Map Algebra and Static Modelling with PCRaster

Author Derek Karssenberg

Introduction

What is PCRaster?

Integration of

- modelling: tools for constructing and running spatial dynamic environmental models
- GIS: a Geographical Information System

Emphasis is on the first!

Other properties

- spatial domain: 2 dimensional maps, raster based (finite difference)
- available for linux and Microsoft Windows
- free of charge, download at http://www.geog.uu.nl/pcraster

PCRaster: modelling

Modellling (or programming) language pcrcalc

Wide range of models can be built, examples:

- Rhineflow: modelling discharge of the river rhine
- Lisflood: runoff model of large catchments in Europe (ISPRA, Italy)
- Ecological models: plant dispersal
- Sedimentological models: river floodplain evolution

PCRaster: GIS

PCRaster provides

- resampling of maps to other cell size or area
- conversion of data with standard GIS systems (e.g. ESRI products)
- interpolation (with the Gstat program written by Edzer Pebesma)

PCRaster does not provide:

- digitizing
- vector data
- fancy printing facilities

Use a 'standard' GIS in addition to PCRaster!

Entities and syntax of functions

Entities used in models

Map

- main variable in a model, almost everything is a map
- binair format

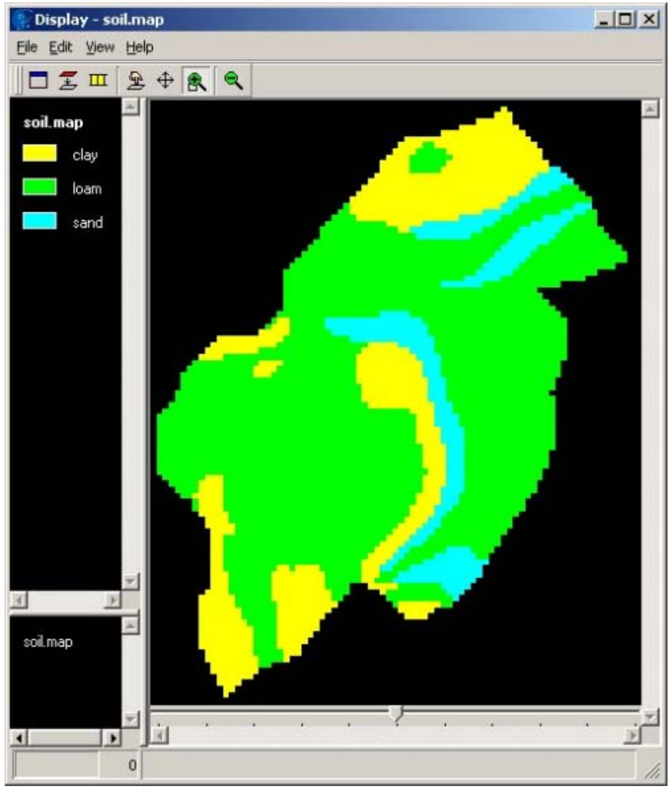
Timeseries

- used in dynamic models
- to represent inputs or storing output variables
- ascii data file

Table

- used in point functions to assign new values as a function of several input maps
- may store statistics of map values
- ascii data file

Entities: map (1)



Entities: map (2)

All maps are raster maps (finite difference)

- constant grid spacing on one map
- different cell sizes are possible, but
- all maps used in a model have the same cell size

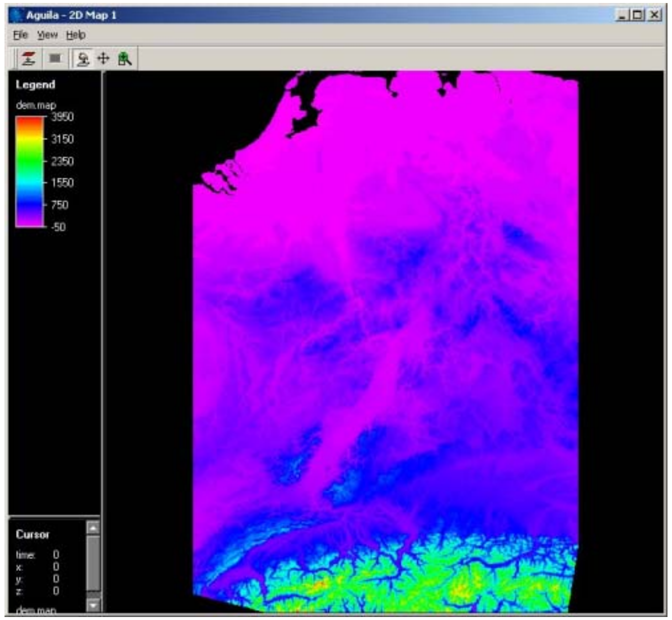
Each map contains one variable (attribute), e.g. pH

