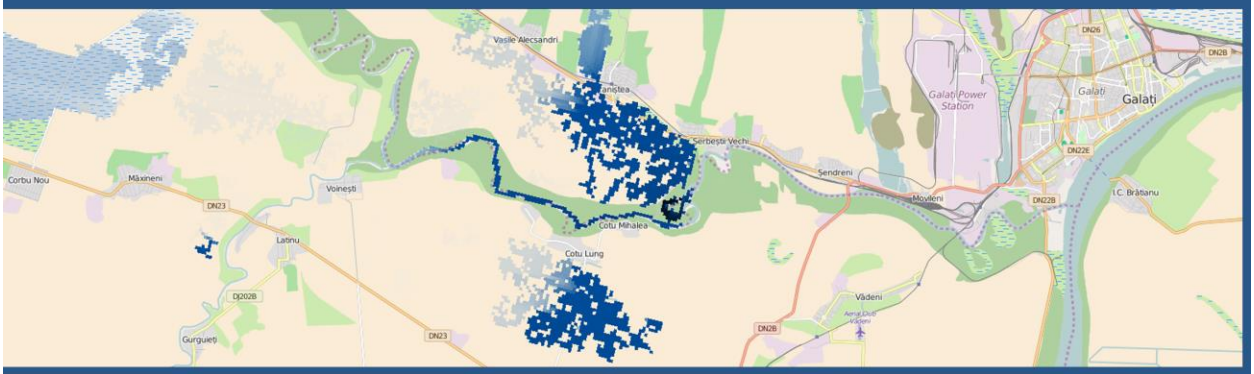
Bogdan Marcel Negrea

Cristian Ionut Iosub



Flood forecasting using open data in the Siret River Basin

AALBORG UNIVERSITY
DENMARK

2016

Aalborg University Copenhagen

A.C. Meyers Vænge 15

2450 Copenhagen SV

Study program and semester: Geoinformatics- 8th semester

Project title: Flood forecasting using open data in the Siret River Basin

Project period: 01.09.2015- 02.06.2016

Supervisor: Henning Sten Hansen

Group members:

Bogdan Marcel Negrea

20142072

Cristian Ionut Iosub

20146135

Number of copies: 2

Number of pages: 105

Number of Appendixes: 11

Abstract: The aim of this project is to create a flooding model able to predict where a flooding event will take place 3-7 days in advance. The model uses as input free accessible data, giving the possibility of being adapted to any region that can only benefit from basic data, like Digital Elevation model, information regarding the soils and land cover. The model also uses free meteorological data that is retrieved in real time. The results are presented using a Web GIS component and displaying the results on top of a map, giving the user a clean and intuitive interface for observing the results. The model is not by any means a very precise tool for flood modelling, working as a prime warning system for future flooding events pinpointing the area in danger, therefore the results need to be considered with caution.

Preface

"All models are wrong, but some might be useful"

Box and Draper (1987), (p. 424)

This report has been written by Bogdan Marcel Negrea and Cristian Ionut Iosub as documentation for their Master Thesis project at the Geo-Informatics Master program at Aalborg University. The project started at the beginning of September 2015 and ended with the hand-in of the report on June 2nd ,2016.

The main aim of this project was to create a flooding model able to predict where a flooding event will take place 3-7 days in advance, using as input free accessible data.

We would like to extend our gratitude to Jay LaPorte of The Dark Sky Company, LLC for giving us insights into how forecast.io obtain and process their data and to GITHUB user DVDME for helping us implement the python wrapper for forecast.io he developed. In addition, we would like to thank our supervisor Henning Sten Hansen who provided us with great support and supervision during the course of the project.

Contents

Preface ii
List of Figures vi
List of Equations viii
List of Tables ix
Abbreviations x
1 Introduction 1
1.1 Background 1
1.2 Research Question and Problem Statement 3
1.3 Study area (General characteristics) 3
1.3.1 Topography and geology 5
1.3.2 Climate 6
1.3.3 Soils 7
1.3.4 Vegetation and land use 8
1.3.5 Hydrography 9
1.4 Testing Area 10
2 Methodology and Theory 12
2.1 Environmental modelling 12
2.1.1 Flood modeling 14
2.2 The hydrological processes in nature 16
2.2.1 The hydrological cycle 16
2.2.2 Components influencing flooding events 17
2.2.3 Flooding Mechanism 19
2.3 Common data inputs for flood modeling 21
2.3.1 Precipitation and snowmelt 21
2.3.2 Digital Elevation Data 22
2.3.3 Soil type and land cover 24
2.3.4 Evapotranspiration and Interception by canopy 25
2.3.5 Remote sensing datasets 26
2.4 Methods used 26
2.4.1 CN Method 26
2.4.2 Day degree method 30
2.4.3 Snow-Melt caused by rainfall 33

2.4.4 Flow accumulation 34
2.4.5 RACA – Rainfall with Cellular Automata method 38
2.5 GIS features for hydrologic modeling 41
2.5.1 GIS and Hydrology tools 41
2.5.2 Interpolation methods in GIS 43
2.5.3 Interpolations comparison 47
3 Data Gathering and Preparation 51
3.1 Necessary data 51
3.2 Data preparation 54
4 Implementation 63
4.1 The perceptual model 63
4.2 The conceptual model 64
4.2.1 Constant data 65
4.2.2 Variable data 65
4.2.3 Rain processes 66
4.2.4 Rain on snow processes 68
4.2.5 RACA Method 69
4.3 The Procedural model 70
4.3.1 Populate precipitation table script 72
4.3.2 Creating precipitation, snow and temperature raster files 74
4.3.3 Calculate runoff from rain 75
4.3.4 Snowmelt 76
4.3.5 Flow Accumulation 78
4.3.6 Rainfall with cellular automata (RACA) script 79
4.4 Model calibration 82
4.5 Model validation 84
4.5.1 Comparing initial abstraction factor with new value 84
4.5.2 Comparing results with other CN (Curve Number) values 88
4.6 Web Service and user interface 90
5 Discussion 96
5.1 Methods used, limitations and lack of timestamp 96
5.2 Uncertainties, errors and parameters 98
5.2.1 Constant data errors 99

5.2.2 Variable data errors 100
5.2.3 Long time calibration 100
5.3 Future improvements 102
6 Conclusion 105
7 Bibliography 106
APPENDIXES 112

List of Figures

Figure 1 - Flooding events and victims/ basin administration(source:www.rowater.ro) 2
Figure 2- Geographical position of the Siret River Basin (Romanescu and Stoleriu 2013) 4
Figure 3 - Flood risk Map for ABA Siret(source: Rowater) 11
Figure 4 - Modeling steps as described by Beven (2014) 15
Figure 5 - Schematic representation of the hydrological cycle 17
Figure 6 - Point representation in vector and raster datasets (ESRI 2013) 22
Figure 7 - Surface represented as raster with different cell sizes (ESRI 2013) 22
Figure 8 - Coordinate systems for raster datasets (ESRI 2013) 23
Figure 9 - Addition operation performed on two raster files 23
Figure 10 - Elevation data represented as raster 24
Figure 11 - Example of tables for evaluation CN from TR-55 document 29
Figure 12- Sink filling- section view 35
Figure 13- The eight neighbors of a cell and the values assigned 36
Figure 14- Assigning the flowing direction to the center cell 36
Figure 15 - Flow accumulation output from ArcGIS and a map for verifying the results 37
Figure 16 - Flow accumulation conceptual output (left) and a conceptual spread of water of the flow accumulation (right) 38
Figure 17 - Moore Neighborhood 39
Figure 18 - IDW Interpolation((ESRI, How IDW Works 2016) 45
Figure 19 - Precipitation data points (red=control, blue=values for interpolation) 48
Figure 20- IDW (left) and Natural neighbor (right) 48
Figure 21- Kriging (left) and Spline (right) 49
Figure 22 - Comparison of retrieved and interpolated precipitation values 49
Figure 23 Average Difference for each interpolation type 50
Figure 24 - Steps for preparing the data in the interest area 55
Figure 25 - Soil type study for establishing working values from TR-55 tables 55
Figure 26 - Land cover data for the whole Europe visualized in ArcGIS 56
Figure 27 - Symbology layer table visualized in ArcGIS 57
Figure 28 - Union operation between the symbology layer and the land cover dataset 57
Figure 29 - Output of the union representing the land cover using the four main classes from CLC 58
Figure 30 - Corresponding CN value assigned for each land cover class 59
Figure 31 - Example of how Equation 4 is applied to the raster having the CN values and the output 59
Figure 32 - Output representation of the raster containing the values for S 60
Figure 33 - Grid of points for which the temperature will be retrieved 60
Figure 34 - Selected area for cellular automata analysis 61
Figure 35 - Model Diagram 64
Figure 36 - Constant data representation 65
Figure 37- Variable data representation 66
Figure 38 - Rain processes representation 66
Figure 39 - Snow melting processes representation 68
Figure 40- Rain on snow processes representation 68
Figure 41- Input raster structure for RACA 69

Figure 42- Overview of the processing chain 71
Figure 43 - Import statement for populate precipitation table script 72
Figure 44 - Column name dictionary 72
Figure 45 - Input of daily data in the precipitation table 73
Figure 46 - Search cursor 73
Figure 47 - Raster creation environment settings 74
Figure 48 - IDW Interpolation 75
Figure 49 - Q calculation using the CN method 75
Figure 50 - Daily snowmelt coefficient increase 76
Figure 51 - Rain on Snow melting 77
Figure 52 - Day Degree method 78
Figure 53 - Flow Accumulation and runoff addition 78
Figure 54 - RACA script modules and environment settings 79
Figure 55 - Beginning of loop and creation of files 80
Figure 56 - Spatial properties of accumulation raster 80
Figure 57 - Creating the water level array 81
Figure 58 - Applying RACA rules 81
Figure 59 - Export of the array back to raster 82
Figure 60 - Precipitation raster for 8th of January 84
Figure 61 - Runoff calculated with a 0.05 abstraction value 85
Figure 62 - Runoff calculated with a 0.2 abstraction value 85
Figure 63 - Runoff from rain with abstraction value of 0.2 86
Figure 64 - Runoff from rain with abstraction value of 0.05 86
Figure 65 - Total runoff with abstraction value of 0.05 87
Figure 66 - Total runoff with abstraction value of 0.2 87
Figure 67 - Newly calculated S values 88
Figure 68 - Runoff from lower CN value 89
Figure 69 - Runoff from higher CN value 89
Figure 70 - *Flow Accumulation with low CN* 90
Figure 71 - Flow Accumulation with higher CN 90
Figure 72 - Document type and language 91
Figure 73 – Map style CSS settings 91
Figure 74- Map div container 92
Figure 75 - Map load function 92
Figure 76 - Potential location function 93
Figure 77 - Webpage header that will contain the buttons 93
Figure 78 - Header CSS style 93
Figure 79 - Button html code 93
Figure 80 - Base map with buttons 94
Figure 81 - Loading of the tile layer 94
Figure 82 - Click event function 95
Figure 83 - Final map with the current day as an overlay 95
Figure 84- Long time calibration steps based on modeling steps described by Beven (2014) 101

List of Equations

Equation 1 - Runoff equation 27
Equation 2 - Initial abstraction formula 28
Equation 3- Runoff equation for 0.2S initial abstraction 28
Equation 4- Maximum recharge capacity of a watershed 28
Equation 5- Degree day equation 31
Equation 6 - Rain on snow equation 33
Equation 7- Rain on snow simplified equation 34
Equation 8 - New proposed formula for CN 83
Equation 9 - New formula for S(recharge capacity) 83
Equation 10 - New runoff formula 83

List of Tables

Table 1-AHP ranking..... 13
Table 2 - Indicators for identifying Process Types at the Regional Scale (Merz R and Bloschl G; 2003) 20
Table 3 - Proposed values for the day degree factor by Wilson(1958) 32

Abbreviations

ABA – Water Basin Administration (ro: Administrația Bazinală de Apa)

AHP - Analytic Hierarchy Process

API – Application Program Interface

ASTER GDEM – Advanced Spaceborne Thermal Emission and Reflection Radiometer Global Digital Elevation Map

CDN - Content Delivery Network

CLC -Corine Land Cover

CSS - Cascading Style Sheets

EEA - European Environment Agency

DEM – Digital Elevation Model

DTM -Digital Terrain Models

ESDAC - European Soil Data Centre

ESDB – European Soil Data Base

ESRI - Environmental Systems Research Institute

GDAL - Geospatial Data Abstraction Library

GIS- Geographical Information System

GLC -Global Land Cover)

HTML – Hyper Text Markup Language

IDW - Inverse Distance Weighting

INSPIRE - Infrastructure for Spatial Information in Europe

I.N.M.H – National Institute of Meteorology and Hydrology (ro: Institutul National de Meteorologie si Hidrologie)

IP – Internet Protocol

JSON - JavaScript Object Notation

OSM – Open Street Map

RACA – Rainfall with Cellular Automata

SRTM – Shuttle Radar Topography Mission

TFL – Tobler's First Law

USDA- United States Department of Agriculture

UTM - Universal Transverse Mercator

WMS – Web Map Service

1 Introduction

1.1 Background

Many studies state that floods are the most common type of natural disaster, affecting many European countries, while the costs related to them are exhibiting a rapid increase (Munich Re Group 2005) (Barredo 2007). Although loss of life has been declining in Europe due to flooding events, the economic implications of such events have risen, with associated costs being in the range of billions of euro for some of the most extreme flooding events (Rotmans, Hulme and Downing 1994) (Mitchell 2003) (Munich Re Group 2005).

Although historical data regarding flooding are not comprehensive across Europe, researchers were not able to distinguish any consistent trend throughout the 20th century (Loster 1999). However, during the late 1990's and the early 2000's, unusual severe flooding events affected a large part of the continent, garnering wide media coverage together with the attention of policy makers both at a national level, as well as at a European level.

According to the European Environment Agency (EEA), in the time period between 1998 and 2009, more than 213 major flooding events occurred in Europe. These events caused 1126 deaths, the displacement of around half a million persons and more than 52 billion Euros in material damage. As a result of these events the European Commission proposed the Directive 2007/60/EC on the assessment and management of flood risks. The directive requires member states to assess all watercourses and coasts at risk from flooding, while taking measures regarding the reduction of these risks. The directive also enforces the right of the public to access such information (European Environment Agency 2011), by requiring all the member states to develop a platform where flooding and water quality information can be easily accessed. This directive, as well as other major efforts accomplished by the members of the states has led to significant steps in implementing a strategy for minimizing the hazard risk that comes with flooding. Despite the progress made be West European countries, East European countries are struggling to implement these measures, due to the lack of infrastructure and poor management (Romanescu and Nistor 2011).

Bearing in mind its large hydrological network and highly varied relief, Romania is facing a high risk of flooding events, with a low capacity of handling them. This risk is further enhanced by the irrational land exploitation and the increasing severity and frequency of extreme weather phenomena such as droughts, storms, heatwaves and cold snaps. Romania's low capacity for handling extreme flooding events is partly

due to the lack of proper flood protection infrastructure as well as the lack of a working early prevention system (Romanescu and Nistor 2011).

Historically, Romania has been the subject of extreme flooding events, with large floods being recorded as far as the 15th century. During recent times, Romania, just like other countries around the world, has witnessed increases in the frequency of extreme flooding events, namely 1 in 100 or 1 in 1000 events happening all across Europe during the early 2000's (Romanescu and Stoleriu 2013).

As previously mentioned, Romania has a complex hydrological network, with 14 big hydrological basins being managed by 11 Water Basin Administration units. One of the biggest is the Siret Basin, which is managed by ABA Siret (Siret Water Basin Administration). Only in the Siret area in the last 30 years, seven important flooding events occurred, resulting in the death of 142 individuals (Figure 1) and measures for future prevention are still not being consistently applied, despite the commitment by the local authorities and the European Commission, both of which invest significant amounts of money into flood prevention and mitigation measures.

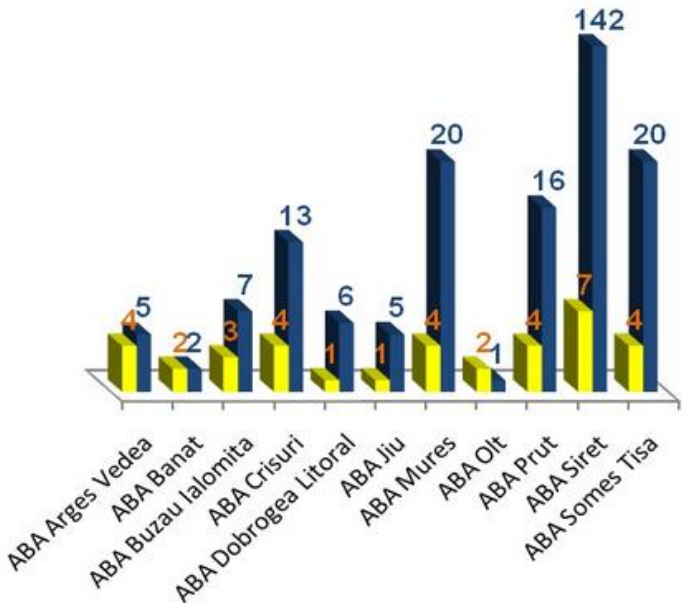


Figure 1 - Flooding events and victims/ basin administration(source:www.rowater.ro)

The predisposition of the Siret Basin to flooding events is noted for the first time in "Description Moldavia", written by the Romanian scholar Dimitrie Cantemir in 1716, where floods, frosts and tidal phenomena are reported along the Siret River and its tributaries.

During 1970, Romania was confronted with catastrophic flooding events, which prompted studies on the general risk phenomena along the Siret River, as well as the consequences of these floods. These studies proposed several protective measures which proved to be ineffective. Moreover, another series of extreme flooding events took place between 2000 and 2005, triggering public debates regarding the effectiveness of prevention measures and the causes of such extreme events (Romanescu and Nistor 2011). In 2008, the last severe flooding event occurred in the northern section of the Siret and Prut River Basins, with costs totaling approximately 2 billion euros in damage (Romanescu and Stoleriu 2013).

1.2 Research Question and Problem Statement

The goal of this project is to create a warning system consisting of a flood model and a Web-GIS application that could provide information to the general public about where the flooding will occur. In order to reach this goal, the following research questions will be answered:

- How to use scarce and free data in order to create a working flooding model for our given area?
- How to access meteorological data in real time and input it as a parameter in our model?
- How to optimize the model in order to obtain the results in the shortest time possible?
- How to make the results of our model accessible to the general public?

1.3 Study area (General characteristics)

The Siret Basin is located in the northeastern part of Romania and is the largest hydrographic basin in the country (Figure 2). The basin has a total surface of 44.811 km^2, most of it being located in Romania and a small part in Ukraine, its springing location. Out of the 42.890 km^2 located inside Romania, only 27.402 km^2 are under the Siret Water Administration supervision, with the rest being divided among the neighboring Water Administration units (ABA Siret(1) 2014), thus allowing better communication with local counties in case of emergencies.

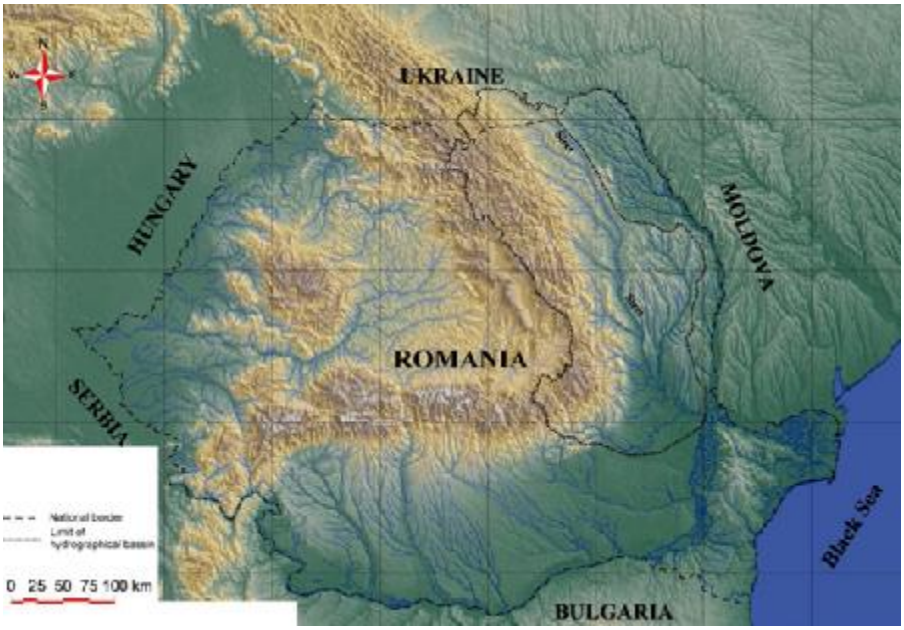


Figure 2- Geographical position of the Siret River Basin (Romanescu and Stoleriu 2013)

The local water administration unit, ABA Siret, was established in 1976 under the direct supervision of the national administration "Romanian Water" or "Apele Romane" as it is known locally, and is regarded as a national focus public institution, benefiting from a high degree of autonomy (ABA Siret(2) 2014).

On the Romanian territory, the Siret hydrographical Basin has a total of 1013 official water streams, counting towards a total 15.175 km in length. While the ABA Siret manages a total of 735 water streams, the rest are managed by other neighboring water administrations.

The Siret River is the most important tributary of the Danube River on the Romanian territory, with a total length of 726 km, with only 559 km being on the Romanian territory. During its long descent through northeastern part of Romania, the river collects most of the water from the Moldavian region, counting upwards to 17% of Romania's total water resources. When joining the Danube, Siret River has a multiannual discharge of approximately 250 m^3/s, thus making it the biggest river in Romania (ABA Siret(1) 2014).

Almost one third of the Siret River Basin comprises of relatively high mountain units, thus explaining the fact that, even though its main water spring is situated on a platform, it still presents the characteristics of the mountain streams. The basin has an elongated dendritic structure, with a high asymmetry coefficient due to the fact that most of the tributaries are on the west side. Furthermore, on the east side, the most important tributary is Barlad River, being important only to its extent (16.5% of the total basin

surface) and not because of the water quantity (5%). Another important aspect is the average height, the Siret Basin having an average height of 515 meters, while the average height of other basins in Romanian is 507 meters.

Regarding the administrative territory, the Siret Basin covers the surface of the following counties: Neamţ, Bacău, Vrancea, Botoşani, Iaşi, Galaţi, Buzău, Covasna, Harghita, Bistriţa Năsăud, Maramureş. The Basin territory has a population of approximately 2.6 million people, out of which 1.5 million live in the urban areas, thus leading to a population density of 94.13 per km^2, this being slightly higher than the national average density of 90 per km^2 (ABA Siret(1) 2014).

1.3.1 **Topography and geology**

Because of its significant surface, the Siret hydrological basin has a great variety of different landforms, each having a big influence on the hydrology of the basin. For example, the high mountain areas are characterized by large discharge values, while lower plateau landforms are characterized by heavy droughts (Munteanu and Tatu 2003).

The basin overlaps 3 major geological units: the Carpathians, the Subcarpathians and the plateaus (Moldavian Platform and Barlad Depression). The Carpathians and Subcarpathians have a similar geosyncline structure comprised of hard rocks, resulting in a high fragmentation density environment and they are especially prone to flooding events. The plateaus have a monocline or horizontal structures and are characterized by easily erodible, soft rocks (Talaba, et al.2005).

Each major landform is heavily individualized, decreasing in height if you travel from West to East as follows:

- **Inner Eastern Carpathians**: they are further subdivided into smaller sections, each having their own characteristics.
    - The volcanic section is composed from hard volcanic rocks, such as andesite and basalt. These rocks are resistant to erosion and, as a result, they have tall landforms, such as Caliman Mountains (2102 m). These landforms are mostly drained by the Bistrita River and its tributaries.
    - On the East side of the volcanic section of the Carpathians was formed the crystalline Mesozoic zone. The formation of this continuous zone results in the presence of high landforms, such as Rodnei Mountains (2305m). When

looking further east of the crystalline zone one can find karst landforms, les tall but with a high degree of fragmentation. Similar to the previous section, this one is being drained by the Bistrita River as well.

- The flysch zone is situated on the outer periphery of the mountain chain. It is the lowest of the mountain areas with the highest peak being in the Ceahlau massif with a height of 1904 meters.

- **The Subcarpathians:** Positioned further East from the Charpathians, the Subcarpathians are composed of slightly softer rocks such as argyle and sandstone. At altitudes higher than 900 m, there can be found harder patches of rocks. (Munteanu and Tatu 2003).
- **The Moldavian Platform:** occupies an important surface of the entire Siret River Basin (more than 30%). It has a typical platform structure, covering the area from the Subcarpathians until the Prut River and slightly decreasing in height from West to East. Landslides occur quite frequently in this area, due to the fact that the Moldavian Platform is composed of soft rocks, such as argyle and sandstone. This unit is also heavily fragmented by a high density network of valleys with asymmetrical profiles specific to this region (Munteanu and Tatu 2003).
- **The Lower Siret Plain:** located in the Barlad Depression geological unit and in the southern part of the River Basin, it represents the lowest section. Referenced to the Black Sea, there are areas where the altitude reaches only 7m above the sea level.

1.3.2 **Climate**

The climate in the Siret Basin is influenced by altitudinal zonality (because of the decrease in altitude from west to east) and latitudinal zonality (due to the basin's latitudinal extent of approximately 3 degrees.)
Average thermal amplitude allows for the characterization of the Siret River Basin area as a region with high thermal amplitude, the values being around 24-25 ^{O}C. Moreover, as a result of the interaction between topography and climate, the air temperatures can reach above 25 ^{O}C, while occasionally reaching above 40 ^{O}C during summer and as low as -20 ^{O}C up to -35 ^{O}C during winter (Talaba, et al. 2005). This factor combined with landforms created the main climate characteristics of the basin. As a result, clear distinctions can be made between mountain climate, platform climate and plain climate, all having specific impacts in regard to the frequency and severity of the flooding events (ABA Siret(1) 2014).

- *The mountain climate* is characterized by low temperatures and wind predominately from the western and northwestern continental Europe. In regard to flooding, it is important to note

that, even though precipitations have high values (800 – 1200l/m^2), they are constant. Because of the low amplitude of temperatures, the snow layer is present for most of the year.

- *The platform climate* is characterized by a higher difference in temperatures, especially close to the Subcarpathians, with an average temperature range from 7-10^{O}C. A high risk for flooding is caused by distribution of precipitations, which is extremely inconstant over the course of the year. As a result, despite the lower values (500 - 800l/m^2) when compared to mountain climate, this climate zone presents a much higher risk of flooding.
- *The plain climate* holds little relevance, considering that very few streams occur in the areas with these characteristics. Average temperatures are around 10-11^{O}C and precipitations vary within the 400 - 500l/m^2 range. (ABA Siret(1) 2014)

The high torrentiality of the Siret Basin is caused by its position of the basin in a geographical area, where Atlantic air masses, wet and with average temperatures, meet with eastern European air masses, more dry and with a higher thermic contrast. The two air fronts meet in the area around the Subcarpathians, and because of the mountain characteristics of the area, they persist for a longer period of time, resulting in a high frequency of flooding events (ABA Siret(2) 2014).

1.3.3 **Soils**

Soils, along with vegetation, are a natural element that heavily influences the frequency and severity of flooding events. Characteristics such as soil type, soil depth, water absorption capacity, are of high importance when modeling such events. Along the studied River Basin, soils present a high variability, but follow the principle of vertical zonality. Therefore, they can be divided based on altitude (ABA Siret(2) 2014) as presented below.

In the mountain area there are podzols at high altitude and cambisoils at medium and low mountain areas. On one hand, Podzols have a thick profile and a high content of sand, but because of their low coverage, have a low hydrological impact. On the other hand, Cambisoils cover more than 80% of the mountain areas, being extremely fertile soils with a fairly high sand content, and many of the areas covered by such soils being used for agriculture. In the ESDB dataset, cambisols are equivalent to a medium soil type, while podzols have a small extent and are not seen in the database over our Basin's territory.

In the Subcarpathian area, there are different types of soils, all interlaced in small patches, with the most dominant types being clay based. Because of the high clay contents, the soils in the Subcarpathian regions have a low permeability factor, which, combined with heavy deforestation and high precipitation values, make this zone vulnerable to extreme flooding events. (Talaba et all 2005)

On the hills and plateaus areas more clay based soils are present. Despite the similar content with the Subcarpathian area, flooding is less frequent here because of lower slope values.

The plain areas are composed of chernozem with varied degrees of permeability and uses. Because of the high loess content, these soils tend to be fine, light and workable. In the ESDB database they would correspond either to a medium fine, or to a fine soil, depending on it's the exact constitution. Similar to the plateaus, the discharge rate is influenced by the low slope of the area (Talaba et all 2005).

1.3.4 **Vegetation and land use**

Along the basin territory, there is a variety of vegetation types which influences discharge. Similarly, to other characteristics of the Basin, vegetation is influenced by the altitude and organized in zones based on altitude.

In the high mountain areas, vast coniferous forests are present, which regulate the discharge rate. Whereas the hard rocks, low permeability, high fragmentation and slope values determine a high discharge speed, the tree crowns retain quite a high water quantity. As a result of the characteristics presented, flash flooding occurs frequently in mountain area, but they are limited in area affected and damages.

In lower mountain areas and plateaus areas hardwood forests are common, regulating the discharge rate in a similar manner to coniferous forests, but having a smaller coverage. In lower mountain areas hardwood forests still have a significant coverage, but in plateaus they are organized in small patches. This coupled with the high argyle contents of the soils cause severe flooding. It is important to note that in plateau areas, deforestation is a major concern, and one of the reasons hardwood forests are extremely patchy in composition (ABA Siret(2) 2014).

In the eastern and south-eastern part of the Basin, vegetation has a steppe characteristic, represented by extremely isolated tree patches being with almost no effect on the discharge rate and water absorption.

In this area, agricultural lands represent a large percentage of the territory, the area being one of the main agricultural areas in Romania (ABA Siret(1) 2014).

1.3.5 **Hydrography**

The physical characteristics of the stream network, with its southern flow, and East to West asymmetry reflects the other characteristics of the basin, such as geology and topography, which follow the same structure. The hydrological characteristics are heavily influenced by the climate, the Siret River Basin being at the confluence of several important high and low pressure areas such as the Azores High, Icelandic Low and Siberian High. (Talaba, et al. 2005) (Budui 2002).

The stream network along the Siret River Basin is extremely well developed with 15 175 Km on Romanian territory, with 9807 km being administered by ABA Siret and the rest being administered by neighboring basin administration units (I.N.M.H 1992). The stream density is significantly higher in the mountain areas, reaching a density of 1-2 km/km^2 and even above 2 km/km^2 in small areas. The stream density diminishes as one descends in height, reaching 0.8-1.5 km/km^2 in the plateau areas and 0.2-0.5 km/km^2 in the plain areas. The average stream density along the entire basin is 0.35 km/km^2, this being slightly higher than the national average of 0.328 km/km^2 (ABA Siret(2) 2014)

The main source of water for the streams is represented by precipitation, with values ranging between 75% and 80 %. Out of mentioned percentages, snowmelt accounts for 28% in the NW and up to 52% in the S and SE (Talaba, et al. 2005).

The Siret River Basin is extremely asymmetric with all its main tributaries, having more than 70% of the total surface, springing from the western side, while the eastern side is represented only by Barlad as a main tributary. All western tributaries basins have various shapes and sizes, which, in connection to the other complex and various factors previously discussed, make hydrological modeling extremely difficult.

The most important water streams are:

- Suceava River with a length of 173 Km and a surface of 2641 Km^2 on Romanian territory.
- Moldova River with a length of 213 Km and a basin surface of 4299 Km^2.
- Bistrita River with a length of 283 Km and a basin surface of 7039 Km^2. It is the largest tributary to the Siret river
- Trotus River with a length of 162 Km and a basin surface of 4456 Km^2.

