

# Användning av parametrisk design vid utformning av stomsystem för höga hus

Wille Dahlblom

Examensarbete för Ingenjör (YH)-examen

Utbildning i byggnads- och samhällsteknik

Raseborg 2024

## EXAMENSARBETE

Författare: Wille Dahlblom  
Utbildning och ort: Utbildning i byggnads- och samhällsteknik, ingenjör (YH), Raseborg  
Inriktning: Konstruktionsplanering  
Handledare: Towe Andersson

Titel: Användning av parametrisk design vid utformning av stomsystem för höga hus

---

Datum: 12.4.2024

Sidantal: 25

---

### Abstrakt

Detta examensarbete behandlar utvecklingen av ett Grasshopper-skript för Ramboll Sverige AB. Skriptet är skapat för att effektivisera den tidiga planeringsprocessen för höga hus. Arbetet börjar med att introducera parametrisk modellering och de program som används av skriptet, för att sedan gå djupare in på hur skriptet har skapats och vad dess huvudsakliga funktioner är.

Genom att integrera den parametriska modelleringen från Grasshopper med FEM-Designs beräkningsegenskaper, skapas ett automatiserat modelleringsverktyg som snabbt kan bygga upp och analysera enkla 3D-modeller. De stödreaktioner och förskjutningar som kommer från analysens resultat utgör grunden för en preliminär bedömning om den valda stömlösningen går att utföra eller inte. Skriptets syfte är att möjliggöra en snabbare uppbyggnad av 3D-modeller för olika stomsystem för en byggnad, så man kan jämföra dem och för att tidigare i planeringsprocessen veta vilka lösningar som är värda att utveckla vidare och vilka som bör bortses från.

Parametrisk design är ett aktuellt ämne inom konstruktionsplanering och målet med examensarbetet är att ge en inblick i hur parametriska modeller kan användas i andra sammanhang än vid formgivning av en byggnad för att underlätta och försnabba en konstruktions utveckling.

---

Språk: Svenska

Nyckelord: parametrisk design, grasshopper, finita elementmetoden, automatisering

## **BACHELOR'S THESIS**

Author: Wille Dahlblom  
Degree Programme: Civil and Construction Engineering, Tammissaari  
Specialisation: Structural Design  
Supervisor: Towe Andersson

Title: Use of Parametric Design in the Development of Structural Systems for Tall Buildings

---

Date: 12.4.2024

Number of pages: 25

---

### **Abstract**

This thesis addressed the development of a Grasshopper script for Ramboll Sverige AB. The script was created to make the early planning process of tall buildings more effective. The thesis began by introducing parametric modelling and the programs used by the script, and then went deeper into how the script is created and what its main functions are.

By integrating the parametric modelling capabilities of Grasshopper with the calculation engine in FEM-design, an automated modelling tool was created which is able to quickly construct and analyse simpler 3D-models. The support reactions and displacements derived from the results of the analysis, was then used to assess if the chosen solution is viable or not. The purpose of the script was to enable a faster construction of 3D-models of different structural solutions for a building, and thereby be able to compare them and also to know which ones are worth developing more and which ones should be disregarded.

Parametric design is a current topic in construction planning, and the aim of this thesis was to provide some insight into how parametric models can be used to ease and speed up the development process of a building in areas other than architectural design.

---

Language: Swedish

Key words: parametric design, grasshopper, finite element method, automation

# Innehållsförteckning

1	Inledning.....	1
1.1	Mål och syfte .....	1
1.2	Metodik.....	2
2	Parametrisk design.....	3
2.1	Exempel: Lansdowne Road Stadium .....	5
3	Program .....	6
3.1	Rhino.....	6
3.2	Grasshopper.....	7
3.2.1	Visuell programmering.....	8
3.2.2	Inputs .....	9
3.2.3	Outputs, listor och vidareföring .....	10
3.3	FEM-Design .....	11
4	Skapande av Grasshopper-skriptet.....	12
4.1	Utgångsläge .....	12
4.2	Uppbyggnad av 3D-modell.....	13
4.3	Lastberäkningar i Grasshopper.....	16
5	Analys i FEM-Design.....	19
5.1	Överföring till FEM-Design .....	19
5.2	Beräkningar .....	20
5.3	Test och optimeringar.....	21
6	Analysresultat.....	22
7	Sammanfattning och diskussion.....	23
8	Källförteckning.....	25

# 1 Inledning

Detta arbete är en beskrivning och genomgång av utvecklingen av ett program och arbets sätt till Ramboll Sverige AB för att i ett tidigt skede av planeringsprocessen kunna utreda olika stomlösningar för höga hus. Eftersom det nuvarande manuella arbets sättet är väldigt tidskrävande, söker de efter ett alternativt arbets sätt innefattande parameterisk design för att för snabba processen.

## 1.1 Mål och syfte

Målet är att göra ett skript med hjälp av Grasshopper och FEM-Design för att effektivisera den tidiga planeringsprocessen. Syftet är att kunden som vill bygga huset, snabbare ska få uppfattningen om olika stomlösningar går att genomföra. Skriptet är inriktat på att räkna ut stödreaktioner för byggnaden och hitta den maximala förskjutningen på grund av vindlast. Detta görs för att se om det uppstår stora dragkrafter i pelarna och för att kontrollera byggnadens stabilitet. Skriptet är till för att ge en överblick om den valda stomlösningen går att utföra efter mera ingående beräkningar. Det som inte beaktas är till exempel kostnader eller miljöpåverkan, men det är något som vid vidareutveckling skulle kunna integreras i skriptet för att få en mer ingående helhetsbild.

Syftet är inte att genom programmet få ut exakta värden för laster och liknande, utan det är till för att få en överblick över ett förslag för att se om stomlösning kan fungera efter mer detaljerade beräkningar. Avsikten med användningen av Grasshopper är att via parametrar kunna ändra till exempel våningsantal och -höjder samt att 3D-modellen med bland annat tillhörande laster och stödpunkter automatiskt ska uppdateras.

Genom att visa exempel på hur parametrisk design kan användas vid den tidiga planeringen av en byggnad är tanken med examensarbetet att skapa en förståelse för hur man i olika sammanhang, genom användningen av parametriska modeller, kan förbättra och för snabba en byggnads utformning.

## 1.2 Metodik

Arbetet börjar med att presentera begreppet parametrisk design och introducerar sedan de program som används av modelleringsverktyget som utvecklats för detta projekt. Skapandet av arbetssättet och Grasshopper-skriptet samt dess funktioner, tas sedan upp för att ge en djupare förståelse för hur en mer automatiserad 3D-modellering kan byggas upp.

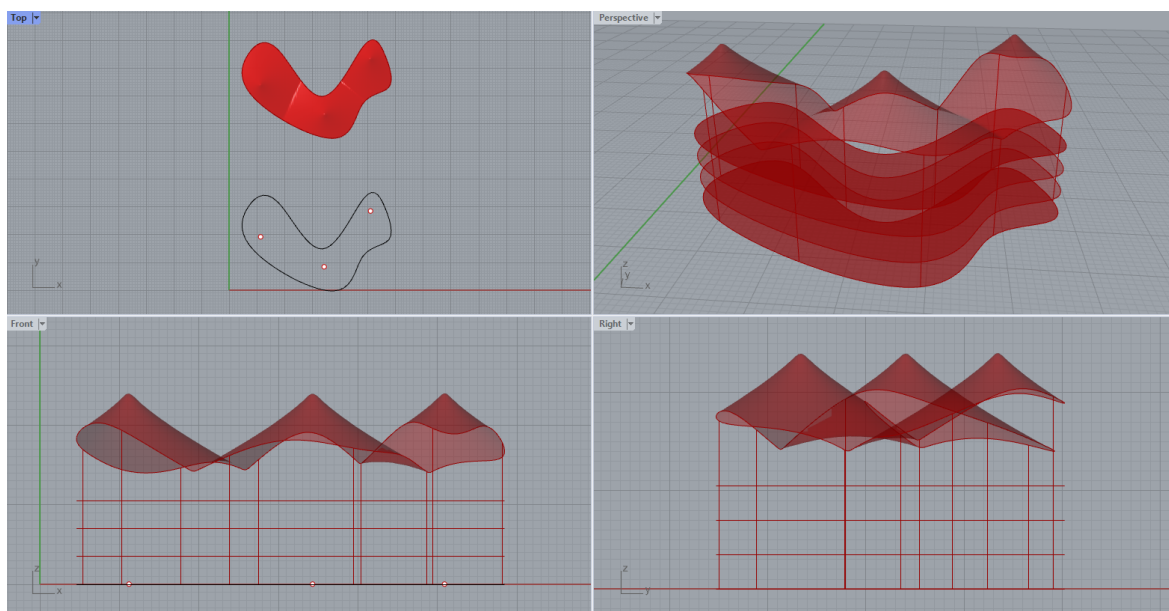
Skriptet utvecklades genom att först använda det visuella programmerings-programmet Grasshopper, där olika inbyggda komponenter kopplades ihop för att bygga upp en 3D-modell. Tilläggskomponenter som laddades ner användes sedan för att överföra denna modell till FEM-Design så att en analys kunde göras. 3D-modellen byggs upp utgående från linjer och punkter som för hand ritas i Rhino.

