



Tukimuurin raudoitus

Algoritminen mallintaminen

Jesse Aalto

OPINNÄYTETYÖ
Huhtikuu 2021

Rakennus- ja yhdyskuntatekniikan tutkinto
Rakennetekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Rakennus- ja yhdyskuntatekniikan tutkinto
Rakennetekniikka

AALTO, JESSE:
Tukimuurin raudoitus
Algoritminen mallintaminen

Opinnäytetyö 39 sivua, joista liitteitä 0 sivua
Huhtikuu 2021

Opinnäytetyössä kehitettiin Petteri Huuhkan diplomityönä toteuttamaa tukimuurin suunnitteluun tehtyä algoritmia. Tavoitteena opinnäytetyössä oli täydentää olemassa olevaa algoritmia. Algoritmiin lisättiin mahdollisuus mallintaa raudoitus sekä kehittää algoritmiin oma käyttöliittymä käytön helpottamiseksi. Toimeksiantajana opinnäytetyössä oli Ramboll Finland Oy. Opinnäytetyö sisältää luottamuksellista tietoa, joten tarkkaa algoritmin rakennetta ei työssä esitellä.

Opinnäytetyössä onnistuttiin jatkokehittämään diplomityön algoritmia sekä luomaan käyttöliittymä. Kehitetyn algoritmin avulla onnistuu tukimuurin geometrian luomisen lisäksi raudoituksen mallintaminen. Kaikki muutokset suunnitteluvaiheessa tehdään omalla käyttöliittymällä, joka helpottaa algoritmin käyttöä. Työ toteutettiin käyttäen Rhino 6 -pintamallinnusohjelmaa, joka sisältää Grasshopper-nimisen graafisen ohjelmointialustan. Algoritmi toteutettiin Grasshopperilla, joka Tekla Live Link -lisäosan avulla mallintaa tukimuurin sekä raudoituksen Tekla Structures -nimiseen BIM-ohjelmaan. Tekla Structures -ohjelmaa käytetään sen erinomaisten dokumentointityökalujen takia. Opinnäytetyössä selitetään eri ohjelmien tarkoitus sekä ohjelmien toiminta.

Kokonaisuutena algoritmi toimii hyvin, mutta vaatii paljon muutoksia, mikäli jokin ohjelmista vaihdetaan uudempaan versioon. Työmaan manuaalinen toiminta johtaa siihen, että raudoitus tulee suunnitella ja mallintaa huolella, jotta työmaalla rautojen asennus onnistuu oikein. Tavoitteena oli automatisoida suunnittelua, vielä alkuperäistä algoritmia enemmän, mutta tällä hetkellä se toimii vain tiettyyn pisteeseen asti. Usean eri ohjelman käyttö mallintamisessa on aihe, jota tulee kehittää lisää, jotta siitä saadaan enemmän hyötyä, kuin mallinnettaessa yhdellä ohjelmalla.

Asiasanat: tukimuuri, raudoitus, algoritmi, grasshopper, mallinnus

ABSTRACT

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

AALTO, JESSE:
Reinforcement of the Retaining Wall
Algorithmic Modelling

Bachelor's thesis 39 pages, appendices 0 pages
April 2021

Ramboll Finland Oy has commissioned many theses in previous years on subjects that are closely related to this thesis. In this case, the objective was to further develop one of the Master's theses made for the company. Ramboll Finland Oy hoped that this would create a better understanding of the subject area and the result of this thesis could be used in the company.

The purpose of this study was to develop the algorithm that had been made in the Master's thesis. The algorithm did the structural analysis and modeled the geometry of the retaining wall. The main point of the further development was to add an algorithm that can model the reinforcement of the retaining wall. A user-friendly interface was also desired because using the algorithm in Grasshopper was slightly difficult.

The algorithm was made by using Grasshopper which is a graphical algorithm editor in Rhino. The Grasshopper works together with Tekla Structures by using Tekla Live Link extension. The modelling itself is done by using Tekla Structures. In this study, a user interface for controlling the modelling was made in Grasshopper using the Human UI.

It was found that the improved algorithm worked. The reinforcement was modelled correctly, and the quality of the reinforcements was found to be good. The user interface created in this study was found to be user-friendly. Using the algorithm in this user interface was far less difficult than using it in Grasshopper.

The findings suggest that this user interface created here / in this study was found to be user-friendly. Tekla and Grasshopper work fine together but there are limits. The automation does not work properly if the software is updated to a newer version. One solution to this would be to use only one particular program of software instead of multiple different ones. But the main reason why absolute automation is hard to accomplish is the construction site. It is very important that the reinforcement is done correctly on the site so that means that the drawings and the documents should be simple and easily readable. This leads to the fact that the design and modelling of the reinforcement must be done properly.

Key words: retaining wall, reinforcement, algorithm, grasshopper, modelling

SISÄLLYS

| | | |
|---|--|----|
| 1 | JOHDANTO | 6 |
| 2 | ALGORITMINEN SUUNNITTELU..... | 7 |
| | 2.1 Algoritminen ajattelu..... | 7 |
| | 2.2 Ohjelmat..... | 8 |
| | 2.2.1 Rhino | 8 |
| | 2.2.2 Grasshopper..... | 8 |
| | 2.2.3 Tekla Structures ja Tekla Live Link..... | 9 |
| | 2.3 Teräsbetonirakenteinen tukimuuri | 9 |
| | 2.4 Lähtökohta | 10 |
| | 2.5 Käytettävyys..... | 11 |
| 3 | ALGORITMIN RAKENNE | 12 |
| | 3.1 Olemassa oleva algoritmi..... | 12 |
| | 3.2 Ohjelmointi Grasshopperissa | 13 |
| | 3.3 Teklan komponentit Grasshopperissa..... | 14 |
| | 3.4 Käyttöliittymä..... | 22 |
| 4 | RAUDOITUKSEN MALLINTAMINEN | 24 |
| | 4.1 Raudoituksen mallintaminen Teklassa..... | 24 |
| | 4.2 Poikittainen raudoitus..... | 26 |
| | 4.3 Pitkittäinen raudoitus..... | 26 |
| | 4.4 Hakaraudoitus..... | 27 |
| | 4.5 Tartuntaraudoitus | 28 |
| | 4.6 Detaljit | 29 |
| 5 | TULOSTEN ANALYSOINTI | 31 |
| | 5.1 Raudoituksen algoritmi..... | 31 |
| | 5.2 Käyttöliittymän käyttö | 32 |
| | 5.3 Haasteet..... | 34 |
| | 5.4 Jatkokehitys | 34 |
| 6 | POHDINTA | 36 |
| | LÄHTEET..... | 38 |

ERITYISSANASTO tai LYHENTEET JA TERMIT (valitse jompikumpi)

| | |
|-------------|--|
| BIM | Building information model |
| NURBS | Non-Uniform Rational B-splines |
| Kangas | Grasshopperin ohjelmointialusta |
| Komponentti | Grasshopperin ohjelmointityökalu |
| Lista | Grasshopperissa käytettävä tiedostomuoto |
| Puu | Grasshopperissa oleva tiedostopolku |
| Mesh | Kolmioista koostuva pinta |
| Curve | Käyrä |
| Input | Komponenttiin syötettävä tieto |
| Output | Komponentista tuleva tieto |
| Clusteri | Usean komponentin algoritmi pakattuna yhteen |

1 JOHDANTO

Teknologian ja erilaisten ohjelmien kehittyessä on huomattu mahdollisuus automatisoida rakennesuunnittelua. Tämän kaltainen automatisointi nopeuttaa erityisesti yksinkertaisten ja rutiininomaisten rakenteiden suunnitteluvaihetta. Tällä hetkellä automatisointi toteutetaan algoritmiavusteisella suunnittelulla.

Opinnäytetyön aihe ”Tukimuurin raudoitus, algoritminen mallintaminen” pureutuu juuri ongelman ytimeen. Työllä on tarkoitus automatisoida rakennesuunnittelun mallinnusvaihetta. Toimeksiantaja on Ramboll Finland Oy, jolla on kokemusta samankaltaisista opinnäyte- ja diplomitöistä. Monet töistä ovat kuitenkin omia irrallisia kokonaisuuksia, eikä niitä ole pystytty hyödyntämään yrityksessä, koska ne vaatisivat jatkokehitystä. Joten tällä kertaa he halusivat jotain muuta, ja päädyimme jatkokehittämään 2018 valmistunutta diplomityötä.

Koko suunnitteluvaiheen automatisointi vaatisi vielä paljon työtä, vaikka iso osa on jo toteutettu diplomityössä. Opinnäytetyössä pyritään lisäämään raudoituksen mallinnus sekä kehittämään käyttöliittymä, jotta algoritmia olisi mahdollisimman helppo käyttää.

Automatisointi toteutetaan käyttäen usean eri ohjelman hyviä puolia. Graafinen ohjelmointi on itselleni lähes uusi asia, mutta olen kuitenkin saanut työpaikalla mahdollisuuden osallistua siellä järjestettyihin koulutuksiin, joissa aihetta on käsitelty ja opetettu. Koulutuksien ansiosta aihe alkoi kiinnostamaan ja onneksi sain mahdollisuuden toteuttaa opinnäytetyöni aiheeseen liittyen. Algoritminen suunnittelu on kiinnostavaa, koska se avaa paljon uusia mahdollisuuksia rakenteiden ja rakennusten suunnitteluun.

Opinnäytetyössä esitellään käytetyt ohjelmat ja niiden toimintaperiaatteet, sekä se miten ne toimivat yhdessä. Tämän lisäksi esitellään algoritmin eri osia, mutta koska opinnäytetyö sisältää luottamuksellista tietoa, tarkkaa algoritmin rakennetta ei työssä esitellä. Lopuksi arvioidaan saavutettuja tuloksia ja pohditaan, miten algoritmia voidaan hyödyntää tai jatkokehittää tulevaisuudessa.

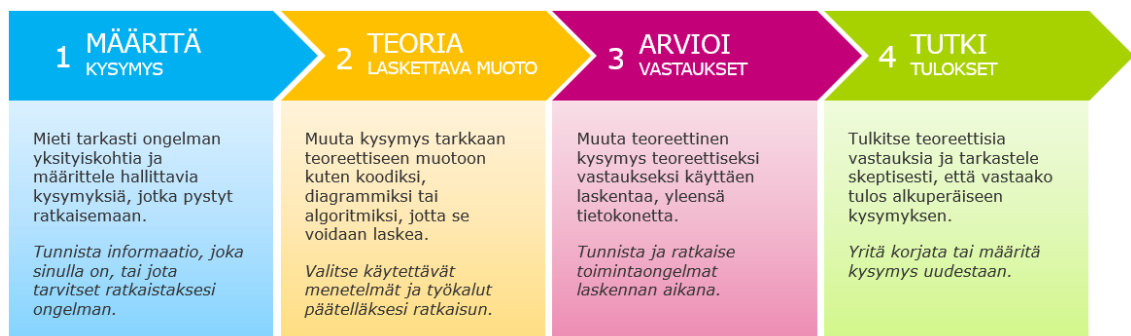
2 ALGORITMINEN SUUNNITTELU

2.1 Algoritminen ajattelu

Algoritmi on yksiselitteinen, mahdollisimman lyhyt toiminto, joka tuottaa ongelmaan ratkaisun (Peda nd; Britannica 2006). Jotta algoritmi toimii, vaati se, että toiminnot ovat oikeassa järjestyksessä sekä että toiminnot ovat yksiselitteisiä. Kun halutaan luoda algoritmi tai automatisoida toimintoja, käytetään algoritmista ajattelua. Algoritminen ajattelu sisältää ongelman purkamisen ja haluttuun lopputulokseen vaadittavien kaavojen tunnistamisen. Algoritmisen mallintamisen tavoitteena on automatisoida suunnittelua eli kyseessä on ongelmien ratkaisujen yleistämistä. (Peda nd.)

Algoritminen ajattelu on helpointa toteuttaa neljä vaiheisella ajatusprosessilla (kuvio 1). Tarkoituksena on määrittää ongelma, johon luodaan jonkinlainen laskettava muoto kuten koodi. Kun kysymys on muutettu laskettavaan muotoon, voidaan ”kysymys” laskea ja siitä saadaan ongelmaan yksi vastaus. Vastausta tulee tämän jälkeen tarkastella ja tutkia, jotta vastaus on todella ratkaisu haluttuun kysymykseen. (ComputationalThinking nd.)

Tässä opinnäytetyössä keskitytään automatisoimaan raudoituksen mallintaminen. Automatisointi vaatii useiden ongelmien ratkaisuja sekä näin ollen algoritmista ajattelua. Automatisoinnilla pyritään helpottamaan ja nopeuttamaan rakenteen suunnittelua.



KUVIO 1. Algoritmisen ajattelun neljä osaa (vapaasti suomennettu: ComputationalThinking nd).

2.2 Ohjelmat

2.2.1 Rhino

Rhino on 3D NURBS -pintamallinnusohjelma. Ohjelmalla voidaan mallintaa tarkasti kookkaita ja monimutkaisia 3D-muotoja, koska fyysisillä ominaisuuksilla ei ole merkitystä mallinnuksen lopputuloksen kanssa. Yleisesti käytössä on kaksi tapaa luoda 3D-muotoja tietokoneella. (Rhino3d 2021b.)

