# My Life with Wolfram: The Beginnings of AFa

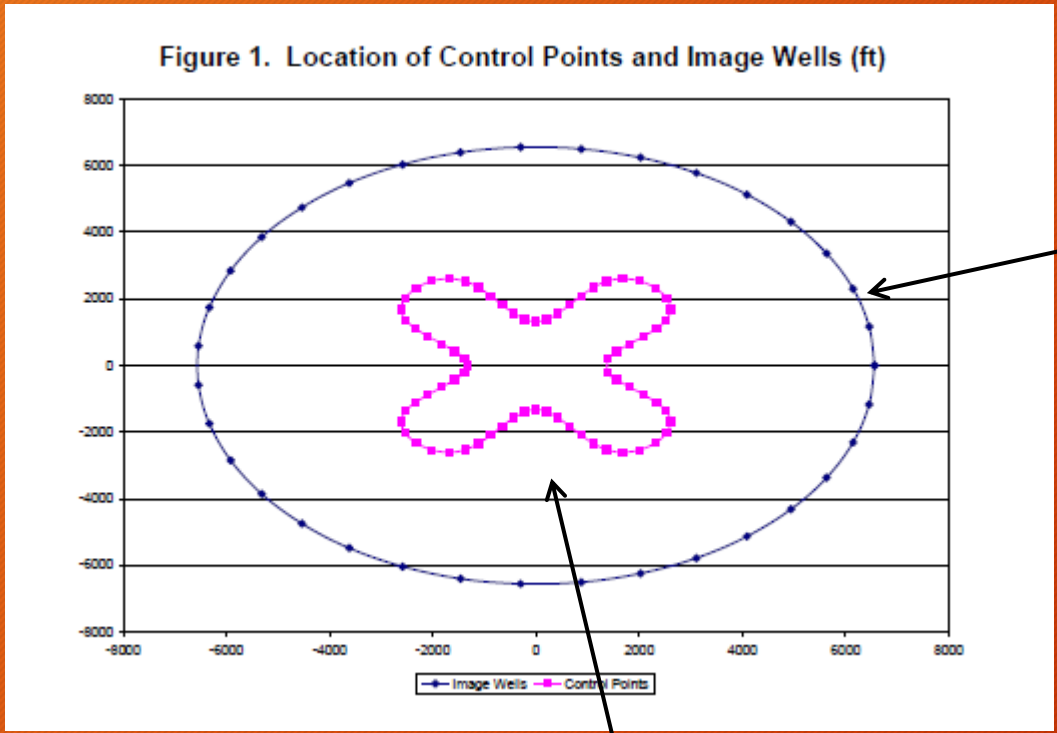
Applications to petroleum engineering ☺

Colin Lyle Jordan, P. Eng
colinlylejordan@gmail.com

January 2020

# How I found the Wolfram Language

"Pseudo" Image Wells to Create X-Shaped Boundary

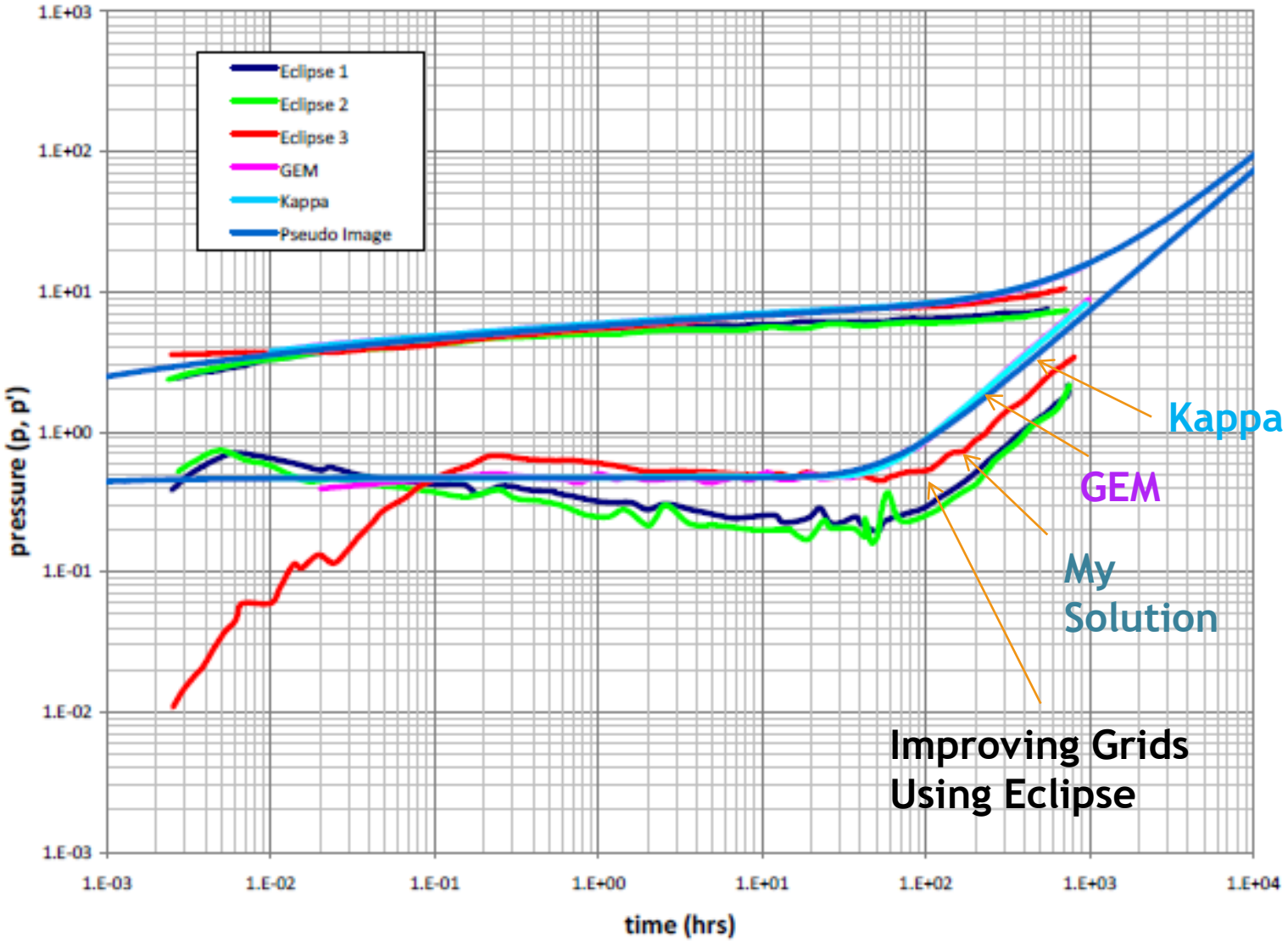



Reservoir

My first real introduction into the Wolfram Language was the development of my personal gas simulator based on using fixed image wells (or pseudo image wells) whose rates were adjusted to create boundaries and/or heterogeneities as desired.

At the time, some friends and I tested it against other simulators such as Rubis by Kappa and Eclipse by Sclumberger (see next slide)

Figure 2. Numerical Validation Example (Archer)




Following the work of Rosalind Archer who researched BEM, we evaluated a well in a closed square reservoir (production well located centrally). In Archers work, simulation runs were evaluated for a number of cells and non-uniform gridding (the same case was also evaluated using the Kappa non-linear simulator).

Most of this work was prototyed in Wolfram!

# Wolfram Basic Concept

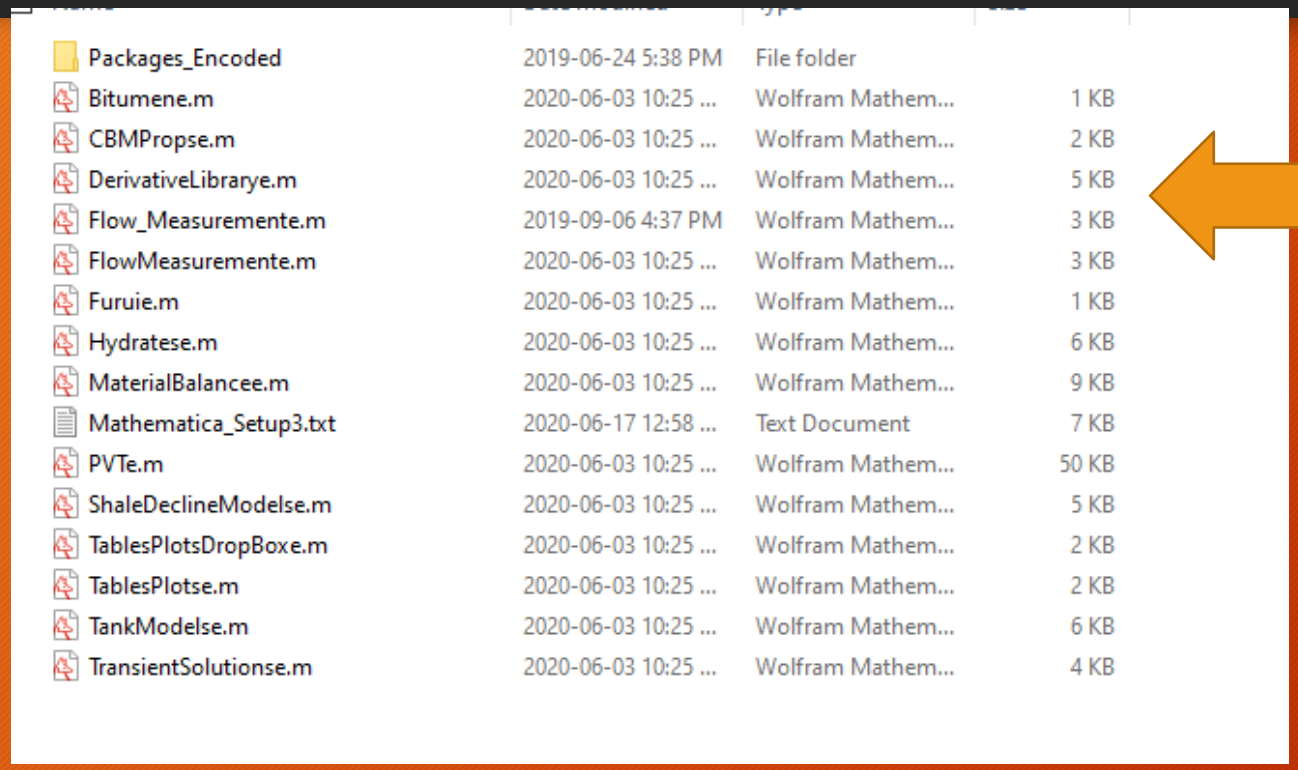
A “Package” is a like a library, which allows reusable code for various applications, and so on. It can be deployed locally or stored in the cloud

A “notebook” is the basic interface. Code and HMI can be written here, and deployed to desktop or cloud.

Manipulate the basic HMI for creating rich interactive content

Wolfram Libraries & Functions.
Anything from default plotting tools through to numerous advanced computation

# Most Calculations are stored in Packages

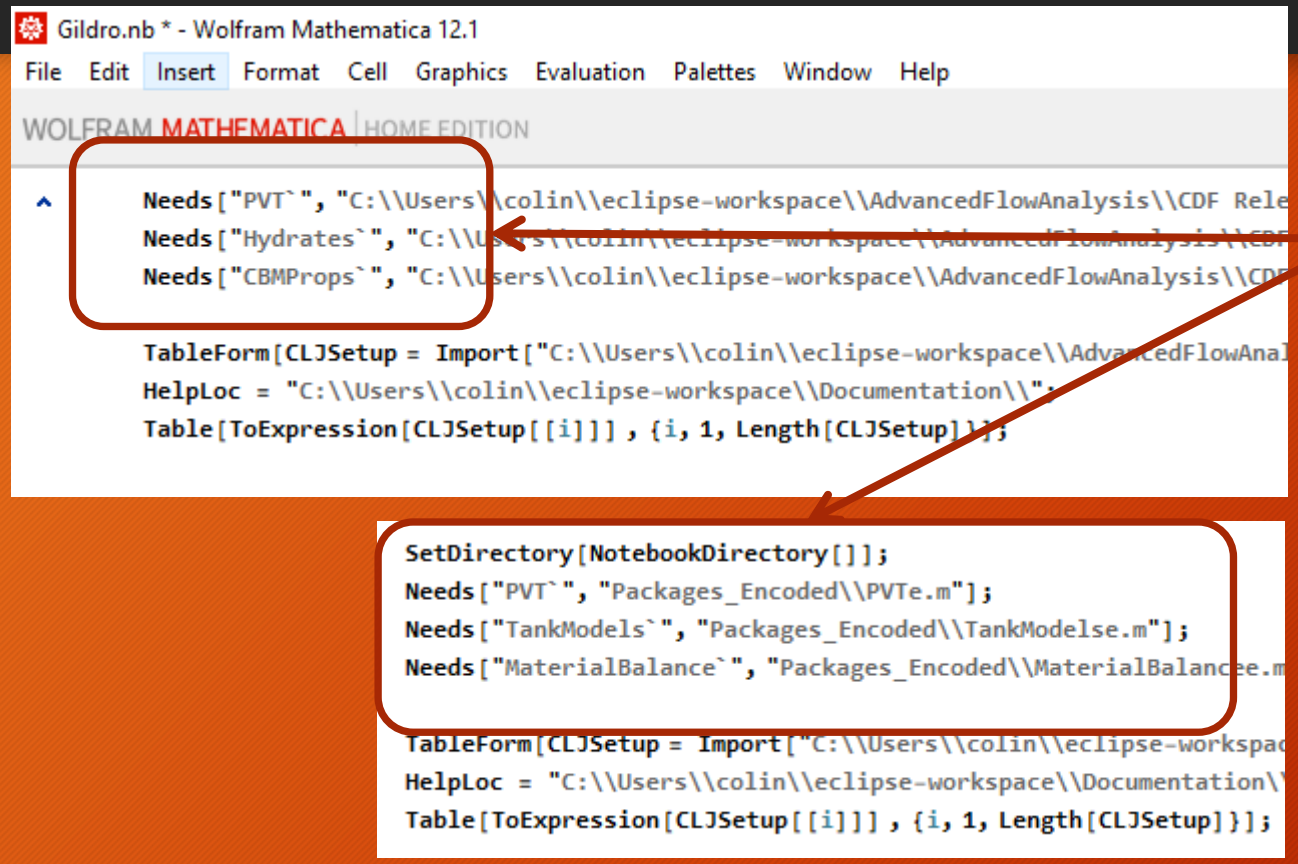


| | | | |
|---|---|---|---|
| Packages_Encoded | 2019-06-24 5:38 PM | File folder | |
| Bitumene.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 1 KB |
| CBMPropse.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 2 KB |
| DerivativeLibraye.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 5 KB |
| Flow_Measuremente.m | 2019-09-06 4:37 PM | Wolfram Mathem... | 3 KB |
| FlowMeasuremente.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 3 KB |
| Furuie.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 1 KB |
| Hydratese.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 6 KB |
| MaterialBalancee.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 9 KB |
| Mathematica_Setup3.txt | 2020-06-17 12:58 ... | Text Document | 7 KB |
| PVTe.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 50 KB |
| ShaleDeclineModelse.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 5 KB |
| TablesPlotsDropBoxe.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 2 KB |
| TablesPlotse.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 2 KB |
| TankModelse.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 6 KB |
| TransientSolutionse.m | 2020-06-03 10:25 ... | Wolfram Mathem... | 4 KB |


Libraries or Packages used for Custom Applications

Functions / algorithms inside Packages

# How Packages Work







I use “Needs” to import my custom libraries into the primary notebook where I use variety wolfram HMI functionality to create my applications as needed.

Packages can be loaded from Cloud or Desktop

From this point forward, one only needs to build the interfaces as they see fit.

# “Manipulate”

In this example, shown on the following page, an interactive isotherm (both 2D and 3D) is generated within the Manipulate

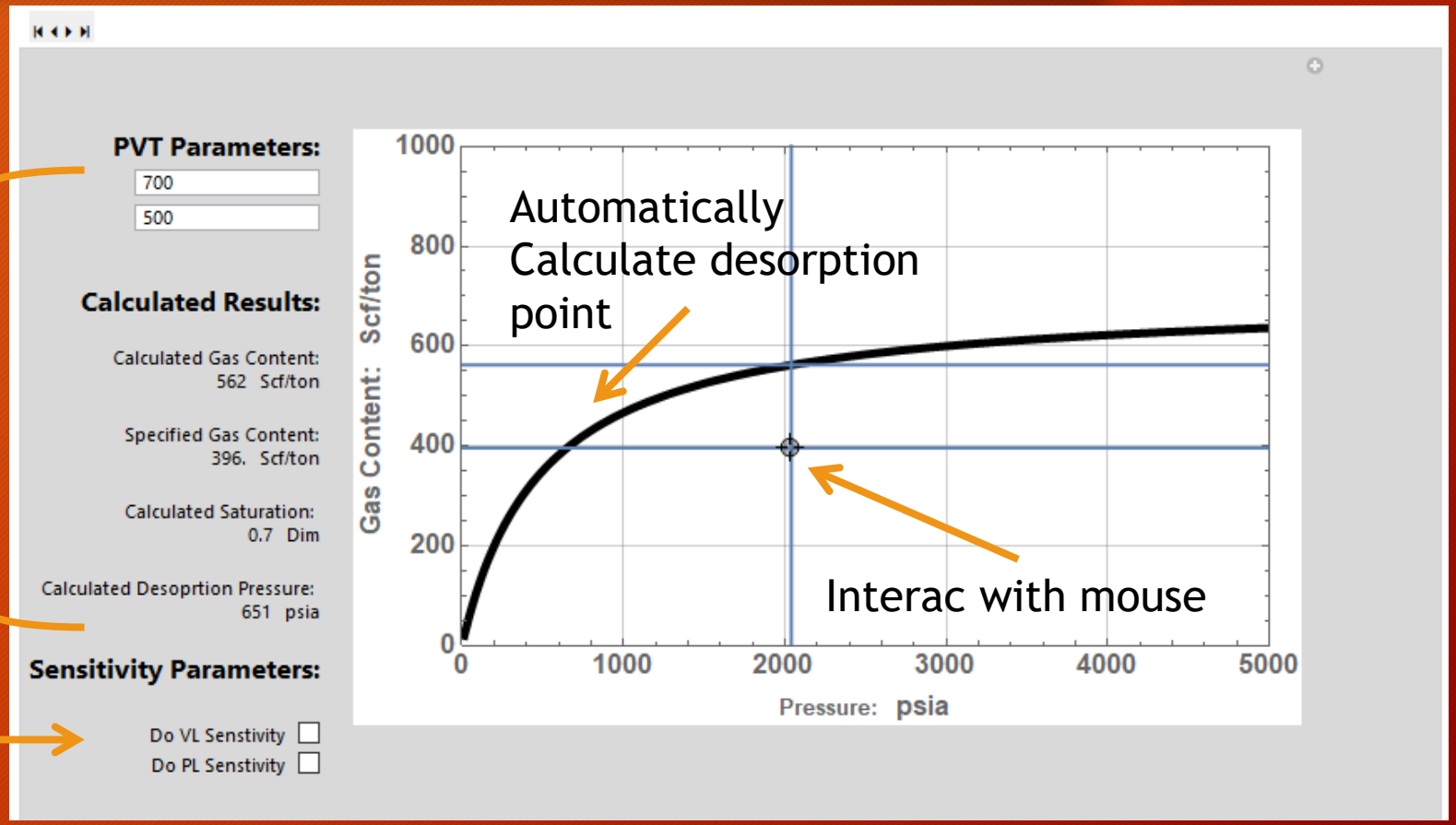
- Manipulate basically gives us pre-packaged interactivity along which can be combined with any number of simple or complex engineering functions!

```
IsothermPlot2D =
    Manipulate[

    Module[{},

        GCcurve=CBMProps`GasContent[PrGCPoint[[1]], VL, PL];(*test point calculation*)
        Pdesorp=CBMProps`DesorptionPressure[PrGCPoint[[2]], VL, PL];
        (*isothermplots3D=Plot3D[CBMProps`GasContent[1000, VL, pl],{pl,14.7,Pmax},{vl,10,VLmax
            AxesLabel -> {Style["PL",FL,FLF], Style["VL",FL,FLF], Style["Gas Content",FL,FLF]}
            ImageSize→PlotSize,
            ColorFunction→ "RustTones"
        ];*)

        isothermplots=
        Show[
        Plot[
            {
            CBMProps`GasContent[p, VL, PL],
            If[DoPLPlot,
            Table[CBMProps`GasContent[p, VL, pls],{pls, 14.7, Pmax, 100}]
            ],
            If[DoVLPlot,
```

