**INSTRUCTION MANUAL: EGS Analytical Model Sensitivity Studies**

**By: Benjamin Willis**

1. **Run Sensitivity Studies**

To run the sensitivity studies download the respective study Jupyter Notebook from the GitHub repository:

<https://github.com/BenjaminWillisINL/AnalyticalDataCSVs/tree/main/RunSensitivityStudies>

**1a. Run Varying Exit Perforations Study**

Make sure that the “FrictionFactors.ipynb” Jupyter Notebook is in the same folder as the “With or Without Exit Perfs.ipynb” Jupyter Notebook.

Make Sure all storage variables are initialized to the correct dimensions as described in the screenshot below:

Graphical user interface, text, application, email

Description automatically generated

Set the “WellsOrientation” variable to 1 to begin the parallel well design simulation in the screenshot below:

A picture containing logo

Description automatically generated

Change the “ActivateProductionPerforations” variable to either 1 or 0 depending on if there are exit perforations or not as in the screenshot below:

Text

Description automatically generated

Run all code cells until Ln 49 is reached, once this line is reached run the code and wait, as it will take some time to store all the data. Use the screenshot below to identify this cell:

Text

Description automatically generated

Once this code is done running, go and change the “WellsOrientation” variable to 3 and re-run the whole code, EXCEPT for the initialization of the storage variables. Do not reinitialize the storage variables. Wait for Ln 49 to finish running for the non-parallel well design. Once the code is done running, run the next line to save the data as CSV files as in the screenshot below:

Graphical user interface, application

Description automatically generated with medium confidence

All headers are included in the CSV files so the content of each column is known. The row numbers represent the number of perforations. With row 1 representing 1 perforation, and row 100 representing 100 perforations.

Now re-run the code again with exit perforations on or off depending on what was started with to get the other set of data.

**1b. Running the Permeability Study**

Make sure that the “FrictionFactors.ipynb” Jupyter Notebook is in the same folder as the “VaryPermeability.ipynb” Jupyter Notebook.

Make sure that all storage variables are initialized to the desired dimensions as shown in the screenshot below:

Graphical user interface, text

Description automatically generated

Set the “WellsOrientation” variable to 1 to begin the parallel well design simulation in the screenshot below:

**Text

Description automatically generated with medium confidence**

Now set the permeability value to the desired value (1e-12 m2 or 1e-14 m2) as shown in the screenshot below:

Text

Description automatically generated

Set control variables to desired parameters, in this scenario the exit perforations were turned off. Use the screenshot below to discern which control variables to use:

Graphical user interface, text, application, email

Description automatically generated

Run all the code until the main flowrate cell is reached, then wait for this code to run, use the screenshot to determine which cell to run and wait:

Graphical user interface, text, application

Description automatically generated

Once the cell above is done running, reset the “WellsOrientation” variable to 3 and re-run the whole process again. Do not reinitialize the storage variables again, only run every cell block below the initialization for the storage variables. Once the flowrate cell is done running for the second time, run the next cell to convert all the stored data to a CSV file, as shown in the screenshot below:

A picture containing table

Description automatically generated

All headers are included in the CSV files so the content of each column is known. The row numbers represent the number of perforations. With row 1 representing 1 perforation, and row 60 representing 60 perforations.

**1c. Running the Well Angle Study**

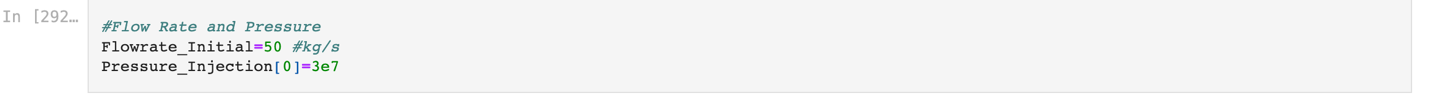
Make sure that the “FrictionFactors.ipynb” Jupyter Notebook is in the same folder as the “Pheta.ipynb” Jupyter Notebook. Also make sure that the “testpheta.CSV” file is in the same folder as the “Pheta.ipynb” Jupyter Notebook.

Make sure that all storage variables are initialized to the desired dimensions as shown in the screenshot below:

Graphical user interface, text, application

Description automatically generated

Set the flowrate to the desired flowrate (starting at 5 kg/s going to 50 kg/s in steps of 5). Also make sure to change the name of the storage CSV at the end of the file to make the desired flowrate. Use the screenshot below as an example of how to do this for 50 kg/s:





Set the control variables to the desired parameters and make sure the “WellsOrientation” variable is set to 3, as the non-parallel is the only well design being tested in this study. Use the screenshot below to locate the control variables:

Graphical user interface, text, application, email

Description automatically generated

Once all the control variables and the desired flowrate has been set, run the flowrate cell and wait for the code to be done. Then save the file as a CSV by running the final cell that was edited to match the flowrate earlier. Use the screenshot below to locate the flowrate cell.

Text

Description automatically generated

Reset the flowrate to the next desired flowrate interval and also change the name of the saved CSV file to match the next flowrate. Re-run the flowrate cell again and save to a CSV when the code is finished running.

All headers are included in the CSV files so the content of each column is known. The row numbers represent the various well angles, starting from 0.1 to 115 in steps of 0.1.

**1d. Running the Number of Fractures Study**

Make sure that the “FrictionFactors.ipynb” Jupyter Notebook is in the same folder as the “FractureREAL.ipynb” and “VaryNumberofPerfsWith1FractureZone.ipynb” Jupyter Notebooks.

