Pre- and post-processing tools for OPM: pycopm and plopm

David Landa-Marbán

Computational Geosciences and Modelling

dmar@norceresearch.no

NORCE Norwegian Research Centre AS

Centre for Sustainable Subsurface Resources (CSSR)

OPM summit, Bergen, May 26-27th, 2025

Outline

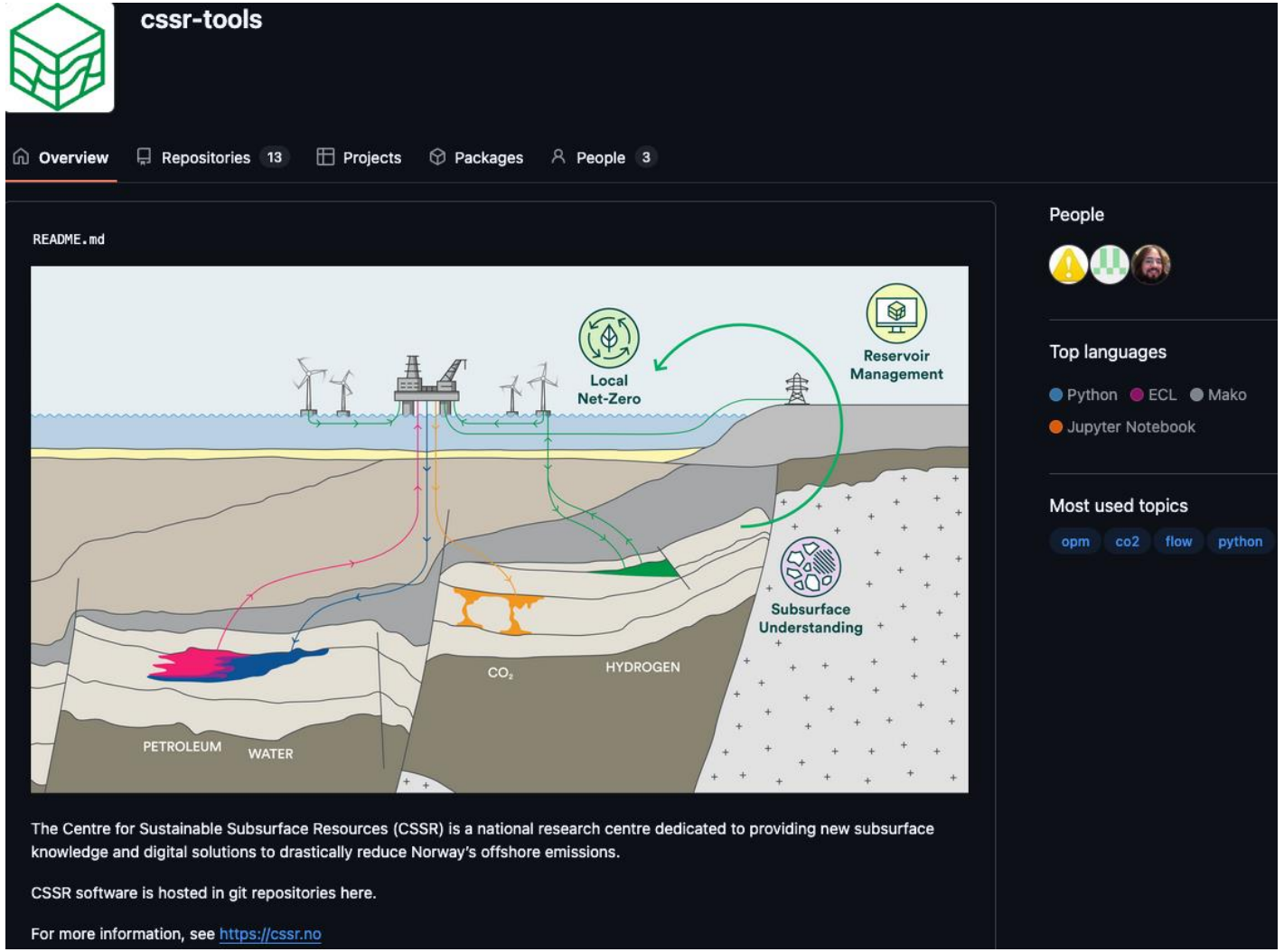
Overview of the cssr-tools GitHub page

Adopted approach to develop them

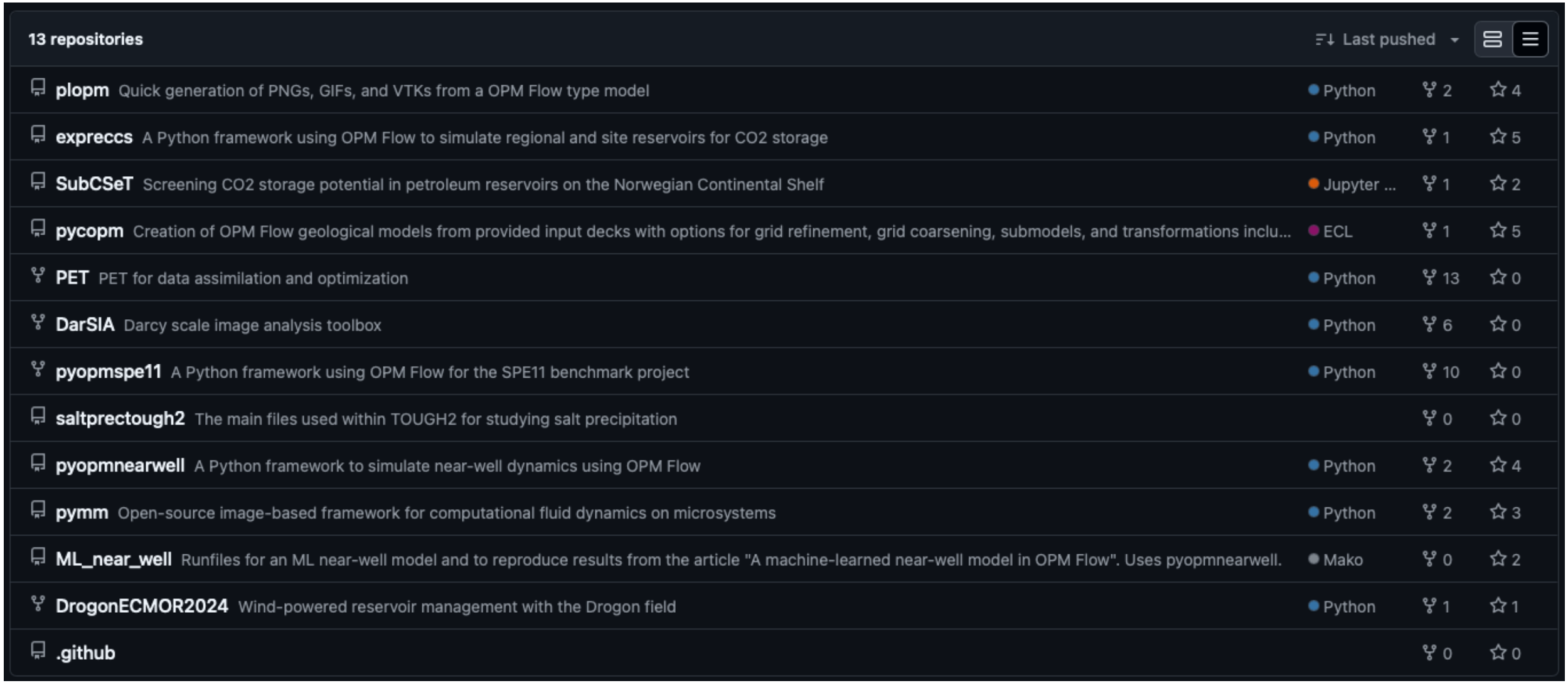
Summary

https://github.com/cssr-tools

- CSSR started in July 2022
- cssr-tools launched in February 2024
- 14 people (NORCE and UiB)
- 9 hosted repositories
- 4 forked repositories



https://github.com/orgs/cssr-tools/repositories



13 repositories — Last pushed

plopm	Quick generation of PNGs, GIFs, and VTKs from a OPM Flow type model	Python	2	4
expreccs	A Python framework using OPM Flow to simulate regional and site reservoirs for CO2 storage	Python	1	5
SubCSeT	Screening CO2 storage potential in petroleum reservoirs on the Norwegian Continental Shelf	Jupyter ...	1	2
pycopm	Creation of OPM Flow geological models from provided input decks with options for grid refinement, grid coarsening, submodels, and transformations inclu...	ECL	1	5
PET	PET for data assimilation and optimization	Python	13	0
DarSIA	Darcy scale image analysis toolbox	Python	6	0
pyopmspe11	A Python framework using OPM Flow for the SPE11 benchmark project	Python	10	0
saltprectough2	The main files used within TOUGH2 for studying salt precipitation		0	0
pyopmnearwell	A Python framework to simulate near-well dynamics using OPM Flow	Python	2	4
pymm	Open-source image-based framework for computational fluid dynamics on microsystems	Python	2	3
ML_near_well	Runfiles for an ML near-well model and to reproduce results from the article "A machine-learned near-well model in OPM Flow". Uses pyopmnearwell.	Mako	0	2
DrogonECMOR2024	Wind-powered reservoir management with the Drogon field	Python	1	1
.github			0	0

User- and developer-friendly features

“Python” as programming language.

