SPE11 scripts and framework

David Landa-Marbán* and Tor Harald Sandve
Computational Geosciences and Modelling
dmar@norceresearch.no

NORCE Norwegian Research Centre AS

Center for Sustainable Subsurface Resources (CSSR)

Outline

Overview of the SPE11 benchmark

pyopmspe11 reporsitory

Current results

Current work

Discussion

Overview of the SPE11 benchmark

https://www.spe.org/en/csp/

Call For Participation

This call for participation in the 11th Society of Petroleum Engineers Comparative Solution Project (the 11th SPE CSP) is motivated by the simulation challenges associated with CO_2 storage operations in geological settings, specifically developing simulations of realistic complexity.

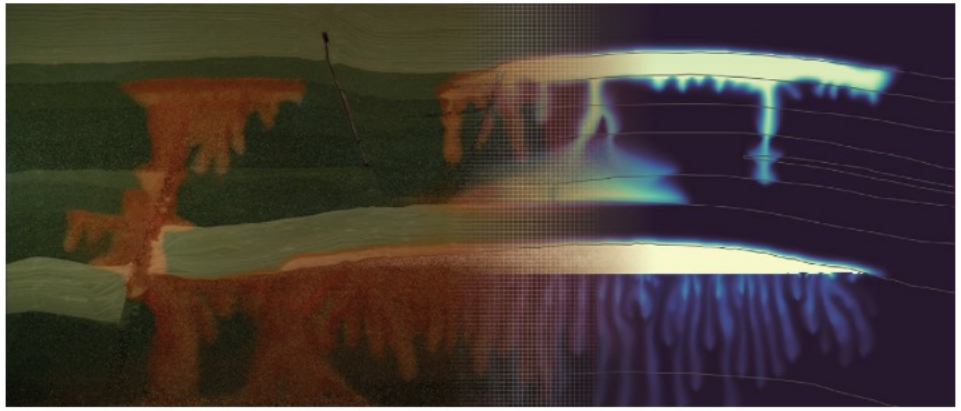


Image credit: Martin Fernø, Lluis Saló-Salgado and Jan Martin Nordbotten

The CSP contains three versions:

> *Version 11A* is a 2D geometry at the laboratory scale, inspired by a recent CO_2 storage forecasting and validation study.
>
> For *Version 11B*, the 2D geometry and operational conditions from 11A are rescaled to field conditions characteristic of the Norwegian Continental Shelf.
>
> Finally, for *Version 11C*, the geometry of version 11B is extruded to a full 3D field model.

More details can be found in the **CSP description** (PDF). Supplementary material such as geometry descriptions and scripts for thermodynamic relations can be found in the associated **GitHub repository**. Discuss the project with other participants by joining the **CSP discussion group on SPEConnect**.

OPM team

EQUINOR
NORCE
OPM-OP
SINTEF DIGITAL
TNO

TIMELINE

29 March 2023
Official announcement at 2023 SPE Reservoir Simulation Conference

16-18 October 2023
Special session at SPE ATCE

1 December 2023
Open call for participation period ends, Signed Agreement for Participation due

1 March 2024
Deadline for submission of early results

14 March 2024
First intercomparison workshop (13:00-17:00 CET, virtual)

1 September 2024
Deadline for submission of final CSP simulation results

30 September 2024
Final intercomparison workshop (hybrid)

December 2024
Completion of draft report on the results of the CSP

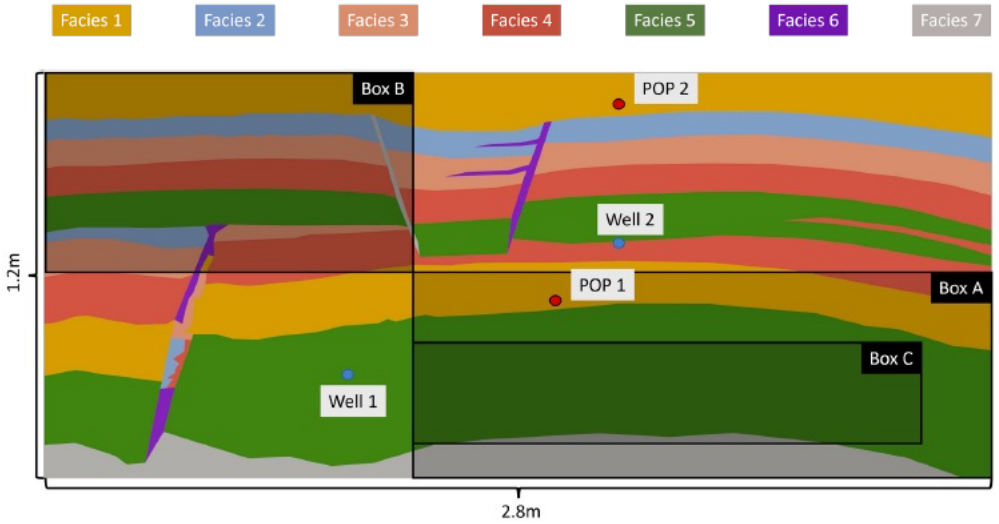
February 2025
Report on the results of the CSP finalized and submitted

March 2025
Special session at the 2025 SPE Reservoir Simulation Conference

CSP description

https://doi.org/10.2118/218015-PA

spe11a

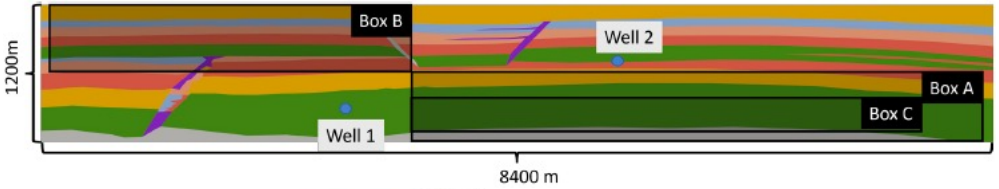


CO2 injection in well 1 for 5 hours

CO2 injection in well 2 for 2.5 hours (start after 2.5 hours)

Monitor until final time (5 days)

spe11b

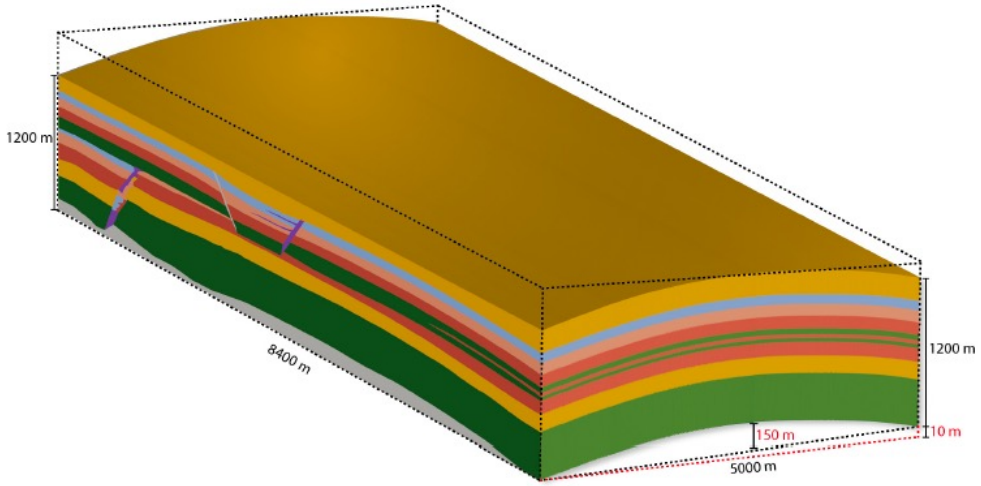


CO2 injection in well 1 for 50 years

CO2 injection in well 2 for 25 years (start after 25 years)

Monitor until final time (1000 years)

spe11c



pyopmspe11

A tool for the three SPE11 Cases (A, B, and C) to write input decks (e.g., .GRDECL for the corner-point grids), run the generated decks using OPM Flow, perform the postprocessing of the data in the format as requested in the benchmark, and generate .png figures for quick inspection of the results.

https://github.com/OPM/pyopmspe11

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.

Installation

You will first need to install

- Flow (https://opm-project.org, Release 2023.10 or current master branches)

You can install the Python requirements in a virtual environment with the following commands:

```
# Clone the repo
git clone https://github.com/OPM/pyopmspe11.git
# Get inside the folder
cd pyopmspe11
# Create virtual environment
python3 -m venv vpyopmspe11
# Activate virtual environment
source vpyopmspe11/bin/activate
# Upgrade pip, setuptools, and wheel
pip install --upgrade pip setuptools wheel
# Install the pyopmspe11 package (in editable mode for contributions/modifications
pip install -e .
# For contributions/testing/linting, install the dev-requirements
pip install -r dev-requirements.txt
```

requirements.txt

```
mako
matplotlib
pandas
Pyarrow
resdata
rtree
scipy
shapely
```

dev-requirements.txt

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

Running pyopmspe11

```
pyopmspe11 -i some_input.txt -o some_output_folder
```

```
pyopmspe11 --help
```

```
usage: pyopmspe11 [-h] [-i INPUT] [-m MODE] [-c COMPARE] [-o OUTPUT] [-t TIME] [-r RESOLUTION] [-g GENERATE] [-u USE] [-w WRITE]

Main script to run the spe11s with OPM Flow.

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        The base name of the input file ('input.txt' by default).
  -m MODE, --mode MODE  Run the whole framework ('all'), only create decks ('deck'), only run flow ('flow'), only write benchmark data ('data'), only create plots ('plot'), deck and run
                        ('deck_flow'), data and plot ('data_plot'), run and data ('flow_data'), or deck, run, and data ('deck_flow_data') ('deck_flow' by default).
  -c COMPARE, --compare COMPARE
                        Generate a common plot for the current folders ('' by default).
  -o OUTPUT, --output OUTPUT
                        The base name of the output folder ('output' by default).
  -t TIME, --time TIME  If one number, time step for the spatial maps (spe11a [h]; spe11b/c [y]) ('5' by default); otherwise, times separated by commas.
  -r RESOLUTION, --resolution RESOLUTION
                        Number of x, y, and z elements to map the simulation results to the dense report data ('8,1,5' by default).
  -g GENERATE, --generate GENERATE
                        Write only the 'dense', 'sparse', 'performance', 'performance-spatial', 'dense_performance', 'dense_sparse', 'performance_sparse', 'dense_performance-spatial', or
                        'all' ('performance_sparse') by default
  -u USE, --use USE     Using the 'opm' or 'resdata' python package ('resdata' by default).
  -w WRITE, --write WRITE
                        Time interval for the sparse and performance data (spe11a [h]; spe11b/c [y]) ('0.1' by default).
```