- Putna River with a length of 153 Km and a basin surface of 2480 Km^2.
- Ramnicu Sarat River with a length of 137 Km and a basin surface of 1063 Km^2 (ABA Siret(2) 2014).

Along the river basin there are 31 lakes being used for various purposes such as flood mitigation, energy generation, irrigation and fishing with a total volume of 1 847 632 million cubic meters ((Ministerul Mediului si Dezvoltarii Durabile 2002)

1.4 **Testing Area**

Bearing in mind the subsection presented above, it can be drawn that the Siret River Basin represents a large landform with a large percentage on the Romanian territory. These characteristics expose several issues when it comes to modelling such a large area.

Therefore, the issue of the uniformity and the availability of data appears. In order for any model to work, it needs to have consistent data covering the entire the area. If consistent data is available, the next issue is that of the complexity of the model. For large areas, the model needs to be as simple as possible in order to allow it to run in real time (Christie, et al. 2011). While our model is fairly simple because of scarce data, due to the low hardware capabilities and almost no experience in program optimization, we chose to focus on only a certain section within the lower Siret River Basin, section know to be predisposed to severe flooding. Studies showed that this region is affected by flooding events almost every year (Romanescu and Nistor 2011), extreme events taking place from time to time.

The latest extreme flooding event that occurred in the lower Siret River Basin area was in 2005. In order to understand the severity of the floods in 2005 we can compare the discharge for that period to the average yearly discharge. As mentioned above, the average discharge of the Siret River is 250 m^3/s and, during the flooding event, it rose up to approximately 4650 m^3/s. The affected area represented 58 323 939 ha, out of which 58.54% was arable land and 4.91% was built up area. The flood also destroyed over 10 000 houses and led to the death of 24 persons and the loss of several thousand domestic animals ((Romanescu and Nistor 2011).

Figure 3 presents a flood risk map developed by ABA Siret, and in the areas outline in red you can see the areas under risk. The area in SE is almost entirely under the risk of flooding. The exact area will be presented later during this paper, in Chapter 3.3.2, as other factors were considered when selecting it.

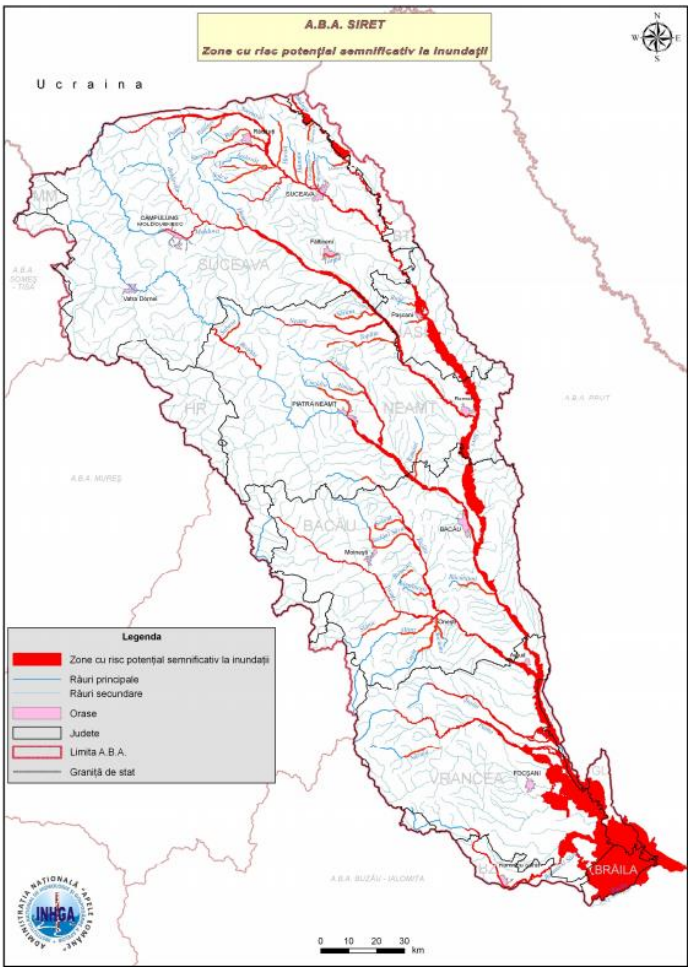


Figure 3 - Flood risk Map for ABA Siret(source: Rowater)

2 Methodology and Theory

This chapter provides the necessary theory for understanding the following sections and the implementation of the model. It begins with a broad presentation of environmental modeling and continues with a more in depth look into flood modeling and the hydrological cycle. The chapter also presents the necessary data and the importance of each factor considered for the model. The last two subchapters present the integration of hydrological modeling within the modern GIS environment and the tools used for modeling.

2.1 Environmental modelling

Modeling in itself represents a manner of simplifying and understanding reality in order to discover a solution to a problem or to gather a better understanding of the processes that are happening in the real world. Modeling can take many forms, from very qualitative to very quantitative, while the most basic one is the course of analyzing a problem systematically (Jakeman,et al. 2008). On one hand, a qualitative model is a display of the most important mechanisms of a system visualized, by using a simplification of its components, while on the other hand, a quantitative model is mathematically focused on representing the components of a system, whilst being usually based on complex formulas and not on visual representation (Bratko 2009). Jakeman et al. (2006) suggest a number of features that should be considered when modeling, as following: (1) the statement should be formulated clearly; (2) developing a study of the expected results and the premises for achieving the results; (3) the testing should be transparent and the errors should be reported for the whole model and not for the individual components; (4) for an increasing level of trust in the model, the problems experienced should be openly discussed in order for a solution to be found.

In contrast with the features presented above that are to be considered, Ravetz (1977) stated that the validation of the individual process is more important than the product and that usually “the quality of a model is assured only by its quality of its production”(p. 9). By combining these statements, the conclusion that can be withdrawn for them suggests that it is equally important for each step to be validated together with the final product, as well as the highest level of quality possible should be achieved for each step or process.

An important part of modeling reality is the number of factors to be considered. It is impossible to account for all the factors and processes that occur at a given time or over a period of time in nature. However, even more difficult than the number of factors is the selection of the factors that would have the highest influence on the model. For example, a number of ten factors would describe all the processes that occur into the product that is to be modeled. Some of the processes are easier to quantify and represent, while others require more time and resources. In the same time, there are processes that simulate the reality alone in a greater proportion than other. Precipitation as a process, is far more important than evapotranspiration for flood modeling for example, and simulates alone the reality better than evapotranspiration only. Using the same process of thinking, Saaty (1987) introduces the Analytic Hierarchy Process (AHP). AHP is a "general theory of measurement" (p. 161) and it works by creating a hierarchy scale based on the level of importance for each factor that is considered. AHP is mostly used for decision making together with methods such as Multi Criteria Decision Analysis where to different factors, different weights are assigned. The principles of studying the level of importance for each factor can be used when choosing the data for a model. Table 1 below presents how factors are sorted by the intensity of importance.

Importance Intensity	**Definition**	**Explanation**
1	Equally important	Factors that contribute to the end result, but alone would not represent reality in the proportion needed.
3	Somehow more important	Factors that alone represents the reality in greater proportion
5	Much more important	Factors that are significantly important and alone would give an broad overview of the reality
7	Very much more important	Factors that are very important in combination with others
9	Absolutely more important	Evidence of this factors being the most important exists
2,4,6,8	Intermediate values	When intermediate levels of importance are needed

Table 1-AHP ranking

Additionally, choosing the accurate factors is extremely important and it depends on the purpose of the modeling. For example, a model meant to simulate the erosion due to sea level rise and act like a screening tool pinpointing the areas in danger will account for less and different data compared to a model meant to simulate the erosion on a specific stretch of coast, where a higher number of factors with detailing specifications are necessary.

Nowadays, models are usually based on computer software, therefore the processes simulated, also called computations, are becoming more and more available to modelers and even to non-modelers. Modeling became a necessary way of understanding a process with the help of technology (Jakeman, et al. 2006).

2.1.1 **Flood modeling**

In his book "Rainfall-Runoff Modeling the Primer", perhaps one of the most comprehensive works addressed to hydrologist and modelers from the past 20 years, Beven (2014) gives an insight on how the modeling process for hydrology should be structured, especially for Rainfall-Runoff. Additionally, he starts the modeling process with an extra stage compared with the rest of the processes found in hydrology books. Below, Figure 4 shows the method he proposes for starting the modeling process. The extra stage in this case is *The Perceptual Model*. This step is necessary to be individual and it refers to the perceptions of the modeler, the training the individual has, together with the knowledge cumulated from the literature reviewed and the experience gathered. Therefore, inevitably the perpetual models developed by different individuals will be distinctive regarding the same area of analysis.

The second step is represented by *The Conceptual model*. In this division, all the thoughts of the modeler from the previous step need to be translated into a mathematical language, while all the processes to be considered for the modeling are inevitably simplifications of reality. The level of simplification depends on the purpose of the model and can vary from very exact reproduction of processes to gross simplifications that are considered accurate enough to model or predict reality. Hydrology modeling works with all kind of equation, varying from simple ones, for instance water balance equation, up to very complex ones, for instance nonlinear partial differential equations. The equation complexity depends again on the purpose of the model and the expected prediction.

In *The Procedural model* stage, the modeler uses the mathematical representation of the process and further translates them into a programing language used by the modeling software. Before the code can

be run and the results analyzed, an extra step is necessary under the form of *Model Calibration*. In this division, the numerous variables from the equations represented in *The Conceptual stage* are calibrated for the particular area of study. This stage is also called a parameter calibration stage. An example of parameter would be soil moisture, precipitation value and most of the meteorological values. The most used calibration technique is adjusting the values of the equations so the prediction represents reality in the greatest approach possible.

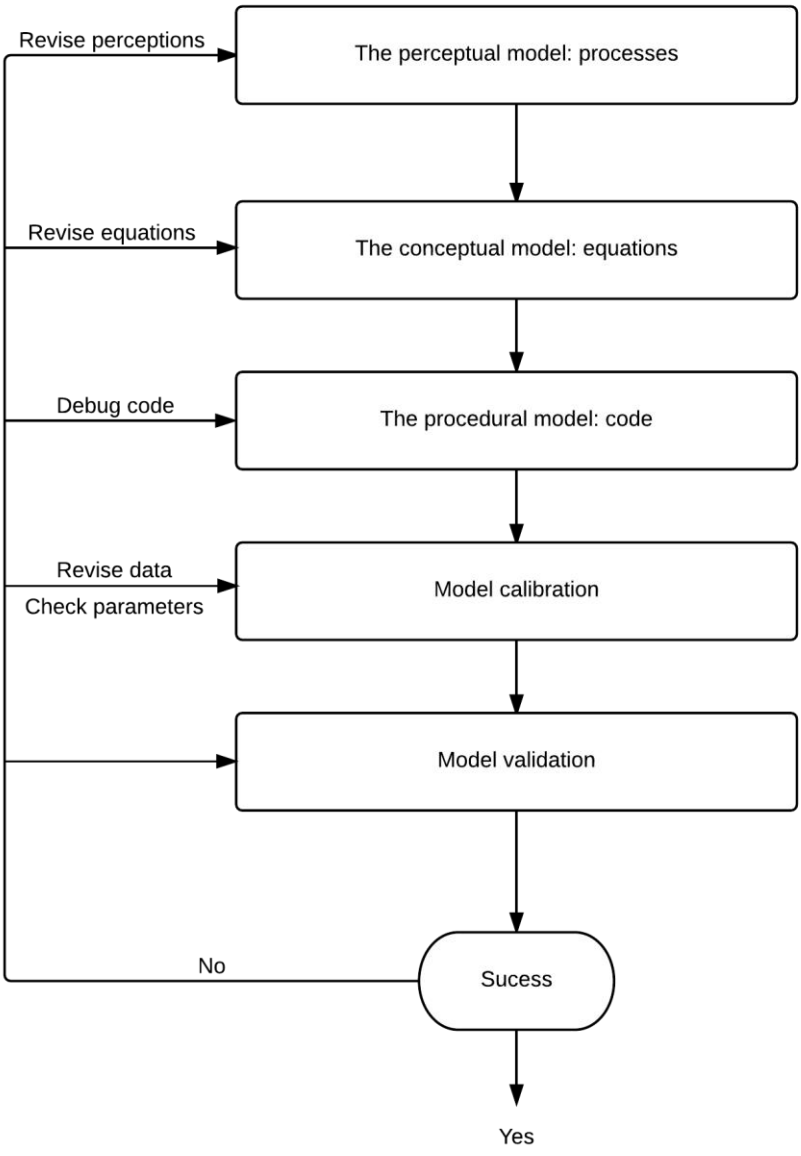


Figure 4 - Modeling steps as described by Beven (2014)

Having the parameters values calibrated, the model can run as a simulation, and a validation or evaluation needs to be made in order to compare the results or the prediction with reality. In theory, based on the evaluation stage, parameter values can be changed in order to get a closer fit. This was proven to be one of the most difficult stages due to the large number of values that can be changed and the high number of good fits. By having available more than one feasible model, it turned out to be extremely hard to choose the right one and to validate an individual one only.

2.2 The hydrological processes in nature

Understanding the hydrological processes that are happening in the real world and, especially under the visible surface of the earth, is limited to our knowledge. Moreover, also our capacity of understanding how these processes are interconnected is fairly limited, the only known area being the one that is tested using a probe, anything else being pure predictions (Beven, Rainfall-Runoff Modeling The Primer Second Edition 2014). The following pages will attempt at specifying and describing the processes that are involved in the flooding phenomenon. It is highly important to understand that each hydrologist might consider some of the processes more important than others and their description might be slightly different from one to another. The paper combines descriptions from some of the most comprehensive volumes written so far.

2.2.1 The hydrological cycle

The water cycle, also called the hydrological cycle, defines the relation between a number of elements such as oceans, land, soil, atmosphere and groundwater from a water exchanging point of view. Figure 5 bellow presents the most important processes that take place in the hydrological cycle in a very schematic representation.

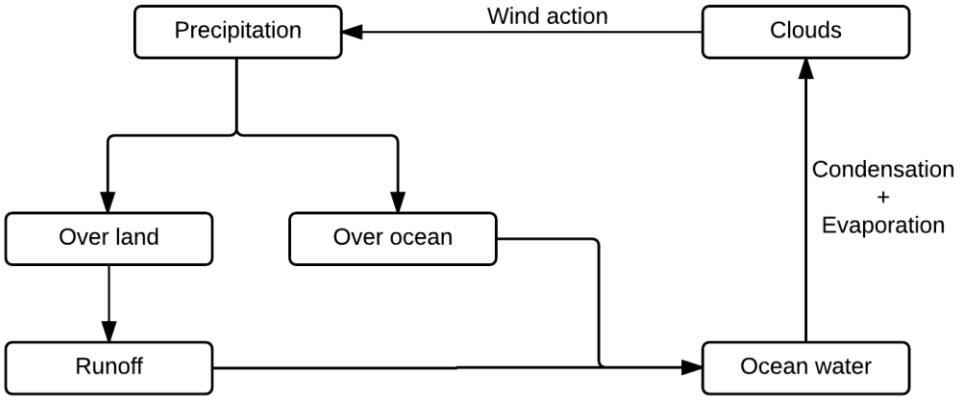


Figure 5 - Schematic representation of the hydrological cycle

When trying to describe the water cycle as a system, there is no starting or ending point, the water molecules being displaced at all time. The two dominant processes in the hydrological cycle are *precipitation* and *evaporation*. Ocean water evaporates and reaches the atmosphere where it is condensed into clouds. The clouds are transported by the wind movements and are eventually precipitated over land or ocean. Approximately 90% of the water evaporated returns into the ocean and approximately 10% reaches the land. The average time of water molecule suspension into the air is 9.2 days. The water that reaches the land as precipitation becomes runoff, and is stored in ground reservoirs, accumulated in snow packs, glaciers, lakes or wetlands and eventually reaches the ocean again, completing the cycle (Moustafa 1992).

2.2.2 **Components influencing flooding events**

The processes of the hydrological cycle that have the highest impact in flooding events are precipitation, snowmelt, evaporation, whereas the physical components of the natural environment that have a high contribution are the type of the soils and land cover (Beven, 2014).

The first dominant hydrological process, *precipitation*, which is the process where the water from the atmosphere reaches the surface of the earth and becomes runoff. Precipitation is formed due to condensation, a process through which water changes its state from vapor to liquid or solid state (Langebein, et al. 1973). Precipitation can reach the surface of the earth as rainfall, snow, hail or sleet, not being a uniform process or a uniform variable that needs to go into the model (USDA 1999). Being very common for the precipitation intensity to vary from place to place, even for a very small area, one must

account for this patterns when modeling and especially when using interpolation methods (Beven, 2014). The most important effect of precipitation is runoff, being defined as a measure of volume and representing the amount of rain that meets the surface of the earth (Solomon 2005). Runoff generation study shows how much water the stream will receive after a rainfall or storm (Beven, 2014). Based on the calculations of the methods presented above, in the hydrology training documents from USDA (United States Department of Agriculture) there can be differentiated into four types of runoff:

- *Channel runoff* - represents the precipitation that reaches the rivers, streams, lakes or wetlands directly, which usually can be considered negligible.
- *Surface runoff* - is created when the soil absorption capacity is lower than the quantity of water received from precipitation. In this case, the soil becomes saturated, no more water can infiltrate and the excess precipitation runs down as surface runoff.
- *Subsurface runoff or subsurface flow* - the water that infiltrates into the soil does not remain in the same place, but presents a movement. It reappears during or after the event as springs or surface runoff.
- *Base flow* - defined as a constant flow that exists before during and after an event. (USDA 1999)

Additionally, a second important process that has a big contribution with regard to flooding events is *snowmelt*, whose rates vary with elevation, air temperature, vegetation cover and solar radiation. When the air temperature is negative, precipitation freezes and touches the earth surface as snow. If the air temperature at the soil level is not low enough, snow does not accumulate, part of it infiltrating into the soil, or if the soil is oversaturated, becoming instant surface runoff. Furthermore, if the air temperature is low and the soil is frozen, snow does not transform into runoff immediately, but gets accumulated as a layer, which can be gathered for months at a time (Beven, Rainfall-Runoff Modeling The Primer Second Edition 2014). When modeling snowmelt, one needs to account for snowmelt runoff as being an effect of precipitation that can date back months (Jonathan, et al. 2014). Snowmelting normally occurs during a melting season, being important for the individual to understand that the melting rates are not constant and comprehend as well that the beginning of the season provides a low melting rate that increases towards the end of the season (Rango and Martinec 1995).

The third important process is *evaporation* which represents the process of the water changing its state to vapors. Evaporation is a term used in hydrology for describing vaporization, a process of turning the water into vapors below boiling point, on which evaporation usually works (Langebein, et al. 1973). Moreover, a form of evaporation is represented by transpiration, this being the process through which

plants eliminate water back into the air. Transpiration from vegetation together with evaporation from soil is called evapotranspiration (USDA 1986).

Land cover is an important component of the natural environment that has a high contribution into flood modeling. On one hand, the soil saturation is usually exceeded in areas with bare soils, while on the other hand, it rarely occurs when dense vegetation is present. Once water drops touch a bare soil surface, the soil particles are rearranged and a layer is formed leading again to instant runoff. This can be controlled by increasing the percentage of vegetation in the affected areas (Annika, et al. 2013). Likewise, another part of the precipitation can be intercepted by canopy, which represents the branches and leaves of the vegetation, forming a continuous layer that makes it almost impossible for light rain to pass through, this process being called interception. Wind and air can also have an influence when understanding rainfall process. Llorens (1997) states that unsaturated air, combined with the canopy and the wind can reduce the amount of water touching the earth surface with up to 30%. The study was made on a period of over two years and includes 185 events showing how vegetation and land cover play a crucial role in hydrological modeling and especially in runoff modeling.

Soils type is another important component that can dictate how the water infiltrates and how much becomes runoff. The porosity of the soil and the size of the component particle have an impact on infiltration rates (Beven, 2014). Soil can only receive water until it becomes full, saturated, the excess water becoming surface runoff. The components of the soil have a big impact on how the water infiltrates, while the types of the soil can vary in consistency from sand to gravel, which are very permissive in letting the water pass, to types of soils, having a big percentage of clay in their composition. In this case, water cannot infiltrate and the probability of surface runoff increases (USDA 1986).

2.2.3 **Flooding Mechanism**

Climate changes in the recent years have increased the mean temperature in Europe by two degrees and have increased the precipitations by around 20%. The increased precipitation values results in a higher risk regarding flooding (Werritty, et al. 2002).

Flooding is described as a result of heavy rainfall or snow melting that is greater than the capacity of soil to absorb the water, evapotranspiration and rivers flow capacity. This results in the inundation of the usually dry land in the proximity of the rivers, streams or coastal areas. The areas in the proximity of a

river are known as floodplains. The rainfalls and the snowmelt are considered to be the biggest cause for flooding events (Merz and Bloschl 2003). Table 2 presents a proposed classification of floods type.

Type of flooding	**Long-Rain**	**Short-Rain**	**Flash-Floods**	**Rain on Snow**	**Snowmelt**
Timing	no seasonality	no seasonality	mainly summer or late summer	changes between cold-warm periods	spring to summer-snowmelt season
Storm duration	>1 day	<1 day	<90 min / high intensity	moderate to high intensity	rainfall unimportant
Rainfall depths, snowmelt	substantial	moderate to substantial	small to moderate	moderate to high intensity	snowmelt
Runoff response	slow	fast	flashy	fast or slow	medium or slow
Spatial coherence	large: $>10^4 km^2$	local or regional	limited: <30 km^2	areas covered by snow	spatial extend of floods
Catchment state	wet	wet	dry or wet	snow cover	snow cover

Table 2 - Indicators for identifying Process Types at the Regional Scale (Merz R and Bloschl G; 2003)

Long rain floods are the effect of the rainfall over an extended period of time, varying from days to weeks. Even by having a low intensity rain fall, during a long period of time, if the maximum soil capacity of absorbing water is reached, the remaining will become runoff and eventually turn into a flood event. Short rain floods take place when a medium to high rainfall event occurs, but if the surface is still saturated from previous events, the catchment is being prone to exceed its saturation limit. The duration of the rainfall for short rain floods may vary from several hours to one day. Flash floods are characterized by a high precipitation volume in a short period of time, usually less than a couple of hours. The flooding might occur even if the soil is relatively dry, taking place in small catchments. Snowmelt floods usually occur in the spring period, when the temperature rises in a little amount of time and snow starts to melt rapidly. Melting snow becomes runoff and flooding events occur if the soil infiltration capacity is exceeded. Rain falling on snow also contributes to flooding in the mountain areas, this usually happening during springtime, due to the increased temperature; snow becomes regular rain and big amounts of runoff can occur, when in contact with the snow cover layer (Merz and Bloschl 2003). Another, rarer cause of

flooding, is the dams and other buildings failure. The break can cause a sudden burst of water that can quickly become a flood.

2.3 Common data inputs for flood modeling

2.3.1 Precipitation and snowmelt

The quality of a model designed to estimate runoff is dependent on the input data and especially on the precipitation data. Having poor input data translates into average results as well as not accurate predictions. Obtaining precipitation data for hydrological modeling used to depend on the ground rain gauges that were manually operated in order to read the values for the precipitation. In recent years, radar technology for measuring rainfall became more and more available and currently is used all over Europe and USA for meteorological prediction. Rainfall radar normally works as an interpolation method and has a resolution between two and four km. If the radar is calibrated and accounts for factors such as wind, altitude of reading, drop size and distribution, the quality of the readings becomes accurate and the data is suitable for being used in modeling (Beven, Rainfall-Runoff Modeling The Primer Second Edition 2014).

Nowadays, precipitation data is available almost cost-free from a variety of sources such as big companies specialized in weather forecasts, for instance BBC Weather or AccuWeather or small providers of meteorological products such as Open Weather Map or Forecast.IO. The last mentioned provides free data, from precipitation, to precipitation type, wind speed and up to sunset time. The user usually is allowed to do a number of requests free of charge, for point data such as coordinates and retrieve the desired information. If the number of request exceeds the number of free calls, a very small fee is applied for each new request. The companies receive the data from more than one source, in order to improve the accuracy of the predictions (ForecastIO 2015).

Data regarding the level of snow is more difficult to compute or predict, usually requiring traditional measuring methods. The available snow level data is registered at ski resorts or scarce meteorological stations, most of the times the average modeler not having access to it. As mentioned above, in order to create a snow-melting model, it might be necessary to account for precipitation value that has fallen months before.

2.3.2 **Digital Elevation Data**

When working with representing topographic entities from the real 3D world to a 2D surface, there are two formats commonly used today: vector or raster. A vector representation of data uses points, lines and polygons for representing the real world. This approach is suitable for entities such as limits, borders, parcels, streets, streams or similar discrete entities. A raster representation of data uses a regular grid of cells composed of cells for representing 3D objects. In a raster dataset, each cell has a x, y coordinate and a value, named cell size. This approach is suitable for surfaces that are continuous in nature, such an elevation map, satellite image or a surface.

Bellow, a representation of different entities are presented in both formats, vector and raster.

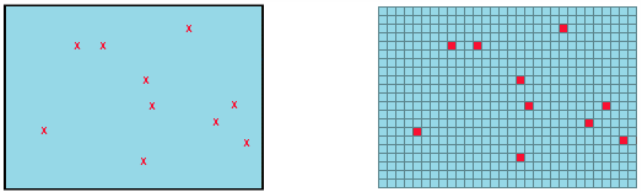


Figure 6 - Point representation in vector and raster datasets (ESRI 2013)

In Figure 6 a conventional point data can be represented as a vector characterized by normal points with x, y coordinates or a cell on the grid. The point, which in theory does not have a dimension, in the raster dataset is considered to have the same dimension as the cell. Having a grid with smaller cell size, increases what is called “raster resolution” and the accuracy of the representation is augmented. Using the same principles, other surfaces from the real world can be represented as seen below in Figure 7.

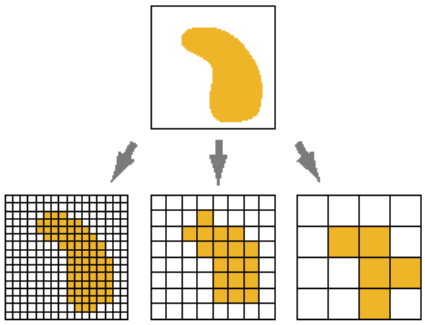


Figure 7 - Surface represented as raster with different cell sizes (ESRI 2013)

The size and the number of the cells dictate the accuracy of the represented data as seen in Figure 7. A higher number of cells translate into a greater accuracy, as well as in a longer processing time. Each cell has its own coordinates in a raster grid, the left lower corner being considered as coordinate point for

identifying the cell. In theory, any arbitrary coordinate system can be chosen, but a Cartesian coordinate system or a projected coordinate system is desired in concordance with the most used software available as seen in Figure 8.

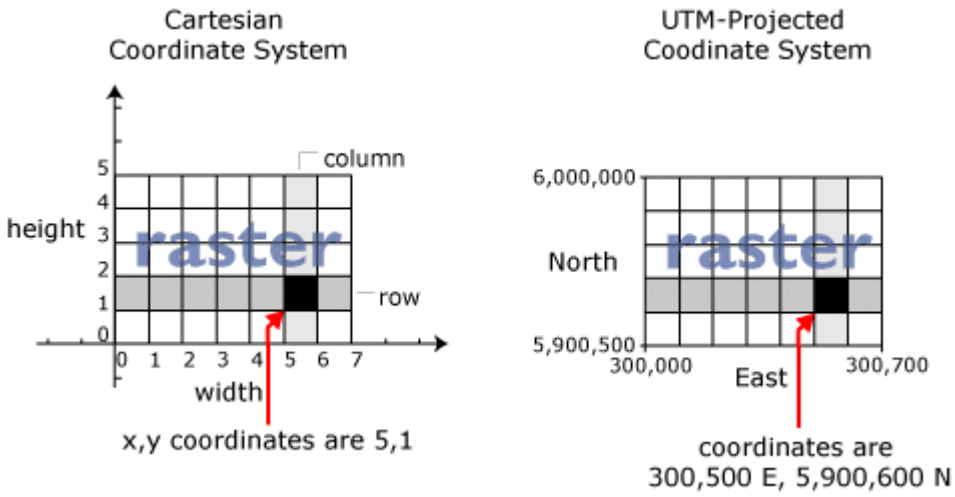


Figure 8 - Coordinate systems for raster datasets (ESRI 2013)

In a Cartesian coordinate system, the rows and column number starts with value 0,0 represented by the first lower left corner cell. In a Projected coordinate system, the left lower corner has a value of the location defined by the real object. In Figure 8, the system used is UTM/ Projected coordinate System and the raster starts with values for east as 300,500 and north as 5,900,600, each cell having a size of 100 by 100 m which means that it represents a surface of 100 m^2 from the real world.

An important property that raster data type has is the ability to perform mathematical function on the values held in the raster's cell as seen in Figure 9.

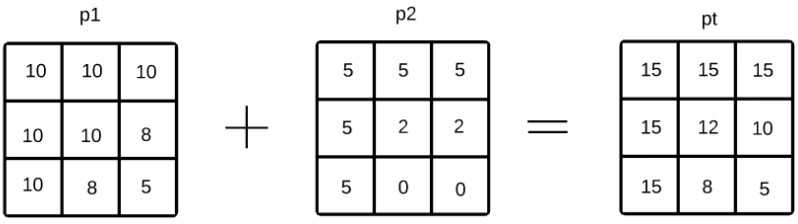


Figure 9 - Addition operation performed on two raster files

Operations can be performed if the raster files have the same number of cells and columns and the same coordinate system. This can be extremely useful if, for example, records of precipitation values are held

for each day in raster datasets, each cell representing an area, and the total amount of precipitation for one week needs to be computed. Depending on the software environment, different operations can be performed, from simple tasks to complex mathematical functions.

In hydrological modeling, the raster data is used as a base for generating a model of the surface to be modeled and is called DEM (Digital Elevation Model) or DTM (Digital Terrain Models). Florisnky (2011) defines a DEM as a" function that describes a topographical surface as a measured or computed values for each grid nodes" (p. 31). There are numerous ways for generating a DEM, the most common being topographic and GPS surveys, laser techniques such as LIDAR, echo measurement, aerial or remote sensing. (Florinsky 2011).

Figure 10 - Elevation data represented as raster

Figure 10 shows how elevation data is represented, each cell representing an area and the mean elevation of the area is assigned to the correspondent cell, a higher number of cells resulting in an increased accuracy. The values of the cell can be positive or negative, Integer or floating point or no data. (ESRI 2015)

2.3.3 Soil type and land cover

Soil is an important factor for hydrology modeling, being involved in almost all the processes that take place over the surface land. While being, in connection with the surface runoff, the subsurface runoff, and the base flow, it provides water for vegetation and its type dictates the rates of water infiltration. (Wharen, et al. 2015). Therefore, it is important to obtain accurate information about the type of the soil, in order for the hydrological model to present proper results. The availability of the data has increased over the years, due to the new soil mapping techniques and the development of GIS techniques (Zhu 2002).

Further on, bearing in mind the technological advances in remote sensing technology, data regarding the vegetation cover, as well as its absence, becomes more and more available to regular modelers. Programs such as GLC (Global Land Cover) project or CLC (Corine Land Cover) are becoming available in many parts of the world cost-free for public use, providing cartographic products regarding vegetation assessment, desertification, farmland assessment, urban developing or water resource monitoring (University of Maryland 2014) (EEA 1995).

2.3.4 Evapotranspiration and Interception by canopy

Estimating evapotranspiration is a matter of data availability, where in practice, methods have the tendency to predict potential or actual values for evapotranspiration. On one hand, potential evapotranspiration methods do not take into account the water availability and they calculate values for evapotranspiration without any limitations regarding the water that can be evaporated from the surface or transpiration by the vegetation. On the other hand, actual evapotranspiration methods are based on actual measurements, where over the years, methods for predicting both types of evapotranspiration were created and the most used ones are mentioned below (Beven 2014).

