Homework 6 Solutions

Problem 1

The following data corresponds to a leak-off test in an offshore well in the Gulf of Mexico.

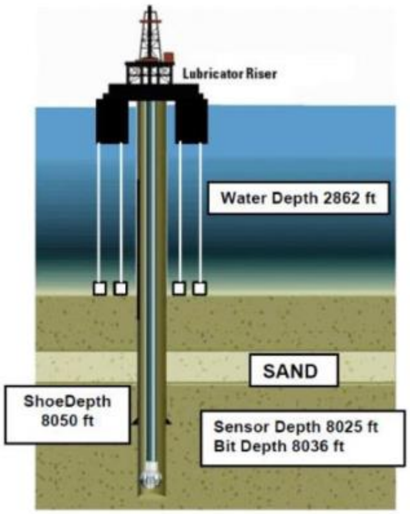
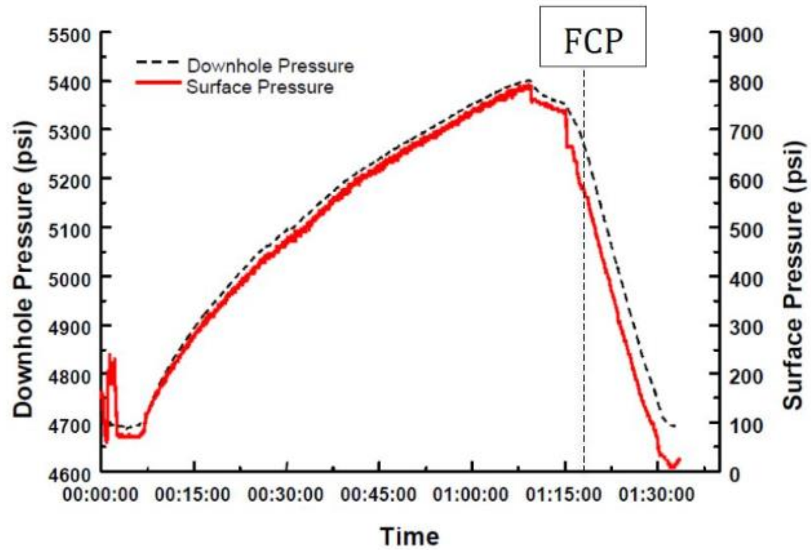


Fig. 5 – Schematic of Auger TLP Well Configuration



a) Estimate S_v at the shoe depth (TVDSS = 8050 ft).

We assume the pore pressure gradient is 0.44 psi/ft and the overburden gradient is 1 psi/ft.

S_v at 8050 ft is the summation of the overlying hydrostatic head of water and the remaining pressure head to that depth:

$$S_v = 2862 \times 0.44 + (8050 - 2862) \times 1 = 6447psi$$

b) Assuming the pore pressure is P_p = 4700 psi, and fracture closure occurs at time 1:18:00, calculate effective vertical stress σ_v and minimum effective stress σ_3.

At 1:18:00, the downhole pressure is approximately 5270 psi and the surface pressure is 550 psi.

$$\sigma_V = S_v - P_p = 6447 - 4700 = 1747psi$$

$$\sigma_{hmin} = P_{FCP} - P_p = 5270 - 4700 = 570psi$$

c) What is the faulting regime? Calculate the effective stress anisotropy ratio σ_v/σ_{hmin}.

Faulting regime is normal since $\sigma_v > \sigma_{hmin}$

$$\frac{\sigma_v}{\sigma_{hmin}} = \frac{1747}{570} = 3.06$$

d) What is the density of the drilling mud?

$BHP = 4600psi$ since downhole pressure reading at time 00:00:00 is 4700 psi and the surface pressure is approximately 100 psi

$$\rho_m = \frac{BHP}{8050ft} \times \frac{8.33ppg}{0.44psi/ft} = 10.81ppg$$

Problem 2

Download the file “MicrofracData.xls” which corresponds to a minifrac field test. The pressure reading corresponds to surface pressure.

a) Plot surface pressure and injection rate in a double y-axis plot as a function of time. Plot the entire interval and make a zoom from 70 to 90 min. TVD is 7503 ft

In [1]:
```
import pandas as pd

excel_file = 'HW6.xlsx'
DataQ2Summary = pd.read_excel(excel_file, sheet_name="Q2")
DataQ2Summary.head(721)
```

