Recharge flooding of collieries in South Africa

by

Eelco Lukas

Submitted in fulfilment of the requirements in respect of the Doctoral degree qualification

Doctor of Philosophy (Geohydrology)

at the
Institute for Groundwater Studies
Faculty of Natural and Agricultural Sciences
University of the Free State
Bloemfontein

Promotor: Prof PD Vermeulen

BLOEMFONTEIN

August 2020

DECLARATION

I, Eelco Lukas, declare that the thesis hereby submitted by me for the Doctor of Philosophy degree at the University of the Free State is my own independent work and has not previously been submitted by me at another university/faculty. I furthermore cede copyright of the thesis in favour of the University of the Free State.

…………………………………………………………….

Eelco Lukas

(2011041890)

Acknowledgements.

The personnel of the IGS for their support.

My promotor, Prof Danie Vermeulen, for his guidance, advice and enthusiasm towards this work.

The late Prof Frank Hodgson under who’s guidance WISH and WACCMAN have started their existence.

The late Prof Gerrit van Tonder from who I have learned more about groundwater than from any body else.

My wife, Annekie, for her motivation when needed the most.

My children, Anton en Anmare for their patience with a busy dad.

Contents

List of Figures.

List of Tables.

List of Abbreviations

C	C programming language
C++	C-plus-plus programming language
CAD	Computer-Aided Design
CDC	Centres for Disease Control and Prevention
GIS	Geographic Information System
IGS	Institute for Groundwater Studies
KPa	Kilo Pascal – Unit of pressure (1 Pa is 1 N/m^2)
M m^3	Mega (million) cubic metres (10^6 m^3)
mamsl	Metres above mean sea level
mbgl	Metres below ground level
Ml	Mega (million) litres (10^3 m^3)
MS-DOS	Microsoft - Disk Operating System
PC	Personal Computer
TIN	Triangular Irregular Network
WACCMAN	Water ACCounting and MANagement
WCP	Water Control Point
Windows NT	Windows New Technology
WISH	Windows Interpretations System for the Hydrogeologist

1 Introduction

1.1 Preface

Coal mining has been documented in South Africa as far back as 1838 and undoubtedly contributed significantly toward the economy as well as job creation (Barnes and Vermeulen 2012).

Many mines in South Africa have changed the geology on a localised level by taking out the coal seams. The change in geology also changed the geohydrology. The voids that are left to fill with water. Most of that water will be groundwater, but some will be surface water and stormwater. In the geohydrology, we talk about the recharge of a mine. The recharge is depending on several factors, some depending on the physical mining methods and others depending on the geological setting.

Recharge and mine flooding (opencast or underground) is nothing new; it occurs in every mine from the first day the mine was created. Recharge is a natural process; inundation is the act of intentionally flooding of land that would otherwise remain dry. Where inundation floods a mine using a single or multiple pumping locations, the recharge is a phenomenon that occurs over the full extent of the mine.

Although recharge is often considered to occur equally over the full extent of the mine, this is not true. Many factors influence the recharge, such as rainfall and rainfall intensity, surface topography, depth of mining, geological structures, presence of subsidence, surface structures, and in case of an opencast mine, the state of rehabilitation.

There are different ways to express recharge and is commonly defined as a volume [L^3], typically m^3. Recharge rates are expressed either as a flux [L^3T^{-1}] or as a flux density [LT^{-1}] (Nimmo, J.R., 2005). Because recharge is depending on rainfall (without rainfall no recharge), the recharge is usually expressed as a percentage of the rainfall.

1.2 Background and rationale.

Almost every coal mine in South Africa has a problem with water, it is either too dry or too wet, or the water quality does not comply with the standard listed in the Water Use License (WUL). When a mine has too much water, and it needs to discharge the (excess) water into the environment (river or stream), the water quality must comply with the Water Use License.

Recharge water, also called *water-make*, entering the mine cavity will be collected in small floor depressions. If the recharge is little, the small amounts of water will evaporate, even before it reaches the floor and is transported by the ventilation system to surface. When the recharge is higher than the evaporation, the volume of mine water will grow and slowly finds its way towards the deeper (lower) parts of the mine floor. But which route will be taken? What will happen when the stream of water grows? Which sections will stay accessible and which sections will flood?

Given the same annual rainfall, the water make will continue to grow while the mine is further developed.

One of the most significant problems in South African collieries is the pyrite contents in the coal seam. Pyrite is iron sulfide (FeS_2) that requires oxygen and moisture for oxidation. This oxidation process releases sulphate, iron and manganese into the water (Hodgson and Lukas, 2011). Because water treatment is expensive, a mine may decide to store excess water in one or more defunct sections rather than treating and releasing it in a river, stream or canal.

Limiting sulphate generation is one aspect in the prevention of water quality deterioration. A mine will have two options to lower the sulphate generation. One, prevent the mine workings from flooding. This way, the mine-water quality can't degrade because there is no water. Or two, the mine may try to flood sections to the roof, thereby excluding oxygen and eliminating the chemical reaction (Vermeulen and Usher, 2006a).

Storing large volumes of water underground is not always without danger. The safest way to store the water is behind the contours, but if that is impossible, special high-pressure seals are constructed to contain the water. These seals have a rating indicating the pressure they can withstand before failure. A typical rating is 400 KPa (4 Bar or 40 m water column).

Not all floodings are intentional; a seal may break freeing the stored water to inundate adjacent parts inside the mine. But what parts of the mine workings will be impacted by a seal failure and where will the water end up? Which route will the water take? Will it go through active areas where it may damage mining equipment, or is there a direct threat to the lives of the workers? In a mining environment, the ultimate water-related questions are:

- How much water is recharged in the mine on an annual basis? (Please read: how much water must be treated?)
- Is it safe to continue mining?
- Is there a place where water can be stored safely?

1.3 Flooding Disasters

Over the years, many accidents have happened in mining. These accidents are divided into eight categories: Fires / Explosions / Flooding / Tailings / Shafts / Subsidence / toxic gas / stockpiles. (Seymour, 2005).

According to Chao Xu and Pingping Gong (2011), water-related disasters in coal mining are divided into four types:

- Surface water disaster
  This type of disaster is kind of expected in opencast mining where surface water may spill over into the daylight workings, but it can also influence underground workings when surface floodwater enters shafts or declines.
- Water disaster by roof collapse (Goaf water)
  When a roof fracture zone collapse and water inrush into the mine.
- Coal floor high pressure
  When artesian water finds it way through the mine floor into the mine.
- Karst collapse column water disaster.
  Karst collapse columns are a unique geological body, commonly found in the Carboniferous-Permian coalfields in northern China. Mining activity near the collapse columns, especially ones that conduct water, can cause serious water inrush accidents.

A table published by Brnich and Kowalski-Trakofker (N.D.) of the Office of Mine Safety and Health (a division of the CDC) shows that the number of flooding disasters is relatively small especially compared to catastrophes resulting from explosions (Table 1).

Table 1: Underground coal mine worker fatalities by disaster type in the USA, 1900-2008 (Brnich and Kowalski-Trakofker, No Date).

Type of Incident	Number of Events	Number of Fatalities
Explosion	420	10 390
Fire	35	727
Haulage	21	145
Ground fall / Bump	14	92
Inundation	7	62
Other	17	199

Generally speaking, mines are more dangerous than non-coal mines because the coal seams and shale deposits may contain methane gas pockets. Methane (CH4) is a colourless, odourless and highly flammable and explosive gas. Opening such a pocket may release the gas in the mine's atmosphere allowing to form an explosive mixture. In the United States, 83% of all accidents and 80% of all fatalities, recorded after 1960, occurred in coal mines (Seymour, 2005).

Too much water is also a recipe for a mining disaster. Opencast mines may get flooded during heavy rainfall events when large volumes of runoff water can flow into the pit. It is also possible for underground workings to get inundated by surface water when surface flood water enters a shaft or decline. It is also possible to initiate a flooding event by drilling through a barrier into a flooded mine compartment. Table 2 shows a list of water-related disasters in the last 50 years.

Table 2: Flooding disaster in the mining industry.

Year	Mine	Type
1968	West Driefontein (ZAF)	Water inrush from solution cavities in dolomite – No Casualties (Seymour, 2005)
1973	Lofthouse Mine (GBR)	Water inrush – Seven Casualties (Seymour, 2005)
1977	Porter Tunnel (USA)	Water inrush – Nine Casualties (US mine rescue association) [accessed 28/03/2020]
1989	Emu Gold Mine (AUS)	Flood water (heavy rains) – Six Casualties. (Mine Accidents and Disasters, 2020)
1989	Mahabir Colliery (IND)	Water inrush from an abandoned shaft – six casualties. (ENVIS, 2020)
1994	Merriespruit (ZAF)	Tailings dam disaster – 17 casualties. (The Minerals Council, 2020)
1996	Gretley mine (AUS)	Water inrush from old workings – four casualties (Mine Accidents and Disasters, 2020)
2002	Quecreek Mine (USA)	Water inrush from old operations – nine miners were trapped for three days. (CNN, 2020)
2003	Rostov-on-Don region (RUS)	Water inrush – 11 miners rescued after six days – 2 casualties. (The Guardian, 2020)
2007	Zhijian coal mine (CHN)	Flash flood caused by heavy rain – 69 coal miners rescued after three days. (Reuters, 2020)
2010	Wangjialing coal mine (CHN)	Water inrush from old operations (The Guardian, 31 Mar 2010) [Accessed 26/03/2020]
2019	A coal mine in Yibin's Gongxian county (CHN)	Mine flooding – four casualties. (South China Morning Post, 2020)

South Africa has not been spared from casualties in mining disasters. The largest or deadliest one being the pillar failure Clydesdale Colliery (Coalbrook mine) near Sasolburg in 1960 when 435 miners lost their lives (SAHO, 2020) and the Merriespruit disaster where a tailings dam collapsed, releasing a massive amount of tailings material killing 17 people (1994). Neither of these disasters are in the category of mine flooding disasters. Up to now, the South African mines have been spared any casualties from mine flooding disasters.

The list in Table 2 is probably not complete. But it shows that the frequency of mining disasters has reduced. According to Philippe Dolozme (2019), the number of mining deaths in the USA has decreased from more than 1000 per year in the early 1900s to about 450 annual fatalities in the 1950s and 141 in the 1970s and even lower, around the 30, in the early 2000s. Although mine floodings are happening less frequent than half a century ago, the dangers are still there and are getting higher. While mining continues, more underground section will become abandoned and are allowed to be flooded. With less coal reserve accessible new mining will be closer to the old workings, increasing the possibility of puncturing a barrier between compartments or mines.

1.4 Problem statement

The volume of water entering a mine working is often expressed as a percentage of the rainfall. This percentage is also called the recharge rate. Many attempts have been made in the past to predict the volume of water entering the mine workings (Hodgson and Krantz, 1998; Vermeulen and Usher, 2006b; Van Tonder *et al*. 2007; Lukas, 2018). These volumes depend on recharge rates which in-turn are depending on local factors such as geology, degree of weathering, fractures local and regional, depth of mining, local hydrogeology, rainfall, evaporation, mine method, ventilation plan. Many of these parameters are difficult to determine on their own. To comprise all parameters in one universal value to calculate the water-make is practically impossible. Setting all these factors aside, once the water is inside the workings, it will settle in the small depressions of the undulating floor. As more and more water enters the mining void, depressions will overflow, small water bodies will merge, and water will start to move to the deeper (lower) parts of the mine.

Every mining operation needs water. The coalmine industry is not different. Water is required for the coal preparation plant, during the cleaning of machines, and for dust suppression. But too much water, especially at the mining face, can become problematic. Foreknowledge of water collection areas can help in the development of the life of mine plans, the placing of sumps and pumps and the creation of water storage facilities.

1.5 Study-specific aims and objectives

The title of this thesis *Recharge Flooding of collieries in South Africa* indicates that this document is all about the flooding of underground and opencast mines. Flooding initiated by recharge. Underground mines have sections where mining is completed. These sections are dormant, and recharge water will slowly start flooding the abandoned area. Depending on the amount of rainfall, the recharge factor, the compartment's shape and floor contours recharge will spill-over into the active parts of the mine.

The flooding or movement of water can be described using the energy of the water, either potential or kinetic or both. When using an energy-driven system, ways must be developed consuming the energy in the form of internal resistance (viscosity) and an external resistance as part of the roughness of the surface over which the liquid (water) flows. Creating an energy-driven model needs enormous amounts of data in the form of water density, viscosity, pumping rates (in or out), very detailed water levels, to determine the speed of flowing water, or the flow rate needs to be measured. Systems like these also need to be calibrated before they can be used.

This aim of this project is to determine the resting places of waterbodies after a volume of water is added to a surface. The volume of water can either be added as a point source or as a distributed source over a part of, or the complete, mine floor. The aim of this research will be reached by describing the mine-floor and if applicable the mine-roof using the geometry of the workings, by analysing the floor for depressions and using these to build potential waterbodies. And by creating a flooding algorithm that can decide which waterbody is the next one to flood and which waterbodies must be merged. The system developed here does not calculate the flow as a moving water body but will estimate where the water will become stagnant.

Although all the calculations described in the thesis are based on science, the decision-making process, determining the parts of the surface that are flooded first and next, is based on logic; the water's physical properties like density and viscosity are not considered nor is the surface roughness property or the actual flow and the kinetic energy of the water. All calculations are based on the surface- and water level elevations and the volume of water added to the surface. The water flow is assumed to be instantaneous.

What not to expect.

This thesis will not assist the reader in determining a recharge factor to calculate the volume of the water- make. It assumes a known recharge rate or rates and will focus on where the recharge water will end up after it has entered the workings.

Recharge water is assumed to enter the workings spread over the extent of the mine. The amount of water entering the workings may vary from area to area and is depending on an assumed recharge rate. When the volume of recharge water is larger than the evaporation, it will accumulate on the floor as the water volume grows, it will flow to the lower part of the mine.

What to expect.

This thesis will show the reader a logical method that can determine the final resting place of water bodies given a rainfall amount (mm) and a recharge rate that may be varying over the extent of the mine. The method will make use of a TIN (Triangular Irregular Network) to describe the shape and the contours of the mine floor. When rainfall is simulated all triangles receive the same rain, but the recharge rate may vary between triangles. The flooding takes into account the recharge of each of the TIN's triangles.

1.6 Thesis structure

The research aims and objectives are reached through the development of six interconnected chapters as follows:

Chapter 1: Introduction

This chapter tells the reader about the background and the need for the study.

Chapter 2: Geohydrological processes related to mining

Chapter 2 is all about the groundwater flow and the interaction between the water flow in the geology and the mine void. The recharge processes at open-cast and underground mines. The influence of run-off on these mines and flooding of the workings – controlled flooding and disaster flooding.

Chapter 3: Gap analysis of available software

This chapter dives in the world of the internet where an extensive search was conducted on the availability of software focused on flooding.

Chapter 4: Bridging the GAP – Redefining WACCMAN

Bridging the gap will start with the history from the early days in the 1980s when we developed WISH's predecessor HydroCOM. The route I took to convert HydroCOM into a Windows program called WISH.

The need that prompted me to build WACCMAN a set of routines that can calculate the volume of water in a mine, and lastly the steps needed to calculate the flooding volumes and the places where the water may end up. I do this by explaining the flooding process giving the reader insight into the routines needed to perform these calculations.

Chapter 5: Testing

Chapter five takes the newly added code out for a test drive. Testing is performed on two generated opencast mine pits where two scenarios are performed.

Chapter 6: Case Studies

There are two distinctly different case studies in this chapter. The first case determines the water distribution in a fictive underground mine after different recharge times. The second study is disaster driven and will show the places that will be inundated after water retaining wall failure.

Chapter 7: Conclusions

This is the last chapter with the final thoughts.

2 Geohydrological processes related to mining

Because mine flooding is directly or indirectly recharge driven we will look in more detail to the recharge process before proper attention can be given to the flooding.

2.1 Mine recharge and recharge rate

Mine recharge is the physical process of water entering the mining void. The water may enter the mine from the top, from the sides and the bottom. Recharge can be natural or manufactured, although in the latter case we call it flooding. Any water leaving the mining void is called discharge. This can be a natural process through the bottom or sides (infiltration), over the top (decant) or a artificial process by pumping. A not completely flooded underground mine will lose some of its floodwaters by evaporation if the air above the water table is not saturated. A rehabilitated opencast mine will lose water by evaporation in the final void and evapotranspiration by grasses and trees.

The mining industry is not always interested in all the different components of recharge and discharge. What is essential for them is the speed at which a mining void fills with water, when the mine is full and how much water will decant or must be abstracted and treated to prevent decant.

It is also difficult to measure all the different elements of the recharge and discharge just because the different parts can not be separated and are related to each other. The one thing that can be measured is the water level. The water level measurements, together with the geometry of the mine and the void space or porosity, allows for the volume to be calculated.

The speed at which the mine fills-up is called the recharge rate and is the sum of all the recharge processes minus all the (natural) discharge processes. The recharge rate is thus the net growth of the water volume in the mining void over a time. The recharge also depends on the availability of water, and that's the reason why it is expressed as a percentage of the rainfall.

2.2 Recharge opencast mines

Some attempts have been made to determine a general opencast recharge rate, but it has been found that the recharge rate calculated differs substantially from pit to pit (Lukas, 2019). The industry in South Africa uses the recharge rates suggested by Hodgson and Krantz (1995). Recharge of opencast mines depends on the following factors:

- type of rehabilitation,
- compaction of the spoil,
- the thickness of the topsoil,
- sloping of the surface, and
- the vegetation.

An opencast pit receives rainfall directly on the (rehabilitated) spoil and the final void and ramps. Rain resulting into run-off can flow onto the spoil, the ramps and into the final void. Depressions in the rehabilitated surface will allow for standing water (Lukas, 2012).

The groundwater component of the total recharge of an opencast mine can be compared to placing the bucket with lots of little holes in a pool of water. The rate at which the water enters the buckets is, besides a few other parameters, highly dependent on the difference of the water levels inside and outside the bucket and the size of the holes. The bucket can be filled with a coarse material like gravel to mimic the spoil or backfill material

Water will continue to flow into the bucket until an equilibrium is reached. Adding water to the bucket will result in a higher water level inside the bucket, water will leave the bucket through the holes and will stop flowing when the water levels inside and outside are the same again. When a large volume of water is quickly added to the bucket, water will not be able to flow through the little holes fast enough, and the bucket will start to overflow or decant, see Figure 2-1 (Lukas, 2012).

A rehabilitated opencast mine in an unconfined, homogeneous and isotropic aquifer without any precipitation and evaporation will not decant. If the pit is still dry it will fill-up until equilibrium is reached between the water level in the pit and the water level in the surrounding ground. Water entering the opencast at the upstream side will leave the pit downstream (Figure 2-2) (Lukas, 2012).

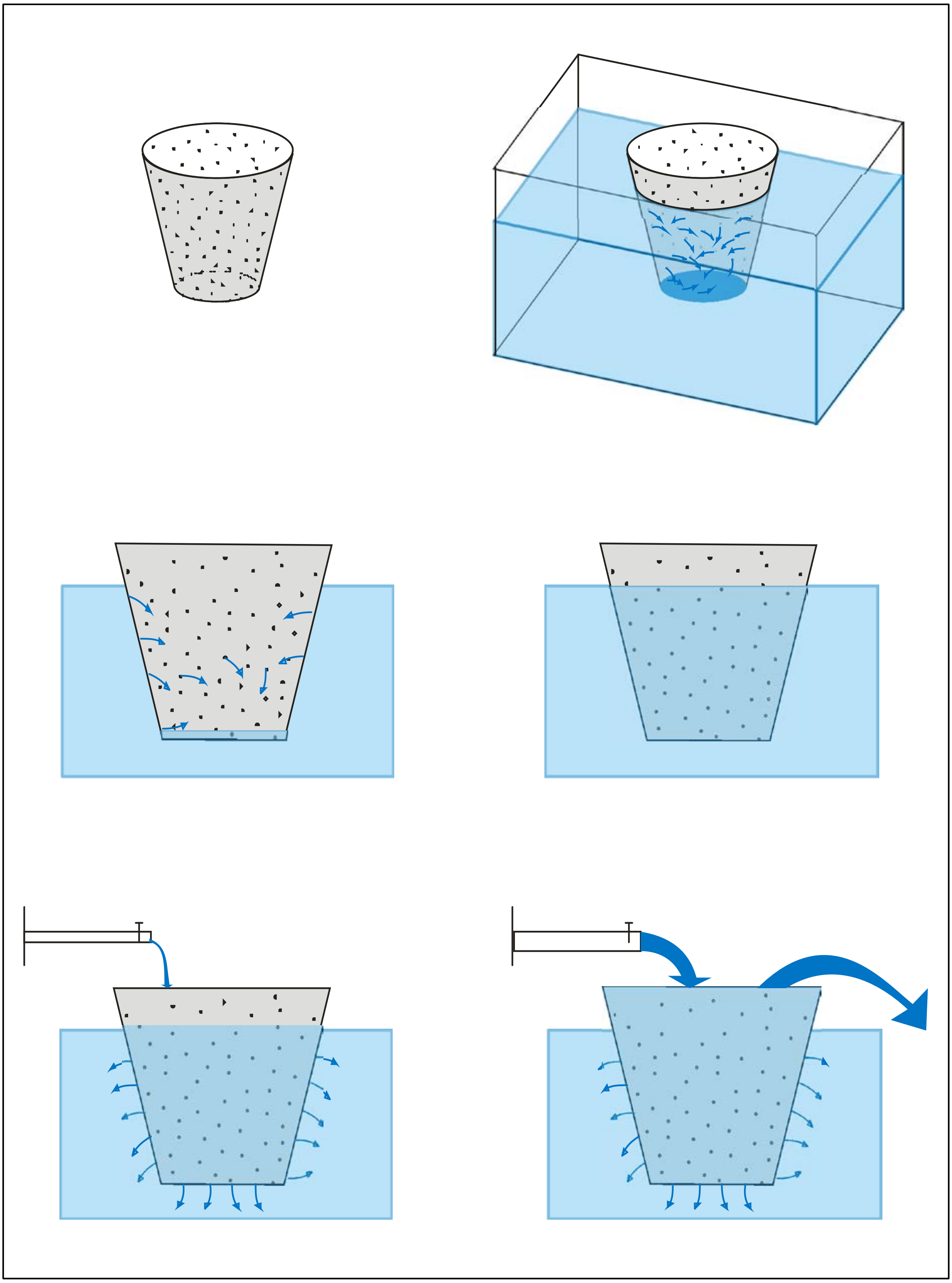


Figure 2-1: Opencast bucket model (Lukas, 2012)

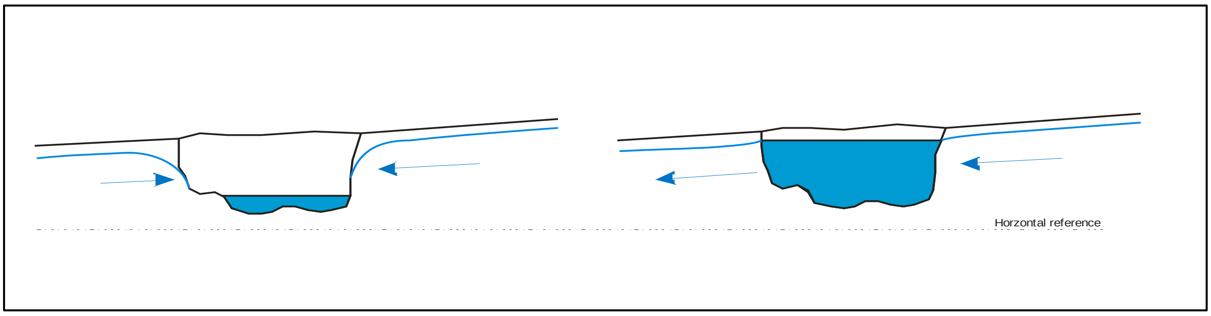


Figure 2-2: Rehabilitated opencast pit without rainfall and evapotranspiration (Lukas, 2012).

The same opencast mine but now with rainfall and evapotranspiration but without run-off will also not decant provided the evapotranspiration is higher than the rainfall or the hydraulic conductivity of the surrounding ground is high enough to allow the water to flow out of the pit. The pit water level will fluctuate with the rainfall events (Figure 2-3).

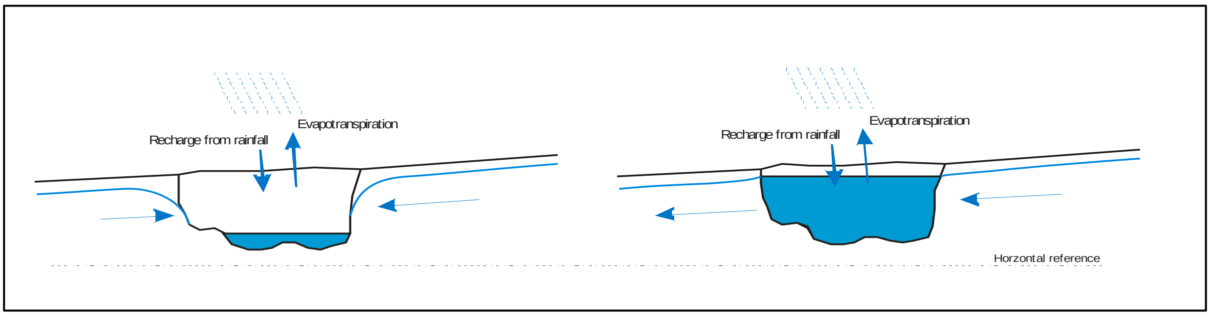


Figure 2-3: Rehabilitated opencast pit with rainfall and evapotranspiration, no run-off (Lukas, 2012).

Adding run-off to the previous scenario changes the picture drastically. The run-off from the surrounding areas towards the rehabilitated spoils and the higher porosity of the spoils (resulting in a higher hydraulic conductance) allows for faster recharge of the spoils (Figure 2-4).

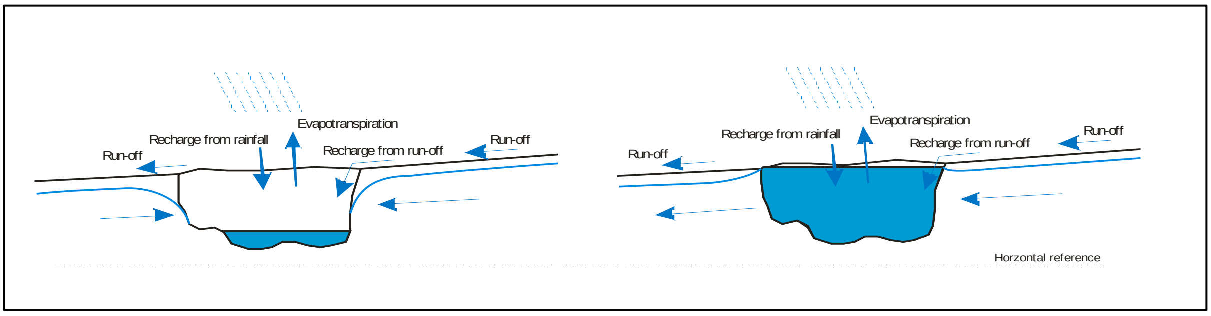


Figure 2-4: Rehabilitated opencast pit with rainfall, evapotranspiration and run-off (Lukas, 2012).

The volume of water that may enter the pit is also dependant on the area surrounding the mine that may create runoff into the pit.

To calculate the volume of water that can enter the pit as run-off It is essential to determine the extent of the area receiving rainfall. All run-off water that flows onto the mining area is considered to be mine water and must be treated before it may be released into the environment. Sometimes this water is necessary to dilute the water already in the pit, but most times the extra water in the pit will result in higher treatment costs. Minimisation of the volume of water in the pit is therefore vital often resulting in in-pit rehabilitation preventing run-off from surrounding areas onto the pit.

Generally, opencast mines have a sizeable areal footprint. Rehabilitating these pits requires a specific rehabilitation plan describing the new surface contours, the depth of the topsoil, and even the types of grasses that are supposed to grow. Different types of rehabilitation or the various stages of rehabilitation will allow for different recharge rates (Tanner 2007; Tanner, 2019). Another influencer is rehabilitation status. Figure 2-5 shows four photos with varying states of rehabilitation. The images illustrate a thick grassland, topsoil only, a thin layer of topsoil and the last picture contains no topsoil at all.

Figure 2-5: Four different states of rehabilitation (E Lukas, 2019)

The surface contour is one of the most important factors when it comes to the flooding of the pit. Ideally, we need to direct all runoff from the pit's rehabilitated surface to the surrounding areas. With less water on the surface, only a small amount of the total rainfall gets the opportunity to infiltrate and percolate through the topsoil past the root zone into the pit. The topsoil, the upper 20~25 cm of the soil and rich of organic material and micro-organisms, is valuable. So valuable that the actual mining process starts by stripping and stockpiling the topsoil. The same topsoil is used to cover the spoil in the pit. During the rehabilitation process, some parts of the surface may not be covered by topsoil for whatever reason, leaving the spoil exposed. The spoil has an open structure, and water can infiltrate quite easily. Because grasses will not grow on the spoil, there will be not much of a root zone, and all water will percolate down.

Table 3 shows the different recharge values assigned to the different areas and the different phases of the rehabilitation process. These recharge values have been calculated from observations at nine opencast collieries within the Olifants Catchment. (Hodgson and Krantz, 1995).

Table 3: Water recharge characteristics for opencast mining (Hodgson and Krantz, 1995).

Water source	Water into opencast [% rainfall]	Suggested average [% rainfall]
Rain onto ramps and voids	20 – 100	70
Rain onto not rehabilitated spoils	30 – 80	60
Rain onto levelled spoils (run-off)	3 – 7	5
Rain onto levelled spoils (Seepage)	15 – 30	20
Rain onto rehabilitated spoils (run-off)	5- 15	10
Rain onto rehabilitated spoils (seepage)	5 – 10	8
Surface run-off from pit surroundings	5 – 15	6
Groundwater seepage	2 – 15	10

"All water has a perfect memory and is forever trying to get back to where it was." – Toni Morrison

---

2.3 Recharge underground mines

Water may enter the underground workings from above through the overlying strata, or from the sides by lateral flow. Due to the layered depositing of the geology, water moves more quickly in the direction of the layering compared to the direction perpendicular to the layering. This results in higher recharge rates from the sides compared to the recharge from above. Due to the shape of the underground mine, the area receiving water from the sides is almost zero in comparison to the area receiving water from the top.

Mines close to the surface using a total- or high extraction mining method have a high risk of collapsing roofs resulting in subsidence on the surface and cracks running from the surface down into the workings. The subsidence will allow for ponding, and the cracks will act like conduits or preferential pathways, enabling water to flow from the surface into the workings (Figure 2-6).

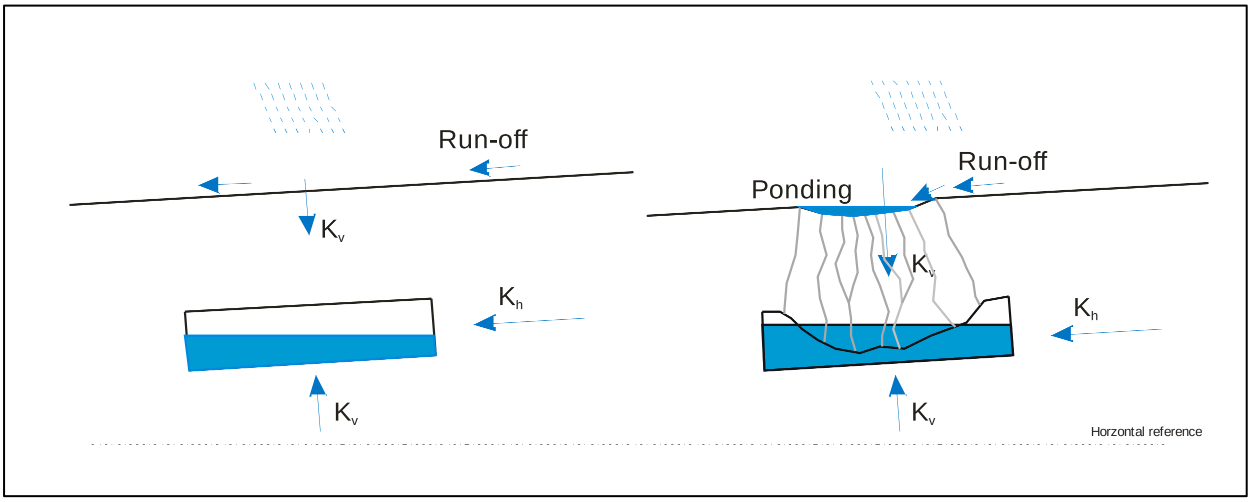


Figure 2-6: Underground mines with and without subsidence (Lukas, 2012).

Without a direct connection between the surface and the underground, all water entering the mine void must flow through the rock. Water will enter the mine void from all sides until the void is completely flooded and the pressure inside the mine and in the rock surrounding the mine is the same (Figure 2-7).

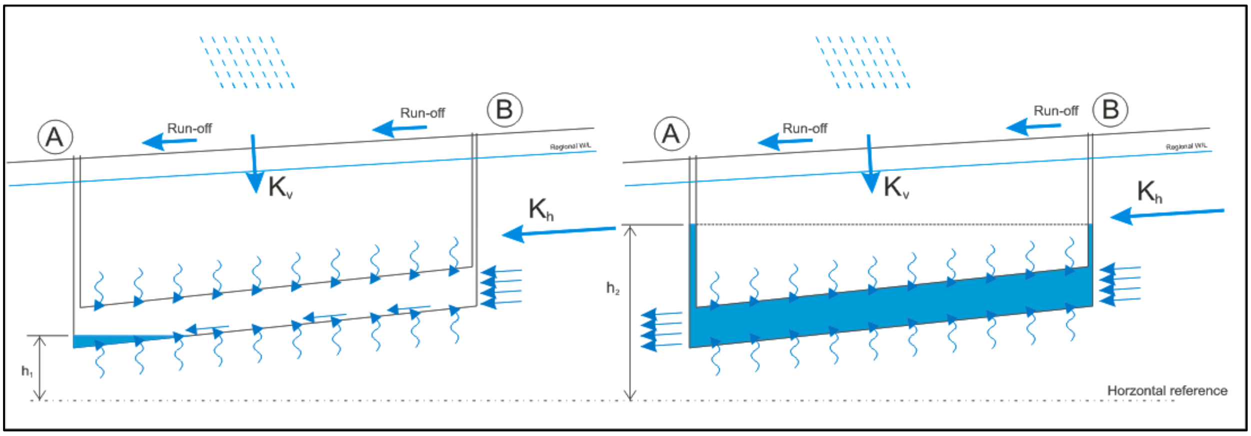


Figure 2-7: Underground workings filling with water (Lukas, 2012).

The hydraulic conductivity of the underground void (the actual mined space) is much higher than that of the surrounding rock. Water will always try to follow the path of least resistance and flow through the workings at a faster rate thereby lowering the pressure inside the workings on the upstream side (B) and creating "space" for extra water to enter from the floor, roof and sidewalls. At the downstream side of the workings, water can leave the filled void with the same rate at which it entered. The extra water will experience congestion when leaving the underground mine (A), the water pressure will be elevated, and the excess water is forced back through the floor and roof into the surrounding rock (Lukas, 2012).

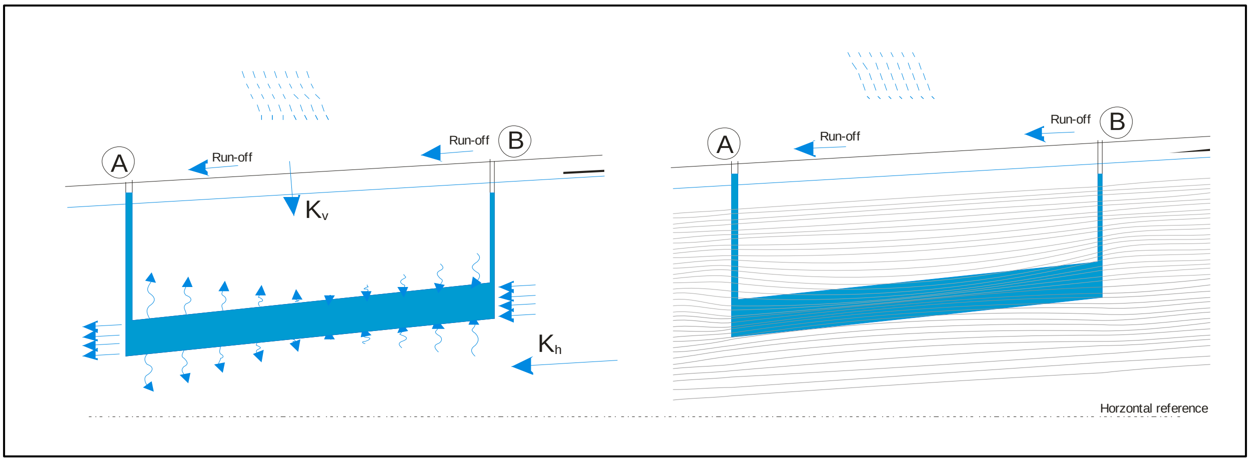


Figure 2-8: Underground working flooded with mine void/formation interaction (E Lukas, 2012).

The elevated pressure will result in a higher water level elevation. Figure 2-8 shows the interaction between the groundwater in the mine void and the formation. A larger throughflow area at the downstream side of the underground will make it easier for the water inside to flow back into the surrounding formation, lowering the pressure and the water level elevation. Removing boreholes-casings at the downstream side is one way of enlarging the throughflow area (Lukas, 2012).

Several investigations into the watermake of underground collieries in South Africa has established a relationship between the depth of mining, the mining method and the influx of water into the Mpumalanga underground mines (Vermeulen 2003; Hodgson *et al*., 2003; Vermeulen and Usher 2006b). It is known that the permeability of the Karoo sediments and dolerite dykes decreases with depth (Flewelling & Sharma, 2014). This is because the calcium carbonate, which is the binding material between the grains of sand (sandstone) and mud (shale), has to some degree been leached by circulating groundwater from the top 40 m of sediments (Annandale *et al*., 2006)

Table 4 below summarises the current understanding of this phenomenon for the Mpumalanga Coalfield (Vermeulen 2003; Hodgson *et al*. 2003).

Table 4: Recharge as a percentage from rainfall.

Mining Method	Recharge as a percentage of the annual rainfall
Shallow Bord-and-Pillar	6 – 9 %
Deep Bord-and-Pillar	1 – 4 %
Partial Stooping	4 – 9 %
Total Extraction	6 – 13 %
Longwall / Shortwall	15 – 20 %
Opencast	20 – 30 %

Deep Bord- and Pillar mining is considered to be more than 40 metres deep. Shallow mining does not allow for total extraction or even partial stooping because this would result in significant subsidence. Partial stooping is the process whereby the remaining pillars are partly removed, leaving thinner (and weaker) pillars to carry the weight of the roof. It is expected that some of the pillars will collapse resulting in limited roof failure. With total extraction, all pillars are entirely removed, roof failure (with surface subsidence) is expected. When a longwall or shortwall mining process is used, the complete coal seam is removed. The mine roof is expected to collapse. The only reason why the recharge rate is lower with the bord- and pillar mining with total extraction is the fact that although the pillars are removed, the walls are still there.

2.4 Runoff

According to Horton (1933), overland flow occurs when rainfall exceeds the infiltration capacity and depression storage capacity. However, for a depression to fill with water, there must either be enough precipitation or runoff. Runoff as such is also a form of overland flow. In a mining environment, focussing on the mine floor, infiltration is assumed to be zero.

Runoff is, from a mining perspective, only of interest when the water, at that time surface water, infiltrates into the soil and becomes part of the recharge water or when it flows directly into a void or ramp in opencast terms or a shaft from an underground mine.

Minimising a mine's possible water-make requires that the surface water from adjacent areas should not drain towards the mine workings. In the case of a rehabilitated opencast mine, the pit surface must have a higher elevation than the surrounding area to ensure runoff towards the virgin ground thereby limiting the surface water resident time on top of the rehabilitated pit. Unfortunately, this is not always possible simply because there is not enough material (spoil and soil) to shape the pit's surface.

Underground mines are most of the time not sensitive towards surface runoff. Some mines have structural problems resulting in roof failures (due to mining depth, mining method, incompetent rock). This may lead to surface subsidence, creating hydraulically preferential pathways from the surface into the mine workings. The subsidence, a depression at the surface, will fill with water (mostly from runoff) and the water will remain there until it has infiltrated or evaporated.

2.5 Flooding

2.5.1 Analysing Flooding

For the researcher to create a computer program emulating or simulating a process, the process needs to be understood in detail. The process needs to be broken up in small parts before it can be described in a computer language. This sub-chapter focus on the first part, the understanding of the flooding process. Many of the processes that occur during flooding are logical to us, but a computer does not know these things. It is a bit like: *You must first learn how to walk before you can run.*

A surface is flooded by adding a liquid, in our case water, onto the surface. When an area is completely level without any local depressions, water can be added equally over the total surface. And the liquid will be distributed evenly, leaving a layer of uniform thickness on the surface. But when the area is tilted, water will flow from the high-end and be concentrated on the low-end of the surface. If the surface contains one or more depressions, water will flow towards the lowest points. The total volume of water will not be different from the amount of water on a level surface; only the place where the water ends up will be different.

For a surface-depression to exist, the depression needs to have a deepest point. It is possible for a surface-depression to have more than one deepest point, but only when there is no rise in the surface between these points.

Puddles of water, *let us call them water bodies*, will always start their existence at and are always centred around the deepest (or lowest) points in the surface depressions. It is also possible to turn this statement around, every lowest point (or cluster of lowest points) will have a single water body.

Therefore a water body will always be associated with a lowest point. Different parts of the surface, *let's call them elements*, may drain towards different depressions. Those parts of the surface where water is flowing towards the same depression are all part of the same waterbody even if they are not flooded. To make things easier, all water that is intercepted by the surface and drains towards the same depression will attribute to the waterbody. It is, therefore, possible to calculate the volumes of water, consisting of intercepted rain, for each waterbody. Although it is needed to calculate the amount of water that falls on the individual surface elements, at the end of the day, we will only work with the total volumes of the water bodies.

After calculating the volume of water for a waterbody, it is possible to calculate the water level of a waterbody.

When more water drains to a waterbody than the waterbody is capable of containing, the waterbody will fill-up and the remaining water will decant onto a part of the surface that is draining towards another waterbody. The volume of the decanting water is added as to the total volume of the receiving waterbody. With more water assigned to the new waterbody, the chances are good that this waterbody also overflows. The overflow point is usually defined as the lowest point of a waterbody where water can freely flow into another waterbody. If this overflow point is the same point where water is entering the waterbody, the two waterbodies will become one when decanting takes place, and the water level in both bodies are the same.

The flow of water or rather the time it takes for the water bodies to stabilise is not considered.

"Everyone should know how to program a computer because it teaches you how to think!" – Steve Jobs

---

2.5.2 Recharge: Underground mine

Flooding as a result of recharge is a very slow or time-consuming process. The speed at which this happens is highly dependant on the permeability of the overlying strata. Using the recharge values suggested in Table 4 and considering an underground mine with a mining height of 3 meters and an extraction rate of 66%. Assume an annual rainfall of 1000 mm, depending on the mining method and the depth of mining between 10 and 200 years, by no means a fast process (Table 5).

Table 5: Time to fill an underground mining void (extraction height 3 meters, extraction rate 66%, rainfall 1000 mm/a)

Mining Method	Recharge as a percentage of the annual rainfall	Time to fill
Shallow Bord-and-Pillar	6 – 9 %	22.2 – 33.3 year
Deep Bord-and-Pillar	1 – 4 %	50.0 – 200 year
Partial Stooping	4 – 9 %	22.2 – 50.0 year
Total Extraction	6 – 13 %	15.4 – 33.3 year
Longwall / Shortwall	15 – 20 %	10.0 - 13.3 year

2.5.3 Recharge: Opencast mines

Performing a similar calculation as for the underground mines shows that the flooding of an opencast mine from recharge will also still take a considerable amount of time. Consider a 40 metre deep filled back opencast pit. With an estimated spoil porosity 20% and an annual rainfall of 1000 mm it will take between 26.7 and 40 years, for 30% and 20% recharge respectively, to flood the rehabilitated opencast pit.

2.5.4 Controlled flooding

The rapid flooding of an open-pit mine can be beneficial for the environment. Quick flooding reduces the amount of sulphate that can be generated by the chemical reaction between pyrite (a mineral found in many coal bodies), water and oxygen because the oxygen is excluded and is replaced by water. In a South African content, most opencast collieries are backfilled with the overburden, levelled, topped with the topsoil and seeded. This is possible because the coal seams are relatively thin compared to the depth where they are found (Wilson and Anhaeusser, 1998). This in contrast with the lignite mines in Germany where the ore bodies are very shallow, almost at the surface. With so little overburden is it impossible to backfill the pit, because there is not enough material. Many examples of actively flooded opencast mines can be found in Germany where large lignite mines are transformed into recreation lakes (Figure 2-9).

Figure 2-9: Timeline (1984-2016) showing the creation of lake Zwenkau by flooding an opencast mine void (Google Earth, accessed).

One way of rapid, but controlled, flooding can be achieved by redirecting a stream or river into the mine void (Schultze, 2012).

Underground mines or parts of underground mines are sometimes flooded in a controlled manner. Flooding of an abandoned mine can not only take place to store water from a water security point of view, but the mine itself can use abandoned mining compartments to be flooded to store excess water during the mining process. When a high watermake occurs, the excess water may, given enough suitable space, be stored underground. This can be done for financial or logistical (or both) reasons, especially when the water is re-used underground as part of the mining processes.

2.5.5 Disaster flooding

A calamity or disaster flooding can happen both in opencast and underground mines. Natural disasters like excessive rainfall may lead to flooding. In many cases, catastrophes like these can be prevented by building diverting walls to redirect floodwater away from a mine void or shaft.

Water entering the workings will not stay in one place. It will collect in the small depressions on the mine floor. When more water is received, the small depressions will start to overflow, and the excess water will flow down to the deeper (lower) parts of the mine. These deeper parts are often the parts where active mining takes place. To make sure the mineworkers have a safe workspace it is necessary that water is intercepted on its route to the deeper parts and diverted to sumps from where it will be pumped to the surface or a storage facility created in an abandoned underground section. When water is stored underground, the sections used to store the water are to be selected strategically. Preferably water will be stored behind the contours. What this means is that the excess water will be pumped to an area that is dipping away from the main road (haulage way). No extra infrastructure is needed (Figure 2-10). If that's not possible because the geometry of the mine does not allow it or if there is too much water, a mine can opt to install seals to retain the water in a section (Figure 2-11). These seals have a rating indicating the pressure they can withstand before failure. A typical rating is 400 KPa (4 Bar or 40 m water column). When a sealed compartment is filled to the roof, any excess recharge water entering the water store via a monitoring borehole can spike the static pressure quickly. Although the researcher is not aware of any seal failure in South Africa, if a seal like this fails, enormous amounts of water will be released into the active mine workings, and the results could be catastrophic.

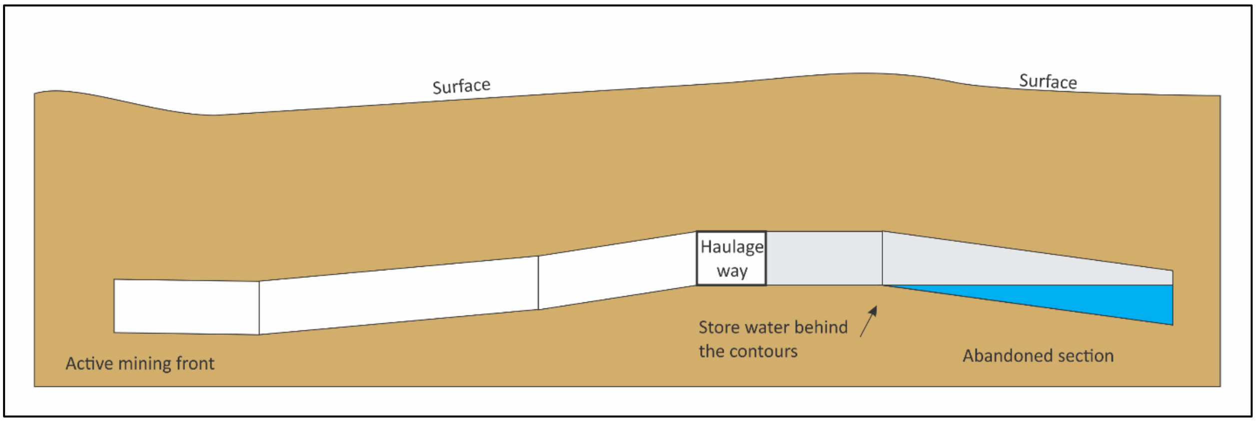


Figure 2-10: Storing water behind the contours.

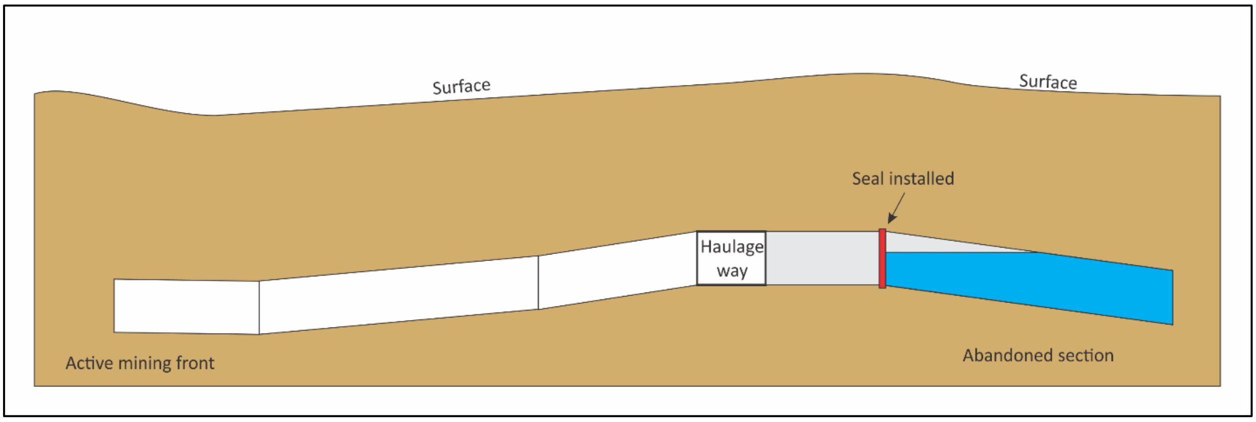


Figure 2-11: Storing water in a sealed-off compartment.

2.6 Previous research

Over the years extensive research has been performed with regard to mine flooding. The bulk of the research is in reports to the mining companies. Many of the research focus on the water quality side of the flooding (Wilkens and van Niekerk, 1991; Scott, 1995). In 2000 Adams and Younger wrote an article accepted in GROUND WATER with the title “A Strategy For Modeling Ground Water Rebound in Abandoned Deep Mine Systems”. The article focus on the turbulent flow of water in a mine void. Marinelli & Niccoli (2000) created an analytical method to calculate water inflow into a mine pit from the sides as well as the floor using symmetrical geometry based on the Dupuit-Forchheimer approximation. The pit was divided into two zones, one representing the pit wall and the other the pit floor. Vermeulen and Usher (2006b) investigated the recharge in South African underground collieries. Van Tonder *et al*. (2007) predicted the mine water rebound in a deep colliery following the closure of the mining operations. In 2007 Rapantova *et al*. presented a paper during the IMWA symposium in Italy with the title “Groundwater Flow Modelling Applications in Mining Hydrogeology”. The article highlights that in the case of backfilling, mine workings represent preferential flow pathways and deep mines are typically situated in hard rocks where the existence of fractures must be expected. Fourie (2015) estimated the decant rate, and with that the recharge rate, from a rehabilitated opencast colliery by making use of a water balance. Dennis & Dennis (2016) used an Analytical Element Model, initially develop by Strack (1989), to calculate water volumes entering the pit with an asymmetrical geometry. Jacek Szczepinski (2019) performed dewatering and flooding of open-pit mines using groundwater models. Donovan and Perry (2019) looked at the mine flooding using a 44-year record of water level fluctuations in a series of adjacent closed underground mines. Ma *et al*. (2020) used a grid-based distributed hydrological model for coal mined-out area. The model is a surface water model where the influence of coal mine subsidence on the runoff is quantified.