Configuration file (.txt)

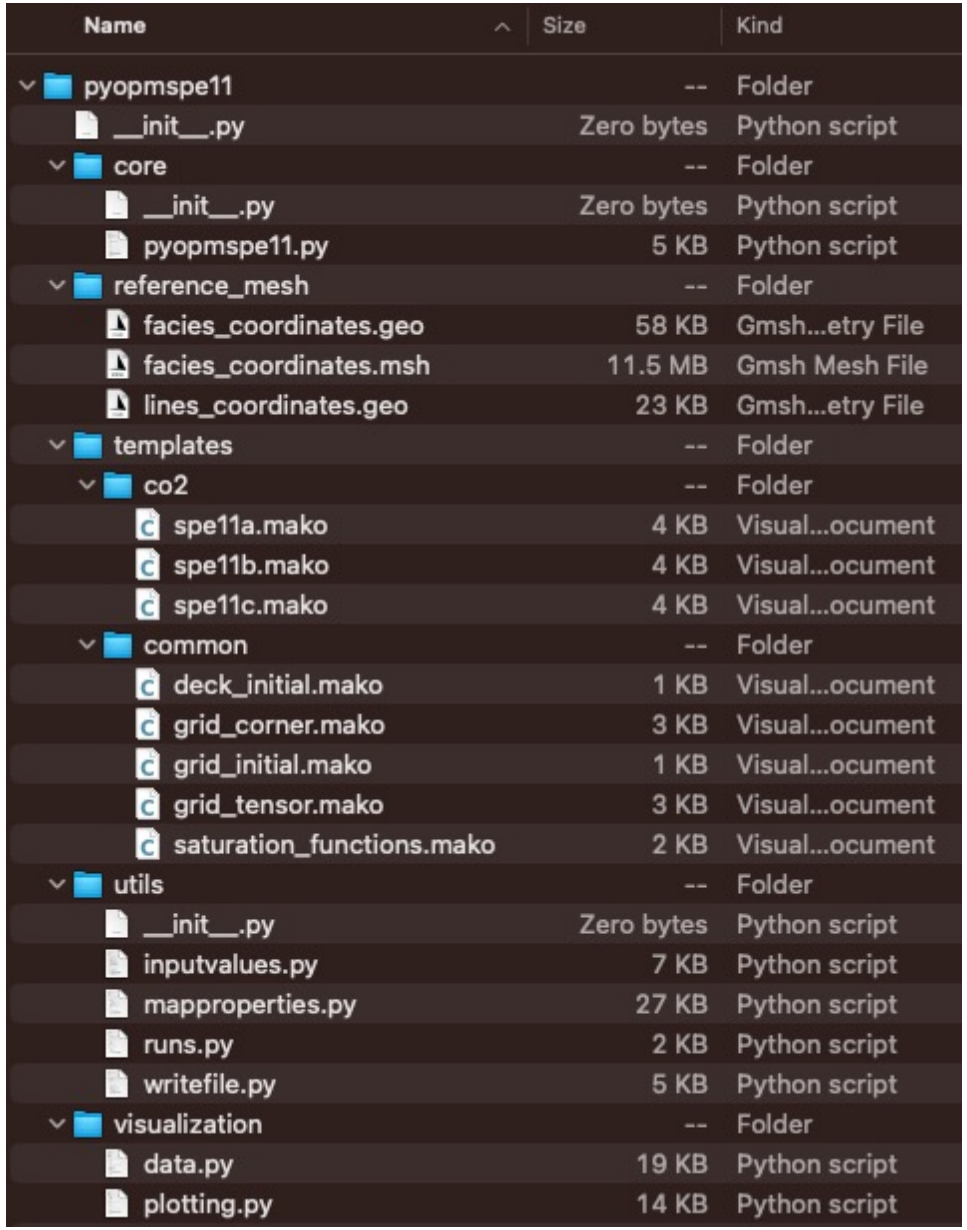
"""Set the full path to the flow executable and flags"""
mpirun -np 71 flow --zoltan-imbalance-tol=1.05 --tolerance-mb=1e-7 --linear-solver=cprw --enable-tuning=true --enable-opm-rst-file=true --output-extra-convergence-info=steps,iterations --newton-min-iterations=1

"""Set the model parameters"""
spe11c master #Name of the spe case (spe11a, spe11b, or spe11c) and OPM Flow version (master or release)
complete gaswater #CO2 model (immiscible or complete) and co2store implementation (gaswater or gasoil [oil properties are set to water internally in OPM flow])
corner-point #Type of grid (cartesian, tensor, or corner-point)
8400 5000 1200 #Length, width, and depth [m]
840 #If cartesian, number of x cells [-]; otherwise, variable array of x-refinement
30,40,50,40,30 #If cartesian, number of y cells [-]; otherwise, variable array of y-refinement [-] (for spe11c)
5,3,1,2,3,2,4,4,8,4,6,6,4,8,10,30,30,6 #If cartesian, number of z cells [-]; if tensor, variable array of z-refinement; if corner-point, fix array of z-refinement (18 entries)
70 36.12 #Temperature bottom and top rig [C]
300 3e7 0.1 #Datum [m], pressure at the datum [Pa], and multiplier for the permeability in the z direction [-]
1e-9 2e-8 #Diffusion (in liquid and gas) [m^2/s]
8.5e-1 2500 #Rock specific heat and density (for spe11b/c)
0 5e4 1 #Added pore volume on top boundary (for spe11a [if 0, free flow bc]), pore volume on lateral boundaries, and width of buffer cell [m] (for spe11b/c)
150 10 #Elevation of the parabola and back [m] (for spe11c)

Configuration file (.txt)

```
"""Set the saturation functions"""
(max(0, (s_w - swi) / (1 - swi))) ** 1.5                                                    #Wetting rel perm saturation function [-]
(max(0, (1 - s_w - sni) / (1 - sni))) ** 1.5                                                #Non-wetting rel perm saturation function [-]
penmax * math.erf(pen * ((s_w-swi) / (1.-swi)) ** (-(1.0 / 1.5)) * math.pi**0.5 / (penmax * 2))  #Capillary pressure saturation function [Pa]
(np.exp(np.flip(np.linspace(0, 5.0, npoints))) - 1) / (np.exp(5.0) - 1)                     #Points to evaluate the saturation functions (s_w) [-]

"""Properties sat functions"""
"""swi [-], sni [-], pen [Pa], penmax [Pa], npoints [-]"""
SWI1 0.32 SNI1 0.1 PEN1 193531.39 PENMAX1 3e7 NPOINTS1 1000
SWI2 0.14 SNI2 0.1 PEN2   8654.99 PENMAX2 3e7 NPOINTS2 1000
SWI3 0.12 SNI3 0.1 PEN3   6120.00 PENMAX3 3e7 NPOINTS3 1000
SWI4 0.12 SNI4 0.1 PEN4   3870.63 PENMAX4 3e7 NPOINTS4 1000
SWI5 0.12 SNI5 0.1 PEN5   3060.00 PENMAX5 3e7 NPOINTS5 1000
SWI6 0.10 SNI6 0.1 PEN6   2560.18 PENMAX6 3e7 NPOINTS6 1000
SWI7      0 SNI7     0 PEN7            0 PENMAX7 3e7 NPOINTS7 2

"""Properties rock"""
"""K [mD], phi [-], disp [m] (dispersion requires a Flow version newer than 17-11-2023), thconr [W m-1 K-1]"""
PERM1 0.10132 PORO1 0.10 DISP1 10 THCONR1 1.90
PERM2 101.324 PORO2 0.20 DISP2 10 THCONR2 1.25
PERM3 202.650 PORO3 0.20 DISP3 10 THCONR3 1.25
PERM4 506.625 PORO4 0.20 DISP4 10 THCONR4 1.25
PERM5 1013.25 PORO5 0.25 DISP5 10 THCONR5 0.92
PERM6 2026.50 PORO6 0.35 DISP6 10 THCONR6 0.26
PERM7    1e-5 PORO7 1e-6 DISP7  0 THCONR7 2.00

"""Wells radius and position"""
"""radius (0 to use the SOURCE keyword instead of well keywords, this requires a Flow version newer than 23-01-2024), x, y, and z position [m] (final positions as well for spe11c)"""
0 2700. 1000. 300. 2700. 4000. 300. #Well 1
0 5100. 1000. 700. 5100. 4000. 700. #Well 2

"""Define the injection values ([hours] for spe11a; [years] for spe11b/c)"""
"""injection time, time step size to write results, maximum solver time step, injected fluid (0 water, 1 co2) (well1), injection rate [kg/s] (well1), temperature [C] (well1), injected fluid (0 water, 1 co2) (well2), ..."""
  25    5 0.1 1 50 10 1  0 10
  25    5 0.1 1 50 10 1 50 10
  50   25 0.1 1  0 10 1  0 10
 400   50 0.1 1  0 10 1  0 10
 500  100 0.1 1  0 10 1  0 10
```

