitä, että käyttäjä ei hallitse ohjelman syntaksia.

9. **Ohjekirjan tarpeellisuus**: jokaiselta sivulta pitäisi päästä selkeisiin ohjeisiin helposti.
10. **Heuristinen arviointi**: arvioinnin avulla pystytään kehittämään ongelmakohdista käytettävyyden kannalta toimivampia.

Hyvän käytettävyyden hyödyt ovat kiistattomat. Vaikka käyttäjät yleensä oppivat käyttämään huonostikin suunniteltua ohjelmaa, ohjelman opetteluun kuluva aika on pois ohjelman tarkoituksenmukaisesta käytöstä. Käyttäjän kannalta hyvä käytettävyys tarkoittaa käytön aloittamisen helppoutta ja ohjelman tarkoituksenmukaista käyttöä. Hyvä käytettävyys myös tehostaa työskentelyä, kun käyttäjä tekee vähemmän virheitä ja viihtyy tehtävänsä parissa paremmin. [57]

Lähtökohtaisesti uuden opiskeluympäristön pitäisi edistää opiskelua, eikä se saisi ainakaan heikentää oppimistuloksia. Oppimisympäristöstä ei saisi myöskään aiheutua kohtuuttomasti lisätyötä opiskelijalle eikä opettajalle [89]. Tutkimustulosten perusteella sähköisistä opetusjärjestelmistä saadut tulokset ovat ainakin samaa tasoa kuin perinteisillä menetelmillä saadut tulokset [50, 70], joten sähköisten opetusjärjestelmien kehittäminen on perusteltua [89].

3.2 MathCheck

Tampereen uuden korkeakouluyhteisön kaikilla kampuksilla käytetään STACK-tehtäviä matematiikan opetuksessa. STACK-järjestelmällä on yleisimpiä sähköisten oppimisympäristöjen vahvuuksia. Ohjelman käyttö ei ole sidottu paikkaan eikä aikaan, siinä on interaktiivisia ominaisuuksia, ja automaattisena tarkastusjärjestelmänä se vapauttaa resursseja muuhun käyttöön. Aktiivisella STACK-tehtävien tekemisellä on todettu olevan myös positiivinen vaikutus ainakin alkuvaiheen opintomenestykselle [76]. Ohjelmaan voi myös ohjelmoida välitöntä, henkilökohtaista ja anonyymiä palautetta opiskelijalle, ja tehtäviä voi yrittää uudelleen. Vastauskertoja voidaan myös rajoittaa tarvittaessa. [69] Sopivassa määrin käytettynä STACK-tehtävät muodostavat mielekkään osan opintojaksojen harjoitusmateriaalia.

Monista hyvistä ominaisuuksistaan huolimatta STACK:in ja vastaavien ohjelmien hyvät puolet ovat samalla niiden heikkouksia. STACK-järjestelmä tarkastaa vain opiskelijan lopullisen vastauksen ennalta valmisteltuihin tehtäviin tai tehtäväsarjoihin [**valmari2016mathcheck**]. Pelkän vastauksen antaminen ohjelmalliseen tehtävään houkuttelee ratkaisemaan tehtävän matemaattisilla työkaluilla matemaattisten taitojen kehittämisen sijaan [**valmari2016mathcheck**].

Opettajalle olisi tärkeää erityisesti tentin yhteydessä nähdä opiskelijan koko päättelyketju. Paperilla tehdyissä tehtävissä tämä on tietenkin mahdollista, mutta opintojen sähköistyessä ja sähköisten opetusmenetelmien kehittyessä etsitään ja kehitetään jatkuvasti tehokkaampia, ajasta ja paikasta riippumattomia opiskeluvälineitä.

MathCheck-kaavatarkastimeen perustuvia sähköisiä opiskeluohjelmia on ollut TTY:n matematiikan opintojaksoilla käytössä kokeiluluonteisesti vuodesta 2015 lähtien. Syksyllä 2017 MathCheck otettiin kokeilukäyttöön rajoitetusti uusien opiskelijoiden matematiikan perusopintojen perustaitotestin kertauksen tehtäväalustana. Uusin MathCheck-kaavatarkastimen sovellus on opintojakson MAT-123 matematiikan perustaitojen kertaus Plussa -sivustolla [82].

MathCheck–kaavatarkastimen perusfilosofia on erilainen kuin esimerkiksi STACK-järjestelmän. Opiskelija voi kirjoittaa nyt samassa ohjelmassa koko vastauksen välivaiheineen. MathCheck tarkastaa opiskelijoiden tekemistä yksilöllisistä välivaiheista koko päättelyketjun matemaattisen oikeellisuuden ilman valmista mallia. Ohjelma antaa myös välittömästi ohjaavaa palautetta opiskelijan ratkaisusta. [**valmari2016mathcheck**]

MathCheck:in perusideologia on *konstruktivistinen* ja sen idea on ohjata opiskelijaa työskentelemään oikean ratkaisun löytämiseksi [**valmari2016mathcheck**]. Konstruktivistisen oppimiskäsityksen mukaan opiskelijan aikaisemmat tiedot ja taidot yhdistyvät ja tuottavat uutta tietoa [29, 65, 73, 92].

Perinteisissä laskuharjoituksissa opiskelija tekee harjoituksen ja tarkastaa sen harjoitustilanteessa usein eri päivänä. MathCheckin avulla on mahdollista parantaa suoritusta välittömästi, kuten assistentin ohjaamissa laskuharjoituksissa. Etuna harjoitustilaisuuteen verrattuna on, että tehtävät voi tehdä milloin ja missä tahansa, kunhan käytössä on internet-yhteys. [**valmari2016mathcheck**]

Toisin kuin esimerkiksi STACK, MathCheck on alun perin tehty opiskelun tukiohjelmaksi antamaan opiskelijalle pedagogisesti hyödyllistä palautetta. Matemaattisen ymmärryksen on katsottu olevan tärkeämpi tavoite kuin aina täsmälleen oikea vastaus [**valmari2016mathcheck**]. Perinteisin menetelmin opiskelija voi kysyä apua esimerkiksi opintojaksojen yhteyteen perustetuissa keskusteluryhmissä tai internetin matematiikkapalstoilta, tai vasta seuraavassa kontaktissa harjoitusten ohjaajan tai luennoitsijan kanssa. Parhaassakin tapauksessa vastaukseen kuluu aina jonkin verran aikaa, ja tehtävä saattaa unohtua. MathCheckin avulla opiskelija saa palautetta ratkaisustaan heti.

Ohjelman merkittävä ominaisuus on mahdollisuus kirjoittaa tehtävään välivaiheita, jotka voidaan löytää ja arvostella. Opiskelija voi myös halutessaan kirjoittaa ohjelmaan muistiinpanoja, jotka näkyvät vastauksessa, mutta joita ohjelma ei tarkasta. Näin opettaja näkee, mitä opiskelija on tehtävää tehdessään ajatellut ja voi arvioida halutessaan opiskelijan osaamista laajemmin kuin vain lyhyen vastauksen perusteella.

Taulukossa 3.1 on erään opiskelijan vastaukset esimerkkitehtävään. Opiskelijan kirjoittama ratkaisu on kirjoitettu mustalla. MathCheck ottaa palautteeseen opiskelijan vastauksen sellaisenaan virheelliseen kohtaan asti, ja tulostaa virheellisen kohdan ja virheilmoituksen punaisella. Ellei virheitä löydy, ohjelma palauttaa opiskelijan vastauksen ja palautteen ("No errors found. Pleace notice that check was not complete") tulostettuna mustalla. Ohjelman tulostama ladonta on MathJax ja AsciiMath -standardien mukainen. Esimerkkitehtävässä nähdään, miten MathCheck on antanut opiskelijalle vihjeitä lasku- ja syntaksivirheistä. Ohjelman tarkistus pysähtyy aina ensimmäiseen virheeseen, ja opiskelija on korjannut tehtävää ohjelman antamien vihjeiden mukaan.

Ensimmäisessä ratkaisuyrityksessä on monikertaa tarkoittava kirjain n. MathCheck on tulkinnut kirjaimen yritykseksi esitellä uusi muuttuja, ja merkinnyt virheellisen kohdan huutomerkillä. Toisessa yrityksessä opiskelija on korjannut vastaustaan, mutta nyt vastaus on väärin ja ohjelma ilmoittaa, että vastaus on väärin. Kolmaskin vastaus on väärin, mutta neljäs yritys onnistuu ja ohjelma ilmoittaa, että virheitä ei löydetty.

Taulukko 3.1. *MathCheck esimerkkivastaus.*

Yritys	Opiskelijan vastaus
1.	<=> cos(8x)=-1 <=> 8x=0+n*2*pi <=> x=(n*pi)/4 <=> x=pi/4. New variables must not be introduced here. <=> cos(8x)=-1 <=> 8x=0+n !
2.	<=>cos(8x)=-1 <=> 8x=0+2*pi <=> x=(pi)/4 <=> x=pi/4 $\Leftrightarrow cos(8x) = -1 \Leftrightarrow 8x = 0 + 2 * \pi$ A teacher given root was lost here.
3.	<=> cos(8x)=-1 <=> 8x=2*pi <=> x=pi/4 <=> x=pi/4 <=> cos(8x)=-1 $\Leftrightarrow 8x = 2 * \pi$ A teacher given root was lost here.
4	<=>cos(8x)=-1 <=> 8x=pi <=> x=pi/8 $\Leftrightarrow cos(8x) = -1 \Leftrightarrow 8x = \pi \Leftrightarrow x = \frac{\pi}{8}$ No errors found. Please notice that the check was not complete.

Vaikka MathCheck-kaavatarkastin suunniteltiin alun perin matematiikan harjoitusohjelmaksi, sitä voidaan käyttää myös testien tekemiseen. Ohjelmassa itsessään ei ole arvosteluominaisuuksia eikä käyttäjän tunnistusta [**valmari2016mathcheck**]. MathCheck –ohjelmaa voidaan käyttää esimerkiksi opiskeluun ja materiaalien jakamiseen tarkoitetussa ympäristössä, jossa käyttäjäkohtainen data voidaan kerätä talteen ja käyttäjätietojen hallinta on valmiiksi toteutettu.

4 TYÖSSÄ TARKASTELTAVIEN TEHTÄVIEN KERTOIMIEN SATUNNAISTAMINEN

Tämän työn tarkoitus on virhetarkastelun lisäksi kehittää kertaustehtäviä. Työn laajuus huomioon ottaen ei ole järkevää muuttaa tehtäviä muutoin, kuin kehittämällä valittujen tehtävätyyppien aitoa satunnaistamista.

Tehtävien aito satunnaistaminen tarkoittaa, että ohjelma tuottaa satunnaisarvoilla aidosti satunnaisia mielekkäitä tehtäviä, joita voidaan generoida samalla ohjelmakoodilla suuria määriä. Tehtävien tulisi olla myös tasalaatuisia ja niiden vaativuustason tulisi samassa tehtävässä olla samalla tasolla. Ohjelma ei saisi myöskään tuottaa tehtäviä, joita ohjelmoija ei tarkoittanut tai jotka helpottavat tehtävän ratkaisua merkittävästi. Kehitettäviksi tehtäviksi valittiin ensimmäisen ja toisen asteen yhtälötehtävät ja trigonometrian yhtälötehtävä.

Satunnaistaminen aloitetaan esimerkiksi pohtimalla, millaisia ominaisuuksia tehtävän vastaukselta halutaan.

- Mitkä tekijät halutaan satunnaistaa?
- Onko vastauksiin odotettavissa tilanteita, jotka halutaan rajata vastauksista pois (esimerkiksi nollalla jakaminen ja liian yksinkertainen vastaus)?
- Mitä muotoa vastauksen halutaan olevan (esimerkiksi kokonaisluku vai murtoluku)?
- Halutaanko kertoimien itseisarvolle asettaa ylä- ja alarajat.
- Miten tehtävät pystytään ohjelmoimaan käytettävälle kertausohjelmalle?

Seuraavaksi päätetään satunnaistamisen periaatteista. Matematiikan harjoitustehtävien satunnaistaminen ei sinänsä ole riippuvainen ohjelma-alustasta, vaan samat periaatteet toimivat useimmilla ohjelma-alustoilla. Kun satunnaistamisen periaatteet ovat selvillä, tarkastellaan ohjelmointialustakohtaisia toteutusmahdollisuuksia. Esimerkiksi käytettävien funktioiden määrässä voi olla rajoituksia ohjelmakohtaisesti. Ohjelmien toteuttaminen käytännössä riippuu käytettävistä ohjelmointityökaluista, kaikille ohjelmille ei voida toteuttaa kaikkia ominaisuuksia. Esimerkiksi lukujen suuruuden vertailu ei onnistu samalla tavalla kaikissa ympäristöissä. Työn esimerkkiohjelmat on toteutettu MathCheck-kaavatarkastimelle. Ohjelmien kirjoittamiseen tarvittavat työkalut löytyvät lähteestä [2] ja tehtävien kirjoittamista voi kokeilla tarkoitukseen tehdyllä verkkosivulla [3].

4.1 Ensimmäisen asteen reaalikertoiminen yhtälö

Tässä kappaleessa on käytetty lähteinä [32, 64]. Ensimmäisen asteen yhtälötehtävän satunnaistaminen kannattaa tehdä satunnaistamalla yhtälön kertoimia. Satunnaistamisen esittelemiseksi ensimmäisen asteen yhtälötehtävässä valittiin ensimmäisen asteen yhtälö, jonka molemmat puolet ovat ensimmäisen asteen polynomeja

$$
\begin{aligned}
&a_1x + a_3 = a_2x + a_4,\\
&\Rightarrow a_1x - a_2x + a_3 - a_4 = 0,
\end{aligned}
$$

jossa $a_i \in \mathbb{R}$ ja $a_i \neq 0$. Tehtävässä etsitään kertoimia a_1, a_2, a_3 ja a_4.

Yhtälö voidaan kirjoittaa matriisimuotoon $A\mathbf{a}$ jossa $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^T$,

$$
\begin{bmatrix} x & -x & 1 & -1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \mathbf{0}.
$$

Muodostetaan ensin matriisin $[A|\mathbf{0}]$ *redusoitu riviporrasmuoto* [64]

$$
[A|\mathbf{0}] = \begin{bmatrix} x & -x & 1 & -1 & 0 \end{bmatrix} \longrightarrow \text{rref}([A|\mathbf{0}]) = \begin{bmatrix} 1 & -1 & \frac{1}{x} & -\frac{1}{x} & 0 \end{bmatrix}, x \neq 0.
$$

Redusoidun vaakariviporrasmuodon ensimmäinen alkio on *johtava ykkönen* [64]. Valitaan johtavaksi muuttujaksi a_1. Koska muita rivejä ei ole, a_1 voidaan esittää a_2:n, a_3:n ja a_4:n avulla. a_2, a_3 ja a_4 voidaan valita vapaasti, ja niitä kutsutaan *vapaiksi muuttujiksi* [64]. . Vapaat muuttujat ja ja niiden avulla määritetty a_1 toteuttavat yhtälön

$$
a_1 - a_2 + \frac{1}{x}a_3 - \frac{1}{x}a_4 = 0,
$$

josta

$$
a_1 = a_2 - \frac{1}{x}a_3 + \frac{1}{x}a_4.
$$

Jos valitaan parametrit

$$
\begin{aligned}
a_2 &= s,\\
a_3 &= t,\\
a_4 &= u,
\end{aligned}
$$

ratkaisu voidaan kirjoittaa vektorimuodossa

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} s - \frac{1}{x}t + \frac{1}{x}u \\ s \\ t \\ u \end{bmatrix} = s \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} -\frac{1}{x} \\ 0 \\ 1 \\ 0 \end{bmatrix} + u \begin{bmatrix} \frac{1}{x} \\ 0 \\ 0 \\ 1 \end{bmatrix} = s' \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t' \begin{bmatrix} -1 \\ 0 \\ x \\ 0 \end{bmatrix} + u' \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Vapaat muuttujat voidaan määritellä parametrien s', t' ja u' avulla. Kertoimet a_1, a_2 ja a_3 voidaan nyt ratkaista parametrien s', t', u' avulla

$$\begin{aligned} a_1 &= s' - t', \\ a_2 &= s', \\ a_3 &= t'x + u', \\ a_4 &= u'. \end{aligned}$$

Tehtävän ratkaisu on

$$x = \frac{a_4 - a_3}{a_1 - a_2},\ a_1 \neq a_2.$$

Koska kyseessä on harjoitustehtävä, halutaan että ratkaisu ei ole nolla. Tämä toteutuu, kun osoittaja ei ole nolla

$$a_4 \neq a_3$$

Tehtävän kertoimille voi asettaa erilaisia ehtoja, jotka muuttavat tehtävän vaikeusastetta, esimerkiksi voidaan etsiä murtolukuvastauksia. Tehtävän perusversiossa on järkevää, että kertoimet ovat kokonaislukuja. Kertoimia voi asettaa ohjelmoimalla arvot sopivasti esimerkiksi niin, että ohjelman palauttamat satunnaisluvut asetetaan kokonaisluvuiksi ja suljetaan nolla pois mahdollisten arvojen joukosta.