3 Gap analysis of available software.

When a mine recharges, water enters the workings (either rehabilitated opencast or underground) and will start to flow to the mine's lower parts, filling local-depressions on its way down. To model the flow of water, we need specialised software. Many models are available also in the water industry. Any software that can emulate/simulate the physical world may be regarded as a model. Many models are focussing on the flow of subsurface water, and there are also many surface water models. Some of them have coupled groundwater and surface water flow. Many models were developed by semi-governmental agencies and are available as opensource software. There are also models developed by organisations and companies on a commercial basis. In general, computer software packages can be divided into five main groups.

3.1 Types of software.

Software packages may be developed from different points of views, different economic models, different development teams, developed by semi-state organisations, private persons, developments houses etc. In general, the software can be divided into five groups:

1. **Proprietary Software**.
   These are software that has restrictions. These restrictions are usually enforced by a proprietor and may have a bearing on how the software is used and whether the user is allowed to copy it. The software can be either commercially- or freely available (Sahoo and Sahoo, 2016).
2. **Open Source Software**.
   Open-source software is computer programmes where the human-readable source code used to create the binaries (executables) is distributed together with the binaries. Any party or person may distribute open-source software. To be considered open-source software, the programs must be distributed freely, and the source code must be included. Because the source code is available, it may also be modified or be used as a whole or in part in another software (Sahoo and Sahoo, 2016).
3. **Shareware**
   Shareware is very similar to open-source software as it can be obtained free of charge. But there it stops. Shareware is copyrighted! Often these free-of-charge downloads have a trial period after which the software must be bought (Sahoo and Sahoo, 2016).

4. **Freeware**

   Freeware is software that can be downloaded from the internet and used for free. The software can also be redistributed free of charge. The software is and remains copyrighted to the original developer/publisher (Sahoo and Sahoo, 2016). Most of these software programs were developed for in-house use. The programs address particular problems and are capable of performing calculations or actions, or combinations thereof, not found in other software. This type of software can not be bought over the shelf, and it can not be downloaded unless a download link is made available explicitly to the requester. A small team of programmers often develops software. Most of these software packages are not user-friendly, do not have extensive help files or even sample data and not a lot of help is available on the internet as the software is intended for a select group of users. (users with the same background or needs.) The learning curve is often steep. The software is usually not for sale.

5. **Commercial Software**.

   This is the group with the best known software packages. Commercial software packages are the computer programs that can be bought over the shelf or via the internet. Many of the packages have large support teams. In the last couple of years, we have seen a new train of thought whereby licenses can no longer be bought but need to be "rented".

3.2 Software requirements.

The usability of software that can predict flooding in an opencast or underground mine depends on the features it supports. Some of these features are general requirements about the interface, and others are necessities more of a technical nature to describe the mine in its environment.

Interface

The interface is that part of the computer program that communicates with the user. Because the mine plans are technical and most of the times complex, the application must be capable of importing the mine drawings. It must also be possible to use existing water level data to create a base map. It must be possible to inspect the mine layout and state of flooding visually. The interface must also be self-explanatory.

Scale

Although collieries can occupy a reasonably large area, for instance, New Denmark mine has a footprint of roughly 13 x 18 km, Sigma mine: 11 x 14 km and Syferfontein 30 x 18 km, the roads and splits in a bord-and-pillar mine are usually less than 6 metres wide (Figure 3-1). To keep the detailed information at hand, we need to be able to work within a large scale with a high resolution.

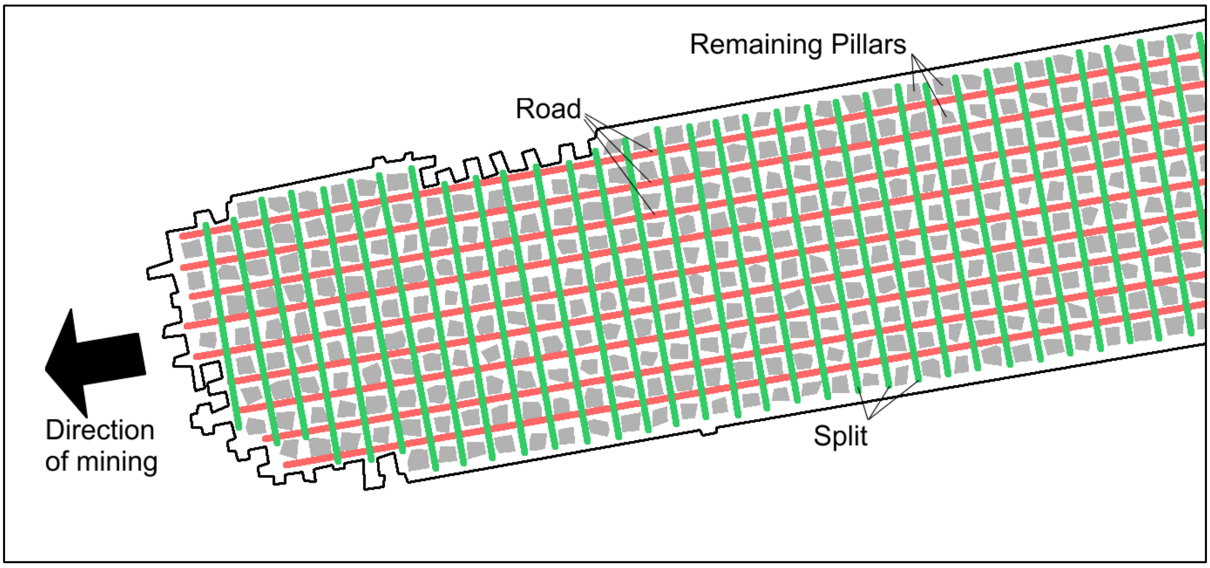


Figure 3-1: Example of a mined section.

Confining heights

Not only the opencast mines but also the underground mines experience mine flooding as a result of recharge. Both mining types are also not immune to disaster flooding. Although the flooding mechanism does not change between opencast and underground mining, the volume calculation do. Flooding with a known volume of water will have a different footprint in an unconfined and a confined system. The program must be capable of simulating a confined system, an unconfined system and even a mixed environment something we can find when an opencast is connected to an underground mine.

Extraction rates

When an underground mine uses the bord-and-pillar method, between 30% and 50% of the orebody is left unmined in the workings to keep the overlaying strata from collapsing. Using software that assumes an open flow like surface flow software or even pipe flow software will make it incredibly complicated if not impossible, to set up the flow domain. If only the mine outline is used to create the model, ignoring the remaining pillars, the calculated volumes of water in the workings will be a gross over-estimation. To compensate the system must make use an extraction factor. The extraction is most probably not the same for the entire mine. Differences in the mining depth and the competence of the overlaying strata will influence the design and measurements of the remaining pillars.

Recharge factors

Different features in an opencast mine have different recharge factor. The ramps and the final void have much higher recharge rates than a rehabilitated surface. Furthermore, the recharge in an opencast mine varies between the different stages of rehabilitation. Recharge in underground mines is dependant on surface structures that allow ponding and the existence of cracks from the mine workings to the surface. In both cases, the recharge rate must be adjustable from area to area. It must be possible to set the recharge rate as low as 0%.

Speed & results

The software must be capable of running swiftly. Long computational times are not an option. The software must always provide an answer and give some sort of a confidence rating.

3.3 A semi-comprehensive grab of software available in the (geo)hydrology

This section lists software that is currently available to model ground and/or surface water. The list also includes some Geographic Information Systems (GISs) as these have modules or extensions available that can analyse or visualise the flow of water. The list contains the most prominent names but is not complete and will never be as the list will keep on changing. A URL where the information about the software was obtained is supplied. After each entry, a little table will highlight the main properties and capabilities of the software.

3.3.1 ArcGIS

ArcGIS is a Geographic Information System that can be extended with many specialised toolsets, including groundwater- and hydrology toolsets. The surface water toolset includes tools to delineate basins, calculate flow direction and accumulation. ArcGIS allows the user to gain more insight using contextual tools to visualise and analyse the data. Furthermore, ArcGIS supports data collaboration and sharing via maps, apps, dashboards and reports.

Name	ArcGIS
Type of software	GIS
Developer	ESRI
Marketing	Commercial
URL	https://www.esri.com/en-us/arcgis/about-arcgis/overview
Date Accessed	30/03/2020
Intended use	Large Area – whole catchment

3.3.2 CCHE2D – Flow

CCHE2D is developed at the National Center for Computational Hydroscience and Engineering, the University of Mississippi. CCHE2D is a numerical model to perform two-dimensional simulation and analyses of free surface flows and the associated processes. It can simulate water flows in rivers, lakes, reservoirs, estuaries, and coasts, including floods and dam-break flows. The processes of sediment transport, morphologic change, pollutant transport, and water quality, etc. dominated by water flows can also be studied using modules of this model.

Name	CCHE2D – Flow
Type of software	Surface Water Model
Developer	National Center for Computational Hydroscience and Engineering, the University of Mississippi
Marketing	Shareware / Proprietary
URL	https://www.ncche.olemiss.edu/cche2d-flw-model/
Date Accessed	30/03/2020
Intended use	Large area

3.3.3 DSS-WISE

DSS-WISE™ Lite is a web-based application that can perform 2D dam-break flood modelling and mapping. DSS-WISE™ Lite is developed by the National Center for Computational Hydroscience and Engineering (NCCHE), the University of Mississippi. The development of the web-based tool and its operation and maintenance is supported by the U.S. Federal Emergency Management Agency (FEMA).

Name	DSS-WISE
Type of software	Model
Developer	National Center for Computational Hydroscience and Engineering, the University of Mississippi with support from the U.S. Federal Emergency Management Agency (FEMA)
Marketing	Shareware / Proprietary
URL	https://www.fema.gov/media-library/assets/documents/175355
Date Accessed	30/03/2020
Intended use	Large area; Dam-break simulation

3.3.4 FEFLOW

FEFLOW is the all-in-one groundwater modelling solution. FEFLOW provides best-in-class for a wide range of applications such as capture zone and risk assessment, geothermal energy, groundwater management, groundwater/surface water interaction, groundwater remediation and natural attenuation, groundwater seepage, saltwater intrusion, porous industrial media and mine-water management.

Name	FEFLOW
Type of software	Model
Developer	MIKE Powered by DHI (part of DHI)
Marketing	Commercial
URL	https://www.mikepoweredbydhi.com/products/feflow
Date Accessed	30/03/2020
Intended use	Groundwater modelling

3.3.5 FLO-2D

The FLO-2D model was conceptualised in 1986 to predict mudflow hydraulics. The US Federal Emergency Management Agency (FEMA) supported the initial model development and first application to Telluride, Colorado, in 1988. Over the past 30 years, FLO-2D has become the most widely used commercially available flood model. What sets FLO-2D apart from other hydrologic and hydraulic models is its capability to simulate urban flooding in high resolution and unlimited detail, including the storm drain system. Using elements as small as 3 m, FLO-2D is a superior model in terms of volume conservation, speed, numerical stability and detail. FLO-2D is simple to set up and even simpler to edit (no mesh regeneration).

Name	Flo-2D
Type of software	Model
Developer	FLO-2D Software, Inc.
Marketing	Commercial
URL	https://www.flo-2d.com/
Date Accessed	30/03/2020
Intended use	Flood-Modelling over large areas.

3.3.6 Flood Modeller

Flood Modeller's industry-leading 1D and 2D solvers allow you to model large and complex river, floodplain and urban drainage systems. It is suitable for a wide range of applications, from calculating simple backwater profiles to modelling entire catchments to mapping potential flood risk for whole countries.

Name	Flood Modeller
Type of software	Model
Developer	Jacobs
Marketing	Commercial
URL	https://www.floodmodeller.com/
Date Accessed	30/03/2020
Intended use	Flood-Modelling over large areas.

3.3.7 GMS

GMS is a Groundwater Modelling System developed by Aquaveo. GMS is used for building and simulating groundwater models; it includes geostatistics capabilities (2D and 3D) and stratigraphic modelling. GMS is the Graphics user interface for several models (MODFLOW, MODPATH, MT3DMS, RT3D, FEMWATER, SEEP2D).

Name	GMS
Type of software	GUI / Visualisation software
Developer	Aquaveo
Marketing	Commercial
URL	https://www.aquaveo.com/software/gms-groundwater-modeling-system-introduction
Date Accessed	30/03/2020
Intended use	Groundwater modelling

3.3.8 GoldSim

GoldSim is a powerful and flexible platform utilising a graphical, object-oriented interface for visualising and dynamically simulating complex systems in engineering, science and business. In a sense, GoldSim is like a "visual spreadsheet" that allows you to create and manipulate data and equations graphically. GoldSim moves beyond spreadsheets. GoldSim is not a model in the sense that it can provide answers on physical problems but instead take the outcomes from other (traditional) models and support the development of what-if scenarios making it easier for the user to evaluate how systems evolve and predict their future behaviour.

Name	GoldSim
Type of software	Visual model
Developer	GoldSim Technology Group
Marketing	Commercial
URL	https://www.goldsim.com/Web/Products/GoldSim/Overview/
Date Accessed	30/03/2020
Intended use	Disaster modelling using what-if scenarios.

3.3.9 GSFLOW

GSFLOW is a coupled Groundwater and Surface-water FLOW model based on the integration of the USGS Precipitation-Runoff Modeling System (PRMS-V) and the USGS Modular Groundwater Flow Model (MODFLOW-2005 and MODFLOW-NWT). GSFLOW was developed to simulate coupled groundwater/surface-water flow in one or more watersheds by simultaneously simulating flow across the land surface, within subsurface saturated and unsaturated materials, and within streams and lakes.

Name	GSFLOW
Type of software	Modelling
Developer	United States Geological Survey
Marketing	Shareware / Freeware
URL	https://www.usgs.gov/software/coupled-ground-water-and-surface-water-flow-model-gsflow/
Date Accessed	30/03/2020
Intended use	Groundwater / Surface water flow model

3.3.10 GSSHA

Developed with the US Army Corps of Engineers Engineering Research and Development Center (USACE ERDC), the GSSHA model is a significant reformulation and enhancement to perform Watershed & Floodplain Analysis of Single Events or Long-term Studies. GSSHA is capable of calculating flows, stream depths, and soil moistures in a variety of hydrologic regimes and conditions.

Name	GSSHA
Type of software	Model (Finite Difference)
Developer	U.S. Army Corps of Engineers / Aquaveo
Marketing	Commercial
URL	https://www.aquaveo.com/software/wms-gssha
Date Accessed	30/03/2020
Intended use	Hydrologic, Hydraulic & Groundwater Modelling

3.3.11 HEC HMS

The Hydrologic Modelling System (HEC-HMS) is designed to simulate the complete hydrologic processes of dendritic watershed systems. The software includes many traditional hydrologic analysis procedures such as event infiltration, unit hydrographs, and hydrologic routing. HEC-HMS also includes procedures necessary for continuous simulation, including evapotranspiration, snowmelt, and soil moisture accounting. Advanced capabilities are also provided for gridded runoff simulation using the linear quasi-distributed runoff transform (ModClark). Supplemental analysis tools are provided for model optimisation, forecasting streamflow, depth-area reduction, assessing model uncertainty, erosion and sediment transport, and water quality.

Name	HEC HMS
Type of software	Modelling
Developer	U.S. Army Corps of Engineers, Hydrologic Engineering Center (HEC)
Marketing	Shareware / Freeware
URL	https://www.hec.usace.army.mil/software/hec-hms/
Date Accessed	30/03/2020
Intended use	

3.3.12 HEC RAS

This software is developed by the U.S. Army Corps of Engineers, in the Hydraulic Engineering Center (HEC) to be used to analyse and calculate river flow. The software allows the user to perform:

- One-dimensional steady flow to calculate water surface profiles based on one-dimensional energy equation. Energy losses are introduced through friction.
- One- and two-dimensional unsteady flow calculations component can be used in all types of flow regimes (subcritical, supercritical, hydraulic jumps and drawdowns).
- Sediment transport/mobile bed computations, and water temperature/water quality modelling.
- Water quality analysis.

Name	HEC RAS
Type of software	Model
Developer	U.S. Army Corps of Engineers, Hydrologic Engineering Center (HEC)
Marketing	Shareware / Freeware
URL	https://www.hec.usace.army.mil/software/hec-ras/
Date Accessed	30/03/2020
Intended use	

3.3.13 Infoworks ICM

InfoWorks ICM is an advanced integrated catchment modelling software. You can model complex hydraulic and hydrologic network elements quickly, accurately, and collaboratively. InfoWorks ICM helps you plan for capacity improvements, system expansions, and emergency scenarios.

Name	Infoworks ICM
Type of software	Modelling software
Developer	Innovyze
Marketing	Commercial
URL	https://www.innovyze.com/en-us/products/infoworks-icm
Date Accessed	30/03/2020
Intended use	Hydraulic and Hydrologic Modelling Software

3.3.14 iRIC

iRIC software is a free numerical simulation platform supporting a wide variety of computational solvers for problems in water science and engineering. The software began as a river-flow and morphodynamics analysis tool but has now expanded to treat a much more comprehensive suite of problems including predictions of flooding, rainfall-runoff generation, tsunami propagation, debris flows, habitat assessment and more.

Name	iRIC
Type of software	Modelling software
Developer	International River Interface Cooperative
Marketing	Freeware
URL	https://i-ric.org/en/about/
Date Accessed	30/03/2020
Intended use	River flow and morphodynamics analysis tool

3.3.15 MIKE Flood

MIKE FLOOD is the unique toolbox for professional flood modellers. It includes a wide selection of specialised 1D and 2D flood simulation engines, enabling you to model any flood problem - whether it involves rivers, floodplains, flooding in streets, drainage networks, coastal areas, dams, levee and dike breaches, or any combination of these. The core elements in MIKE FLOOD are our well-proven models, MIKE HYDRO River for rivers, MIKE URBAN for collection systems and MIKE 21 for 2D surface flow. These are coupled to form a unique and trend-setting three-way coupled modelling tool.

Name	MIKE Flood
Type of software	Surface flow model
Developer	MIKE Powered by DHI (part of DHI)
Marketing	Commercial
URL	https://www.mikepoweredbydhi.com/products/mike-flood
Date Accessed	30/03/2020
Intended use	Catchment modelling

3.3.16 MIKE SHE

Integrated catchment modelling, MIKE SHE delivers truly integrated modelling of groundwater, surface water, recharge and evapotranspiration. MIKE SHE includes all important aspects of hydrology when your project requires a fully integrated model. No other tool or combination of tools can match MIKE SHE in terms of seamless integration of all the important processes of the hydrological cycle.

Name	MIKE SHE
Type of software	Surface water and groundwater coupled model
Developer	MIKE Powered by DHI (part of DHI)
Marketing	Commercial
URL	https://www.mikepoweredbydhi.com/products/mike-she
Date Accessed	30/03/2020
Intended use	Catchment modelling

3.3.17 MIKE URBAN

MIKE URBAN is the urban water modelling software of choice when important parameters for model selection are stability, workflow, openness, flexibility, GIS integration and physical soundness. MIKE URBAN covers all water networks in the city, including water distribution systems, stormwater drainage systems, and sewer collection in separate and combined systems.

Name	MIKE URBAN
Type of software	Model
Developer	MIKE Powered by DHI (part of DHI)
Marketing	Commercial
URL	https://www.mikepoweredbydhi.com/products/mike-urban
Date Accessed	30/03/2020
Intended use	Water network modelling

3.3.18 OpenFlows FLOOD

OpenFlows FLOOD is a complete flood modelling software for understanding and mitigating flood risks in urban, riverine, and coastal areas. OpenFlows FLOOD uses fully spatially-distributed numerical models to simulate all hydrological and hydraulic processes that occur in river basins, including rainfall, infiltration, surface runoff, channel flow, and groundwater flow. Additionally, the model can be used to simulate flooding in coastal areas due to storm surges.

Name	OpenFlows FLOOD
Type of software	Modelling
Developer	Bentley
Marketing	Commercial
URL	https://www.bentley.com/en/products/product-line/hydraulics-and-hydrology-software/openflows-flood
Date Accessed	30/03/2020
Intended use	Flood modelling in river basins

3.3.19 OpenFoam

OpenFOAM is the free, open-source CFD software developed primarily by OpenCFD Ltd since 2004. It has a large user base across most areas of engineering and science, from both commercial and academic organisations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to acoustics, solid mechanics and electromagnetics.

Name	OpenFoam
Type of software	Toolbox
Developer	OpenCFD Ltd 2019
Marketing	Opensource
URL	https://www.openfoam.com/
Date Accessed	30/03/2020
Intended use	Toolbox in programming

3.3.20 PumpSim

3D Flow Simulation Software For Pipe Networks

Design, optimise and simulate your pump and liquid reticulation system with Pumpsim™ 3D Pumping Simulation Software. Build a network of pipes, channels, pumps, tanks, valves and sprays to simulate flows and pressures of liquids, slurries and many other fluids.

Name	PumpSim
Type of software	Model
Developer	Howden VentSim (Australia)
Marketing	Commercial
URL	http://www.pumpsim.com/
Date Accessed	30/03/2020
Intended use	Pumping simulation software

3.3.21 QGIS

QGIS is a professional GIS application that is built on top of and proud to be itself Free and Open Source Software (FOSS). QGIS is a user friendly Open Source Geographic Information System (GIS) licensed under the GNU General Public License. QGIS is an official project of the Open Source Geospatial Foundation (OSGeo). It runs on Linux, Unix, Mac OSX, Windows and Android and supports numerous vector, raster, and database formats and functionalities. Many extensions exist for QGIS.

Name	QGIS
Type of software	GIS
Developer	None (*Community-based software*)
Marketing	Opensource
URL	https://qgis.org/en/site/
Date Accessed	30/03/2020
Intended use	

3.3.22 MODFLOW

MODFLOW is the USGS's modular hydrologic model. MODFLOW, first time published in 1984 as a groundwater-flow simulation code, is considered an international standard for simulating and predicting groundwater conditions and groundwater/surface-water interactions. MODFLOW-related programs include capabilities to simulate coupled groundwater/surface-water systems, solute transport, variable-density flow, aquifer-system compaction and land subsidence, parameter estimation, and gw-management.

Name	Modflow
Type of software	Model
Developer	United States Geological Survey
Marketing	Opensource
URL	https://www.usgs.gov/software/modflow-6-usgs-modular-hydrologic-model
Date Accessed	30/03/2020
Intended use	Groundwater modelling

3.3.23 ModelMuse

ModelMuse is a graphical user interface (GUI) for the U.S. Geological Survey (USGS) models MODFLOW 6, MODFLOW–2005, MODFLOW-LGR, MODFLOW-LGR2, MODFLOW-NWT, MODFLOW-CFP, MODFLOW-OWHM, MODPATH, ZONEBUDGET, PHAST, SUTRA 2.2, SUTRA 3.0, MT3D-USGS, and WellFootprint and the non-USGS model MT3DMS. ModelMuse provides a GUI for creating the flow and transport input file for PHAST and the input files for the other models.

Name	ModelMuse
Type of software	GUI / Visualisation software
Developer	United States Geological Survey
Marketing	Opensource
URL	https://www.usgs.gov/software/modelmuse-a-graphical-user-interface-groundwater-models
Date Accessed	30/03/2020
Intended use	Front-end for Modflow, Modpath, Sutra and MT3D

3.3.24 Spring

SPRING (**S**imulation of **PR**ocesses **IN** **G**roundwater) is based on development by the Ruhr University of Bochum. It allows the user to calculate three-dimensional groundwater flow, surface water, heat and contaminant transport.

Name	Spring
Type of software	Model
Developer	Delta-H
Marketing	Commercial
URL	https://spring.delta-h.de/en/index.html
Date Accessed	30/03/2020
Intended use	Groundwater modelling

3.3.25 Surfer

Surfer is used to visualising your data. Surfer has built-in modelling tools to display the data while maintaining accuracy and precision. Numerous analysis tools helps with analysing the data. Adjust interpolation and gridding parameters, assess the spatial continuity of data with variograms, define faults and breaklines, or perform grid calculations such as volumes, transformations, smoothing, or filtering.

Name	Surfer
Type of software	Visualisation
Developer	Golden Software
Marketing	Commercial
URL	https://www.goldensoftware.com/products/surfer
Date Accessed	30/03/2020
Intended use	

3.3.26 SWAT

The Soil & Water Assessment Tool is a small watershed to river basin-scale model used to simulate the quality and quantity of surface and groundwater and predict the environmental impact of land use, land management practices, and climate change. SWAT is widely used in assessing soil erosion prevention and control, non-point source pollution control and regional management in watersheds.

Name	SWAT
Type of software	Model
Developer	Texas A&M University
Marketing	Opensource
URL	https://swat.tamu.edu/
Date Accessed	30/03/2020
Intended use	

3.3.27 SWMM

EPA's Storm Water Management Model (SWMM) is used throughout the world for planning, analysis, and design related to stormwater runoff, combined and sanitary sewers, and other drainage systems. It can be used to evaluate grey infrastructure stormwater control strategies, such as pipes and storm drains, and is a useful tool for creating cost-effective green/grey hybrid stormwater control solutions. SWMM was developed to help support local, state, and national stormwater management objectives to reduce runoff through infiltration and retention, and help to reduce discharges that cause impairment of water bodies.

Name	Storm Water Management Model (SWMM)
Type of software	Model
Developer	U.S. Environmental Protection Agency (EPA)
Marketing	Opensource
URL	https://www.epa.gov/water-research/storm-water-management-model-swmm
Date Accessed	30/03/2020
Intended use	Stormwater management

3.3.28 WISH

WISH, or the Windows Interpretation System for the Hydrogeologist, is a computer program capable of displaying thematic maps with data and graphs depicting the data in a more specialized way. WISH consists of:

- A mapping / graphing facility
- A link to data sets

The mapping facility enables drafting and displaying of maps. By linking data sets databases containing hydro-geological data may be superimposed on the map. The databases can be either in Microsoft Excel or Microsoft Access format. Many data interpretation options are included:

- Time series analysis
- Specialised chemical diagrams
- Pumping test analysis
- Hydrogeological logs.
- Spatial analysis (using point data on a map or plotting point data as contours)

Water balance calculations

- Volume calculations
- Stage curves

WISH was developed especially for the hydrogeologist. WISH is a hybrid between a CAD system, a Geographical Information System, Chemical analysis package, pumping test programs and all other programs a Hydrogeologist will use. Although many of these programs are of a specialist nature and are very powerful, it is the combination that makes WISH unique.

Name	Windows Interpretation System for the Hydrogeologist (WISH)
Type of software	Groundwater interpretations software.l
Developer	Eelco Lukas (the researcher of this thesis)
Intended use	General groundwater interpretation

3.4 Gap analysis.

The software packages listed in the previous sub-chapter are very diverse. A quick recap is shown in Table 6 below. Two of the software packages can be described as GIS packages. Although these programs are not developed to simulate surface hydrology, both packages have plugins that can help with the understanding of surface water problems.

Almost half of the packages are numerical models either for surface water or groundwater or both, but they are developed for use on a drainage regions scale. Most of these packages are only applicable in the surface water (unconfined).

Table 6: Recap of the software list.

Software	GIS	GUI / Visualisation Software	Numerical model	Analytical Model	Groundwater	Surface Water	Stream / River flow	Channel / Pipe flow	Disaster Flooding	Run-off	Extraction Rates / Porosity	Unconfined systems	Confined Systems	Scale	Customisable by the author	Remarks
ArcGIS	x	x		x		x				x				S	½	via spatial analyst
CCHE2D			x			x	x			x		x		S		
DDS-WISE						x			x			x		L-S		
FEFLOW		x	x		x						x	x	x	L-S		
FLO-2D			x			x								L		
Flood Modeller			x			x	x							S		
GMS			x		x						x	x	x	L-S		
GoldSim		x		x												
GSFLOW			x		x	x						x		S		
GSSHA			x		x	x			x			x		S		
HEC HMS			x			x	x					x		S		
HEC RAS			x				x					x		S		Sediment transport
Infoworks ICM			x			x						x		S		
iRIC			x			x	x		x	x		x		S		
MIKE Flood		x	x			x	x		x			x		S		
MIKE SHE		x	x			x								S		
MIKE URBAN		x	x					x		x				S		
Openflows FLOOD		x	x		x	x	x		x	x				L-S		
OpenFoam		x	x	x											½	Software toolkit
PumpSim				x				x					x			Pump simulation SW
QGIS	x	x		x		x								S	½	via plugins
MODFLOW			x		x							x	x	L-S		
ModelMuze		x														GUI for MODFLOW
SPRING		x	x		x	x					x	x	x	L-S		
Surfer		x		x										L-S		
SWAT			?	?	x	x								S		
SWMM			?	?				x						L-S		
WISH	½	x		x	½	x				x	x	x	x	L-S	xxx	

Conclusion

None of the software packages listed has the capabilities needed to perform the flooding simulation in a mining environment. Some packages are GIS-based other specialize in groundwater modelling or are focussing on surface water flow. The groundwater modelling software is not capable of simulating overland flow. Many of the surface flow packages are designed to simulate river flow and flooding planes. Software intended to simulate overland flow cannot be used in confined situations. Software developed for pipe flow can handle confined situations, but for these, it is not possible to implement an extraction rate. Although it is theoretically possible to simulate the water flow in an underground mine using pipe-flow software, practically it is not advisable as it will result in a too complex model trying to describe the roads and splits in a bord-and-pillar mine.

Without the availability of software capable of simulating the flooding, customisation of existing software is needed. Some of the software packages can be customised by making use of macros, extensions and function replacements. Still, the user is always restricted by the functionality the developers allowed to be modified. Many of these customisations via macros and extensions are computer programs written in a language such as Visual Basic which is run through an interpreter, which means that the code is evaluated and executed line by line making the execution of the code slow compared to compiled code.

The Windows Interpretation System for Hydrogeologists (WISH) is a composite package as it has GIS components, but it is not a GIS, it has groundwater components, but it is not a groundwater modelling program. WISH has capabilities not found in the other software. WISH can perform water accounting and management functions by calculating volumes of standing water on a surface and by creating stage volume curves. Customisation of WISH does not depend on macros or interpreters. As I am the original developer of WISH, I also am the holder of the full source code (all 300 000 lines of it). I have full knowledge of all internal workings of WISH and that I can change the functionality of WISH at the core. Theoretically, there is no limitation in the customizability of WISH.

4 Bridging the Gap – Redefining WACCMAN.

4.1 History

The Windows Information System for Hydrogeologist (WISH) was developed over the last 25 years.

The life of WISH started even a few years before Windows came into existence. At that time (end 1980s beginning of 1990s) the Institute for Groundwater Studies (IGS) was developing a package called HydroCom. HydroCom consisted of a database called HydroBase, a mapping facility called HydroMap and a graphing program called HydroGraph. Hydrobase was developed in Clipper, a computer language to create and maintain databases and the front-end (The history of computing project, 2020), and the two parts that evolved into WISH, HydroMap and HydroGraph, were written in Pascal. ProCAD, a locally developed CAD, was used as the graphical user interface (GUI) that ran as an MS-DOS-based program. MS-DOS was the computer operating system developed by the Microsoft Cooperation (Duncan, 1988) that ran on most personal computers (PCs). With the "birth" of Windows NT in 1993, the first 32-bit Windows operating system, the researcher decided to port all the routines from Pascal to **C++ (pronounced: C-plus-plus)** the language of choice to do serious Windows development in those days. With that, the Windows Interpretation System for the Hydrogeologist (WISH) was born. Many parts of WISH were re-developed a couple of times over to result in what we have today, a computer program capable of displaying thematic maps with data and graphs depicting the data in a more specialised way.

NOTE: All text describing the working and internals of WISH and WACCMAN is the work of the researcher and is therefore not referenced.

WISH consists of a mapping / graphing facility and a link to external data. The main window in WISH is used to display spatial background information, superimposing hydro-geological data from the linked data set(s). The main window is also the interface for editing of the background map by adding, removing or changing items on the map. The linked data sets can be either in Microsoft Excel or Microsoft Access format.

"Writing in C or C++ is like running a chain saw with all the safety guards removed." – Bob Gray

---

Many data interpretation options are included:

- Time series analysis
- Specialised chemical diagrams (Piper / Durov / Expanded Durov / S.A.R. / Stiff)
- Pumping test analysis (Cooper-Jacob / Theis / Hantush / Step Draw-down)
- Hydrogeological logs.
- Spatial analysis (using point data on a map or plotting point data as contours)
- Salt load / flux calculations.

Water balance calculations

- Volume calculations
- Stage curves

WISH was developed with the hydrogeologist in mind. WISH is a hybrid between a CAD system, a geographical information system, Chemical analysis package, pumping test programs and all other programs a hydrogeologist will use. Although many of these programs are of a specialist nature and very powerful, it is the combination that makes WISH unique (Lukas, 2012).

The mapping part of WISH enables drafting and displaying of maps. In addition to global settings, the map consists of a layer-list, where all the layers are defined. The layers are always drafted from the bottom layer to the top layer. Layers may be added to or deleted from the list. Layers can be switched on or off and can be moved up and down in the list. Every layer has an item list. The items in these lists are also drawn from bottom to top (or from back to front). The items stored in the layer may be of the following type:

- Point
- Polyline / Polygon
- Rectangle
- Circle / Ellipse
- Text
- Raster (photos and other bitmaps)
- Triangular Irregular Network (TIN)
- Special objects (Cross-section, North arrow, Scale bar, Coordinate grid)

Nested layers (layers in layers) are currently not supported.

The map-items can be added, modified or deleted from the layers.. The appearance of every map item depends on the formatting applied. The format tells WISH how to draw the item on the map. The different formats may be saved as styles to ensure uniformity and rapid formatting (Figure 4-1). The formats contain the following information:

- Fill colour
- Line colour / type / thickness
- Line endpoints (arrows)
- Point type / size / text colour / background colour
- Hatching
- Font typeface / size / attribute (italic, bold, underline etc)

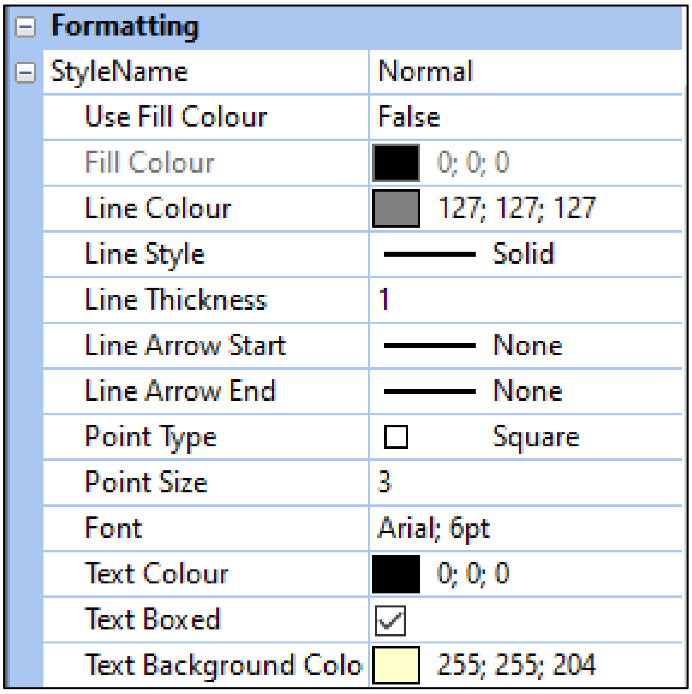


Figure 4-1: Formating settings for a map item.

The WISH document structure is displayed in Figure 4-2. Every map-item also has two sets of properties. The first set is the object properties; this set will be different for different types of objects. These properties describe the object type, the layer on which it resides, minimum and maximum coordinates plus the object dependent properties such as *Perimeter*, *Area*, *Number of nodes* etc. The second set of properties are independent of the object type, and it includes properties like *Name*, *Elevation*, *Contour value* and also properties used in calculations such as *Rainfall*, *RunOff*, *Recharge* and *Active life*.

To create a map from nothing is a timely process. This process can be accelerated by importing existing maps. Two different main file types may be imported:

- Vector type (Shape Files (.shp), AutoCAD (.dxf), Surfer (.bln), Microstation (.dgn) and WISH (.ws2));
- Raster files (Windows bitmap (.bmp), Jpeg (.jpg) and TIFF (.tif))

Regional maps are usually in latitude/longitude, local maps and technical maps are always in x,y.

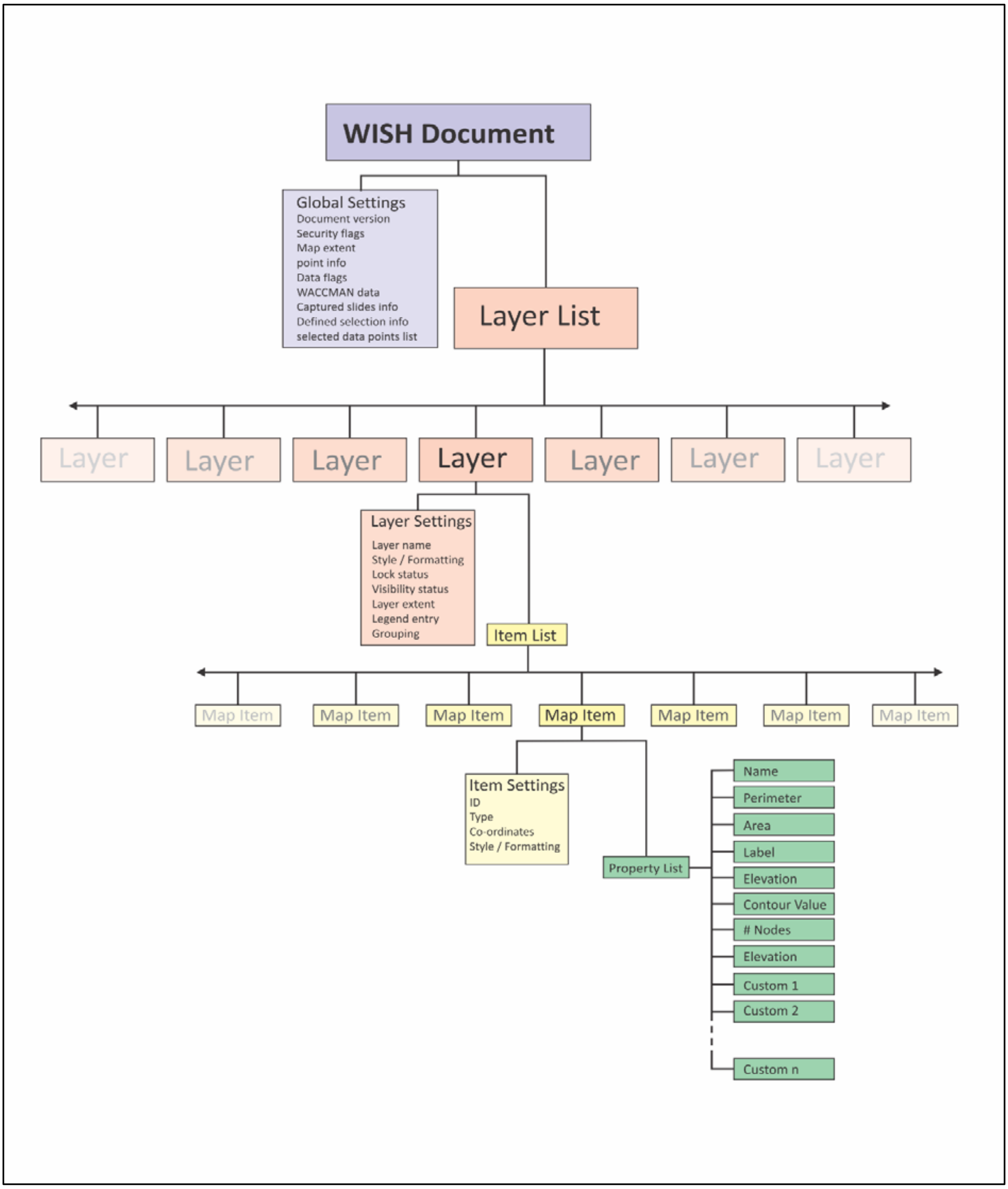


Figure 4-2: WISH Document Structure (E Lukas, 2012).

4.2 Classes in WISH.

To give functionality to every object in WISH, WISH needed to be written in an object-oriented language. Classes were developed for all map items like lines, rectangles, text and TINs (Triangular Irregular Network). A class is an evolution of a variable, it can store data, and it can supply functionality. For example, a class describing a line in WISH, CWMapPoly, has code added that allows the line to draw itself on the map and it has code that will change the line's appearance when the line is selected or deselected (to name two functionalities).

The object-oriented language chosen to write WISH is C++. C++ is a general-purpose, high-level object-oriented programming language created by Bjarne Stroustrup (1997). The language was developed as an extension of the C programming language[1]. Being an object-oriented language, C++ uses classes to describe objects. WISH consists of close to 400 classes, many of them have a sole purpose of handling the individual dialogue boxes (little windows that interact with the user). Classes were developed for all map items like lines, rectangles, text and TINs. The classes do not only contain data like the position of the object on the map, the colours in which the map item must be displayed, but also the functionality, for instance how the item must draw itself on the map or how the object must behave when it is selected or printed. Table 7 shows a short list with the most prominent map objects in WISH.

Table 7: Classes used in WISH to describe the map and the map items.

Class Name		Purpose
CWMap		Class describing the map
CWMapLayer		Class describing the layers of the map.
CWMapItem		Base class for all Map-Objects
CWMapDIB		Class describing Device-independent Bitmaps
CWMapPoint		Class describing Points
CWMapPoly		Class describing Polygons
CWMapRect		Class describing Rectangles and Ellipses
CWMapText		Class describing Text
CWMapTIN		**Class describing Triangular Irregular Networks**
	CWMapTIN_Node	**Class describing the nodes of the TIN**
	CWMapTIN_Element	**Class describing the elements (triangles) of the TIN**
CWMapStyle		Class describing the formatting of the map objects

NOTE: Of the classes displayed, this thesis focus only on the CWMapTIN class and its subclasses.

[1] The C programming language was developed in 1972-73 by Kernighan and Ritchie; All UNIX system programs as well as the majority of the UNIX operating system are written C (Kernighan and Ritchie, 1988).

The user will never work with the classes directly; instead, the user will create and use items or objects by using the graphical user interface (GUI).

4.3 Triangular Irregular Networks.

WISH has the functionality to create 3-D surfaces. These 3-D surfaces can be from the topographic surface, but also from a geological formation, a lithological layer, the floor of a mine cavity or even the roof of the mine cavity. The 3-D surfaces make use of TINs (Triangular Irregular Networks) to describe the surface.

WISH uses the Delaunay triangulation (Delaunay, 1934; De Berg *et al*., 2008) where each node is connected to its nearest neighbour in such way that the vertex forms the side of a triangle. When a circle is constructed through the three nodes of the triangle, the circle will not contain any other node inside its interior.

Every TIN is also described in a class. But the TIN's class is different in the sense that it is a conglomeration of other classes. The TIN class is the overarching class with subclasses for the nodes and elements. The TIN's class, CWMapTIN, contains the 2D geometry of the TIN as well as some initialisation and status data, interpolation settings and lists with the nodes and elements. Besides all the data, the class also provides the functionality to create the nodes and elements, interpolate and assign elevations.

The nodes generated by the TIN are stored in the node-list. The nodes have no physical size, just a position. The position in the horizontal plane is fixed and can not be changed. Elevations are assigned through the interpolations functionality of the TIN. Each of the nodes contains a node number for identification, a list of connecting nodes (every node knows its neighbouring nodes), a list of the elements the node is sharing. The WACCMAN mode of the nodes allows for storing confining elevations and water level elevations.

The elements created by the TIN are stored in the element list. Elements do not have any coordinates assigned; instead, they rely on the positions of the nodes that define them. Each element has an element number for identification and references to the three corner nodes and the three adjoining elements. The WACCMAN mode of the elements stores a recharge factor, extraction factor and a water level elevation.

The creation of a TIN always starts with a polygon. The polygon should not have any duplicate nodes or bowties (crossings). The user needs to specify the maximum distance between the nodes. The node generation process starts with the polygon vertices. Extra border nodes are created between the vertices on an interval of a tenth of the specified distance. Nodes generated in the polygon are on the specified spacing vertically and horizontally (Figure 4-3).

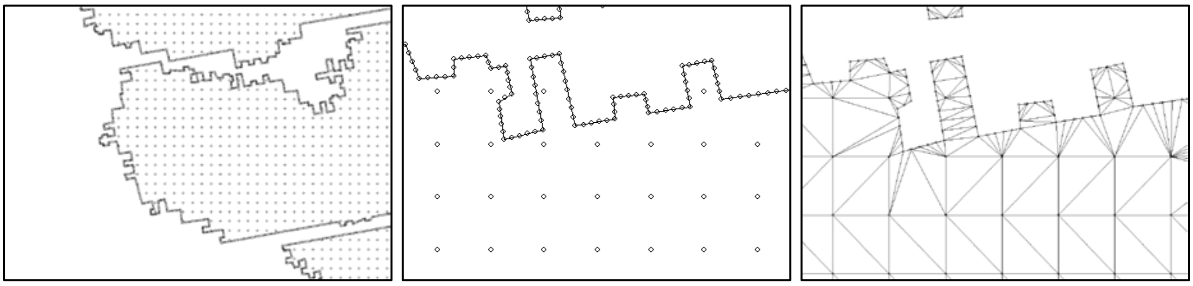


Figure 4-3: Node and element generation.

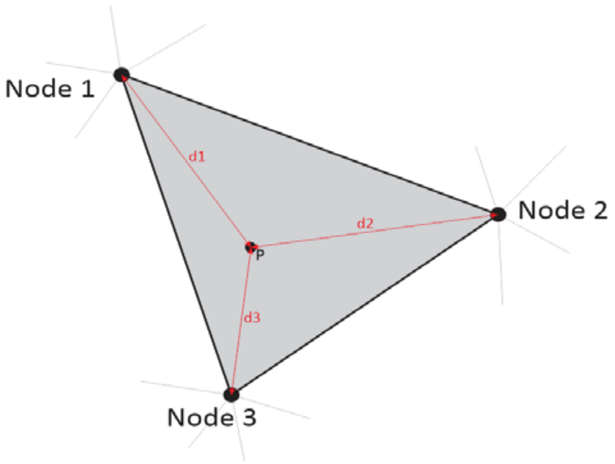
All nodes and all elements are numbered, and all nodes will have x- and y coordinates but a z-coordinate is not yet assigned. The TIN's x- and y coordinates cannot be changed; this means that the TIN cannot be moved in a horizontal plane.

The nodes in a TIN have no dimension; they are infinite small points in space. Only the elements have a size. Because the nodes are considered dimensionless, elements can only connect to each other by their sides. Only the nodes have coordinates (x,y,z). Elements do not have coordinates but are defined by three nodes. If a node's coordinates are changed so are the elements that use this node.

Z-values (coordinates) can be assigned to all the nodes by interpolation of data point values or x-y-z data from an external file. The inverse distance weighting interpolation technique is used to calculate the node values. It is also possible to assign values manually to a single node or a group of nodes. WISH displays a contour as a solid surface (as in contrast to a wireframe where only lines are visible). A set of colours ranging from warm to cool colours are assigned to the nodes. Other colours sets can be assigned manually.

It is also possible to assign values to a TIN using one or two existing TINs. In this process, the values from an existing TIN can be assigned directly (to make a copy) or used in a simple calculation by adding, subtracting, dividing or multiplying a constant or the value of the secondary TIN.

The value can be determined at any place on the TIN, by moving the mouse over the TIN, the x, y and z coordinates are displayed in the status bar; the z-value will also be displayed in a tooltip. Positioning the mouse pointer directly on top of a node will show the nodal value but placing the mouse inside an element an inverse distance weighting is returned. (Figure 4-4).



$$P = \frac{\frac{node1}{d1} + \frac{node2}{d2} + \frac{node3}{d3}}{\frac{1}{d1} + \frac{1}{d2} + \frac{1}{d3}}$$

Figure 4-4: Interpolating a value at a position inside an element using inverse distance weighting.

Each element makes provision for storing the references of three elements directly connected to the sides of the element. When one of the elements is flooded water will flowing to the connected element when the water level reaches the lowest node connecting the elements. This level is called the *connecting elevation.*

4.4 The volume between two TINs.

To calculate the volume of an underground mine WISH needs to know the contours of the mine floor and the contours of the mine roof. Each TIN has the capacity to stoor the floor- and the confining layer's elevations. Floor elevations are assigned from internal or external data, as described in the previous section. The confining values are assigned by:

1. specifying an extraction- or mining height, or
2. by copying the elevations from another TIN.

Option 1 is straightforward in the sense that the confining elevation is calculated by adding the extraction height to the floor elevation.

Option 2 needs a bit more explanation. When two TINs are plotting as layers above each other, WISH can calculate the volume between the two TINs. WISH can do this irrespective of the geometry of the TIN, elements and nodes. WISH can calculate the contour value at any position on the TIN. When that position does not match with a node, WISH will interpolate the value. While this is a fast process is still may take a considerable amount of time if the calculation is repeated a 100K times (the number of nodes for a moderate-sized TIN). To save time it is advisable to use copies of the same TIN for the different layers. Copying a TIN to another layer will duplicate the TIN, but the Z-values are omitted.

Another dataset can be assigned. By preventing the movement of a TIN, identical TINs stay identical. The only difference will be the z-coordinate and calculations can be performed on a nodal basis without a need for interpolation.

When calculating with two TINs, WISH will first determine whether the two TINs have an equal number of nodes, the same extremities and corresponding node positions. Meeting these criteria allows for calculations with just the node values. If not, every value must be interpolated using the inverse distance weighting algorithm.

Figure 4-5 shows two TINs with the same number of nodes, the same extremities and the same node positions. For each node in the upper TIN a corresponding node in the lower TIN exists and vice versa.

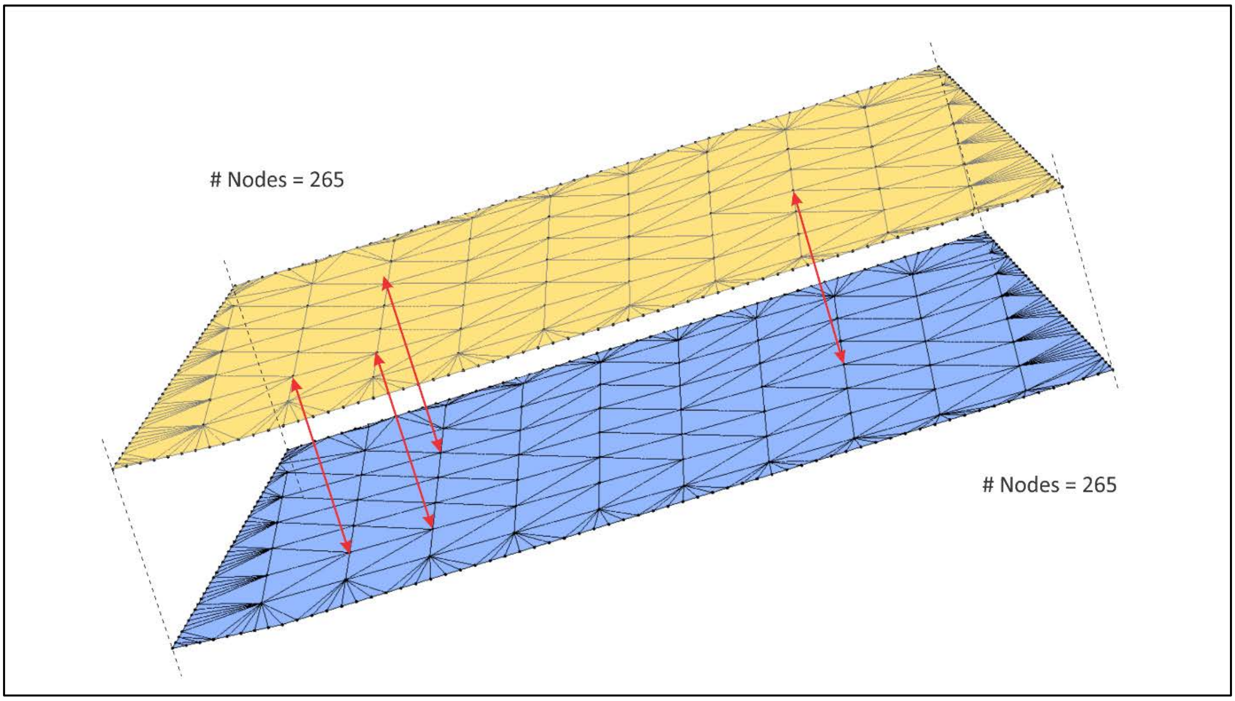


Figure 4-5: Two TINs with corresponding nodes allow calculations just using the node values.

Figure 4-6 shows two TINs with the same extremities but with a different number of nodes. The upper TIN is generated with half the nodal distance from the lower TIN. Because of that, most of the lower TIN's nodes will correspond with the upper TIN's nodes. Evaluating from the other side most of the nodes from the top will not have a node with the same x,y coordinates on the bottom TIN.