The best method for predicting potential evapotranspiration is the Penman-Monteith equation. Through this equation, the vapor taken away from the earth surface is calculated, taking into account the energy balance of the surface and the turbulence. Even though this method can deliver an accurate prediction, the amount of data required varies from net radiation, dry and wet bulb temperature, wind speed and coefficients regarding the canopy interception and aerodynamic resistance that, in most of the cases, are not facile to obtain (Zotarelli 2010). However, the simplest method is the Hamon's equation, but due to the fact that it does not take into account any meteorological factors, this method is considered to be less accurate.

The present project does not take into consideration evapotranspiration and interception by canopy due to lack of data and the purpose it serves: a broad analysis of the areas that is to be flooded. This is the motive why the actual equations are not presented here. This small subchapter could be approached for future improvements where more inputs could be added in case of further development.

2.3.5 Remote sensing datasets

Remote sensing is defined as an operation of collecting and interpreting data about an object or area without being in physical contact with it. The early attempts in remote sensing were represented by flights performing photography of the earth surface, but developments in technology brought new techniques such as satellite imagery that use more than one wavelength, for instance infrared, thermal infrared or microwave. Each of these types of different wavelengths have a different purpose, from observing vegetation, to wetlands research, farm productivity or snow distribution. (Sanderson 2010)

In hydrology, when working with extensive areas, remote sensing techniques can be used for determining the snow distribution, observing the stream level or vegetation density for estimating interception and evapotranspiration. Lately, there are being created models that are based on satellite data regarding snow coverage, and alongside with the traditional hydrological data give promising results for the future of hydrological modeling (Berezowsky, et al.2015).

2.4 Methods used

This subchapter presents different methods used in the model for determining the quantity of runoff from each cell. The first three methods, CN Method, Day-Degree Method and Snow-Melt caused by rainfall, are used for establishing how much of the precipitation, snowmelt and melted water from the rain on snow events will become runoff. The next two methods, Flow Accumulation and Cellular Automata, count for the second part of the analysis that refers to how and where the water from runoff gathers and which areas are affected by the accumulation, this being the actual area that is considered to be affected by flooding events.

2.4.1 CN Method

Developed by the US Soil Conservation service in 1965 to be applied for small watersheds, the CN model makes use of a fairly simple and limited amount of data for determining the runoff that results after a rainfall event. The runoff calculation is based on data such as precipitation, land use, soil type and soil moisture conditions. The model was successfully applied in hydrology during the years, being referred to as a “blue collar” due to its longevity and applicability (Tae-Woong, et al. 2014).

The purpose of the model is to calculate the quantity of runoff that results after a rainfall. In order for this to happen, the soils from the study area are divided into four hydrological groups, depending on their composition and water permeability. The soil classification is presented by the National Engineering Handbook (NRCS 1997) as follows:

- Group A is characterized by soils with less than 10% clay and more than 90% gravel or sand, which translate into a very low runoff potential due to the possibility of the water to travel freely through the soil.
- Group B is characterized by soils with clay between 10%-20% and 50%-90% sand. Soils from this group are considered to have a moderately low runoff potential.
- Group C is characterized by soils with and 20%-40% clay and less than 50% sand. Soils from this group are considered to have a highly moderated runoff potential, due to the increased percentage of clay that restricts the water infiltration.
- Group D is characterized by soils with more than 40% clay and less than 50% sand and is considered as having the highest runoff potential due to the inability of water to infiltrate.

The CN method calculates the runoff using Equation 1 as presented below (USDA 1986):

$$Q = \frac{(P - I_a)^2}{(P - I_a) + S}$$

Equation 1 - Runoff equation

Where:

Q= daily runoff (mm)

P= daily amount of precipitation (mm)

S= maximum recharge capacity or storage of a watershed after five days of antecedent rainfall

I_a= initial abstraction

I_a is a parameter that includes factors such as evaporation, infiltration, surface depression and interception by canopy. As expected, I_a is not constant, but it is correlated to the type of soil type and to the land cover while a relation, Equation 2 , was developed after numerous studies.

$$I_a = 0.2S$$

Equation 2 - Initial abstraction formula

Using the 0.2S as initial abstraction value, the formula for runoff becomes Equation 3 and the maximum recharge capacity is calculated using Equation 4:

$$Q = \frac{(P-0,2S)^2}{P+0,8S} \qquad P \geq 0.2S$$

$$Q = 0 \qquad P < 0.2S$$

Equation 3- Runoff equation for 0.2S initial abstraction

$$S = \left(\frac{1000}{CN}\right) - 10$$

Equation 4- Maximum recharge capacity of a watershed

Where:

CN= curve number

For calculating the S value, the tables and the diagrams for determining the CN number were created and presented in detail in TR-55 document. The CN value depends on land cover, soil group, soil treatment, hydrologic condition and antecedent runoff condition. The soil group classes are presented above, land cover is represented by vegetation and soil treatment refers to the state and condition of agricultural land, cultivation methods and crop rotation. Hydrologic condition creates a link between the type of land cover, the type of soil and its treatment and quality. Antecedent runoff conditions and accounts for the changes that the soil and land cover can suffer in between two storms. The values for CN presented in TR-55 account for average conditions regarding the antecedent runoff and, in order for these values to change, on site analysis need to be performed.

As mentioned above, TR-55 provides tables and diagrams for evaluating the CN number suitable for the studied area as seen below in Figure 11:

Cover description		Curve numbers for hydrologic soil group			
Cover type and hydrologic condition	Average percent impervious area 2/	A	B	C	D
Fully developed urban areas (vegetation established)					
Open space (lawns, parks, golf courses, cemeteries, etc.) 3/:					
Poor condition (grass cover < 50%)		68	79	86	89
Fair condition (grass cover 50% to 75%)		49	69	79	84
Good condition (grass cover > 75%)		39	61	74	80
Impervious areas:					
Paved parking lots, roofs, driveways, etc. (excluding right-of-way)		98	98	98	98
Streets and roads:					
Paved; curbs and storm sewers (excluding right-of-way)		98	98	98	98
Paved; open ditches (including right-of-way)		83	89	92	93
Gravel (including right-of-way)		76	85	89	91
Dirt (including right-of-way)		72	82	87	89
Western desert urban areas:					
Natural desert landscaping (pervious areas only) 4/		63	77	85	88
Artificial desert landscaping (impervious weed barrier, desert shrub with 1- to 2-inch sand or gravel mulch and basin borders)		96	96	96	96

Figure 11 - Example of tables for evaluation CN from TR-55 document

Based on the type of the surface, the CN value is determined. The same type of tables can be found for all types of land cover such as forests or agricultural land. The tables and the diagrams can be found in Appendixes 9, 10 and 11.

Maximum recharge capacity, namely S, ranges between 0 and infinity, but in order to have accurate calculations, its range was limited to a practical one between 0 and 15. The CN value ranges from 0 to 100, but the practical values are between 40 and 98. While the 40 value would be applicable to soils constituted of sand or gravel, covered by a dense vegetation such as forest with low runoff conditions, a value of 98 is suitable for surfaces that cannot take any more runoff such as artificial surfaces without any drainage or wetlands, rivers or lakes (USDA 1999).

Despite its popularity between hydrologist and engineers, the CN method received numerous criticism over the years, due to its direct fashion of predicting and calculating runoff. Garen and Moore (2005) argued that the method was many times wrongly applied, being used for studies such as determining soil erosion to pesticide carrying into the stream when it should have been only used for predicting the quantity of runoff that is created after an event. Tae-Woong, et al. (2014) argue that the reason why the method is so popular is due to it requiring easy accessible data as an input; therefore, it is being applied in situations where data is not available or scarce. In the original TR-55 technical document the limitations of the method are also widely presented as followed:

- The curve numbers presented in the tables refer to average conditions. An eventual historic storm would decrease the accuracy of the result.
- The equations should be used carefully, the method not taking any temporal factors into consideration such as duration or intensity.
- The initial abstraction factor should be understood and eventually calibrated for the area where the model applies. The value presented in TR-55 is an average value considered for agricultural land. The calibration is especially important when the study area has a large variety of soil groups and surfaces.
- The method cannot be applied for computing runoff as a result of snowmelt.
- The results are more accurate when the runoff is greater than 12.5 mm.
- When the CN value should be considered less than 40, another method should be used for runoff determination.

2.4.2 **Day degree method**

Snowmelt can undoubtedly be considered as the reason why a flooding event takes place, therefore representing the phenomenon in a flood-modeling environment is extremely important. Even though snowmelt may account for a large amount of the water that becomes runoff, there are just a few modeling methods developed over the years (Beven 2014). These can be categorized as Energy balance models or Temperature index models (Hock 2003). The first type of models, Energy balance models, are characterized by complex equations that take into account different factors, such as different types of shortwave radiation, wind speed, temperature, precipitation or vapor pressure. Due to the large number of inputs necessary, using Energy balance models in environments dominated by mountains where data is scarce and its creation is expensive, make Energy balance models not so popular, especially for the modelers that do not have the resources for creating the input data. The second type of models is the Temperature Index models, these being based on a simple relation between the air temperature and the snowmelt. These methods proved themselves over the years, the first model being created in 1887 by Finsterwalder and Schunk and applied on a glacier (Hock 2003). Temperature index models are preferred for the following reasons:

-Temperature data is the easiest data to obtain

- Having access to a large amount of temperature data, most of the interpolation methods give good results and relative small errors

- Their easiness to implement due to their simplicity (Hock 2003)

- Accuracy is comparable to more advanced Energy balance models (Rango and Martinec 1995)

The most used Temperature index model is the Degree-Day Method, which calculates the depth of snowmelt as seen below in Equation 5 :

$$M = C_b(T_a - T_b)$$

Equation 5- Degree day equation

Where:

M= snowmelt (mm)

C_b = degree day factor or ratio (mm C^{0-1} d^{-1})

T_a= temperature registered (C^0)

T_b= reference (base) temperature (C^0)

The formula represents a linear relation between the degree-days registered in a unit of time and a degree day ratio. A degree day is defined as ‘’a departure of one degree per day in the daily mean temperature from an adopted reference temperature” (Linsley 1958, p 340). In other words, the difference between a reference temperature and the medium temperature registered for a unit of time, in this case a day, is called degree-day where the measurement unit is C^0.

The reference temperature is usually 0^0 C and represents the level from which melting starts if the temperature rises or stops if the temperature drops. The registered temperature can be determined as a mean between the minimum and the maximum temperature registered for a day, or unit of time in general. (Hock 2003)

Additionally, different variations in the method of calculating the degree-days exist, Rango and Martinec (1995) suggesting the use of daily temperature as a mean between the maximum positive value registered during a day and the base temperature. In the Engineer Manual from Army Corps of Engineers (1998), it

is suggested that a temperature around 4.44 C^0 can be used as a base temperature, only if maximum daily temperature is used as a mean daily temperature.

The Degree day factor is a factor of proportionality that is not constant and is usually measured in the field using different methods such as snow depth, ablation or shortwave radiation. Linsley (1943) finds it hard to calculate the volumes of snowmelt, and also calculating a snowmelt factor based on the general basin runoff generated by precipitation. He also proved that the factor varies depending on the snowmelt season period. The recommended values for the degree day factor varies from 0.1 cm (1 mm) in March when the melting season starts up to 0.7cm (7mm) in June when the snowmelt season reaches a peak. In another study, Weiss and Wilson (1958) show that the day degree factor is influenced by land cover and that the forest has the greatest effect on it as shown in Table 3 below.

Beginning of April	
Forest	1.85 mm
Open area	3.7 mm
Beginning of June	
Forest	3.7 mm
Open area	7.4 mm

Table 3 - Proposed values for the day degree factor by Wilson(1958)

However, Rango and Martinec (1995) are more reserved while using the Day Degree method on short intervals of time, stating that the method should only be used for general runoff together with precipitations. Their value for the factor is also between the same approximate values, 3.3 and 6 mm.

Despite being one of the mostly used methods for calculating snowmelt due to the easiness of getting necessary input data, there are also limitations. Rango and Martinec (1995) stated that one of the most common error users do when using the method is assuming that the factor value is constant during the entire melting season. In this case, the result of the method is only accurate for less than two months per year. By using a variable factor, as Weiss and Wilson (1958), Linsley (1943) and others also stated, that the accuracy of the method is significantly improved. Additionally, other limitation for using the method are presented in the same study by Rango and Martinec, who show that for short periods of time, together

with the temperature and melting factor as an input, when calculating the snow depth or the snowline retreat, also solar radiation should be considered for accurate results. If not enough data is available for integrating more factors into the day-degree formula, the temperature and the melting factor can only be used for general runoff, the inaccuracies being reduced by the basin response.

2.4.3 Snow-Melt caused by rainfall

The rain on snow events are considered to occur frequently at low elevation and quite rare at higher elevations. However, when a rain on snow event happens at high altitudes where snowpack is present, especially during the melting season, the effect can be seen in runoff quantity (Judah, et al.2015). Additionally, the rain on snow events are dependent on temperature and they occur when the ground is covered in snow, the precipitation is present as rain and the air temperature has a positive value. Putkonen and Roe (2003) show how rain on top of snow can heat up the soil to 0° which otherwise would be well below 0°. When the soil is warmer than the snow, the melting process is increased and the runoff is greater.

Currently, there are not any numerous models available for rain on snow events and there does not exist a high amount of research regarding these processes, despite the big significance it can have in flood forecasting (Linsley 1943) (McCabe, et al. 2006). However, two energy balance models were developed by National Weather Service and Washington University in USA, both accounting for factors such as humidity, temperature, precipitations, solar radiation, albedo and other factors that have challenging availability.

Based on the Day-Degree method presented in the subchapter above, Linsley (1943) presents an equation for calculating the depth of water that results after a rain on snow event using only temperature and precipitation and input as seen in Equation 6:

$$D = \left(\frac{specific\ heat\ of\ water}{heat\ of\ fusion\ of\ ice}\right) P(T - temperature\ of\ snow)$$

Equation 6 - Rain on snow equation

Where:

- Specific heat of water is one British thermal unit per pound
- Heat of fusion of ice is 144 British thermal units per pound
- P is the rainfall depth in mm
- T is the temperature of the rain. A good estimation of the temperature is the wet bulb temperature during the rain.
- Temperature of snow is considered to be 0 C^0
- D is the depth of water melted by the rain measured in mm

A British thermal unit is defined as the amount of work needed to rise the temperature of one pound of water by one degree Fahrenheit. Dividing the specific heat of water by the heat of fusion of ice is a constant value (1/144= 0,007), therefore there is no need to convert imperial into metric units in this case.

Wet bulb temperature is the air temperature measured at 100% relative humidity and exposed to solar radiation, while the dry bulb temperature is the temperature measured if a thermometer is shielded from humidity and moisture. The formula above takes the wet bulb temperature as input data, but due to the fact that during rain humidity rises and the difference between wet and dry bulb temperature become negligible, the more easily to obtain dry bulb temperature can be used instead (Linsley 1958).

Equation 6 can be rewritten as Equation 7 :

$$D = \left(\frac{1}{144}\right) P\ T = 0{,}007\ P\ T$$

Equation 7- Rain on snow simplified equation

The rain on snow equation was developed by Linsley (1958) to work with available data and predict the depth of water melted by the rain. Even after 70 years since this equation was developed, the information for creating an energy balance model is still difficult to obtain, making Linsley's (1958) equation still usable in current times.

2.4.4 **Flow accumulation**

The flow accumulation method is a distinguished method used to derive information regarding the stream network, catchments and watersheds. Developed around 40 years ago, the method is based on algorithms

that are constantly improving in the present, being used by the most known software for hydrological analysis. The method works with a raster datasets and consists of counting the number of cells that flow into each other. The analysis is based on the altitude difference between the cells, the lowest neighbor of a cell is being considered as the cell that would receive the quantity of water. At the end of the analysis the cells that receive the highest amount of water are highlighted, representing also the natural flow of streams (Jenson and Domingue 1988). As presented by Jenson and Domingue (1988) in their study, in order to perform a Flow Accumulation, three steps are necessary to be followed, which are presented below:

Filling the depression or the sinks in a Digital Elevation Model

The first step consists of identifying the areas that have a lower level than the surrounding areas and modify its level so it reaches the lowest value of its neighbor. When a Digital Elevation Data is created, commonly all the time, the data contains errors in the form of sinks. Consequently, if these depressions are not filled, areas of internal drainage are created, which represent the areas where the water accumulates and the analysis starts an infinite loop.

On a raster, an area is considered to have 8 neighbors, unless the area is at the edge of the DEM, which has only 3 or 5 neighbors. If an area, which is represented as a pixel, has a lower value than all the 8 neighbors, its level is brought to the lowest level of the surrounding neighbor.

In a cross section, the filing process would look as seen in Figure 12.

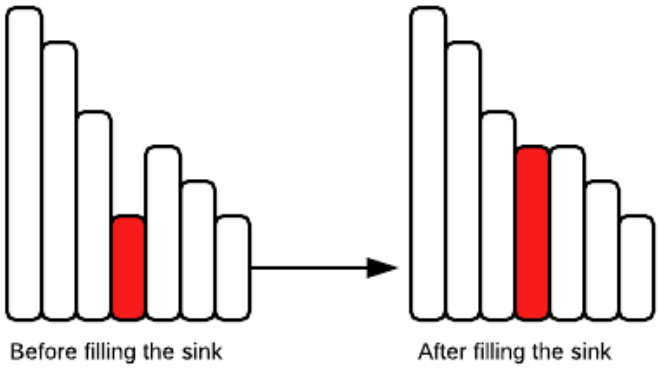


Figure 12- Sink filling- section view

Filing the sinks assure as well that each area has a drainage neighbor or that the water from one cell has where to flow. In reality, there might be a natural sink in the area, but their level usually is not much lower than the surroundings, therefore for this operation the end results is not influenced in a highly manner.

Calculating Flow Direction

The second step in determining the accumulation of water is to map where each cell drains in relation to each other, for each of the eight neighbors of a cell, a value being assigned. The traditional values are powers of two and the counting starts from the east part of a cell as in Figure 13.

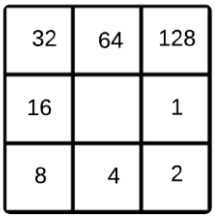


Figure 13- The eight neighbors of a cell and the values assigned

For each 3x3 cell neighborhood the cell with the lowest elevation in relation with the middle cell is selected and the right value is assigned, as seen in Figure 14. In this example, the neighbor cell with the lowest elevation is the west cell, thus, in order to symbolize that the middle cell will drain in the west side, value 16 is assign according with the directions mapped in Figure 13.

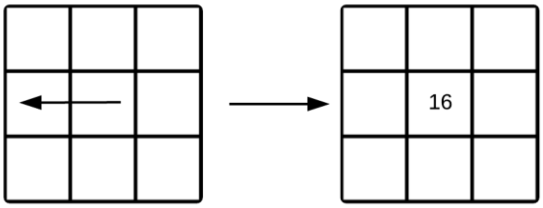


Figure 14- Assigning the flowing direction to the center cell

The result of the Flow Direction operation is a raster with the same structure as the original DEM, where each cell has one of the 8 values representing directions, being possible to calculate how the water is draining over the DEM and which is the order it happens.

Calculating Flow Accumulation

The third and last step consists in assigning a value for each cell equal to the number of the cells that drains into the respective cell, the result being a raster representing areas where the flow is concentrated. Furthermore, this results can be used to determine streams, pour points or basins which are not presented, being beyond the purpose of this model. An example of Flow Accumulation raster output is presented below in Figure 15, the white lines representing the areas where the water accumulates.

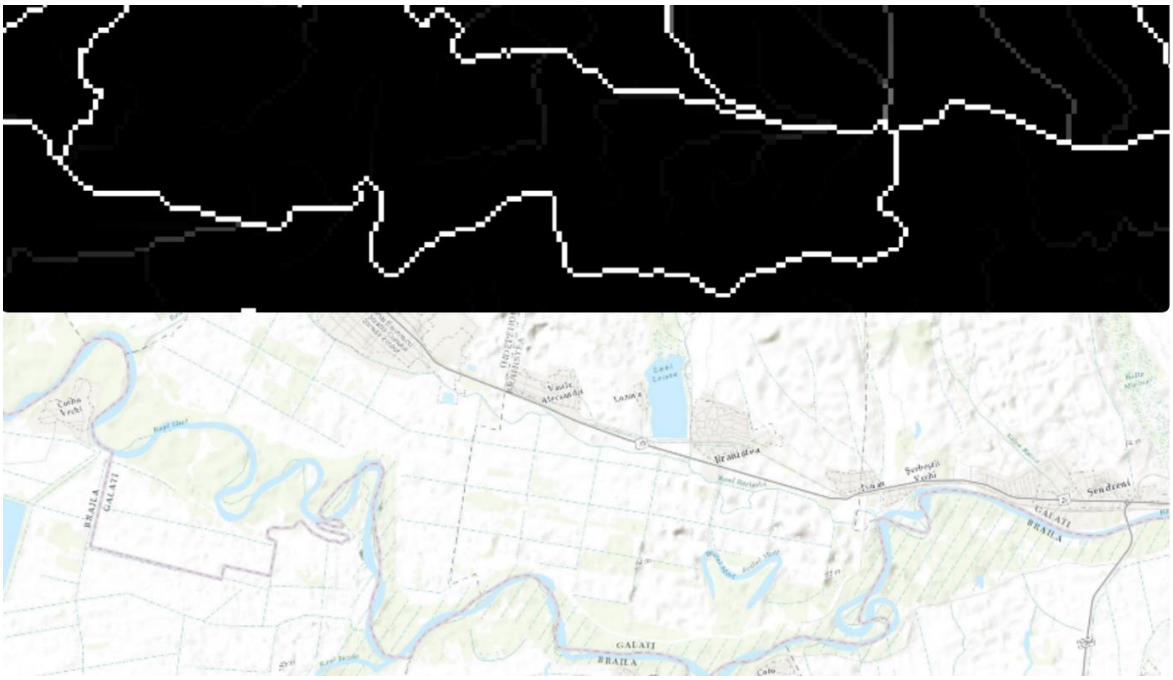


Figure 15 - Flow accumulation output from ArcGIS and a map for verifying the results

In order to be able to observe the quantity of water that accumulates at the end of Flow Accumulation procedure, another raster for holding the quantity of precipitation or runoff for each cell needs to be available. Consequently, besides counting the number of cells that accumulates and forms the streams, the quantity of water that accumulates can be calculated as well. This version of Flow Accumulation is named weighted flow accumulation, where the weight is given by the amount of precipitation or runoff for each cell.

In this phase, the output of flow accumulation provides the place where the water accumulates, which is represented by the raster cells, and the amount of water accumulated in those cells. If the weighted raster previously mentioned represents runoff values, the output raster represents exactly the amount of water that drains into the respective cells from all the basin. However, this situation becomes only partially

useful in a flooding modeling, due to the quantity of water being only represented as a “column of water”. In reality, water would spread over the DEM and a floodplain would be represented. Figure 16 below shows a conceptual output of Flow Accumulation as a section. A column of water (blue bar) is represented on top of the cell where the water accumulates (red cell). The right side of the figure shows how the water would normally spread over the DEM, process which proved to be hard to derive from the output of flow accumulation, extensive programing skills being required.

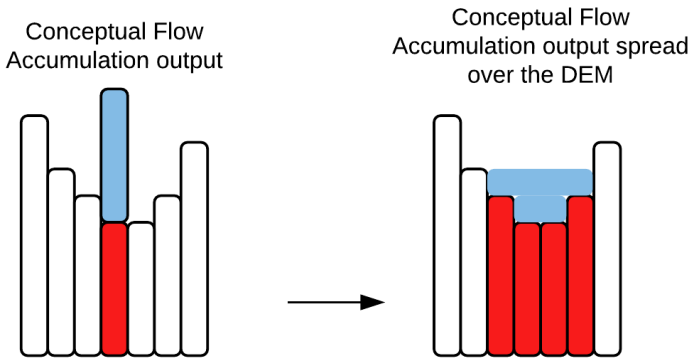


Figure 16 - Flow accumulation conceptual output (left) and a conceptual spread of water of the flow accumulation (right)

This limitation of spreading the volume of water over the DEM determined us to encounter another method that will provide the expected result, while used in conjunction with flow accumulation.

2.4.5 **RACA – Rainfall with Cellular Automata method**

In a Cellular Automata system, space is seen as cells. Each cell has the same size and is characterized by its state, which in turn, is determined by the state of the surrounding cells alongside the state itself. The cell itself and the cells around are called a neighborhood, where the state of all cells is changed once in one iteration or time step (Liu 2009) . Moreover, depending on the size and shape, there are a few types of neighborhoods, but in the current model, the classical Moore Neighborhood (Figure 17) with 8 neighbors is used (Torrens 2000).

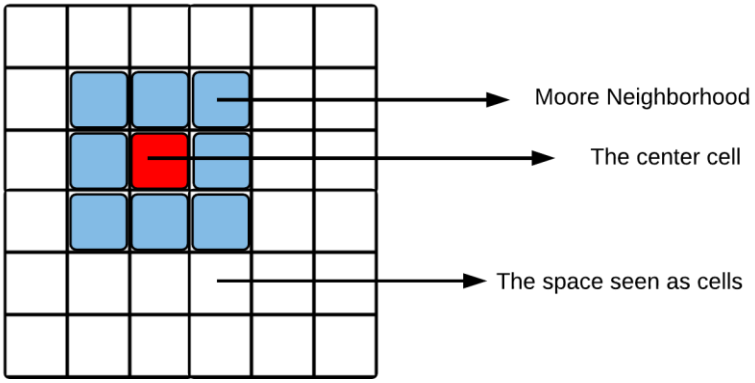


Figure 17 - Moore Neighborhood

Liu (2009) describes a Cellular Automata process as consisting of 5 elements:

- *The cell* represents the basic space unit a Cellular Automata works with. The most common space form is two dimensions, being used for the most geographical related simulation.
- *The state* of a cell is defined by a set of possible attributes, or states, that the respective cell takes at one time.
- *The neighborhood,* as presented above, is the space around one cell.
- *The transition rule* is the most important component of a Cellular Automata system, dictating how the reaction between the neighborhood cells and itself is changing the state of a cell.
- *The time* represents how many times the states of the cells will be changed, also referred to as number of iterations

RACA (Rainfall with Cellular Automata) is a Cellular Automata method whose rules were created by Espinola,et al. (2014) to simulate the precipitation accumulation over a DEM. The method is designed to take a raster representing the quantity of precipitation for each cell and transfer the water from the cell with high altitude to the cell with a lower altitude. This operation is done multiple times, and based on the rules presented below, it stops when the water stagnates (there is no lower altitude cell in the neighborhood) or after a predefined number of iterations.

Based on the five elements presented above, RACA can be described as:

- The cell has a 1x1m dimension
- Two states of the cells are used, where state 1 is represented by the pixel value from the DEM representing the altitude and state 2 is the level reached by each cell when the water volume is added or subtracted. The used neighborhood is Moore Neighborhood, as presented above.

- Regarding time, the method runs until there is no more movement of the water or until the preferred number of iterations is reached
- The method uses two transition rules as presented below:
    - Rule 1. *If* $water\_flow < q_{(i)water\ level}$ *:*

      $q_{(i)water\_level} = q_{(i)water\_level} - water\_flow$

      $q_{(j)water\_level} = q_{(j)waterl\_evel} + water\_flow$

    - Rule 2 *If* $water\_flow \geq q_{(i)water\ level}$ *:*

      $q_{(i)water\_level} = 0$

      $q_{(j)water\_level} = q_{(j)waterlevel} + q_{(i)waterlevel}$

    Where $q_{(i)}$ = current cell

    $q_{(j)}$ = neighboring cell with lower height

    and:

$$water\_flow = \frac{(\text{q(i)terrain\_altitude} + \text{q(i)water\_level}) - (\text{q(j)terrain\_altitude} + \text{q(j)water\_level}}{2}$$

The water flow is the most important part of the transition rule, because the comparison between it and the current level determines which rule to be followed. Espinola,et. Al (2014) used a simple equation to calculate and define water flow in their research, where water flow is the difference in total altitude (terrain altitude and water level) divided by 2, between the center cell and its lowest altitude neighbor.

After determining the water flow between the two cells, the newly calculated value is compared to the water value of the central cell, and based on the result of the comparison, there are two possible transition rules. In the first situation, if the water level is higher than the calculated water flow, then the water flow value is deducted from the center cell and added to the lowest neighboring cell. In the second situation, if the water level is lower that the calculated water flow, the entire amount of water in the center cell gets transmitted to the neighboring cell. This process is iterated N times, until the water level stabilizes and further iteration would not change the states of the cells.

2.5 GIS features for hydrologic modeling

Before reaching this subchapter, all the methods presented above were described in a theoretical fashion. Furthermore, in order to make a functional system that can incorporate these methods and produce some results, a binding agent that can connect everything is required, which is presented in the following subchapter as GIS (Geographical Information System). The subchapter has the aim of explaining how the GIS environment can provide the necessary tools for managing and working with data, while serving as a brief explanation of some of the terms used in Implementation chapter, where the workflow is presented in detail.

2.5.1 GIS and Hydrology tools

The original and the most thorough definition of GIS is considered to be the one given by Burrough (1986), who defines a GIS system as being "… a powerful set of tools for collecting, storing, retrieving at will, transforming, and displaying data from the real world for a particular set of purposes ..." (p. 6). Additionally, other satisfactory definition is given by Dueker and Kjerne (1989 p. 7) and later modified by Brimicombe (2010) defines a GIS system as "a system of hardware, software, data, people, organizations and institutional arrangements for collecting, storing, analyzing, visualizing and disseminating spatial information about areas of the Earth" (p. 46). As it can be observed from the definitions presented above, the term GIS is quite broad and does not refer to a singular process, but describes a collection of entities and processes. Due to its wide definition, GIS can be successfully implemented into a wide range of domains such as geography, engineering or hydrology, the last being the domain research of this project. The present project is an example on how hydrology can make excellent use of GIS. It can be seen that the GIS work starts with the idea for the project, in this case the problem statement and the research questions and continues with the data gathering which involves communication with people or organizations for obtaining the necessary data. Additionally, the GIS environment in which the project or the model in this case will be created is chosen. There are numerous environments that allow GIS work, the present project making use of ESRI's ArcGIS platform. Beside this, other environments to work in conjunction with ArcGIS were chosen, these being Python as programing language and XAMPP to serve as a server for displaying the results.

Moreover, ArcGIS environment offers different tools that work with different types of data, as well as a range of particularized tool sets suitable for a more specialized field, such as hydrology. The tools used for preparing and manipulating the data are presented below for a better understanding.

Flow accumulation can be computed using Python by following the steps presented above in Chapter 2.4. In the same time, ArcGIS environment offers a set of tools that can be used to achieve the same results:

- Fill: This tool uses the DEM as an input and outputs a so called “hydrological conditioned DEM” by filling in the sinks. Sinks are cells that have an undefined drainage direction, meaning that no neighboring cells are lower. This step is extremely important because all sinks need to be removed in order to use the next tool that will calculate a flow direction for each cell.

-Flow Direction: This tool creates a new raster using the hydro DEM created with the Fill tool. The new raster uses direction coding (Figure 13) and assigns the value of the steepest downslope neighbor. If all sinks have been removed properly by using the Fill tool, then each cell will have a direction coding that can be used as input for the Flow Accumulation tool. This tool is based on the D8 single-flow method developed by O'Callaghan and Mark (1984) which that has the disadvantage of only using one direction, the steepest one (Tarboron 1997).

-Flow accumulation tool creates a raster of accumulated flow to each cell using the flow direction raster created before. This tool is based on the algorithm developed by (Jenson and Domingue 1988). An optional input raster that represents a weight applied to each cell can be used for calculating flow accumulation. The flow accumulation ran with this optional input would have as result a raster that for each cell, instead of number of cells that flow into each other will have the cumulated value of the weight raster. In hydrological analysis the weight raster can represent the quantity of precipitation or runoff, the result of flow accumulation being the areas where the highest quantity of precipitation or runoff gathers.

Except for the data preparation phase, ArcGIS is not used for running the model, the tools being accessed through a Python command line, ArcGIS environment having Python as a programing language for the tools.

