GENTÆNK GRUNDVANDSMODELLERING - NYE METODER TIL EFFEKTIV MODELUDVIKLING

T.N. Vilhelmsen, S. Christensen, R.R. Frederiksen, E. Auken, A.V. Christiansen

AARHUS UNIVERSITY
DEPARTMENT OF GEOSCIENCE

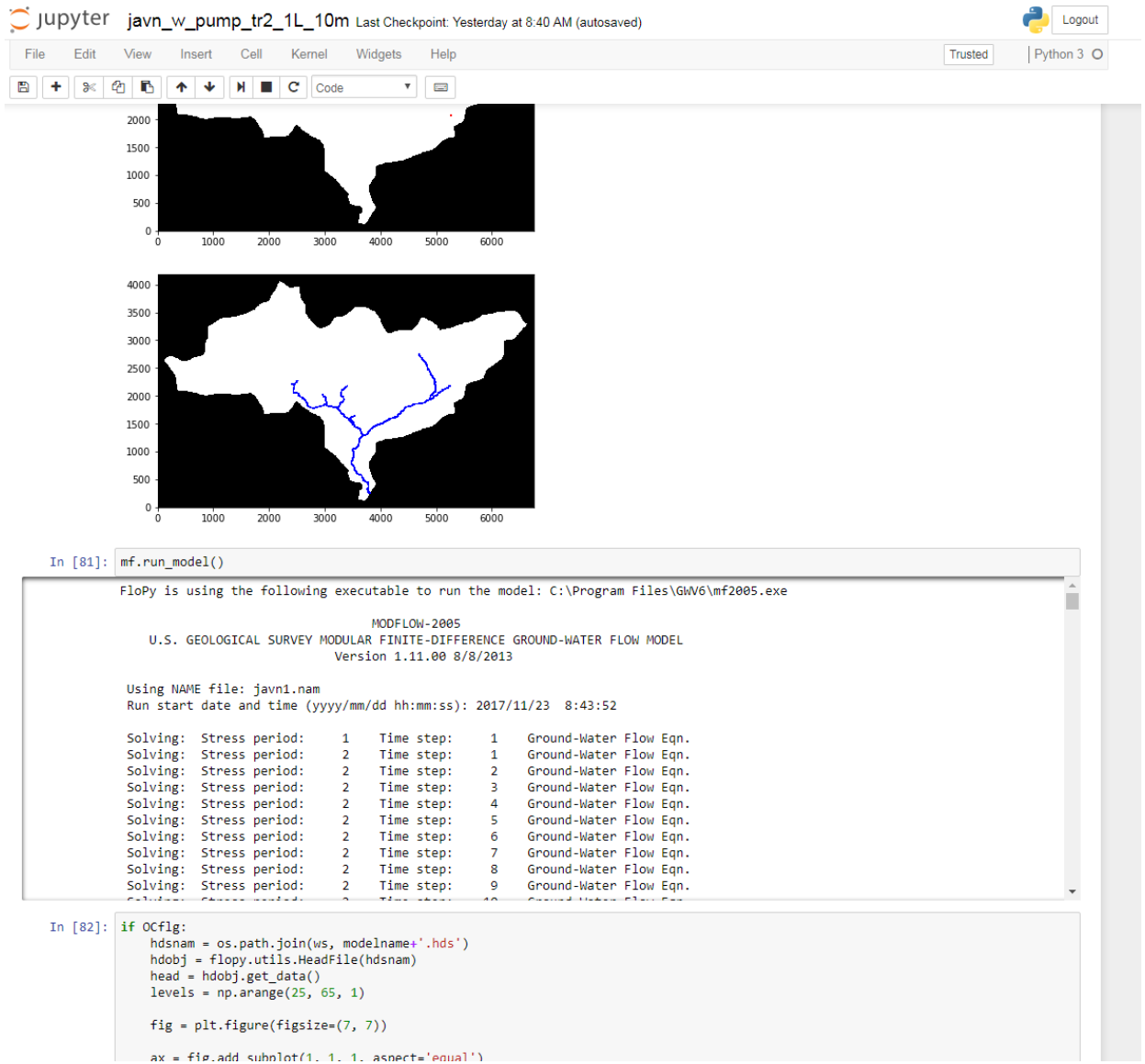
ATV-VINTERMØDE 2018
7. MARTS 2018

TROELS NORVIN VILHELMSEN
POSTDOC

HydroGeophysics Group
AARHUS UNIVERSITY

INDHOLD

- Introduktion – rOpen forskningsprojektet
- Script-baseret grundvandsmodellering
- Eksempler på data mining
- Eksempler på modelopbygning
- Konklusion og perspektiver



ROPEN FORSKNINGSPROJEKTET

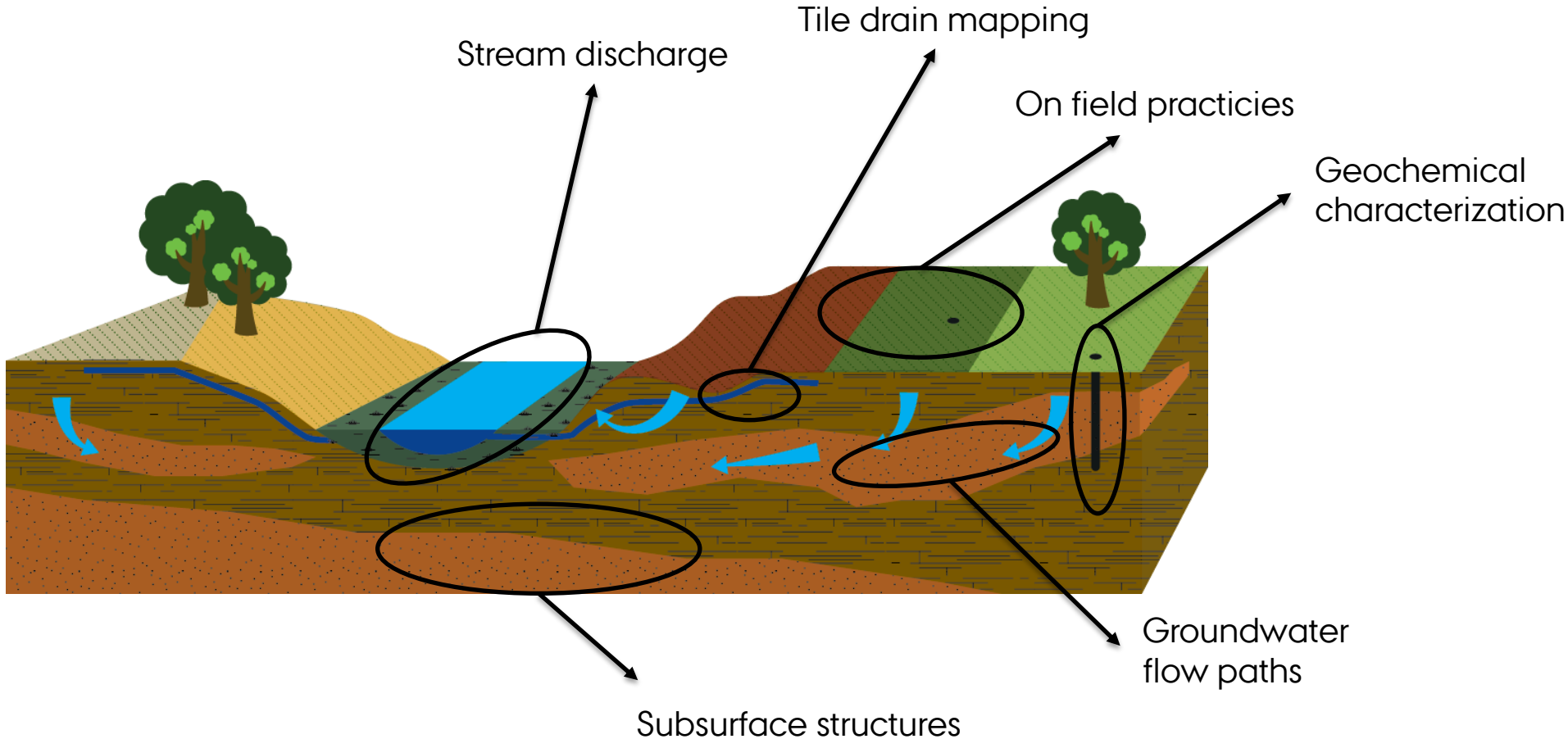
Photo: SEGES

”Identifikation of landbrugsarealer som er følsomme over for udvaskning af nitrat fra rodzonen”

Innovation Fund Denmark

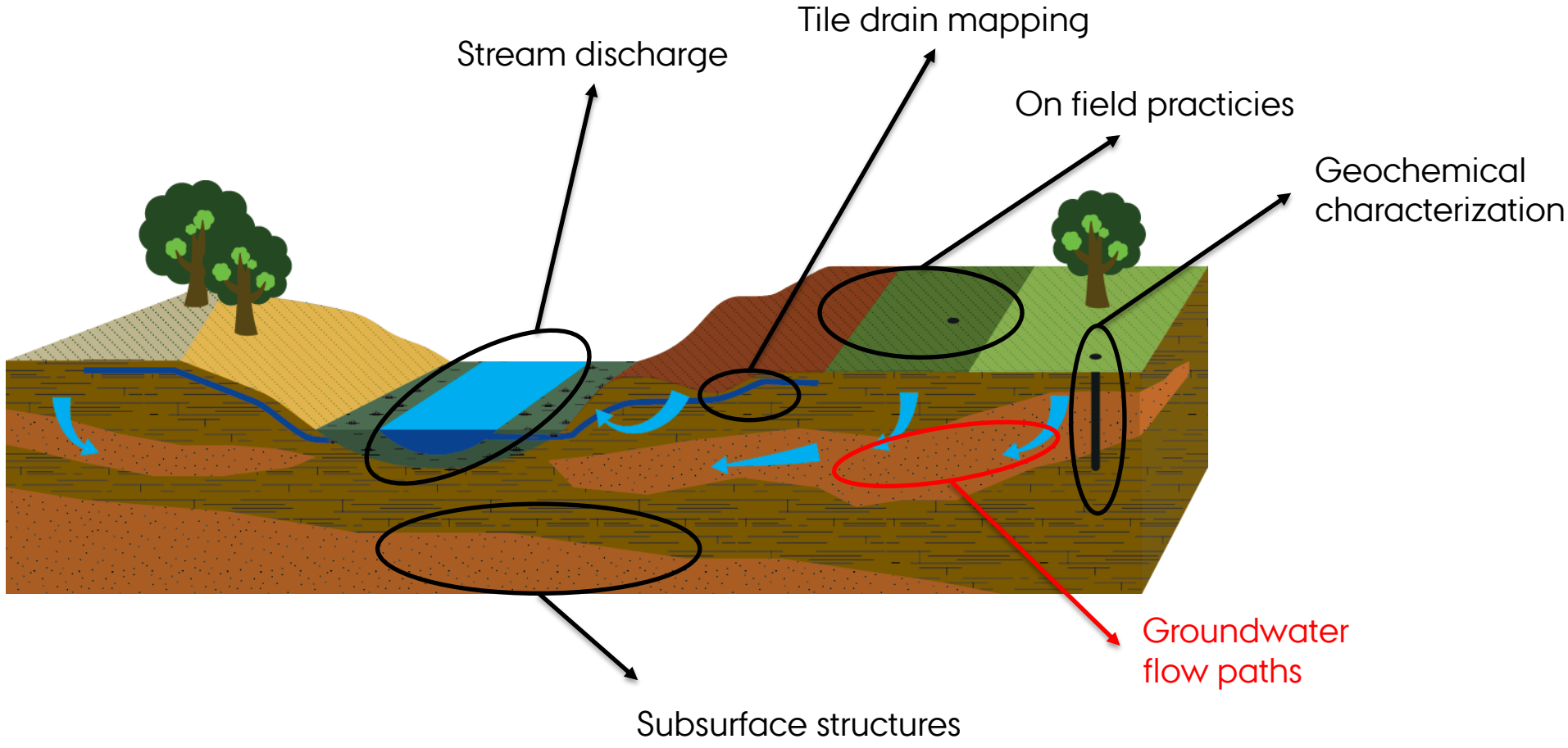
ROPEN FORSKNINGSPROJEKTET

Analysis and management



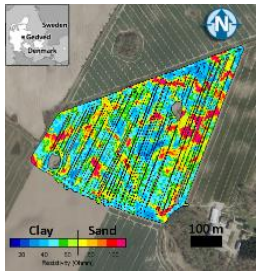
ROPEN FORSKNINGSPROJEKTET

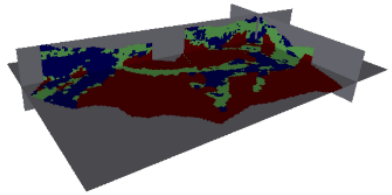
Analysis and management



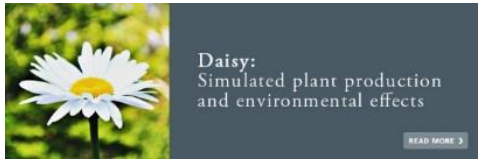
Dynamic model update with new data collection

Geophysical mapping and modelling





Soil – crop modelling

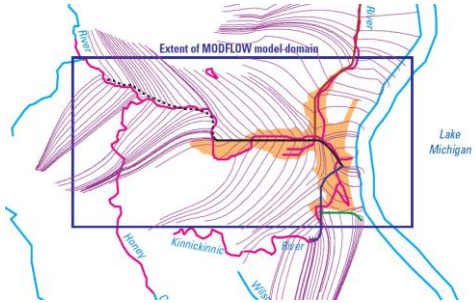


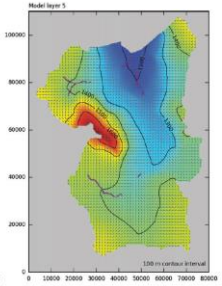
Hansen, Søren., Per Abrahamsen, Carsten T. Petersen, and Merete Styczen (2012). Daisy: Model use, calibration, and validation. Trans. ASABE 55(4): 1315-1333.