Ensimmäinen tapa toimii siten, että se yhdistää kolmioita toisiinsa ja näin saadaan sekä yksinkertaisia että monimutkaisia 3D-muotoja aikaiseksi. Näitä kolmioista muodostettuja pintoja kutsutaan nimellä mesh. Myös mesh:illä päästään hyvään tarkkuuteen pienentämällä kolmioita. Mutta koska pinta koostuu kolmioista, se ei ikinä ole täysin tarkka. (Rhino3d 2021a.)

Toinen tapa on käyttää NURBS. NURBS tulee sanoista Non-Uniform Rational B-Splines eli ei yhtenäisiä rationaalisia matemaattisia käyriä. NURBS voi esittää tarkasti yksinkertaisia 2D-muotoja kuten viivoja ja ympyröitä tai erittäin monimutkaisia 3D-muotoja, koska ne luodaan tarkasti käyttäen matemaattisia kaavoja. (Rhino3d 2021c.) Monet CAD, CAM, CAE ja CAID -ohjelmat käyttävät NURBS, koska niiden tuottamat 3D-muodot ovat tarkkoja vaativienkin kohteiden suunnitteluun (Rhino3d 2021a).

Ohjelman käyttöä helpottaa se, että se tukee laajasti monia eri tiedostomuotoja ja näin ollen helpottaa työskentelyä muiden ohjelmien kanssa. Rhino on yhteensopiva esimerkiksi BIM-, CAD-, CAM- ja eri visualisointiohjelmistojen kanssa. (Mad 2020.)

2.2.2 Grasshopper

Rhino sisältää myös Grasshopper-nimisen graafisen ohjelmointialustan. Grasshopperin käyttö on suhteellisen helppoa, koska sen opettelemiseen ei vaadita muiden ohjelmointikielien osaamista (Rhino3d 2020). Graafinen ohjelmointikieli toimii siten, että ihminen pystyy visuaalisesti esittämään, mitä ohjelmoinnissa

tapahtuu. Tekstiperusteisessa ohjelmoinnissa koodaajan täytyy ajatella kuten tietokone, jotta ohjelmointi onnistuu. Kun taas visuaalisessa ohjelmoinnissa koodaajan on mahdollista kuvailla ohjelmointiprosessia tavalla, joka meille ihmisille on järkevää. (Matthew Rivell 2019.)

Grasshopperin avulla voidaan luoda algoritmeja, joiden perusteella lähtötietoja muuttamalla, mallissa tapahtuu halutut muutokset (Grasshopper3d 2020).

2.2.3 Tekla Structures ja Tekla Live Link

Tekla Structures on edistynyt BIM-ohjelma, jolla pystyy mallintamaan erilaisia 3D-rakenteita (Tekla 2020). BIM-mallinnus mahdollistaa tärkeiden tietojen dokumentoimisen malliin. Maailman kasvava väkimäärä vaatii järkeviä ja tehokkaita ratkaisuja. BIM-mallinnuksen avulla sekä rakentaminen että suunnittelu, on sujuvampaa ja tehokkaampaa. Rakennuksen loppukäyttöä ajatellen, jo rakentamisen aikana saadaan kerättyä tietoa rakennuksen toiminnasta. Tämä tieto voi auttaa rakennuksen korjaamisessa ja huoltamisessa jonain päivänä. BIM-mallia voidaan siis käyttää koko rakennuksen elinkaaren ajan suunnittelusta loppukäyttöön ja huoltoon sekä korjaukseen. (Autodesk 2020.)

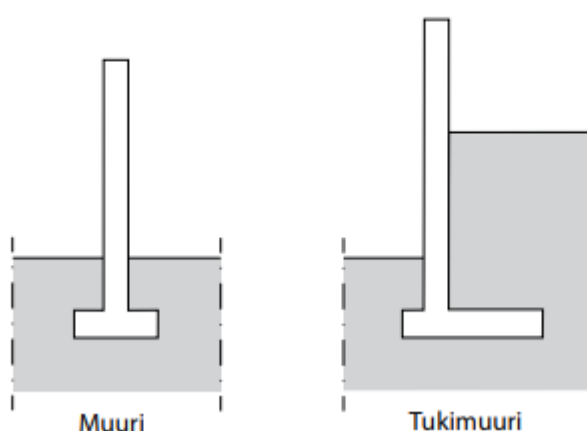
Tällä hetkellä Grasshopperin avulla tehty malli viedään Tekla Structures -ohjelmaan, koska siinä on paremmat ominaisuudet rakennepiirustusten tekemiseen ja dokumentoimiseen. Jotta Grasshopper ja Tekla Structures toimivat yhdessä, vaaditaan erillinen Tekla Live Link -lisäosan. Tämän lisäosan avulla Grasshopperilla tapahtuva algoritminen mallinnus toimii samanaikaisesti myös Tekla Structuresissa. (Teklastructures 2020.)

2.3 Teräsbetonirakenteinen tukimuuri

Tukimuureja käytetään rajaamaan erilaisia alueita, kun maastossa on korkeuseroja. Tukimuurin materiaali voi olla esimerkiksi teräsbetoni, kivi tai puu. (RT 89-11175 2015.) Teräsbetoni on materiaali, jossa betonia on vahvistettu teräksellä.

Rakenteessa käytetään hyväksi teräksen hyvä vetokestävyys ja betonin hyvä puristuskestävyys (The Editors of Encyclopaedia 2020).

Tukimuurin vastaanottaa vaakasuoria voimia, kuten maanpaine. Tässä opinnäytetyössä käsitellään kulmatukimuuria eli poikkileikkaukseltaan L-kirjaimen muotoista tukimuuria (kuva 1). Kulmatukimuurin etuna on, että sen pohjalaatan päälle kasattu maa yhdessä tukimuurin oman painon kanssa aikaansaa rakenneteknisen vakavuuden. (RT 89-11175 2015.)



KUVA 1. Tukimuurin periaatteellinen poikkileikkaus (RT 89-11175 2015).

2.4 Lähtökohta

Opinnäytetyö keskittyy jatkokehittämään Petteri Huuhkan diplomityötä Automation of Construction Planning of a Reinforced Concrete Retaining Wall. Huuhka (2018) tutki diplomityössään mahdollisia menetelmiä rakennesuunnittelun automatisointiin. Esimerkkinä työssä tehdään tapaustutkimus teräsbetonirakenteisesta kulmatukimuurista. Automaattiseen systeemiin tähtäävä algoritmi on tehty käyttäen Rhino 6 ja Grasshopper-ohjelmia. (Huuhka 2018.)

Nykyinen algoritmi mitoittaa ja mallintaa koko tukimuurin geometrian. Tukimuurin on mahdollista mitoittaa käyttäen eurokoodia tai NCCI 7, joka on geotekniseen suunnitteluun keskittyvä eurokoodin soveltamisohje.

Huuhka esittää diplomityössä yhdeksi jatkokehityskohteeksi raudoituksen mallintamisen. Raudoituksen mallinnus on luonteva lisä algoritmiin, koska sillä saadaan lisättyä automatisointia tukimuurin suunnitteluun.

2.5 Käytettävyys

Käytettävyydellä tarkoitetaan tässä opinnäytetyössä algoritmin käyttöä. Käyttöliittymän suunnittelussa tulisi huomioida se, että käyttäjän virheellinen toimintatapa johtuu epäonnistuneesta käyttöliittymästä eikä käyttäjästä itsestään. Käyttöliittymän suunnittelussa tulee ottaa huomioon monia asioita. Eri kulttuureissa on erilaiset tavat toimia, jolloin universaalien käyttöliittymien aikaansaaminen vaatii työtä. Tämän lisäksi ihmiset ovat oppineet elämänsä aikana erilaisia asioita sekä heillä on erilaiset tavat tietotekniikan kanssa. Tähän kun lisätään ihmisen henkilökohtaiset taidot ja käyttäytyminen, joihin vaikuttavat muun muassa ajattelu, hahmotaminen, lukutaito ja motoriikka niin hyvän ja toimivan käyttöliittymän suunnitteluun kannattaa käyttää aikaa. (Röksä 2020.)

Käyttöliittymä toteutetaan Grasshopperiin saatavalla lisäosalla Human UI. Human UI:n avulla on mahdollista luoda käyttöliittymä, joka näyttää ja tuntuu oikealta Windows ohjelmalta. Se ei myöskään ole riippuvainen Grasshopperin käyttöliittymästä ja näin ollen Grasshopperissa tehty algoritmi on helppokäyttöinen. (Food4rhino 2020.)

3 ALGORITMIN RAKENNE

3.1 Olemassa oleva algoritmi

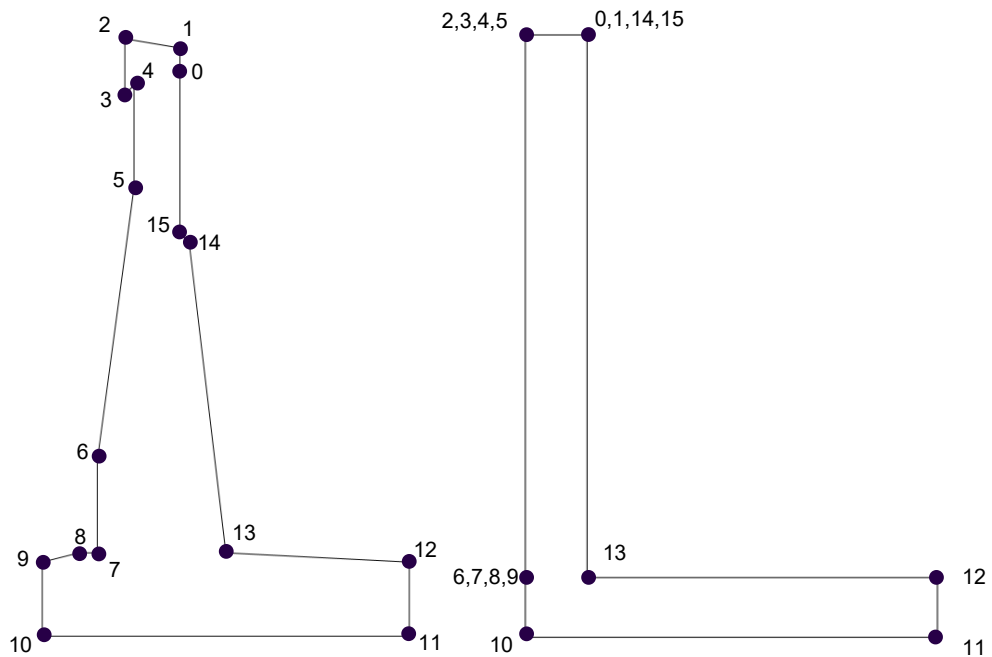
Ennen tämän opinnäytetyön aloittamista, kävimme Huuhkan kanssa lyhyen kokouksen, jossa hän kertoi tarkemmin omasta diplomityöstään. Sekä Grasshopper että Rhino olivat molemmille uusia ohjelmia, jolloin diplomi- sekä opinnäytetyössä opeteltiin käyttämään ja ymmärtämään kyseisiä ohjelmia. Näin ollen tuotoksena syntynyt algoritmi ei välttämättä ole paras mahdollinen.

Huuhkan tekemä algoritmi on toimiva, mutta se vaatii oman aikansa, jotta sen toiminnan ymmärtää. Algoritmi on laaja ja Grasshopperin ympäristö ei ole kaikista helpoin käyttää. Hänen algoritminsä koostuu kolmesta suuremmasta osa-alueesta: lähtötietojen syöttäminen, geometrian muodostaminen ja laskenta. Hän on myös optimoinut suunnittelun eli rakenteen pystyy minimoimaan tai maksimoimaan halutulla tavalla.

Algoritmi vaatii toimiakseen käyttäjän määrittämät lähtötiedot sekä kohteen maanpinnat että tukimuurilinjan. Maanpinnat sekä tukimuurilinja tulee olla Rhinossa, jotta niitä voidaan käyttää algoritmissa. Maanpinnat eli tukimuurin takapuolella oleva yläpuolen maanpinta, että etupuolella oleva alapuolen maanpinta tulee syöttää Mesh-muodossa. Tukimuurilinja eli tukimuurin sijainti ja pituus tulee syöttää curve-muodossa.

Algoritmissa on myös asioita, joita tulee ottaa huomioon raudoitusta mallinnettaessa. Käyttäjä pystyy valitsemaan raudoituspinnalle, että käytetäänkö koko tukimuurin alueella samaa raudoitusta vai jaetaanko se viiteen eri alueeseen. Eli käyttäjä päättää, että onko raudoituspinnalla samaa rautaa ja samaa k-jakoa vai viittä erilaista rautaa ja k-jakoa.

Diplomityössä toteutettu algoritmi luo maanpintojen ja tukimuurilinjan avulla poikkileikkauksen. Poikkileikkaus muodostuu 16 pisteestä (kuva 2) ja niiden sijaintia voidaan muuttaa vaihtamalla lähtötietoja. Näin saadaan muodostettua haluttu ja rakenteellisesti kestävä poikkileikkaus.

