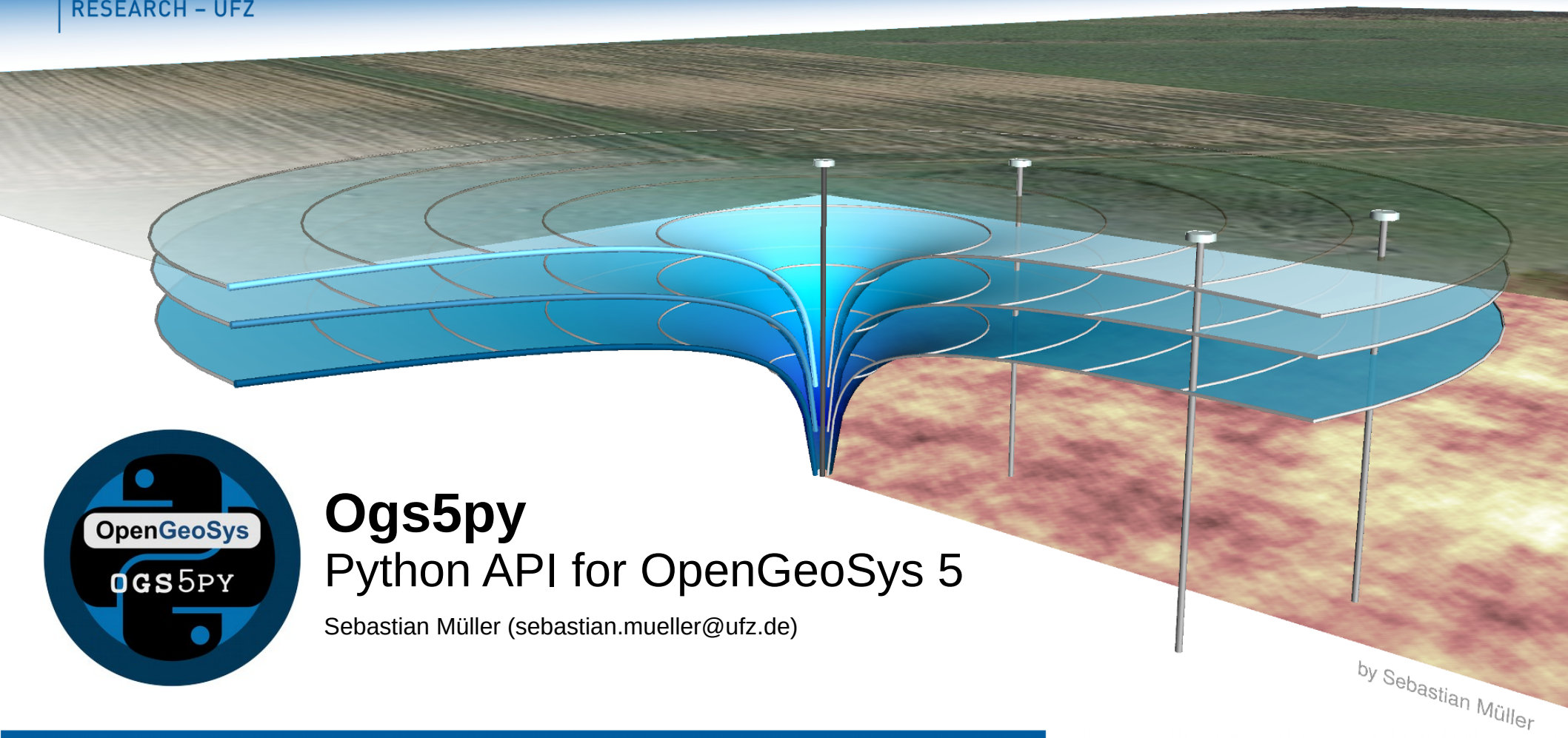


Ogs5py

Python API for OpenGeoSys 5

Sebastian Müller (sebastian.mueller@ufz.de)

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

GeoStat Framework

GeoStat Framework

https://geostat-framework.org/

Suchen

GEOSTAT

GeoStat Framework

Python framework for geostatistical simulations

Why an API?