# Entire Isotherm Analysis is contained within Manipulate

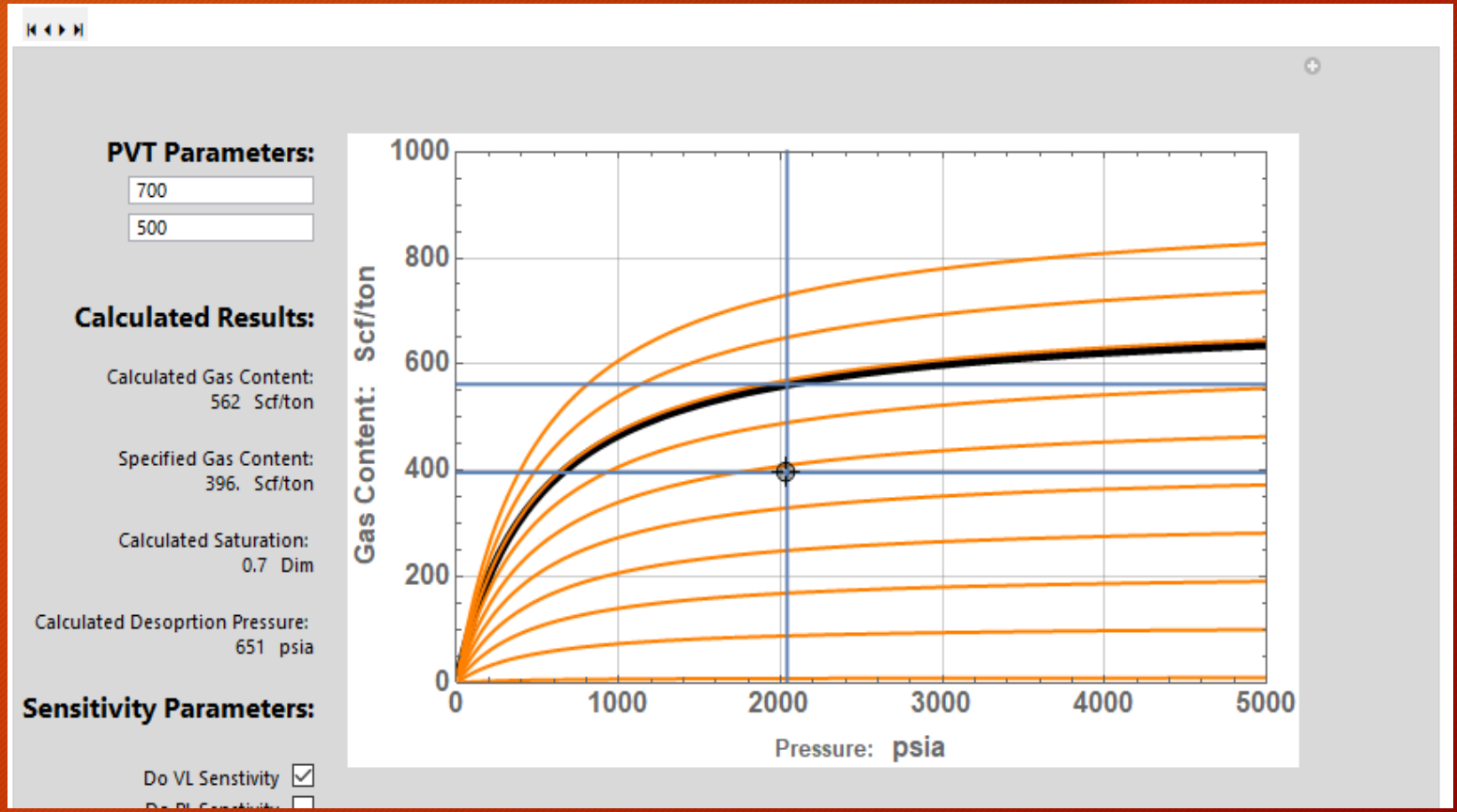
Interactively, one can maneuver the reservoir pressure with the mouse and all calculations are updated automatically

We are able to run sensitivities of isotherms relative to both Langmuir Parameters



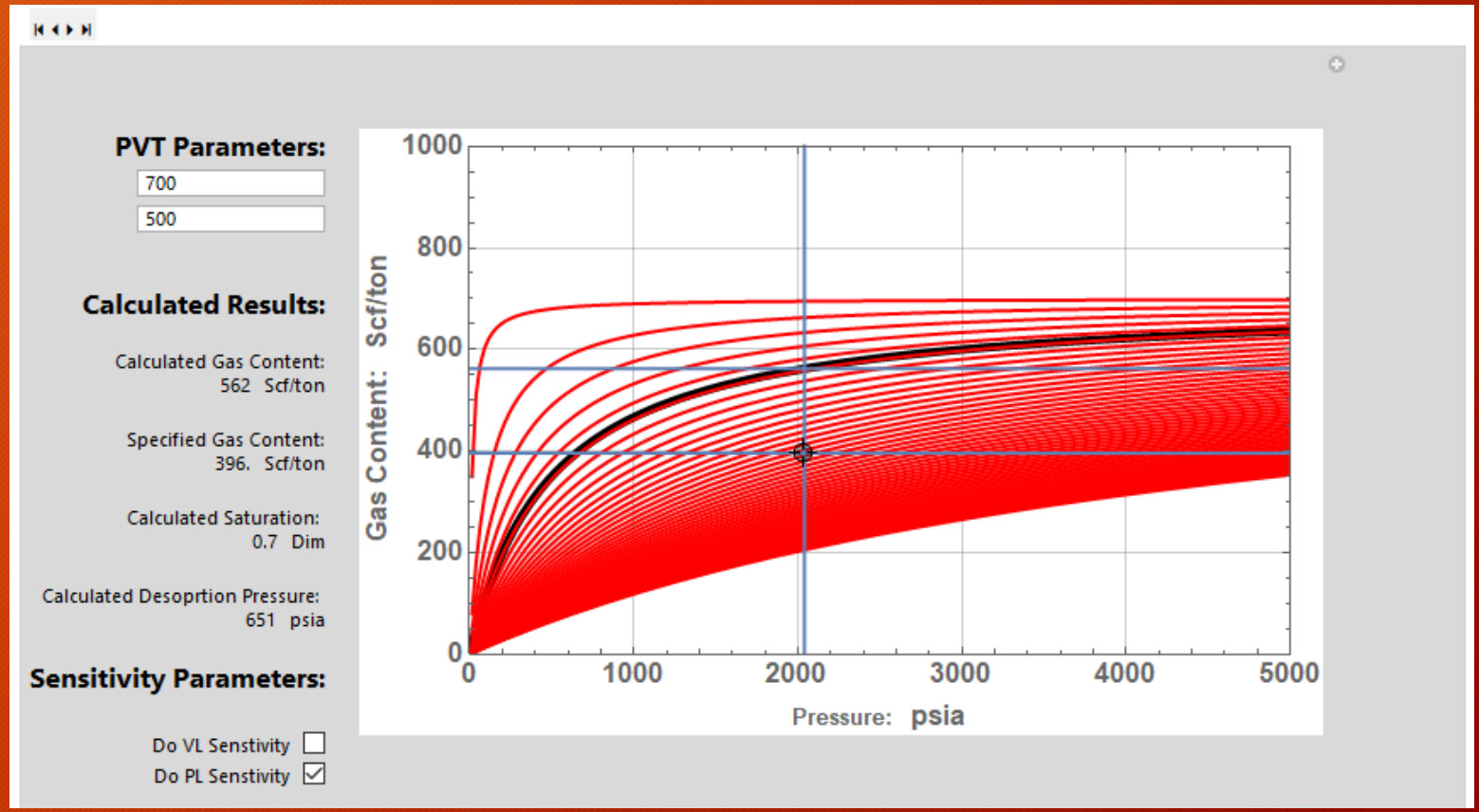
# Sensitivities to VL!

In later slides, you can see a full description of the development of Monte Carlo Simulation using the Wolfram Language

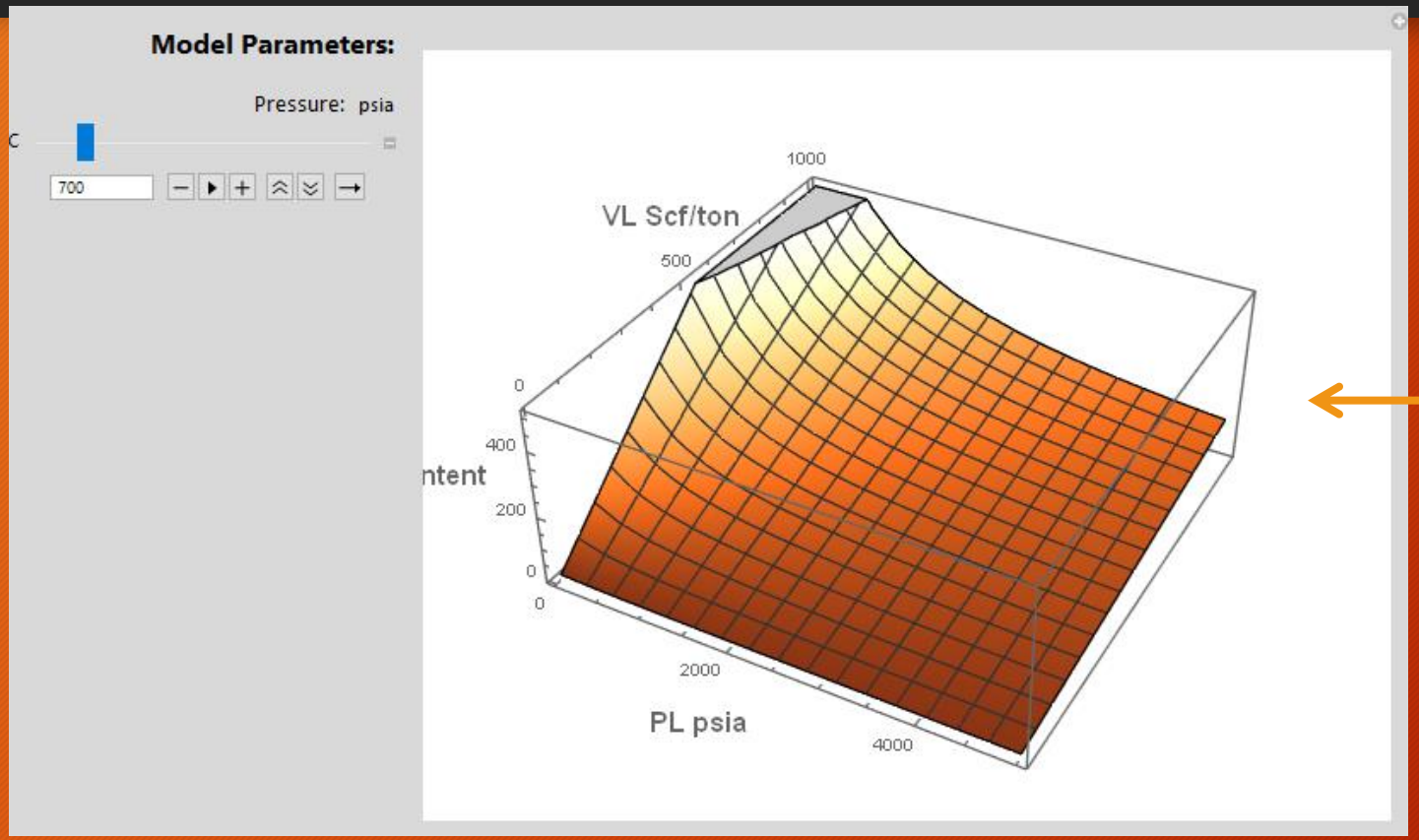


# Sensitivities to PL!

The Wolfram Language provides a great set of tools for rapiding and coding of algorithms and related models



# 3D Interactive Plotting

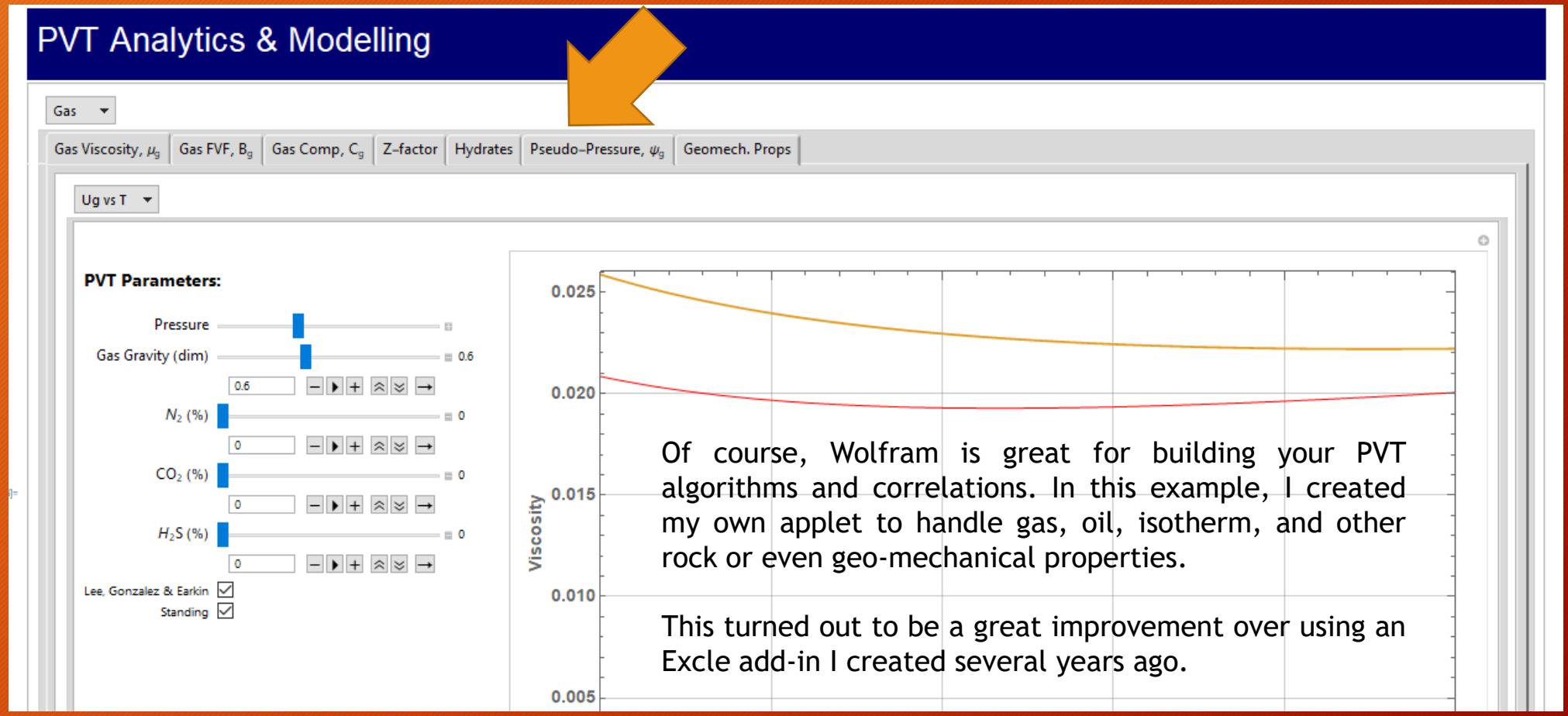



In this scenario, I used Wolfram Language to create interactive 3D Modelling.

