ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

A UTILITY PROGRAM FOR SHIP HULL FORM MANIPULATION AND RESISTANCE PREDICTION FOR INITIAL DESIGN STEP

M.Sc. THESIS

Çağan Birant PEKKÜÇÜK

Department of Naval Architecture and Marine Engineering

Naval Architecture and Marine Engineering Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

A UTILITY PROGRAM FOR SHIP HULL FORM MANIPULATION AND RESISTANCE PREDICTION FOR INITIAL DESIGN STEP

M.Sc. THESIS

Çağan Birant PEKKÜÇÜK
(508151006)

Department of Naval Architecture and Marine Engineering

Naval Architecture and Marine Engineering Programme

Thesis Supervisor: Assoc.Prof. Devrim Bülent DANIŞMAN

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BAŞLANGIÇ TASARIMI EVRESİ İÇİN GEMİ TEKNE YÜZEYİ İŞLEME VE DİRENÇ TAHMİNİ PROGRAMI

YÜKSEK LİSANS TEZİ

Çağan Birant PEKKÜÇÜK
(508151006)

Gemi İnşaatı ve Gemi Makinaları Anabilim Dalı

Gemi İnşaatı ve Gemi Makinaları Programı

Tez Danışmanı: Doç.Dr. Devrim Bülent DANIŞMAN

HAZİRAN 2019

Çağan Birant PEKKÜÇÜK, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 508151006 successfully defended the thesis entitled "A UTILITY PROGRAM FOR SHIP HULL FORM MANIPULATION AND RESISTANCE PREDICTION FOR INITIAL DESIGN STEP", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc.Prof. Devrim Bülent DANIŞMAN** Istanbul Technical University	....
Jury Members :	**Assoc. Prof. Devrim Bülent DANIŞMAN** Istanbul Technical University	....
	Assoc. Prof. Erdem ÜÇER Istanbul Technical University	....
	Dr. Lect. Ziya SAYDAM Piri Reis University	....

Date of Submission : 13 May 2019
Date of Defense : 13 June 2019

FOREWORD

First and foremost, I would like to express my sincerest gratitudes to Prof. Devrim Bülent DANIŞMAN for supporting me in the area i prefer to study. His guidance and knowledge helped me discover many ways of building and developing the technical background of this study and also let me improve my scientific perspective of seeing the nature.

I wish to express my appreciation to Assist. Prof. Dr. Serdar Aytekin KÖROĞLU for helping me explore the main software by using which i have completed this thesis study. I would like to express special thanks to Prof. Dr. Ömer GÖREN, Assoc. Prof. Dr. Ertekin BAYRAKTARKATAL, Assoc. Prof. Dr. Bahadır UĞURLU, Assoc. Prof. Dr. Şebnem HELVACIOĞLU, Assist. Prof. Dr. Yalçın ÜNSAN and Prof. Dr. Osman Azmi ÖZSOYSAL for their patience, desire to teach and endless support, which lead me keep on improving myself in scientific areas.

My special thanks go to İbrahim CENGİZ, Efe SÜTMAN, Hakan ÖZKUL, Mete ULUTAŞ and Erkay AÇIKGÖZ for their friendship and making me feel surrounded by people like myself, during my educational life. I would like to thank Can BOLEL, Çağatay Onur ŞENGÖR, Taha YILDIRIM, Mahmoud Abo ALABAs and Buğra Uğur YAZICI for their technical and mental support in my times of need. I also thank Levent TATLIDİL and Emre YEŞİL for their support during my educational life.

Finally my deepest thanks go to my father Yalçın PEKKÜÇÜK and mother Türkan PEKKÜÇÜK for their belief in me, endless mental and financial support, and love that provided me the space to discover science and nature in my childhood. Also my deepest thanks go to Makbule Gizem KIREVLİYASI for her endless support on every aspect of life and her love which has always felt special and encouraged me on my studies.

May 2019 — Çağan Birant PEKKÜÇÜK
(Naval Architect and Marine Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

CFD **:** Computational Fluid Dynamics
LoA **:** Length Over All
SAC **:** Sectional Area Curve

LIST OF TABLES

Page

LIST OF FIGURES

Page

A UTILITY PROGRAM FOR SHIP HULL FORM MANIPULATION AND RESISTANCE PREDICTION FOR INITIAL DESIGN STEP

SUMMARY

As many differing transportation options, marine transportation has been a vital and growing way of transportation since the earliest ages of humanity. Mathematicians and engineers have been studying to enhance and develop better, more efficient, more comfortable and more trustworthy designs to obtain the optimum balance for varying purpose floating veichles.

Since ship design has many subbranches such as hydrodynamic design, structural design, stability design and etc, more than one team or department have to exhibit an interdiciplinary teamwork to overcome this complex task. Thus, numerous studies have been made until today to analyze the most complicated problems faced during the designation of varying parts of floating veichles and to provide solutions based on reasonably acceptable ways by mathematicians and engineers who have specialized on differing branches.

Speaking of underwater form design and optimization of a vessel, hydrodynamic properties are the foregrounding features. Hydrodynamic performance of a vessel affects the adaptation of its hull form with water and thereby it directly influences the resistance amount faced during the operation, efficient machinery power and suitable propeller design to maintain the desired cruise speed, fuel consumption and maneouvrability characteristics. Since these features mostly represent the economical properties of a ship not during its building process but especially its service life, hydrodynamic design is one of the most important parts of ship design when it is also considered that any ship owner would not prefer a vessel which is not cost-effective.

There have been many groundbreaking solutions provided to the problems faced during many hull form designs for various ship types until today. Bulbous bows are one of the examples to these solutions, which create an opposing wave pattern to the bow waves that occur at the fore part of the ship while operational and absorbs them. Thus it lowers the total amount of wave resistance that the ship faces and correspondingly provides higher cruise speeds and lower fuel consumptions at the design speed.

There are also many different underwater shapes for varying purpose of vessels. For instance a VLCC has a fuller mid-section and hull form to provide a large cargo space and cruise speed optimization where a naval ship has a thinner hull form compared to a VLCC to maintain higher cruise speeds and maneouvrability. For all types of ships, notwithstanding to their specific hull shapes, the balance between requirements and performance has to be met, which puts the importance of hydrodynamic design on the table.

There are many softwares used by companies according to their specific needs for hydrodynamic design and hull form optimization. Rhinoceros and Napa are the examples for surface modelling at initial design, where Maxsurf resistance, Ansys fluent and OpenFOAM are some of the examples for resistance calculation step. Also there are many CFD tools used for hydrodynamic optimization, mostly for reducing calm-water drag and wave patterns, by which many studies have been made until today and can be found in literature. Since building and editing a hull surface within some certain constraints, especially manual editing, is a complex work all by itself, the use of automated software is a growing tendency by designers.

In this study a script has been developed by using Grasshopper plug-in for Rhinoceros, which automates some part of the hull surface editing process. The script is also linked with a resistance prediction program called Michell.exe and together they provide automated surface editing and resistance prediction as an initial design utility program. Three different surface manipulation procedure has been applied to a single hull form with this program and resistance prediction values have been compared with the original form. Comments have been made on the obtained results.

BAŞLANGIÇ TASARIMI EVRESİ İÇİN GEMİ TEKNE YÜZEYİ İŞLEME VE DİRENÇ TAHMİNİ PROGRAMI

ÖZET

Deniz ulaşımı en eski çağlarından beri gelişmekte olan önemli bir ulaşım türüdür. Çeşitli amaçlar doğrultusunda üretilen yüzer araçlarda optimum dengeyi yakalamak adına ve daha verimli, daha konforlu ve daha emniyetli tasarımlar ortaya koyabilmek için matematikçiler ve mühendisler tarafından yapılmış bir çok çalışma bulunmaktadır.

Gemi tasarımı hidrodinamik, yapısal analiz ve stabilite gibi birçok alt başlığa sahip olduğundan ötürü bu karmaşık yapının tasarımının tamamlanmasında birden fazla takım veya departmanın disiplinlerarası bir takım çalışması ortaya koyması gerekmektedir. Bu yüzden yüzer taşıtların çeşitli kısımlarının tasarımında karşılaşılan en karmaşık sorunların analizi ve bu sorunlara makul kabuller içeren çözümler üretebilmek adına günümüze kadar sayısız çalışmalar yapılmıştır.

Bir geminin su altı formunun tasarımı ve optimizasyonu söz konusu olduğunda hidrodinamik özellikler ön plana çıkmaktadır. Bir geminin hidrodinamik özellikleri tekne formunun suya adaptasyonunu etkiler ve böylece operasyonel durumda karşılaştığı direnç miktarı, efektif makina gücü ve istenilen seyir hızını sağlamak için gerekli olan pervane tasarımı, yakıt tüketimi ve manevra karakteristiği gibi konularla ilişkilenir. Bu özellikler geminin inşaa sürecinden ziyade servis ömrü boyunca sahip olacağı ekonomik özelliklerini temsil ettiğinden ötürü, hiçbir armatörün ekonomik olmayan bir gemiyi tercih etmeyeceği de göz önünde bulundurulduğunda hidrodinamik dizaynın gemi tasarımındaki en önemli başlıklardan biri olduğu anlaşılabilmektedir.

Günümüze dek farklı gemi türlerinin tekne formlarının tasarımında karşılaşılan problemlere bir çok çığır açan çözüm üretilmiştir. Bu çözümlere verilebilecek örneklerden biri balblı baştır. Balblar gemi baş formunun akımla karşılaştığı esnada ortaya çıkan baş dalgalarına zıt dalgalar üreterek bu dalgaların sönümlenmesini sağlarlar. Böylece geminin karşılaştığı toplam dalga direnci düşer ve buna bağlı olarak daha yüksek seyir hızlarına ulaşılır ve daha düşük yakıt tüketimi sağlanır.

Farklı türlerdeki gemilerin su altı şekillerinde de çeşitlilik görülmektedir. Örneğin bir ham petrol tankerinin yüksek kargo kapasitesi ve optimum hızı sağlamak için dolgun bir orta kesiti bulunurken yüksek seyir hızı ve yüksek manevra kabiliyetine ihtiyaç duyan savaş gemilerinin orta kesitleri ham petrol tankerlerine kıyasla daha narin olmaktadır. Her tür gemi için kendilerine has tekne geometrilerinden bağımsız olacak şekilde gereksinim ve performans dengesi sağlanmalıdır. Bu durum da hidrodinamik tasarımın önemini vurgular niteliktedir.

Hidrodinamik tasarım ve tekne formu optimizasyonunda şirketlerin kendi ihtiyaçları doğrultusunda kullandıkları çeşitli yazılımlar mevcuttur. Başlangıç tasarım süreci yüzey modelleme işlemleri için Rhinoceros ve Napa başlıca örneklerdendir. Direnç

analizi aşaması için ise Maxsurf resistance, Ansys fluent ve OpenFOAM yazılımları örnek verilebilir. Ayrıca çoğunlukla sakin su direncini düşürme ve dalga düzenlerinin analizinde kullanılan hidrodinamik optimizasyon odaklı bir çok CFD aracı bulunmaktadır. Bu araçların geliştirilmesi ve kullanılması üzerine yapılmış bir çok çalışma literatürde yer almaktadır. Belirli kısıtlar çerçevesinde bir tekne yüzeyi oluşturmak ve düzenlemek karmaşık bir işlem olduğundan ötürü tasarımcılar bu süreci otomatik hale getirebilen yazılımları kullanmaya eğilim göstermektedirler.

Bu çalışmada Rhinoceros programının Grasshopper eklentisi kullanılarak tekne yüzeyi düzenleme işleminin belirli bir kısmını otomatik hale getiren bir kod geliştirilmiştir. Kod aynı zamanda Michell.exe adında bir direnç tahmini programıyla bağlantılı hale getirilmiştir ve birlikte başlangıç tasarım aşamasında faydalı olabilecek bir otomatik yüzey düzenleme ve direnç hesaplama programı oluşturulmuştur.

Yaygın şekilde kullanılan bir program olması, eğimli objelerle çalışma konusunda tutarlı bir performans sunabilmesi ve kullanılabilirliği sebebiyetiyle bu çalışmada Rhinoceros yazılımı tercih edilmiştir. Grasshopper, Rhinoceros programına parametrik tasarım özelliği katan bir eklentidir. Rhinoceros ortamında yapılabilen işlemlerin büyük bir kısmı Grasshopper ortamında da yapılabilmektedir. Aynı zamanda iki yazılımın birçok komutu aynı veya benzerdir.

Programlar Grasshopper komutlarının sunulan alana sürüklenmesi ve birbirleri arasında bağlantı kurulması şeklinde oluşturulur. Komutların girdi kısımlarına istenen bir geometri, sayı veya bir sayı kaydıracı bağlanabilmektedir. Sayı kaydırıcıları değişken bir girdiyi temsil eder ve limitleri ile hassasiyeti ayarlanabilmektedir. Bu sayede tamamlanmış bir tasarım üzerinde sonradan yapılmak istenen değişiklikler kolaylıkla yapılabilir. Ayrıca Grasshopper ortamında oluşturulmuş bir tasarım Rhinoceros ortamına kolaylıkla aktarılabilir.

Grasshopper komutları matematik, diziler, vektör, eğri, yüzey, kesişim, dönüşüm gibi ana başlıklar altında toplanmıştır. İsteğe bağlı olarak programa dahil edilebilen farklı komut grupları da bulunmaktadır. Üç boyutlu modelleme dışında çeşitli disiplinlere yönelik fonksiyonları da bünyesinde barındırdığı için yapısal mühendislik, mimarlık ve sanat gibi alanlarda da kullanılabilmektedir.

Oluşturulmuş olan program ile üzerinde çalışılmak istenen tekne formu Grasshopper'a tanımlanır. Form üzerinde düzensiz ölçeklendirmek işlemi yapılır ve ana boyutlar analiz edilir. Sonraki aşamada ise kullanıcıya kesit değişimi ile yüzey manipülasyonu imkanı sunan kod bileşeni bulunmaktadır. Bu bileşenin ardından sırasıyla lineer boy verme ve direnç tahmini bileşenleri yer almaktadır.

Direnç tahmini aşamasında nihai tekne formunun kullanıcının belirlediği sayıda su hattı ve posta sayısına göre yarı genişlik değerleri analiz edilir ve bu değerler kendi içlerinde belirli gruplar haline getirilip düzenlenir. Bu düzenleme işlemi sonrasında Michell.exe programının girdi dosyalarının formatına uygun hale getirilmiş değerler ilgili dosyalara aktarılır. Programın sunduğu tüm özellikler tek bir tasarım üzerinde aynı anda kullanılabileceği gibi istenen bir form üzerinde sadece tek bir fonksiyonu da uygulanabilir.

Bu program ile bir tekne formu üzerinde üç farklı yüzey düzenleme işlemi gerçekleştirilmiştir. Yapılan çalışmada düzensiz ölçeklendirme, kesit değişimi ile yüzey işleme ve lineer boy verme işlemleri tek bir tekne formu üzerinde ayrı ayrı uygulanmıştır. Orijinal formun ve düzenleme işlemleri sonrasında elde edilen formların direnç değerleri hesaplanmıştır. Elde edilen direnç değerleri orijinal formun direnç değerleriyle kıyaslanmış ve çalışmanın sonuçları doğrultusunda yorumlar yapılmıştır.

1. INTRODUCTION

Decisions on ship properties are made with little information on ship hull geometry during preliminary stages of ship design. [1] A succesful coordination between different disciplines, with the objective of creating a valuable and optimum design is required for ship design, which is not a trivial activity. To reach a final design that balances all considerations and satisfies all constraints, a sequential process has to be followed for the design work, in which the details are increased by each step [13].

Hydrodynamic properties of a hull form directly influences the resistance, speed and effective power requirements of the ship, which is related with financial properties of a vessel during its service life. However the seakeeping, stability and maneouvering characteristics of the ship are also highly dependent on its hull form, which are mostly related with ergonomic and safety features of it. Considering all of these vital characteristics of a ship being directly related to its underwater shape, it is inevitable that design and optimization of a ship hull form requires working by means of many different disciplines and aspects to provide a trustworthy, economic and ergonomic design.

Regarding objective functions, a number of requirements, including calm-water drag, speed, loads and motions, payload, propulsion and wakes need to be included in hydrodynamic optimizaton. Since minimization of ship motions without regard to drag, or of drag without regard to seakeeping characteristics, can lead to unaccaptable designs, an integrated approach that simultaneously considers motions and calm-water drag should be used. However, the much simpler task of minimizing objective functions associated with calm-water flow features is considered in all the CFD applications to hull form optimization [7].

When it comes to Computer Aided Design (CAD), there are many software which are specialized at a wide range of varying tasks. Some of the examples are CATIA of Dassault Systemés, Solidworks, Rhinoceros and Friendship Systemes. Rhinoceros is used more often at the design of smaller floating veichles such as powerboats and sailing yachts, since they provide a more flexible work space for designers at the marine industry. Providing many plug-ins such as Orca-3D, Nemo and Expressmarine, also makes Rhinoceros a choice at shape design and optimization areas considering its practicability. Moreover, with the Grasshopper Plug-in it is also possible to create algorithmic modellings and automating the editing process.

Speaking of resistance prediction, CFD tools are being used more and more extensively in practical ship hull form design today. Viscous-flow computations are used to provide detailed predictions of the flow around the hull, the wake field, occurence of flow separation, for appendage alignment, etc. Based on the predictions and the insight obtained the hull form design may be modified to improve the flow quality, prior to any model testing. This CFD use already has enabled significant advances in design.

In this study a Grasshopper code has been developed and linked with a resistance prediction program to automate the shape editing process of an initial hull form. Also the changes in resistance values have been verified with CFD applications, comments and suggestions have been made on the results.

1.1 Aim of the Study

As mentioned before, at the initial steps of hull form design, various requirements have to be considered and all of the parameters have to be adjusted to obtain a balanced design. Since the hull form affects many other principals of a vessel, too many reworks and editings are done at the beginning phase of design, which of course lead to large amount of time loss. Engineers and designers tend to minimize this time loss and thus they need practical and wieldy tools during initial steps of hull form design.

This study aims to provide a visualized utility program by combining features of Rhinoceros, Grasshopper and Michell.exe, which allows users to easily import and manipulate their initial hull form by some certain aspects and track the change of the hull form resistance instantaneously while applying changes to parameters of it. Program requires the underwater hull form and cruise speed as inputs. All of the manipulations are done by moving the number sliders provided in Grasshopper. Resistance prediction is made with a program called “michell.exe”. Thus the 3D editing process is automated and dramatically expedited, errors are made easier to detect and resistance prediction become a one-click process.

Instead of changing the shape of the bulb and the most aft part of the hull form, program provides changes around the mid-body. The cause of excluding the bow and aft form is the complicate being of a bulbous bow. Since the program does not provide creating a hull form just by integrating the main dimensions and criterias, including all kinds of manipulations lead to a more complex and unstable design. Also the aim of coding the program in a way that will work with all types of hull forms leaves a very low margin to the possible bugs and errors. Therefore it is decided to add the other fundamental surface manipulation components as a different plug-in, as future work.

2. LITERATURE REVIEW

Any ship design involves optimization, as competing requirements and design parameters force the design to evolve, and as designers strive to deliver the most effective and efficient platform possible within the constraints of time, budget, and performance requirements. A significant number of applications of computational fluid dynamics (CFD) tools to hydrodynamic optimization, mostly for reducing calm-water drag and wave patterns, demonstrate a growing interest in optimization, see Figure 2.1 [6].

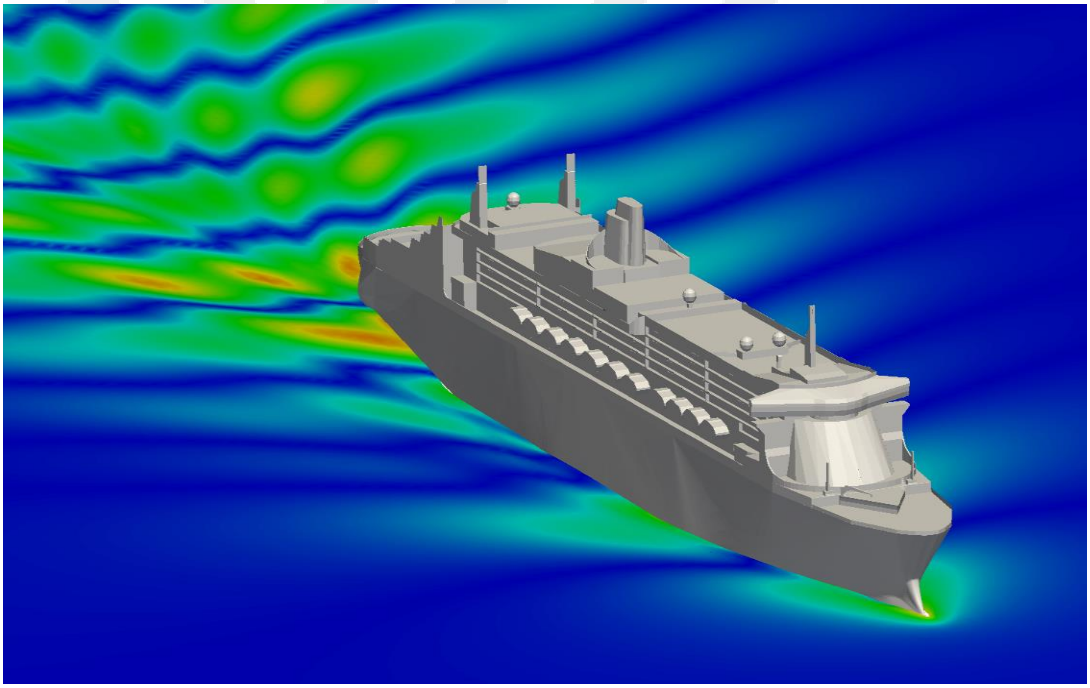


Figure 2.1 : A CFD model with Helyx Marine [5].

It is widely known that the decisions made in the earliest stages of design can have the greatest effect on the through life costs of a vessel. However, most computer-based ship design software focuses on the detailed phases of design related to structure and production. In the Figure 2.2 an example of structural analysis is shown. To investigate and develop innovative solutions, the designer requires a tool that does not enforce detailed definition and allows easy reconfiguration of arrangements and systems. In addition, many calculation exercises may be automated allowing the designer to spend more time focusing on the solution [15]. This tendency on speeding up the process led engineers to work on the complex parts of the ship and trying to automate and parametrize these parts seperately.

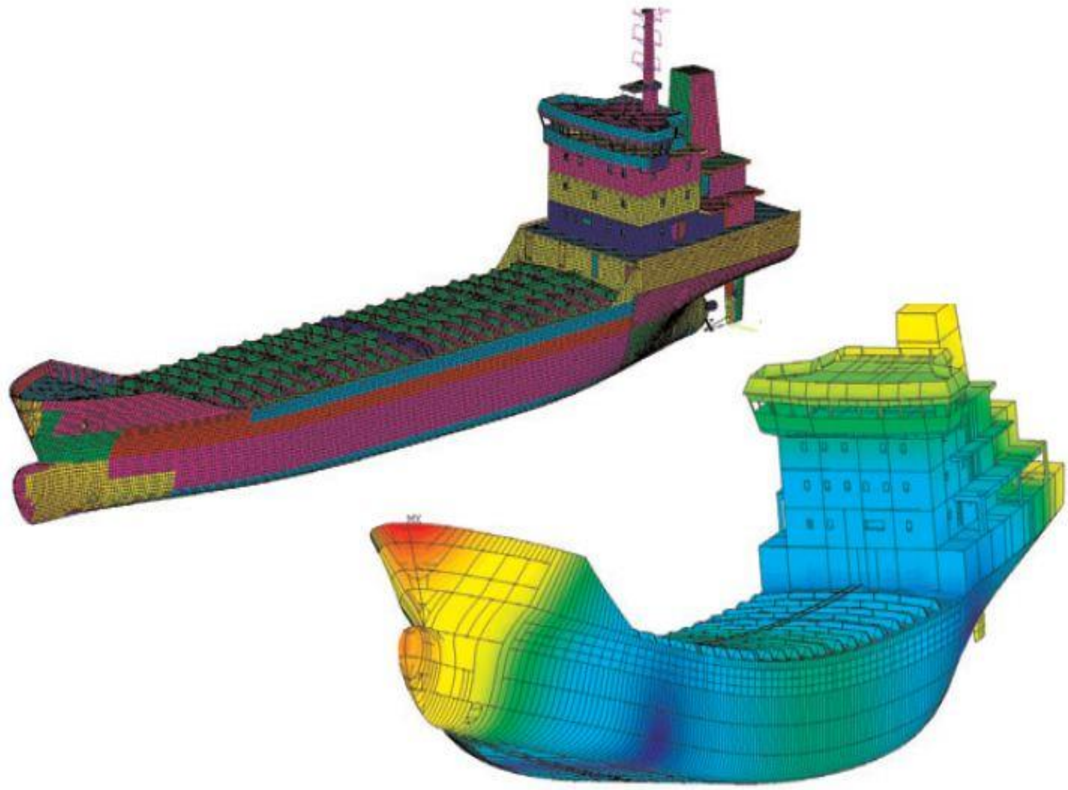


Figure 2.2 : Ship hull plating thickness properties and deflection from rule wave calculation [17].

In ship design, the fully-parametric design was first used by Nowacki et al (1977), who started to model hull curves by means of a cubic B-Spline with seven vertices on the basis of 14 form parameters. Since some parts of the hull are more difficult to analyse, characterize and generate, some particular studies have been made, such as the work of Kracht (1978) focused on bulbous bows in which are presented a set of form coefficients to characterize the bulb, see Figure 2.3. He concluded that the bulb section directly influences the hydrodynammic properties, and affects both the vertical volume distribution and the amplitude of the bulb waves. The bulb section can be classified int three main types, according to the position of its centroid. It was concluded that the volumetric parameter has the largest influence on the wave making resistance and on the phase lag of the longitudinal volume distribution of the bulb area [13].

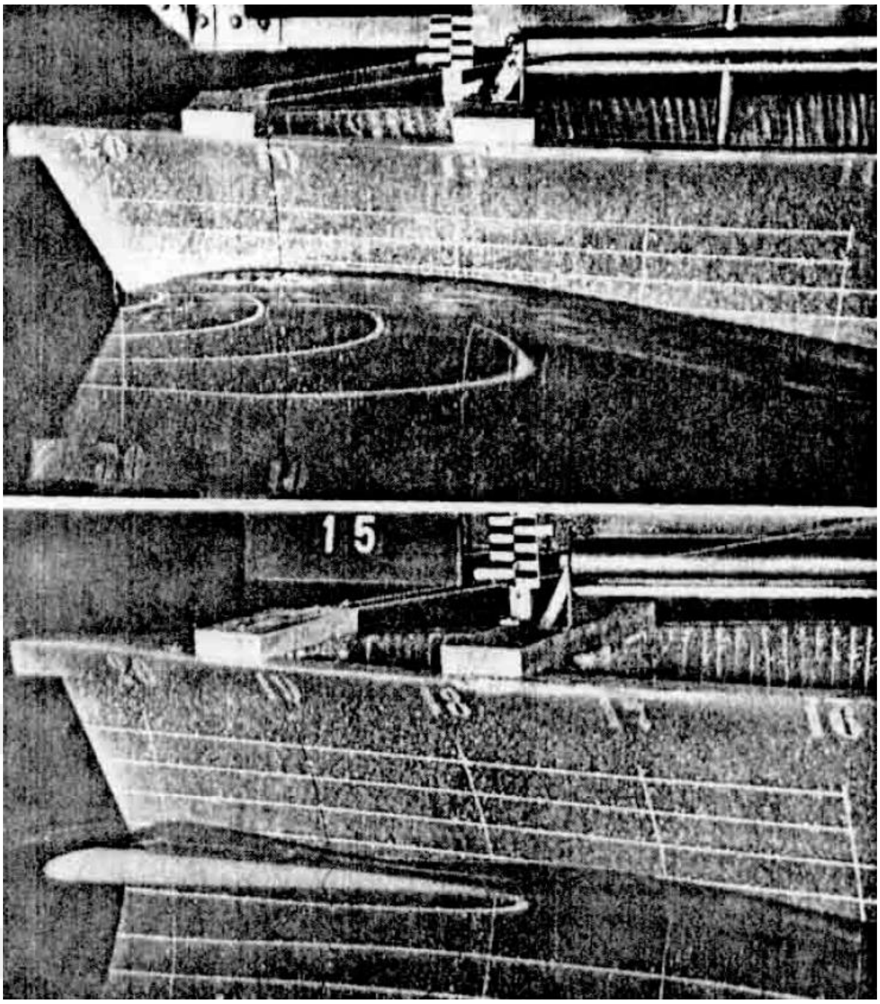


Figure 2.3 : Bow wave pattern of a model with (lower picture) and without bulbous bow (upper picture) [14].

Over the years some research work has been carried out aiming the hull parametric modelling. Jacobsen and Kracht (1992) that presented a new model-series, called D-series, originating from a twin screw round-bilge hull form. This model uses some parameters previously defined by Kracht and some other new parameters. They also presented a list of the twelve longitudinal curves, six primary and six secondary, that, according to them, could fully represent the hull form. Most of the primary basic curves were created with three segments: two curved segments representing the run and the entrance body, and a linear segment representing the parallel midbody. When the length of the parallel midbody is zero, this last linear segment disappears [13].

Harries and Abt (1997) presented a set up to 24 parameters to defined the SAC, containing the important data as the displacement, position of maximum section, centres of buoyancy and slopes at the aft and forward perpendiculars. For the primary curves previously presented, a set up to 13 parameters, representing positional, integral and differential shape requirements, was defined. With this set of form-parameters, the planar curve is more flexible and able to adopt almost any shape requested by the designer on the basis of the geometric properties [13].

Not as a scientific study but as an open source code, the Grasshopper Parametric Ship Hull Modelling Script has been published in 2015, by using which a test hull model has been generated to use in this study, see Figure 2.4. Two published scripts provide modelling tools that are capable of model a bulbous bow and pram aft body type hull. Both solutions are based on Rapid Hull Modeling Methodology which is the manual way to design a ship hull in Rhinoceros. First solution is only a script which contains a fully parametrized hull in which also the loft curves are scripted. Only opening a new session in Rhinoceros and activating the script is required to use it. The second solution is a combination of a Rhinoceros file that contains the loft curves as well as a script that contains the rest of the solution. This enables manually editing or replacing the loft curves, which are the input of the script, and creating new ship type solutions easily.

