A TOPOLOGICAL SYSTEM FOR CODIFICATION OF RIVER BASINS IN TURKEY

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

BY

İBRAHİM ULUCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

JUNE 2021

Approval of the thesis:

A TOPOLOGICAL SYSTEM FOR CODIFICATION OF RIVER BASINS IN TURKEY

submitted by **İBRAHİM ULUCAN** in partial fulfillment of the requirements for the degree of **Master of Science in Geodetic and Geographic Information Technologies, Middle East Technical University** by,

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

Prof. Dr. Zuhal Akyürek
Head of the Department, **Geodetic and Geographic Information Technologies** _________________

Prof. Dr. Zuhal Akyürek
Supervisor, **Civil Engineering, METU** _________________

Examining Committee Members:

Prof. Dr. İsmail Yücel
Department of Civil Engineering, METU _________________

Prof. Dr. Zuhal Akyürek
Department of Civil Engineering, METU _________________

Prof. Dr. Pınar Karagöz
Department of Computer Engineering, METU _________________

Assoc. Prof. Dr. Yakup Darama
Department of Civil Engineering, Atılım University _________________

Assist. Prof. Dr. Semih Kuter
Department of Forest Engineering, Çankırı Karatekin University _________________

Date: 23.06.2021

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

Name, Last name : İbrahim, Ulucan

Signature:

ABSTRACT

A TOPOLOGICAL SYSTEM FOR CODIFICATION OF RIVER BASINS IN TURKEY

Ulucan, İbrahim
Master of Science, Geodetic and Geographic Information Technologies
Supervisor : Prof. Dr. Zuhal Akyürek

June 2021, 79 pages

Recently, owing to incredible development in Geographic Information Systems (GIS) and software fields, besides effectively using the network and topology instruments, great advances have been made in the field of hydrologic coding through robust interaction among multidisciplinary sciences. If all hydrological properties can be combined in a network that has its own sui generis situation and topological properties, meaningful and holistic analyses can be formed more easily. Additionally, GIS topology is an essential phenomenon for performing different types of network analysis, therefore the topological network model manages spatial relationships by illustrating shapes as the graph of topological instruments (node, edge, etc.) via computer geometry.

In this study, the Pfafstetter method, which stands out with its topological feature and the most common hydrological codification systems all over the world, was coded. It was written in Object-Oriented Program (OOP) language and formatted as a binary tree data structure. The proposed code was applied to three different river basins namely Çakıt Basin and Yeşilırmak Basin which are located in Turkey, and

Thames River Basin which is located in the UK. It is observed that successful results were obtained.

Keywords: Geographic Information Systems (GIS), Algorithm, Topology, River Network, Pfafstetter Coding System

ÖZ

TÜRKİYE'DEKİ NEHİR HAVZALARININ KODLANMASI İÇİN TOPOLOJİK BİR SİSTEM

Ulucan, İbrahim
Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri
Tez Yöneticisi : Prof. Dr. Zuhal Akyürek

Haziran 2021, 79 sayfa

Son yıllarda, Coğrafi Bilgi Sistemleri (CBS) ve yazılım alanlarındaki büyük gelişmeler ile özellikle ağ ve topoloji enstrümanlarının etkin bir şekilde kullanılmasıyla, çok disiplinli bilimler arasındaki güçlü etkileşim ve iletişim sayesinde, hidrolojik kodlama alanında büyük ilerlemeler kaydedilmiştir. Tüm hidrolojik özellikler kendine özgü durumla topolojik özelliklere haiz bir ağ yapısında birleştirilebildiği taktirde anlamlı ve bütüncül analizlere daha rahat kapı aralanabilir. Ağ topolojisinin gücünün CBS ile birleşmesi ile çok zengin ve çeşitli imkanlara olanak sağlanmış, bu bağlamda bilgisayar geometrisi ve grafik veri yapıları aracılığıyla topolojik araçların (düğüm, kenar vb.) uzamsal ilişkileri etkin bir şekilde yönetilebilir hale gelmiştir.

Bu çalışmada, topolojik özelliği ile ön plana çıkan ve dünya genelinde yaygın kullanıma sahip hidrolojik kodlama sistemlerinden biri olan Pfafstetter yöntemi kodlanmıştır. Bu bağlamda, Nesne Yönelimli Programlama (NYP) dili ile ikili ağaç yapısına dayalı bir algoritma ile geliştirilmiştir. Geliştirilen algoritma, Türkiye'de bulunan Çakıt Havzası, Yeşilırmak Havzası ve İngiltere'de yer alan Thames Nehri Havzası üzerinde uygulanmıştır. Başarılı sonuçların elde edildiği görülmüştür.

Anahtar Kelimeler: Coğrafi Bilgi Sistemleri (CBS), Algoritma, Topoloji, Nehir Ağları, Pfafstetter Kodlama Sistemi

To my lovely family

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Prof. Dr. Zuhal Akyürek, for not only her guidance, advice, criticism, encouragements, and insight throughout the research; but also supporting me spiritually my challenges on all steps in terms of GIS software and applications.

I would especially like to state that I used very useful information (Mapping, GIS, DBMS, Topology, DEM, Flow Direction & Accumulation, TIN, Network, Shortest Distance and D8 Algorithms etc.) that I learned in the Introduction to GIS (CE-413) course I took from Prof. Dr. Zuhal Akyürek in all stages of this thesis. I also thank Mr. Kenan Bolat for the useful applications in the laboratory within the scope of this course. If I had not taken the Introduction of GIS, this thesis would not be possible successfully. Also, I would like to special thank Mr. Çağrı Hasan Karaman for providing useful data (DEM, River Network, etc.) from HIDRO-ODTU.

I am completely appreciative of my family. I would like to thank especially my dear wife and two lovely little kids for their endless love and their consideration for my achievements.

This work has the feature of providing the topological capability to the HIDRO-ODTU (Determination of Hydrological Cycle Parameters with a Conceptual Hydrological Model, TUBITAK 115Y041).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
BFS	Breadth First Search
BT	Binary Tree
CCM	Catchment Characterization and Monitoring
DFS	Depth First Search
DEM	Digital Elevation Model
ERICA	European Rivers and Catchments
ECRINS	European Catchments and Rivers Network System
EEA	European Environment Agency
EU	European Union
FA	Flow Accumulation
FD	Flow Direction
FINISH	Finnish Coding System
GDAL	Geospatial Data Abstraction Library
GIS	Geographic Information Systems
GRASS	Geographic Resources Analysis Support System
HUC	Hydrologic Unit Code
LAWA	German Länderarbeitsgemeinschaft Wasser
MCM	Multi-tree Coding Method
NASA	National Aeronautics and Space Administration
NSSS	National Severe Storms Laboratory

OOPL	Object Oriented Programming Language
OSM	Open Street Map
QGIS	Quantum Geographic Information System
RDBMS	Relational Database Management System
REGINE	Norwegian Catchment Coding System
SAGA	System for Automated Geoscientific Analyses
SQL	Structured Query Language
SRTM	Shuttle Radar Topography Mission
UK	United Kingdom
USA	United States of America
USGS	United States Geological Survey
UTM	Universal Transverse Mercator
WISE	Water Information System for Europe
WFD	Water Framework Directive

CHAPTER 1

INTRODUCTION

1.1 River Basin Coding Systems

The river basin is defined as the area of land from which all surface run-off flows through a sequence of streams, rivers and, possibly, lakes into the sea at a single river mouth, estuary, or delta. Coding is important for hydrological and non-hydrological features in river basins. The hydrological stream codification is represented by its own role at the catchment. River basin coding is crucial to manage sustainable water resources effectively for all agencies in the world. In terms of the role of preparing action plans in decision making, the codification of river network helps in defining resource information with different level of river basin. Besides, river basin coding has an important role in order to provide the systematic arrangement of hydrological information (Khan et al., 2001). With the developments in the field of software, river basin delineation and codification can be developed in Geographic Information Systems (GIS) applications. River basin coding has become more meaningful and important as it has gained topological properties. Thus, river basin coding might be used more effectively with two key phenomena: GIS and topology.

Topology in GIS can be explained as the spatial relationships between connecting or adjacent vector features (points, polylines, and polygons) in terms of having a more accurate and precise geometric data structure model. GIS topological network might be carried out to solve various GIS problems such as improving GIS digitizing errors especially during processing vector data. GIS topology is an essential phenomenon for performing different types of network analysis, therefore the topological network model manages spatial relationships by illustrating shapes as the tree of topological instruments (node, edge, etc.) via computer geometry.

Since the river network has a topological feature, it can be demonstrated by which paths of any branch on the river connects to where it follows. In this way, decision-makers can be provided with an idea about which branches in the basin could affect which branches in a possible flood. Likewise, it may be possible to determine the effects of global warming and drought on each element of the river network, which are linked to basins. In summary, the relational integrity between the elements in a river network that lacks topological properties cannot be observed. Hydrological coding methods with topological features provide a significant contribution to hydrological coding as they allow the labeling of the river network with all its elements. Therefore, topological structure for hydrological codification systems is very important in terms of getting relations among every single element of the river network.

Historically, many different hydrological coding systems have been developed since the 19th century. Some of them are used in different geographies around the world, and they have been tried to be classified over the years (Table 1.1).

Table 1.1. Hydrological Coding Systems

River Basin Coding Systems	*Year*	*Regions*
Gravelius (Graveliusi)	1914	Global
Horton	1945	Global
Strahler	1957	Global
Shreve	1966	Global
Scheidegger	1966	Global
Pfafstetter	1989	Global
Hydrologic Unit Code (HUC)	1987	USA
European Rivers and Catchments (ERICA)	1998	EU
Water Information System for Europe (WISE)	2005	EU
European Catchments and Rivers Network System (ECRINS)	2011	EU

Table 1.1. (cont’d)

Länderarbeitsgemeinschaft Wasser (LAWA)	1993	Germany
Norwegian Catchment Coding System (REGINE)	-	Norway
Australian River Assessment System (AusRivAS)	1994	Australian
Finnish Coding System (FINISH)	-	Finland
Binary Tree (BT) Code System	2006	Specific
Multi-tree Coding Method (MCM)	2012	Specific

"Horton-Strahler" is the most widely used stream ordering method among the ones described above. It is heavily used for the classification of the streams and basins in terms of hydrological ranking (de Bartolo et al., 2009). Figure 1.1a shows the method of designating stream orders based on Horton-Strahler methodology (Strahler, 1957). The junction degree (parentheses), the hierarchical order (brackets), and the outlet order maximum (Ω) are specified in terms of Horton-Strahler river network tree data structure on the Figure 1.1b (de Bartolo et al., 2009).

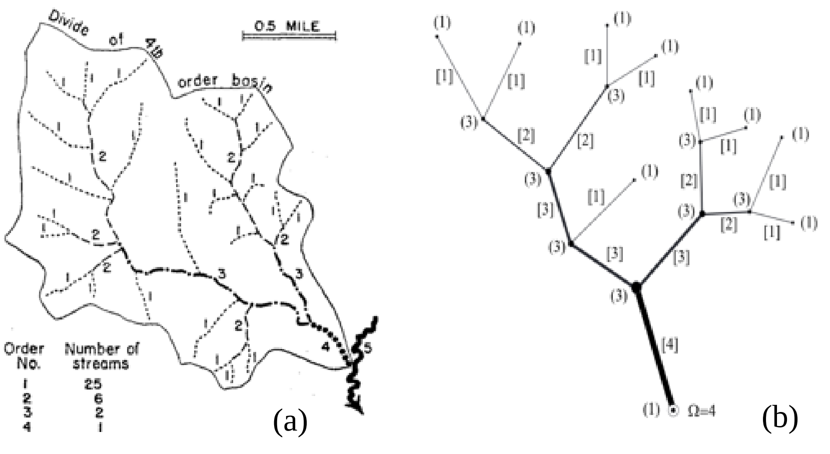


Figure 1.1. Method of Designating Stream Orders (a) (Strahler, 1957); Horton-Strahler River Network Tree (b) (de Bartolo et al., 2009)

"Pfafstetter" system is the most popular among the topological river basin coding methods. The Pfafstetter codification scheme defines a hierarchical decomposition of terrain into small hydrological units as a unique identification label (Verdin & Verdin, 1999). Pfafstetter method has two main basic rules as follows:

- 1st rule: The four tributaries with the highest flow accumulations (largest drainage areas) along the main stem are assigned even digits from downstream to upstream and labeled as “2, 4, 6, 8".
- 2nd rule: Odd digits are assigned to the interbasins along the main stem and labeled as "1, 3, 5, 7, 9".

In the Pfafstetter system, basin 9 is defined as a main stem basin and has a higher flow accumulation value than tributary 8. Figure 1.2 depicts the implementation of the three different levels of the Pfafstetter method for individual watersheds numbering by upstream direction (Furnans & Olivera, 2001)

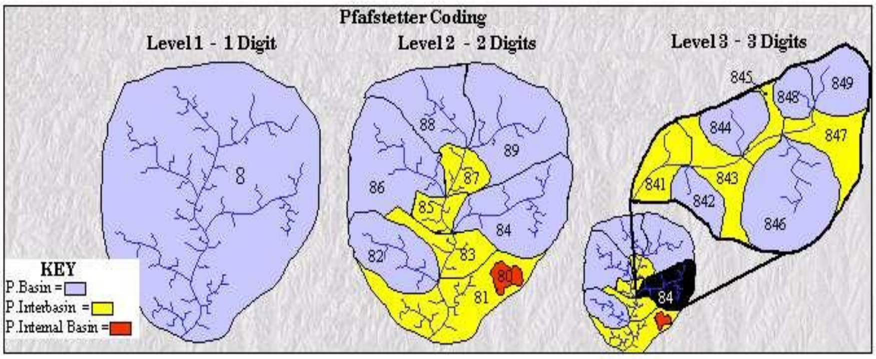


Figure 1.2. Pfafstetter Watersheds Numbering (Furnans & Olivera, 2001)

A large part of the hydrological coding methods used by different countries throughout the world, especially in America and Europe, is mainly based on the Pfafstetter method. Some of the proposed coding system is largely inspired by the work of Otto Pfafstetter with extra information for hydrological codification such as oceans, seas, islands, and lakes. Pfafstetter labeling system can easily provide topological properties like upstream and downstream relation (de Jager & Vogt, 2010).

1.2 Problem Statement

Commercial applications such as Arc Hydro (Fürst & Hörhan, 2009) are generally used for coding Pfafstetter. However, since these commercial applications are concentrated on spherical solutions, they do not always give sufficient results in local basins with special hydrological characteristics. Since the close source codes cannot be accessed entirely, it does not allow to customize the solution. The existing hydrological coding examples, the presence of the detailed and precise Pfafstetter codification examples such as the Amazon River Basin in the US (Kristine L Verdin, 2017) and the Thames River Basin (de Jager & Vogt, 2010) in the EU; however, there are no such similar applications based on the Pfafstetter coding method for Turkey.

1.3 Contribution of the Thesis

A software script and required algorithms are developed as the Object-Oriented Programming Language (OOPL) concept in order to highlight the designated problem effectively. Particularly, followings are the contributions of this study:

- The river network belonging to any basin is classified topologically by adhering to the Pfafstetter method with open-source data and software, so anybody can access the resources freely.
- The basic class and object structures of the software are presented in a solid selection within the discipline of data structures with robust OOP features,
- The Developed software is coded with "Python" which is more flexible and integrated into GIS applications and HIDRO-ODTU.
- The compiled code was applied to 3 different river basins (Çakıt, Yeşilırmak, and Thames) with different geographical features and successful results were obtained.

CHAPTER 2

LITERATURE SURVEY

2.1 The Codification of Hydrological Networks

Many different hydrological coding systems have been developed since the 19th century until today. There are commonly two main types employed for the codification of hydrological networks: hierarchical ranking and topological encoding schemes (Dooley, 2002). On one hand, the ranking schemes are mainly used as a method of interconnected rivers of various sizes and values within a network structure. On the other hand, the main aim of the topological schemes is to build the connection within river networks, broader watershed or river catchment networks, and feature layers in terms of monitoring and analyzing networks. (Britton, 2002).

2.1.1 Ranking Codification Methods

There have been several ranking hydrological ordering codification methods, since 1914. First, Gravelius (Graveliusi) provided basic theory in terms of classification of the river network in basin. Horton put different ordering method, then Strahler modified this method which is the most common application in all over the world. The most well-known ranking coding methods (Dooley, 2002) are provided in Table 2.1.

Table 2.1. Synopsis of Ranking Classification Schemes (Dooley, 2002)

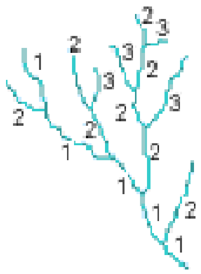
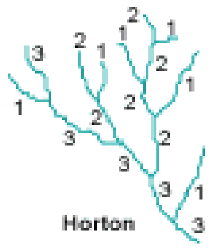
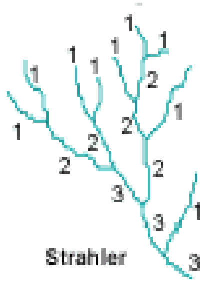
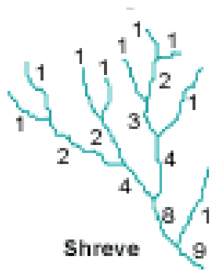
Synopsis of Ranking Classification	*Schemes*
Gravelius (Graveliusi) (1914): The first classification hydrological river basin network method was given by Gravelius by following the sequence logic. As a drainage basin, the largest mainstream was assigned as 1st order stream, and the tributary which directly flows into the 1st rank was assigned as the 2nd order stream. We must face the two biggest challenges to detect the difference between the main stems and tributaries in the river basins. The reason for those differences seldom was insignificant in some river-basin stream networks, even there would be a striking variation among the sizes of drainage basins in terms of ranking the stream network (Gravelius, 1914).	
Horton (1945): Hydrological stream network ranking initiates on the smallest headwater segments being assigned as the 1st, and proceeds in recursively continue with increasing by "1" value whenever two streams of equal order join. The river network is relabeled as the main stem taking the highest order number located in the network, iteratively. The order of the major tributaries in the river network is next reordered recursively, meanwhile, every single junction in which two segments reach the longest or most direct upstream segment is relabeled to the biggest rank encoded for the interbasin (Horton, 1945).	

Table 2.1. (cont'd)

Strahler (1957): Strahler method is mainly depended on a modification of Horton's earlier study; it is also known as "Horton-Strahler" system conceptually it makes sense to present this method first. The Strahler method begins with the smallest headwater of each river segment being assigned as 1st order. The rank increases through the direction of the downstream by 1 value whenever two adjacent streams of equal order join. Furthermore, two 2nd streams order join to form the 3rd order element, except that the stream order does not increase while a higher order stream is attached by a lower order stream (Strahler, 1957)



Shreve (1966): The Shreve magnitude is more easily linked to predicted flood flow than other ordering methods, the method is based on the degree of the stream segment reached on a junction is the sum of the degrees of the two adjacent tributaries. Besides, the confluence of the 1st degree and 3rd degree stream forms the 4th degree of the stream, respectively. Hence, the magnitude of any stream segment equals the number of its magnitude 1 sources. (Smart, 1969)



Scheidegger (1966): Scheidegger stream order method is like Shreve. According to Scheidegger, all segments in the river basin, are twice as the Shreve magnitude with respect to the base 2 logarithm integer (Scheidegger, 1966).

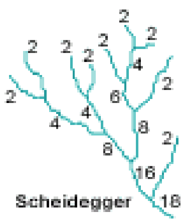


Figure 2.1 depicts the comparison of different stream ordering systems on the same river network (Jasiewicz & Metz, 2011). All results are generally classified as close to each other because the same logic underlies all methods.