Den slutgiltiga versionen av Grasshopper-skriptet som examensarbetet behandlar har kommit fram genom att konstant utveckla och förbättra ett skript som började väldigt simpelt. Funktionen att kunna importera enkel data och utveckla den till en 3D-modell och sedan göra en analys var basen för skriptet. De olika versionerna har lagt till mer automatisering och fler valmöjligheter i både import samt uppbyggnad och analys av modellen. Detta för att minska den manuella inmatningen och därmed göra den tidiga planeringsprocessen snabbare.

## 2 Parametrisk design

Parametrisk design eller parametrisk modellering är ett sätt att skapa digitala modeller, oftast i 3D, med hjälp av bland annat matematik och variabler. Ordet "parametrisk" kommer från matematiken och hänvisar till användandet av vissa variabler eller parametrar för att påverka en ekvations resultat. Ofta används parametrisk design för att generera komplexa och invecklade geometrier som i vanliga fall skulle vara svåra att göra för hand i modelleringsprogram. Två vanliga program som används inom parametrisk design är Dynamo och Grasshopper, där Grasshopper är det mest populära. Båda programmen är grafiska algoritmredigerare, där man utan tidigare kunskap inom kodning enkelt kan skapa visuella skript. (Eltaweel & SU, 2017)

Nedan i figur 1 ser man ett exempel på en struktur med komplicerad geometri som skapats genom parametrisk design. Strukturens form fås från den handritade kurvan till vänster i figuren, och takets toppar samt lutning baserar sig på de utmärkta punkterna inuti kurvan. Höjden på taket och våningarna samt våningsantalet, kontrolleras med hjälp av parametrar och pelarnas höjd är kopplade till takets form.



Figur 1. Exempel på parametriskt modellerad struktur (Wille Dahlblom, 2024)

Dynamo är en open source programvara för visuell programmering som är tillgänglig att ladda ner gratis men som även automatiskt installeras som en del av Autodesk Revit sedan Revit 2020 togs i bruk. (Dynamo, 2023)

Grasshopper är likt Dynamo också ett program med en visuell programmeringsmiljö som är tätt integrerat med programmet Rhinos 3D-modelleringsverktyg. Grasshopper går däremot inte att ladda ner och använda som ett fristående program, utan kan bara användas genom Rhino vilket behöver en licens. (Rhinoceros 3D, 2023)

Det finns många områden och branscher som använder sig av parametrisk design bland annat inom arkitektur, stadsplanering och mode. Inom de flesta områden används parametrisk design för just design, ofta för att skapa naturligt formade och visuellt dynamiska strukturer och stilar. I andra mer analytiska områden såsom stads- och byggnadsplanering kan det användas för att till exempel snabbare optimera gångvägar eller att lättare räkna laster i komplicerade geometrier. (Eltaweel & SU, 2017)

En av de vanligaste implementeringarna av parametrisk modellering är att skapa relationer mellan olika delar av modeller och ofta med olika nivåer. Ett exempel kan vara att man definierar väggars höjd genom att koppla dem till nivåer på golv och tak. När positionerna av taket eller golvet sedan ändras, anpassas även väggarna och allt som är kopplat vidare från dem. Andra exempel som finns, är att koppla fönsterstorlek till golvyta eller att styra proportioner mellan ett sadeltaks lutning och höjden på väggarna som det hålls upp av. (Parametric modelling, 2021)

Allt detta går förstås också att bestämma på förhand och göra utan att använda sig av parametriska modeller, men vid eventuella ändringar kan arbetsprocessen förenklas och för snabbas avsevärt eftersom den slutliga modellen följer ändringarna. Parametrisk design möjliggör även att enklare optimera byggnader enligt specifika förhållanden, till exempel relationen mellan en byggnads form och de vindlaster som uppkommer. (Parametric modelling, 2021)



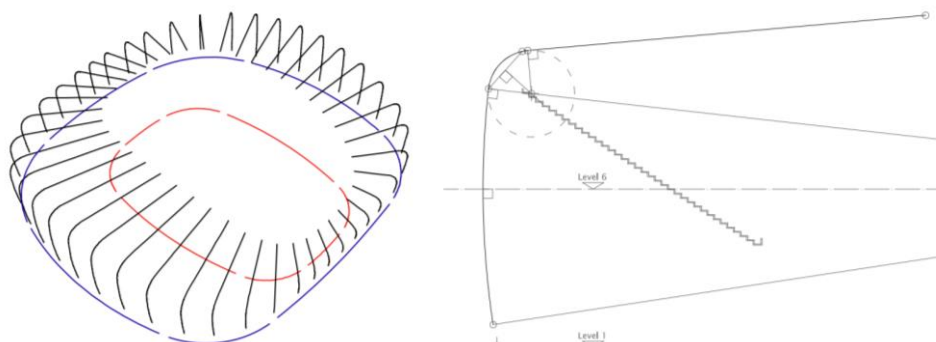
## 2.1 Exempel: Lansdowne Road Stadium

Fotbollsarenan Lansdowne Road Stadium/Aviva Stadium i Dublin, Irland, i figur 2 nedan är ett exempel på hur parametrisk design kan användas för både formgivning och planering av en byggnad. Den ursprungliga modellen gjordes manuellt och följde krav på bland annat solljus och sittplatser samt begränsningar av höjd på grund av omgivningen. Men eftersom arkitekterna förväntade sig att ändringar och justeringar skulle uppkomma i senare skeden, valde man att återskapa modellen parametriskt så att dessa skulle vara lättare att åtgärda. (Sheperd & Hudson, 2007)



Figur 2. Lansdowne Road Stadium/Aviva Stadium ([William Murphy](#), 2016)

Arenans fotavtryck och den inre kanten på taket, se blåa respektive röda kurvorna i figur 3, definierades av åtta tangentiella bågar vardera. De olika sektionerna byggdes sedan upp utgående från bland annat skärningen av dessa kurvor och ett radiellt rutnät, samt tre stycken kontrollkurvor som styrde de inre och yttre höjderna. (Sheperd & Hudson, 2007)



Figur 3. Referenskurvor för arenans fotavtryck, taklinje och sektionens form (Sheperd & Hudson, 2007)

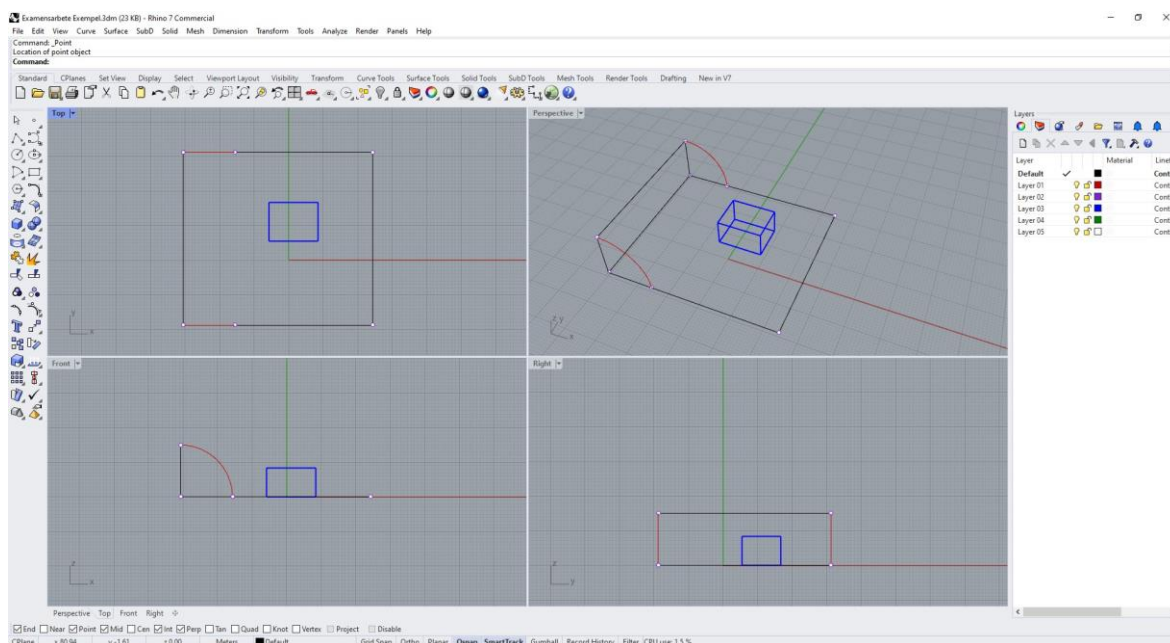
## 3 Program

De program som använts för att utveckla skriptet och arbetsättet som detta examensarbete behandlar är huvudsakligen Rhinoceros 3D, Grasshopper 3D och FEM-Design. Licenserna som behövs till både Rhinoceros 3D samt FEM-Design tillhör Ramboll.