+ $N\text{-}LES_4$

Kristian Kristensen, Jesper Waagepetersen, Christen Duus Børgesen, Finn Pilgaard Vinther, Ruth Grant and Gitte Blicher-Mathiesen (2008), Reestimation and further development in the model N-LES - $N\text{-}LES_3$ to $N\text{-}LES_4$

Groundwater modelling

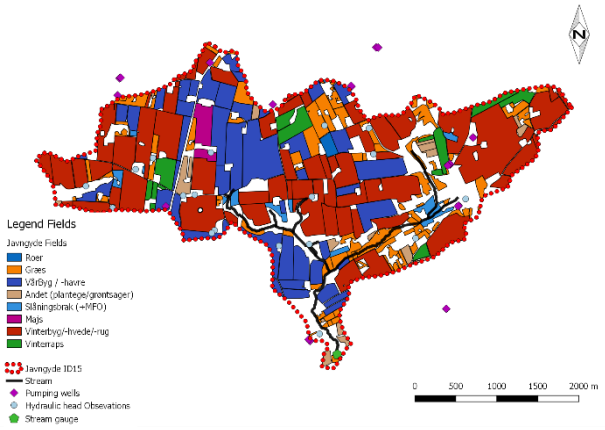




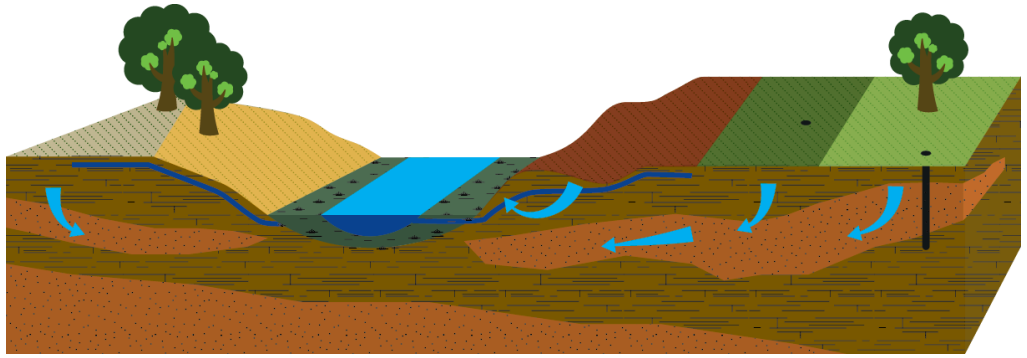
Harbaugh, A.W., 2005, MODFLOW-2005, The U.S. Geological Survey modular ground water model—the Ground-Water Flow Process: U.S. Geological Survey Techniques and Methods 6-A16, variously p

Pollock, D.W., 2016, User guide for MODPATH Version 7—A particle-tracking model fo MODFLOW: U.S. Geological Survey Open-File Report 2016–1086, 35 p., http://dx.doi.org/10.3133/ofr20161086.

GIS, land use, lithology, etc.

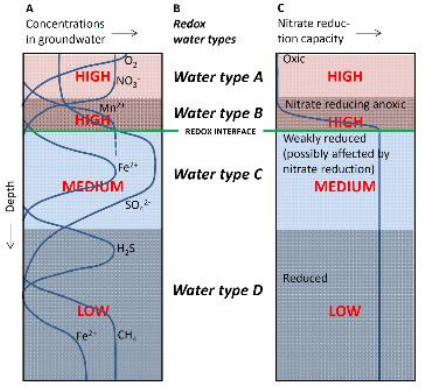


rOpen Decision support



- N-transport
- Vulnerability, streams and groundwater
- Nitrate degradation
- Identification of sensitive areas

Geochemical modelling



ROPEN FORSKNINGSPROJEKTET

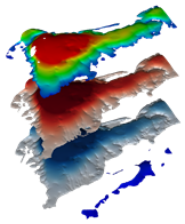
HVAD ER SCRIPT MODELLERING?

Groundwater

Methods Note/

Scripting MODFLOW Model Development Using Python and FloPy

by M. Bakker[1], V. Post[2,3], C. D. Langevin[4], J. D. Hughes[4], J. T. White[5], J. J. Starn[6], and M. N. Fienen[7]



FloPy$_3$

a Python package to create, run, and post-process MODFLOW-based models

```
1  def cf_model(model, layer, row, col, base, Q):
2      model.remove_package('WEL')
3      lrcQ = {1: [[layer, row, col, Q]]}
4      wel = fpm.ModflowWel(model=model, stress_period_data=lrcQ)
5      wel.write_file()
6      model.run_model(silent=True)
7      hfile = fpu.HeadFile('DG.hds', precision='double')
8      cfile = fpu.CellBudgetFile('DG.cbc', precision='double')
9      step_period_list = hfile.get_kstpkper()
10     h = hfile.get_ts((layer, row, col))
11     cap_frac = np.zeros((len(step_period_list)))
12     for tstep, step_period in enumerate(step_period_list):
13         if h[tstep, 1] == model.lpf.hdry:
14             capfrac[tstep] = np.nan
15         else:
16             v1 = cfile.get_data(kstpkper=step_period,
17                                 text='DRAINS', full3D=True)
18             v2 = cfile.get_data(kstpkper=step_period,
19                                 text='STREAM LEAKAGE', full3D=True)
20             v3 = cfile.get_data(kstpkper=step_period, text='ET',
21                                 full3D=True)
22             cap_frac[tstep] = ((v1[0].sum() + v2[0].sum() +
23                                 v3[0].sum()) - base) / (-Q)
24     return cap_frac
```

Python function to compute capture fraction of a well in a specific cell.

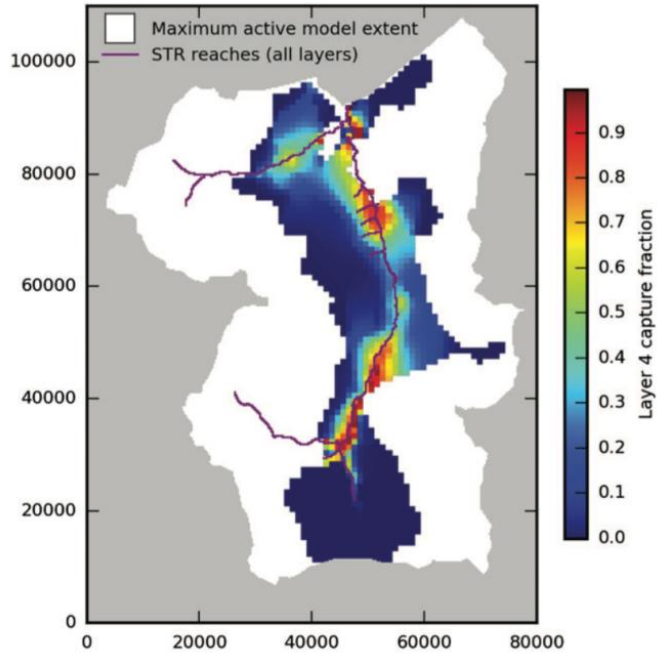


Figure 5. FloPy generated map showing the computed capture fraction of water from head-dependent boundaries as a function of well location in the Upper San Pedro Basin model layer corresponding to the lower basin fill after 10 years of pumpage. The maximum areal extent of the active model domain and the location of stream boundary conditions in all model layers are also shown.

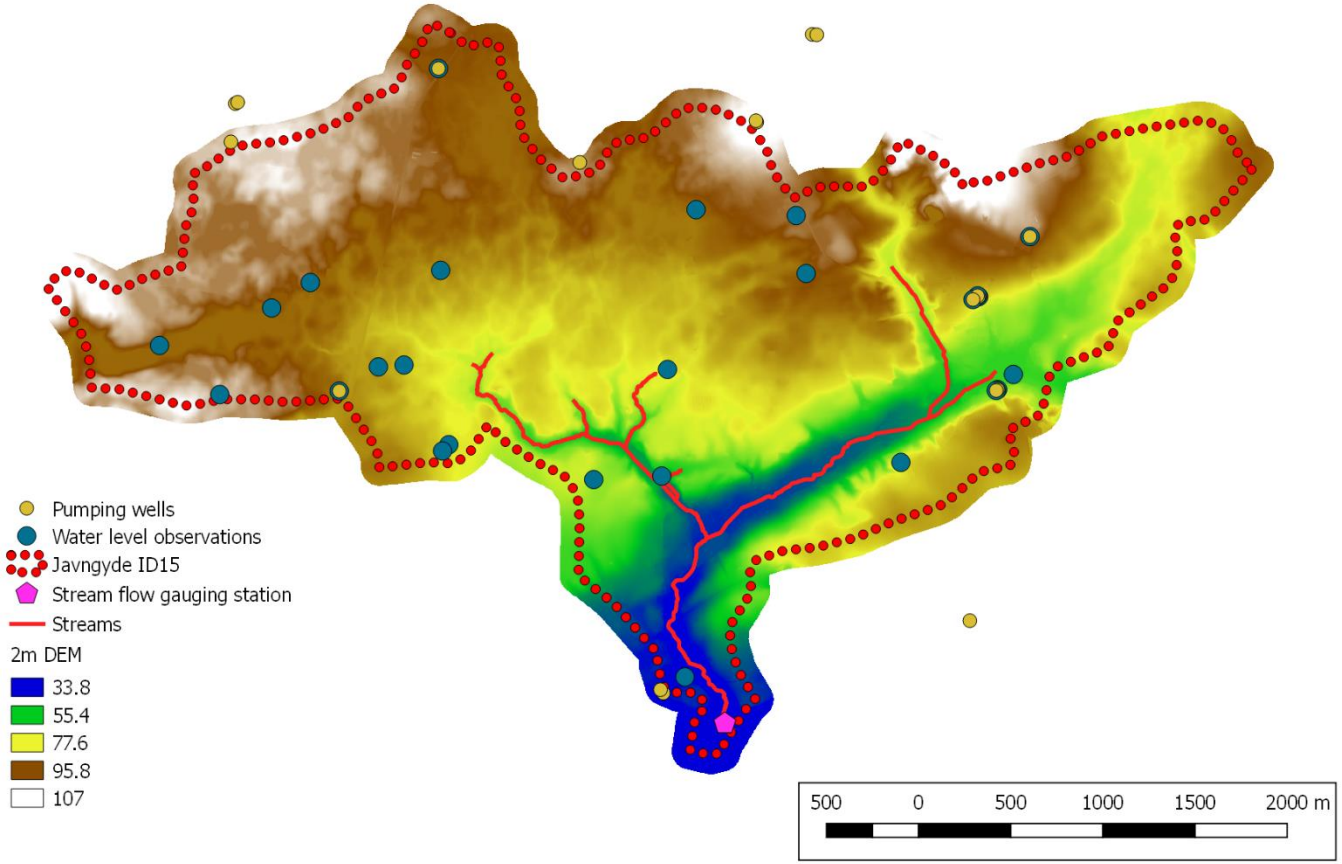
flopy has advantages both in assisting in efficient MODFLOW model building and in scripting scenarios for rapid evaluation. The example here shows compact open-source Python code used to generate the image of call capture

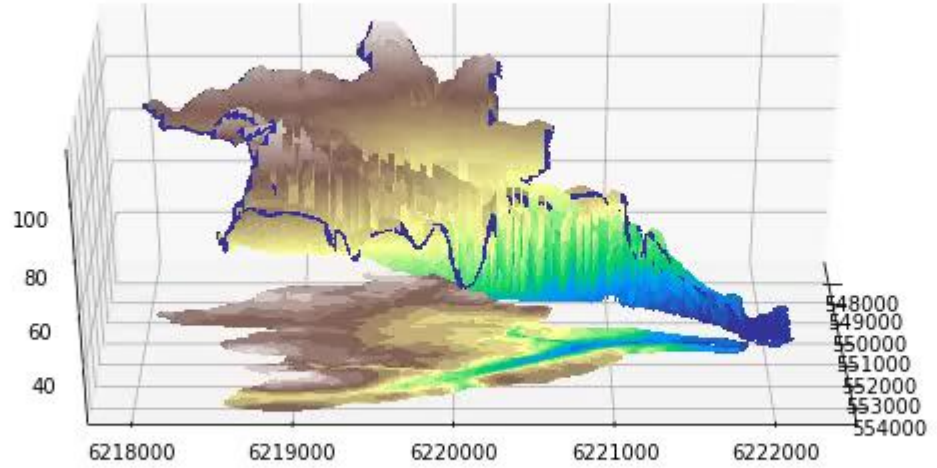
HVORFOR SCRIPT MODELLERING?

Hvorfor?

- Mange arbejdsgange er generiske og gentages hver gang en ny grundvandsmodel sættes op – disse kan i høj grad automatiseres
- I Danmark har vi en relativt unik tilgang til gratis offentlige data i databaser – udtrækning af information til grundvandsmodeller kan i høj grad automatiseres
- Typen af modelanalyser, der kan udføres, kan udvides, da fleksibiliteten i modelopsætningen forøges
- Modeller kan nemmere opdateres med ny data indsamling, da dette kan gøres semiautomatisk
- Modelopbygningen og dataanalysen er dokumenteret og gennemsigtig