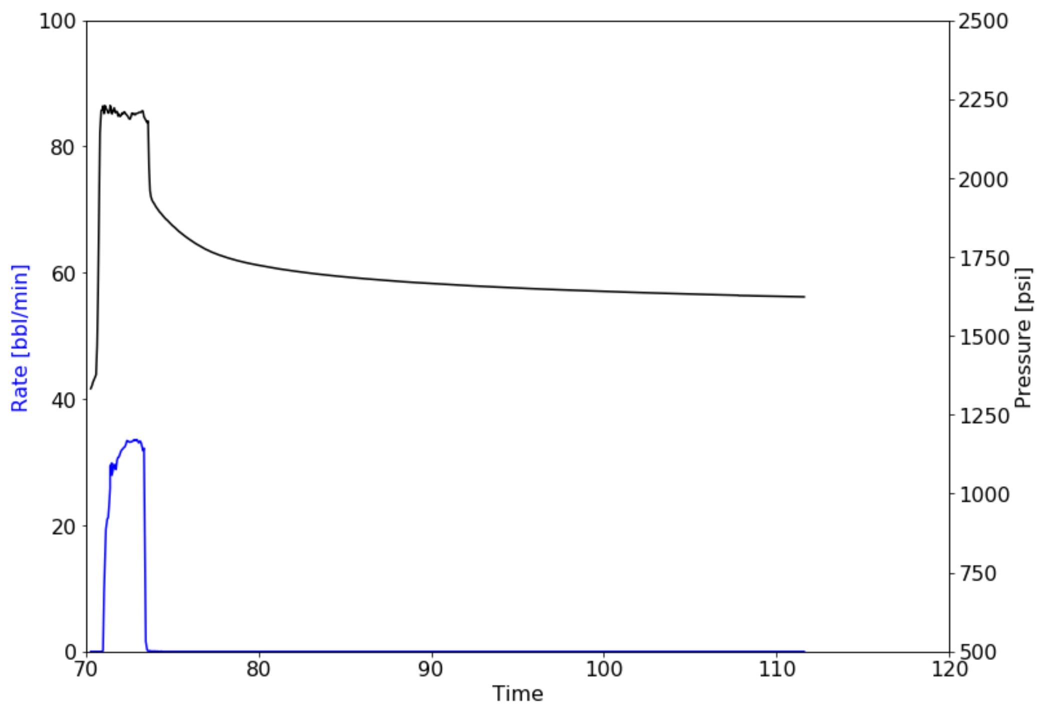
Out[1]:

	Time (minutes)	Pressure(psi)	Rate (gpm)	Sqrt(Time)
0	70.2900	1332.848	0.000	NaN
1	70.3626	1342.264	0.000	NaN
2	70.4220	1353.660	0.000	NaN
3	70.4946	1362.405	0.000	NaN
4	70.5474	1369.632	0.000	NaN
5	70.6002	1377.376	0.000	NaN
6	70.6596	1467.873	0.000	NaN
7	70.7124	1632.455	0.000	NaN
8	70.7652	1867.525	0.000	NaN
9	70.8246	2142.580	0.000	NaN
10	70.8972	2213.464	0.000	NaN
11	70.9500	2216.214	0.000	NaN
12	71.0028	2228.248	0.165	NaN
13	71.0622	2203.894	9.647	NaN
14	71.1150	2229.238	14.729	NaN
15	71.1678	2221.307	19.349	NaN
16	71.2470	2213.090	20.988	NaN
17	71.2998	2205.423	21.208	NaN
18	71.3526	2207.029	23.078	NaN
19	71.4120	2224.453	25.839	NaN
20	71.4120	2229.337	29.392	NaN
21	71.5176	2206.853	29.843	NaN
22	71.5176	2203.124	27.929	NaN
23	71.6496	2222.077	29.612	NaN
24	71.7024	2207.315	29.557	NaN
25	71.7552	2210.626	28.831	NaN
26	71.8146	2210.582	30.294	NaN
27	71.8674	2196.315	30.745	NaN
28	71.9202	2202.893	30.800	NaN
29	71.9796	2194.852	31.251	NaN
...	...	...	...	...

