BMKG

Royal Netherlands
Meteorological Institute
Ministry of Infrastructure and the Environment

PUSAIR

Deltares
Enabling Delta Life

JOINT COOPERATION PROGRAMME

Component C2: Water management datasets for river basins

Document C2.2

PPPs Training on the *Use of Delft-OMS and set-up of hydro-meteorological database and interface*

Bandung and Jakarta

28-29 July and 1-4 August 2011

Project: 1201430.000

Client: Water Mondiaal
Partners for Water
Royal Netherlands Embassy in Jakarta

Period: January 2011 – March 2013

BMKG

Royal Netherlands
Meteorological Institute
Ministry of Infrastructure and the
Environment

PUSAIR

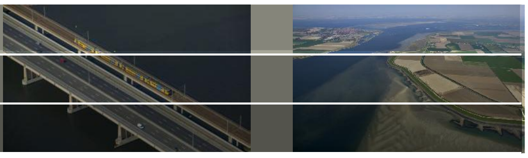
Deltares
Enabling Delta Life

Table of Contents

Program 3rd JCP-training

28-29 July 2011 – Bandung
1 – 4 August 2011 - Jakarta

Ronald Vernimmen
ronald.vernimmen@deltares.nl



Program

Bandung:

1. 28 July: Introduction to PCRaster

2. 29 July: Hydrometeorological reporting accessible via a clickable map interface
question: BMKG year reports?

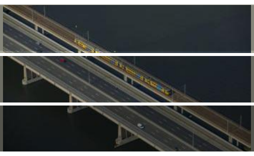
Jakarta:

3. 1 August: Application of PCRaster (climatic maps: Schmidt-Ferguson)
questions: official publication, jpgs of maps, legenda colours?
other maps currently in use?
ideas or interest for other PCRaster applications?

4. 2 August: Implementation SPI for pilot area Pemali-Comal

5. 3 August: Working with NetCDF and grib data formats
question: additional NetCDF files (Wido had question earlier)

6. 4 August: Catchment water balance as check for quality discharge data



Schedule 29 July

Hydrometeorological reporting accessible via a clickable map interface

0900 – 0915 Update of DEWMS

0915 – 1000 Demonstration of prototype hydrometeorological reports and clickable map + explanation manual + evaluation form

1000 – 1015 coffee break

1015 – 1100 presentation Pak Irfan: hydrological database and webservices

1100 – 1230 Exercises 1 + 2

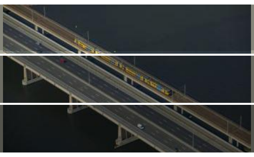
1230 – 1330 Lunch

1330 – 1415 presentation Ade: SIG webservices

1415 – 1500 Exercises 3

1500 – 1515 coffee break

1515 – 1700 Exercises 3 + 4



Update configuration DEWMS

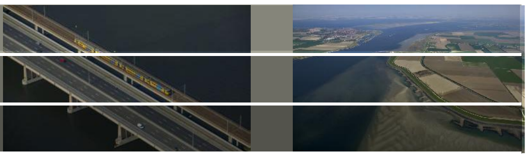
1. new metadata file (additional columns as well: WS, warning levels,etc.)
2. import PusAir telemetry data (not automatic yet, via Hec-DSS database)
3. updated the explorer (rivers + DEM 900 m for whole of Indonesia) + link to online wiki
4. corrected Oldeman script

(5. implemented Schmidt-Ferguson classification) - EXERCISE

6. imported 1990-2009 Pemali Comal daily P and calculate SPI (Python script)
7. implement automatic TRMM download (via Python script and Windows task scheduler)
8. implement prototype of hydrometeorological annual reports

All changes can be found in the “logbook changes configuration.txt” in the <DEMWS> directory!

Schedule 1 August



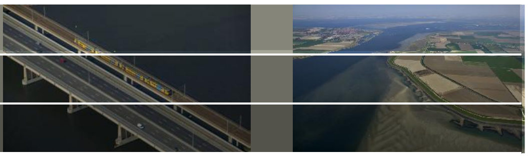
Application of PCRaster

1000 – 1015 Feedback training Friday, collect evaluation forms, update manual (Provinces shape layers, closing log box, data viewer panel to remove redundant information)
Questions:
BMKG year reports?
Presentation on downscaling?
NetCDF example?
Publication Schmidt-Ferguson, other maps?
invite BMKG staff generating SPI maps (tomorrow afternoon)

1015 – 1045 Introduction to Schmidt - Ferguson

1045 – 1500 Using PCRaster create Schmidt – Ferguson map for Indonesia

Schedule 2 August



Application of PCRaster

0900 – 0915 Feedback Schmidt Ferguson training (gdal_translate options explained:
http://www.gdal.org/gdal_translate.html) + nominal vs scalar
Questions:
Evaluation forms course Friday
BMKG year reports?
Presentation on downscaling?
NetCDF example?
Publication Schmidt-Ferguson, other maps?
invite BMKG staff generating SPI maps (tomorrow afternoon)
new update software + configuration!

0915 – 0945 Explain steps SPI today
create timeseries (BMKG format to CSV format)
import timeseries
compare BMKG series with PusAir series
use PusAir series to create SPI (Excel method)
compare PusAir Excel method with python method
other interpolation methods / ***filling missing values***

1400 – 1500 Discussion, how to expand to whole of Indonesia?

date format of csv file

date format is:

yyyy-mm-dd <space> hh:mm:ss

Schedule 3 August

Working with NetCDF and grib data formats

0930 – 1000 Feedback Implementation SPI Pemali Comal (csv import; use for all further data import, or via Data Editor) SPI comparison Excel vs Python, Griddisplay
Presentation on downscaling?
NetCDF example?
1000 – 1500 Use of software panoply to visualize NetCDF and grib data
import NetCDF file 3B43
Exercise: compare 3B43 with 3B42RT product
import grib file (ECMWF seasonal forecast, P only)
resample data from 0.5 degree to 0.25 degree
Exercise: compare ECMWF seasonal forecast (Dec 10 – Jun 11) with 3B42RT
Exercise: import different parameter ECMWF
Exercise: import your own NetCDF

???? downscaling presentation by Mamie and/or Leni

Results BMKG vs PusAir comparison 2 August

Conclusions:

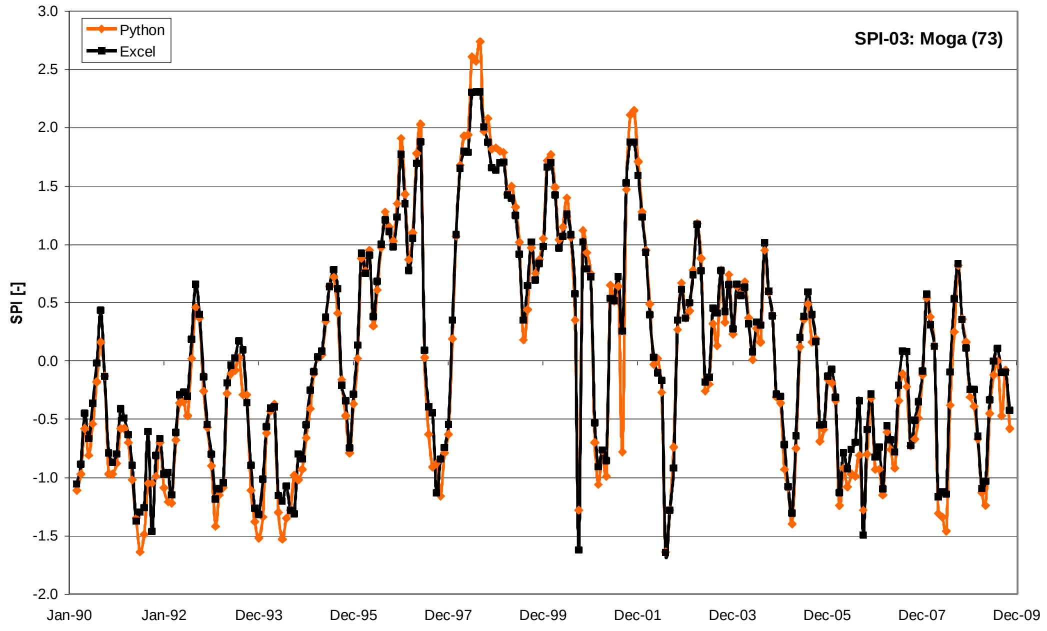
(1) Data not always the same.

(2) Date shift occur, typing errors, or complete different data altogether (data from different station, data from different month?

(3) date format of the CSV file is very important!!! **(yyyy-mm-dd hh:mm:ss)**

Results SPI comparison 2 August

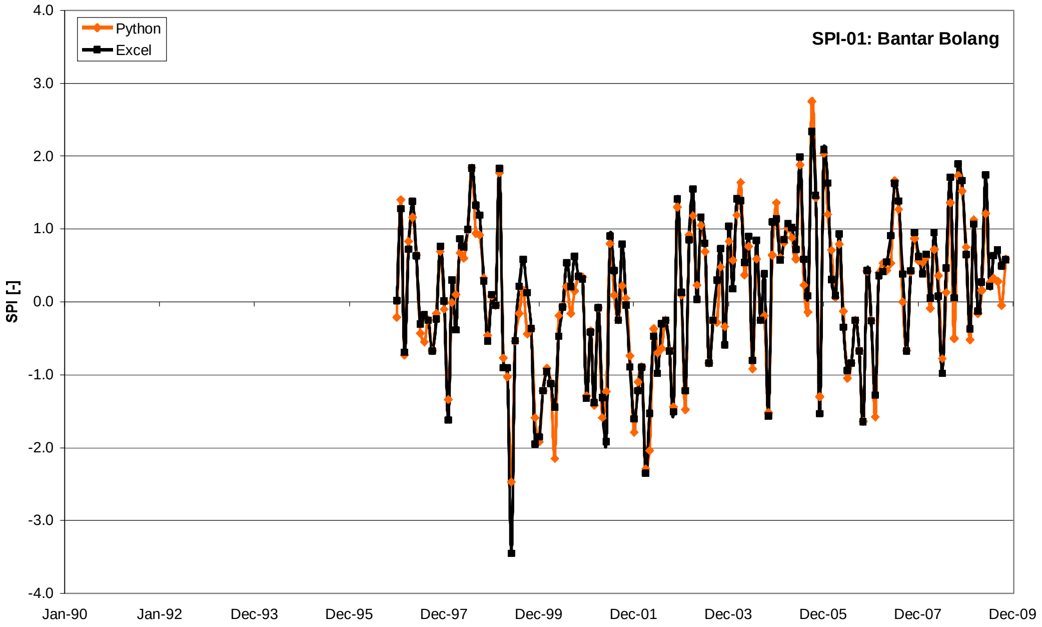
2 results: Moga (SPI-03) and Bantar Bolang (SPI-01)



DEWMS method tends to generate slightly more extremes

Results SPI comparison 2 August

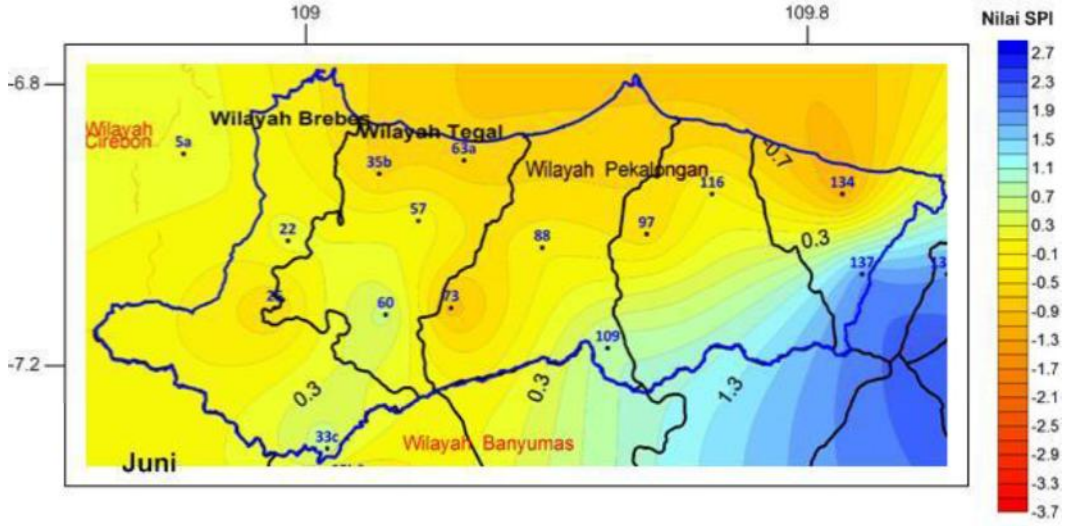
2 results: Moga (SPI-03) and Bantar Bolang (SPI-01)



DEWMS method tends to generate slightly more extremes

SPI–03: April – June for Pemali Comal

Using kriging interpolation (linear variogram, slope = 1, aniso = 1,0) implemented using Surfer software

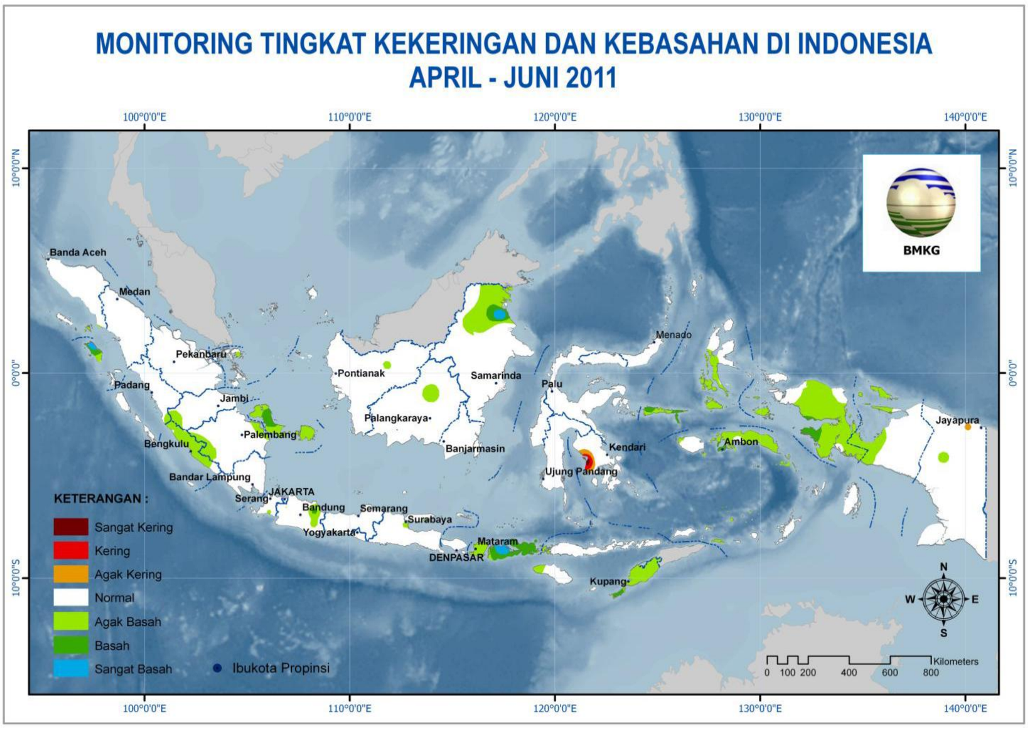


Kriging interpolation will be implemented in DEWMS next (currently Thiessen used)

source: PusAir

SPI–03: April – June for whole Indonesia

Questions: which interpolation method used in SPI Indonesia below >> inverse distance with standard settings (no range or number of stations to be included)



Station list available (coordinates?)

