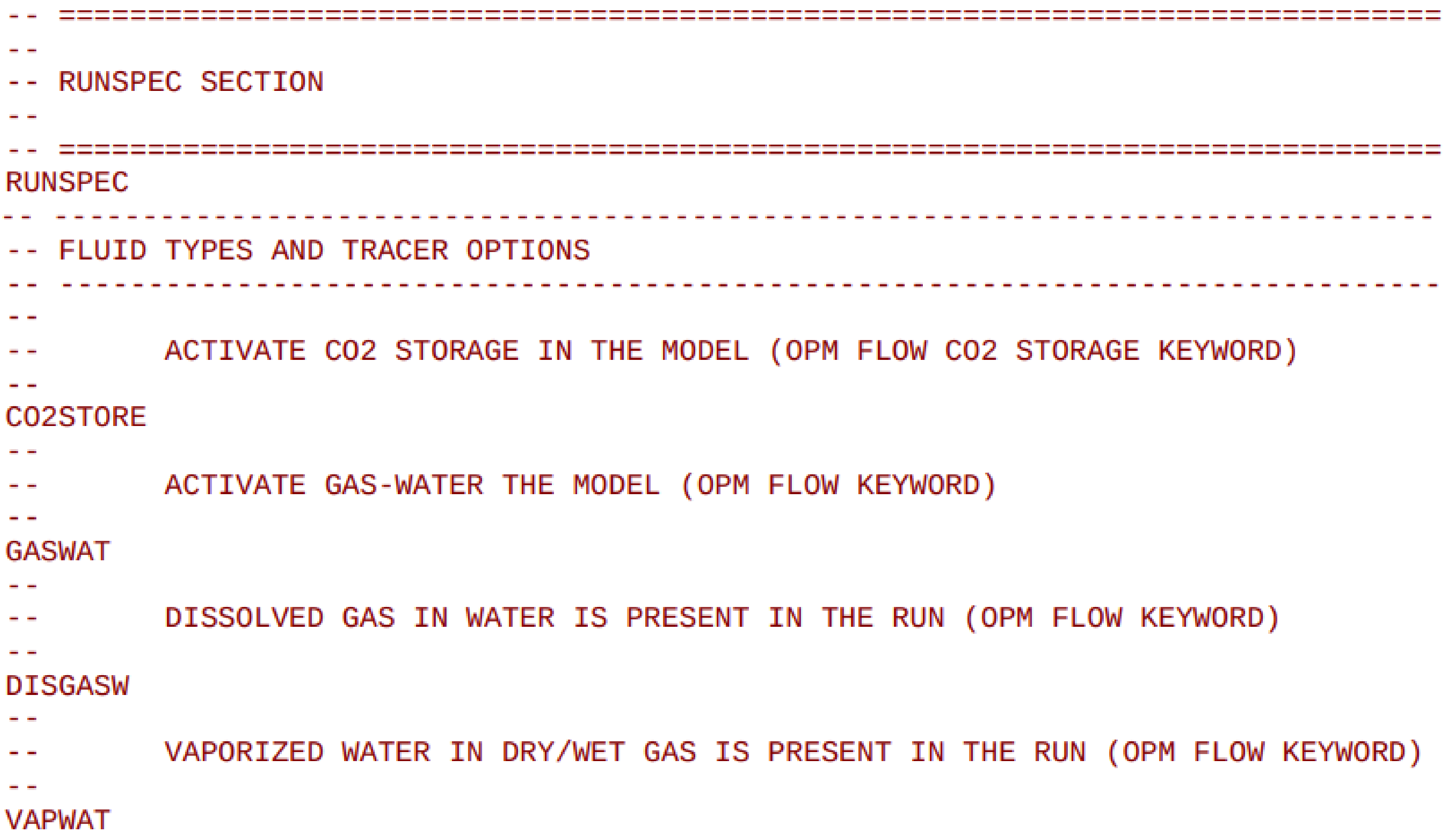
Flow CCS functionality

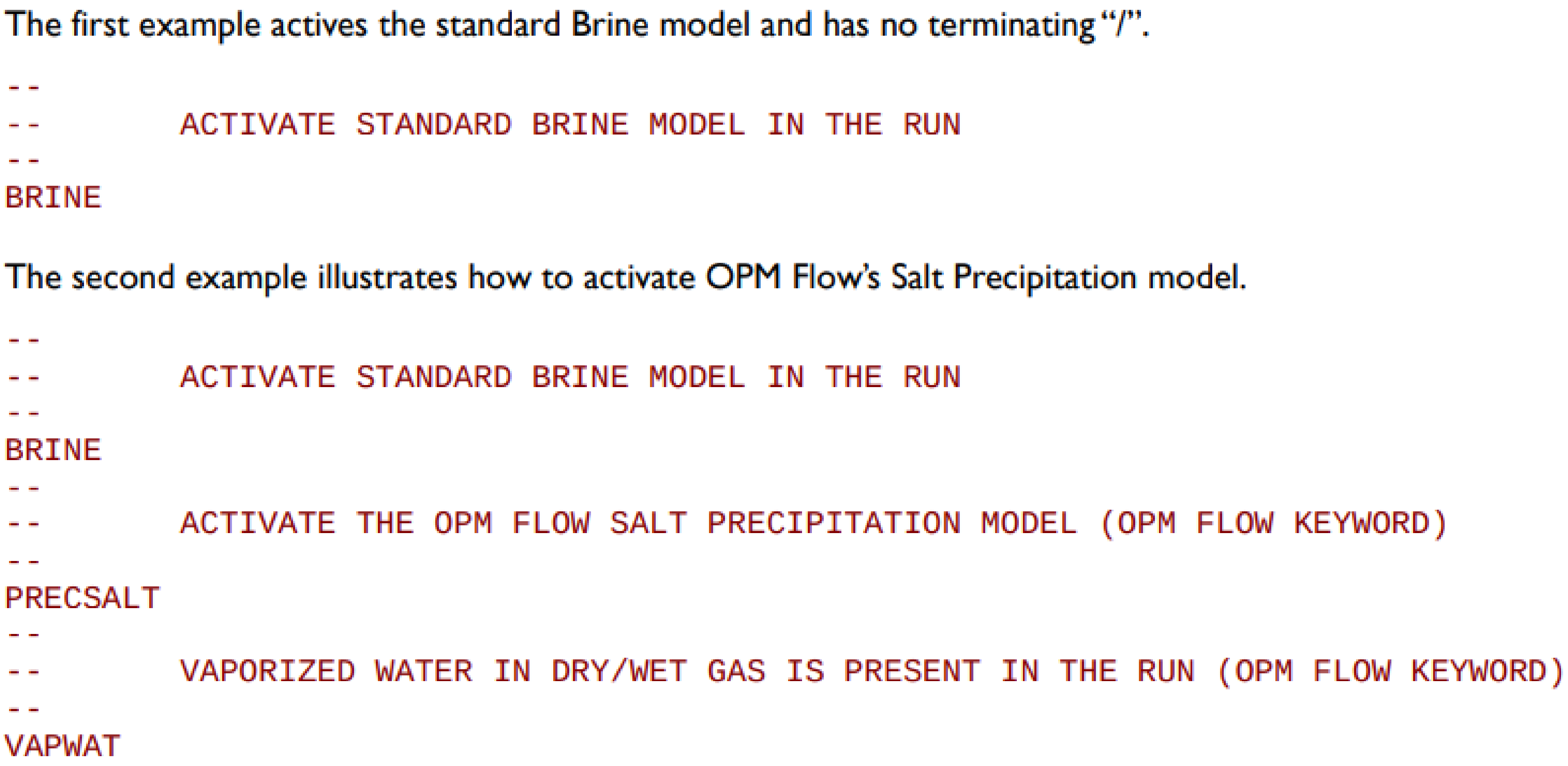
Alf B. Rustad with *huge* help from Tor Harald Sandve

CO2STORE in Flow



```
-- ==============================================================================
--
-- RUNSPEC SECTION
--
-- ==============================================================================
RUNSPEC
-- ------------------------------------------------------------------------------
-- FLUID TYPES AND TRACER OPTIONS
-- ------------------------------------------------------------------------------
--
--       ACTIVATE CO2 STORAGE IN THE MODEL (OPM FLOW CO2 STORAGE KEYWORD)
--
CO2STORE
--
--       ACTIVATE GAS-WATER THE MODEL (OPM FLOW KEYWORD)
--
GASWAT
--
--       DISSOLVED GAS IN WATER IS PRESENT IN THE RUN (OPM FLOW KEYWORD)
--
DISGASW
--
--       VAPORIZED WATER IN DRY/WET GAS IS PRESENT IN THE RUN (OPM FLOW KEYWORD)
--
VAPWAT
```

Salinity in Flow



The first example actives the standard Brine model and has no terminating "/".

```
--
--        ACTIVATE STANDARD BRINE MODEL IN THE RUN
--
BRINE
```

The second example illustrates how to activate OPM Flow's Salt Precipitation model.

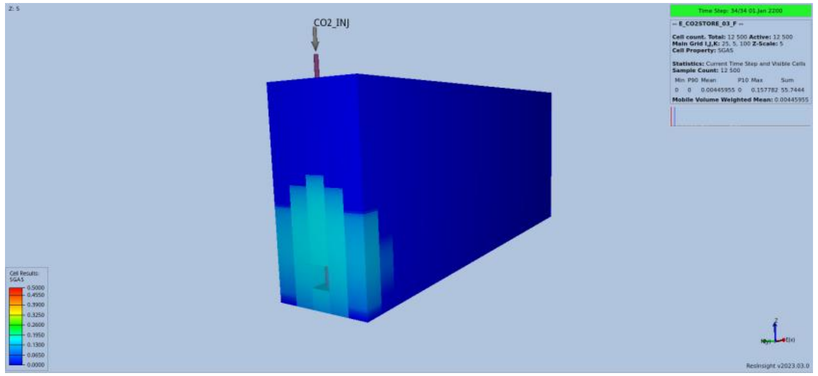
```
--
--        ACTIVATE STANDARD BRINE MODEL IN THE RUN
--
BRINE
--
--        ACTIVATE THE OPM FLOW SALT PRECIPITATION MODEL (OPM FLOW KEYWORD)
--
PRECSALT
--
--        VAPORIZED WATER IN DRY/WET GAS IS PRESENT IN THE RUN (OPM FLOW KEYWORD)
--
VAPWAT
```

Case 1; synthetic models

- The models are homogeneous and isothermal
- Injection rate is 1x10^6 Sm^3 (0.68 mtpa) from a single injector in 25 years
- With and without salinity

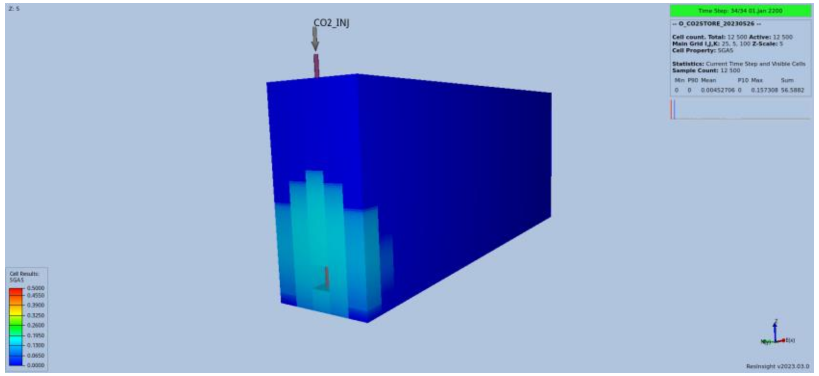
The grids

E300



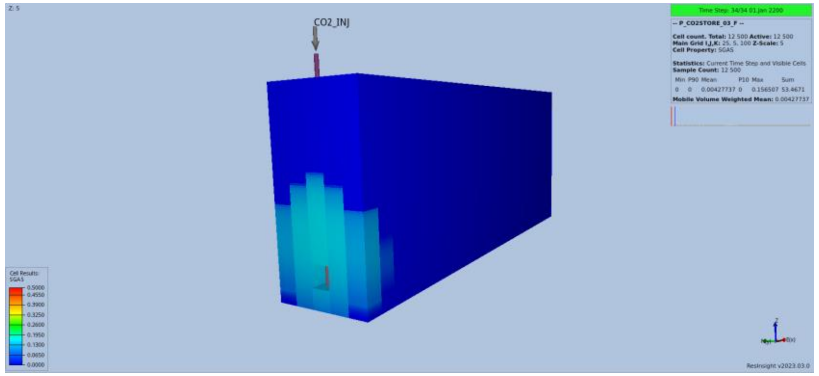
```
Time Step: 34/34 01 Jan 2200
-- E_CO2STORE_03_F --

Cell count. Total: 12 500 Active: 12 500
Main Grid I,J,K: 25, 5, 100 Z-Scale: 5
Cell Property: SGAS

Statistics: Current Time Step and Visible Cells
Sample Count: 12 500
 Min  P90  Mean        P10  Max       Sum
 0    0    0.00445955  0    0.157782  55.7444
Mobile Volume Weighted Mean: 0.00445955
```

OPM Flow



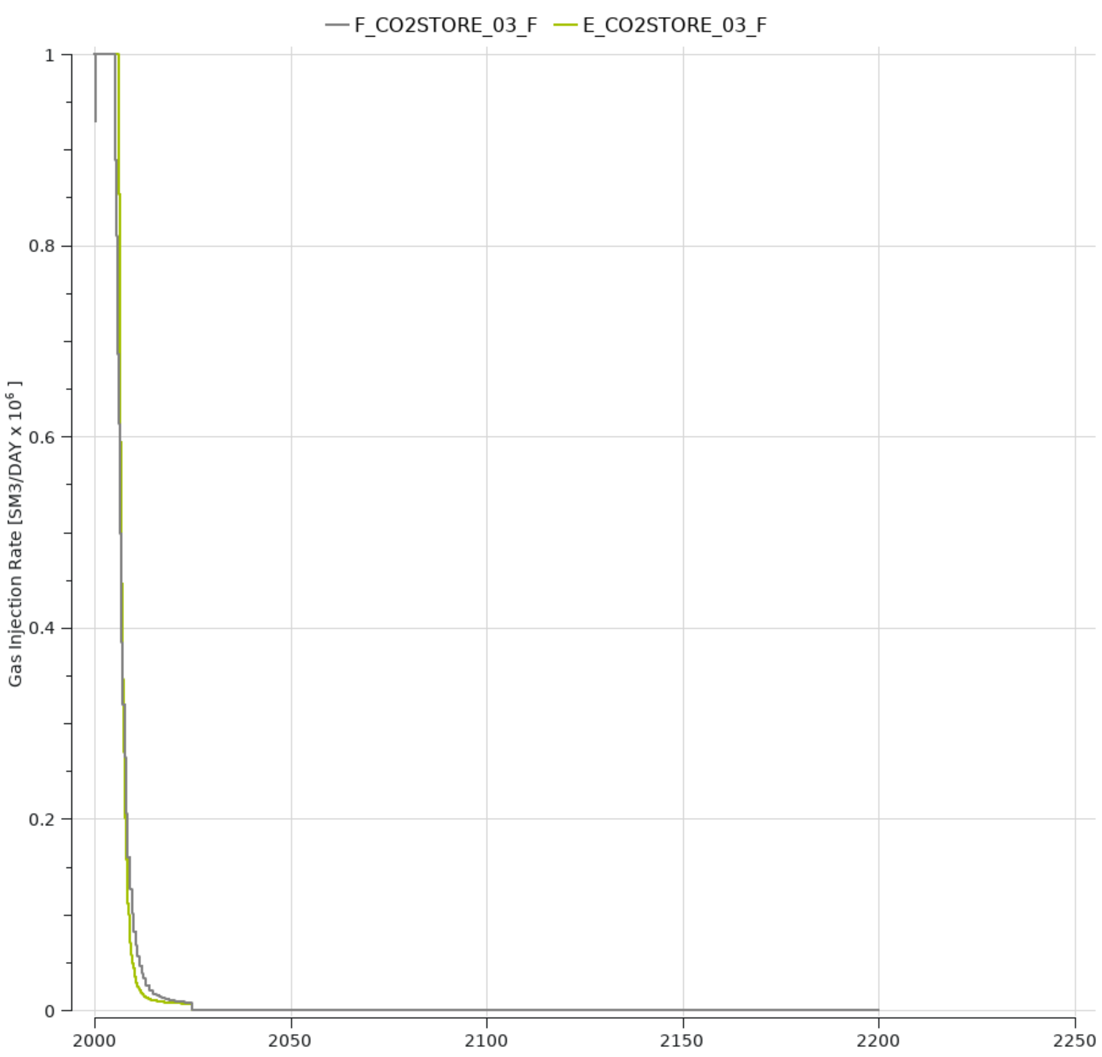
```
Time Step: 34/34 01 Jan 2200
-- O_CO2STORE_20230526 --

Cell count. Total: 12 500 Active: 12 500
Main Grid I,J,K: 25, 5, 100 Z-Scale: 5
Cell Property: SGAS

Statistics: Current Time Step and Visible Cells
Sample Count: 12 500
 Min  P90  Mean        P10  Max       Sum
 0    0    0.00452706  0    0.157308  56.5882
```

