



**SABİT BAĞLAMA TASARIMI İÇİN BİLGİSAYAR
PROGRAMI GELİŞTİRİLMESİ**

Betül BAYRAKTAR YÜKSEL

Yüksek Lisans Tezi

Anabilim Dalı: İnşaat Mühendisliği

Danışman: Prof. Dr. Muhammet Emin EMİROĞLU

TEMMUZ 2019

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SABİT BAĞLAMA TASARIMI İÇİN BİLGİSAYAR PROGRAMI GELİŞTİRİLMESİ

Yüksek Lisans Tezi

BETÜL BAYRAKTAR YÜKSEL

161115104

Tezin Enstitüye Verildiği Tarih: 01.07.2019

Tezin Savunulduğu Tarih: 19.07.2019

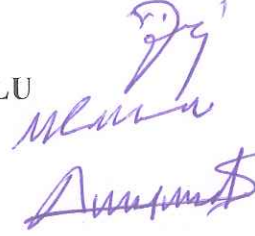
Anabilim Dalı: İnşaat Mühendisliği

Programı: Hidrolik

Danışman: Prof. Dr. M. Emin EMİROĞLU

Dr.Öğr.Üyesi Mustafa ULAŞ

Dr.Öğr.Üyesi Yusuf DOĞAN



TEMMUZ-2019

ÖNSÖZ

Lisans ve lisansüstü eğitimim boyunca gerek ders dönemimde gerekse tez dönemimde beni bilgisiyle, gayretiyle hiç yalnız bırakmayan çalışmamın her aşamasında akademik tecrübelerini benimle paylaşan araştırmamın yürütülmesi ve yazıya aktarılması sırasında çalışmama yol gösteren ve katkılarda bulunan değerli danışmanım Prof. Dr. Muhammet Emin EMİROĞLU'na saygı ve teşekkürlerimi sunarım.

Tez çalışmamın büyük bir bölümünü kapsayan web tabanlı program oluşturukod yazma kısmında bana yardımcı olan yazılım mühendisliği öğrencisi Samet Eray ERDEM'e teşekkür ederim.

Hayatımın her döneminde olduğu gibi bu dönemde de tüm sıkıntıları ve mutlulukları benimle paylaşan beni bu zamana getiren dedelerim Dursun BAYRAKTAR, Kemal AYAN ve annemle babama teşekkür ederim.

Son olarak tez çalışmam boyunca maddi ve manevi desteklerini bir an olsun esirgemeyen anlayış ve desteği ile her an yanımda olan değerli eşim Gencay YÜKSEL'e teşekkürü borç bilirim.

Betül BAYRAKTAR YÜKSEL

Elazığ-2019

İÇİNDEKİLER

	Sayfa No:
ÖNSÖZ	I
İÇİNDEKİLER	II
ÖZET	IV
SUMMARY	V
ŞEKİLLER LİSTESİ	VI
TABLolar LİSTESİ	VIII
1. GİRİŞ.....	1
1.1. Literatür Özeti	2
2. BAĞLAMALARIN TASARIMI.....	4
2.1 Bağlamaların İnşa Amaçları.....	4
2.2 Bağlamaların Sınıflandırılması	5
2.2.1 Bağlamaların Taşkın Debisine Göre Sınıflandırılması.....	5
2.2.2 Yapısal Tasarıma Göre Sınıflandırılması	5
2.3. Sabit (Dolu Gövdeli) Bağlamalar.....	6
2.4 Hareketli (Kapaklı) Bağlamalar	7
2.5 Karma Bağlamalar	10
2.6 Şişirilebilir Lastik Bağlamalar (Rubber Dam)	10
2.7 Sifon Bağlamalar.....	12
2.8. Mansap Su Seviyesine Göre Sınıflandırılması.....	12
2.9. Bağlamalarda Yapı Elemanları	14
2.10. Su Alma Yapısı Elemanları.....	15
2.11. Dolu Gövdeli Bağlamaların Hidrolik Hesapları	17
2.12. Bağlama Genişliği (Kret Uzunluğunun) Bulunması	18
2.13 Savak Kapasitesi	19
2.14. Çakıl Geçidi	20
2.15. Hidrolik Profilin Tayini	22
2.16. Düşü Havuzlu Hesapları	24
2.16. Düşü Havuzu Boyunun Bulunması (Hidrolik Sıçrama Uzunluğu).....	26
3. SABİT BAĞLAMA TASARIMI İÇİN PROGRAM GELİŞTİRİLİRKEN	
KULLANILAN TEKNOLOJİLER	31

4. SABİT BAĞLAMA TASARIMI İÇİN YAZILIM GELİŞTİRİLMESİ	34
4.1. Yazılımın Yapım Aşamaları.....	34
4.2. Kaplamasız Kanal	39
4.3. Kaplamalı Ya Da Kaplamasız Olma Durumu.....	41
4.4. Çökeltim Havuzunun Boyutlandırılması.....	42
4.5. Çökeltim Havuzunun Sonundaki Eşiğin Boyutlandırılması	44
4.6. Bağlama Genişliği ve Yüksekliğinin Tayini	44
4.7. Su Alma Ağzının Boyutlandırılması.....	46
4.8. Yıkama Kanalıının Byutlandırılması	46
4.9. Yıkama Kanalıının Eğimi.....	47
4.10. Çakıl Geçidi Debisinin Tayini	47
4.11. Hidrolik Profilin Tayini	49
4.12. Bağlamanın Statik Profilinin Tayini	49
4.13. Devrilme Tahkiki	50
4.14. Kayma Tahkiki.....	51
4.15. Taban Gerilmelerinin Tayini.....	52
4.16. Bağlama Düşüm Yatağının Tayini.....	53
4.17. Düz Düşüm Yataklarının Kalınlık Tayini	55
4.18. Palplanş Perde Çakılması.....	56
SONUÇLAR VE ÖNERİLER.....	60
KAYNAKLAR	61
EKLER	64
ÖZGEÇMİŞ	130

ÖZET

Bağlamalar su mühendisliğinde yaygın olarak kullanılan önemli hidrolik yapılardandır. Bağlamalar özellikle akarsu santrali tesislerinde ve sulama tesislerinde sık kullanılmaktadır.

Bu çalışmada sabit bağlamaların tasarımı için bir bilgisayar programı geliştirilmiştir. Geliştirilen program web tabanlı bir uygulama olup herhangi bir browser (chrome, safari, firefox) üzerinden erişim sağlanarak kullanılabilme kolaylığına sahiptir.

Program geliştirilirken Visual Studio Code metin editörü kullanılmıştır. Diğer metin editörleri arasından bu kodun tercih edilmesinin sebebi akıllı kod tamamlama, kodu yeniden yapılandırma ve çok fazla eklenti desteği gibi birçok özelliğinin bulunmasıdır. Visual Studio Code kullanılarak oluşturulan sabit bağlama tasarımında uygulamayı geliştirmek için Javascript, Html, css, bootstrap, chart js, python, numpy, matplotlib, flank, canvas, npm, node.js, axios ve pycharm gibi birçok teknoloji de kullanılmıştır.

Hazırlanan yazılım eğitim amaçlı olarak planlanmıştır. Farklı seçenekler için sabit bağlama tasarımı, geliştirilen program sayesinde yapılabilmektedir. Programda bütün aşamalar görülebilmekte ve tasarım aşamaları açıklamalardan takip edebilmektedir. Bu çalışmada, programda bulunan butonlar ve bunların kullanım amaçları, uygulamanın nasıl çalıştırılacağı sunulmaya çalışılmıştır. Bu tasarım için grafikler kod yazılarak oluşturulmuştur. Bu kodların grafiğe dökülmesinde chart js, python, matplotlib teknolojilerinden faydalanılmıştır. Gerekğinde kullanıcıya grafikler hakkında açıklamalar da verilmiştir.

Program iki farklı katmandan oluşmaktadır. Birinci katman iş tabakası (*business layer*) olup, bütün mantıksal işlemler ve matematiksel döngüler burada gerçekleştirilmiştir. Uygulamanın bu katmanındaki kodlarda basit matematiksel işlemler (dört işlem) programlama dilinin kendi işlevi içerisinde yapılırken kök ve bir sayının üssünü alma gibi daha ileri matematiksel işlemler için Javascript' in math. kütüphanesinden yararlanılmıştır. Üst düzey matematiksel işlemler (2° veya 3°den denklem çözme, denklemin grafiğini çizdirme) python, numpy ve matplotlib teknolojileri ile gerçekleştirilmiştir. İkinci katman ise sunum tabakasıdır (*presentation layer*). Uygulamanın kullanıcı ile etkileşimi olan arayüzün yapılandırıldığı sunum katmanıdır. Sunum katmanı ile iş katmanında hazır hale getirilen verilerin kullanıcıya gidecek olan görünümü belirlenerek uygulama tamamlanmış olmaktadır.

Anahtar Kelimeler: Bağlama, Bağlama Tasarımı, Visual Studio Code, Hidrolik Yapı

SUMMARY

Computer Program Development for the Design of Diversion Weir

Weirs are important hydraulic structures commonly used in the water engineering. Weirs are often used especially in river power plants and irrigation facilities.

In this study, a computer program has been developed for the design of diversion weir. Developed program which is a web-based application has the ease of use by enabling the access via any browser (chrome, safari, firefox).

Visual Studio Code text editor was used in the development of the program. Among other text editors, this code has been preferred because it has many features, such as intelligent code completion, code reconstruction, and lots of plug-in support. In order to develop the application in diversion weir design using Visual Studio Code, many technologies such as Javascript, HTML, css, bootstrap, chart js, python, numpy, matplotlib, flank, canvas, npm, node.js, axios and pycharm have been used.

The programme was especially for educational purposes. Design of diversion weir for different options has been made through the developed program. All stages can be presented and stages of design has been followed from the given explanations. In this study, the buttons in the program and their intended use, how to run the program are given. For this design, graphics have been created by writing code. Chart js, python, matplotlib technologies are used in the graphing of these codes. The user is given explanations about the graphics, when it is necessary.

The program consists of two different layers. The first layer is the business layer and all logical operations and mathematical cycles are performed here. In the code in this layer of the application, simple mathematical operations (four operations) are performed within the function of the programming language itself, while Javascript. Library is used for further mathematical operations such as taking the root and exponent of a number. High-level mathematical operations (solving equations from 2nd or 3rd degrees, plotting the equation) have been carried out using python, numpy and matplotlib technologies. The second layer is the presentation layer. Presentation layer is the layer which is the user- application interface where user interacts with application. With the presentation layer, the view of the data made ready in the business layer to the user is determined and the application is completed

Keywords: Diversion weir, Weir design, Visual Studio Code, Hydraulic Structure

ŞEKİLLER LİSTESİ

Sayfa No:

Şekil 2.1.	Akarsu santralleri için bağlama inşa ederek düşü elde edilmesi	4
Şekil 2.2.	Bağlama çeşitleri	6
Şekil 2.3 (a).	Sabit bağlama tipik en kesidi	6
Şekil 2.3 (b).	Sabit bağlama tipik yerleşim planı	7
Şekil 2.3 (c).	Baklan Ovası Sulamasına ait Erenler Regülatörü (dolu gövdeli, mansaptan ve membadan görünüş)	7
Şekil 2.4.	Mustafa Kemal Paşa Regülatörü (kapaklı regülatör-mansaptan görünüş)	8
Şekil 2.5.	Konya - Çumra Kısıkyayla Regülatörü - 1913 (kapaklı regülatör) (mansaptan ve membadan görünüş)	8
Şekil 2.6.	Hareketli bağlama plan ve kesitleri	9
Şekil 2.7 (a).	Lastik bağlama	10
Şekil 2.7 (b).	Lastik bağlama (baraj) enkesidi	11
Şekil 2.8.	Lastik savaklı regülatör (mansaptan görünüş)	11
Şekil 2.9.	Aşağı Büyük Menderes Nehri üzerindeki lastik savaklı regülatör (işletme halinde)	12
Şekil 2.10.	Gönen Regülatörü (mansaptan ve membadan görünüş)	12
Şekil 2.11.	Batmış akımlı bağlama	13
Şekil 2.12.	Serbest akımlı bağlama	13
Şekil 2.13.	Sağ ve sol sahilde sahilde su alma ağızı bulunan bağlamanın plan görünümü	16
Şekil 2.14.	Kabartma kotu	17
Şekil 2.15.	Keskin kenarlı savak üzerindeki akım (Erkek ve Ağırlioğlu, 2013)	22
Şekil 2.16.	Ogee hidrolik profili (Erkek ve Ağırlioğlu, 2013)	24
Şekil 4.1.	Metin (text) editörü <i>visual studio code</i>	34
Şekil 4.2.	Javascript dosyası oluşturma	35
Şekil 4.3.	Bağlama enkesiti	36
Şekil 4.4.	Anahtar eğrisi	37
Şekil 4.5.	Islak alan ve akım derinliği arasındaki değişim	38
Şekil 4.6.	Islak çevre ve yükseklik arasındaki değişim	38

Şekil 4.7.	Hidrolik yarıçap ve akım derinliği arasındaki değişim	39
Şekil 4.8.	Akarsu enkesi üzerine bağlama ve çakıl geçidinin yerleşimi	45
Şekil 4.9.	Anahtar eğrisi	48
Şekil 4.10.	Hidrolik profil alt ve üst nap	50
Şekil 4.11.	Tatbiki profilin belirlenmesi	53



TABLÖLER LİSTESİ

Sayfa No:

Tablo 2.1.	μ_1 ve μ_2 savak katsayıları	20
Tablo 2.2.	Orifis katsayısı değerleri (Tuna, 1985).....	21
Tablo 2.3.	Creager profiline göre alt ve üst nap değerleri	23
Tablo 4.1.	Akarsu enkesitinin ıslak alan, ıslak çevre ve hidrolik yarıçap hesapları..	36
Tablo 4.2.	Akarsuyun debi hesapları ve akım derinlikleri.....	37
Tablo 4.3.	İsale kanalı optimizasyonu (Kaplamasız kanal)	40
Tablo 4.4.	İsale kanalı optimizasyonu (Kaplmalı kanal)	41
Tablo 4.5.	Kanalın kaplamalı ya da kaplamasız seçilmesi	42
Tablo 4.6.	Çökeltim havuzu boyutları sonuçları.....	43
Tablo 4.7.	Çökeltim havuzu göz açıklığı	43
Tablo 4.8.	P_e ve h_e	44
Tablo 4.9.	Sonuçların ekrana yazdırılması	45
Tablo 4.10.	Su alma ağzının boyutlandırılması	46
Tablo 4.11.	Yıkama kanalının boyutlandırılması	47
Tablo 4.12.	Yıkama kanalı eğimi.....	47
Tablo 4.13.	$bç$, $Qç$ ve μ değerleri.....	48
Tablo 4.14.	Hidrolik profil hesabı	49
Tablo 4.15.	h' , γ' , $c2$, $c1$, $\tan\alpha$ ve α değerleri	50
Tablo 4.16.	Devrilme tahkiki	51
Tablo 4.17.	$Mö$, Md ve nd değerleri	51
Tablo 4.18.	Kayma tahkiki	52
Tablo 4.19.	Taban gerilmeleri tayini.....	52
Tablo 4.20.	Bağlama düşüm yatağı tayini	55
Tablo 4.21.	Düz düşüm yataklarının kalınlık tayini	56
Tablo 4.22.	Palplanş Perde Hesabı	57

1. GİRİŞ

Bağlama bir akarsudan depolama yapılmadan suyun derlenmesi için gerekli kabarmayı sağlayarak istenilen seviyede ve arzu edilen miktarda su almayı sağlayan yapılardır. Bağlamanın en temel amacı su seviyesini yükseltmek olup inşa edilen tesisin maksat ve ihtiyacına göre yapılış amaçları değişkenlik gösterebilir. Bağlamalar sulama tesisi, akarsu santralleri, akarsu ulaşımı, meskûn bölge içme ve kullanma suyu temini tesislerinde yaygın olarak kullanılmaktadır.

Çeşitli bağlama tipleri mevcuttur. Bunları sabit bağlama, hareketli bağlama, karma bağlama, sifon bağlama, lastik (kauçuk) bağlama şeklinde sıralamak mümkündür. Bunlar içerisinde en çok tercih edilen tür, sabit bağlamadır. Hidrolik yapı tasarımı uzmanlık gerektiren bir konudur. Bu çalışmada sabit bağlamaların eğitim amaçlı olarak tasarımı için bir program geliştirilmiştir.

Bu çalışmada program geliştirmek için Visual Studio Code kullanılmıştır. Bu tez kapsamında; uygulamanın nasıl çalıştırılacağı anlatılarak Visual Studio Code metin editörü hazır duruma geldikten sonra new file butonuna tıklanıp Javascript uzantılı dosya oluşturulup kod yazma işlemine başlanır. Bu kodlar herbir kullanıcının projesine uygun olarak yazılmış olup her proje için farklı ve doğru sonuçlar elde edecek şekilde düzenlenmiştir. Kodlar yazılırken kullanılan veriler kullanıcının kendisine ait bağlama projesi föyünden alınmış olup hazırlanan projeye göre bazı veriler sabit tutulup bazı verilerin ise kullanıcı tarafından dışarıdan girilmesine müsaade edilmiştir. Sabit bağlama tasarımındaki grafikler kod yazılarak oluşturulmuştur. Bu kodların grafiğe dökülmesinde chart.js, python, matplotlib teknolojilerinden faydalanılmıştır.

Eğitim amaçlı hazırlanan arayüzler lisans ve lisansüstü eğitiminde yaygın bir şekilde kullanılmaktadır. Eğitim amaçlı hazırlanan mühendislik çalışmalarının yayımlandığı dergiler de mevcuttur. Örnek olarak; “Journal of Engineering Education”, “European Journal of Engineering Education”, “Computer Applications in Engineering Education” ve “Engineering Education” gibi saygın dergileri saymak mümkündür.

Bağlamaların türlerini ve her bir türün hidrolik karakteristiklerini görsellerle kullanıcıya sunmak ve bu hidrolik yapıların bütün tasarım kriterlerini belirlemek amaçlanmaktadır. Bu çalışma kapsamında bir sabit bağlamanın tasarlanması Visual Studio Code kullanılarak yapılmıştır. Kullanıcı gerekli bilgileri girdikten sonra sabit

bağlamanın tasarımını yapabilmektedir. Hazırlanan programın eğitim amaçlı olarak geliştirilmiştir. Geliştirilen programın; su mühendisliği dalındaki lisans, yüksek lisans, doktora öğrencilerin eğitiminde ve uygulamadaki genç mühendislerle yarar sağlaması amaçlanmıştır.

1.1. Literatür Özeti

Hidrolik yapıların tasarımında bilgisayar uygulamaları yaygın bir şekilde kullanılmaktadır. Tasarım mühendisleri için yazılımlar veya programlar kullanmak zaman kazanımı açısından çok önemlidir.

Literatürde hidrolik yapıların tasarımı için birçok paket yazılım ve programlar geliştirilmiştir. Aşağıda su kaynakları problemlerinde bilgisayar uygulamalarına örnekler verilmiştir.

Cihangir (1993); Yandan alıslı bir sulama yapısına sahip bağlamanın ön tasarımı için DWEIR isimli Q basic dilinde yazılmış bir program geliştirmiştir. Bu çalışma bir yüksek lisans tezidir. Yaptığı programda söz kaonusu yapını maliyetine de karar verilmektedir. Yazar hazırladığı programın kolay ve esnek olduğunu ifade etmiştir. Chau ve Zhang (1995) içme suyu şebekelerinin hidrolik tasarımını MS Excel kullanarak yapmışlardır. Bu arayüzün hem hidrolik tasarımcılar ve hem de eğitim amacıyla kullanılabileceğini belirtmişlerdir. Yanmaz ve Ozaydın (2000) bağlamaların optimum boyutlandırılması için bir yaklaşım önermişlerdir. Geliştirdikleri programın kullanıcı için kolay anlaşılır ve kullanılabilir olduğunu vurgulamışlardır. Geliştirdikleri program ile ilgili bir uygulamayı hazırladıkları makalede vermişlerdir. Turan (2004); yandan alıslı bir bağlamanın optimum tasarımı için Visual Basic.Net programlama dilinde yazılmış WINDWEIR isimli, Windows işletim sistemi altında çalışan bir program geliştirmiştir. Yazar hazırladığı programın görsel içerikler ihtiva ettiğini ve kullanımının kolay olduğunu ifade etmiştir. Hazırlanan programda yapının maliyeti de hesaplanabilmektedir. Bu çalışma bir yüksek lisan tezidir. Yanmaz ve Turan (2005) yandan alıslı bir sabit bağlamanın hidrolik tasarımını bilgisayar destekli olarak yapmışlardır. Wu (2007) su şebekelerindeki kaçakların tespiti için kural tabanlı bir uzman sistem geliştirmişlerdir. Alimoğlu vd. (2014) nehir tipi hidroelektrik tesislerin bilgisayar destekli tasarımını yapmışlardır. Bu tasarımda ön planlama için sabit

bağlamanın boyutlandırması da verilmiştir. Brkić (2017) sürtünme katsayısını tespit etmek için Colebrook Denklemine çözümünü Excel’de yapmıştır.

Yukarıdaki çalışmalar bakıldığında, hazırlanan programlarda amaç eğitimde kullanılması olmadığı görülmektedir. Bu çalışma sabit bağlamanın bütün elemanları tasarım kriterleri dikkate alınarak, bütün aşamalar kullanıcıya öğretilerek yapılmaya çalışılmıştır. Kullanıcıya tasarım ve hesaplamalar hakkında açıklamalar, görseller verilmektedir. Farklı seçenekler için ayrı ayrı yapılabilen tasarım eğitim amaçlı olarak kullanıcıya sunulmaktadır. Bu açılarından geliştirilen program literatüre ve eğitime katkı koymak amacıyla geliştirilmiştir.



2. BAĞLAMALARIN TASARIMI

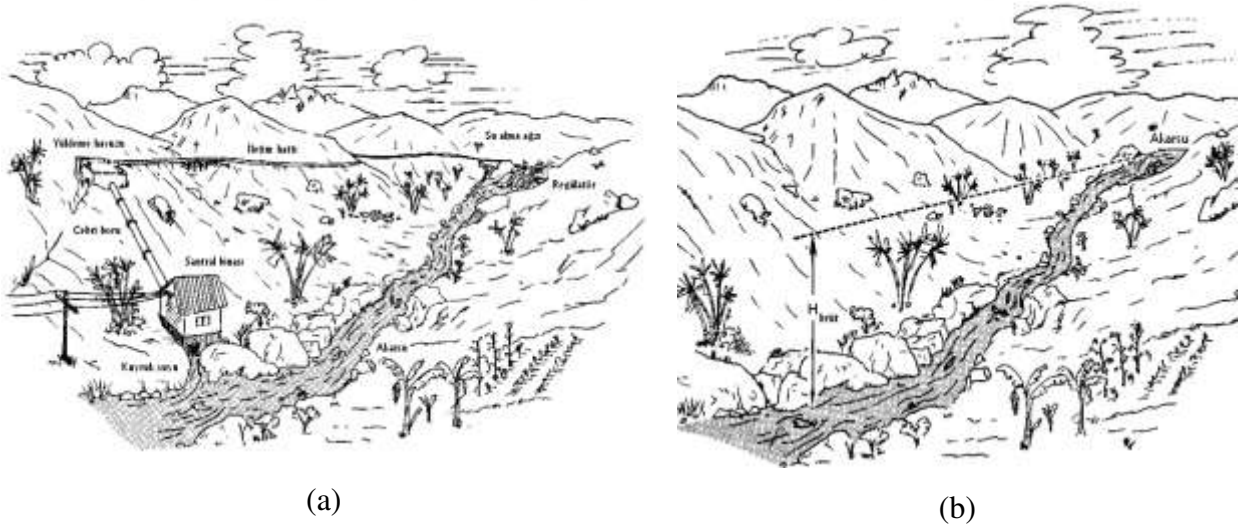
Bir akarsudan, depolama yapılmadan suyun derlemesi için gerekli kabarmayı sağlayarak istenilen seviyede ve arzu edilen miktarda su almayı temin eden yapılara bağlama denir. Daha geniş anlamda yardımcı yapıları ile beraber çevirme yapıları olarak da adlandırılmaktadır.

Bağlamaların esas amacı su seviyesi yükselterek suyun çevrilmesini sağlamaktır. Ancak amaçları tesisin maksat ve ihtiyacına göre değişmektedir. (Baban 1995; Erdem 2006)

2.1 Bağlamaların İnşa Amaçları

Bağlamaların yapım amaçları aşağıda özetlenmiştir.

1) Hidroelektrik Tesisleri için düşü elde etmek amacıyla bağlamalar inşa edilmektedir (Özbek, 2009). Şekil 2.1’de şematik olarak akarsu santrallerinin elemanları gösterilmiştir.



Şekil 2.1. Akarsu santralleri için bağlama inşa ederek düşü elde edilmesi
(<http://www.kimyamuhendisi.com>)

2) Akarsu ulaşımı için gerekli derinliği sağlamak (küçük gemilerin işletilmesi) amacıyla bağlamalar inşa edilmektedir.

3) Yeraltı sularını beslemek ve yükseltmek amacıyla bağlamalar inşa edilmektedir.

4) Akarsuyun kendi kendini temizlemesine yardım etmek amacıyla bağlamalar inşa edilmektedir.

5) Akarsu seviyesini yükselterek arazinin sulanmasını sağlamak ve akarsuyun eğimini toplamak suretiyle su kuvvetleri, sulama ve su getirme tesisleri için isale kanalı boyunu kısaltmak amacıyla bağlamalar inşa edilmektedir.

6) Çok nadir de olsa akışları düzenlemek içinde kullanılmaktadırlar.

2.2 Bağlamaların Sınıflandırılması

Bağlamalar çeşitli kriterlere göre sınıflandırılabilir, bu sınıflandırmalar alt bölümlerdeki gibidir (Erkek ve Ağı).

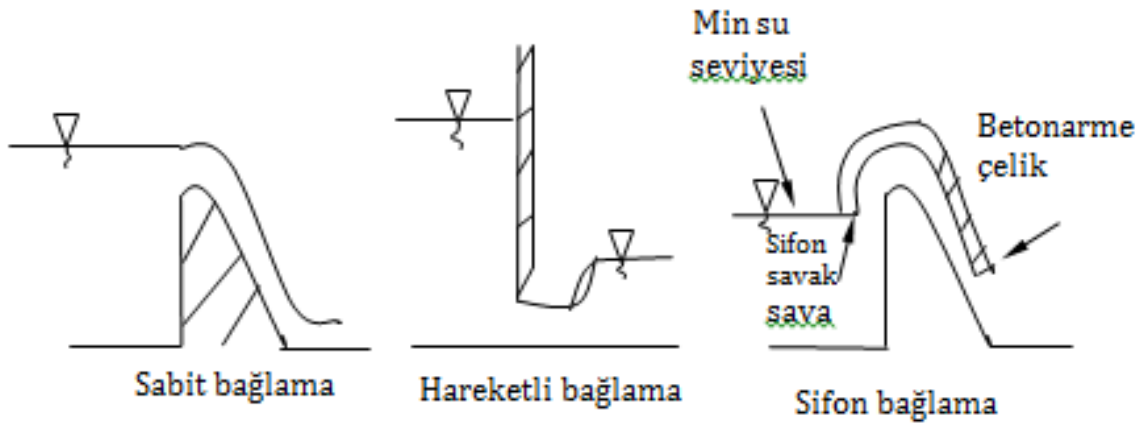
2.2.1 Bağlamaların Taşkın Debisine Göre Sınıflandırılması

- a) Küçük Bağlamalar:** 100 yıllık frekanslı debisi $0-100 \text{ m}^3/\text{s}$ arasında olan bağlamalardır. Genellikle kapaklıdır ve yanlarında dolu savakları vardır. ($Q_{100} < 100 \text{ m}^3/\text{s}$)
- b) Orta Bağlamalar:** 100 yıllık frekanslı debisi $100 - 500 \text{ m}^3/\text{s}$ arasında olan bağlamalardır. Hareketli ve dolu gövdelidirler. ($100 < Q_{100} < 500 \text{ m}^3/\text{s}$)
- c) Büyük Bağlamalar:** 100 yıllık frekanslı debisi $500 \text{ m}^3/\text{s}$ 'den fazla olan bağlamalardır. ($Q_{100} > 500 \text{ m}^3/\text{s}$)

2.2.2 Yapısal Tasarıma Göre Sınıflandırılması

Bağlamaların tipi belirlenirken dere yatağının durumu ve bağlama ile su alınacak tesisin maksat ve gereksinimi dikkate alınır. Genellikle bağlamalar üç tiptedir. Şekil 2.2'de bağlama çeşitlerine ait kesitler verilmiştir.

