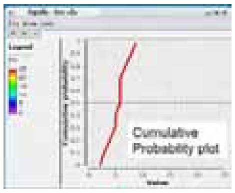
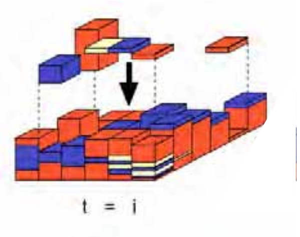
Dynamic modelling in GIS: modelling in three spatial dimensions and error propagation modelling

Derek Karssenberg





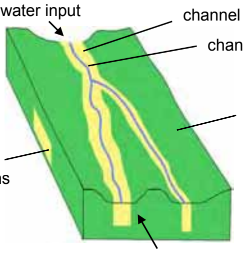
PCRaster research team, Department of Physical Geography,
Faculty of Geosciences, Utrecht University

Computer modelling in the earth sciences

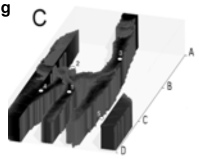
- Forward modelling through time: time steps
- Process based modelling: numerical solution of physical equations
- Spatial processes: 2D and 3D spatial domain
- Error propagation through Monte Carlo simulation

Example model: process-based alluvial architecture modelling

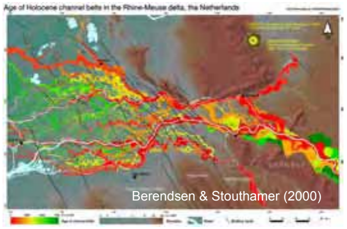
- forward modelling
- processes of erosion and deposition
- channel bifurcation and avulsion



Relevance of alluvial architecture modelling

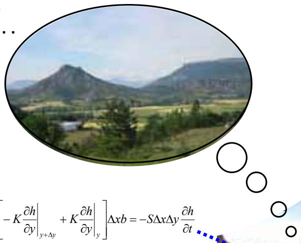


- predicting oil and water resources
- studying fluvial systems under external forcing (climate change)


Age of Holocene channel belts in the Rhine-Meuse delta, the Netherlands

Berendsen & Stouthamer (2000)

Researchers are good in science . .



$$\left[-K\frac{\partial h}{\partial x}\bigg|_{x+\Delta x}+K\frac{\partial h}{\partial x}\bigg|_{x}\right]\Delta yb+\left[-K\frac{\partial h}{\partial y}\bigg|_{y+\Delta y}+K\frac{\partial h}{\partial y}\bigg|_{y}\right]\Delta xb=-S\Delta x\Delta y\frac{\partial h}{\partial t}$$

Researchers are not programmers

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <vsa.h>
#include <vsa_font.h>

extern unsigned _stklen = 20000;
void main()
{
            char text[100];
            clock_t clk,delta;
            unsigned mode,x[4096],y[4096];
            unsigned i,num_lines,xl,yt,xr,yb;
            float timer;
            printf("input graphics mode (hex): ");
            scanf("%x",&mode);
            vsa_init(mode);
```

Software engineers are good programmers . .

$$\left[-K\frac{\partial h}{\partial x}\bigg|_{x+\Delta x}+K\frac{\partial h}{\partial x}\bigg|_{x}\right]\Delta yb+\left[-K\frac{\partial h}{\partial y}\bigg|_{y+\Delta y}+K\frac{\partial h}{\partial y}\bigg|_{y}\right]\Delta xb=-S\Delta x\Delta y\frac{\partial h}{\partial t}$$

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <vsa.h>
#include <vsa_font.h>

extern unsigned _stklen = 20000;
void main()
{
            char text[100];
            clock_t clk,delta;
            unsigned mode,x[4096],y[4096];
            unsigned i,num_lines,xl,yt,xr,yb;
            float timer;
            printf("input graphics mode (hex): ");
            scanf("%x",&mode);
            vsa_init(mode);
```