Tarkastellaan seuraavaksi ehtojen täyttyminen. Koska kertoimet $a_3 = t'x + u'$, ja $a_4 = u'$, niin $a_3 - a_4 = t'x + u' - u' = t'x$, jolloin ehto $a_3 \neq a_4$ pätee.

Toinen ehto toteutuu, kun kertoimet $a_1 = s' - t'$, ja $a_2 = s'$, jolloin $a_1 - a_2 = s' - t' - s' = -t'$ ja $a_1 \neq a_2$ pätee.

Ehtoja voi asettaa ohjelmoimalla. Kolmas ehto voidaan toteuttaa ohjelmoimalla niin, että satunnaisluku ei ole nolla. Tehtävässä edellytetään, että mikään kertoimista a_1, a_2, a_3 ja a_4 ei ole nolla. Koska $a_1 = s' - t'$, pitää asettaa ehto $|s'| \neq |t'|$. Lisäksi esimerkiksi tehtävän kertoimille itseisarvolle voidaan asettaa ala- ja yläraja. Kun merkitään suurinta haluttua kokonaislukukerrointa K:lla, niin annetaan lisäehdot

$$
\begin{aligned}
|a_i| \leq K \quad & \Leftrightarrow -K \leq a_i \leq K, \\
|s', t', u'| \leq K \quad & \Leftrightarrow -K \leq k_i \leq K, \\
|s' - t'| \leq K \quad & \Leftrightarrow -K \leq s' - t' \leq K, \\
|t'x + u| \leq K \quad & \Leftrightarrow -K \leq t'x + u \leq K.
\end{aligned}
$$

Edellä esiteltiin ensimmäisen asteen yhtälön kertoimien satunnaistamista. Tehtävän lopullinen toteutus riippuu käytettävästä ohjelmasta. MathCheck-kaavatarkastimelle kirjoitettu esimerkkiohjelma 4.1 esitellään sivulla 32.

Esimerkkiohjelman rivillä 8 päätetään suurin satunnaisluku *random(k)*, jonka perusteella muuttujat satunnaistetaan. Muuttuja t arvotaan muuttujan s perusteella niin, että niiden itseisarvot eivät ole samat.

Rivillä 12 satunnaistetaan muuttujaa x ohjelmassa kuvaava y ja edellisten etumerkit satunnaistetaan *select*-funktiolla riveillä 15–17. Rivillä 13 määritellään apumuuttuja w, jonka avulla määritellään alkuperäisen yhtälön muuttuja a_3. Rivillä 22 satunnaistetaan muuttuja u niin, että u:n ja w:n itseisarvot eivät ole samat. Tehtävän muuttujat $a_1 = a = s, a_2 = b = s + t, a_3 = c = v + u$ ja $a_4 = d = u$ määritellään rivillä 24. Tehtävän tulostuksen muotoilu ja tulostus esitetään riveillä 26–30. Komennolla *#hidden* piilotetaan tehtävän vastaus ja *equation* määrittelee vastauksen rivillä 33. Tehtävän sievennysaste (kompleksisuus) asetetaan rivillä 35 komennolla *f_ nodes* (13). Rivillä 37 tehtävän ratkaisuyritystä verrataan alkuperäiseen tehtävänantoon ja lopuksi ohjelma tulostaa vastausruudun *#initial* ja mahdolliset ohjelmoijan antamat lisämerkinnät (esimerkiksi $\Leftrightarrow$ tai $x =$).

4.2 Toisen asteen yhtälöt

Lauseessa 2.6 sivulla 18 osoitettiin yhteys toisen asteen yhtälön $a_2x^2 + a_1x + a_0 = 0$ juurten summalle ja tulolle

$$
\begin{aligned}
x_1 + x_2 &= -\frac{a_1}{a_2}, \\
x_1 \cdot x_2 &= \frac{a_0}{a_2}, \ a \neq 0.
\end{aligned}
$$

Toisen asteen yhtälötehtävän kertoimet kannattaa satunnaistaa valitsemalla ensin juuret x_1, x_2 ja toisen asteen termin kerroin a_1. Etsittävät kertoimet a_0 ja a_1 ovat

$$
a_1 = -a_2(x_1 + x_2) \tag{4.1}
$$

$$
a_0 = a_2(x_1 x_2). \tag{4.2}
$$

Välttämätön ehto kertoimille on että $a_1 \neq 0$. Lisäehtoina halutaan, että kumpikaan juurista ei ole nolla, jolloin $a_1 \neq 0$ ja $a_0 \neq 0$. Molemmat ehdot toteutuvat valitsemalla ohjelmoitaessa ohjelman palauttamat kertoimet niin, ne ovat erisuuria kuin nolla. Lisäehtona voidaan

```
#title Helppo ensimmaisen asteen yhtalo ax + c = bx + d,
jossa muuttuja ja kertoimet ovat kokonaislukuja.
#let
    k := 8
    // Päätetään K
    s := random(k)+1
    // Arvotaan a=s, (pidetään s vielä positiivisena.)
    t := s +(random(s)
    // Arvotaan t s:n perusteella niin, että niiden s:n ja t:n
    // itseisarvot ovat erisuuret.
    y := random(k)+1
    // Arvotaan muuttuja y positiivisena.
    w := t*y
    // Lasketaan apumuuttuja w.
    s := select(random(2),−1,1)*s
    t := select(random(2),−1,1)*t
    y := select(random(2),−1,1)*y
    // Arvottiin parametrit s , t ja muuttujan y etumerkit.
    v := t*y
    // t*x = v = t*y oikeilla arvotuilla etumerkeillä.
    u := w+random(w)+1
    // Arvotaan u w:n perusteella koska halutaan että
    //|u| on erisuuri kuin |w|.
    a := s  b := s+t  c := v+u  d := u
    V := "x"
    A := pterm(a, V) B := pterm(b, V) C := Pz(c) D := Pz(d)
    // Tehtävänannon muotoilu.
#question
Ratkaise ‘#A #C = #B #D‘.
// Tulostetaan tehtävänanto.
#hidden
// Piilotetaan vastaus.
equation #V = #y ends
// Määritellään vastaus.
f_nodes 13
// Määritellään kompleksisuusaste.
#A #C = #B #D
#initial <=>#
// Tulostetaan tehtävään vastausruutu.
```

Ohjelma 4.1. *Ensimmäisen asteen yhtälön esimerkkikoodi.*

```
1  #title Helppo toisen asteen yhtalo.
2  #let
3      k := 6
4      //Valitaan suurin satunnaisluku k.
5      x := random(k) +1
6      y := x +(random(x)+1)
7      // Valittiin juuret siten, että juurten itseisarvot
8      //eivät ole yhtä suuret.
9      x := select(random(2), −1, 1)∗x
10     y := select(random(2), −1, 1)∗y
11     // Valitaan juurten etumerkki select-funktiolla
12     z := random(4) + 2
13     // Valitaan z, jonka avulla muodostetaan kertoimet a, b, c.
14     a := z  b := a∗(−x−y)  c := a∗x∗y
15     g := gcd(a,b)  g := gcd(g,c)
16     a := a/g  b := b/g  c := c/g
17     // Kertoimet a, b, c ja sievennys.
18     V := select( random(1),"x" )
19     A := pterm(a, V+"^2")
20     B := Pterm(b, V)
21     C := Pz(c)
22     // Tulostetaan kysymys.
23 #question
24 Ratkaise ‘#A #B #C = 0‘.( vastaus #x or #y)
25 #hidden
26 equation #V = #x or #V = #y ends
27 // Vastaus
28 f_nodes 13
29 #A #B #C = 0
30 #initial <=>#
```

Ohjelma 4.2. *Toisen asteen yhtälön esimerkkikoodi.*

asettaa esimerkiksi toinen tai molemmat juuret murtoluvuiksi. Lisäksi on järkevää asettaa kertoimille ala- ja yläraja

$$
\begin{aligned}
|x_i| \leq K \quad &\Leftrightarrow -K \leq x_i \leq K, \\
|a_i| \leq K \quad &\Leftrightarrow -K \leq a_i \leq K.
\end{aligned}
$$

Ohjelman lopullinen toteutus riippuu aina käytetystä ohjelmasta. Sivulla 33 esitetään eräs MathCheck-kaavatarkastimelle kirjoitettu esimerkkiohjelma 4.2 "Helppo toisen asteen yhtälö". Muuttujien satunnaistaminen tapahtuu samalla tavalla ja periaatteilla kuin ensimmäisen asteen yhtälön yhteydessä. Rivillä 14 muuttujat valitaan kaavojen 4.1 ja 4.2 mukaisesti. Rivillä 15 muuttujat sievennetään funktiolla *gcd* (suurin yhteinen tekijä). Muut vaiheet toteutettiin vastaavasti kuin esimerkkiohjelmassa 1.

4.3 Trigonometriset yhtälöt

Trigonometrisiä yhtälöitä käsitellään tehtäväsarjassa 10. Satunnaistettaviksi valittiin kertaustehtävien kaltainen tehtävä. Tehtävän ratkaisu perustuu trigonometristen funktioiden palautuskaavoihin 2.5 ja 2.7 sivulla 20, jotka on johdettu reaalifunktioiden jaksollisuuden ja symmetriaominaisuuksien perusteella.

Kertaustehtävät pitäisi pystyä laskemaan lukiotason perustiedoilla, ilman laskinta päättelemällä ja pelkkien annettujen vihjeiden ja ohjeiden perusteella. Laadukkaita, halutun tasoisia ja vaikeusasteeltaan samanlaisia mutta erilaisia tehtäviä pitäisi pystyä tuottamaan kuitenkin suurehko määrä.

MathCheck ei tunnista trigonometristen funktioiden monikertoja, joten tehtävässä pitää rajautua ensimmäiseen kertaluokkaan tai monikerralle on annettava numeroarvo. Satunnaistaminen aloitetaan esimerkiksi pohtimalla, millaisia ominaisuuksia tehtävän vastaukselta halutaan.

Tehtävään valittiin kriteerit, joissa

- kysymys on muotoa $\sin(a_1x) = \cos(a_2x + \frac{\pi}{a_3})$ tai $\cos(a_1) = \sin(a_2x + \frac{\pi}{a_3})$,
- kertoimet a_i on satunnaistettu $a_1, a_2 \in \mathbb{Z}\backslash\{0\}$, $\frac{\pi}{a_3} \in [0, 2\pi]$, $a_i \neq 0$,
- vastaus ei ole nolla,
- vastaus on muotoa π jaettuna kokonaisluvulla,
- tehtävässä etsitään ensimmäisen kertaluokan ratkaisua tai kertaluokalle annetaan numeroarvo,
- vastauksessa ei päädytä jakamaan nollalla.

Satunnaistettava tehtävä on muotoa

$$F(a_1x) = G(a_2x + \frac{\pi}{a_3}),\ \text{jossa } a_1, a_2 \in \mathbb{Z} \text{ ja } F, G \in \{\sin, \cos\},\ \frac{\pi}{a_3} \in [0, 2\pi].$$

Tehtävässä etsitään ensimmäisen kertaluokan ratkaisua.

Tehtävän ratkaisussa käytetään sinin ja kosinin palautuskaavoja 2.5 ja 2.7:

$$\begin{aligned}
&\sin(a_1x) = \cos\left(a_2x + \frac{\pi}{a_3}\right) = \sin\left(\frac{\pi}{2} - \left(a_2x + \frac{\pi}{a_3}\right)\right), \\
&\Rightarrow a_1x = \frac{\pi}{2} - a_2x - \frac{\pi}{a_3} \quad \vee \quad a_1x = \frac{\pi}{2} + a_2x + \frac{\pi}{a_3} \\
&\Rightarrow a_1x + a_2x + \frac{\pi}{a_3} = \frac{\pi}{2} \quad \vee \quad a_1x - a_2x - \frac{\pi}{a_3} = \frac{\pi}{2}.
\end{aligned}$$

Trigonometrian polynomiyhtälötehtäviä voi satunnaistaa samaan tapaan kuin edellä ensimmäisen asteen yhtälötehtävien. Yhtälö voidaan kirjoittaa matriisimuotoon ja ratkaista ratkaisuvektori vapaasti valittavien kerrointen s ja t ja virittäjävektorien avulla. Nyt

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = s \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} -1 \\ 0 \\ x \end{bmatrix} + \begin{bmatrix} \frac{\pi}{2x} \\ 0 \\ 0 \end{bmatrix}$$

Vapaat muuttujat voidaan määritellä parametrien s ja t avulla. Merkitään kertoimet a_1, a_2, a_3 parametrien s ja t avulla

$$\begin{aligned} a_1 &= -s - t + \frac{\pi}{2x}, \\ a_2 &= t, \\ a_3 &= tx. \end{aligned}$$

jossa kertoimet a_1, a_2 ja s, t kuuluvat kokonaislukujen joukkoon.

Tehtävän ratkaisu on

$$x = \frac{\frac{\pi}{2} - \frac{\pi}{a_3}}{a_1 + a_2}, \ a_1 \neq -a_2.$$

Halutaan lisäksi, että ratkaisu ei ole nolla. Tämä toteutuu, kun

$$a_3 \neq 2.$$

Tehtävälle voi asettaa käytettävästä ohjelmasta riippuen lisäehtoja, esimerkiksi huomioimalla kerrannaiset $n2\pi$. Tehtävistä voi myös muokata yksinkertaisempia versioita esimerkiksi valitsemalla tehtävänannossa yhtälön molemmille puolille joko $\sin$ tai $\cos$ tai valitsemalla $a_3 = 0$.

Tehtävässä on järkevää, että kertoimet ovat kokonaislukuja. Kertoimia voi asettaa ohjelmoimalla arvot sopivasti esimerkiksi niin, että ohjelman palauttamat satunnaisluvut asetetaan kokonaisluvuiksi ja suljetaan nolla pois mahdollisten lukujen joukosta.

Tehtävän kertoimille voidaan asettaa ala- ja yläraja samalla tavalla kuin edellisissä tehtävissä

$$\begin{aligned} |a_i| &\leq K & &\Leftrightarrow -K \leq a_i \leq K \\ |s, t| &\leq K & &\Leftrightarrow -K \leq s, t \leq K, \\ |sx| &\leq K & &\Leftrightarrow -K \leq sx \leq K. \end{aligned}$$

Tehtäviä voidaan toteuttaa muillakin tavoilla. Esimerkkinä ohjelma 4.3 sivulla 36. Ohjelmassa on riveillä 3–6 valittu suurin satunnaisluku ja sen perusteella satunnaistettu kertoimet s, t ja u. Kerroin t on valittu niin, että kerrointen s ja t itseisarvot ovat erisuuret.

```
1  #title Trigonometrinen yhtalo.
2  #let
3      k := 4
4      s := random(k)+1
5      t := s+random(s)+1
6      u := random(k)+3
7      // Valittiin K ja kertoimet s, t ja u.
8      s := select(random(2),−1,1)∗s
9      t := select(random(2),−1,1)∗t
10     // valittiin kertoimien s ja t etumerkit.
11     v := random(2)
12     y := select(v,−1,1)∗u
13     // Valitaan y apumuuttujan v avulla. |y|=u, tarvitaan
14     // vastaukseen, u tarvitaan kysymyksen muotoiluun koska
15     // pi/u etumerkki valitaan erikseen (= C)
16     V := "x"
17     A := pterm(s,V)
18     B := pterm(t,V)
19     C := select(v,"−", "+")
20     w := random(2)
21     F := select(w,"sin","cos")
22     G := select(w,"cos","sin")
23     // Tulostusasetukset ja arvotaan funktioiden F ja G arvot.
24     x := y−2
25     z := 2∗y∗(s+t)
26     // Asetetaan vastausasetukset muuttujien F ja G perusteella.
27 #question Ratkaise #F(#A)=#G(#B #C pi/#u)`
28 "Anna␣vastauksessasi␣pienin␣positiivinen␣ratkaisu␣radiaaneissa."
29 // Tehtävänanto.
30 #hidden
31 equation x=pi∗#x/#z ends
32 // Vastaus.
33 f_nodes 8
34 // Tehtävän monimutkaisuusaste
35 #F(#A)=#G(#B #C pi/#u)
36 #initial <=>
```

Ohjelma 4.3. *Trigonometrisen yhtälön esimerkkikoodi.*

Kertoimelle u on lisäksi ehto $u \neq 2$, koska haluttiin että $\frac{\pi}{a_3}$ ei ole yhtä suuri kuin $\frac{\pi}{2}$. Rivillä 11 valitaan apumuuttuja v tehtävänantoa varten. Apumuuttujan avulla satunnaistetaan tehtävänannon viimeisen termin $\frac{\pi}{u}$ etumerkki. Rivillä 12 esitellään muuttuja y ratkaisun muotoilua varten. Riveillä 16–22 määritellään kysymyksen tulostusasetukset [2]. Riveillä 24 ja 25 määritellään ratkaisussa tarvittavat muuttujat x ja z. Tehtävänanto on rivillä 28.