Maps are always rectangular

- outside study area: missing values

Entities: map (3)

Visualisation: display (2D display) or aguila (2D and 3D display)



Entities: map (3)

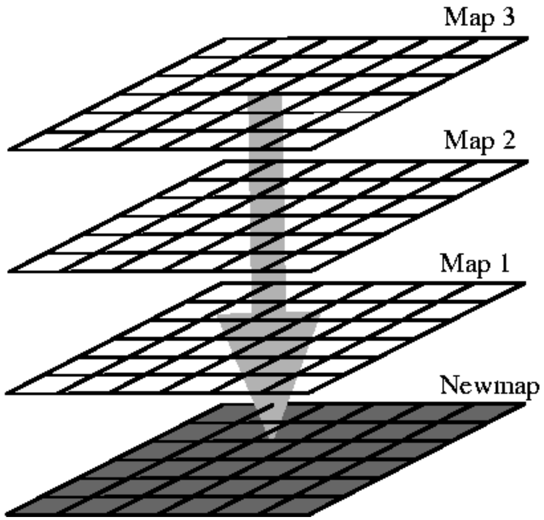
Data types

data type	description attributes	domain	example
boolean	boolean	0 (false), 1 (true)	suitable/unsuitable visible/non visible
nominal	classified, no order	0...255, whole values	soil classes, administrative regions
ordinal	classified, order	0...255, whole values	succession stages, income groups
scalar	continuous, lineair	-10exp(37)...10exp(37), real values	elevation, temperature
directional	continuous, directional	0 to 2 pi (radians), or to 360 (degrees), and -1 (no direction), real values	aspect
ldd	local drain direction to neighbour cell	1...9 (codes of drain directions)	drainage networks, wind directions

Operators and functions

Operators and functions are the building blocks of a model

Derive a single map (sometimes two) from one or more input maps (or tables, timeseries)



Syntax of operators

Result = *expression1* operator *expression2*

operator:

- the name of the operator

expression1, *expression2* are the arguments (i.e. inputs):

- maps
- expressions resulting in a map (i.e., nesting of expressions is possible)

Result is the return value (i.e. what is created):

- one map

Example, multiply two maps (for each cell), arguments are maps:


```
MapA = MapB * MapC
```

Example, multiply maps (for each cell), second arguments is anothe expressions:

```
MapA = MapB * (MapC+MapD)
```

Syntax of functions

Result = function(*expression1*, *expression2*,..,*expressionn*)

function:

- the name of the function

expression1, *expression2*,..,*expressionn* are the arguments (i.e. inputs):

- maps
- expressions resulting in a map

Result is the return value (i.e. what is created):

- one map (sometimes two)

Example, water flow over a local drain direction network (`accuflux` function):

```
RunoffMap = accuflux(LddMap,1000*RainMm)
```

Command line mode

Single statements can be entered at the command line, e.g.:

```
C:\myfiles\pcrcalc mapa.map=mapb.map * mapc.map <Enter>
C:\myfiles\display mapa.map mapc.map
```

Multiplies two maps and displays the output and an input map

Note:

- single maps are stored as files mostly using `.map` as extension of the filename
- in PCRaster scripts (programs) single 'words' are used to refer to map variables, e.g. `MapA`, `Runoff`, `Interception`

Spatial static modelling

Spatial static model: theory

Static model:

z(1..m)=f(*i*(1..n), *p*(1..l))

i(1..n): inputs (maps)