### 3.1 Rhino

Rhinoceros 3D förkortat Rhino, är ett 3D modellerings- och visningsprogram som olikt många andra program inte är inriktat på en speciell bransch eller för en viss typ av modellering. Rhino används inom många olika industrier däribland allt mellan arkitektur och ingenjörskonst till film- och smyckesindustrin tack vare programmets mångsidighet. Rhino marknadsför även sin programvara som rimligt prissatt och lyfter fram att det är ett engångsköp som inte behöver återkommande betalningar för bland annat underhåll och uppdateringar. (Rhinoceros 3D, 2023)

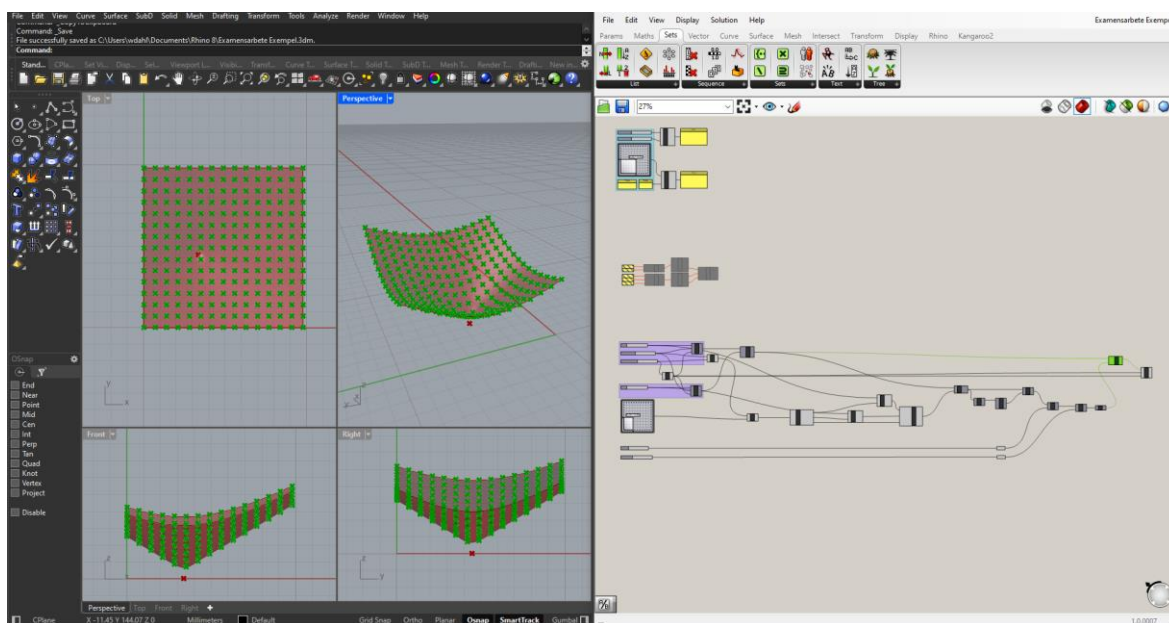
Nedan i figur 4 ser man exempelgeometri som ritats i Rhino. De olika vyerna är standardkonfigurationen från när man startar Rhino och de visar modellen i 2D ovanifrån, framifrån och från höger samt även i 3D perspektiv.



Figur 4. Exempelgeometri i standardkonfigurationen av Rhinoceros 3D (Wille Dahlblom, 2024)

## 3.2 Grasshopper

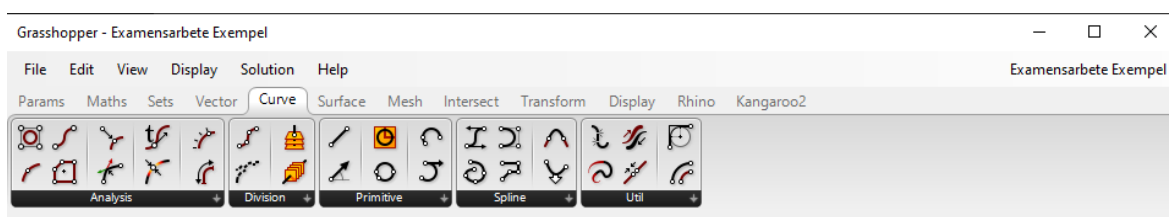
Grasshopper är som tidigare nämnt ett plugin-program inbyggt i Rhinoceros 3D, och är gjort för att grafiskt skapa och redigera algoritmer och parametriskt modellera i Rhino. Vid användning av Grasshopper kan man i realtid se hur 3D-modellen ändras i Rhino, i figur 5 nedanför ser man både Rhinos gränssnitt till vänster och Grasshoppers till höger. I fönstret för Grasshopper kan man se ett exempel på hur ett visuellt skript kan bli att se ut, och i Rhinos fönster ser man den geometri som skapats av skriptet. I detta fall är även en komponent som skapar punkter markerad i Grasshopper vilket gör att just de punkterna också blir markerade i Rhino.



Figur 5. Exempelbild från Rhino med Grasshopper vyn öppen (Wille Dahlblom, 2024)

### 3.2.1 Visuell programmering

Programmering i program som Dynamo och Grasshopper sker med olika komponenter kallade noder, som man kopplar samman genom att dra linjer mellan de olika nodernas ingångsplatser (*inputs*) och utgångsplatser (*outputs*). I Grasshopper hittas alla dessa komponenter i en meny överst på skärmen vilket syns i figur 6 nedan. Komponenterna är indelade i olika kategorier beroende på vilken sorts data de behandlar och är sedan indelade i olika grupper efter deras funktion. De olika noderna kan till exempel vara gjorda för att skapa, analysera eller modifiera bland annat nummer, linjer eller ytor. (Pavel, 2022)

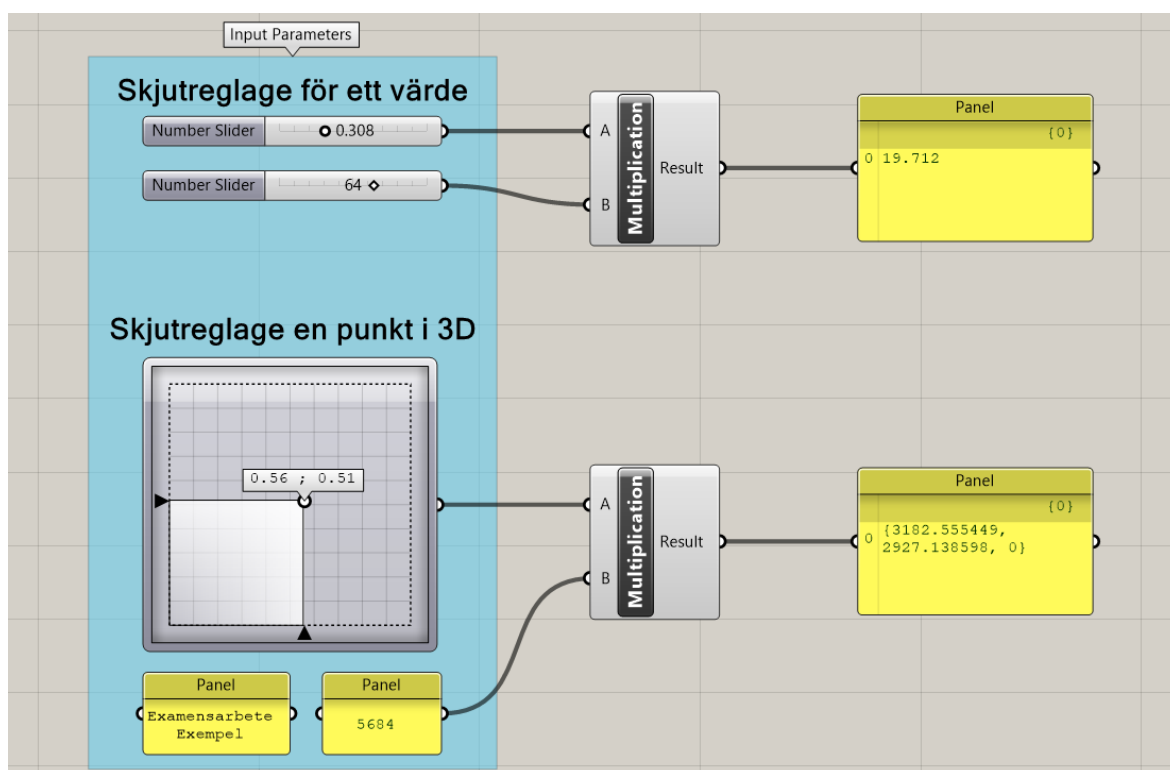


Figur 6. Komponentmenyn i Grasshopper (Wille Dahlblom, 2024)

Fördelen med att använda visuell programmering och konkreta komponenter i stället för att programmera i text, är att man utan djupa kunskaper inom kodning kan se vilka parametrar och noder som leder vart. Namnen eller ikonerna på komponenterna i Grasshopper är också till en del självförklarande, vilket gör att man snabbt kan förstå deras funktion. Om man klickar på en komponent markeras, förutom den, även dess kopplingar som gör det lättare att se varifrån dess inputdata kommer och vart den sedan leds. Nackdelen är däremot att stora skript med många komponenter kan bli röriga. En orsak till det är att kopplingarna som ritas upp kan gå över varandra och det kan då vara svårt att urskilja vart de leder. Ett exempel är om samma parameter används både tidigt och sent i skriptet och dess kopplingar då går igenom flera andra noder den inte är kopplad till. Man kan underlätta detta genom att antingen dölja kopplingar eller att bygga koden uppåt eller neråt i stället för bara vänster till höger.

