

Parametrisk modellering av standardhallar i Grasshopper och Tekla Structures

Antonia Stolpe

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för byggnads- och samhällsteknik

Vasa 2023

EXAMENSARBETE

Författare: Antonia Stolpe
Utbildning och ort: Byggnads- och samhällsteknik, Vasa
Inriktning: Konstruktionsteknik
Handledare: Anders Borg, Novia
Joel Enbacka, Nordec Oy

Titel: Parametrisk modellering av standardhallar i Grasshopper och Tekla Structures

Datum: 16.05.2023

Sidantal: 31

Bilagor: 1

Abstrakt

Detta examensarbete har utförts på beställning av Nordec Oy. Syftet har varit att undersöka möjligheterna med parametrisk modellering vid planering av stålhallar och i detta examensarbete, envåningshallar. Manuell modellering i Tekla Structures kan bli väldigt repetitiv men med en parametriskt modellerad hall kan man eliminera flera tidskrävande moment som ingår i modelleringsprocessen. Målet med examensarbetet har varit att skapa en kod som bildar en parametrisk hall, alltså modellera en hall genom visuell kodning.

För att ta reda på vilka möjligheter den parametriska modelleringen medför har ett exempel på en standardhall modellerats upp med algoritmisk design genom visuell kodning i programmet Grasshopper. Modellen kan tack vare dess parametriska funktioner ändras enligt projektspecifika behov, exempelvis längd, bredd och höjd på hallen. De parametriska funktionerna möjliggörs med de komponenter som Grasshopper har att erbjuda.

I resultatet presenteras den modell som skapats för examensarbetet samt överföringen av modellen till Tekla Structures via en Live Link.

Språk: svenska

Nyckelord: parametrisk modellering, visuell kodning, algoritmisk design, Grasshopper, Rhinoceros 3D, Tekla Structures, BIM

OPINNÄYTETYÖ

Tekijä: Antonia Stolpe
Koulutus ja paikkakunta: Rakennus- ja yhdyskuntatekniikka, Vaasa
Suuntautumisvaihtoehto: Rakennesuunnittelu
Ohjaajat: Anders Borg, Novia
Joel Enbacka, Nordec Oy

Nimike: Standardihallien parametrinen mallinnus Grasshopperissa ja Tekla Structuresissa

Päivämäärä: 16.05.2023

Sivumäärä: 31

Liitteet: 1

Tiivistelmä

Tämä opinnäytetyö on tehty Nordec Oy:n toimeksiannosta. Tavoitteena on ollut tutkia parametrisen mallinnuksen mahdollisuuksia teräshallien suunnittelussa. Manuaalisesta mallintamisesta Tekla Structuresissa voi tulla hyvin toistuvaa, mutta parametrisestä mallinnetulla hallilla voi poistaa useita mallinnusprosessiin kuuluvia aikaa vieviä vaiheita. Opinnäytetyön päämäärä on ollut luoda koodi, joka muodostaa parametrisen hallin, eli mallintaa halli visuaalisella koodauksella.

Parametrisen mallinnuksen mahdollisuuksien selvittämiseksi on Grasshopper-ohjelmassa mallinnettu algoritmisella suunnittelulla esimerkki standardihallista visuaalisella koodauksella. Parametristen toimintojensa ansiosta mallia voidaan muuttaa projektikohtaisten tarpeiden mukaan, esimerkiksi hallin pituuden, leveyden ja korkeuden mukaan. Parametriset toiminnot ovat mahdollisia Grasshopperin tarjoamien komponenttien avulla.

Tuloksessa esitetään opinnäytetyötä varten luotu malli ja mallin siirto Tekla Structuresiin Live Linkin kautta.

Kieli: ruotsi

Avainsanat: parametrinen mallintaminen, visuaalinen koodaus, algoritmisen suunnittelu, Grasshopper, Rhinoceros 3D, Tekla Structures, BIM

BACHELOR'S THESIS

Author: Antonia Stolpe
Degree Programme: Construction and Civil Engineering
Specialization: Structural Engineering
Supervisors: Anders Borg, Novia
Joel Enbacka, Nordec Oy

Title: Parametric Modeling of Single Stories in Grasshopper and Tekla Structures

Date: 16.05.2023

Number of pages: 31

Appendices: 1

Abstract

This Bachelor's thesis has been made on behalf of Nordec Oy. The aim was to investigate the possibilities of parametric modeling when planning single-story buildings. Manual modeling in Tekla Structures can become very repetitive, but with a parametrically modeled building, one can eliminate several time-consuming steps that are part of the modeling process. The goal of this thesis has been to create a code that forms a parametric building, in other words to model a building through visual coding.

In order to find which possibilities parametric modeling entails, an example of a standard, single story building has been made with algorithmic design through visual coding in the program Grasshopper. Thanks to its parametric functions, the model can be altered according to project-specific needs, for example the length, width and height of the building. The parametric functions are made possible with the components that Grasshopper has to offer.

The result presents the model created for this thesis and also the transfer of the model from Grasshopper to Tekla Structures via a Live Link.

Language: Swedish

Key words: parametric modeling, visual coding, algorithmic design, Grasshopper, Rhinoceros 3D, Tekla Structures, BIM

Innehållsförteckning

1	Inledning	1
1.1	Beställare	1
1.2	Bakgrund	1
1.3	Mål och syfte.....	2
1.4	Forskningsfrågor	2
1.5	Avgränsningar	2
2	Parametrisk modellering	3
2.1	Algoritmisk design.....	6
2.2	Design genom visuell kodning	8
3	Program	9
3.1	Rhinoceros 3D 7.0	10
3.2	Grasshopper.....	11
3.2.1	Användarvy.....	13
3.2.2	Komponenter.....	14
3.3	Tekla Structures	19
3.3.1	Live Link	19
4	Modellen.....	20
4.1	Grasshopper-koden	24
4.2	Live Link till Tekla Structures.....	25
5	Resultat.....	26
5.1	För- och nackdelar med parametrisk modellering	26
5.2	Besvarade forskningsfrågor	27
6	Diskussion	28
7	Källförteckning.....	30

Bilaga

Terminologi

BIM

Building information modeling, vilket innefattar skapandet och användandet av en intelligent, digital modell av en konstruktion utifrån vilken man kan planera projekt, samt informera och förmedla projektbeslut. BIM underlättar samarbetet mellan olika planeringsavdelningar såsom byggnadskonstruktion, arkitekt, el och VVS. (Lorek, 2022).

Rhino

Rhinoceros, ett 3D-modelleringsprogram utvecklat av Robert McNeel & Associates. (Rhinoceros 3D b).

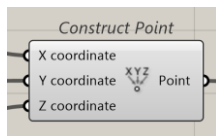
Tekla

Tekla Structures, 3D-modelleringsprogram för detaljering av stål-, trä- och betongkonstruktioner. (Trimble, u.d.).

Plug-in

Ett datorprogram som installeras i ett annat program för att utöka dess funktioner. (Sterne).

Komponent



En komponent är som en byggsten för att bygga upp en helhet. I Grasshopper används komponenter genom att para ihop två eller flera komponenter för att utföra en uppgift eller för att bestämma information. En komponent kan till exempel vara en angiven koordinat, ett siffervärde, en addition av två andra komponenter eller en uppgift för två eller flera andra komponenter.

Auto-CAD

Ett digitalt ritprogram utvecklat av Autodesk. (AutoDesk, u.d.).

CAD

Computer Aided Design, vilket innebär digitalt baserad design och skapandet av tekniska ritningar. (CAD (computer-aided design)).

1 Inledning

I detta examensarbete undersöks möjligheterna med att använda parametrisk modellering vid modellering av stålhallar. Detta innebär i praktiken att en visuell kod, som bygger upp modellen, har tagits fram för att kunna användas och modifieras beroende på projekt och behov. Den visuella koden har skrivits i programmet Grasshopper och sedan tagits in i Tekla Structures via en så kallad Live Link. Då modellen tagits in i Tekla kan vidare detaljarbete fortsätta, men den delen tas inte upp i detta examensarbete.

Före utförandet av examensarbetet har skribenten gått en kurs i programmet Grasshopper. Kursen heter "Learn Grasshopper with Kris" och har dragits av en expert, Krzysztof Wojslaw. Kursen har även fungerat som källa i examensarbetet.

1.1 Beställare

Detta examensarbete har utförts på beställning av Nordec Oy. Nordec är ett finskt stålbyggnadsföretag och är en av de ledande leverantörerna av stomkonstruktioner för byggnader, fasader och stålbroar i Norden.

Nordec har med 40 år i branschen stor erfarenhet av att designa, tillverka och installera krävande konstruktioner. Företaget är uppdelat i olika organisationer vilka specialiserar sig på olika typer av konstruktioner. Dessa är: Single Storey, som specialiserar sig på enplans objekt, Multi Storey, som specialiserar sig på flerplansobjekt och Industry and Bridges, som specialiserar sig på broar och industribyggnader.

Nordec har omkring 600 anställda i sex länder och största delen av företaget finns i de nordiska länderna men företaget har också fabriker i både Polen och Litauen. (Nordec Oy).

1.2 Bakgrund

Beställaren för detta examensarbete är intresserad av att undersöka möjligheterna med den parametriska modelleringen för att se om, och i så fall hur, man kunde implementera den i vardaglig modellering.

I början av en modelleringsprocess behöver man bestämma en halls ytterväggsmått samt var bland annat pelare, balkar, fackverk och stag ska placeras. Allt detta ska sedan för hand

modelleras in i Tekla. Pelare ska prickas ut, var för sig, balkar ska kopplas mellan pelartoppar och så vidare. Men med den parametriska modelleringen kan allt detta snabbas upp. Genom att ange vilka mått på hallen och vilka halldelar, alltså pelare, balkar, fackverk och stag, som ska placeras var, kan användaren men bara några knapptryck modellera en hel hall, förutsatt att en kod redan finns tillgänglig.

1.3 Mål och syfte

Målet med detta arbete var att ta fram en parametrisk modell som designers kan använda i början av ett projekt för att snabba upp processen och eliminera repetitiva moment. Med modellen, som parametriskt kan modifieras med några knapptryck, kan designers snabbt och enkelt ändra höjd, längd och bredd på hallen innan den tas in i Tekla, var designern kan fortsätta modellera.

I detta examensarbete låg fokus på modelleringen i Rhinoceros, med hjälp av Grasshopper.

1.4 Forskningsfrågor

För examensarbetet har följande forskningsfrågor tagits fram och jobbats emot:

- Hur kan modellen underlätta för designers i praktiken och när har man användning för den?
- Vilka andra arbetsmoment kunde snabbas upp med hjälp av parametrisk design?

1.5 Avgränsningar

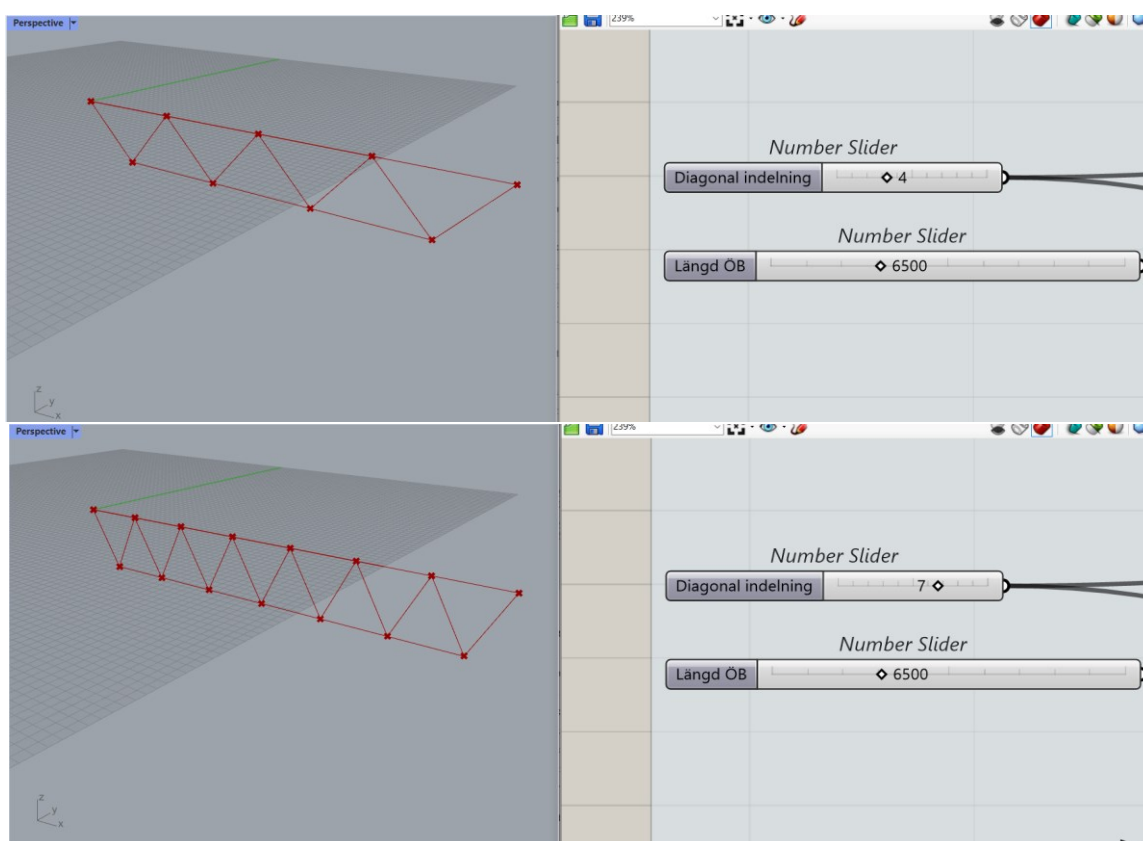
Utöver modelleringen av en standard hall kunde man även undersöka möjligheterna med att använda parametrisk modellering för andra objekt. Objekt som kan vara tidskrävande att modellera för hand. Dock är modelleringen och kodningen för en hall tillräckligt komplex och tidskrävande så all fokus har lagts på det.