- **Data from different sources**
  - Boundary Conditions
  - Meshes/Geometries
  - Conductivity Fields
- **Dynamical model setup**
  - Single script
  - Ensemble runs
  - Modify existing models
- **Post-processsing**
  - Access output with Python
  - Reader for VTK/TecPlot
  - Sorted by PCS/primary-var
- **Why Python?**
  - Easy to use
  - Open source
  - Fully flexible language

How to get ogs5py?

- **Github:**
  - https://github.com/GeoStat-Framework/ogs5py
  - Documentation in the making (badum-tss)
- **Install via pip:**
  - `pip install https://github.com/GeoStat-Framework/ogs5py/archive/master.zip`
  - PyPI release planned
- **Works on:**
  - Python 2 + 3
  - Unix + Windows

OGS5 input files

- **Block files** (main case, data handled block-wise)
  - ```
    #<main_key>   or   #<main_key>
     $<sub_key>          <content>
      <content>          …
      …                  …
    #STOP                #STOP
    ```
    } one block
  - BC, CCT, DDC, FCT, GEM, IC, KRC, MCP, MFP, MMP, MPD, MSP, NUM, OUT, PCS, REI, RFD, ST, TIM
- **Line-wise files** (additional files handled as list of lines)
  - ASC (from TIM, PCS, GEM …), PQC + phreeqc.dat, GEMS3K files (dch, ipm, dbr)
- **Special files**
  - GLI, MSH (extended functionality provided)
  - PCT, RFR (other handling)
    ```

Workflow

```python
from ogs5py import OGS              # base class for the model
from ogs5py.reader import readpvd   # reader for PVD output
model = OGS(
    task_root="test_folder",        # folder for input files
    task_id="model",                # model name
    output_dir="output",            # output directory
)

### setup input ######################################

model.write_input()                 # write the input files
success = model.run_model()         # run ogs (return state)

out = readpvd(                      # get a dictionary
    task_root="output",             # with all pvd output
    task_id="model",                # sorted by process
    pcs="ALL",
)
```

- **OGS class attributes**
  - Files: `bc, cct, ddc, fct, gem, gli, ic, krc, mcp, mfp, mmp, msh, msp, num, out, pcs, pct, pqc, pqcdat, rei, rfd, st, tim`
  - Lists: `mpd, gli_ext, rfr, gem_init, asc, copy_files`
  - Strings: `task_root, task_id, output_dir, top_com, bot_com`
- **OGS class methods**
  - `add_/del_[mpd, gli_ext, …]()`
  - `load_model()`
  - `reset()`
  - `run_model()`
  - `write_input()`

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Modify block files

```python
from ogs5py import OGS

model = OGS(
    task_root="test_folder",
    task_id="model",
    output_dir="output",
)

model.ic.add_block(
    PCS_TYPE="GROUNDWATER_FLOW",
    PRIMARY_VARIABLE="HEAD",
    GEO_TYPE="DOMAIN",
    DIS_TYPE=["CONSTANT", 0.0],
)
```

```
|-------------- Written with ogs5py --------------|
#INITIAL_CONDITION
  $PCS_TYPE
   GROUNDWATER_FLOW
  $PRIMARY_VARIABLE
   HEAD
  $DIS_TYPE
   CONSTANT 0.0
  $GEO_TYPE
   DOMAIN
#STOP
|-- Written with ogs5py (0.5.0rc1) on: 2019-01-10_14-45-46 --|
```

- **BlockFile class attributes**
  - MKEYS (possible main keywords)
  - SKEYS (possible sub keywords)
  - is_empty (state if file is empty)
- **BlockFile class methods**
  - add_block()
  - update_block() (modify existing)
  - get_block()
  - read_file() (load existing file)
  - write_file() (to model folder)
  - save() (arbitrary location)

The MSH file

```
 9 model.msh.generate(
10     "rectengular",
11     dim=2,
12     mesh_origin=(-100.0, -100.0),
13     element_no=(200, 200),
14     element_size=(1.0, 1.0),
15 )
16 # or load an unstructured mesh
17 model.msh.import_mesh("mesh.vtk")
```

- **MSH class methods**
  - combine_mesh() (with another ogs-mesh)
  - [import/export]_mesh() (multiple types)
  - generate()
  - load() / save() (ogs meshes IO)
  - rotate() / shift() / transform()

- **MSH class attributes**
  - AXISYMMETRY
  - CROSS_SECTION
  - ELEMENTS (sorted by type)
  - ELEMENT_[ID/NO]
  - GEO_[NAME/TYPE]
  - LAYER
  - MATERIAL_ID[_flat]
  - NODES[_NO]
  - PCS_TYPE
  - centroids[_flat] (centers of elem.)
  - volumes[_flat] (volumes of elem.)

The GLI file