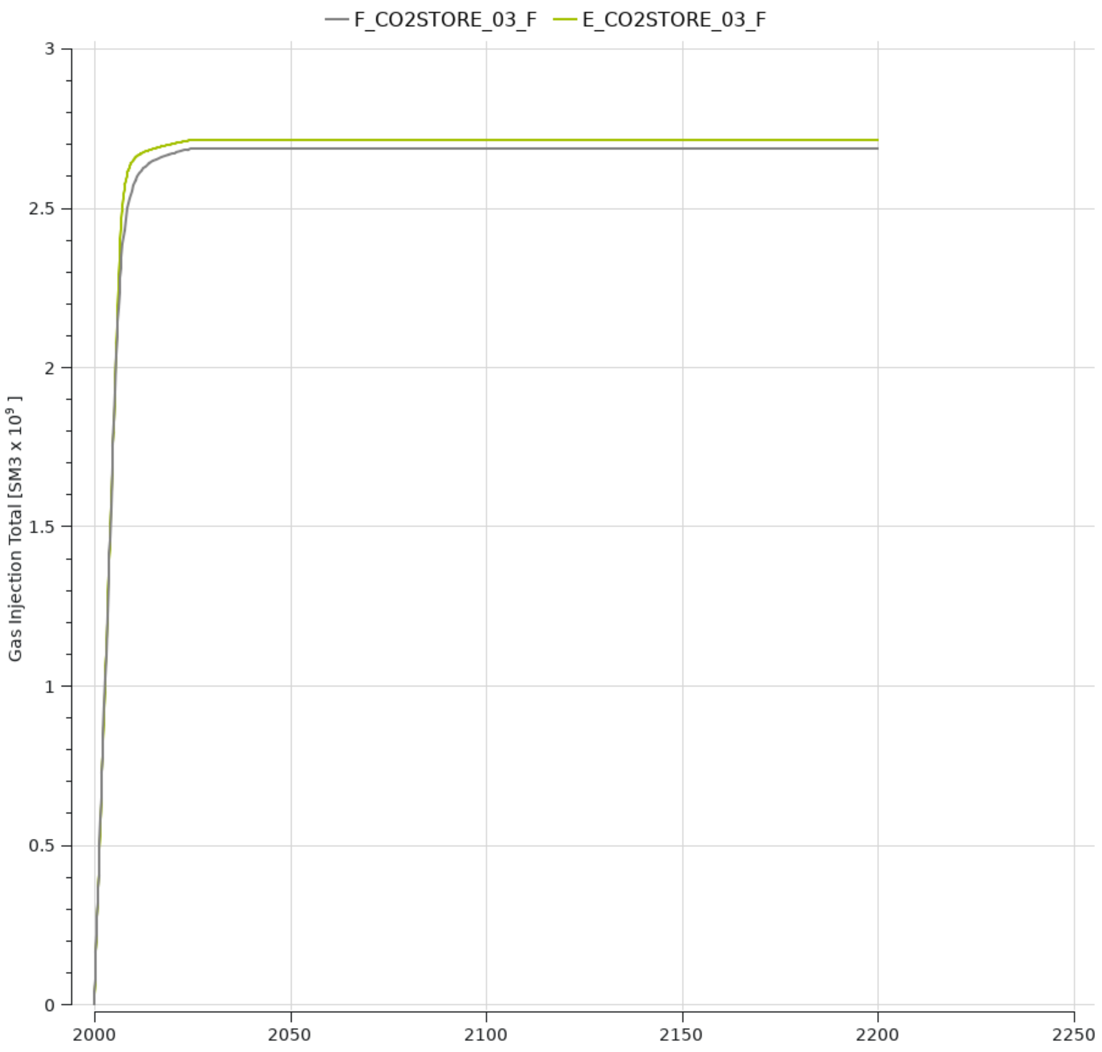
Pflotran



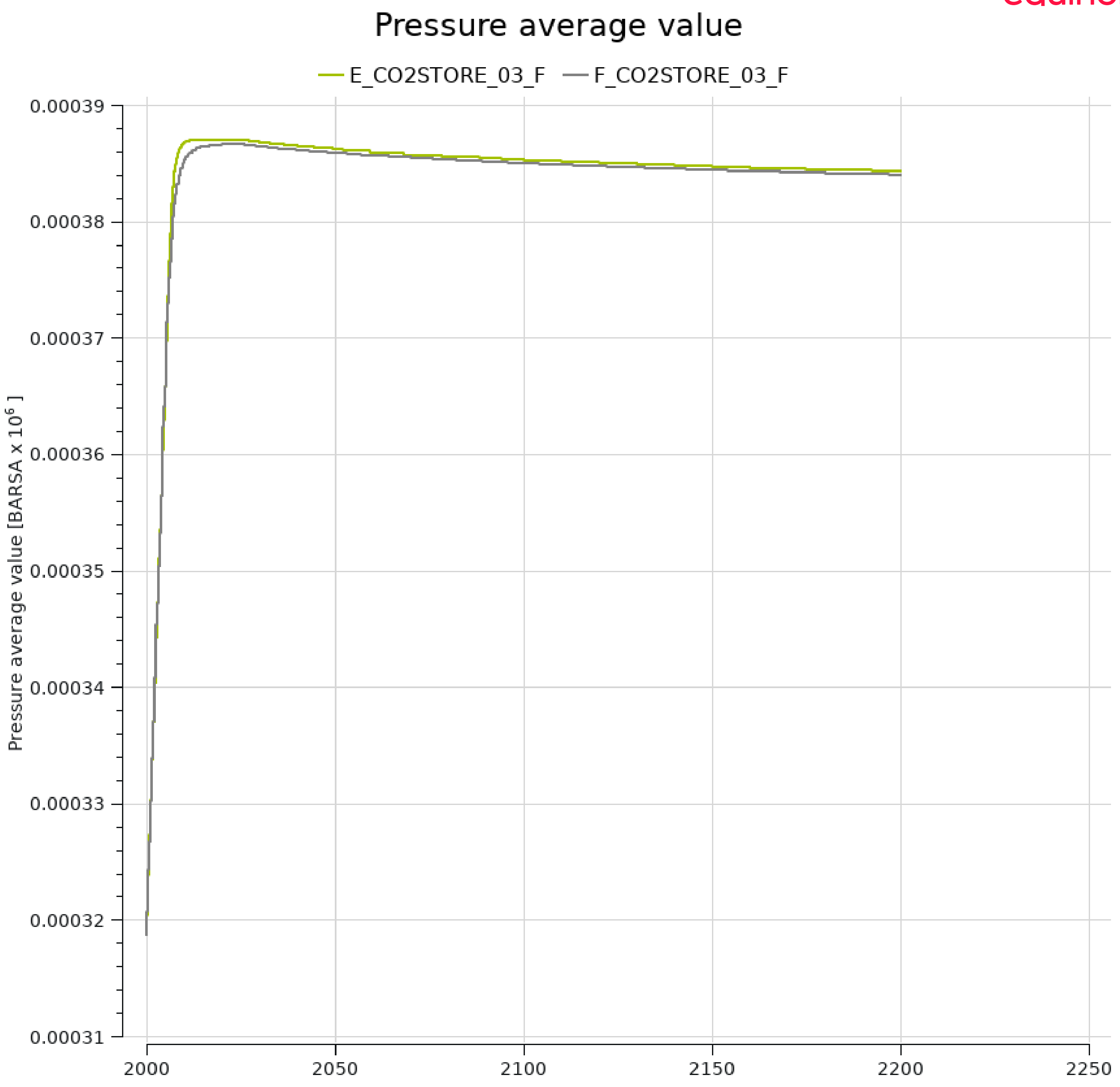
```
Time Step: 34/34 01 Jan 2200
-- P_CO2STORE_03_F --

Cell count. Total: 12 500 Active: 12 500
Main Grid I,J,K: 25, 5, 100 Z-Scale: 5
Cell Property: SGAS

Statistics: Current Time Step and Visible Cells
Sample Count: 12 500
 Min  P90  Mean        P10  Max       Sum
 0    0    0.00427737  0    0.156507  53.4671
Mobile Volume Weighted Mean: 0.00427737
```