This study consists of two Jupyter Notebooks because of the two sets of data desired from this study. The first Jupyter Notebook “FractureREAL.ipynb” is used to test different numbers of fractures from 1 to 30 fracture zones with 1 inlet perforation. The second Jupyter Notebook “VaryNumberofPerfsWith1FractureZone.ipynb” is used to test different number of inlet perforations from 1 to 30 with 1 fracture zone. These plots are then mirrored on top of each other for comparison.

This study was only done for the parallel well design, therefore the “WellsOrientation” variable for both Ju[yter Notebooks should always be set to 1, as shown in the screenshot below:

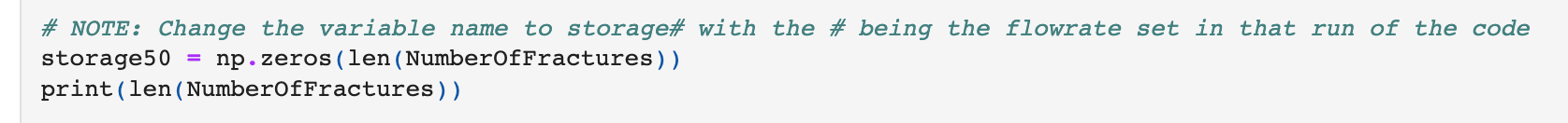
A picture containing timeline

Description automatically generated

Using the “FractureREAL.ipynb” Notebook, set the flowrate to the desired flowrate (starting at 5 kg/s to 50 kg/s in steps of 5). Also make sure to initialize the storage variable to the correct flowrate name, an example for 50 kg/s is shown in the screenshot below:

Graphical user interface

Description automatically generated with medium confidence

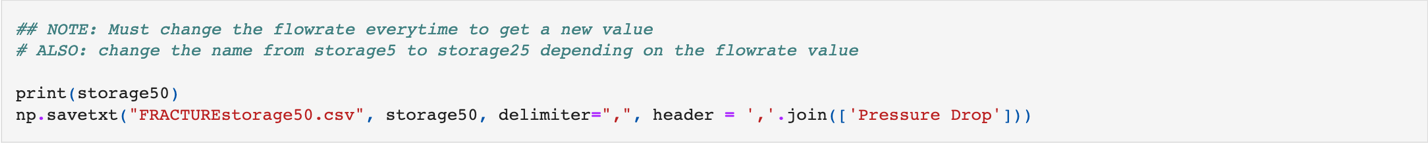


Once this is done, the name of the storage variable in the flowrate cell should also match this new storage initialization variable name as shown in the screenshot below:

A picture containing text

Description automatically generated

Also make sure to change the storage variables name in the final cell where it will be saved to a CSV, as shown in the screenshot below:



Once all the storage variable names are changed, run the main flowrate cell and wait for the code to finish, once the code is finished running save the data to a CSV file by running the last code cell. Use the screenshot below to identify where the flowrate cell is:

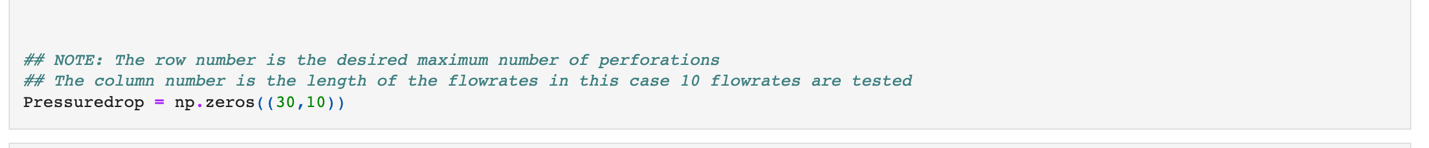
Text

Description automatically generated

Re-run the code with the next desired flowrate and change all the storage variable names and the “Flowrate\_Initial” variable.

All headers are included in the CSV files so the content of each column is known. The row numbers represent the number of fracture zones in the “FRACTUREstorage#.CSV” files. With row 1 representing 1 fracture zone and row 30 representing 30 fracture zones.

Next, open the “VaryNumberofPerfsWith1FractureZone.ipynb” Jupyter Notebook. Initialized the “Pressuredrop” variable to the correct dimensions as shown in the screenshot below:



Make sure all control variables are set to the desired parameters as shown in the screenshot below:

Graphical user interface, text, application

Description automatically generated

Run the main flowrate cell and wait for the code to finish, once the code is finished run the last cell of code and save it to a CSV. There is no need to change variable names with flowrates here, as all the flowrates are calculated one after the other automatically. Use the screenshot below to discern where the main flowrate cell is located:

Text

Description automatically generated with medium confidence

All headers are included in the CSV files so the content of each column is known. The row numbers represent the number of perforations in the “1Pressuredrop.CSV” file. With row 1 representing 1 inlet perforation and row 30 representing 30 inlet perforations.

1. **Generate Plots**

Depending on the desired plot select the appropriate “Compare” folder in the GitHub repository. Each “Compare” folder will have all the CSV files required to generate the plots as well as the Jupyter Notebooks used to generate those plots. Make sure that the respective Jupyter Notebook used has all the associated CSV files in the same folder in order to read the data. In each of the Jupyter Notebooks used to generate the plots a path is defined the save the plots to. Change this path to the desired path where the plots will be saved. The screenshot below shows what this path variable looks like and where to find it:

Graphical user interface

Description automatically generated

Once this is done, then all plots should be generated and saved to the desired paths.