Hallen som modellerats i detta examensarbete är en envånings hall, men man kunde även göra flervånings hallar med liknande principer.

2 Parametrisk modellering

Enligt Feng Fu (Fu, 2018) är parametrisk modellering en modelleringsprocess där man har möjligheten att ändra form på en geometrisk modell genom att endast ändra på dess dimensionsvärden. Parametrisk modellering är implementerat genom designprogramkoden, vilken är som ett manus för att definiera dimensionerna för en struktur eller en modell. Genom att ange vissa parametrar och värden, kan man skapa en geometrisk modell som ändrar struktur vartefter man ändrar, tar bort eller lägger till information.

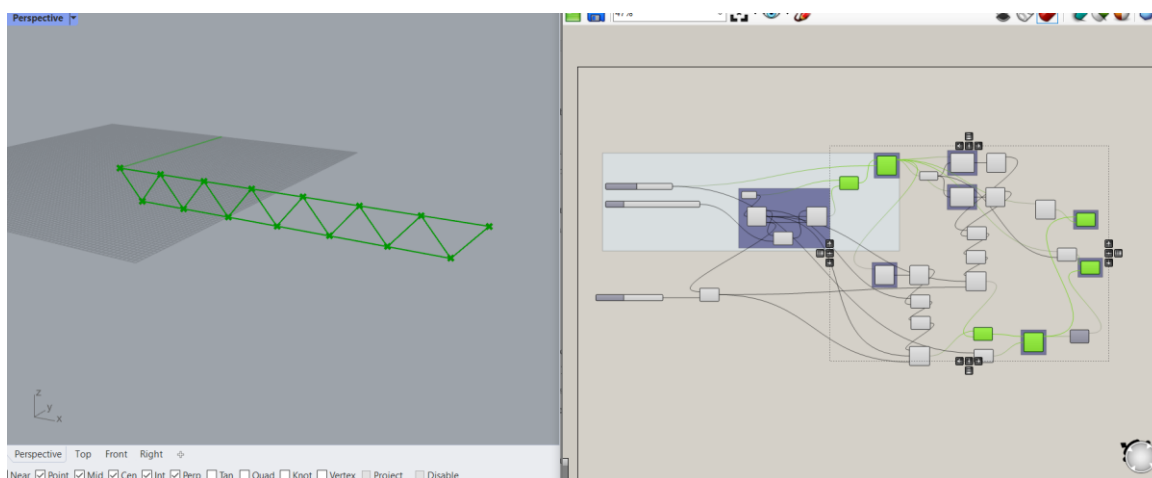
För att visualisera detta kan man i figur 1 se ett fackverk med en övrebom på 6,5 meter. I fackverket på den övre bilden finns det åtta diagonaler medan det i fackverket på den nedre bilden finns 14 diagonaler. Denna ändring är gjord endast genom att ändra värdet för Diagonal indelning, det vill säga hur många indelningar övrebommen har, från fyra mellanrum till sju mellanrum.



Figur 1. Till vänster: 3D modell i Rhinoceros 3D som visar ett enkelt, parallellt, fackverk. Till höger: två sliders ur Grasshopper som anger längd på fackverkets övrebom och indelningen för diagonalerna.

Till skillnad från Feng Fu, beskriver Daniel Davis (Davis, 2013a) en parametrisk modell som *"ett antal ekvationer som uttrycker en geometrisk modell genom tydliga funktioner av ett antal parametrar"*. Vidare beskriver Davis en parametrisk modell som unik. Inte för att den har parametrar, all design, per definition, har parametrar, inte för att den kan ändras, andra designverktyg kan också ändra på en modell och inte heller för att den är ett verktyg. En parametrisk modell är inte unik för vad den gör, snarare för hur den skapades. En parametrisk modell skapas av en designer som uttryckligen anger hur resultatet ska härledas ur de angivna parametrarna.

Genom att använda parametrisk modellering blir designern tvungen att tänka parametriskt. Alltså, förutom att man måste tänka på vad det är man designar, måste man också tänka på den logiska sekvensen av formler, parametrar och de relationer som bildar designen. (01.02. Computational thinking, Learn Grasshopper, 2023). I figur 2 kan man se samma fackverk som tidigare, men nu ser man också hela koden i Grasshopperfönstret och man kan se hur alla delar i koden har kopplingar till minst en annan komponent i koden.



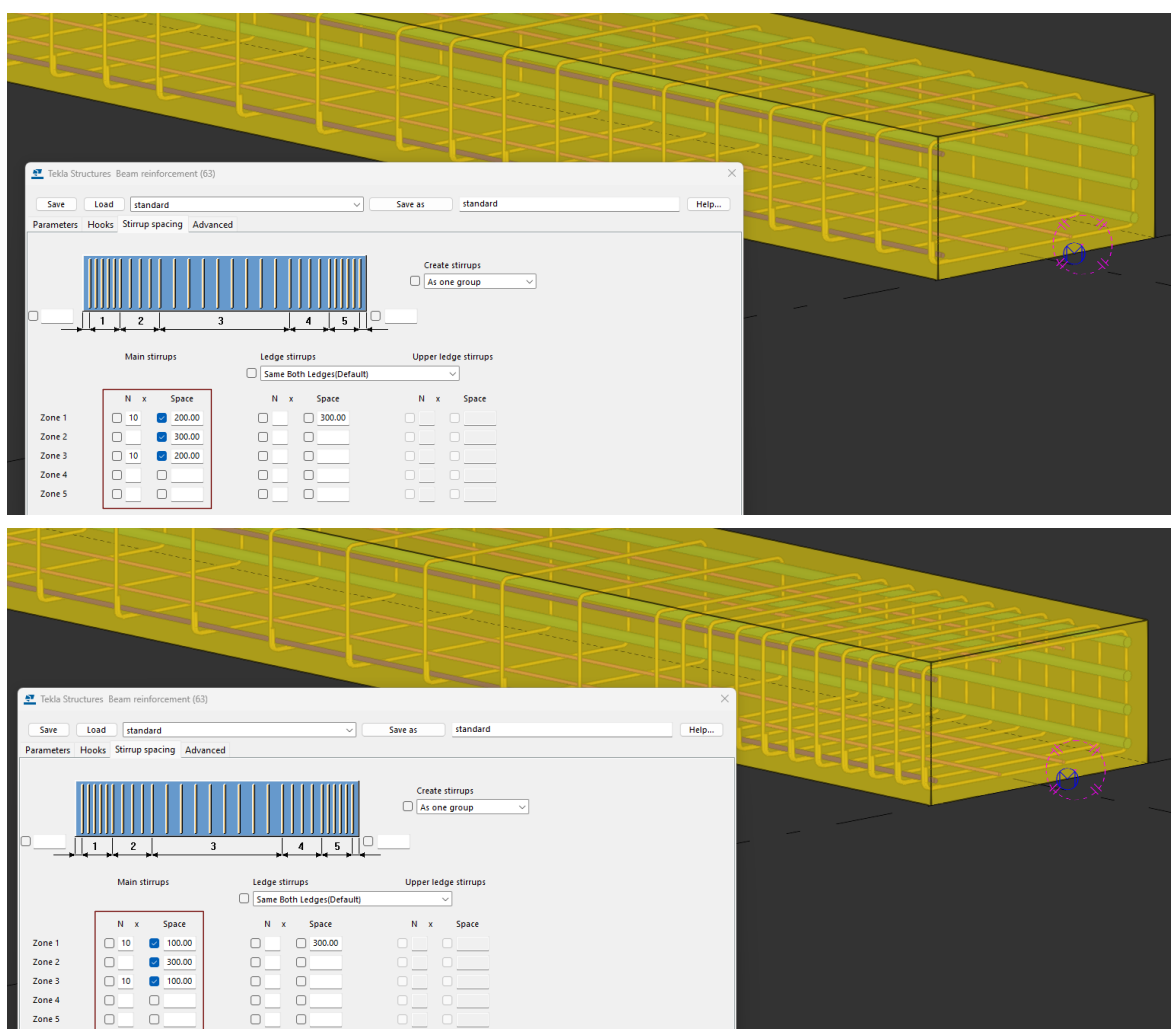
Figur 2. Visuellt exempel på modell och kod av ett fackverk. De markerade komponenterna är de delar som anger modellens geometri.

Den parametriska modellen kan visualiseras i 3D-modelleringsprogram för att efterlikna det verkliga beteendet för den skapade modellen. Detta gör det betydligt enklare att skriva en kod i kodeditorn då man samtidigt kan se modellen bredvid på skärmen. Det är ganska vanligt att en parametrisk modell använder funktionsbaserade modelleringsverktyg för att manipulera modellens attribut, ett sådant exempel är olika sorters komponenter i modelleringsprogrammet Tekla. (Fu, 2018)

Ett känt exempel är armeringsverktyget i Tekla, vilket går att manipulera genom att ändra på dess parametrar i komponentmenyn. Då man ändrar ett eller flera värden för komponenten, ändrar hela armeringshelheten i modellen utan att man manuellt behöver göra något i själva modellen.

Denna möjlighet finns också i andra komponenter i Tekla, till exempel olika anslutningskomponenter var två eller flera plåtar ska anslutas genom svetsning eller med bultar. Genom att ändra parametrarna för anslutningskomponenten, ger den olika resultat i modellen.

I figur 3 kan man se ett exempel på en armeringskomponent i Tekla. Med några ändringar i värdena för komponenten så ändrar dess utseende i modellen.



Figur 3. Armerad balk i Tekla, två olika alternativ på bygelavstånd. Endast för att visualisera ändringen.

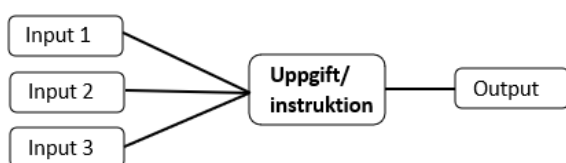
2.1 Algoritmisk design

Algoritmisk design möjliggör för modellering av mycket komplexa objekt som annars skulle medföra vissa utmaningar med manuella designverktyg. Algoritmisk design innebär en parametrisk modelleringsfilosofi, vilket i sig innebär att ett objekts design kan manipuleras genom olika parametrar. Med hjälp av algoritmer kan designers modellera både smidigare och mera tidseffektivt. (Algorithmic Design for Architecture, u.d.).

Algoritmisk- eller beräkningsbaserad design följer en uppsättning instruktioner för att utföra uppgifter, till exempel för att generera en modell av en struktur. Instruktionerna har angivna parametrar, alltså variabler, som kan generera ett antal olika varianter av en och samma kod, men genom att manipulera variablerna blir slutresultatet annorlunda. Algoritmisk design hänvisar till en specifik process för att lösa ett problem. Det blir då designerns uppgift att ta fram de instruktioner och variabler som behövs. På så vis skiljer sig den algoritmiska designen från manuell design, i stället för att manuellt ändra på objekts geometri, ändrar man på dess data och information. (Heiskanen, 2019).

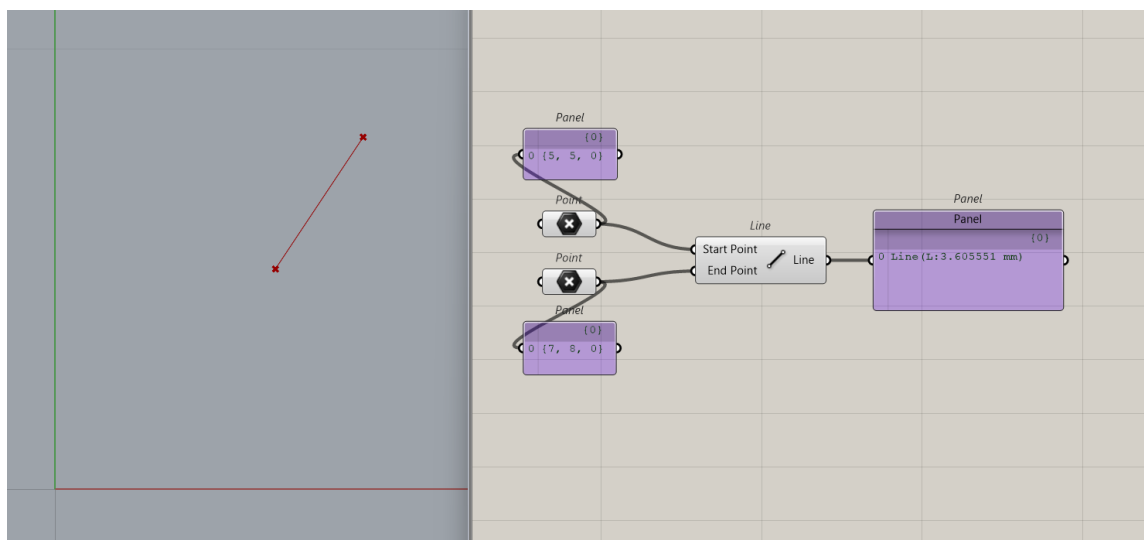
För att jobba med parametrisk modellering genom algoritmisk design behöver man åtminstone två olika datorprogram. Man behöver ett program som fungerar som modelleringsmiljö och ett program där koden skapas, en kooditor. I kooditorn skapar man koden som visas i modelleringsmiljön, alltså, koden visas som en modell eller som någon form av geometri i modelleringsmiljön. Detta kan underlätta då man skapar kod, speciellt ifall man inte har erfarenhet av kodning sen tidigare. I detta examensarbete har man skapat koden endast genom visuell kodning, men den kan också bestå av bara textkod, eller som en kombination av båda. (01.06. What's Rhino and Grasshopper, Learn Grasshopper, 2023).

