枯竭油气藏中二氧化碳封存的水合物生成评估
Hydrate modelling Workflow for CCS using CMG software

17 June 2025

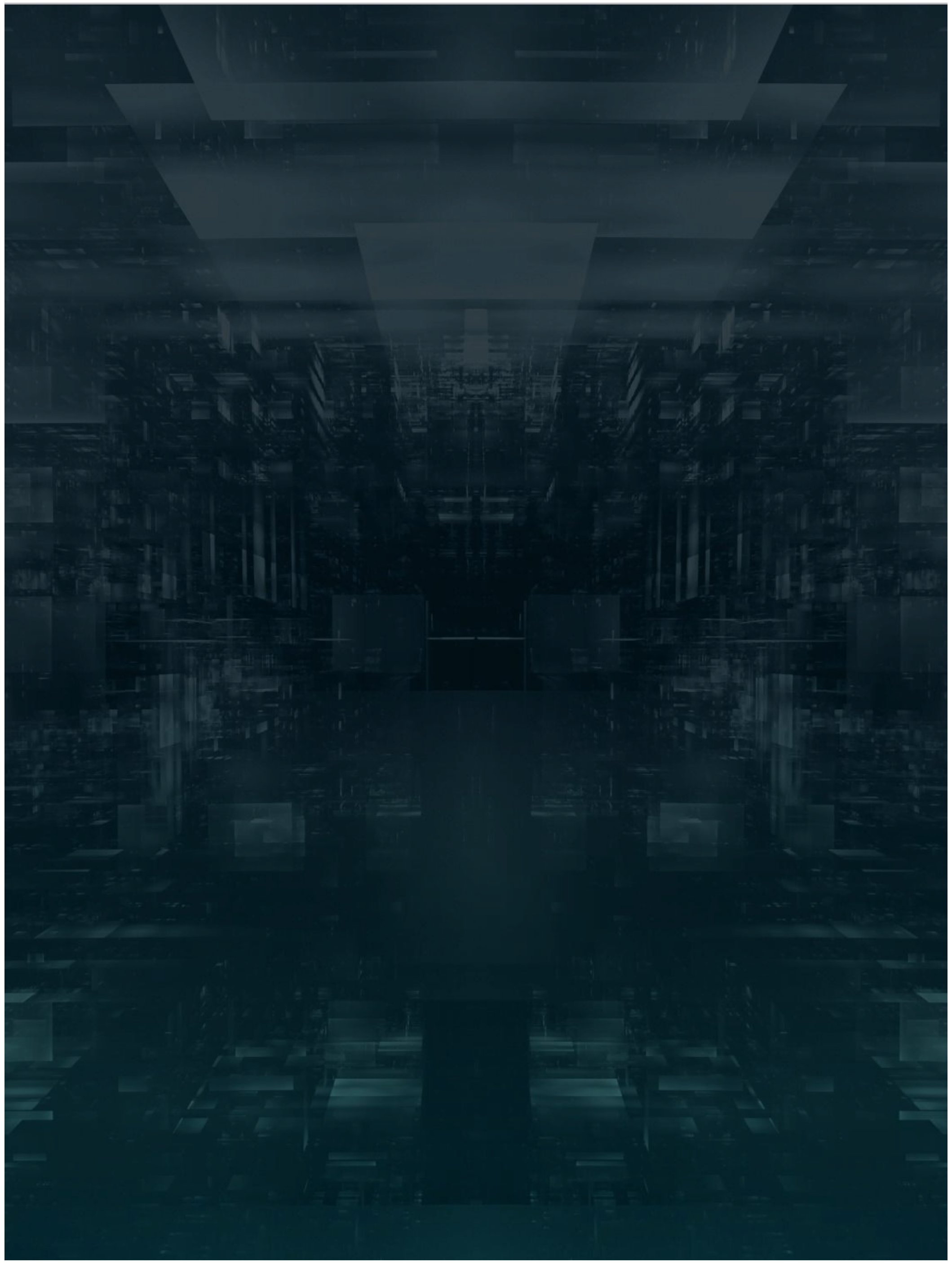
CMG

AGENDA

01 Introduction to Hydrates modelling

02 Case study #1 → Post-processing approach

03 Case study #2 → Real-time coupling approach



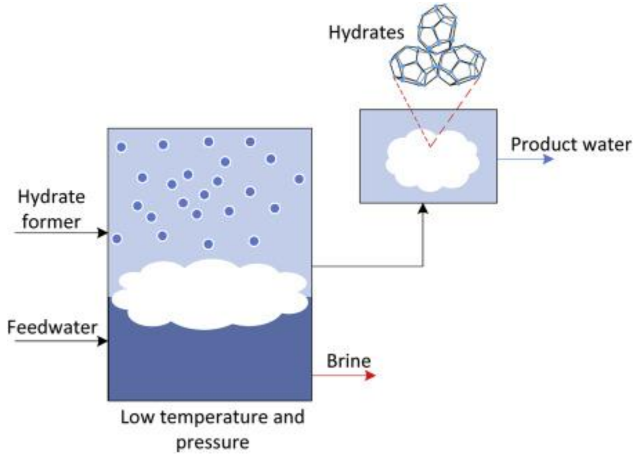
CMG

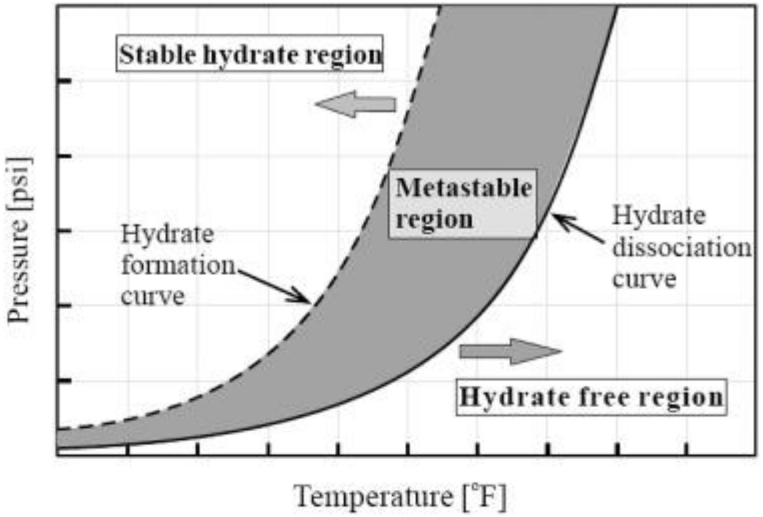
Introduction to Hydrates modelling

CMG

Natural Gas Hydrates

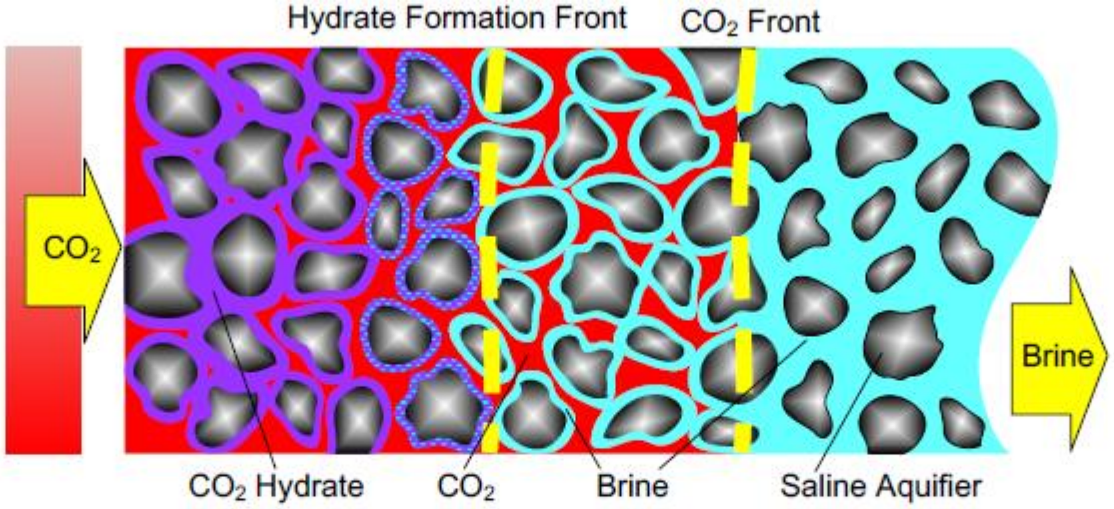
- Natural gas hydrate is formed when methane molecules—the primary component of natural gas—are trapped in a microscopic cage of water molecules under certain pressure and temperature conditions (Katz and Lee, 1990).
- As a rough rule of thumb, methane hydrate will form in a natural gas system if free water is available at a temperature as high as 40°F (4.4°C) and a pressure as low as 170 psig (1200kPa).
- Decreasing temperature and increasing pressure are favorable for hydrate formation