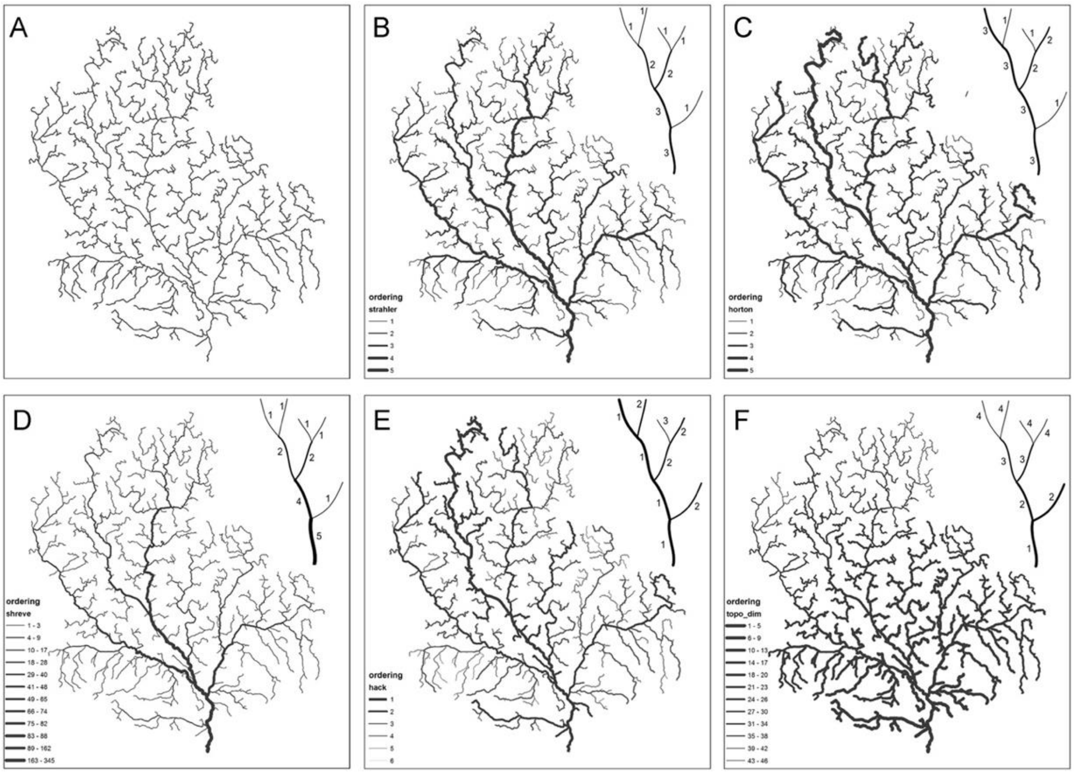


Figure 2.1. Different Stream Ordering Systems: River Network (A), Strahler (B), Horton (C), Shreve (D), Hack (E), and Topological Diameter (F) (Jasiewicz & Metz, 2011)

"Horton-Strahler" is one of the most widely used among the other stream ordering methods. It is commonly used for the classification of the streams and basins in terms of hydrological ranking. The classification rules of Horton-Strahler method are based on delineating the geometrical morphology, here is the summary of the rules:

- 1^{st} rule: The tributary which originates at a source and has no tributary injecting is designated as order "1".
- 2^{nd} rule: The junction of two streams with same order u is designated as order $u+1$.

- 3rd rule: The junction of two streams with unequal order *u* and *v*, where $u<v$, is designated as order *max (u, v)*.

The function of the classification rules formulated as:

$$n = \max(u, v) + \delta_{u,v}, \qquad \text{where } \delta_{u,v} = \begin{cases} 1, & u = v; \\ 0, & u \neq v \end{cases}$$

where *n* is the order of the network formed by two streams with unequal order both *u* and *v* (Zhang et al., 2007).

In theory, encoding river network that is based on any of the above ranking methods is such a basic programming issue, however; in practice, it is hard to find that real river dataset has a verified connectivity between every single of the elements. Generally, river ordering classifications are not commonly executed to watershed networks or models, however; in cases where a topological encoding scheme has been used to link the stream network into the watershed model, the transfer of such attributes can be easily facilitated without the need to resort to any GIS software (Dooley, 2002).

2.1.2 Topological Codification Systems

Within the scope of the network, basic topological relationships (connectivity, adjacency, and enclosure) are of great importance in the context of our analysis by establishing a relationship between the data set like the river network. Due to advances in the processing and analysis of hydrological data with the help of the developing GIS software, important attempts have been made throughout the world in hydrological coding methods today. Some of these topological coding systems will be mentioned here.

The United States of America - Hydrologic Unit Code (HUC): The most well-known topologically river basin labelling hydrological coding system is the HUC created by the United States Geological Survey (USGS). The system builds the labeling encoding hydrological system based on network topology in order to manage drainage basin areas along the continental the United States of America (Seaber et al., 1987). HUC label is consisting of numbers from two to eight digits uniquely depend on the four levels of classification in all over the continent (USGS Water Resources, 2001). Actually, HUC is a kind of labelling system and represents simply surface areas related to regions (2 digits HUCs), subregions (4 digits HUCs), accounting units (6 digits HUCs), and cataloging units (8 digits HUCs) (Verdin & Verdin, 1999). Figure 2.2 shows the US water regions map and Marble Canyon in California Region depends on HUC system (USGS Water Resources, 2001).

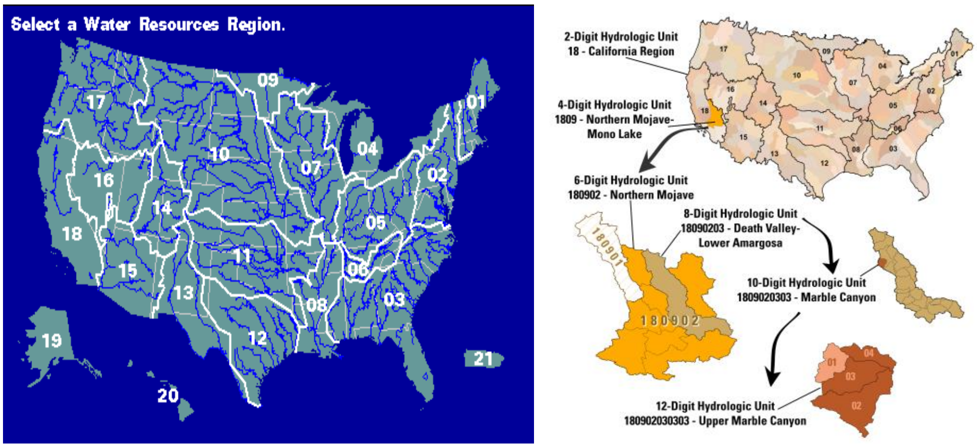


Figure 2.2. Map of Water Resources Regions (USGS Water Resources, 2001)

Moreover, the HUC system is mainly based on classification of four different ranking levels. The summary of the Hierarchy of hydrologic units is shown in Table 2.2 (USGS Water Resources, 2001).

Table 2.2. HUC Classification Order (USGS Water Resources, 2001)

Level	*HUC Classification Order*
1st level 2-digit *RR#21*	The 1st level divides the continent into 21 major geographic region (RR) which contain the drainage area of a series of rivers thorough the USA.
2nd level 4-digit *SS#221*	The 2nd level separates the 21 regions into 221 subregions (SS). A subregion consists of the area drained by a basin system.
3rd level 6-digit *AA#378*	In the 3rd level, there are many of the subregions accounting units (AA). These 378 hydrologic accounting units are nested within the subregions.
4th level 8-digit *CC#2264*	The 4th level is a cataloging unit (CC) which is a geographic area representing part of the basin.

European Union - European Rivers and Catchments (ERICA) (1998): EU catchment coding system mainly depends on European Rivers and Catchments (ERICA), European Catchments and Rivers Network System (ECRINS), Catchment Characterization and Monitoring (CCM) and Water Information System for Europe (WISE) or Water Framework Directive (WFD) (Flavin et al., 1998; EEA, 2008; EU Project, 2009; European Commission, 2010; European Communities, 2003; European Environment Agency, 2012; Globevnik et al., 2010; Vogt al., 2007). Major Rivers and River Basins of Europe are shown in Figure 2.3 (Vogt et al., 2007).

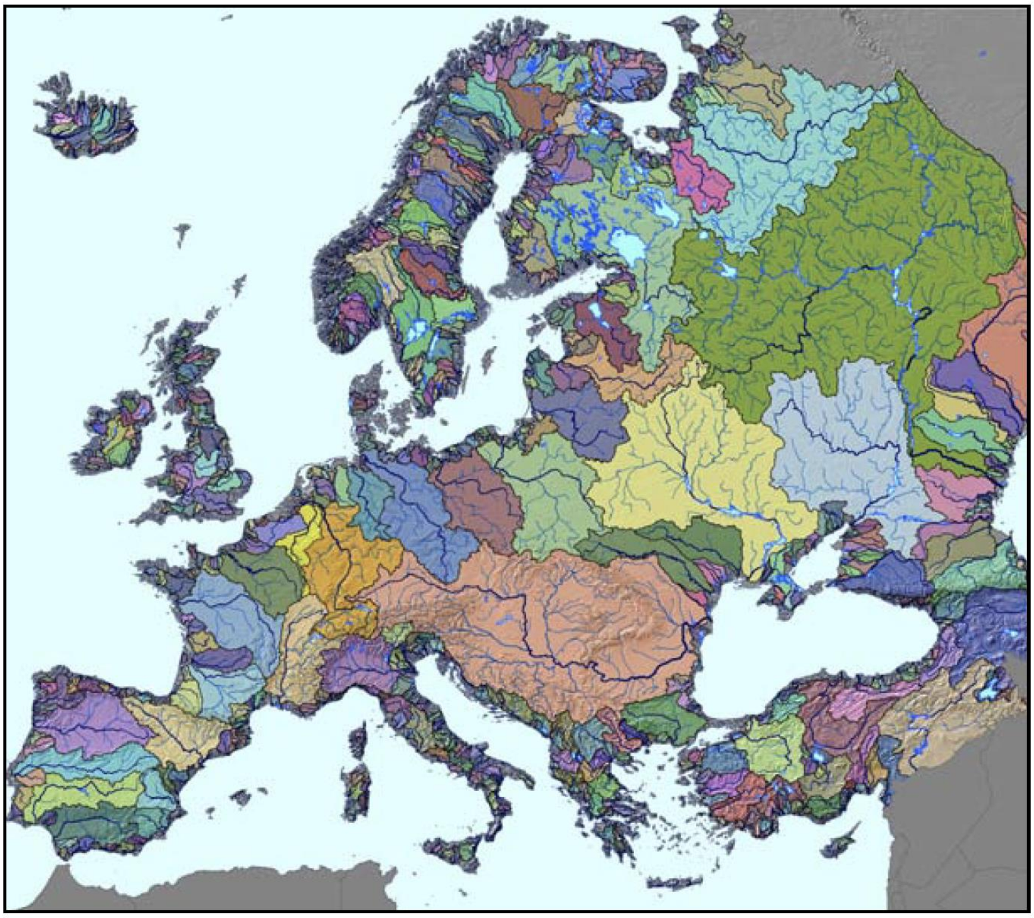


Figure 2.3. Major Rivers and River Basins of Europe (Vogt al., 2007)

The ERICA Coding System (ERICA-CS) provides explicit information about areas draining to a given sea/ocean or coastal stretch in all over the European Continent. According to the ERICA, the sample code is explained as follows: "MM BBB N1 N2 N3 N4 A" (Flavin et al., 1998).

- "MM" : 2-digit marine code
- "BBB" : 3-digit marine border code
- "N1-N2-N3-N4" : 2-digit nested catchment codes
- "A" : 1 character code driving from area

Figure 2.4 depicts the example of the development and demonstration of a structured hydrological feature coding system that is heavily based on Pfafstetter method for Europe (de Jager & Vogt, 2010).

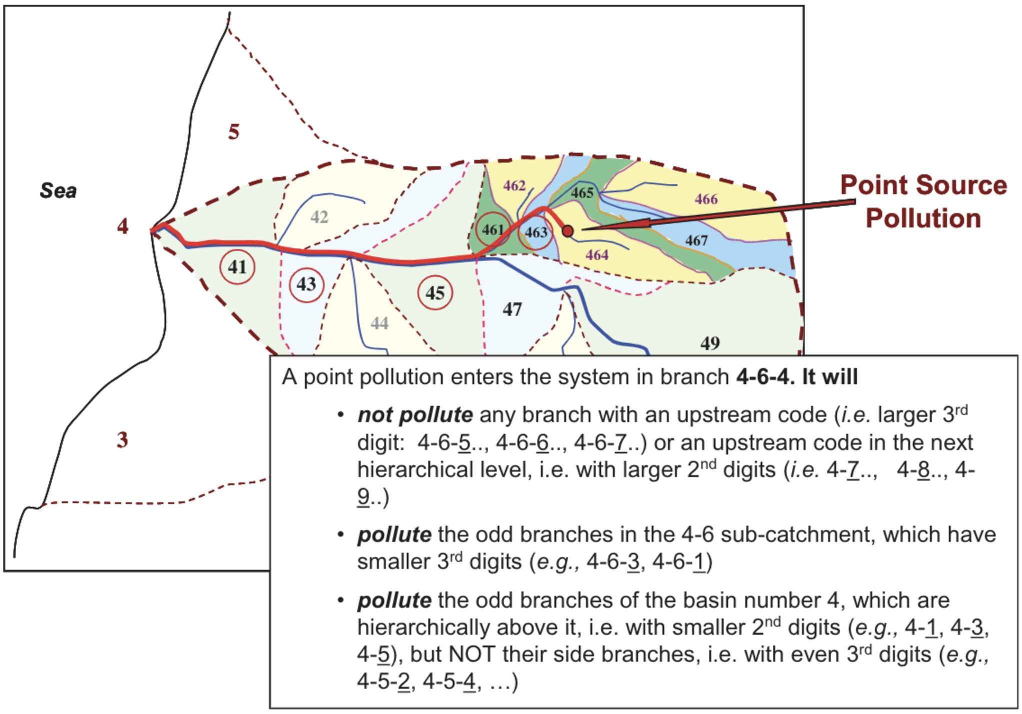


Figure 2.4. Pfafstetter Codes for EU (de Jager & Vogt, 2010)

European Catchments and Rivers Network System (ECRINS) (2011) is a very detailed coding format. It is the newest and most used system in Europe by 2021. Using the CCM2 (Catchment Characterization and Modeling) GIS database, the data can be downloaded from internet. The entire European basin database can be downloaded and used free of charge from the website. Likewise, it can use different databases. CCM database has been expanding with each passing day and also was included Turkey in 2009. Large lakes such as Lake Van and Lake Tuz are named completely in this system. European Environment Agency (EEA) left ERICA and started to use ECRINS. All major basins in Europe have been defined and coded in a long list. Nested coding is not allowed. It has a fixed coding system, and its format is quite long (Globevnik et al., 2010).

Water Information System for Europe (WISE) is a reference system for the ECRINS project. There are also places where WFD (Water Framework Directive) is written as coding. Object type codes are also valid in this system. Lake, river, etc. objects can be easily coded with it. Coding is much simpler and clearer than ECRINS. There is also an expression such as a country code in this code. There is a serious problem in defining closed basins (Globevnik et al., 2010).

Catchment Characterization and Monitoring (CCM) is a project run by the European Commission, Joint Research Centre, Institute for Environment and Sustainability within the scope of providing database and structured hydrological feature data set on river and catchment codification systems for Europe. Using a connected data set in which every river basin discharging to sea, one can achieve this ordering automatically for hydrological feature coding system (Globevnik et al., 2010).

The Norwegian Register of Catchment Areas (REGINE) is part of the Norwegian Water Information System (NORWIS), which is developed by the Norwegian Water Resources and Energy Administration (NVE) (Flavin et al., 1998).

German Catchment Coding System (LAWA) is a kind of hydrological codification method of catchment coding. It has been established by a committee of the Länderarbeitsgemeinschaft Wasser (LAWA) (Flavin et al., 1998).

The Finnish codification system (FINISH) in Finland, it is a structured hierarchical approach and successfully applied to many real practices; however, the ordering method applying with 18 characters to code river networks results in too complicated in terms of labeling structure (Britton, 2002). Therefore, it makes hard to code a new interbasin or tributary which connects and joins together into the networks because the coded network must be re-coded again and again (Britton, 2002).

Australian River Assessment System (AusRivAS) is a type of hydrological codification system that is used for a variety of purposes, water resource condition auditing, and demonstrating the natural resource and river network situation (Flavin et al., 1998). Figure 2.5 illustrates an example of Australian River Assessment System (AusRivAS) (Gray, 2004).

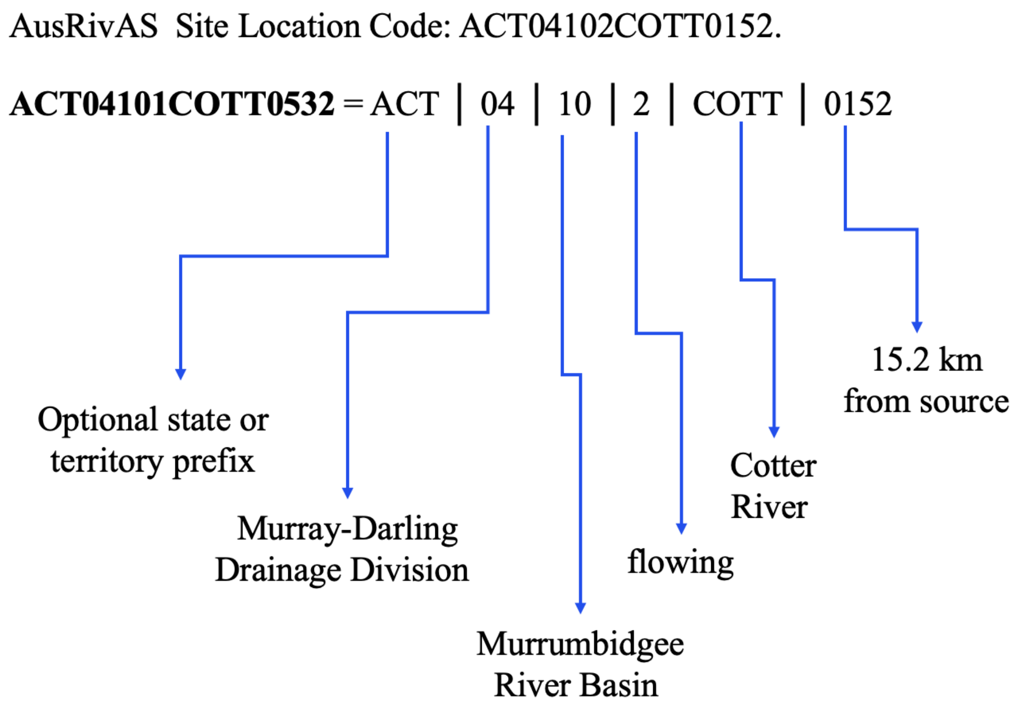


Figure 2.5. Australian River Assessment System (AusRivAS) (Gray, 2004)

Binary Tree (BT) Codification: It is a type of ordering method where the classification on the network is based on tree data structure. It was thought to be more appropriate to the parallel computation for a distributed hydrologic model to simulate rain-runoff processes in a large-scale river basin (Zhang et al., 2007). Basically, the codification of the stream network beginning with the downstream of 0 is put at the exit of the main course of the stream. Binary digit 0 or 1 (basic logic True or False) coding is done along the main course from downstream to upstream. If it is on the main branch, the value of binary digit 0 is added to the right of the previous code, and 1 if it is the branch, and the coding process continues until all branches in the coding stream network are finished recursively (Figure 2.6.) (Zhang et al., 2007). The drainage network is presented as a binary tree which is shown in Figure 2.7 (Zhang et al., 2007).

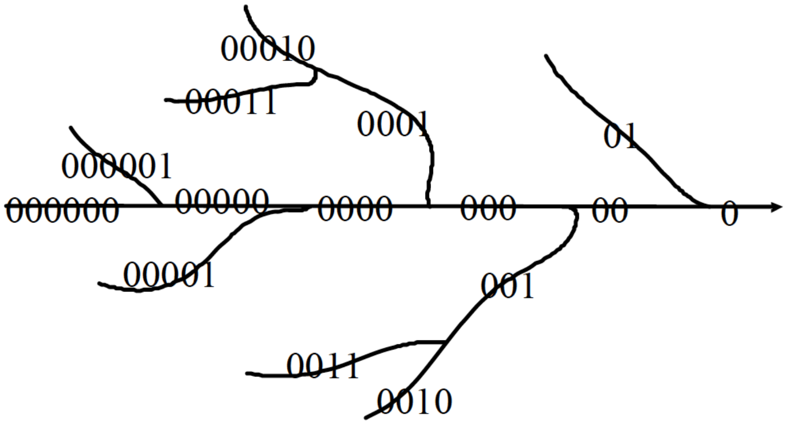


Figure 2.6. Binary String Codification of a Drainage Network (Zhang et al., 2007)

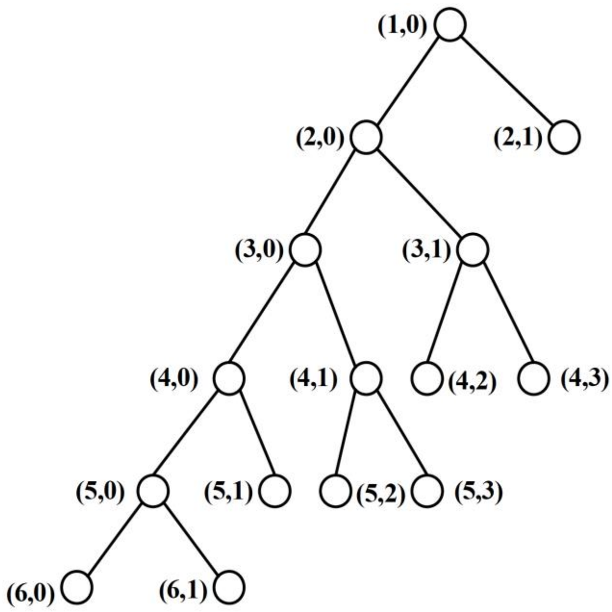


Figure 2.7. The BT Representation of a Drainage Network (Zhang et al., 2007)

Multi-tree Coding Method (MCM): MCM is a coding system and based on tree data structure like Binary Tree Codification method. Additionally, it is possible to build the special query with the help of SQL (Structured Query Language). The method is providing a river reach, or a sub-basin consists of three sections: Layer Number, Node Number, and P Node Number. Figure 2.8 explains a constructed drainage network and MCM of all sub-basins: 1st code component is Layer Number, 2nd is Node Number, and 3rd is P Node Number, respectively (Wang et al., 2013).