Gas injection

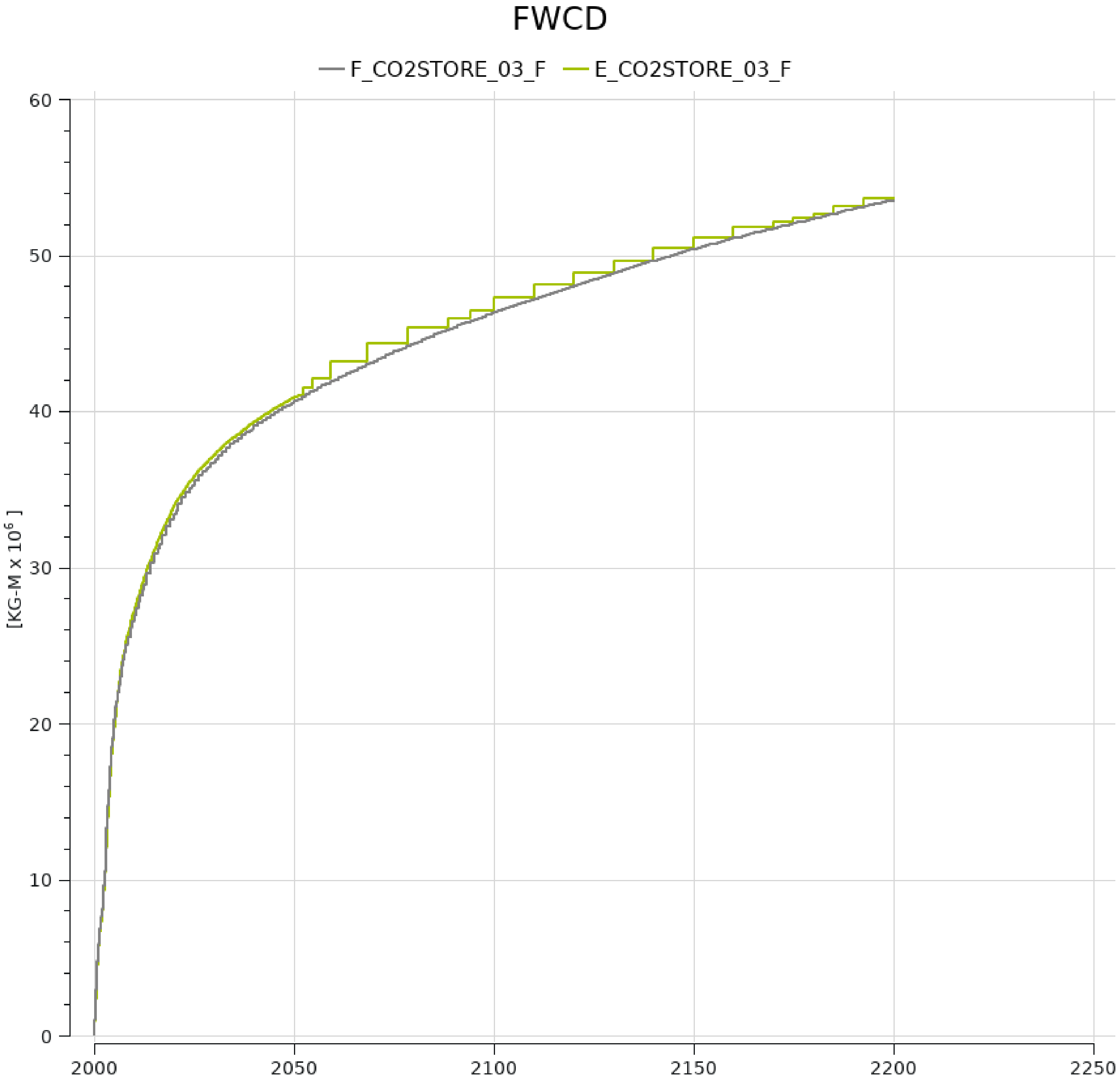

Gas Injection Rate


Gas Injection Total

Average reservoir pressure

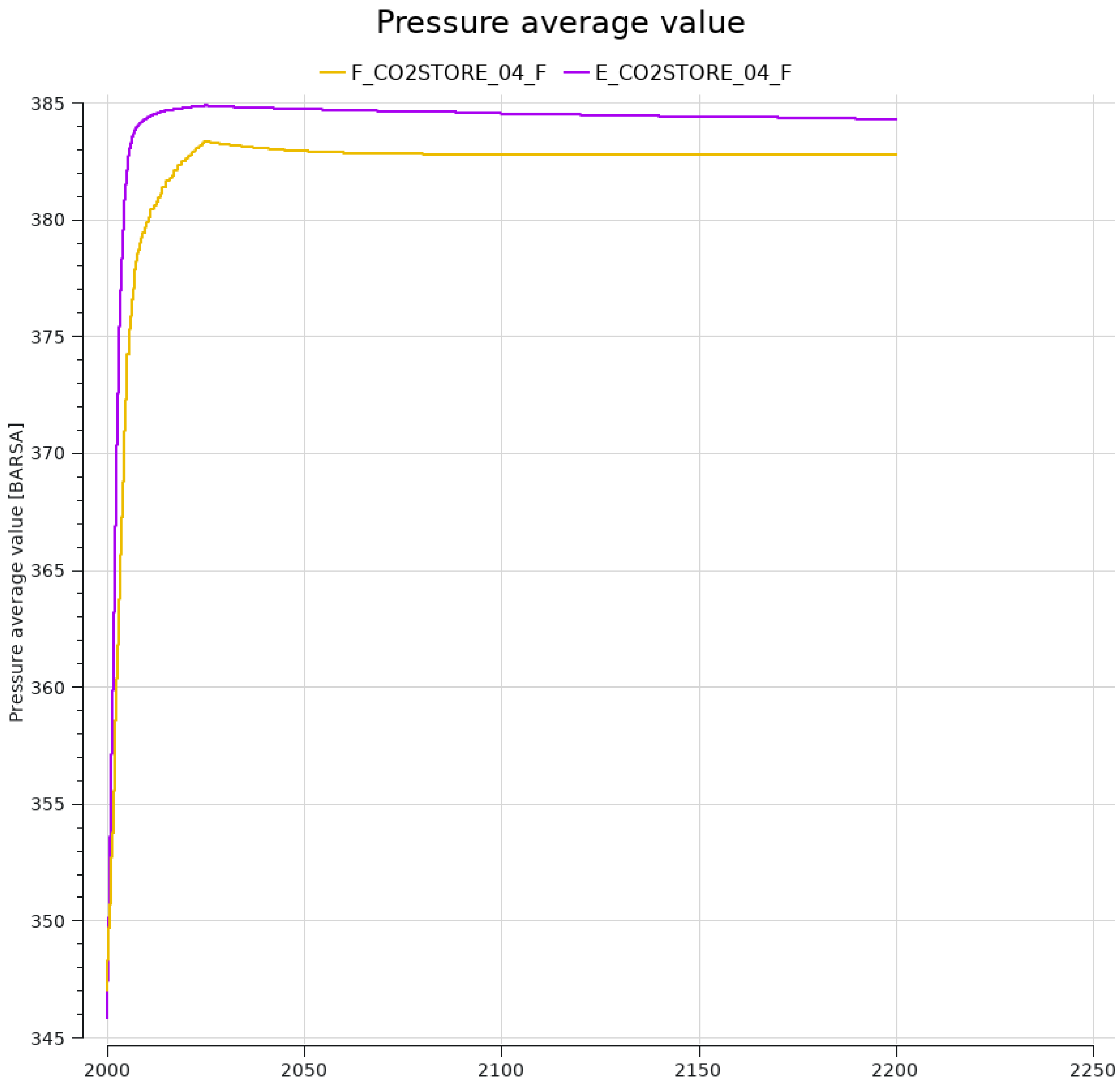


Dissolved CO2 mass

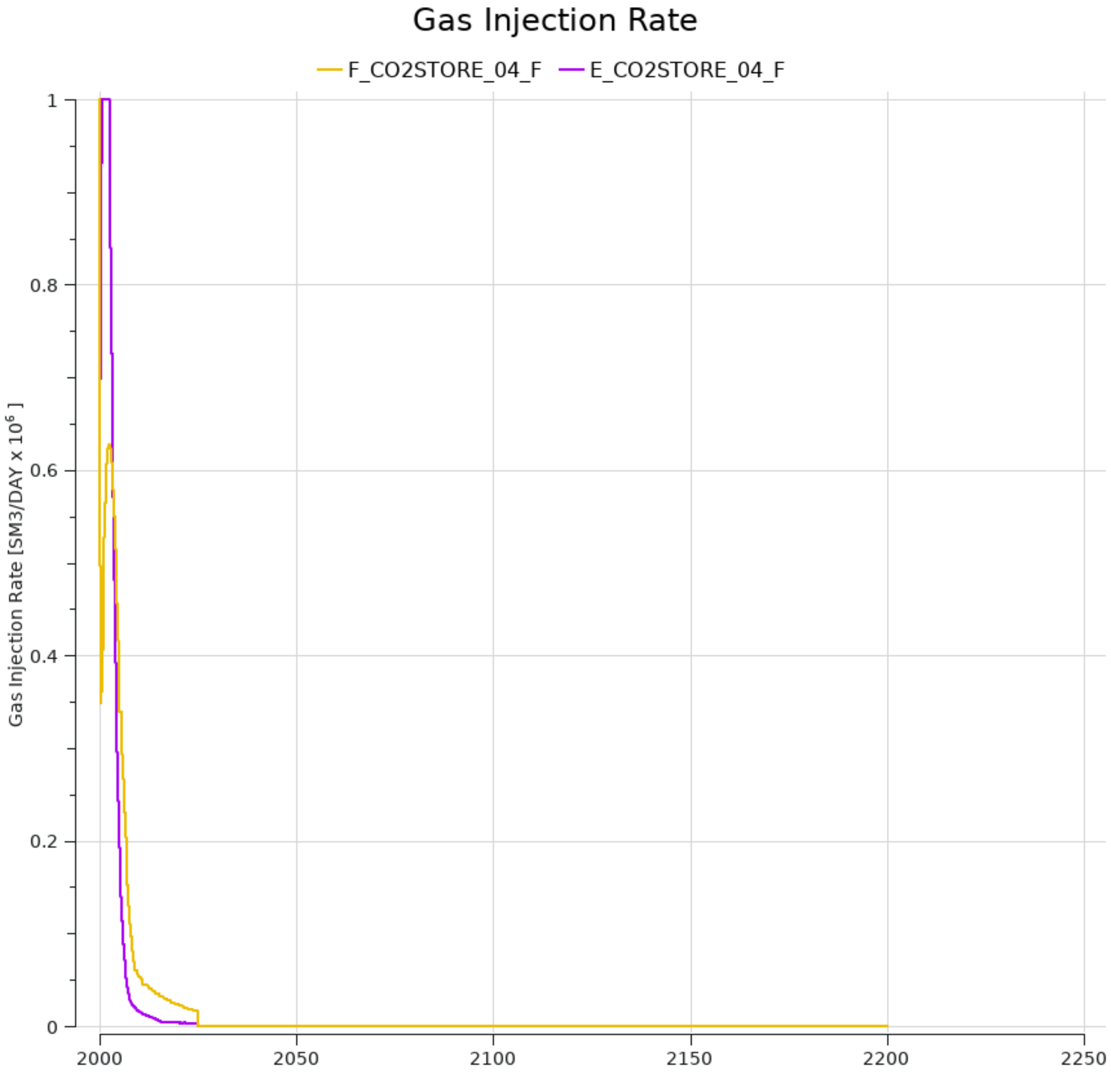


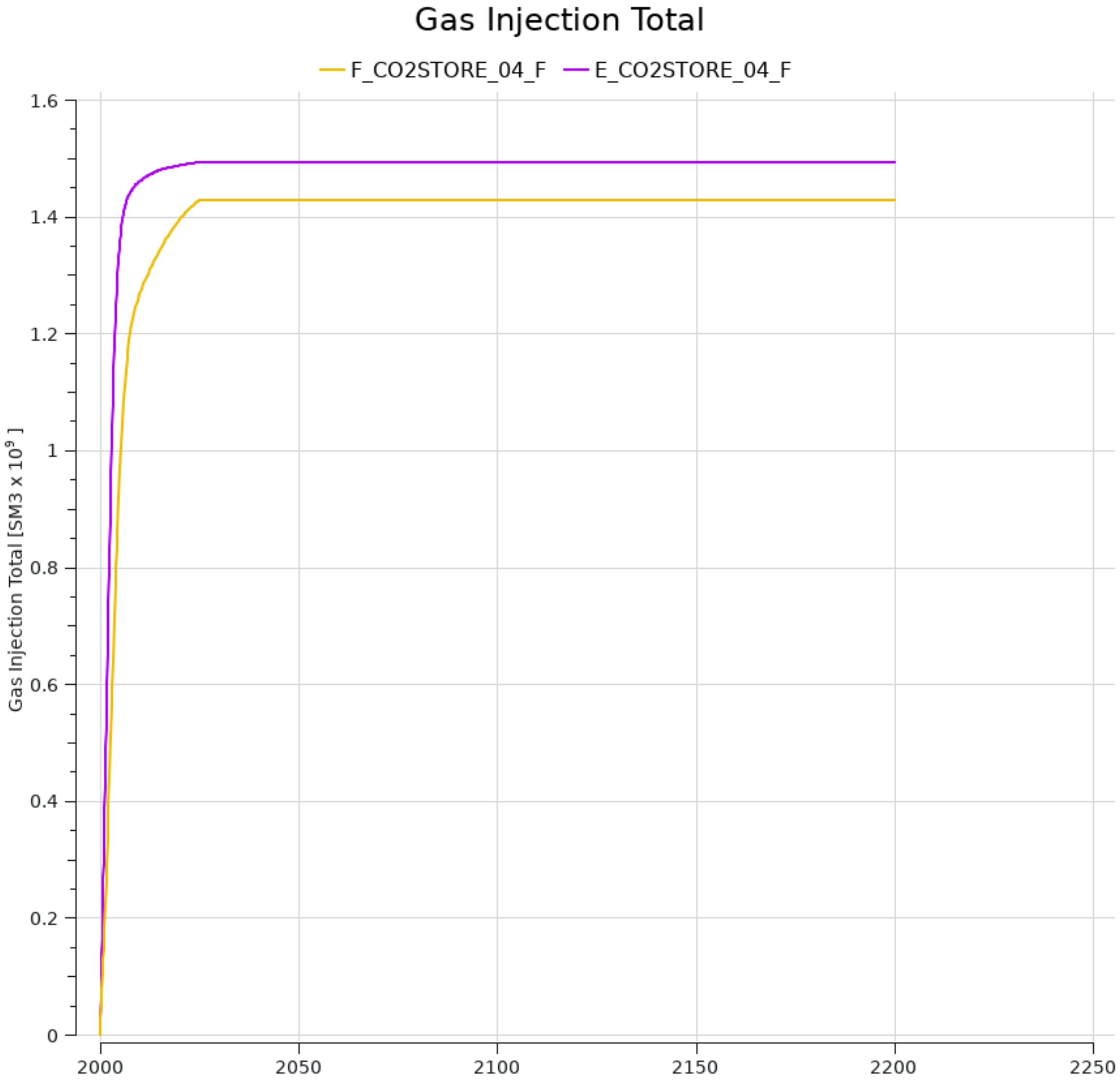
- Model with salinity

Average pressure

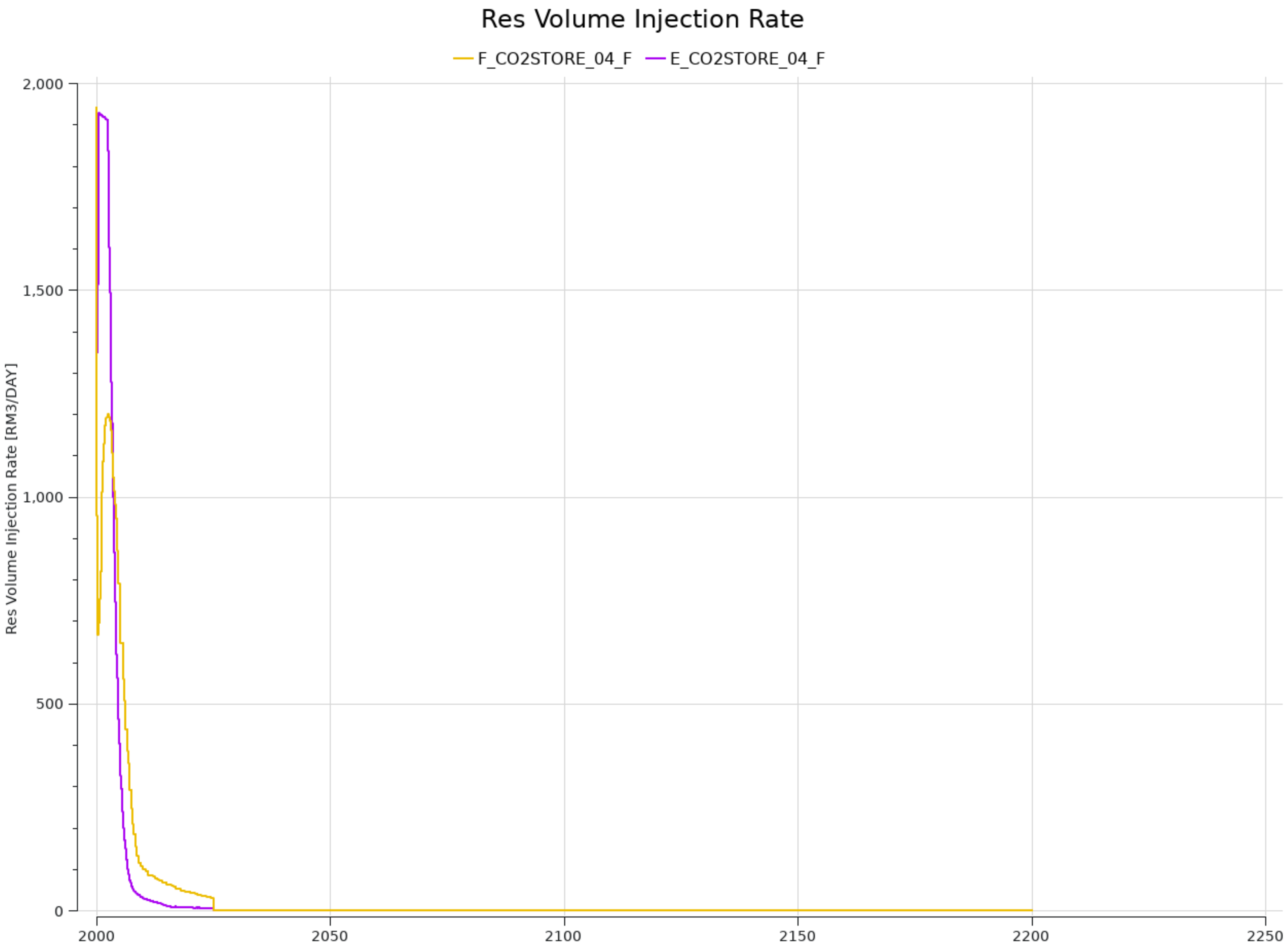


Injection rates

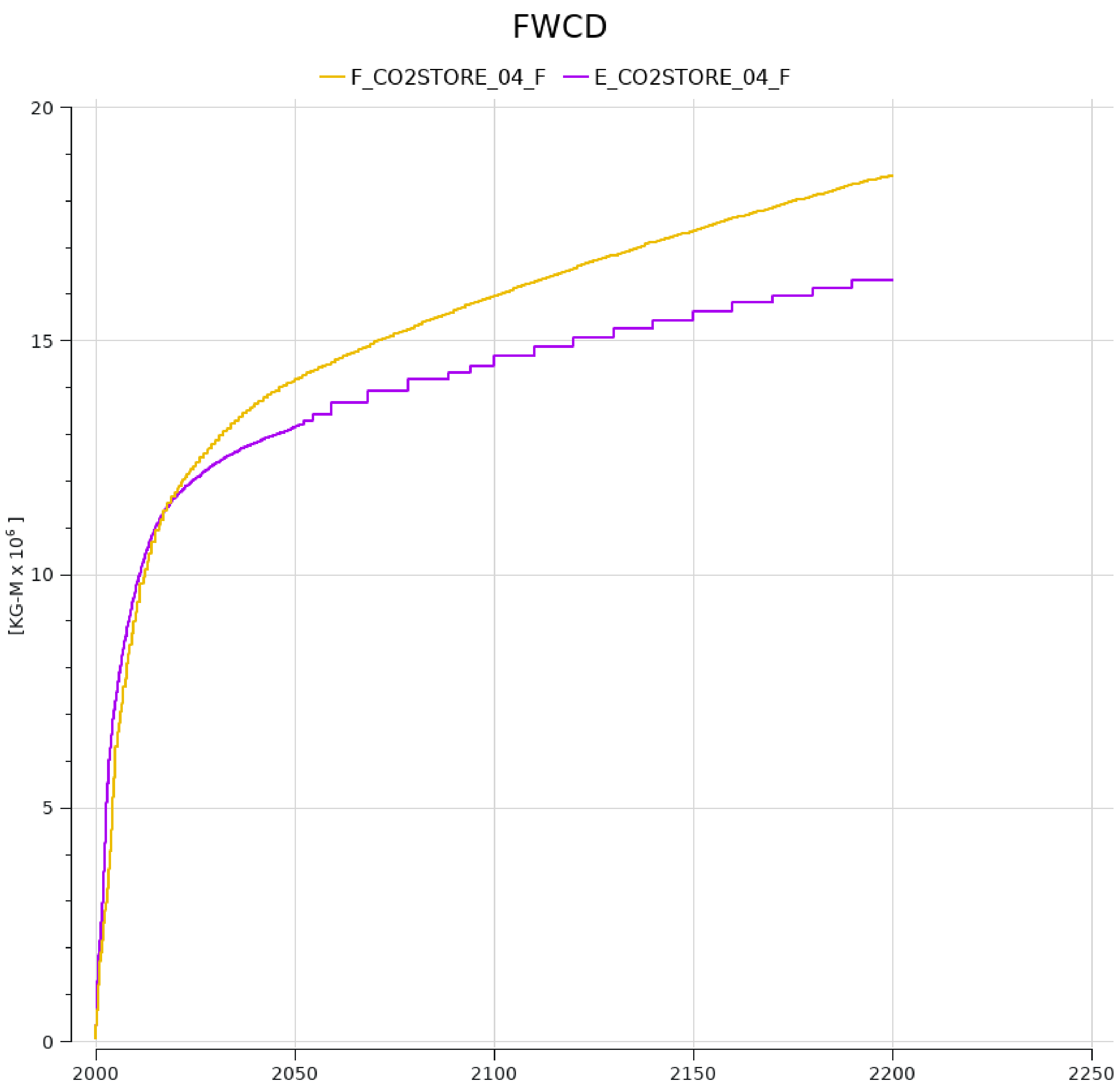




Reservoir rate



Dissolution



Observations

- For models without salinity
    - Generally good agreement between simulators

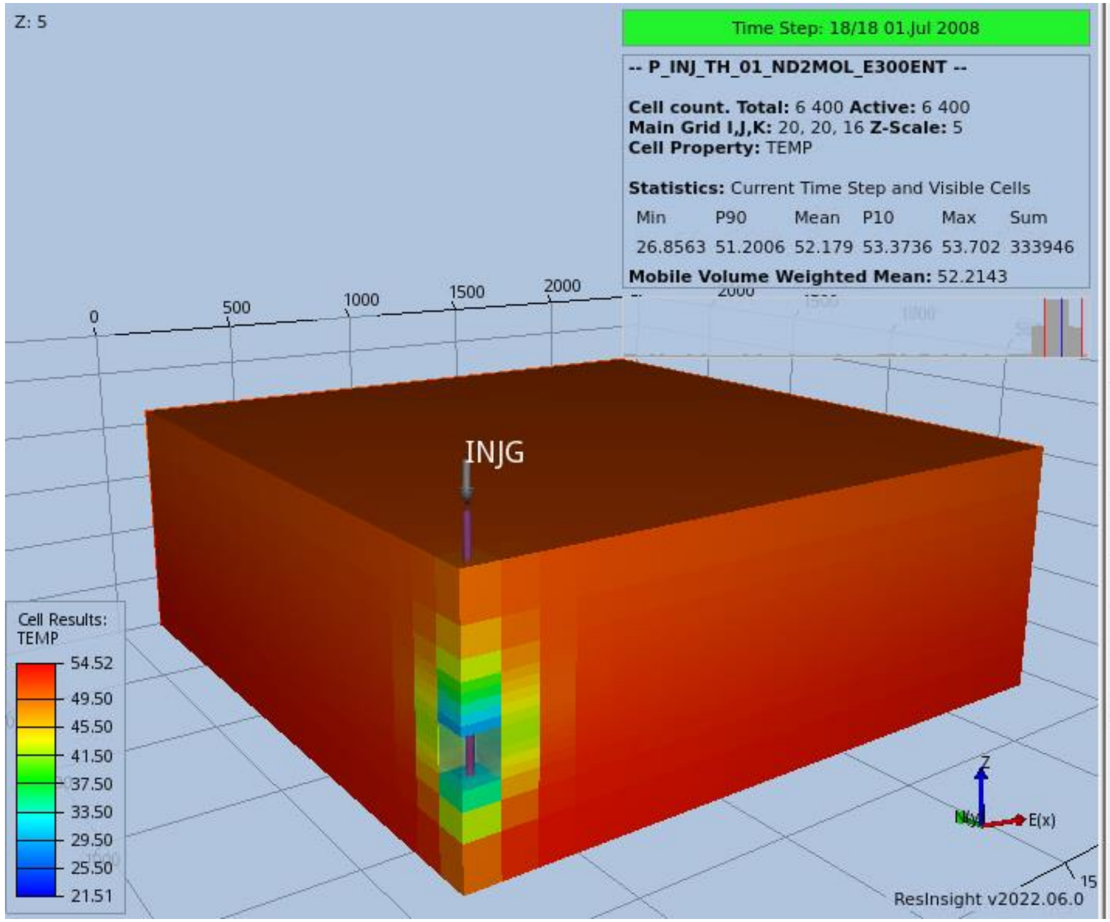
- For models with salinity deviations are observed