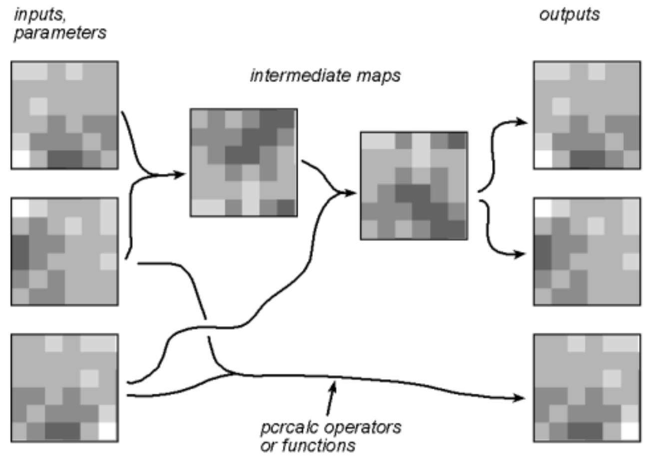
p(1..l): parameters (maps)

z(1..m): outputs (model variables)

f: the static model

Spatial static model: representation in PCRaster (1)

z(1..m)=f(*i*(1..n), *p*(1..l)), where f is represented by a set of pcrcalc functions:



Spatial static model: representation in PCRaster (2)

The function f in:

z(1..m)=f(*i*(1..n), *p*(1..l))

is represented by a set of pcrcalc functions in a *static (or cartographic) modelling* script:

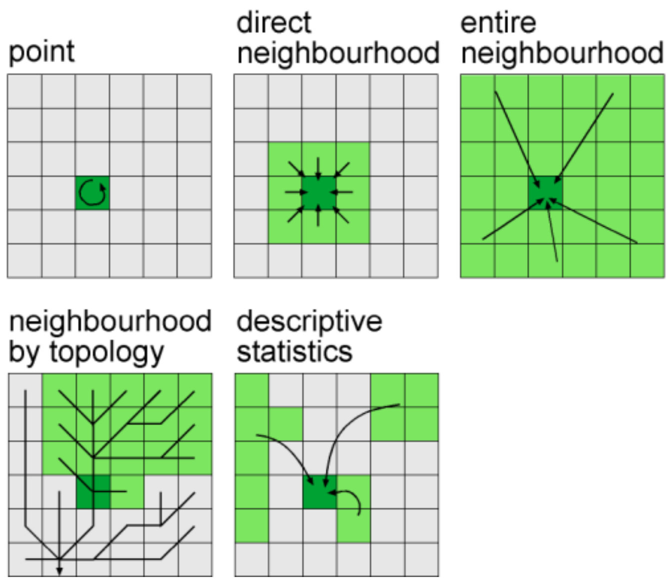
```
pcrcalc statement;
pcrcalc statement;
...
...
```

operations are performed from top to bottom:

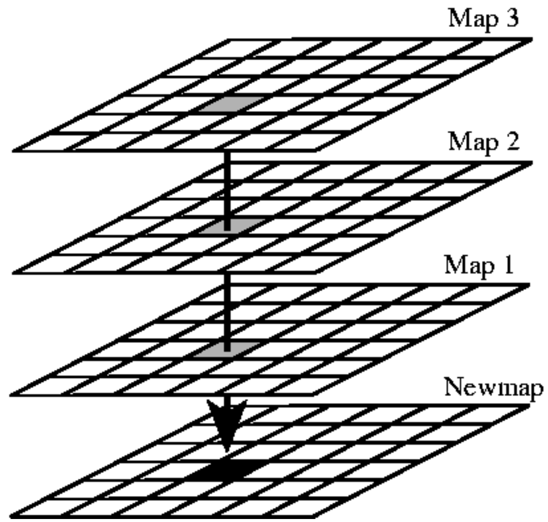
```
# this is a cartographic modelling script file
Friction = lookupscalar(friction.tbl,LandUse);
Friction = 2.5 * Friction;
CostDist = spread(Start,0,Friction);
```