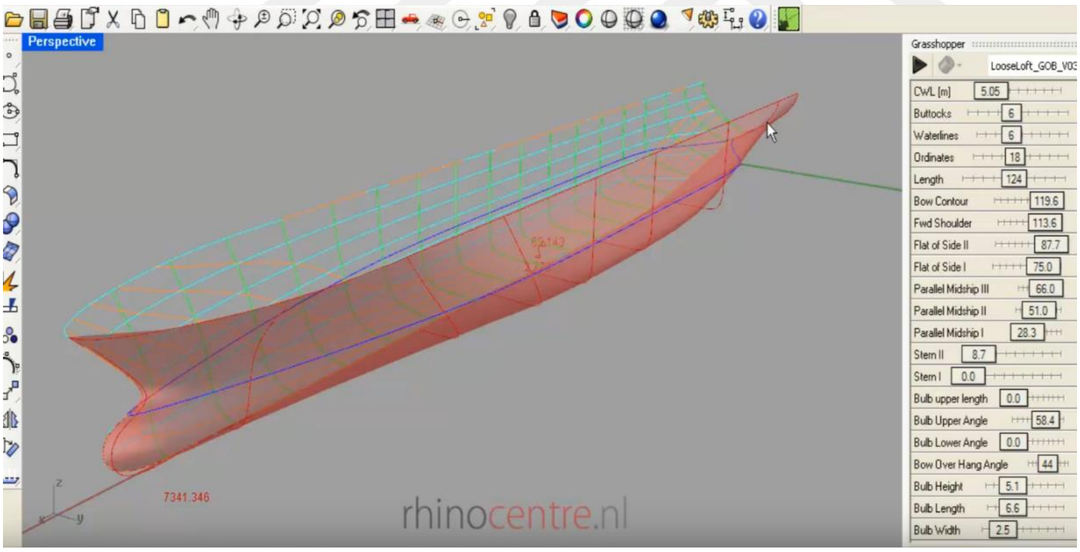


Figure 2.4 : A perspective view of ship hull modelled by Parametric Ship Hull Modelling Script [18].

3. GRASSHOPPER

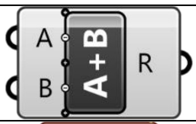
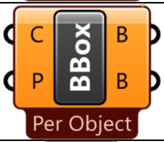
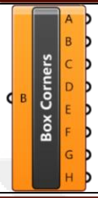
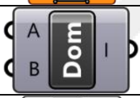
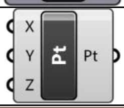
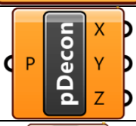
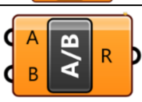
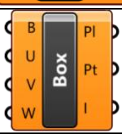
Grasshopper is a visual programming language and environment developed by David Rutten at Robert McNeel & Associates, that runs within the Rhinoceros 3D computer aided design (CAD) application. Programs are created by dragging components onto a canvas. The outputs to these components are then connected to the inputs of subsequent components.

The first version of Grasshopper was released in September 2007, and titled Explicit History. Grasshopper has become part of the standard Rhino toolset in Rhino 6.0 and later.

Grasshopper is primarily used to build generative algorithms, such as for generative art. Many of Grasshopper's components create 3D geometry. Programs may also contain other types of algorithms incuding numeric, textual, audio-visual and haptic applications. Advanced uses of Grasshopper include parametric modelling for structural engineering, parametric modelling for architecture and fabrication, lighting performance analysis for eco-friendly architecture and building energy consumption.

Grasshopper commands are generally wieldy and easily understandable. A command's brief explanation is shown when the cursor is moved over its symbol or name, which is also valid for explanations of input and outputs of a command. In the table 3.1, figures, names and funcionalities of most frequently used Grasshopper commands are included.

Table 3.1 : Explanations of most frequently used Grasshopper commands.

	Addition	Applies a mathematical addition of two values that are connected or set to A and B inputs.
	Boundary Surfaces	Creates planar surfaces from a collection of boundary edge curves that are connected or set to Edges input
	Bounding Box	Solves oriented geometry boundary boxes. Geometries are connected to Content input and base planes are connected to Plane input.
	Box Corners	Extracts all 8 corners of a box, which is connected to Box input.
	Brep / Brep	Solves intersection events for two breps, which are connected or set to A and B inputs.
	Brep Join	Joins a number of breps together, which are connected or set to Brep input.
	Clean Tree	Removes all null, invalid or empty values from a data tree. Data tree is connected to Tree input and True or False is set to Null, Invalid and Empty inputs to obtain desired removal process.
	Construct Domain	Creates a numeric domain from two numeric extremes which are connected to A and B inputs.
	Construct Point	Constructs a point from x y and z coordinates whose corresponding values are connected to x y and z inputs.
	Curve	Contains a collection of curves which are connected to left side input of the command.
	Deconstruct	Deconstructs a point into its component parts. Points are connected or set to Point input and components are obtained from x y and z outputs.
	Deconstruct Box	Deconstructs a box into its constitutent parts. Box is connected or set to Box input and plane, x y and z dimensions are obtained by corresponding outputs of the command.
	Division	Applies a mathematical division to two numbers. Number to divide is connected or set to A input while the divisor number is connected or set to B input.
	End Points	Extracts the end points of a curve. Curve is connected to curve input and its start and end points are obtained by corresponding outputs of the command.
	Evaluate Box	Evaluates a box in 10ormal10c10y UVW space. Box is connected or set to Base box input. U, V and W parameters are connected or set to corresponding inputs. Base plane, point and Inclusion data are obtained by corresponding outputs.

4. DETAILED EXPLANATION OF THE DEVELOPED CODE

In this section the command sequence of the grasshopper code will be explained in detail. Complete code is able to scale the initial hull form non-uniformally, manipulate the surface by selecting three sections that can be edited by four selective points over them and extending the hull form by one selective section linearly.

The Grasshopper code requires the right half of the hull form relevant to the bow-stern front look angle. Also the centerline should be set parallel to the x axis of rhinoceros while bow part of the hull surface should face the positive x direction and the form should not be a multi-surface. Only one surface can be edited at a time.

4.1 Surface Importation and Orientation

After the previous conditions are met, the surface can be imported to the Grasshopper by right clicking on the first surface command of the code and clicking the "set one surface" command, as can be seen in Figure 4.1. After this part, the "surface orientation" sub-code is applied to the hull form first, which moves the initial hull form to the origin point of Rhinoceros by taking the starting point of hull's centerline as reference.

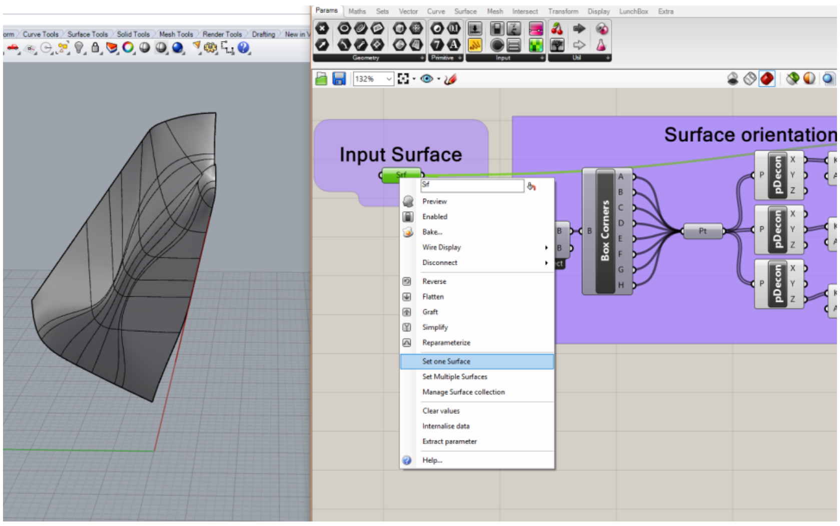


Figure 4.1 : Importing the hull form to surface input in Grasshopper.

Main surface is connected to a Bounding Box command which simply sets the minimum sized limiting outer box to a given surface's limit length over the axis of three coordinates. Then the top-side output of bounding box which exports the aligned bounding box in world coordinates is connected to a Box Corners command. This command lets user obtain the corner points of the bounding box. These point coordinates are then connected to a Point command by which the next commands can recognize these points as a point cloud contained within a single source. This point command is then connects to three Deconstruct commands, which extracts x y and z axis coordinates of a given point. All three Deconstruct commands are connected to three Sort List commands seperately by their x y and z outputs matching with the top to bottom array of their placements as each of them is only connected by one axis value of their outputs, as can be seen in Figure 4.2.

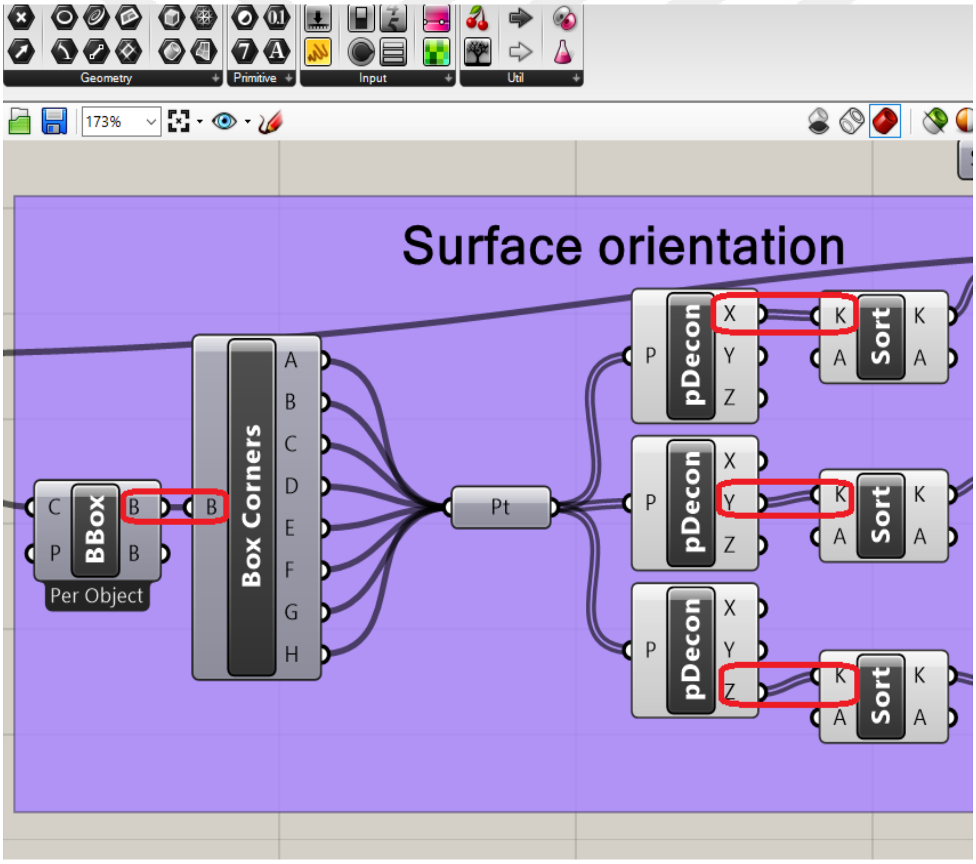


Figure 4.2 : Command sequence of obtaining the position of imported hull form relevant to Rhinoceros origin.

Sort List commands basically sort the corresponding axis values from smallest to largest and sends the information to three List Item commands whose function is picking a single data from a given list of values. Since the reference point from which the main surface will be moved on to the origin point is the one with smallest x y and z values, all the index values of List Item commands are set to zero as can be seen in Figure 4.3, which represent the first number of a given list and in this case it recalls the smallest value of each list.

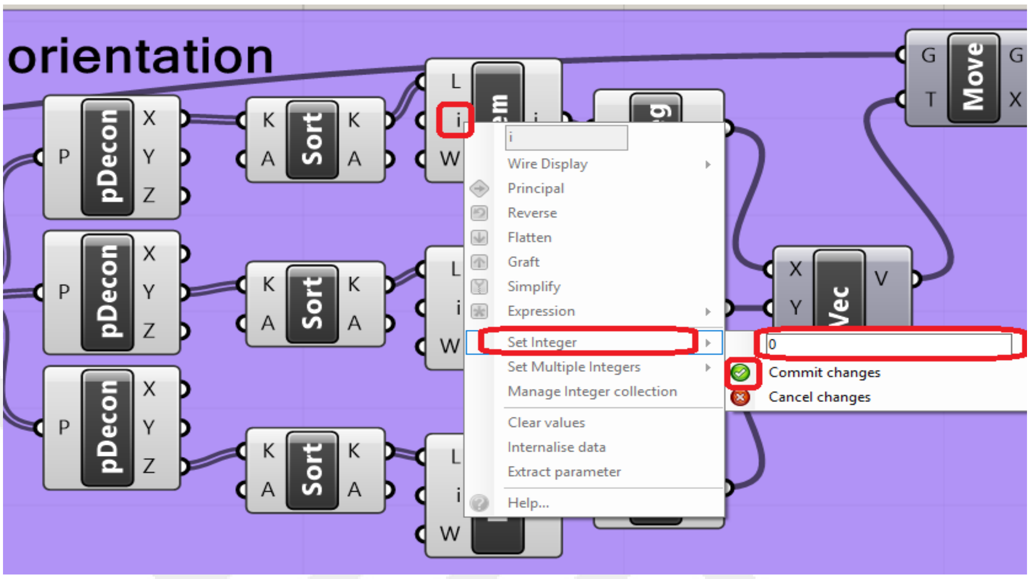


Figure 4.3 : Setting the index input of list item command to zero.

Outputs of List Item commands are connected to three Negative commands seperately, which basically multiplies the given value with a minus, to help constructing the movement vector for the main surface. Each negative command is connected to a Vector XYZ command which builds a vector through given x y and z values. Since the motion vector is ready, it is connected to a Move command's Motion input, while the main surface command is connected to the Geometry input of the same Move command to complete the placement procedure of the main hull to the origin point of Rhinoceros. As it is shown in Figure 4.4, this part of the code is grouped and specified in the Grasshopper Sheet as "Surface Orientation".

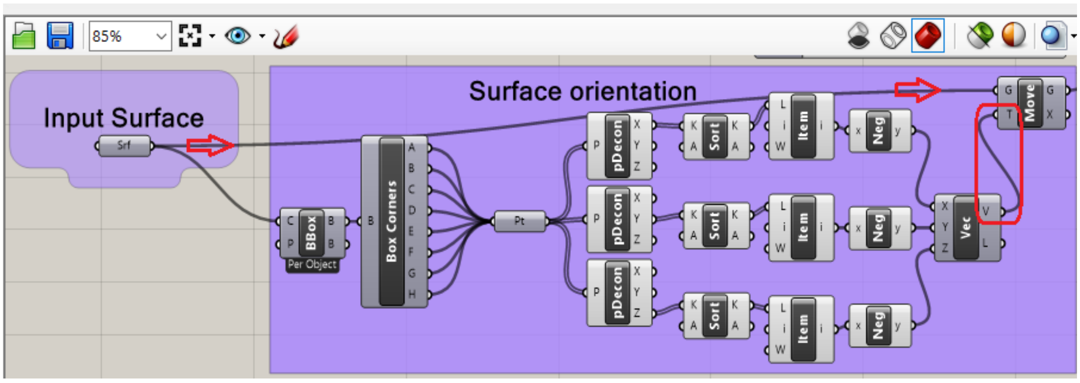


Figure 4.4 : Command chain of moving the imported hull form to Rhinoceros origin.

4.2 Scale Non-Uniform and Obtaining Main Dimensions

After placing the hull form to the origin, Geometry output of Move command is connected to a ScaleNU (Scale Non Uniform) command, which provides scaling a given geometry among x y and z axises separately. As it is shown in Figure 4.5, three number sliders whose default values are 1.0 is connected to x y and z inputs of ScaleNU command separately to let users change the scale of the hull form and one XY plane whose origin input is set as the Rhinoceros origin point as default is connected to Base Plane input of scaleNU command. Numbers on sliders larger than one will scale the surface in positive direction to the relevant axis, while lesser numbers will scale it in negative direction. The Geometry output of ScaleNU command is then connected to a Surface command to specify the surface more clearly and avoid disorder in command sheet.

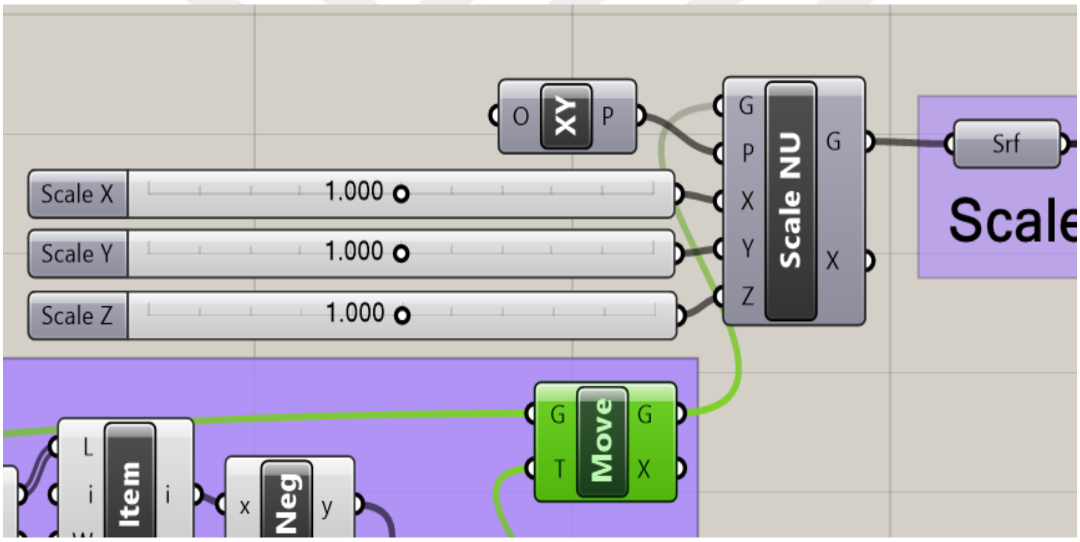


Figure 4.5 : Command chain of subjecting the imported hull form to scaling non-uniformally.

Scaled surface connects to one BoundingBox and one SplitBrep command, but before SplitBrep, the BoundingBox will be explained since the later outputs of command series which start with BoundingBox will also connect in the other command series later on.

After scaled surface is connected to BoundingBox, its Aligned Bounding Box output is connected to a DeconstructBox command to obtain the main dimensions of the box through x y and z axises. All outputs excepting the plane output are connected to separate Number commands and each of them are connected to first three Panel commands and secondly Number commands again. The last number to panel and panel to number command chain make zero change on data but provide a monitoring of the values by the help of Panel command and lets users be able to track changes or potential errors on their main dimensions data of the half hull surface. As it is shown in Fiugre 4.6, this part of the code is grouped and specified in the Grasshopper Sheet as “Main Dimensions”.

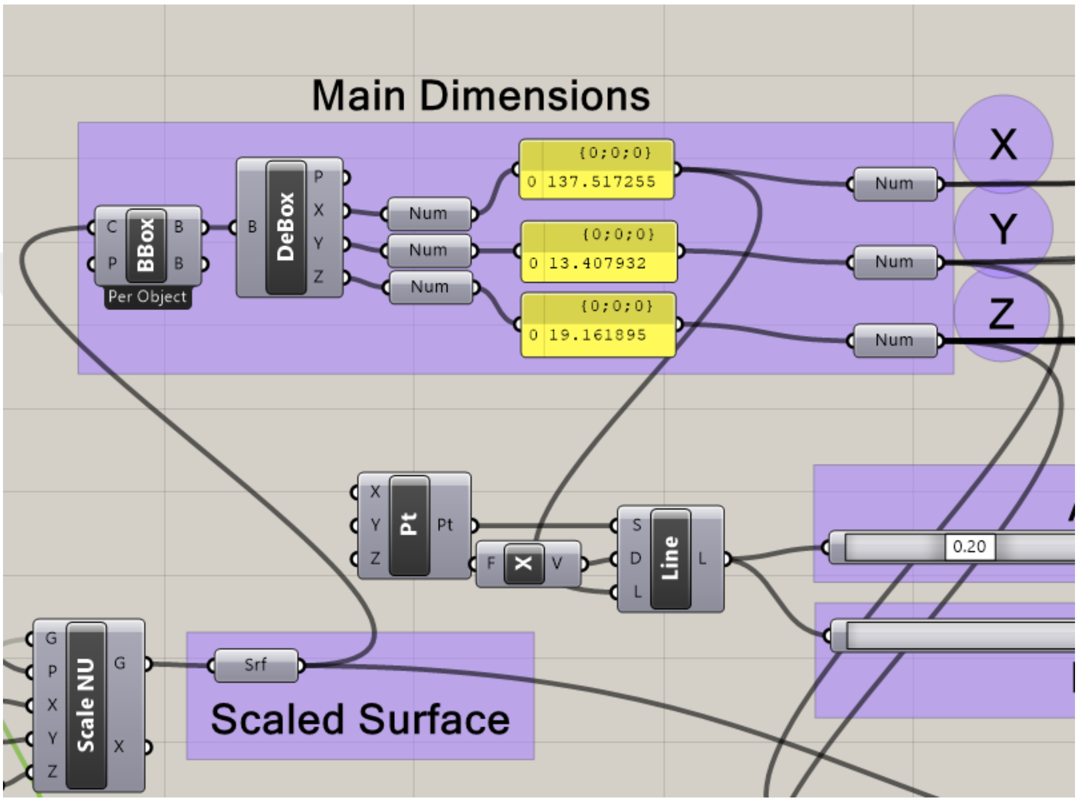


Figure 4.6 : Command chain of obtaining the main dimensions of the scaled hull form.

4.3 Mid Surface Interval Selection and Surface Splitting

Since building a hull surface is a pretty difficult task with a single Rhinoceros or Grasshopper command especially if the hull has a complicated bow or aft geometry (e.g. bulbous bow), this code limits the surface manipulation area for next steps by removing the bow and aft part from the hull surface in order to avoid surface generation failures and interruptions to automated process.

In the main dimensions part, X value output of Deconstruct Box command is connected to a Line SDL command’s Length input, which draws a line using the starting point, direction and length informations. Starting point is set as Rhinoceros origin point by connecting a Construct Point command with default input values to start input and direction is set as connecting a Unit X command to direction input of Line SDL command. This command chain creates a baseline whose length is equal to the overall length of the hull form.

Line SDL command's output is connected to two separate Point on Curve commands. These commands let users pick two points on a curve based on a percentage change of the length of curve. These commands are then connected to two seperate YZ plane commands to create two planes whose origins are the selected points. Two number commands in Main Dimensions part who represents Y and Z values are then connected to B input of seperate Construct Domain commands. Since these commands' A input are set to zero by default, they simply create a domain which represents the interval between zero and the value which is connected to B input. Outputs of Construct Domain commands are connected to two separate Plane Surface command's X and Y inputs with the same array, and YZ planes are also connected to Plane Surface commands' Plane inputs separately. Thus the trimming surfaces are built, see Figure 4.7.

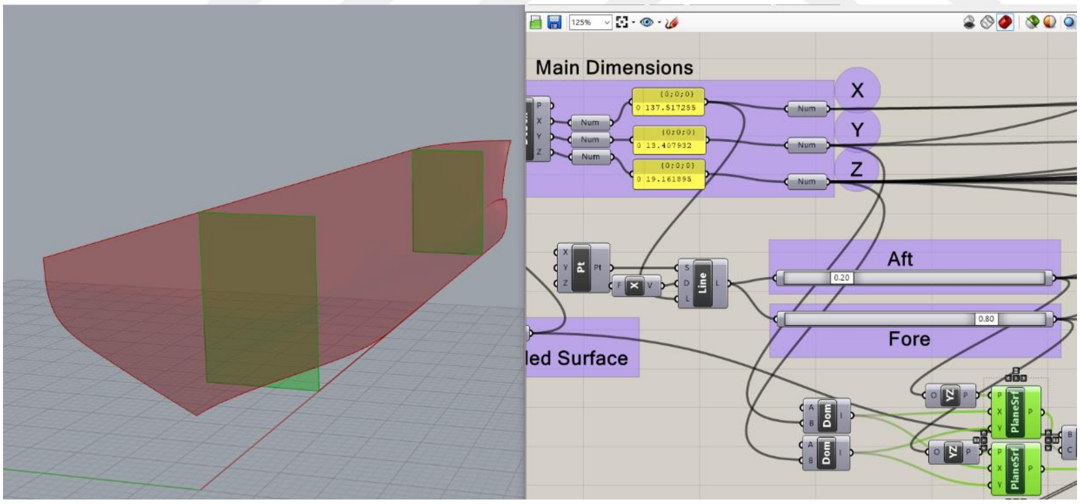


Figure 4.7 : Command chain of selecting the bow and aft trimming points and building the trimming surfaces.

Surface trimming is a two step procedure and each step includes separation of the given surface into two pieces. Scaled surface is connected to a Split Brep command's Brep input and Plane surface command which represents the aft part separation is connected to Cutter input. After first splitting is done, output of Brep Split command are connected to two different List Item commands whose index inputs are set as zero and one to pick different pieces of the splitted surfaces. These List Item commands are then connected to separate Bounding Box commands and Bounding Box commands are connected to Evaluate Box commands to obtain properties of the box. Each Evaluate Box command's point outputs are connected to one Deconstruct Point command. Both Deconstruct Point commands' X outputs are connected to a SortList Command's Keys input and List Item commands are connected both to the same SortList command's Values A input to sort the surfaces based on their x dimension values. Since the first cutting surface is closer to the origin point and has a lower x value, Sort List command will start the list with the aft part of the hull form by matching it with its X value. Thus, the sort list command's Values A output is connected to two List Item commands with their index items set to 0 and 1 separately to represent the different pieces of hull form. The List Item command with the index value 0 represents the aft part while the one with index value 1 represents the fore part. As it is shown in Figure 4.8, this part of the code is grouped and specified on the command sheet as "Surface Order-1" and used within the subsequent parts of the Grasshopper code.

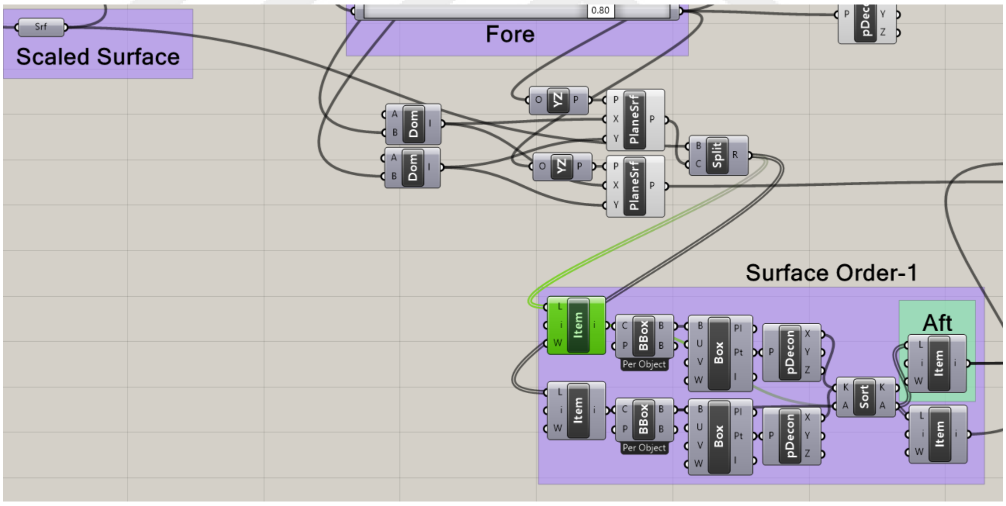


Figure 4.8 : Command sequence of splitting a surface and analysing the positions of splitted surface parts relevant to each other.

For the next splitting process fore part in the last Surface Order group is connected to a Split Brep command's Brep input and the second PlaneSurface which was built for the fore part splitting process before the Surface Order group is connected to Cutter input of the same Split Brep command. Output of this command is subjected to the same "Surface Order-1" code.

It should be noted here that since the Aft and Fore point selectors can travel along the whole baseline, in some cases users may face errors or undesired final surfaces.

Causes of these errors are mostly because of the small differences between the sliders' number values since it will lead the longest part to be the aft or fore one, and the bigger Aft slider number value than the Fore slider as it would lead the exchange of these parts within themselves.

4.4 Section Selection and Point Picking

The two Point on Curve commands are connected to Line command as the aft point goes into A input and Fore point goes into B input. As shown in Figure 4.9, the output of Line command connects to three Point on Curve commands which will represent the locations of three section curves that can be edited on the hull surface.

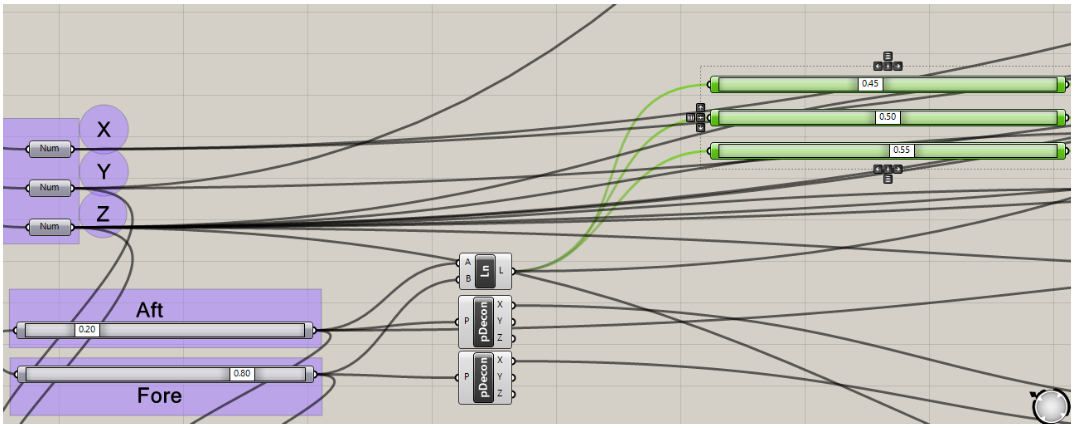


Figure 4.9 : Obtaining three selective points to build the editing sections along the centerline of the hull form.

Outputs of Point on Curve commands are connected to three corresponding Line SDL commands' Start inputs separately, whose length input is connected to maximum Z length of the ship and direction input is connected to a Unit Z command. As can be seen in Figure 4.10 and Figure 4.11, by this command chain, the lines to be projected on the surface are obtained.

