

---

# *Estimation and automation process of hydrological quantities in the Netherlands*

BMI – paper Daphne van Leeuwen

Augustus 2011

---



## Preface

This paper is part of the master program Business Mathematics and Informatics. Business Mathematics and Informatics is a multidisciplinary study which is aimed at optimization of business processes by using methods out of these three disciplines. Since I want to do a second master in Meteorology I wanted the subject to be a combination of these two studies. Another important aspect of writing a paper like this is the relevance. If you can work on something that is useful for others it gives more motivation to work hard to achieve this goal. These are the arguments where I based my choice on.

I had some conversations with the company FutureWater, a company that is specialized in hydrological processes. They had an interesting assignment for me to work on. They developed a model, called the HydroS model to map the soil moisture and evaporation in the Netherlands. They used the daily precipitation and evaporation data of one station as input. Intergrated this in the model, then this model produced several maps as output. These were maps like an estimation of soil moisture, actual evaporation, relative evaporation.

The assignment for me was improving this model, by means of the Mathematics and Informatics aspects of my study. The first aspect that I had to add to this model was the use of all the meteo stations with their daily data for precipitation and evaporation instead of one station where the model was based on. This was implemented by means of interpolation techniques, the mathematical part. The second one is automating the process of retrieving the daily input data for the model and generating the output automatically on a website, the informatics part.

I would like to thank Peter Droogers and Wilco Terink for giving me the oppurtunity to work on this interesting assignment and their help for my lack of knowledge about the hydrological processes. I would also like to thank my supervisor Fetsje Bijma for her help and support for writing this paper.

## Management summary

This paper describes the estimation and automation process for the soil moisture and fractional evaporation in the Netherlands using an extension of the so called HydroS model. The company FutureWater has developed the HydroS model for the estimation of the soil moisture and fractional evaporation to model floods and water shortage in the Netherlands. In its original state the HydroS model needs input in a certain format of the daily evaporation and precipitation, which is based on one input meteo station, named the Bilt. Then it generates an estimation of the soil moisture and fractional evaporation. This model can be improved in two ways. First, data from more than one meteo station should be taken into account, and second, the estimation process should be automated. In the paper these two steps are described in reversed order, because this was a more logical structure.

The first step was to use multiple meteo stations as input instead of one. These input data could be downloaded from the KNMI website. For the precipitation there were two kinds of input available. The first one was data from 35 automated meteo stations. These record the daily precipitation (0 - 24 hour) and are directly available online. The second option was data from 326 voluntary rain stations. These record the precipitation from 8 AM the previous day till 8 AM the day indicated. A disadvantage of these stations is that the data are not directly available, which makes it useless for up to date approximation. Nevertheless, we used the 326 stations data as a control for the 35 stations data, and could confirm that the 35 stations data are sufficient.

When using the data from multiple stations, there is still area in between the stations for which the values are unknown. For these points a spatial interpolation technique must be used to estimate the value. We used two interpolation techniques in this paper, Inverse Distance Weighted (IDW) interpolation and Ordinary Kriging. IDW estimates the value in a point by a weighted average of the values in points nearby. The weights are inversely proportional to their distance from the point to estimate. Kriging also makes use of weighted averages, giving higher weights to points nearby. The difference, however, is that kriging takes into account the spatial dependence between points. For ordinary kriging the spatial dependence is modeled by a variogram, which must be fitted to the data. When we compare the output maps from both techniques we concluded that kriging outperforms IDW.

The second step was to automate the process. By this we mean that the model can run by pressing only one button. For this we used various programs. When we now run the model, we automatically download the data from the selected period from the KNMI website. These data are converted to the right format for the interpolation. Then result maps from the interpolation output serve as input for the HydroS model. This is then ran for the period indicated and output maps are generated.

The output maps from the HydroS model that were interesting for us were the estimations of the soil moisture and relative evaporation. Although the difference between the two interpolation techniques was smaller than in the precipitation maps, kriging still outperforms IDW. Moreover, there was less difference between the maps based on the 35 or 326 stations data. Hence, we concluded that the 35 stations data suffice for the moisture and evaporation maps.

## Content

BMI – paper Daphne van Leeuwen.....	1
Augustus 2011.....	1
Preface .....	3
Management summary.....	4
Content .....	5
1 Introduction .....	7
2 Methods.....	9
2.1 Implementation of the estimation process .....	9
2.1.1 Preproces of the data.....	9
2.1.2 Daily data .....	10
2.1.3 Interpolating the spatial data points .....	10
2.1.4 The Simulation .....	10
2.2 Tools for implementation .....	11
2.2.1 PCraster.....	11
2.2.2 Python .....	11
2.2.3 GNU Wget .....	11
2.2.4 Gstat.....	11
2.3 Spatial Interpolation techniques.....	11
2.3.1 Deterministic methods.....	11
2.3.2 Kriging .....	12
2.4 Interpolation techniques used for the problem .....	13
2.4.1 Inverse distance weighted interpolation to a power.....	13
2.4.2 Ordinary Kriging .....	14
2.4.3 Variogram .....	17
3 Results.....	20
3.1 Inverse distance weighted interpolation .....	20
3.1.1 Varying p-value for 35 stations .....	20
3.1.2 Varying p-value for 326 stations.....	21
3.1.3 Comparison between 35 station and 326 stations .....	22
3.2 Ordinary kriging .....	23
3.2.1 Fitting of the variogram .....	23

3.2.2	Ordinary kriging using 35 stations .....	26
3.2.3	Ordinary kriging using 326 rain stations .....	26
3.3	Comparison between kriging and inverse distance .....	27
3.4	Soil moisture results.....	29
3.5	Fractional evaporation results.....	30
4	Discussion and conclusion .....	31
4.1	Limitations and shortcomings of the KNMI data .....	31
4.2	Limitations and shortcomings of PCraster .....	31
4.3	Next steps .....	32
4.4	Conclusion.....	33
5	References .....	34
6	Appendices.....	35
6.1	Starting the Program.....	35
6.2	Implementation of the program .....	35
6.2.1	Download of KNMI script using wget.....	35
6.2.2	Using Gstat for kriging and inverse distance .....	36

## 1 Introduction

Water management is an important subject in the Netherlands. A lot of companies are doing or have done research in this area. FutureWater, which is situated in Wageningen, is such a company.

FutureWater is a consultancy company that combines scientific research with practical solutions for water management. They carry out projects in the field of state-of-the-art and applied sciences and consult in practical solutions for water management. Besides performing projects for other companies, FutureWater also performs research for extending their own expertise.

One of those projects is presenting a daily estimation of the soil moisture and relative evaporation in the Netherlands on the FutureWater website<sup>[19]</sup>. The estimation of the soil moisture and relative evaporation is computed by use of a estimation model. This model is called the HydroS<sup>[1][2]</sup> model.

The reason FutureWater developed this model was the request for a model that estimates the effect of floods and water shortage in a relatively simple way. This was especially requested for local floods and water shortage in the Netherlands. Since there wasn't such a model FutureWater created this and called it the HydroS model. They wanted the model to be relatively simple. The reasons for that were that results do not always give better estimations when every aspect is modeled in detail. Another reason was that the simpler the model the easier to understand, making it more suitable for adaptation later on.

The reasons above resulted in a raster based model with parameterized processes in a flexible programming environment. The basic concept is a coupled double layer tray model and a vegetation layer, see figure 1.

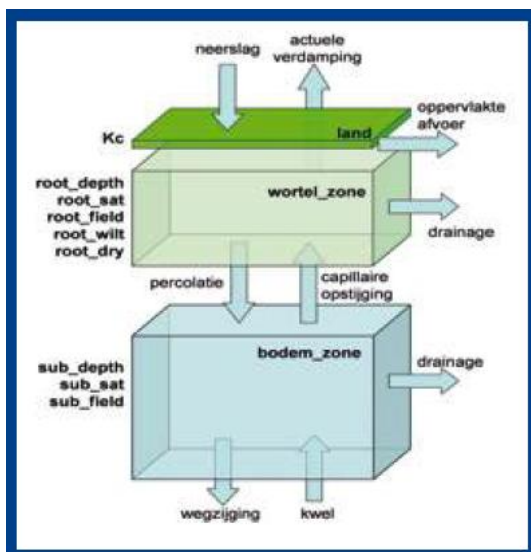


Figure 1: The tray model used in the HydroS model

The root zone layer, the light green box, determines the fluxes and thus total amount of water in this layer derived from physical soil properties. The sub zone, the light blue box, is based on physical soil

properties as well, which won't be discussed here in detail. The most important input parameters are: land use, soil, surface level, average level of open water, precipitation and evaporation.

These input parameters define the different types of land, since for example the daily amount of sunlight which results in a potential evaporation value gives an actual evaporation value based on the type of land for the estimated location. In cities less evaporation will occur than in an area with a lake. Another important example is the maximum drainage. The drainage value is limited to environmental factors. They have modeled this by distinguishing three types of areas: urban areas, polder areas and free drainage areas. These three all have their restriction on the total amount of drainage per time period. This resulted in mapping floods.

The HydroS model is implemented in the programming language PCraster, which is based on Python. PCraster is developed for the simulation of dynamical space-time models and other applications. The first version of the HydroS model was set up in a spatial resolution of 250 m by 250 m and the time steps were in days. The computing times were very short. It can model a whole year in this state in 5 minutes on a normal computer.

The output values of the model were compared to measurements and they were comparable. Examples that were compared were evaporation shortages and potential evaporation. They were compared to satellite measurements at the Bilt since their model used the daily input of this station.

There were a couple of steps that still needed to be implemented in the hydroS model. The model was able to calculate the output when input was given, but the input wasn't an automated process. The process of getting this data and preparing it as input for the model, the automation process, is described in section 2.1. The input data for the model is the amount of precipitation and evaporation for each time period, which in this case is per day, over the whole area. This data can be downloaded from the KNMI website. The data from the KNMI website is the daily precipitation and evaporation at each of their meteo stations. Which takes us to the second step described in this paper: The model was based on the data of one station, The Bilt. In this paper the use of multiple stations is used as input. Interpolation techniques are used to estimate the areas in between. Interpolation techniques are described in section 2.3 and 2.4.

We need to interpolate these data points to give an estimation of the unknown area for each point, which will be on a grid of 250 m by 250 m. There are a lot of interpolation techniques. Some important techniques for interpolation of spatial data will be explained in section 2.3. There will also be a brief explanation of the methods that will be used for our data. Then in the next section, section 2.4, we will discuss in detail the interpolation techniques used for the problem. In the first part of the results section, section 3.1 till 3.3, the rain and evaporation maps with different interpolation methods and varying parameters will be compared. Then in section 3.4 and 3.5 the result maps of soil moisture and relative evaporation calculated with the hydroS model by using the interpolated data will be discussed.

Section 4 is the discussion and conclusion section. The discussion part, 4.1 and 4.2, will first describe limitations of the various tools used and then describe next steps that should be taken for this project. The conclusions drawn for this project can be read in section 4.3.