Point functions and operators

Groups of spatial functions and operators



Group of point functions



Speakernotes

Point functions: arithmic, trigonometric, exponential, logarithmic operators

examples of this group

operator	description
*	multiply
+	plus
cos	calculate cosine

sqrt	square root
...	etc

For example:

```
pcrcalc volume.map = area.map * height.map
```

Speakernotes

plus.mod

Point functions: comparison functions

Resulting map is always a Boolean

Comparison holds: result is TRUE

Comparison does not hold: result is FALSE

examples of this group

operator	**description**
eq or ==	equals
ge or >=	greater than or equal
gt or >	greater than
ne or !=	not equal
...	etc

For example:

```
pcrcalc low.map = elevation.map < 12.5
```


Speakernotes

gt.mod

Point functions: conditional statements

Result = if(*condition* then *expression*)

condition:

- Boolean map

expression:

- a map
- a non-spatial (just a number)

condition is TRUE: expression is assigned

condition is FALSE: missing value is assigned

For example:

```
pcrcalc riverw.map = if(river.map,riverwidth.map)
```

Speakernotes

if.mod

Point functions: Boolean operators

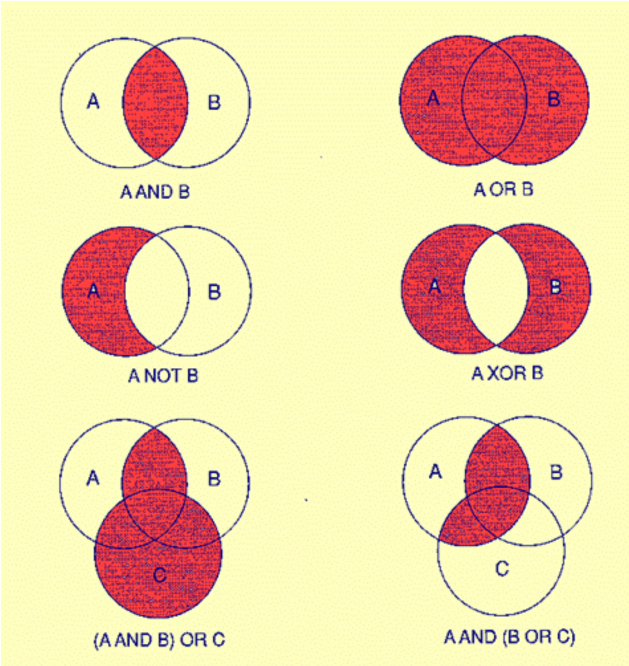
Result = if(*expression1* operator *expression2*)

all inputs and result are Boolean maps

operator can be: and, not, or, xor

For example:

```
pcrcalc bridge.map = river.map and roads.map
```

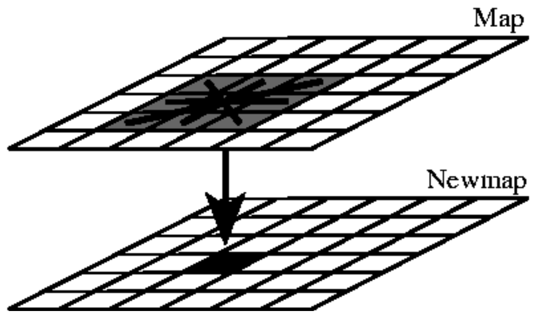


Direct neighbourhood functions

Direct neighbourhood functions

Functions in a window:

- window functions ('filters')
- derivatives of elevation maps



Direct neighbourhood functions: window functions (filters)

Calculate statistics in a square window of any size

Syntax:

Result = window...(*expression*,*windowlength*)

expression: variable over which statistics are calculated

windowlength: length of the window

window...: one of the window functions

examples of this group

operator	description

windowaverage	average of *expression*
windowtotal	sum of *expression* in window
windowdiversity	number of unique classes of *expression* in window
...	etc

For example:

```
pcrcalc smooth.map = windowaverage(dem.map, 5 * celllength(
```