Monthly values can be included in DEWMS?

source: http://www.bmkg.go.id/BMKG_Pusat/Klimatologi/Indeks_Presipitasi_Terstandarisasi.bmkg

Schedule 4 August

Catchment water balance as check for quality discharge data

0930 – 1000 Feedback NetCDF + grib data (discuss use of wget)

1000 – 1500 **automatic import TRMM (via Windows Task Scheduler)**

rating curves
discharge m^3/s to m^3
Exercise 1: compare discharge in mm to precipitation in mm
Exercise 2: import another ECMWF parameter (3 August)
Exercise 3: import parameter from NetCDF file Wido (3 August)

???? downscaling presentation by Mamie and/or Leni

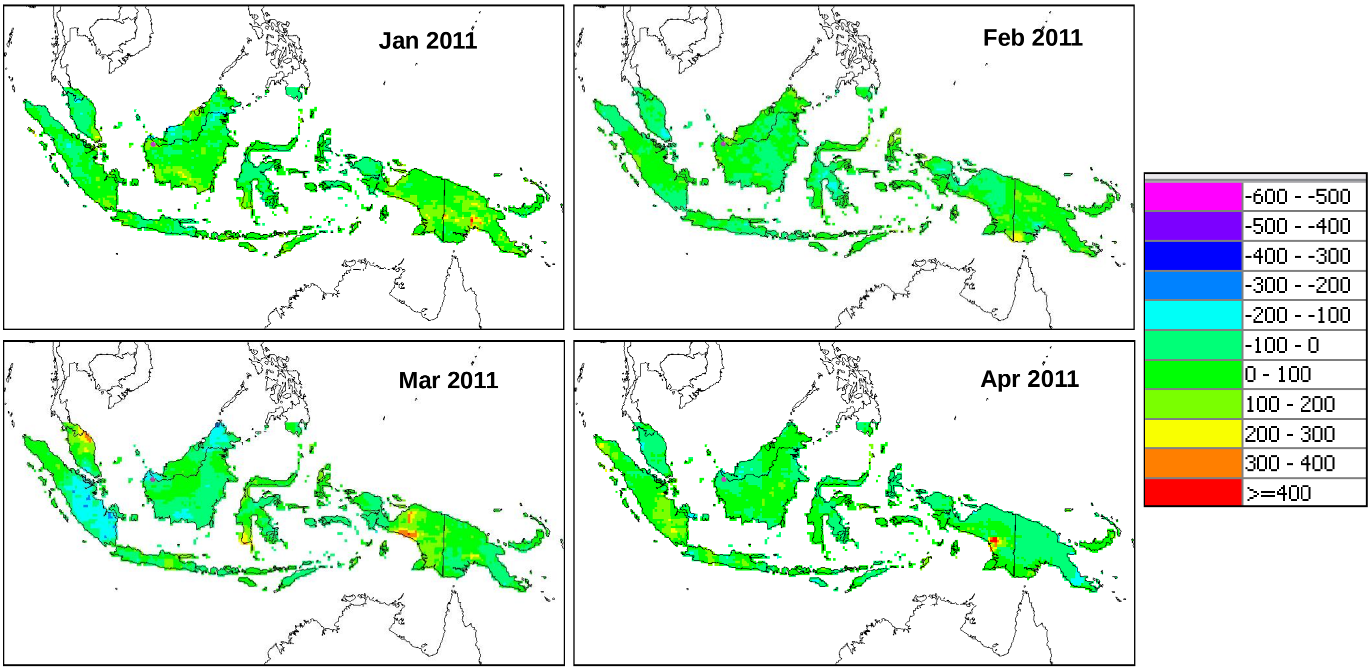
1430 – 1500 further planning and closing

General notes 3 August

variables / filenames of maps in PCRaster can not start with value, always with character!

Difference Maps – 3B43 – 3B42RT bias corr.

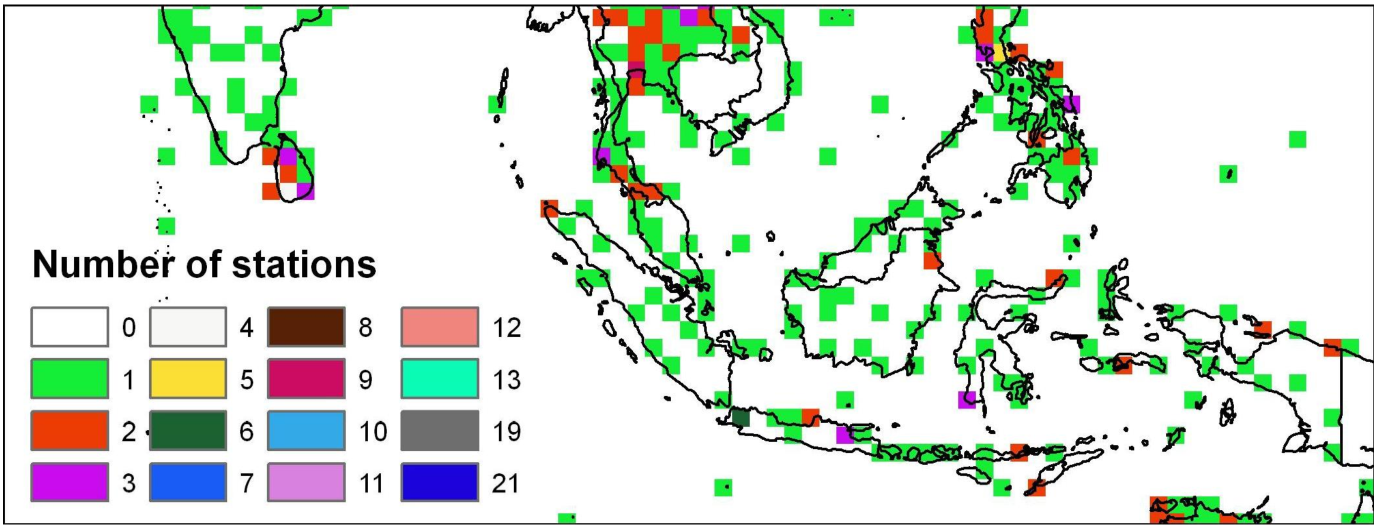
Now also included in the DEWMS!



Conclusion: Apart from March 2011 (surprisingly) not much difference between the 3B43 and bias corrected 3B42RT product! **STILL NEED TO CHECK IF THIS IS ALSO THE CASE FOR OTHER YEARS AND FOR ALL SEASONS!**

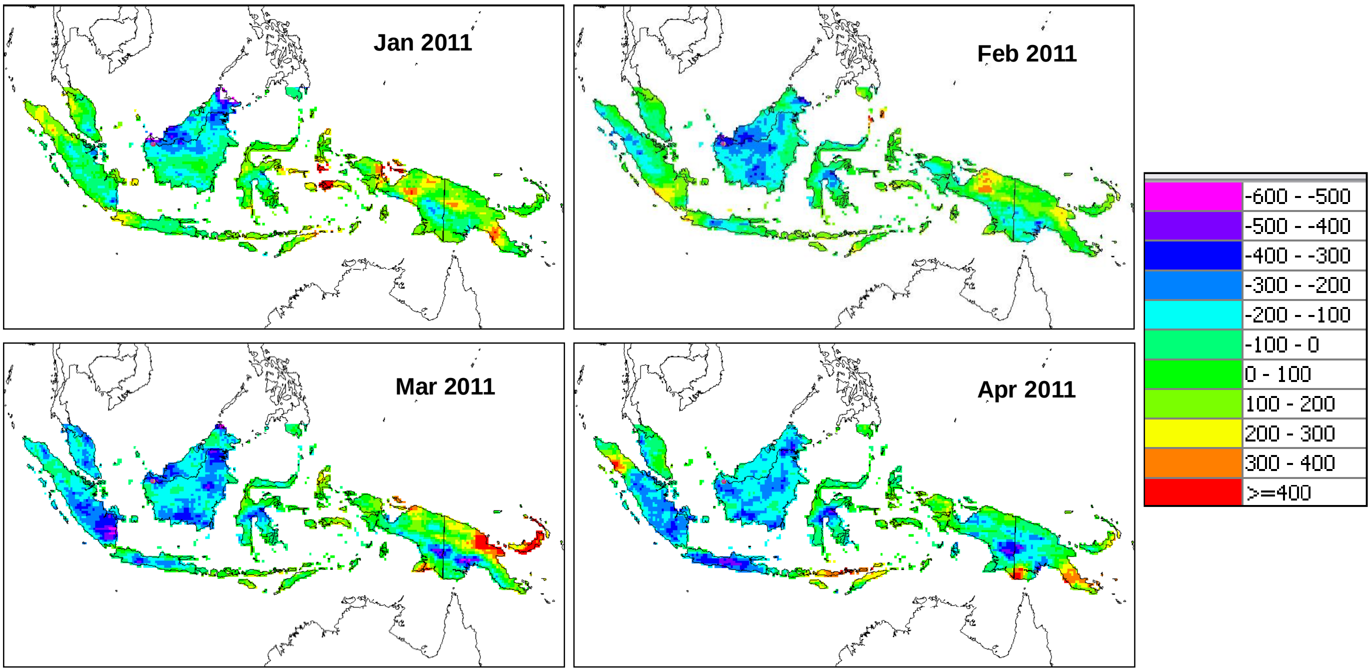
Station coverage GPCP Indonesia

Distribution of stations GPCP (or GPCC?) monitoring product January 2008 used in deriving 3B43 product



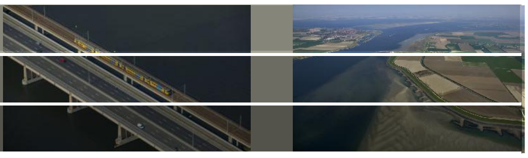
Difference Maps – ECMWF – 3B42RT bias corr.

Now also included in the DEWMS!



Conclusion: Skill for Sumatra and Java reasonably good for 2nd month (Dec. 2010 not shown here yet). Forecast severly underestimates on Borneo and later in the forecast also in Sumatra and Java. Skill clearly deteriorates with longer lead times!

Announcements



new updates of the DEWMS during training:

1. fixed the tooltip issue for meteorological stations in reports
2. fixed the heading of the precipitation timeseries chart in the reports
3. fixed typing error in Oldeman workflow
4. added SIG-SDA river layers + lakes + irrigation
5. added Kecamatan + Desa layers
6. fixed error in IdMapping SPI, SPI workflow did not run
7. added calculation of difference maps 3B43 and ECMWF with 3B42RT bias corr.
8. implemented rating curves and conversion of discharge m^3/s to mm

Planning until next visit (half September)

Everybody in close coordination with Bayu and Mamie:

1. continue data collection Pemali Comal (P until recent), fill missing years, higher resolution data (hourly as well), climatic stations! (evaporation calculation!), water level data, rating curve data points, update the metadata file and check position of meteorological and hydrological stations within the catchments

start data collection for following WS:

2. Jratunseluna (Kali Garang, FFWS) (hydrological data with Oky)
3. Bengawan Solo (IFAS-ICHARM project) (hydrological data with Oky)

expansion of SPI calculation:

4. AWS stations BMKG for SPI implementation whole Indonesia (via server?)

Koninklijk Nederlands Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Deltares
Enabling Delta Life

PCRaster course

Daniel Tollenaar
Neeltje Goorden

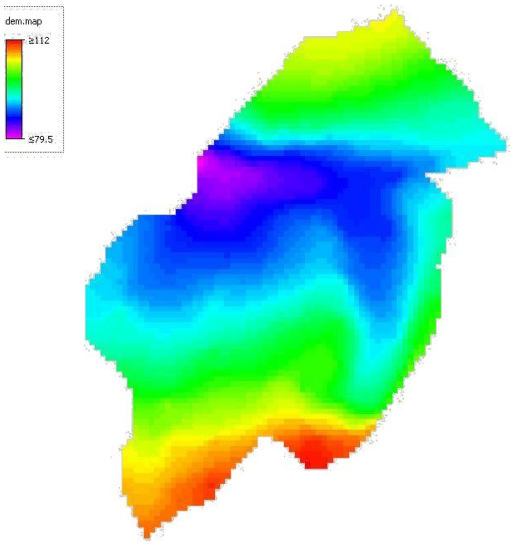
Preface

Objective workshop:

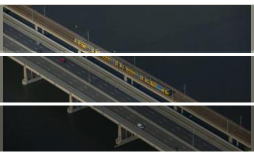
Building GIS models in a simple model environment => using **PCRaster**

PCRaster: developed at the University of Utrecht by W.P.A. Van Deursen and C.G. Wesseling

In this workshop we focus mainly on **land subsidence modelling** in PCRaster

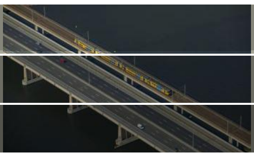


Contents



1. Introduction
2. Software installation
3. Raster data in PCRaster
4. Tables in PCRaster
5. Time series in PCRaster
6. Functions
7. Dynamic modeling

Introduction



Some PCRaster characteristics:

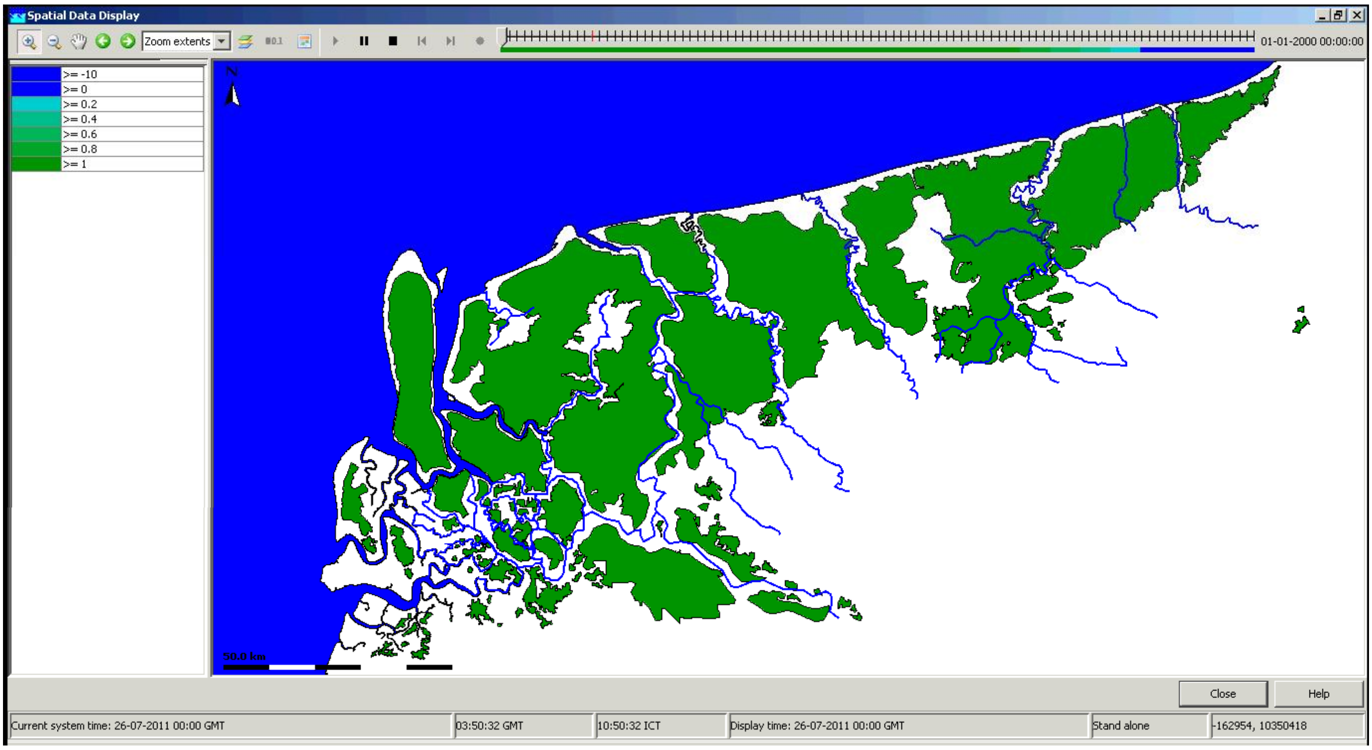
- Uses **raster data** as spatial data (.map files)
- Many tools to edit your raster data. For example filling up gaps by missing data
- Can be used to build cartographic (**static**) and **dynamic** models. For example:
    - Static: Maximum water level state at a specific time instant.
    - Dynamic: Water level over time

Some advantages:

- **<u>It is free!!!</u>**
- Easy scripting
- Compatible with Delft-OMS

Introduction

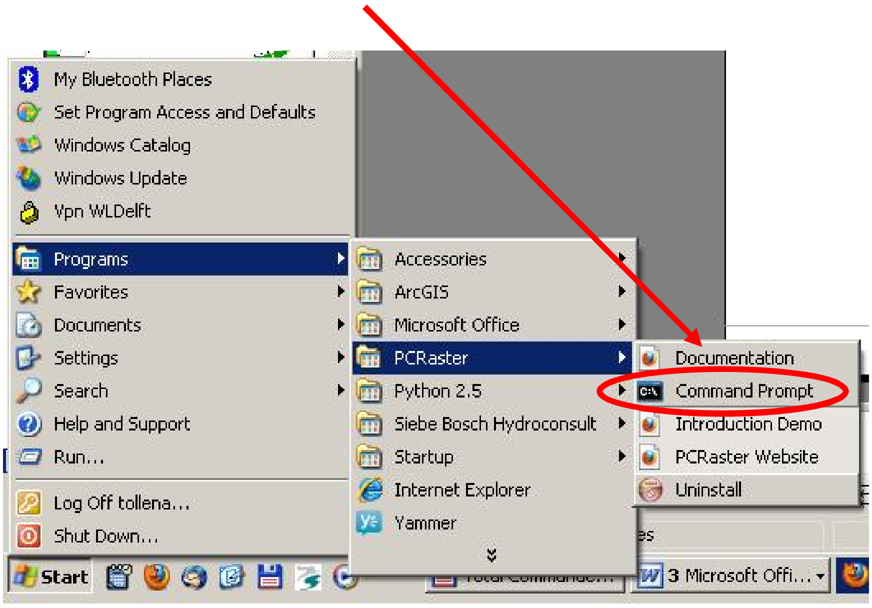
Dynamic model example – Subsidence model of Rajang Delta in Delft-OMS



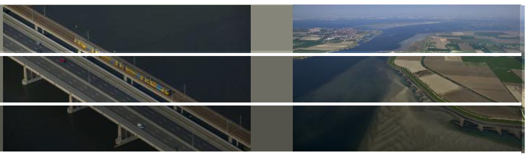
Current system time: 26-07-2011 00:00 GMT | 03:50:32 GMT | 10:50:32 ICT | Display time: 26-07-2011 00:00 GMT | Stand alone | -162954, 10350418

Software Installation

1. Execute and install PCRaster-3.0.0-beta-091201.exe in the **PCRaster** directory
2. Open the **Command Prompt** via Start



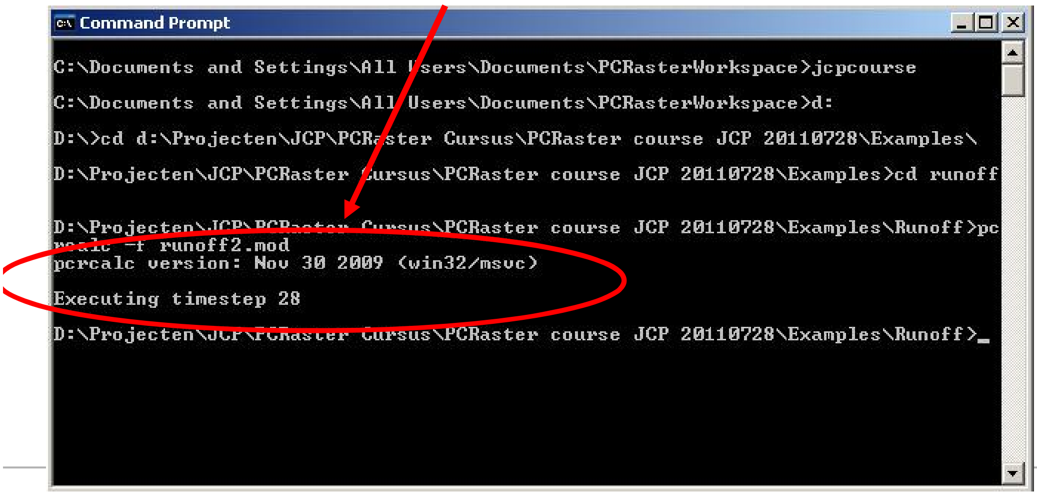
Software Installation



3. Check if software is working. Go with DOS commandos (cd: …\...) to the directory *…\examples\runoff* and enter:

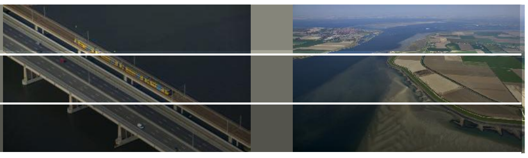
pcrcalc –f runoff2.mod   <enter>

If the model executed 28 timesteps, it is installed OK!



```
C:\Documents and Settings\All Users\Documents\PCRasterWorkspace>jcpcourse

C:\Documents and Settings\All Users\Documents\PCRasterWorkspace>d:

D:\>cd d:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Examples\

D:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Examples>cd runoff

D:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Examples\Runoff>pc
rcalc -f runoff2.mod
pcrcalc version: Nov 30 2009 (win32/msvc)

Executing timestep 28

D:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Examples\Runoff>_
```

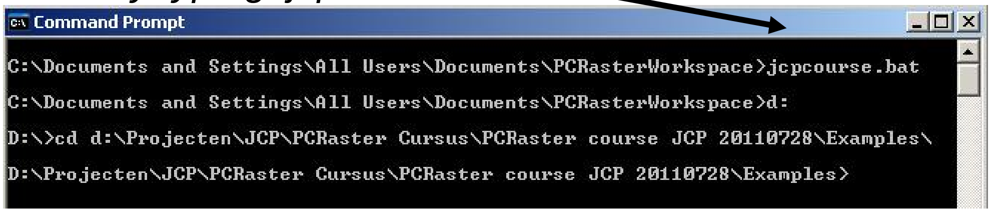
Software Installation



Tip: When opening the **Command Prompt**, the **PCRasterWorkspace** will appear. Creating a batch file referring to the directory *…\examples* makes life easier!!!

Example of batch file (files\batch files\jcpcourse.bat):

- *Copy the jcpcourse.bat to the PCRasterWorkspace map*
- *Open the jcpcourse.bat file and you maybe have to change the path name. It depends on where you have saved the PCRaster course.*
- *Go to the PCRaster command prompt and run the batch file by typing: jcpcourse.bat*



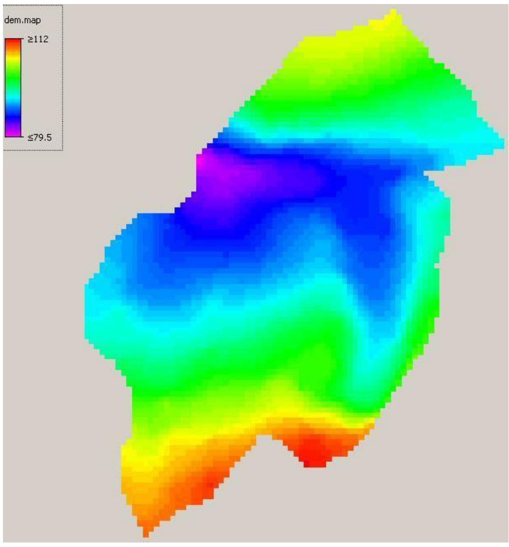
```
C:\Documents and Settings\All Users\Documents\PCRasterWorkspace>jcpcourse.bat

C:\Documents and Settings\All Users\Documents\PCRasterWorkspace>d:

D:\>cd d:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Examples\

D:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Examples>
```

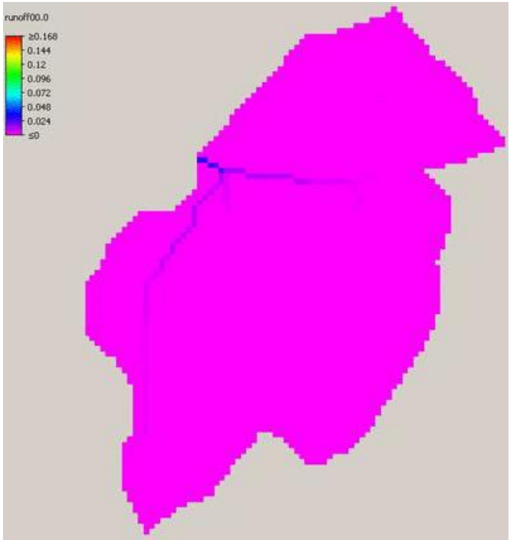
Map files – Static and dynamic map files

PCRaster uses Map files: spatial data in raster format.

Two types of maps in time:

Static maps	**Dynamic maps**
(Data constant at a specific time instant)	**(Data changes in time)**

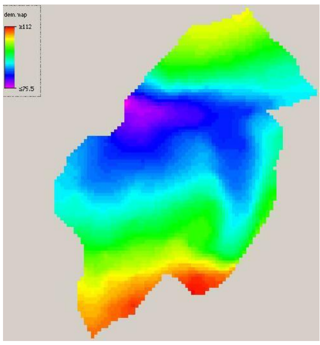




Map files – Static and dynamic map files

Static maps

- Filename can be chosen yourself (not depend
  format, like number of letters in name)
- Extension of file = *.map (for example: runoff.map)

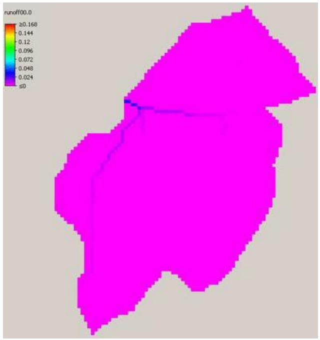


For example:

DEM.map = file with runoff at certain time instant

Dynamic map series

- Extension = time step (=3 numbers)
- Fixed name length = 8 characters



For example:

runoff00.001 = file with runoff at timestep 1

runoff00.002 = file with runoff at timestep 2

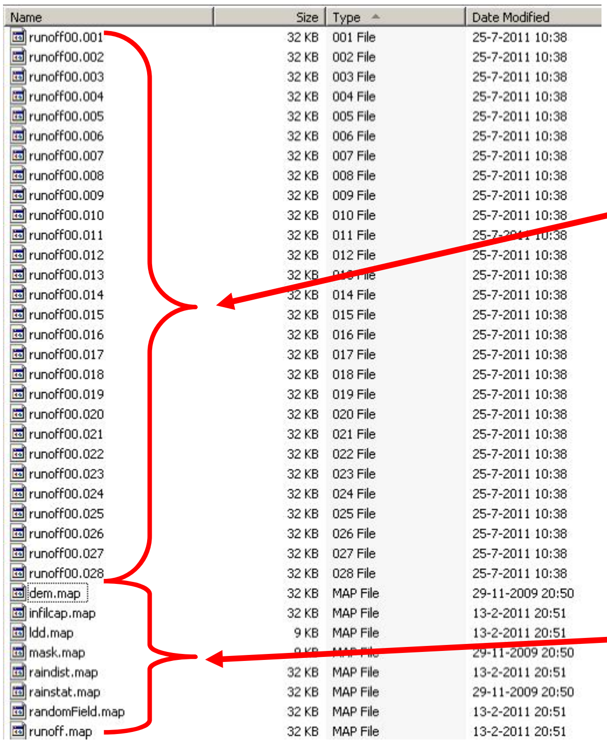
….

runoff00.028 = file with runoff at timestep 28

8 characters

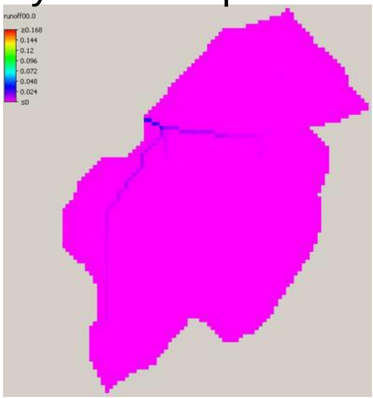
Map files – Static and dynamic map files