## 2 Methods

Our primary goal is to show detailed maps of daily soil moisture and evaporation fraction over time at the FutureWater website. In this section we will first describe every step in the process from initial data to the google map images on the website. More details over the exact implementation can be found in Appendix A. The tools used for the implementation are described in short in the next subsection. Then background information over spatial interpolation techniques is given followed by a subsection describing the theory behind the spatial interpolation techniques used for the problem.

### 2.1 Implementation of the estimation process

For the implementation we used several programs, to connect all these programs we used a python script.

#### 2.1.1 Preproces of the data

Our initial data source is the KNMI<sup>[7]</sup>. This company is the primary source of spatial meteorological data in the Netherlands. They have enourmous databases.

We need the exact location of all the meteorological stations which we will use later to map meteo station data to the location the station is at. The locations, long-latitude, KNMI provides us with, needs to be converted to be compatible with Gstat. Gstat uses grid locations. We use GIS to do this conversion.

The hydroS model makes use of “Rijksdriehoekcoördinaten<sup>[8]</sup>”. This method is used for coordinates at a national level and is the basis of most geographic indications and files. In GIS these values are calculated, the value of a grid point where a station is situated is changed in a number that is assigned to that station. One map is created with the locations of the automated 35 meteo stations and one map with the locations of the 326 voluntary precipitation stations.

For the automated stations the data is available hourly and daily and is up to date. The daily data is based on the day indicated from 0 to 24 hour. The voluntary precipitation stations give only data of the precipitation. This is updated once a month within 3 weeks after the end of each month. These measured values are based on 8 AM the day previous to the day indicated till 8 AM the day indicated. Figure 2 and 3 show the exact location of the stations on a map for the 35 and 326 stations.

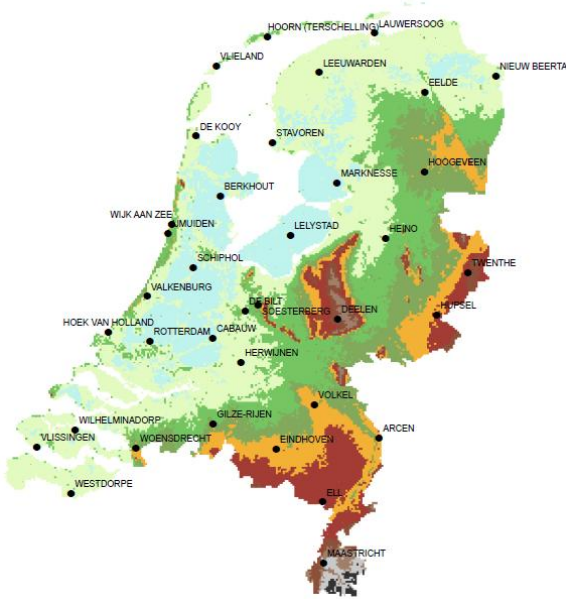


Figure 2: Locations of the 35 automated meteo stations

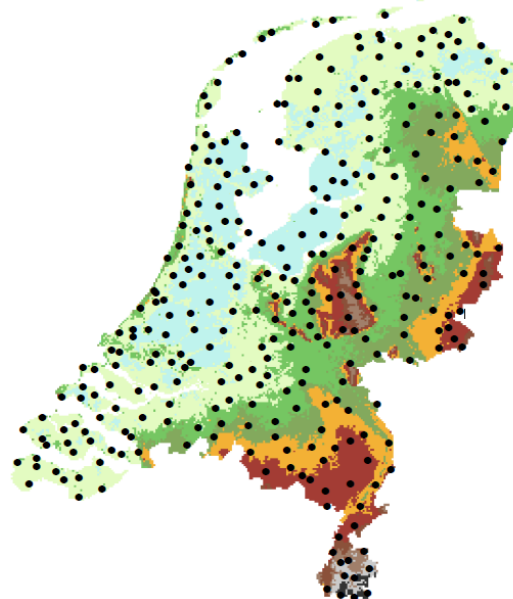


Figure 3: Location of the 326 voluntary precipitation stations

### 2.1.2 Daily data

We used the KNMI web services to obtain the daily precipitation and evaporation data of all the available meteo stations. At the KNMI website the daily precipitation and evaporation can be found. This can be downloaded as a text file. We automated the data request by using Wget from within our main python script. Wget is a program to facilitate web requests from the command line or within programs. To read more about the exact method to get this process automated, read the appendix section 6.1.

### 2.1.3 Interpolating the spatial data points

The HydroS model can be applied on map data only. At this point we only have measurements from the KNMI meteo stations and their locations on the grid. Gstat<sup>[13]</sup> will need these two data sources to generate an interpolation map.

The data we receive from the KNMI needs to be converted into the Gstat input format. We use python directly to generate a text file in this format.

We can configure Gstat<sup>[11]</sup> to use several interpolation techniques to generate maps. In our first tests we use *inverse distance weighted*, later we use *ordinary kriging*. See section 2.3 and 2.4 for details on both and read in the appendix section 6.2.2 to get the information of the implementation in gstat.

Gstat outputs map data which shows estimates of every point on the grid. This is a .map file which can be read directly by our next step, the Simulation step. For the simulation 100 maps of daily precipitation and evaporation will be generated as input.

### 2.1.4 The Simulation

When the maps for precipitation and evaporation are generated the simulation in the HydroS model can start. The HydroS model is written in the programming language PCraster, which is based on python.

The program is executed by our python script. The model will simulate over a period of 100 days to calculate the soil moisture and potential evaporation. The output of the simulation is a map with an estimation of the soil moisture.

## 2.2 Tools for implementation

The tools that we used for the simulation model are described here.

### 2.2.1 PCraster

The hydroS model for computing the soil moisture is written in the program PCraster<sup>[14]</sup>. This program is developed at the Department of Physical Geography, Faculty of Geosciences at the Utrecht University and at PCraster Environmental Software(PES). It is collection of software that is targeted at the development and deployment of spatio-temporal environment models. The program can be used in combination with the editor NutShell. PCraster is based on the programming language Python and PCraster functions can be imported in Python.

### 2.2.2 Python

This language is developed by a Dutch person, Guido van Rossum<sup>[6]</sup>. Since the PCraster program is based on python, to connect all components together the programming language python is used. Python is a suitable language to glue code together of command line programs. This is what the program does in this model.

### 2.2.3 GNU Wget

Wget is a computer program that retrieves content from web servers, and is part of the GNU Project. This program is used to retrieve the precipitation and evaporation of the KNMI website by an interactive selection of the data. The output is a text file.

### 2.2.4 Gstat

Gstat is an open source computer code for multivariable geostatistical modeling, prediction and simulation. It can be used as a stand-alone executable or can be used by a package in programs like R. We used Gstat program for modeling and prediction. The output is compatible with PCraster which makes this a suitable program.

## 2.3 Spatial Interpolation techniques

Several interpolation techniques exist for estimating unknown points in spatial data sets. These techniques can be divided into two main groups, deterministic methods and kriging. We will explain each of them and the most important techniques of each group will be discussed briefly.

### 2.3.1 Deterministic methods

Deterministic interpolators<sup>[12]</sup> can be divided into two groups, exact and inexact interpolators<sup>[9]</sup>. An exact interpolator predicts values identical to the measured data. An inexact or smoothing interpolator predicts its values using a smoothing function, which will avoid sharp peaks and deep valleys in the surface. Examples of exact interpolators are polygonal estimator, inverse distance weight and radial basis function. An inexact interpolator is: global/local polynomial. These will now be discussed briefly.

### **Polygonal estimator**

In this interpolation technique the estimate is based on the sample point closest to the point to estimate. This polygonal estimator can be seen as a weighted linear estimator that gives all of its weight to the nearest sample value. This yields a discontinuous surface.

### **Inverse distance weighted interpolation to a power**

Inverse Distance Weighted<sup>[3,4,12]</sup> (IDW) interpolation implements the assumption that points close to each other are more alike than points that are more distant. To estimate the unknown values, IDW estimates the unknown point from surrounding known points by a weighted mean. Points closer to the location to estimate get higher weights than points that are further away. The weights assigned to each point are inversely proportional to its distance from the unknown point.

### **Radial basis function**

This interpolation technique tries to fit the amount of rainfall at the surface by minimizing the total curvature. The results that this method produces can be best compared to results gained by kriging. It gives the best results when the data is dense.

### **Global and local polynomial interpolation**

This technique uses a polynomial function to fit a smooth function to the measured data points. The difference between this method and radial basis function is that this is an inexact method, which means that the surface doesn't need to pass through all the measured points. This method is used for a surface that gradually changes. A disadvantage is that it is very sensitive to outliers in the data set, especially at the edges.

## **2.3.2 Kriging**

Kriging<sup>[3,4,10]</sup> is a group of geo-statistical techniques to estimate the value of a random field at an unobserved location from observed values nearby. The main difference between kriging and deterministic techniques is that kriging is a statistical approach, assuming the data to be a realization of a random field. It takes into account the spatial dependence quantified by for example a variogram<sup>[4,15]</sup> or the covariance in the random field.

Kriging is often associated with the acronym B.L.U.E. that stands for: "best linear unbiased estimator". It is a linear estimator, because the estimates are computed by a weighted linear combination of the observed points. There are different types of kriging which differ in the way these weights are calculated. The method for calculating these weights depends on the stochastic properties of the random field. These methods of kriging will be explained later. The term "best" in B.L.U.E. means that its aim is to minimize the variance of the errors.

There are several advantages of kriging over deterministic interpolation. The most striking difference is that kriging not only considers the distance, but also takes into account the direction, and therefore attempts to decluster the available sample data. This is called isotropy. When there is a spatial correlation in distance or directional bias present in the data this method will generate better estimations than deterministic methods. The degree of smoothing in ordinary kriging is higher than in other methods. Still, these estimates maintain closer resemblance of the true 'shape' of the data.

Different types of kriging are suitable for different types of data. There are two techniques which are applicable to the rain data. These will be explained briefly.

### Simple kriging

This is mathematically the least complicated form of kriging. This method is based on three assumptions. First it assumes that the observations are a partial realization of a random field  $V(x)$ , where  $x$  denotes the spatial location. The second assumption is that the random function is second order stationary, which means that  $cov(V(x_1), V(x_2))$  does only depend on  $|x_1 - x_2|$ . The last assumption is the most limiting one: it assumes the mean to be known. This is the reason that simple kriging is not the most general kriging method.

### Ordinary kriging

The difference between ordinary kriging and simple kriging is that ordinary kriging assumes the mean to be unknown. Most data sets do not know their ‘true’ mean value, this makes ordinary kriging the most commonly used method.

## 2.4 Interpolation techniques used for the problem

For the computation of the rain and evaporation over the whole area, we implemented two interpolation techniques. These two will be compared later. These techniques are: inverse distance weighted interpolation to a power and ordinary kriging. First we will describe why these methods were chosen over the other methods

The first choice that we had to make was to choose between an exact or inexact interpolator. Since the measurements could be accepted as reliable measurements an exact method seemed the best option.