The difference between the z-Coordinates of nodes of the two TINs can be calculated. The volume for each element is calculated by taking the average values of the differences between the floor and the roof of the three nodes and multiplying it with the area of the triangle in the horizontal plane. Special care must be taken when the difference is negative. A negative difference occurs when two surfaces cross each other, and the bottom surface is on top of the upper surface. It is the user's task to check that the layers do not cross!

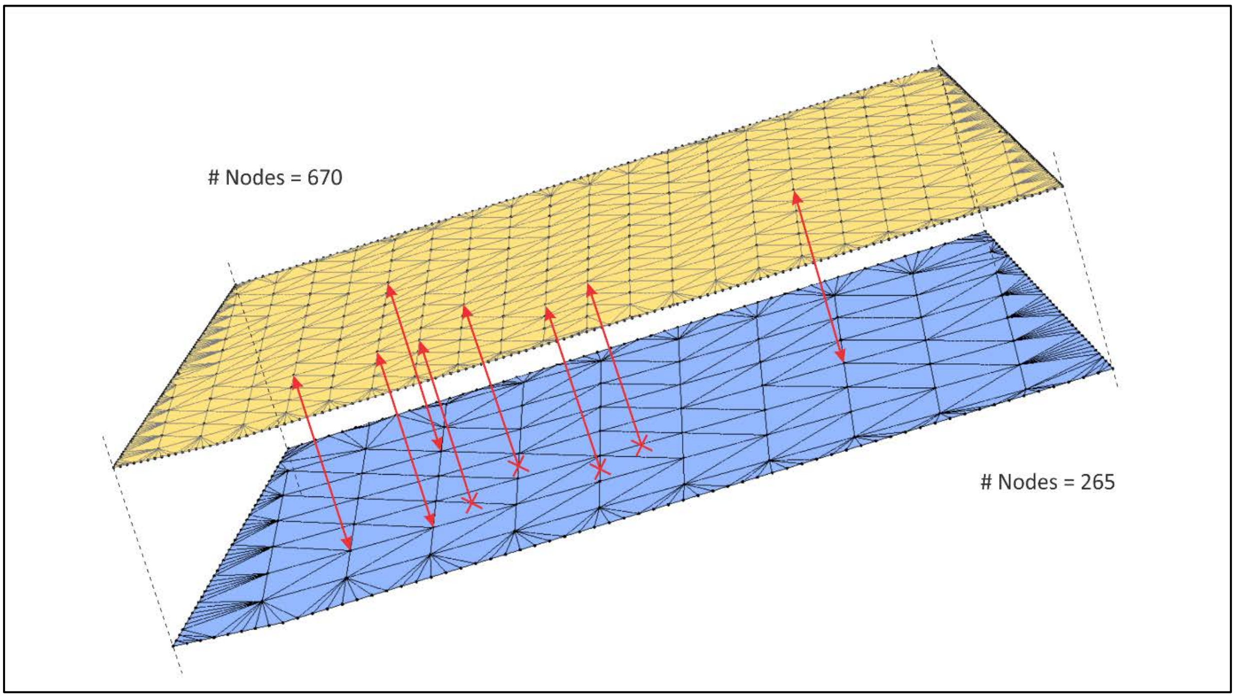


Figure 4-6: Two TINs with different nodes positions requires interpolation to perform calculations.

The volume calculation is performed for each element separately and added together for the TIN. The volume calculation starts the assignment of the confining values for each of the nodes. Next, the area of the floor-element is calculated. The Gauss's area calculation formula (also called the *shoelace formula*) is used to calculate the area of a polygon (Laaksonen, 2018). After the area is known, the volume can be calculated by multiplying it by the average of the confining elevation minus the floor elevation for all three nodes (Figure 4-7). The same method can be used to calculate the amount of water inside the element; special care is taken when some (1 or 2) of the element's nodes are flooded above the confining elevation.

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand." — Martin Fowler

---

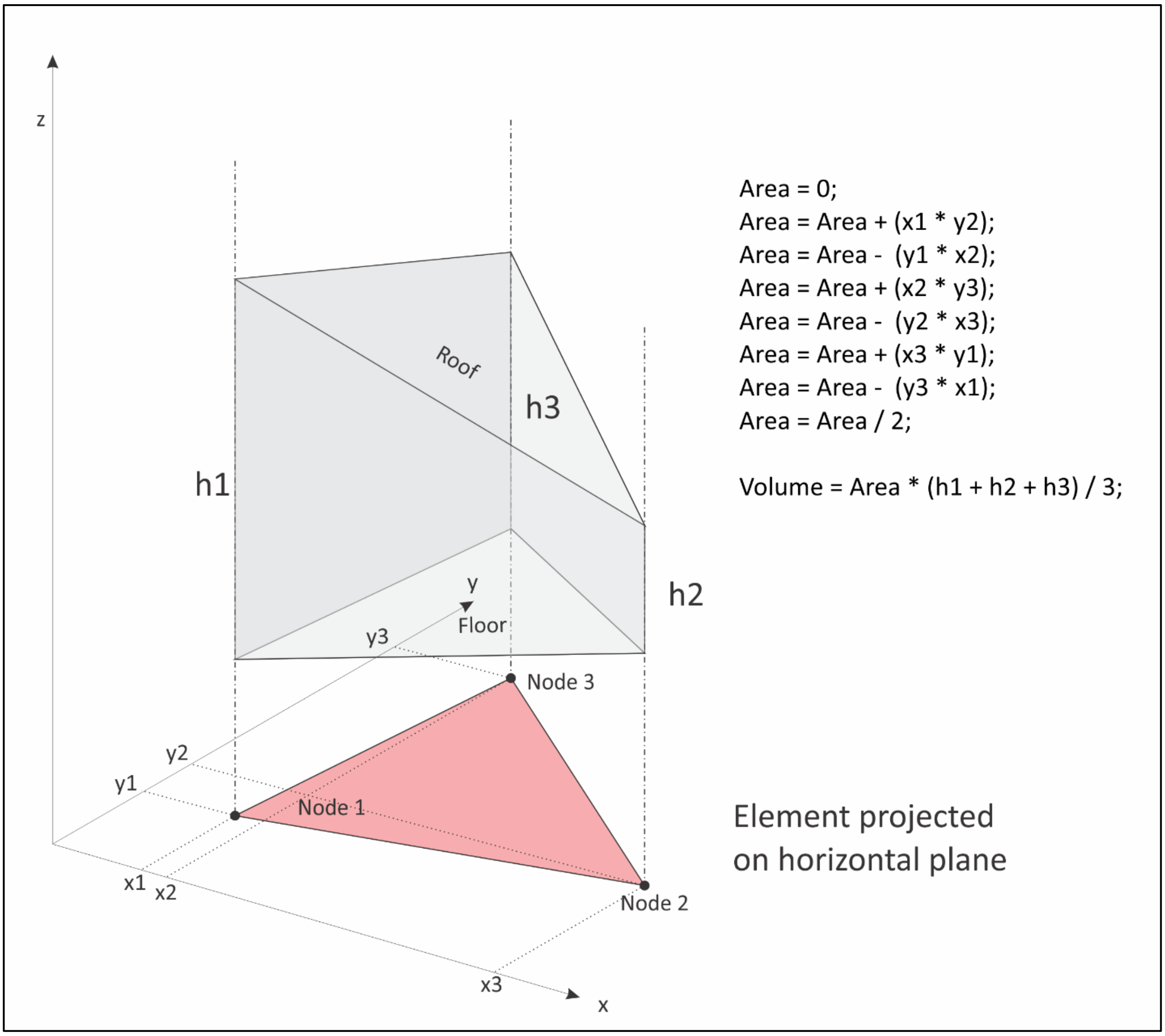


Figure 4-7: TIN's Element with area and volume calculations.

4.5 Water ACCounting and MANagement.

While applying WISH in the mining sector, the need arose to calculate volumes of water that are stored or can be stored in the workings of underground- and opencast mines. Many mines have from time to time a surplus of water. Water that cannot be discharged must be stored for a while.

An extension called WACCMAN was developed to perform **W**ater **ACC**ounting and **MAN**agement operations. Given one or more water level elevations, WACCMAN calculates the maximum amount of water that can be stored on the surface. Water control points (WCP) are used to indicate the water levels on the TIN. A water control point has an x,y coordinate and a water level elevation.

WACCMAN calculates the maximum amount of water by flooding the surface to its highest elevation after which it will remove water element by element starting at the WCP to the level's location where the water level elevation *(target elevation)* is known.

When there is more than one water control point, it is possible that WCPs influence each other. Areas controlled by more than one WCP will always stabilise with the lowest water level. WACCMAN will, therefore, always start dewatering using the lowest WCP as this is a potential time-saver. Elements with water levels lowered according to the lowest WCP, often do not need to be lowered again by higher Water Control Point. Figure 4-8 is used to describe the flooding/dewatering process.

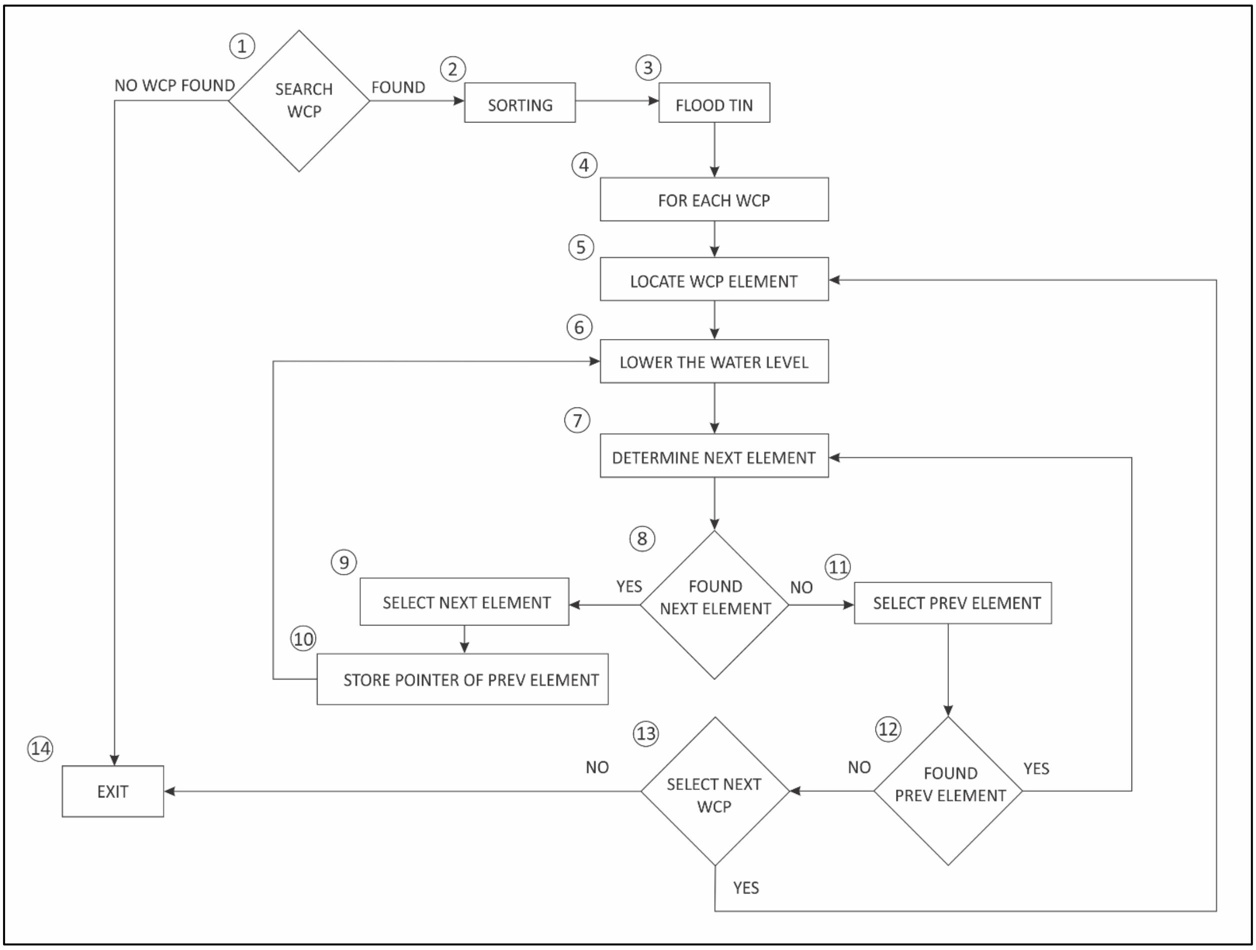


Figure 4-8: Schematic TIN dewatering flowchart.

(**1**) The process begins by identifying the water control points if no WCP was found the function terminates, and no flooding takes place. (**2**) The found WCP are sorted on the target water level elevations. (**3**) The whole TIN is flooded to the highest point of the TIN. If the TIN is confined, WACCMAN will flood using the highest elevation of the confining layer. (**4**) Starting a loop for each Water Control Point, (**5**) the dewatering start-element is determined. The start-element is the TIN's element or triangle where the WCP resides. (**6**) The water level in that element is lowered to the *target water level*. (**7**) The process now selects a connected triangle that has a connecting elevation lower- and a water level higher than the target elevation. (**8**) Here is where the process splits; (**9**) if a suitable triangle is found, the process moves to this element, (**10)** a reference is saved from the previous element, and the process will repeat itself from number (6). If no suitable element was found, the process walks back on its tracks (**11**) by using the references stored in (10). (**12**) If a previous triangle was available, the process returns to number (7). If no last element was stored, it means that the process is back on the place where it started at the start-element. It also means that the dewatering from this Water Control Point is complete. (**13**) The process will now try to select the next WCP. If another WCP is available, the process jumps back to number (5) where it will locate a new start element. When no next WCP is available, (**14**) the dewatering process is finished.

The example in Figure 4-9 shows that the process starts at element number 1 and run to number 13, backtrack to 9, moving forward to 15, backtrack to 1, moving the other way from 16 to 19.

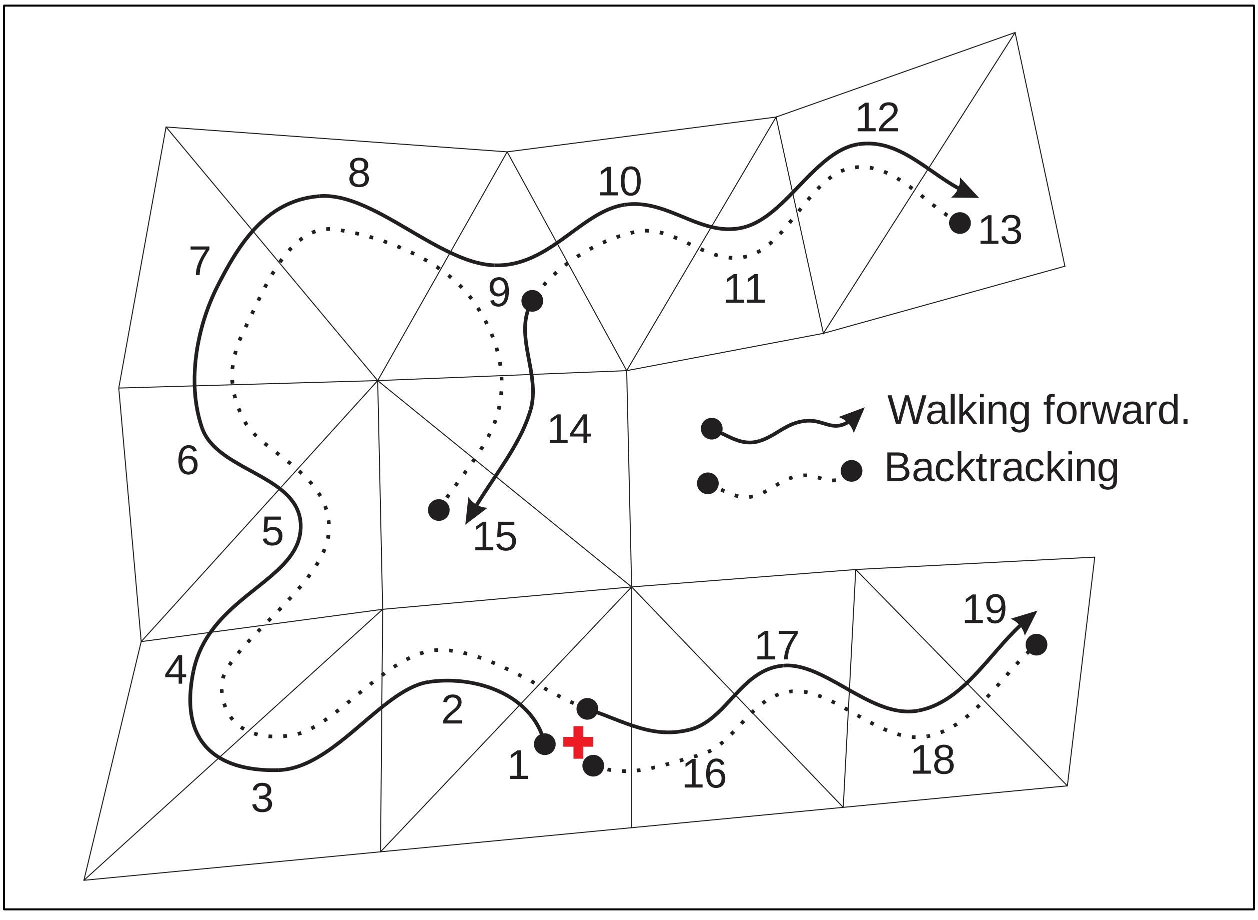


Figure 4-9: Example of the path walked to dewater a flooded area.

This process is repeated for all known water level elevations; the water remaining in connected elements is merged into water bodies. Neighbouring elements with the same water level elevation and a lower connecting-elevations are considered to be one waterbody enabling WISH to calculate a water volume for each waterbody. The volume calculated will be correct for lakes and dams, but the volume of water stored in a rehabilitated opencast mine is affected by the overburden that is placed back in the pit. In these cases, the amount of void space between backfill material, comparable to the storativity, is essential. In underground mines with bord and pillar mining, pillars are left in the mining cavity to prevent roof failure. The extraction factor (percentage of material removed from the mine) is needed to calculate the water volume.

When applying this in the mining industry, extraction factors can be set for every individual element or groups of elements. Extraction factors can usually be obtained from the surveying department on the mine, but if the factors are not available, WISH can help you with determining the extraction factor. For this, you will need the layout of the mine (or a representative part of the mine) and the positions and sizes of the remaining walls and pillars. WISH calculates the perimeter and area for every polygon in WISH, combined polygons are treated as doughnuts, and the area(s) of the hole(s) are subtracted from the area of the outer polygon (Figure 4-10).

While it is theoretically possible to create a TIN of just the mined space in a mining environment, it is not advisable. The polygon used as the source for the TIN describing the outline will need to be combined with the pillars creating holes in the polygon (like a doughnut with multiple holes). Special care must be taken that all the pillars are perfect polygons without gaps, duplicate points, tracking back on the same line, or bow ties (places where the polyline cross itself) before the combining can take place. With so many holes inside the polygon, the circumference will be larger, which means that more nodes must be generated and the creation of the triangles will take longer. Not only the TIN generation but also the assigning of values will take longer as well as the drawing process and subsequent calculations. The file size will grow as well as will the memory requirements. The example is displayed in Figure 4-11 without the pillars 28353 elements / 19318 nodes with the pillars 111463 elements / 97613 nodes (roughly five times the number of elements and the number of nodes).

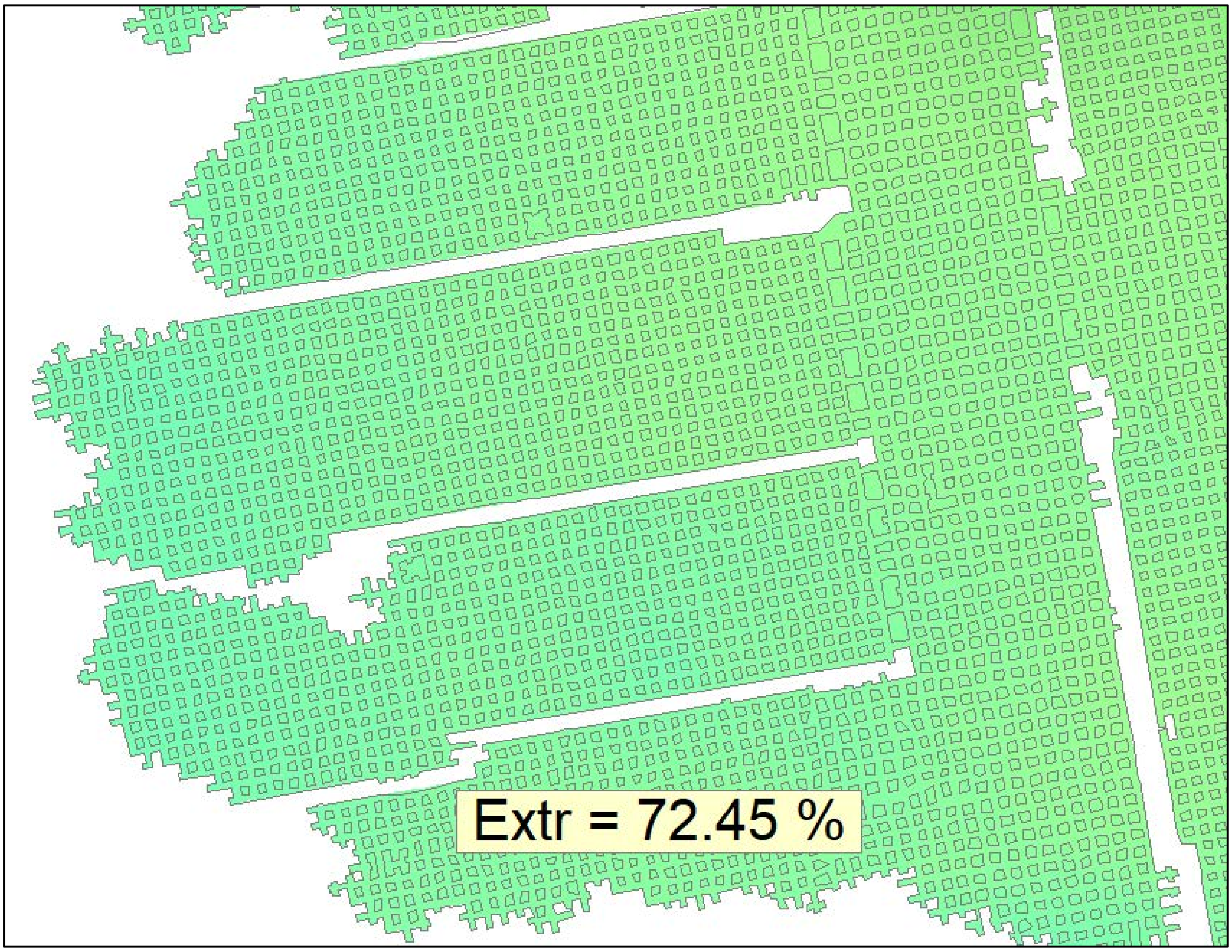


Figure 4-10: Extraction factor.

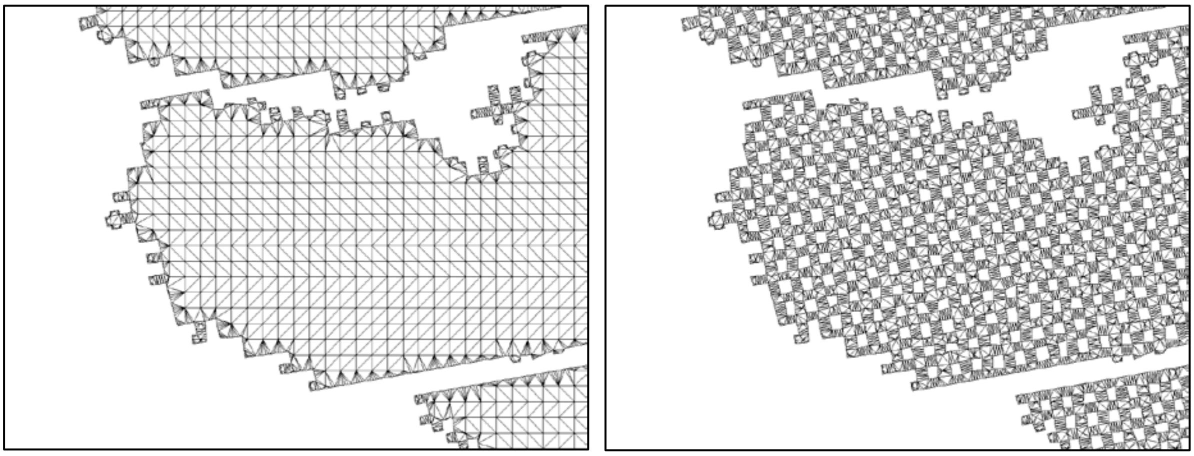


Figure 4-11: The difference in the network-layout without and with the remaining pillars.

In the case of an underground mine, the volume of water that can be stored inside the working is limited by the roof. The confining layer can either be defined by a separate TIN or by a constant height measured from the floor. A 3D example of a mine-floor with a confining layer is displayed in Figure 4-12.

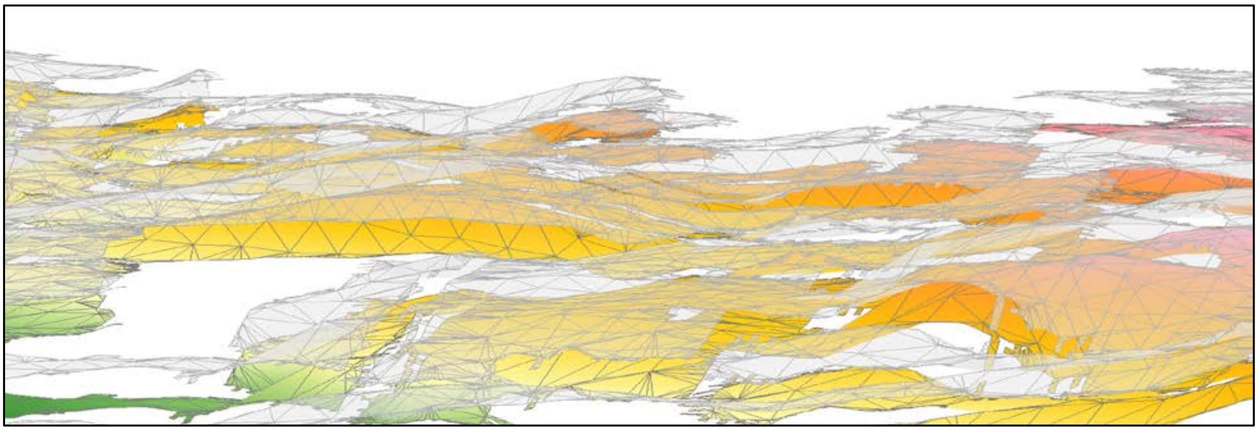


Figure 4-12: 3-D view of the floor of a mine with the roof (confining layer) displayed.

Below is a map of a partially flooded underground mine (Figure 4-13). The known water level elevations are at 1526.25, 1535.51 and 1535.60 mamsl. There are two different tones of blue. The darker blue indicates that the water level is above the roof elevation. The lighter blue indicates that there is still air between the water table and the roof of the mine, in other words, the water level has not reached the ceiling of the underground working. The example uses three water control points, but it is possible to add many more. All waterbodies have a higher or equal water level elevations than the lowest water control point.

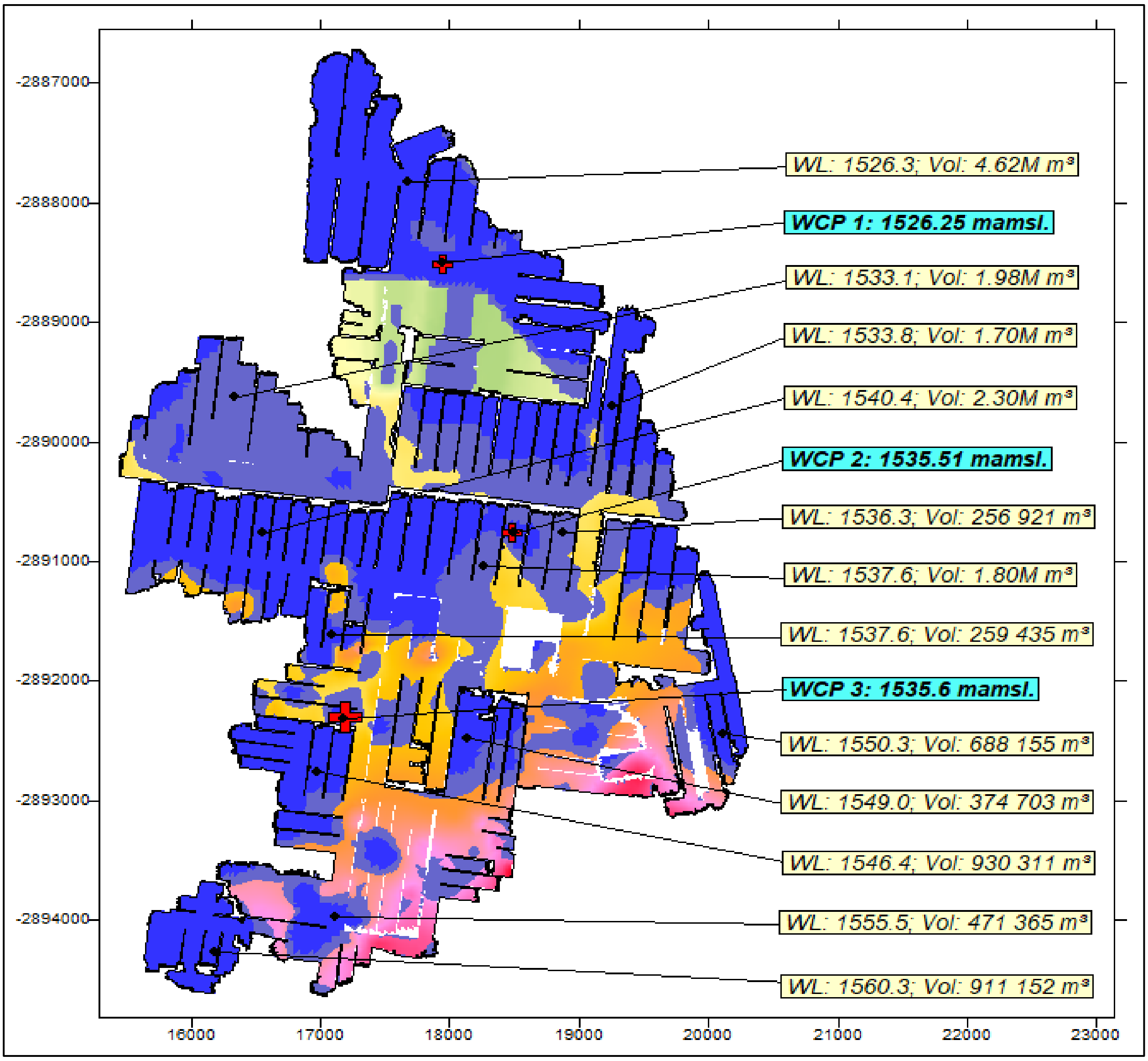


Figure 4-13: Partially flooded workings of an underground mine.

When more information is available, a more realistic answer can be calculated. Not only a water level but also the absence of water is valuable information in the creation of a water map. A layer can be created with polygons indicating areas that are known to be dry (haulage ways, transport belts, safety routes).

When using multiple water control points, some inconsistencies may arise. Differences may occur due to conflicting water level elevations where the elevation is calculated lower than the one specified at the water control point. These errors are almost always the result of inaccurate data (inaccurate floor contours, missing water retaining walls or openings in walls, etc.).

Another feature of the WACCMAN extension is the stage-volume curve. A stage-volume curve describes the relationship between the water level elevation and the volume of water stored on the

TIN. The stage-volume curve uses the full extent of the TIN and assumes a horizontal water table. If there are multiple water bodies, they all have the same water level elevation (Figure 4-14). An example of a stage-volume curve is displayed in Figure 4-15.

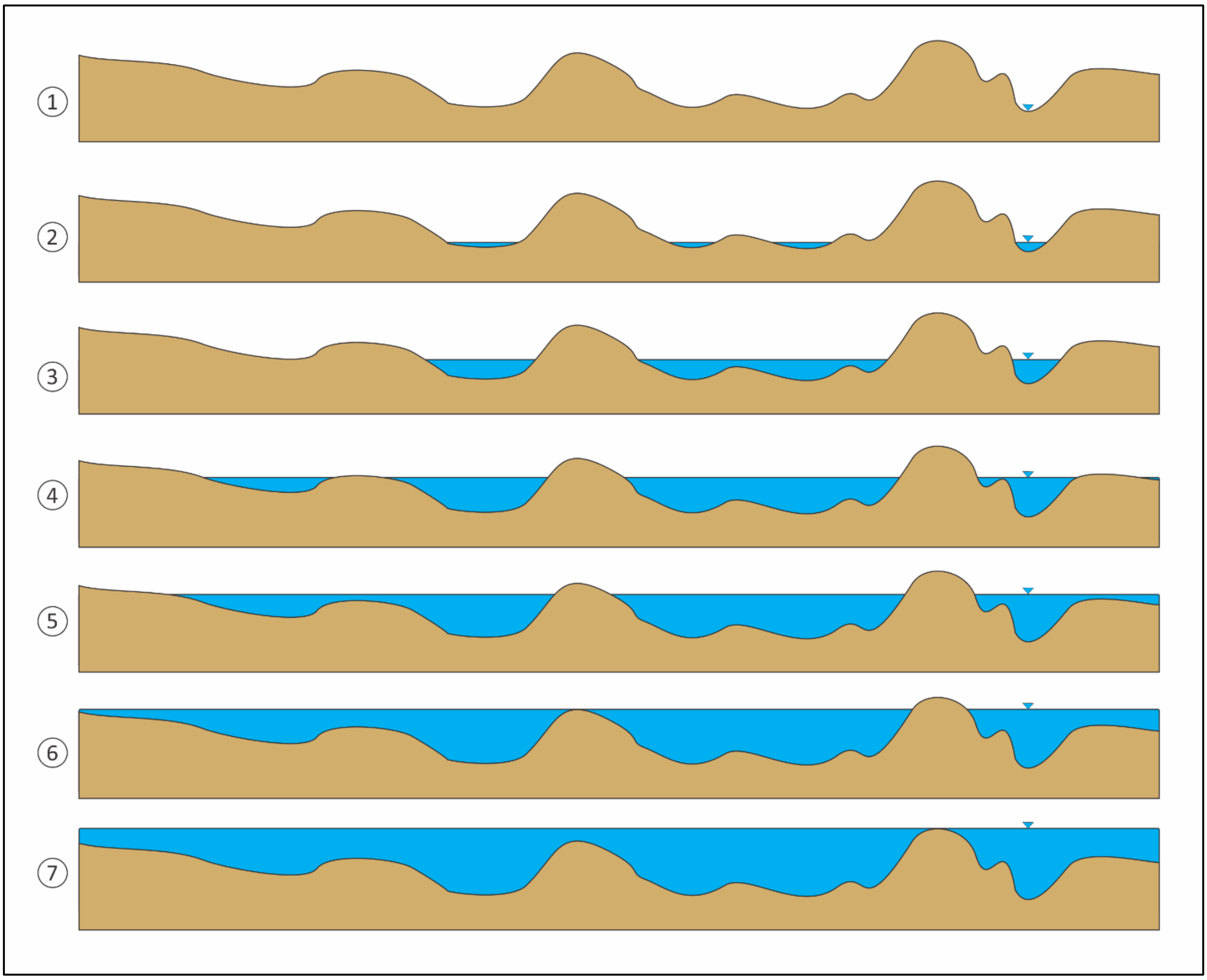


Figure 4-14: Water bodies used in the stage volume curve.

The stage-volume curve is a convenient chart that shows the volume of water stored on a surface at a specified water level elevation. But the graph has its drawbacks when calculating the amounts of water stored. It is assumed that the water level elevations are the same for all water bodies over the extent of the TIN. This can be true in flooded opencast mine but is almost impossible to occur in an underground mine where due to the maze of roads and the undulating floor many larger and smaller water bodies will be in existence. A water-body-stage-volume-curve was developed to calculate a stage-volume curve for a single body of water.

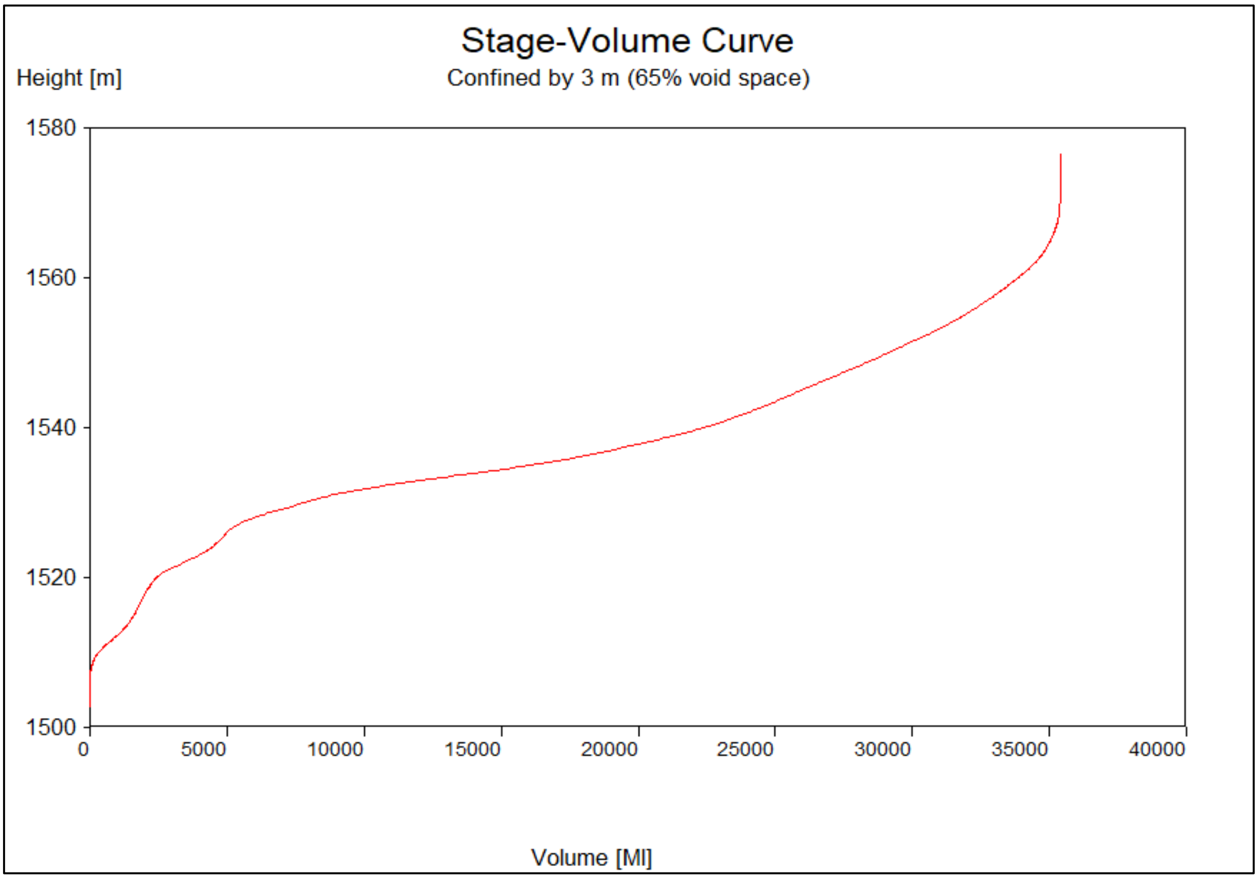


Figure 4-15: An example of a Stage-Volume curve for an underground mine.

The WACCMAN volume calculations are fast. It is a mixture between an analytical model and a numerical model. It is numerical because the area is divided into small regions, triangles in this case. Every triangle is calculated separately but is visited only once for the first Water Control Point (WCP). Subsequent WCPs will only visit those triangles where water levels can be lowered. This is also the reason why the lowering of water levels starts with the lowest target water level. The model is considered to be static and in equilibrium. Time is not considered in the calculation. The model calculates the maximum amount of water in the workings given the surface, if applicable a confining layer and one or more water control points.

In short the WACCMAN model:

- Is fast
- Is accurate (given the information available)
- Will always have a solution
- Needs no calibration.

Although the above model works fine, it is essential to remember the model works on the principle of taking water away from an already flooded surface. Figure 4-16 illustrates the principle of removing water from an undulating floor. The method starts by identifying the WCP, after which the whole TIN is flooded to the highest level on the surface. A Water Control Point can be compared with a pump that is installed at the given elevation. Lowering the water level to WCP will result in water being trapped behind local high points. The water bodies left on the surface will always have a water level elevation equal to or higher than the WCP. The water left on the surface will be the maximum amount considering the surface and the Water Control Points.

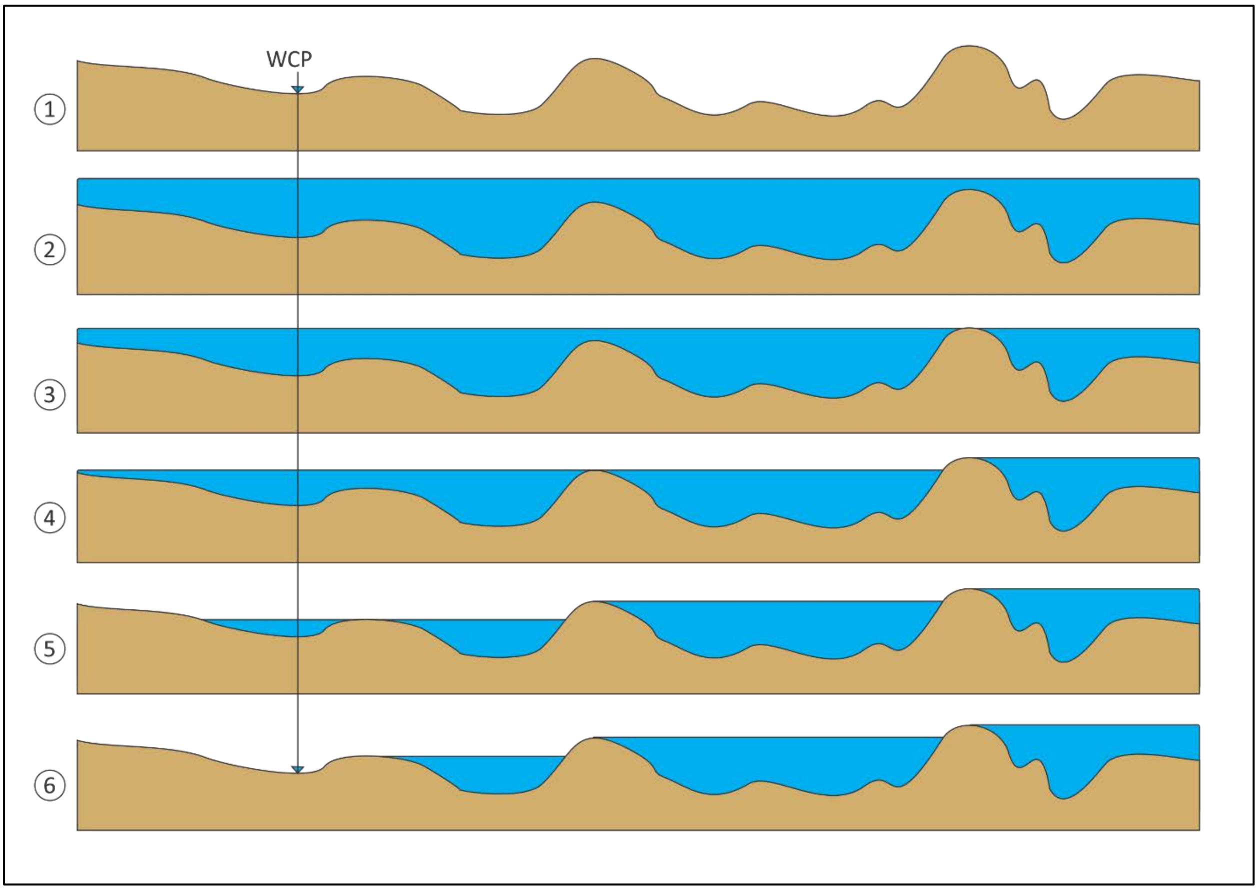


Figure 4-16: Removing water from a flooded surface.

Flooding a surface, as in flooding by recharge, is not just merely reversing the pumping process. During the dewatering (pumping) process, water levels on a surface can never be lower than the lowest installed WCP. While in a flooding process, raising a water level at a point will never allow water to rise above that mark anywhere on the surface. Another fundamental difference is the impact the two different processes have at the beginning. The lowering of the water levels will impact the far-away waters from the moment the dewatering starts. In the case of flooding, the surface will start to be inundated from the WCP and than move outwards (Figure 4-17).

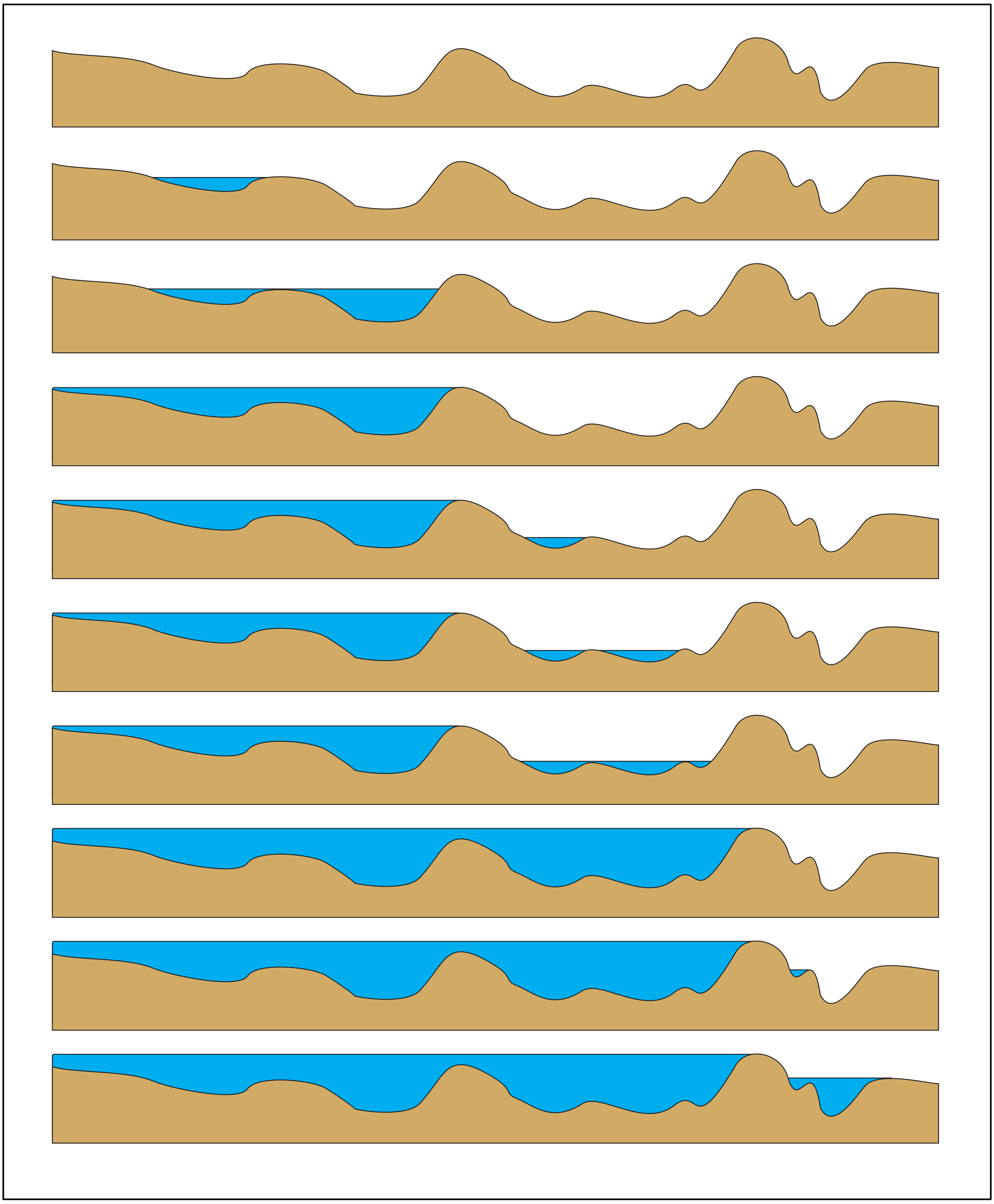


Figure 4-17: Flooding process.

The current WACCMAN model does not consider surface runoff. This is a severe drawback. Without surface runoff calculations, it is impossible to determine where the recharge water will end up.

This thesis only looks at the runoff where it starts and where it ends up. The floor of the mine is considered to be impermeable – infiltration is not possible!

"A computer problem is often located between the back of the chair and the computer monitor." – author unknown.

4.6 The **modified** structure of TINs in WISH

The "old" WACCMAN model works by flooding first and then taking water away. This method calculates, given one or more water level elevations, the maximum volume of water inside the mining void. First, the surface is inundated to the highest surface elevation. Or, in case of an underground mine, to the highest point of the confining layer. Then water is removed at the specified positions to the required elevations as described in section 4.5. This implies that the water level elevation can never be lower than the water level elevation specified. The volume of water on the surface is dependant on the water table elevation.

When the process is turned around, and the water level elevations must be determined based on a volume of water that is added to the surface (either by recharge, pumping or accidental release) the same method can no longer be used. When recharge water enters the mine, it will accumulate, forming water bodies in the depressions of the mine floor. It is of vital importance to know how the elements are connected, to determine the route that a water droplet will take on its way towards a local depression. As more water enters, the waterbody will grow until it is full and starts to overflow. Many of these small water bodies will merge into larger ones. To simulate where the recharge water will accumulate a set of functions needed to be added to the WACCMAN code. A list of these functions is provided in Table 8, and a detailed description of each of these functions is described in the remainder of this chapter. Function- and variable names are always displayed in *italics*.

Table 8: Functions and functionality added to WACCMAN.

Function added	Class affected	Functionality added
BuildConnectivity()	CWMapTIN	Determine the elements surrounding an element.
AnalyseNodeConnections()	CWMapTIN	Build lists of nodes and elements linked to a node.
FindRechargeNodes()	CWMapTIN	Find the lowest points in the local depressions.
BuildWaterbodies()	CWMapTIN & CWMapTIN_WaterBody	Group elements and nodes to describe a water body (WB).
Set TIN properties	CWMapTIN_Element	Change properties such as recharge rate and extraction factor.
FindLowestNodeAboveWB()	CWMapTIN_WaterBody	Identify the lowest not flooded node of a WB.
CalculateCapacity()	CWMapTIN_WaterBody	Calculates the volume that can be stored before overflowing.
CalculateVolumeAtElevation()	CWMapTIN_WaterBody	Calculate the water volume stored at a specified elevation.
FloodToNode()	CWMapTIN_WaterBody	Flood the WB to a given node.
FloodToElevation()	CWMapTIN_WaterBody	Flood the WB to an elevation.
FloodToVolume()	CWMapTIN_WaterBody	Flood the WB with a set volume of water.
MergeWaterBody()	CWMapTIN_WaterBody	Merge two water bodies.
VerifyWaterBodies()	CWMapTIN_WaterBody	Verify the integrity of the WB.
RemoveWaterBodies()	CWMapTIN_WaterBody	Remove waterbodies that are flagged as deleted.
RemoveWaterBodyReference()	CWMapTIN_WaterBody	Remove references of deleted WB.
RechargeByRainfall()	CWMapTIN	Initiate the process of recharge.
FloodTIN()	CWMapTIN	Simulate the flooding of an underground mine.
FloodCalamity()	CWMapTIN_WaterBody	Flood multiple waterbodies with a volume of water.

4.6.1 Build Connectivity – *BuildConnectivity()*

Every triangle or element consists of three nodes and three sides. Each of these sides may be shared with neighbouring elements (also called the connected elements). Each triangle can have a maximum of three connected elements. The first connecting element always joins the side between Node 1 and Node 2. The second join at the side defined by Node 2 and Node 3, and the third one joins at the remaining side (0). The elevation of the lowest shared node between two elements is called the connection height (Figure 4-18).

Air circulation is vital in an underground mine. Without ventilation, not enough oxygen will be available for the miners to do their work. Abandoned mine sections are closed, to direct the flow of air so efficient as possible, using ventilation walls. Without ventilation, the air humidity in these sections will quickly reach its maximum, preventing evaporation and resulting in a built-up of recharge water.