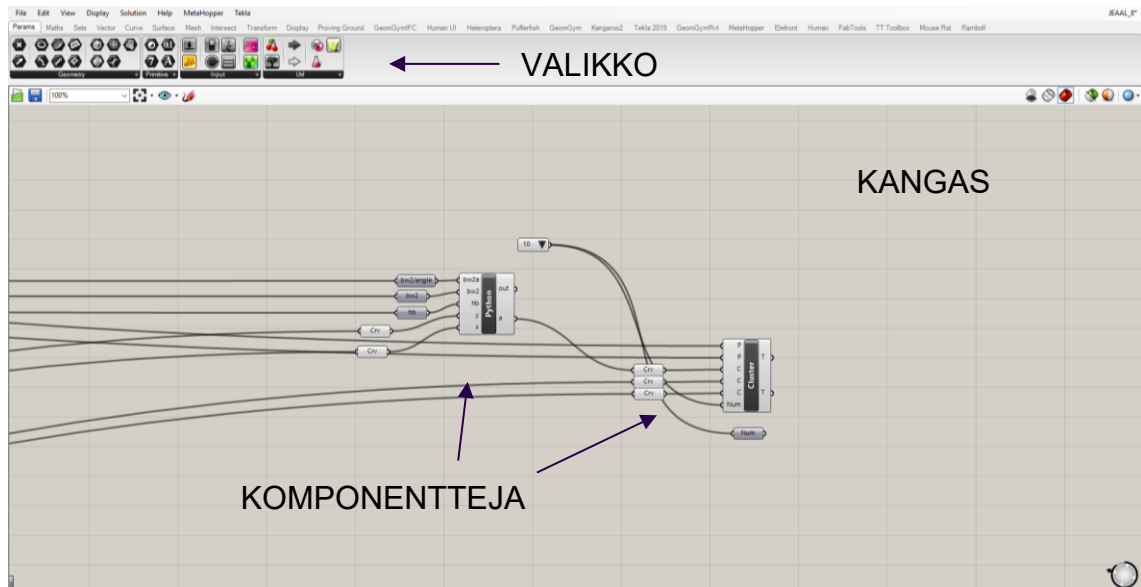


KUVA 2. Tukimuurin poikkileikkauksen luomiseen käytetyt pisteet.

Algoritmissa voidaan päättää poikkileikkausväli eli kuinka usein algoritmi luo poikkileikkauksen. Tämän jälkeen poikkileikkaukset yhdistetään, jolloin tuloksena on yksi geometrinen muoto. Raudituksen geometrian luomisessa on hyödynnetty poikkileikkauksien luomiseen käytettyjä pisteitä.

3.2 Ohjelmointi Grasshopperissa

Grasshopperin toiminta perustuu listoihin ja puihin. Näiden toiminnan ymmärtäminen on tärkeää, jotta ymmärtää miten Grasshopper toimii. Ohjelmointi Grasshopperissa tapahtuu käyttäen ”komponentteja” ja yhdistämällä niitä toisiinsa. Kaiken pohjana on kangas, johon lisätään haluttuja komponentteja (kuva 3). Yhdistäminen tapahtuu yksinkertaisesti yhdistämällä ”johto” oikeaan kohtaan.



KUVA 3. Grasshopperin käyttöliittymä, kangas ja komponentteja.

Ramboll on luonut sisäiset ohjeet algoritmien luomiseen Grasshopperissa. Ohjeen tarkoitus on yhtenäistää ja selkeyttää algoritmeja, jolloin algoritmin käyttäminen ja ymmärtäminen onnistuu muiltakin kuin tekijältä. Tämä on erittäin tärkeää asia, koska tavoitteena on hyödyntää algoritmiavusteisen suunnittelun etuja. Mikäli käyttäjä ei ymmärrä algoritmia tai tekijän täytyy aina ohjeistaa algoritmin käytössä, ei algoritmi ole tällöin hyvä työkalu suunnittelijalle tai yritykselle.

Esimerkiksi ryhmiä tulee käyttää mahdollisimman paljon. Ryhmät koostuvat useista komponenteista, jotka muodostavat yhden osan algoritmista. Ryhmät tulee nimetä ja värittää oikealla värillä. Ryhmien värit tulevat siitä, mitä sen sisältävät komponentit tekevät. Eli jos ryhmä koostuu Tekla Live Link -komponenteista tulee ryhmän olla eri värillä kuin esimerkiksi "normaaleista" Grasshopper komponenteista koostuva ryhmä. Tällä tavoin algoritmin ymmärtäminen helpottuu, koska väreistä näkee suoraan mitä ryhmä tekee. Muita erillisiä ryhmiä ovat muun muassa lähtötiedot ja visualisointi.

3.3 Teklan komponentit Grasshopperissa

Jotta Rhino/Grasshopper toimii Teklassa, vaatii se Tekla Live Link -lisäosan. Tämän ansiosta Grasshopperissa on omat työkalunsa Teklassa mallintamiseen.

Tämän aiheuttaa myös ongelmia, koska kyseessä on kaksi eri ohjelmaa. Teklan komponenteissa on omat rajoitteensa, mikä tekee algoritmisesta mallintamisesta vähemmän automatisoitua.

Kun Teklassa mallinnetaan, toteutetaan raudoitukset omina ryhminä. Tämä aiheuttaa hankaluuksia. Tällä hetkellä voit luoda algoritmilla niin monta raudoitusryhmää, kun algoritmissa on ryhmien luomiseen tarkoitettuja komponentteja. Mikäli tukimuurissa on enemmän raudoitusryhmiä eli tukimuuri on pidempi kuin on tarkoitettu, algoritmi ei toimi.

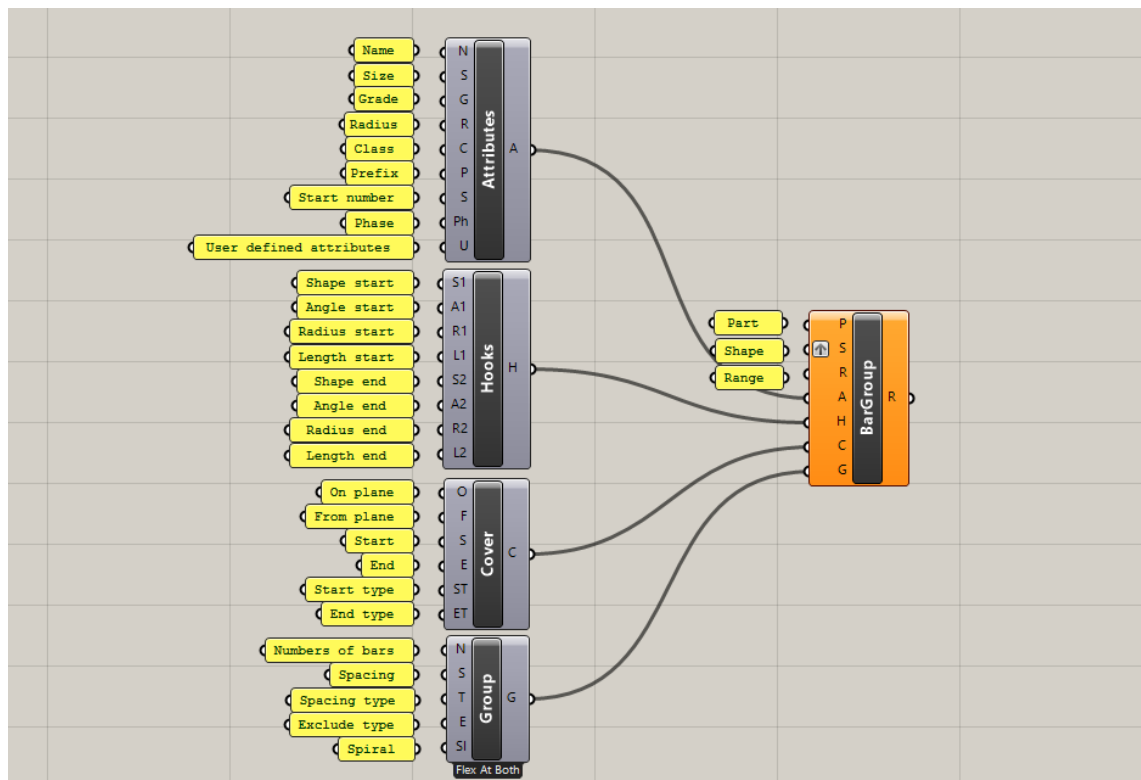
Koska Tekla on BIM ohjelma, niin niitä ominaisuuksia tulee hyödyntää. Tämän takia pelkän raudoitusryhmän luominen ei riitä vaan raudoille pitää myös antaa muita ominaisuuksia (TAULUKKO 1).

TAULUKKO 1. Raudoituksen ominaisuudet Teklassa.

| Yleiset tiedot | Koukut | Peitepaksuus ja k-jako |
|-----------------------|----------------|-------------------------------|
| nimi | koukku alussa | pinnasta |
| teräksen tyyppi | kulma | sivusta |
| halkaisija | taivutussäde | alusta |
| taivutussäde | jalan pituus | lopusta |
| luokka | koukku lopussa | ryhmän k-jaon tyyli |
| numerointi | kulma | rautojen määrä |
| | taivutussäde | rautojen tavoite k-jako |
| | jalan pituus | rautojen tarkka k-jako |
| | | poistetaanko joku rauta |

Taulukossa 1. mainitut asiat tulisi syöttää myös Grasshopperin kautta. Onneksi Tekla Live Link tuo kaikki tarvittavat komponentit Grasshopperiin, jolloin tietojen syöttäminen on yksinkertaista (kuva 4). Ominaisuudet on tärkeää syöttää oikein, koska niitä voidaan hyödyntää myös rakennuksen elinkaaren myöhemmässä vaiheessa. Kuvassa 4 on vasemmassa reunassa ominaisuuksien syöttämiseen tarkoitettut Tekla Live Link -komponentit. Attributes-komponenttiin syötetään yleisiä tietoja kyseisestä raudoitusryhmästä. Seuraavana on hooks-komponentti, jonka

avulla voidaan lisätä raudan geometriaan haluttu koukku. Kolmantena on cover-komponentti ja tällä voidaan syöttää rakenteen vaatima suojabetonipaksuus. Komponentilla voidaan vaikuttaa myös raudan pään tyyppiin eli onko syötetty mitta suojabetonipaksuus vai raudan jalan mitta. Viimeisenä on raudoitusryhmän asetukset eli voidaan määrittää rautojen määrä sekä k-jako, että valita kuinka tarkasti k-jakoa noudatetaan.

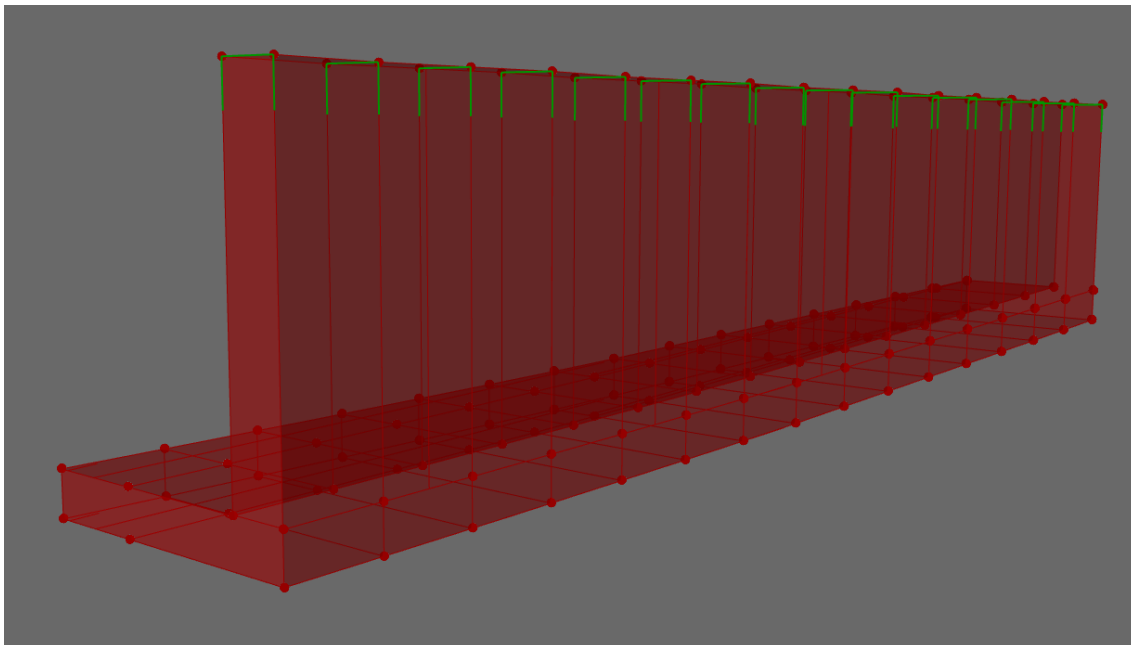


KUVA 4. Raudoituksen tietojen syöttäminen Grasshopperissa.