    - For FWCD the mixing model matters
    - For reservoir rates we need to understand difference

- Thermal simulations ✓
    - With salinity

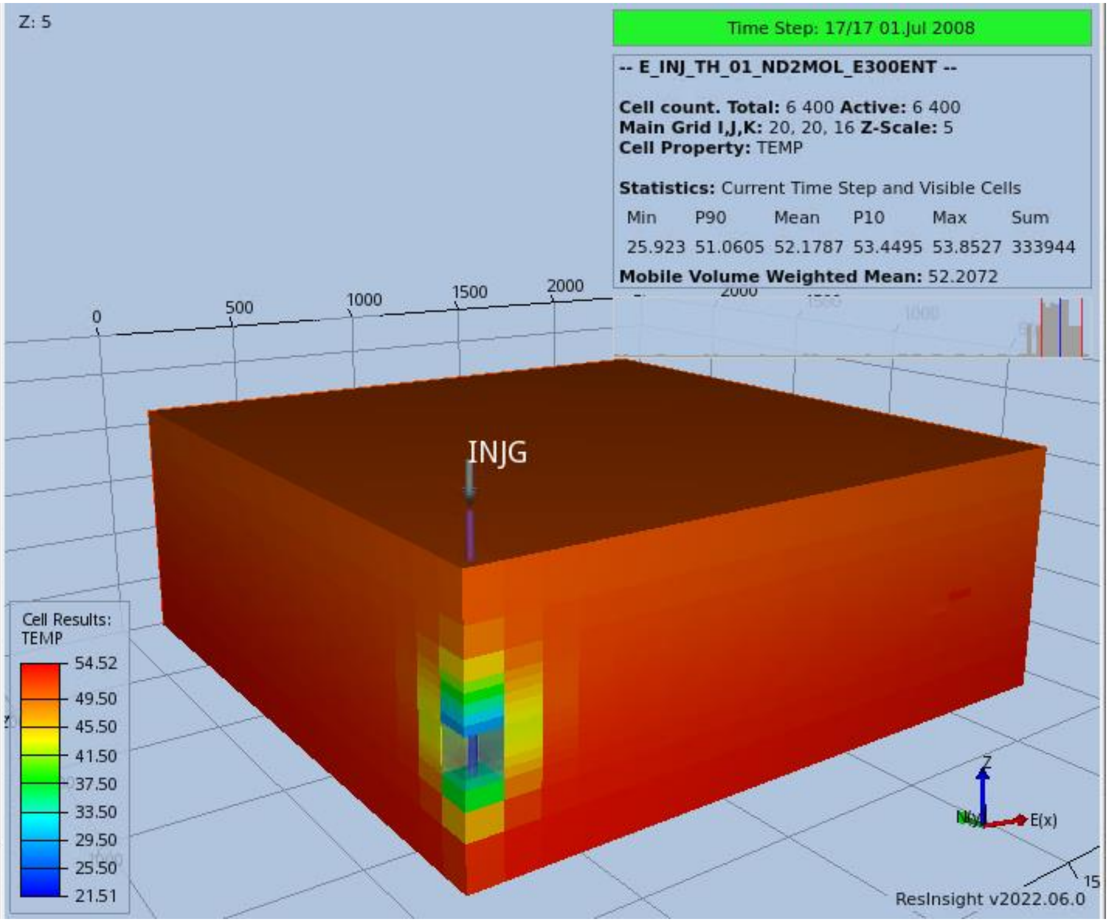
The grids

Pflotran

E300

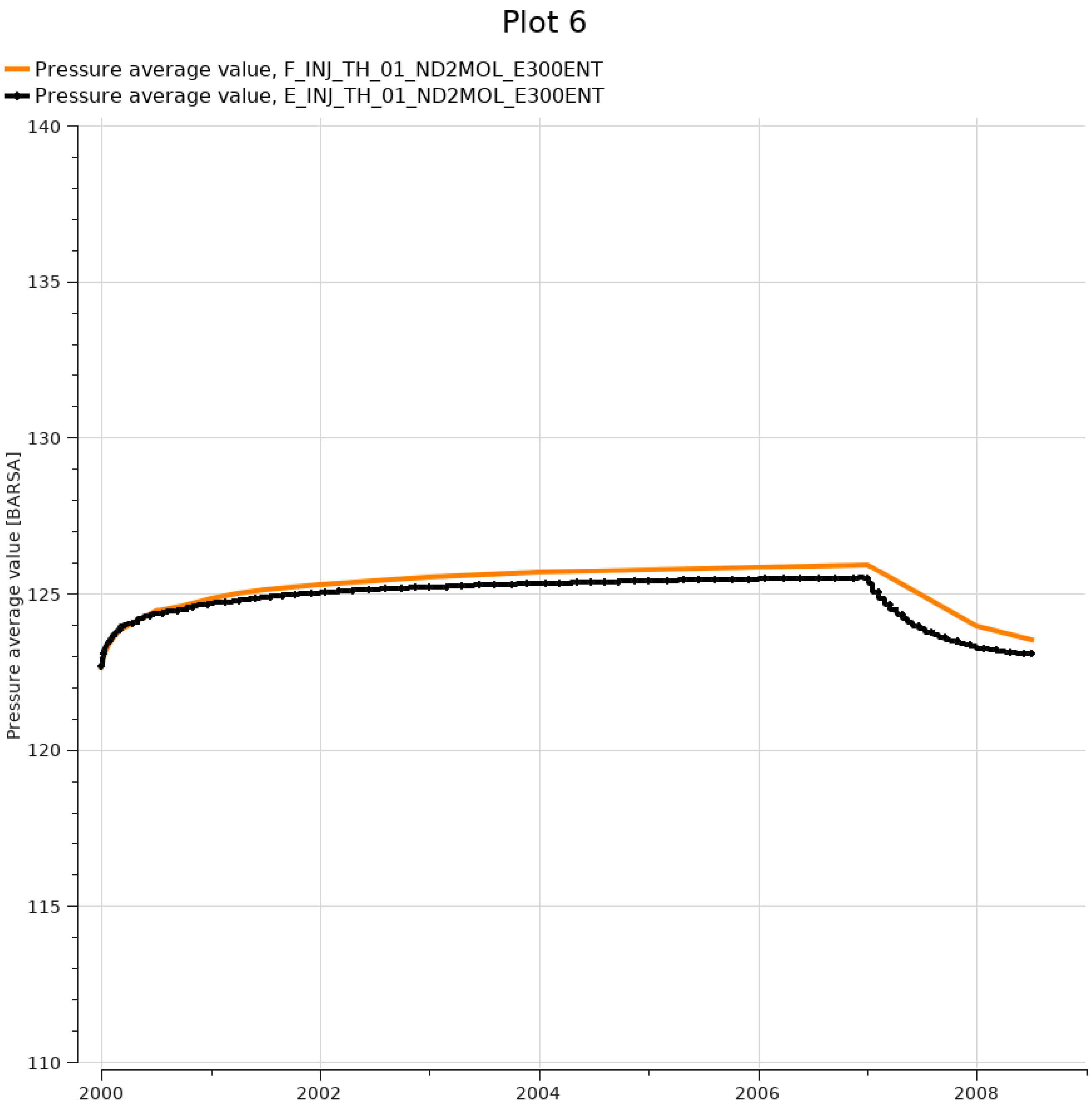


```
Time Step: 18/18 01.Jul 2008

-- P_INJ_TH_01_ND2MOL_E300ENT --

Cell count. Total: 6 400 Active: 6 400
Main Grid I,J,K: 20, 20, 16 Z-Scale: 5
Cell Property: TEMP

Statistics: Current Time Step and Visible Cells
 Min      P90      Mean    P10      Max     Sum
 26.8563  51.2006  52.179  53.3736  53.702  333946
Mobile Volume Weighted Mean: 52.2143
```