Speakernotes

smooth.mod

Direct neighbourhood functions: derivatives of elevation

Calculate derivatives of an elevation model over a 3 x 3 cell window

Example 1:

Result = slope(*expression*)

expression: digital elevation model

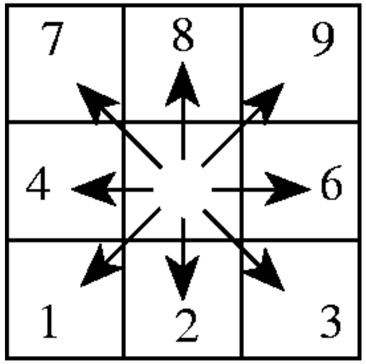
Example 2:

Result = lddcreate(*expression*, *outflowdepth*, *corevolume*, *corearea, catchmentprecipitation*)

expression: digitial elevation model

other arguments: pit removal options (see manual and practical)

Result: local drain direction map, each cell has a direction of flow to one of its neighbours



Speakernotes

demderiv.mod

Functions with a neighbourhood defined by topology

Functions with a neighbourhood defined by topology

Functions with topology given as an ldd:

- catchment analysis
- functions for transport of material (storage based modelling

Catchment analysis

Functions using the local drain direction map to extract catchment characteristics

examples of this group

operator	description
catchment	calculates catchment of a set of cells
path	calculates downstream path from a set of cells
upstream	assigns value of upstream neighbour to each cell
...	etc

For example:

```
pcrcalc catchment.map = catchment(ldd.map, outlocs.map)
```

Speakernotes

catch.mod

Functions for transport of material: cell as open system

Accu.. family of functions:

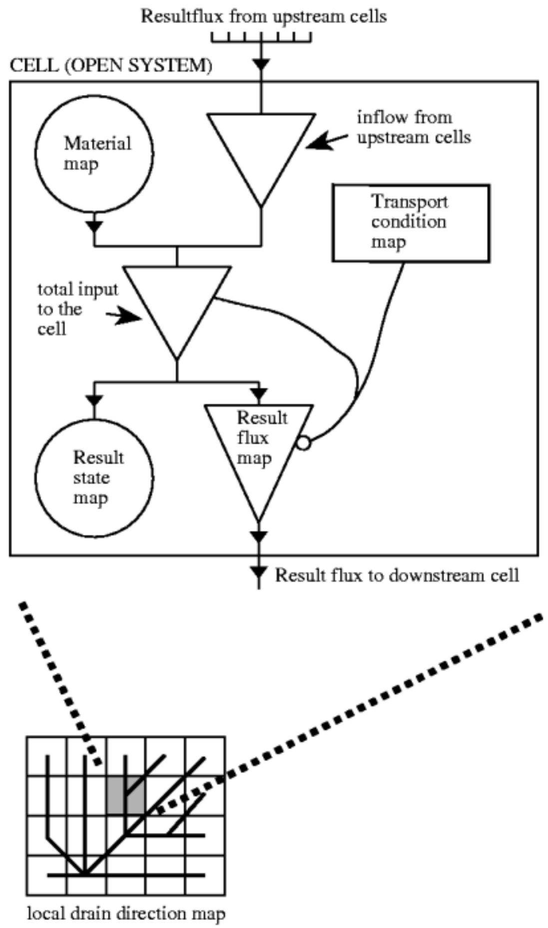
Flux = accu...flux(*Ldd*,*Material*,*TransportCondition*)

State = accu...state(*Ldd*,*Material*,*TransportCondition*)

Ldd: local drain direction map

Material: input of material (e.g. water)

TransportCondition: infiltration capacity



Speakernotes

mod_05

Functions for transport of material: cell as open system

Different types of transport conditions:

examples of this group

functions	description
accuflux	everything flows down
accufractionflux	for each cell, fraction flows down and rest stays in cell
accuthreshold	for each cell, outflow occurs when threshold is exceeded
...	etc

For instance

```
pcrcalc runoff.map, infil.map = accuthresholdflux, accuthre
```