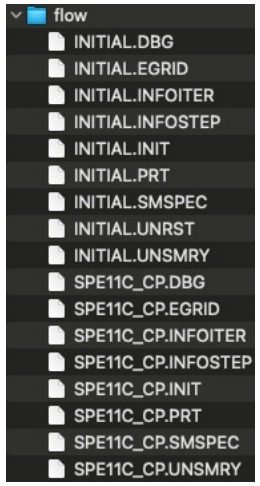
src folder content

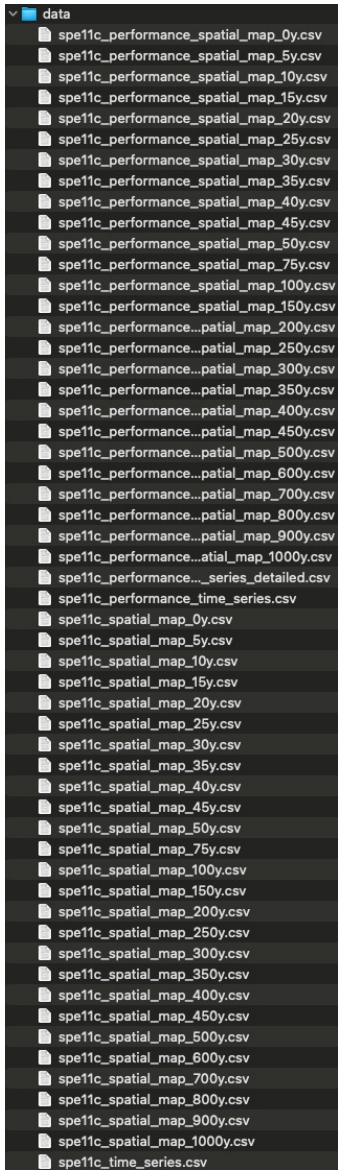


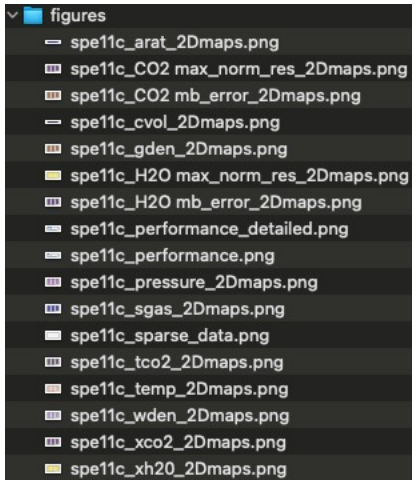
Name	Size	Kind
pyopmspe11	--	Folder
__init__.py	Zero bytes	Python script
core	--	Folder
__init__.py	Zero bytes	Python script
pyopmspe11.py	5 KB	Python script
reference_mesh	--	Folder
facies_coordinates.geo	58 KB	Gmsh...etry File
facies_coordinates.msh	11.5 MB	Gmsh Mesh File
lines_coordinates.geo	23 KB	Gmsh...etry File
templates	--	Folder
co2	--	Folder
spe11a.mako	4 KB	Visual...ocument
spe11b.mako	4 KB	Visual...ocument
spe11c.mako	4 KB	Visual...ocument
common	--	Folder
deck_initial.mako	1 KB	Visual...ocument
grid_corner.mako	3 KB	Visual...ocument
grid_initial.mako	1 KB	Visual...ocument
grid_tensor.mako	3 KB	Visual...ocument
saturation_functions.mako	2 KB	Visual...ocument
utils	--	Folder
__init__.py	Zero bytes	Python script
inputvalues.py	7 KB	Python script
mapproperties.py	27 KB	Python script
runs.py	2 KB	Python script
writefile.py	5 KB	Python script
visualization	--	Folder
data.py	19 KB	Python script
plotting.py	14 KB	Python script

Output folder

```
pyopmspe11 -i spe11c_cp.txt -o spe11c_cp -m all -g all -r 168,100,120 -t 0,5,10,15,20,25,30,35,40,45,50,75,100,150,200,250,300,350,400,450,500,600,700,800,900,1000
```