2.5.2 Interpolation methods in GIS

The international supplier of Geographic Information System software, web GIS and geodatabase management applications, ESRI published online the GIS dictionary, which defines spatial interpolation as "the estimation of surface values at unsampled points based on known surface values of surrounding points" (ESRI, GIS DIctionary 2016). Moreover, the dictionary explains that interpolation can be utilized in order to "estimate elevation, rainfall, temperature, chemical dispersion, or other spatially-based phenomena", but while commonly being "a raster operation, (…) it can also be done in a vector environment using a TIN surface model" (ESRI, GIS DIctionary 2016).

Spatial interpolation is a widely used feature in many GIS projects, both explicitly in analysis, and in various data preparation procedures as well (Longley,et al. 2010). Interpolation is necessary in GIS because of the inability to sample every point in space and time, this resulting in a GIS analyst usually working with irregular sampled data, as well as needing to interpolate various data in points where that data is not available. This is done using values from known points. By using values from known points and due to the fact that there is no modality of knowing the actual values of the interpolated points, studies claim that "spatial interpolation is a process of intelligent guesswork in which the investigator (and the GIS) attempt to make a reasonable estimate of the value of a continuous field at places where the field has not been measured" (Longley,et al. 2010, p. 373). Interpolation is only useful with certain types of data, namely of phenomena that are continuously distributed in space.

As mentioned above, spatial interpolation calls for "intelligent guesswork" (Longley,et al. 2010, p. 373), which can be accomplished by using various methods, all having their own strengths and weaknesses. Despite their differences, all spatial interpolation methods are based on one underlying principle, namely Tobler's Law (Longley,et al. 2010). Tobler's First Law of Geography (TFL) states that "everything is related to everything else, but nearby things are more related than distant things" (W. R. Tobler 1970, p. 236). Over time, TFL has become synonymous with the concept of spatial dependence, which is the foundation of spatial analysis, and its most basic application in interpolation is the Inverse Distance Weighting method. Subsequent research has proven that while Tobler's first Law holds true, it only applies up to certain distances, and geographical phenomena are complex and influenced by other factors (Miller 2004). This has led to the development of other more complex interpolation method, known collectively as geo-statistical interpolation methods, the most widely used being kriging.

Finding the right interpolation method can be a challenging endeavor. The modelled fields can be very complex, with data being far from optimally sampled and discontinuities or noise can be present.

Additionally, the datasets can have a very large extent and originate from different sources. Consequently, interpolation methods need to satisfy several conditions: accuracy and predictive power, robustness and flexibility in describing various types of phenomena, smoothing for noisy data, d-dimensional formulation, direct estimation of derivatives (gradients, curvatures), applicability to large datasets, computational efficiency, and ease of use (Longley, et al. 2005).

Considering the current development of spatial interpolation methods, it is difficult to find a method that fulfills all previously stated conditions. In this context, the choice of an adequate interpolation method with the appropriate parameters for the particular application and dataset becomes crucial (Longley,et al. 2010) (Caruso and Quarta 1998). Different methods can lead to significantly different results which can further affect any simulation based on the interpolated data (Mitas and Mitasova 1999).

There are many different interpolation methods available for a variety of purposes, but this chapter will treat only the ones available within the ArcGIS software suite. These methods are the following:

- Inverse Distance Weighting(IDW)
- Kriging
- Natural Neighbor method
- Spline

Inverse Distance Weighting (IDW)

The Inverse Distance Weighting (IDW) represents the most accessible from the interpolation method available within GIS systems and it is the most widely used method by GIS analysts (Longley,et al. 2010). It is directly based on Tobler's Law and it determines the values of unknown points as averages of the sample points around it, with closer points having a higher weight (Figure 18). This method relies heavily on the concept of weighting the distance to the unknown point. There are various methods of defining the weights with the most common being the inverse square of distances (ESRI 2016). The simplicity of this method makes it accessible to implement and, as a result, it is available in most GIS software, but not without being affected by several drawbacks (Longley, et al. 2005).

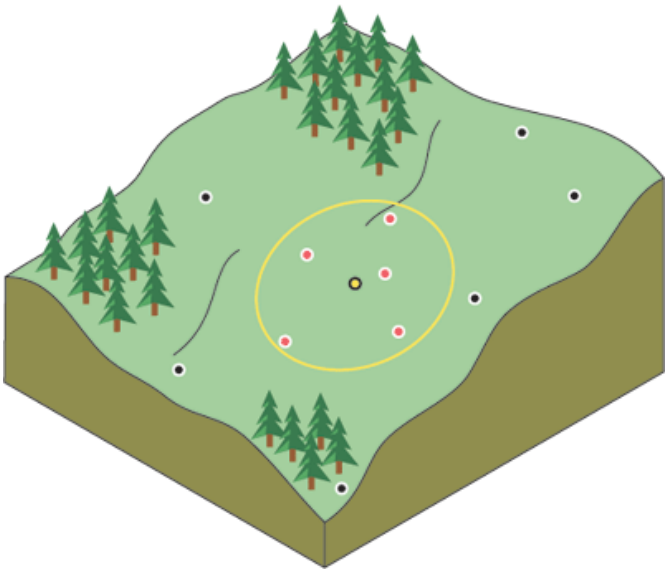


Figure 18 - IDW Interpolation((ESRI, How IDW Works 2016)

Firstly, one drawback is represented by the values of the weight factor that are extremely difficult to determine, since the method is not linked to any real process. The more the value increases, the more emphasis can be put on closer points, but in itself it is an arbitrary value and the user has no possibility of making an informed selection of the weight factor (ESRI 2016).

Secondly, another drawback of the method is in the way it handles peaks and pits in the data. When dealing with a peak or a pit that has not been measured, the method will yield counterintuitive results. In the case of peaks, the interpolated values would be lower, on the grounds that it is influenced by other lower points, while in the case of the pits the same process is applied, only in reverse, creating small peaks. A similar problem arises when extrapolating results with IDW always indicating a regression to the mean. Being a method that calculates averages based on nearby points, no point in the interpolated surface can have a higher value than the highest measured value or a lower value than the lowest measured value (Longley,et al. 2010).

While a number of improvements have been suggested to this method (Tobler and Kennedy 1985), none of them have made their way in GIS software. Despite the drawbacks presented, this method's simplicity and ease of implementation makes it the most popular interpolation method used by GIS analysts (Longley, et al. 2010).

Kriging

Kriging represents a geo-statistical method that generates an estimated surface using a random function, assuming that the surface is the result of a function with a certain spatial covariance. Thus, Kriging uses the dataset twice, the first time for generating a function that fits the data and second time to interpolate unknown points (Longley, et al. 2005). As a consequence, Kriging is considered as making the most convincing claim of being grounded in good theoretical principles (Longley,et al. 2010).

When talking about the assigned weight to measured values, Kriging operates in a similar modality as IDW. However, one difference between them is the fact that the weight is not influenced solely by distance, and by fitting a statistical function to the data, Kriging can respond to both distance and direction of data. (Naoum and Tsanis 2004). Moreover, by including autocorrelation (statistical relationship between measured points) in the process, it can provide not only a predictive surface, but a measure of certainty to the predictions as well (ESRI 2016). Further on, another difference between them represents the fact that kriging requires attention from the user, which needs to be familiar with the area of interest and with the phenomena analyzed. By having the possibility to choose the function that best fits the data, the user needs to be better informed about the process and is required to spend more time on the task. Kriging is also slightly more computational demanding, as a result of the increased complexity of the equation (ESRI 2016).

Natural Neighbor

Natural Neighbor is seen as another method of interpolation used within the GIS software, also known as Sibson or “area-stealing” interpolation (Longley,et al. 2010). This method functions by creating a Voronoi diagram of all points and then overlapping a Voronoi (Thiessen) polygon while assigning weights based on the percentage of overlap. It is a local method, using only points close to the inferred point and it has the same problems as IDW when dealing with peaks and pits, therefore it is unable to produce them unless they are present in the measured data (ESRI 2016) and it adapts to the structure of the input data (Arun 2013). The number of points varies based on the configuration of known points. By using areas or volumes rather than distances, it produces a smooth surface, that passes through data points and works equally well for both regular and irregular data (Arun 2013).

Spline

Spline is an interpolation tool that applies a mathematical function in order to minimize surface curvature and result in a smooth surface. It derives its name from the special type of polynomial called spline and its best usage is for generating gently varying surfaces (ESRI Help) (Arun 2013). Because it creates a smooth surface, it fails to account for rapid changes that can occur in the vicinity of the data points.

2.5.3 Interpolations comparison

As mentioned in the previous subchapter, interpolation is an "intelligent guesswork" (Longley,et al. 2010, p. 373). As a result, it is important to ensure that the method chosen fits the data and gives the best possible results. The best way to compare the results of any interpolation is by comparing it with the actual values from the field. The precipitation data acquired for this project is already interpolated (Jay LaPorte personal communication March 2016) using values from measuring stations, but we assume that the interpolation done by the data provider is done using extensive research beforehand. With this in mind, we can run all forms of interpolations previously presented and compare the results with points retrieved from the provider but not used in the interpolation.

There are several studies testing the accuracy of interpolation methods, but none of them offer any significant conclusion (Harkamp, et al. 1999), this resulting in our decision to create our own comparison, in order to select which interpolation method is most appropriate to implement.

In order to achieve the desired outcome, we selected a small area in the middle of our Basin and created a fishnet of points for which we retrieved additional precipitation information (Figure 19), emphasizing on the fact that the data points do not overlap. After retrieving the new data, we create several raster files from our daily precipitation data corresponding to each interpolation method we want to test, in our case IDW and Nearest Neighbor (Figure 20), Kriging and Spline (Figure 21).

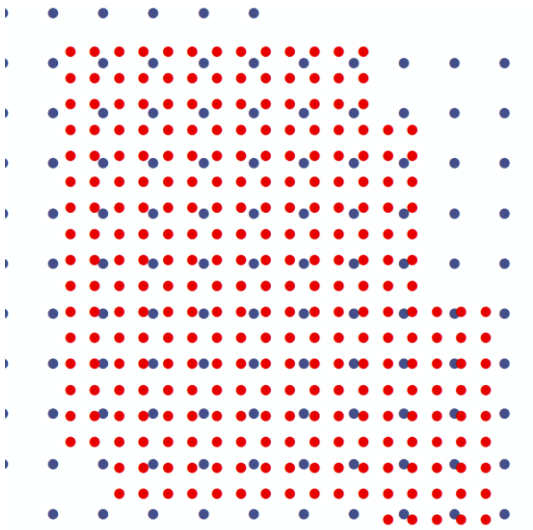


Figure 19 - Precipitation data points (red=control, blue=values for interpolation)

Figure 20- IDW (left) and Natural neighbor (right)

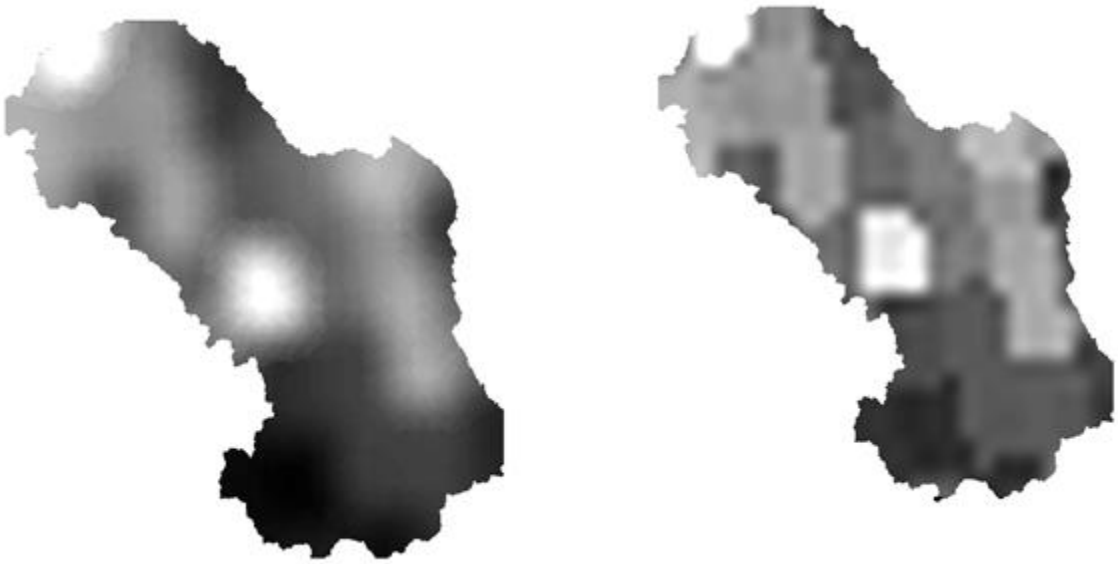


Figure 21- Kriging (left) and Spline (right)

The first step represented the visual inspection and it reveals some expected faults in the results, such as the strong bullseye aspect in the IDW. As suggested by the literature, this situation occurred due to the method applied, especially for the mountain areas (Luo, Taylor and Parker 2007) .The Natural Neighbor method also has a similar issue as IDW. By tacking the structure of our initial grid points, it has a strong gridded aspect. Additionally, whereas the Kriging result has a smooth aspect, the Spline result presents a jagged aspect, this happening despite the claims in the literature suggesting that Spline delivers a smooth aspect (Arun 2013). All methods present a gridded aspect, due to the nature and resolution of data.

Prec_2016_05_12	Kriging	Spline	IDW	NN	Diff_Kriging	Diff_spline	Diff_IDW	Diff_NN
0.203199998	0.088468581	0.085440055	0.084461465	0.085107729	0.114731416	0.11776	0.118739	0.118092
0.203199998	0.089991756	0.070541881	0.079701521	0.076208383	0.113208242	0.132658	0.123498	0.126992
0.203199998	0.09292803	0.066156454	0.084040023	0.076489031	0.110271968	0.137044	0.11916	0.126711
0.203199998	0.099779449	0.081250794	0.08138489	0.084396951	0.103420548	0.121949	0.121815	0.118803
0.203199998	0.107769705	0.116538413	0.117031761	0.113899879	0.095430292	0.086662	0.086168	0.0893
0.114299998	0.168402642	0.225309819	0.182251245	0.203199998	-0.05410264	-0.11101	-0.06795	-0.0889
0.114299998	0.172394723	0.220088959	0.188541234	0.203200221	-0.05809473	-0.10579	-0.07424	-0.0889
0.114299998	0.184505016	0.216870159	0.1908748	0.205438465	-0.07020502	-0.10257	-0.07657	-0.09114
0.114299998	0.179345608	0.222047538	0.194604039	0.209162474	-0.06504561	-0.10775	-0.0803	-0.09486
0.119400002	0.176388919	0.232332617	0.189778745	0.210800007	-0.05698892	-0.11293	-0.07038	-0.0914
0.119400002	0.170352116	0.236476287	0.18489629	0.210800007	-0.05095211	-0.11708	-0.0655	-0.0914
0.119400002	0.152471989	0.201639637	0.16992335	0.185343862	-0.03307199	-0.08224	-0.05052	-0.06594
0.119400002	0.138594732	0.138344437	0.13056168	0.135375857	-0.01919473	-0.01894	-0.01116	-0.01598
0.106700003	0.114331938	0.101858936	0.114981584	0.106700003	-0.00763194	0.004841	-0.00828	0
0.106700003	0.110550962	0.105832092	0.106770918	0.106700003	-0.00385096	0.000868	-7.1E-05	0

Figure 22 - Comparison of retrieved and interpolated precipitation values

Precipitation is a regional phenomenon and the resolution of our meteorological data has a low spatial resolution that does not allow the capture of such sensitive features.

The following step represents the construction of a quantitative analysis of the interpolation results by comparing them with the data retrieved from the provider. While retrieving the values from the interpolated surface, we input them into a shapefile containing the points and the values retrieved (Figure 22).

By comparing the interpolated values with the new values we retrieved, we can observe that the interpolated values are all within similar ranges and some can differ when compared to the retrieved values quite a lot, while other points can be very similar in other cases. Afterwards, we calculate the difference between each interpolation method and the retrieved data, and calculate an average difference for each method (Figure 23).

Average_difference Kriging	-0.020963976
Average_difference IDW	-0.021182364
Average_difference Natural Neighbour	-0.021677057
Average_difference Spline	-0.02182781

Figure 23 Average Difference for each interpolation type

As it can be observed, the overall difference between each method and the retrieved data is fairly similar with an advantage for Kriging, thus suggesting that Kriging is the best method four our situation. This being said, we must also consider other factors such as processing time, and here, due to its simplicity, IDW has an advantage.

3 Data Gathering and Preparation

3.1 Necessary data

Getting free access to geographical data in Europe has proven to be an extremely important initiative, achieved mostly through the INSPIRE (Infrastructure for Spatial Information in the European Community) directive. Proposed for the first time in 2002 and adopted in 2004, the directive was gradually implemented by the West European countries in the beginning and, in the recent years, also countries form the eastern part of Europe have made huge improvements in this direction. The directive's aim is to create a European spatial data infrastructure that will facilitate data sharing between organizations and will allow public access to spatial information in Europe. INSPIRE has been constructed based on few principles: (1)data should be collected only once and be maintained efficiently;(2) users should be able to access data from different sources around Europe and share it; (3) data should be collected at multiple scales, so it can serve diverse purposes (INSPIRE 2015).

Following these principles, each country had the task of gathering environmental data and build a database that will be afterwards shared in Europe. The datasets created varies from DEMs, administrative boundaries, hydrology, up to atmospheric conditions, energy resources or habitats and biotopes.

An important part of data creation in INSPIRE is the creation of metadata. The directive has clear rules about the structure of a dataset and how it should be designed. Having good metadata gives the possibility of better structuring the data and works as a tag system making it easier to locate the dataset while providing the necessary information about its structure and how it can and should be used (ESRI 2002).

As presented in the Chapter 2.3, for hydrological modeling a specific set of data is necessary. Most of the datasets used for this thesis were obtained directly from the INSPIRE geoportal and are presented in the flowing subchapter together with their structure and metadata.

Soil data - ESDB v2 - Raster Library 1kmx1km

Data regarding the soils is created and managed by the ESDAC (European Soil Data Centre) and is freely available on the base of a request. The present data is a raster dataset with a resolution of 1km x 1km and is derived from the European Soil Database (ESDB) v2.0. The dataset contains most attributes (73 in total) from the European Soil Database. The grids are aligned with the reference grid recommended in INSPIRE

and the coordinate system is ETRS Lambert Azimuthal Equal Area (ETRS_LAEA), the grid extent covering 25 European countries. ESDAC created the raster dataset using “feature to raster tool” and ArcGIS software, the feature layer being the shapefile from ESDB (ESDAC 2015).

For our study, only the dominant structural class attributes were necessary. Also ESDAC provides a symbology layer for each dataset for better visualization.

Land cover- CORINE land cover

CORINE (coordination of Information on the environment) is a project initiated by the European Union in 1985 as a prototype, whose aim was to work with multiple environmental problems. One of the goals was to create a cartographic product for presenting the land cover of each country in the European Union, 12 countries at that time. The aim was to provide an overview of the existing information regarding the environment and help the decision-makers in the future. The data is captured using remote-sensing solutions, while the resolution of the photos and the scale of the map depend of the needs and the type of decision that is to be made (EEA 1995).

CORINE land cover nomenclature is organized on three levels: first level contains 4 major classes and the second and third has 15, respectively 44 subdivisions. The four major classes are artificial surfaces, agricultural areas, forest and semi-natural areas and wetland, the next divisions going more in detail for each of these major classes.

Nowadays, the datasets are managed by EEA (European Environment Agency) and are free for download. For the present project, a raster dataset 100m x100m created in 2006 and revised in 2014 was used. The coordinate system is the same as the soil dataset, ETRS Lambert Azimuthal Equal Area (EEA, Corine Land Cover 2006 raster data 2014).

DEM

The Elevation data necessary for the modeling was obtained once more from European Environment Agency. The data was created during the Copernicus program whose aim was to create a cartographic product in the form of a Digital Surface or Elevation Model representing the first surface as illuminated by the sensors. The dataset is available to download for the entire of Europe, at a resolution of 25 meters.

According to the methodology presented on the EEA website, this product is a hybrid of SRTM and ASTER GDEM. This fusion was achieved using a weighted averaging approach and resulted in a contiguous dataset divided into 1 by 1 degree tiles, following the SRTM naming convention (EEA 2015). A disclaimer, present on the website as well, states that the dataset is provided as is, without any form of validation. That same page states that an internal validation was scheduled for 2014, but there is no information regarding the results of said validation. Because of its coverage, the dataset is available in the following projections: ETRS89 geographic (EPSG code 4258) and ETRS89-LAEA (EPSG code 3035).

Administrative boundaries

As presented in Chapter 1.3, Romania has 14 hydrological basins mostly organized after administrative criteria, rather than a hydrological one. Therefore, it was important to have a dataset containing the boundaries of the administrative unit and more importantly, the limits of the Siret Basin. Most of the data presented above comes in different formats, grids that cover the entire country or just a 1000km square. The Basin boundary data was used to create a mask or a limit for extracting or clipping only the necessary data from each dataset, moreover, being also necessary for reducing processing time and having a better view of the work area. The dataset was obtained from Apele Romane, the authority that manages the basins in Romania.

Precipitation data

There are multiple websites or organizations that offer free meteorological data such as Open Weather Map, BloomSky, CliMate and many others, some more reliable than others. Our choice was based on a few criteria that the meteorological source needed to accomplish, where the first one was the data quality and the second one was the way we can integrate the received data into the project. Forecat.IO was the final choice based on a few advantages that it had over its competitors: the data is retrieved from up to 19 sources and meteorological institutes from Europe and around the world, as well as being aggregated together statistically in order to provide a very accurate forecast. Forecast.IO has a very simple and clean API (Application program interface) that can be integrated in most of the programing environments. Additionally, it allows the user to make up to 1000 free requests for point locations per day. The meteorological products for a given point vary from precipitation and its intensity and type, air

temperature up to ozone layer measurement or sun rise and sunset time making it a complete source of meteorological information.

3.2 Data preparation

Multiple sources of data translate to different extents on how the downloaded data is presented. The download containing the soil data is for the entire Europe, the DEM dataset cover the interest area but also a part of Bulgaria and Moldavia. CORINE land cover also represents the entire continent, having more attributes than needed. The following pages will present how these datasets are prepared for the implementation phase presented in Chapter 4. The software chosen for preparing the data was ArcGIS, and some of the tools presented in Chapter 2.5.1 were used. The only dataset that was not altered is the administrative limit for the Siret Basin that will serve as a clipping mask for all of the other datasets.

DEM preparation

As seen in Figure 24, a number of steps needed to be applied in order to get the area of interest ready for the implementation phase. The downloaded DEM has a rectangular shape of 1000x1000 km and covers also parts of Romania's neighbors. First step (step 1 below) was to clip only the surface limited by the Romanian administrative boundaries. This step was realized using "Extract by mask" tool available in ArcGIS, having the Romanian national boundary as input polygon or mask. The second step (step 2 below) was to find all the sinks in the DEM and calculate Flow direction for each cell. This operation was executed using the" Sink" and "Flow Direction" told in ArcGIS. The last step was to clip the result from step 2, the Flow Direction map using the Siret basin administrative boundaries, as seen below. Clipping was done once more using "Extract by mask tool". Having the area ready, resampling was necessary in order to transform the raster resolution from 25x25m into 100x100m, 100 m being the chosen resolution. We chose to calculate flow direction for the entire country and not only for the interest area in order to have the correct information in the cells closest to the edges. Calculating flow direction after the clipping of the

Siret Basin, the closest cells to the edges would have no information because there are no neighbors for calculating which cells contribute to the respective cell.

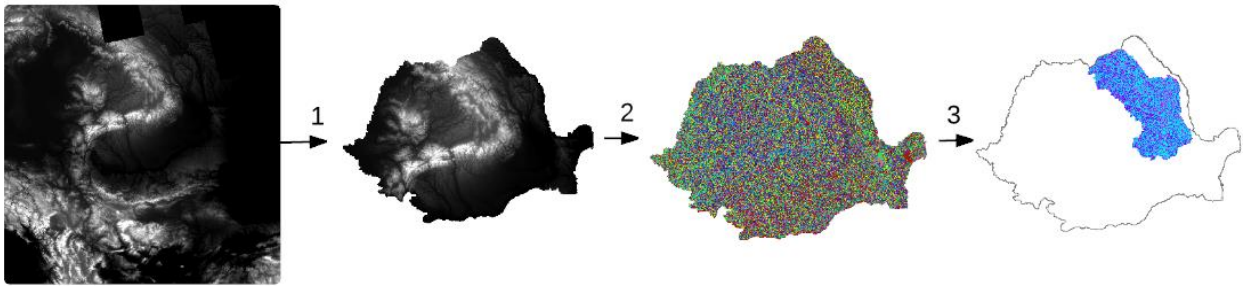


Figure 24 - Steps for preparing the data in the interest area

Soil map preparation- analysis

As shown by our literature, in order to be able to calculate the runoff for each cell, a better understanding of the soils discussed in the project was necessary. The CN method operates with four classes of soil, classified in groups from A to D, the A group representing the soils with the highest percentage of sand and gravel, as well as a low percentage of clay, whereas the D group is being characterized by a large percentage of clay, as well as a low percentage of sand and gravel. For this analysis, the dataset containing the first class of textural soil types was downloaded for the entire Europe and the area of interest, while Siret Basin was isolated for determining the soil type.

The vast majority of the soils can be categorized as having medium type of soil with just a few regions where the level of clay is higher, as suggested by Figure 25. This analysis is extremely important since it gives the first indication on how to use the tables representing the CN value and what values to choose.

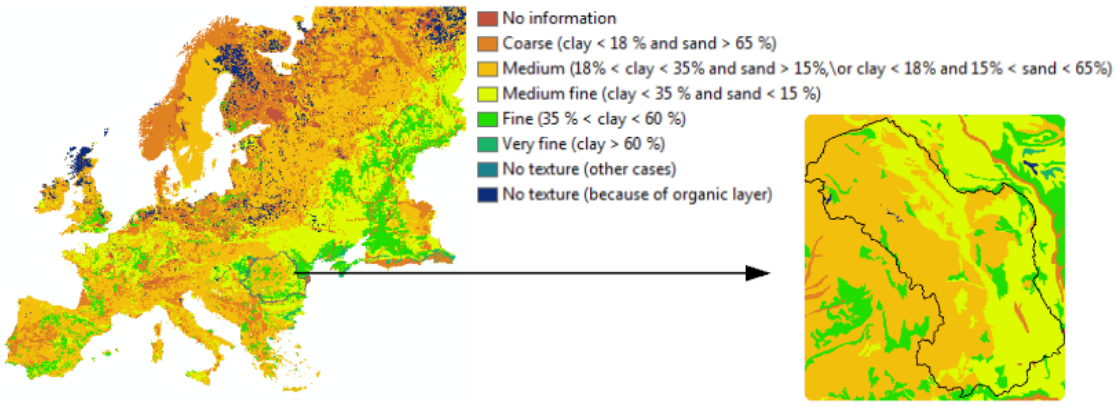


Figure 25 - Soil type study for establishing working values from TR-55 tables

Land cover

Besides the soils, another important component that needs to be considered is the land cover. An A group soil will have a great runoff coefficient, if the land cover is characterized by artificial surfaces such as urban areas or roads. The data is downloaded once more for the entire Europe and the actual dataset comes with a symbology layer and an extra dbf file that contains the CORINE land cover codes for all four levels, as presented above in the Data Gathering.

While regarding the soil dataset where a visualization of the soil types was sufficient for the analysis, when it comes to the land cover dataset an attribute table is required in order to assign the CORINE information provided as dbf file for each type of land cover. A default attribute table for a raster dataset has only three columns created by default by ArcGIS, as seen below in the Figure 26.

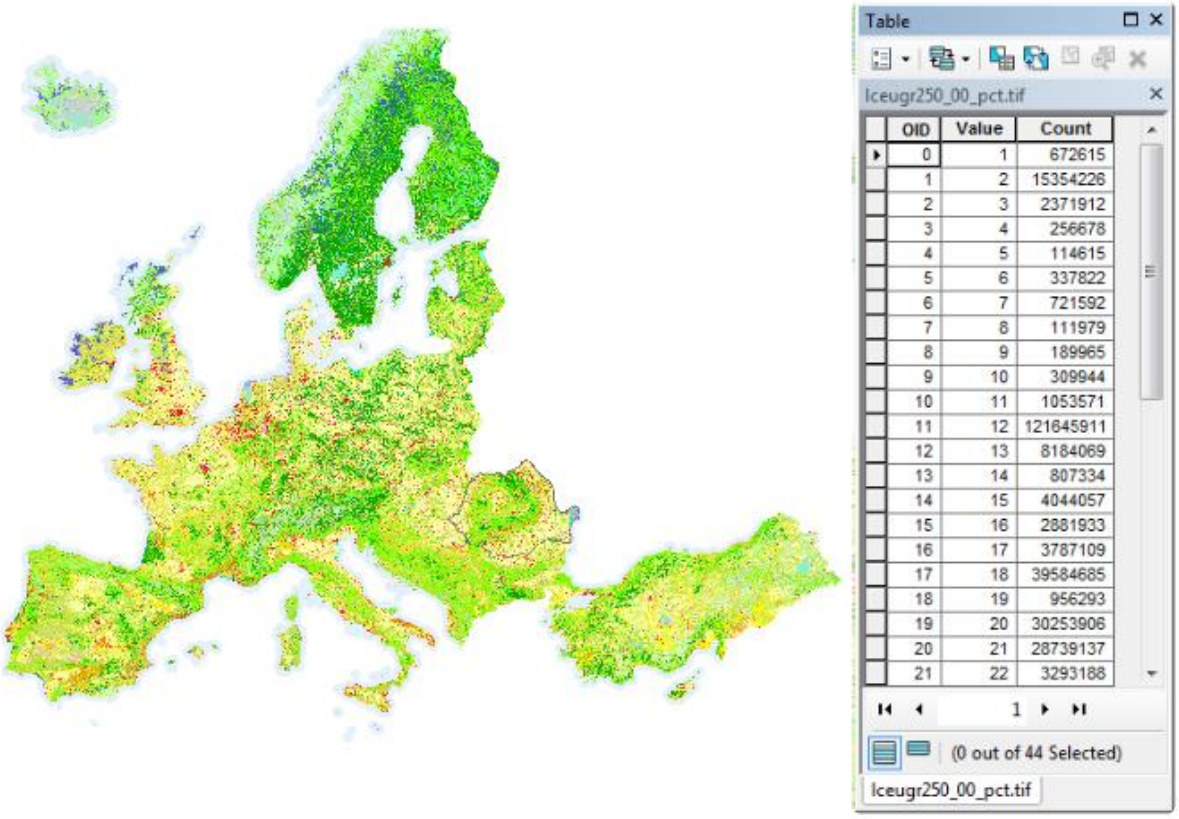


Figure 26 - Land cover data for the whole Europe visualized in ArcGIS

The OID (ObjectID) is a unique system defined object identifier for each row in the table and Value is a list of each unique cell value in the raster datasets. COUNT represents the number of cells in the raster dataset with the cell value in the VALUE column.