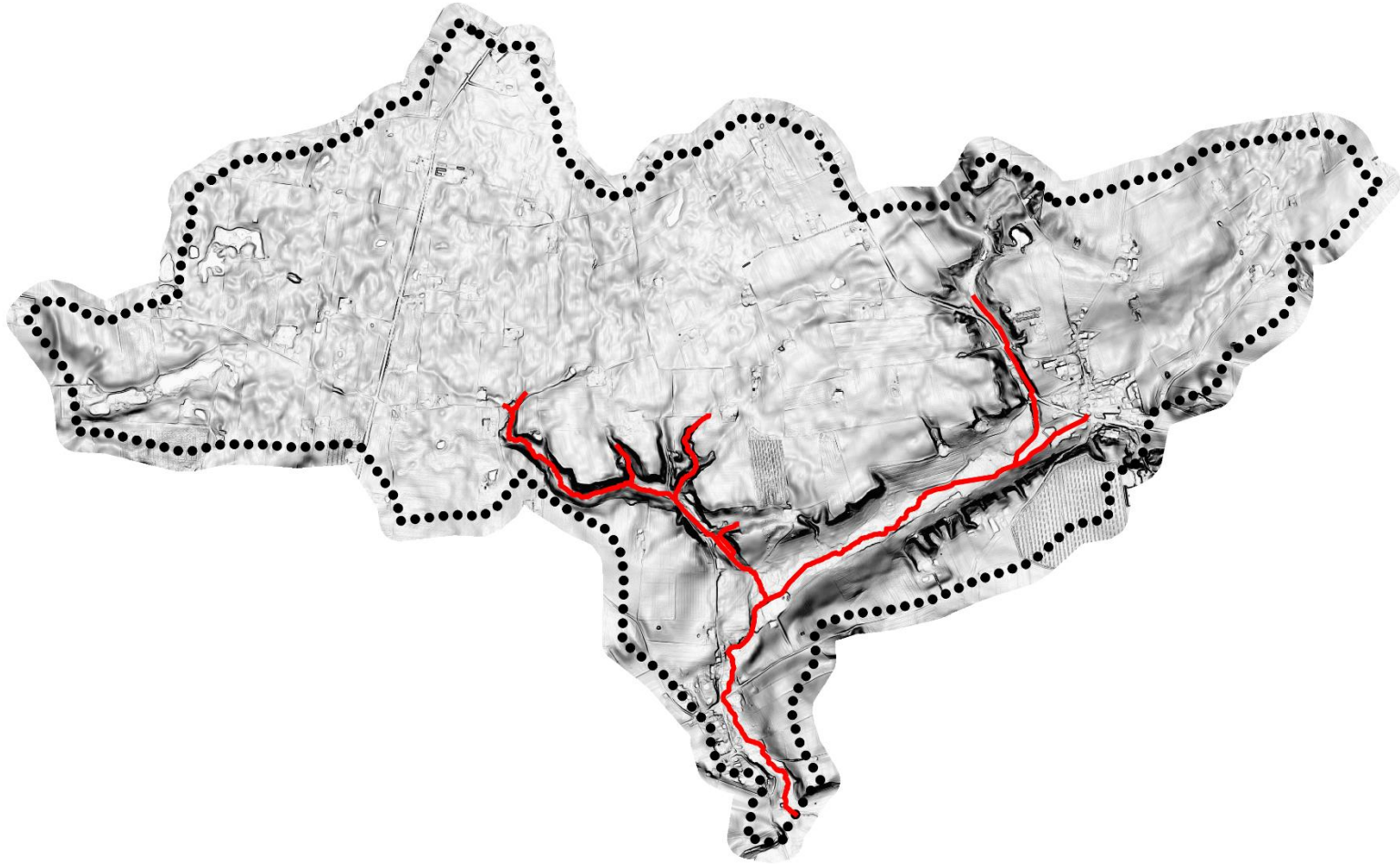
GENERERING AF VANDLØBSINPUT FRA DHM



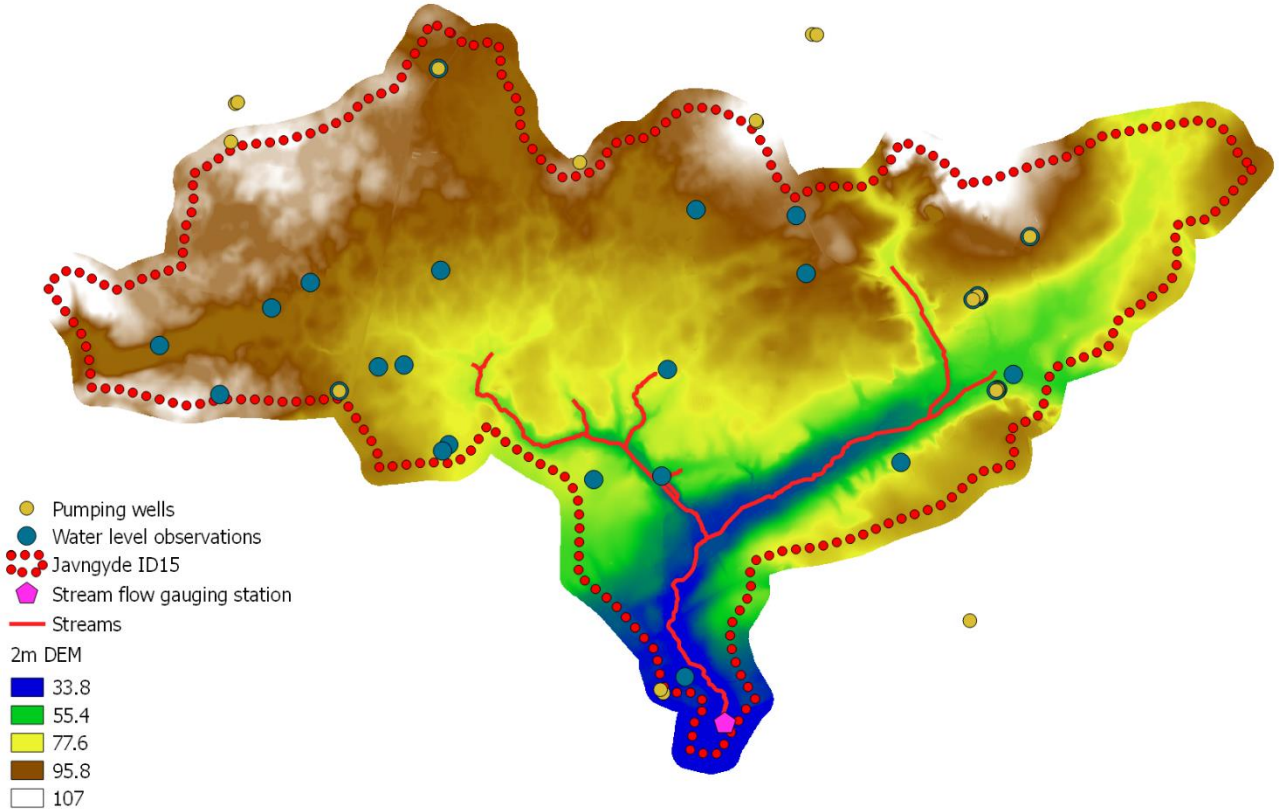


GENERERING AF VANDLØBSINPUT FRA DHM

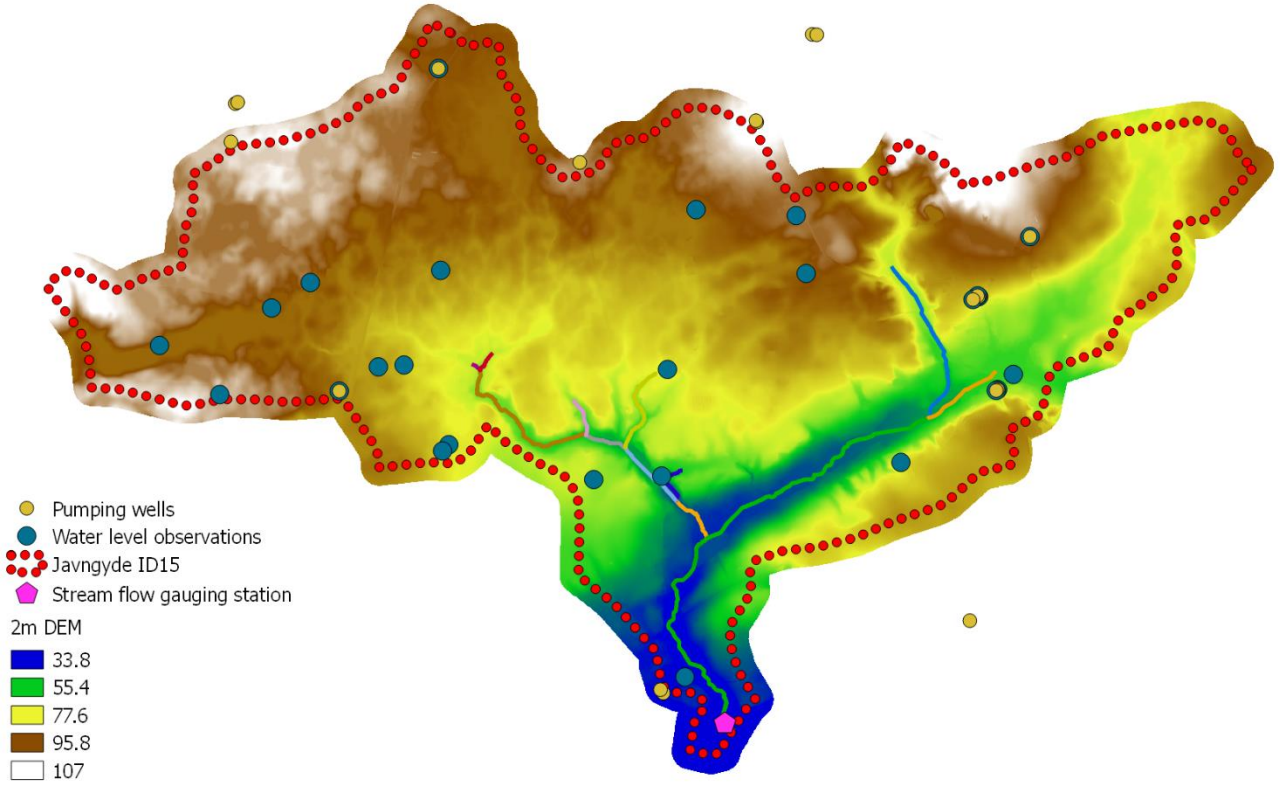
Terrain slope



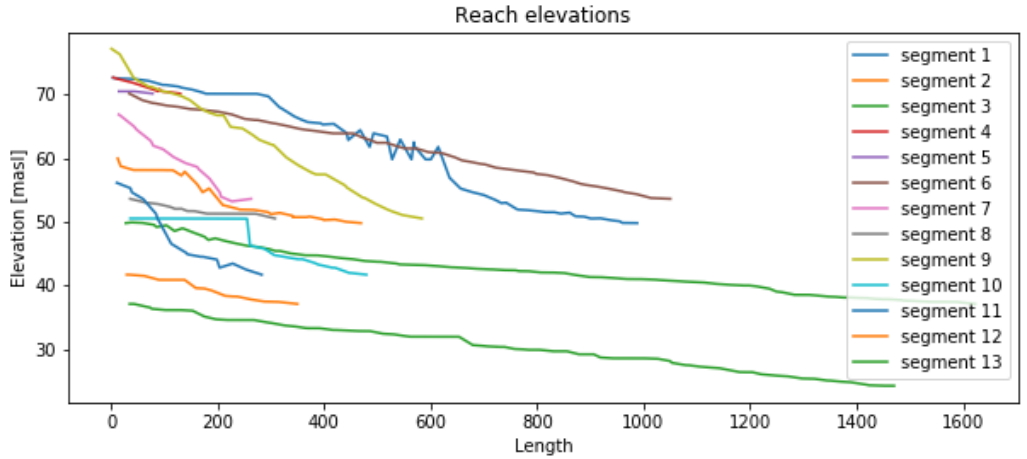
GENERERING AF VANDLØBSINPUT FRA DHM

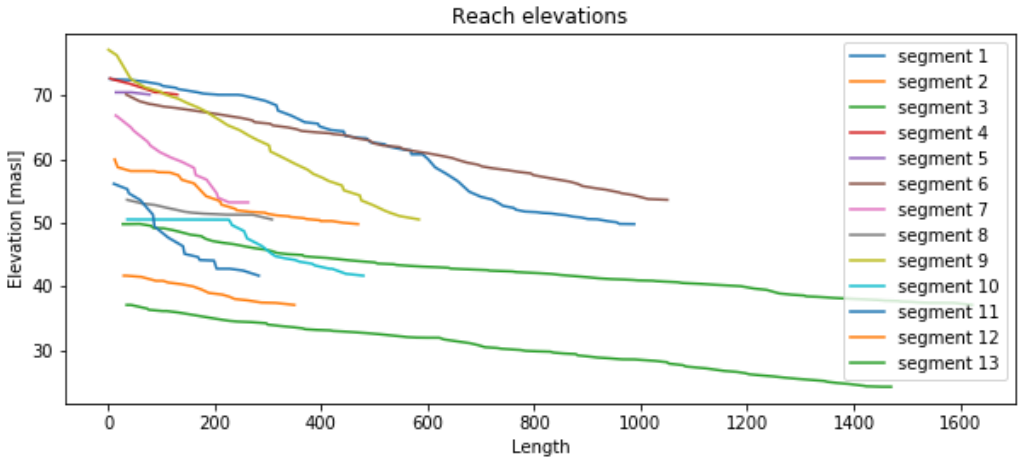


GENERERING AF VANDLØBSINPUT FRA DHM



Manuel definering af vandløbssektioner ud fra lokal kendskab





DATA MINING FRA JUPITER

Grundvandsindvinding fra Jupiter

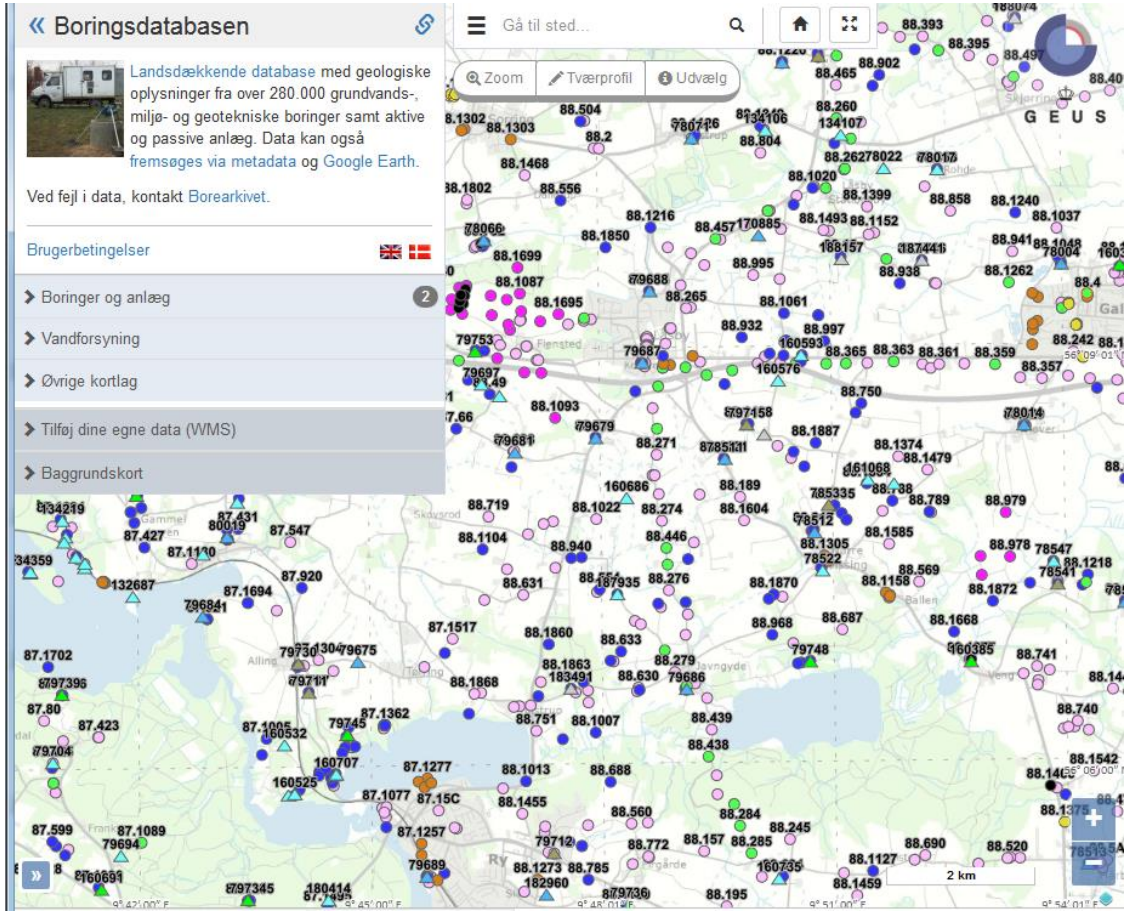
Define inputs for extracting information:

NB! format is hardwired to:

length: meters

time: days

In [2]:
```
# name of database. Must be Jupiter database en access 2000 format or newer:
data_base = 'z:\\Projects\\2017_rOPEN\\WP4\\jupiter\\db\\Jupiter_javngyde_xl_1.mdb'
# Base name for the outputs written to the disk:
area_name = 'Javngyde'
# Area outline: [xmin, xmax, ymin, ymax]
coordinates = [547490,554205,6218120,6222100]
# folder where csv of pandas dataframes are stored:
datadir = os.path.join('data')
# folder where shape files are stored:
shpdir =  os.path.join('shp')
# Length of each data period: 'AS' -> Anually start of year
period_length = 'AS'
# start of groundwater model simulation period:
start = '1993-01-01'
# end of groundwater model simulation period
end = '2005-01-01'

# check if directories exist. If not create them:
if not os.path.exists(datadir):
    os.makedirs(datadir)
if not os.path.exists(shpdir):
    os.makedirs(shpdir)
```



DATA MINING FRA JUPITER

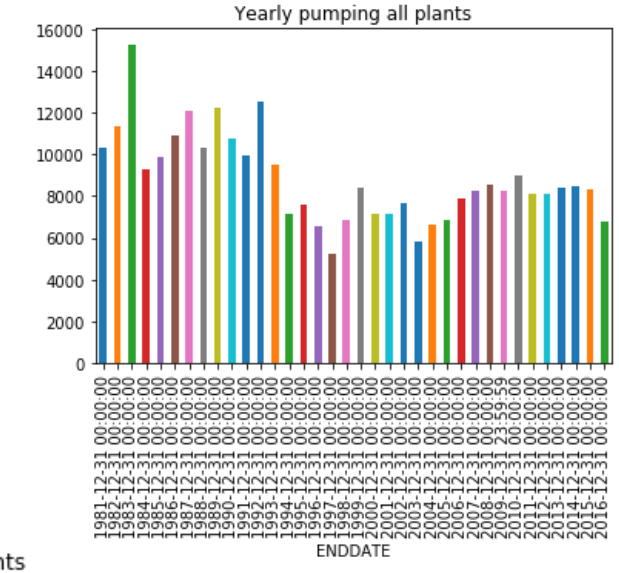
Define inputs for extracting information:

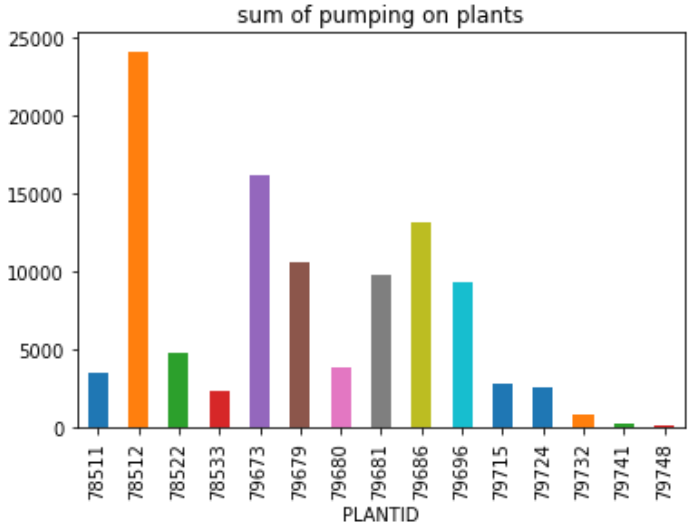
NB! format is hardwired to:

length: meters

time: days

In [2]:
```
# name of database. Must be Jupiter database en access 2000 format or newer:
data_base = 'z:\\Projects\\2017_rOPEN\\WP4\\jupiter\\db\\Jupiter_javngyde_xl_1.mdb'
# Base name for the outputs written to the disk:
area_name = 'Javngyde'
# Area outline: [xmin, xmax, ymin, ymax]
coordinates = [547490,554205,6218120,6222100]
# folder where csv of pandas dataframes are stored:
datadir = os.path.join('data')
# folder where shape files are stored:
shpdir =  os.path.join('shp')
# Length of each data period: 'AS' -> Anually start of year
period_length = 'AS'
# start of groundwater model simulation period:
start = '1993-01-01'
# end of groundwater model simulation period
end = '2005-01-01'

# check if directories exist. If not create them:
if not os.path.exists(datadir):
    os.makedirs(datadir)
if not os.path.exists(shpdir):
    os.makedirs(shpdir)
```





DATA MINING FRA JUPITER

Define inputs for extracting information:

NB! format is hardwired to:

length: meters

time: days

In [2]:
```
# name of database. Must be Jupiter database en access 2000 format or newer:
data_base = 'z:\\Projects\\2017_rOPEN\\WP4\\jupiter\\db\\Jupiter_javngyde_xl_1.mdb'
# Base name for the outputs written to the disk:
area_name = 'Javngyde'
# Area outline: [xmin, xmax, ymin, ymax]
coordinates = [547490,554205,6218120,6222100]
# folder where csv of pandas dataframes are stored:
datadir = os.path.join('data')
# folder where shape files are stored:
shpdir =  os.path.join('shp')
# Length of each data period: 'AS' -> Anually start of year
period_length = 'AS'
# start of groundwater model simulation period:
start = '1993-01-01'
# end of groundwater model simulation period
end = '2005-01-01'

# check if directories exist. If not create them:
if not os.path.exists(datadir):
    os.makedirs(datadir)
if not os.path.exists(shpdir):
    os.makedirs(shpdir)
```

Pumperate tilskrevet boringer

Pumping on plant 78511

	AMOUNT	BoreholeRate
1993-01-01	5054.0	6.923288
1994-01-01	3051.0	4.179452
1995-01-01	8639.0	11.834247
1996-01-01	3280.0	4.493151
1997-01-01	1946.0	2.665753
1998-01-01	2621.0	3.590411
1999-01-01	0.0	0.000000
2000-01-01	4091.0	5.604110
2001-01-01	2148.0	2.942466
2002-01-01	2148.0	2.942466
2003-01-01	2102.0	2.879452
2004-01-01	2246.0	3.076712
2005-01-01	2501.0	3.426027

****

Pumping on plant 78512

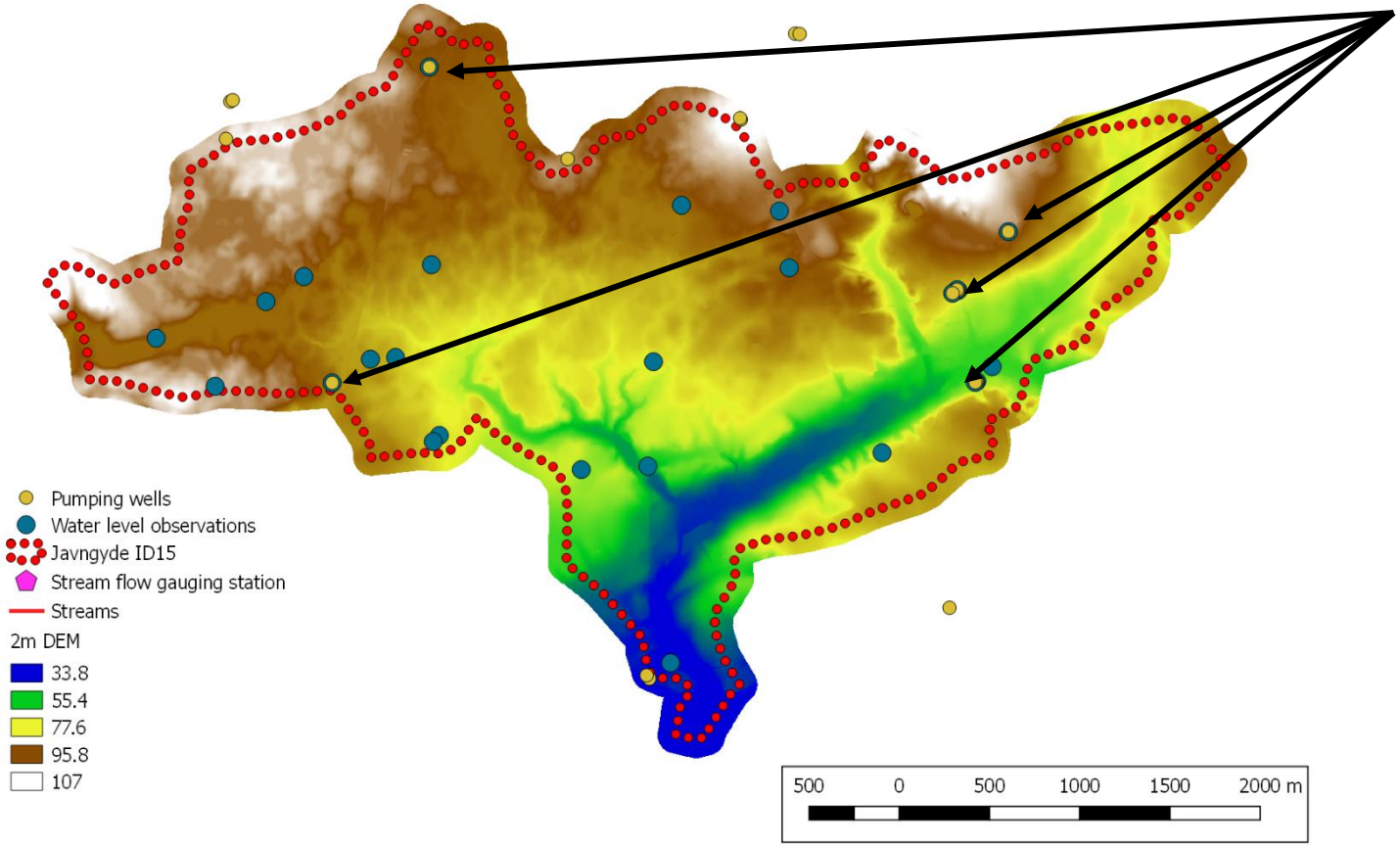
	AMOUNT	BoreholeRate
1993-01-01	25710.0	23.479452
1994-01-01	25940.0	23.689498
1995-01-01	18395.0	16.799087
1996-01-01	18035.0	16.470320
1997-01-01	19090.0	17.433790
1998-01-01	19700.0	17.990868
1999-01-01	21420.0	19.561644
2000-01-01	25225.0	23.036530
2001-01-01	25292.0	23.097717
2002-01-01	26173.0	23.902283
2003-01-01	26305.0	24.022831
2004-01-01	25805.0	23.566210
2005-01-01	24357.0	22.243836

****

Pumping on plant 78522

	AMOUNT	BoreholeRate
1993-01-01	800.0	1.095890

DATA MINING FRA JUPITER



Boringer inden for opland

Pumping on plant 78511

	AMOUNT	BoreholeRate
1993-01-01	5054.0	6.923288
1994-01-01	3051.0	4.179452
1995-01-01	8639.0	11.834247
1996-01-01	3280.0	4.493151
1997-01-01	1946.0	2.665753
1998-01-01	2621.0	3.590411
1999-01-01	0.0	0.000000
2000-01-01	4091.0	5.604110
2001-01-01	2148.0	2.942466
2002-01-01	2148.0	2.942466
2003-01-01	2102.0	2.879452
2004-01-01	2246.0	3.076712
2005-01-01	2501.0	3.426027

****

Pumping on plant 78512

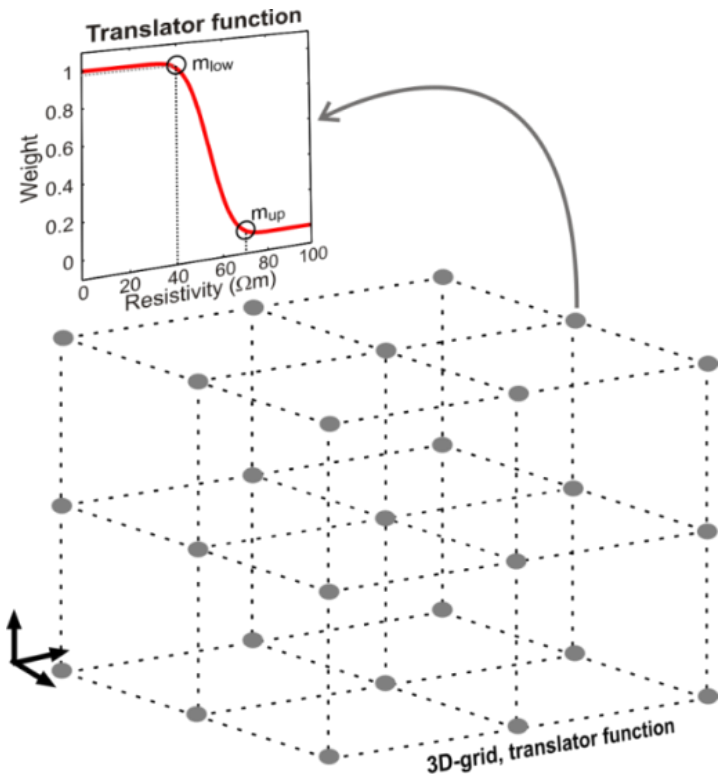
	AMOUNT	BoreholeRate
1993-01-01	25710.0	23.479452
1994-01-01	25940.0	23.689498
1995-01-01	18395.0	16.799087
1996-01-01	18035.0	16.470320
1997-01-01	19090.0	17.433790
1998-01-01	19700.0	17.990868
1999-01-01	21420.0	19.561644
2000-01-01	25225.0	23.036530
2001-01-01	25292.0	23.097717
2002-01-01	26173.0	23.902283
2003-01-01	26305.0	24.022831
2004-01-01	25805.0	23.566210
2005-01-01	24357.0	22.243836

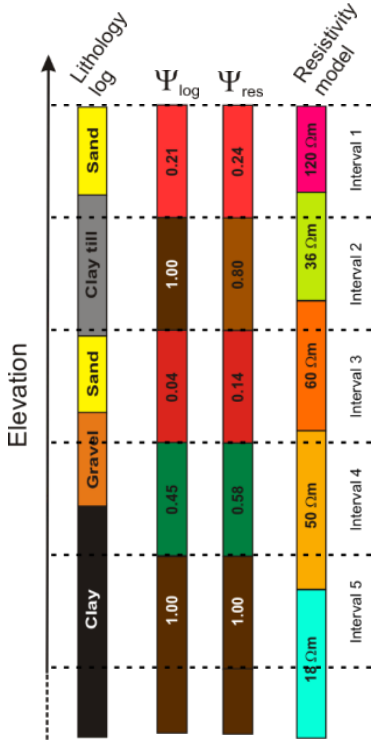
****

Pumping on plant 78522

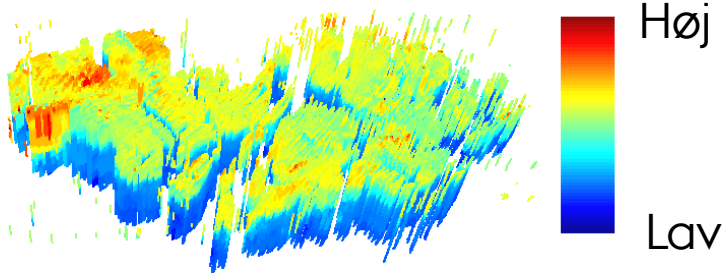
	AMOUNT	BoreholeRate
1993-01-01	800.0	1.095890