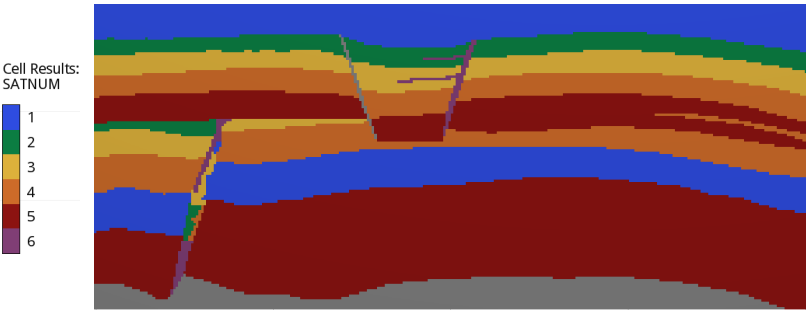




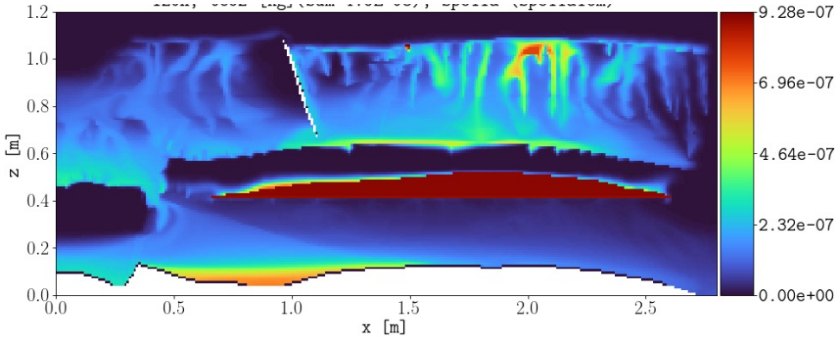


SPE11A

Cartesian 1 cm

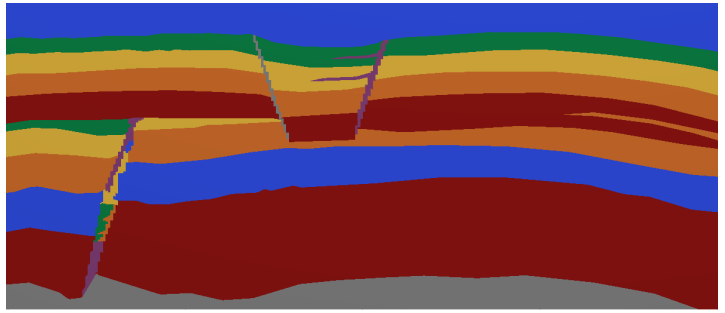


No active cells: 31034

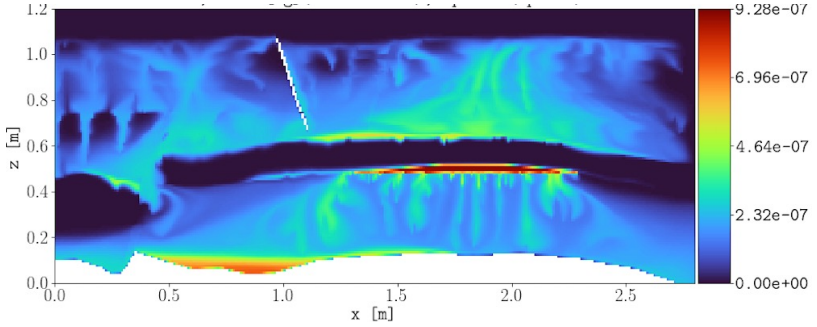


Simulation time: 20 minutes

corner point 1cm

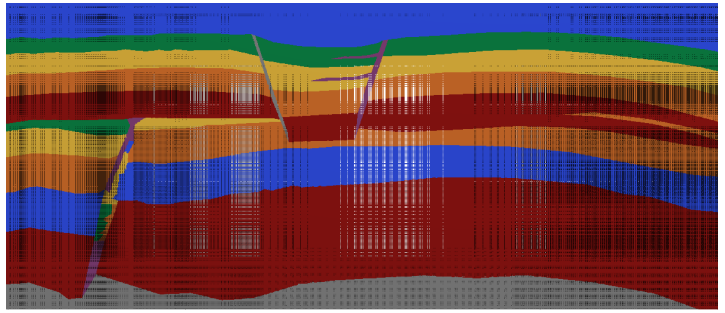


No active cells: 51425



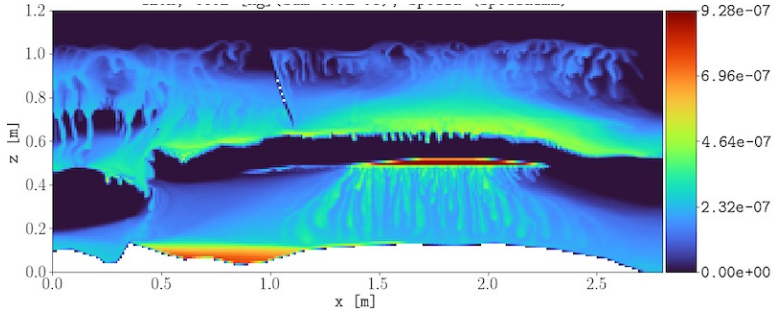
Simulation time: ca. 45 minutes

Cartesian 1 mm



*Visualization issues with ResInsight at this grid size

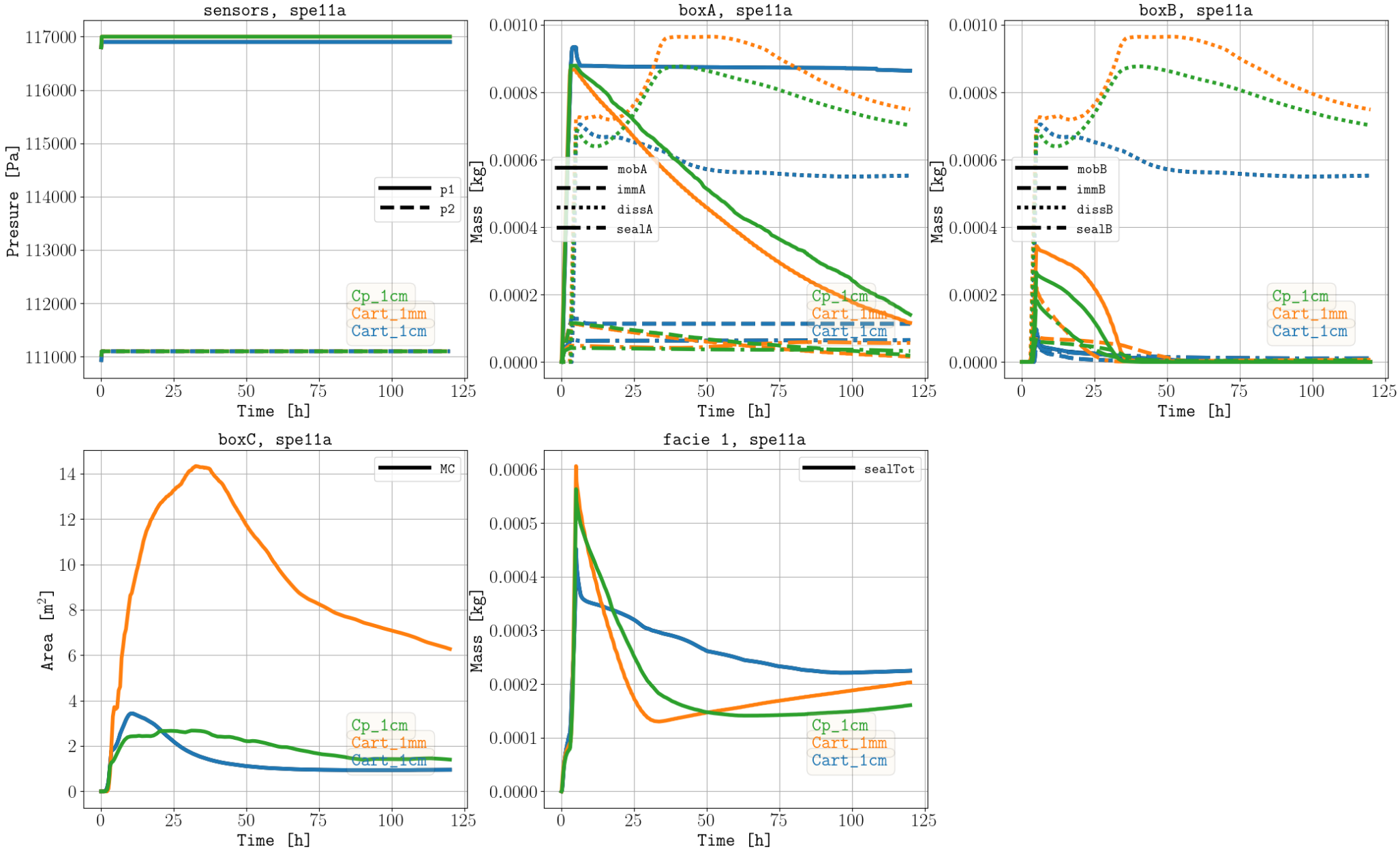
No active cells: 3103039



Simulation time: ca. 18 days

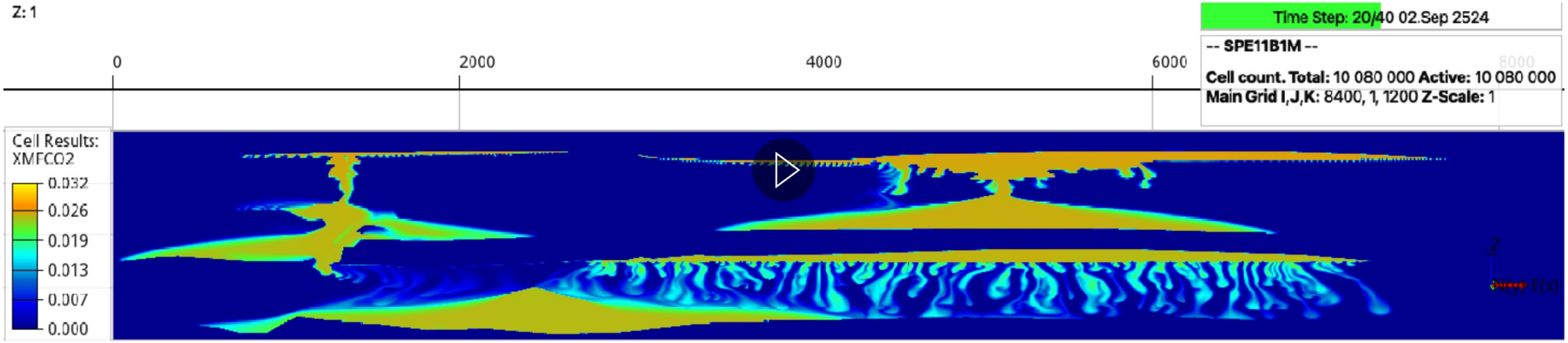
SPE11A

pyopmspe11 -c spe11a

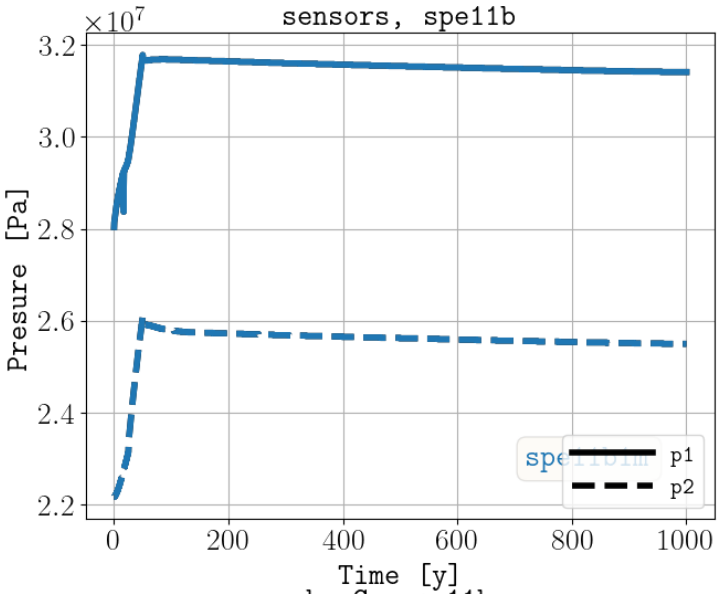


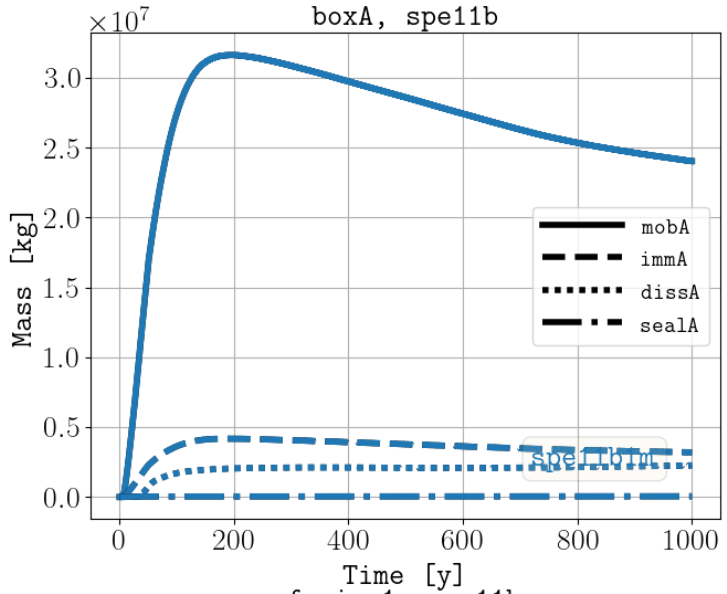
SPE11B

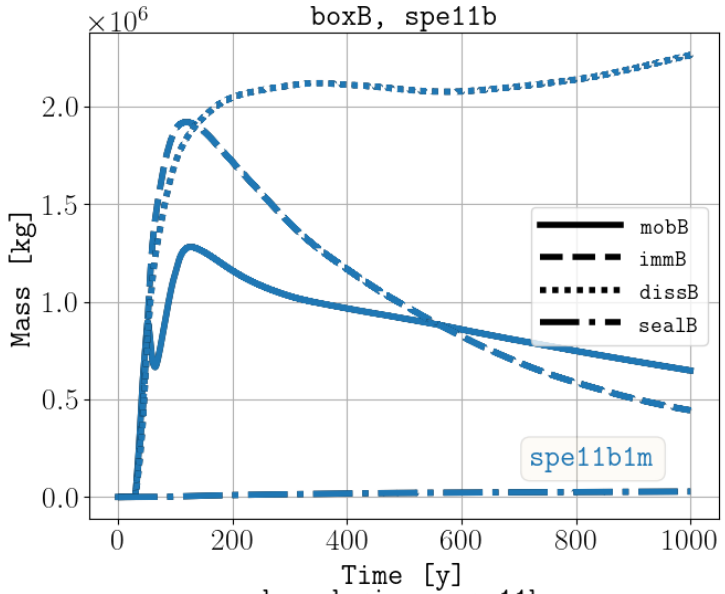
Simulation time: ca. 8 days (mpirun -np 68, max tstep 3.65 days)