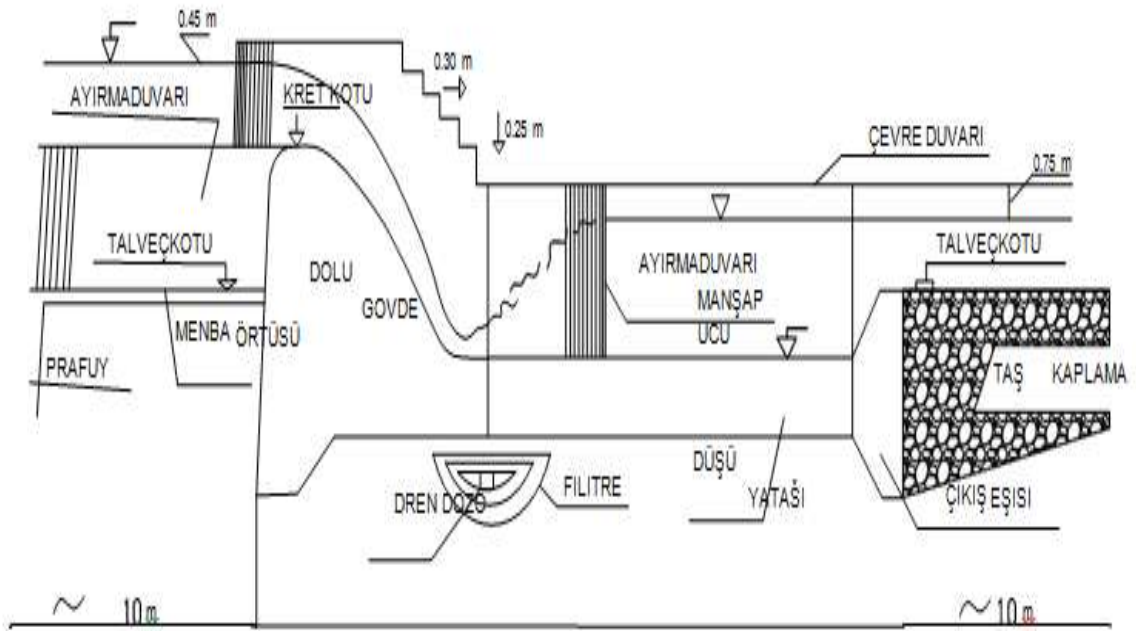
- a) Sabit (Dolu Gövdeli) Bağlamalar,
- b) Hareketli Bağlamalar,
- c) Karma (Kombine) Bağlamalar,
- d) Şişirilebilir lastik bağlama (Rubber Dam),
- e) Sifon Bağlamalar.



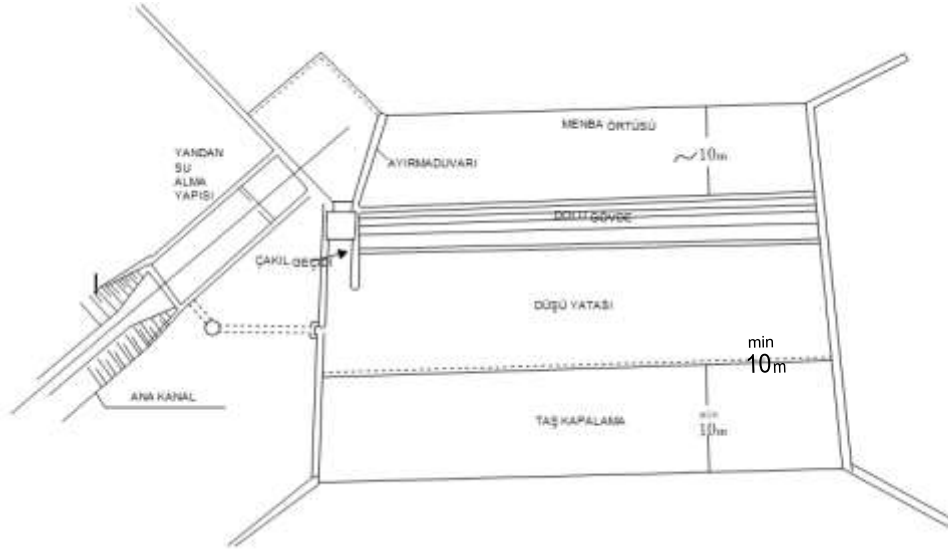
Şekil 2.2 Bağlama çeşitleri (Tuna, 1985)

2.3. Sabit (Dolu Gövdeli) Bağlamalar

Sabit bağlamalar, genellikle suyun bağlama gövdesi üzerinden serbestçe savaklanması esasına dayanan yapılardır (Şekil 2.3 (a)). Sabit bağlama değişmeyen kretli sabit bir kabartma gövdesinden ibarettir (Şekil 2.3 (b)). Sabit bağlama, su seviyesi değişiminin fyezan geçişinin ve birikintilerin herhangi bir zarara sebep olmadığı yerlerde kullanılır (Öziş, 1983).



Şekil 2.3 (a). Sabit bağlama tipik en kesidi (Öziş, 1983)



Şekil 2.3 (b). Sabit bağlama tipik yerleşim planı (Öziş,1983)

Su seviyesini yükseltmek için Şekil 2.3 (c).’te görüldüğü gibi nehir boyunca dolusavak inşa edilir (Tuna, 1985).



Şekil 2.3 (c). Denizli-Baklan Ovası Sulamasına ait Erenler Regülatörü (dolu gövdeli, mansaptan ve membadan görünüş) (Gürsoy, 2010)

2.4 Hareketli (Kapaklı) Bağlamalar

Memba su seviyesinin sabit tutulması ve düzenlenmesi veyahut akışın düzenlenmesi için kullanılır.

Şekil 2.4 te görüldüğü gibi nehirde köprü ayakları arasına inşa edilen kapağın indirilmesiyle su yüzeyindeki yükseltinin artması sağlanır. Bu tip bağlamalar debi dalgalanmaları durumunda akarsu su seviyesini kontrol altında tutmaya yardımcı olur. Ayrıca kapaklar membada biriken sürüntü malzemesinin kolayca yıkanmasını sağlar.

Ancak kapakların sürekli yüksek dinamik etki altında olması bazı problemlere sebep olabilir. Bu da kapaklı bağlamaların dezavantajıdır, bu yüzden kapaklı bağlama yapılırken detaylandırılmış araştırmalar dikkate alınmalıdır (Şekil 2.5.).



Şekil 2.4. Mustafa Kemal Paşa Regülâtörü (kapaklı regülâtör - mansaptan görünüş) (Gürsoy, 2010)



Şekil 2.5. Konya - Çumra Kısıkyayla Regülâtörü - 1913 (kapaklı regülâtör) (mansaptan ve membadan görünüş) (Gürsoy, 2010)

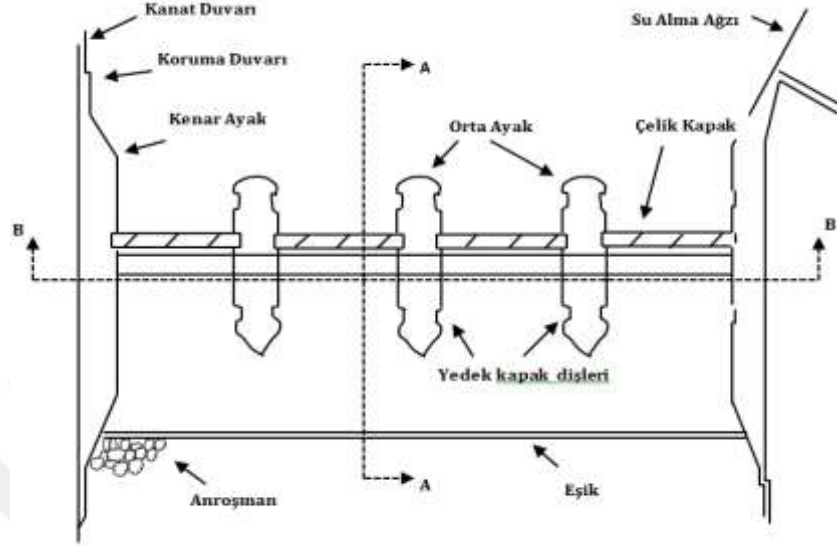
Ayaklar arasında açıklık bulunan ve ayakların üzerinde kapakların hareket ettiği yuvaların bulunduğu bağlama türüdür. Menba tarafındaki su seviyesi kapaklar sayesinde istenilen katta tutulur. Taşkın sırasında kapaklar açılarak bölgenin zarar görmesi engellenir (Tuna, 1985; Erkek ve Ağırlioğlu 2013).

Kapaklı bağlamalar katı maddelerin (buz, yüzen cisim) ve sürüntü malzemelerinin mansaba aktarılmasını kolaylaştırır (Berkün, 2005).

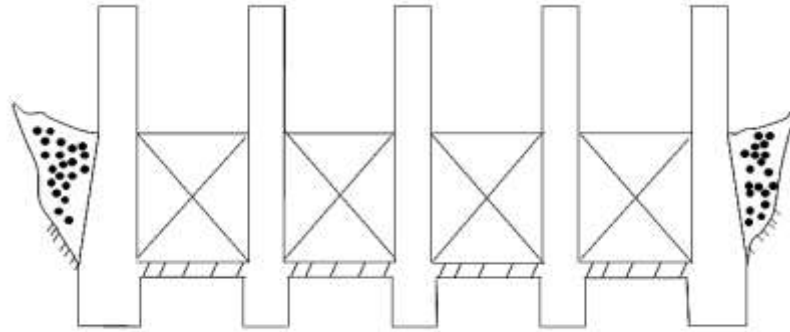
Hareketli Bağlama Yapı Elemanları (Şekil 2.6.):

- 1) Kapaklar,
- 2) Orta ayaklar,
- 3) Kenar ayaklar,

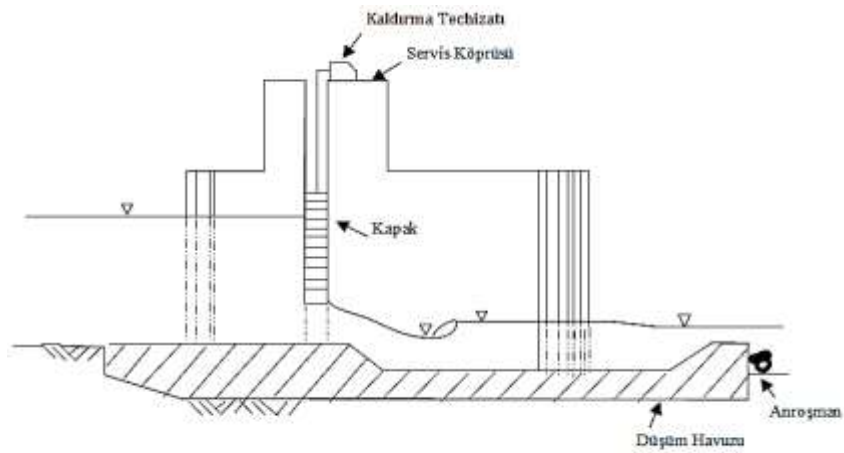
- 4) Düşüm yatağı,
- 5) Taş dolgu (Anroşman),
- 6) Sızdırmazlık yapıları (Palplanş, saplama duvarı, memba örtüsü).



(a) Plan



(b) B-B Kesiti



(c) A-A Kesiti

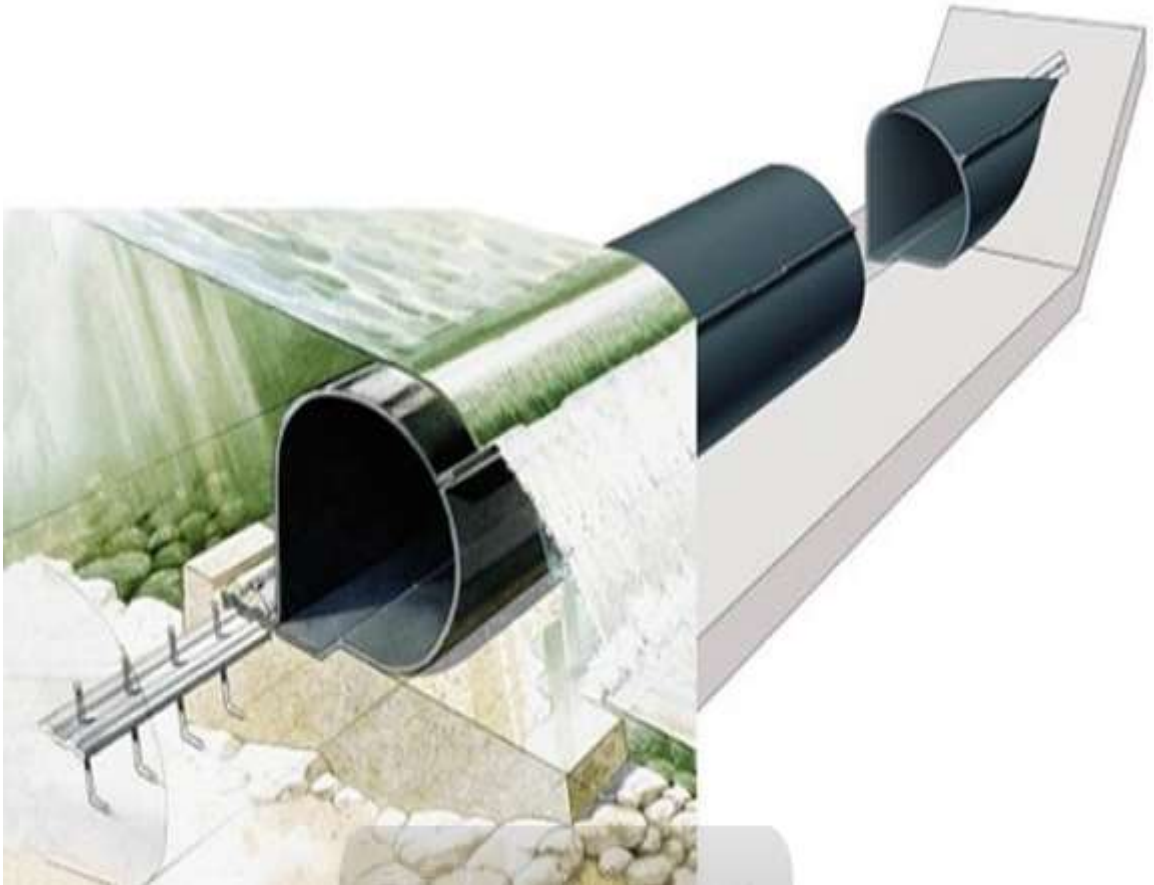
Şekil 2.6. Hareketli bağlama plan ve kesitleri

2.5 Karma Baęlamalar

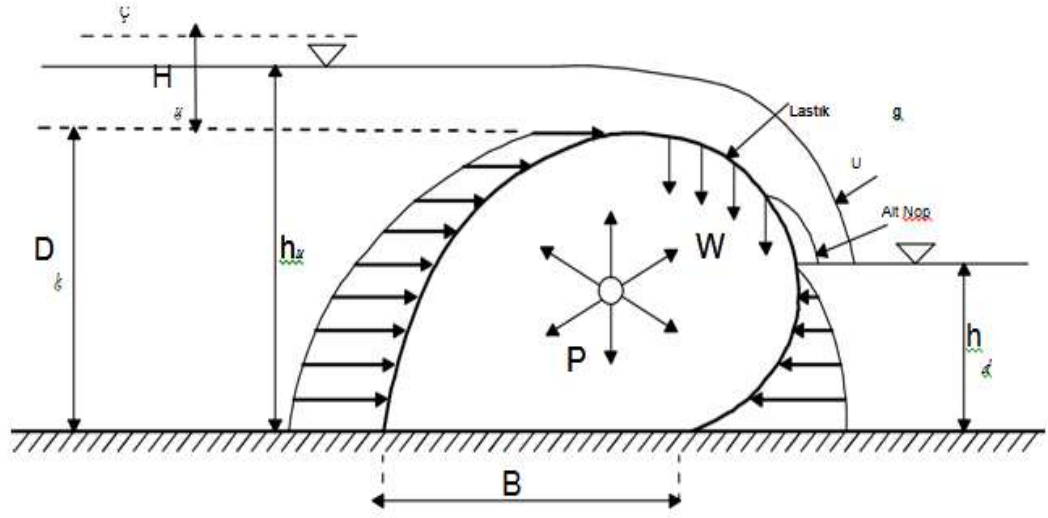
Bu tip baęlamalar dolusavaklı ve kapaklı baęlamaların birleřimiyle olan baęlamalardır.

2.6 řiřirilebilir Lastik Baęlamalar (Rubber Dam)

Yayvan ve geniř akarsu yataklarında, ok byk tařkın debilerine maruz kalan akarsuların zerinde, maksimum ve minimum debiler arasındaki farkın byk olduęu durumlarda yoęun srnt malzemesi ve rsup tařıyan akarsularda, akarsuların denize dkldę yerlerde yatakların doęal řartları korunarak otomatik ve/veya manel iřletme imkanı saęlayabilen lastik savaklı reglatrlerin projelendirilmesi uygundur (řekil 2.7(a)., řekil 2.7(b)., řekil 2.8., řekil 2.9.).



řekil 2.7(a). Lastik baęlama (Grsoy, 2010)



Şekil 2.7(b). Lastik bağlama (baraj) enkesidi



Şekil 2.8. Lastik savaklı regülatör (mansaptan görünüş) (Gürsoy, 2010)



Şekil 2.9. Aşağı Büyük Menderes Nehri üzerindeki lastik savaklı regülatör (işletme halinde) (Gürsoy, 2010)

2.7 Sifon Bağlamalar

Sabit bağlamalar gibi hareketli kapaklara sahip olmaları su seviyesinin düzenlenmesinde kullanılır. Bu bağlama tipinde feyez an veya debi sifonlar ile savaklanır, sifon savaklar memba su seviyesini mümkün olduğu kadar sabit tutar (Şekil 2.10.).



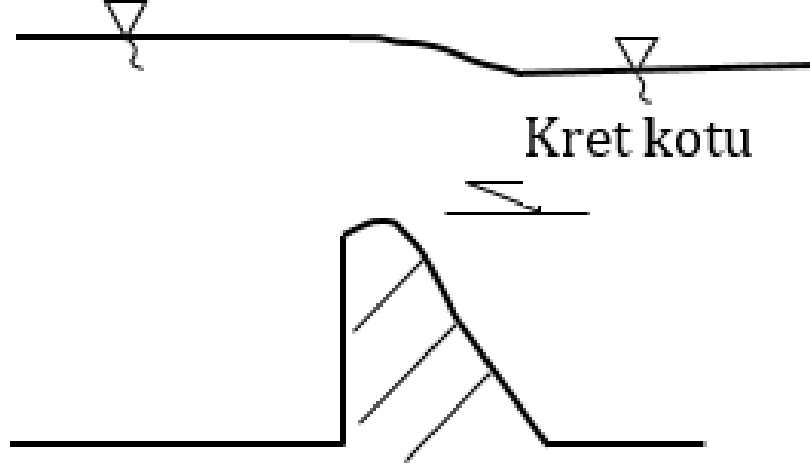
Şekil 2.10. Gönen Regülatörü (mansaptan ve membadan görünüş) (Gürsoy, 2010)

2.8. Mansap Su Seviyesine Göre Sınıflandırılması

Bağlamalar mansap su seviyesi ile kret kotunun yükseklik durumuna göre batmış ve serbest akımlı bağlama olarak ikiye ayrılır.

a) Batmış Akımlı Bağlama

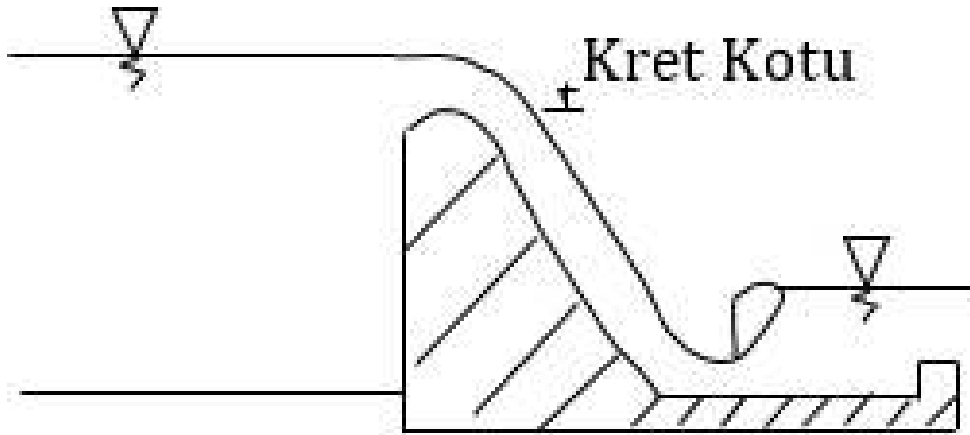
Bağlama kret kotu mansap su seviyesinin altında ise batmış bağlama adı verilir (Şekil 2.11.).



Şekil 2.11. Batmış akımlı bağlama

a) Serbest Akımlı Bağlama

Bağlama kret kotu mansap su yüksekliğinin üzerinde ise serbest akımlı bağlama olarak tanımlanır (Şekil 2.12.), (DSİ, 1988).



Şekil 2.12. Serbest akımlı bağlama

2.9. Baęlamalarda Yapı Elemanları

Çevre Duvarları veya Seddeleri: Menba ve mansap duvarları veya seddeleri diye ikiye ayrılırlar. Yapıyı planda sınırlayıp beton veya betonarme istinat duvarı olarak çalışırlar. Max su kotundan daha yüksekte yapıp suyun çevreye zarar vermesini engellerler (Şekil 2.13.).

Dolu Gövde veya Kapalı Kısım: Bir çevirme ve kabartma yapısıdır. Çünkü istenilen miktarda suyu toplayıp, kabartıp istenilen kotta prize (su alma yapısına) iletir. Gövde denilen sabit yapı veya hareketli bağlamalarda kapaklarla kabartma yapılır. Ayrıca dolu gövde ve kapaklar savaklama yapısı görevini görür. Akarsuda çok fazla katı madde varsa savak sırtı taş kaplanabilir veya bu kısmın agregası özel granülometri ile belirlenir. Savak sırtı çelik levha ile kuvvetlendirilip savak boyu da içi dolu gövde, eğik kırık veya eğri biçimde düzenlenerek uzatılabilir (Şekil 2.13.).

Çakıl Geçidi ve Ayırma Duvarı: Bağlamanın menba kısmında biriken katı maddeleri temizlemek amacıyla yapılmaktadır. Çakıl geçitleri sürüntü maddelerini tutarak su alma ağızına temiz su iletilmesini sağlar. Genişlikleri 2 – 6 m olarak seçilir. Gövde yüksekliği 2,5 m den fazla alanlarda çakıl geçitlerine dalgıç perde yapılması ekonomik açıdan kâr sağlar. Genişliği 50 m den fazla olan sabit bağlamalarda çift taraflı çakıl geçidi yapılması önerilir (Şekil 2.13.).

Düşü Havuzu (Enerji Kırıcı Havuz) : Mansap tarafında akımın sel rejiminden nehir rejimine geçişinde meydana gelen hidrolik sıçrama suyun enerjisinin çevrinti ile kırılmasını sağlayıp yatak tabanının oyulması ve aşınması önler. Düşü havuzu için kaya, kum ve çakıl iyi zemin oluşturur, öyle ki geçirimsiz zeminlerde düşü havuzu yapılmayabilir (Şekil 2.13.).

Anroşman (Taş Kaplama) : Düşü havuzunda kırılmayan enerjinin yatağı aşındırmasını önlemek amacıyla yapılır. Mansap parafuyunun taban kotundan başlanarak yapılır (Şekil 2.13.). En az 10 m olarak yapılır derinliği 75 cm yapılır.

Memba Blanketi ve Parafuylar: Bağlamadaki su kot farklılıklarından dolayı sızma zemindeki ince tanelerin yıkanmasına neden olduğu için zamanla borulanma meydana gelir. Bu nedenle zemin göçebilir sızma boyunu artırıp alttan kaldırmayı ve borulanmayı önlemek için memba blanketi ve parafuylar yapılır. Memba blanketi 5 x 5 m² 'lik anolar halinde dökölüp sızmayı önlemek için derzler lastik çantalı ve şaşırtmalı olarak yapılır. Plakların kalınlığı min 30 cm olmalı ki iyi bir sıkıştırma elde edilsin.

Parafuylar blanketin memba tarafında ve gerekli ise düşü havuzu tarafında zemine gömülmüş düşey duvarlardır (Şekil 2.13.).

2.10. Su Alma Yapısı Elemanları

a) **Çevre Duvarları:** Su alma ağzından suyun taşmasını engeller. İstinat duvarı olarak çalışır (Şekil 2.13.).

b) **Priz Giriş Yapısı:** Kanala su verilen kısımdır. Giriş kısmı ayaklı ve kapaklıdır. Çökelti maddelerinin kanala girişini önlemek için eşik üst kotu talveg kotundan 60 cm yukarıda yapılmalıdır. Giriş kısmındaki orta ayaklar arasına yerleştirilmiş kapakların açılması ile su alınır. Kapağın önünde dalgıç perdesi vardır. Dalgıç perdelerde yüzen maddelerin içeri girmesi engellendiği gibi kapak boyutlarında küçük tutulmuş olur. Yüzen maddenin geçişini önlemek için kapağın önüne ızgara da konulabilir (DSİ, 1977).

Izgaralar kalın, ince ve elektrikli olmak üzere üç çeşittir. Izgara yük kaybı $H = \frac{k.V^2}{2g}$ formülüyle hesaplanır. Burada k ızgara biçimine çubuk kalınlığına aralığına ve eğim açısına bağlı bir katsayıdır. Izgara aralıklarının küçülmesi katı madde açısından iyidir ama yük kaybını artırır. Elektrikli ızgaralar su canlılarının geçişini engeller.

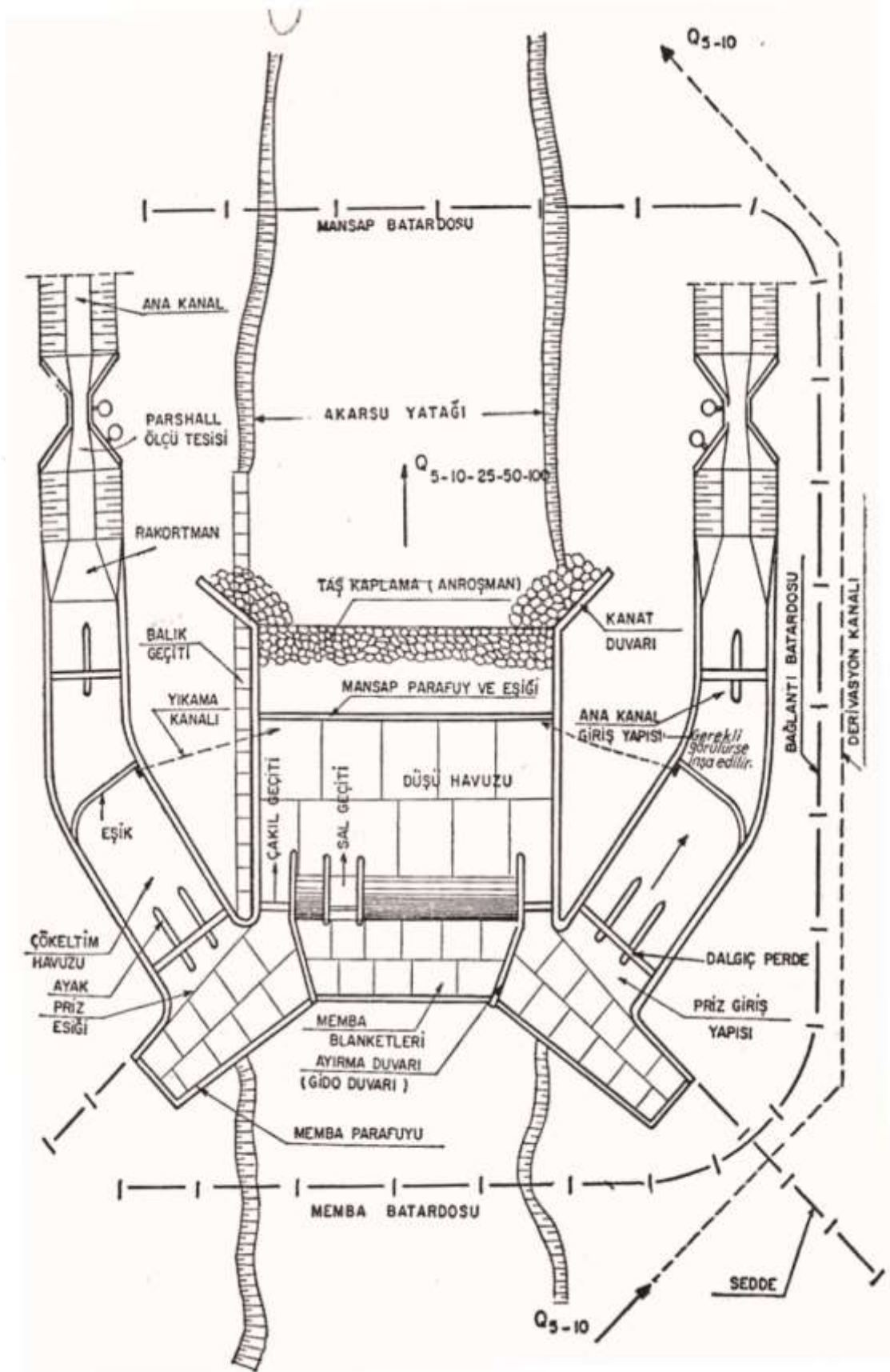
Bağlama akarsuyun kıvrımlı kısmında yapılacaksa ve priz tek tarafta olacaksa priz kıvrımın dış kısmına konulur. Böylece daha temiz su alınmış olur.

