

BIM-BASED MARINA DESIGN AND SEMI-AUTOMATED CODE
CHECKING PROCESS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY



BY
CANSU BALKU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

FEBRUARY 2021

Approval of the thesis:

**BIM-BASED MARINA DESIGN AND SEMI-AUTOMATED CODE
CHECKING PROCESS**

submitted by **CANSU BALKU** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ahmet Türer
Head of the Department, **Civil Engineering**

Assist. Prof. Dr. Aslı Akçamete Güngör
Supervisor, **Civil Engineering, METU**

Assist. Prof. Dr. Gülizar Özyurt Tarakcıoğlu
Co-Supervisor, **Civil Engineering, METU**

Examining Committee Members:

Prof. Dr. Rıfat Sönmez
Civil Engineering, METU

Assist. Prof. Dr. Aslı Akçamete Güngör
Civil Engineering, METU

Assist. Prof. Dr. Gülizar Özyurt Tarakcıoğlu
Civil Engineering, METU

Prof. Dr. Ahmet Cevdet Yalçiner
Civil Engineering, METU

Assist. Prof. Dr. Saman Aminbakhsh
Civil Engineering, Atılım University

Date: 12.02.2021



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Cansu Balku

Signature:

ABSTRACT

BIM-BASED MARINA DESIGN AND SEMI-AUTOMATED CODE CHECKING PROCESS

Balku, Cansu
Master of Science, Civil Engineering
Supervisor: Assist. Prof. Dr. Aslı Akçamete Güngör
Co-Supervisor: Assist. Prof. Dr. Gülizar Özyurt Tarakcıoğlu

February 2021, 176 pages

The design process of marina projects is time-consuming due to the scale and complexity of the projects. It also involves an iterative process for studying layout alternatives. There are international and national codes and guidelines that should be considered during design to provide safe navigation and berthing for boats, enable smooth operation, meet market and user expectations, and avoid design mistakes. However, checking the design against codes after each design revision takes considerable amount of time. Therefore, a limited number of alternatives can be explored and code checking is performed only after the design is complete. In construction projects, code compliance checking is mostly performed by designers or local authorities manually. Manual checking process is vulnerable to inconsistent results even when performed by experts.

This study aims to facilitate marina design and code checking by utilizing Building Information Modeling (BIM). BIM can be used for visualization, design review, documentation, and maintenance of marine infrastructure projects. However, in the literature and practice, BIM utilization for marina design is in its infancy. Therefore, in this study, the BIM model of a marina project was developed using Autocad Civil

3D and Revit software tools, by creating the required marina objects. The layout alternatives, by placement of finger piers, anchors and boats, were generated automatically. Moreover, with the help of Dynamo tool, the code checking process was executed semi-automatically. The proposed process has a potential to evaluate more marina layout alternatives in a shorter time than the traditional design review process. Thus, the design can be optimized, and the quality of the projects can be improved. Additionally, it is expected that this study can present the potential of using BIM technologies and concepts in coastal structure projects.

Keywords: Marina Design, Building Information Modeling, Automated Code-Checking, Marina Layout

ÖZ

YBM TEMELLİ MARINA TASARIMI VE YARI OTOMATİK KOD KONTROL SÜRECİ

Balku, Cansu
Yüksek Lisans, İnşaat Mühendisliği
Tez Yöneticisi: Dr. Öğr. Üyesi Aslı Akçamete Güngör
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi Gülizar Özyurt Tarakcıoğlu

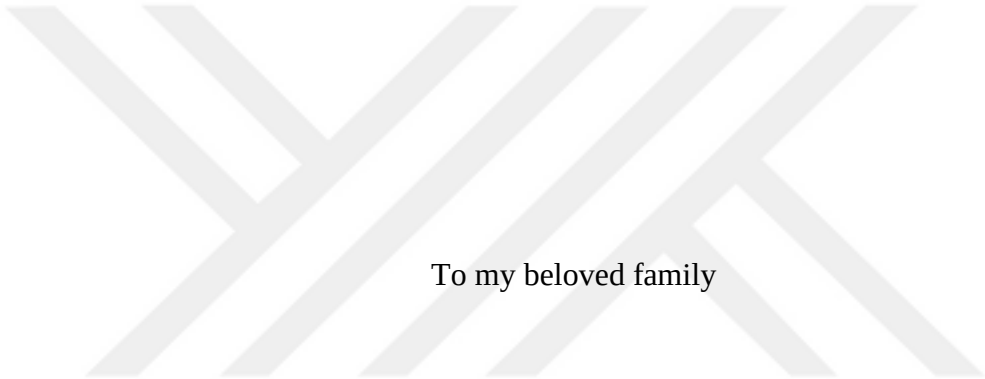
Şubat 2021, 176 sayfa

Marina projelerinin tasarım süreci, projelerin ölçeği ve karmaşıklığı nedeniyle zaman alıcıdır. Ayrıca, düzen alternatiflerini incelemek için yinelemeli bir süreci içerir. Tekneler için güvenli seyir ve yanaşma sağlamak, sorunsuz çalışmayı sağlamak, pazar ve kullanıcı beklentilerini karşılamak ve tasarım hatalarını önlemek için tasarım sırasında dikkate alınması gereken uluslararası ve ulusal kurallar ve yönergeler vardır. Bununla birlikte, her tasarım revizyonundan sonra tasarımı kodlara göre kontrol etmek önemli miktarda zaman alır. Bu nedenle, sınırlı sayıda alternatif araştırılabilir ve kod kontrolü ancak tasarım tamamlandıktan sonra gerçekleştirilir. İnşaat projelerinde, kod uygunluk kontrolü çoğunlukla tasarımcılar veya yerel yetkililer tarafından manuel olarak yapılır. Manuel kontrol süreci, uzmanlar tarafından yapıldığında bile tutarsız sonuçlara açıktır.

Bu çalışma, Yapı Bilgi Modellemesi (BIM) kullanarak marina tasarımını ve kod kontrolünü kolaylaştırmayı amaçlamaktadır. BIM, deniz altyapı projelerinin görselleştirilmesi, tasarımının incelenmesi, dokümantasyonu ve bakımı için kullanılabilir. Bununla birlikte, literatürde ve uygulamada, marina tasarımı için BIM kullanımı başlangıç aşamasındadır. Bu nedenle, bu çalışmada, Autocad Civil 3D ve Revit yazılım araçları kullanılarak gerekli marina nesneleri oluşturulup bir marina

projesinin BIM modeli geliştirilmiştir. Parmak iskele, çapa ve botların modele yerleştirilmesiyle yerleşim alternatifleri otomatik olarak oluşturulmuştur. Ayrıca Dynamo aracı yardımıyla kod kontrol işlemi yarı otomatik olarak gerçekleştirilmiştir. Önerilen süreç, geleneksel tasarım inceleme sürecinden daha kısa sürede daha fazla marina yerleşim alternatifini değerlendirme potansiyeline sahiptir. Böylelikle tasarım optimize edilebilir ve projelerin kalitesi iyileştirilebilir. Ek olarak, bu çalışmanın kıyı yapısı projelerinde BIM teknolojilerini ve kavramlarını kullanma potansiyelini ortaya koyması beklenmektedir.

Anahtar Kelimeler: Marina Tasarımı, Bina Bilgi Modellemesi, Otomatik Kod Kontrolü, Marina Yerleşimi



To my beloved family

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor Asst. Prof. Dr. Aslı Akçamete Güngör for her brilliant guidance and comments, criticism and encouragement throughout this study.

I would like to gratefully thank to Asst. Prof. Dr. Gülizar Özyurt Tarakcıoğlu for her vital guidance, support, and contributions for my research.

I would like to express my gratitude to Dr. Erhan Karaesmen and Dr. Engin Karaesmen for their moral supports and enjoyable and guiding conversations about academic, professional career, and life.

I wish to thank my coordinator Saniye Öktem and my team leader Orhan Ertuğral, and my colleagues Ergün Erdoğan and Tarık Büken for their technical help, understanding and supports about my thesis.

I would also like to thank my dearest friends Duygu Polat, Merve Çobanoğlu, Elife Çakır, Emrecañ Işık for their precious friendship, support and love.

I dedicated this thesis to my parents, Nuray Balku and Fatih Balku, for their continuous supports and loves. They always stand by my side for all my decisions without any question.

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS.....	xi
LIST OF TABLES.....	xiv
LIST OF FIGURES	xvii
LIST OF ABBREVIATIONS	xxii
CHAPTERS	
1 INTRODUCTION	1
2 LITERATURE REVIEW	7
2.1 MARINA DESIGN	7
2.2 CODE COMPLIANCE CHECKING FOR DESIGN	10
2.3 PREVIOUS APPLICATION OF BIM IN MARINE PROJECTS	15
3 RESEARCH METHODOLOGY.....	21
3.1 MARINA LAYOUT	21
3.2 DEVELOPING MARINA MODEL	28
3.2.1 MODELING BATHMETRY IN CIVIL 3D.....	29
3.2.2 CREATING BREAKWATER IN CIVIL 3D	37
3.2.3 MARINA MODEL IN REVIT	41
3.2.4 RULE CHECKS	63
3.2.5 PLACING FAMILIES ON MAIN PIERS	81
4 CASE STUDY	85
4.1 Finger Pier Arrangement	85

4.1.1	Slip (Berth) Width Distance	89
4.1.2	Inner Channel Width (B_1).....	92
4.1.3	Distance Between Main Piers.....	94
4.1.4	Water Level Check	96
4.1.5	Berthing Length (B)	100
4.2	Mediterranean (Anchor) Arrangement	102
4.2.1	Slip (Berth) Width Distance	106
4.2.2	Inner Channel Width (B_1).....	109
4.2.3	Distance Between Main Piers.....	110
4.2.4	Water Level Check	112
4.2.5	Berthing Length (B)	115
4.3	Combined Layout	117
4.3.1	Slip (Berth) Width Distance	122
4.3.2	Inner Channel Width (B_1).....	125
4.3.3	Distance Between Main Piers.....	126
4.3.4	Water Level Check	128
4.3.5	Berthing Length (B)	131
5	CONCLUSION AND FUTURE RECOMMENDATIONS	133
	REFERENCES	137
	APPENDICIES.....	141
A.	SLIP (BERTH) DISTANCE (W) DYNAMO SCRIPT AND PYTHON CODES	141
B.	INNER CHANNEL WIDTH (B_1) AND INNER CHANNEL WIDTH (COMBINED) DYNAMO SCRIPTS AND PYTHON CODES	146

C.	DISTANCE BETWEEN TWO MAIN PIERS (S) DYNAMO SCRIPT AND PYTHON CODES.....	151
D.	PYTHON CODES IN DISTANCE BETWEEN TWO MAIN PIERS (COMBINED) (S) DYNAMO SCRIPT	154
E.	WATER LEVEL CHECK DYNAMO SCRIPT AND PYTHON CODES 158	
F.	BERTHING LENGTH (B) DYNAMO SCRIPT AND PYTHON CODES 162	
G.	PLACING FAMILIES ON MAIN PIERS DYNAMO SCRIPT AND PYTHON CODES	165

LIST OF TABLES

Table 3.1 Revit Parameter Types	42
Table 3.2 Typical Design Parameters of The Boats	64
Table 3.3 Dimensional Characteristics of Super Yachts	65
Table 3.4 Minimum and Preferred Slip (Berth) Width Dimensions for Super Yachts	67
Table 3.5 Fender Diameter According to Boat LOA	68
Table 4.1 Initial Incorrect Design Data for Finger Pier Arrangement	86
Table 4.2 Marina Capacity According to Incorrect Design Data For Finger Pier Arrangement.....	86
Table 4.3 Pier Parameter Values for Finger Pier Arrangement.....	88
Table 4.4 Marina Capacity According To Intended Design For Finger Pier Arrangement.....	89
Table 4.5 Calculated Slip Distanced for Both Incorrect Design and Intended Design for Finger Pier Arrangement	90
Table 4.6 Slip (Berth) Distance Partial Results for Finger Pier Arrangement	91
Table 4.7 Inner Channel Width Results for Finger Pier Arrangement Layout	93
Table 4.8 Results of Distance Between Piers for Finger Pier Arrangement	95
Table 4.9 The Water Level Check Result For Finger Pier Arrangement Layout....	98
Table 4.10 The Part of The Lowest Point Results For Finger Pier Arrangement Layout.....	99
Table 4.11 Part of The Berthing Length Results for Finger Pier Arrangement	101
Table 4.12 Initial Incorrect Design Data for Mediterranean Arrangement	104
Table 4.13 Marina Capacity According To Incorrect Design Data For Mediterranean Arrangement.....	104
Table 4.14 Pier Parameter Values for Mediterranean Arrangement.....	105
Table 4.15 Marina Capacity According To Intended Design For Mediterranean Arrangement.....	106

Table 4.16 Calculated Slip Distanced For Both Incorrect Design and Intended Design for Mediterranean (Anchor) Arrangement.....	107
Table 4.17 Part of “Clearance Check between Berthing Place” Results For Mediterranean Arrangement.....	108
Table 4.18 Inner Channel Width Results for Mediterranean Arrangement Layout	110
Table 4.19 Results of Distance Between Piers for Mediterranean Arrangement Layout	112
Table 4.20 The Water Level Check Result for Mediterranean Arrangement.....	114
Table 4.21 The Part of The Lowest Point Results for Mediterranean Arrangement	115
Table 4.22 Part of The Berthing Length Results for Mediterranean Arrangement	117
Table 4.23 Incorrect Design Data for Combined Layout.....	119
Table 4.23 Incorrect Design Data for Combined Layout (continued)	120
Table 4.24 Marina Capacity According to Incorrect Design Data for Combined Layout	120
Table 4.25 Pier Parameter Values for Combined Layout	120
Table 4.25 Pier Parameter Values for Combined Layout (continued).....	121
Table 4.26 Marina Capacity According to Intended Design Data for Combined Layout	121
Table 4.26 Marina Capacity According to Intended Design Data for Combined Layout (continued).....	122
Table 4.27 Measured and Calculated Slip Distanced for Combined Layout.....	123
Table 4.27 Calculated Slip Distanced For Incorrect Design and Intended Design for Combined Layout (continued)	124
Table 4.28 Part of “Slip Distance” Results for Combined Arrangement Layout .	124
Table 4.28 Part of “Slip Distance” Results For Combined Arrangement Layout (continued)	125
Table 4.29 Results of Distance Between Piers in Combined Arrangement Layout	128

Table 4.30 Water Level Check Result for Combined Layout 130

Table 4.31 A Part of the Lowest Point Results for Combined Layout..... 131

Table 4.32 Part of the Berthing Length Results for Combined Arrangement..... 132



LIST OF FIGURES

Figure 3.1. Typical Marina Layout (Adapted from Coastal Structures Planning and Design Technical Manual by AYGM, 2016).....	22
Figure 3.2. Finger Pier Arrangement	22
Figure 3.3. Mediterranean (Anchor) Arrangement.....	23
Figure 3.4. Combine Marina Layout. (a) General (b) Detail	24
Figure 3.5. Rubble Mound Breakwater. (AWmaritime, 2017).....	25
Figure 3.6. The Mediterranean Mooring. (MacKenzie, 2016).....	27
Figure 3.7. Yachts in Yalıkavak Marina, Turkey. ([Megayachts], 2020).....	27
Figure 3.8. Zone Settings in Autodesk Civil 3D.....	30
Figure 3.9 Settings of New Point File Format	31
Figure 3.10. Importing Point Data in Civil 3D	32
Figure 3.11. Points of Izmir Bay Area Region in Civil 3D	33
Figure 3.12. Creating Surface names as “Gediz”.....	34
Figure 3.13. Bathymetry Model in Civil 3D.....	35
Figure 3.14. Creating a Boundary For Gediz Area	36
Figure 3.15. Export Preferences in Civil 3D.....	36
Figure 3.16. Bathymetry Model in Revit	37
Figure 3.17. The Cross-Section of The Breakwater.....	38
Figure 3.18. Create Alignment Properties	38
Figure 3.19. Surface Profile Through Breakwater Alignment.....	39
Figure 3.20. The Solid Created From The Corridor	39
Figure 3.21. Breakwater Model in Civil 3D	40
Figure 3.22. Breakwater Model in Revit	40
Figure 3.23. Parameter Properties.....	44
Figure 3.24. “Common Family Parameter” and Its Values for A Family in The Model	46
Figure 3.25. Representation of Clearance Between Main Piers	47
Figure 3.26. Maximum Boat Width.....	48
Figure 3.27. Maximum Boat Length.....	49

Figure 3.28. Maximum Boat Draft	49
Figure 3.29. Properties of “Main Pier” Family	51
Figure 3.30. Dimension Measurement of “Main Pier” Family. (a) Depth Measurement (b) Width and Length Measurement	52
Figure 3.31. Parameters of “Main Pier” Family	53
Figure 3.32. Properties of “Finger Pier” Family	54
Figure 3.33. Dimensions of “Finger Pier” Family (a) Length and Width Measurements (b) Height Measurement	55
Figure 3.34. Parameters of “Finger Pier” Family	57
Figure 3.35. Parameters of “Anchor” Family	58
Figure 3.36. Berthing Length Measurement for “Anchor” Family	59
Figure 3.37. Properties of “Boat Family	60
Figure 3.38. Parameters of “Boat” Family	61
Figure 3.39. Dimensions of “Boat” Family. (a) Length and Width Dimensions (b) Draft and Height Dimensions	63
Figure 3.40. Slip (Berth) Width Dimension Representation in a Layout. The upper Part (Main Pier 2A) is Finger Pier Arrangement. The lower Part (Main Pier 3A) is Mediterranean Arrangement.	66
Figure 3.41. Slip (Berth) Dimension Dynamo Script Logic	68
Figure 3.42. Inner Channel Representation	70
Figure 3.43. Inner Channel Width (B1) Dynamo Script Logic	71
Figure 3.44. Inner Channels For Combined Layout	72
Figure 3.45. “Inner Channel Width (Combined)” Dynamo Script Logic for Inner Channel-2 Case	73
Figure 3.46. Representation of Clearance Between Main Piers	74
Figure 3.47. Distance between Main Pier Dynamo Script Working Logic	75
Figure 3.48. Main Pier Placement Layout	76
Figure 3.49. Combined Layout Plan	77
Figure 3.50. Water Level Check Dynamo Script Working Logic	79
Figure 3.51. “Berthing Length (B)” Dynamo Script Working Logic	80

Figure 3.52. The Working Logic of “Placing Families on Main Piers”	81
Figure 4.1. Finger Pier Arrangement Layout. (a) Plan (b) 3D.....	87
Figure 4.2. Slip (Berth) Distance Rule Checking Result	90
Figure 4.3. Inner Channel Width Rule Checking Result for Finger Pier Arrangement	93
Figure 4.4. Main Pier Distance That is Controlled for Finger Pier Arrangement Layout	94
Figure 4.5. Distance Between Main Piers Rule Checking Result for Finger Pier Arrangement	95
Figure 4.6. Finger Pier Arrangement Plan for Water Level Check	97
Figure 4.7. Water Level Check Result For Finger Pier Arrangement	98
Figure 4.8. Berthing Length Measurement For Finger Pier Arrangement Layout	100
Figure 4.9. Berthing Length Rule Checking Results For Finger Pier Arrangement	101
Figure 4.10. Mediterranean Arrangement. (a) Plan Layout (b) 3D View.....	103
Figure 4.11. Berthing Length Measurement for Mediterranean Arrangement.....	105
Figure 4.12. Slip (Berth) Distance Rule Checking Result for Mediterranean (Anchor) Arrangement.....	107
Figure 4.13. Inner Channel Width Rule Checking Result for Mediterranean Arrangement	109
Figure 4.14. Measurement That is Used in The Control for Mediterranean Arrangement	111
Figure 4.15. Distance Between Main Piers Rule Checking Result for Mediterranean Arrangement	111
Figure 4.16. Mediterranean Arrangement Plan for Water Level Check.....	113
Figure 4.17. Water Level Check Result For Mediterranean Arrangement.....	114
Figure 4.18. Berthing Length Measurement For Mediterranean Arrangement.....	116
Figure 4.19. Berthing Length Rule Checking Results for Mediterranean Arrangement	116

Figure 4.20. Combined Layout. (a) Pier A and Pier B Layout (c) General Layout	118
Figure 4.21. Slip (Berth) Distance Rule Checking Result for Combined Layout	123
Figure 4.22. Inner Channel-2 Code Check Result for Combined Layout	126
Figure 4.23. Distance Between Main Piers Rule Checking Result for Combined Layout.....	127
Figure 4.24. Combine Layout Plan for Water Level Check.....	129
Figure 4.25. Water Level Check Result For Combined Layout.....	130
Figure 4.26. Berthing Length Rule Checking Results for Combined Layout	132
Figure A.1. Slip (Berth) Distance (W) Dynamo Script	141
Figure A.2. “Filtering” Code in “Slip (Berth) Distance (W)” Dynamo	143
Figure A.3. “Measuring Distance” Python Code in “Slip (Berth) Distance (W)” Dynamo Script.....	143
Figure A.4. “Controlling” Code in “Slip (Berth) Distance (W)” Dynamo	144
Figure C.1. Distance Between Two Main Piers (S) Dynamo Script	151
Figure C.2. “Selection By Name” Python Code in “Distance Between Two Main Piers (S)” Dynamo Script	152
Figure C.3. “Measuring Distance” Python Code in “Distance Between Two Main Piers (S)” Dynamo Script	153
Figure C.4. “Controlling” Python Code in “Distance Between Two Main Piers (S)” Dynamo Script.....	153
Figure D.1. “Selection by Name” Python Code in “Distance Between Two Main Piers (Combined) (S)” Dynamo Script.....	154
Figure D.2. “Measuring Distance” Python Code in “Distance Between Two Main Piers (Combined) (S)” Dynamo Script.....	155
Figure D.3. “Controlling” Python Code in “Distance Between Two Main Piers (Combined) (S)” Dynamo Script.....	156
Figure E.1. Water Level Check Dynamo Script.....	158
Figure E.3. “Filtering” Python Code in “Water Level Check” Dynamo Script	160
Figure E.4. “Controlling” Python Code in “Water Level Check” Dynamo Script	161

Figure F.1. Berthing Length (B) Dynamo Script	162
Figure F.1. “Checking Berthing Length” Python Code in “Berthing Length (B)” Dynamo Script	163
Figure G.1. Element Selection, Berthing Length Calculation and Obtainin Main Piers’ Surfaces Steps of "Placing Families on Main Piers" Dynamo Script.....	165
Figure G.2. Placing Elements on Main Piers’ Left and Right Faces Steps of "Placing Families on Main Piers" Dynamo Script.....	166
Figure G.3. Placing Boat and Setting Parameters to Placed Elements Steps in "Placing Families on Main Piers" Dynamo Script.....	167
Figure G.4. Reporting Results in "Placing Families on Main Piers" Dynamo Script	168
Figure G.5. “Filtering” Python Code in “Placing Families on Main Piers” Dynamo Script	168
Figure G.6. “Setting Parameters” Python Code in “Placing Families on Main Piers” Dynamo Script	171

LIST OF ABBREVIATIONS

Abbreviation	Definition
AEC	Architecture, Engineering and Construction
ASCE	American Society of Civil Engineers
ARC	Automated Rule Checking
AYGM	T.C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı Altyapı Yatırımları Genel Müdürlüğü
B	Boat Width
B_1	Inner Channel Width
B_1	Finger Pier Berth Length
B_2	Anchoring Berth Length
BIM	Building Information Modelling
BL	Berthing Length
B_{max}	The width of the longest boat berthing to the pier
BS	British Standards
C&O	Construction and Operability
CAD	Computer-Aided Design
CERI	Civil Engineering Research in Ireland
CIM	Civil Information Modelling
CIRIA & CUR	Construction Industry Research and Information Association
CNVR 2016	Proceedings of the 16th International Conference on Construction Applications of Virtual Reality
CORENET	Construction and Real Estate NETwork
CRC for CR	The Australian Cooperative Research Centre for Construction Innovation
D	Boat Draft
f_d	fender diameter for specified superyacht length
GEBCO	The General Bathymetric Chart of the Ocean
GPS	Global Positioning System

GSA	General Service Administration
IAHS	International Association for Housing Science
ICC	International Code Council
ICE	Institution of Civil Engineering
ICRIIS	International Conference on Research and Innovation in Information Systems
IFC	Industry Foundation Classes
IHO	International Hydrographic Organization
IMO	International Maritime Organization
L	Boat Length
LOA	Length Overall
MATEC	Materials Science, Engineering and Chemistry
mc	maneuverability coefficient
MTCE	International Maritime-Port Technology and Development Conference
OCDI	The Overseas Coastal Area Development Institute of Japan
OCIMF	The Oil Companies International Marine Forum
PIANC	Permanent International Association of Navigation Congresses
ROM	Spanish Standards
S	Distance Between Piers
UK	United Kingdom
US	United States
USA	United States of America
UTM	The Universal Transverse Mercator
W	Slip (Berth) Width
WGS	The World Geodetic System
YBM	Yapı Bilgi Modellemesi

CHAPTER 1

INTRODUCTION

Yachting is a recreational activity that has continued for a very long time. It also has tourism value that should be taken into consideration for countries that have coastlines. For this reason, marinas play a vital role in yacht tourism. Klancnik and Philips (2010) defined marinas as “basins in a body of water that provide protection from the elements (waves, wind, tides, ice, currents, etc.) for a variety of commercial and recreational watercraft ranging in size and type.”. According to Guidelines for Marina Design (Report No 149 - Part1) (PIANC - Recreational Navigation Commission, 2016), “Marinas provide infrastructure for recreational navigation, including tourism and sports, also called pleasure navigation.”. Marinas have various facilities and services for not only vessels and their users but also the non-boater community in the region.

Marinas are not just park spaces for yachts or other vessels. In the design process, related design parameters should be taken into consideration to achieve maximum benefit, functional use, and safe berthing space. Besides, safety in a marina is an important factor in the preference of the users. Bilski (2015) indicated that the users look out for a marina's functionality and utility besides its aesthetic architectural design. The marina should provide a safe area for maneuvering and mooring.

Furthermore, in Docklands Waterways Vessel Traffic Study And Related Strategic Documents (City of Melbourne & VicUrban, 2011), it is indicated that the entrance of the marina is vital for water quality. It plays a vital role in the water circulation in the marina basin. Improper design of the entrance may cause the accumulation of organic material, pollutants, and sediments. Besides that, a proper marina layout configuration may also support flushing in the basin. Also, the boat distribution should be taken into consideration carefully while designing a marina. If the berthing

structures' configuration and dimensions are designed for mostly smaller boats, there may be problems in maneuver safety for larger boats. On the contrary, if the larger boats are considered mostly in design, a potentially hazardous situation such as collision may occur for small recreational boats.

Additionally, the recent developments in marine engineering and naval architecture increase the number of recreational boats and their sizes. Therefore, the demand for safe berthing facilities is increased. As a result, public or private investors start to consider new berthing facility developments or improving conditions of existing facilities in order to meet that increasing and changing demand. However, new marina developments are expensive projects. Therefore, investors prefer to improve the conditions of existing marinas. The existing facilities are evaluated in order to identify needs and required improvements to meet the new demand.

The results of the moorage assessment in Port of Grays Harbor Westport Marina Demand Analysis Draft Report (BTS Associates, 2020) during the redevelopment of Westport Marina showed that larger boats have to be berthed to smaller float due to an increase in boat length. This operation affects the life-expectation of the floats and causes navigational problems for boats. The current situation in the Westport Marina is explained such that the vessels' forces on the berthing structures are increasing significantly as the length of vessel increases. Besides that, the geometry of mooring changes according to the size of the vessel. The moorage geometry may result in extra twisting and torsional forces on the structures. The concrete has less endurance under tensile forces compared to under compression forces. As a result, the damages like cracks occur that leads water into the float's inner layers. The water causes more damage over time (BTS Associates, 2020). This report shows that marina design should not only consider the present factors but also reflect the possible changes in the user demands and technological improvements while ensuring the safety at all times.

The design process of the marina layout starts with the development of layout alternatives in accordance with site specific conditions. Then, these alternatives are

evaluated according to project goals and specific technical requirements such as compliance to the guidelines. Based on the results of the evaluation, the alternatives are updated, and the evaluation starts again. The design is an iterative process for all kinds of construction projects. In marina projects, besides the initial design process, the marina design can be modified according to changes in physical and user properties during the operation. Customer profile or the boat distribution modify over time because of trends in the boating community and technological developments in marine engineering. Also, the seasonal change can result in need to alter the marina layout design. Therefore, marinas require a dynamic approach to the design and management.

There are not any strict rules to follow during marina design. However, there are several international guidelines that should be considered during design to provide safe navigation and berthing for boats, enable smooth operation, meet market demands and user expectations. Besides the international design guidelines, Ministry of Transport, Maritime Affairs and Communications of the Republic of Turkey, General Directorate of Infrastructure Investments (T.C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı, Altyapı Yatırımları Genel Müdürlüğü) had introduced a national guideline which should be followed during marina design. However, poor implementation of design guidelines causes some structural and safety problems at the marina's operational phase.

According to Gucma et al. (2018), the guidelines are often not effectively implemented. Two hundred ten marina layouts are analyzed by Gucma et al. (2018) in their study “Statistical Analysis of Parameters of Selected Worldwide Yacht Ports and Marinas in Terms of Design Guidelines.” The study results show that the width of the entrance is not designed according to the parameters in the guidelines and is even smaller than the minimum distance allowed, in most of the marina. Moreover, the lengths of the berthing line in marinas are shorter than the minimum requirement. In the end, Gucma et al. (2018) pointed out that the design guidelines are not applied in the designs of most of the marinas properly and effectively, which results in poor navigation safety and yacht services. Therefore, code compliance checking of marina

projects is an important step to ensure safety and operability at all times due to the dynamic aspect of marina design process.

In construction projects, code compliance checking is mostly performed by designers or local authorities manually. Manual rule checking process is vulnerable to inconsistent results even though the person, who performs the checking, is an expert (Fiatach, 2012 as cited in Warren, 2019). Therefore, it is stated as an inefficient process (Warren, 2019). Also, the projects, like marina projects, can be classified as complex projects because they are large projects in scale and have many variables and elements. According to Ismail et al. (2017), as the project's complexity increases, the complexity and the number of building codes that should be implemented in design will be increased. This statement can be applicable for marina projects. As mentioned before, marina projects have many elements like piers, finger pier, breakwater, anchoring to enable safe berthing and mooring for different sizes of boats, and some utilities and structures to provide electricity, fuel, maintenance and repair services. Also, marinas contain recreational facilities for recreation, accommodation, shopping and entertainment. The physical and spatial relationships of all these elements and facilities are indicated in the marina design guidelines. Even though this study's main concern is only the marina layout that contains the boats and the berthing structures that form the marina layout, controlling all these requirements with a lot of different elements can be time-consuming and error-prone.

On the other hand, the code compliance checking should be performed after every change in design. Because of that, it requires extra time and labor, which increases costs. Moreover, these iterative works are considered as a burden by the project parties. They prefer to finish reviewing, finalize the design, and continue to the next step of the project. As a result, there are not enough resources or willingness to explore and evaluate all possible design options. Code compliance checks are performed on only one design option as the design evolves. That is considered as a drawback for design improvement and quality (Sakikhales & Stravoravdis, 2015).

Moreover, if the designer or controller cannot identify the problems regarding code compliance due to iterative and heavy work of the design process, all design problems are transferred to the construction phase. This situation causes spending of more resources like money, labor, and time (Lee et al., 2015). Besides that, violation of any kind of design codes negatively affects the quality of design (El-Diraby, 2019). Ismail et al. (2017) also mentioned that some changes in design could be made unintentionally during the evaluation of the design by other parties. The designer or controller may not be aware of these changes, which results in code violations. El-Diraby (2019) also stated that senior members of the team usually perform the code compliance checking processes to minimize errors and time allocated. However, their salary is higher than a junior member, causing an increase in cost overall in the project budget.

Considering the importance of code compliance checking in marina projects as well as the limitations of the implementation in the highly iterative process of marina design, this study aims to facilitate marina design and code checking by utilizing Building Information Modeling (BIM). BIM can be used for visualization, design review, documentation, and maintenance of marine infrastructure projects (Cheng et al., 2016 and Costin et al., 2018). However, in the literature and practice, BIM utilization for marina design is in its infancy. Therefore, in this study, the BIM model of a marina project was developed using Autodesk Civil 3D and Revit software tools, by creating the required marina objects. The layout alternatives, by placement of finger piers, anchors and boats, are generated automatically. Moreover, with the help of Dynamo tool, the code checking process was executed semi-automatically. Thus, it is investigated whether, by using this process, the configuration of a marina can be controlled for compliance with the national design guidelines and for suitability for the target boat users. In this way, the semi-automated code checking process can be used to evaluate different marina layout configurations in a relatively shorter time than the traditional design review process and in a more accurate way. Also, placing finger piers, anchors, and boats in the model automatically can enable generating more marina layout alternatives to optimize the design and better respond to the

needs of the customers. Hence, it is aimed to increase efficiency and quality in the design process. Additionally, it is expected that this study can present the potential of using BIM technologies and concepts in coastal structure projects.

In Chapter 2, marinas and marina design are briefly explained. General information about Turkish guideline which should be followed during marina design is given. Also, a literature review regarding code compliance checking for design is presented. The possible advantages of code compliance checking are discussed through previous studies. In the last part, the current situation of BIM use in marine projects is given.

Chapter 3 provides information about developed marina models, the Revit families and the parameter used for modelling marinas and checking code compliance. Also, the design rules which are investigated in the scope of this study are explained. The dynamo scripts used in the developed semi-automated code checking process are given in detail. Additionally, the process of semi-automated marina modelling and marina capacity analysis are described.

In Chapter 4, three different marina layout cases are created semi-automatically via Dynamo scripts explained in Chapter 3. These layout cases are (1) Finger Pier Arrangement, (2) Mediterranean (Anchor) Arrangement, and (3) Combined Layout. The semi-automated code compliance is applied to these three cases, and the results are presented for each case.

Finally, Chapter 5 summarizes the study and discusses the results and limitations. Moreover, recommendations for the continuation of the study are presented.

CHAPTER 2

LITERATURE REVIEW

In this chapter, general information about marinas and marina design are given. Also, Turkish guidelines about marina design are mentioned. On the other hand, code compliance checking is introduced briefly. Its possible impacts such as improving quality, optimization on design procedure and previous studies are discussed. Lastly, previous applications of BIM in marine projects are mentioned.

2.1 MARINA DESIGN

Permanent International Association of Navigation Congresses - PIANC (2016) mentioned that the origin of “Marina” word is Latin. It refers to female, and means “from the sea.”. “Marina” term is firstly used for a recreational boating facility by the National Association of Engine and Boat Manufacturers. The terms “small craft harbor” or “yacht marina” are also used for marinas.

Nowadays, the importance of marinas for both the boating and the regional community are continuously increasing. Klancnik and Philips (2010) indicated that three parties affect initial project goals. The first one is the boating community demanding safer, more convenient facilities. The second one is the general public requesting better visual and physical access to the water. The last party is the real estate developer requiring an efficient and profitable marina operation. Besides these demands from the mentioned parties, a marina project has to be shaped according to market and site conditions, regulations, and financial limitations.

Site surveys, water area conditions, basin flushing and silting are some main subjects that are considered during marina design and planning. According to the site conditions and user and customer needs, layouts, shoreline structures and basin

protection alternatives, fuelling and other boat utilities, land area services, and facilities, boatyards, and dry stacks store systems are design and planned.

The first steps of planning a small craft harbor project consist of determining the goals that meet the parties' requirements mentioned above and preparing a preliminary program for the facility's design, construction, and operation. A market demand analysis that includes the requirements and expectations of the owner and the community is needed for the scope of the development. Whether there are other marinas in the region and their capacities or the boat types that they offer services for should be considered during the market demand analysis. These are the important factors to determine recreational and commercial facilities. Moreover, the boat demographics should be determined to decide the marina capacity and the size of the development.

The physical planning takes form according to the demands of boaters and non-boaters. At the end of the physical planning, a concept plan, phasing recommendations, associated costs, and financial projections for harbor development are obtained. All of these forms an implementation strategy.

Marina Layout Development is stated by PIANC (2016) as the physical planning of marinas' land and water area. Layout plan alternatives are designed according to collected data and client's requirements. Therefore, geotechnical investigation, initial wave modeling, etc. are important. It is also advised in PIANC (2016) that market targets, environmental considerations, added value to upland areas, public access to the waterfront, etc. should be considered during the layout development phase.

Generated layout options for both land and water area will be used for foreseen dredging, excavation and/or reclamation activities, and wave protection structures. Therefore, they should comply with geometrical guidelines, which will be mentioned in the following chapters.

The layout alternatives are evaluated according to some concerns which are given in PIANC (2016) as costing, programming, timing, phased development, expansion possibilities, environmental impact issues, interfaces with existing surrounding and with other physical plans, traffic circulation, and hinterland connection/access to the marina from waterside and landside, etc. Different structural solutions are also considered during the evaluation of alternatives. This analysis helps identifying some geometrical and physical values of structures that will positively affect the design change orders. In the end, more accurate cost estimations and commercial feasibility can be obtained.

All the steps and the requirements for developing a marina project are specified in several international and national guidelines. ASCE (2012) and PIANC reports are the international guidelines that are commonly used. On the other hand, Turkey had developed its own national guidelines that guides designing and implementing any kind of coastal structure.

“Kıyı Yapıları Planlama ve Tasarım Teknik Esasları” is translated as “Coastal Structures Planning and Design Technical Manual”. This manual is published in 2016 under the leadership of the respective Turkish academicians who specialized in coastal engineering and with the support of Ministry of Transport, Maritime Affairs and Communications of the Republic of Turkey, General Directorate of Infrastructure Investments (T.C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı, Altyapı Yatırımları Genel Müdürlüğü - AYGM). Some private companies and foreign academicians also contributed to the preparation process. It provides information regarding the performance-based design of coastal and marine structures. The manual is based on the international guidelines such as Permanent International Association of Navigation Congresses (PIANC), American Society of Civil Engineers (ASCE), British Standards (BS), The Overseas Coastal Area Development Institute of Japan (OCDI), Spanish Standards (ROM), International Hydrographic Organization (IHO), International Maritime Organization(IMO), Institution of Civil Engineering (ICE), Eurocodes, The Oil Companies International Marine Forum (OCIMF), and Construction Industry Research and Information

Association (CIRIA & CUR). The related articles in some journals like Coastal Engineering Journal, Journal of Waterway, Coastal, Harbour, and Ocean Engineering are also considered. It is advised that the principles in the manual should be taken into consideration during the physical and numerical experiments for coastal and structural design. This manual has detailed information about performance-based design of coastal structures such as breakwater, berthing structures, coastal protection structure. It also contains hydraulic design principles about wind waves, long waves, water level changes, current, sediment transport and morphology. Moreover, planning and design fundamentals of ports, yacht marinas, fishing ports, cruise harbors, shipyard. The concern of this study is the planning and design fundamentals of yacht marinas. The information about marinas and their design principles are given in the Methodology Chapter.

2.2 BIM-BASED CODE COMPLIANCE CHECKING FOR DESIGN

In this section, the definition of BIM and the features and the capabilities of a BIM model is explained briefly. Then, the advantages of the BIM model in the design stages are mentioned shortly. Later, the limitations of the manual code compliance checking in the design stage are discussed and the automated code compliance checking concept is introduced. Also, the role of BIM in the automated code compliance process is mentioned. Lastly, the advantages of automated code compliance checking for design is presented.

2.2.1 BIM-BASED DESIGN

U.S. National Building Information Model Standard Project Committee (n.d.) defines Building Information Modeling (BIM) as “a digital representation of physical and functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility forming a reliable basis for decisions during its lifecycle from inception onward.”. BIM is specified as “process” that “generate

and maintain building information in an interoperable and reusable way” by Lee et al. (2006). They stated that “A BIM system is a system or a set of systems that “enables” users to integrate and reuse building information and domain knowledge through the lifecycle of a building.” (Lee et al., 2006).

The current technology used for utilizing BIM is based on object-based parametric modeling. The history of object-based parametric modeling dates back to 1980s. The most important feature of the object-based parametric modeling is that the geometries and non-geometric properties of the objects are defined by set of parameters and rules. Also, it has the ability to modify the geometries or the properties of the objects by updating the parameters and rules. These parameters and the rules can be generated and defined by the designer of the software. Moreover, the user can customize the objects according to the needs of the projects by updating the pre-existing parameters and rules or generating new ones. The object attributes defined by the parameters and the rules can be used for analysis, cost estimation, and code compliance checking, etc. (Eastman et al., 2008)

Eastman et al. (2008) specified that a parametric BIM object has: (i) geometric definitions, data, and rules, (ii) non-redundant geometry that prevents inconsistencies, (iii) parametric rules that modify the geometry, (iv) object definition at a different level of aggregation and at any level of hierarchy levels, (v) object rules that identify the violation of object feasibility, (vi) ability to link, receive, export, broadcast of attributes of the objects.

BIM models or object-based parametric models provide numerous advantages in the design stage as through the lifecycle of a construction project. The most known and recognizable feature of the BIM model is visualization. It enables to present a visual form of the design that is consistent at every view. As mentioned above, the objects of a BIM model can be updated or modified via parameters that results in reducing effort for every design change. The ability of automatic and consistent generation of 2D drawings helps to reduce time and error for that task. BIM technology improve the collaboration in project disciplines even in the early stages of the design.

Therefore, design problems are identified at earlier stages of the project which provides opportunity to optimize and improve the design. Detailed and accurate cost estimations can be obtained from a BIM model. This feature helps the project parties to recognize how a design change affects the cost of the project. Lastly, the energy and sustainability analysis can be performed and the results of them can be used to improve the performance of the structure (Eastman et al., 2008).

2.2.2 AUTOMATED CODE COMPLIANCE CHECKING FOR DESIGN

A proposed design for any kind of project should be controlled according to the requirements or the relevant standards. There are variety of regulation, codes and guidelines that should be applied for the structures' safety, stability, accessibility, energy efficiency, etc. throughout the lifecycle of the structure. The designers, the engineers or the authorities in the construction sector control the structure at every step of the project to make sure that it conforms to all necessary regulations, codes and guidelines. Even in the early stages of the design, the controlling process takes a considerable time. Preidel and Borrmann (2018) stated that the code compliance checking is iterative and generally manual process. Therefore, it requires high volume of effort, cost and time so it is open to errors. Manually checking the design is a burden for a designer or a person that supervise the design. Also, the process is dependent mainly on the qualification of the designer or the controller. Lee et al. (2015) stated that the designer might not be aware that the design is incompatible with the standards due to the heavy work of code checking and possible lack of experience. Additionally, Lee et al. (2016) performed a study and a series of experiments about the cognitive challenges of drawing-based review. The results reported that "... experienced professionals could detect less than 3% of known errors on drawings during the experiments. This poor detection rate occurs because traditional drawing-based building design review imposes a heavy cognitive load on

engineers, detailers, and project coordinators.”. The result of this situation costs more money and time in the later stages of the project.

Moreover, the design process is iterative. At every iteration, the involving parties may change the design unintentionally during controlling the iterations or solving the design problems. Whether it is intentionally changed or not, the design should be controlled according to the specific requirements for the structure. On top of that, if the project contains a complex structure with numerous design requirements, the checking process will consume more time and resources.

The developments in the digital world brought the capability of Building Information Modeling (BIM) that enables parametric design and automated code compliance. Akbas (2019) stated that BIM models enable automated code checking via their components that have semantically rich information. The BIM model should have the necessary information to support the automated code checking process which is a process using functions and calculations and requires data for these functions and calculations (Nawari, 2013 & Akbas, 2019). Eastman et al. (2008), as cited in Lee et al. (2015), state that automated rule checking is one of the vital benefits of BIM for project participants during controlling and validating the design iterations according to the design requirements

Automated code compliance is defined by Eastman et al. (2009) as “...software that does not modify a building design, but rather assesses a design on the basis of the configuration of objects, their relations or attributes. Rule-based systems apply rules, constraints or conditions to a proposed design, with results such as “pass”, “fail” or “warning”, or ‘unknown’ for cases where the needed data is incomplete or missing.”. Different terms are used for the process such as automated code compliance, automated rule checking, or automated code checking. Eastman et al. (2009) identified the stages of the automated code compliance to implement it properly and obtain consistent results: (1) Rule interpretation, where the codes that are written in human language are transformed into a form that a computer can recognize and implement, (2) Model preparation, where the model of the structure that contains

information used for the code checking is prepared, (3) Rule execution, where the process is performed, (4) Reporting, where the results of the checking are obtained.

Eastman et al. (2009) reviewed existing rule checking systems and their capabilities. They used Industry Foundation Classes (IFC) models to evaluate the five major industrial efforts: CORENET (Singapore), Statsbygg (Norwegian), CRC for CI (Australia), ICC (USA), GSA (USA). Also, rule checking systems are mentioned as design development systems through various platforms such as an add-in to design tool, a desktop application, or a web-based application. All options enable the designer to check the design at any stage during the development of the design. It is stated that every design change can be tracked and checked whether the change violates any design rule with the help of these systems.

Automated code compliance is emerged due to the need to eliminate errors and increase quality. Lee et al. (2015) indicate that automated code checking helps decreasing design and human errors in the long term. At the same time, it can improve the design process and the quality of the data exchange.

.

Martins et al. (2016) performed research regarding the advantages and limitations of using automated rule-checking procedures for the design process. It is stated that automated code checking procedures can help the designer develop design options and support design decisions. It is specified as an information procedure that can be used to check whether their design is applicable to the design standards in the early design stages.

El-Diraby (2019) investigated the future of automated rule checking (ARC) on behalf of buildingSMART International Regulatory Room. It is indicated that automated rule checking is not just a tool for design compliance but also provides a capability for generative design. In other words, the design rules or standards can be used to obtain an optimum solution. Moreover, it helps to explore more design options by multiple iterations in a small amount of time. El-Diraby (2019) also

mentioned that the procedure is used for layout optimization and other analyses such as programmatic spaces, cost analysis, consumption, and preliminary energy building circulation. It is stated that as automated rule checking is used repetitively, it can produce enough data to enable the implementation of machine learning.

2.3 PREVIOUS APPLICATION OF BIM IN MARINE PROJECTS

Costin et al. (2018) stated that BIM has the potential to improve the quality in infrastructure projects over its lifecycle. It is mentioned that it can ensure more productive, safer, and sustainable management solutions while decreasing risks and costs. According to SmartMarket Report (Dodge Data & Analytics, 2017), BIM users in the infrastructure industry in the US, UK, France, and Germany believe that BIM helped to reduce error and increase cost predictability accuracy. Also, it is indicated that the businesses that use BIM standardized their project delivery systems and increased customer satisfaction. Dodge Data & Analytics (2017) reported the result of the research about BIM use in infrastructure projects as “...grew from 20% in 2015 to 52% in 2017”. It is stated that “...engineers and contractors...expect to be (2019) using BIM on 50% or more of the transportation infrastructure projects” (Dodge Data & Analytics, 2017). However, this tendency is not observed clearly in marine projects. BIM use in marine projects is still limited (Costin et al., 2018).

Cheng et al. (2016) investigated the adaptation of BIM in civil infrastructure facilities such as transportation, energy, utility and recreational facilities. They used “Civil Information Modeling (CIM)” term for these kinds of projects. They categorized the projects into nine groups as (i) Bridges, (ii) Roads, (iii) Railways, (iv) Tunnels, (v) Airports, ports and harbors, (vi) Energy infrastructure, (vii) Utility infrastructure, (viii) Recreational facility infrastructure, and (ix) Water management infrastructure. 171 case studies and 62 academic papers about these categories are evaluated. They concluded that the level of CIM implementation in civil infrastructure projects except for bridges, roads, and tunnel projects is low. Airport,

harbors, and port projects are categorized together. The conclusion and the suggestion for this category are that “there are few CIM uses conducted based on LoD300 and LoD400 models. More effort should be put into CIM technology for this civil infrastructure category.” (Cheng et al., 2016).