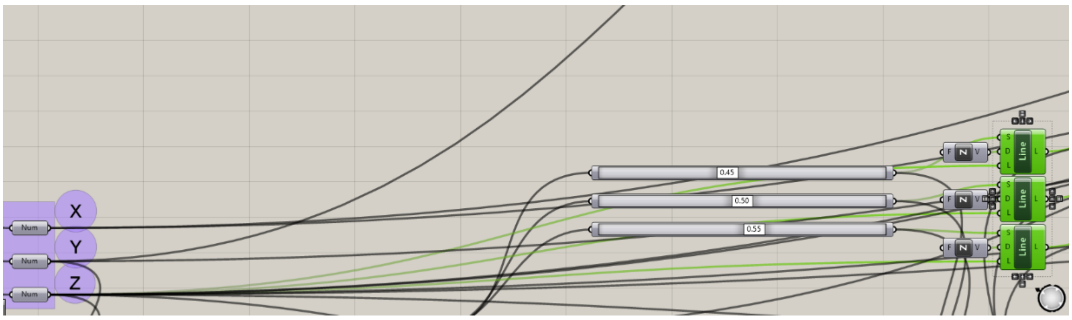


Figure 4.10 : Command chain between "Main Dimensions" part, Point on Curve commands and Line SDL commands.

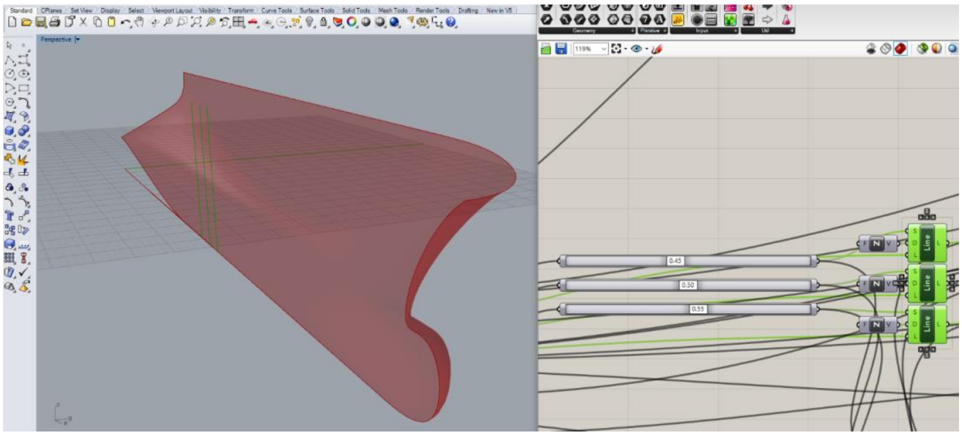


Figure 4.11 : View of hull form with three lines built for projection.

The Line SDL commands' outputs are connected to three separate Project command's Curve input. Since the positive y axis side of the hull form is set at the beginning, the projection direction should also be in positive y direction which requires a Unit Y command at the Direction inputs of Project commands. After mid part of the surface is connected to Brep input of Project command, the section curves to be edited are obtained, see Figure 4.12.

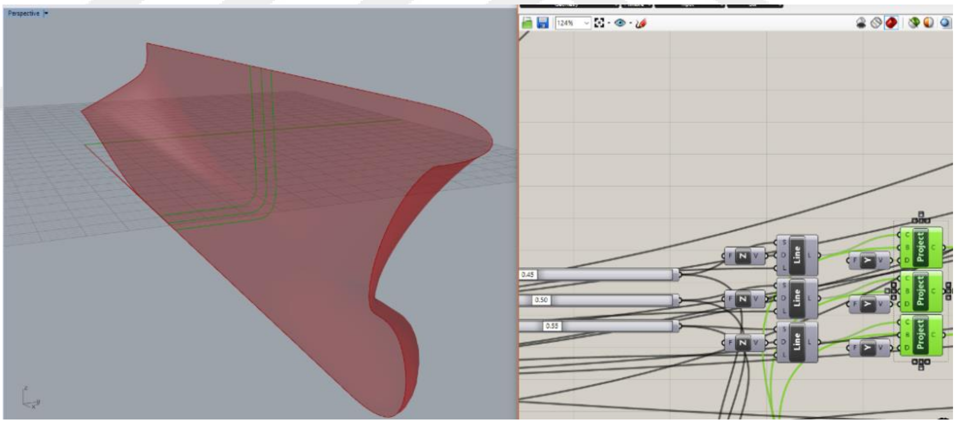


Figure 4.12 : Command chain of projecting the lines onto the surface and view of the section lines.

The code lets users select three points to be edited on the selected sections. The intersection point of the curves and the wetted line can also be edited but the points on the centerline will only be used as contribution to building the edited sections.

Each three selected sections are connected to one End Points curve and three Point on Curve commands. End Points commands' outputs are connected to two Point commands to set the five points on the selected curve together on the command sheet. As shown in figure 4.13, each Point on Curve command groups of three let user adjust the points to be selected on curve. Also all points are set in an order from small to big number value as going from up to down through Point and Point on Curve commands groups of fives.

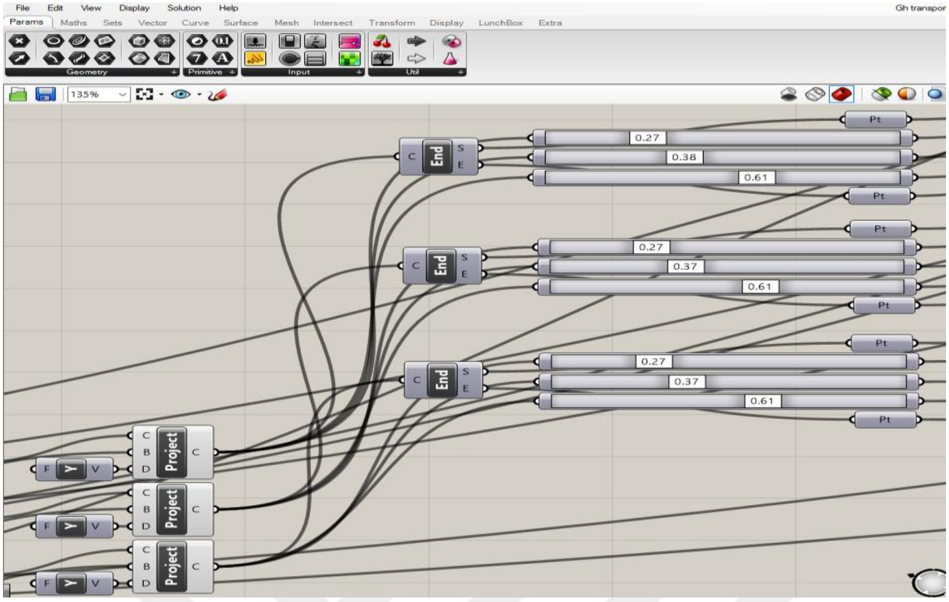


Figure 4.13 : Command sequence of selecting three points on each section curves.

4.5 Parametrization of Point Movements and Construction of New Frame Curves

To provide the y axis movements of points on selected sections, editing interval along y axis is limited to %10 of the maximum breadth of the form within “Interval Domain” part of the code, as it is shown in Figure 4.14. To establish this, half of the maximum breadth of hull form which is obtained in “Main Dimensions” part is connected to a Multiplication command’s A input. To set the limit based on maximum breadth, B input of Multiplication command is set to 0,2 as default. It can be changed by right clicking to B input and following Set Data Item tab, if desired.

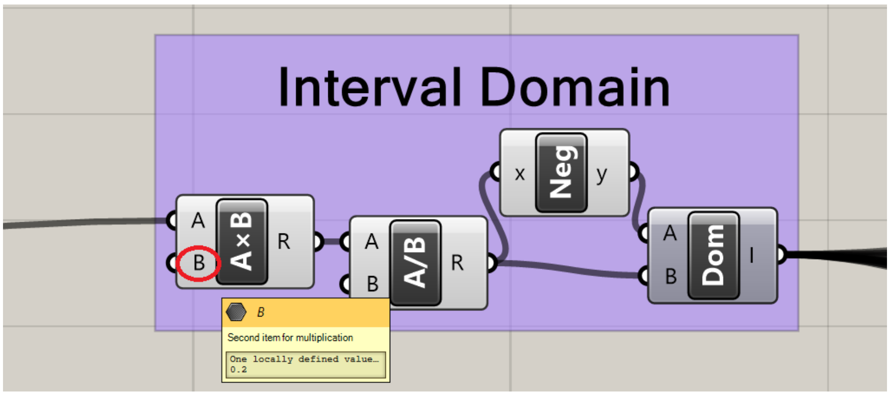


Figure 4.14 : Command chain of “Interval Domain” part.

Result of multiplication is connected to a Division command’s A input and its B input is set as 2 for the next steps of creating the number domain. Result of Division command is connected to a Construct Domain command as one connected via a Negative command to A input while the other is directly connected to B input.

Since four points are allowed to be edited for three sections, four Remap Numbers are created for each section and four number sliders whose default values are 0,5 are connected to its Value input. Source Domain inputs of Remap Numbers commands are set to zero to one as default and Target Domain inputs are connected to previous Construct Domain command. This command chain enables the conversion of zero to one number interval to the created domain interval by Construct Domain command, see Figure 4.15.

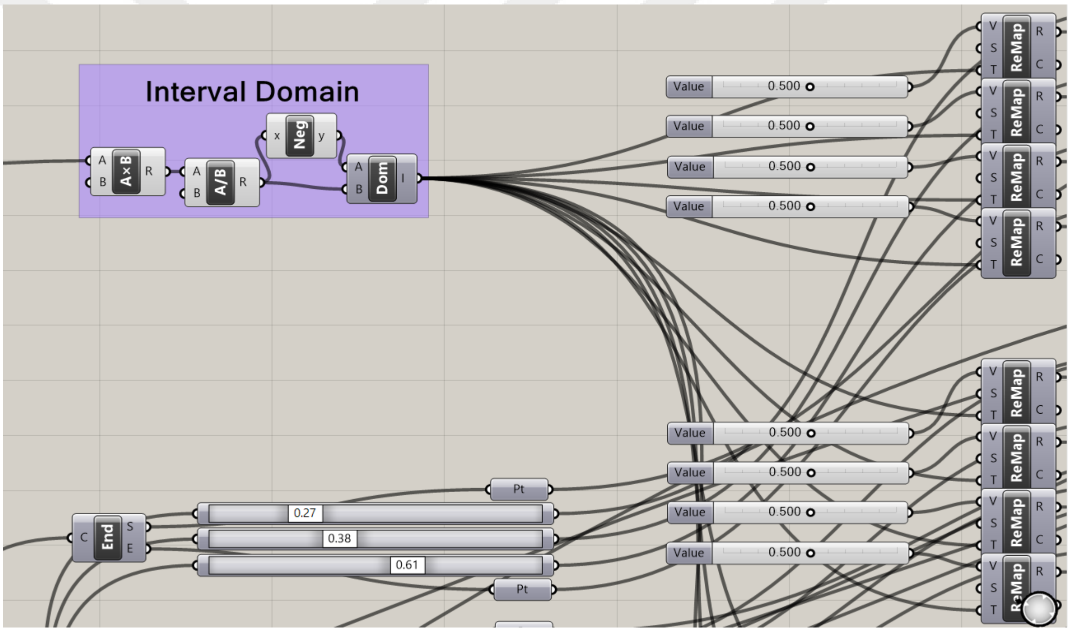


Figure 4.15 : Command chain of “Interval Domain” part and Remap Numbers commands.

Each Mapped outputs of Remap Numbers commands are connected to Move commands’ Motion input via Unit Y commands. Point and Point on Curve commands excepting the ones that represent the points on center line are connected to these Move commands’ Geometry input to complete manipulation part of the selected points on selected sections of the code, see Figure 4.16.

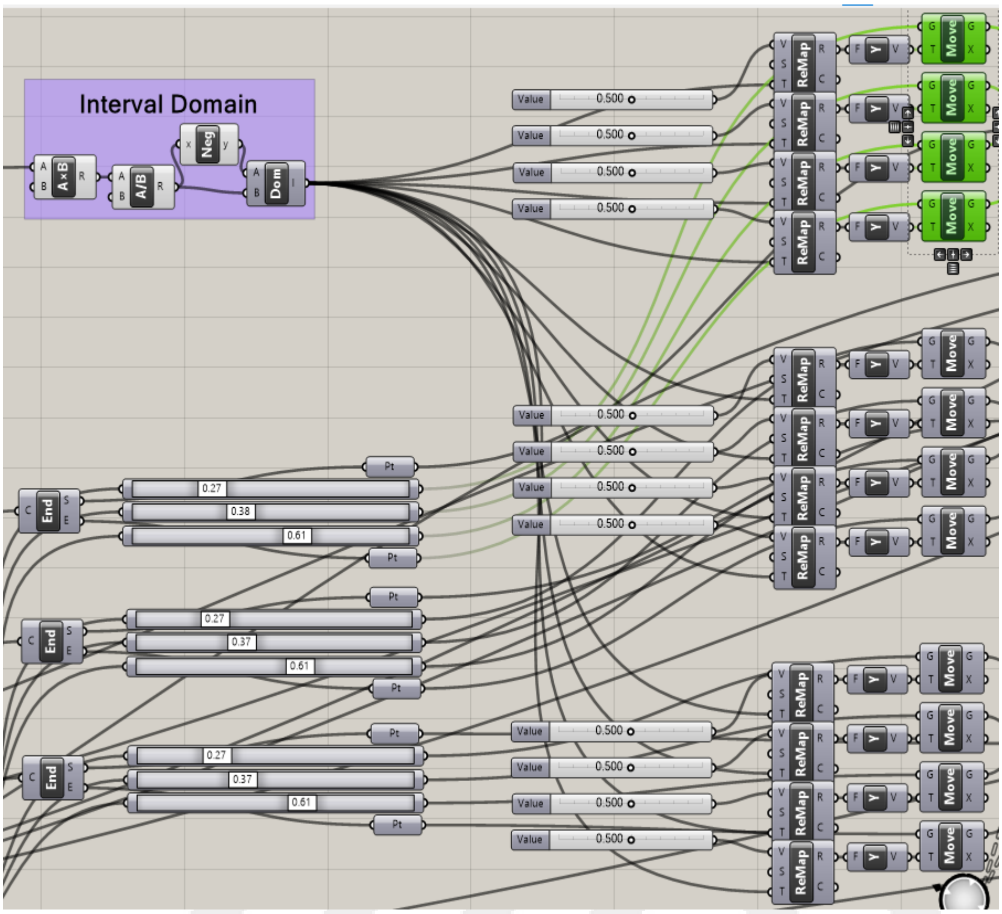


Figure 4.16 : Command groups of curve control point editors.

Since the next step is constructing the manipulated sections, four edited point sets and their fifth point on centerline are connected to three separate point commands. The data inside these three Point commands are flattened by right clicking on the command and selecting Flatten option to sort the points and avoid undesired curve shapes. As it is shown in Figure 4.17, these Point commands are then connected to three separate Interpolate commands to build the new section curves.

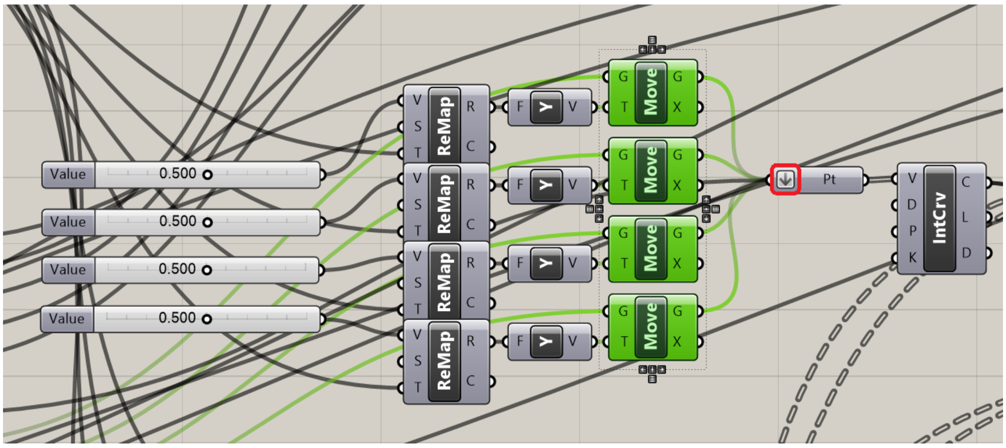


Figure 4.17 : Command chain of building the manipulated curves.

Since three sections of mid part of the surface has been changed, a new mid part surface is needed. To obtain a more accurate surface, assistant sections will be generated starting from the edited curves group and through fore and aft limits of mid-surface.

4.6 Construction of Assistant Frames and New Mid-Surface

This part of the code starts with three Point on Curve commands who represent the location of selected sections along x axis. As it is shown in Figure 4.18, each of these commands are connected to separate Deconstruct commands to obtain their x values. X outputs of Deconstruct commands are then subjected to a series of Minimum and Maximum commands to be sorted based on their x values.

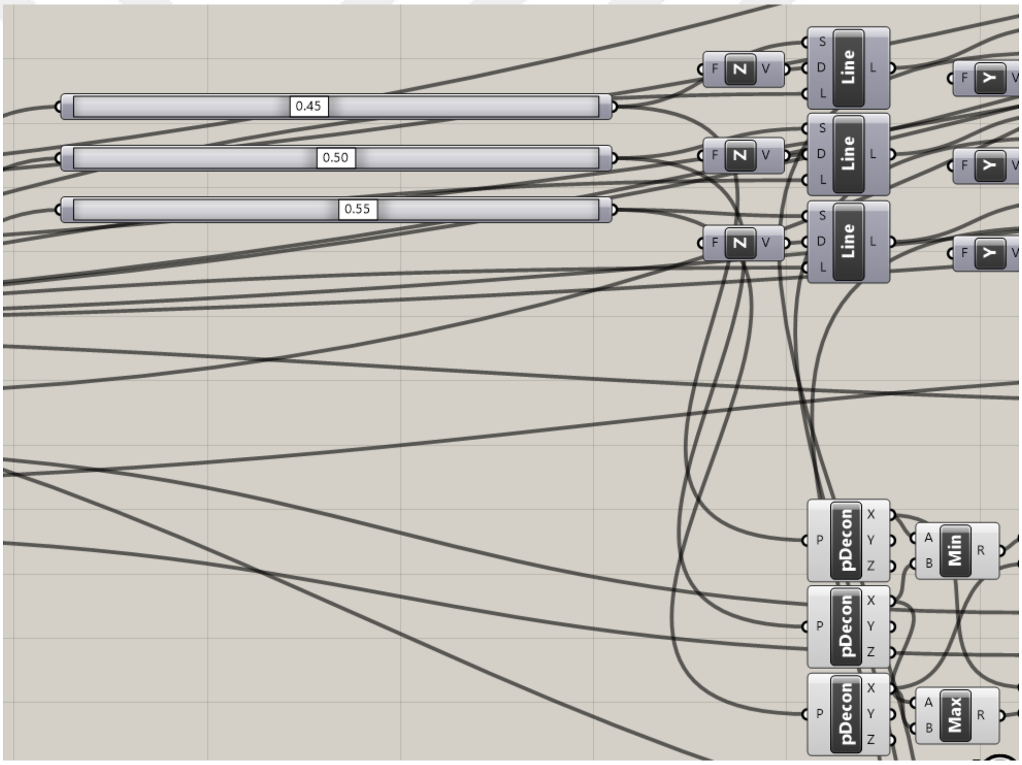


Figure 4.18 : Deconstruction of preselected three points into their components.

Two x outputs are connected to a Minimum command first and the third x output and the output of the first Minimum command are connected to another Minimum command together. This sequence is repeated again but this time with a Maximum command instead, see Figure 4.19. Last Minimum and Maximum commands' outputs represent the two closest sections to fore and aft limits of the mid-surface.

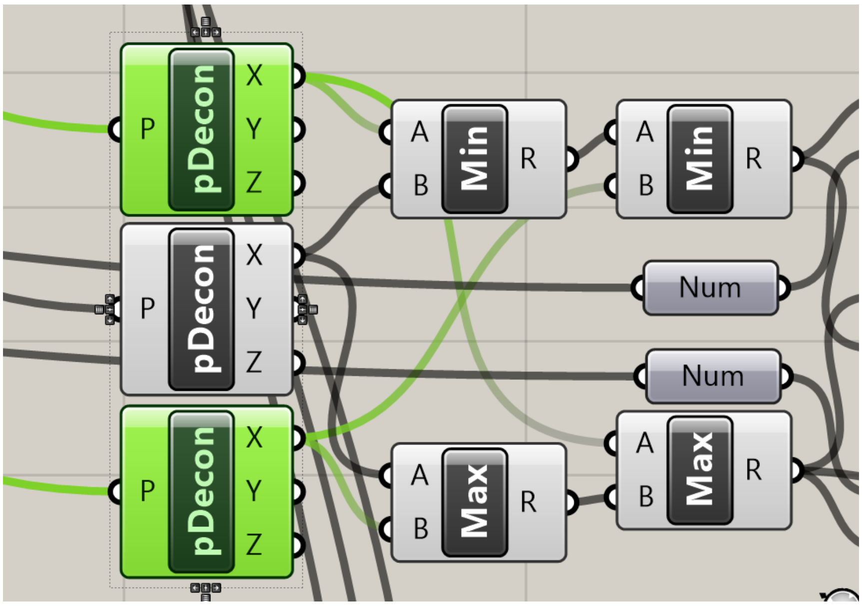


Figure 4.19 : Command chain of analysing the positions of section curves along x axis.

Two Deconstruct commands' x outputs which represent the limits of the mid-surface are connected to two Number commands and placed between the last Minimum and Maximum commands for easy tracking of commands and connections, see Figure 4.20. Three subtraction operations are made to establish the x value of intervals between; fore limit of mid-surface and the edited section with largest x axis value, aft limit of mid-surface and the edited section with lowest x axis value, edited sections with lowest and highest x axis values.

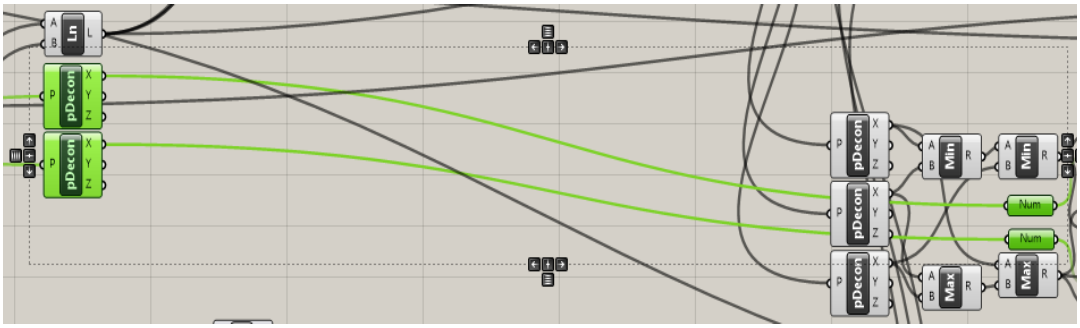


Figure 4.20 : Command conections between Deconstruct commands and Number commands.

The latest mentioned subtraction operation will be made first and the output will be divided by 2 by a Division command to obtain the approximate step size for assistant sections. The output of Division command is connected to B inputs of two separate Division commands. These commands' inputs are connected to the Subtraction commands number one and three as can be seen in the Figure 4.21.

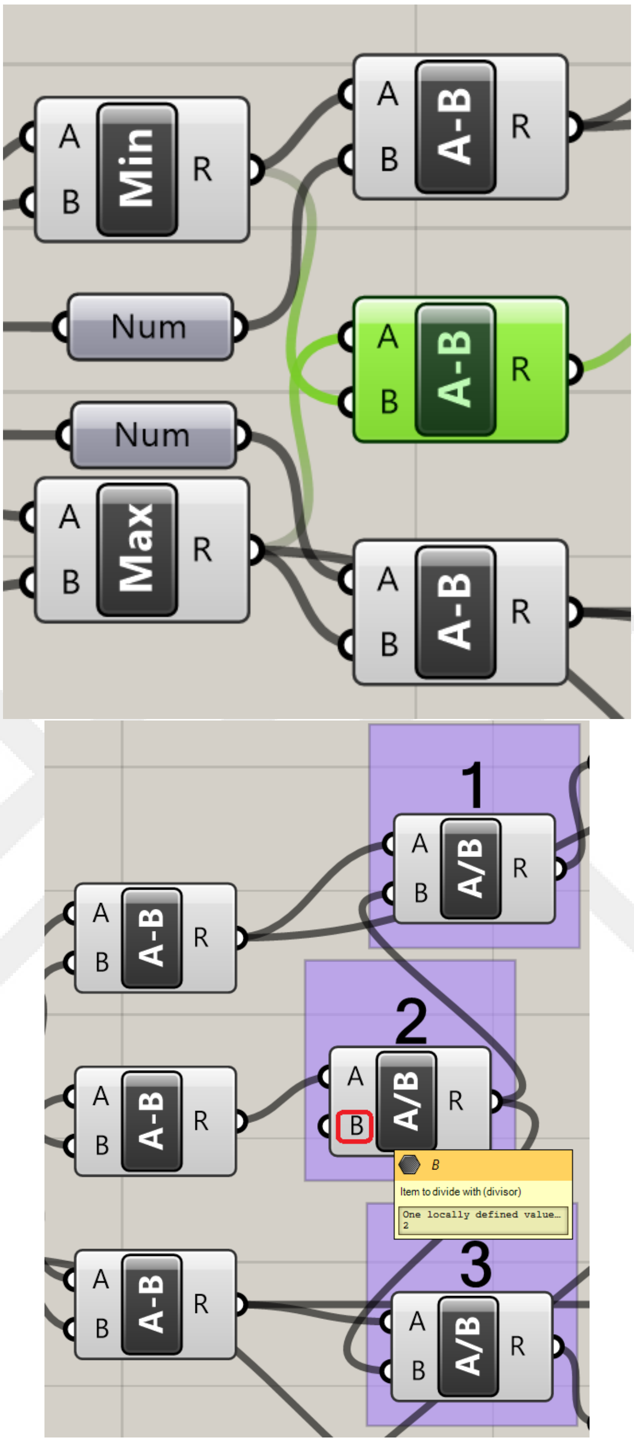


Figure 4.21 : Command connections between Minimum, maximum and algebraic commands.

Divison commands with number one and three are then connected to separate Round commands which basically round a given number to an integer. Output of Round commands are connected to separate Division commands as divisiors, where the dividened numbers are the outputs of Subtraction commands placed before the Division commands number one and three.

Assistant sections will follow a procedure as drawing a line, copying it through x axis equispaced and projecting them onto hull surface. Two Line SDL commands will be used for drawing the line. First line of Set 1 has its starting point at the aft limit of mid-surface. As it is shown in Figure 4.22 and Figure 4.23, Z output of "Main Dimensions" is connected to its Length input where a Unit Z command is connected to its Direction input. A Series command whose Step and Count inputs are connected to Division and Round commands respectively and placed before Set 1 is used to form the copying sequence of line command.

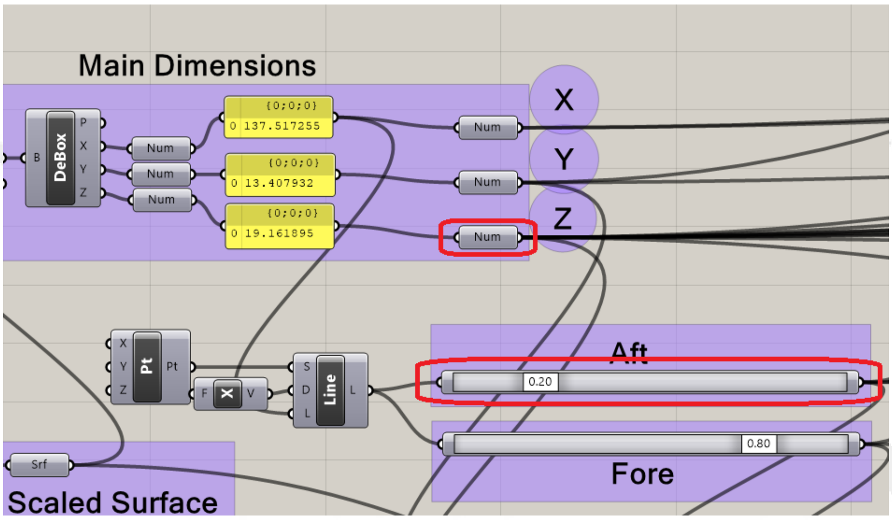


Figure 4.22 : Z output of "Main Dimensions" part and Aft point selector.

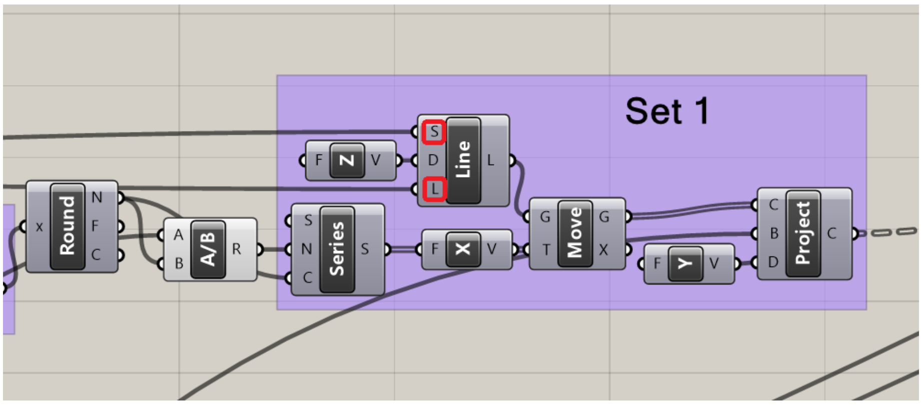


Figure 4.23 : Connection points to the Line SDL command.

Start input of Series command is set as zero by default. Output of Series command is connected to a Move command's Motion input via a Unit X command. Output of Line SDL command is connected to Geometry input of the same Move command. Geometry output of Move command is connected to a Project command's Curve input. Mid-surface is connected to Brep input and a Unit Y command is connected to Direction input of the same Project command, see Figure 4.24.