### 3.2.2 Inputs

Det finns många olika sätt att skapa utgångsdata eller parametrar i Grasshopper, nedan i figur 7 är några exempel. Att skapa numeriska värden som snabbt går att ändra görs enklast genom att använda komponenten *Number Slider*, med vilken man drar ett skjutreglage till det nummer man vill ha. Inuti komponenten kan man ändra bland annat start- och slutvärden på skalan samt önskad mängd decimaler. I komponenten *Panel* kan man själv skriva siffror eller text och använda som inmatning, men den kan även användas till att visa vilka värden en annan nod producerar. (Bachman, 2017)



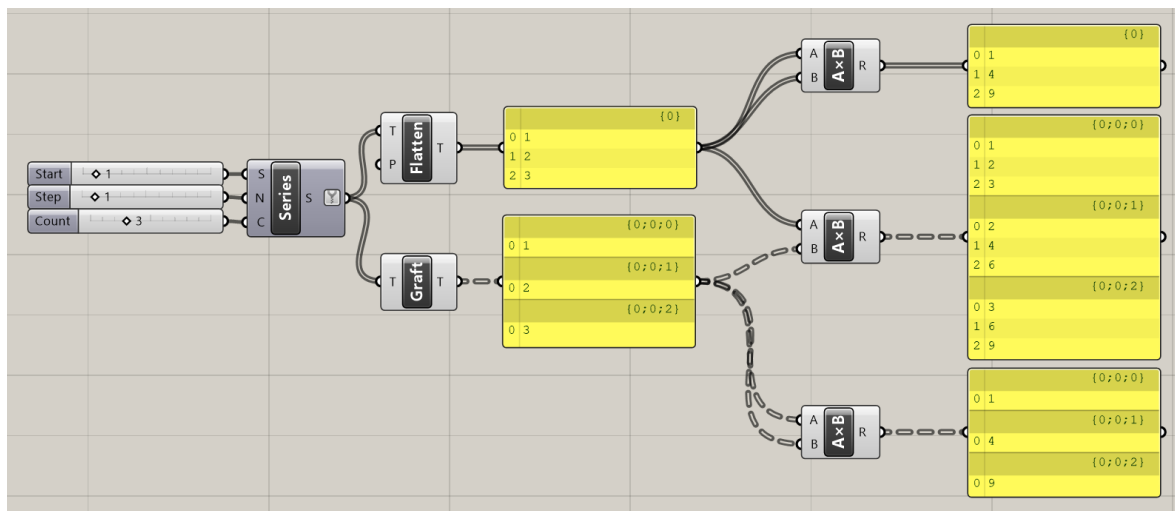
Figur 7. Exempel på olika input i Grasshopper (Wille Dahlblom, 2024)

Data kan även importeras till Grasshopper, antingen från Rhino, andra program eller direkt från filer. Exempelvis kan man på fri hand rita upp punkter, linjer eller annan geometri i Rhino och sedan importera den geometrin till Grasshopper. Den data man importerar fungerar på samma sätt som om man skulle skapat den med parametrar inuti Grasshopper.

### 3.2.3 Outputs, listor och vidareföring

Den data som en komponent behandlat eller skapat matas ut på nodens högra sida i form av listor. Om komponenten matar ut fler än ett dataobjekt grupperas de i listor, dessa listor kan placeras i olika nivåer av ett så kallat *data tree*. Listorna kan sedan förgrenas (*Graft*) eller sammanfogas (*Flatten*), vilket påverkar hur de olika dataobjekten behandlas i komponenter. (Bachman, 2017)

I figur 8 nedan har en serie med värden skapats, och till höger i figuren kan man se hur formatet på ingångsdata påverkar resultatet när man multiplicerar värden med varandra. De vänstra siffrorna i de gula panelerna är de olika värdenas index-nummer i deras listor och siffrorna uppe till höger i panelerna är hela listans numrering, all numrering börjar med noll. Det man också kan se är att om båda inputen bara är en lista, kommer värdena med samma index att multipliceras med varandra. Däremot om input A består av en lista med flera värden och input B består av flera listor, kommer listan från A multipliceras med alla listor i B.



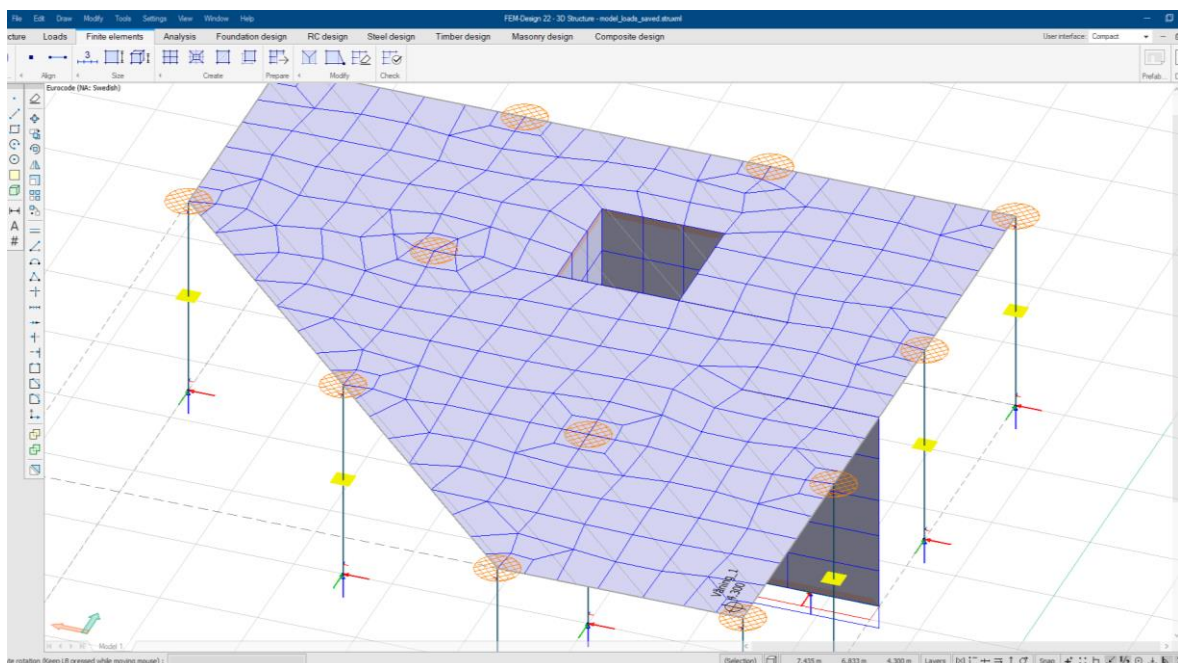
Figur 8. Multiplikation av en serie värden i Grasshopper (Wille Dahlblom, 2024)

### 3.3 FEM-Design

FEM-Design är ett program från det svenska företaget StruSoft AB och är utvecklat för att 3D-modellera, designa och analysera strukturer av stål, betong eller trä. Programmet kan göra analyser på enstaka konstruktionsdelar eller hela konstruktioner utgående från Eurokoder och nationella bilagor. Analyserna beräknas genom att använda sig av Finita elementmetoden, förkortat FEM. (FEM-Design, 2023)

Den finita elementmetoden innebär att en geometri delas in i ett nät (*mesh*) av mindre ihopkopplade regioner, kallade element. Hörnen på dessa element kallas noder och grundidén med FEM är att räkna ut bland annat förflyttningar och krafter i dessa noder för att sedan uppskatta motsvarande värden för den hela geometrin. Dessa lösningar blir endast ungefärliga och inte exakta, men de kan förbättras och göras mer precisa genom att bland annat dela in huvudgeometrin i mindre underregioner. (Rao, 2017)

FEM-Design genererar automatiskt de nät som delar upp de olika konstruktionsdelarna och används vid analys. De finita elementens storlek och form optimeras för konstruktionen de tillhör och anpassas bland annat enligt positioner på laster och stöd. Nedan i figur 9 ser man hur det blåa finita elementnätet (*Finite element mesh*) delar upp konstruktionen och hur den dras samman där pelarna kommer upp. (FEM-Design Wiki, 2022)



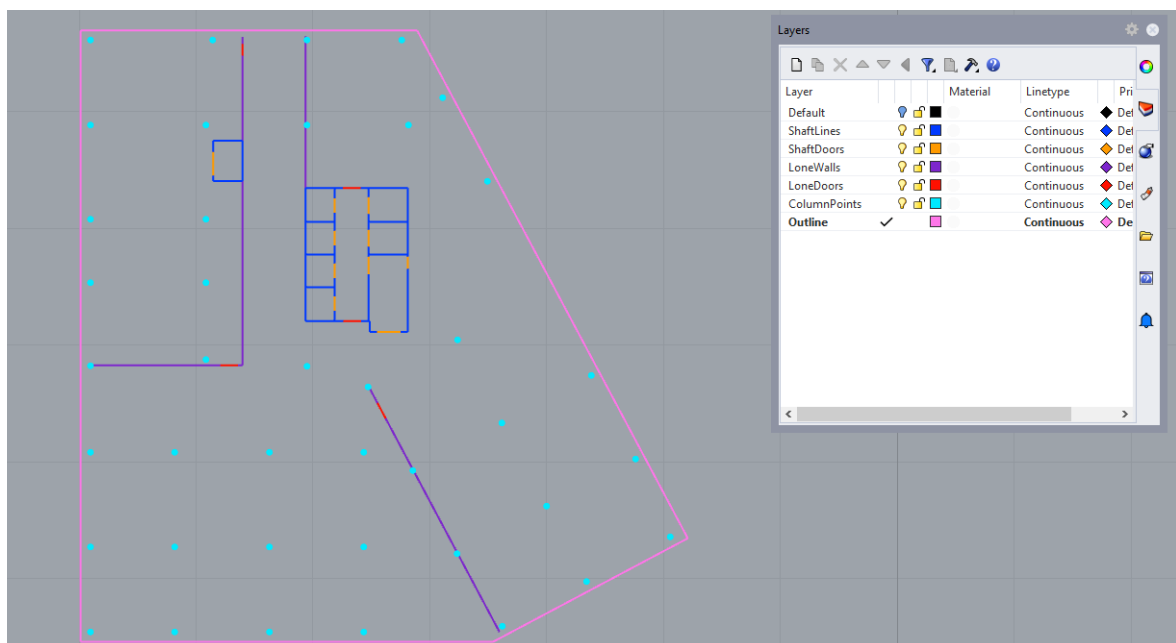
Figur 9. Finite element mesh för en exempelkonstruktion i FEM-Design (Wille Dahlblom, 2024)