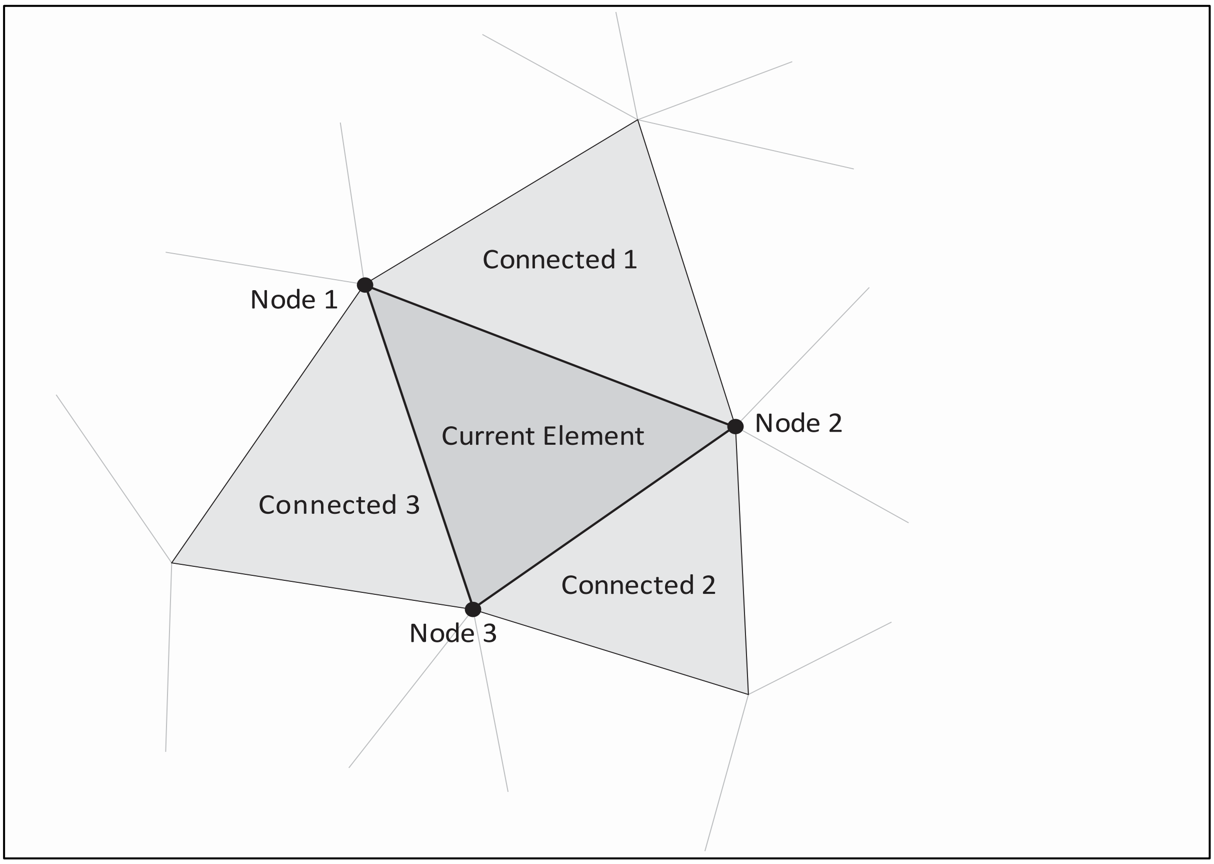


Figure 4-18: Connected elements.

The accumulation of recharge water is allowed as long as this does not oppose any danger to the mining operation. In the older mines, these ventilation walls were built with bricks and mortar, and they are so strong that they also act as water retaining walls. When the existence of water on a surface (TIN) is emulated, water in adjacent elements will have the same elevation as long as the *connected height* is lower than the water level. When a water-retaining wall is erected in real life, a similar wall must be created on the TIN.

WISH makes it is possible to mark the sides of an element as a non-flowing border (like the non-flow border in a groundwater modelling program). There are two ways to create a non-flowing border. The first is during the creation of the TIN. When selecting the polygon that acts as the outline of the TIN, any additional selected polylines inside the shape will force WISH to create nodes on these polylines. The nodes will automatically be flagged as no-flow, resulting in no-flow borders. No-flow border is on a TIN identifiable with a red line. See Figure 4-19.

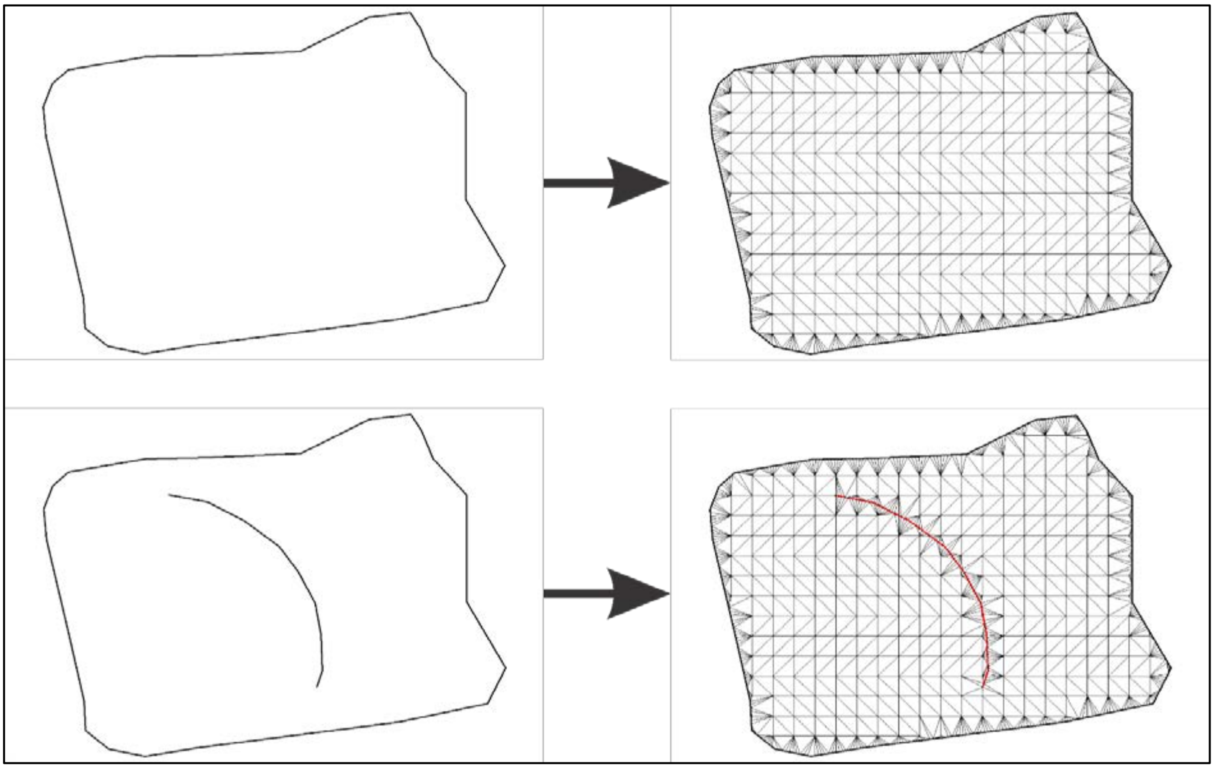


Figure 4-19: A TIN created without (top) and with (bottom) a no-flow border.

To create a no-flow border after generating the TIN the following steps need to be followed. Select the TIN and turn on “Show Nodes” and “Show Elements” (object properties). The nodes are indicated with circles and the elements as triangles.

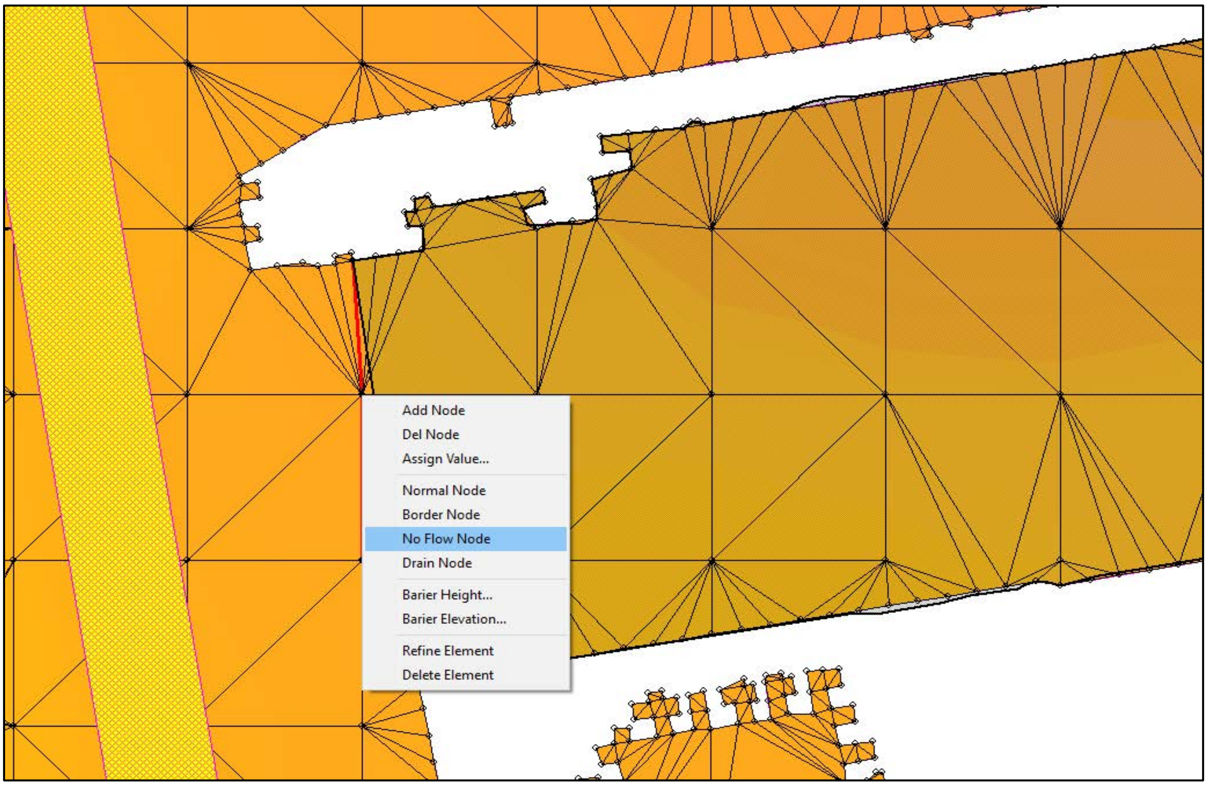


Figure 4-20: Setting node No-flow property.

With the TIN still selected, press the right-hand mouse button to show the context-sensitive menu and select "Edit Mode". Move the mouse over the first node that must be flagged and select "No Flow Node" from the context-sensitive menu. The node will now be displayed in a solid black colour. Move on to the next node and repeat. When two nodes of an element are marked as no-flow, the line or element border between the two nodes is marked as no-flow. When all no-flow borders are set, cancel the "Edit Mode" by deselecting the TIN.

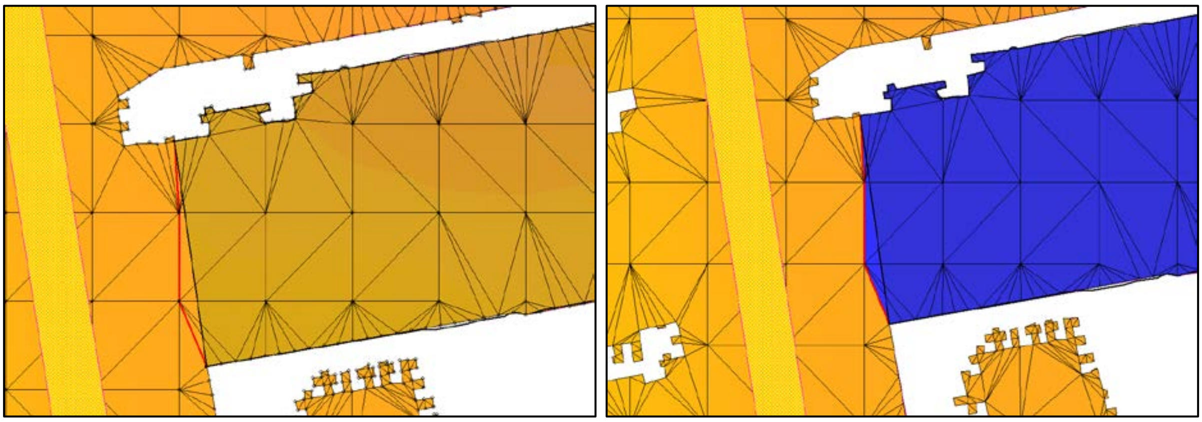


Figure 4-21: Detail of no-flow border to contain compartment water.

When a side of an element is marked as no-flow, the side is marked in red, and the neighbouring element is no longer considered connected. This feature allows the program to mimic walls inside a mine (ventilation walls, water retaining walls) inside the TIN (Figure 4-22).

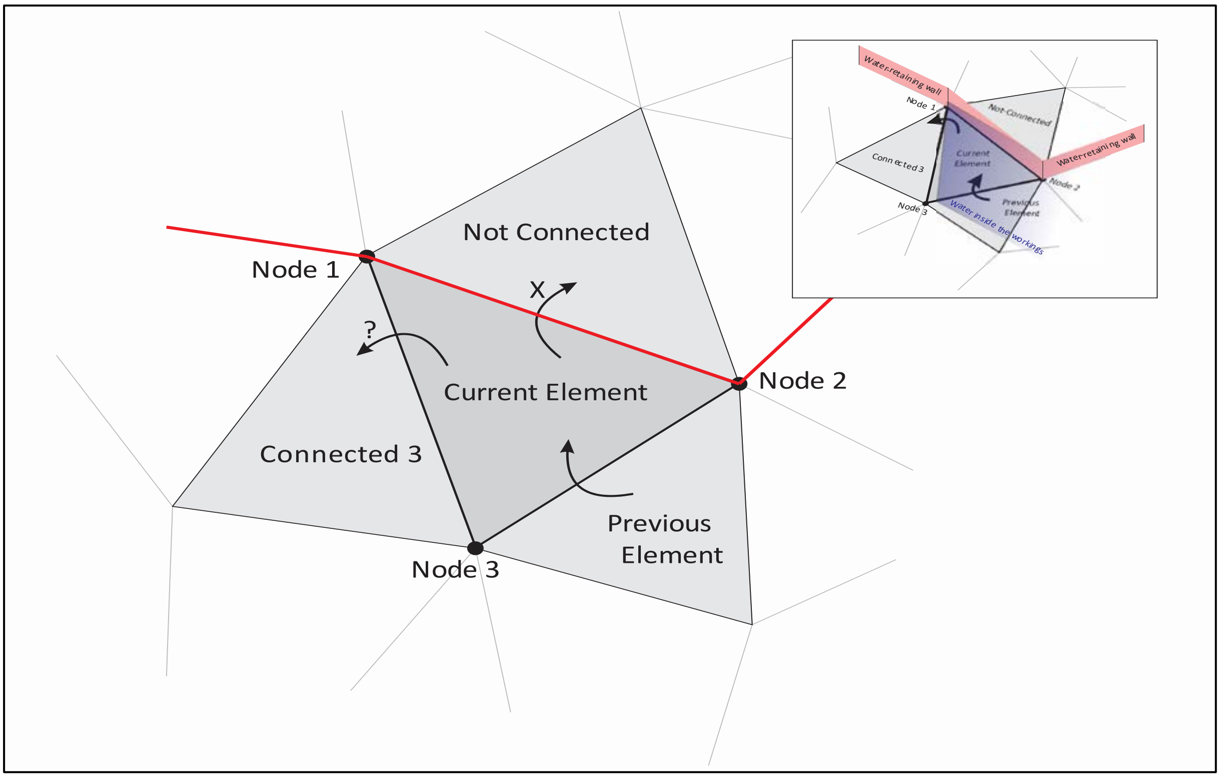


Figure 4-22: Connected elements with virtual barriers.

A fence is usually seen as an obstacle sealing any flow between elements, but the wall can also be created as a barrier. A barrier can either have a fixed overflow elevation or a constant height above the floor allowing water to flow only when the water table has reached that overflow elevation.

4.6.2 Build Node Node List – AnalyseNodeConnections()

When developing a computationally heavy program, programmers have the choice between re-calculating a value or storing the value in a variable. Storing the value in a variable will require extra memory, but it can save huge amounts of time, especially when the value is used many times. The node connections list is such a variable. A node connections list (*m_NodeNodeList*) is created for every node. This variable contains a list of all directly connected nodes. The list is created at start-up and does not need to be maintained as the connected nodes cannot change as long as the node/element configuration does not change. Figure 4-23 shows the current element (grey) with its three nodes, Node 1, 2 and 3 in orange, blue and green, respectively. All nodes directly connected to Node 1 are smaller and have an orange colour. All nodes directly connected to Node 2 are smaller and have a blue colour. Nodes connected to two of the element's nodes are shared and are indicated by dual colours. Using the schematic from Figure 4-23, the *m_NodeNodeLists for Node 1 contains: a,b,c,e,g,i and j; for Node 2 contains: f,c,d,h,k,l and j; and for Node 3 contains: g,l,n,m,i and f.*

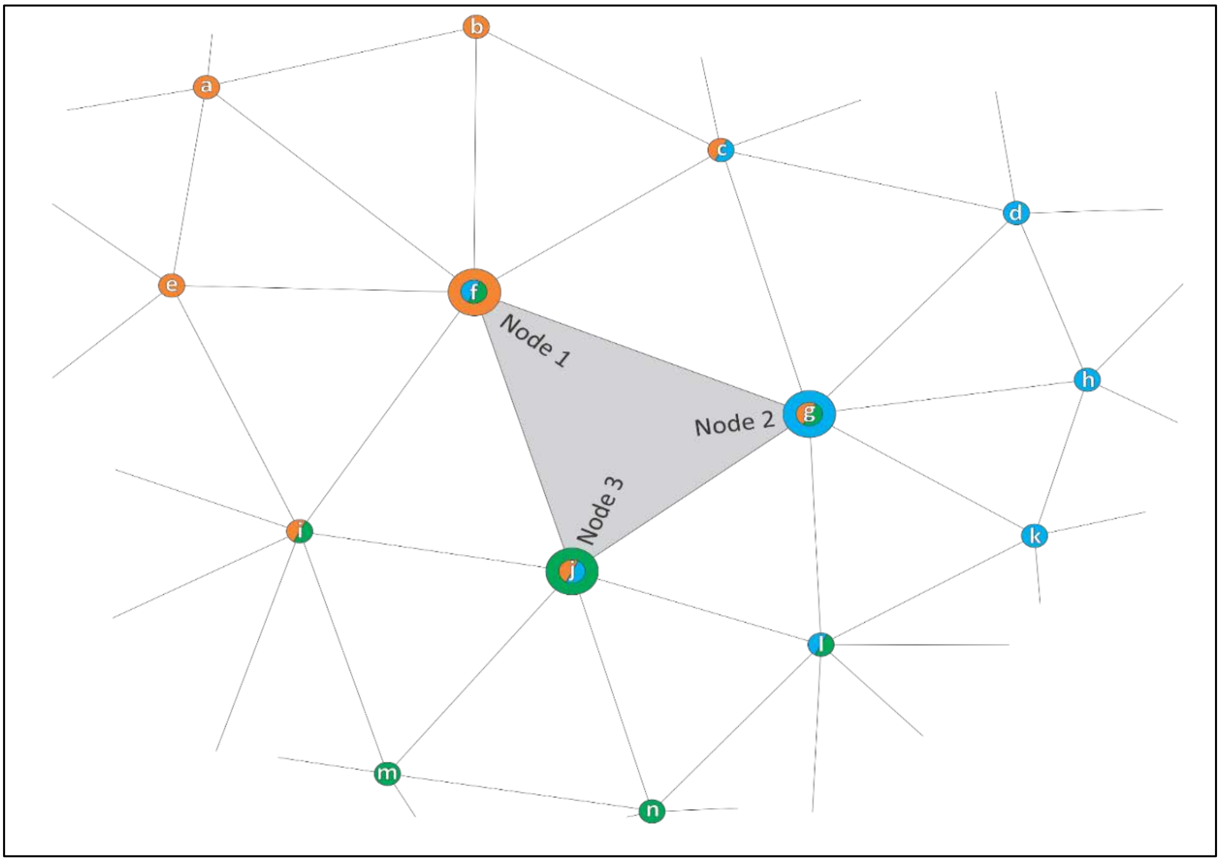


Figure 4-23: Connected nodes.

4.6.3 Build Node Element List – AnalyseNodeConnections()

An element list (*m_NodeElementList*) is created for every node. The list is created at start-up and does not need to be maintained as the list cannot change as long as the node/element configuration does not change. If a node is underwater, all connected elements are, at least partly, below the waterline. There is no specific order of the elements added to the Node_Element lists. Figure 4-24 Shows three nodes in orange, blue and green. The connected elements are displayed using the same colour.

"If debugging is the process of removing bugs, then programming must be the process of putting them in." – Edsger W. Dijkstra

---

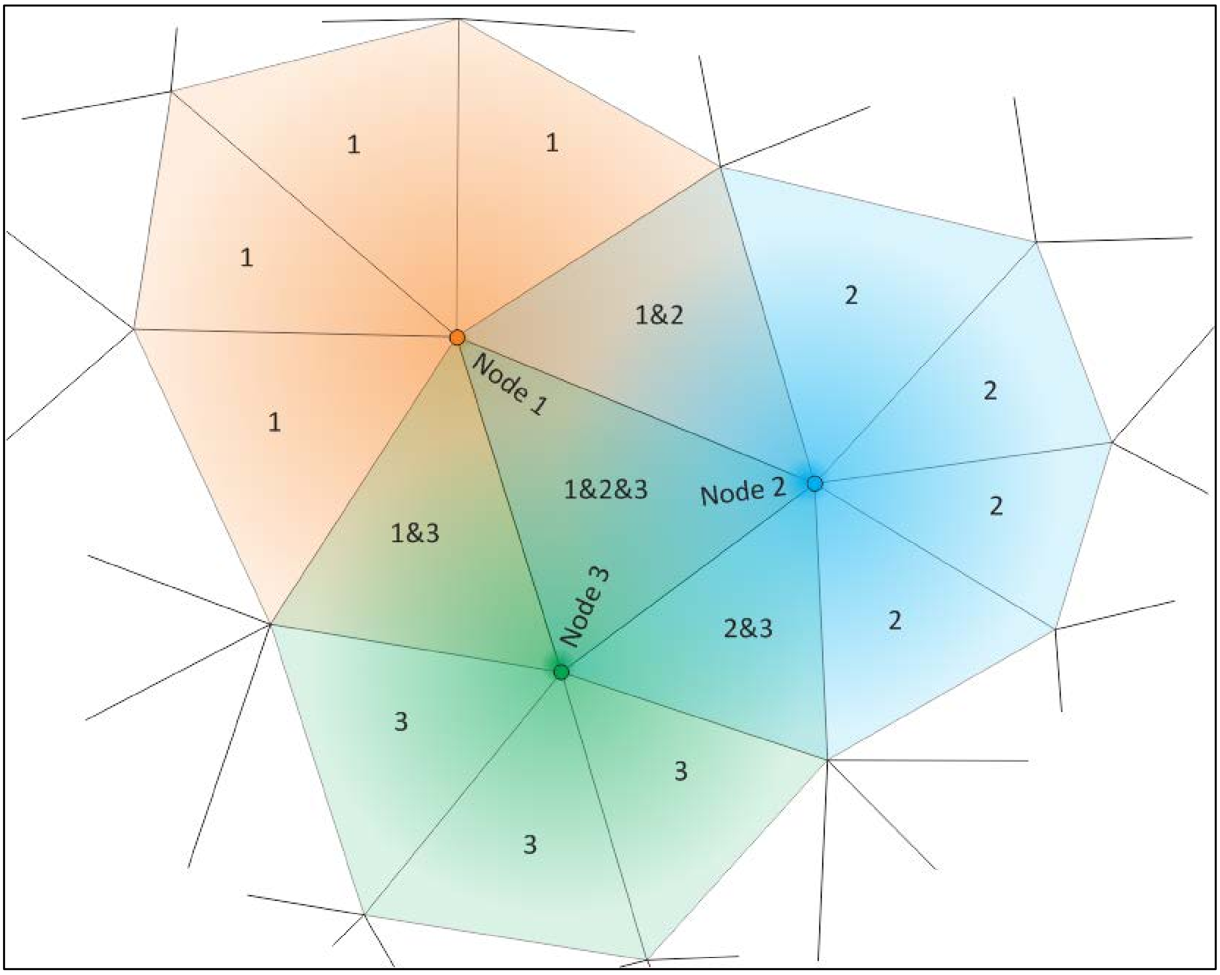


Figure 4-24: Connected Elements.

4.6.4 Recharge nodes – FindRechargeNodes()

The process to create the water bodies starts with the identification of the recharge nodes on the surface. The recharge nodes are those nodes where water will start to accumulate on a surface. Nodes connected to this node will all have higher surface elevations. Due to the undulating nature of a typical mine floor, many recharge nodes will be identified. Each node and element will have a recharge node assigned. In other words, the recharge node is the node where water would end up when it is allowed to flow freely. The recharge node is not necessarily the deepest or lowest node of the TIN, but it will be the lowest node in a depression.

As recharge may occur at any position on the surface, every element and node will have a recharge-node assigned. In the subroutine *FindRechargeNodes()* WISH will cycle through all the TIN's nodes and determine the recharge node for each one. The flow-chart in Figure 4-26 shows the above process graphically.

The process starts (**1**) with selecting the first node in the TIN's node list and making it the current node. Because this is part of a loop and all nodes will pass through here.

(**2**) The *current_node's recharge_node* is already assigned. (**3**) If this is the last node in the list, all nodes must have a *recharge_node* assigned, and the process is finished, if not, (**4**) the next node is selected and assigned to the *current_node*.

(**2**) If the *current_node*'s recharge node is not yet assigned (**5**) the *current_node* is stored in a list called the local_lower_list. The assumption is, that when a droplet of water is released on a node, it will roll-down using the steepest path to the next node.

(**6**) If there are nodes connected to the *current_node* that have a lower elevation (**7**) the node with the steepest downhill gradient is selected from the node's list of connected nodes (*m_NodeNodeList*).

(**8**) if the selected node's recharge node is not yet assigned, (**9**) the selected node becomes the current node, and the process jumps back to (5).

(**8**) if the selected node's recharge node is assigned, the deepest node is found. (**11**) The same recharge node must be assigned to the nodes stored in the local_node_list. The list must be emptied.

(**6**) If non of the nodes connected is lower than the *current_node*, the *current_node* is the lowest node and will also be the recharge node. (**11**) Tracking back the path the water droplet travelled all the nodes stored in the local_node_list *recharge_nodes* are assigned this lowest node. The list must be emptied.

After (11) the process is back at (3) the check whether the node is the last in the list.

The method enables WISH to show the streamlines/flow directions on a TIN (Figure 4-25).

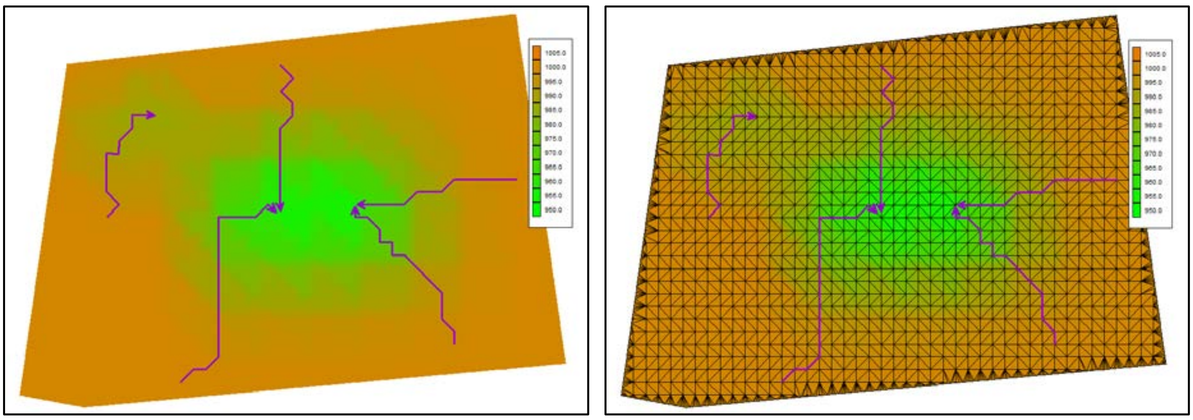


Figure 4-25: Streamlines / Flow directions on a TIN.

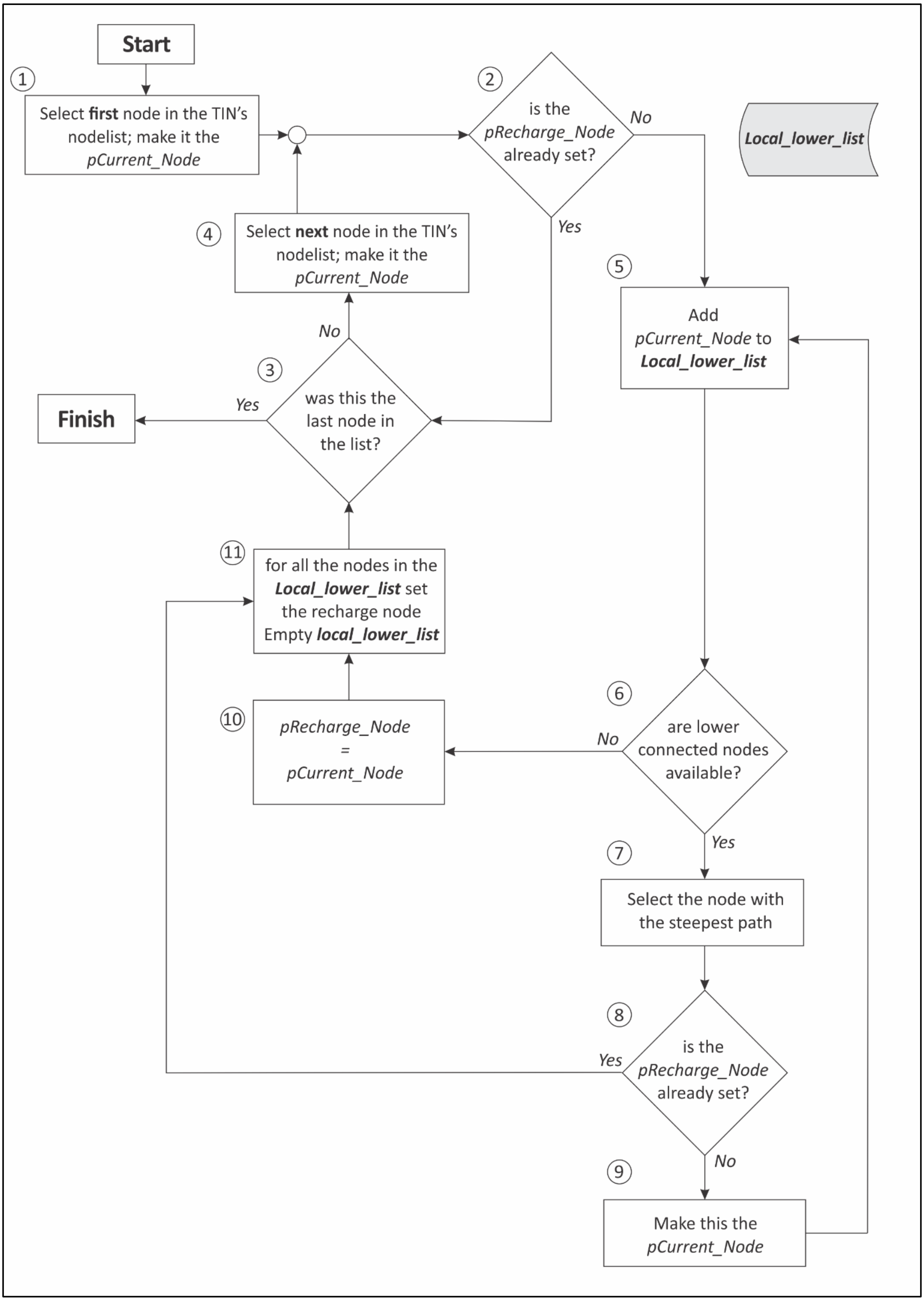


Figure 4-26: Flow chart to determine recharge nodes.

4.6.5 Create Water bodies – BuildWaterbodies()

A mine floor may have many depressions, some deep, some shallow. Every depression is a potential water body, and every water body needs to be described and must be accessible in WISH's water management routines. A new class, *CWMapTIN_Waterbody,* is developed to support potential and existing water bodies in WISH (Figure 4-29). This new class encapsulate all variables and functionality needed to calculate the state (position, elevation, volume and more) of the water bodies after recharge has occurred.

After identifying the recharge nodes, the water bodies are created one for each recharge node. The recharge nodes from the previous section are used as a starting point to create the water bodies. The water bodies are potential water bodies because they contain no water. These are waterless water bodies have a water level elevation that is equal to the elevation of the recharge node (the lowest node).

Without going through the whole class, there are a few variables and functions that need some explaining. There are four dedicated object lists of potential and flooded nodes and elements (*m_PotentialElements, m_Potential Nodes, m_WB_Elements and m_WB_Nodes*). The potential node- and element-lists are populated at the time the waterbody object is created. The two lists will not change unless the water body merges with another water body. The potential node list is sorted on the node's surface elevation from low to high. The flooded nodes- and elements lists keep track of those nodes and elements that are already under the waterline. These two lists are maintained and updated during recharge or flooding processes. Every water body has the following references:

Table 9: References declared in a waterbody.

Variable name	Variable class	Description
m_pPrevWB	CWMapTIN_WaterBody	Waterbody where water is received from
m_pNextWB	CWMapTIN_WaterBody	Waterbody where water is cascading into
m_pWB_RechargeNode	CWMapTIN_Node	Lowest node in the waterbody
m_pWB_HighestNode	CWMapTIN_Node	Highest node flooded by this waterbody
m_pWB_InflowNode	CWMapTIN_Node	The node where the water is received from
m_pWB_OverflowNode	CWMapTIN_Node	The node where cascading will start
m_pWB_NextNode2Flood	CWMapTIN_Node	Next node to be flooded in this waterbody

From Figure 4-27 the following is observed. Waterbody -**A**- receives water from an outside source (rainfall or pumping) therefore, the reference to the previous waterbody (*m_pPrevWB*) and the inflow node (*m_pWB_InflowNode*) is NULL (meaning not existing). The lowest node in the water body m_*pWB_RechargeNode* equals -**a**-. The waterbody is overflowing, therefore the overflow-node (*m_pWB_OverflowNode*) and the highest-node (*m_pWB_HighestNode*) are both set to -**b**-. Because waterbody -**A**- is overflowing into the second waterbody, NextWB is -**B**-.

Waterbody -**B**- receives water from waterbody -**A**- therefore, *m_pPrevWB* is -**A**-. The node where the water is flowing from A to B (*m_pWB_InflowNode*) is -**b**-. The lowest node in the water body m_*pWB_RechargeNode* equals -**c**-. The waterbody is overflowing; therefore the overflow-node (*m_pWB_OverflowNode*) and the highest-node (*m_pWB_HighestNode*) are both set to -**d**-. Because waterbody -**B**- is overflowing into the third waterbody, NextWB is -**C**-.

Waterbody -**C**- receives water from waterbody -**B**- therefore, *m_pPrevWB* is -**B**-. The node where the water is flowing from **B** to **C** (*m_pWB_InflowNode*) is -**d**-. The lowest node in the water body m_*pWB_RechargeNode* equals -**e**-. The waterbody is not overflowing the highest node underwater is -**f**-, the overflow node   And because waterbody -**C**- is not overflowing NextWB is NULL.

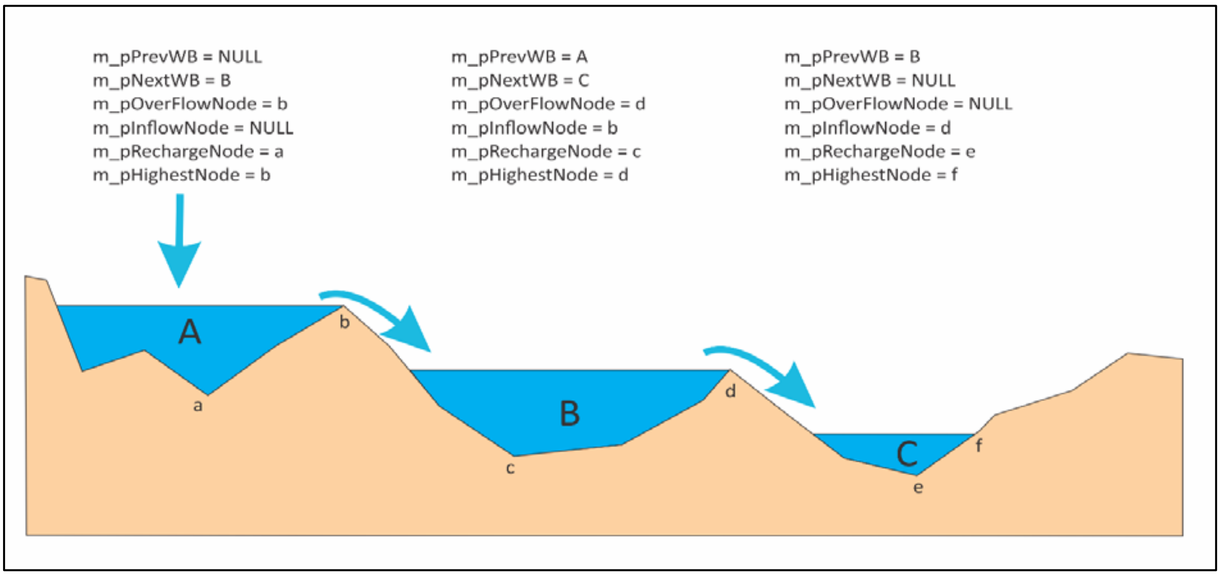


Figure 4-27: Cascading waterbodies.

To enable WISH to identify and work with water bodies, the CWMapTIN_WaterBody class was created to define the waterbody. Figure 4-28 shows the potential water bodies in a part of an underground mine. The water bodies distinguish themselves by randomly assigned colours.

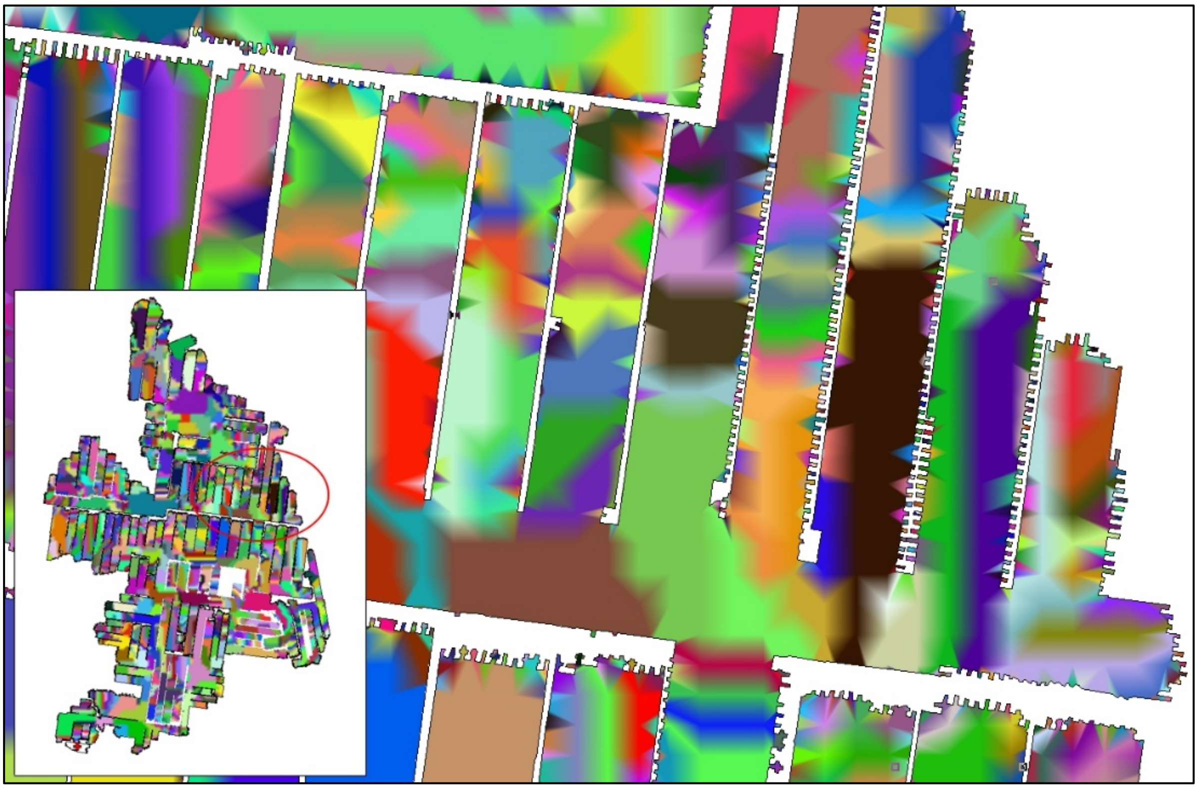


Figure 4-28: Potential water bodies on an underground floor (using random colours).

Figure 4-29 shows the class definition as it currently is in WISH. The class shows the constructor and deconstructor and some 28 variables and 34 functions.

When a waterbody is created, it will receive the status *DORMANT*; when the waterbody receives water the status changes to *FLOODING*; when the water level reaches the overflow node the status changes to *OVERFLOW* and lastly when the waterbody merges with another, the water body must be deleted, the waterbody gets the status *DELETED*. The waterbody will remember its current water level elevation and the area of the footprint and the total volume of water is updated when the water level changes.

A node that is part of a watershed is assigned to two water bodies, but elements can only be part of one water body.

When a node becomes flooded (the node's elevation is lower or equal than the waterbody's water level elevation) the node is added to the water body's node-list (m_WB_Nodes). At that point, all elements connected to the node are added to the flooded elements list (m_WB_Elements). A node and an element can be flooded by only one water body. When a second waterbody floods a node (or element) the two waterbodies must merge.

With the potential elements and nodes, it is possible to determine the highest water level before it starts to overflow into an adjacent waterbody.

```
class CWMapTIN_WaterBody : public CObject  // CWMapItem
{
public:
  CWMapTIN_WaterBody(CWMapTIN *pParent);
  virtual ~CWMapTIN_WaterBody();

  CWMapTIN           *m_pTIN;                 // The TIN this water body reside on
  CWMapTIN_Node      *m_pParentNode;          // The node used to create the waterbody
  int                 m_nInitialisationCode;  // Internal code
  int                 m_nWaterBodyNum;        // Number to identify and reference the waterbody
  int                 m_nRemVolNr_Old;        // Old remaining volume number
  unsigned int        m_nStatus;              // _WB_DORMANT = 0 | _WB_FLOODING = 1 | _WB_OVERFLOWING = 2 |
                                              // _WB_TO_BE_DELETED = 3
  unsigned int        m_nInList;              // number of times the waterbody is in the list.
  unsigned int        m_nTemp;                // Temporary number.
  double              m_dElevation;           // water level elevation
  double              m_dMaxElevation;        // water level elevation before overflowing.
  double              m_dVolume;              // water Volume
  double              m_dCapacity;            // Maximum volume before overflowing
  double              m_dArea;                // water Area
  double              m_dRechargeVolume;      // Volume of water still to be recharged to this waterbody
  BOOL                m_bFlooding;            // Flag to prevent multiple (recursive) flooding calls
  BOOL                m_bPotentialNodesMustBeSorted;
  COLORREF            m_ColorRef;             // to draw a colour map with waterbodies

  CObList             m_PotentialElements;    // Potential elements (elements not necessarily flooded)
  CObList             m_PotentialNodes;       // Potential nodes (nodes not necessarily flooded)
  CObList             m_WB_Elements;          // List with (partly) flooded elements
  CObList             m_WB_Nodes;             // List with flooded nodes

  CWMapTIN_WaterBody *m_pPrevWB;        // Waterbody source, overflowing from *m_pBody_Inflow into *this
  CWMapTIN_WaterBody *m_pNextWB;        // Waterbody target, overflowing fron *this into *m_pBody_Outflow

  CWMapTIN_Node      *m_pWB_RechargeNode;     // Lowest node in the waterbody
  CWMapTIN_Node      *m_pWB_HighestNode;      // Highest node flooded by this waterbody
  CWMapTIN_Node      *m_pWB_InflowNode;
  CWMapTIN_Node      *m_pWB_OverflowNode;     // lowest node from m_WB_Nodes that will start to overflow
  CWMapTIN_Node      *m_pWB_NextNode2Flood;   // Next node to flood for this waterbody

  BOOL                AddNode(CWMapTIN_Node *pNode);
  BOOL                FloodNode(CWMapTIN_Node *pNode);
  BOOL                RemoveNode(CWMapTIN_Node *pNode);
  BOOL                NodeIsFlooded(CWMapTIN_Node *pNode)  {return (m_dElevation >= pNode->m_dZSurface);}
  BOOL                AddElement(CWMapTIN_Element *pElem);
  BOOL                FloodElement(CWMapTIN_Element *pElem, BOOL bIncludeNodes);
  BOOL                RemoveElement(CWMapTIN_Element *pElem);
  BOOL                FindReceivingWaterBodies();
  BOOL                BuildNodeListFromElements();

  double              Elevation();
  double              Elevation(double dElevation);
  void                SortPotentialNodesOnElevation();
  double              CalculateCapacity();
  double              CalculateVolumeAtElevation(double Elevation);
  void                Calculate_Area_And_Volume();
  double              CalculateArea();
  double              Flood(double dVolumeToFlood);
  double              FloodToNode(CWMapTIN_Node *pNode, BOOL bReportMode);
  double              FloodToElevation(double dElevation, BOOL bReportMode);
  double              FloodToVolume(double dVolumeToFlood);
  void                FloodToCapacity();

  int                 ExtendWaterBody(double dElevation);
  int                 ExtendWaterBody(CWMapTIN_Node *pNode);
  int                 MergeWaterBody(CWMapTIN_WaterBody *pWB, double dInc);

  CWMapTIN_Node      *FindLowestNodeAboveWB();
  CWMapTIN_Node      *NextNode2Flood();
  CWMapTIN_Node      *FindOverflowNode();
  CWMapTIN_Node      *FindLowestNode();
  CWMapTIN_Node      *FindHighestNode(BOOL bFloodedOnly);
  CWMapTIN_Node      *FindHighestConnectedNode();
  CWMapTIN_Element   *GetNextElementWithRemainingWater(CWMapTIN_Element *pElem);

  CWMapTIN_WaterBody *WaterBodyToMergeWith();              // Check whether this waterbody needs to merge
  CWMapTIN_WaterBody *FindReceivingWaterBody();
  CWMapTIN_WaterBody &operator=( CWMapTIN_WaterBody &WB ); // Assign all variables to another waterbody
};
```

Figure 4-29: Definition of class CWMapTIN_Waterbody (from the header file WMapTIN.h).

Water bodies have a built-in capability to merge with another waterbody thereby updating the potential- and flooded nodes and elements.

A waterbody can be flooded with specified volume water, or to a given elevation.

When the water level of a waterbody rises to the elevation where it may spill over into another waterbody, pointers are set indicating the receiving waterbody. When the flooding continues and the water level of the receiving waterbody reaches the water level elevation of the cascading waterbody the water bodies are merged.

Every element will receive a volume of water depending on the area of the triangle, the recharge factor specified, and the amount of rain. That volume of water is redirected to the recharge node. All the recharge node all the redirected volumes are added and the total volume of water is then added to the waterbody.

For every water body it is known the volume of water currently in the waterbody and the total volume the waterbody can store before it cascades into an adjacent waterbody. The waterbody takes the extraction factor of each element into account while calculating volumes.

4.6.6 Verify Waterbodies – VerifyWaterBodies()

Working with many water bodies and many variables per waterbody, it is time-consuming to verify references each time they are used. The function *VerifyWaterBodies()* is created to verify all references for all water bodies before the flooding process starts. Three checks are performed:

- All potential elements must have the correct recharge element.
- All elements must belong to a water body and one only.
- Each element can only be listed once.

4.6.7 Assignment of Recharge Factor and Extraction Factor

Generally, due to a lack of information (or effort), we do not have separate recharge rates for the different areas on an open cast mine; instead, one average recharge rate is used for the whole mined area. One exception is the final void which is easily identifiable and can be traced on a map using detailed maps or aerial photography.

Underground mines will often have different recharge factors for different areas as the recharge is dependent on the type and the depth of mining.

It is possible to assign different recharge rates by identifying and delineating the different areas. Recharge values can be assigned by selecting these areas and using the Set TIN properties option. Precipitation is assumed to be distributed equally over to whole mining area. Figure 4-30 shows a selected polygon superimposed on a surface and the dialogue window where the *Recharge factor* is selected, and a value of *0.20* is entered. Upon clicking the OK button, all elements within the polygon will be assigned a 20% recharge factor.

Assignment of the extraction factor is performed similarly.

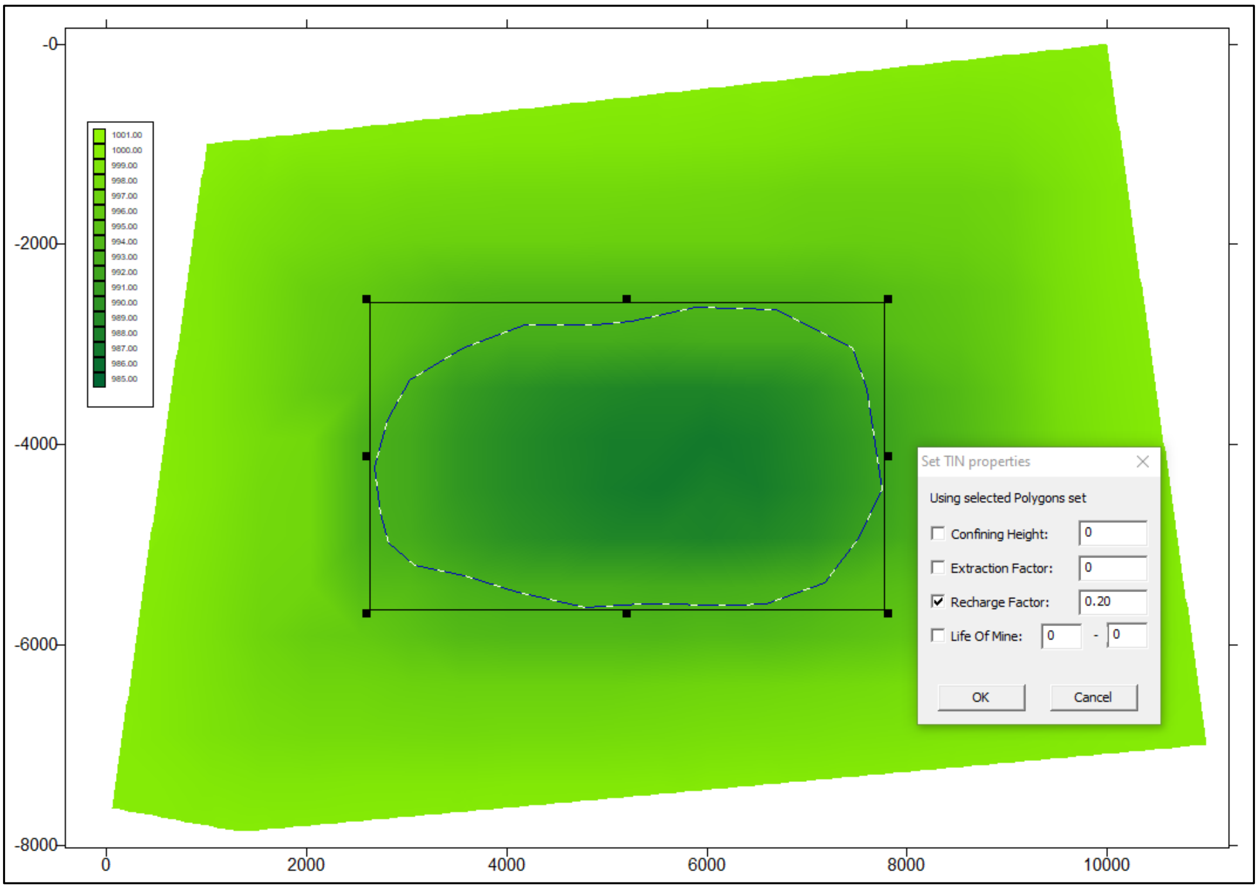


Figure 4-30: Assigning a recharge factor.