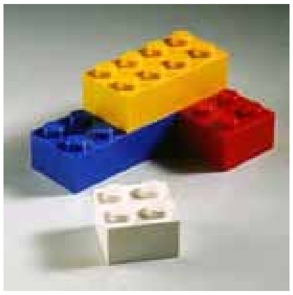
Solution: model construction with building blocks

1) Software engineers develop a model construction toolbox:

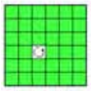
- building blocks: spatio-temporal functions
- framework to glue together functions
- visualisation routines that read data written by the framework

Solution: model construction with building blocks

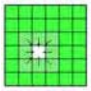
2) Researchers construct models with the toolbox



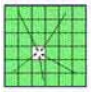
Building blocks: classic functions on raster maps



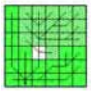
point functions



direct neighbourhood functions

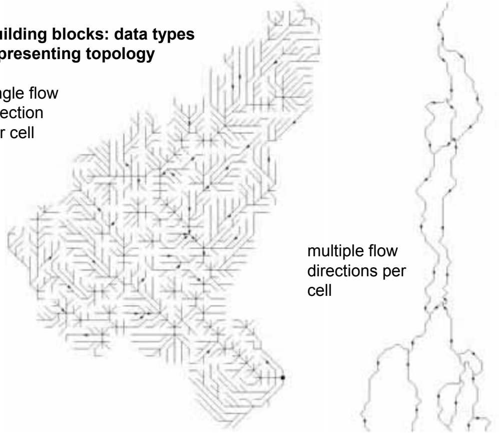


entire neighbourhood functions



neighbourhood functions with a given topology

Building blocks: data types representing topology



single flow direction per cell

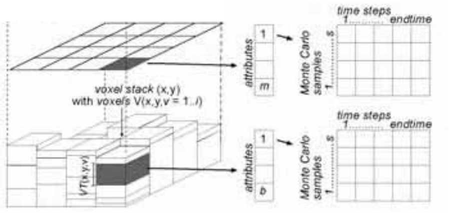
multiple flow directions per cell

Video dynamic evolution of channels

3D building blocks

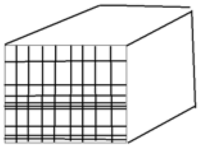
map

block

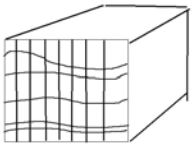


Vertical discretization

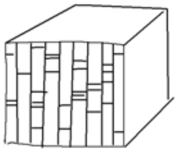
both



and

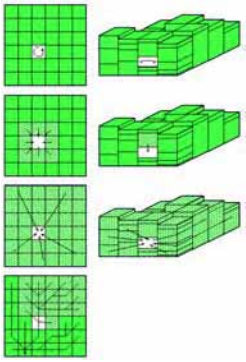


are subsets of



(applied here)

Building blocks: standard functions on maps and blocks



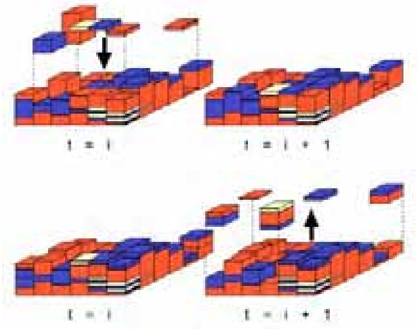
point functions

direct neighbourhood functions

entire neighbourhood functions

neighbourhood defined by topology

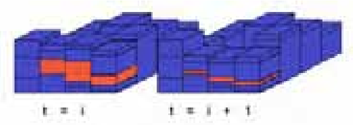
Building blocks: functions on blocks, change of form