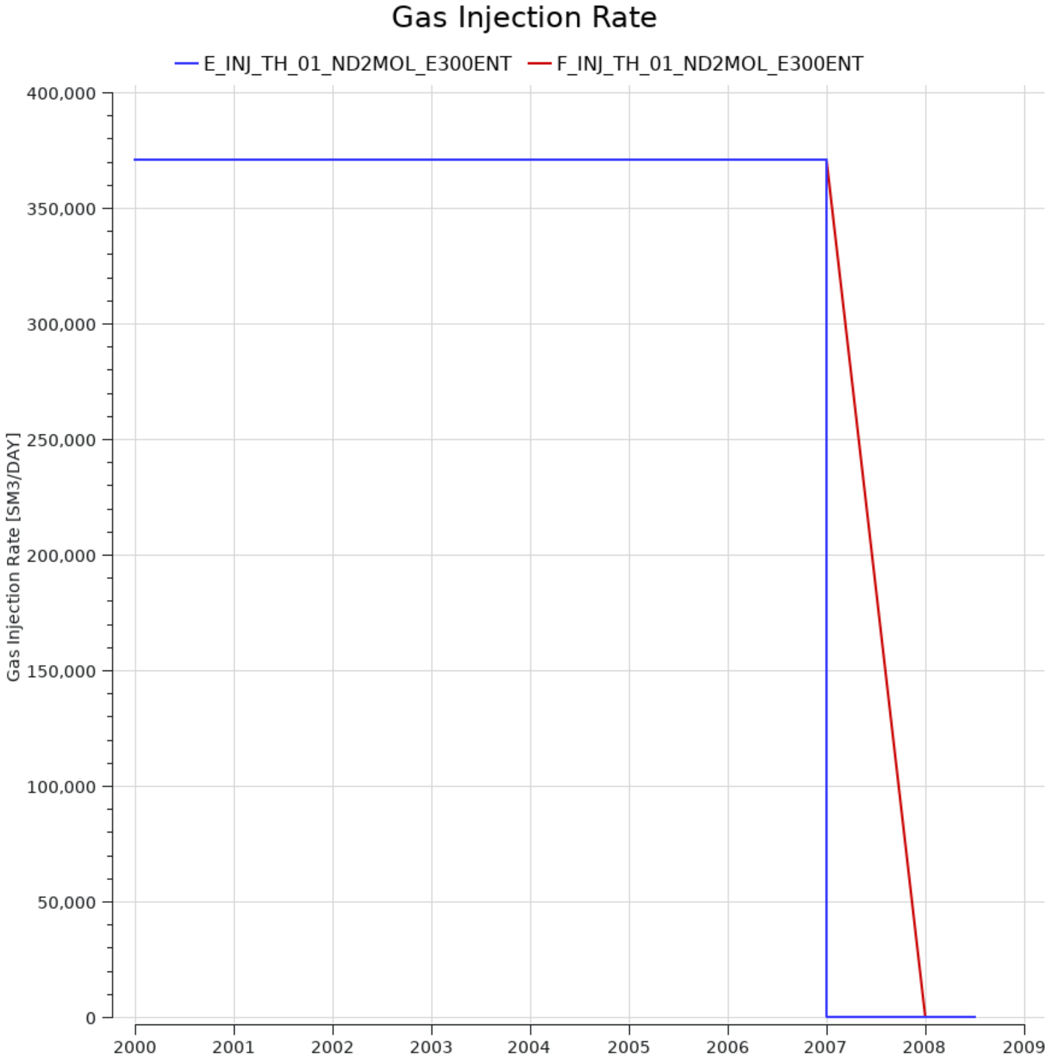


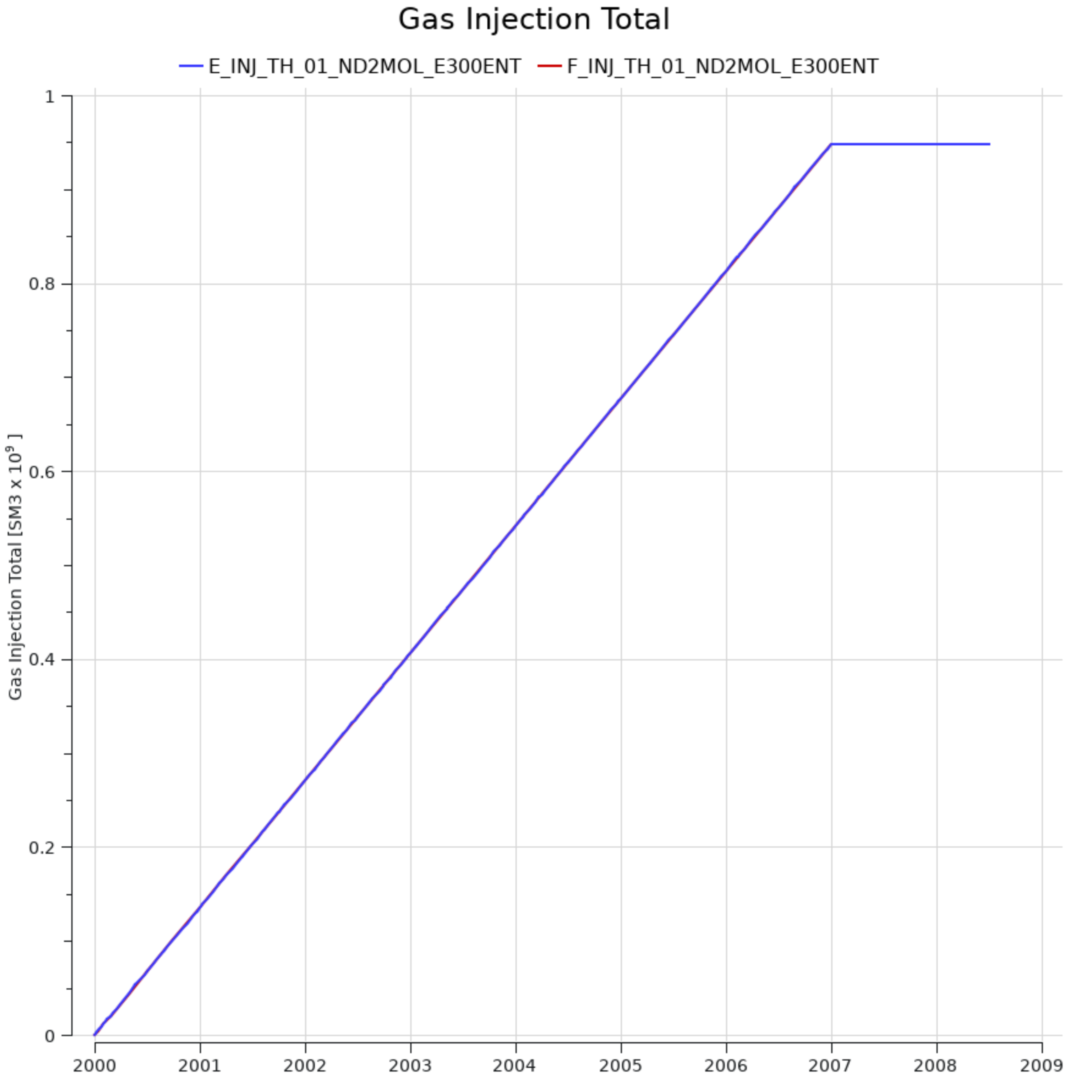
```
Time Step: 17/17 01.Jul 2008

-- E_INJ_TH_01_ND2MOL_E300ENT --

Cell count. Total: 6 400 Active: 6 400
Main Grid I,J,K: 20, 20, 16 Z-Scale: 5
Cell Property: TEMP

Statistics: Current Time Step and Visible Cells
 Min     P90      Mean     P10      Max      Sum
 25.923  51.0605  52.1787  53.4495  53.8527  333944
Mobile Volume Weighted Mean: 52.2072
```

Average pressure



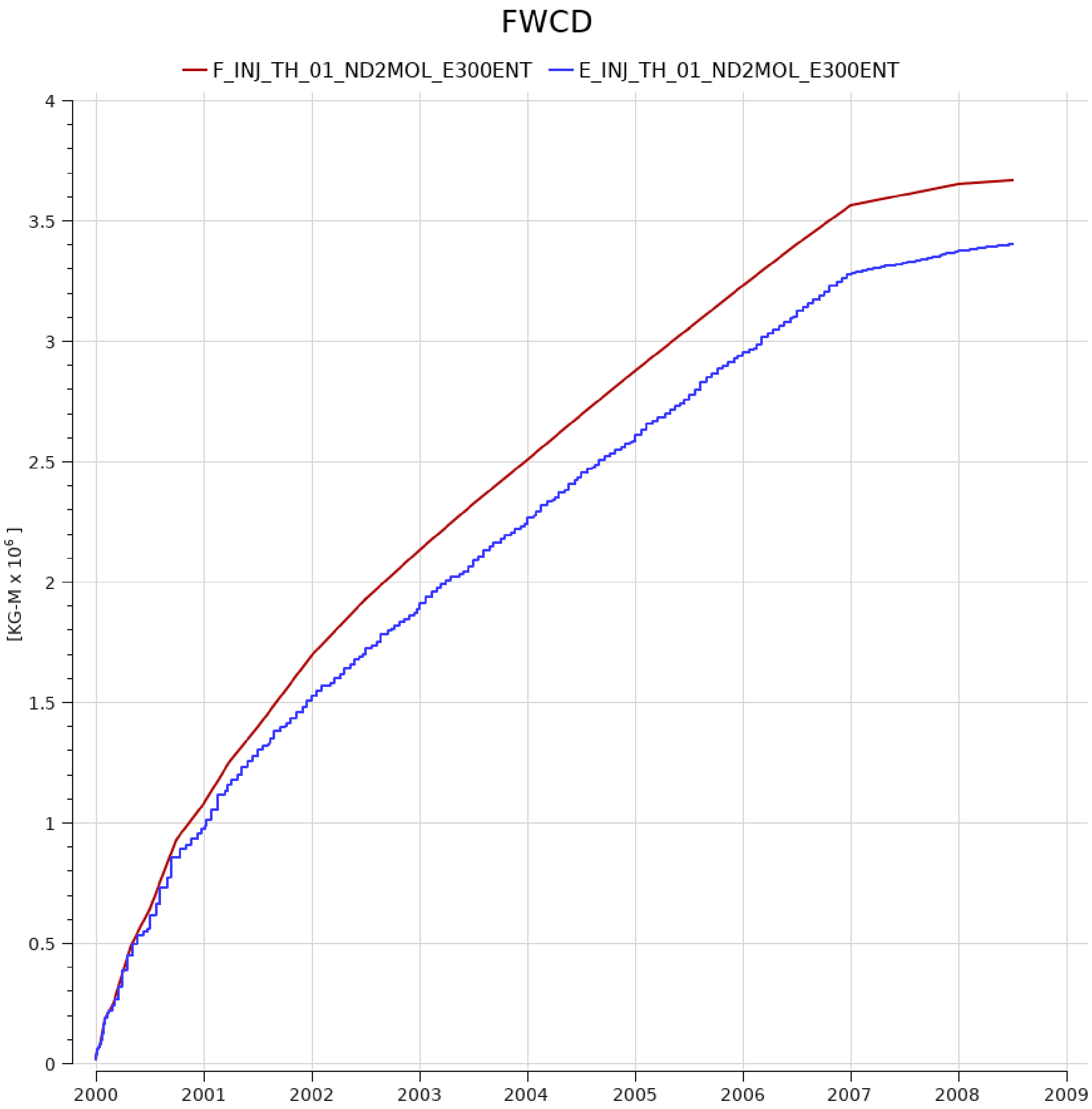
Injection rates





Open

Dissolution



CO2 densities

$$P = \left(\frac{RT_K}{V - b_{mix}}\right) - \left(\frac{a_{mix}}{{T_K}^{1/2} V (V + b_{mix})}\right)$$

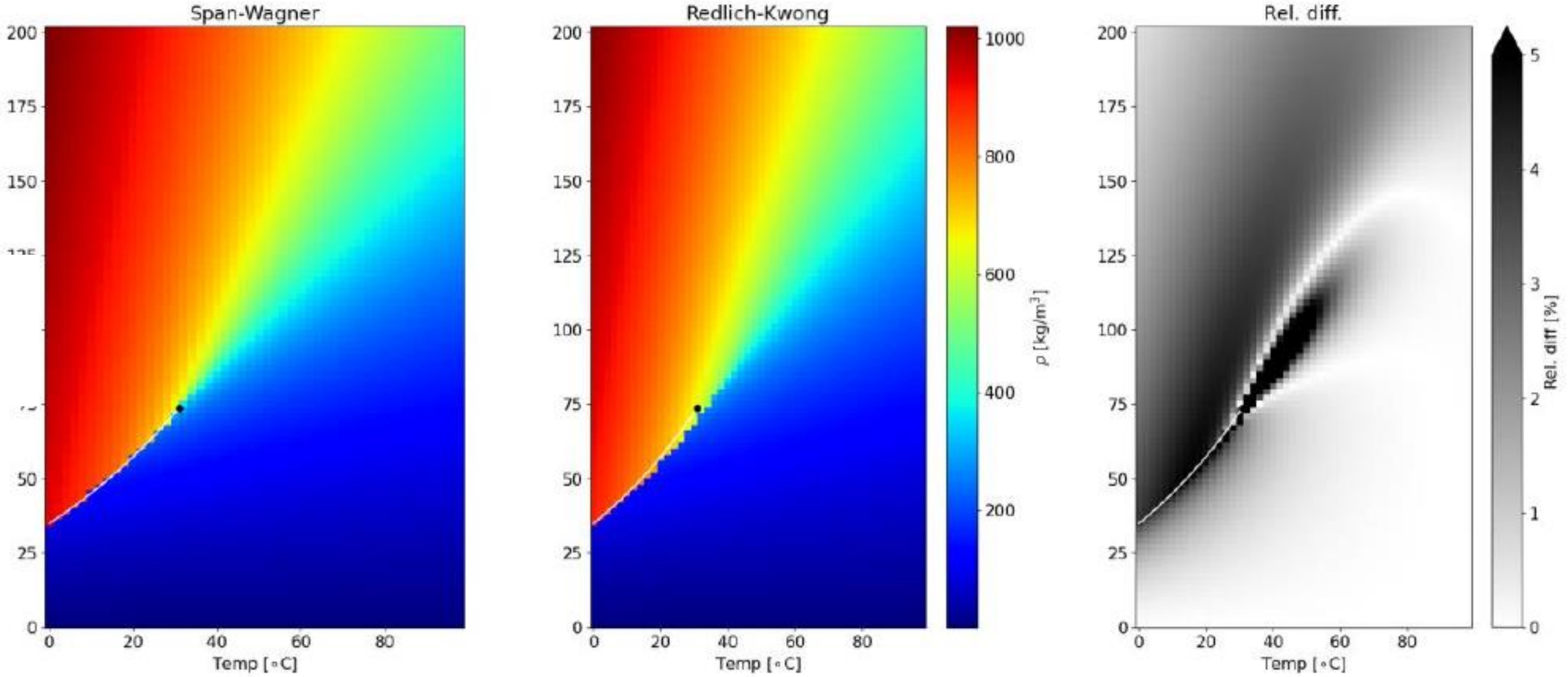
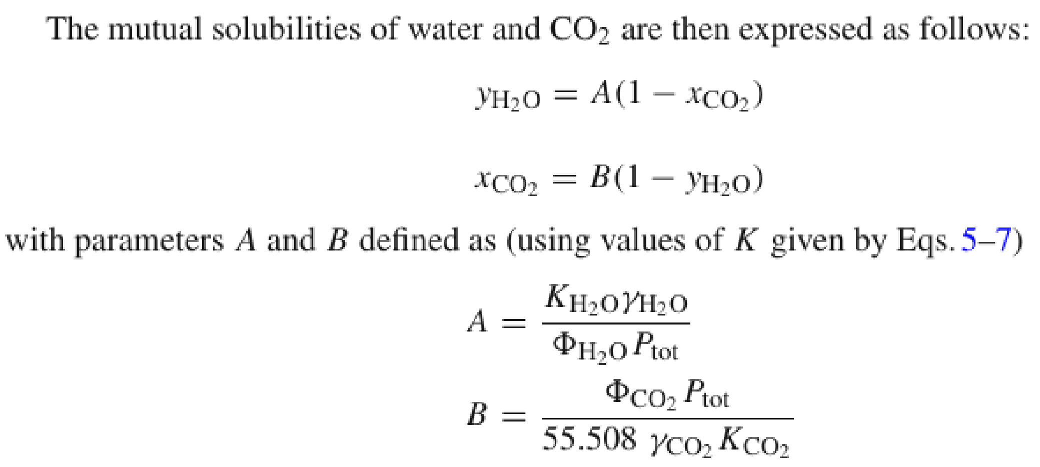


Figure 3. Comparison of density between Span-Wagner used by OPM Flow and Redlich-Kwong.