The polygonal estimator doesn’t seem an appropriate technique because it just takes the nearest sample for unknown location. This creates a discontinuous surface and that would not be a good reflection of the truth. The radial basis function does not seem like a good option either, because of the necessity of dense data. Since there are only 35 data points the data are not dense at all.

The inverse distance method is an easy to implement method that gives weights to nearby points. This is a common method for spatial interpolation. A disadvantage of this method is the so called “bull’s eyes”, which will be explained in the next subsection, around known data points.

Kriging is known to be a good estimation method for spatial interpolation. Ordinary kriging was a suitable method for interpolating our data. A disadvantage of kriging is that it isn’t an easy method to implement. This is why we chose to implement two interpolation methods: inverse distance weighted and ordinary kriging. Thus we can compare the two methods and decide which one is the best.

### 2.4.1 Inverse distance weighted interpolation to a power

As was explained in short in the previous subsection; Inverse Distance Weighted (IDW) interpolation estimates the unknown point from surrounding known points by a weighted mean. Points closer to the location to estimate get higher weights than points further away. This is calculated by assigning a weight to each point that is inversely proportional to the distance to the unknown point:

$$\tilde{v} = \frac{\sum_{i=1}^n \frac{1}{d_i} v_i}{\sum_{i=1}^n \frac{1}{d_i}} \quad (4.1)$$

Where  $d_1, \dots, d_n$  are the distances from each of the  $n$  sample locations to the point being estimated and  $v_1, \dots, v_n$  the sample values. The weights are divided by  $\sum_{i=1}^n \frac{1}{d_i}$  in order to normalize.

With this formula the weights will decrease rapidly as the distance increases. In Equation (4.1) the weights are inversely proportional to the distance, but the weights can be made inversely proportional to any power of the distance:

$$\tilde{v} = \frac{\sum_{i=1}^n \frac{1}{d_i^p} v_i}{\sum_{i=1}^n \frac{1}{d_i^p}} \quad \text{with } p \geq 0 \quad (4.2)$$

Different values for  $p$  will result in different estimates. If  $p = 0$  there is no decrease in distance. Geo-statistical analysts use powers of  $p$  between 1 and 2. When  $p = 2$  this is known as the inverse distance squared weighted interpolation, this is also the most common choice. One of the reasons is that a  $p$ -value of 2 is more common than lower  $p$ -values is that otherwise points further away have too much influence on the point to estimate. Figure 4 shows the influence of the power  $p$ .

The optimal  $p$ -value can be determined by minimizing the root mean squared prediction error, which can be found by using cross validation.

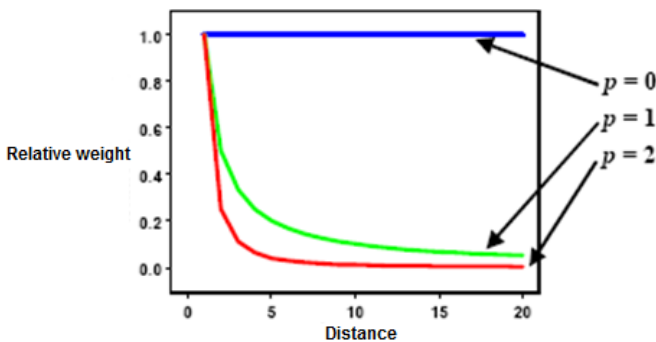


Figure 4: Influence of the  $p$ -value in IDW interpolation

A disadvantage of IDW is that it has the tendency to generate “Bulls eyes”, which are counters around the known data points.

### 2.4.2 Ordinary Kriging

As was already explained in short, kriging estimates are computed by a weighted linear combination of the random variables from the available samples.  $V(x)$  is a partial realization of a random field. There were three assumptions stated in paragraph 3.2. First it is assumed that  $E(V(x)) = EV$  independent of  $x$  and second the  $Cov(V(x_1), V(x_2)) = C(|x_1 - x_2|)$ , independent of  $x_1$ .

To estimate the unknown value the following formula can be used:

$$\hat{V}(x_0) = \sum_{i=1}^n w_i * V(x_i) \quad (4.3)$$

Where  $\hat{V}(x_0)$  is the estimation of the value at location  $x_0$ ,  $w_i$  are the computed weights and  $V(x_i)$  are the observed values.

To find the BLUE<sup>[4]</sup>, best linear unbiased estimator; the mean error needs to be zero and the error variance minimized. The estimation error is defined as:

$$R(x_0) = \sum_{i=1}^n w_i * V(x_i) - V(x_0) \quad (4.4)$$

To attain a zero bias at any location the expected value of  $R(x_0)$  should equal 0 for all  $x_0$ , i.e:

$$\begin{aligned} E(R(x_0)) &= E\left(\sum_{i=1}^n w_i * V(x_i) - V(x_0)\right) \\ &= \sum_{i=1}^n w_i E(V(x_i)) - E(V(x_0)) \\ &= EV(\sum_{i=1}^n w_i - 1) = 0 \end{aligned} \quad (4.5)$$

This yields the following condition for the weights  $w_i$ :

$$\sum_{i=1}^n w_i = 1 \quad (4.6)$$

Now that the mean error is set to zero the mean squared error needs to be minimized which gives:

$$\begin{aligned} \min E((\hat{V}(x_0) - V(x_0))^2) &= \min E((R(x_0))^2) \\ &= \min(VarR(x_0)) \end{aligned} \quad (4.7)$$

Since  $R(x_0) = \hat{V}(x_0) - V(x_0)$  and  $E(R(x_0)) = 0$  for every  $x$ . The variance of the error can be expressed as:

$$\begin{aligned} Var(R(x_0)) &= Var(\hat{V}(x_0) - V(x_0)) \\ &= Var(\hat{V}(x_0)) + Var(V(x_0)) - 2Cov(\hat{V}(x_0), V(x_0)) \end{aligned} \quad (4.8)$$

Since the  $Var(\hat{V}(x_0))$  is equal to the covariance of  $\hat{V}(x_0)$  with itself and  $\hat{V}(x_0)$  is a linear combination of other random variables  $Var(\hat{V}(x_0))$  can be written as:

$$\begin{aligned} Var(\hat{V}(x_0)) &= Cov(\sum_{i=1}^n w_i V(x_i), \sum_{j=1}^n w_j V(x_j)) \\ &= \sum_{i=1}^n \sum_{j=1}^n w_i w_j * Cov(V(x_i), V(x_j)) = \sum_{i=1}^n \sum_{j=1}^n w_i w_j \tilde{C}_{i,j} \end{aligned} \quad (4.9)$$

Where  $\tilde{C}_{i,j} = C(|x_i - x_j|)$ .

Another assumptions is that  $V(x_0)$  has variance  $C(0)$  for all  $x_0$ , which is denoted by  $\tilde{\sigma}^2$ . Then the second term of equation (4.8) can be expressed as:

$$Var(V(x_0)) = \tilde{\sigma}^2 \quad (4.10)$$

The last term of equation (4.8) can be written as:

$$\begin{aligned} Cov(\hat{V}(x_0), V(x_0)) &= Cov(V(x_0), \sum_{i=1}^n w_i V(x_i)) \\ &= \sum_{i=1}^n w_i Cov(V(x_i), V(x_0)) = \sum_{i=1}^n w_i \tilde{C}_{i,0} \end{aligned} \quad (4.11)$$

Substituting 4.9, 4.10 and 4.11 into equation (4.8) gives the following expression for the error variance  $\tilde{\sigma}_R^2$ :

$$\tilde{\sigma}_R^2 = Var(R(x_0)) = \tilde{\sigma}^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j \tilde{C}_{i,j} - 2 \sum_{i=1}^n w_i \tilde{C}_{i,0} \quad (4.12)$$

This equation needs to be minimized. This is usually accomplished by setting the n first order partial derivatives of  $\tilde{\sigma}_R^2$  with respect to  $w_1, \dots, w_n$  to 0. Since the condition of unbiasedness (4.6) must be fulfilled, the number of equations exceeds the number of unknowns by one. This can be solved by using the technique of Lagrange Multipliers. This will add another unknown without adding a new equation. This means that there are now n+1 equations with n+1 unknowns. The new parameter, which is the Lagrange parameter, will be called  $\mu$ . This gives:

$$\tilde{\sigma}_R^2 = \tilde{\sigma}^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j \tilde{C}_{i,j} - 2 \sum_{i=1}^n w_i \tilde{C}_{i,0} + \underbrace{2\mu(\sum_{i=1}^n w_i - 1)}_0 \quad (4.13)$$

The last term will be 0 since  $\sum_{i=1}^n w_i = 1$  so the last term does not affect the equality.

To minimize the error variance the n+1 first order partial derivatives will be set to 0 with respect to each of these variables. To produce the unbiasedness condition the first order partial derivative with respect to  $\mu$  will be set to 0. This will only include the last term since the first three terms do not contain  $\mu$ :

$$\begin{aligned} \frac{\partial(\tilde{\sigma}_R^2)}{\partial\mu} &= \frac{\partial(2\mu(\sum_{i=1}^n w_i - 1))}{\partial\mu} \\ &= 2 \sum_{i=1}^n w_i - 2 = 0 \end{aligned} \quad (4.14)$$

The error variance will be minimized by calculating the n+1 first order partial derivatives and setting each one to 0. The first term on the right-hand side in equation (4.13) doesn't depend on  $w_1$ . The second equation will give:

$$\begin{aligned} \frac{\partial(\sum_{i=1}^n \sum_{j=1}^n w_i w_j \tilde{C}_{ij})}{\partial w_1} &= \frac{\partial(w_1^2 \tilde{C}_{11} + 2w_1 \sum_{j=2}^n w_j \tilde{C}_{1j})}{\partial w_1} \\ &= 2w_1 \tilde{C}_{11} + 2 \sum_{j=2}^n w_j \tilde{C}_{1j} = 2 \sum_{j=1}^n w_j \tilde{C}_{1j} \end{aligned} \quad (4.15)$$

The third term on the right-hand side of equation (4.13) gives:

$$\frac{\partial(\sum_{i=1}^n w_i \tilde{C}_{i0})}{\partial w_1} = \frac{\partial(w_1 \tilde{C}_{10})}{\partial w_1} = \tilde{C}_{10} \quad (4.16)$$

The last term of equation (4.13) will give:

$$\frac{\partial(\mu(\sum_{i=1}^n w_i - 1))}{\partial w_1} = \frac{\partial(\mu w_1)}{\partial w_1} = \mu \quad (4.17)$$

So then the first derivative of  $\tilde{\sigma}_R^2$  with respect to  $w_1$  can be written as:

$$\frac{\partial(\tilde{\sigma}_R^2)}{\partial w_1} = 2 \sum_{j=1}^n w_j \tilde{C}_{1j} - 2\tilde{C}_{10} + 2\mu \quad (4.18)$$

To minimize the error (4.18) will be set to zero:



$$2 \sum_{j=1}^n w_j \tilde{C}_{1j} - 2\tilde{C}_{10} + 2\mu = 0$$

$$\sum_{j=1}^n w_j \tilde{C}_{1j} + \mu = \tilde{C}_{10} \quad (4.19)$$

Similarly, for  $w_i$  this can be written as:

$$\sum_{j=1}^n w_j \tilde{C}_{ij} + \mu = \tilde{C}_{i0} \quad \text{for all } i = 1, \dots, n \quad (4.20)$$

Solving (4.20) for  $i=1, \dots, n$  together with (4.14) for  $w_1, \dots, w_n, \mu$  yields the optimal weights.