För att den visuella koden ska fungera krävs det att man har en väldefinierad funktion med inputs och outputs. Inputs är det som funktionen kräver för att utföra en uppgift, alltså en algoritm. (01.02., Learn Grasshopper, 2023).



Figur 4. Enkelt exempel på algoritm.

För att enkelt visualisera en algoritm kan man visa inputs kopplade till en instruktion och ett resultat som beror på inputen och uppgiften, se figur 4. För att tillämpa det här enkla exemplet i praktiken skapas, i figur 5, en linje genom en algoritm. Som input används i exemplet två angivna punkter i koordinatsystemet. Instruktionen, eller uppgiften, kan vara nästan vad som helst, men i exemplet nedan är uppgiften att koden ska skapa en linje mellan de punkter som angetts i inputen. Det som blir output, eller resultatet i detta fall, är en linje som är 3,6 mm på längden.



Figur 5. Exempel på algoritm i Grasshopper (komponenter som skapar en linje mellan två angivna punkter).

En algoritm kan enkelt definieras som en steg-för-steg process eller en formel för att lösa ett problem, eller som en lista på instruktioner formulerad för att göra en specifik uppgift. Ett av de enklaste exemplen på en algoritm är ett vanligt recept, där måste man följa stegen i receptet för att få en färdig produkt. (01.02., Learn Grasshopper, 2023).

Algoritmisk design är speciellt användbart för arkitektoniska former som med manuella designverktyg skulle vara väldigt tidskrävande att designa, eller för objekt var den algoritmiska designen skulle eliminera många repetitiva moment. (Heiskanen, 2019).

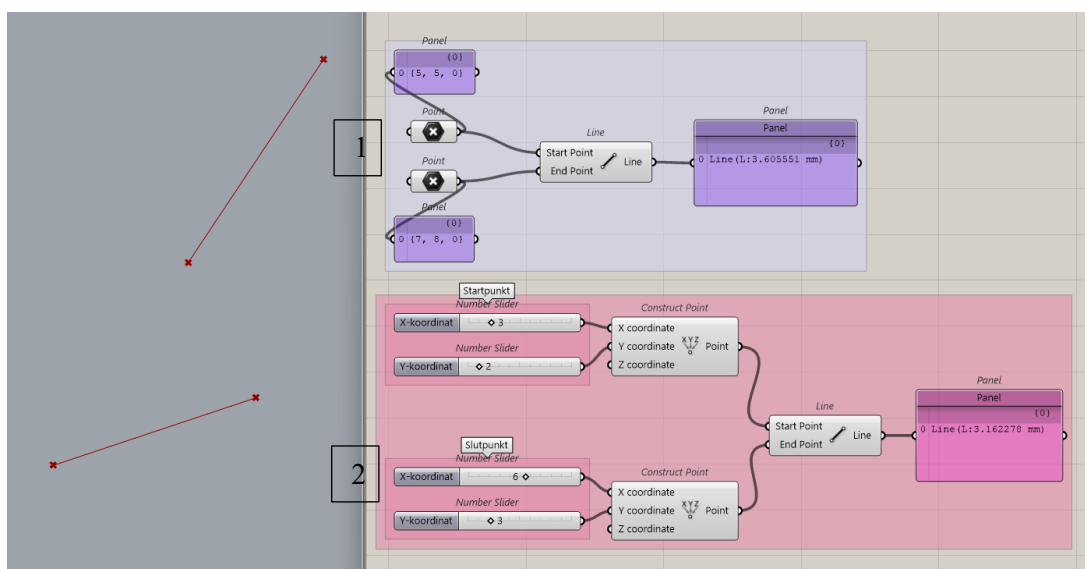
För att modellera en hall krävs inte väldigt svåra modelleringsprocesser, men det blir snabbt många, väldigt repetitiva, moment man behöver göra manuellt. Till exempel placera ut pelare och sedan kopiera ut dem på rätt ställen. Med en parametriskt kodad hall kunde man snabbt ange vilka parametrar man behöver, för att sedan låta koden göra resten. Allt går på bara några knapptryck tack vare algoritmerna man redan angett. I detta

examensarbete definieras hallen genom bredden mellan pelare i x- och y-led (cc-avstånd), antal pelare i x- och y-led och höjden på pelarna.

2.2 Design genom visuell kodning

Visuell kodning, eller visuell programmering, ger användaren möjligheten att programmera utan att behöva lära sig skriva kod. I stället för att skriva en traditionell textkod, kan programmeraren använda sig av komponenter som kodeditorn erbjuder för att bygga upp koden. Då programmeraren kopplar ihop komponenterna bildas nät av information. Den visuella koden kan betraktas som textkodning eftersom samma resultat kan uppnås med båda typer av kod. Visuell kodning ses ofta som ett lättare sätt att arbeta på eftersom koden består av färdiga komponenter i stället för text. (01.03. Programming, Learn Grasshopper, 2023).

Visuell kodning är ett användbart verktyg när man arbetar med algoritmisk design. Man behöver nödvändigtvis inte någon tidigare kunskap om textkodning, men textkodning kan användas i samband med den visuella kodningen, vilket gör algoritmen alltmer effektiv. Med hjälp av visuell kodning kan man modellera en sak på flera olika sätt. Ofta beror metoden man använder sig av på vilken information man har att börja med. Till exempel kan man använda sig av information som i ett tidigare skede varit ett resultat, eller så kan man ange parametrar för hand som sedan används som information i ett följande skede. (01.03., Learn Grasshopper, 2023).



Figur 6 Visualisering av metodval och skiljaktigheter för att skapa en linje mellan två punkter.

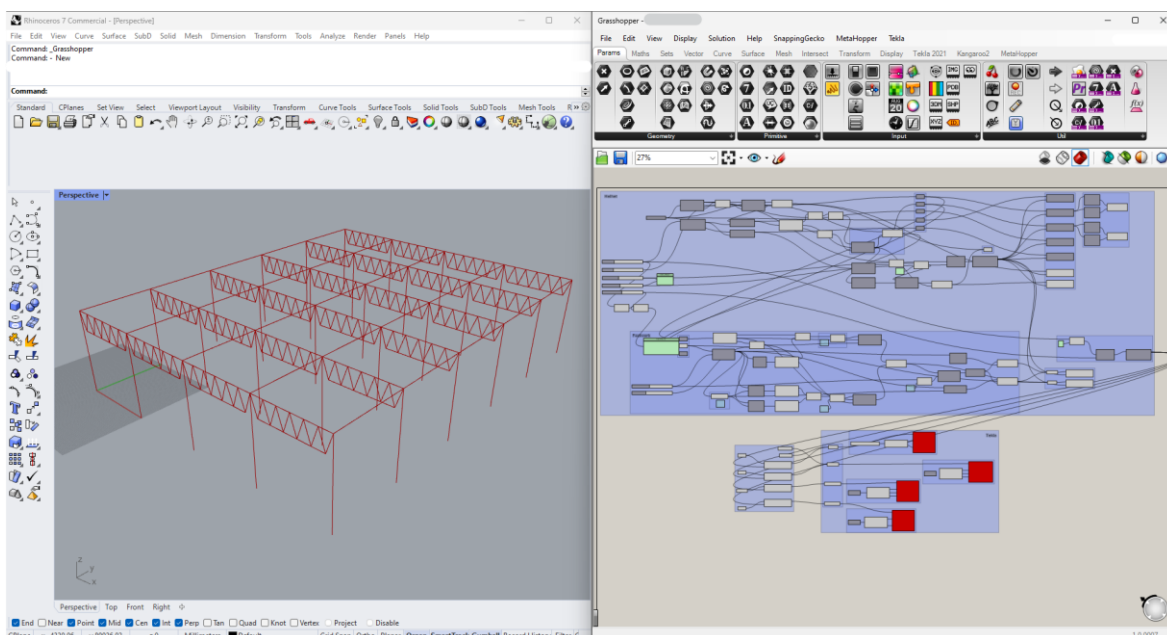
Figur 6 visar två enkla exempel på metodval för att skapa en linje mellan två punkter. Exempel 1 består alltså av två punkter och en linje mellan dem. Punkterna är färdigt angivna i själva Point-komponenten och de angivna koordinaterna visas i textrutorna bredvid dem, komponenten Line skapar en linje mellan punkterna.

Exempel 2 är lite mera avancerat men enklare att modifiera i efterhand. Värdena på start- och slutpunkterna är angivna med så kallade Sliders, en komponent där man enkelt kan ändra ett värde genom att dra pilen längs med komponenten. I detta fall används sliders för att ange x- och y-koordinaterna. Slutligen dras en linje mellan de angivna punkterna med komponenten Line.

I detta examensarbete har visuell kod använts för att skapa en standardhall som kan användas och modifieras enligt behov. Den visuella koden har skrivits i programmet Grasshopper och sedan tagits in för vidare modellering i Tekla Structures.

3 Program

I examensarbetet har huvudsakligen två program använts, Rhinoceros 3D och dess plug-in program, Grasshopper. Modellen som har skapats för Nordec har kodats visuellt i Grasshopper och Rhinoceros 3D. Då man skriver, eller bygger upp, koden i Grasshopper kan man se hur allting börjar ser ut i Rhinoceros fönstret i en stickmodell. Detta visualiseras i figur 7.

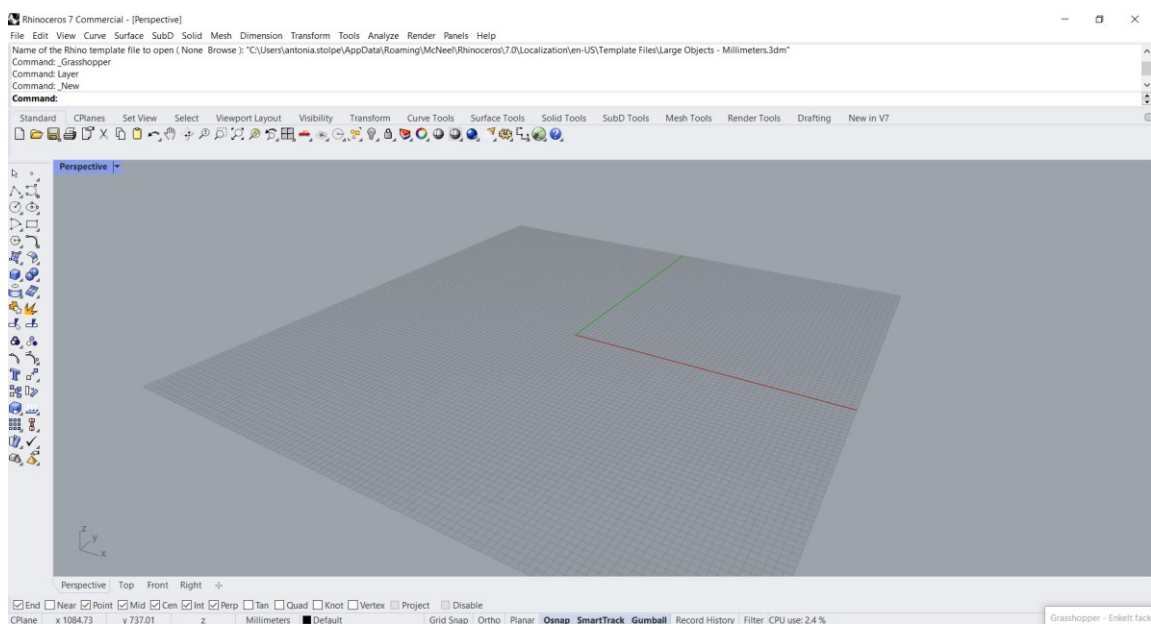


Figur 7. Överblick över modelleringsvy för Grasshopper (höger) tillsammans med Rhinoceros 3D (vänster).

3.1 Rhinoceros 3D 7.0

Programmet Rhinoceros 3D, vanligtvis förkortat Rhino, är ett 3D-applikationsprogram som utvecklades och släpptes år 1998 av Robert McNeel & Associates. Robert McNeel & Associates är ett amerikanskt företag som grundades 1969. Rhinoceros geometri är baserat på den matematiska modellen NURBS. NURBS fokuserar på att producera matematiskt exakt representation av kurvor och friformsytor i datorgrafik. Rhino används för CAD, datorstödd design, CAM, datorstödd tillverkning, 3D-printning och inom industrier såsom arkitektur, industriell design samt grafisk design. (Rhinoceros 3D b).

För att använda programmet behöver man köpa en användarlicens via Rhinoceros hemsida. Man kan också erhålla en gratis testversion som är giltig i 90 dagar ifall man vill testa på programmet. För att använda Grasshopper bör man först ha programmet Rhino installerat och efter det kan man gratis installera Grasshopper. (Rhinoceros 3D a).



