



TAMPEREEN
AMMATTIKORKEAKOULU

PAIKALLA VALETTUJEN SILTOJEN ALGORITMIPOHJAINEN MALLINTAMINEN

Joni Nieminen

Opinnäytetyö
Huhtikuu 2018
Rakennustekniikka
Talonrakennustekniikka



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Rakennustekniikka
Talonrakennustekniikka

NIEMINEN, JONI:

Paikallavalettujen siltojen algoritmipohjainen mallintaminen

Opinnäytetyö 33 sivua, joista liitteitä 1 sivu
Huhtikuu 2018

Opinnäytetyössä selvitettiin algoritmipohjaisen mallintamisen mahdollisuuksia siltojen suunnittelussa. Opinnäytetyön toimeksiantajana toimi Ramboll Finland Oy. Tavoitteena oli luoda yritykselle algoritmipohjainen työkalu paikallavalettujen siltojen parametriseen mallintamiseen. Opinnäytetyöhön sisältyy luottamuksellista tietoa, joten yksityiskohtainen tieto algoritmin rakenteesta on jätetty esittelemättä tässä työssä.

Opinnäytetyössä onnistuttiin luomaan toimiva algoritmipohja, jolla voidaan mallintaa parametrisesti yleissuunnitelmatasoinen tietomalli teräsbetonisille ulokelaattasilloille. Jatkokehityksen myötä algoritmipohjasta saadaan muokattua käyttäjäystävällinen työkalu siltojen suunnitteluun. Työ toteutettiin käyttämällä Tekla Structures rakennesuunnitteluohjelmistoa, johon tietomalli luotiin käyttäen NURBS pohjaista Rhinoceros 3D -pintamallinnusohjelmaa ja sen visuaaliseen ohjelmointiin tarkoitettua lisäosaa Grasshopperia. Työssä esitellään ohjelmien perusteet, niiden tehtävät tietomallin luomisessa ja käydään läpi luodun algoritmipohjan rakenne ja toiminta.

Siltojen vaikeiden geometrioiden hallitseminen vaatii mallinnustyökaluilta soveltumista vaikeampiin rakenteisiin kuin perinteisessä talonrakentamisessa. Osaltaan tämän vuoksi opinnäytetyössä esitellyt ja käytetyt ohjelmat tulevat olemaan tulevaisuudessa tärkeässä roolissa siltojen suunnittelussa. Tämänhetkisten haasteiden vuoksi on tärkeää, että käytössä olevia ohjelmia kehitetään siltojen mallintamiselle paremmin soveltuviksi.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Construction Engineering
Building Construction

NIEMINEN, JONI:
Algorithm-Aided Modelling of Cast-In-Place Bridges

Bachelor's thesis 33 pages, appendices 1 page
April 2018

This thesis was started because of the need to explore the possibilities of algorithm-aided modelling of bridges. The commissioner of the thesis was Ramboll Finland Oy. The main objective was to create an algorithm-aided parametric modelling tool for cast-in-place bridges for the company. The thesis includes confidential information about an algorithm, which was not included in the public version.

In this study algorithm template for modelling of cast-in-place bridges was successfully created. It was implemented with the structural design software Tekla Structures where the data model was created by using the NURBS based Rhinoceros 3D CAD software and its visual programming add-on called Grasshopper. The basics of these programs, their role in the creation of the data model and basic structure of the aforementioned algorithm template are presented in this thesis.

Modelling programs have to be capable of processing the challenging geometries of bridges. Partly because of this the modelling programs presented here will play an important role in the future. Because of current problems and challenges it would be important to develop modelling programs that are better suited for designing bridges.

Key words: algorithm-aided modelling, grasshopper, parameter, cast-in-place bridge

SISÄLLYS

1	JOHDANTO.....	6
2	TYÖN TAVOITTEET JA RAJAUKSET	7
3	ALGORITMIPOHJAINEN MALLINTAMINEN.....	8
3.1	Rhinoceros 3D	8
3.1.1	Rhinocerosin historia	9
3.2	Grasshopper 3D	9
3.2.1	Grasshopperin historiaa.....	10
3.3	Tekla Structures ja Tekla-live link.....	11
3.4	Ohjelmien käyttö ja tietomallin muodostuminen.....	11
3.4.1	Perinteisen ohjelmoinnin käyttö algoritmipohjassa	12
3.4.2	Sketch editor ja variable cross section	13
4	PAIKALLAVALETUT TERÄSBETONISILLAT	16
4.1	Teräsbetonisen ulokelaattasillan yleissuunnitelma.....	16
4.2	Ulokelaattasillan suunnittelu.....	17
4.2.1	Kansi ja reunapalkki.....	17
4.2.2	Päätypalkki ja siipimuurit	18
4.2.3	Välituet.....	19
4.2.4	Kaiteet	19
5	LÄHTÖTIEDOT JA ALGORITMIN RAKENNE	21
5.1	Sillan tietomallin lähtötiedot.....	21
5.2	Algoritmin rakenne ja toiminta.....	21
5.2.1	Tasausviivan käyttö.....	22
5.2.2	Sillan muodostuminen.....	22
5.2.3	Peruslaattojen kaivannot	23
5.3	Muuttuvat parametrit	24
5.3.1	Tekla poikkileikkaukset	25
5.3.2	Teklan osien attribuutit	26
6	LOPPUTULOKSEN ANALYSOINTI	27
6.1	Algoritmipohjan tekoprosessi	27
6.2	Algoritmipohjan toimivuus.....	27
6.3	Haasteet.....	27
6.4	Jatkokehitys	28
7	POHDINTA.....	30
	LÄHTEET.....	31
	LIITTEET.....	32
	Liite 1	33

ERITYISSANASTO

Rhino	Rhinoceros 3D -pintamallinnusohjelma
Grasshopper	visuaalisen ohjelmoinnin lisäosa Rhinolle
Tekla Structures	Tekla on rakennesuunnitteluohjelmisto
Tietomalli	rakennuksen elinkaaren aikaisten tietojen kokonaisuus digitaaliossa muodossa
Komponentti	Grasshopperin työympäristöön lisättävä ja toiminnon suorittava objekti
Parametri	komponentille välitettävä muutettavissa oleva lähtötieto
Algoritmi	prosessi, joka koostuu järjestyksessä olevista toiminnoista
Algoritmpohja	komponenteista koostuva ja parametrisesti säädeltävä algoritmi Grasshopperissa
Kangas	Grasshopperin työympäristön osa, jolle algoritmi luodaan
Attribuutti	Teklan osalle määritettävä ominaisuus
BIM-ohjelmisto	ohjelmisto, jolla luodaan rakennuksen tietomalli
NURBS	matemaattinen malli, jolla luodaan ja esitetään käyriä pintoja
TSV	väylän tasausviiva

1 JOHDANTO

Tietomallintamisen ja algoritmipohjaisen suunnittelun yleistyminen luo painetta insinööritoimistoille. Varsinkin algoritmipohjaisen suunnittelun osaajia on melko vähän suhteessa suunnittelijoiden määrään. Valmistuvista insinööreistä ei ole lähtökohtaisesti apua yrityksille, koska mallintamisen ja algoritmipohjaisen suunnittelun opetus ammattikorkeakouluissa sekä yliopistoissa rakennustekniikan tutkinnoissa on todella vähäistä. Oppiminen jää liian usein työpaikalle ja oman kiinnostuksen varaan.

Tietomallintamisen kasvu johtuu lähinnä sen tuomista mahdollisuuksista sekä tilaajien vaatimuksista. Mittatarkat kolmiulotteiset tietomallit ja koko ajan kehittyvät FEM-ohjelmat antavat mahdollisuuksia yhä monimutkaisempien rakenteiden suunnitteluun. Tietomallista voidaan helposti tutkia rakenteiden toimivuutta ja lisäksi tarvittavat piirustukset saadaan tulostettua luodun mallin pohjalta.

Yhä useammat tilaajat eri rakentamisen aloilla vaativat tietomallipohjaisia suunnitelmia. Suomessa silta-alalla painetta luo Liikenneviraston käynnistämä digitalisaatiohanke, jonka seurauksena uudet sillat ja taitorakenteet tullaan suunnittelemaan tietomallipohjaisesti.

Suunnittelun muunneltavuus, tarkkuus ja nopeus ovat keskeisiä hyötyjä, joita algoritmipohjaisella suunnittelulla voidaan oikein toteutettuna saavuttaa. Lisäksi algoritmipohjasta voidaan luoda yleinen suunnittelutyökalu. Itse algoritmista suunnittelijan ei tarvitse ymmärtää, kunhan pohja on luotu niin, että muunneltavat parametrit on esitetty selkeästi ja mallin luominen onnistuu lyhyiden ohjeiden avustuksella.

2 TYÖN TAVOITTEET JA RAJAUKSET

Opinnäytetyön tekoon lähdettiin tarpeesta luoda algoritmipohja paikallavalettujen teräsbetonisiltojen yleissuunnitelmatasoiseen suunnitteluun. Työn päätavoitteeksi asetettiin, että algoritmipohjalla pystytään toteuttamaan tulevista kohteista 60 – 70 % ja lisäksi kuka tahansa ohjelmien peruskäytön tunteva voisi sitä helposti käyttää. Tavoitteiden täytyessä teräsbetonisten siltojen suunnitteluprosessi nopeutuisi huomattavasti. Lisäksi algoritmipohja olisi hyvä työkalu suunnittelijoille, jotka eivät omaa juurikaan kokemusta tietomallintamisesta.