Lets explore *examples\runoff* (**after execution!!**)

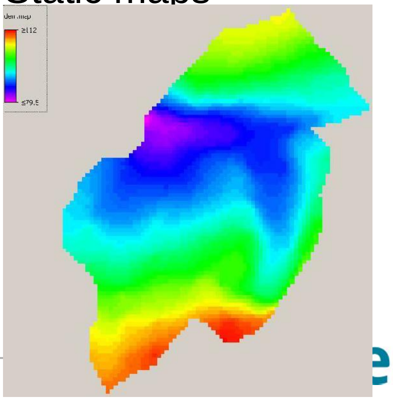


Name	Size	Type	Date Modified
runoff00.001	32 KB	001 File	25-7-2011 10:38
runoff00.002	32 KB	002 File	25-7-2011 10:38
runoff00.003	32 KB	003 File	25-7-2011 10:38
runoff00.004	32 KB	004 File	25-7-2011 10:38
runoff00.005	32 KB	005 File	25-7-2011 10:38
runoff00.006	32 KB	006 File	25-7-2011 10:38
runoff00.007	32 KB	007 File	25-7-2011 10:38
runoff00.008	32 KB	008 File	25-7-2011 10:38
runoff00.009	32 KB	009 File	25-7-2011 10:38
runoff00.010	32 KB	010 File	25-7-2011 10:38
runoff00.011	32 KB	011 File	25-7-2011 10:38
runoff00.012	32 KB	012 File	25-7-2011 10:38
runoff00.013	32 KB	013 File	25-7-2011 10:38
runoff00.014	32 KB	014 File	25-7-2011 10:38
runoff00.015	32 KB	015 File	25-7-2011 10:38
runoff00.016	32 KB	016 File	25-7-2011 10:38
runoff00.017	32 KB	017 File	25-7-2011 10:38
runoff00.018	32 KB	018 File	25-7-2011 10:38
runoff00.019	32 KB	019 File	25-7-2011 10:38
runoff00.020	32 KB	020 File	25-7-2011 10:38
runoff00.021	32 KB	021 File	25-7-2011 10:38
runoff00.022	32 KB	022 File	25-7-2011 10:38
runoff00.023	32 KB	023 File	25-7-2011 10:38
runoff00.024	32 KB	024 File	25-7-2011 10:38
runoff00.025	32 KB	025 File	25-7-2011 10:38
runoff00.026	32 KB	026 File	25-7-2011 10:38
runoff00.027	32 KB	027 File	25-7-2011 10:38
runoff00.028	32 KB	028 File	25-7-2011 10:38
dem.map	32 KB	MAP File	29-11-2009 20:50
infilcap.map	32 KB	MAP File	13-2-2011 20:51
ldd.map	9 KB	MAP File	13-2-2011 20:51
mask.map	9 KB	MAP File	29-11-2009 20:50
raindist.map	32 KB	MAP File	13-2-2011 20:51
rainstat.map	32 KB	MAP File	29-11-2009 20:50
randomField.map	32 KB	MAP File	13-2-2011 20:51
runoff.map	32 KB	MAP File	13-2-2011 20:51

Dynamic maps



Static maps



Map files – Visualization

Visualisation of map files

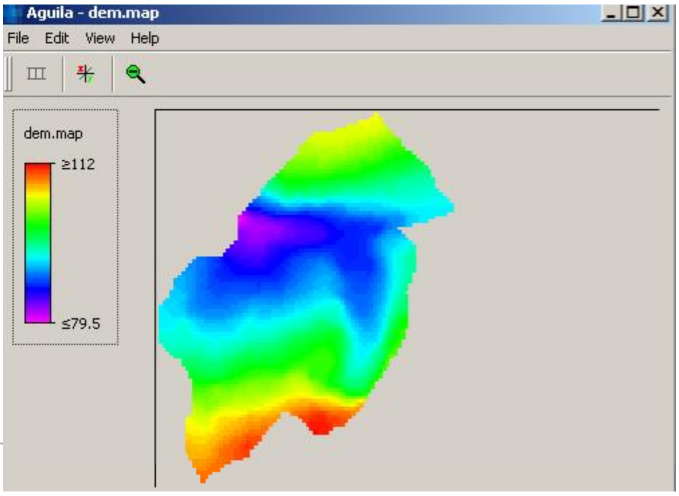
Options:

- ArcGIS 9.3: =>add data button: load: *.map files
- Aguila: The PCRaster internal viewer installed on your computer in PCRaster\apps folder: View the DEM in the runoff example:

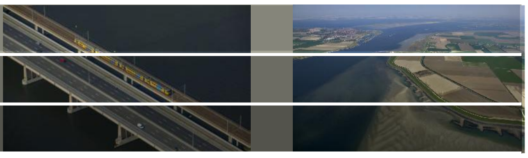
  Type in command prompt: aguila dem.map <enter>

Tip: You can use Aguila as the default program to open .map files. Aguila can be found in the directory *…\PCRaster\apps*

Note: dynamic maps cannot be "played" within Aguila, but they can within Delft FEWS!!!



Map files – Data types

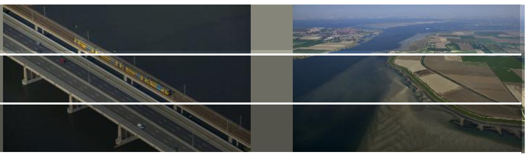


Summary of data types:

Data type	Discription	Domain
Boolean	**True/False data**	**0 (false), 1 (true)**
Nominal	**Classified, but no order**	**0…255, whole values**
Ordinal	Classified and order	0…255, whole values
Scalar	**Continuous, lineair**	**-10^37 to 10^37, real values**
Directional	Continuous, directional	0..2Pi (radians) or 0 to 360 (degrees), -1 (no direction)
Ldd	Local drain direction to neighbour cell	1…9 (codes of drain directions)

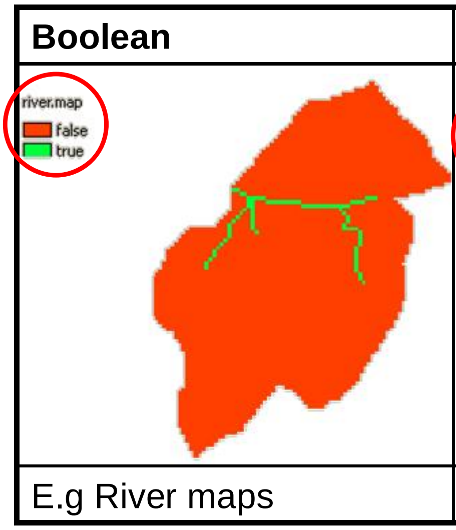
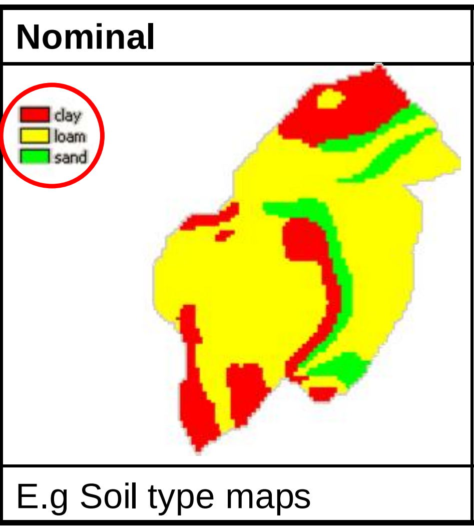
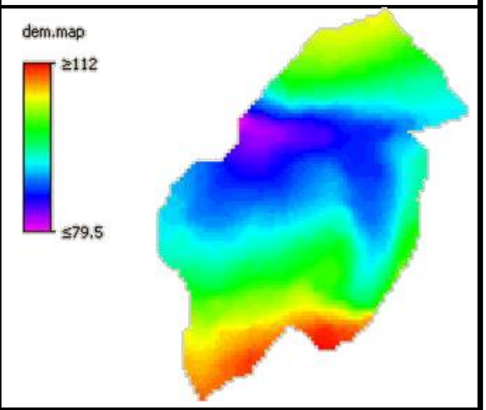
The blue coloured data types are the most likely used

Map files – Data types



- **Different data type** of maps in PCRaster. Important to choose the right data type.

  For example: The dem.map should have the **scalar** data type, since it represents a continuous linear phenomenon:

Boolean	**Nominal**	**Scalar**
		
E.g River maps	E.g Soil type maps	E.g Digital Elevation Models

Note: with commando: **legend** (e.g. *legend soil.map <enter>*) legend **names** can be given for the values for **boolean**, **nominal** or **ordinal** data.

Map files – Data conversion

Why data conversion?

PCRaster can only read *.map files (for raster data) => You need to`convert files to this format

Procedure to convert ascii grid files (ESRI) to PCRaster input

> **Use program asc2map** to create from your *.asc file a *.map file. This is the file with all the data of your ascii file (it does not have a location and extent, like cell size)

> **Use program mapattr** to make a clone map which defines the location and the extent of your map.

Now we will create: "dem.map" (in…*\examples\createdem)*

Deltares

Map files – Data conversion

Procedure: creating "dem.map"

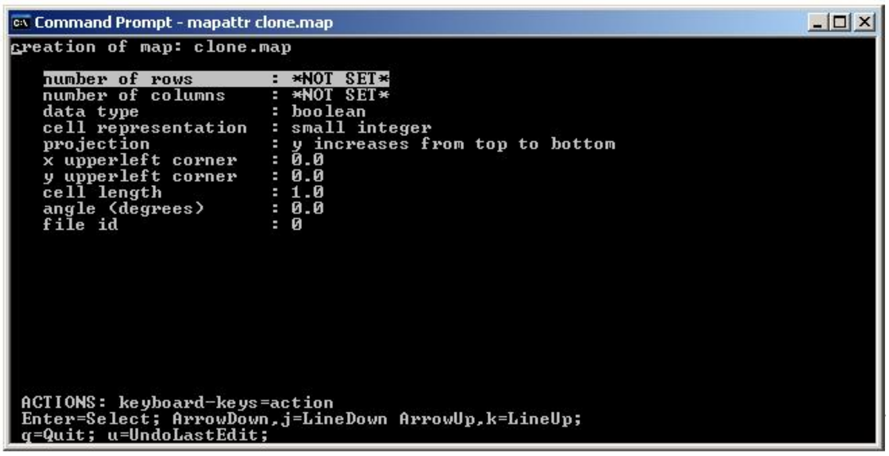
<u>Step 1: Creating the clone map with mapattr</u> (to define the location and the extent of your raster map)

> Go to: …*\examples\createdem.*

> Go to the right path in **Command Prompt** and type:

mapattr clone.map   <enter>

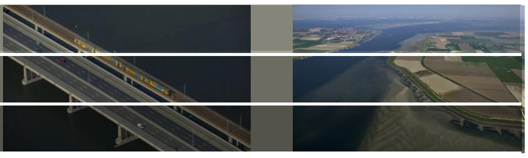
The following screen will appear:



```
creation of map: clone.map

   number of rows         : *NOT SET*
   number of columns      : *NOT SET*
   data type              : boolean
   cell representation    : small integer
   projection             : y increases from top to bottom
   x upperleft corner     : 0.0
   y upperleft corner     : 0.0
   cell length            : 1.0
   angle (degrees)        : 0.0
   file id                : 0



 ACTIONS: keyboard-keys=action
 Enter=Select; ArrowDown,j=LineDown ArrowUp,k=LineUp;
 q=Quit; u=UndoLastEdit;
```

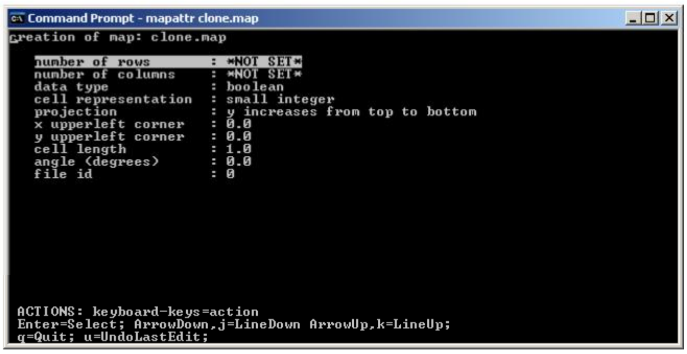
Deltares

Map files – Data conversion



Data needed for clone map:

- The number of rows and columns of our grid
- The x,y location of the upperleft corner
- The cellsize (cell length)



```
creation of map: clone.map

   number of rows         : *NOT SET*
   number of columns      : *NOT SET*
   data type              : boolean
   cell representation    : small integer
   projection             : y increases from top to bottom
   x upperleft corner     : 0.0
   y upperleft corner     : 0.0
   cell length            : 1.0
   angle (degrees)        : 0.0
   file id                : 0



 ACTIONS: keyboard-keys=action
 Enter=Select; ArrowDown,j=LineDown ArrowUp,k=LineUp;
 q=Quit; u=UndoLastEdit;
```

=> **Look in ASCII file and fill in dos screen with the (2)Clone map file data:**

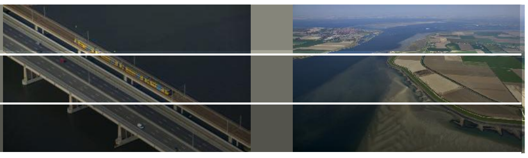
(**1) Ascii file data (header):** **(2) Clone map file data:**

```
NCOLS 80
NROWS 100
XLLCORNER 182140.000000
YLLCORNER 326880.000000
CELLSIZE 10.000000
NODATA_VALUE        1e31
```

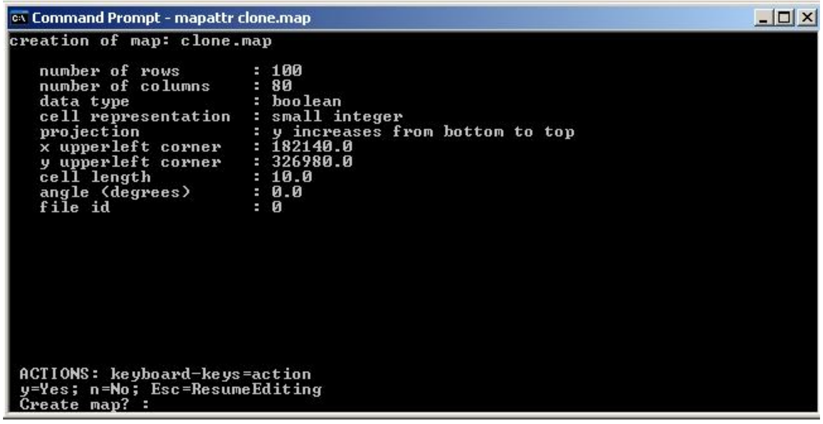
Number of rows = 100
Number of columns = 80
X ulc = 182140
Y ulc = 326880 + **100** = 326980
Cell length = 10

Note: The corner in the ASCII file is **lower** left, while mapattr asks for the **upper** left

Map files – Data conversion



The result of filling is now as below:



```
creation of map: clone.map

   number of rows         : 100
   number of columns      : 80
   data type              : boolean
   cell representation    : small integer
   projection             : y increases from bottom to top
   x upperleft corner     : 182140.0
   y upperleft corner     : 326980.0
   cell length            : 10.0
   angle (degrees)        : 0.0
   file id                : 0



 ACTIONS: keyboard-keys=action
 y=Yes; n=No; Esc=ResumeEditing
 Create map? :
```

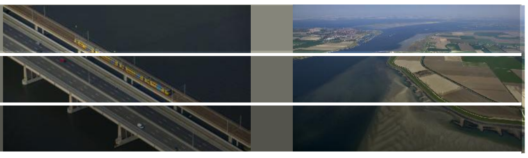
Now type in dos screen:

q *<enter>*
y *<enter>*

clone.map should now be in the example directory …*examples**createdem*

Note: The projection should be the same for all maps (like in …\examples\runoff). Otherwise e.g if your rivers (river.map) are not on the right location of your surface level map and you can get weird flooding results)

Map files – Data conversion



Check whether clone.map is right

> Use mapattr for also review your map attributes:

mapattr –p clone.map   <enter>

Not right clone map?