Costin et al. (2018) performed a literature review and critical analysis of BIM for transportation infrastructure. They reviewed schemas and file formats from 9 categories and 34 areas related to transportation infrastructure and 189 publications. It is stated that the study focused on roads, highways and bridges mainly, so that the result of the study showed an increase in BIM use in transportation projects. However, the results for port and harbor projects are not promising as other transportation projects. It is indicated that BIM is not preferred in the design, planning and the management stages of port and harbor projects. They suggested using BIM in these projects for effective maintenance plans, monitoring the structures' conditions, and preparing repair alternatives in the management stage. Furthermore, it is mentioned that BIM can contribute planning of ports and harbors that results in more efficient operation (Costin et al., 2018).

Collery et al. (2016) presented the experience of BIM use in the Clacton and Holland-on-Sea Coastal Protection project. This project is mentioned as “one of the first coastal protection infrastructure projects in the UK”. The scope of the project is to design and construct the groyne structures for coastal protection. The benefits of BIM use are stated for each state of the project. In the design stage, the groynes and the beach profile model helped to evaluate and optimize the design. Also, more realistic material volumes were obtained from the detailed design of the groynes. Collery et al. (2016) stated approximately 30% saving on design time and costs. Furthermore, it is mentioned that the 3D model was printed and used for better communication and information sharing with stakeholders. The 3D model was also used in the tender stage. The result of this use was summarized as more accurate tender submissions and less tender queries. Additionally, tender participants mentioned that they spent less time developing take-off. Moreover, BIM model was used by excavators for excavation and the placement of the rocks. The excavators

had a GPS control system that enabled to obtain detailed real-time survey data. As a result, the surveyors could avoid a hazardous situation during the survey. This realistic survey data was used to develop an as-built model of the structures used to compare with the design model. The comparison process was also automated by setting some rules and parameters.

Hua (2020) compared the traditional construction method of port breakwater and the construction method of port breakwater based on BIM technology. The project was divided into parts. Each part was modeled in Revit by using CAD drawings, required data and parameters. After modeling, clash detection was performed to obtain a coordinated model. Later, the quantities were obtained to estimate and control the cost. Additionally, a construction schedule was developed. During construction, with the help of the information model and related technologies, quality and safety management were performed effectively. It is indicated that the data produced during the construction process added to the information model in order to evaluate the process and current situation, which has helped responding quickly in case of need. Also, it is mentioned that this information can be used in the operation and maintenance processes. Hua (2020) concluded that "...the construction method of port breakwater based on BIM technology can improve the load-bearing performance of island breakwater structure, which is of great significance to the improvement of port breakwater engineering quality."

Wahab et al. (2011) aimed to generate an object-oriented model and perform construction and operability (C&O) assessments before the construction phase. It is stated that there were some attempts for C&O assessments that did not include all required factors and the time factor in the design of port terminals. Wahab et al. (2011) claimed that without object-oriented models, the assessment process becomes more difficult. Also, due to the complexity of the project, more effective ways are needed for coordination. At this point, BIM models are expected to help data integration between model alternatives and their constructability principles. It is concluded that the previous studies of modeling ports expose that the available data is not sufficient for C&O purposes (Wahab et al., 2011).

Mahdi et al. (2019) indicated that marine construction projects are not managed effectively due to change orders, inadequate planning, insufficient site investigation data, poor supply, and lack of awareness of health and safety precautions. In their study, they aimed to obtain and share information efficiently between both project phases and the project parties. The proposed framework consists of project modeling, simulating operation phase, entering soil properties, structural design, construction logistics, developing multi-dimension plans, and monitoring and controlling. The proposed integrated information system was implemented in a case study. As a result, the project parties improved their capabilities. The decrease in time, cost, and expected risk were observed. Also, it allowed the evaluation of sustainable material alternatives before construction.

Garibin and Ol'Khovik (2018) aimed to create a BIM model of a port facility for its life cycle. The initial BIM model contained basic geometries, material, physical and mechanical properties, location and deformation information. Then, the purpose is to monitor loads, deformations, and climatic conditions automatically. The problem in creating an information model of existing port facilities is the missing documentation of the project, or the available information cannot be digitized. Also, previous load cycles are not known. The developed BIM model is considered as “an electrical passport”. It can be a helpful tool for planning the following surveys and repairs objectively. The condition of the structure can be recorded and tracked during its lifecycle. Moreover, this electrical passport can eliminate excessive and inefficient operations, and increase the reliability of the facility.

In conclusion, the previous applications of BIM in marine engineering focus especially on port projects. The implementations target the operation phases of the project. Monitoring conditions of the structures, damage control, and maintenance scenario development are mentioned as beneficial uses of BIM in management and operation phases. Additionally, the previous studies generally stated that the design and the construction process of marine engineering projects are not managed effectively. BIM is suggested to solve the management issues, decrease cost and

labor, optimize the design, and improve the quality of the design and the design process.

On the other hand, the design phase in every project is an iterative process. The specifications, standards or guidelines are followed and implemented by the designer, engineers or architects in every construction project. The design should be controlled according to them at every stage of the design procedure. As the design change, it should be controlled whether the design or the design changed is performed according to the standards or the guidelines. This procedure is prone to errors and it is time consuming. Due to the nature of this procedure, the design development and optimization is not performed effectively.

The literature review shows that there were not enough examples of BIM applications on marine structures and the limited previous studies were mostly about the port projects. It was indicated that Building Information Modeling (BIM) can be used in the management and construction process of the port structures. Also, it was stated that BIM model of the port structures enables monitoring the situation of the structure during its lifecycle and helps to plan maintenance strategy. The studies mentioned that the benefits of BIM technologies such as cost management, clash detection and sharing information between the parties can be beneficial for the effectiveness of the projects. In addition to BIM application in the port projects, there was an example of a coastal protection project in which the capabilities of BIM such as optimization of design, accurate tender submissions, and quantity take-offs and increase in health and safety in the construction stage were utilized. As a result, BIM model of a marina and the code compliance checking in the marina projects has not been investigated yet. Thus, the aim of this study is to improve design process in marina projects via semi-automated code compliance checking by utilizing Building Information Modeling (BIM). A semi-automated code compliance process is proposed to evaluate the BIM models of marina layouts whether the national marina design guidelines is followed during design in an effective way. Another goal of the study is to generate marina layout by placing some marina elements such as finger piers, anchors and boats in the model automatically. Thus, the marina design can be

optimized and meet the expectations of the target users. As a result, the efficiency and quality can be enhanced in the design process. Moreover, it is expected that this study can make a way for future BIM implementations in coastal structure projects.



CHAPTER 3

RESEARCH METHODOLOGY

In this chapter, the methodology for code compliance checking of marinas is introduced. Firstly, the marina layout is described. Secondly, modeling of the marina using BIM methodology is described. Finally, the developed algorithms to be used for code compliance checking are explained in detail.

3.1 MARINA LAYOUT

Marina layout can be considered as the allocation of physical elements of marinas to provide a safe berthing area for yachts. Besides the water area used by the yachts and their owner, the marina layout contains land areas that ensure a variety of services and recreational activities for non-boater users. However, the scope of this study is only focused on the water area of a marina.

Marina layouts consist of breakwaters, piers, floating piers, finger piers, anchors, and boats. Additionally, there are some harbor structures to supply repair, maintenance, and storage services. Figure 3.1 presents an example marina layout showing breakwaters, main pier, fixed and floating piers, finger piers. In this study, three layouts are modeled and analyzed: (a) Finger Pier Arrangement shown in Figure 3.2, (b) Mediterranean (Anchor) Arrangement shown in Figure 3.3, (c) Combined Layout shown in Figure 3.4.

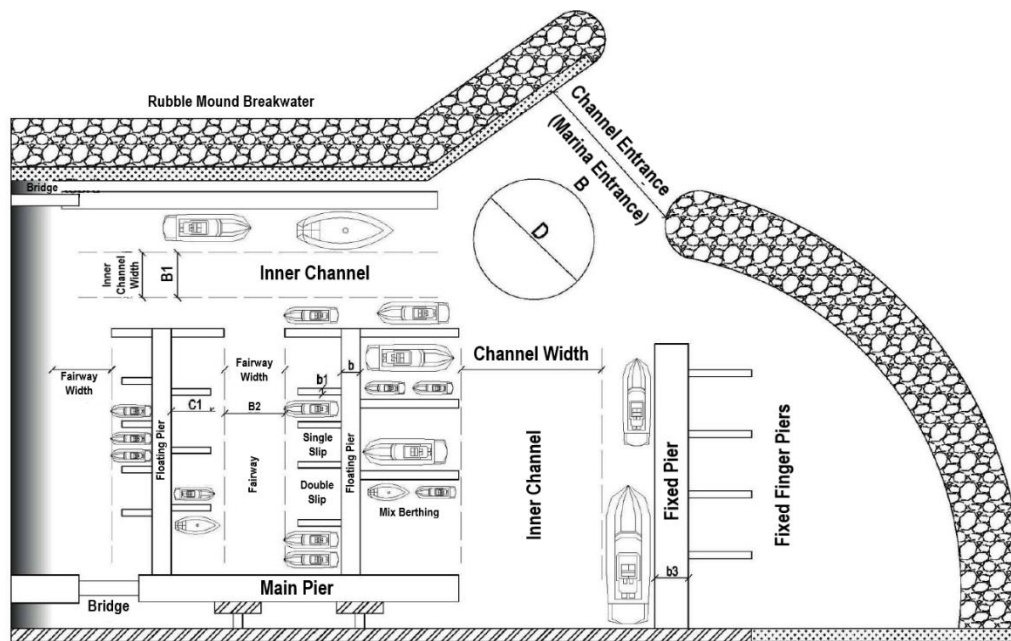


Figure 3.1. Typical Marina Layout (Adapted from Coastal Structures Planning and Design Technical Manual by AYGM, 2016).

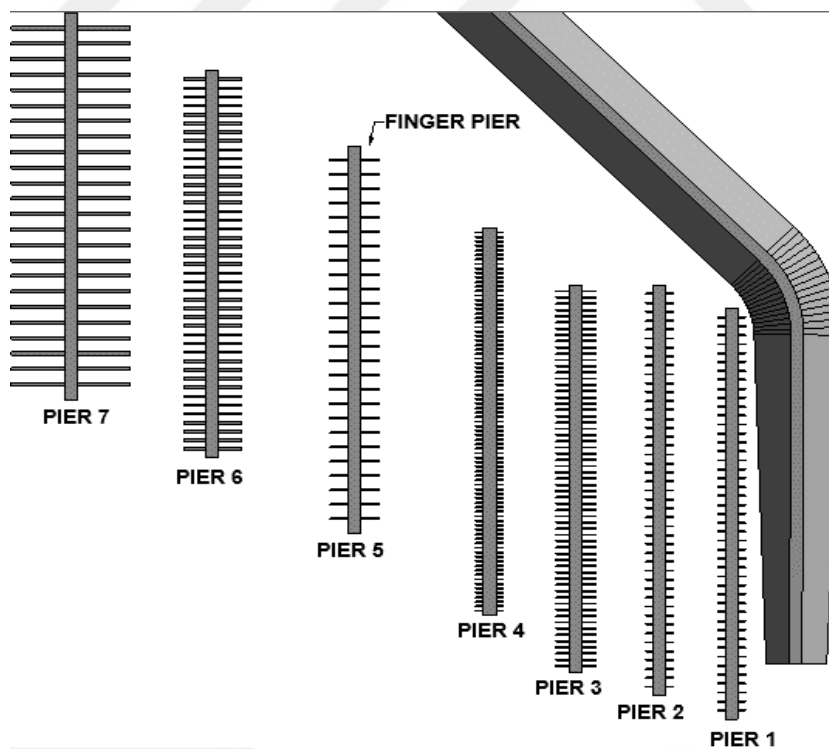


Figure 3.2. Finger Pier Arrangement

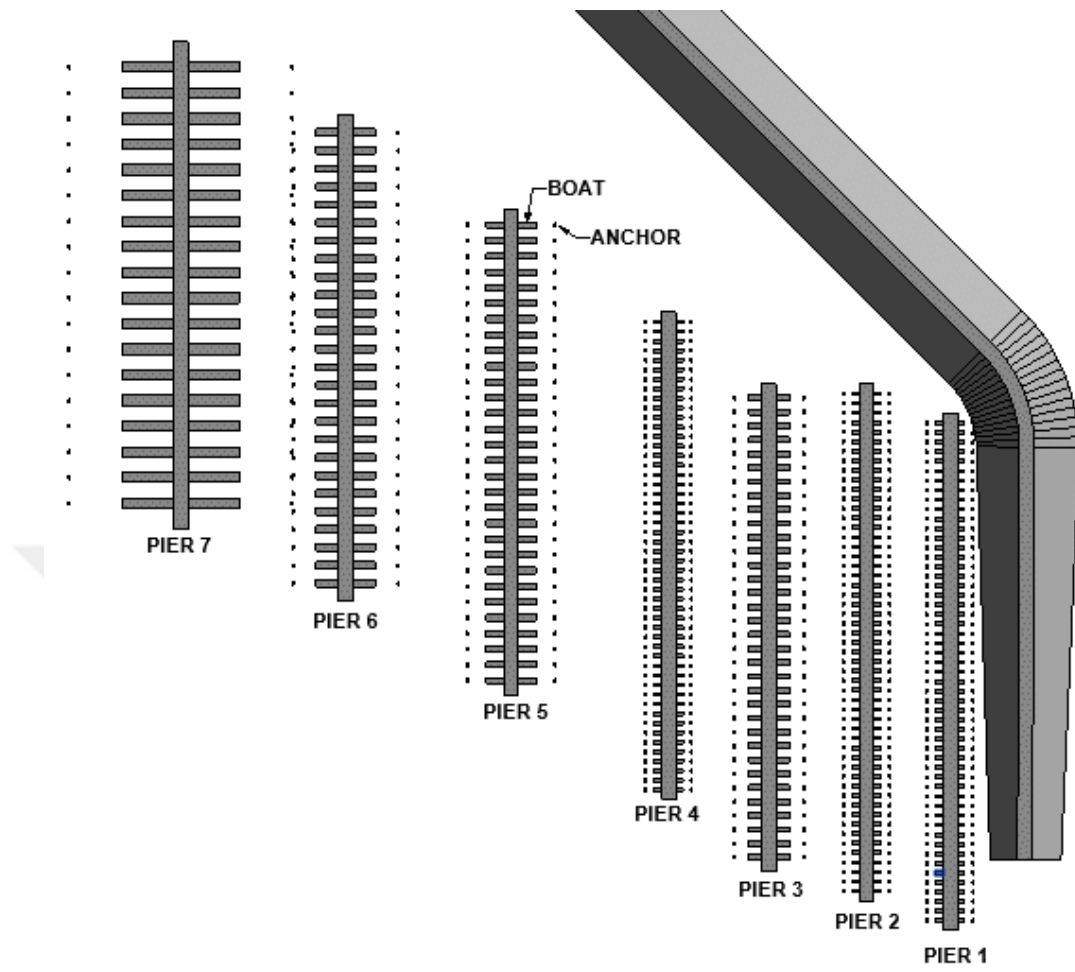


Figure 3.3. Mediterranean (Anchor) Arrangement

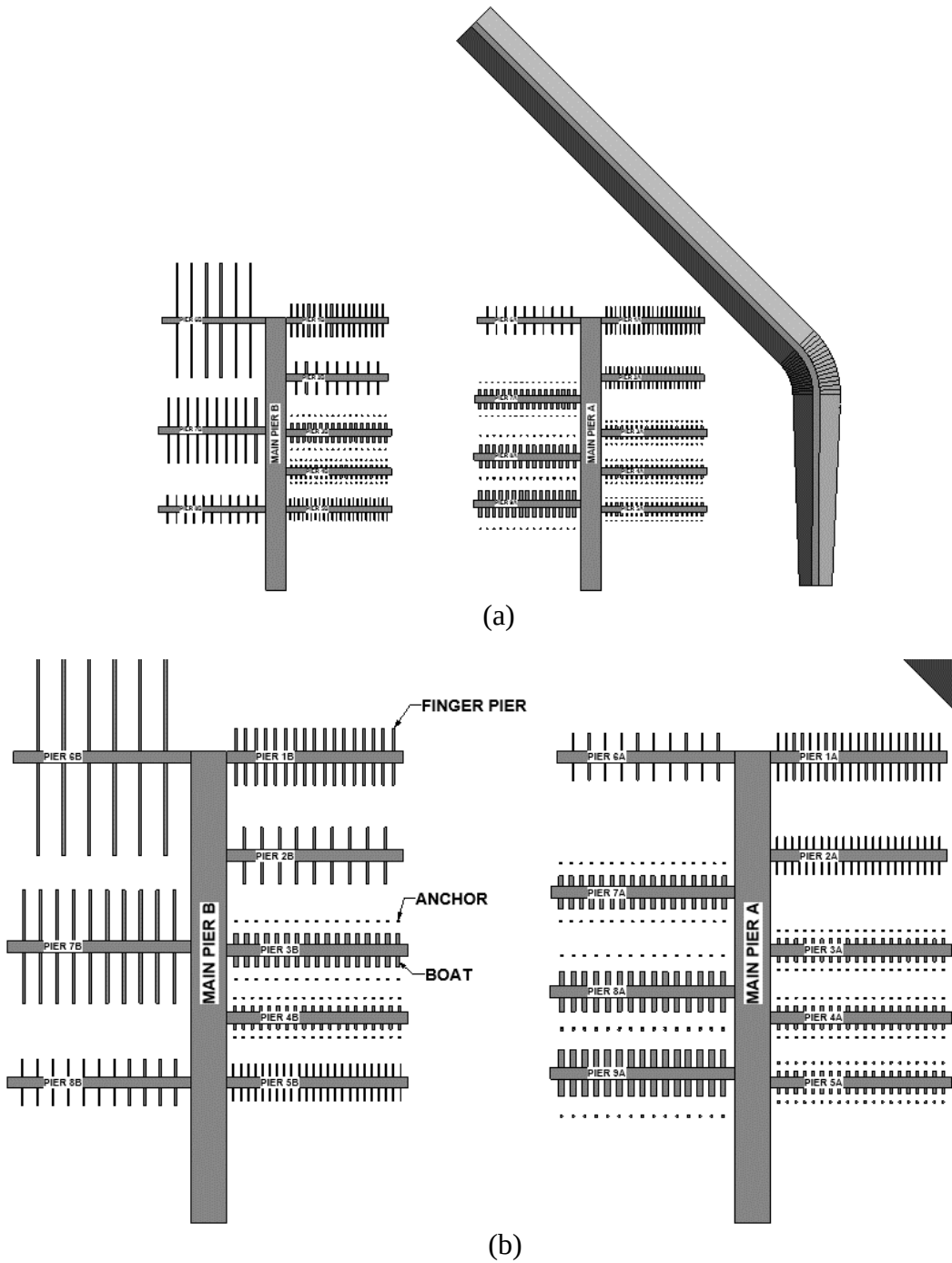


Figure 3.4. Combine Marina Layout. (a) General (b) Detail

Modeling and checking the breakwater, pier, finger piers, anchors, and boats are in the scope of this study. Their physical interaction in a marina layout is investigated. Their functions are explained below.

- **Breakwater:** The main function of the breakwater structure is to provide protection from waves and currents. It is described as “those devices or structures that effectively block 90% of the wave action” by PIANC (2016b). It has different types such as fixed, floating, solid, porous, etc. Generally, different sizes of natural rocks are used to build a breakwater. Some artificial structures like x-blocks can also be used. Figure 3.5 shows a rubble mound breakwater.

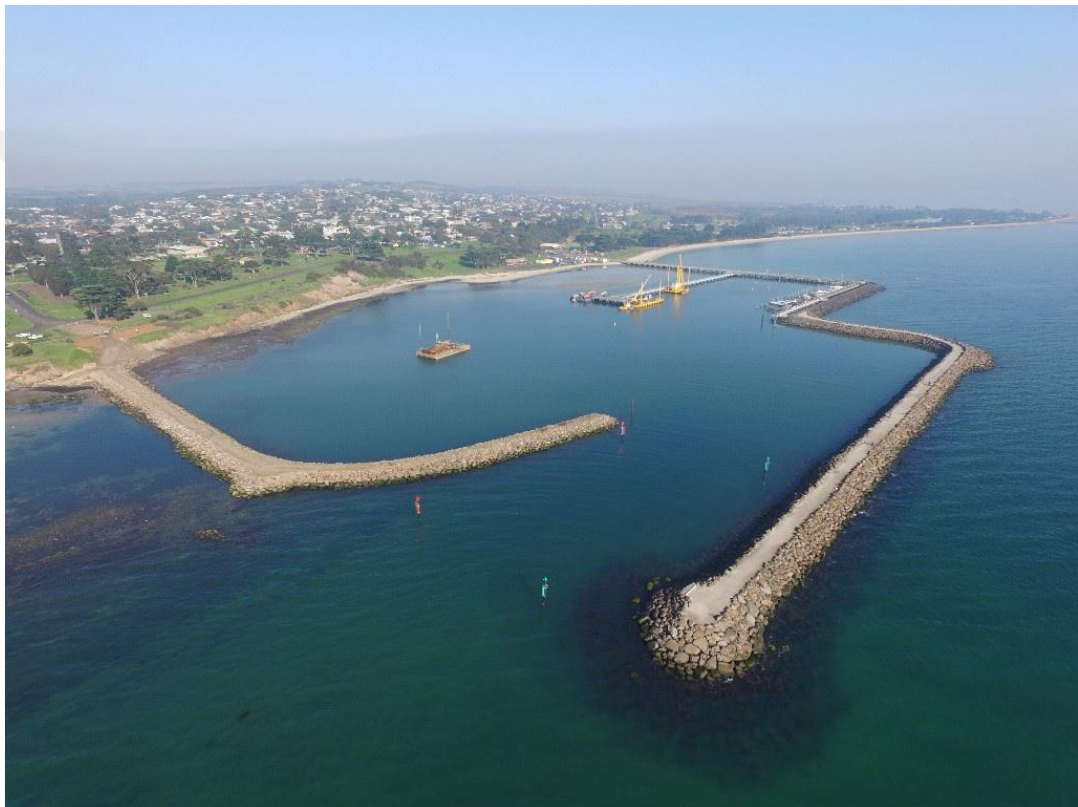


Figure 3.5. Rubble Mound Breakwater. (AWmaritime, 2017)

- **Pier:** Piers can also be named as docks. This structure is referred to as piers in this study. Piers are categorized as floating piers and fixed piers. The selection depends on the functionality and environmental conditions such as water depth, water fluctuations and soil conditions. Fixed piers are generally permanent structures that are constructed on site. Floating piers can be both temporary and permanent. The temporary floating piers can be moved to

change the marina layout to respond to changing boat distribution in time. In this study, the piers are not specified as floating or fixed structures. Their geometrical representations and layouts are the main concern of the study. Two types of piers are modeled: Pier and Main Pier.

- *Main Pier*: It is used as a single large structure that piers are placed on its sides. It is only used in “Combined Layout” case. “Combined Layout” can be seen in Figure 3.4.
- *Pier*: It is a relatively smaller berthing structure than “Pier”. It is used in all layout cases. The finger piers, boats, or anchors are placed on its left and right sides in the models.
- **Finger Pier**: It is a berthing arrangement structure that is given in Figure 3.1. Generally, this arrangement is used in North America. Finger piers are used for boat separation, more secure berthing, and easy boarding from the boat. It is preferred in areas that especially have wave and boat wake action (Tobiasson & Kollmeyer, 1991).
- **Anchor**: Anchors are one of the main structures in Mediterranean mooring arrangements. In Mediterranean mooring, boats are berthed side by side with the help of anchors. The boat's stern is connected to the pier, and the bow of the boat is connected to the anchor. Mediterranean mooring enables the berthing of a large number of boats in marinas (ASCE, 2012). The Mediterranean mooring style can be seen in Figure 3.6.

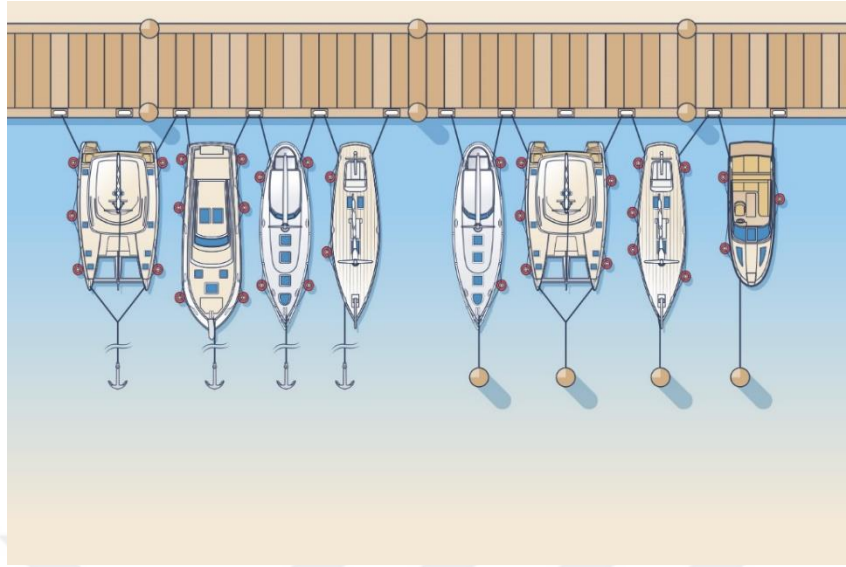


Figure 3.6. The Mediterranean Mooring. (MacKenzie, 2016)

- Boat:** The name of boat, vessel, and yacht refers to the marine transportation crafts used mainly for pleasure. Figure 3.7 demonstrates the several yachts that berthed in the Yalıkavak Marina in Turkey. Their general geometrical features are the main concern of the study, so they are not modeled in detail. Detailed information about boat types and their dimension are given in “3.2.4. RULE CHECKS” section.



Figure 3.7. Yachts in Yalıkavak Marina, Turkey. ([Megayachts], 2020)

The allocation of elements is unique for each marina due to variations of physical, economic, environmental conditions, and customer demands. Therefore, there are not any strict guidelines for marina layout design, as mentioned before. Nonetheless, some guidelines state minimum planning requirements for marina layout to provide safe areas. In Turkey, “Coastal Structures Planning and Design Technical Manual (AYGM, 2016)” is used as a marina design guideline. Therefore, in the scope of this study, it is followed during the code checking process of a marina design. The following are controlled for a marina layout by using the proposed code compliance method in the study:

- i. Slip (berth) width
- ii. Inner channel width
- iii. Distance between piers (Fairway Width)
- iv. The required minimum water level in a marina basin (Water level check)
- v. Berth length

All of these controlled distances are represented in Figure 3.1. In addition to the semi-automated code compliance check, the marina layouts are created by placing finger piers, anchors, and boats on the previously located piers according to these rules.

3.2 DEVELOPING MARINA MODEL

Autodesk Civil 3D 2021 and Autodesk Revit 2018 software are used for creating marina models in this study. The bathymetry model and the breakwater model that follows the bathymetry geometry are first created in Autodesk Civil 3D. After that, they are imported to Autodesk Revit. The other elements like boats, piers, main piers, finger piers, and anchors are created in Autodesk Revit.

In this section, the modeling steps of a marina are explained. Firstly, the bathymetry and breakwater model are described. The steps and the information related to

creating a marina model in Revit are stated in detail. Lastly, the rule checking process via Dynamo tool is described.

3.2.1 MODELING BATHMETRY IN CIVIL 3D

Bathymetry can be explained basically as the topography model of sea or ocean bottom. In this study, it is required for two reasons. The first one is to create a more realistic physical representation of the breakwater in the project. The second reason is to evaluate the water level in the marina area. The bathymetry is modeled by using point data of the Izmir Bay Area. The point data is obtained from The GEBCO (General Bathymetric Chart of the Oceans) (2014). The GEBCO_2014 Grid dataset (Version 20150318) is a continuous terrain model for ocean and land with a spatial resolution of 30 arc seconds. The points consist of longitude, latitude, and elevation data in .txt format. All modeling process is performed in the Civil 3D environment. A detailed explanation of each modeling step will be given in the following sub-sections.

3.2.1.1 Importing Points into Civil 3D

The points of the Izmir Bay Area should be imported into the Civil 3D to create a realistic representation of the region. Before importing points, some setting should be adjusted. Otherwise, the model may not be created in true geolocation.

First of all, “Zone” and “Coordinate System” are set as “Turkey” and “UTM-WGS 1984 datum, Zone 35 North, Meter; Cent. Meridian 27d E”, respectively. Figure 3.8 demonstrates the zone settings that are mentioned.

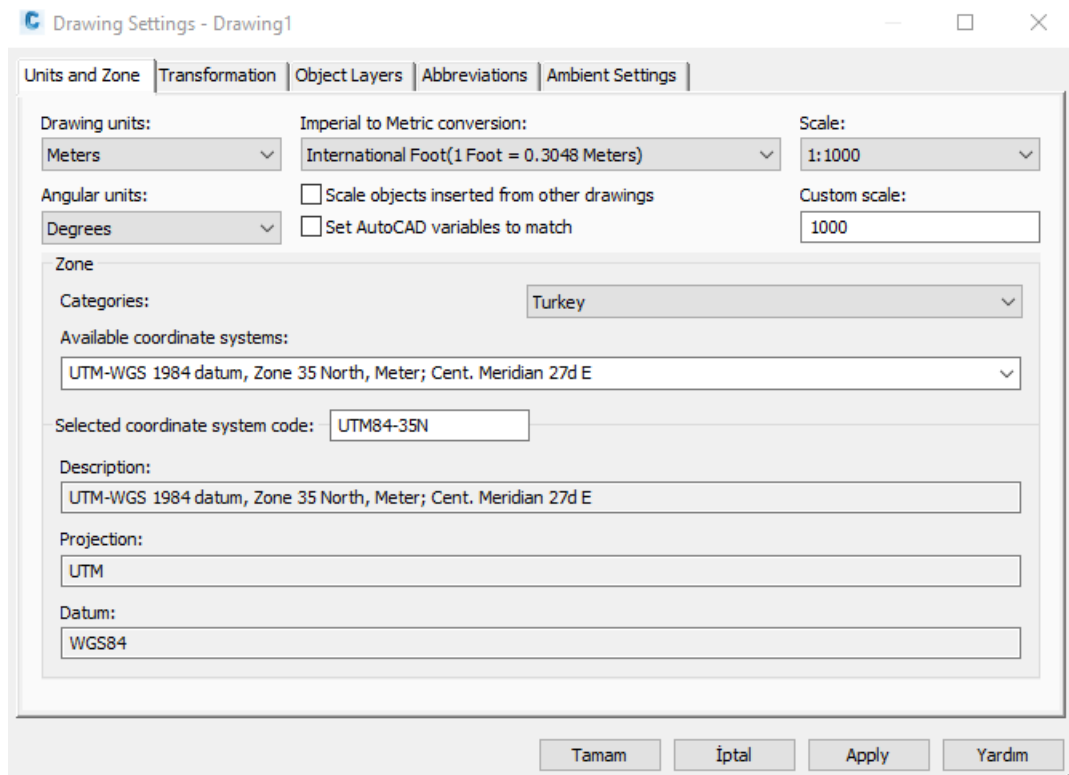


Figure 3.8. Zone Settings in Autodesk Civil 3D

After that, a specific point file format which is matching the data is created. The format name is given as “Long Lat Z” which specifies the columns’ values in the point file of the Izmir Bay Area region (see Figure 3.6). The data in the first and second column is longitude and latitude values of the points, respectively. The elevations of the points are given in the last column. Default file extension is set to .txt which is the extension of the point file. Also, Format option is set to the “Delimited by – Space”. After the format settings, the columns in the file are specified in the column region. All these settings are given in Figure 3.9.

Point File Format

Format name:
Long Lat Z

Default file extension:
.txt

Comment Tag:

☐ Coordinate zone transform
Zone:

Format options

☐ Columnated

☒ Delimited by: ☐

☐ Read no more than 0 points

☐ Sample every 0 points

DECDEG Longitude	DECDEG Latitude	Point Elevation	<unused>	<unused>	<unused>	<unused>	<unused>

Load... Parse OK Cancel Help

Figure 3.9 Settings of New Point File Format

After the file format is created, the Gediz region's points are imported by “Point Creation Tool” under the “Points” tab. Figure 3.10 shows a preview of the point data.

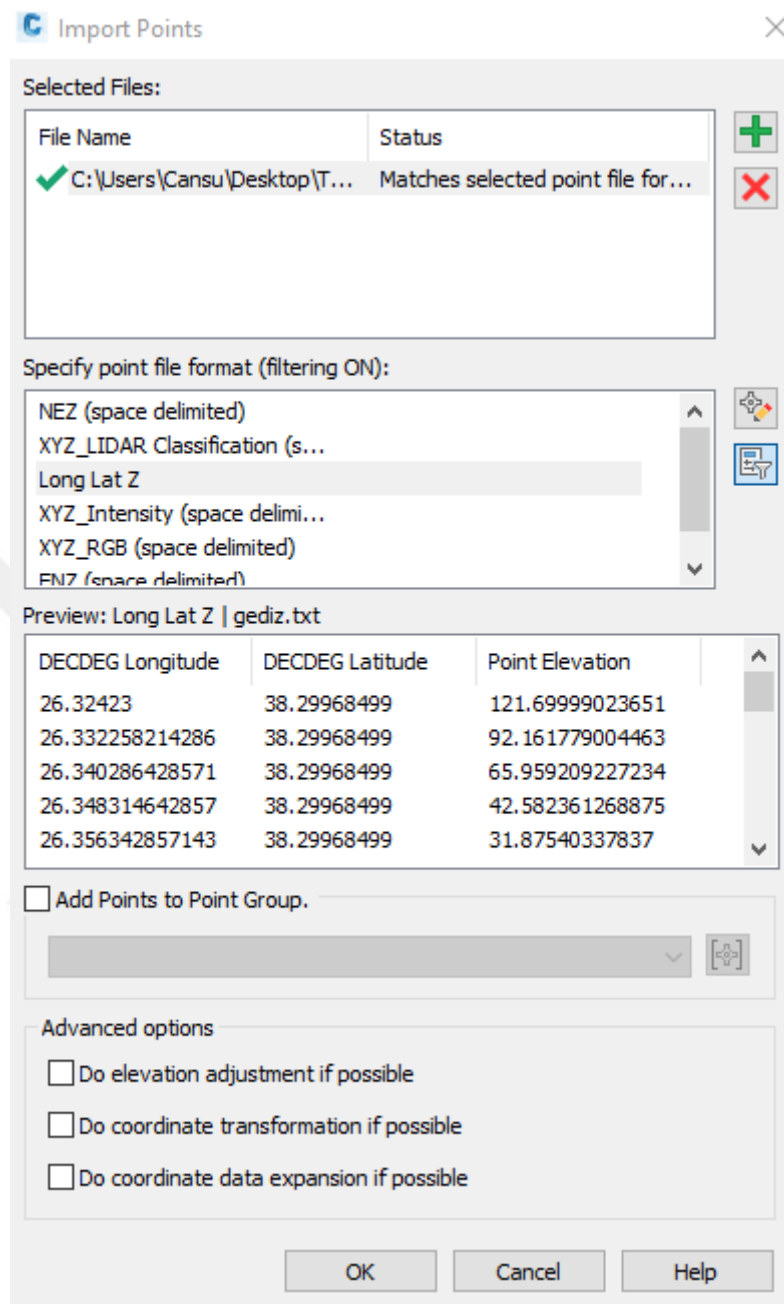


Figure 3.10. Importing Point Data in Civil 3D

The points are imported as COGO points in Civil 3D. The final result can be seen in Figure 3.11.

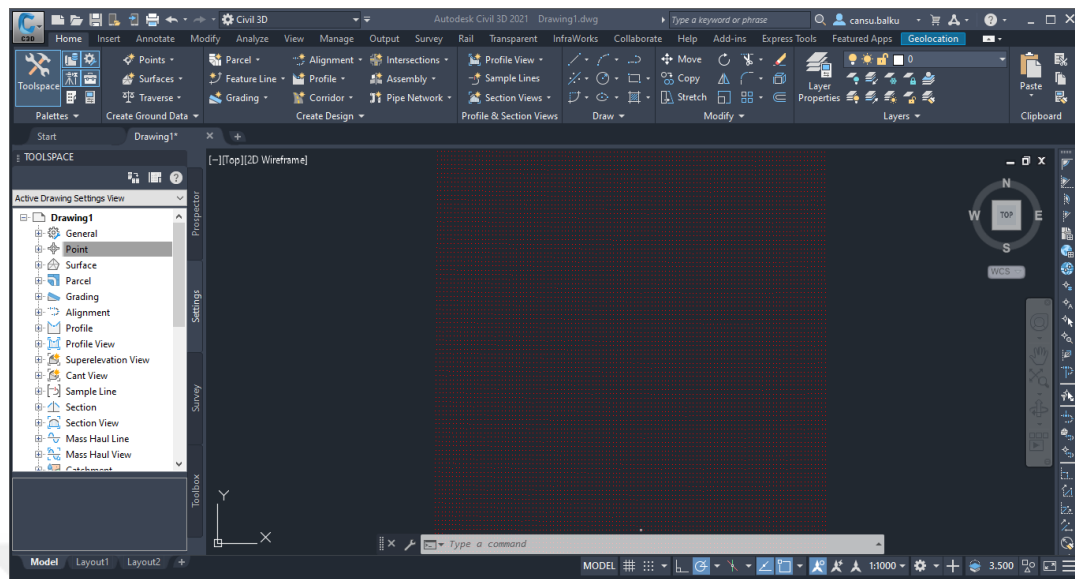
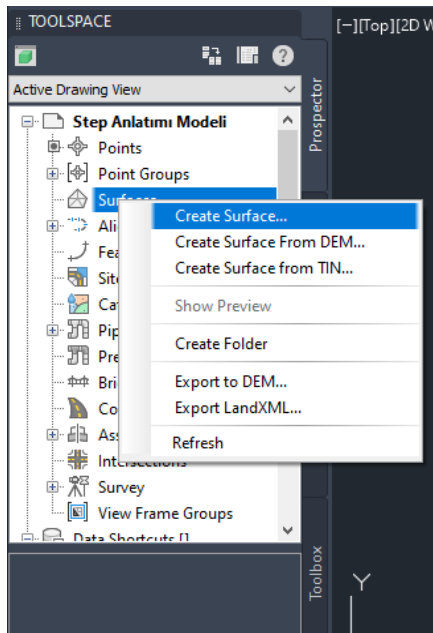


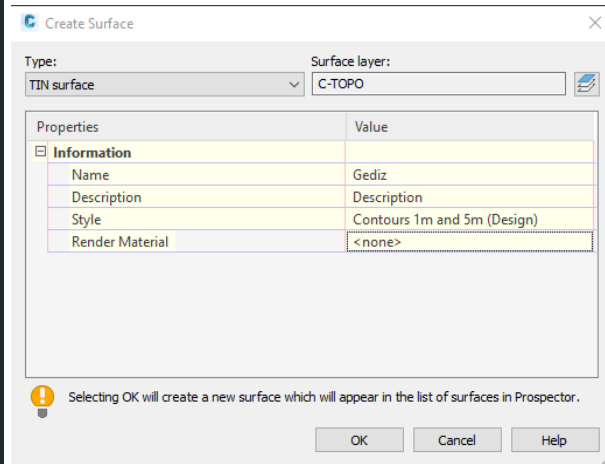
Figure 3.11. Points of Izmir Bay Area Region in Civil 3D

3.2.1.2 Creating Surface from Imported Points in Civil 3D

The surface is created from the imported points by the “Surfaces” part in “TOOLSPACE”. At first, there is not any surface elements in the model yet. The first step enables only entering surface information into the model as shown in Figure 3.12.



(a)



(b)

Figure 3.12. Creating Surface names as “Gediz”

Under “Gediz”, there are some categories. “Definition” category has some options to create surfaces. In this project, the “Point Groups” option was used. All imported points are used to create a surface that represents the Izmir Bay Area. Figure 3.13 displays the surface with a real map in the background.

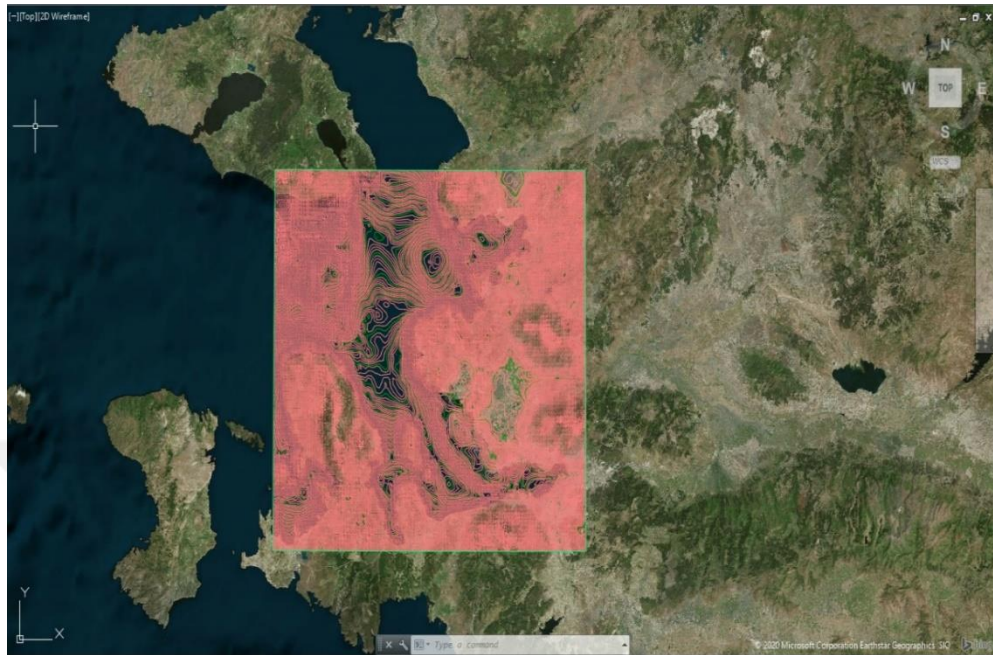


Figure 3.13. Bathymetry Model in Civil 3D

Nevertheless, the surface generated contains a big area. It is difficult to import it to Revit environment and run Dynamo scripts with it. Therefore, the region should be reduced. A rectangle around the Gediz area that the marina model will be created is drawn, as shown in Figure 3.14. This rectangle is used as a “Boundary” element to rebuild the surface in this boundary. Finally, the region that the marina will be located is recreated.

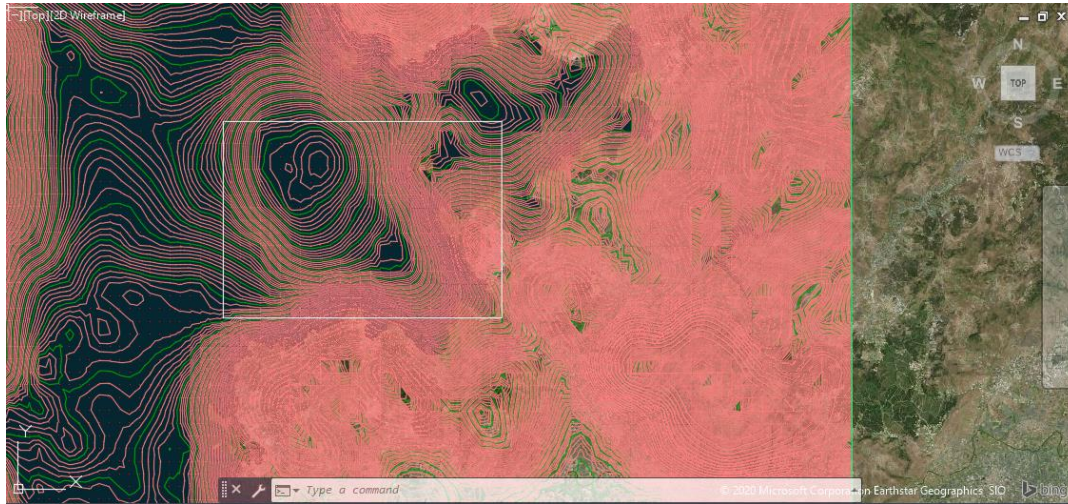


Figure 3.14. Creating a Boundary For Gediz Area

After the operations explained above, the surface is exported by the Civil 3D drawing option. Figure 3.15 indicates the export preferences.

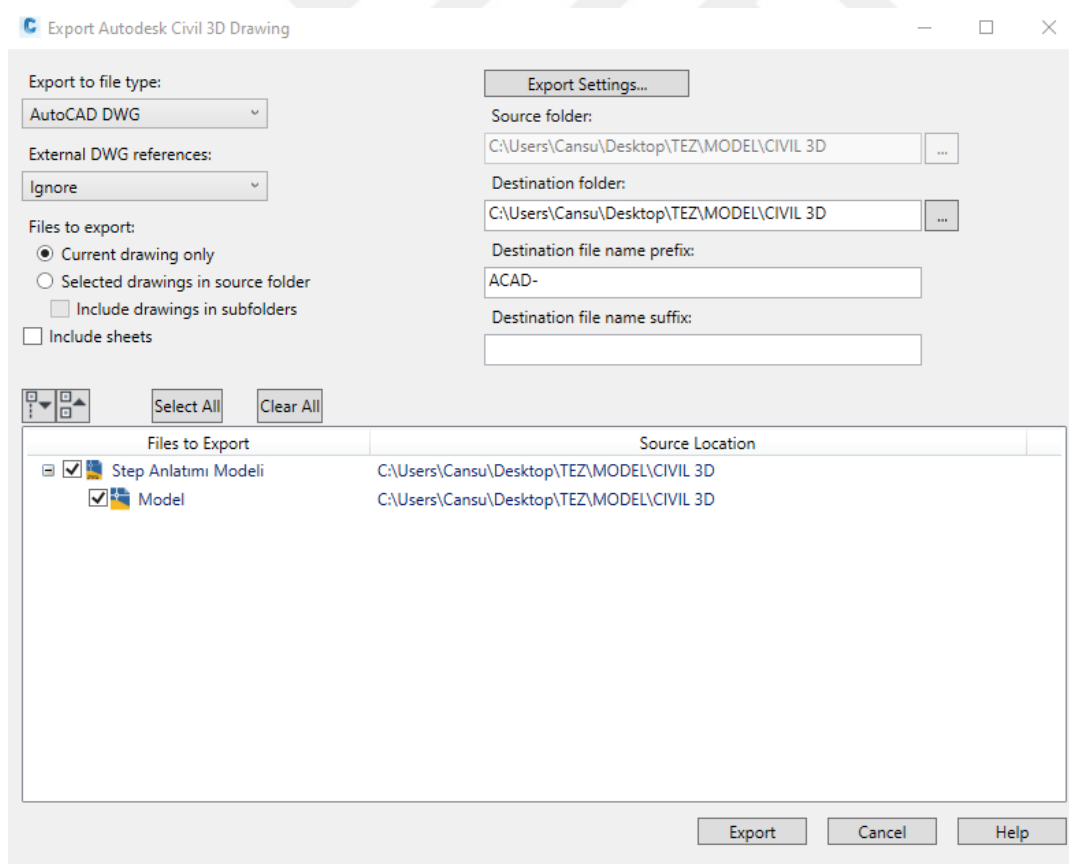


Figure 3.15. Export Preferences in Civil 3D

Later, this export was used as an import object for the Topography tool in Revit to create the Revit topography element. The bathymetry model in the Revit environment can be seen in Figure 3.16.