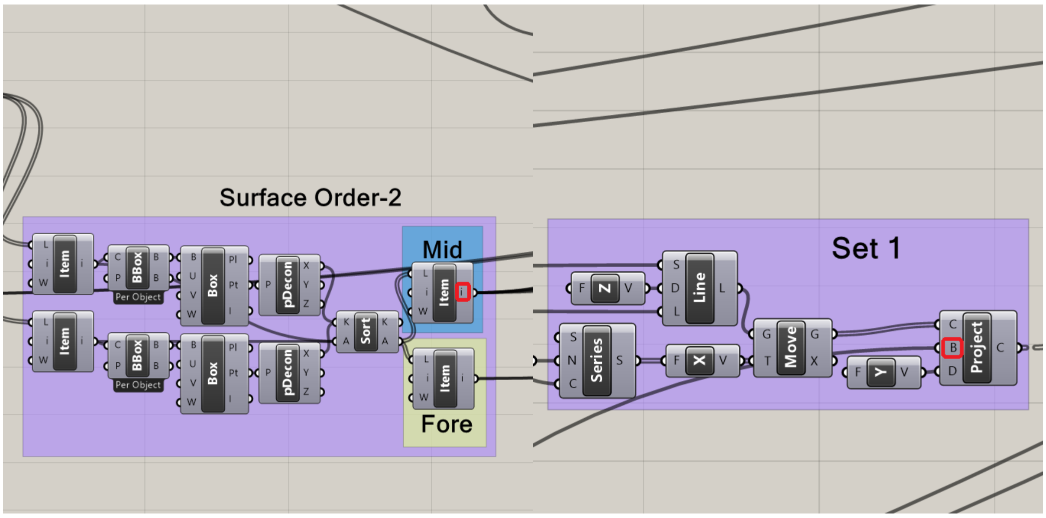


Figure 4.24 : Connection between List Item command and Project command.

Second line of Set 2 has the same starting point as the edited curve who is foremost along x axis. Z output of “Main Dimensions” is connected to its Length input where a Unit Z command is connected to its Direction input. As it is shown in Figure 4.25, output of Division command placed before Set 2 is connected to both Start and Step inputs of a Series command. Output of Round command placed before Set 2 is connected to Count input of the same Series command.

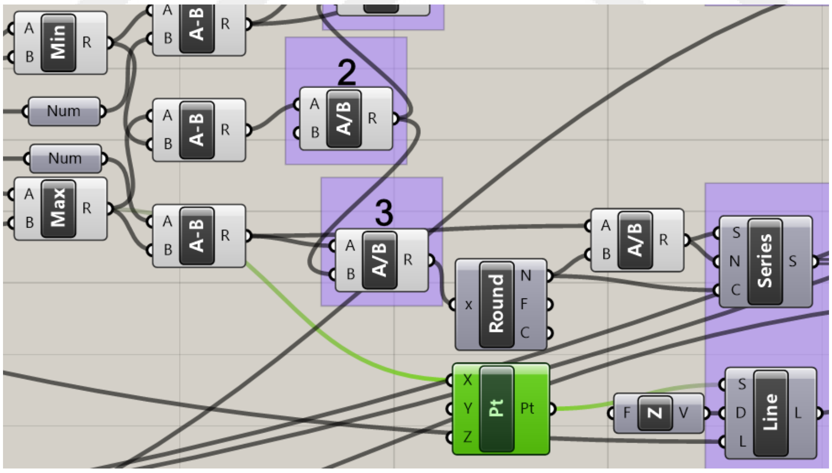


Figure 4.25 : Initial command sequence for Set 2.

Output of Series command is connected to Motion input of a Move command via a Unit X command. Output of Line command is connected to Geometry input of the same Move command. Geometry output of Move command is connected to a Project command’s Curve input. Mid-surface and a Unit Y command are connected to Brep and Direction inputs of the same Project command respectively. Command groups Set-1 and Set-2 can be seen in Figure 4.26.

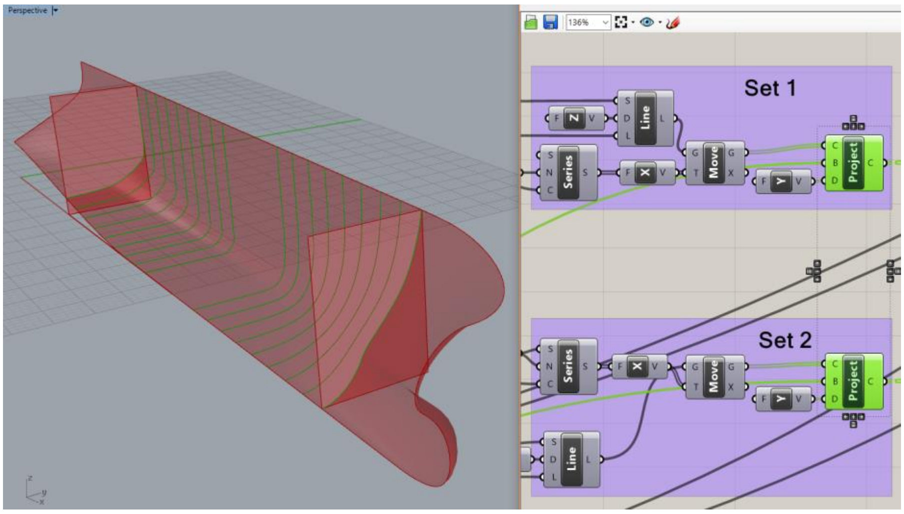


Figure 4.26 : Command sequences of Set 1 and Set 2.

Two Project commands who represent the assistant sections and the three edited curves are connected to a single Curve command and flattened. This curve command is connected to an End Points command and its Start output is connected to a Deconstruct command. X output of Deconstruct command is connected to Keys input of a Sort List command while the Curve command is connected to the Values input of the same Sort List command. This command chain provides a list of curves that are sorted based on their positions along x axis. Values output of Sort List command which include the section curves is connected to Curves input of a Loft command. A Loft Options command is connected to Options input of the same Loft command. Type input of Loft Options command is set as Loose and all the other inputs of this command are left as their default values. Loft command then generates the new manipulated mid part of the surface, see Figure 4.27.

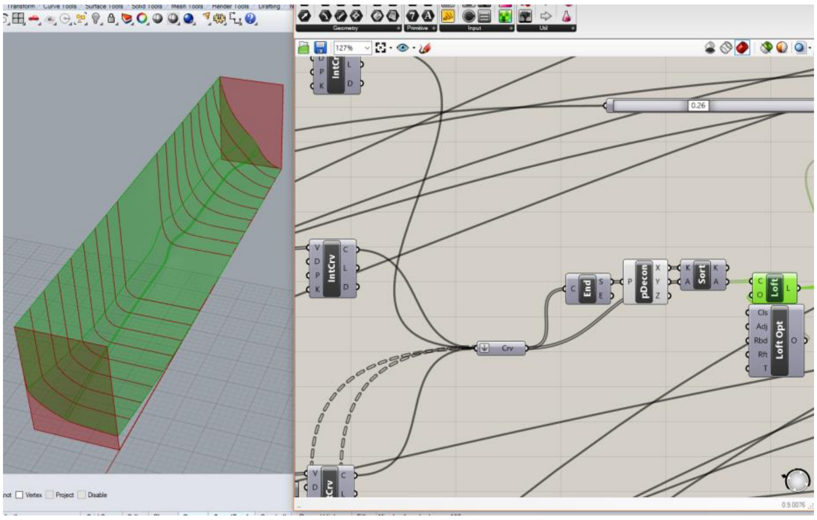


Figure 4.27 : Command sequence and view of assistant sections and loft surface.

4.7 Linear Extrusion and Obtaining the Final Hull Form

In the next part of the Grasshopper code, command chains that provide linear extention to the length of the hull surface will be explained. The code provides users capability of selecting the section from which the extension will be made with a Number Slider command and then splits the mid surface into two parts at the selected section. Extrusion of the selected section is made with an amount based on user selection, surface parts which are ahead of extrusion section are moved by the same amount of extrusion through positive x axis and all surface pieces are joined back together to obtain the final form.

Output of Loft command is connected to a Bounding Box command whose output is connected to a Deconstruct Box command. X, y and z outputs of Deconstruct Box command are connected to three separate number commands to convert a number interval to a single value. Since there haven't been any changes on the length of mid-surface yet, output of the Line command by using which the three section selection points are constructed in the previous part is connected to a Point on Curve command, see Figure 4.28.

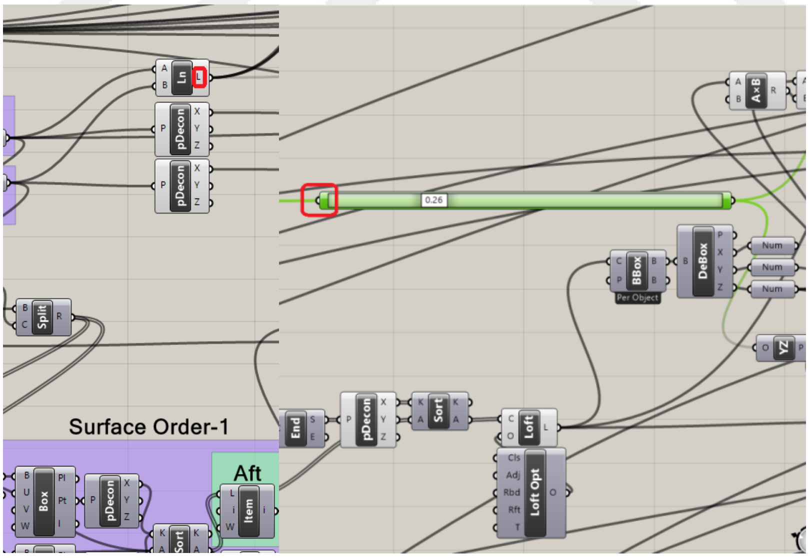


Figure 4.28 : Connection between Line and Point on Curve commands.

This command is connected to two command groups, one of which generates the extruded surface and the other splits the manipulated surface into two parts. First connection is made to Start input of a Line SDL command, to whose Direction and Length inputs a Unit Z command and the Z output of Deconstruct Box command via a Number command over Multiplication and Addition commands are connected respectively. The Multiplication command’s B input is set as 0,5 and the addition command’s B input is set as 1 by default. The aim here is extending the length of the line in order to make sure the line projects onto the surface successfully. Output of Line SDL command is connected to a Project command’s Curve input, to whose Brep and Direction commands, output of Loft command and a Unit Y command are connected respectively. This command chain generates the section at a user selected point, see Figure 4.29.

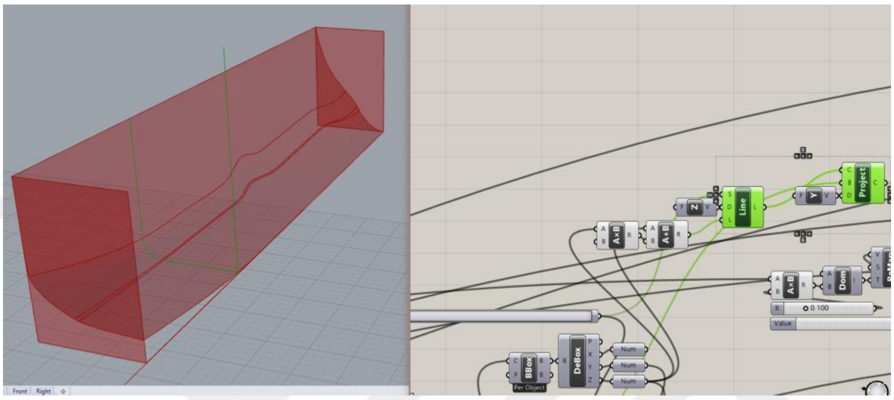


Figure 4.29 : Command sequence and view of the section, at where the linear extension will be made.

Extrusion amount is parametrized and limited as a function of overall length of the hull form. X output of Main Dimensions part is connected to a Multiplication command’s A input to whose B input a Number Slider command is connected. Minimum and maximum values of this Number Slider command are zero and 0,5 respectively. Default value is set to 0,1 which limits the amount of length extension to %10 of the overall length of the ship. As it is shown in Figure 4.30, limits of this Number Slider command can be changed by right clicking on it and following the edit option when desired.

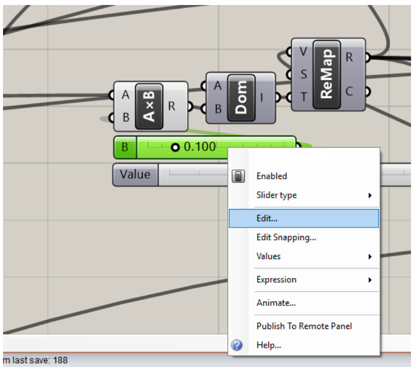


Figure 4.30 : Command chain of limiting the extrusion amount as a function of overall length.

Output of Multiplication command is connected to a Construct Domain command’s B input, whose A input is set as zero by default. Output of this command is connected to a Remap Numbers command’s Target input, to whose Value input a Number Slider command within the interval of zero to one is connected. The Source input of Remap Numbers command is set as one by default. Second Number Slider command determines the amount of extension and Remap Number command translates the two intervals within each other which are zero to one and zero to x percentage of the overall length of the hull form. Output of the Remap Numbers command is connected to a Unit X command, whose output is connected to Direction input of an Extrude command. Output of the Project command is connected to Base input of the same Extrude command. This command chain generates the linearly extruded surface using the selected section, see Figure 4.31.

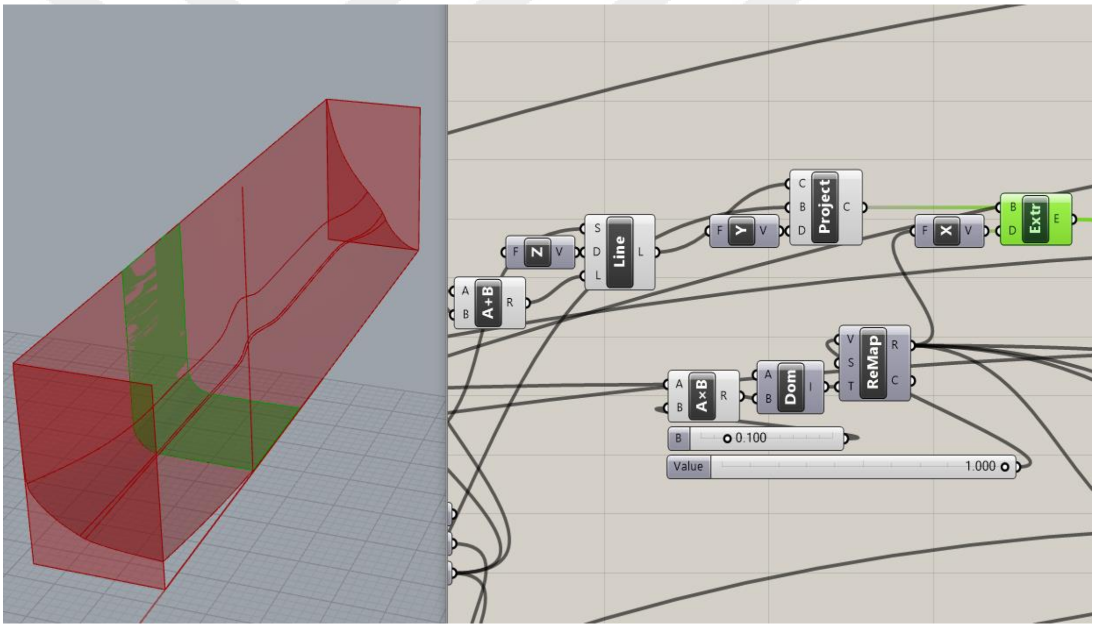


Figure 4.31 : Command chain and view of linear extrusion process.

Second connection of Point on Curve command is made to a YZ Plane command, whose output is connected to a Plane Surface command’s Plane input. Second connection of Point on Curve command is made to a YZ Plane command, whose output is connected to a Plane Surface command’s Plane input. Y and Z outputs of Deconstruct Box command are connected to A inputs of two separate Construct Domain commands via two separate negative commands. Y and Z outputs of Deconstruct Box command are connected to B inputs of the same two Construct Domain commands. Outputs of these Construct Domain commands are connected to

X and Y inputs of the Plane surface command. Y and Z outputs of Deconstruct Box command are connected to A inputs of two separate Division commands whose B inputs are set as 2 by default. Outputs of Division commands are connected to Y and Z inputs of a Construct Vector command whose Vector output is connected to Motion input of a Move command. Output of Plane Surface command is connected to Geometry input of the same Move command. By this command chain the trimming surface is built and moved to the appropriate location for trimming process. Output of Move command is connected to one Split Brep command's Cutter input, to whose Brep input the output of Loft command is connected. As it is shown in Figure 4.32, output of Split Brep command is subjected to "Surface Order" command chain and specified as "Surface Order-3".

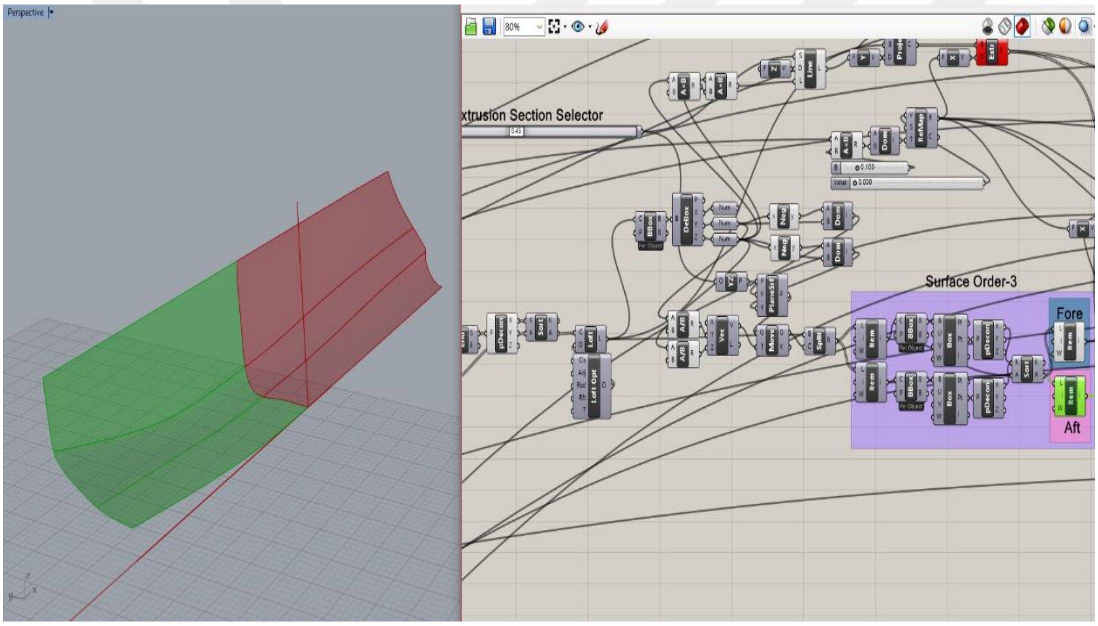


Figure 4.32 : Splitting the mid surface into fore and aft parts.

Since the extrusion and splitting procedures are done, the next part is shifting the surfaces which are in front of the extrusion section with respect to positive x axis and joining them with the three remaining surface parts that are behind the extrusion section, again with respect to positive x axis. Mentioned surfaces are the ones specified as Fore in the "Surface Order-2" and "Surface Order-3" parts. As it is show in Figure 4.33, outputs of these List Item commands are connected to Geometry input of a Move command. Output of Remap Numbers command is connected to Motion input of the same Move command via a Unit X command.

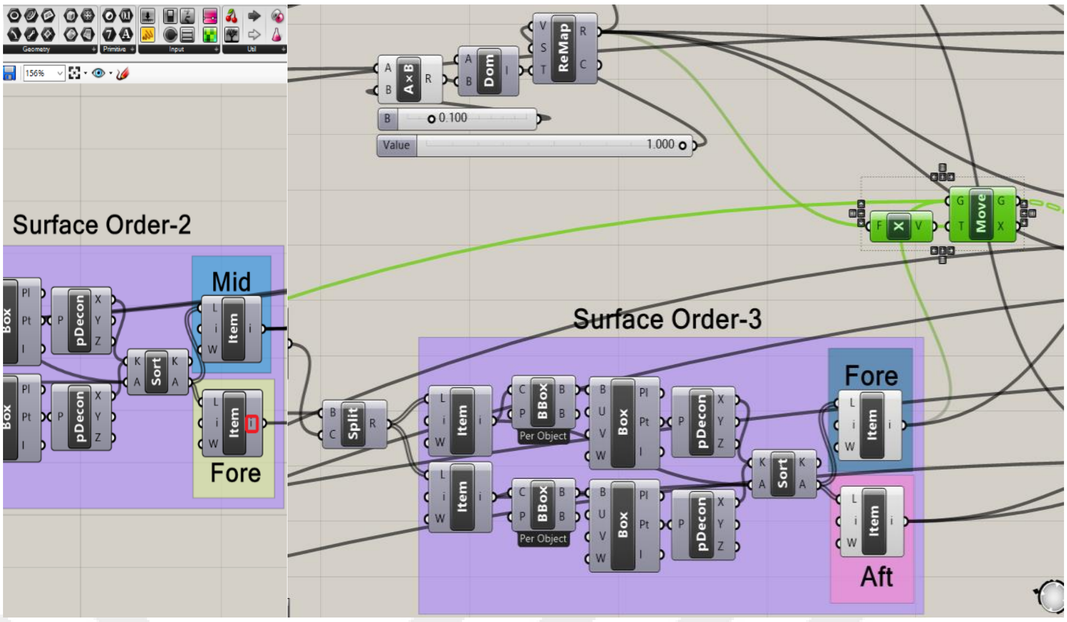


Figure 4.33 : Connecting the fore parts within surface order 2 and surface order 3 parts to the move command.

Output of this Move command, List Item command specified as Aft in the "Surface Order-1" part, List Item command specified as Aft in "Surface Order-3" part and the Extrude commands are connected to a Brep Join command to combine all surface parts into the fully manipulated hull form surface, see Figure 4.34. Output of extrusion command is connected to Brep Join command via a Brep command specified as "extrusion" to let users enable or disable this function. The Brep Join command is used to monitor the fully manipulated hull form surface, to avoid bugs, parts of the finalized hull form surface will be subjected to other command groups seperately.

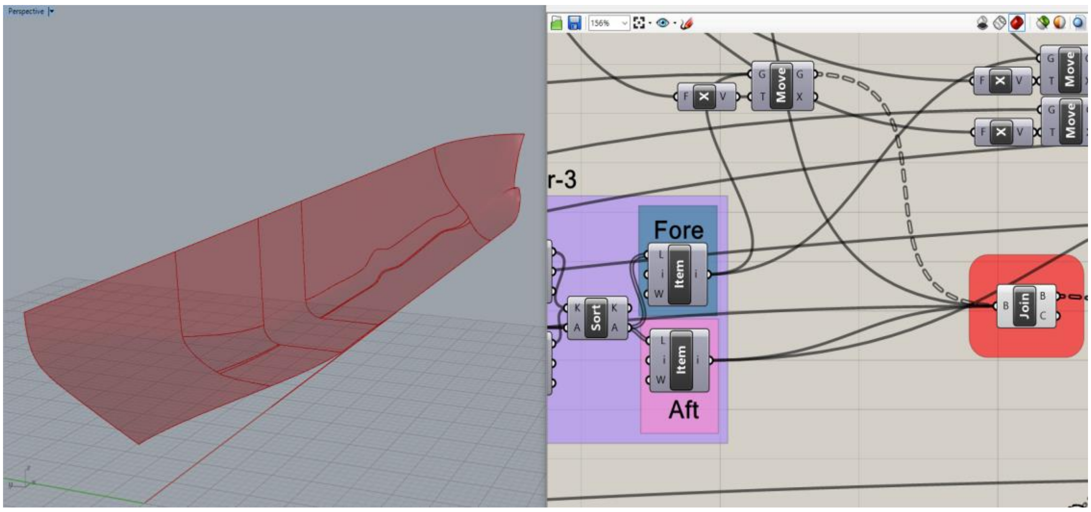


Figure 4.34 : Connecting the surface parts into one Brep join command which represents the final hull form.

4.8 Obtaining the Offset Table of Finalized Hull Form

Since the final form is ready and there will not be any manipulations applied to it, the next and final part of the code is extraction of the control points which will be used to constitute required files for the michell.exe to estimate the resistance. The waterline and section numbers are the user selected values and to establish this, two number sliders are created and specified as “Waterline Number” and “Section Number”. Limits of these sliders are set as 0 to 10 and 0 to 100 for waterline and section numbers respectively. These limit values can be changed by right clicking on the related number slider and proceeding to Edit tab, see Figure 4.35. Also their default values are set as 6 for waterline and 16 for section numbers.

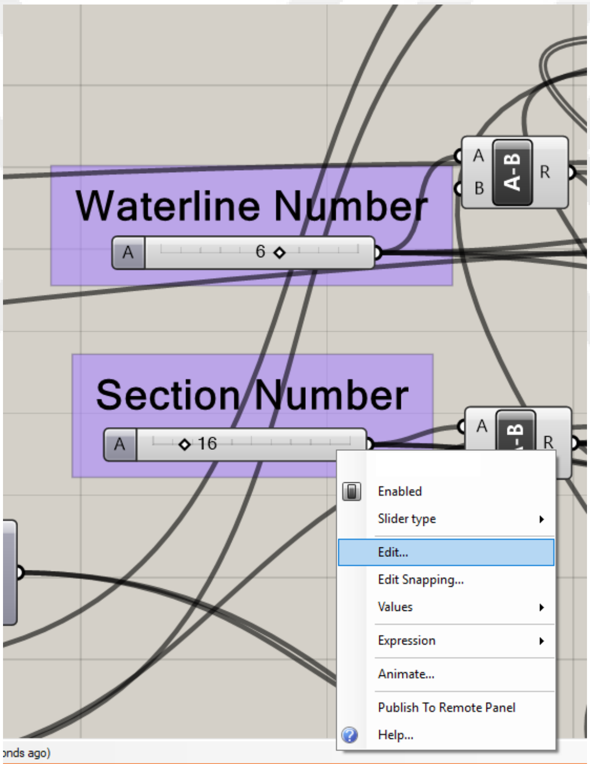


Figure 4.35 : Editing the section number slider properties.

4.8.1 Obtaining the planar surfaces

The main idea of obtaining the control points is based on obtaining the intersection points of section lines and planar surfaces which are in array with the waterline spacing through positive z axis. Since there are many problematic cases that have been faced during the development of the code, which interrupted the automated process, planar surfaces have been prefered instead of waterline curves for intersecting with the section curves. The progress includes three parts instead of directly intersecting the section lines with planar surfaces due to requirements of michell.exe program. Considering the bulbous bow of a vessel for instance, a section curve over it will naturally include an amount of zeros within, depending on the position of the curve on x axis, on the contrary of a mid section curve. Also the waterline which represents the baseline of the vessel is a straight line composed of zeros. Michell.exe requires all of the halfbreadth table and since the first waterline and the last section line have to be composed of zeros only, these specific lines will be calculated seperately, and the remaining points will be obtained by the intersection method.

Number slider which represents the waterline number is connected to a subtraction command whose B input is set to 1 by default, see Figure 4.36. This subtraction idea, which is used again in further steps, is prefered because lists in Grasshopper start with zero instead of one and also six equispaced waterlines mean five equal intervals. Output of subtraction command is first connected to a division command's B input, to whose A input output of the draught number slider is connected, to divide the depth length of the hull form to five equal pieces.

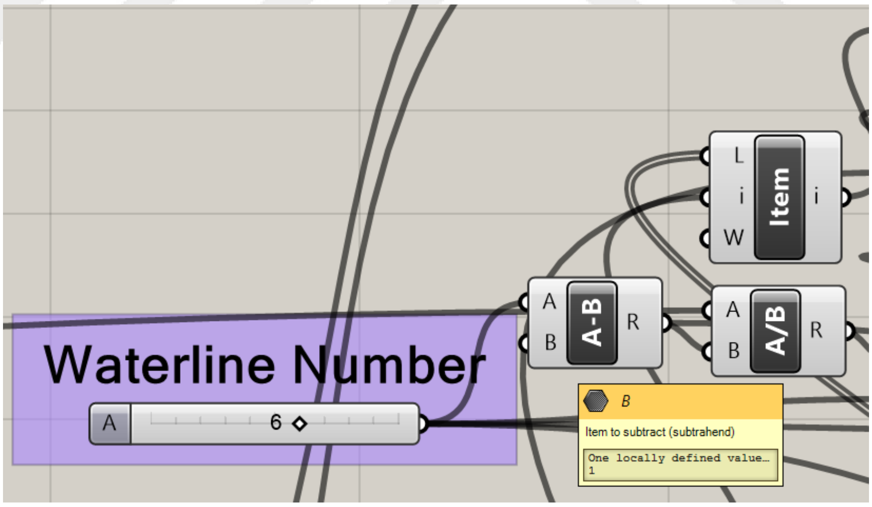


Figure 4.36 : Command chain starting with waterline number slider.

Output of division command is connected to a Series command's Step input. Waterline number slider is connected to Count input of the same Series command, whose Start input is set as zero. Before using the output of Series command, the line and surface which will contribute to creation of the planar surface array is required. To obtain these, length of the wetted line data needs to be calculated. Also since input files of Michell.exe will require data of the underwater form, a surface trimming procedure should be applied at the user selected draught depth. Only change on the overall length of the hull form is the extrusion length which has been used before. As it is shown in Figure 4.37 and Figure 4.38, X output of “Main Dimensions” part and Mapped output of Remap Numbers command are connected to A and B inputs of an Addition command.

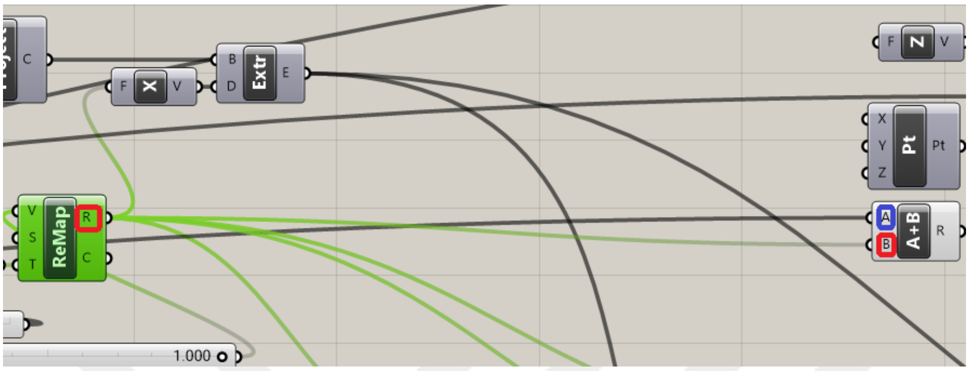


Figure 4.37 : Connections to Addition command – 1.