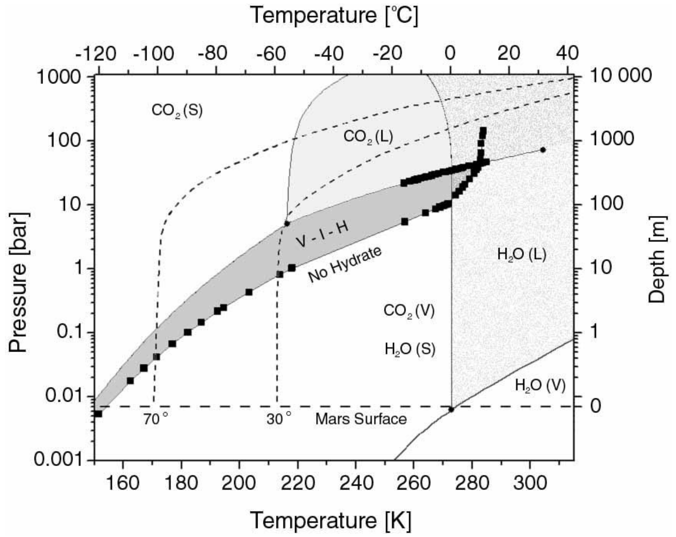




CO2 Hydrates

- Similarly, during CO2 injection into cold reservoir/aquifer, CO2 hydrate may be created.
- Joule-Thompson effects around the injector can significantly cool down the formation and injection fluid.
- Hydrate formation negatively impacts CO2 displacement around the injector and reduces injectivity.
- It may create high pressure pockets around the injector.





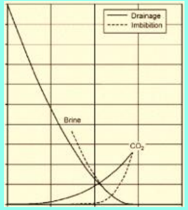
https://doi.org/10.1016/j.egypro.2014.11.326

GEM – Tool to Model CO2 Geological Storage

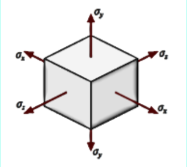
Generalized Equation-of-State Modeling (GEM)

➢ Thermal Multiphase Compositional Reservoir Simulator

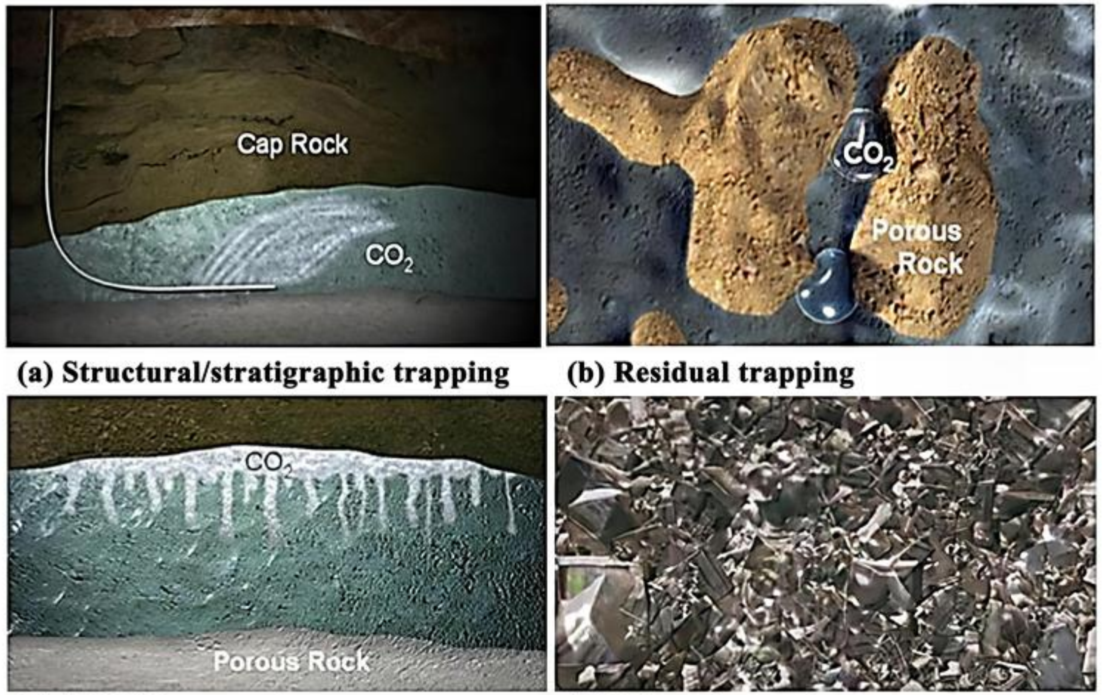
- Gas Solubility in aqueous phase
- H_2O Vaporization
- Geochemistry (Aqueous/Mineral Reaction)
- Joule-Thompson effect included



- Relative permeability hysteresis (Residual gas trapping)



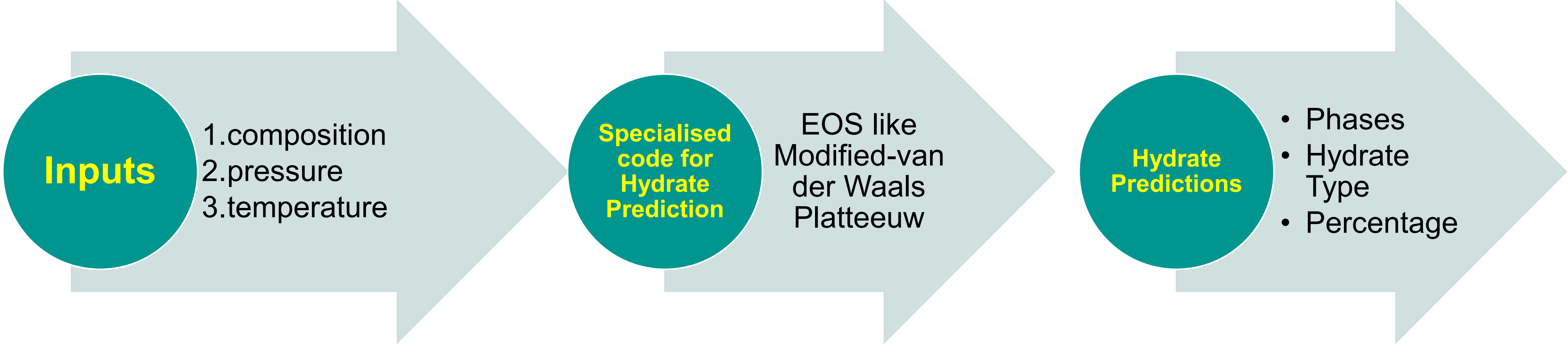
- Geomechanics (cap rock integrity / thermal fracturing / faults reactivation)



(a) Structural/stratigraphic trapping (b) Residual trapping (c) Solubility trapping (d) Mineral trapping

Hydrate Modelling in Chemical Engineering Domain

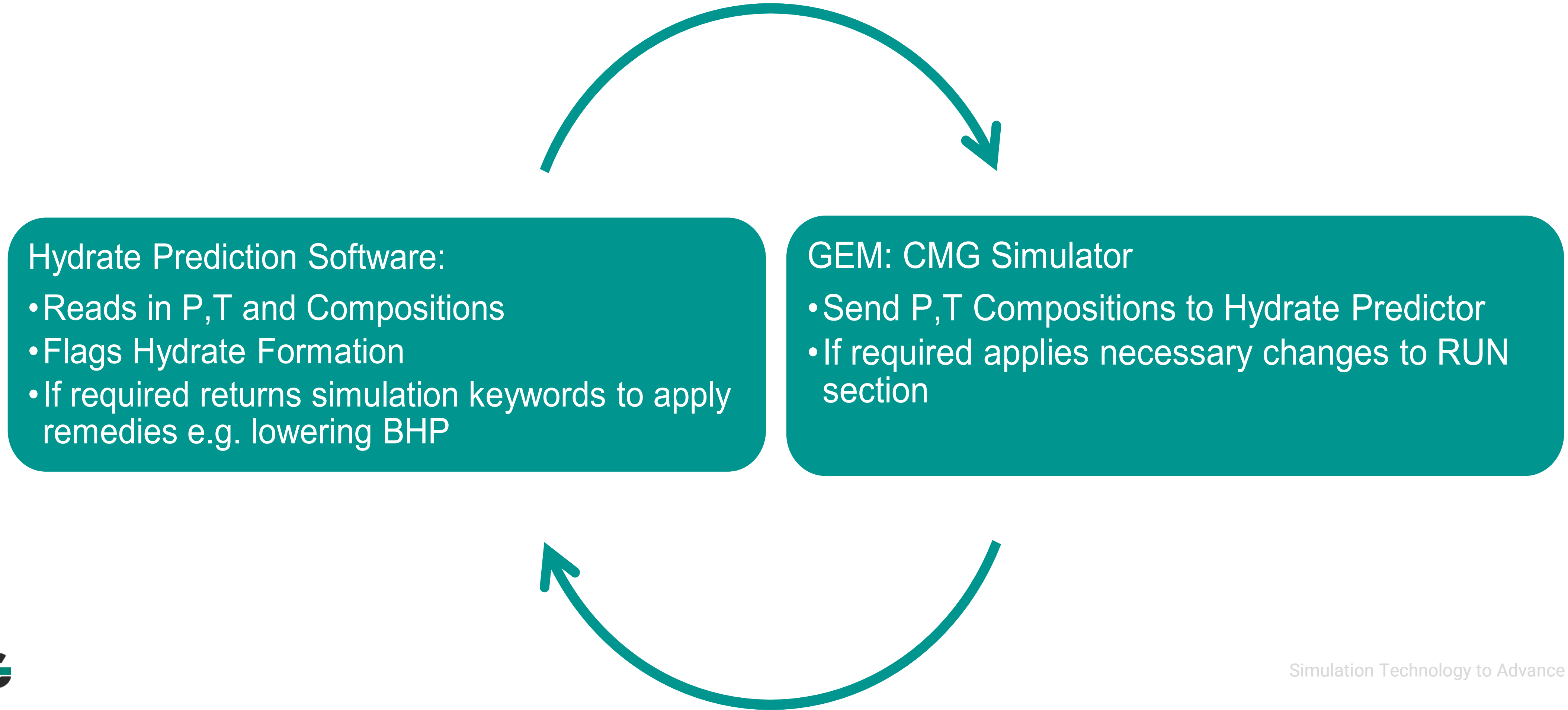
- Comprehensive lab and simulation research have been carried out to predict hydrate formation.
- Mathematical methods for predicting Hydrate are numerically expensive using complex non-standard EOSs:



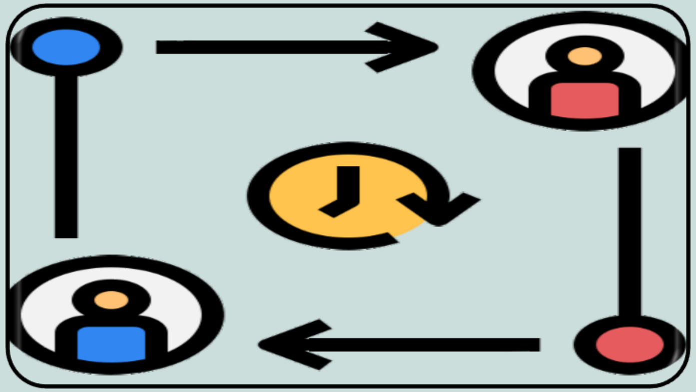
- This doable for a pipeline or a storage tank but not in a subsurface model with potentially millions of blocks.
- To best of our knowledge no Reservoir Simulation has implemented such workflows except CMG.

Outboard Functionality in CMG simulators

- Outboard is a simulator feature that can be used to dynamically control the simulation by software external to the simulator
- In chemical engineering literature, there are open-source codes for predicting hydrate.
- They can be coupled with CMG simulators like **GEM**.



Options for dealing with Hydrate



Post-Processing

- Exclusively a Post-Processing workflow to check
- Use RESULTS to ensure P&T of all blocks are within a safe threshold
- Use Hydrate formation calculator to check the riskiest blocks
- Minimum CPU overhead
- Least accurate

Post-Processing with manual remedies

- After processing manually apply remedies in the simulation to steer away from hydrate formation conditions.
- Modest CPU overhead

Real-time coupling with Hydrate formation calculator

- Live coupling between GEM and Hydrate formation calculator
- GEM automatically takes remedial action when and if necessary
- Highest CPU overhead
- Most accurate

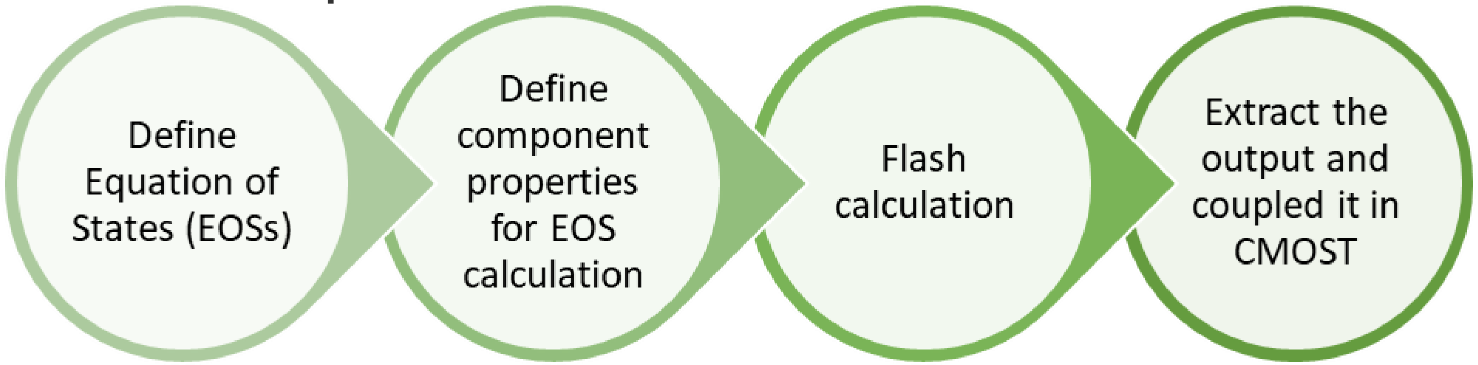
Case study #1

Post-processing for hydrate formation detection using CMOST and python (SPE-215293-MS)

CMG

First Step -> Identify an open-source hydrate EOS

- ✓ The flash calculation algorithm was taken from an open-source library named **hydrateflash**, which is available on github.
- ✓ It consists of set of algorithms to solve for the thermodynamic stability of clathrate hydrates and originates from work done by scientists working on hydrate prediction (Ballard et al., 2002, Jaeger et al., 2003, Ballard et al., 2004).
- ✓ The developers essentially created an open-source version of **CSMGem**, which is an in-house software developed by Colorado School of Mines for the prediction of thermodynamically stable hydrate structures and cage occupancy at given pressure, temperature and composition conditions.



- To determine the partial fugacity of each component given mole fractions, pressurre and temperature.
- To set all relevant properties for each component and to calculate Gibbs Free Energy for each component.
- To predict the amount and presence of hydrate with minimum input
- Command the output so it can read by CMOST
- Put the python file in the CMOST advanced objective function

Using python and CMOST to detect hydrate formation and do SA

Define Advanced Objective Functions in CMOST

	Name	Unit Label	Advanced Objective Function Type	Max Execution Time
1	Hydrate_Flag_EOS_withOutput2		External Executable Calculation	
2	Hydrate_Count_12p5C	%	External Executable Calculation	

Generic post-simulation command settings

External executable path: C:\ProgramData\Anaconda3\python.exe

Command line switches: "D:\OneDrive - Computer Modelling Group LTD\Documents\Support\230210_PET_Ahmad_Hydrates\Hydrate\HYDRA.cmpd\ObjFunc\Hydrate_Count_12.5C.py"

Experiment Name tagged command line arguments:

	Argument Switch	Argument File Type
1		Experiment-Name-Tagged SR3 file (.sr3)
2		Experiment-Name-Tagged External Executable Result File