“GitHub” as hosting service.

“Black” for code formatter/beautifier.

“Pylint” for code analysis.

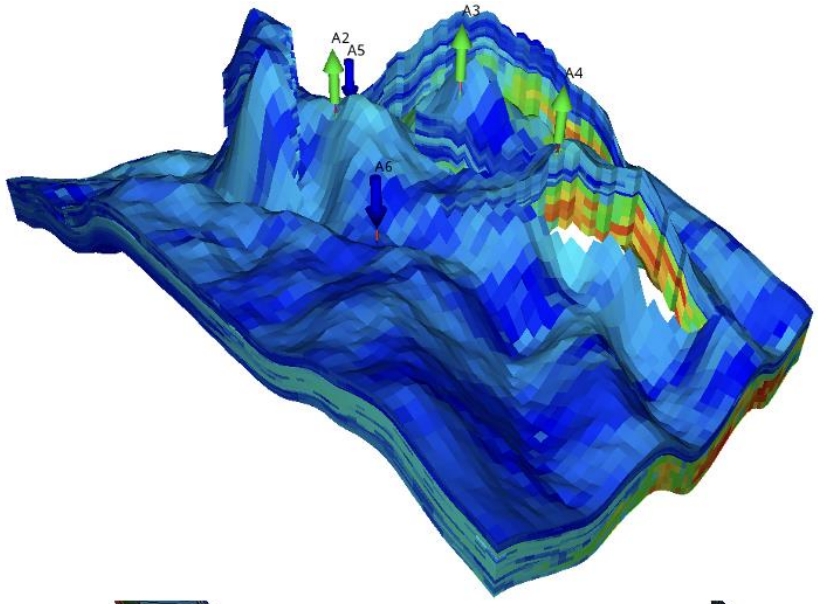
“pytest” for code testing.

“Read the Docs” for documentation.

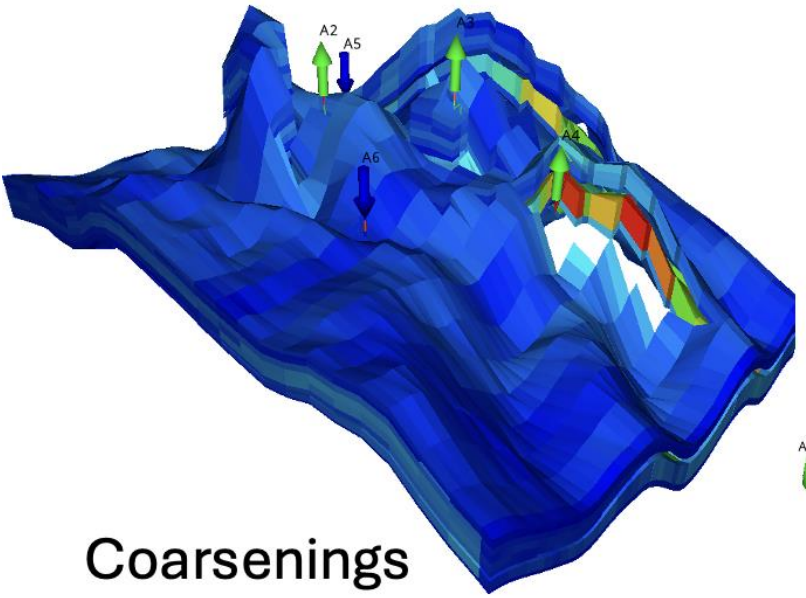
“TOML” for configuration files.

Preprocessing tool

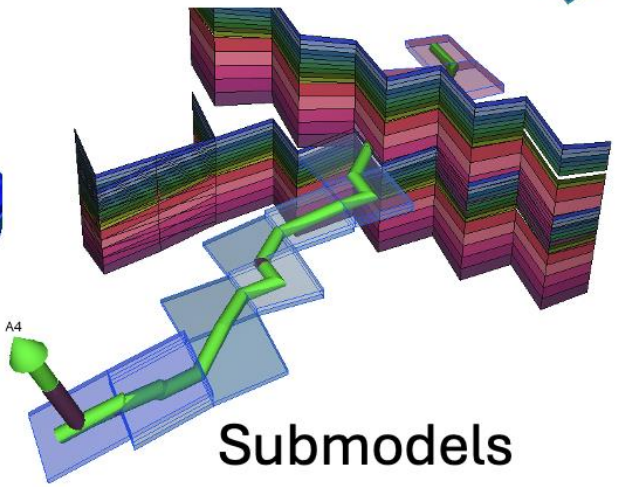
pycopm



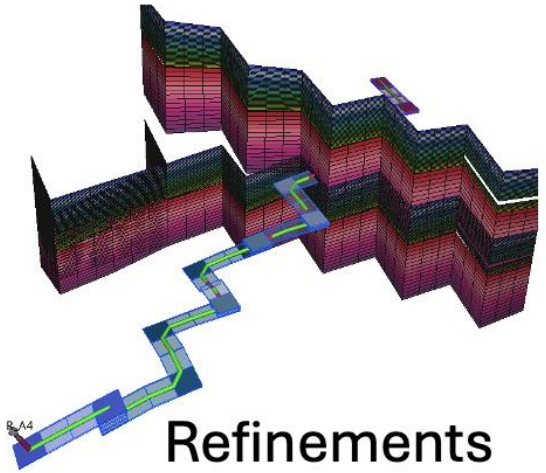
User-friendly creation of OPM Flow geological models from provided input decks