Päätavoitteen saavuttamiseksi pohjan tulee olla yksinkertainen käyttää, lähtöparametrien tulee olla selkeästi esillä ja samaa parametria ei voi säätää kuin yhdestä paikasta. Näiden kriteerien täytyessä suunnitelman muunneltavuus suunnittelun edetessä on helppoa ja suunnittelun aikaisten muutosten aiheuttamat ongelmat saadaan minimoitua.

Opinnäytetyö rajautui työtä tehdessä koskemaan teräsbetonisia ulokelaattasiltoja. Opinnäytetyössä luodulla ohjelmalla pystytään luomaan yleissuunnitelmatasoinen tietomalli. Raudoitusten mallintamista ei pystytä tekemään luodun algoritmipohjan avulla. Näiden lisäksi algoritmiin lähdettiin luomaan toimintoa, jolla voidaan mallintaa sillan anturoiden kaivannot ja laskea niiden tilavuudet.

3 ALGORITMIPOHJAINEN MALLINTAMINEN

Rakenteiden suunnittelussa algoritmipohjainen mallintaminen on apuväline tietomallin luomiseksi. Algoritmi itsessään tarkoittaa jonkin prosessin kulkua, joka koostuu sarjasta järjestyksessä olevia toimintoja, ja sillä on alku ja loppu. Mallintamisella tarkoitetaan yleensä kolmiulotteisessa maailmassa käsiteltävää dataa, josta muodostuu viivoja, pinta-aloja ja kiinteitä kolmiulotteisia kappaleita. Mallin kappaleilla voi olla myös tietoja muun muassa sen materiaaliominaisuuksista. Näistä muodostuvaa kokonaisuutta kutsutaan tietomalliksi. Algoritmipohjaisessa mallintamisessa siis luodaan säädeltävistä parametreista koostuva algoritmi, jonka avulla tuotetaan helposti muokattavissa olevaa tietomallia. Algoritmipohjan luomiseksi, opinnäytetyössä on käytetty Rhinoceros 3D -ohjelmaa ja tämän lisäosaa nimeltä Grasshopper.

Algoritmipohjaista mallintamista mahdollistavia ohjelmia on hyvin niukasti markkinoilla. Yksi vaihtoehto olisi Revitin ja Dynamon yhdistelmän käyttö. Revit on Autodeskin kehittämä tietomallinnusohjelmisto ja Dynamo on visuaalisen ohjelmoinnin työkalu. Toinen vaihtoehto olisi Bentley Systemin GenerativeComponents, joka on parametriseen suunnitteluun kehitetty ohjelmisto. (Revit, 2018; Dynamo, 2018; Bentley System, 2018.)

Syy Teklan, Rhinon ja Grasshopperin käyttöön perustuu ohjelmien luoman yhdistelmän parempaan soveltuvuuteen silta-alalle. Esimerkiksi siltojen geometriat ovat usein jopa kaksoiskaarevia, mikä luo haasteita tietomalliohjelmistoille ja näin ollen parempaan lopputulokseen päästään tällä hetkellä kyseisillä ohjelmilla.

3.1 Rhinoceros 3D

Rhinoceros 3D tuttavallisemmin Rhino on helppokäyttöinen 3D NURBS -pintamallinnusohjelma, jonka avulla voidaan luoda hankaliakin kolmiulotteisia muotoja helposti ja tarkasti. Rhino sopii hyvin kaikkeen tarkkaan kolmiulotteiseen mallintamiseen. Mallien monimutkaisuudella tai fyysisellä koolla ei ole merkitystä. (M.A.D. 2018).

Rhinoa käytetään monipuolisesti eri aloilla ja Suomessa rakenteiden suunnittelussa sen käyttö on kasvamaan päin yhdessä Grasshopperin kanssa. Opinnäytetyössä keskitytään Grasshopperin käyttöön, ja Rhino toimii vain alustana sen käytölle.

3.1.1 Rhinocerosin historia

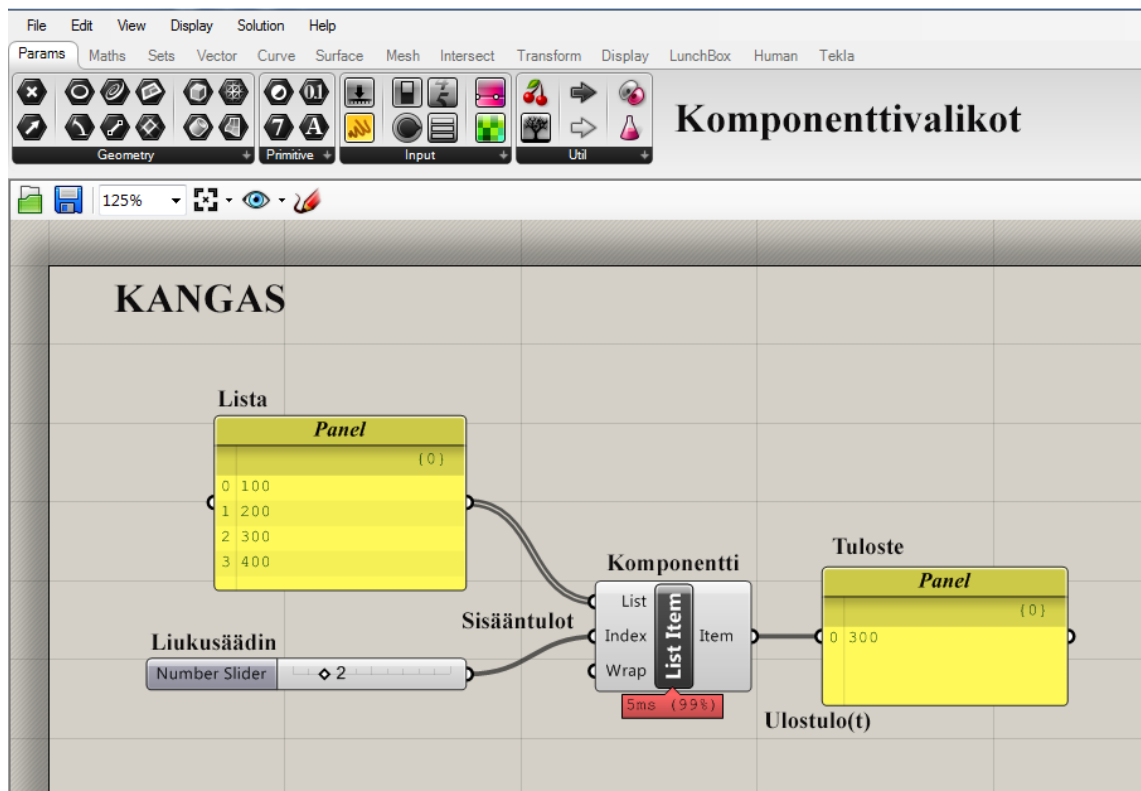
Rhinocerosin on kehittänyt amerikkalainen Robert McNeel & Associates. Yritys on perustettu vuonna 1980. Rhinon historia alkaa vuoden 1992 toukokuulta, kun yrityksessä pidettiin ensimmäinen tapaaminen yhteistyössä AutoCadin kanssa liittyen NURBS-mallintamiseen. Noin vuoden päästä maaliskuussa 1993 julkaistiin ensimmäinen ohjelma nimeltä Sculptura, jolla pystyttiin mallintamaan verkkopintoja. Saman vuoden heinäkuussa saatiin valmiiksi prototyypiversio Sculptura 2, joka oli valmis hyödyntämään NURBS -mallinnusta. Kehityksen edetessä Sculptura 2 sai lempinimen Rhinoceros. Lopulta se päättyi ohjelman nimeksi vuonna 1994 tavaramerkkiin liittyvien kiistojen vuoksi. (McNeel Wiki, n.d.)

Vuoden 1998 lokakuussa julkaistiin ohjelman ensimmäinen versio nimeltä Rhino 1.0 Windows -käyttöjärjestelmälle. Vuonna 2001 julkaistuun Rhino 2.0:aan julkaistiin renderointiohjelma Flamingo. Tämän jälkeen Rhinon versioihin kehiteltiin luonnos ja piirrosluontoinen renderointiohjelma Penguin vuonna 2003 ja animointiohjelma Bongo vuonna 2004. Viimeisin versio Rhino 6.0 julkaistiin helmikuussa 2018. (McNeel Wiki, n.d.)

3.2 Grasshopper 3D

Grasshopper on visuaaliseen ohjelmointiin tarkoitettu ohjelma, joka toimii tiiviisti Rhinoceros 3D -ohjelman kanssa. Grasshopper ei tarvitse osaamista perinteisestä tekstimuotoisesta ohjelmoinnista. (Vapaasti suomennettu: Davidson, S. Grasshopper 2018.)

Grasshopper sisältää sekä visuaalisen ohjelmointikielen, että työympäristön. Työympäristönä toimii kangas. Kankaalle lisätään valmiiksi ohjelmoituja komponentteja. Komponentin sisääntulot kertovat, minkälaista dataa se tarvitsee toimiakseen. Esimerkiksi data voi olla geometriaa, tekstiä tai numeroita. Käytännössä Grasshopperin komponenttien avulla käytetään Rhinon olemassa olevia työkaluja. Kuvassa 1 on esitetty Grasshopperin työympäristö ja yksinkertainen komponenttiketju. Kankaalle on lisätty komponentti *list item*, jolle on syötetty lista ja indeksi. Indeksillä valitaan listalta tietty arvo, joka tulee ulostulona tyhjälle paneelille tekstimuodossa.



