TECHNOLOGY ENHANCED MATHEMATICS TEACHING I

Place and time: In MP102 on Thursday, Jan 4, at 10:30–12:00

Organizers: Simo Ali-Löytty (Tampere University of Technology)
Terhi Kaarakka (Tampere University of Technology)
Martti Pesonen (University of Eastern Finland)
Antti Rasila (Aalto University)

Contact email: `simo.ali-loytty@tut.fi`

Matemaattisten käsitteiden dynaaminen luonne — kokemuksia oppimateriaalista Moodle-Maxima-Stack-JSXgraph-systeemissä

Martti Pesonen *(University of Eastern Finland)*, `martti.pesonen@uef.fi`
Henri Tanskanen *(University of Eastern Finland)*, `henri.tanskanen@uef.fi`

Abstract. Matematiikan peruskäsitteet kuten relaatiot, funktiot, laskutoimitukset ja lineaariavaruuden kanta ovat lukiosta tuleville opiskelijoille hyvin abstraktia ja uutta kokemusmaailmaa. Abstraktien olioiden ”päässälasku”, siis osata kuvitella operaatioita käsin koskematta ja pystymättä tekemään niillä mitään konkreettista, on yleinen – ja monille ylipääsemätön – este oppia sujuvaa matemaattista ajattelua ja toimintaa.

Olemme koettaneet kehittää oppimateriaaleja ja oppimisen testaustapoja, joiden tarkoitus on avittaa ja tukea abstraktioprosessia ruokkimalla oppijoiden mielikuvitusta tilanteilla, joissa on tehtävä arvauksia ja otaksumia siitä mistä onkaan kysymys. Matemaattisten aineiden opiskelussa tärkeitä näkökulmia ovat ilmiöiden verbaaliset ja visuaaliset piirteet yhdistettyinä niiden mallintamiseen symbolimuotoisilla esitystavoilla kuten pukeminen funktioiksi tai muiksi relaatioiksi. Tärkeäksi olemme kokeneet käsitteiden dynaamiset esitystavat, joissa ei ole mitään sellaista symbolimuotoista informaatiota, joka heti herättäisi vaistonvaraisen laskemisrefleksin. Näiden ”mustien laatikoiden” tarkoitus on viekoitella kokeilemaan, tekemään havaintoihin pohjautuvia kuvitelmia; siis ajattelemaan. Muutenkin ne pyrkivät osoittamaan, etteivät esimerkiksi funktiot tarvitse olla eksplisiittisesti joillain kaavoilla määriteltyjä, vaan voivat toimia ennalta arvaamattomasti.

Esittelemme konkreettisten esimerkkien kautta ABACUS-projektin Moodle-ympäristöön niveltämäämme materiaalia, jota on käytetty 1. vuoden yliopistomatematiikan kursseilla Matematiikan johdantokurssi ja Lineaarialgebra. Kerromme myös (liian) monista kohtaamistamme teknisistä ongelmista.

MathCheckin uudet ominaisuudet

Antti Valmari *(Tampere University of Technology)*, `antti.valmari@jyu.fi`

Abstract. MathCheck on uudentyyppinen ohjelma palautteen antamiseksi matematiikan ja logiikan tehtävien vastauksista. Sille voi antaa pitkiä sievennysketjuja tai yhtälöiden ratkaisuja, ja se tarkastaa ne vaihe vaiheelta. Se sisältää myös uudentyyppisiä tehtävälajeja: lausekkeen kirjoittaminen sen lausekepuuta esittävän kuvan perusteella, taulukosta puhuvan väittämän kirjoittaminen predi-

kaattilogiikalla, sekä päätteleminen propositiologiikassa tai jäännösluokkarenkaassa $\mathbb{Z}_m$, missä $2 \leq m \leq 25$. Valtaosa toiminnoista on uusia verrattuna Matematiikan päivien esitelmään vuonna 2016. Olen rakentanut MathCheckiä pätkissä noin kolmen vuoden aikana. Sitä voi kokeilla osoitteessa http://math.tut.fi/mathcheck/.