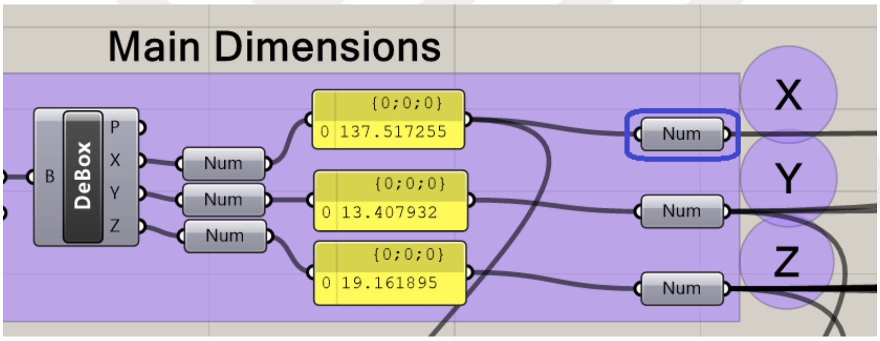


Figure 4.38 : Connections to Addition command -2.

Output of Addition command which represents the final overall length of the hull form is connected to A inputs of two separate Multiplication commands, whose B inputs are set to 1.5 and 0.25 respectively. Output of first Multiplication command is connected to X input of a Plane Surface command, where the output of the second Multiplication command is connected to X input of a Construct Vector command via a Negative command.

Y output of Deconstruct Box command which represents the maximum halfbreadth of the mid surface is connected to A inputs of two separate Multiplication commands, whose B inputs are again set to 1.5 and 0.25 respectively. Output of first Multiplication command is connected to Y input of the Plane Surface command, where the output of the second Multiplication command is connected to Y input of the Construct Vector command via a Negative command. Number slider command which represents the draught value is connected to Z input of the same Construct Vector command, see Figure 4.39 and Figure 4.40.

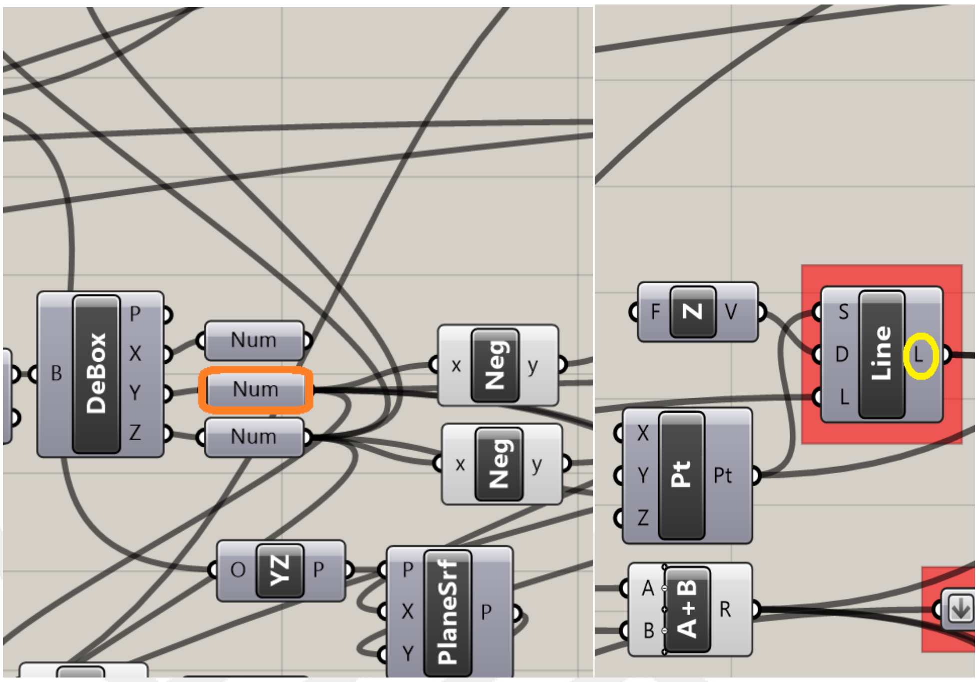


Figure 4.39 : Command connections for building the trimming surfaces-1.

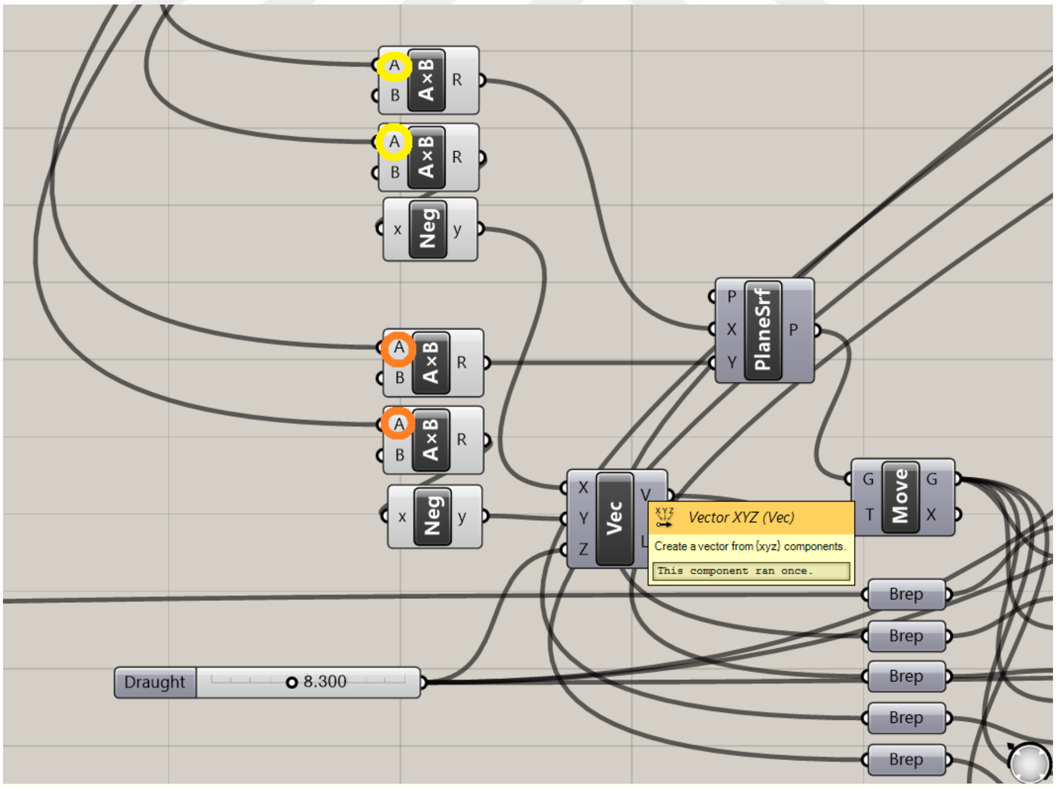


Figure 4.40 : Command connections for building the trimming surfaces-2.

Outputs of the Plane Surface and Construct Vector commands are connected to Geometry and Motion inputs of a Move command. By this command chain the trimming surface is obtained for the underwater form obtaining process.

Outputs of two List Item commands who represent the foremost parts according to the extrusion line are connected to Geometry inputs of two separate Move commands, to whose motion inputs output of the Remap Number command is connected via Unit X commands, to complete the surface movements according to extrusion amount. Outputs of the commands who include the finalized surface parts are connected to five separate Brep commands. Outputs of these Brep commands are connected to Brep inputs of five separate Split Brep commands, to whose cutter inputs the output of Move command that represents the trimming surface is connected. Each Split Brep command is connected to one Box Properties command via a Bounding Box command. Center outputs of Box Properties commands are connected to separate Deconstruct Point commands, whose Z outputs are connected to Key inputs of five separate Sort List commands to order the surface pieces based on their positions along z axis, see Figure 4.41.

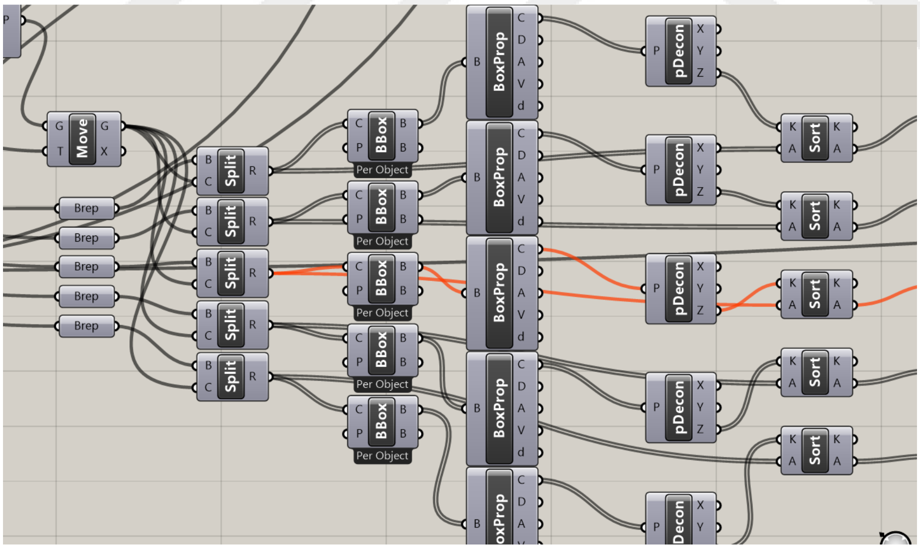


Figure 4.41 : Command chain of obtaining the wetted line.

Outputs of Split Brep commands are connected to Values inputs of corresponding Sort List commands. Values outputs of Split Brep commands are connected to List inputs of five separate List Item commands whose index values are set to zero. Outputs of List Item commands are connected to Deconstruct Box commands via Bounding Box commands. X outputs of Deconstruct Box commands are connected to an Addition command specified as “LwL” with five inputs via Number commands. Y outputs of Deconstruct Box commands are connected to Key input of a Sort List command via Number commands. Key output of Sort List command is connected to A input of a Subtraction command via a List Length command whose B input is set to one. Key output of Sort List command and output of Subtraction command are connected to List and Index inputs of a List item who is specified as “Bmax” respectively. By this command chain, surface pieces of underwater hull form, wetted length and maximum halfbreadth of underwater form which are required for Michell.exe input files are obtained, see Figure 4.42 and Figure 4.43.

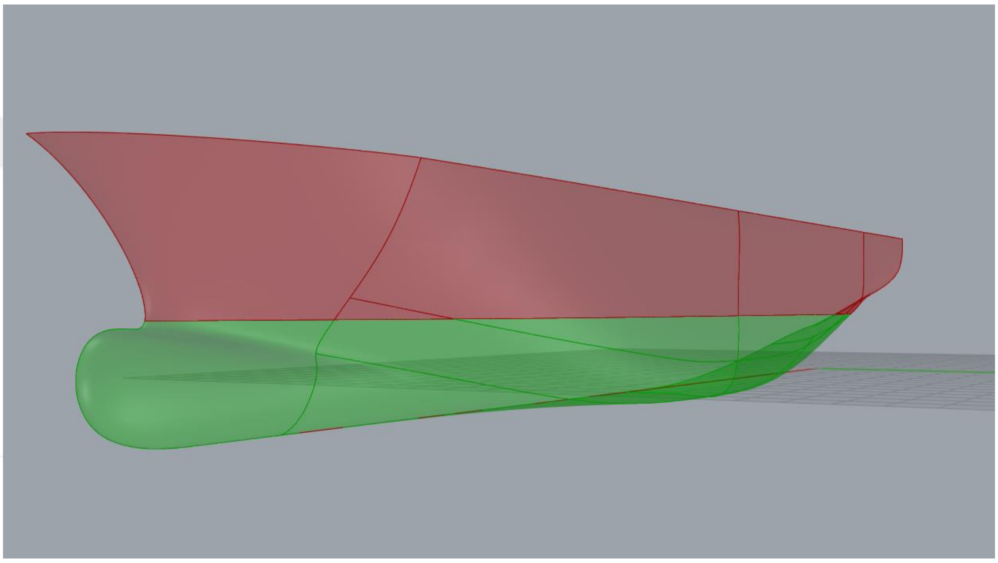


Figure 4.42 : View of the underwater form (green coloured surface).

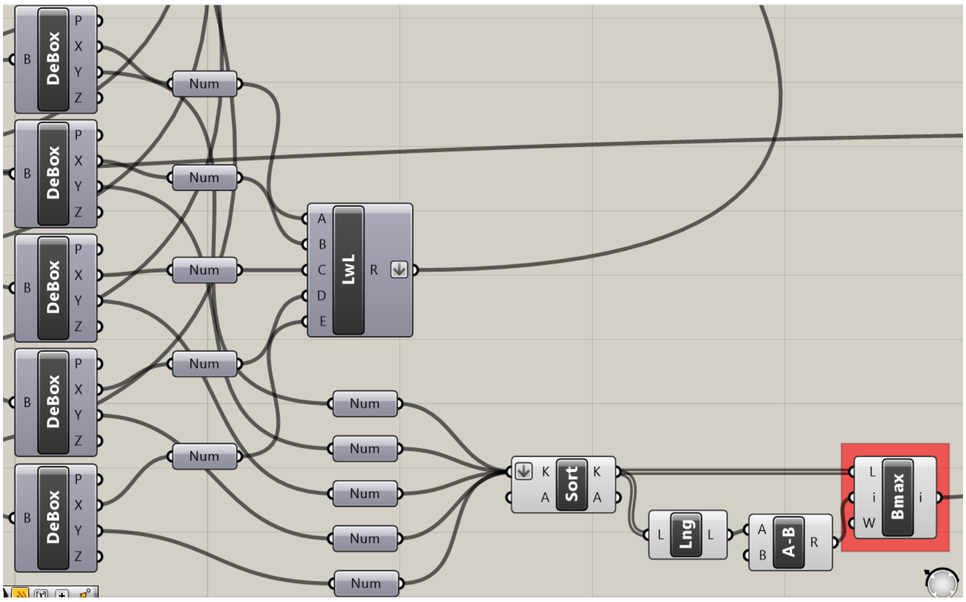


Figure 4.43 : Command chain of obtaining the maximum half breadth of underwater form.

A Construct Point command is created whose inputs are all set as zero. Output of this Construct Point command is connected to a Line SDL command's Start input, to whose Direction and Length inputs a Unit X command and the output of the Addition command is connected respectively. Output of Line SDL command is connected to a Division command's A input via a Length command, whose B input is set as 10. Output of Division command is connected to a Extend Curve command's Start and End command, whose Type input is set as Line. Output of Line SDL command is connected to Curve input of the same Extend Curve command, see Figure 4.39. By this command chain, the baseline of the hull form is extended by %10 of its own length at its both ends, see Figure 4.44.

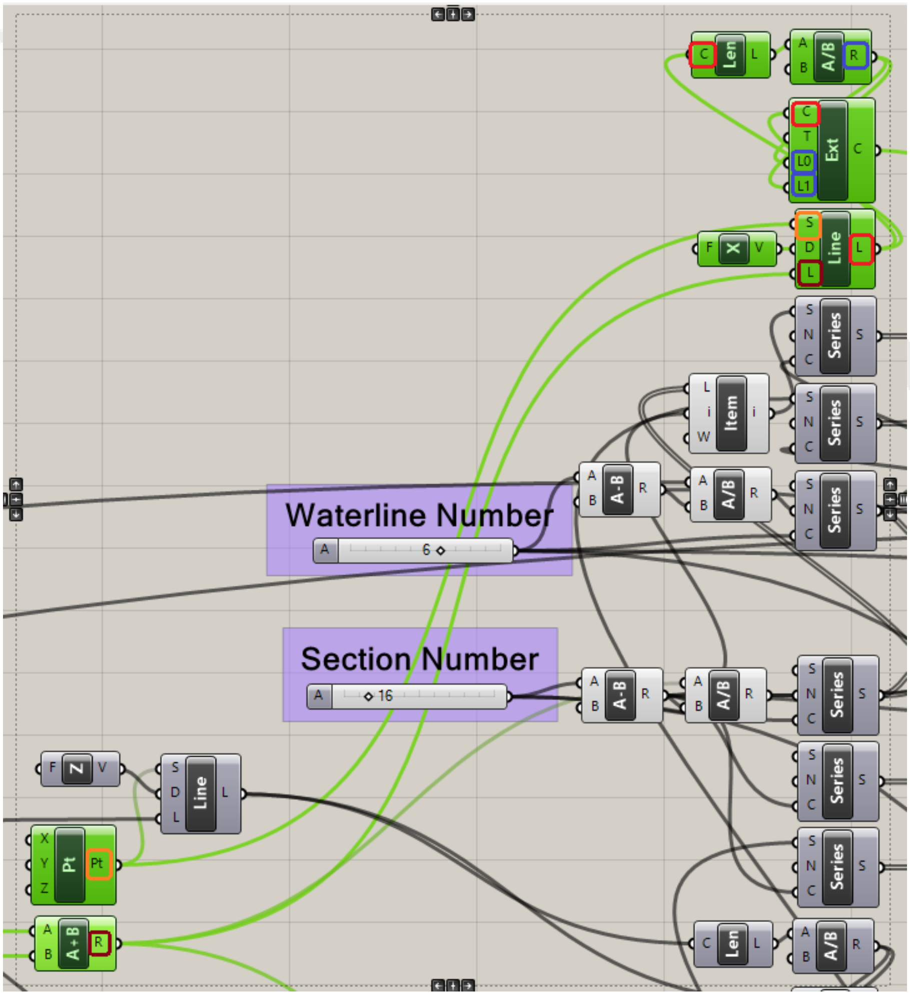


Figure 4.44 : Command chain of obtaining the extended line parallel to x axis.

Output of the Extend Curve command is connected to a Move command's Geometry input. Output of the Series command is connected to Motion input of the same Move command via a Unit Z command. Output of Move command is connected to an End Points command, whose Start output is connected to a Deconstruct command's Point input. Z output of Deconstruct command is connected to a Sort List command's Key input, to whose Values input the Geometry output of the Move command is connected, see Figure 4.45.

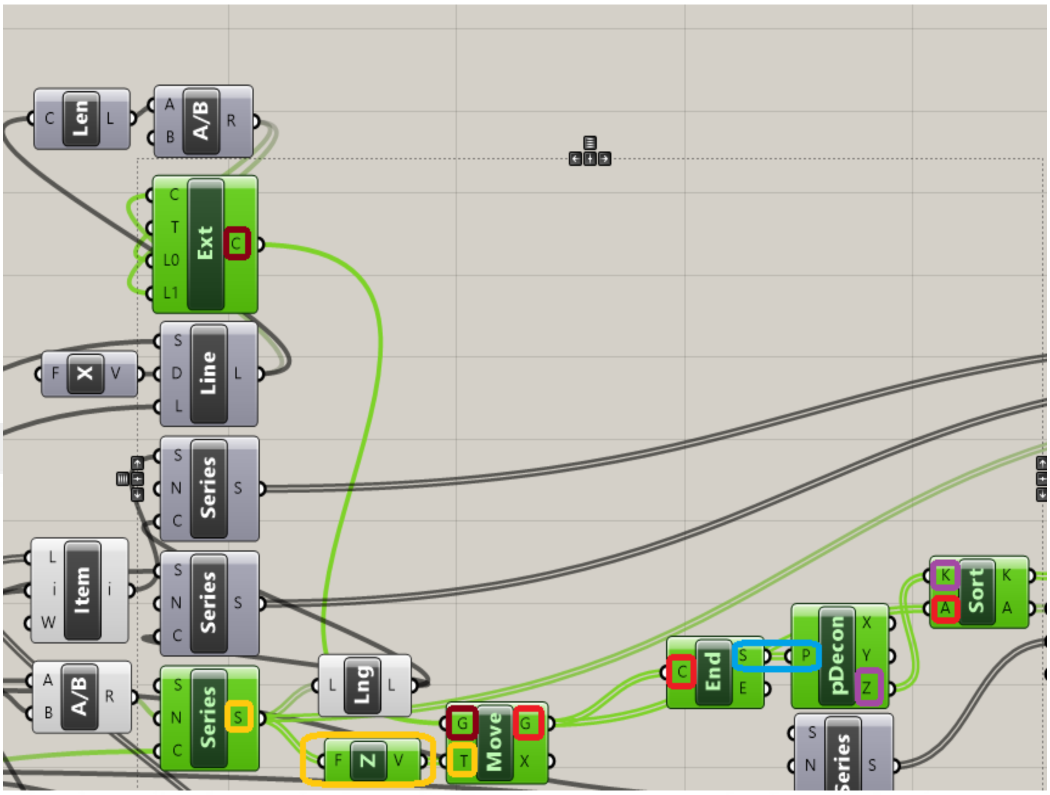


Figure 4.45 : Command chain of building and sorting the extended lines array.

The output of Subtraction command which is connected to waterline number slider is connected to a Series command's Count input. Start and Step inputs of this Series command are both set to one, which will lead to a series starting from one, increasing by one and ending at five in this case. Values output of Sort List command is connected to a List Item command, to whose Index input the output of Series command is connected. Output of List Item command is connected to an Extrude command's Base input. Z output of "Main Dimensions" part is connected to an Addition command's A input, whose B input is set as 10. Output of this Addition command is connected to the Extrude command's Direction input via a Unit Y command. As it is shown in Figure 4.46, by this command chain five planar surfaces are created excluding the one based on xy plane with z = 0.

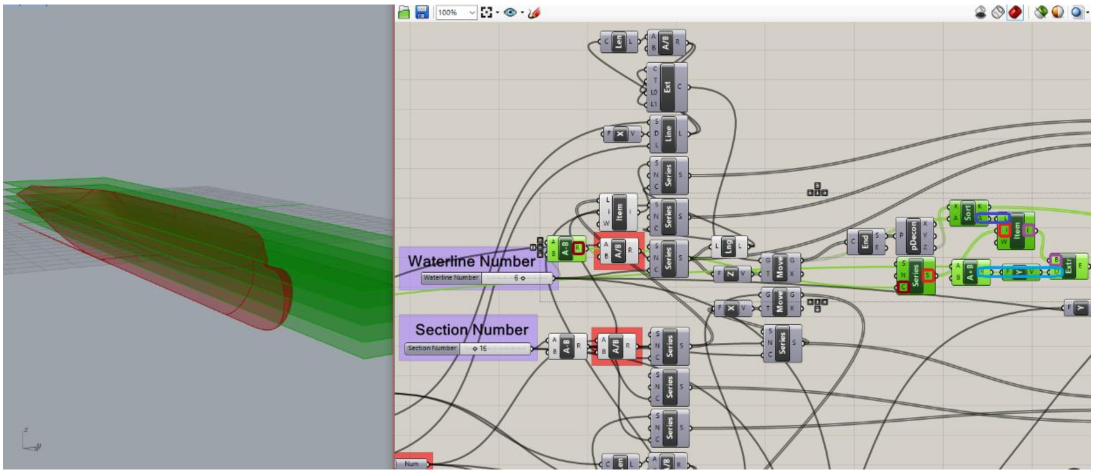


Figure 4.46 : Command chain and view of the plane surfaces.

4.8.2 Obtaining the section lines and extension curves

Section number slider is connected to A input of a Subtraction command, whose B input is set as 1. Output of List Item command which is specified as "LwL" is connected to a Division command's A input via a Number command, to whose B input the output of the Subtraction command is connected. Output of this Division command is connected to Step input of a Series command which includes a zero in its Start input. Section number slider is connected to same Series command's Count input. The output of Construct Point command, which is placed on the Addition command used to calculate the final overall length of the hull form, is connected to a Line SDL command's Start input, to whose Direction input a Unit Z command is connected. Output of number slider command specified as "Draught" is connected to Length input of the same Line SDL command, see Figure 4.47.

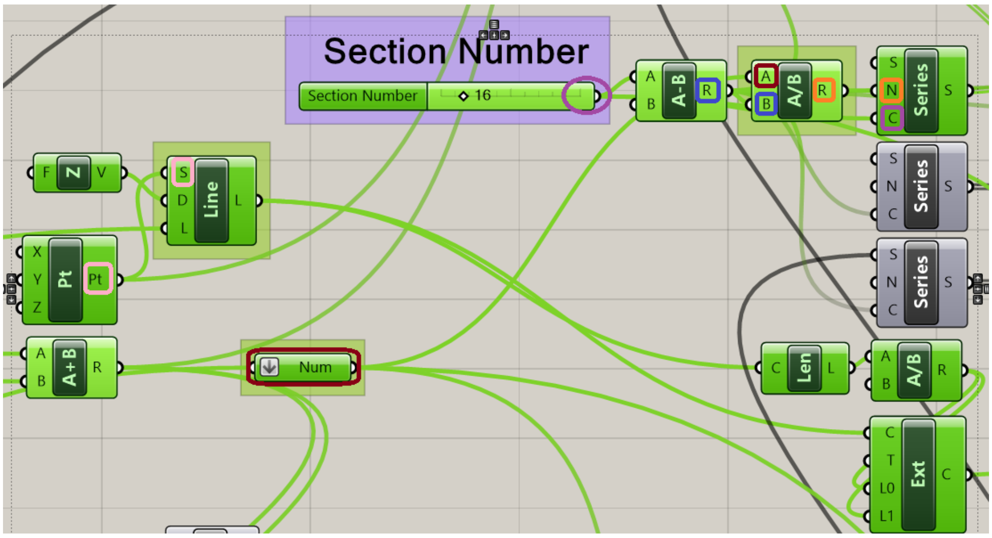


Figure 4.47 : Command chain of obtaining the extended line parallel to z axis.

Output of this command is connected to a Division command's A input via a Length command. B input of the same Division command is set as 10 and its output is connected to Start and End inputs of an Extend Curve command, whose Type input is set as Line. Output of Line SDL command is connected to Curve input of the same Extend Curve command. Output of Extend Curve command is connected to Geometry input of a Move command. Outputs of Subtraction and Division commands that are placed after the section number slider are connected to Count and Step inputs of a Series command with its Start input set as zero, respectively. Output of this Series command is connected to Motion input of the same Move command via a Unit X command, to whose Geometry input the output of Extend Curve command is connected. By this command chain, the line array through x direction, which will be projected onto surface to obtain section curves are obtained, see Figure 4.48 and Figure 4.49. Extensions are made due to avoid errors that may occur during projection process and for this case 15 lines are created to satisfy the conditions of three step point generation procedure, as mentioned before.

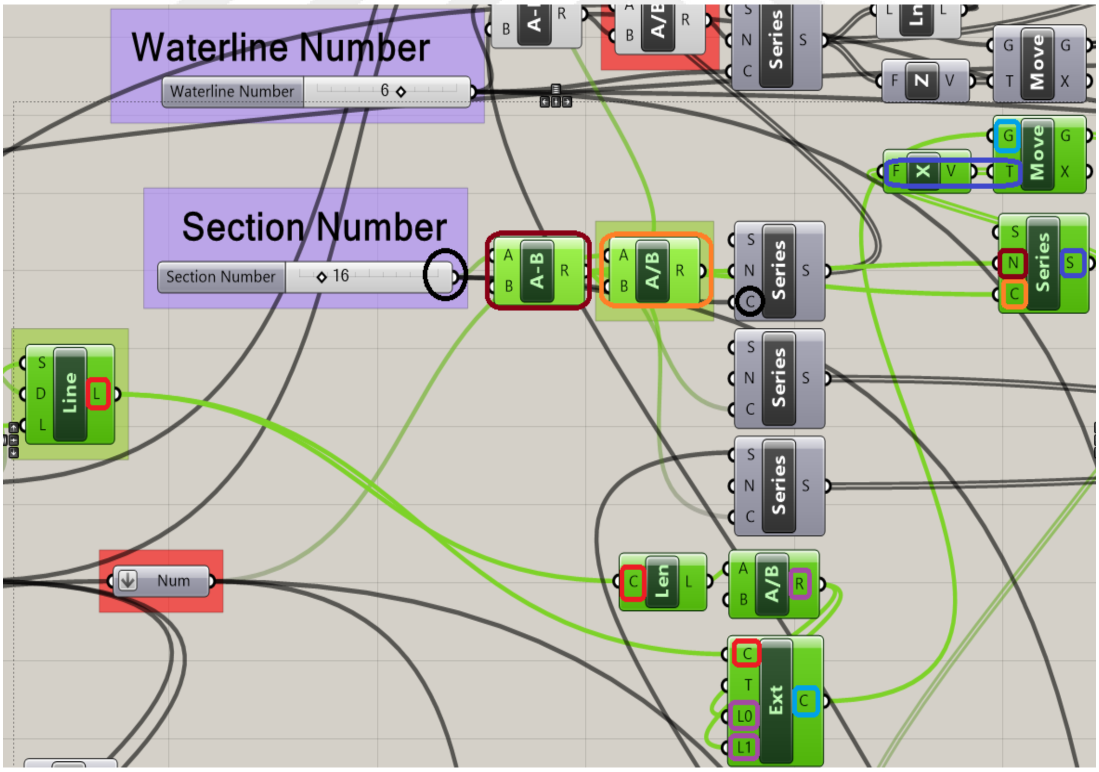


Figure 4.48 : Command chain of building and sorting the extended vertical lines array.

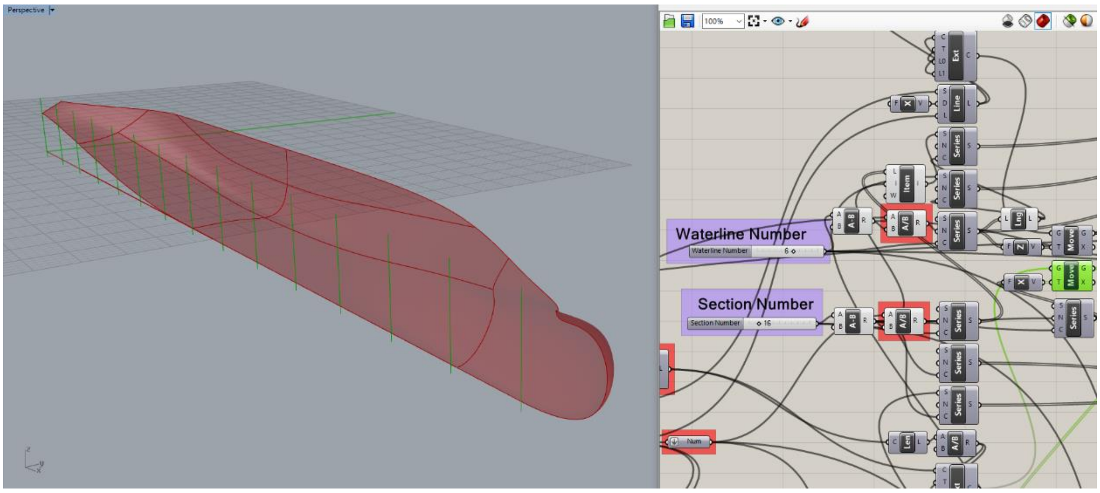


Figure 4.49 : Command chain and view of the projecting lines.