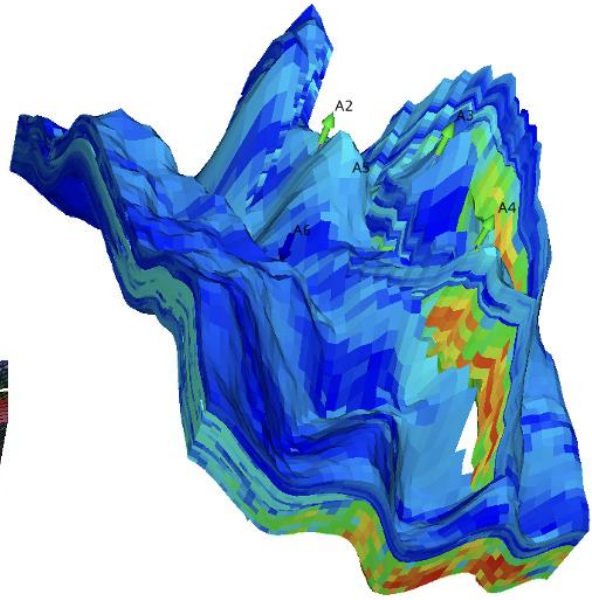
Coarsenings



Submodels

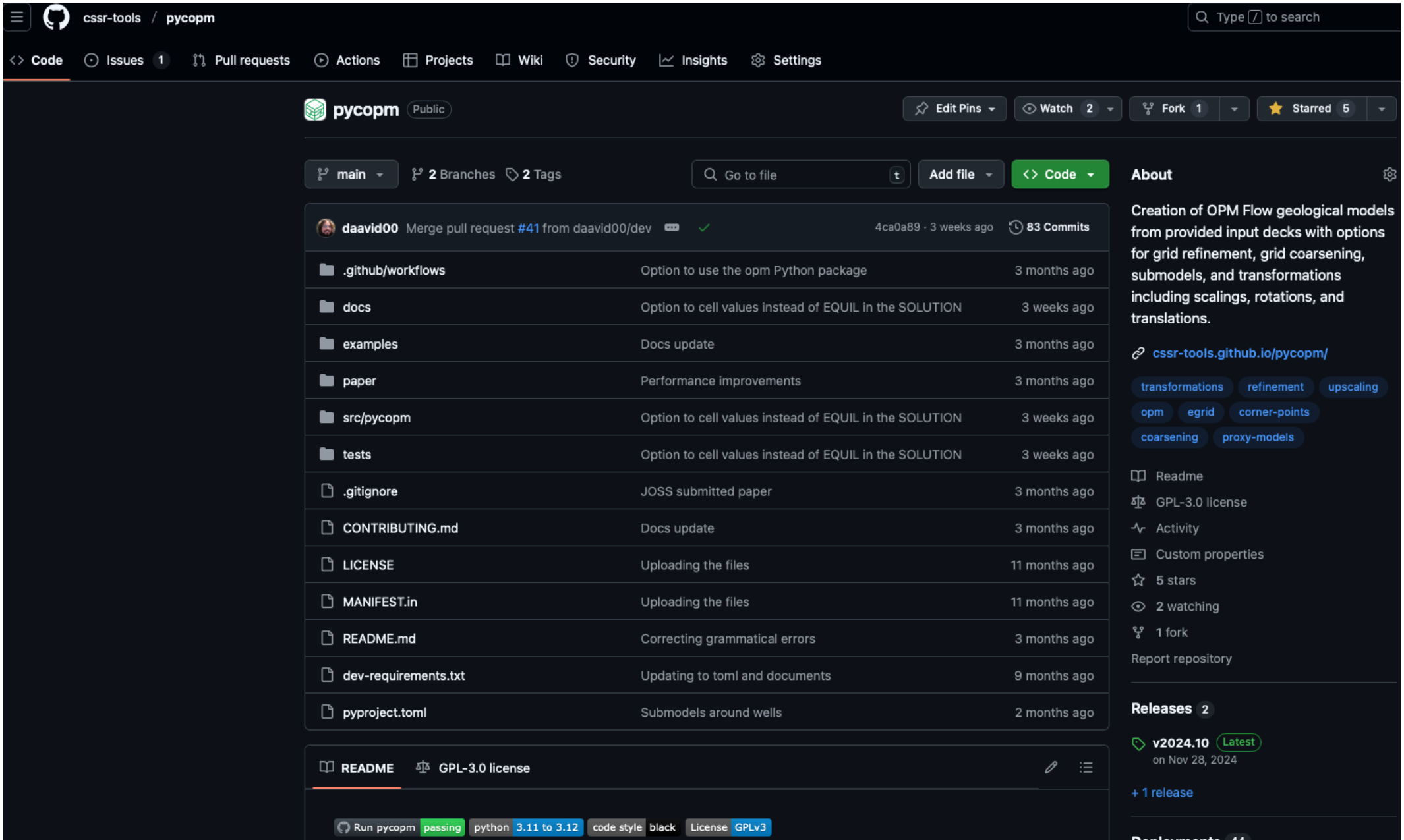


Refinements



Transformations

https://github.com/cssr-tools/pycopm



https://cssr-tools.github.io/pycopm/

Search docs

Welcome to pycopm's documentation!

https://cssr-tools.github.io/pycopm/installation.html

Search docs

Installation View page source

Installation

The following steps work installing the dependencies in Linux via apt-get or in macOS using brew or macports. While using package managers such as Anaconda, Miniforge, or Mamba might work, these are not tested. In addition, the current supported Python versions are 3.11 to 3.12. We will update the documentation when Python3.13 is supported (e.g., the resdata Python package is not yet available via pip install in Python 3.13).

Python package

To install the **pycopm** executable from the development version:

```
pip install git+https://github.com/cssr-tools/pycopm.git
```

If you are interested in a specific version (e.g., v2024.10) or in modifying the source code, then you can clone the repository and install the Python requirements in a virtual environment with the following commands:

```
# Clone the repo
git clone https://github.com/cssr-tools/pycopm.git
# Get inside the folder
cd pycopm
# For a specific version (e.g., v2024.10), or skip this step (i.e., edge version)
git checkout v2024.10
# Create virtual environment (to specific Python, python3.12 -m venv vpycopm)
python3 -m venv vpycopm
# Activate virtual environment
source vpycopm/bin/activate
# Upgrade pip, setuptools, and wheel
pip install --upgrade pip setuptools wheel
# Install the pycopm package
pip install -e .
# For contributions/testing/linting, install the dev-requirements
pip install -r dev-requirements.txt
```

pyproject.toml

```
dependencies = [
    "ert==13.0.4",
    "mako",
    "resdata"
]
```

dev-requirements.txt

```
black
numpydoc
mypy
pylint
pytest-cov
sphinx
sphinx-rtd-theme
```

Command-line interface

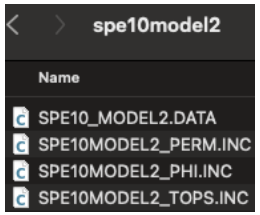
```
pycopm -i name_of_input_file
```

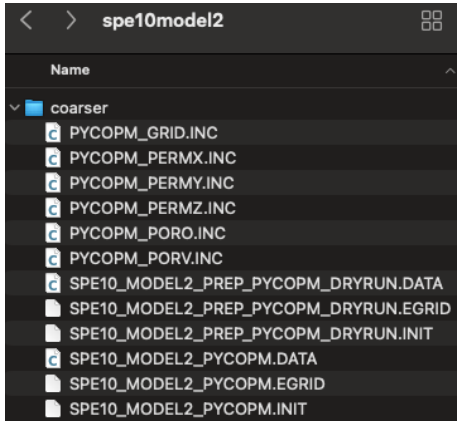
```
pycopm -help
```

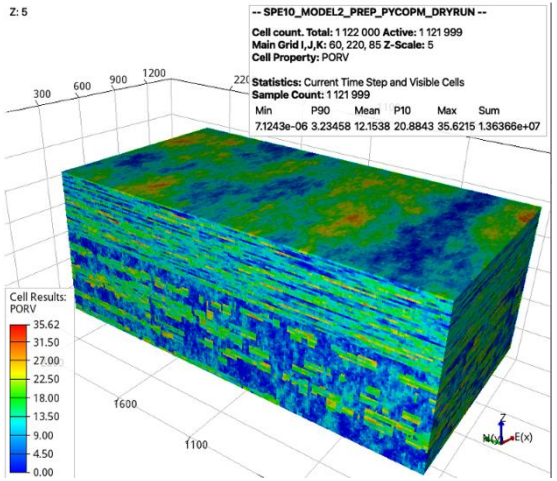
Examples: coarsenings

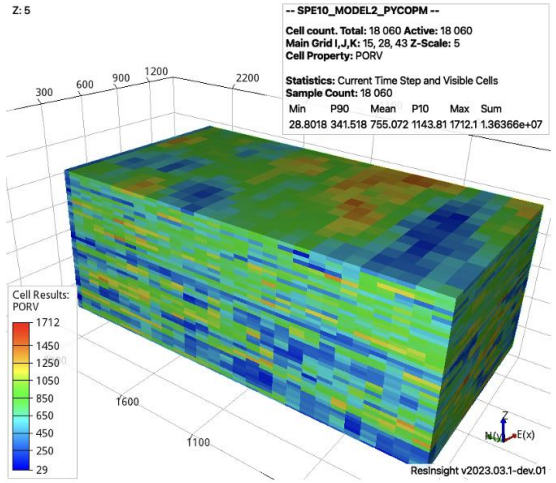
Deck available at https://github.com/OPM/opm-data/tree/master/spe10model2

```
pycopm -i SPE10_MODEL2.DATA -o coarser -c 4,8,2 -s pvmean -m all
```

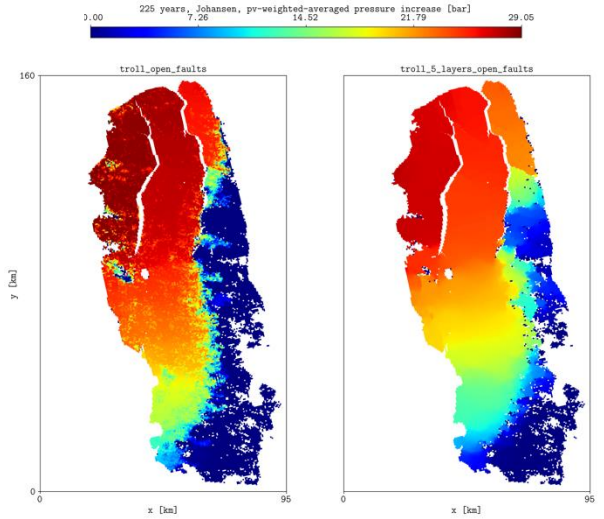




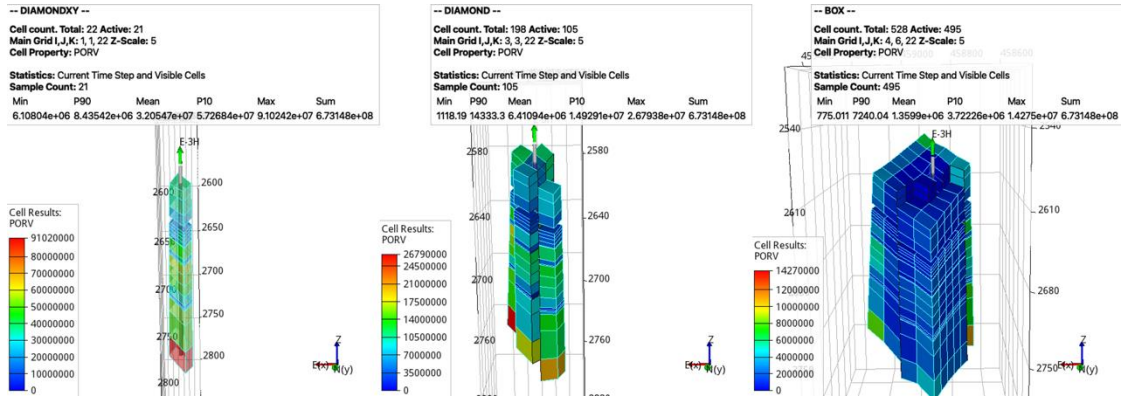




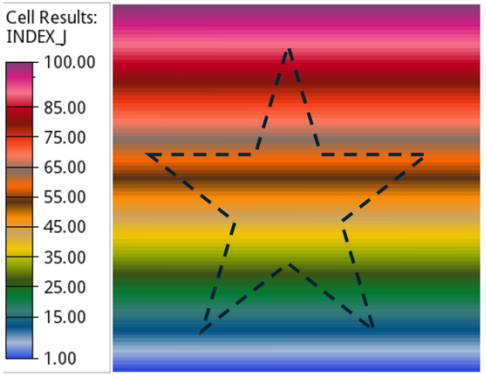
A study using the coarsening will be presented at TCCS-13 (16-19 June 2025), followed by a paper.