KUVA 1. Grasshopperin työympäristö

3.2.1 Grasshopperin historiaa

Grasshopperin kehitti itävaltalainen David Rutten, joka työskentelee Robert McNeel & Associatesilla. Grasshopper kehitettiin Rhinon käyttäjille tavaksi automatisoida tehtäviä ilman tekstimuotoista ohjelmointia. Alun perin ohjelmaa kutsuttiin nimellä ”*Explicit History*”, koska siinä oli eri lähestymistapa mallinnushistorian käsittelyyn kuin Rhinossa. (David Rutten, 2013.)

Ensimmäinen versio Grasshopperista julkaistiin syyskuussa 2007. Maaliskuusta 2008 eteenpäin Grasshopper oli mahdollista ladata lisäosaksi Rhinoceros 4.0 versioon. Vasta vuonna 2014 huhtikuussa saatiin julkaistua käyttöön ensimmäinen toimiva ja vakaa versio Grasshopper 1.0. Grasshopper on ollut monta vuotta vain ladattavana lisäosana, mutta Rhino 6.0 versiosta lähtien se on lisätty kiinteäksi osaksi ohjelmaa. (McNeel Wiki, n.d.)

3.3 Tekla Structures ja Tekla-live link

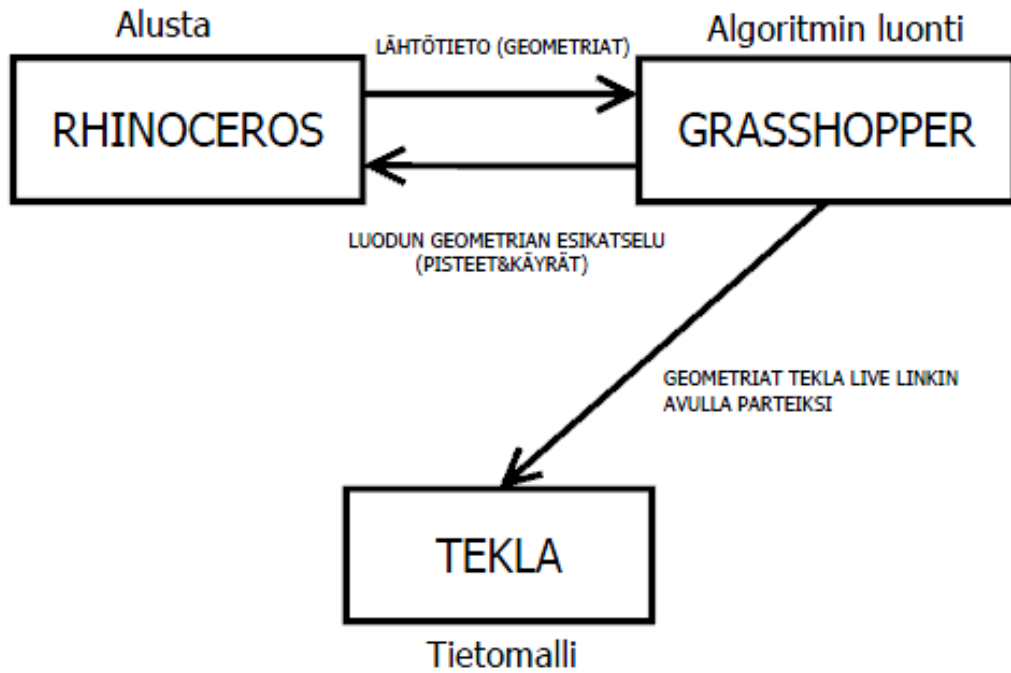
Tekla Structures on BIM-ohjelmisto, joka on kehitelty korkealaatuiseen rakennesuunnitteluun. Tekla Structuresilla luodut mallit sisältävät sen tarkan, luotettavan ja yksityiskohdallisen tiedon, jota tarvitaan onnistuneeseen rakentamisen tietomallinnukseen ja toteutukseen. Tekla Structures tarjoaa sujuvamman työkulun ja toteuttamiskelpoiset tietomallit. (Trimble. 2018. Tekla Structures). Opinnäytetyössä käytetään versiota Tekla Structures 2016i.

Grasshopperin Tekla-live link mahdollistaa algoritmipohjaisen mallintamisen Tekla Structuresiin käyttämällä Rhino – Grasshopper yhdistelmää. Linkki sisältää komponentteja, joiden avulla voidaan luoda ja muokata objekteja reaaliajassa Teklaan. (Trimble. 2018. Tekla-live link)

Opinnäytetyössä Teklan toiminnan ymmärtäminen on melko suuressa roolissa, jotta algoritmipohjasta saadaan toimiva. Lisäksi Tekla luo haasteita yleispätevän algoritmin luomiselle. Vaikka Grasshopperissa luodun datan muuttaminen Teklan osille sopivaksi ei ole vaikeaa, on yleispätevien parametrusten poikkileikkausten tekeminen ja niiden muuttajien määrittäminen algoritmipohjan kautta jo työläämpää.

3.4 Ohjelmien käyttö ja tietomallin muodostuminen

Ohjelmien yhteistyön onnistumiseksi ohjelmat tulee käynnistää järjestyksessä Tekla, Rhino ja Grasshopper. Malli syntyy Tekla-komponenttien kautta Teklaan. Rhino toimii tässä tapauksessa vain katseluohjelmana Grasshopperissa luoduille geometrioille, kuten pisteille ja käyrille. Pisteiden käyttö ja hallitseminen on tärkeässä osassa opinnäytetyötä, koska Teklan mallinnettavat osat tarvitsevat aina määrittelypisteet, joiden välille kappale muodostuu. Ohjelmien yhteyttä toisiinsa on avattu kuviossa 1.



KUVIO 1. Opinnäytetyössä käytettyjen ohjelmien yhteydet toisiinsa

Grasshopper sisältää useita valmiiksi ohjelmoituja komponentteja, joita yhdistelemällä luodaan algoritmipohja. Komponenteissa voi olla useita sisään- ja ulostuloja. Jokaista näistä voidaan säätää muutamalla komponentin sisäisellä komennolla ja samaan sisään-tuloon voidaan viedä useasta lähteestä dataa.

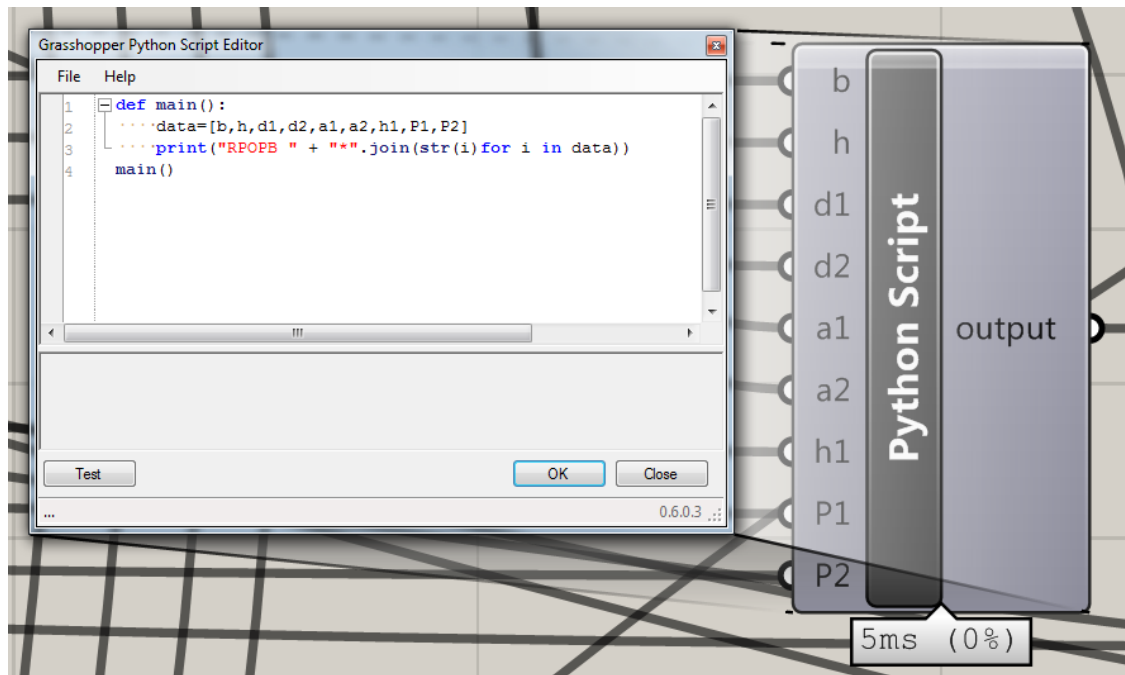
Yleensä komponenttiin voi viedä sisään vain tietyssä muodossa olevaa tietoa tai muuten komponentti ei toimi ja näin ulostulevaa tietoa ei synny. Komponenteille voidaan myös tuoda tietoa muun muassa Rhinosta tai Teklasta. Esimerkiksi Rhinossa olevan käyrän voi määrittää manuaalisesti *curve* komponentille.