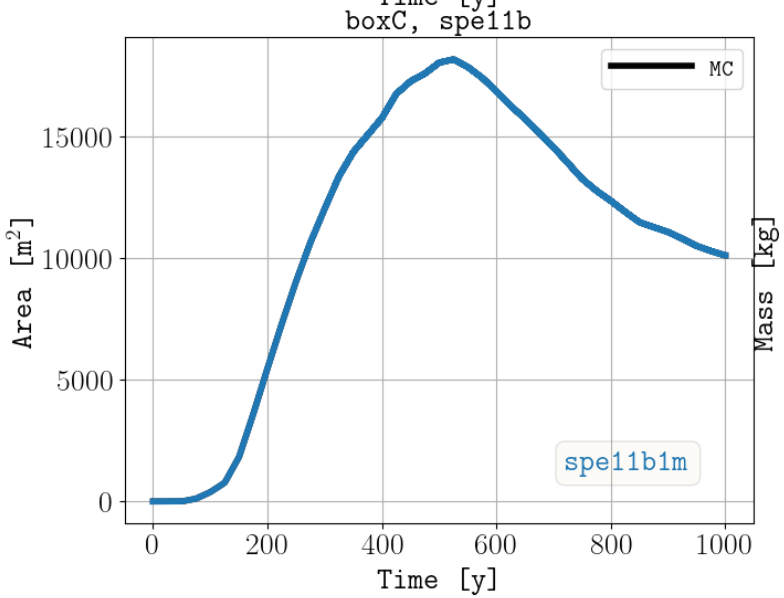


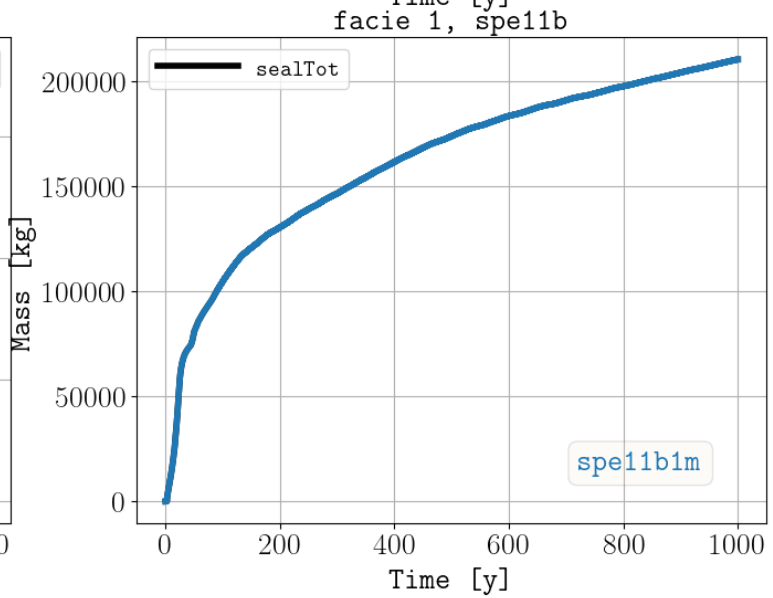
SPE11B

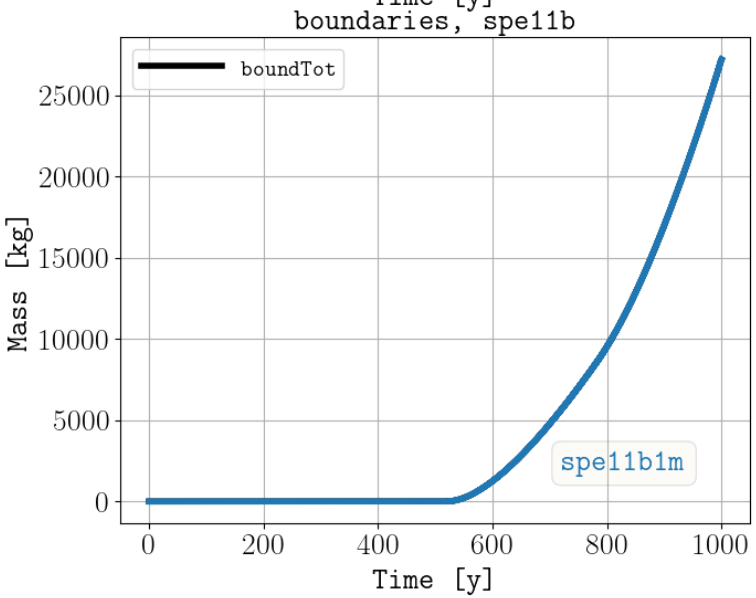

sensors, spe11b
×10^7
Presure [Pa]
Time [y]
p1
p2
spe11b1m


boxA, spe11b
×10^7
Mass [kg]
Time [y]
mobA
immA
dissA
sealA
spe11b1m

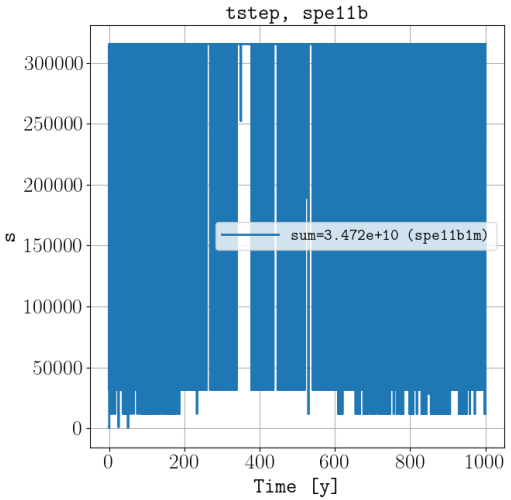

boxB, spe11b
×10^6
Mass [kg]
Time [y]
mobB
immB
dissB
sealB
spe11b1m

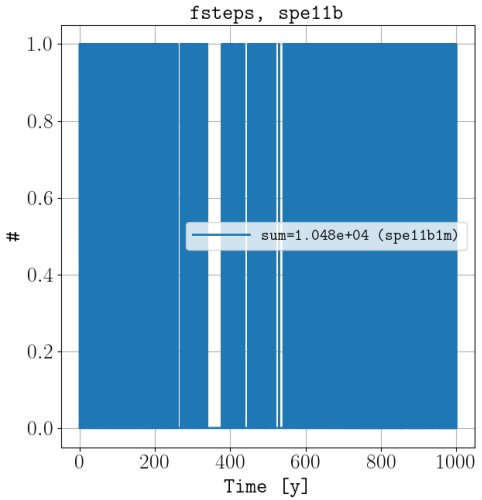

boxC, spe11b
Area [m^2]
Time [y]
MC
spe11b1m

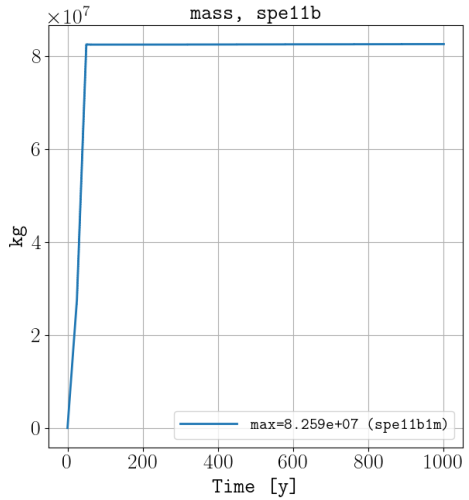

facie 1, spe11b
Mass [kg]
Time [y]
sealTot
spe11b1m

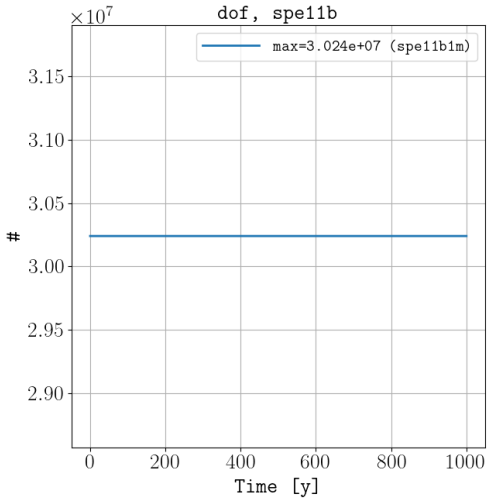

boundaries, spe11b
Mass [kg]
Time [y]
boundTot
spe11b1m

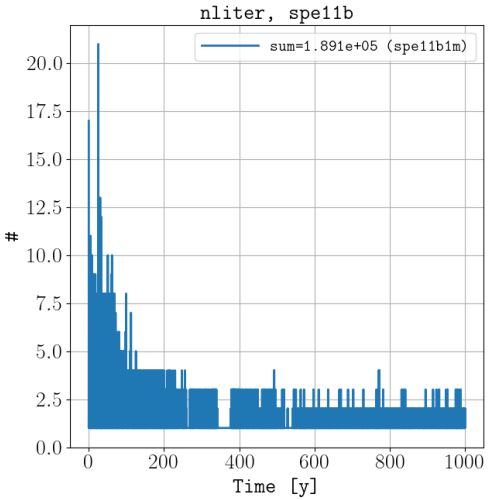
SPE11B

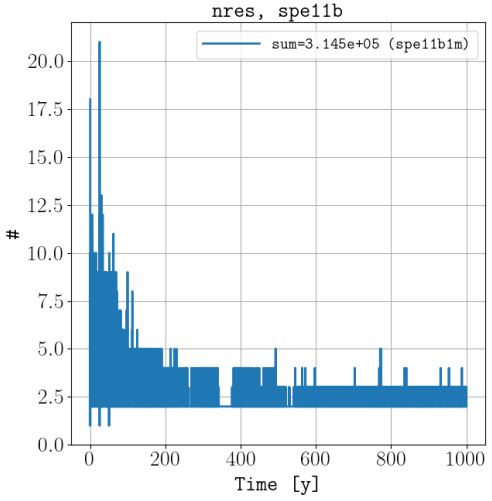

tstep, spe11b
sum=3.472e+10 (spe11b1m)
s
Time [y]

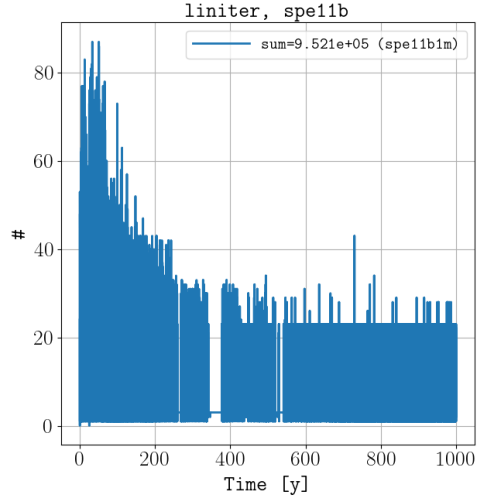

fsteps, spe11b
sum=1.048e+04 (spe11b1m)
#
Time [y]

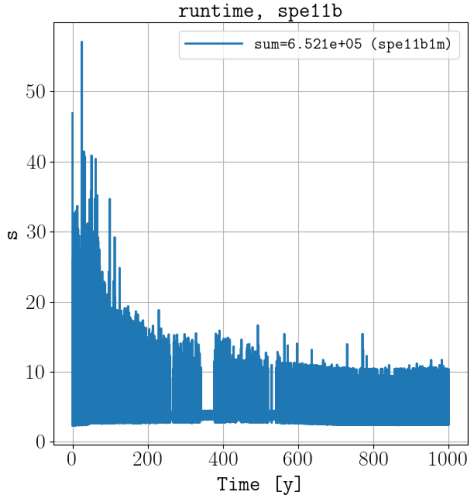

mass, spe11b
max=8.259e+07 (spe11b1m)
kg
Time [y]

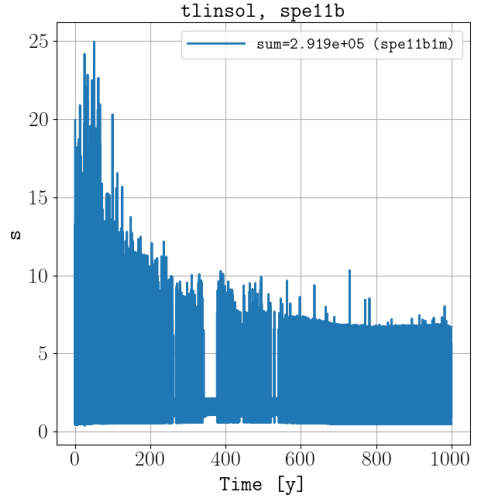

dof, spe11b
max=3.024e+07 (spe11b1m)
#
Time [y]


nliter, spe11b
sum=1.891e+05 (spe11b1m)
#
Time [y]


nres, spe11b
sum=3.145e+05 (spe11b1m)
#
Time [y]


liniter, spe11b
sum=9.521e+05 (spe11b1m)
#
Time [y]


runtime, spe11b
sum=6.521e+05 (spe11b1m)
s
Time [y]