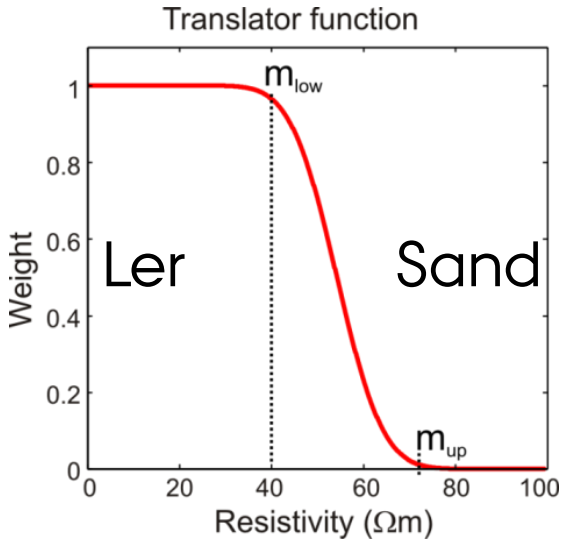
GENERERING AF MODEL STRUKTUR





Resistivitet modeller

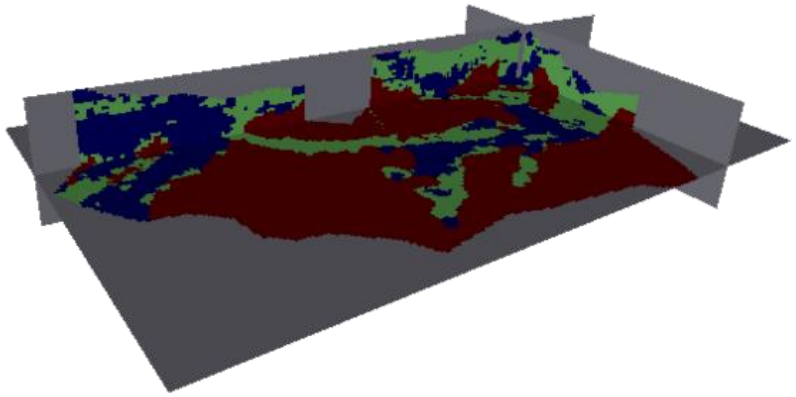




Foged et al., Hydrol. Earth Syst. Sci., 18, 4349–4362, 2014

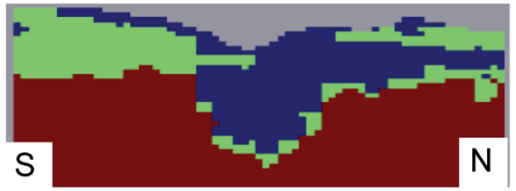
GENERERING AF MODEL STRUKTUR

IK mode model

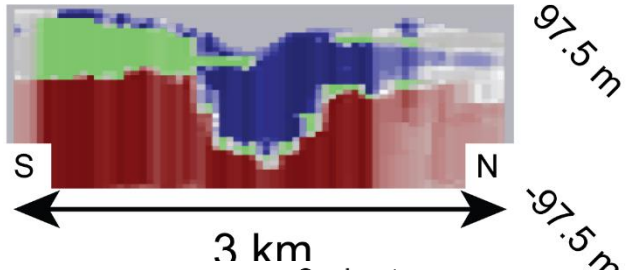


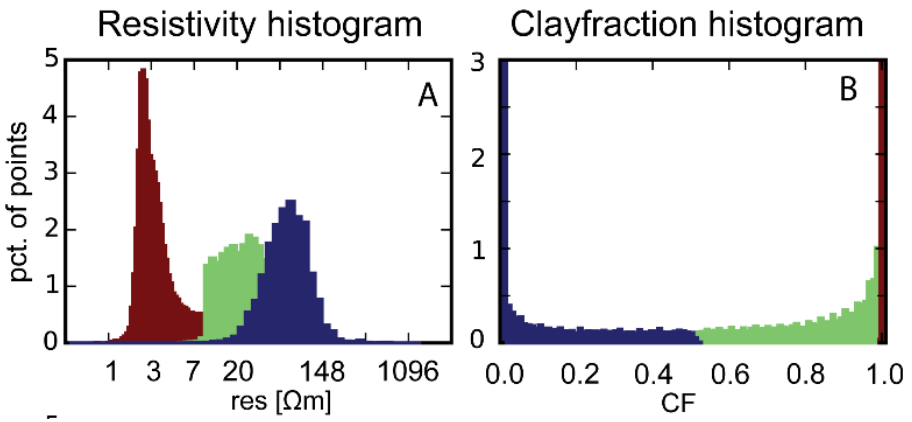
Cross section Ristrup

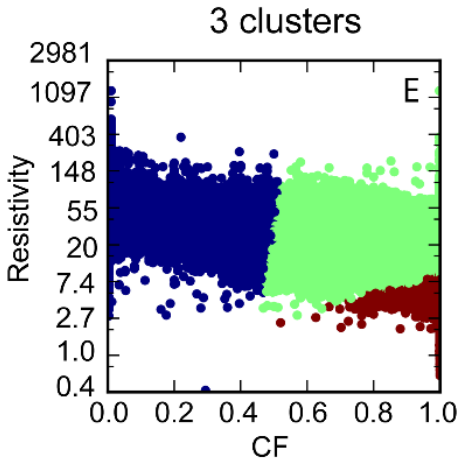
3 cluster mode model **A**



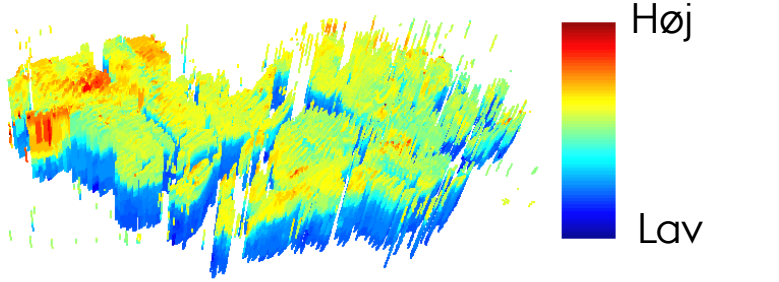
3 cluster model with unc. **B**



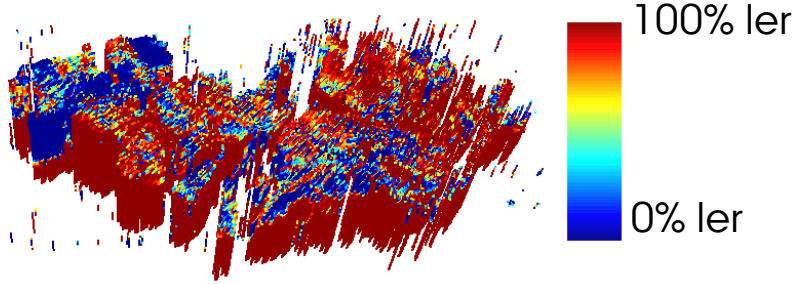




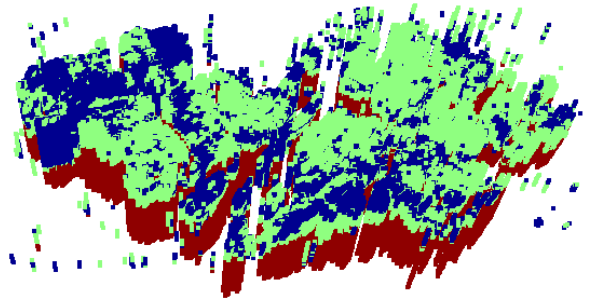
Resistivitet modeller



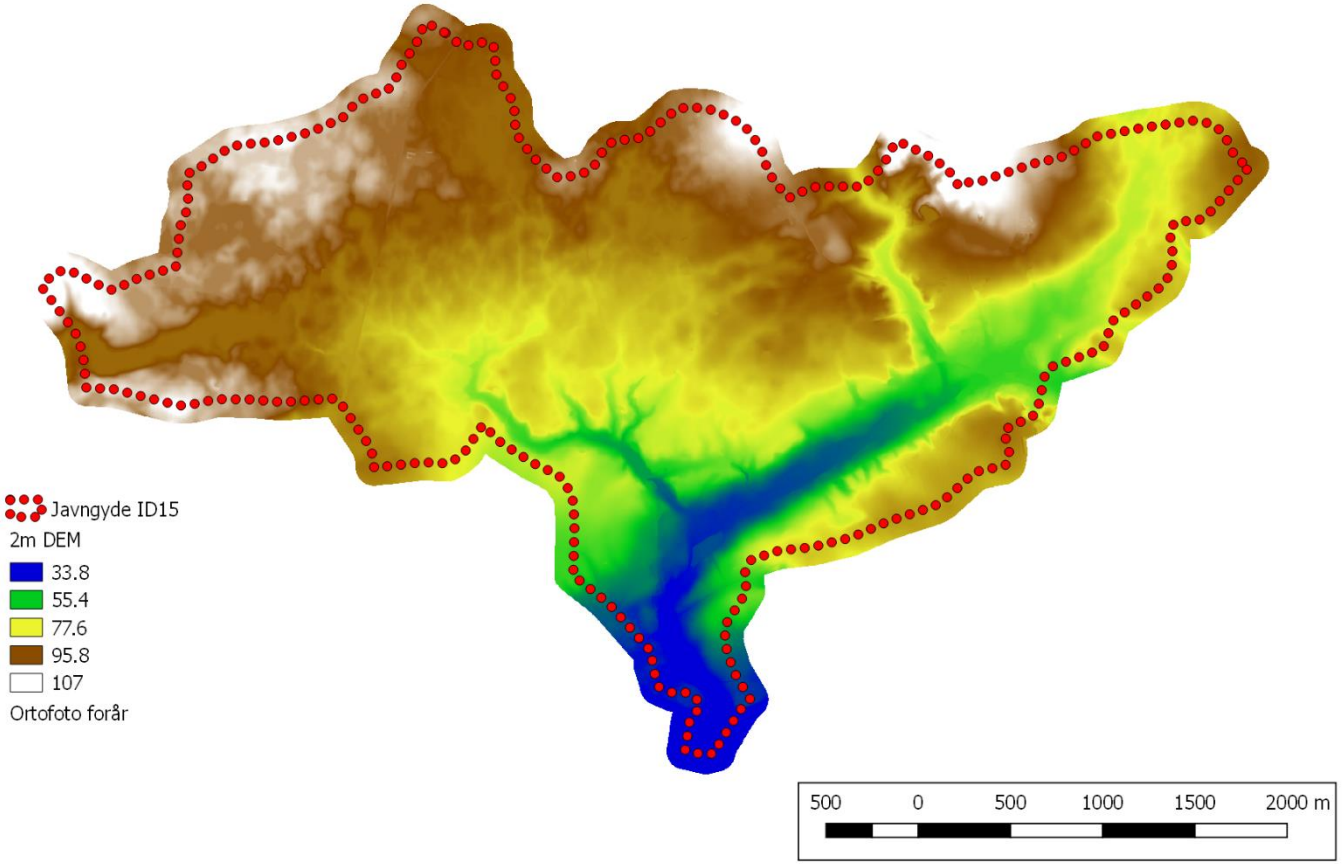
