SPE11 scripts and framework

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

- Overview of the SPE11 benchmark
- pyopmspe11 repository
- 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 CO₂ storage operations in geological settings, specifically developing simulations of realistic complexity.

The CSP contains three versions:

Version 11A is a 2D geometry at the laboratory scale, inspired by a recent CO₂ 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

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)

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)
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:

```bash
# 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

```
1 mako
2 matplotlib
3 pandas
4 Pyarrow
5 resdata
6 rtree
7 scipy
8 shapely
```

dev-requirements.txt

```
1 black
2 mypy
3 pylint
4 pytest-cov
5 sphinx
6 sphinx-rtd-theme
```
Running pyopmspe11

pyopmspe11 -i some_input.txt -o some_output_folder

pyopmspe11 --help


Main script to run the spells 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 (spell [h]; spellib/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 (spell [h]; spellib/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)
5,3,1,2,3,2,4,4,8,6,4,6,4,8,10,30,30,6  
#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)
****Set the saturation functions****

\[
\text{Wetting rel perm saturation function [-]} \quad \text{(max(0, (s_w - swi) / (1 - swi)) ** 1.5)}
\]

\[
\text{Non-wetting rel perm saturation function [-]} \quad \text{(max(0, (1 - s_w - sni) / (1 - sni)) ** 1.5)}
\]

\[
\text{Capillary pressure saturation function [Pa]} \quad \text{(penmax * math.erf(\frac{\text{pen} \times (s_w - swi)}{\sqrt{1 - swi}}) \times \frac{\pi}{\text{penmax} \times 2})}
\]

\[
\text{Points to evaluate the saturation functions (s_w [-])}
\]

****Properties sat functions****

\[
\text{SWI1: 0.32, SNI1: 0.1, PEN1: 193531.39, PENMAX1: 3e7, NPOINTS1: 1000}
\]

\[
\text{SWI2: 0.14, SNI2: 0.1, PEN2: 8654.99, PENMAX2: 3e7, NPOINTS2: 1000}
\]

\[
\text{SWI3: 0.12, SNI3: 0.1, PEN3: 6120.00, PENMAX3: 3e7, NPOINTS3: 1000}
\]

\[
\text{SWI4: 0.12, SNI4: 0.1, PEN4: 3870.63, PENMAX4: 3e7, NPOINTS4: 1000}
\]

\[
\text{SWI5: 0.12, SNI5: 0.1, PEN5: 3060.00, PENMAX5: 3e7, NPOINTS5: 1000}
\]

\[
\text{SWI6: 0.10, SNI6: 0.1, PEN6: 2560.18, PENMAX6: 3e7, NPOINTS6: 1000}
\]

\[
\text{SWI7: 0, SNI7: 0, PEN7: 0, PENMAX7: 3e7, NPOINTS7: 2}
\]

****Properties rock****

\[
\text{PERM1: 0.10132, PORO1: 0.10, DISP1: 10, THCONR1: 1.90}
\]

\[
\text{PERM2: 101.324, PORO2: 0.20, DISP2: 10, THCONR2: 1.25}
\]

\[
\text{PERM3: 202.650, PORO3: 0.20, DISP3: 10, THCONR3: 1.25}
\]

\[
\text{PERM4: 506.625, PORO4: 0.20, DISP4: 10, THCONR4: 1.25}
\]

\[
\text{PERM5: 1013.25, PORO5: 0.25, DISP5: 10, THCONR5: 0.92}
\]

\[
\text{PERM6: 2026.50, PORO6: 0.35, DISP6: 10, THCONR6: 0.26}
\]

\[
\text{PERM7: 1e-5, PORO7: 1e-6, DISP7: 0, THCONR7: 2.00}
\]

****Wells radius and position****

\[
\text{Well 1: [2700.00, 1000.00, 300.00, 2700.00, 4000.00, 300.00]}
\]

\[
\text{Well 2: [5100.00, 1000.00, 700.00, 5100.00, 4000.00, 700.00]}
\]

****Define the injection values ([hours] for spe11a; [years] for spe11b/c)****

\[
\text{Injection time, time step size to write results, maximum solver time step, injected fluid (0 water, 1 co2) (well1), injection rate [kg/s] (well1), temperature [C] (well1), injected fluid (0 water, 1 co2) (well2), ...}
\]

\[
25 \quad 5 \quad 0 \quad 1 \quad 50 \quad 10 \quad 1 \quad 0 \quad 10
\]

\[
25 \quad 5 \quad 0 \quad 1 \quad 50 \quad 10 \quad 5 \quad 10
\]

\[
50 \quad 25 \quad 0 \quad 1 \quad 0 \quad 10 \quad 1 \quad 0 \quad 10
\]

\[
400 \quad 50 \quad 0 \quad 1 \quad 0 \quad 10 \quad 1 \quad 0 \quad 10
\]

\[
500 \quad 100 \quad 0 \quad 1 \quad 0 \quad 10 \quad 1 \quad 0 \quad 10
\]
src folder content
SPE11A

Cartesian 1 cm

corner point 1cm

Cartesian 1 mm

No active cells: 31034

No active cells: 51425

No active cells: 3103039

Simulation time: 20 minutes

Simulation time: ca. 45 minutes

Visualization issues with ResInsight at this grid size

Simulation time: ca. 18 days
SPE11A

sensors, spella

boxA, spella

boxB, spella

boxC, spella

facie 1, spella
Simulation time: ca. 8 days (mpirun -np 68, max tstep 3.65 days)
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, spellc**
  - Time [y]
  - Pressure [Pa]
  - p1
  - p2

- **boxA, spellc**
  - Time [y]
  - Mass [kg]
  - catA
  - imA
  - disA
  - sealA

- **boxB, spellc**
  - Time [y]
  - Mass [kg]
  - nobB
  - imB
  - disB
  - sealB

- **boxC, spellc**
  - Time [y]
  - Area [m^2]
  - MC

- **facie 1, spellc**
  - Time [y]
  - Mass [kg]
  - seal1

- **boundaries, spellc**
  - Time [y]
  - Mass [kg]
  - bound1

---

19/23
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

\[ M(t) = \int_{\text{Box C}} \nabla \left( \frac{\chi_c}{\chi_{\text{max}}} \right) dV. \]

• 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].