3.4.1 Perinteisen ohjelmoinnin käyttö algoritmipohjassa

Vaikka Grasshopper onkin pääasiallisesti visuaalista ohjelmointia, siinä voidaan käyttää hyväksi myös perinteistä tekstimuotoista ohjelmointia. Esimerkiksi opinnäytetyössä muodostetussa algoritmipohjassa on käytetty erikseen ladattavaa *Python script* komponenttia, jolla voidaan hyödyntää Python ohjelmointikieltä. Pythonin lisäksi Grasshopper

tukee muun muassa C# ja VB.net kieliä. Ohjelmointikieliä voidaan hyödyntää Grasshopperissa muutenkin, kuin skriptatuilla komponenteilla. Grasshopperiin on mahdollista itse ohjelmoida laajennuksia esimerkiksi Visual Studion avulla.

Pythonia päädyin käyttämään sen yksinkertaisuuden, selkeyden ja luettavuuden vuoksi. Lisäksi Pythonia käytetään Grasshopper-yhteisössä laajalti, joten kirjastoja ja esimerkkejä löytyy käyttäjien foorumeilta melko paljon. *Python script* komponentti antaa hyvät lähtökohdat ohjelmoinnin harjoittelulle, koska komponentti toimii muiden komponenttien tavoin. Se luo mahdollisuuden luoda uusia vaihtoehtoja valmiiden komponenttien käytölle, sekä suorittaa toimintoja, joihin valmiilla komponenteilla ei pystytä. Esimerkiksi komponentilla voidaan luoda yksinkertaisia ehtolausekkeita keskelle algoritmiketjua. Kuvassa 2 on esitetty esimerkki Pythonin käytöstä algoritmipohjassa.

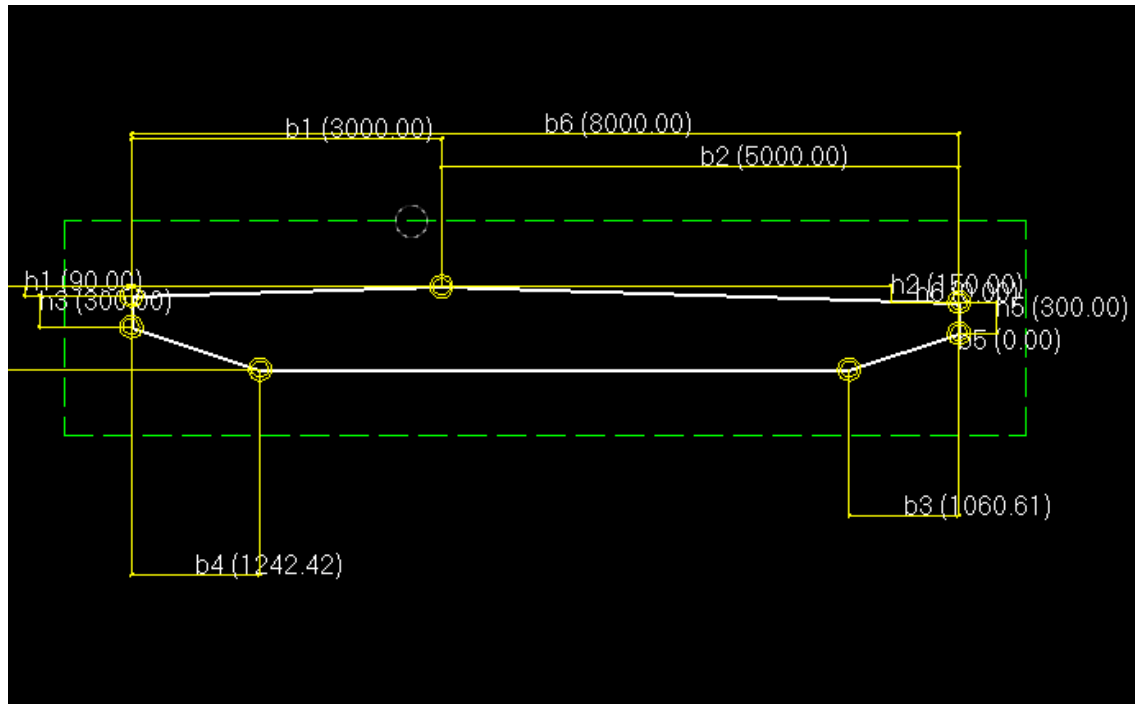


KUVA 2. *Python script* komponentti Grasshopperissa

3.4.2 Sketch editor ja variable cross section

Teklassa on mahdollista tehdä omia poikkileikkauksia monella eri tavalla. Opinnäytetyössäni päädyin luomaan poikkileikkaukset käyttämällä sketch editoria. Sillä voidaan luoda parametrisia poikkileikkauksia, jonka ansiosta muuttujia on mahdollista hallita al-

goritimpohjan kautta. Lisäksi sketch editorissa määritetään poikkileikkauksen määrittelypisteet seuraamaan dimensioiden muutoksia. Kuvassa 3 on esimerkki sketch editorilla luodusta poikkileikkauksesta.



KUVA 3. Kannen poikkileikkaus sketch editorissa

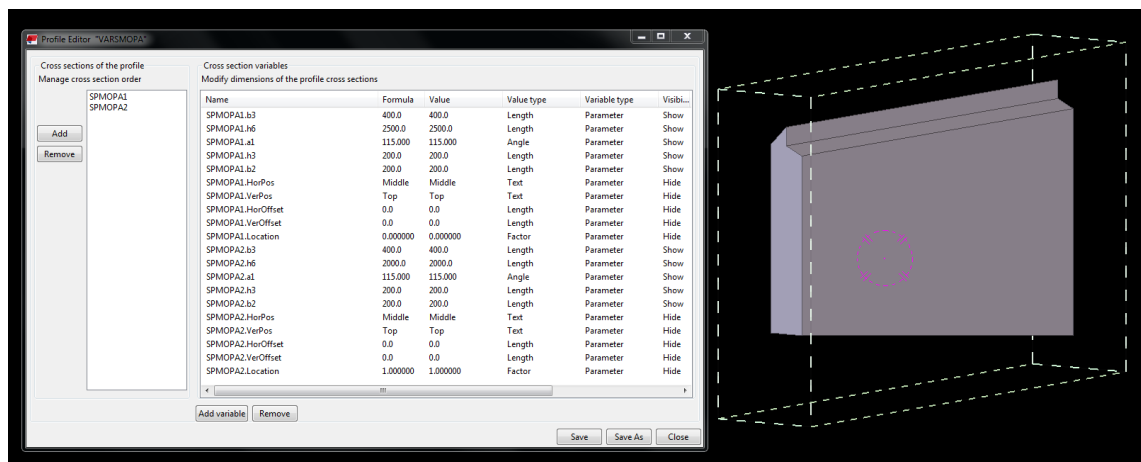
Luotu poikkileikkaus syötetään Teklan palkkiobjektille attribuuttitietona. Palkkiobjekti muodostuu annettujen määrittelypisteiden välille. Opinnäytetyössä palkkiobjektien määrittelypisteet on sijoitettu tasausviivalle.

Poikkileikkaukselle määritetään halutut dimensiot ja kerrotaan ohjelmalle ne dimensiot, joita voidaan muuttaa. Dimensioiden välille voidaan luoda myös riippuvuuksia toisistaan. Näin saadaan vähennettyä syötettäviä muuttujia. Lisäksi poikkileikkaukset toimivat paremmin, koska parametrien välille ei voi tulla ristiriitoja. Poikkileikkauksen parametrista on esitetty kuvassa 4.

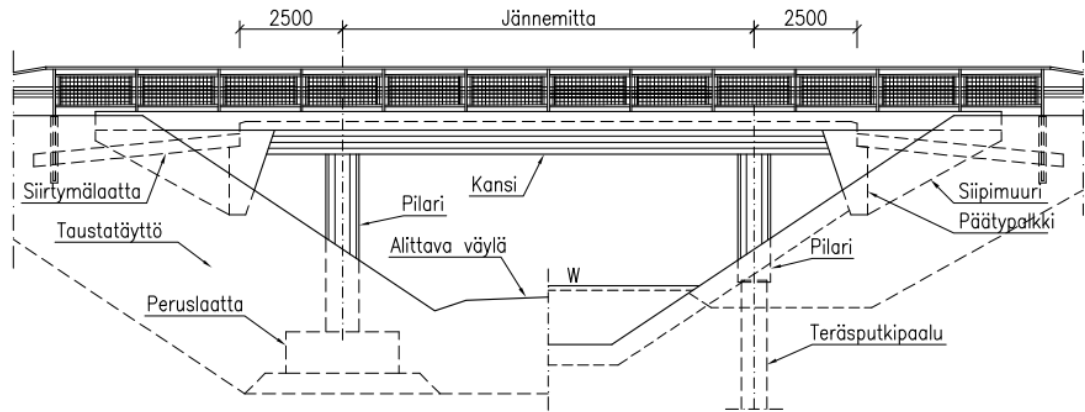
Name	Formula	Value	Value type	Variable type	Visibility	Label in dialog box
h1	=P1*b1	90.0	Length	Distance	Hide	
b1	3000.0	3000.0	Length	Distance	Show	
b2	5000.0	5000.0	Length	Distance	Show	
h2	=P2*b2	150.0	Length	Distance	Hide	
h3	300.0	300.0	Length	Distance	Show	
h5	=h3	300.0	Length	Distance	Hide	
h4	800.0	800.0	Length	Distance	Show	
b4	=(h4-h1-h3)/P3	1242.4	Length	Distance	Hide	
b3	=(h4-h2-h3)/P3	1060.6	Length	Distance	Hide	
b5	0.0	0.0	Length	Distance	Hide	
h6	0.0	0.0	Length	Distance	Hide	
b6	=b1+b2	8000.0	Length	Distance	Hide	
P1	0.030000	0.030000	Factor	Parameter	Show	Parameter1
P2	0.030000	0.030000	Factor	Parameter	Show	Parameter2
P3	0.330000	0.330000	Factor	Parameter	Show	Parameter3