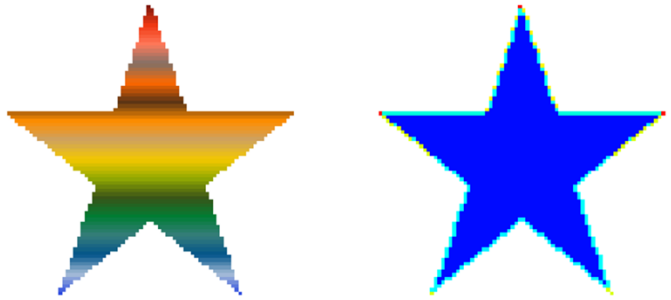


Examples: submodels and transformations

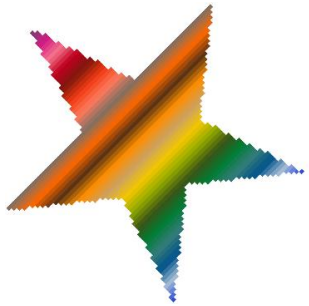


The submodel in NORNE_ATW3013.DATA by executing “-v ‘E-3H diamondxy 0’ -p 1”, “-v ‘E-3H diamond 1’ -p 1”, and “-v ‘E-3H box [-1,2] [-2,3] [-1,1]’ -p 1” respectively.





The shape to extract the sudmodel corresponds to “-v ‘xypolygon [50,90] [60,60] [90,60] [65,40] [75,10] [50,30] [25,10] [35,40] [10,60] [40,60] [50,90]’”

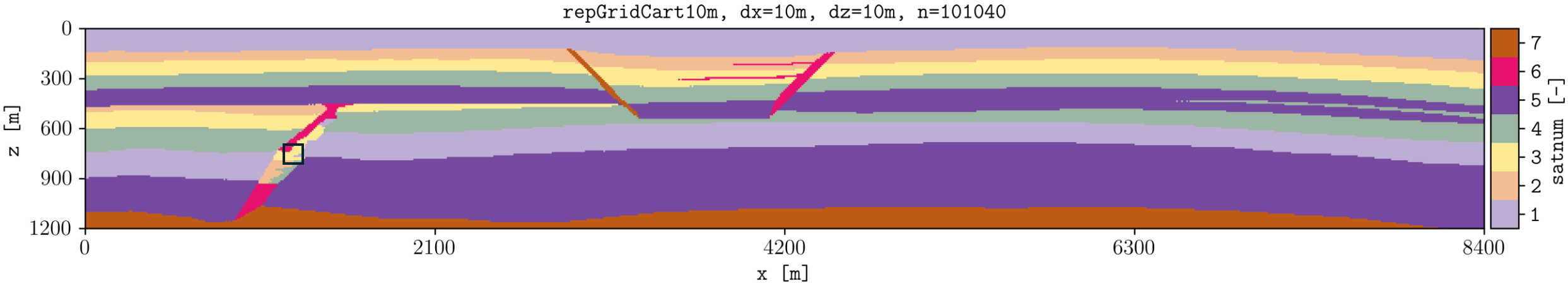


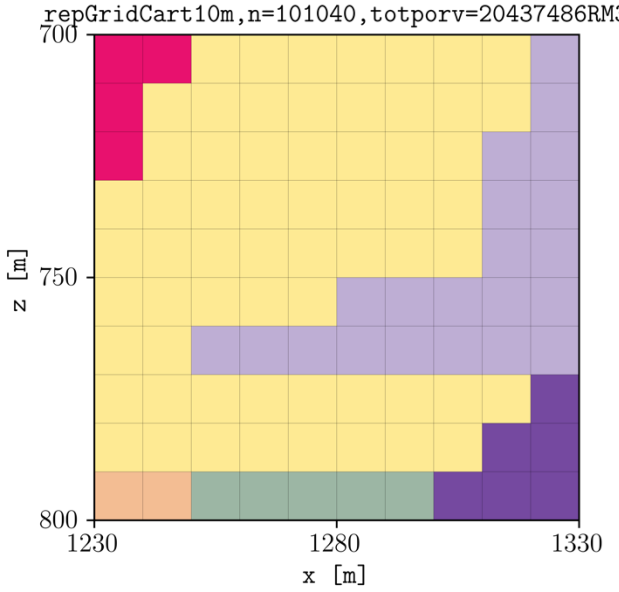
Previous extracted shape after a rotation “-d ‘rotatexy 45’” (left) and scaling “-d ‘scale [1,0.25,1]’” (right)

https://cssr-tools.github.io/pycopm/theory.html

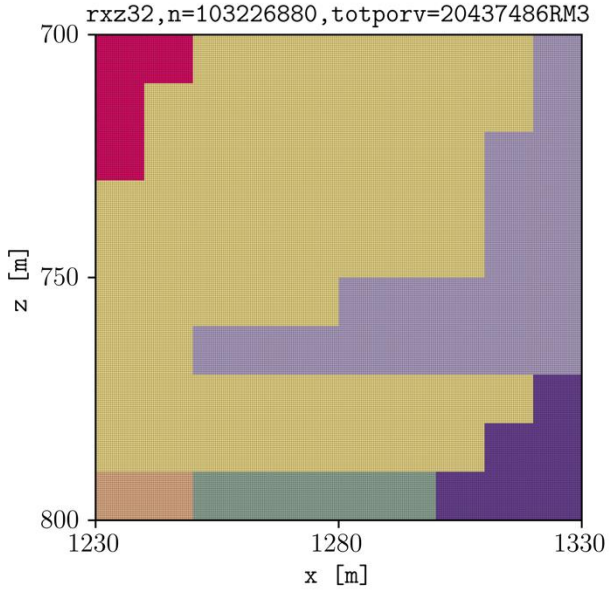
Examples: refinements

Using https://github.com/OPM/pyopmspe11 to generate the SPE11B case



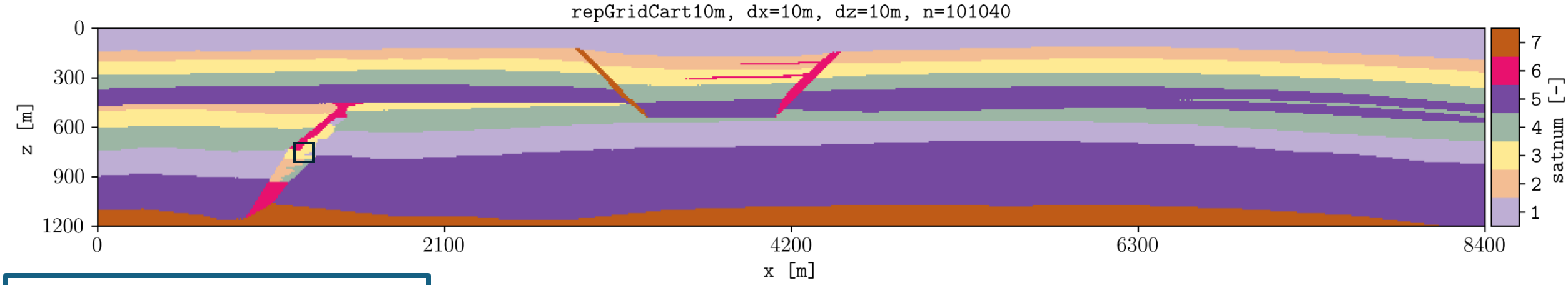



```
pycopm -i REPGRIDCART10M.DATA -u opm
-w RXZ32 -g 31,0,31
```



Examples: refinements

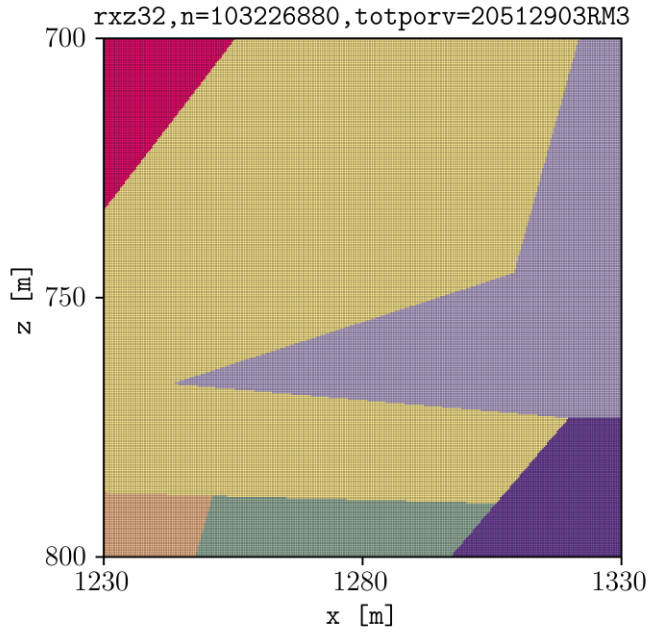
Using https://github.com/OPM/pyopmspe11 to generate the SPE11B case