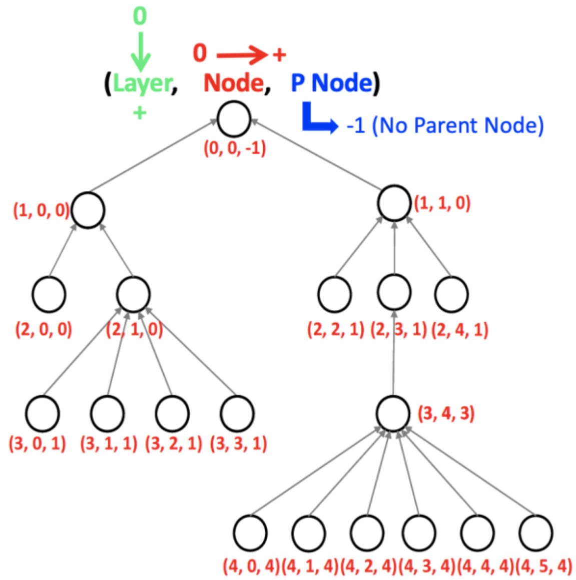


Figure 2.8. The Drainage Network with MCM (Wang et al., 2013)

Figure 2.9 depicts the data structure of the basin with MCM coding. For computer memory, the square demonstrates a type of data collection such as an array and list. The element stores the physical address of one sub-basin node. The size of array rows is equal to the size of layers of the tree, and the size of the columns in each array row is equal to the number of nodes in each multi-tree layer. (Wang et al., 2013)

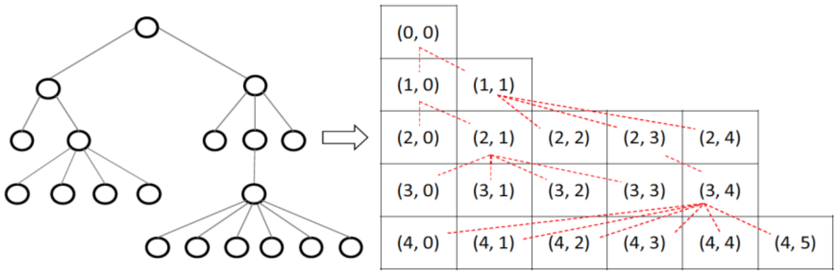


Figure 2.9. The Multi-Tree Drainage Network (Wang et al., 2013)

Table 2.3 demonstrates the summary of common properties among different topological encoding schemes being employed in the European Union (Britton, 2002).

Table 2.3. The Comparison of Topological Encoding Systems (Britton, 2002)

The Comparison of Topological Encoding Systems								
Coding System	Topology Captured	Structured Nested	Easy to Understand	Extra Digit(s)	Hydrological Catchment	#Nested Sub-Catchment	#Nested Tributary	Indicates Catchment
Pfafstetter (USA)	+	+	+	+	+	9	4	-
ERICA (EU)	+	+	+	+	+	99	49	+
LAWA (Germany)	+	+	+	+	+	9	4	-
REGINE (Norway)	+	+	-	-	-	33	-	-
FINISH (Finland)	+	+	-	+	+	-	100	+

Table 2.4 depicts the summary of several basin and stream gauge codification schemes (Verdin & Verdin, 1999).

Table 2.4. Summary of Several Basin Codification Schemes (Verdin & Verdin, 1999)

Organization/system	*Country*	*Basis*	*Extends*	*No. digits*
USGS/HUCS	USA	Basin	National	8
USGS/NWIS	USA	Gauge	National	8
ORSTOM	France	Gauge	Continental	9
DNAEE	Brazil	Gauge	National	8
GRDC	United Nations	Gauge	Global	7

As a result, a large part of the hydrological coding methods used by different countries throughout the world, especially in America and Europe, is mainly based on the Pfafstetter method. Besides some of the proposed coding system is largely inspired by the work of Otto Pfafstetter with extra information for hydrological codification such as oceans, seas, islands, and lakes (de Jager & Vogt, 2010).

2.2 Pfafstetter Codification Method

The Pfafstetter river basin codification method was offered as a global labeling system based on stream network topology. In 1989, the idea of river basin labeling system was developed by a Brazilian engineer Otto Pfafstetter, who was serviced as engineer at the Departamento Nacional de Obras de Saneamento (DNOS) in Brazil (Verdin & Verdin, 1999).

The Pfafstetter method heavily depends on the topology of the drainage network and the size of the surface area drained. It provides identification numbers recursively to the degree of the smallest subbasins extractable from a Digital Elevation Model (DEM) (Britton, 2002).

Actually, Pfafstetter codification method is a kind of numbering scheme in order to use labeling drainage networks in a river basin domain (Verdin & Verdin, 1999). There are two key phenomena in the rules of Pfafstetter codification: "basin" (drainage area towards to a reach of the main stem) and "interbasin" (between the drainage areas that drained by any tributaries). An idealized river basin showing subdivision into coded basins and interbasins is shown in Figure 2.10. If closed basins (internal basins) are encountered, the largest one is assigned the number zero. (Verdin & Verdin, 1999).

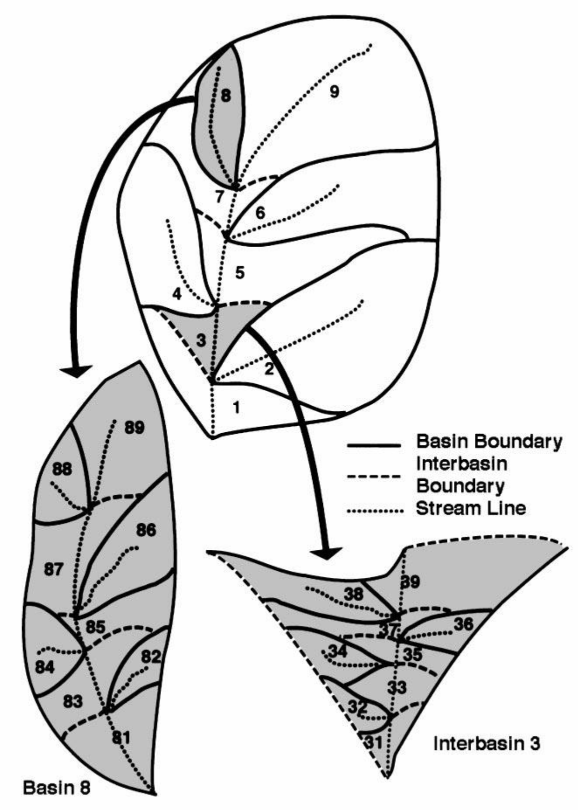


Figure 2.10. Sample subdivisions of a basin and an interbasin obtained by applying the rules of Pfafstetter codification (Verdin & Verdin, 1999)

The implication of the Pfafstetter method stands out among other catchment coding systems in terms of simple, easy-to-understand, and global applicability (Stein, 2018). Figure 2.11 depicts Pfafstetter decomposition of a basin: The four largest tributary catchments are coded with even digits "2, 4, 6, and 8" in the order from basin outlet to the upstream of the basin. The intervening areas draining to the main stem are coded with odd digits 1 to 9. Subbasins are successively subdivided using the same system as long as four tributaries remain (Stein, 2018).

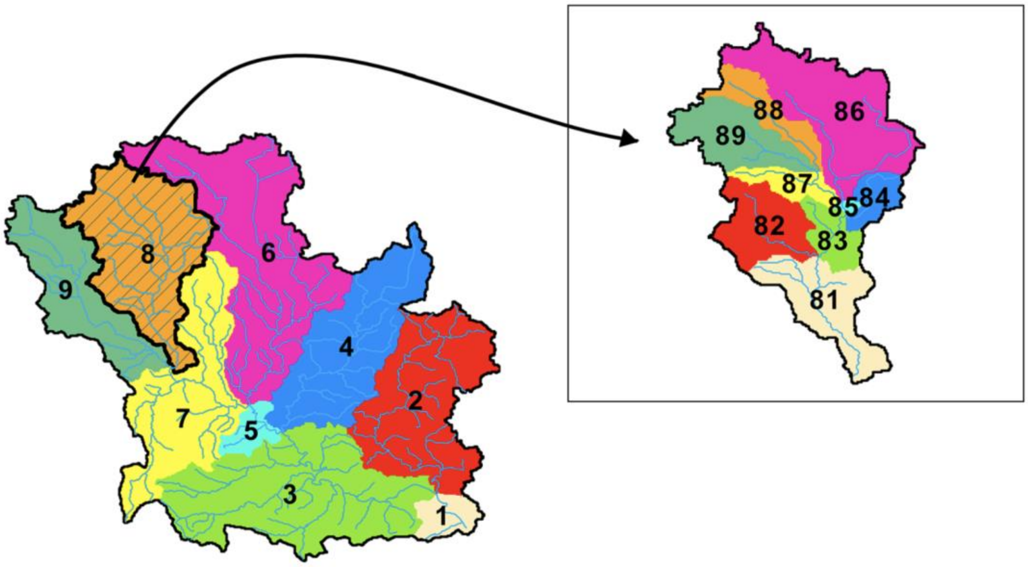


Figure 2.11. Pfafstetter Decomposition of River Basin Network (Stein, 2018)

The logic of Pfafstetter methodology says that main river basins are split into three different parts which are basins, interbasins, and internal basins. A basin represents a closed zone. It should not take drainage from any other drainage region. An interbasin shows watershed that receives flow from upstream watersheds. An internal basin is a kind of drainage area. It should not provide flow to either another watershed or to a waterbody (such as an ocean or lake).(Furnans & Olivera, 2001).

The basic labelling algorithm of Pfafstetter is based on unique identification numbers with respect to river basin network (basin, interbasin, internal basin), and the process is executed recursively. This process continues until there are no unlabeled branches for each level. Meanwhile, the direction of the labelling is always from downstream or outlet (seaside) to upstream or source vice versa flow accumulation. Here is the basic Pfafstetter labelling steps:

- First of all, determining the path of main stem and tributary is very important in order to find the right labelling on right way otherwise it is hard to know whether it is the main stem or tributary. Before igniting the labelling process, the best practice is double check with lengths and depth values for each level on the river network. Horton-Strahler method might be helpful to find the longest or deepest path with respect to high level order ranking.
- Secondly, select the four tributaries according to the largest drainage areas along the main stem of the river. The four tributaries are marked even digits through the direction (from the watershed outlet to the source) from downstream to upstream. The watersheds containing these four tributaries are basins. Assign each basin the code "2," "4," "6," or "8", the most downstream basin gets the "2", the next most downstream basin gets the "4" and so on (Furnans & Olivera, 2001).
- Thirdly, assign odd digits to the interbasins along the main stem of the river network. Marking each interbasin the code "1," "3," "5," "7," or "9" in the upstream direction, the most downstream interbasin gets "1," the next most downstream interbasin gets "3," etc. Therefore, interbasins are the watersheds between basins. Due to the power of network topology, in the Pfafstetter system, basin 9 is defined as a main stem basin and has a higher flow accumulation value than tributary 8 (The National Severe Storms Laboratory (NSSS), 2004). Figure 2.12 depicts the steps of Pfafstetter labelling (de Bartolo et al., 2009).

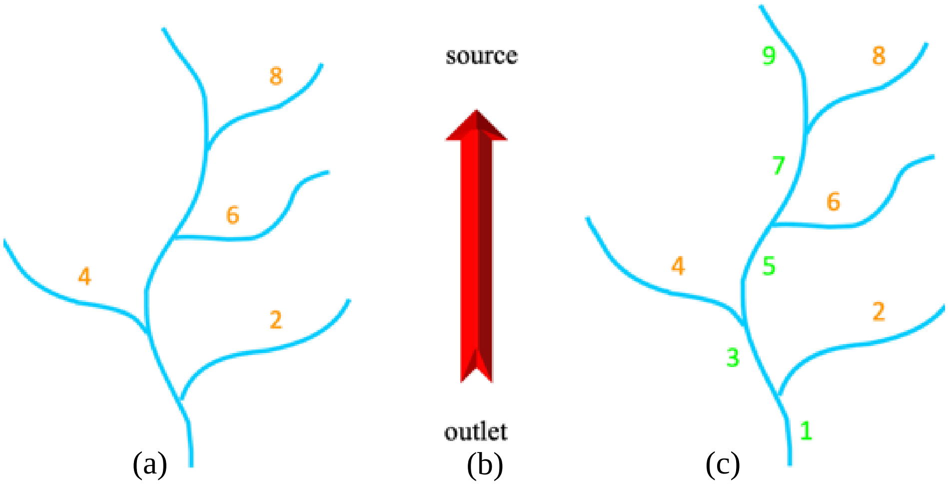


Figure 2.12. The Tributary Basin Pfafstetter Number (2, 4, 6, 8) (a); The Pfafstetter Labelling Direction (b); The Main Stem Interbasin Pfafstetter Number (1, 3, 5, 7, 9) (c) (NSSS, 2004)

Even if there are less than four tributaries in the system of Pfafstetter labelling, the number of basins "9" is always the main stem of the river network (Figure 2.13) (NSSS), 2004). Additionally, if an area contains internal basins, the largest internal basin is assigned the code "0" and all other internal basins are incorporated into a surrounding basin or interbasin (de Bartolo et al., 2009).

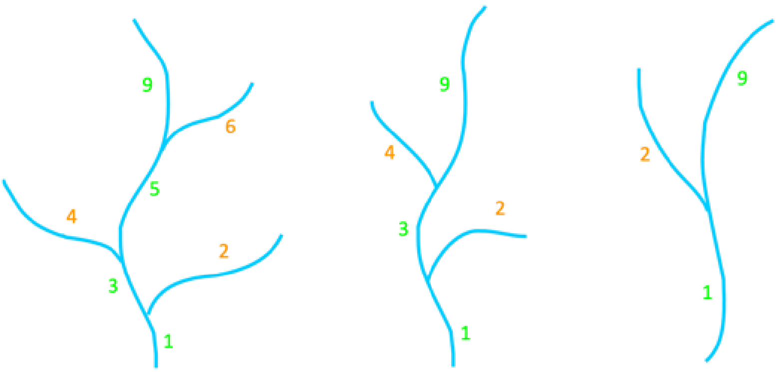


Figure 2.13. Pfafstetter Codification (less than four basins or interbasins) (NSSS, 2004)

The Subdivision of Basins and Interbasins of Pfafstetter Codification: These assigned codes are then appended on to the end of the Pfafstetter code of the next lowest level, and this rule will be continued recursively on every single level on the river network. Furthermore, in assigning level 3 codes, each level 2 watershed is divided into at most ten sub-watersheds, and these sub-watersheds all have the level 2 code XY. The level 3 codes of these sub-watersheds become XY0, XY1, XY2, etc. (de Bartolo et al., 2009). Figure 2.14 shows the subdivision of interbasins and basins labelling.

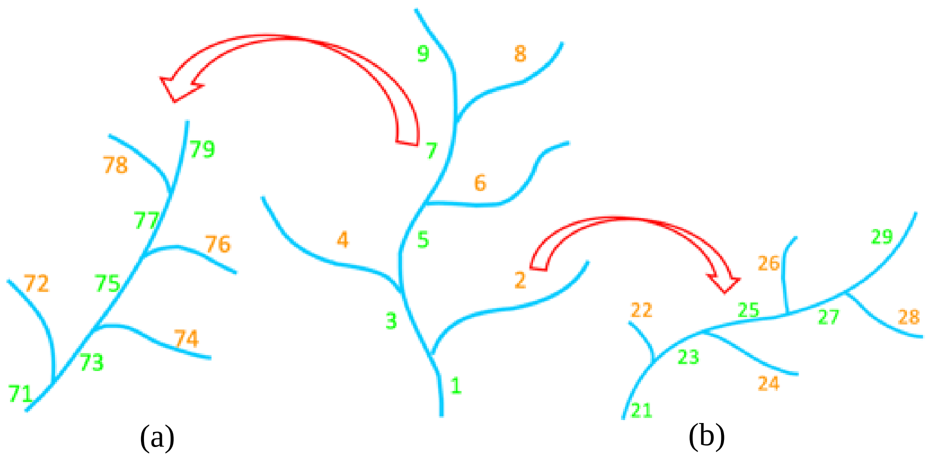


Figure 2.14. Pfafstetter Codification Subdivision of Interbasins (a); Subdivision of (Tributaries) Basins (b) (NSSS, 2004)

Within the scope of the determination of coastal region, similarly sea region coding, the focal point of the Pfafstetter codification is that four major catchments within each sea region are assigned by even numbers (2, 4, 6, 8) in a clockwise direction and a coastal region between these major catchments is assigned with odd numbers (1, 3, 5, 7, 9). (Globevnik et al., 2010).

The disadvantage of Pfafstetter coding is being limited to the maximum 9 basins, if there are more than 4 main branches, the remaining branches should be ignored. On the other hand, the method of Pfafstetter provides the robust topological capability through the river network. There are also modified uses especially in the EU (Verdin & Verdin, 1999). In other words, this method can be used by modifying it as desired. All over the world, most of the hydrological coding methods are heavily based on the Pfafstetter methodology.

Topological Characteristics of the Pfafstetter Codification: Pfafstetter digits indicate relative upstream and downstream positions at the river network. Therefore, these numbers can be used for providing more detail and clearer picture in terms of basin topological analyses. The more we use the combination of Digital Elevation Models (DEM) and subsequent exploitation with both Database Management System (DBMS) and Geographic Information Systems (GIS) software, the better and more robust network topology can be obtained (Furnans & Olivera, 2001). Assuming that there is no topological feature, Pfafstetter or similar hydrological coding methods will lack a relational feature for any network structure. In fact, the application of Pfafstetter coding is due to the strength of topology in the river network.

In order to manage topology, we need to know relationship among the input network data. The Pfafstetter ID number provide sufficient information about the correlation between upstream and downstream. Figure 2.15 demonstrates that the topological relations in terms of upstream and downstream are relative to location. The area of X is at the downstream of the area of Y; therefore, one can be said that the location of Y must be the upstream of X respectively (Furnans & Olivera, 2001).

Figure 2.15. Topological Relations between Upstream and Downstream (Furnans & Olivera, 2001)

The topological aspect of downstream navigation, 5th interbasin is the first target of the watershed, 3rd interbasin is the downstream of 5th interbasin (Figure 2.16) (Furnans & Olivera, 2001). Similarly, 3rd interbasin becomes the target watershed, where 1st interbasin is downstream of 3rd interbasin. At last, there is no watershed in the downstream of 1st interbasin, the navigation is finished, and all previous downstream areas have the value of "trace function" equals to "2" (Furnans & Olivera, 2001).

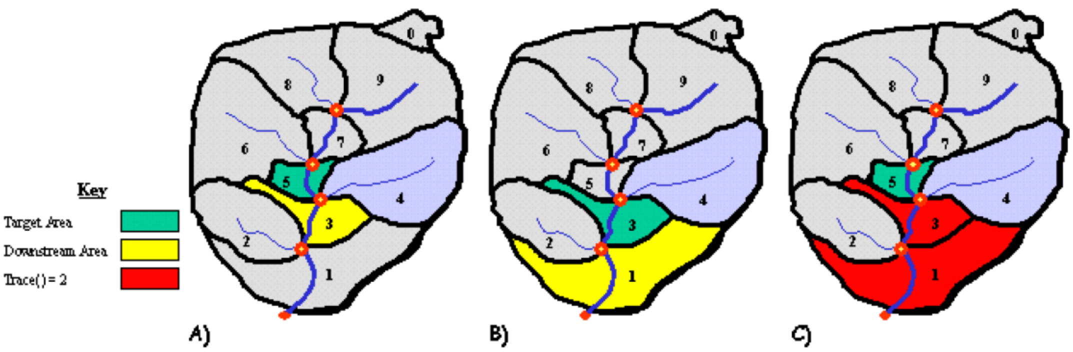


Figure 2.16. Downstream Topological Navigation (Furnans & Olivera, 2001)

Pfafstetter codification scheme defines a hierarchical decomposition of a terrain into small hydrological units as a unique identification label (Verdin & Verdin, 1999). Pfafstetter method can easily apply on the river network by using unique numbers. These numbers represent topological properties in terms of upstream and downstream adjacency and neighborhood (Karaman et al., 2019).