# Creating PVT Calculators with Wolfram Language

# Custom Fluid Property Calculations

Each tab represents a “Manipulate” contained within a “TabView”

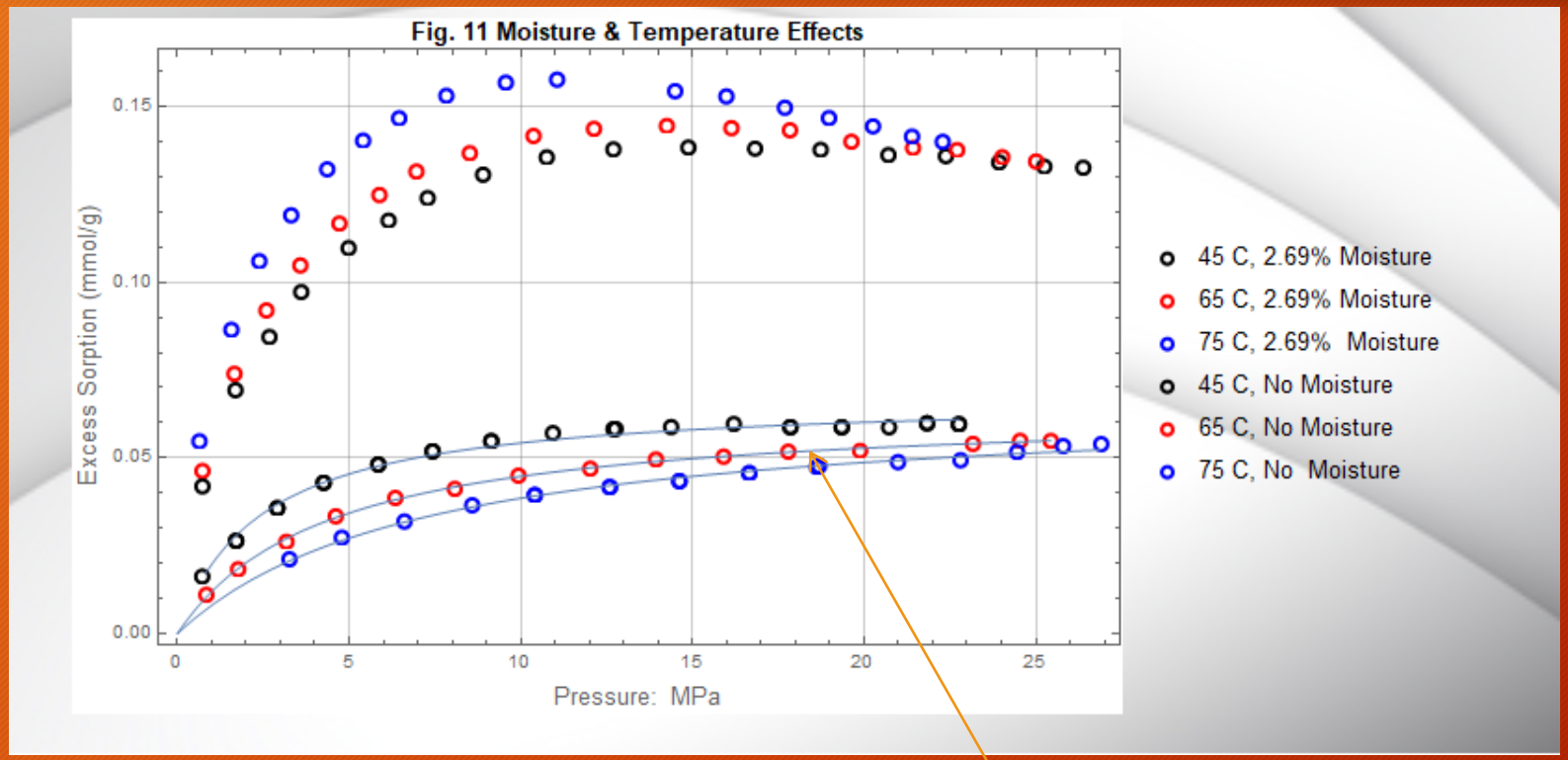


# Extended Examples of Wolfram Usage

$$\text{PRS5GAS5Sol} = \text{FindFit}\left[\text{PRS5GAS5}, \frac{x\,\text{VL5a}}{x + \text{PL5}}, \{\text{VL5a}, \text{PL5}\}, x\right];$$

PRS5GAS5 is data from excel data stored externally which is auto-imported and fitted to the Langmuir Equation

# Analyzing Isotherms! And Wolfram Cloud




In this example, I import tables of excel based isotherm data which is automatically history matched for Langmuir parameters (VL & PL).

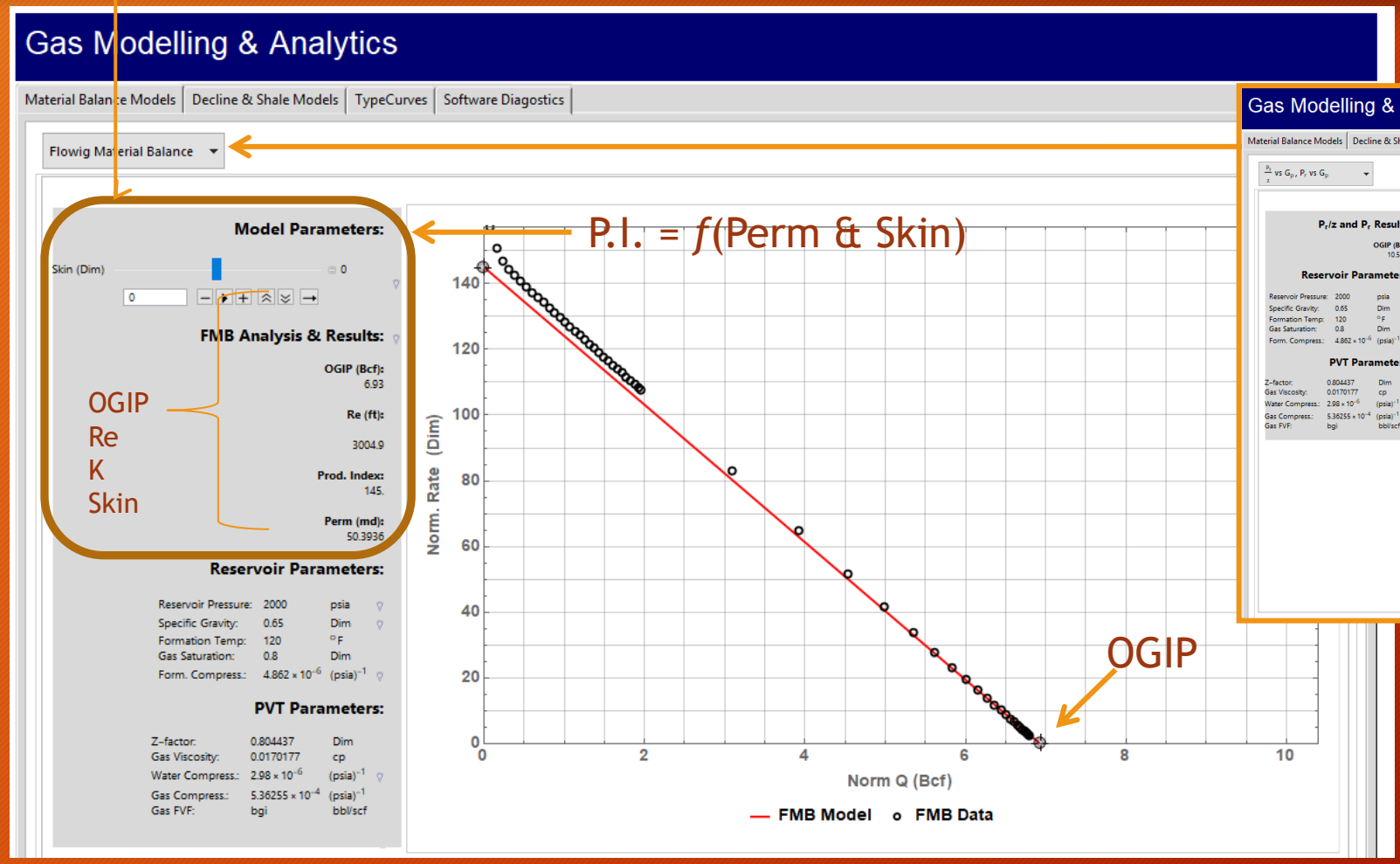
From there, such parameters can be written back to a cloud based historian and automatically imported into CBM production / forecasting models

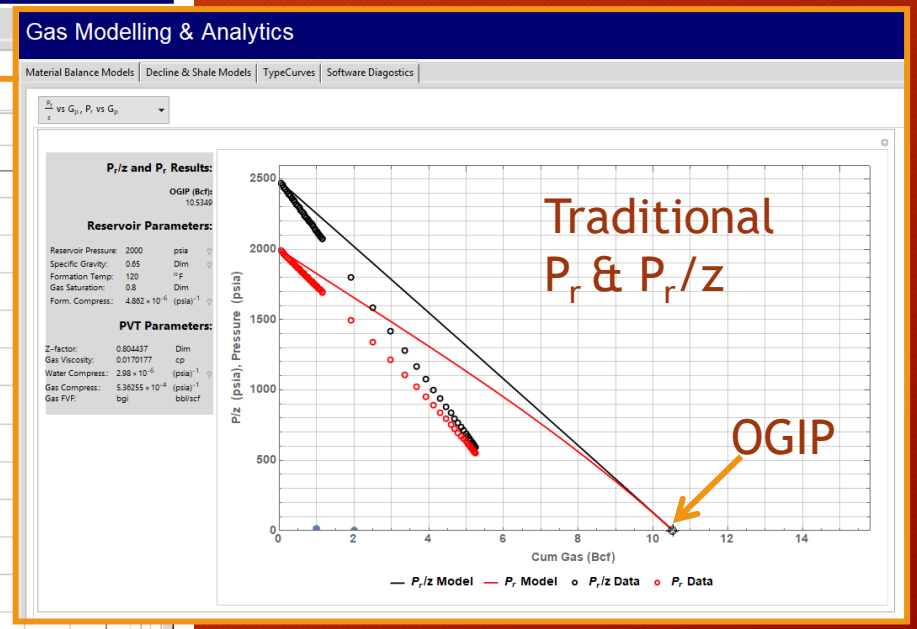
Data is grabbed from online source and automatically fitted to the traditional Langmuir Equation

# FMB Calculations

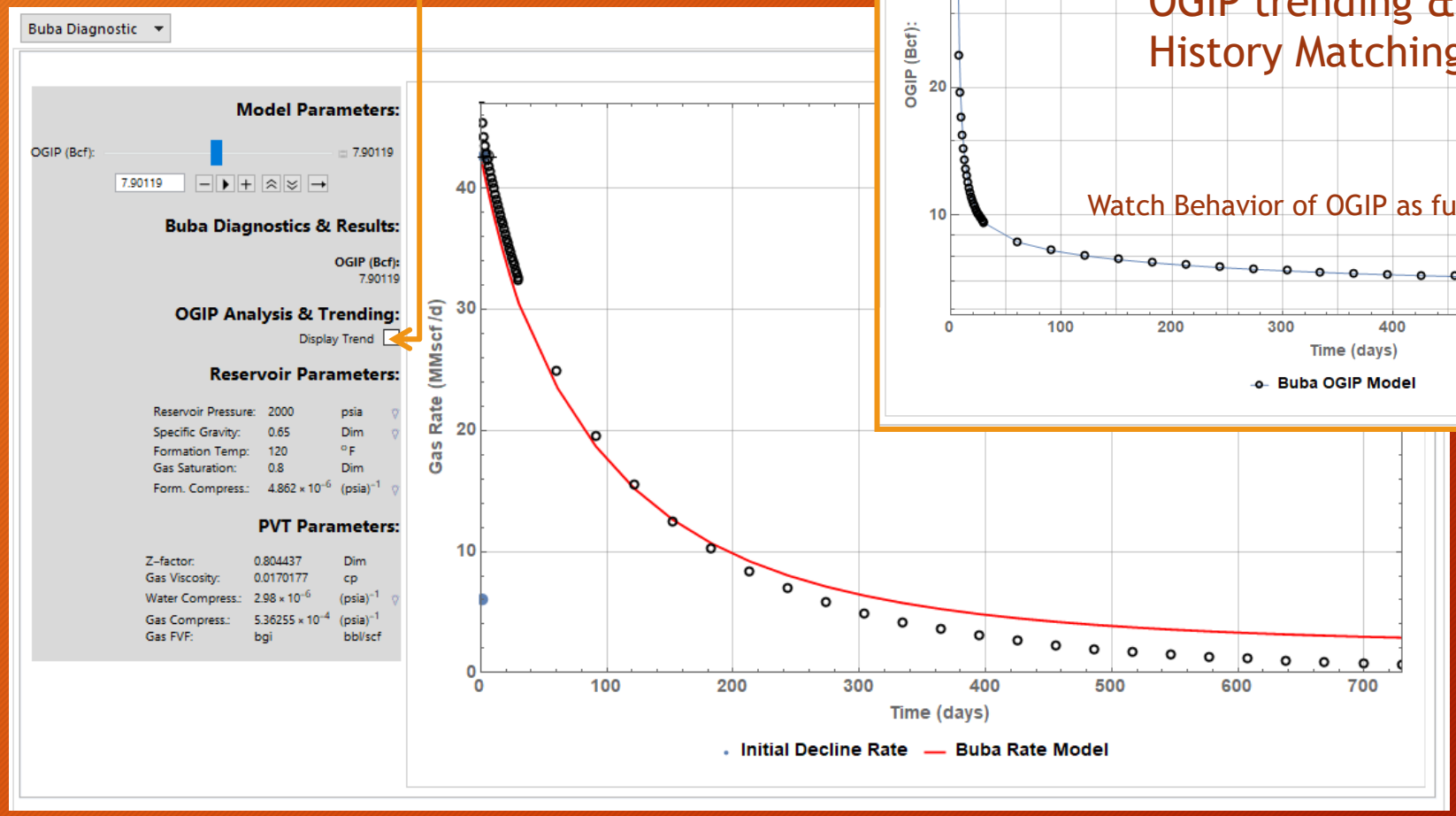
In addition to traditional OGIP estimation, the modelling process can be used to extract permeability based on estimated skin.

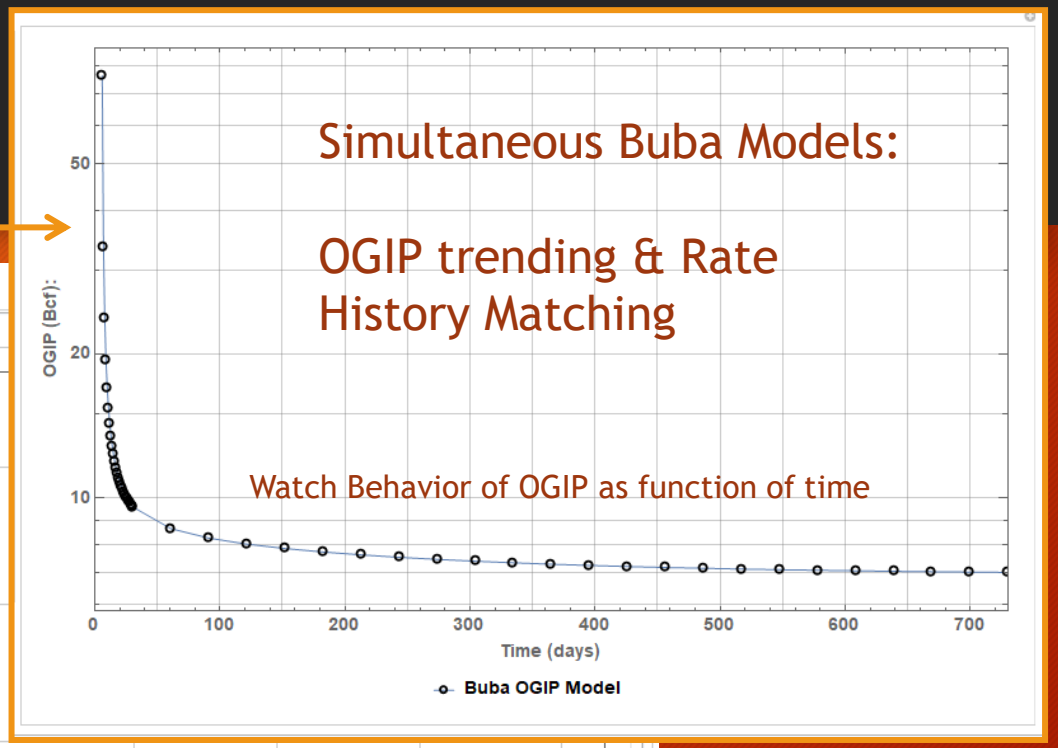
In the future, a transient model will be included in the FMB analysis, for simultaneous OGIP, Permeability, and Skin estimation.