Lerfraktion

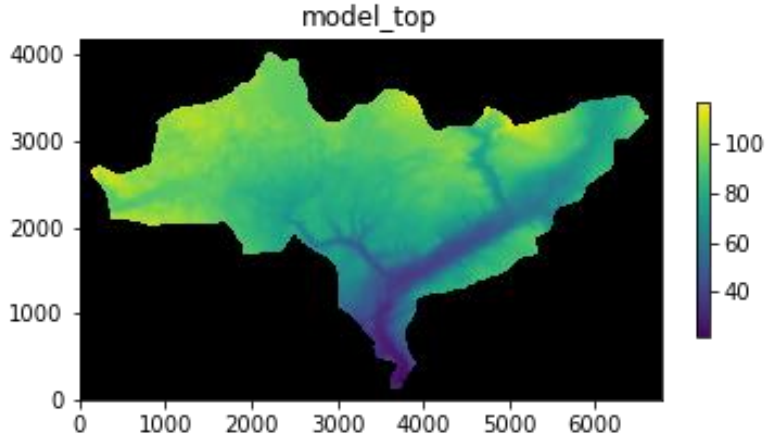


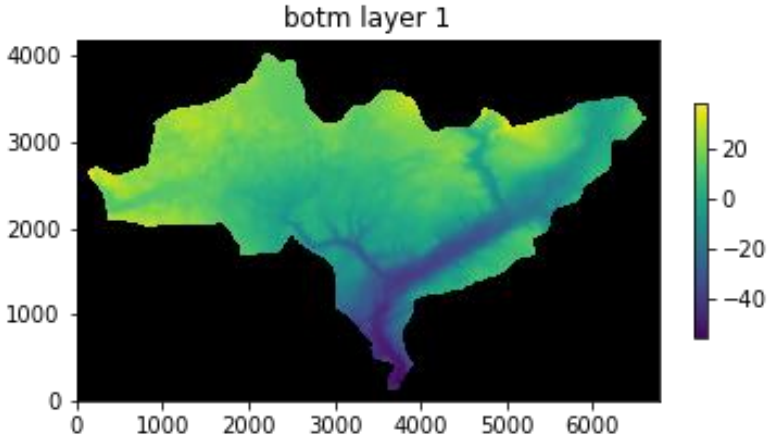
K-means clustering



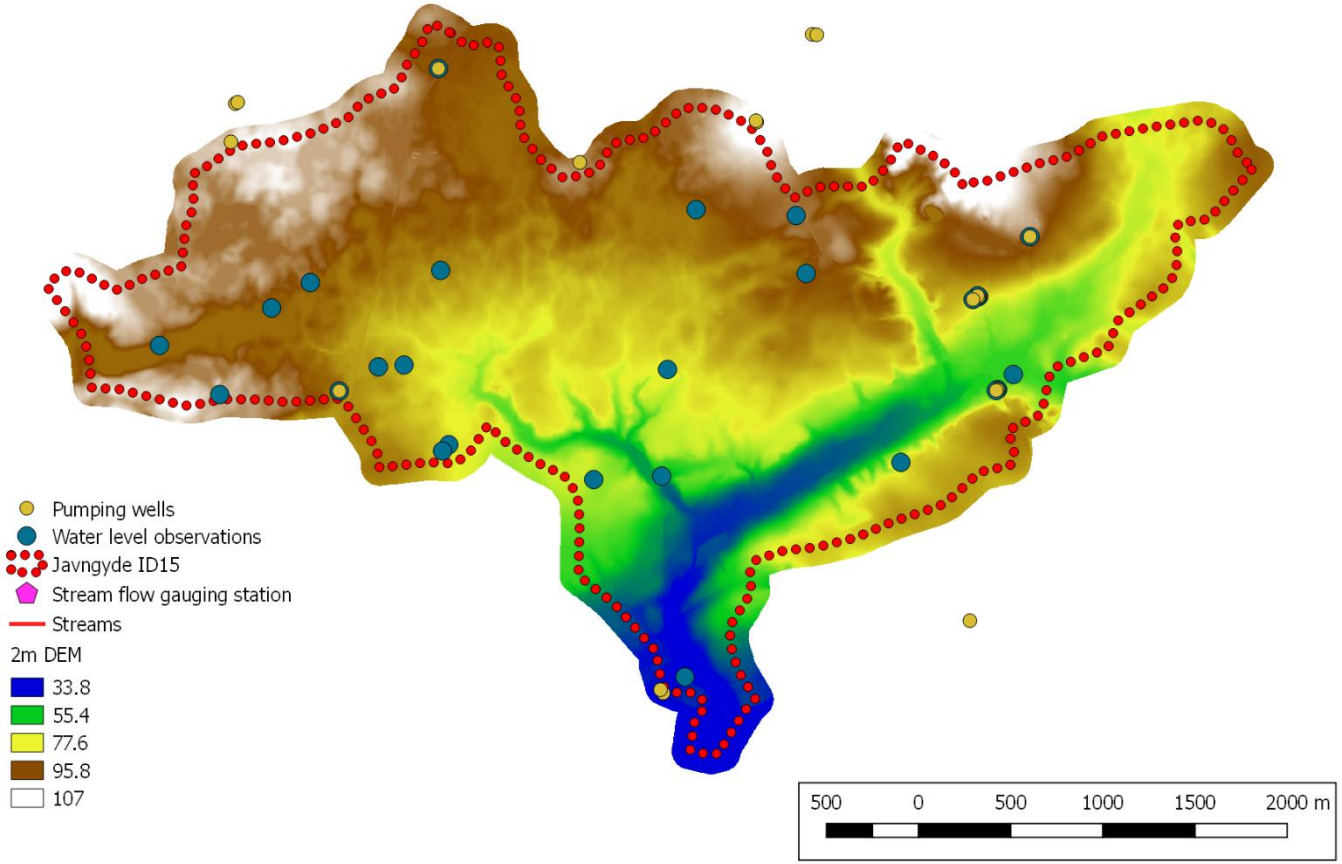
PROJECTION OF DATA ONTO MODEL





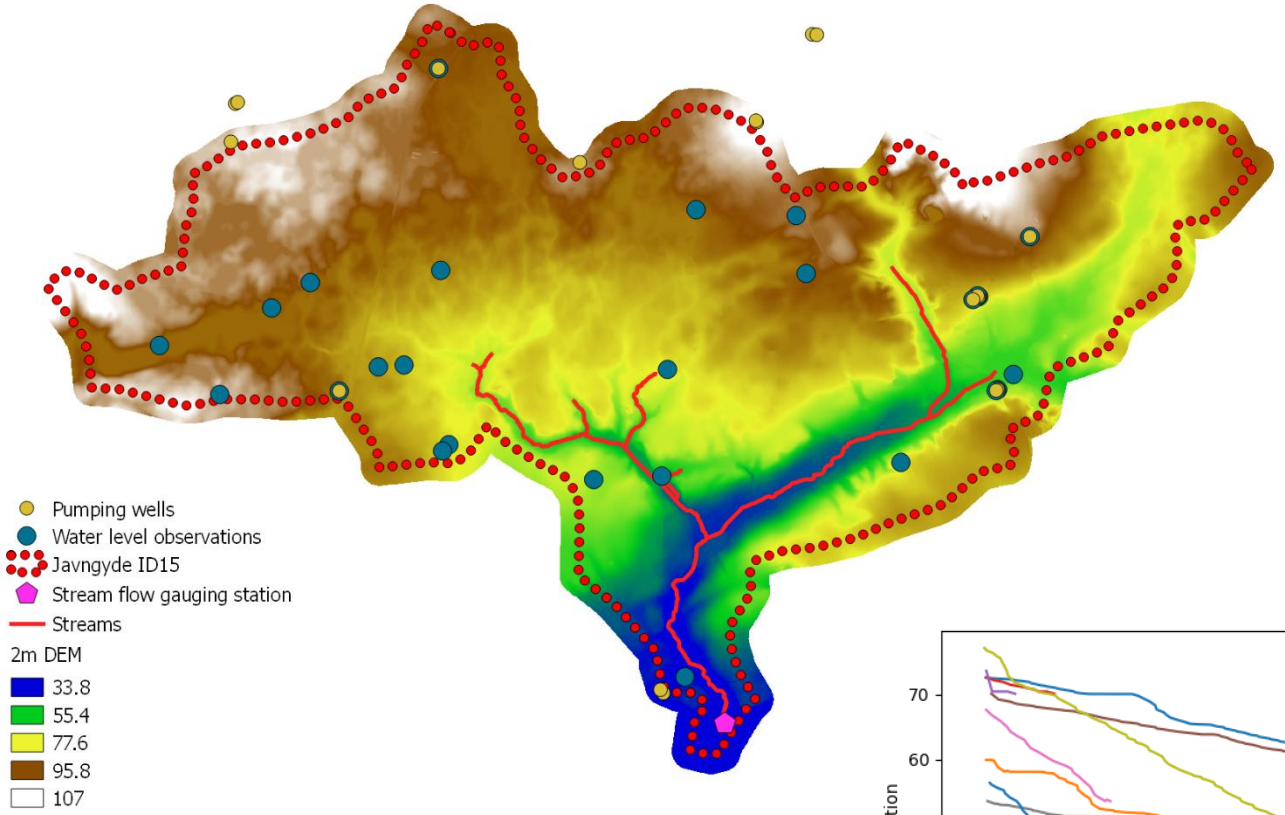


PROJECTION OF DATA ONTO MODEL

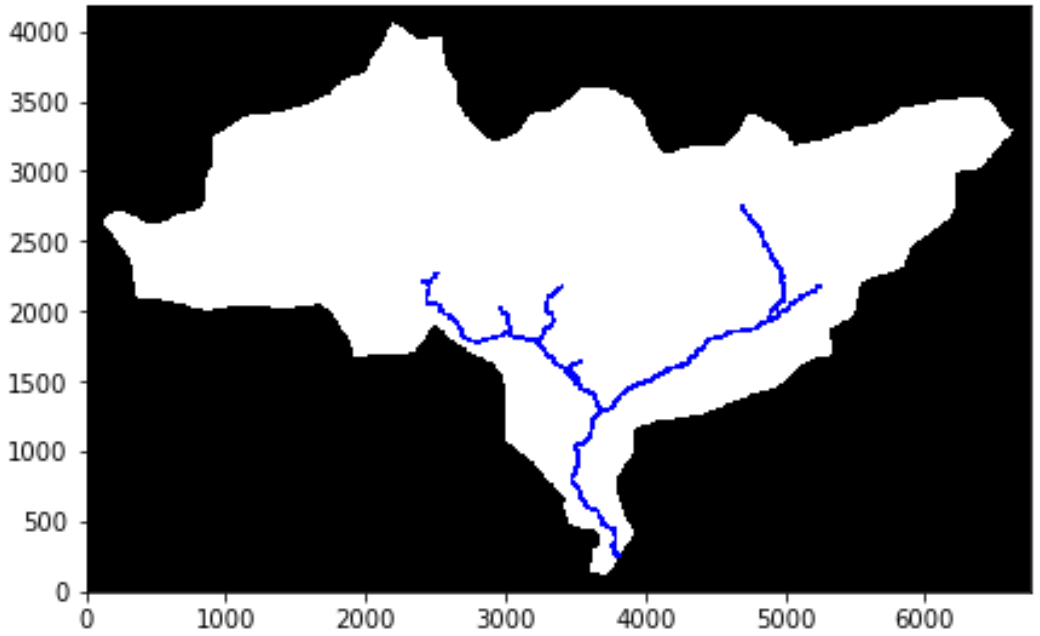


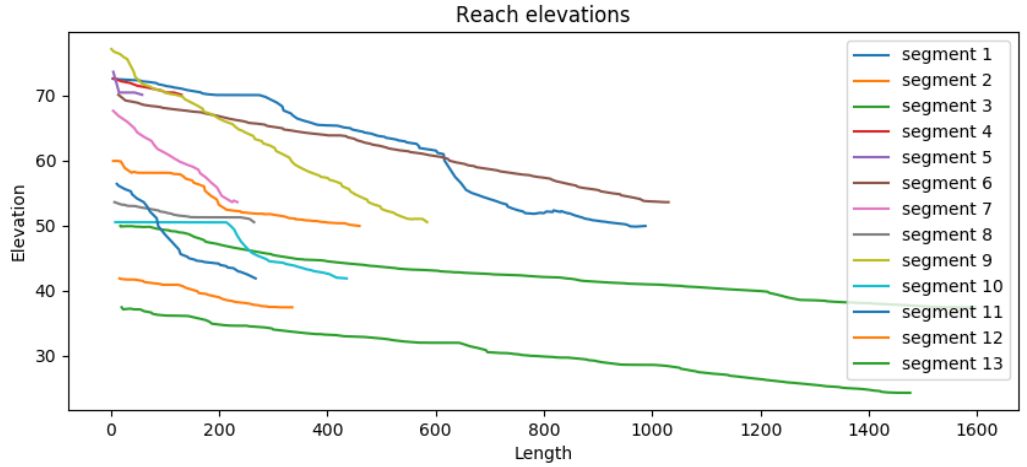
Indvindingsboringer

PROJECTION OF DATA ONTO MODEL

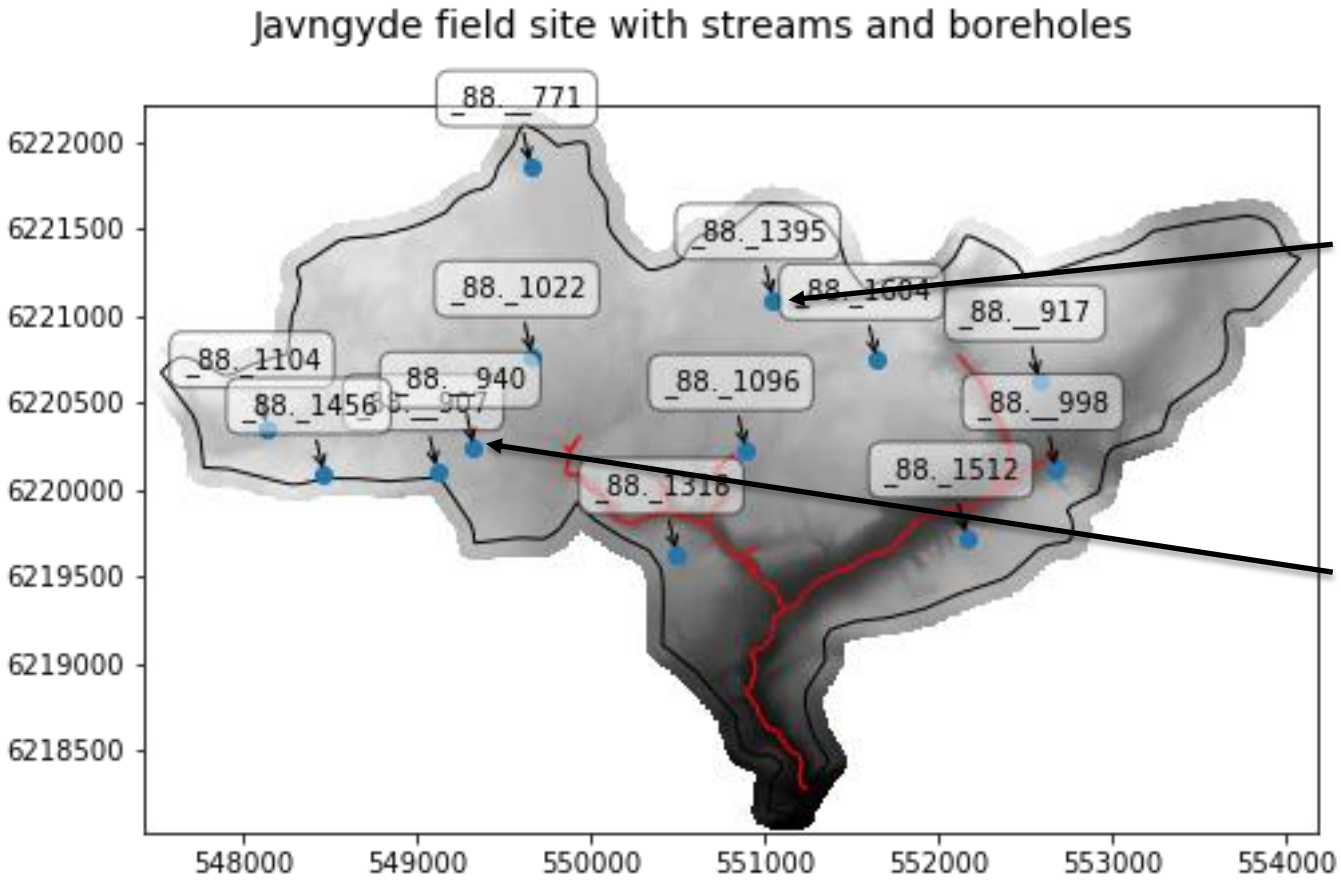


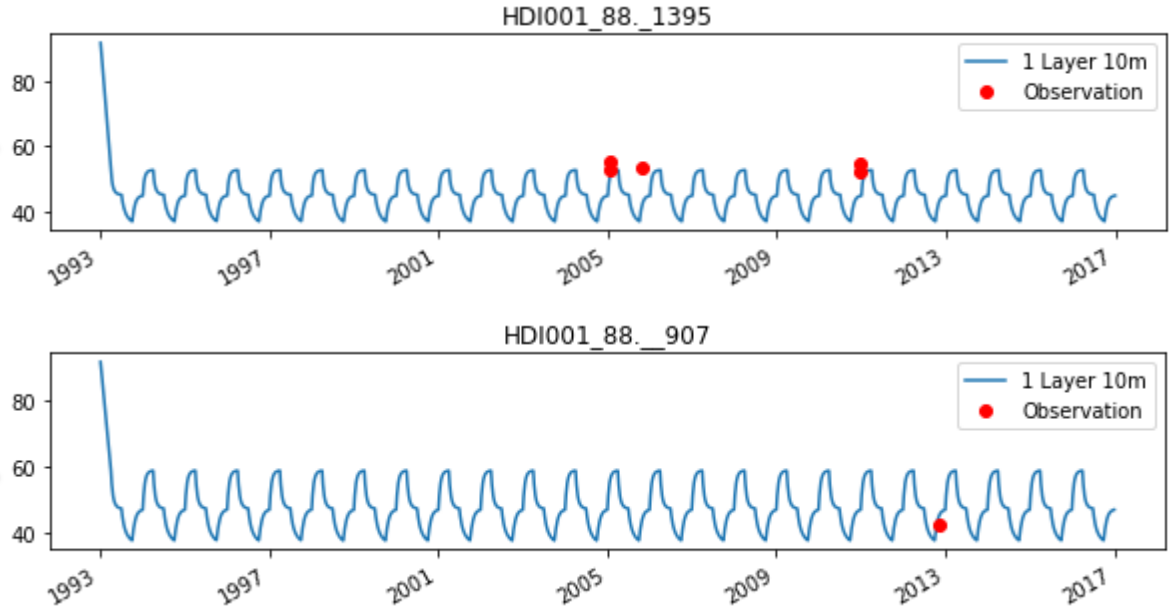
Vandløb





OUTPUT EXAMPLES





Eksemplet er genereret med en triviel nedbørstidsserie, og uden struktur i undergrunden

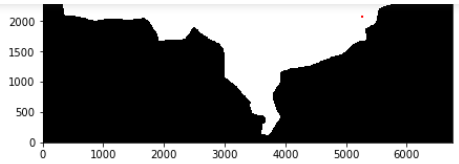
HVORDAN SER DET UD I PRAKSIS?