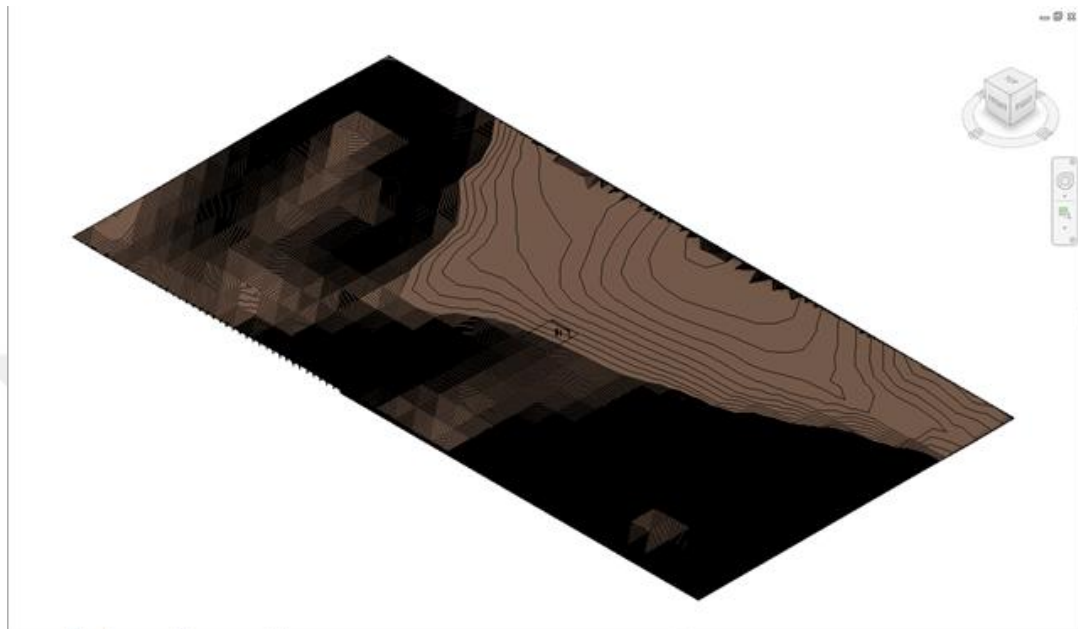


Figure 3.16. Bathymetry Model in Revit

3.2.2 CREATING BREAKWATER IN CIVIL 3D

The steps about creating breakwater in the Civil 3D environment will be explained in this section.

First of all, the cross-section of the breakwater and the center alignment of the breakwater are drawn. A subassembly from the breakwater cross-section is created, and the cross section's origin is specified. After that, a shape is created from the subassembly. The cross-section of the breakwater displayed in Figure 3.17.

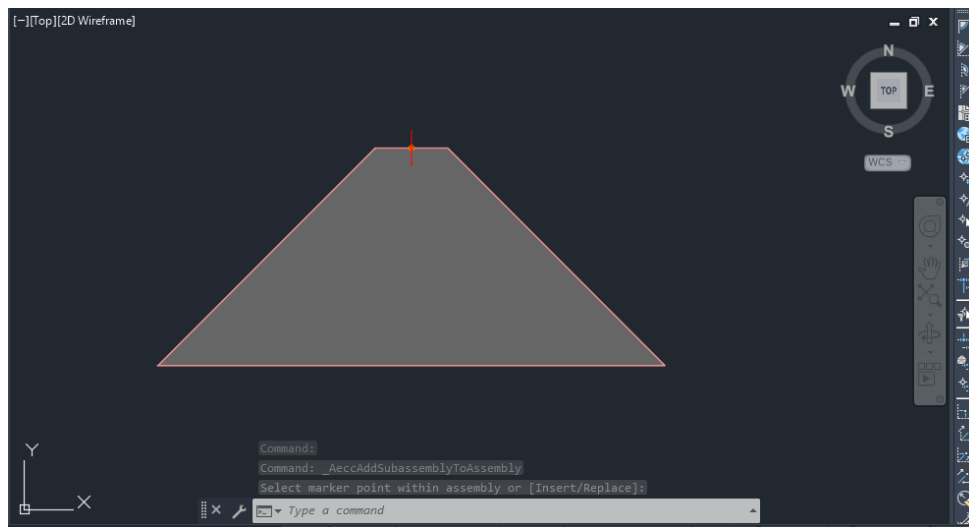


Figure 3.17. The Cross-Section of The Breakwater

Next, the breakwater alignment is created. “Create Alignment” properties are shown in Figure 3.18.

Figure 3.18. Create Alignment Properties

Later, a surface profile is created through breakwater alignment by the “Create Surface Profile” command. This surface profile follows the elevation changes through the breakwater alignment, as shown in Figure 3.19. An elevation line was drawn in this profile by “Profile Creation Tools”.

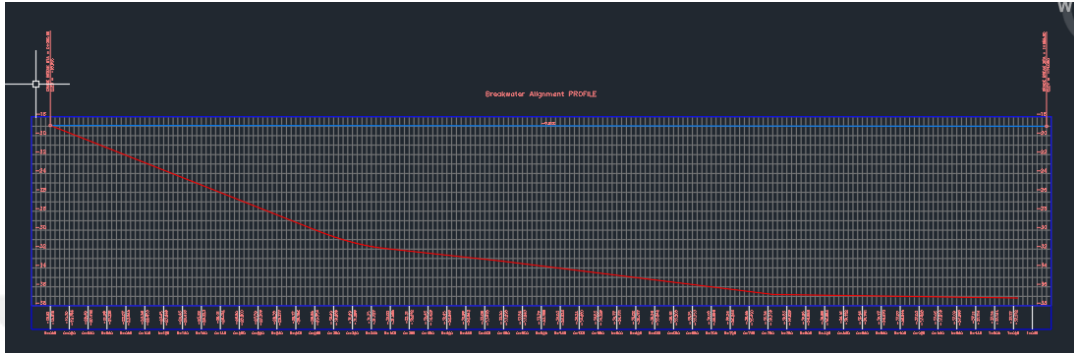


Figure 3.19. Surface Profile Through Breakwater Alignment

The next step is to create a corridor through the area where the breakwater is located. As a result, the breakwater cross-section is placed at every 5 meters through its alignment. After creating a corridor, solid is extracted from that corridor, as shown in Figure 3.20. That solid represents the breakwater. However, the bottom of the breakwater is not compatible with the bathymetry.

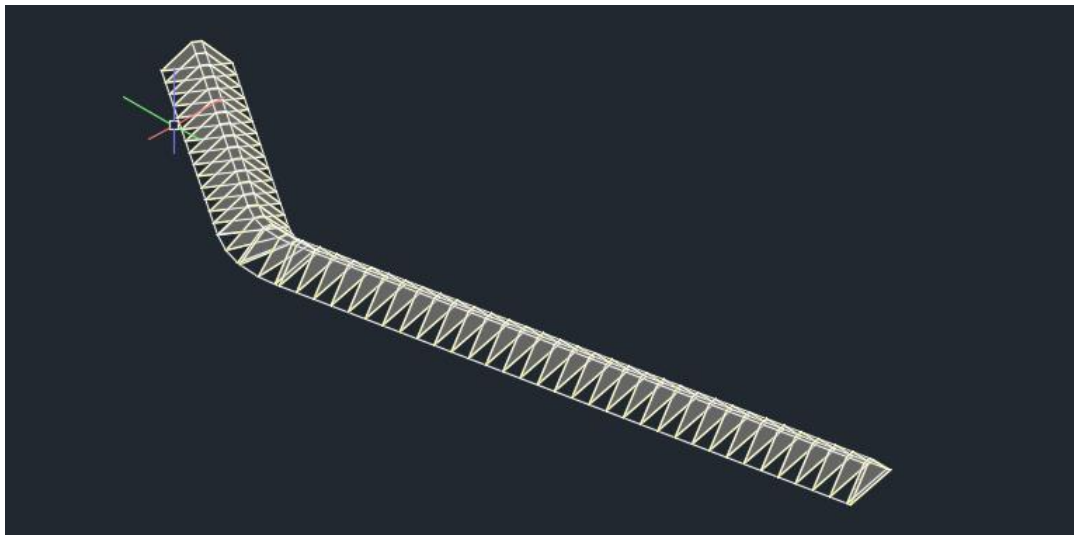


Figure 3.20. The Solid Created From The Corridor

The following steps are performed to match the base of the breakwater and the bathymetry surface. Another solid extraction is performed from the surface model by the “Extract Solid from Surface” command. After creating solids from the surface and corridor, the solid from the surface is subtracted from the breakwater solid. The result represents a realistic breakwater model, whose base follows the bathymetry surface. The final result of the breakwater model element is displayed in Figure 3.21.

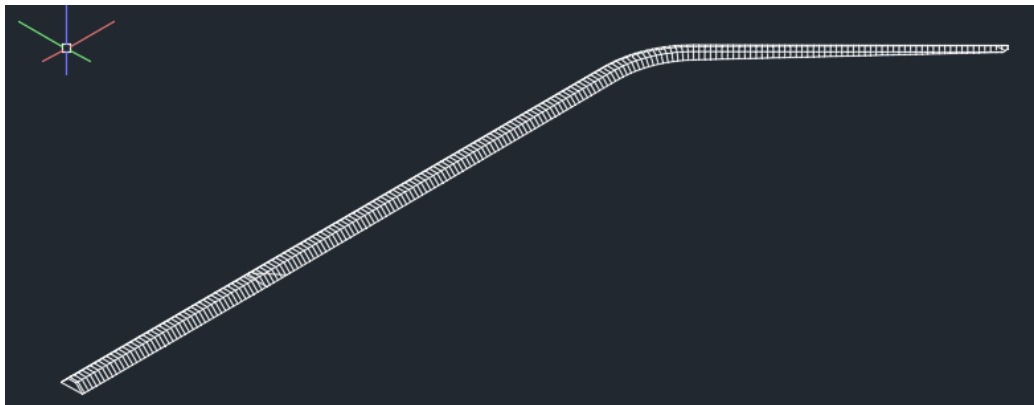


Figure 3.21. Breakwater Model in Civil 3D

This model is exported to create a breakwater in Revit. The export is imported in the General Model family template. Therefore, it is converted to a Revit family element (Figure 3.22).



Figure 3.22. Breakwater Model in Revit

3.2.3 MARINA MODEL IN REVIT

In this section, creating a marina model in Revit is presented in steps. Firstly, the general information about BIM model in Revit is explained. Later, parameters that are used in Revit families are described. Lastly, the features of the families in the marina model are given.

3.2.3.1 GENERAL INFORMATION ABOUT BIM MODEL IN REVIT

A BIM model can be considered as a digital representation of a structure. It contains both physical and functional information. The real-world elements can be created in the digital world. Their functionalities such as the relationships to their surrounding elements, their behaviors, and capabilities, can be implemented in the digital world as in the real world. Autodesk Revit software, used in this study, enables creating marina elements as BIM objects. Besides the predefined attributes to reflect features of real-life elements, Revit provides flexibility to customize objects to control the geometry and behavior of the elements.

Every element in Revit is created by a family. Revit family is described as "a group of elements with a common set of properties, called parameters, and a related graphical representation" (Autodesk Knowledge Network, n.d.). Revit family contains geometric and behavioral features of a structural component such as finger piers, anchors, or boats in this study. In Revit, a 3D model of an element is created, then, by using parameters, its geometry can be changed or updated in a BIM model according to the needs and alterations in the design. Also, every element's functionality can be identified and represented, or stored as information via parameters. The parameter is described as "Parameters store and communicate information about all elements in a model. Parameters are used to define and modify elements, as well as to communicate model information in tags and schedules." (Autodesk Knowledge Network, n.d.). Parameters can be created specifically for a

project or for an element in the project. There are four types of custom parameters in Revit, as explained in Table 3.1.

Table 3.1 Revit Parameter Types

Parameter Type	Description	Examples
Project Parameters	Project parameters are specific to a single project file. They are added to elements by assigning them to multiple categories of elements, sheets, or views. Information stored in project parameters cannot be shared with other projects. Project parameters are used for scheduling, sorting, and filtering in a project.	A project parameter can be used to categorize views within a project.
Family Parameters	Family parameters control variable values of the family, such as dimensions or materials. They are specific to the family. A family parameter can also be used to control a parameter in a nested family by associating the parameter in the host family to the parameter in the nested family.	Family parameters such as Width and Height may be used in a Door family to control the dimensions of the different door types.

Table 3.1 Revit Parameter Types (continued)

Parameter Type	Description	Examples
Shared Parameters	Shared parameters are parameter definitions that can be used in multiple families or projects. After you add a shared parameter definition to a family or project, you can use it as a family or project parameter. Because the definition of a shared parameter is stored in a separate file (not in the project or family), it is protected from change. For this reason, shared parameters can be tagged and scheduled.	If a parameter in a family or project needs to be scheduled or tagged, that parameter must be shared and loaded in both the project (or element family) and the tag family. Shared parameters can be used when elements in 2 different families are scheduled together. For example, if you need to create 2 different Isolated Foundation families, and need the Thickness parameter of both families scheduled in the same column, Thickness needs to be a shared parameter that is loaded in both Isolated Foundation families.
Global Parameters	Global parameters are specific to a single project file, but are not assigned to categories. Global parameters can be simple values, values derived from equations, or values taken from the model using other global parameters. Use global parameters to drive and report values.	A global parameter can assign the same value to multiple dimensions. You can also set the position of one element by the size of another element. For example, beams can be driven to consistently offset from the floor they support. If the floor design changes, beams will respond accordingly.

Adapted from (Autodesk Knowledge Network, n.d.)

In this study, family parameters are used to control the families' geometrical features and provide information that can be used in the rule check process. All the parameters are the instance parameters, as shown in Figure 3.23. Instance parameters mean that the provided value in the parameter is specific to that single element. As mentioned before, every element is created by a family. When an element is placed in a model, that particular element is accepted as an instance of that family. When an instance parameter of that instance is changed, only that particular instance is affected from that change. Other instances created by the same family are not affected.

Parameter Properties

Parameter Type

☒ Family parameter
(Cannot appear in schedules or tags)

☐ Shared parameter
(Can be shared by multiple projects and families, exported to ODBC, and appear in schedules and tags)

Select... Export...

Parameter Data

Name:
Max Boat Draft

Discipline:
Common

Type of parameter:
Length

Group parameter under:
Other

Tooltip description:
<No tooltip description. Edit this parameter to write a custom tooltip. Custom t...
Edit Tooltip...

☐ Type

☒ Instance

☐ Reporting Parameter
(Can be used to extract value from a geometric condition and report it in a formula or as a schedulable parameter)

[How do I create family parameters?](#)

OK Cancel

Figure 3.23. Parameter Properties

3.2.3.2 PARAMETERS

In this study, there are six parameters that are defined and used in all families. They are categorized as “Common Family Parameters” and explained in detail in the following section. Other parameters specific to the families are described in “3.2.3.3 FAMILIES” section as a part of family information.

3.2.3.2.1 Common Family Parameters

In this part, the family parameters that are assigned to more than one family is explained. The “Common Family Parameters” are not for controlling the geometry of the families. They aim to provide information for rule checking processes. Figure 3.24 shows the “Common Family Parameter” and its values for a family in the marina model.

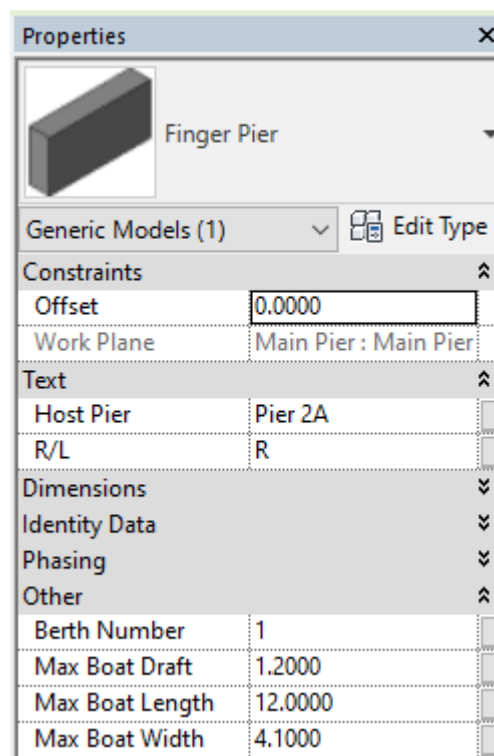
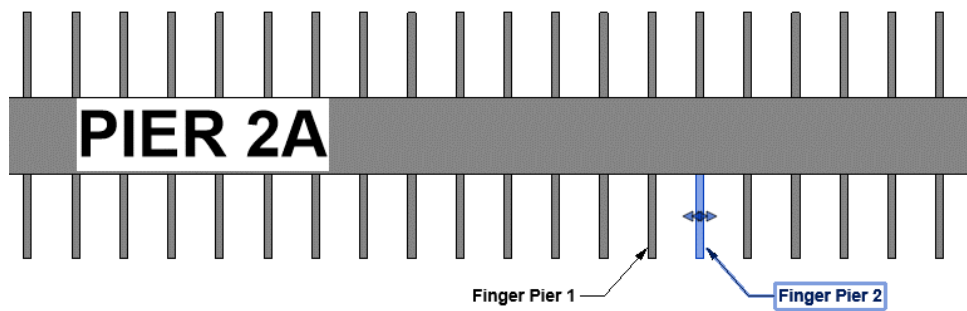


Figure 3.24. “Common Family Parameter” and Its Values for A Family in The Model

- **Host Pier:** “Host Pier” parameter is a common parameter for finger pier, anchor, and boat families. It contains information about the pier on which finger piers, anchors, or finger piers are placed. It is assigned as a family instance parameter to the elements in the project. It can be defined by the user differently for the instances that are modeled. The information can also be

assigned to the parameter by the dynamo scripts explained in the following chapters.

- **R/L (Right/Left):** “R/L” is an abbreviation of “Right or Left”. The parameter is assigned as a family instance parameter to components. It indicates whether the berthing piers, boats, or anchors are placed on the pier's right-hand or left-hand side. It can be specified by users or via “Placing Families on Piers” dynamo script for other families.

The information in “R/L” and “Host Pier” parameters help to group the berthing places, anchors, or boats. Therefore, the distances between the elements in these groups can be measured and compared according to the guidelines. For example, the required clearance according to the design guideline between Pier 2 and Pier 3, shown in Figure 3.25, can be calculated using the left-hand side berthing place length of Pier 2 and the right-hand side berthing place length of Pier 3.

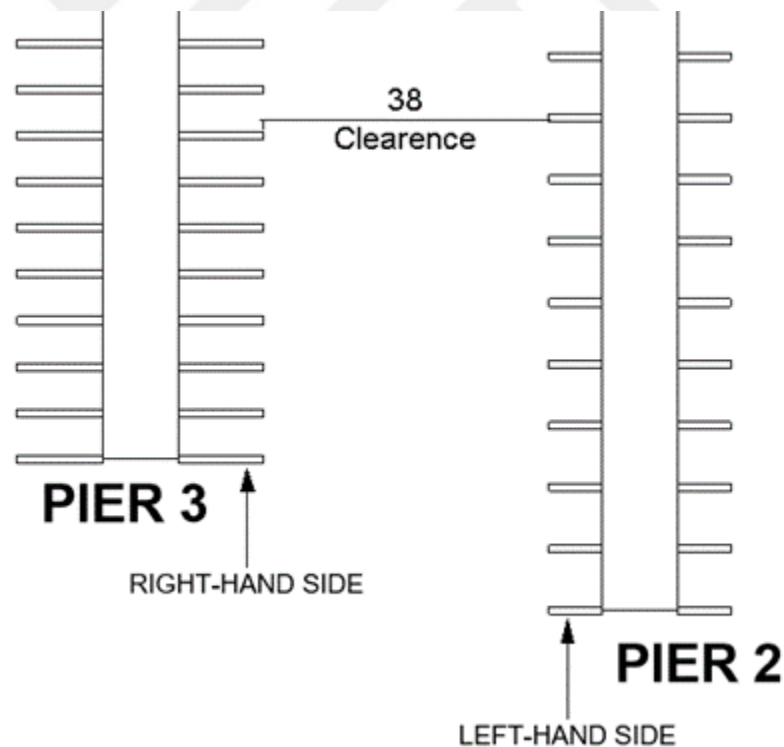


Figure 3.25. Representation of Clearance Between Piers

- **Maximum Boat Width:** “Maximum Boat Width” represents the widest width dimension of the largest boat seen in Figure 3.26 that berths on the pier or any dedicated berthing place. It is also known as the boat beam. It is assigned as a family instance parameter to piers, finger piers, anchors, and boats. The input for this parameter can be taken from beam to length ratio tables. However, this ratio is not applicable to every boat. The boat designer can also use different dimensions outside of the ratio. Therefore, the boat width is determined as a user-defined parameter for piers in this study in order to avoid design errors. It can be specified by users or via “Placing Families on Piers” dynamo script for other families.

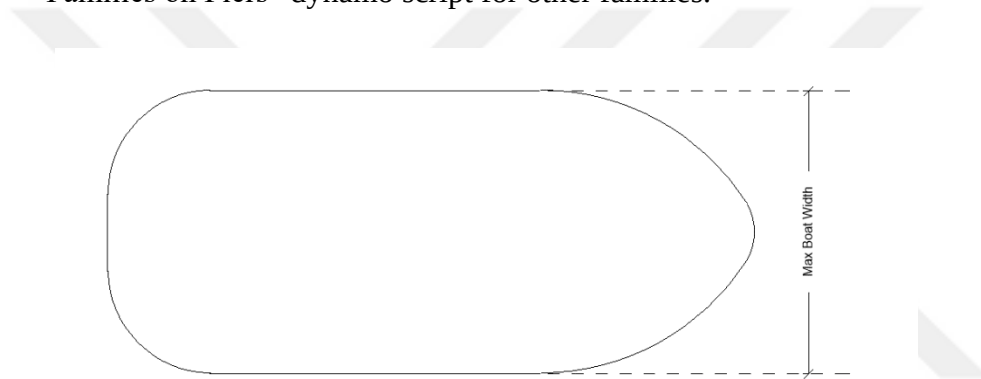


Figure 3.26. Maximum Boat Width

- **Maximum Boat Length:** “Maximum Boat Length” represents the length of the largest boat that berths on the pier or any dedicated berthing place. It is assigned as a family instance parameter to piers, finger piers, anchors, and boats. It can be defined by the user differently for the instances that are modeled via the same Revit family. As stated in *Marinas and Small Craft Harbors* (Tobiasson & Kollmeyer, 1991), boat length, in marina design, should be the total extreme length of the boat, including all projections for use in design considerations. Figure 3.27 illustrates the maximum boat length. The boat length is also determined as a user-defined parameter for piers in order to avoid design errors. It can be specified by users or via “Placing Families on Piers” dynamo script for other families.

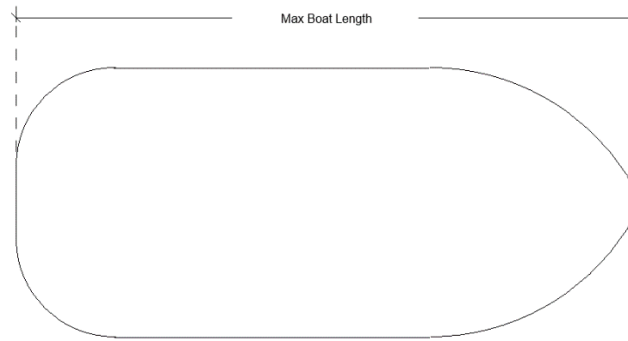


Figure 3.27. Maximum Boat Length

- Max Boat Draft:** The draft is a vertical dimension of a boat. It is defined as the maximum dimension between the keel of the boat and the water level in Coastal Structures Planning and Design Technical Manual (AYGM, 2016). “Max Boat Draft” parameter represents the draft dimension of the largest boat. It is assigned as a family instance parameter to piers, finger piers, anchors, and boats, and shown in Figure 3.28. The boat draft is determined as a user-defined parameter for piers. It can be specified by users or via “Placing Families on Piers” dynamo script for other families.

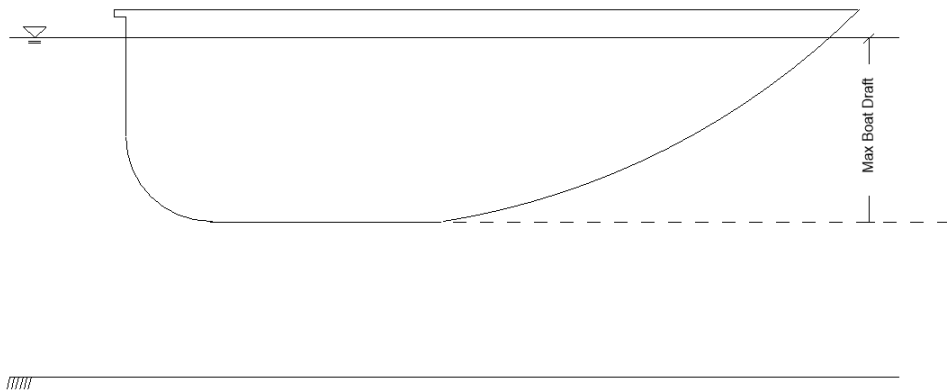


Figure 3.28. Maximum Boat Draft

- Berth Number:** It is a required parameter, while finger piers are used in marina configuration. Two boats or one boat berth between two finger piers.

Therefore, “Berth Number” parameter should be assigned as one if one boat berths between two finger piers or two if two boats berth between two finger piers.

3.2.3.3 FAMILIES

Four families are used to model a marina in this study. Their geometric and parametric features will be explained in this part.

3.2.3.3.1 Pier

“Pier” family is created to represent piers in the marinas. Its shape is a basic rectangular prism. It is modeled by Extrusion tools in the Generic Model Family Template.

Its properties are “Work Plane Based” and “Always Vertical” as seen in Figure 3.29. Its work plane is the “Center (Left/Right)” reference plane. Work-Planned based families move according to their host families. If a family is not associated with any work-plane, they can move independently. Moreover, the families with “Always Vertical” property remain vertical even though the host family has no slope. Also, it has ‘Shared’ properties. When a nested family is loaded into the project, even used in another family, it can be seen in Project Browser and in schedules.

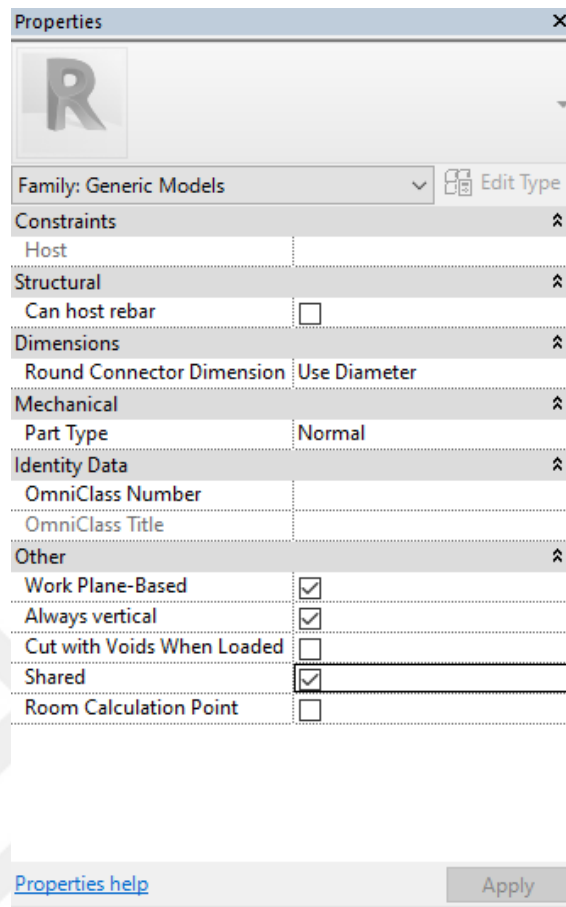


Figure 3.29. Properties of “Pier” Family

The top of the family is fixed to the “Top” reference plane, which is pinned. In other words, the reference plane has a constraint against the movement. The bottom of the family is fixed to the “Bottom” reference plane that can move freely. The intersection of the “Top” reference plane and “Center (Left/Right)” reference plane defines the origin of the family. The sides of the family are fixed to the “Left” and “Right” reference planes. There is an equality constraint between “Left”, “Right” and “Center (Left/Right)” reference plane. The front of the family is also fixed to the “Center (Front/Back)” reference plane that is pinned in place. The back of the family is fixed to “Back” reference plane that can move freely.

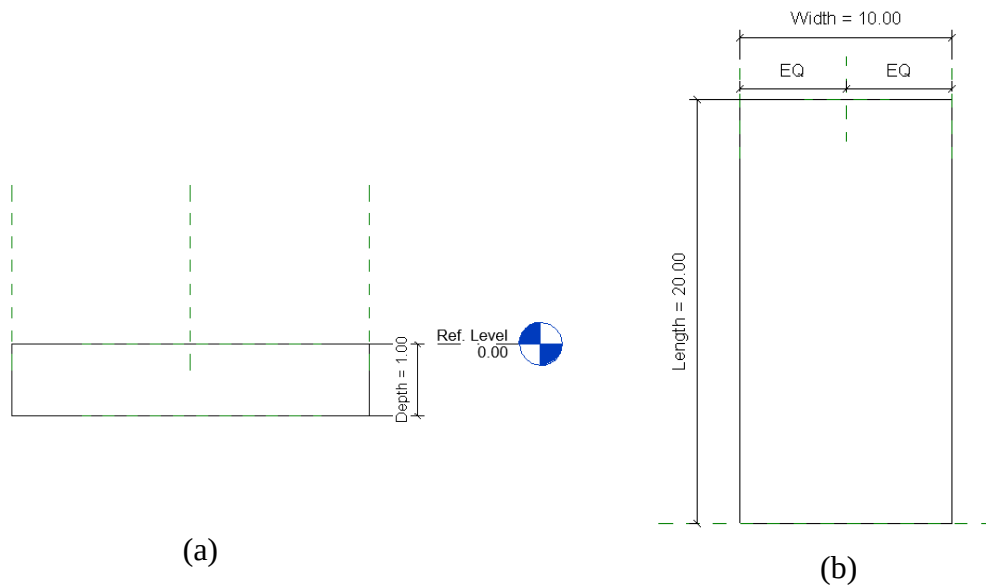


Figure 3.30. Dimension Measurement of “Pier” Family. (a) Depth Measurement (b) Width and Length Measurement

Figure 3.30 displays the parameters of the family that are explained below.

- **Depth:** It specifies the depth of the pier from the water level. The depth is measured between the “Top” reference plane and the “Bottom” reference plane, as illustrated in Figure 3.30 (a). If the parameter's value is changed, the “Bottom” reference plane will move as the family's bottom face will be move accordingly.
- **Width:** It determines the width of the pier. This parameter measures from the “Left” reference plane to the “Right” reference plane. Also, there is a segmented dimension that has equality constrain. This constraint enables to extend of the width without moving the center of the family.
- **Length:** It determines the length of the pier. Figure 3.30 (b) demonstrates that the parameter measures from the “Front” reference plane to the “Back” reference plane. As its value is changed, the front face stays in place, and the end face moves away from the “Front” reference plane.

- **Pier Name:** It is a text parameter that specifies the name of the pier. It is used to distinguish the piers. This information also identifies other elements such as finger piers, boats, or anchoring hosted on the pier.
- **Finger Pier Placement:** It is a “Yes/No” parameter. If the pier hosts the finger piers, the parameter status should be checked, which means “Yes”. If the pier host other elements, the parameter status should be unchecked, that means “No”. It plays a key role in calculations for code compliance checks explained in the following sections in detail.

Besides the family-specific parameter, “Pier” family has “Max Boat Length”, “Max Boat Width”, “Max Boat Draft” and “Berth Number” parameters. All these parameters are explained in Section 3.2.3.2.

Family Types

Type name:

Search parameters

Parameter	Value	Formula	Lock
Constraints			
Default Elevation	0.0000	=	☐
Text			
Pier Name (default)		=	
Dimensions			
Depth (default)	1.0000	=	☐
Length (default)	330.0000	=	☐
Width (default)	10.0000	=	☐
Other			
Berth Number (default)	2	=	☐
Finger Pier Placement (default)	☒	=	
Max Boat Draft (default)	0.0000	=	☐
Max Boat Length (default)	0.0000	=	☐
Max Boat Width (default)	0.0000	=	☐
Identity Data			

Manage Lookup Tables

[How do I manage family types?](#)

Figure 3.31. Parameters of “Pier” Family

Users should define these parameters because later analyses are performed according to the piers and their information. Pier family should be placed with the “Place on Work Plane” command. Then, the level that the family should be placed on should be selected.

3.2.3.3.2 Finger Pier

“Finger Pier” family is created by using the Generic Model family template. The Extrusion tool is used to form a rectangular prism. It has “Work-Plane Based”, “Always Vertical”, and “Shared” properties as seen in Figure 3.32. Its work plane is “Center (Front/Back)” reference plane.

Properties	
Family: Generic Models Edit Type	
Constraints	
Host	
Structural	
Can host rebar	☐
Dimensions	
Round Connector Dimension	Use Diameter
Mechanical	
Part Type	Normal
Identity Data	
OmniClass Number	
OmniClass Title	
Other	
Work Plane-Based	☒
Always vertical	☒
Cut with Voids When Loaded	☐
Shared	☒
Room Calculation Point	☐
Properties help Apply	

Figure 3.32. Properties of “Finger Pier” Family

The top of the family is fixed to the Reference Level, of which elevation is equal to 0. At the same level, there is a reference plane that is called “Top”. It is a pinned plane. In other words, it has a constraint against the movement. The bottom of the family is fixed to the “Bottom” reference plane that can move freely. The intersection of the “Top” reference plane and the “Center (Left/Right)” reference plane defines the origin of the family. Figure 3.33 (a) shows the “Reference Level” view. The family’s left side is fixed to the “Left” reference plane, the family’s right side is fixed to the “Right” reference plane. The front of the family is also fixed to the “Front” reference planes pinned in place. All of these constraints enables to change dimensions of the family by moving these reference planes.

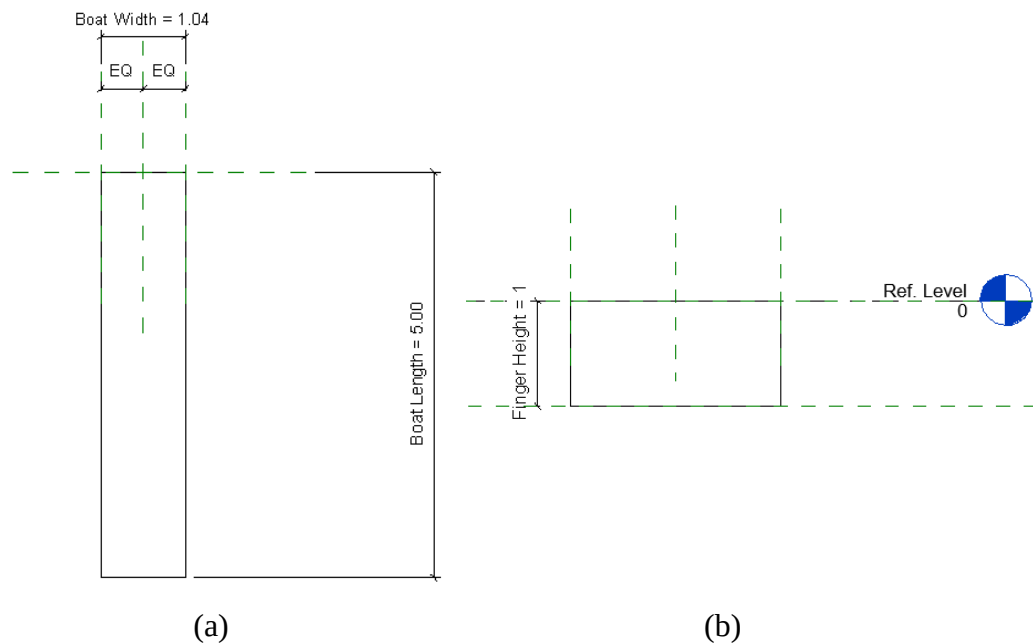


Figure 3.33. Dimensions of “Finger Pier” Family (a) Length and Width Measurements (b) Height Measurement

Figure 3.34 displays the parameters of the family that are described below.

- **Finger Height:** It specifies the height of the finger pier. “Pier Height” measures between the “Top” reference plane and the “Bottom” reference plane as illustrated in Figure 3.33 (b). If the parameter's value is changed, the

“Bottom” reference plane moves as the family's bottom face moves accordingly. The user should define this parameter.

- **Finger Width:** It determines the width of the finger pier. This parameter measures from the “Left” reference plane to the “Right” reference plane. Also, there is a segmented dimension that has equality constrain. This constraint enables to extend of the width without moving the center of the family. The user should define this parameter.
- **Finger Length:** It determines the length of the family. Figure 3.33 (a) demonstrates that the parameter measures from the “Front” reference plane to the end of the family. As its value is changed, the front face stays in place, and the end face moves away from the “Front” reference plane. The user can define this parameter through its dependent parameter, or it can be defined as the end result of “Placing Families on Piers” dynamo script.
- **Berthing Length:** Berthing length is the required distance for safe maneuver for boats during berthing and leaving. It is defined by a formula for the finger piers in Coastal Structures Planning and Design Technical Manual (AYGM, 2016). It is explained in “3.2.4 RULE CHECKS” section. For the finger piers case, the berthing length is equal to the length of the finger pier. The user can define this parameter. It can also be specified as a result of “Placing Families on Piers” dynamo script.

Besides the family-specific parameter, “Finger Pier” family has “Host Pier”, “R/L”, “Max Boat Length”, “Max Boat Width”, “Max Boat Draft” and “Berth Number” parameters. All these parameters are explained in Section 3.2.3.2.

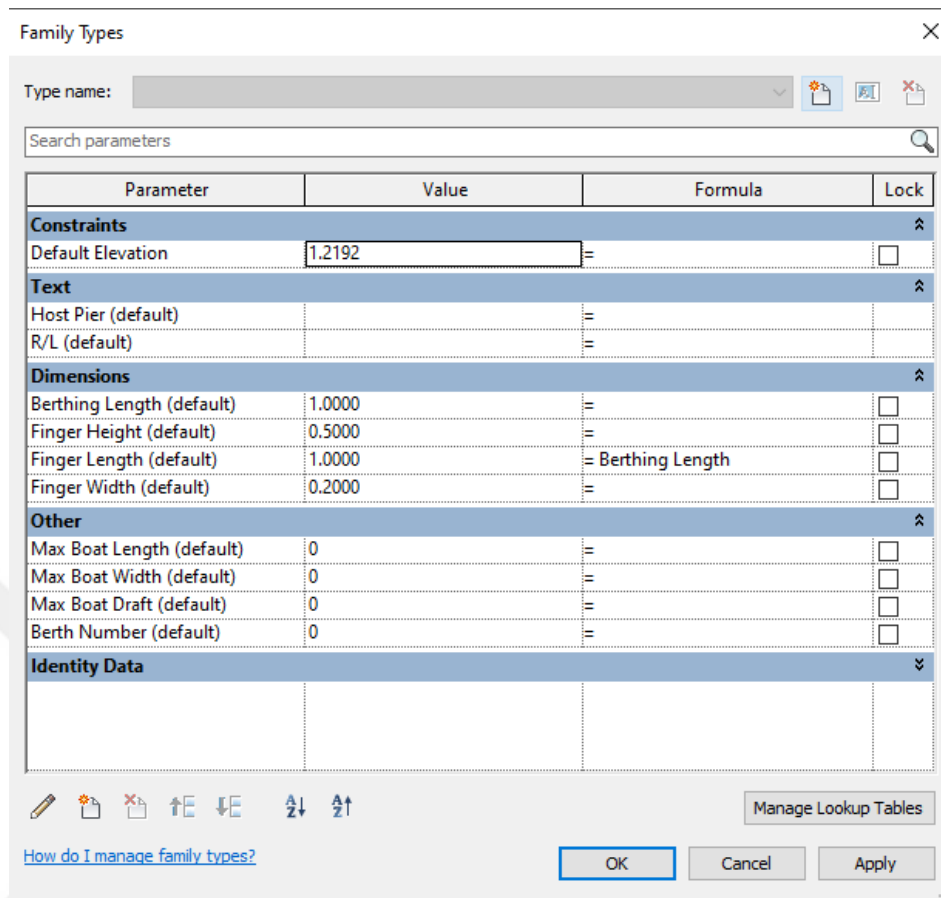


Figure 3.34. Parameters of “Finger Pier” Family

3.2.3.3.3 Anchor

Anchors are mainly used in Mediterranean mooring. The true representation of anchoring is not concerned in this study. Therefore, it is represented by a basic prism. Also, it has not any parameters related to its geometry. Like other families, it is created by the Extrusion tool in the Generic Model family template.

The top of the family is fixed to the “Reference Level” of which elevation is equal to 0. The front face of the family is fixed to the “Front” reference level. The family's work plane is ‘Center (Front/Back)’ reference plane. It has “Work Plane-Based”, “Always Vertical” and “Shared” properties. Figure 3.35 displays the parameters of the family.

Family Types

Type name:

Search parameters

Parameter	Value	Formula	Lock
Constraints			
Default Elevation	1.2192	=	☐
Text			
Host Pier (default)		=	
R/L (default)		=	
Dimensions			
Berthing Length (default)	1.0000	=	☐
Other			
Max Boat Draft (default)	0.0000	=	☐
Max Boat Length (default)	0.0000	=	☐
Max Boat Width (default)	0.0000	=	☐
Identity Data			

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Manage Lookup Tables

[How do I manage family types?](#)

OK Cancel Apply

Figure 3.35. Parameters of “Anchor” Family

- Berthing Length:** As explained in the Finger Pier family, berthing length is the distance to provide safe maneuver for boats between the anchor and pier. It is measured between the “Front” reference plane and the “Center (Front/Back)” reference plane, as seen in Figure 3.36. The “Center (Front/Back)” reference plane is pinned in place. Therefore, as the parameter is changing, the “Front” plane and the anchoring is moving.

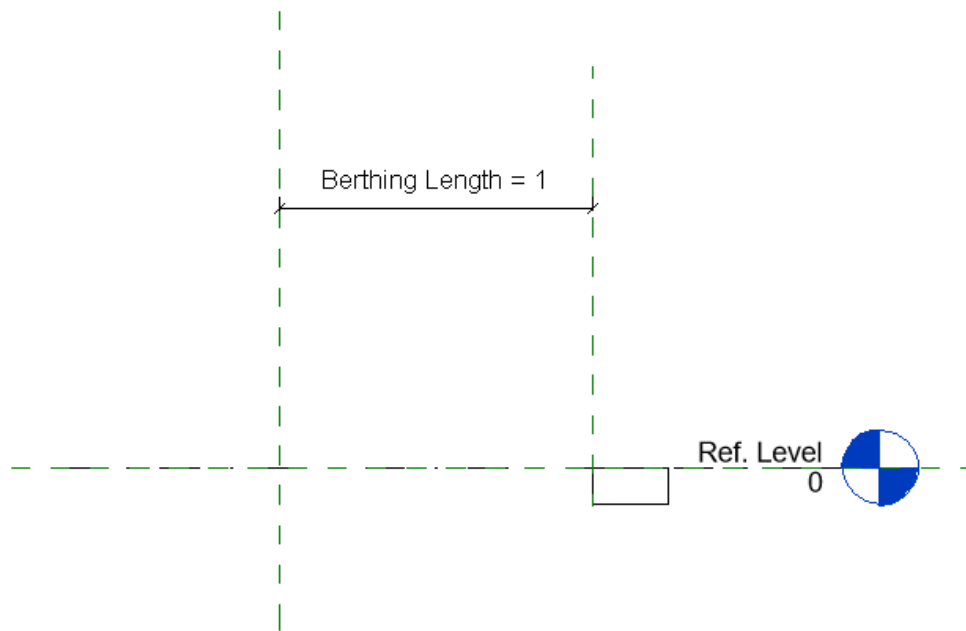


Figure 3.36. Berthing Length Measurement for “Anchor” Family

“Berthing Length” parameter can be defined by users as input or can be defined by as the results of “Placing Families on Piers” dynamo script.

Besides the family-specific parameter, “Anchor” family has “Max Boat Length”, “Max Boat Width”, and “Max Boat Draft” parameters. All these parameters are explained in Section 3.2.3.2.

3.2.3.3.4 Boat

“Boat” Revit family is created by using the Generic Model family template. It represents a boat that docking perpendicular to the berthing spaces. It has a basic rectangular prism shape and is modeled by using the Extrusion tool.

The Properties window in Figure 3.37 shows that it is a work-planned-based family. The work plane of the Boat family is the “Front” plane.

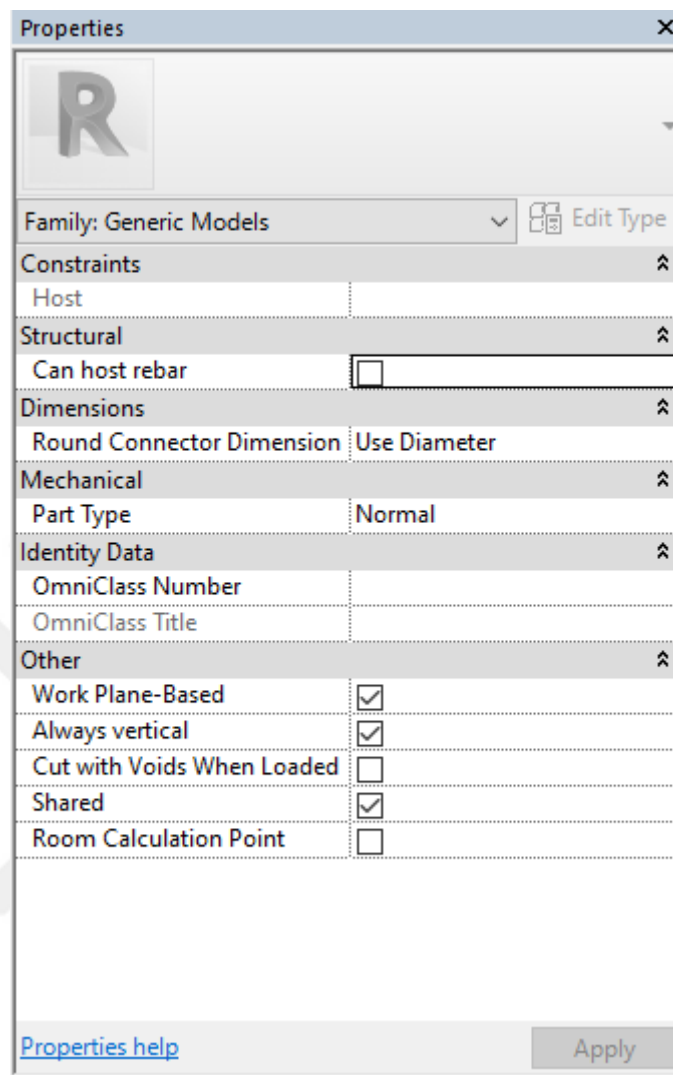


Figure 3.37. Properties of “Boat Family

The families with “Always Vertical” property remain vertical even though the host family has no slope. Also, it has ‘Shared’ properties.

Family Types

Type name:

Search parameters

Parameter	Value	Formula	Lock
Constraints			
Default Elevation	1.2192	=	☐
Text			
Host Pier (default)		=	
R/L (default)		=	
Materials and Finishes			
Pavement Material	<By Category>	=	
Dimensions			
Boat Draft (default)	0.4000	= Max Boat Draft	☐
Boat Height (default)	1.4000	= Max Boat Draft + 1 m	☐
Boat Length (default)	5.0000	= Max Boat Length	☐
Boat Width (default)	1.0385	= Max Boat Width	☐
Max Boat Draft (default)	0.4000	=	☐
Max Boat Length (default)	5.0000	=	☐
Max Boat Width (default)	1.0385	=	☐
Identity Data			

[How do I manage family types?](#)

Figure 3.38. Parameters of “Boat” Family

Figure 3.38 shows that it has four different family parameters that define its dimensions.

- **Boat Draft:** Draft is defined as a vertical dimension of a boat in Family Parameter Part. Thus, this parameter represents the draft dimension of the boat. Also, it is an instance family parameter. It is dependent on “Max Boat Draft” parameter.
- **Boat Height:** It is an instance parameter that represents the overall height of the boat. It includes draft and height above the water. It depends on a formula that includes “Boat Draft” parameter value and the distance above the water level. The user can define the distance above the water level.

- **Boat Length:** It is an instance parameter that represents the largest dimension from head to stern. It is dependent on to “Max Boat Length” parameter.
- **Boat Width:** It is an instance parameter that represents the width of the boat. It is dependent on the “Max Boat Width” parameter.