3.4 Raudoituksen geometrian luominen Grasshopperissa

Raudoituksen geometrian luomiseen on käytetty poikkileikkauksen pisteitä. Jokaisen raudoituspinnan raudoituksen geometria on luotu samalla tyyllillä. Esimerkkinä tässä käytetään tukimuurin yläosan haka. Geometrian luomisessa tuli kiinnittää huomiota vaihtelevaan poikkileikkaukseen. ”Normaalissa” tilanteessa haka on U-mallinen, mutta mikäli tukimuurin yläosassa on reunapalkki, on haka tällöin umpinainen. ”

Kun tukimuurissa ei ole reunapalkkia on haka u-mallinen. Raudan geometria luodaan käyttäen poikkileikkauksen pisteitä 1 ja 2 (kuva 5). Näiden pisteiden lisäksi tarvitaan vielä kaksi muuta pistettä. Nämä pisteet luodaan siirtämällä pisteitä yksi ja kaksi halutun verran alaspäin. Tämä matka on myös haan jalan pituus.

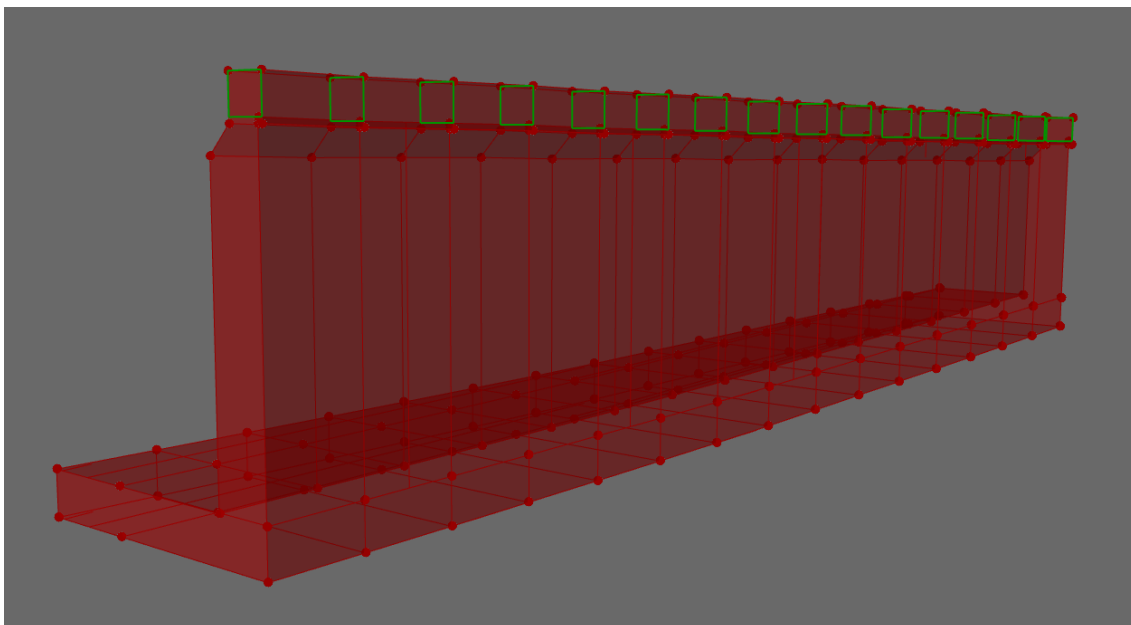


KUVA 5. Grasshopperissa luotu u-haan geometria Rhinossa.

Reunapalkillisessa tukimuurissa on umpihaka. Haan koko geometria voidaan luoda Rhinolla ja Grasshopperilla, mutta Teklan ”ominaisuuksien” takia Rhinon puolella luodaan vain umpinainen neliö ilman limityksiä tai jalkoja. Syy tähän on se, että Tekla ei osaa määrittää jalallisen umpihaan suojabetonipaksuutta. Tämän takia jalkojen pituus ja limitys tapahtuu Teklan omalla komponentilla myöhemmässä vaiheessa.

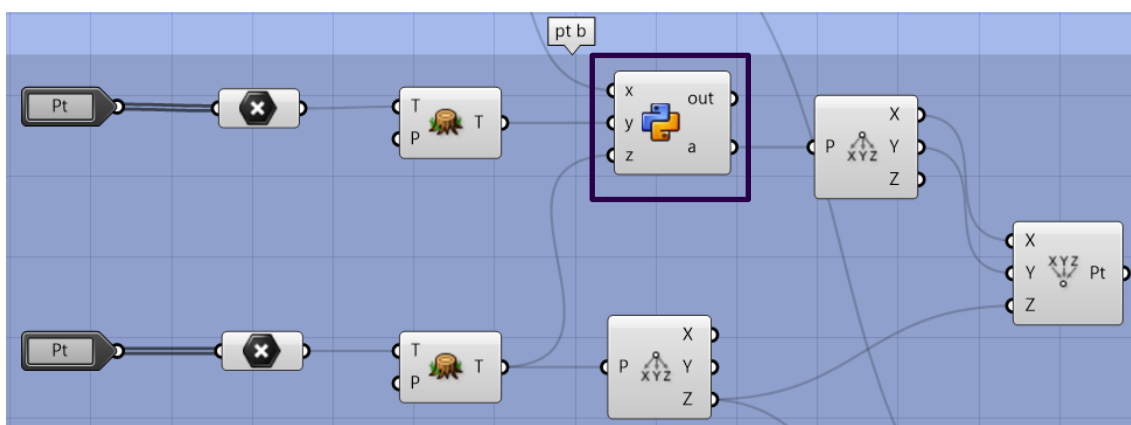
Umpihaan geometrian luomiseen käytetään poikkileikkauksen pisteitä: 1, 2, 3, 4 ja 15 (kuva 2). Pisteitä on tämän jälkeen siirretty ja muutettu riippuen reunapalkin muodosta siten, että lopullisen haan geometrian muodostaa neljä pistettä (kuva 6). Nämä pisteet on nimetty aakkosten mukaan. Raudan geometrian luomisessa käytetty a-piste on luotu käyttämällä poikkileikkauksen 2. pisteen x- ja y-koordinaattia sekä 1. pisteen z-koordinaattia. B-piste on täysin sama kuin poikkileikkauksen 2. piste. C-piste on 1. pisteen x- ja y-koordinaatti sekä 4. pisteen z-koordinaatti.

D-piste koostuu poikkileikkauksen 3. pisteen x- ja y-koordinaattista sekä 4. pisteen z-koordinaatista.



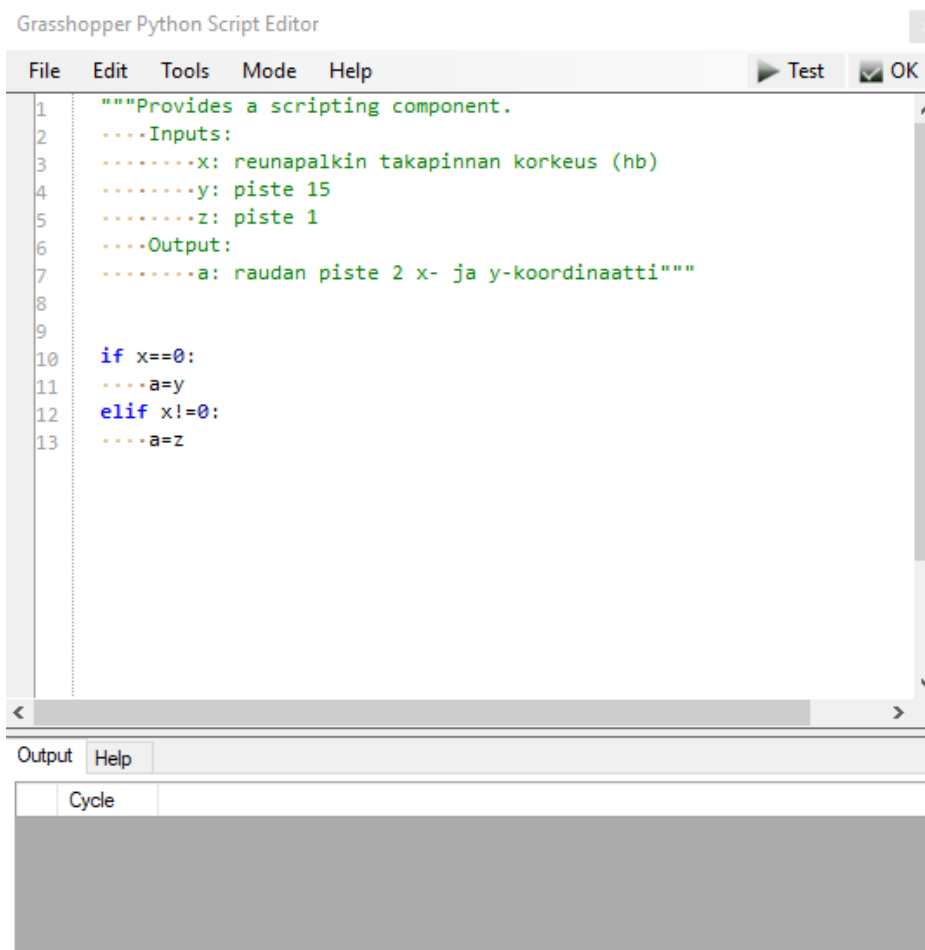
KUVA 6. Grasshopperissa luotu umpihaan geometria Rhinossa.

Jotta raudoituksen geometria on oikeanlainen, on algoritmissa käytetty python-komponenttia (kuva 7). Tämän komponentin sisällä voidaan käyttäen Python-ohjelmointikieltä. Erityisesti jos-lause on hyödyllinen, kun tietyt ehdot johtavat tiettyyn ratkaisuun. Jos-lauseen käyttöön on Grasshopperissa muitakin komponentteja, ja käyttäjä voi valita omaan käyttöönsä sopivimman.



KUVA 7. Umpihaan b-pisteen muodostaminen ja Python-komponentti.

Reunapalkin raudoitukseen geometrian muodostamiseen tehty python-komponentti, on yksinkertainen esimerkki jos-lauseesta (kuva 8). Komponentin input parametreista x-muuttuja on reunapalkin takapinnan korkeus. Kaksi muuta ovat poikkileikkauksen pisteitä, y-muuttuja on piste 15 ja z-muuttuja on piste 1 (kuva 2).



```

Grasshopper Python Script Editor
File Edit Tools Mode Help Test OK
1 """Provides a scripting component.
2   ....Inputs:
3   .....x: reunapalkin takapinnan korkeus (hb)
4   .....y: piste 15
5   .....z: piste 1
6   ....Output:
7   .....a: raudan piste 2 x- ja y-koordinaatti"""
8
9
10  if x==0:
11     ....a=y
12  elif x!=0:
13     ....a=z

```

Output Help

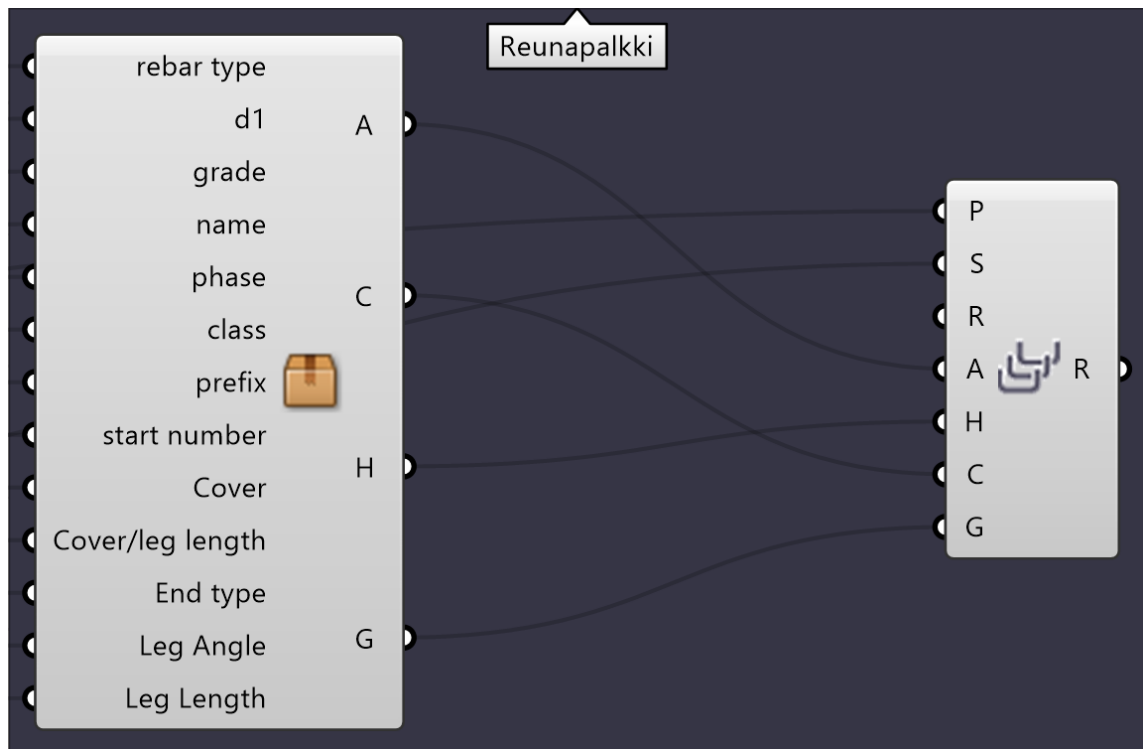
| Cycle |
|-------|
| |

KUVA 8. Python-komponentin sisältö.