## 4 Skapande av Grasshopper-skriptet

Den 3D-modell som i denna fallstudie byggs upp och analyseras, är ett exempel på hur en höghusbyggnad kan se ut. Skriptet i Grasshopper är skapat för just denna exempelbyggnad, men det är även gjort tillräckligt generellt för att utan större modifieringar kunna användas i framtida projekt. Eftersom skriptet endast är gjort för att skapa en förenklad 3D-modell av en byggnad, antas att planlösningarna för alla våningar är identiska.

### 4.1 Utgångsläge

För att skriptet i Grasshopper ska kunna bygga upp en modell, måste först en basgeometri definieras. Genom att rita ut punkter och linjer i rätt lager i Rhino, importeras de sedan till Grasshopper för att därefter bygga upp sina respektive delar av den slutliga 3D-modellen. I detta exempel har geometrin ritats av utgående från en arkitekturritning som importerats som PDF till Rhino. Nedan i figur 10 kan man se hur de olika färgerna på geometrin visar att de har separerats till olika lager. De linjer som väggar och dörröppningar baseras på, har delats upp i två olika lager vardera. Ett lager var för de väggar och öppningar som tillhör hisschakt eller andra rum där den inneslutna ytan inte ska modelleras som golv, samt ett lager för ensamma bärande väggar och ett för tillhörande dörröppningar.



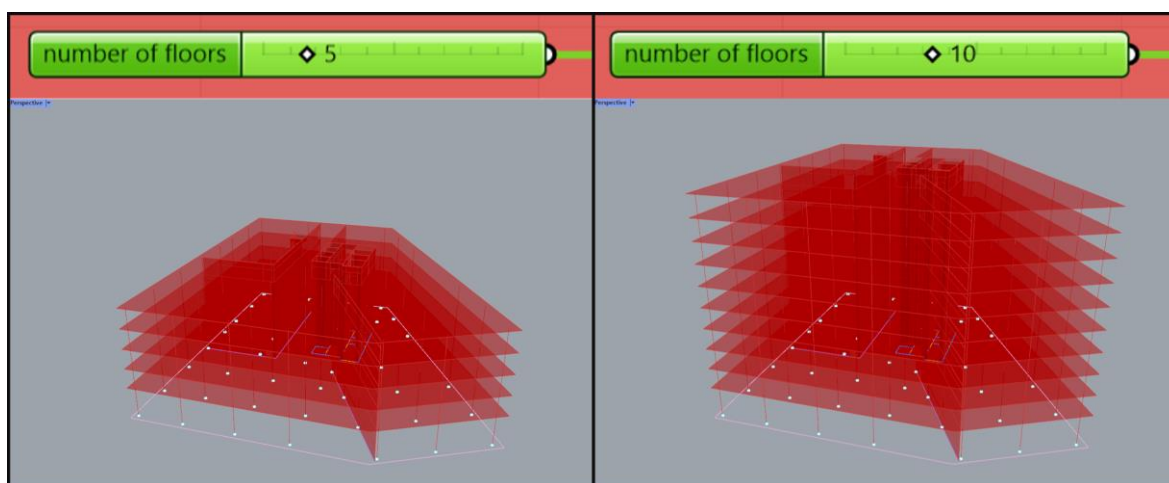
Figur 10. Exempelbyggnadens basgeometri i Rhino (Wille Dahlblom, 2024)



De blåa punkterna i figur 10 ovanför är centrumpunkter för bärande pelare och den rosa linjen som omger all annan geometri utgör den yttre konturen för golvytorna i konstruktionen. Anledningen till att byggnadens konturlinje, pelarnas centrumpunkter samt linjerna för de olika typerna av väggar ritas i separata lager i Rhino, är för att de ska kunna skiljas åt efter importen till Grasshopper. Detta behövs till exempel när områdena som avskiljs av hisschakten ska subtraheras från golvytan som genereras av konturlinjen. Grasshopper-komponenten *Pipeline* gör det möjligt att importera en specifik typ av geometri från ett visst lager i Rhino helt automatiskt, i stället för att manuellt välja den. Komponenten fungerar som en direktförbindelse och gör att geometrin i Grasshopper uppdateras automatiskt om något läggs till eller tas bort från det kopplade lagret.

## 4.2 Uppbyggnad av 3D-modell

Efter att basgeometrin importerats till Grasshopper genereras en 3D-modell baserat på ett antal parametrar som definieras i början av skriptet. Dessa parametrar styr antalet våningar, höjden på dem samt höjden på den första våningen om man valt att den skall vara annorlunda. Höjden på dörröppningarna i de bärande väggarna går också att justera med hjälp av en parameter. I figur 11 nedan har skjutreglaget som styr våningsantalet ändrats från fem till tio och därmed har antalet våningar i modellen i Rhino också ändrats i realtid.



Figur 11. Ändring av exempelkonstruktionens våningsantal i Grasshopper (Wille Dahlblom, 2024)

Höjdparametrarna kopplas samman för att skapa listor med alla våningars höjder och vilken nivå de börjar på. Dessa listor används senare i skriptet för att bland annat räkna ut vindlaster och för att styra uppbyggnaden av väggar och pelare. Väggarnas nedre del skapas genom att först extrudera deras baslinjer upp till den angivna höjden för dörröppningarna. Baslinjerna för väggar och öppningar flyttas då också till samma höjd så att de kan extruderas till den specificerade våningshöjden, och därmed skapa en vägg som inkluderar dörröppningarnas överstycken.

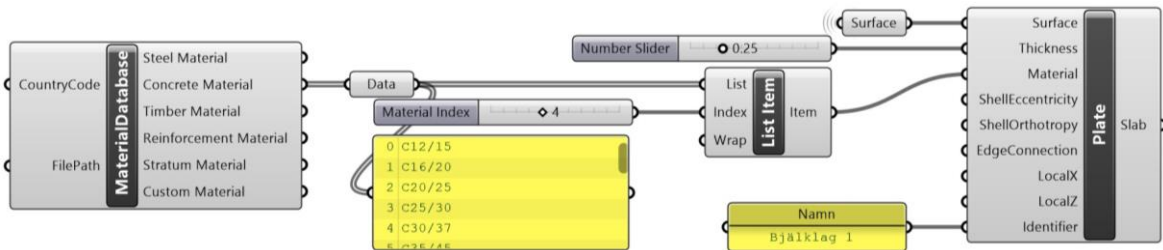
Modelleringen av pelarna sker på ett liknande sätt som för väggarna, men eftersom några dörröppningar inte behöver beaktas är processen enklare. Pelarnas baspunkter kopieras i stället direkt upp till den angivna våningshöjden och skriptet skapar därefter linjer mellan punkterna och deras förflyttade kopia.

Mer detaljerad information om de olika konstruktionsdelarna som bland annat material, tvärsnittprofil och tjocklek, har i detta skede av skriptet ännu inte definierats för att göra skapandet av Rhino-modellen snabbare. När pelarna endast visualiseras som linjer, och bjälklag samt väggar endast som ytor, förkortas modelleringstiden avsevärt jämfört med om alla konstruktionsdelar skulle genereras i en högre detaljgrad. Eftersom Rhino modellen inte är den modell som ska analyseras, utan endast fungerar som en bas, har modelleringshastighet också kunnat prioriterats över detaljnoggrannhet.

Basmodellen i Rhino är även uppbyggd med vidareföring till FEM-Design i åtanke, och är därför modellerad med de typer av Rhino geometri som enklast kan omvandlas till geometri som FEM-Design kan behandla.