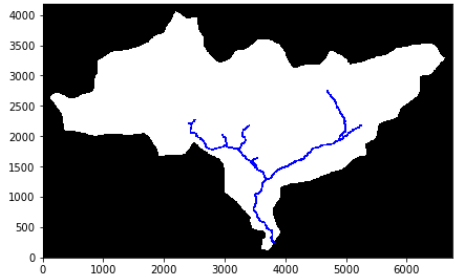
jupyter javn_w_pump_tr2_1L_10m Last Checkpoint: Yesterday at 8:40 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code





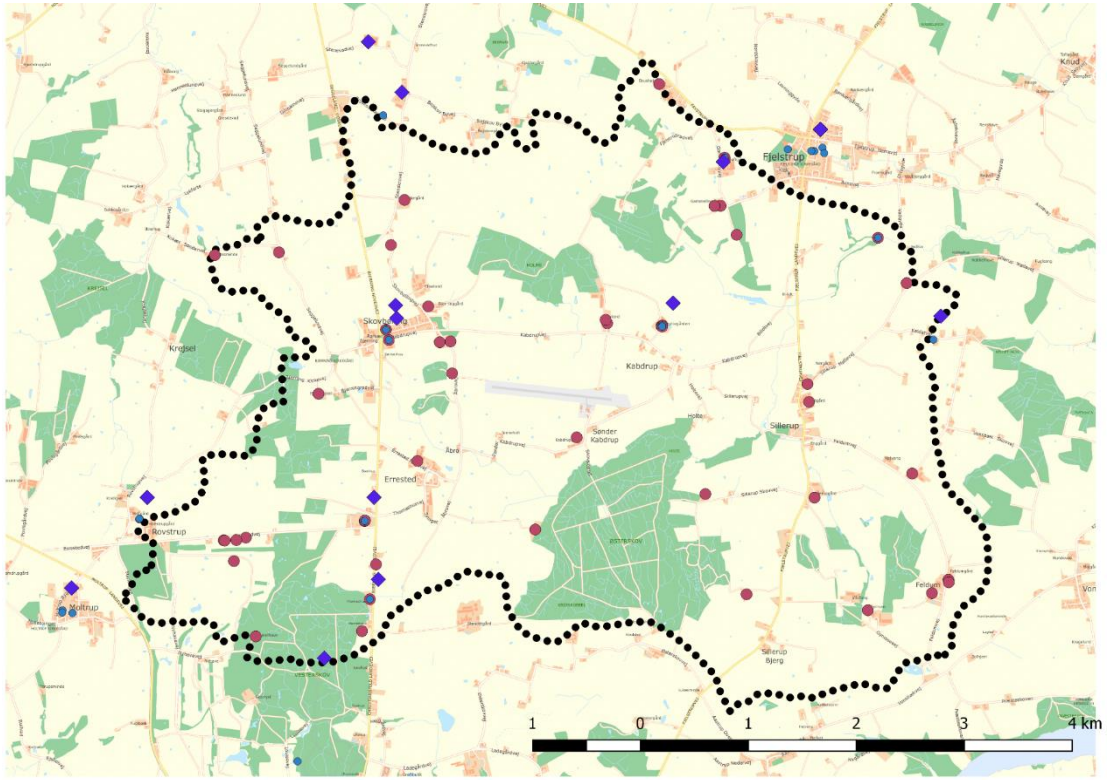
In [81]:
```
mf.run_model()
```

```
FloPy is using the following executable to run the model: C:\Program Files\GWV6\mf2005.exe

                                MODFLOW-2005
    U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER FLOW MODEL
                             Version 1.11.00 8/8/2013

 Using NAME file: javn1.nam
 Run start date and time (yyyy/mm/dd hh:mm:ss): 2017/11/23  8:43:52

 Solving:  Stress period:     1    Time step:     1    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     1    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     2    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     3    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     4    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     5    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     6    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     7    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     8    Ground-Water Flow Eqn.
 Solving:  Stress period:     2    Time step:     9    Ground-Water Flow Eqn.
```

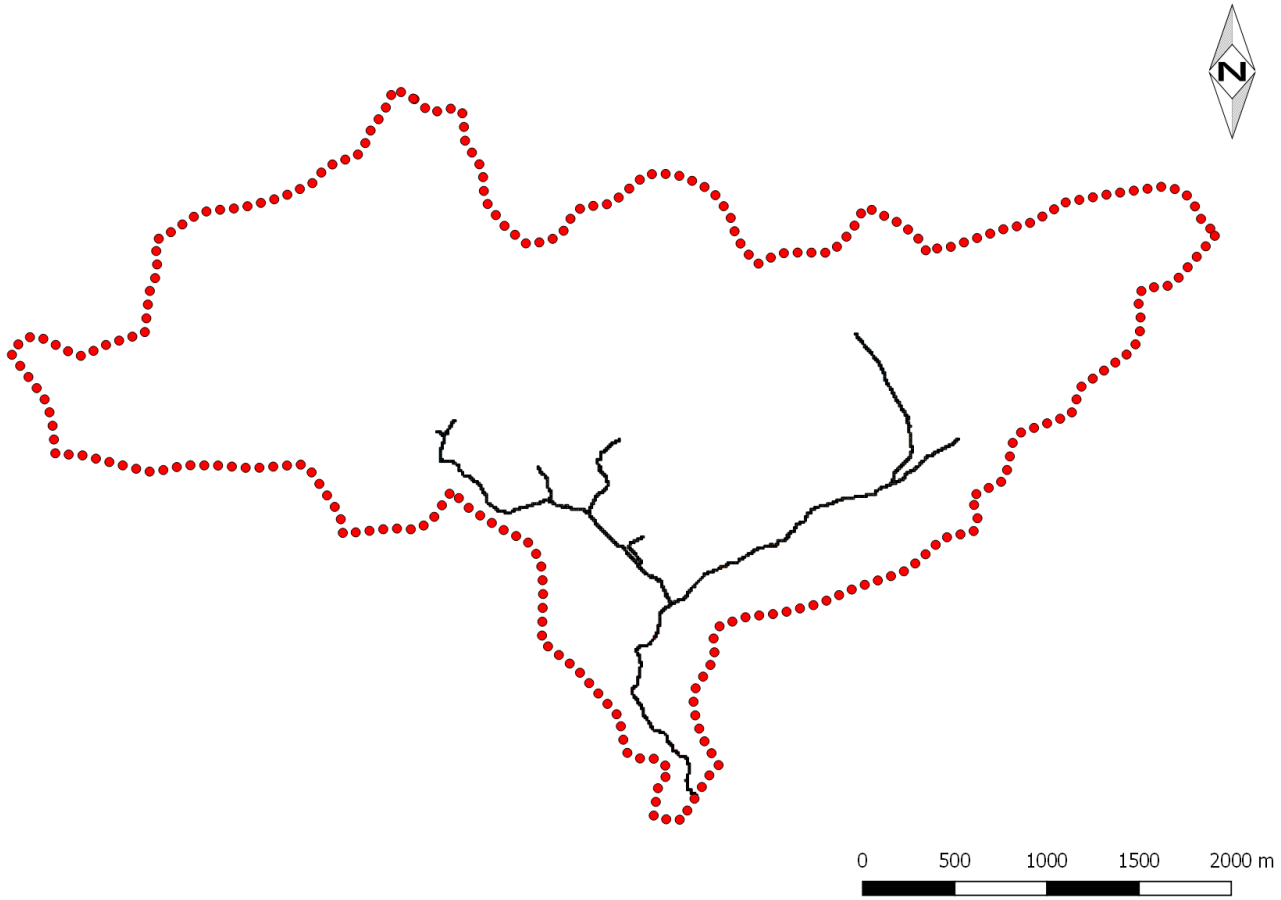
In [82]:
```
if OCflg:
    hdsnam = os.path.join(ws, modelname+'.hds')
    hdobj = flopy.utils.HeadFile(hdsnam)
    head = hdobj.get_data()
    levels = np.arange(25, 65, 1)

    fig = plt.figure(figsize=(7, 7))

    ax = fig.add_subplot(1, 1, 1, aspect='equal')
```

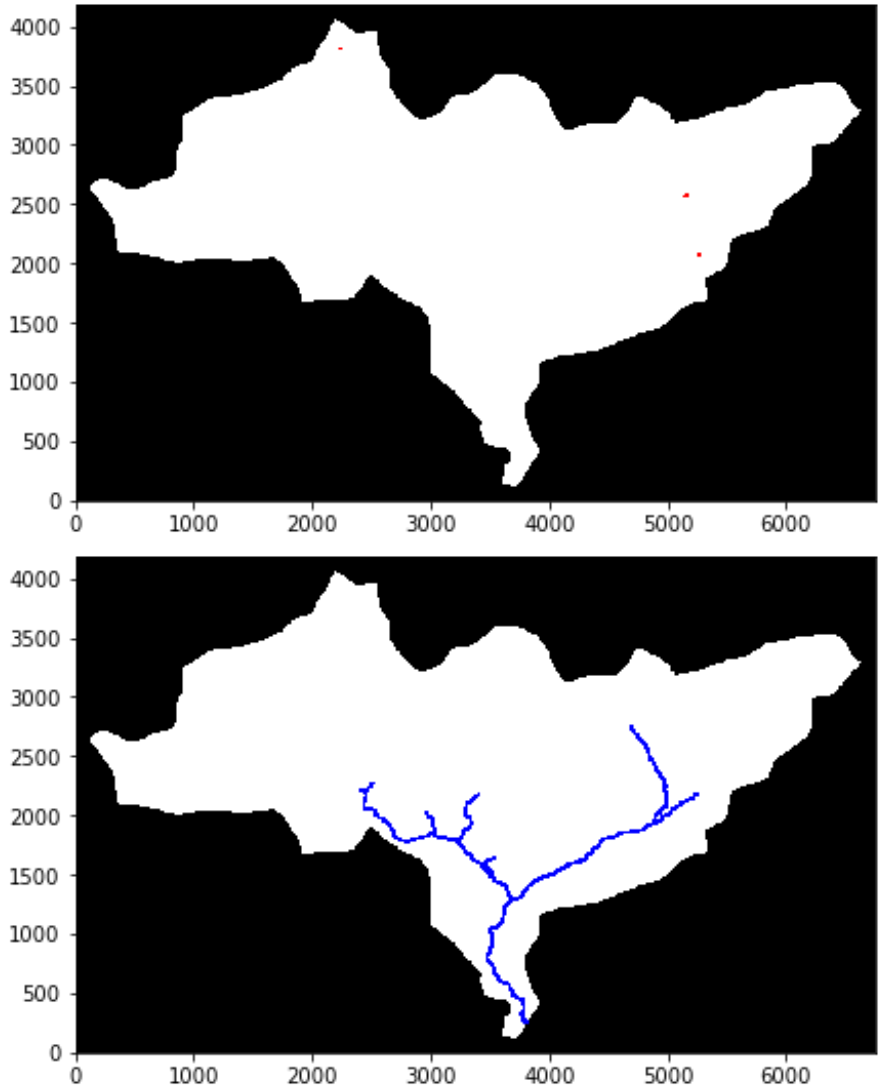
Setting up and running groundwater model