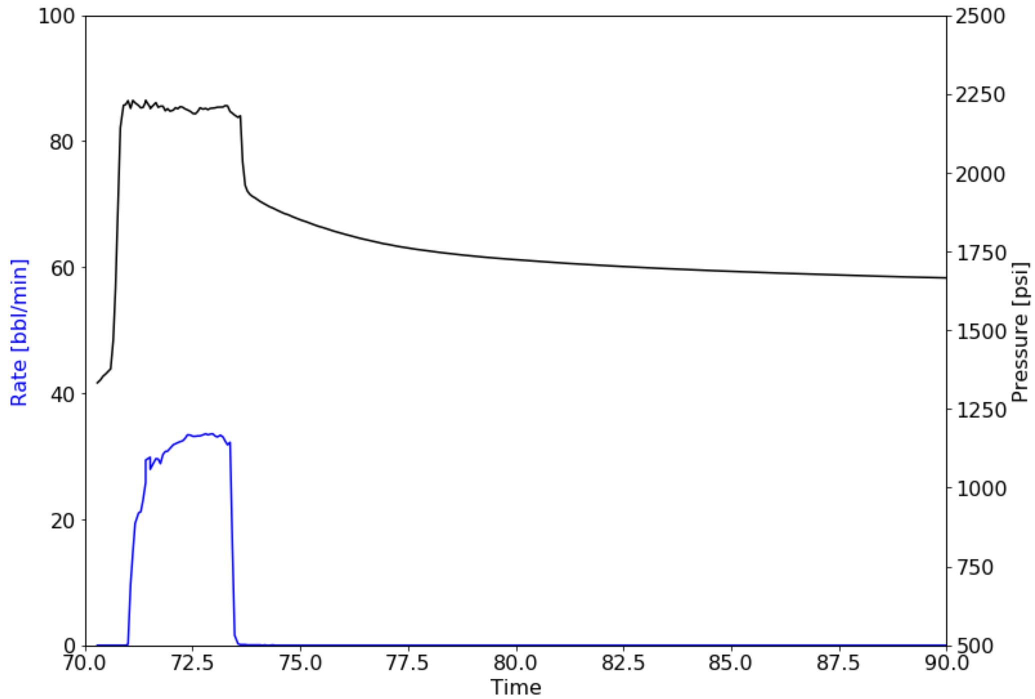
In [4]:
```python
import numpy as np
import matplotlib.pyplot as plt

Time = DataQ2Summary['Time (minutes)']
SQRT_Time = DataQ2Summary['Sqrt(Time)']
Pressure = DataQ2Summary['Pressure(psi)']
Rate = DataQ2Summary['Rate (gpm)']

# Plot data
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(Time, Rate, 'b-')
ax2.plot(Time, Pressure, 'k-')
# Plot labels
ax1.set_xlabel('Time')
ax1.set_ylabel('Rate [bbl/min]', color='b')
ax2.set_ylabel('Pressure [psi]', color='k')
# Axis range
plt.xlim([60, 120])
ax1.set_ylim(0, 100)
ax2.set_ylim(500, 2500)
plt.show()
# Change plot size
plt.rcParams["figure.figsize"] = (10,9)
```



In [5]:
```python
# Plot data
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(Time, Rate, 'b-')
ax2.plot(Time, Pressure, 'k-')
# Plot labels
ax1.set_xlabel('Time')
ax1.set_ylabel('Rate [bbl/min]', color='b')
ax2.set_ylabel('Pressure [psi]', color='k')
# Axis range
plt.xlim([70, 90])
ax1.set_ylim(0, 100)
ax2.set_ylim(500, 2500)
plt.show()
# Get current figure size
fig_size = plt.rcParams["figure.figsize"]
# print ("Current size:", fig_size)
# Set figure size
fig_size[0] = 12
fig_size[1] = 9
plt.rcParams["figure.figsize"] = fig_size
# Set font size
plt.rcParams.update({'font.size': 16})
```



b) Find the instantaneous shut-in pressure (ISIP) and make a plot of surface pressure as a function of square root of time. Find the fracture closure pressure (FCP) [surface pressure].

ISIP is estimated to be 1941 psi at time of 73.77 minutes.