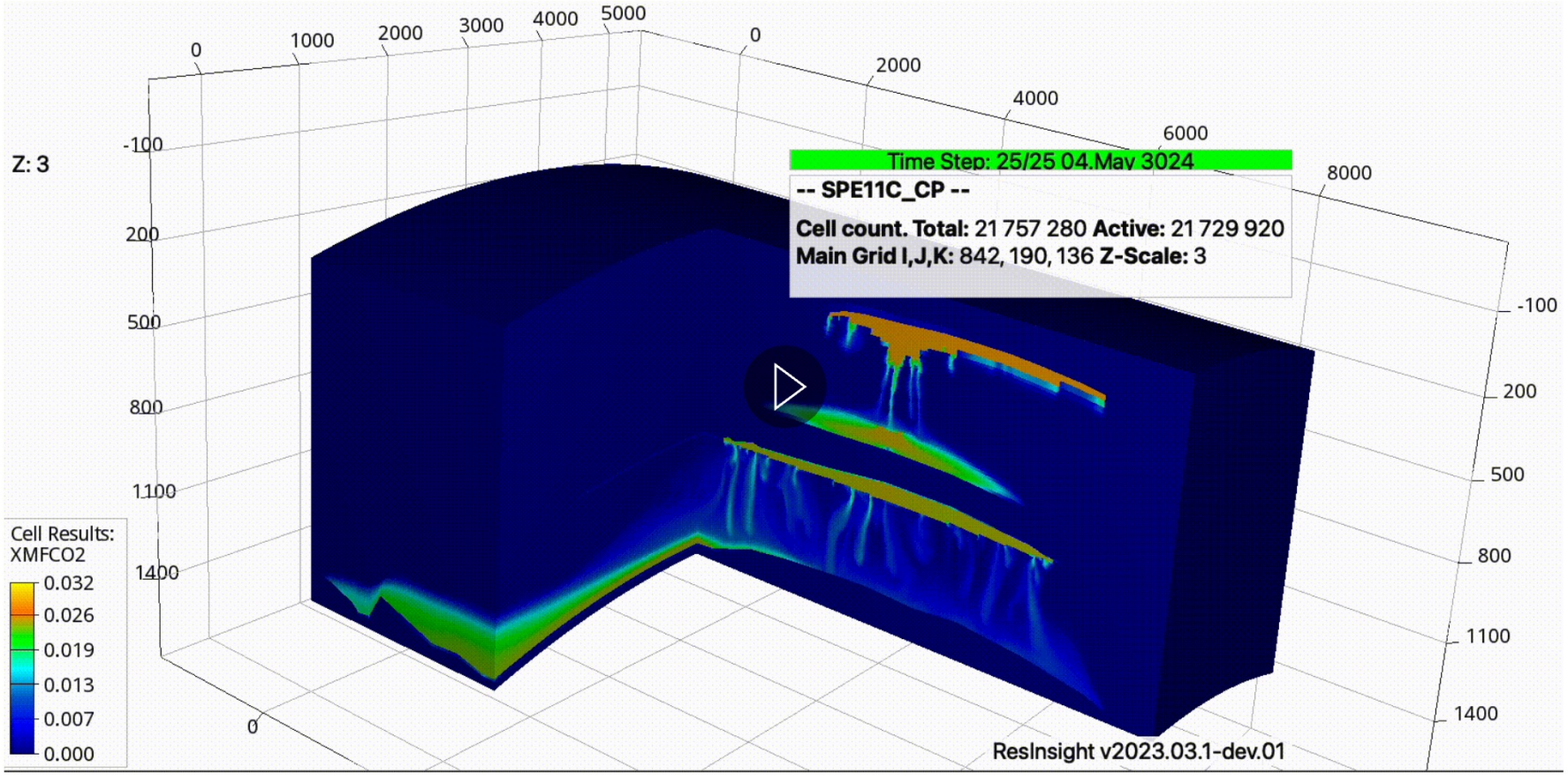

tlinsol, spe11b
sum=2.919e+05 (spe11b1m)
s
Time [y]

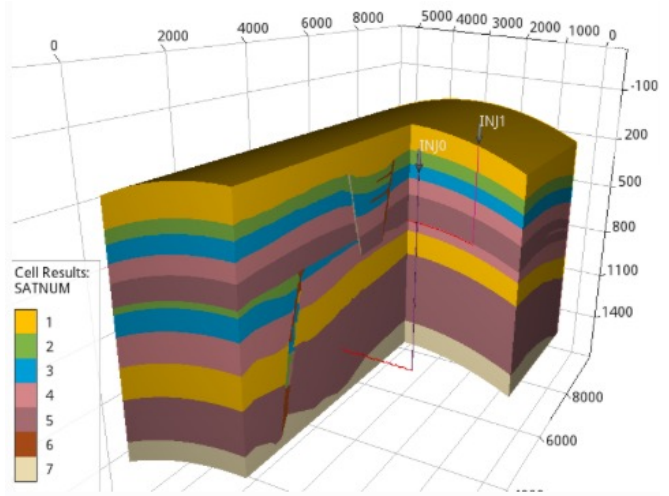
SPE11C

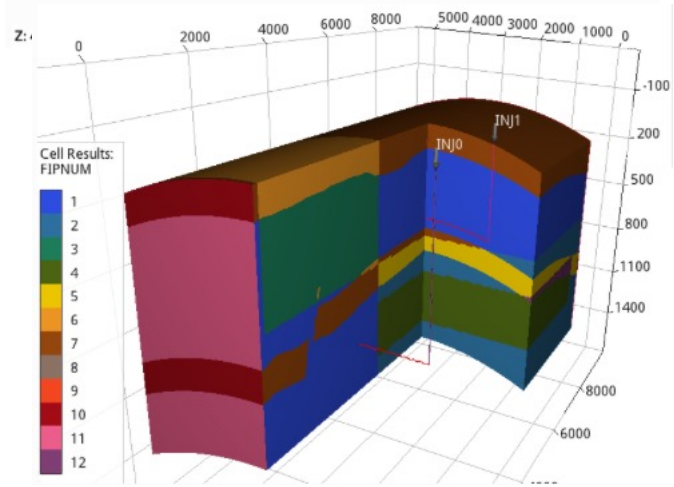
Time to generate the input files: ca. 25 minutes (ca. 8 hours for ca. 160 M cells)

Simulation time: ca. 8 days (mpirun -np 71, max tstep 36.5 days)

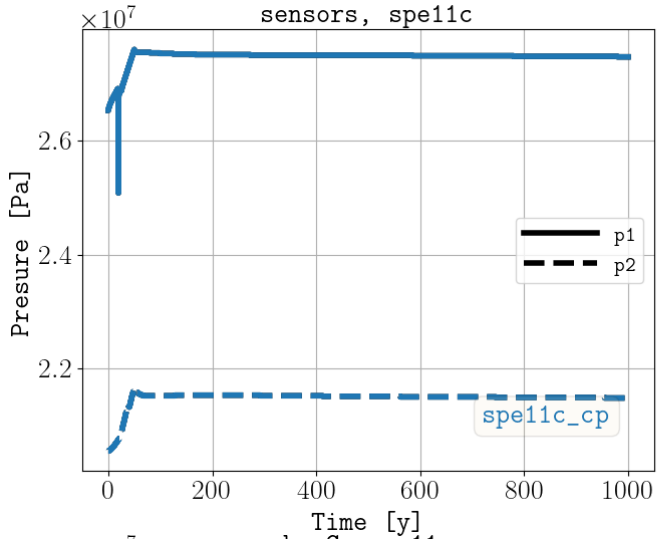
Time to postprocess the data: ca. 5 hours