Solubility model and ACTCO2S



The mutual solubilities of water and CO_2 are then expressed as follows:

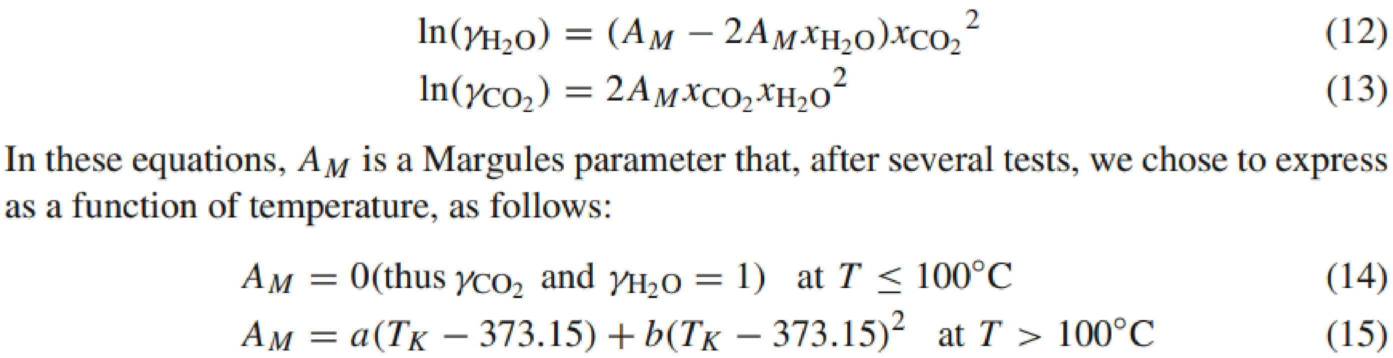
$$y_{H_2O} = A(1 - x_{CO_2})$$

$$x_{CO_2} = B(1 - y_{H_2O})$$

with parameters A and B defined as (using values of K given by Eqs. 5–7)

$$A = \frac{K_{H_2O}\gamma_{H_2O}}{\Phi_{H_2O}P_{tot}}$$

$$B = \frac{\Phi_{CO_2}P_{tot}}{55.508\ \gamma_{CO_2}K_{CO_2}}$$



$$\ln(\gamma_{H_2O}) = (A_M - 2A_M x_{H_2O})x_{CO_2}{}^2 \tag{12}$$

$$\ln(\gamma_{CO_2}) = 2A_M x_{CO_2} x_{H_2O}{}^2 \tag{13}$$

In these equations, A_M is a Margules parameter that, after several tests, we chose to express as a function of temperature, as follows:

$$A_M = 0(\text{thus } \gamma_{CO_2} \text{ and } \gamma_{H_2O} = 1) \quad \text{at } T \leq 100°\text{C} \tag{14}$$

$$A_M = a(T_K - 373.15) + b(T_K - 373.15)^2 \quad \text{at } T > 100°\text{C} \tag{15}$$

Duan and Sun 2003 (Spycher and Pruess 2005)

Rumpf et al. 1994 (Spycher and Pruess 2005)

Spycher and Pruess 2009

The effect of salt

$$\gamma'_{CO_2} = \left(1 + \frac{\sum m_{i \neq CO_2}}{55.508}\right) \exp\{2\lambda(m_{Na} + m_K + 2m_{Ca} + 2m_{Mg})$$
$$+ \xi m_{Cl}(m_{Na} + m_K + m_{Ca} + m_{Mg}) - 0.07\, m_{SO_4}\}$$

$$B' = \frac{\Phi_{CO_2} P_{tot}}{55.508 \gamma_{CO_2} \gamma'_{CO_2} K_{CO_2}}$$

The mutual solubilities of water and CO_2 are then expressed as follows:

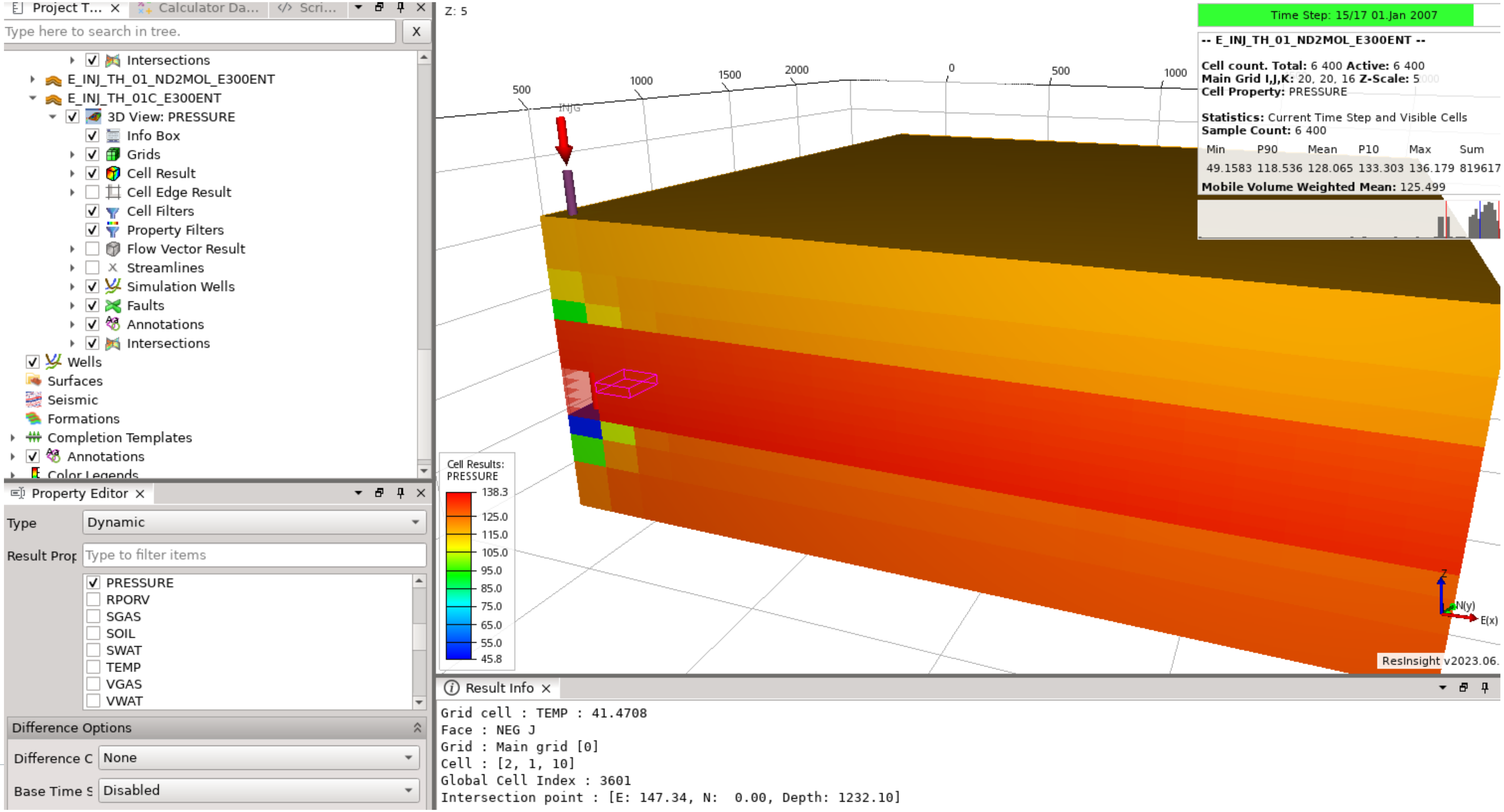
$$y_{H_2O} = A(1 - x_{CO_2})$$

$$x_{CO_2} = B(1 - y_{H_2O})$$

with parameters A and B defined as (using values of K given by Eqs. 5–7)

$$A = \frac{K_{H_2O} \gamma_{H_2O}}{\Phi_{H_2O} P_{tot}}$$
$$B = \frac{\Phi_{CO_2} P_{tot}}{55.508\ \gamma_{CO_2} K_{CO_2}}$$

The models



Project T... ✕ | Calculator Da... | Scri...

Type here to search in tree.

- ☑ Intersections
- E_INJ_TH_01_ND2MOL_E300ENT
- E_INJ_TH_01C_E300ENT
  - ☑ 3D View: PRESSURE
    - ☑ Info Box
    - ☑ Grids
    - ☑ Cell Result
    - ☐ Cell Edge Result
    - ☑ Cell Filters
    - ☑ Property Filters
    - ☐ Flow Vector Result
    - ☐ Streamlines
    - ☑ Simulation Wells
    - ☑ Faults
    - ☑ Annotations
    - ☑ Intersections
- ☑ Wells
- Surfaces
- Seismic
- Formations
- Completion Templates
- ☑ Annotations
- Color Legends

Property Editor ✕

Type: Dynamic

Result Prop: Type to filter items

- ☑ PRESSURE
- ☐ RPORV
- ☐ SGAS
- ☐ SOIL
- ☐ SWAT
- ☐ TEMP
- ☐ VGAS
- ☐ VWAT

Difference Options

Difference C: None

Base Time S: Disabled

Time Step: 15/17 01.Jan 2007

-- E_INJ_TH_01_ND2MOL_E300ENT --

Cell count. Total: 6 400 **Active:** 6 400
Main Grid I,J,K: 20, 20, 16 **Z-Scale:** 5
Cell Property: PRESSURE

Statistics: Current Time Step and Visible Cells
Sample Count: 6 400

Min	P90	Mean	P10	Max	Sum
49.1583	118.536	128.065	133.303	136.179	819617

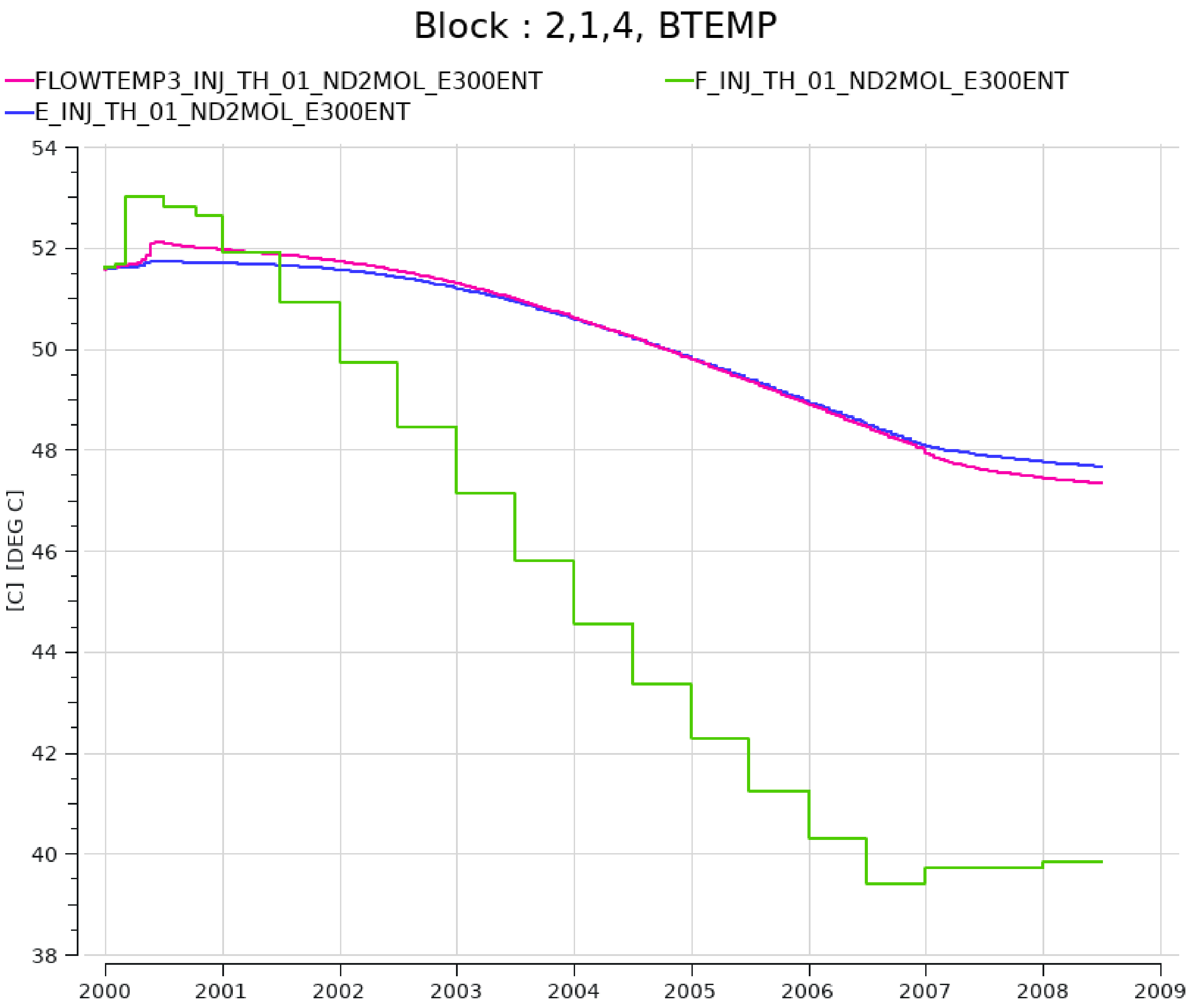
Mobile Volume Weighted Mean: 125.499

Result Info ✕

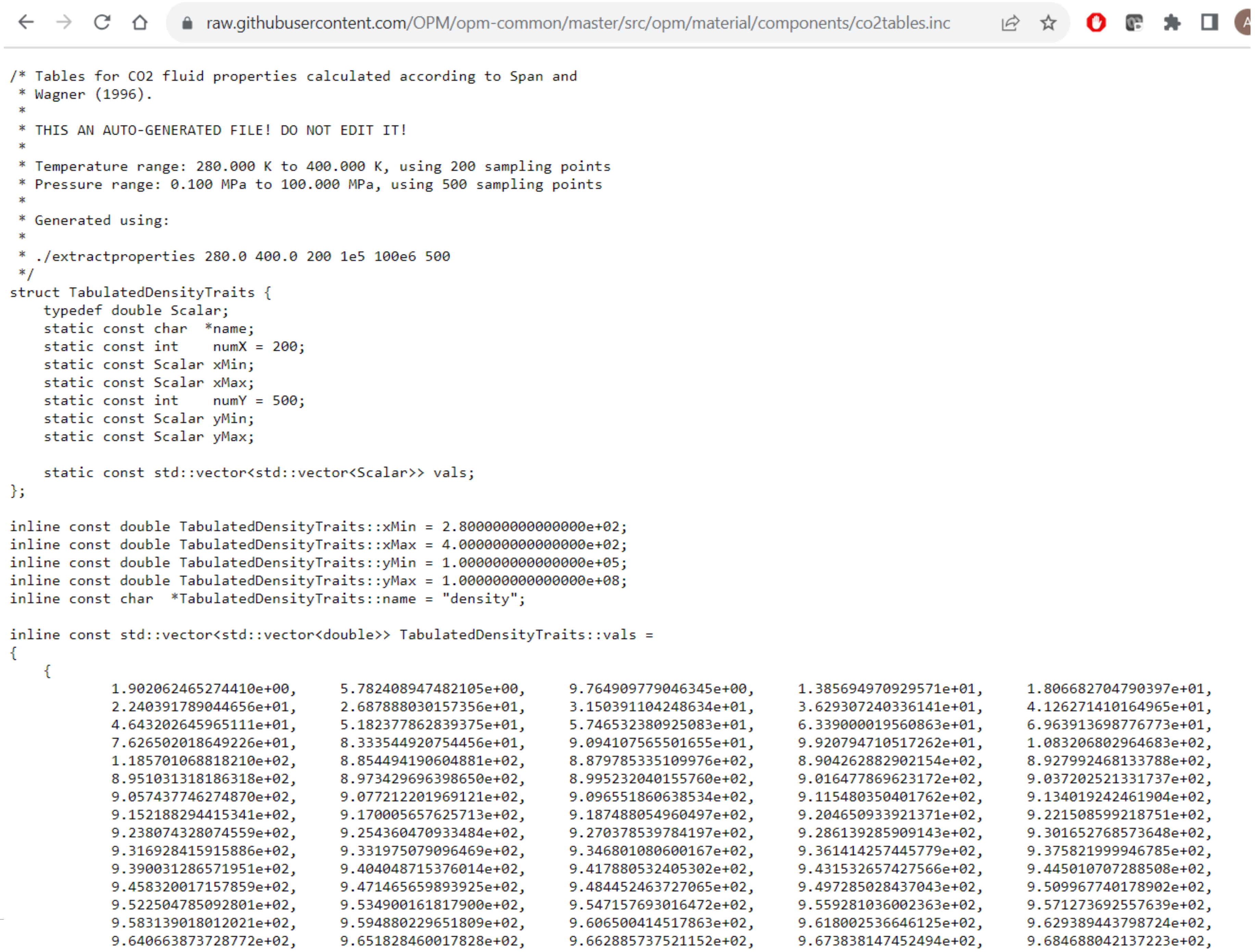
```
Grid cell : TEMP : 41.4708
Face : NEG J
Grid : Main grid [0]
Cell : [2, 1, 10]
Global Cell Index : 3601
Intersection point : [E: 147.34, N:  0.00, Depth: 1232.10]
```