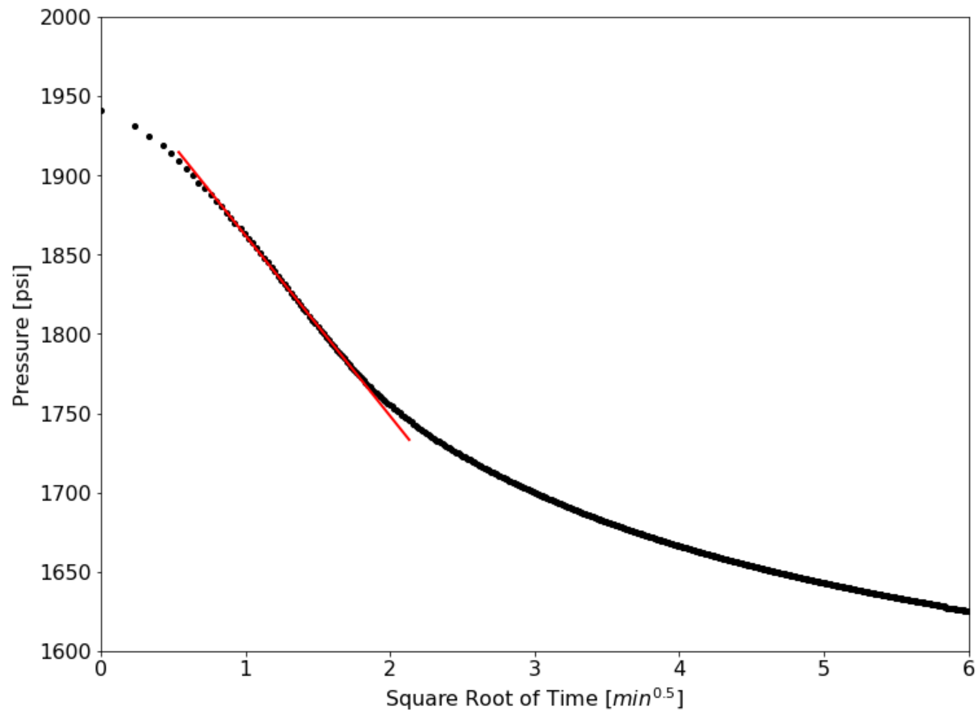
From the below figure, FCP is approximately 1770 psi (pressure departure from linear portion)

In [6]:

```
import numpy as np
import matplotlib.pyplot as plt

# Segment of the data set to plot
SQRT_Time1 = SQRT_Time.iloc[60:720]
Pressure1 = Pressure.iloc[60:720]
# Segment of the data set where pressure varies linearly with time^0.5
SQRT_Time2 = SQRT_Time.iloc[70:100]
Pressure2 = Pressure.iloc[70:100]
# Segment of the data set where we will plot the linear trend line
SQRT_Time3 = SQRT_Time.iloc[65:140]

# Plot data
plt.plot(SQRT_Time1, Pressure1, 'ok',markersize=4)
# Calculate the simple linear regression fit
coefficients = np.polyfit(SQRT_Time2, Pressure2, 1)
yy = np.poly1d(coefficients)
# Plot trendline
plt.plot(SQRT_Time3,yy(SQRT_Time3),"-r",linewidth=2)
# Plot labels
plt.xlabel('Square Root of Time [$min^{0.5}$]')
plt.ylabel('Pressure [psi]', color='k')
# Axis range
plt.xlim([0, 6])
plt.ylim([1600, 2000])
plt.show()
# Get current figure size
fig_size = plt.rcParams["figure.figsize"]
# print ("Current size:", fig_size)
# Set figure size
fig_size[0] = 12
fig_size[1] = 9
plt.rcParams["figure.figsize"] = fig_size
# Set font size
plt.rcParams.update({'font.size': 16})
```