The dbf table provided with the data has a similar structure with the attribute table, as seen in Figure 27 below:

clc_legend

OID	GRID_CODE *	CLC_CODE	LABEL1	LABEL2	LABEL3	RGB
0	1	111	Artificial surfaces	Urban fabric	Continuous urban fabric	230-000-077
1	2	112	Artificial surfaces	Urban fabric	Discontinuous urban fabric	255-000-000
2	3	121	Artificial surfaces	Industrial, commercial and transp	Industrial or commercial unit	204-077-242
3	4	122	Artificial surfaces	Industrial, commercial and transp	Road and rail networks and	204-000-000

Figure 27 - Symbology layer table visualized in ArcGIS

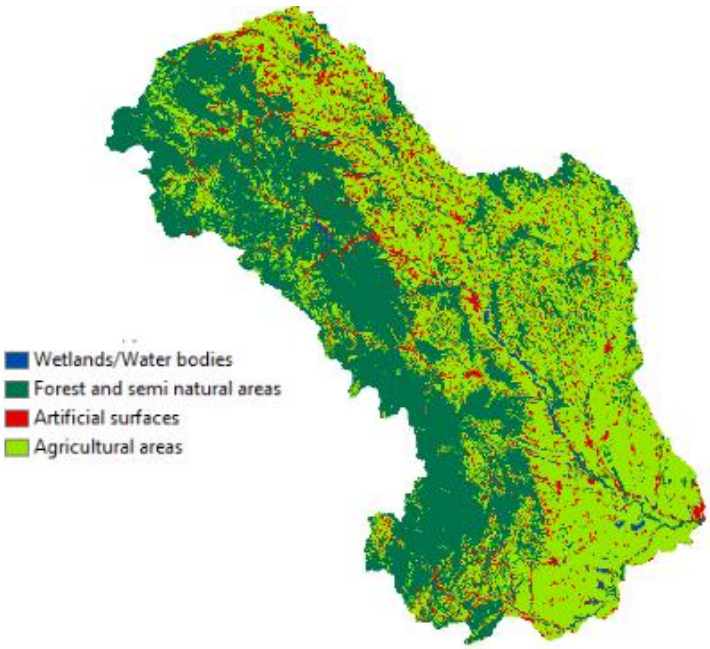
Having a similar structure and the same number of records, the two tables can be combined using the field OID or VALUE-GRID_CODE* as a common identifier between the tables. The result would be a dataset in which for each count of the cells, as seen in the previous figure, the type of land cover is assigned. In ArcGIS the two tables were combined executing a "Union" between the two tables having OID as a common field. The result is shown in the Figure 28 below.

CLC06Siret

OBJECTID *	Value	Count	OID_1	GRID_CODE	CLC_CODE	LABEL1	LABEL2	LABEL3
2	2	307607	1	2	112	Artificial surfaces	Urban fabric	Discontinuous urban fabric
3	3	23997	2	3	121	Artificial surfaces	Industrial, commercial and transport units	Industrial or commercial units
4	4	1167	3	4	122	Artificial surfaces	Industrial, commercial and transport units	Road and rail networks and associate
5	5	84	4	5	123	Artificial surfaces	Industrial, commercial and transport units	Port areas
6	6	630	5	6	124	Artificial surfaces	Industrial, commercial and transport units	Airports

Figure 28 - Union operation between the symbology layer and the land cover dataset

The following step implied clipping the dataset using the Siret Basin boundaries, to delete the unnecessary fields in the attribute table (LABEL2, LABEL3, GRID_CODE, OID_1) and to reclassify the Label 1 values. Because for the analysis only the LABEL1 field is necessary, all the fields with the same value can form a single record having in the end only 4 records representing the 4 big classes presented in the CORINE nomenclature. These operations were executed in ArcGIS using the "Extract by mask "and "Reclassify" tool and the result can be seen in the Figure 29 below.



Count	Label1Copy
178734	Forest and semi natural areas
209580	Agricultural areas
337578	Artificial surfaces
76957	Wetlands/Water bodies

Figure 29 - Output of the union representing the land cover using the four main classes from CLC

The last step represents a CN curve number that should be assigned for each of the land cover classes. The procedure of assigning a CN value based on the type of soil and land cover is explained in detail in the theory chapter, Chapter 2. In this case, the soil type was considered as being in between class B and C, a medium content of clay and sand as seen in Figure 25 and soil condition were considered as being good. The assumption regarding the quality of soil is made due to lack of data, the quality of soil being considered the same as in other regions with same type of climate and relief. In the end, for each type of land cover, knowing the soil type and the soil condition and using the tables, the values were assigned to the dataset. For the artificial surfaces group, satellite photos were studied in order to determine if the surface is generally concrete or other artificial surface with a large coefficient.

As seen in Figure 30, the CN coefficient for Forest and semi natural areas have the lowest value, getting higher for Agricultural areas, Artificial surfaces and finally the Wetlands that cannot receive any more run-off.

	OBJECTID *	Value	Count	Label1Copy	CN
▶	1	40	178734	Forest and semi natural areas	40
	2	65	209580	Agricultural areas	65
	3	80	337578	Artificial surfaces	80
	4	90	76957	Wetlands/Water bodies	90

Figure 30 - Corresponding CN value assigned for each land cover class

Having the CN coefficient assigned for each cell of the raster containing the land use, the S value, or maximum recharge capacity after five days antecedent rainfall can be estimated using the following empirical formula, Equation 4 as presented in the theory chapter and again below:

$$S = \frac{1000}{CN} - 10$$

In essence, for each of the cells of the raster containing the CN values, the formula is applied and a new raster with the same number of cells is created containing the S values corresponding to each CN value. Figure 31 below shows a very basic representation of the process.

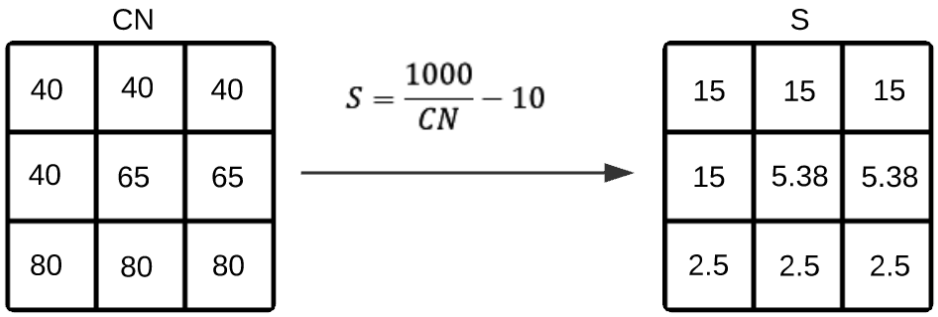


Figure 31 - Example of how Equation 4 is applied to the raster having the CN values and the output

The Raster Calculator tool from ArcGIS uses the same process for executing a mathematical expression using the Map Algebra interface. The equation above in the Raster Calculator tool will have the following expression: ("CN_LandCover" / 1000)-10 where CN_LandCover is the name of the raster containing the CN values.

The result of the raster calculator tool, Figure 32, is the raster holding the value for the S coefficient.

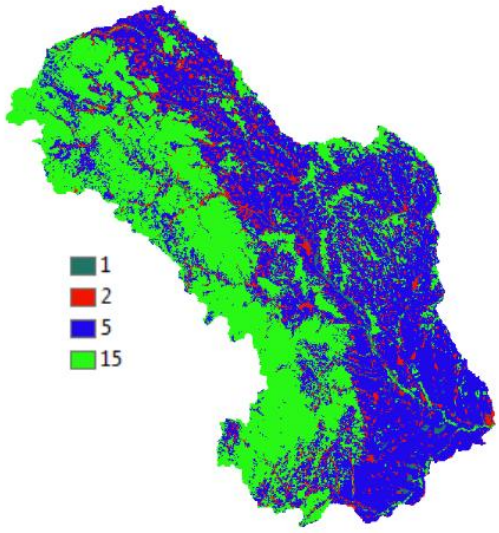
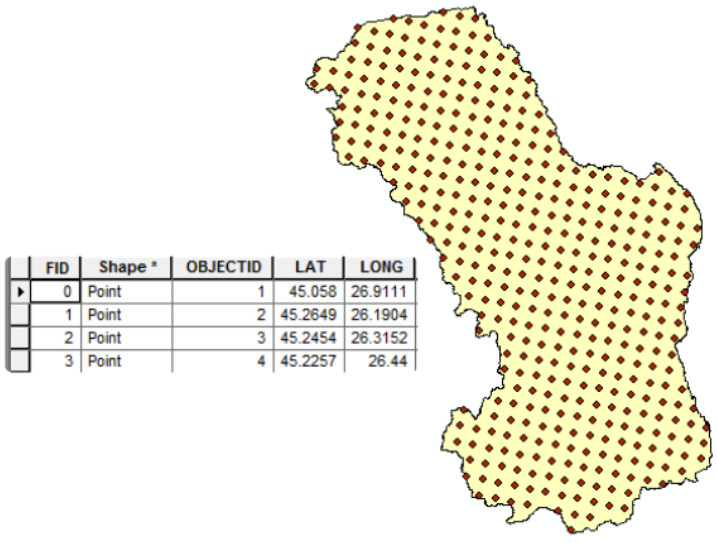


Figure 32 - Output representation of the raster containing the values for S

Precipitation data preparation

Precipitation data is requested using a Python script, which will be presented in detail in the Implementation Chapter. A grid of points was created in ArcGIS at a distance of 10 km between each point, the geographical coordinates for these points were found afterwards and added to the attribute table of the point feature class. The coordinates will later serve as an input for the precipitation value request made through.



FID	Shape *	OBJECTID	LAT	LONG
0	Point	1	45.058	26.9111
1	Point	2	45.2649	26.1904
2	Point	3	45.2454	26.3152
3	Point	4	45.2257	26.44

Figure 33 - Grid of points for which the temperature will be retrieved

The coverage of the area with points and the attribute table can be seen in the Figure 33 above. The number of points is limited by the number of the free request that can be made daily, whereas having a higher number of points would result in better interpolation methods.

Data preparation for Cellular Automata method

Because of the large area of study and the high computational requirements, it was necessary to create a smaller area in which to run the cellular automata analysis, while still receiving input from the rest of the basin. In order to achieve this objective, a smaller area in the lower Siret Basin was chosen, area that has been affected by flooding several times, making it suitable for our purposes.

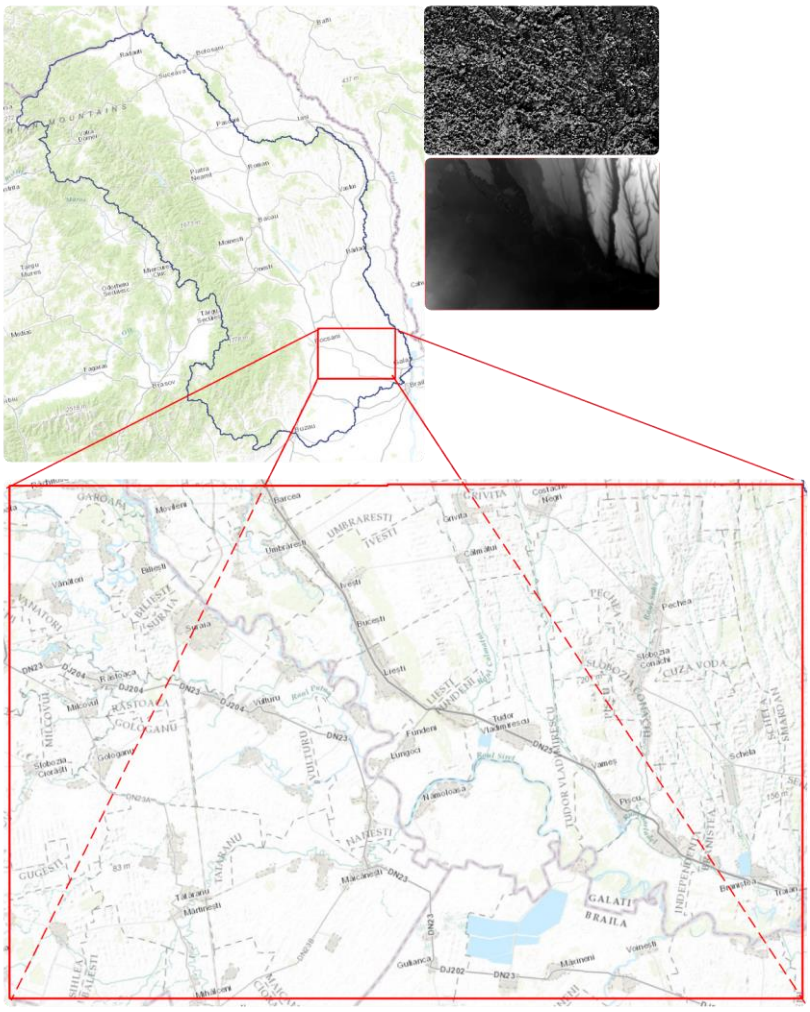


Figure 34 - Selected area for cellular automata analysis

In order to be able to transform it into arrays and ease the processing for further steps, a rectangular shape was chosen for this area, as seen in Figure 34 . When creating this polygon using the Create Rectangle option in the Create Feature toolset, it was important to ensure that it would be snapped to the raster cells in order to allow easy combination of the various datasets. After creating the 1st polygon, we expanded it by 100 meters in each direction to create the second polygon and used this newly created polygon to cut the DEM and Flow Direction raster datasets.

4 Implementation

As presented in the introduction chapter, the goal of this thesis is to create a system that can predict the areas where a fooding event might happen using only available data for the study area.

Following Bevens (2014) modeling diagram presented in the beginning of the theory chapter, the implementation chapter will present the first stages of the modeling process: The perceptual model, The conceptual model and The Procedural model. The end of the implementation model presents the last two stages: Model Calibration and Model Validation.

4.1 The perceptual model

This stage is represented by our perceptions on what processes are to be modeled based on literature and research. The main idea behind the model is to get data regarding the weather forecast in advance, for the next 3-7 days, and based on that data to model where the flooding might occur. A big problem when dealing with modeling is obtaining the data, especially in countries such Romania where free data in not yet available. Not having access to any local data, our attention had to be oriented towards free European datasets or other providers that can deliver data covering our study area. Moreover, many of the already available models for flood prediction are Energy balance models that have as characteristics a large and complex number of input data. Therefore, more simple methods that are based on easy to find data were necessary to find.

Based on this limitation, the decision was made that the model should focus on the most important processes, such as, precipitation as rain, snow accumulation and snowmelting and rain on snow events.

The hierarchy process was based at a conceptual level on the AHP method presented in the Theory Chapter and we considered that precipitation and snowmelt are first level factors that together contribute in the largest amount to the end result. The next level is represented by rain on snow events that contribute to the end results and in some situations might be extremely important. The rain on snow events are not on the first level, because their occurrence probability is quite low compared with rain and snow melting, which are considered sure processes.

4.2 The conceptual model

The perceptual model ends with the general idea of the processes that are to be modeled and the conceptual model takes the process further by finding the proper methods to represent modeler's ideas from the Perceptual model. In this case, the methods were already presented in Chapter 2 and they are as follows: Runoff from rain, runoff from snowmelt and rain on snow events, which are modeled using the first three methods presented in Chapter 2: The CN method, Day-Degree method and Rain on Snow equation. The model accounts for each of the tree processes at the same time and, for each process, it calculates the runoff value that is used to compute where all the water that results after each process is accumulated.

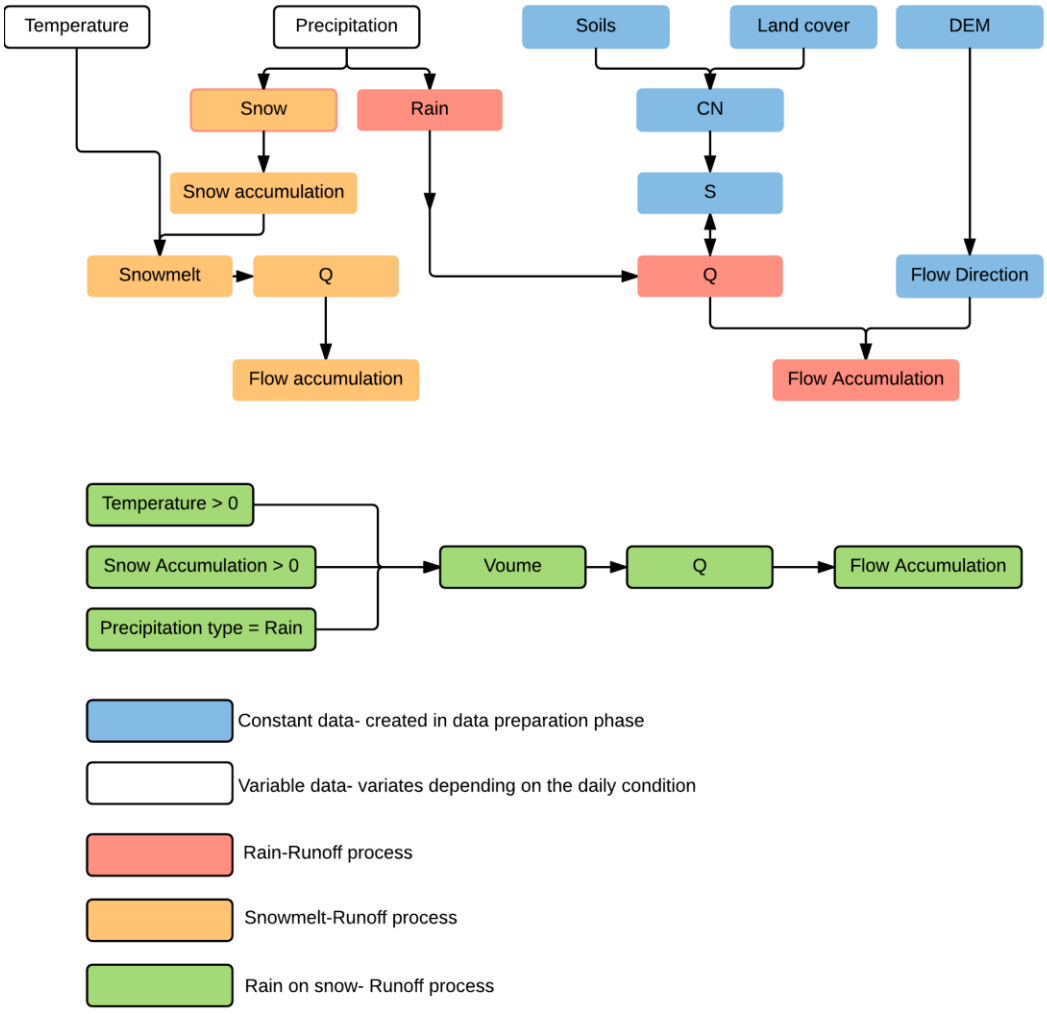


Figure 35 - Model Diagram

Having the runoff from all three processes, the cells where this runoff accumulates can be calculated using the Flow Accumulation method. Figure 35 above shows how the processes work and how everything is connected.

Further on, each of the three processes is described in detail, together with the equations used presented in the Chapter 2, and the data necessary and its preparation presented in Chapter 3.

4.2.1 Constant data

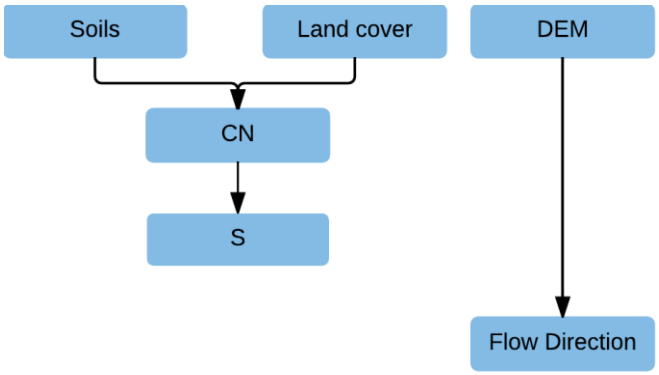


Figure 36 - Constant data representation

Constant data represented with blue color in Figure 35 and above in Figure 36 represents data gathered and prepared in advance that remains constant during the modeling process. Soil and land cover is used for generating the curve number values raster, named CN. Based on the CN raster, the maximum recharge capacity raster, named S, is created. These two operations, the creation of CN and S raster are the first steps in calculating the runoff value, named Q. The DEM represents the foundation of the calculations also being a constant data. The flow direction raster is created using the flow direction tool from ArcGIS and is based on the DEM. Chapter 3, Data Gathering and Preparation presents in detail the operations made to prepare soil raster, land cover, curve number raster, maximum recharge capacity raster and the DEM preparation.

4.2.2 Variable data

Temperature and Precipitation data, as shown in Figure 37, are the variable inputs of the model. The data is retrieved for the current day and the next 7 days from Forecast.IO, a website specialized in delivering

meteorological forecast. The website communicates with most known programing environments through an API (supplication program interface) which allows the user to write a request for retrieving the interest data. Based on this request, the website sends back the information.

Figure 37- Variable data representation

4.2.3 **Rain processes**

Rain processes, represented with red color in Figure 35 and Figure 38 below, is the part of the model that works with the runoff from direct precipitation as rain.

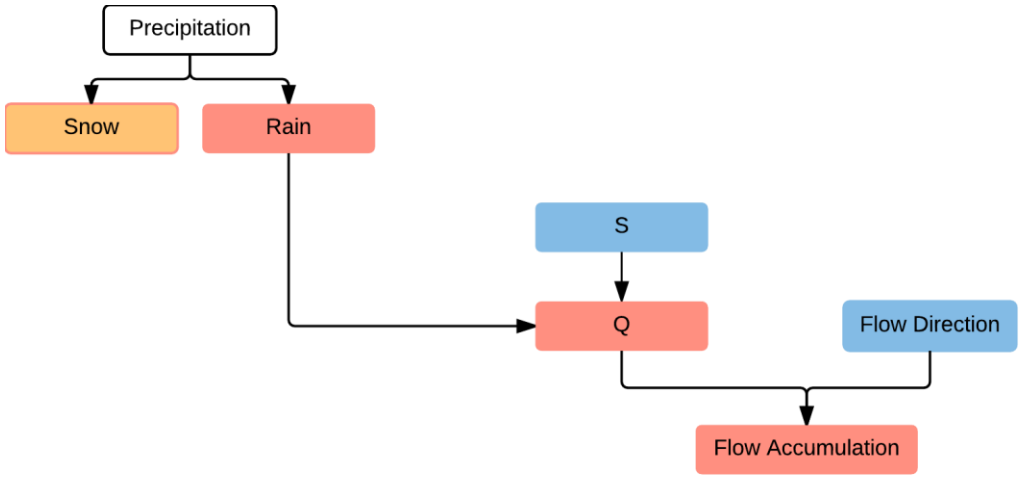


Figure 38 - Rain processes representation

Precipitation type, in this case rain or snow, is retrieved together with its value. If the of the precipitation is rain, the value is used for runoff computation using Equation 3 presented in Chapter 2.4.1 and once more bellow.

$$Q = \frac{(P - 0{,}2S)^2}{P + 0{,}8S}$$

Q represents the runoff value, P is the precipitation value as rain retrieved daily, and S is the maximum recharge capacity raster, considered constant. Having the runoff value and the flow

direction raster already calculated, the Flow Accumulation tool in ArcGIS is used, the output being a raster representing the area where the calculated runoff accumulates.

4.2.4 Snow melting processes

For snow melting computation, both temperature and precipitation are considered input data. The model accounts for precipitation that fell in the past as snow and calculates the accumulation until the moment when snow melting occurs. The model registers the snow accumulation as a water reserve and not the density of snow or snow coverage. During the snow melting season, the temperature increases and snow starts to melt. Using Equation 5 presented in Chapter 2.4.2 and once more below, the snowmelt that will become runoff is calculated and its value is subtracted from the total snow accumulation.

$$M = C_b(T_a - T_b)$$

M is the snowmelt in mm, C_b is a variable degree day factor, T_a is the temperature registered and T_b is the reference temperature. The equation is explained in detail in Chapter 2.4.2.

Figure 39 presents how the processes are connected and how the model deals with the operations. If the precipitation type is snow, then a snow accumulation is computed. The snow accumulation increases while the temperature is negative. When temperature is greater than 0 C^o, the result of Equation 2 is subtracted from the total snow accumulation. The resulted runoff from snow melting is further fed into flow accumulation as weight factor together with the runoff from rain using the flow direction raster as input.

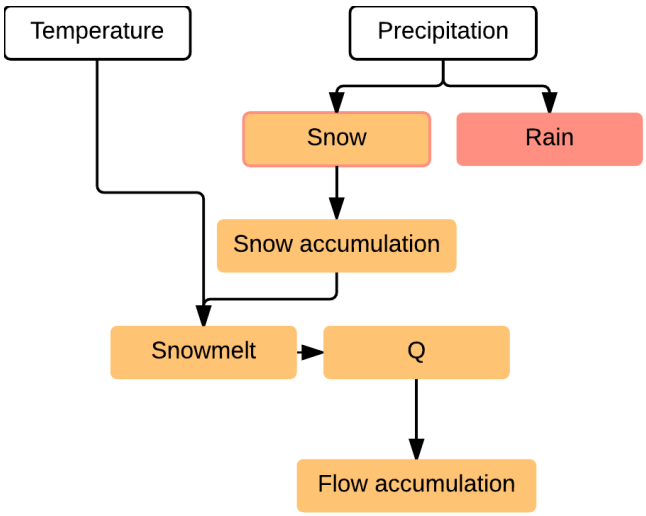


Figure 39 - Snow melting processes representation

4.2.4 **Rain on snow processes**

Rain on snow processes occurs more rarely than rain or snow melting but the outcome can have a big impact on flood forecasting and modeling (Linsley 1943). This process takes place when the temperature is positive, the ground is still covered in snow and precipitation as rain type is present. Figure 40 shows how the processes are connected in the model. Based on the three logical conditions, using Equation 7 presented in Chapter 2.4.3 and again below, the depth of the snowmelt due to rain is computed.

$$D = 0{,}007\ P\ T$$

D is the depth of snowmelt, P is the precipitation value and T represents temperature. The formula and the method is presented above and in detail in Chapter 2.4.3.

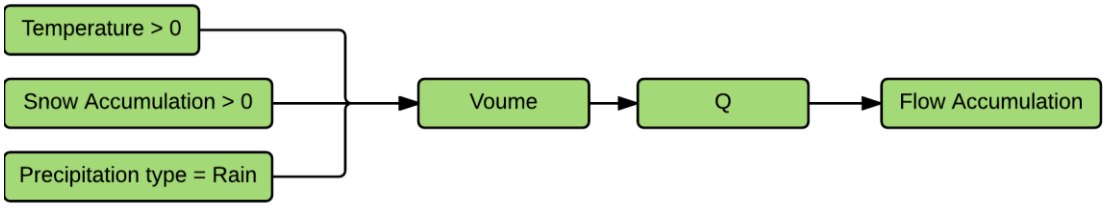


Figure 40- Rain on snow processes representation

The processes presented are computed for each day, and the Daily Flow Accumulation is run using as weight the sum of the runoff values produced by precipitation, snowmelt and rain on snow events. For flood predictions, data regarding temperature and precipitation is retrieved for the next 7 days and for each day the Daily Flow Accumulation is created. Having these layers, the accumulation value for the period of interest can be calculated by adding the flow accumulation layer for the respective days.

4.2.5 **RACA Method**

Flow accumulation results so far only point out the areas where the runoff eventually accumulates but this quantity of water is not significant, unless it is spread on top of the DEM in the region pointed by flow accumulation result.

Because of the high computational demands, RACA cannot be run for the entire Basin. In order to avoid this problem, flow accumulation is used to calculate the input of the rest of the Basin for our test area, and inside our test area we combine a few cells from flow accumulation and the actual runoff values in order to calculate the input from the entire basin, as seen below in Figure 41. The values from flow accumulation (represented by blue color) surrounds the value representing the total runoff for the area (represented by orange color). In this way, RACA method can be applied for a smaller area considering in the same time the input from the whole basin.

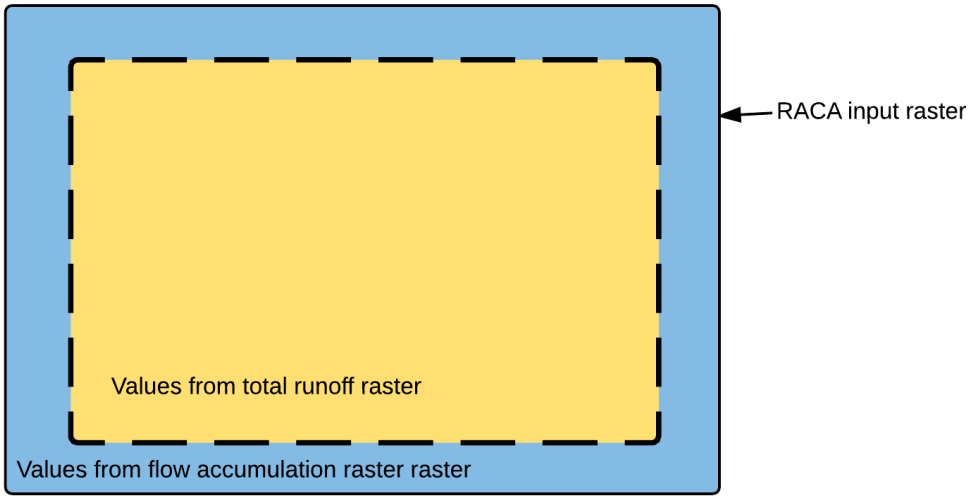


Figure 41- Input raster structure for RACA

The script can then run a fixed number of iterations, or until there is no change in the resulting water level raster between the last 2 iterations.

The next step is to transpose the conceptual model into code that can be automatized and run independently on a computer. This step is explained further in the Procedural model.

4.3 The Procedural model

In order to achieve all the steps outlined in the previous chapters, several Python scripts were written to process the data automatically. Figure 42 below presents a brief overview of the scripts and their relation with each other. The first script, “Create precipitation file” is presented in Chapter 3.2 and was used for creating the table structure that later holds the values for the data and does not have an active role in day to day processing. On the contrary, all the other scripts are active for every single output that the model delivers. “Populate precipitation file” script uses the API to retrieve data and populates the table created in data preparation phase. Based on this table, “Create precipitation, temperature and snow accumulation raster” script interpolates the values retrieved by the previous script, and creates different raster files for precipitation, temperature and snow accumulation for each day. The next two scripts “Create runoff from precipitation” and “Snowmelt” compute the daily runoff using the methods presented in Chapter 2.4. “Flow accumulation” script adds the runoff values together and uses this value as weight and computes flow accumulation. Further on, the “RACA” script clips the daily total runoff and the flow accumulation and computes the RACA algorithm. The resulting raster are further fed into our webserver which creates tiles and serves them to the web map.

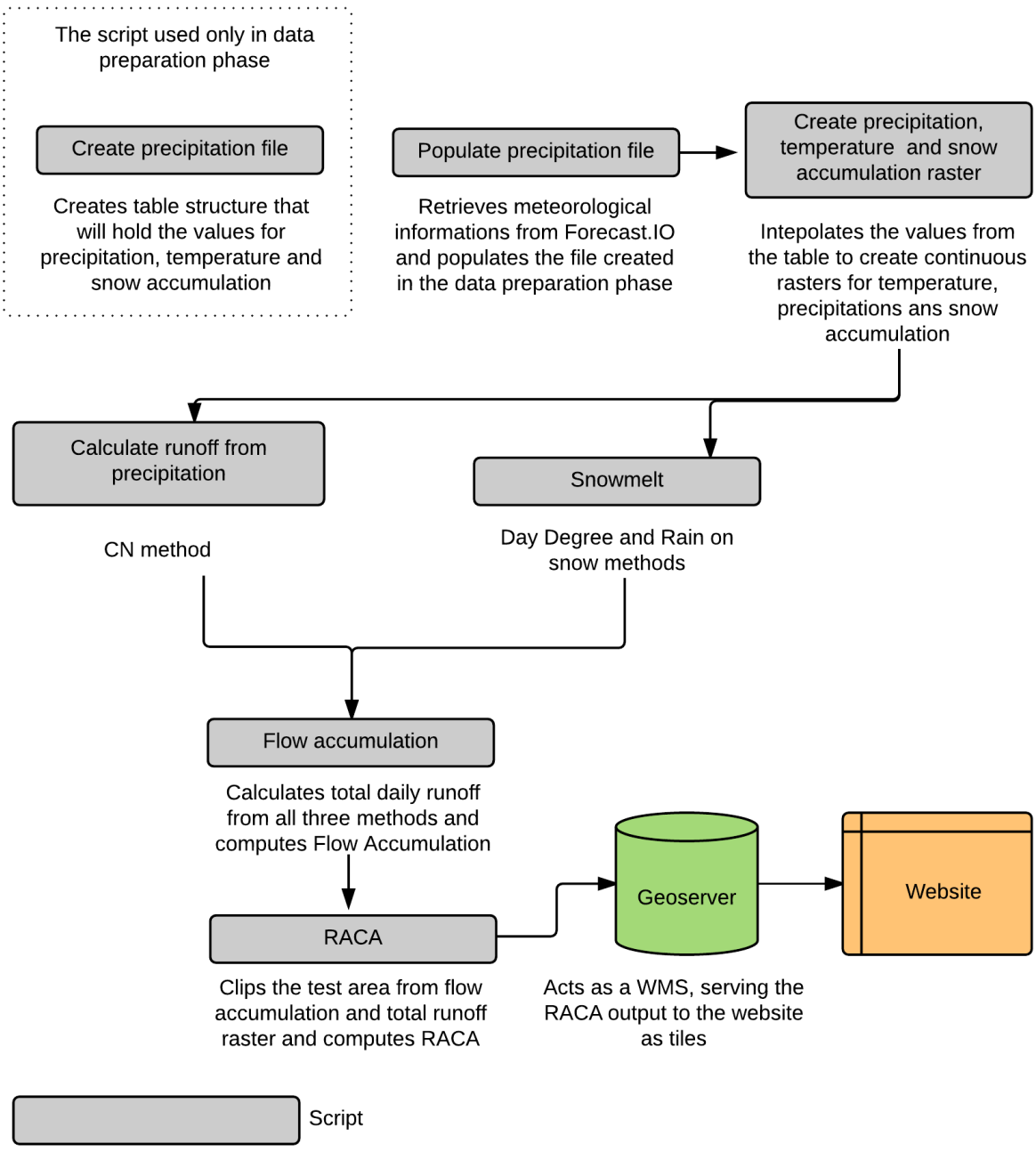


Figure 42- Overview of the processing chain

Further, the following subchapters explain the key points in the scripts structure, their complete content being attached in Appendixes 1 - 8.