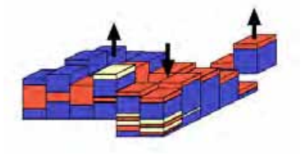
addition

removal

Building blocks: functions on blocks, change of form



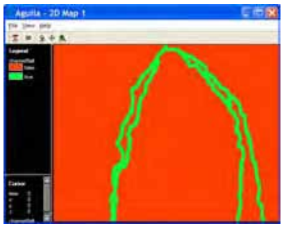
inside block form

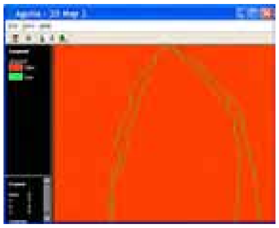


vertical movement

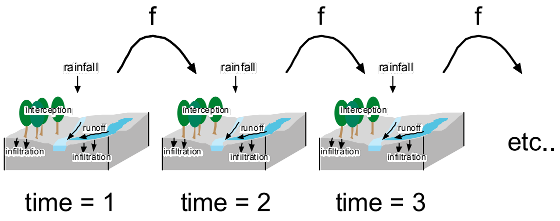
Glueing together the building blocks: functions in a script

```
...
beltRadius= channelBeltWidth/2
channelBelt = spread(channel,0,1) < beltRadius
...
```





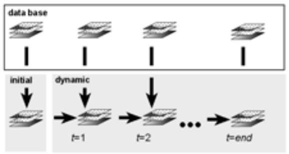
Framework to represent time



Order of calculations:

```
run functions calculating initial state
for t in timesteps:
    run functions representing f
```

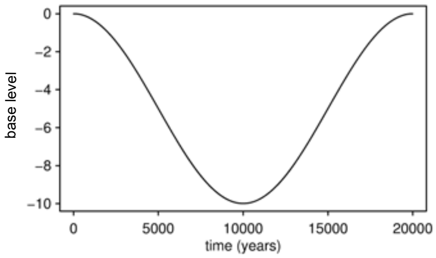
Framework: structured script

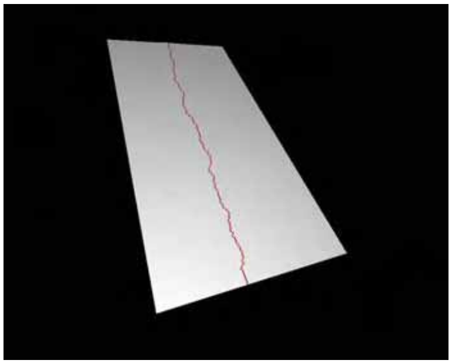


```
initial
  # sequence of functions

dynamic (nrtimesteps=…)
  # sequence of functions
```

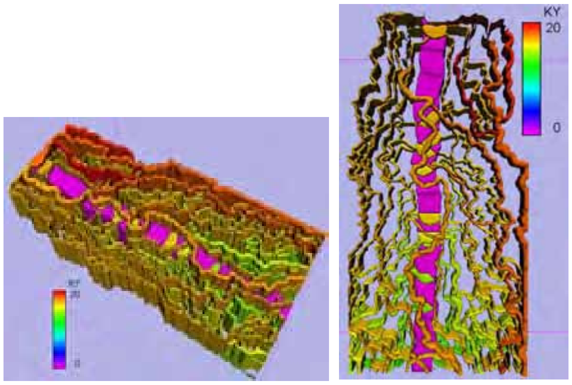
Standard run

- parameters and boundary conditions comparable to Rhine-Meuse
- cell size 200 m, modelling area 30 x 60 km
- external forcing: base level change