För att via Grasshopper kunna skapa en modell och använda andra funktioner i FEM-Design, måste ett API för FEM-Design först laddas ner. Ett API (application programming interface), på svenska applikationsprogrammeringsgränssnitt, är en mellanhand mellan två olika program. I detta fall kopplas de två olika programmen ihop genom att ett tilläggspaket till med komponenter laddas ner till Grasshopper. API:et är framtaget för att man ska kunna använda FEM-Designs funktioner utan att behöva använda programmets egna grafiska gränssnitt. (FEM-Design API, 2024)

Grasshopper-komponenterna som inkluderas i tilläggs paketet replikerar funktioner som finns i inbyggda FEM-Design, till höger i figur 12 nedan är till exempel den komponent som skapar FEM-Design geometri för plattor. De andra komponenterna i figuren nedan bestämmer tjocklek, material och namn för plattan. (FEM-Design API, 2024)



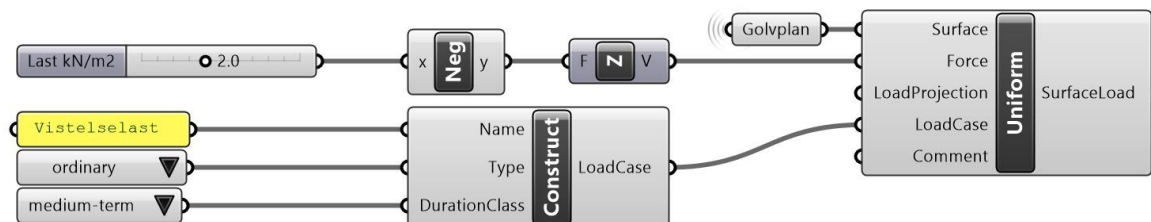
Figur 12. Skript i Grasshopper för att skapa en FEM-Design platta. (Wille Dahlblom, 2024)

Komponenten längst till vänster i figur 12 ovan fungerar som en databas för standardmaterial och deras vanligaste hållfasthetsklasser. Genom att till exempel använda komponenten *list item*, kan man välja vilket material från den valda listan som plattan i detta fall ska ha. En liknande komponent med en databas för tvärsnitt finns också, den används bland annat när man ska skapa balkar eller pelare. Förutom de standardmaterial och -tvärsnitt som finns i dessa två databaser, kan man själv också skapa egna varianter. Utöver att skapa FEM-Design-element, kan man genom att använda API:et också generera laster, göra analyser eller hämta beräkningsresultat. (FEM-Design API, 2024)

Genom de komponenter som skapar FEM-Design-element kan man även bland annat specificera eventuella excentriciteter för de olika konstruktionsdelarna som ska modelleras och bestämma deras anslutningstyper till varandra. Namnen på elementen används för att identifiera dem i modellen och de genereras automatiskt av skriptet. Namnet baseras på vilken typ av konstruktionsdel det är och vilken våning elementet är på, till exempel har ett element fått namnet "Pelare\_3.37" då det är den 37:e pelaren på tredje våningen.

### 4.3 Lastberäkningar i Grasshopper

Eftersom skriptet främst är skapat för att undersöka höga byggnaders stomsystem med fokus på vindlasternas inverkan är de andra eventuella lasterna förenklade ner till två olika ytlaster. Dessa laster verkar på alla de golvytor som modelleras, och de är som standard angivna till  $1,0\text{kN/m}^2$  som permanentlast samt  $2,0\text{ kN/m}^2$  som variabelast i medellång tidsklass. Storleken på lasterna kan i skriptet ändras med hjälp av ett skjutreglage som syns till vänster i figur 13 nedan. I mitten av figuren ser man också komponenten *LoadCase* som skapar ett lastfall samt definierar lasternas typ och i vilken tidsklass de verkar. Egenvikten av de olika konstruktionsdelarna beräknar FEM-Design automatiskt med hjälp av de egenskaper, bland annat material och tjocklek, som man angett när elementen skapats.



Figur 13. Skapandet av en FEM-Design ytlast i Grasshopper (Wille Dahlblom, 2024)

Vindlasten som verkar på byggnaden kan antingen räknas ut automatiskt av Grasshopper-skriptet, eller manuellt definieras i form av en enhetlig ytlast på någon av byggnaden sidor. Oberoende om vindlasten är manuellt eller automatiskt definierad, räknas den först om från en ytlast till en linjelast som verkar på varje våningsplan och sedan omvandlas den till en punktlast i golvytanens tyngdpunkt föra lasten överförs till FEM-Design. Vinkeln på vindlasten går att ändra på parametriskt och formen på byggnaden är inte alltid rektangulär, vilket gör att bredden och djupet av konstruktionen med hänsyn till vindlasten måste räknas ut.

Skriptet skapar genom komponenten *bounding box*, i samma riktning som vindlasten, en rätblocksformad låda som omskriver våningsplanen. Byggnadens bredd vinkelrätt mot vindriktningen kan då fås ut utgående från lådans närmaste sidolängd. Vindlasten som ytlast multipliceras sedan med byggnadens bredd och med höjden på den yta av fasad som varje våning tar upp vind från för att få en resultantkraft som kan överföras till FEM-Design.

Den vindlast som Grasshopper-skriptet kan räkna ut utgår från byggnadens höjd och form, och även om beräkningarna är en del förenklade så baseras de på Eurokod: EN 1991-1-4 samt Boverkets föreskrifter om tillämpning av europeiska konstruktionsstandarder. Från den svenska nationella bilagan användes formel (1) nedan för att räkna det karakteristiska hastighetstrycket samt formel (2) för turbulensintensiteten:

$$q_p(z) = \left[ 1 + 2 \cdot k_p \cdot I_v(z) \right] \cdot \left[ k_r \cdot \ln\left(\frac{z}{z_0}\right) \cdot c_0(z) \right]^2 \cdot q_b \quad (1)$$

$$I_v = \frac{1}{c_0(z) \cdot \ln\left(\frac{z}{z_0}\right)} \quad (2)$$

$$q_b = \frac{1}{2} \cdot \rho \cdot v_b^2 \quad (3)$$

Referenshöjden  $z$  baseras på byggnadens höjd och bredd samt får inte vara lägre än  $z_{min}$ . Värdena för  $z_{min}$  och  $z_0$  varierar beroende på terrängklass och fås ur tabell 1 nedan, terrängfaktorn  $k_r$  baseras också på vilken terrängklass byggnaden befinner sig i. De förenklingar som gjorts är att byggnadens egenfrekvens och omgivningens topografi inte beaktas, vilket gör att  $k_p$  sätts som 3,0 och  $c_0(z)$  blir 1,0 (Boverket, 2022). Formel (3) ovanför för referenshastighetstrycket  $q_b$  fås från Eurokod: EN 1991-1-4 och baserar sig på vindens referenshastighet  $v_b$ , vilken såväl som terrängfaktorn kan ändras parametriskt i skriptet. För vindens densitet  $\rho$  används värdet 1,25 kg/m<sup>3</sup> (Boverket, 2022).

Tabell 1. Terrängkategorier enligt Eurokod EN 1991-1-4

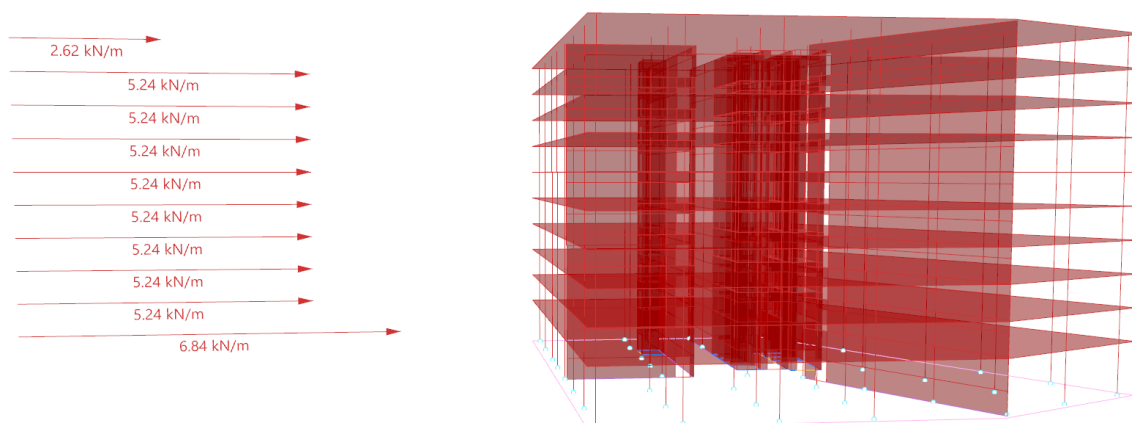
Terrain category	$z_0$ m	$z_{min}$ m
0 Sea or coastal area exposed to the open sea	0,003	1
I Lakes or flat and horizontal area with negligible vegetation and without obstacles	0,01	1
II Area with low vegetation such as grass and isolated obstacles (trees, buildings) with separations of at least 20 obstacle heights	0,05	2
III Area with regular cover of vegetation or buildings or with isolated obstacles with separations of maximum 20 obstacle heights (such as villages, suburban terrain, permanent forest)	0,3	5
IV Area in which at least 15 % of the surface is covered with buildings and their average height exceeds 15 m	1,0	10

NOTE: The terrain categories are illustrated in A.1.

(SFS-EN 1991-1-4+AC+A1, 2011)

Beräkningen av vindlasten i skriptet tar i beaktande hur lastens intensitet förändras med höjden på högre byggnader. Om byggnadens höjd är större än dess bredd, delas vindlasten automatiskt upp i två olika zoner. Referenshöjden  $z$  som finns med i både formel (1) och (2) blir då i enlighet med Eurokod: EN 1991-1-4,  $z_e = b$  för den nedre delen av byggnaden och  $z_e = h$  för den övre.