Besides the family-specific parameter, the “Boat” family has “Max Boat Length”, “Max Boat Width”, “Max Boat Draft”, “Host Pier” and “R/L” parameters. All these parameters are explained in Section 3.2.3.2.

Boat Draft, Boat Height, Boat Length, and Boat Width are dependent on other parameters because the boats modeled in the project are the largest boats that use the berthing places. Therefore, the dimensions should correspond to the largest boat dimension assigned in the piers or the berthing places. They can be defined through parameters they depend on by users when code compliance checking is performed. They can also be calculated by “Placing Families on Piers” dynamo script when model generation is performed.

Figure 3.39 (a) shows how the length and width dimensions are measured. As shown in Figure 3.39 (a), there are two reference lines whose intersection defines the family's origin. While “Boat” family is placed in the project, that intersection acts like the family's placement point. A segmented dimension measures the distance from the left and the right sides to the reference line. It has equality constraints. It means that if the width dimension is changed, the family's left and the right sides will be enlarged equally from the reference line. Thus, the origin of the family will not move. Moreover, draft and height dimensions are shown in Figure 3.39 (b).

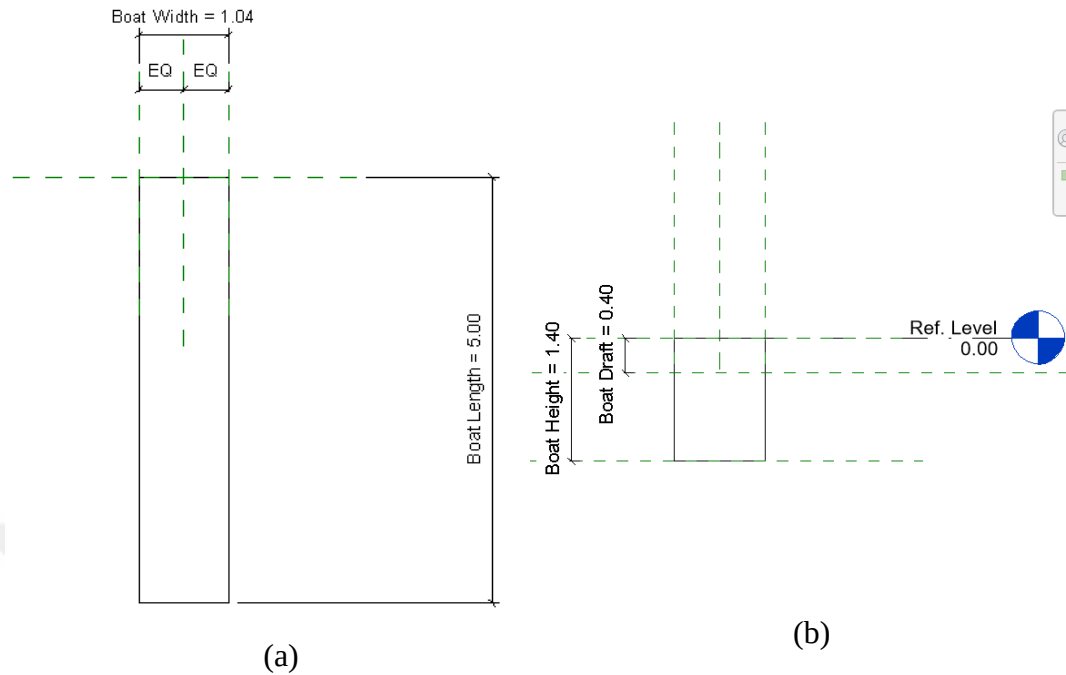


Figure 3.39. Dimensions of “Boat” Family. (a) Length and Width Dimensions (b) Draft and Height Dimensions

When placing in a model manually, it should be placed in a section view or elevation view to be placed perpendicular to the host element.

3.2.4 RULE CHECKS

Dynamo tool is used to perform code compliance checks in this study. Dynamo is “A visual programming tool that aims to be accessible to both non-programmers and programmers alike. It gives users the ability to visually script behavior, define custom pieces of logic, and script using various textual programming languages.” (What Is Dynamo?, n.d.). It can be used as stand-alone software or as add-in for Revit and other software. In this study, it is used as add-in for Revit. The main idea in Dynamo tool is to connect pre-defined packaged nodes for performing tasks. The nodes are connected to define relationship and actions. At the end, an algorithm can be created without using any Programming Language. On the other hand, the

customized nodes created by Programming Languages such as Python or C can be used alongside the pre-packaged nodes.

In this study, the pre-packaged nodes and the customized nodes containing Python scripts are used for code compliance checking. Dynamo scripts that are created for the code compliance checks are explained in this section. Coastal Structures Planning and Design Technical Manual (AYGM, 2016) is taken as the guidelines for these checks.

3.2.4.1 Boat Types and Dimensions

In this part, the boat types and dimensions are introduced briefly. There are two main classifications for boats: motorboats and sailing boats. The general dimensional characteristics of recreational boats are given in Table 3.2.

Table 3.2 Typical Design Parameters of The Boats

Length (m)	Draft (D) (m)		Width (B) (m)	
	Motorboats	Sailing Boats	Motorboats	Sailing Boats
0-5	0.80	1.40	2.20	1.80
5-9	1.00	2.00	3.60	3.00
9-12	1.20	2.40	4.10	3.40
12-15	1.04	2.08	4.80	3.90
15-20	1.66	3.40	5.30	4.40

Reprinted from AYGM (2016)

In this study, the dimensions of the motorboats are considered during rule check processes because their beam dimensions are larger than the sailing boats. Also, the lengths of the boat are given as a range in “Length” column in Table 3.2. The largest dimensions on the ranges are taken into consideration in order to be on the safe side.

On the other hand, the boat lengths are continuously increasing due to the developments in marine engineering. Therefore, new terms are emerged, such as

mega yachts and superyachts. The boats whose lengths are 24 m to 36 m are mentioned as mega yachts, and the boats whose lengths are larger than 36 m are considered superyachts in Coastal Structures Planning and Design Technical Manual (AYGM, 2016). Moreover, it is stated that some specially crafted yachts are longer than 70 m are called gigayachts. Table 3.3 gives information about the dimensions of these super yachts.

Table 3.3 Dimensional Characteristics of Super Yachts

LOA (m)	Mean Draft (D) (m)	Mean Width (B) (m)
25-35	2.5	6.8
35-45	2.8	8.3
45-55	3.0	9.6
55-65	3.2	10.2
65-75	3.4	10.6
75-85	3.7	12.2
LOA = Length Overall		

Adapted from PIANC (2013) cited in AYGM (2016)

The characteristics of the super yachts require special considerations during marina design. They will be mentioned in the following chapters in detail.

3.2.4.2 Slip (Berth) Width

The slip widths are depended on the boat width in general. Berth number is an important factor for the cases that finger piers are used. Berth number represents the number of boats berthing between two finger piers. There are two cases: single slip and double slip.

The required minimum distances between two finger piers are calculated according to these formulas shown below.

For single slip: $W_1 = B_{\max} + 0.6\text{m}$

For double slip: $W_2 = 2 * B_{\max} + (1.5 \sim 2.0\text{m})$

W: Distance between finger piers

B_{max} : The width of the longest boat berthing to the pier

For double slip, $W_2 = 2*B_{maks}+2.0m$ is executed to be on the safe side in this study.

The formula to calculate the required minimum distances between two anchors is:

$$W_3 = B_{max}+(1.5\sim 2.0m)$$

The additional dimension in the formula is taken as 2.0 m to be on the safe side.

Figure 3.40 represents the clearance between two berthing places in an example layout.

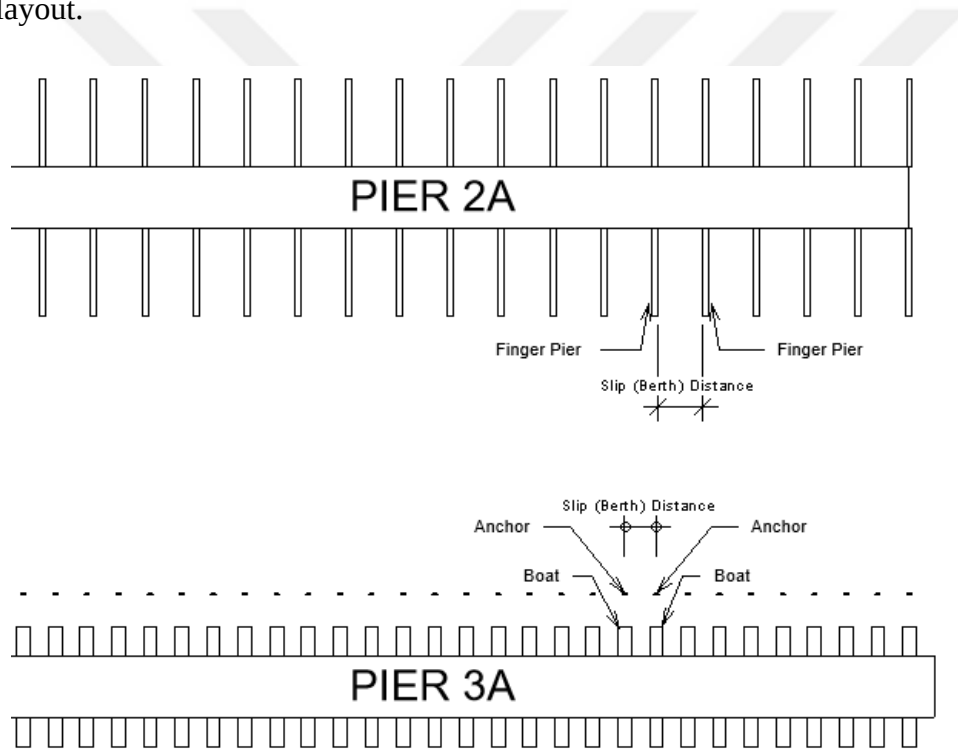


Figure 3.40. Slip (Berth) Width Dimension Representation in a Layout. The upper Part (Pier 2A) is Finger Pier Arrangement. The lower Part (Pier 3A) is Mediterranean Arrangement.

In super yacht cases, the minimum and preferred slip (berth) width dimensions are given in Table 3.4. The preferred distances are taken into consideration for

controlling slip width. Besides Table 3.4, the calculations of the slip width are given in PIANC (2013) cited by AYGM (2016) as:

$$W_{\text{single}} = 2(mc \cdot f_d) + B \quad W_{\text{double}} = 4(mc \cdot f_d) + 2B$$

mc : maneuverability coefficient, ranging from 1.0 for minimum conditions to 1.5 for recommended conditions

f_d : fender diameter (meters) for specified superyacht length as given in Table 3.5

B : boat width (m)

Table 3.4 Minimum and Preferred Slip (Berth) Width Dimensions for Super Yachts

L _{berth} (m)	W _{double} (m)		W _{single} (m)	
	Minimum	Preferred	Minimum	Preferred
25	16.5	18.0	8.5	9.0
30	17.5	19.0	9.0	9.5
35	18.5	20.0	9.5	10.0
40	20.5	22.0	10.5	11.0
45	22.5	24.0	11.5	12.0
50	26.0	27.5	13.0	14.0
55	27.0	28.5	13.5	14.5
60	28.0	29.5	14.0	15.0
65	30.0	32.5	15.0	16.5
70	31.0	33.5	15.5	17.0
75	33.0	35.5	16.5	18.0
80	36.0	39.0	18.0	19.5
85	37.0	40.0	18.5	20.0
90	38.0	41.0	19.0	20.5
95	40.0	43.0	20.0	21.5
100	41.0	44.0	20.5	22.0

Reprinted from PIANC(2013) as cited in AYGM (2016)

Table 3.5 Fender Diameter According to Boat LOA

LOA (m)	Fender Diameter (m)
25-45	0.61
45-60	0.91
61-75	1.22
76-91+	1.52

Reprinted from PIANC(2013) as cited in AYGM (2016)

Additionally, the slip dimensions should be designed according to the single slip calculation in Mediterranean arrangements for superyachts (PIANC, 2013). In this study, the preferred slip distance information is used for superyachts.

Measuring and checking the distances between berthing places are performed by the dynamo script called “Slip (Berth) Dimensions (W)”. The developed dynamo script is shown in Figure A.1. Moreover, Figure 3.41 illustrates the working logic of the dynamo script.

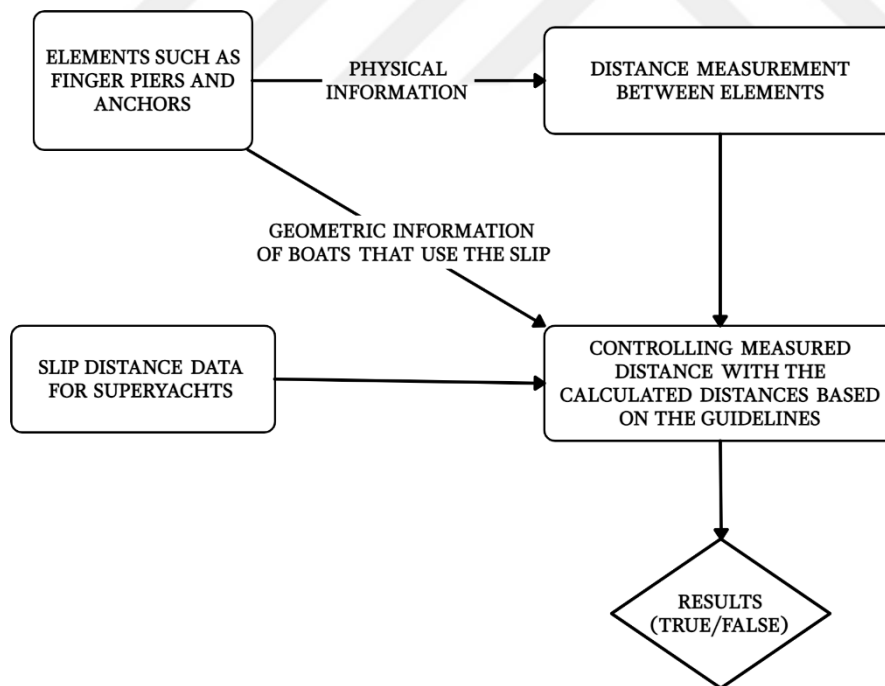


Figure 3.41. Slip (Berth) Dimension Dynamo Script Logic

In this script, the first step is to select all finger piers and anchors in the model. The python code of the selection process can be seen in Figure A.2.

First of all, all generic models are listed in the “family collector” list. Then, the families with the “Host Pier” parameter are looked for. After identifying finger piers and/or anchors, they are grouped according to their “Host Pier” parameter values and “R/L” parameter values.

Secondly, the distances between the finger piers and/or anchors are measured. The piers in each list are evaluated within themselves. “Measuring distance” node measures the distance between a finger pier/anchor itself and the finger pier/anchor next to it. In other words, the measurement is performed between Finger Pier 1 and Finger Pier 2, as shown in Figure 3.40. Then, it continues by measuring the distance between Finger Pier 2 and the next one. Since there is no next one for the last items in the lists, for equating the number of rows of output lists to the number of rows of input lists, 0 is added to the ends of the lists. The python code of the measuring process is given in Figure A.3.

On the other hand, slip width dimension information for superyachts are obtained from the node group of “Slip Dimension Data for Superyachts”. The data is imported from the excel file “Superyacht Slip Distance”. It contains the preferred slip distances given by AYGM (2016).

Then, the list of elements, lists of measurements are inputs of the python code of “Controlling” node shown in Figure A.4. In this code, the measurements are compared to the minimum required distances. The required distance is calculated according to values in “Berth Number” and “Max Boat Width” parameters that are assigned to the Finger Pier family. It is also performed for the Anchor family by using the “Max Boat Width” parameter. The calculation and controlling process is performed for superyachts larger than 20 m and the others separately. The results of comparisons are given as “True/False” for each element. The node gives four lists as a result: List of elements, list of measured distances, list of required minimum

distances, and list of “True/False” results. In the end, all results are exported to an Excel sheet.

3.2.4.3 Inner Channel Width (B₁)

Inner channel width should be larger than ($1.5 \cdot L$) and minimum 20 m. L represents the length of the longest boat in the marina.

Inner channel width is checked by the dynamo scripts called “Inner Channel Width (B₁)” and “Inner Channel Width (Combined)”. Since the inner channel area could not be assigned in the model, the elements that limit the area have to be selected by the user. In the example model, the inner channel is considered as the area between breakwater and piers. It is represented as the red hatch area in Figure 3.42.

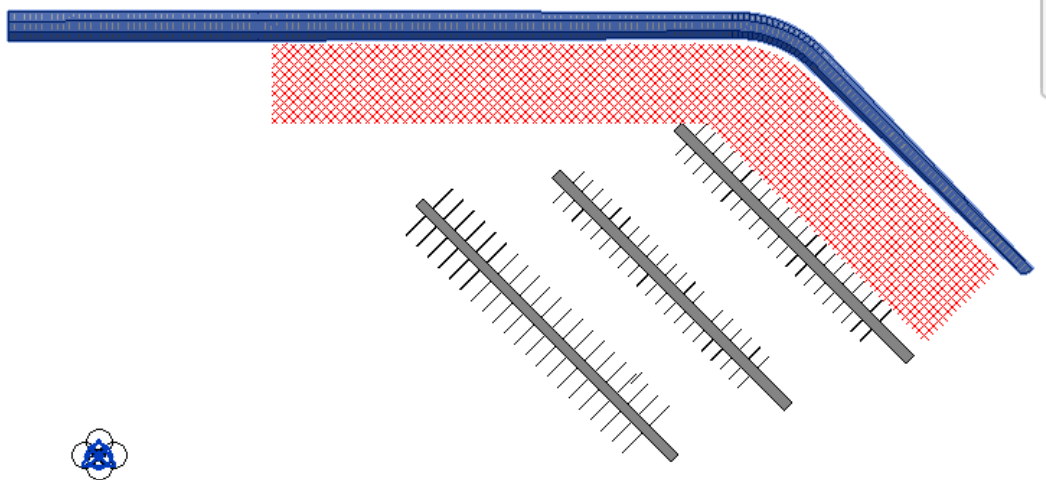


Figure 3.42. Inner Channel Representation

There are two separate dynamo scripts for Finger/Mediterranean Arrangement and Combined Arrangement.

3.2.4.3.1 Finger Pier Arrangement and Mediterranean Arrangement

For these layouts, “Inner Channel Width (B1)” dynamo script can be used. The working principle and the structure of this dynamo script exists in the “Inner Channel Width (Combined)” dynamo script in Appendix B. The working logic of the dynamo script is displayed in Figure 3.43.

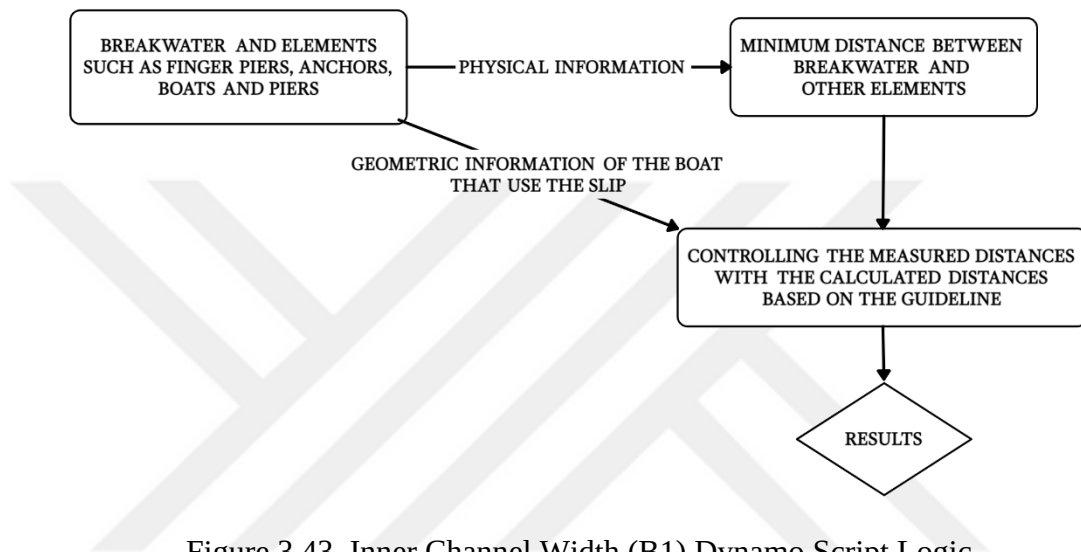


Figure 3.43. Inner Channel Width (B1) Dynamo Script Logic

In this dynamo script, the breakwater is selected separately in one node. All other elements are selected together in another node. Then, their geometries are obtained. The closest points of piers to the breakwater and vice versa are determined. Lines are created between these points, and the lengths of the lines are listed. “Inner Channel Width (B1)” dynamo script is shown in Figure B.1.

In the end, the piers and the length are used as inputs to control the inner channel distance in the “Checking” node. If the “Boat Length” parameter values of all piers are obtained in this node, the maximum value in the list is determined. The lengths of the created lines between closest piers are checked whether they are greater than 20 m. If they are greater than 20 m, it is checked whether they satisfied $(1.5 \cdot L)$ m distance requirement. If the measured distance satisfied the $(1.5 \cdot L)$ m requirement, the element that the measurement is obtained from is put into a list called “t”. If it

did not satisfy any of the requirements, the element is put into a listed “F1”. If the measurement did not satisfy even the 20 m requirement, it is put into “F2” list. Figure B.3 illustrates the python code that is used in the “Checking” node.

In the end, F1 and F2 lists are reported in excel with the column headings “Not satisfied ‘1.5L’ criteria” and “Not satisfied min. 20m criteria”, respectively. Reporting steps of dynamo script are given in Figure B.1(c).

3.2.4.3.2 Combined Layout

“Inner Channel Width (Combined)” dynamo script contains two different code checks for two different inner channels. The first code check is the same as in finger pier and Mediterranean arrangement cases. It is used for the Inner Channel-1, shown in Figure 3.44. The second code check is performed for the Inner Channel-2 in Figure 3.44 that is between two piers.

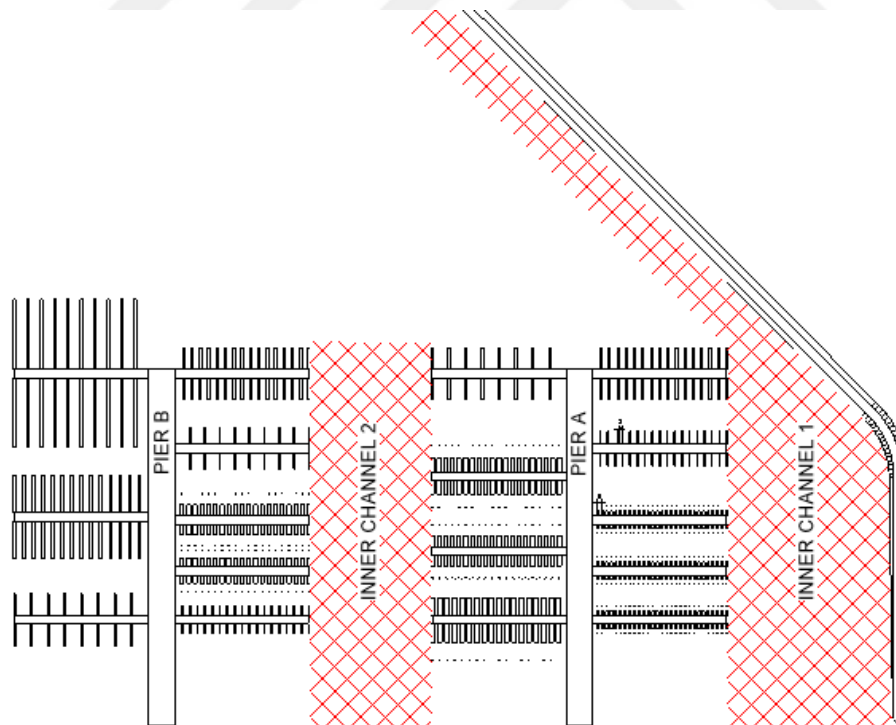


Figure 3.44. Inner Channels For Combined Layout

The simple logic of the dynamo script used for Inner Channel-2 is given in Figure 3.45.

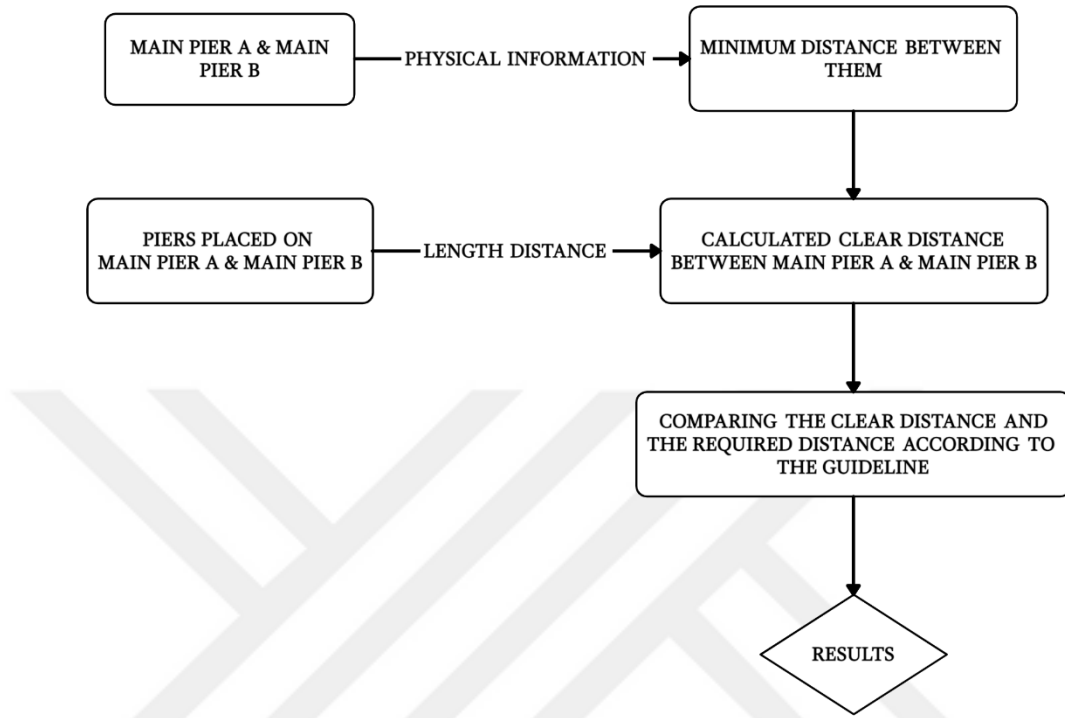


Figure 3.45. “Inner Channel Width (Combined)” Dynamo Script Logic for Inner Channel-2 Case

For the “Inner Channel 2” case, Pier A and Pier B are selected by two separate nodes. The dynamo nodes measure the distance between them. Then, the piers on them are collected by the “Selection By Name” node. The lengths of the longest piers on each pier and the biggest value of the “Max Boat Length” parameter are obtained. The lengths are used to calculate the clear distance between the two piers. The parameter values are used to calculate the required minimum distance. In the end, the clear distance and the required minimum distance are compared to decide whether the inner channel provides a safe passing distance for the boats. According to the comparison, the “Controlling” node gives three different warnings: (1) Inner channel is smaller than 20 m. Not enough distance! (2) Inner channel is smaller than 1.5L

criteria. Not enough distance! (3) Inner channel has enough distance. Figure B.2 gives “Inner Channel Width (Combined)” dynamo script.

3.2.4.4 Distance Between Piers (S)

Distance between two piers is annotated as “S”. The representation of “S” distance is displayed in Figure 3.46. This distance should be larger than $(1.5 \sim 2.0) L$. L represents the length of the largest boat among the boats berth at two piers. In other words, the largest boats allowed to berth at Pier 1 and Pier 2 should be known separately. Then, the lengths of these two boats are compared, and the length of the longest one is used to determine “S”.

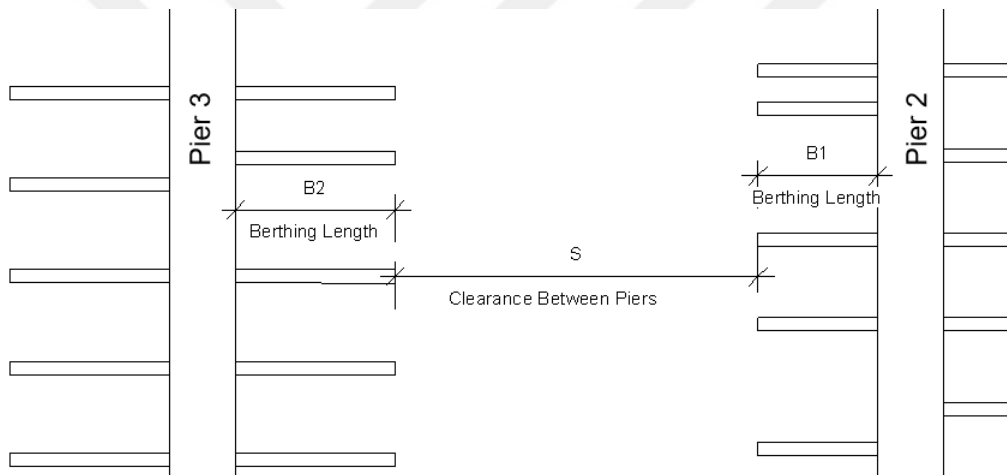


Figure 3.46. Representation of Clearance Between Piers

In this study, $2.0L$ should be used as the minimum distance to be on the safe side. However, in real practice, the coefficient value can be down to 1. Therefore, the dynamo code is arranged in such a way that the user can specify this coefficient.

“S” distance could not be measured in this study. Therefore, the total distance between the two piers is measured, as shown in Figure 3.46. The required distance between the two piers is calculated as the sum of berthing distances (B) of both piers and “S” distance. The comparison is performed between this calculated distance and the measured distance.

The controlling process has some differences for the combined marina layout. Therefore, two different dynamo scripts are prepared for this process.

3.2.4.4.1 Finger Pier Arrangement and Mediterranean Arrangement

Distances between piers are checked for finger pier and Mediterranean layouts by dynamo script called “Distance between Two Piers (S)”. The dynamo script structure is displayed in Figure C.1. On the other hand, working logic of this dynamo script is shown in Figure 3.47.

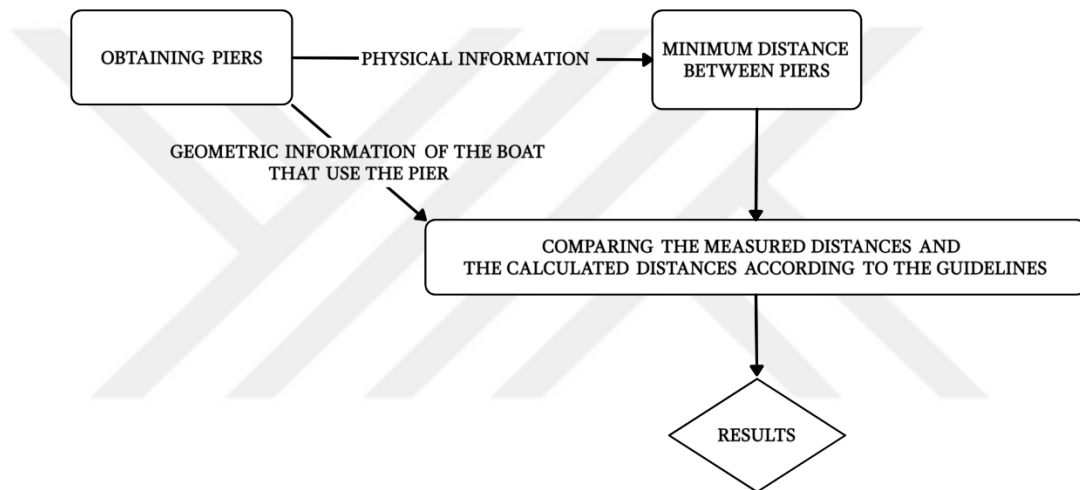


Figure 3.47. Distance between Pier Dynamo Script Working Logic

Firstly, the piers are selected by “Selection By Name” node. The python code of “Selection By Name” node is shown in Figure C.2. All generic models are collected in this node, and “Pier” word is searched through their type names. If it contains “Pier”, it will be added to the list of “Piers”. Secondly, geometries of the pier bodies and distances between them are obtained by “Measuring Distance” node that contains the python code in Figure C.3.

Then, the list of elements and list of measurements are inputs of the “Controlling” node. The values of “Max Boat Length” of a pier and next to it are obtained (L_1 and L_2). These values are used to calculate berthing lengths (BL_1 and BL_2). After that,

the maximum one of the boat lengths is used to calculate “S”, called the required distances between piers except for berthing lengths. The berthing lengths specific to each pier (BL_1 and BL_2) and the “S” value are summed up to find total distances between the two piers. The python code in “Controlling” node is shown in Figure C.4.

This process is performed for all piers in the “Piers” list. piers are listed in the ascending order of their element IDs in the “Piers” list. Therefore, the IDs of the main piers next to each other should be in ascending order to perform the check correctly. On the other hand, Revit's element IDs are assigned ascendingly as they are placed in the model. As a result, after a main pier is placed, the next pier that must be placed should be the closest one. If the example layout in Figure 3.48 is taken as a reference, “Pier 1” should be placed first. Then, “Pier 2” should be placed. Lastly, “Pier 3” should be placed. According to this placement order, Pier 1 and Pier 2 are controlled by Dynamo script at first. Then, Pier 2 and Pier 3 are controlled.

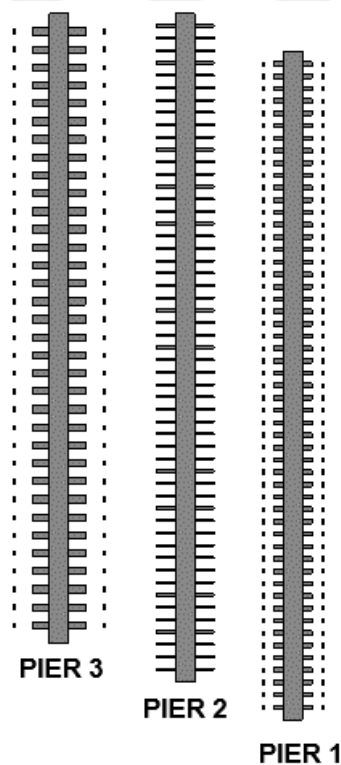


Figure 3.48. Pier Placement Layout

The next step after the measurement is that the measured distances are compared with the calculated requirement distance. There are four results of “Controlling” Node: Elements, Measured Distances, Required Minimum Distances, and True/False statement according to the comparison. Besides these, element IDs of piers are obtained by the “Element.Id” node. All of the information is combined in an excel file.

3.2.4.4.2 Combined Layout

Distance between piers is checked for combined layout by dynamo script called “Distance between Two Piers (S) (Combined)”. The general structure and the code's logic are the same, as shown in Figure C.1 and Figure 3.47 respectively.

In combined layout, piers are placed on other families are called “Main Pier”, as seen in Figure 3.49. The “Pier Name” values of the main piers are named as “A” and “B”. The piers are grouped according to these values in “Selection By Name” node that contains python code given in Figure D.1.

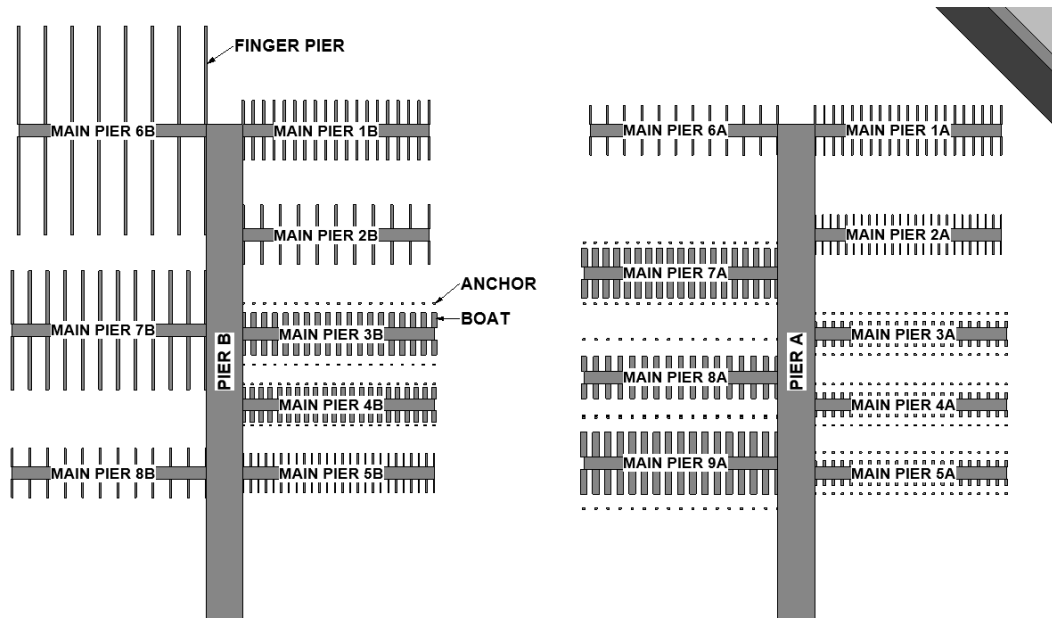


Figure 3.49. Combined Layout Plan

The measurement and the controlling process is performed through “A” and “B” groups. The measurement is performed by “Measuring Distances” node. Its python code is given in Figure D.2. Also, as mentioned before, the piers are evaluated according to their element IDs' ascending order. In the combined layout case, the “Pier 6A” is placed in the model after the “Pier 5A”. They come after another in the list of measurement and control. Therefore, the “Measuring Distance” node measures the distance between “Pier 5A” and “Pier 6A”. Then, the “Controlling” node controls that measurement according to the guideline. This situation is also valid for “Pier 5B” and “Pier 6B”. The designer or the controller should be aware of this operation logic and evaluate the results accordingly.

The python code in “Controlling” node for the combined layout is displayed in Figure D.3. Even though the controlling process's main idea is the same as the one in Figure C.4, there are some differences in the code. In the end, the result list contains sublists that are created based on "A" and "B" groups for this example layout. This is the reason for the difference in the code. The reporting process is the same as “Finger and Anchor Layout” case.

3.2.4.5 Water Level Check

Water level check is performed by dynamo script called “Water Level Check” shown in Figure E.1. The working logic of the dynamo script is also given in Figure 3.50. In this dynamo script, the bathymetry model and other elements in the model are used.

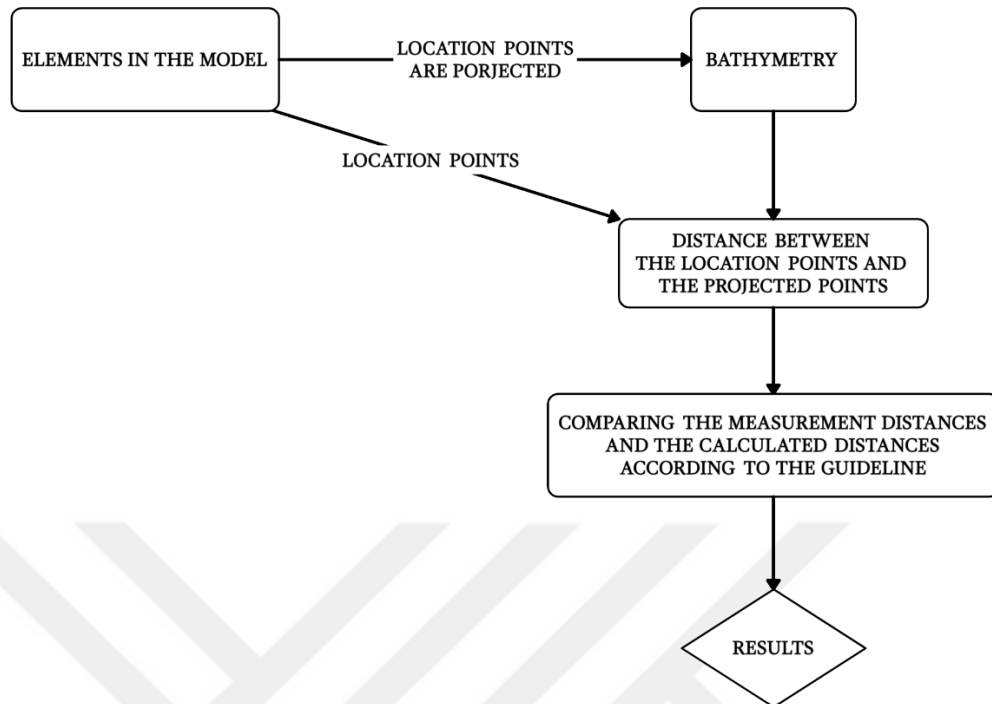


Figure 3.50. Water Level Check Dynamo Script Working Logic

When the bathymetry model is too large, it is in the possibility that the code running process can crash. Thus, and to shorten the run time, it is advised that a subregion that limits the marina area should be created. Then, this limited region should be selected as the bathymetry model.

After the selection of bathymetry and elements, location points of elements are obtained and projected to bathymetry. Then, lines are created between the location points of the elements and the projected points. The lengths of these lines are checked in the “Controlling” node. This node contains the python code given in Figure E.3. In this node, values of the “Max Boat Draft” parameters assigned to the elements are obtained. The “Underkeel Clearance” distance is stated by the user and taken as input to the node to calculate the minimum water level. After that, the lengths of the lines are checked to whether they satisfy the required minimum water level. If they satisfy, they are put into the “true” list. If not, they are put into the ‘false’ list. Moreover, the bathymetry model points that do not satisfy the required minimum water level are reported in a “lowest_point” list.

3.2.4.6 Berth Length (B)

Berth length depends on the length of the longest boat which will use the berthing place. Finger pier and anchoring type (Mediterranean type) berthing arrangements are in the scope of this study. The formulation of berth lengths according to arrangement types are given in below:

- **Finger Pier:** $B_1 = (0.7 \sim 1.2) L$
- **Anchoring:** $B_2 = (1.5 \sim 2.0) L$

L represents the length of the longest boat that berths to the place. In both cases, the largest coefficients are used to be on the safe side.

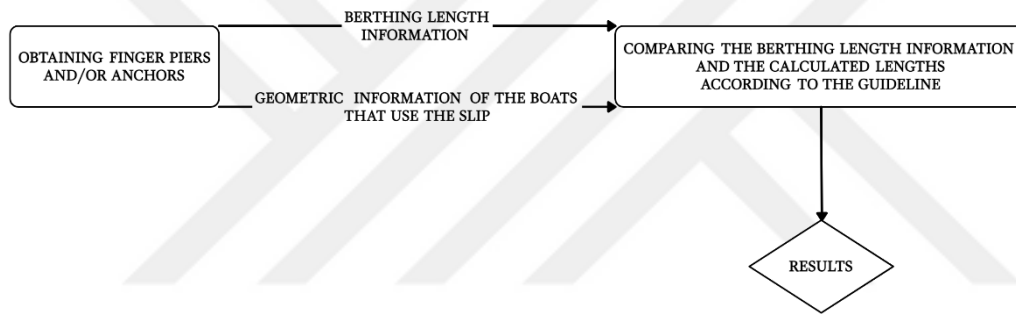


Figure 3.51. “Berthing Length (B)” Dynamo Script Working Logic

The berth lengths are checked by dynamo script called “Berthing Length (B)”. The code structure of the dynamo script is given in Figure F.1. The script’s working logic is illustrated in Figure 3.51. After generic families in the model are collected in this dynamo script, their names are checked whether it contains “Finger” or “Anchor”. According to their names, they are listed separately under “Finger” or “Anchor” lists. Then, the lengths of the fingers and anchoring are obtained from the “Berthing Length” parameter. The berthing length values are compared to the calculated minimum length with the help of the “Max Boat Length” parameter values. If the lengths satisfy the condition, their element IDs are listed under the “true”. Otherwise, their element IDs are listed under the “false”. All these processes are performed by “Checking Berthing Length” node. It contains the python code shown in Figure F.2.

3.2.5 PLACING FAMILIES ON PIERS

“Placing Families on Piers” dynamo script automates the placement of finger piers, anchors, and boats. The working logic of the script is illustrated in Figure 3.52. The main purpose of this dynamo script is initially to create a script that helps model marina layout alternatives for the case studies. This need resulted in an opportunity to generate different marina layout alternatives automatically. In other words, this script enables the designers to generate design alternatives automatically to reach optimum design in an effective way. On the other hand, Tobiasson and Kollmeyer (1991) mentioned that the conflict between the developer and the designer during the design stage. The main purpose of the developer is to increase number of the berthing places in order to increase the capital costs. The designer aims to provide user-friendly layout for safe maneuver and safe berthing. It is suggested that the designer should be meet the needs of the developer, the boat owners and the regulatory agencies. Additionally, Tobiasson and Kollmeyer (1991) mentioned a marina layout example in real-life. In this case, the designer only considered the expectations of the developer who does not aware of the needs and want a dense marina to earn more income. In result, the final layout did not provide enough space for maneuver and it jeopardized the safety of the boat owners. In other words, it is stated that poorly designed marina layouts cause the operational and the safety problems in marinas (Tobiasson and Kollmeyer, 1991).

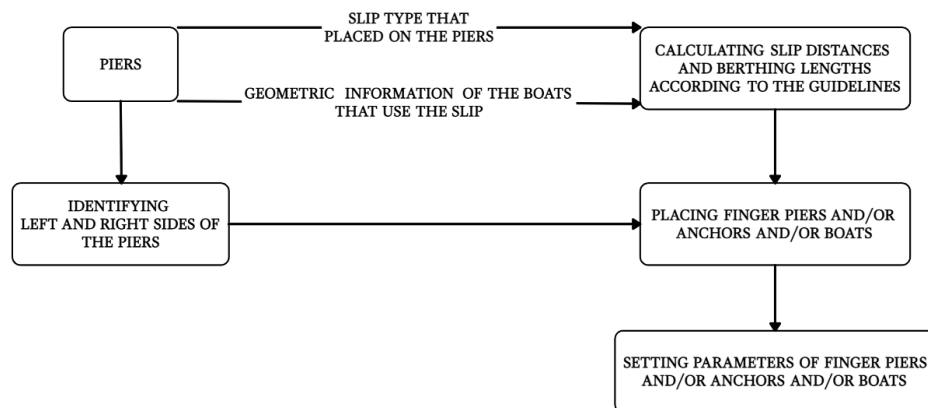


Figure 3.52. The Working Logic of “Placing Families on Piers”

Before running the script, the piers are placed in the model. Also, the values of parameters such as “Max Boat Length”, “Max Boat Width”, “Max Boat Draft”, “Finger Pier Placement” and “Berth Number” should be assigned by the designer. The dynamo script uses the values of “Finger Pier Placement” and “Berth Number” to identify the arrangement option of the pier. Other parameters provide information about the dimensions of the boat that are designated to use that specific pier. The slip distances and the berthing lengths are calculated by the dynamo script according to the dimensions of the boats.

After the placement of the piers and the assignment of the parameter by the designer, the process of the script starts with selecting the piers in the model. Later, three functions are defined in the code: (i) to calculate the slip distances for the yacht whose lengths are smaller than 20 m, (ii) to calculate the slip distances for the superyachts whose lengths are longer than 20 m, (iii) to calculate the berthing length for all yachts. For these three calculations, “Finger Pier Placement”, “Max Boat Length” and “Berth Number” parameters are used.

First of all, the functions for slip distances identify whether finger piers are placed to the pier or not. Then, the distances between two finger piers or two anchors placed on the piers are calculated. Then, the calculation of the berthing length is performed. Also, the finger pier widths or the anchor widths are added to the slip distances to identifying more accurate placement points for the finger piers, the anchor, and the boats.

At the end of the first step, selected piers, assigned berth numbers, assigned boat length, assigned boat widths, calculated distances between berthing places, and calculated berth lengths are listed in order. The python codes for the explained steps are given in Figure H.5.

Then, the pier faces are identified as left and right sides separately, as shown in Figure H.1 and Figure H.2.