```python
model.gli.generate(
    "rectengular",
    dim=2,
    ori=(-100.0, -100.0),
    size=(200.0, 200.0),
    name="boundary",
)
# add the pumping well
model.gli.add_points([0.0, 0.0, 0.0], "pwell")
# add observations
model.gli.add_points([10.0, 0.0, 0.0], "owell")
# add line
model.gli.add_polyline(points=["pwell", "owell"], name="line")
```

- **GLI class methods**
  - add_[points/polyline/surface/volume]
  - generate()
  - load() / save()
  - rotate() / shift() / transform()

- **GLI class attributes**
  - POINTS[_MD/_NANES/_NO]
  - POLYLINES[_NAMES/_NO]
  - SURFACES[_NAMES/_NO]
  - VOLUMES[_NAMES/_NO]

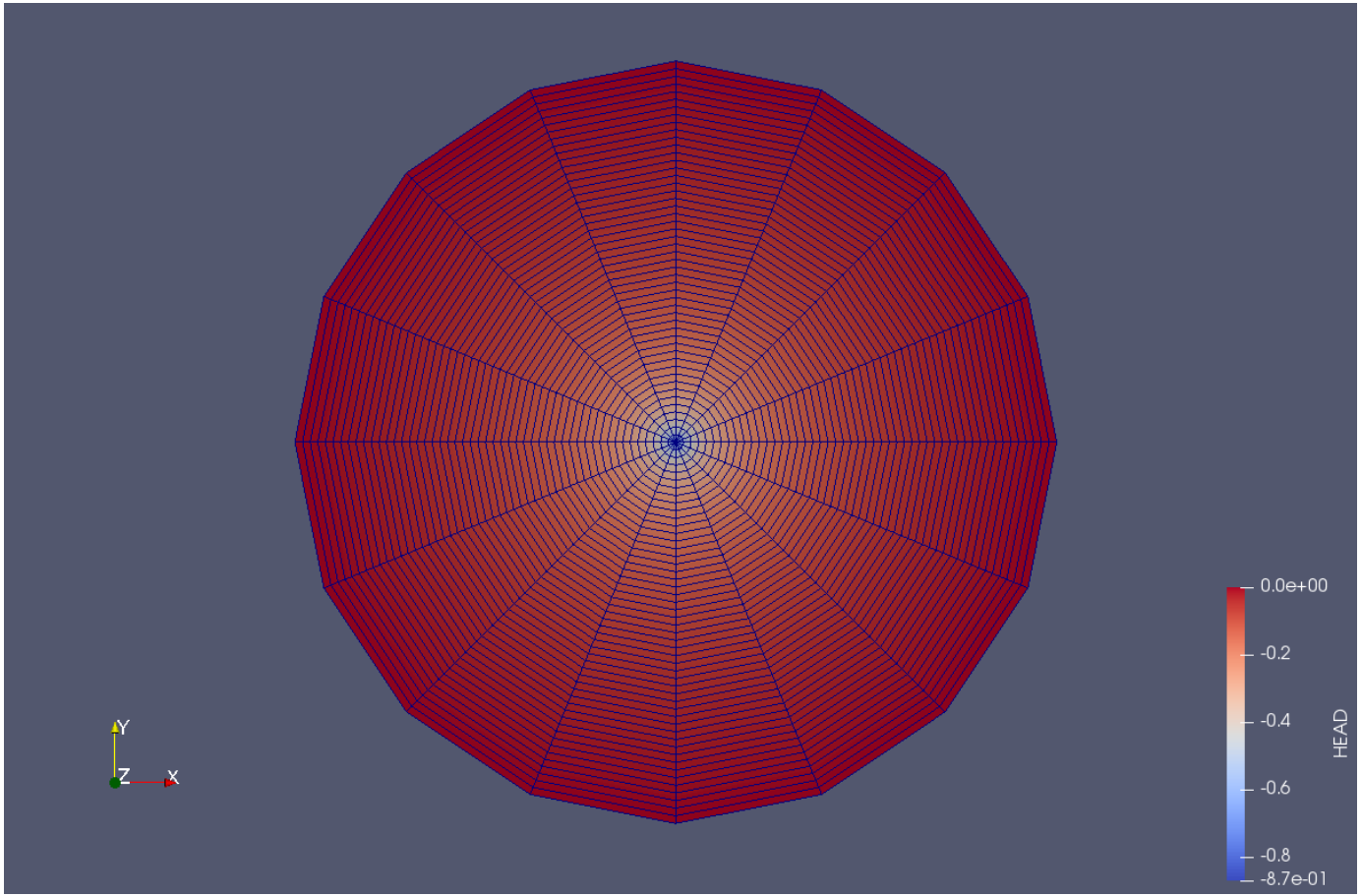
```
|-------------- Written with ogs5py --------------|
#POINTS
0 -100.0 -100.0 0.0
1 100.0 -100.0 0.0
2 100.0 100.0 0.0
3 -100.0 100.0 0.0
4 0.0 0.0 0.0 $NAME pwell
5 10.0 0.0 0.0 $NAME owell
#POLYLINE
  $NAME
   boundary
  $POINTS
   0
   1
   2
   3
   0
#POLYLINE
  $NAME
   line
  $POINTS
   4
   5
#STOP
|-- Written with ogs5py (0.5.0rc1) on: 2019-01-10_23-42-14 --|
```

Pumping test example