4.3.1 Populate precipitation table script

This first script is responsible of retrieving the precipitation data from Forecast.IO through the use of their API and a custom python wrapper written by github user DVDME. By using this wrapper, we can use python commands that get parsed by the wrapper to the API, with the wrapper also returning the data in a usable format inside Python. This allows us to place the data where we want while also allowing us to perform certain operations in an easy and fast way.

The first step for every script is the import of the modules that will be used inside each script. Most scripts use the same modules, with only certain changes. In this case we also import forecastiopy to allow us to retrieve meteorological data (Figure 43) .We also set the environment, avoiding the need to write absolute paths for each file.

```
1  from forecastiopy import *
2  import datetime
3  import arcpy
4  from arcpy import env
5
6  env.workspace = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb"
7  fc = "Precipitations"
8
```

Figure 43 - Import statement for populate precipitation table script

The next step is to create a loop that will add all field names inside a dictionary. This dictionary will then be used to retrieve only the columns we want to edit from the precipitations table (Figure 44). We use the datetime module we imported earlier to create the required strings matching the column names.

```
13 #Create a list with the name of the fields required for the search cursor
14 fields = ['OBJECTID', 'LAT', 'LONG']
15 for days in range(0, 7):
16     d = datetime.date.today() + datetime.timedelta(days)
17     datePrec = "Prec_" + str(d)
18     dateAcum = "Acum_" + str(d)
19     dateTemp = "Temp_" + str(d)
20     fields.append(datePrec.replace("-", "_"))
21     fields.append(dateAcum.replace("-", "_"))
22     fields.append(dateTemp.replace("-", "_"))
```

Figure 44 - Column name dictionary

We also retrieve the object ID, latitude and longitude values that will be used later on. The next step is to create a search cursor that will loop through each row of the table, retrieve the required information (latitude and longitude) and create the API call as presented in Figure 45 . The API call contains the latitude and longitude information obtained earlier, the API key which ensures that the receiver can identify us and can count the number of calls we make. All other parameters of the call are provided by the wrapper. It is important that we mention the units, in our case SI units, to ensure that we use the same units everywhere. By using the SI argument, we make sure we use SI units, in case of precipitations the values being in mm/hour and for the temperature degrees Celsius. We chose SI units because they are the most widely used system of measurements, with Romania being one of the users.

```
#Search Cursor loop
with arcpy.da.UpdateCursor(fc, fields) as cursor:
    for row in cursor:
        #Retrieve coordinates for forecast API call
        lat = row[1]
        long = row[2]
        #Forecast API call
        fio = ForecastIO.ForecastIO(api_key,
                                    units=ForecastIO.ForecastIO.UNITS_SI,
                                    lang=ForecastIO.ForecastIO.LANG_ENGLISH,
                                    latitude=lat, longitude=long)
```

Figure 45 - Input of daily data in the precipitation table

```
        #Check if forecast object has daily data
        if fio.has_daily() is True:
            daily = FIODaily.FIODaily(fio)
            try:
                #Check precipitation type and place data acordingly for each day
                if daily.day_1_precipType == 'snow':
                    row[3] = 0
                    cursor.updateRow(row)
                    row[4] = daily.day_1_precipIntensity
                    cursor.updateRow(row)
                    row[5] = (int(daily.day_1_temperatureMin)+int(daily.day_1_temperatureMax))/2
                    cursor.updateRow(row)
                else:
                    row[3] = daily.day_1_precipIntensity
                    cursor.updateRow(row)
                    row[4] = 0
                    cursor.updateRow(row)
                    row[5] = (int(daily.day_1_temperatureMin)+int(daily.day_1_temperatureMax))/2
                    cursor.updateRow(row)
            except:
                row[3] = 0
                cursor.updateRow(row)
                row[4] = 0
                cursor.updateRow(row)
                row[5] = (int(daily.day_1_temperatureMin)+int(daily.day_1_temperatureMax))/2
                cursor.updateRow(row)
                pass
```

Figure 46 - Search cursor

This call should return a JSON object, containing all the required information. JSON stands for JavaScript Object Notation and is a widely used format for transmitting data consisting of attribute-value pairs (Documentation PSF 2016). According to Forecast.IO, information might not be available all the time for every region, therefore we make sure to check the object exists before attempting to do anything with the data. If there is no data available, the script returns an error warning us that there is no daily information. To avoid the issue of only specific points not having any data, we wrap all our statements in a try/except statement (Figure 46). This ensures that if no data is available for the precipitation, we assume there is none and input a value of 0. Since there is no value for snow, but we have a field with the precipitation type, we create an if/else statement that places the information in different columns based on the precipitation type. For the daily temperature, Forecast.IO does not provide an average temperature, thus we use the daily minimum and daily maximum to calculate an average that will be used further on. This process is repeated for each of the following 7 days, using the same procedure.

4.3.2 Creating precipitation, snow and temperature raster files

The next step is the interpolation of the data inside the table. This is done using a different script that calls on the ArcPy tools available to us. Since we are working with ArcGIS tools and raster files, there are a few other settings we need to make in order to ensure that everything runs smoothly and no errors are present (Figure 47).

```
3    arcpy.env.overwriteOutput = True
4    arcpy.CheckOutExtension("GeoStats")
5    arcpy.env.workspace = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb"
6    arcpy.env.mask = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb/Basin_Boundary"
7    arcpy.env.snapRaster = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb/Boundary"
```

Figure 47 - Raster creation environment settings

We first enable the overwriting of outputs. Because we create data for 7 days in advance, we will be creating the same file several times. We do this because, as we approach the date, the predictions are increasingly accurate. We then activate the extension containing the interpolation tools and set the environment settings. The environment settings ensure that the resulting raster will have the same extent, resolution and that the cells will overlap. This ensures easier processing further down the line.

We use the GeoStatistical extensions because inside this toolset, the interpolation methods allow for greater control over the interpolation. At this point in time, default values are used, but during the process of calibration, these values can be improved.

```
arcpy.IDW_ga(Precip, str(Field_name_prec), "", "C:/xampp/htdocs/FloodSite/Data/Rasters/Precip/"
             + str(Field_name_prec) + ".tif", 100, 2,
             arcpy.SearchNeighborhoodStandard(30000, 30000, 0, 15, 10, 'ONE_SECTOR'), )
arcpy.IDW_ga(Precip, str(Field_name_acum), "", "C:/xampp/htdocs/FloodSite/Data/Rasters/Snow/"
             + str(Field_name_acum) + ".tif", 100, 2,
             arcpy.SearchNeighborhoodStandard(30000, 30000, 0, 15, 10, 'ONE_SECTOR'), )
arcpy.IDW_ga(Precip, str(Field_name_temp), "", "C:/xampp/htdocs/FloodSite/Data/Rasters/Temp/"
             + str(Field_name_temp) + ".tif", 100, 2,
             arcpy.SearchNeighborhoodStandard(30000, 30000, 0, 15, 10, 'ONE_SECTOR'), )
```

Figure 48 - IDW Interpolation

We create a similar loop like in the previous script and use the datetime module to create the name strings. We then access the tools and output the results as .tiff files, each type belonging to a specific folder for easier management (Figure 48). As mentioned in other chapters, we chose the IDW interpolation method because it is the fastest one available and it returns similar results when compared with the other methods available.

4.3.3 **Calculate runoff from rain**

After interpolating the rain values, we use those values to calculate the actual runoff using the CN method presented in Chapter 2. We proceed in a similar fashion as with the other scripts, by creating a loop for the next days and using datetime to retrieve the file names as strings containing the date. These will be the precipitation raster files. We also import the file containing the S values for our watershed as it will be used as input (Figure 49).

```
for num in range(0, 7):
    Date = datetime.date.today() + datetime.timedelta(days=num)
    date = str(Date)
    Field_name_prec = 'Prec_' + date.replace("-", "_")
    Prec = "C:/xampp/htdocs/FloodSite/Data/Rasters/Precip/" + str(Field_name_prec) + ".tif"
    S = Raster("S005")
    P = Raster(Prec)
    Q = pow(P - (0.05 * S), 2)/(P + (0.95*S))
    Qfinal = Con((P<0.05*S), 0, Q)
    Qfinal.save("C:/xampp/htdocs/FloodSite/Data/Rasters/Q05/Q_" + date.replace("-", "_") + ".tif")
    print("Q calculated")
```

Figure 49 - Q calculation using the CN method

The actual calculation done in line 15 is achieved using both python and ArcPy tools. Even though pure python commands would be faster, we wanted to ensure that the calculations are done on a cell by cell basis and thus used a hybrid of the two type of methods.

The next line contains a conditional statement that sets the runoff value to 0 if the precipitation value is smaller than 0.05S.

4.3.4 **Snowmelt**

In the theory chapter, we presented 2 methods for calculating snowmelt, the day degree method dealing with snowmelt due to the temperature and the rain on snowmelt equation. Both methods are included inside the same script as they both deal with snowmelt, but in order to make the structure of our files easier, the result of each method will be a different file that will be later added with precipitation runoff file to return total runoff.

The first step is calculating the coefficient for the day degree method. Since the values vary based on the day, we decided to create a range of days within which the coefficient would be calculated. By using the range, we can compute the daily rate by which the coefficient will increase. As mentioned in the theory chapter, the day degree factor varies between open areas and forest areas, thus we take this into account and calculate the daily rate for each scenario (Figure 50). This daily rate will by multiplied by the number of the day within the snow melting season range, resulting in the actual coefficient.

```
11  #Calculate coefficient for day degree method
12  Days = datetime.date(2016, 6, 1) - datetime.date(2016, 4, 1)
13  Days = int(Days.days)
14  Melt_forest = 3.7 - 1.85
15  Melt_open = 7.4 - 3.7
16  Rate_forest = Melt_forest / Days
17  Rate_open = Melt_open / Days
```

Figure 50 - Daily snowmelt coefficient increase

We then again proceed to creating the filename strings inside a loop to run the script for the next 7 days. Since we will need all of them for the equations, we import the precipitation, temperature and snow accumulation raster files, created earlier.

After loading all the files, we add the daily snow accumulation to the total accumulation raster and proceed to calculating the amount of snowmelt. We add the snow accumulation to a master file because, during winter times where temperatures are below 0° C ,no snowmelt occurs and the values just add up.

```
#Rain on snow melting
Rainonsnow_melt = Raster(Prec) * Raster(Temp) * 0.007
RainOnSnow_melt = Con(Rainonsnow_melt > 0, Rainonsnow_melt, 0)
RainOnSnow_melt.save("RainOnSnow_melt")
arcpy.CalculateStatistics_management("RainOnSnow_melt")
arcpy.Minus_3d("NewSnow",  "RainOnSnow_melt", "Snow_Diff")
SnowDiff_neg = Con(Raster("Snow_Diff") > 0, "Snow_Diff",0)
SnowDiff_neg.save("SnowDiff_neg")
arcpy.Minus_3d("NewSnow",  "SnowDiff_neg",
               "C:/xampp/htdocs/FloodSite/Data/Rasters/SnowMelt/SnowOnRain_melt_"
               + date.replace("-", "_") + ".tif")
arcpy.Minus_3d("NewSnow", 'C:/xampp/htdocs/FloodSite/Data/Rasters/SnowMelt/SnowOnRain_melt_'
               + date.replace("-", "_") + ".tif",
               "Tot_Snow_Today")
```

Figure 51 - Rain on Snow melting

Figure 51 above displays the rain on snow melting part of the script. We begin by calculating the Rain on snowmelt equation and then using a conditional to remove negative values. The result of this operation will be the maximum amount of snow that can melt. We also need to take into account the amount of snow already available so we use a series of subtractions in order to calculate the actual amount of snowmelt and the remaining amount of snow.

After calculating the rain on snow melting, we calculate the snowmelt from the day degree method. Figure 52 shows this process. We first calculate the number of days that passed since the beginning of the snowmelt season and use that number to multiply the daily factor calculated at the beginning of the script. This will result in the actual C_b coefficient mentioned in the theory chapter. We then use the conditional tool to create a new layer that contains the coefficient values. We use the CN files because we know that values of 25 in the CN file correspond to forests. All other values take the open area coefficient. After calculating the coefficient, we calculate the snowmelt value and remove negative values. We then proceed again with a series of subtractions and conditionals to ensure that the value is subtracted from the accumulated snow and do not create more snowmelt than existing snow.

```
# DayDegree Snowmelt
Number_of_days = Date - datetime.date(2016, 4, 1)
days = str(Number_of_days)
Coefficient_open = int((days[:2])) * Rate_open
Coefficient_forest = int((days[:2])) * Rate_forest
Coefficient_layer = Con("CN005" == 25, Coefficient_forest, Coefficient_open)
Day_degree_snowmelt = Times(Coefficient_layer, Temp)
SnowMelt = Con(Day_degree_snowmelt > 0, Day_degree_snowmelt, 0)
SnowMelt.save("DayDegree_Snowmelt")
arcpy.Minus_3d("Tot_Snow_Today", "DayDegree_Snowmelt", "Snow_Diff")
DD_Diff = Con(Raster("Snow_Diff") > 0, "Snow_Diff", 0)
DD_Diff.save("DD_Diff")
arcpy.Minus_3d("Tot_Snow_Today", "DD_Diff",
               "C:/xampp/htdocs/FloodSite/Data/Rasters/SnowMelt/DayDegree_Snowmelt_"
               + date.replace("-", "_") + ".tif")
arcpy.Minus_3d("Tot_Snow_Today",
               "C:/xampp/htdocs/FloodSite/Data/Rasters/SnowMelt/DayDegree_Snowmelt_"
               + date.replace("-", "_") + ".tif",
               "C:/xampp/htdocs/FloodSite/Data/Rasters/Tot_Snow_Today.tif")
DD_Diff = Con("Snow_Diff" > 0, "Snow_Diff", 0)
arcpy.CopyRaster_management("C:/xampp/htdocs/FloodSite/Data/Rasters/Tot_Snow_Today.tif",
                            "Tot_Snow_Today")
print(date + " snowmelt has been processed")
```

Figure 52 - Day Degree method

4.3.5 **Flow Accumulation**

After calculating the runoff, the next step is to add them up and calculate the flow accumulation for the entire basin. These 2 raster files will be used inside the RACA script. The first step is to add the resulting runoff from the rain and snowmelt scripts to obtain the total runoff inside the basin (Figure 53). After calculating the total runoff, we save it, to have it as backup in case we might need it at a later date and proceed to calling the flow accumulation function trough ArcPy. We use the flow direction raster, and the newly calculated runoff raster(Q) as weight, while also ensuring the data will be of FLOAT type.

```
for num in range(-11,7):
    Date = datetime.date.today() + datetime.timedelta(num)
    date = str(Date)
    Q = 'Q_' + date.replace("-", "_")
    RainQ = "C:/xampp/htdocs/FloodSite/Data/Rasters/Q05/" + str(Q) + ".tif"
    SnowMelt = "C:/xampp/htdocs/FloodSite/Data/Rasters/SnowMelt/DayDegree_Snowmelt_" \
               + date.replace("-", "_") + ".tif"
    RainOnSnow = "C:/xampp/htdocs/FloodSite/Data/Rasters/SnowMelt/SnowOnRain_melt_" \
                 + date.replace("-", "_") + ".tif"
    arcpy.Plus_3d(SnowMelt, RainOnSnow, "TotalSnowMelt")
    arcpy.Plus_3d(RainQ,"TotalSnowMelt","C:/xampp/htdocs/FloodSite/Data/Rasters/TotalQ/TotalQ_"
                  + date.replace("-", "_") + ".tif")
    inFlowDirRaster = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb/FlowDir2100m"
    # Execute FlowDirection
    outFlowAccumulation = arcpy.sa.FlowAccumulation(inFlowDirRaster,
                                                    "C:/xampp/htdocs/FloodSite/Data/Rasters/TotalQ/TotalQ_"
                                                    + date.replace("-", "_") + ".tif", "FLOAT")
    #  Save the output
    outFlowAccumulation.save("C:/xampp/htdocs/FloodSite/Data/Flow_accum_rain05.gdb/FlowAcum_Rain_"
                             + date.replace("-", "_"))
    print(date + " was processed")
```

Figure 53 - Flow Accumulation and runoff addition

4.3.6 Rainfall with cellular automata (RACA) script

This script varies somewhat from the previously written scripts, because of the use of several new modules. The rules of the RACA method have been presented in previous chapters, and this chapter will only deal with the programing and computational aspects. As mentioned, we use several new modules in this script. One of them is Numpy, a python module developed for mathematical and statistical processing. By converting the raster files into Numpy arrays we can perform several operations. Even though the script uses Numpy arrays for processing of the data instead of the original raster datasets, the code is still pythonic in nature, and uses very little the Numpy commands. By switching the raster datasets to Numpy arrays the speed of the loops have been increased significantly. The speed can be further enhanced by using Numpy specific syntax and vectorising the arrays and the calculations used. Unfortunately, this is beyond our skill, as Numpy, although being a python module, has significantly different syntax.

```
import arcpy
import datetime
from arcpy import env
from arcpy.sa import *
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("Spatial")
import numpy as np
import itertools

#Set envionment settings and load DEM and FlowDirection rasters
env.workspace = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb"
inDEM = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb/DEMSiret"
flowdir = "C:/xampp/htdocs/FloodSite/Data/Precipitations.gdb/FlowDir2100m"

#Load rasters as numpy arrays
DEM = arcpy.RasterToNumPyArray(inDEM)
FlowDir = arcpy.RasterToNumPyArray(flowdir)
```

Figure 54 - RACA script modules and environment settings

Figure 54 presents the beginning of the script. It starts in the same manner as our other scripts by calling the python modules to be used inside the script. An additional module to our previous script is imported, namely itertools, which is responsible for creating the iterations of our Cellular Automata. It has several advantages over traditional “for” loops, the main benefit being speed and better memory management.

```
for num in range(-133,7):
    #Set environment and date variables to import DEM and Runnoff rasters
    current_date = datetime.date.today() + datetime.timedelta(days=num)
    date = str(current_date)
    # Prepare files for RACA
    # Cut the flow accumulation and Runoff rasters
    arcpy.Clip_management("C:/xampp/htdocs/FloodSite/Data/Flow_accum_rain05.gdb/FlowAcum_Rain_"
                          + date.replace("-", "_"), "5649300 2638400 5706700 2675800",
                          "Flow_clip", "AOI_Big_polygon", "-3.402823e+038",
                          "ClippingGeometry", "MAINTAIN_EXTENT")
    arcpy.Clip_management("C:/xampp/htdocs/FloodSite/Data/Rasters/TotalQ/TotalQ_"
                          + date.replace("-", "_") + ".tif", "5649300 2638400 5706700 2675800",
                          "Q_clip", "AOI_Small_polygon", "-3.402823e+038",
                          "ClippingGeometry", "MAINTAIN_EXTENT")
    RACA_raster = Con(Raster("Conditional") > 1, Raster("Q_clip"), Raster("Flow_clip"))
    RACA_raster.save("C:/xampp/htdocs/FloodSite/Data/Rasters/RACA/RACA_"
                     + date.replace("-", "_") + ".tif")
    inRaster = "C:/xampp/htdocs/FloodSite/Data/Rasters/RACA/RACA_" \
               + date.replace("-", "_") + ".tif"
```

Figure 55 - Beginning of loop and creation of files

The next few lines (Figure 55) form a loop that loads the runoff files for a range of days. After loading both the flow accumulation and runoff raster, we proceed to combining them. Using a conditional raster that has the values 0 in the center where we want the runoff values, and 1 in the sides where we want to keep the values from the flow accumulation. This process is done inside the loop because we want to repeat it for each day.

The DEM and Flow Direction files are loaded outside the loop as they are the same for each iteration of the loop. The transformation to Numpy arrays is also done outside the loop. This ensures that the process is not done with each iteration, thus saving time and improving performance. It is important to note that the DEM file has been modified to be in mm, in order to coincide with the values in the water_level raster. This was done by multiplying the original DEM by 1000.

```
21
22      #Retrieve raster properties for later export
23      dsc=arcpy.Describe(inRaster)
24      sr=dsc.SpatialReference
25      ext=dsc.Extent
26      ll=arcpy.Point(ext.XMin,ext.YMin)
27
28      #Import Accumulation raster as numpy array
29      Accumulation = arcpy.RasterToNumPyArray(inRaster)
30
```

Figure 56 - Spatial properties of accumulation raster

Figure 56 shows an important part of the script. By importing the raster datasets into Numpy array all spatial properties of the raster are lost. By retrieving them using ArcPy commands, we can later use them to save our new water levels raster with the same properties. The properties required for this are the spatial reference, extent and minimum values for X and Y. The raster dataset is then transformed to a Numpy array for further processing.

We then retrieve the shape of the newly created Numpy array and proceed to creating a new array called N_water_level that has the same shape as our old raster (Figure 57). This ensures that the rasters use the same indexing and prevents further errors.

```
#Create WaterLevel change Array
[rows,cols]= Accumulation.shape
N_water_level = np.zeros((rows,cols), np.float)
```

Figure 57 - Creating the water level array

The next steps involve creating a series of nested loops that use the flow direction array to determine the lowest neighboring cell and then calculate the water flow according to the formulas presented earlier in this thesis and also in (Espinola, et al. 2014). The direction coding provided by ESRI, also presented in the theory chapter(Figure 13) is used to calculate the position of all the cells neighbors (Figure 58).

```
try:
    if flow == 32:
        Water_Flow = ((DEM[i, j] + Accumulation[i, j]) -
                      (DEM[i - 1, j - 1] + Accumulation[i - 1, j - 1])) / 2
        if Water_Flow < Accumulation[i, j]:
            N_water_level[i, j] = Accumulation[i, j] - Water_Flow
            N_water_level[i - 1, j - 1] = Accumulation[i - 1, j - 1] + Water_Flow
        else:
            N_water_level[i, j] = 0
            N_water_level[i - 1, j - 1] = Accumulation[i - 1, j - 1] + Accumulation[i, j]
```

Figure 58 - Applying RACA rules

The calculation and flow direction coordinates are wrapped in a series of if/else commands to ensure that the proper formula is applied to each situations (Figure 58). This is done for each flow value using the if/elif statements. The difference between them is the fact that when using if/else you can only account for 2 situations, while the elif command allows the use of any number of arguments for each function.

All these nested loops are wrapped inside a try/exception clause. The reason behind this is to avoid the script from exiting in case of an error. In our case we want to ignore any potential Index Errors that might

cause the program to crash. The reason we might receive an Index Error is the fact that at the edge of the array, the final cell does not have any neighbors, but the script will still try to search for them and attempt to create the calculations. If this situation arises the script normally throws an Index Error and exits the program. This situation is prevented by using the exception clause. In order to prevent it from ignoring any other errors, the clause is given an argument stating that it can only ignore Index Errors.

```
Accumulation_raster = arcpy.NumPyArrayToRaster(Accumulation, ll,
                                               dsc.meanCellWidth, dsc.meanCellHeight)
Accumulation_raster.save("C:/xampp/htdocs/FloodSite/Data/Rasters/RACA/RACA_"
                         + date.replace("-", "_") + ".tif")
arcpy.CopyRaster_management("C:/xampp/htdocs/FloodSite/Data/Rasters/RACA/RACA_"
                         + date.replace("-", "_") + ".tif",
                            "C:/xampp/htdocs/FloodSite/Data/Rasters/Website/RACA_day"
                            + str(num) + ".tif", "onebit_to_eightbit")
print(date +" done")
```

Figure 59 - Export of the array back to raster

In Figure 59 we can see the new water levels file being saved to disk. This process occurs outside the loop, as it is the final version. Here we also create a copy of the file and name it accordingly to the initial loop of 7 iterations for 7 days. This is to ensure a consistent naming schema for the files that will be used inside our webpage.

4.4 **Model calibration**

CN method is a fairly old method developed by USDA and applied in small, experimental watersheds. The parameters present in the paper were also developed for determining runoff in small watersheds. In the recent years, a number of studies have investigated the parameters used by the original method in order to make it more accurate and more applicable for larger watersheds.

The initial abstraction ratio, presented in the theory chapter as I_s was created as having a value of 0.2S but its origins are not well documented in the original document from the 1950s.

Hawkins, et al. (2002) have made the most comprehensive study where they analysed the runoff in a number of 307 of different sized watersheds trying to find a better value for the initial abstraction. The original 0.2 value is responsible for big runoff values that result after applying the method for larger watersheds. Hawkins, et al. (2002) study indicates that a much smaller value of 0.05S is a way better fit

for determining the runoff value, especially for when the precipitation value is low. In 90% of the cases a much lower value than 0.2 was a better fit. Using the new 0.05 value he found that in 5 in 6 cases the fitting is better, 252 of the 307 cases analyzed.

The CN values presented in the USDA tables are suited for 0.2 abstraction factor, therefore in order for a new value to be used, the CN values needs to be changed. Based on the original methods, Hawkins, et al. (2002) developed a new set of equations in order to account for the lower abstraction factor. Equations 8,9,10 below presents the new proposed equations based on the old ones that were using the 0.2 value.

$$CN_{0.05} = \frac{100}{1.879\left[\frac{100}{CN_{0.20}} - 1\right]^{1.15} + 1}$$

Equation 8 - New proposed formula for CN

$CN_{0.05}$ represents the new value that is to be computed and $CN_{0,20}$ is the original value. Having the new CN values calculated, the new maximum recharge capacity or S can be recalculated using Equation 9 or using the formula presented in the theory chapter.

$$S_{0.05} = 1.33S_{0.20}^{1.15}$$

Equation 9 - New formula for S(recharge capacity)

With the new abstraction ration of 0.05, the runoff formula is:

$$Q = \frac{(P - 0.05\,S_{0.05})^2}{P + 0.95\,S_{0.05}} \qquad P \geq 0.05S$$

$$Q = 0 \qquad P < 0.05S$$

Equation 10 - New runoff formula

Hawkins's, et al. (2002) lower abstraction factor was used in studies all over the world by (Kyoung, et al. 2006), (Manoj and Mrugen 2014), Tae-Wong, et al. (2014), (Baltas, Dervos and Mimikou 2007) with better results than the initial 0.2.

4.5 Model validation

4.5.1 Comparing initial abstraction factor with new value

As mentioned in the Chapter 2.4.1, the CN method uses an abstraction factor to calculate the runoff. This abstraction factor influences the values significantly thus it is important that it has the right value. Originally, studies used the abstraction factor of 0.2, but newer research suggests that this value, especially for large watersheds, results in an overestimation of the runoff (Hawkins, et al. 2002) (Tae-Woong, et al. 2014).

Our original models used an abstraction value of 0.2 just as it is presented in the original documentation. This led to some very large values, that we knew not to be true, and as a result we changed the value to the one suggested by much recent literature. This subchapter will compare the two abstraction factors and their influence on the results by comparing a day with rainfall, and checking the runoff and flow accumulation results.

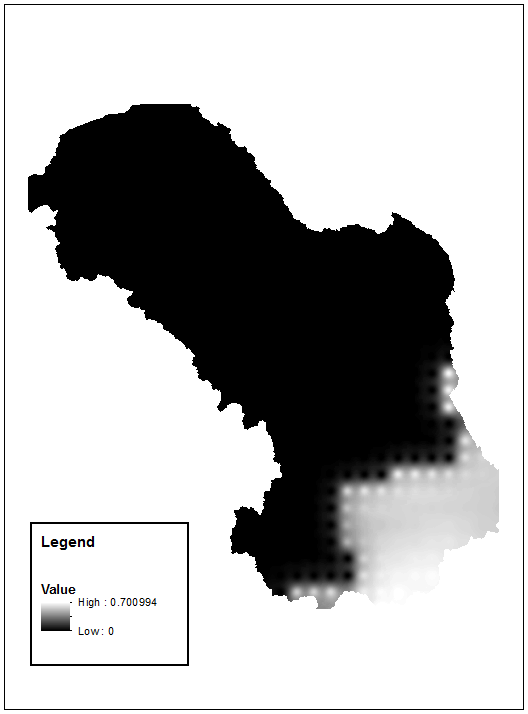


Figure 60 - Precipitation raster for 8th of January

The date chosen for this comparison was 8th of January 2016. This was because on this date, precipitations had a high value in the south area of the Basin. We wanted to choose a day with fairly high precipitations values so that we could exemplify the influence of the abstraction factor on runoff.

Figure 60 shows the precipitation raster for that day. As it can be seen the values are up to 0.7 mm an hour in the southern part.