The five List Item commands which include the parts of the underwater surface are connected to five separate Brep commands. Aim here is to avoid possible errors and the complication on the Grasshopper canvas, see figure 4.50.

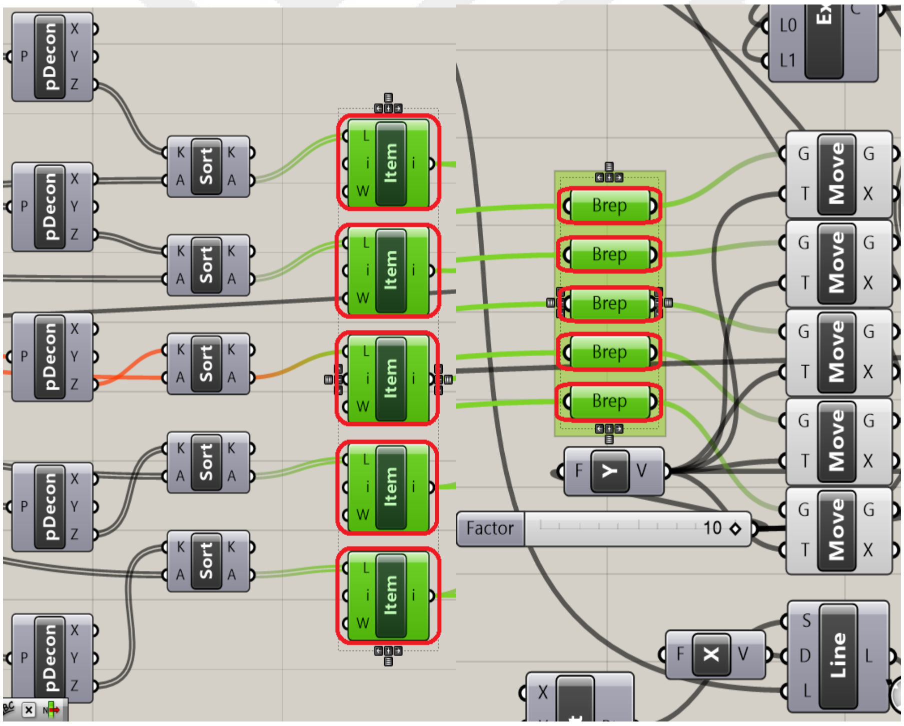


Figure 4.50 : Command chain of transferring the underwater surface pieces to another location.

Outputs of these seperate Brep commands are connected to Geometry input of five seperate Move commands. A number slider specified as "Factor" with default value as 10 is connected to Motion inputs of these Move commands via a Unit Y command. This replacement process has also been decided to make not to face unexpected errors during line projection to surfaces. Outputs of these Move commands are connected to a Project command’s Brep input and flattened, see Figure 4.51.

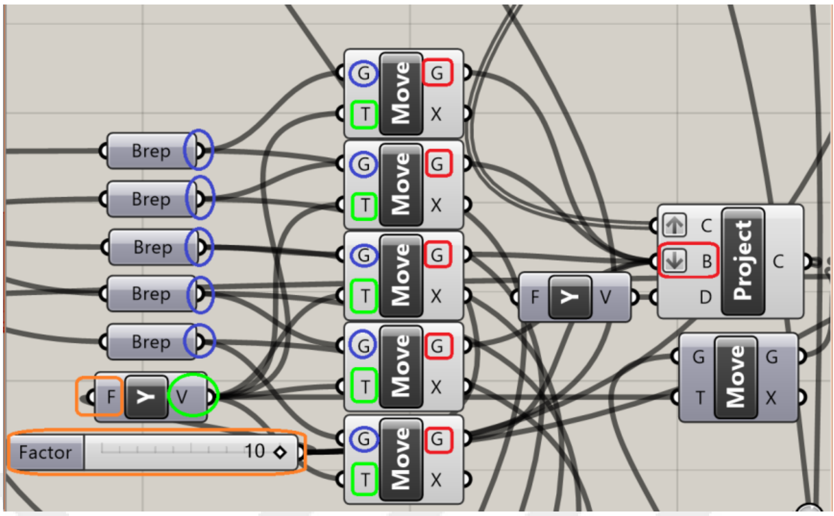


Figure 4.51 : Command connections of preparing the surface componenets for projection.

Output of Move command which includes the line array prepared for projection onto breps are connected to same Project command’s Curve input and grafted. As it is shown in Figure 4.52, a Unit Y command is connected to Direction input of the same Project command, who represents the section curves.

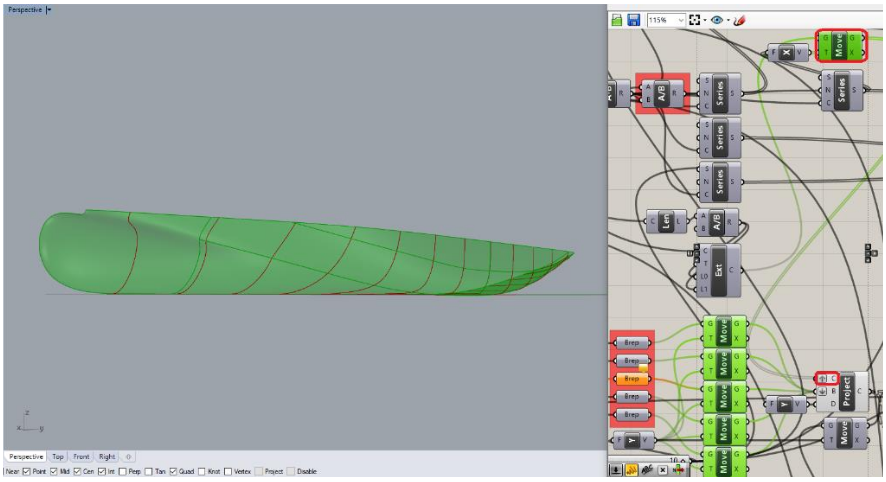


Figure 4.52 : Connecting the line array to project command and obtaining the section curves.

Trying to intersect the planar surfaces with section curves will not provide all the desired control points in every hull form, because for some hull forms and specific waterline numbers the planar surfaces may not find any curves to intersect, through the aft part of the hull form due to very low section volume for instance. To be able to obtain points even in these cases, vertical lines should be added to the end points of the section curves that have their y axis values zero and these lines should be intersected with the planar surfaces also. To satisfy this case, a Construct Point command is created with all input values set to zero, which represents the Rhinoceros origin point. Output of this Construct Point command is connected to Start input of two Line SDL commands. Output of Number command which represents the final overall length of the underwater hull form is connected to one Line SDL command's Length input, to whose Direction input a Unit X command is connected. Output of Draught number slider is connected to the other Line SDL command's Length input, to whose Direction input a Unit Z command is connected, see Figure 4.53.

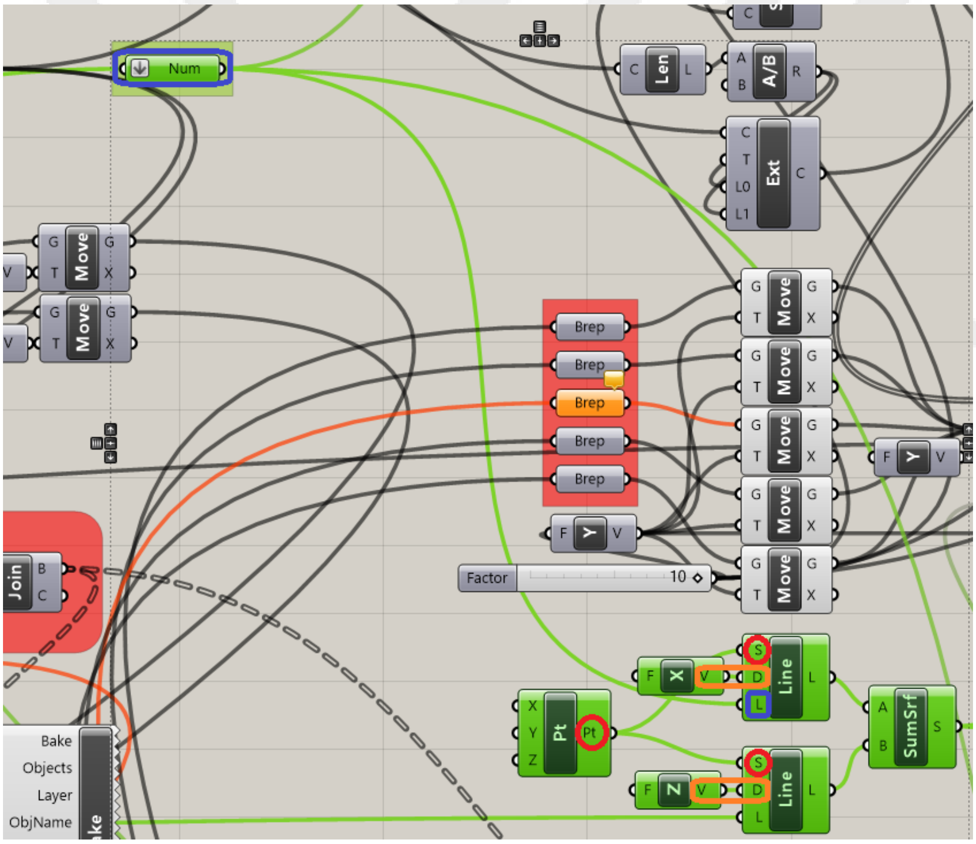


Figure 4.53 : Command sequence of obtaining the sum surface.

Outputs of two Line SDL commands are connected to a Sum Surface command's A and B inputs seperately. Output of the Sum Surface command is connected to a Move command's Geometry input, to whose Motion input the factor number slider is connected via a Unit Y command. Output of this Move command and the Project command that represents the section lines are connected to Surface and Curve inputs of a Surface/Curve command respectively. By this command chain the point cloud from which the vertical lines will be created is obtained, see Figure 4.54.

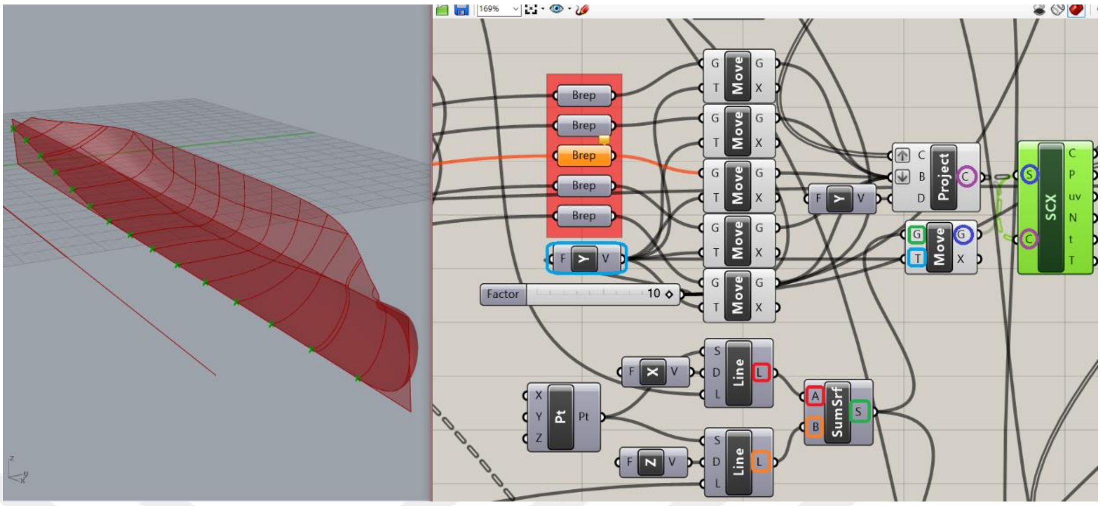


Figure 4.54 : Command sequence of obtaining the end points of section curves.

Point output of Surface/Curve command is connected to Start input of a Line SDL command, to whose Direction input a Unit Z command is connected via a Negative command. Z output of “Main Dimensions” part is connected to an Addition command’s A input, whose B input is set as 1 and output is connected to Length input of the Line SDL command. Outputs of Line SDL and Project commands are connected to seperate Curve commands. Outputs of the Curve commands are connected to Curve input of a Join Curves command and flattened. By this command chain the section curves and extension lines are obtained and merged together, see Figure 4.55.

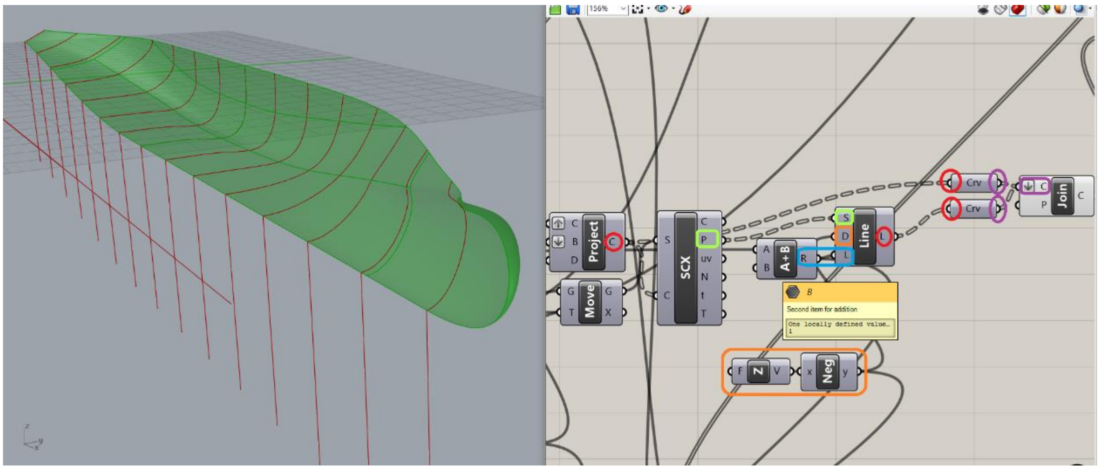


Figure 4.55 : Command sequence of obtaining the extension lines and joining them with section curves.

Output of the Extrude command which represents the planar surfaces is connected to a Move command's Geometry input. Factor number slider is connected to Motion input of the same Move command via a Unit Y command to move the planar surfaces through positive y direction with the same amount the hull form parts have been moved before. Outputs of this Move command and Join Curves command are connected to Surface and Curve inputs of a Surface/Curve command, with its Curve input grafted. By this command chain section curves with extension lines and planar surfaces are intersected and the control points are obtained, see Figure 4.56.

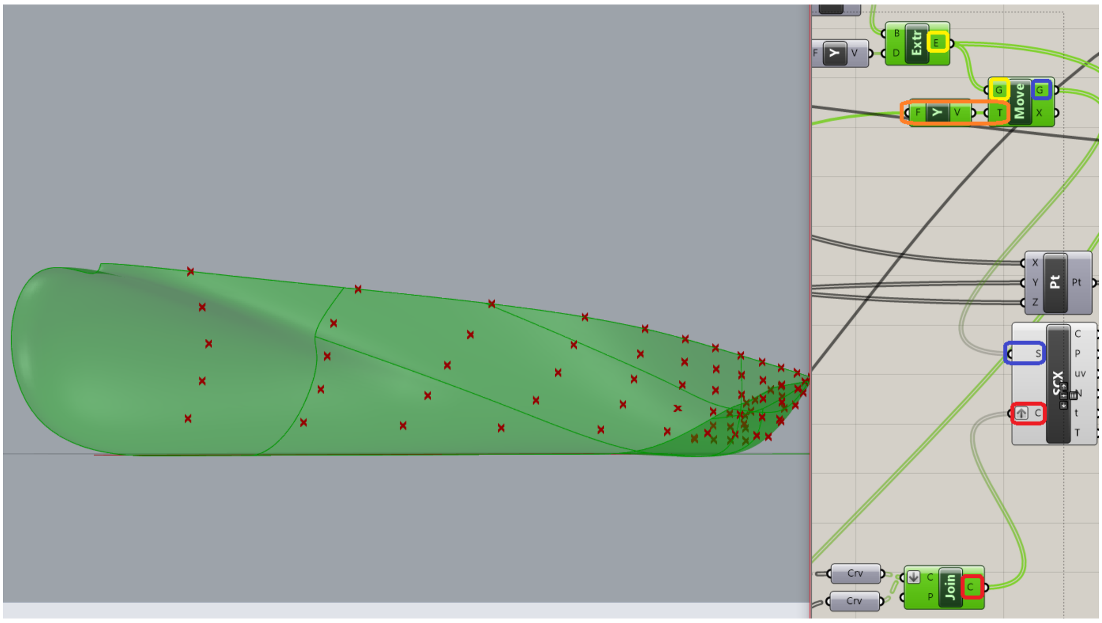


Figure 4.56 : Command sequence of intersecting the plane surfaces and section curves with extension lines and obtaining the control points.

4.8.3 Building the fore section and first waterline curve points

Output of Series command which is placed after section number slider, Subtraction and Division commands is connected to List input of a List Item command. Output of mentioned Subtraction command is connected to Index input of the same List item command. Output of this List Item command is connected to Start input of a Series command, whose Step input is set as zero and Count input is connected with the output of Series command via a List Length command which is placed after waterline number slider, Subtraction and Division commands. Another Series command is created and its Start input is connected with factor number slider, Step input is set as zero and Count input is connected with the Series command via a List Length command, see Figure 4.57.

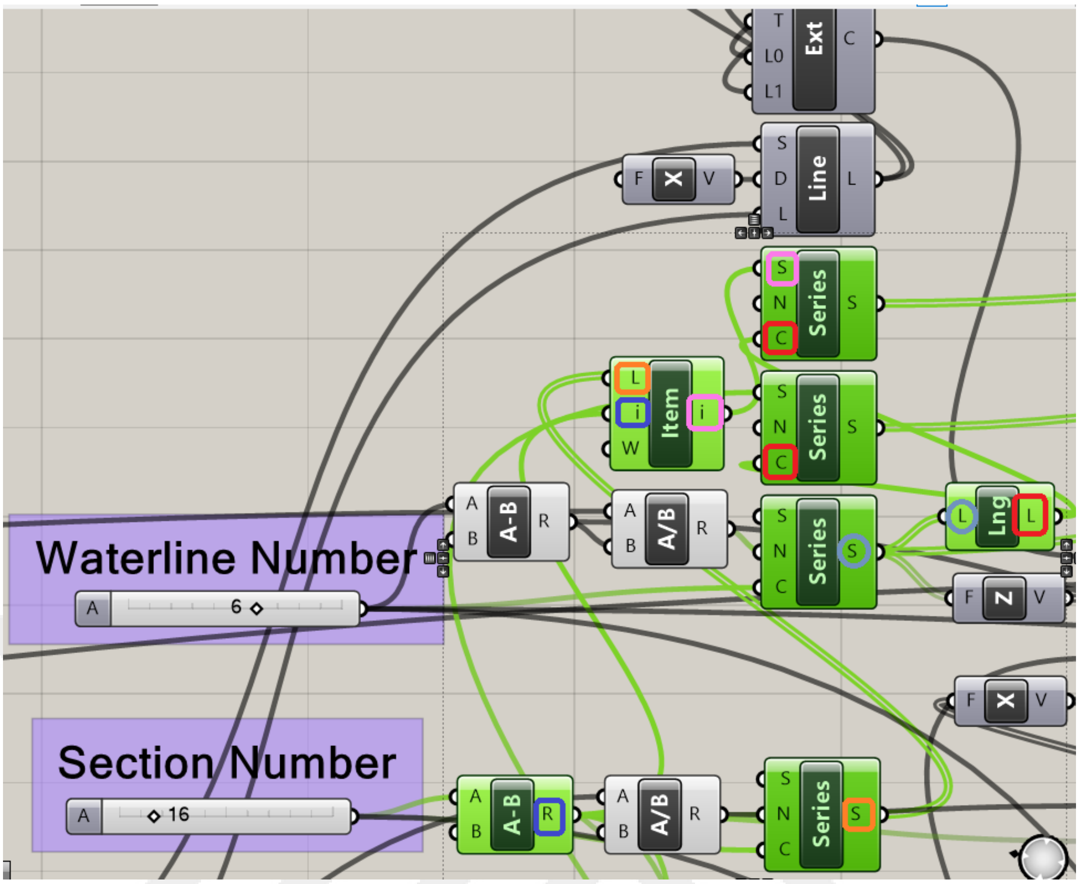


Figure 4.57 : Command sequence of arranging the coordinate points of last section curve.

As it is shown in Figure 4.58, outputs of these Series commands are connected to x, y and z inputs of a Construct Point command by the order of their placements downwards way, matching the inputs. By this command chain the point list with the biggest x values are obtained.

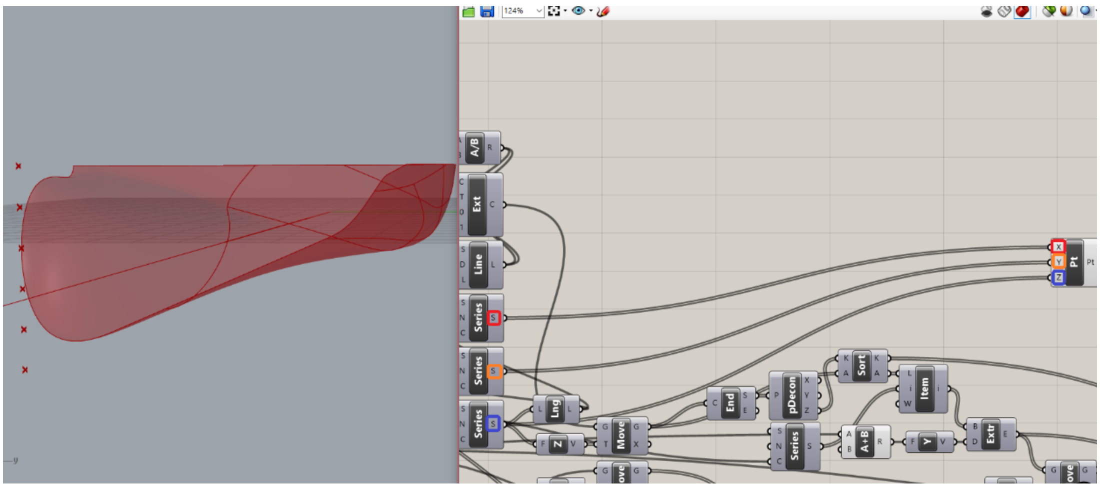


Figure 4.58 : Building the fore section curve points by connecting coordinates to Consturct point command.

Output of Series command which is placed before the Move command that includes the line array is connected to x input of a Construct Point command. Output of Subtraction command which is placed after section number slider is connected to Count inputs of two seperate Series commands. Both Step inputs of these Series commands are set to zero and one also has its Start input zero, while another one has its Start input connected with factor number slider. Outputs of these two Series commands are connected to a Construct Point command’s z and y inputs respectively. By this command chain the point list with z = 0 is obtained, see Figure 4.59.

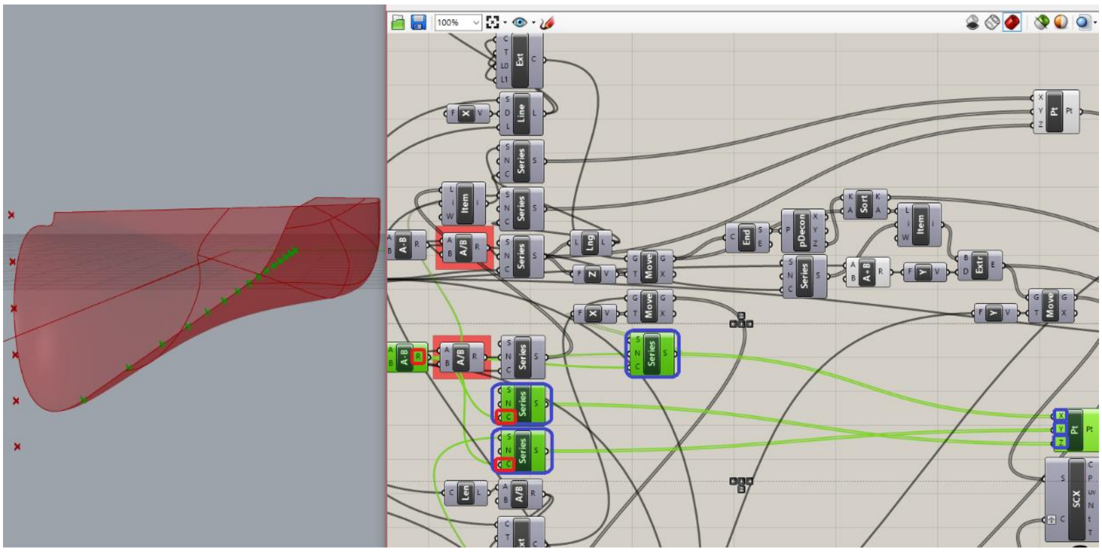


Figure 4.59 : Building the first waterline curve points by connecting the coordinates to Construct Point command.

Outputs of latest obtained two Construct Point commands and the Surface/Curve command are connected to a Point command and flattened. Output of this Point command is connected to input of one List Length command and one Deconstruct command. Output of List Length command is connected to Count input of a Series command with its Step input set to zero and Start input is connected with factor number slider. Output of this Series command is connected to B input of a Subtraction command, whose A input is connected with Y output of Deconstruct command. Output of Subtraction command is connected to y input of a Construct Point command, where the x and z outputs of Deconstruct command is directly connected to x and z inputs of the same Construct Point command, see Figure 4.60. By this command chain the effect of surface movement on the change of y values of the points that have been made previously have been reversed.

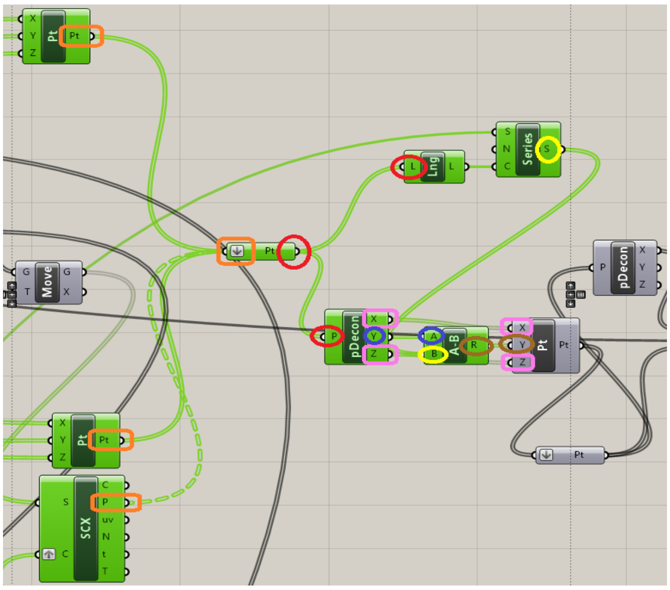


Figure 4.60 : Command chain of subtracting back the added value to y components of control points.

Output of this Construct Point command is is connected to a Point command and flattened, whose output is connected to one Deconstruct and one Sort List command. X output of Deconstruct command is connected to Key input of the same Sort List command, whose Value output is connected to List input of a Partition List command. Size input of Partition List command is connected with waterline number slider command. Output of partition list is again connected to one Deconstruct and one Sort List command. Z output of Deconstruct command is connected to Key input of Sort List command, see Figure 4.61. By this command chain all points are arranged according to increasing x and z values in groups of sixes.

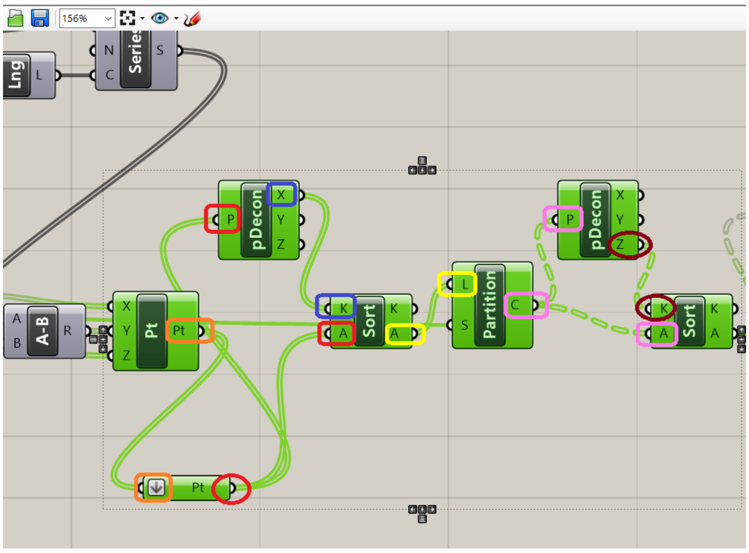


Figure 4.61 : Command chain of sorting the points based on increasing x component values.

4.9 Building the Input Data for Michell.Exe With GhPython Command

Michell.exe requires four input files which are named as “input halfbreadth”, “input parameter”, “input section” and “input waterline”. Input halfbreadth.dat includes y axis values of obtained points according to increasing x and z values. Input parameter.dat includes waterline numbers, section numbers, length of wetted line, maximum underwater breadth, draught and speed informations. Input Section.dat and Input Waterline.dat include x and z axis values of obtained points respectively. All of the obtained data until this point are arranged suitable to these input files. Data arranging and exporting is made with GhPython command.

4.9.1 Arranging the data for halfbreadth and section inputs

Values output of Sort List command is connected to a Panel command, by which all the points with x y and z coordinates can be monitored as groups of sixes in each branch. Same output of Sort List command is connected to a Deconstruct command. X and y outputs of Deconstruct commands are both connected to one Panel and one Number commands each. By panel commands, only x and y coordinates of all the points can be monitored seperately, six in each branch, see Figure 4.62. Number commands are connected to x inputs of two different Python Script commands, whose x inputs are specified as “numbers”.