Esitetty esimerkkiohjelma on vain yksi ratkaisu. Tehtävää voi kehittää kulloinkin tarvittavalla tavalla, esimerkiksi laatimalla tehtävän yhteyteen voi vaikeutuvan tehtäväsarjan. MathCheck-kaavatarkastinta kehitetään edelleen, ja uudet ominaisuudet lisäävät tulevaisuudessa tehtävien laadintamahdollisuuksia.

5 TUTKIMUKSEN TOTEUTUS

Tämän tutkimus pohjautuu TTY:n Matematiikan laboratorion projektiin, jossa tuotettiin kesällä 2017 MathCheck-kaavatarkastimeen [**valmari2016mathcheck**, 4] perustuva matematiikan perustaitojen kertausohjelma aiemman Math-Bridge [12] -alustalle tehdyn Matematiikkajumppa [83] -kertausohjelman rinnalle ja korvaajaksi. Tulos tästä projektista oli uusi Matematiikkajumppa [84] -sivusto.

Kertausohjelman toteutuksen lisäksi ohjelmalla tehdyistä vastauksista haluttiin etsiä samankaltaisia ratkaisuja, joita voitaisiin hyödyntää automaattisen arvioinnin kehittämisessä. Uudenlaisesta kertausohjelmasta haluttiin myös kerätä talteen käyttäjien kokemuksia kertausohjelman jatkokehitystä varten. Tätä tarkoitusta varten käyttäjille tehtiin tutkimusta tukeva vapaaehtoinen kyselytutkimus.

5.1 Tutkimuksen kohdejoukko

Uudella ohjelmalla toteutettu matematiikan perustaitojen kertaus tehtiin kahden matematiikan ensimmäisen opintojakson, Matematiikka 1 (MAT-1) ja Insinöörimatematiikka 123 (IMA-123) perustaitotestin yhteydessä. Harjoitustehtäviä oli 80 kappaletta lukiomatematiikan keskeisiltä alueilta [85]. Perustaitotestissä heikosti suoriutuneet opiskelijat kertasivat ohjelmalla perustaitotestin uusintaa varten. Opiskelijoille luvattiin myös opintojakson läpäistyn tentin pistemäärään lisättäviä, kurssin arvosanaan vaikuttavia, lisäpisteitä kertaustehtävien tekemisestä.

Ryhmät olivat opiskelutaustaltaan hyvin erilaisia. Suurin osa insinööriopiskelijoista suorittaa ensimmäisen vuoden aikana Insinöörimatematiikka (IMA) 1-3 ja Algoritmimatematiikan kurssin tai kurssit IMA 1-4. MAT-1 opiskelijat tulevat opiskelemaan nimenomaan luonnontieteitä, tai tarvitsevat normaalia laajemmat matematiikan perusopinnot. MAT-123 taas on matematiikan pohjaopintoja AMK-tasolta yliopistomatematiikan tasolle täydentävä opintojakso.

Ryhmiin MAT-1 ja IMA-123 hakeutuvat opiskelijat edustavat matematiikan pohjataitojen osalta lähtötasoltaan yleensä kaikkien ensimmäisen vuoden insinööriopiskelijoiden ääripäitä. Tutkimukseen osallistuneiden opiskelijoiden taustatietoja selvitettiin kyselytutkimuksella. Opintojakson MAT-1 opiskelijoiden matematiikan perustaidot ovat paremmat kuin insinööriopiskelijoilla keskimäärin, ja ryhmäläisten edellisistä matematiikan opinnoista on kulunut keskimäärin 1,2 vuotta (mediaani 1). Opintojakson IMA-123 opiskelijoiden matematiikan perustaidot ovat yleensä heikommat, ja opiskelijoilla on edellisistä opinnoista kulunut keskimäärin 3,3 vuotta (vaihteluväli 1-18 vuotta, mediaani 3 vuotta).

5.2 Tutkimusaineisto

Tutkimusaineistona käytettiin perustaitotestin kertausohjelmasta saatua vastausdataa. Uudelta Matematiikkajumppa -sivustolta ei voitu kerätä dataa opiskelijoiden vastauksista, joten kertausohjelmasta tehtiin toinen versio Jyväskylän yliopiston ylläpitämälle sähköisen opiskelun The Interactive Material (TIM) -alustalle [30]. Vastausdatan keräämisen lisäksi tietojen säilytys ja varastointi onnistui TIM-alustalla tietoturvavaatimusten mukaisesti [31]. Matematiikan perustaitojen kertauksen jälkeen sähköiset vastaukset kerättiin, lajiteltiin ja järjestettiin tutkimustarkoituksiin sopivalla tavalla.

5.2.1 Vastausdata

TIM-alustalle [30] asennetusta Matematiikkajumppa -ohjelmasta saatiin vastausdatana noin 10000 opiskelijoiden kirjoittamaa ratkaisua kertaustehtäviin. Ensimmäisessä vaiheessa vastausdatasta kerättiin MATLAB-ohjelmalla [42] kaikki samaa tarkoittavat ratkaisut yhteen ja niiden lukumäärät laskettiin. Tätä ensimmäistä ohjelmallista käsittelyä kutsutaan tässä työssä *lajitteluksi*.

Lajittelussa datassa havaittiin häiriötekijöitä. Tulosten joukossa oli esimerkiksi tutkimuksen kohdejoukkoon kuulumattomien vastauksia, jotka vääristivät lopputulosta jonkin verran. Vastauksista kuitenkin etsittiin vain keskenään ekvivalentteja vastauksia, eivätkä ylimääräiset vastaajat vaikuttaneet juurikaan lajittelun lopputulokseen.

Kertauksen aikana ja sen jälkeen saadun palautteen seurauksena haluttiin tarkastella lisäksi syntaksivirheiden määrää ja laatua. Myös ohjelman toimintaan liittyvien tekijöiden, kuten ohjelman vaatiman vastauksen muodon, aiheuttamia virheitä haluttiin tarkastella. Siksi valituista tehtävistä otettiin listaus, joka puhdistettiin tutkimuksen kohdejoukkoon kuulumattomista ja vastaajien tunnistetiedoista. Tätä puhdistettua listausta tarkasteltiin kirjaamalla virheitä käsin. Tätä laskentaa kutsutaan *käsinkirjaukseksi*.

Ero ratkaisuyritysten määrässä johtuu aineiston epätarkkuuden lisäksi siitä, että kaikkia ratkaisuja ei käsinkirjauksessa huomioitu. Osa ratkaisuista oli epäselviä tai opiskelija oli tehnyt enemmän kuin 10 yritystä samaan tehtävään. Myös osa oikein tehdyistä ratkaisuista karsiutui laskennasta koska ohjelman sievennysvaatimus ei hyväksynyt kaikkia itseisarvotehtävien ratkaisuja, vaikka ne oli tehty oikein. Käsinkirjauksessa nämä hylätyt ratkaisut voitiin hyväksyä, ja virhelaskenta kyseisen vastaajan kohdalla lopettaa siihen. Ratkaisuyritysten todellinen määrä jäi silloin pienemmäksi kuin alkuperäinen yritysten määrä.

Samaa tarkoittavien virheiden etsimisen lisäksi TIM-alustasta saatu data järjestettiin toisen kerran ohjelmallisesti niin, että tuloksena saatiin tehtävissä onnistuneiden ja epäonnistuneiden määrät tehtäväkohtaisesti ja opiskelijaryhmittäin. Tätä ohjelmallista käsittelyä kutsutaan *vertailuksi*. Vertailuss etsittiin MATLAB [42] -ohjelman t-testiin perustuen tehtäviä, joissa eri ryhmien osaamiselle löydettiin eroa, jolla on tilastollista merkitystä.

5.2.2 Kyselytutkimus

Kertauksen jälkeen tutkimukseen valittujen ryhmien opiskelijoille tehtiin kyselytutkimus, jonka tuloksilla kartoitettiin tutkimukseen osallistuneiden ryhmien taustaa ja käyttäjäkokemuksia sähköisesti toteutetusta matematiikan opiskelusta ja kertausohjelman toiminnasta. Vapaaehtoiseen tutkimukseen vastasi 156 opiskelijaa, joista 53 kuului ryhmään IMA-123 ja 103 ryhmään MAT-1. Kyselytutkimuksessa oli vastaajan taustatietoja kartoittavien kysymysten lisäksi kaksi neliportaista Likert-vastausasteikolla (0-1-2-3) arvioitua opiskelijoiden asenteita mittaavaa kysymystä ja kahdeksan avointa kysymystä (Liite B). Kysely luotiin Webropol kysely-ja raportointityökalulla [95]. Likert-väittämien vastauksia analysoitiin tilastollisesti ohjelman raportointityökalulla. Avoimien kysymysten vastauksia teemoiteltiin aineistolähtöisellä sisällönanalyysillä.

Kyselytutkimus tehtiin tutkimuksen tuottamien käyttäjäkokemusten selvittämiseksi. Kaikkea kyselytutkimuksen tuottamaa tietoa ei analysoitu tutkimuksessa. Kyselytutkimuksen tarkasteltiin valittuja kohtia, joissa opiskelijat ilmaisivat mielipiteensä ohjelman käytöstä. Tarkasteluun otettiin mukaan sekä likert-asteikollisia kysymyksiä että sanallisia vastauksia.

5.3 TIM-datan analysointi

Lajittelussa lajiteltiin koko kertausohjelmasta saadusta vastausdatasta tehtäväkohtaisesti opiskelijoiden tekemät keskenään ekvivalentit ratkaisut ja laskettiin niiden lukumäärät. Kaikkia ratkaisuja ja niiden vastauksia ei lajittelussa huomioitu, koska tavoitteena oli etsiä toistuvat virhetyypit. Lajittelussa ei myöskään voitu huomioida erikseen syntaksi- ja sievennysvirheitä, määrittelemättömiä vastauksia eikä keskeyttäneitä. Virheiden käsinkirjauksessa voitiin myös syntaksi- ja sievennysvirheet, määrittelemättömät vastaukset ja keskeyttäneet huomioida omiksi luokikseen.

Sekä lajittelussa että käsinkirjauksessa löydettyjen virheiden luokittelussa käytettiin virheiden luokittelumallia [17], jota on sovellettu tehtäväkohtaisesti. Käytettyyn luokittelumalliin lisättiin luokka *määrittelemätön*, koska opiskelijat tekivät paljon epämääräisiä ratkaisuja, joista ei voinut päätellä, mitä ne tarkoittivat. Luokan vastaukset olisi voitu jättää myös kokonaan pois tästä tarkastelusta.

Virheiden käsinkirjauksessa oli mukana myös luokka *keskeytynyt*. Tehtävä on tulkittu keskeytyneeksi, jos tehtävän tekeminen on lopetettu ennen kuin tehtävä saatiin oikein tai virheellisiä vastauksia samaan tehtävään on tehty yli kymmenen.

Tutkimuksessa virhetarkastelun ensisijaisena tavoitteena oli löytää suuresta vastausjoukosta tyypilliset virheelliset ratkaisut, jotka voitaisiin arvioida automaattisesti yhtenä eränä. Toisena virhetarkastelun tavoitteena oli vertailla keskenään eri analyysimenetelmiä ja tehdä päätelmiä niiden luotettavuudesta [87].

5.4 Aiheen rajaus

Tutkimuksessa kerättiin materiaalia enemmän kuin oli tämän työn yhteydessä järkevää käsitellä. Aihepiirin rajauksessa ryhmien vertaamiseksi tehdyn toisen ohjelmallisen laskennan käsittely jäi tässä työssä vähäiseksi, sen tuloksiin viitataan lyhyesti tulokset -osuudessa. Samoin kyselytutkimuksen osuus rajattiin lähinnä virhetarkastelua ja sähköisen opiskelun onnistumisen arviointia tukevaksi tarkasteluksi. Työssä keskityttiin pääasiassa virhetarkastelun tuloksiin ja eri laskentamenetelmien vertailuun.

6 TULOKSET

Tässä luvussa tarkastellaan valituissa kertaustehtävissä tehtyjä virheellisiä ratkaisuja. Jokaisesta tehtävätyypistä esitellään ensin kaikki tehtäväsarjan tehtävät. Seuraavaksi esitellään tarkasteluun valituista tehtävistä ohjelmallisesti lajitellut keskenään ekvivalentit virheelliset ratkaisut. Tarkasteluun valittujen tehtävien virheitä on laskettu myös käsin, ja eri menetelmillä löydettyjä tuloksia verrataan keskenään ja todetaan eri tarkastelutapojen edut ja puutteet. Vertailun tulosten ja käsinkirjattujen tulosten perusteella tehdään päätelmät tehtävissä onnistumisesta.

Ohjelmallisissa lajitteluissa ja käsinkirjauksessa saatiin eri määrät tekijöitä ja ratkaisuyrityksiä. Tarkasteluun valittiin *itseisarvotehtävä*, tehtävät *ensimmäisen asteen yhtälöistä*, *toisen asteen yhtälö* ja *trigonometrinen yhtälötehtävä*.

6.1 Itseisarvotehtävät

Opiskelijat vastasivat kertaustehtävien itseisarvotehtäviin vaihtelevasti. Lähempään tarkasteluun valittiin tehtävä 2–3 (Katso taulukko 6.1). Toisen ohjelmallisen laskennan mukaan 58 opiskelijaa onnistui tehtävässä. Virheiden käsinkirjauksessa yhteensä 50 opiskelijaa teki 265 yritystä, joista 172 huomioitiin virhelaskennassa.

Tehtävästä kolme saatiin kattavasti näkemys opiskelijoiden tekemistä virhetyypeistä, eikä muiden tehtävien tarkastelua katsottu tarpeelliseksi. Taulukossa 6.1 esitellään kertauksen itseisarvotehtävät.

Taulukko 6.1. *Perustaitojen kertauksen itseisarvotehtävät.*

2	Tehtävä
2-1	Ratkaise luvun $a - b$ itseisarvo
2-2	Ratkaise luvun $5 - 13$ itseisarvo
2-3	Ratkaise luvun$\sqrt{3} - 6$ itseisarvo
2-4	Ratkaise luvun 6-ln$\left(e^{11}\right)$ itseisarvo
2-5	Ratkaise luvun $\sin\left(e^{7}\right) + 6 - 7\sqrt{2}$ itseisarvo

Itseisarvotehtävissä tehtiin paljon syntaksivirheitä. Erityisesti erikoismerkkien kirjoittaminen tehtävissä 2–3, 2–4 ja 2–5 tuotti syntaksivirheitä ratkaisuihin. Tehtävissä 2–1 ja 2–2 tarkasteltiin pelkkiä lukuarvoja ja niissä tehdyt virheet olivat lähinnä pieniä lasku- tai näppäilyvirheitä.

Opiskelijat tekivät tehtävässä 2–3 suuren määrän erilaisia ratkaisuyrityksiä. Ohjelmallisessa laskennassa oli yhteensä 25 erilaista ratkaisua, joille löytyi samaa tarkoittava ratkaisu vastausten joukosta. Oikeita ratkaisuja tehtiin 128.

Varsinaisia virheitä aiheutti lukujen suuruuden vertaamisessa esimerkiksi neliöjuuren itseisarvon arvioiminen.

Taulukossa 6.2 esitetään kaikki ohjelmallisen laskennan keskenenään ekvivalentit vastaukset ja niiden määrät. Vastausten virheet luokiteltiin Gill & Greenhow [17] virhetyyppien mukaan.