Based on these precipitation values, the runoff values were calculated using both abstraction values. We started by using the original abstraction value, 0.2 (Figure 62). As it can be seen the values are already quite high, being up to 15 mm per cell. The high value runoff is even more obvious when compared to the runoff values obtained with a 0.05 abstraction value (Figure 61). In the second runoff calculation these values are only as high as 0.23 mm. While 15 mm seems as a high value, there is no objective way to determine that, thus we proceeded to calculating the flow accumulation

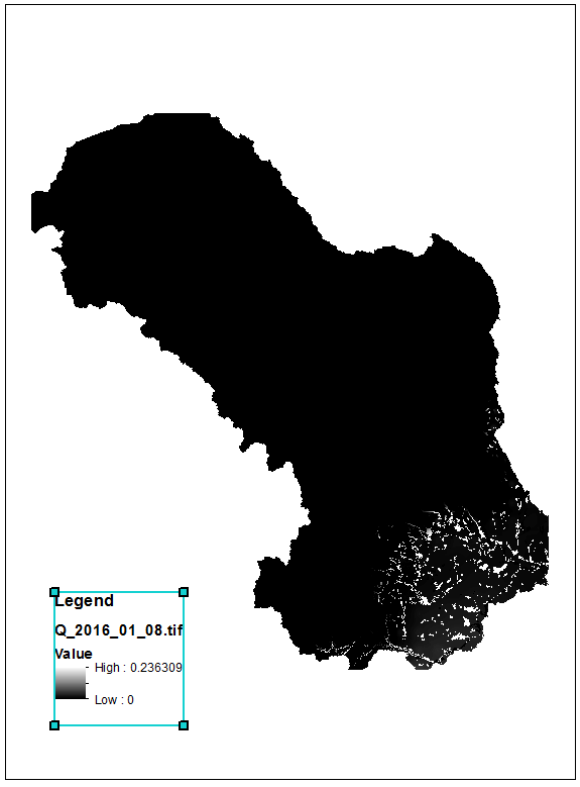


Figure 61 - Runoff calculated with a 0.05 abstraction value

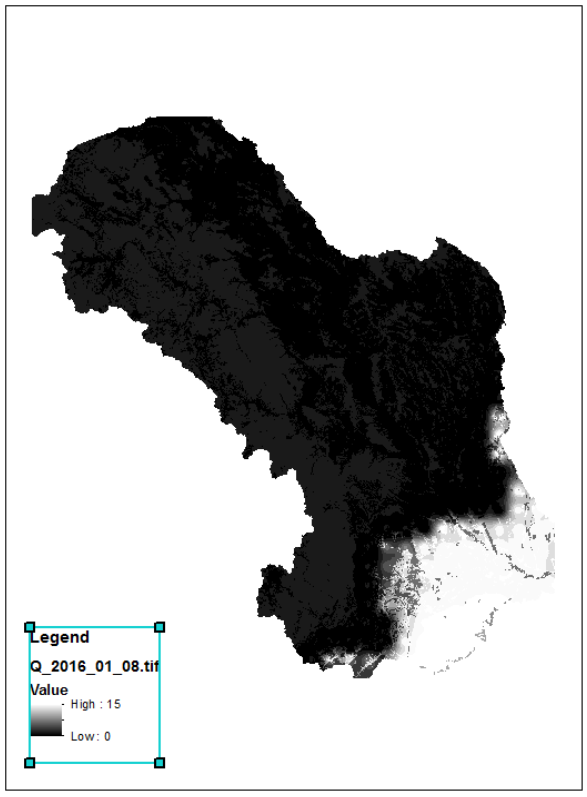


Figure 62 - Runoff calculated with a 0.2 abstraction value

By calculating the flow accumulation, we can get a better idea of what the values compare to real life. When doing this it is important to keep in mind the units of our data. What we have are mm/hour, but while they might still seem high after dividing them by 3600 to account for the fact that our precipitation is hourly, they seem more realistic. For example, Figure 63 which shows the Flow Accumulation value for the 0.2 abstraction factor, seems incredibly high, but after taking into account the timespan within which the precipitation falls, it is only 1.258 meters. The 0.05 gives a much lower value (Figure 64) of only 3.45

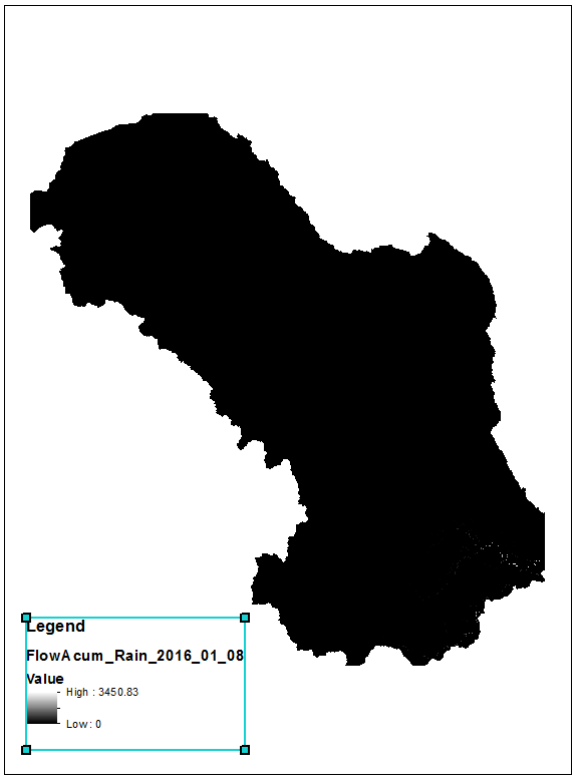


Figure 64 - Runoff from rain with abstraction value of 0.05

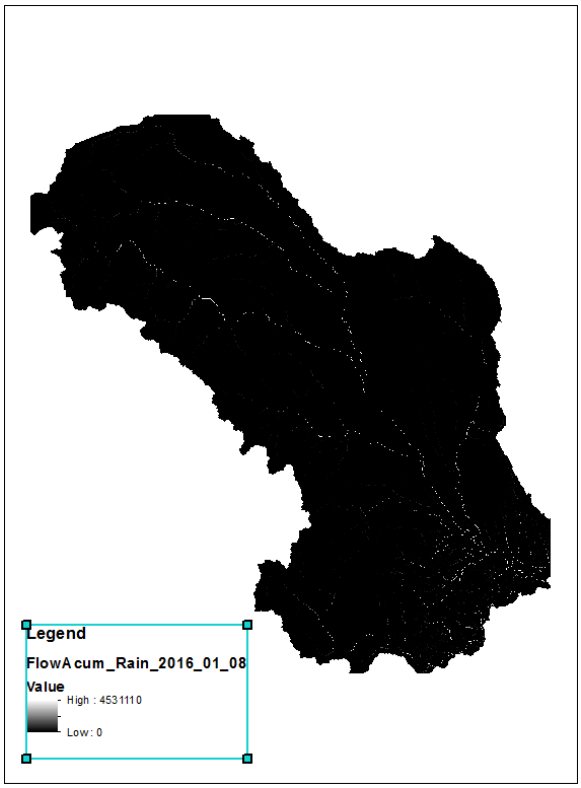


Figure 63 - Runoff from rain with abstraction value of 0.2

meters, which translates into only 0.95 millimeters. It is important to note the period of the precipitation, and put the average values in perspective. While 0.7mm per hour is one of our highest recorded values, during the flooding events of 2005 for example, there were rain events of up to 140mm/hour, and historically, up to more than 200mm/hour (ABA Siret(2) 2014). When putting these values into the CN equation we would get extremely high values. It is also important to note that these values do not take into account snowmelt, which is a big factor in the flooding events that occur in that area, snowmelt contributing to a high amount of the discharge, especially in the spring season.

We proceeded to add snowmelt to the flow accumulation and see what results it returns. As it can be seen in Figure 60, rain covers only the southern part of the basin. For the rest of the basin we had little snow, and also some temperatures over 0 C^{o}. This has led to snowmelt occurring in our basin. Moreover,

we had snow deposited in the areas where it rained, snow which also melted due to rain on snow melting processes.

When accounting for the snowmelt processes, the runoff values are higher (Figure 66 and Figure 65). Further on, when dividing these values to account for the format of our data, we obtain a value of 0.4 m in case of the 0.05 CN number and of 1.6 m for a 0.2 CN value. These values are the depth of the water at any second in time. Initially we wanted to compare these values with the discharge values at gauging stations, but due to the size of our raster cells, this comparison would not have been accurate. Instead, we chose to compare with average height values. Although the Siret River has a high discharge value of over 250 m^3/s at its confluence with the Danube River, its depth ranges between 0.2 m to 0.7 m, and can go up to 1 m in the spring. Values higher than these usually cause damage because of the flat nature of the river banks (ABA Siret(2) 2014). When comparing these average values with the results of the flow accumulation we clearly see that the abstraction value of 0.2 is overestimating the runoff.

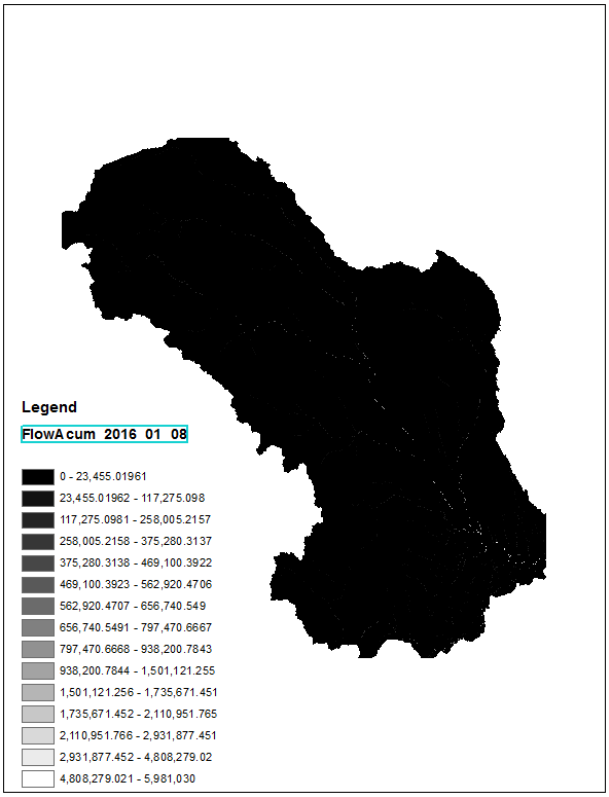


Figure 66 - Total runoff with abstraction value of 0.2

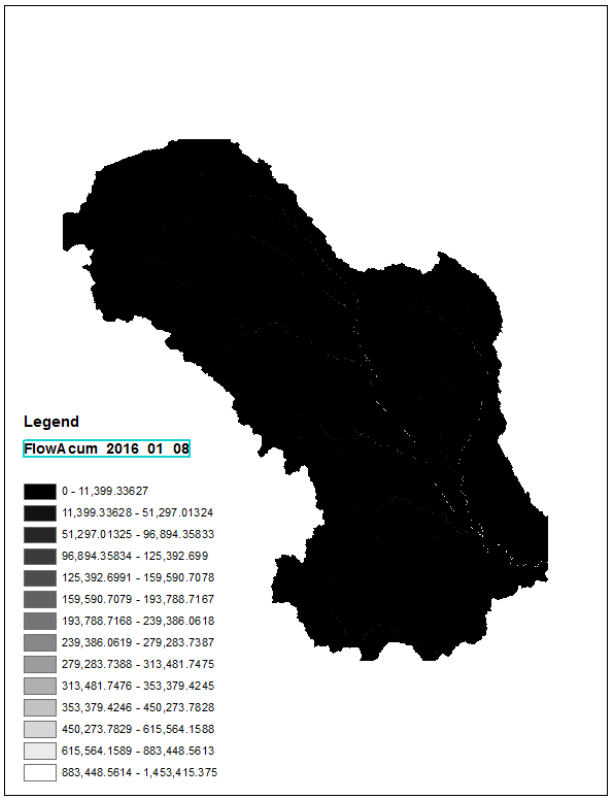


Figure 65 - Total runoff with abstraction value of 0.05

A water depth of 1.6 m, as calculated with the 0.2 abstraction coefficient would have most likely resulted in flooding in the lower basin area. First of all, flooding is highly uncommon to occur during the month of

January, and by checking the news we could not find any information of unusual high water levels during that period.

This has led us to the conclusion that the original abstraction factor used is unsuitable for our basin, and we decided to use the 0.05 factor. Of course this could be further improved, some literature suggesting that this factor could be calculated periodically (Baltas, et al. 2007).

4.5.2 Comparing results with other CN (Curve Number) values

Since several assumptions were made when assigning the CN values, we decided to test the quality of our assumptions by comparing the current runoff values with other values computed if the CN is larger by 10 and smaller by 10 respectively. This will allow us to, first of all, test if the CN values return good results, and secondly, help us understand the effect CN has on our runoff calculation. Figure 30 in Chapter 32 shows the assigned CN values used for the runoff calculation. We created two new CN files with values smaller by 10, and larger by 10. These raster files were used to calculate the new S values (Figure 67). As it can be seen from the figure, the change in CN resulted in significant changes in the S values.

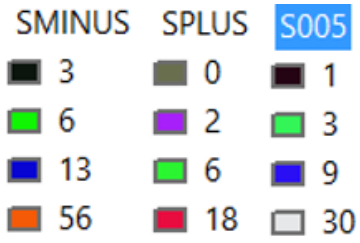


Figure 67 - Newly calculated S values

The next step is to calculate the runoff using these new S values. The same day of 8th of January was used, in order to be consistent. After calculating the new runoff values, we can see significant differences in the values obtained. When using the higher CN values we obtain the highest value of 0.66 mm (Figure 68), as opposed to the original value of 0.23 mm. This new value is almost 3 times larger than the original. By using a lower CN, we also have a significant difference, the highest value being 0.07 mm (Figure 69).

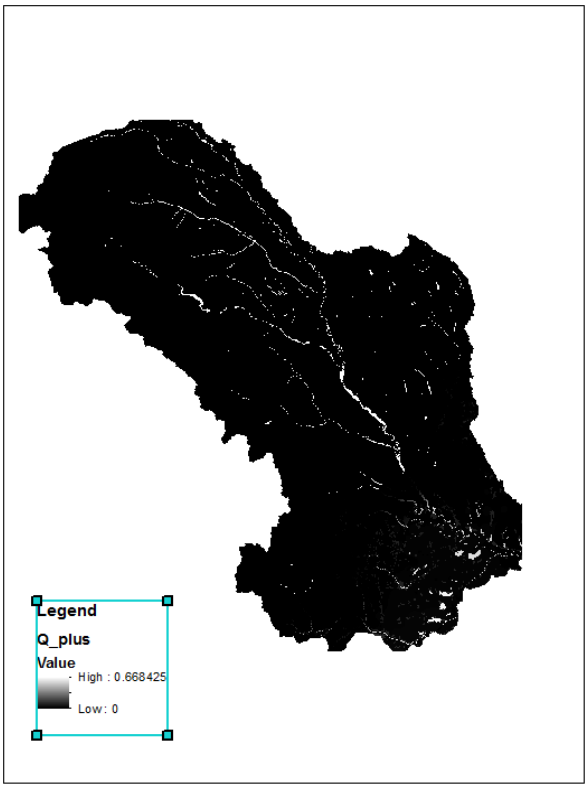


Figure 69 - Runoff from higher CN value

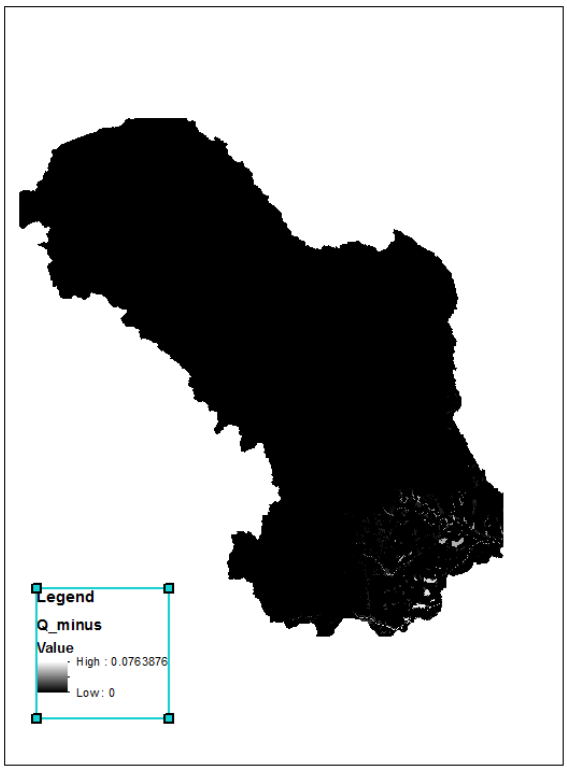


Figure 68 - Runoff from lower CN value

After obtaining the new runoff values, we calculate flow accumulation for both new scenarios in order to have an objective comparison. The newly created flow accumulation raster files (Figure 70 and Figure 71) have different values from the 3450 mm obtained with the original CN. In the case of the larger CN value, it is as high as 11456 mm and for the lower CN values is 783 mm. By again taking account of the time frame for our records, we obtain a height of 3.1 mm for the higher CN values and 0.2 mm for the lower CN.

As seen below, if the CN value is wrongly estimated, the results can have a significant effect on the end result. As seen in Appendixes 10 and 11, the TR-55 tables are constituted of a lot more classes of land cover, soils and soil quality. By using more detailed subdivisions of the land cover dataset the precision of the results can be improved. The implications of using a more detailed dataset is discussed in detail in Chapter 5.

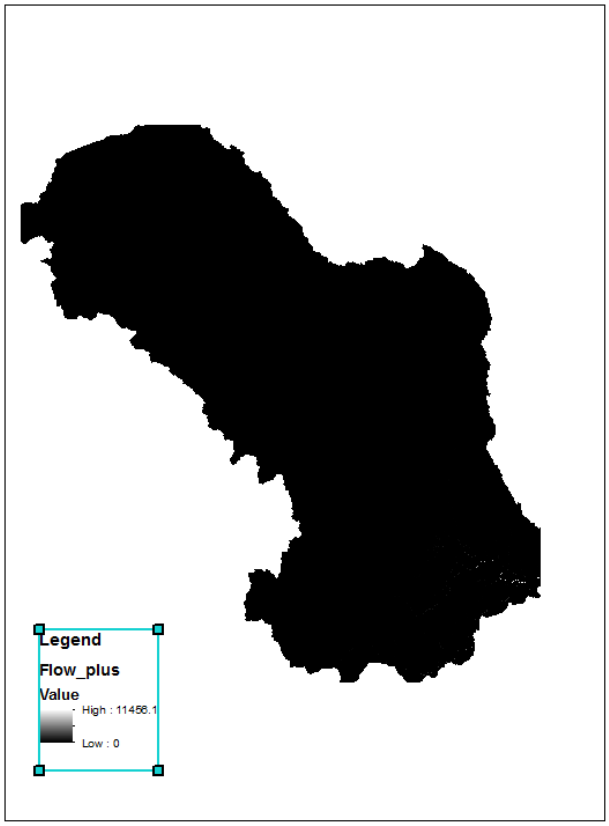


Figure 71 - Flow Accumulation with higher CN

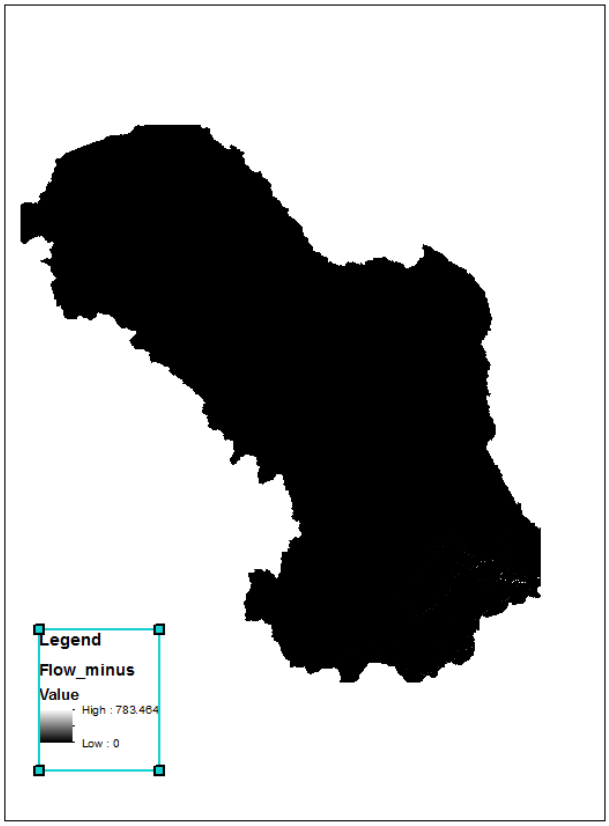


Figure 70 - Flow Accumulation with low CN

4.6 **Web Service and user interface**

One of the purpose of our project was to make our results available to the general public. There are many ways to disseminate information to the public, but we decided that the best way to do that would be to create a web map that displays the areas under threat of flood. The webpage would be a simple web map that can display the results of our analysis, all as layer tiles above a base-map.

For the creation of the web page we decided on using a combination of Leaflet, JQuery and JQuery Mobile. All of these are JavaScript libraries are popular and known to work together. Likewise, because of their popularity, there is an incredible amount of support available within the community.

For the base map we chose to use OSM (Open Street Map). Since this is an open-source, community driven map, the terms and conditions allow us an unlimited number of views as opposed to google maps that limits the number of views.

Since this is a website, the file responsible for this part was an .html file, as opposed to all our other scripts that were Python. Since this is an .html file that binds all our JavaScript libraries together, the file follows a classic html structure with tags and inside each tags the modules and parts responsible for the action required.

The first section of the document sets the type. By declaring it as an html type document (Figure 72), we ensure that the browser is aware of what file type it is reading and treats it accordingly. We also specify the language of the website. This ensures that all characters are treated accordingly to their language and get parsed as such. This allows to use language specific letters inside the code without the need of special strings. While the area covered is in Romania, and the language should be Romanian, this website is designed more as a demo for how we would like to disseminate the information to the public. Being a demo, and being reviewed by non-Romanian speaking users, we decided to create all the menus in English, which is also the language of the project.

```
1    <!DOCTYPE html>
2    <html lang="en">
```

Figure 72 - Document type and language

The next step is to create the <head> tags and place inside them the JavaScript libraries that we will be using later on. Because we want this website to be displayed properly on any device, right after declaring the title of the website, we add additional lines of code that tells the browser to disable scaling and display the website at its actual size. After this, we start importing the required libraries form CDN's (Content Delivery Network). Most libraries have links to the actual JavaScript library and also to the CSS (Cascading Style Sheets) file responsible for its styling and visual aspect.

```
<style type="text/css">
    #map { min-height: 320px; height: 100%;}
    body {
        padding: 0;
        margin: 0;
    }
    html, body, {
        height: 100%;
    }
</style>
```

Figure 73 – Map style CSS settings

After importing all the libraries another important step is the style of the map. Since we want it to cover the entire view, we set the height to 100% and set no paddings or margins(Figure 73).

```
    <div data-role="page"><div id="map" style="width:100%;height:100%;position:absolute;"></div>
```

Figure 74- Map div container

After creating all the style settings, and importing the libraries we proceed to creating the map. This is a 2 step process, and we start by creating a container (Figure 74). Here we mention the size of the map, and set its position to absolute. This, combined with the settings in the <head> section ensure that the map will be displayed full-screen on mobile devices and cover the entire extent of the browser on pc's and laptops. The second step in creating the map (Figure 75) is creating the load function that will ensure the map is loaded every time the website is open. We begin by creating variables that will hold the link to the tile layers. We also create a link to the OSM website that will be used when creating the map attributes in accordance to OSM terms and conditions.

We then create a new variable called map where we use the layer created in the previous line and set the view to a specific set of coordinates.

```
// Basemaps
var osmLink = '<a href="http://openstreetmap.org">OpenStreetMap</a>';
var osmUrl = 'http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
    osmAttrib = '&copy; ' + osmLink + ' Contributors';
var osmMap = L.tileLayer(osmUrl, {attribution: osmAttrib});
//Map settings
var map = L.map('map', {
    zoomControl: false,
    layers: [osmMap]
    })
    .setView([45.62, 27.42], 12);
//Baselayers name
var baseLayers = {
    "OSM Mapnik": osmMap
};
```

Figure 75 - Map load function

The view of the map could be modified in order to use the GPS location of the user using a very simple line of code to access a function within leaflet (Figure 76). Because the development is done in a different location then the Area of Interest, we decided on using coordinates, at least for the time being.

```
map.locate({setView: true, maxZoom: 16, watch: true});
```

Figure 76 - Potential location function

After loading the map, we create the buttons. The structure of the buttons will be simple, and keep in line with the overall simplicity of the webpage. We wanted to display the results of our script for the following seven days so we created a header item that will hold 7 buttons, one for each day we wanted to display data.

In order to create the buttons, we need to create a header element that will hold them (Figure 77). Since elements do not allow too much customization, we use CSS to further style them, by adding a transparent background to the header and centering them (Figure 78).

```
<div data-role="header" data-theme="a" div id="header" data-position="fixed" data-fullscreen="true" class="ui-content">
    <div data-role="controlgroup" div id="controlgroup" data-type="horizontal" >
```

Figure 77 - Webpage header that will contain the buttons

```
//Style for header and buttons
$("#header").css('background','rgba(0,0,0,0)');
$("#controlgroup").css('text-align', 'center');
```

Figure 78 - Header CSS style

With the header ready, we use JQuery's Mobile to create the buttons. Initially we wanted to create a dropdown button but JQuery's select class does not feature all the events we needed in order to display the WMS tiles, thus we proceeded on using traditional buttons. All 7 buttons have the same structure as the ones presented in Figure 79. We start by defining the data role, which is button, assign it an icon and the most important part, assign it an ID. This ID will be used later in in our display function.

```
<a href="#" data-role="button" data-icon="grid" id="WToday" data-inline="true"
   data-theme="a">Water Level Today</a>
<a href="#" data-role="button" data-icon="grid" id="WTomorrow" data-inline="true"
   data-theme="a">AWater Level Tomorrow/a>
```

Figure 79 - Button html code

All the settings applied so far result in the following website and button menu (Figure 80).

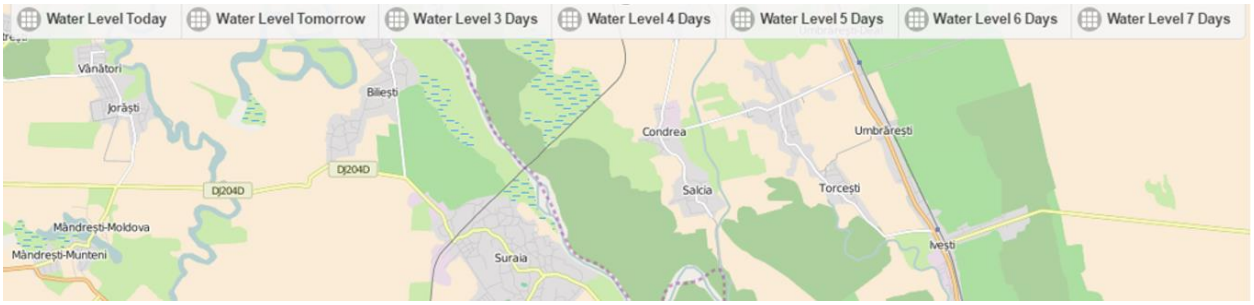


Figure 80 - Base map with buttons

After creating the button elements, we need to create the function to retrieve the maps from our server, and overlay them to the current base-map.

Unfortunately, we cannot simply overlay the results of our scripts on our new web map. We need to serve these raster files through a server. Further on, in order to provide zoom capabilities, the server needs to provide this raster in the form of tiles. There are many ways to achieve this, but we chose to do it using GeoServer. While GeoServer offers us the ability to create the map tiles and serve them to our website quite easily, this needs to be done manually. Likewise, inside GeoServer we needed to create a style .xml file. This would allow us to control how the tiles look, which is very important because we wish to make null values and values of 0 transparent.

We create a new variable named floodToday and inside that variable we use the leaflet command L.tileLayer.wms to make a call to our server (Figure 81). We provide the command with the address of our server and the port. The server being hosted locally, we can either us the IP value of 127.0.0.1 or we could call it localhost. Because localhost is such a vague variable, we encountered difficulties, therefore we had to use the numerical IP address to point to our map tile server. After giving the name of the required layer, we also mention the file type as well. This is important as the .png file type enables us to use transparency settings. We then tell the command what version the WMS server is in order to make sure the request is formatted properly and set the transparency to true.

```
var floodToday = L.tileLayer.wms("http://127.0.0.1:8080/geoserver/wms", {
layers: 'Floodsite:RACA_day0',
format:'image/png',
version: '1.1.1',
transparent: true
})
```

Figure 81 - Loading of the tile layer

This new layer need to be wrapped inside a new function that displays the layers on clicking the buttons created earlier. We create a function that triggers at a click event (Figure 82), and we also bind the function to the button by using the ID we assigned to each button earlier in the file. After inserting the code from Figure 81 inside to function we proceed to using the addLayer command available in leaflet.

```
$("#WToday").click(function(event) {
```

Figure 82 - Click event function

For each button and layer, we go through the exact same steps, making sure that we have the right layer from the tile server and the right button ID assigned to the function. After assigning a function to each button, the resulting map with the water levels for the current day is displayed below in (Figure 83).

We can see how the water follows the river on the base map and we can see as well how certain areas next to the river are flooded. These are small values under 10 mm in water height. We chose to display these here to show exactly the results of our model, but if this would be created for the general public, we could modify the style .xml file and ensure that all values under a certain threshold are transparent.

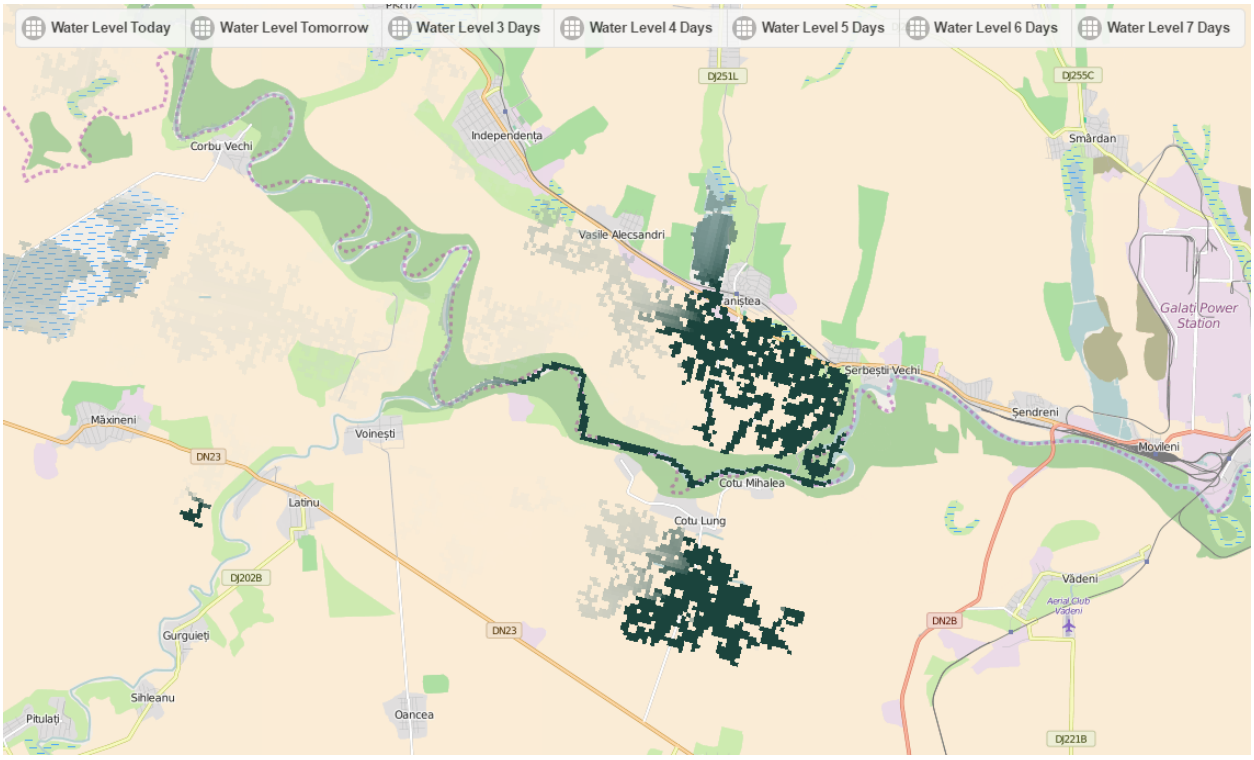


Figure 83 - Final map with the current day as an overlay

5 Discussion

The purpose of this section is to debate if the model could actually be used in real situations and if it could be able to make prediction that are, at least in theory, close to reality. The chapter also discussed the methods used, the uncertainties correlated to the methods and the different methods used, and what further improvements can be made.

5.1 Methods used, limitations and lack of timestamp

The idea behind project was the creation model that it should work with a limited amount of data and, more importantly, to use data that is free or easily available. Areas such as Siret Basin as well as a multitude of other basins in the Eastern Europe do not benefit from elaborated environmental datasets that can help in the creation of a flooding model, this being the reason why simple methods such us CN and Day Degree method were used. Both of them take only precipitation and temperature as input which make them perfect candidates for areas where data is scarce.

The downside of using such simple methods for representing complex natural processes leads to the necessity of a high degree of simplification, alongside with an assembly of processes accounting for the quality of the end results, which are not taken into consideration. The two methods account for temperature, but most importantly, precipitation, which represents the reality in the greatest measure, based on AHP hierarchy method. While having access to datasets that are not free and as easy to obtain would improve the quality of our results, the project would move away from the initial goal of using only freely available data.