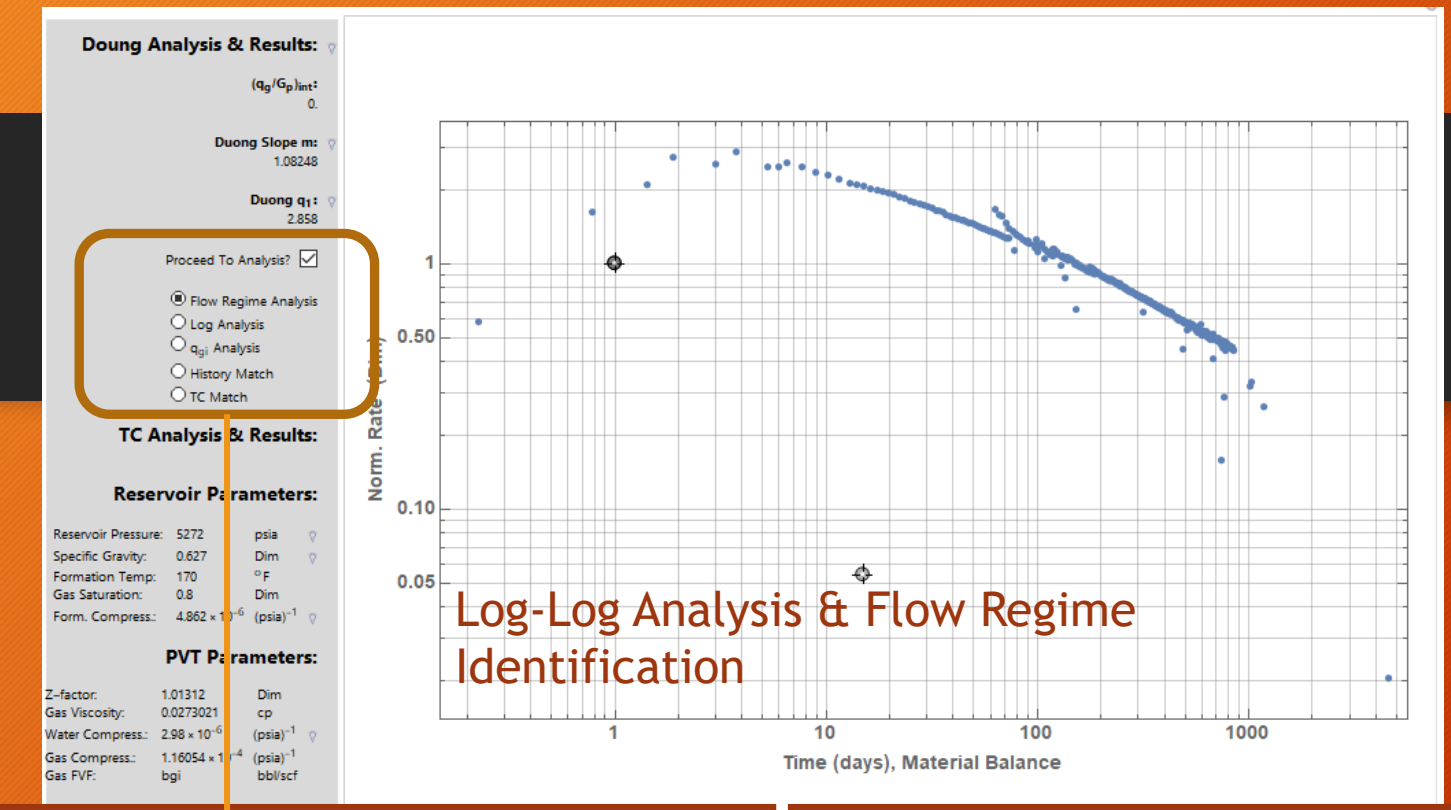
# Other Material Balance Models

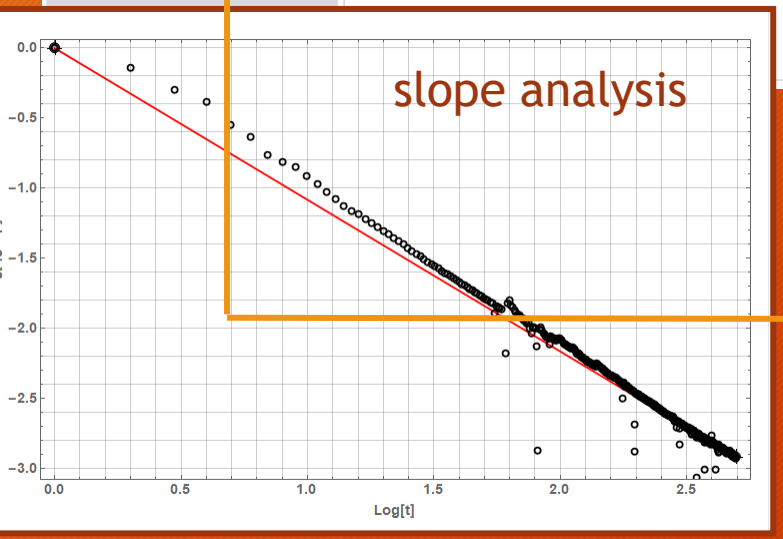


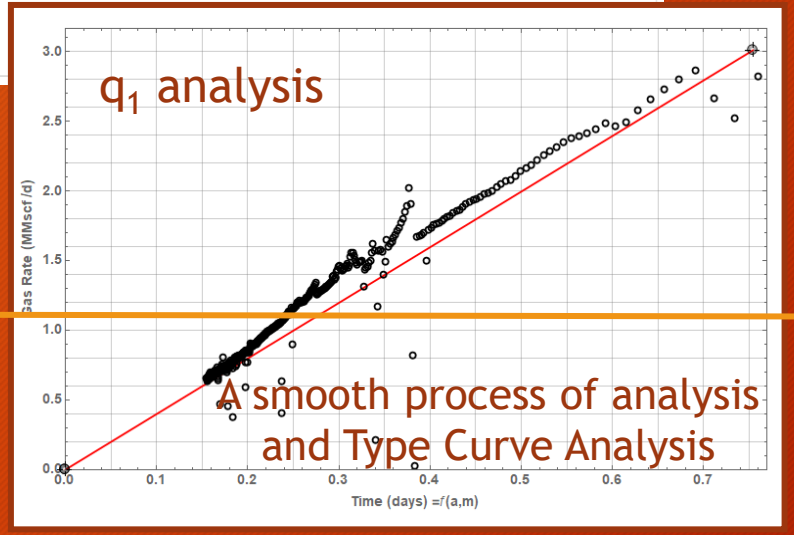



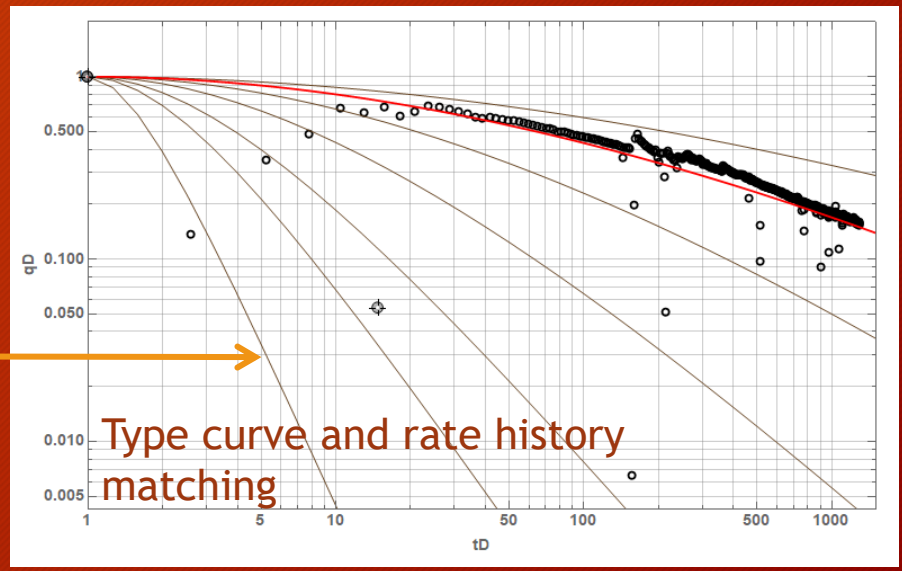
# Shale Models

I expanded my decline work to include more sophisticated models for Shale analysis, such as the Duong Decline Model

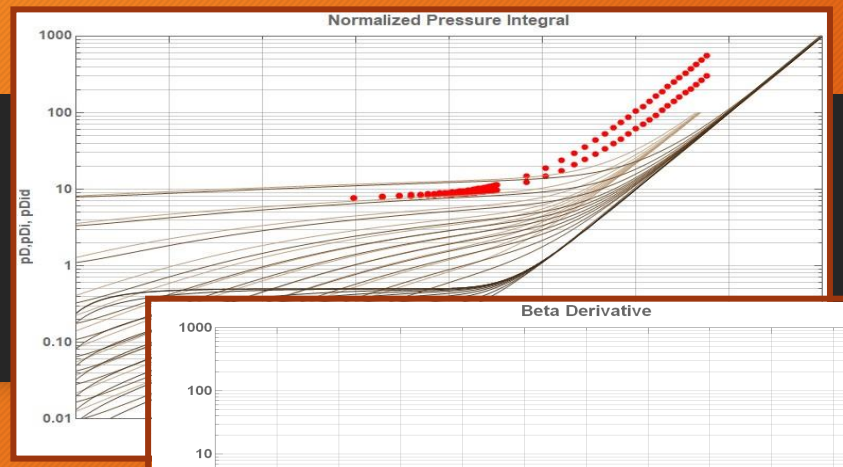


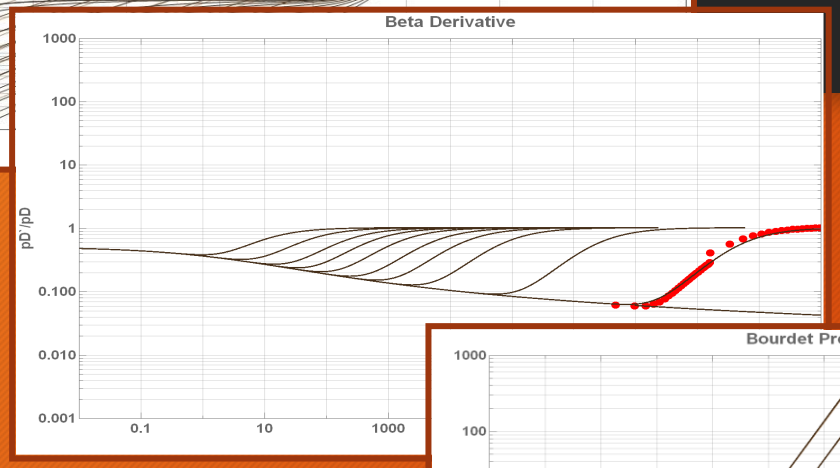



# Building Type Curves

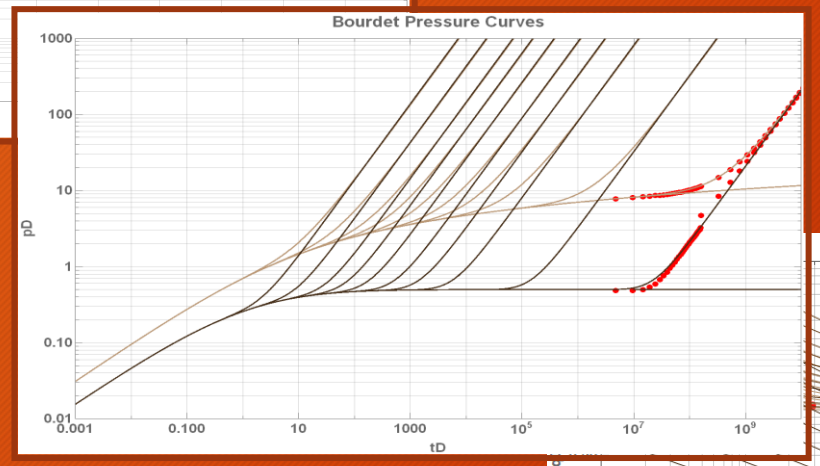



Normalized Pressure Integral

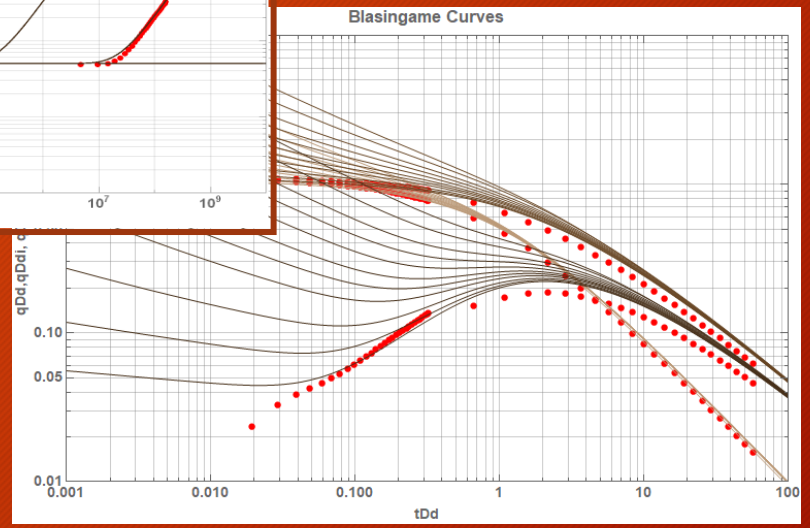



Beta Curves

*Using Tab or Menu features, I was able to create simultaneous analyses using various dashboards.*

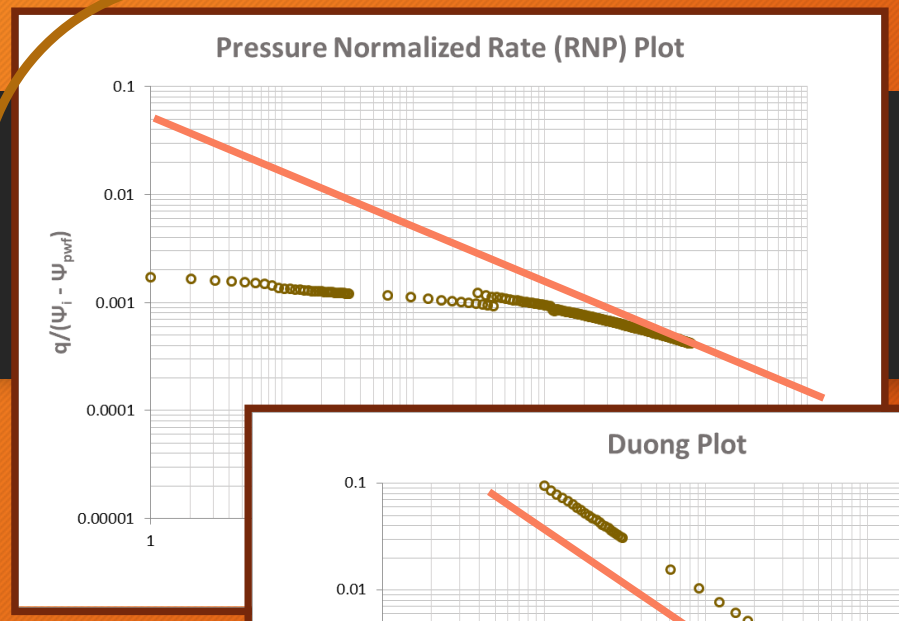


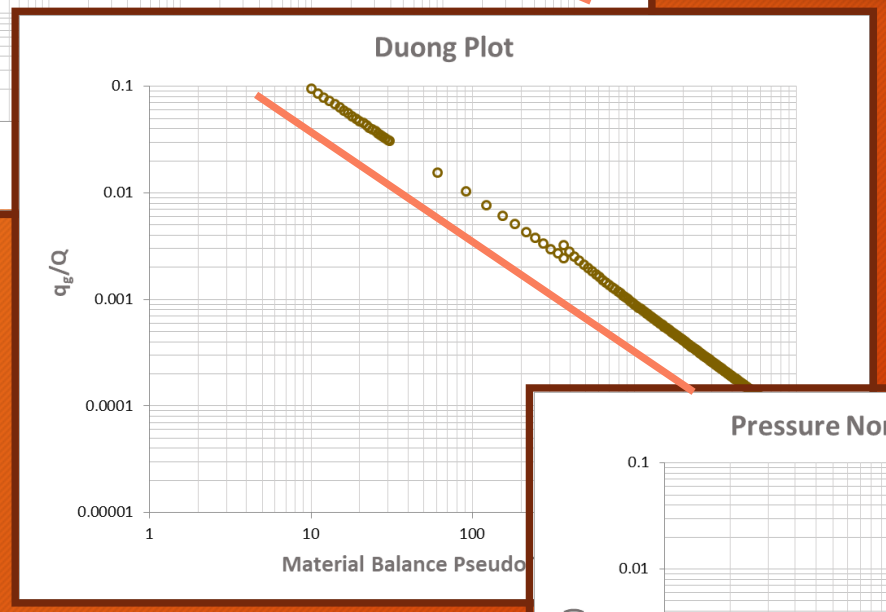

Traditional Bourdet Curves

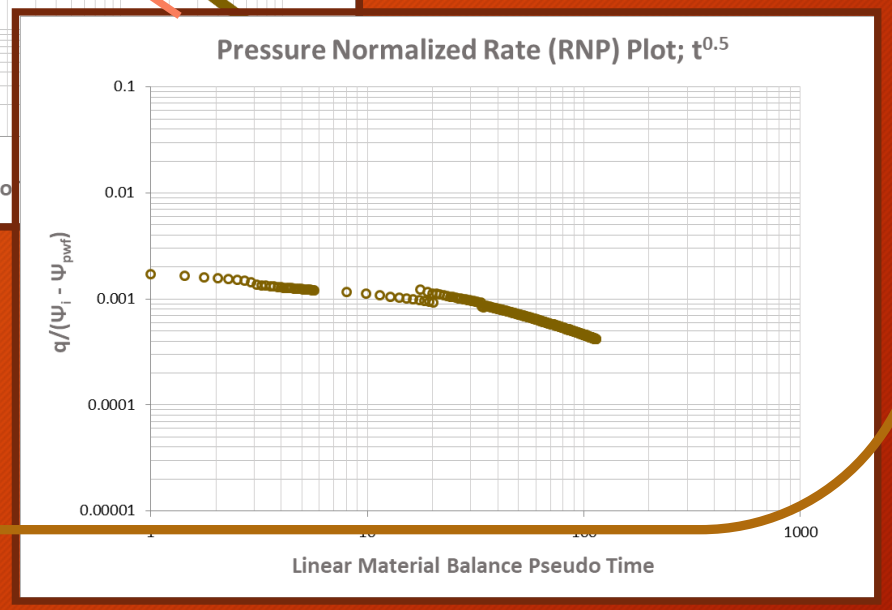



**Blasingame Curves**

# Specialized Analyses



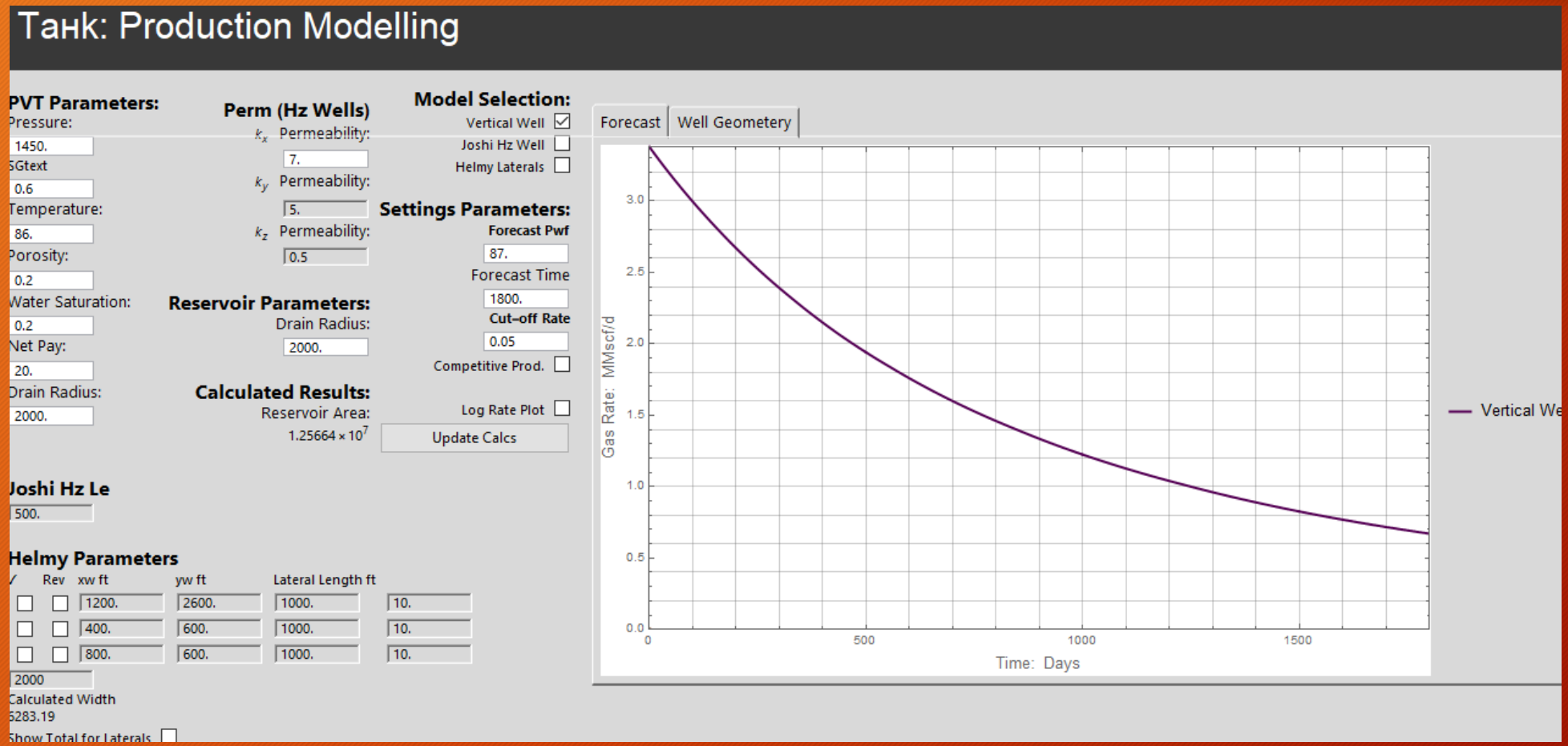

I used Wolfram language for a variety of specialized plots RTA/PTA plots to determine flow regimes for:

- Horizontal wells
- Vertical wells
- MFHZ
- Stimulated wells

Evaluate data for:

- Fracture conductivity
- Fracture half-length
- Reservoir kh
- Formation damage
- Flow Regime

# Full-on Production Modelling!




Like traditional software, there are dedicated libraries for:

- > PVT properties
- > Material Balance
- > HMI / GUI

This allows me to build new tools without having to adapt pre-existing algorithms to new tools or modules that I build for myself.