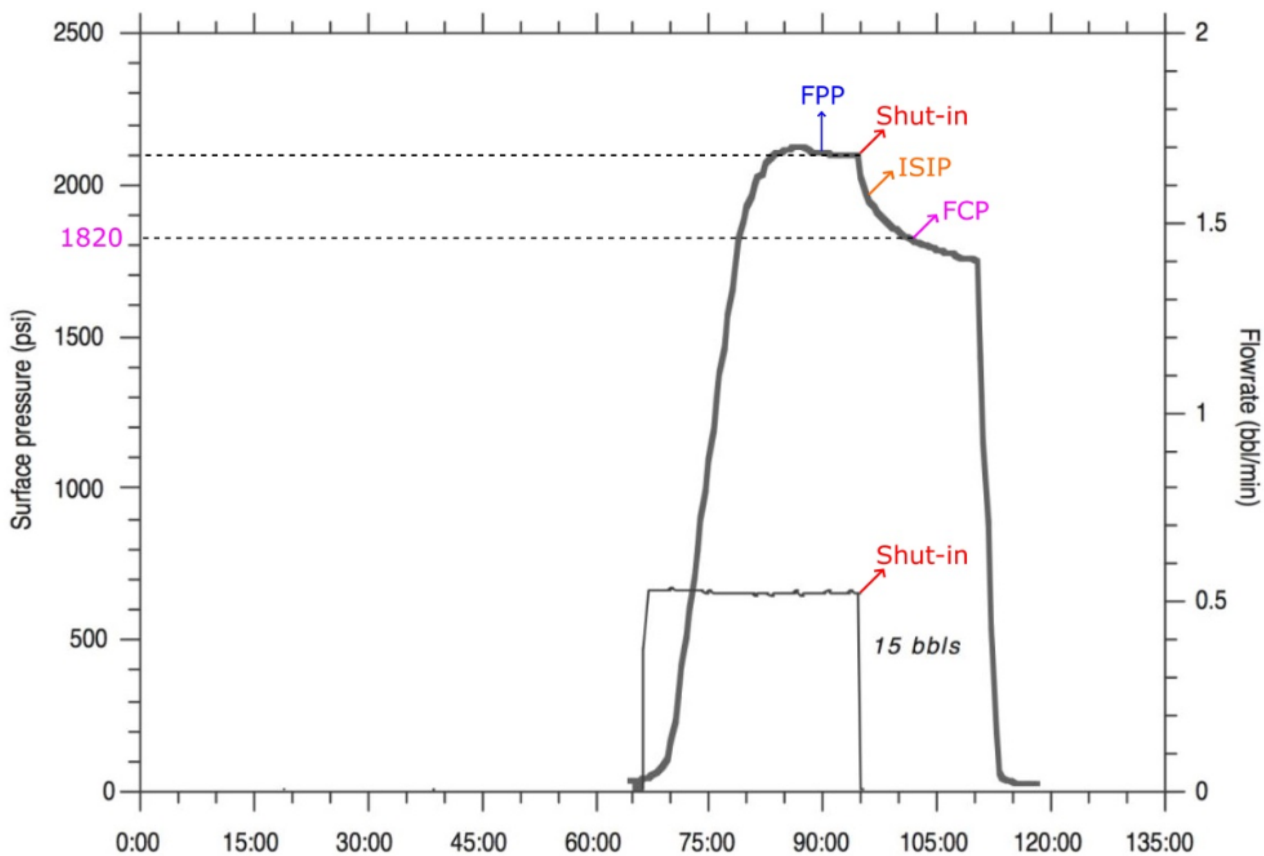


c) The true depth is 7,503 ft. Assuming a hydrostatic pressure gradient inside the wellbore of 0.44 psi/ft, calculate the minimum total principal stress S_3 in this place.

$$S_3 = FCP + P_W = 1770 + 0.44 \times 7503 = 5070psi$$

Problem 3

The figure below shows the results of a DFIT test (data from Zoback 2007).



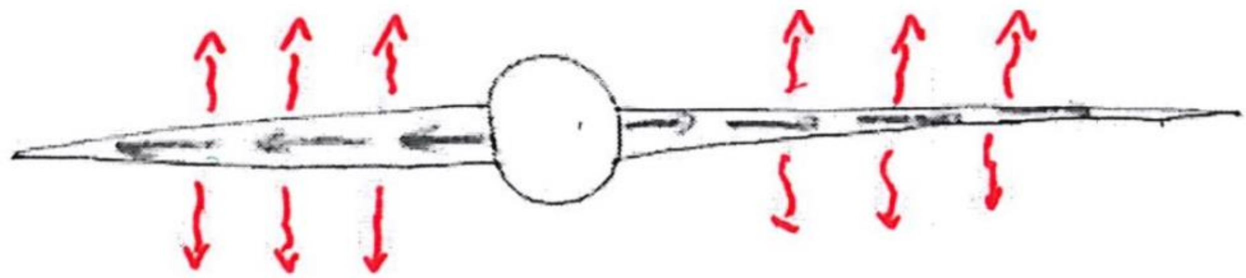
a) How many barrels of fracturing fluid were used in this test?