Nedan i figur 14 har vindlasten via Grasshopper-skriptet valts att visualiseras i Rhino, referensvindhastigheten är 26 m/s och terrängklassen är 0. Byggnaden i figuren är inte högre än vad den är bred vilket gör att vindlasten blir enformig över byggnaden. Det man däremot kan se är att lasten för den nedersta våningen är högre än för de andra på grund av att våningshöjden är större, samt att lasten på den översta våningen är mindre då den endast tar upp vind från hälften av den yta som de övriga våningarna.



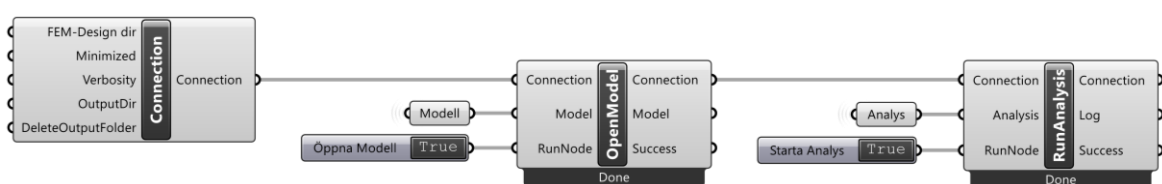
Figur 14. Visualisering av vindlaster för en exempelkonstruktion i Rhino (Wille Dahlblom, 2024)

## 5 Analys i FEM-Design

För att de element, stöd, laster och annat som skapats via Grasshopper-skriptet ska kunna överföras till FEM-Design, sätts de först samman till en modell genom komponenten *construct model*. Efter att modellen skapats kan den analyseras på flera olika sätt, med hjälp av komponenten *RunCalculation* öppnas FEM-Design i bakgrunden och gör analysen. Men när beräkningarna är klara så stängs programmet ner igen och resultaten sparas endast i komponenten i Grasshopper. När detta alternativ körs så fryser det däremot både Grasshopper och Rhino tills resultaten är klara, vilket kan bli problematiskt då man inte vet när programmen kan användas igen. Skriptet som examensarbetet behandlar använder sig därför av en annan metod, där en direktkoppling skapas och modellen öppnas i stället i FEM-Design. Analysen kan då köras både genom en komponent i Grasshopper eller genom den traditionella vägen i FEM-Design.

### 5.1 Överföring till FEM-Design

Komponenten *FEM-Design Connection* som syns i figur 15 nedan, öppnar en direktkoppling från Grasshopper till FEM-Design som det sedan kan byggas vidare på. Man kan till exempel välja att öppna en ny modell eller att modifiera en befintlig, man kan också starta finita element analyser eller hämta resultat från en färdig analys med hjälp av direktkopplingen. Om en analys har satts igång kommer den att göras helt och hållet i FEM-Design vilket gör att både Grasshopper och Rhino kan användas under tiden. Om något i modellen ändras när en analys körs, kommer även komponenterna som öppnar och analyserar modellen att uppdateras. Det här kan leda till att analysen misslyckas och behöver göras om, och därför är det bättre att stänga av både öppningen av modellen och analysen före något ändras. (FEM-Design API, 2024)



Figur 15. Grasshopper-komponenter för direktkoppling till FEM-Design

## 5.2 Beräkningar

FEM-Design kan med hjälp av finita element metoden beräkna bland annat nedböjningar, inre krafter, stödreaktioner och mängder. Programmet kan även göra analyser för till exempel egenfrekvenser, stabilitet och seismiska rörelser. (FEM-Design Wiki, 2022)

När man i Grasshopper-skriptet ställer in vad som ska beräknas i analysen i FEM-Design, har man även möjligheten att välja om analysen ska göras med standard eller med noggrannare finita element. Figur 16 nedan visar ett exempel på skillnaden mellan de två olika elementtyperna för ett fyrkantigt finit element, standardversionen har endast fyra noder men den noggrannare versionen har nio. Resultaten av analysen blir mer exakta om man använder sig av den noggrannare varianten men beräkningstiden kan bli upp till fyra gånger längre jämfört med om standardelement skulle använts. (FEM-Design Wiki, 2022)



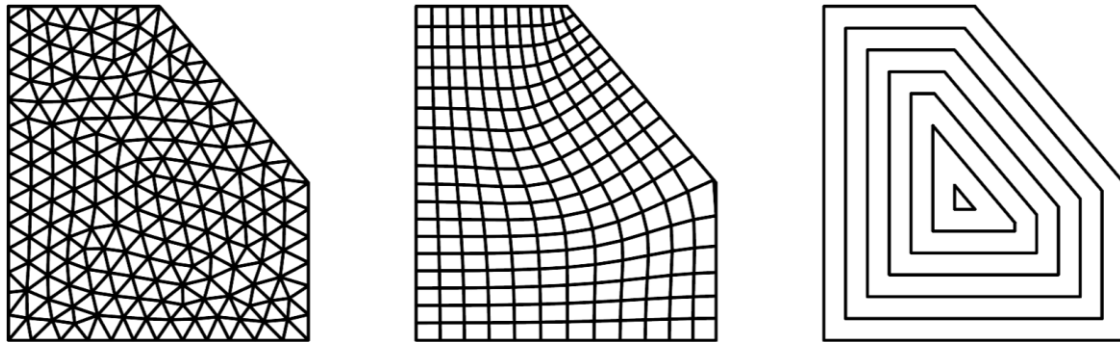
Figur 16. Skillnaden mellan standard och noggranna finita element (Wille Dahlblom, 2024)

För att göra analysen och beräkningarna snabbare och mer förenklade, skapas via Grasshopper-skriptet förutom bland annat konstruktionselement, också något som kallas *Diaphragm*. Ett *Diaphragm* är ett membran som i beräkningarna i FEM-Design antas vara oändligt styvt i det horisontella ledet. Det betyder att alla noder från de genererade finita element näten som är innanför ett av dessa membran rör sig enhetligt vågrätt, men kan förflyttas lodrätt oberoende av varandra. Skriptet skapar ett Diaphragm för varje våning och de får formen från våningsplanens yttre konturlinje. (FEM-Design, 2023)



### 5.3 Test och optimeringar

För att göra beräkningarna i FEM-Design mer effektiva, delades golvytorna i mindre delar i Grasshopper innan de omvandlades till FEM-Design-element. Tanken var att om uppdelningen till mindre underregioner påbörjades redan av Grasshopper-skriptet, i stället för i den automatiska generationen i FEM-Design, så skulle beräkningsprocessen för snabbas. De tre olika metoderna som prövades för att dela upp golvytorna, syns i figur 17 nedan. Först delades ytorna i mindre trianglar, sedan i firsidiga regioner och till sist delades golvytorna genom att golvytornas kant försköts mot mitten.



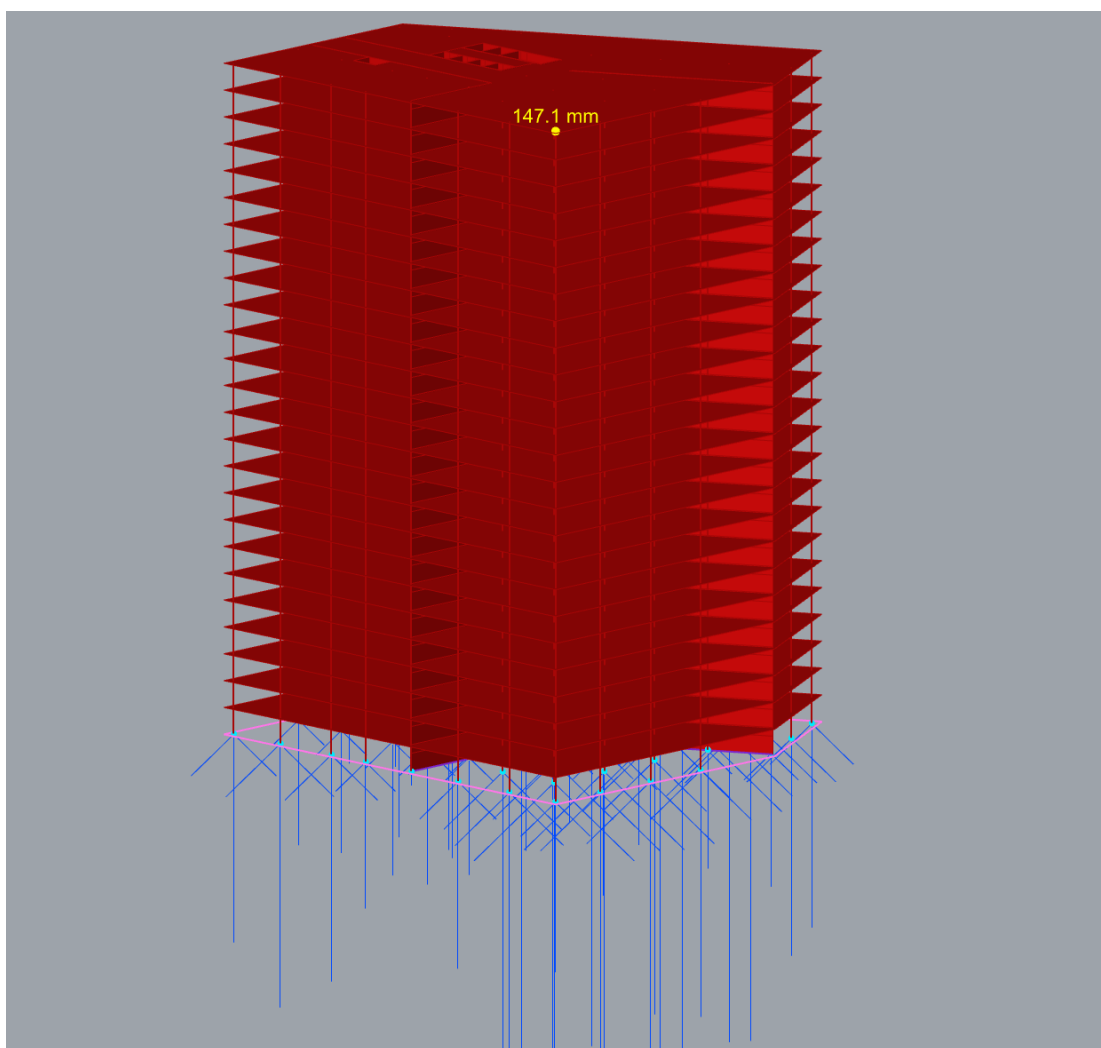
Figur 17. Uppdelningstest av en exempelgeometri i Grasshopper (Wille Dahlblom, 2024)