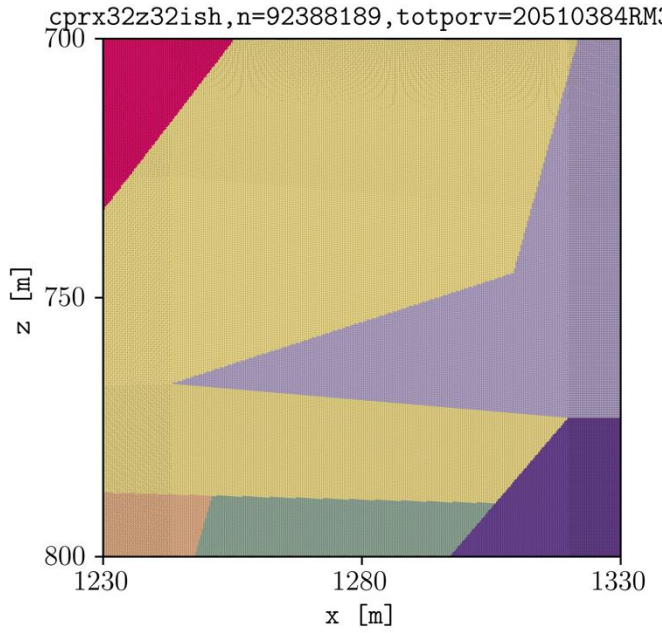


For finer grids which follows the facie lines, then one can use directly pyopmspe11.

Which tool is used to generate these figures?

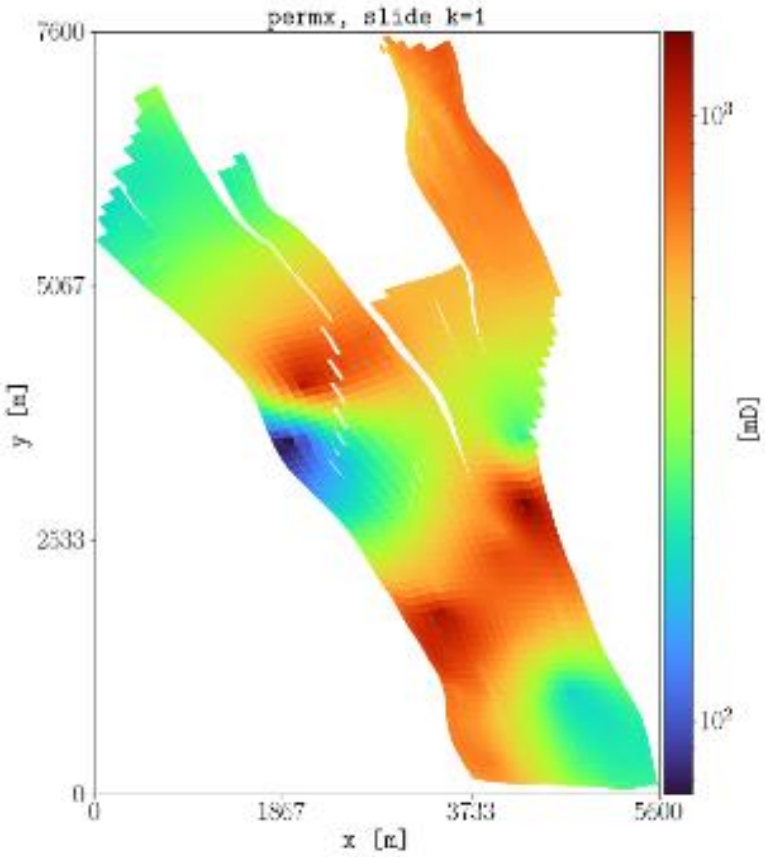
plopm



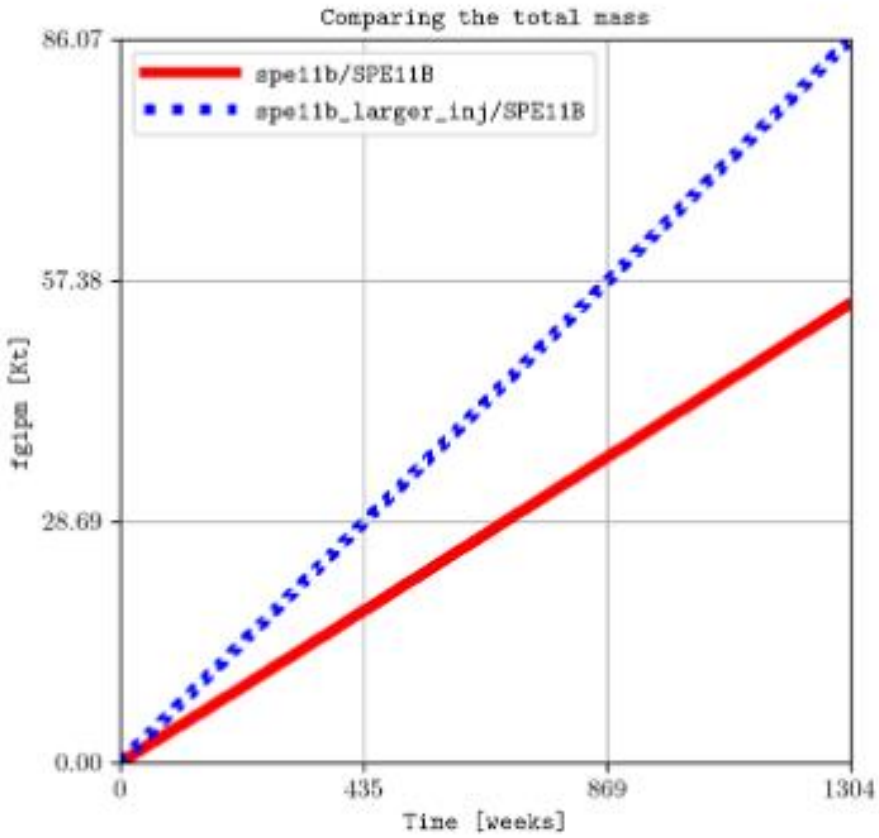


Postprocessing tool

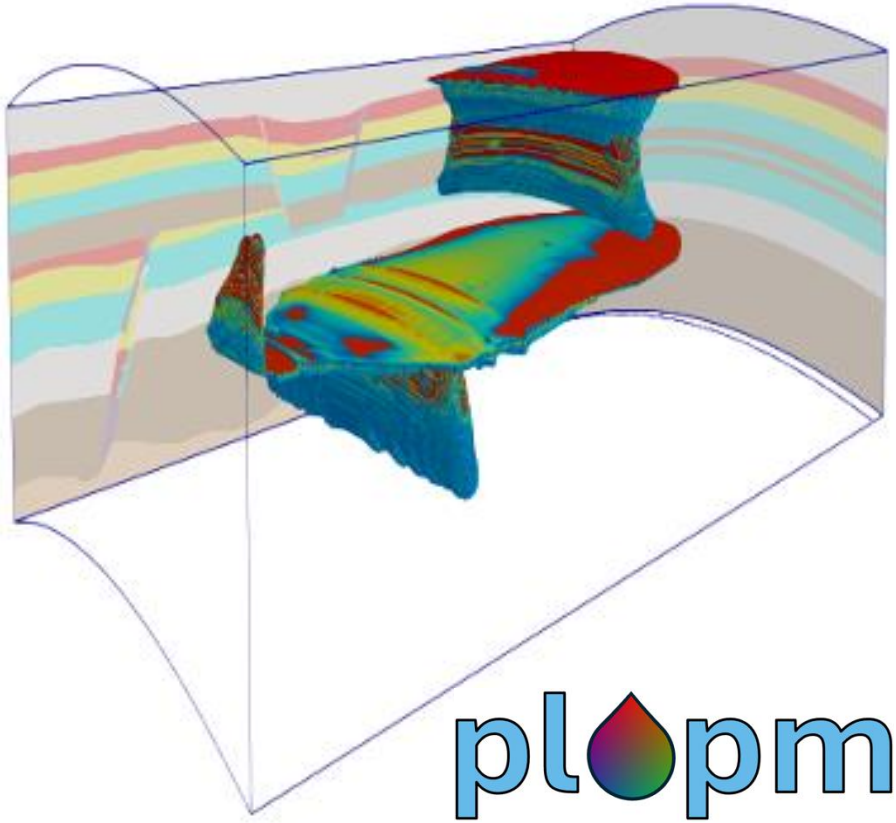
2D maps from any slide



Summary comparison



VTK conversion



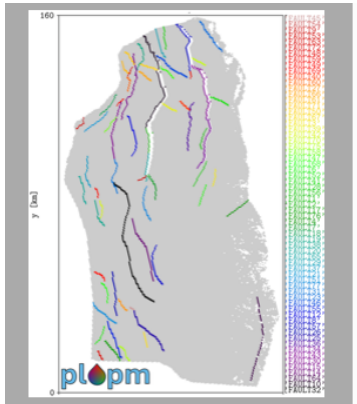
plopm

```
pip install git+https://github.com/cssr-tools/plopm.git
```

```
plopm -i name(s)_of_input_file(s)
```

```
plopm -help
```

https://cssr-tools.github.io/plopm/examples.html#



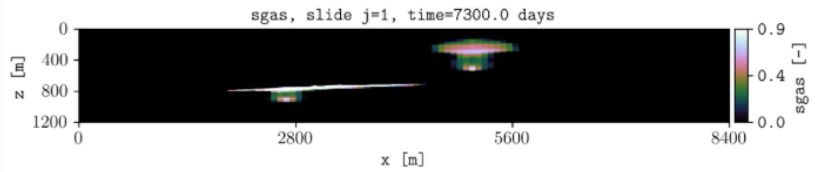
Search docs

Introduction
Installation
Examples
Hello world
Generic deck
Rotation, translation, and zoom
Projections and subfigures
Convert to VTK
Different input files and ensembles
GIF and mask
plopm Python API
Contributing
Related
About plopm

See the Overview or run *plopm -h* for the definition of the argument options, as well as using *-printv* flag to output the available summary, init, and restart available variables given an input deck.