Taulukko 6.2. *Tehtävän 2-3 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

Vastaus	Virhetyyppi	Luokittelu	Määrä
$6-\sqrt{3}$	Oikea vastaus		128
4.267949	Tulkintavirhe	Desimaalilukuvastaus	14
$\sqrt{3}-6$	Laskuvirhe	Etumerkkivirhe	13
$3^2-6=3$	Määritelmävirhe	Määritelmää ei tunneta	10
$\sqrt{3}+6$	Laskuvirhe	Etumerkkivirhe	7
33	Määrittelemätön		5
$4,26$	Tulkintavirhe	Desimaalilukuvastaus	5
$4,5$	Laskuvirhe	Useita, desimaalilukuvastaus	4
$\sqrt{3}+12\cdot\sqrt{3}-6^2$	Määritelmävirhe	Puutteelliset tiedot	4
$\sqrt{3}$	Tieto	Puutteelliset tiedot	4
$7,732051$	Tulkintavirhe	Desimaalukuvastaus	3
$\sqrt{3}+12\cdot\sqrt{3}+6^2$	Tieto.	Puutteelliset tiedot	3
$4,267$	Tulkintavirhe	Desimaalilukuvastaus	2
$-4,5$	Tulkintavirhe	laskuvirhe	2
$7,5$	Laskuvirhe	Useita, desimaalilukuvastaus	2
$3^{\frac{1}{3}}+6$	Tieto	Puutteelliset tiedot	2
$12\cdot\sqrt{3}+39$	Tieto	Puutteelliset tiedot	2
$-6\sqrt{3}$	Tieto	Puutteelliset tiedot	2
$4,26795$	Tulkintavirhe	Desimaalilukuvastaus	2
27	Määrittelemätön		2
$-\left(\sqrt{3}+6\right)$	Laskuvirhe	Etumerkkivirhe	2
$6-9=-3$	Määritelmävirhe	Puutteelliset tiedot	2
$\sqrt{-3}$	Tieto	Puutteelliset tiedot	2
$\sqrt{33}$	Tieto	Puutteelliset tiedot	2
$4,2679$	Tulkintavirhe	Desimaalilukuvastaus	2

Taulukossa 6.2 on ohjelmallisen laskennan keskenään ekvivalentit ratkaisut, niiden määrät ja virhetyypit. Vaikka ohjelmallisesti tehtyyn laskentaan sisältyy virhetekijöitä,

Taulukosta 6.2 saadaan kuitenkin realistinen kuva vastausten jakautumisesta.

Taulukkoon 6.3 on koottu taulukon 6.2 ohjelmallisesti lajitellut vastaukset niin, että samankaltaiset virheelliset vastaukset on luokiteltu samaan luokkaan ja voidaan arvostella yhteisellä palautteella. Luokittelukriteerejä on helppo tarkentaa ja lisätä tai yhdistää luokkia tarvittavan tarkkuuden mukaan. Kun tieto- ja määritelmävirheet on tulkittu samaan luokkaan ja käytettyyn luokitteluun [17] on lisätty luokka määrittelemätön, virheet on voitu luokitella neljään luokkaan.

Taulukko 6.3. *Taulukon 6.2 samankaltaiset virheet koottuna yhteen.*

Virhetyyppi	Luokittelu	Määrä
Tulkintavirhe	Oikeanlainen, desimaalilukuvastaus	29
Laskuvirhe	Oikeanlainen vastaus, mutta etumerkkivirhe	27
Tieto	Puutteelliset tiedot	25
Määrittelemätön		17

Taulukosta 6.3 puuttuvat sievennys- ja syntaksivirheet, koska ohjelma on huomioinut vain laskennan tuloksena saadun vastauksen.

Taulukossa 6.4 on virheiden käsinkirjauksen tulokset, jossa myös syntaksi- ja sievennysvirheet on voitu luokitella. Samalla huomattiin, että syntaksivirheiden suuri määrä on tunnusomaista itseisarvotehtäville. Toisaalta syntaksivirheet ovat usein kasaantuneet samoille vastaajille.

Taulukko 6.4. *Tehtävän 2-3 käsin kirjatut ja luokitellut [17] vastaukset.*

Virhetyyppi	Luokittelu	Määrä
Tulkinta	Oikeanlainen, desimaalilukuvastaus	37
Tieto	Puutteelliset tiedot	31
Syntaksi		27
Määrittelemätön		19
Laskuvirhe	Oikeanlainen vastaus, mutta etumerkkivirhe	13
Laskuvirhe	Sievennys kesken	10
Keskeytti		7

Taulukossa 6.5 esitetään vastaajien onnistuminen tehtävässä 2–3 vertailun ja käsinkirjauksen perusteella. Vertailu ei huomioinut tehtyjen vastausyritysten kokonaismäärää.

Taulukko 6.5. *Opiskelijoiden onnistuminen tehtävässä 2–3.*

Laskenta	Vastaajia	Yrityksiä	Ei onnistunut	Onnistui
Vertailu	67	–	19	48
Käsinkirjaus	50	157	9	41

Käsinkirjauksessa jokainen vastaus voitiin käsitellä erikseen ja saadut luvut ovat melko tarkkoja. Joka tapauksessa molemmat laskennat antavat melko samanlaisen kuvan tehtävässä onnistumisesta.

Ohjelman toiminnallisuudesta on aiheutunut jonkin verran virheitä. Esimerkiksi ohjelma hyväksyy vastauksen $|\sqrt{3}-6|$, mutta sen voi rajata pois oikeiden vastausten joukosta sievennysasteen pienentämisellä. Tehtävien ohjelmoijan asettama pienempi sievennysasteen vaatimus (kompleksisuusarvo) ei hylkää vastausta kokonaan, vaan kehottaa sieventämään lisää. Samalla myös sinänsä oikea vastaus $-\sqrt{3}+6$ rajautuu pois oikeiden vastausten joukosta. Näiden tekijöiden merkitys tutkimustuloksen kannalta on hyvin pieni, mutta ne kannattaa ottaa huomioon tehtävien laadinnassa ja ohjeistuksessa tulevaisuudessa.

Vaikka taulukoiden 6.2 ja 6.3 ohjelmallisesti lajiteltuihin tuloksiin sisältyykin virhetekijöitä, niistä nähdään kuitenkin tehtävässä tehdyt toistuvat virheet ja virhetyypit. Käsinkirjaamalla myös sievennys- ja syntaksivirheet on voitu huomioida. Kokonaisuutena eri laskennat antoivat realistisen kuvan kaikista virhetyypeistä.

6.2 Ensimmäisen asteen yhtälö

Ensimmäisen asteen yhtälötehtävät ovat helppoja perustehtäviä. Taulukossa 6.6 esitellään kertauksen ensimmäisen asteen yhtälötehtävät.

Taulukko 6.6. *Perustaitojen kertauksen ensimmäisen asteen yhtälötehtävät.*

5	Ensimmäisen asteen yhtälötehtävät
5–1	Ratkaise x yhtälöstä $x+4=4$
5–2	Ratkaise x yhtälöstä $x+5=2x$
5–3	Ratkaise x yhtälöstä $x+3=2x+4$
5–4	Ratkaise x yhtälöstä $9x+7=5x+3$
5–5	Ratkaise x yhtälöstä $ex+6=\pi x+8$

Eniten ratkaisuyrityksiä tehtiin tehtävään 5–5, jossa suurin osa tehtävän virheellisistä ratkaisuyrityksistä aiheutui syntaksivirheitä aiheuttavista erikoismerkeistä. Esimerkkitehtäväksi ja tarkasteluun valittiin tehtävä 5–4, koska se edusti virheiltään keskimääräistä ensimmäisen asteen yhtälötehtävää, ja oli toisaalta hyvä tehtävä tehtävien satunnaistamisen esittelyyn. Taulukossa 6.7 on kaikkien vastaajien ohjelmallisesti lajitellut keskenään samaa tarkoittavat vastaukset.

Taulukko 6.7. *Tehtävän 5–4 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

Vastaus	Virhetyyppi	Luokittelu	Määrä
$x=-1$	Oikea vastaus		67
$x=1$	Laskuvirhe	Etumerkkivirhe	2
$x=-\frac{4}{5}$	Laskuvirhe		2
$x=\frac{5}{2}$	Laskuvirhe	Useita	2

Useimmat (28/37) saivat tehtävän oikein ensi yrittämällä, ja vain yksi jätti tehtävän kesken. Ohjelma on tulkinnut luvut -1 ja $-\frac{4}{4}$ oikeiksi vastauksiksi. Tämä ei sinänsä ole ongelma, mutta ohjelma olisi tulkinnut esimerkiksi $-\frac{9}{9}$ olevan oikein. Sievennysvaatimusta tulisi pienenentää yhdellä, jolloin ohjelma hyväksyy vain sievennetyn vastauksen.

Tehtävässä 5–4 virheitä laskettiin myös käsin. Taulukossa 6.8 esitetään virheiden käsinkirjauksessa löydetyt ja luokitellut virheet. Tulokset ovat samansuuntaisia kuin ohjelmallisessa laskennassa. Lisäksi syntaksi- ja sievennysvirheet on voitu huomioida.

Taulukko 6.8. *Tehtävässä 5–4 käsinkirjauksessa havaittujen virhetyyppien luokittelu.*

Virhetyyppi	**Luokittelu**	**Määrä**
Laskuvirhe	Oikein, sievennysvirhe	6
Laskuvirhe	Muuttujalukujen laskuvirhe	3
Laskuvirhe	Etumerkkivirhe	2
Tieto	Useita laskuvirheitä	2
Syntaksi	Oikein, mutta syntaksivirhe	2

Taulukossa 6.9 koottuna opiskelijoiden onnistuminen tehtävässä 5–4 vertailun ja käsinkirjauksen perusteella.

Taulukko 6.9. *Opiskelijoiden onnistuminen tehtävässä 5–4.*

Laskenta	**Vastaajia**	**Yrityksiä**	**Ei onnistunut**	**Onnistui**
Vertailu	54	–	2	52
Käsinkirjaus	37	52	1	36

Kaikki ensimmäisen asteen yhtälötehtävät osattiin vertailun mukaan hyvin. Vain tehtävässä 5–5 erikoismerkit aiheuttivat ongelmia, ja noin neljännes opiskelijoista ei onnistunut tehtävässä.

Tehtävän 5–4 virhetyypit edustavat tyypillisiä virheitä ensimmäisen asteen yhtälötehtävissä. Useimmiten virheitä aiheutui muuttujien laskutoimituksissa ja etumerkeissä. Syntaksivirheitä esiintyi jonkinverran, mutta tehtävät olivat kuitenkin niin yksinkertaisia, että useimmat saivat ne lopulta oikein.

6.3 Toisen asteen yhtälö

Toisen asteen yhtälöitä harjoiteltiin tehtäväsarjassa 6. Virhetarkasteluun valittiin helppo tehtävä 6–1 ja yleisen toisen asteen yhtälön ratkaisumenetelmän hallintaa edellyttävät tehtävät 6–3 ja 6–4. Taulukossa 6.10 esitellään sarjan tehtävät.

Tehtävät olivat useampivaiheisia, ja erilaisia virhevariaatioita tehtiin enemmän kuin edellisissä kahdessa tarkastelussa. Valituissa tehtävissä tehtyjä ratkaisuja on lajiteltu ohjelmallisesti ja joidenkin tehtävien ratkaisuja on tarkasteltu myös kirjaamalla virheitä käsin.

Vaikka ensimmäisen tehtävä on helppo, sen mukaan ottaminen tarkasteluun on perustel-

Taulukko 6.10. *Perustaitojen kertauksen toisen asteen yhtälötehtävät.*

6	Toisen asteen yhtälö
6–1	Ratkaise toisen asteen yhtälö $x^2 - 9x = 0$.
6–2	Ratkaise toisen asteen yhtälön $2x^2 - 9x + 3 = 0$ diskriminantti.
6–3	Ratkaise $x^2 + 9x + 20 = 0$.
6–4	Ratkaise $-9x2 - 24x - 7 = 0$. Kannattaa käyttää tässä laskinta diskriminantin laskemiseen.
6–5	Ratkaise $8x^2 - 2x - 10 = 0$.

tua taulukon 6.11 perusteella. Taulukosta nähdään, että yleisimmät virhetyypit ovat puutteita tiedoissa ja laskuvirheitä.

Taulukko 6.11. *Tehtävän 6–1 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

6–1 vastaus	Virhetyyppi	Luokittelu	Määrä
$x = 0 \vee x = 9$	Oikea vastaus		106
$x = 0$	Tieto	vain toinen juuri	16
$x = 9$	Tieto	vain toinen juuri	13
$x^2 = 9x \Leftrightarrow x = 3$	Tieto	Puutteelliset tiedot	9
$x = 0 \vee x = 3$	Laskuvirhe	Oikeanlainen vastaus	2
$x = \frac{9+\sqrt{77}}{2} \vee$ $x = \frac{9-\sqrt{77}}{2}$	Laskuvirhe	Oikeanlainen vastaus	8
$x = \frac{9+(\sqrt{9^2-4\cdot1\cdot0}}{2\cdot1}$	Tieto	Puutteelliset tiedot	2

Taulukko 6.12. *Tehtävän 6–1 samankaltaiset virheet koottuna yhteen.*

Virhetyyppi	Luokittelu	Määrä
Tieto	Vain toinen juuri	19
Tieto	Puutteelliset tiedot	11
Laskuvirhe	Oikeanlainen vastaus	10

Yleisin virhe tehtävässä oli, että huomattiin vain toisen asteen yhtälön toinen juuri. Kertausohjelmassa tehtävien yrityskertoja ei ole rajoitettu. Jokainen teki tehtävässä keskimäärin 4 yritystä (käsinkirjauksessa laskettiin yhteensä 135 hyväksyttyä yritystä) ennen kuin onnistuivat. Käsinkirjauksen mukaan lähes kaikki (28/32) kaikki saivat tehtävän lopulta oikein. Keskeyttäneistäkin 3 sai tehtävästä lopulta oikean vastauksen, vaikka yrityksiä tehtiin niin paljon, että tehtävä tulkittiin keskeytyneeksi. Tehtävässä 6–3 (Taulukko 6.14) erilaisia virheitä esiintyy paljon vähemmän.

Käsinkirjauksen perusteella tässäkin tehtävässä tehtiin paljon syntaksivirheitä. Suurin osa virheellisistä ratkaisuyrityksistä oli tässä tehtävässä syntaksivirheitä ja lajittelussa havaittuja tieto-virhetyypin virheitä.

Taulukossa 6.14 esitetään tehtävän 6–3 lajittelun tulokset. Taulukosta nähdään, että erilaisia virheitä on tehty vähemmän verrattuna ensimmäiseen tehtävään. Nyt suurin osa

Taulukko 6.13. *Opiskelijoiden onnistuminen tehtävässä 6–1.*

Laskenta	Vastaajia	Yrityksiä	Ei onnistunut	Onnistui
Vertailu	50	–	3	47
Käsinkirjaus	32	135	4	28

virheistä on sellaisia, jotka voidaan luokitella samaan virheluokkaan.

Taulukko 6.14. *Tehtävän 6–3 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

6–3 vastaus	Virhetyyppi	Luokittelu	Määrä
$x = -4 \vee\ x = -5$	Oikea vastaus		76
$x = 4 \vee\ x = 5$	Laskuvirhe	Etumerkkivirhe	8
$x = 5$	Tieto	Vain toinen juuri	4
$x = -4$	Tieto	Vain toinen juuri	4

Tehtävässä 6–3 useimmat hallitsevat tehtävän ratkaisutavan. Tehdyt virheet ovat sellaisia, että tehtävässä on käytetty oikeanlaista ratkaisutapaa. Tehtävätyypin tyypillisimmät virheet ovat väärä etumerkki tai vastauksena on annettu vain toisen asteen yhtälön toinen juuri.

Tehtävien käsintarkastelussa huomattiin, että edelleen tehtävissä on paljon syntaksi- ja sievennysvirheitä. Nyt ohjelman sievennysvaatimus oli sopiva, koska keskeneräiset vastaukset eivät kelvanneet ohjelmalle. Koska tehtävät vaikeutuivat loppua kohti, tarkastellaan vielä tehtäviä 6–4 ja 6–5. Tehtävät olivat samantyyppisiä, mutta nyt tehtävän ratkaisut olivat murtolukuja, ja virhetyypit lisääntyivät.

Taulukko 6.15. *Opiskelijoiden onnistuminen tehtävässä 6–3.*

Laskenta	Vastaajia	Yrityksiä	Ei onnistunut	Onnistui
Vertailu	49	–	7	42
Käsinkirjaus	27	62	6	21

Tehtävässä 6–4 (Taulukko 6.16) erilaisia virheitä on paljon enemmän kuin tehtävässä 6–3. Nyt oikean vastauksen murtolukujuuret tuottavat virheinä desimaalilukuvastauksia ja muita virhetyyppejä, joista kuitenkin näkee, että vastaus on oikeansuuntainen. Tehtävän 6–4 samankaltaiset virheet on luokiteltu samaan luokkaan taulukossa 6.17.