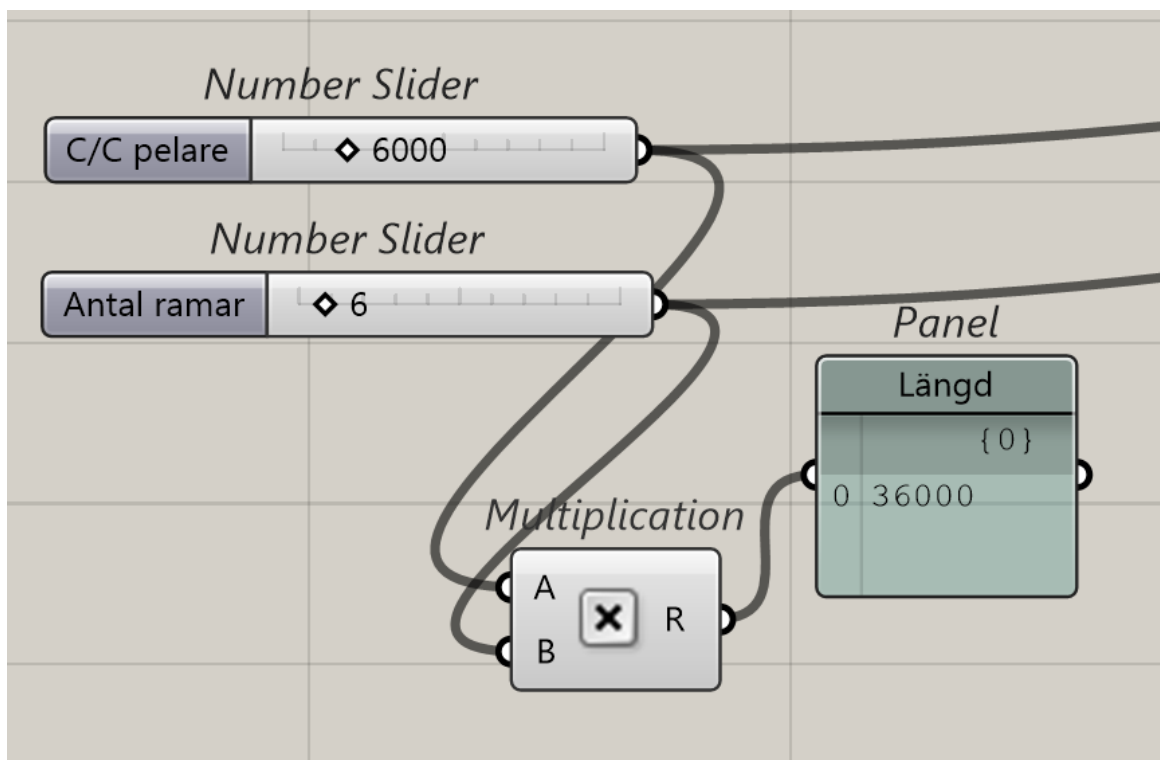
Figur 8. Rhinoceros 7 arbetsvy, 3D-fönster.

Programmet är relativt enkelt att lära sig ifall man har en viss erfarenhet av något annat CAD-program sedan innan. Arbetsvyn, se figur 8, är relativt användarvänlig och man hittar många instruktionsvideor både på Rhinos egna webbsida, och på exempelvis Youtube. I själva arbetsvyn har man fyra olika vyportar att använda sig av, tre av dem är 2D och en är i 3D. I programmet kan man modellera mycket för hand, men i detta examensarbete sker all modellering i Rhino via Grasshopper koden. (02.01. Rhino Interface 1/2, Learn Grasshopper, 2023).

3.2 Grasshopper

Grasshopper är ett plug-in program för modelleringsprogrammet Rhinoceros 3D. Grasshopper är baserat på visuell kodning och är utvecklat av David Rutten med Robert McNeel & Associates. Det ursprungliga namnet på Grasshopper var Explicit History och publicerades 2007 av Rhino 4.0. Men år 2008 ändrades namnet på programmet till Grasshopper. (Grasshopper 3D).

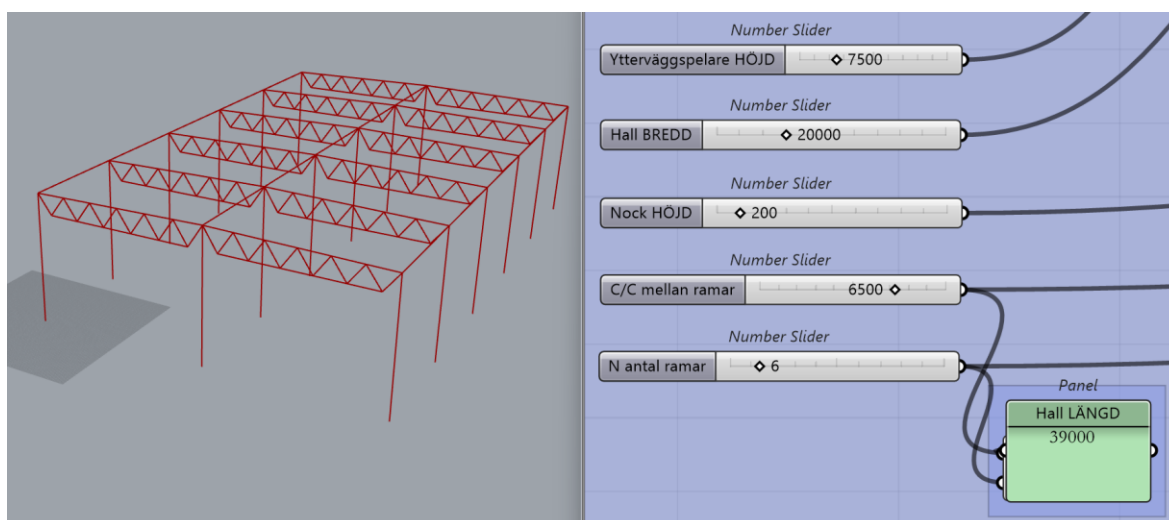
Grasshopper som program kan ta en stund att lära sig. Programmet är fullt av komponenter och andra verktyg som man kan använda sig av. Det är nästintill omöjligt att lära sig om alla komponenter, eller att ha användning av allihop. Den stora mängden möjligheter, tack vare alla verktygen och komponenterna, gör att det inte finns några gränser för vad man kan skapa i programmet. Den information man behöver om programmet och alla dess komponenter hittas på internet, exempelvis via programmets egen webbsida. Med Grasshopper kan användaren skapa en visuell kod genom att koppla ihop tidigare nämnda komponenter med så kallade Wires, på svenska kabel eller linje. Se exempel i figur 9. Den gråa linjen mellan komponenterna är det som kopplar ihop informationen med instruktionen eller någon annan komponent. (03.01. Grasshopper interface 1/3, Learn Grasshopper, 2023).



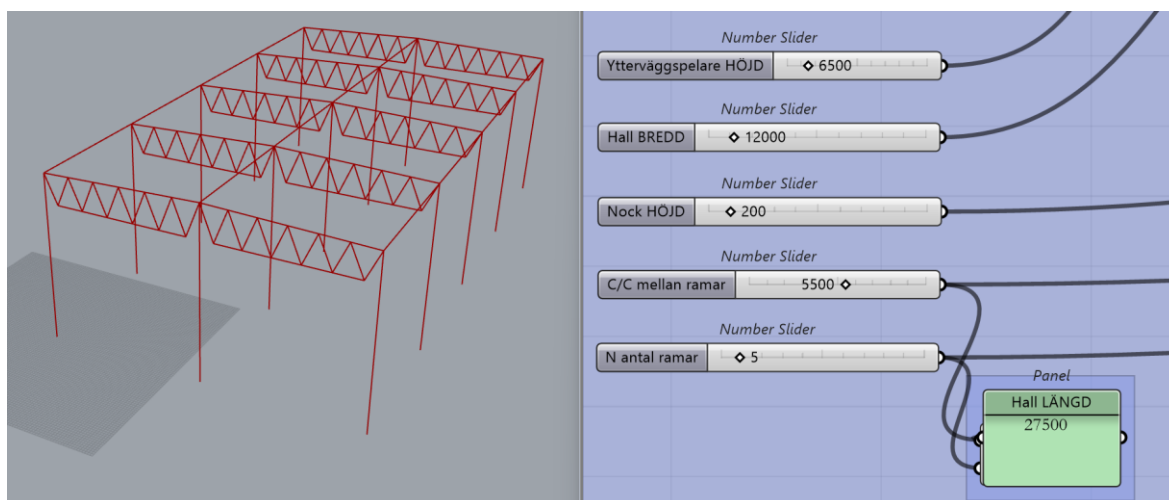
Figur 9. Komponenter kopplade med varandra. Här för att ändra ram-antal och c/c-mått på en hall.

Även om det handlar om kodning i Grasshopper, är koden skapad visuellt och man behöver, som tidigare nämnt, inte ha någon tidigare kunskap om kodning. Med Grasshopper kan man skapa oändligt många lösningar genom att modifiera koden eller dess parametrar. En tacksam funktion med Grasshopper är att man kan uppdatera modellen i realtid i andra program, i detta examensarbete Rhino och Tekla.

För att visualisera hur enkelt man kan ändra på dimensionerna för ett objekt har en modellhall byggts upp i Grasshopper och Rhino (inte slutliga hallen för examensarbetet), se figur 10 och 11. De parametrar som anger modellhallens geometri i Grasshopper, det vill säga längd, bredd och höjd, ses på höger sida och stickmodellen i Rhino ses på vänster sida. Genom att dra eller skriva in nya värden i så kallade nummer sliders kan man modifiera hallen på bara några sekunder.



Figur 10 Exempel A. En hall med bredden 20 meter och längden 39 meter.

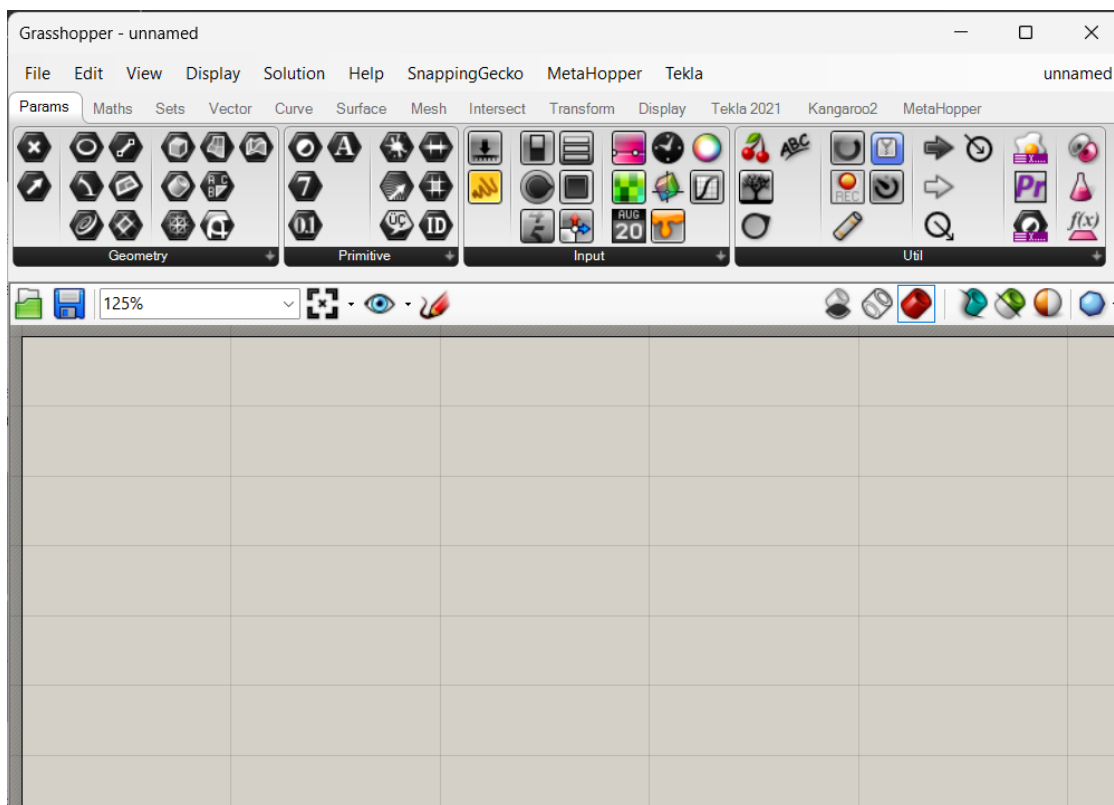


Figur 11 Exempel B. En hall med bredden 12 meter och längden 27,5 meter.

Som man i figurerna ovan kan se, ändrar modellens geometri enligt hur kodens inputs ändras. Här i exempel A är hallen totalt 20 meter bred, 39 meter lång och med en ytterväggshöjd på 7,5 meter. Också höjden på nocken kan ändras för att få rätt lutning på fackverken. I exempel B har man modifierat inputen så att hallen är 12 meter bred, 27,5 meter lång och med en ytterväggshöjd på 6,5 meter. Parametrarna för hallen har alltså ändrats genom att man skrivit in de önskade värdena i input komponenterna.

3.2.1 Användarvy

I Grasshopper jobbar man primärt på en canvas, på svenska en så kallad duk. I programmets övre kant hittar man verktygsfältet med alla komponenterna. Man kan endera dra komponenterna till duken eller dubbelklicka på duken och sedan söka på just den komponent man vill använda. Komponenterna placeras på duken och kan sedan flyttas, organiseras och gömmas enligt behov. För att skapa något kopplas två eller flera komponenter ihop med vajrar, det är alltså den visuella kodningen. I figur 12 kan man se hur programmet ser ut då man öppnar en ny fil. I övre kanten har man kontrollpanelen och nedanför har man den grå duken på vilken koden skrivs. (03.01. Grasshopper Interface 1/3, Learn Grasshopper, 2023).



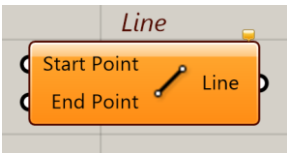
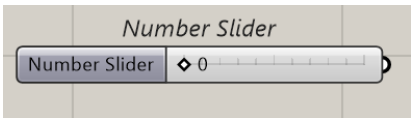
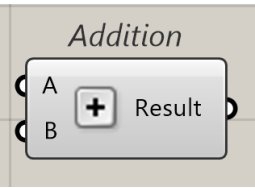
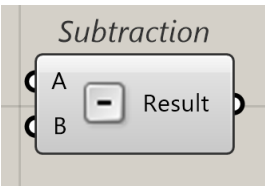
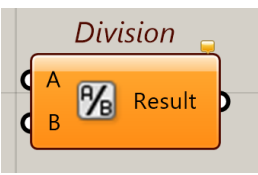


Figur 12. Grasshopper användarvy.