# Tank: Production Modelling

Using Helmy's approach, I have been able to model various combinations of vertical and horizontal/lateral wells, and more. Perform sensitivities or contribute to FDP's

**PVT Parameters:**
Pressure: 2000.
Gas Gravity: 0.65
Temperature: 120.
Porosity: 0.2
Water Saturation: 0.2
Net Pay: 20.
Drain Radius: 2000.

Anisotropy

**Perm (Hz Wells)**
$k_x$ Permeability: 20.
$k_y$ Permeability: 5.
$k_z$ Permeability: 0.5

**Reservoir Parameters:**
Drain Radius: 2000.

**Calculated Results:**
Reservoir Area: $1.25664 \times 10^7$

**Model Selection:**
Vertical Well ☐
Joshi Hz Well ☐
Helmy Laterals ☑

**Settings Parameters:**
Forecast Pwf: 500.
Forecast Time: 1500.
Cut–off Rate: 0.5
Competitive Prod. ☐
Log Rate Plot ☐
Update Calcs

**Joshi Hz Le**
500.

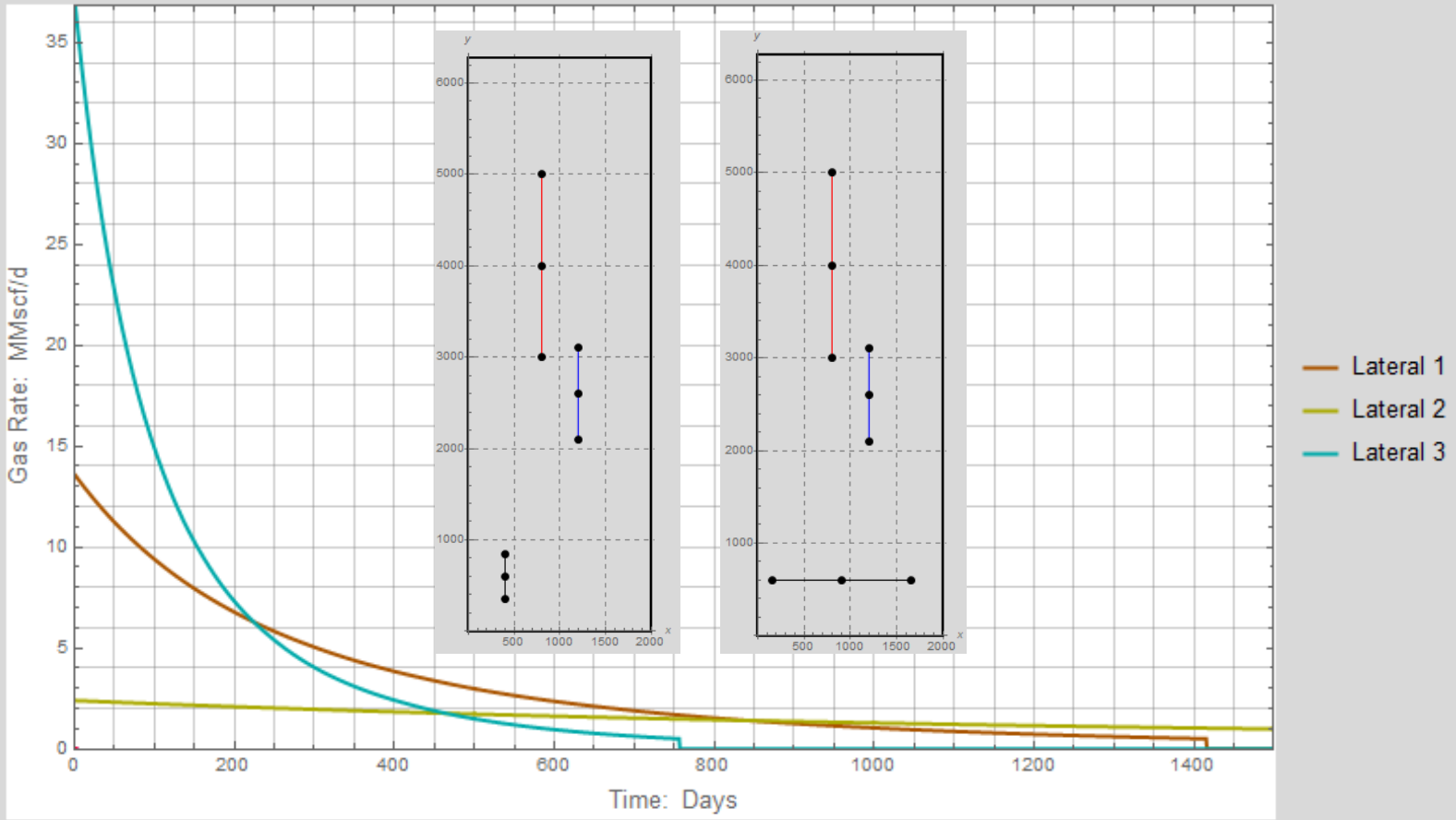
**Helmy Parameters**

Variable combinations of lateral placement

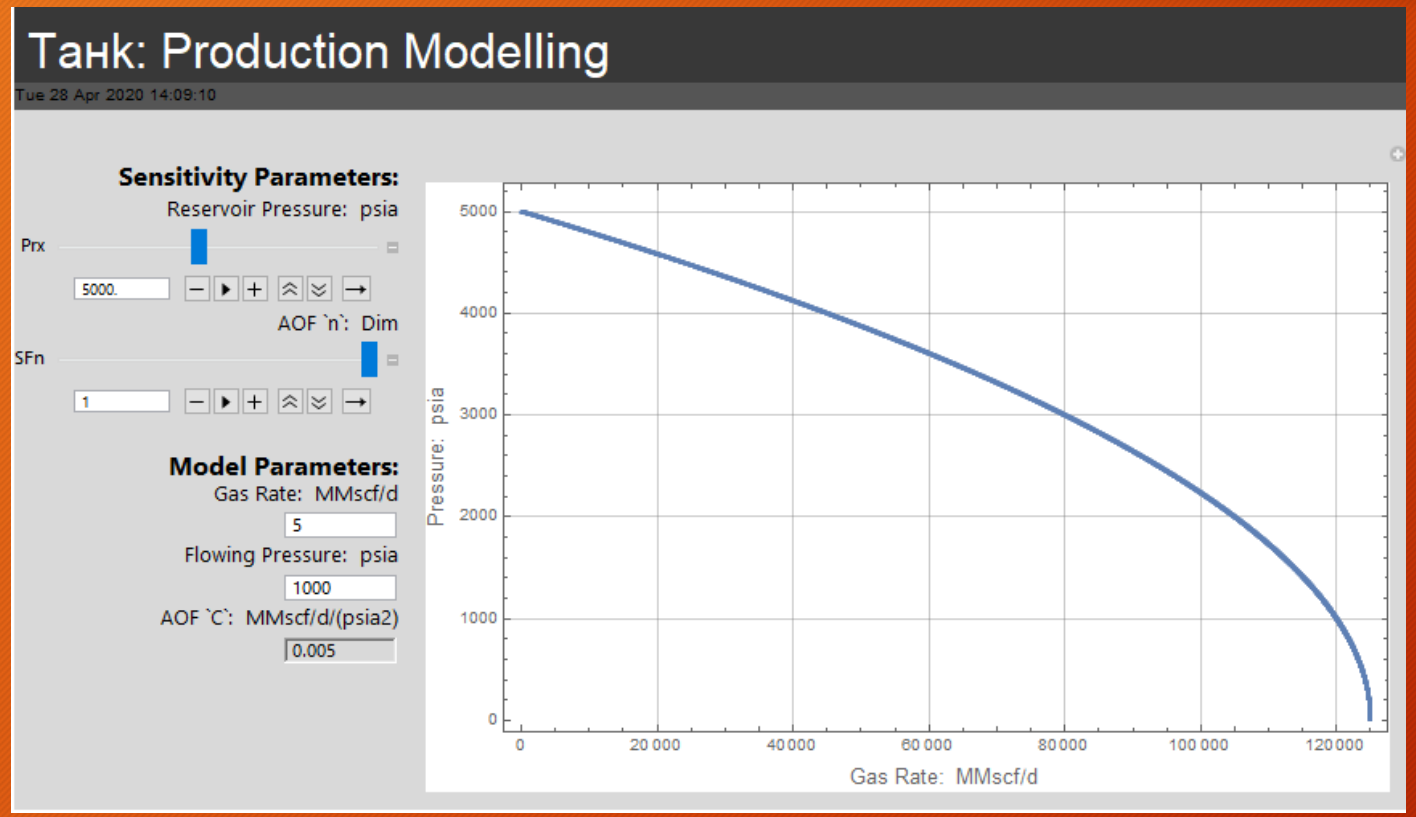
| ✓ | Rev | xw ft | yw ft | Lateral Length ft | |
|---|---|---|---|---|---|
| ☑ | ☐ | 1200. | 2600. | 1000. | 10. |
| ☑ | ☐ | 400. | 600. | 500 | 10. |
| ☑ | ☐ | 800. | 4000. | 2000. | 10. |

2000
Calculated Width

Forecast | Well Geometery



# Of Course, Simple & Interactive IPRs



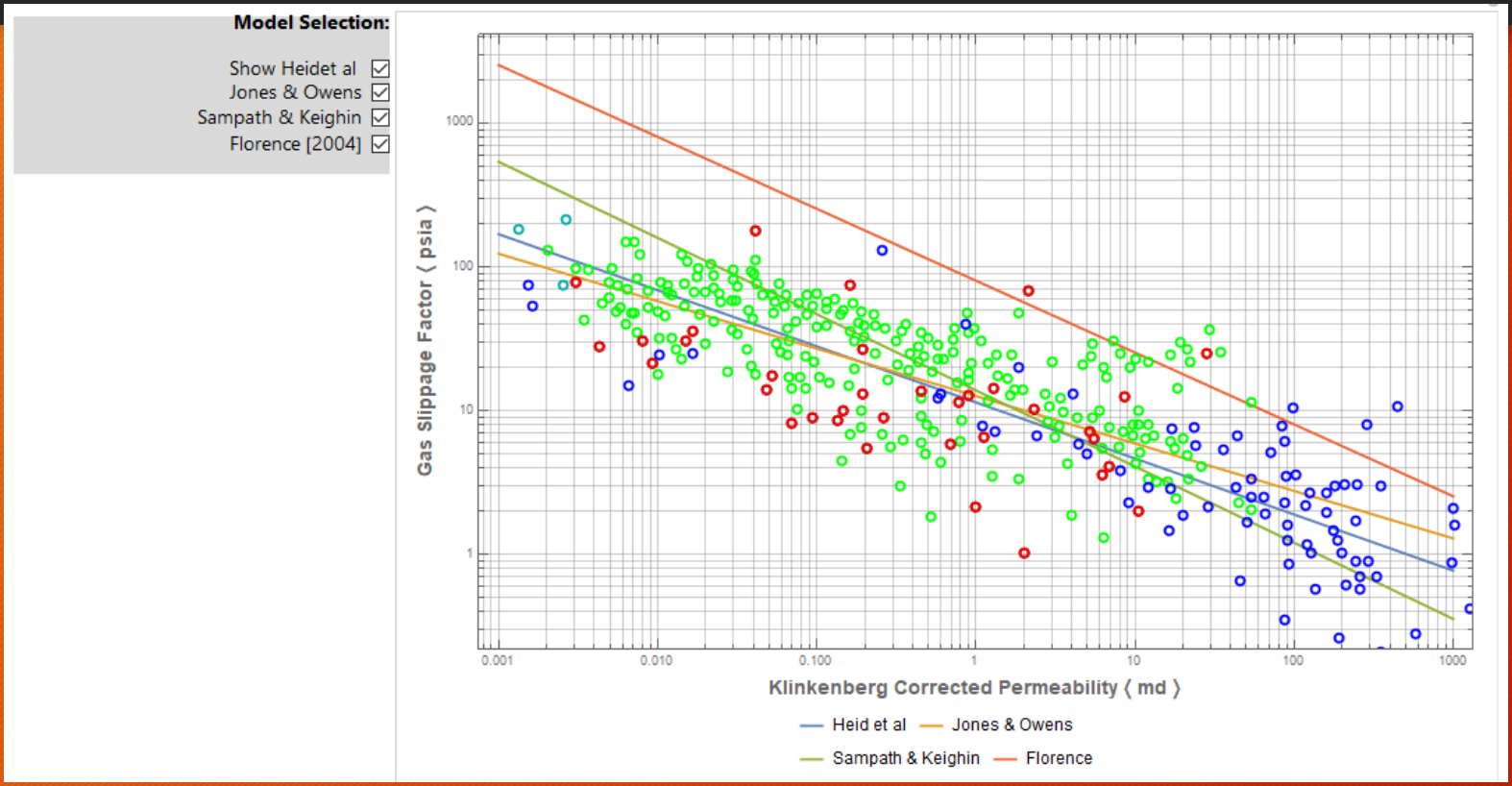

But who cant do that in Excel? Right?

However, Wolfram has ability to deploy any self contained module (or manipulate) as seen right to interactive players or even interactive frames in your personal webpage or intranet!

```
AOFMan = Manipulate[

   FSCn = FieldSize → Small;
   SFTable = Table[{SFqgTest / SFPwfTest * (Prx^2 - Pwfx^2) ^SFn, Pwfx}, {Pwfx, 14, Prx}
             ];

   AOFplot = ListPlot[SFTable,
                ImageSize → 600,
                Frame → FrameForm,
                Background → White,
               FrameLabel → {Grid[{{RateLabel, GasRateunit}}], Grid[{{PresLabel, punits}}]},
                GridLines → Automatic],

   Grid[{
     {SNSparams},
     {Grid[{{ResPresLabel, punits}}]},
     {Control[{{Prx, 5000}, Dynamic[SFPwfTest], 10000, Appearance → {"Open"}}]},
     {Grid[{{AOFLabeln, dimunit}}]},
     {Control[{{SFn, 1}, 0.5, 1, 0.001, Appearance → {"Open"}}]},
     {BlankLine},
     {MODparams},
     {Grid[{{RateLabel, GasRateunit}}]},
     {Dynamic[InputField[ Dynamic[SFqgTest], Number, FieldSize → Tiny]]},
     {Grid[{{PwfPresLabel, punits}}]},
     {Dynamic[InputField[ Dynamic[SFPwfTest], Number, FieldSize → Tiny]]},
     {Grid[{{AOFLabelC, AOFunitP2}}]},
     {Dynamic[InputField[ Dynamic[N[SFqgTest / SFPwfTest]] , Number, Enabled → False, FieldSize → Tiny, Background → BCTV]]}

    },
    Alignment → Right,
    Background → BCTV
   ],
```

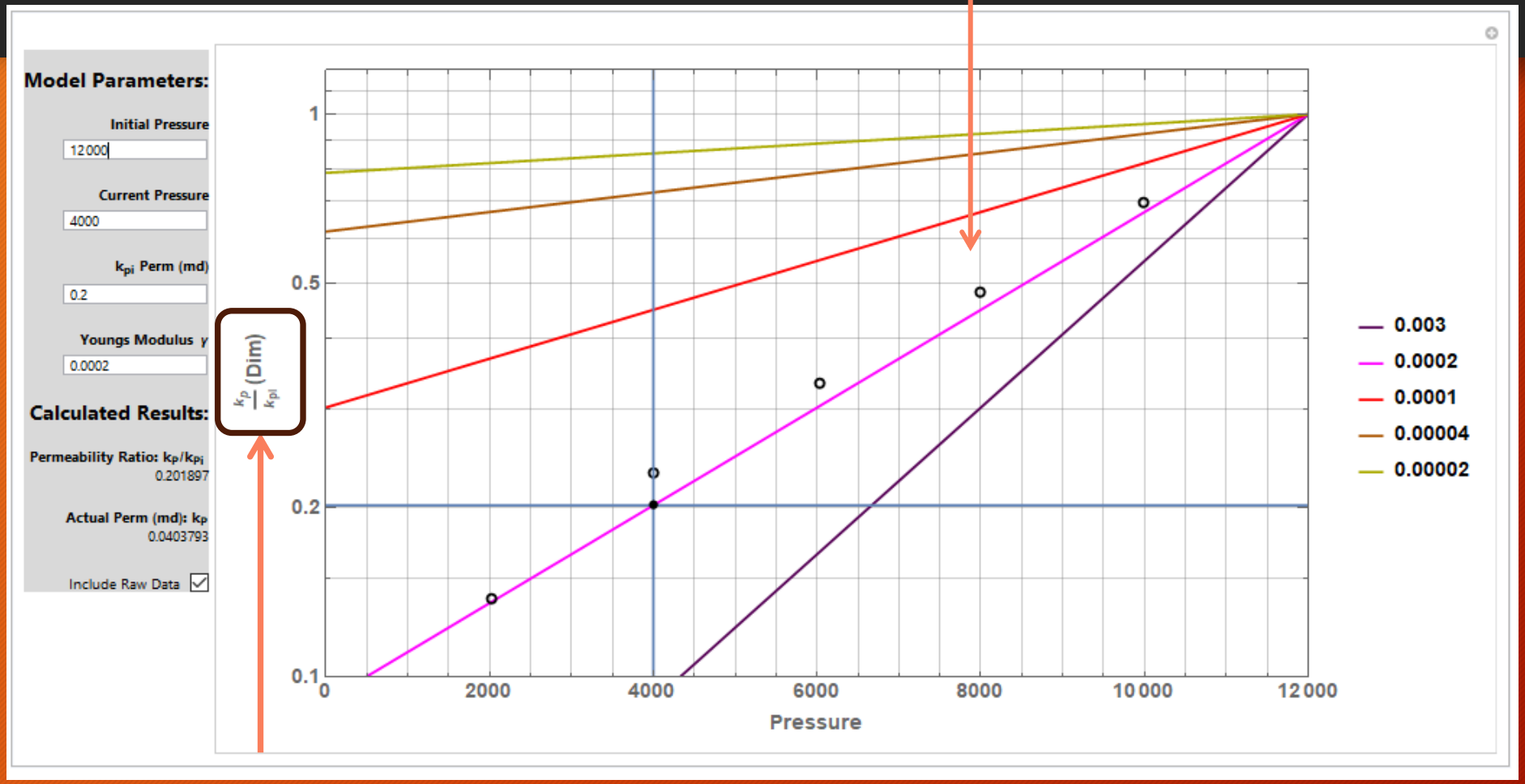
Just a sample of a “Module” (or more correctly, a Manipulate Function) which I can deploy as a stand alone application, an embedded interactive frame in a webpage, or even its own cloud deployed application