The lengths of each surface are obtained. The numbers of placement points are calculated by dividing the lengths of surfaces by required distances between berthing structures like finger piers or anchors.

Then, the points are placed on the surfaces. The dynamo node used for the placement of the points locates the points on the corner of the surfaces. It results in unnecessary element placement. To avoid this problem and to provide safer maneuver areas at the ends of the piers, the placement points at those corners are removed. After this correction step, the points' locations and the normals to surface on these points are obtained, used to place the finger piers, anchors or boats. Also, the number of points is used to calculate the boat capacity of each pier.

By “FamilyInstance.ByFace” node, the finger pier family and/or anchor family is placed on the faces. The obtained points are used as location points. The normals are used as reference directions. After placement of the anchors, the placement points of them are also used to place boats.

After the placement, the berthing lengths are set according to calculated values in the first step. Also, the parameter which indicated whether finger piers, anchors, and boats are placed on the right-hand side or the left-hand side is set for all placed families. Then, the maximum boat length, width, and draft values are assigned to them. Lastly, the host pier parameter is assigned to finger piers, anchors, and boats. “Setting Parameter” process is displayed in Figure H.3. The python code in “Setting Parameter” node is given in Figure H.6.

As shown in Figure H.4, the piers, their ID numbers, their assigned maximum boat length, width and draft and the boat capacity are reported in an excel file in the last step of the process.

In summary, the script uses the information in the related parameter regarding the characteristic of the boats and the intended berthing arrangement that are assigned to the piers in the model in order to place finger piers, anchors or boats automatically. It helps to provide the correct slip distances and berthing lengths for safe berthing by

reducing calculation and modeling mistakes in the design process. Therefore, it can be a useful tool for the generation of optimized marina layout alternatives. However, as it is not a fully automated process, the design at the end result of this script can still have design mistakes such as the inner channel distances, the distances between piers or the insufficient water level for safe navigation. For these points, the scripts explained in “3.2.4. RULE CHECKS” can be used by the designer as a control mechanism.



CHAPTER 4

CASE STUDY

The code compliance rules that are explained in Chapter 3 are demonstrated in this chapter. Three different case studies are examined: Finger Pier Arrangement, Mediterranean (Anchor) Arrangement, and Combined Arrangement. These three marina arrangement are created specifically for this study by using “Placement Families on Piers” dynamo script. They do not represent any real-life marina design. The bathymetry model in all of these case studies are the same. As mentioned in Chapter 3, the bathymetry is modeled by using The GEBCO_2014 Grid dataset (Version 20150318). The model represents the Gediz region. Also, the breakwater that is used in these case studies is modeled according to the surface of the bathymetry. In other words, these case studies have the same bathymetry that represents the Gediz region and also, have the same breakwater model. Moreover, the models are created by using erroneous design data to evaluate the performance of the proposed code checking process. The modeling process for each case is explained in the following sections.

4.1 Finger Pier Arrangement

In this arrangement, only finger piers are used to provide berthing places. Seven piers, one breakwater, and bathymetry are modeled initially. Then, the finger piers are placed on the right and the left sides of the piers using “Placement Families on Piers” dynamo script. Incorrect design data in Table 4.1 is used as input during the initial placement. The data shown in bold in Table 4.1 indicates the incorrect design data and the piers that these data are used as input for. Therefore, the results of the code checks can be controlled to see whether design parameters not conforming to the manual can be found or not. Moreover, “Boat Capacity” column in Table 4.1

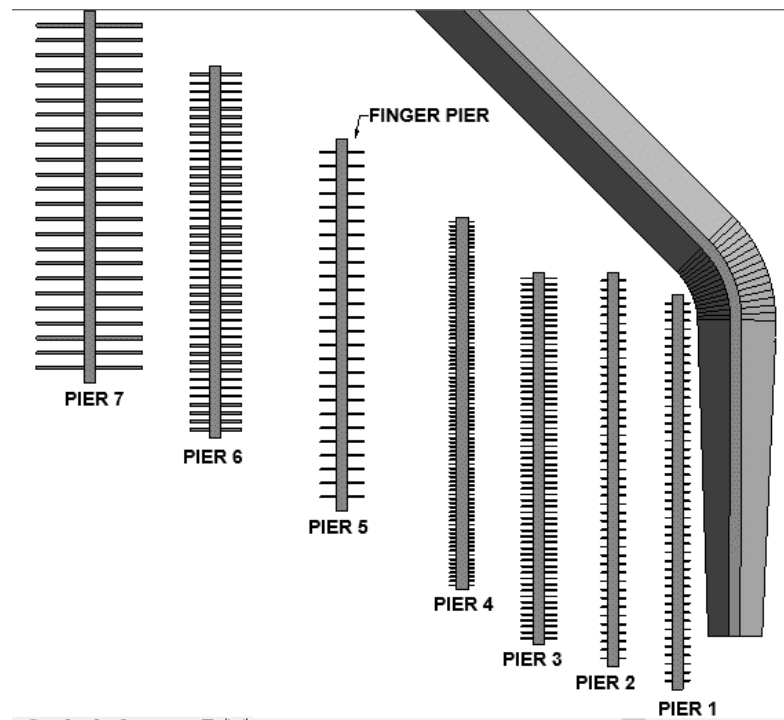
shows the total boat number that the pier can provide berthing. Table 4.2 also gives the boat distribution and the total capacity of the marina. The boat numbers in Table 4.1 and Table 4.2 are calculated according to incorrect design data by “Placing Families on Piers” dynamo script and reported to the excel. The plan layout and 3D view are shown in Figure 4.1.

Table 4.1 Initial Incorrect Design Data for Finger Pier Arrangement

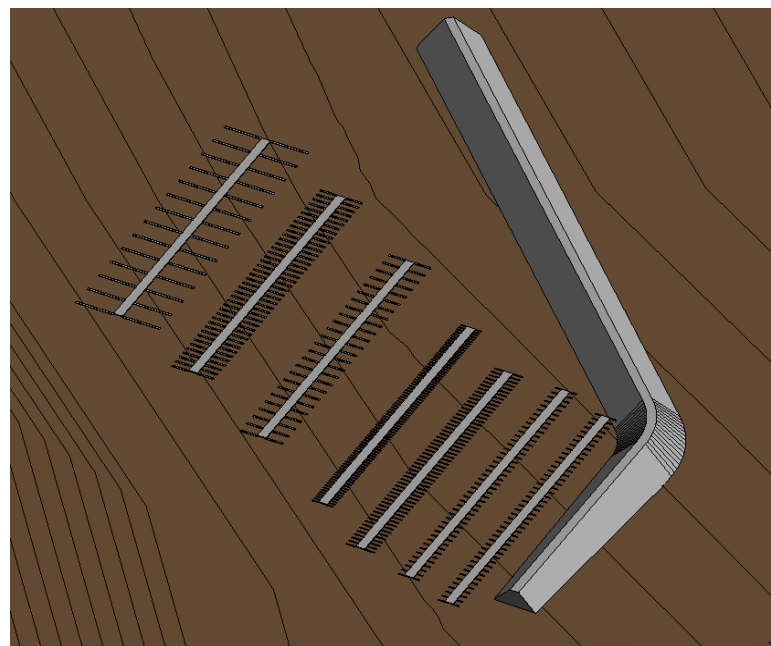
Piers	<i>Max Boat Length (m)</i>	<i>Max Boat Width (m)</i>	<i>Max Boat Draft (m)</i>	<i>Finger Pier Placement</i>	<i>Berth Number</i>	<i>Boat Capacity</i>
Pier 1	5.00	2.20	0.80	Yes	2	184
Pier 2	5.00	2.20	0.80	Yes	2	184
Pier 3	9.00	3.60	1.00	Yes	1	124
Pier 4	5.00	2.20	0.80	Yes	1	172
Pier 5	12.00	4.10	1.20	Yes	2	108
Pier 6	15.00	4.80	1.40	Yes	1	88
Pier 7	35.00	6.80	2.50	Yes	1	50

Table 4.2 Marina Capacity According to Incorrect Design Data For Finger Pier Arrangement

Boat Length (m)	<i># of Boats</i>
5.00	540
9.00	124
12.00	108
15.00	88
35.00	50
Total	910



(a)



(b)

Figure 4.1. Finger Pier Arrangement Layout. (a) Plan (b) 3D

Later, “Max Boat Length”, “Max Boat Width”, “Max Boat Draft” and “Berth Number” parameters are specified for both the piers and the finger piers for the intended design. The values of these parameters for the finger piers are the same as the piers that they are placed on. Besides that, the dimensions of the finger piers are indicated by “Berthing Length”, “Finger Height”, and “Finger Width” parameters. The “Berthing Length” parameter determines the length of the finger piers. The widths of the fingers are decided according to “*Table 4.5 Minimum Floating Finger Pier Widths (Tablo 4.5 Minimum yüzer parmak iskele genişlikleri)*” in Coastal Structures Planning and Design Technical Manual (AYGM, 2016). The “Finger Pier Placement” parameters are checked as “TRUE” for all piers. The correct design data that are assigned to the parameters and the boat capacity of the each pier are given in Table 4.3. Also, the boat distribution and the marina capacity are given in Table 4.4

Table 4.3 Pier Parameter Values for Finger Pier Arrangement

Piers	Max Boat Length (m)	Max Boat Width (m)	Max Boat Draft (m)	Finger Pier Placement	Berth Number	Boat Capacity
Pier 1	5.00	2.20	0.80	Yes	2	184
Pier 2	9.00	3.60	1.00	Yes	2	132
Pier 3	12.00	4.10	1.20	Yes	1	106
Pier 4	12.00	4.10	1.20	Yes	1	106
Pier 5	15.00	4.80	1.40	Yes	2	92
Pier 6	20.00	5.30	1.66	Yes	1	76
Pier 7	35.00	6.80	2.50	Yes	1	50

Table 4.4 Marina Capacity According To Intended Design For Finger Pier
Arrangement

Boat Length (m)	<i># of Boats</i>
5.00	184
9.00	132
12.00	212
15.00	92
20.00	76
35.00	50
Total	654

The code compliance checks are performed, and their results are given in the following sub-sections.

4.1.1 Slip (Berth) Width Distance

The distances between two finger piers are controlled by “Slip (Berth) Distance (W)”. Figure 4.2 shows the finger piers as red that do not provide enough distance for berthing.

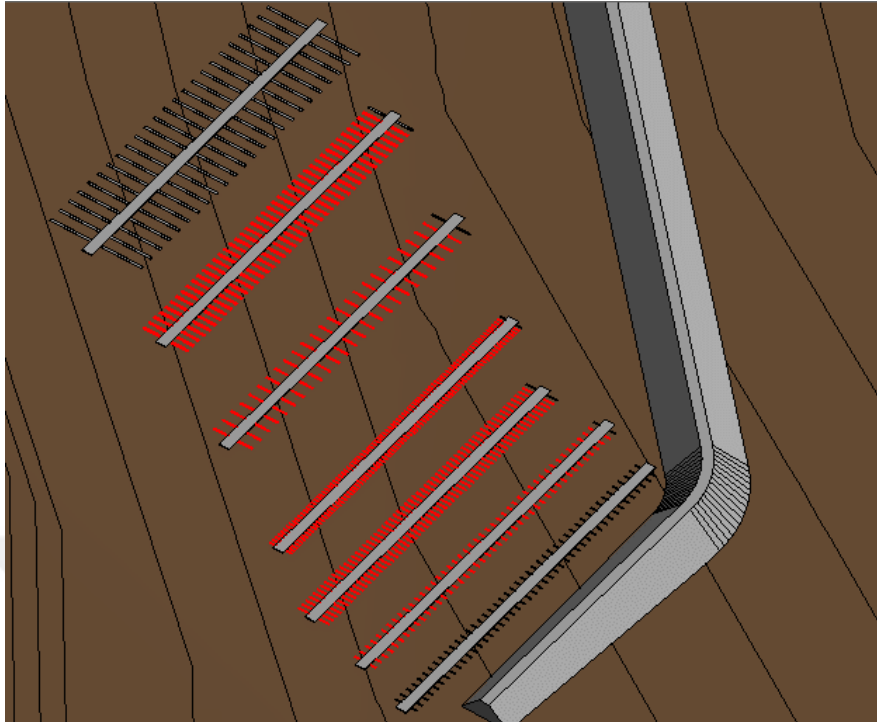


Figure 4.2. Slip (Berth) Distance Rule Checking Result

Table 4.5 provides slip distances calculated using incorrect data and the intended design data for better comparison. The values that have different as a result of the calculation are shown in bold.

Table 4.5 Calculated Slip Distanced for Both Incorrect Design and Intended Design for Finger Pier Arrangement

Piers	<i>Using Incorrect Design Data</i>	<i>Using Intended Design Data</i>
	(m)	(m)
Pier 1	6.40	6.40
Pier 2	6.40	9.20
Pier 3	4.20	4.70
Pier 4	2.80	4.70
Pier 5	10.20	11.60
Pier 6	5.40	5.90
Pier 7	20.00	20.00

The result of this check is exported to “Slip (Berth) Distance” page in “Finger Pier Arrangement” excel file.

Table 4.6 Slip (Berth) Distance Partial Results for Finger Pier Arrangement

IDs	Elements	Distance	Min. Distance	T/F
242106	Finger Pier	4.32	4.7	FALSE
242108	Finger Pier	4.32	4.7	FALSE
242110	Finger Pier	4.32	4.7	FALSE
242112	Finger Pier	4.32	4.7	FALSE
242114	Finger Pier	4.32	4.7	FALSE
242116	Finger Pier	4.32	4.7	FALSE
242118	Finger Pier	4.32	4.7	FALSE
242120	Finger Pier	4.32	4.7	FALSE
242122	Finger Pier	4.32	4.7	FALSE
242124	Finger Pier	4.32	4.7	FALSE
242126	Finger Pier	4.32	4.7	FALSE
242128	Finger Pier	4.32	4.7	FALSE
242130	Finger Pier	4.32	4.7	FALSE
242132	Finger Pier	4.32	4.7	FALSE
242134	Finger Pier	4.32	4.7	FALSE
242136	Finger Pier	4.32	4.7	FALSE
242138	Finger Pier	4.32	4.7	FALSE
242140	Finger Pier	4.32	4.7	FALSE
242142	Finger Pier	4.32	4.7	FALSE
242144	Finger Pier	4.32	4.7	FALSE

Part of the results is seen in Table 4.4. The result contains the following items:

- **Element IDs:** With this information, the user can find the exact elements that do not satisfy the required distance.
- **Element Name:** This information contains the family name of the element.
- **Distance:** It is the measurement value between a finger pier and the next one.

- **Min. Distance:** It is the required minimum distance between two finger piers. It is calculated according to “Max Boat Width” parameter of the element whose ID is given in the row.
- **T/F:** It stands for “True/False”. It gives the results of the comparison between “Min. Distance” and “Distance. “False” result means that the measured distance does not satisfy the required minimum distance. “True” result means that the measured distance is larger than the required minimum distance.

4.1.2 Inner Channel Width (B_1)

“Inner Channel Width” dynamo script is used for this check. The user specifies the inner channel area before performing the check. For this layout, the inner channel area is the same, as shown in Figure 3.44. Figure 4.3 shows the elements that do not satisfy the 1.5L criteria as red. If there is an element that does not ensure the minimum 20 m criteria, it will be shown in blue. However, in this model, there is not any element that does not satisfy the minimum 20 m criteria.

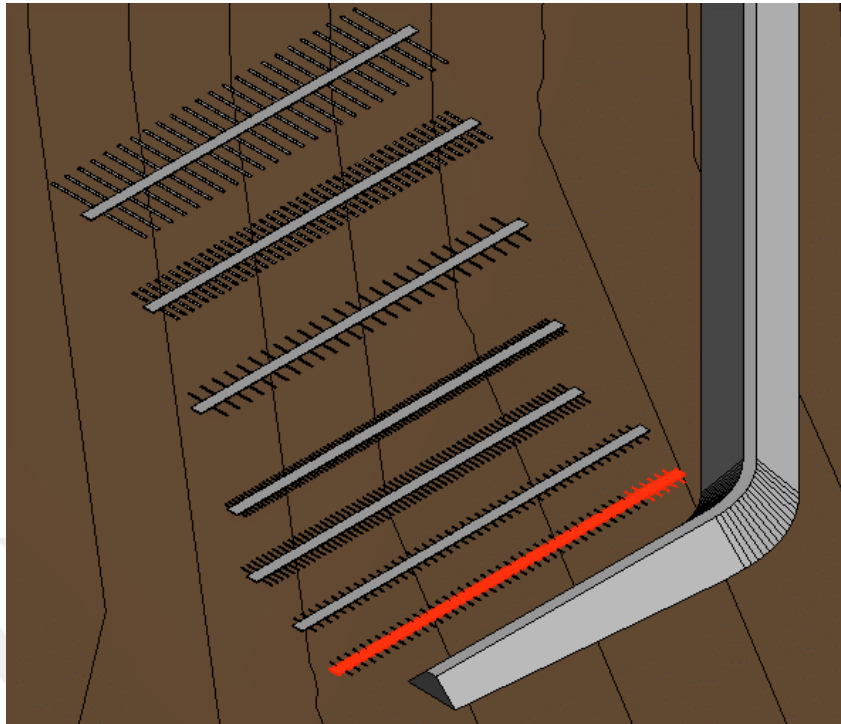


Figure 4.3. Inner Channel Width Rule Checking Result for Finger Pier Arrangement

The result is also given in Table 4.5 is exported to “Inner Channel Width” page in “Finger Pier Arrangement” excel file.

Table 4.7 Inner Channel Width Results for Finger Pier Arrangement Layout

Not satisfied the '1.5L' criteria	Not satisfied the 'Min 20 m' criteria
217891	
241350	
241352	
241354	
241356	
242006	
242008	
242010	
242012	
242014	

The results are given in two different columns:

- **Not satisfied the ‘1.5L’ criteria:** The elements that do not provide the required minimum distance calculated by $1.5 \times L$ formula are given under this headline.
- **Not satisfied the ‘Min. 20 m’ criteria:** As indicated before, the inner channel width must be minimum 20 m according to the design guideline. The elements that do not provide the minimum 20 m are given under this headline.

4.1.3 Distance Between Piers

The distance between the two piers is checked with “Distance Between Piers (S)” dynamo script. The distance that is the concern of this code checking process is displayed in Figure 4.4.

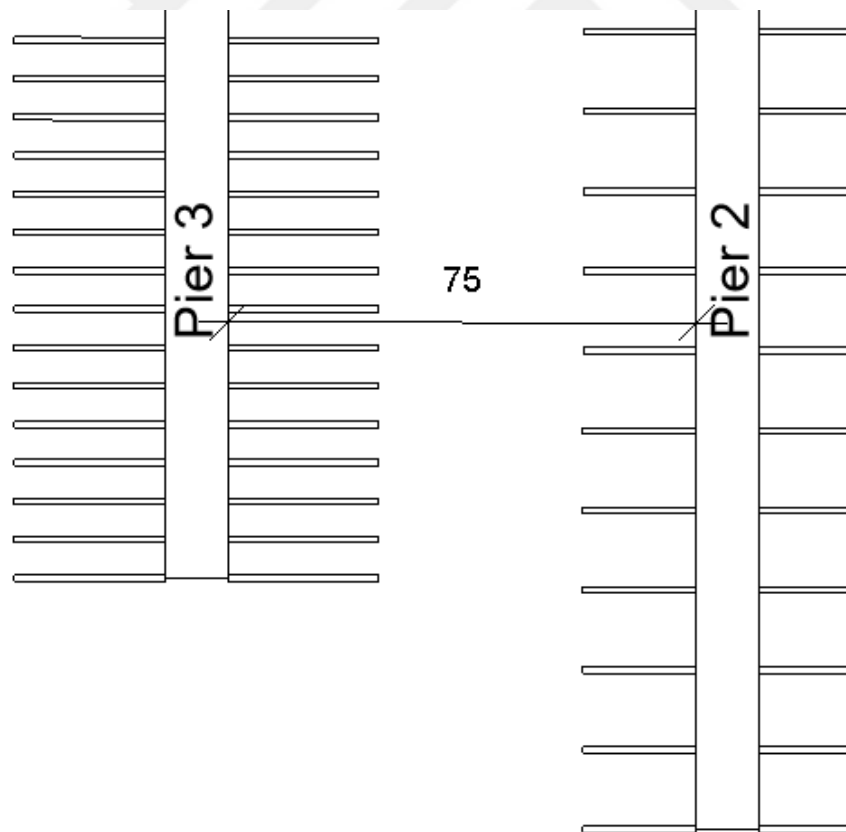


Figure 4.4. Pier Distance That is Controlled for Finger Pier Arrangement Layout

The piers that do not provide enough distance between themselves with the one on their right in plan view are given as blue, as shown in Figure 4.5.

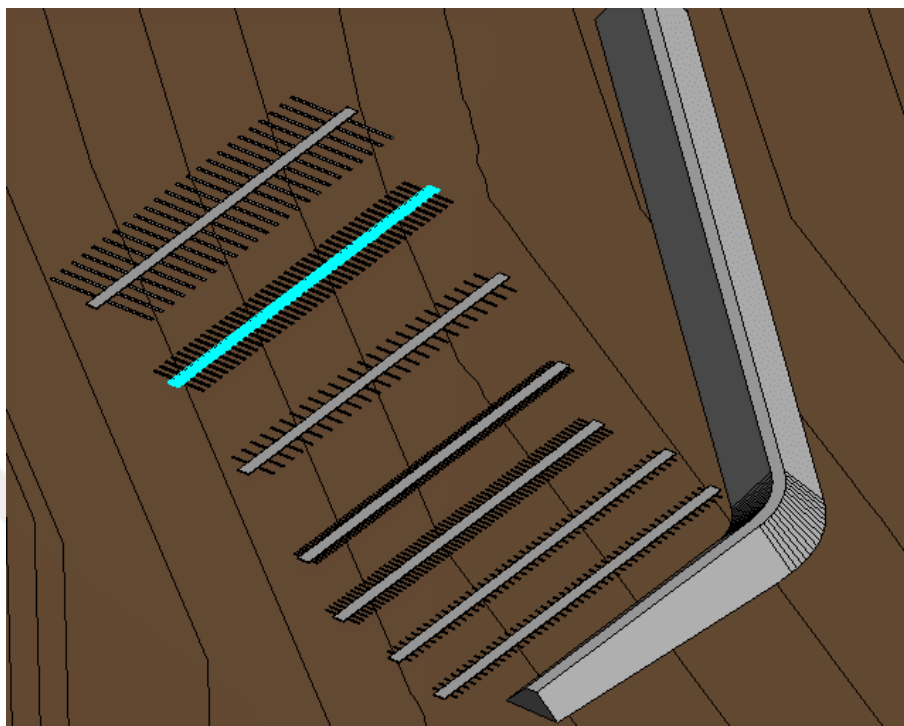


Figure 4.5. Distance Between Piers Rule Checking Result for Finger Pier Arrangement

Moreover, the results are exported to “S” page in “Finger Pier Arrangement” excel file and given in Table 4.6.

Table 4.8 Results of Distance Between Piers for Finger Pier Arrangement

IDs	Elements	Distance	Min.Distance	T/F
217891	Pier	46.48	34.8	TRUE
217899	Pier	56.06	49.2	TRUE
217907	Pier	58.19	52.8	TRUE
217915	Pier	96.55	62.4	TRUE
217923	Pier	102.08	82	TRUE
217931	Pier	101.87	136	FALSE
217939	Pier	0		

The table contains the following information:

- **IDs:** With this information, the user can find the exact elements that do not satisfy the required distance.
- **Element Name:** This information contains the family name of the element.
- **Distance:** It is the measured distance between the two piers, as shown in Figure 4.2.
- **Min. Distance:** It is the required minimum distance between two piers. It depends on “Max Boat Length” parameter values of two piers next to each other. The calculation logic is explained in the Methodology Chapter.
- **T/F:** It stands for “True/False”. It gives the results of the comparison between the calculated minimum distance and the measured distance between two piers. “False” result means that the measured distance does not satisfy the required minimum distance. “True” result means that the measured distance is larger than the required minimum distance. The result is for the distance between the pier given in the row and the pier next to it in the model.

The last row contains only IDs, Element, and Distance information because the elements are compared with the next one in the element list. The last row contains the last element in the list. Therefore, it does not have a next element to compare.

4.1.4 Water Level Check

The water level in the marina region is controlled with “Water Level Check” dynamo script. The original layout is modeled according to the real coastal line. Therefore, on the first run of the code, no elements were identified that did not satisfy the water level check. Hence, Pier 7 is relocated toward to coastal line for performing a controlled experiment. The resulted layout is given in Figure 4.6.

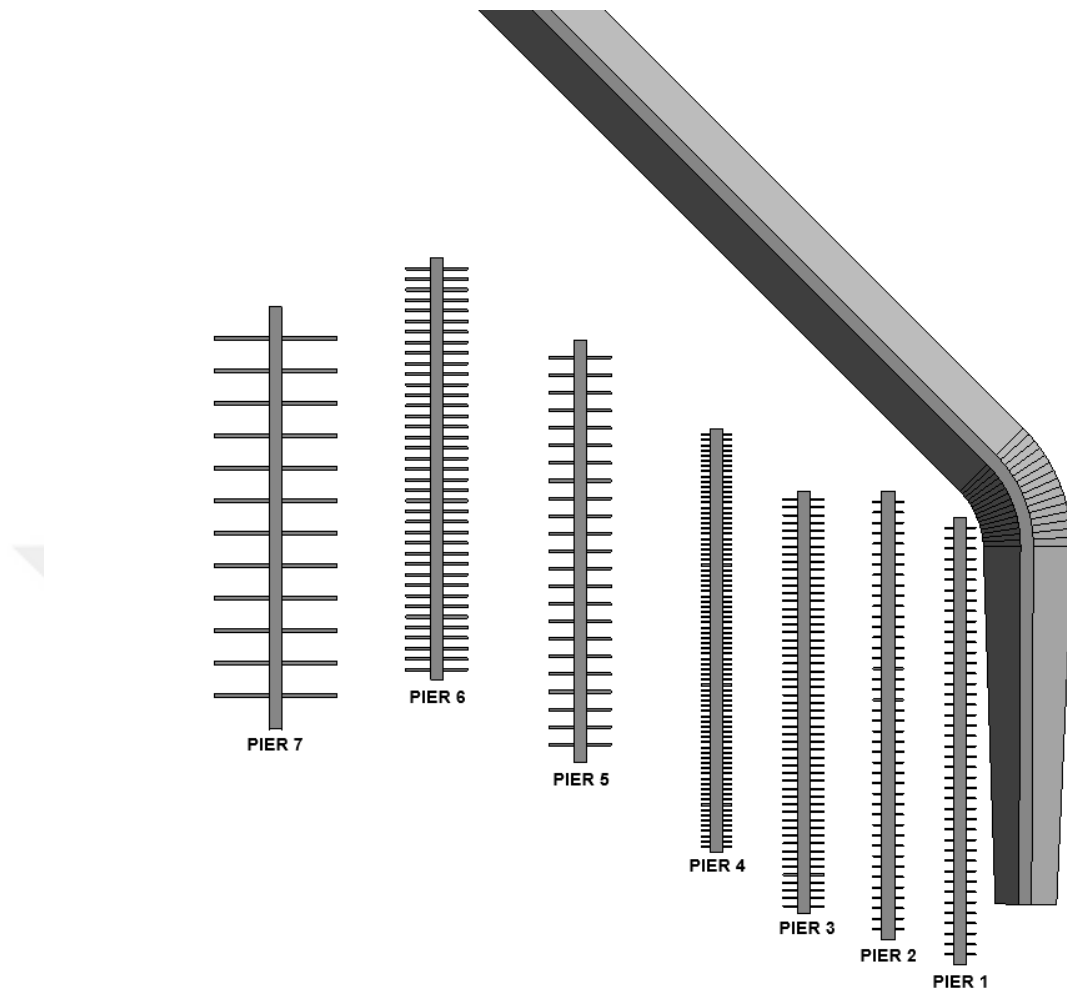


Figure 4.6. Finger Pier Arrangement Plan for Water Level Check

The yellow finger piers that are shown in Figure 4.7 do not satisfy the water level requirement.

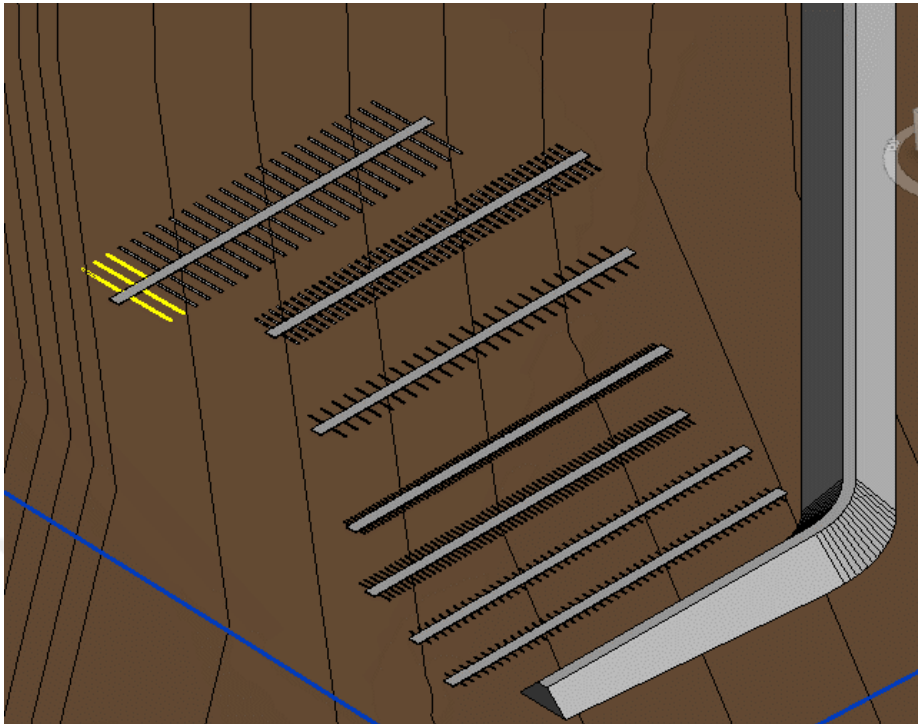


Figure 4.7. Water Level Check Result For Finger Pier Arrangement

As a result, the dynamo script gives two different results in “Finger Pier Arrangement” excel file. One of them is a list of elements that do not satisfy the minimum water level. It is given on “Water Level Check” page in excel. Part of the result is seen in Table 4.7.

Table 4.9 The Water Level Check Result For Finger Pier Arrangement Layout

IDs	Elements
242536	Finger Pier
242538	Finger Pier
241878	Finger Pier
241880	Finger Pier
241882	Finger Pier

The second list of the results on “Lowest Point” page in excel provides the lowest point locations in the marina region. The elevations of these points are lower than the required minimum water level. The aim to identify the lowest points is to

determine whether the area requires dredging to provide water level distance for safe marina operations. Part of the result is given in Table 4.8.

Table 4.10 The Part of The Lowest Point Results For Finger Pier Arrangement
Layout

Point
Point(X = 456.461, Y = -610.173, Z = 1.235)
Point(X = 463.891, Y = -610.173, Z = 0.954)
Point(X = 466.093, Y = -610.173, Z = 0.871)
Point(X = 496.216, Y = -610.173, Z = -0.266)
Point(X = 515.675, Y = -610.173, Z = -1.000)
Point(X = 992.227, Y = -610.173, Z = 0.000)
Point(X = 992.441, Y = -610.173, Z = 0.000)
Point(X = 992.548, Y = -610.173, Z = 0.000)
Point(X = 992.748, Y = -610.173, Z = 0.000)
Point(X = 992.947, Y = -610.173, Z = 0.000)
Point(X = 992.947, Y = -610.041, Z = 0.000)
Point(X = 992.947, Y = -609.909, Z = 0.000)
Point(X = 992.947, Y = -609.838, Z = 0.000)
Point(X = 992.947, Y = -609.695, Z = 0.000)
Point(X = -66.053, Y = -78.421, Z = -1.196)
Point(X = -66.053, Y = -103.190, Z = 0.043)
Point(X = -66.053, Y = -115.262, Z = 0.647)
Point(X = 12.484, Y = -198.706, Z = 1.000)
Point(X = 24.716, Y = -190.613, Z = 0.000)
Point(X = 36.948, Y = -182.519, Z = -1.000)
Point(X = 405.241, Y = -534.364, Z = 1.000)
Point(X = 440.545, Y = -545.988, Z = 0.000)
Point(X = 475.849, Y = -557.612, Z = -1.000)
Point(X = 208.862, Y = -366.535, Z = 1.000)
Point(X = 232.630, Y = -368.300, Z = 0.000)
Point(X = 256.398, Y = -370.065, Z = -1.000)
Point(X = 110.673, Y = -282.621, Z = 1.000)
Point(X = 128.673, Y = -279.456, Z = 0.000)
Point(X = 146.673, Y = -276.292, Z = -1.000)
Point(X = 307.052, Y = -450.450, Z = 1.000)
Point(X = 336.588, Y = -457.144, Z = 0.000)

4.1.5 Berthing Length (B)

The berthing length displayed in Figure 4.8 is controlled by “Berthing Length (B)” dynamo script. The berthing length in the case of finger pier arrangement is equal to the finger length.

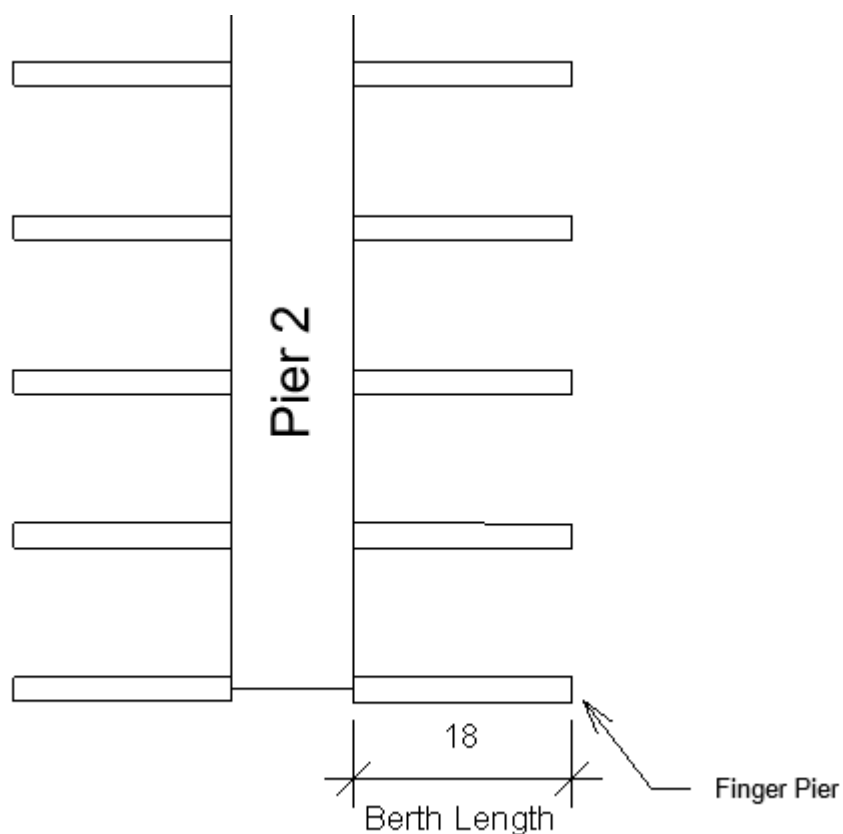


Figure 4.8. Berthing Length Measurement For Finger Pier Arrangement Layout

The magenta finger piers shown in Figure 4.9 do not satisfy the minimum berthing length requirement.

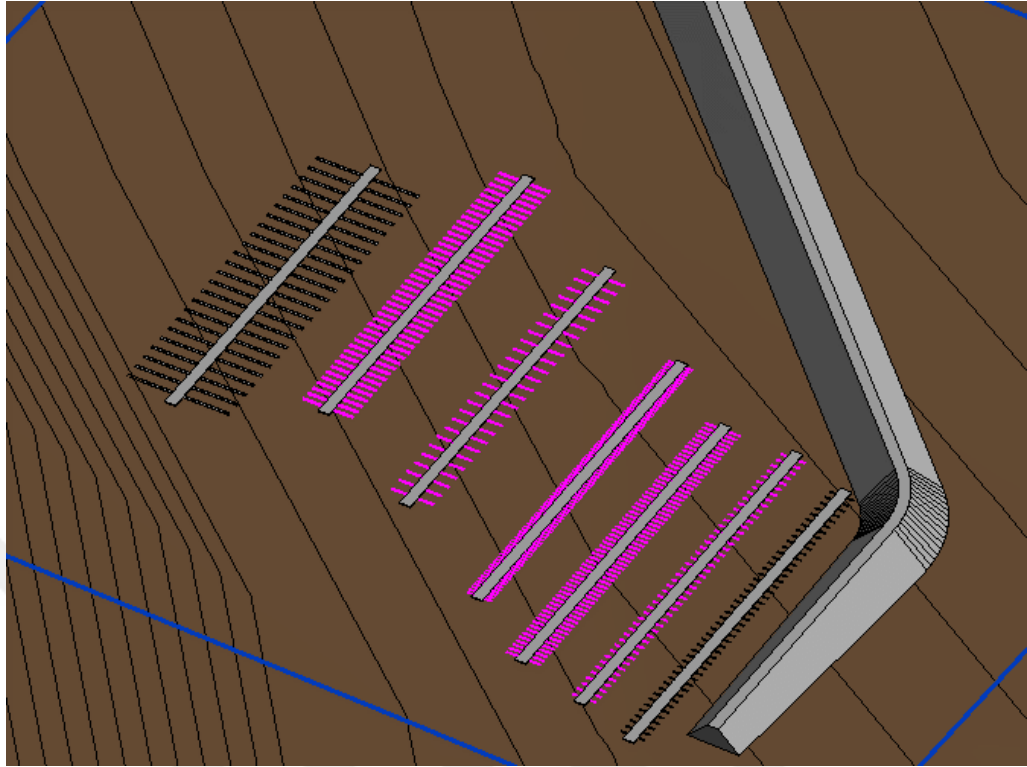


Figure 4.9. Berthing Length Rule Checking Results For Finger Pier Arrangement

The code check results are also given on “Berthing Length” sheet in “Finger Arrangement” excel file. There are two columns in the results, as shown in Table 4.9. The results are given partially because it has IDs of all elements in the model. “TRUE” column consists of element IDs whose lengths are equal to or larger than the calculated berthing distance. On the other hand, “FALSE” column provides IDs of elements whose lengths are smaller than the calculated berthing length.

Table 4.11 Part of The Berthing Length Results for Finger Pier Arrangement

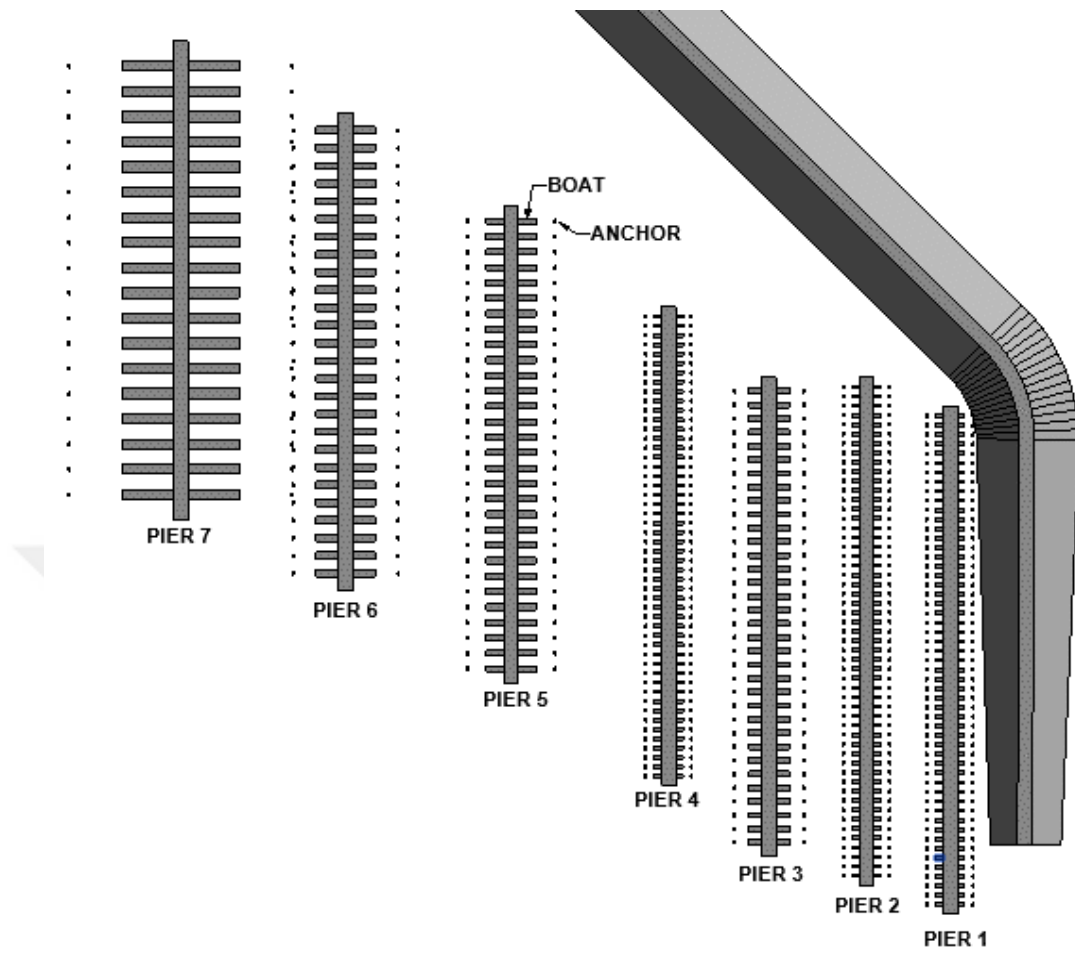
TRUE	FALSE
241268	241358
241270	241360
241272	241362
241274	241364
241276	241366
241278	241368
241280	241370

Table 4.11 Part of The Berthing Length Results for Finger Pier Arrangement
(continued)

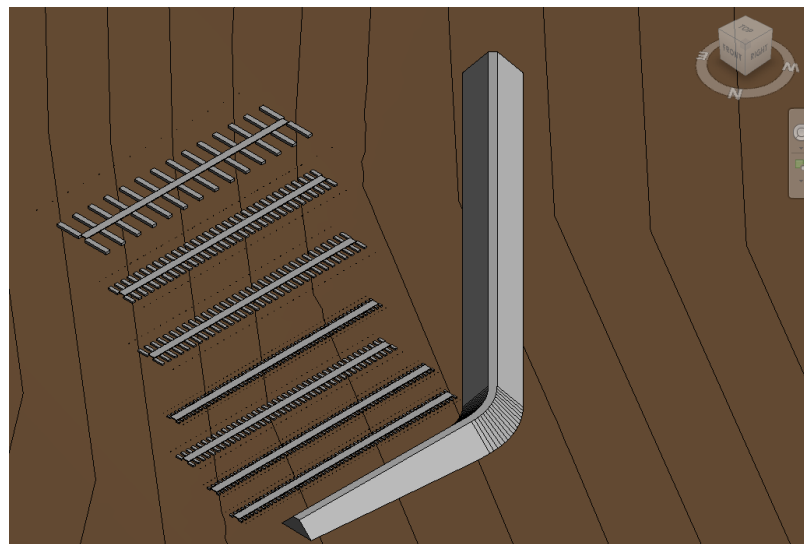
TRUE	FALSE
241282	241372
241284	241374
241286	241376
241288	241378
241290	241380
241292	241382
241294	241384
241296	241386
241298	241388
241300	241390
241302	241392
241304	241394
241306	241396

4.2 Mediterranean (Anchor) Arrangement

The Mediterranean arrangement consists of anchors and boats. The boats berth next to each other with the help of anchors, as shown in Figure 3.6. The modeling process is the same as the finger pier arrangement case. Firstly, seven piers and one breakwater are placed in the model. Then, the anchors and the boats are placed on the right and the left sides of the piers by using “Placement Families on Piers” dynamo script. As in the finger pier arrangement case, initial placement data is incorrect, which is given in Table 4.10. Also, the boat capacities of each pier are reported in Table 4.10. The incorrect design data and the piers that these data are used as input for are shown in bold in Table 4.10. Table 4.11 also gives the boat distribution and the total capacity of the marina. The plan layout and 3D view are shown in Figure 4.10.



(a)



(b)

Figure 4.10. Mediterranean Arrangement. (a) Plan Layout (b) 3D View

Table 4.12 Initial Incorrect Design Data for Mediterranean Arrangement

Piers	<i>Max Boat Length</i> (m)	<i>Max Boat Width</i> (m)	<i>Max Boat Draft</i> (m)	<i>Finger Pier Placement</i>	<i>Berth Number</i>	<i>Boat Capacity</i>
Pier 1	5.00	2.20	0.80	No	N/A	106
Pier 2	5.00	2.20	0.80	No	N/A	106
Pier 3	9.00	3.60	1.00	No	N/A	68
Pier 4	5.00	2.20	0.80	No	N/A	100
Pier 5	12.00	4.10	1.20	No	N/A	60
Pier 6	15.00	4.80	1.40	No	N/A	52
Pier 7	35.00	6.80	2.50	No	N/A	36

Table 4.13 Marina Capacity According To Incorrect Design Data For Mediterranean Arrangement

Boat Length (m)	# of Boats
5.00	312
9.00	68
12.00	60
15.00	52
35.00	36
Total	528

“Max Boat Length”, “Max Boat Width”, and “Max Boat Draft” parameters are specified for all elements in the model except for the breakwater. The values of these parameters are the same as the piers placed on for the boats and the anchors. The boats dimension depend on “Max Boat Length”, “Max Boat Width”, and “Max Boat

Draft”. On the other hand, anchors do not have any dimensional features. The berthing distance that is displayed in Figure 4.11 is indicated by “Berthing Length” parameter. “Finger Pier Placement” parameters are unchecked for all piers in this case. The values assigned to parameters are given in Table 4.11.

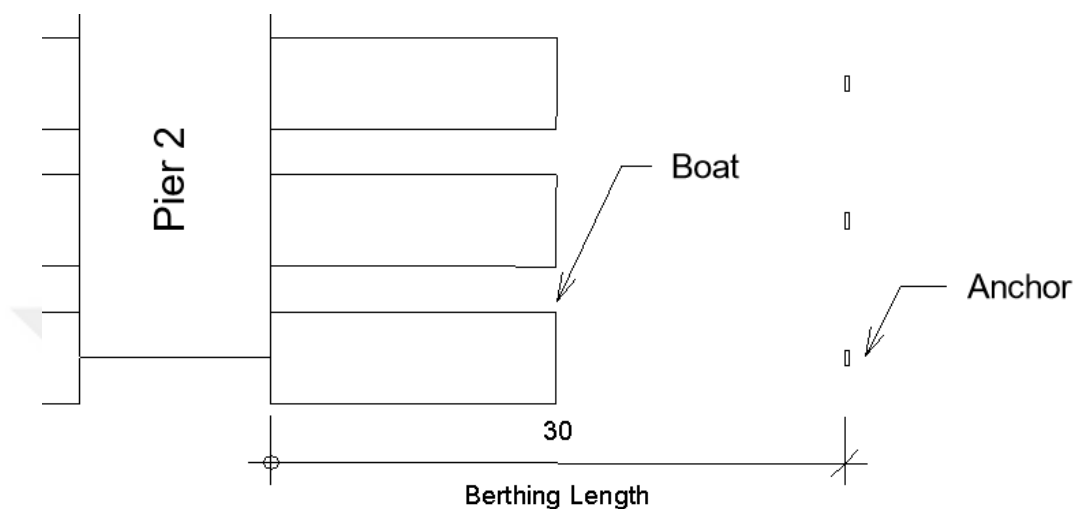


Figure 4.11. Berthing Length Measurement for Mediterranean Arrangement

Table 4.14 Pier Parameter Values for Mediterranean Arrangement

Piers	Max Boat Length (m)	Max Boat Width (m)	Max Boat Draft (m)	Finger Pier Placement	Berth Number	Boat Capacity
Pier 1	5.00	2.20	0.80	No	N/A	106
Pier 2	9.00	3.60	1.00	No	N/A	72
Pier 3	12.00	4.10	1.20	No	N/A	60
Pier 4	12.00	4.10	1.20	No	N/A	60
Pier 5	15.00	4.80	1.40	No	N/A	52
Pier 6	20.00	5.30	1.66	No	N/A	48
Pier 7	35.00	6.80	2.50	No	N/A	36

The code compliance checks are performed and their results are given in the following sub-sections.