Med relativt få ændringer kan modellen genereres for et alternativt opland

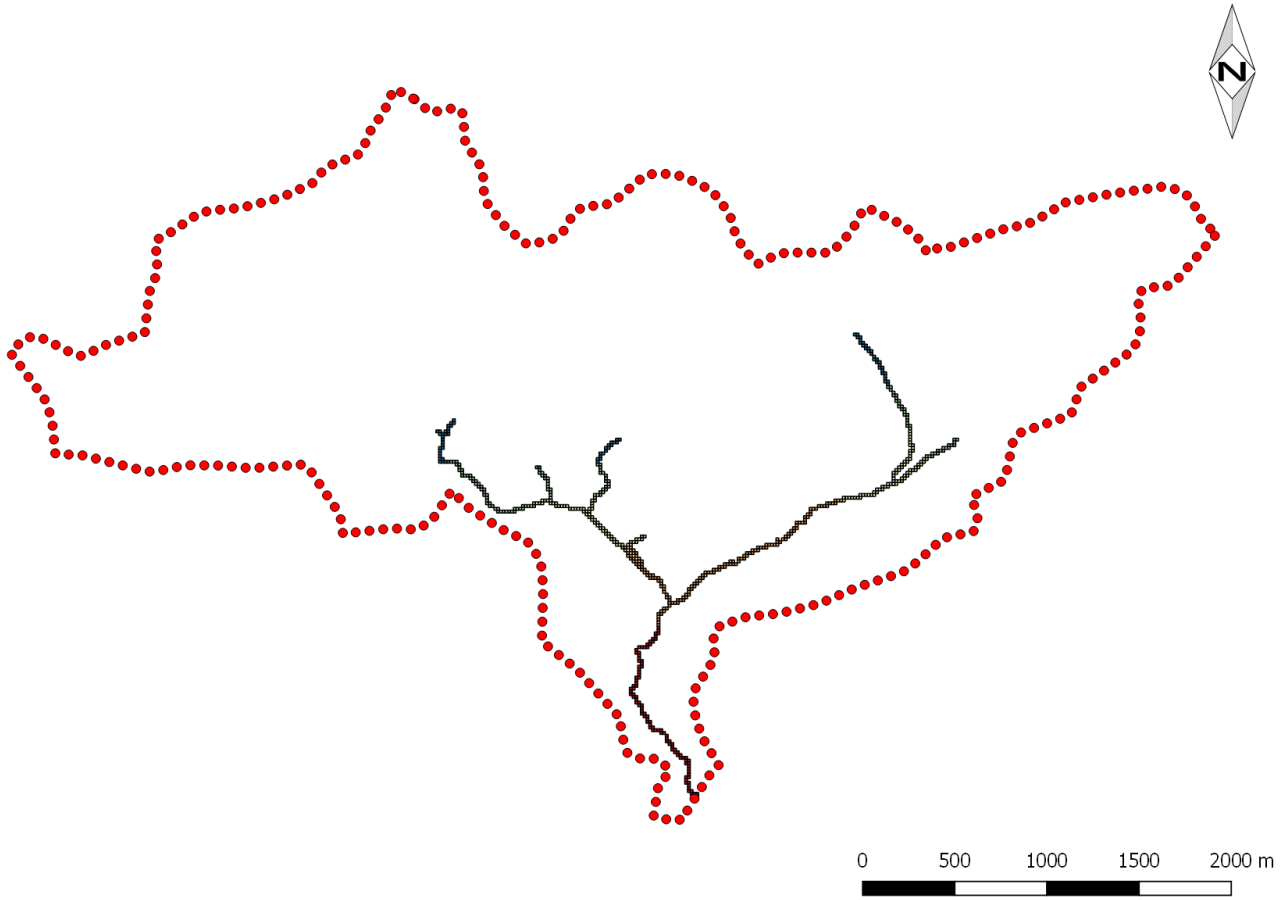


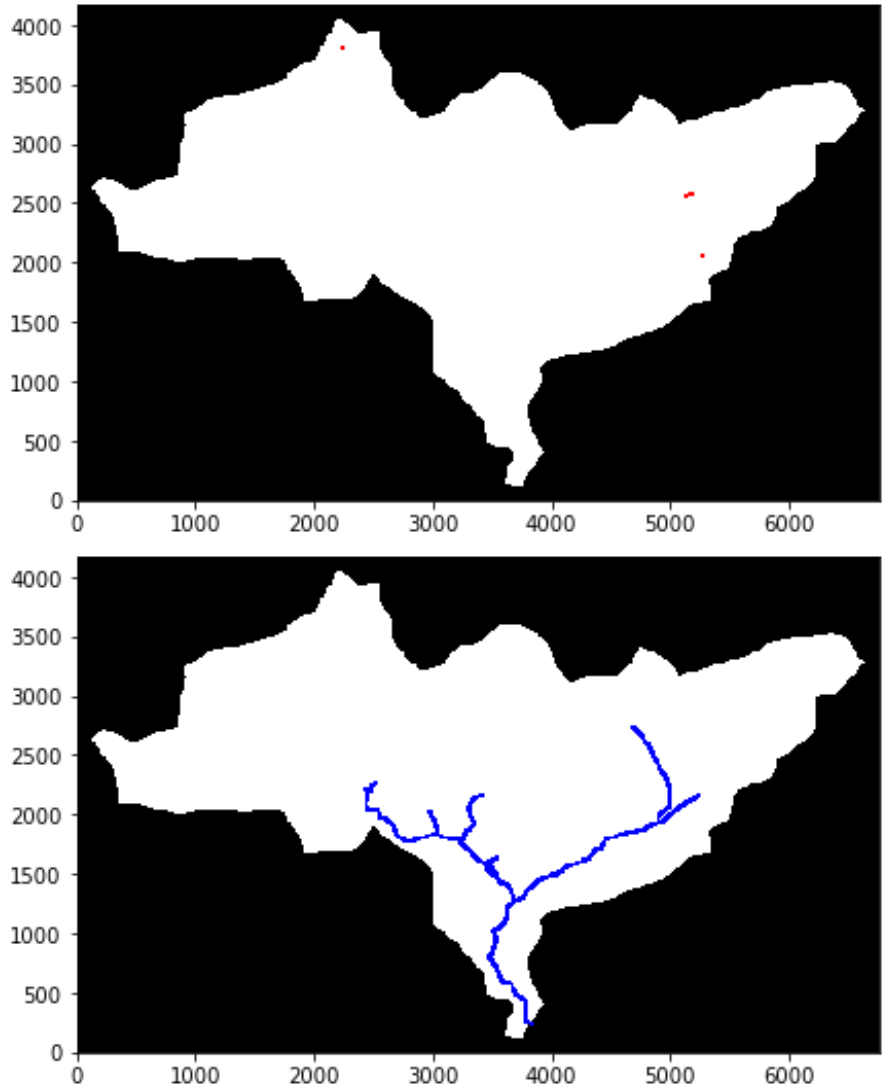
...ELLER MED EN ANDEN GRIDOPLØSNING



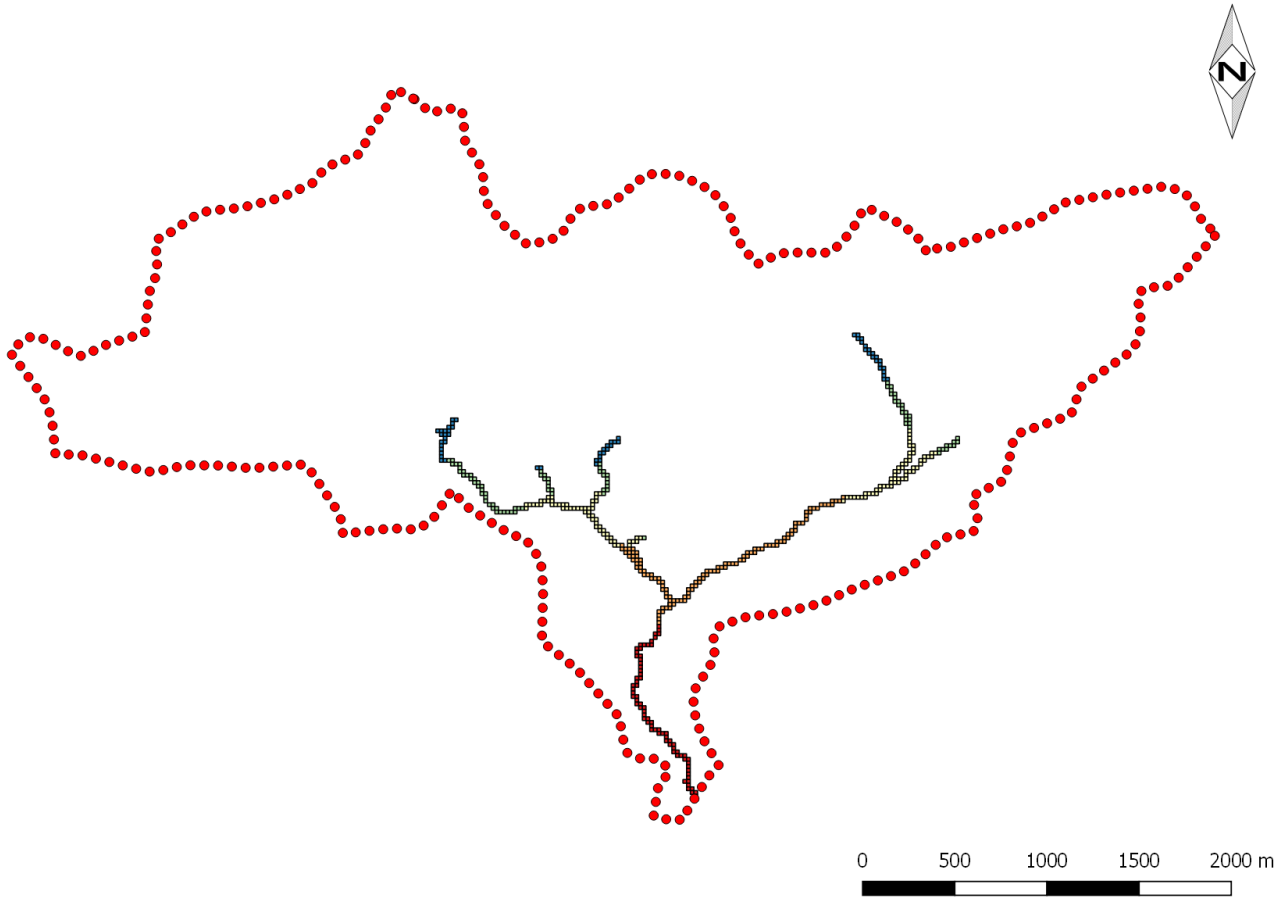


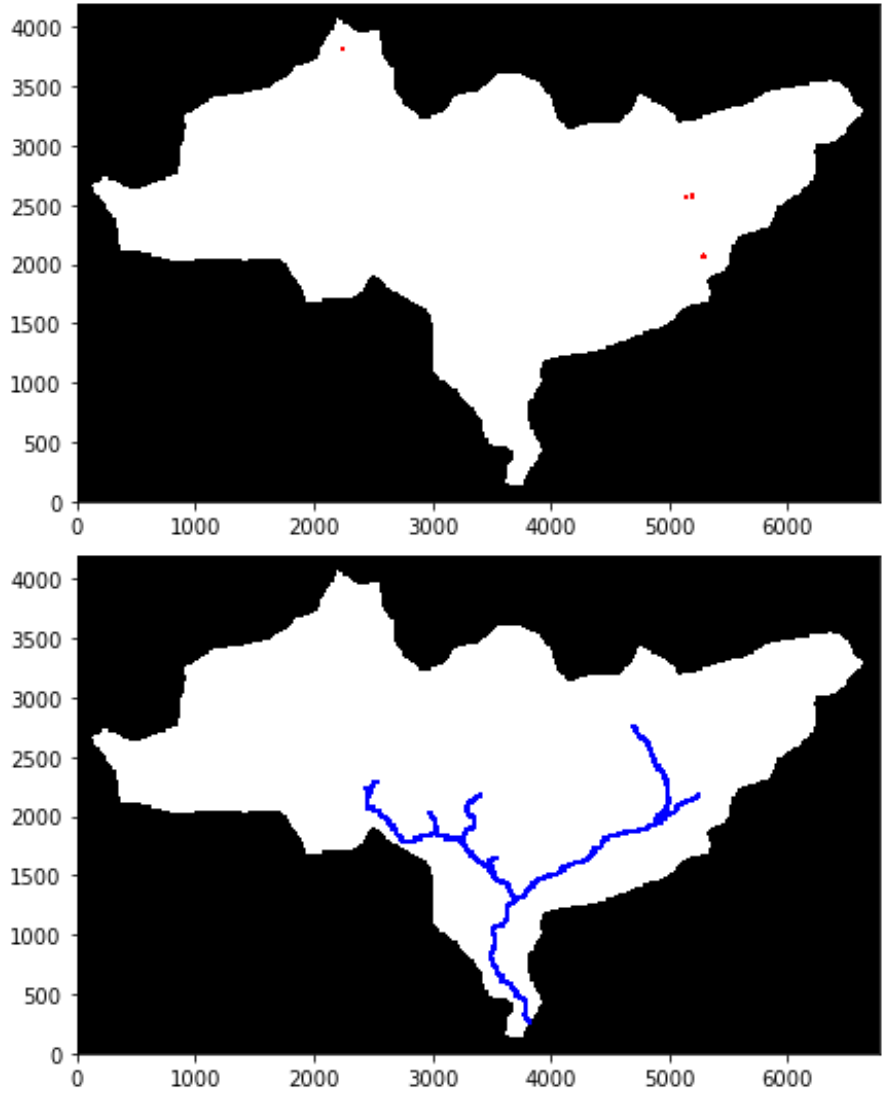
...ELLER MED EN ANDEN GRIDOPLØSNING



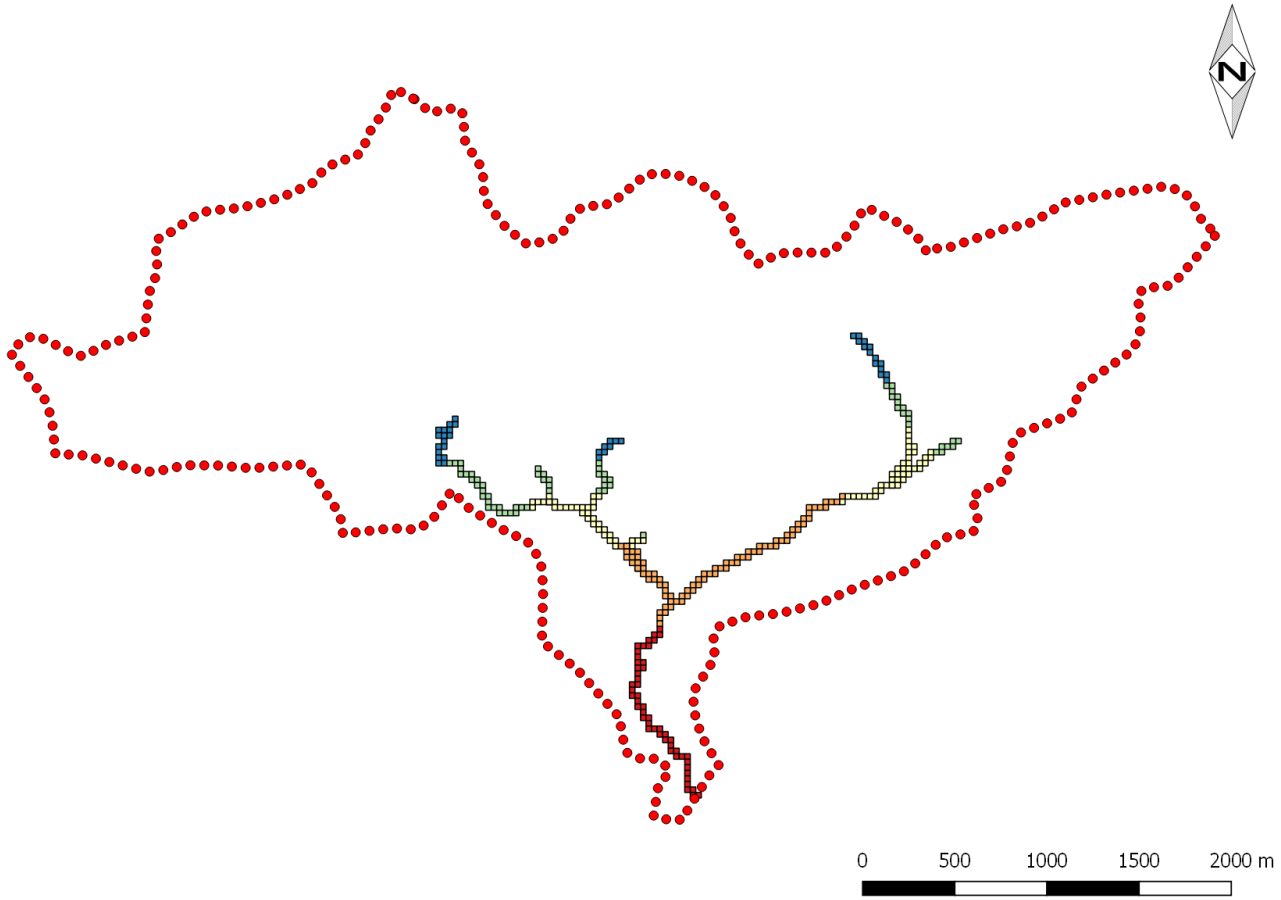


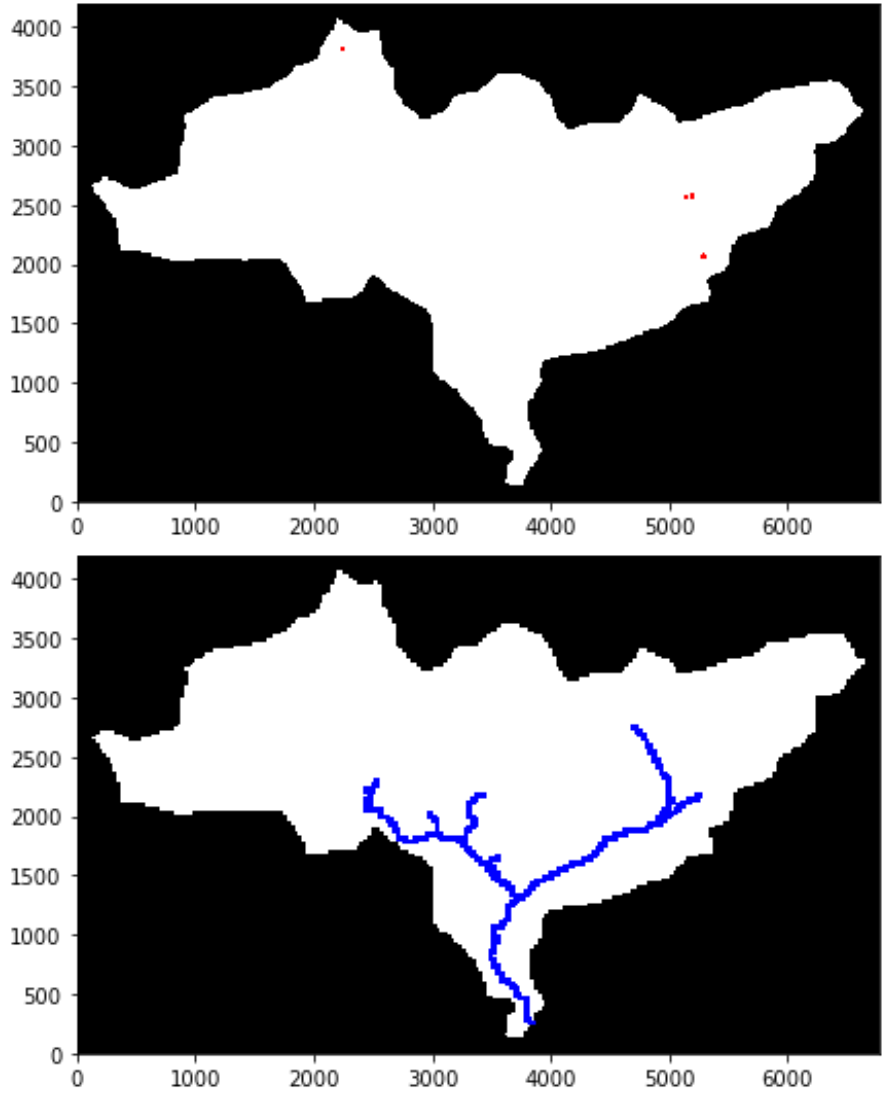
...ELLER MED EN ANDEN GRIDOPLØSNING





...ELLER MED EN ANDEN GRIDOPLØSNING





KONKLUSIONER OG PERSPEKTIVER

- Hele workflowet er implementeret I python
- Det vil være Open source når rOpen er færdiggjort
- Model opbygningen er hurtig og data-dreven
- Det skal være fleksibelt med hensyn til analysetyper
- Det skal være simpelt at opdatere med ny data indsamling

Innovation Fund Denmark