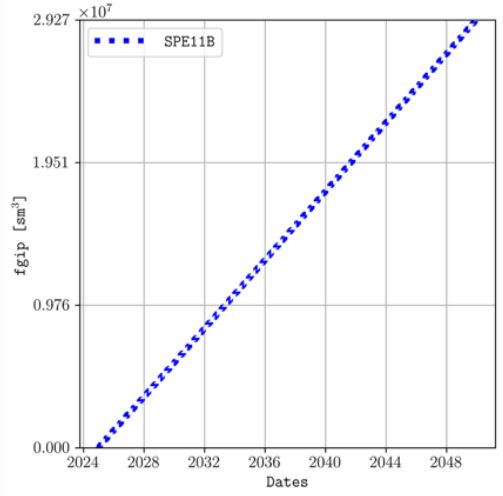
For example, for the gas saturation at the report step number 4 using a given colormap (-c):

```
plopm -i SPE11B -v sgas -r 4 -c cubehelix
```



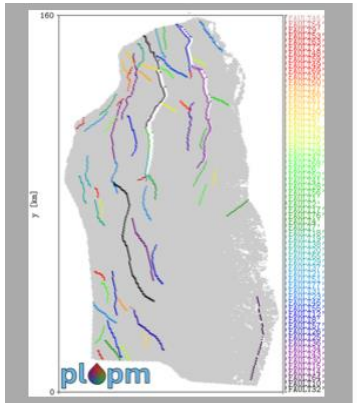
and for the gas in place summary vector given a color, line style, font size, dimension of the figure, line width, and using dates for the times:

```
plopm -i SPE11B -v fgip -c b -e dotted -f 12 -d 5,5 -lw 4 -tunits dates
```



To plot cell values over time, this can be achieved by:

```
plopm -i 'SPE11B SPE11B SPE11B' -v 'pressure - 0pressure' -s '1,1,1 41,1,29 83,1,58' -labels '
```



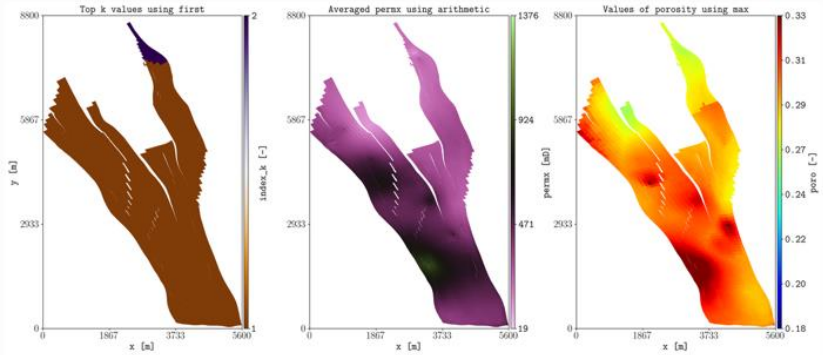
Search docs

Introduction
Installation
Examples
Hello world
Generic deck
Rotation, translation, and zoom
Projections and subfigures
Convert to VTK
Different input files and ensembles
GIF and mask
plopm Python API
Contributing
Related
About plopm

Projections and subfigures

Here is an example of making a single figure plotting subfigures and using different approaches to project the quantities:

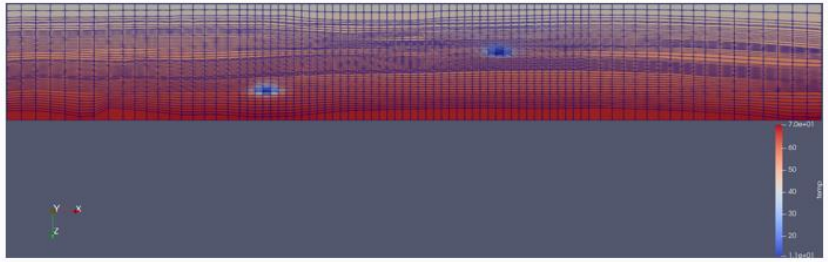
```
plopm -i NORNE_ATW2013 -v 'index_k,permx,poro' -s ',,1:22 ,,1:22 ,,1:22' -how 'first,arithmeti
```



Convert to VTK

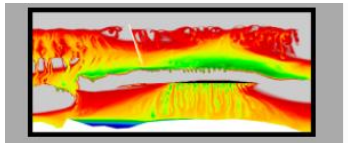
Inside the examples folder, then we can create VTKs from the OPM Flow simulation results (i.e., .EGRID, .INIT, .UNRST). For example, to create VTKS for the temperature, fipnum, the co2 mass, and the co2 mass fraction in the liquid phase from the restart files from the initial (0) to the number 5 restart, using a OPM Flow build from source in a given path, this can be achieved by:

```
plopm -i SPE11B -v temp,fipnum,co2m,xco2l -vtkformat Float32,UInt16,Float64,Float32 -r 0,5 -m
```



Visualization using paraview of the grid and temperature after 25 years of CO2 injection.

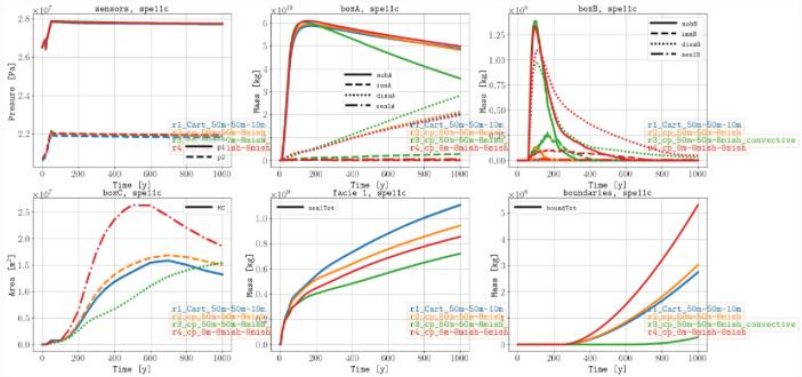
Additional examples



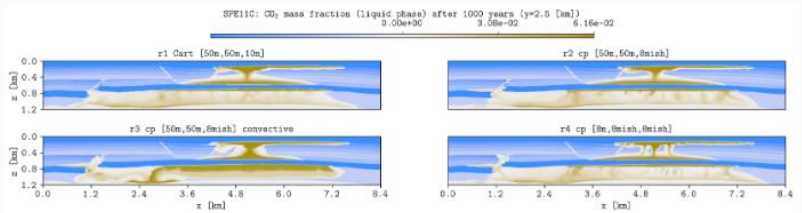
Search docs

- Introduction
- Installation
- Configuration file
- Examples
- Benchmark
  - SPE11A
  - SPE11B
  - SPE11C
- pyopmspe11 Python API
- Output folder
- Contributing
- Related
- About pyopmspe11

Sparse data



Spatial maps



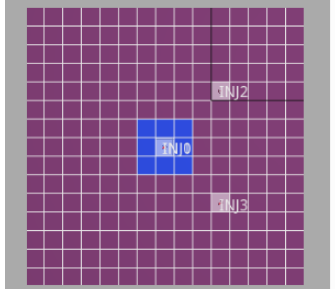
Note

To show the high resolution results, all spatial maps (spe11a, spe11b, and spe11c) corresponds to the actual simulation grid (not the benchmark reporting grid), and can be generated by typing in the terminal:

```
plopm -v xco2l -i 'r1_Cart_1cm/flow/R1_CART_1CM r2_Cart_1cm_capmax2500Pa/flow/R2_CART_1CM_C
plopm -v xco2l -i "r1_Cart_10m/flow/R1_CART_10M r2_cp_10mish/flow/R2_CP_10MISH r3_cp_10mish
plopm -v xco2l -i "r1_Cart_50m-50m-10m/flow/R1_CART_50M-50M-10M r2_cp_50m-50m-8mish/flow/R2
```

Tip

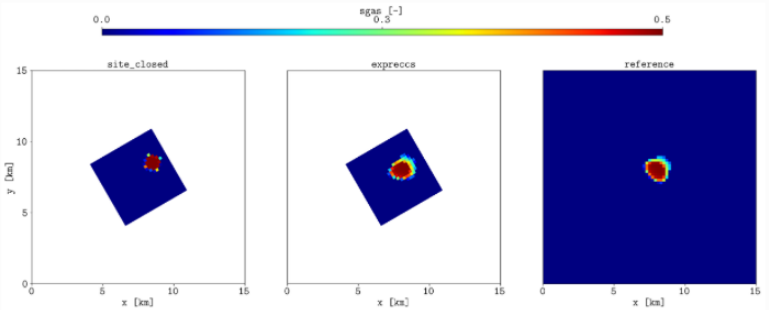
You can install plopm by executing in the terminal: pip install git+https://github.com/cssr-tools/plopm.git.