Write your code using python

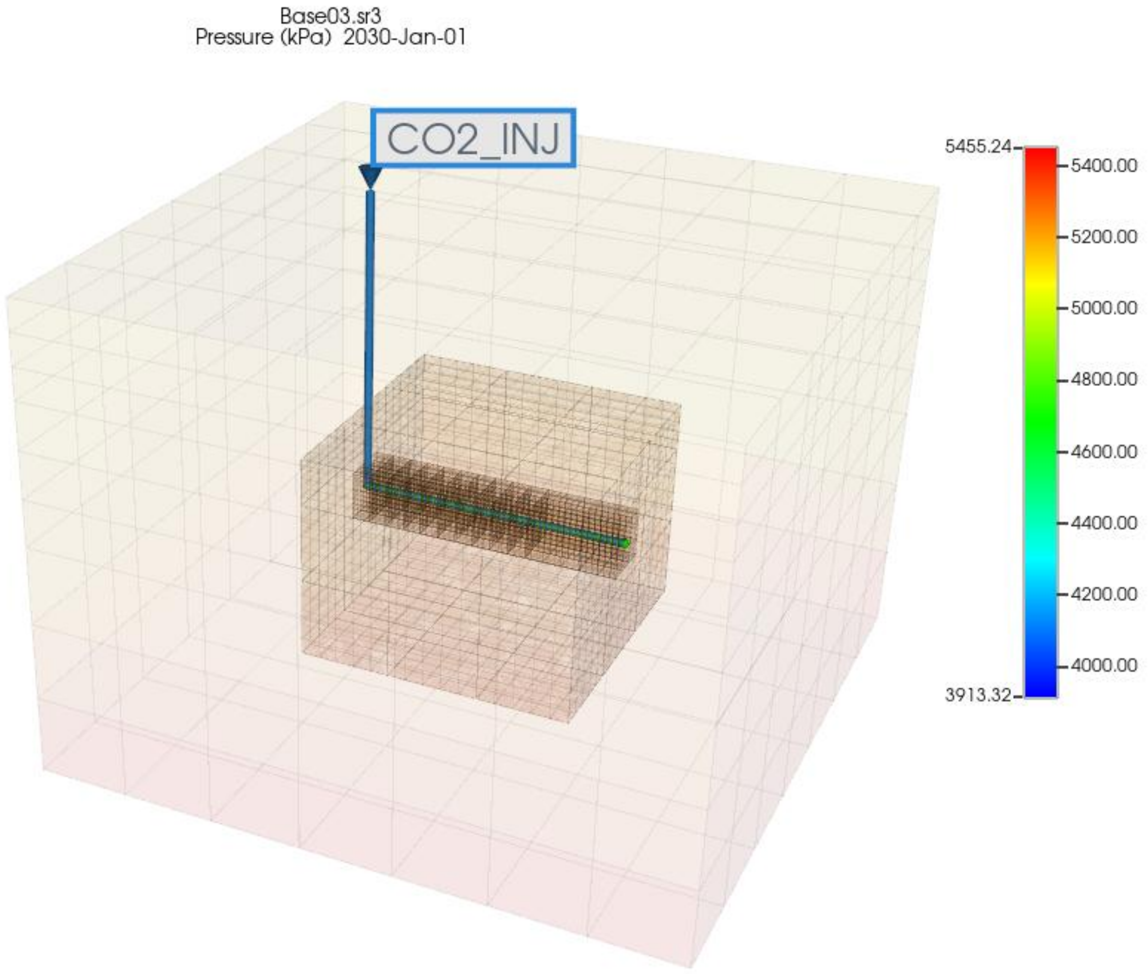
```
# Calculation
count_hydrate = 0
for date, gmf_h2o, gmf_ch4, gmf_co2, temp, pres_bar in fset:
    if float(temp)<float(285.65):
        try:
            if float(gmf_ch4)<=(1e-6):
                flash = fc.FlashController(components=['water', 'co2'])
                output = flash.main_handler(
                    compobjs=flash.compobjs,
                    z=np.asarray([float(gmf_h2o), float(gmf_co2)]),
                    T=float(temp),
                    P=float(pres_bar))
            else:
                flash = fc.FlashController(components=['water', 'ch4','co2'])
                output = flash.main_handler(
                    compobjs=flash.compobjs,
                    z=np.asarray([float(gmf_h2o), float(gmf_ch4), float(gmf_co2)]),
                    T=float(temp),
                    P=float(pres_bar))
```

Automatic processing done by CMOST

ID	Generator	H2OVAP		Hydrate_Flag_EOS_withOutput2	Hydrate_Count_12p5C
0	Reuse, One Parameter At A Time		1	1	32.207947
1	One Parameter At A Time		1	1	32.207947
2	One Parameter At A Time		1	1	27.495407
3	One Parameter At A Time		1	1	21.820411
4	One Parameter At A Time		1	1	17.051122
5	One Parameter At A Time		1	1	10.366826
6	One Parameter At A Time		1	1	5.8311856

