

PCRaster : logiciel pour la modélisation environnementale

Pour : les logiciels libres au service des SIG

7 décembre 2015 (14h00 / 17h30)

Paul van Dijk



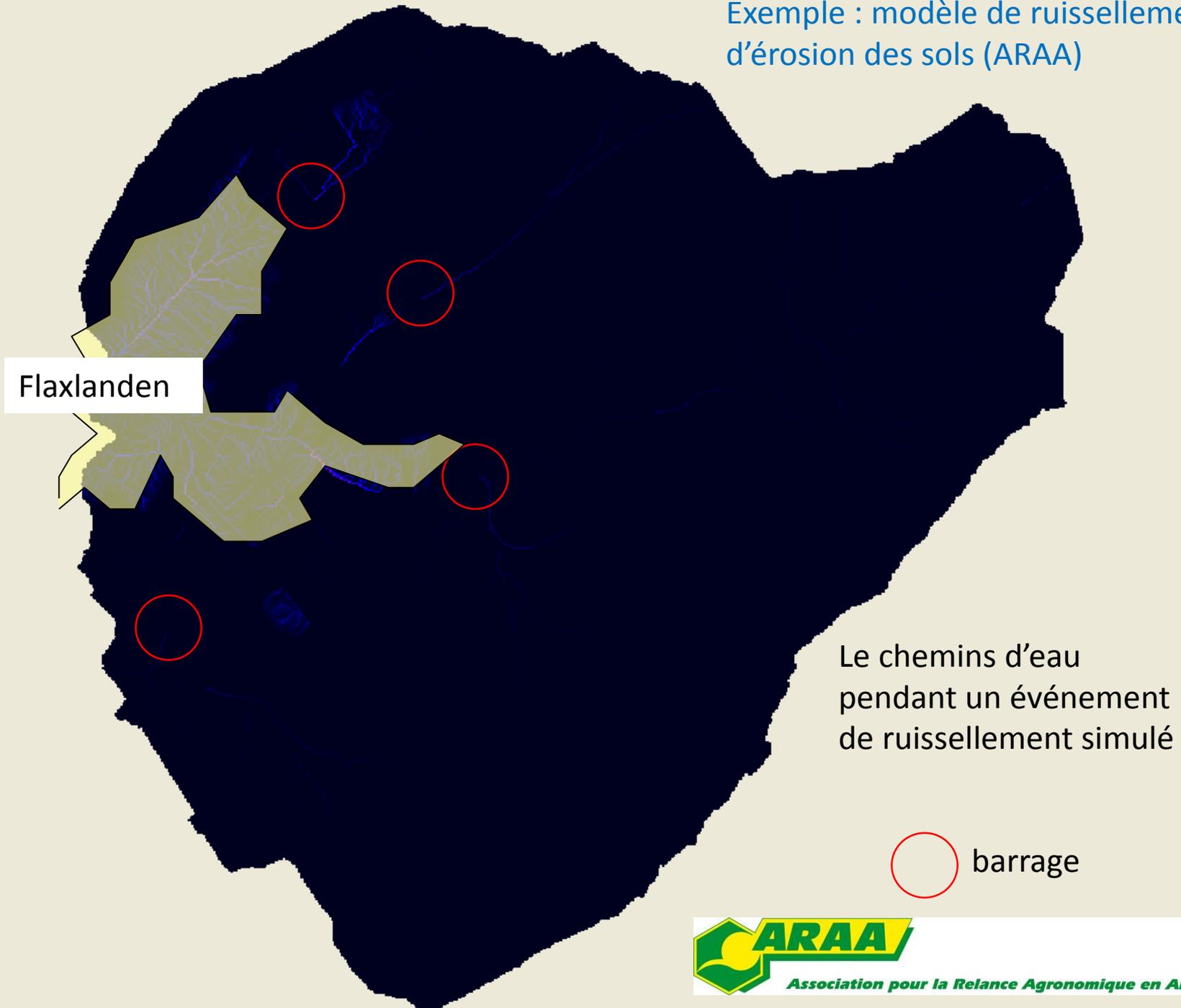
PCRaster : outils pour la construction des modèles environnementaux dynamiques et spatialisés

- Opensource GPLv3, as of version 4) pour windows et linux
- Téléchargement gratuit sur : <http://pcraster.geo.uu.nl/>
- Langage très accessible pour créer des scripts permettant de développer ses propres modèles de simulation environnementale
- Aussi disponible en module PYTHON
- Mode raster
- Surtout utilisé en géographie, géomorphologie, hydrologie et écologie, ... Exemples : modèles d'écoulements de surface et des eaux souterraines (version de MODFLOW existe en PCRaster), érosion des sols, transferts des pesticides, compétition de entre des espèces de végétation, la neige en montagne avec le changement climatique, stabilité des pentes, ...
- Bonne documentation : <http://pcraster.geo.uu.nl/pcraster/4.1.0/doc/manual/index.html>
- Rapide
- Contient des outils de visualisation des données spatio-temporelles

Pcraster n'est pas :

- Un GIS complet. A utiliser en combinaison avec des SIG de base comme QSIG (qui peut lire les cartes en format PCRaster = .map)

Exemple : modèle de ruissellement et d'érosion des sols (ARAA)



Flaxlanden

Le chemins d'eau pendant un événement de ruissellement simulé

○ barrage

Exemple : outil de visualisation de PCRaster

The screenshot displays the PCRaster visualization tool interface, which is organized into several panels:

- Top Row:**
 - Left:** A map showing a variable 's' with a color scale from 0.0 (green) to 0.5 (red). A legend indicates values of 2000, 1000, and 0.
 - Middle:** A cumulative probability plot for variable 's'. The x-axis is 'Value' (0 to 0.5) and the y-axis is 'Cumulative probability' (0 to 0.8). A red curve shows the distribution, with a horizontal line at approximately 0.5.
 - Right:** A time series plot for variable 's'. The x-axis is 'Time step' (0 to 150) and the y-axis is 'Value' (0 to 0.5). A red line shows the value over time, with a vertical line at time step 145.
- Second Row:** Similar to the top row, but with a legend for 's' ranging from 0.0 to 0.5 and a legend for precipitation values of 4000, 3000, 2000, and 1000.
- Bottom Left:** A code editor showing Python code for a dynamic model:


```

def dynamic(self):
    temperatureObserved = self.readmap("tavgo")
    precipitationObserved = self.readmap("pr")
    precipitation = max(0, precipitationObserved * (
    temperature = temperatureObserved - self.temperature
    snowFall = ifthenelse(temperature < 0, precipitation
    self.snow = self.snow + snowFall
    potentialMelt = ifthenelse(temperature > 0, temper
    actualMelt = min(self.snow, potentialMelt)
    self.snow = max(0, self.snow - actualMelt)
    rain = ifthenelse(temperature >= 0, precipitation
    discharge = accuflux(self.ldd, actualMelt + rain)
    self.report(self.snow, "s")
    self.report(discharge, "q")

def postmclloop(self):
    ~/snowModel/snowModel.py FF=unix TYPE=PYTHON RC=27
      
```
- Bottom Center:** An 'Animation Dialog' window with a slider set to 80%, playback controls, and a 'Time step' field set to 145. The 'Animation interval' is set to 300ms.
- Bottom Right:** A 'Cursor And Values' window showing dimensions and a table of dataset values:

dataset	value	properties
1 s	0.323428	
2 dem.map	2531	