Sievennysvirheen kohdalla MathCheck antaa vastaesimerkin ja usein myös piirtää virheellisen relaation molemmin puolin olevien lausekkeiden edustamien funktioiden kuvaajat. Opiskelija saa näin palautetta, joka näyttää virheen paikan ja auttaa virheen tunnistamisessa. Hän voi sen ansiosta muodostaa pitkänkin sievennysketjun vaihe vaiheelta tarvitsematta apua opettajalta. Tämä on merkittävä etu verrattuna ohjelmiin, jotka tarkastavat vain lopullisen lausekkeen.

Sievennysketjussa voi opettajan niin halutessa esiintyä relaation $=$ lisäksi relaatioita $\geq$ ja $>$ tai $\leq$ ja $<$. Opettaja voi myös asettaa vaatimuksen, että lopullisen lausekkeen tulee olla tiettyä muotoa (esim. tulojen summa) ja riittävän yksinkertainen. Tällaisten vaatimusten lisäksi MathCheck tarkastaa vain relaatioiden paikkansapitävyyden, joten sievennyksen ei tarvitse edetä opettajan ennalta määräämää reittiä vaan mikä tahansa matemaattisesti pätevä sievennys kelpaa. Näin ollen opiskelija voi käyttää MathCheckiä myös itsenäiseen harjoitteluun ilman valmiita tehtäviä ja mallivastauksia. MathCheck kiinnittää myös jonkin verran huomiota määrittelyalueisiin. Se ei esimerkiksi hyväksy sievennystä $\frac{b}{b} = 1$, ellei sille ole sanottu, että b ei voi olla 0.

Gödelin kuuluisan tuloksen vuoksi ei ole olemassa algoritmia, jolla voitaisiin aina selvittää, pitääkö vain symboleita 0, 1, $+$, $\cdot$, $=$, $\neg$, $\wedge$ ja $\forall$ sekä muuttujia käyttämällä muodostettu luonnollisia lukuja koskeva väittämä paikkansa. Reaaliluvuille tällainen algoritmi on, mutta ei ole jos lisäksi sallitaan sin, ei vaikka samalla rajoitetaan väittämä muotoon $\exists x : f(x) = 0$ tai muotoon $\exists x : f(x) > 0$. Siksi mikään sievennysvirheiden paljastamiseen tarkoitettu ohjelma ei voi olla täydellinen. MathCheckin osalta tavoitteena on, että se ei koskaan hylkää oikeaa ratkaisua, mutta saa toisinaan jättää virheitä huomaamatta.

Sievennysten tarkastus perustuu kokeilemiseen joukolla testiarvoja sekä relaatioiden $<$, $\leq$, $\geq$ ja $>$ tapauksessa myös iteroimiseen kohti virhettä. Koulumatematiikan funktioiden jatkuvuus- ja derivoituvuusominaisuuksista seuraa, että käytännössä virheet löytyvät melkein aina kahta poikkeusta lukuunottamatta. Nollalla jakamisen pois sieventämisiä (esim. $\frac{x-7}{x-7} = 1$ tai $1 = \frac{x-7}{x-7}$) MathCheck huomaa vain rajatusti, ja lattia- tai kattofunktiota käytettäessä virheiden löytämiskyky heikkenee. MathCheck osaa myös hieman symbolista laskentaa, joten toisinaan se ilmaisee todistaneensa relaation pätevyyden.

Koska tietokoneet eivät laske reaaliluvuilla tarkasti, MathCheck käyttää tarkkaa rationaalilukuaritmetiikkaa silloin kun pystyy ja muulloin se käyttää ala- ja ylälikiarvon muodostamaa väliä. Lisäksi mukana on tieto, voiko arvo olla määrittelemätön, kuten $\frac{1}{0}$ tai $\sqrt{-1}$. MathCheck raportoi virheen vain jos ala- ja ylälikiarvon perusteella on varmaa, että kyse on virheestä. Toisinaan MathCheck antaa varoituksen mutta jatkaa sievennysketjun tarkastusta eteenpäin, esimerkiksi kun on vahva syy uskoa virheellisyyteen vaikka täyttä varmuutta ei olekaan.

Yhtälön ratkaisun MathCheck tarkastaa kokeilemalla, että jokainen opettajan antama juuri toteuttaa kunkin välivaiheen ja että opiskelijan löytämät juuret toteuttavat alkuperäisen yhtälön. Tällä toimintaperiaatteella on se suuri etu, että se soveltuu mihin tahansa yhtälöihin. Se ei siis ole rajoitettu esimerkiksi ensimmäisen ja toisen asteen yhtälöihin.