De olika uppdelningssätten prövades flera gånger för en exempelkonstruktion men det framkom att tidsförbättringen var minimal, eftersom den var ungefär lika stor som skillnaden mellan den snabbaste och långsammaste tiden för testen av den icke uppdelade konstruktionen. För att undvika att göra Grasshopper-skriptet mer avancerat än nödvändigt, användes FEM-Designs egna automatiska finite mesh generation i stället för att färdigt dela in ytorna i mindre regioner.

## 6 Analysresultat

Analysresultaten kan visas antingen i FEM-Design eller så kan man via Grasshopper-skriptet välja att de ska hämtas tillbaka och visualiseras i Rhino i stället. De resultat som kan hämtas via skriptet är stödreaktionerna i både pelare och väggar samt förflyttningen av finita elementnätets noder. Eftersom väggarna har linjestöd kan deras stöd krafter visas som en linjelast eller som en resultant per väggelement.

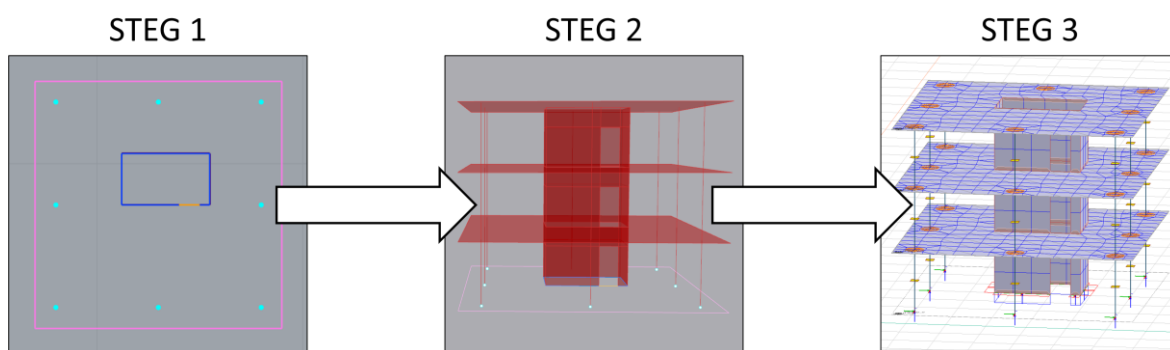
I figur 18 nedan har en exempelkonstruktion med 25 våningar modellerats och analyserats. Grasshopper-skriptet har sedan importerat resultaten av analysen och visualiserar dem i Rhino. De resultat som valts att visas är den maximala horisontella förflyttningen och stödreaktionerna för de bärande pelarna. För att underlätta synligheten i figur 18 nedan, visas inte värdena för pelarnas stöd krafter och inte heller väggarnas stödreaktioner.



Figur 18. Visualisering av analysresultatet från FEM-Design i Rhino

## 7 Sammanfattning och diskussion

Figur 19 nedan sammanfattar huvudfunktionerna för Grasshopper-skriptet som utvecklades för examensarbetet. I steg 1 har en basgeometri ritats upp utgående från hur planlösningen ser ut för den byggnad man vill modellera och analysera. I steg 2 har modellen parametriskt modellerats enligt värdena på olika höjdparametrar som definierats i början av skriptet. Det tredje steget visar modellen efter att den importerats till FEM-Design, delats upp i mindre finita element och sedan analyserats.



Figur 19. Sammanfattning av Grasshopper-skriptets funktioner (Wille Dahlblom, 2024)

Att manuellt rita upp en viss byggnad i FEM-Design eller i något 3D-modelleringsprogram och att sedan göra en analys, kan vara snabbare än att använda sig av det skript och arbetsätt som är utvecklat för examensarbetet. Poängen med projektet var däremot inte att rita upp bara en modell för att sedan nöja sig med den, utan att så enkelt och snabbt som möjligt kunna förändra hela modellen från början till slut. Om den tid som satts ned för att utveckla detta skript räknas med i modelleringstiden för den exempelbyggnad som arbetet behandlar, skulle befintliga tillvägagångssätt varit betydligt mer effektiva. Anledningen till att detta projekt ändå kan anses vara lyckat, är att vid nästa användning kommer skriptet att underlätta den ursprungliga modelleringsprocessen avsevärt.

En basförståelse för Rhino och Grasshopper är något som behövs för att kunna använda sig av det här arbetssättet eftersom Grasshopper-skriptet faktiskt måste öppnas för att man ska få tillgång till alla de parametrar som går att ändra. Skapandet av basgeometrin i Rhino måste också ske i lager med rätt namn för att skriptet skall fungera automatiskt, vilket betyder att en mall med stor sannolikhet behöver användas. Erfarenhet av FEM-Design är också något som kan vara bra att ha för att man skall förstå vilka värden som ska fyllas i och på vilka ställen, till exempel för att analysen ska göras korrekt.

Då skriptet endast beaktar utförbarheten för en stomlösning med tanke på stödreaktioner och förskjutningar, kunde kopplingar till andra viktiga delar av en byggnads tidiga planeringsprocess göras. Kostnad och klimatavtryck är exempel på faktorer som också påverkar beslutsförloppet vid utformningen av en byggnad. Beräkningar för dessa faktorer skulle kunna införas i skriptet eller göras samtidigt i andra program för att skapa en bättre helhetsbild av stomlösningens verkliga potential. Förbättringar av arbetssättet skulle förutom direkta uppgraderingar av Grasshopper-skriptets funktioner och detaljnoggrannhet, också kunna vara en förenkling av gränssnittet för att göra det mer användarvänligt.

## 8 Källförteckning

- Bachman, D. (2017). *Grasshopper: Visual Scripting for Rhinoceros 3D*. Industrial Press, Inc.
- Boverket. (2022). *Boverkets föreskrifter och allmänna råd (2011:10) om tillämpning av europeiska konstruktionsstandarder (eurokoder)*. Hämtat från <https://www.boverket.se/sv/lag--ratt/forfattningssamling/gallande/eks---bfs-201110/> den 11 Mars 2024
- Dynamo. (2023). Hämtat från <https://dynamobim.org/> den 9 Januari 2024
- Eltaweel, A., & SU, Y. (2017). Parametric design and daylighting: A literature review. *Renewable and Sustainable Energy Reviews*, 73, 1086-1103. doi:10.1016/j.rser.2017.02.011
- FEM-Design. (2023). Hämtat från StruSoft: <https://strusoft.com/software/3d-structural-analysis-software-fem-design/> den 5 Februari 2024
- FEM-Design API. (2024). Hämtat från <https://femdesign-api-docs.onstrusoft.com/docs/grasshopper/get-started> den 23 Mars 2024
- FEM-Design Wiki. (2022). Hämtat från <https://wiki.fem-design.strusoft.com/xwiki/bin/view/Main/> den 12 Mars 2024
- Parametric modelling. (2021). (Cohesive) Hämtat från BIM Wiki: [https://www.designingbuildings.co.uk/wiki/Parametric\\_modelling](https://www.designingbuildings.co.uk/wiki/Parametric_modelling) den 9 Januari 2023
- Pavel, A. (2022). *Parametric modelling: Grasshopper basics tutorial*. LAB University of Applied Sciences, Bachelor of Civil and Construction Engineering. Hämtat från <https://urn.fi/URN:NBN:fi:amk-2022111622818> den 16 Januari 2024
- Rao, S. S. (2017). *The Finite Element Method in Engineering* (Vol. 6). Butterworth-Heinemann.
- Rhinoceros 3D. (2023). Hämtat från <https://www.rhino3d.com/> den 9 Januari 2024
- SFS-EN 1991-1-4+AC+A1. (2011). *Eurocode 1: Actions on structures. Part 1-4: General actions. Wind actions*. SUOMEN STANDARDISOIMISLIITTO SFS. Hämtat den 10 Mars 2024
- Sheperd, P., & Hudson, R. (2007). Parametric Definition of Lansdowne Road Stadium. *IASS Symposium 2007 - Structural Architecture, Venice*.