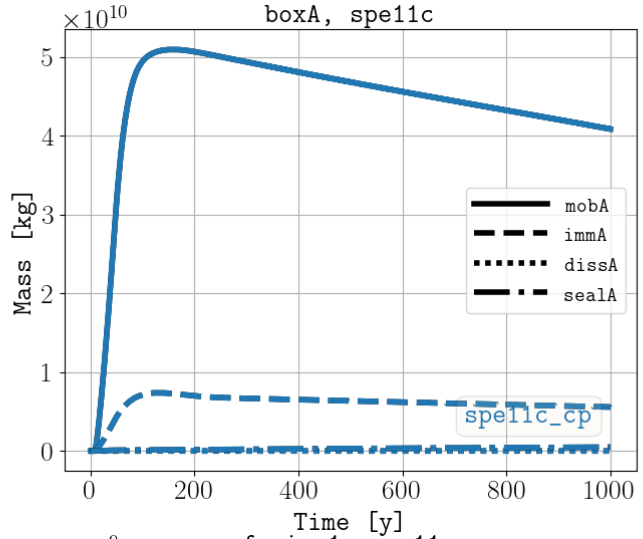


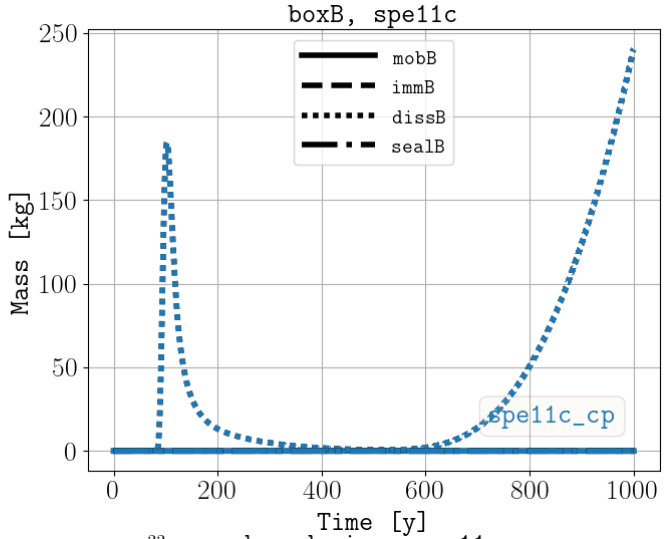


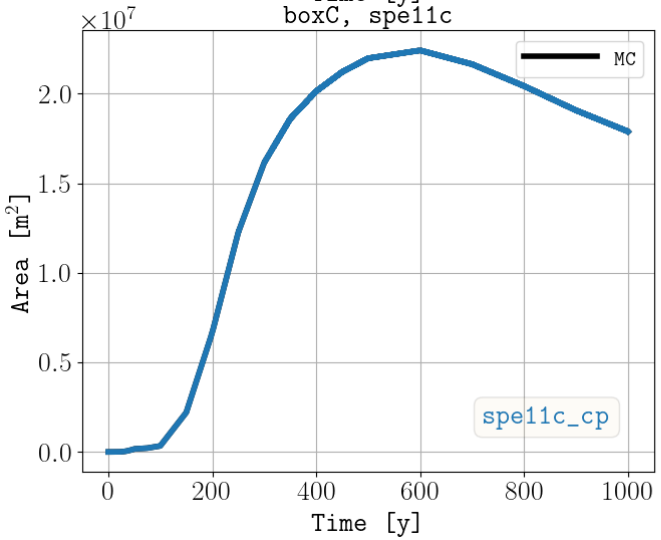


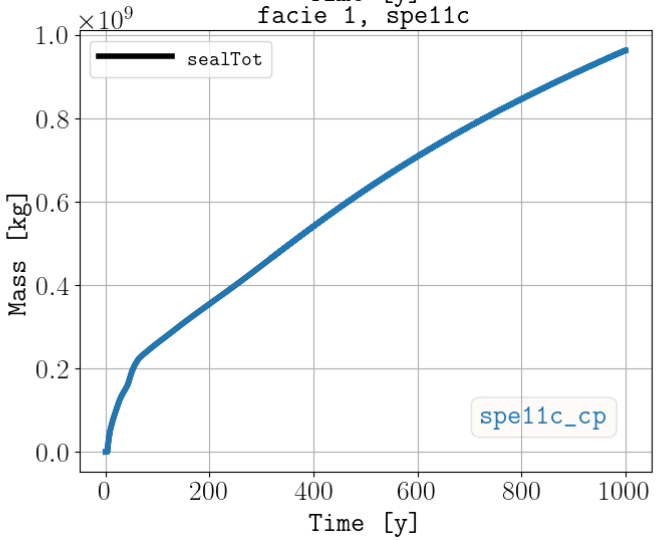
SPE11C

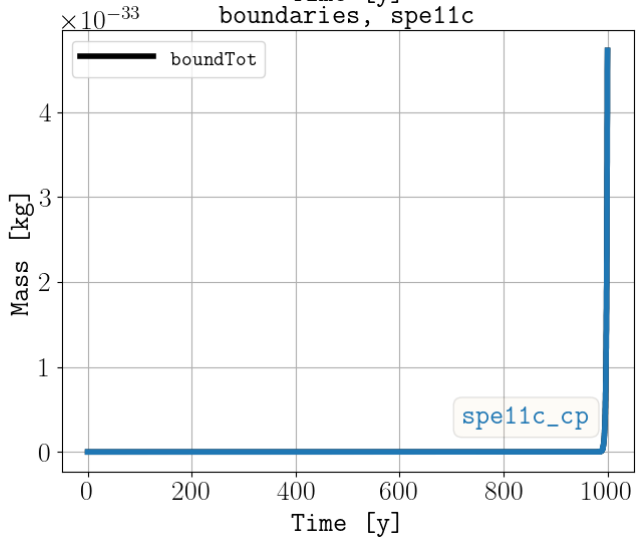

sensors, spe11c
×10^7
Presure [Pa]
Time [y]
p1
p2
spe11c_cp


boxA, spe11c
×10^10
Mass [kg]
Time [y]
mobA
immA
dissA
sealA
spe11c_cp

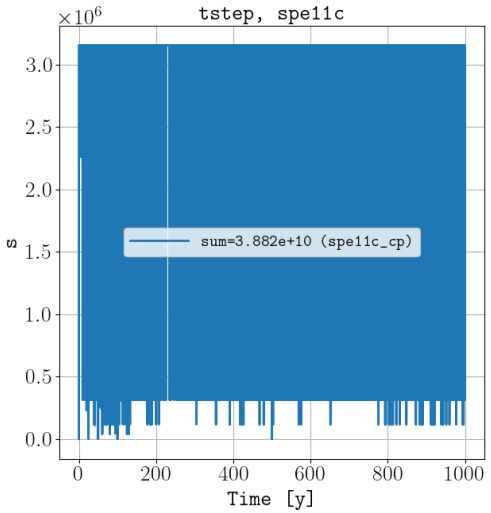

boxB, spe11c
Mass [kg]
Time [y]
mobB
immB
dissB
sealB
spe11c_cp

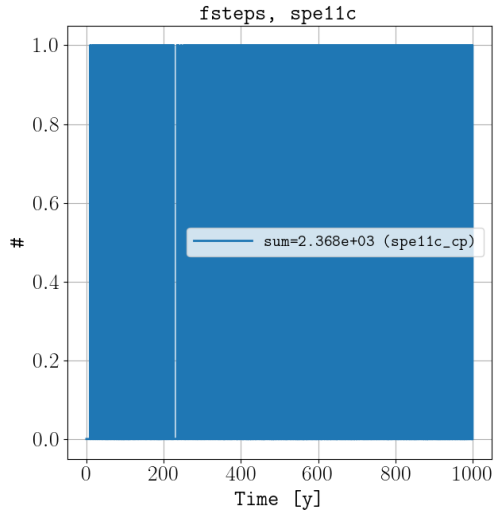

boxC, spe11c
×10^7
Area [m^2]
Time [y]
MC
spe11c_cp

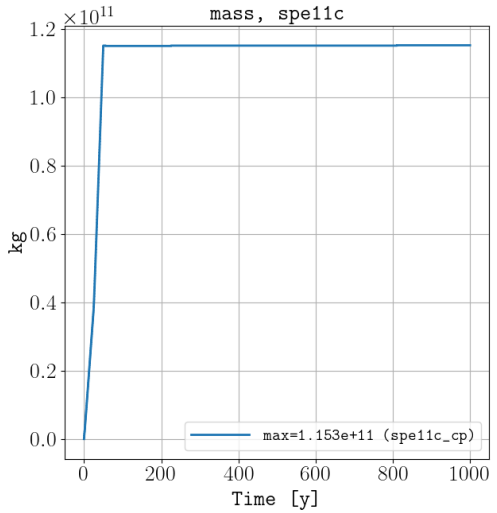

facie 1, spe11c
×10^9
Mass [kg]
Time [y]
sealTot
spe11c_cp

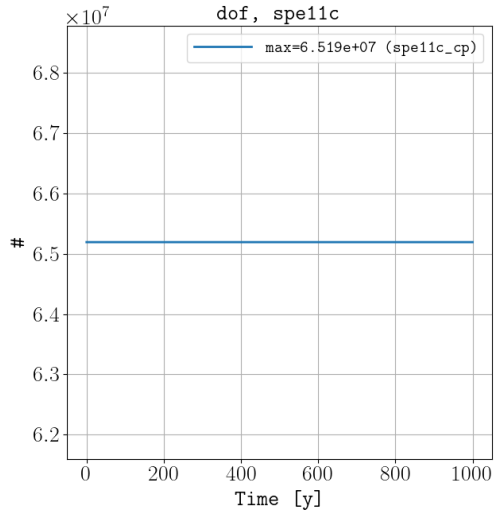

boundaries, spe11c
×10^−33
Mass [kg]
Time [y]
boundTot
spe11c_cp

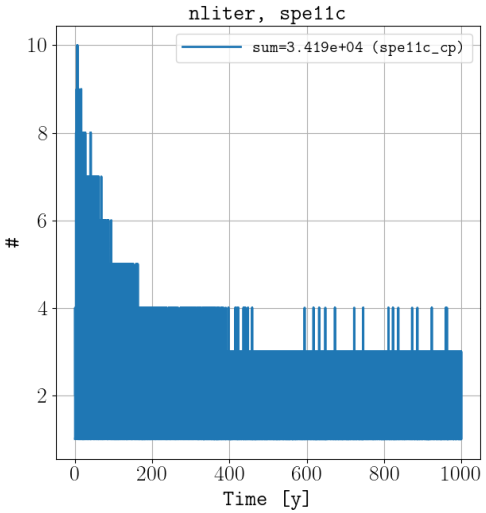
SPE11C

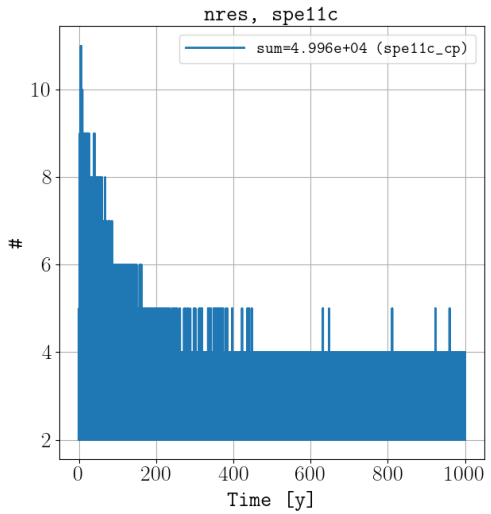

tstep, spe11c
$\times 10^6$
s
sum=3.882e+10 (spe11c_cp)
Time [y]

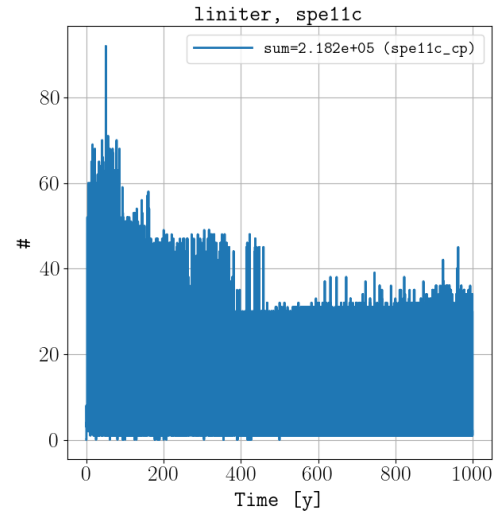

fsteps, spe11c
#
sum=2.368e+03 (spe11c_cp)
Time [y]

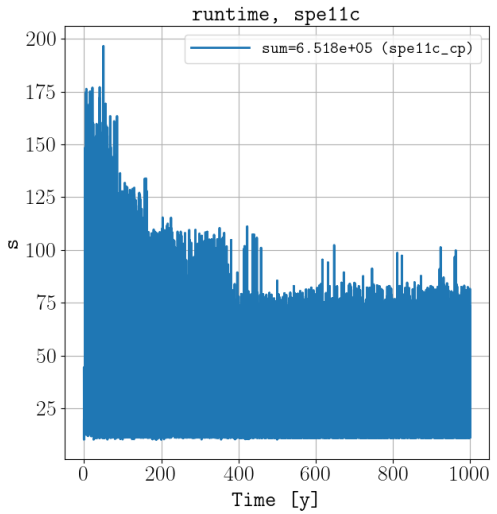

mass, spe11c
$\times 10^{11}$
kg
max=1.153e+11 (spe11c_cp)
Time [y]