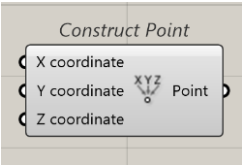
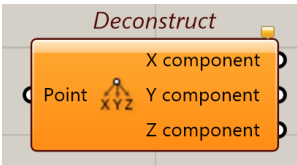
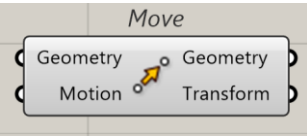
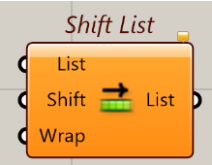
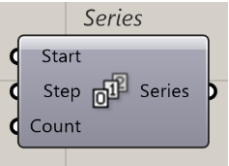
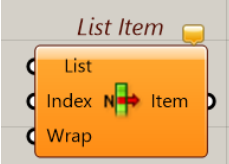
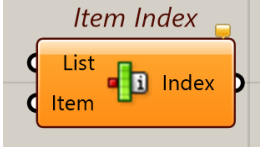

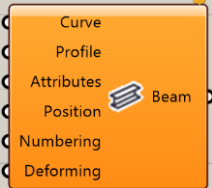
3.2.2 Komponenter

Även om Grasshopper har väldigt många komponenter man kan använda sig av, klarar man sig långt på bara en handfull komponenter. I detta examensarbete är de komponenter som presenteras i tabell 1 de som använts mest för att skriva koden.

Tabell 1. DEL 1: Lista på vanliga komponenter i Grasshopper.

<p>Punkt (Point)</p>		<p>En fast punkt vars koordinater endera kan anges i komponenten eller tas ur en annan komponent. (Container komponent).</p>
<p>Kurva (Curve)</p>		<p>En geometri, kan vara en linje, en kurva eller dylikt. (Container komponent).</p>
<p>Linje (Line)</p>		<p>En linje som skapas mellan två punkter. Kan användas för att visualisera exempelvis pelare, balkar och fackverksdelar. (Allmän komponent).</p>
<p>Nummer slider (Number slider)</p>		<p>Ett reglage som anger ett numeriskt värde. Man kan ändra värdet genom att flytta pilen eller skriva in värdet. (Input komponent).</p>
<p>Addition</p>		<p>Matematisk addition av två värden. (Allmän komponent).</p>
<p>Subtraktion (Subtraction)</p>		<p>Matematisk subtraktion av två värden. (Allmän komponent).</p>
<p>Division</p>		<p>Matematisk division av två värden. (Allmän komponent).</p>

Tabell 1. DEL 2: Lista på vanliga komponenter i Grasshopper.

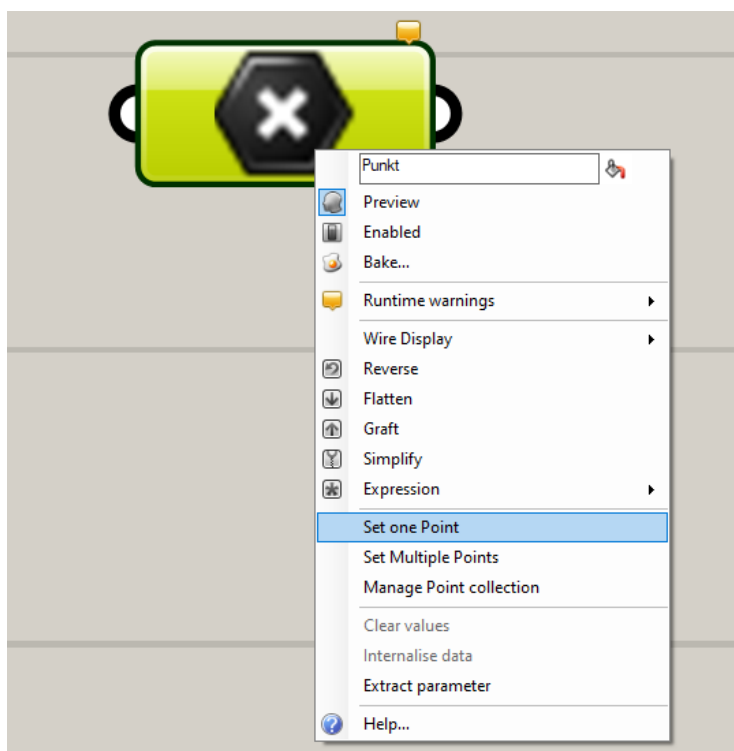
<p>Konstruera punkt <i>(Construct point)</i></p>		<p>Konstruerar en punkt utifrån angivna x-, y- och z-koordinater. <i>(Container komponent).</i></p>
<p>Dekonstruera punkt <i>(Deconstruct point)</i></p>		<p>Dekonstruerar en punkt till dess x-, y- och z-koordinater. <i>(Container komponent).</i></p>
<p>Flytta <i>(Move)</i></p>		<p>Kopierar ett objekt längs en vektor. <i>(Allmän komponent).</i></p>
<p>Förflytta lista <i>(Shift list)</i></p>		<p>Förflyttar objekt i en lista enligt givna steg (shift). <i>(Allmän komponent).</i></p>
<p>Serie <i>(Series)</i></p>		<p>Bildar en serie av siffror/data. <i>(Allmän komponent).</i></p>
<p>Lista objekt <i>(List item)</i></p>		<p>Plockar ut specifik data ur en lista. <i>(Allmän komponent).</i></p>
<p>Objekt index <i>(Item index)</i></p>		<p>Plockar ut index för ett specifikt objekt ur en lista. <i>(Allmän komponent).</i></p>
<p>Sortera ut n:de <i>(Cull N:th)</i></p>		<p>Sorterar ut n:de objekten i en lista. <i>(Allmän komponent).</i></p>
<p>Tekla geometri</p>		<p>Skapar objekt i Tekla genom att ta nödvändig data ur koden i Grasshopper. <i>(Allmän komponent).</i></p>

(Grasshopperdocs, 2022).

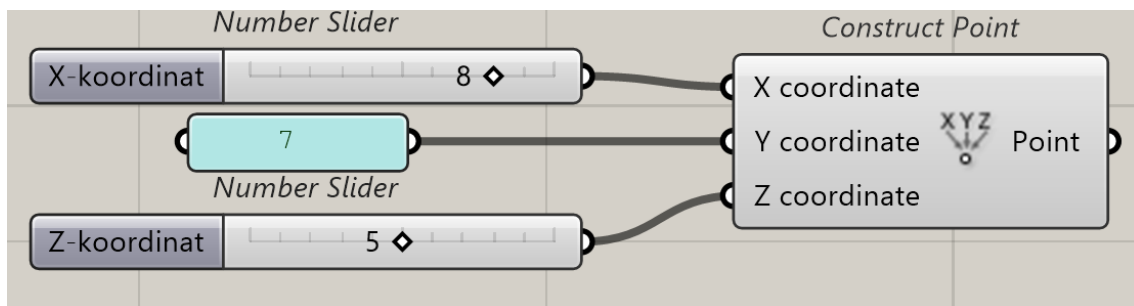
I Grasshopper finns det egentligen tre olika typer av komponenter, men i detta examensarbete benämns alla sorts komponenter endast som komponent.

1. **Allmänna komponenter** som utför en uppgift, till exempel komponenten Line eller någon av de enklare matematiska komponenterna.
2. **Inputs** som exempelvis anger ett värde, till exempel en nummer slider eller annan data såsom en textruta med manuellt inskrivna värden.
3. **Containrar**, vilka kan innehålla nästan vilken data som helst, till exempel specifik geometri, som en punkt eller en linje. Kan också vara större mängd data som grupperats ihop i en Data container.

Det är dock värt att nämna att de komponenter som fungerar som inputs och containrar också kan kallas för parametrar. Det som skiljer en parameterkomponent från en allmän komponent är att en parameter lagrar data, medan en allmän komponent processar datan ur en parameterkomponent. Man kan definiera parametrars data på ett antal olika sätt, till exempel genom att manuellt ange data i komponenten, se figur 13, eller genom att koppla data ur en annan komponent till den komponent vart datan ska definieras, se figur 14. Ett tredje exempel för att ange data, till exempel geometri, är att importera ett objekt från Rhino genom att markera objektet i Rhino och tilldela det som komponentens data i Grasshopper. (03.01-07.02, Learn Grasshopper, 2023).

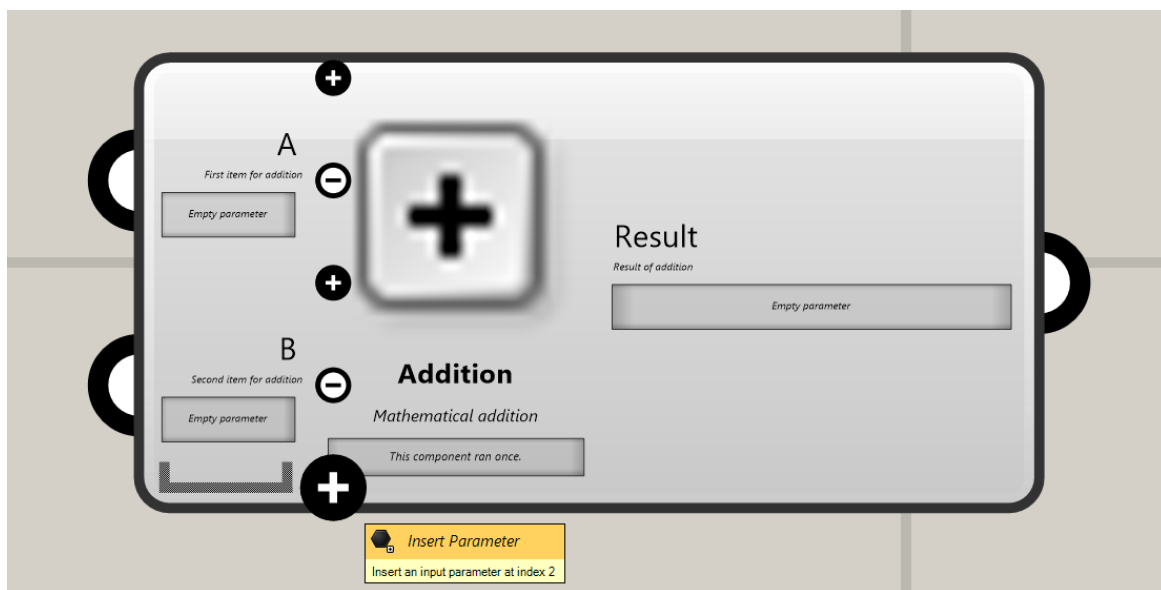


Figur 13. Manuellt angivna koordinater för en punkt i Grasshopper med Set one point.



Figur 14. Adaptiva koordinater för en punkt i Grasshopper.

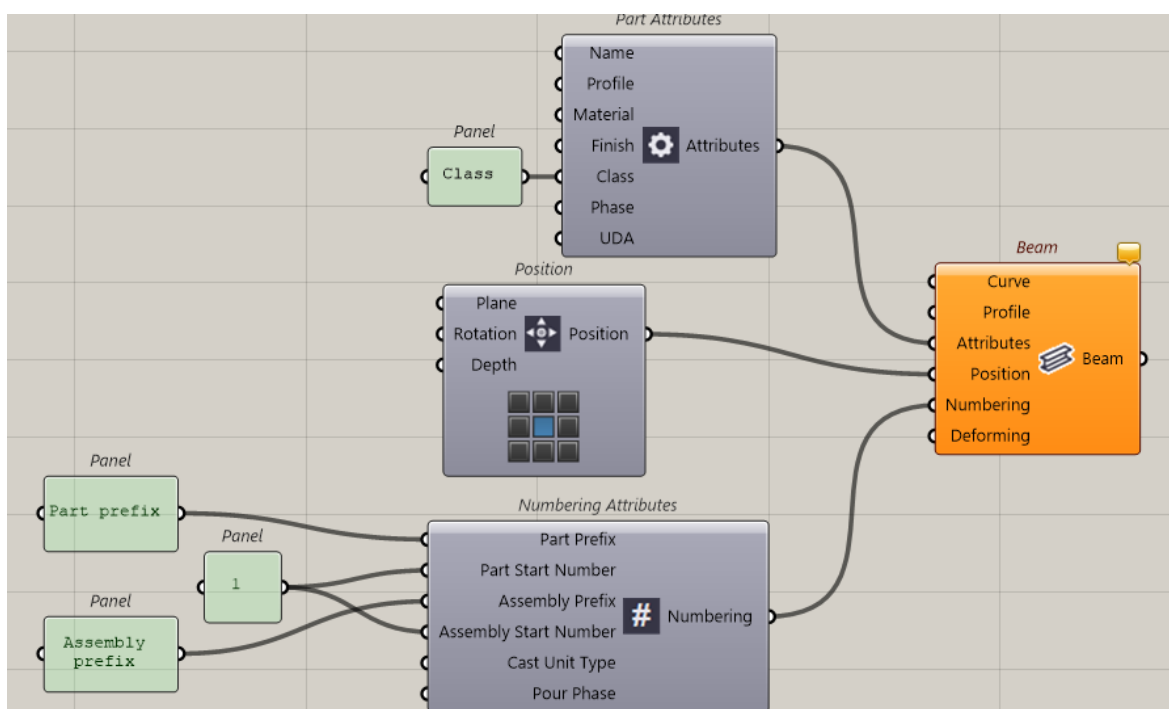
I en del komponenter kan man modifiera antalet inputs och outputs, alltså de portar som komponenter tar emot och ger ut data ifrån. Denna funktion är användbar då man behöver ge in ett flertal parametrar i en enda komponent för att få det resultat man behöver eller vill ha. För att ändra antalet inputs eller outputs kan man zooma in på komponenten och sedan lägga till eller ta bort input- eller output portar. Se exempel i figur 15, var man zoomat in på en additionskomponent. (04.04. Grasshopper Components, Learn Grasshopper, 2023).