Speakernotes

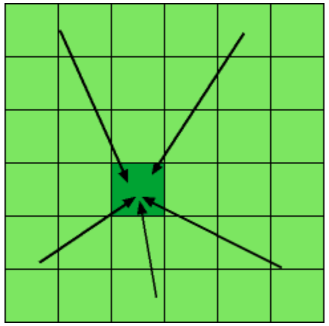
runoff.mod

Entire neighbourhood functions

Entire neighbourhood functions

Value of a cell is function of all other cells on the map

- distance calculation (spread)
- viewshed analysis (visible areas)
- groundwater flow



Functions calculating descriptive statistics

Functions calculating descriptive statistics

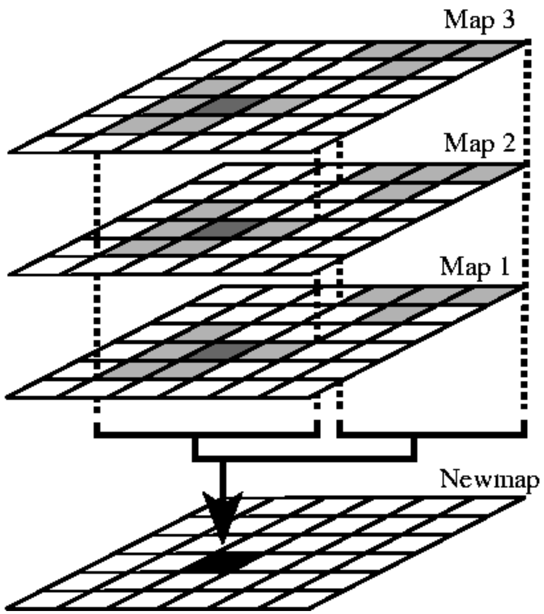
Value of cell is function of cell values in a certain area

Calculate descriptive statistics over an area, e.g. average height per soil class

For instance

```
AverageInfiltration=areaaverage(Infiltration,Soils)
```

calculates the average infiltration for each class on Soils and assigns it to all cells in that class



Example: geomorphological mapping with the DEM

Terraces

First terrace: height between 930 and 970 m

Second terrace: height between 845 and 890

Speakernotes

terrace_st1.mod, terrace_st2.mod

Tithonique.., rivers..

Slope..?

What else can we use?

Speakernotes

titho.mod, geom.mod

Spatial dynamic modelling

Theory

$z(1..m)(t+1)=f(z(1..m)(t), i(1..n)(t), p(1..l), t)$

$i(1..n)$: inputs (maps)

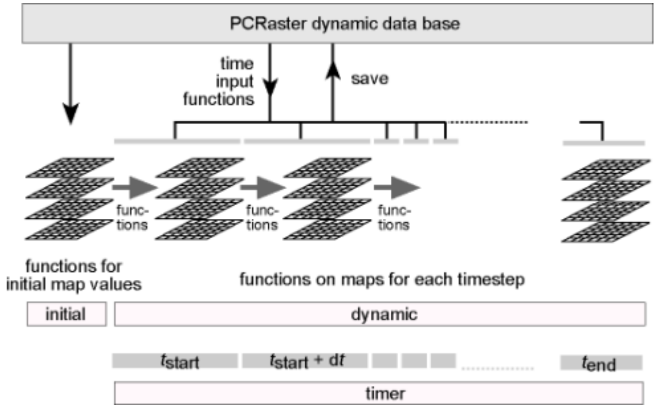
$p(1..l)$: parameters (maps)

$z(1..m)$: model variables

f: the model (set of functions run for each timestep)

Representation in PCRaster

Dynamic modelling script: representing*f*, *z*, *i* and *p* for all timesteps



Spatial dynamic modelling: example

Modelling denudation of the French Alpes

What do we need to calculate for each time step (e.g. 1000 yr)?:

- amount of erosion for each cell
- subtract erosion from elevation model

How can we calculate erosion?

Spatial dynamic modelling: stream power

Erosion (*E*) over a certain period of time is

E = *Runoff*^*A* * *Slope*^*B*

with:

Runoff: total runoff (m)

A: parameter

Slope: slope (-)

B: parameter

Speakernotes

erosion.mod