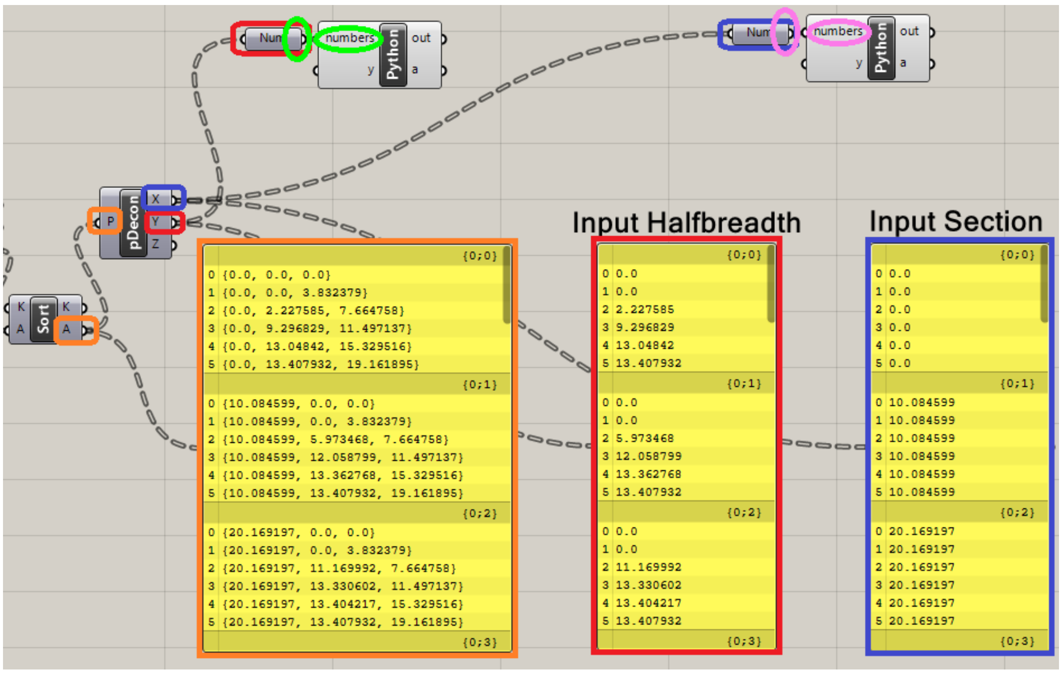


Figure 4.62 : Command chain of connecting the components of points to Python Script and Panel commands.

4.9.2 Arranging the data for parameter and waterline inputs

Section number slider and waterline number slider commands are connected to seperate Panel commands placed over Breadth and Depth Panel commands, specified as "Section Number" and "Waterline Number" by two Scribble commands. Number command which represents final overall length of the underwater hull form is connected to a Number command and flattened. Output of the number command is connected to a Panel command and specified as "LwL", see Figure 4.63. Another Panel command is created and specified as "Speed" and set as 8 by default, to let users enter the cruise speed of their vessels. Six Panel commands are connected to a Number command by order and output of this Number command is connected to a Panel command and x input of a Python Script command, which is specified as "numbers". Key output of Sort List command placed before the latest Extrude command is connected to a Number command. As it is shown in Figure 4.64, output of this Number command is also connected to one Panel command and x input of a Python Script command, which is again specified as "numbers".

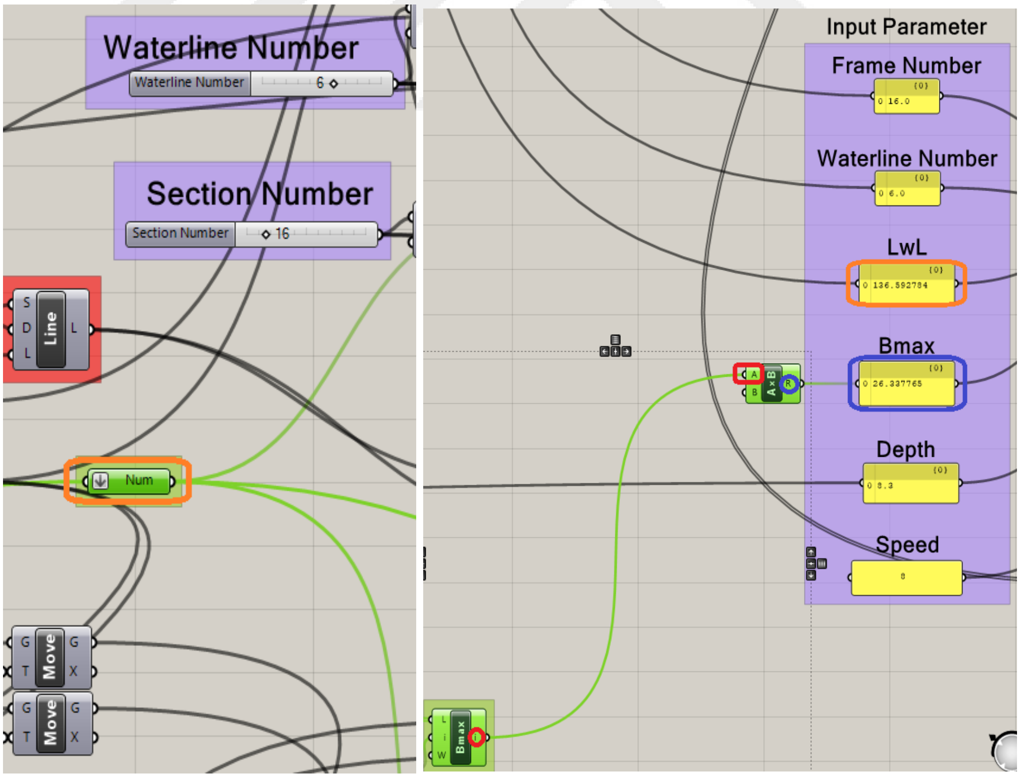


Figure 4.63 : Connecting the overall length and breadth data to Panel command.

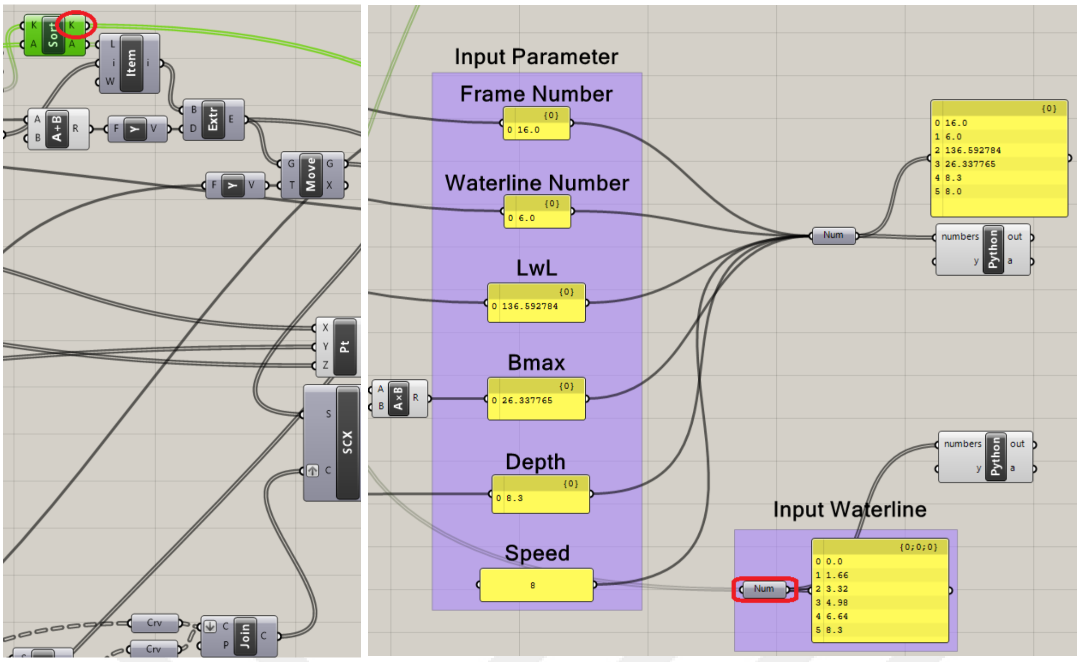


Figure 4.64 : Command chain of connecting waterline and parameter values to Python Script commands for arrangement.

4.9.3 Python scripts

Four Python Script commands include Python codes inside to arrange the output data in the way Michell.exe requires. Python Scrpit command whose input is y components of points reverses sorting of the sublists, replacing the first section with the last one and arranging it into a 6x16 matrix for default amounts of section and waterline numbers, see Figure 4.65. Python Script command whose input is x components of points picks one value from between 6 equal values from each sublist and arranges into a 1x16 matrix for default amount of section numbers, see Figure 4.66. Python Script command whose input contains values of “Input Paramerter” part converts the data to 1x6 matrix without changing the sorting of values, see Figure 4.67. Python Script command whose x input is waterline data converts the list into a 1x6 matrix for default amount of waterline numbers, see Figure 4.68. All Python Script commands export the manipulated data inside to relevant .dat files. The code inside Python Script commands can be checked or edited by right clicking on Python Script command and proceeding on “Open Editor” tab. It should be noted that x inputs of Python Script commands and the code indicating the data tree inside the editor should be named the same. Also the x inputs of Python Script commands should be set as “Tree Access” by right clicking on x input and proceeding to relevant tab.

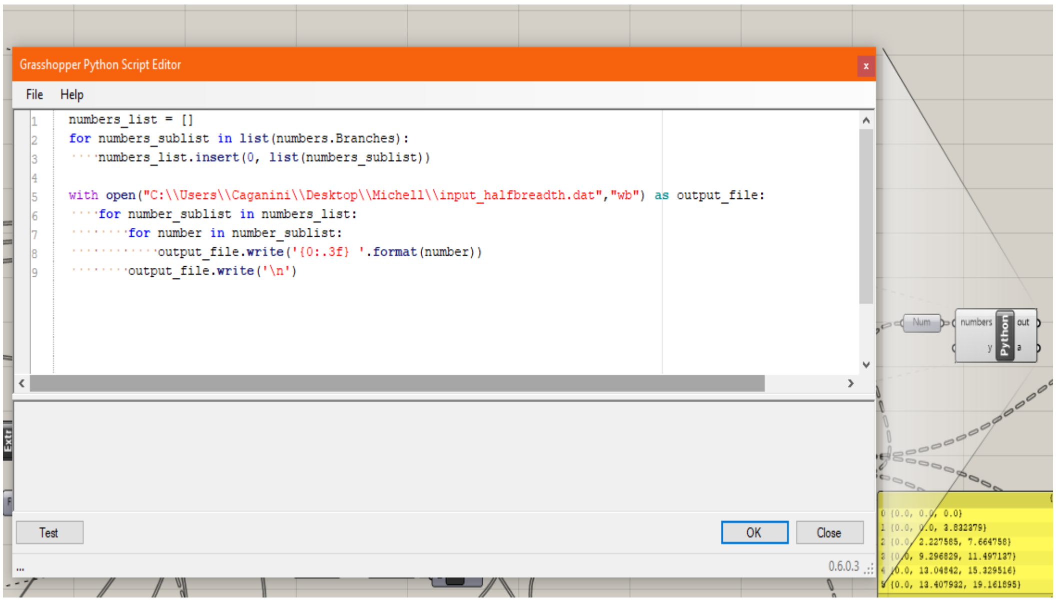


Figure 4.65 : Python script which arranges data required for “input_halfbreadth.dat” file.

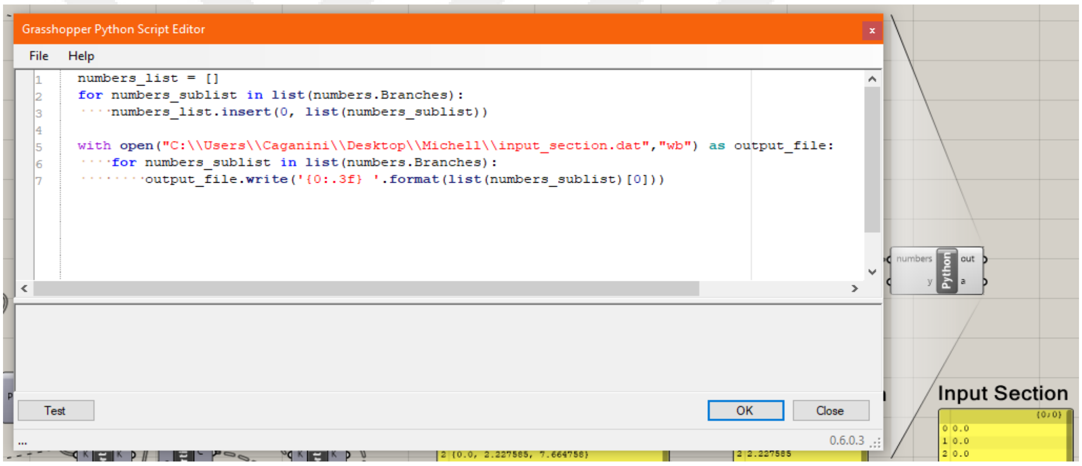


Figure 4.66 : Python script which arranges data required for “input_section.dat” file.

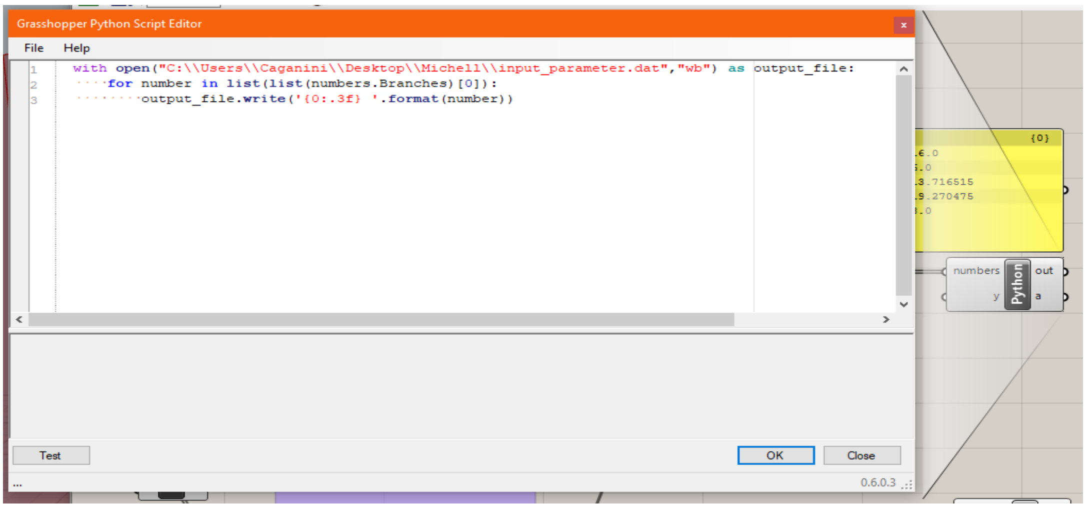


Figure 4.67 : Python Script command which arranges data required for “input_parameter.dat” file.

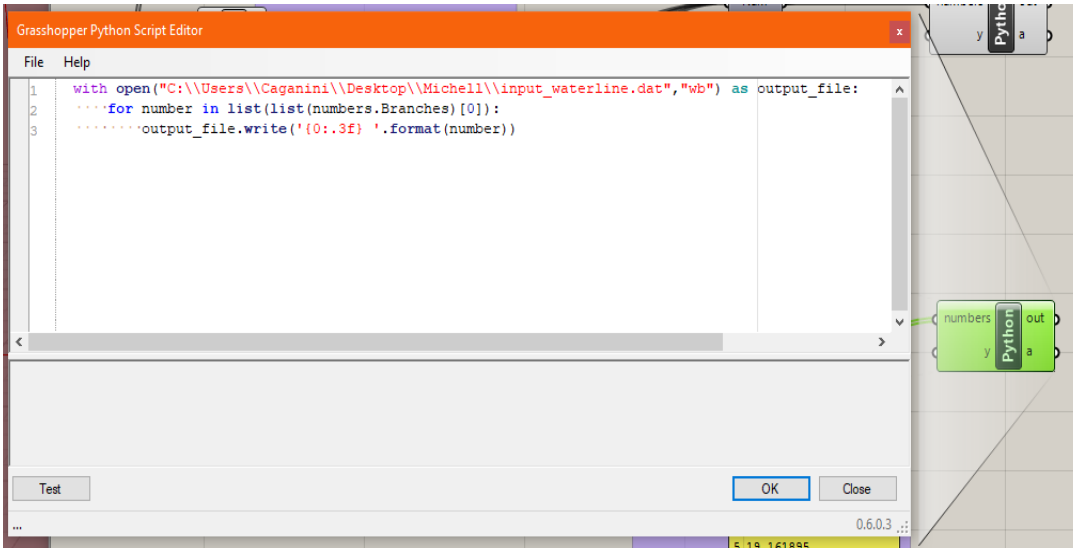


Figure 4.68 : Python Script command which arranges data required for "input_waterline.dat" file.

Every single change that effects the shape of hull form (e.g. changes in number sliders, Point on Curve commands etc.) will be transferred to final output data collection and also to the output files of Python Script commands. Michell.exe has not been adjusted to run in every single parameter change, because in case a number slider is used not by double clicking and entering a value but moving the slider, too many calculations will be included in total output and this will result in unnecessary amounts of resistance predictions in a list. Henceforth it is left to users to determine when to use the test button in Remote Control Panel and predict the resistance of the hull form.

4.10 Aft Part Custom Section Curve

The Grasshopper code has an additional sub-part inside, which is made for a specific aft section analysing problem data exportation. In some cases control point obtaining process may fail especially at the aft part of the hull form. This is a problem which occurs at curve projection process. Since control points of first waterline are calculated seperately, one less number of total waterlines should be intersected with the first section, which is the rearmost one. To overcome this, a custom line is created where x=0 and adjusted to be moved by users for a small amount through positive x axis, then projected to the surface.

First, a Construct Point command is created with Y and Z inputs set to zero and a number slider is connected to its X input, specified as "x coordinate". Output of

Construct Point command is connected to a Line SDL command's Start input, to whose Direction and Length inputs a Unit Z and a Number Slider commands are connected. The Number Slider is set to 20 by default, its limit and current values may need a change depending on the imported surface, to make sure the whole line intersects with the hull surface. Output of Line SDL command is connected to Curve input of a Project command, to whose Brep and Direction commands the Brep command that includes the underwater surface and a Unit Y command are connected respectively, see Figure 4.69. The Project command draws the section curve and its position can be controlled by the x coordinate number slider. Larger values of the number slider will lead to larger error margines.

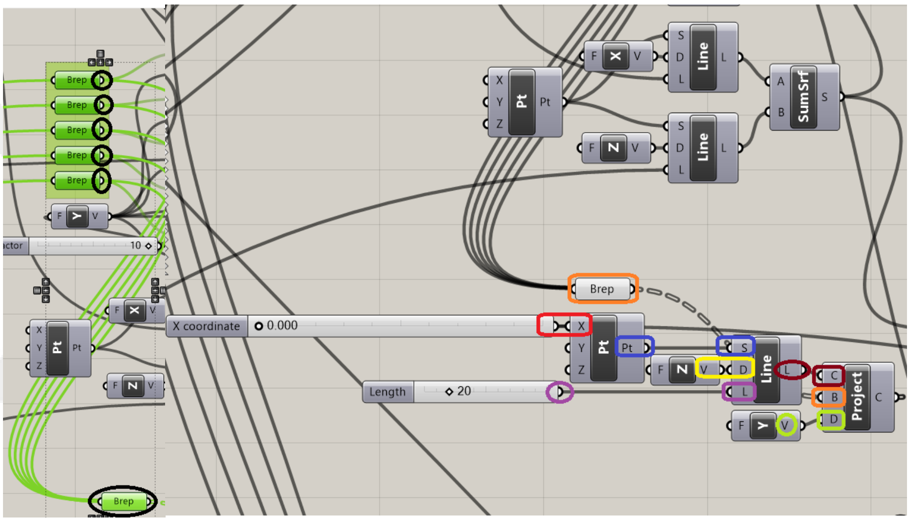


Figure 4.69 : Command chain of generating the custom section curve.

Output of Project command is connected to a Surface/Curve command's Curve input, to whose Surface input the ouıtput of Sum Surface command is connected. Point output of Surface/Curve command is connected to a Line SDL command. Output of Addition command used to calculate the length of extension lines to section curves is connected to Length input of the same Line SDL command, to whose Direction input the combination of Unit Z and Negative commands are connected. Outputs of Project and Line SDL commands are connected to Curves input of a Join Curves command and flattened. Output of Join Curves command is connected to Curve input of a Surface/Curve command, to whose Surface input the output of Extrude command that represents the planar surfaces is connected. Point output of Surface/Curve command is connected to one Deconstruct and one List Length command. Output of List Length command is connected to Count input of a

Series command, whose step input is set to zero. X coordinate number slider is connected to Start input of the same Series command, see Figure 4.70.

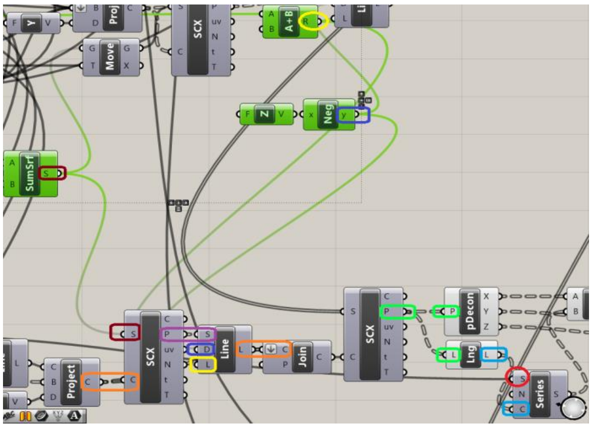


Figure 4.70 : Command chain of joining the custom section curve and extension line together.

X output of Deconstruct command is connected to A input of a Subtraction command, to whose B input the output of Series command is connected. Output of Subtraction command is connected to X input of a Construct Point command, to whose Y and Z inputs Y and Z outputs of the Deconstruct command are connected respectively, see Figure 4.71. By this command chain, a section and its extension line with user defined x axis position is obtained, intersected with planar surfaces, and replaced back to its supposed position by subtracting the user defined amount of X values.

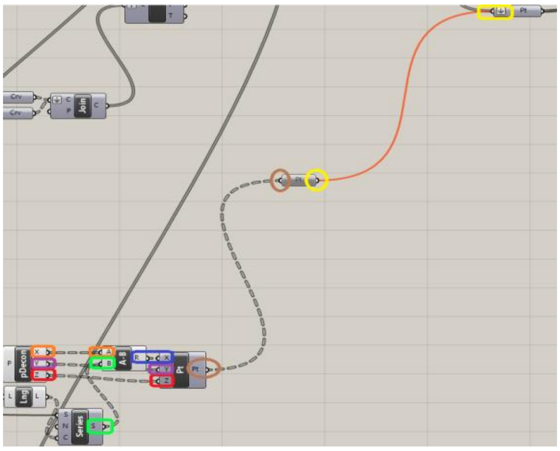


Figure 4.71 : Command chain of obtaining the custom section points and counting in.

To add these point coordinates to other calculated points list, output of Construct Point command is connected to a Point command first. This command is disabled by right clicking on it and selecting “enabled” tab, then connected to Point command which contains previously calculated points. This part of the code can be activated by enabling the last Point command, see Figure 4.72.

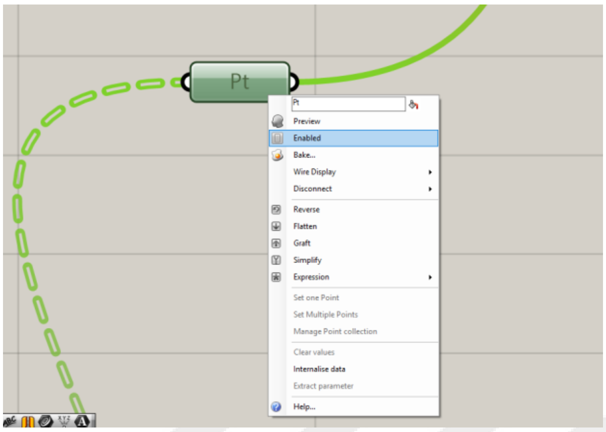


Figure 4.72 : Enabling and disabling a command in Grasshopper.

4.11 Exporting the Final Surface to Rhinoceros and Creating the Link With Michell.Exe

To export the finalized hull surface to Rhinoceros and also a custom preview of it, Brep output of Brep Join comand is connected to Objects input of an Object Bake command. A Boolean Toggle command and Colour Swatch command are connected to Bake and Object Colour inputs of the Object Bake command, see Figure 4.73. When the Boolean Toggle is set to “True” by double clicking on the button, the hull form will be exported to Rhinoceros. Also the colour of the hull form can be edited by clicking the Colour Swatch command.

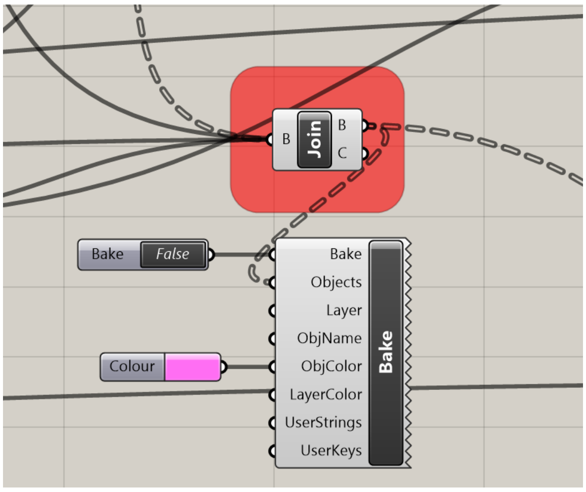


Figure 4.73 : Command connections of surface export and colouring.

Also a Launch Application command is created and a Boolean Toggle is connected to its Launch input. A File Path command is created and connected to Path input of the Launch Application command. As it is shown in Figure 4.74, Michell.exe is connected to the File Path command by right clicking on the command, selecting “set one file path” tab, and setting the program. By double clicking on Boolean Toggle command, the Michell.exe program can be run.

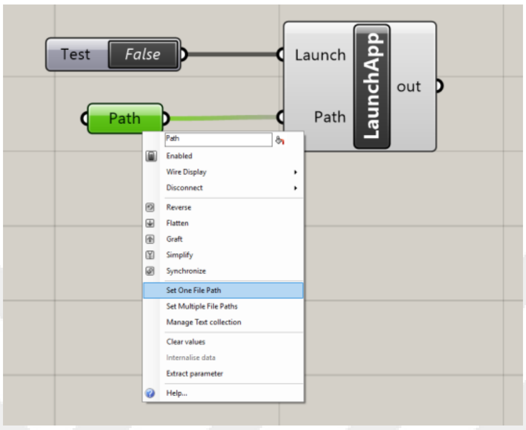


Figure 4.74 : Command connections of launching an external application and setting a file path.

4.12 Creating the User Interface

The whole code has a complicated view on Grasshopper canvas and does not provide an ease of use. To overcome this, all of the possible commands which are basically number sliders and toggle commands, are published to Remote Control Panel which is provided within Grasshopper. This has been done by right clicking on all of the desired commands and selecting “publish to remote panel” tab, see Figure 4.75.

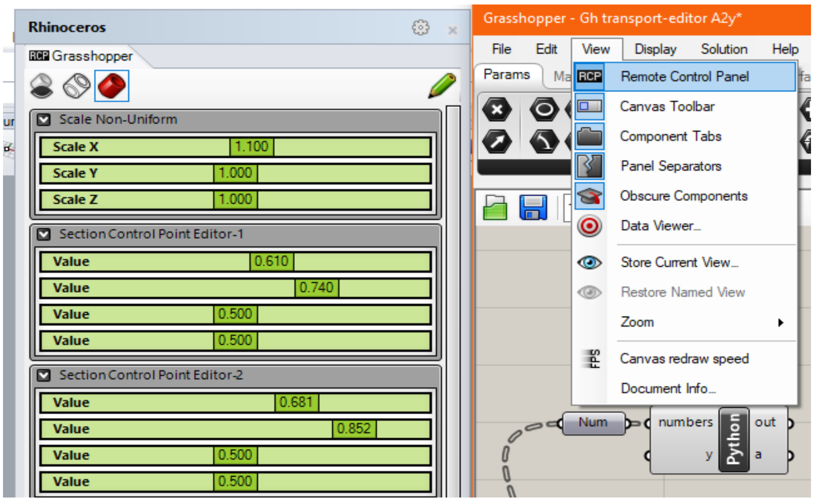


Figure 4.75 : Adding a remote control panel.

Unfortunately the Point on Curve commands are not supported for the used version of Grasshopper. To provide a better user interface, all the Point on Curve commands are grouped and named within theirselves and collected together by an top down array, so that users can reach all of the controlling commands within the same program window, which is shown in Figure 4.76.