### 2.4.3 Variogram

For calculating the ordinary kriging weights a pattern of spatial continuity for the random model must be defined. This is the variogram, often called by the name semivariogram, since that this is the most common used variogram. The steps for constructing the sample variogram is to first define an experimental variogram<sup>[17]</sup>. From the experimental variogram a model variogram can be derived. Mostly multiple experimental variograms are plotted before one is chosen. For computing this variogram it is impractical to define all the variances between all the points. This is why the points will be divided into intervals based on their distance, these intervals are called lags<sup>[16]</sup>. The variances for all points in the same lag is computed and averaged.

The formula for calculating the lag values for the semivariogram is:

$$\gamma(h) = \frac{1}{2n} * \sum_{i=1}^n (V(x_{1i}) - V(x_{2i}))^2 \quad (4.21)$$

Where the sum runs over those  $i$  for which the distance between  $x_{1i}$  and  $x_{2i}$  is in the lag labeled  $h$ . This means that for all distances within the bin range, defined as  $h$ , we calculate the variance between the two points. The average value is the variogram value. The division by two is because the semivariogram is defined as  $2\gamma_{ij}(h)$  and dividing by two makes it comparable to the covariance's used by ordinary kriging.

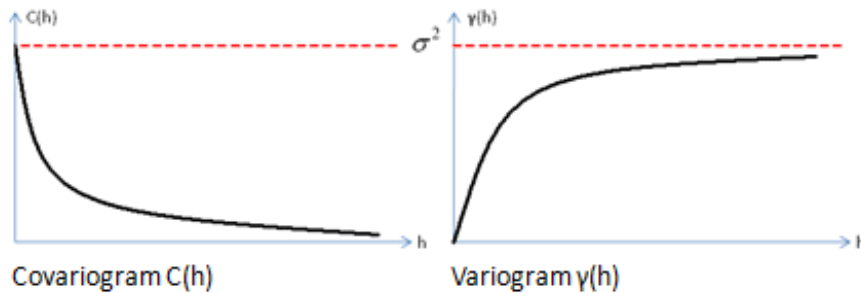


Figure 5: An example of a variogram model and a covariance function

Since ordinary kriging calculates its weights from the covariances, we have to rewrite the variogram values into covariances. Figure 5 shows the relation between covariance and variogram. We assumed that the random values in our random function model all have the same mean and variance. These assumptions allow us to develop the following relationship between these two models:

$$\begin{aligned}
 \gamma_{ij} &= \frac{1}{2}E(V(x_i) - V(x_j))^2 \\
 &= \frac{1}{2}E(V_i^2) + \frac{1}{2}E(V_j^2) - E(V_i * V_j) \\
 &= E(V^2) - E(V_i * V_j) \\
 &= E(V^2) - (EV)^2 - (E(V_i * V_j) - (EV)^2) \\
 &= \tilde{\sigma}^2 - \tilde{C}_{ij}
 \end{aligned} \tag{4.22}$$

The ordinary kriging system can be rewritten in terms of the variogram as:

$$\begin{aligned}
 \sum_{j=1}^n w_j \tilde{\gamma}_{ij} - \mu &= \tilde{\gamma}_{i0} \quad \text{for all } i = 1, \dots, n \\
 \sum_{i=1}^n w_i &= 1
 \end{aligned} \tag{4.23}$$

### Model types

Once the experimental variogram is defined a model variogram needs to be constructed to fit the experimental one. There are several types of variogram models<sup>[4,5]</sup>. The two most common types are described here.

**Spherical model.** This might be the most commonly used variogram model. Its standardized equation is:

$$\gamma(h) = \begin{cases} 1.5 \frac{h}{a} - 0.5 \left(\frac{h}{a}\right)^3 & \text{if } h \leq a \\ 1 & \text{otherwise} \end{cases} \tag{4.24}$$

Where  $a$  is the range. It has linear behavior near the origin and flattens out at larger distances.

**Exponential model.** Its standardized equation is:

$$\gamma(h) = 1 - \exp\left(-\frac{3h}{a}\right) \tag{4.25}$$

The range is the value at which the distance is 95% of the sill. These words will be described below in the subsection model parameters.

### Model parameters

**Nugget effect.** This is the discontinuity (offset) at the origin. The height is represented by the variable  $C_0$  which is the minimum variance between two points with a very short distance. In figure 6 this is from 0.0 till the nugget line.

**Range.** This value is here defined as  $a$  and is the distance beyond which the variogram value remains almost constant.

**Sill.** This is the limit of the variogram. The maximum variance at which points will be included to estimate a certain point. This is defined as  $C_0 + C_1$ .

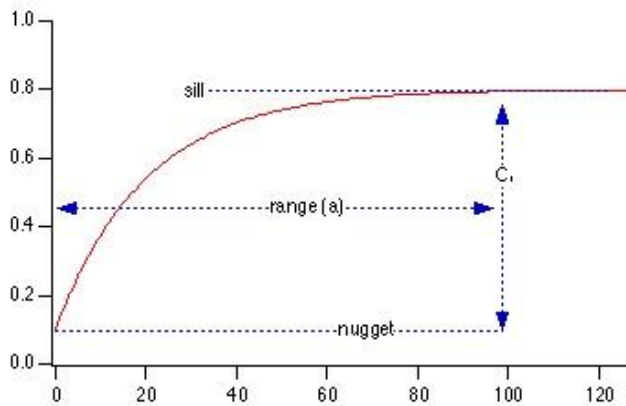


Figure 6: Shows the parameters for a variogram model

## 3 Results

### 3.1 Inverse distance weighted interpolation

The first method we used for interpolation of the precipitation and evaporation is the inverse distance weighted (IDW) method. This method has only one parameter that can change; this is the p-value. First the results for a varying p-value for 35 meteo stations are shown. Then figures will be shown for a varying p-value for the 326 rain stations. The two will be compared to each other to show what the difference in estimates is when fewer stations are used.

#### 3.1.1 Varying p-value for 35 stations

First we implemented the inverse distance weighted (IDW) method. This method has only one parameter that changes, this is the p-value. The standard p-value is 2. Which means that an inverse distance squared weighted interpolation is used. The first maps were created with this p-value.

To see the influence of the p-value we created maps with three p-values, a p-value of 1, 2 and 3. This can be seen in the figures 7, 8 and 9 respectively, for 14 July 2011. The p-value of 2 will be compared to a p-value of 1 and 3, since 2 is the standard p-value. As you can see there was a lot of rainfall in the west. As we have already described in section 2.4, IDW has the tendency to create “bull's eye's” which are circles around a data points. A lot of these circles can be seen in these images. For example in figure 8, in the west, three meteo stations can be distinguished, the red bullets. Probably in between these stations is more rainfall than is shown in this picture.

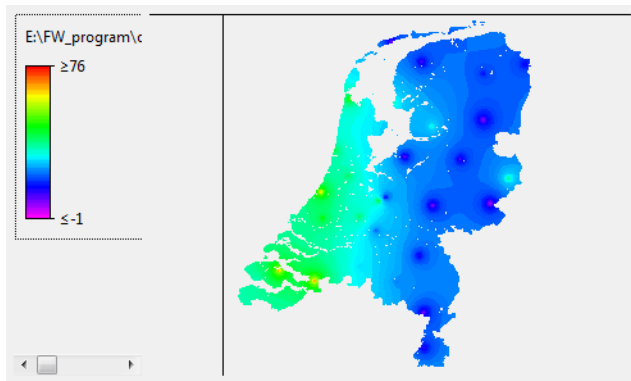


Figure 7: IDW with a p-value of 1 for 35 stations on 14 July 2011

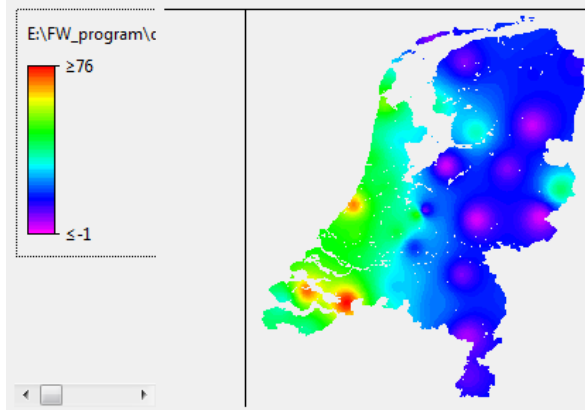


Figure 8: IDW with a p-value of 2 for 35 stations on 14 July 2011

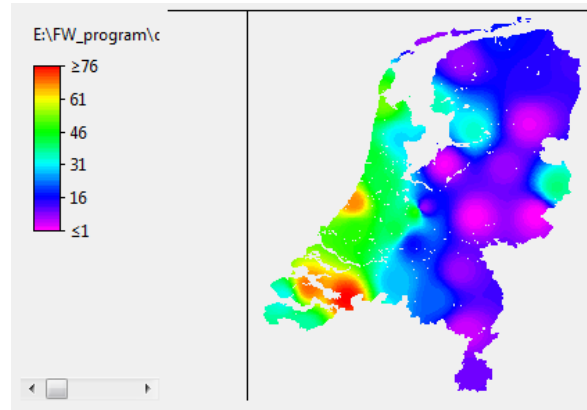


Figure 9: IDW with a p-value of 3 for 35 stations on 14 July 2011

If we look at the same day, but then with a p-value of 1 the model generates the output shown in figure 7. As is shown in this figure the “bull’s eyes” are smaller. The influence of nearby stations is minor. This p-value is probably not the right one for this number of stations.

So this means that the p-value for this number of stations should at least be two. A p-value of three will give the output that can be seen in figure 9. As can be seen the meteo stations in the west have a bigger influence in this area. Another distinction is the purple circles. At these points almost no rainfall is shown. The areas between the purples circles are blue, which implies more rainfall. This is probably an overestimation, due to the points in the west. The average rainfall for each day is calculated as well. The models gave as output for  $p=1$  is 25.6 mm, for  $p=2$  this 23.8 mm and for  $p=3$  it is 22.8 mm. We thought that the points in the west gave an underestimation of the rainfall, but the total amount of rain decreases with a rising p-value. Probably the total amount of rainfall in the east is a bigger overestimation.