Table 4.15 Marina Capacity According To Intended Design For Mediterranean Arrangement

Boat Length (m)	<i># of Boats</i>
5.00	106
9.00	72
12.00	120
15.00	52
20.00	48
35.00	36
Total	434

4.2.1 Slip (Berth) Width Distance

The distances between two anchors and boats are controlled by “Slip (Berth) Distance (W)” dynamo script. Red boats shown in Figure 4.12 do not have enough distance for safe berthing.

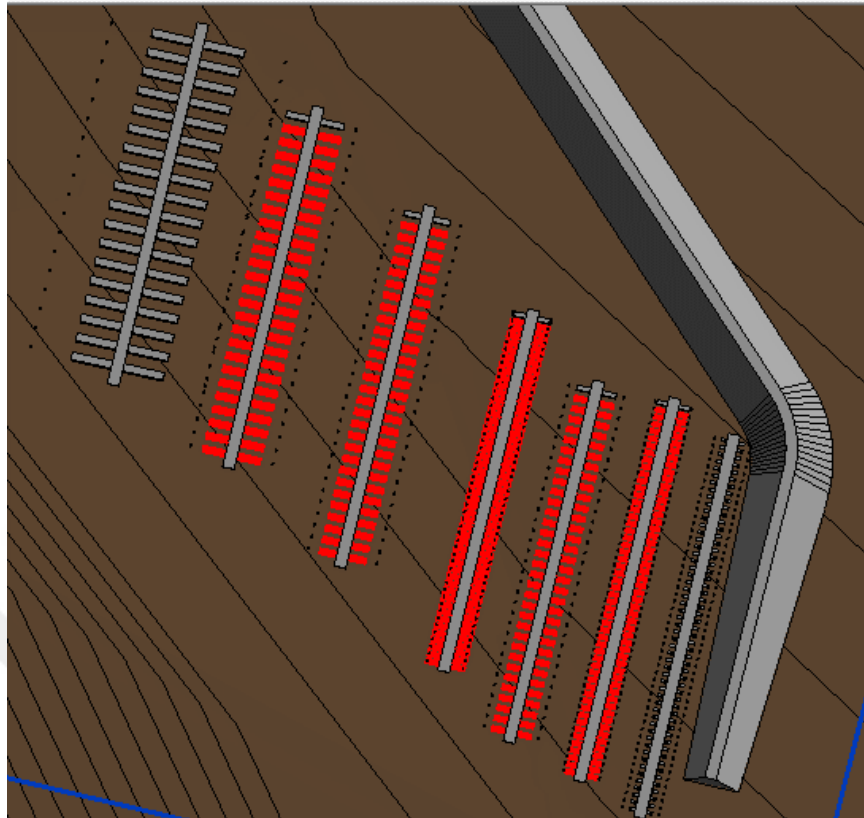


Figure 4.12. Slip (Berth) Distance Rule Checking Result for Mediterranean (Anchor) Arrangement

The slip distances calculated according to the incorrect design data and the intended design data are compared in Table 4.16. The different results are shown in bold.

Table 4.16 Calculated Slip Distanced For Both Incorrect Design and Intended Design for Mediterranean (Anchor) Arrangement

Piers	<i>Using Incorrect Design Data</i>	<i>Using Intended Design Data</i>
	(m)	(m)
Pier 1	4.20	4.20
Pier 2	4.20	5.60
Pier 3	5.60	6.10
Pier 4	4.20	6.10
Pier 5	6.10	6.80

Table 4.16 Calculated Slip Distanced For Incorrect Design and Intended Design for Mediterranean (Anchor) Arrangement (continued)

Piers	<i>Using Incorrect Design Data</i>	<i>Using Intended Design Data</i>
	(m)	(m)
Pier 6	6.80	7.30
Pier 7	20.00	20.00

The results part of which is given in Table 4.17 is exported to “Slip (Berth) Distance” sheet of “Mediterranean Arrangement” excel file.

Table 4.17 Part of “Clearance Check between Berthing Place” Results For Mediterranean Arrangement

IDs	Elements	Distance	Min. Distance	T/F
242720	Boat	5.33	6.1	FALSE
242722	Boat	5.33	6.1	FALSE
242724	Boat	5.33	6.1	FALSE
242726	Boat	5.33	6.1	FALSE
242728	Boat	5.33	6.1	FALSE
242730	Boat	5.33	6.1	FALSE
242732	Boat	5.33	6.1	FALSE
242734	Boat	5.33	6.1	FALSE
242736	Boat	5.33	6.1	FALSE
242738	Boat	5.33	6.1	FALSE
242740	Boat	5.33	6.1	FALSE
242742	Boat	5.33	6.1	FALSE
242744	Boat	5.33	6.1	FALSE
242746	Boat	5.33	6.1	FALSE
242748	Boat	5.33	6.1	FALSE
242750	Boat	5.33	6.1	FALSE
242752	Boat	5.33	6.1	FALSE
242754	Boat	5.33	6.1	FALSE

4.2.2 Inner Channel Width (B_1)

Inner Channel Width that is specified as shown in Figure 3.44 is checked by “Inner Channel Width” dynamo script. Figure 3.44 illustrates the inner channel area for this layout. The results of this code check are displayed in Figure 4.13. The red coloring means that those elements do not satisfy the 1.5L criterion. On the other hand, the blue-colored element does not satisfy even the minimum 20 m criterion. Figure 4.13 displays the elements that do not satisfy the 1.5L criterion in red. Also, the element shown in blue does not satisfy the minimum 20 m criterion.

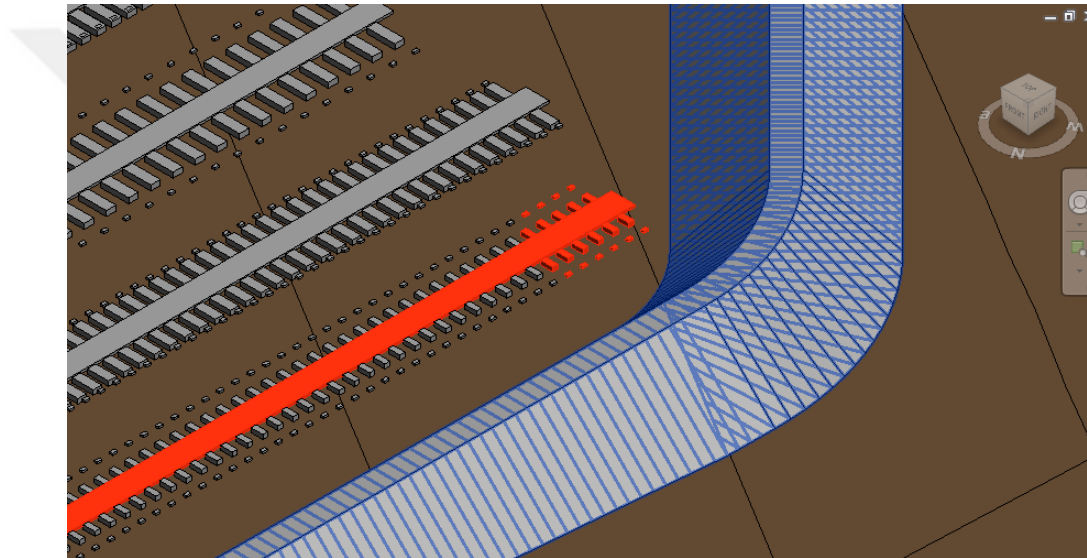


Figure 4.13. Inner Channel Width Rule Checking Result for Mediterranean Arrangement

The results given in Table 4.18 is exported to “Inner Channel Width” sheet in “Mediterranean Arrangement” excel file.

Table 4.18 Inner Channel Width Results for Mediterranean Arrangement Layout

Not satisfied the '1.5L' criteria	Not satisfied the 'Min 20 m' criteria
217891	
241548	
241550	
241552	
241554	
241556	
242078	
242080	
242082	
242084	
242602	
242604	
242606	
242608	
242610	
242612	
243130	
243132	
243134	
243136	
243138	
243140	

4.2.3 Distance Between Piers

The distance between the two piers is checked with “Distance Between Piers (S)” dynamo script. The distances that are the concern of this code checking process is displayed in Figure 4.14.

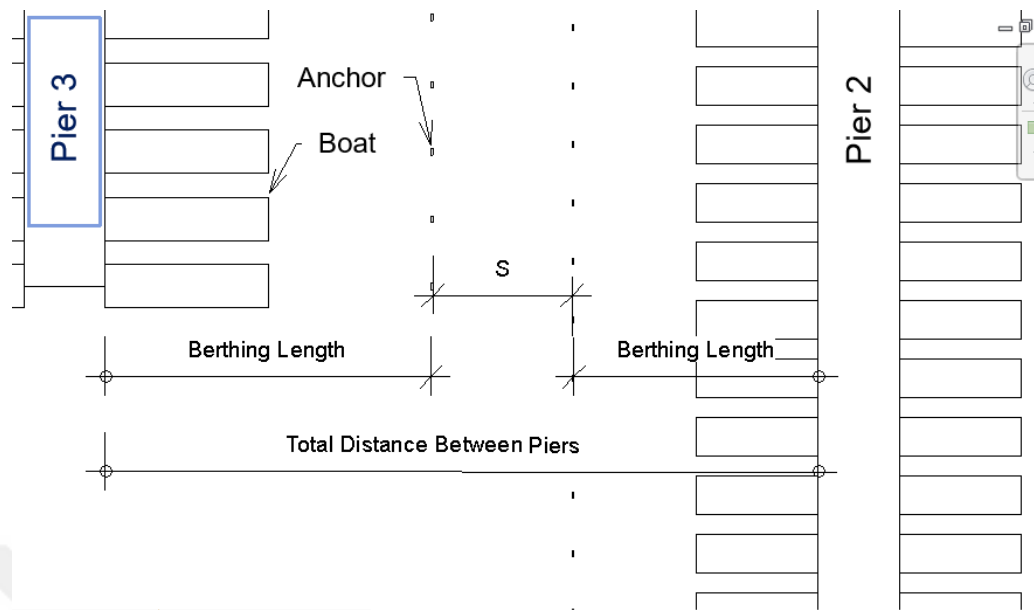


Figure 4.14. Measurement that is Used in the Control for Mediterranean Arrangement

There is not enough space between the blue pier in Figure 4.15 and the one on its right in the plan view.

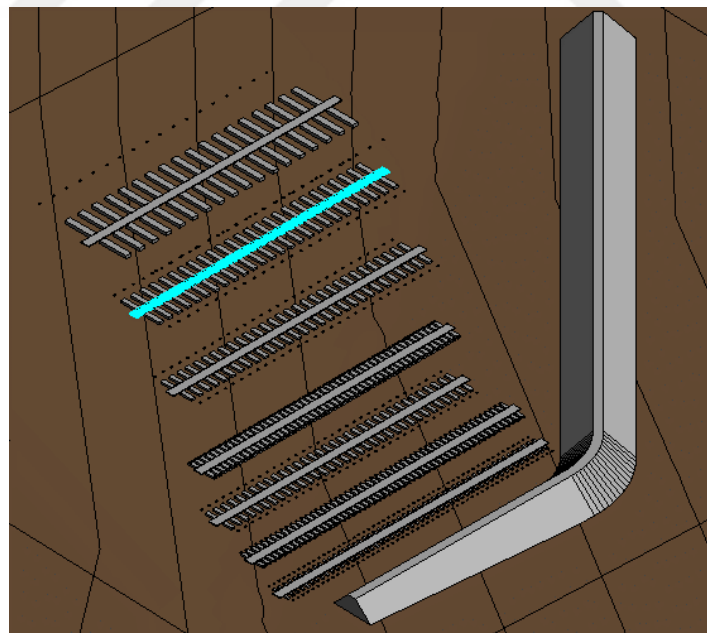


Figure 4.15. Distance Between Piers Rule Checking Result for Mediterranean Arrangement

The results of this check are exported to “S” sheet in “Mediterranian Arrangement” excel file and given in Table 4.19.

Table 4.19 Results of Distance Between Piers for Mediterranean Arrangement
Layout

IDs	Elements	Distance	Min.Distance	T/F
217891	Pier	46.48	34.8	TRUE
217899	Pier	56.06	49.2	TRUE
217907	Pier	58.19	52.8	TRUE
217915	Pier	96.55	62.4	TRUE
217923	Pier	102.08	82	TRUE
217931	Pier	101.87	136	FALSE
217939	Pier	0		

4.2.4 Water Level Check

“Water Level Check” dynamo script is used in this check. The original layout is modeled according to the real coastal line, so Pier 5 is relocated toward to coastal line for performing a controlled experiment. The resulted layout is given in Figure 4.16.

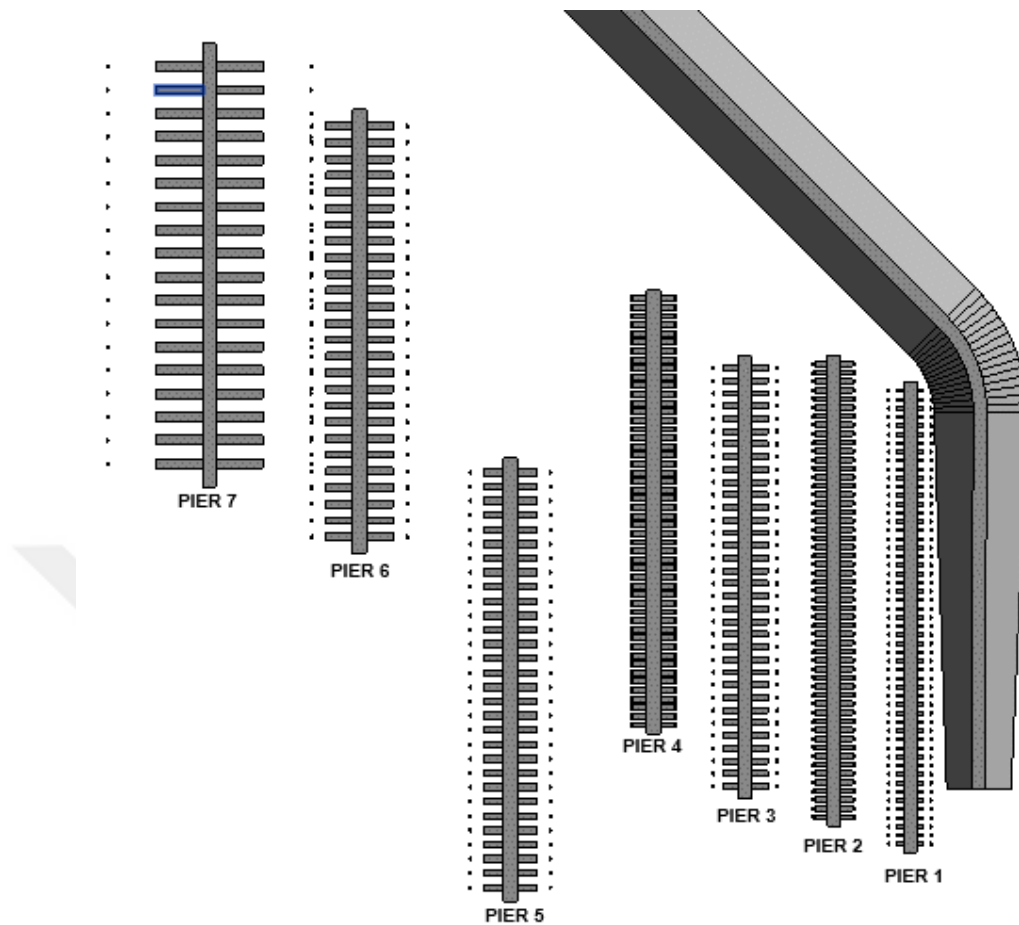


Figure 4.16. Mediterranean Arrangement Plan for Water Level Check

The anchors that do not satisfy the water level requirement are shown in yellow in Figure 4.17. Nonetheless, some elements in yellow are expected to satisfy the minimum water level. The dynamo node obtains the elements' location points and their projection points to the bathymetry to measure water level distance. However, from the original model in Autodesk Civil 3D until the run in Dynamo script, the bathymetry area is narrowed down twice. Therefore, some point data can be lost during these steps and the Dynamo nodes cannot obtain the point data from the model correctly. As a result, in some locations, the node cannot obtain the projected point from the bathymetry. In these situations, the measurements are resulted as “null”. Thus, the elements in these kinds of situations are listed in “False” list and are colored in yellow. The highlighted elements in Table 4.20 are the ones that gave

“null” results. The other elements in Table 4.20 do not have enough water level for safe berthing.

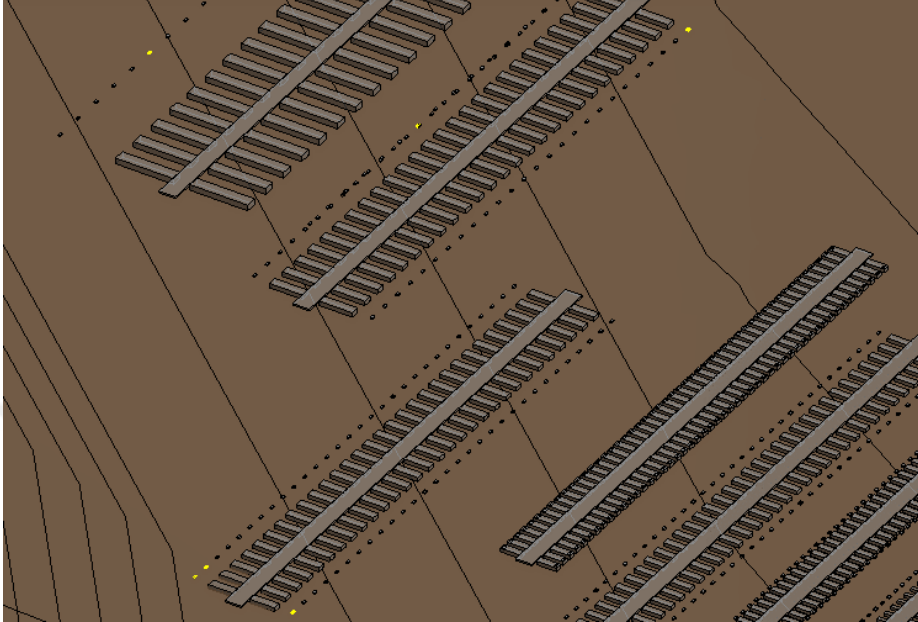


Figure 4.17. Water Level Check Result for Mediterranean Arrangement

Table 4.20 The Water Level Check Result for Mediterranean Arrangement

IDs	Elements
243416	Anchor
243526	Anchor
243540	Anchor
242360	Anchor
242362	Anchor
242482	Anchor

Table 4.21 provides the part of the point list that specifies the lowest regions in the marina area.

Table 4.21 The Part of The Lowest Point Results for Mediterranean Arrangement

Point
Point(X = 68.001, Y = -215.886, Z = -0.632)
Point(X = 68.001, Y = -223.549, Z = -0.219)
Point(X = 68.001, Y = -238.271, Z = 0.575)
Point(X = 493.522, Y = -638.799, Z = 0.655)
Point(X = 506.602, Y = -638.799, Z = 0.161)
Point(X = 523.470, Y = -638.799, Z = -0.476)
Point(X = 537.366, Y = -638.799, Z = -1.000)
Point(X = 949.059, Y = -638.799, Z = 0.000)
Point(X = 949.195, Y = -638.799, Z = 0.000)
Point(X = 949.264, Y = -638.799, Z = 0.000)
Point(X = 949.471, Y = -638.799, Z = 0.000)
Point(X = 949.678, Y = -638.799, Z = 0.000)
Point(X = 949.678, Y = -638.662, Z = 0.000)
Point(X = 949.678, Y = -638.525, Z = 0.000)
Point(X = 949.678, Y = -638.479, Z = 0.000)
Point(X = 949.678, Y = -638.388, Z = 0.000)
Point(X = 405.241, Y = -534.364, Z = 1.000)

4.2.5 Berthing Length (B)

“Berthing Length (B)” dynamo script is used to control the distance of berthing length. The berthing length measurement for the anchor is shown in Figure 4.18.

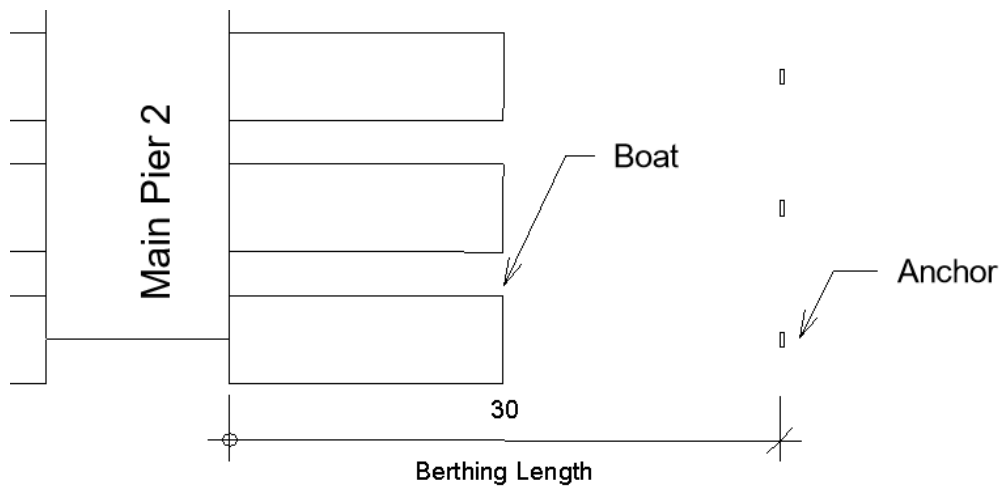


Figure 4.18. Berthing Length Measurement for Mediterranean Arrangement

The anchors that do not satisfy the minimum berthing length requirement are colored in magenta. Figure 4.19 shows those magenta anchors.

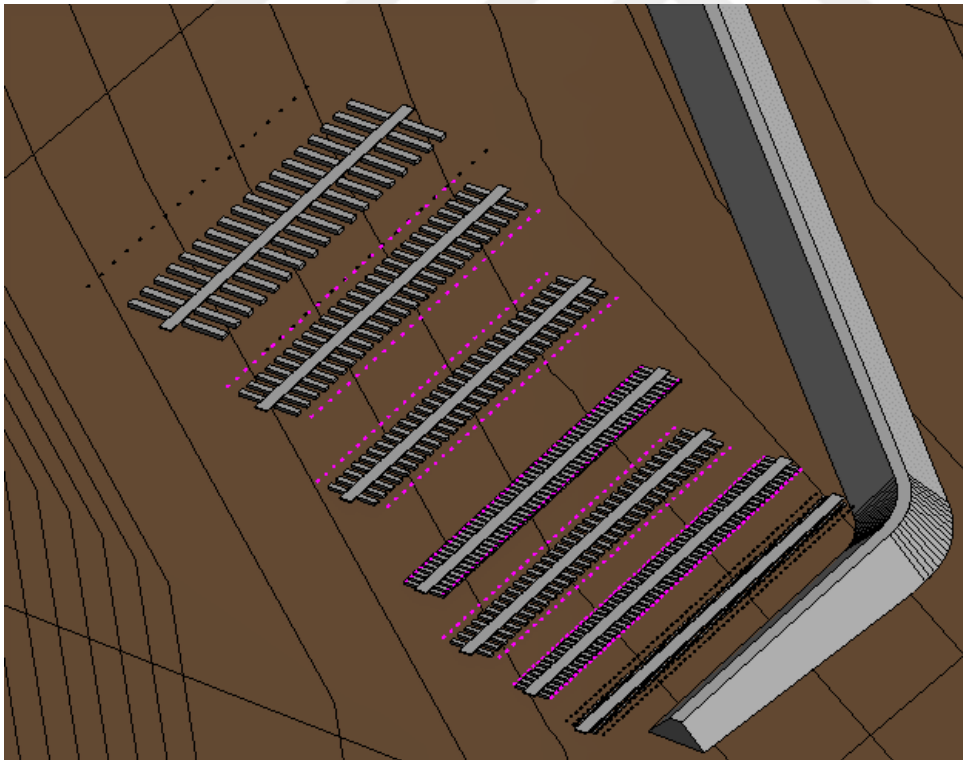


Figure 4.19. Berthing Length Rule Checking Results for Mediterranean Arrangement

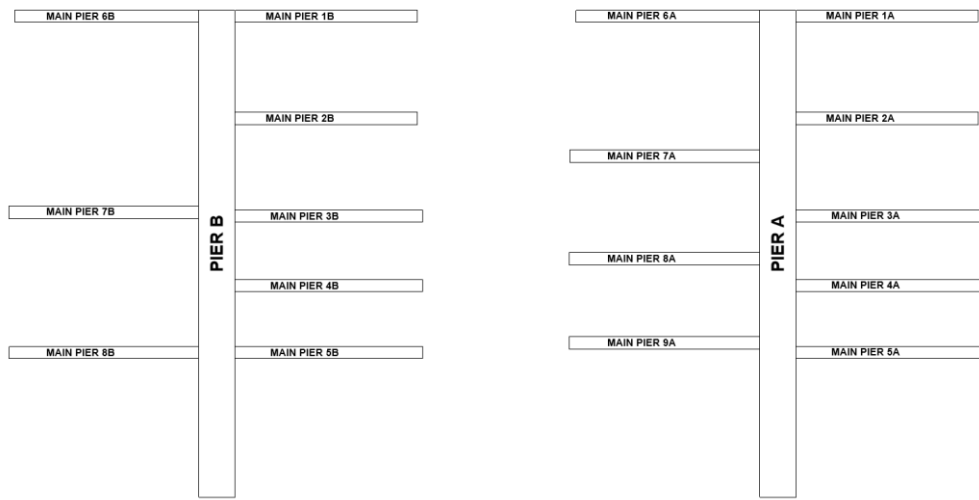
The code check results are given on “Berthing Length” page in “Mediterranean Arrangement” excel file. The result of the code check is given partially in Table 4.22.

Table 4.22 Part of The Berthing Length Results for Mediterranean Arrangement

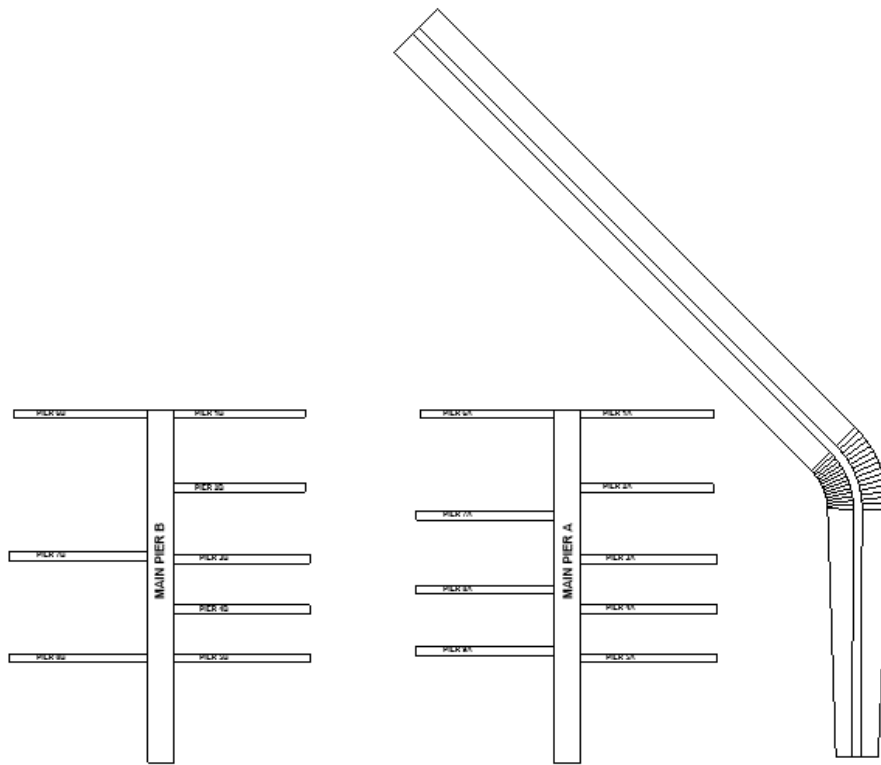
TRUE	FALSE
241980	242086
241982	242088
241984	242090
241986	242092
241988	242094
241990	242096
241992	242098
241994	242100
241996	242102
241998	242104
242000	242106
242002	242108
242004	242110
242006	242112
242008	242114
242010	242116
242012	242118
242014	242120
242016	242122

4.3 Combined Layout

The combined layout contains both finger pier and Mediterranean arrangements. The arrangement for this case is displayed in Figure 4.20. There is an extra element different from the other case: Main Pier family. After deciding the locations of main pier families, the piers are located to the right and left sides. They are placed in alphabetical order of their names. According to their assigned largest boat information, the fingers or the anchors are then arranged on the piers' sides.



(a)



(b)

Figure 4.20. Combined Layout. (a) Main Pier A and Main Pier B Layout (c)
General Layout

As in “Finger Pier Arrangement” and “Mediterranean (Anchor) Arrangement” cases, the first modeling is performed by “Placing Families on Piers” dynamo script. The incorrect design parameters are used in this initial modeling. Table 4.23 provides these incorrect design parameters for the combined layout case. Table 4.24 also gives the boat distribution and the total capacity of the marina according to the incorrect design data. On the other hand, the intended design's data is given in Table 4.25. The changed parameters in the intended design data are bolded in Table 4.23. Moreover, Table 4.26 shows the boat distribution and the total capacity of the marina according to the intended design data.

Table 4.23 Incorrect Design Data for Combined Layout

Piers	<i>Max Boat Length (m)</i>	<i>Max Boat Width (m)</i>	<i>Max Boat Draft (m)</i>	<i>Finger Pier Placement</i>	<i>Berth Number</i>	<i>Boat Capacity</i>
Pier 1A	12.00	4.10	1.20	Yes	1	48
Pier 2A	9.00	3.60	1.00	Yes	1	56
Pier 3A	5.00	2.20	0.80	No	N/A	46
Pier 4A	5.00	2.20	0.80	No	N/A	46
Pier 5A	5.00	2.20	0.80	No	N/A	46
Pier 6A	12.00	4.10	1.20	Yes	2	48
Pier 7A	9.00	3.60	1.00	No	N/A	36
Pier 8A	12.00	4.10	1.20	No	N/A	34
Pier 9A	15.00	4.80	1.40	No	N/A	32
Pier 1B	15.00	4.80	1.40	Yes	1	38
Pier 2B	15.00	4.80	1.40	Yes	2	40
Pier 3B	9.00	3.60	1.00	No	N/A	36
Pier 4B	5.00	2.20	0.80	No	N/A	44
Pier 5B	9.00	3.60	1.00	Yes	1	56
Pier 6B	65.00	10.20	3.20	Yes	1	14
Pier 7B	35.00	6.80	2.50	Yes	1	22

Table 4.24 Incorrect Design Data for Combined Layout (continued)

Piers	<i>Max Boat Length (m)</i>	<i>Max Boat Width (m)</i>	<i>Max Boat Draft (m)</i>	<i>Finger Pier Placement</i>	<i>Berth Number</i>	<i>Boat Capacity</i>
Pier 8B	12.00	4.10	1.20	Yes	2	48

Table 4.25 Marina Capacity According to Incorrect Design Data for Combined Layout

Boat Length (m)	# of Boats
5.00	182
9.00	184
12.00	178
15.00	110
35.00	22
65.00	14
Total	690

Table 4.26 Pier Parameter Values for Combined Layout

Piers	<i>Max Boat Length (m)</i>	<i>Max Boat Width (m)</i>	<i>Max Boat Draft (m)</i>	<i>Finger Pier Placement</i>	<i>Berth Number</i>	<i>Boat Capacity</i>
Pier 1A	20.00	5.30	1.66	Yes	1	34
Pier 2A	12.00	4.10	1.20	Yes	1	8
Pier 3A	5.00	2.20	0.80	No	N/A	46
Pier 4A	5.00	2.20	0.80	No	N/A	46
Pier 5A	5.00	2.20	0.80	No	N/A	46

Table 4.27 Pier Parameter Values for Combined Layout (continued)

Piers	<i>Max Boat Length (m)</i>	<i>Max Boat Width (m)</i>	<i>Max Boat Draft (m)</i>	<i>Finger Pier Placement</i>	<i>Berth Number</i>	<i>Boat Capacity</i>
Pier 6A	20.00	5.30	1.66	Yes	2	36
Pier 7A	15.00	4.80	1.40	No	N/A	32
Pier 8A	12.00	4.10	1.20	No	N/A	34
Pier 9A	20.00	5.30	1.66	No	N/A	30
Pier 1B	20.00	5.30	1.66	Yes	1	34
Pier 2B	15.00	4.80	1.40	Yes	2	40
Pier 3B	12.00	4.10	1.20	No	N/A	34
Pier 4B	9.00	3.60	1.00	No	N/A	36
Pier 5B	9.00	3.60	1.00	Yes	1	56
Pier 6B	65.00	10.20	3.20	Yes	1	14
Pier 7B	35.00	6.80	2.50	Yes	1	22
Pier 8B	20.00	5.30	1.66	Yes	2	36

Table 4.28 Marina Capacity According to Intended Design Data for Combined Layout

<i>Boat Length (m)</i>	<i># of Boats</i>
5.00	138
9.00	92
12.00	116
15.00	72
20.00	170
35.00	22
65.00	14

Table 4.29 Marina Capacity According to Intended Design Data for Combined Layout (continued)

Boat Length (m)	<i># of Boats</i>
Total	624

“Max Boat Length”, “Max Boat Width”, and “Max Boat Draft” parameters are also specified for boats, anchors, and finger piers in the model during the placement process. The values of the parameters are the same as the piers placed on for the boats and the anchors. Moreover, “Berthing Length” values are assigned to finger piers and anchors.

The code compliance checks are performed, and their results are given in the following sub-sections.

4.3.1 Slip (Berth) Width Distance

The distances between two anchors and boats are controlled by “Slip (Berth) Distance (W)”. Figure 4.21 shows the result of the code check. The red elements do not satisfy the minimum slip distance criteria.

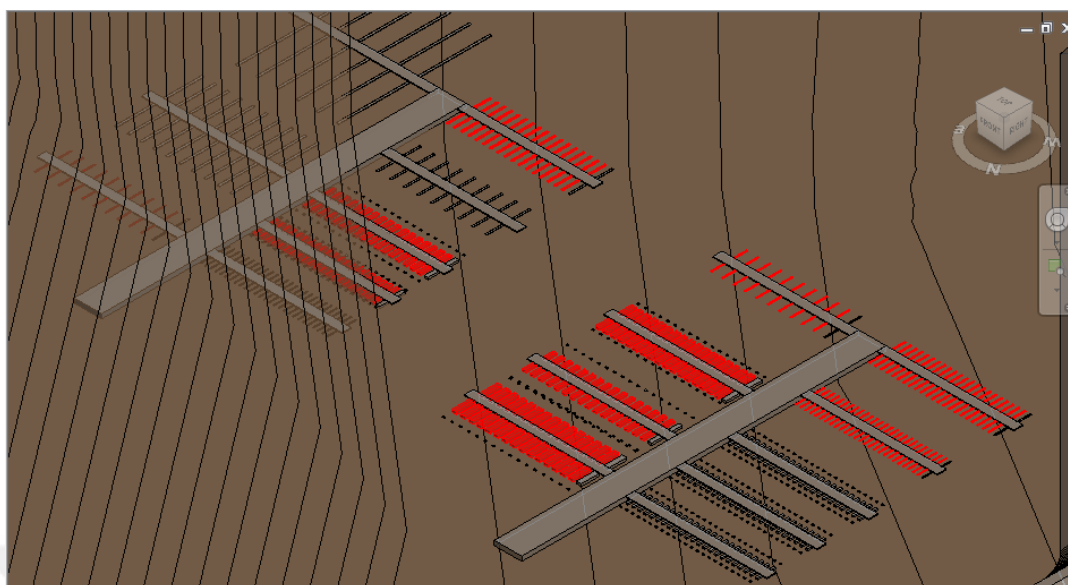


Figure 4.21. Slip (Berth) Distance Rule Checking Result for Combined Layout

Table 4.27 provides slip distances measured in the model and calculated ones using intended design data for better comparison.

Table 4.30 Measured and Calculated Slip Distanced for Combined Layout

Piers	<i>Using Incorrect Design</i>	<i>Using Intended Design</i>
	<i>Data (m)</i>	<i>Data (m)</i>
Pier 1A	4.70	5.90
Pier 2A	4.20	4.70
Pier 3A	4.20	4.20
Pier 4A	4.20	4.20
Pier 5A	4.20	4.20
Pier 6A	10.20	12.60
Pier 7A	5.60	6.80
Pier 8A	6.10	6.10
Pier 9A	6.80	7.30
Pier 1B	5.40	5.90
Pier 2B	11.60	11.60

Table 4.31 Calculated Slip Distanced For Incorrect Design and Intended Design for Combined Layout (continued)

Piers	<i>Using Incorrect Design Data (m)</i>	<i>Using Intended Design Data (m)</i>
Pier 3B	5.60	6.10
Pier 4B	4.20	5.60
Pier 5B	4.20	4.20
Pier 6B	16.50	16.50
Pier 7B	10.00	10.00
Pier 8B	10.20	12.60

The results, a part of which is given in Table 4.28 is exported to “Slip Distance” sheet of “Combined Arrangement” excel file.

Table 4.32 Part of “Slip Distance” Results for Combined Arrangement Layout

IDs	Elements	Distance	Min.Distance	T/F
261855	Boat	4.26	4.2	TRUE
261857	Boat	4.26	4.2	TRUE
261859	Boat	4.26	4.2	TRUE
261861	Boat	4.26	4.2	TRUE
261863	Boat	4.26	4.2	TRUE
261865	Boat	4.26	4.2	TRUE
261867	Boat	4.26	4.2	TRUE
261869	Boat	4.26	4.2	TRUE
261871	Boat	4.26	4.2	TRUE
261873	Boat	4.26	4.2	TRUE
261875	Boat	4.26	4.2	TRUE
261877	Boat	4.26	4.2	TRUE
261879	Boat	4.26	4.2	TRUE
261881	Boat	4.26	4.2	TRUE
261883	Boat	4.26	4.2	TRUE
261885	Boat	4.26	4.2	TRUE

Table 4.33 Part of “Slip Distance” Results For Combined Arrangement Layout
(continued)

IDs	Elements	Distance	Min.Distance	T/F
261887	Boat	4.26	4.2	TRUE
261889	Boat	4.26	4.2	TRUE
261891	Boat	4.26	4.2	TRUE
261893	Boat	4.26	4.2	TRUE
261895	Boat	4.26	4.2	TRUE
261897	Boat	4.26	4.2	TRUE
262085	Boat	2.92	5.6	FALSE
262087	Boat	2.92	5.6	FALSE
262089	Boat	2.92	5.6	FALSE
262091	Boat	2.92	5.6	FALSE
262093	Boat	2.92	5.6	FALSE
262095	Boat	2.92	5.6	FALSE
262097	Boat	2.92	5.6	FALSE

4.3.2 Inner Channel Width (B_1)

For combined layout, “Inner Channel Width (Combined)” dynamo script is used. Inner Channel-1 and Inner Channel-2, shown in Figure 3.48, are analyzed in two separate node flow in this script. The distances between the elements that specify the Inner Channel-1 area is larger than the required minimum distances. Therefore, there is no result to be presented for this area.

Main Pier A and Main Pier B are selected by two separate nodes to analyze the Inner Channel-2 area. The result is displayed in Figure 4.22.

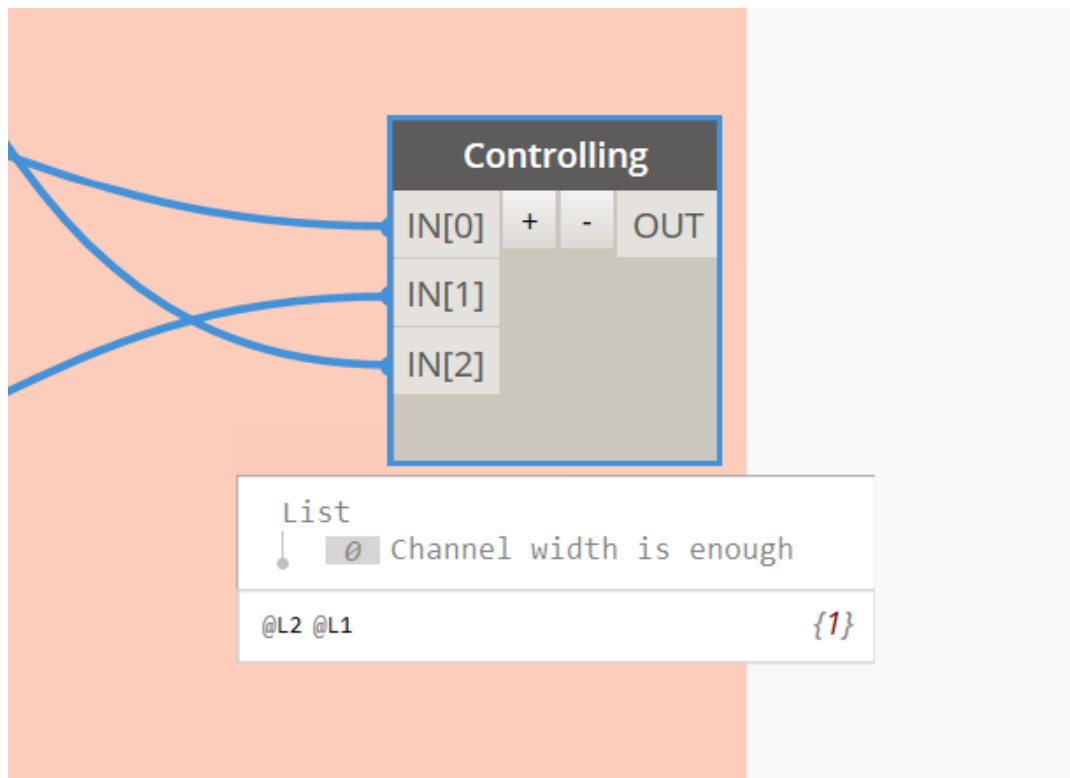


Figure 4.22. Inner Channel-2 Code Check Result for Combined Layout

4.3.3 Distance Between Piers

The distance between the two piers is checked with “Distance Between Piers (S) (Combined)” dynamo script. Figure 4.23 shows the result of the code checks. The blue piers do not provide enough distance between themselves and the next one in alphabetical order.

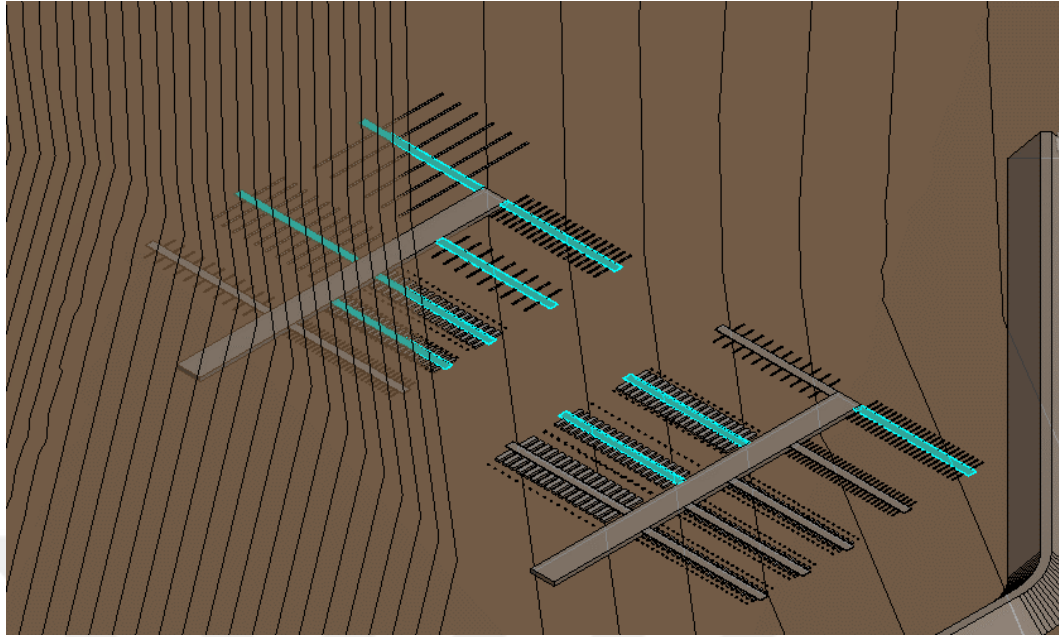


Figure 4.23. Distance Between Piers Rule Checking Result for Combined Layout

As seen in Table 4.29, some measurements are out of range. The reason for this is that the piers are placed in the model in alphabetical order. Therefore, their ID numbers are assigned according to that order. The code lists the elements by their IDs. It measures the distance between an element and the next element in the list. In other words, at some point, the code measures the distance between Pier 5A and Pier 6A, which is given as 267.68 m in Table 4.29.

On the other hand, this out of range values is larger than the calculated required distance according to design values of Pier 5A and Pier 6A. Thus, at the end of the comparison process, it falls into the “true” category. Even though it is the case, the user should be aware of this situation during the controlling process.

Additionally, the result of this check is exported to “S” sheet in “Combined Arrangement” excel file. The results are given in Table 4.29.

Table 4.34 Results of Distance Between Piers in Combined Arrangement Layout

IDs	Elements	Distance	Min.Distance	T/F
238128	Pier	74	78.4	FALSE
239162	Pier	70	48.4	TRUE
239163	Pier	47	30	TRUE
239164	Pier	45	30	TRUE
239165	Pier	267.68	74	TRUE
239166	Pier	105	94	TRUE
239167	Pier	74	84	FALSE
239168	Pier	59	104	FALSE
239169	Pier	0	0	TRUE
239170	Pier	74	82	FALSE
239171	Pier	70	72	FALSE
239172	Pier	47	66	FALSE
239173	Pier	45	46.8	FALSE
239174	Pier	267.68	218.8	TRUE
239175	Pier	151.01	250	FALSE
239176	Pier	105	136	FALSE
239177	Pier	0	0	TRUE

4.3.4 Water Level Check

“Water Level Check” dynamo script is used to control the water level in the marina area. However, the layout is changed due to the computer's lack of technical capacity used for the code check. As shown in Figure 4.24, Main Pier B is completely removed, and Main Pier A is relocated through the coastal line.