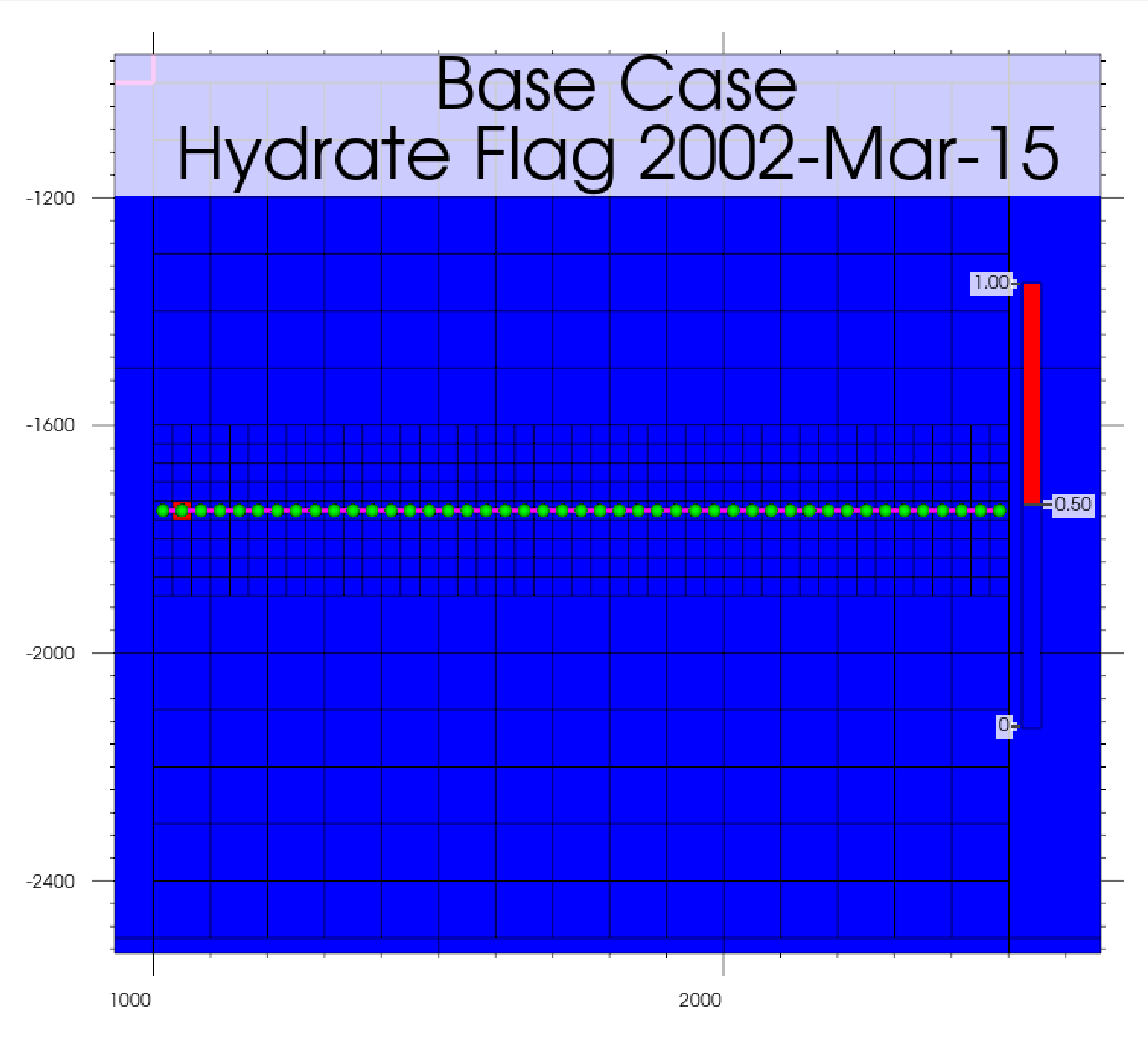
Base Model

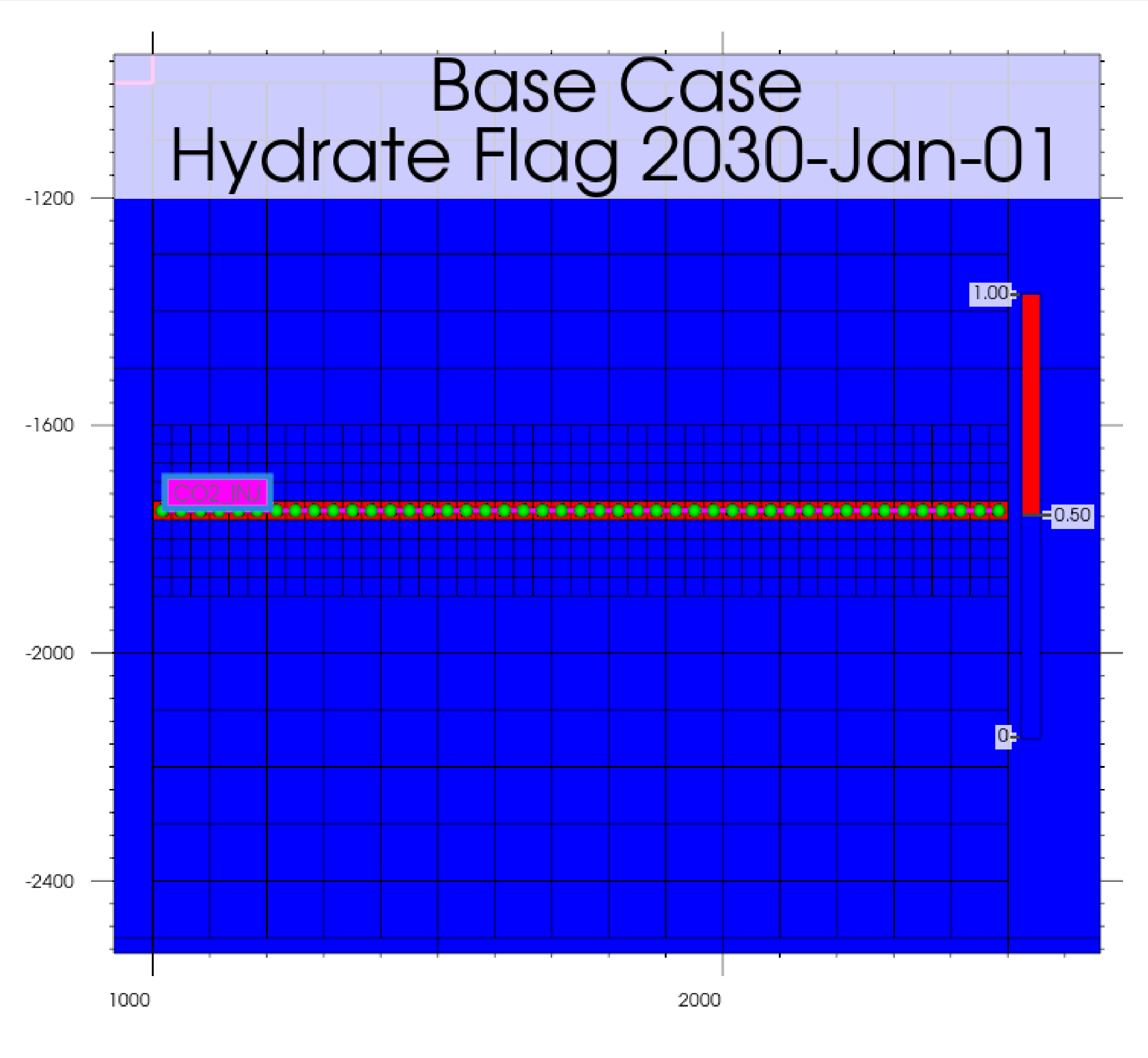
- ✓ Initial Field Pressure: 2000 psi (Used as maximum injection BHP)
- ✓ Field Abandonment Pressure: 1200 psi (Used as initial model pressure – REFPRES)
- ✓ Reference Depth: 4300 ft
- ✓ Initial temperature: 260 F
- ✓ Initial Composition: CO2=60% & C1=40%
- ✓ Average Porosity: 20%
- ✓ Average Permeability: 100 md
- ✓ GIP: 0.68 TSCF
- ✓ Injection rate: 20 MMSCF/DAY
- ✓ Injection period: 30 years (2000 to 2030)
- ✓ Bottom-hole Injection Temperature: 50F (10 degC)
- ✓ Rock heat capacity, thermal conductivity and heat loss parameters are all set to GEM default.



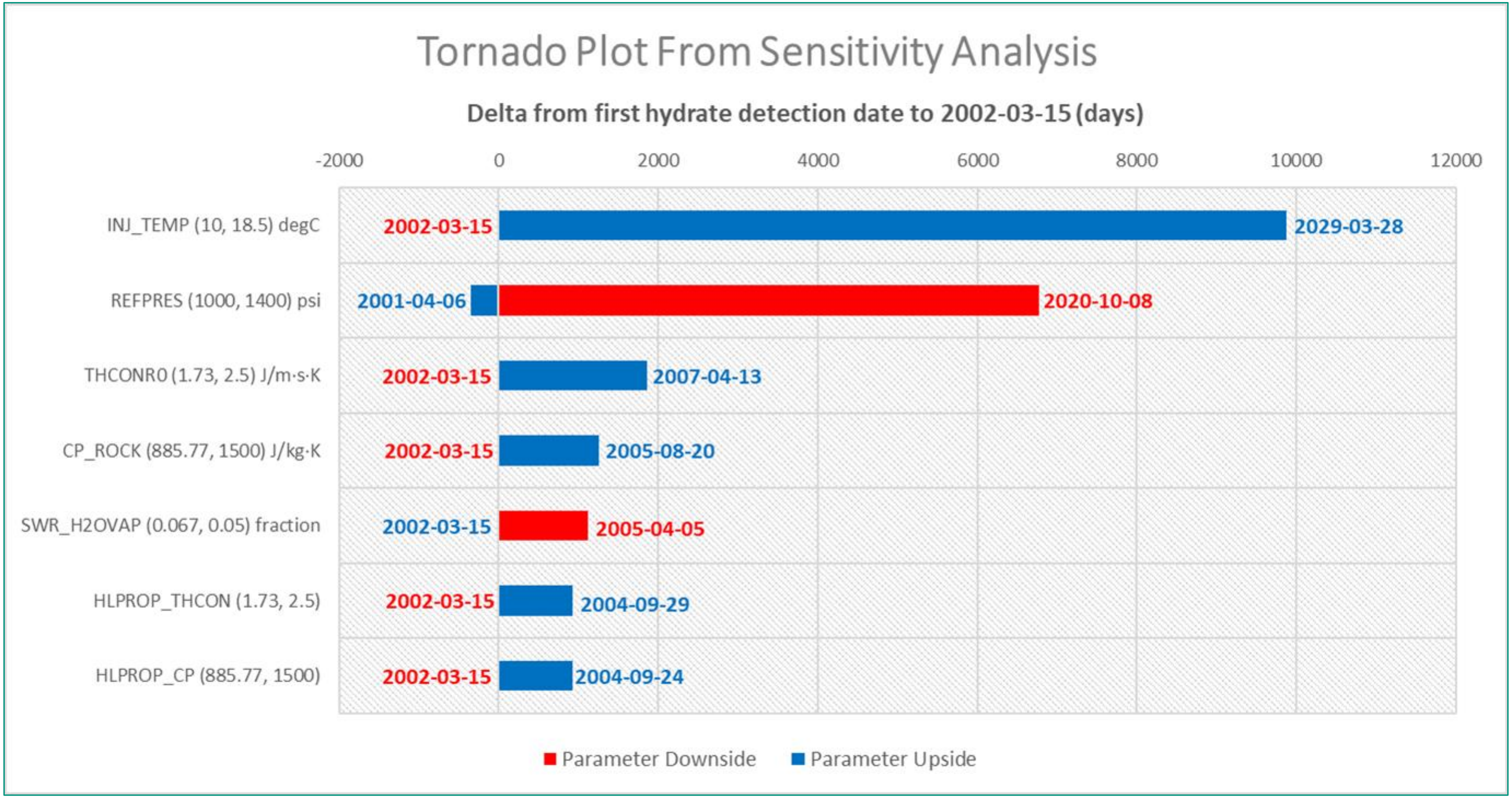
Base Case Results

Hydrate first detected on 2002-03-15.





Using CMOST to perform sensitivity analysis for the date of first hydrate detection



Case study #2

Real-time coupling between GEM and PyControl for hydrate formation detection and permeability blockage (WIP)

Continuing the work from the previous SPE paper

- Using the same GEM dataset (model).
- Using the same hydrateflash open-source python libraries.
- Using the same hydrate detection algorithm to flag hydrate formation.

What is new?

- Real-time coupling between GEM and python code using PyControl
- Writing an algorithm to set perforation grid-block permeability to zero when hydrate is detected.
- Currently the workflow assumes irreversible hydrate formation due to computation constraints.

What is PyControl?

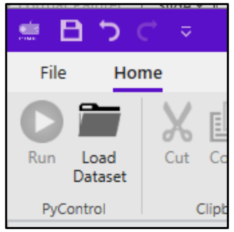
PyControl uses outboard to facilitate the work of using Python code over the following aspects:

- Coupling with simulator
- Running the cases
- Re-use of codes in/from other simulators

PyControl is to load your datasets, write the necessary python calculations in the script, then run and check the outputs

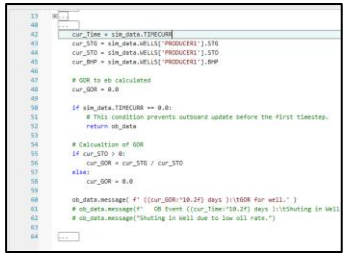
Load simulation files

- .dat
- python script



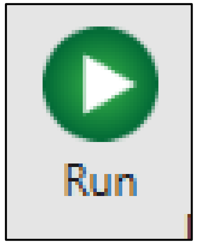
Python scripting

- Adding var.
- Calculations
- Writ. keywords
- Output results.