15 barrels of fracturing fluid

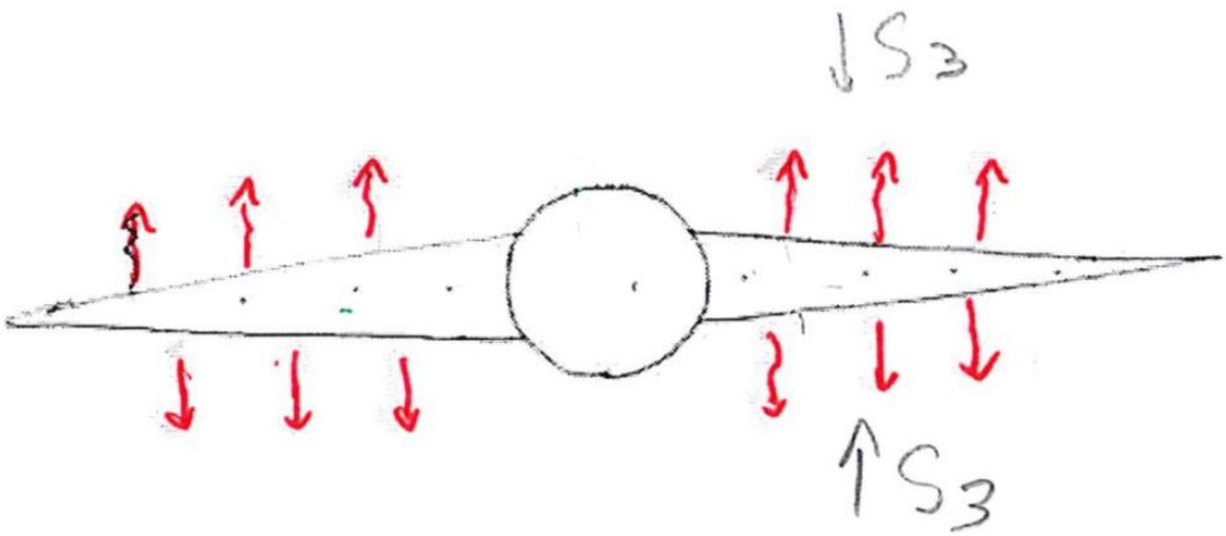
b) Indicate fracture propagation pressure (FPP), instantaneous shut-in pressure (ISIP) and fracture closure pressure (FCP) (as well as you can without analyzing the data in detail).

c) Describe and sketch the flow behavior around the wellbore before and after shut-in. ¶

Before shut-in,



After shut-in,



d) At surface pressure of 0 psi the bottom-hole pressure is 5,500 psi. What is the minimum principal stress in this formation?

Minimum principal stress is $S_3 = FCP + P_W = 1820 + 5500 = 7320psi$. Notice that the pressure needed for fracture propagation is a few hundred psi higher.

Problem 4

Interpret the following step-rate test data (i.e., find the formation parting pressure). Plot pressure vs. time for all steps.

In [5]:
```python
import pandas as pd

excel_file = 'HW6.xlsx'
DataQ4Summary = pd.read_excel(excel_file, sheet_name="Q4_Data")
DataQ4Summary.head(14)
```

Out[5]:

	Step #	Test Rate [bbl/min]	Test Rate (% of Max Rate)	-	-.1	-.2	-.3	-.4	-.5	-.6	-.7
0	1	0.2	5	Time [min]	0	5	10	15	20	25	30
1	-	-	-	Pressure [psi]	0	99	105	108	109	110	110
2	2	0.4	10	Time [min]	0	5	10	15	20	25	30
3	-	-	-	Pressure [psi]	88	187	204	215	219	220	220
4	3	0.8	20	Time [min]	0	5	10	15	20	25	30
5	-	-	-	Pressure [psi]	209	358	424	431	438	439	440
6	4	1.6	40	Time [min]	0	5	10	15	20	25	30
7	-	-	-	Pressure [psi]	418	770	869	871	875	878	882
8	5	2.4	60	Time [min]	0	5	10	15	20	25	30
9	-	-	-	Pressure [psi]	825	1089	1133	1199	1265	1298	1321
10	6	3.2	80	Time [min]	0	5	10	15	20	25	30
11	-	-	-	Pressure [psi]	1210	1375	1459	1507	1529	1535	1540
12	7	4	100	Time [min]	0	5	10	15	20	25	30
13	-	-	-	Pressure [psi]	1485	1595	1650	1683	1727	1749	1760

In [6]:
```python
import pandas as pd

excel_file = 'HW6.xlsx'
DataQ4Summary = pd.read_excel(excel_file, sheet_name="Q4_Plot1")
DataQ4Summary.head(55)
```

Out[6]:

	Step	Cumulative Time [min]	Time [min]	Pressure [psi]	Rate [bbl/min]
0	1	0	0.0	0.0	0.2
1	-	5	5.0	99.0	0.2
2	-	10	10.0	105.0	0.2
3	-	15	15.0	108.0	0.2
4	-	20	20.0	109.0	0.2
5	-	25	25.0	110.0	0.2
6	-	30	30.0	110.0	0.2
7	Pseudo	30	NaN	NaN	0.4
8	2	30	0.0	88.0	0.4
9	-	35	5.0	187.0	0.4
10	-	40	10.0	204.0	0.4
11	-	45	15.0	215.0	0.4
12	-	50	20.0	219.0	0.4
13	-	55	25.0	220.0	0.4
14	-	60	30.0	220.0	0.4
15	Pseudo	60	NaN	NaN	0.8
16	3	60	0.0	209.0	0.8
17	-	65	5.0	358.0	0.8
18	-	70	10.0	424.0	0.8
19	-	75	15.0	431.0	0.8
20	-	80	20.0	438.0	0.8
21	-	85	25.0	439.0	0.8
22	-	90	30.0	440.0	0.8
23	Pseudo	90	NaN	NaN	1.6

24	4	90	0.0	418.0	1.6
25	-	95	5.0	770.0	1.6
26	-	100	10.0	869.0	1.6
27	-	105	15.0	871.0	1.6
28	-	110	20.0	875.0	1.6
29	-	115	25.0	878.0	1.6
30	-	120	30.0	882.0	1.6
31	Pseudo	120	NaN	NaN	2.4
32	5	120	0.0	825.0	2.4
33	-	125	5.0	1089.0	2.4
34	-	130	10.0	1133.0	2.4
35	-	135	15.0	1199.0	2.4
36	-	140	20.0	1265.0	2.4
37	-	145	25.0	1298.0	2.4
38	-	150	30.0	1321.0	2.4
39	Pseudo	150	NaN	NaN	3.2
40	6	150	0.0	1210.0	3.2
41	-	155	5.0	1375.0	3.2
42	-	160	10.0	1459.0	3.2
43	-	165	15.0	1507.0	3.2
44	-	170	20.0	1529.0	3.2
45	-	175	25.0	1535.0	3.2
46	-	180	30.0	1540.0	3.2
47	Pseudo	180	NaN	NaN	4.0
48	7	180	0.0	1485.0	4.0
49	-	185	5.0	1595.0	4.0
50	-	190	10.0	1650.0	4.0
51	-	195	15.0	1683.0	4.0
52	-	200	20.0	1727.0	4.0
53	-	205	25.0	1749.0	4.0
54	-	210	30.0	1760.0	4.0

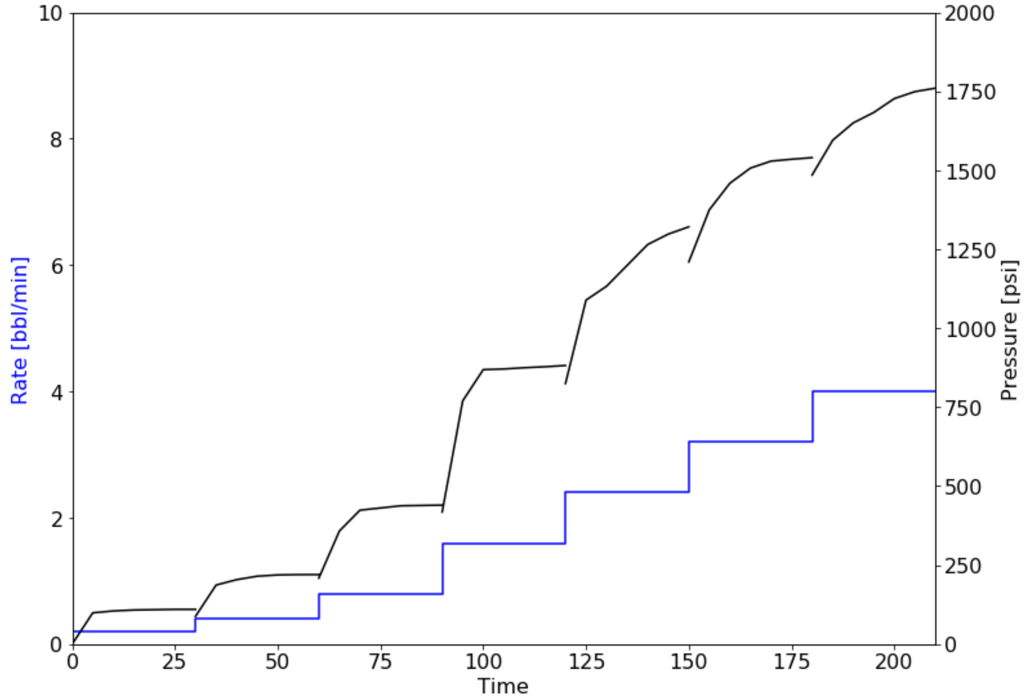
From the following plot, we can pick the maximum pressure at steady-state conditions