Where we left off

- Deviation between E300/pFlotran and Flow was close to 9 degrees Celsius
- Pflotran was claimed to use Span-Wagner?
- It was commented that temperature could be critical, Sleipner as example

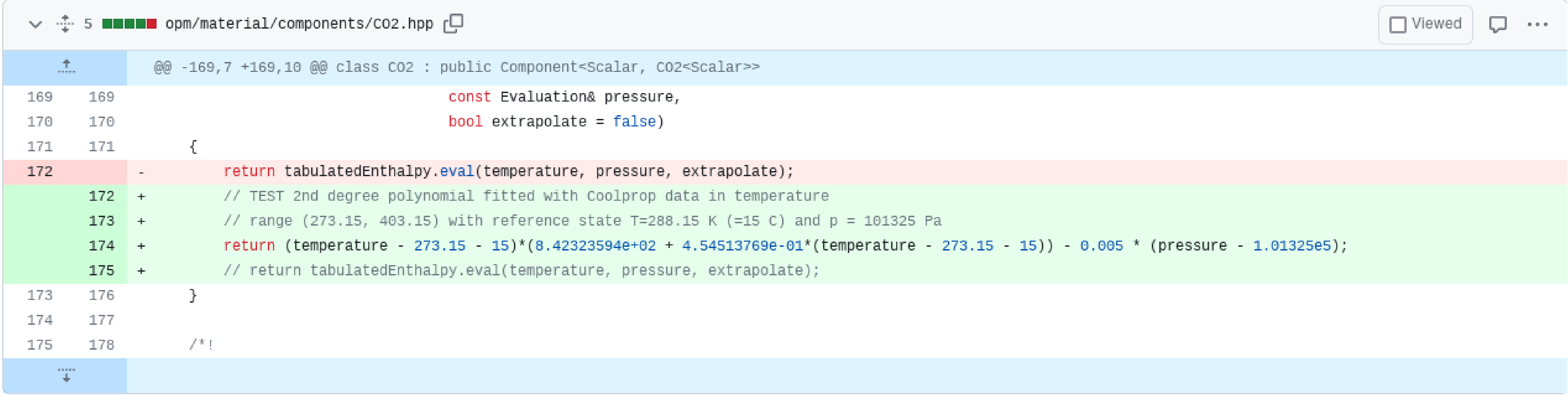


Span & Wagner


raw.githubusercontent.com/OPM/opm-common/master/src/opm/material/components/co2tables.inc

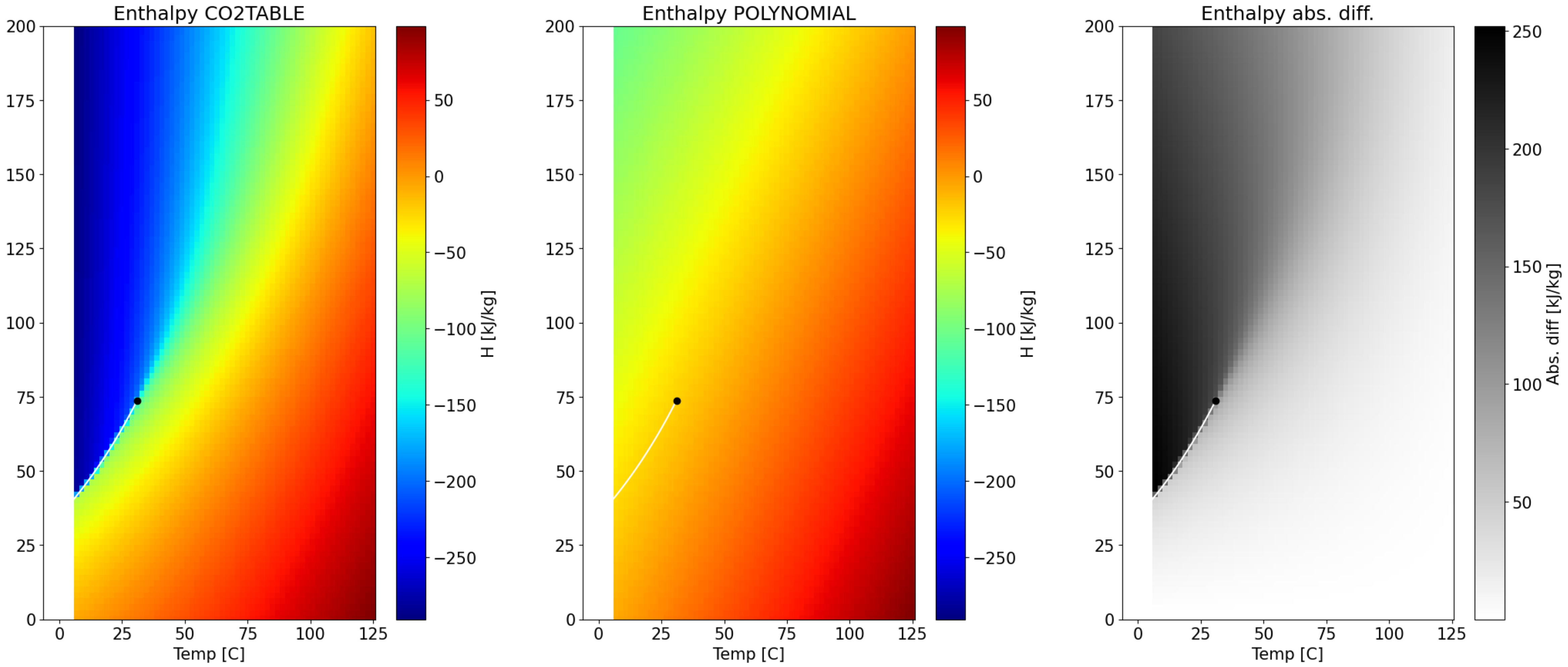
```
/* Tables for CO2 fluid properties calculated according to Span and
 * Wagner (1996).
 *
 * THIS AN AUTO-GENERATED FILE! DO NOT EDIT IT!
 *
 * Temperature range: 280.000 K to 400.000 K, using 200 sampling points
 * Pressure range: 0.100 MPa to 100.000 MPa, using 500 sampling points
 *
 * Generated using:
 *
 * ./extractproperties 280.0 400.0 200 1e5 100e6 500
 */
struct TabulatedDensityTraits {
    typedef double Scalar;
    static const char  *name;
    static const int    numX = 200;
    static const Scalar xMin;
    static const Scalar xMax;
    static const int    numY = 500;
    static const Scalar yMin;
    static const Scalar yMax;

    static const std::vector<std::vector<Scalar>> vals;
};

inline const double TabulatedDensityTraits::xMin = 2.800000000000000e+02;
inline const double TabulatedDensityTraits::xMax = 4.000000000000000e+02;
inline const double TabulatedDensityTraits::yMin = 1.000000000000000e+05;
inline const double TabulatedDensityTraits::yMax = 1.000000000000000e+08;
inline const char  *TabulatedDensityTraits::name = "density";

inline const std::vector<std::vector<double>> TabulatedDensityTraits::vals =
{
    {
            1.902062465274410e+00,    5.782408947482105e+00,    9.764909779046345e+00,    1.385694970929571e+01,    1.806682704790397e+01,
            2.240391789044656e+01,    2.687888030157356e+01,    3.150391104248634e+01,    3.629307240336141e+01,    4.126271410164965e+01,
            4.643202645965111e+01,    5.182377862839375e+01,    5.746532380925083e+01,    6.339000019560863e+01,    6.963913698776773e+01,
            7.626502018649226e+01,    8.333544920754456e+01,    9.094107565501655e+01,    9.920794710517262e+01,    1.083206802964683e+02,
            1.185701068818210e+02,    8.854494190604881e+02,    8.879785335109976e+02,    8.904262882902154e+02,    8.927992468133788e+02,
            8.951031318186318e+02,    8.973429696398650e+02,    8.995232040155760e+02,    9.016477869623172e+02,    9.037202521331737e+02,
            9.057437746274870e+02,    9.077212201969121e+02,    9.096551860638534e+02,    9.115480350401762e+02,    9.134019242461904e+02,
            9.152188294415341e+02,    9.170005657625713e+02,    9.187488054960497e+02,    9.204650933921371e+02,    9.221508599218751e+02,
            9.238074328074559e+02,    9.254360470933484e+02,    9.270378539784197e+02,    9.286139285909143e+02,    9.301652768573648e+02,
            9.316928415915886e+02,    9.331975079096469e+02,    9.346801080600167e+02,    9.361414257445779e+02,    9.375821999946785e+02,
            9.390031286571951e+02,    9.404048715376014e+02,    9.417880532405302e+02,    9.431532657427566e+02,    9.445010707288508e+02,
            9.458320017157859e+02,    9.471465659893925e+02,    9.484452463727065e+02,    9.497285028437043e+02,    9.509967740178902e+02,
            9.522504785092801e+02,    9.534900161817900e+02,    9.547157693016472e+02,    9.559281036002363e+02,    9.571273692557639e+02,
            9.583139018012021e+02,    9.594880229651809e+02,    9.606500414517863e+02,    9.618002536646125e+02,    9.629389443798724e+02,
            9.640663873728772e+02,    9.651828460017828e+02,    9.662885737521152e+02,    9.673838147452494e+02,    9.684688042137223e+02,
```

CO2 Enthalpy main difference



```
@@ -169,7 +169,10 @@ class CO2 : public Component<Scalar, CO2<Scalar>>
                                          const Evaluation& pressure,
                                          bool extrapolate = false)
      {
-         return tabulatedEnthalpy.eval(temperature, pressure, extrapolate);
+         // TEST 2nd degree polynomial fitted with Coolprop data in temperature
+         // range (273.15, 403.15) with reference state T=288.15 K (=15 C) and p = 101325 Pa
+         return (temperature - 273.15 - 15)*(8.42323594e+02 + 4.54513769e-01*(temperature - 273.15 - 15)) - 0.005 * (pressure - 1.01325e5);
+         // return tabulatedEnthalpy.eval(temperature, pressure, extrapolate);
      }

      /*!
```

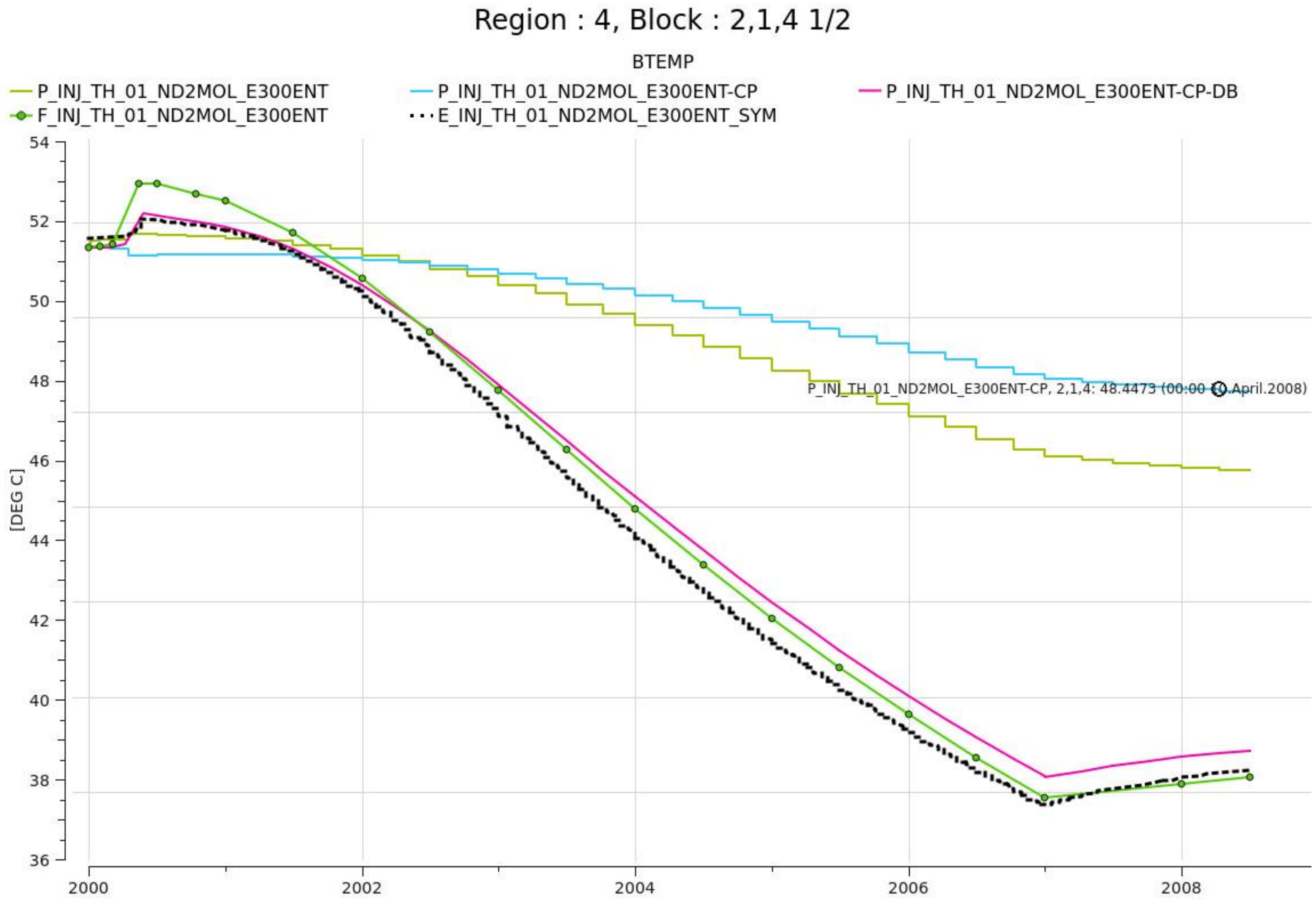
Comparing Enthalpy representations



Comparison with Pflotran and E300

- Pflotran confirmed not to have heat of dissolution

- Both Pflotran and E300 confirms the low temperatures of OPM Flow when Span-Wagner properties for CO2 enthalpy are used