Some example applications are selected from the famous river networks in the world. The Amazon River Basin in South America and the Thames River Catchment in the UK are presented. Figure 2.17 explains that Pfafstetter subdivision of the Amazon River Basin. Figure 2.17 explains the Pfafstetter subdivision of the Amazon River Basin as follows:

- Firstly, in Pfafstetter Level-1, Amazon River Basin is assigned with the Pfafstetter code 54. The digit of 5 is the continental number (South America), and 4 is the 1st level of the Pfafstetter label for the Amazon Basin (A).
- Secondly, in Pfafstetter Level-2, Amazon River Basin is divided into subdivisions of the basins to identify the largest four tributaries. Amazon River is labeled with even numbers (2, 4, 6, 8). The headwaters of the subbasin are 9 (source), and the interbasins are assigned with odd digits (1, 3, 5, 7) (B).
- Thirdly, in Pfafstetter Level-3, The subdivisions of the Amazon River Basin are assigned with the even (four largest tributaries) and odd digits (interbasins) recursively like level-2. For each iteration, 9 must be labeled with source (C).
- Lastly, in Pfafstetter Level-4, the four biggest tributaries are labeled with even numbers (2, 4, 6, 8) again. As always 9 is the headwaters for the basin of 5422. The interbasins are assigned with odd numbers (1, 3, 5, 7) (D) (Kristine L Verdin, 2017).

Similarly, Figure 2.18 depicts different level of Pfafstetter codification for the Thames River Basin (de Jager & Vogt, 2010).

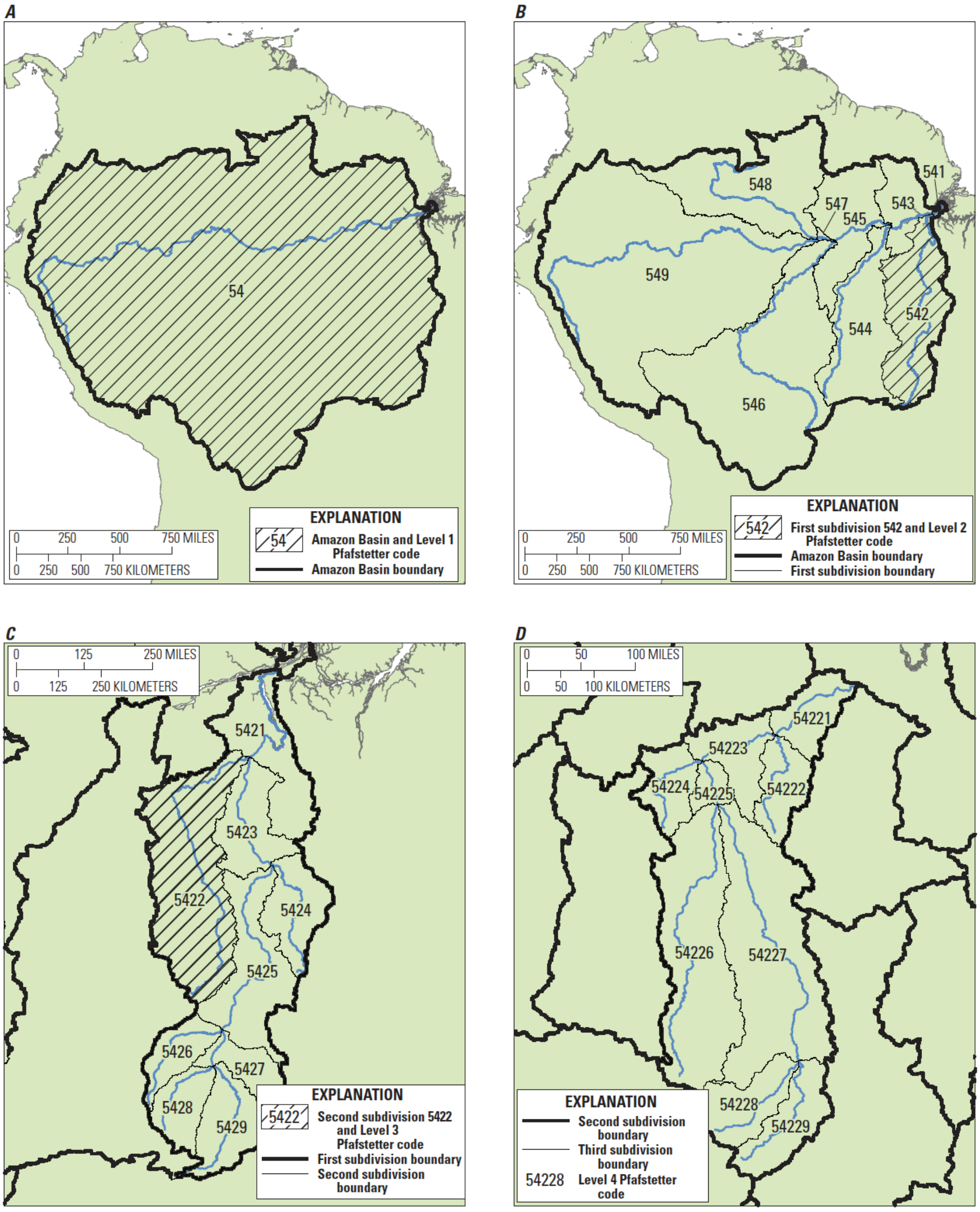


Figure 2.17. The Iterative Pfafstetter Codification of the Subdivision of the Amazon River Basin (Kristine L Verdin, 2017)

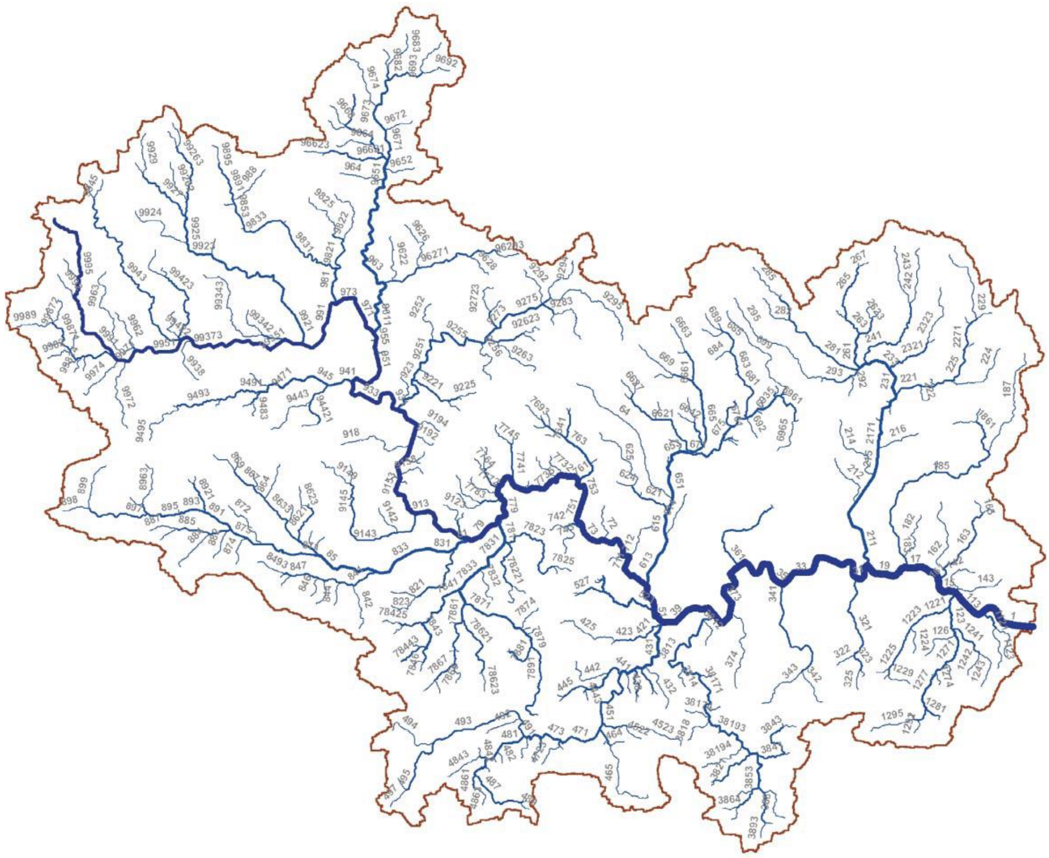


Figure 2.18. The Full Pfafstetter Codification for the River Reaches of the Thames River Catchment (de Jager & Vogt, 2010)

An example of a Turkish river basin coding depends on the combination of European Catchments and Rivers Network System (ECRINS) and the Pfafstetter codification method (Darama & Seyrek, 2016). Figure 2.19 explains the developed coding system.

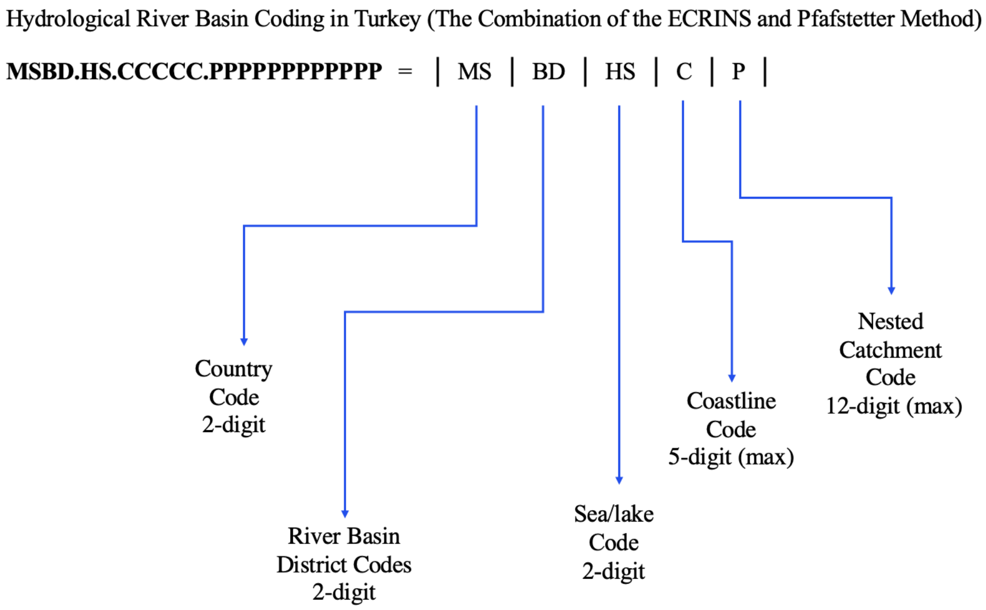


Figure 2.19. Hydrological River Basin Coding in Turkey (The Combination of the ECRINS and Pfafstetter Method) (Darama & Seyrek, 2016)

The application of Pfafstetter method to Yeşilırmak River Basin, which is one of the major river basins and located in the coastal region of the Black Sea in the Northern sector of Turkey (Darama & Seyrek, 2016). The main drainage area of the Yeşilırmak River is assigned as "TR14.M5.2.", basin is separated into nine sub-basins, and all sub-basins are coded with respect to the Pfafstetter method: the main sub-basins (on the main stem) are coded as even numbers "2, 4, 6, 8"; the interbasins (on the tributaries) are coded as odd numbers "1, 3, 5, 7, 9" as it is shown in Figure 2.20 (Darama & Seyrek, 2016).

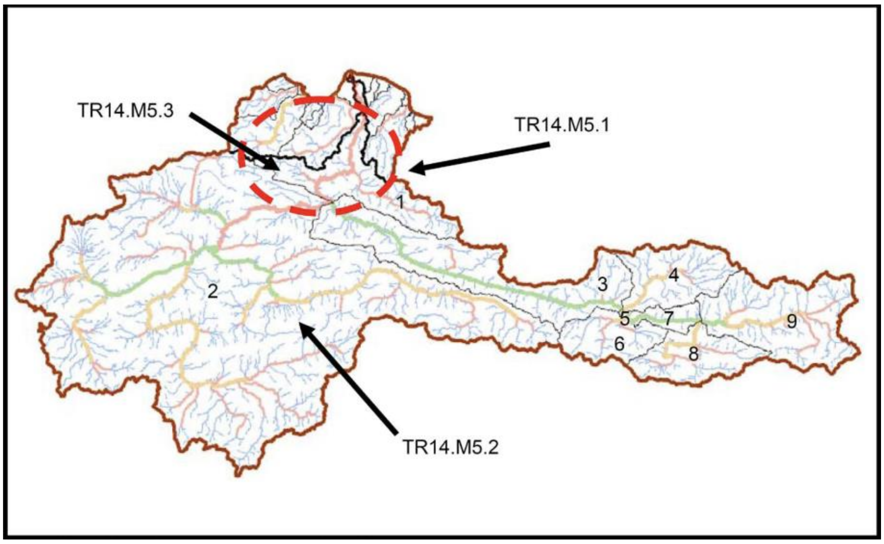


Figure 2.20. Main Sub-areas of the Yeşilırmak River Basin with respect to Pfafstetter Codification (Darama & Seyrek, 2016)

According to Pfafstetter system, the coding is a bit different for the streams discharging into the sea (Figure 2.21a) or discharging to a natural lake, the coding of the drainage areas is done in the "clockwise direction" (Figure 2.21b) (Darama & Seyrek, 2016).

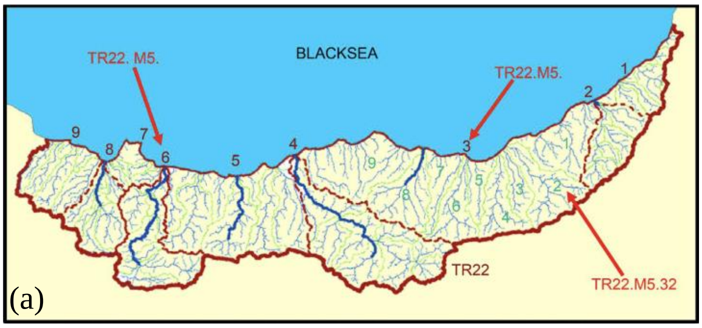


Figure 2.21. The Codification of Coastal Zones (Eastern Black Sea) (a); the Closed Basin Coding (b) with respect to Pfafstetter (Darama & Seyrek, 2016)

As it can be seen from the existing hydrological coding examples, there is no detailed and precise coding example in Turkey like the ones in the US (Kristine L Verdin, 2017) and the EU (de Jager & Vogt, 2010).

CHAPTER 3

METHODOLOGY

3.1 Methodology

The network data structure is one of the most important phenomena that is mainly based on two kinds of mathematical disciplines: network and topology (how the segments in the network are connected to each other) (Demšar et al., 2008). The power of the topological relationships led to revolutionary advances in GIS, especially at network applications like river basin coding (Curtin, 2007). Basically, the methodology consists of four key elements: GIS, data structure, network, and topology. In the field of hydrology, the Pfafstetter labeling system stands out as the most well-known and popular coding method all over the world. The river network data structure heavily depends on the binary tree data structure. What is more, "trees" are graphs those have the following attributes (Celko, 2012):

- A tree is a connected graph that has no cycles. A connected graph is one in which there is a path between any two nodes. No node sits by itself, disconnected from the rest of the graph.
- Every node is the root of a subtree, and the most trivial case is a subtree of only one node.
- Every two nodes in the tree are connected by one and only one path. In a tree, the number of edges is one less than the number of nodes.

Within the scope of the data structure of the river network, "binary tree" is a kind of tree, and it has the following characteristics (Goodrich et al., 2013):

- Every node has at most two children. Each child node is marked as either left child or right child.
- A left child precedes a right child in the order of children of a node.

The subtree rooted at either left or right child of an internal node is named as a left or right subtree, respectively (Goodrich et al., 2013). Furthermore, it might be possible to put spatial relationships on this data model as shown in Figure 3.1 (Demir & Szczepanek, 2017), and in Figure 3.2 (Arge et al., 2006).

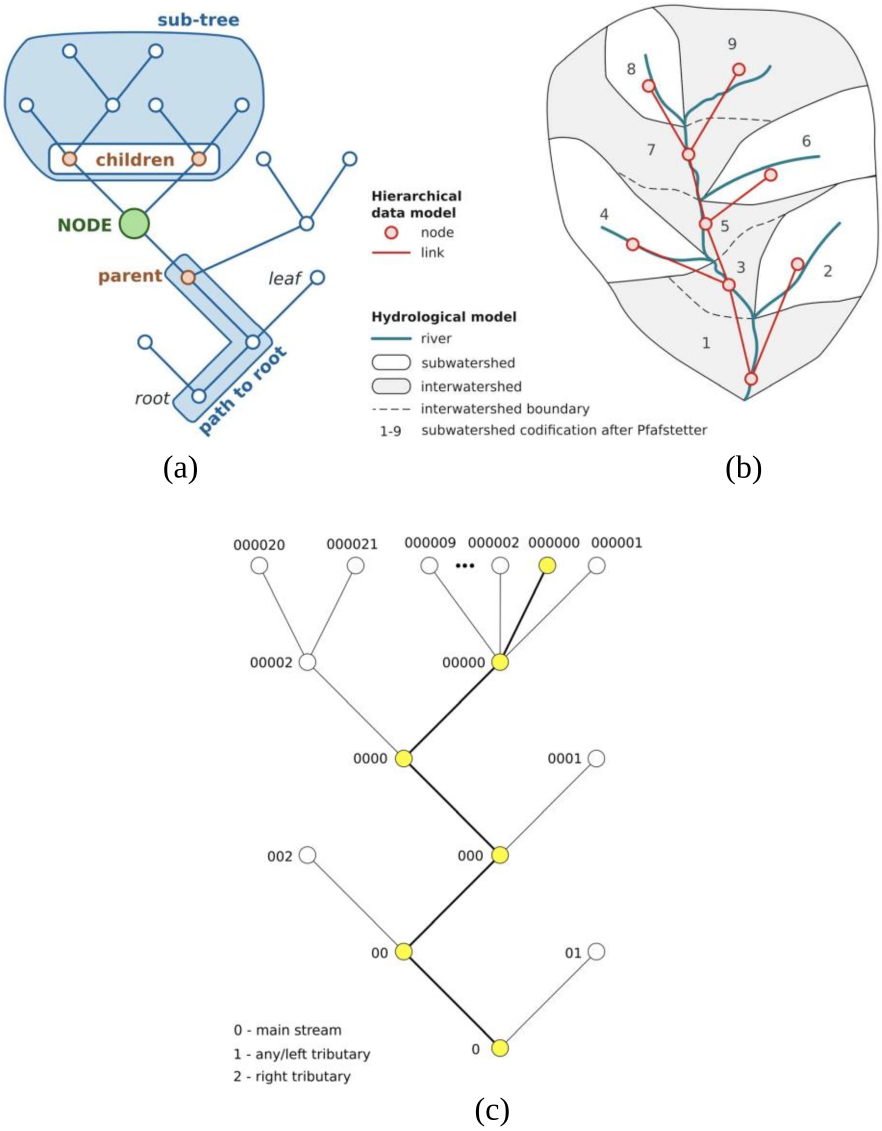


Figure 3.1. Binary Tree Data Model (a); Topology of River Network Codification Proposed by Pfafstetter (b); Stream Network Data Structure Model (c) (Demir & Szczepanek, 2017)

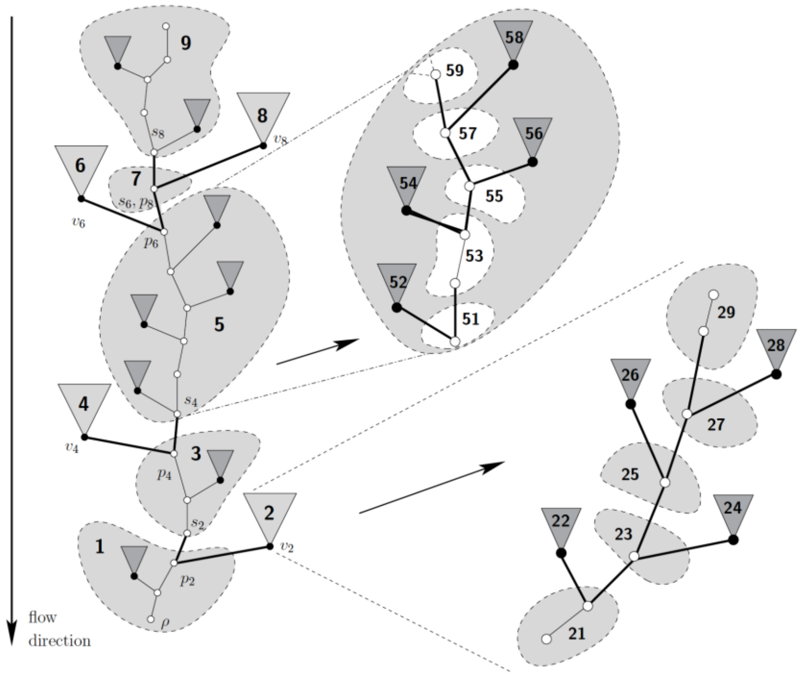


Figure 3.2. Flow Direction and Binary Tree Data Model corresponding to Iterative Pfafstetter River Network Codification (Arge et al., 2006)

In this study, the river network is formatted as a binary tree data structure (Hagberg et al., 2020). After selecting the outlet of the river network, which is the root of the binary tree, traversing algorithms should be applied to find the desired elements (root, parent, child, leaf, etc.). Then, we could start the labeling process, which means assigning the correct Pfafstetter number for each element, recursively.