### 3.1.2 Varying p-value for 326 stations

For older data the rainfall at 326 points in the Netherlands are given. This should give a better estimation of the rainfall since there is more data available. As can be seen from figure 10 much more detail is shown with 325 rain stations. For this estimation a p-value of 3 is used. In this figure it looks like there is a heavy rainfall in the west and at the east side of the country there almost isn’t any rainfall. This should imply that a bigger p-value will give better estimates, because in the west the rainfall is probably bigger than is shown and in the east vice versa.

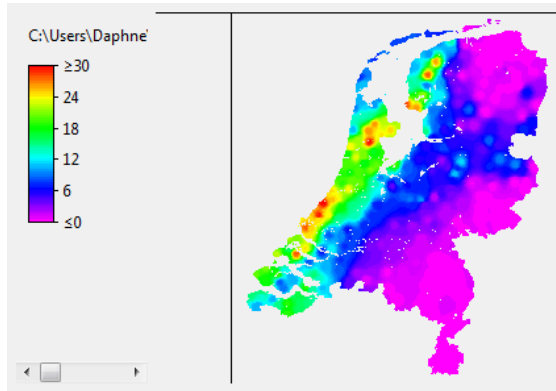


Figure 10: IDW with a p-value of 3 for 326 stations on 7 June 2011

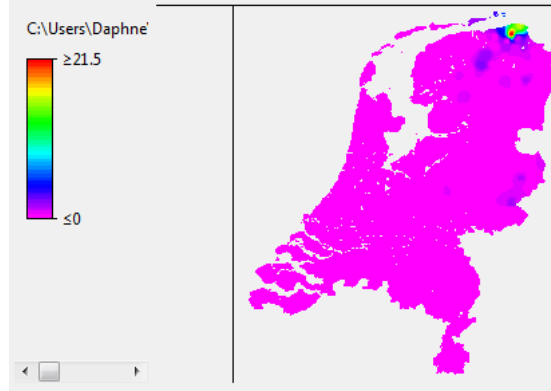


Figure 11: IDW with a p-value of 3 for 326 stations on 14 May 2011

If we look at figure 11, which is also estimated with a p-value of 3, than probably we see a local rain shower. When a larger p-value is chosen the measurements of local showers could have too much influence. From these conclusions we decided to base our other estimation for IDW on a p-value of 3.

### 3.1.3 Comparison between 35 station and 326 stations

Obviously the main difference between 35 stations and 326 stations is the amount of detail that can be shown. Figure 12 and 13 show estimation of precipitation for the same day. As can be seen in figure 12 a big red dot is visible. If you compare this figure to figure 13 we can conclude that this data point has too much influence and was probably a local rain shower. Nevertheless the 35 stations don't always give bad estimates. This is shown in figure 14 and 15.

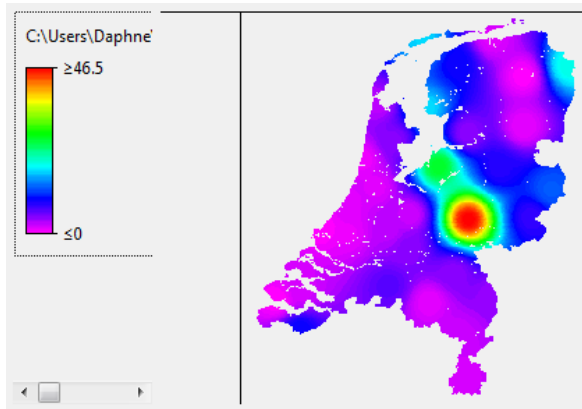


Figure 12: IDW with a p-value of 3 for 35 stations on 6 May 2011

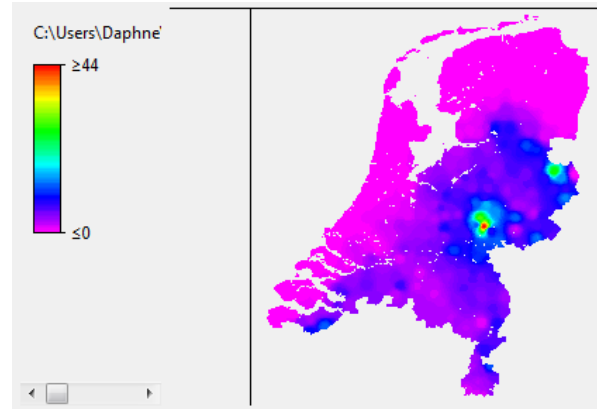


Figure 13: IDW with a p-value of 3 for 326 stations on 6 May 2011

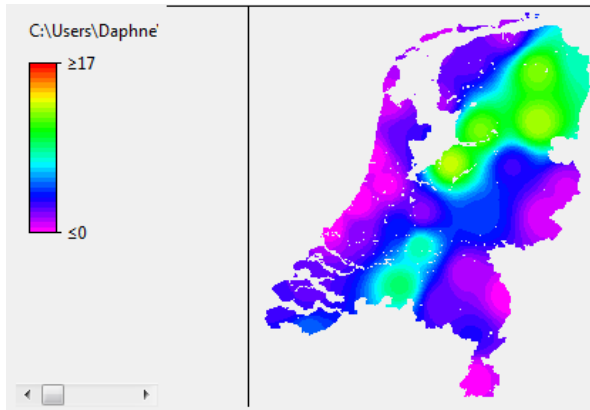


Figure 14: IDW with a p-value of 3 for 35 stations on 26 May 2011

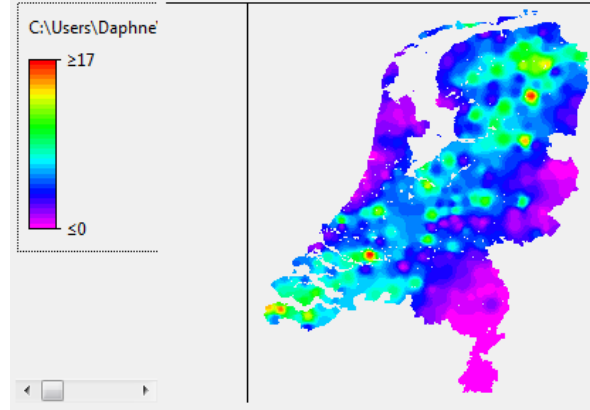


Figure 15: IDW with a p-value of 3 for 326 stations on 26 May 2011

## 3.2 Ordinary kriging

Before we can start the simulation with ordinary kriging as interpolation technique, we first need to specify the variogram. The variogram parameters must be specified. How these parameters were chosen will be described in the following subsection. The results for the different input parameters will be shown and compared to each other in the next subsection.

### 3.2.1 Fitting of the variogram

For the variogram a model and input parameters need to be set. Information about the various models and input parameters can be found in section 2.4.3. First we need to specify the model. We chose between the two most common model types; exponential and spherical.

The input parameters for each of those two models are the sill, nugget and range. These values can be fitted to the data input. First initial parameters serve as an input. Then a program like Gstat can fit the input data to variogram by changing the initial parameters. The initial parameters need to make some sense, otherwise the output won't give proper results. We will now describe the procedure for fitting the variogram.

The initial parameters that we chose for the 35 meteo stations were 10 as the sill and 50000 for the range, and a nugget of 1. In the Gstat program a nugget of 1 means we do not take the nugget into account. The value 50000 is chosen, because 50 km range seemed an appropriate number. The variance of 10 is found after trying several values.

Figure 16 shows the results of the fitting process. As you can see, the lags are created and each data point shows how many pairs are in this lag, what their average distance is and the semi variance value. This is plotted in figure 18. The +value gives how many pairs are in this lag, what the average distance is in the lag and the average semi variance value. A rule of thumb is to choose at least 30 points per lag. This isn't the case for this output. A possible solution is taking less lags, but this would result in too big lag distances.

```

#sample semivariogram
#Tue Aug 02 11:48:36 2011
#data(evaporation): 'evapdata_old76.txt', x=1, y=2, v=3;
#[1] mean: 2.69375 variance: 0.50254
#cutoff: 123001 interval width: 8200.06
#direction: total
#
# from      to      n_pairs  av_dist  semivariogram
16400.1  24600.2      5  21991.4    0.12
24600.2  32800.2      9   28905  0.252778
32800.2  41000.3     22  37586.7  0.301136
41000.3  49200.3     15  45919.5  0.0873333
49200.3  57400.4     21  53420.2    0.19
57400.4  65600.5     17  62027.3  0.197647
65600.5  73800.5     22  70175.1  0.419318
73800.5  82000.6     20  78401.4   0.5995
82000.6  90200.6     32  86319.4  0.395625
90200.6  98400.7     16  94477.7  0.253437
98400.7  106601      26  102931  0.384423
106601   114801      28  109864  0.622143
114801   123001      27  119343  0.827407

```

Figure 16: Output fitting of the variogram for 35 stations for evaporation on 7 June 2011

We compared two models to each other, the exponential and the spherical. Figure 17 shows the fit for one day of data input for the 326 rain station, the exponential model seemed the best fit. This doesn't seem like a perfect fit. The difficulty is that the input data changes per day and we haven't found an option to automatically fit the data. Although the line doesn't follow the flow of the points perfectly, the output isn't really bad. Since the function to automatically fit the variogram to the daily data wasn't found, one fixed set of estimated parameters was used.

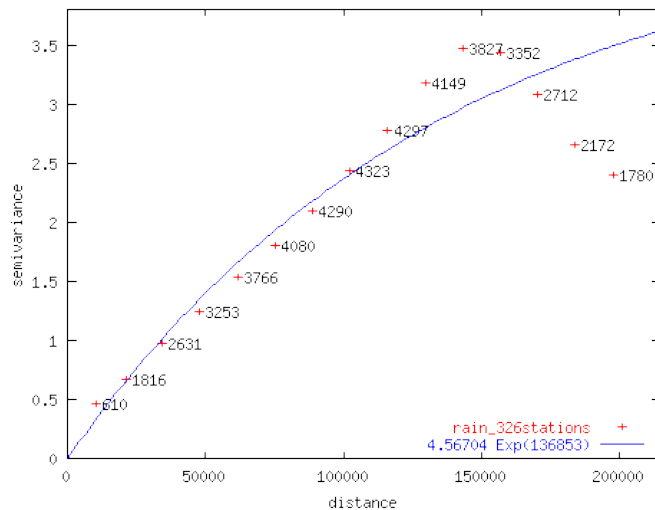


Figure 17: Semivariogram for the 326 rain station with an exponential fitting model.

The variogram for the 35 stations was even more complicated to fit, see figure 18. The points are rather distributed and the data is very sparse. The data is too sparse to give an accurate estimation for each day. Figure 18 shows a fit for one day for the evaporation. As can be seen in the picture, none of the two common models will fit these data properly. This is why we used our initial values for the input. We



think that the only option for better fitting results for kriging is more measured data as input, or more advanced variogram models.