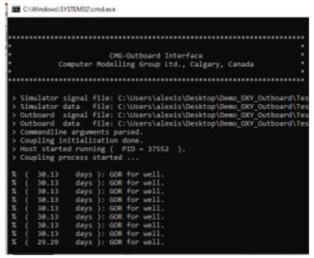
Run

- Simple to run



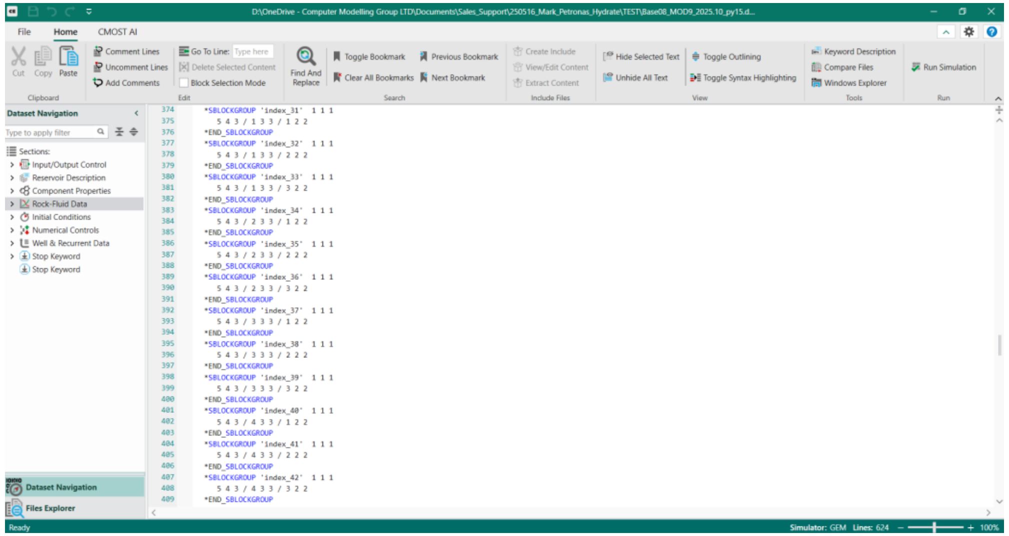
Results

- check/compare output in Results

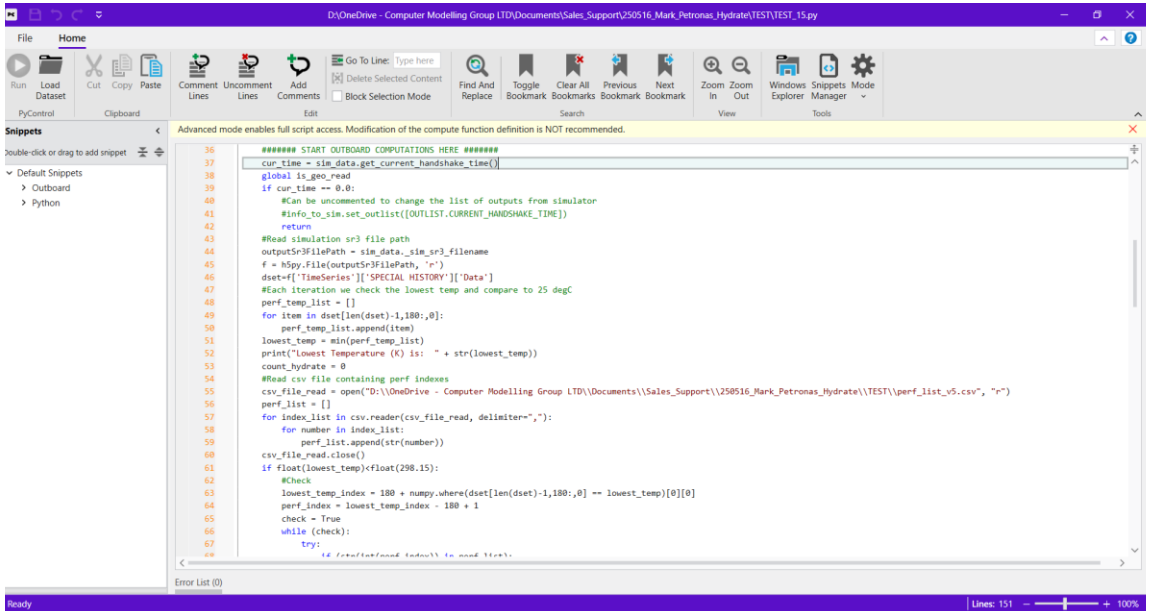


Using GEM and PyControl to detect hydrate and perform blockage

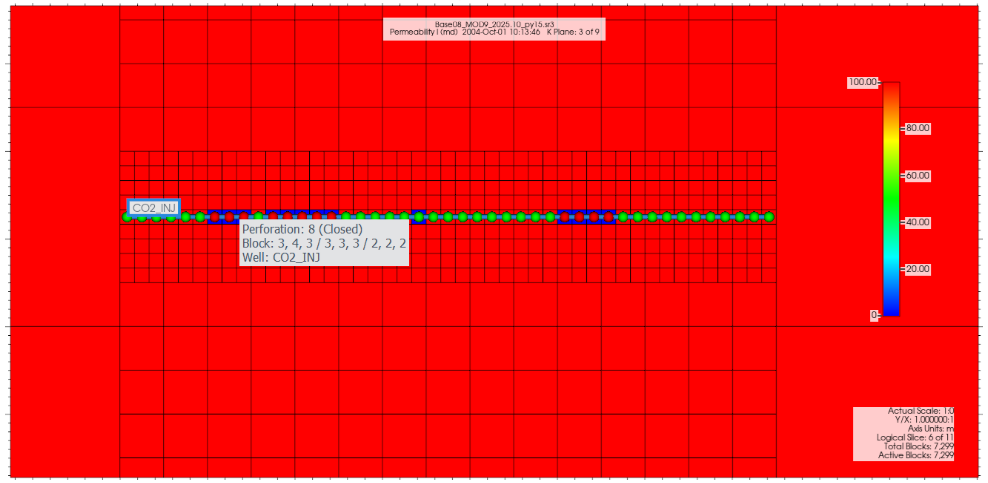
Setup your dataset with Builder or cEdit