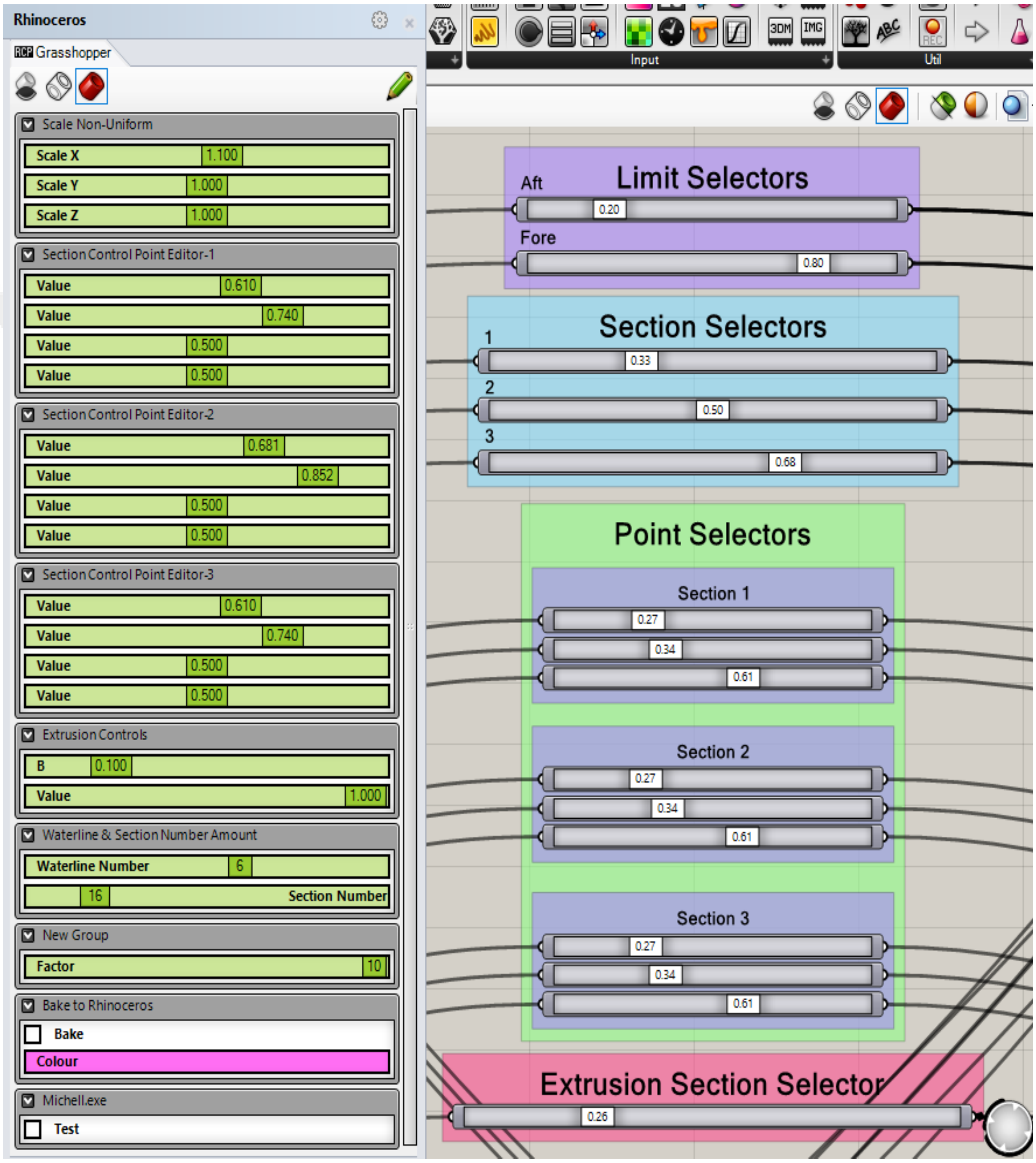


Figure 4.76 : Aggregated parameter controllers.

5. APPLICATONS

In this section a hull form will be subjected to several surface manipulations and resistance tests. The hull form is generated by free rapid hull modeling and fairing script which is provided at food4Rhino web page. This hull form is subjected to three different presets of the developed Grasshopper code. Also the resistance predictions made with michell.exe of varying presets will be added to results in this section.

Features of the Grasshopper code have been used in compatibile ways at each specific preset to animate realistic applications of surface manipulations.

Manipulation controllers will be provided both within Grasshopper canvas and Remote Control Panel. Controllers on the Grasshopper canvas are Point on Curve commands and they represent a point on a line or a curve by means of a percentage of its length. There are four groups of controllers specified as Limit Selectors, Section Selectors, Point Selectors and Extrusion Section Selector. Limit Selectors represent the interval along x axis, within which interval the manipulation will be applied, or after and before which points the aft and fore part of the body will be removed. Section Selectors represent the locations of three sections which will be edited, along x axis within the extracted mid body. Point Selectors represent the locations of three editable points along each selected section curve. Extrusion Section Selector represents the location of the section where the linear extrusion will be started through positive x axis, within the extracted mid body, along x axis.

Controllers on the remote control panel are number sliders, boolean toggles and colour swatches. Number sliders represent numerical values, but because of used Remap Numbers and Construct Domain commands, some of them represent percentages and proportional values. Boolean toggles represent buttons to activate a command group or code. Colour swatches let changing a colour of an object or an item.

There are 9 groups of controllers specified as Scale Non-Uniform, Section Control Point Editor-1, Section Control Point Editor-2, Section Control Point Editor-3, Extrusion Controls, Waterline&Section Number Amount, Factor, Bake to Rhinoceros and Michell.exe. Scale Non-Uniform controllers represent the scaling factors for x, y and z axises for the hull form by means of percentages. Section Control Point Editors represent the movements in positive and negative y axis of preselected points and the intersection point of highest waterline and related section curve, by means of attained percentage. Extrusion controls represent the setting of maximum extrusion amount as a function of overall length and the amount of linear extrusion within selected boundaries. Waterline&Section Number Amount represents the amount of waterlines and sections by which the required value lists will be formed. Bake to Rhinoceros provide an exportation of final hull form to Rhinoceros with a desired colour. Michell.exe provide resistance prediction of finalized hull form.

Instead of applying all features of the program to the hull form at the same time, varying modifications have been done separately. By means of scale non-uniform, x parameter is adjusted as 1.02, y parameter is adjusted to 0.98 while z parameter stays constant. For section editing, all parameter groups of four are adjusted with an order of 0.5, 0.55, 0.65 and 0.7. For linear extrusion the extrusion amount is set to 0.436.

All of the presets for the hull form and the comparative figures which represent before and after manipulation schemes and michell results are provided in figures. Red coloured surfaces are edited hull forms, where blue coloured surfaces are the initials, see the following figures.

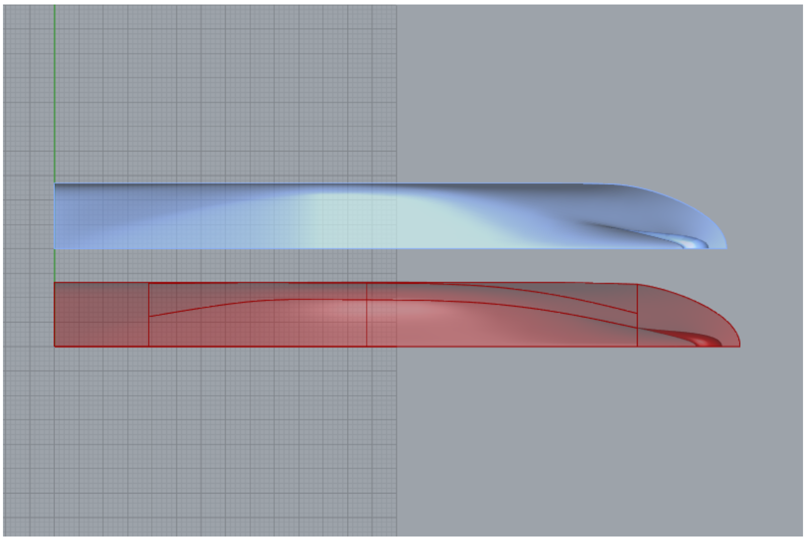


Figure 5.1 : Scale non-uniform modification top view.

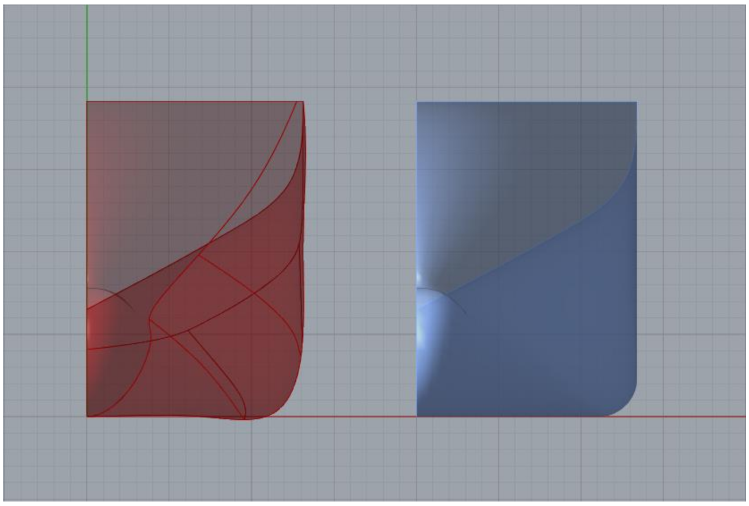


Figure 5.2 : Scale non-uniform modification front view.

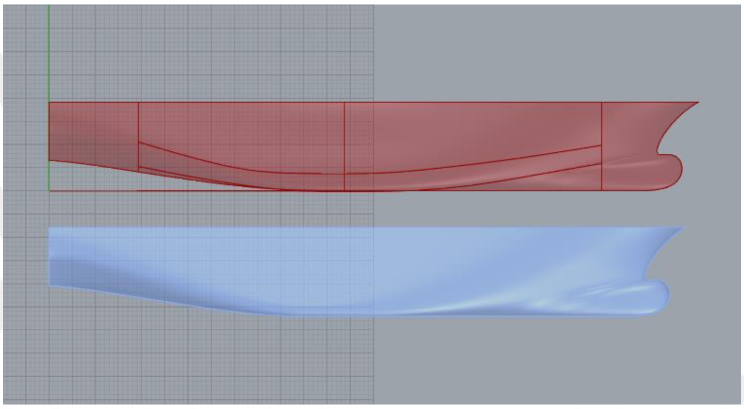


Figure 5.3 : Scale non-uniform modification right view.

```
PARTICULARS OF THE SHIP :

L=133.914 M
B= 26.806 M
T=  8.300 M
V=  8.000 M/SEC


WETTED SURFACE AREA=  8272.312 (SQUARE METERS)
CF=  .4413546D-01     RF=  .4179174D+06 (NEWTONS)

CW=  .1745559D-01     RW=  .1652865D+06 (NEWTONS)
```

```
PARTICULARS OF THE SHIP :

L=136.593 M
B= 26.338 M
T=  8.300 M
V=  8.000 M/SEC


WETTED SURFACE AREA=  8248.432 (SQUARE METERS)
CF=  .4638349D-01     RF=  .4156859D+06 (NEWTONS)

CW=  .1595321D-01     RW=  .1429716D+06 (NEWTONS)
```

Figure 5.4 : Comparison of original (above) and edited (below) resistance values of scale non-uniform modification obtained with michell.exe.

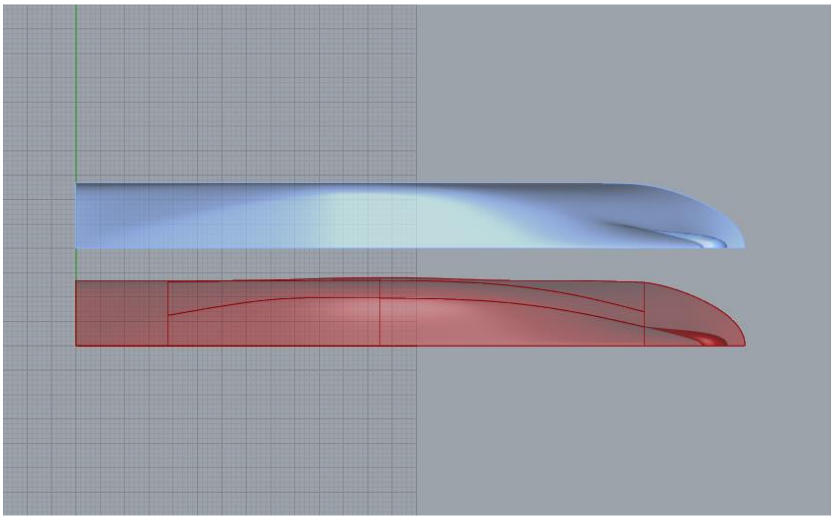


Figure 5.5 : Section editing modification top view.

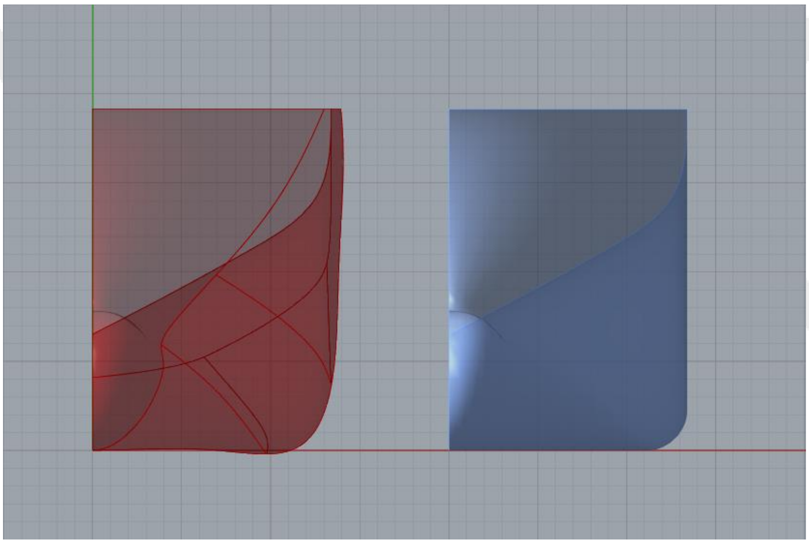


Figure 5.6 : Section editing modification front view.

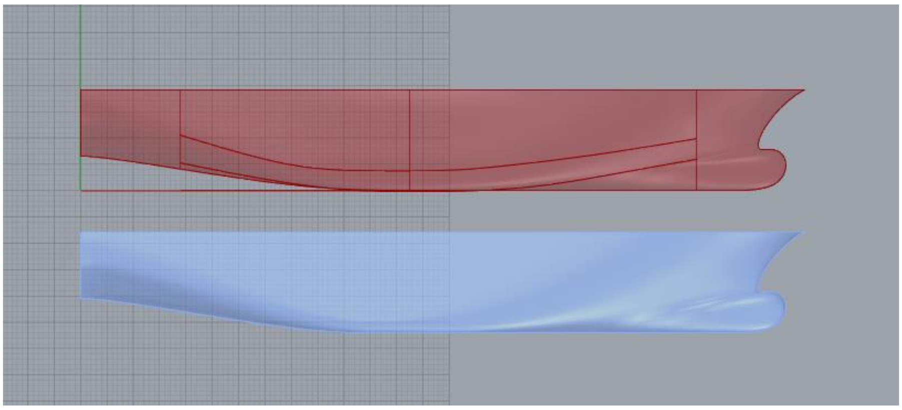


Figure 5.7 : Section editing modification right view.

```
PARTICULARS OF THE SHIP :

L=133.914 M
B= 26.806 M
T=  8.300 M
V=  8.000 M/SEC


WETTED SURFACE AREA=  8272.312 (SQUARE METERS)
CF=  .4413546D-01     RF=  .4179174D+06 (NEWTONS)

CW=  .1745559D-01     RW=  .1652865D+06 (NEWTONS)
```

```
PARTICULARS OF THE SHIP :

L=133.914 M
B= 27.542 M
T=  8.300 M
V=  8.000 M/SEC


WETTED SURFACE AREA=  8357.177 (SQUARE METERS)
CF=  .4223704D-01     RF=  .4222048D+06 (NEWTONS)

CW=  .1682790D-01     RW=  .1682130D+06 (NEWTONS)
```

Figure 5.8 : Comparison of original (above) and edited (below) resistance values of section editing modification obtained with michell.exe.

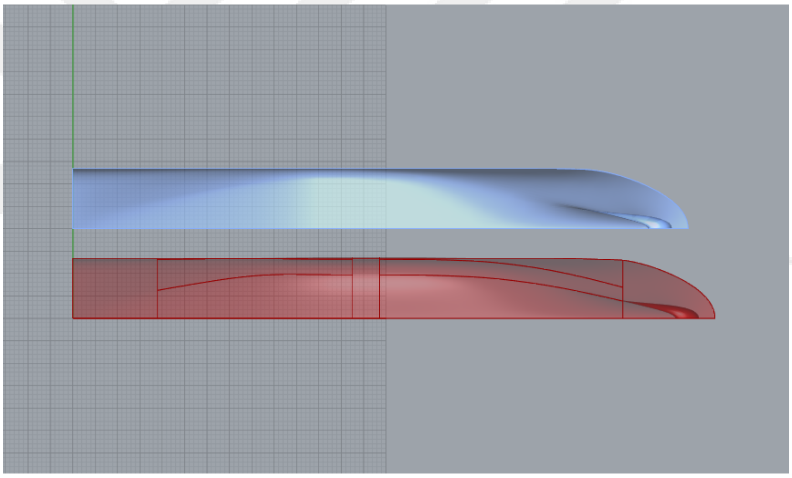


Figure 5.9 : Linear extrusion modification top view.

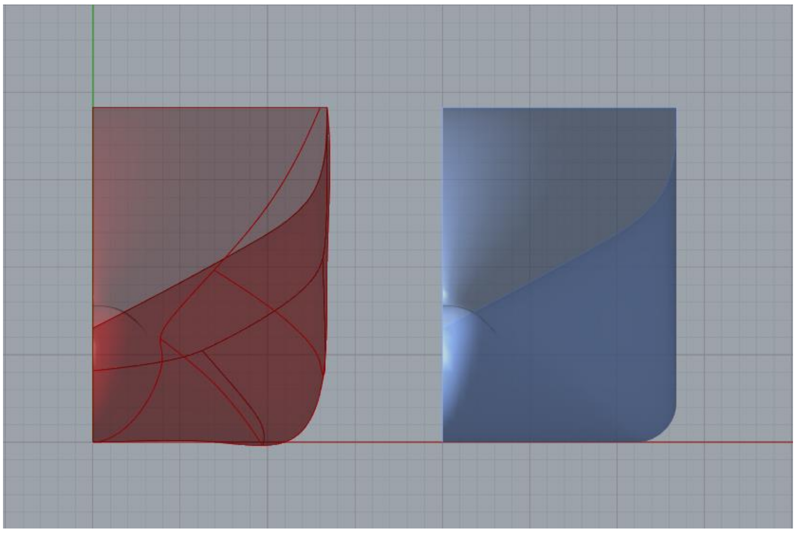


Figure 5.10 : Linear extrusion modification front view.

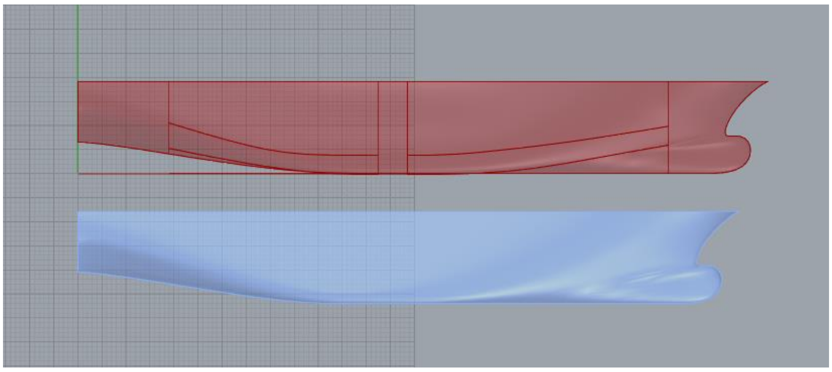


Figure 5.11 : Linear extrusion modification right view.

```
PARTICULARS OF THE SHIP :

L=133.914 M
B= 26.806 M
T=  8.300 M
V=  8.000 M/SEC


WETTED SURFACE AREA=  8272.312 (SQUARE METERS)
CF=  .4413546D-01    RF=  .4179174D+06 (NEWTONS)

CW=  .1745559D-01    RW=  .1652865D+06 (NEWTONS)
```

```
PARTICULARS OF THE SHIP :

L=139.910 M
B= 26.806 M
T=  8.300 M
V=  8.000 M/SEC


WETTED SURFACE AREA=  8917.692 (SQUARE METERS)
CF=  .4943930D-01    RF=  .4480767D+06 (NEWTONS)

CW=  .1726912D-01    RW=  .1565129D+06 (NEWTONS)
```

Figure 5.12 : Comparison of original (above) and edited (below) resistance values of linear extrusion modification obtained with michell.exe.

6. DISCUSSION

In this study a utility program which provides ship hull form manipulation and resistance prediction for initial design step is created and subjected to varying applications. In detail; surface importation and splitting, section editing and building new surfaces, adding new surface pieces, obtaining and analysing control points of a surface, arranging and exporting numerical data and creating an automated process of basic ship design steps have been achieved. Most frequently used Grasshopper operations included obtaining a section with projecting lines, intersecting curves and lines with surfaces, using bounding boxes for dimension analysis, using number series for geometry arrays and sorting geometry based on their properties.

Various problems have been faced during the study, some of which are limiting factors for the design process. Some of the problematic cases can be solved easily since the same cases also occur in Rhinoceros. Major issues occur when projection, intersecting and splitting operations are desired to be made. In some cases projecting a line onto a surface ends up in partial curves and projecting a point onto a surface completely fail. It is realized that these problems mostly occur when the surface and the object which is desired to project touch each other at one or more than one points. Thus placing the surface and objects away from each other became the solution in many cases. Also intersecting two curve groups may fail and the easiest solution for this case is using planar surfaces instead of one of the curve groups, only if one of the curve groups include all planar curves.

Splitting process can be done without any issues and changing the cutter geometry (line, curve, surface etc.) is usually the solution to the potential problems. But after a splitting is done, joining the surface pieces together with Brep Join component work as the group command in Rhinoceros, instead of forming a new one-piece surface. This case leads to problems when a group of geometry is desired to project onto the splitted surface pieces for instance.

One of the most important things is the need of ordering the objects based on a selected property before subjecting them to a process together. This is where the starting points of a line or curve group matters when it is desired to form a surface. If all of the curves in a curve array don’t have their starting points in similar side or they are connected to a surface building component with a random order, the result will be twisting and rounding surfaces instead of a continous and fair one.

Since the mentioned problematic cases are mostly very basic three dimensional design operations, it is obvious that Grasshopper requires a different design perpective. This is the point where using other software for some steps of the design process might be advantageous. However, a well planned and built algorithm in Grasshopper provides a very powerful, customized and visualized solution for any design progress. Since changing properties of an already finalized 3D model can be tedious in some cases, building an algorithm for the desired work make it convenient for later changes. Especially for routine and repetitive processes, being able to build an algorithm specific to one custom problem is a very useful solution. Considering these opportunities contribute to reducing time loss and errors in design progress, using Grasshopper can be very beneficial when properly adapted to the desired workflow.

Constituted Grasshopper program has successfully achieved the non-uniform scaling, section editing based surface manipulation, linear extrusion and resistance estimation processes. Since these processes can be applied separately, the program can also be used for only one of its features such as resistance prediction. The reaction time to the changes made is dependent on the computer power and the way the algorithm has been built. However the required computer power is not at the scale of CFD softwares’. Considering these information and all of the opportunities the program provides, a useful tool has been obtained for the initial design step. Instead of removing desired amount of the aft and fore parts of the surface at the initial step, a more precise command chain that is able to handle the problems at the points of surface and curve discontinuities can be integrated to the process which also includes the bulbous bow modifications and other utilities such as predicting the location of longitudinal center of buoyancy, as the future work.

REFERENCES

[1] **Zelazny, K.,** "A method of calculation of ship resistance on calm water useful at preliminary stages of ship design", Scientific Journals, Vol. 38, pp. 125-130, 2014.

[2] **Lagios, K., Niemasz, J., Reinhart, C. F.,** "Animated Building Performance Simulation (ABPS) – Linking Rhinoceros/Grasshopper With Radiance/Daysim", (Bu conference talk)

[3] **Dejhalla, R., Vukovic, Z. M.,** "Application of Genetic Algorithm for Ship Hull Form Optimization", Int. Shipbuild. Progr., vol. 48, pp. 117-133, 2000.

[4] **Raven, H. C., Van der Ploeg, A., Starke, A.R.,** "Towards a CFD-Based Prediction of Shp Performance – Progress in Predicting Full-Scale Resistance and Scale Effects", Transactions of the Royal Institution of Naval Architects Part A: International Journal of Maritime Engineering,

[5] **URL-1** (https://engys.com/applications/marine) Access Date March 10, 2019.

[6] **Wilson, W., Hendrix, D., Gorski, J.,** "Hull Form Optimization for Early Stage Ship Design", Naval Engineers Journal, Vol. 122, pp. 53 – 65, 2010.

[7] **Percival, S., Hendrix, D., Noblesse, F.,** "Hydrodynamic Optimization of Ship Hull Forms", Applied Ocean Research, Vol. 23, pp. 337-355, 2001.

[8] **Abt, C., Bade, S. D., Birk, L., Harries, S.,** *Practical Design of Ships and Other Floating Structures*, Elsevier Science Ltd., 2001.

[9] **Day, A. H., Doctors, L. J.,** "Rapid Estimation of Near- and Far-Field Wave Wake from Ships and Application to Hull Form Design and Optimization", Journal of Ship Research, Vol. 45, pp. 73-84, 2001.

[10] **Tuck, E. O.,** "The Wave Resistance Formula of J. H. Michell (1898) and Its Significance to Recent Research in Ship Hydrodynamics", J. Austral. Math. Soc. Ser. B, Vol. 30, pp. 365-377, 1989.

[11] **Choi, J. E., Min, K. S., Kim, J. H., Lee, S. B., Seo, H. W.,** "Resistance and Propulsion Characteristics of Various Commercial Ships Based on CFD Results", Ocean Engineering, Vol. 37, pp. 549-566, 2010.

[12] **Cox, M. G.,** "The Numerical Evaluation of B-Splines", J. Inst. Mathematics and Its Applications, Vol. 10, pp. 134-149, 1972.

[13] **Sanches, F. M.,** *Parametric Modelling of Hull Form for Ship Optimization*, Master Thesis, Técnico Lisboa, 2016.

[14] **Kracht, A. M.,** "Design of Bulbous Bows", SNAME Transactions, Vol. 86, pp. 197-217, 1978.

[15] **Bole, M., Forrest, C.,** *Early Stage Integrated Parametric Ship Design,* ICCAS, 2005.

[16] **Michell, J. H.,** "The Wave-Resistance of a Ship", Philosophical Magazine, Vol. 45., pp. 106-123, 1898.

[17] **URL-2** (http://www.qednaval.co.uk/?page_id=942) Access Date February 16, 2019.

[18] **URL-3** (http://rhinocentre.blogspot.com/2009/12/grasshopper-parametric-ship-hull.html) Access Date September 24, 2018.

APPENDIX

EK A. Explanations of most frequently used Grasshopper commands.

Table A.1 : Explanations of most frequently used Grasshopper commands.

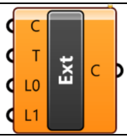
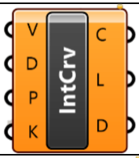
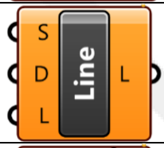
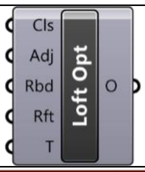
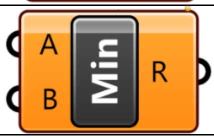
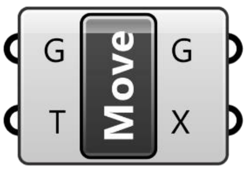
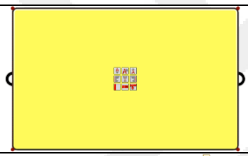
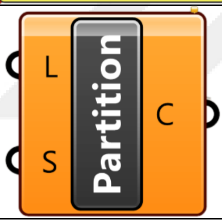
	Extend Curve	Extends curves by a specified distance. Curve, type of extension and extension 74ormal74 at the start and end of the curve are connected or set to corresponding inputs and final curve is obtained by Curve output.
	Extrude	Extrudes curves and surfaces along a specified vector. Geometry to extrude is connected or set to Base input and extrusion vector is connected to Direction input of command. Result is obtained by Extrusion output.
	Interpolate	Creates an interpolated curve through a set of points. Requires the interpolation points, curve degree, periodic being of curve and knot spacing style as input. Resulting curve, length and domain of the curve can be obtained by outputs of the command.
	Length	Measures the length of a curve. Curve or line is connected to curve input and length is obtained by output of the command.
	Line	Creates a line between two points. Points are connected to A and B inputs. Resulting line can be obtained by output of the command.
	Line SDL	Creates a Line segment by using starting point, tangent and length datas. Corresponding datas are connected to inputs of the command and resulting line is obtained by the output of the command.
	List Item	Retrieves a specific item from a list. Data list is connected to list input, item index is connected to index input and wrap choice is connected to wrap index input as true or false. Result can be obtained by the output of the command.
	List Length	Measures the length of a list. List is connected or set to input of the command and length of the list can be obtained by the output of the command.
	Loft	Creates a lofted surface through a set of section curves. Section curves are connected to Curves input and a loft options command is connected to options command to obtain the lofted surface by output of the command.
	Loft Options	Creates a loft options input for a Loft command. Closed loft, adjust seams, rebuild count, refit tolerance and loft type are required inputs to create the loft options.
	Maximum	Returns the greater of two items. Items are connected to A and B inputs, result is obtained by the output of the command.
	Minimum	Returns the lesser of two items. Items are connected to A and B inputs, result is obtained by the output of the command.
	Move	Moves an object or objects along a vector. Objects are connected to geometry input and translation vector is connected to Motion input. Translated geometry and transformation data are the outputs of the command.

Table A.1 (Cont.) : Explanations of most frequently used Grasshopper commands.

	Multiplication	Applies a mathematical multiplication. Input values are connected or set to A and B inputs where the result can be obtained by the result output of command.
	Negative	Computes the negative of a value. Value is connected to x input and the result is obtained by the y output.
	Number	Contains a collection of floating point numbers. Different numbers are set or connected to input of the command and they can be subjected to further manipulation together by the output of the command.
	Number Slider	Numeric slider for single values. Minimum and maximum values, slider accuracy and similar properties of the slider can be edited by right clicking on the command and proceeding to the edit tab.
	Panel	Monitors custom notes or text and number values. Useful for text inputs or monitoring or tracking the changes of a list of values.
	Partition List	Partitions a list into sub-lists. Main list is connected or set to List input and partition size is set or connected to size input. Resulting list which includes sub-lists can be obtained by the output of the command.

BACKGROUND

Name-Surname : Çağan Birant PEKKÜÇÜK

Date of Birth and Birth Place : 23.03.1990 / Balıkesir

E-mail : pekkucukc@itu.edu.tr

caganpekkucuk@gmail.com

EDUCATION :

- **Graduation** : 2015, Istanbul Technical University, Naval Architecture and Ocean Engineering Faculty, Naval Architecture and Marine Engineering
- **Post Graduate** : Istanbul Technical University, Department of Naval Architecture and Marine Engineering, Naval Architecture and Marine Engineering Program