KUVA 4. Kannen poikkileikkauksen parametrit

Sketch editorilla luodusta poikkileikkauksesta voidaan muokata muuttuva poikkileikkauksinen eli *variable cross section*. Tämä tarkoittaa esimerkiksi sitä, että tietyssä kohdassa mallinnettua palkkiobjektia sen korkeutta voidaan muokata. Muokkaus mahdollistetaan määrittämällä kyseinen mitta muokattavaksi parametriksi sketch editorissa. Toiminto mahdollistaa esimerkiksi siipimuurien viisteen parametriseen mallintamiseen. Algoritmipohjassa on hyödynnetty juuri tätä ominaisuutta. Kuvassa 5 on esitetty siipimuuriobjekti muuttuvana poikkileikkauksena.



KUVA 5. Muuttuvapoikkileikkauksinen siipimuuriobjekti



KUVA 7. Teräsbetonin ulokelaattasilta. (Liikennevirasto, 2017. Teräsbetoninen ulokelaattasilta (Bul) LO 32/2017, 7)

4.2 Ulokelaattasilan suunnittelu

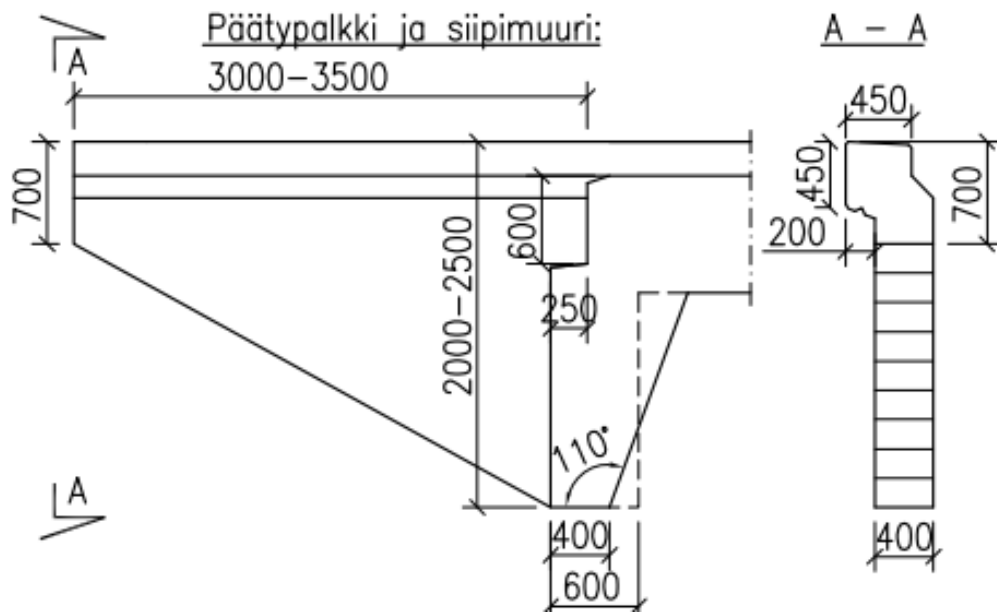
Suomessa teräsbetonista ulokelaattasiltaa käytetään yleisimmin alikulkukäytävänä, vesistösiltaana tai risteyssiltana. (Liikennevirasto. 2017. Teräsbetoninen ulokelaattasilta (Bul) LO 32/2017, 7.)

Algoritmipohjassa on hyödynnetty Liikenneviraston teräsbetonisten ulokelaattasiltojen suunnitteluohjetta sekä betonirakenteiden suunnitteluohjeita. Suunnitteluohjeessa on esitetty tyyppiratkaisu ulokelaattasilasta.

Algoritmipohjan säädettäviä parametreja on rajattu ohjeiden mukaisiksi ja kaikki luodut poikkileikkaukset noudattavat ohjeita. Pohjaa ei kuitenkaan ole luotu toimimaan täysin tyyppiratkaisujen ehtoilla, koska sitä on tarkoituksena kehittää tulevaisuudessa koskemaan paremmin myös muita teräsbetonisiltatyyppejä. Lisäksi parametreja pitää olla mahdollista säätää poikkeustapauksia varten.

4.2.1 Kansi ja reunapalkki

Ulokelaattasiltojen suunnitteluohjeessa on määritetty sillan pituussuunnassa mahdolliset vaihtoehdot jännemitoille, kannen hyötyleveyksille ja rakennepaksuuksille. Näiden parametrien mukaan määräytyy myös pilarien määrä, koko ja sijainnit tukilinjalla. Kuvassa 8 on esitetty ohjeissa esitetty tyyppiratkaisu kannen profiilista. Kannen hyötyleveyden



KUVA 9. Siipimuurin ja päätypalkin tyyppiratkaisut (Liikennevirasto, 2017. Teräsbetoninen ulokelaattasilta (Bul), LO 32/2017, 23)

4.2.3 Välituet

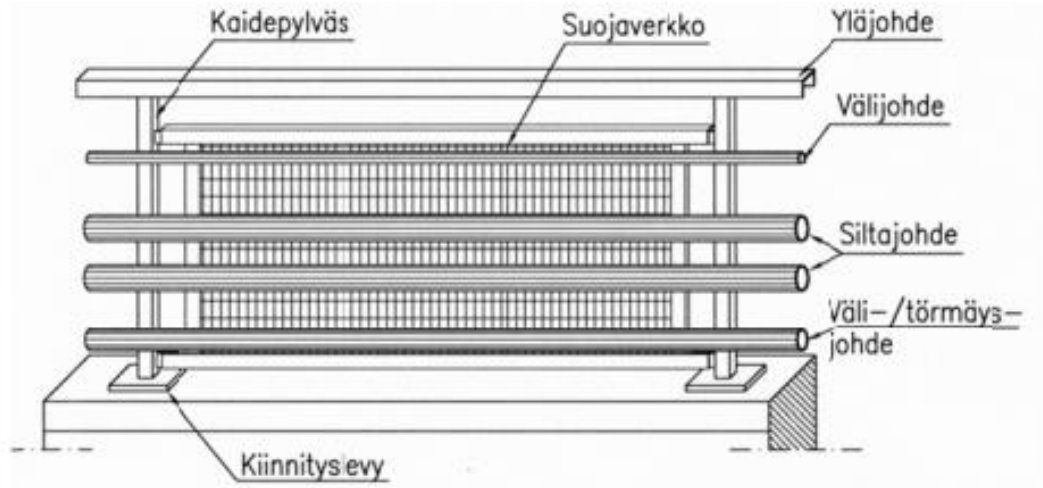
Välituille tulevien pilarien halkaisijat ovat siis sidonnaisia sillan jännemittoihin. Pilareita voi olla yhdellä tuella ohjeen mukaan maksimissaan vain kolme ja vähintään kaksi. Pilarien määrä on sidonnainen kannen hyötyleveyteen. Välitukien vinous määräytyy alittavien väylien mukaisesti.

Ulokelaattasilta voidaan perustaa maan- tai kallionvaraisesti peruslaatoilla. Peruslaatat suunnitellaan tapauskohtaisesti. Lisäksi teräspaalu-perustus on mahdollinen lyötävillä teräspalkkipaaluilla tai porapaaluilla. Tällöin paalu jatkuu teräsbetonipilarina kiinni kanteen. Sillan alusrakenteet suunnitellaan tapauskohtaisesti perustettaessa paaluille.

4.2.4 Kaitteet

Kaikilla silloilla tulee olla kaitteet, jotka mahdollistaa turvallisen liikkumisen sillalla. Kaitteet kuuluvat sillan yksityiskohtaiseen suunnitteluun ja ne pitää suunnitella aina hankekohtaisesti. Sillankaitteet suunnitellaan Liikenneviraston ohjeella 25/2012 Siltojen kaitteet. Kaitteena voi toimia teräsbetonisella ulokelaattasilalla esimerkiksi teräskaitte tai teräsbe-

toninen melukaide. Kaiteet asennetaan reunapalkkien päälle. Opinnäytetyön algoritmipohjalla sillalle voidaan luoda yksinkertainen teräksinen sillankaide. Kuvassa 10 on esitetty sillan teräskaitteen rakenneseosia.