Tehtävässä 6–1 oli paljon sekalaisia virheitä, mutta kolmessa viimeisessä tehtävässä virheet ovat samantyyppisiä. Vastauksista näkee, että tehtäviä on ratkaistu oikealla tavalla, mutta selvästi laskurutiini on vielä puutteellinen. Yhtälötehtäviä olivatkin ratkoneet lähinnä ryhmän MAT-123 opiskelijat. Joka tapauksessa suoritukset ovat parantuneet harjoitellessa. Tehtävässä 6–4 (taulukko 6.16 on paljon erityyppisiä virheitä, mutta tehtävässä 6–5 (Taulukko 6.19) esiintyy pelkästään tehtävätyypille ominaisia virheitä.

Taulukko 6.16. *Tehtävän 6–4 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

6–4 vastaus	Virhetyyppi	Luokittelu	Määrä
$x = -\frac{7}{3} \vee x = -\frac{1}{3}$	Oikea vastaus		65
$x = \frac{24+\sqrt{-24^2-4\cdot(-9)\cdot(-7)}}{-18} \vee$ $x = \frac{24+\sqrt{(-24)^2-4\cdot(-9)\cdot(-7)}}{-18}$	Laskuvirhe,	Useita virheitä	5
$x = -\frac{7}{3} \vee x = \frac{1}{3}$	Laskuvirhe	Etumerkkivirhe	5
$x = \frac{7}{3} \vee x = \frac{1}{3}$	Laskuvirhe	Etumerkkivirhe	3
$x = -0.3333 \vee x = -2.3333$	Tulkinta	Desimaalivastaus	3
$x = -\frac{42}{18}$	Tieto	Useita virheitä	3
$x = \frac{42}{-9} \vee x = \frac{6}{-9}$	Laskuvirhe	Sievennys	3
$x = -3 \vee x = -21$	Laskuvirhe	Useita	2
$x = 3.5 \vee 5.5$	Tulkinta	Useita virheitä	2
$x = 0$	Määrittelemätön		2
$x = \frac{24^2+\sqrt{324}}{-18}$	Tieto	Useita virheitä	2
$x = \frac{(9+18)}{-18}$	Tieto	Useita virheitä	2
$x = 21 \vee x = 3$	Laskuvirhe	Useita virheitä	2
$x = -\frac{7}{6} \vee x = -\frac{2}{3}$	Laskuvirhe	Useita virheitä	2

Taulukko 6.17. *Tehtävän 6–4 samankaltaiset virheet koottuna yhteen.*

Virhetyyppi	Luokittelu	Määrä
Tieto	Useita virheitä	18
Laskuvirhe	Etumerkkivirhe	8
Tulkinta	Desimaalivastaus	3
Laskuvirhe	Sievennys	3
Määrittelemätön		2

6.4 Trigonometrian yhtälötehtävät

Edellisten kertausosioiden tehtäväsarjat olivat melko helppoja perustehtäviä, eikä keskiverto-opiskelijalla perustaitotestien tilastojen [45] perusteella ole ollut niissä vaikeuksia. Trigonometrian tehtävät taas ovat perinteisesti olleet opiskelijoille vaikeampia. Trigonometria, derivaatta ja logaritmit ovat olleet koko perustaitotestin historian ajan perustaitotestin heikoiten osattuja aihealueita [45].

Sama ilmiö nähdään myös tässä työssä käsitellyssä aineistossa. Vertailun perusteella itseisarvoja ja yhtälötehtäviä ovat tehneet pääasiassa taustaltaan vähiten matematiikkaa opiskelleet, ja trigonometrian tehtäviä ovat harjoitelleet lähes yksinomaan matemaattisesti suuntautuneet opiskelijat. Tämän perusteella voi päätellä, että trigonometrian tehtäviä pidetään vaikeampina, ja harjoitustehtävistä annettavien lisäpisteiden saamiseksi pisteitä on kerätty helpommista tehtävistä. Myös opiskelijoille tehdyssä kyselyssä trigonometrian

Taulukko 6.18. *Opiskelijoiden onnistuminen tehtävässä 6–4.*

Laskenta	Vastaajia	Yrityksiä	Ei onnistunut	Onnistui
Vertailu	47	–	3	44
Käsinkirjaus	28	135	3	25

Taulukko 6.19. *Tehtävän 6–5 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

6-5 vastaus	Vastaus	Virhetyyppi	%
$x = \frac{5}{4} \vee x = -1$	Oikea vastaus		$82,8$
$x = \frac{5}{4} \vee x = 1$	Laskuvirhe		$8,7$
$x = \frac{5}{4} \vee x = -\frac{8}{9}$	Laskuvirhe		$6,3$
Tyhjä	Määrittämätön		$3,1$

tehtäviä pidettiin vaikeina.

Tässä kappaleessa tarkastellaan trigonometrian yhtälötehtävän 10-1 vastauksia. Taulukossa 6.20 listattuna perustaitojen kertauksen trigonometrian tehtävät.

Taulukko 6.20. *Perustaitojen kertauksen trigonometrian yhtälötehtävät.*

10	Trigonometriset yhtälöt
10.1	Laske yhtälön $\sin(7x) = \cos(x)$ ratkaisu radiaaneissa? Olkoon $x \in 0, 2\pi$. Etsi taulukosta sinin ja kosinin palautuskaavat ja muunna niin, että yhtälön molemmat puolet ovat muotoa $\sin(a) = \sin(b)$. Anna pienin positiivinen ratkaisu.
10.2	Laske yhtälön $\cos(8x) + 1 = 0$ ratkaisu radiaaneissa. Anna jälleen pienin positiivinen ratkaisu. Vinkki: Milloin $\cos$ saa arvon -1 ?
10.3	Laske lausekkeen $2\sin(-\frac{\pi}{3}) + \cos\left(\frac{2\pi}{3}\right)$ arvo. Kulmien tarkat arvot muistikolmioista tai trigonometristen funktioiden taulukosta. Tässä tehtävässä kirjoita vastauksesi suoraan yhtäsuuruusmerkin perään.
10.4	Mikä on kulman x suuruus radiaaneissa, kun $\tan(8x) = 3 - \sqrt{7}$? Tangentin arvoja löytyy trigonometrian taulukoista. Anna pienin positiivinen ratkaisu.
10.5	Laske yhtälön $\sin(6x) = \sin(3x + \pi 2)$ ratkaisu. Trigonometriset funktiot ovat jaksollisia, niiden kuvaajat toistuvat samanlaisina jaksoittain. Esim. sinille ja kosinille sama kulma toistuu 2π:n välein. Anna vastauksessasi pienin positiivinen ratkaisu radiaaneina.

Trigonometriset yhtälöt -tehtäväsarjaan saatiin huomattavan paljon sekä oikeita että vääriä vastausyrityksiä. Käsinkirjauksen mukaan tehtävää teki kaikkiaan 127 opiskelijaa, siihen tehtiin yhteensä 396 yritystä ja 13 tulkittiin keskeyttäneeksi. Lähes kaikki (124) saivat tehtävän kuitenkin lopulta oikein, vaikka tehtävä tulkittiinkin käsinkirjauksessa keskeytyneeksi. Yrityskertojen määrä ennen oikeaa vastausta oli 3,8 ja ensimmäisellä kerralla tehtävän sai oikein 40 opiskelijaa.

Taulukko 6.21. *Tehtävässä 10-1 ohjelmallisesti järjestetyt ja luokitellut [17] vastaukset.*

10-1 vastaus	Virhetyyppi	Luokittelu	Määrä
$x = \frac{\pi}{16}$	Oikea vastaus		266
$x = \frac{\pi}{8}$	Laskuvirhe	Oikeanlainen vastaus	15
$x = \frac{\pi}{4}$	Laskuvirhe	Oikeanlainen vastaus	10
$x = \frac{\pi}{12}$	Laskuvirhe	Oikeanlainen vastaus	8
$x = \frac{1}{28} * \pi$	Laskuvirhe	OIkeanlainen vastaus	5
$x = \frac{\pi}{2}$	Laskuvirhe	Oikeanlainen vastaus	4
$x = \frac{\pi}{18}$	Laskuvirhe	Oikeanlainen vastaus	3
$x = \frac{\pi}{6}$	Laskuvirhe	Oikeanlainen vastaus	3
$x = 0$	Tieto	Puutteelliset tiedot	2
$x = -\frac{\pi}{4}$	Laskuvirhe	Oikeanlainen vastaus	2
$x = \pi$	Tieto	Puutteelliset tiedot	2
$x = \frac{\pi}{28} + 2 * \pi$	Tieto	Puutteelliset tiedot	2
$x = \frac{\pi+4*\pi}{16}$	Tieto	Puutteelliset tiedot	2
$x = \sqrt{\frac{1}{7}}$	Määrittelemätön		2

Tehtävässä 10–1 taulukossa 6.21 on paljon oikeanlaisia virheellisiä vastauksia, joissa on väärä kerroin. Ohjelmallisesti lajiteltujen vastausten virheet on helppo jakaa kolmeen luokkaan, joille voi antaa yhteisen palautteen. Tehtävässä tuli myös paljon vastauksia, joita ohjelma ei osannut käsitellä. Käsinkirjauksessa syntaksivirheiksi tulkittavia oli melko paljon, n. 20 % kaikista vastauksista.

Taulukko 6.22. *Tehtävässä 10-1 ohjelmallisesti järjestetyt ja luokitellut [17] samankaltaiset vastaukset koottuna yhteen.*

10-1	Virhetyyppi	Luokittelu	Määrä
	Laskuvirhe	Oikeanlainen vastaus,	45
	Tieto	Puutteelliset tiedot	8
	Määrittelemätön		2

Taulukossa 6.23 tehtävän 10–1 vastaukset luokiteltuna niin, että samankaltaiset oikeanlaiset vastaukset on kerätty yhteen. Taulukosta nähdään, että suurin osa (82%) virheistä voidaan luokitella samaan luokkaan.

Tehtävässä 10–1 virheiden Käsinkirjaus antaa paremman kokonaiskuvan tehtävässä tehdyistä virheistä. Tehtävässä on tehty paljon syntaksi ja sievennysvirheitä. Ohjelmallisen lajittelun tuloksena on kuitenkin löydetty tyypillisimmät tehtävässä tehdyt virheet. Kun huomioidaan käsinkirjauksessa löydetyt syntaksi- ja sievennysvirheet, saadaan realistinen kuva tehtävässä tehdyistä virheistä.

Taulukko 6.23. *Tehtävässä 10-1 käsinkirjauksessa havaittujen virhetyyppien luokitellut [17] samankaltaiset vastaukset koottuna yhteen.*

Virhetyyppi	**Luokittelu**	**Määrä**
Laskuvirhe	Oikeanlainen vastaus	85
Syntaksi		79
Tieto	Puutteelliset tiedot	70
Sievennys		46
Määrittelemätön		2

6.5 Kyselytutkimuksen tuloksia

Kyselytutkimuksella selvitettiin opiskelijoiden kokemuksia kertausohjelmasta ja sen uusista ominaisuuksista. Taulukkoon 6.24 sivulla koottu ohjelman tärkeimpiä ominaisuuksia koskevia neliportaisella Likert-asteikolla arvioituja kysymyksiä. Vastausvaihtoehdot ovat 0 = ”täysin eri mieltä”, 1 = ”jokseenkin eri mieltä”, 2 = ”jokseenkin samaa mieltä” ja 3 = ”täysin samaa mieltä” [86]. Vastaajia Likert-asteikollisiin kysymyksiin oli 156.

Mahdollisuus tehdä tehtäviä paikasta ja ajasta riippumatta sai suurimman joukon mielestä (87%, 85%) myönteisimmän arvion ja vastausten keskiarvot olivat 2,83 ja 2,81. Mahdollisuus kirjoittaa välivaiheita oli myös pidetty ominaisuus, ja se sai keskiarvon 2,45. Kommenttien kirjoittamista ei mielletty yhtä tärkeäksi, ja sen keskiarvo oli 1,88.

Taulukko 6.24. *Kertausohjelman uusia ominaisuuksia koskevat kysymykset.*

Vastaus	**0**	**1**	**2**	**3**	**Keskiarvo**
Oli hyvä, että saatoin tehdä kertaustehtäviä milloin vain.	2	1	18	135	2,83
Oli hyvä, että saatoin tehdä tehtäviä missä vain.	2	3	18	133	2,81
Ohjelman mahdollisuus kirjoittaa välivaiheita näkyviin on hyvä ominaisuus.	6	7	54	89	2,45
Oli hyvä, että ohjelmassa pystyi kirjoittamaan kommentteja.	12	42	54	48	1,88
Sain korjattua vastaustani ohjelman antaman palautteen perusteella.	11	75	80	27	1,79
Ohjelman antama palaute oli mielestäni riittävää.	24	68	57	7	1,3

Suurin osa onnistui korjaamaan vastaustaan ohjelman antamien vihjeiden perusteella ainakin jonkin verran (keskiarvo 1,79), mutta ohjelman antama palaute ei ollut opiskelijoiden mielestä riittävää (Keskiarvo 1,3). Lisäksi kertaustehtävät koettiin hyväksi tavaksi kerrata matematiikan perustaitoja (Keskiarvo 2,31).

Sanallisilla vastauksilla haluttiin selvittää opiskelijoiden kokemuksia ohjelmasta omin sanoin. Sanallisiissa kysymyksissä vastaajia oli kaikkiaan 149. Ohjelma ei pakottanut vastaamaan kaikkiin kohtiin, joten vastaajamäärä vaihtelee hieman kysymyskohtaisesti.

- Mitä hyvää kertausohjelmassa oli?
- Mitä huonoa kertausohjelmassa oli?

Kysymyksiin tuli suuri määrä erilaisia vastauksia. Vastauksista etsittiin yhteisiä piirteitä niin, että saman luokan vastauksista löydettiin yhteinen kuvaava sana. Ensimmäiseen kysymykseen, "Mitä hyvää kertausohjelmassa oli", löydettiin luokat *helppous*, *monipuolisuus*, *hyöty*, *joustavuus* ja *palaute*. Näihin luokkiin mahtui 79 % vastauksista.

Taulukko 6.25. *Luokitellut vastaukset kysymykseen "Mitä hyvää kertausohjelmassa oli?".*

Helppous	Monipuolisuus	Hyötynäkökulma	Joustavuus	Palaute
30	29	28	19	11

Taulukossa 6.25 on viiteen luokkaan luokitellut kyselytutkimuksen sanalliset positiiviset vastaukset. Lisäksi kysymykseen saatiin 18 vastausta, joita ei saatu luokiteltua viiteen pääryhmään. Vastaajia oli kaikkiaan 135.

Toiseen kysymykseen, "Mitä huonoa kertausohjelmassa oli", saatiin 140 vastausta. Kielteiset mielipiteet luokiteltiin samalla tavalla kuin edellisen kysymyksen vastaukset toistuvien kuvaavien sanojen mukaan. Tarkasteluun valittiin viisi useimmin toistuvaa vastausta. Nyt löydettiin luokat *syntaksi*, *epäselvä*, *välivaiheet*, *ohjeistus*, ja *ei hyväksy vastausta*. Näihin luokkiin mahtui 76 % vastauksista ja luokittelemattomia vastauksia oli 33. Huonoiksi koetut ominaisuudet ominaisuudet luokiteltuna taulukossa 6.26.

Taulukko 6.26. *Luokitellut vastaukset kysymykseen "Mitä huonoa kertausohjelmassa oli?".*

Syntaksi	Palaute	Välivaiheet	Ohjeistus	Ei hyväksy vastausta
57	24	11	10	5

Kertausohjelman huonoista puolista selvästi yleisimmin on mainittu syntaksi (41 % kaikista vastauksista). Käsinkirjauksessakin havaittiin, että syntaksivirheet ovat monille yleisin virhetyyppi. Toisaalta useimmat saivat tehtävät kuitenkin tehtyä. Käsinkirjauksessa useiden syntaksivirheiden takia keskeytyneiksi merkityt tehtävät saatiin lopulta tehtyä oikein, vaikka yrityskertoja saattoi kertyä paljon keskimääräistä enemmän.

Kertausohjelman kehittäjän näkökulmasta erityisesti huonoimmiksi koettujen ominaisuuksien löytäminen on hyödyllistä. Kertausohjelman toteutuksen kannalta erityisesti syntaksivirheisiin liittyvät kommentit, käsinkirjauksessa havaittu syntaksivirheiden suurehkä määrä ja kertausohjelman käytettävyytteen liittyvät kommentit auttavat ohjelman kehittäjiä parantamaan ainakin tehtävien ohjeistusta ja muokkaamaan tehtävänantoja uudelleen.

7 POHDINTA JA YHTEENVETO

Tässä työssä tarkasteltiin opiskelijoiden sähköisessä muodossa tekemiä ratkaisuja matematiikan tehtäviin ja niiden automaattista arviointia. Tutkimus tehtiin Tampereen teknillisen yliopiston perustaitotestin kertauksen yhteydessä. Työssä tutkittiin ensisijaisesti, kuinka opiskelijoiden vastauksia voidaan lajitella niin, että samankaltaiset virheelliset ratkaisut voitaisiin luotettavasti kerätä samaan kategoriaan ja arvostella tulevaisuudessa automaattisesti.