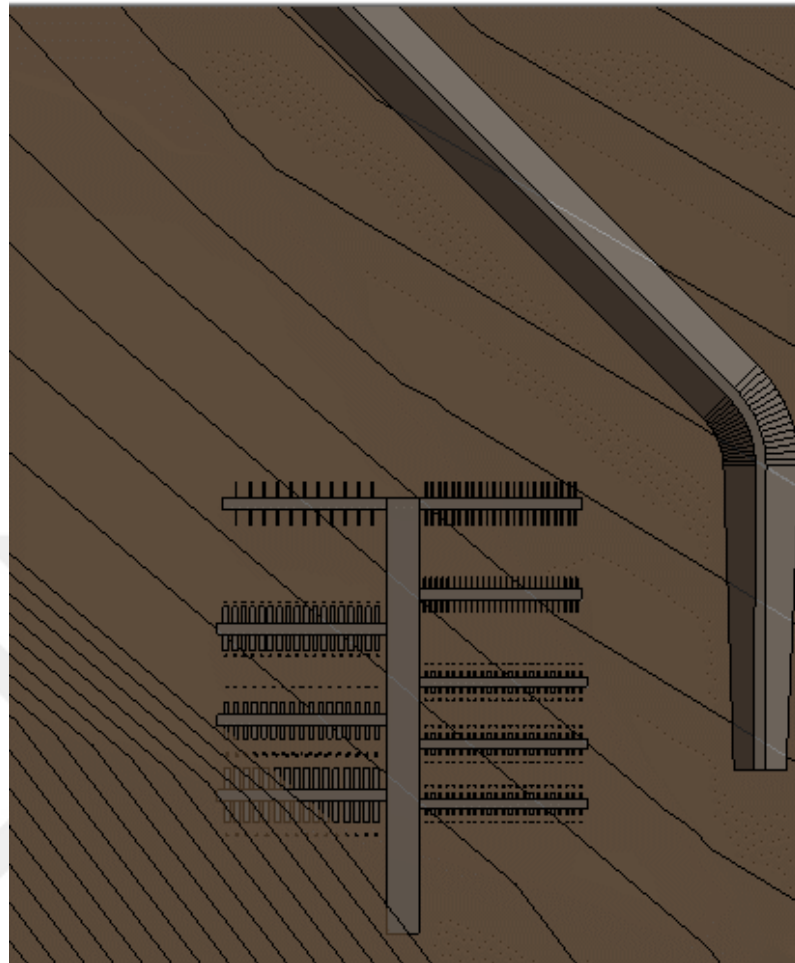


Figure 4.24. Combine Layout Plan for Water Level Check

The yellow elements in Figure 4.25 do not satisfy the minimum water level requirement. However, some elements in yellow are expected to satisfy the minimum water level. The explanation about the same situation in the Mediterranean Arrangement is valid in the Combine Layout case. The highlighted elements in Table 4.30 are the ones that gave “null” results in Dynamo. The other elements in Table 4.30 do not have enough water level for safe berthing.

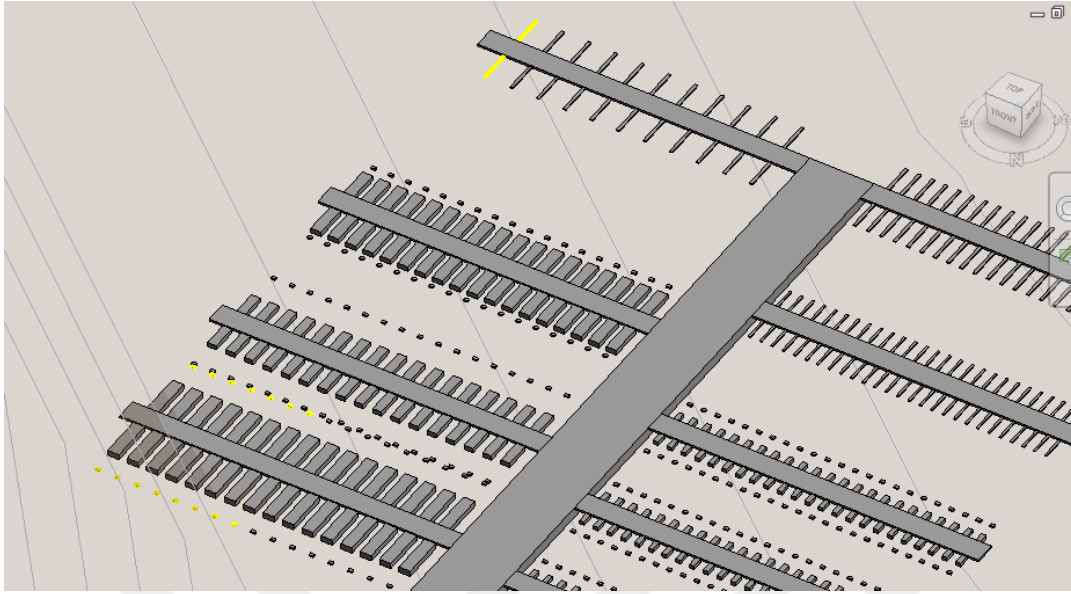


Figure 4.25. Water Level Check Result for Combined Layout

Table 4.35 Water Level Check Result for Combined Layout

IDs	Elements
262459	Anchor
262461	Anchor
262463	Anchor
262465	Anchor
262467	Anchor
262469	Anchor
262471	Anchor
262367	Finger Pier
263383	Anchor
263385	Anchor
263387	Anchor
263389	Anchor
263391	Anchor
263393	Anchor
263395	Anchor
263397	Anchor
263291	Finger Pier

The lowest points in the marina areas are given partially in Table 4.31.

Table 4.36 A Part of the Lowest Point Results for Combined Layout

Point
Point(X = -185.878, Y = 11.973, Z = 0.112)
Point(X = 489.596, Y = -655.587, Z = 1.283)
Point(X = 493.343, Y = -655.587, Z = 1.142)
Point(X = 519.762, Y = -655.587, Z = 0.144)
Point(X = 521.675, Y = -655.587, Z = 0.072)
Point(X = 547.286, Y = -655.587, Z = -0.894)
Point(X = 550.086, Y = -655.587, Z = -1.000)
Point(X = 550.386, Y = -655.587, Z = -1.011)
Point(X = 12.484, Y = -198.706, Z = 1.000)
Point(X = 24.716, Y = -190.613, Z = 0.000)
Point(X = 36.948, Y = -182.519, Z = -1.000)
Point(X = 405.241, Y = -534.364, Z = 1.000)
Point(X = 440.545, Y = -545.988, Z = 0.000)
Point(X = 475.849, Y = -557.612, Z = -1.000)
Point(X = 208.862, Y = -366.535, Z = 1.000)
Point(X = 232.630, Y = -368.300, Z = 0.000)
Point(X = 256.398, Y = -370.065, Z = -1.000)
Point(X = -131.184, Y = -58.978, Z = 1.000)
Point(X = -114.243, Y = -55.465, Z = 0.000)
Point(X = -97.301, Y = -51.951, Z = -1.000)

4.3.5 Berthing Length (B)

Berthing length measurement is the same as both finger arrangement and anchor arrangement cases. It is controlled by “Berthing Length (B)” dynamo script. Figure 4.26 shows elements that do not satisfy the minimum berthing length requirement in magenta.

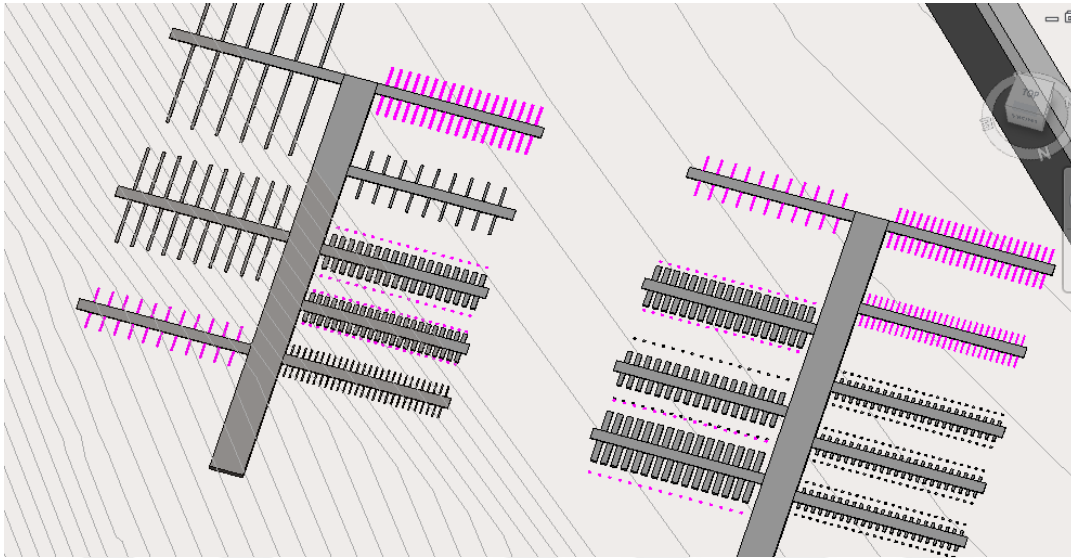


Figure 4.26. Berthing Length Rule Checking Results for Combined Layout

The code check results are given on “Berthing Length” page in “Combined Arrangement” excel file. The result of the code check is given partially in Table 4.32.

Table 4.37 Part of the Berthing Length Results for Combined Arrangement

TRUE	FALSE
262527	262129
262529	262131
262531	262133
262533	262135
262535	262137
262537	262139
262539	262141
262541	262143
262543	262145
262625	262147
262627	262149
262629	262151
262631	262153
262633	262155
262635	262157
262637	262159
262639	262161

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATIONS

In this study, a semi-automated code checking process for marina design is introduced. Besides the semi-automated code checking process, a semi-automated marina model is generated using BIM. The main purpose of this study is to provide a mechanism that can reduce difficulty in the model control and avoid non-compliance errors before finalizing a marina design.

The marina model is created by using Autodesk Civil 3D 2020 and Autodesk Revit 2018. The BIM model of the marina contains information regarding not only the geometries, locations of the elements but also, the functions of the marina elements such as the type of berthing, the geometric information of the boats that use the associated element. All of these information are the key inputs of the marina layout generation and the code checking processes. These proposed processes are constructed in Dynamo tool. Dynamo uses the berthing type and the boat information that are assigned to piers that are placed into the model manually in order to generate the marina layout automatically by placing finger piers, anchors and boats on the piers. Also, it assigns the geometric and functional information to those elements via the parameters, at the end of the placement process. Besides the marina layout generation, the dynamo scripts evaluate the layout configuration whether it is designed according to the related design guidelines. As in the layout generation process, this evaluation uses the information that is obtained from the BIM model of the marina. Coastal Structures Planning and Design Technical Manual is the main design guideline for this study. The study is focused on checking the “Slip (Berth) Distance”, “Inner Channel Width”, “Distance between Piers”, “Berthing Length”, and the “Water Level” in a marina.

The proposed semi-automated code checking process is applied to three different marina layout cases to evaluate the process. In each case, the proposed process managed to report the results both in excel form and visually. The visualization of the code checks enables the user to identify the modeling mistakes instantly. Moreover, each rule check performed in minutes. Therefore, the user can save time and labor.

On the other hand, as less time and fewer resources spent on the design evaluation process, more design alternatives can be generated and evaluated. Also, “Placing Families on Piers” dynamo script contributes to creating design alternatives in a shorter time than the manual design creation. The required spaces or the distances are calculated automatically by using the predefined design parameters. Thus, the human errors in calculation and modeling stages are expected to be minimized.

In conclusion, the proposed semi-automated code checking process can help the authorities who used to control the code compliance of a marina model manually for detecting the design mistakes in a shorter time. Thus, the authorities can save time and labor. Also, even though the person who performs the manual code compliance has a high level of knowledge, experience, and skills, the process is still error-prone. For this reason, semi-automated code compliance checking can eliminate the need for highly experienced and skilled labor and also the risk of missed nonconformance. In addition, “Placing Families on Piers” dynamo script, which provides the opportunity to generate marina layouts automatically, can contribute the design process by decreasing the error margin of the designers in calculation and modeling steps during the design procedures. This dynamo script does not affect the inner channel, the distance between piers and water level in the marina area. The designer can use the semi-automated code checking method to make sure that these aspects of the design are performed properly. In other words, the designer can use both of the proposed marina layout generation and the code checking processes to improve and optimize the marina layout design. On the other hand, there are some limitations of the proposed study. The first one is about modeling steps. There is a point that the user should be aware of during the modeling stage in Revit environment. It is

especially important in “Distance Between Piers” control. The elements are controlled in the ascending order of their element IDs. The element IDs are assigned ascendingly to each element when they are placed in the model. Thus, the modeling order of the elements should be arranged so that their controlling order will be correct.

The second limitation is the lack of processor capacity of the computer which is used in this study. The computer's processor capacity that is used for code compliance checking does not allow to perform full capacity modeling and code checking processes. The initial bathymetry created in Civil 3D covers a larger area. However, the computer could not transform the whole Civil 3D bathymetry into Revit topography element. Therefore, the model was bounded by a small area. This reduction and the transportation of the model from Civil 3D to Revit causes losing some part of the bathymetry information. Then, “Water Level Check” dynamo script could not process the whole bathymetry in Revit environment. For this check, a smaller sub-region is created, and it is used for water level control. All these reductions are resulted in loss of some bathymetry points that are necessary for more accurate results.

Additionally, “Combined Layout” case has a higher number of elements than the other two cases. The software cannot handle all of the elements for “Water Level Check”. Thus, Pier B is removed in “Combined Layout” for this rule check this case.

Another issue that the users should be conscious about is related to the calculations made in the python codes in Dynamo nodes. The design guideline gives ranges for the values of the factors affecting the design, such as the boat length or the coefficients in the distance formulas. In this study, the highest values in these ranges are generally used to be on the safe side.

The tendency to select the highest values is not followed in every case. In some cases, like “Distance Between Piers” rule check, the user can specify the coefficients. Another exception is in the “Water Level Check”. The user specifies the underkeel clearance value according to the wave analysis or her/his preferences. Besides, all

codes can be manipulated according to the user preferences easily. The coefficients or the calculation methods can be changed in the codes by users to create more suitable code compliance checking for their marina models. In addition, the flexibility of the change in codes enables to implement other national or international guidelines for the code compliance.

This study can be considered an initial attempt to create a BIM model of a marina project and automate the marina layout generation and the code checking process in the marina projects. This semi-automated code checking process can be improved by using better software technology and with better coding language knowledge. Moreover, more rules mentioned in the guideline and other national and the international guidelines can be implemented in this process. The study's promising results show that the code compliance for the marina projects can be fully automated in the future. The optimum design can be achieved for the marinas by using generative design tools for the different boat distributions. Also, these proposed processes can be improved and used for the future needs of the marinas such as upgrading the small craft marinas into the super-yacht marinas or modifying the marina configurations to meet the needs of the changing boat distribution in time. This study focuses on the marina layout generation and the code compliance of a marina information model. Therefore, the BIM of the marina is generated according to these purposes. However, the marina contains more than the berthing structures. It consists of the infrastructure to provide services such as electric, water, fuel, etc. Thus, these aspects of the marina project can be implemented into the BIM model of the marinas in the future. Additionally, this fully integrated model of a marina can be used in the operation phase. It enables to manage daily operations such as keeping track of the empty berthing places and arranging and planning of those spaces according to the customer demands.

REFERENCES

- Akbas, Y. S. (2019). *A Formal Representation of Building Codes to Facilitate BIM-Based Automated Code-Checking* (Master's Thesis). Middle East Technical University, Ankara, Turkey.
<https://etd.lib.metu.edu.tr/upload/12624470/index.pdf>
- American Society of Civil Engineers (ASCE). (2012). *Planning and Design Guidelines for Small Craft Harbors (ASCE Manual and Reports on Engineering Practice)* (Third Edit, Issue 50). American Society of Civil Engineers.
- Autodesk Knowledge Network. (n.d.). Retrived January 2021, from <https://knowledge.autodesk.com/support/revit-products>
- Bilski, M. (2015). Selected Human Factors in Marina Design. *Procedia Manufacturing*, 1646–1653. <https://doi.org/10.1016/j.promfg.2015.07.482>
- BST Associates. (2020, January). *Port of Grays Harbor Westport Marina Demand Analysis Draft Report*. Port of Grays Harbor.
<https://portofgraysharbor.com/news/2020/MarinaMarketAnalysisFinal.pdf> .
- Cheng, J. C. P., Lu, Q., & Deng, Y. (2016). Analytical review and evaluation of civil information modeling. *Automation in Construction*, 67, 31–47.
<https://doi.org/10.1016/j.autcon.2016.02.006>
- City of Melbourne & VicUrban. (2011, March) *Docklands Waterways Vessel Traffic Study And Related Strategic Documents*.<https://www.melbourne.vic.gov.au/about-council/committees-meetings/meeting-archive/MeetingAgendaItemAttachments/524/8816/5.1%20ccl.pdf>.
- Collery, D. D., McCloud, A., Deakin, V., & Ennos, J. (2016, August 29-30). Clacton and Holland on Sea Coastal Protection Case Study to Assess The Pros and Cons of Using BIM on Coastal Protection Schemes [Presentation]. *Civil Engineering Research in Ireland (CERI)*, Galway, Ireland.
<http://programme.exordo.com/ceri2016/delegates/presentation/93/>
- Costin, A., Adibfar, A., Hu, H., & Chen, S. S. (2018). Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations. *Automation in Construction*, 94, 257–281. <https://doi.org/10.1016/j.autcon.2018.07.001>
- Dodge Data & Analytics. (2017). *The Business Value of BIM for Infrastructure 2017 SmartMarket Report*.
<https://www2.deloitte.com/content/dam/Deloitte/us/Documents/finance/us-fas-bim-infrastructure.pdf>
- Eastman, C, Lee, J., Jeong, Y., & Lee, J. (2009). Automatic Rule-Based Checking

- of Building Designs. *Automation in Construction*, 18(8), 1011–1033. <https://doi.org/10.1016/j.autcon.2009.07.002>
- Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2008). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors* (1st ed.). Wiley. <https://www.wiley.com/go/permissions>
- El-Diraby, T. (2019). Beyond e-permitting : Framing the Business Case for Automated Rule Checking in AEC in the Era of Big Data (TR 1012). buildingSMART. <https://www.buildingsmart.org/wp-content/uploads/2020/06/buildingSMART-RR-TR1012-Framing-the-Business-Case-for-Automated-Rule-Checking-v1.1-Final-Dec-2019.pdf>
- Garibin, P., & Ol'Khovik, E. (2018). Development of requirements for the BIM model of the Arctic port facilities exploitation. In I. Ilin & O. Kalinina (Eds.), *MATEC Web of Conferences*. EDP Sciences. <https://doi.org/10.1051/mateconf/201817003027>
- Gucma, L., Drwięga, K., & Butrymowicz, R. (2018). Statistical Analysis of Parameters of Selected Worldwide Yacht Ports and Marinas in Terms of Design Guidelines. *Annual of Navigation*, 25(1), 205–217. <https://doi.org/10.1515/aon-2018-0014>
- Hua, X. (2020). Application of BIM Technology in Port Breakwater Construction. *Journal of Coastal Research*, 103(sp1), 548. <https://doi.org/10.2112/si103-111.1>
- Ismail, A. S., Ali, K. N., & Iahad, N. A. (2017). A Review on BIM-based automated code compliance checking system. *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*. <https://doi.org/10.1109/ICRIIS.2017.8002486>
- Lee, G., Park, J., Won, J., Park, H. K., Uhm, M., & Lee, Y. (2016, November). Can an Experience Overcome the Cognitive Challenges in Drawing-Based Design Review? – Design Review Experiments. In J.C.P. Cheng, N. Dawood, & J.S. Kuang (Eds.), *Proceedings of the 16th International Conference on Construction Applications of Virtual Reality (CNVR 2016)*, (pp.144–156). The Hong Kong University of Science and Technology.
- Lee, G., Sacks, R., & Eastman, C. M. (2006). Specifying parametric building object behavior (BOB) for a building information modeling system. *Automation in Construction*, 15(6), 758–776. <https://doi.org/10.1016/j.autcon.2005.09.009>
- Lee, Y. C., Eastman, C. M., & Lee, J. K. (2015, June). Automated Rule-Based Checking for the Validation of Accessibility and Visibility of a Building Information Model. *Computing in Civil Engineering 2015*. <https://doi.org/10.1061/9780784479247.071>
- MacKenzie, E. (2016, February). [Med-mooring]. CURISING WORLD.

<https://www.cruisingworld.com/med-mooring-made-easy/>

- Mahdi, I. M., Mohamedien, M., Ibrahim, H. M.H., & Khalil, M.A.E.M. (2019). Proposed Management System of Marine Works Based On BIM Approach (Technology). *Journal of Engineering and Applied Science*, 66(6), 771–790. <https://www.researchgate.net/publication/335455461>
- Martins, J. P., Rangel, B., & Abrantes, V. (2016). Automated Rule-Checking - A Tool For Design Development. *41st IAHS WORLD CONGRESS Sustainability and Innovation for the Future, September*.
- [Megayachts]. (2020). Yalıkavak Marina. <https://yalikavakmarina.com.tr/yalikavak-marina-hosts-the-first-mega-yachts-of-the-season/>
- National Institute of Building Sciences. (n.d.). *National BIM Standard-United States*. Retrieved February 4, 2021, from <https://www.nationalbimstandard.org/about>
- Nawari, N. O. (2013). SmartCodes and BIM. *Structures Congress 2013*, 928–937. <https://doi.org/10.1061/9780784412848.082>
- PIANC.(2016). Guidelines for Marina Design (Report No:149/Part I-2016). Association for Waterbone Transportation Infrastructure. <https://www.pianc.org/>
- Preidel C., Borrmann A. (2018) BIM-Based Code Compliance Checking. In: Borrmann A., König M., Koch C., Beetz J. (eds) *Building Information Modeling*. Springer, Cham. https://doi.org/10.1007/978-3-319-92862-3_22
- Sakikhales, M. H., & Stravoravdis, S. (2015, September). Using BIM to facilitate iterative design. *Building Information Modelling (BIM) in Design, Construction and Operations*, 9–19. <https://doi.org/10.2495/BIM150021>
- The GEBCO (The General Bathymetric Chart of the Ocean). (2014). *The GEBCO_2014 Grid* (Version 20150318) [Dataset]. <https://www.gebco.net/>
- T.C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı Altyapı Yatırımları Genel Müdürlüğü (AYGM). (2016). *Kıyı Yapıları: Planlama ve Tasarım Teknik Esasları*. http://www.ubak.gov.tr/BLSM_WIYS/DLH/tr/DOKUMANLAR/20151002_151304_10288_1_64.pdf
- Tobiasson, B. O., & Kollmeyer, R. C. (1991). *Marinas and Small Craft Harbors*. Springer. <https://doi.org/10.1007/978.1.4757.0943.8>
- Visual scripting environment for designers | Dynamo*.(2020, June 15). Dynamo BIM. <https://dynamobim.org/>
- Wahab, R., Zayed, T., Goh, T., & Wang, C. M. (2011). Sustainability and Maintainability of Port Infrastructure Using BIM / 4D CAD Simulation Model. *Proceedings of International Maritime-Port Technology and Development*

Conference 2011 (MTCE2011).

Warren, D. (2019). Parametric Modelling in Construction: Investigating The Quality Of Rule-Based Checking. *Building Information Modelling (BIM) in Design, Construction and Operations III*, 1, 57–68.
<https://doi.org/10.2495/BIM190061>



APPENDICIES

A. SLIP (BERTH) DISTANCE (W) DYNAMO SCRIPT AND PYTHON CODES

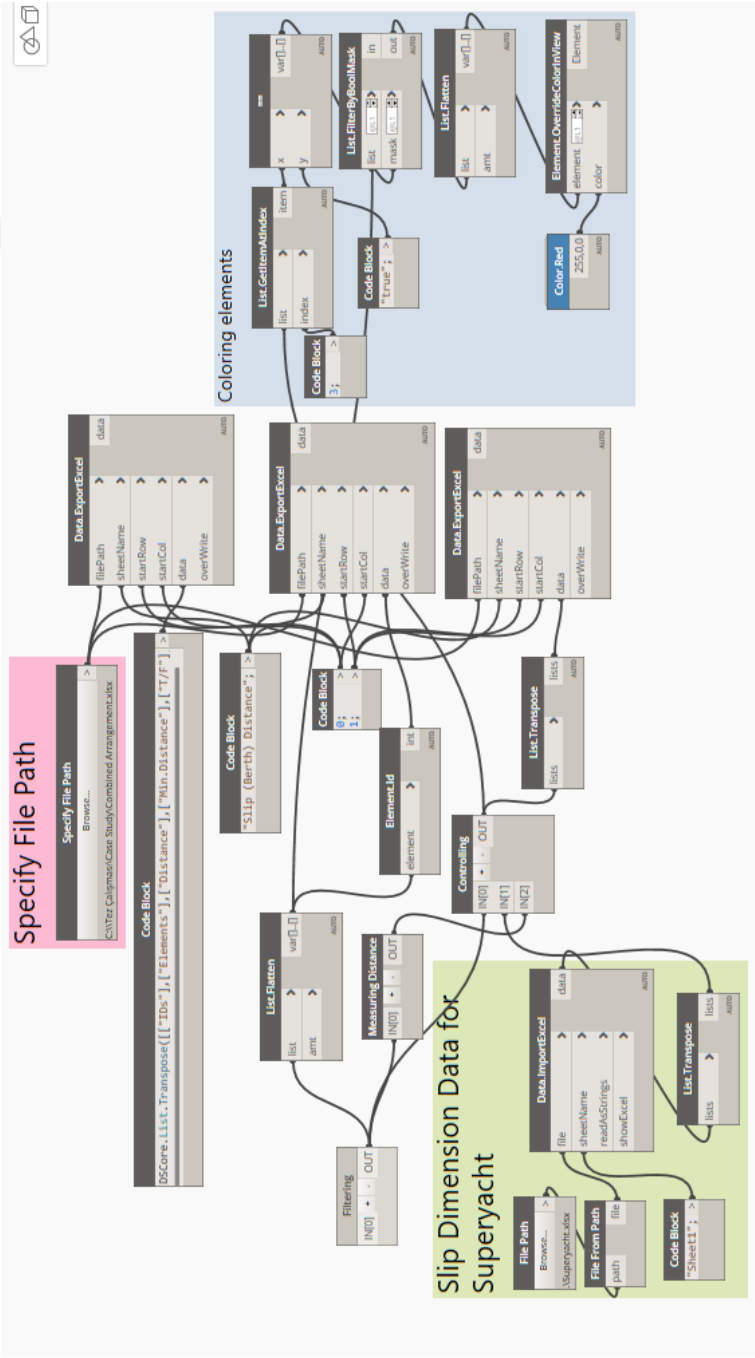


Figure A.1. Slip (Berth) Distance (W) Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. clr.AddReference("RevitNodes")
7. import Revit
8. clr.ImportExtensions(Revit.Elements)
9. clr.AddReference("RevitServices")
10. import RevitServices
11. from RevitServices.Persistence import DocumentManager
12. clr.AddReference("RevitAPI")
13. import Autodesk
14. from Autodesk.Revit.DB import *
15. import System
16. from System.Collections.Generic import *
17. from Autodesk.DesignScript.Geometry import *
18. doc = DocumentManager.Instance.CurrentDBDocument
19. family_collector = FilteredElementCollector(doc).OfCategory(BuiltIn
    Category.OST_GenericModel).WhereElementIsNotElementType().ToElement
    s() #all generic models are collected
20.
21. HOST = [] #List of value of 'Host Pier' parameter
22. Pierlist = [] #List of Finger Piers/Anchors
23. listR = [] #List of Finger Piers/Anchors that have 'R' value i
    n 'R/L' parameter
24. listL= [] #List of Finger Piers/Anchors that have 'L' value i
    n 'R/L' parameter
25. hostset = [] #List where duplicate values in HOST list are remov
    ed
26. Hostgroup = [] #List that contains 'listR' and 'listL'
27.
28. #Looking families that have 'Host Pier' parameter and 'Host Pier' p
    arameter values
29. for family in family_collector:
30.     h=family.LookupParameter('Host Pier')
31.     if h:
32.         name = family.Name
33.         if name == 'Boat' or name=='Finger Pier':
34.             HOST.append(h.AsString())
35.             Pierlist.append(family)
36. hostset = set(HOST) # removes duplicates
37.
38. #Looking for finger piers/anchors that are placed 'right' side of t
    he main piers and grouping them according their 'Host Pier' paramet
    er value
39. for i in hostset:
40.     for ii in enumerate(HOST):
41.         if ii[1] == i:
42.             d=Pierlist[ii[0]].LookupParameter('R/L').AsString()
43.             if d == 'R':
44.                 listR.append(Pierlist[ii[0]])
45.         if listR != []:
46.             Hostgroup.append(listR)
47.             listR = []
48. #Looking for finger piers/anchor that are placed 'left' side of the
    main piers and grouping them according their 'Host Pier' parameter
    value

```

```

49. for i in hostset:
50.     for ii in enumerate(HOST):
51.         if ii[1] == i:
52.             d=Pierlist[ii[0]].LookupParameter('R/L').AsString()
53.             if d == 'L':
54.                 listL.append(Pierlist[ii[0]])
55.         if listL !=[]:
56.             Hostgroup.append(listL)
57.             listL = []
58. OUT = Hostgroup

```

Figure A.2. “Filtering” Code in “Slip (Berth) Distance (W)” Dynamo

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4.
5. from itertools import islice
6. import math
7. el = IN[0] #List of elements
8. Geo=[] #List of geometries of elements
9. g=[] #Flatten list of geometries
10. Distance=[] #List of distances between elements
11. d=[] #Flatten list of distances between elements
12. l=[] #Length of sublists of 'el'
13. k=[] #Length of sublists of 'Geo'
14. m=[] #list of distances of last elements in the sub-
    lists of 'el' to itself
15.
16.
17. #Obtaining geometries as flattening the list
18. for i in el:
19.     for j in i:
20.         g.append(j.Geometry())
21.
22. #Obtaining list lengths of the sub-list of 'el'
23. for i in range(len(el)):
24.     b = len(el[i])
25.     l.append(b)
26.
27. #Slicing 'g' into sublist as list of 'Geo'
28. input=iter(g)
29. Geo=[list(islice(input,el)) for el in l]
30.
31. #Measuring distances between geometries and adding to list of 'd'
32. for i in range(len(Geo)):
33.     for j in range(len(Geo[i])-1):
34.         r = Geometry.DistanceTo(Geo[i][j][0],Geo[i][j+1][0])
35.         d.append(round(r,2))
36.
37. #Obtaining list lengths of the sub-list of 'Geo'
38. for i in range(len(Geo)):
39.     b = len(Geo[i])-1
40.     k.append(b)
41.
42. #Slicing 'd' into sublists as list of 'Distance'
43. input=iter(d)
44. Distance=[list(islice(input,d)) for d in k]

```

```

45.
46. OUT = Distance

```

Figure A.3. “Measuring Distance” Python Code in “Slip (Berth) Distance (W)”
Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4.
5. import math
6.
7.
8. elem = IN[0] #List of elements
9. data = IN[1] #dataset of preferred slip distances for
  r superyachts
10. list_e=[] #Flatten list of 'elem'
11. list_d=[] #Flatten list of 'distance'
12. Width=[] #List of minimum slip distances
13. distance = IN[2] #List of distances between elements
14. result = [] #list of comparison of measurements with
  h minimum berthing distance values (T/F)
15.
16. #Flattening 'elem'
17. for i in elem:
18.     i.pop(-1)
19.     for j in i:
20.         list_e.append(j)
21.
22. #Flattening 'distance'
23. for i in distance:
24.     for j in i:
25.         list_d.append(round(j,2))
26. #Function to calculate required slip distance for yachts whose leng
  ths are equal to 20 m or smaller than 20 m
27. def slip(l):
28.     name = l.Name
29.     if name == 'Finger Pier':
30.         anchoring_parameter = l.LookupParameter('Berth Number').AsI
  nteger()
31.         if anchoring_parameter == 1:
32.             boat_w = l.LookupParameter('Max Boat Width').AsDouble()
33.             W = (boat_w/3.2808)+0.6
34.             return W
35.         if anchoring_parameter == 2:
36.             boat_w = l.LookupParameter('Max Boat Width').AsDouble()
37.             W = 2*(boat_w/3.2808)+2
38.             return W
39.     if name == 'Boat':
40.         boat_w=l.LookupParameter('Max Boat Width').AsDouble()
41.         W = (boat_w/3.2808)+2
42.         return W
43. #Function to obtained the preferred slip distance for superyachts t
  hat are longer than 20 m
44. def SSlip(l,boat_l):
45.     name = l.Name

```

```

46.     if name == 'Finger Pier':
47.         anchoring_parameter = l.LookupParameter('Berth Number').AsInteger()
48.         row = data[0].index(boat_1)
49.         W = data [anchoring_parameter][row]
50.         return W
51.     if name == 'Boat':
52.         row = data[0].index(boat_1)
53.         W = data [1][row]
54.         return W
55. #Listing required slip distances
56. for l in UnwrapElement(list_e):
57.     boat_1 = round(((l.LookupParameter('Max Boat Length').AsDouble(
58.     ))/3.2808),2)
59.     if boat_1 <= 20:
60.         Width.append(round(slip(1),2))
61.     if boat_1 > 20:
62.         Width.append(round(SSlip(1,boat_1),2))
63. #Comparing the measurements with the min. berthing distances
64. for d, w in zip (list_d,Width):
65.     if d>=w:
66.         result.append("true")
67.     else: result.append("false")
68.
69.
70.
71. OUT=list_e, list_d, Width, result,

```

Figure A.4. “Controlling” Code in “Slip (Berth) Distance (W)” Dynamo

B. INNER CHANNEL WIDTH (B1) AND INNER CHANNEL WIDTH (COMBINED) DYNAMO SCRIPTS AND PYTHON CODES

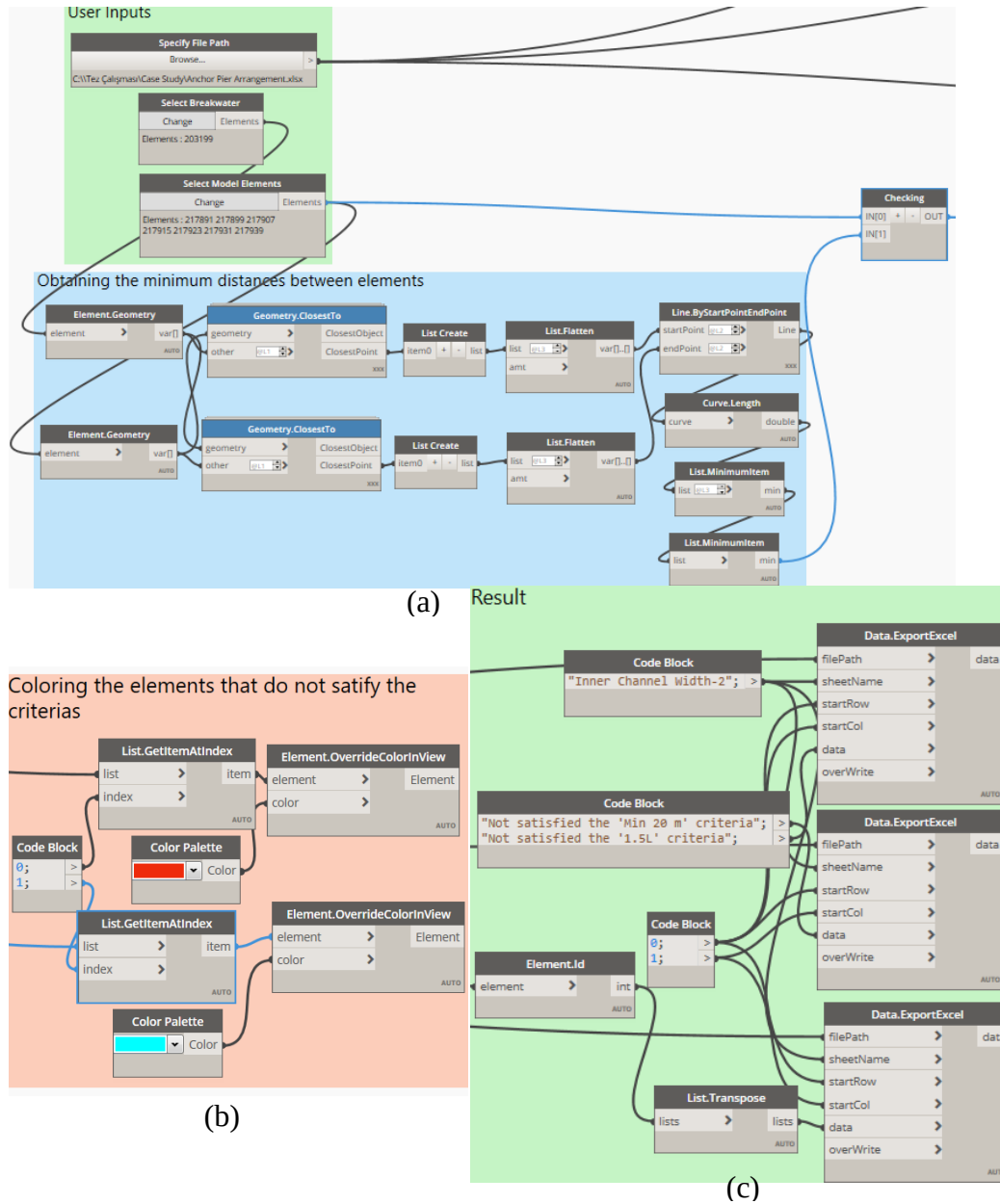


Figure B.1. Inner Channel Width (B1) Dynamo Script.(a)Node groups of user inputs and minimum distances between elements.(b)Node group of coloring the elements that do not satisfy the criteria.(c)Node group of reporting result to excel

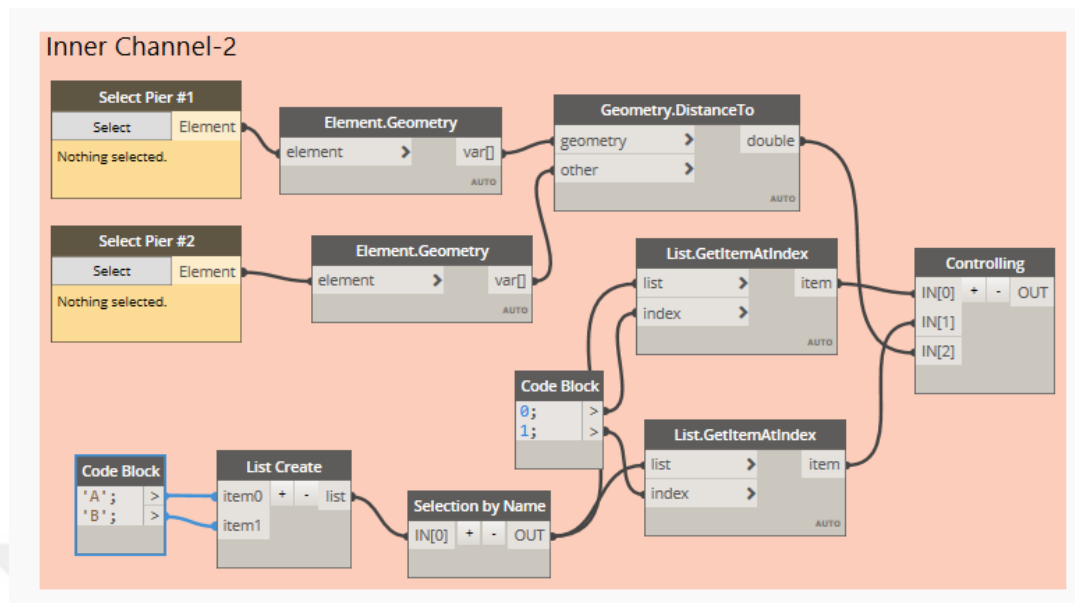


Figure B.2. Inner Channel-2 Width Control in Inner Channel Width (Combined)
Dynamo Script

```

1. import clr
2. clr.AddReference('RevitAPI')
3. from Autodesk.Revit.DB import *
4.
5. clr.AddReference('RevitServices')
6. import RevitServices
7.
8. clr.AddReference('ProtoGeometry')
9. from Autodesk.DesignScript.Geometry import *
10.
11.
12. elements = UnwrapElement(IN[0]) #list of elements that are selected
13. distance = IN[1] #list of minimum distances that are measured between selected elements
14. Boat_Length = [] #list of values that are given in parameter "Max Boat Length"
15. t = [] #list of elements that are satisfied the rule
16. F1 = [] #list of elements that are not satisfied the "min 1.5L" rule
17. F2 = [] #list of elements that are not satisfied the "min 20 m" rule
18.
19. #Listing the values of Max Boat Length parameter
20. for e in elements:
21.     b = round(((e.LookupParameter('Max Boat Length').AsDouble())/3.2808),2)
22.     Boat_Length.append(b)
23.
24. #Finding the largest value in Max Boat Length values
25. rule = max(Boat_Length)*1.5
26.

```

```

27. #Controlling whether the measured distances are satisfied the rule
28. for e, d in zip(elements,distance):
29.     if d > 20:
30.         if d > rule:
31.             t.append(e)
32.         else:
33.             F1.append(e)
34.     else:
35.         F2.append(e)
36.
37. OUT = F1, F2

```

Figure B.3. “Checking” Python Code in for Inner Channel-1 “Inner Channel Width (B1)” and “Inner Channel Width (Combined)” Dynamo Scripts

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. # import RevitNodes
7. clr.AddReference('RevitNodes')
8. import Revit
9. clr.ImportExtensions(Revit.Elements)
10. # import Revit Services
11. clr.AddReference('RevitServices')
12. import RevitServices
13. from RevitServices.Persistence import DocumentManager
14. # import Revit API
15. clr.AddReference('RevitAPI')
16. import Autodesk
17. from Autodesk.Revit.DB import *
18. # import system.
19. import System
20. from System.Collections.Generic import *
21.
22. from Autodesk.DesignScript.Geometry import *
23. import math
24.
25. doc = DocumentManager.Instance.CurrentDBDocument
26.
27. # Collecting all generic models
28. family_collector = FilteredElementCollector(doc).OfCategory(BuiltIn
    Category.OST_GenericModel).WhereElementIsNotElementType().ToElement
    s()
29.
30.
31. Piers = [] #List of piers
32. list_g=[[ for i in enumerate(IN[0])] #empty lists generated for ea
    ch pier. Empty lists contains piers that are attached to the main p
    iers
33.
34. # Searching 'Pier' word in the names of all generic models
35. for family in family_collector:

```

```

36.     n = family.Name
37.     if "Pier" == n:
38.         Piers.append(family)
39.
40. # Piers are grouped according to the values of "Pier Name" parameter
41. for p in Piers:
42.     name=p.LookupParameter("Pier Name").AsString()
43.     for i in IN[0]:
44.         if i in name:
45.             list_g[IN[0].index(i)].append(p)
46.
47.
48. OUT= list_g

```

Figure B.4.“Selection by Name” Python Code for Inner Channel-2 in “Inner Channel Width (Combined)” Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4.
5. import math
6. from itertools import islice
7.
8. elem1 = UnwrapElement(IN[0]) #List of elements
9. elem2 = UnwrapElement(IN[1]) #List of elements
10. distance = IN[2]
11. BL=[]
12. L1=[]
13. L2=[]
14. MaxBL=[]
15. MaxL1=[]
16. MaxL2=[]
17. B = []
18. s=[]
19. #Calculating total distances between main bodies of two piers( ith
    and (i+1)th)
20. for e in elem1:
21.     BL.append(e.LookupParameter('Max Boat Length').AsDouble())
22.     L1.append(e.LookupParameter('Length').AsDouble())
23.
24. for e in elem2:
25.     BL.append(e.LookupParameter('Max Boat Length').AsDouble())
26.     L2.append(e.LookupParameter('Length').AsDouble())
27.
28. MaxBL = round((max(BL)/3.2808),2)
29. MaxL1 = round((max(L1)/3.2808),2)
30. MaxL2= round((max(L2)/3.2808),2)
31.
32. B = round((distance[0]-(MaxL1+MaxL2)),2)
33.
34. if B <= 20:
35.     s.append(str('Inner Channel is smaller than 20 m.Not enough width!'))

```

```
36. elif B <= (1.5*MaxBL):  
37.     s.append(str('Inner Channel is smaller than 1.5L. Not enough wi  
    dth!'))  
38. else: s.append(str('Channel width is enough'))  
39.  
40.  
41. OUT = s
```

Figure B.5. “Controlling” Python Code for Inner Channel-2 in “Inner Channel Width (Combined)” Dynamo Script



C. DISTANCE BETWEEN TWO PIERS (S) DYNAMO SCRIPT AND PYTHON CODES

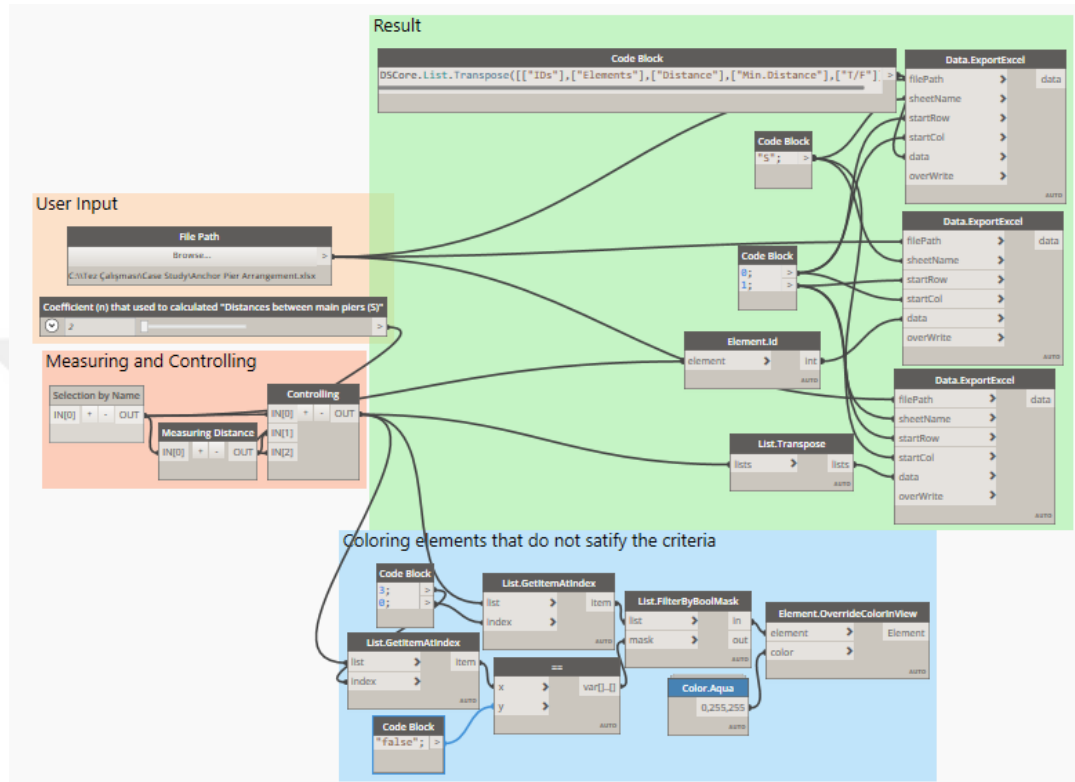


Figure C.5. Distance Between Two Piers (S) Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. # import RevitNodes
7. clr.AddReference('RevitNodes')
8. import Revit
9. clr.ImportExtensions(Revit.Elements)
10. # import Revit Services
11. clr.AddReference('RevitServices')
12. import RevitServices
13. from RevitServices.Persistence import DocumentManager
14. # import Revit API
15. clr.AddReference('RevitAPI')
16. import Autodesk
17. from Autodesk.Revit.DB import *
18. # import system.