KUVA 10. Teräksisen sillankaiteen rakenneseosia (Liikennevirasto, 2012. Siltojen kaiteet LO 25/2012, 6)

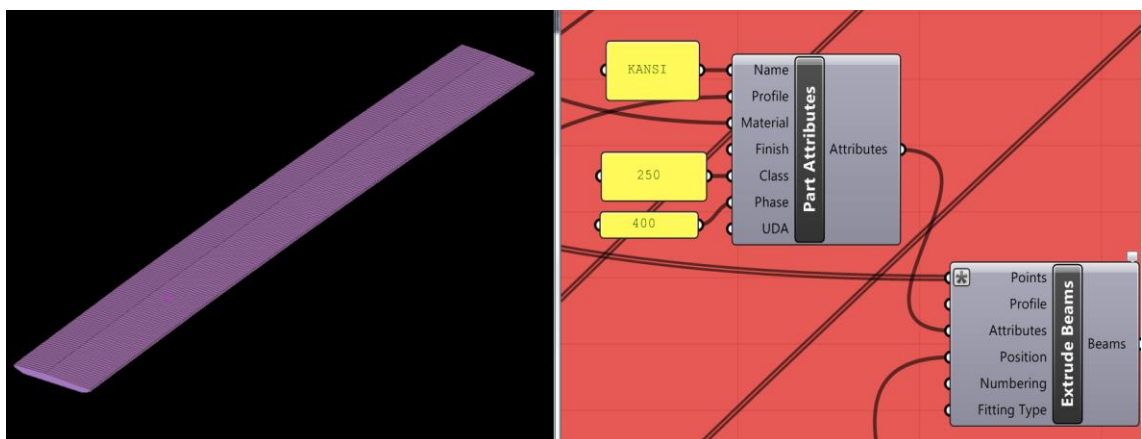
5 LÄHTÖTIEDOT JA ALGORITMIN RAKENNE

5.1 Sillan tietomallin lähtötiedot

Suunnitteluun lähdetessä tarvitaan siltapaikasta tietyt lähtötiedot oikeissa tiedostomuodoissa. Sillan tietomalliin tarvittavat lähtötiedot on määritetty työnantajan toimesta jo ennen kuin algoritmipohjaa lähdettiin kehittämään. Selvää on, että tietomallin luomiseksi tarvitaan hyvin erilaisia lähtötietoja, kuin vanhanaikaisessa suunnittelussa. Tarvittavat lähtötiedot on esitetty liitteessä 1.

5.2 Algoritmin rakenne ja toiminta

Algoritmi koostuu yksinkertaistettuna kolmesta alueesta. Alussa on muuttuvien parametrien lista. Tämän jälkeen on itse algoritmin toimivuuden kannalta tärkein alue, jossa luodaan tarvittavat pisteet sillan pursottamiseksi sekä anturoiden kaivuja varten tarvittavat pinnat. Pursottamisella tarkoitetaan tässä tapauksessa kannen mallintamista. Kannen pursottamiseksi tasausviivalle luodaan pisteitä halutuilla väleillä, esimerkiksi 0,5 metrin välein ja tämä pistejoukko syötetään Tekla live linkin komponenttiin nimeltä *extrude beams*, joka mallintaa sillan kannen Teklaan. Kuvassa 11 on esitetty *extrude beams* komponentilla pursotettu sillan kansi. Algoritmin viimeinen alue koostuu Teklaan liittyvistä komponenteista ja poikkileikkauksien parametrien säätimisestä.



KUVA 11. Kannen pursotus *extrude beams* komponentilla

5.2.1 Tasausviivan käyttö

Algoritmin toimivuuden kannalta kriittisessä roolissa on tasausviivan hallitseminen. Tasausviiva on avaruudessa kulkeva käyrä, joka koostuu taso- ja pystygeometriasta. Taso- ja pystygeometriat voivat olla suoria, kaaria sekä klotoideja eli muuttuvasäteisiä kaaria. Tasausviiva kuvaa väylän geometriaa sidottuna väylän paalutukseen. Paalutuksella kuvataan sijaintia tasausviivalla, esimerkiksi paalulta 320 paalulle 370 on 50 metriä mitattuna mittaviivaa pitkin.

Tasausviivan pituutta ja sillan jännemittoja mitataan tasossa. Tämän vuoksi TSV pitää muistaa projisoida tasoon. Tällöin siitä tulee kaksiulotteinen käyrä. Projisoidulla käyrällä mitatut pisteet on tämän jälkeen helppo projisoida takaisin Z-suunnassa avaruuden käyrälle.

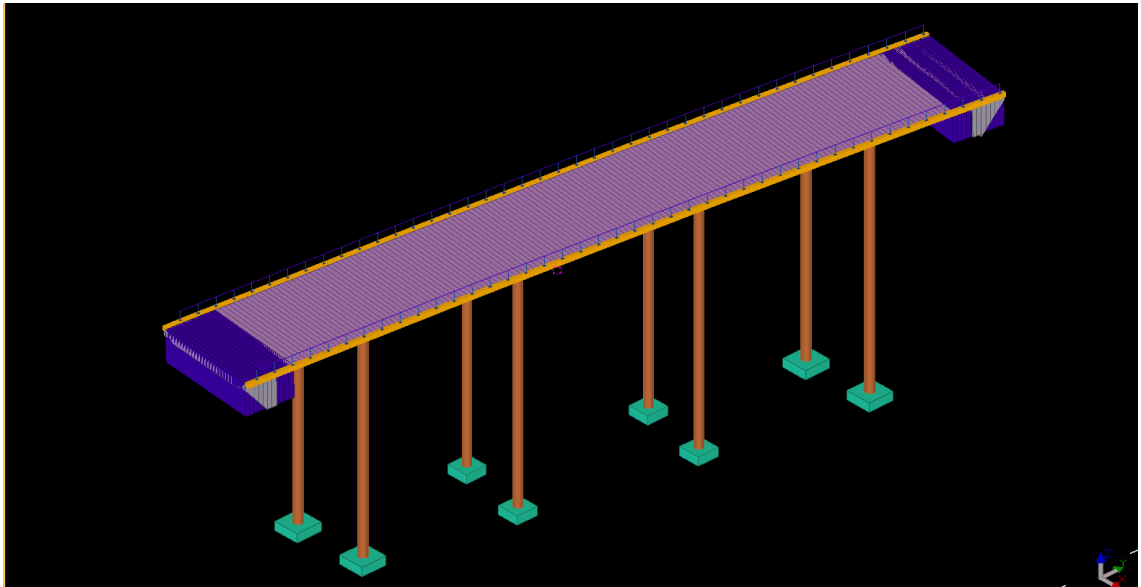
Sillan tasausviiva muodostetaan Rhinoon xls-muodossa olevasta geometriatiedosta. Tiedostossa on ilmoitettu tasausviivan geometria paaluväleittäin. Tiedosto saadaan muutettua käyräksi erillisellä algoritmilla, joka on luotu tätä toimenpidettä varten. Muunnoksen jälkeen TSV-käyrä voidaan *bake* komennolla leipoa Rhinoon. Käyrän niin sanottu leipominen ei ole pakollista, mutta sen avulla TSV pystytään tallentamaan omalle tiedostolle. Ilman leipomista TSV:n geometriatieto pitäisi aina lukea uudestaan Grasshopperin kautta.

5.2.2 Sillan muodostuminen

Sillan mallintamiseksi Grasshopperiin tarvitsee määrittää Rhinoon leivottu sillan tasausviiva ja mahdollisesti syöttää paalulukema. Säättämällä paalulukeman säädintä varmistetaan, että silta osuu varmasti syötetylle tasaukselle. Lisäksi siltapaikan maanpintamallin avulla voidaan suunnitella pilarit ja peruslaatat haluttuun tavoitetasoon, koska algoritmi katkaisee pilarit maanpinnan mukaan. Pilarien pituudet voidaan syöttää myös manuaalisesti.

Algoritmin oletusasetuksilla silta tulee olemaan yksinkertainen yksiaukkoinen ulokelaattasilta. Lähtötilanteen sillan muodostuttua sitä pääsee muokkaamaan muuttamalla parametrilistan arvoja. Sillan parametreja voi myös muokata suuntaa antaviksi ennen tasauksen ja paalulukeman määrittystä.

Kaikkia sillan rakenteita ei saada mallinnettua algoritmipohjan kautta oikeanlaisiksi. Esimerkiksi päätypalkin ollessa vino, se pitää leikata erikseen Teklassa manuaalisesti. Tämän vuoksi algoritmi mallintaa sillan kannen ja päätypalkit pidemmiksi, kuin on tarkoitus. Tämä voidaan havaita kuvasta 12, jossa on esitetty oletusasetuksilla mallinnettu ulokelaattasilta. Lisäksi algoritmi luo päätypalkin paksuuden mukaisesti leikkaustasot. Tämä mahdollistaa komponentin *plane cut* käytön. Komponentille määritetään leikkaustasot sekä päätypalkin ja kannen pursotus ja näin osat leikkaantuvat oikean muotoisiksi.