Search docs

- Introduction
- Installation
- Configuration file
- Examples
  - Via configuration files
  - Via OPM Flow decks
- expreccs Python API
- Contributing
- Output folder
- Related
- About expreccs

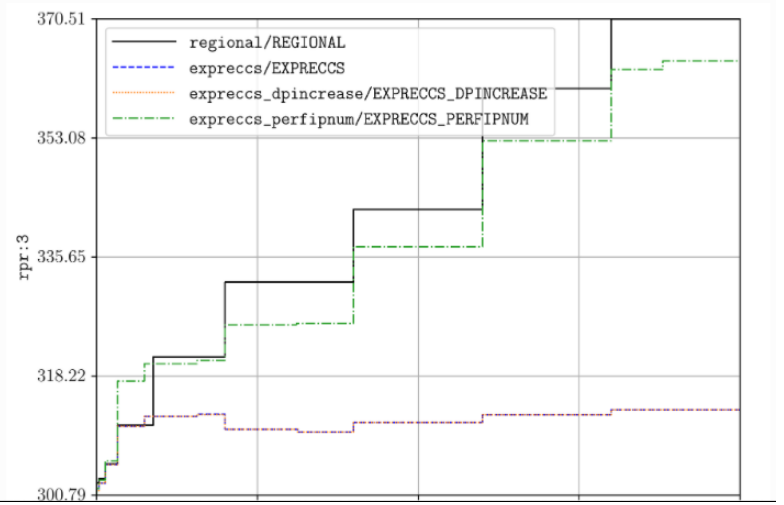
```
plopm -i 'tests/configs/rotate/output/site_closed/SITE_CLOSED tests/configs/rotate/output/expr
```



Comparison of the gas saturation on the top cells at the end of the simulations.

See also the test_4_site_regional.py, where the two-stage approach is demonstrated in a site and regional deck generated using sandwich.toml. In this test the functionality to select if the bc mappings are written per fipnums, i.e., if there is an offset between the site and regional models along the z direction, then using **-z 1** builds the interpolators per corresponding fipnums in the regional and site models. In addition, the flag **-e 0** set the interpolators to be defined by pressure increase and not by pressure values. If you run that test, then using plopm:

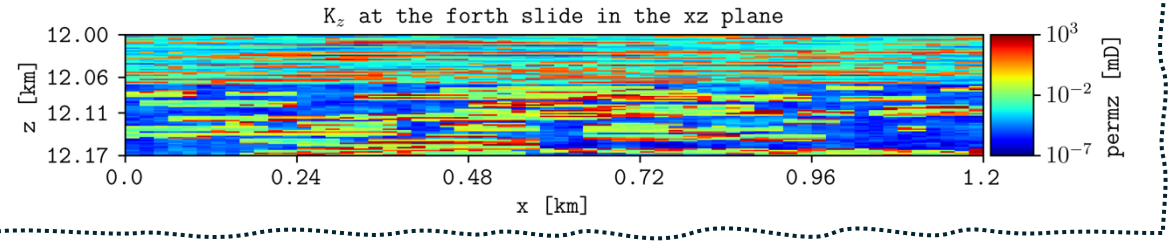
```
plopm -i 'regional/REGIONAL expreccs/EXPRECCS expreccs_dpincrease/EXPRECCS_DPINCREASE expreccs_
```

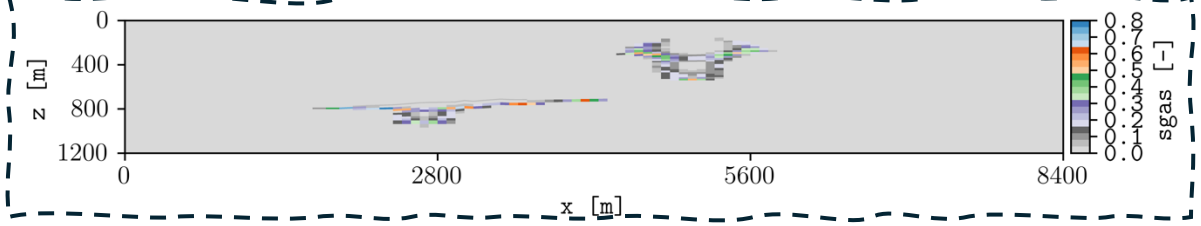


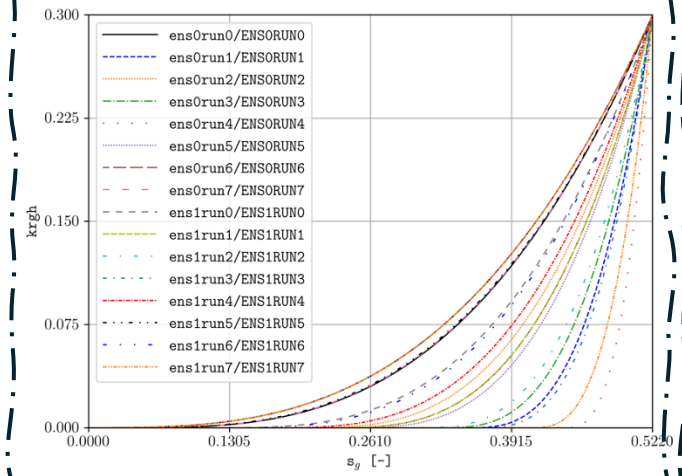
https://opm.github.io/pyopmspe11/benchmark.html

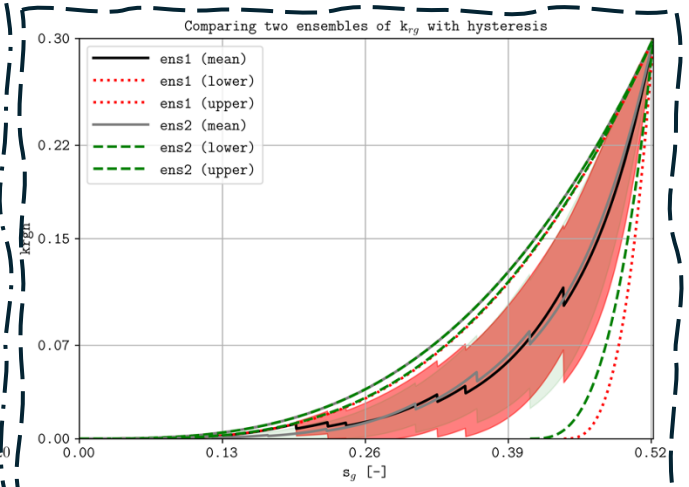
https://cssr-tools.github.io/expreccs/examples.html

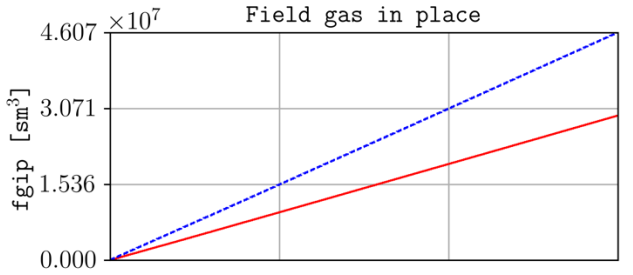
Gallery

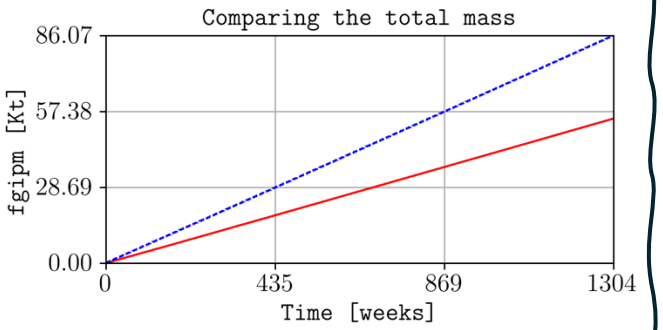

K_z at the forth slide in the xz plane
z [km]
x [km]
permz [mD]