Haittana on, että jos virhe aiheuttaa ylimääräisiä juuria mutta ei hävitä oikeita juuria, MathCheck huomaa sen vasta kun ylimääräinen juuri näkyy välivaiheesta eksplisiittisesti. Kun näin käy, MathCheck kertoo virheen tarkan paikan etsittyään sen kokeilemalla ylimääräistä juurta kuhunkin välivaiheeseen ensimmäisestä alkaen. Haittana on myös, että opiskelija ei voi käyttää yhtälönratkaisumoodia itse keksimiensä tehtävien ratkaisemiseen. Siksi kehitysajatuksiin kuuluu lisätä MathCheckiin kyky etsiä itsenäisesti juurille likiarvot numeerisesti. Haaveisiin kuuluu myös mahdollistaa epäyhtälöiden ratkaiseminen.

Lausekepuutehtävillä voi harjoitella lausekkeen rakenteen hahmottamista ja sitovuussääntöjä. Ne auttavat oivaltamaan syntaksin ja semantiikan eroa. Esimerkiksi $(x + y) + 1$ on eri lauseke kuin $x + (y + 1)$, vaikka ne tuottavat aina saman arvon. Lausekkeiden samuus määräytyy syntaksin tasolla, joten se ei riipu käytettyjen operaattoreiden merkityksestä, toisin kuin $(x \circ y) \circ 1 = x \circ (y \circ 1)$, joka pätee jos $\circ$ on $+$ mutta ei päde jos $\circ$ on $-$. MathCheckin tapa piirtää lausekepuut tekee näkyväksi, että $=$, $\geq$, $\Leftrightarrow$ jne. eivät sido vasemmalle (kuten esim. $+$) eivätkä oikealle (kuten esim. potenssi), vaan ovat konjunktionaalisia. Lausekepuutehtäville on olemassa hyvin tehokas algoritmi, joten MathCheck tarkastaa vastauksen kattavasti.

Taulukosta puhuva väittämä voi olla esimerkiksi $\forall i; 1 \leq i \leq n : A[i] = A[n - i]$. Jos se annetaan vastaukseksi tehtävään, jossa pyydetään sanomaan kaavana, että taulukon $A[1 \dots n]$ sisältö on palindromi, niin MathCheck sanoo, että kun $A = [0]$ ja $n = 1$, niin mallivastaus tuottaa toden mutta opiskelijan vastaus määrittelemättömän. MathCheck tarkastaa vastauksen kokeilemalla kaikilla taulukoilla, joiden koko on enintään 4 ja alkioina on kokonaislukuja 0, ..., 3.

Tämä tehtävälaji harjoituttaa käsitteen muodostamisen, erittelyn ja formuloinnin taitoa. Päättelytaitoakin voi harjoituttaa esimerkiksi käskemällä opiskelijaa antamaan lyhyen väittämän, joka tarkoittaa samaa kuin $\forall i; 1 \leq i \leq n : \exists j; 1 \leq j \leq n : A[i] < A[j]$. Tässä tehtävälajissa tulee toisinaan yllätyksiä, kun MathCheck paljastaa, että "itsestään selvästi" oikea vastaus ei olekaan oikein jossakin erikoistilanteessa. Esimerkiksi äskeinen väittämä ei ole identtisesti epätosi.

Propositiologiikka- ja $\mathbb{Z}_m$-tehtävien ratkaisut MathCheck tarkastaa täydellisesti kokeilemalla kaikki vaihtoehdot. Siksi opettajalla on suuri vapaus tehtävien laadinnassa ja opiskelija voi käyttää näitä toimintamuotoja ilman ennalta annettua tehtävää. Haittana on tietenkin, että propositiologiikasta ei ole helppo keksiä kovin monenlaisia mielekkäitä tehtäviä, ja modulaarinen aritmetiikka on syrjässä matematiikan opintojaksojen ydinalueelta.