```python
from ogs5py import OGS
model = OGS(task_root="pump_test", task_id="model")

model.msh.generate("radial", dim=2, rad=range(51))
model.gli.generate("radial", dim=2, rad_out=50.)
model.gli.add_points([0., 0., 0.], "pwell")
model.gli.add_points([1., 0., 0.], "owell")

model.bc.add_block(  # boundary condition
    PCS_TYPE='GROUNDWATER_FLOW',
    PRIMARY_VARIABLE='HEAD',
    GEO_TYPE=['POLYLINE', "boundary"],
    DIS_TYPE=['CONSTANT', 0.0],
)
model.st.add_block(  # source term
    PCS_TYPE='GROUNDWATER_FLOW',
    PRIMARY_VARIABLE='HEAD',
    GEO_TYPE=['POINT', "pwell"],
    DIS_TYPE=['CONSTANT_NEUMANN', -1.0e-04],
)
model.ic.add_block(  # initial condition
    PCS_TYPE='GROUNDWATER_FLOW',
    PRIMARY_VARIABLE='HEAD',
    GEO_TYPE='DOMAIN',
    DIS_TYPE=['CONSTANT', 0.0],
)
model.mfp.add_block(  # fluid properties
    FLUID_TYPE='LIQUID',
    DENSITY=[1, 1.0e+03],
    VISCOSITY=[1, 1.0e-03],
)
model.mmp.add_block(  # medium properties
    GEOMETRY_DIMENSION=2,