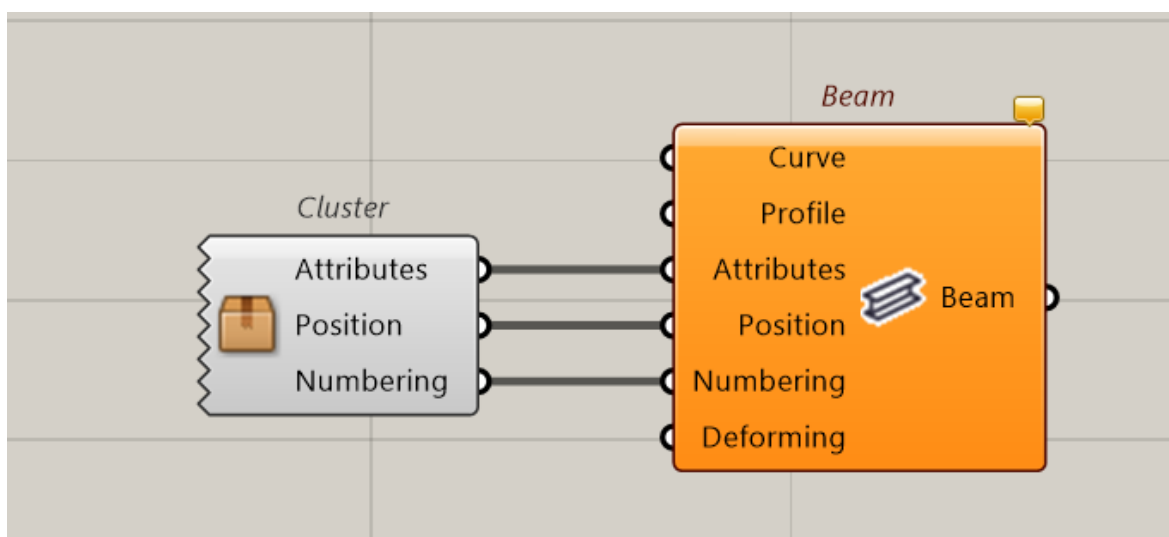
Figur 15. Inzoomad additionskomponent, med plus- och minustecken kan man lägga till eller ta bort inputs.

I allmänna komponenter behöver man ange data för att komponenten ska utföra dess uppgift. Ifall man inte vet vilken data som ska anges vart kan Grasshopper hjälpa en. Genom att föra pilen över den specifika delen av komponenten kan Grasshopper berätta vad som borde kopplas in i vilken input. (04.04., Learn Grasshopper, 2023).

Ett kluster är en funktion som ger användaren möjligheten att gruppera ihop flera komponenter till en enskild komponent. Klusterkomponenten kan öppnas och redigeras vid behov. Ett kluster är användbart då man vill få en stor mängd information som inte behöver vara lättillgänglig till en ensam komponent. Se exempel i figurerna 16 och 17. I figur 16 är alla komponenter som kopplats till Teklakomponenten utspridda och synliga i koden. Men i figur 17 har komponenterna grupperats ihop till ett kluster och för att modifiera komponenterna måste man nu öppna klustret genom att dubbelklicka på komponenten. (03.02. Grasshopper Interface 2/3, Learn Grasshopper, 2023).



Figur 16. Komponenter i Grasshopper.



Figur 17. Samma komponenter som i figur 16, men här grupperade till ett kluster.

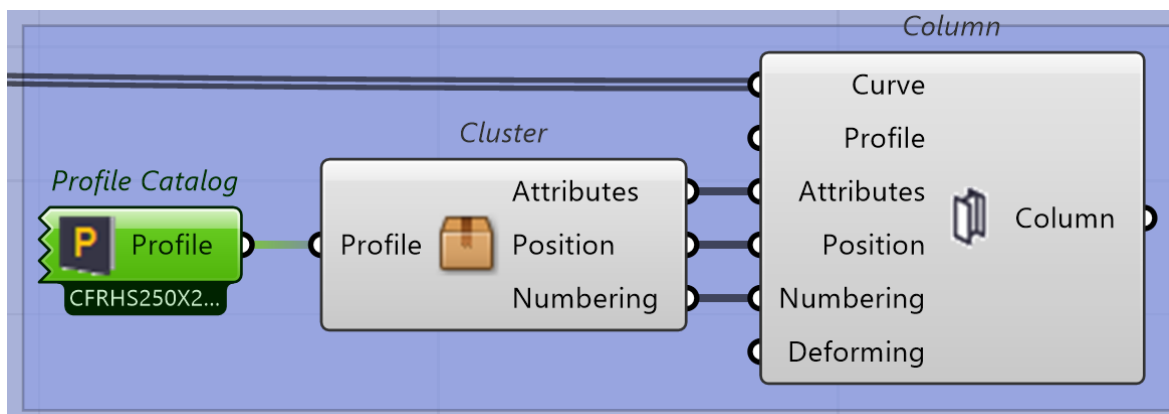
3.3 Tekla Structures

Tekla, kort för Tekla Structures, är ett BIM-program som används för 3D modellering och ritningsproduktion. I programmet kan man modellera objekt i olika byggnadsmaterial, de material som användaren kan modellera i är stål, betong, trä och glas. Nordec använder nästan uteslutande stål i sina projekt och därför används stål som byggnadsmaterial i detta examensarbetsexempel. (Trimble, u.d.).

I detta examensarbete används utöver Rhino och Grasshopper också Tekla med en Live Link funktion. Tekla används bara för att visualisera koden ur Grasshopper som en modell i Tekla. Det vill säga, Tekla används inte i detta examensarbete för något annat än att visualisera den kod som skrivits i Grasshopper. Vidare detaljering kan göras i Tekla men den delen tas inte upp i detta examensarbete.

3.3.1 Live Link

En live-link möjliggör för algoritmisk design i Tekla med Rhino och Grasshopper. Länken mellan programmen fungerar på samma sätt som länken mellan Rhino och Grasshopper. Då koden skrivs eller modifieras i Grasshopper kan man i realtid se hur modellen byggs upp eller ändras i Rhino. På samma sätt fungerar det med Tekla, man kan alltså ange kodens geometri ur Grasshopper som objekt i Tekla, exempelvis balkar eller pelare, och sedan se hur modellen ändras i Tekla då man modifierar kodens parametrar i Grasshopper. I figur 18 har geometrin ur koden kopplats till den komponent som kopplar Grasshopper med Tekla. Då koden är klar i Grasshopper kan man koppla loss och fortsätta jobba med vidare modellering i bara Tekla, då fungerar dock inte längre live linken. (Tekla Support).



Figur 18. Komponenter som används för att ange geometrin ur Grasshopper som, exempelvis, pelare (column) i Tekla.

Då man angett i komponenten, se figur 18 och inputporten Curve, vilken geometri ur koden det är man vill använda kan man se objektet i Tekla. För att vidare specificera olika attribut för objektet kan man skilt ange vilken profil Tekla ska använda samt prefix- och numreringsattribut. I detta exempel, se figur 18, har attributen grupperats ihop till ett så kallat kluster för att spara på utrymme i koden. (Tekla Support).

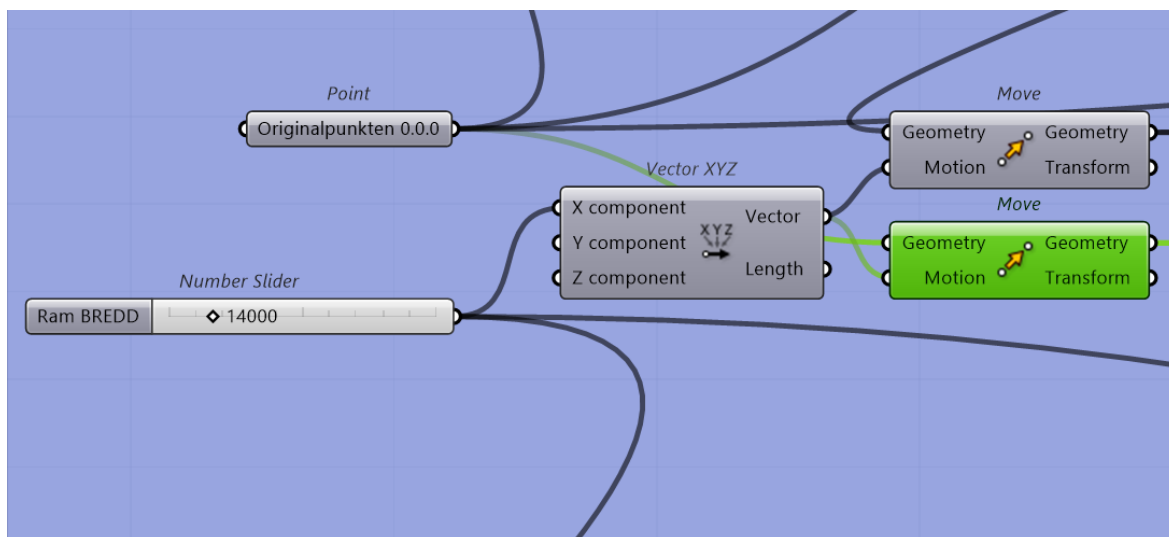
För att använda en live-link med Tekla behöver användaren ha både en licens för Rhinoceros och en för Tekla Structures, sedan kan man ladda ner Grasshopper Tekla link från food4rhino.com. (McNeel Europe, u.d.).

4 Modellen

För att skapa den modell som eftersträvs och med vilken målet med detta examensarbete har varit, har man redan i början av arbetet bestämt sig för en taktik. Då man ska skapa en modell med visuell kodning behöver man välja vilket sätt man ska jobba på, hur man ska gå till väga för att få ut specifika resultat. Detta eftersom det oftast finns flera sätt att göra en specifik sak på, vilket beskrivs mera ingående i kapitel 2.2 med ett visuellt exempel.

Val av taktik beror på flera olika saker, men viktigast är ifall man har parametrar att utgå ifrån redan vid start. Om man redan har koordinater, punkter eller annan geometri att utgå ifrån är valet av taktik troligtvis ett annat än då man kan börja sin modell utan några krav på startparametrar. Då man valt en taktik, ett arbetssätt, kan man börja bygga koden.

Modellen som skapats för detta examensarbete är uppbyggd nästan uteslutande av punkter och linjer. Dessa punkter och linjer bildar det som i modellen ska efterlikna pelare, balkar, stag och fackverk. För att skapa helheten av en modell har dessa punkter och linjer kopierats ut enligt givna parametrar, exempelvis cc-mått mellan pelare och höjder på pelare och fackverk. Denna taktik har valts eftersom det är ett relativt enkelt sätt att jobba på och det är lätt att hålla koll på de olika delarna i koden. Genom att kopiera ut en originalgeometri kan en viss del av koden användas flera gånger vilket sparar en del tid vid kodandet.



Figur 19. Komponenter kopplade till komponenten Move, här grönmarkerad, som används för att kopiera ut geometri.

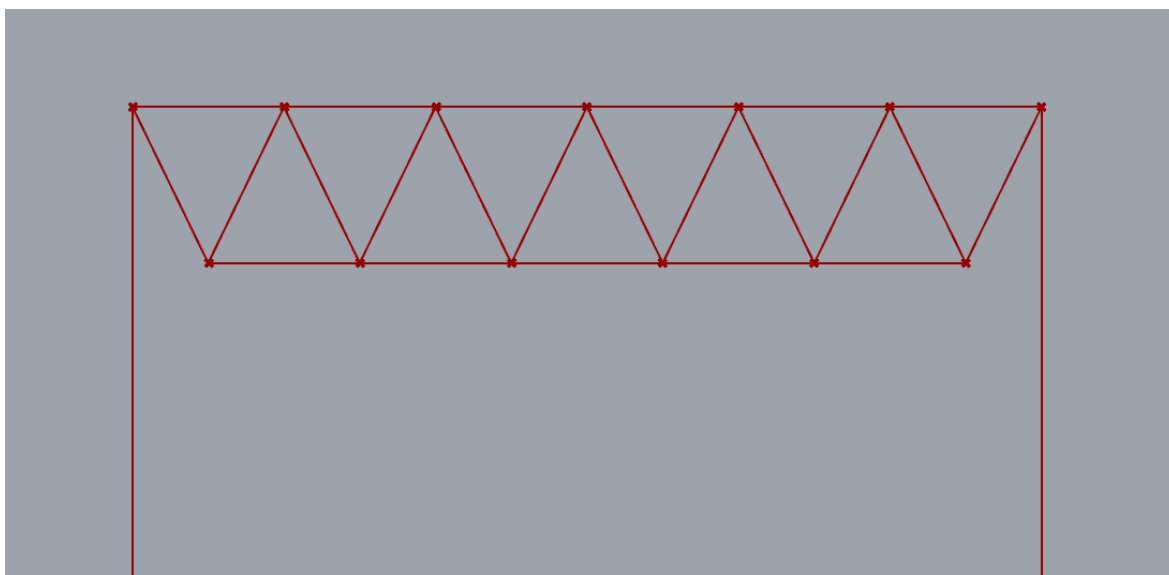
Första steget i skapandet av examensarbetets modell var att ange en startpunkt. I denna modell blev startpunkten, för enkelhetens skull, i koordinaten 0,0,0. Denna punkt har därefter kopierats ut genom att använda samma x- och y-koordinater som för originalpunkten, men med en ny z-koordinat i en Construct point komponent. Originalpunkten kopieras ut i z-led för att ange höjden på den första pelaren. Höjden på den första pelaren har angetts med en nummer slider, vilket betyder att höjden på alla pelare enkelt kan ändras i efterhand med den nummer slidern. Mellan dessa två punkter som anger start och slut på den första pelaren, skapas en linje. Denna trio av komponenter anges som geometri i en Move komponent varefter man kan kopiera ut pelare i både x- och y-led. Komponenten tar given geometri och kopierar ut den enligt en given vektor, se figur 19. Vektorn som angetts i Move komponenten är också skapad med en nummer slider vilket gör att man kan ändra cc-mått mellan pelarna, i figur 19 i x-led.