```

```

19. import System
20. from System.Collections.Generic import *
21.
22. from Autodesk.DesignScript.Geometry import *
23.
24. import math
25.
26. doc = DocumentManager.Instance.CurrentDBDocument
27.
28. # Collecting all generic models
29. family_collector = FilteredElementCollector(doc).OfCategory(BuiltIn
    Category.OST_GenericModel).WhereElementIsNotElementType().ToElement
    s()
30.
31.
32. Piers = [] #List of piers
33.
34. # Searching 'pier' word in the names of all generic models
35. for family in family_collector:
36.     n = family.Name
37.     if " Pier" == n:
38.         Piers.append(family)
39.
40. OUT= Piers

```

Figure C.6. “Selection By Name” Python Code in “Distance Between Two Piers (S)” Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4.
5. from itertools import islice
6.
7. el = IN[0] #List of elements
8. Geo=[] #List of geometries of elements
9. Distance=[]#List of distances between elements
10.
11.
12. #Obtaining geometries
13. for i in el:
14.     Geo.append(i.Geometry())
15.
16. #Measuring distances between geometries and adding to list of 'Distance'
17. for i in range(len(Geo)-1):
18.     r = Geometry.DistanceTo(Geo[i][0],Geo[i+1][0])
19.     Distance.append(round(r,0))
20.
21.
22. #list of distances of last elements in the list of 'el' to itself
23. f = Geometry.DistanceTo(Geo[-1][0], Geo[-1][0])
24.
25.

```

```

26. #Adding values in 'f' into list of 'Distances'
27. Distance.append(f)
28.
29. OUT = Distance

```

Figure C.7. “Measuring Distance” Python Code in “Distance Between Two Piers (S)” Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4.
5. import math
6.
7. elem = UnwrapElement(IN[0]) #List of elements
8. distance = IN[1] #List of distances between elements
9. D=[] #List of total distance between main bo
    dies of two piers
10.
11. result = [] #list of comparison of measurements wit
    h minimum berthing distance values (T/F)
12.
13. #Calculating total distances between main bodies of two piers( ith
    and (i+1)th)
14. for e in range(len(elem)-1):
15.     L1 = elem[e].LookupParameter('Max Boat Length').AsDouble()
16.     L2 = elem[e+1].LookupParameter('Max Boat Length').AsDouble()
17.     BL1 = 1.2* (L1/3.2808)
18.     BL2 = 1.2* (L2/3.2808)
19.     MaxL = max(L1,L2)
20.     S = IN[2]*(MaxL/3.2808)
21.     Total = (BL1+BL2+S)
22.     D.append(round(Total,2))
23.
24.
25. #Comparing the measurements with the min. berthing distances
26. for d, w in zip (distance,D):
27.     if d>=w:
28.         result.append("true")
29.     else: result.append("false")
30.
31.
32. OUT = elem,distance,D,result

```

Figure C.8. “Controlling” Python Code in “Distance Between Two Piers (S)” Dynamo Script

D. PYTHON CODES IN DISTANCE BETWEEN TWO PIERS (COMBINED) (S) DYNAMO SCRIPT

```
1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. # import RevitNodes
7. clr.AddReference('RevitNodes')
8. import Revit
9. clr.ImportExtensions(Revit.Elements)
10. # import Revit Services
11. clr.AddReference('RevitServices')
12. import RevitServices
13. from RevitServices.Persistence import DocumentManager
14. # import Revit API
15. clr.AddReference('RevitAPI')
16. import Autodesk
17. from Autodesk.Revit.DB import *
18. # import system.
19. import System
20. from System.Collections.Generic import *
21. from Autodesk.DesignScript.Geometry import *
22. import math
23.
24. doc = DocumentManager.Instance.CurrentDBDocument
25.
26. # Collecting all generic models
27. family_collector = FilteredElementCollector(doc).OfCategory(BuiltInCategory.OST_GenericModel).WhereElementIsNotElementType().ToElements()
28.
29. Piers = [] #List of piers
30. list_g=[[ for i in enumerate(IN[0])] #empty lists generated for each pier
. Empty lists contains piers that are attached to the main piers
31.
32. # Searching 'Pier' word in the names of all generic models
33. for family in family_collector:
34.     n = family.Name
35.     if "Pier"== n:
36.         Piers.append(family)
37.
38. # Piers are grouped according to the values of "Pier Name" parameter
39. for p in Piers:
40.     name=p.LookupParameter("Pier Name").AsString()
41.     for i in IN[0]:
42.         if i in name:
43.             list_g[IN[0].index(i)].append(p)
44. OUT= list_g
```

Figure D.9. “Selection by Name” Python Code in “Distance Between Two Piers
(Combined) (S)” Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4. from itertools import islice
5.
6. el = IN[0] #List of elements
7. Geo=[]    #List of geometries of elements
8. g=[]      #Flatten list of geometries
9. Distance=[] #List of distances between elements
10. d=[]     #Flatten list of distances between elements
11. l=[]     #Length of sublists of 'el'
12. k=[]     #Length of sublists of 'Geo'
13. m=[]     #list of distances of last elements in the sub-
            #lists of 'el' to themselves
14.
15. #Obtaining geometries as flattening the list
16. for i in el:
17.     for j in i:
18.         g.append(j.Geometry())
19.
20. #Obtaining list lengths of the sub-list of 'el'
21. for i in range(len(el)):
22.     b = len(el[i])
23.     l.append(b)
24.
25. #Slicing 'g' into sublist as list of 'Geo'
26. input=iter(g)
27. Geo=[list(islice(input,el)) for el in l]
28.
29. #Measuring distances between geometries and adding to list of 'd'
30. for i in range(len(Geo)):
31.     for j in range(len(Geo[i])-1):
32.         r = Geometry.DistanceTo(Geo[i][j][0],Geo[i][j+1][0])
33.         d.append(round(r,2))
34.
35. #Obtaining list lengths of the sub-list of 'Geo'
36. for i in range(len(Geo)):
37.     b = len(Geo[i])-1
38.     k.append(round(b,0))
39.
40. #Slicing 'd' into sublists as list of 'Distance'
41. input=iter(d)
42. Distance=[list(islice(input,d)) for d in k]
43.
44. #list of distances of last elements in the sub-lists of 'el' to itself
45. for i in range(len(Geo)):
46.     f = Geometry.DistanceTo(Geo[i][-1][0], Geo[i][-1][0])
47.     m.append(round(f,2))
48.
49. #Adding values in 'm' into list of 'Distances'
50. for i,j in zip(Distance,m):
51.     i.append(j)
52.
53. OUT = Distance

```

Figure D.10. “Measuring Distance” Python Code in “Distance Between Two Piers (Combined) (S)” Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. from Autodesk.DesignScript.Geometry import *
4.
5. import math
6. from itertools import islice
7.
8. elem = UnwrapElement(IN[0])      #List of elements
9. distance = IN[1]                 #List of distances between elements
10. D=[] for i in enumerate(elem) #Generating empty list for the calculated
    total distances as the number of sublists of 'elem'
11. m=[]                             #list of distances of last elements in the
    sub-lists of 'elem' to themselves
12. me=[]                           #flatten list of 'D'
13. dis=[]                          #flatten list of 'distance'
14. result = []                     #list of comparison of measurements with m
    inimum berthing distance values (T/F)
15. l=[]                             #list lengths of the sub-list of 'Geo'
16. r=[]                             #Sliced list of 'results'
17.
18. #Calculating total distances between main bodies of two piers( ith and (i+
    1)th)
19. for e in range(len(elem)):
20.     for k in range(len(elem[e])-1):
21.         L1 = elem[e][k].LookupParameter('Max Boat Length').AsDouble()
22.         finger=elem[e][k].LookupParameter('Finger Pier Placement').AsValue
            String()
23.         if finger == 'Yes':
24.             BL1 = 1.2*(L1/3.2808)
25.         else:
26.             BL1 = 2*(L1/3.2808)
27.         L2 = elem[e][k+1].LookupParameter('Max Boat Length').AsDouble()
28.         finger=elem[e][k+1].LookupParameter('Finger Pier Placement').AsVal
            ueString()
29.         if finger == 'Yes':
30.             BL2 = 1.2*(L2/3.2808)
31.         else:
32.             BL2 = 2*(L2/3.2808)
33.         MaxL = max(L1,L2)
34.         S = IN[2]*(MaxL/3.2808)
35.         Total = (BL1+BL2+S)
36.         D[e].append(round(Total,2))
37.
38. #list of distances of last elements in the sub-lists of 'el' to itself
39. for i in range(len(D)):
40.     m.append(0)
41.
42. #Adding values in 'm' into list of 'D'
43. for i,j in zip(D,m):
44.     i.append(j)
45.
46. #Flatten the list of 'D'
47. for d in D:
48.     for k in d:
49.         me.append(k)
50. #flatten the list of 'distance'
51. for i in distance:
52.     for j in i:
53.         dis.append(j)

```

```

54.
55. #Comparing the measurements with the min. berthing distances
56. for d, i in zip(me, dis):
57.     if i>=d:
58.         result.append("true")
59.     else:result.append("false")
60.
61. #Obtaining list lengths of the sub-list of 'Geo'
62. for i in range(len(D)):
63.     b = len(D[i])
64.     l.append(b)
65.
66. #Slicing 'results' into sublists as the length of sub-list of 'Geo'
67. input=iter(result)
68. r=[list(islice(input,result)) for result in l]
69.
70.
71. OUT = elem,distance,D,r

```

Figure D.11. “Controlling” Python Code in “Distance Between Two Piers
(Combined) (S)” Dynamo Script

E. WATER LEVEL CHECK DYNAMO SCRIPT AND PYTHON CODES

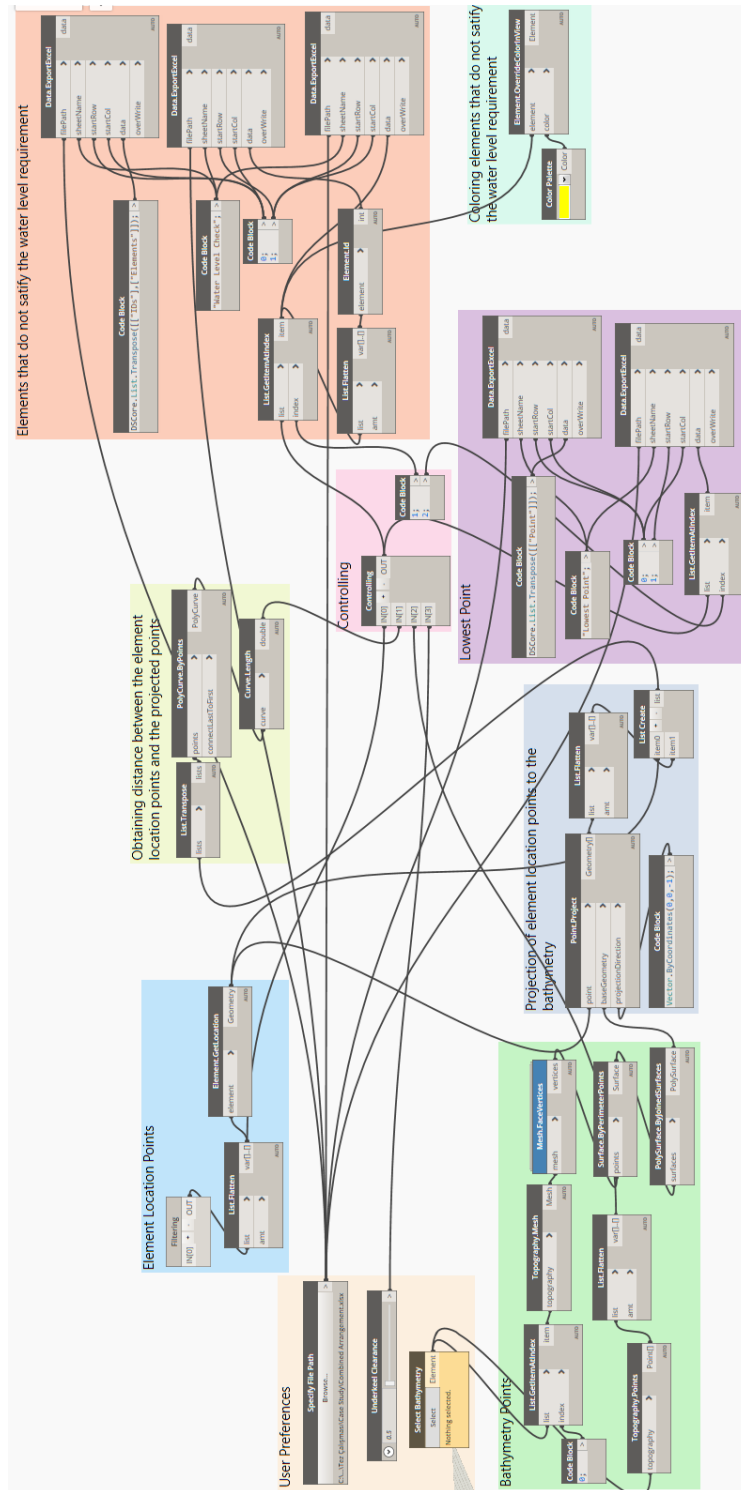


Figure E.12. Water Level Check Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. clr.AddReference("RevitNodes")
7. import Revit
8. clr.ImportExtensions(Revit.Elements)
9. clr.AddReference("RevitServices")
10. import RevitServices
11. from RevitServices.Persistence import DocumentManager
12. clr.AddReference("RevitAPI")
13. import Autodesk
14. from Autodesk.Revit.DB import *
15. import System
16. from System.Collections.Generic import *
17. from Autodesk.DesignScript.Geometry import *
18. doc = DocumentManager.Instance.CurrentDBDocument
19. family_collector = FilteredElementCollector(doc).OfCategory(BuiltIn
Category.OST_GenericModel).WhereElementIsNotElementType().ToElement
s() #all generic models are collected
20.
21. HOST = [] #List of value of 'Host Pier' parameter
22. Pierlist = [] #List of Finger Piers/Anchors
23. listR = [] #List of Finger Piers/Anchors that have 'R' value i
n 'R/L' parameter
24. listL = [] #List of Finger Piers/Anchors that have 'L' value i
n 'R/L' parameter
25. hostset = [] #List where duplicate values in HOST list are remov
ed
26. Hostgroup = [] #List that contains 'listR' and 'listL'
27.
28. #Looking families that have 'Host Pier' parameter and 'Host Pier' p
arameter values
29. for family in family_collector:
30.     h=family.LookupParameter('Host Pier')
31.     if h:
32.         name = family.Name
33.         if name != 'Boat':
34.             HOST.append(h.AsString())
35.             Pierlist.append(family)
36. hostset = set(HOST) # removes duplicates
37.
38. #Looking for finger piers/anchors that are placed 'right' side of t
he main piers and grouping them according their 'Host Pier' paramet
er value
39. for i in hostset:
40.     for ii in enumerate(HOST):
41.         if ii[1] == i:
42.             d=Pierlist[ii[0]].LookupParameter('R/L').AsString()
43.             if d == 'R':
44.                 listR.append(Pierlist[ii[0]])
45.         if listR != []:
46.             Hostgroup.append(listR)
47.             listR = []
48. #Looking for finger piers/anchor that are placed 'left' side of the
main piers and grouping them according their 'Host Pier' parameter
value
49. for i in hostset:

```

```

50.     for ii in enumerate(HOST):
51.         if ii[1] == i:
52.             d=Pierlist[ii[0]].LookupParameter('R/L').AsString()
53.             if d == 'L':
54.                 listL.append(Pierlist[ii[0]])
55.         if listL !=[]:
56.             Hostgroup.append(listL)
57.             listL = []
58. OUT = Hostgroup

```

Figure E.13. “Filtering” Python Code in “Water Level Check” Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. # import RevitNodes
7. clr.AddReference('RevitNodes')
8. import Revit
9. clr.ImportExtensions(Revit.Elements)
10. # import Revit Services
11. clr.AddReference('RevitServices')
12. import RevitServices
13. from RevitServices.Persistence import DocumentManager
14. # import Revit API
15. clr.AddReference('RevitAPI')
16. import Autodesk
17. from Autodesk.Revit.DB import *
18. # import system.
19. import System
20. from System.Collections.Generic import *
21. from Autodesk.DesignScript.Geometry import *
22. import math
23.
24. elements = UnwrapElement(IN[0]) #List of elements
25. water_level = IN[1] #List of lengths of lines between t
    he location points of element and the projected points
26. points = IN[2] #List of points of bathymetry
27. drafts = [] #List of calculated minimum water l
    evel according to values of 'Draft' parameter
28. true = [] #List of elements which are satisfi
    ed the condition
29. false = [] #List of elements which are not sat
    isfied the condition
30. lowest_point=[] #List of points which are lower tha
    n minimum required water level
31.
32. #Checking whether water level below elements satisfy the minimum wat
    er level requirement
33. for e,l in zip(elements,water_level):
34.     d = IN[3]+((e.LookupParameter('Max Boat Draft').AsDouble())/3.2
        808)
35.     drafts.append(d)
36.     if l >= d:

```

```
37.         true.append(e)
38.     else: false.append(e)
39.
40. #Finding bathymetry points which don't satisfy the minimum water level
41. for point in IN[2]:
42.     z = point.Z
43.     if abs(z) < min(drafts):
44.         lowest_point.append(point)
45.
46. OUT=true, false, lowest_point
```

Figure E.14. “Controlling” Python Code in “Water Level Check” Dynamo Script

F. BERTHING LENGTH (B) DYNAMO SCRIPT AND PYTHON CODES

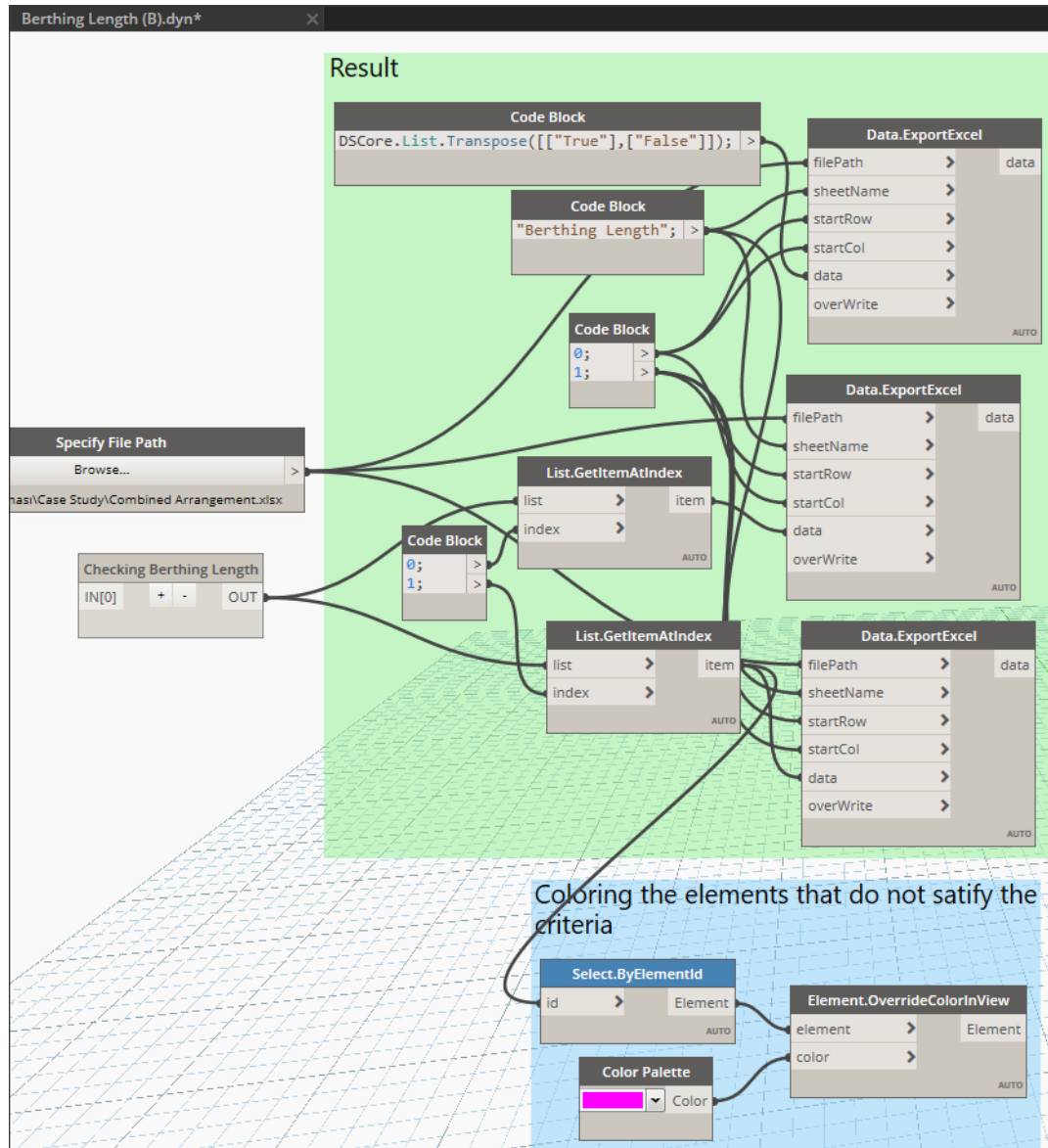


Figure F.15. Berthing Length (B) Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. # import RevitNodes
7. clr.AddReference('RevitNodes')
8. import Revit
9. clr.ImportExtensions(Revit.Elements)
10. # import Revit Services
11. clr.AddReference('RevitServices')
12. import RevitServices
13. from RevitServices.Persistence import DocumentManager
14. # import Revit API
15. clr.AddReference('RevitAPI')
16. import Autodesk
17. from Autodesk.Revit.DB import *
18. # import system.
19. import System
20. from System.Collections.Generic import *
21.
22. from Autodesk.DesignScript.Geometry import *
23.
24. import math
25.
26. doc = DocumentManager.Instance.CurrentDBDocument
27.
28. # Collecting all generic models
29. family_collector = FilteredElementCollector(doc).OfCategory(BuiltIn
    Category.OST_GenericModel).WhereElementIsNotElementType().ToElement
    s()
30.
31.
32. Finger = [] #List of Finger piers
33. Anchor = [] #List of anchorings
34. true = []
35. false = []
36.
37.
38. # Searching 'Finger Pier' and/or 'Anchor' word in the names of all
    generic models
39. for family in family_collector:
40.     n = family.Name
41.     if "Finger" in n:
42.         Finger.append(family)
43.     if 'Anchor' in n:
44.         Anchor.append(family)
45.
46. for f in Finger:
47.     boat_length = round(((f.LookupParameter('Max Boat Length').AsDo
        uble())/3.2808),2)
48.     length = round(((f.LookupParameter('Berthing Length').AsDouble(
        ))/3.2808),2)
49.     B = round((1.2* boat_length),2)
50.     if length >= B:
51.         true.append(f.Id)
52.     else: false.append(f.Id)
53.
54. for f in Anchor:

```

```
55.     boat_length = round(((f.LookupParameter('Max Boat Length').AsDouble())/3.2808),2)
56.     length = round(((f.LookupParameter('Berthing Length').AsDouble())/3.2808),2)
57.     B = round((2* boat_length),2)
58.     if length >= B:
59.         true.append(f.Id)
60.     else: false.append(f.Id)
61.
62.
63. OUT= true, false
```

Figure F.16. “Checking Berthing Length” Python Code in “Berthing Length (B)”
Dynamo Script

G. PLACING FAMILIES ON PIERS DYNAMO SCRIPT AND PYTHON CODES

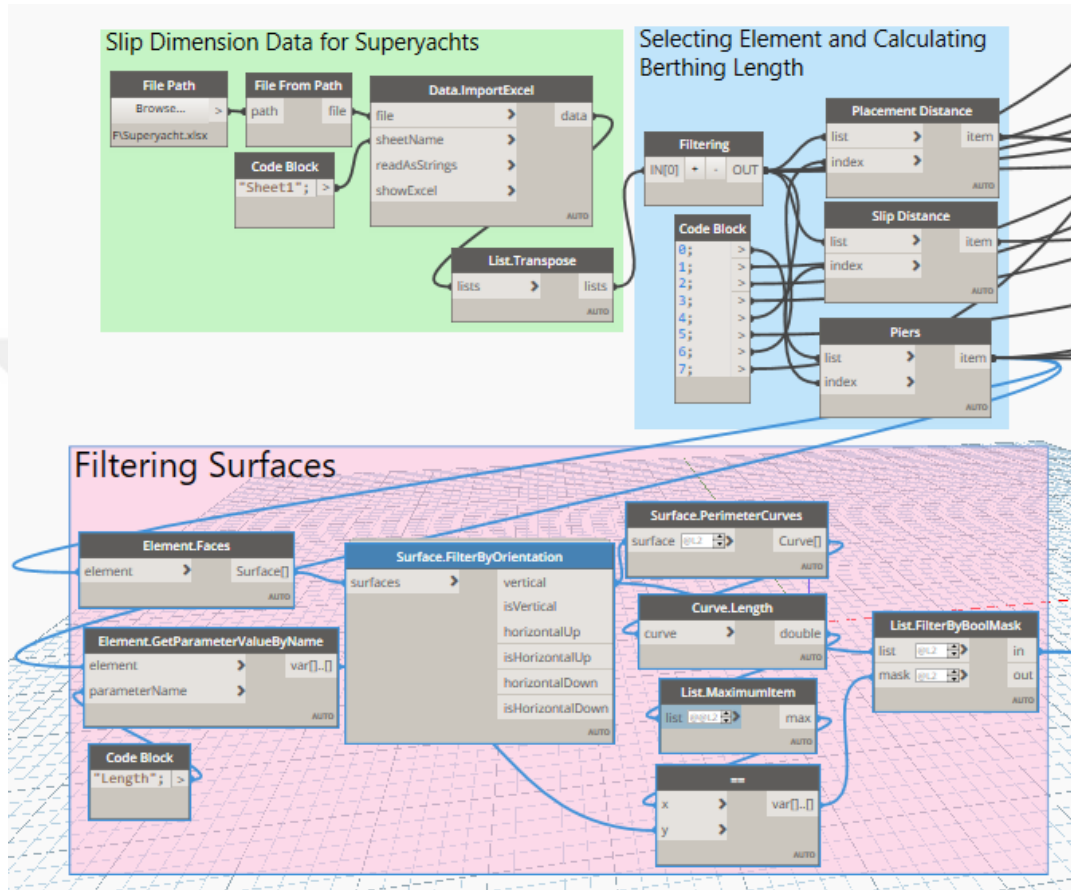
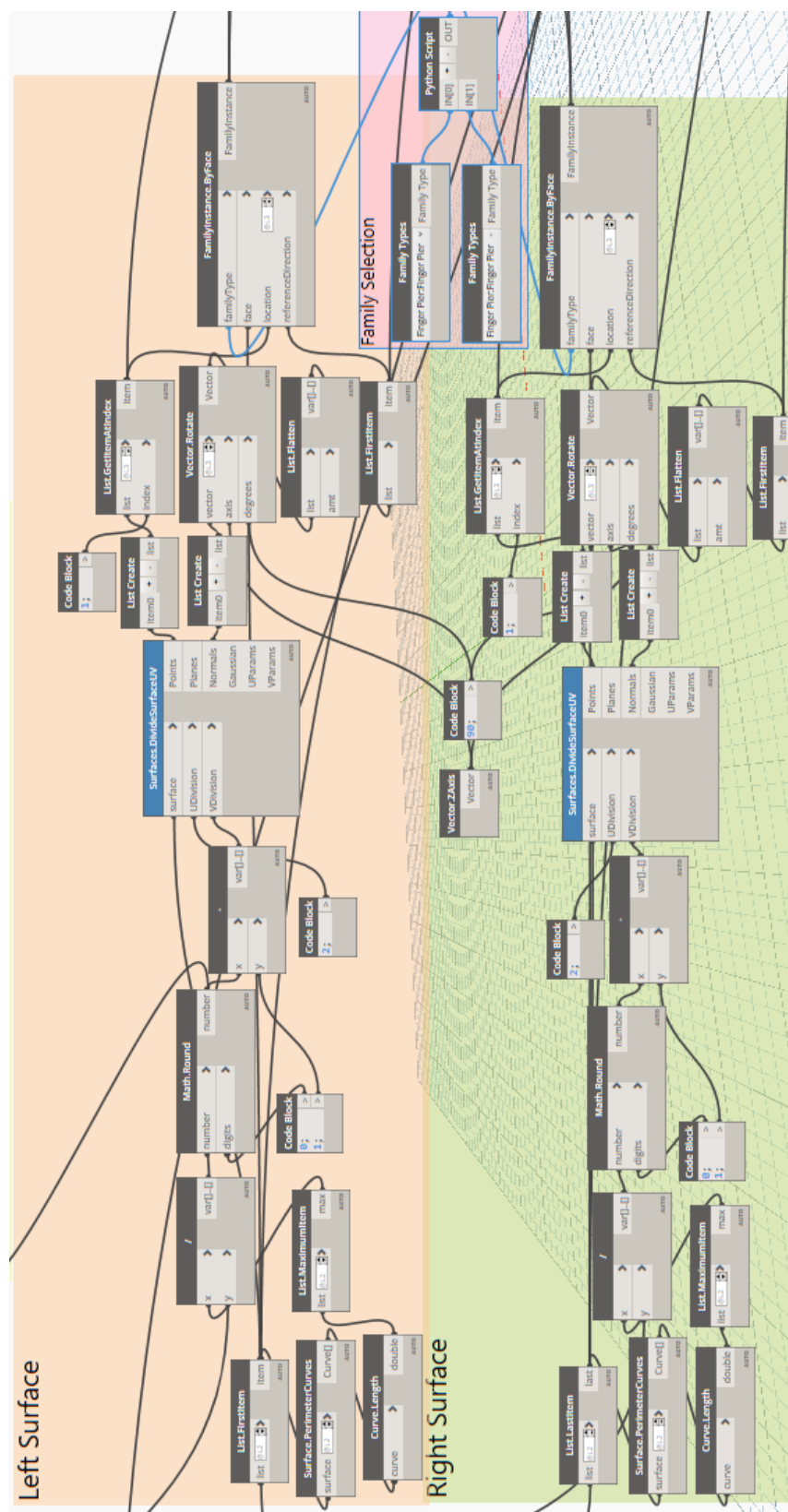
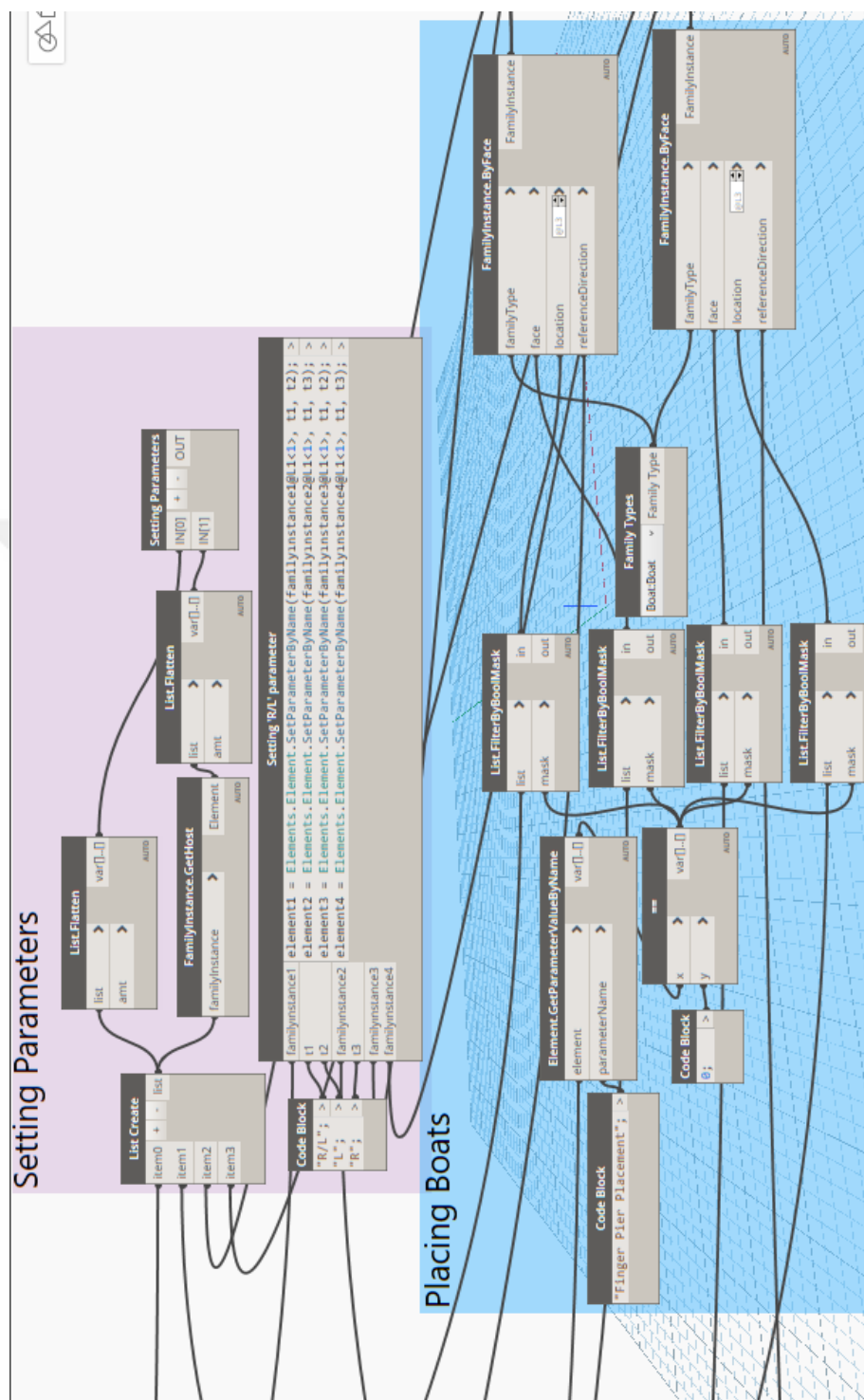


Figure G.17. Element Selection, Berthing Length Calculation and Obtainin Piers' Surfaces Steps of "Placing Families on Piers" Dynamo Script





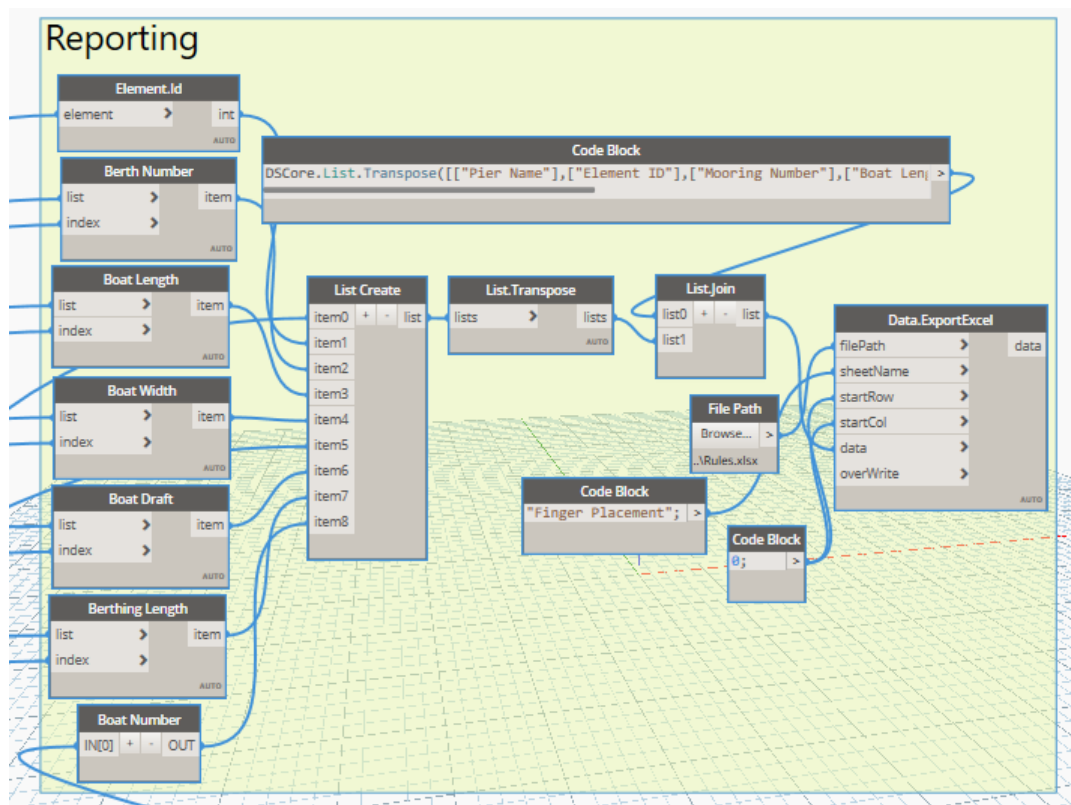


Figure G.20. Reporting Results in "Placing Families on Piers" Dynamo Script

```

1. import clr
2. clr.AddReference('ProtoGeometry')
3. clr.AddReference('RevitAPI')
4. clr.AddReference('RevitAPIUI')
5. import clr
6. # import RevitNodes
7. clr.AddReference('RevitNodes')
8. import Revit
9. clr.ImportExtensions(Revit.Elements)
10. # import Revit Services
11. clr.AddReference('RevitServices')
12. import RevitServices
13. from RevitServices.Persistence import DocumentManager
14. # import Revit API
15. clr.AddReference('RevitAPI')
16. import Autodesk
17. from Autodesk.Revit.DB import *
18. # import system.
19. import System
20. from System.Collections.Generic import *
21. from Autodesk.DesignScript.Geometry import *
22. import math
23.
24. doc = DocumentManager.Instance.CurrentDBDocument

```

```

25.
26. #Collecting all generic families in the model
27. family_collector = FilteredElementCollector(doc).OfCategory(BuiltIn
    Category.OST_GenericModel).WhereElementIsNotElementType().ToElement
    s()
28.
29. data = IN[0]          #dataset of preferred slip distances for su
    peryachts
30. Slip_distance = []    #List of calculated distances between finge
    r piers and anchor
31. Piers = []           #List of piers
32. Boat_length = []      #List of length of the longest boat berthn
    g to finger piers and anchors
33. Boat_width = []       #List of width of the longest boat berthing
    to finger piers and anchors
34. Boat_draft=[]        #List of draft of the longest boat berthing
    to finger piers and anchors
35. Berth_length=[]      #List of calculated berth length
36. Anchoring_number=[]  #List of mooring number between two finger
    piers
37. Placement_D = []     #List of width of fingers
38.
39. for family in family_collector:
40.     n = family.Name
41.     if 'Pier' == n:
42.         Piers.append(family)
43.
44. #Function to calculate required slip distance for yachts whose leng
    ths are equal to 20 m or smaller than 20 m
45. def slip(family):
46.     Finger= family.LookupParameter('Finger Pier Placement').AsValue
        String()
47.     if Finger == 'Yes':
48.         anchoring_parameter = family.LookupParameter('Berth Number'
        ).AsInteger()
49.         if anchoring_parameter == 1:
50.             boat_w = family.LookupParameter('Max Boat Width').AsDou
                ble()
51.             W = (boat_w/3.2808)+0.6          #Calculating distan
                ce between two finger piers
52.             return W
53.         if anchoring_parameter == 2:
54.             boat_w = family.LookupParameter('Max Boat Width').AsDou
                ble()
55.             W = 2*(boat_w/3.2808)+2
56.             return W
57.     if Finger == 'No':
58.         boat_w=family.LookupParameter('Max Boat Width').AsDouble()
59.         W = (boat_w/3.2808)+2          #Calculating distan
                ce between two anchors
60.         return W
61.
62. #Function to obtained the preferred slip distance for superyachts t
    hat are longer than 20 m
63. def SSslip(family,boat_l):
64.     Finger= family.LookupParameter('Finger Pier Placement').AsValue
        String()
65.     if Finger == 'Yes':

```

```

66.         anchoring_parameter = family.LookupParameter('Berth Number'
67.         ).AsInteger()
68.         row = data[0].index(boat_l)
69.         W = data [anchoring_parameter][row]
70.         return W
71.     if Finger == 'No':
72.         row = data[0].index(boat_l)
73.         W = data [1][row]
74.         return W
75. #Function to calculate required berth length for yachts
76. def blength(family):
77.     Finger= family.LookupParameter('Finger Pier Placement').AsValue
78.     String()
79.     if Finger == 'Yes':
80.         boat_l = family.LookupParameter('Max Boat Length').AsDouble
81.         ()
82.         B = (boat_l/3.2808)*1.2           #Calculating for fi
83.         nger piers
84.         return B
85.     if Finger == 'No':
86.         boat_l = family.LookupParameter('Max Boat Length').AsDouble
87.         ()
88.         B = (boat_l/3.2808)*2           #Calculating for an
89.         chors
90.         return B
91. #Listing required slip distances
92. for family in Piers:
93.     boat_l = round((family.LookupParameter('Max Boat Length').AsDou
94.     ble())/3.2808)
95.     if boat_l <= 20:
96.         w= round(slip(family),2)
97.         b=round(blength(family),2)
98.         Slip_distance.append(round(w,2))
99.         Berth_length.append(round(b,2))
100.         d = round(((family.LookupParameter('Max Boat Width').AsDoub
101.         le())/3.2808),2)
102.         Finger= family.LookupParameter('Finger Pier Placement').AsV
103.         alueString()
104.         if Finger == 'Yes':
105.             if b <= 12:
106.                 Placement_D.append(round((w+1),2))
107.             elif 12 < b <= 15:
108.                 Placement_D.append(round((w+1.4),2))
109.             elif 15 < b <=20:
110.                 Placement_D.append(round((w+2),2))
111.             elif 20 < b:
112.                 Placement_D.append(round((w+2.5),2))
113.             else: Placement_D.append(round((w+d),2))
114.         if boat_l > 20:
115.             w=round(SSlip(family,boat_l),2)
116.             b=round(blength(family),2)
117.             Slip_distance.append(round(w,2))
118.             Berth_length.append(round(b,2))
119.             d = round(((family.LookupParameter('Max Boat Width'
120.             ).AsDouble())/3.2808),2)
121.             Finger= family.LookupParameter('Finger Pier Placeme
122.             nt').AsValueString()

```

```

114.         if Finger == 'Yes':
115.             if b <= 12:
116.                 Placement_D.append(round((w+1),2))
117.             elif 12 < b <= 15:
118.                 Placement_D.append(round((w+1.4),2))
119.             elif 15 < b <=20:
120.                 Placement_D.append(round((w+2),2))
121.             elif 20 < b:
122.                 Placement_D.append(round((w+2.5),2))
123.             else: Placement_D.append(round((w+d),2))
124.
125.         #Collecting values of 'Boat Length', 'Boat Width' and 'Berth
        h Number' parameters
126.         for family in Piers:
127.             boat_l = family.LookupParameter('Max Boat Length').AsDo
        uble()
128.             boat_w = family.LookupParameter('Max Boat Width').AsDo
        ble()
129.             boat_d = family.LookupParameter('Max Boat Draft').AsDo
        uble()
130.             anchoring_parameter = family.LookupParameter('Berth Num
        ber').AsInteger()
131.             Boat_length.append(round((boat_l/3.2808),2)) #Listin
        g assigned boat lengths to piers
132.             Boat_width.append(round((boat_w/3.2808),2)) #Listin
        g assigned boat widths to piers
133.             Boat_draft.append(round((boat_d/3.2808),2))
134.             Anchoring_number.append(anchoring_parameter) #Li
        sting mooring numbers for piers
135.             OUT=Piers, Anchoring_number, Boat_length, Boat_width, Slip_
        distance, Berth_length, Placement_D, Boat_draft

```

Figure G.21. “Filtering” Python Code in “Placing Families on Piers” Dynamo Script

```

1. import clr
2. import sys
3. clr.AddReference('RevitAPI')
4. from Autodesk.Revit.DB import *
5. clr.AddReference('RevitAPIUI')
6. clr.AddReference('RevitServices')
7. import RevitServices
8. from RevitServices.Persistence import DocumentManager
9. from RevitServices.Transactions import TransactionManager
10. clr.AddReference('ProtoGeometry')
11. from Autodesk.DesignScript.Geometry import *
12. import math
13.
14. doc = DocumentManager.Instance.CurrentDBDocument
15. app = DocumentManager.Instance.CurrentUIApplication.Application
16. uiapp =DocumentManager.Instance.CurrentUIApplication
17. app = uiapp.Application
18.
19. hosted = UnwrapElement(IN[0]) #List of Finger Piers, Anchors and
        Boats as hosted elements
20. host = UnwrapElement(IN[1]) #List of piers as host elements
21. P_Name=[]
22. Boat_L=[]

```

```

23. Boat_W=[]
24. Boat_D=[]
25. Berth_length=[]
26. Berth=[]
27. name = []
28. L=[]
29.
30.
31.
32.
33. #Function to calculate required berth length for yachts
34. def blength(h):
35.     Finger= h.LookupParameter('Finger Pier Placement').AsString()
36.     if Finger == 'Yes':
37.         boat_l = h.LookupParameter('Max Boat Length').AsDouble()
38.         B = (boat_l/3.2808)*1.2 #Calculating for finger piers
39.         return B
40.     if Finger == 'No':
41.         boat_l = h.LookupParameter('Max Boat Length').AsDouble()
42.         B = (boat_l/3.2808)*2 #Calculating for anchor piers
43.         return B
44.
45.
46. for h in host:
47.     P_Name.append(h.LookupParameter('Pier Name').AsString())
48.     Boat_W.append(round((h.LookupParameter('Max Boat Width').AsDouble()/3.2808),2))
49.     Boat_D.append(round((h.LookupParameter('Max Boat Draft').AsDouble()/3.2808),2))
50.     Boat_L.append(round((h.LookupParameter('Max Boat Length').AsDouble()/3.2808),2))
51.     Berth.append(h.LookupParameter('Berth Number').AsInteger())
52.     Berth_length.append(round((blength(h)),2))
53.
54. #Setting 'Host Parameter', 'Max Boat Length', 'Max Boat Width', 'Max Boat Draft' to Finger Piers, Anchors, Boats
55. for h, p, l, w, d in zip(hosted, P_Name, Boat_L, Boat_W, Boat_D):
56.     host_pier=h.LookupParameter('Host Pier')
57.     boat_l=h.LookupParameter('Max Boat Length')
58.     boat_w=h.LookupParameter('Max Boat Width')
59.     boat_d=h.LookupParameter('Max Boat Draft')
60.     TransactionManager.Instance.EnsureInTransaction(doc)
61.     host_pier.Set(p)
62.     boat_l.Set(round((l*3.2808),2))
63.     boat_w.Set(round((w*3.2808),2))
64.     boat_d.Set(round((d*3.2808),2))
65.     TransactionManager.Instance.TransactionTaskDone()
66.
67. #Setting 'Berthing Length' to Finger Piers, Anchors
68. for h, b in zip(hosted, Berth_length):
69.     n=h.Name
70.     if n!='Boat':
71.         TransactionManager.Instance.EnsureInTransaction(doc)
72.         finger_l=h.LookupParameter('Berthing Length')
73.         finger_l.Set(round((b*3.2808),2))
74.         TransactionManager.Instance.TransactionTaskDone()

```

```

75.
76. #Setting 'Berth Number' to Finger Piers
77. for h, n in zip(hosted, Berth):
78.     m = h.Name
79.     if m == 'Finger Pier':
80.         TransactionManager.Instance.EnsureInTransaction(doc)
81.         berth_n=h.LookupParameter('Berth Number')
82.         berth_n.Set(n)
83.         TransactionManager.Instance.TransactionTaskDone()
84.
85. for h, b in zip(hosted, Berth_length):
86.     n=h.Name
87.     if n == 'Finger Pier':
88.         finger_l=h.LookupParameter('Berthing Length').AsDouble()
89.         l = finger_l/3.2808
90.         L.append(l)
91.         if l <= 12:
92.             TransactionManager.Instance.EnsureInTransaction(doc)
93.             finger_w = h.LookupParameter('Finger Width')
94.             finger_w.Set(1*3.2808)
95.             TransactionManager.Instance.TransactionTaskDone()
96.         elif 12 < l <= 15:
97.             TransactionManager.Instance.EnsureInTransaction(doc)
98.             finger_w = h.LookupParameter('Finger Width')
99.             finger_w.Set(1.4*3.2808)
100.            TransactionManager.Instance.TransactionTaskDone
101.            ()
102.            elif 15< l <=20:
103.                TransactionManager.Instance.EnsureInTransaction
104.                (doc)
105.                finger_w = h.LookupParameter('Finger Width')
106.                finger_w.Set(2*3.2808)
107.                TransactionManager.Instance.TransactionTaskDone
108.                ()
109.                elif 20< l:
110.                    TransactionManager.Instance.EnsureInTransaction
111.                    (doc)
112.                    finger_w = h.LookupParameter('Finger Width')
113.                    finger_w.Set(2.5*3.2808)
114.                    TransactionManager.Instance.TransactionTaskDone
115.                    ()
116.
117. OUT = 0

```

Figure G.22. “Setting Parameters” Python Code in “Placing Families on Piers”
Dynamo Script