After setting the parameters, the values can be checked by hovering the mouse over the TIN. The parameter values, as well as other useful information, is displayed in the status bar (Figure 4-31). The details that can be inspected are:

- WL Water level elevation;
- Cnt Contour values (surface elevation);
- Vol Volume of water body + the volume of the current element;
- WB Water body number [# elements (# potential elements) # nodes (# potential nodes);
- WCP Associated Water Control Point (0 if the water body was created by recharge);
- ExtrF Extraction factor;
- ExtrH Extraction height;
- RF Recharge factor;
- Area Total area of the water body.

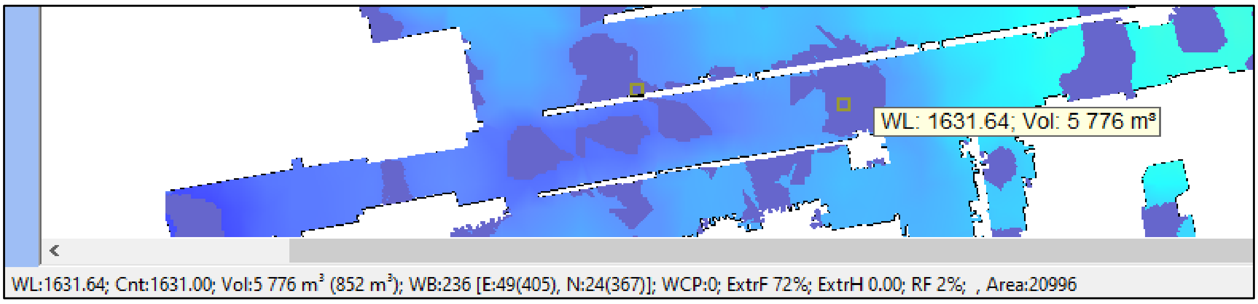


Figure 4-31: The WISH status bar with valuable information.

4.6.8 Next Node To Flood – FindLowestNodeAboveWB()

When "pouring" water in a depressions the water level rises from node to node and all elevations in-between. A list with potential nodes is created and sorted on node elevation for each waterbody before the flooding takes place to minimise the computational effort. The list contains the lowest node up to the first node where the waterbody may spill over into another waterbody. It is because of the sorting that the water body "knows" which nodes are flooded and also what node would be next in line to be flooded when water is added to the waterbody.

```
CWMapTIN_Node *CWMapTIN_WaterBody::FindLowestNodeAboveWB()
{
  CWMapTIN_Node *pLowestNode = NULL;

  POSITION pos = m_PotentialNodes.GetHeadPosition();  // list is sorted low -> high
  while (pos)
  {
    CWMapTIN_Node *pNode = (CWMapTIN_Node *) m_PotentialNodes.GetNext(pos);
    if (pNode->m_dZSurface > m_dElevation)
    {
      pLowestNode = pNode;
      break;
    }
  }
  return pLowestNode;
}
```

Figure 4-32: Function CWMapTIN_WaterBody::FindLowestNodeAboveWB() from CWMapTIN.cpp.

4.6.9 The Capacity of a Waterbody – CalculateCapacity()

The capacity of a waterbody is defined as the total volume of water in a waterbody before it decants or overflows. The function uses the elevation of the overflow node and passes it on to *CalculateVolumeAtElevation()* which in turn returns the volume. To know the maximum amount of water is a potential time-saver. When a waterbody is flooded by recharge water, the recharging volume of water is known. If that volume of water is more than the capacity minus the current volume of the waterbody, the waterbody may be flooded to the maximum elevation without following a stepped approach.

4.6.10 The Capacity of a Waterbody at an Elevation – CalculateVolumeAtElevation()

This function calculates the volume of water in a waterbody at a given water level elevation. The method is the same for opencast or underground mines. The volume of water above the floor TIN is calculated for each element separately, after which all the volumes are totalled. With an underground mine, the process is repeated to calculate the volume of water above the roof or ceiling, after which the "roof volume" is subtracted from the "floor volume".

4.6.11 Flooding to a Node – FloodToNode()

When WACCMAN floods a waterbody, it will do it from node to node, which means that it will determine the waterbody's lowest node above the water level and subsequently flood to that elevation. Each water body has two node-lists, a list of flooded nodes and a list of potential nodes. The list with potential nodes is sorted on the surface elevation of the nodes.

Using this sorted list, the waterbody can look-up which node is the next node to flood. Because it is possible that the node flooded is a potential overflow node, the list of waterbody-nodes and potential-nodes is updated every time a node is flooded. After each flooding iteration, the volume of water is also re-calculated using *CalculateVolumeAtElevation()*.

4.6.12 Flooding to Elevation – FloodToElevation()

A waterbody will always be flooded from one node to the next node. When the volume of floodwater is less than the volume of water needed to flood the next node, the final water level is between nodes. To determine the exact level WACCMAN will use the bisection method (also called the interval halving method) in conjunction with the *CalculateVolumeAtElevation()* to find the elevation where all the flood water is used and nothing more (Figure 4-33).

```
double CWMapTIN_WaterBody::FloodToElevation(double dElevation, BOOL bDeltaVolume)
{
  // bDeltaVolume = False -> Return total volume of waterbody after water level rise
  // bDeltaVolume = True  -> Returns volume added to waterbody due to water level rise

  double dVolume = m_dVolume;

  Elevation(dElevation);  // set waterlevel elevation and calculate m_dArea & m_dVolume

  dVolume = (bDeltaVolume) ? (m_dVolume – dVolume) : m_dVolume;

  return dVolume;
}
```

Figure 4-33: Function CWMapTIN_WaterBody:: FloodToElevation() from CWMapTIN.cpp.

4.6.13 Flooding to Volume – FloodToVolume()

This function floods the waterbody to a target volume. During the flooding-process WISH determines the next node above the water level (*m_pWB_NextNode2Flood*). The node's elevation is considered the target elevation. Subsequently, it calculates the volume of water inside the waterbody if the water level is raised to the target elevation. If the volume is still smaller than the target volume, the flooding takes place and a new node to flood is determined. When the volume is larger than the target volume WISH will use the bisection method (also called the interval halving method) to determine the new water level elevation to which the water body must be flooded (Figure 4-34).

```
  if (dNewVolume < dTargetVolume)
  {
    // there is not enough space to store or the water - so we fill the waterbody to the max
    FloodToCapacity();
    FindReceivingWaterBodies();
  }
  else
  {
    double dDiff = 0;
    double dHi = m_pWB_NextNode2Flood->m_dElevation;
    double dLo = m_dCurrentElevation;
    double dElevation = (dHi + dLo)/2.0;
    do
    {
      dDiff = dTargetVolume - CalculateVolumeAtElevation(dElevation);
      if (dDiff > 0)
      {
        // dTargetVolume > calculated volume  ->  Waterlevel elevation must go up
        dLo = dElevation;
        dElevation = (dLo + dHi)/2.0;
      }
      if (dDiff < 0)
      {
        // dTargetVolume < calculated volume  ->  Waterlevel elevation must go down
        dHi = dElevation;
        dElevation = (dLo + dHi)/2.0;
        dDiff *= (-1);
      }

      if ((dHi - dLo) < 0.001) // less then 1 mm
      {
        dElevation = (dHi + dLo)/2.0;
        dDiff = 0;
      }
    } while (dDiff > 0.001);  // 1 Litre

    FloodToElevation(dElevation, true);
  }
```

Figure 4-34: Part of Function CWMapTIN_WaterBody:: FloodToVolume(double dVolume), determining the water level elevation to flood to.

4.6.14 Cascading into another waterbody

Flooding takes place from node to node. As mentioned earlier, WISH floods a water body from node to node. After reaching a node, WISH will use the node's list of connected nodes to determine whether the waterbody may start to decant into an adjacent water body. It will iterate through the list looking for the node not part of the current water body and with the steepest descent. After identifying this node, the waterbody cascading into is also known. The source water body's overflow node is set to this node as is the inflow node of the receiving water body (Figure 4-35).

```
if (dVolumeStillToFlood > 0.001)  // more than 1 litre
{
  double dL, dH, deltaHopL = 0;

  CWMapTIN_WaterBody *pWB = NULL;
  CWMapTIN_Element *pElem = NULL;

  POSITION pos = m_pWB_OverflowNode->m_NodeElementList.GetHeadPosition();
  while (pos)
  {
    pElem=(CWMapTIN_Element*)m_pWB_OverflowNode->m_NodeElementList.GetNext(pos);
    if (pElem->m_pWaterBody && (pElem->m_pWaterBody != this))
    {
      dL = sqrt((m_pWB_OverflowNode->m_dX-pElem->m_dCx)*(m_pWB_OverflowNode->m_dX –
                pElem->m_dCx)+(m_pWB_OverflowNode->m_dY-pElem->m_dCy) *
               (m_pWB_OverflowNode->m_dY - pElem->m_dCy));
      dH = m_pWB_OverflowNode->m_dZSurface - pElem->m_dzOrg;

      if ((dH/dL) > deltaHopL)
      {
        deltaHopL = dH/dL;
        pWB = pElem->m_pWaterBody;
      }
    }
  }
}
```

Figure 4-35: Part of Function CWMapTIN_WaterBody:: Flood(double dVolume), determining the next waterbody cascading into.

The ***m_pWB_RechargeNode*** is always the lowest node in the waterbody. When two waterbodies merge, the recharge node will become the lowest from the two bodies. ***m_pWB_HighestNode*** is the highest flooded node. The ***m_pWB_OverflowNode*** is the lowest node that will allow the waterbody to overflow into another. It is also this node's surface elevation that is used to calculate the waterbody's capacity. ***m_pNextWB*** is a reference to the waterbody receiving water when the water capacity is reached, and the waterbody is overflowing. Walking backwards, every receiving water body has a ***m_pPrevWB*** referencing the water body it from which it receives water and ***m_pWB_InflowNode*** *is set to be the same as the m_pWB_OverflowNode* is the from which water is entering the waterbody.

The next four figures show a sequence of what will happen during a flood event. The sequence starts with two water bodies, Waterbody 1 with a water level elevation 1101.50 mamsl and Waterbody 2 with a water level elevation of 1100 mamsl. Waterbody 1 consists of seven elements (triangles) when water is added to this water body, it will start to overflow when it reaches 1102 mamsl (this is the elevation of the lowest node above the waterline). For both waterbodies the recharge nodes are indicated, these are the lowest node in each waterbody. Waterbody 1 receives water. The highest node flooded is the recharge node.

In the second figure, the water level has increased to the elevation of the overflow node (1102). At that moment, the ***m_pWB_OverflowNode*** of Waterbody 1 becomes the ***m_pWB_InflowNode*** for Waterbody 2. Any water added to Waterbody 1 will decant in Waterbody 2 (leaving the water level elevation in Waterbody 1 constant. For Waterbody 1 the highest node flooded is now the overflow node. For Waterbody 2 it is the node at 1100.

While adding water to Waterbody 1, the level of Waterbody 2 will keep rising. This will continue until there is no water left to add or until the water level of Waterbody 2 reaches the ***pWB_InflowNode,*** and the two water bodies merge. In the fourth figure, the two water bodies are replaced by one (merged) waterbody. A new recharge node is calculated as well as a new overflow node and the capacity of the merged waterbody.

4.6.15 Lowest connected water body – FindLowestConnectedWB()

Before any flooding can start, we need to know which elements form part which water body and how the water bodies are connected. The connectivity builder, *BuildConnectivity()*, (Section 4.6.1) updates every element in the TIN with references to the connected elements. A list of water bodies is created by *BuildWaterBodies()* (Section 4.6.5), and each water body has lists of elements and nodes.

The function *FindLowestConnectedWB()* is called from a water body, and it determines the lowest node on the waterbody border and the waterbody to which it connects. The function starts by iterating through the elements-list checking for each element the three connected elements, and the waterbodies they belong to. If found that the element reports to a waterbody different from the one the function is called from the function will look for the node with the lowest elevation shared between the two water bodies. This process is repeated for all elements. The lowest node is assigned to the waterbody's overflow node (*m_pWB_OverflowNode),* and the waterbody is recorded as the next waterbody, or the waterbody it will decant into.

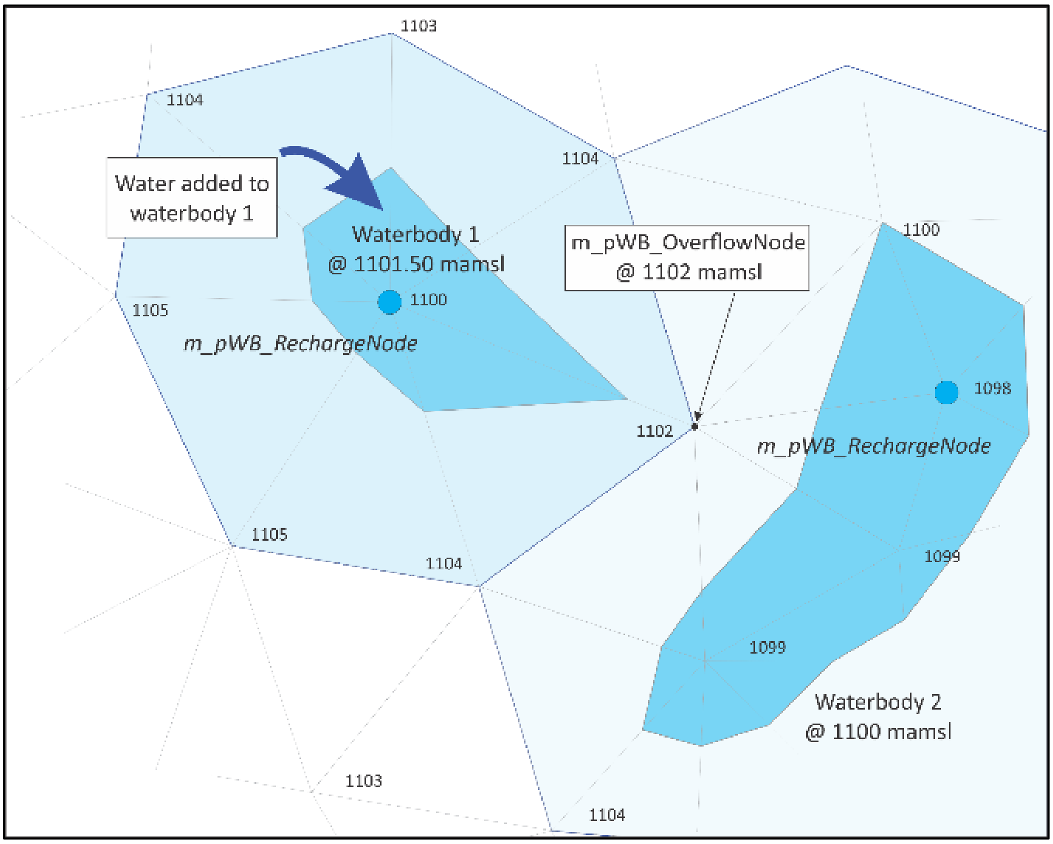


Figure 4-36: Flooding sequence 1 of 4.

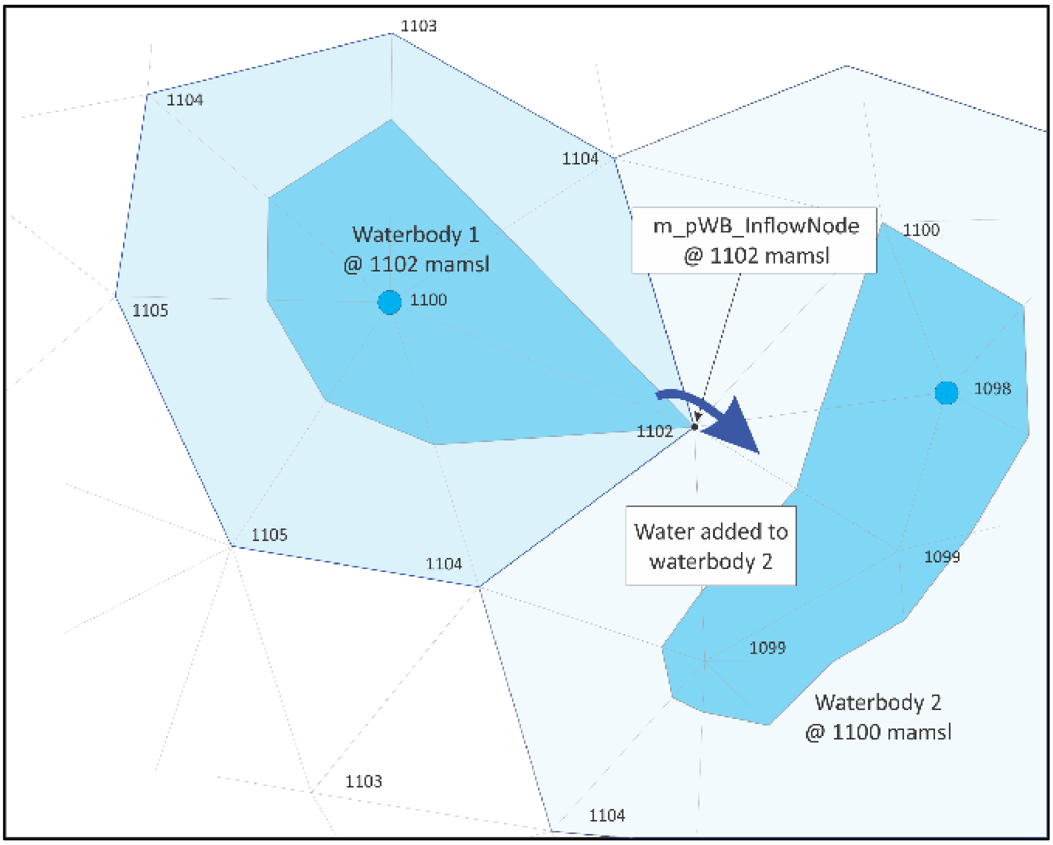


Figure 4-37: Flooding sequence 2 of 4.

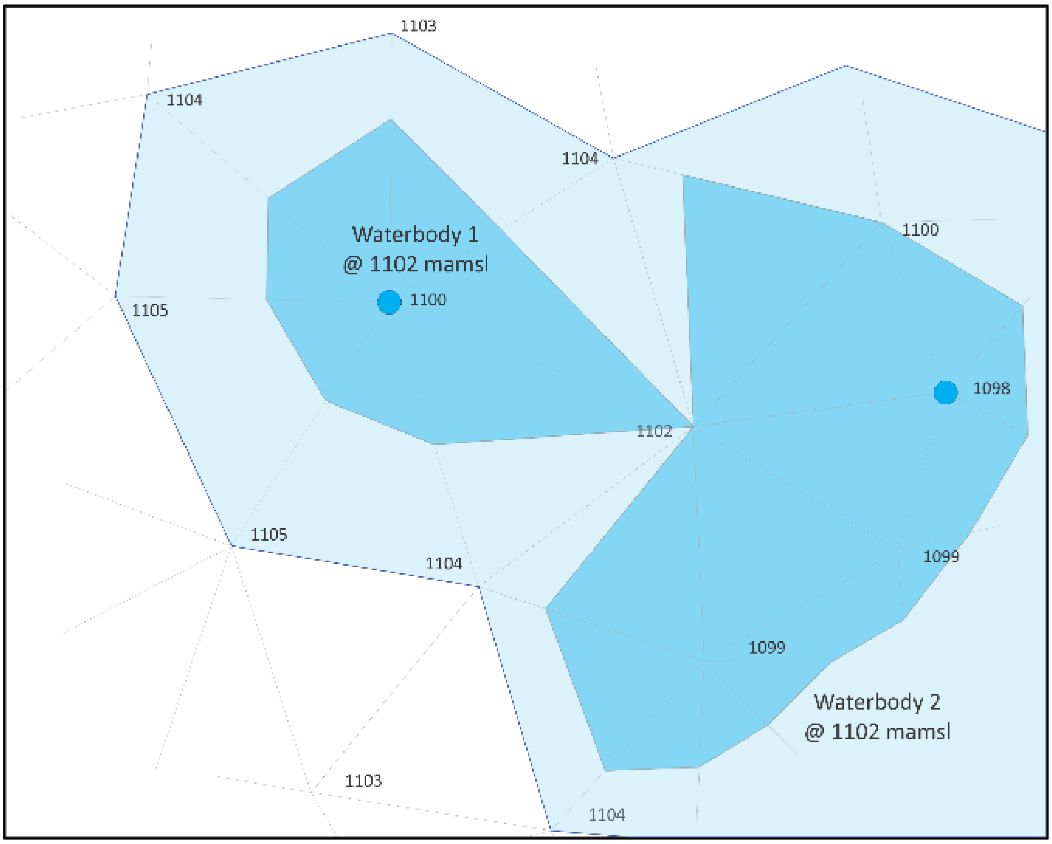


Figure 4-38: Flooding Sequence 3 of 4.

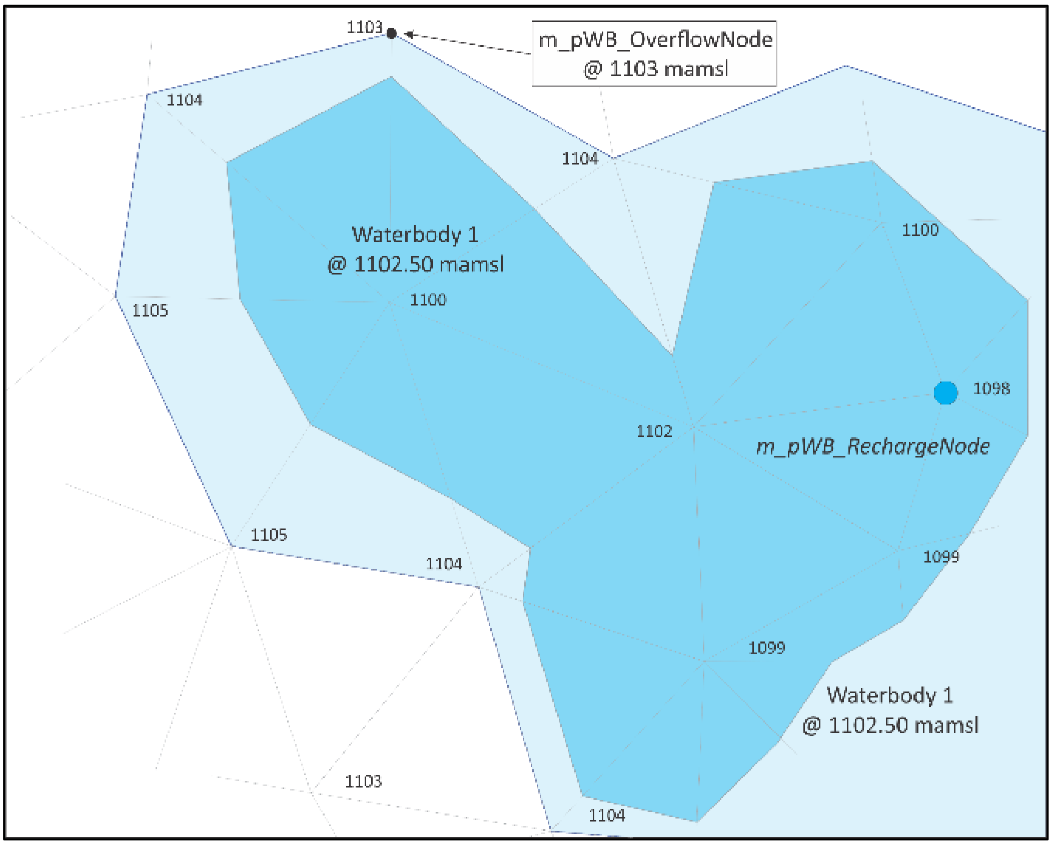


Figure 4-39: Flooding Sequence 4 of 4.

On the side of the connected waterbody, the inflow-node (*m_pWB_InflowNode)* and the previous waterbody need to be assigned, but because it is possible that a previous call to this function (initiated from a different waterbody) found the same connected waterbody the surface elevation of the inflow name must be compared to the one already recorded and the inflow-node and previous water must only be updated when the elevation is lower.

4.6.16 Merging Waterbodies – MergeWaterBody()

When a waterbody is cascading, the receiving water body's water level will rise. When the receiving water body reaches the same water level elevation as the cascading or decanting one, then and only then the two water bodies need to be merged. During the merger, the variables of the two water bodies need to be consolidated, including the joining of the element- and node lists. The call to the *MergeWaterBody()* function is initiated from the receiving waterbody. The status of the waterbody will remain _WB_FLOODING, meaning that the merged water body still receives water. The decanting waterbody is now obsolete but cannot be removed at this point as this will result in an unstable list. Instead, the status is changed the _WB_DELETED. When the program is iterating through the list of water bodies, deletion of this waterbody may result in WISH trying to access a deleted waterbody which in turn will result in a program failure. The object will be deleted at an appropriate time.

```
m_nStatus                 = _WB_FLOODING;
m_pNextWB                 = NULL;
m_pPrevWB                 = pWB->m_pPrevWB;
m_dRechargeVolume        += pWB->m_dRechargeVolume;
Elevation(m_dElevation);  // waterlevels inside the WB are set to m_dElevation
m_pWB_HighestNode         = FindHighestNode(true);  // check only the flooded nodes
m_pWB_RechargeNode        = FindLowestNode();
m_pWB_NextNode2Flood      = FindLowestNodeAboveWB();
m_pWB_OverflowNode        = FindOverflowNode();
CalculateCapacity();      // Calculate new capacity
```

Figure 4-40: Part of the source code of the merging function.

4.6.17 Removing references made to waterbodies – RemoveWaterBodyReference()

Nodes, elements and water bodies have variables containing references (also called pointers) to water bodies. When a waterbody is deleted, the pointer becomes invalid. Accessing an invalid pointer will result in a program crash. The function *RemoveWaterBodyReference*(CWMapTIN_WaterBody *pWaterBody) will scan all references and remove any invalid pointer.

4.6.18 Deleting a water body – RemoveWaterBodies()

After two water bodies (A and B) merge, the first waterbody remains as the merged waterbody, The status flag will remain _WB_FLOODING, the second body's status flag is changed to _WB_DELETED. The waterbody still exists, but can no longer be used or referenced.

It is, in most instances, not possible to delete an unused water body while iterating through the water bodies list without rendering the position pointers invalid and ultimately crashing the WISH application. *RemoveWaterBodies()* is called to delete all obsolete water bodies.

```
void CWMapTIN::RemoveWaterbodies(int nStatus)
{
  CObList TempList;

  while (m_WaterBodiesList.GetCount())
  {
    CWMapTIN_WaterBody *p = (CWMapTIN_WaterBody *) m_WaterBodiesList.RemoveHead();
    if (p->m_nStatus != nStatus)
      TempList.AddTail(p);
    else
    {
      RemoveWaterBodyReference(p);
      delete p;
    }
  }

  while (TempList.GetCount())
  {
    CWMapTIN_WaterBody *p = (CWMapTIN_WaterBody *) TempList.RemoveHead();
    m_WaterBodiesList.AddTail(p);
  }
}
```

Figure 4-41: The function CWMapTIN_WaterBody::RemoveWaterbodies(int nStatus), uses a temporary object list to store valid water bodies while those with the specified nStatus are deleted.

---

*"If we wish to count lines of code, we should not regard them as **lines produced** but as **lines spent**." — Edsger W. Dijkstra*

4.6.19 Recharge – RechargeByRainfall()

The RechargeByRainfall() function calculates the volume of water that recharges and distributes it over the surface by simulating a water movement to the lower parts of the surface, filling the depressions and merging the puddles of water. The process, in more detail, is described using Figure 4-42. The emphasised (bold) number between parenthesis is a reference to which part is described. A similar number but not emphasised is a reference to the next statement. (**1**) The process is started by calling FindRechargeNodes() and BuildWaterBodies() to initialize the lowest points on the surface and their associated water bodies. The recharge volumes are calculated for each element and are added to the water bodies. Each water body that has a volume2Flood, it's status flag is set to WB_FLOODING.

(**2**) Iterating through the list of water bodies and starting with the first water body. (**3**) A test is performed by checking the water bodies status flag whether the body is either overflowing or deleted. If this is true, (**4**) the next waterbody is selected from the list, and the process jumps back to (3); If the test at (**3**) yielded false, the waterbody is valid and has capacity left. (**5**) The lowest node that will allow water to decant from the water body is determined. This node is called the overflow-node. The capacity of the waterbody is calculated.

(**6**) If the capacity is larger than the volume water recharged, (**7**) all recharge water is assigned to the waterbody, the water level elevation of the water body is calculated and the volume of water still to flood is set to zero.

(**6**) If the capacity is smaller than the volume water recharged, (**8**) the water level is raised to the level of the overflow node, and the volume still to flood is adjusted by subtracting the quantity that was added to the water body.

(**9**) If another water body is decanting in the current water body, it is in the code referred to as the previous water body (WB*prev)*. If the inflow node is the same as the calculated overflow node from (5), the two water bodies need to merge. (**11**) The merging process starts with combining all potential and flooded nodes and elements, determining a new recharge (lowest) node, calculating the area, flooded volume and capacity of the WB*prev* and the current waterbody after which the status flag of the previous water body is changed to WB_DELETED. This flag value indicates that the water-body object can no longer be used. The water body will be removed from the water bodies list and deleted after the flooding process is completed. If the previous waterbody received water from a decanting water body, the present water body's previous-water-body is changed to the decanting waterbody's previous water body.

(**9**) If the Overflow node and the inflow node are not the same, (**10**) the waterbody will start to decant, and the status flag is changed to WB_OVERFLOW. The receiving water body (WB*next*) is determined, and the volume that is still left for flooding is transferred and added to the amount that will flood WB*next*. A pointer is set that the receiving water body knows the decanting water body (WB*next* ->WB*prev*).

When finished with (7), (10) or (11), the total volume left to inundate the surface is checked, if there is more than one cubic metre left, a quick check has to reveal if the water body is the last one in the list. If not, the next water body is selected if it is the last one in the list, the program jumps back to the first water body of the list is selected, in both cases, the next step is to return to (3). If the total volume left to inundate is less than one cubic metre, the process is ended.

The process of finding the lowest connected waterbody the flooding and consequently, the merging of the water bodies are explained using eleven pictures of an area that must be flooded. The first picture (Figure 4-43) displays the area with 18 potential waterbodies or catchments numbered from A to R. The blue dots indicate the lowest point in each catchment. The dots on the catchment borders indicate the lowest nodes between the two basins. When both catchments fill up, the water bodies will merge at the lowest nodes. The node has the surface elevation indicated as well as the name of the node. The name is derived from the two catchments, e.g. the node separating waterbody **B** and waterbody **F** is called **bf**. Most of the lowest points have a red colour, but three are in green. When allowing the catchments to fill with water, each one to the lowest catchment border node, the green nodes are the ones where the water bodies will merge. The recharge process will start by creating a list with all the catchments. The list begins with waterbody A and ends with R. The recharge function will iterate through the list and calculate the amount of water that is collected by each catchment.

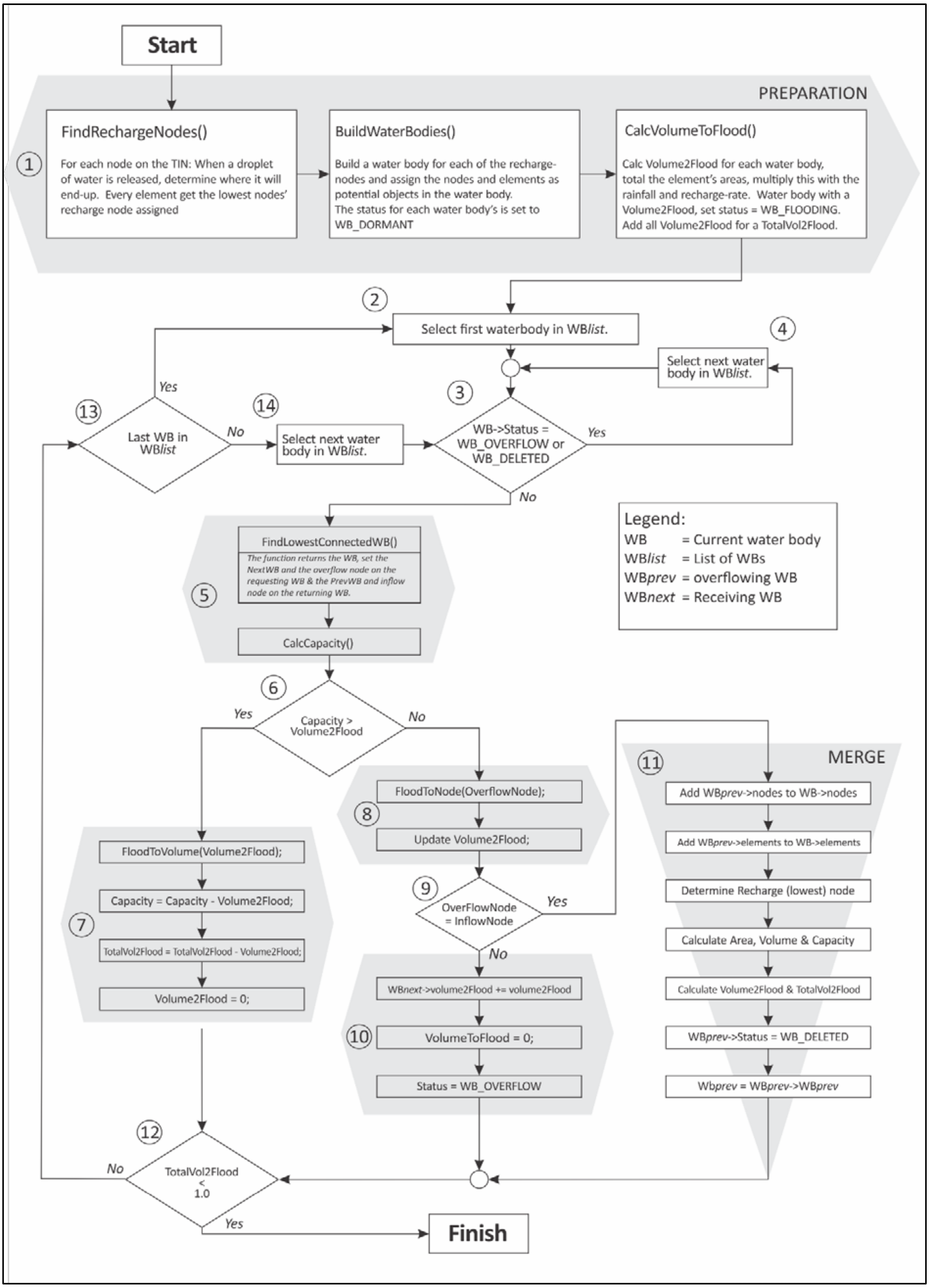


Figure 4-42: Flow diagram for CWMapTIN_Waterbody::RechargeByRainfall(double dRainfall).

The amounts are multiplied with the recharge factor and assigned to the waterbody as water that needs to be recharged. For the sake of simplicity, let us assume that there is ample recharge water.

With an unlimited amount of water available, the process can flood each catchment to the level where it will try to overflow into an adjacent waterbody. To determine that elevation *FindLowestConnectedWB()* is called for each waterbody. The function will identify the waterbody the calling waterbody will decant into as well as the position and elevation at which this will happen. Using the example in Figure 4-43 to explain the process catchment **-C-** and **-D-** will have the same water level elevation and are about to be joined at node **cd**. The same will occur with **-M-** and **-P-** at node **mp** and with **-O-** and **-R-** at node **or**. Because the process works sequentially through the waterbodies list, when waterbody **-C-** is reached the function *FindLowestConnectedWB()* will identify waterbody **-D-** as the lowest connected waterbody and assign **-D-** to the next waterbody, the waterbody into which this one (**-C-**) will decant. The function also identifies node **cd** as the overflow node. One waterbody further in the list is waterbody **-D-**. The function *FindLowestConnectedWB()* will locate **-C-** as the lowest connection at the same overflow node **cd**. The recharge function can only initiate *MergeWaterBody()* when two water bodies are found with the same overflow node (and the water levels in the catchments must be at the elevation of the overflow node). It is not enough to know that **-C-** is about to decant in **-D-**. It must also be clear that **-D-** is about to decant into **-C-**. After the first cycle (iteration through the list) three pairs of water bodies are merged, and three water bodies are deleted. Cycle 2 starts with the iteration from the top of the list. (Figure 4-44) Water level elevations are raised in the merged waterbodies to the maximum level before decanting and waterbodies **-D-** and **-H-** are found to decant into each other (as are -N- and -P-, and -Q- and -R-). Table 10 shows the nine cycles that are needed to merge all waterbodies.

Table 10: Merging waterbodies.

WB	START	1	2	3	4	5	6	7	8	9	END	
A	ab	ab	ab	ah	ah							I
B	bc	bd	bh									H
C	cd										Merging with	D
D	cd	dh										H
E	de	de	eh	eh								H
F	af	af	af	af	af	fh	fh					H
G	cg	dg	gh	gh	gi	gh						I
H	ch	dh	bh	eh	ah	gh	fh & hi					I
I	ei	di	hi	hi	ai	hi	hi	ir				R
J	jq	jq	jr	jr								R
K	kl	kl	kl	ik	ik	ik	ik	ik	kr			R
L	lo	lr	lr									R
M	mp											P
N	np	np										P
O	or											R
P	mp	np	pr	pr	pr	pr	pr	pr	pr	pr		R
Q	qr	qr										R
R	or	qr	lr	jr	gr	gr	hr	ir	kr	pr		

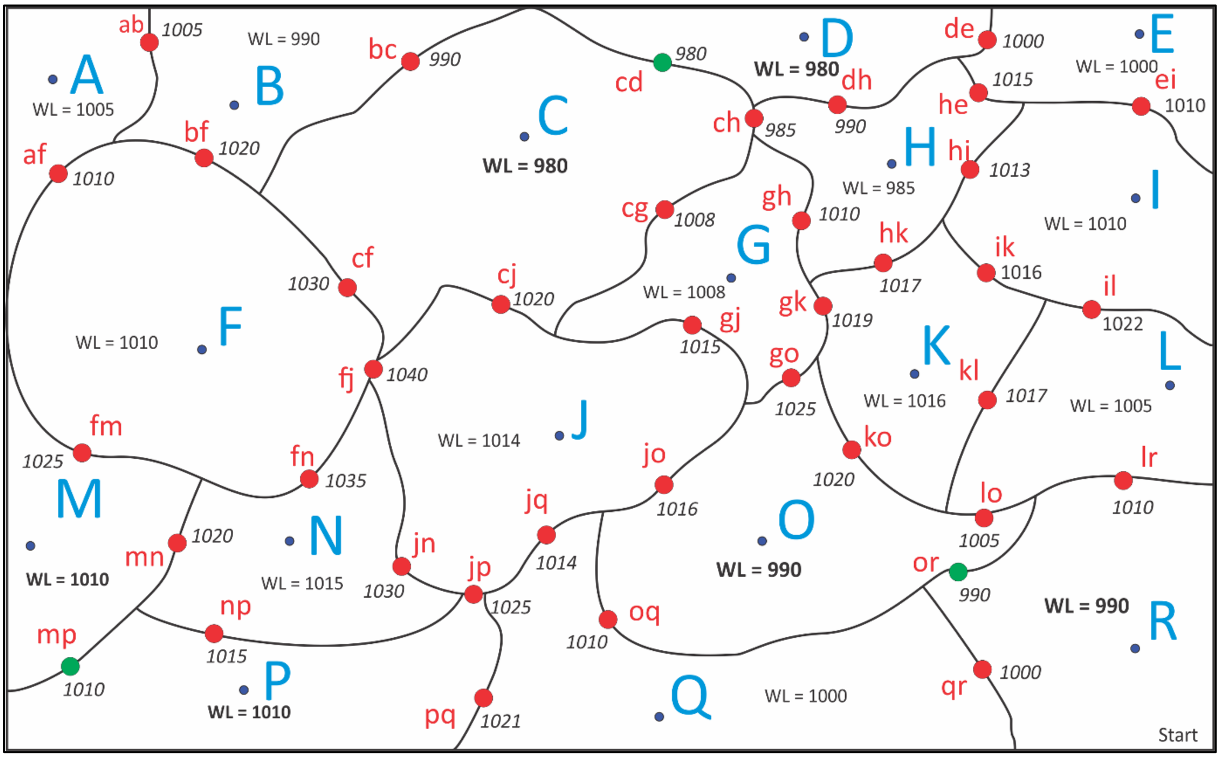


Figure 4-43: Area with 18 catchments.

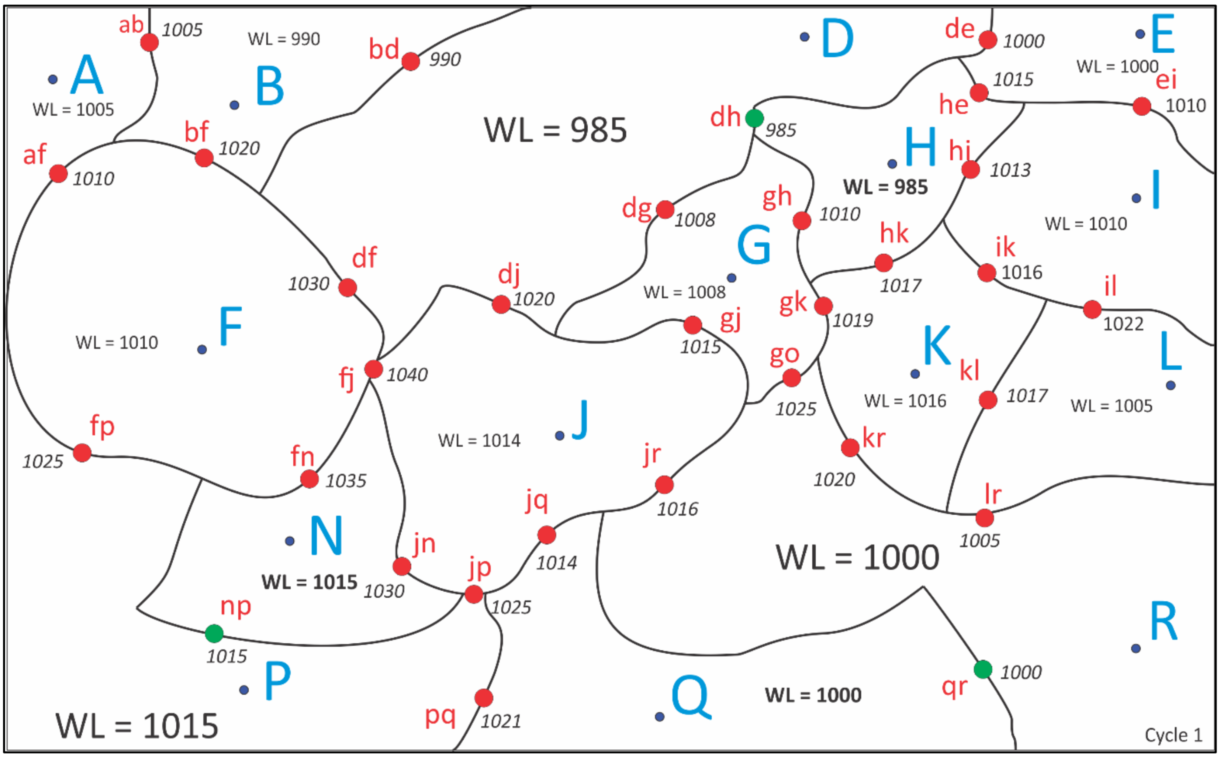


Figure 4-44: Area with 15 catchments.

Figure 4-45 shows the remaining eight cycles and the final map where all waterbodies are merged.

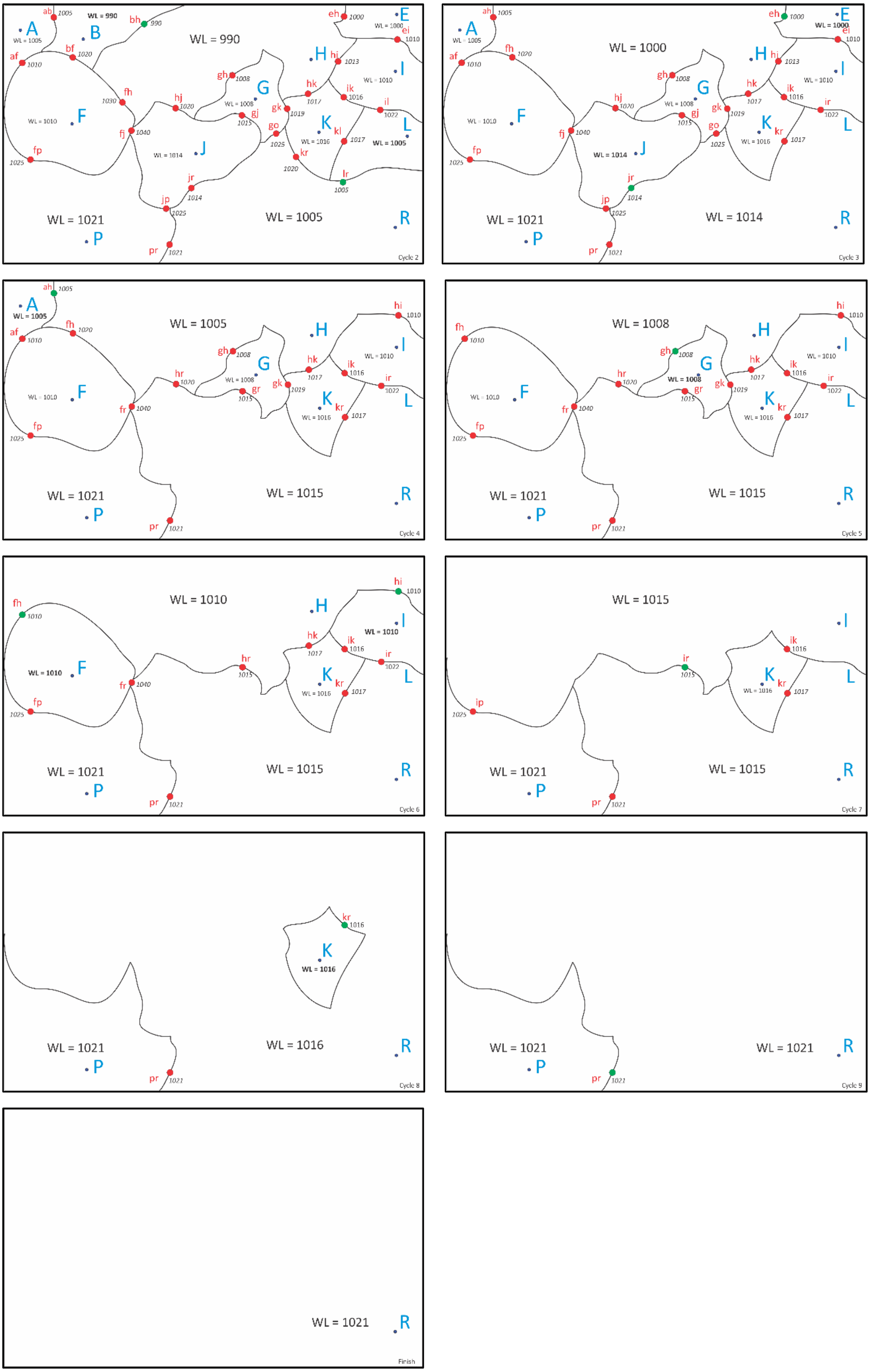


Figure 4-45: Merging waterbodies.

4.6.20 Flooding an underground with one large volume of water

This method calculates which parts of the surface will be inundated when a known volume of water is released at a location. In theory, the method is the same as the one used during the recharge-flooding. The procedure is, in essence, a little bit easier to understand because there is only one water body that is receiving water.

Using Figure 4-46 as reference. The function starts by (1) initialising the analysing all connections between nodes and elements, Identifying the recharge nodes and building of the water bodies. (2) Using the known location where the volume of water is released onto the surface (TIN), the receiving node and element are determined. A simple check is performed to make sure that the starting element is indeed valid. If the TIN did not cover the known location, a NULL-element would have been returned. With the starting element WISH also has a reference to the water body where the water is released onto the TIN. The water body is set to be the current water body.

The flooding routine will now call *FloodCalamity()*. FloodCalamity function starts whether the previous and next waterbodies are set. If the water body receives decant-water (*m_pPrevWB* is set) and it is also decanting (*m_pNextWB* is set) and both references points to the same waterbody (the current waterbody receives and spills water from and to the same waterbody) than it's time to merge the current waterbody with the previous (or next) waterbody. If the merging was successful, all references to the pPrevWB are removed, and the *m_pPrevWB* is deleted.

Next, the volume_to_flood is updated, and the lowest connected waterbody is determined. When the lowest connected waterbody is known, so will be the overflow node and its member the surface elevation, allowing the capacity to be calculated.

If the capacity is larger then the volume of water to be flooded, the volume of water is placed in the water body, and the flooding process is complete. If the capacity is smaller than the volume of water, the current waterbody is replaced by *m_pNnextWB*, and the function *FloodCalamity()* is called from within the *FloodCalmity()* function.

"To iterate is human, to recurse divine." – L. Peter Deutsch

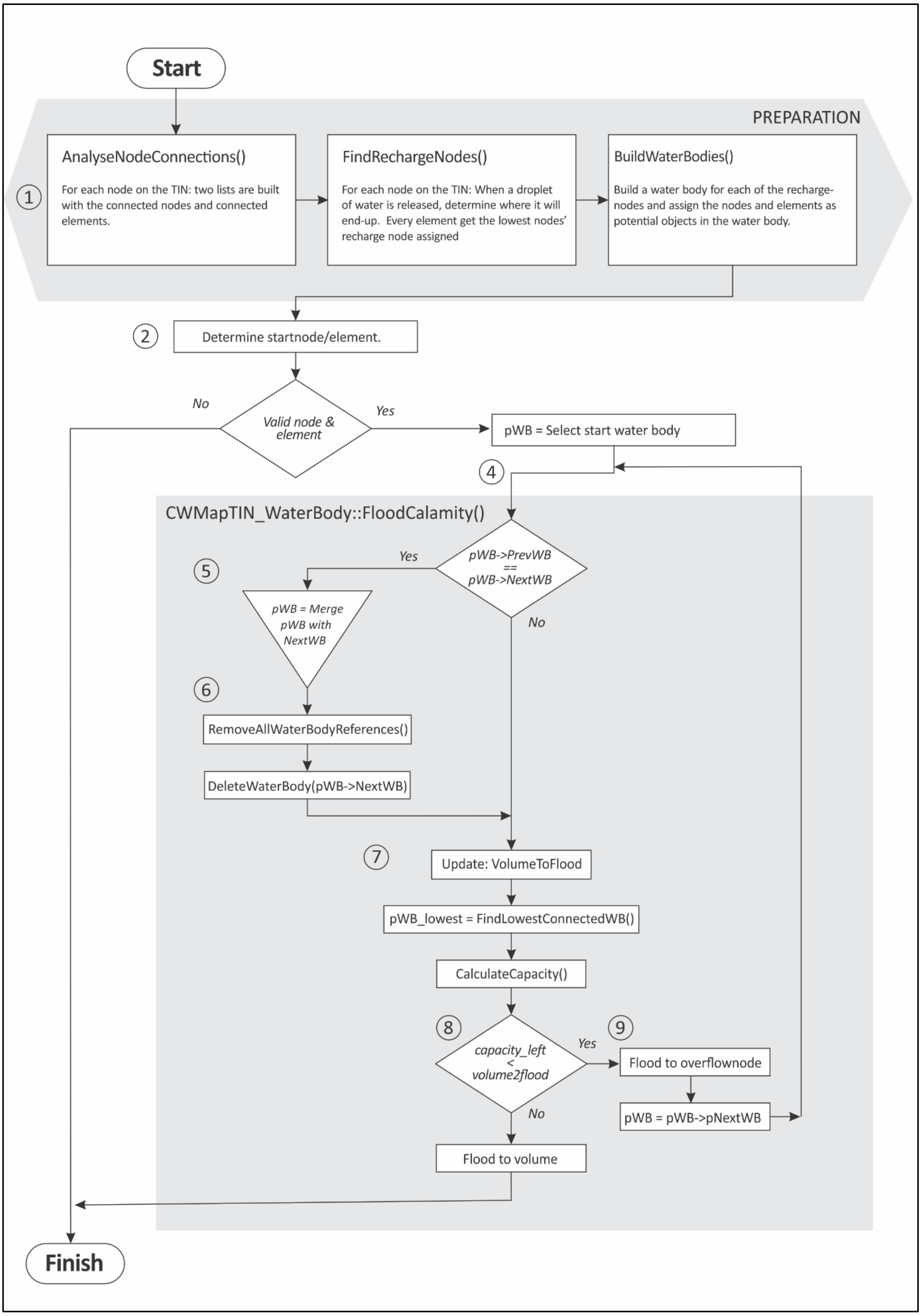


Figure 4-46: Flow diagram for CWMapTIN_Waterbody::FloodTIN(double x, double y, double volume).

```
double CWMapTIN_WaterBody::FloodCalamity(double dTotalVolume)
{
  if (m_pPrevWB && (m_pPrevWB == m_pNextWB) && m_pWB_InflowNode &&
       (m_pWB_InflowNode == m_pWB_OverflowNode))
  {
    CWMapTIN_WaterBody *pWaterbody = m_pNextWB;  // backup to the pointer
    if (1 == MergeWaterBody(m_pNextWB, _MAX_MERGE_WB_DIFF))
    {
      m_pTIN->RemoveWaterBodyReference(pWaterbody);
      m_pTIN->DeleteWaterBody(pWaterbody);
    }
  }

  double dVolumeToFlood = dTotalVolume - m_pTIN->TotalWaterInWaterBodies();

  if (dVolumeToFlood <= 0)
    return 0;

  if (FindLowestConnectedWB())
  {
    if (m_pWB_OverflowNode)
    {
      CalculateCapacity();
      if ((m_dCapacity-m_dVolume) < dVolumeToFlood)
      {
        FloodToNode(m_pWB_OverflowNode, true);
        m_pNextWB->FloodCalamity(dTotalVolume);
      }
      else
      {
        FloodToVolume(dVolumeToFlood);
        dVolumeToFlood = 0;
      }
    }
  }
  return 1;
}
```

Figure 4-47: Part of the source code of the FloodCalamity() function.