Breadth First search (BFS) and Depth First search (DFS) are the two most fundamental searching and tree traversal algorithms (Everitt & Hutter, 2015). For our input, the river network, basin outlet can be presented as the root of tree, and every other node might be assigned with parent-child relationship, respectively.

3.1.1 Workflow

The workflow for computing the river networks in Pfafstetter codification algorithm is demonstrated in Figure 3.3.

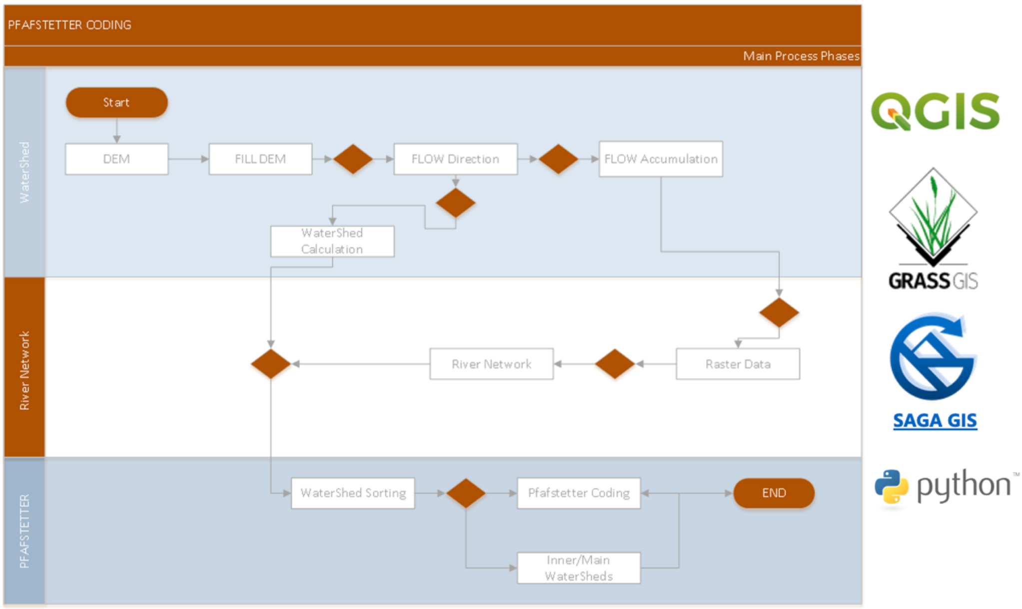


Figure 3.3. The Workflow for Computing the River Networks in the Pfafstetter Codification Algorithm

The flow of process consists of three main steps of action in order to execute the developed codification algorithm.

1st step is called as "Watershed (Catchment)" which is constructed by SAGA, GRASS, and GDAL tools in QGIS software and plug-in. First of all, Digital Elevation Model (DEM) is obtained from NASA Shuttle Radar Topography Mission (SRTM) Global 1 arc-second (about 30 meters) (NASA, 2020). After interpolating voids and filling missing data in DEM for catchment delineation with using Raster analysis tools in GDAL, filling the sinks algorithm in SAGA tools (Terrain Analysis -> Hydrology) is used. Strahler order of the river network can be obtained with SAGA -> Terrain Analysis -> Channels in QGIS.

2nd step is called as "River (Channel or Stream) Network". This is built via SAGA and GRASS tools in QGIS domain. By using SAGA tools (SAGA -> Terrain Analysis -> Channels -> Channel Network & Drainage Basins), we can get some outputs such as Flow Direction, Drainage Basin (raster), Channels, and Drainage Basin (vector). Similarly, stream and catchment delineation can be built via GRASS tools (Figure 3.4.) (Jasiewicz & Metz, 2011).

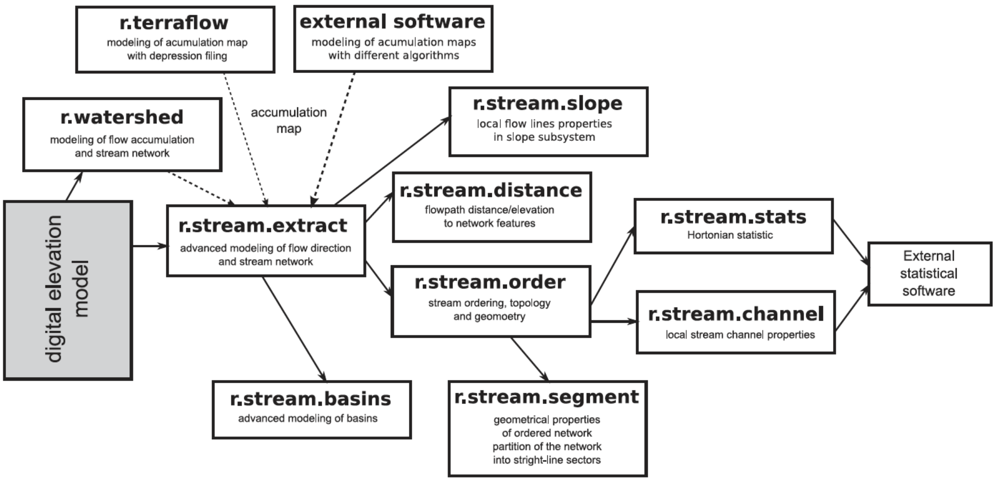


Figure 3.4. The Structure of the GRASS and Data Flow Between Particular Modules and External Software (Jasiewicz & Metz, 2011)

Figure 3.5. shows the details of the operations performed in the above two steps in order to reach the river network (Lauermann et al., 2016).

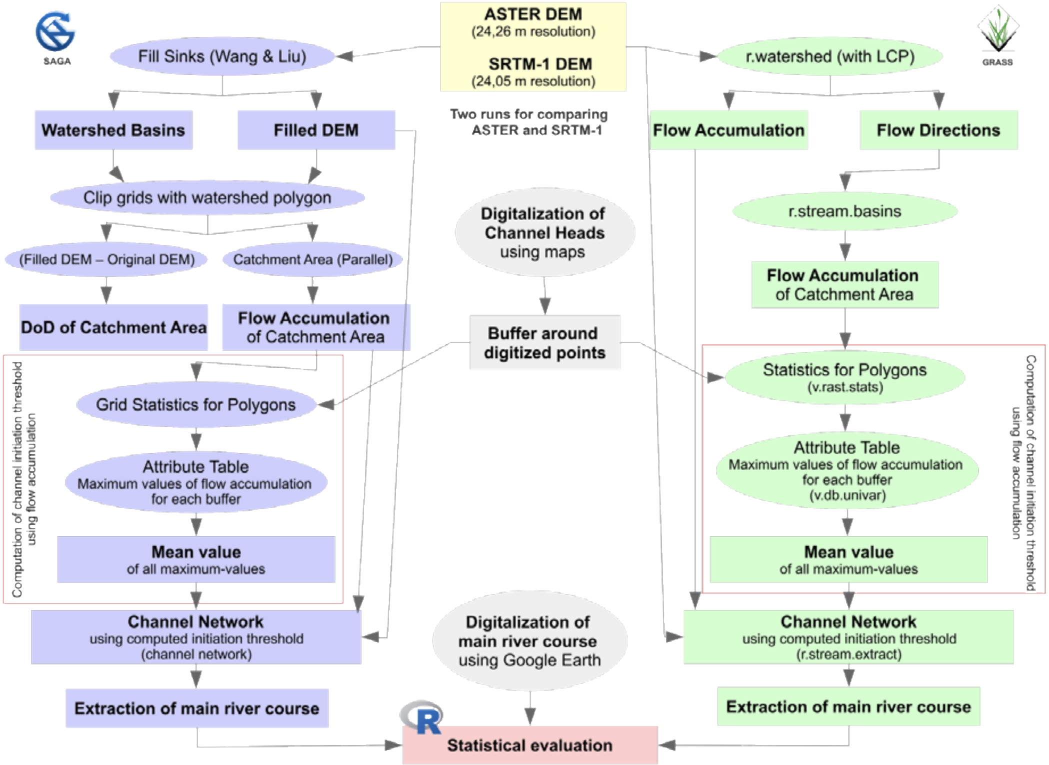


Figure 3.5. Workflow for the Channel Networks in SAGA and GRASS (Lauermann et al., 2016)

3rd step is called as "Pfafstetter Codification Algorithm". It is coded by python programming language. The developed software can be embedded in GIS applications.

The mathematical model of this study is constructed with using python builder in QGIS software, main outputs of the model are Filled DEM, River Network, and Root Node (outlet), which are shown as boxes with green background in Figure 3.6. The source python code of the model is shown in Appendix-A (QGIS, 2020).

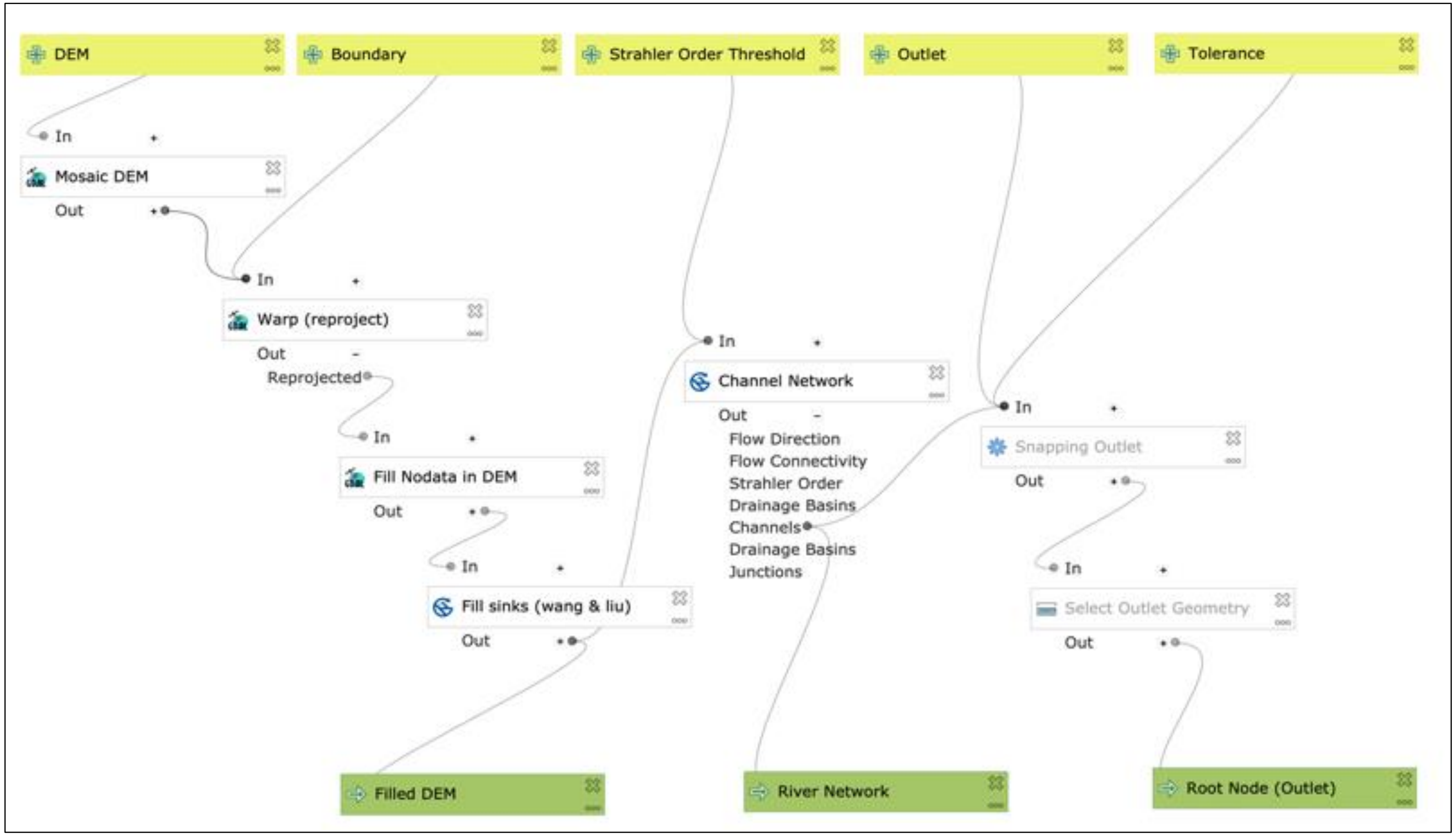


Figure 3.6. Mathematical Model used in QGIS Software in the Study

The Flow Direction is derived from the DEM, it is defined as the direction of steepest slope distance, which is calculated among the center of the cell with respect to Euclidean distance) from each cell. The eight-direction (D8) is represented as the directions of relating to the eight adjacent cells into which flow could go (Jenson & Domingue, 1988). The elevation surface, flow direction, and 8 direction coding are given in Figure 3.7, respectively (ESRI, 2000).

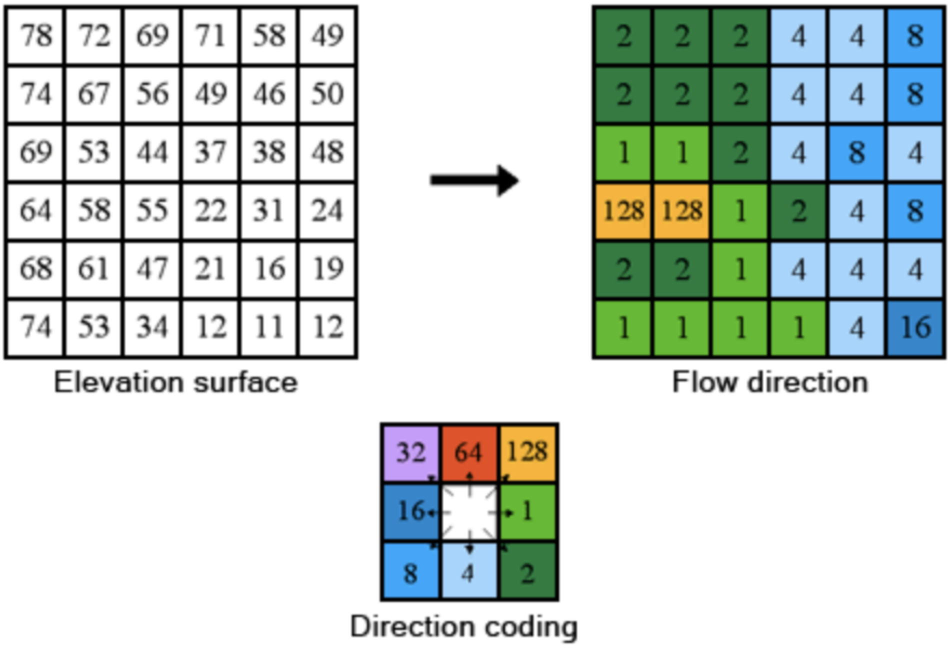


Figure 3.7. The Flow Direction with D8 (ESRI, 2000)

The Flow Accumulation is used for hydrological analysis in order to account accumulated flow as the collected weight of all cells leaking into each downslope cell in the yield raster data. The weight of value 1 is practiced when there is no weight raster data, and the value of cells in the output raster data is the number of cells that flow into each cell. Figure 3.8 depicts the relationship between Flow Direction and Flow Accumulation (ESRI, 2000).

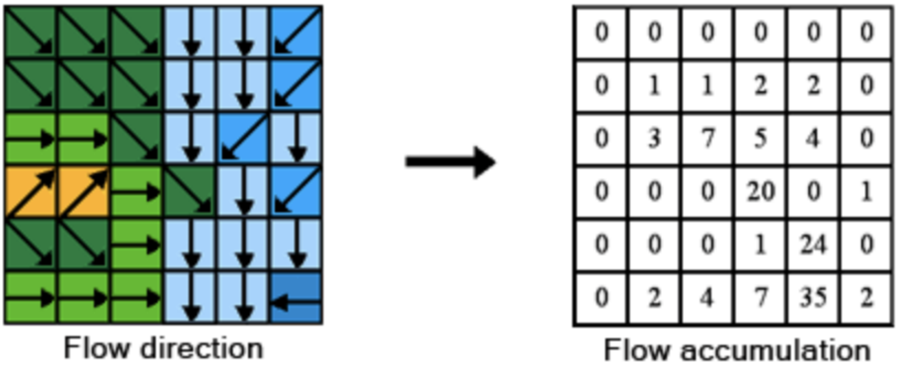


Figure 3.8. The Flow Direction and the Flow Accumulation (ESRI, 2000)

3.1.2 Developed Pfafstetter Codification Algorithm

The main purpose of the developed algorithm is to classify the river network belonging to any basin hydrologically by adhering to the Pfafstetter method. While coding the developed algorithm, the concept of the Object-Oriented Programming Language (OOPL) was taken as basis (Graser & Olaya, 2015). In this context, the basic class and object structures of the software are presented in a solid selection within the discipline of data structures. Furthermore, the software is coded with "Python" which is more flexible and can be integrated into GIS applications (Lee & Hubbard, 2015).

While coding the Pfafstetter method, it was observed that there are three key points that must be covered. These can be summarized as selecting suitable input data structure, manipulating with using searching-sorting-traversing algorithms, and repeating the finding solution case by case recursively.

While dealing with the problem, processing the input data set is of vital importance in terms of time and space complexity. Firstly, one of the most challenging things is selecting the appropriate data structures. Within the scope of the river network to be used in the study, the fact that the input data set can be obtained from completely open sources with our own means is a great advantage in terms of understanding the input data structure and choosing the most suitable data structure. Thus, it has been determined that the data structure has binary tree properties.

After selecting the outlet of the river network, which is the root of the binary tree, traversing algorithms should be applied to find the desired elements (root, parent, child, leaf, etc.). Then, we could start the labeling process, which means assigning the correct Pfafstetter number for each element, recursively.

Although there are many traversing algorithms for tree structures in the literature, the most well-known ones are Depth First Search (DFS) and Breadth First Search (BFS). Due to the unbalance structure of the river network, the DFS algorithm in which priority goes to the deepest node is preferred in terms of performance.

Eventually, in order to code all the elements in the tree structure in accordance with the Pfafstetter logic, we need to define different situations to respond to all options on the table. To solve the problem recursively, in accordance with the Pfafstetter method (1^{st} rule: tributaries are assigned as "2, 4, 6, 8" with respect to the selected largest four drainage areas; 2^{nd} rule: main stems are labeled as "1, 3, 5, 9", note that 1 and 9 must always be labeled) four base cases were determined at the beginning. These cases were determined based on the depth of the tree (Figure 3.9). Considered in the binary tree data structure, the main tree consists of sub-trees. Therefore, the major tree might be adapted by sub-tree owing to the suitable cases with respect to tree depth, recursively.

Unified Modeling Language (UML) class diagram, the pseudo code, and the main class of the developed Pfafstetter algorithm are shown in Appendix-B, Appendix-C, and Appendix-D, respectively.