Jonkin verran lisää mielenkiintoa tuo se, että MathCheck hallitsee myös kolmiarvoisen propositiologiikan, jossa on mukana totuusarvo "määrittelemätön". Eksakti kaksiarvoinen logiikka on täysin riittämätön jo tavallisessa yhtälöiden ratkaisemisessa, mutta miksi näin on ja miten kolmiarvoisuus pitää toteuttaa on niin monimutkainen asia, että se ansaitsee oman esitelmänsä. Tavallisesti yhtälöiden ratkaisuissa käytetään epäeksaktia logiikkaa, joka toimii ihmisillä, mutta sitä ei voi toteuttaa tietokoneohjelmana.

Mielikuvitusta käyttämällä MathCheckiä voi soveltaa monipuolisemmin kuin tehtävälajien kuvauksen perusteella ensin näyttää. Esimerkiksi tehtävä voi olla "kuinka monta vertailua n:n funktiona tämä tietokoneohjelma tekee", ja MathCheck tarkastaa, että vastaus esittää samaa funktiota kuin $\frac{1}{2}n(n+1)$. Voidaan siis teettää harjoituksia, joissa opiskelijan tehtävänä ei ole lausekkeen sieventäminen vaan muodostaminen. Suoraa tukea induktiotodistusten harjoittelulle MathCheckissä ei (vielä?) ole, mutta lausekepuutehtävien avulla opiskelijaa voidaan harjoituttaa kirjoittamaan pohjatapaus ja induktioaskeleen tavoite oikein.

Kaikki muu MathCheckissä on itse tehtyä paitsi matematiikan ladonta. Siihen käytetään MathJaxia, joka on de facto -standardi. Valitettavasti MathJax on nykyisin harmillisen hidas. Itse tekeminen on usein pienemmän riesan tie kuin yrittää saada muualta hankittuja ohjelmia tekemään asioita joihin niitä ei ole suunniteltu, kuten tuottamaan vastaesimerkki silloin kun sievennysaskel ei ole oikein. Itse tekemällä saa ominaisuuksista sellaisia kuin haluaa, ja uusia ominaisuuksia voi lisätä kun tarvetta ilmenee, kuten komennon, jolla voi vaatia, että lopullinen vastaus on muodoltaan summien tulo. Sen avulla voi teettää tekijöihinjakotehtäviä. Itse tekemällä välttää myös lisenssi- ja teknisiä yhteensopivuusongelmia.

Motivointia Matlabin opiskeluun matematiikkaa unohtamatta

TERHI KAARAKKA *(Tampere University of Technology)*, `terhi.kaarakka@tut.fi`

Abstract. Usein matemaattisten ohjelmistojen opetus sisältyy matematiikan opintojaksojen yhteyteen, jonne se myös luonnollisesti sopiikin hyvin. Tampereen teknillisellä yliopistolla Matlabin perusteet opetetaan insinöörimatematiikan opintojaksojen tietoja opettaessa. Usein kuitenkin opiskelijoiden motivointi Matlabin opiskeluun ja käyttöön on hankalaa. Yhtenä syynä tähän on, että tenteissä ei perinteisesti käytetä mitään apuvälineitä. Tämä käytäntö on hyvä siinä mielessä, että sallittaessa opiskelijoille laskimien ja muiden apuvälineiden käyttö, heiltä jää moni perusasia opettelematta ja syvällisesti ymmärtämättä, jolloin matematiikan oppiminen jää pinnalliselle tasolle.

Ammattiaineiden opinnoissa toivotaan, että Matlabin perustoiminnot ovat opiskelijoille tuttuja ja sen takia on tärkeää löytää riittävät motivointikeinot. Lisäksi linjakkaan opetuksen perusideologiaan kuuluu niiden asioiden, joita opintojaksolla opiskellaan, arviointi myös arvosanaa annettaessa.

Edellä kuvattu tilanne on melko ristiriitainen. Olemme pohtineet TTY:llä edellä esiteltyä ongelmaa ja etsineet ratkaisuja kysymyksiin:

- Kuinka motivoida opiskelijat käyttämään Matlabia?
- Kuinka vahvistaa matematiikan oppimista ja osaamista?

Syksyllä 2017 kokeilimme löytää ratkaisun ongelmaan seuraavin keinoin

1. Lisäsimme laskuharjoituksissa Matlabin käyttöä:

    – Sähköisesti palautettavissa ja arvioitavissa STACK harjoituksissa oli viikottain osio, jossa harjoiteltiin matlabin käyttöä. Puolet opiskelijoista käytti matlabin oppimiseen suunniteltua minäpystyvyyttä vahvistavaa opetusohjelmaa ja puolet harjoitteli opetusvideoita ja tekstipohjaisia ohjeita käyttäen.