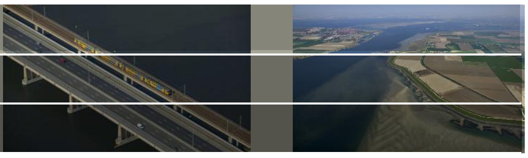
> *Use the edit function*

mapattr -e clone.map   <enter>

For more **options** please see help documentation:
http://pcraster.geo.uu.nl/documentation/pcrman/r16156.htm
(can also be invoked via the PCRaster tab under the Windows start menu)

Now we are ready to convert ASCII files to map files!

Map files – Data conversion



Step 2: Creating the dem.map

Procedure:

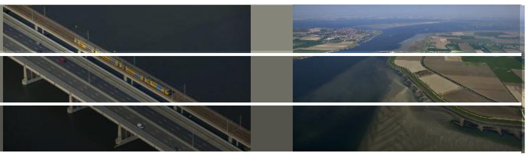
> Go to: …\examples\createdem.

> Go to the right path in **Command Prompt** and type:

asc2map --clone clone.map –a –S dem.asc dem.map   <enter>

Explanation terms in asc2map commando
--*clone:* define the clone map we have made in step 1
-*a* is a **option** to define that your ascii file is a ESRI ASCII file
-*S* is a **option** to define your **data type** is **scalar**. Other types can be chosen with:
-*B* for boolean data
-*N* for nominal data

For more options please see help documentation:
http://pcraster.geo.uu.nl/documentation/pcrman/r14841.htm
(can also be invoked via the PCRaster tab under the Windows start menu)

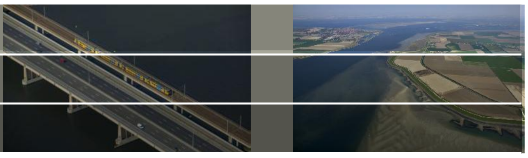


Map files – Exercise 1

Exercise 1: Making all the base map files needed for the the Rajang Delta subsidence model

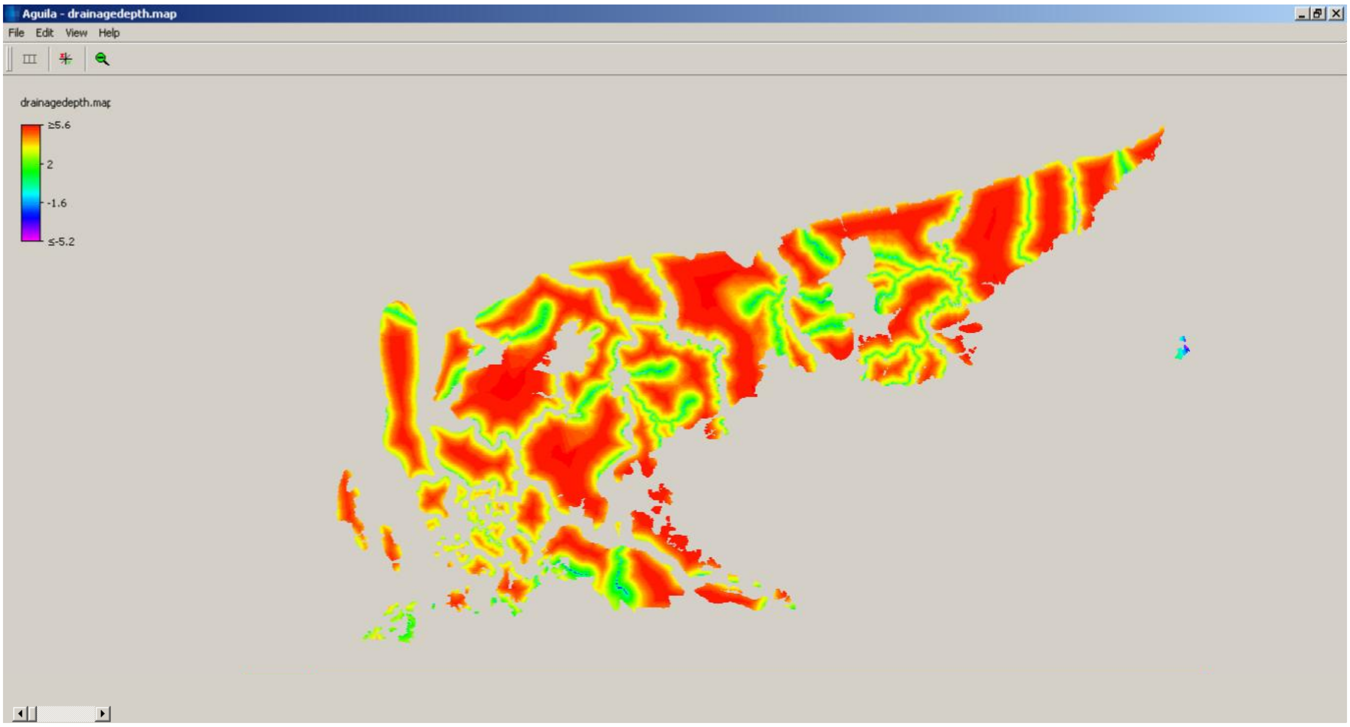
- Go to: ...\Exercises\1 CreateMaps
- Create a clone map *CloneScalar.map* with data type **scalar** using the program **mapattr** (look at example from sheet 17). Use *y increases from bottom 2 top at projection*
- Create the following maps using asc2map, giving names and data types as given in the table below
- Run *SarawakDrain.bat* in the **Command Prompt**. See if the model runs. Press enter, *DrainageDepth.map* should appear

ASCII	Map name	Data type
peatextent.asc	PeatExtent.map	boolean
mainrivers.asc	pointer_mainriv_kal.map	
allrivers.asc	pointer_riv_kal.map	
sea.asc	pointer_sea_kal.map	



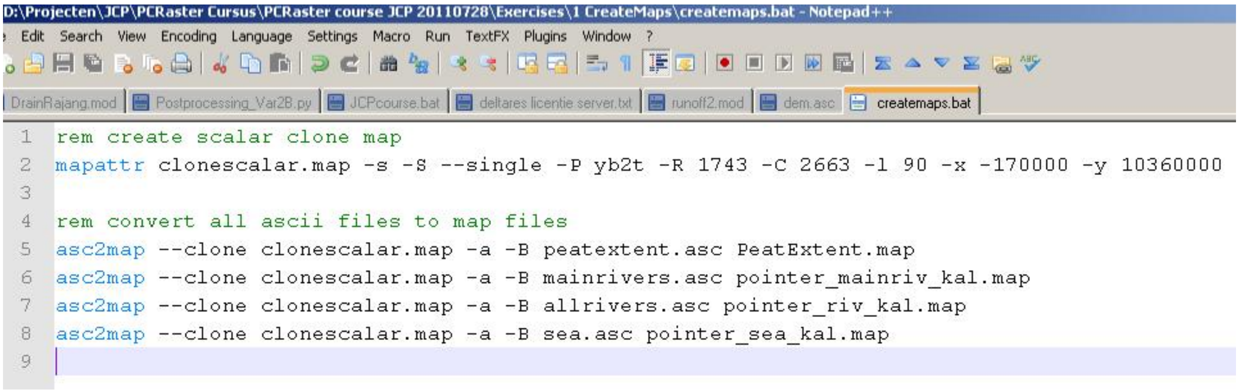
Map files – Exercise 1

The resulting maps should look as follows (click to switch between maps):



Map files – Exercise 1

Tip: Use a batch file when converting muliple maps


D:\Projecten\JCP\PCRaster Cursus\PCRaster course JCP 20110728\Exercises\1 CreateMaps\createmaps.bat - Notepad++
Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
DrainRajang.mod
Postprocessing_Var2B.py
JCPcourse.bat
deltares licentie server.txt
runoff2.mod
dem.asc
createmaps.bat

```
rem create scalar clone map
mapattr clonescalar.map -s -S --single -P yb2t -R 1743 -C 2663 -l 90 -x -170000 -y 10360000

rem convert all ascii files to map files
asc2map --clone clonescalar.map -a -B peatextent.asc PeatExtent.map
asc2map --clone clonescalar.map -a -B mainrivers.asc pointer_mainriv_kal.map
asc2map --clone clonescalar.map -a -B allrivers.asc pointer_riv_kal.map
asc2map --clone clonescalar.map -a -B sea.asc pointer_sea_kal.map

```

The batch-file can be run within the PCRaster command prompt

For more info please see help documentation:

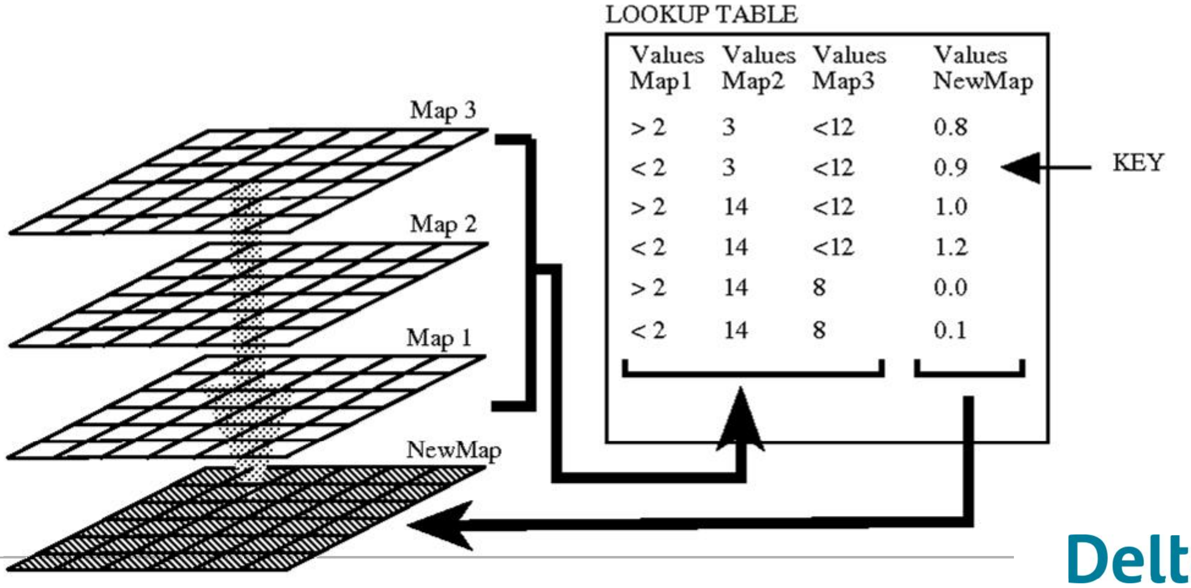
http://pcraster.geo.uu.nl/documentation/pcrman/r16156.htm

(can also be invoked via the PCRaster tab under the Windows start menu)

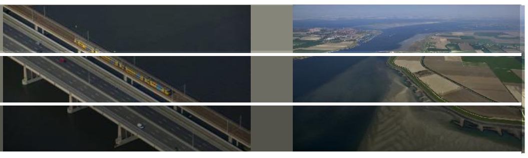
Lookup tables - introduction

Tables are used to specify relations between PCRaster maps. The combination of values for each cell is used to define a new value.

In example below **three maps** are used as input. The table at the right determines how a **new map** is generated based on the combination of values of the three input maps



Values Map1	Values Map2	Values Map3	Values NewMap
> 2	3	<12	0.8
< 2	3	<12	0.9
> 2	14	<12	1.0
< 2	14	<12	1.2
> 2	14	8	0.0
< 2	14	8	0.1



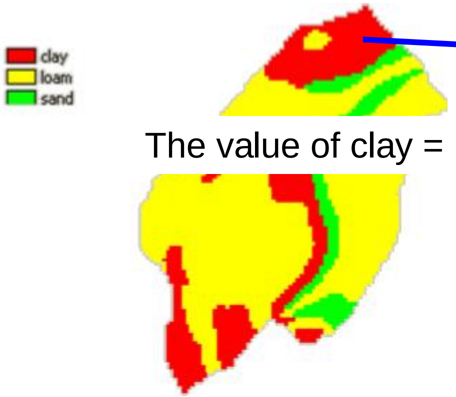
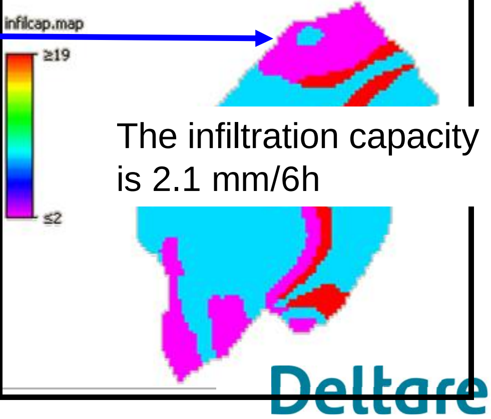
Lookup tables - example

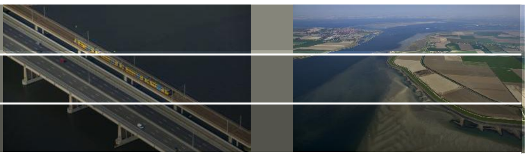
Why using lookup tables?