5 Model verification.

Two simple models were developed to test the recharge and flooding scenarios before applying the calculations to a full-size model.

5.1 Test site with one depression.

The first model is a simple polygon with a total size of almost 7000 ha (69718309.84 m² calculated by WISH). The polygon was the basis for creating a TIN with 670 nodes and 949 elements (nodal distance 500 m). In the first model, I created only one depression. The surface elevation was set to 1000 mamsl and a single depression of 13 meters created. The surface elevations ranged from 987 to 1000 m. A stage-volume curve shows that the maximum volume of water that can be stored in the depression 263400 Ml. At that time the water level will be at 1000 m (Figure 5-2).

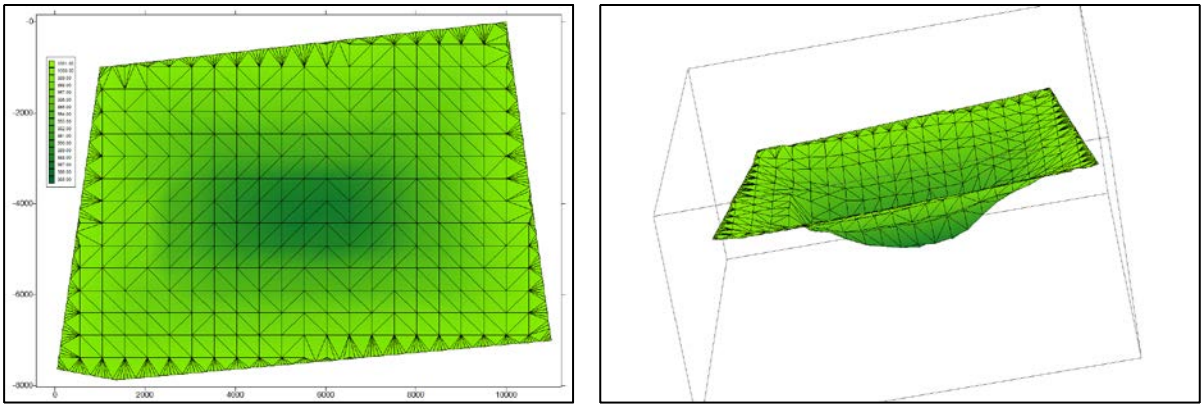


Figure 5-1: Single depression test TIN with the elements visible.

Since a recharge percentage was not assigned, the default recharge rate of 100% is applied. This means that every drop of rainfall intercepted by the surface will count towards the flooding of the depressing. The emulation starts with a surface analysis that determines a recharge-node for each triangle of the TIN. This process is described in the previous chapter. Next, WISH calculates the volume of water falling on each triangle, and this is transferred and added to the recharge volume of the recharge-node. The recharge node is the lowest in a (potential) water body. The following step is to flood the water body with the total volume of water assigned to the recharge-node.

"Program testing can be used to show the presence of bugs, but never to show their absence!" – Edsger W. Dijkstra.

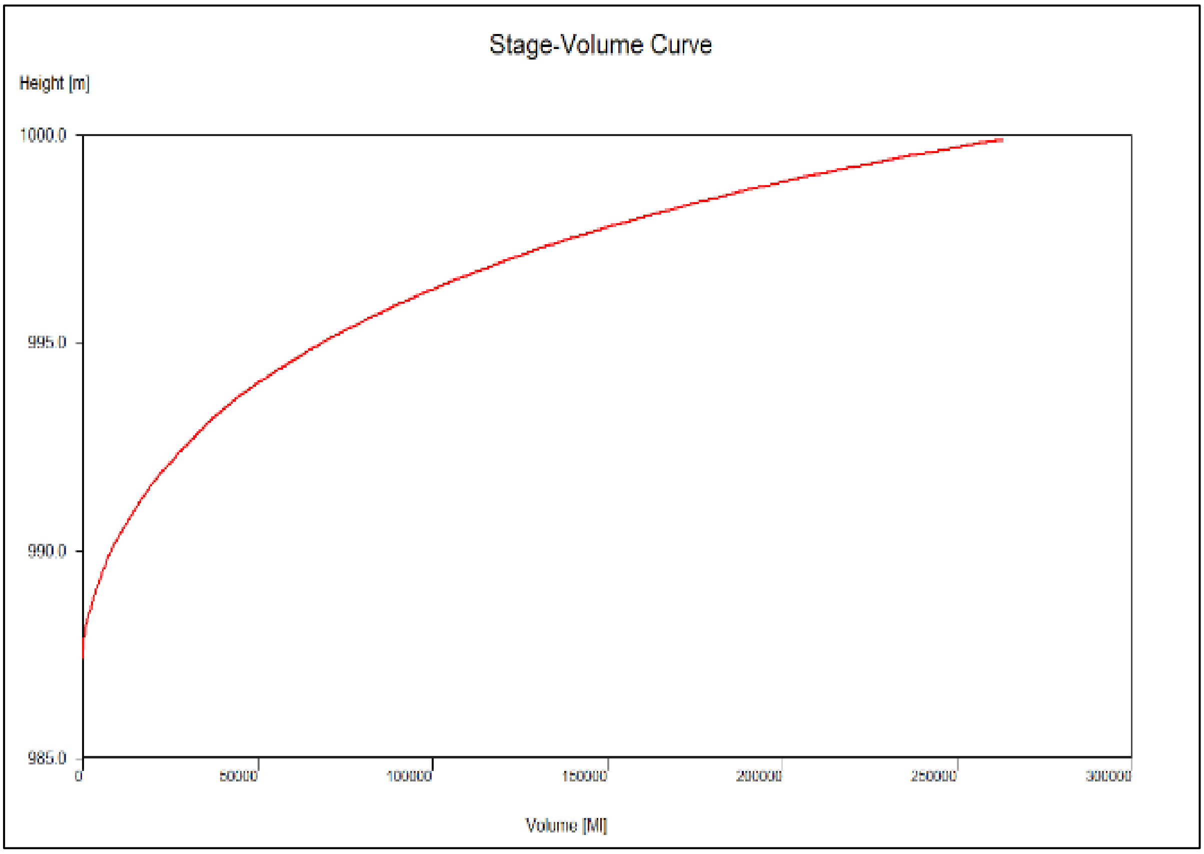


Figure 5-2: Stage-Volume curve for water stored on the surface.

Because this TIN has only one depression, it also has one (potential) water body (Figure 5-3) and thus one recharge node. All the rainfall intercepted by the entire TIN is transferred to the lowest node after which the water body is flooded to the volume of all rain collected.

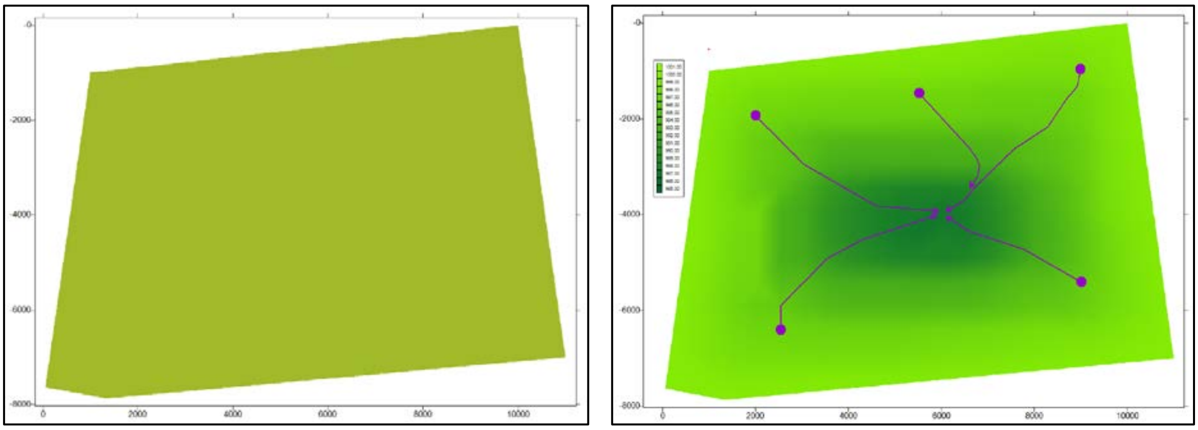


Figure 5-3: The single water body (left) and the flow paths from five positions on the TIN (right).

Two different rainfall happenings were generated, the first one had rainfall of 0.10 m and the second used a 0.20 m rainfall resulting in waterbodies of 6.97 x 10^6 m^3 and 13.94 x 10^6 m^3 respectively.

WISH will search for the water level elevation belonging to the water volume. First WISH will sort the potential nodes of this water body from the lowest to the highest elevation. During the next step, the water level elevation in the water body is simulated using the node elevations from the sorted list. The volume of water is calculated for every node surface elevation. When the node is reached where the volume is larger than the recharge volume, an interpolation method is used to find the correct elevation between the last two nodes. With the right water level elevation, the water body is flooded, and the map is updated to depict the water body. Moving the mouse on top of the water body will display a label with the water level elevation and the volume water (Figure 5-4).

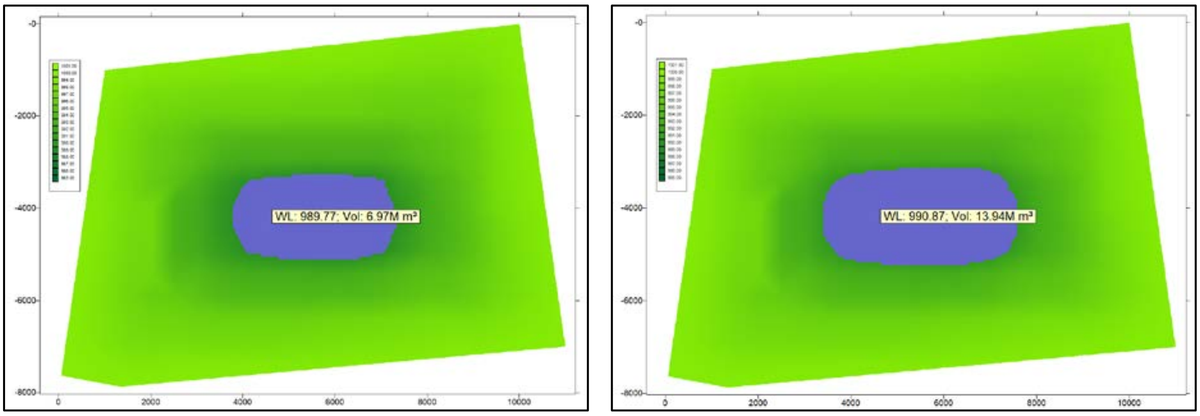


Figure 5-4: SImulation with 10 cm rainfall (left); with 20 cm rainfall (right).

5.2 Test site with two depressions.

The second model is based on the same TIN as the previous one; the only difference this model has two depressions. The total area number of nodes and elements and even the geometry of the nodes and elements is the same. On the surface, two 13 meters deep depressions were created.

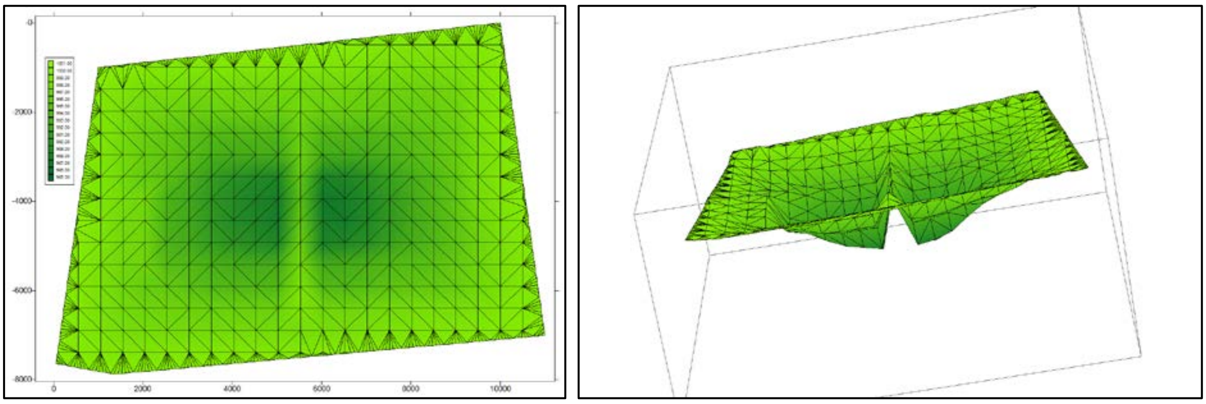


Figure 5-5: Dual depression test TIN with the elements visible.

The surface elevations ranged from 987 to 1000 m. A stage-volume curve shows that the maximum volume of water that can be stored in the depression is 250000 Ml. At that time the water level will be at 1000 m.

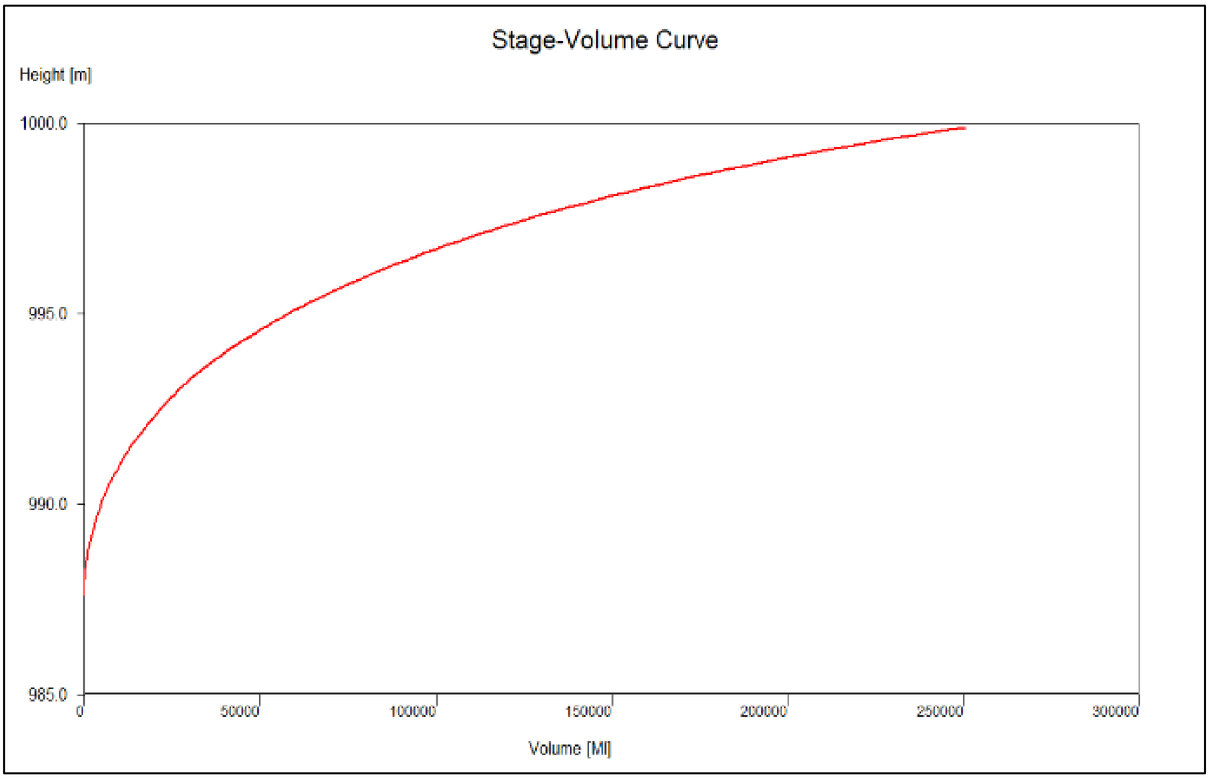


Figure 5-6: Stage-Volume curve for water stored on the surface.

Because this surface has two depressions, two potential water bodies are created. For diagnostic reasons a display mode is developed to show the different water bodies and which nodes or elements are draining towards these waterbodies. Every water body is assigned a random colour. Figure 5-7 (left) shows the two areas draining towards the two depressions and the right figure show flow path calculated from the five points on the surface towards the lowest points in the two water bodies.

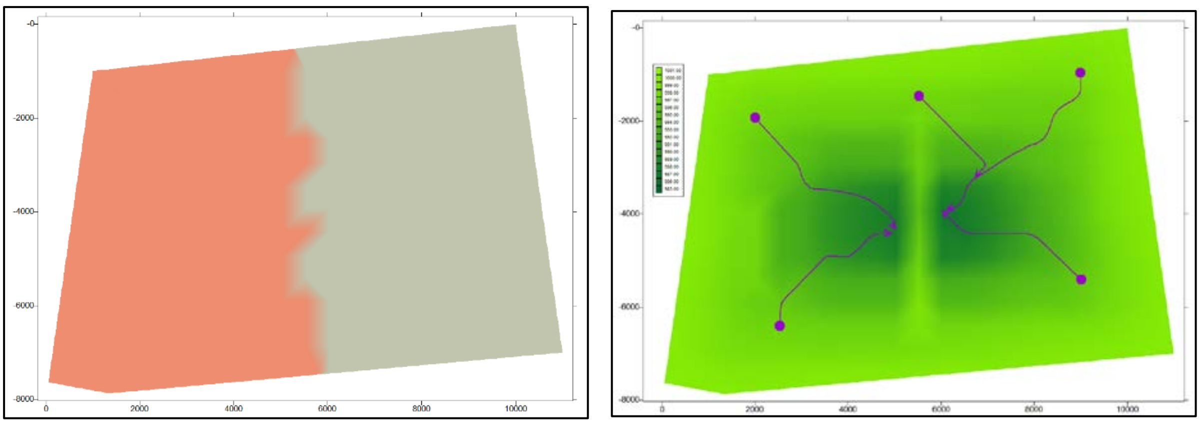


Figure 5-7: The two water bodies (left) and the flow paths from five positions on the TIN (right).

Simulating a 10 cm and a 20 cm rainfall two separate water bodies develop. Both water bodies, left and right, have more or less the same volume because the two areas draining towards the lowest points are equal in size. Adding the left and right volumes result in the same amount as seen in the previous test example (3.47 + 3.50 = 6.97 M m^3 and 6.94 + 7.01 = 13.95 M m^3). Simulating a 40 cm rainfall doubles the total amount of water on the surface. The two water bodies are now connected and merged into one water body (Figure 5-9).

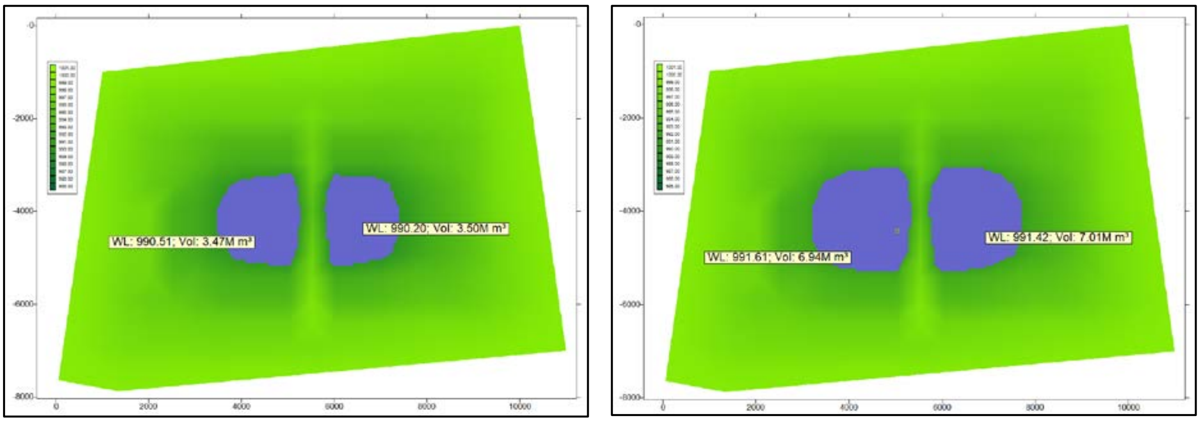


Figure 5-8: SImulation with 10 cm rainfall (left); with 20 cm rainfall (right).

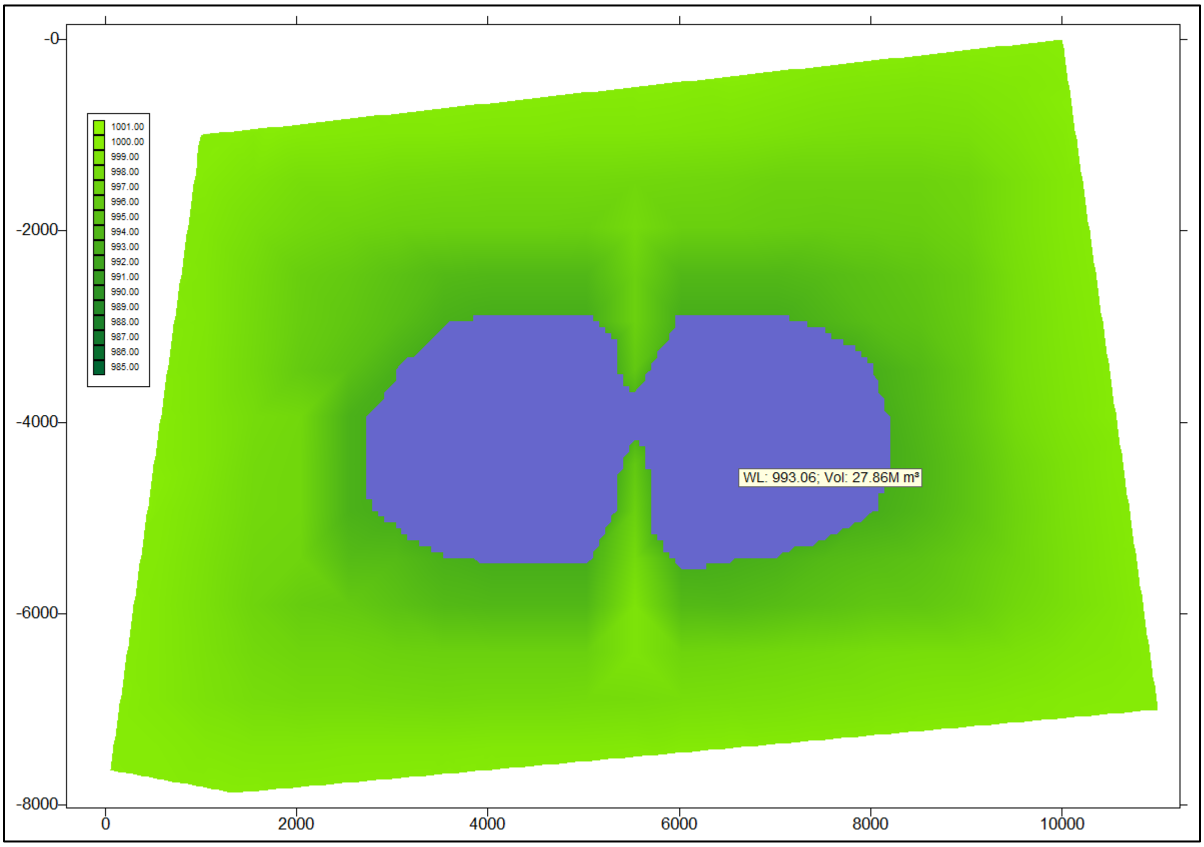


Figure 5-9: Simulation with 40 cm rainfall.

6 Case studies.

The site used for the case studies is fictive. The mine's layout is a combination of several existing underground workings as are the surface and coal seam floor contours. The site is created to demonstrate two different scenarios: recharge by rainfall and flooding by stored water.

"Whiskey is for drinking; water is for fighting over."
- Unknown (widely attributed to Mark Twain, but this is under dispute)

6.1 Site characteristics:

The site is situated in the Mpumalanga Province. The total footprint of the mine 1560 ha. A gently flowing surface with an average elevation around 1750 mamsl. The mine has an average depth 100 m and an extraction rate 72%. Figure 6-1 shows the mine outline and a 100 m buffer.

Figure 6-1: Layout of the underground mine (peach) and a ~ 100 m buffer (green).

Questions to be answered:

1. Planners of an underground mine in its development phase need to know how much water will recharge and how it will be distributed throughout the mine.
2. Storing water underground has its dangers. Determine which areas will be after a seal failure.

6.2 Preparation:

The preparation for both case-studies are the same and are outlined in the next few steps.

6.2.1 Select mine outline

The first step is building a 3-D model of the mine's floor, roof and surface contours. The outline of the mine in the form of a polygon may not have any crossing or duplicate line segments, must also be continuous and closed. Selecting the polygon and bringing up the context-sensitive menu (right-hand mouse button) displays the option to build the TIN.

6.2.2 Create TIN

The Create-TIN dialogue will ask for a nodal distance and fixed node position if any. For the underground floor, I have created a TIN with a nodal distance of 60 metres, without any from a separate file on a new layer called *Floor*.

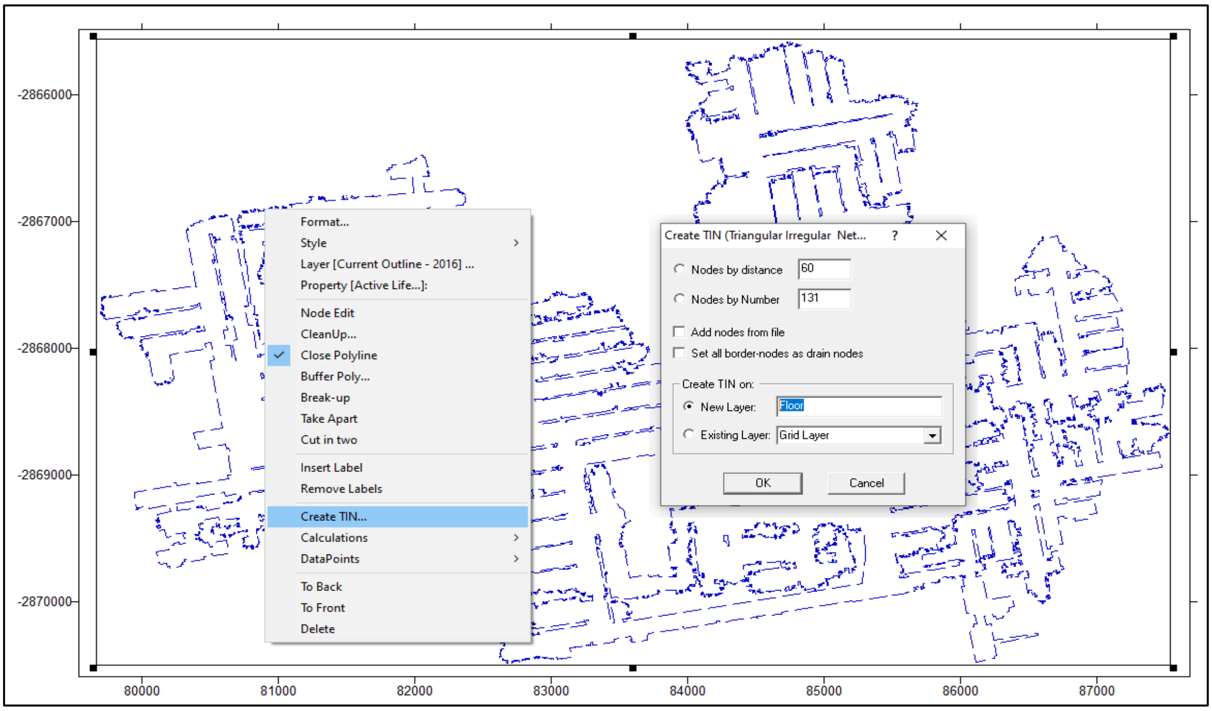


Figure 6-2: Creating a TIN.

The TIN generator starts with the placement of the nodes first on the outline using a node distance a tenth of what was specified. The nodes inside the outline are placed on the distance specified where after the element generator starts working. This is an automatic process, and no interaction of the user is required. After the triangles (elements) are built, they do not know or have no reference of their neighbouring elements. The connectivity checker analyses all triangles and store references of the neighbouring triangles inside each element. The connectivity checker takes a bit of time to complete but is, in the long run, a time-saving process because it eliminates the need to find and identify the adjoining triangles every time the dewatering process visit an element. Figure 6-3 shows the empty TIN.

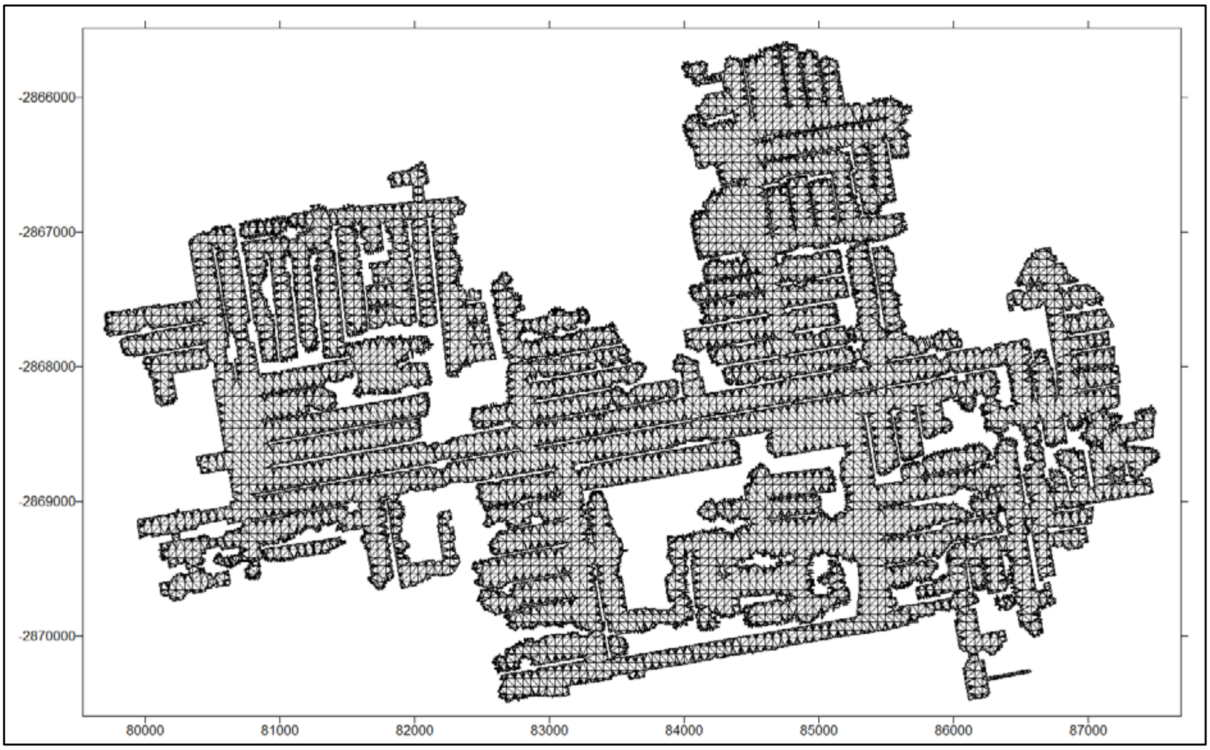


Figure 6-3: The TIN created - element visibility turned on.

6.2.3 Assign Z-values (Floor)

The TIN consists of 42835 nodes and 47045 elements and is considered empty because it does not have any elevation data assigned. Without elevation data (or any other data for that matter) the triangles have no colours assigned and are therefore transparent. It is not possible to turn the elements visibility off as this will result in an invisible TIN. To add elevation data, select the TIN and select from the menu *Add Contour Data…* A dialogue will open, allowing the user to choose the data source, the search method and interpolation settings. After selecting the OK button, the user is requested to select the data file after which the first eight lines of the file are listed displaying the data layout, and the user can specify the right columns (Figure 6-4).

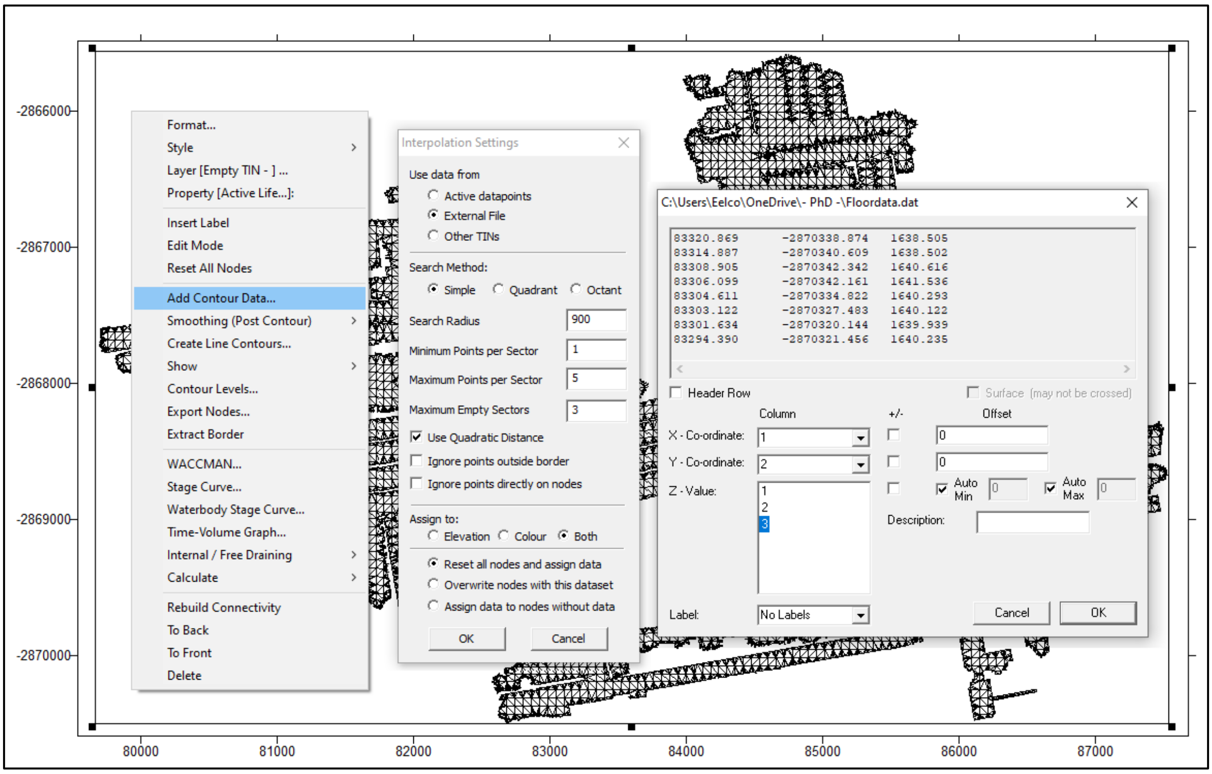


Figure 6-4: Menu and dialogue windows used to specify the contouring options.

After the values are assigned to the nodes, colours are associated with the values. The warm colours will depict the values on the high end, the cool colours the values on the low end of the scale. Figure 6-5 shows the mine outline and the TIN from the previous step with floor elevations assigned.

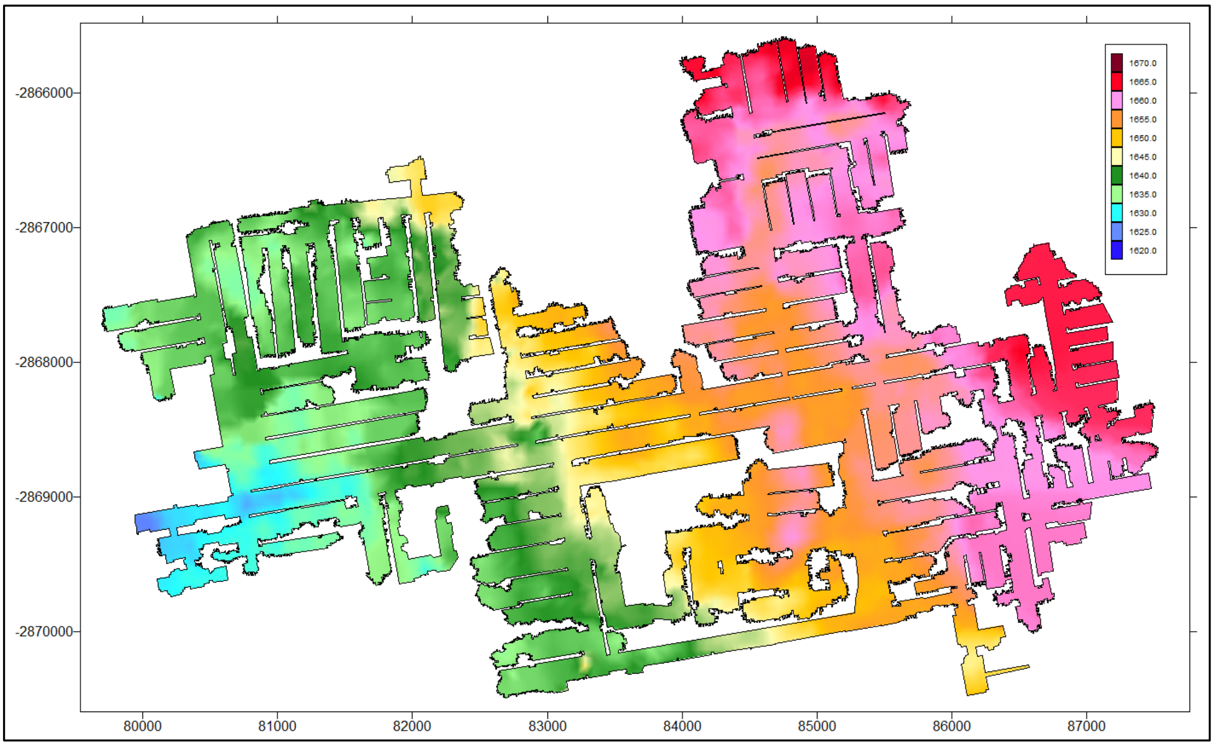


Figure 6-5: Mine-outline with floor contours.

The case study is related to an underground mine and the roof contours must be created.

- Copy TIN (from step 2)
- Assign the roof z-values (making use of peg elevations or a mining height)

To create the surface TIN, I have used a 100 m buffered outline. The buffered outline results in a "better" 3D picture, using a rectangular shape for the surface TIN would have obscured more than half of the floor TIN. This change is purely cosmetic; it does not make a difference to the result.

- Select Outline
- Create a 100 buffer (See Figure 6-1).
- Select the buffer
- Create TIN
- Assign surface elevation values (Figure 6-6 shows two TIN's depicting the surface and floor contours.)

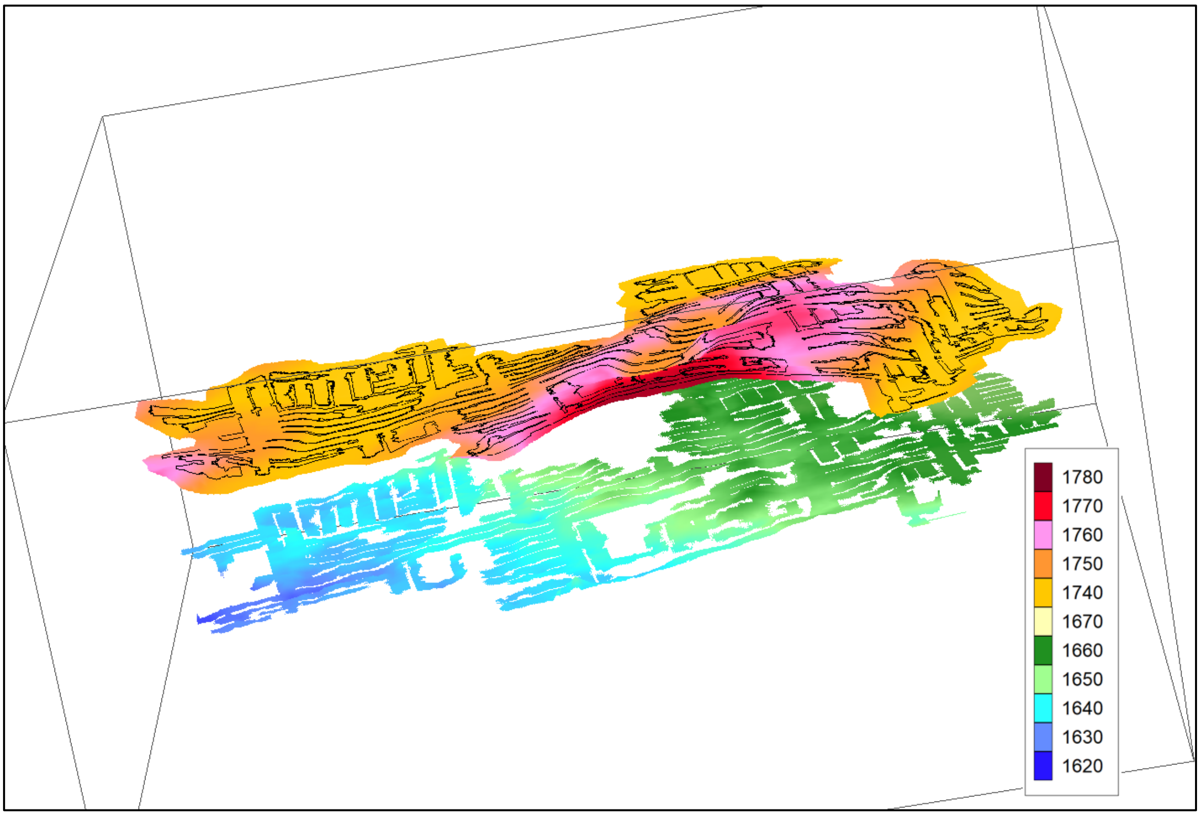


Figure 6-6: 3D view of the underground mine in relation to the surface.

To determine the depth of mining (roof thickness)

- Copy TIN (from step 2)
- Assign data by subtracting the TIN from step 4 from the surface TIN (Figure 6-7).

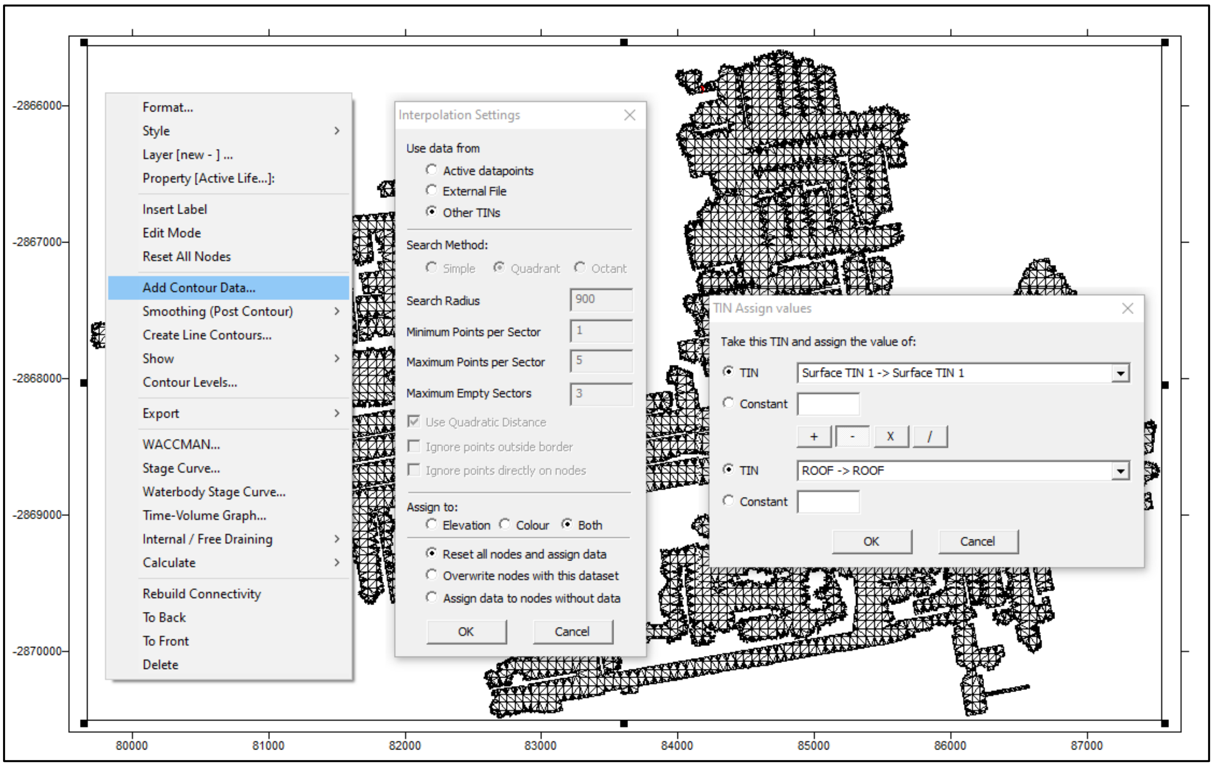


Figure 6-7: Menu and dialogue windows used to specify contour values calculated from existing TINs.

The results, a contour of the roof thickness, or the depth of mining, is displayed in Figure 6-8. The colour scheme is automatically assigned.

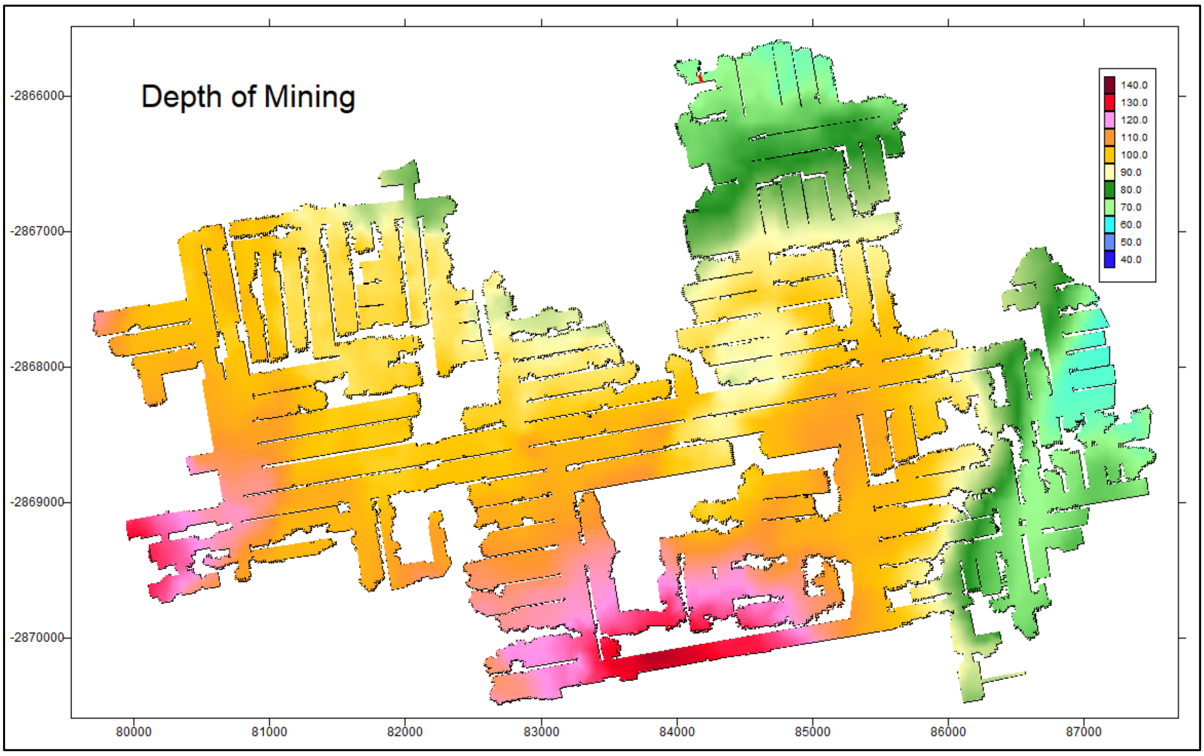


Figure 6-8: Underground mine displaying the roof thickness.

The preparations to start simulating the flooding is now complete. The TIN properties are displayed in Table 11.

Table 11: TIN properties.

Coordinates	Min: 79694.92 , -2865583.61 Max: 87529.31 , -2870491.12
Perimeter	316389 m
Area	15665261 m²
Nodal distance	60 m
# Nodes	42835
# Elements	47045
Surface Elevation	Min: 1729.13 – Max: 1777.50 – Avg: 1748.67 mamsl
Mine-floor Elevation	Min: 1624.58 – Max: 1667.00 – Avg: 1649.27 mamsl
Depth of Mining	Min: 62.99 – Max: 135.90 – Avg: 96.63 m

6.3 Case Study 1 – Flooding by recharge

Flooding by recharge occurs when, as the term suggests, enough water recharges through the overlying strata. Although recharge is also happening from lateral flow this water is usually included in the recharge rate. The recharge rates are determined for general conditions in the Mpumalanga province. Simplication of the real world – no surface features are considered – except by the different recharge factors.

Recharge factors are assigned depending on the depth of mining by using the values from Table 4. Figure 6-9 shows the TIN with the roof thickness or depth of mining values. Superimposed are the areas with mining depth: Shallow (red) – intermediate (yellow) – Deep (green) with 2 – 4 – 8 % recharge.

These three polygons can be used to assign the different recharge rates to the individual elements (triangles), by selecting a polygon and clicking on the set TIN properties option for each polygon. The area for each recharge rate was determined for control purposes (Figure 6-10), making it possible to calculate the expected total volume of recharge water recharging the mining void. The separate quantities were calculated per recharge rate and totalled for different amounts of rainfall (Table 12).