Perustaitojen testin kertauksessa ensimmäistä kertaa käytetty MathCheck-kaavatarkastin on uudenlainen opiskeluohjelma, joka mahdollistaa vastausten kirjoittamisen kokonaisuudessaan välivaiheineen. Aiemmat ohjelmat eivät ole sallineet opiskelijan kirjoittaa vastausta harjoitus- tai testikysymykseen välivaiheineen yhdellä ja samalla ohjelmalla niin, että välivaiheet ovat osa arvioitavaa vastausta. MathCheck osaa myös antaa palautetta ja ohjaa opiskelijaa tehtävän ratkaisussa ilman opettajaa.

Toissijaisena tutkimuskohteena oli, kuinka opiskelijat onnistuivat sähköisten kertaustehtävien tekemisessä, ja miten valmiita uudet opiskelijat olivat omaksumaan pelkästään sähköisen opiskelujärjestelmän matematiikan tehtävien yhteydessä. Tutkimuksessa haluttiin lisäksi kerätä tietoa uudenlaisen kertausohjelman toiminnasta.

Tutkimusprojektia esiteltäessä yleisölle osalle kuulijoista nousi huoli matematiikan opintojen sähköistymisen mielekkyydestä. Opintojen sähköistyminen on joka tapauksessa myös matematiikassa pysyvä ja vahvistuva suuntaus. Työ ei ole kannanotto opintojen sähköistymisen puolesta tai vastaan, vaan tarkoituksena oli tutkia millä tavalla sähköistyviä opintoja voitaisiin tehdä mielekkäämmiksi sekä opiskelijoille että opettajille.

Koska perustaitotestissä on hyvin erityyppisiä tehtäviä, myös niissä tehtyjä virheitä tarkasteltiin tehtäväalueittain. Löydettyjen virheiden luokitteleminen on yleensä tapauskohtaista, ja luokittelumalli kehittyy sitä käytettäessä. Koska nyt ei ollut aiempaa kokemusta matematiikan tehtävien virheiden luokittelusta, tässä työssä valittiin käyttöön yleinen malli [17], jota sovellettiin tehtävätyyppikohtaisesti.

Työn ensimmäisessä vaiheessa lajiteltiin koko kertausohjelmasta saatu vastausdata tehtävittäin niin, että kaikki samaa tarkoittavat ratkaisut kerättiin yhteen ja laskettiin. Kiinnostuksen kohteena oli, kuinka tehtävien lajittelu onnistuu ohjelmallisesti ja ovatko tulokset luotettavia. Tulosten tarkistamiseksi ja ohjelmallisessa laskennassa piiloon jäävien virheiden löytämiseksi valituissa tehtävissä tehtyjä virheitä laskettiin myös käsin.

Kyseessä oli kaikille TTY:n opiskelijoille avoin kertausohjelma, ja vastausten joukossa oli tutkimuksen kohdejoukkoon kuulumattomien vastauksia. Kertausohjelman yrityskertoja ei ollut rajoitettu, ja vastausdatan joukossa oli myös suuri määrä sekalaisia vastauksia, joilla ei ollut merkitystä tämän tutkimuksen kannalta. Nämä tekijät aiheuttivat vastausdataan epätarkkuutta, mutta tarkoituksena oli löytää tyypillisimmät virheet vastausdatan joukosta, eikä ylimääräisillä vastauksilla ollut merkitystä tämän työn tavoitteen kannalta.

Ohjelmallisen lajittelun tulosten vertaamiseksi virheitä laskettiin tarvittaessa myös käsin. Käsinkirjausta varten vastausdata puhdistettiin sekä ylimääräisistä vastaajista että vastaajien tunnistetiedoista. Siitä saatiin samankaltaisia tuloksia kuin ohjelmallisessa laskennassakin.

Käsinkirjaus oli työlästä, mutta sillä oli yksi merkittävä etu. Ohjelmallinen laskenta ei pystynyt erottelemaan esimerkiksi syntaksista aiheutuvia virheitä eikä keskeyttäneitä. Käsinkirjauksessa käytiin samat tehtävät läpi kuin ohjelmallisestikin, ja kaikki vastaukset voitiin huomioida. Kaikilla käsinkirjauksen tuloksilla ei ole tässä yhteydessä merkitystä, mutta ne kertovat esimerkiksi opiskelijoiden valmiudesta vastata tehtäviin sähköisesti. Myös ohjelman ohjeistusta voidaan parantaa käsinkirjauksen tulosten perusteella.

Itseisarvoista tarkasteltiin lähemmin tehtävää 2–3 (Katso taulukko 6.1) Itseisarvotehtävissä useimmat virheet (30 %) olivat sinänsä oikeanlaisia, mutta vastaus oli desimaalimuodossa, kun vastaukseksi odotettiin tarkkaa arvoa. Toiseksi eniten itseisarvotehtävässä tehtiin etumerkkivirheitä (28 %). Loput virheet olivat sekalaisia, määrittelemättömiä ja puutteellisistä tiedoista johtuvia virheitä (43 %). Kaikki virheet ovat listattuna taulukossa 6.2. Kuvaavaa itseisarvotehtäville oli sekalaisten vastausten suuri määrä. Käsinkirjauksessa myös syntaksivirheet korostuivat.

Ensimmäisen asteen yhtälötehtävässä 5–4 tehtiin vähän virheitä. Yhtälötehtävissä (Taulukko 6.6) tehdyt virheet olivat lähinnä pieniä laskuvirheitä ja etumerkkivirheitä. Vain tehtävässä 5–5 osalle opiskelijoita oudompi neperin luku e luku ja π kertoimena aiheuttivat ongelmia.

Toisen asteen yhtälötehtävässä (Taulukko 6.10), jossa juuret ovat kokonaislukuja, tyypilliset virheet olivat etumerkkivirheitä tai oli löydetty vain toinen juuri. Jos tehtävästä tuli vastauksena murtolukujuuria, virhevariaatioiden ja sekalaisten virheiden määrä kasvoi, samoin desimaalilukuvastausten. Erityisesti yhtälötehtävissä havaittiin selvästi, että opiskelijat pystyivät parantamaan suoritustaan mitä enemmän harjoittelivat. Useimmat myös lopulta onnistuivat tehtävissä.

Trigonometrian yhtälötehtävässä 10–1 (Taulukko 6.20) vastauksena oli π jaettuna kokonaisluvulla. Tyypillinen virheellinen vastaus oli oikeanlainen, mutta nimittäjä oli laskettu väärin (82 %). Trigonometrian yhtälötehtävissä tuli suhteessa kaikkiin tehtäväsarjoihin eniten epäonnistumisia. Kyselytutkimuksen mukaan opiskelijat pitivät trigonometrian tehtäviä vaikeina, ja monet mainitsivat syyksi laskinten käytön aiemmissa opinnoissa.

Opiskelijoiden onnistumista tehtävissä voidaan arvioida toisen ohjelmallisen laskennan ja käsinkirjauksen perusteella. Toisen ohjelmallisen laskennan perusteella matemaattisemmin suuntautuneella ryhmällä MAT-1 suurin osa epäonnistumisista oli juuri trigonometristen yhtälöiden tehtävissä. Toisessa, IMA-123 ryhmässä virheitä tuli tasaisemmin. Selittävänä tekijänä on ainakin edeltävistä matematiikan opinnoista kulunut aika.

Ryhmiä verrattiin toisessa ohjelmallisessa lajittelussa toisiinsa. Vaikeammissa tehtävissä löydettiin tilastollisesti merkitsevää eroa ryhmien välillä, mutta samalla huomattiin, että vaikeimpia tehtäviä olivat tehneet pääasiassa matemaattisesti suuntautuneet opiskelijat. Sinänsä tasoero perustaidoissa jo opiskelutaustan ja opintoihin hakeutumisen perusteella on selvä, vaikka poikkeuksiakin yleensä on.

Käsinkirjauksen perusteella suurin osa opiskelijoista osasi korjata vastaustaan melko hyvin ohjelman palautteen perusteella. Tehtyjen yritteiden määrässä oli tehtäväkohtaista vaihtelua, mutta virheellisiä yritteitä tuli tyypillisesti muutamia kullekin opiskelijalle. Käsinkirjauksessa tehtävä merkittiin keskeytyneeksi kymmenen yrityskerran jälkeen, mutta useimmat saivat tehtävät kuitenkin lopulta oikein. Kertausohjelmana käytetyn MathCheck-kaavatarkastimen hyvä puoli onkin, että se antaa heti palautetta tehtävistä, jolloin opiskelijat voivat yrittää tehtävää välittömästi uudelleen. Ohjelman antamien ohjeiden ja vihjeiden huomaaminen ja tulkitseminen tuotti luultavasti aluksi vaikeuksia, koska sähköinen matematiikan tehtävien kirjoittaminen oli monille uutta.

Ohjelmallisen lajittelun todettiin olevan riittävän tarkka etsittyjen samaa tarkoittavien vastausten lajitteluun. Nyt ongelmana oli erilaisten ja sekalaisten vastausten suuri määrä. Jos tehtävissä olisi ollut esimerkiksi rajoitettu määrä vastausmahdollisuuksia, lajittelu olisi ollut huomattavasti tarkempi. Tämän työn tarpeisiin ohjelmallisella lajittelulla saatiin kuitenkin riittävän tarkat tulokset. Ohjelmallinen lajittelu löysi tehtävien tyypillisimmät virheet, ja listasi ne helposti luettavaan muotoon niiden lukumäärän mukaan. Käsinkirjauksessa saatiin käsitys tyypillisistä virheistä, joita ohjelma ei havainnut.

Jos samankaltaista ohjelmaa käytettäisiin vastaavan kertauksen pitämiseen, hyvä vaihtoehto olisi antaa opiskelijalle muutama korjausmahdollisuus jonka jälkeen tekijä voisi lähettää vastauksen tarkastettavaksi. Vaihtoehtoisesti ohjelman palauteominaisuuden voi kytkeä näkymättömiin. Opiskelija saisi korjata vastaustaan niin monta kertaa kuin haluaa, mutta hänellä olisi ainoastaan yksi mahdollisuus palauttaa se. Tällöin MathCheck taustalla joko hyväksyisi tai hylkäisi vastauksen ja tehtävä voitaisiin arvostella automaattisesti.

Ohjelman alkuperäisessä käyttötarkoituksessa matematiikan harjoitusohjelmana ohjelma toimii sellaisenaan hyvin, kunhan käyttäjä on perehtynyt ohjelman käyttämään syntaksiin. Opiskelijat pitivät vaikeaa syntaksia ohjelman huonoimpana puolena, ja ohjelman käyttöön pitäisi ainakin vähemmän ohjelmistoja käyttäneitä opiskelijoita perehdyttää paremmin. Jonkinlainen pakollinen syntaksin harjoittelutehtävä esimerkiksi aina uuden tehtävätyypin alussa voisi olla hyvä ratkaisu. Vaikeaksi koetusta syntaksista huolimatta useimmat kuitenkin saivat tehtävät tehtyä, vaikka joillekin kertyi kymmeniä yrityksiä. Toisaalta voisi ajatella, että opiskelijoiden pitäisi nähdä vähän vaivaa ainakin ohjelmien peruskäytön opettelemiseksi, koska useissa matematiikkaohjelmissa syntaksi on samankaltainen.

Edellinen kappale osaltaan vastasi kysymykseen, miten opiskelijat onnistuivat sähköisessä kertauksessa. Vaikka jotkut tarvitsivat useita yrityksiä onnistuakseen, lähes kaikki onnistuivat lopulta. Silloin ohjelma toimi niin kuin pitikin.

Myös MathCheck-kaavatarkastimen uusien ominaisuuksien toiminnasta oltiin kiinnostuneita. Kyselytutkimuksen perusteella opiskelijat pitivät ohjelman ominaisuuksia hyvinä, erityisesti mahdollisuutta tehdä tehtäviä valitsemassaan paikassa ja silloin kun itselle sopii. Myös ohjelman ohjaavia ominaisuuksia pidettiin hyvinä, vaikka osa opiskelijoista ei kokenut ohjelman antamaa palautetta riittäväksi. Joka tapauksessa MathCheck-kaavatarkastimen ominaisuuksien hyödyntämiseen ja kehittämiseen kannattaa kiinnittää huomiota, jotta niitä voitaisiin hyödyntää nykyistä laajemmin myös yliopistomaailman ulkopuolella.

Kehitystehtävänä tulevaisuudessa voisi olla sekä automaattisen arvioinnin kehittäminen että MathCheck-kaavantarkastimen hyödyntämisessä laajemmin. Tehtäviä voisi rakentaa esimerkiksi niin, että jos kertaustarvetta on enemmän, saman kokonaisen tehtävän voisi toteuttaa pienemmissä osissa tehtäväsarjana, jonka jälkeen palataan alkuperäiseen tehtävään. Myös tehtävien ohjeistusta voisi lisätä jokaisen tehtävätyypin yhteyteen. Harkita voi myös pitäisikö tehtäväsarjassa päästä eteenpäin ennen kuin ohjelma on hyväksynyt ensimmäiset tehtävät.

Työssä löydettiin vastaukset tutkimuskysymyksiin. Ohjelmallisella lajittelulla löydettiin toistuvat virheelliset ratkaisut tehtävätyypeittäin. Opiskelijoiden todettiin myös onnistuvan sähköisesti toteutetuissa matematiikan harjoitustehtävissä pääosin hyvin, vaikka ohjelman ohjeistuksessa todettiin pieniä puutteita. Kertausohjelman ominaisuuksiin suhtauduttiin kuitenkin positiivisesti ja se kannustaa kehittämään kertausohjelmaa myös tulevaisuudessa.

LÄHDELUETTELO

[1] L. R. Aiken Jr. Language factors in learning mathematics. *Review of Educational Research* 42.3 (1972), 359–385.

[2] Antti Valmari. *MathCheck-kaavatarkastimen tehtävien kirjoitustyökalut*. Saatavissa `http://math.tut.fi/mathcheck/cgi-bin/make_problem.out`. Verkkosivu; luettu 17.toukokuuta 2019.

[3] Antti Valmari. *MathCheck-kaavatarkastimen tehtävien laadintaikkuna*. Saatavissa `http://www.math.tut.fi/mathcheck/MathCheck2017_4.html`. Verkkosivu; luettu 17.toukokuuta 2019.

[4] Antti Valmari. *MathCheck-kaavatarkastin*. Saatavissa `http://math.tut.fi/mathcheck/`. Verkkosivu; luettu 26 syyskuuta 2018.

[5] M. Barry ja N. Steele. A core curriculum in mathematics for the European engineer: an overview. *International Journal of Mathematical Education in Science and Technology* 24.2 (1993), 223–229.

[6] L. Blåfield. Matematiikan verkko-opetus osana perusopetuksen kehittämistä Teknillisessä korkeakoulussa. *Pro gradu-tutkielma. Helsingin yliopisto* (2009).

[7] B. S. Bloom. *Taxonomy of educational objectives: The classification of educational goals*. Longman Group, 1956.

[8] G. Brousseau, R. B. Davis ja T. Werner. Observing students at work. Teoksessa: *Perspectives on mathematics education*. Springer, 1986, 205–241.

[9] B. R. Bryant ja D. P. Rivera. Educational assessment of mathematics skills and abilities. *Journal of Learning Disabilities* 30.1 (1997), 57–68.

[10] G. T. Buswell ja C. H. Judd. *Summary of educational investigations relating to arithmetic*. 27. University of Chicago, 1925.

[11] L. S. Cox. Diagnosing and Remediating Systematic Errors in Addition and Subtraction Computations. *Arithmetic Teacher* 22.2 (1975), 151–157.

[12] Educational Technology Laboratory-EdTec Lab. *Math-Bridge - European mathematics online bridging courses*. Saatavissa `https://edtec.dfki.de/en/projekt/mathbridge-brueckenkurse-zur-mathematik-fuer-studienanfuenger/`. Verkkosivu; luettu 26 syyskuuta 2018.

[13] C. H. Edwards ja D. E. Penney. *Calculus: early transcendentals: matrix version*. Prentice Hall, 2002.

[14] S. Erlwanger. Case studies of children's conceptions of mathematics (Doctoral dissertation). *University of Illinois at Urbana-Champaign* (1974).

[15] European Society for Engineering Education. *SEFI-MWG*. Saatavissa `http://sefi.htw-aalen.de/`. Verkkojulkaisu; luettu 04.02.2019.

[16] J. E. Friend. Column Addition Skills. *Journal of Mathematical Behavior* 2 (1979), 24–58.