För att ange just höjd på och avstånd mellan pelarna används i detta examensarbete dessa nummer sliders, vilka beskrivs i kapitel 3.2.2. Med en nummer slider kan man bland annat justera x-, y- och z-koordinaterna för en punkt, men endast ifall punkten skapats med komponenten Construct point.

Modellen som skapats är en enkel modell med två pelare och ett fackverk som skapar en ram. Den ramen kan man sedan kopiera ut på längden och bredden. Mellan pelarna i x-led finns fackverk och mellan pelarna i y-led finns stag. Dessa stag skulle kunna bytas ut mot

sekundärfackverk men det har inte gjorts i detta examensarbete. Möjligheterna med sekundärfackverk tas upp i kapitel 6, Diskussionen.

Fackverken som finns mellan pelarna är skapade genom att man har skapat två parallella linjer, en övrebom och en undrebom. Linjerna har delats upp med hjälp av punkter för att skapa avstånden mellan diagonaländorna. Övrebommen har indelningen x och undrebommen har indelningen $x-1$, detta för att skapa lika stora mellanrum mellan diagonalerna och för att de ska bli just diagonaler och inte vertikaler. Mellan punkterna på övrebommen och punkterna på undrebommen skapas linjer, alltså diagonalerna. I figur 20 kan man se punkterna som skapar start- och slutpunkter för diagonalerna på över- och undrebommen.



Figur 20. Stickmodell ur Rhino för fackverk.

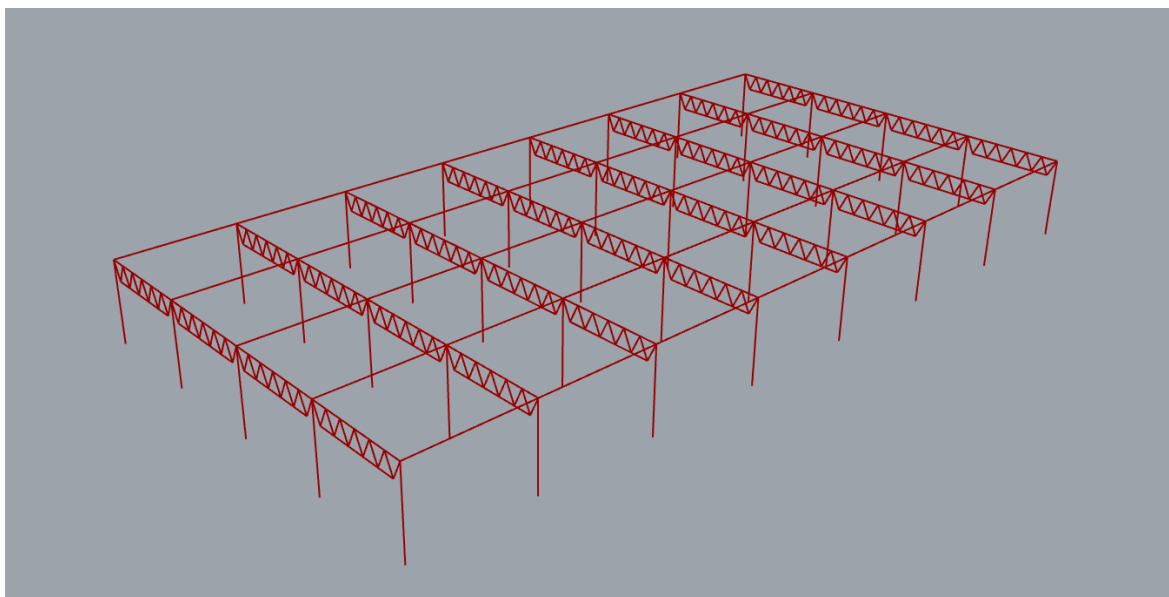
De parametrar för fackverket som skapats med nummer sliders är höjden på fackverket och indelningen av övrebommen. Dessa parametrar går att modifiera i efterhand tack vare dessa nummer sliders, se figur 21. Fackverket kan som sagt ändras enligt behov, men längden på fackverkets övrebom motsvarar c/c avståndet mellan pelarna. Också snedställda fackverk går att använda, men i detta exempel är fackverken parallella.



Figur 21. Nummer sliders för fackverkets geometri.

Första problemet som uppstod med modellen var att den enkla uppbyggnaden av koden blev svår att hantera då modellen blev större. För en större hall behöver man ha möjligheten att rensa bort en del pelare som i den verkliga modellen blir överflödiga. Detta diskuteras vidare i kapitel 6.0.

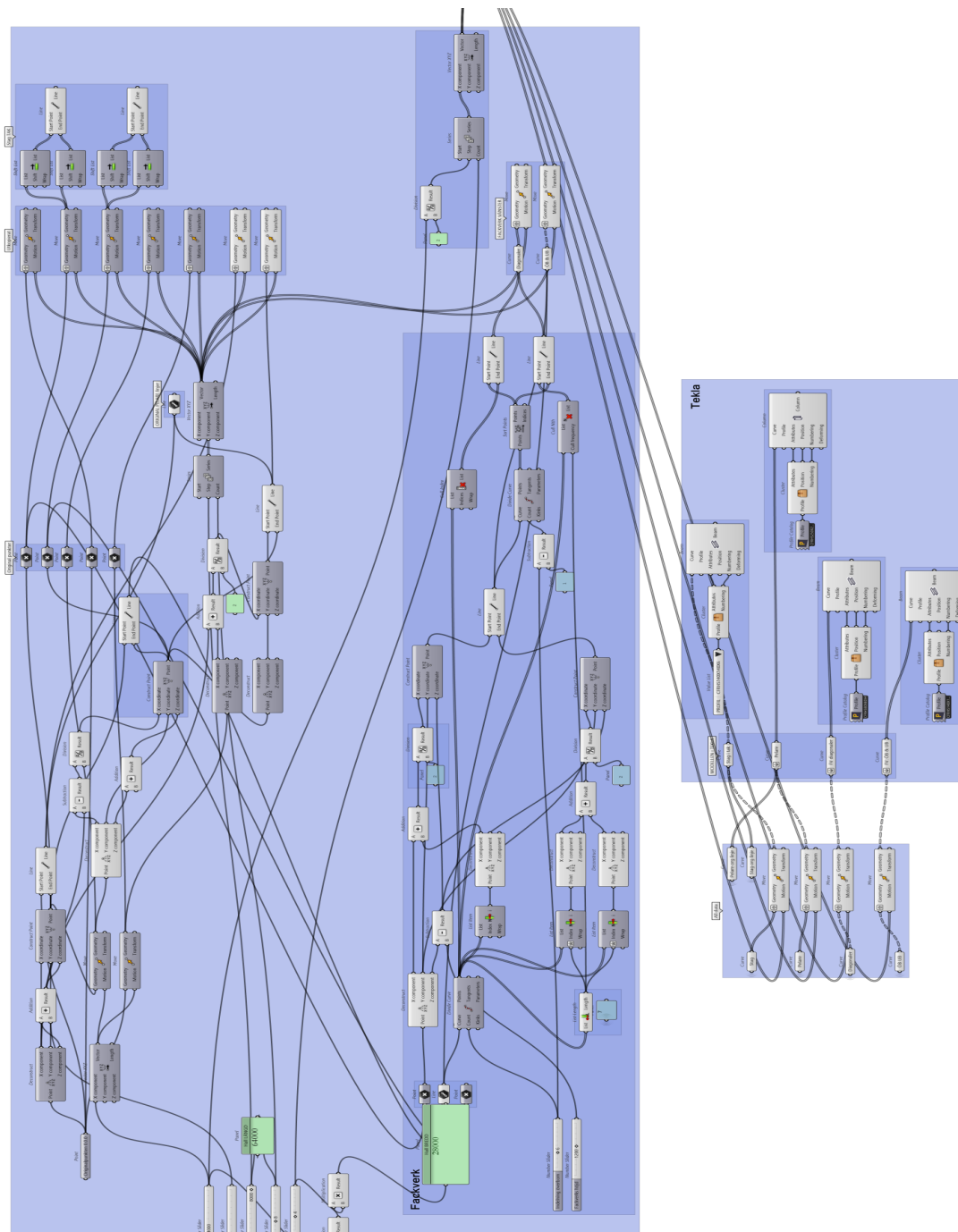
Den färdiga modellen blev en enklare modell än vad som från början hade planerats. Hallen går att ändra parametriskt på bredden, längden och höjden. Fackverken mellan pelarna håller samma längd som cc-måttet mellan pelare men man kan justera höjden och diagonalantalet. Modellen i Rhinoceros ses i figur 22.



Figur 22. Stickmodellen i Rhinoceros, skapad med kod i Grasshopper.

4.1 Grasshopper-koden

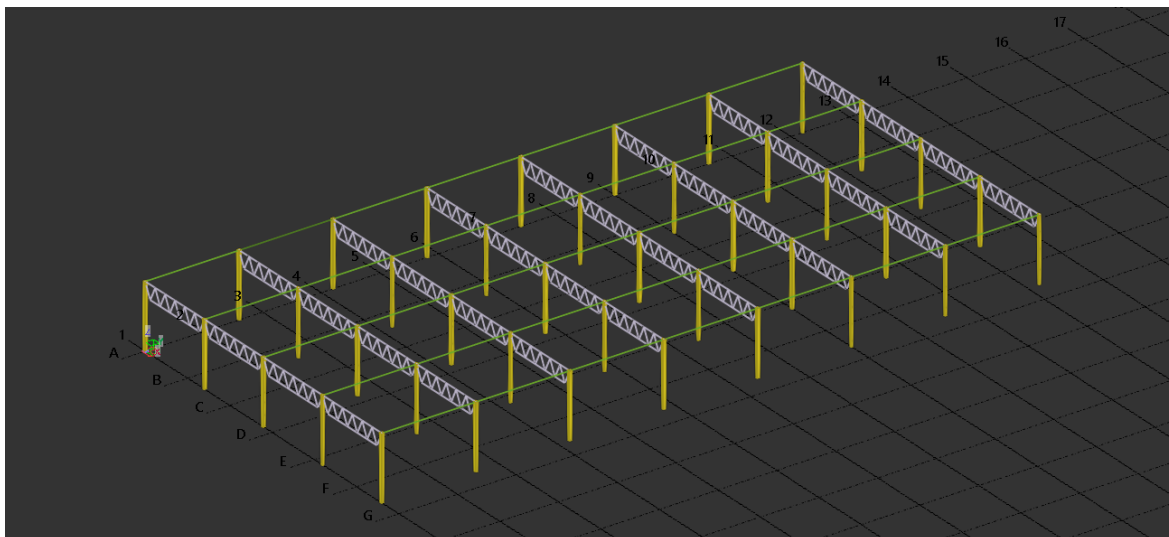
Koden är uppbyggd i fyra större grupper, en del för hallen med punkterna och linjerna som skapar geometrin för pelare och stag, en del för originalfackverket och dess Move komponent som kopierar ut fackverket med resten av modellen. De två mindre delarna är en samling av all data i modellen och den del som kopplar ihop Grasshopper med Tekla via Live Linken. De två sistnämnda delarna har ingen betydelse för hallen som helhet utan används endast för Tekladelen av arbetet. Den fullständiga koden kan ses i figur 23 och Bilaga 1.



Figur 23. Examensarbetets kod ur Grasshopper.

4.2 Live Link till Tekla Structures

För att koppla modellen till Tekla samlades all data i skilda komponenter. Pelare, stag, fackverkens övre- och undrebom och fackverkens diagonaler kopplades alla skilt till varsin Curve komponent. Den samlade datan för varje del kopplades in i motsvarande Teklaspecifika komponent. Datan för pelare kopplades ihop med geometri-inputen för Teklakomponenten Column och så vidare med samma princip för alla halldelarna.



Figur 24. Grasshopper-koden kopplad till Tekla Structures med Live Link.

Live Linken fungerade som planerat och modellen kunde direkt ses i Tekla. Den parametriska delen fungerade också som tänkt och modellen gick att ändra i realtid med hjälp av nummer sliders i koden. Också de objektspecifika attributen fungerade som de skulle och informationen överfördes från Grasshopper-komponenterna till objekten i Tekla. Modellen i Tekla kan ses i figur 24.

5 Resultat

Målet med detta examensarbete var att undersöka möjligheterna med den parametriska modelleringen vid modellering av envåningshallar. Intressant för arbetet var att ta reda på ifall den parametriska modelleringen kunde eliminera repetitiva och tidskrävande moment i modelleringsprocessen. Det konkreta slutresultatet blev en parametriskt modellerad hall som fungerar i princip som planerat. Hallen kan modifieras enligt behov och önskemål. Hallen som är uppbyggd av pelare, stag och fackverk kan modifieras genom att ändra på cc-avstånden för pelare, antal pelarrader och höjd på pelare och fackverk.

Med hjälp av den parametriska modellen kan man absolut starta upp en modell i Tekla och jobba vidare från den, men för en van Tekladesigner snabbar nog inte modellen i detta examensarbete upp den initiala modelleringen. Designern blir fortfarande tvungen att manuellt lägga in en del pelare, balkar och stag men också radera en del av geometrin som skapas av koden.

En sak som kunde gjorts annorlunda är uppbyggnaden av koden. Ifall man redan vid start hade strukturerat upp koden på ett annat sätt hade den varit lättare att hantera då koden, och då också modellen, blev större och mera komplicerad. Tacksamt hade varit ifall koden gjorde det möjligt att välja ut specifika data som kunde raderas eller hanteras på annat sätt. En sådan process är med den nuvarande modellen också möjlig, men besvärlig. Denna konklusion möjliggör för vidareutveckling av arbetet.