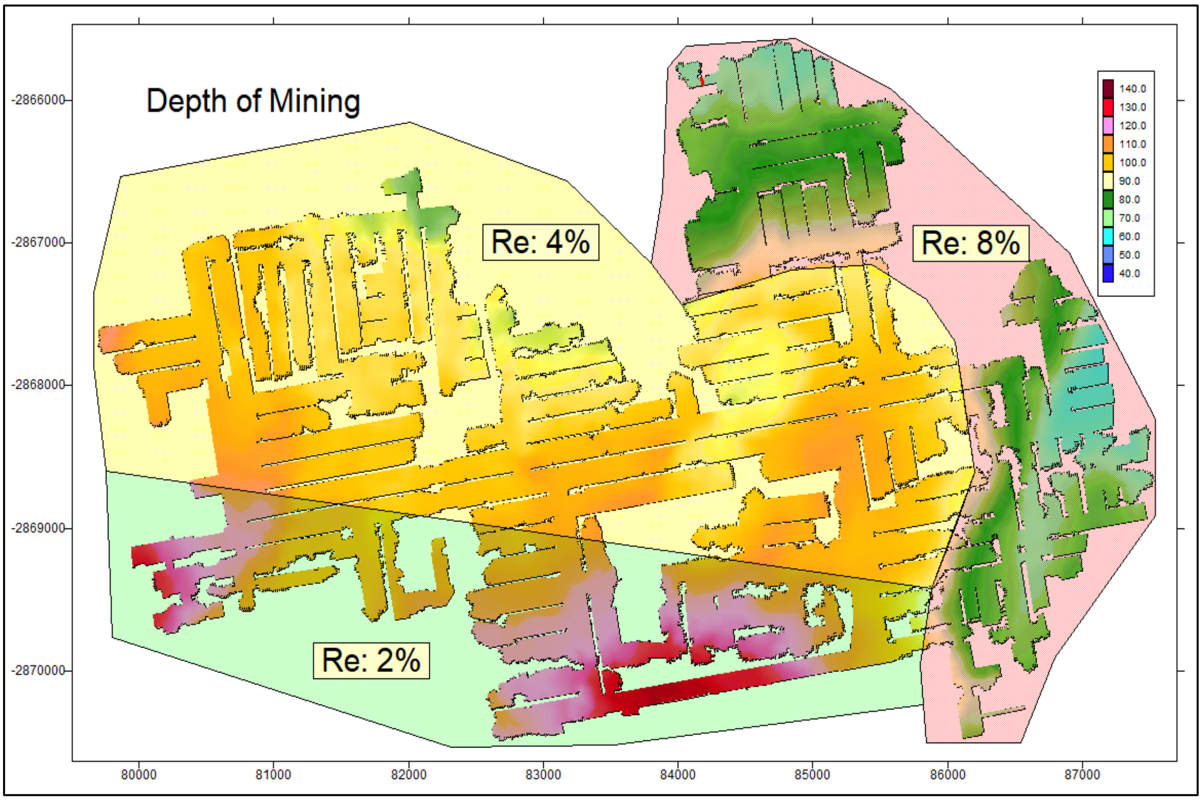


Figure 6-9: Proposed recharge areas and percentages based on the depth of mining.

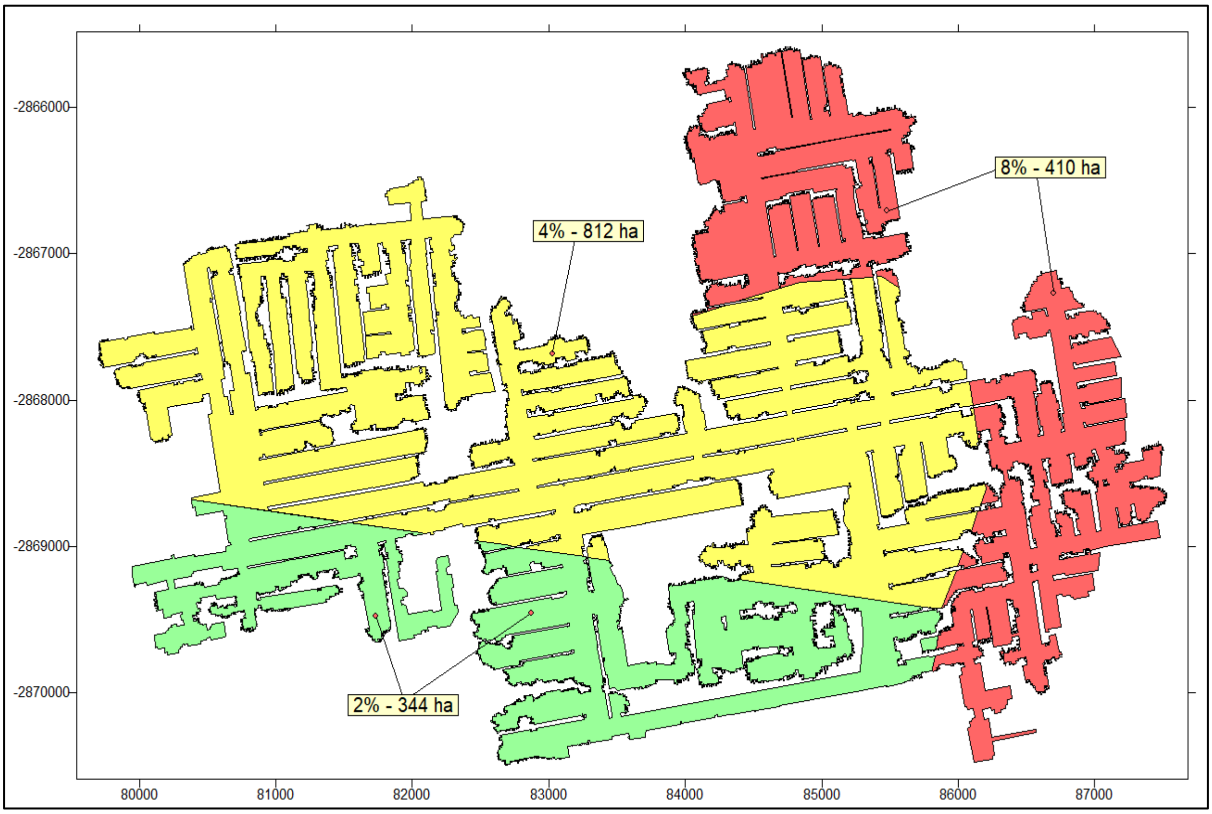


Figure 6-10: Underground mine with assigned recharge rates.

The recharge flooding simulation is started by selecting the surface; in this case, the floor-TIN and bringing up the context-sensitive menu and selecting *Calculate* and *Recharge*. The recharge-dialogue box will pop-up where the user must enter the amount of rainfall in metres (Figure 6-11).

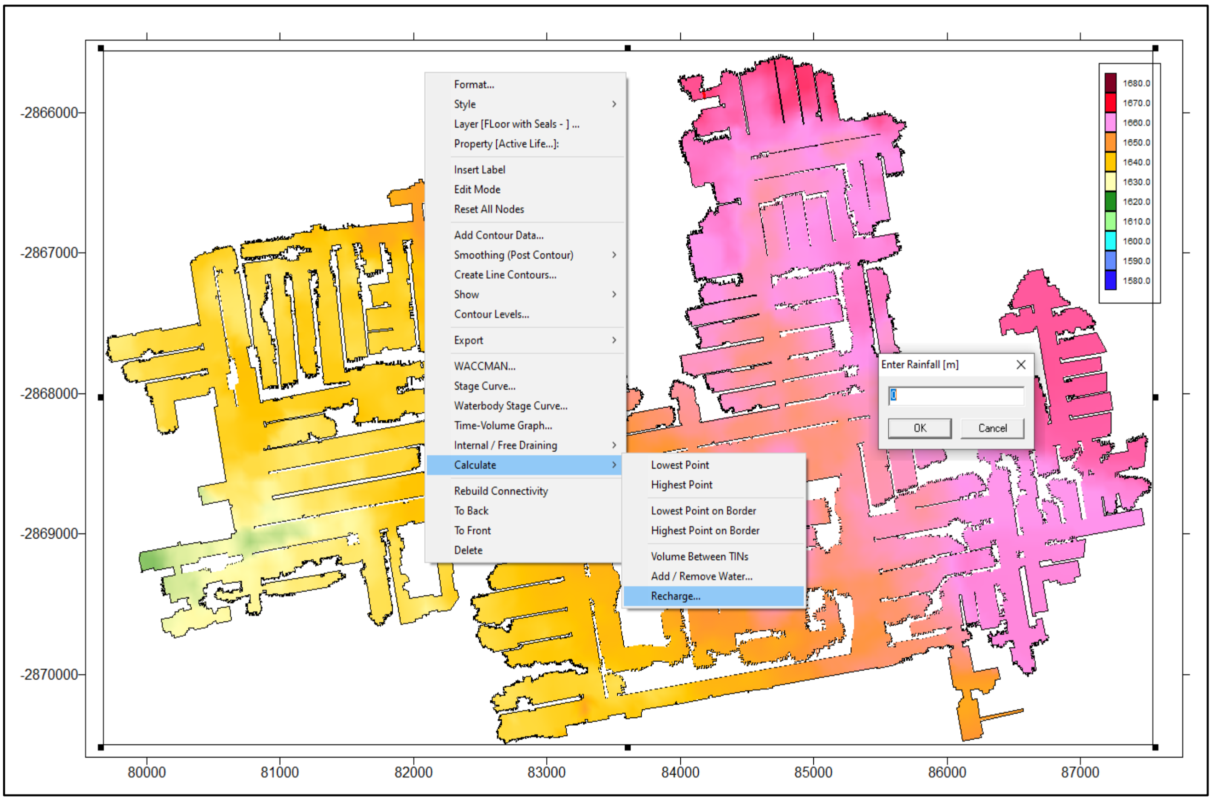


Figure 6-11: Opening the recharge dialogue.

Six rainfall scenarios were tested from a low rainfall of 100 mm to a high rainfall of 3000 mm. Table 12 shows the recharge water expected in the underground based on the areas with the different recharge rates and the total rainfall applied.

Table 12: Expected recharge for each of the recharge zone.

	Rainfall [mm]	100	250	500	1000	2000	3000
Recharge Rate	Area [ha]	Vol [Ml]	Vol [Ml]	Vol [Ml]	Vol [Ml]	Vol [Ml]	Vol [Ml]
8%	413	33.0	82.6	165.2	330.4	660.8	991.2
4%	815	32.6	81.5	163.0	326.0	652.0	978.0
2%	345	6.9	17.3	34.5	69.0	138.0	207.0
Total	**1566**	**72.5**	**181.4**	**362.7**	**725.4**	**1450.8**	**2176.2**

The mine floor is often considered a flat surface, but in reality, it is a surface with many depressions. The amount of lows depends on the detail of the measurements. Selecting the mine floor TIN and starting WACCMAN initiates the initialization process where element connectivity is established (*BuildConnectivity()*), node- and element lists are created (*AnalyseNodeConnections()*) and the TIN is analysed for all low points on the surface (*FindRechargeNode()*). WISH reports that the mine floor has 3002 separate depressions or recharge nodes in the surface. A recharge node is a node that has a lower elevation than all directly connected nodes. When the recharge nodes are identified, the (potential) water bodies are created. The waterbody consists of all the nodes and elements that drain towards the recharge node. Figure 6-12 shows a detail view of the potential waterbodies. The small little squares indicate the positions of the recharge nodes for each water body.

WACCMAN starts with the identification of recharge nodes (the lowest points on the surface) and the creation of the waterbodies. Next, it will calculate the amount of precipitation that is intercepted by each triangle (element). The water volumes of all the elements draining towards the same waterbody are accumulated and transferred to the recharge node.

The program will iterate through all waterbodies filling them with the recharge water. If the recharge water is more than the capacity of the waterbody, the excess volume will spill over into the next waterbody. Water that is in the waterbody is no longer allocated to the recharge node. When two adjacent water bodies touch, the water bodies are merged.

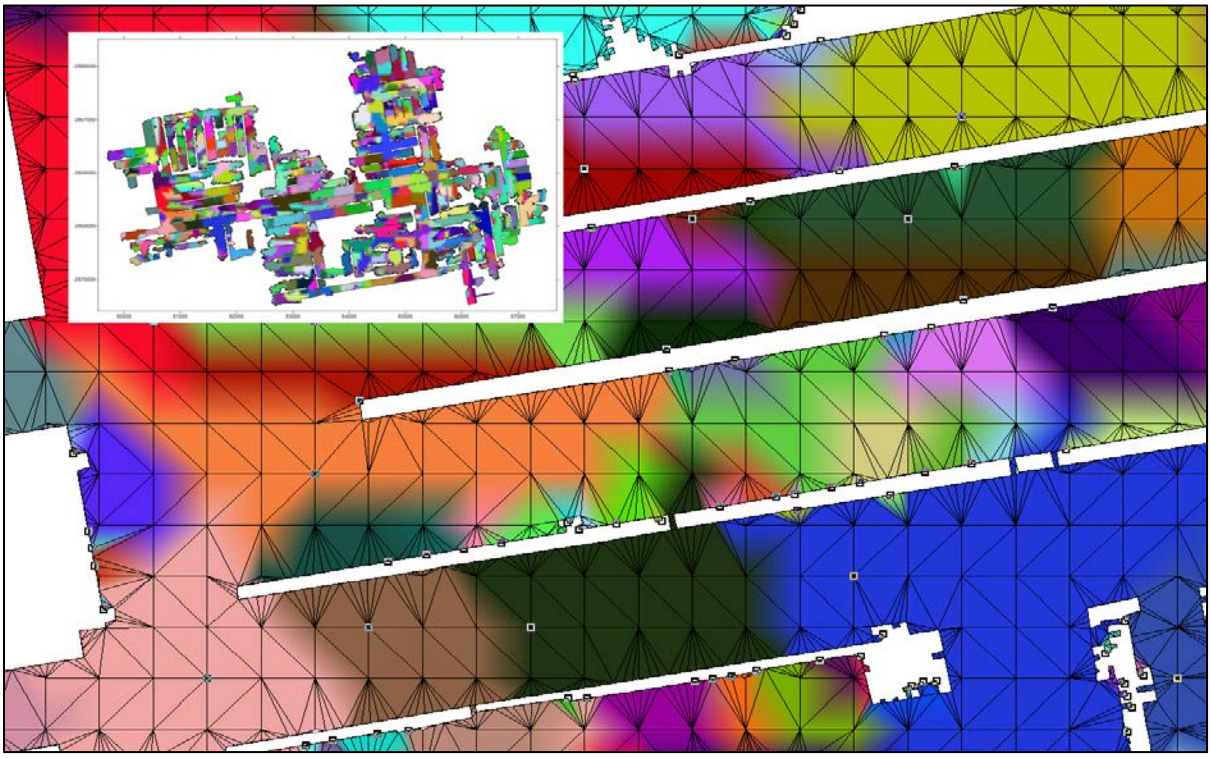


Figure 6-12: Detail of waterbodies. The lowest node in each waterbody indicated by a square point.

Table 13 shows the results of the six flooding scenarios. It shows the expected recharge, as calculated by WISH, the recharge after the simulation is completed, the difference between the expected and simulated recharge in cubic metres and a percentage of the predicted recharge, the total area flooded, the number of water bodies left and the number of iteration it took as well as the number of times simulation needed to go through the list of water bodies.

These values in the table are reported by WISH via a notification window that indicates the end of the simulation process (Figure 6-13). The difference between the expected and simulated volumes are minimal, below 0.5% and in most cases, closer to 0%. The absolute error grows with larger recharge volumes. The error can be associated with the flooded area. As the flooding occurs in millimetre increments, the change in volume between rises will be more substantial with larger flooded areas. Larger recharge volumes also result in a decline in the number of waterbodies.

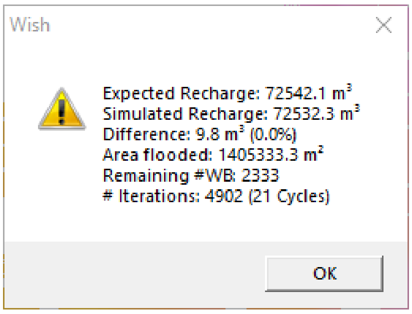


Figure 6-13: Notification window after recharge simulation with 100 mm rainfall.

Table 14 lists the progress of the recharging simulation. During the first pass through the waterbody-list, almost half of all allocated water to be recharged is transferred to the waterbodies.

Table 13: Expected and simulated recharge.

Rainfall [mm]	Recharge Exp. [m^3]	Recharge Sim. [m^3]	Difference [m^3] (%)	Flooded area [ha]	Waterbodies Left #	Iterations # (Cycles #)
100	72542	72532	-9.8 (0%)	140.5	2333	4902 (21)
250	181355	181381	+25 (0%)	205.6	2141	5397 (23)
500	362710	362690	-20 (0%)	272.2	1993	5657 (25)
1000	725422	725572	+151 (0%)	358.7	1846	5982 (32)
2000	1450843	1450561	-282 (0%)	451.9	1688	6536 (72)
3000	2176264	2175968	-296 (0%)	507.8	1594	6771 (66)

The first pass also resulted in 3002 iterations (the number of water bodies in the list). During which many of the water bodies were filled to the brim, and the overflow nodes and target waterbodies were identified. The second pass only had 850 iterations (3852-3002), only those water bodies with capacity left were visited and 301 water bodies were merged. During the third pass, another 110 water bodies combined and the error between the volume that must be recharged and the amount already in the water bodies declined to 30%. After the 21st pass, the absolute difference between expected and simulated recharge is smaller than 10 m^3, there are 2333 water bodies left (669 were merged), and the simulation took 4902 iterations. The recharge progress tables for the simulated rainfall of 250, 500, 1000, 2000 and 3000 mm are available in the appendix.

Table 14: Simulation progress – 100 mm of rainfall.

Cycle	Expected Recharge [m^3]	Simulated Recharge [m^3]	Difference [m^3]	Difference [%]	# WB	Iterations
1	72542.1	31528.2	41014.0	56.5	3002	3002 (3002)
2	72542.1	45970.1	26572.1	36.6	2701	850 (3852)
3	72542.1	50801.9	21740.3	30.0	2591	382 (4234)
4	72542.1	58328.9	14213.2	19.6	2516	221 (4455)
5	72542.1	61838.3	10703.9	14.8	2469	130 (4585)
6	72542.1	63376.0	9166.1	12.6	2436	90 (4675)
7	72542.1	66627.1	5915.0	8.2	2411	61 (4736)
8	72542.1	67350.5	5191.6	7.2	2391	43 (4779)
9	72542.1	68421.8	4120.4	5.7	2380	30 (4809)
10	72542.1	68866.1	3676.0	5.1	2370	25 (4834)
11	72542.1	69293.1	3249.1	4.5	2361	17 (4851)
12	72542.1	69629.7	2912.5	4.0	2354	11 (4862)
13	72542.1	70420.5	2121.6	2.9	2351	9 (4871)
14	72542.1	70794.6	1747.5	2.4	2346	9 (4880)
15	72542.1	71990.4	551.7	0.8	2342	7 (4887)
16	72542.1	72205.9	336.3	0.5	2339	4 (4891)
17	72542.1	72293.4	248.7	0.3	2338	3 (4894)
18	72542.1	72364.6	177.6	0.2	2336	2 (4896)
19	72542.1	72402.4	139.8	0.2	2336	2 (4898)
20	72542.1	72523.3	18.8	0.0	2334	3 (4901)
21	72542.1	72532.3	9.8	0.0	2333	1 (4902)

Figure 6-14 shows the recharge simulation results as water bodies on the mine floor. As expected, the water bodies are scattered over the extent of the mine floor. Water bodies smaller than 10 m³ are filtered from the view; this is a setting that can be adjusted from the main WACCMAN dialogue window. Figure 6-15 shows the same mine floor but now with just more than 181 000 m³ of water as the result of 250 mm rainfall.

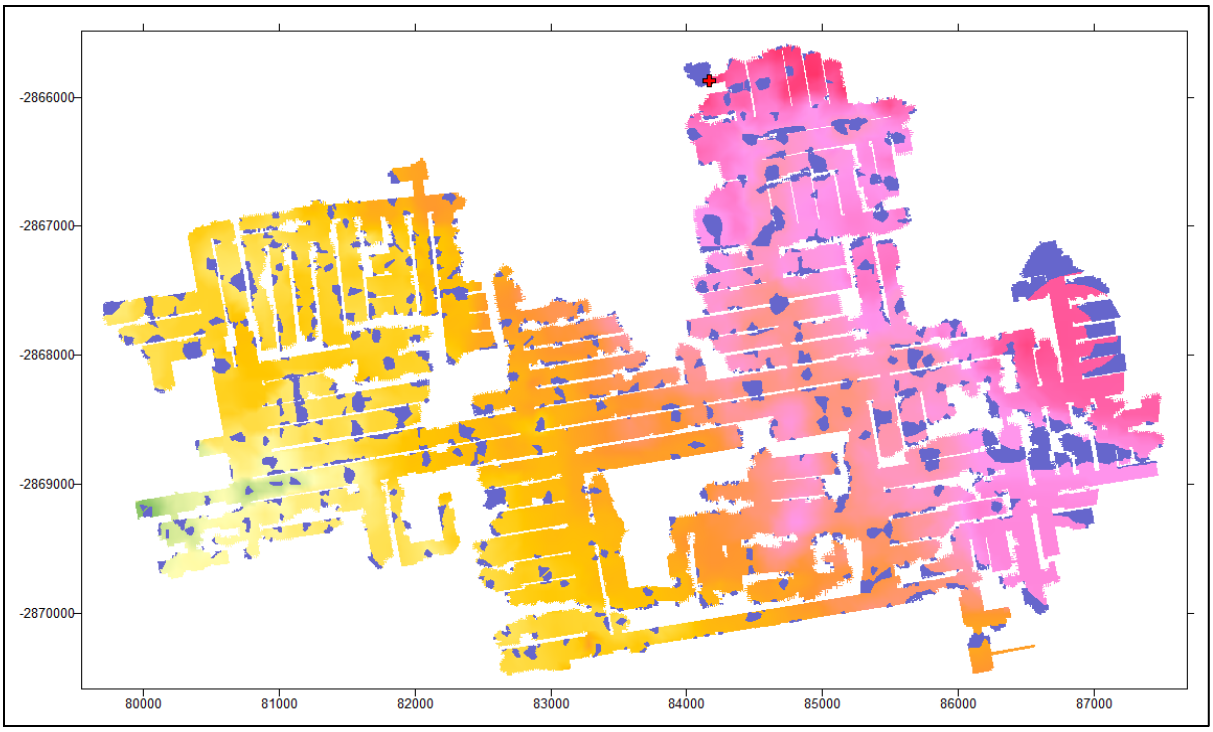


Figure 6-14: Rainfall 0.1 m – 72532 m³ recharged in 4902 iterations.

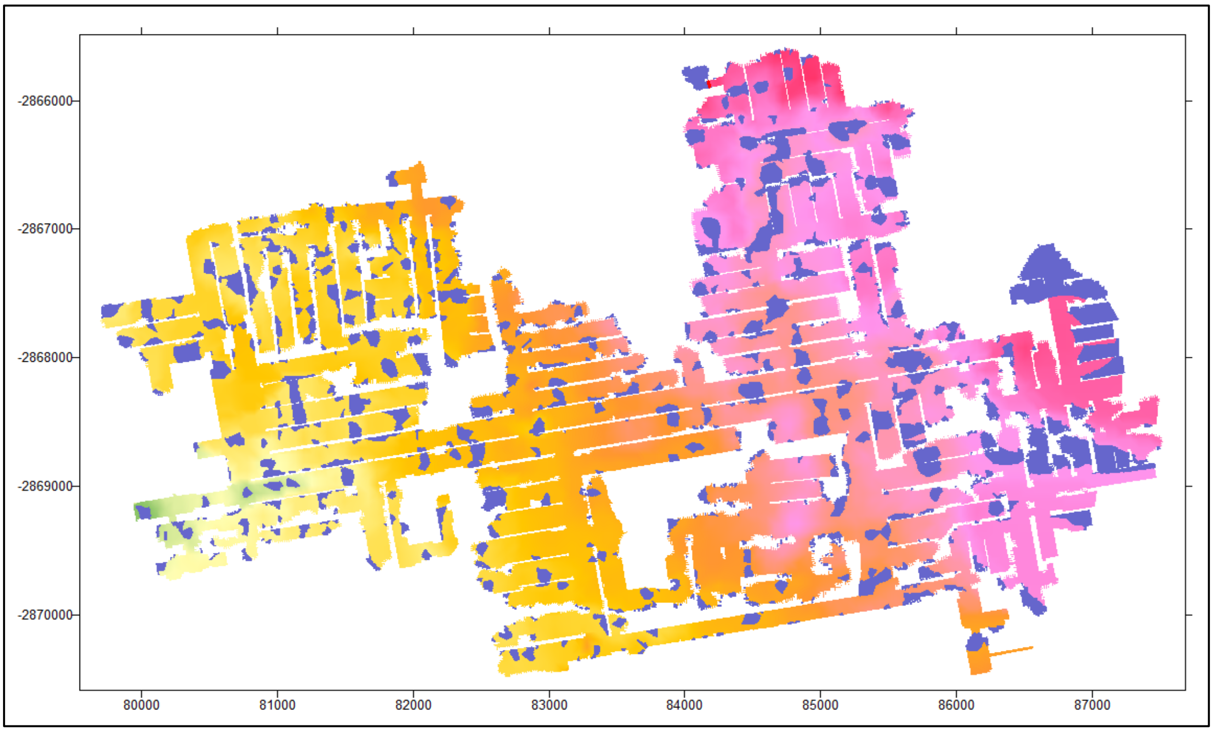


Figure 6-15: Rainfall 0.25 m – 181381 m³ recharged in 5397 iterations.

Figures 6-16, 6-17, 6-18 and 6-19 shows the water distribution after 250, 500, 1000, 2000 and 3000 mm respectively. Although individual waterbody volumes are not offered, the growth is easily observed.

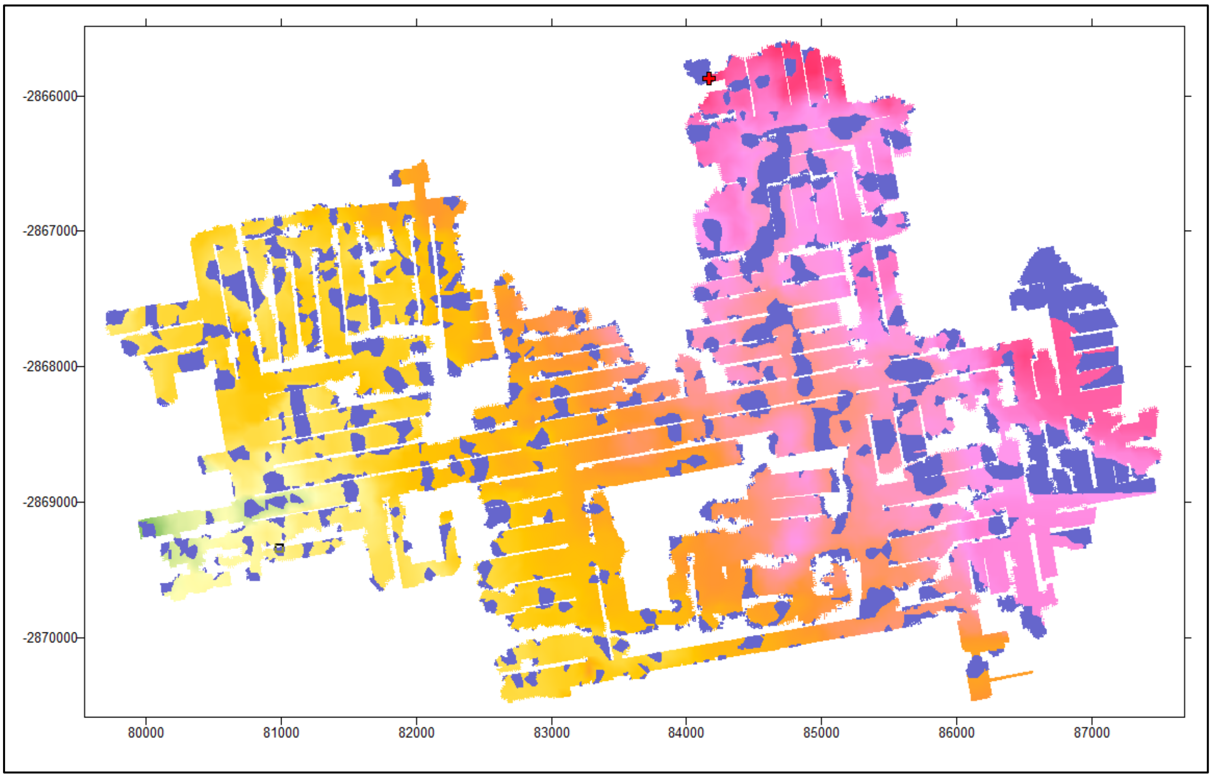


Figure 6-16: Rainfall 0.5 m – 362690 m³ recharged in 5657 iterations.

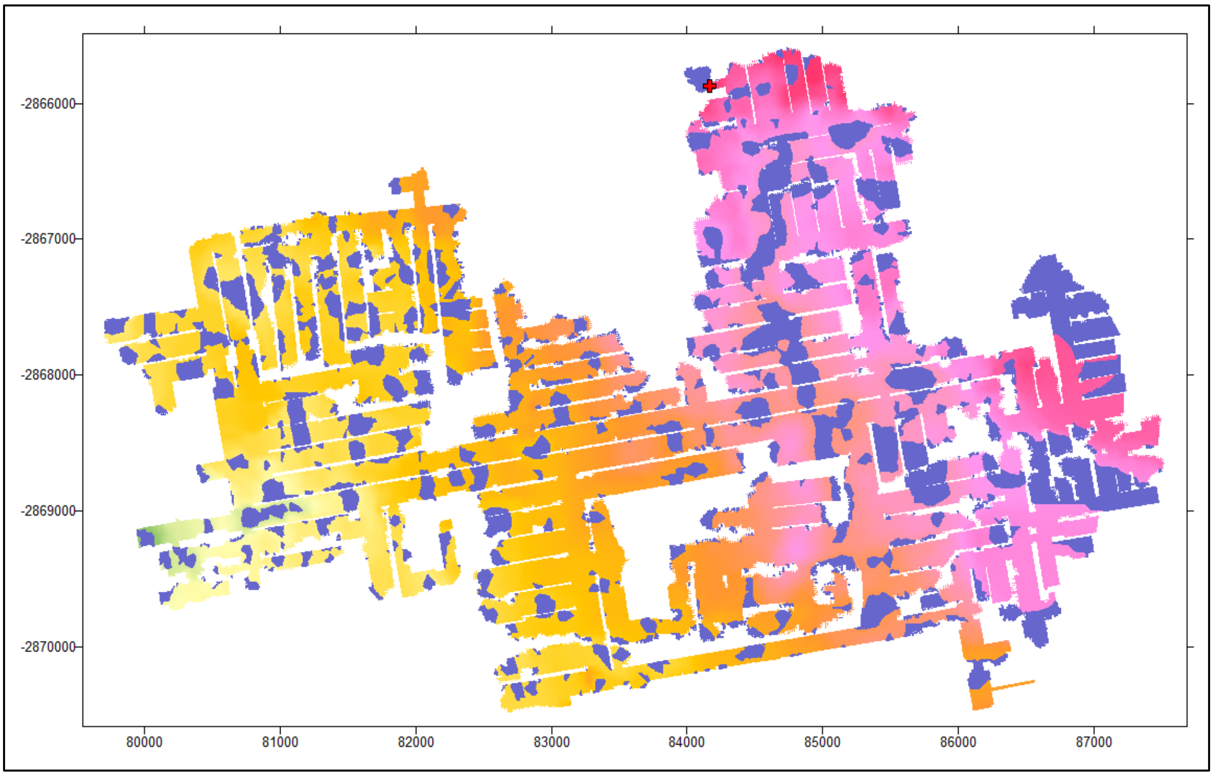


Figure 6-17: Rainfall 1.0 m – 725572 m³ recharged in 5982 iterations.

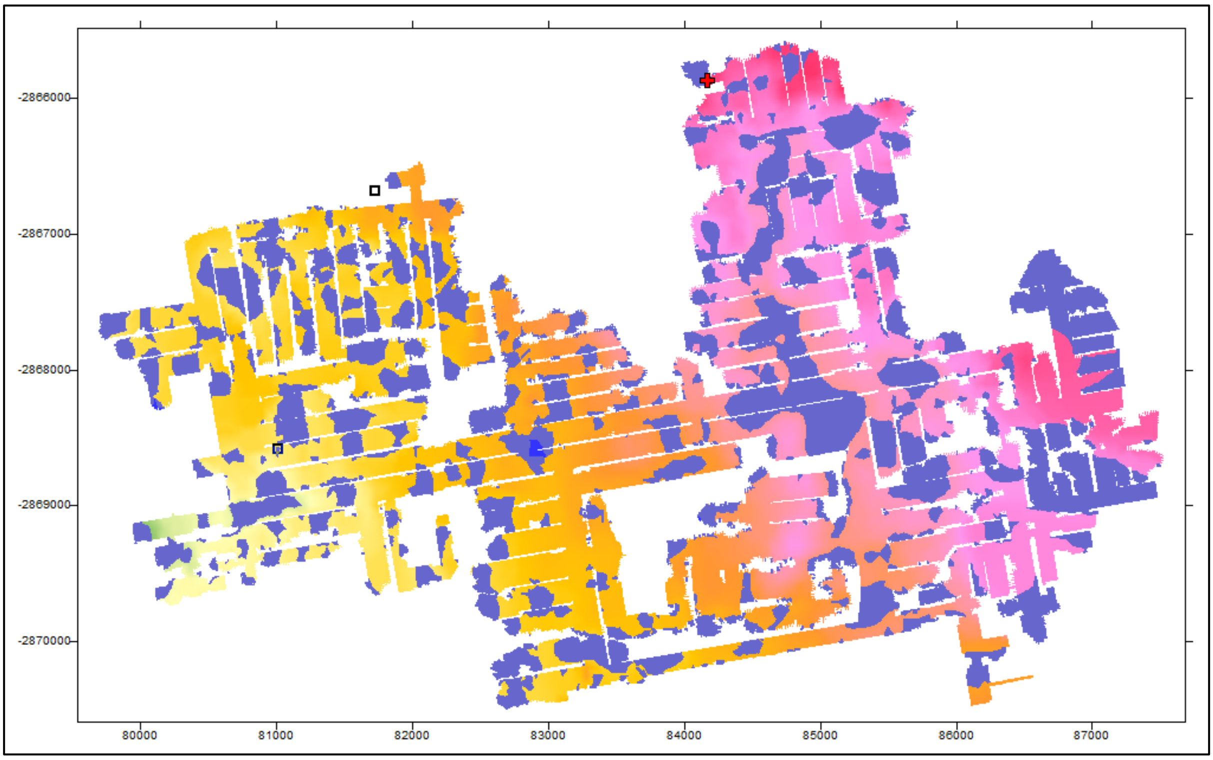


Figure 6-18: Recharge after 2.0 m – 1450561 m³ recharged in 6536 iterations.

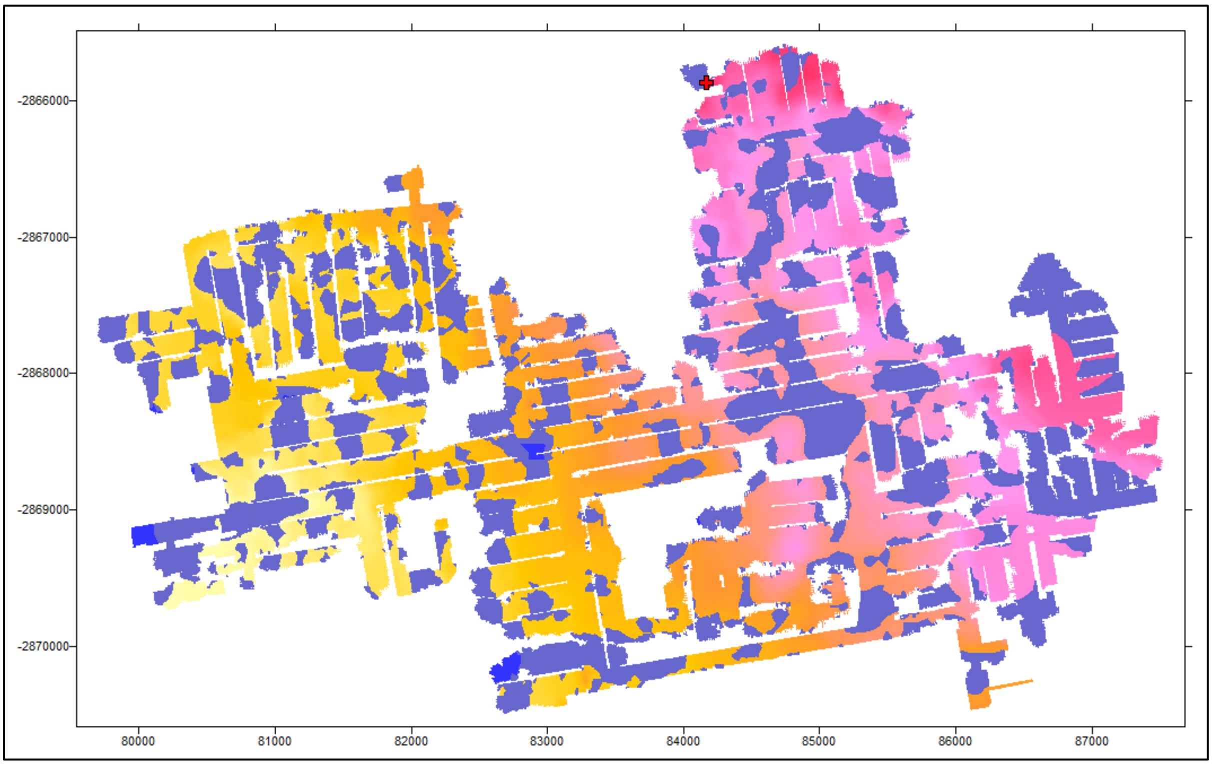


Figure 6-19: Rainfall 3.0 m - Recharge 2175968 m³ in 6771 iterations.

6.4 Case Study 2 – Flooding a single compartment

An active underground mine needs ventilation for the workers to keep working. The ventilation systems blow large amounts of fresh "clean" air inside the working and dispose of the same amounts of dirty air from the workings. The ventilation allows most of the recharge water to evaporate. The ventilation system does not service abandoned compartments, and the air inside these compartments becomes humidified minimising the possibility of any evaporation. Recharge water will flood the compartment. If such a chamber receives enough recharge water, it may overflow into a still active section.

In the next example, a dormant chamber of 70.73 ha is identified (Figure 6-20). The compartment is located in an area with a 4% recharge rate. To simulate the recharge in the abandoned section the recharge is set to 4%, the recharge for the active mining region is assumed to be 0%. The 0% compensate for the evaporation made possible by the ventilation system. All water recharged is evaporated by the ventilation system.

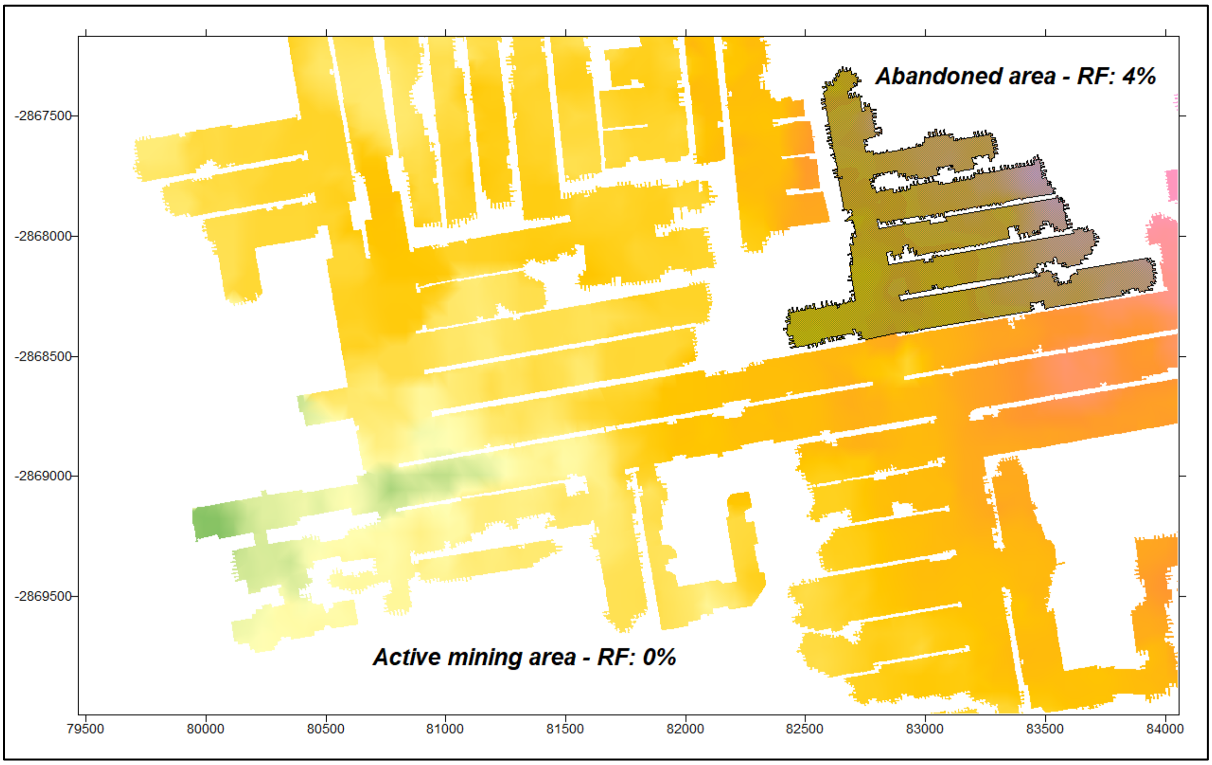


Figure 6-20: Abandoned compartment.

Four different precipitation amounts were simulated, 750, 1500, 3750 and 7500 mm representing an average rainfall for 1, 2, 5 and 10 years respectively. The simulations took fewer iterations, this is understandable because a much smaller area with fewer water bodies is affected, but more cycles were needed because the water bodies receive more water than they can handle. Water needed to be carried over to adjoining water bodies (Table 15).

Table 15: Recharge in a single compartment.

Rainfall [mm]	Recharge Exp. [m^3]	Recharge Sim. [m^3]	Difference [m^3] (%)	Flooded area [ha]	Waterbodies Left #	Iterations # (Cycles #)
750	21219	21219	-0.3 (0%)		2928	469 (13)
1500	42438	42293	+144.8 (0%)		2918	515 (15)
3750	106095	106095	0.0 (0%)		2905	549 (20)
7500	212190	212190	0.0 (0%)	36.4	2892	599 (45)

The water distributions after recharging the compartment with the different rainfall amounts are displayed below. For the first two conditions, the volume of water spilt in the haul way is neglectable. The volume spillage after 5-years of average rainfall is substantial (Figure 6-21).

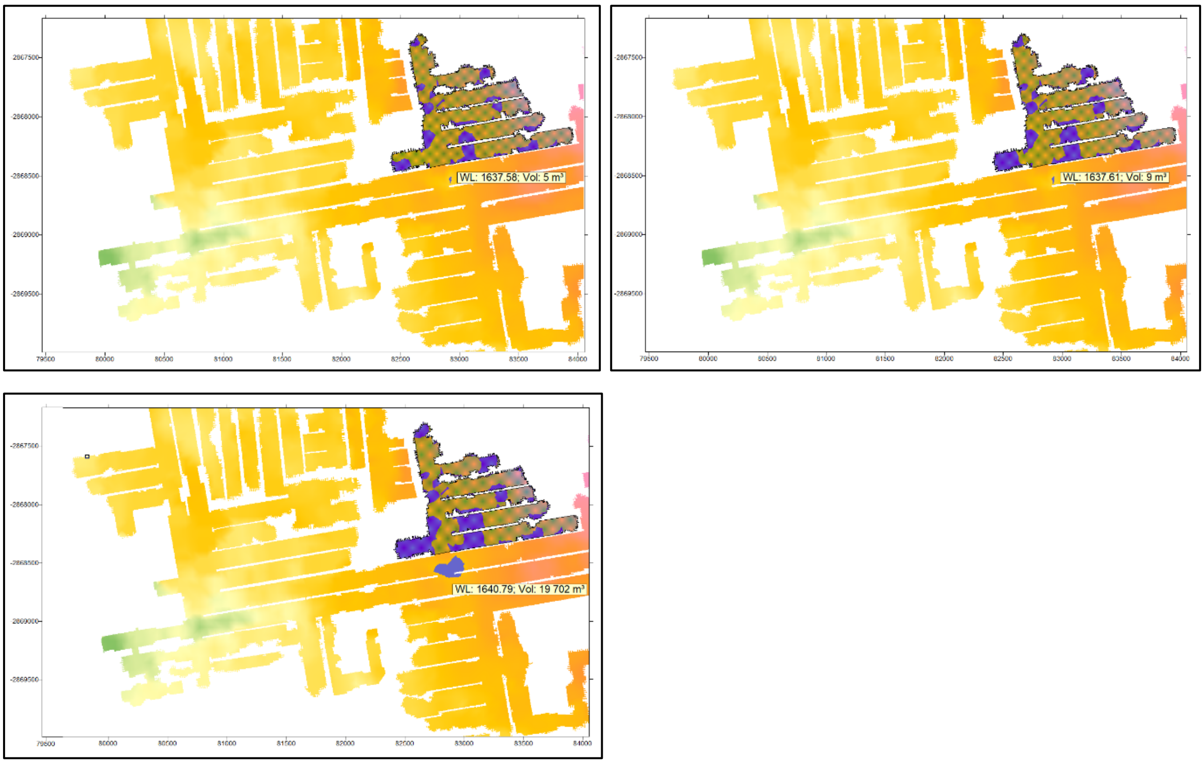


Figure 6-21: Recharge of the abandoned compartment with 750, 1500 and 3750 mm Rainfall.

After ten years with an average rainfall of 750 mm, almost 112 000 m^3 of water has spilt in the haul ways (Figure 6-22). The largest water body blocks both primary haul ways. The water spillage into the main road can be prevented by inserting a water-retaining wall at the entrance of the compartment.

This introduces a secondary problem: a growing volume of recharge water behind a wall. A larger water volume behind a wall must either be pumped to the surface or to another underground compartment to protect the wall and the workers inside the mine. Another option is to install a high-pressure seal to keep the water in the chamber. Figure 6-23 displays the water distribution in 3-D.

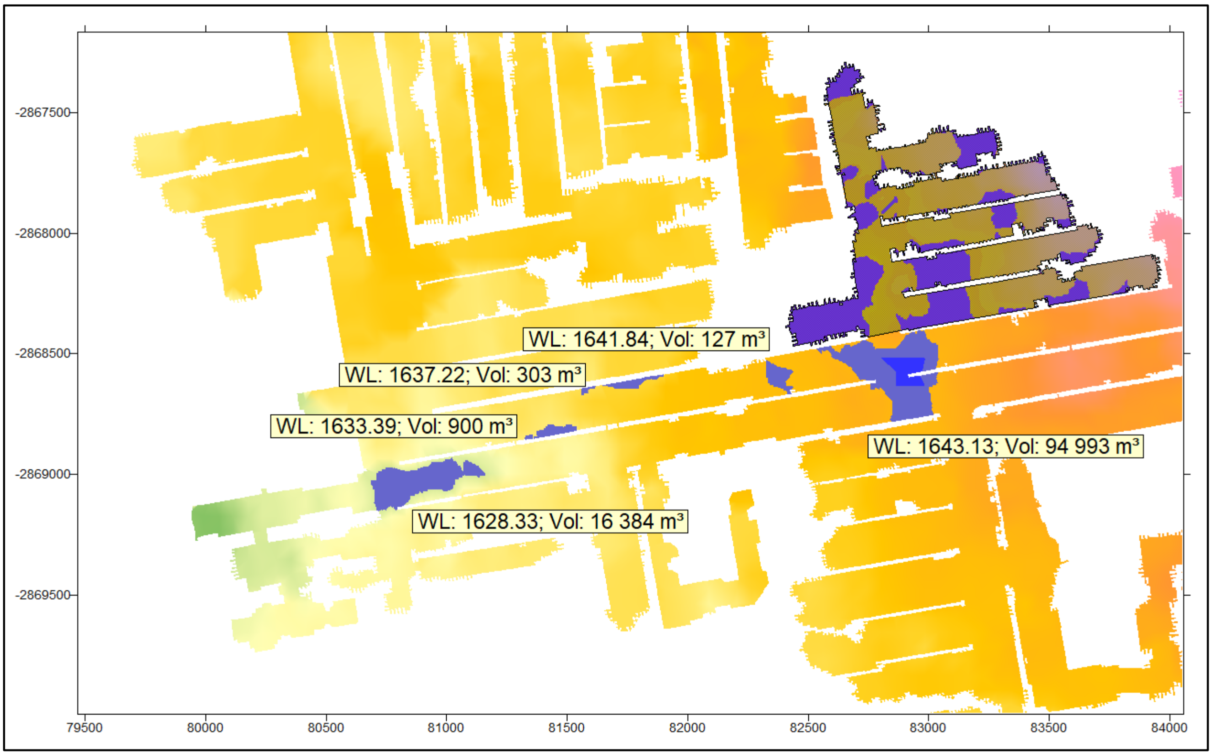


Figure 6-22: Water distribution after ten years of rainfall.

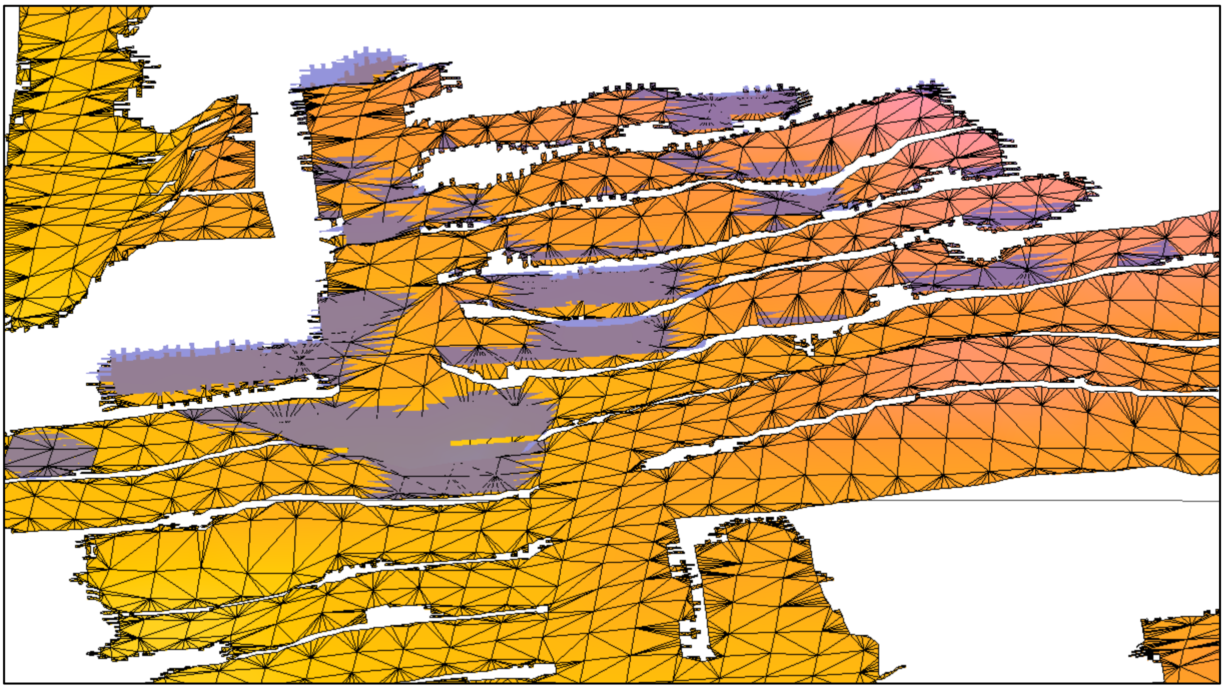


Figure 6-23: Water distribution after ten years of rainfall in 3D

6.5 Case Study 3 – Flooding by stored water

The third case study looks at the possibility of disaster flooding as a result of seal failure. Two separate water compartments were created to simulate a real-life situation. The two chambers have a water holding capacity of 0.5 and 1.5 M m³ each (Figure 6-24).