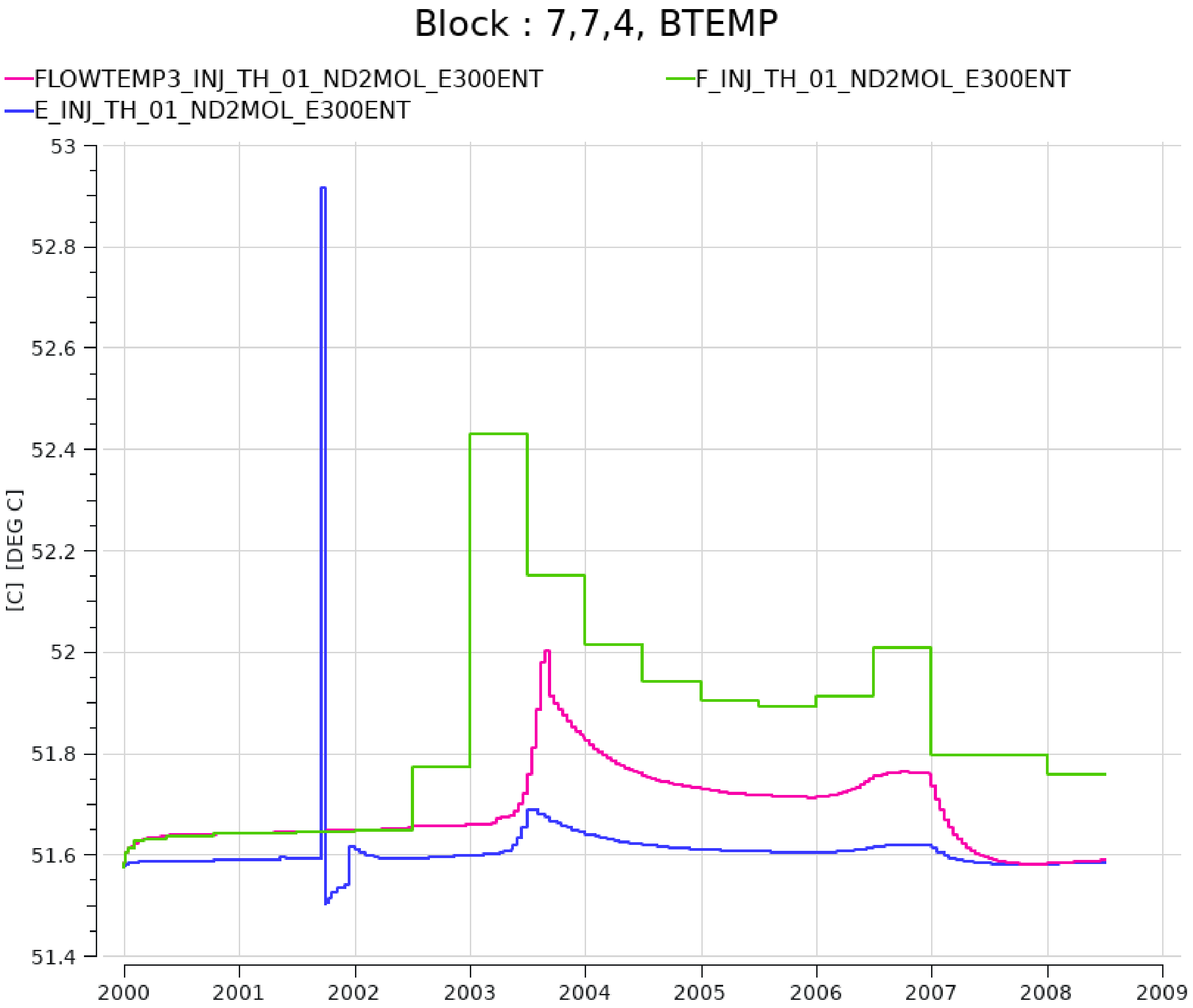


CO2 dissolution secondary effect

- Dissolution effect for CO2 gives a temperatur «bump»

- Removing dissolution model from Flow removes the main difference

- What about temperature effect of vaporization?



Duan & Sun

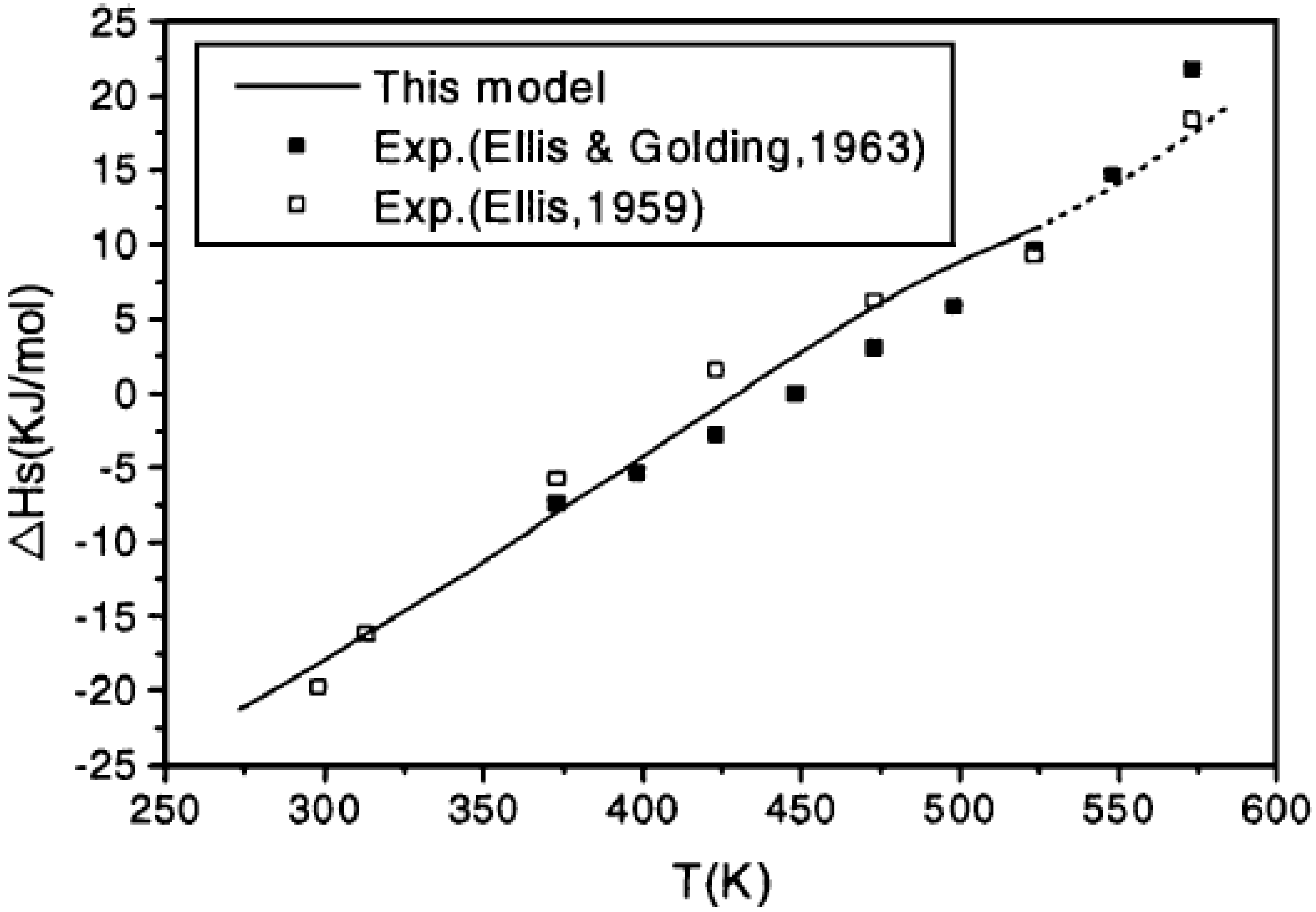
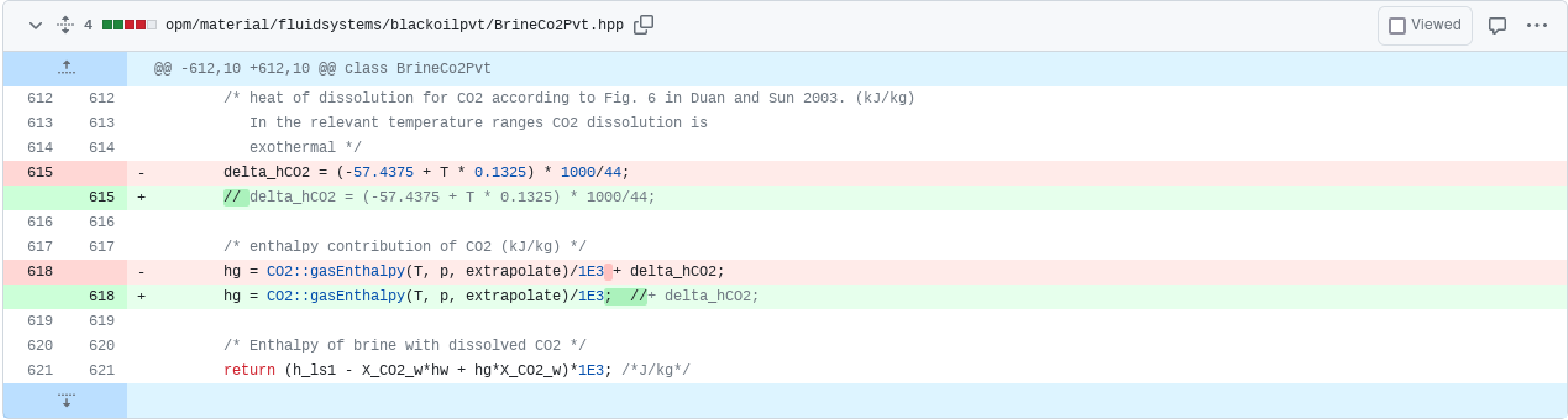


Fig. 6. The heat of solution of CO_2 in water (the model of this study vs. experimental data).

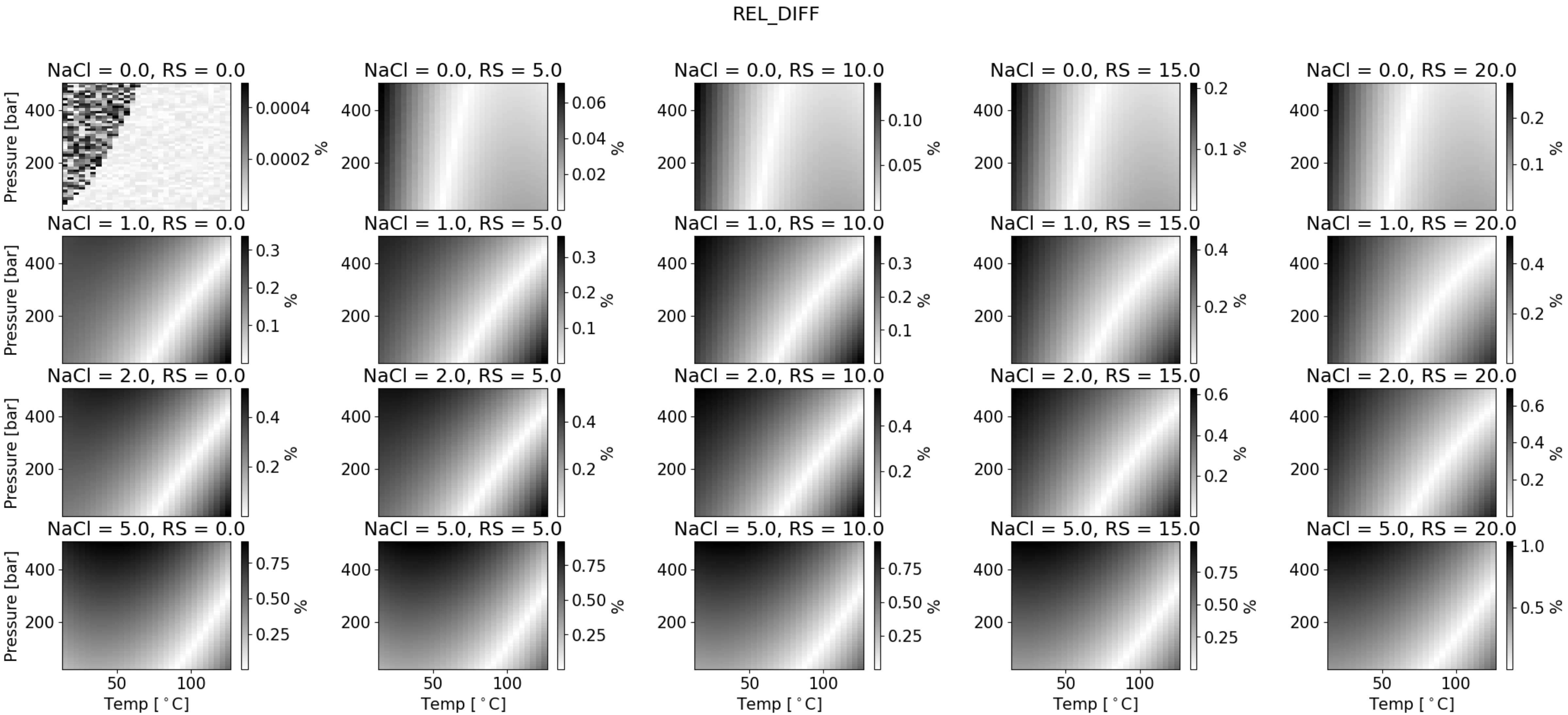
CO2 Dissolution in Flow



opm/material/fluidsystems/blackoilpvt/BrineCo2Pvt.hpp

```diff
@@ -612,10 +612,10 @@ class BrineCo2Pvt
         /* heat of dissolution for CO2 according to Fig. 6 in Duan and Sun 2003. (kJ/kg)
            In the relevant temperature ranges CO2 dissolution is
            exothermal */
-        delta_hCO2 = (-57.4375 + T * 0.1325) * 1000/44;
+        // delta_hCO2 = (-57.4375 + T * 0.1325) * 1000/44;

         /* enthalpy contribution of CO2 (kJ/kg) */
-        hg = CO2::gasEnthalpy(T, p, extrapolate)/1E3 + delta_hCO2;
+        hg = CO2::gasEnthalpy(T, p, extrapolate)/1E3;  //+ delta_hCO2;

         /* Enthalpy of brine with dissolved CO2 */
         return (h_ls1 - X_CO2_w*hw + hg*X_CO2_w)*1E3; /*J/kg*/
```

Water density (diff with Ezrkohi, RS is CO2 content)



Density models for saline water in E300, Intersect, Pflotran and Flow

- Flow uses Garcia for CO2 solution impact on water density and Batzle & Wang salinity impact on density of water, ref. https://library.seg.org/doi/10.1190/1.1443207 https://www.osti.gov/biblio/790022

- E300 and Intersect uses Ezrokhi for both (CO2 solution and salinity impact on water density). Model described in book: Zaytsev, I.D., Aseyev, G.G.: Properties of Aqueous Solutions of Electrolytes 1993.

- Pflotran supports Batzle & Wang in addition to Ezrokhi for density of saline water, while the CO2 solution impact is from Duan & Sun, ref. https://docs.opengosim.com/manual/input_deck/thermodynamic_props/eos_water/

- Note that Batzle & Wang is more accurate than Ezrokhi. However, as implemented in Flow it only taks NaCl into account, will need to be extended to take other salts into account. While Ezrokhi allows to manually provide parameters to the model to account for salt combination.