[17] M. Gill ja M. Greenhow. How effective is feedback in Computer-Aided Assessments? *Learning, Media and Technology* 33.3 (2008), 207–220.

[18] H. Ginsburg. *Children's arithmetic: The learning process.* D. van Nostrand, 1977.

[19] G. B. Greer ja G. Mulhern. *New directions in mathematics education*. Other, 1989.

[20] S. M. W. Group et al. Mathematics for the european engineer. A curriculum for the twenty-first century. *online], http://sefi. htw-aalen. de/Curriculum/sefimarch2002. pdf* (2002).

[21] L. Haapasalo. Konstruktivistisen pedagogiikan problematiikasta. Teoksessa P. Räsänen, P. Kupari, T. Ahonen & P. Malinen (Toim.) *Matematiikka–näkökulmia opettamiseen ja oppimiseen* (1997), 51–79.

[22] L. Haapasalo. *Oppiminen, tieto ja ongelmanratkaisu*. Medusa-Software, Joensuu, 2011.

[23] M. Hassenzahl ja N. Tractinsky. User experience-a research agenda. *Behaviour & information technology* 25.2 (2006), 91–97.

[24] J. Hiebert ja P. Lefevre. Conceptual and procedural knowledge in mathematics: An introductory analysis. *Conceptual and procedural knowledge: The case of mathematics* 2 (1986), 1–27.

[25] S. Huhtala. *Lähihoitajaopiskelijan oma matematiikka*. Helsingin yliopiston opettajankoulutuslaitos, 2000.

[26] T. Hytönen. Algebran peruslause lukiolaisille. *Matematiikkalehti Solmu* 3 (2011). Verkkosivu, verkkojulkaisu; luettu 18.3.2019.

[27] *Ergonomic Requirements for Office Work with Visual Display Terminels. Part 11 (LEFM) approach*. Standard. Geneva, CH: International Organization for Standardization, maaliskuu 2000.

[28] J. Joutsenlahti. *Lukiolaisen tehtäväorientoituneen matemaattisen ajattelun piirteitä-1990-luvun pitkän matematiikan opiskelijoiden matemaattisen osaamisen ja uskomusten ilmentämänä*. Tampere University Press, 2005.

[29] Jyväskylän ammattikorkeakoulu. *Oppimiskäsitykset*. Verkkojulkaisu, (Viitattu 21.10.2018) https://oppimateriaalit.jamk.fi/oppimiskasitykset/.

[30] Jyväskylän yliopisto. *TIM - The Interactive Material*. Verkkojulkaisu, (Viitattu 26.9.2018) https://tim.jyu.fi/.

[31] Jyväskylän yliopisto. *TIM-rekisteriseloste*. verkkojulkaisu, viitattu 26.9.2018 https://tim.jyu.fi/view/tim/Rekisteriselosterekisterin-suojauksen-periaatteet.

[32] T. Kaarakka ja H. Orelma. *Matriisilaskentaa Insinöörien tarpeisiin, luentomoniste*. 2013.

[33] R. B. Kane, M. A. Byrne ja M. A. Hater. *Helping children read mathematics*. American Book Co., 1974.

[34] N. Karppinen et al. Matematiikan oppimisen tukeminen: Case: Kainuun ammattiopisto/Matkailu-ja ravitsemisala. *Kehittämishankeraportti* (2006).

[35] J. Kauhanen. *Insinöörimatematiikka 1&3, luentomoniste*. 2016.

[36] J. Kela. Ohjelmistot sähköisessä tenttimisessä. Diplomityö. Tampereen teknillinen yliopisto, 2017.

[37] D. Kember. A reconceptualisation of the research into university academics' conceptions of teaching. *Learning and instruction* 7.3 (1997), 255–275.

[38] J. Kilpatrick ja J. Swafford. *J. and Findell, B.(2001) Adding it up: Helping children learn Mathematics*. Washington, DC: National Academy Press.

[39] P. Kupari ja H. Korhonen. Miten matematiikkaa arvioidaan OECD/PISA-ohjelmassa. *Dimensio* 5.2000 (2000), 10–13.

[40] K. Kurki-Suonio ja R. Kurki-Suonio. *Fysiikan merkitykset ja rakenteet*. Limes, 1994.

[41] J. Leino. Konstruktivismi matematiikan opetuksessa. *Teoksessa P. Räsänen, P. Kupari, T. Ahonen & P. Malinen (toim.) Matematiikka–näkökulmia opettamiseen ja oppimiseen. Jyväskylä: Niilo Mäki Instituutti* (2004), 20–31.

[42] MatWorks. *MATLAB*. Saatavissa `https://se.mathworks.com/products/matlab.html`. Verkkosivu, verkkojulkaisu; luettu 26.9.2018.

[43] G. Mulhern. Between the ears: making inferences about internal processes. *New Directions in Mathematics Education.. Londres: Routledge* (1989).

[44] L. Mustoe, B. Olsson-Lehtonen, C. Robinson ja D. Velichova. A Framework for Mathematics Curricula in Engineering Education (2013).

[45] T. Myllykoski, S. S. Ali-Löytty ja S. Pohjolainen. Opiskelijoiden oppimistyökalujen käyttö tietokoneavusteisessa Matematiikkajumppa-tukiopetuksessa. *FMSERA Journal* (2017), 54–65.

[46] A. Nevgi ja S. Lindblom-Ylänne. Johdanto yliopistopedagogiikkaan. *Teoksessa Lindblom-Ylänne, S. & A. Nevgi (toim.): Yliopisto-ja korkeakouluopettajan käsikirja* (2002), 14–28.

[47] J. Nielsen. *Usability engineering*. Elsevier, 1994.

[48] J. Nielsen. *Designing web usability: The practice of simplicity*. New Riders Publishing, 1999.

[49] J. Nielsen. *Why you only need to test with 5 users*. Useit. com Alertbox, 2000.

[50] M. Nieminen. *Ilmavoimien kadetit verkossa: kokemuksia verkkopohjaisen oppimisympäristön käytöstä matematiikan perusopetuksessa*. University of Jyväskylä, 2008.

[51] M. Niss. Mathematical competencies and the learning of mathematics: The Danish KOM project. Teoksessa: *3rd Mediterranean conference on mathematical education*. 2003, 115–124.

[52] M. Niss ja T. Højgaard. Competencies and mathematical learning. *Ideas and inspiration for the development of mathematics teaching and learning in Denmark* (2011).

[53] P. Noras et al. *Konstruktivismin alalajien ilmenemisestä lukion fysiikan opetussuunnitelman perusteissa*. Pro gradu -tutkielma, Helsingin yliopisto, 2017.

[54] Opetushallitus. *Lukion opetussuunnitelman perusteet 2015*. Saatavissa `https://www.oph.fi/download/172124_lukion_opetussuunnitelman_perusteet_2015.pdf`. Verkkojulkaisu; luettu 21.10.2018.

[55] Opetushallitus. *Perusopetuksen opetussuunnitelman perusteet 2014*. Saatavissa `https://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf`. Verkkojulkaisu; luettu 21.10.2018.

[56] Oulun yliopisto W5W^2-hanke 2009. *Näin asennat osaamistavoitteet opetussuunnitelmaasi*. Saatavissa `https://www.uef.fi/documents/677821/698710/Näin+asennat+osaamistavoitteet+opetussuunnitelmaasi-+laaja+oppimäär"a_OY.pdf/260788ff-6aee-4d52-9aee-21d33687051f`. Verkkojulkaisu; luettu 13.2.2019.

[57] S. Ovaska, A. Aula, P. Majaranta et al. Käytettävyystutkimuksen menetelmät (2005).

[58] E. Pehkonen. Opetuksen arviointi ja kehittäminen. Luentomoniste (2009).

[59] E. Pehkonen. Mitä on matematiikka ja miten sitä osataan koulussa. *Arkhimedes 2001: 3* (2001).

[60] E. Pehkonen. Matemaattinen ajattelu ja ymmärtäminen. *Teoksessa E. Pehkonen (Toim.) Luokanopettajaopiskelijoiden matematiikkataidoista. Helsinki: Helsingin yliopisto* (2011), 11–28.

[61] G. Pippig. Rechenschwächen in psychologischer Sicht. *Mathematik in der Schule* 13.11 (1975).

[62] G. PIPPIG. Dialektisches Denken und seine Entwicklung aus pädagogisch-psychologischer Sicht. *Deutsche Zeitschrift für Philosophie* 32.2 (1984), 180–188.

[63] S. Pohjolainen, A. Rasila ja K. Kuosa. Matematiikan oppimisen tukeminen teknillisessä yliopistokoulutuksessa. *Teoksessa J. Joutsenlahti, H. Silfverberg &P. Räsänen (toim.)Matematiikan opetus ja oppiminen.Jyväskylä: Niilo Mäki Instituutti* (2018), 450–474.

[64] D. Poole. *Linear algebra: A modern introduction (2nd ed.)* Thomson Brooks/Cole, 2006.

[65] T. Puolimatka. *Opetuksen teoria: konstruktivismista realismiin*. Tammi, 2002.

[66] H. Radatz. Error analysis in mathematics education. *Journal for Research in mathematics Education* (1979), 163–172.

[67] H. Radatz. Students' errors in the mathematical learning process: a survey. *For the learning of Mathematics* 1.1 (1980), 16–20.

[68] P. Räsänen ja T. Ahonen. Arithmetic disabilities with and without reading difficulties: A comparison of arithmetic errors. *Developmental Neuropsychology* 11.3 (1995), 275–295.

[69] A. Rasila. Automaattisesti tarkastettavat tehtävät matematiikan opetuksessa. *Tuovi 5* (2007), 27–32.

[70] A. Rasila, M. Harjula ja K. Zenger. Automatic assessment of mathematics exercises: Experiences and future prospects. Teoksessa: *ReflekTori 2007 Symposium of Engineering Education*. Vol. 1. 2007, 70–80.

[71] A. Rasila, L. Havola, H. Majander ja J. Malinen. Automatic assessment in engineering mathematics: evaluation of the impact. Teoksessa: *ReflekTori 2010 symposium of engineering education*. 2010, 37–45.

[72] A. Rasila, J. Malinen ja H. Tiitu. On automatic assessment and conceptual understanding. *Teaching Mathematics and its Applications: An International Journal of the IMA* 34.3 (2015), 149–159.

[73] M. Rauste-von Wright, J. Von Wright ja T. Soini. *Oppiminen ja koulutus*. WSOY, 2003.

[74] W. Rice. *Moodle 1.9 E-Learning Course Development*. Packt Publishing Ltd, 2008.

[75] P. Ruohotie. Motivaatio, tahto ja oppiminen. Helsinki: Oy Edita Ab. *Räisänen, A* (1994), 22–28.

[76] J. Ruutu. Tietoliikennetekniikan fuksiopintojen edistyminen ja edistäminen-Uusi opiskelija osaksi tiedeyhteisöä. *Diplomityö. Aalto-yliopisto* (2010).

[77] C. Sangwin. *Computer aided assessment of mathematics*. OUP Oxford, 2013.

[78] C. J. Sangwin ja N. Köcher. Automation of mathematics examinations. *Computers & Education* 94 (2016), 215–227.

[79] C. J. Sangwin ja P. Ramsden. Linear syntax for communicating elementary mathematics. *Journal of Symbolic Computation* 42.9 (2007), 920–934.

[80] K. Silius, S. Pohjolainen, T. Miilumäki, J. Kangas ja J. Joutsenlahti. Korkeakoulumatematiikka teekkarin kompastuskivenä? *Korkeajännityksiä-kohti osallisuutta luovaa korkeakoulutusta* (2011).

[81] Simo Kivelä. *Polynomin tekijöihin jako*. Saatavissa `https://matta.hut.fi/matta2/cluvut/cluvut12.pdf`. Verkkojulkaisu; luettu 25.3.2019.

[82] Tampereen teknillinen yliopisto. *Insinöörimatematiikka 123, Plussa, TTY*. Saatavissa `https://plus.cs.tut.fi/ima123/2018-01/`. Verkkosivu; luettu 23 lokakuuta 2018.

[83] Tampereen teknillinen yliopisto. *Matematiikkajumppa*. Saatavissa `http://math.tut.fi/jumppa`. Opetusohjelma, verkkojulkaisu; luettu 26.9.2018.

[84] Tampereen teknillinen yliopisto. *Matematiikkajumppa*. Saatavissa `http://math.tut.fi/mcjumppa`. Opetusohjelma, verkkojulkaisu; luettu 26.9.2018.

[85] Tampereen teknillinen yliopisto. *TUTWiki*. Saatavissa `http://wiki.tut.fi/MatoOpas/3MatematiikanOpiskeluTTY:llä`. Verkkojulkaisu; luettu 26.9.2018.

[86] Tampereen yliopisto. *Menetelmäopetuksen tietovaranto*. Saatavissa `https://www.fsd.uta.fi/menetelmaopetus/mittaaminen/ominaisuudet.html`. Verkkojulkaisu; luettu 20.52019.

[87] S. Tella ja J. Lavonen. *Tutkielma-oppimisen oiva osoitus: opas tutkielman tekoon ja raportointiin*. Helsingin yliopiston opettajankoulutuslaitos, 1994.

[88] T. termipankki. *Taksonomia*. Saatavissa `http://tieteentermipankki.fi/wiki/Biologia:taksonomia`. Verkkosivu; luettu 23 lokakuuta 2018.

[89] H. Tiitu et al. Lähtökohtia matematiikan verkko-oppimisympäristöjen käytettävyyden tutkimiselle harjoitustehtävien virheitä analysoimalla. *Tuovi 11: Interaktiivinen tekniikka koulutuksessa 2013-konferenssin tutkijatapaamisen artikkelit* (2013), 4.

[90] H. Tiitu. Anna Karenina-periaate. Kohti matematiikan oppimisympäristössä toteutettua virheiden luokittelua. Pro gradu -tutkielma. Helsingin yliopisto, 2016.

[91] H. Tiitu. Oikeisiin virheisiin! Virheelliset vastaukset ja niiden tulkinta automaattisesti tarkastetuissa matematiikan harjoitustehtävissä. Diplomityö. Aalto-yliopisto, 2017.

[92] P. Tynjälä. *Oppiminen tiedon rakentamisena: konstruktivistisen oppimiskäsityksen perusteita*. Kirjayhtymä, 1999.

[93] M. Vakali. Children's thinking in arithmetic word problem solving. *The Journal of Experimental Education* 53.2 (1985), 106–113.

[94] A. Valmari ja T. Kaarakka. MathCheck: A tool for checking math solutions in detail. Teoksessa: *44th SEFI Conference, Engineering Education on Top of the World: Industry University Cooperation*. 2016, 12–15.

[95] Webropol konserni. *Webropol kysely- ja raportointityökalu*. Saatavissa `https://webropol.fi/`. Verkkojulkaisu; luettu 20.52019.

[96] E. Weisstein et al. *Wolfram mathworld*. Saatavissa `http://mathworld.wolfram.com/`. Verkkojulkaisu; luettu 22.3.2019. 2007.

[97] B. Wilson. Evaluation of learning in art education. *Handbook on formative and summative evaluation of student learning* (1971), 499–558.

[98] E. Wilson. The plight of taxonomy. *Ecology* 52.5 (1971), 741–741.

[99] I. Yakimanskaya. Individual differences in spatial thinking in schoolchildren. *Voprosy Psychologii* (1976).

[100] Ylioppilastutkintolautakunta. *Digitaalinen ylioppilastutkinto*. Saatavissa `https://www.ylioppilastutkinto.fi/ylioppilastutkinto/digitaalinen-ylioppilas-\tutkinto`. Verkkojulkaisu; luettu 21.10.2018.

[101] R. Yrjönsuuri. *Matematiikka mieluisaksi: psykologinen lähestymistapa opetukseen ja opiskeluun sekä matemaattisen ajattelun osaamisen arvioimiseen Raija Yrjönsuuri*. Oppilo, 2007.