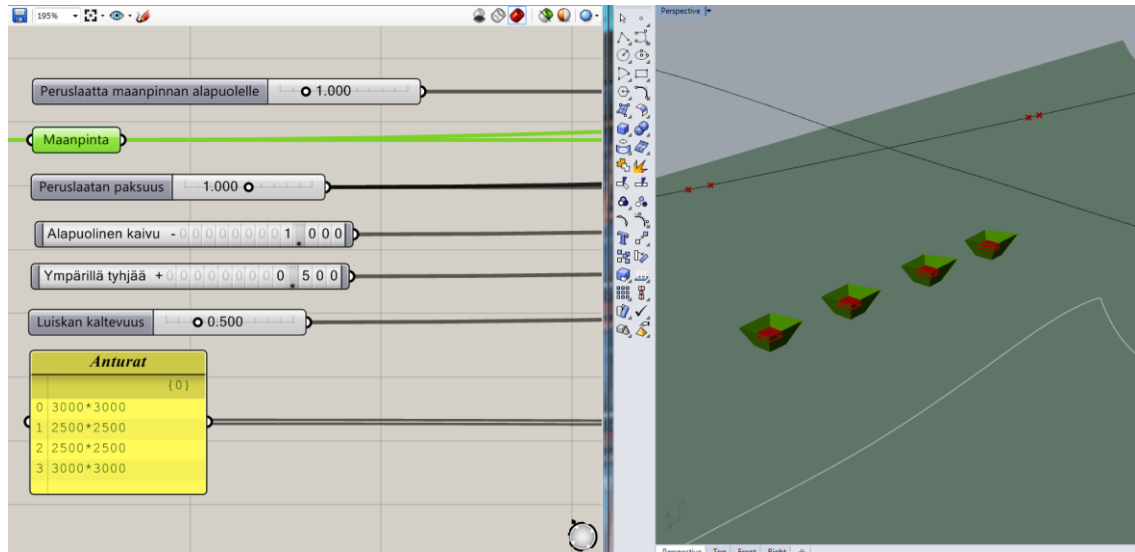


KUVA 12. Oletusasetuksilla mallinnettu ulokelaattasilta

5.2.3 Peruslaattojen kaivannot

Peruslaattojen tarvitsemat kaivannot voidaan mallintaa parametrisesti Grasshopperin avulla Rhinoon ja tarvittaessa viedä pintamallit myös Teklaan. Kaivanto luodaan peruslaatan dimensioiden pohjalta halutun suuruisiksi ja muotoiseksi parametrien avulla. Kaivanto leikataan syötetyn maanpintamallin mukaisesti. Syntyneen kappaleen tilavuus saadaan ulos siihen tarkoitettulla komponentilla. Peruslaattojen ollessa lähellä toisiaan, niiden kaivannot helposti leikkaavat toisiaan, joten kaivantojen algoritmia luodessa tämä pitää

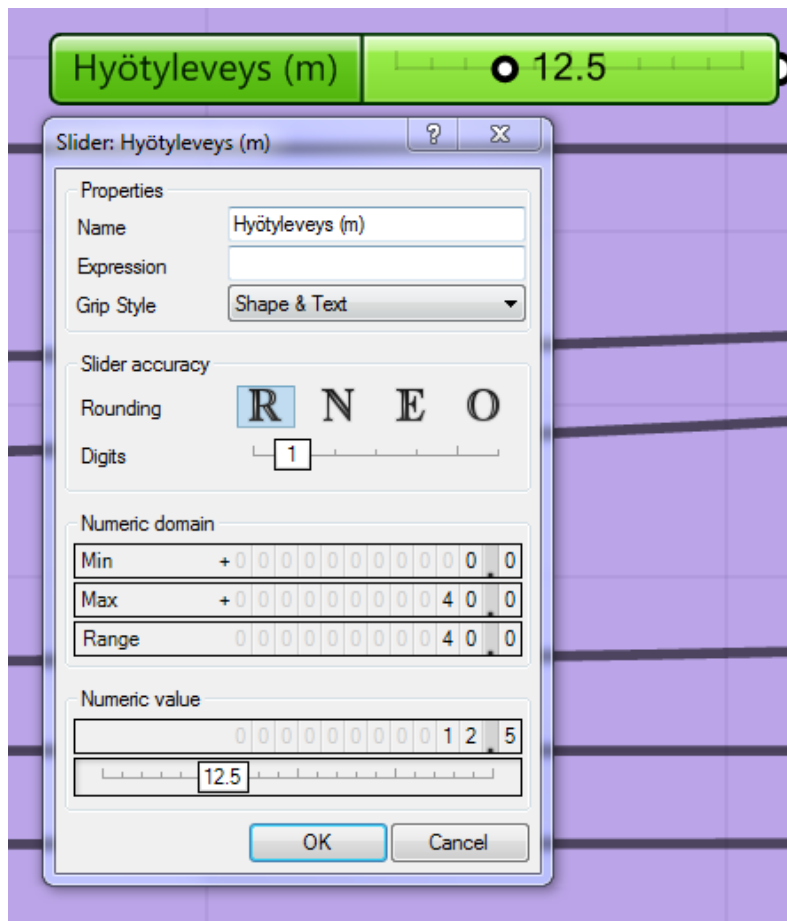
ottaa huomioon. Muuten tilavuuksia laskettaessa saadaan liian suuria määriä. Kuvassa 13 näkyy mallinnetut kaivannot Rhinossa ja niiden parametrit Grasshopperissa.



KUVA 13. Sillan peruslaattojen kaivannot

5.3 Muuttuvat parametrit

Tavoitteena oli kasata mahdollisimman tiivis ja suunnitteluprosessia tukeva lista parametreja, joiden avulla silta saadaan mallinnettua tarkasti. Parametreja voidaan säätää käsin kirjoittamalla tai käyttämällä *number slider* komponenttia. Liukusäädin on esitetty kuvassa 14. Algoritmipohjan liukusäätimien minimi- ja maksimiarvot on määritetty siten, että algoritmin käyttäjä ei voi syöttää arvoja, jotka aiheuttavat algoritmin jumiutumisen. Tätä ei voida hallita paneleihin käsin syötettävissä arvoissa. Esimerkiksi jännemitat syötetään paneleihin.



KUVA 14. Hyötyleveys määritetään algoritmista liukusäätimen avulla

Sillan parametrien lisäksi algoritmipohjasta löytyy, jokaiselle Teklan poikkileikkaukselle tarvittavat säätimet. Poikkileikkaukset valitaan pudotusvalikoista ja osa poikkileikkauksista on kytkettyä toisiinsa. Esimerkiksi kannen poikkileikkaus määrittelee käytettävän päätypalkin profiilin, koska profiilien yläpään kaatojen pitää täsmätä.

5.3.1 Tekla poikkileikkaukset

Teklaan tehdyt poikkileikkaukset on nimetty yhtenäisellä tavalla ja kaikista poikkileikkauksista tullaan tekemään kuvalliset ohjeet algoritmipohjan tulevia käyttäjiä varten. Poikkileikkauksien määrittelypisteet on kiinnitettyinä tasausviivaan ja algoritmi laskee annettujen parametrien avulla palkkiobjektit automaattisesti paikalleen syöttämällä niille positiotiedot.

5.3.2 Teklan osien attribuutit

Siltojen mallintamisen ja numeroinnin selkeyttämiseksi yrityksessä on luotu Teklan siltasuunnittelun numerointiohje. Siinä on määritetty Teklan osien attribuuttien nimeäminen. Näin ollen algoritmipohjaan on syötetty valmiiksi tiedot, joita ei ole syytä perusteettomasti muuttaa. Osien nimi, *phase* ja *class* ovat tällaisia valmiiksi syötettyjä tietoja. Ainoastaan osien poikkileikkausta sekä materiaalia voidaan muokata. Sillan osien materiaaliominaisuudet on syötetty valmiiksi algoritmiin ja jokaiselle osalle voidaan erikseen valita halutut ominaisuudet pudotusvalikoista.

6 LOPPUTULOKSEN ANALYSOINTI

6.1 Algoritmipohjan tekoprosessi

Ohjelmien käytön sisäistäminen oli hyvin oma-aloitteista ja vaati useita tunteja opettelua. Työpaikalta saatu opastus auttoi pääsemään hyvin alkuun. Tämän lisäksi haasteeksi muodostui kokemattomuus siltojen suunnittelusta. Vähäinen tietämys siltojen suunnitteluprosessista korostui työn edetessä, mutta oppimista ja kehittymistä tapahtui hyvinkin nopeasti. Työn alkaessa työpaikalla pidettiin viikoittaisia palavereita ja näin työ saatiin lähdössä haluttuun suuntaan ja viikon aikana heränneille kysymyksille sai vastauksia. Työn etenemistä helpotti se, että algoritmipohjaa pääsi luomaan oikealle siltapaikalle heti alusta lähtien.

6.2 Algoritmipohjan toimivuus

Tavoitteisiin verrattaessa algoritmipohja toimii lähestulkoon niin kuin pitäisikin. Pohjan kautta saadaan mallinnettua ulokelaattasilta ja sen parametrit on selkeästi esillä. Käytävissä on kahta eri kansi-, reunapalkki-, siipimuuri- ja päätypalkkiprofiilia. Kaikki sillan osat eivät vielä toimi täydellisesti kaikilla mahdollisilla asetuksilla sekä muitakin pieniä ”ohjelmointivirheitä” löytyy. Lisäksi algoritmipohjan käyttöohjeet yritykselle ovat vasta tekeillä. Toimivuuden parantamiseksi algoritmipohja tarvitsee käyttäjäkokemuksia ja uusia kohteita, joissa algoritmia voidaan testata.