For now we only work with one or two input maps.

Example: the runoff soil types (left) are matched to the lookup table with infiltration capacity (middle) to generate an infiltration map (right) (**note** Clay = 1, Loam = 2, Peat = 3)

Soil type	Lookup Table	Infiltration capacity


The value of clay = 1
	1 2.1
	2 8.3
	3 19.0
		
		The infiltration capacity is 2.1 mm/6h



Lookup tables - syntax

Type of data in lookup tables: value/ranges

- **Values** are typically used for **boolean** or **nominal** type data. **Ranges** are used for **scalar** type data.

Example: Values are defined in PCRaster table file in the following format

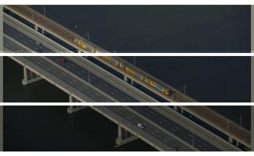
1	2.1	-> if input is 1, output is 2.1
2	8.3	-> if input is 2, output is 8.3
3	19.0	-> if input is 3, output is 19.0

Ranges are defined in the PCRaster table file in the following format:

<,0]	0	-> if input is 0 or lower, output is 0
<0,0.5]	2.17	-> if input is between 0 and 0.5, output is 2.17
<0.5,2]	3.35	-> if input is between 0.5 and 2, output is 3.35
<2,>	5.30	-> if input is 2 or higher, output is 5.30

Note, if you use two maps, a matrix lookup table may be easier to use. For the syntax please see the manual: http://pcraster.geo.uu.nl/documentation/pcrman/x521.htm

Lookup tables – Exercise 2



Exercise 2: Make your own lookup table for creating a waterlevel.map

Two maps are in …\exercises\2 relatemaps :

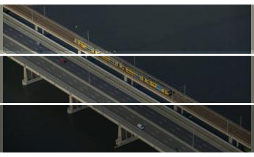
- riverclass.map:Indicating the river class
- dariver.map: A map with the distance to the rivers in meter
- Data to use for making the lookup table:In this table the maximum water level is given in **blue** depending on the river **distance** along the river and the **river class** (these table data links the two maps)

Distance River class	< 10000 meters	10000 - 50000 meters	> 50000 meters
12	1.20	1.50	2.50
14	0.80	1.10	3.50

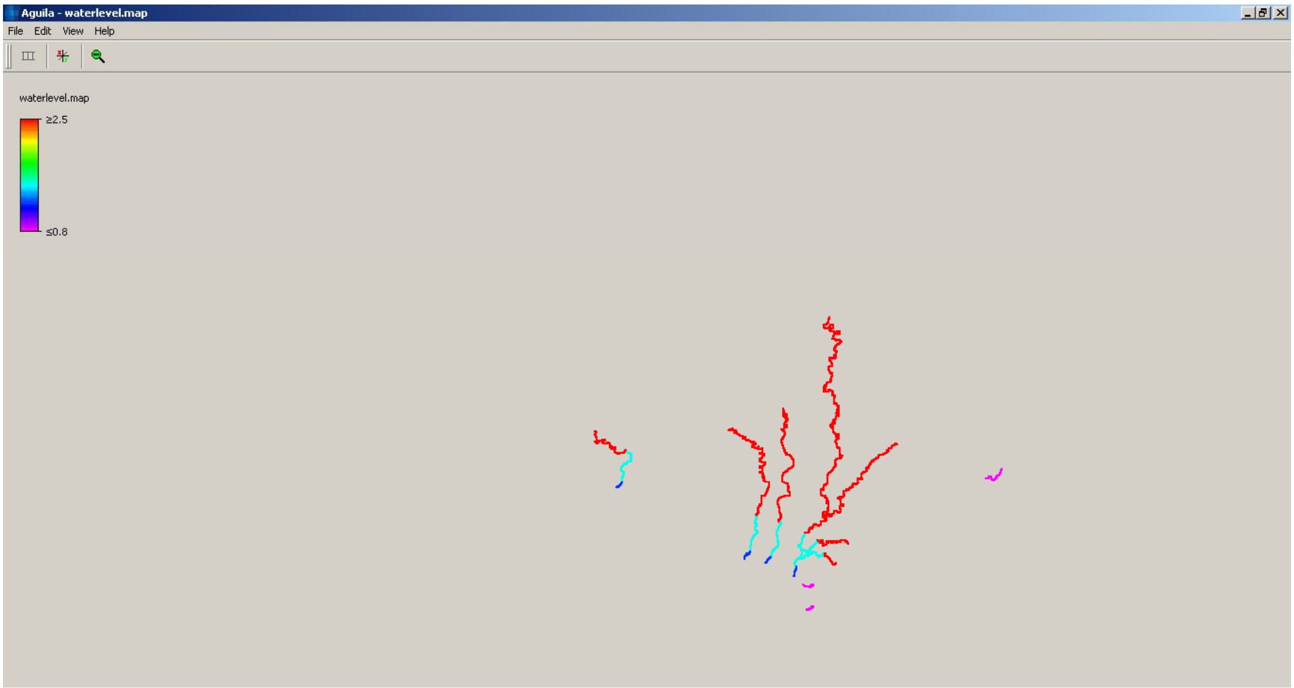
Procedure to create water level map:

1. Create a file with the name **WLmax.tbl.**
2. Write in this file three columns wich relates the **distance** and **river class** to the **drainage depth.** Use the data from the table, see previous sheet for right format.
3. Run the relatemaps.bat by typing in the PCRaster command prompt (in this file the lookup function is used): *Relatemaps.bat* *<enter>*
4. Result after running batch=> WaterLevel.map

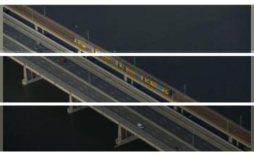
Lookup tables – Exercise 2



The resulting map should look as follows



Note: the values are purely fictive and only for this exercise!!!



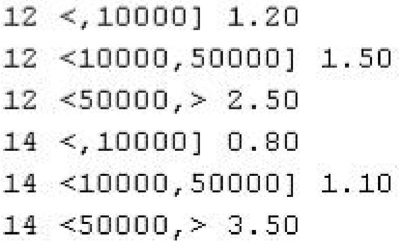
Lookup tables – Exercise 2

Tip: Use Matrix table instead of Column table!

For this exercise we reduced the amount of distance and river classes. Originally 35 river classes where used and over 100 distance ranges. **This would require > 3500 rows if a column table is used!!!**

In this case a matrix table is far more useful!

Column Table

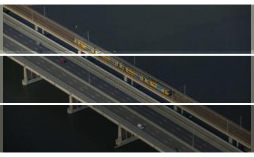


```
12 <,10000] 1.20
12 <10000,50000] 1.50
12 <50000,> 2.50
14 <,10000] 0.80
14 <10000,50000] 1.10
14 <50000,> 3.50
```

Matrix Table

```
-99 12  14
<,10000] 1.20 0.80
<10000,50000] 1.50 1.10
<50000,> 2.50 3.50
```

Note: Open Relatemaps.bat and look at the commando given. See also the manual http://pcraster.geo.uu.nl/documentation/pcrman/r8621.htm. When you use a matrix table, you should use the option **--matrixtable**!!!



Time series - introduction

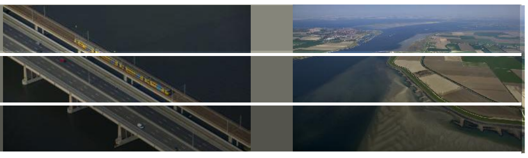
Dynamic modelling with time series

Time series are used as **input** or **output** for **dynamic modeling** (see slide 34).

They are not spatial like maps, but relate to a location, for instance a meteo station or a catchment.

Variables which are typically available in time series are:

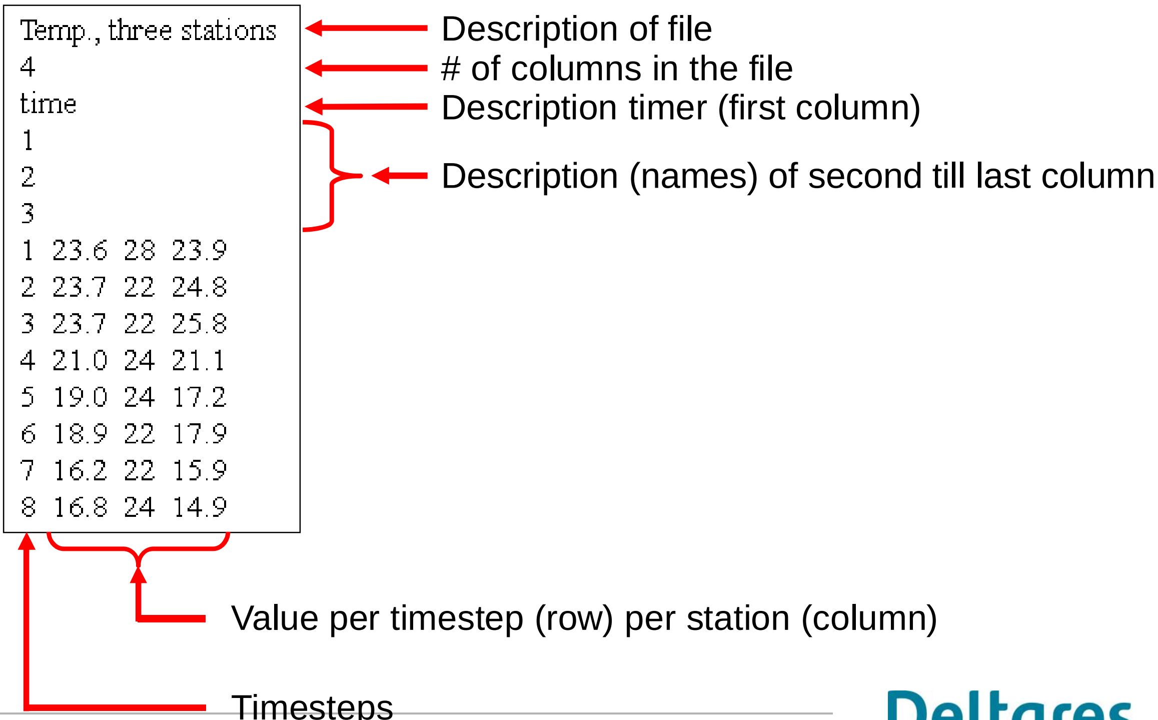
- Temperature at a meteorological station
- Precipitation at a meteorological station
- Discharge at a hydraulic station



Time series - Format

Format time serie files

Time series are ASCII files with the extension .tss with the following format:



```
Temp., three stations
4
time
1
2
3
1  23.6  28  23.9
2  23.7  22  24.8
3  23.7  22  25.8
4  21.0  24  21.1
5  19.0  24  17.2
6  18.9  22  17.9
7  16.2  22  15.9
8  16.8  24  14.9
```

Description of file

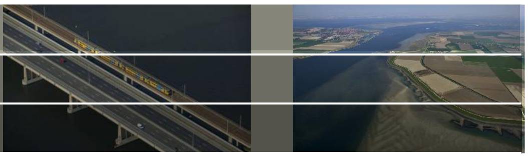
# of columns in the file

Description timer (first column)

Description (names) of second till last column

Value per timestep (row) per station (column)

Timesteps



Time series - Exercise 3

Excercise 3: Make precipitation time serie files in PCRaster format

In this exercise you will make average monthly and annual precipitation based on the precipitation table at the right side of this slide:

1. Make a time series file based on the table right according to the format explained in the previous slide.
2. Save this file in …\exercises\3 Rainfall under the name **rain.tss**
3. Run the file letitrain.bat by typing in command prompt:

Letitrain.bat *<enter>*

=>Result: twelve maps created, **rain0000.001**, **rain0000.002**, ….., **rain0000.0012** with monthly rainfall.

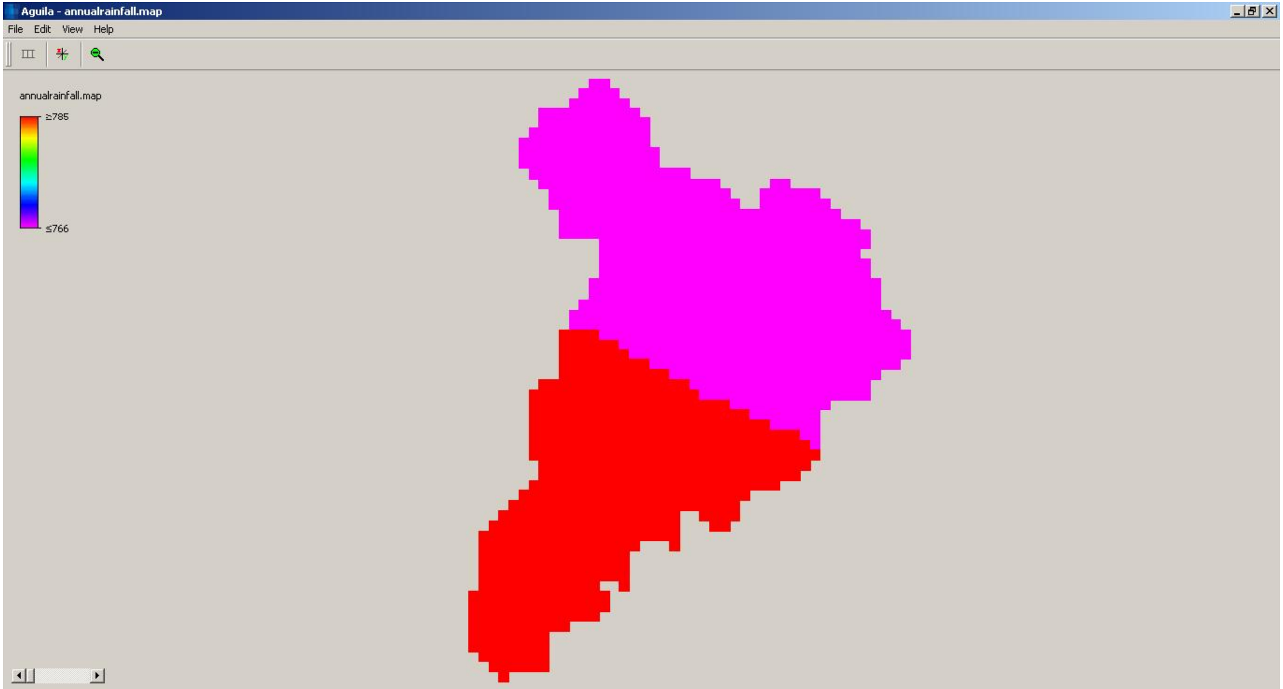
AnualRainfall.map should give the annual rainfall distributed over the area.