Output parametri on a-muuttuja ja siihen syötetään y- tai z-muuttuja. Muuttujien määrää pystyy muuttamaan ja nimeämään sekä input että output puolella. Mikäli ehto toteutuu eli reunapalkin takapinnan korkeus on 0 syötetään a-muuttujaan piste 15 eli y-muuttuja. Jos reunapalkin takapinnan korkeus on erisuuri kuin 0, syötetään a-muuttujaan piste 1 eli z-muuttuja. eli jompikumpi poikkileikkauksen pisteistä

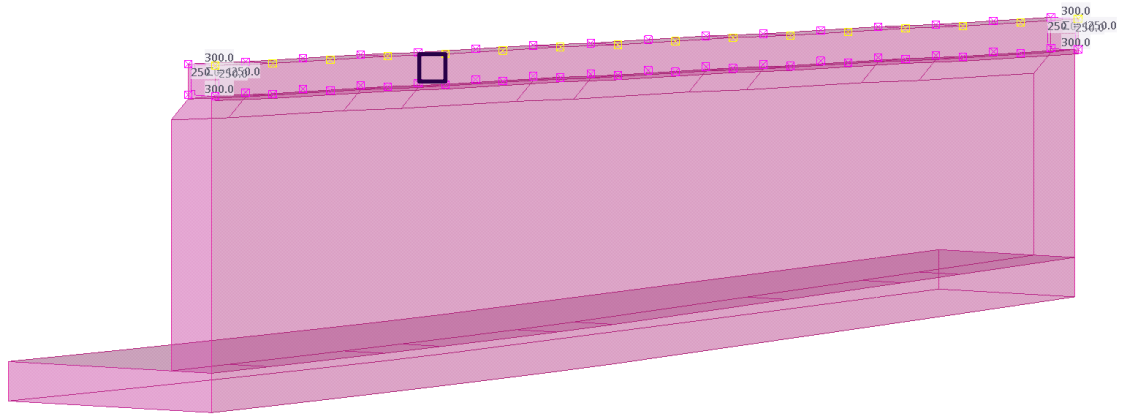
Raudoituksen geometrian valintaan käytetään Python-komponenttia. Mikäli jokin reunapalkin mitoista on 0, ei reunapalkkia mallinneta ja tällöin raudoituksen geo-

metriana käytetään u-hakaa. Jos taas jokainen reunapalkin muodostamiseen käytetty muuttuja on erisuuri kuin 0, reunapalkki mallinnetaan ja haan geometriana käytetään umpihakaa. Kuvassa 9. on Tekla Live Link -komponentti, joka luo tässä tapauksessa haan-raudoitusryhmän. Raudoitusryhmälle annetaan halutut arvot käyttäen clusterin sisältämiä Tekla Live link -komponentteja kuten kuvassa 4.



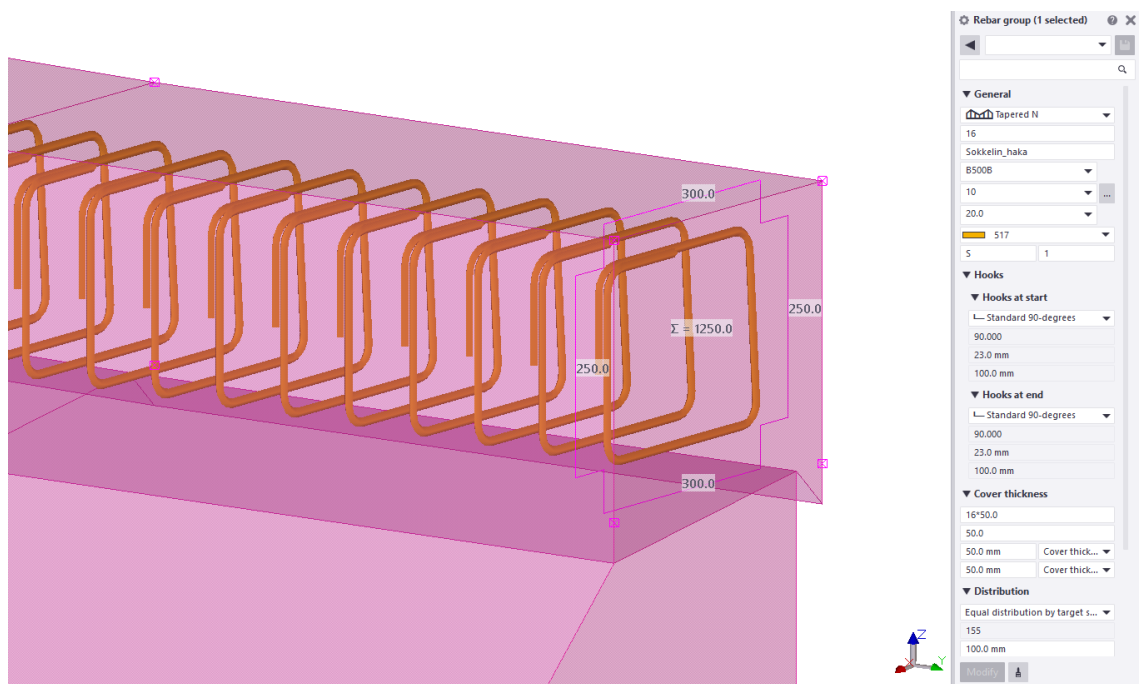
KUVA 9. Haan mallintaminen Teklaan.

Kun vertaillaan kuvia 6 ja 10, huomataan että Rhinossa olevat geometriat ovat Teklassa raudoituksen nurkkapisteitä. Jokaisen raudoituspinnan raudan geometriat on toteutettu melkein samalla tavalla. Hyödynnetty poikkileikkauksen mallintamiseen tehtyjä pisteitä ja tämän jälkeen siirretty niitä raudoituksen vaatimiin paikkoihin ja yhdistämällä ne viivoilla, jolloin tuloksena saadaan raudan vaatima geometria.



KUVA 10. Umpihaan raudoitusryhmän nurkkapisteet Teklassa.

Kuten alussa kerrottiin muodostetaan haan vaatima jalkojen limitys Tekla Live Link -komponentilla. Limitys tapahtuu hooks-komponentilla ja sen avulla saadaan toteutettua reunapalkin vaatima umpihaka (kuva 11). Kuvassa 11 näkyy myös raudoitusryhmälle syötetyt ominaisuudet Teklassa.

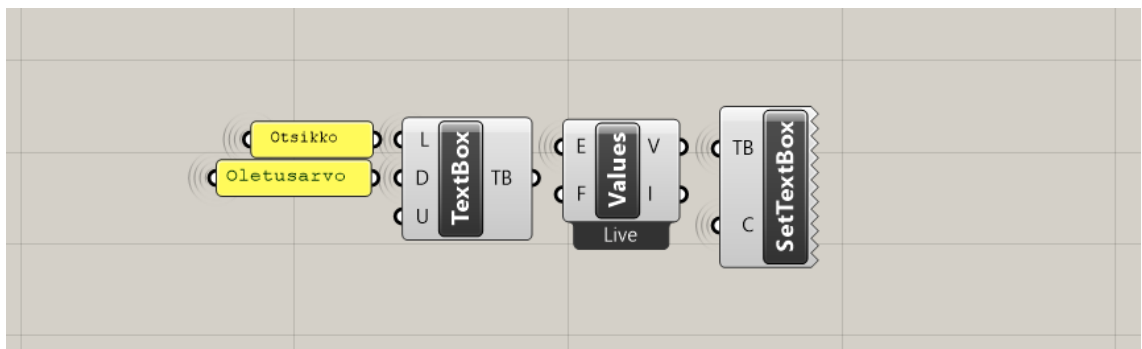


KUVA 11. Valmis umpihaka Teklassa.

3.5 Käyttöliittymä

Kuten kohdassa 2.5 on kerrottu, on käytettävyys toteutettu käyttäen Human UI lisäosaa Grasshopperissa. Lisäosan avulla tehdyllä käyttöliittymällä Grasshopperin käyttö on sujuvampaa ja helpompaa. Sen käytön hyviä puolia on myös se, että se ei ole riippuvainen Grasshopperin toiminnasta. Algoritmi toteutetaan samalla tavalla Grasshopperissa kuten kaikki muukin. Jotta kaikki tarvittavat komponentit saadaan Grasshopperista sisällytettyä myös käyttöliittymään vaatii se huomattavaa työmäärää. Algoritmista on monia lähtötietoja, joita pitää pystyä muuttamaan myös käyttöliittymästä. Jotta tieto käyttöliittymästä päivittyy Grasshopperiin tulee käyttää erillistä komponenttia, joka päivittyy reaaliajassa.

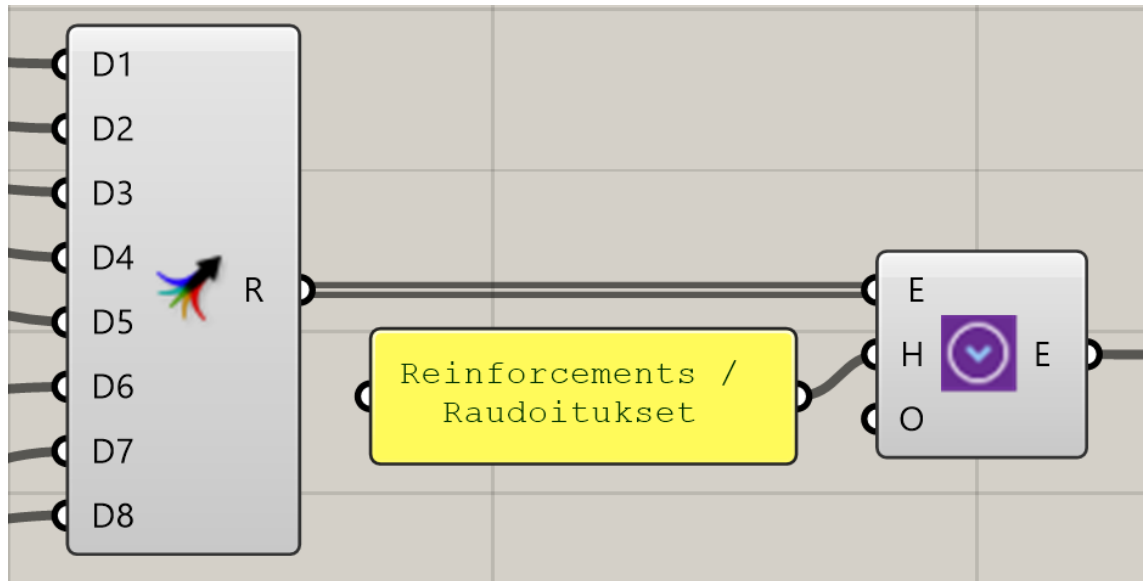
Joidenkin komponenttien muotoa piti muuttaa, jotta niitä olisi helpompi ja sujuvampi käyttää. Esimerkiksi liukusäädin on vaihdettu yksinkertaiseen syöttökenttään (kuva 12). Käyttöliittymää on lähdetty toteuttamaan käyttäjä edellä. Siitä on pyritty tekemään mahdollisimman helppokäyttöinen, jotta virheitä ei tapahtuisi. Mukaan on myös liitetty havainnollistavia kuvia ja ohjeita, jotta käyttäjä ymmärtää, mitä milloinkin tapahtuu.



KUVA 12. Syöttökentän Grasshopperin komponentit.

Käyttöliittymässä on omat välilehtensä kaikille eri asioille. Välilehdillä on saatu ryhmiteltyä samanlaiset asiat samaan paikkaan. Tämän lisäksi on käytetty laajenevia valikoita (kuva 13), jotta yleinen näkymä on siisti. Kuvassa 13 olevat komponentit muodostavat kuvassa 14 olevan alasvetovalikon. Valikon voi avata ja sulkea sinisestä alueesta (kuva 14).

Human UI tuottama käyttöliittymä on raaka ja yksinkertainen, mutta pienellä säätämällä siitä saadaan selkeä ja toimiva kokonaisuus. Esimerkiksi otsikoiden sijaintia muuttamalla käyttämällä sarkainnäppäintä tekstikentässä saatiin hyvältä näyttävä käyttöliittymä.



KUVA 13. Laajenevan valikon luonti Human UI.