Prizin iki tarafta olmadığı durumlarda dolu gövde üzerine ızgara yapılarak su direkt olarak kanala aktarılır (Şekil 2.13.).

c) **Çökeltim Havuzu:** Gerekli tüm tedbirlere rağmen prizden geçebilen çökeltim malzemelerinin biriktirilmesini sağlar. Kesitleri didörtgen şeklinde olan bu havuzların derinlikleri fazladır. Çöken maddelerin kanala geçişini engellemek için havuz sonuna eşikler yapılır (Şekil 2.13.).

d) **Yıkama Kanalı:** Boşaltım ya da deşarj kanalı da denir. Çökeltim havuzunu temizlemek amacıyla inşa edilir (Şekil 2.13.).

e) **Ana Kanal Giriş Yapısı:** Gerekli durumlarda yapılır. Priz giriş yapısına benzer ana kanal için su ve çökelti maddesi kontrolünü sağlar.

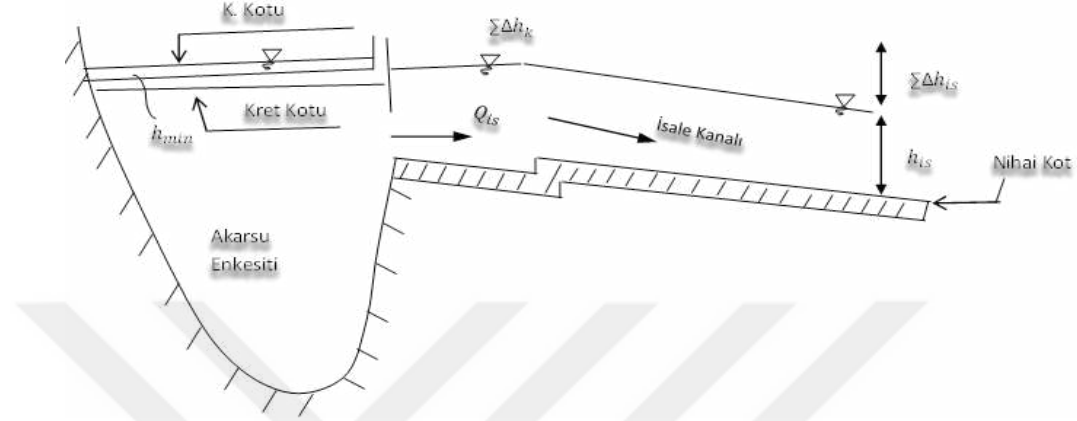


Şekil 2.13. Sağ ve sol sahilde sahilde sualma ağı bulunan bağlamanın plan görünümü (Özbek 2009)

2.11 Dolu Gövdeli Bağlamaların Hidrolik Hesapları

Bağlama Kret Kotunun Tayini

Kret kotunun belirlenebilmesi için kabartma kotunun bilinmesi gerekir (Şekil 2.14.).



Şekil 2.14. Kabartma kotu

$$\text{Kabartma Kotu} = \text{Nihai kot} + h_{is} + \sum h_{is} + \sum h_k \quad (2.1)$$

h_{is} = İsale kanalı akım yüksekliği

$\sum h_{is}$ = isale kanalındaki toplam kayıp miktarı

$\sum h_k$ = Su alma tesisindeki toplam kayıp

İhtiyaç debisinin kanalına alınmasında min debilerde kritik durum meydana geldiğinden kret kotunun belirlenmesinde iki durum söz konusu olmaktadır.

I. Durum

$Q_{\min} < Q_{is}$ olduğu durumda bağlama üzerinden su asmayacağından:

Kret Kotu = Kabartma Kotu + 0.1 alınarak emniyetli olacak şekilde dikkate alınmaktadır.

II. Durum

$Q_{\min} > Q_{is}$ olduğu durumda $Q'_{\min}$ debi büyüklüğünde bağlamanın üzerinden su aşar.

$Q'_{\min} = Q_{\min} - Q_{is}$ bağlama kret kotu kabartma kotudan savak yükü çıkarılarak bulunur.

$$Q'_{min} = \frac{2}{3} \cdot \mu \cdot h_{min} \cdot \sqrt{2gh_{min}} \quad (2.2)$$

$$h_{min} = \sqrt[3]{\left(\frac{3Q'_{min}}{2\mu b \sqrt{2g}}\right)^2}$$

Kret kotu = Kabartma kotu - h_{min} (2.4)

Burada h_{min} = Savak yükünü gösterir.

2.12. Bağlama Genişliği (Kret Uzunluğunun) Bulunması

Bağlama genişliği doğrudan maliyeti etkilediği için kısa tutulma istenir. Kısa tutulması da savak yükünün artırılmasıyla mümkündür. Savak yükünün artırılması durumunda debinin enerjisi fazla olacağı için mansaptaki oyulmalar ve taban su basıncı artar. Bu sebeple birim genişlikten geçen debinin düşüm yatağındaki hızı max 15 m/s olacak şekilde belirlenmelidir.

Sabit bağlamaların uzunluğu tahmini olarak birim genişlikten geçen debinin $q = 5 \text{ m}^3/\text{s.m}$ olacak şekilde belirlenebilir. Büyük taşkın debisi olan akarsularda bu değer esas alındığında sabit bağlama genişliği çok büyük çıkacağından özellikle baraj dolu savaklarında $20 - 30 \text{ m}^3/\text{s.m}$ gibi çok büyük birim debilere müsaade etmek zorunda kalınır. Bu durumda aşağıdaki çözümler uygulanır.

- 1) Uygun zemin şartlarında taşkın esnasında 8 – 10 m varan savak yüklerine müsaade etmek
- 2) Dolu gövde tepesinde kapaklar öngörmek
- 3) Sabit bağlama yerine hareketli bağlama seçmek (Ağırlioğlu, Erkek, 2013)

Bağlama uzunluğu başlangıçta çakıl geçidi ve diğer yaptırımlar da dahil olmak üzere akarsu yatağında fazla bir daralma veya genişleme yapmayacak şekilde seçilmelidir. Prensip olarak bağlama uzunluğu buz yığılmalarını ve erozyonu önlemek için akarsu yatağı genişliğinin 0,5 - 0,6 katından daha küçük olmamalıdır. (Ağırlioğlu, Erkek, 2013)

Sabit bağlamalarda çakıl geçidi üzerine yapılan servis köprüsü ayakları ve kenar ayakları akım çizgilerinde bir büzölmeye sebep olur. Bu yüzden etkili bağlama tepe uzunluğu aşağıdaki formülle bulunur.

$$B_e = b - 2(nK_0 + K_a)h \quad (2.5)$$

Burada: B_e = Etkili bağlama tepe uzunluğu

b = Ayaklar arasındaki toplam bağlama tepe uzunluğu

n = Orta ayak adedi

K_0 = Orta ayakların büzülme katsayısı (Yuvarlak başlıklı ayaklarda 0,035 – 0,01 dikdörtgen kesitli ayaklarda 0,1, sivri uçlularda 0,00 olarak alınır.)

K_a = Kenar ayakların büzülme katsayısı (0,00 ile 0,2 arasında değişir, genellikle K_0 in 2,5 katı alınır.

h = Savak üzerindeki su yüküdür.

2.13 Savak Kapasitesi

1. Serbest Akım Durumu

Sabit bağlama üzerinden savaklanan debi aşağıdaki formülle bulunur (Şentürk, 1988).

$$Q = \frac{2}{3} \cdot \mu \cdot b \cdot \sqrt{2g} \left[\left(h + \frac{V_0^2}{2g} \right)^{\frac{3}{2}} - \left(\frac{V_0^2}{2g} \right)^{\frac{3}{2}} \right] \quad (2.6)$$

Q = Toplam debi

μ = Savak katsayısı

b = Bağlama net uzunluğu

V_0 = Yaklaşım hızı

h = Savak yükü

g = Yerçekim ivmesi

$\left(\frac{V_0^2}{2g} \right)$ savak yüküne göre çok küçük olduğundan ihmal edilir.

$$Q = \frac{2}{3} \cdot \mu \cdot b \cdot \sqrt{2g} \cdot h^{\frac{3}{2}} \quad (2.7)$$

eşitliği elde edilir.

1. Batmış Akım durumu

Batmış akım durumunda savakların su miktarı orifis ve serbest akım formülleri kullanılarak hesaplanır.

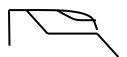
$$Q = Q_1 + Q_2 = \frac{2}{3} \mu_1 \cdot b \cdot \sqrt{2g} \left[\left(h + \frac{v_0^2}{2g} \right)^{\frac{3}{2}} - \left(\frac{v_0^2}{2g} \right)^{\frac{3}{2}} \right] + \mu_2 \cdot ab \cdot \sqrt{2g} \cdot \sqrt{h + \frac{v_0^2}{2g}} \quad (2.8)$$

Yaklaşım hızı ihmal edilirse

$$Q = Q_1 + Q_2 = \frac{2}{3} \mu_1 \cdot b \cdot \sqrt{2g} \cdot h^{\frac{3}{2}} + \mu_2 \cdot ab \cdot \sqrt{2gh} \quad (2.9)$$

Kret şekli ve akım durumuna göre μ_1 ve μ_2 savak katsayıları Tablo 2.1.' deki gibi seçilir.

Tablo 2.1. μ_1 ve μ_2 savak katsayıları (Erkek ve Ağırlioğlu, 2013)

Kret Şekli	Batmamış Akım (μ)	Batmış Akım $\mu_1 = \mu_2$			
	0,47-0,51	h/z			
	0,50-0,55				
	0,65-0,73	1	2	3	4
	0,64	0,64	0,63	0,64	0,67
	0,75	0,75	0,74	0,77	0,81
	0,79	0,79	0,78	0,81	0,86

2.14. Çakıl Geçidi

Çakıl geçitlerinde büyük kapaklar yüksek maliyetli olduğundan 3 – 3,5 m genişliğinde yapılması daha uygun olur. Çakıl geçitleri dolu gövdeden gido duvarı ile ayrılır. Sürüntü malzemesinin çok olduğu akarsularda çakıl geçidinin ebatlarını artırmak yerine birden fazla çakıl geçidi inşa etmek daha uygundur. Yüksekliği 2,5 m ye kadar olan bağlamalardaki çakıl geçitlerine bağlama yüksekliği kadar kapak yapılır. Daha yüksek bağlamalarda ise kapak ile dalgıç perde yapılması işletme ve ekonomi açısından daha avantajlıdır.

1. Dalgıç Perdesiz Çakıl Geçidi

Dalgıç perdesiz çakıl geçitlerinde kapağın tam açılması durumunda akım batmış akıma benzer olacağı için batmış savak formülü kullanılır (Tuna, 1985).

$$Q = Q_1 + Q_2$$

$$Q = \frac{2}{3} \cdot \mu \cdot b \cdot \sqrt{2g} \cdot \left[(h_1 + h_a)^{\frac{3}{2}} - h_a^{\frac{3}{2}} \right] + \mu \cdot b \cdot a \sqrt{2gh_1}$$

$$h_a = \frac{V_0^2}{2g} \cong 0 \quad \text{alınırsa}$$

$$Q = \frac{2}{3} \cdot \mu \cdot b \cdot \sqrt{2g} \cdot h_1^{\frac{3}{2}} + \mu \cdot b \cdot a \sqrt{2gh_1} \quad (2.10)$$

$\mu \cong 0,65$ alınabilir.

2. Dalgıç Perdeli Çakıl Geçidi

Dalgıç perde olması halinde akım orifisteki akım gibi olacağı için batmış ve batmamış duruma göre hesaplanır (Tablo 2.2).

- $a < 0,2 \cdot h$ olması durumunda

$$Q = \mu \cdot b \cdot a \cdot \sqrt{2gh_1} \quad (2.11)$$

- $a > 0,2 \cdot h$ olması durumunda

$$Q = \frac{2}{3} \mu \cdot b \cdot \sqrt{2g} \cdot \left(h_2^{\frac{3}{2}} - h_1^{\frac{3}{2}} \right) \quad (2.12)$$

b = Çakıl geçidinin net genişliği

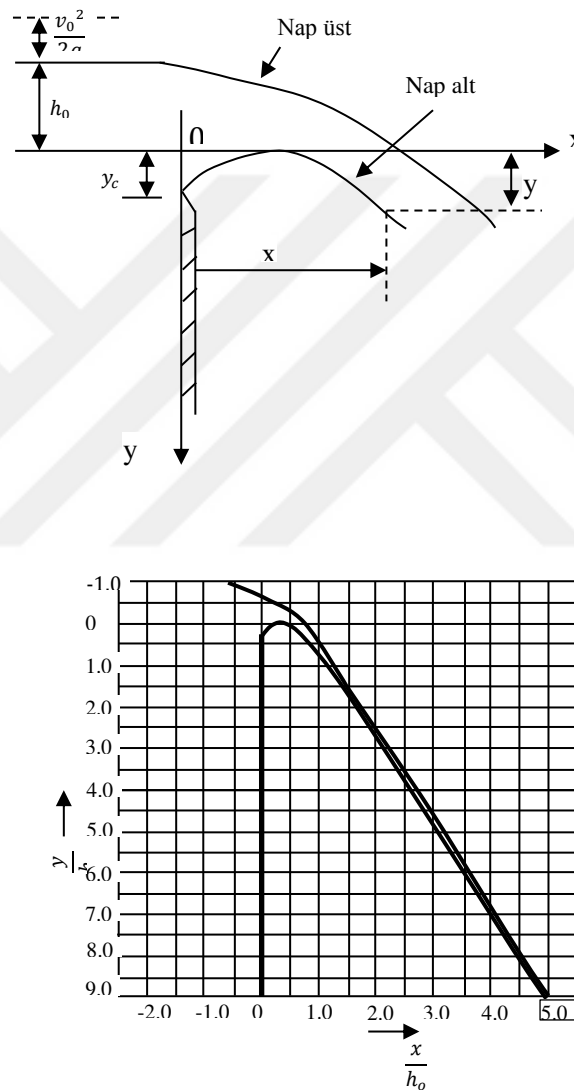
μ = Orifis katsayısı 0,65 – 0,70 arasında alınır.

Tablo 2.2 Orifis katsayısı değerleri (Tuna, 1985)

a/b	0	0.5	1	2
μ	0.678	0.64	0.582	0.438

2.15. Hidrolik Profilin Tayini

Sabit bağlamalarda hidrolik profil, dolu gövdenin üzerinden aşan su şekline uygun olmalıdır. Bu nedenle akımın, havalandırılmış su napının keskin kenarlı bir savaktan geçerken oluşturduğu alt yüzeyinin şekline uygun olarak projelendirilir (Şekil 2.15.). Nap alt yüzeyinin dolu gövde olduğu düşünülerek hidrolik profil elde edilir (Erkek ve Ağralıoğlu, 2013).



Şekil 2.15. Keskin kenarlı savak üzerindeki akım (Erkek ve Ağırlioğlu, 2013)

Hidrolik profil küçük debideki savak yüküne göre belirlenirse daha büyük debi geldiği zaman bağlama üzerinden geçen akım ileriye doğru kayar. Bu nedenle hidrolik profilin yüzeyi ile su napının alt yüzeyi arasında vakum oluşacağı için beton yüzeyinde

delikler oluşur. (Kavitasyonu önlemek için hidrolik profilin tayini max debideki su napının alt yüzeyine göre belirlenir.

Creager $h = 1$ m savak yükü için oluşan su napının alt ve üst yüzeylerinin koordinatlarını belirlemiştir. (Tablo 2.3) 1 m den büyük savak yükleri için Creager tablosu kullanarak hem x hem de y değerleri proje savak yükü ile çarpılır hidrolik profil tayin edilir.

Tablo 2.3. Creager profiline göre alt ve üst nap değerleri

x	y ₁ (Nap altı)	y ₂ (Nap üstü)	x	y ₁ (Nap altı)	y ₂ (Nap üstü)
0.00	0.126	-0.831	1.40	0.590	-0.030
0.10	0.036	-0.803	1.70	0.920	0.305
0.20	0.007	-0.772	2.00	1.315	0.693
0.30	0.000	-0.740	2.50	2.100	1.500
0.40	0.007	-0.702	3.10	3.110	2.500
0.60	0.063	-0.620	3.50	4.260	3.660
0.80	0.153	-0.511	4.00	5.610	5.000
1.00	0.267	-0.380	4.50	7.150	6.540
1.20	0.410	-0.219			

Hidrolik profilin bulunmasında aşağıdaki bağıntı da kullanılabilir.

$$y = k \cdot h_0^{1-n} \cdot x^n$$

$$k = 0.461 - 0.556$$

$$n = 1.8 - 2$$

Creagerde

$$k = 0.47$$

$$n = 1.8 \text{ dir.}$$

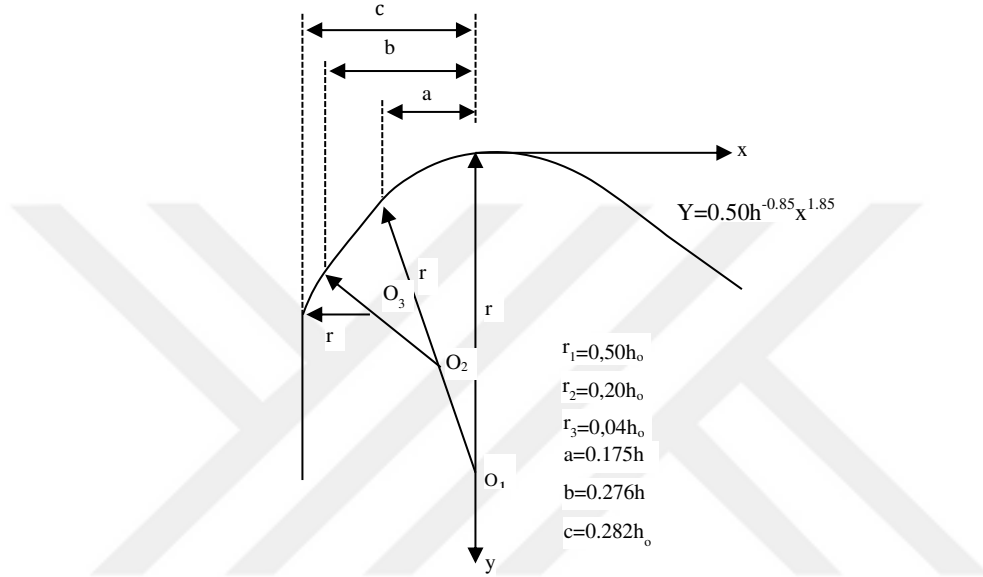
Uygulamada ogee hidrolik profili kullanılır (Şekil 2.16.)

$$\text{Ogee hidrolik profili için } \frac{\omega}{h_0} > 0.5$$

$$y = \frac{0,5}{h_a^{0,85}} \times 1.85$$

ω = Bağlama yüksekliği

$$h_0 = h + \frac{V_0^2}{2g} \quad \text{olmak üzere savak proje yükü}$$



Şekil 2.16. Ogee hidrolik profili (Erkek ve Ağralıoğlu, 2013)

2.16. Düşü Havuzlu Hesapları

Dolu gövdeli ve hareketli bağlamalardaki akım menbadan mansaba geçerken sel rejimindedir. Sel rejimindeki akımların hızı çok yüksektir. Mansab kısmındaki akım sürekli sel rejiminde devam etmez. Nehir rejimine geçiş olur. Bu geçiş sırasında hidrolik sıçrama meydana gelir. Hidrolik sıçrama bağlamanın taban ve kıyılarında oyulmalara (borulanma) sebep olur. Bağlamaların mansabında meydana gelecek oyulmaları engellemek için düşü havuzu (enerji kırıcı havuz) inşa edilir. Böylece akımın enerjisi sönmölenir.

h_1 = Hidrolik sıçramadan önceki su derinliği

h_2 = Hidrolik sıçramadan sonraki su derinliği

h_3 = Mansabdaki su derinliği (Hassas anahtar eğrisinden bulunur)

h = Kabarma yüksekliği

a_d = Düşü havuzu derinliği

Dolu gövdeli bağlamalarda Şekil 2.17'deki a ile 1 kesitleri arasında Bernoulli denklemi uygulanarak bulunur.

$$z + p + h + H_0 = h_1 + \frac{V_1^2}{2g} + \Delta H \quad (2.13)$$

z = Karşılaştırma düzlemine olan uzaklık

$H_0 = h + \frac{V_0^2}{2g}$ = Kretten itibaren enerji yüksekliği

V_0 = Suyun dolu gövdeye yaklaşım hızı

ΔH = Enerji kaybı = $0.1 \frac{V_1^2}{2g}$

$$V_1 = \frac{Q}{B \cdot h_1} \quad (2.14)$$

B = Dolu gövde genişliği

$h_2 = h_1$ ile h_2 arasında implus momentum teoremine göre aşağıdaki formüle göre belirlenir.

$$\frac{h_1}{h_2} = \frac{1}{2} \left[\sqrt{[8Fr^2 + 1]} - 1 \right] \quad (2.15)$$

$$Fr = \frac{V_1}{\sqrt{gh_1}}$$

Yukarıdaki formüllere dayanarak sel rejimi olabilmesi için $h_1 < h_{kr}$ olacak şekilde deneme yanılma yöntemiyle h_1 bulunur. Daha sonra (3) formülünde yerine yazılarak h_2 bulunur.

Bir bağlamada hidrolik sıçrama üç şekilde oluşabilir, bunlar aşağıdaki şekillerdeki gibidir.

h_2 = Sıçramadan sonraki derinlik

h_3 = Mansap suyu derinliği

Bağlamalarda akım tipine göre sıçramanın meydana gelip gelmeyeceği belirlenir.

$$Fr = 1 \text{ ise kritik akım } (V = V_{kr}, h = h_{kr} = \sqrt[3]{\frac{Q^2}{g \cdot B^2}})$$

$Fr > 1$ ise sel rejimi ($V > V_{kr}, h < h_{kr}$)

$Fr < 1$ ise nehir rejimi ($V < V_{kr}, h < h_{kr}$)

$Fr > 1$ ise sıçrama meydana gelir diğer durumlarda sıçrama meydana gelmez.

1. Durum $h_2 = h_3$ olduğu zaman hidrolik sıçrama bağlamanın önünde olur.

$h_2 = h_3$ olduğu durum en iyi durumdur.

2. Durum $h_2 < h_3$ batmış sıçrama durumudur. 1. Duruma göre sıçrama daha önde gerçekleşir.

3. Durum $h_2 > h_3$ olması halinde akım hidrolik sıçrama için gerekli su yüksekliğini h_3 değerini bulamadığı için yeterli bir h_1 derinliğinin hasıl olacağı yere devam eder. $h_2 = h_3$ şartı gerçekleştiği zaman hidrolik sıçrama meydana gelir. (Özbek, 1987)

3. duruma göre enerjinin sönmülenebilmesi için düşü havuzu derinliği $a = h_2 - h_3$ kadar olmalıdır.

a değerinin belirlenebilmesi için aşağıdaki yol izlenir. İlk olarak tahmini derinlikte (a) düşü havuzu kabul edilir. 1 eşitliğinde tatonmanla $h_1 < h_{kr}$ olacak şekilde h_1 bulunur, 3 eşitliğinden h_2 bulunur.

$h_2 - h_3 = a$ değerini verirse ilk seçilen derinlik uygun olur. Değer farklı ise yeni bir a kabul edilerek yukarıdaki işlemler tekrarlanır. Max hava derinliği 5 farklı debi için işlem yapılarak bulunur. Çok karışık olan bu işlemler yerine bazı hazır tablo ve grafiklerden h_1 ve h_2 tayin edilebilir.

2.16. Düşü Havuzu Boyunun Bulunması (Hidrolik Sıçrama Uzunluğu)

Düşü havuzunun boyu bulunurken hidrolik sıçramanın uzunluğu dikkate alınır. Bunun için Fr sayısı kullanılır.

$$Fr = \frac{V_1}{\sqrt{gh_1}} \text{ sonucuna göre havuz tipi belirlenir.}$$

$Fr > 2,5$ ise radye tipi havuz yapılır.

$2,5 < Fr < 4,5$ için Tip I

$Fr > 4,5$ $V_1 < 15 \text{ m/s}$ için Tip II

$Fr > 4,5$ $V_1 > 15 \text{ m/s}$ için Tip III

Sıçrama uzunluğu $L/h_2 - Fr_1$ arasında çizilen grafiklerden seçilir. Sıçrama uzunluğu veya düşüm yatağı uzunluğu formüllerinden bazıları aşağıdaki gibidir.

Bakmateff	: $L_D = 5(h_2 - h_1)$	Serbest düşüm yatağı uzunluğu
Brodly Peterka	: $L_D = 6.13 h_2$	Serbest düşüm yatağı uzunluğu
Kazeny	: $L_D = gh_1(Fr_1 - 1.57)$	Sıçrama uzunluğu
Safranez	: $L_D = 6h_1Fr_1$	Serbest düşüm havuzu uzunluğu



3. SABİT BAĞLAMA TASARIMI İÇİN PROGRAM GELİŞTİRİLİRKEN KULLANILAN TEKNOLOJİLER

Bu çalışma kapsamında program geliştirilirken aşağıdaki teknolojiler kullanılmıştır.

JavaScript: Yaygın olarak web tarayıcılarında kullanılan etkili bir programlama dilidir. JavaScript kullanılarak yazılan istemci tarafı betikler ile tarayıcının kullanıcıyla etkileşimi sırasında tarayıcının kontrolü asenkron şekilde sunucu ile iletişime geçilmesi, web sayfasının içeriğinin değiştirilmesi gibi görevler sağlanır. JavaScript, Node.js vb platformlar sayesinde sunucu tarafında da sıkça kullanılmaktadır (<https://tr.wikipedia.org/wiki/JavaScript>).

Html: Hiper Metin İşaretleme Dili (İngilizce *Hypertext Markup Language*, ksc. HTML) web sayfalarını oluşturmak amacıyla kullanılan standart metin işaretleme dili HTML'dir. Bu bir programlama dili olarak ifade edilemez çünkü HTML kodlarıyla tek başına çalışan bir program yapılamaz. Ancak Hiper Metin İşaretleme Dilini yorumlayabilen ve çalışabilen programlar yazılabilir. Programlama dili olarak tanımlanamamasının sebebi tam olarak budur. Esas amacı video, görüntü, yazı gibi farklı verileri ve bunlardan oluşan sayfaları birbirine kolayca bağlamak, bunların yanı sıra bahsi geçen sayfaların web tarayıcısı yazılımları tarafından doğru bir şekilde görüntülenmesi için lüzumlu kuralları saptamaktır. Hiper Metin İşaretleme Dilini kodunu web tarayıcıları okur, görsel hale dönüştürür ve yorumlar bu yüzden aynı HTML kodunun çeşitli tarayıcılarda çeşitli sonuç vermesi ihtimali yüksektir (Tim Berners-Lee, "Information Management: A Proposal." CERN (March 1989, May 1990).

Css (Cascading Style Sheets): (Basamaklı Biçim Sayfaları ya da Basamaklı Stil Şablonları olarak bilinen ksc. CSS'dir). HTML'e ek olarak format biçimlendirme ve metin alanında ekstra olanaklar sağlayan bir web teknolojisidir. İnternet sayfaları için kabul gören şablonlar oluşturma imkanı sağladığı gibi, bağımsız olarak harflerin stilini, yani yazı tipi, renk, büyüklük gibi özelliklerini değiştirmek amacıyla kullanılabilir. Css'in en etkili özelliği kullanımındaki esnekliktir.

<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQ2FzY2FkaW5nX1NeWxlX1NoZWV0cw>).

Bootstrap: Twitter Bootstrap (ksc. Bootstrap): Açık kaynak kodlu, web sayfaları ya da uygulamaları geliştirmek amacıyla kullanılabilecek araçların tümü ve önyüz çatısı. Bootstrap, web sayfaları ya da uygulamalarında kullanımı sağlanacak, CSS ve HTML tabanlı tasarım şablonlarını kapsar. Bu şablonlar navigasyon çubuğu, form, buton vb arayüz bileşenleri oluşturulurken kullanılabilmektedir. Temmuz 2015'den itibaren Bootstrap, Github üzerinde yaklaşık 83 binin üzerinde "star" ve 33 binin üzerinde "fork" sayılarına erişerek, sitenin en çok tanınan projelerinden biri olmuştur. <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQm9vdHN0cmFwXyJdtm55w7x6X8OnYXTEsXPEsSk>).