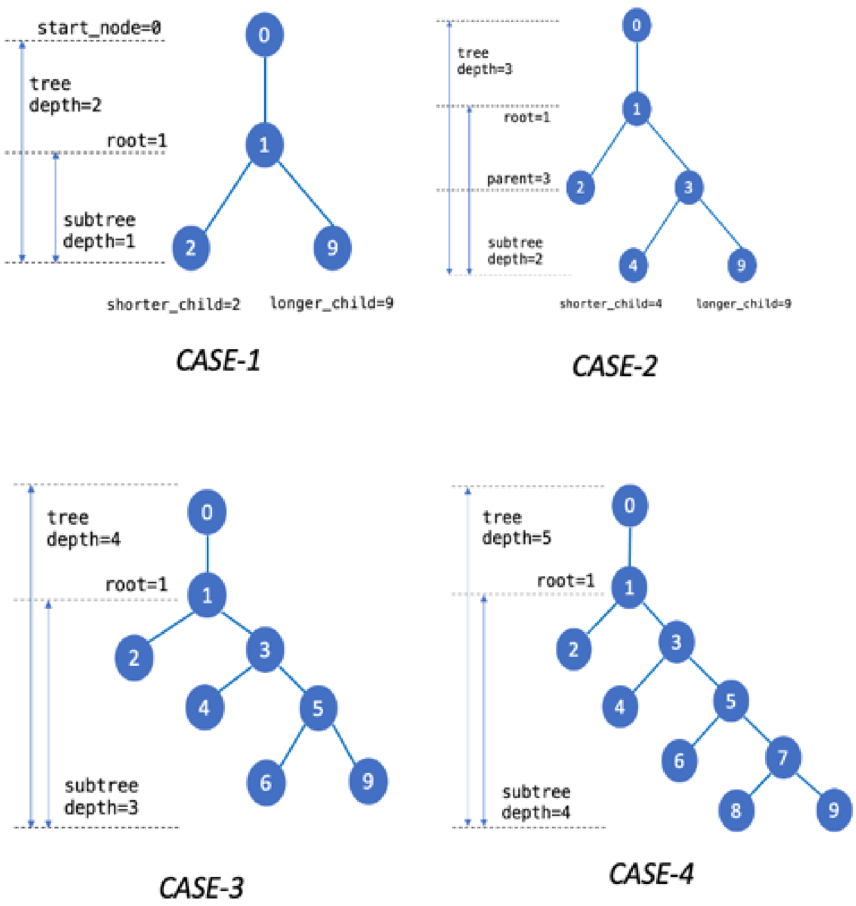


Figure 3.9. All Basic Cases of Developed Pfafstetter Algorithm

3.1.3 Data Visualization of the River Network Structure

In order to better analyze the accuracy of the results of the developed code, the code block is encoded in the python language using the Network-X library (sample code block is shown in Appendix-E) (NetworkX, 2020). It is aimed to display the whole picture in a more understandable and practical way.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, the outcomes of the developed Pfafstetter algorithm are presented for selected three different river basins, which are Çakıt basin in Adana (the southern part of Turkey), Yeşilırmak basin (the northern part of Turkey), and Thames River basin in London (the southern part of England). Similarities and differences between the results of this study and other studies in the literature are discussed.

4.1 The Çakıt Basin

Çakıt Basin, which is a branch of the Seyhan river, pours into the Seyhan reservoir. The largest tributary of Çakıt River is nearly 40 km long. The map of Turkey showing Çakıt Basin is shown in Figure 4.1.

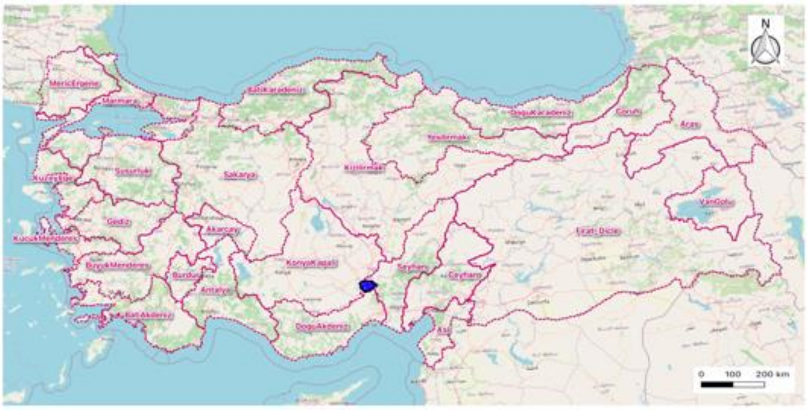


Figure 4.1. The Çakıt River Basin

Within the scope of stream delineation in QGIS using SAGA and GRASS tools, different threshold values were selected, while obtaining Çakıt river network at a scale 1/25000. One can also choose the default values for both GIS tools. Stream network of the basin is created by SAGA (blue lines) and GRASS (red lines) tools in QGIS from SRTM DEM (Figure 4.2).

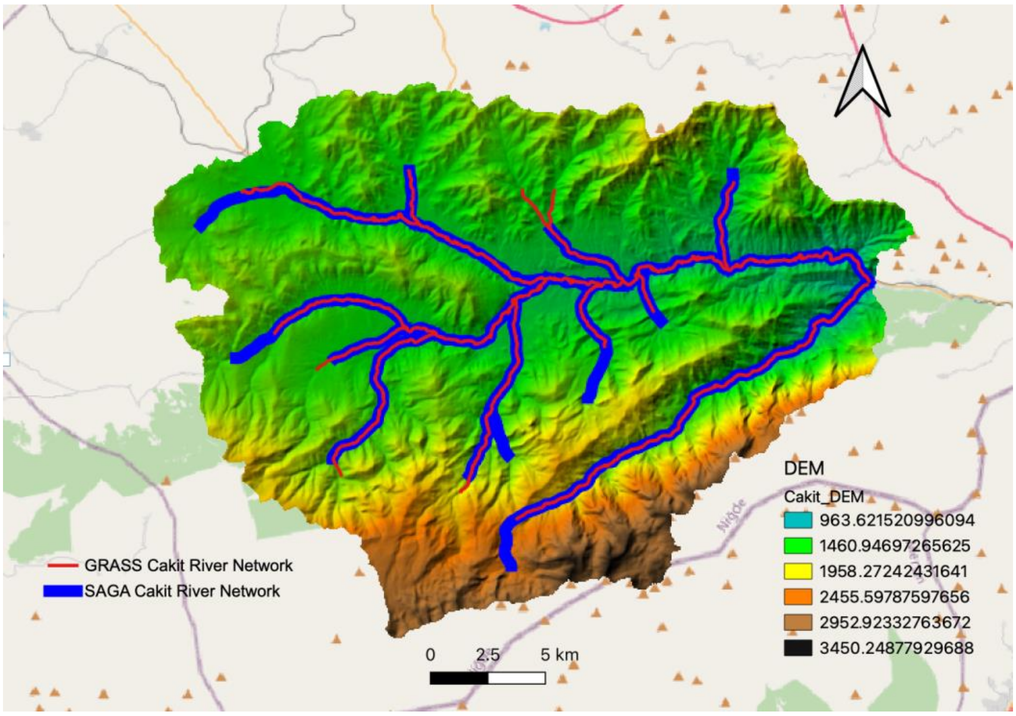


Figure 4.2. The Çakıt Stream Network

The first part of the attribute table [SEGMENT_ID (Unique ID), NODE_A (End Node), NODE_B (Start Node), BASIN, ORDER, ORDER_CELL (Strahler Order), LENGTH (distance among two adjacent nodes as meters)] is obtained with the GRASS tool in QGIS; the last column of the attribute table [PFAFSTTER (Pfafstetter codification code in level 3)] is derived from the developed software. NODE_A, NODE_B, and LENGTH are the inputs of the algorithm in order to calculate the Pfafstetter code (Figure 4.3).

	SEGMENT_ID	NODE_A	NODE_B	BASIN	ORDER	ORDER_CELL	LENGTH	PFAFSTETTER CODE
1	1	1	5	1	1	7	7552,9350596000	29
2	2	2	5	1	1	7	4573,1580054000	22
3	3	3	7	1	1	7	5093,4523779000	69
4	6	6	7	1	1	7	2507,9393924000	62
5	4	4	11	1	1	7	9990,5800795000	8
6	13	13	11	1	2	8	1633,6753237000	91
7	8	8	13	1	1	7	6716,8333291000	92
8	10	10	13	1	1	7	9723,5952314000	99
9	16	17	14	1	3	9	7914,7012947000	1
10	7	7	15	1	2	8	5545,5844123000	61
11	11	11	15	1	2	8	4825,5844123000	7
12	9	9	16	1	1	7	6274,1125497000	369
13	18	19	16	1	3	9	2265,8073580000	362
14	5	5	17	1	2	8	16498,6623250000	21
15	20	21	17	1	3	9	8299,9199077000	31
16	15	16	18	1	3	9	1925,5129855000	361
17	22	24	18	1	2	8	4665,2900398000	37
18	14	15	19	1	3	9	2135,5129855000	5
19	24	25	19	1	2	8	7180,5086528000	41
20	12	12	20	1	1	7	2912,4978336000	34
21	17	18	20	1	3	9	1103,9696962000	35
22	19	20	21	1	3	9	5130,7315985000	33
23	23	23	21	1	2	8	2063,9696962000	321
24	25	26	23	1	1	7	1176,3961031000	322
25	30	31	23	1	1	7	2735,5129855000	329
26	31	32	24	1	1	7	3824,0411229000	38
27	32	33	24	1	1	7	4145,8787848000	39
28	26	27	25	1	2	8	5925,2900398000	43
29	33	34	25	1	1	7	4148,8939367000	42
30	21	22	27	1	1	7	6372,2748879000	44
31	27	28	27	1	2	8	332,1320343600	45
32	28	29	28	1	1	7	749,1168824500	46
33	29	30	28	1	1	7	1293,3809512000	49

Figure 4.3. The Attribute Table of Çakıt River Network with Pfafstetter Code

In this context, it is basically enough to use just start (NODE_B) and end node (NODE_A) relationship to construct the network structure as a complete binary tree. Finding the root note is the first step. In this example, "14" is the root node of the tree, since it is not repeated at NODE_B column. "14" is the parent node of "17", similarly "17" has two children which are "5" and "21”. It might be possible to put the rest of the nodes into the tree using the same logic. At the end, the attribute table in the example can be encoded easily in accordance with the binary tree data structure (Figure 4.3).

We can join two tables via Relational Database Management System (RDBMS) to merge two different data (the first one is derived from QGIS, and the second one is created by the developed Pfafstetter code) on the same platform. In this example, "SEGMENT_ID" is the primary key and "PFAFSTTER CODE" is foreign key, respectively (Figure 4.4).

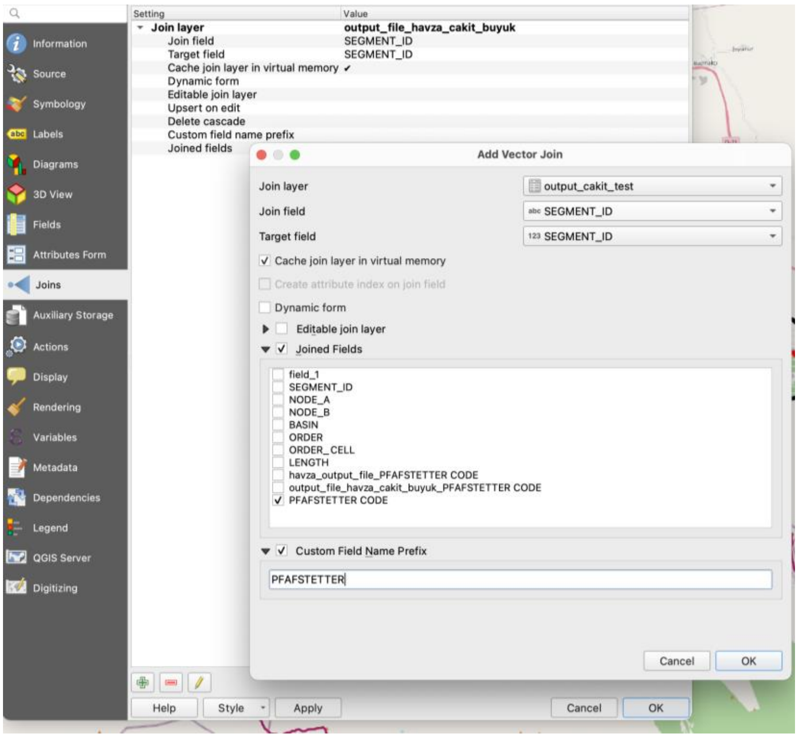


Figure 4.4. RDBMS in QGIS

If it is looked at the attribute table in order by NODE_B, one can easily see the root node of the network with the merely one start node that is not repeated in the array. Therefore, the root of the Pfafstetter code is "1" logically.

The tree consists of "Start Node" (the root and/or parent node, shown as blue circle), "Length" (real world length between two adjacent nodes as meters), and "Label" (Pfafstetter number according to the result of the result of the developed code), all elements of the tree are shown in Figure 4.5. The stream network with Pfafstetter coding for Çakıt basin is given in Figure 4.6.

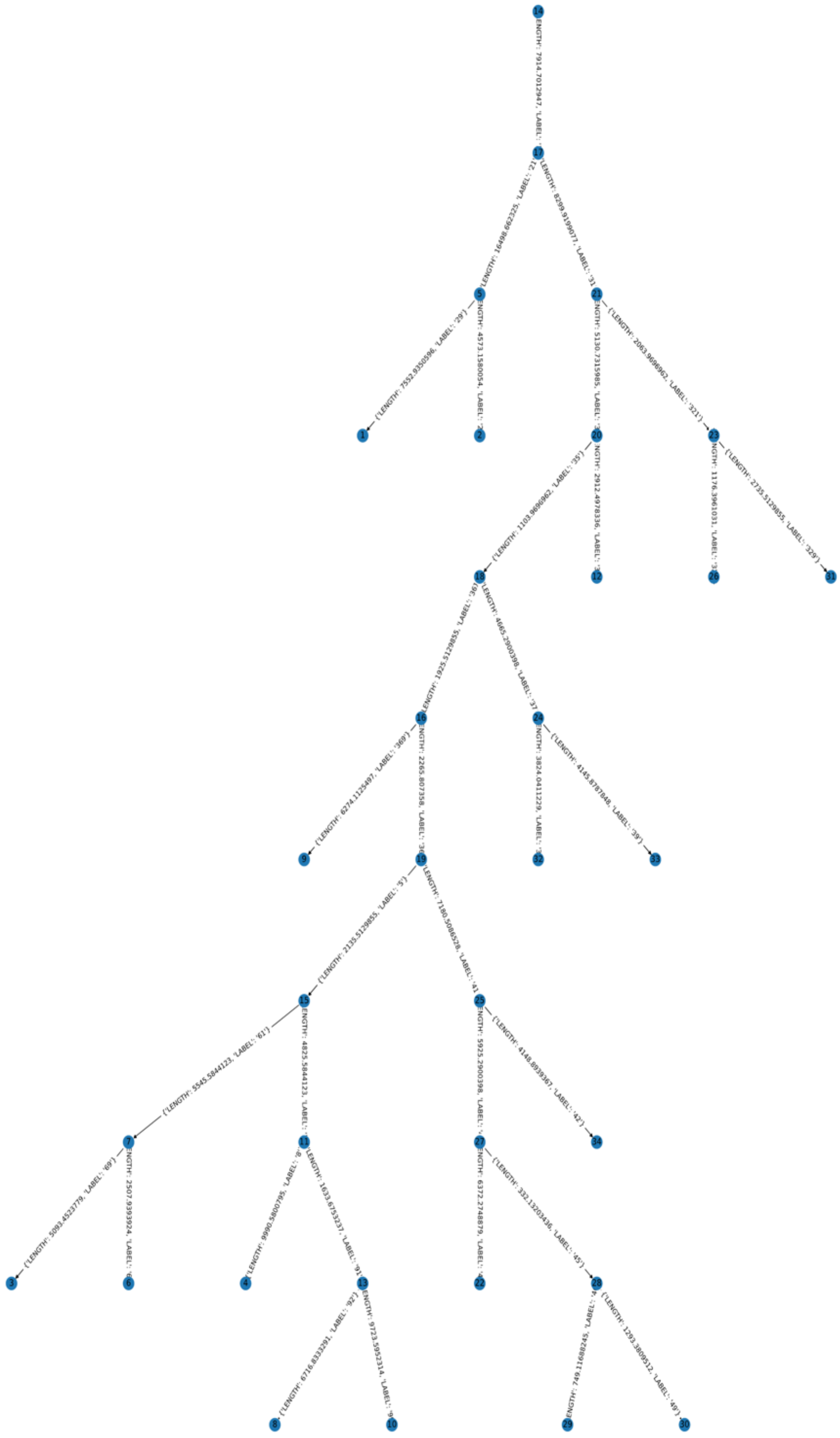


Figure 4.5. Data Visualization of the River Network as Binary Tree (Çakıt Basin)

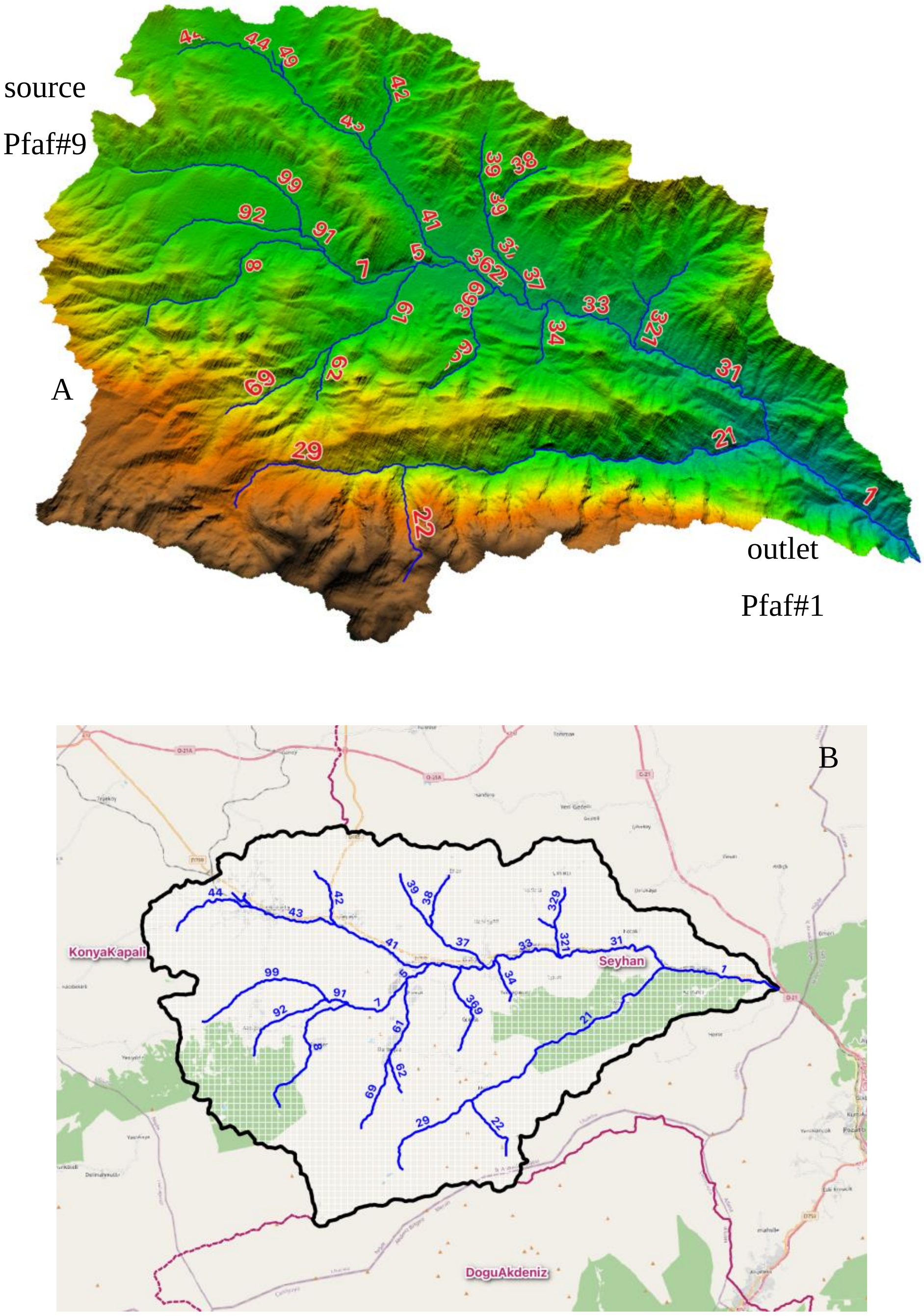


Figure 4.6. Pfafstetter coding obtained from the developed code (a) and manually (b) for Çakıt Basin

As seen in Figure 4.6, the result is tested with the manual coding and it is found that the code is running correctly.

4.2 The Yeşilırmak Basin

Yeşilirmak Basin is one of Turkey's 25 basins and it is discharged into the Black Sea (Figure 4.7). Yeşilirmak basin which has an area of 39.628 km^2, is the third largest catchment of Turkey. Covering 519 km Yeşilırmak River originates from the slopes of Köse and Kızıldağ mountains located within the borders of Sivas province. The largest tributary of Yeşilırmak River is nearly 320 km long. While reaching Samsun-Çarşamba from here, it finally joins with three important branches (Çekerek River, Tersakan Stream and Kelkit Stream) before flowing into the Black Sea. The areas of 11 provinces, namely Tokat, Samsun, Amasya, Çorum, Sivas, Yozgat, Gümüşhane, Giresun, Erzincan, Ordu and Bayburt, are located within the basin boundaries at varying rates (Yeşilırmak Basin Flood Management Plan, 2015).