# Introduction of Non-Darcy Flow




- As my interests moved away from conventional gas, it expanded into situations where I might require non-linear behavior.
- Using libraries, I am able to create generic permeability / pseudo-pressure call which allow me to use various combinations of non-linear properties in various analyses

# Evaluate/ Match Test Data




If need be, I can history match model to laboratory data to obtain empirical values for Young's modulus, and k(p) – which are subsequently used in production models

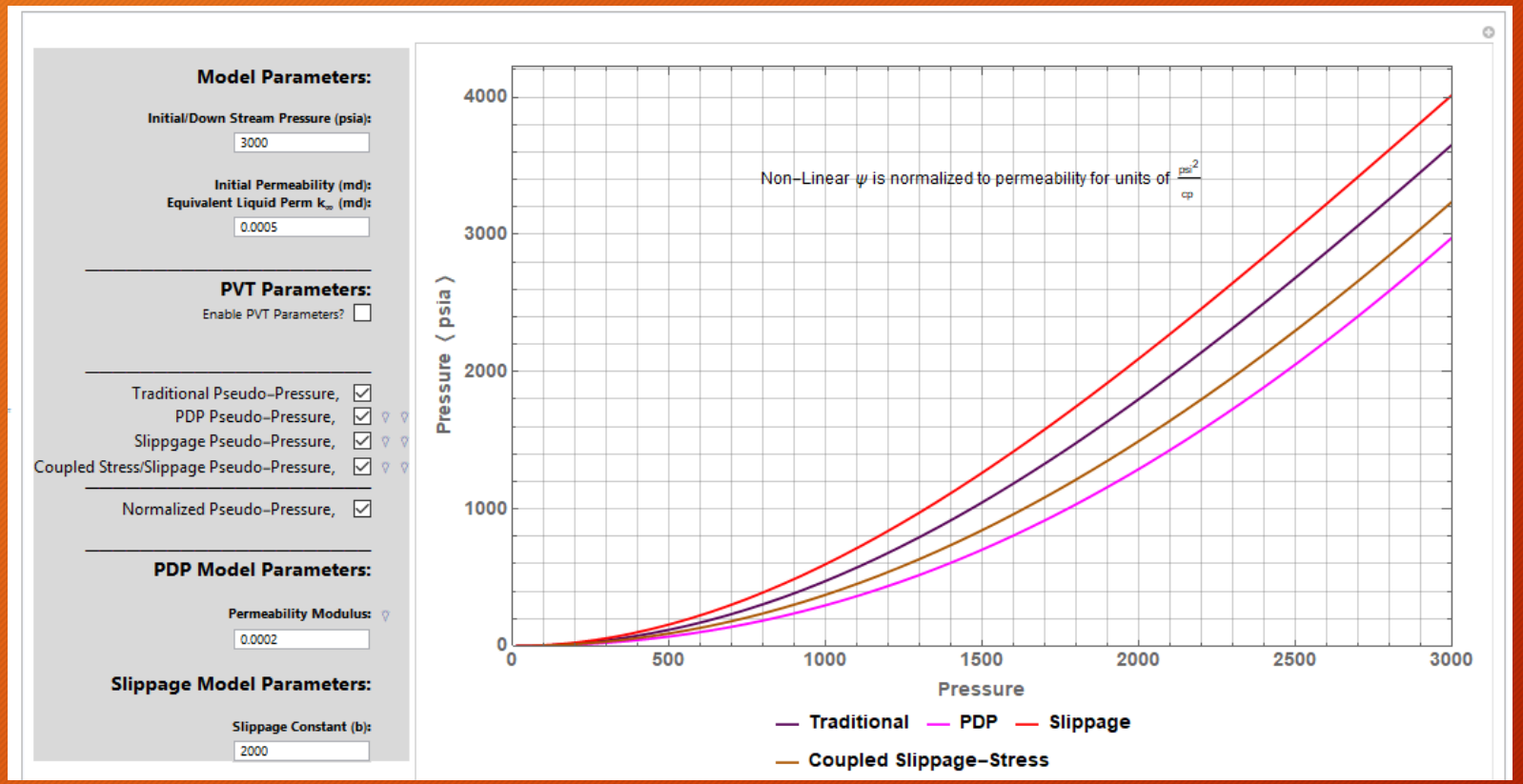
$k_r = k(p)/k_i$: Find permeability as function of drawdown & incorporate in production analysis

# Incorporating Non-Linear Permeability into Models

In previous slides, I demonstrated the application to type-curve analysis and more. Of course, many of these semi-analytical models require both a pseudo-pressure and pseudo-time calculation.

In this section, I demonstrate the use of the Wolfram Language in calculating pseudo-pressure with a number of non-linear effects commonly seen in tight gas and shale gases

# In My Code, All Non-Linear Permeability is Incorporated through Pseudo Pressure




In my production models, gas slippage and PDP is incorporated through pseudo-pressure and pseudo-time.

$$t_a = \frac{2}{k_i}\int_0^t \frac{\bar{k}}{(u_t c_g)}\,dt$$

$$\psi = \frac{2}{k_{eq}}\int_{kP_{ref}}^{P} \frac{p\,\bar{k}}{(u_t z_g)}\,dt$$

# Creating a Generic Pseudo Pressure Library

$$\psi = \frac{2}{k_{ref}} \int_{kP_{ref}}^{P} \frac{p\,\bar{k}}{(u_t z_g)} dt$$

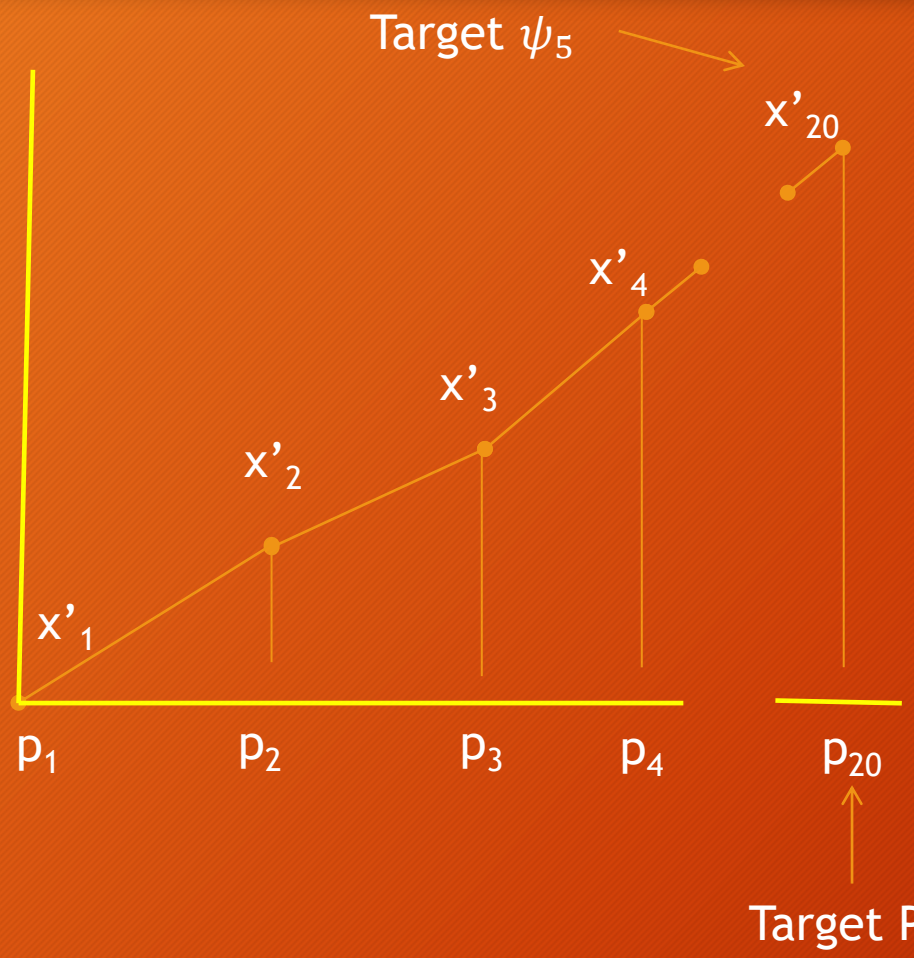
Any relationship for permeability can be used including:

- Stress dependent perm
- Slippage considerations
- Coupled models
- Any other potential models

$k_{ref}$ is used to normalize pseudo-pressure to ensure it can be used easily in existing well / deliverability models

$$q_g = \frac{C\ kh(\psi_i - \psi_{wf})}{T_f \left[ Ln\left(\frac{r_w}{r_e}\right) - \frac{3}{4} + s \right]}$$

# Calculating Generic Pseudo  Pseudo Value




$$\psi_1 = \frac{p_2 + p_1}{2}(x_2) + 0$$

$$\psi_2 = \frac{p_3 + p_2}{2}(x_2) + \psi_1$$

$$\psi_3 = \frac{p_4 + p_3}{2}(x_3) + \psi_2$$

$$\psi_{20} = \frac{p_4 + p_3}{2}(x_{20}) + \psi_{20-1}$$

$$x_n = \frac{p_n k(p_n)}{u(p_n) z(p_n)} \frac{1}{k_{ref}}$$

I tend to use a minimum of 20 steps to calculate a single pseudo pressure point.

There are of-course optimization options

# Example Pseudo Pressure Code

- Example Stress Dependent Permeability Code

```
mpStress[Testpsia_, temp_, SG_, n2_, co2_, h2s_, gamma_, ki_, Pinitial_]:=Module[
    {p=Testpsia, mpSolutionStress = 0, Pold = 0, Xold = 0, Xnew = 0, Step =20, Pstep =0,Pnew =0, Pinit=Pinitial, gam=gamma, kinit=ki, kp=0},
    Pstep = p / Step;
    Do[
        Pnew = Pold + Pstep;
        kp=kperm[gam,ki,Pinit,Pnew];
        Xnew = 2 * Pnew*kp / ZGas[Pnew, temp, SG, n2, co2, h2s] / Ug[Pnew, temp, SG, n2, co2, h2s];
        mpSolutionStress = mpSolutionStress + (Xold + Xnew) / 2 * Pstep;
        Pold = Pnew;
        Xold = Xnew;
        ,
        {20}];
    mpSolutionStress  (*Returns units of md-psi2/cp*)
```

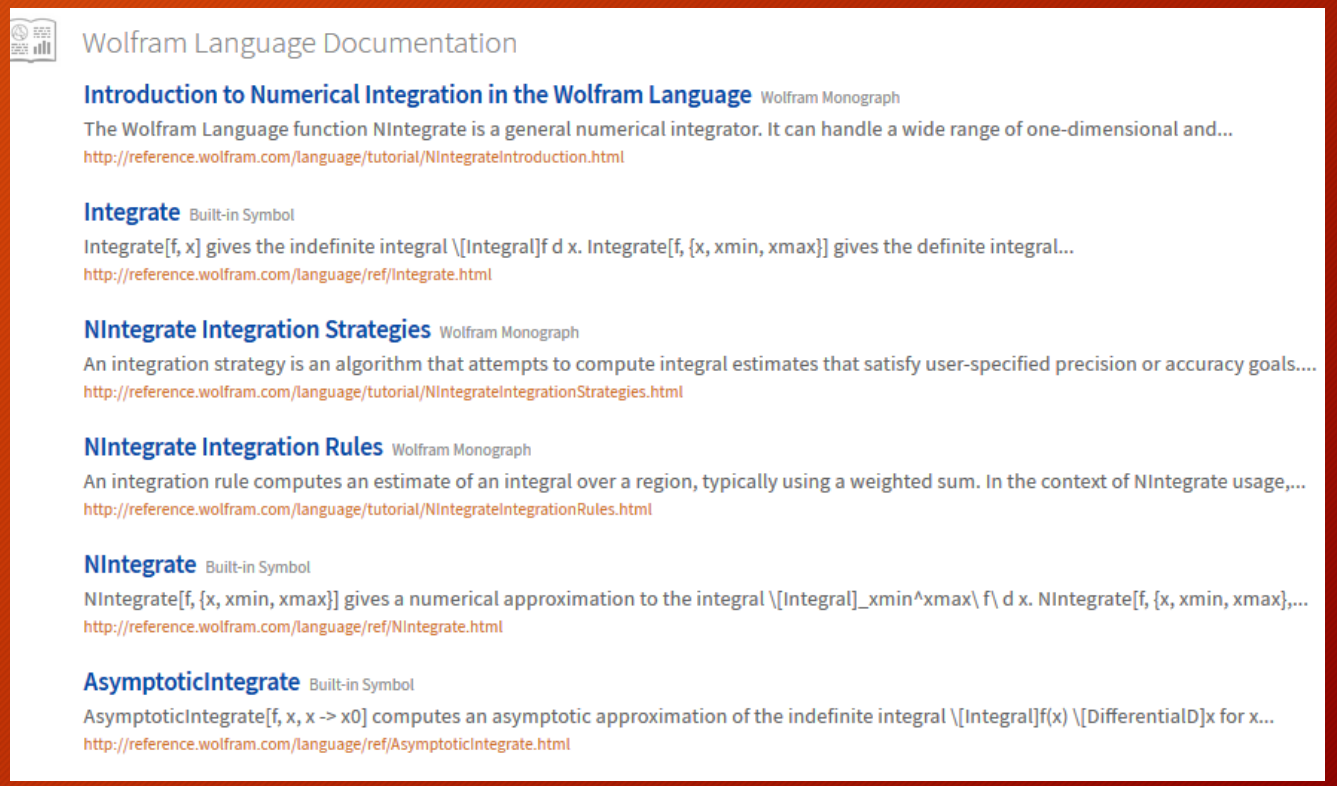
Call any function I have for stress dependent perm.

Can be any model for k, u, or z

Perform twenty (20) steps to any pseudo-pressure calculations. Code can be optimized

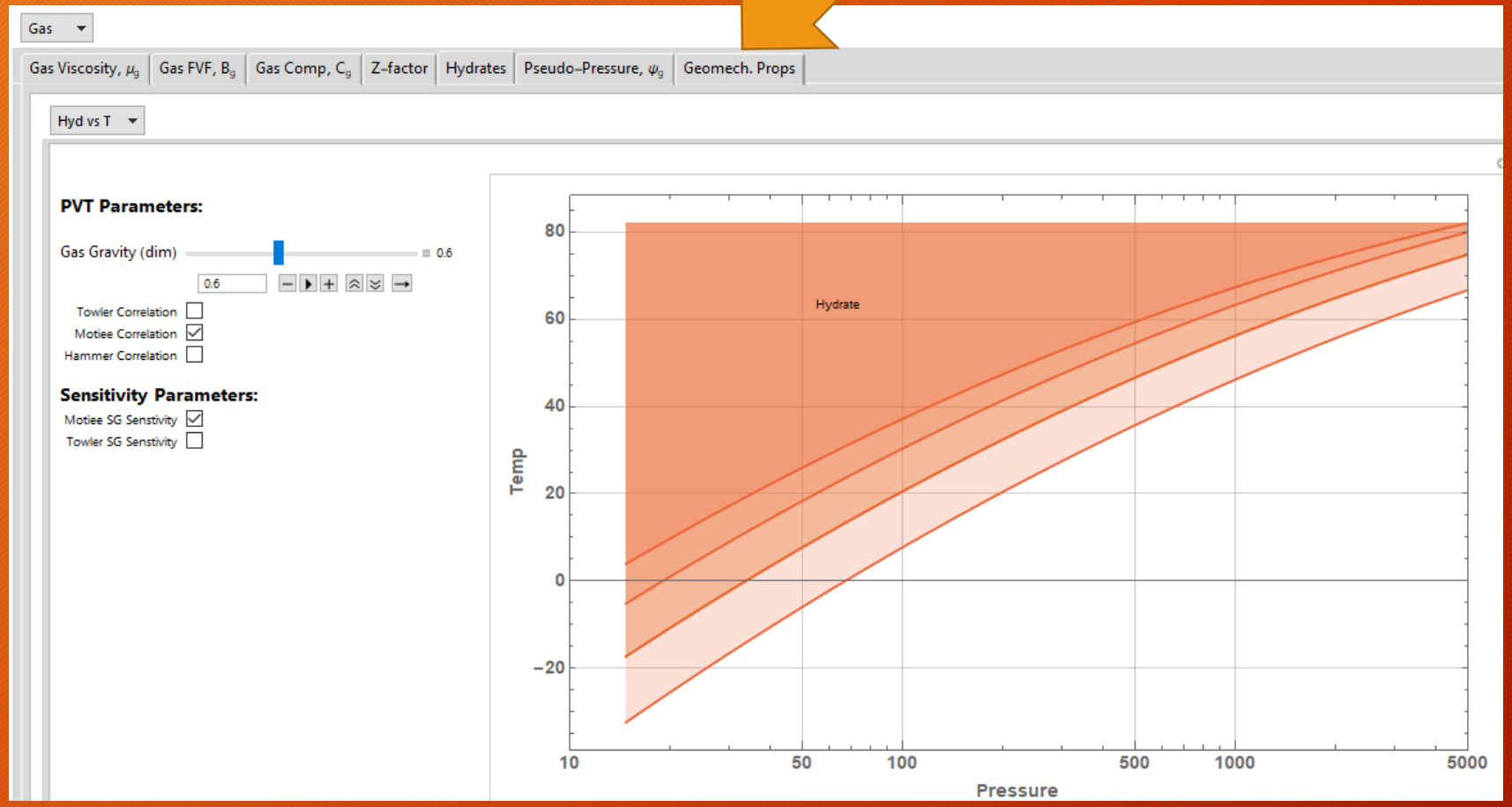
# Options to Manually Coding Integration