Chart. js: Grafikler çizmek amacıyla kullanılan bir javascript kütüphanesidir.

Python: Nesne yönelimli, birimsel (modüler), yorumlamalı ve etkileşimli yüksek seviyeli programlama dili olarak bilinir. Girintilere bağlı basit sözdizimi, dilin öğrenilmesini aynı zamanda akılda kalmasını kolaylaştırır. Bu özellik ona söz diziminin ayrıntılarıyla zaman kaybetmeden programlama yapabilmeyi sağlayan bir dil olma özelliği imkanı sunar.

Modüler yapısı, sınıf dizgesini ve her çeşit veri alanı girişini destekler. Neredeyse her türlü platformda çalışabilmektedir (Windows, Unix , Linux, Mac, , Amiga, Symbian). Python kullanılarak sistem, ağ, web, uygulama ve veritabanı yazılımı programlama gibi çeşitli alanlarda yazılım geliştirebilir. Büyük yazılımların çabuk bir şekilde prototiplerinin oluşturulması ve kullanımın denenmesi gerektiği zamanlarda da C veya C++ gibi dillere tercih edilir. (<https://docs.python.org/3/faq/general.htm>).

NumPy: Python programlama dili için bir koleksiyondur, büyük, matrisler ve çok boyutlu diziler için destek sağlar. Bu dizilerde çalışacak büyük bir üst düzey matematiksel işlev kütüphanesi sunar. (<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTnVtUHk>).

Matplotlib: Sayısal matematik uzantısı ve Python programlama dili olan NumPy için bir çizim kütüphanesidir. (<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvTWF0cGxvdGxpYg>).

Flask: Restfull servis yazabilmek için bir python kütüphanesidir.

Canvas: Web tarayıcısı üzerinde çizim yapabilmek amacıyla kullanılan html tagıdır.

Npm: Javascript betik dili için oluşturulmuş ve Node.js'in standart olarak kullandığı bir paket yönetim sistemidir. Npm komut satırından çalıştırılır, uygulamalar için bağımlılık yönetimini sağlar. (Schlueter, Isaac Z. (25 Mar 2013). "Forget CommonJS. It's dead. **We are server side JavaScript.**". *GitHub*).

Node.js: Sunucu tarafında çalışan, açık kaynaklı ve ağ bağlantılı uygulamalar için yapılmış bir çalıştırma ortamıdır. Node.js uygulamaları genellikle istemci tarafı betik diliyle JavaScript kullanılarak geliştirilir. (<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTm9kZS5qcw>).

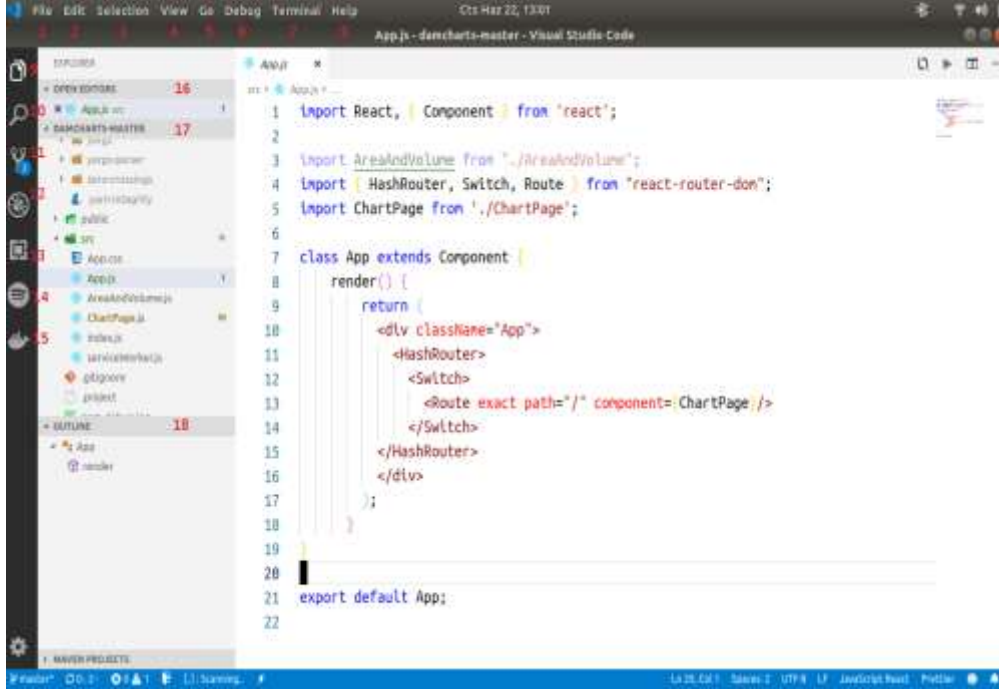
Axios: Tarayıcılar ve Node.js için bir http client'dır.

Visual Studio Code: Microsoft tarafından Windows, MacOS ve Linux için oluşturulan kaynak kodu düzenleyicisidir. Gömülü Git kontrolü, hata ayıklama, akıllı kod tamamlama, sözdizimi vurgulama, snippet'ler ve kod yeniden yapılandırma desteği içermektedir. (<http://arstechnica.com/information-technology/2015/11/visual-studio-now-supports-debugging-linux-apps-code-editor-now-open-source/>).

PyCharm: Çapraz platformda Python geliştirme ortamıdır. Grafiksel hata ayıklamacısı, kod analizleri, versiyon kontrol sistemi ile birleştirilmiş ve Django ile Python web geliştirmelerini sağlamaktadır. Çapraz platformu OS X, Windows ve Linux işletim sistemleri üzerinde çalışmaktadır. (<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvUHlDaGFyYQ>).

4. SABİT BAĞLAMA TASARIMI İÇİN YAZILIM GELİŞTİRİLMESİ

4.1 Yazılımın Yapım Aşamaları



Şekil 4.1. Metin (text) editörü *visual studio code*

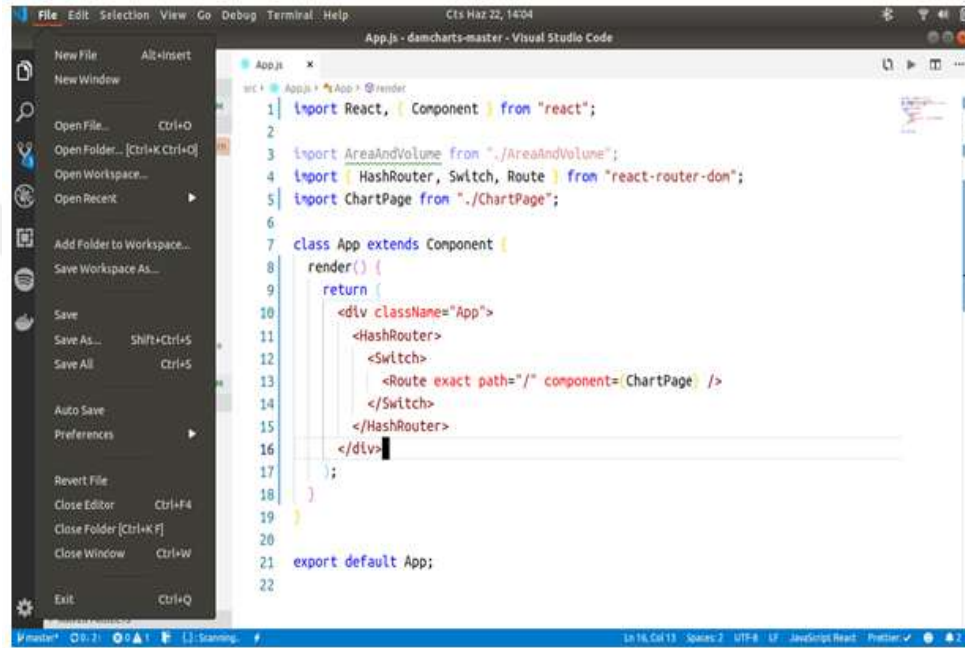
Şekil 4.1 uygulamayı geliştirirken kullanılan metin (text) editörü *visual studio code*'du gösterir.

Butonlar ve bunların kullanım amaçları aşağıda belirtilmiştir.

- 1) Uygulamadan çıkma, dosya açma, kapama gibi işlemler yapılır.
- 2) Kodda bir değişiklik yaptıktan sonra geri alma ileri gitme gibi işlemlerin yapılmasını sağlar.
- 3) Kod satırı seçme işlemleri yapılır.
- 4) Uygumanın farklı sekmeleri arasında geçiş yapmaya olanak sağlar.
- 5) Kodda bulunan implementasyonlara, methodlara gitme işlemlerini yapar.
- 6) Uygulamanın debug modda çalışması sağlanır.
- 7) External komut satırı açılır.
- 8) Web üzerinden yardım alabileceğimiz internet sayfaları açılır.
- 9) Açtığımız klasör içindeki dosyaları görüntüler.

- 10) Açtığımız klasör içinde arama yapmamızı sağlar.
- 11) Debug modda çalıştırdığımız zaman koyduğumuz breakpointleri incelememizi sağlar.
- 12) Yüklediğimiz eklentileri gösterir ve eklenti silmeye, yeni eklenti yüklemeye yardımcı olur.

Program çalışır konumu getirildikten sonra file butonun üzerine gelinir. New file butonu seçilir ve burada bir javascript dosyası oluşturulur (Şekil 4.2.).



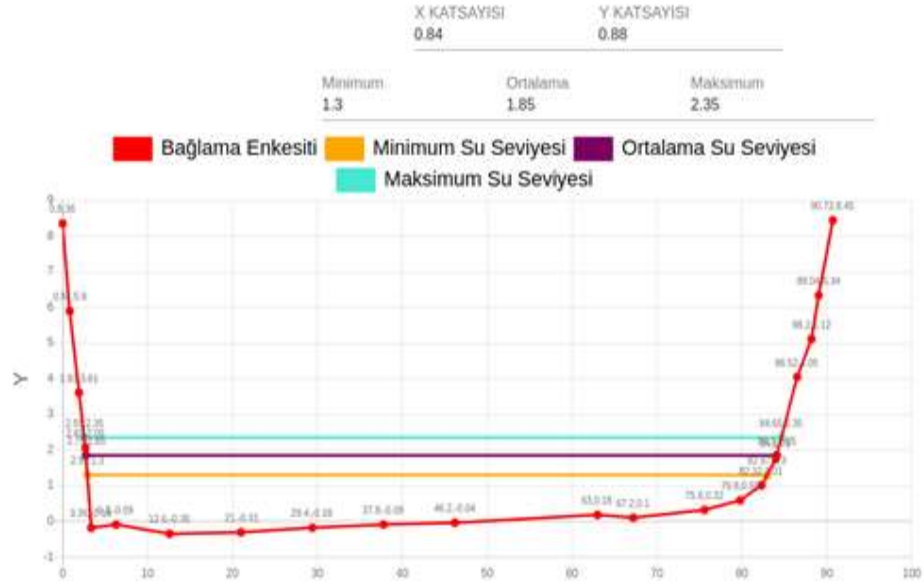
Şekil 4.2. Javascript dosyası oluşturma

Bağlama Projesi kodu yazılarak uygulamaya ad verilir. Elimizde bulunan bağlama projesi föyündeki x ve y katsayıları yine föyde verilmiş olan alfa ve beta katsayılarıyla çarpılır. x ve y katsayılarının kullanıcı tarafından girilmesi Ek-1'deki kod ile sağlanmaktadır.

Bulunan x, y değerleri ve bağlama projesi föyünde verilen enkesitten geçen minimum ortalama maksimum su seviyeleri girilerek bağlama en kesiti çizilir. Ek-2.

Yukarıdaki kodlar yazılarak maksimum minium ve ortalama su seviyelerini dışarıdan girilmesine müsaade edilir. Minimum ortalama ve maksimum su seviyelerinin y kordinatlarının bulunabilmesi için Thales Teoremi Ek-3' teki gibi yazılmıştır.

Tüm bu değerlerle beraber bağlama enkesiti çizilir (Şekil 4.3.). Ek-4' te yazılan kodlarla beraber değerler grafik üzerine aktarılır.



Şekil 4.3. Bağlama enkesiti

Cross Metodu yardımıyla minimum, ortalama, maksimum su seviyelerinin alanı çevrilerek hesaplanır. Alan hesabı için yazılmış kod Ek-5’teki gibidir.

Bulunan alan ıslak çevreye bölünerek hidrolik yarıçap Ek-6’daki gibi hesaplanır (Tablo 4.1).

Ek-6’da bulunan değerler Ek-7’deki kodlara yazılarak ekran tasarımı yapıp, ekranda gösterilir.

Tablo 4.1 Akarsu enkesitinin ıslak alan, ıslak çevre ve hidrolik yarıçap hesapları

Çevre Hesabı	
Minimum Çevre	81.27m
Ortalama Çevre	83.11m
Maksimum Çevre	84.39m
Alan Hesabı	
Minimum Alan	103.89m ²
Ortalama Alan	148.3m ²
Maksimum Alan	189.15m ²
Yarıçap Hesabı	
Minimum Yarıçap	1.28m
Ortalama Yarıçap	1.78m
Maksimum Yarıçap	2.24m

Akarsu genel pürüzlülüğü katsayısı ve akarsu eğimi için tanımlanmış katsayılar oluşturulmuştur. Bu katsayılar Ek-8’ deki kodla tasarlanmıştır

Bu katsayılar içinden herhangi biri seçilip $Q = \frac{1}{n} \times R^{2/3} \times J_o^{1/2} \times A$ formülü kullanılarak minimum, ortalama ve maksimum debi hesaplanır. Debi hesabı için Ek-9’deki kod yazılmıştır.

Talveg kotu yardımıyla minimum, ortalama ve maksimum yükseklik hesaplanır ve bunun için Ek-10’ daki kod yazılmıştır (Tablo 4.2).

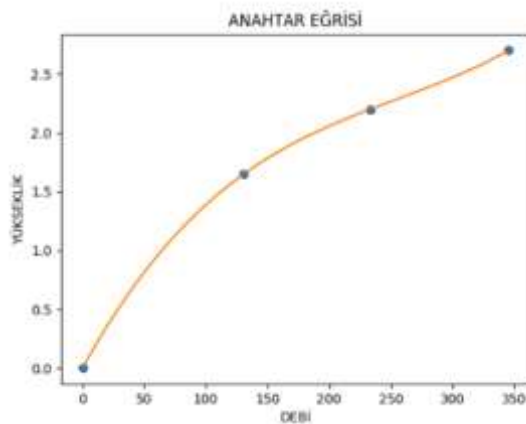
Ek-10’ da bulunan yükseklik ve debi değerleri Ek-11’ deki kod yardımıyla ekrana yansıtılır (Şekil 4.4.).

Tablo 4.2 Akarsuyun debi hesapları ve akım derinlikleri

Akarsuyunun Genel Pürüzlülüğü	
0.029	
(1) Enkesit tabanı oldukça muntazam, sahilleri düz ve sağlam, hiç bitki örtüsü yok	
Akarsu Eğimi	
11.0°0.0001	
Debi Hesabı	
Minimum Debi	140.10m
Ortalama Debi	249.14m
Maksimum Debi	370.38m
Yükseklik Hesabı	
Minumum Yükseklik	1.65m
Ortalama Yükseklik	2.20m
Maksimum Yükseklik	2.70m

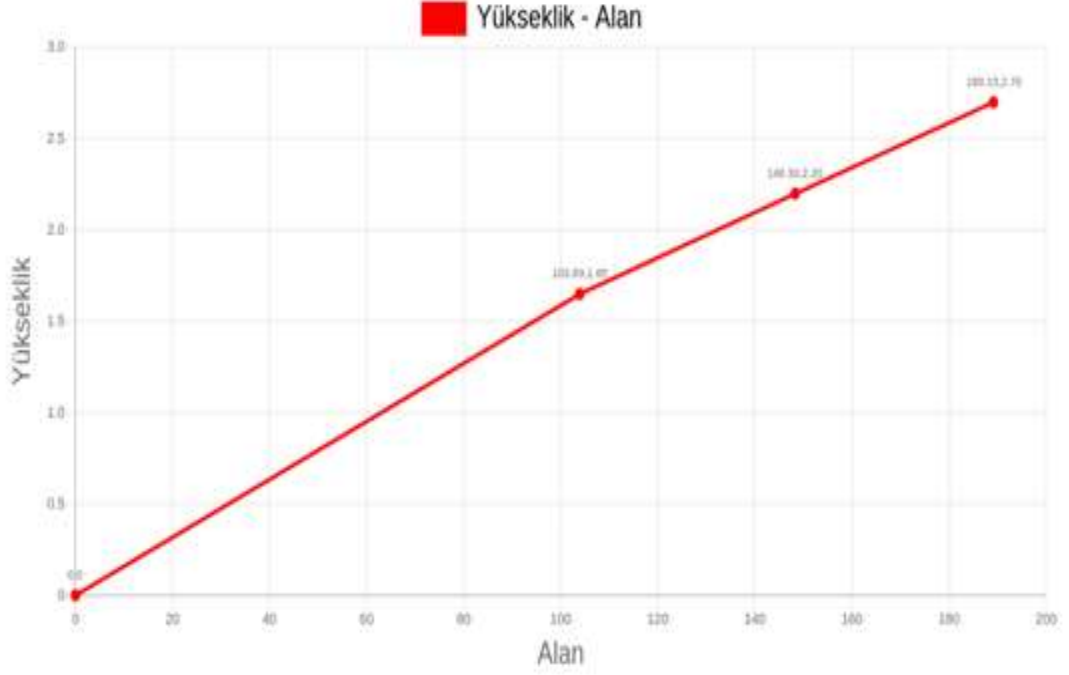
Bulunan yükseklik, minimum, ortalama ve maksimum debi yardımıyla anahtar eğrisi çizilir. Ek-12’ deki kod yardımıyla anahtar eğrisi denklemi bulunur.

Bu kodlarla grafiği oluşturulan anahtar eğrisi Ek-13’ teki kodlar ile ekrana yansıtılır.

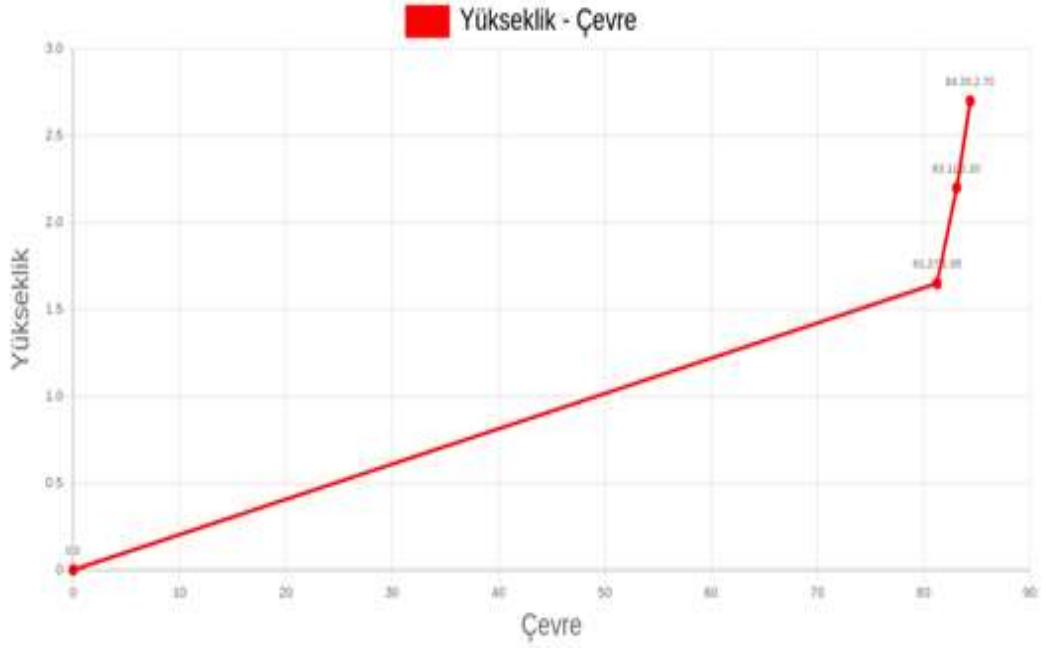


Şekil 4.4. Anahtar eğrisi

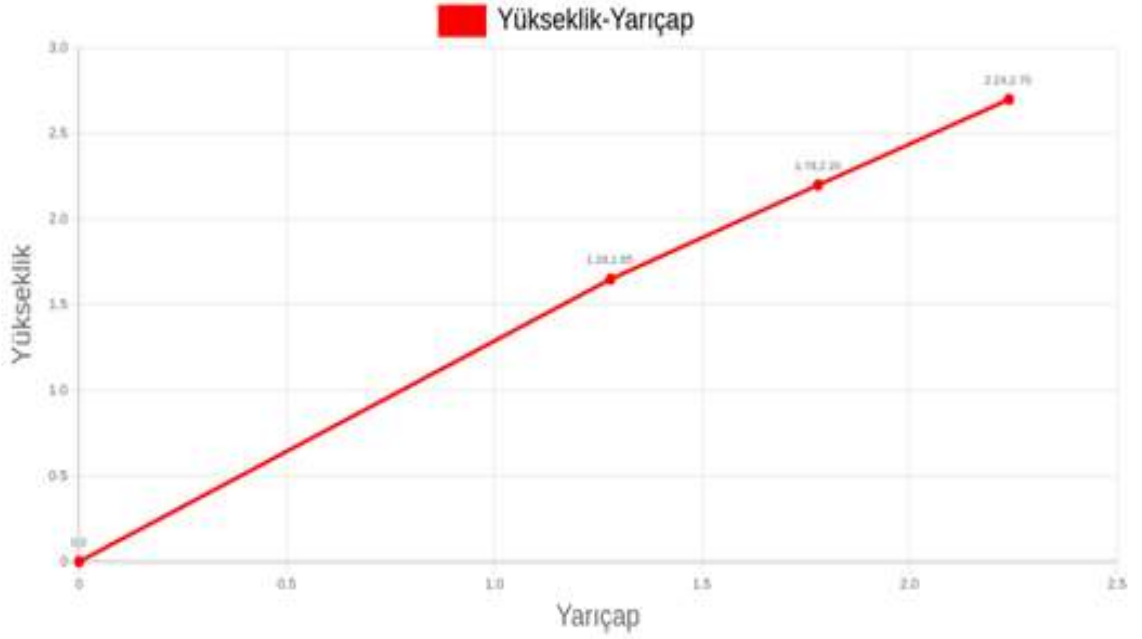
Bulunan yükseklik – alan (Şekil 4.5.), yükseklik- çevre (Şekil 4.6.), yükseklik – yarıçap (Şekil 4.7.) değerleriyle grafik çizilir. Bu grafikleri oluşturmak için Ek-14’ teki kodlar kullanılmıştır.



Şekil 4.5. Islak alan ve akım derinliği arasındaki değişim



Şekil 4.6. Islak çevre ve yükseklik arasındaki değişim



Şekil 4.7. Hidrolik yarıçap ve akım derinliği arasındaki değişim

4.2. Kaplamasız Kanal

a katsayısı kullanıcının bağlama projesi föyünde % olarak tanımlanmış olup kullanıcıya göre değişkendir. $Q_{isale} = Q_{min} \times a$ formülü ile Q_{isale} hesaplanır .

Ek-15' teki kod yardımı ile a katsayısı değerleri kullanıcın seçimine bırakılmıştır .

h katsayısı kullanıcının seçimine bırakılmış olup kullanıcıya göre değişkendir. Ek-16' daki kod yardımı ile h katsayısı değerleri kullanıcın seçimine bırakılmıştır.

Kaplamasız kanal için m, n, eşel sıfır kotu, isale kanalı uzunluğu için tablo oluşturulmuş olup kullanıcının seçimine bırakılmıştır. Ek-17' deki kod yardımıyla bu işlemler sağlanır.

$$A = b.h + m.h^2, \quad \zeta = b + 2h\sqrt{1 + m^2}, \quad R = \frac{A}{\zeta}$$

Kabartma Kotu hesaplanır. Bir tabloya aktarılıp (Tablo 4.3.) *Kabartma Kotu – Eşel sıfır Kotu* $\cong 4 - 6$ değerleri aralığında olması istenir. Tablodaki değerler için yazılan kodlar Ek-18' deki gibidir.

Tablo 4.3. İsale kanalı optimizasyonu (Kaplamasız kanal)

Derinlik h (m)	Taban Genişliği b (m)	Alan A(m ²)	Çevre Ç (m)	Hidrolik Yarıçap R (m)	Taban Eğimi J (-)	Akım Hızı V (m/s)	Kabartma Kotu (m)	K.K-E.S.K (m)
1.60	8	16.90	14.04	1.20	0.00169		964.87	11.87
1.62	8	17.16	14.11	1.22	0.00161		964.18	11.18
1.64	8	17.42	14.19	1.23	0.00154		963.59	10.59
1.66	8	17.69	14.26	1.24	0.00148		963.08	10.08
1.68	8	17.96	14.34	1.25	0.00142		962.57	9.57
1.70	9	19.92	15.42	1.29	0.00111		959.93	6.93
1.72	9	20.21	15.49	1.30	0.00106		959.51	6.51
1.74	9	20.50	15.57	1.32	0.00101		959.09	6.09
1.76	9	20.80	15.64	1.33	0.00098		958.83	5.83
1.78	9	21.09	15.72	1.34	0.00094		958.49	5.49
1.80	9	21.38	15.79	1.35	0.00090		958.15	5.15
1.82	9	21.68	15.87	1.37	0.00086		957.81	4.81
1.84	9	21.98	15.94	1.38	0.00083		957.55	4.55
1.86	9	22.28	16.02	1.39	0.00080		957.30	4.30
1.88	9	22.58	16.09	1.40	0.00077		957.04	4.04
1.90	10	24.78	17.17	1.44	0.00062		955.77	2.77
1.92	10	25.10	17.25	1.46	0.00059		955.51	2.51
1.94	10	25.42	17.32	1.47	0.00057		955.35	2.35
1.96	10	25.75	17.40	1.48	0.00055		955.17	2.17
1.98	10	26.07	17.47	1.49	0.00053		955.00	2.00
2.00	10	26.40	17.55	1.50	0.00052		954.92	1.92

$$A = b \cdot h + m \cdot h^2, \quad \zeta = b + 2h\sqrt{1 + m^2}, \quad V = \frac{1}{n} \cdot R^{2/3} \cdot J^{1/2}, \quad R = \frac{A}{\zeta}$$

formülleri yardımıyla A, Ç, J, R, V ve Kabartma Kotu hesaplanır. Bir tabloyu aktarılıp (Tablo 4.4.) Kabartma kotu-Eşel Sıfır Kotunun 4-6 değerleri aralığında olması istenir. Tablodaki değerler için yazılan kodlar Ek-19' daki gibidir.

Tablo 4.4 Isale kanalı optimizasyonu (Kaplama kanalı)

Derinlik h (m)	Taban Genişliği b (m)	Alan A(m ²)	Çevre Ç (m)	Hidrolik Yarıçap R (m)	Taban Eğimi J (-)	Akım Hızı V (m/s)	Kabartma Kotu (m)	K.K-E.S.K (m)
1.00	5	8.84	8.61	1.03	0.0049	4.4430	992.15	39.15
1.02	5	9.04	8.68	1.04	0.0046	4.3447	989.60	36.60
1.04	5	9.23	8.75	1.05	0.0043	4.2552	987.05	34.05
1.06	5	9.43	8.82	1.07	0.0041	4.1650	985.35	32.35
1.08	5	9.63	8.89	1.08	0.0038	4.0785	982.80	29.80
1.10	6	10.93	9.97	1.10	0.0029	3.5934	975.15	22.15
1.12	6	11.16	10.04	1.11	0.0028	3.5193	974.30	21.30
1.14	6	11.38	10.11	1.13	0.0026	3.4513	972.60	19.60
1.16	6	11.61	10.18	1.14	0.0025	3.3829	971.75	18.75
1.18	6	11.83	10.25	1.15	0.0023	3.3200	970.05	17.05
1.20	6	12.06	10.33	1.17	0.0022	3.2567	969.20	16.20
1.46	7	16.59	12.26	1.35	0.0010	2.3674	959.00	6.00
1.48	7	16.85	12.34	1.37	0.0009	2.3309	958.15	5.15
1.50	8	18.62	13.41	1.39	0.0007	2.1093	956.45	3.45
1.52	8	18.90	13.48	1.40	0.0007	2.0781	956.45	3.45
1.54	8	19.19	13.55	1.42	0.0007	2.0467	956.45	3.45
1.56	8	19.48	13.62	1.43	0.0006	2.0162	955.60	2.60
1.58	8	19.77	13.70	1.44	0.0006	1.9866	955.60	2.60
1.60	8	20.06	13.77	1.46	0.0006	1.9579	955.60	2.60
1.62	8	20.35	13.84	1.47	0.0006	1.9300	955.60	2.60
1.64	8	20.65	13.91	1.48	0.0005	1.9020	954.75	1.75

4.3. Kaplamalı Ya Da Kaplamasız Olma Durumu

$\tau = \gamma \cdot R \cdot J$, $\tau_{kr} = 0.06(\gamma_{su} - \gamma) \cdot d$ formülleriyle τ_{kr} ve τ hesaplanır. $\tau > \tau_{kr}$ ise kaplamalı kanal yapılması şartı aranır. Bu hesap için yazılmış olan kod Ek-20’ gibidir.