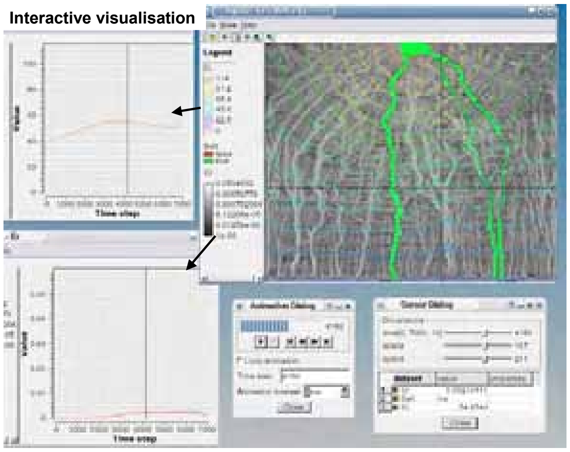




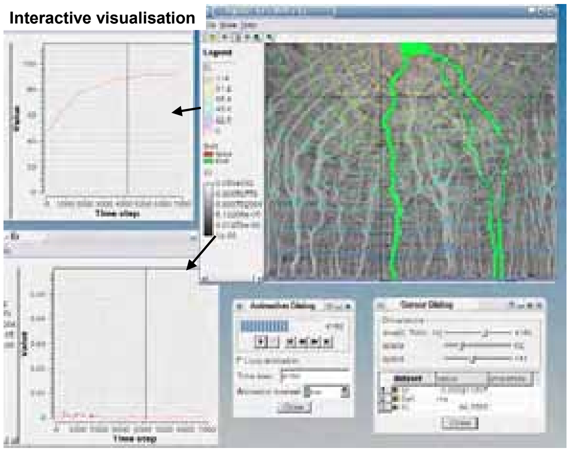
Channel belt distribution



Interactive visualisation



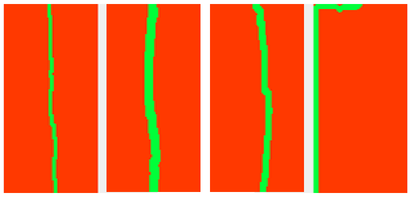
Interactive visualisation



Dealing with stochastic variables: Monte Carlo simulation

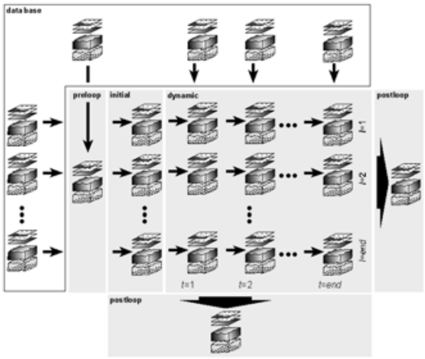
- Many models use stochastic variables
- Application of Monte Carlo simulation

Realizations →



channel
belt location
through time,
25 steps

Framework Monte Carlo simulation

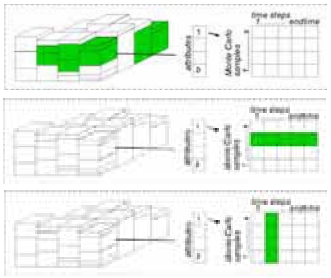


Representing the stochastic dimension in a script: framework

```
preloop
 # sequence of functions

initial (nrloops=…)
# sequence of functions

dynamic (nrtimesteps=…)
# sequence of functions

postloop
# sequence of functions
```