Timestamp is a component that was not used in the present project due to unavailability of data, methods used and the general purpose of project. Timestamp has been used for projects of which purpose is an immediate or very precise result. In this case, a prediction of the areas that will be flooded in the next three to seven days is not considered to be an extremely precise estimation, but more a screening analysis where the temporal component can be excluded.

The last two methods, Flow Accumulation and RACA Method presented in Chapter 2 are used to screen the areas that could be in danger due to flooding. As previously mentioned in the Theory Chapter, Flow Accumulation results only shows the cell in the raster that receives the most water. This proved to be a

problem for the actual representation of the area in danger because, in theory, the water accumulates in one cell and would look as a “column of water”, which in reality is not possible, the water spreading around the adjacent landscape with lower altitude. Spreading the water from Flow Accumulation is possible using data which is not available for us, such as river depth in certain points (using on field work) or hydraulic coefficients such speed of water or also more advanced programing skills.

RACA (Rainfall with Cellular Automata) method was used by Espinola, et al. (2014) to work as a flow acccumulation, the result beeing the surfaces where the water accumulates. The theory sugests that this method alone could have been used just by considering the runoff value as input. The downside of this process represents the computation time, it being extremly large, therefore, for a limited processing power and a flooding prediction analysis this method alone is not suitable.

Additionaly, one more limitation which applies to both the RACA method and flow accumulation is the use of a flow dirrection raster file as input. As already presented in previous chapters, the flow direction tool in ArcGIS uses the D8 single direction flow algorithm (O'Callaghan and Mark 1984). Because the results of this tool are used in our RACA script and by the flow accumulation tool in ArcGIS, the limitations of the D8 method get transmited to the other tools. By using only one direction for flow, the results are not completely acurate. One of the reasons the RACA method was chosen was due to its compatiblility with the flow direction file. In reality, water would flow in all directions were the altitude is lower than that of the center cell, this resulting in the volume of water flowing to the neighbouring cells, which, alongside with the speed of water, would be dependent on a roughness coefficient such as Manning’s N or other roughness coefficients. While this situation could be problematic, it does not affect the model in a considerable manner. Due to the resolution of the DEM, 100 m, the difference in altitude between neighboring cells is significant enough to allow the flow of considerable quantities of water to move before this limitation of flow direction affects the results in a substantial way. In case of extreme flooding events, due to the fact that the already existing water cannot overflow and because huge amounts of water travel downstream through the river channel, the floodplains could be difficult to estimate. However,an advantage to the RACA method, is the simplicity of its rules, thus resulting in its easiness to program. There are no complicated equations requiring additional information, which as previously presented, is one of the difficulties when dealing with this region.

Despite the limitations, the chosen solution wasa mix of the two methods, Flow accumulation tool from ArcGIS beeing fairly capable to run in a short time, while the RACA method gives acceptable results for

small areas. This is the reason why we choose to run RACA only for small areas that in the past had suffered from flooding.

Originaly, we attempted to implement RACA method through ArcPy, but this proved to be an extremely slow process. If implemented through ArcPy, the workflow would have been to create a loop that goes through the raster on a cell by cell basis, using focal statistics to find the lowest cell and then proceeds with the calculations as presented in the Theory Chapter. As formely mentioned, this is an extremely slow process, taking around 0.5 seconds per cell. With only a 100 by 100 cell raster that would result in around 400 seconds, or, in other words, more than 6 minutes per iteration. Considering that our AOI raster has 574 columns and 374 rows, the processing time would be around 3577 minutes per iteration. This is highly problematic because we wanted to run the script for several hundred of iterations each day, and we had data for 7 days in advance.

This is the moment when the Numpy proved to be higly useful. By converting the raster into a Numpy array, we were able to process everything in a much faster way. With the help of this module, one single iteration takes less that 1 second , which allows us to run the script in advance for up to 7 days.

Despite the lack of proper optimization, and because of the simplicity of the model and the scarce datasets used as input, our model has wide applicability. One one hand, in most West European and North American countries, data availability is not an issue, citizens having to ability to access huge amounts of data. On the other hand, East European contries do not have that luxury, the data available being extremely scarce and difficult to obtain. This problem is not confined to East European countries, African and Southeastern Asian countries having scarce data availability and also being highly vulnerable to flooding events (Douglas, et al. 2016).

If long term monitoring proves that the results of our model are usable for flood prediction, by adapting some coefficients inside each formula to account for the area's characteritics, this model could be adapted and applied for different areas around the globe.

5.2 Uncertainties, errors and parameters

When discussing uncertainties in environmental modeling and primarily in hydrological modeling, different individuals have different opinions on how the uncertainty should be measured or even what uncertainty mean. This is happening due to the high number of variables that should be included in a

perfect model, as well as the subjective views of each modeler. Particularly, in hydrology modeling the water movement and runoff might differ from hydrologist to hydrologist. Keith Beven, known as one of the greatest hydrologist of our time, published the result of his PhD thesis in 2001, almost 35 years after receiving his diploma due to poor results. Based on Box and Draper's (1987) statement that "all models are wrong, but some might be useful" (p. 424) Beven (2009) recommended that each modeler should look at the model with a pragmatic realist philosophy (Beven 2009).

Bearing in mind this philosophy, the uncertainties that can occur can be categorized in known uncertainties and unknown uncertainties. As it can be observed even from the name, the unknown uncertainties are yet to be revealed by becoming known uncertainties during the process of calibration, validation and even during the eventual functioning life of the model. On the opposite side, the known uncertainties can and have to be discussed, considering that for each step of the modeling process, the possible errors that can interfere are discussed together with the fashion in which that can influence the final result.

5.2.1 **Constant data errors**

The model works with several datasets that are assumed to be constant, meaning that the data does not change its structure during the modeling process. On one hand, this could be valid for certain periods, but giving the fact that earth surface is continually changing, an older version of the input data may have a negative impact on the end result. One the other hand, the soils databases can be considered as the data that changes the least over time. Even though the dataset is provided by the European Environment Agency, the maps are based on old soil maps developed more than 50 years ago in Romania.

Digital Elevation data is created in Europe regularly and the quality is continuously improving. A good practice would be updating the DEM data that the model is based on to the last available dataset. Depending of the model requirements, a higher resolution dataset would improve the end results, due to more accurate representation of the surface. Processing a higher resolution DEM implies a larger amount of time used for processing and better equipment, where the modeler has to choose the best suitable solution based on the requirements. In the present project, the DEM has a resolution of 100 m even though the original dataset had a 25 m resolution. This was chosen due to limitations regarding the processing power available on the personal computers. In theory, a higher DEM accuracy would improve the end result.

Land cover is the element that presents the highest degree of transformation. The lack of governance over the environmental issues in Romania can vastly transform the surrounding environment and especially the land cover. Deforestation is a common practice in remote areas within the country, while having a rapidly changing landscape. As presented in the theory chapter, the CN Method makes use of soil type and, most importantly of land cover, where, the surfaces that after a rain event would prevent runoff creation are exactly the afforested areas. Therefore, changing the state of the environment without updating the Land cover data results in the highest errors in the end results as seen in the Chapter 4.5.1.

5.2.2 Variable data errors

The model works with two variable data that are retrieved daily from a web site specialized in providing meteorological data. As presented in the Data chapter, temperature is represented by the mean hourly value for 24 hours. There are several discussions on how exactly the mean temperature should be considered for the Day Degree method, the most commonly used being the mean temperature as registered hourly for one day.

The quality of the data retrieved "varies by location and by how far in the future we're forecasting for, but we're broadly comparable to other weather providers" (Jay LaPorte personal communication March 2016). The data has as a source several well-known data providers from all over the world and the accuracy is comparable with the data that a modeler can obtain free or buy without having access to private or specialized equipment. A more precise data would be available from the actual measuring stations available in the area eliminating the possible unknown error from the online meteorological data providers.

5.2.3 Long time calibration

As presented above, in environmental modeling, uncertainty estimation and validation of a model based on the results can become almost an impossible task. A way for a model to improve its capacity of delivering results that represents the reality in a higher measure is the long time calibration also called post calibration.

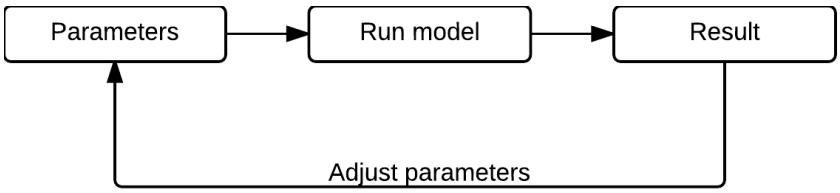


Figure 84- Long time calibration steps based on modeling steps described by Beven (2014)

The result of the model must be compared with the real world observations and it must be based on the level of goodness, while the parameters should be adjusted until a better measure of fit results (Beven 2014), as seen above in Figure 84.

The Day Degree method has two components that can be calibrated over time. The first component, the way that the mean temperature is calculated might improve the results. This would imply changing the mean temperature used in the method from mean value calculated from hourly mean temperature reading into calculating a mean between the highest temperature and the reference temperature as Rango and Martinec (1995) presents. Moreover, another way would be to increase the reference temperature from 0 C^o to 4.4 C^o and use the maximum daily temperature as a mean, as presented in Engineer Manual from Army Corps of Engineers. Having three different studies saying that the suggested way for determining the mean temperature used in the method is recommended, we believe that only through post calibration and multiple adjustment the best suitable method can be found.

The second and most important component that can be calibrated over long periods of time is the Day Degree factor. The current model uses a factor that increases during the snow melting season and the value being based on studies presented in the Chapter 2. Besides the calibration, if possible, the values should be improved by having access to field equipment and including more factors into the development of the Day Degree factor. Rango and Martinec (1995) state that adding the solar radiation in the Day Degree equation can result in higher accuracy, this however not being possible in the present model due to lack of data and involvement from the Romanian authority in charge with the Siret basin.

The CN method uses the abstraction coefficient which was heavily discussed in several studies during the years and presented in detail in Chapter 2. Over time, the coefficient should be observed and its value should be corrected upon observing the quality of the results. Another way the CN method could improve in accuracy is by using more detailed CN values. This can be realized by using the subdivisions of land cover data provided by EEA and for each subdivision, a more accurate CN value can be attributed. This is a

practice that can be associated with any type of data being no limit on how much the quality of data can be improved. Likewise, the balance between the quality of data and the processing resources are factors that need to be considered. Even if Romania would have the benefit from extremely precise DEM data as Denmark for example, there is an important decision that the modelers have to take on how precise the data used for the model should be. An extremely precise dataset would be able give better results, but may not be suitable in terms of processing time and resources.

5.3 Future improvements

As previously mentioned, the process of calibration for some equations takes time in order to give results that are as accurate as possible, while also applying to the methods chosen, and the implementation of the specified methods.

One method that can be improved is the RACA method used for obtaining the area affected by flooding, the limitations of this method being already mentioned in the previous chapter. These limitations stream both from the structure of the method, but also from the input flow direction file used.

The first potential improvement to the RACA method would be the use of the actual elevation data from the DEM when determining where the water flows. By implementing this feature inside our script, we would have no need to use the flow direction file as input. However, if the feature would be implemented, then it would be required to modify the equation as well in order to determine the amount of water that flows into each cell. There are several methods available which, minimal effort and programing knowledge, could be adapted to our model, in order to improve the results (Cai, et al. 2014).

Additionally, one more improvement that could be made to this project would involve the migration from the Arc environment to using GDAL. GDAL stands for Geospatial Data Abstraction Library, and it is an open source library capable of doing almost everything ArcPy has the abilities to perform. Over the years, GDAL has developed and matured into a stable platform, used by many GIS professionals worldwide. If we were able to utilize GDAL instead of ArcPy, we could have delivered all the components of this project as an open source, making it replicable for almost any site around the world. GDAL has been seen as holding excellent documentation, combined with the trustworthiness of a highly capable and supportive community. The reason why ArcPy was chosen in the beginning is our familiarity with the Arc environment and its availability to us. Unfortunately, ArcGIS is a proprietary software, and it might not be available for

everyone. Moreover, an important issue with ArcPy is the lack of control when it comes to how the tools work, ArcGIS being called a “black box” in terms of functionality and user understanding of said functionality.

Being open source, GDAL offers the user the possibility to customize each specific tool to his purpose if his programing skills are good enough to achieve this. By using GDAL, we could also eliminate the need to use GeoServer, and, as a result, fully automatize the workflow. As mentioned in Chapter 4.6, while GeoServer is simple to use, it does not provide the ability to automatize its function, but replacing it with GDAL would enable us to generate the tiles automatically, while also giving us more control over the process.

Improvements could also be made to the script’s structure as well. Due to our decision to utilize ArcPy, the structure was created to accommodate the tools available within it, meaning that many loops were used inside the scripts to obtain the desired results. While this works, with the scripts calling ArcPy functions, the limitations of this structure become apparent during the RACA script which takes a significant amount of time. While looping through files and tools does not create any performance issues, looping through a raster on a cell by cell basis becomes a tedious endeavor. And while the RACA script would benefit the most from a change in structure, all other scripts would see improvements in performance and speed. The raster looping issue could be improved by using a process called vectorization. By vectorising the process that retrieves each cell and its neighbor, significant improvements could be made to the performance of the scripts.

Additionally, there are improvements that could be made to the quality of the input data, more specifically the interpolated data. As it was presented in Chapter 2.5.3, Kriging gives slightly better results although having a higher processing time. An advantage of Kriging is the fact that it allows the fitting of a function to generate a better interpolation. Over time, with proper calibration, we could improve the quality of the interpolation and this would justify the longer processing time. Moreover, one more fashion to improve the interpolation quality would be to obtain better meteorological data, while having a higher density of points. We were limited to a small number of points, but if this model would be implemented with support from a commercial or administrative entity, a higher number of points would be available for a surcharge.

One of the aims of this project was to present the results to the general public, thus we achieved this by creating a website which is able to present the results of our model. While the general public can see the affected area of a potential flood, it would require them to check our website regularly. This can be

avoided by creating an alarm system inside our site and by creating a database of users, they would be able to register and provide their address. This address would be stored inside our database, and the coordinates of the address would be compared to the water level raster for each day, and in the case of an overlap, the user could be notified via email or text message. This would ensure that end users can be notified even if they do not check our website regularly.

6 Conclusion

The goal of this project was to create a flooding model that can provide information regarding the area where a flooding event might occur in the coming 3-7 days. This proved to be a more difficult task than initially expected, due to the limitations that we had in term of data availability and the different approaches we had to consider. The first major challenge was to find the methods that work with the scarce data available for our region and try to apply them in order to obtain suitable results. The second challenge was to combine all these methods and create a system that can run by itself and be able to make the result accessible to the general public.

By putting together various individual methods, describing the processes that have a big impact on flood events and finding a way to interconnect them and work together, we believe that we managed to fulfill the goal. Moreover, being able to access free meteorological data in real time and use it as an input for the project was another goal that we wanted to achieve. By using an API provided by Forecast.IO, we were able to access up to date meteorological data at any given time and update our predictions in real time.

The last two goals are just partially achieved. Even though theoretically the methods chosen should work well together, without long term monitoring of the area and the results of the model we cannot be certain with regard to the accuracy of our results. Additionally, because of our lack of experience in programing, we were not able to fully optimize the codes in order to obtain fast results. All of this is discussed more in depth in Chapter 5.

The last goal was achieved in a very basic sense of having some results available to the general public. The example presented Chapter 4.6 is just an idea on how the results would look. A database with users and more functionality is missing, as presented in Chapter 5.

Although our project has some shortcomings, as presented earlier, if the results of our model withstand the scrutiny of long term monitoring and validation, this system could be extremely useful in predicting flooding events and informing the population in a timely manner. Also, if long term monitoring confirms the validity of the model, because of the scarce datasets used, it can be adapted and applied fairly easy to any part of the world.

7 Bibliography

University of Maryland. 2014. *Global Land cover Facility.* Accessed February 04, 2046. http://glcf.umd.edu/data/.

ABA Siret(1). 2014. *Siret Management Plan.* Management Plan, Bucharest: RoWater. Accessed January 24, 2016. http://www.rowater.ro/dasiret/Planul%20de%20Management%20al%20Bazinului%20Hidrografic%20Sire/Planul%20de%20management%20al%20spatiului%20hidrografic%20Siret/Plan%20Management%20SH%20Siret%20-%20vol.%20I.pdf.

ABA Siret(2). 2014. *Planul pentru Prevenirea, Protecția și Diminuarea Efectelor Inundațiilor (PPPDEI) în bazinul hidrografic Siret.* Prevention Plan, Bacau: ABA Siret.

Annika, Badorreck, H Gerke Horst, and F Huttl Reinhard. 2013. "Morphology of physical soil crusts and infiltration patterns in an artificial catchment." *Soil & Tillage Research* 1-8.

Arun, P.,V. 2013. "A comparative analysis of different DEM interpolation methods." *The Egyptian Journal of Remote Sensing and Space Science* 133–139.

Bakir, Mohamad, and Zhang Xingnan. 2008. "GIS-Based hydrological modeling: A comparative study of HEC-HMS and the Xinanjian Model." *IWTC12.* Alexandria.

Baltas, E., A., N., A. Dervos, and M., A. Mimikou. 2007. "Technical Note: Determination of the SCS initial abstraction ratio in an experimental watershed in Greece." *Hydrology and Earth System Sciences* 1825-1829.

Barredo, José I. 2007. "Major flood disasters in Europe: 1950–2005." *Natural Hazards* 125-148.

Berezowsky, Tomasz, Jaroslaw Chormanski, and Okke Batelaan. 2015. "Skill of remote sensing snow products for distributed runoff prediction." *Journal of Hydrology* 718-719.

Beven, Keith. 2009. *Environmental Modelling: An uncertain future?* Taylor & Francis.

—. 2014. *Rainfall-Runoff Modeling The Primer Second Edition.* Lancester: Wiley-Blackwell.

Box, G.E P, and N R Draper. 1987. *Empirical Model Building and Response Surfaces.* New York: John Wiley & Sons.

Bratko, Ivan. 2009. *Qualitative Modelling.* University of Ljubljana, p8, p10-13.

Brimicombe, Allan. 2010. *GIS, Modeling and Engeneering.* Second Edition. CRC Press.

Britannica, Enciclopedia. 2016. *Loess .* Accessed 5 24, 2016. http://global.britannica.com/science/loess.

Budui, Vasile. 2002. "Precipitations Generating Factors in the Central Moldavian Tableland between Stavnic and Siret." *Analele Universităţii „ŞTEFAN CEL MARE" Suceava.*

Burrough, P A. 1986. *Principles of geographical information systems for land resources assessment.* Oxford: Claredon Press.

Cai, Xin, Yi Li, Xing-wen Guo, and Wei Wu. 2014. "Mathematical model for flood routing based on cellular automaton." *Water Science and Engineering* 133–142.

Caruso, C., and F. Quarta. 1998. "Interpolation Methods Comparison." *Computers & Mathematics with Applications* 109-126.

Christie, Mike, Andrew Cliffe, Philip Dawid, and S., Stephen Senn. 2011. *Simplicity, Complexity and Modelling.* Wiley Press.

Documentation, PSF. 2016. *JSON encoder and decoder.* May 07. Accessed May 09, 2016. https://docs.python.org/2/library/json.html.

Douglas, Ian, Kurshid Alam, Marianne Maghenda, Yasmin McDonnell, Louise McLean, and Jack Campbel. 2016. "Unjust waters: climate change, flooding and the urban poor in Africa." *Environment & Urbanization* 187- 205.

Dueker, K J, and D Kjerne. 1989. *Multi-purpose cadastre: Terms and definitions.* American Society of Photogrammetry & Remote Sensing and.

EEA. 1995. *CORINE Land Cover.* Report, EEA.

—. 1995. "CORINE Land Cover." *European Enviroment Agency.* January 01. Accessed March 14, 2016. http://www.eea.europa.eu/publications/COR0-landcover.

—. 2014. *Corine Land Cover 2006 raster data.* April 8. Accessed March 14, 2016. http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2006-raster-3#tab-metadata.

—. 2015. *Digital Elevation Model over Europe (EU-DEM).* June 25. Accessed November 14, 2015. http://www.eea.europa.eu/data-and-maps/data/eu-dem/#tab-metadata.

Engineers, Corps of. 1956. "Snow Hidrology." Sumary report, Oregon.

Engineers, U.S.Army Corps of. 1998. "Engineering and design- Runoff from Snowmelt." *Engineer Manual*, March 31: 45.

ESDAC. 2015. *European Soil Database V2 Raster.* http://esdac.jrc.ec.europa.eu/content/european-soil-database-v2-raster-library-1kmx1km.

Espinola, Moises, Jose Piedra-Fernandez Antonio, Rosa Ayala, Luis Iribarne, and Saturdino Leguizamon. 2014. "Modeling Rainfall Features Dynamics in a DEM SAtellite Image with Cellular Automata." *Lecture Notes in Computer Science* pp 238-247.

ESRI. 2013. *ArcGis Help.* Accessed 04 02, 2016. http://help.arcgis.com/EN/ArcGISDesktop/10.0/Help/index.html#/What_is_raster_data/009t00000002000000/.

—. 2016. *Comparing interpolation methods.* Accessed February 16, 2016. http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/comparing-interpolation-methods.htm.

—. 2016. *GIS DIctionary.* Accessed JAnuary 24, 2016. http://support.esri.com/en/knowledgebase/Gisdictionary/browse.

—. 2016. *How IDW Works.* Accessed March 20, 2016. source http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/how-idw-works.htm.

—. 2002. "Metadata and GIS- An ESRI white paper." *www.esri.com.* October. http://www.esri.com/library/whitepapers/pdfs/metadata-and-gis.pdf.

—. 2015. *What is raster data?* Accessed February 1, 2016. http://desktop.arcgis.com/en/desktop/latest/manage-data/raster-and-images/what-is-raster-data.htm.

European Environment Agency. 2011. *http://www.eea.europa.eu/.* October 7. Accessed January 20, 2016. http://www.eea.europa.eu/highlights/natural-hazards-and-technological-accidents.

Florinsky, Igor V. 2011. *Digital Terain Analasys in Soil Science and Geology.* Pushchino: Elsevier.

ForecastIO. 2015. *Forecast IO.* January 29. Accessed January 29, 2016. https://developer.forecast.io/docs/v2.

Garen, C David, and S Daniel Moore. 2005. "Curve number hydrology in water quality modeling: uses, abuses and future directions." *Journal of american water resources association* 377-388.

Harkamp, Dewi A, Alfred Stein, Kirsten De Beurs, and Jeffrey W White. 1999. "Interpolation Techniques for Climate Variables." *NRG-GIS Series 99-01.*

Hawkins, Richard H, Jiang Ruiyun, Donald E Woodward, Allan T Jr Hjelmfelt, and J E VanMullem. 2002. "Runoff curve number method: examination of the initial abstraction ratio." *Proceedings of the Seond Federal Interagency Hydrologic Modeling Conferance.* Las Vegas, Nevada. 1-6.

Hock, Regine. 2003. "Temperature index model melt modelling in mountain areas." *Jurnal of Hydrology* 104-115.

I.N.M.H. 1992. *Atlas of the Waters Survey Romania.* Bucharest: I.N.M.H.

INSPIRE. 2015. *About Inspire.* 11 15. http://inspire.ec.europa.eu/index.cfm/pageid/48.

Jakeman, A J, A A Voinov, A E Rizzoli, and S H Chen. 2008. *ENVIRONMENTAL MODELLING,SOFTWARE AND DECISION SUPPORT.* Elsevier.

Jakeman, A J, R A Letcher, and J P Norton. 2006. "Ten iterative steps in development and evaluation of environmental models." 604.

Jenson, S.,K., and J., O. Domingue. 1988. "Extracting topographic structure from digital elevation data for geographic information system analysis." *Photogrammetric Engineering & Remote Sensing* 1593–1600.

Jonathan, Kult, Choi Woonsup, and Choi Jinmu. 2014. "Sensitivity of the Snowmelt Runoff Model to snow covered area and." *Applied Geography 55* 30-38.

Judah, Cohen, Hengchun Ye, and Jones Justin. 2015. "Trends and variability in rain on snow events." *Geophysical Research letters* 2-4.

Kyoung, Jae Lim, A Engel Bernard, Muthkrishnan Suresh, and Harbor Jon. 2006. "Effects of initial abstraction and urbanization on estimated runoff using CN technology." *Journal of the american water resources association* 629-638.

Langebein, W B; Kathleeden, I Iseri. 1973. "General Introduction and Hydrologic Definitions." *USGS.* Accessed January 29, 2015. http://water.usgs.gov/wsc/glossary.html#Runoff.

Leopold, L B; Wolman, M G; Miller, J P. 1964. *Fluvial Processes in Geomorphology.* San Francisco: W H Freeman.

Linsley , Ray K Jr. 1958. "Hydrology for Engineers." 340.

Linsley, Ray K Jr. 1943. "A simple procedure for the day to day forecasting of runoff from snowmelt."

Liu, Yan. 2009. *Modelling Urban Development with Geographical Information Systems and Cellular Automata.* New York: Taylor & Francis Group.

Llorens, Pilar. 1997. "Rainfall interception by a Pinus sylvestris forest patch overgrown in a Mediterranean mountainous abandoned area." *Journal of Hydrology* 331-333.

Longley, A., Paul,, J., Mike, Goodchild, J., David, Maguire, and W., David Rhind. 2010. *Geographic Information Systems and Science.* Hohn WIley & Sons.

—. 2005. *Geographical Information Systems: Principles, Techniques, Management and Applications (Abridged Edition).* New York: Wiley.

Loster, Thomas. 1999. "Flood Trends and Global Change." *EuroConference on Global Change and Chatastrophe Risk Management.* Laxenburg: IIASA.

Luo, W., M., C. Taylor, and S., R. Parker. 2007. "A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from England and Wales." *International Journal of Climatology* 947–959.

Manoj, Gundalia, and Dholakia Mrugen. 2014. "Impact of Monthly Curve Number on Daily Runoff Estimation for Ozat Catchment in India." *Open Journal of Modern Hydrology* 144-155.

McCabe, Gregory J, Martyn P Clark, and Lauren E Hay. 2006. "Rain-on-snow evemts in the Western United States." *American Metheorologycal society* 320.

Merz, R, and G Bloschl. 2003. "A process typology of regional floods." *Water resources research* 1-20.

Miller, J., Harvey. 2004. "Tobler's First Law and Spatial Analysis." *Annals of the Association of American Geographers* 284–289.

Ministerul Mediului si Dezvoltarii Durabile. 2002. *Patrimoniul si Cadastrul apelor.* Accessed January 24, 2016. http://cadastru.8k.com/catalog.html.

Mitas, L., and H. Mitasova. 1999. "Spatial interpolation." In *Geographical Information Systems: Principles, Techniques, Management and Applications*, by A., Paul,, J., Mike, Goodchild, J., David, Maguire, and W., David Rhind Longley, 481-492. Wiley.

Mitchell, James K. 2003. "European River Floods in a Changing World." *Risk Analysis* 567–574.

Moustafa, T Chahine. 1992. "The hydrological cycle and its influence on climate." *Nature* 373-374.

Munich Re Group. 2005. *Annual review: Natural catastrophes 2004.* Anuall review, Munich: Munich Re Group.

Munteanu, Marian, and Mihai Tatu. 2003. "The East-Carpathian Crystalline-Mesozoic Zone (Romania): Paleozoic Amalgamation of Gondwana- and East European Craton-derived Terranes." *Gondwana Research* 185-196.

Naoum, S., and I., K. Tsanis. 2004. "Ranking Spatial Interpolation Techniques using a GIS-based DSS." *GlobalNEST International Journal* 1-20.

NRCS. 1997. "Hydrologic Soil Groups." *National Engineering Handbook Hydrology*, September: Chapter 7.

O'Callaghan, F., John, and M., David Mark. 1984. "The Extraction of Drainage Networks from Digital Elevation Data ." *Computer vision, graphics and image processing* 323-344.

Putkonen, J., and G. Roe. 2003. "Rain-on-snow events impact soil temperatures and affect ungulate survival." *Geophysical Research Letter.*

Rango, A, and J Martinec. 1995. "Revisiting the degree-day method for snowmelt computations." *JAWRA Journal of the American Water Resources Association* (4): 657-669.

Ravetz, Jerome R. 1977. "Integrated Environmental Assessment Forum:developing guidelines for "good practice"." ULYSSES Working Paper.

Romanescu, Gheorghe, and Cristian Stoleriu. 2013. "Causes and effects of the catastrophic flooding on the Siret River (Romania) in July–August 2008." *Natural Hazards* 1351-1367.

Romanescu, Gheorghe, and Ioan Nistor. 2011. "The effects of the July 2005 catastrophic inundations in the Siret River's Lower Watershed, Romania." *Natural Hazards* 345-368.

Rotmans, Jan, Mike Hulme, and E., Thomas Downing. 1994. "Climate change implications for Europe: An application of the ESCAPE model." *Global Environmental Change* 97-124.

Saaty, R T. 1987. "THE ANALYTIC HIERARCHY PROCESS-WHAT IT IS AND HOW IT IS USED." *Mathl Modelling* 161-164.

Sanderson, Robert. 2010. "Introduction to remote sensing." In *Introduction to remote sensing*, by Sanderson Robert, 1-5. New Mexico.

Solomon, Harddema. 2005. *A GIS based Surface Runoff Modelling and Analysis of Contributing Factors.* Phd Thesis, Enschede: ITC.

Tae-Woong, Kim, Ahn Jae-Hyun, Moon Geon-Woo, and Ajmal Muhamad. 2014. "Investigation of SCS-CN and its inspired modified models for runoff estimation in South Korean watersheads." *Journal of Hydro-enviroment research* 593.

Talaba, Ion, Lucian Flaiser, and L., Daniela Covalinschi. 2005. *Euroregiunea Siret-Prut-Nistru.* Iasi: Editura Performantica.

Tarboron, G., David. 1997. "A new method for the determination of flow directions and upslope areas in grid digital elevation models." *WATER RESOURCES RESEARCH* 309-319.

Tobler, R., Waldo, and Susan Kennedy. 1985. "Smooth Multidimensional Interpolation." *Geographical Analysis* 251–257.

Tobler, W., R. 1970. "A Computer Movie Simulating Urban Growth in the Detroit Region." *Economic Geography* 234-240.

Torrens, Paul M. 2000. "How cellular models of urban systems work (1. Theory)." *Working Paper.* London: Copyright CASA, UCL. ISSN: 1467-1298, November. 22-23.

USDA. 1999. "SCS Runoff Ecuation." Engineering Training Series.

USDA. 1986. *Urban Hydrology for small Watersheds, TR-55.* Technical Release, Unites States Deparment of agriculture.

Vieux, E., Baxter. 2013. *Distributed Hydrologic Modeling Using GIS.* Springer Science & Business Media.

Weiss, L, and W T Wilson. 1958. "Snow-melt degree-day ratios determined." *Transactions of the American Geophysical 39* 680-670.

Werritty, Alan, Andrew Black, Rob Duck, Bill Finlinson, Neil Thurston, and Simon Shackley. 2002. *Climate change: Flooding accurances review.* Review, Dundee: Scottish Executive Central Research Unit, 1-2.

Wharen, Tavares Filipa, Stefan Julich, Pedro Nunes Joao, Gonzales Pelayo Oscar, Hawtree Daniel, Heinz Feger karl, and Jacob Keizer Jan. 2015. "Combining digital soil mapping and hydrological modeling in a data scarce watershed in north-central Portugal." *Geoderma*, August 14: 350-354.

Zhu, A-Xing. 2002. "SoLIM: A New Technology For Soil Mapping." Overview.

Zotarelli, Lincoln. 2010. "Step by Step Calculation of the Penman-Monteith." *Agricultural and Biological Engineering Department, UF/IFAS Extension*, February: 1-9.

APPENDIXES