Write your code using PyControl

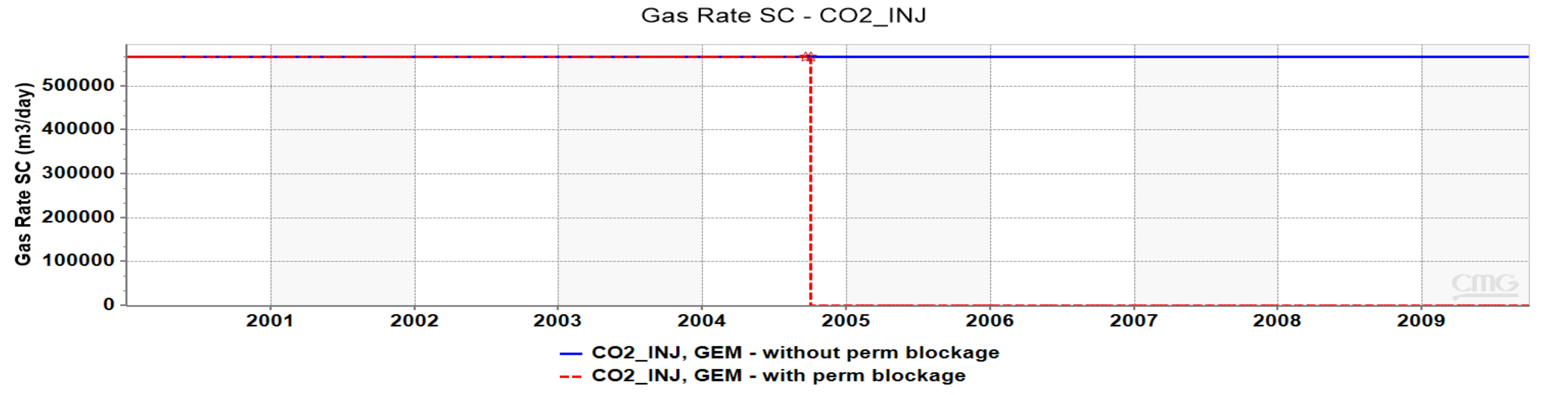


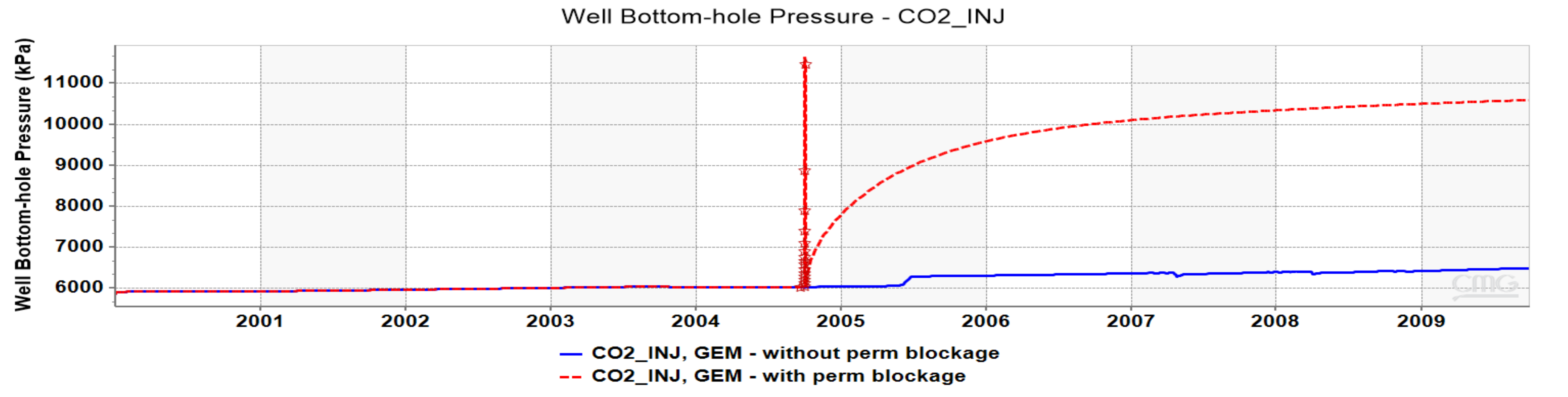
Run the real-time coupling and view the Results