Mallinnus käyttöliittymän avulla tapahtuu yksinkertaisesti valitsemalla haluttu painike. Painike aktivoi kyseisen komponentin ja näin mallintaa sen Teklaan. Mallinnus on tarkoituksella jätetty viimeiseksi osaksi käyttöliittymässä, jotta tukimuuria tai rautoja ei turhaan mallinneta Teklaan asti. Ennen mallinnusta, tulee käydä läpi tukimuurin muut osat ja laskelmat, jotta vältytään turhilta Tekla-mallin päivityksiltä.

| KULLMÄTKUMUURI | |
|---|--|
| DIRECTIONS START GEO LOADS GEOMETRY REINFORCEMENT CALCULATIONS TEKLA COST | |
| General information Rebars | |
| Special points of the shear reinforcement / Leikkausraudoituksen erityisyhtymykset | Calculation method / Laskentatapa NCCI 2 |
| Special points of the shear reinforcement / Leikkausraudoituksen erityisyhtymykset | Compression not taken into account / Puristusta ei huomioida |
| Shear reinforcement / Leikkausraudoitus | Calculation with no designated shear reinforcement / Laskenta leikkausraudoittamattomana |
| Conditions of the concrete surface / Betongin pinnan ominaisuudet | Very smooth / Hyvin sileä |
| Anchoring length rounding / Ankkurointipituuden pyöristys [mm] | 10 |
| Auxiliary reinforcement / Tyyteris: d_w [mm] | 10 |
| EC2 The angle of the virtual concrete diagonal / EC2 Betonin virtuaalisen puristusosuuden kulma [deg] | 45 |
| Indentation in the wall surface (eg. decorative surface design) / Muurin pinnan syvennys (esim. koristemuotollusta) [mm] | 0 |

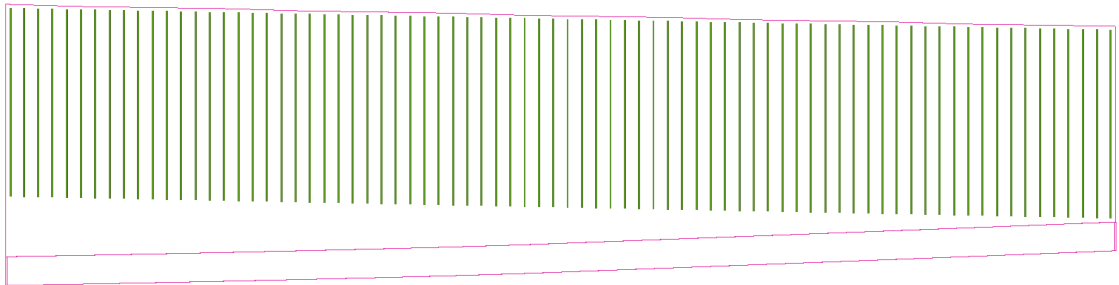
KUVA 14. Raudoituksen yleiset tiedot.

4 RAUDOITUKSEN MALLINTAMINEN

4.1 Raudoituksen mallintaminen Teklassa

Työssä pyritään automatisoimaan raudoituksen mallintamista. Raudoituksen geometria on suhteellisen helppo toteuttaa Teklaan Grasshopperin avulla. Mutta tällä hetkellä työmaan toiminta on vielä täysin manuaalista, jolloin tässä opinnäytetyössä tuli ottaa huomioon myös vahvasti työmaan toiminta. Yksi merkittävä asia on Teklan tuottama raudoitusluettelo. Teklalla voidaan tuottaa raudoitusluettelo, mutta raudoituksen mallinnus vaikuttaa paljon taulukon oikeellisuuteen.

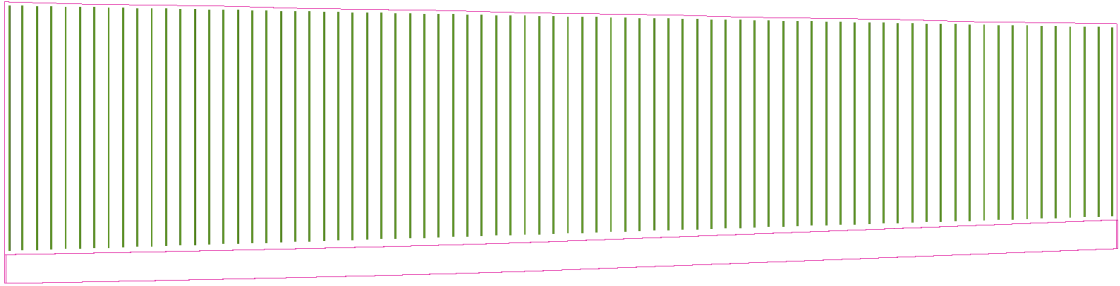
Teklassa on mahdollista mallintaa erilaisia raudoitusryhmiä ja kaksi käytetyintä ovat "suora ryhmä" ja "luseeraava ryhmä". Suora ryhmä (kuva 15) tarkoittaa sitä, että raudan geometrian luomiseen on käytetty yhden raudan päätepisteitä. Eli tällöin jokainen rauta siinä ryhmässä on samanlainen.



KUVA 15. Suora raudoitusryhmä Teklassa.

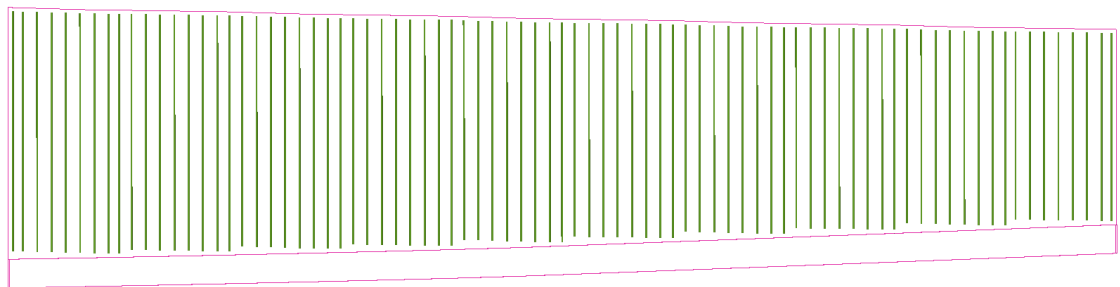
Luseeraava ryhmä (kuva 16) tarkoittaa sitä, että raudan geometrian luomiseen on käytetty ryhmän molemmissa päissä olevien rautojen päätepisteitä. Eli tällöin ryhmän toinen pää on eri mittainen kuin toinen pää ja jokainen rauta on oman pituinen. Luseeraavassa ryhmässä rautojen pituus muuttuu aina lineaarisesti. Luseeraavaa ryhmää pyritään käyttämään vain, jos raudoitusta ei voi toteuttaa usealla suoralla ryhmällä.

Mutta mikäli rakenteen reuna on kaareva, täytyy raudoitus mallintaa useammassa luseeraavassa ryhmässä. Näin ollen jokaisella ryhmällä on oma lineaarinen muutos.



KUVA 16. Luseeraava raudoitusryhmä Teklassa.

Lähes aina pyritään käyttämään suoraa ryhmää. Mikäli rakenteen päiden mitat ovat eri, eli rakenteessa on viisto reuna, mallinnetaan raudoitus useammassa eri pituisessa suorassa ryhmässä (kuva 17). Näin ollen työmaan on helpompi asentaa oikeat raudat oikeaan paikkaan.



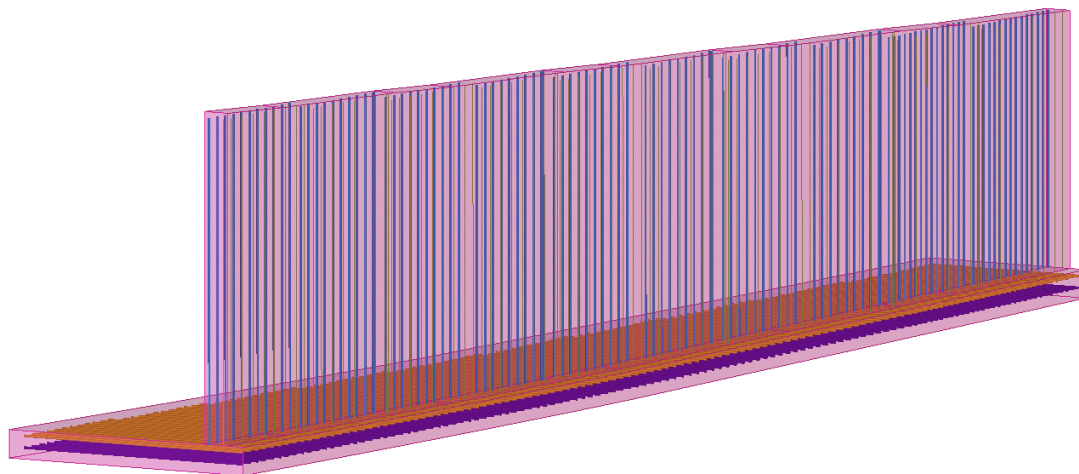
KUVA 17. Useammalla suoralla ryhmällä toteutettu raudoitus.

Kuten kuvista 16 ja 17 huomataan, useammalla suoralla ryhmällä toteutettu raudoitus, ei eroa paljoa luseeraavasta ryhmästä, mikäli rakenteen mitat eivät ole liian erilaiset.

Raudoitusluettelo tuottaa suorasta ryhmästä yhden mitan, joka on tietenkin kaikilla raudoilla sama. Luseeraavasta ryhmistä Tekla tuottaa kolme mittaa raudoitusluettelon. Mitat koostuvat lyhyimmästä ja pisimmästä mitasta, sekä peräkkäisten rautojen pituuserosta. Tämän takia luseeraavaa ryhmää pyritään välttämään, ja käytetään enemmän useampaa suoraa ryhmää. Työmaalla oikean raudan löytäminen on erittäin hankalaa, jos peräkkäisten rautojen pituusero on 5-10 mm.

4.2 Poikittainen raudoitus

Poikittainen raudoitus on tukimuurissa pääraudoitus (kuva 18). Pääraudoitus on rakenteen tärkein raudoitus, koska se on suunniteltu kestävämmään rakenteelle tulevat kuormat.

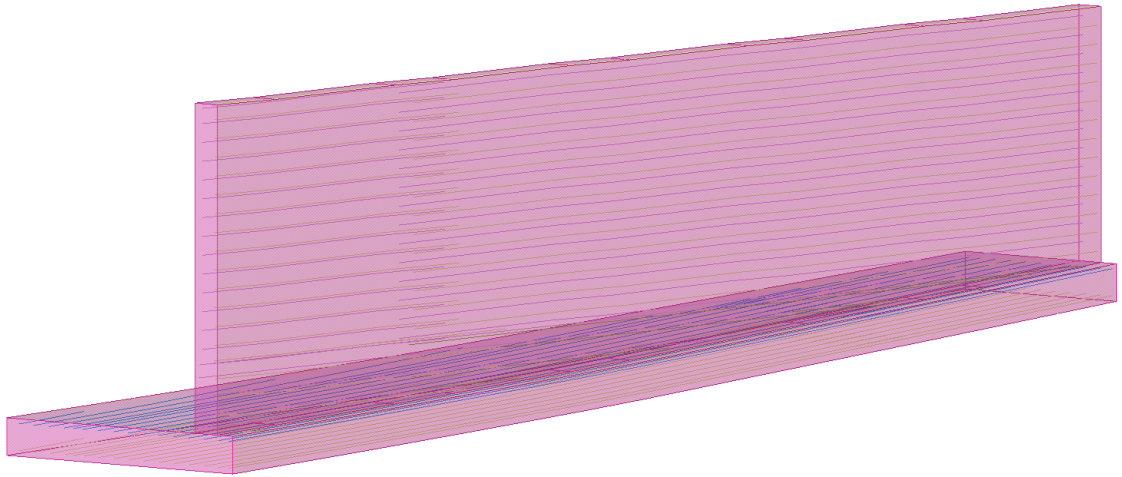


KUVA 18. Tukimuurin pääraudoitus anturassa ja seinässä.

Poikittaisten rautojen geometria tulee suoraan poikkileikkauksen muodosta. Antura tai seinä voi olla vino toisesta reunasta, jolloin raudoitus täytyy ryhmitellä, jotta onnistutaan välttämään luseeraava ryhmä. Poikittainen raudoitus on suunniteltu toimivaksi laskennan kanssa, koska laskennassa on mahdollista valita raudoitusten koko. Raudoituksen koko ja k-jako voi olla samaa koko tukimuurin alueella tai se voidaan jakaa viiteen erityyppiseen rautaan ja k-jakoon.

4.3 Pitkittäinen raudoitus

Pitkittäinen raudoitus on tukimuurissa jakoraudoitus (kuva 19). Jakoraudoituksen tehtävinä on jakaa kuormaa, parantaa rakenteen kestävyyttä ja toimia halkeiluraudoituksena.



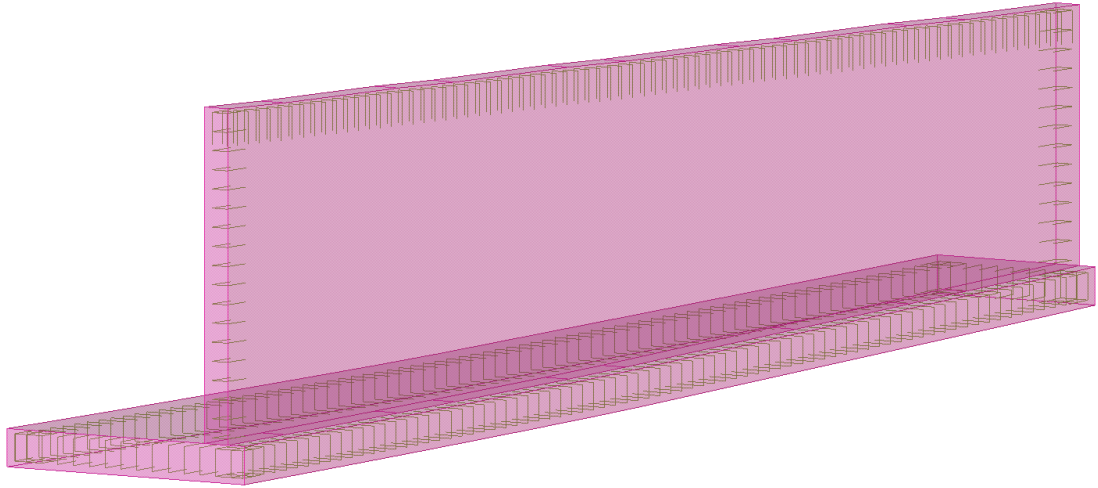
KUVA 19. Tukimuurin jakoraidoitus anturassa ja seinässä.