Bu işlem sonucunu ekrana yazdırabilmek için Ek-21’ deki kod yazılmıştır.

Tablo 4.5 Kanalin kaplamalı ya da kaplamasız seçilmesi

0.0001	1.00062 < 12.786
	Kaplamalı Seç

$\tau_{kr} < \tau_0$ olduğundan kaplamalı kanal yapılması şarttır (Tablo 4.5.)

4.4. Çökeltim Havuzunun Boyutlandırılması

$V = \alpha \cdot \sqrt{d}$ formülüyle hız hesaplanır. Buradaki alfa değeri için

$$d > 1mm \text{ ise } \alpha = 36$$

$$d \geq 1 \geq 0.1mm \text{ ise } \alpha = 44$$

$d < 0.1 mm$ ise $\alpha = 51$ durumları göz önüne alınarak seçim yapılır. Bağlama projesi föyünde verilen d değerinin istenilen aralıkta oluşması için Ek-22’ deki karar yapısı oluşturulmuştur.

α değerinin seçimi sonucunda hızın bulunabilmesi için yazılan kod Ek-23’ deki gibidir.

-Laminer Akım İçin

$$W = \frac{9}{d} \left(\sqrt{1 + 150d^3} - 1 \right) \text{ sonuç } W = \frac{mm}{s} \rightarrow m/s' \text{ ye çevirilir.}$$

Formülü kullanılarak W hesaplanır. Kod Ek-24’ teki gibidir.

$$\frac{W}{h} = \frac{V}{L} \rightarrow L = \frac{V \cdot h}{W} \text{ formülüyle laminer akım bulunur. Kodu Ek-25’ teki gibidir.}$$

-Türbülanslı Akım

$$W_a = \frac{0.132V}{\sqrt{h}} \rightarrow m/s, \quad L = \frac{V \cdot h}{W - W_a} \text{ ve } L_{em} = (1.5 - 2)L \text{ Formülleri yardımı ile } W_a, L \text{ ve } L_{em} \text{ hesaplanır. Kodu Ek-26’ daki gibidir.}$$

$1.1Q_{is} = V \cdot h \cdot B$ formülü yardımıyla B hesaplanır (Tablo 4.6.). Yazılan kod Ek-27’ deki gibidir.

Yukarıdaki tüm formüllerin ekrana yazdırabilmek için yazılmış olan kodlar ve ekran alıntısı görüntüsü Ek-28’ deki gibidir.

Tablo 4.6. Çökeltim havuzu boyutları sonuçları

Çökeltim Havuzunun Boyutlandırılması

V_ç = 44cm/s
V_ç = 0.44m/s
Laminer Akım
W= 101.59mm/s
W= 0.10159m/s
h değeri

1.5

Lj= 12.99m
Türbülanslı Akım
W_a =0.034m/s
L_t =19.39m
L_e =38.79m
B_{çh}=32.67m
b = 4 ile 6 arasında olacak

Çökeltim Havuzu Göz Açıklığı Tablosu

Çökeltim havuzunun göz açıklığı sayısının bulabilmek için göz açıklığı sayısı $b=B/n$ ile bulunur (Tablo 4.7.). Burada b nin 4-6 arası olması şartı aranır. Bunun için hazırlanmış olan tablo kodu Ek-29’ daki gibidir.

Tabloyu ekrana yazdırma kodu Ek-30’ daki gibidir.

Tablo 4.7. Çökeltim havuzu göz açıklığı

Çökeltim Havuzu Göz Açıklığı Tablosu

n	b
1	32.67
2	16.33
3	10.89
4	8.17
5	6.53
6	5.44
7	4.67
8	4.08
9	3.63
10	3.27

Seçilen göz sayısı = 6 ; b= 5.44m

4.5. Çökeltim Havuzunun Sonundaki Eşiğin Boyutlandırılması

$Q_{is} = \frac{2}{3} \cdot \mu \cdot B \cdot \sqrt{2g} \cdot h_e^{\frac{3}{2}}$, $P_e = h - h_e$ Formülleri yardımıyla h_e ve P_e hesaplanır (Tablo 4.8.). Bu hesap için yazılan kod Ek-31' deki gibidir.

h_e ve P_e değerlerinin ekrana yansıtılabilmesi için ekrana yazılan kod Ek-32' deki gibidir.

Tablo 4.8. P_e ve h_e

Çökeltim Havuzu Sonundaki Eşiğin Boyutlandırılması
he=2.196 m
Pe=0.804 m

4.6. Bağlama Genişliği ve Yüksekliğinin Tayini

$B_{net} = \frac{Q_{max}}{5}$ Formülüyle B_{net} bulunur. Bunun için yazılan kod aşağıdaki gibidir.

this.bnet = this.maksimumdebi / 5;

Bulunan B_{net} 'in $\frac{1}{3} \cdot B \leq B_{net} < B$ aralığında olması gerekmektedir.

$Q_{min} - 1.1Q_{is} = \frac{2}{3} \cdot \mu \cdot B_{net} \cdot h_{min}^{\frac{3}{2}} \cdot \sqrt{2g}$ formülüyle h_{min} Ek-33' deki kod yardımıyla bulunur.

$KretKotu = KabartmaKotu - h_{min}$, $P = KretKotu - Taban Kotu$ formülleriye Kret kotu ve P hesaplanır. Bunun için yazılmış olan kodlar Ek-34' deki gibidir.

$Q_{ort} - 1.1Q_{is} = \frac{2}{3} \cdot \mu \cdot B_{net} \cdot h_{ort}^{\frac{3}{2}} \cdot \sqrt{2g}$, $Ort. Kabartma Kotu = KretKotu + h_{ort}$ formülleri aracılığıyla h_{ort} ve $Ort. Kabartma Kotu$ Ek-35' deki kodlar yardımıyla bulunur.

$Q_{max} - 1.1Q_{is} = \frac{2}{3} \cdot \mu \cdot B_{net} \cdot h_{max}^{\frac{3}{2}} \cdot \sqrt{2g}$, $Max. Kabartma Kotu = KretKotu + h_{max}$ formülleri aracılığıyla h_{max} ve $Max. Kabartma Kotu$ Ek-36' daki kodlar yardımıyla bulunur.

$Bağ. Ayak Yük. = max. kabartmakotu + 0.5$ Formülü yardımıyla bağlama ayak yüksekliği tayini Ek-37' deki kod ile bulunur.

$H_{0min} = P + h_{min}$, $H_{0ort} = P + h_{ort}$ ve $H_{0max} = P + h_{max}$ formülleri yardımıyla minimum su yüksekliği (H_{0min}), ortalama su yüksekliği (H_{0ort}) ve maksimum su yüksekliği (H_{0max}) bulunur (Tablo 4.9.). Bunun için yazılan kodlar Ek-38’deki gibidir.

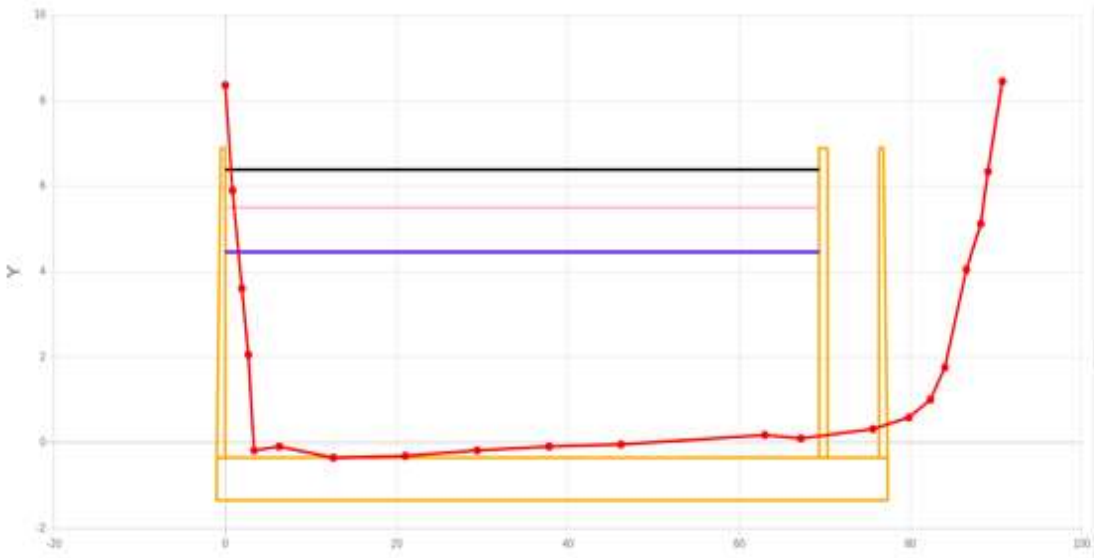
Yukarıda yapılan tüm bu işlemlerin ekrana yansıtılabilmesi için yazılmış olan kodlar Ek-39’deki gibidir.

Tablo 4.9. Sonuçların ekrana yazdırılması

Bnet=69.318 m
hmin=0.689 m
Kret Kotu=957.461 m
Taban Kotu
953
P=4.811 m
hort=1.154 m
Ort Kabartma Kotu=958.615 m
hmax=1.575 m
Mak Kabartma Kotu=959.036 m
Bağ ayak Yük=959.536 m
Min Su Yüksekliği(Homin)=5.500 m
Ort Su Yüksekliği(Hoort)=5.965 m
Mak Su Yüksekliği(Homak)=6.386 m

Bağlama yüksekliği ve genişliğinin (Şekil 4.8.) grafik üzerine aktarılabilmesi için Ek-40’teki kod yazılmıştır.

Grafığın ekranda gösterilebilmesi için Ek-41’deki kod yazılmıştır.



Şekil 4.8. Akarsu enkesi üzerine bağlama ve çakıl geçidinin yerleşimi

4.7. Su Alma Ağzının Boyutlandırılması

$A = \frac{1.1Q_{is}}{V}$, $A = a \cdot b$ formülleri yardımıyla A ve b bulunur. $A = \frac{1.1Q_{is}}{V}$ formüldeki hız $V = 0.8 - 1 \text{ m/s}$ aralığından seçilir. $V = 1 \text{ m/s}$ olarak alınmıştır (Tablo 4.10.). Bu seçimin yapılabilmesi için hız tablosu Ek-42’ deki kodlarla oluşturulmuştur.

İşlemlerin sonucu Ek-43’ teki kod ile bulunmuştur.

Bu işlemlerin sonucunun ekranda gösterilebilmesi için yazılmış olan kod ve ekran görüntüsü Ek-44’ teki gibidir.

Tablo 4.10 Su alma ağzının boyutlandırılması



Su Alma Ağzının Boyutlandırılması	
a=4.400 m	
(V = 0.8 - 1.2 m/s 1 arasında seçile)	
A=43.120 m²	
b=9.800 m	

4.8. Yıkama Kanalının Byutlandırılması

$V_y = 2 - 2,5 \frac{m}{s}$ V_y ’ nin bu aralıkta tanımlanabilmesi için yazılan kod Ek-45’ deki gibidir.

$V_y = 2 \text{ m/s}$ seçilmiştir .

$Q_y = 0.1 \cdot Q_{is}$, $A_y = \frac{Q_y}{V_y}$, $(Dairesel Kesit) A_y = \pi r^2$ formülleri aracılığıyla Ek-46’

da yazılan kodlar ile Q_y , A_y ve r bulunmuştur (Tablo 4.11.).

İşlemlerin ekrana yazdırılabilmesi için yazılan kodlar ve ekran görüntüsü Ek-47’ deki gibidir.

Tablo 4.11. Yıkama kanalının boyutlandırılması

Yıkama Kanalının Boyutlandırılması
$(V_y = 2 - 2.5 \text{ m/s}$ 2 arasında seçilir)
$Q_y = 3.920 \text{ m}^3/\text{s}$
$A_y = 1.960 \text{ m}^2$
$r = 0.790 \text{ m}$
Yıkama Kanalının Eğimi
$R_y = 0.395 \text{ m}$

4.9. Yıkama Kanalının Eğimi

$R_y = \frac{\text{Çap}}{4} = \frac{\text{Yarıçap}}{2} = \frac{r}{2} \rightarrow \text{Dairesel Kesit}$ formülü aracılığıyla R_y Ek-48’deki kod ile bulunur.

$V_y = k \cdot R_y^{\frac{2}{3}} \cdot J_y^{\frac{1}{2}}$ formülüyle J_y hesaplanır. Ancak buradaki k katsayısı için seçim tablosu oluşturulmuştur (Tablo 4.12.). Oluşturulan bu tablonun ekranda gösterilmesi için yazılan kod Ek-49’deki gibidir.

V_y ’nin hesaplanabilmesi için yazılan kod Ek-50’deki gibidir.

Tüm bu işlemlerin ekrana yazdırılabilmesi için yazılan kodlar ve ekran görüntüsü Ek-51’deki gibidir.

Tablo 4.12. Yıkama kanalı eğimi

Yıkama Kanalının Eğimi
$R_y = 0.395 \text{ m}$
k katsayısı
$J_y = 0.00170 \text{ m}$

4.10. Çakıl Geçidi Debisinin Tayini

b_ϕ ve μ değerlerinin belirli aralıklarda tanımlanabilmesi için hazırlanan tablo Ek-52’deki kodlarla oluşturulmuştur.

$\mu = 0.90$, $b_{\zeta} = 6$ seçilmiştir.

$Q_{\zeta} = \frac{2}{3} \cdot \mu \cdot b_{\zeta} \cdot (H_{0_{max}} - h_{3_{max}})^{\frac{2}{3}} \cdot \sqrt{2g} + \mu \cdot h_{3_{max}} \cdot b_{\zeta} \cdot \sqrt{2g(H_{0_{max}} - h_{3_{max}})}$ formülü yardımıyla Q_{ζ} Ek-53' deki kod ile bulunmuştur (Tablo 4.13.).

Bütün bu işlemlerin ekrana yazdırılabilmesi için yazılan kodlar ve ekran görüntüsü Ek-54' teki gibidir.

Tablo 4.13. b_{ζ} , Q_{ζ} ve μ değerleri

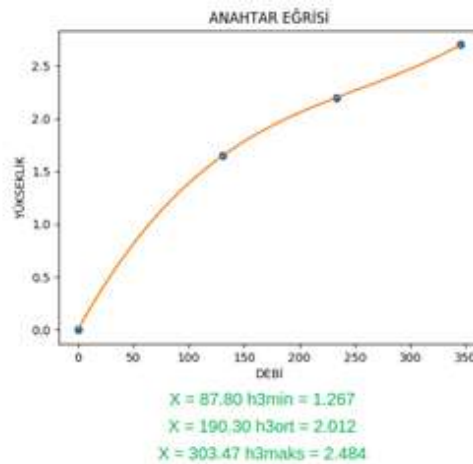
$(b_{\zeta} = 3 - 6 \text{ m/s}$ 6 arasında seçilir)
$(\mu = 0.90 - 0.95 \text{ m/s}$ 0.9 arasında seçilir)
$Q_{\zeta} = 240.04 \text{ m}^3/\text{s}$

-Anahtar eğrisi yardımıyla h_{3min} , h_{3ort} ve h_{3maks} değerlerini bulunması:

Anahtar eğrisinin noktalarının kordinatları işaretlendikten sonra curve fitting algoritması kullanılarak en uygun 3. dereceden denklem bulunup eğri çizildi. Denklem $7.748e-08x-5.923e-05x^3+0.01903x^2+4.441e-16$ olarak bulundu.

Grafik yardımıyla her bir x kordinattan $1.1 \cdot q_{isale}$ geri gelinerek x koordinatı bulunur. Bulunan x koordinatı denklemde yerine yazılarak h_{3min} , h_{3ort} ve h_{3max} değerleri bulunur (Şekil 4.9.). Tüm bu işlemleri grafikte gösterebilmek için yazılan kod Ek-55' teki gibidir.

h_{3min} , h_{3ort} ve h_{3max} değerleri Ek-56' daki kod ile ekrana yazdırılmıştır.



Şekil 4.9. Anahtar eğrisi

4.11. Hidrolik Profilin Tayini

$x' = x \cdot h_{max}$ $y_1' = h_{max} \cdot y_1 \rightarrow AltNap$ $y_2' = h_{max} \cdot y_2 \rightarrow ÜstNap$ formülleriyle x' y_1' ve y_2' hesaplanır ardından tabloya aktarılır (Tablo 4.14.). Hesap ve tablo için yazılan kod Ek-57' deki gibidir.

Tablonun ekrana yazdırma kodu Ek-58' deki gibidir.

Tablo 4.14. Hidrolik profil hesabı

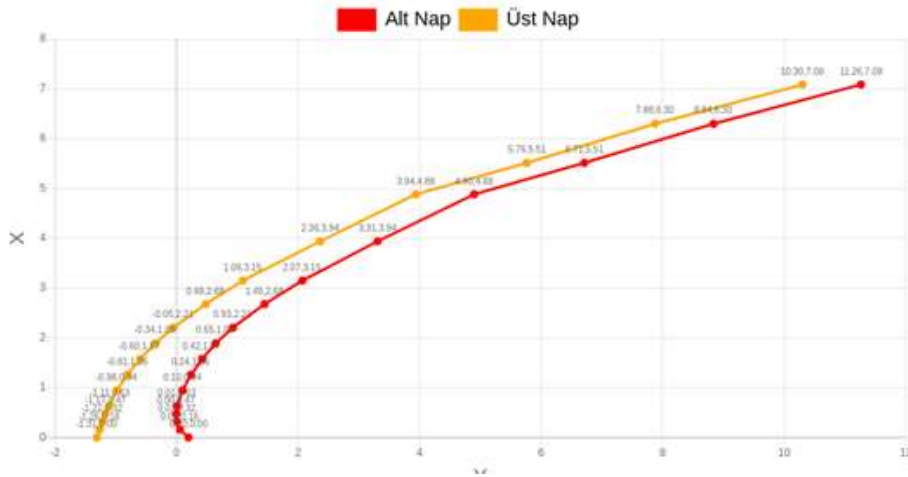
x	y1	y2	x'	y1'	y2'
0.0	0.126	-0.831	0.000	0.198	-1.309
0.1	0.036	-0.803	0.158	0.057	-1.265
0.2	0.007	-0.772	0.315	0.011	-1.216
0.3	0.000	-0.74	0.473	0.000	-1.166
0.4	0.007	-0.702	0.630	0.011	-1.106
0.6	0.063	-0.62	0.945	0.099	-0.977
0.8	0.153	-0.511	1.260	0.241	-0.805
1.0	0.267	-0.38	1.575	0.421	-0.599
1.2	0.410	-0.219	1.890	0.646	-0.345
1.4	0.590	-0.03	2.205	0.929	-0.047
1.7	0.920	0.305	2.678	1.449	0.480

2.0	1.315	0.693	3.150	2.071	1.092
2.5	2.100	1.500	3.938	3.308	2.363
3.1	3.110	2.500	4.883	4.898	3.938
3.5	4.260	3.660	5.513	6.710	5.765
4.00	5.610	5.000	6.300	8.836	7.875
4.5	7.150	6.540	7.088	11.262	10.301

4.12. Bağlamanın Statik Profilinin Tayini

Hidrolik profil tayini tablosunu oluşturmak için yazılan kod Ek-59' dadır (Şekil 4.10.).

Ekrana yazdırabilmek için yazılan kod Ek-60' taki gibidir.



Şekil 4.10. Hidrolik profil alt ve üst nap

$c_2 = 1 - (h')^2 \cdot (3 - 2h')$, $c_1 = 1 - (h')^2 \cdot (2 - h')$, $h' = \frac{h_{max}}{H_{o_{max}}}$, $\gamma' = \frac{\gamma_b}{\gamma}$ ve $\tan \alpha = \sqrt{\frac{c_2}{c_1 \cdot \gamma' - m}}$ formülleriyle c_2 , c_1 , h' , γ' ve $\tan \alpha$ bulunur (Tablo 4.15.). Bunlar için yazılan kodlar Ek-61' deki gibidir.

Bu işlemlerin ekrana yazdırılma kodu Ek-62' deki gibidir.

Tablo 4.15. h' , γ' , c_2 , c_1 , $\tan \alpha$ ve α değerleri

$h' = 0.25$
 $\gamma' = 2.40$
 $c_2 = 0.85$
 $c_1 = 0.89$
 $\tan \alpha = 0.70$
 $\alpha = 34.88$

4.13. Devrilme Tahkiki

Mevcut Kuvvetler:

$$G = \frac{1}{2} \cdot \gamma_b \cdot H_0^2 \cdot \tan \alpha, \quad G_0 = \frac{1}{2} \cdot \gamma_b \cdot h_{max}^2 \cdot \tan \alpha, \quad x = \frac{1}{2} \cdot \gamma \cdot H_0^2, \quad u = \frac{1}{2} \cdot m \cdot \gamma \cdot H_0^2 \cdot \tan \alpha$$

$$x_0 = \frac{1}{2} \cdot \gamma \cdot h_{max}^2$$

B noktasına olan moment mesafeleri:

$\frac{2}{3} \cdot H_0 \cdot \tan \alpha$, $H_0 \cdot \tan \alpha - \frac{1}{3} \cdot h_{max} \cdot \tan \alpha$, $\frac{1}{3} H_0$, $H_0 - \frac{2}{3} \cdot h_{max}$ ve $\frac{2}{3} \cdot H_0 \cdot \tan \alpha$ formülleri yardımıyla mevcut kuvvetler ve B noktasına olan moment mesafeleri Ek-63' teki kodlar yardımıyla bulunur (Tablo 4.16.).

Bütün bu formülleri ekrandaki tabloda gösterilebilmesi için aşağıdaki kodlar yazılmıştır.

Ekran görüntüsü Ek-64' teki gibidir.

B Noktasına Olan Moment Mesafeleri Ek-65' teki kodlarla bulunur.

Tablo 4.16. Devrilme tahkiki

Devrilme Tahkiki	
B Noktasına Olan Moment Mesafeleri	Mevcut Kuvvetler
G= 334.04	2.97m
G ₀ = 20.32	4.09m
x= 199.63	2.13m
x ₀ = 12.14	5.34m
u = 55.66	2.97m

$M_{\bar{o}} = G \cdot \frac{2}{3} \cdot H_0 \cdot \tan \alpha + x_0 \cdot (H_0 - \frac{2}{3} h_{max})$, $M_d = G_0 \cdot (H_0 \cdot \tan \alpha - \frac{1}{3} \cdot h_{max} \cdot \tan \alpha) + x \cdot \frac{1}{3} \cdot H_0 + u \cdot \frac{2}{3} H_0 \cdot \tan \alpha$ ve $n_d = \frac{M_{\bar{o}}}{M_d} > 1.5$ formülleri yardımıyla $M_{\bar{o}}$, M_d ve n_d bulunur (Tablo 4.17.). Bunun için yazılmış olan kodlar Ek-66' daki gibidir.

Ekranada yazdırılabilmesi için kullanılan kodlar Ek-67' deki gibidir.

Tablo 4.17. $M_{\bar{o}}$, M_d ve n_d değerleri

$M_{\bar{o}}=1056.189 \text{ m}$
 $M_d=673.175 \text{ m}$
 $n_d=1.569 > 1.500$ ilave temele gerek yoktur.

4.14. Kayma Tahkiki

$f \cdot (\text{toplam düşey kuvvetler}) > (\text{toplam yatay kuvvetler})$, $f \cdot (G - G_0 - u) > x - x_0$ formülleri yardımıyla istenilen şartın sağlanıp sağlanmadığını bakılır. Burada bulunan f katsayısı bağlama yüzeyiyle temel arasındaki sürtünme katsayısı olup seçim yapılabilmesi için bir tablo oluşturulmuştur bu tablo için Ek-68' deki kodlar yazılmıştır (Tablo 4.18.).

Kayma tahkikinin kontrolü için yapılan işlemin kodu Ek-69' daki gibidir.

Bu işlemin ekrana yazdırılma kodu aşağıdaki gibi olup ekran görüntüsü Ek-70' tedir.

Tablo 4.18. Kayma tahkiki

Kayma Tahkiki
$N(G-G_0-u) > x \cdot x_0$
193.544 > 187.48

4.15. Taban Gerilmelerinin Tayini

$$G_{net} = N = G - G_0, M = M_0 - M_d, M = \left(\frac{b}{2} - e\right) \cdot N \rightarrow e \text{ bulunur}, \sigma = \frac{N}{A} \pm \frac{M_0}{w},$$

$$A = b \cdot 1$$

$w = \frac{1 \cdot b^2}{6}$, $M_0 = N \cdot e$ formülleri aracılığıyla G_{net} , M , e , M_0 , A , w ve σ değerleri Ek-71' deki kodlar yardımıyla bulunmuştur (Tablo 4.19.).

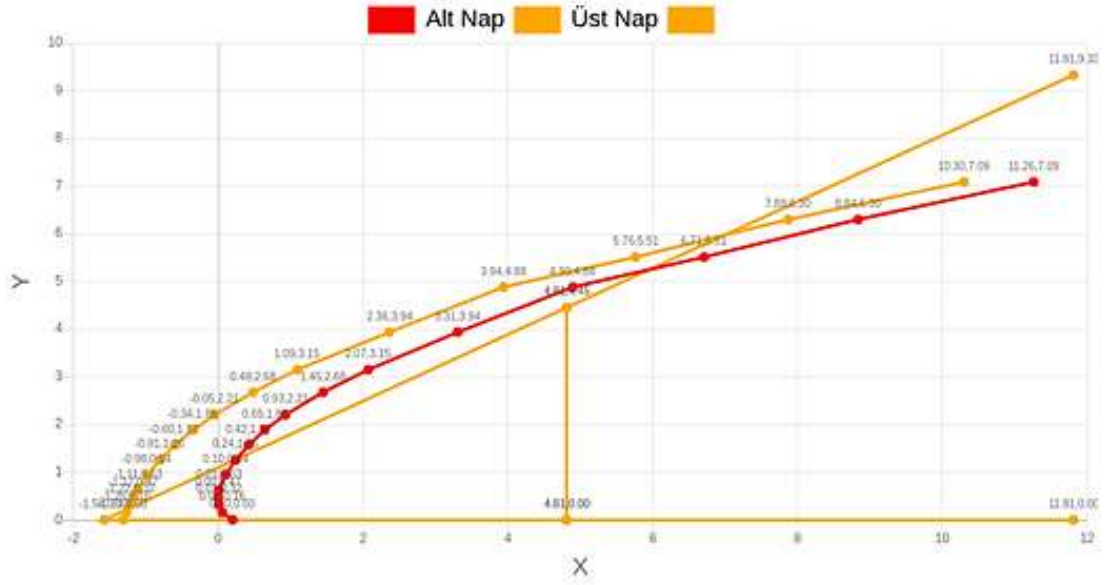
Bu işlemlerin sonucunda sonuçların ekranda gösterilebilmesi için Ek-72' deki kod yazılmıştır.

Tablo 4.19. Taban gerilmeleri tayini

Taban Gerilmelerinin Tayini
$G_{net} = N = 258.059$
$M = 383.014$
$b = 4.452$
$e = 0.742$
$M_0 = 191.394$
$A = 4.452$
$W = 3.303$
$\sigma_{max} = 115.913 \text{ t/m}^2 = 1137.11 \text{ kN/m}^2$
$\sigma_{min} = 0.023 \text{ t/m}^2 = 0.22 \text{ kN/m}^2$

Şeklin grafik üzerinde gösterilebilmesi için yazılan kodlar Ek-73' teki gibidir (Şekil 4.11.).

Ekrana yazdırabilmek için Ek-74' teki kod kullanılır.



Şekil 4.11. Tatbiki profilin belirlenmesi

4.16. Bağlama Düşüm Yatağının Tayini

$$q_{min} = \frac{Q_{min} - 1,1q_{isale}}{B_{net}}, \quad q_{ort} = \frac{Q_{ort} - 1,1q_{isale}}{B_{net}}, \quad q_{max} = \frac{Q_{max} - 1,1q_{isale}}{B_{net}},$$

$$h_{min} + P = H_{0min}, \quad H_{0min} = h_{1min} + 1,1 \cdot \frac{q_{min}^2}{2g \cdot h_{1min}^2},$$

$$h_{ort} + P = H_{0ort}, \quad H_{0ort} = h_{1ort} + 1,1 \cdot \frac{q_{ort}^2}{2g \cdot h_{1ort}^2},$$

$$h_{max} + P = H_{0max}, \quad H_{0max} = h_{1max} + 1,1 \cdot \frac{q_{max}^2}{2g \cdot h_{1max}^2}$$

$$h_{2min} = \frac{-h_{1min}}{2} + \sqrt{\frac{h_{1min}^2}{4} + \frac{2 \cdot q_{min}^2}{g \cdot h_{min}}},$$

$$h_{2ort} = \frac{-h_{1ort}}{2} + \sqrt{\frac{h_{1ort}^2}{4} + \frac{2 \cdot q_{ort}^2}{g \cdot h_{ort}}},$$

$$h_{2max} = \frac{-h_{1max}}{2} + \sqrt{\frac{h_{1max}^2}{4} + \frac{2 \cdot q_{max}^2}{g \cdot h_{max}}}$$

formülleri ile q_{min} , q_{ort} , q_{max} , H_{0min} , H_{0ort} , H_{0max} , h_{1min} , h_{1ort} , h_{1max} , h_{2min} , h_{2ort} , h_{2max} değerleri yukarıdaki formüller yardımıyla bulunur. Bu formüller Ek-75' teki kod ile çözülmüştür.