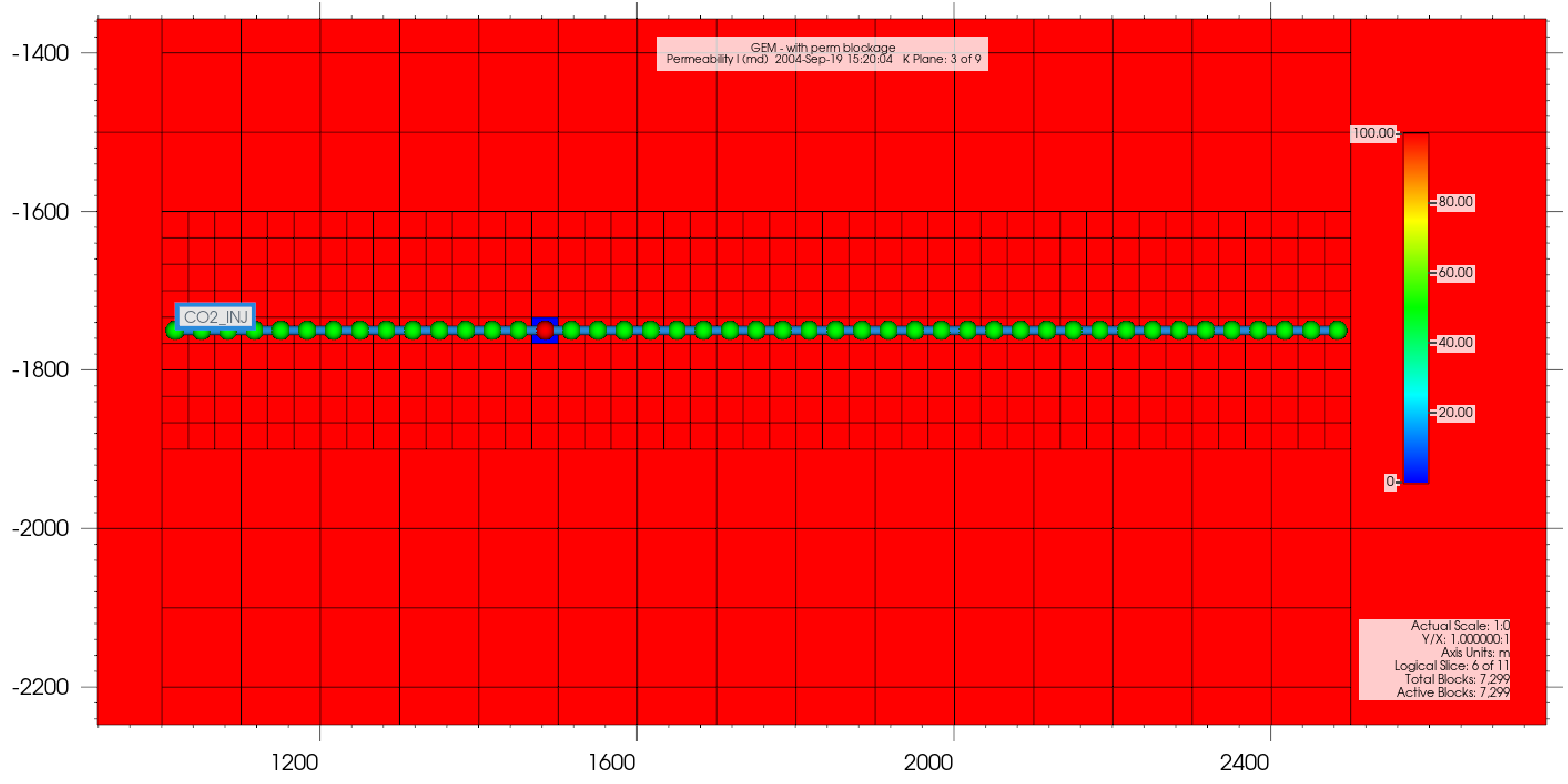
CMG

Results - Time series plots

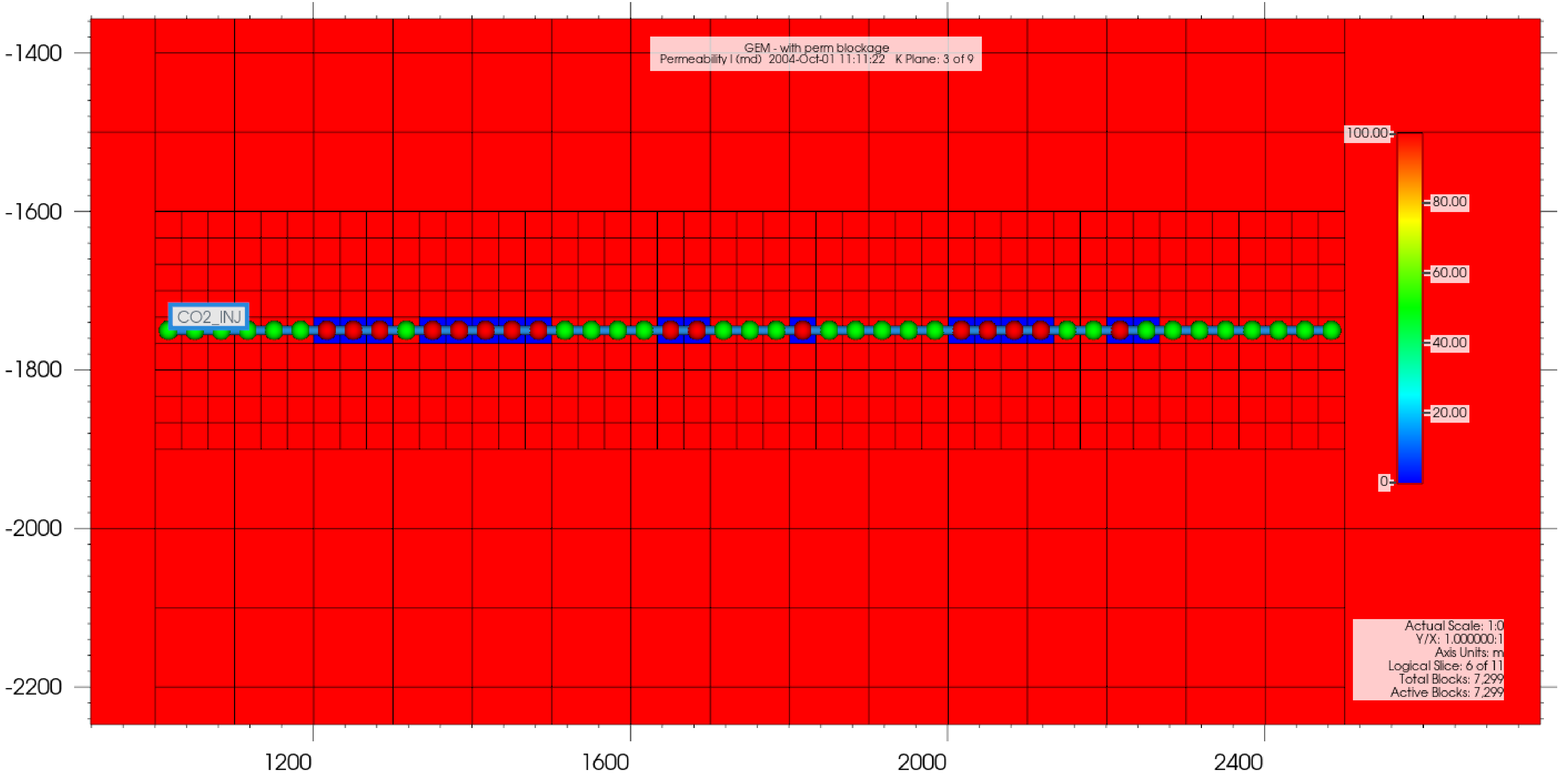




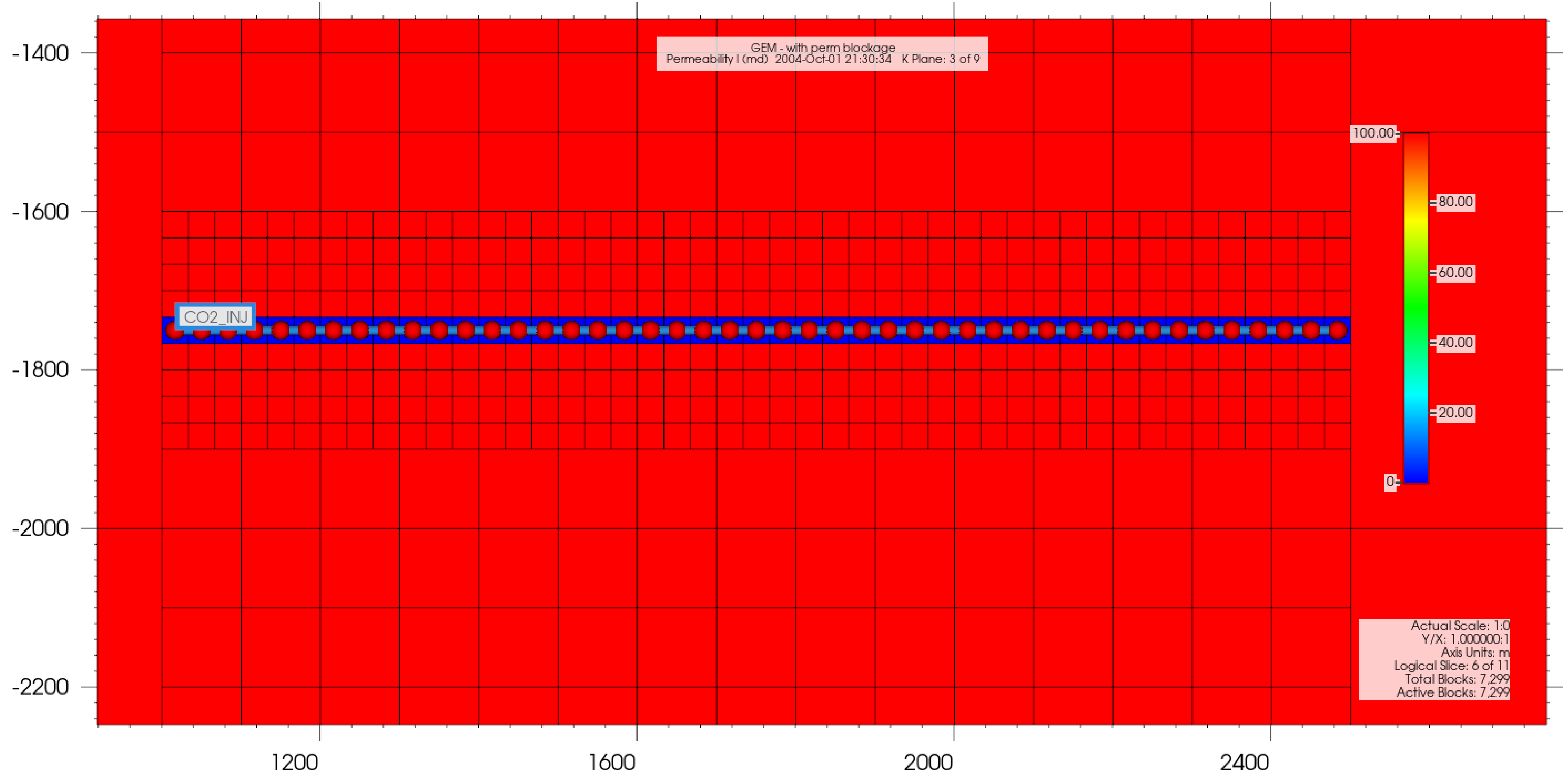
Results - 3D results - First hydrate formation and blockage



Results - 3D results - More and more perm blockage

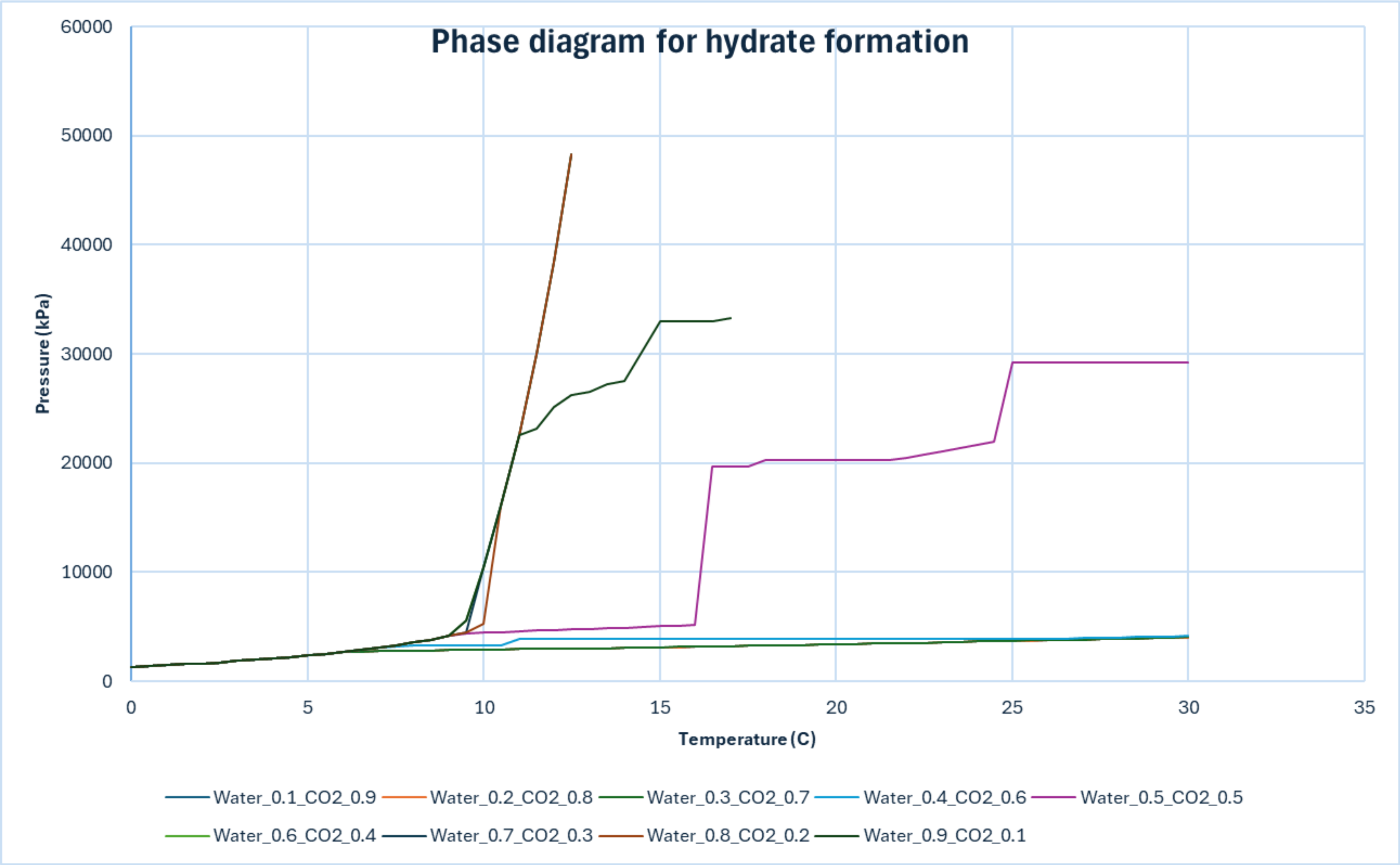


Results - 3D results - Well perforations all closed off



Next steps to improve the workflow

1. Make the permeability blockage reversible with changing conditions of hydrate formation.
2. Make the computations more effective by using a lookup of hydrate curves generated from the open-source EOS instead of doing flash calculations.



Questions & Discussion