- Wolfram contains a number of libraries for use in integration and derivative applications including:



# Hydrate Modelling / Paraffin Management

Each tab represents a “Manipulate” contained within a “TabView”



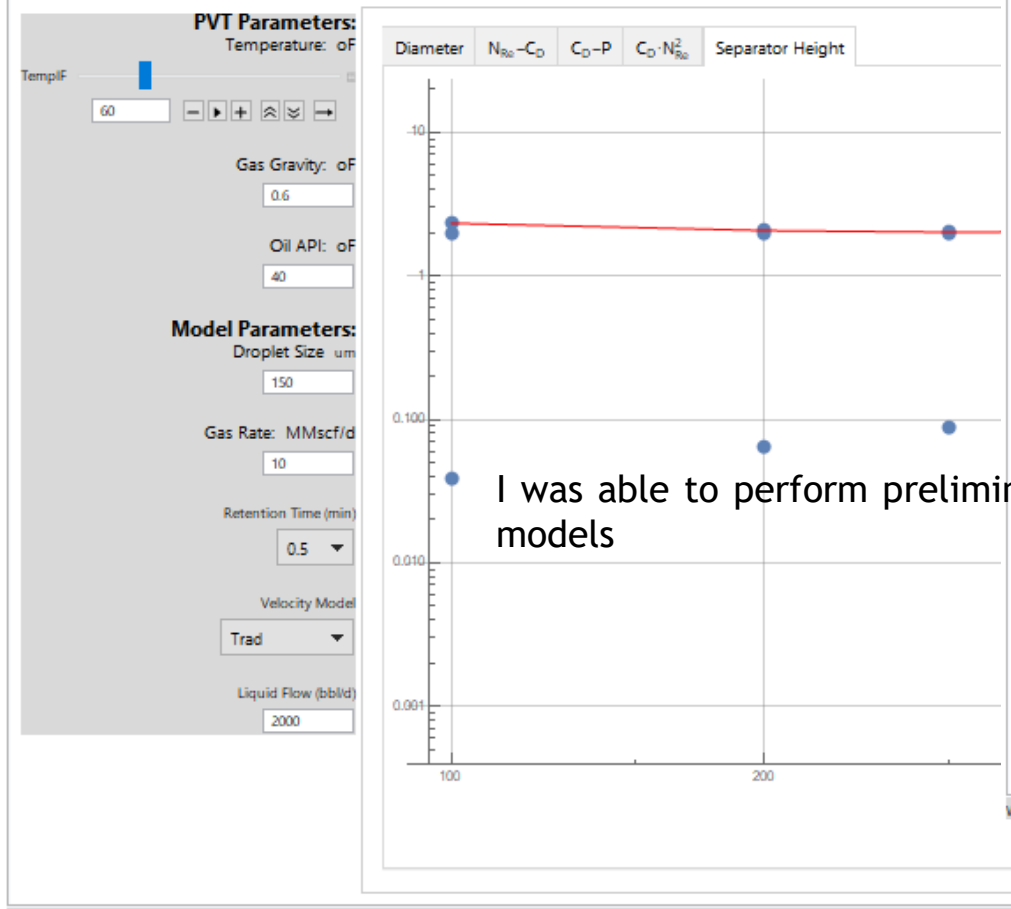

In this example, I used Wolfram to predict/compare the behavior of different hydrate correlations (and possibly import P & T data from sensors or SCADA historians)

# Operations / Process Example

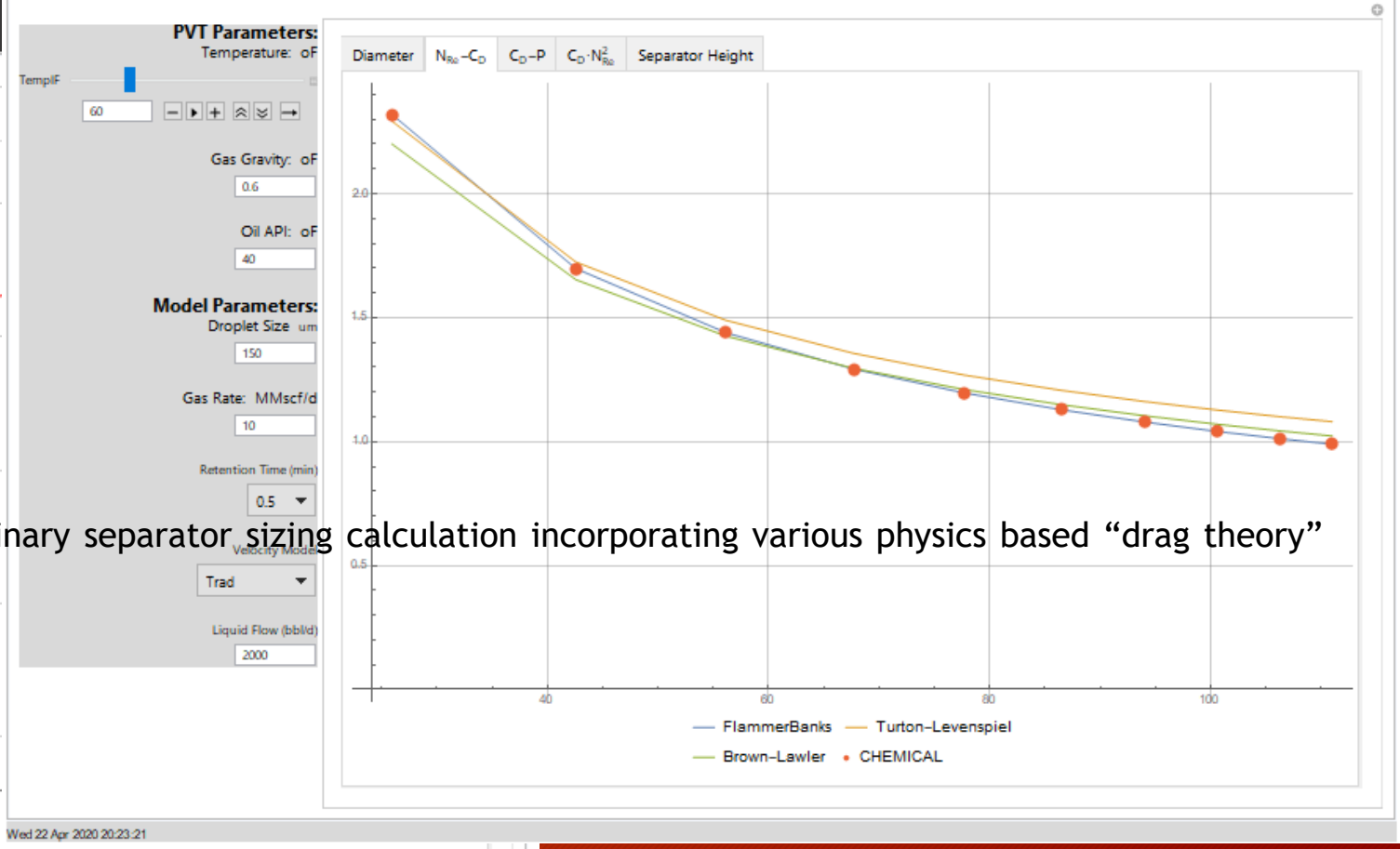
During my contract at NAIT, I had time to work with some of the Chemical Engineering Technology Students leading to the expansion of my simulation to include separator design

# Process Applications

## Kemikəl: Separator Modelling




## Kemikəl: Separator Modelling




I was able to perform preliminary separator sizing calculation incorporating various physics based “drag theory” models

# Big Huge Monte Carlo Simulation Example

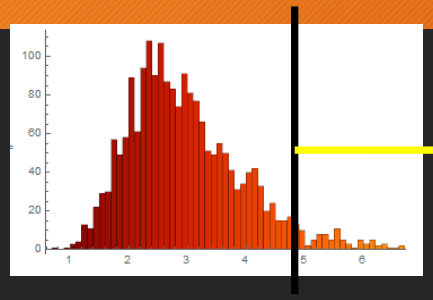
In the next examples, I show the basic logic of how I created a Wolfram based Monte Carlo Simulation with applications to:

OGIP Estimation

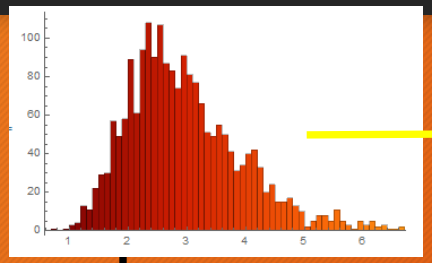
Production Forecasting

And Recoverable Gas!

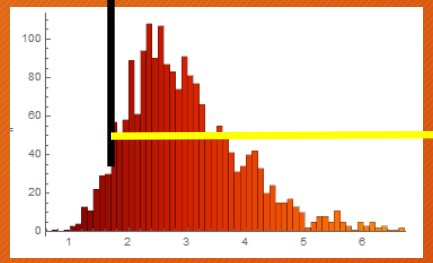
# Monte Carlo Simulation Logic



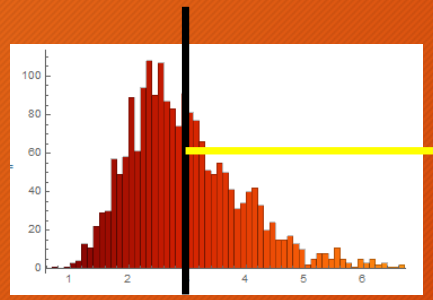
Porosity



Permeability
(dependence on Φ )



Net Pay



Saturations/
Others

Volumetric Model

Well Model/
Deliverability Model

Single Forecast

Horizontal/Vertical/
Other Well Model

Permeability Models

PVT Models

Material Balance

The procedure is repeated "n" times to create various distributions

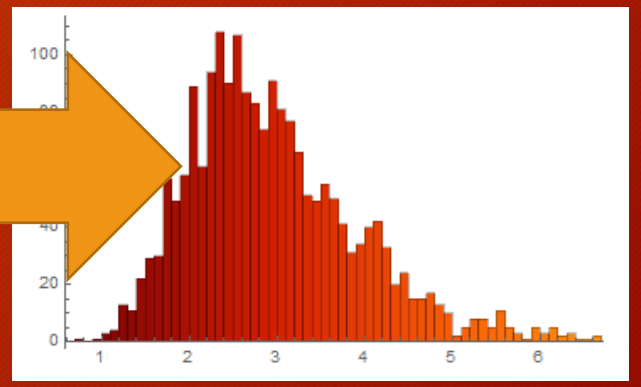
# Generating Distributions for Initial Conditions

Number of realizations in generating distributions. Each value is stored in "n" index of an array

```
(*Loop to Generate Plots*)
For[n=1., n≤testn, n++,

Por[[n]]=RandomReal[
    TruncatedDistribution[{0.,Infinity},
    NormalDistribution[MeanPor,SDPor]],1.]/100;

Perm[[n]]=RandomReal[
    TruncatedDistribution[{0.,Infinity},
    NormalDistribution[MeanPerm,SDPerm]],1.];
```

Bound distribution from 0 to infinite values

RandomReal[ ] generates a single random value according to μ and σ



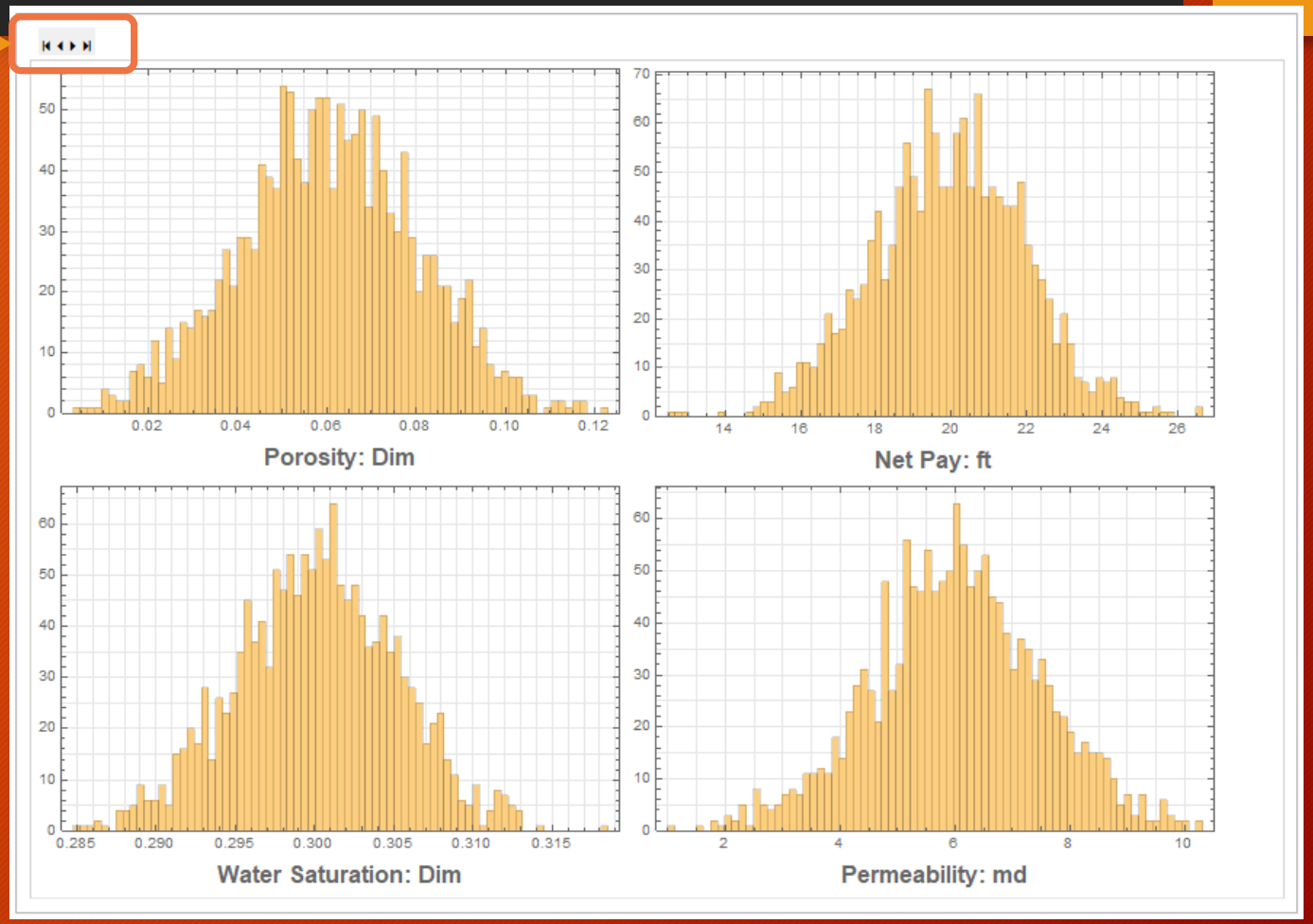

Distributions can be normal, lognormal, and other combinations

Distributions are generated for net pay, porosity, saturations, FVF, and so on

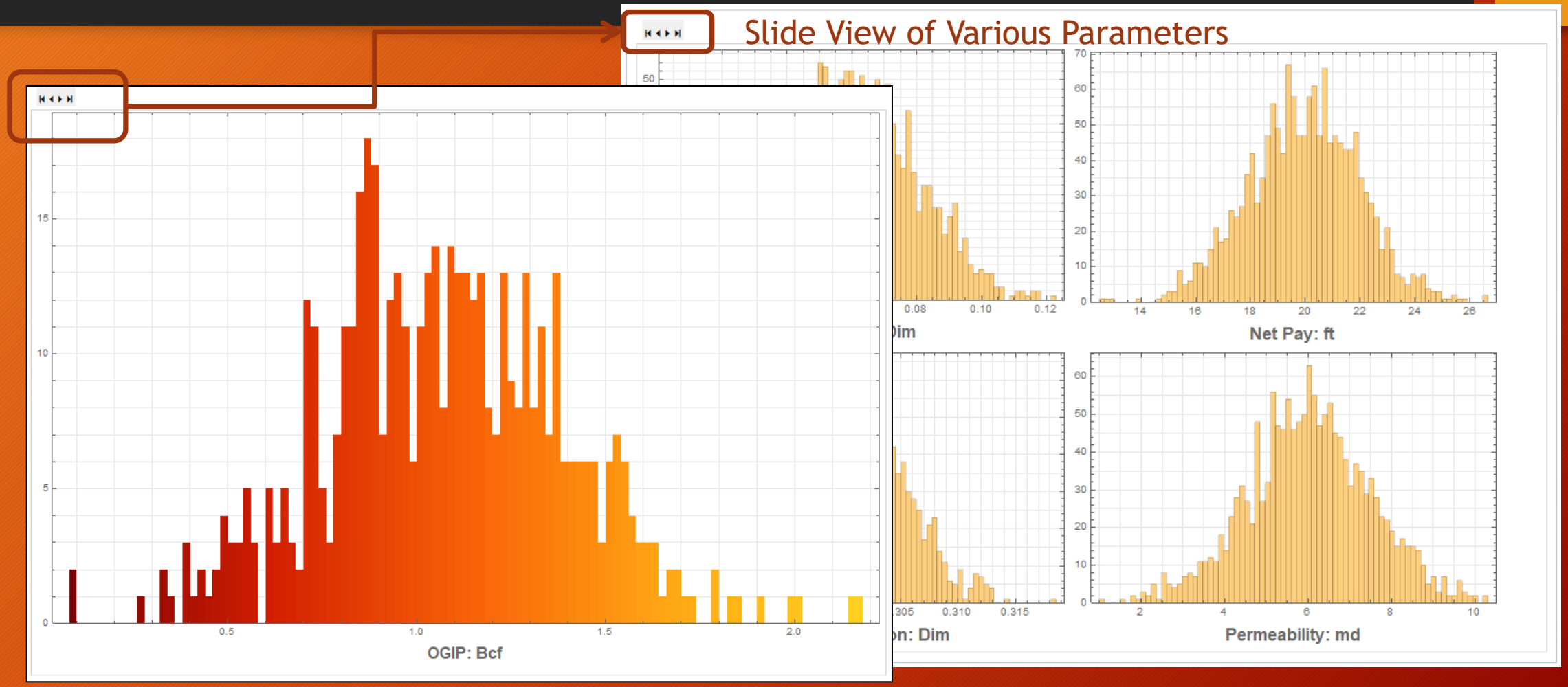
# Typical Parameter Distribution Plots

Scroll through a variety of distribution plots of the various petrophysical parameters