Table - Precipitation (mm)

Station / Month	Station 1 (ID =1)	Station 2 (ID =2)
January	74	75
February	71	71
March	74	76
April	58	59
May	40	40
June	40	42
July	45	47
August	62	64
September	80	78
October	80	85
November	75	79
December	67	69
Total	766	785

Time series - Exercise 3

Annualrainfall.map should look like:



Note: More info on time series can be found in the manual:http://pcraster.geo.uu.nl/documentation/pcrman/x720.htm

Functions - introduction

Explanation of using static functions (at a certain time instant):

Functions should be operated as follows in Command Prompt:

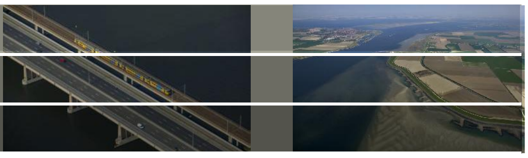
pcrcalc result=expression <enter>

pcrcalc is the program you use to execute expressions. It is similar to aguila (for displaying maps), asc2map (for file conversion) and mapattr (for editting and viewing map attrigutes)

result is the result of the expression. In static models this is a map file

expression is the function from which the result is derived. An expression is compiled from **operators, maps** and/or **constants**

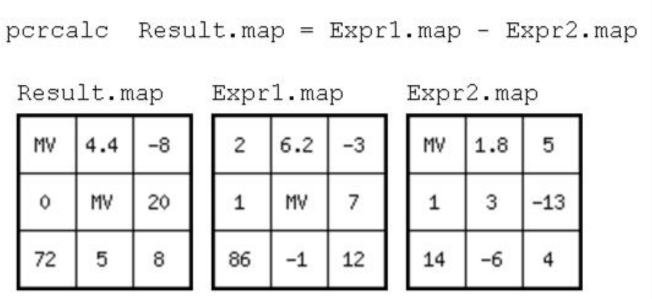
Functions - simple operations



Two examples of using a the operation function

Input data: dem.map from the runoff example
Example 1: Correct the DEM for 0.2 m subsidence using a **constant**:
Type in PCRaster command prompt:
pcrcalc demsubside.map=dem.map-0.2
Example 2: Calculate the drainage depth based on a groudwater level **map:**
Type in PCRaster command prompt:
pcrcalc draindepth.map=dem.map-gwlevel.map

Explanation operation function by maps:



```
pcrcalc  Result.map = Expr1.map - Expr2.map
```

Result.map

MV	4.4	-8
0	MV	20
72	5	8

Expr1.map

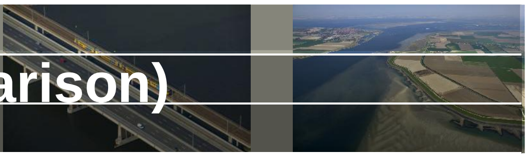
2	6.2	-3
1	MV	7
86	-1	12

Expr2.map

MV	1.8	5
1	3	-13
14	-6	4

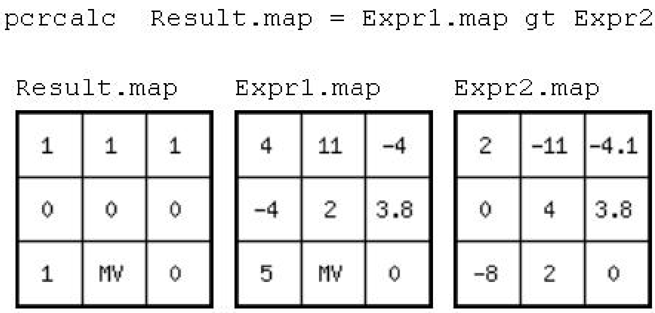
You can use all algebraic operators, such as +, - , / , *

Functions - conditions (comparison)



Step 1: making a conditional map

Example of using the conditional statement: “greather than” (=“gt” in PCRaster)



```
pcrcalc  Result.map = Expr1.map gt Expr2.map
```

Result.map

1	1	1
0	0	0
1	MV	0

Expr1.map

4	11	-4
-4	2	3.8
5	MV	0

Expr2.map

2	-11	-4.1
0	4	3.8
-8	2	0

When Expr1.map is greather than Expr2.map => Result.map = 1

Else, result.map = 0. (Result.map has a **boolean** data type)

Missing value in Expr1.map and/or Expr2.map gives a missing value in Result.map

Notes:

Other conditional statements in PCRaster are: “lt” (less than), “ge” (greather or equal to), “le” (les or equal to), “eq” (equal to) and “ne” (not)

Functions - conditions (if statement)

Step 2: using conditional map in a if statement

```
pcrcalc  Result.map = if(Cond.map then Expr1.map)
```

Result.map

2	0	MV
4	MV	-6
MV	MV	3

Cond.map

1	1	0
1	0	1
0	MV	1

Expr1.map

2	0	3
4	MV	-6
-7	1	3

This statement reads, if the Cond.map = 1, then result.map = Expr1.map.

Cond.map should have a **boolean** data type

- The operator if should always be accompanied with brackets ()

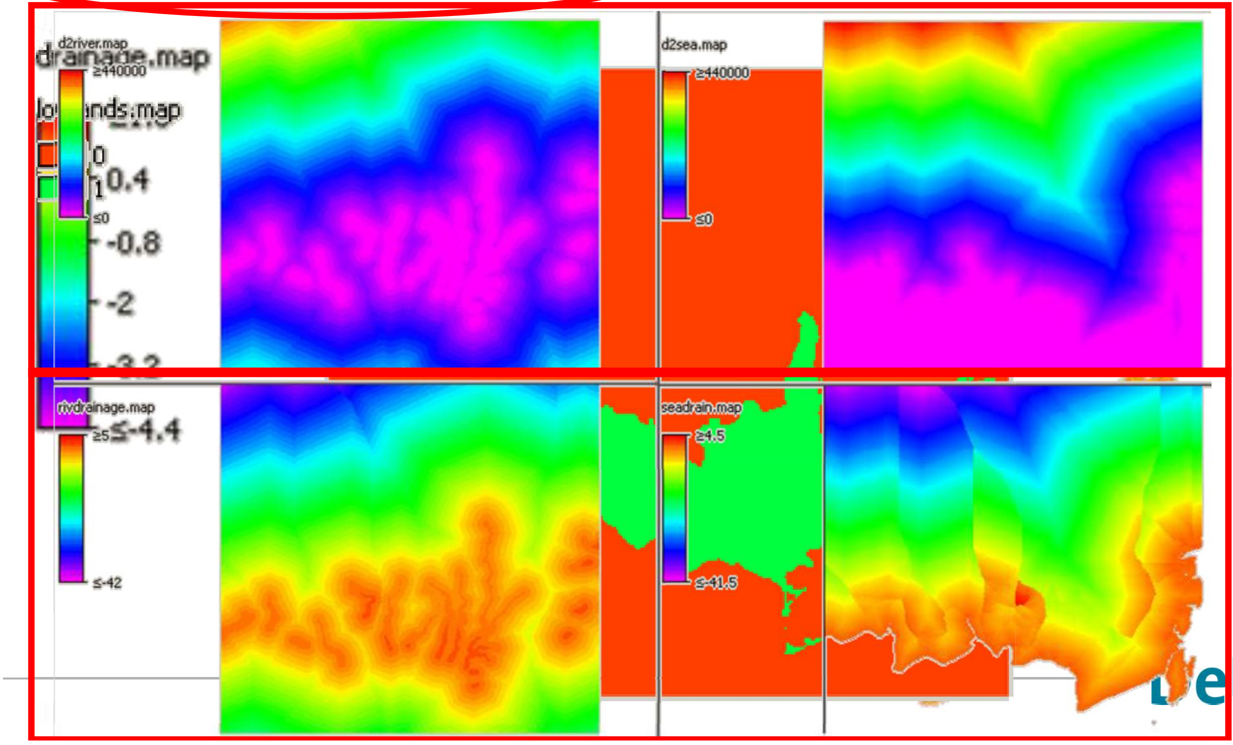
- “then” can be replaced with a comma “,”

- Cond.map can be derived from the expression at the previous slide. It is also possible to replace cond.map with this expression. Be sure to use brackets

Functions - conditions

An example:

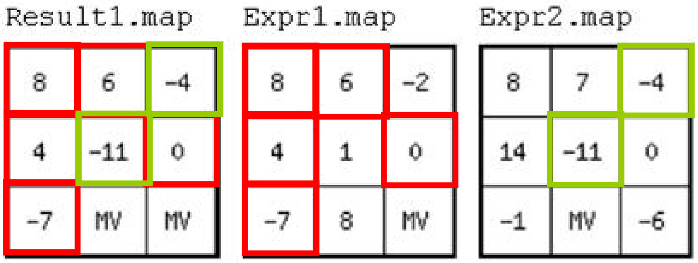
pcrcalc *drainage.map=****if****(lowlands.map* ***eq*** *1* ***then if****(d2river.map* ***le*** *d2sea.map* ***then*** *rivdrainage.map* ***else*** *seadrain.map))*



Functions - minimum and maximum

To determine the lowest cell values of multiple maps, you can write the following function:

```
pcrcalc  Result1.map =  min(Expr1.map,Expr2.map)
```



Result1.map

8	6	-4
4	-11	0
-7	MV	MV

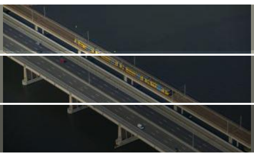
Expr1.map

8	6	-2
4	1	0
-7	8	MV

Expr2.map

8	7	-4
14	-11	0
-1	MV	-6

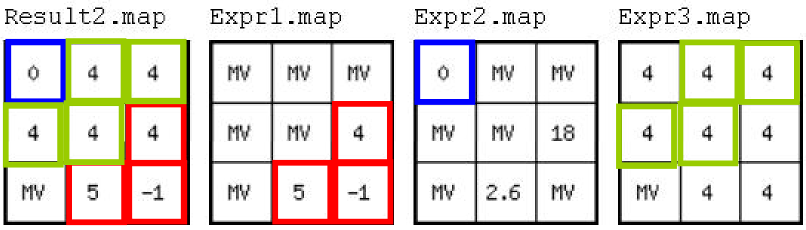
The cell values of Result1.map are the lowest values of Expr1.map and Expr2.map. If one of the two input maps contains missing values (MV), the resulting cell will also contain a missing value.



Functions - cover

Cover can be used to merge multiple maps into one new map:

```
pcrcalc  Result2.map = cover(Expr1.map,Expr2.map,Expr3.map)
```



Result2.map

0	4	4
4	4	4
MV	5	-1

Expr1.map

MV	MV	MV
MV	MV	4
MV	5	-1

Expr2.map

0	MV	MV
MV	MV	18
MV	2.6	MV

Expr3.map

4	4	4
4	4	4
MV	4	4

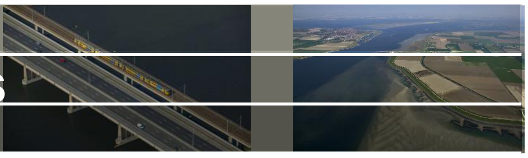
Result2.map will if exist first be filled with the values from Expr1.map.

The remaining cells will be filled with values from Expr2.map (if exist)

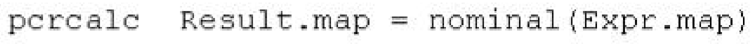
The remaining cells will be filled with values from Expr3.map (if exist)

Cells with no values in any of the maps will remain missing value (MV)

Functions - change data types

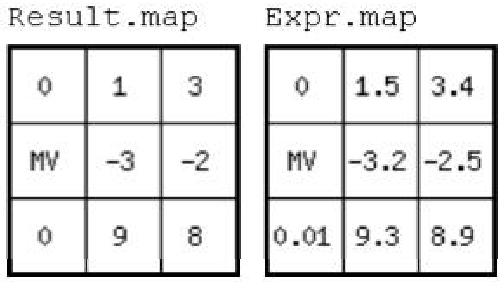


Transferring data types



```
pcrcalc  Result.map = nominal(Expr.map)
```

a, you can use this expression:



Result.map

0	1	3
MV	-3	-2
0	9	8

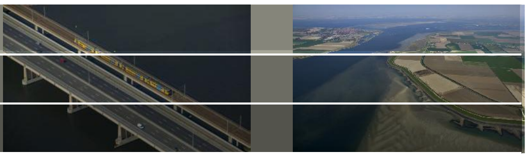
Expr.map

0	1.5	3.4
MV	-3.2	-2.5
0.01	9.3	8.9

Result.map are the integers of the scalar type Expr.map. The data type from expression is scalar (in this example) and **nominal** of result.map

With boolean(Expr.map) and scalar(Expr.map) data can be transfered to these data types.