Functions calculating descriptive statistics

calculate a statistical value of attribute values

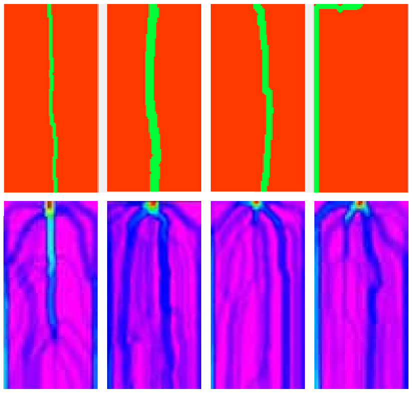


spatial dimension

time dimension

stochastic dimension

Realizations →

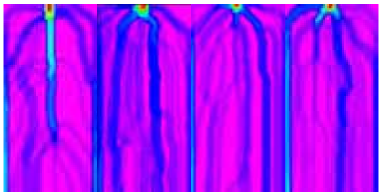


channel
belt location
through time,
25 steps

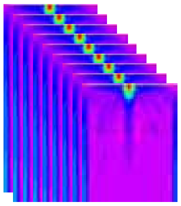
|
aggregate
over time
↓

number
of stacked
channel
belts at
end of run

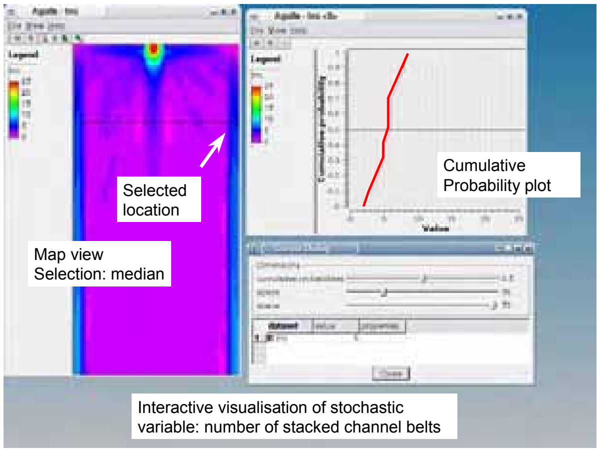
Realizations (number of stacked channel belts)



aggregate over Monte Carlo realizations →



Stochastic variable represented by quantiles (e.g. 0.1,0.2,.. ,0.9)

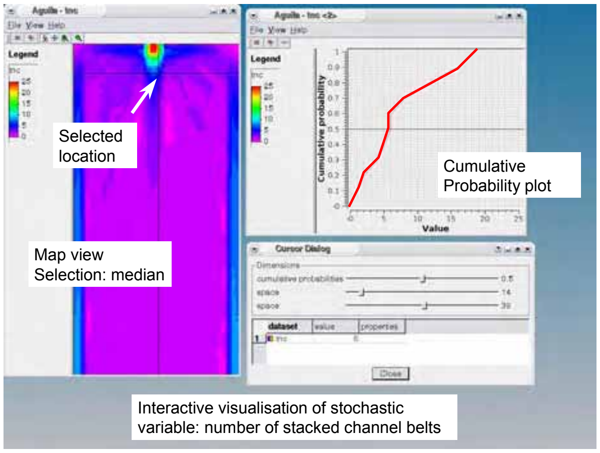


Selected location

Map view
Selection: median

Cumulative Probability plot

Interactive visualisation of stochastic variable: number of stacked channel belts



Selected location

Map view
Selection: median

Cumulative Probability plot

Interactive visualisation of stochastic variable: number of stacked channel belts

How do we provide these tools to the user?

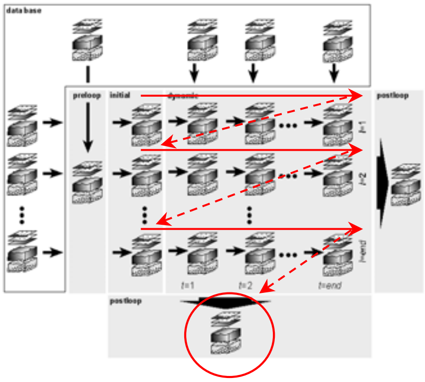
PCRaster PCRcalc

- language completely tailored to modelling in time and space
- very easy to use

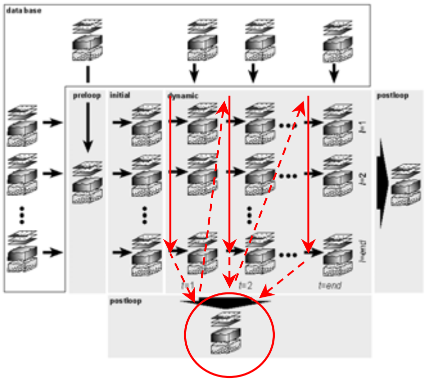
PCRaster Python

- Python: generic scripting language (used in Arc/Info, Google)
- all functionality for spatial temporal modelling is available in Python (Python extension)
- somewhat more difficult to use, but more flexibility

Challenges for the future: order of calculation and I/O



Challenges for the future: order of calculation and I/O



Final remarks, conclusions

- programmers developing blocks and researchers building models by combining these blocks is a very efficient approach in research
- standard tools are available now for 3D modelling and error propagation modelling
- optimization in a language completely tailored to dealing with all dimensions is a challenge for the future

Acknowledgements

Peter Burrough, Willem van Deursen, Kor de Jong,
Edzer Pebesma, Cees Wesseling

And many other people providing all kinds of support and ideas

Info at: http://pcraster.geo.uu.nl