5.1 För- och nackdelar med parametrisk modellering

Parametrisk modellering underlättar vid stora förändringar i en modell. Sådana ändringar som kommer vartefter modellen utvecklas. Det är lätt, tack vare den parametriska modelleringen, att ändra på modellens geometri i efterhand. För att ändra på en modell behöver användaren eller designern bara ändra önskade input för att få ut det resultat man söker.

Den kod som byggs upp för en modell kan lätt användas många gånger. Genom att man ändrar på de input som anger modellens geometri kan man få ut flera olika resultat fastän man använder samma koduppbyggnad. Den parametriska modelleringen för med sig många möjligheter just på grund av dess mångsidighet.

De nackdelar som finns med parametrisk modellering är fram för allt att användaren bör ha tillräcklig kunskap om både programmen som används och de komponenter som behövs för att skapa den modell man önskar få som resultat. Också det faktum att koden lätt blir rörig kan vara en nackdel. Ifall koden blir rörig är det svårt att hålla koll på alla delar i koden. Sådana komponenter som inte direkt har en synlig funktion blir snabbt bortraddade ifall designern inte har ordning på komponenterna.

5.2 Besvarade forskningsfrågor

Hur kan modellen underlätta för designers i praktiken och när har man användning för den?

Ifall den modell som skapats för detta examensarbete skulle utvecklas, kunde den användas för den initiala modelleringen vid projektstart. Då skulle den underlätta för designers eftersom den skulle eliminera de repetitiva moment som ingår i modelleringen. Exempelvis utplacering av pelare, balkar, stag och fackverk. Modellens önskade mått och profiler skulle anges i kodens inputs och designern skulle därefter kunna lossa modellen till Tekla via Live Linken. Efter det är det upp till designern att manuellt modellera detaljer.

Vilka andra arbetsmoment kunde snabbas upp med hjälp av parametrisk design?

Andra arbetsmoment som kunde snabbas upp med en kodad modell skulle exempelvis kunna vara modelleringen av trappor, och mera specifikt spiraltrappor, vilka är väldigt tidskrävande att modellera för hand. Med en parametrisk modell kunde designern ange höjder och bredd för trappan samt vilka profiler som ska användas och sedan lossa den färdiga spiraltrappan till Tekla via Live Linken. Också för håldäcksräcken kunde den principen användas, användaren kan ange spannet för räckets, måtten för mellanrummen och de önskade profilerna och sedan koppla loss modellen till Tekla.

6 Diskussion

Under ingenjörarbetets gång har jag fått väldigt mycket ny kunskap. Till att börja med var ämnet parametrisk modellering helt nytt för mig. Jag har lärt mig om möjligheterna med parametrisk modellering och studerat algoritmisk design och visuell kodning. Två av de tre program som jag jobbat med i examensarbetet var också nya. Programmet Rhinoceros var relativt lätt att lära sig medan programmet Grasshopper var betydligt svårare och mera komplicerat. En annan sak som jag vill uppmärksamma är: då man jobbar med parametrisk modellering i Grasshopper måste man lära sig att tänka på ett annat sätt än vid manuell modellering. Man måste redan i början av koden tänka på hur man vill att resultatet ska bli och hur man ska gå till väga för att uppnå det, eftersom hela koden hör ihop. Logiskt tänkande är en viktig del i den parametriska modelleringen.

Jag har under arbetets gång läst flera artiklar och andra publikationer där personer med erfarenhet inom området säger att parametrisk modellering är framtiden. Då arkitekturen blir mera avancerad krävs också mera avancerade modelleringsverktyg, så som algoritmisk design. Detta är jag beredd att hålla med om. Jag tror att jag kommer ha stor nytta i framtiden av det jag lärt mig i och med detta examensarbete.

Som tidigare nämnt i kapitel 5.0 som behandlar resultatet finns det en del saker som kunde gjorts annorlunda i arbetet. Till att börja med hade det varit önskvärt att redan vid start satt ut fler och tydligare avgränsningar för arbetet samt börjat koden på ett annat sätt. Ifall jag hade jobbat lite annorlunda och satt mera fokus på att jobba med datahanteringen i stället för att manuellt koda stora delar av koden hade slutresultatet antagligen blivit snäppet bättre och mera användbart.

En annan sak som kan tänkas fattas i arbetet är möjligheten att använda både primär- och sekundärfackverk. För en mera realistisk hall skulle stagen som finns mellan pelartopparna i y-led gärna bytas mot fackverk. De fackverken skulle fungera som sekundärfackverk. Helst av allt skulle man ha möjligheten att välja på vilka linjer man vill använda primärfackverk, på vilka man vill använda sekundärfackverk och på vilka man vill använda stag. För att använda den metoden borde arbetet utvecklas med Data Trees och fackverksdelarna grupperas ihop till kluster.

Vid en utveckling av arbetet skulle jag ändra arbetsmetod och jobba med en typ av komponenter som kallas just Data Trees. Det som skiljer den metoden åt från metoden som använts i arbetet är att det är lättare att jobba med stora mängder information om datahanteringen sker via dessa Data Trees. En annan sak jag skulle utveckla är att jag skulle bygga ihop vissa delar av koden till egna komponenter. Detta görs möjligt via Tekla Structures och Live Link men har inte använts i detta examensarbete. Jag skulle också utnyttja och modifiera de komponenter som Tekla redan har att erbjuda för att underlätta vid kodning av exempelvis fackverk. Som tidigare nämnts behöver en realistisk modell möjligheten att använda både primär- och sekundärfackverk. Det kunde göras enkelt genom att använda Teklas komponenter som bas för fackverken. Ytterligare skulle arbetet kunna utvecklas genom att testa att koda andra objekt också, till exempel trappor, räcken eller håldäck.

Jag är själv relativt nöjd med resultatet även om jag anser att en del av koden och modellen blev lite väl enkla. I och med att ämnet var nytt för mig har det många gånger blivit en del försök och misstag, men jag har lärt mig av mina misstag och det i sig är en lärdom. Jag är glad över att ha fått möjligheten att lära mig om parametrisk modellering och visuell kodning och jobbar gärna vidare inom ämnet i framtiden.

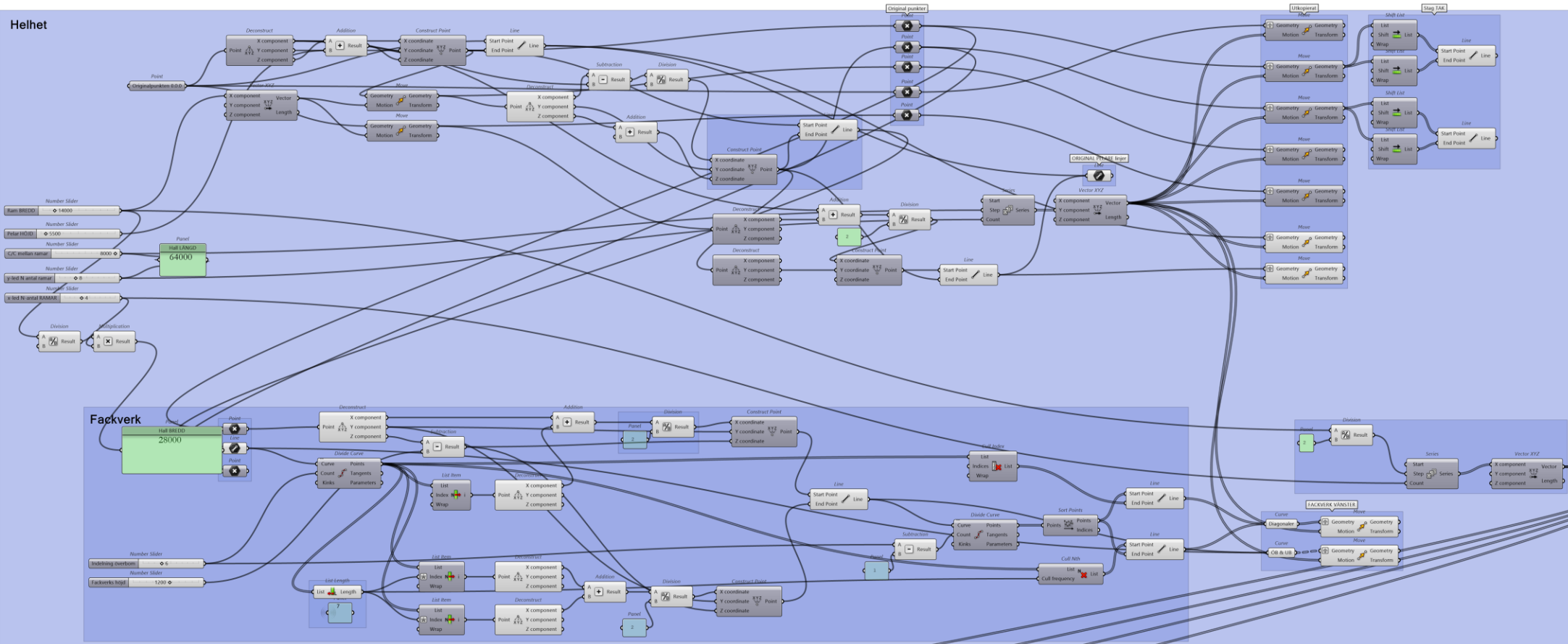
7 Källförteckning

- Algorithmic Design for Architecture. (u.d.). *Modelling*. Hämtat från <https://algorithmicdesign.github.io/expertise/modelling.html> den 10. 1. 2023
- AutoDesk. (u.d.). *AutoCAD product*. Hämtat från <https://www.autodesk.com/products/autocad/overview?term=1-YEAR&tab=subscription> 2023
- Chai, W. (u.d.). *CAD (computer-aided design)*. Hämtat från <https://www.techtarget.com/whatis/definition/CAD-computer-aided-design> den 20. 2. 2023
- Davis, D. (2013a). *Thesis, Chapter 2*. Hämtat från <https://www.danieldavis.com/thesis-ch2/> den 2. 2. 2023
- Davis, D. (2013b). *A History of Parametric*. Hämtat från <https://www.danieldavis.com/a-history-of-parametric/> den 3. 2. 2023
- Fu, F. (2018). *Design and Analysis of Tall and Complex Structures*. Hämtat från <https://www.sciencedirect.com/topics/engineering/parametric-modeling> den 27. 1. 2023
- Grasshopper 3D. (u.d.). Hämtat från Wikipedia: https://en.wikipedia.org/wiki/Grasshopper_3D den 17. 02. 2023
- Grasshopperdocs. (den 8. 3. 2022). *Component Index*. Hämtat från <https://grasshopperdocs.com/completelIndex.html> den 16. 3. 2023
- Heiskanen, A. (den 30. 4. 2019). *AEC-business*. Hämtat från <https://aec-business.com/how-algorithmic-design-improves-collaboration-in-building-design/> den 10. 1. 2023
- Lorek, S. (den 6. 4. 2022). *Trimble Construction*. Hämtat från <https://constructible.trimble.com/construction-industry/what-is-bim-building-information-modeling> den 11. 4. 2023
- McNeel Europe. (u.d.). *food4rhino.com*. Hämtat från www.food4rhino.com den 20. 2. 2023
- Nordec Oy. (u.d.). *About us*. Hämtat från <https://nordec.com/about-us/> den 16. 1. 2023
- Rhinoceros 3D a. (u.d.). *Download Rhino*. Hämtat från <https://www.rhino3d.com/download/> den 23. 2. 2023
- Rhinoceros 3D b. (u.d.). Hämtat från Wikipedia: https://en.wikipedia.org/wiki/Rhinoceros_3D den 20. 2. 2023
- Sterne, J. (u.d.). Hämtat från Britannica.com: <https://www.britannica.com/technology/plugin> den 20. 2. 2023
- Tekla Support. (u.d.). *Grasshopper-Tekla Live Link*. Hämtat från <https://support.tekla.com/fi/node/107964> den 16. 3. 2023

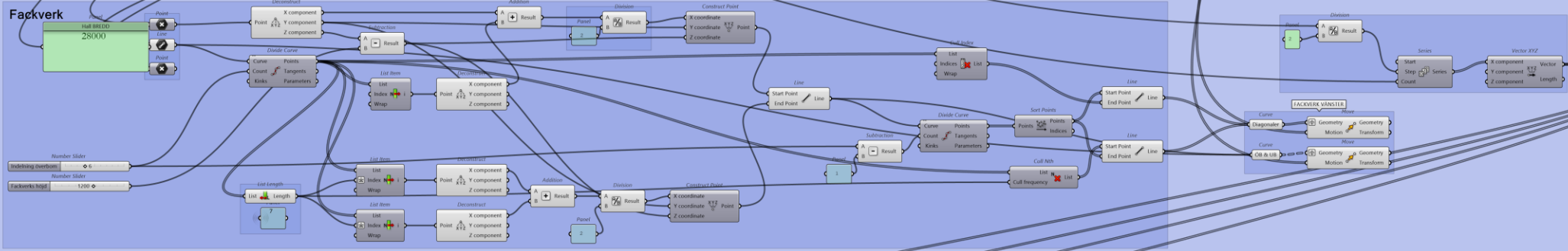
Trimble. (u.d.). *Tekla Structures*. Hämtat från <https://www.tekla.com/se/produkter/tekla-structures> den 13. 3. 2023

Trimble: Tekla Structures. (u.d.). *Overcoming the limitations of creating complex shapes with parametric modeling*. Hämtat från Tekla Resources: <https://www.tekla.com/resources/articles/overcoming-the-limitations-of-creating-complex-shapes-with-parametric-modeling> den 28. 2. 2023

Helhet



Fackverk



Tekla