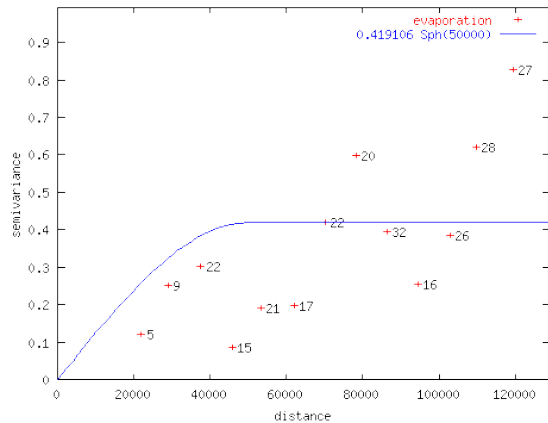


Figure 18: Semivariogram for evaporation using 35 stations with a spherical fitting model

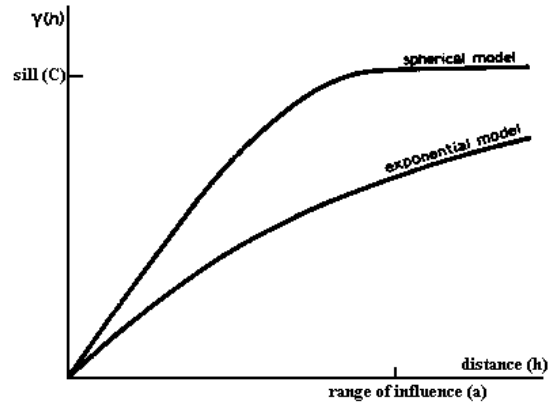


Figure 19: Graph of the two models used for the variogram

We compared the two models to each other for each of the two precipitation stations input, 35 and 326. For the Spherical model we chose as parameters: 0 as nugget, 10 as a sill and 50000 as the range. For the exponential model we chose the parameters: 0 as nugget, 4 as the sill and 130000 as the range. These parameters were not taken equally, because the variogram fit method gave different parameter fits for both models. The exponential seemed like a better fit for the 35 stations as well as for the 326 stations, but when the same parameters were used this might be different. Figure 19 shows a graph of the two models when the same parameters are used. Figures 20 and 21 show the output for these two models. As you can see figure 20 gives a smoother result, and the method that is shown in figure 20 has the tendency to create “bull’s eyes”. This is why we chose to use the exponential method as the basis for all our estimations. These values are probably not optimal, but due to lack of time they are not further optimized.

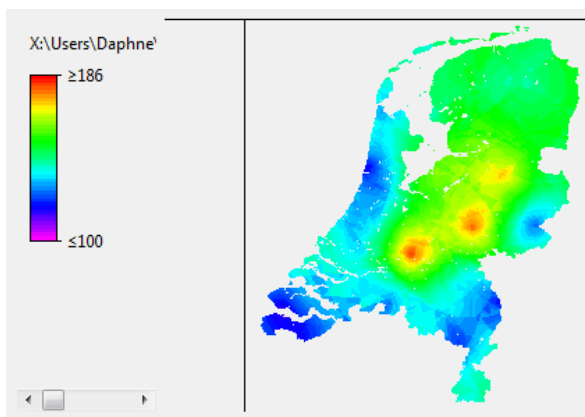


Figure 20: Sum of the rain from 22 March till 30 June 2011 using the Spherical method

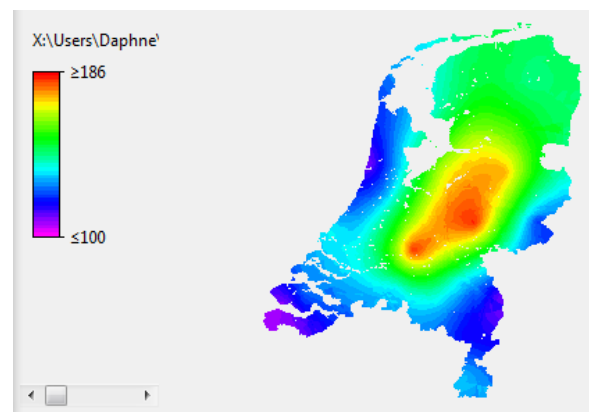


Figure 21: Sum of the rain from 22 March till 30 June 2011 using the Exponential method

### 3.2.2 Ordinary kriging using 35 stations

As we have mentioned above it was difficult to find good estimates for the parameters for 35 stations. Although the parameters are probably not optimal, the images look plausible. The image shown in figure 22 is taken from the same input day as figure 14 and 15. In the south-west the rainfall that can be seen in figure 15 isn't shown in 14 and 22. This is because the automated rain stations near that area did not measure this, which made it impossible to estimate in the model.

Another result from the kriging method that can be seen in this figure is the area between stations with a high precipitation value. Four points can be distinguished in this picture, but they aren't clear, the areas in between give a high precipitation value as well. These are plausible results, because when two rain stations near each other give high measurements the values in the area in between are likely to be also high.

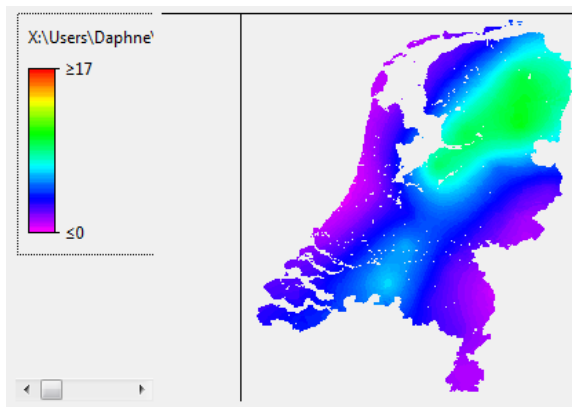


Figure 22: Ordinary kriging for 35 stations on 26 May 2011

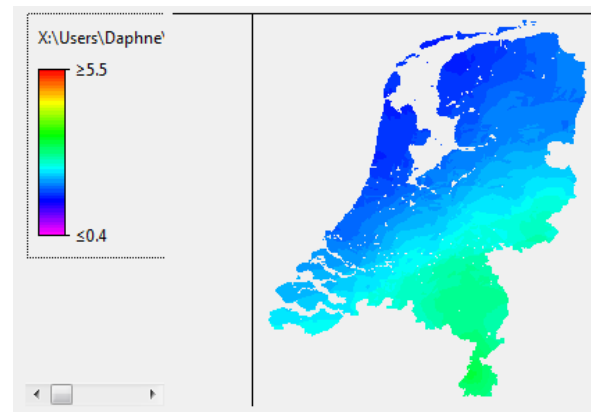


Figure 23: Map of the estimation of evaporation with ordinary kriging for 35 stations on 26 May 2011

We think that evaporation shows better results than precipitation. We think that the reason is that the amount of sunshine is more constant over a bigger area than for example local rain showers that can occur. Figure 23 shows the estimation of the evaporation on the same day as the maps from the figures 14, 15 and 22. In the North West there was less sunshine, probably because of the clouds. In the South East there were probably fewer clouds during the day which caused more evaporation.

### 3.2.3 Ordinary kriging using 326 rain stations

For the ordinary kriging method with 326 rain stations the outputs were very smooth maps. Figure 24 shows an estimation of the precipitation on 26 May 2011. This day was already used for the other method and rain station number. When you compare them, this figure probably gives the best result.

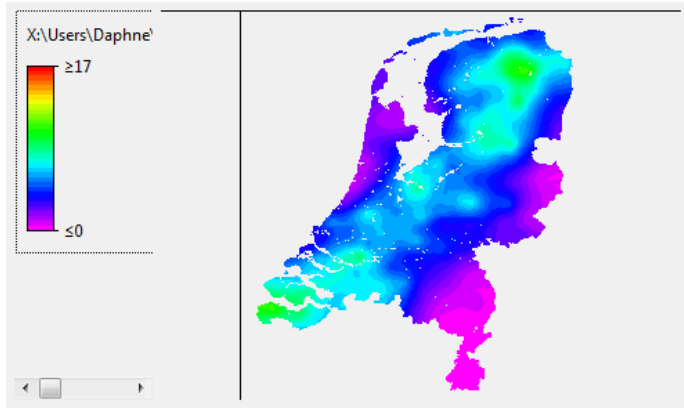


Figure 24: Ordinary kriging for 326 stations on 26 May 2011

### 3.3 Comparison between kriging and inverse distance

When we compare the output results from inverse distance weighted (IDW) and ordinary kriging the most prominent difference is the “bull’s eyes” in the results of IDW where the stations are situated. These circles aren’t so obvious in the kriging output maps. The kriging method tries to create areas with the high values when a couple of near stations have high measurements.

Another noticeable difference between the kriging and IDW maps are the heights of the measurement values. Kriging shows lower estimates, this was already mentioned in section 2.4. Kriging has the tendency to give an underestimation of the ‘true’ values. Nevertheless we think that it gives better results in the area around the stations than IDW.

On the KNMI website we found radar images<sup>[18]</sup> of 26 May; the day for which we already compared the two methods with varying number of rain stations. These can be found in the figures 14, 15, 22 and 24. These are shown in figure 25. These are to some extent comparable to our maps, because the radar images are measured from 10 AM the day indicated till 10 AM the next day. This while the results for 326 stations gives the daily precipitation from 0 till 24 hours on the day indicated and the results for 35 stations are from 8 AM the day indicated till 8 AM the next day. Although this is the case, the radar images of 26 May are comparable with our results from kriging and inverse distance. Notice that light blue in figure 25 stands for approximately 1 mm.

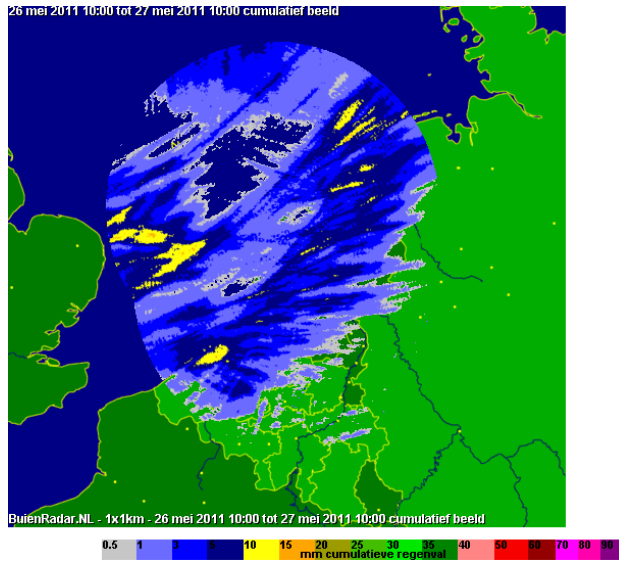


Figure 25: Radar image KNMI website on 26 May 2011

The four figures 26 until 29 show the total sum of rain. As can be seen the inverse methods shows “bull’s eyes” which we have mentioned before. The kriging method gives underestimations where values are high and overestimations where values are low<sup>[5]</sup>, when you compare them to the inverse method. This is why we chose to make the legend for ordinary kriging less variable to make the images more comparable.