Tukimuurin monenlaiset poikkileikkaukset aiheuttivat hankaluuksia. Tämän lisäksi tukimuurin pituus on hyvin todennäköisesti suurempi, kuin rautojen maksimipituus. Käyttäjä voi päättää ankkurointipituuden, mutta jatkospituus on toteutettu käyttäen NCCI 2 ankkurointi ja limitysjatkospituuksia. Eli käyttäjä määrittää ankkurointipituuden ja algoritmi kertoo sen kahdella, jolloin tuloksena on NCCI 2 vaatima limityspituus (NCCI 2 2017).

Pitkittäiset raidoitukset koostuvat yhdestä ”pääryhmästä” eli suorasta osuudesta. Sekä tämän lisäksi ne koostuvat ”viisteryhmästä” eli luseeraavasta osuudesta. Mikäli rakenne on suora, ei viisteryhmää muodosteta.

4.4 Hakaraidoitus

Hakaraidoituksen tarkoituksena on tehdä raidoituksesta yhtenäinen. Tukimuurissa hakaraidoitusta käytetään anturan ja seinän reunoissa sekä seinän yläosassa (kuva 20), mikäli tukimuurissa ei ole reunapalkkia. Jos tukimuurissa on reunapalkki, on siinä omanlaisensa hakaraidoitus.



KUVA 20. Hakarautoitus tukimuurissa.

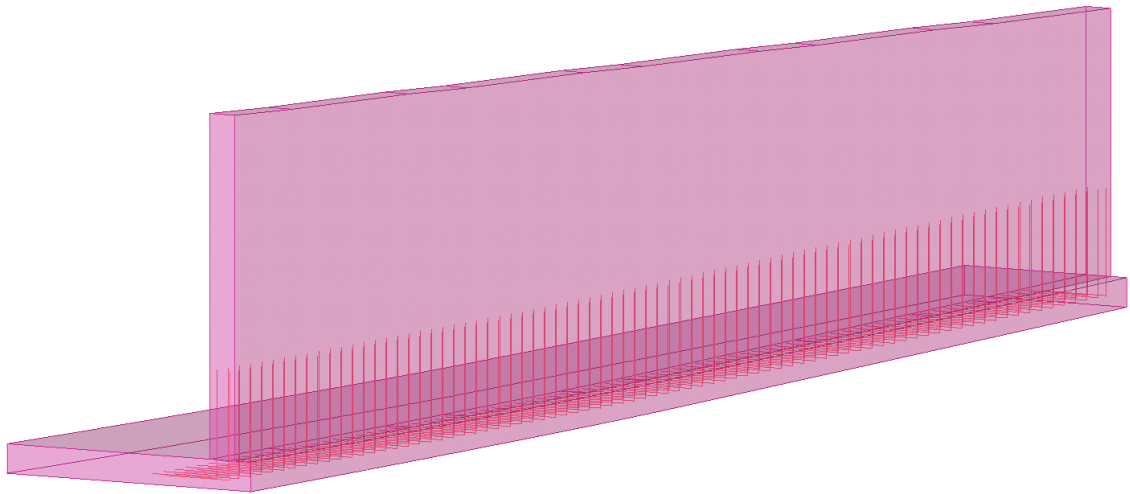
Anturan hakarautoitus on toteutettu käyttäen apuna anturan poikkileikkauksien pisteitä. Pisteet on tämän jälkeen kopioitu ja siirretty käyttäjän määrittämän pituuden päähän. Käyttäjän voi määrittämä pituus on hakaraudan jalanpituus. Tämän jälkeen pisteet on yhdistetty yhdeksi viivaksi, joka on hakaraudan geometrinen muoto. Nämä muodot on syötetty Grasshopperin Teklan komponenttiin. Jokaisella anturan reunalla on oma rautoitusryhmänsä.

Tukimuurin seinämäisessä osassa hakarautoitus on toteutettu samalla tavalla kuin anturassa. Siirretty pisteitä, jotta saadaan muodostettua haan muoto. Myös jalan pituutta on käyttäjän mahdollista muuttaa. Koska seinällä voi olla erittäin monimutkainen poikkileikkaus, on käytetty python-komponenttia, jotta on voitu määritellä mitä geometrioita käytetään missäkin tilanteessa. Yksinkertaisessa tilanteessa on vain yksi ryhmä, mutta maksimissaan algoritmi voi mallintaa viisi ryhmää, mikäli poikkileikkaus on monimutkainen.

4.5 Tartuntarautoitus

Tukimuuuri valetaan yleensä kahdessa osassa; ensin antura ja sen jälkeen seinä. Rautoitus voitaisiin toteuttaa isoilla, pitkillä ja taivutetuilla rautoilla, mutta työ-

maalla sen toteuttaminen on mahdotonta. Näin ollen, jotta raudat olisivat mahdollisimman yksinkertaisia ja järkeviä, käytetään tartuntarautoitusta (kuva 21). Tartuntarautoituksen etuna on helppokäyttöisyys, jolloin seinällä on omat yksinkertaiset rautansa ja anturalla omansa. Tartuntarautoitus luo tartunnan anturan ja seinän rautojen välille siten, että osa raudasta on limittäin seinän rautojen kanssa ja osa taas limittäin anturan rautojen kanssa. Tartuntarautoitus asennetaan anturaan, koska antura valetaan ensimmäisenä betonilla.



KUVA 21. Tartuntarautoitus tukimuurissa.

Tartuntarautoitus on toteutettu algoritmista siten, että on käytetty hyväksi seinän alareunan poikkileikkausten pisteitä. Pisteitä on siirretty sekä ylös- että alaspäin, jotta saadaan haluttu jalan pituus. Käyttäjällä on myös mahdollisuus vaikuttaa jalan suuntaan. Mikäli tukimuurin jalka eli antura on toisella puolella, pystyy käyttäjä vaihtamaan tartuntaraudan jalan suuntaa.

4.6 Detaljit

Rautoituksen suunnittelussa on paljon pientä detaljiikkaa, joka täytyy ottaa huomioon rautoitusta mallinnettaessa. Erityisesti huomiota pitää kiinnittää nurkkiin ja kulmiin. Esimerkiksi tämän opinnäytetyön tukimuurin poikkileikkauksessa voi

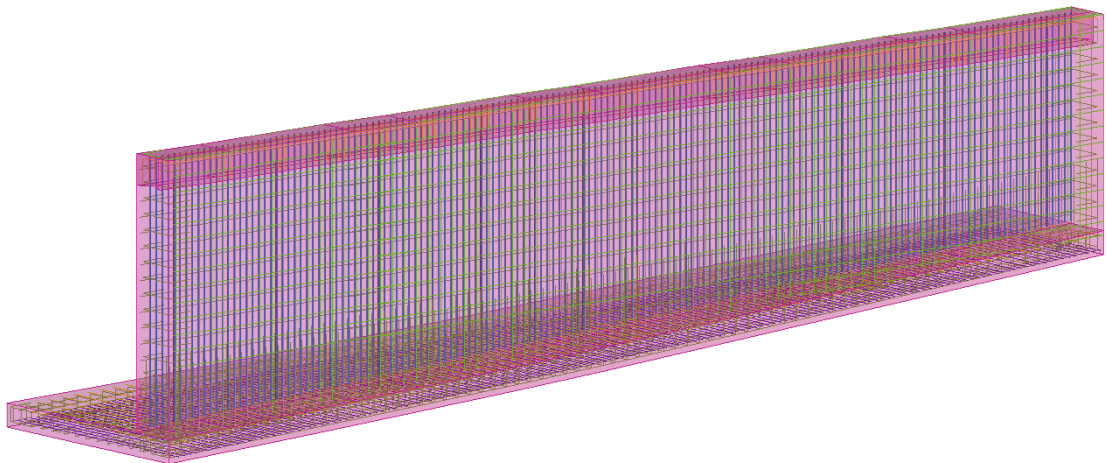
olla kulma puristuspuolella eli tukimuurin etupuolella. Tässä tilanteessa ei yleensä taivuteta rautaa kulmaan, vaan raudat jatkavat sivunsuuntaisina kulman ohi ankkurointipituuden verran.

Yleensä tukimuuri on leikkausraudoittamaton, mutta se voi olla myös leikkausraudoitettu, jolloin myös leikkausraudoitus tulisi mallintaa. Anturassa ja seinässä tulee ottaa huomioon myös sivujen jakoraudoitus. Näiden lisäksi varsinkin silta-rakenteissa raudoituksen yhteyteen asennetaan kontaktitapit eli terästäpit, joiden avulla tehdään sähkökemiallisia mittauksia. Kontaktitapit hitsataan pääraudoitukseen, joka on lähimpänä rakenteen pintaa. Nämä tulisi myös esittää mittapiirustuksissa.

5 TULOSTEN ANALYSOINTI

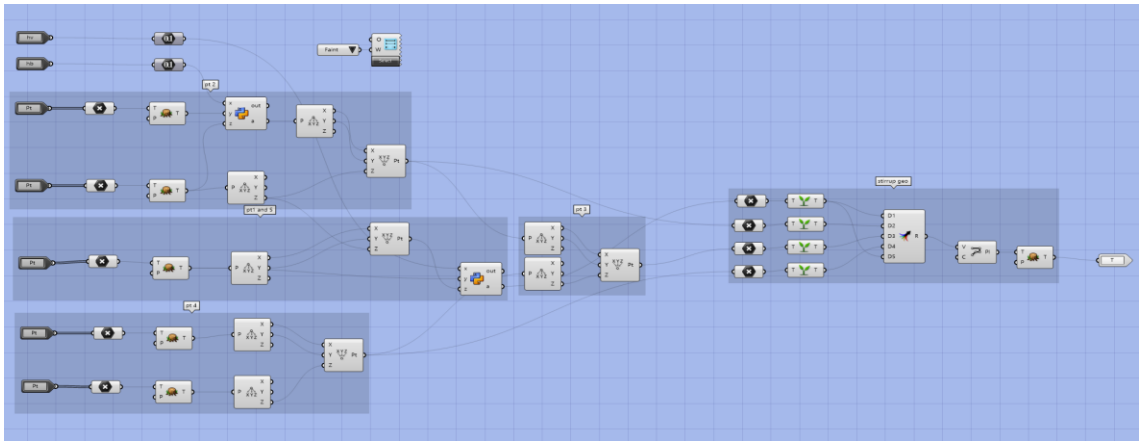
5.1 Raudoitus

Raudoituksen geometrinen mallinnus onnistui pääsääntöisesti hyvin. Algoritmi on pyritty toteuttamaan siten, että sillä pystyy mallintamaan raudoituksen mihin tahansa tukimuurin poikkileikkaukseen, mikä algoritmilla on mahdollista luoda. Ongelmia saattaa kuitenkin vielä ilmetä tietynlaisissa tukimuurin poikkileikkauksen geometrioissa, mutta yleisimmät tapaukset algoritmilla voidaan varmasti toteuttaa. Tukimuurin poikkileikkaus voi vaihdella rajusti ja raudoituksessa on useita eri osa-alueita ja jokainen osa-alue vaatii erilaisia ratkaisuja. Perinteisen tukimuurin raudoitus (kuva 22) kokonaisuudessaan onnistuu hyvin.



KUVA 22. Tukimuurin raudoitus kokonaisuudessaan.

Grasshopperin käyttö helpottui ja lopussa huomasi muutamia asioita, joita olisi tehnyt eri tavalla. Algoritmissa pyrittiin toteuttamaan clustereita (kuva 23) eli komponentteja, jotka koostuvat useasta eri komponentista. Eli clustereiden sisällä on oma algoritmi. Näillä komponenteilla saadaan siistin näköinen algoritmi sekä oikein toteutettuna myös helposti kopioitava algoritminosa.



KUVA 23. Clusterin sisällä oleva algoritmi.

5.2 Käyttöliittymä

Grasshopperilla tehty käyttöliittymä täyttää tavoitteet eli tekee algoritmin käytöstä helpompaa. Käyttöliittymässä on käytetty kolmea erilaista tapaa syöttää tietoa: syöttökenttä, alasvetovalikko tai liukusäädin. Kuvassa 24 on geoteknisten tietojen välilehti. Välilehdellä on monta eri syöttökenttää, johon syötetään kyseisen otsikon tiedot.

| KULMATUKIMUURI | |
|--|------|
| DIRECTIONS START GEO LOADS GEOMETRY REINFORCEMENT CALCULATIONS TEKLA COST | |
| PARTIAL FACTORS AND GEOTECHNICAL PROPERTIES / OSAVARMUUSLUVUT JA GEOTEKNISET ARVOT | |
| γ_{ϕ} | 1.25 |
| γ_c | 1.25 |
| γ_{cu} | 1.5 |
| γ_{qu} | 1.5 |
| γ_{Rv} | 1.55 |
| γ_{Rh} | 1.1 |
| LANDFILL IN FRONT OF THE WALL / MUURIN EDUSTAN TÄYTTO | |
| Effective density / Tehokas tilavuuspaino [kN/m ³] | 25 |
| Angle of repose / Kitkakulma [deg] | 38 |
| Cohesion / Koheesio | 0 |
| LANDFILL IN THE BACK OF THE WALL / MUURIN TAUSTATÄYTTO | |
| Effective density / Tehokas tilavuuspaino [kN/m ³] | 25 |
| Angle of repose / Kitkakulma [deg] | 38 |
| Cohesion / Koheesio | 0 |
| SOIL UNDERNEATH THE WALL / MUURIN ALAPUOLINEN MAA-AINES | |
| Effective density / Tehokas tilavuuspaino [kN/m ³] | 25 |
| Angle of repose / Kitkakulma [deg] | 38 |
| Cohesion / Koheesio | 0 |

KUVA 24. Esimerkki käyttöliittymän geoteknisten tietojen välilehdestä.