    STORAGE=[1, 1.0e-04],
    PERMEABILITY_TENSOR=['ISOTROPIC', 1.0e-4],
    POROSITY=0.2,
)
```

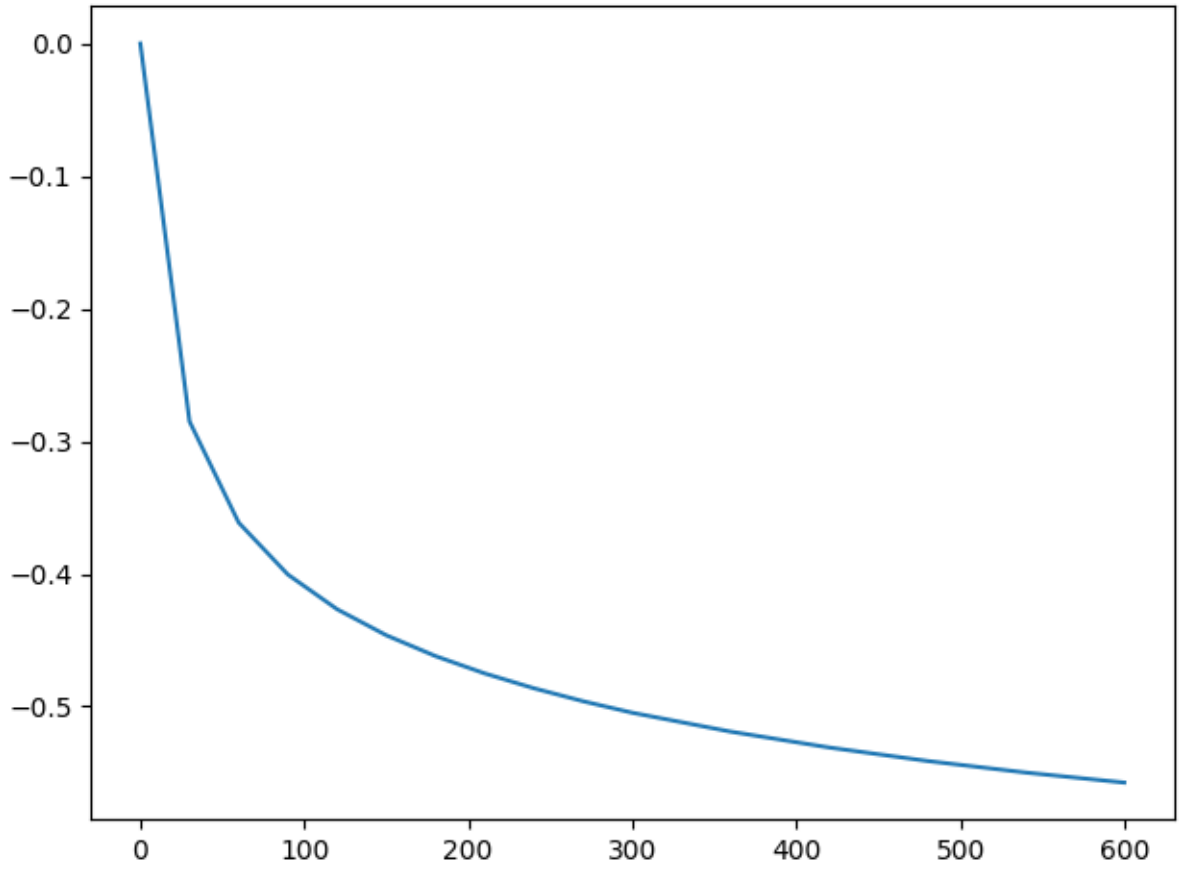
```python
model.num.add_block(  # numerical solver
    PCS_TYPE='GROUNDWATER_FLOW',
    LINEAR_SOLVER=[2, 5, 1.0e-14, 1000, 1.0, 100, 4],
)
model.out.add_block(  # domain output
    PCS_TYPE='GROUNDWATER_FLOW',
    NOD_VALUES='HEAD',
    GEO_TYPE='DOMAIN',
    DAT_TYPE='PVD',
    TIM_TYPE=['STEPS', 1],
)
model.out.add_block(  # point observation
    PCS_TYPE='GROUNDWATER_FLOW',
    NOD_VALUES='HEAD',
    GEO_TYPE=['POINT', "owell"],
    DAT_TYPE='TECPLOT',
    TIM_TYPE=['STEPS', 1],
)
model.pcs.add_block(  # set the process type
    PCS_TYPE='GROUNDWATER_FLOW',
    NUM_TYPE='NEW',
)
model.tim.add_block(  # set the timesteps
    PCS_TYPE='GROUNDWATER_FLOW',
    TIME_START=0,
    TIME_END=600,
    TIME_STEPS=[[10, 30], [5, 60]],
)

model.write_input()
success = model.run_model()
```

Pumping test example



Pumping test example