A PERUSTAITOJEN SÄHKÖISET KERTAUSTEHTÄVÄT

Taulukko A.1. *Kertaustehtävien tehtäväsarjojen 1–4 tehtävät*

	Tehtävät
1	**Peruslaskutoimitukset**
1.1	Määritä luvun -13 käänteisluku.
1.2	Määritä luvun 6 vastaluku.
1.3	Laske lausekkeen $\frac{9}{d^2} + 4$ arvo,kun $d = 4$. Anna vastaus kokonaislukuna, kun a ja b ovat kokonaislukuja
1.4	Laske lausekkeen $x^4 + x^3$ arvo, kun $x = 3$
1.5	Laske lausekkeen $\frac{-3}{-4} : \frac{7}{4}$ arvo murtolukuna
2	**Itseisarvo**
2.1	Ratkaise luvun $a - b$ itseisarvo
2.2	Ratkaise luvun $5 - 13$ itseisarvo
2.3	Ratkaise luvun$\sqrt{3} - 6$ itseisarvo
2.4	Ratkaise luvun $6 - ln(e^{11})$
2.5	Ratkaise luvun $sin(e^7) + 6 - 7\sqrt{2}$ itseisarvo
3	**Lausekkeiden sieventäminen**
3.1	Sievennä $\frac{30(54-30)}{36}$
3.2	Sievennä $\cos\frac{\pi}{3} + \cos\frac{2\pi}{3}$
3.3	Sievennä $(x + y)^2 - (x - y)^2 + 2xy$
3.4	Sievennä $\frac{(x+3)(x+6)}{x+3} - x$
3.5	Sievennä $\frac{x^2+6x+5}{x+1} - (x + 2)$. Oletetaan, että $x > 0$
4	**Osamurtokehitelmä**
4.1	Määritä osamurtohajotelman $\frac{1}{x(x+9)} = \frac{A}{x} + \frac{B}{x+9}$ kertoimet A ja B. Kirjoita kehitelmän oikea puoli uudelleen laskemillasi arvoilla. Oletetaan, että $x \neq -9$.
4.2	Määritä osamurtokehitelman $\frac{9}{x(x-8)} = \frac{A}{x} + \frac{B}{x-8}$ kertoimet A ja B. Kirjoita kehitelmän oikea puoli uudelleen laskemillasi arvoilla. Oletetaan, että $x \neq 0$ ja $x \neq 8$.
4.3	Määritä osamurtokehitelmän $\frac{1}{(x-10)(x+10)} = \frac{A}{x-10} + \frac{B}{x+10}$ kertoimet A ja B. Kirjoita kehitelmän oikea puoli uudelleen laskemillasi arvoilla. Oletetaan, että $x \neq 10$ ja $x \neq -10$.
4.4	Määritä osamurtokehitelman $\frac{3x}{(x-3)(x-4)} = \frac{A}{x-3} + \frac{B}{x-4}$ kertoimet A ja B. Kirjoita hajotelman oikea puoli uudelleen laskemillasi arvoilla. Oletetaan, että $x \neq 3$ ja $x \neq 4$.
4.5	Määritä osamurtokehitelmän $\frac{6}{x(x-5)^2} = \frac{A}{x} + \frac{B}{x-5} + \frac{C}{(x-5)^2}$ kertoimet A ja B. Kirjoita kehitelmän oikea puoli uudelleen laskemillasi arvoilla. Oletetaan, että $x \neq 0$ ja $x \neq 5$.

Taulukko A.2. *Kertaustehtävien tehtäväsarjojen 5–9 tehtävät*

	Tehtävät
5	**Ensimmäisen asteen yhtälö**
5.1	Ratkaise x yhtälöstä $x + 4 = 4$.
5.2	Ratkaise x yhtälöstä $x + 5 = 2x$.
5.3	Ratkaise x yhtälöstä $x + 3 = 2x + 4$.
5.4	Ratkaise x yhtälöstä $9x + 7 = 5x + 3$.
5.5	Ratkaise x yhtälöstä $ex + 6 = \pi x + 8$.
6	**Toisen asteen yhtälö**
6.1	Ratkaise toisen asteen yhtälö $x^2 - 9x = 0$.
6.2	Ratkaise toisen asteen yhtälön $2x^2 - 9x + 3 = 0$ diskriminantti.
6.3	Ratkaise $x^2 + 9x + 20 = 0$.
6.4	Ratkaise $-9x2 - 24x - 7 = 0$. Kannattaa käyttää tässä laskinta diskriminantin laskemiseen.
6.5	Ratkaise $8x^2 - 2x - 10 = 0$.
7	**Polynomiepäyhtälöt**
7.1	Ratkaise ne muuttujan x arvot, jotka kuuluvat välille $[0, \pi]$ ja toteuttavat yhtälön $sin(x) = \frac{\sqrt{3}}{2}$. Anna vastaukset radiaaneissa tarkkana arvona.
7.2	Ratkaise kaikki lukua 7 pienemmät kokonaisluvut, jotka toteuttavat yhtälön $x^2 - 5x + 4 < 0$.
7.3	Ratkaise kaikki lukua 3 pienemmät kokonaisluvut, jotka toteuttavat yhtälön $0 < x(x^2 - 49)$. Kannattaa piirtää kuva.
7.4	Ratkaise kaikki lukua 9 pienemmät kokonaisluvut, jotka toteuttavat murtoyhtälön $\frac{x+1}{x-5} < 0$. Nyt murtoyhtälön osoittaja ja nimittäjä pitää tarkastella erikseen.
7.5	Ratkaise kaikki lukua 4 pienemmät kokonaisluvut, jotka toteuttavat epäyhtälön $x^2 - 1 < 3x + 3$.
8	**Epäyhtälöt**
8.1	Etsi kokonaislukuarvot, joilla $ln(x - 5) > 0$, kun $x < 10$.
8.2	Ratkaise kaikki kokonaisluvut, jotka toteuttavat yhtälön $\|x + 3\| < 4$.
8.3	Ratkaise reaaliluvut, jotka toteuttavat yhtälön $x = \sqrt{x} + 2$.
8.4	Ratkaise reaaliluvut, jotka toteuttavat yhtälön $e^{2x} = 6e^x$
8.5	Ratkaise reaaliluvut, jotka toteuttavat yhtälön $e^{2x} = x - 9$.
9	**Toisen asteen yhtälö**
9.1	Ilmoita radiaaneina $x = 50$ astetta.
9.2	Ilmoita asteina $5\ rad$.
9.3	Laske yhtälön $sin(x) = \frac{\sqrt{3}}{2}$ ratkaisut välillä $[0, 2\pi]$. Ilmoita vastaus radiaaneissa. Vastauksessasi $\pi = pi$.
9.4	Laske yhtälön $cos(x) = \frac{\sqrt{3}}{2}$ ratkaisut välillä $[0, 2\pi]$ Käytä positiivista kiertosuuntaa ja ilmoita vastaus radiaaneissa. Vastauksessasi $pi = \pi$.
9.5	Milloin tangentti $tan(x)$ saa arvon $\sqrt{3}$? Etsi muistikolmioiden avulla kulmat x. Tangentin arvo toistuu $\pi : n$ välein. Olkoon lisäksi $x \in 0, 2\pi$.

Taulukko A.3. *Kertaustehtävien tehtäväsarjojen 10–13 tehtävät*

	Tehtävät
10	**Trigonometriset yhtälöt**
10.1	Laske yhtälön $sin(7x) = \cos(x)$ ratkaisu radiaaneissa? Olkoon $x \in 0, 2\pi$. Etsi taulukosta sinin ja kosinin palautuskaavat ja muunna niin, että yhtälön molemmat puolet ovat muotoa $sin(a) = sin(b)$. Anna pienin positiivinen ratkaisu.
10.2	Laske yhtälön $\cos(8x) + 1 = 0$ ratkaisu radiaaneissa. Anna jälleen pienin positiivinen ratkaisu. Vinkki: Milloin $\cos$ saa arvon -1 ?
10.3	Laske lausekkeen $2sin(\pi 3) + \cos(\frac{\pi}{3})$ arvo. Kulmien tarkat arvot muistikolmioista tai trigonometristen funktioiden taulukosta. Tässä tehtävässä kirjoita vastauksesi suoraan yhtäsuuruusmerkin perään.
10.4	Mikä on kulman x suuruus radiaaneissa, kun $tan(8x) = \sqrt{3}$? Tangentin arvoja löytyy trigonometrian taulukoista. Anna pienin positiivinen ratkaisu.
10.5	Laske yhtälön $sin(6x) = sin(3x + 2)$ ratkaisu. Trigonomiset funktiot ovat jaksollisia, niiden kuvaajat toistuvat samanlaisina jaksottain. Esim. sinille ja kosinille sama kulma toistuu 2π:n välein. Anna vastauksessasi pienin positiivinen ratkaisu radiaaneina.
11	**Eksponentti- ja logaritmifunktio**
11.1	Ratkaise logaritmiyhtälö $ln(x) = 13$. Olkoon $x \neq 0$
11.2	Ratkaise eksponenttiyhtälö $6x = 16$.
11.3	Ratkaise yhtälön $e^3 - 4x = e^{7x}$ arvo.
11.4	Ratkaise logaritmiyhtälö $ln(9x) = ln(x + 3) + 1$.
11.5	Ratkaise yhtälö $e{-}4x = 5$.
12	**Yhtälön ratkaisu**
12.1	Sievennä $\frac{(x+5)(x+7)}{x+6}$. Olkoon $x \neq -6$
12.2	Ratkaise muuttujan x suhteen yhtälö $2^{-5x-3} = 4$
12.3	Ratkaise murtoyhtälö muuttujan x suhteen $\frac{11x}{x-2} = \frac{2}{3} + \frac{1}{x-2}$. Jatka tehtävän alkua ja viimeisen rivin tulee olla muodossa $\Leftrightarrow x = ...$
12.4	Ratkaise $\sin 2(x) + \cos(x) = 1$. Mieti, miten Pythagoraan lause toteutuu yksikköympyrässä $(hypotenuusa = 1)$ niin, että yhtälöön jää vain kosinia. Valitse yhteinen tekijä sopivasti... Olkoon $x \in [0, \pi]$
12.5	Ratkaise yhtälö $4x + 6\cos\frac{2pi}{3} + 2\cos^2\frac{\pi}{4} - 7sin\frac{\pi}{2} = 0$. Vastaa kokonais- tai murtoluvulla.
13	**Alkeisfunktioiden derivointi**
13.1	Laske funktion $f(x) = 4x^{13}$ derivaatta.
13.2	Laske $\frac{d}{dx}7\cos(x)$.
13.3	Laske $\frac{d}{dx}6ln(5x)$.
13.4	Laske funktion $f(x) = -2e^{4x} + 2^x$ derivaatta.
13.5	Laske funktion $f(x) = (2 - 3x)^5$ derivaatta.

Taulukko A.4. *Kertaustehtävien tehtäväsarjojen 14–16 tehtävät*

	Tehtävät
14	**Derivointi**
14.1	Laske funktion $f(x) = 4x^2 - 3x + 7$ derivaatta.
14.2	Laske $\frac{d}{dx}(\frac{4x}{x^2+5})$.
14.3	Laske $\frac{d}{dx}(x^2(x^2+6)$
14.4	Laske $\frac{d}{dx}(3sin(x) + 8x^2)$.
14.5	Laske $\frac{d}{dx}(ln(9x) + 2e^{2x}$.
15	**Alkeisfunktioiden integrointi**
15.1	Integroi $f(x) = x^2 + 3x$.
15.2	Laske $\int(6t-2)dt$.
15.3	Integroi $f(x) = 3x^3 - 2x - 3 + \frac{7}{x^2}$. Olkoon $x \neq 0$.
15.4	Integroi funktio $f(x) = 2x^9 + 4 + \cos(x)$.
15.5	Integroi $f(x) = \frac{6}{6x}$. Oletetaan, että $x > 0$.
16	**Integrointi**
16.1	Integroi $f(x) = e^{6x} + 4\cos(3x)$.
16.2	Integroi funktio $f(x) = xe^x$.
16.3	Integroi funktio $f(x) = x^2e^x$.
16.4	Integroi funktio $f(x) = sin(x)\cos(x)$.
16.5	Integroi funktio $f(x) = xe^x + sin(x) + \sqrt{x}$.

B KYSELYTUTKIMUKSEN KYSYMYKSET

Kysely matematiikan perustaitojen kertauksesta matematiikkajumpan(MCjumppa) ja TIM-sivuston avulla. Molempien sivustojen kertaustehtävät toteutettiin MathCheck-ohjelmalla.
Kurssi

◯ MAT-04600 Insinöörimatematiikka 123
◯ MAT-01160 Matematiikka 1
◯ MAT-01010 Johdatus yliopistomatematiikkaan

Perustaitotestin suoritustapa ja harjoitteluun käytetty aika, anna arviosi tunteina.

◯ Läpäisin perustaitotestin kertaamatta.
◯ Olen suorittanut perustaitotestin aiemmin.
◯ En ole suorittanut perustaitotestiä.
◯ Kertasin MCjumppa-sivustolla.
◯ Kertasin TIM-sivustolla.
◯ Kertasin laskutuvassa.

3. Missä opiskelit matematiikkaa viimeksi ennen tätä syksyä?

◯ Lukio, pitkä matematiikka.
◯ Lukio, lyhyt matematiikka.
◯ Ammattikorkeakoulu.
◯ yliopisto.
◯ Muu, mikä?

Kauanko siitä on, kun opiskelit matematiikka viimeksi?

◯ Vuosi tai vähemmän
◯ 1–2 vuotta.
◯ 2–5 vuotta.
◯ 5–10 vuotta.
◯ Yli kymmenen vuotta

Sukupuoli

◯ Nainen. ◯ Mies. ◯ En halua vastata.

Vastauksessasi

0=Täysin eri mieltä 1=Täysin samaa mieltä 2=Osittain samaa mieltä 3 Täysin samaa mieltä

Osio 1	**0**	**1**	**2**	**3**
Oli hyvä, että saatoin tehdä kertaustehtäviä missä vain.	◯	◯	◯	◯
Oli hyvä, että saatoin tehdä kertaustehtäviä milloin vain.	◯	◯	◯	◯
Tein tehtävät täysin itsenäisesti.	◯	◯	◯	◯
En käyttänyt lainkaan apuna matematiikkaohjelmistoja kertaustehtäviä tehdessäni (Matematiikkaohjelmistoja ovat esim. Matlab, WolframAlpha,CAS-laskimet).	◯	◯	◯	◯
MCjumppa ja TIM-sivuston kertaustehtävät oli hyödyllinen tapa kerrata matematiikan perustaitoja.	◯	◯	◯	◯
Tein kertaustehtäviä vain koska perustaitojen testi on pakollinen osa kurssin suoritusta.	◯	◯	◯	◯
MCjumppa oli mielestäni parempi kertaukseen kuin TIM-sivusto.	◯	◯	◯	◯
Ohjelma motivoi minua tekemään tehtäviä.	◯	◯	◯	◯
Halusin oppia lisää kerratessani.	◯	◯	◯	◯
En pystynyt arvioimaan osaamistani kertaustehtävien avulla.	◯	◯	◯	◯
Olisin halunnut kerrata enemmän perinteisesti kynällä ja paperilla esimerkiksi laskutuvassa.	◯	◯	◯	◯

Osio 2	**0**	**1**	**2**	**3**
Ohjelman mahdollisuus kirjoittaa tehtävien väliaiheita näkyviin on hyödyllinen ominaisuus.	◯	◯	◯	◯
Oli hyvä, että ohjelmassa pystyi kirjoittamaan kommentteja. (Kommentit ovat muistiinpanoja tms. välituloksia, joita ohjelma ei tarkista).	◯	◯	◯	◯
En hyödyntänyt välivaiheita tehtävissä lainkaan.	◯	◯	◯	◯
Hyödynsin usein kommentteja tehtävissä.	◯	◯	◯	◯
Ohjelman antamasta palautteesta ei ollut minulle hyötyä.	◯	◯	◯	◯
Sain korjattua vastauksiani ohjelman antaman palautteen perusteella.	◯	◯	◯	◯
En kokenut oppivani mitään uutta kerratessani ohjelman avulla.	◯	◯	◯	◯
Ohjelman antama palaute oli mielestäni riittävää.	◯	◯	◯	◯
Vastausten syöttäminen ohjelmaan oli helppoa.	◯	◯	◯	◯
Sain riittävästi ohjeistusta vastausten syöttämiseksi ohjelmaan.	◯	◯	◯	◯
En käyttänyt moodlessa olleita tukimateriaaleja lainkaan.	◯	◯	◯	◯
Ohjelma motivoi minua tekemään tehtäviä.	◯	◯	◯	◯
En pystynyt arvioimaan osaamistani kertaustehtävien avulla.	◯	◯	◯	◯

Osio 3. sanalliset vastaukset

Mitä hyvää kertausohjelmassa oli?

Mitä huonoa kertausohjelmassa oli?.

Kumpiin tehtäviin oli helpompi vastata, MathCheck- vai Stack -tehtäviin ja millä tavalla?

Kuinka monta tehtävää arvioit tehneesi harjoitellessasi?

Painottuiko jokin osa-alue harjoitellessasi? Jos, niin miksi?

Mitä hyötyä koit kertaustehtävistä?

Mitkä kaksi osa-aluetta koit vaikeimmiksi kerratessasi?

Mitä kertaustehtävien toteutuksessa voisi parantaa?

Kerro vielä mitä mieltä olet koko matematiikan perustaitojen kertausprosessista?