Ek-76' daki kodlar ile işlemler hesaplanmıştır.

1. Durum

$$h_2 > h_3 \text{ ise}$$

a) Düşüm Havuzu

$$t = \frac{h_2 - h_3}{1,4}$$

$$L = 5 \cdot h_2$$

$$F_r = \frac{V_1}{\sqrt{g \cdot h_1}} \leq 17$$

c) 2. eşikli düşüm yatağı

$$F_r \leq 12$$

$$P_e = h_2 - h_e$$

$$Q_{max} - 1.1Q_{is} = \frac{2}{3} \cdot \mu \cdot b \cdot h_e^{\frac{3}{2}} \cdot \sqrt{2g} \rightarrow h_e \text{ bulunur}$$

$P_e > h_3 \rightarrow$ ise çözüm doğrudur

$P_e < h_3 \rightarrow$ ise çözüm yanlıştır. Batmış akım formülü kullanılmalıdır.

$$q = \frac{2}{3} \mu \cdot b \cdot (h_2 - h_3)^{\frac{3}{2}} \cdot \sqrt{2g} + \mu \cdot a \cdot \sqrt{2g \cdot (h_2 - h_3)} \rightarrow a \text{ bulunur}$$

$$P_e = h_3 - a$$

d) kavisli düşüm yatağı

$$F_r = 17$$

2. Durum

$$h_3 > h_2$$

Meyilli düşüm yatağı seçilir.

3. Durum

Küçük debilerde $h_3 > h_2$

Büyük debilerde $h_2 > h_3$

1. Durum düşüm yataklarından biri seçilir.

4. Durum

Küçük debilerde $h_2 > h_3$

Büyük debilerde $h_3 > h_2$

2. Durum düşüm yataklarından biri seçilir.

Bu durumlar dikkate alınarak kıyas yapılır yapılan bu kıyas Ek-77' deki kodla gösterilmiştir (Tablo 4.20.).

Tablo 4.20. Bağlama düşüm yatağı tayini

Bağlama Düşüm Yatağının Tayini

```
qmin = 1.267 m3/s
qort = 2.745 m3/s
qmak = 4.378 m3/s
h1min = 0.12942
h1ort = 0.27245
h1mak = 0.42456
h2min = 1.52637
h2ort = 2.24246
h2mak = 2.82889
h3maks = 2.484 < h2maks = 2.82889
h3ort = 2.012 < h2ort = 2.24246
h3min = 1.267 < h2min = 1.52637
h2 > h3 birinci durumun b seçeneği seçilir
11.316 = 4.h2maks
```

4.17. Düz Düşüm Yataklarının Kalınlık Tayini

$d_0 = 0,15 \cdot V_1 \cdot \sqrt{h_1}$, $v_{1max} = \frac{q_{max}}{h_{1max}}$ formülleriyle d_0 ve v_{1max} Ek-78' deki kodlar yardımıyla bulunur.

Analitik yol ile

1) Laneye göre

c katsayısı seçiminde oluşturulan tablo için yazılan kod Ek-79' daki gibidir.

c= 6 seçilir.

$P = H_{0max} - h_{max}$, $\tan \alpha = 1/c$ ve $L_g = c \cdot \Delta H$ formülleriyle P tan α ve L_g Ek-80' deki kodlar yardımıyla hesaplanır.

P_3 ve P_4 değerleri Thales Teoremi ile hesaplanıp Ek-81’deki koda aktarılmıştır.
d nin hesabı için yazılan kod Ek-82’dedir.

Lane’ye göre

$$L_m = L_{düşey} + \frac{1}{3}L_{yatay}$$

$L_m = L_{1-2} + \frac{1}{3}L_{2-3} + \frac{1}{3}L_{3-4} + L_{4-5}$ formülleriyle L_m Ek-83’teki kod ile hesaplanır.

$L_g > L_m$ olduğundan palplanş perde çakılması gerekir (Tablo 4.21)

Tüm bu işlemlerin ekrana yazdırılabilmesi için Ek-84’teki kodlar yazılmıştır.

Tablo 4.21. Düz düşüm yataklarının kalınlık tayini



4.18. Palplanş Perde Çakılması

$c. \Delta H = L_{1-2} + x + x + \frac{1}{3}(L_{2-3} + L_{3-4}) + L_{4-5}$, formülü yardımıyla x mesafesi Ek-85’teki kod ile bulunmuştur.

L_m Ek-86’daki kod ile bulunmuştur.

$L_m \geq L_g$ şartı sağlandı P_3 ve P_4 değerleri Ek-87’deki kod yardımıyla bulunur .

d değeri Ek-88’deki kod yardımıyla bulunur (Tablo 4.22.).

Tüm bu işlemlerin ekrana yazdırılması Ek-89’daki kod yardımıyla sağlanmıştır.

Tablo 4.22. Palplanş Perde Hesabı

Palplanş Perde Çakılması

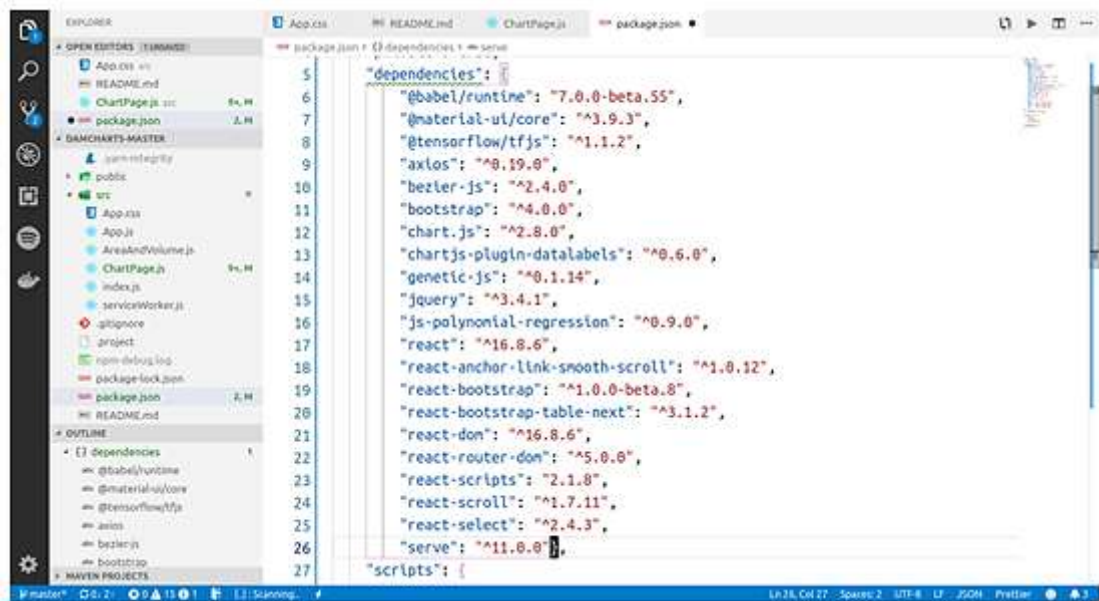
$$\begin{aligned}x &= 10.890 \text{ m} \\L_m &= 28.866 \text{ m} \\L_m &= 28.866 \geq L_g = 28.866 \\P_3 &= 0.954 \text{ t/m}^2 = 9.36 \text{ kN/m}^2 \\P_4 &= 0.168 \text{ t/m}^2 = 1.65 \text{ kN/m}^2 \\d &= 0.401 \text{ t/m}^2 = 3.93 \text{ kN/m}^2 \\d_0 &= 1.008 > d = 0.401\end{aligned}$$

-Geliştirilen Yazılımın Çalıştırılması

```
Terminal: Local +
(venv) stvie@hendrix:~/PycharmProjects/untitled1$ python myrestservice.py
* Serving Flask app "myrestservice" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 780-784-381
```

Python uygulamanın bulunduğu konuma gidilir ve komut satırı açılır. python myrestservice.py komutu verilir. İleri seviye matematiksel işlemler yaptığımız servisimiz artık hazır durumdadır.

Daha sonra web uygulamamızı ayağa kaldırmamız gerekmektedir. Bunun için ilk önce bağımlılıkların (uygulama dizininde package.json dosyası) kurulması gerekmektedir. Bunun için ise uygulamanın bulduğu dizine gidilir npm install komutu verilir.



```

damcharts-mast +
root@hendrix:/home/stevie/damcharts-master# npm install
npm WARN rollup-plugin-visualizer@1.1.1 requires a peer of rollup@>=0.60.0
but none is installed. You must install peer dependencies yourself.
npm WARN tfjs@1.0.1 requires a peer of @tensorflow/tfjs@^0.11.x but none i
s installed. You must install peer dependencies yourself.
npm WARN ts-pnp@1.0.1 requires a peer of typescript@* but none is installed
. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_module
s/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsev
ents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","ar
ch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_module
s/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsev
ents@1.2.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","ar
ch":"x64"})

audited 42431 packages in 15.026s
found 77 vulnerabilities (63 low, 9 moderate, 5 high)
  run 'npm audit fix' to fix them, or 'npm audit' for details
root@hendrix:/home/stevie/damcharts-master# npm start

```

Daha sonra npm start komutu verilerek uygulama açılır. Uygulamanın kurulumu için bütün gereklilikler bu aşamadan sonra tamamlanmış bulunmaktadır. React js için default port 3000 dir. Localhost:3000 konumuna giderek uygulamayı kullanmaya başlayabilirsiniz.



SONUÇLAR VE ÖNERİLER

Bağlamalar su mühendisliğinde önemli hidrolik yapılardandır. Bu çalışma kapsamında sabit bağlamaların hidrolik tasarımı için eğitim amaçlı bir program geliştirilmiştir.

- Hidrolik yapıların planlama tasarımında geliştirilen programlardan yararlanmak proje hazırlanma sürecini kısaltmaktadır.
- Eğitim amaçlı program geliştirilmesi ve bunların ilgili kullanıcıların hizmetine sunulması eğitim kalitesini arttırmaktadır.
- Geliştirilen programlarda ve yazılımlarda kullanıcıya görsellerin sunulması öğrenmeyi ve öğrenilenin kalıcı olmasını sağlamaktadır.
- Visual Studio Code metin editörü ve ilave teknolojilerden yararlanılarak hidrolik yapı tasarımının mümkün olduğu gösterilmiştir.
- Geliştirilen programın kullanıcılardan alınacak geri besleme ile güncellenmesi kolaylıkla yapılabilir şekildedir.
- Planlamaya kattığı en büyük faydalardan biri de kullanıcıya tasarım ile ilgili detaylı açıklamaların program tarafından veriliyor olmasıdır.
- Geliştirilen programda akarsu enkesiti ve çevre özellikleri, debi değerleri gibi planlama parametrelerinde kullanıcıya seçenekler sunulmaktadır. Geliştirilen program girilen ve seçilen parametrelere göre ilgili hesapları yapmakta ve kullanıcıya sunmaktadır.
- Geliştirilen programın lisans, yüksek lisans ve doktora öğrencilerin eğitiminde ve uygulamadaki genç mühendislerin kendilerini bu konuda geliştirmelerinde yararlı olacağı düşünülmektedir.
- Geliştirilen parogram sabit bağlamalar için yapılmıştır. Aynı şekilde hareketli bağlamalar, sifon bağlamalar, karma bağlamalar ve lastik bağlamalar için de programlar geliştirilebilir.

KAYNAKLAR

- Alimoğlu, E., Bozkuş, Z., Yanmaz, A. M.** (2014). Nehir tipi hidroelektrik tesislerin bilgisayar destekli tasarımı, İMO Teknik Dergi, 6925-6941, yazı: 426.
- Baban, R.,** 1995, Design of Diversion Weirs, Wiley, United Kingdom
- Berkün, M.,** 2005, Su Kaynakları Mühendisliği, Birsen Yayınevi, 1. Baskı, İstanbul
- Chau, K.W, Zhang, X.N.** (1995). An MS Excel tool for water distribution network design in environmental engineering education, Advances in Engineering Software, Volume 22, Issue 3, 1995, Pages 139-146.
- Cihangir, E.** 1993, Computer Aided Hydraulic Design of Diversion Weirs with Sidewise Intakes, Middle East Technical University, A Master's Thesis, Ankara
- Brkić, D.** (2017). Solution of the Implicit Colebrook Equation for Flow Friction Using Excel. Spreadsheets
- DSİ, 1988, Regülatör Projesi Kriterleri, Devlet Su İşleri Matbaası, Ankara
- DSİ, Su Yapıları Cilt II, Devlet Su İşleri Matbaası, Ankara
- Erdem, M.,** 2006, Küçük Hidroelektrik Santrallerin Tasarım Ölçütleri, Yüksek Lisans Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Denizli
- Erkek, C. ve Ağırlioğlu, N.,** 2013, Su Kaynakları Mühendisliği, Beta Yayınevi, 6. Baskı.
- Herrera, V., Siy, E. ve Magallanes, C.,** 2013, Design of Diversion Headwork In Tunasan River, Muntinlupa City, In Partial Fulfillment of the Requirements For the Degree of Bachelor of Science in Civil Engineering, Mapua Institute of Technology, Manila City
- Özbek, T.** (2009). Açık Kanal Akımlarının Hidroliği ve Hidrolik yapılar, teknik Yayınevi, Ankara.
- Öziş, Ü.,** 1983, Su Yapıları, Ege Üniversitesi Matbaası, İzmir

Şentürk, F. (1988). Barajların Projelendirilmesinde Hidrolik Esaslar, T.C. Bayındırlık ve İskan bakanlığı, Devlet Su İşleri Genel Müdürlüğü, Ankara.

Tuna A., Su Kaynakları Mühendisliği, Fırat Üniversitesi Matbaası, Elazığ

Turan, K. 2004, Development of a Computer Program For Optimum Design Of Diversion Weirs, A Thesis Submitted to The Graduate School of Natural and Applied Sciences of Middle East Technical University, Ankara

Wu, D.L. et al. (2007). Inference and learning methodology of belief-rule-based expert system for pipeline leak detection, Expert System with Applications, 103-113.

Yanmaz, A.M. (2006). Applied Water Resources Engineering, METU Press, Ankara.

Yanmaz, A.M., Turan, K.H. (2005). Computer-assisted design of diversion weir, Computing in Civil Engineering, International Conference on Computing in Civil Engineering, July 12-15, 2005, Cancun, Mexico.

Yanmaz, A. M., Ozaydin, V. (2000). An Approach to Optimum Hydraulic Design of Diversion Weirs, Watershed Management and Operations Management Conferences 2000, Edited by Marshall Flug; Donald Frevert; and David W. Watkins Jr., June 20-24, Fort Collins, Colorado, U.S., 1-10.

Gürsoy, E., Regülatörlerin Projelendirme Esasları, DSİ, 2010

İnternet Kaynakları

(<http://www.kimyamuhendisi.com>)

(<https://tr.wikipedia.org/wiki/JavaScript>).

(Tim Berners-Lee, "Information Management: A Proposal." CERN (March 1989, May 1990).

(<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQ2FzY2FkaW5nX1N0eWxlX1NoZWV0cw>).

(<https://docs.python.org/3/faq/general.htm>).

(<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTnVtUHk>).

(<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTm9kZS5qcw>).

([http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dp
a2kvTWF0cGxvdGxpYg](http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dp
a2kvTWF0cGxvdGxpYg)).

(Schlueter, Isaac Z. (25 Mar 2013). "Forget CommonJS. It's dead. **We are server side JavaScript.**". *GitHub*).

(<http://arstechnica.com/information-technology/2015/11/visual-studio-now-supports-debugging-linux-apps-code-editor-now-open-source/>).

([http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3d
pa2kvUHlDaGFybQ](http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3d
pa2kvUHlDaGFybQ)).



EKLER

Ek-1

```
2782 <TextField
2783   id="standard-number"
2784   label="X KATSAYISI"
2785   value={this.state.x}
2786   onChange={this.handleChange("x")}
2787   type="number"
2788   className={"x"}
2789   InputLabelProps={{
2790     shrink: true
2791   }}
2792   margin="normal"
2793 />
2794 <TextField
2795   id="standard-number"
2796   style={{
2797     fontSize: "26px"
2798   }}
2799   label="Y KATSAYISI"
2800   value={this.state.y}
2801   onChange={this.handleChange("y")}
2802   type="number"
2803   className={"y"}
2804   InputLabelProps={{
2805     shrink: true
2806   }}
2807   margin="normal"
2808
698   for (let i = 0; i < this.data1.length; i++) {
699     this.data1[i].x = parseFloat(
700       (this.data1copy[i].x * mystate.x).toFixed(2)
701     );
702     this.data1[i].y = parseFloat(
703       (this.data1copy[i].y * mystate.y).toFixed(2)
704     );
705   }
706 };
707
```

Ek-2

```
<TextField
id="standard-number"
style={{
fontSize: "26px"
}}
label="Minimum"
value={this.state.minimum}
onChange={this.handleChange("minimum")}
type="number"
className={"y"}
InputLabelProps={{
shrink: true
}}
margin="normal"
/>
<TextField
id="standard-number"
style={{
fontSize: "26px"
}}
label="Ortalama"
value={this.state.ortalama}
onChange={this.handleChange("ortalama")}
type="number"
className={"y"}
InputLabelProps={{
shrink: true
}}
margin="normal"
/>
<TextField
id="standard-number"
style={{
fontSize: "26px"
}}
label="Maksimum"
value={this.state.maksimum}
onChange={this.handleChange("maksimum")}
type="number"
className={"y"}
InputLabelProps={{
shrink: true
}}
margin="normal"
/>
```

Ek-3

```
for (let j = 0; j < 3; j++) {
for (let i = 0; i < this.data1.length; i++) {
if (this.kesitdegerleri[j] > this.data1[i].y) {
let uzunykenari = this.data1[i - 1].y - this.data1[i].y;
let kisay = this.data1[i - 1].y - this.kesitdegerleri[j];
let altuzunkenar = this.data1[i].x - this.data1[i - 1].x;
let cevap = (kisay * altuzunkenar) / uzunykenari;
cevap = cevap + this.data1[i - 1].x;

this.charts[j][0].x = parseFloat(cevap.toFixed(2));
```

```

for (let z = i + 1; z < this.data1.length; z++) {
  if (this.kesitdegerleri[j] < this.data1[z].y) {
    let uzunykenar1 = this.data1[z].y - this.data1[z - 1].y;
    let kisay1 = this.data1[z].y - this.kesitdegerleri[j];
    let altuzunkenar1 = this.data1[z - 1].x - this.data1[z].x;
    let cevap1 = (kisay1 * altuzunkenar1) / uzunykenar1;
    cevap1 = cevap1 + this.data1[z].x;

    this.charts[j][1].x = parseFloat(cevap1.toFixed(2));
    break;
  }
}
break;
}
}
}

```

Ek-4

```

new Chart(this.ctx, {
  plugins: [ChartDataLabels],
  type: "scatter",
  data: {
    datasets: [
      {
        label: "Bağlama Enkesiti",
        data: this.data1,
        showLine: true,

        pointStyle: "circle",
        backgroundColor: "red",
        fill: false,
        lineTension: 0,
        borderColor: "red",
        pointRadius: 3,
        borderWidth: 3,
        datalabels: {
          formatter: function(value, context) {
            return value.x + "," + value.y;
          },

          font: {
            size: 10
          },
          align: "top"
        },
        {
          label: "Minimum Su Seviyesi",
          data: this.charts[0],
          showLine: true,

          pointStyle: "circle",
          backgroundColor: "orange",
          fill: false,
          lineTension: 0,
          borderColor: "orange",
          pointRadius: 3,
          borderWidth: 3,

```

```

datalabels: {
  formatter: function(value, context) {
    return value.x + "," + value.y;
  },

  font: {
    size: 10
  },
  align: "top"
},
{
  label: "Ortalama Su Seviyesi",
  data: this.charts[1],
  showLine: true,

  pointStyle: "circle",
  backgroundColor: "#731963",
  fill: false,
  lineTension: 0,
  borderColor: "#731963",
  pointRadius: 3,
  borderWidth: 3,
  datalabels: {
    formatter: function(value, context) {
      return value.x + "," + value.y;
    },

    font: {
      size: 10
    },
    align: "top"
  },
  {
    label: "Maksimum Su Seviyesi",
    data: this.charts[2],
    showLine: true,

    pointStyle: "circle",
    backgroundColor: "#72E1D1",
    fill: false,
    lineTension: 0,
    borderColor: "#72E1D1",
    pointRadius: 3,
    borderWidth: 3,
    datalabels: {
      formatter: function(value, context) {
        return value.x + "," + value.y;
      },

      font: {
        size: 10
      },
      align: "top"
    }
  }
],
},
options: {

```

```

scales: {
  xAxes: [
    {
      beginAtZero: true,
      type: "linear",
      position: "bottom",
      scaleLabel: {
        display: true,
        labelString: "EŞEL SIFIR KOTU",
        fontSize: 22,
        padding: {
          left: 15,
          top: 0
        }
      }
    }
  ],
  yAxes: [
    {
      beginAtZero: true,
      type: "linear",
      position: "top",
      scaleLabel: {
        display: true,
        labelString: "Y",
        fontSize: 22
      }
    }
  ],
  backgroundColor: "red"
},
response: true,
legend: {
  labels: {
    fontColor: "black",
    fontSize: 22
  }
}
});

```

Ek-5

```

for (let j = 1; j < maksimumdizi.length - 1; j++) {
  if (maksimumdizi[j].y >= 0) {
    let sonuc =
maksimumdizi[j].y * (maksimumdizi[j - 1].x - maksimumdizi[j + 1].x);
cevaplar[2] += sonuc;
  } else if (maksimumdizi[j].y < 0) {
    let sonuc =
maksimumdizi[j].y * (maksimumdizi[j - 1].x - maksimumdizi[j + 1].x);

cevaplar[2] += sonuc;
  }
}

if (maksimumdizi[0].y < 0) {
  let sonuc =
maksimumdizi[0].y *
(maksimumdizi[maksimumdizi.length - 1].x - maksimumdizi[1].x);

```

```

cevaplar[2] -= sonuc;
} else {
let sonuc =
maksimumdizi[0].y *
(maksimumdizi[maksimumdizi.length - 1].x - maksimumdizi[1].x);

cevaplar[2] += sonuc;
}
if (maksimumdizi[maksimumdizi.length - 1].y < 0) {
cevaplar[2] -=
maksimumdizi[maksimumdizi.length - 1].y *
(maksimumdizi[maksimumdizi.length - 2].x - maksimumdizi[0].x);
} else {
cevaplar[2] +=
maksimumdizi[maksimumdizi.length - 1].y *
(maksimumdizi[maksimumdizi.length - 2].x - maksimumdizi[0].x);
}

this.minimumalan = parseFloat(cevaplar[0] / 2).toFixed(2);
this.ortalamaalan = parseFloat(cevaplar[1] / 2).toFixed(2);
this.maksimumalan = parseFloat(cevaplar[2] / 2).toFixed(2);

islak çevreyi hesaplamak için aşağıdaki kod yazılmıştır .
this.islakcevre = [0, 0, 0];
for (let i = 0; i < 3; i++) {
let starting = 0;

while (this.charts[i][0].y < this.data1[starting].y) starting++;

const startingyedek = starting;
while (this.charts[i][0].y > this.data1[starting + 1].y) {
this.islakcevre[i] += Math.sqrt(
Math.pow(this.data1[starting].x - this.data1[starting + 1].x, 2) +
Math.pow(this.data1[starting].y - this.data1[starting + 1].y, 2)
);

starting++;
}

this.islakcevre[i] += Math.sqrt(
Math.pow(this.data1[startingyedek].x - this.charts[i][0].x, 2) +
Math.pow(this.data1[startingyedek].y - this.charts[i][0].y, 2)
);

this.islakcevre[i] += Math.sqrt(
Math.pow(this.data1[starting].x - this.charts[i][1].x, 2) +
Math.pow(this.data1[starting].y - this.charts[i][1].y, 2)
);
}

this.minimumislakcevre = parseFloat(this.islakcevre[0].toFixed(2));
this.ortalamaislakcevre = parseFloat(this.islakcevre[1].toFixed(2));
this.maksimumislakcevre = parseFloat(this.islakcevre[2].toFixed(2));

```

Ek-6

```

this.minimumyaricap = parseFloat(
this.minimumalan / this.minimumislakcevre

```

```

).toFixed(2);
this.ortalamayaricap = parseFloat(
this.ortalamaalan / this.ortalamaaislakcevre
).toFixed(2);
this.maksimumyaricap = parseFloat(
this.maksimumalan / this.maksimumislakcevre
).toFixed(2);

```

Ek-7

```

<Container>
<Row>
<Col>
<div className="p-3 mb-2 bg-danger text-white">Çevre Hesabı</div>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Minimum Çevre</p>
</Col>
<Col>
<p className="bg-primary text-white">
{this.minimumislakcevre + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Ortalama Çevre</p>
</Col>
<Col>
<p className="bg-primary text-white ">
{this.ortalamaaislakcevre + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">Maksimum Çevre</p>
</Col>
<Col>
<p className="bg-primary text-white">
{this.maksimumislakcevre + "m"}
</p>
</Col>
</Row>
<Row />
<Row>
<Col>
<div className="p-3 mb-2 bg-danger text-white">Alan Hesabı</div>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Minimum Alan</p>
</Col>
<Col>
<p className="bg-primary text-white">{this.minimumalan + "m²"}</p>

```

```

</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Ortalama Alan</p>
</Col>
<Col>
<p className="bg-primary text-white ">
{this.ortalamaalan + "m²"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">Maksimum Alan</p>
</Col>
<Col>
<p className="bg-primary text-white">
{this.maksimumalan + "m²"}
</p>
</Col>
</Row>
<Row />
<Row>
<Col>
<div className="p-3 mb-2 bg-danger text-white">
Yarıçap Hesabı
</div>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Minimum Yarıçap</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.minimumyaricap).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Ortalama Yarıçap</p>
</Col>
<Col>
<p className="bg-primary text-white ">
{parseFloat(this.ortalamayaricap).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">Maksimum Yarıçap</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.maksimumyaricap).toFixed(2) + "m"}
</p>
</Col>
</Row>

```

</Container>

Ek-8

```
<Row>
<Col>
<h4 className={"text-warning"}>Akarsuyunun Genel Pürüzlülüğü</h4>
<br />
<select
value={this.state.ndegeri}
className="browser-default custom-select"
onChange={this.akarsupuruzlulugudegeri}
>
<option value="0.029">0.029</option>
<option value="0.031">0.031</option>
<option value="0.033">0.033</option>
<option value="0.035">0.035</option>
<option value="0.045">0.045</option>
<option value="0.060">0.060</option>
</select>
<br />
<h5 className={"text-success"} id={"hadibe"}>
{this.myMap.get(this.state.ndegeri)}
</h5>
</Col>
</Row>
<br />
<Row>
<Col>
<h4 className={"text-warning"}>Akarsu Eğimi</h4>
<br />
<select
value={this.state.jdegeri}
className="browser-default custom-select"
onChange={this.akarsuegimi}
>
<option value="0.0008">8.0*0.0001</option>
<option value="0.00085">8.5*0.0001</option>
<option value="0.0009">9.0*0.0001</option>
<option value="0.001">10.0*0.0001</option>
<option value="0.0011">11.0*0.0001</option>
<option value="0.00115">11.5*0.0001</option>
<option value="0.0012">12.0*0.0001</option>
<option value="0.00125">12.5*0.0001</option>
</select>
</Col>
</Row>
```

Ek-9

```
this.minimumdebi =
(1 / nextstate.ndegeri) *
Math.pow(this.minimumyaricap, 2 / 3) *
Math.pow(nextstate.jdegeri, 1 / 2) *
this.minimumalan;
this.maksimumdebi =
(1 / nextstate.ndegeri) *
Math.pow(this.maksimumyaricap, 2 / 3) *
```

```

Math.pow(nextstate.jdegeri, 1 / 2) *
this.maksimumalan;
this.ortalamadebi =
(1 / nextstate.ndegeri) *
Math.pow(this.ortalamayaricap, 2 / 3) *
Math.pow(nextstate.jdegeri, 1 / 2) *
this.ortalamaalan;