Figure 4.7. The Map of Turkey Showing Hydrological Basins Containing the Yeşilırmak Basin (Yeşilırmak Basin Flood Management Plan, 2015)

The major tributaries of Yeşilırmak River are Kelkit, Çekerek, Çorum and Tersakan Creeks. The altitude decreases from 3050 meters in the mountainous regions of the basin towards sea level on the Black Sea coast (Yeşilırmak Basin Map, 2020).

The Yeşilırmak River is selected to see the performance of the developed code on a large basin and compare the results with the available coding presented in Figure 4.8 (Darama & Seyrek, 2016).

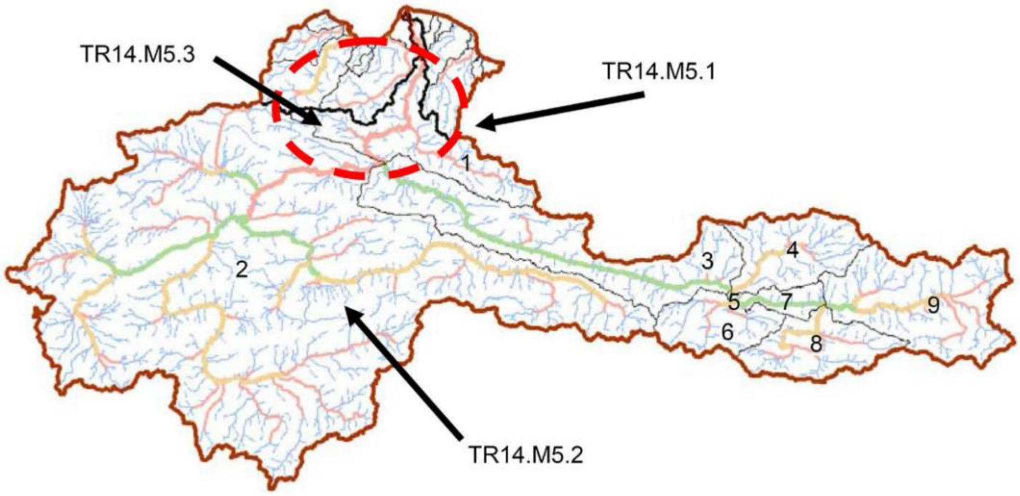
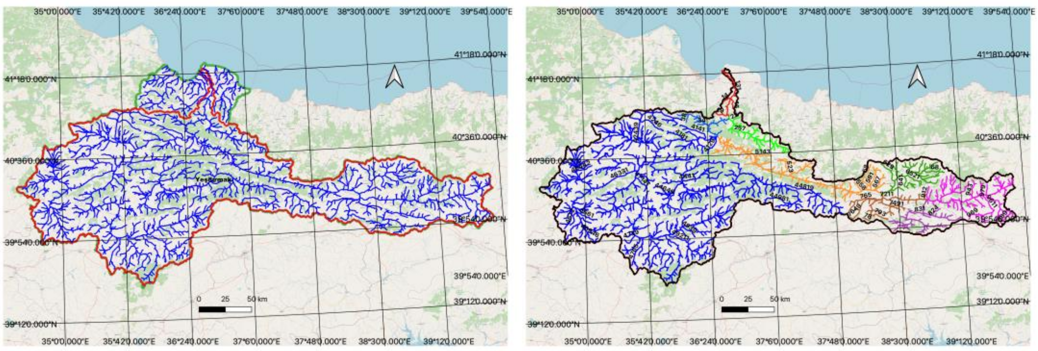


Figure 4.8. Yeşilırmak Basin Pfafstetter Coding (Darama & Seyrek, 2016)

The Yeşilırmak basin border (green polygon), stream network (blue line), and the selected largest sub-basin are presented in Figure 4.9a. Besides, Figure 4.9b depicts the Yeşilırmak river network with all levels of developed Pfafstetter codification label.



(a) (b)

Figure 4.9. The Selected River Network in the largest Sub-basin (red line) of Yeşilırmak (a), Developed Pfafstetter Digits (b)

Figure 4.10 illustrates that the Yeşilırmak Basin with Pfafstetter Code Level-1 colored in range 1 to 9, the small sample part of the attribute table, and the classification of the Pfafstetter codification rank.

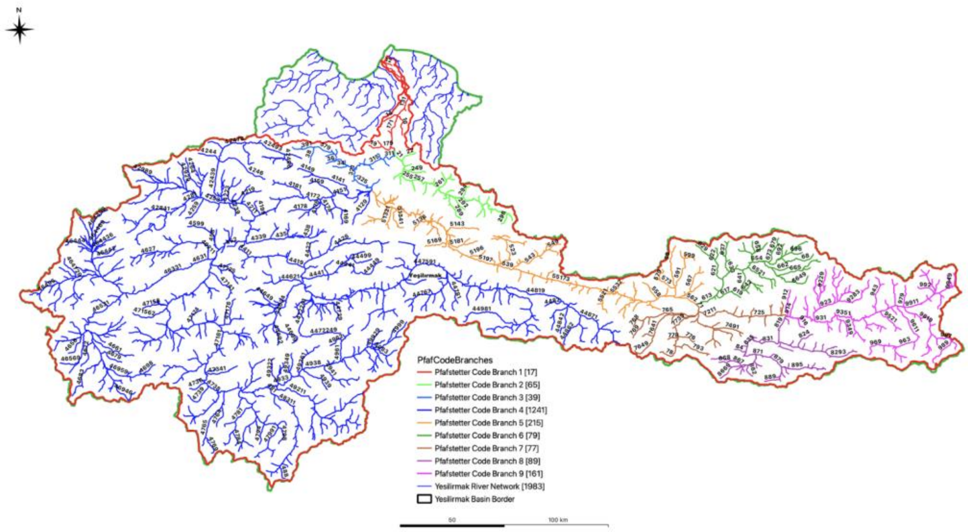


Figure 4.10. All Level Developed Pfafstetter Code and Branch Legend

The developed algorithm was run on the Yeşilırmak Basin and correct results were obtained for all levels. When we look at the number of the nodes in the network in order to understand how large the tree is, it is seen that in Yeşilırmak Basin there are totally 1983 nodes: Pfafstetter Code Branch 1 (means that Pfafstetter code is beginning number 1, such as 11, 12, and 133 etc.) has 17 nodes, Branch 2 has 65 nodes, Branch 3 has 39 nodes, Branch 4 has 1241 nodes, Branch 6 has 79 nodes, Branch 7 has 77 nodes, Branch 8 has 89 nodes, and Branch 9 has 161 nodes, respectively. Figure 4.10 shows all levels of Pfafstetter code and branch legend with the number of the nodes in the Yeşilırmak Basin. Actually, the developed algorithm runs recursively on a tree with approximately 2000 nodes.

For "Yeşilırmak Basin", common and non-common properties between the result of similar studies in the literature (Figure 4.8) (Darama and Seyrek, 2016) and the outcomes of the developed algorithm (Figure 4.10.) are listed below:

- The similar study in literature provides the Pfafstetter numbers at just first level order from 1 to 9. On the other hand, the developed script can create the Pfafstetter digits at higher levels (for example "4644329" is the 7^{th} level Pfafstetter number).
- Within the scope of the number of nodes, it is observed that approximately 2000 nodes are labeled in accordance with the Pfafstetter method with the algorithm developed, but only 9 node numbers are labeled in the referred study.

4.3 The Thames River Catchment

The River Thames is the second largest river in the UK, covering 354 km between its root in the Cotswold Hills and its tidal limit at Teddington Lock in south-west London. The Thames catchment reaches an area of 9948 km^2, including London, Swindon, Oxford, Slough, and Reading (Freshwater Information Platform, 2017). Figure 4.11 illustrates the Thames River Basin in London (Freshwater Information Platform, 2017).

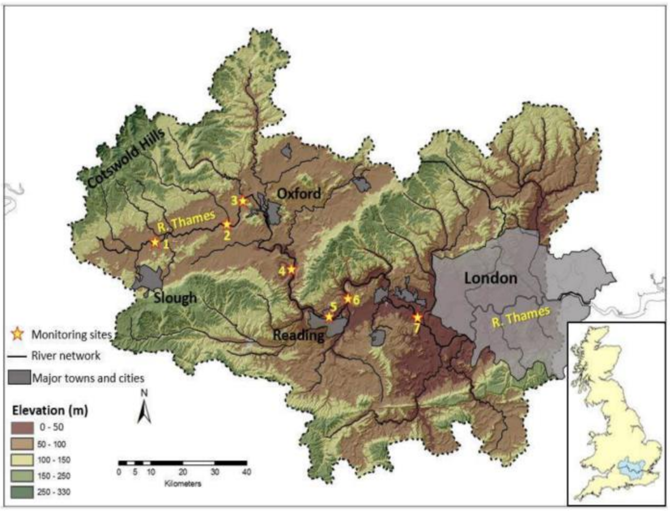


Figure 4.11. The Map of the Thames River Basin in the UK (Freshwater Information Platform, 2017)

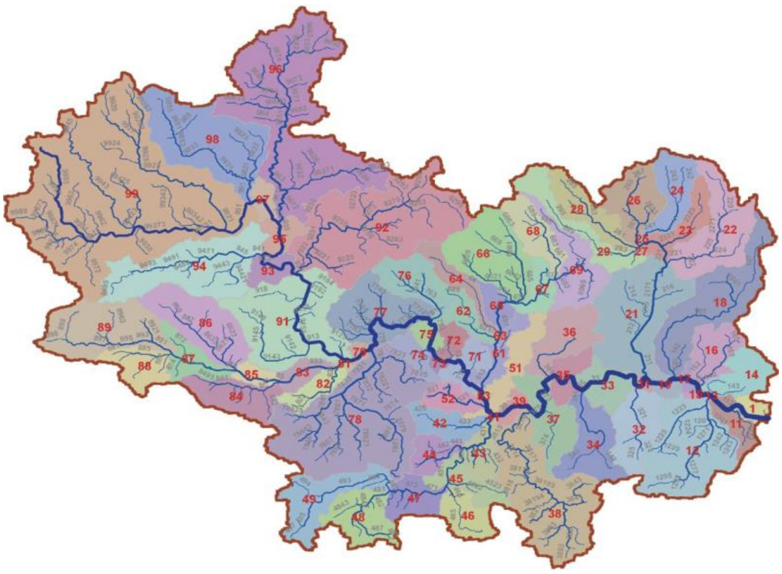


Figure 4.12. The Pfafstetter for the Thames River (de Jager & Vogt, 2010)

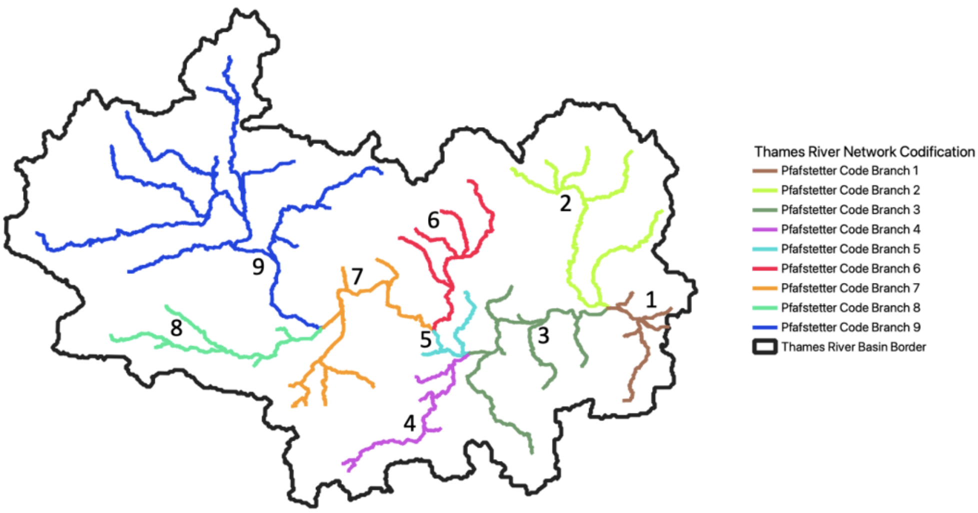


Figure 4.13. The Thames River Catchment with Developed Pfafstetter Codification

Figure 4.13 shows The Thames River Catchment, when the results of the developed algorithm are compared with previous studies (Figure 4.12) (de Jager & Vogt, 2010) on the coding of the Thames River Basin codification, it is seen that they show a great similarity.

The river basins located in different areas and having own hydrological characteristics intentionally are selected in terms of getting opportunities for various testing spectrum on the result of the developed Pfafstetter codification algorithm. The reasons for choosing three different river basins can be explained in detail as follows:

- The developed algorithm was first tested with the Çakıt Basin, which is a small basin. The largest tributary of Çakıt River is nearly 40 km long. In this way, a balance of time and space complexity has been established on the tree created with few nodes and edges. It was easily seen that the labeling made by visualizing the results of the tree created in the binary tree structure is fully fitted with the Pfafstetter method. Especially in bug and debug operations of the code, it has been checked that the labeling created in theory and practice by working in a small network structure is compatible with each other. In short, the Çakıt Basin has made it possible to use it as a benchmark for the realization of the designed methodology, where 33 nodes were used.
- The Yeşilirmak Basin, which is a larger basin, is used to test the code with a larger number of the nodes (nearly 2000). As mentioned earlier, each node in the tree must be labeled according to the Pfafstetter method. The developed algorithm is designed as tagging recursively. Therefore, as the number of nodes increases, the run time increases logarithmically. Although the run time of Çakıt basin is approximately a minute, it takes hours for Yeşilirmak due to having more nodes.
- Eventually, 2523 nodes were labeled with respect to Pfafstetter method for the Thames River Network in this study. The result of a previous study in the literature (de Jager & Vogt, 2010) is given a good opportunity to make a cross-check with the developed algorithm. Besides, the Thames River basin implementation gives possibility to test the inputs having different datum and coordinates.

The proposed coding system has some limitations. These can be listed as follows:

- The developed algorithm accepts the river network to be given as input as a binary tree data structure. It gives an error if this structure is not provided.
- In terms of scale-dependent, 30 m resolution SRTM and 1/25,000 scaled maps are used in this study. Theoretically, the developed algorithm can be run with different SRTM resolutions and mapping scales.
- Pfafstetter system could also be applied for the streams discharging into the sea (Figure 2.21a) and discharging to a natural lake (Figure 2.21b). However, in this study, the design of the developed algorithm is not suitable for running over these types of rivers. Due to the fact that the entire basin cannot be given as input to the system, the networks within the basin must be selected and processed separately. If it is desired to encode the entire basin at the same time, the inputs to the separate system must be combined or processed manually.

The developed Pfafstetter coding has been examined in all aspects and a SWOT (Strengths, Weaknesses, Opportunities, Threats) analyses has been made in Figure 4.14.

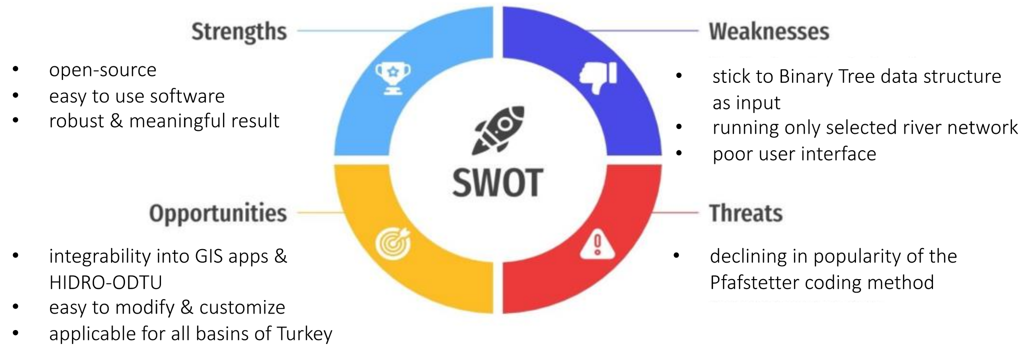


Figure 4.14. The SWOT Analysis of the Developed Pfafstetter Coding

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Recently, owing to the incredible development in the GIS (Geographic Information Systems) and software fields, besides effectively using the network and topology instruments, great advances have been made in the field of hydrologic coding through strong interaction and communication between multidisciplinary sciences.

If all hydrological properties can be combined in a network that has its own sui generis situation and topological properties, meaningful and holistic analyzes can be formed more easily. Additionally, GIS topology is an essential phenomenon for performing different types of network analysis, therefore the topological network model manages spatial relationships by illustrating shapes as the binary tree of topological instruments (node, edge, etc.) via computer geometry.

The main purpose of the developed algorithm is to classify the river network belonging to any basin hydrologically by adhering to the Pfafstetter method. While coding the developed algorithm, the concept of the Object-Oriented Programming (OOP) was taken as a basis.

The developed hydrological codification algorithm might be integrated easily with python plug-in into the GIS programs (SAGA, GRASS, QGIS, etc.) in order to improve the hydrological analysis capabilities.

The biggest advantage of the developed algorithm is open-source code. Also, it can be easily modified and integrated into any GIS application. In this study, one of the most well-known GIS software as QGIS was preferred that is the reason for it might be easily reached as an open-source and embedded python plug-in. The scripts are available via GitHub (https://github.com/GIStechno/Graph.git).

The compiled code was applied to three different river basins with different geographical features and successful results were obtained. As in the examples of such a large tree (Yeşilirmak and Thames River), it has been confirmed that the algorithm developed yields very sufficient and successful results even in a large river network with a large number of nodes.

5.2 Future work

The developed codification algorithm might be carried out to improve the hydrological analysis tools such as SAGA, GRASS tools in QGIS, and even special software like HIDRO-ODTU (Determination of Hydrological Cycle Parameters with a Conceptual Hydrological Model-TUBITAK 115Y041). Therefore, HIDRO-ODTU will gain a topological feature, allowing more detailed hydrological analysis to be made to its current version.

As the number of nodes in the group increases, time complexity increases as each run recursively traverse up to the deepest child. With computational complexity analysis, the balance of time and space complexity of the algorithm can be considered, therefore running time can be optimized with the help of parallel programming and other methods.

The developed algorithm accepts the river network to be given as input as a binary tree data structure. However, since there is no perfect data in practice, the data cleaning to be processed as input can be added to the developed methodology. By detecting which element (node or edge) of input has a problem, feedback on the quality of the data can be provided to the developers.

The developed code for Pfafstetter encoding of river basins can be applied to all basins of Turkey. It would be possible to update river basins dataset that is needed for primary catchment coding and integrate the system to European level.

REFERENCES

Arge, L., Danner, A., Haverkort, H., & Zeh, N. (2006). I/O-efficient hierarchical watershed decomposition of grid terrain models. *Progress in Spatial Data Handling - 12th International Symposium on Spatial Data Handling, SDH 2006*. https://doi.org/10.1007/3-540-35589-8_51

Britton, P. (2002). *Review of Existing River Coding Systems For River Basin Management and Reporting (Draft). Version 1.8.* http://edcdaac.usgs.gov/gtopo30/hydro/P311.html

Celko, J. (2012). Joe Celko's Trees and hierarchies in SQL for smarties. Elsevier.

Curtin, K. M. (2007). Network Analysis in Geographic Information Science: Review, Assessment, and Projections. *Cartography and Geographic Information Science*, *34*(2), 103–111. https://doi.org/10.1559/152304007781002163

Darama, Y., & Seyrek, K. (2016). Determination of Watershed Boundaries in Turkey by GIS Based Hydrological River Basin Coding. *Journal of Water Resource and Protection (JWARP)*. https://www.scirp.org/html/2-9402910_71186.htm

de Bartolo, S., Dell'Accio, F., & Veltri, M. (2009). Approximations on the Peano river network: Application of the Horton-Strahler hierarchy to the case of low connections. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *79*(2), 026108. https://doi.org/10.1103/PhysRevE.79.026108

de Jager, A. L., & Vogt, J. V. (2010). Development and demonstration of a structured hydrological feature coding system for Europe. *Hydrological Sciences Journal*, *55*(5), 661–675. https://doi.org/10.1080/02626667.2010.490786

Demir, I., & Szczepanek, R. (2017). Optimization of river network representation data models for web-based systems. *Earth and Space Science*, *4*(6). https://doi.org/10.1002/2016EA000224

Demšar, U., Špatenková, O., & Virrantaus, K. (2008). Identifying critical locations in a spatial network with graph theory. *Transactions in GIS*, *12*(1). https://doi.org/10.1111/j.1467-9671.2008.01086.x

Dooley, J. F. (2002). *Food and Agriculture Organization of the United Nations Spatial Standards & Norm Task Force (SSN-TF).*

EEA. (2008). *Building the EEA European Catchment and Rivers Network System (ECRINS) from CCM v2.1 EEA Technical report.*

ESRI. (2000). *ArcGIS Help 10.1 - An overview of the Hydrology toolset*. http://resources.arcgis.com/en/help/main/10.1/index.html#/An_overview_of_the_Hydrology_tools/009z0000004w000000/

EU Project. (2009). *Trans-Boundary River Management Phase II for the Kura River basin – Armenia, Georgia, Azerbaijan*.