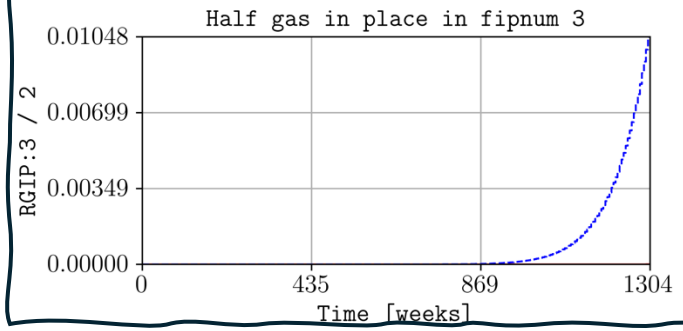

z [m]
x [m]
sgas [-]

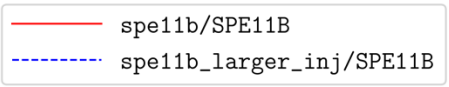

krgh
s_g [-]


Comparing two ensembles of k_{rg} with hysteresis
krgh
s_g [-]


Field gas in place
fgip [sm^3]


Comparing the total mass
fgipm [Kt]
Time [weeks]


Half gas in place in fipnum 3
RGIP:3 / 2
Time [weeks]


spe11b/SPE11B
spe11b_larger_inj/SPE11B

Gallery (behind the scenes)

```
plopm -i SPE10_MODEL2 -v permz -s ,4, -log 1 -xunits km -yunits km -xlnum 6 -yformat .2f -t 'K$_z$ at the forth slide in the xz plane' -b '[1e-7,1e3]'
```

```
plopm -i spe11b_larger_inj/SPE11B -v sgas -r 3 -diff spe11b/SPE11B -remove 0,0,0,1 -c tab20c_r -b '[0,0.8]' -cnum 9
```

```
plopm -i '.' -v krgh
```

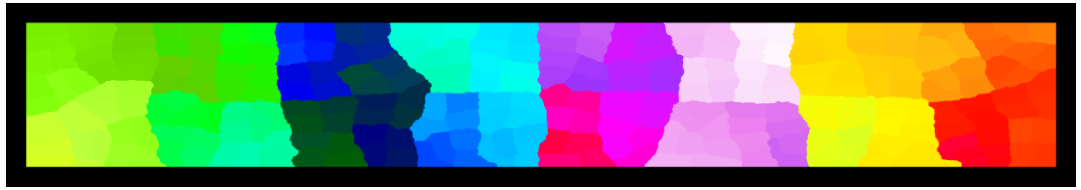
```
plopm -i 'ens0/ ens1/' -v krgh -ensemble 3 -label 'ens1 (mean) ens1 (lower) ens1 (upper) ens2 (mean) ens2 (lower) ens2 (upper)' -bandprop r,0.5,g,0.1 -c k,grey,r,g -e solid,solid,dotted,dashed -lw 2,2 -t 'Comparing two ensembles of k$_{rg}$ with hysteresis' -y '[0,0.3]' -yformat .2f -xformat .2f
```

```
plopm -v mpi_rank -remove '1,1,1,1' -c nipy_spectral -s ,,1 -facecolor k -u opm -i R4_CP_8M-8MISH-8MISH
```

```
plopm -i 'spe11b/SPE11B spe11b_larger_inj/SPE11B' -v 'fgip,fgipm,RGIP:3 / 2' -a 1,1e-6 -tunits w -d 10,5 -c r,b -e 'solid,dashed' -t 'Field gas in place Comparing the total mass Half gas in place in fipnum 3' -f 14 -subfigs 2,2 -delax 1 -loc empty,empty,empty,center -save comparison
```

Summary

- **github/cssr-tools** hosts Python-based user-friendly software solutions that might be of interest for OPM users/developers.
- Continuously updating the documentation and working examples is key to “keeping alive” the tools.
- Peer-review journals to publish the code (https://doi.org/10.21105/joss.07357).
- There is still a lot of work to extend functionality and improve performance in the tools.
- Issues and contributions are more than welcome using the fork and pull request approach 🙂.



Funding acknowledgement

HPC Simulation Software for the Gigatonne Storage Challenge project

[project no. 622059]

Center for Sustainable Subsurface Resources (CSSR)

[project no. 331841]

Expansion of Resources for CO2 Storage on the Horda Platform (ExpReCCS)

[project no. 336294]

Funded by The Research Council of Norway

https://github.com/cssr-tools

Centre for Sustainable Subsurface Resources

NORCE

```
cssr-tool -i input -o output
```

python

OpenFOAM®

OPM OPEN POROUS MEDIA

gmsh

equinor/**ert**

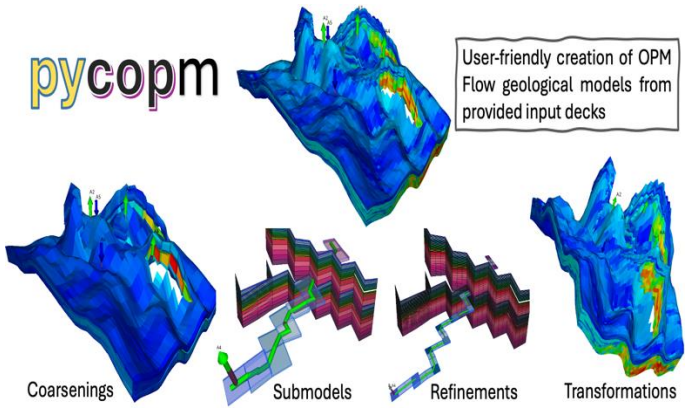
ERT - Ensemble based Reservoir Tool - is designed for running ensembles of dynamical models such as reservoir models, in...

equinor

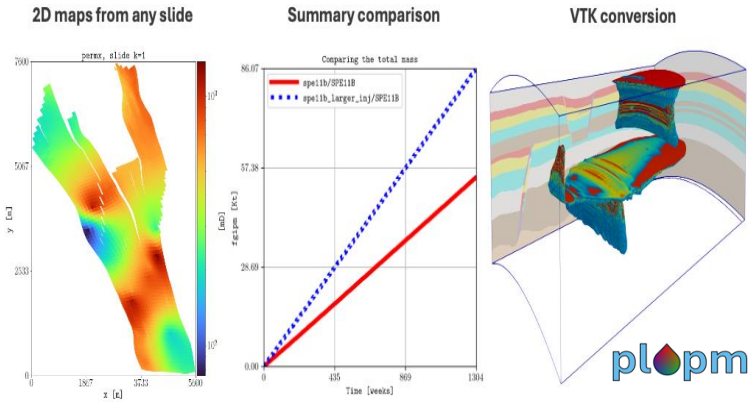
Python-Ensemble-Toolbox /**PET**

PET for data assimilation and optimization

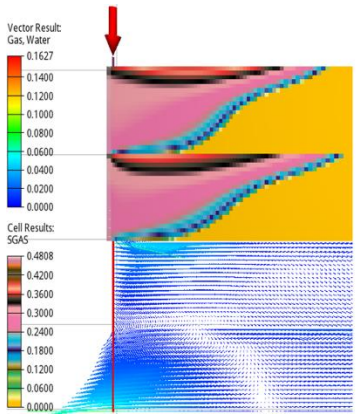
pycopm



plopm



pyopmnearwell



pyopmnearwell

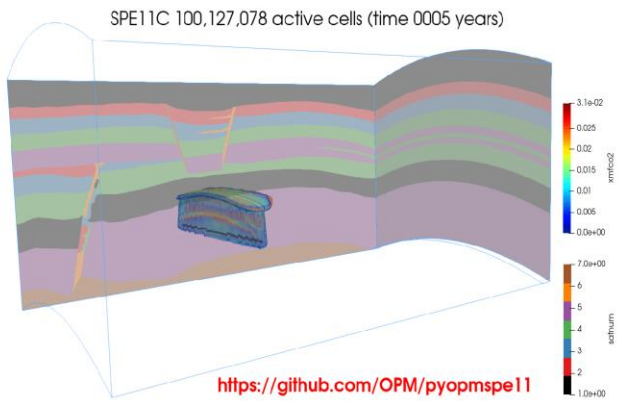
Simplified and flexible testing framework for near-well simulations via a configuration file using the OPM Flow simulator.

H_2store
CO_2store
Saltprec
CO_2EOR
Foam

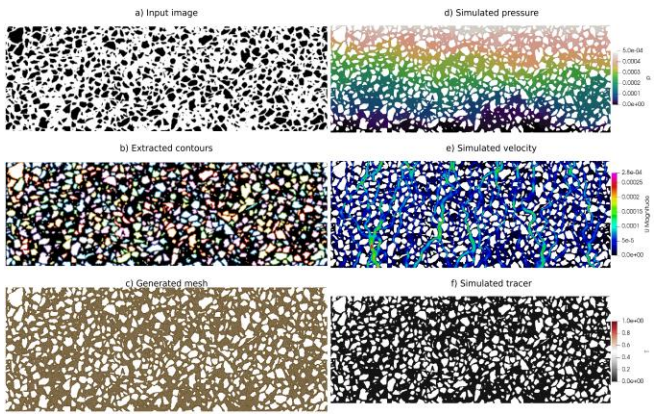
Radial grids
Core samples
Box grids

Well scheduling

pyopmspe11



pymm



expreccs