6.3 Haasteet

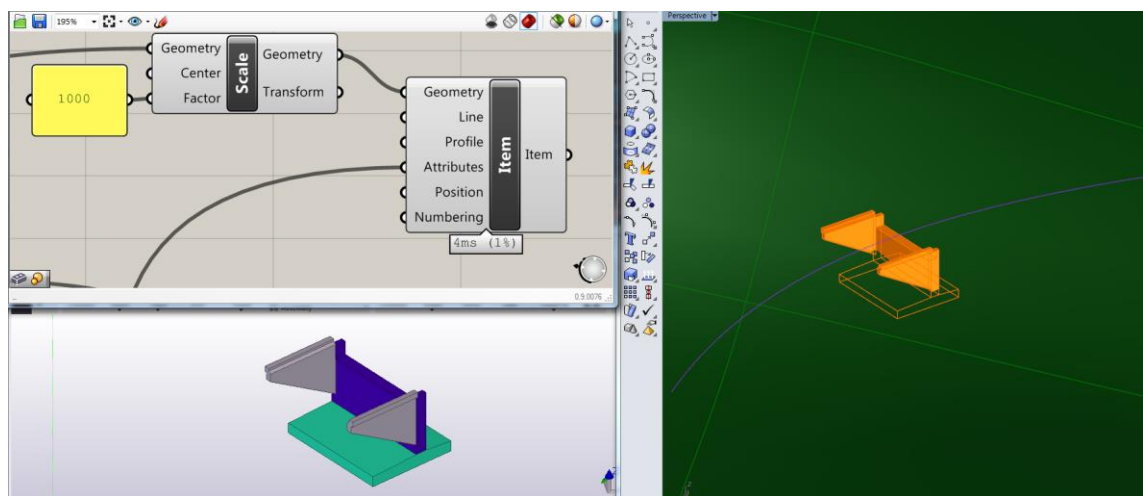
Suurimman haasteen algoritmipohjan yleistämiseksi kaikkiin teräsbetonisiltatyyppeihin aiheuttaa Tekla. Normaalista poikkeaviin kohteisiin pitäisi luoda aina omat poikkileikkaukset ja tämän jälkeen lisätä sen tarvitsemat parametrit algoritmipohjaan. Lisäksi hankaluuksia aiheuttaa leveydeltään muuttuvat kannet sekä sillat, joihin liittyy esimerkiksi ramppi. Opinnäytetyön rajaus yleissuunnitelmatasolle johtuu hyvin pitkälle Teklan kömpelöistä raudoitustyökaluista, joiden hallitseminen Grasshopperin kautta on lähes mahdollonta.

6.4 Jatkokehitys

Opinnäytetyön algoritmissa on melko paljon mahdollisuuksia jatkokehitykselle. Lisäksi olemassa olevia toimintoja voi ja tulee kehittää vielä paremmiksi. Aiemmin mainittuna kehityskohteenä esiin nousi algoritmin muokkaaminen soveltuvaksi muillekin teräsbetonisilloille. Ydinalgoritmi soveltuu jo nyt melko hyvin palkkisilloille, mutta tarvittavat parametriset poikkileikkaukset puuttuvat Teklasta ja parametrien säätimet algoritmista. Tulevaisuudessa onkin tärkeää luoda kirjastoa käytettävistä poikkileikkauksista ja luoda ne mahdollisesti tietyn standardin mukaisesti.

Seuravana esteenä muiden kuin ulokesiltojen mallintamiselle muodostuu massiivisten maatukien puuttuminen algoritmista. Maatukien mallintamiseen voisi tulevaisuudessa käyttää hyödyksi Tekla item -objekteja. Tekla 2017 -version kanssa yhteensopivan Tekla-live linkin kautta on mahdollista mallintaa Tekla itemeitä. Tekla item on kiinteä kolmiulotteinen kappale. Se eroaa muista objekteista esimerkiksi siten, että sillä ei ole määrittelypisteitä.

Tekla itemin mallintaminen Grasshopperin kautta tapahtuu melko eri tavalla, kuin normaalien palkkiobjektien. Grasshopperilla luodaan suljettuja kappaleita, jotka syötetään *item* komponentille. Näin ollen Rhinoon muodostuu myös mallinnettava silta näkyvinä pintoina, toisin kuin käytettäessä *extrude beams* komponenttia. Kuvassa 15 on esitetty esimerkki maatuesta, joka on mallinnettu Tekla itemina.



KUVA 15. Esimerkki maatuen mallintamisesta *item* komponentilla

Opinnäytetyössä keskityttiin tuottamaan algoritmipohjaa pelkästään Teklaa varten. Tulevaisuudessa on mahdollista, että Tekla jää pois siltojen mallintamisesta ja tilalle tulee uusia ohjelmia. Yhtenä vaihtoehtona markkinoilla on saksalaisen Nemetschek Groupin BIM-ohjelmisto Allplan. Algoritmipohja on hyvin haavoittuvainen tällaiselle muutokselle, koska siinä on keskitytty luomaan silta pelkästään pistejoukoista. Tämän muuttujan varjolla tuleekin miettiä, että mihin suuntaan algoritmipohjaista mallintamista ja sen hyödyntämistä tullaan kehittämään jatkossa.

7 POHDINTA

Opinnäytetyön tavoitteena oli mahdollistaa paikallavalettujen siltojen nopeampi, tehokkaampi ja muutoksille vähemmän altis mallintaminen. Lisäksi työn ohella pääsin henkilökohtaisesti melko hyvin kiinni algoritmiavusteisiin mallinnustyökaluihin sekä tietämys siltojen suunnittelusta kasvoi.

Luodulla algoritmipohjalla saadaan parametrisesti mallinnettua teräsbetonisia ulokelaattasiltoja, joten pidän työtä onnistuneena. Kehitystyön jatkuessa algoritmin käytöstä tehdään vielä enemmän käyttäjäystävällisempi ja mukautuvampi mahdollisille erikoistapauksille.

Algoritmipohjaisen mallintamisen ja suunnittelun arkipäiväistymiseen laaja-alaisesti rakennusalalla on vielä jonkin verran matkaa. Kehitys on ollut melko hidasta ja Grasshopperin käyttäjistä vain pieni osa käyttää sitä rakenteiden mallintamisessa. Yhteisön foorumeilla on hyvin niukasti esimerkkejä ja keskustelua liittyen Teklan ja Grasshopperin yhteiskäyttöön, vaikka Tekla-live link laajennuskin on ollut jo useamman vuoden käytössä.

Opinnäytetyö oli opettavainen ja hieno mahdollisuus päästä syventymään aiheeseen, joka tulevaisuudessa tulee melko varmasti olemaan suuremmassa roolissa rakennesuunnittelussa.

LÄHTEET

Liikennevirasto, 2012. Liikenneviraston ohjeita 25/2012. Siltojen kaiteet.

Liikennevirasto, 2017. Liikenneviraston tilastoja 8/2017. Liikennevirastonsillat 1.1.2017.

Liikennevirasto, 2017. Liikenneviraston ohjeita 31/2017. Eurokoodin soveltamisohje: Betonirakenteiden suunnittelu – NCCI 2.

Liikennevirasto, 2017. Liikenneviraston ohjeita 32/2017. Teräsbetoninen ulokelaattasilta (Bul).

Autodesk. Revit www-sivu. 2018. Luettu 10.4.2018. <https://www.autodesk.fi/products/revit-family/overview>

Bentley System. GenerativeComponents www-sivu. 2018. Luettu 10.4.2018. <https://www.bentley.com/en/products/product-line/modeling-and-visualization-software/generativecomponents>

Danil Nagy. Generative Design. Using Python in grasshopper. 4.2.2017. Luettu 1.4.2018. <https://medium.com/generative-design/using-python-in-grasshopper-77bfca86e84b>

David Rutten. Grasshopper forum. GH's Origin. 25.7.2013. Luettu 5.4.2018. <http://www.grasshopper3d.com/forum/topics/gh-s-origin>

Davidson, S. 2018. Grasshopper www-sivu. Luettu 7.2.2018. <http://www.grasshopper3d.com/>

Dynamo www-sivu. 2018. Luettu 10.4.2018. <http://dynamobim.org/learn/>

McNeel Wiki. n.d. The history of Rhino. Luettu 1.4.2018. <https://wiki.mcneel.com/rhino/rhinohistory>

M.A.D. 2018. Rhinoceros. Luettu 7.2.2018. <https://mad.fi/tuotteet/rhinoceros>

Trimble. 2018. Tekla Structures. Luettu 7.2.2018. <https://www.tekla.com/fi/tuotteet/tekla-structures>

Trimble. 2018. Grasshopper-Tekla live link. Luettu 7.2.2018. https://teklastructures.support.tekla.com/not-version-specific/en/ext_grasshopperteklalink

LIITTEET

Liite 1. Sillan tietomallin lähtötiedot

Liite 1

Sillan tietomallin lähtötiedot

- Perinteiset siltapaikka-asiakirjat
Edelleen tarvitaan väylien poikki- ja pituusleikkaukset
- Sillan ylittävien ja alittavien väylien tasausviivojen geometria paaluväleittäin xls-muodossa.
- Sillan kaidelinjat viivoina tasossa ja dwg-muodossa.
- Siltaan liittyvien **väylien pintamallit** (ylittävä ja alittava väylä) 3D-facet ja tai-teviivat samaan tiedostoon 1 metrin pistetiheydellä.
Väylien pintamallien tulee ulottua sillanpäistä ainakin 50 m yli.
- Maastomalli ainakin 50 m säteellä sillasta 3Dface-pintana dwg-muodossa
- Kalliopinta ainakin 50 m säteellä sillasta 3Dface-pintana dwg-muodossa

HUOM! Maastomalli ja väylien pintamallit tulee olla eri tiedostoissa.