Functions - spread functions



Explanation use spread function

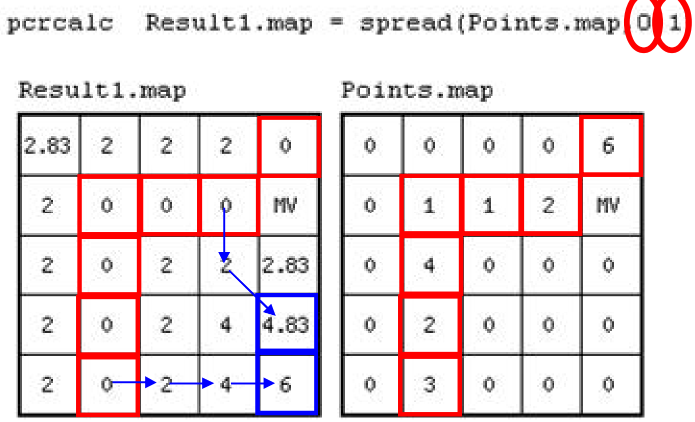
An often used function is the spread function:

pcrcalc Result.map = spread(**Points**, **InitialFrictionDistance**, **Friction**)

- **Points** are boolean, nominal or ordinal maps. From this map the shortest distance to every cell > 0 is calculated
- **Initial Friction Distance** is the initial friction distance at the point locations. This can be supplied with a map or with a value representative for the whole map.
- **Friction** the increase of friction travelling from one cell center to its neighbor

Functions - spread functions

Example spread function with a map (cell size = 2):



```
pcrcalc  Result1.map = spread(Points.map,0,1)
```

Result1.map

2.83	2	2	2	0
2	0	0	0	MV
2	0	2	2	2.83
2	0	2	4	4.83
2	0	2	4	6

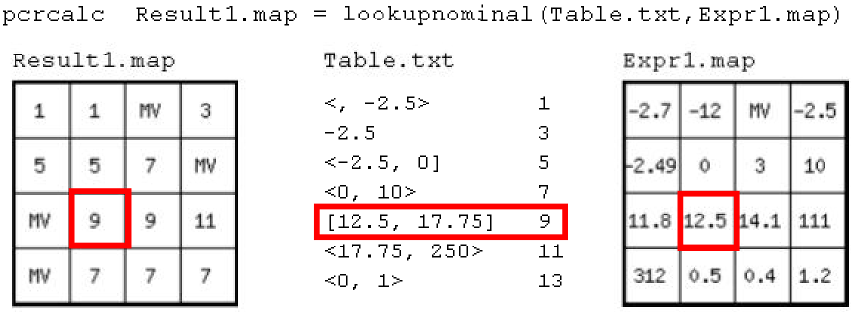
Points.map

0	0	0	0	6
0	1	1	2	MV
0	4	0	0	0
0	2	0	0	0
0	3	0	0	0

- The points.map determines to which cells the shortest distance path is calculated
- To these cells the initial friction distance is 0
- The friction traveling from one cell to the other is 1
- The friction to the blue cell equals 6. The friction is calculated as *cell size * friction* for horizontal and vertical adjacent cells. The friction path is three cells long, cell size = 2 and friction = 1. The total friction yields 2 * 1 + 2 * 1 + 2 * 1 = 6
- The friction to the blue cell equals 4.83 The friction is calculated as *cell size * friction* for horizontal and vertical adjacent cells and 2Pi * *cell size * friction* for diagonal adjacent cells. The cell size = 2 and friction = 1. The total friction yields 2 * 1 + 2.83 * 1 = 4.83

Functions - lookup

Tables can be used to generate new maps with the help of the **lookup** function. The lookup function comes many flavors depending on the requested result data type. Here we will use **lookupnominal.**



```
pcrcalc  Result1.map = lookupnominal(Table.txt,Expr1.map)
```

Result1.map

1	1	MV	3
5	5	7	MV
MV	9	9	11
MV	7	7	7

Table.txt

```
<, -2.5>          1
-2.5              3
<-2.5, 0]         5
<0, 10>           7
[12.5, 17.75]     9
<17.75, 250>      11
<0, 1>            13
```

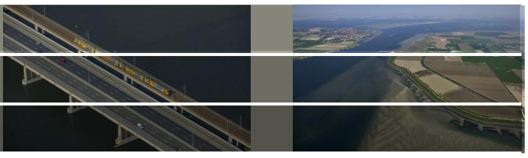
Expr1.map

-2.7	-12	MV	-2.5
-2.49	0	3	10
11.8	12.5	14.1	111
312	0.5	0.4	1.2

The resulting value 9 is related to 12.5, since it is in the range 12.5 - 17.75.

Lookupscalar and lookupboolean will result in outputmaps with the scalar and boolean data type.

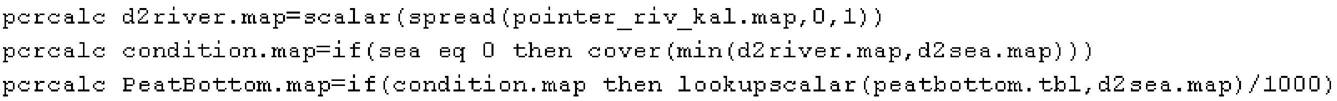
Functions - scripting



A strength of PCRaster is the ability to use scripts. There are two ways of using scripts:

1. DOS batch file
2. PCRaster mod file

A DOS batch file. In this file multiple PCRaster commandos can be written, which will be executed subsequently.

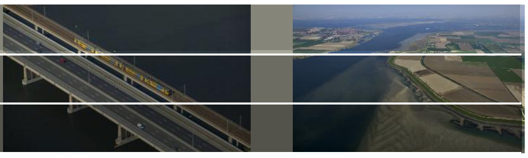


```
pcrcalc d2river.map=scalar(spread(pointer_riv_kal.map,0,1))
pcrcalc condition.map=if(sea eq 0 then cover(min(d2river.map,d2sea.map)))
pcrcalc PeatBottom.map=if(condition.map then lookupscalar(peatbottom.tbl,d2sea.map)/1000)
```

In this script the result of line 1, d2river.map is used in the expression in line 2. The result of line 2 is used in the expression of line 3.

Note: In a batch file, PCRaster commando's can be combined with DOS commandos, which is a clear advantage.

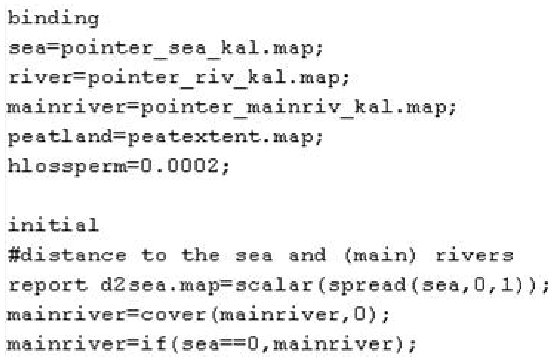
Functions - scripting



A strength of PCRaster is the ability to use scripts. There are two ways of using scripts:

1. DOS batch file
2. PCRaster mod file

A PCRaster mod-file. This file is mostly used for dynamic modeling. However, you can also use it for giving subsequent PCRaster commandos:

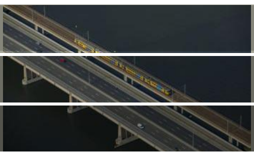


```
binding
sea=pointer_sea_kal.map;
river=pointer_riv_kal.map;
mainriver=pointer_mainriv_kal.map;
peatland=peatextent.map;
hlossperm=0.0002;

initial
#distance to the sea and (main) rivers
report d2sea.map=scalar(spread(sea,0,1));
mainriver=cover(mainriver,0);
mainriver=if(sea==0,mainriver);
```

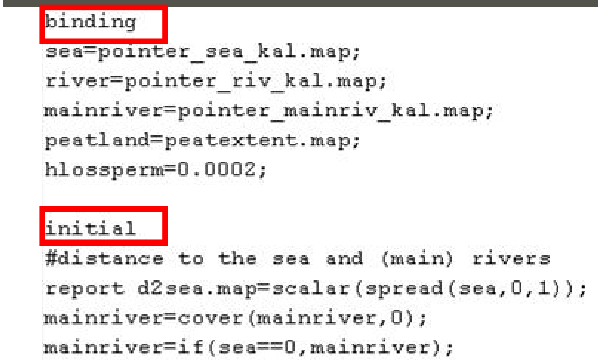
The script should have a .mod extension. The script can be run via command prompt:

*pcrcalc **-f** scriptname.mod <enter>*

Be aware! Without **-f** the model will not run!!



Functions - scripting



```
binding
sea=pointer_sea_kal.map;
river=pointer_riv_kal.map;
mainriver=pointer_mainriv_kal.map;
peatland=peatextent.map;
hlossperm=0.0002;

initial
#distance to the sea and (main) rivers
report d2sea.map=scalar(spread(sea,0,1));
mainriver=cover(mainriver,0);
mainriver=if(sea==0,mainriver);
```

A mod-script requires at least a **binding** section and a **initial** section:

- In the **binding** section, maps can be called as script variables. If maps are not called in the binding section, the maps need to be directly adressed in the initial section.

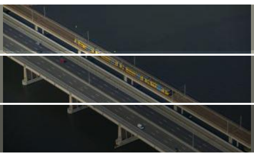
- In the **initial** section, scripts can be written similar as explained before. However, if variables are called in the binding section, you only have to refer to the variable.

Notes:

- mod files can be used when you want to build models with a lot of expressions.

- The syntax requires a "**;**" at the end of every line

- If you want to write output-files to the same directory as the mod-file, you should should put "report" before the function (see script above)



Functions - Exercise

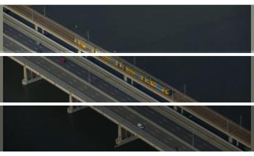
In this exercise you will complete the PCRaster Sarawak drainage model. This version of the model calculates current drainability and the drainability in 30 years. All the input files are generated. However, you have to complete the script using the functions discussed before you can run it.

1. Open the file exercises\4 DrainSarawak\DrainRajang.mod

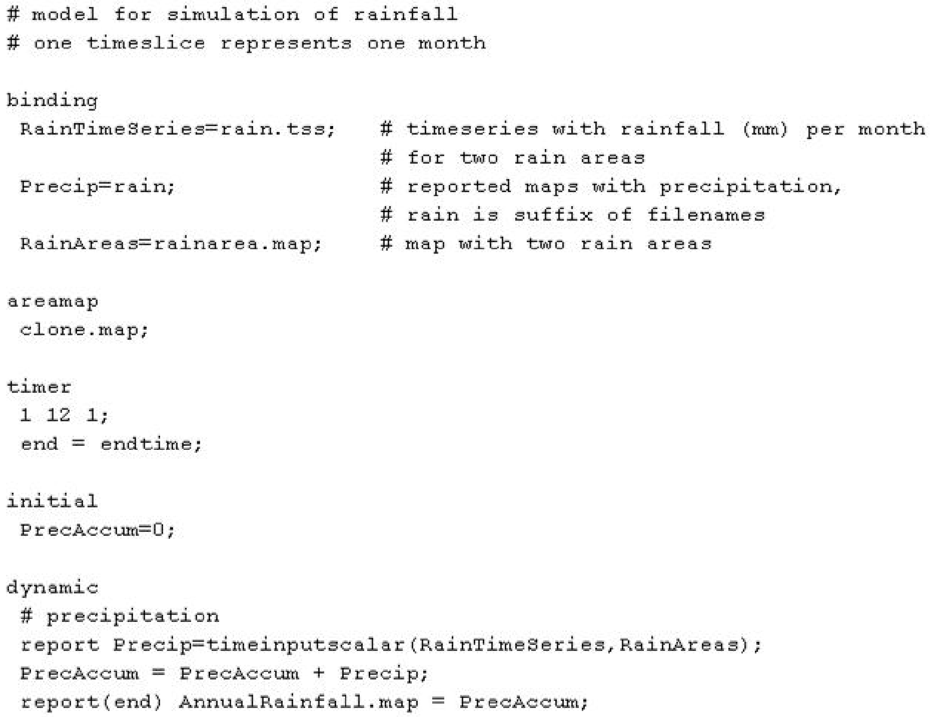
2. In **line 21** we need to make a spread function to calculate the distance to rivers. This function looks very similar to line 18, where the distances are calculated to the mainrivers.

3. In **line 34** we need to make a peat bottom map, indicating the elevation of the mineral layer. We do this with one complex function, similar to the one used in line 33 for calculating the elevation of the surface.

Dynamic models



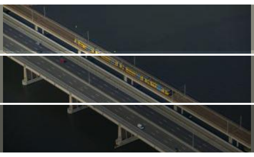
Dynamic models can be used for calculating evolutions in time. Herefore you should supply PCRaster with a mod-file. See for example rain.mod in the rainfall exercise:

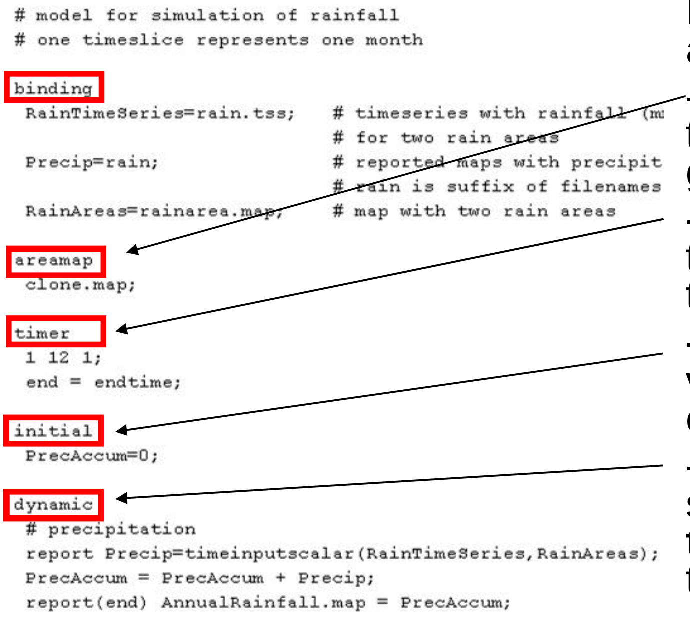


```
# model for simulation of rainfall
# one timeslice represents one month

binding
 RainTimeSeries=rain.tss;    # timeseries with rainfall (mm) per month
                             # for two rain areas
 Precip=rain;                # reported maps with precipitation,
                             # rain is suffix of filenames
 RainAreas=rainarea.map;     # map with two rain areas

areamap
 clone.map;

timer
 1 12 1;
 end = endtime;

initial
 PrecAccum=0;

dynamic
 # precipitation
 report Precip=timeinputscalar(RainTimeSeries,RainAreas);
 PrecAccum = PrecAccum + Precip;
 report(end) AnnualRainfall.map = PrecAccum;
```

Deltares

Dynamic models





```
# model for simulation of rainfall
# one timeslice represents one month

binding
 RainTimeSeries=rain.tss;    # timeseries with rainfall (mm
                             # for two rain areas
 Precip=rain;                # reported maps with precipit
                             # rain is suffix of filenames
 RainAreas=rainarea.map;     # map with two rain areas

areamap
 clone.map;

timer
 1 12 1;
 end = endtime;

initial
 PrecAccum=0;

dynamic
 # precipitation
 report Precip=timeinputscalar(RainTimeSeries,RainAreas);
 PrecAccum = PrecAccum + Precip;
 report(end) AnnualRainfall.map = PrecAccum;
```

Besides the **binding** section, which are allready explained in slide 47, you need:

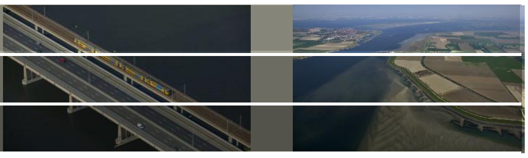
- **areamap** providing one map from which the **map attributes** will be used for the generation of new maps
- a **timer** indicating the start time, end time and time slice. For the format see the rain.mod
- the **initial** section, which gives the initial values of variables before the dynamic calculation comences.
- the **dynamic** section. The script in this section is looped as many times as the **timer** suggests. In this case, this part of the script will be looped 12 times.

Note: the sections should always be placed in the correct order!

For more info, please see the manual: file:///D:/PCRaster/doc/pcrman/ch05.html

Deltares

How you can procede



But with PCRaster you can do much more!!

Please look at the manual background information and much more functions:

file:///D:/PCRaster/doc/pcrman/index.html

In *..\RajangDrainageModel.zip*. There you will find the complete dynamic Rajang drainability model which you can run with:
pcrcalc -f drainrajang.mod

In *..\pcrcourse.zip* you will find the original PCRaster tutorial course.

Deltares