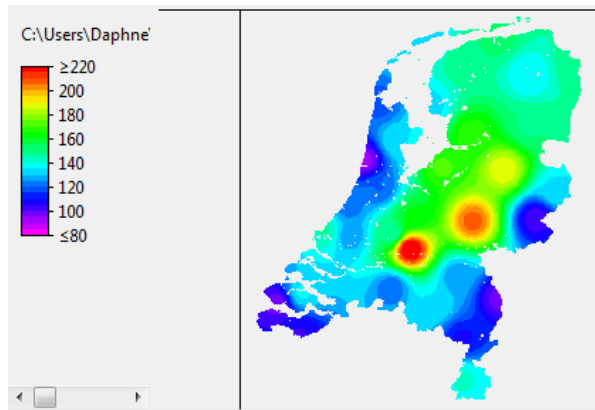


Figure 26: Sum of the precipitation over 100 days using IDW with 326 stations

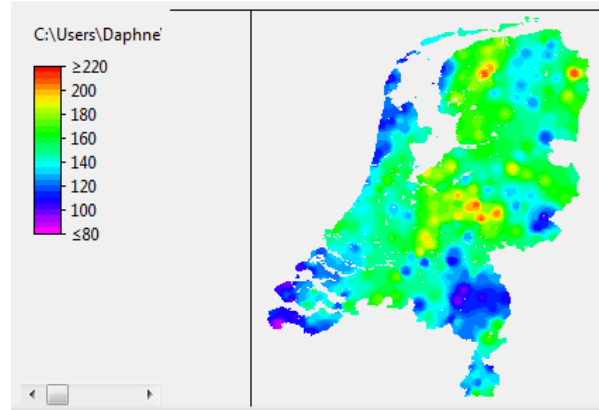


Figure 27: Sum of the precipitation over 100 days using IDW with 35 stations

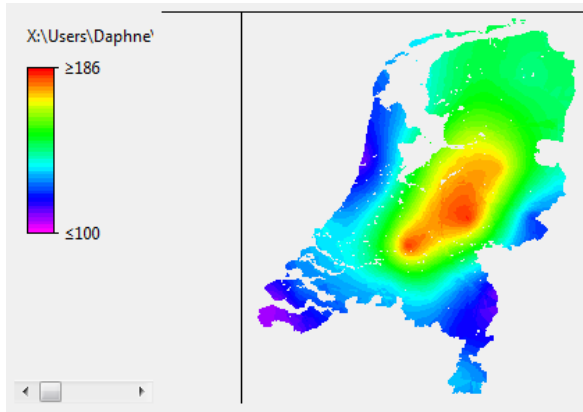


Figure 28: Sum of the precipitation over 100 days using ordinary kriging with 35 stations

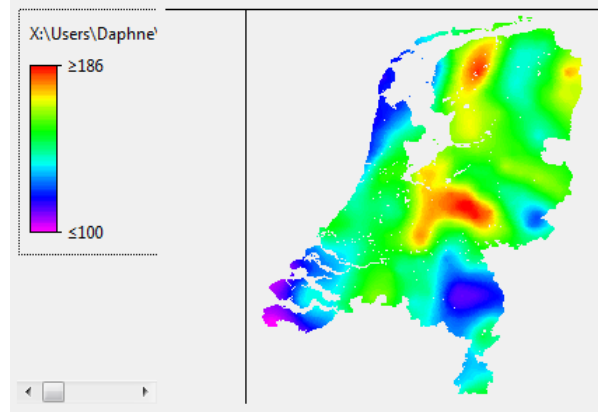


Figure 29: Sum of the precipitation over 100 days using ordinary kriging with 326 stations

When we compare the 35 stations to the 326 there is a noticeable difference. In the North there are shown some red areas for 326 stations and these are not visible for 35 stations. We cannot conclude where this comes from, since there are meteo stations in that area for the 35 stations.

### 3.4 Soil moisture results

The final results are the maps of the soil moisture and fractional evaporation which are based on the maps of precipitation and evaporation an example of the soil moisture can be seen in figure 30. The reason why we ran the model for 100 days is that the soil moisture and fractional evaporation does not depend on one day of precipitation and evaporation input. The drought of the months of this year in March, April and May is still noticeable when you compare them to other years. We won't make comparisons between years, since this is beyond the scope of this paper.

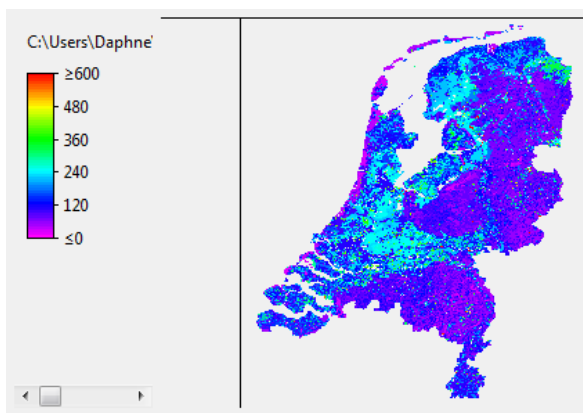


Figure 30: Estimation of the soil moisture with kriging for 326 rain stations on 30 June 2011

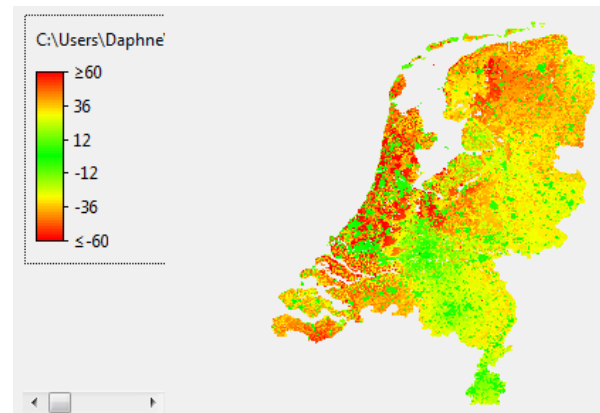


Figure 31: Difference map for soil moisture between 30 June and 31 July with kriging using 35 rain stations

One thing that we can show is the difference between the map of 30 June and the map of 31 July. In July 2011 was a very rainy month. On average 140 mm of rain is fallen this month. If you compare them to the maps shown above which showed the precipitation over 100 days this is obviously a lot more. Figure 31 shows the difference using the kriging method for 35 rain stations.

When we looked at the maps for the different methods we didn't see much of a difference by the eye. This is why we made difference maps. Figure 32 shows the difference between kriging and IDW both for 35 stations on 30 June 2011. As can be seen the biggest estimation error for the IDW method is made by the two stations that were already seen in the precipitation sum maps over 100 days.

In figure 33 the difference between kriging for 35 and 326 stations can be seen. This difference is not very big. The red area in the North shows the biggest difference, which we already couldn't explain. For the rest these maps are comparable to each other. So the use of 35 rain stations instead of 326 rain stations shouldn't give a big error.

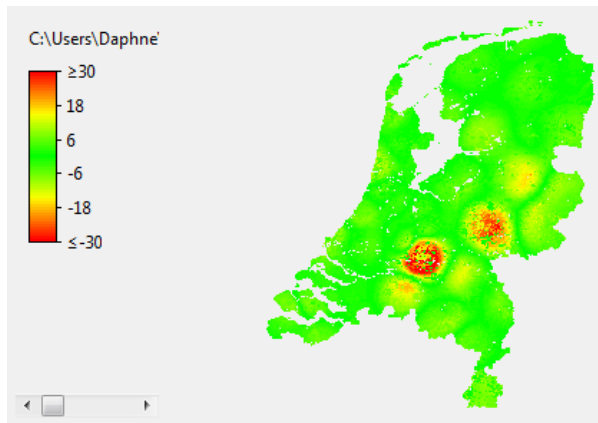


Figure 32: Difference map for soil moisture between kriging and IDW using 35 stations on 30 June 2011.

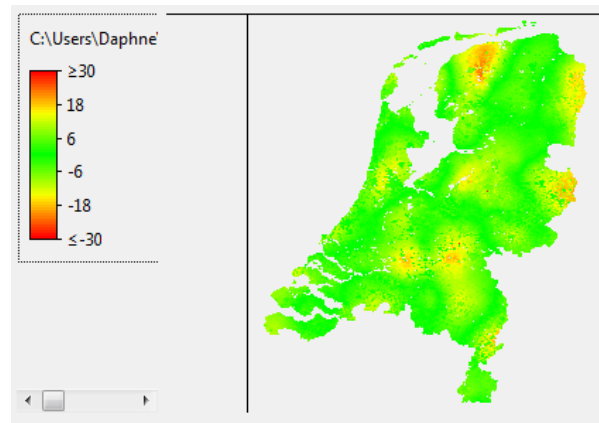


Figure 33: Difference map for soil moisture between kriging for 35 rain stations and 326 rain station on 30 June 2011

### 3.5 Fractional evaporation results

The results for the fractional evaporation also depend on the 100 days of input data, but the output can differ per day. The fractional evaporation depends on the estimated soil moisture and evaporation of one day. In periods of drought the fractional evaporation will be low. In periods where the soil is moist the fractional evaporation will give mostly an evaporation factor of 1, which is 100%. This can be seen in figures 34 and 35. The months March, April and May were very dry which resulted in a very dry soil.

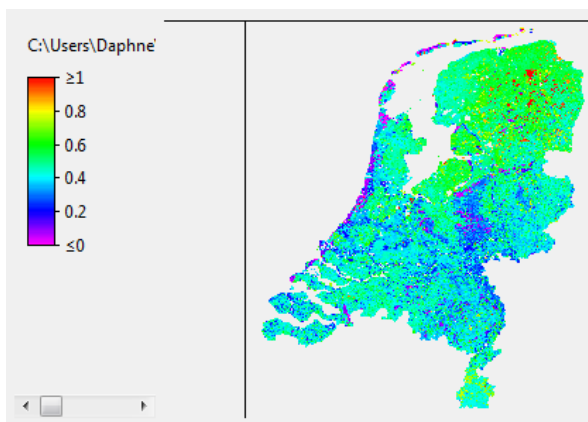


Figure 34: Fractional Evaporation using kriging with 35 rain stations on 4 June 2011

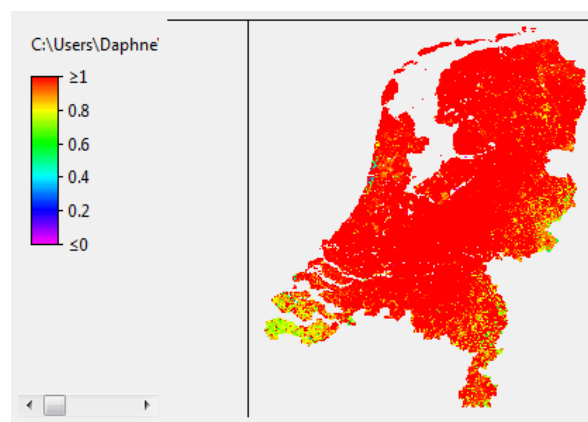


Figure 35: Fractional Evaporation using kriging with 35 rain stations on 30 June 2011

## 4 Discussion and conclusion

The programming tools aren't the most common used tools. This caused some limitations for the model. These and solutions for these shortcomings will first be explained. There were also some errors in data input which will be described. Since this was only a project for four weeks and a model can always be improved, assumptions and possible improvements will be described. Finally conclusions are drawn.