European Commission. (2010). *European Coding System for Hydrological Features. Guidance Document No.22 - Updated Guidance on Implementing the Geographical Information System (GIS) Elements of the EU Water Policy, WFD, 101-106.*

European Communities. (2003). *Common Implementation Strategy for the Water Framework Directive (2000/60/EC).*

European Environment Agency. (2012). *EEA Catchments and Rivers Network System - ECRINS Rationales, building and improving for widening uses to Water Accounts and WISE applications, EEA Technical Report No 7/2012.*

Everitt T., Hutter M. (2015) Analytical Results on the BFS vs. DFS Algorithm Selection Problem. Part I: Tree Search. In: Pfahringer B., Renz J. (eds) AI 2015:

Advances in Artificial Intelligence. AI 2015. Lecture Notes in Computer Science, vol 9457. Springer, Cham.

Flavin, R. W., Andrews, A. J., Kronvang, B., Müller-Wohlfeil, D., Demuth, S., & Birkenmayer, A. (1998). *ERICA. European Rivers and Catchments. European Environment Agency) Copenhagen.*

Freshwater Information Platform. (2017). *Case Study Synthesis Thames - MARS project (Managing Aquatic ecosystems and water Resources under multiple Stress)*. http://fis.freshwatertools.eu/index.php/thames.html

Furnans, F., & Olivera, F. (2001). *Watershed topology: The Pfafstetter system* (Vol. 21).

Fürst, J., & Hörhan, T. (2009). Coding of watershed and river hierarchy to support GIS-based hydrological analyses at different scales. *Computers and Geosciences*, *35*(3), 688–696. https://doi.org/10.1016/j.cageo.2008.04.007

Globevnik, L., & Kirn, T. (2009). *Small water bodies - Assessment of status and threats of standing small water bodies. Report of the European Environment Agency (No. EEA/ADS/06/001-Water)*.

Globevnik, L., Vrenko, D. Z., Jensen, H. M., & Kovacic, M. (2010). *Proposed Coding of European Rivers for Water Framework Directive* (Vol. 29).

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). Data structures and algorithms in Python. John Wiley & Sons Ltd.

Graser, A., & Olaya, V. (2015). Processing: A python framework for the seamless integration of geoprocessing tools in QGIS. *ISPRS International Journal of Geo-Information*, *4*(4). https://doi.org/10.3390/ijgi4042219

Gravelius, H. (1914). *Rivers. G.J. göschen Publishing*. Berlin - Google Scholar.

Gray, B. (2004). *Australian river assessment system: National guidelines for mapping AusRivAS*. http://www.deh.gov.au/water/rivers/nrhp/act/copyright.html

Hagberg, A., Schult, D., & Swart, P. (2020). *NetworkX Reference Release 2.5*.

Horton, R. E. (1945). Erosional development of streams and their drainage basins; Hydrophysical approach to quantitative morphology. *Bulletin of the Geological Society of America*, *56*(3), 275–370. https://doi.org/10.1130/0016-7606(1945)56[275:EDOSAT]2.0.CO;2

Jasiewicz, J., & Metz, M. (2011). A new GRASS GIS toolkit for Hortonian analysis of drainage networks. *Computers and Geosciences*, *37*(8), 1162–1173. https://doi.org/10.1016/j.cageo.2011.03.003

Jenson, S. K., & Domingue, J. O. (1988). Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis. In *Citeseer* (Vol. 54, Issue 11). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.6487&rep=rep1&type=pdf

Karaman, Ç., Bolat, K., & Akyürek, Z. (2019). *HIDRO-ODTU: Web-Based Conceptual Hydrological Modeling Tool*.

Khan, M. Q. I., Venkataratnam, L., Rao, B. R. M., Rao, D. P., & Subrahmanyam, C. (2001). International Classification and Codification of Watersheds and River Basins. *Journal of Water Resources Planning and Management*, *127*(5), 306–315. https://doi.org/10.1061/(asce)0733-9496(2001)127:5(306)

Lauermann, M., Betz, F., & Cyffka, B. (2016). Channel Network Derivation from Digital Elevation Models – An Evaluation of Open Source Approaches. *Bulletin of KSUCTA*, *3*(53).

Lee, K. D., & Hubbard, S. (2015). Data Structures and Algorithms with Python. In *Undergraduate Topics in Computer Science*.

NASA. (2020). *EarthExplorer*. https://earthexplorer.usgs.gov/

NetworkX. (2020). *Labels and Colors - NetworkX 2.5 documentation*. https://networkx.org/documentation/stable/auto_examples/drawing/plot_labels_and_colors.html

QGIS. (2020). https://www.qgis.org/en/site/

Scheidegger, A. E. (1966). Effect of map scale on stream orders. *International Association of Scientific Hydrology. Bulletin*, *11*(3). https://doi.org/10.1080/02626666609493480

Seaber, P. R., Kapinos, F. P., & Knapp, G. L. (1987). *Hydrologic unit maps: US Geological Survey water-supply paper 2294. US Geological Survey*.

Smart, J. S. (1969). Topological properties of channel networks. *Bulletin of the Geological Society of America*, *80*(9), 1757–1774. https://doi.org/10.1130/0016-7606(1969)80[1757:TPOCN]2.0.CO;2

Stein, J. L. (2018). An Enhanced Pfafstetter Catchment Reference System. *Water Resources Research*, *54*(12), 9951-9963. https://doi.org/10.1029/2018WR023218

Strahler, A. N. (1957). Quantitative analysis of watershed geomorphology. *Eos, Transactions American Geophysical Union*, *38*(6).

The National Severe Storms Laboratory (NSSS). (2004). *NSSS Basin Delineation - Presentations and Reports*. https://www.nssl.noaa.gov/projects/basins/reports.php (last access:2020-06-03)

USGS Water Resources. (2001). *USGS Hydrologic Units website*. https://water.usgs.gov/GIS/regions.html

Verdin, K L, & Verdin, J. P. (1999). *A topological system for delineation and codification of the Earth's river basins*.

Verdin, K. L. (2017). Hydrologic Derivatives for Modeling and Analysis - A new global high-resolution database (No. 1053). US Geological Survey.

Vogt, J., Soille, P., de Jager, A., Rimaviči, E., Rimaviči, R., Mehl, W., Foisneau, S., Bódis, K., Dusart, J., Paracchini, M. L., Haastrup, P., & Bamps, C. (2007). *A pan-European river and catchment database. European Commission, EUR, 22920, 120.* https://doi.org/10.2788/35907

Wang, H., Fu, X., & Wang, G. (2013). Multi-tree Coding Method (MCM) for drainage networks supporting high-efficient search. *Computers and Geosciences*, *52*, 300–306. https://doi.org/10.1016/j.cageo.2012.11.005

Yeşilırmak Basin Flood Management Plan. (2015). https://www.tarimorman.gov.tr/SYGM/Belgeler

Yeşilırmak Basin Map. (2020). http://www.yesilirmak.org.tr/documents/harita/t%C3%BCm.pdf

Zhang, L., Wang, G. Q., Dai, B. X., & Li, T. J. (2007). *Classification and Codification Methods of Stream networks in a River Basin: A Review*.

APPENDICES

A. Python Code Block of the QGIS Model

Table A.1. Python Code Block of the QGIS Model (QGIS, 2020)

Python Code Block of the QGIS Model

```
import qgis
from qgis.core import QgsProcessingAlgorithm
from qgis.core import QgsProcessingParameterFeatureSink
class DemModel(QgsProcessingAlgorithm):
    def initAlgorithm(self, config=None):
        self.addParameter(QgsProcessingParameterMultipleLayers('dem', 'DEM',
layerType=QgsProcessing.TypeRaster, defaultValue=None))
        self.addParameter(QgsProcessingParameterVectorLayer('boundary',
'Boundary', types=[QgsProcessing.TypeVectorPolygon], defaultValue=None))
self.addParameter(QgsProcessingParameterNumber('strahlerorderthreshold',
'Strahler Order Threshold', type=QgsProcessingParameterNumber.Integer,
minValue=1, maxValue=20, defaultValue=8))
        self.addParameter(QgsProcessingParameterVectorLayer('outlet', 'Outlet',
types=[QgsProcessing.TypeVectorPoint], defaultValue=None))
        self.addParameter(QgsProcessingParameterNumber('tolerance', 'Tolerance',
type=QgsProcessingParameterNumber.Double, minValue=30, maxValue=5000,
defaultValue=100))
self.addParameter(QgsProcessingParameterVectorDestination('ChannelsStreamNet
work', 'Channels (Stream Network)',
self.addParameter(QgsProcessingParameterRasterDestination('FilledDem', 'Filled
DEM', createByDefault=True, defaultValue=None))
```

```
    def processAlgorithm(self, parameters, context, model_feedback):
        # Build Virtual Raster - Mosaic DEM
        alg_params = {
        outputs['BuildVirtualRasterMosaicDem'] =
processing.run('gdal:buildvirtualraster', alg_params, context=context,
feedback=feedback, is_child_algorithm=True) feedback.setCurrentStep(1)
        # Warp (reproject)
        alg_params = {
            'INPUT': outputs['BuildVirtualRasterMosaicDem']['OUTPUT'],
            'NODATA': -9999,
            'TARGET_RESOLUTION': 30,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT}
        # Fill nodata (fill voids in DEM)
        outputs['FillNodataFillVoidsInDem'] = processing.run('gdal:fillnodata',
alg_params, context=context,
        # Fill sinks (wang & liu)
        alg_params = {
            'ELEV': outputs['FillNodataFillVoidsInDem']['OUTPUT'],
            'MINSLOPE': 0.01,
            'FILLED': parameters['FilledDem'],
            'WSHED': QgsProcessing.TEMPORARY_OUTPUT}
        outputs['FillSinksWangLiu'] = processing.run('saga:fillsinkswangliu',
alg_params, context=context,
        results['FilledDem'] = outputs['FillSinksWangLiu']['FILLED']
        # Channel network and drainage basins
        alg_params = {
```

Python Code Block of the QGIS Model (continued)

```
            'DEM': outputs['FillSinksWangLiu']['FILLED'],
            'THRESHOLD': parameters['strahlerorderthreshold'],
            'BASIN': QgsProcessing.TEMPORARY_OUTPUT,
            'NODES': QgsProcessing.TEMPORARY_OUTPUT,
            'ORDER': QgsProcessing.TEMPORARY_OUTPUT,
            'SEGMENTS': parameters['ChannelsStreamNetwork']}
        outputs['ChannelNetworkAndDrainageBasins'] =
processing.run('saga:channelnetworkanddrainagebasins', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
        results['ChannelsStreamNetwork'] =
outputs['ChannelNetworkAndDrainageBasins']['SEGMENTS']
        alg_params = {
            'BEHAVIOR': 0,
            'INPUT': parameters['outlet'],
    def name(self):
        return 'DEM Model'
    def displayName(self):
        return 'DEM Model'
    def group(self):
        return 'PFAF'
```

B. The Pseudo Code of the Developed Pfafstetter Algorithm

Table B.2. The Pseudo Code of the Developed Pfafstetter Algorithm

The Developed Pfafstetter Algorithm Pseudo Code

```
Step0: Importing "attribute table" as data [] (collection types)
#df = pd.read_excel('input.xlsx')
Step1: Finding the start node
#def find_startNode():
  Search end nodes for the missing sequence number (root-outlet)
  assign NODE_A column --> data [end_node]
  parse  data [end_node] and find missing sequence number;
        for (i=1; i< data [end_node].lenght; i+=2)
          if data [i]+1 != data [i+1]
             root = data [i]+1
          break #startNodeID = 15
Step2: Creating the tree (Binary Tree Data Stucture):
#define a data type for unordered Binary Tree
BinaryTree = object of Node [data [end_node][].lenght]
  Step21: Define a Node
      - id
      - parent id
      - left child id
      - right child id
      - left child length
      - right child length
      - main stems or tributaries
      - pfaf_number
```

The Developed Pfafstetter Algorithm Pseudo Code (continued)

```
 class Node #example of the class structure {
     FINAL id = null --> data [SEGMENT_ID]
     parent_id = null --> data [SEGMENT_ID]
     left_child_id = null --> data [SEGMENT_ID]
     right_child_id = null --> data [SEGMENT_ID]
     left_child_length = null --> data [LENGTH]
     right_child_length = null --> data [LENGTH]
     main_stem = null --> True or False
     pfaf_number --> integer [array]}
step22: place Root as root
#def find_child(startNodeID):
for (i=0;i<= data [end_node].lenght; i++)
 BinaryTree[0] ={
     Root:
       .id = data [SEGMENT_ID] --> 1
       .parent = null
       .left_child_id = data [SEGMENT_ID]
       .right_child_id = data [SEGMENT_ID]
       .left_child_length = data [LENGTH]
       .right_child_length = data [LENGTH] }
step23: check root to end node from [NODE_B] column
       - assign the left and right child with respect to the length
      #searching recursively all subtrees
       - go to left child
       - search left child id in [NODE_B] column
       - if not exist go back
       - go to right child
       - search right child id in [NODE_B] column
```

The Developed Pfafstetter Algorithm Pseudo Code (continued)

```
Step3: Finding the deepest child (1->9) #main stems
  #using traversal method (post-order)
  - start node id = 1
  - end node id = 9
 Step31:
   - if there are two children at the deepest level,
        then compare length. select longer one.
   - mark the nodes on the deepest path as main stem (main=1)
   - mark the other nodes as tributaries (main=0)
 Step32: Sort all tributaries with respect to the length of nodes, then
          the pickup the first 4 length values (the tributaries)
Step42: BFS (Breadth First Search) or DFS (Depth First Search)
      BFS #with using the queue
       build the queue
       assign v as visited and put v into queue
      DFS #with using the stack
       build a stack
       assign v as visited and put v into stack
Step5: Define odd main stems (3-5-7) #interbasins
    - define the path from main nodes to even
        tributaries as odd mains
```

C. UML Class Diagram of the Developed Pfafstetter Algorithm

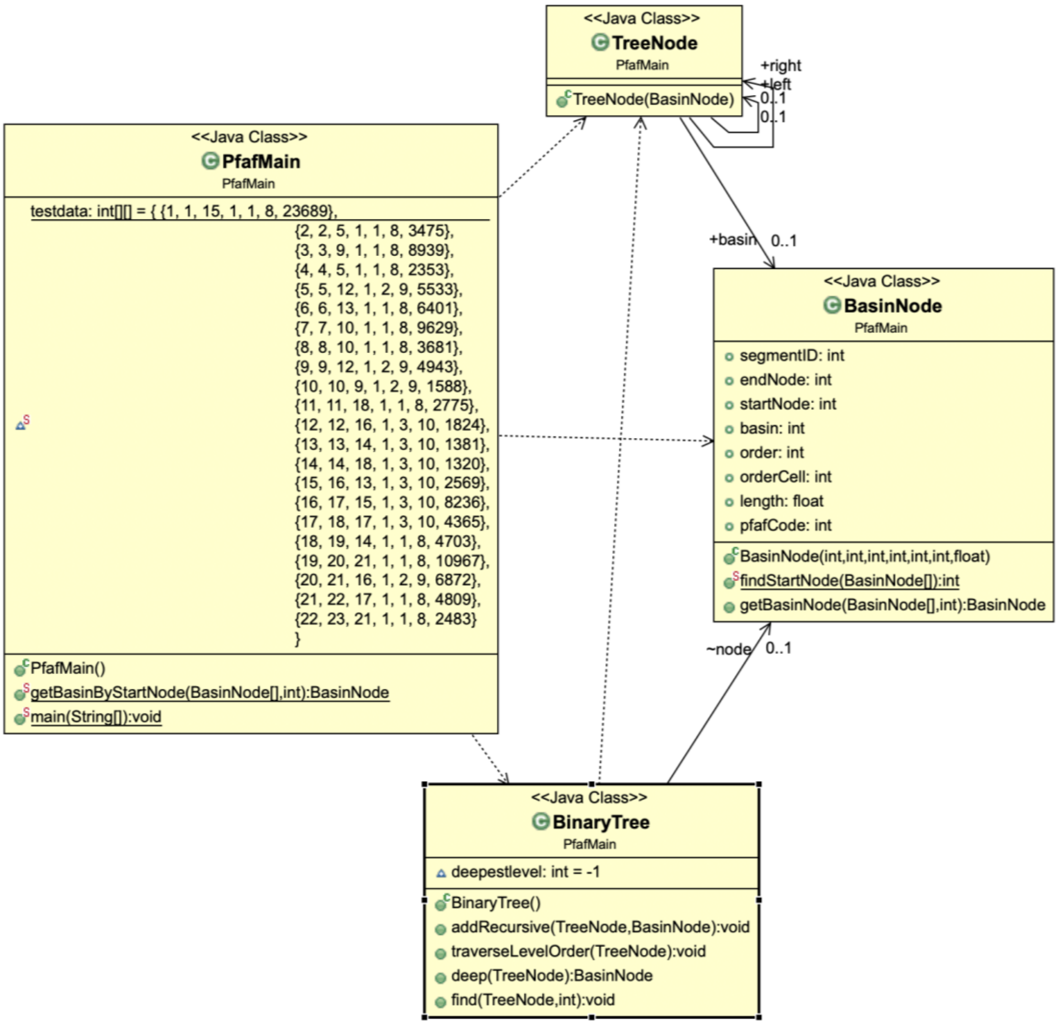


Figure C.1. UML (Unified Modeling Language) Class Diagram

D. Java Code Block of the Main Class of the Developed Pfafstetter Algorithm

Table D.3. Java Code Block of the Main Class of the Developed Algorithm

Java Code Block of the Main Class of the Developed Algorithm

```
package PfafMain;
import java.util.Stack;
public class PfafMain{
static int[][] testdata = { {1, 1, 15, 1, 1, 8, 23689} ..., };
   public static BasinNode getBasinByStartNode(BasinNode[] data, int startNode){
      BasinNode temp = null;
      for (int i = 0; i < data.length; i++)
         if (data[i] != null && data[i].startNode == startNode) {
            temp = data[i];
            data[i] = null;
            return temp;}
      return temp;}
public class BinaryTree {
   int deepestlevel = -1;
   BasinNode node = null;
   public void addRecursive(TreeNode current, BasinNode basin) {
      if (basin.startNode == current.basin.endNode) {
         if (current.left == null ) {
            current.left = new TreeNode(basin); }
         else if (current.right == null) {
            current.right = new TreeNode(basin);}}
       else {
```

```
        if (current.left != null) addRecursive(current.left, basin);
        if (current.right != null) addRecursive(current.right, basin); }}
public void traverseLevelOrder(TreeNode root) {
    if (root == null) {
        System.out.println("Nothing to display!");
        return;}
    Queue<TreeNode> nodes = new LinkedList<>();
    nodes.add(root);
    while (!nodes.isEmpty()) {
        TreeNode node = nodes.remove();
        System.out.println(" " + node.basin.endNode);
        if (node.left != null) {
            nodes.add(node.left);}
        if (node.right != null) {
            nodes.add(node.right);}}}
public BasinNode deep(TreeNode root) {
    find(root, 0);
    return node; }
public void find(TreeNode root, int level) {
    if (root != null) {
        find(root.left, ++level);
        if (level > deepestlevel) {
            node = root.basin;
            deepestlevel = level;}
        find(root.right, level);}}}
```

E. A Sample Code Block of the Data Visualization

Table E.4. A Sample Code Block of Visualization (NetworkX, 2020)

A Sample Code Block of Visualization

```
import matplotlib.pyplot as plt, import networkx as nx
G = nx.cubical_graph()
pos = nx.spring_layout(G)  # positions for all nodes
# nodes
options = {"node_size": 500, "alpha": 0.8}
nx.draw_networkx_nodes(G, pos, nodelist=[0, 1, 2, 3], node_color="r", **options)
nx.draw_networkx_nodes(G, pos, nodelist=[4, 5, 6, 7], node_color="b", **options)
# edges
nx.draw_networkx_edges(G, pos, width=1.0, alpha=0.5)
nx.draw_networkx_edges(
    G,
    pos,
    edgelist=[(0, 1), (1, 2), (2, 3), (3, 0)],
    width=8,
    alpha=0.5,
    edge_color="r",)
nx.draw_networkx_edges(
    G,
    pos,
    edgelist=[(4, 5), (5, 6), (6, 7), (7, 4)],
    width=8,
    alpha=0.5,
    edge_color="b",)
```

```
# some math labels
labels = {}
labels[0] = r"$a$"
labels[1] = r"$b$"
labels[2] = r"$c$"
labels[3] = r"$d$"
labels[4] = r"$\alpha$"
labels[5] = r"$\beta$"
labels[6] = r"$\gamma$"
labels[7] = r"$\delta$"
nx.draw_networkx_labels(G, pos, labels, font_size=16)
plt.axis("off"); plt.show()
```