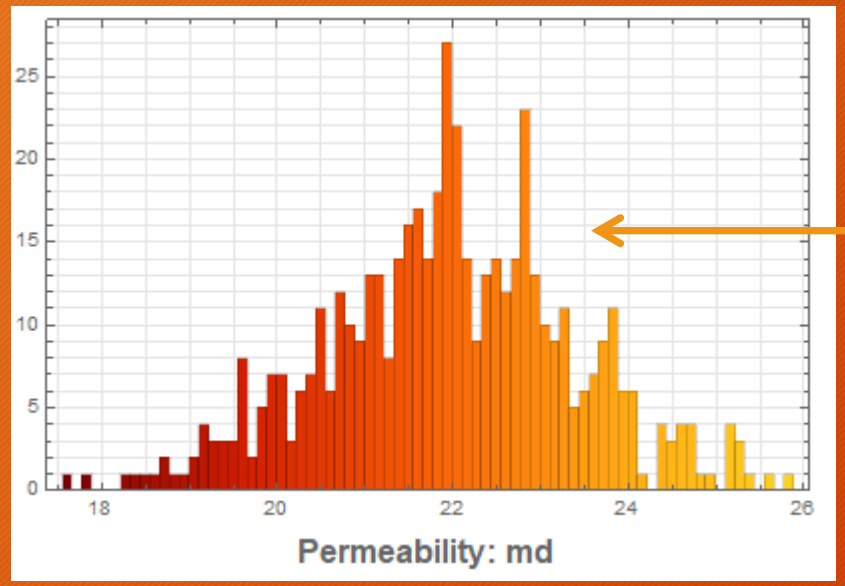
Data can be modelled as normal, lognormal, and more

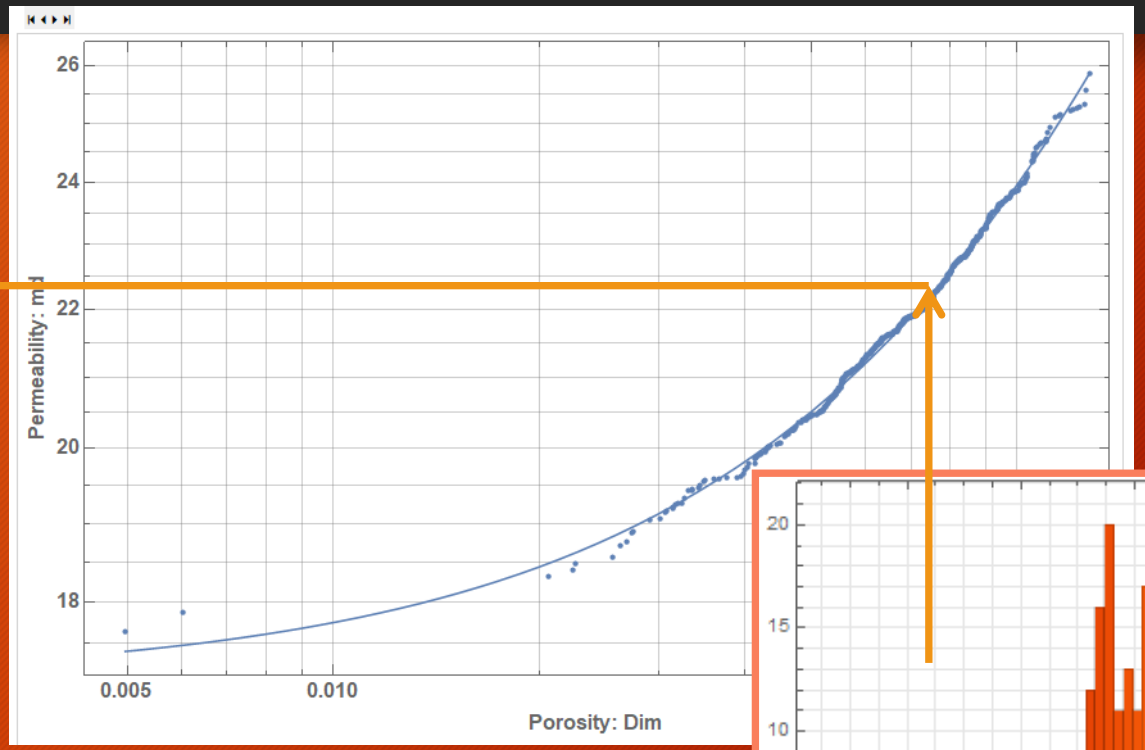


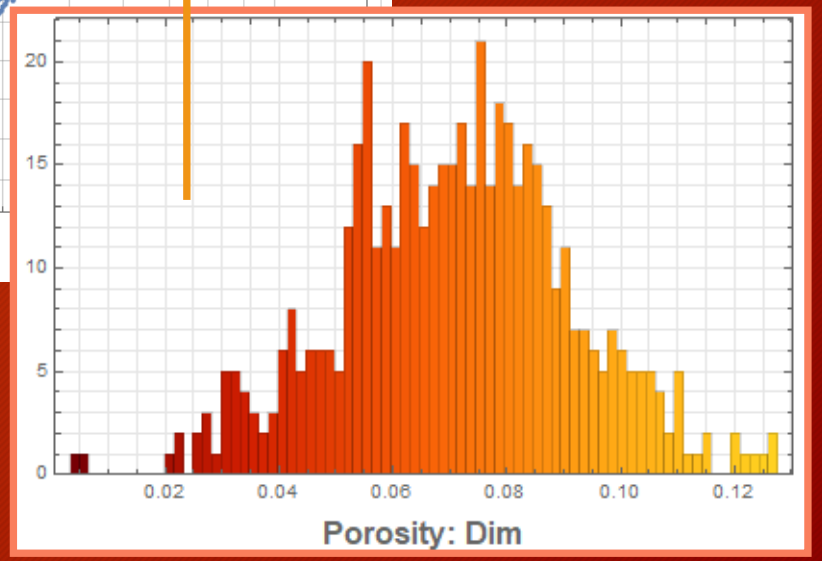
# Typical Parameter Distribution Plots



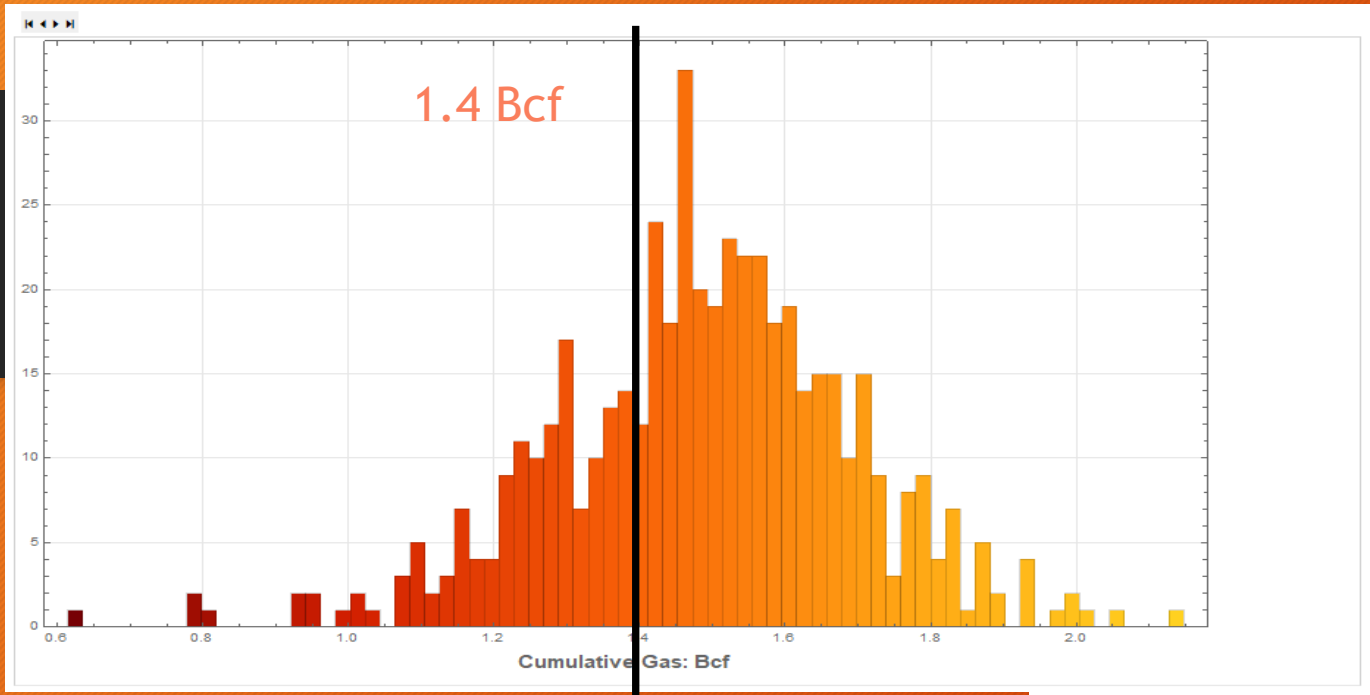
# Incorporate Dependencies: Poro-Perm Distributions



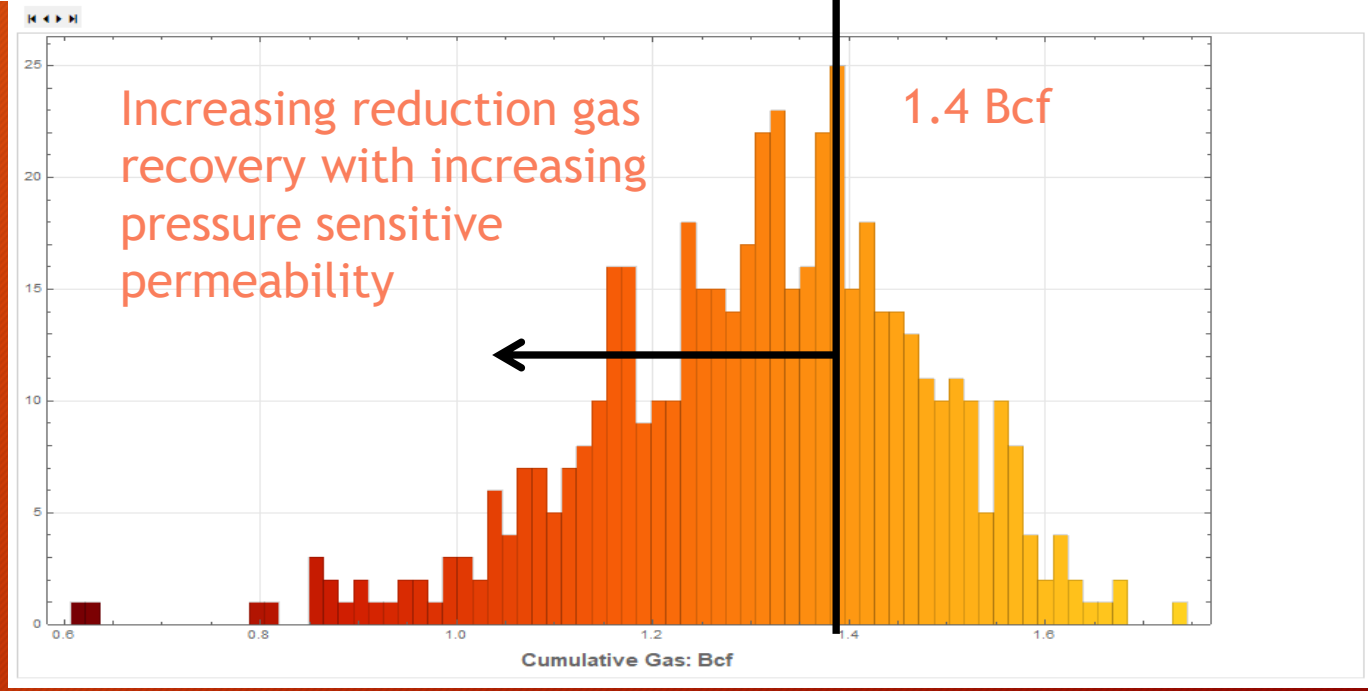




I was able to build basic property dependencies (well at least to what I understand them to be!)

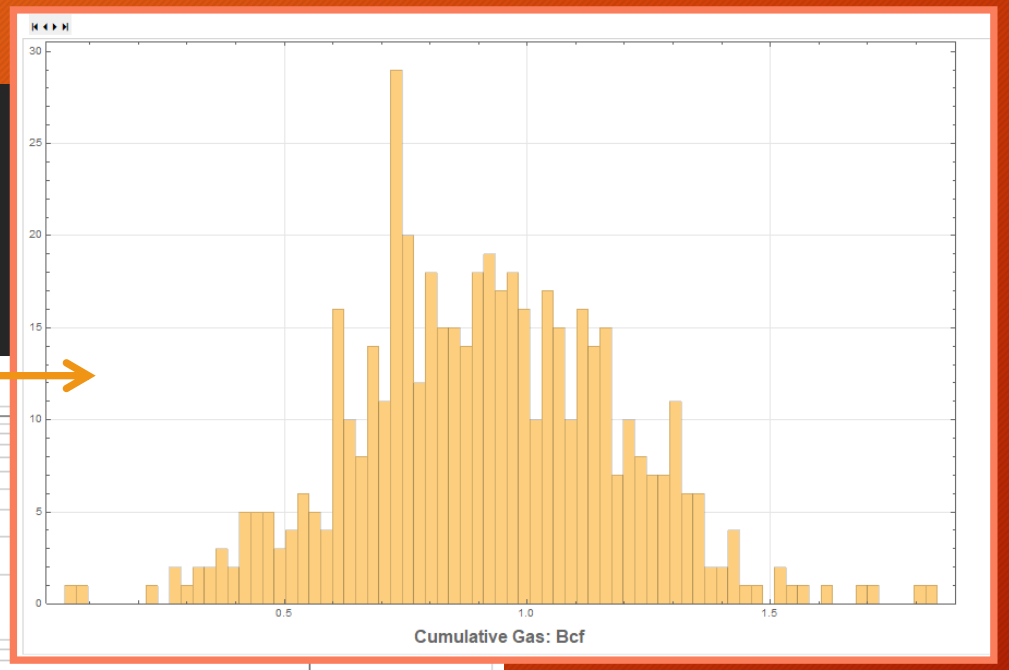


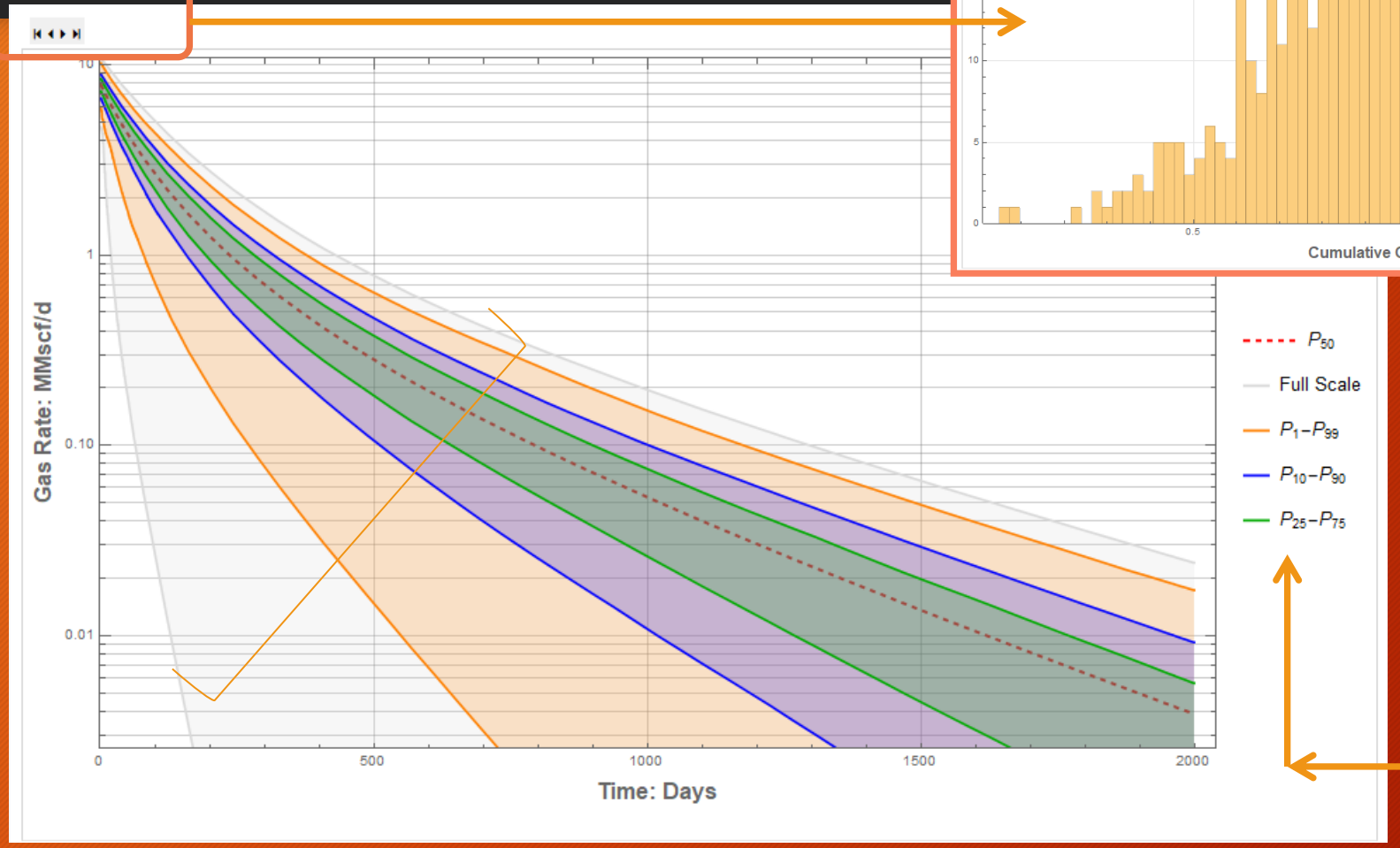

In this example, I used my Wolfram Coe to evaluate the impact of stress sensitive permeability.

In this example, increasing permeability modulus results in a shift of cumulative gas towards a lower recovery.



# Uncertainty & Forecast




Rate forecasts, including cumulative production distributions, can be generated for any percentile desired by the user.

In this example, we have:

$P_1$-$P_{99}$

$P_{10}$-$P_{90}$

$P_{25}$-$P_{75}$

$P_{0.01}$-$P_{99.99}$