```
70 from ogs5py.reader import readtec_point
71 from matplotlib import pyplot as plt
72
73 point = readtec_point(
74     task_root="pump_test",
75     task_id="model",
76     pcs='GROUNDWATER_FLOW',
77 )
78 time = point['owell']["TIME"]
79 head = point['owell']["HEAD"]
80
81 plt.plot(time, head)
82 plt.show()
```



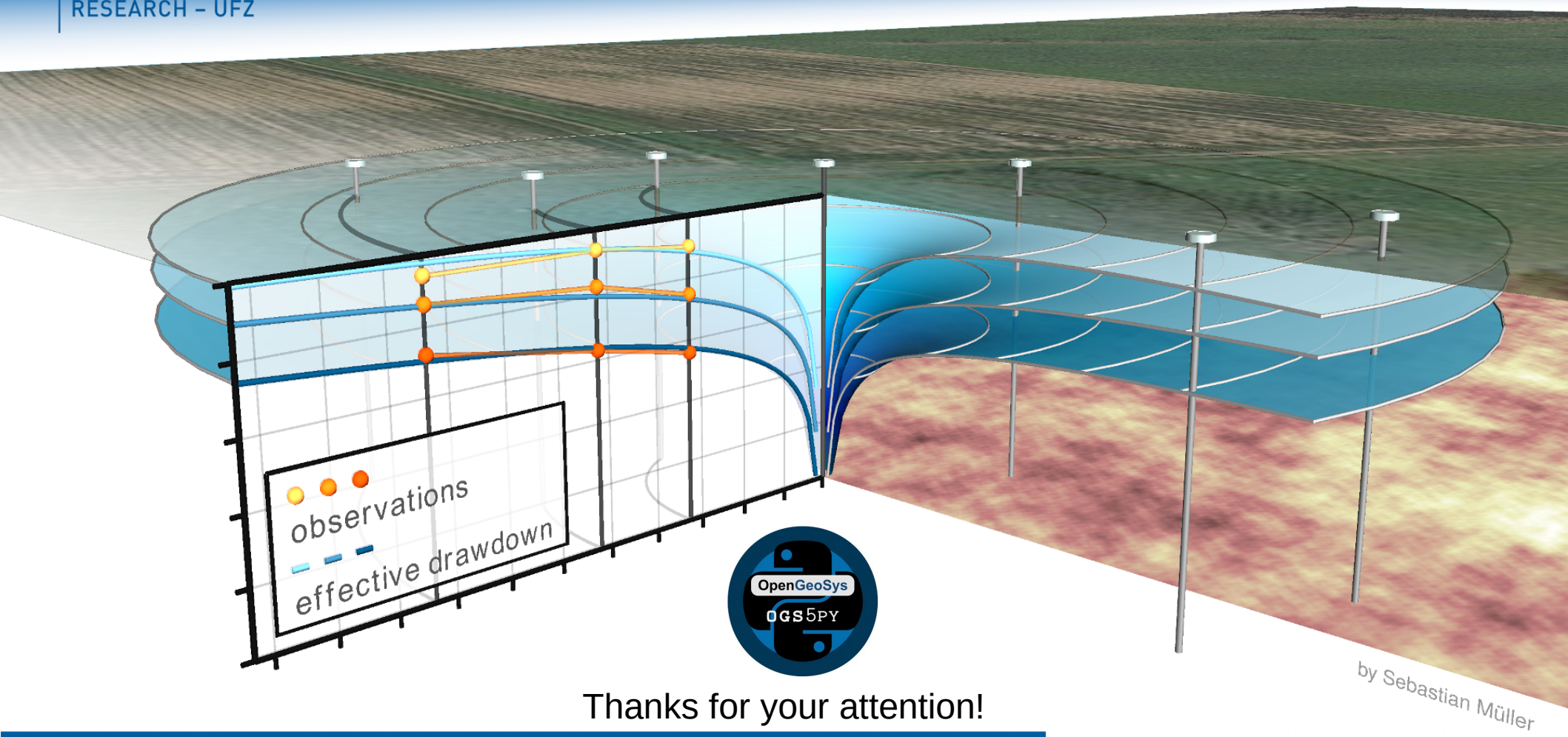
HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Pumping test example

```
          ######################################################
          ##                                                  ##
          ##                OpenGeoSys-Project                ##
          ##                                                  ##
          ##  Helmholtz Center for Environmental Research     ##
          ##    UFZ Leipzig - Environmental Informatics       ##
          ##                    TU Dresden                    ##
          ##                University of Kiel                ##
          ##              University of Edinburgh             ##
          ##           University of Tuebingen (ZAG)          ##
          ##         Federal Institute for Geosciences        ##
          ##            and Natural Resources (BGR)           ##
          ##  German Research Centre for Geosciences (GFZ)    ##
          ##                                                  ##
          ##    Version 5.7(WH/WW/LB)  Date 07.07.2015        ##
          ##                                                  ##
          ######################################################

          File name (without extension): model

------------------------------------------
Data input:
GEOLIB::readGLIFile open stream from file model.gli ... done
read points from stream ...  ok, 18 points read
read polylines from stream ...  ok, 1 polylines read
tag #SURFACE not found or input stream error in GEOObjects
PCSRead ... done, read 1 processes
MFPRead
```

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

OGS User Meeting (Jan 2019)



Thanks for your attention!