```

Ek-10

```

let minimum = this.data1[0].y;
for (let i = 1; i < this.data1.length; i++) {
  if (this.data1[i].y < minimum) {
    minimum = this.data1[i].y;
  }
}

this.minimumyukseklk = nextstate.minimum - minimum;
this.ortalamayukseklk = nextstate.ortalama - minimum;
this.maksimumyukseklk = nextstate.maksimum - minimum;

```

Ek-11

```

<Row>
<Col>
<div className="p-3 mb-2 bg-danger text-white">Debi Hesabı</div>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Minimum Debi</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.minimumdebi).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Ortalama Debi</p>
</Col>
<Col>
<p className="bg-primary text-white ">
{parseFloat(this.ortalamadebi).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">Maksimum Debi</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.maksimumdebi).toFixed(2) + "m"}
</p>
</Col>
</Row>

```

```

<Row>
<Col>
<div className="p-3 mb-2 bg-danger text-white">
Yükseklik Hesabı
</div>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Minumum Yükseklik</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.minimumyukseklık).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Ortalama Yükseklik</p>
</Col>
<Col>
<p className="bg-primary text-white ">
{parseFloat(this.ortalamayukseklık).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">Maksimum Yükseklik</p>
</Col>
<Col>
<p className="bg-primary text-white">
{" "}
{parseFloat(this.maksimumyukseklık).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row />

```

Ek-12

```

from flask import Flask, jsonify
from flask import make_response
from flask import request
import numpy as np
import matplotlib.pyplot as plt
from flask_cors import CORS
import base64
from PIL import Image
from io import BytesIO

app = Flask(__name__)
cors = CORS(app, resources={r"/api/*": {"origins": "*"}})
fdegeri = None

## first lane
@app.route('/api/anahtar', methods=['POST'])

```

```

def create_product():
    a = 0

    content = request.json

    points = np.array(content)
    # get x and y vectors
    x = points[:, 0]
    y = points[:, 1]
    # calculate polynomial
    z = np.polyfit(x, y, 3)
    f = np.poly1d(z)

    print(f(87.80))
    print(f(190.30))
    print(f(303.47))

    # calculate new x's and y's
    x_new = np.linspace(x[0], x[-1], 50)
    y_new = f(x_new)
    print(content)
    plt.plot(x, y, 'o', x_new, y_new)
    plt.xlabel("DEBI")
    plt.ylabel("Y")
    plt.title("ANAHTAR ")
    # plt.xlim([x[0] - 0, x[-1] + 1])

    plt.savefig('books_read.png')
    with open("books_read.png", "rb") as image_file:
        encoded_string = base64.b64encode(image_file.read())
    print("xx")
    return encoded_string

@app.route('/api/anahtar1', methods=['POST'])
def create_product12():
    content12 = request.data
    print(content12)
    print("mm")
    return "32321"

if __name__ == '__main__':
    app.run(debug=True) # !flask/bin/python

```

Ek-13

```

<img
className="vayne"
src={this.anahtaregrisiresmi}
alt="AnahtarEğris"
/>

```

Ek-14

```

new Chart(this.ctx4, {
    plugins: [ChartDataLabels],
    type: "scatter",
    data: {
        datasets: [
            {

```

```

label: "Yükseklik - Alan",
data: this.yukseklik_alanfigi,
showLine: true,
pointStyle: "circle",
backgroundColor: "red",
fill: false,
lineTension: 0,
borderColor: "red",
pointRadius: 3,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return value.x + "," + value.y;
  },

  font: {
    size: 10
  },
  align: "top"
},
options: {
  scales: {
    xAxes: [
      {
        beginAtZero: true,
        type: "linear",
        position: "bottom",
        scaleLabel: {
          display: true,
          labelString: "Alan",
          fontSize: 22,
          padding: {
            left: 15,
            top: 0
          }
        }
      }
    ],
    yAxes: [
      {
        beginAtZero: true,
        type: "linear",
        position: "top",
        scaleLabel: {
          display: true,
          labelString: "Yükseklik",
          fontSize: 22
        }
      }
    ],
    backgroundColor: "red"
  },
  response: true,
  legend: {
    labels: {
      fontColor: "black",
      fontSize: 22
    }
  }
}

```

```

    }
    }
    });
    new Chart(this.ctx5, {
    plugins: [ChartDataLabels],
    type: "scatter",
    data: {
    datasets: [
    {
    label: "Yükseklik - Çevre",
    data: this.yuksekklik_islakcevre,
    showLine: true,

    pointStyle: "circle",
    backgroundColor: "red",
    fill: false,
    lineTension: 0,
    borderColor: "red",
    pointRadius: 3,
    borderWidth: 3,
    datalabels: {
    formatter: function(value, context) {
    return value.x + "," + value.y;
    },

    font: {
    size: 10
    },
    align: "top"
    }
    }
    ],
    },
    options: {
    scales: {
    xAxes: [
    {
    beginAtZero: true,
    type: "linear",
    position: "bottom",
    scaleLabel: {
    display: true,
    labelString: "Alan",
    fontSize: 22,
    padding: {
    left: 15,
    top: 0
    }
    }
    }
    ],
    yAxes: [
    {
    beginAtZero: true,
    type: "linear",
    position: "top",
    scaleLabel: {
    display: true,
    labelString: "Yükseklik",

```

```

fontSize: 22
}
}
],
backgroundColor: "red"
},
response: true,
legend: {
labels: {
fontColor: "black",
fontSize: 22
}
}
}
});
new Chart(this.ctx3, {
plugins: [ChartDataLabels],
type: "scatter",
data: {
datasets: [
{
label: "Yükseklik-Yarıçap",
data: this.yukseklık_yaricapgrafigi,
showLine: true,

pointStyle: "circle",
backgroundColor: "red",
fill: false,
lineTension: 0,
borderColor: "red",
pointRadius: 3,
borderWidth: 3,
datalabels: {
formatter: function(value, context) {
return value.x + "," + value.y;
},
font: {
size: 10
},
align: "top"
}
}
]
},
options: {
scales: {
xAxes: [
{
beginAtZero: true,
type: "linear",
position: "bottom",
scaleLabel: {
display: true,
labelString: "Yarıçap",
fontSize: 22,
padding: {
left: 15,
top: 0
}
}
}
]
}
}
}

```

```

    }
  ],
  yAxes: [
    {
      beginAtZero: true,
      type: "linear",
      position: "top",
      scaleLabel: {
        display: true,
        labelString: "Yükseklik",
        fontSize: 22
      }
    }
  ],
  backgroundColor: "red"
},
response: true,
legend: {
  labels: {
    fontColor: "black",
    fontSize: 22
  }
}
}
});

```

Ek-15

```

<TextField
  id="standard-number"
  label="a değeri"
  value={this.state.adegeri}
  onChange={this.handleChange("adegeri")}
  type="number"
  className={"x"}
  InputLabelProps={{
    shrink: true
  }}
  margin="normal"
/>

```

Ek-16

```

<TextField

  id="standard-number"

  label="h katsayısı"

  value={this.state.hdegeri}

  onChange={this.handleChange("hdegeri")}

  type="number"

  className={"x"}

  InputLabelProps={{

    shrink: true

  }}


```

```
margin="normal"
/>
```

Ek-17

```
<Container>

<Col>
<h4 className={"text-warning"}>
Kaplamasız kanal için m değerleri
</h4>
</Col>
<Row>
<select
value={this.state.mdegeri}
className="browser-default custom-select"
onChange={this.mchanger}
>
<option value="0">0</option>
<option value="1.6">1.6</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
</select>
</Row>
</Container>

<h5 className={"text-success"}>
{this.mdegerli.get(this.state.mdegeri)}
</h5>
<Container>
<Col>
<h4 className={"text-warning"}>
Kaplamasız kanal için n değerleri
</h4>
</Col>
<Row>
<select
value={this.state.ndegeriisale}
className="browser-default custom-select"
```

```

onChange={this.nisalechanger}
>
<option value="0.040">0.040</option>
<option value="0.033">0.033</option>
<option value="0.020">0.020</option>
</select>
<Col>
<h4 className={"text-success"}>
{this.isale_n_degerleri.get(this.state.ndegeriisale)}
</h4>
</Col>
</Row>
</Container>
<Container>
<Col>
<h4 className={"text-warning"}>
Kaplamasız kanaliçin eşel sıfır kotu
</h4>
</Col>
<Row>
<select
value={this.state.eselsifirkotu}
className="browser-default custom-select"
onChange={this.eselsifirkotu}
>
<option value="303">303</option>
<option value="600">600</option>
<option value="607">607</option>
<option value="953">953</option>
<option value="1020">1020</option>
<option value="2000">2000</option>
</select>
</Row>
</Container>

```

Ek-18

// h ve b DEĞERLERİ

```

for (let i = this.state.hdegeri; i <= 3.02; i += 0.02) {
  this.kaplamasizzemintablo.push({
    h: parseFloat(hsondeger).toFixed(2),
    b: Math.round(5 * hsondeger)
  });
  hsondeger += 0.02;
}
// ALAN
for (let i = 0; i < this.kaplamasizzemintablo.length; i++) {
  this.kaplamasizzemintablo[i].alan = parseFloat(
    this.kaplamasizzemintablo[i].h * this.kaplamasizzemintablo[i].b +
    this.state.mdegeri * Math.pow(this.kaplamasizzemintablo[i].h, 2)
  ).toFixed(2);
}
// ISLAK ÇEVRE
for (let i = 0; i < this.kaplamasizzemintablo.length; i++) {
  this.kaplamasizzemintablo[i].cevre = parseFloat(
    this.kaplamasizzemintablo[i].b +
    2 *
    this.kaplamasizzemintablo[i].h *
    Math.sqrt(1 + Math.pow(this.state.mdegeri, 2))
  ).toFixed(2);
}
// HİDROLİK YARIÇAP
for (let i = 0; i < this.kaplamasizzemintablo.length; i++) {
  this.kaplamasizzemintablo[i].R = parseFloat(
    this.kaplamasizzemintablo[i].alan / this.kaplamasizzemintablo[i].cevre
  ).toFixed(2);
}
this.isale_n_degerleri.set(0.04, "Gevşek zemin");
this.isale_n_degerleri.set(0.033, "Toprak zemin");
this.isale_n_degerleri.set(0.025, "Sıkı zemin");
let qisale = this.minimumdebi * this.state.adegeri;
console.log(qisale + "qisale");
let qn = qisale * this.state.ndegeriisale;
// TABAN EĞİMİ
for (let i = 0; i < this.kaplamasizzemintablo.length; i++) {
  let rikiboluuc =

```

```

Math.pow(this.kaplamasizzemintablo[i].R, 2 / 3) *
this.kaplamasizzemintablo[i].alan;

let cevap = qn / rikiboluuc;
let cevap2 = Math.pow(cevap, 2);
this.kaplamasizzemintablo[i].j = parseFloat(cevap2).toFixed(5);
}
// KABARTMA KOTU
for (let i = 0; i < this.kaplamasizzemintablo.length; i++) {
this.kaplamasizzemintablo[i].kabartmakotu = parseFloat(
this.state.kanalinsonundakisuyuzukotu +
this.kaplamasizzemintablo[i].j * this.state.isalekanaliuzunlugu +
0.5
).toFixed(2);
}
// KK-ESK
for (let i = 0; i < this.kaplamasizzemintablo.length; i++) {
this.kaplamasizzemintablo[i].kk_esk = parseFloat(
this.kaplamasizzemintablo[i].kabartmakotu - this.state.eselsifirkotu
).toFixed(2);
}
bulunan tablo değerleri;
<BootstrapTable

striped
hover
keyField="h"
data={this.kaplamasizzemintablo}
columns={this.columns}
trStyle={this.rowClassNameFormat}
/>

```

Ek-19

```

hsondeger = 1;

//b ve h DEĞERLERİ
this.kaplamalikanal = [];
for (let i = this.state.hdegeri; i <= 3.02; i += 0.02) {
this.kaplamalikanal.push({

```

```

h: parseFloat(hsondeger).toFixed(2),
b: Math.round(5 * hsondeger)
});
hsondeger += 0.02;
}
// ALAN
for (let i = 0; i < this.kaplamalikanal.length; i++) {
this.kaplamalikanal[i].alan = parseFloat(
this.kaplamalikanal[i].h * this.kaplamalikanal[i].b +
this.state.kaplamalimdegeri *
Math.pow(this.kaplamasizzemintablo[i].h, 2)
).toFixed(2);
}
// ISLAK ÇEVRE
for (let i = 0; i < this.kaplamalikanal.length; i++) {
this.kaplamalikanal[i].cevre = parseFloat(
this.kaplamalikanal[i].b +
2 *
this.kaplamalikanal[i].h *
Math.sqrt(1 + Math.pow(this.state.kaplamalimdegeri, 2))
).toFixed(2);
}
// HİDROLİK YARIÇAP
for (let i = 0; i < this.kaplamalikanal.length; i++) {
this.kaplamalikanal[i].R = parseFloat(
this.kaplamalikanal[i].alan / this.kaplamalikanal[i].cevre
).toFixed(2);
}
let qisale2 = qisale * this.state.kaplamalindegeri;
// TABAN EĞİMİ
for (let i = 0; i < this.kaplamalikanal.length; i++) {
let rikiboluuc =
Math.pow(this.kaplamalikanal[i].R, 2 / 3) *
this.kaplamalikanal[i].alan;

let cevap = qisale2 / rikiboluuc;
let cevap2 = Math.pow(cevap, 2);
this.kaplamalikanal[i].j = parseFloat(cevap2).toFixed(4);
}

```

```

}

//HIZ
for (let i = 0; i < this.kaplamalikanal.length; i++) {
  this.kaplamalikanal[i].hiz = parseFloat(
    qisale / this.kaplamalikanal[i].alan
  ).toFixed(4);
}

// KABARTMA KOTU
for (let i = 0; i < this.kaplamalikanal.length; i++) {
  this.kaplamalikanal[i].kabartmakotu = parseFloat(
    this.state.kanalinsonundakisuyuzukotu +
    this.kaplamalikanal[i].j * this.state.isalekanaliuzunlugu +
    0.5
  ).toFixed(2);
}

// KK-ESK
for (let i = 0; i < this.kaplamalikanal.length; i++) {
  this.kaplamalikanal[i].kk_esk = parseFloat(
    this.kaplamalikanal[i].kabartmakotu - this.state.eselsifirkotu
  ).toFixed(2);
}

bulunan tablo deęerleri;

<BootstrapTable
  striped
  hover
  keyField="h"
  data={this.kaplamalikanal}
  columns={this.columns}
  trStyle={this.rowClassNameFormat}
/>

```

Ek-20

```

this.kamplamasizto = this.state.gamasu * alinanr * alinanj;
this.tkr = 0.06 * (26487 - 9810) * this.state.kanald;

```

Ek-21

```
<h5 className={"text-success"}>
{this.tkr + "<" + parseFloat(this.kamplamasizto).toFixed(3)}
</h5>
```

Ek-22

```
if (this.state.danecapi < 0.1)
this.af = 51;
else if (this.state.sdanecapi > 1)
this.state.af = 36;
else
this.state.af = 44;
```

Ek-23

```
this.v = this.state.af * Math.sqrt(this.state.danecapi);
```

Ek-24

```
let sol = 9 / this.state.danecapi;
let sag = Math.sqrt(1 + 150 * Math.pow(this.state.danecapi, 3))
- 1;
this.w = sol * sag;
```

Ek-25

```
this.v1 = (this.secondv * this.state.hcokeltimderinlik) / (this.w /
1000);
```

Ek-26

```
this.v1 = (this.secondv * this.state.hcokeltimderinlik) / (this.w /
1000);
console.log(this.v1);
this.wa = (0.132 * this.secondv) /
Math.sqrt(this.state.hcokeltimderinlik);
this.lt =
(this.secondv * this.state.hcokeltimderinlik) / (this.w / 1000 -
this.wa);
```

```
console.log(this.lt);
```

Ek-27

```
this.bch =
```

```
(1.1 * this.qisale123) / ((this.v / 100) *  
    this.state.hcokeltimderinlik)
```

Ek-28

```
<h3 className={"text-success"}>Çökeltim Havuzunun  
Boyutlandırılması</h3>
```

```
<h5 className={"text-success"}>{"Vç = " + this.v + "cm/s"}</h5>
```

```
<h5 className={"text-success"}>{"Vç = " + this.secondv + "m/s"}</h5>
```

```
<h5 className={"text-success"}>Laminer Akım</h5>
```

```
<h5 className={"text-success"}>
```

```
    {"W= " + parseFloat(this.w).toFixed(2) + "mm/s"}  
</h5>
```

```
<h5 className={"text-success"}>
```

```
    {"W= " + parseFloat(this.w / 1000).toFixed(5) + "m/s"}  
</h5>
```

```
<h5 className={"text-warning"}>h değeri</h5>
```

```
<Container>
```

```
<Row>
```

```
<select
```

```
    value={this.state.hcokeltimderinlik}
```

```
    className="browser-default custom-select"
```

```
    onChange={this.dchangder123}
```

```
>
```

```
<option value="1.5">1.5</option>
```

```
<option value="2">2</option>
```

```
<option value="2.5">2.5</option>
```

```
<option value="3.0">3</option>
```

```
<option value="3.5">3.5</option>
```

```
<option value="4.0">4</option>
```

```
</select>
```

```
</Row>
```

```
</Container>
```

```
<h5 className={"text-success"}>
```

```

{"L1= " + parseFloat(this.v1).toFixed(2) + "m"}
</h5>
<h3 className={"text-success"}>Türbülanslı Akım</h3>
<h5 className={"text-success"}>
{"Wa =" + parseFloat(this.wa).toFixed(3) + "m/s"}
</h5>

<h5 className={"text-success"}>
{"Lt =" + parseFloat(this.lt).toFixed(2) + "m"}
</h5>
<h5 className={"text-success"}>
{"Le =" + parseFloat(this.le).toFixed(2) + "m"}
</h5>
<h5 className={"text-success"}>
{"Bçh=" + parseFloat(this.bch).toFixed(2) + "m"}
</h5>
<h5 className={"text-success"}>b = 4 ile 6 arasında olacak</h5>

```

Ek-29

```

for (let j = 0; j < 10; j++) {

this.table.push({
indis: j + 1,
b: parseFloat(this.bch / (j + 1)).toFixed(2)
});
}

```

Ek-30

```

<div className="container" style={{ marginTop: 50 }}>

<BootstrapTable
striped
hover
keyField="h2"
data={this.table}
columns={this.columns2}
trStyle={this.rowClassNameFormat}
/>
</div>

```

Ek-31

```
let secilenvalue = 5.44;

let value = this.qisale123 / (2.214723459 * secilenvalue);
this.he = Math.pow(value, 2 / 3);
console.log(this.he + "he");
this.pe = this.state.hcokeltimderinlik - this.he;
```

Ek-32

```
<h5 className={"text-success"}>

Çökeltim Havuzu Sonundaki Eşiğin Boyutlandırılması

</h5>
<h5 className={"text-success"}>
{"he=" + parseFloat(this.he).toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Pe=" + parseFloat(this.pe).toFixed(3) + " m"}
</h5>
```

Ek-33

```
let ust = 3 * (this.minimumdebi - 1.1 * this.qisale123);

let alt = 2 * 0.75 * this.bnet * Math.sqrt(19.62);
let cevap1 = ust / alt;
this.hmin = Math.pow(cevap1, 2 / 3);
```

Ek-34

```
this.kretkotu = this.kabartmakotuk1 - this.hmin;

this.p = this.kretkotu - this.state.tabankotu + 0.35;
console.log(this.kretkotu + "kret");
```

Ek-35

```
ust = 3 * (this.ortalamadebi - 1.1 * this.qisale123);

alt = 2 * 0.75 * this.bnet * Math.sqrt(19.62);
cevap1 = ust / alt;
this.hort = Math.pow(cevap1, 2 / 3);
this.ortalamakabartmakotu = this.kretkotu + this.hort;
```

```
console.log(this.ortalamakabartmakotu);
```

Ek-36

```
ust = 3 * (this.maksimumdebi - 1.1 * this.qisale123);  
alt = 2 * 0.75 * this.bnet * Math.sqrt(19.62);  
cevap1 = ust / alt;  
this.hmax = Math.pow(cevap1, 2 / 3);  
this.maksimumkabartmakotu = this.kretkotu + this.hmax;
```

Ek-37

```
this.baglamaayakyuk = this.maksimumkabartmakotu + 0.5;
```

Ek-38

```
this.minimumsuyukseklik = this.p + this.hmin;  
this.ortalamasuyukseklik = this.p + this.hort;  
this.maksimumsuyukseklik = this.p + this.hmax;
```

Ek-39

```
<h5 className={"text-success"}>  
  {"Bnet=" + parseFloat(this.bnet).toFixed(3) + " m"}  
</h5>  
<h5 className={"text-success"}>  
  {"hmin=" + parseFloat(this.hmin).toFixed(3) + " m"}  
</h5>  
<h5 className={"text-success"}>  
  {"Kret Kotu=" + parseFloat(this.kretkotu).toFixed(3) + " m"}  
</h5>  
<TextField  
  id="standard-number"  
  label="Taban Kotu"  
  value={this.state.tabankotu}  
  onChange={this.handleChange("tabankotu")}  
  type="number"  
  className={"x"}  
  InputLabelProps={{  
    shrink: true  
  }}  
  />
```

```

margin="normal"
/>
<h5 className={"text-success"}>
{"P=" + parseFloat(this.p).toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"hort=" + parseFloat(this.hort).toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Ort Kabartma Kotu=" +
parseFloat(this.ortalamakabartmakotu).toFixed(3) +
" m"}
</h5>
<h5 className={"text-success"}>
{"hmak=" + parseFloat(this.hmax).toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Mak Kabartma Kotu=" +
parseFloat(this.maksimumkabartmakotu).toFixed(3) +
" m"}
</h5>
<h5 className={"text-success"}>
{"Bağ ayak Yük=" + parseFloat(this.baglamaayakyuk).toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Min Su Yüksekliği (Homin)=" +
parseFloat(this.minimumsuyukseklik).toFixed(3) +
" m"}
</h5>
<h5 className={"text-success"}>
{"Ort Su Yüksekliği (Hoort)=" +
parseFloat(this.ortalamasuyukseklik).toFixed(3) +
" m"}
</h5>
<h5 className={"text-success"}>
{"Mak Su Yüksekliği (Homak)=" +
parseFloat(this.maksimumsuyukseklik).toFixed(3) +
" m"}

```

</h5>

Ek-40

```
new Chart(this.ctx6, {
  plugins: [ChartDataLabels],
  type: "scatter",
  data: {
    datasets: [
      {
        label: "Enkesit",
        data: this.data1,
        showLine: true,

        pointStyle: "circle",
        backgroundColor: "red",
        fill: false,
        lineTension: 0,
        borderColor: "red",
        pointRadius: 3,
        borderWidth: 3,
        datalabels: {
          formatter: function(value, context) {
            return "";
          },

          font: {
            size: 10
          },
          align: "top"
        }
      },
      {
        label: "p",
        data: [
          {
            x: 0,
            y: minimumy
```

```

    },
    {
      x: this.bnet + this.state.bch + 1 + 1,
      y: minimumy
    }
  ],
  showLine: true,

  pointStyle: "circle",
  backgroundColor: "orange",
  fill: false,
  lineTension: 0,
  borderColor: "orange",
  pointRadius: 0,
  borderWidth: 3,
  datalabels: {
    formatter: function(value, context) {
      return "";
    },

    font: {
      size: 10
    },
    align: "top"
  },
  {
    label: "x",
    data: [
      { x: -1, y: minimumy },
      {
        x: -1,
        y: minimumy - 1
      },
      {
        x: this.bnet + this.state.bch + 1 + 1,
        y: minimumy - 1
      }
    ]
  }
]

```

```

],
showLine: true,

pointStyle: "circle",
backgroundColor: "orange",
fill: false,
lineTension: 0,
borderColor: "orange",
pointRadius: 0,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return "";
  },

  font: {
    size: 10
  },
  align: "top"
},

{
  label: "Kret Kotu",
  data: [
    {
      x: 0,
      y: this.kretkotu - this.state.eselsifirkotu
    },
    {
      x: this.bnet,
      y: this.kretkotu - this.state.eselsifirkotu
    }
  ],
  showLine: true,

  pointStyle: "circle",
  backgroundColor: "blue",

```

```

fill: false,
lineTension: 0,
borderColor: "blue",
pointRadius: 0,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return "";
  },

```

```

font: {
  size: 10
},
align: "top"
}
},
{
  label: "Minimum Kabarma Kotu",
  data: [
    {
      x: 0,
      y: this.p + this.hmin
    },
    {
      x: this.bnet,
      y: this.p + this.hmin
    }
  ],
  showLine: true,

```

```

pointStyle: "circle",
backgroundColor: "pink",
fill: false,
lineTension: 0,
borderColor: "pink",
pointRadius: 0,
borderWidth: 3,
datalabels: {

```

```

formatter: function(value, context) {
return "";
},

```

```

font: {
size: 10
},
align: "top"
}

```

```

},
{
label: "Maksimum Kabarma Kotu",
data: [
{
x: 0,
y: this.p + this.hmax
},
{
x: this.bnet,
y: this.p + this.hmax
}
],
showLine: true,

```

```

pointStyle: "circle",
backgroundColor: "black",
fill: false,
lineTension: 0,
borderColor: "black",
pointRadius: 0,
borderWidth: 3,
datalabels: {
formatter: function(value, context) {
return "";
},

```

```

font: {
size: 10

```

```

},
align: "top"
}
},
{
label: null,
data: [
{
x: 0,
y: this.p + this.hmax + 0.5
},
{
x: 0,
y: minimumy
},
{
x: -1,
y: minimumy
},
{
x: -0.5,
y: this.p + this.hmax + 0.5
},
{
x: 0,
y: this.p + this.hmax + 0.5
}
],
showLine: true,

pointStyle: "circle",
backgroundColor: "orange",
fill: false,
lineTension: 0,
borderColor: "orange",
pointRadius: 0,
borderWidth: 3,
datalabels: {

```

```

formatter: function(value, context) {
return "";
},

```

```

font: {
size: 10
},
align: "top"
}

```

```

},
{
label: "four",
data: [
{
x: this.bnet,
y: this.p + this.hmax + 0.5
},
{
x: this.bnet + 1,
y: this.p + this.hmax + 0.5
},
{
x: this.bnet + 1,
y: minimumy
},
{
x: this.bnet,
y: minimumy
},
{
x: this.bnet,
y: this.p + this.hmax + 0.5
}
],
showLine: true,

```

```

pointStyle: "circle",
backgroundColor: "orange",

```

```

fill: false,
lineTension: 0,
borderColor: "orange",
pointRadius: 0,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return "";
  },

  font: {
    size: 10
  },
  align: "top"
},
{
  label: "four",
  data: [
    {
      x: this.bnet + this.state.bch + 1,
      y: this.p + this.hmax + 0.5
    },
    {
      x: this.bnet + this.state.bch + 1 + 0.5,
      y: this.p + this.hmax + 0.5
    },
    {
      x: this.bnet + this.state.bch + 1 + 1,
      y: minimumy
    },
    {
      x: this.bnet + this.state.bch + 1,
      y: minimumy
    },
    {
      x: this.bnet + this.state.bch + 1,
      y: this.p + this.hmax + 0.5
    }
  ]
}

```

```

    }
  ],
  showLine: true,

  pointStyle: "circle",
  backgroundColor: "orange",
  fill: false,
  lineTension: 0,
  borderColor: "orange",
  pointRadius: 0,
  borderWidth: 3,
  datalabels: {
    formatter: function(value, context) {
      return "";
    },

    font: {
      size: 10
    },
    align: "top"
  },
  {
    label: "four",
    data: [
      {
        x: this.bnet + this.state.bch + 1 + 1,
        y: minimumy
      },
      {
        x: this.bnet + this.state.bch + 1 + 1,
        y: minimumy - 1
      }
    ],
    showLine: true,

    pointStyle: "circle",
    backgroundColor: "orange",

```

```

fill: false,
lineTension: 0,
borderColor: "orange",
pointRadius: 0,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return "";
  },

  font: {
    size: 10
  },
  align: "top"
}
],
},
options: {
  scales: {
    xAxes: [
      {
        beginAtZero: true,
        type: "linear",
        position: "bottom",
        scaleLabel: {
          display: true,
          labelString: "X",
          fontSize: 22,
          padding: {
            left: 15,
            top: 0
          }
        }
      }
    ],
    yAxes: [
      {

```

```

beginAtZero: true,
type: "linear",
position: "top",
scaleLabel: {
display: true,
labelString: "Y",
fontSize: 22
}
},
],
backgroundColor: "red"
},
response: true,
legend: {
labels: {
fontColor: "black",
fontSize: 22
}
}
}
});

```

Ek-41

```

<div className="myChartdiv27">

<canvas id="myChart6" width="1300px" height="700px" />

</div>