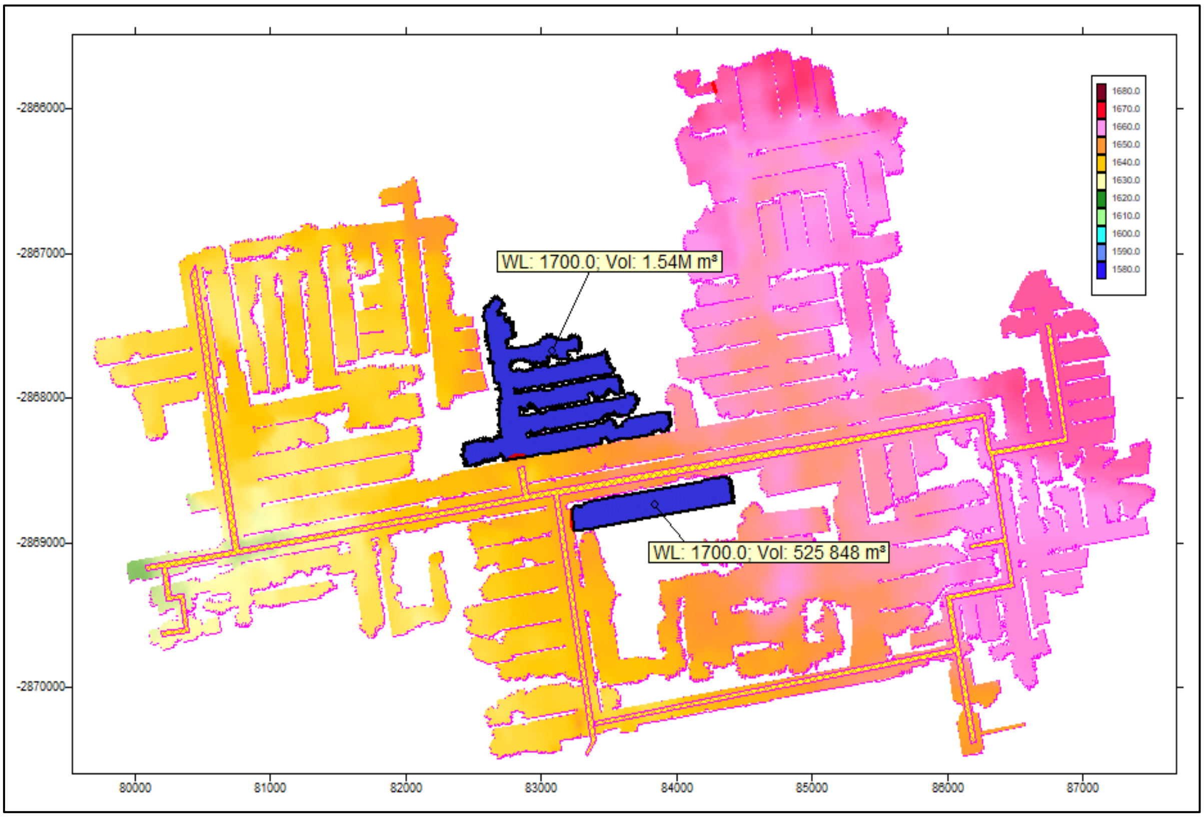


Figure 6-24: Underground mine with flooded compartments.

Stage-Volume curves for both compartments shows that the must be flooded to 1657 and 1658 mamsl respectively to be filled to capacity (Figure 6-25 and Figure 6-26). Both stage-volume curves use an extraction factor of 72.45% and the roof TIN as the confining layer. The roof TIN is a copy of the floor with the peg elevation used for the z-values. Because the roof TIN is a copy of the floor, the node and elements configuration is identical, saving time and eliminating a secondary interpolation of the roof elevations. Both compartments needed to be closed off by high-pressure seals to keep the water inside. But what happens when the seals break? Scenario 1 shows the flooding that will take place when the smaller of the two compartments fails.

The seal failure is simulated by using the original TIN, the one without the non-flow borders, and adding a volume of water to the TIN. For this, the TIN needs to be selected and WACCMAN initialised.

Move the move to the position of the compartment and select from the context-sensitive menu ***Calculate->Add Water***. A dialogue window will appear asking the volume of water to be added.

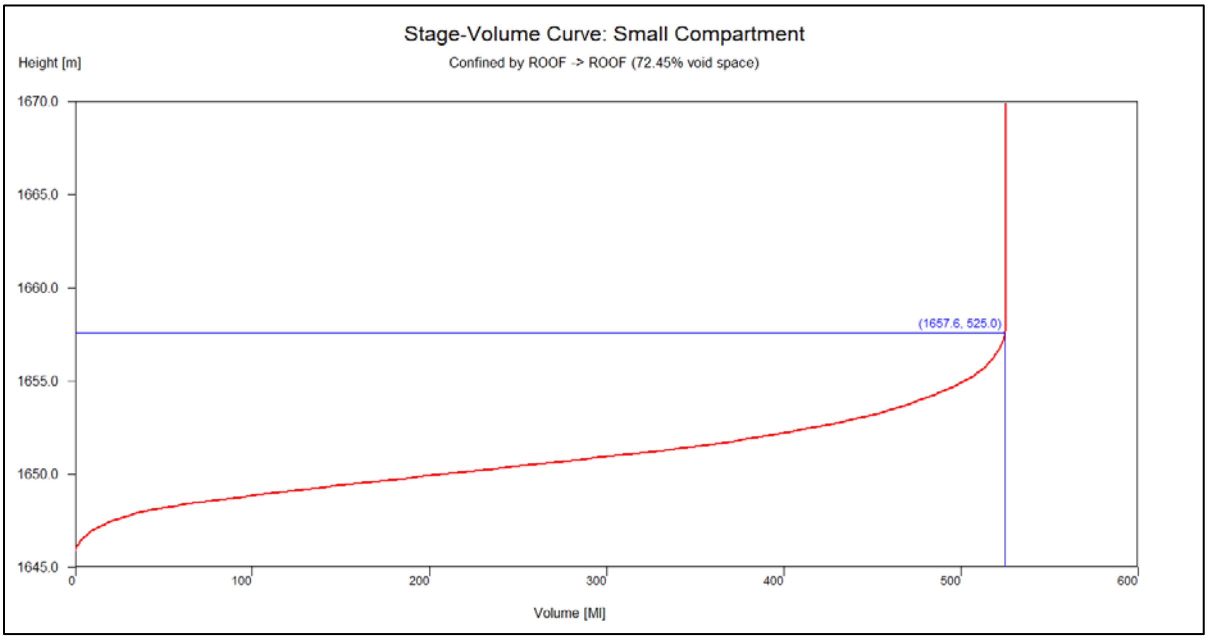


Figure 6-25: Stage curve for the small compartment.

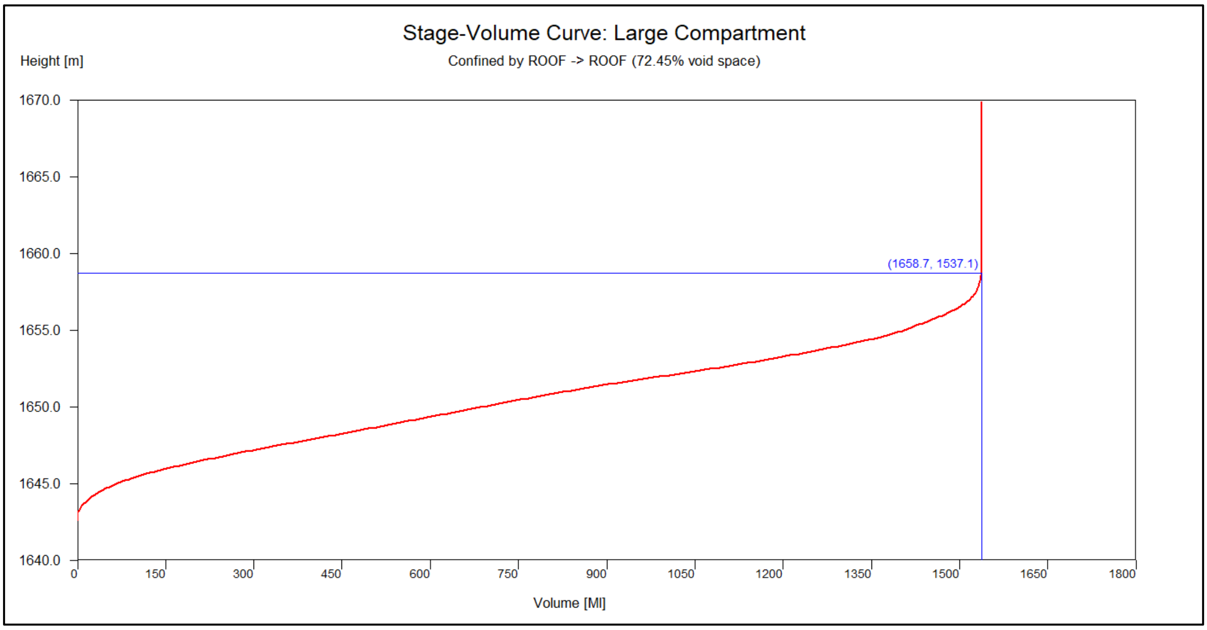


Figure 6-26: Stage-volume curve for the large compartment.

For scenario no 1, 500000 m^3 water was released in the near the back of the "compartment". The results are displayed in Figure 6-27. There are a total of seven water bodies varying in size between 8 000 and 171 000 with a total volume of 525 716 m^3, 132 m^3 short of the total released. The 132 is less than 0.03% and is probably contained in very small water bodies. By default, water bodies smaller than 10 m^3 are filtered from the view. The darker shade of blue in the water bodies indicates that the compartment is flooded to the roof. Figure 6-28 shows the same results but now in a 3D view. The top of the water bodies it represents the potential energy line of the water body.

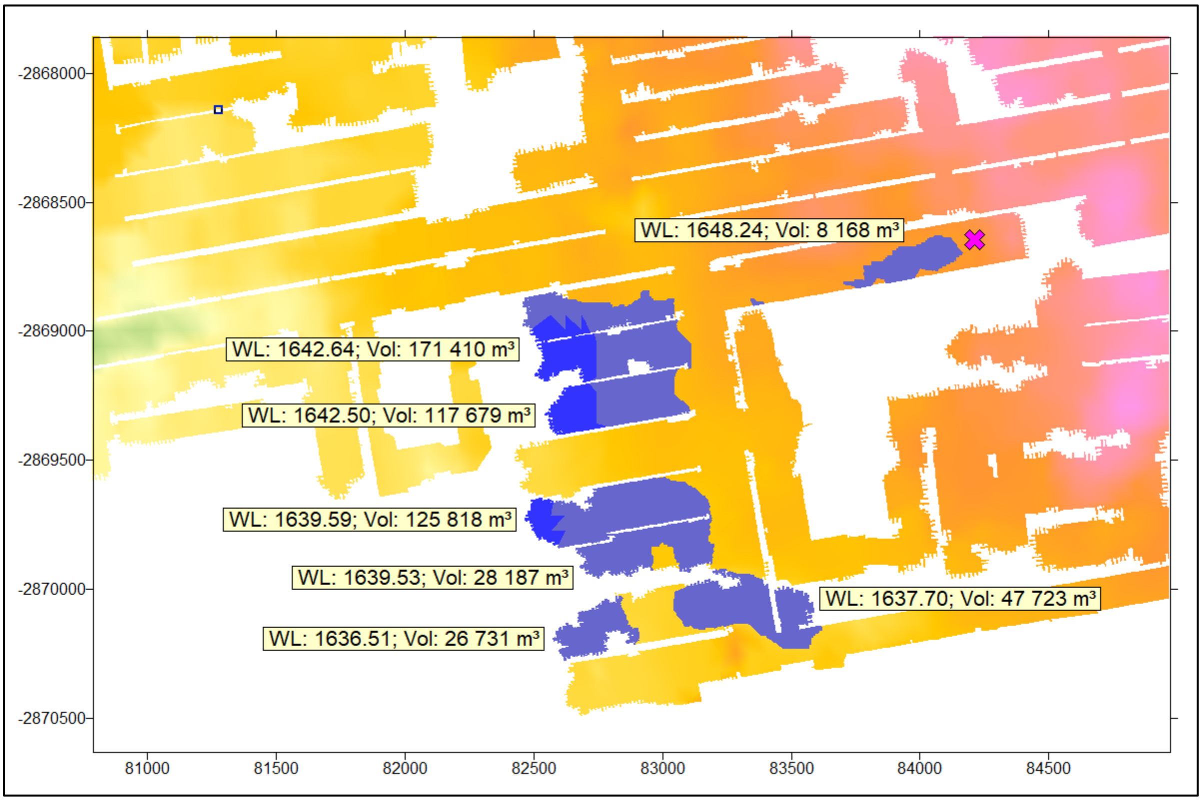


Figure 6-27: Water distribution after dropping 525 848 m³ drop position indicated with the magenta cross.

Figure 6-28: 3D View of water distribution after releasing 525 848 m³ water.

Scenario two is broken up in five stages to bring in some sort of time factor by visualising different volumes released at the same location. Figure 6-29 shows the water distribution after releasing 100K, 250K, 500K and 1000K m³ of water. Figure 6-30 shows the final stage after flooding with 1500K m³.

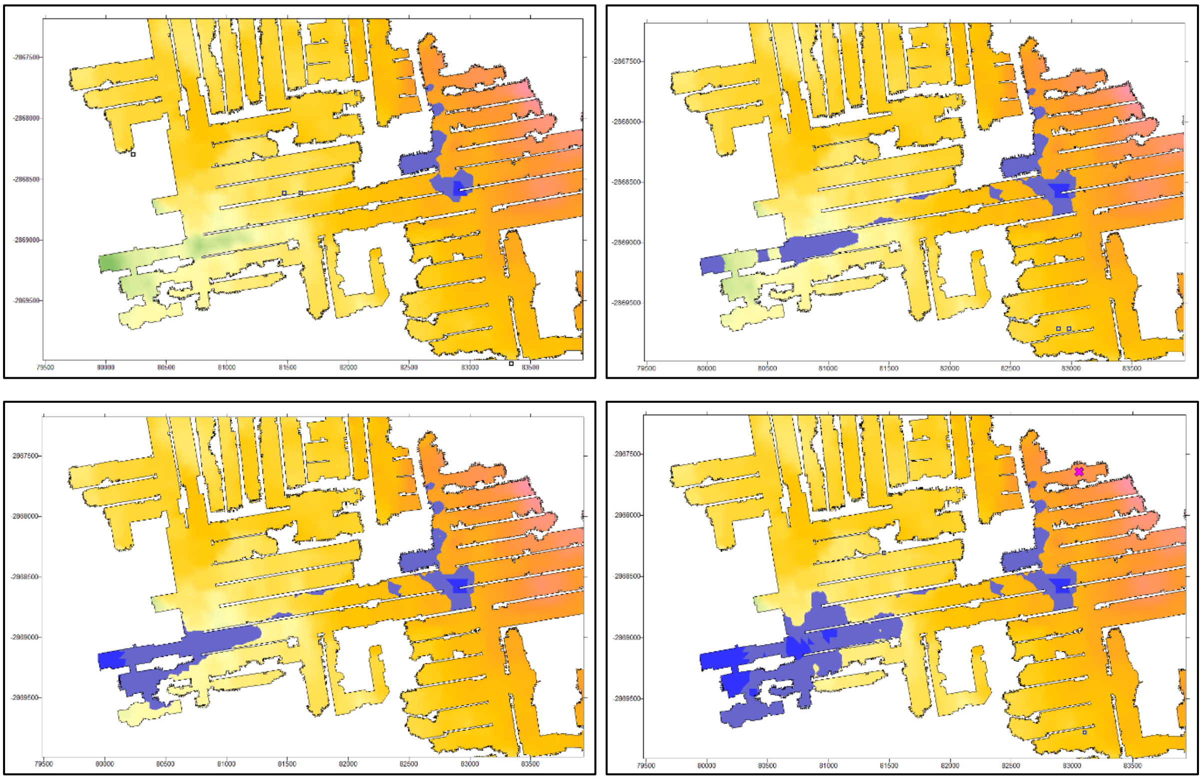


Figure 6-29: Water distribution after flooding with 100K, 250K, 500K and 1000K m³ water at the position indicated with the magenta cross.

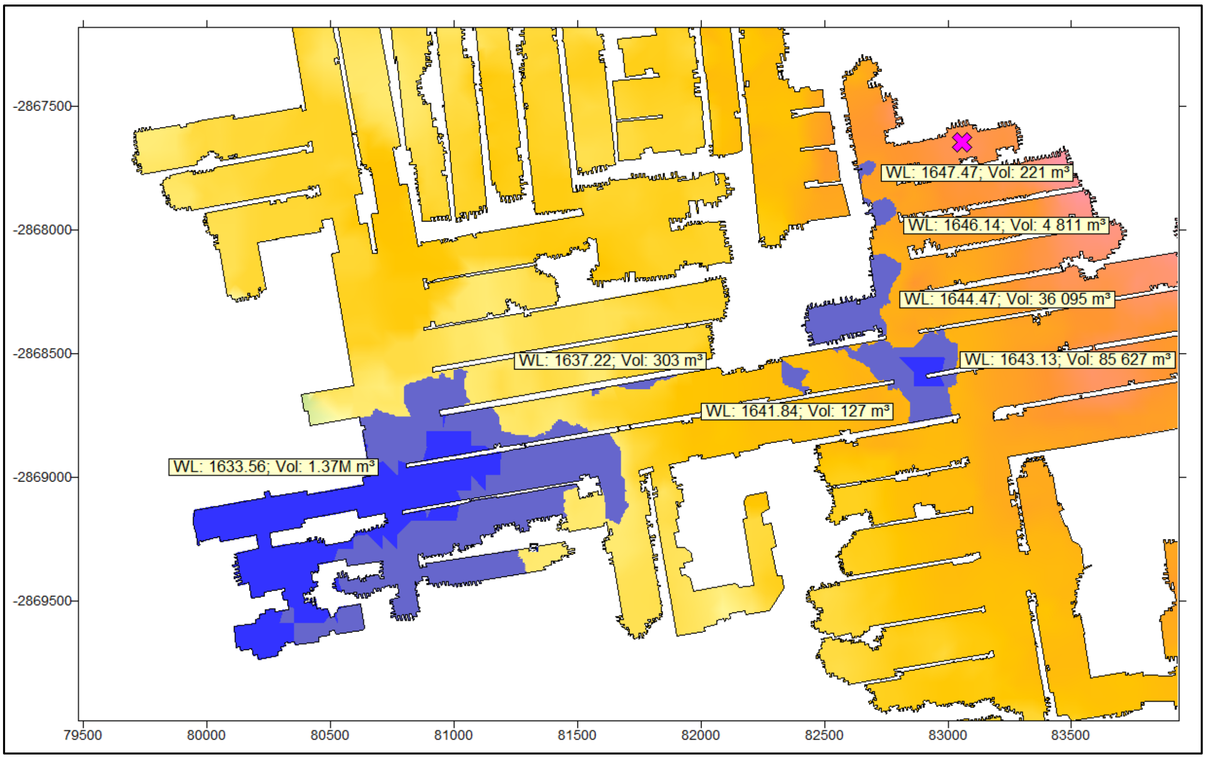


Figure 6-30: Water distribution after inserting 1 500 000 m³ at the magenta cross.

A 3-D view of the flooded parts of the underground working is displayed in Figure 6-31. The triangles are made visible to enhance the differences in elevation. Using the 3-D view, a streamline was (manually) created to indicate the route followed by the water on its way down to the deeper parts of the mine Figure 6-32.

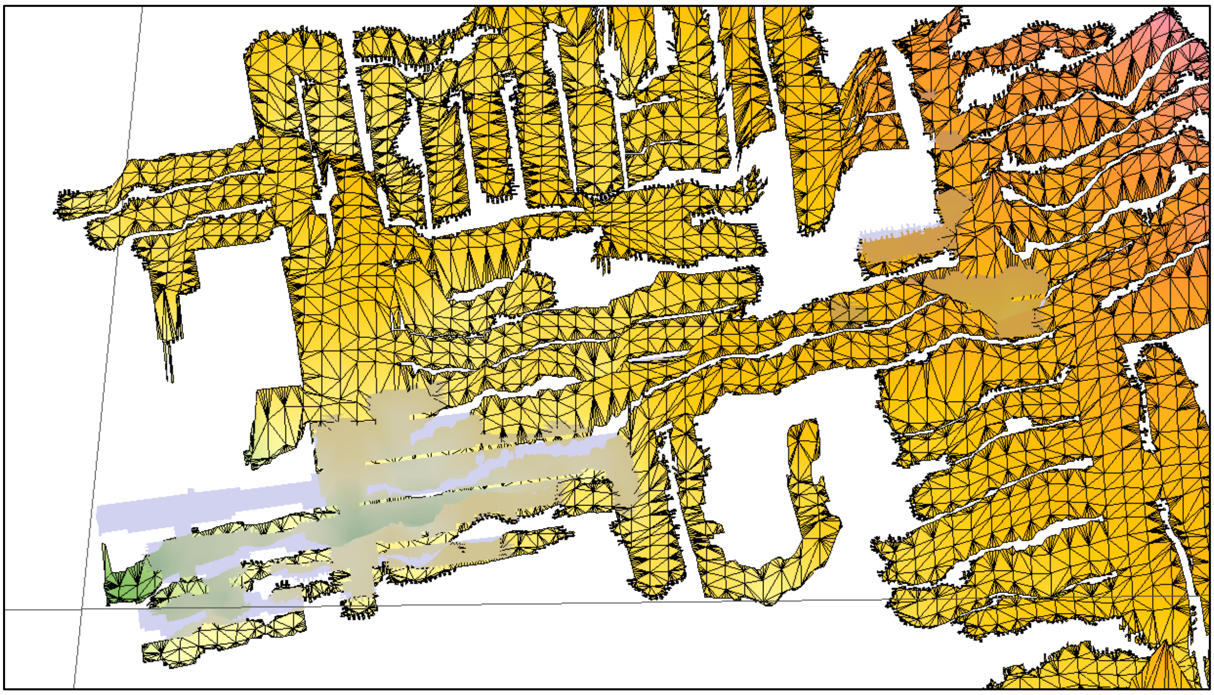


Figure 6-31: Water distribution in 3D.

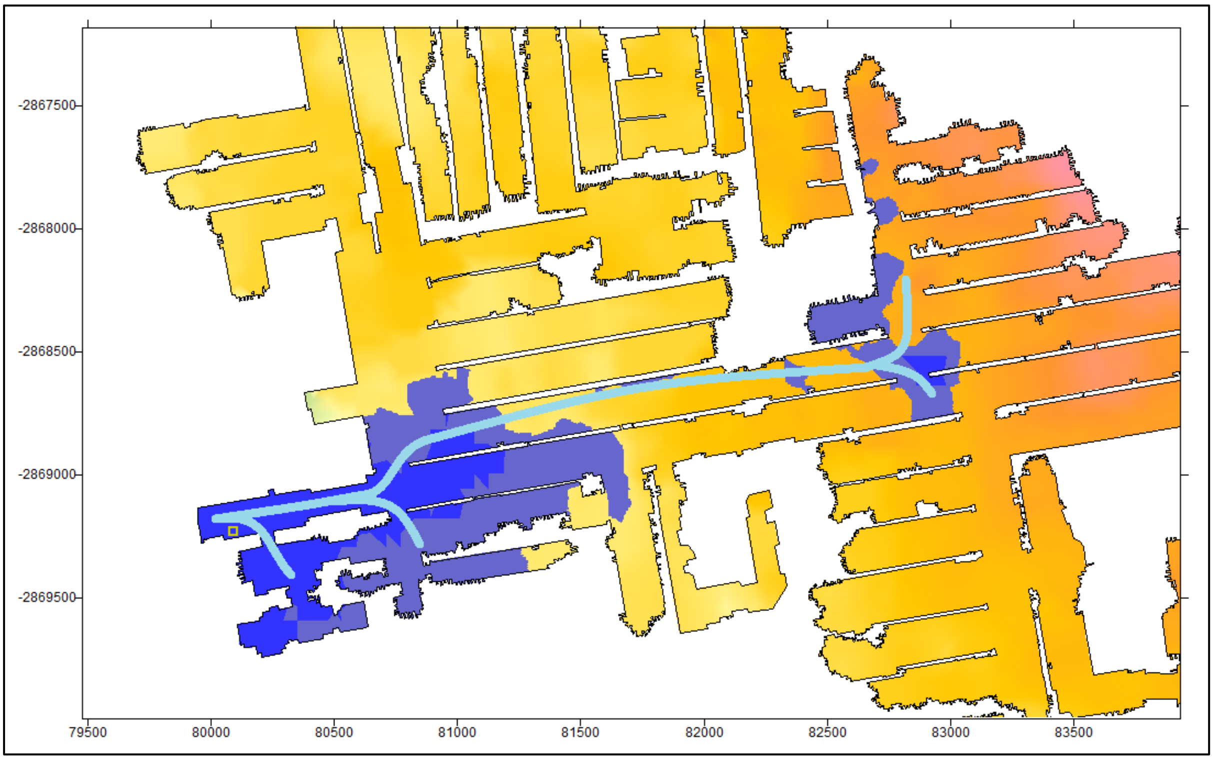


Figure 6-32: Water distribution with streamline.

7 Conclusions.

The work in the previous chapters showed a method and implementation of a flooding algorithm. Many flooding simulation programmes exist, but not one of them is geared for use in opencast and underground mines where spatially different recharge factors, different extraction rates and different confining heights may and will exist. Additionally, many underground mines are large in extent and have a maze of tunnels and a complicated outline.

Although the mine floor looks like a smooth surface, the elevation data indicates that the surface is not flat but that it consists of many depressions. The depressions are the spots where the water bodies will start to exist the moment recharge becomes visible. Each depression can also be viewed as a drainage region.

Traditional water-make calculations are performed over the entire extent of a mine. The calculations result in a single value of *water-make* for the combined workings. Just having a *water-make* value does not give a picture on the water distribution in a mine. It does not specify which compartments or parts of the compartments are flooded first. Changing the recharge volume calculation to a per depression (or drainage region) calculation established a distributed flooding method. The total amount of recharge water is still the same, but the places where the flooding starts are now scattered over the entire surface.

All the depressions are described as (possible) water bodies and placed in a list. There is no preference on which water body comes first. Starting with the first water body in the list, the adjacent water body with the lowest connection is determined. This connection is also the maximum water level elevation used to compute the water holding capacity of the depression. When the volume of recharge water is less than the capacity, the water body is filled available water. When the volume of recharge water is more than the capacity, the depression is filled to the maximum level and because water will now start to overflow all the excess recharge water will be added to the recharge water of the connected water body. The water added to the water body is subtracted from the recharge water. If the water level in the connected body is at the same level, the two water bodies are merged. The next water body in the list is selected, and the process starts over again. After all water bodies in the list are processed, we start over from the beginning of the list. Water bodies without any recharge water left are skipped. The simulation finishes when all recharge water is transferred into the waterbodies.

Although the method described above calculates the distribution for the entire workings, it can just as well be applied for a small part of the mine. It such a case only the small part will have recharge factors assigned while the remainder of the workings will have a zero (0) recharge rate. WISH calculates not only the water distribution in that part of the mine but also any recharge water spillage into the remainder of the mine. Knowledge about the volume of water spilt and where it flows is essential for the safety of the workers underground. When the spillage is little, the water will evaporate and be removed by the ventilation system. When the spillage becomes substantial the mine may decide to build a water retaining wall to keep the active workings dry.

Using the same flooding technique, it is also possible to simulate a calamity-flooding by “releasing” a large volume of water in a single water body and calculating the overflow into the adjacent water bodies. An example of a calamity-flooding can be the failure of a water-retaining wall.

Many mines in South Africa started their operations long before the digital era, mining data was recorded in logbooks and physical drawings; some (the older ones) even on cloth. Mines changed hands as they were sold or traded between mining houses, but crucial mining information was not always handed over. In later years the information was transferred to computer files, but even this is no guarantee that the data is complete and accurate.

When mining in close proximity of old mines, there is always the possibility that a barrier gets punctured. It is crucial to know how much water resides inside such a compartment to estimate the risk associated with a possible puncture. This new software makes it possible to calculate the flooding conditions in the workings without having the luxury to measure this physically.

The flooding simulation software relies heavily on the data available from the mine and the expertise of the geohydrologists with regard to the recharge rates. The new simulation software can be used not only during the planning stages but also during the development of the mine to determine the placing of transports belts and which roads can be used as escape routes in case of flooding and ultimately save equipment and lives.

Some last words

The development of WISH and WACCMAN has become my life's task. The development of the flooding part is not finished as there is always room for optimisation to bring down the memory requirements and the CPU time usage. Software that is no longer in development is stagnant. And stagnant software will grow old and obsolete in a very short time.

8 References

Annandale J.G., Jovanovic N.Z., Hodgson F.D.I., Usher B.H., Aken M.E., Van der Westhuizen A.M., Bristow K.L. and Steyn J.M., 2006. Prediction of the environmental impact and sustainability of large-scale irrigation with gypsiferous mine-water on groundwater resources. Water SA Vol. 32 No. 1

BARNES M.R. and VERMEULEN P.D., 2012. Guide to groundwater monitoring for the coal industry. Water SA Vol. 38 No. 5. DOI: 10.4314/wsa.v38i5.25

BRNICH M.J. and KOWALSKI K.M., No Date, Trakofker of the Office of Mine Safety and Health (a division of the CDC) Underground Coal Mine Disasters 1900 – 2010: Events, Responses, and a Look to the Future. https://www.cdc.gov/niosh/mining/userfiles/works/pdfs/ucmdn.pdf [Accessed 30/03/2020].

CNN Cable News Network, https://edition.cnn.com/2013/09/12/us/quecreek-mine-rescue-fast-facts/index.html [Accessed 30/03/2020].

DE BERG M., CHEONG O., KREVELD M., and OVERMARS M., 2008. Computational Geometry: Algorithms and Applications. Springer-Verlag.

DELAUNAY B., 1934. "Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk," pp. 793–800.

DENNIS R. and DENNIS I., 2016. Open pit mine flooding prognosis making use of analytical element modelling in fractured hard rock. Congress of the International Mine Water Association (IMWA), Freiberg, Germany.

DOLOZME P., 2019. Specific and Non-Specific Hazards in Underground Mines – 5 Hazards of Underground Mining. https://www.thebalancesmb.com/specific-and-non-specific-hazards-in-underground-mines-2367338 [Accessed 27/03/2020]

DONOVAN J.J. and PERRY E.F., (2019). Mine Flooding History of a Regional Below-Drainage Coalfield Dominated by Barrier Leakage (1970–2014), Hindawi Geofluids Volume 2019, Article ID 5703108. DOI: 10.1155/2019/5703108

DUNCAN R. (ed.), 1988. The MS-DOS Encyclopedia. Redmond: Microsoft Press

ENVIS, 2020. Centre on Environmental Problems in Mining, Http://ismenvis.nic.in/Database/Mahabir_Colliery_13111989_9459.aspx [Accessed 30/03/2020].

FLEWELLING S.A., & SHARMA M. (2014). Constraints on upward migration of hydraulic fracturing fluid and brine. Ground Water No.1 Vol. 52, 52(1), 9-19.

FOURIE F.D., (2015). Estimating the decant rate from a rehabilitated opencast colliery using a water balance approach. Mine Water and the Environment 35 199-213 (2016), Springer

The Guardian 2020, https://www.theguardian.com/world/2003/oct/29/russia [Accessed 30/03/2020].

HODGSON F.D.I. and KRANTZ R.M. (1995). Groundwater Quality Deterioration in the Olifants River Catchment above the Loskop Dam with Specialized Investigation in the Witbank Dam Sub-Catchment.-WRC Report No.-291/1/95.

HODGSON F.D.I. and LUKAS E., 2011. Mine-water balances for Sigma Colliery – report to SASOL.

HORTON R.E., 1933. The Rôle of infiltration in the hydrologic cycle, Transactions American Geophysical Union

KERNIGHAN B.W. and RITCHIE D.M., 1988. The C Programming Language, Prentice-Hall.

LAAKSONEN, A, 2018, Guide to competitive programming – Learning and improving algorithms through contests, Springer International. DOI: 10.1007/978-3-319-72547-5

LUKAS E., 2012. WISH as a water management tool in opencast and underground collieries, MSc dissertation, University of the Free State.

LUKAS E. and VERMEULEN D., 2016. Optimum Colliery: Update of the Mine-water Balance, Report No 2016/01/PDV.

LUKAS E., 2018. Union Mine Volumetric Calculations – report to Union Mine.

LUKAS E., 2019. Recharge in Rehabilitated Opencast Collieries, COALTECH Report.

Ma M., Wen L., Hao S., Zhao G., Zhou M., Liu C., Wang H. and Wang Z. (2020) 'A grid-based distributed hydrological model for coal mined-out area', Journal of Hydrology, 588, pp. 124990.

Marinelli F. and Niccoli W.L. (2000). Simple analytical equations for estimating groundwater inflow to a mine pit. Ground Water, Vol 2, No 3, 311 - 314

Mine Accidents and Disasters, 2020. http://www.mineaccidents.com.au/mine-events/all/au [Assessed 30/03/2020].

NIMMO J.R., HEALY R.W., and STONESTROM D.A., 2005, Aquifer Recharge, in ANDERSON MG and BEAR J, eds., Encyclopedia of Hydrological Science: Part 13. Groundwater: Chichester, UK, Wiley, v. 4, p. 2229-2246, doi:10.1002/0470848944.hsa161a.

RAPANTOVA N., GRMELA A., VOJTEK D., HALIR J. and MICHALEK B., (2007). Groundwater flow modelling applications in mining hydrogeology. Mine Water and the Environment, Springer.

Reuters 2020, Trapped Chinese miners rescued after 3 days, https://www.reuters.com/article/us-china-miners/trapped-chinese-miners-rescued-after-3-days-idUSPEK34233420070801 [Accessed 30/03/2020].

SAHO, 2020. More than 400 miners killed in what was the worst mining disaster in South Africa, South Africa History Online. https://www.sahistory.org.za/dated-event/more-400-miners-killed-what-was-worst-mining-disaster-south-africa. [Accessed 30/03/2020].

SAHOO R. and SAHOO G., 2016, Computer science with C++, New Saraswati House New Delhi (India) Pvt. Ltd]

SZCZEPINSKI J., (2019). The Significance of Groundwater Flow Modeling Study for Simulation of Opencast Mine Dewatering, Flooding, and the Environmental Impact. Water 2019, 11, 848; doi:10.3390/w11040848

SCHULTZE M., 2012. The filling and remediation of pit lakes in former open cast lignite mines, (Dissertation) Technischen Universitat Carolo-Wilhelmina zu Braunschweig, Germany.

SCOTT R., 1995. Flooding of central and east Rand Gold Mines, An investigation into controls over the inflow rate, water quality and the predicted impacts of flooded mines. Report to the Water reseach Commission, South Africa. Report No 486/1/95

SEYMOUR C, 2005. Mining Disasters – What lessons can be learnt. Queensland Mining Industry Safety and Health Conference Proceedings.

SINGH R.N., DENBY B. and REED M., 1985. The effect of groundwater re-establishment on the settlement of opencast mine backfills in the United Kingdom. Second International Congress of the International Mine Water Association (IMWA), Granada, Spain, September 1985.

South China Morning Post 2020, Flood at Chinese coal mine kills four, leaves 14 trapped, https://www.scmp.com/news/china/society/article/3042164/flood-chinese-coal-mine-kills-four-leaves-14-trapped [Accessed 30/03/2020].

STROUSTRUP B., 1997. The C++ Programming Language (Third ed.). Addison-Wesley Publishing Company, Reading, Massachusetts.

TANNER P., 2007. Guidelines for the Rehabilitation of Mined Land. Coaltech Research Association / Chamber of Mines of South Africa

TANNER P., 2019. Land Rehabilitation Guidelines for Surface Coal Mines. Land Rehabilitation Society of Southern Africa, Coaltech, Minerals Council of South Africa

The history of computing project, 2020. Clipper 1984 USA. http://www.thocp.net/software/languages/clipper.htm [Accessed 30/03/2020].

The Mineral Council of South Africa, 2020. https://www.mineralscouncil.org.za/industry-news/we-care-we-remember/324-we-remember-merriespruit [Accessed 30/03/2020].

Tonder G.J. van, Usher B.H., Dennis I., Vermeulen P.D., 2007. Predicting rebound in a deep colliery in South Africa. Mine Water and the Environment, Springer-Verlag. 26 (2), 79–87.

US MINE RESCUE ASSOCIATION, 2020, Mine Disasters in the United States, https://usminedisasters.miningquiz.com/saxsewell/portertunnel.htm [Accessed 30/03/2020].

VERMEULEN D, LUKAS E and STEYL G, 2013. Eikeboom Mine: Groundwater Assessment for Pillar Extraction. Mine Water Consultants, Report No 2013/05/PDV.

VERMEULEN P.D. and USHER B.H, 2006a. Sulphate generation in South African underground and opencast collieries. Environmental Geology, Springer-Verlag. DOI: 10.1007/s00254-005-0091-2.

VERMEULEN P.D. and USHER B.H, 2006b. An investigation into recharge in South African underground colieries. The journal of the Southern African Institute of Mining and Metallurgy.

Wilkens S.E., and van Niekerk A.M., 1993. Opencast mine water simulation modelling. In: Proceedings of the International Mine Water Association Symposium, Chililabombwe, Zambia, September 1993

WILSON M.G.C. and ANHAEUSSER C.R. (edd), 1998, The Mineral Resources of South Africa. 6th ed. Council for Geoscience, Silverton, South Africa.

XU C. and GONG P., 2011. Water disaster types and water control measures of Hanxing coal mine area. Paper presented at the 2011 Xi'an International conference on fine geological exploration and groundwater & gas hazards control in coal mines. Elsevier Ltd. DOI: 10.1016/j.proeps.2011.09.104

9 APPENDIX

The appendix consists of five tables containing the recharge status data from case study 1, after every completed cycle. It depicts the progress the recharge subroutine has made after completing iterating through the list of water bodies. The output is generated only in debug mode.

Tables in the appendix:

1. Rainfall 250 mm
2. Rainfall 500 mm
3. Rainfall 1000 mm
4. Rainfall 2000 mm
5. Rainfall 3000 mm

Rainfall 250 mm (WISH output collected in Debug mode).

Cycle	Expected Recharge [m³]	Simulated Recharge [m³]	Difference [m³]	Difference [%]	# WB	Iterations
1	180443.8	60112.7	120331.1	66.7	3002	3002
2	180443.8	96757.0	83686.7	46.4	2647	3938
3	180443.8	107043.3	73400.5	40.7	2512	4398
4	180443.8	121625.2	58818.5	32.6	2415	4690
5	180443.8	133419.2	47024.6	26.1	2353	4871
6	180443.8	138992.6	41451.2	23.0	2299	5026
7	180443.8	148993.6	31450.2	17.4	2259	5130
8	180443.8	152561.4	27882.3	15.5	2229	5202
9	180443.8	155974.3	24469.5	13.6	2207	5253
10	180443.8	159143.1	21300.7	11.8	2195	5291
11	180443.8	162926.3	17517.4	9.7	2183	5324
12	180443.8	169268.2	11175.6	6.2	2170	5350
13	180443.8	170387.2	10056.5	5.6	2166	5362
14	180443.8	170839.0	9604.8	5.3	2161	5370
15	180443.8	175040.4	5403.4	3.0	2158	5376
16	180443.8	175423.6	5020.2	2.8	2155	5381
17	180443.8	176439.3	4004.4	2.2	2153	5385
18	180443.8	176797.6	3646.1	2.0	2151	5390
19	180443.8	176797.6	3646.1	2.0	2151	5392
20	180443.8	176861.5	3582.3	2.0	2150	5395
21	180443.8	177887.8	2555.9	1.4	2149	5398
22	180443.8	178924.1	1519.7	0.8	2148	5400
23	180443.8	180470.8	-27.0	-0.0	2147	5401

Rainfall 500 mm (WISH output collected in Debug mode).

Cycle	Expected Recharge [m³]	Simulated Recharge [m³]	Difference [m³]	Difference [%]	# WB	Iterations
1	362710.7	85321.1	277389.7	76.5	3002	3002
2	362710.7	148563.4	214147.3	59.0	2606	3960
3	362710.7	174214.0	188496.7	52.0	2453	4456
4	362710.7	206775.0	155935.7	43.0	2338	4768
5	362710.7	226813.9	135896.9	37.5	2260	4986
6	362710.7	239708.8	123002.0	33.9	2197	5163
7	362710.7	274848.8	87861.9	24.2	2154	5286
8	362710.7	283103.8	79606.9	21.9	2116	5379
9	362710.7	291577.0	71133.7	19.6	2086	5452
10	362710.7	297650.0	65060.7	17.9	2068	5501
11	362710.7	305402.4	57308.3	15.8	2050	5539
12	362710.7	310925.6	51785.1	14.3	2037	5567
13	362710.7	323243.4	39467.3	10.9	2029	5587
14	362710.7	332269.0	30441.7	8.4	2020	5600
15	362710.7	337369.4	25341.4	7.0	2016	5610
16	362710.7	338494.6	24216.2	6.7	2011	5619
17	362710.7	341588.6	21122.2	5.8	2007	5629
18	362710.7	345567.0	17143.7	4.7	2004	5635
19	362710.7	347778.3	14932.5	4.1	2002	5639
20	362710.7	351251.1	11459.7	3.2	2001	5644
21	362710.7	352979.0	9731.8	2.7	1998	5649
22	362710.7	355236.7	7474.0	2.1	1996	5652
23	362710.7	358004.3	4706.5	1.3	1995	5654
24	362710.7	361490.7	1220.1	0.3	1994	5656

25	362710.7	362690.5	20.2	0.0	1993	5657

Rainfall 1000 mm (WISH output collected in Debug mode).

Cycle	Expected Recharge [m³]	Simulated Recharge [m³]	Difference [m³]	Difference [%]	# WB	Iterations
1	725421.5	117525.1	607896.4	83.8	3002	3002
2	725421.5	225786.4	499635.1	68.9	2575	4003
3	725421.5	269218.7	456202.7	62.9	2409	4515
4	725421.5	320784.6	404636.8	55.8	2283	4857
5	725421.5	372588.2	352833.2	48.6	2196	5096
6	725421.5	409008.3	316413.2	43.6	2130	5287
7	725421.5	437069.7	288351.8	39.7	2080	5425
8	725421.5	450766.3	274655.2	37.9	2039	5533
9	725421.5	493428.4	231993.1	32.0	1998	5630
10	725421.5	512342.0	213079.4	29.4	1974	5695
11	725421.5	521701.8	203719.7	28.1	1949	5754
12	725421.5	539553.1	185868.4	25.6	1924	5803
13	725421.5	568554.6	156866.9	21.6	1908	5838
14	725421.5	589135.5	136286.0	18.8	1896	5862
15	725421.5	595532.8	129888.6	17.9	1886	5885
16	725421.5	597400.0	128021.5	17.6	1879	5902
17	725421.5	608646.1	116775.4	16.1	1871	5915
18	725421.5	619790.4	105631.1	14.6	1868	5923
19	725421.5	620293.7	105127.8	14.5	1866	5928
20	725421.5	640399.5	85021.9	11.7	1864	5935
21	725421.5	645984.7	79436.8	11.0	1861	5944
22	725421.5	655553.0	69868.5	9.6	1858	5949
23	725421.5	667802.3	57619.2	7.9	1856	5954
24	725421.5	671586.4	53835.1	7.4	1853	5960

25	725421.5	694289.8	31131.7	4.3	1850	5964
26	725421.5	694487.0	30934.5	4.3	1849	5968
27	725421.5	718605.6	6815.9	0.9	1846	5974
28	725421.5	724537.4	884.1	0.1	1846	5977
29	725421.5	724537.4	884.1	0.1	1846	5979
30	725421.5	724537.4	884.1	0.1	1846	5980
31	725421.5	724537.4	884.1	0.1	1846	5981
32	725421.5	725572.7	-151.2	-0.0	1846	5982

Rainfall 2000 mm (WISH output collected in Debug mode).

Cycle	Expected Recharge [m³]	Simulated Recharge [m³]	Difference [m³]	Difference [%]	# WB	Iterations
1	1450843.0	152167.3	1298675.7	89.5	3002	3002
2	1450843.0	330700.5	1120142.4	77.2	2552	4019
3	1450843.0	406486.8	1044356.2	72.0	2379	4563
4	1450843.0	504474.1	946368.8	65.2	2248	4930
5	1450843.0	600284.7	850558.2	58.6	2152	5210
6	1450843.0	653321.2	797521.7	55.0	2072	5436
7	1450843.0	702024.1	748818.9	51.6	2017	5592
8	1450843.0	738691.2	712151.7	49.1	1967	5717
9	1450843.0	806051.9	644791.1	44.4	1924	5818
10	1450843.0	845544.7	605298.2	41.7	1898	5901
11	1450843.0	873214.9	577628.1	39.8	1864	5975
12	1450843.0	904437.4	546405.6	37.7	1836	6047
13	1450843.0	971234.3	479608.7	33.1	1813	6111
14	1450843.0	1024368.1	426474.8	29.4	1793	6163
15	1450843.0	1043992.5	406850.5	28.0	1780	6201
16	1450843.0	1061324.7	389518.3	26.8	1766	6232
17	1450843.0	1077562.9	373280.0	25.7	1758	6255
18	1450843.0	1113409.7	337433.3	23.3	1749	6283
19	1450843.0	1118139.3	332703.7	22.9	1747	6294
20	1450843.0	1126797.6	324045.4	22.3	1741	6310
21	1450843.0	1138901.3	311941.6	21.5	1736	6321
22	1450843.0	1141658.8	309184.2	21.3	1733	6330
23	1450843.0	1157896.8	292946.1	20.2	1729	6349
24	1450843.0	1172630.5	278212.4	19.2	1725	6358

25	1450843.0	1200332.9	250510.1	17.3	1721	6365
26	1450843.0	1273849.1	176993.9	12.2	1719	6372
27	1450843.0	1277496.1	173346.8	11.9	1717	6382
28	1450843.0	1329602.2	121240.8	8.4	1715	6388
29	1450843.0	1329682.5	121160.5	8.4	1714	6393
30	1450843.0	1337572.7	113270.2	7.8	1712	6399
31	1450843.0	1347772.2	103070.8	7.1	1711	6405
32	1450843.0	1359089.7	91753.3	6.3	1708	6414
33	1450843.0	1375995.4	74847.5	5.2	1706	6418
34	1450843.0	1375995.4	74847.5	5.2	1706	6422
35	1450843.0	1376142.8	74700.2	5.1	1705	6436
36	1450843.0	1414823.3	36019.7	2.5	1704	6441
37	1450843.0	1415263.0	35580.0	2.5	1704	6447
38	1450843.0	1417001.8	33841.1	2.3	1703	6451
39	1450843.0	1417080.6	33762.4	2.3	1702	6454
40	1450843.0	1417080.6	33762.4	2.3	1702	6458
41	1450843.0	1417448.9	33394.1	2.3	1701	6462
42	1450843.0	1418115.4	32727.6	2.3	1700	6467
43	1450843.0	1418115.4	32727.6	2.3	1700	6469
44	1450843.0	1418115.4	32727.6	2.3	1700	6471
45	1450843.0	1418115.4	32727.6	2.3	1700	6482
46	1450843.0	1418115.4	32727.6	2.3	1700	6484
47	1450843.0	1418187.9	32655.1	2.3	1700	6487
48	1450843.0	1418187.9	32655.1	2.3	1700	6489
49	1450843.0	1419007.2	31835.8	2.2	1699	6492
50	1450843.0	1419552.3	31290.6	2.2	1698	6494

51	1450843.0	1419552.3	31290.6	2.2	1698	6496
52	1450843.0	1419715.9	31127.1	2.1	1697	6499
53	1450843.0	1419821.6	31021.4	2.1	1696	6501
54	1450843.0	1419821.6	31021.4	2.1	1696	6503
55	1450843.0	1420579.4	30263.6	2.1	1695	6505
56	1450843.0	1420579.4	30263.6	2.1	1695	6507
57	1450843.0	1429263.3	21579.6	1.5	1694	6509
58	1450843.0	1429263.3	21579.6	1.5	1694	6511
59	1450843.0	1432396.7	18446.3	1.3	1693	6513
60	1450843.0	1432396.7	18446.3	1.3	1693	6515
61	1450843.0	1435780.6	15062.3	1.0	1692	6517
62	1450843.0	1435780.6	15062.3	1.0	1692	6519
63	1450843.0	1438773.0	12070.0	0.8	1691	6521
64	1450843.0	1438773.0	12070.0	0.8	1691	6523
65	1450843.0	1438800.0	12043.0	0.8	1690	6525
66	1450843.0	1438800.0	12043.0	0.8	1690	6527
67	1450843.0	1442272.7	8570.3	0.6	1689	6530
68	1450843.0	1450561.4	281.5	0.0	1688	6532
69	1450843.0	1450561.4	281.5	0.0	1688	6533
70	1450843.0	1450561.4	281.5	0.0	1688	6534
71	1450843.0	1450561.4	281.5	0.0	1688	6535
72	1450843.0	1450561.4	281.5	0.0	1688	6536

Rainfall 3000 mm (WISH output collected in Debug mode).

Cycle	Expected Recharge [m³]	Simulated Recharge [m³]	Difference [m³]	Difference [%]	# WB	Iterations
1	2176264.5	169455.2	2006809.2	92.2	3002	3002
2	2176264.5	400146.3	1776118.2	81.6	2546	4026
3	2176264.5	496149.4	1680115.1	77.2	2371	4572
4	2176264.5	617149.4	1559115.1	71.6	2239	4944
5	2176264.5	727877.9	1448386.5	66.6	2142	5225
6	2176264.5	830385.3	1345879.2	61.8	2057	5462
7	2176264.5	886185.7	1290078.8	59.3	1997	5627
8	2176264.5	953406.0	1222858.4	56.2	1946	5766
9	2176264.5	1053447.7	1122816.7	51.6	1897	5890
10	2176264.5	1159037.8	1017226.7	46.7	1865	5992
11	2176264.5	1189023.6	987240.9	45.4	1827	6085
12	2176264.5	1229457.6	946806.8	43.5	1798	6166
13	2176264.5	1285966.9	890297.6	40.9	1772	6234
14	2176264.5	1347705.7	828558.7	38.1	1750	6292
15	2176264.5	1365144.2	811120.3	37.3	1733	6337
16	2176264.5	1418897.1	757367.3	34.8	1715	6378
17	2176264.5	1460445.1	715819.3	32.9	1704	6417
18	2176264.5	1502570.1	673694.3	31.0	1692	6451
19	2176264.5	1518170.1	658094.3	30.2	1685	6468
20	2176264.5	1572102.2	604162.2	27.8	1677	6488
21	2176264.5	1585867.9	590396.5	27.1	1670	6502
22	2176264.5	1590258.9	586005.6	26.9	1665	6515
23	2176264.5	1597037.4	579227.0	26.6	1659	6537
24	2176264.5	1619622.6	556641.8	25.6	1654	6547

25	2176264.5	1636202.3	540062.1	24.8	1650	6557
26	2176264.5	1713949.0	462315.5	21.2	1646	6567
27	2176264.5	1722697.8	453566.6	20.8	1644	6581
28	2176264.5	1729072.4	447192.1	20.5	1642	6588
29	2176264.5	1747160.4	429104.0	19.7	1640	6596
30	2176264.5	1752283.1	423981.4	19.5	1636	6603
31	2176264.5	1752283.1	423981.4	19.5	1636	6608
32	2176264.5	1757925.3	418339.1	19.2	1633	6618
33	2176264.5	1760390.4	415874.1	19.1	1631	6623
34	2176264.5	1760390.4	415874.1	19.1	1631	6628
35	2176264.5	1761295.5	414968.9	19.1	1629	6644
36	2176264.5	1761827.9	414436.5	19.0	1628	6649
37	2176264.5	1798108.4	378156.0	17.4	1627	6655
38	2176264.5	1810221.5	366042.9	16.8	1626	6659
39	2176264.5	1813453.8	362810.7	16.7	1625	6663
40	2176264.5	1865906.0	310358.5	14.3	1625	6668
41	2176264.5	1923380.1	252884.4	11.6	1623	6673
42	2176264.5	1923510.6	252753.9	11.6	1622	6679
43	2176264.5	1926502.9	249761.6	11.5	1621	6683
44	2176264.5	1937591.1	238673.3	11.0	1620	6687
45	2176264.5	1937618.2	238646.3	11.0	1619	6700
46	2176264.5	1939869.5	236395.0	10.9	1618	6705
47	2176264.5	1946245.1	230019.4	10.6	1616	6710
48	2176264.5	1962999.5	213265.0	9.8	1614	6714
49	2176264.5	1969917.0	206347.4	9.5	1613	6718
50	2176264.5	1971821.5	204443.0	9.4	1611	6723

51	2176264.5	2003430.9	172833.6	7.9	1609	6728
52	2176264.5	2021566.0	154698.4	7.1	1607	6731
53	2176264.5	2021566.0	154698.4	7.1	1607	6734
54	2176264.5	2061846.1	114418.4	5.3	1605	6738
55	2176264.5	2089024.6	87239.9	4.0	1604	6742
56	2176264.5	2124002.8	52261.7	2.4	1602	6745
57	2176264.5	2124011.8	52252.7	2.4	1602	6748
58	2176264.5	2137893.4	38371.1	1.8	1600	6753
59	2176264.5	2152243.2	24021.2	1.1	1598	6758
60	2176264.5	2166237.4	10027.0	0.5	1596	6761
61	2176264.5	2166237.4	10027.0	0.5	1596	6764
62	2176264.5	2175968.7	295.7	0.0	1594	6767
63	2176264.5	2175968.7	295.7	0.0	1594	6768
64	2176264.5	2175968.7	295.7	0.0	1594	6769
65	2176264.5	2175968.7	295.7	0.0	1594	6770
66	2176264.5	2175968.7	295.7	0.0	1594	6771