### 4.1 Limitations and shortcomings of the KNMI data

An important input error was the missing station in the text file of the KNMI. In the comments each meteorostation is given by its longitude and latitude. There is no data given for the meteorostation number 265 in Soesterberg, although it is mentioned in the comments in the same file. The program doesn't check at the moment if the meteorostations from the KNMI file match with the station in the coordinates file.

- There aren't up to date values available for the rain stations. These values will be checked each month and this process takes at most three weeks. This means that the data that you can use is at least three weeks old.
- For the 326 rain stations script the function `stns=ALL` doesn't work. This is solved at the moment to give every meteorostation number as an input.
- Another important limitation is the difference between the daily values of the rain stations and the automated stations. The daily values of the rain stations are measurements from 8 AM of the previous day till 8 AM of the day set, while the automated stations have values from 0 to 24 hours of the set day. This makes them hard to compare. A possible solution for this is to subtract hourly values from the automated station of the last 8 hours and add the 8 hours of the next day.

### 4.2 Limitations and shortcomings of PCraster

- PCraster maps can contain a couple of scale types. It can contain a boolean, nominal or ordinal type. For the representation of the meteorostations in the map we would like to use a nominal type. The rain stations have 326 spots. This can't be set to a nominal type since the maximum number for this is 256, instead a scalar type will be assigned. A scalar type can't be used as input for the location of the rain stations in PCraster. The solution that is chosen is to interpolate the data before it is used in PCraster. The inverse distance interpolation method for 326 stations is now calculated in Gstat.
- Another limitation is that an interpolated result map from Gstat can be of a Boolean type. This is the case when the total amount of rainfall measured over the whole area is 0. This type can't be changed to a scalar, because all the values are the same, namely 0. The solution for this problem that we chose is to check whether there wasn't any rainfall measured. If this is the case, than we change one of the measurement location to a very small value. We chose this value to be .00001 mm. This small amount provides the map type to be scalar. One case has not been checked,



Namely when all the stations have exactly the same amount of rainfall. This will probably give an error, but is very unlikely to occur.

### 4.3 Next steps

The model could of course endlessly be improved. The most important steps that will improve this model will be described here. These steps can be seen as the next steps that should be taken to improve the model.

- The most important improvement step that should be preformed is getting the results on-line on the FutureWater website. The model gives realistic results and is thus ready to show its results. The daily results should be updated automatically on the website an example of how this should look like can be seen in figure 36.



Figure 36: Image from the future water website showing an example the computed soil moisture

- The variogram method and parameters are not dynamically fitted. There is a function in Gstat where you can fit a variogram to the input data when initial parameters are chosen. In the program we build this function is not used. This is, because the code that you need to use for this was not found. By using this function better parameters can be used which should result in a better estimations.
- Another important step is to calculate the estimation error. This makes it easier to optimize the parameters and model used for kriging. Since the true data values are not available over the whole area a validation method should be used. The one we recommend is cross validation. Cross validation leaves a couple of points with known values out and then tries to estimate these points.
- For the 326 stations the data input and the coordinates of the rain stations aren't connected to each other. This means that the model assumes that all the data is available, otherwise the



output will give an error. Since this did not happen for the last couple of months we can conclude that the volunteers were very devoted. When one volunteer isn't devoted over a period this should give errors. To prevent this, the input data should be checked, when a rain station doesn't have input over the period you want to calculate over, this station should be removed in the output.

- The model can run for the several methods at the moment. As we can see the kriging method with 326 rain stations gives the best result. A disadvantage of this method is that 326 stations data is not up to date. Since we need to run the model for at least 100 days to get a good estimation of the soil moisture and fractional evaporation we could partly calculate the model for 326 and the rest with 35 stations. When we build in an automatic check through which date the data for 326 rain stations is available we can improve the model.

#### 4.4 Conclusion

The main conclusion is that the kriging method outperforms the IDW method. The IDW method will also give a good approximation of the soil moisture and fractional evaporation, but there will be some over and under estimations where the meteorological stations are situated.

For the kriging method we used two models, the exponential with a nugget of 0, a sill of 4 and a range of 130.000 and the spherical with a nugget of 0, a sill of 10 and a range of 50.000. The exponential method gave better results. This could be further investigated by varying the parameters. The parameters that were chosen could also be better optimized.

When we compare the results for the 35 meteorological stations and 326 rain stations we clearly see a difference in the result maps. When we calculate the soil moisture maps from the input data, the difference between them is not so big. So it is not necessary to use the 326 stations data as input for the model.

When we looked at the radar images of the KNMI website the output of our precipitation map was quite similar. From this we can conclude that our model gives reliable output that can be used for projects where this kind of output can be used for.

## 5 References

### Articles:

- [1] P. Drooger, W. Immerzeel, *Wat is het beste model?*, H2O,4 - 2010
- [2] W. Immerzeel, *Klimaatatlas, ontwikkeling wateroverlastkaarten*, H2O, 10 - 2010

### Books:

- [3] E.H. Isaaks, R.M. Srivastava, *An introduction to applied geostatistics*, Oxford University press, 1989.
- [4] T.C.Bailey, A.C.Gatrell, *Interactive spatial data analysis*, Longman, 1996
- [5] J.M. Schuurmans, *Hydrological now- and forecasting*, A-D druk Zeist, 2008

### Websites:

- [6] [http://nl.wikipedia.org/wiki/Python\\_\(programmeertaal\)](http://nl.wikipedia.org/wiki/Python_(programmeertaal))
- [7] <http://www.knmi.nl/klimatologie/>
- [8] <http://nl.wikipedia.org/wiki/Rijksdriehoeksco%C3%B6rdinaten>
- [9] <http://www.bisolutions.us/A-Brief-Introduction-to-Spatial-Interpolation.php>
- [10] <http://en.wikipedia.org/wiki/Kriging>
- [11] <http://www.gstat.org/manual/node30.html>
- [12] <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//003100000023000000.htm>
- [13] <http://www.gstat.org/s.html>
- [14] <http://pcraster.geo.uu.nl/>
- [15] <http://en.wikipedia.org/wiki/Variogram>
- [16] [http://www.xmswiki.com/xms/GMS:Variogram\\_Editor#Lags](http://www.xmswiki.com/xms/GMS:Variogram_Editor#Lags)
- [17] <http://www.otago.ac.nz/sas/stat/chap34/sect12.htm>
- [18] <http://www.buienradar.nl/historie.aspx>
- [19] <http://www.futurewater.nl/watermaps/vochtmonitor>

## 6 Appendices

### 6.1 Starting the program

Here is an explanation about the steps you need to take before you can run the program.

First you'll need to set the whole folder, FW\_program in a directory of your choice. Then a couple of programs need to be installed. These are:

- Download python at [www.python.org/getit/](http://www.python.org/getit/) and install this
- Download Gstat at [www.gstat.org/download.html](http://www.gstat.org/download.html) and put the `gstat.exe` and install this
- Copy the `gstat.exe` to the FW\_program folder
- Download PCRaster at <http://pcraster.geo.uu.nl/download> and install this

Now the program should work. If you want to run the programming we first have to set some parameters. To change these parameters you should first open the file `start_hydroS_model.py` as a text file from the FW\_program directory, preferably use notepad++.

In the method `getDataFromKNMI_35stations()` and `getDataFromKNMI_326stations()` you can set the start and end date. Then in the cmd file from PCRaster from the interpolation technique you must set the right number of days that you want the model to run. Unfortunately there was no option to automatically set the right number of days using the start and end date.

In the `Start()` method you can decide which interpolation methods you want to use. By marking the ones you don't want to use as comment.

When you want to run the program you must first check whether there is data of an old simulation in the directories `data/output_kriging`, `data/output_inverse326`, `data/output_kriging326`. If there is, you must delete them before running the program.

To run the program you can open your command prompt, then go to the directory where you have put the `start_hydroS_model.py` file. Then just type in `start_hydroS_model.py` to run the model.

### 6.2 Implementation of the program

#### 6.2.1 Download of KNMI script using wget

To automate the process of retrieving the daily evaporation and precipitation data the Wget tool is used. To use this you need to download this program and install it. The executable file must be in the same folder from where the script called. The script file is executed from our python script by the following command for the 35 meteo stations:

```
os.system("wget -O directory/data_knmi.txt --post-  
data=\"stns=ALL&vars=RH:EV24&byear=%s&bmonth=%s&bday=%s&eyear=%s&emonth=%s&eday=%s\"")
```

`http://www.knmi.nl/klimatologie/daggegevens/getdata_dag.cgi" % (byear, bmonth, bday, eyear, emonth, eday) )`

Where the parameters: (byear, bmonth, bday, eyear, emonth, eday) need to be given as input.

For the 326 meteorostations the code needs a little adjustment. The URL is:

`http://www.knmi.nl/klimatologie/monv/reeksen/getdata_rr.cgi` and 'vars' can't be specified. Another thing to mention is that the function `stns=ALL` doesn't work so all 326 stations need to be specified.

More information about this can be found on <http://www.knmi.nl/klimatologie/uurgegevens/scriptxs-nl.html#uur>

### 6.2.2 Using Gstat for kriging and inverse distance

Gstat can be used for the two interpolation techniques used, IDW and ordinary kriging. The following file is written for the implementation of ordinary kriging, every line is explained:

```
1: data(evap): 'data/evapdata.txt', x=1, y=2, v=3, radius = 75000, max = 20, min = 15, force;
2: variogram(evap): 1 Nug(0) + 4 Exp(130000);
3: mask: 'rainstat_input_kriging.asc';
4: set fit = 3;
5: predictions(evap): 'result.map';
```

The first line sets the data used, 'data/evapdata.txt'. With `data(evap)` 'evap' you can set the variable name that can be used. The x, y and v are: x-coordinate, y coordinate and value. The radius is the radius that that is used to search for points. With min and max you can set a number of points that should at least or at maximum be used.

In the second line we defined that we want to use a variogram voor de evap data, with the parameters we want to use. The third line shows where the known locations on the grid are, and all the unknown locations that you want to estimate should be 0.

Line 4 says 'set fit=3'. This defines a fitting method that should fit the variogram to the input data, but as far as we have tested this, this didn't work. This is something that needs to be checked. The last line gives the predictions for all the unknown location in a map that can directly be used as an input for the HydroS model.

In reference 11 you can find more examples for implementing interpolation methods. A useful example to use is example 1. This starts the variogram modelling user interface where you can fit the variogram to the data and see the effects of the various parameters and models.