```

Ek-42

```

<TextField

id="standard-number"
label="(V = 0.8 - 1.2 m/s arasında seçilir)"
value={this.state.vhiz}
onChange={this.handleChange("vhiz")}
type="number"
className={"x"}
InputLabelProps={{

```

```

shrink: true
}}
margin="normal"
/>

```

Ek-43

```

this.leta = this.minimumsuyukseklilik - 1.1;

this.letAb = (1.1 * this.qisalel23) / this.state.vhiz;

```

Ek-44

```

<h5 className={"text-success"}>
  {"a=" + parseFloat(this.leta).toFixed(3) + " m"}
</h5>

<h5 className={"text-success"}>
  {"A=" + parseFloat(this.letAb).toFixed(3) + " m2"}
</h5>

<h5 className={"text-success"}>
  {"b=" + parseFloat(this.letb).toFixed(3) + " m"}
</h5>

```

Ek-45

```

<h3 className={"text-success"}>
  {"Yıkama Kanalı'nın Boyutlandırılması"}
</h3>

<TextField
  id="standard-number"
  label="(Vy = 2 - 2.5 m/s arasında seçilir)"
  value={this.state.vy}
  onChange={this.handleChange("vy")}
  type="number"
  className={"x"}
  InputLabelProps={{
    shrink: true
  }}
  margin="normal"

```

```
/>
```

Ek-46

```
this.qy = 0.1 * this.qisale123;  
  
this.ay = this.qy / this.state.vy;  
  
this.rf = Math.sqrt(this.ay / Math.PI);
```

Ek-47

```
<h5 className={"text-success"}>  
  {"Qy=" + parseFloat(this.qy).toFixed(3) + " m3/s"}  
</h5>  
<h5 className={"text-success"}>  
  {"Ay=" + parseFloat(this.ay).toFixed(3) + " m2"}  
</h5>  
<h5 className={"text-success"}>  
  {"r=" + parseFloat(this.rf).toFixed(3) + " m"}  
</h5>
```

Ek-48

```
this.rf = Math.sqrt(this.ay / Math.PI);  
  
this.ry = this.rf / 2;
```

Ek-49

```
<Row>  
  
<select  
  value={this.state.kkatsayisi2}  
  className="browser-default custom-select"  
  onChange={this.smarthandleforselect("kkatsayisi2")} >  
  <option value="30">30</option>  
  <option value="60">60</option>  
  <option value="90">90</option>  
</select>  
</Row>
```

Ek-50

```
this.jy = Math.pow(
this.state.vy / (this.state.kkatsayisi2 * Math.pow(this.ry, 2 / 3)),
2
);
```

Ek-51

```
<h5 className={"text-success"}>
{"Ry=" + parseFloat(this.ry).toFixed(3) + " m"}
</h5>

<h5 className={"text-success"}>
{"Jy=" + parseFloat(this.jy).toFixed(5) + " m"}
</h5>
```

Ek-52

```
<TextField
id="standard-number"
label="(μ = 0.90 - 0.95 m/s arasında seçilir)"
value={this.state.muh}
onChange={this.handleChange("muh")}
type="number"
className={"x"}
InputLabelProps={{
shrink: true
}}
margin="normal"
/>
```

```
<TextField
id="standard-number"
label="(bç = 3 - 6 m/s arasında seçilir)"
value={this.state.bch}
onChange={this.handleChange("bch")}
type="number"
className={"x"}
InputLabelProps={{
```

```

shrink: true
}}
margin="normal"
/>

```

Ek-53

```

this.qc =
(2 / 3) *
this.state.muh *
this.state.bch *
Math.pow(this.maksimumsuyukseklik - 2.5, 1.5) *
Math.sqrt(19.62) +
this.state.muh *
2.5 *
this.state.bch *
Math.sqrt(19.62 * (this.maksimumsuyukseklik - 2.5));

```

Ek-54

```

<h5 className={"text-success"}>
{"Qç=" + parseFloat(this.qc).toFixed(2) + " m3/s"}

</h5>

```

Ek-55

```

@app.route('/api/anahtar', methods=['POST'])
def create_product():
a = 0
content = request.json
points = np.array(content)
x = points[:, 0]
y = points[:, 1]
z = np.polyfit(x, y, 3)
f = np.poly1d(z)
print(f(87.80))
print(f(190.30))
print(f(303.47))
# calculate new x's and y's

```

```

x_new = np.linspace(x[0], x[-1], 50)
y_new = f(x_new)
print(content)
if __name__ == '__main__':app.run(debug=True) # !flask/bin/python

```

Ek-56

```

<img

className="vayne"
src={this.anahtaregrisiresi}
alt="AnahtarEğrisi"
/>
<h5 className={"text-success"}>
{"X = " + this.minx1.toFixed(2) + " h3min = " + this.minx1y}
</h5>
<h5 className={"text-success"}>
{"X = " + this.ortx1.toFixed(2) + " h3ort = " + this.ortx1y}
</h5>
<h5 className={"text-success"}>
{"X = " + this.makx1.toFixed(2) + " h3maks = " + this.makx1y}
</h5>

```

Ek-57

```

for (let m = 0; m < this.secondtable.length; m++) {

this.secondtable[m].xussu = parseFloat(
this.secondtable[m].x * this.hmax
).toFixed(3);
}

for (let x = 0; x < this.secondtable.length; x++) {
this.secondtable[x].ylussu = parseFloat(
this.secondtable[x].yalt * this.hmax
).toFixed(3);
}

for (let z = 0; z < this.secondtable.length; z++) {
this.secondtable[z].y2ussu = parseFloat(
this.secondtable[z].yust * this.hmax
).toFixed(3);
}

```

Ek-58

```
<div className="container" style={{ marginTop: 50 }}>

<BootstrapTable
  striped
  hover
  keyField="h2"
  data={this.secondtable}
  columns={this.columns3}
  trStyle={this.rowClassNameFormat}
/>
```

Ek-59

```
for (let i = 0; i < this.secondtable.length; i++) {
  this.xtable[i] = {
    y: this.secondtable[i].xussu,
    x: this.secondtable[i].ylussu
  };
  this.xltable[i] = {
    y: this.secondtable[i].xussu,
    x: this.secondtable[i].y2ussu
  };
}

new Chart(this.ctx8, {
  plugins: [ChartDataLabels],
  type: "scatter",
  data: {
    datasets: [
      {
        label: "Alt Nap",
        data: this.xtable,
        showLine: true,

        pointStyle: "circle",
        backgroundColor: "red",
        fill: false,
        lineTension: 0,
```

```

borderColor: "red",
pointRadius: 3,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return (
      parseFloat(value.x).toFixed(2) +
      "," +
      parseFloat(value.y).toFixed(2)
    );
  },

  font: {
    size: 10
  },
  align: "top"
},
{
  label: "Üst Nap",
  data: this.x1table,
  showLine: true,

  pointStyle: "circle",
  backgroundColor: "orange",
  fill: false,
  lineTension: 0,
  borderColor: "orange",
  pointRadius: 3,
  borderWidth: 3,
  datalabels: {
    formatter: function(value, context) {
      return (
        parseFloat(value.x).toFixed(2) +
        "," +
        parseFloat(value.y).toFixed(2)
      );
    },
  },

```

```

font: {
  size: 10
},
align: "top"
}
}
],
},
options: {
  scales: {
    xAxes: [
      {
        beginAtZero: true,
        type: "linear",
        position: "bottom",
        scaleLabel: {
          display: true,
          labelString: "Y",
          fontSize: 22,
          padding: {
            left: 15,
            top: 0
          }
        }
      }
    ],
    yAxes: [
      {
        beginAtZero: true,
        type: "linear",
        position: "top",
        scaleLabel: {
          display: true,
          labelString: "X",
          fontSize: 22
        }
      }
    ]
  }
}

```

```

],
backgroundColor: "red"
},
response: true,
legend: {
labels: {
fontColor: "black",
fontSize: 22
}
}
}
});

```

Ek-60

```

<div className="myChartdiv27">

<canvas id="myChart8" width="1300px" height="700px" />

</div>

```

Ek-61

```

this.hussul = this.hmax / this.maksimumsuyukseklik;

this.c21 = 1 - Math.pow(this.hussul, 2) * (3 - 2 * this.hussul);
this.c12 = 1 - Math.pow(this.hussul, 2) * (2 - this.hussul);
this.tanjantalfa = Math.sqrt(this.c21 / (this.c12 * this.gamaussul -
0.4));
let mq = Math.atan(this.tanjantalfa);

this.alfal = (mq * 180) / Math.PI;

```

Ek-62

```

<h5 className={"text-success"}>

{"h' = " + parseFloat(this.hussul).toFixed(2)}

</h5>

<h5 className={"text-success"}>

{"γ' = " + parseFloat(this.gamaussul).toFixed(2)}

</h5>

<h5 className={"text-success"}>

{"c2 = " + parseFloat(this.c21).toFixed(2)}

```

```

</h5>
<h5 className={"text-success"}>
{"c1 = " + parseFloat(this.c12).toFixed(2)}
</h5>
<h5 className={"text-success"}>
{"tan $\alpha$  = " + parseFloat(this.tanjantalfa).toFixed(2)}
</h5>
<h5 className={"text-success"}>
{" $\alpha$  =" + parseFloat(this.alfa1).toFixed(2)}
</h5>

```

Ek-63

```

this.G =
0.5 * 23.5 * Math.pow(this.maksimumsuyukseklik, 2) * this.tanjantalfa;
console.log(this.hmax);
this.G0 = 0.5 * 23.5 * Math.pow(this.hmax, 2) * this.tanjantalfa;
this.x12 = 0.5 * 9.79 * Math.pow(this.maksimumsuyukseklik, 2);
this.x0 = 0.5 * 9.79 * Math.pow(this.hmax, 2);
this.u1 =
0.5 *
0.4 *
9.79 *
Math.pow(this.maksimumsuyukseklik, 2) *
this.tanjantalfa;
this.gkuvvet = (2 / 3) * this.maksimumsuyukseklik * this.tanjantalfa;
this.g0kuvvet =
this.maksimumsuyukseklik * this.tanjantalfa -
(1 / 3) * this.hmax * this.tanjantalfa;
this.g30kuvvet = (1 / 3) * this.maksimumsuyukseklik;
this.g40kuvvet = this.maksimumsuyukseklik - (2 / 3) * this.hmax;
this.g50kuvvet = (2 / 3) * this.maksimumsuyukseklik *
this.tanjantalfa;
this.mo =
this.G * (2 / 3) * this.maksimumsuyukseklik * this.tanjantalfa +
this.x0 * (this.maksimumsuyukseklik - (2 / 3) * this.hmax);
console.log(this.u1);

```

Ek-64

```
<Container>
<Row>
<Col>
<p className="p-3 mb-2 bg-danger text-white">
```

Ek-65

```
</p>
</Col>
<Col>
<p className="p-3 mb-2 bg-danger text-white">Mevcut Kuvvetler</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">G= {this.G.toFixed(2)}</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.gkuvvet).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white">Go= {this.G0.toFixed(2)}</p>
</Col>
<Col>
<p className="bg-primary text-white ">
{parseFloat(this.g0kuvvet).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">x= {this.x12.toFixed(2)}</p>
</Col>
```

```

<Col>
<p className="bg-primary text-white">
{parseFloat(this.g30kuvvet).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">
x0 = {this.x0.toFixed(2)}
</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.g40kuvvet).toFixed(2) + "m"}
</p>
</Col>
</Row>
<Row>
<Col>
<p className="bg-primary text-white ">u = {this.u1.toFixed(2)}</p>
</Col>
<Col>
<p className="bg-primary text-white">
{parseFloat(this.g50kuvvet).toFixed(2) + "m"}
</p>
</Col>
</Row>
</Container>

```

Ek-66

```

this.mo =
this.G * (2 / 3) * this.maksimumsuyukseklik * this.tanjantalfa +
this.x0 * (this.maksimumsuyukseklik - (2 / 3) * this.hmax);
console.log(this.u1);
this.MD =
this.G0 *
(this.maksimumsuyukseklik * this.tanjantalfa -

```

```

(1 / 3) * this.hmax * this.tanjantalfa) +
this.x12 * (1 / 3) * this.maksimumsuyukseklik +
this.u1 * (2 / 3) * this.maksimumsuyukseklik * this.tanjantalfa;
this.nd = this.mo / this.MD;

```

Ek-67

```

this.mo =

this.G * (2 / 3) * this.maksimumsuyukseklik * this.tanjantalfa +
this.x0 * (this.maksimumsuyukseklik - (2 / 3) * this.hmax);
console.log(this.u1);
this.MD =
this.G0 *
(this.maksimumsuyukseklik * this.tanjantalfa -
(1 / 3) * this.hmax * this.tanjantalfa) +
this.x12 * (1 / 3) * this.maksimumsuyukseklik +
this.u1 * (2 / 3) * this.maksimumsuyukseklik * this.tanjantalfa;
this.nd = this.mo / this.MD;

```

Ek-68

```

<Container>

<Row>
<select
value={this.state.surtunmekatsayisi}
className="browser-default custom-select"
onChange={this.smarthandleforselect("surtunmekatsayisi")} // handle et
>

<option value="0.75">0.75</option>
<option value="0.7">0.7</option>
<option value="0.4">0.4</option>
<option value="0.3">0.3</option>
<option value="0.8">0.8</option>
</select>
</Row>
</Container>

```

Ek-69

```
this.kiyas1 = this.state.surtunmekatsayisi * (this.G - this.G0 - this.u1);  
  
this.kiyas2 = this.x12 - this.x0;
```

Ek-70

```
<h5 className={"text-success"}>{"f(G-Go-u)>x-xo"}</h5>  
  
<h5 className={"text-success"}>  
  {parseFloat(this.kiyas1).toFixed(3) +  
  ">" +  
  parseFloat(this.kiyas2).toFixed(2)}  
</h5>  
  
<h1 className={"text-success"}>{"Taban Gerilmelerinin Tayini"}</h1>
```

Ek-71

```
this.Gnet = this.G - this.G0 - this.u1;  
  
this.m1 = this.mo - this.MD;  
this.b1 = this.maksimumsuyukseklik * this.tanjantalfa;  
this.e1 = this.b1 / 2 - this.m1 / this.Gnet;  
this.M0 = this.Gnet * this.e1;  
this.A1 = this.b1;  
this.W1 = Math.pow(this.b1, 2) / 6;  
this.sigmamax = this.Gnet / this.A1 + this.M0 / this.W1;  
this.sigmamin = this.Gnet / this.A1 - this.M0 / this.W1;
```

Ek-72

```
<h5 className={"text-success"}>  
  {"Gnet = N = " + parseFloat(this.Gnet).toFixed(3)}  
</h5>  
  
<h5 className={"text-success"}>  
  {"M = " + parseFloat(this.m1).toFixed(3)}  
</h5>  
  
<h5 className={"text-success"}>
```

```

{"b = " + parseFloat(this.b1).toFixed(3)}
</h5>
<h5 className={"text-success"}>
{"e = " + parseFloat(this.e1).toFixed(3)}
</h5>
<h5 className={"text-success"}>
{"MO = " + parseFloat(this.M0).toFixed(3)}
</h5>
<h5 className={"text-success"}>
{"A = " + parseFloat(this.A1).toFixed(3)}
</h5>
<h5 className={"text-success"}>
{"W = " + parseFloat(this.W1).toFixed(3)}
</h5>
<h5 className={"text-success"}>
{"σmax = " +
parseFloat(this.sigmax).toFixed(3) +
" t/m2 = " +
(this.sigmax * 9.81).toFixed(2) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"σmin = " +
parseFloat(this.sigmin).toFixed(3) +
" t/m2 = " +
(this.sigmin * 9.81).toFixed(2) +
" kN/m2"}
</h5>

```

Ek-73

```

new Chart(this.ctx9, {
plugins: [ChartDataLabels],
type: "scatter",
data: {
datasets: [
{
label: "Alt Nap",

```

```

data: this.xtable,
showLine: true,

pointStyle: "circle",
backgroundColor: "red",
fill: false,
lineTension: 0,
borderColor: "red",
pointRadius: 3,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return (
      parseFloat(value.x).toFixed(2) +
      "," +
      parseFloat(value.y).toFixed(2)
    );
  },

  font: {
    size: 10
  },
  align: "top"
},
{
  label: "Üst Nap",
  data: this.x1table,
  showLine: true,

  pointStyle: "circle",
  backgroundColor: "orange",
  fill: false,
  lineTension: 0,
  borderColor: "orange",
  pointRadius: 3,
  borderWidth: 3,
  datalabels: {

```

```

formatter: function(value, context) {
return (
parseFloat(value.x).toFixed(2) +
", " +
parseFloat(value.y).toFixed(2)
);
},

font: {
size: 10
},
align: "top"
},
{
label: " ",
data: [
{
x:this.p+7,
y:0
},
{
x: this.p,
y: 0
},
{
x: -this.hmax,
y: 0
},
{
x: this.p,
y: this.maksimumsuyukseklik * this.tanjantalfa
},
{
x:this.p+7,
y:((this.maksimumsuyukseklik *
this.tanjantalfa)*(this.hmax+this.p+7)/(this.hmax+this.p))
},

```

```

{
  x: this.p,
  y: this.maksimumsuyukseklik * this.tanjantalfa
},
{
  x: this.p,
  y: 0
},

],
showLine: true,

pointStyle: "circle",
backgroundColor: "orange",
fill: false,
lineTension: 0,
borderColor: "orange",
pointRadius: 3,
borderWidth: 3,
datalabels: {
  formatter: function(value, context) {
    return (
      parseFloat(value.x).toFixed(2) +
      "," +
      parseFloat(value.y).toFixed(2)
    );
  },

  font: {
    size: 10
  },
  align: "top"
}
},
options: {

```

```

scales: {
  xAxes: [
    {
      beginAtZero: true,
      type: "linear",
      position: "bottom",
      scaleLabel: {
        display: true,
        labelString: "X",
        fontSize: 22,
        padding: {
          left: 15,
          top: 0
        }
      }
    },
    {
      beginAtZero: true,
      type: "linear",
      position: "top",
      scaleLabel: {
        display: true,
        labelString: "Y",
        fontSize: 22
      }
    }
  ],
  backgroundColor: "red"
},
response: true,
legend: {
  labels: {
    fontColor: "black",
    fontSize: 22
  }
}
}

```

```
}  
});
```

Ek-74

```
<div className="chart-container">  
<canvas id="myChart9" width="1300px" height="700px" />  
</div>
```

Ek-75

```
this.qmin = (this.minimumdebi - 1.1 * this.qisale123) / this.bnet;  
  
this.qort = (this.ortalamadebi - 1.1 * this.qisale123) / this.bnet;  
this.qmak = (this.maksimumdebi - 1.1 * this.qisale123) / this.bnet;  
let deger = 0.0000001;  
let cevap = 0;  
do {  
cevap =  
deger + (1.1 * Math.pow(this.qmin, 2)) / (19.62 * Math.pow(deger, 2));  
  
deger = deger + 0.000001;  
} while (cevap > this.minimumsuyukseklik);  
this.hlmin = deger;  
deger = 0.000001;  
do {  
cevap =  
deger + (1.1 * Math.pow(this.qort, 2)) / (19.62 * Math.pow(deger, 2));  
  
deger = deger + 0.000001;  
} while (cevap > this.ortalamasuyukseklik);  
this.hlort = deger;  
deger = 0.000001;  
do {  
cevap =  
deger + (1.1 * Math.pow(this.qmak, 2)) / (19.62 * Math.pow(deger, 2));  
  
deger = deger + 0.000001;  
} while (cevap > this.maksimumsuyukseklik);  
this.hlmak = deger;  
console.log(this.hlmak);
```

```

this.h2min =
-this.hlmin / 2 +
Math.sqrt(
Math.pow(this.hlmin, 2) / 4 +
(2 * Math.pow(this.qmin, 2)) / (9.81 * this.hlmin)
);

```

```

this.h2ort =
-this.hlort / 2 +
Math.sqrt(
Math.pow(this.hlort, 2) / 4 +
(2 * Math.pow(this.qort, 2)) / (9.81 * this.hlort)
);

```

```

this.h2mak =
-this.hlmak / 2 +
Math.sqrt(
Math.pow(this.hlmak, 2) / 4 +
(2 * Math.pow(this.qmak, 2)) / (9.81 * this.hlmak)
);

```

Ek-76

```

<h1 className={"text-success"}>{"Bağlama Düşüm Yatağının Tayini"}</h1>
<h5 className={"text-success"}>
{"qmin = " + parseFloat(this.qmin).toFixed(3) + " m3/s"}
</h5>
<h5 className={"text-success"}>
{"qort = " + parseFloat(this.qort).toFixed(3) + " m3/s"}
</h5>
<h5 className={"text-success"}>
{"qmak = " + parseFloat(this.qmak).toFixed(3) + " m3/s"}
</h5>
<h5 className={"text-success"}>
{"hlmin = " + parseFloat(this.hlmin).toFixed(5)}
</h5>
<h5 className={"text-success"}>
{"hlort = " + parseFloat(this.hlort).toFixed(5)}
</h5>

```

```

<h5 className={"text-success"}>
{"h1mak = " + parseFloat(this.h1mak).toFixed(5)}
</h5>
<h5 className={"text-success"}>
{"h2min = " + parseFloat(this.h2min).toFixed(5)}
</h5>
<h5 className={"text-success"}>
{"h2ort = " + parseFloat(this.h2ort).toFixed(5)}
</h5>
<h5 className={"text-success"}>
{"h2mak = " + parseFloat(this.h2mak).toFixed(5)}
</h5>

```

Ek-77

```

<h5 className={"text-success"}>
{"h3maks = " + this.makx1y + "< h2maks = " + this.h2mak.toFixed(5)}
</h5>
<h5 className={"text-success"}>
{"h3ort = " + this.ortx1y + "< h2ort = " + this.h2ort.toFixed(5)}
</h5>
<h5 className={"text-success"}>
{"h3min = " + this.minx1y + "< h2min = " + this.h2min.toFixed(5)}
</h5>

```

Ek-78

```

this.V1maks = this.qmak / this.h1mak;

this.d0 = 0.15 * this.V1maks * Math.sqrt(this.h1mak);

```

Ek-79

```

<Container>

<Row>

<select
value={this.state.kumtipi}
className="browser-default custom-select"
onChange={this.smarthandleforselect("kumtipi")} // handle et
>

<option value="6">6</option>

```

```

<option value="8.5">8.5</option>
<option value="7">7</option>
<option value="6">6</option>
<option value="5">5</option>
<option value="4">4</option>
<option value="3">3</option>
<option value="2">2</option>
<option value="2.5">2.5</option>
</select>
</Row>
</Container>

```

Ek-80

```

this.tanjantalfal = 1 / this.state.kumtipi;

let a = Math.atan(this.tanjantalfal);
this.atantanjantalfal = (a * 180) / Math.PI;
this.Lg = this.state.kumtipi * this.p;

2-3 arası mesafe aşağıdaki kod yardımıyla bulunmuştur .
this.iki_ucmesafe = (this.maksimumsuyukseklık * this.tanjantalfal) /
3;

3-4 arası mesafe aşağıdaki kod yardımıyla bulunmuştur .
this.ucdortmesafe = (5 * this.h2mak) / 3;

```

Ek-81

```

this.p3 =

((this.d0 + this.ucdortmesafe) * this.p) /
(2 * this.d0 + this.iki_ucmesafe + this.ucdortmesafe);
this.p4 =
(this.d0 * this.p) /
(2 * this.d0 + this.iki_ucmesafe + this.ucdortmesafe);

```

Ek-82

```

this.dupmethod = (this.p3 + this.p4) / (2 * (2.4 - 1));

```

Ek-83

```

this.lm = this.d0 + this.iki_ucmesafe + this.ucdortmesafe + this.d0;

```

Ek-84

```
<h5 className={"text-success"}>

{"Vlmak = " + this.Vlmaks.toFixed(3) + " m/s"}

</h5>

<h5 className={"text-success"}>

{"do = " + this.d0.toFixed(3) + " m"}

</h5>

<Container>

<Row>

<select
value={this.state.kumtipi}
className="browser-default custom-select"
onChange={this.smarthandleforselect("kumtipi")} // handle et
>

<option value="6">6</option>
<option value="8.5">8.5</option>
<option value="7">7</option>
<option value="6">6</option>
<option value="5">5</option>
<option value="4">4</option>
<option value="3">3</option>
<option value="2">2</option>
<option value="2.5">2.5</option>
</select>

</Row>

</Container>

<h5 className={"text-warning"}>

{this.kumcesitleri.get(this.state.kumtipi)}

</h5>

<h5 className={"text-success"}>

{"tan $\alpha$  = " + this.tanjantalfal.toFixed(3)}

</h5>

<h5 className={"text-success"}>

{" $\alpha$  = " + this.atantanjantalfal.toFixed(3)}

</h5>

<h5 className={"text-success"}>

{"Lg = " + this.Lg.toFixed(3) + " m"}
```

```

</h5>
<h5 className={"text-success"}>
{"2-3 Arası Mesafe = " + this.iki_ucmesafe.toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"3-4 Arası Mesafe = " + this.ucdortmesafe.toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"P3 = " +
this.p3.toFixed(3) +
" t/m2 = " +
(this.p3 * 9.81).toFixed(3) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"P4 = " +
this.p4.toFixed(3) +
" t/m2 = " +
(this.p4 * 9.81).toFixed(2) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"d = " +
this.dupmethod.toFixed(3) +
" t/m2 =" +
(this.dupmethod * 9.81).toFixed(3) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"d = " + this.dupmethod.toFixed(3) + "> do = " + this.d0.toFixed(2)}
</h5>
<h3 className={"text-success"}>{"Laneye Göre"}</h3>
<h5 className={"text-success"}>
{"Lm = " + this.lm.toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Lg = " +
this.Lg.toFixed(3) +

```

```
"> Lm =" +
this.lm.toFixed(2) +
" Olduğundan Palplanş perde çakılır"}
</h5>
```

Ek-85

```
this.xupmethod =
(this.state.kumtipi * this.p -
(2 * this.d0 + this.ucdortmesafe + this.iki_ucmesafe)) /
2;
```

Ek-86

```
this.lmxupmethod =this.d0 +this.iki_ucmesafe +this.xupmethod
+
this.xupmethod +
this.ucdortmesafe +
this.d0;
```

Ek-87

```
this.p3upmethod = this.tanjantalfal * (this.d0 + this.ucdortmesafe);
this.p4upmethod = this.tanjantalfal * this.d0;
```

Ek-88

```
this.dupmethod = (this.p3upmethod + this.p4upmethod) / (2 * (2.4 -
1));
```

Ek-89

```
<h1 className={"text-success"}>{"Palplanş Perde Çakılması"}</h1>
<h5 className={"text-success"}>
{"x = " + this.xupmethod.toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Lm = " + this.lmxupmethod.toFixed(3) + " m"}
</h5>
<h5 className={"text-success"}>
{"Lm = " +
this.lmxupmethod.toFixed(3) +
"> Lg = " +
```

```

this.Lg.toFixed(3) }
</h5>
<h5 className={"text-success"}>
{"P3 = " +
this.p3upmethod.toFixed(3) +
" t/m2 =" +
(this.p3upmethod * 9.81).toFixed(2) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"P4 = " +
this.p4upmethod.toFixed(3) +
" t/m2 =" +
(this.p4upmethod * 9.81).toFixed(2) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"d = " +
this.dupmethod.toFixed(3) + " t/m2 =" + (this.dupmethod
*9.81).toFixed(2) +
" kN/m2"}
</h5>
<h5 className={"text-success"}>
{"d0 = " +this.d0.toFixed(3) + "> d=" + this.dupmethod.toFixed(3)}
</h5>

```

ÖZGEÇMİŞ

29.04.1989 yılında doğan Betül BAYRAKTAR YÜKSEL ilköğretimini Hatay Cemal Gürsel İlköğretim Okulunda, lise öğrenimini Hatay Naim Atakaş Anadolu Lisesinde, lisans öğreniminide Elazığ Fırat Üniversitesi Mühendislik Fakültesi İnşaat Mühendisliği Bölümünde 2008-2012 yılları arasında tamamlamıştır. Mezun olduktan sonra Fen Bilimleri Enstitüsü İnşaat Mühendisliği Bölümü Hidrolik Ana Bilim dalında yüksek lisans öğrenimine 2016 yılında başlamıştır. 2012-2014 yılları arasında İstanbulda Pehlivanoğlu İnş. Emlak San. ve Tic. Ltd. Şti' de saha mühendisi olarak, 2014-2017 yılları arasında Tus Yapı Denetim Ltd.Şti.'de ise kontrol mühendisi olarak çalışmıştır. 2018 yılında Dsi 8. Bölge Müdrülüğü Proje inşaat Şubesine atanmış olup kısa bir süre çalıştıktan sonra yine aynı yılda Karayolları 8. Bölge Müdrülüğü Ar-Ge Başmühendisliğine atanıp hala daha aynı birimde Yol ve Yapı Malzemeleri Mühendisi olarak çalışmaktadır.