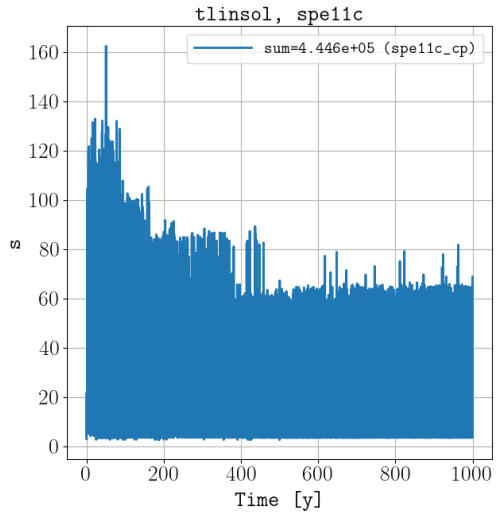

dof, spe11c
$\times 10^7$
#
max=6.519e+07 (spe11c_cp)
Time [y]


nliter, spe11c
#
sum=3.419e+04 (spe11c_cp)
Time [y]


nres, spe11c
#
sum=4.996e+04 (spe11c_cp)
Time [y]

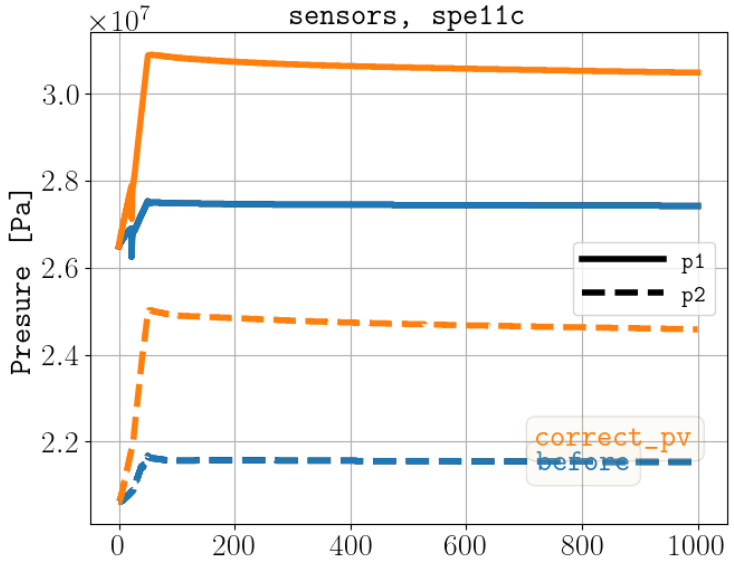

liniter, spe11c
#
sum=2.182e+05 (spe11c_cp)
Time [y]

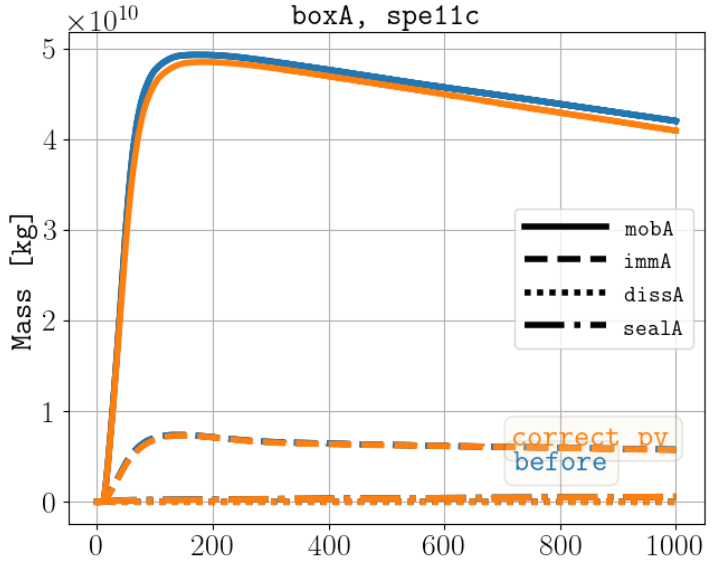

runtime, spe11c
s
sum=6.518e+05 (spe11c_cp)
Time [y]

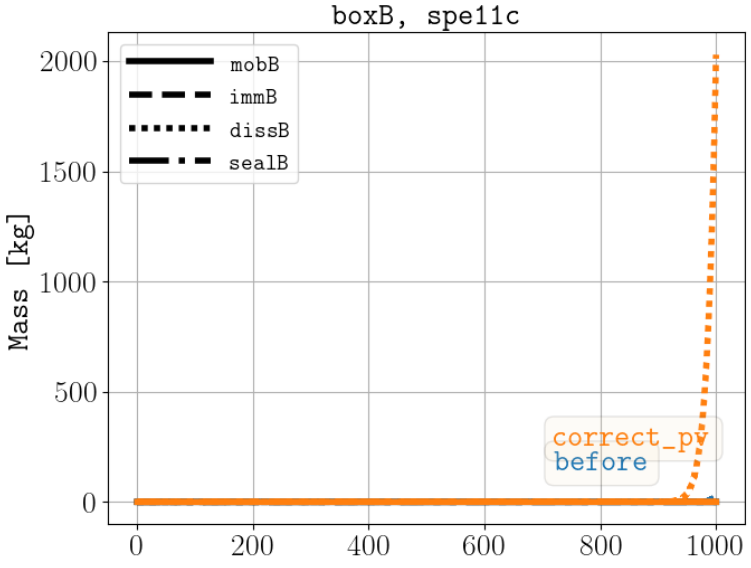

tlinsol, spe11c
s
sum=4.446e+05 (spe11c_cp)
Time [y]

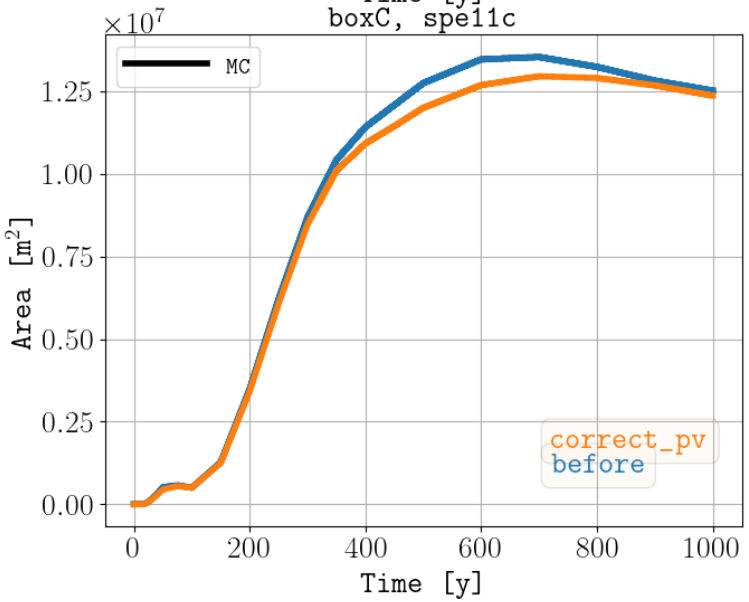
Correct PV on boundary

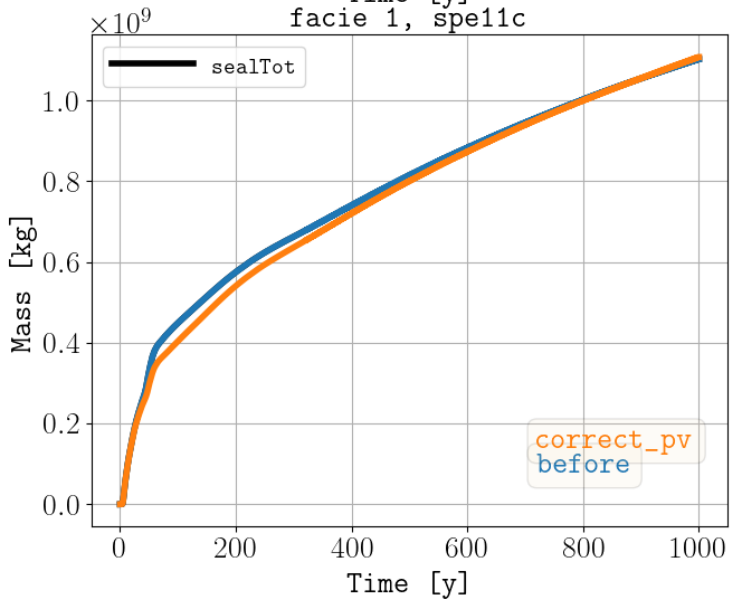
spe11c using the report grid

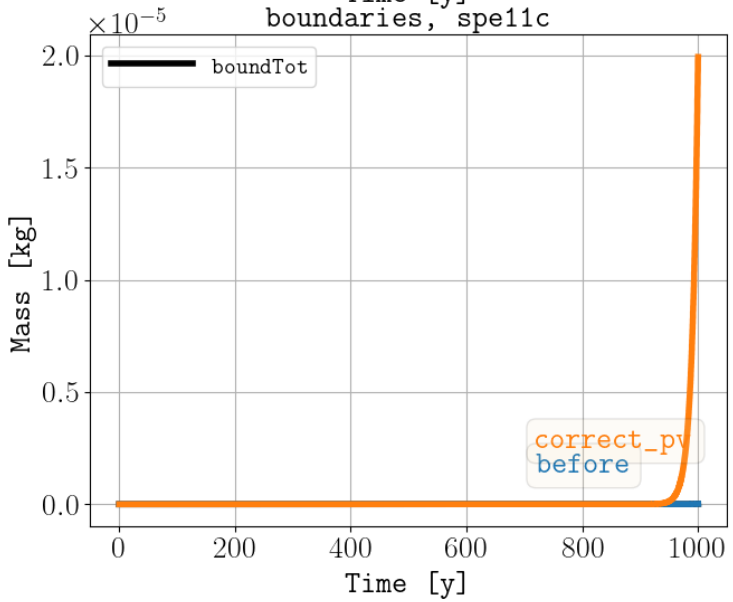












Current work

- Update b/c results in the Docs with the corrected PV
- Output in summary $M(t) \equiv \int_{\text{Box } C} \left| \nabla \left(\frac{\chi_c^w}{\chi_{c,\max}^w} \right) \right| dV.$
- A posteriori estimate of the error
- Performance speed up/tuning
- Update convective mixing
- Run on supercomputer
- Setup adaptive grid using Alugrid
- Thermal implementation

Acknowledgement

The pyopmspe11 package is being funded by the HPC Simulation Software for the Gigatonne Storage Challenge project [project no. 622059] and Center for Sustainable Subsurface Resources (CSSR) [project no. 331841].

Centre for Sustainable Subsurface Resources

NORCE