    – Kolmella kuudesta harjoituskerrasta yksi tehtävä tuli tehdä matlabin live-editorilla ja palauttaa pdf-muodossa oppimisalustalle.

- Pyrimme tekemään kynnyksen Matlabin ja etenkin Live-editorin käyttöön matalaksi tarjoamalla myös videomateriaalia itseopiskelun tueksi sekä kokoamalla lopuksi viikottain eteentulleet komennot yhteen lyhyeen ja selkeään dokumenttiin.

2. Luennoilla läpikäytiin enemmän luentoesimerkkejä Matlabilla.
3. Otimme Matlabin käyttöön osalla tehtäviä myös tentissä.

Motivoitumista (laskuharjoituspisteiden lisäksi) vahvistimme siis sillä, että tentti jaettiin kahteen osaan. Perinteisessä paperitentissä tehtiin puolet tehtävistä ja puolet tehtiin Matlabin live-editorilla EXAM-tenttiluokassa.

Vaikka opiskelijoilla oli tentissä käytössä Matlab, niin pyrimme matemaattisesti pysymään vähintäänkin yhtä korkealla tasolla kuin aiemminkin. Valitsimme Matlab tehtäviin ominaisarvot ja matriisien keskeisimmät aliavaruudet sekä testasimme matriisin kääntyvyyteen liittyvää teoriaa seuraavan esimerkin kaltaisia monivalintatehtäviä hyödyntäen.

Tarkastellaan 4×4-matriisia B, jolle $N(B) = \operatorname{span}\left(\begin{bmatrix}1\\1\\0\\0\end{bmatrix}\right)$.

* Onko matriisi B kääntyvä?
Vastausvaihtoehdot (a) kyllä, (b) ei, (c) lähtötietojen perusteella ei voi tietää

* Mitä on sarakeavaruuden $R(B)$ dimensio?
Vastausvaihtoehdot (a) 1, (b) 2, (c) 3, (d) 4

* Mitä on $\det(B)$?
Vastausvaihtoehdot (a) 1, (b) 0, (c) -1, (d) lähtötietojen perusteella ei voi tietää

* Onko 0 matriisin B ominaisarvo?
Vastausvaihtoehdot (a) kyllä, (b) ei, (c) lähtötietojen perusteella ei voi tietää

* Onko olemassa vektoria $\mathbf{b} \in \mathbb{R}^4$, joka ei kuulu matriisin B sarakeavaruuteen?
Vastausvaihtoehdot (a) kyllä, (b) ei, (c) lähtötietojen perusteella ei voi tietää

* Ovatko matriisin B sarakkeet lineaarisesti riippumattomia?
Vastausvaihtoehdot (a) kyllä, (b) ei, (c) lähtötietojen perusteella ei voi tietää

Tehtävän oikea ratkaisu vaatii hyvän onnen lisäksi (väärästä ratkaisusta sakotettiin 1/2 pistettä) melko hyvää ymmärrystä lineaarialgebran perusteista. Tehtävässä pystyi kokeilemaan asioita matlabilla, mutta sinänsä matlabista ei tässä tehtävässä ollut paljon hyötyä. Toinen tehtävä oli matemaattisesti kevyempi, mutta siinä pystyi Matlabin osaamista hyödyntämään enemmän. Tehtävässä annettiin kaksi matriisia, joista piti selvittää seuraavat asiat

(a) Laske matriisin A aste?

(b) Ratkaise matriisin A sarakeavaruuden kanta.

(c) Ratkaise matriisin A nolla-avaruuden kanta.

(d) Ratkaise matriisin B ominaisarvot.

(e) Ratkaise matriisi V s.e. $B = VDV^{-1}$, missä D on diagonaalimatriisi.

(f) Onko kohdassa (e) ratkaistu matriisi V ortogonaalinen.

Muista perustella jokaisen kohdan vastaukset.

Tulokset tenteistä sekä opiskelijoiden palautteen saamme joulukuussa 2017, minkä jälkeen voimme paremmin kertoa, onnistuiko kokeilu vai jääkö se tähän yhteen kertaan.

Yhteistyökumppaneina S. Ali-Löytty ja J. Kangas.