Käyttöliittymän käyttöä on helpotettu upotetuilla ohjeilla. Ohjeet on siirretty Grasshopperista ja ne saa esiin painamalla haluttua ohjetta. Selventäviä ohjeita on muun muassa tukimuurin poikkileikkauksen mittoihin ja kuormien sijaintiin, jotka Huuhka on tehnyt omaan diplomityöhönsä. Lisäksi käyttöliittymään tehty ikkuna, jossa näkyy sen hetkinen poikkileikkaus 2D-muodossa. Sen avulla hahmottaa nopeasti, miltä kyseisen tukimuurin poikkileikkaus näyttää. Poikkileikkauksessa ei näy mittoja, jotka parantaisivat sen käyttöä huomattavasti.

Raudoitusvälilehdeltä (kuva 25) voidaan hallita raudoituksen ominaisuuksia. Välilehdellä voidaan valita esimerkiksi raudan halkaisija sekä k-jako. Muita ominaisuuksia, joita voidaan muuttaa ovat Teklassa olevia raudoitusryhmän asetuksia, kuten nimi ja prefix.

KULMATUKIMUURI

DIRECTIONS START GEO LOADS GEOMETRY REINFORCEMENT CALCULATIONS TEKLA COST

General information Rebars

REINFORCEMENT ZONES / RAUDOITUKSEN ALUEET

Five zones / Viisi kenttää

WALL FOOTING SIDE (TENSION SIDE) / SEINÄ TAKAPINTA (VETO)

Name: Sokkelin_sisäpinta_pystyteräs

Phase: 101

Class: 515

Prefix (FI): S

Start number: 1

| Area 1 / Alue 1 (Hole area / Koko alue) | Area 2 / Alue 2 | Area 3 / Alue 3 |
|---|---------------------------|---------------------------|
| Distance / K-jako: 100 | Distance / K-jako: 150 | Distance / K-jako: 150 |
| Diameter / Halkaisija: 25 | Diameter / Halkaisija: 25 | Diameter / Halkaisija: 25 |

| Area 4 / Alue 4 | Area 5 / Alue 5 |
|---------------------------|---------------------------|
| Distance / K-jako: 150 | Distance / K-jako: 150 |
| Diameter / Halkaisija: 25 | Diameter / Halkaisija: 25 |

WALL VISIBLE SIDE (COMPRESSION SIDE) / SEINÄ ETUPINTA (PURISTUS)

TRANSVERSE REINFORCEMENT / POIKITTAINRAUDOITUS

FOOTING BOTTOM SURFACE / ANTURAN ALAPINTA

FOOTING TOP SURFACE / ANTURAN YLÄPINTA

FOOTING TRANSVERSE REINFORCEMENT / ANTURAN POIKITTAINRAUDOITUS

KUVA 25. Raudoituksen hallinnan välilehti.

Jokaisella välilehdellä on otsikkoon liittyvä ominaisuuksia ja niitä voidaan hallita riippuen kohdasta, käyttäen alasvetovalikkoa tai tekstikenttää. Käyttöliittymä on toteutettu siten että se etenee johdonmukaisesti alusta loppuun.

5.3 Haasteet

Suurin haaste oli pitkittäisten raudoitusten vaatimat jatkokset ja luseeraavat ryhmät vinon reunan puolella. Suorat ryhmät ja niiden jatkokset oli suhteellisen helppo toteuttaa, mutta luseeraava ryhmä olikin paljon vaativampi. Se onnistui kyllä toteuttaa, mutta uskon, että ammattilainen olisi kehittänyt toisenlaisen ratkaisun.

Toinen haaste on raudoitusryhmän luomiseen käytettävä komponentti Grasshopperissa. Jotta Teklasta saadaan sen hyödyt irti eli hyödynnettyä sen hyviä dokumentointi työkaluja, tulee raudoitus luoda käyttäen tätä komponenttia. Se vaatii sen, että algoritmissa on tietty määrä näitä komponentteja. Esimerkiksi tässä opinnäytetyössä kehitetyssä algoritmissa on mahdollista olla viisi pitkittäistä raudoitusryhmää. Eli jos tukimuurin pituus on suurempi kuin viiden raudan pituus, miinus jatkospituudet, niin algoritmi ei toimi.

Ensimmäisen käyttökokemuksen jälkeen käyttöliittymän käytössä huomattiin ongelma. Ongelma liittyi Rhinossa olevien geometrioiden kokoon ja sijaintiin. Tukimuurilinjan tulee olla maanpintojen välillä, eikä missään tapauksessa saa ylittää niitä. Korjaaminen oli onneksi helppoa eikä muita ongelmia kokeilussa löytynyt.

5.4 Jatkokehitys

Jatkokehitystä tarvitaan ja algoritmia tulee päivittää, jos sitä halutaan käyttää uudempien Tekla ja Rhino versioiden kanssa. Myös algoritmin siistiminen pitäisi tapahtua ammattilaisen toimesta, jolloin siitä saataisiin tehokas, siisti ja toimiva kokonaisuus. Tulisi myös miettiä onko tämä usealla eri ohjelmalla tapahtuva kokonaisuus oikea vai pitäisikö keskittyä vain yhdellä ohjelmalla tapahtuvaan suunnitteluun varsinkin mallinnuksen ja piirustusten osalta.

Huuhkan tekemä algoritmi pystyy mallintamaan anturan kolmessa osassa eli neljän erilaisen poikkileikkauksen avulla, mutta tässä opinnäytetyössä onnistuttiin mallintamaan raudoitus vain käyttäen kahta poikkileikkausta. Tällöin tukimuurin

antura voi olla eri levyinen anturan päädyissä tai kokonaan saman levyinen. Tämä on yksi erittäin hyvä jatkokehitys kohde tulevaisuutta ajatellen.

Seuraava askel algoritmin osalta olisi automatisoida piirustukset, jolloin koko suunnitteluprosessi olisi automatisoitu.

6 POHDINTA

Usean eri ohjelman käyttö mallintamisessa avaa monia uusia mahdollisuuksia. Tällöin voidaan hyödyntää jokaisen ohjelman parhaita puolia. Kuitenkaan aina usealla ohjelmalla toteutettu mallintaminen ei välttämättä ole järkevää. Erityisesti yksinkertaiset ja pienet kohteet on helppo toteuttaa yhdellä ohjelmalla, kuten Tekla Structures.

Tulevaisuudessa automatisointi ja algoritmiavusteinen suunnittelu tulee yleistymään sen helppouden ja nopeuden vuoksi. Tämä kuitenkin vaati sen, että työpaikoilla on ammattitaitoisia ihmisiä, jotka osaavat käyttää nykyisiä ja tulevia ohjelmia. Tietotekniikka ja laskenta-, sekä mallinnusohjelmat kehittyvät vauhdilla, jolloin on tärkeää pysyä kehityksessä mukana. Esimerkiksi korkeakouluissa tulisi opettaa erilaisten ohjelmien käyttöä nykyistä enemmän, jolloin vastavalmistuneilla ja tulevilla työntekijöillä olisi hyvä pohja alkaa käyttämään työpaikalla käytettyjä ohjelmia.

Yksi merkittävä ongelma automatisoinnissa on työmaan manuaalisuus. Haastavaa tästä tekee sen, että suunnittelussa toteutettu mallinnus tulee pystyä toteuttamaan järkevästi myös reaali maailmassa. Toinen haaste tietomallinnuksen ja työmaan yhteensovittamisen kanssa ovat muutosten paljous. Tietomallin hyviä puolia on muutosten ja revisioiden helppous, mutta jokaisesta muutoksesta tulisi informoida myös työmaata. Ja koska työmaa käyttää piirustuksia tulee jokaisesta muutoksesta päivittää piirustus ja toimittaa se työmaalle. Olisi tärkeää saada myös heidän toimintansa enenevässä määrin mukaan tietomallinnuksen maailmaan. Tällä voitaisiin saavuttaa muun muassa paperiton työmaa eli suunnittelussa tapahtuvat muutokset saataisiin siirrettyä suoraan tietomallilla työmaalle. Tämä vähentäisi virheitä, koska tällöin muutoksia ei tarvitsi päivittää manuaalisesti piirustuksiin.

Algoritmisella suunnittelulla on vielä heikkouksia, mutta tiedon, osaamisen ja ohjelmien kehittyessä, tullaan algoritmisella suunnittelulla nopeuttamaan ja helpottamaan monia eri vaiheita rakennesuunnittelussa. Algoritmit toimivat parhaiten

yksinkertaisille ja toistuville rakenteille, jolloin niistä saadaan eniten hyötyä. Algoritmiaivusteisen suunnittelun saavuttama automatisointi ei kuitenkaan poista manuaalista suunnittelu- ja laskentatyötä, koska edelleen suunnitellaan ja rakennetaan algoritmeille tuntemattomia rakenteita. Nämä rakenteet ovat yleensä yksittäisiä, haastavia ja monimutkaisia suunnitella, jolloin niiden toimintaa ei ole pystytty yleistämään algoritmeihin.

LÄHTEET

Autodesk. 2020. What are the benefits of BIM? Luettu 10.12.2020.
<https://www.autodesk.com/solutions/bim/benefits-of-bim>

ComputationalThinking. nd. What is computational thinking? Luettu 31.03.2021.
<https://www.computationalthinking.org/>

Foof4Rhino. 2020. Human UI. Luettu 10.12.2020.
<https://www.food4rhino.com/app/human-ui>

Grasshopper3d. 2020. Tutorials. Päivitetty? Luettu 10.12.2020.
<https://www.grasshopper3d.com/page/tutorials-1>

Huuhka, P. 2018. Automation Of Construction Planning Of A Reinforced Concrete Retaining Wall. Tampereen teknillinen yliopisto. Diplomi.insinöörin tutkinto-ohjelma. Diplomityö. <https://trepo.tuni.fi/bitstream/handle/123456789/26830/Huuhka.pdf?sequence=4&isAllowed=y>

Räksä, J. 2020. Kaikki keinotekoiset asiat ovat suunniteltuja – Ihmistuntemus ja digipalveluiden käyttöliittymäsuunnittelu. Julkaisija. Luettu 10.12.2020.
<https://www.humak.fi/blogit/kaikki-keinotekoiset-asiat-ovat-suunniteltuja-ihmistuntemus-ja-kayttoliittymasuunnittelu/>

M.A.D. 2020. Rhinoceros. Luettu 10.12.2020.
<https://www.mad.fi/tuotteet/rhinoceros>

Matthew Revell. 2019. Outsystems. Luettu 31.03.2021.
<https://www.outsystems.com/blog/posts/what-is-visual-programming/>

Liikennevirasto. 2017. Eurokoodin soveltamisohje betonirakenteiden suunnittelu – NCCI 2. Luettu 10.12.2020.
https://julkaisut.vayla.fi/pdf8/lo_2017-31_ncci2_web.pdf

Peda. 2020. Mitä on algoritminen ajattelu? Luettu 10.12.2020
<https://peda.net/jyu/it/koulutusteknologia/op/keos-2017/aa/moaa>

Rhino3d. 2021a. Accuracy. Luettu 01.04.2021.
<https://www.rhino3d.com/features/accuracy/>

Rhino3d. 2020b. Features. Luettu 10.12.2020.
<https://www.rhino3d.com/features/>

Rhino3d. 2021c. Nurbs. Luettu 01.04.2021.
<https://www.rhino3d.com/features/nurbs/>

RT 89-11175 Muurit ja tukimuurit. 2015. Rakennustieto Oy. Vaatii käyttöoikeuden. Luettu 10.12.2020.
<https://kortistot-rakennustieto-fi.libproxy.tuni.fi/kortit/RT%2089-11175>

Tekla Structures. 2020. Grasshopper-Tekla Live Link. Päivitetty 02.03.2021. Luettu 10.12.2020. https://teklastructures.support.tekla.com/not-version-specific/en/ext_grasshopperteklalink

The Editors of Encyclopaedia Britannica. 1998. Algorithm. Päivitetty 13.06.2006. Luettu 31.03.2021. <https://www.britannica.com/science/algorithm>

The Editors of Encyclopaedia Britannica. 2009. Reinforced concrete. Päivitetty 07.02.2020 Luettu 10.12.2020. <https://www.britannica.com/technology/reinforced-concrete>

Trimble. nd. Tekla Structures. Luettu 10.12.2020. <https://www.tekla.com/fi/tuotteet/tekla-structures>