In [9]:

```
import numpy as np
import matplotlib.pyplot as plt

Time = DataQ4Summary['Cumulative Time [min]']
Pressure = DataQ4Summary['Pressure [psi]']
Rate = DataQ4Summary['Rate [bbl/min]']

# Plot data
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(Time, Rate, 'b-')
ax2.plot(Time, Pressure, 'k-')
# Plot labels
ax1.set_xlabel('Time')
ax1.set_ylabel('Rate [bbl/min]', color='b')
ax2.set_ylabel('Pressure [psi]', color='k')
# Axis range
plt.xlim([0, 210])
ax1.set_ylim(0, 10)
ax2.set_ylim(0, 2000)
plt.show()
# Get current figure size
fig_size = plt.rcParams["figure.figsize"]
# print ("Current size:", fig_size)
# Set figure size
fig_size[0] = 12
fig_size[1] = 9
plt.rcParams["figure.figsize"] = fig_size
# Set font size
plt.rcParams.update({'font.size': 16})
```



In [10]:
```
import pandas as pd

excel_file = 'HW6.xlsx'
DataQ4Summary = pd.read_excel(excel_file, sheet_name="Q4_Plot2")
DataQ4Summary.head(7)
```

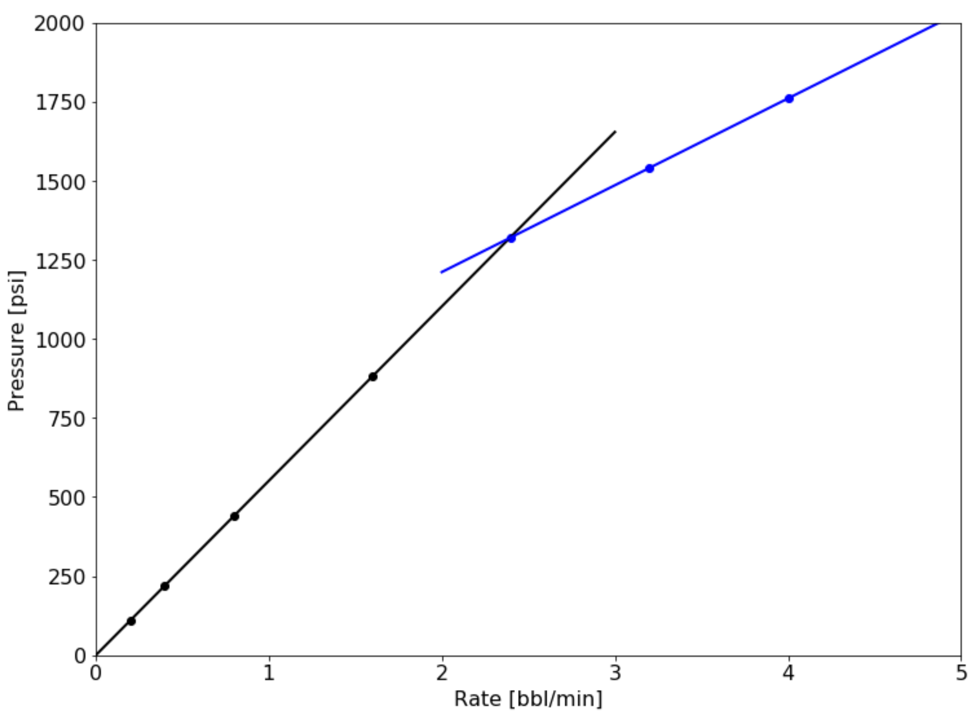
Out[10]:

	Rate [bbl/min]	Max Pressure [psi]	Formation Parting Pressure [psi]	1318.7482691666505
0	0.2	110	-	-
1	0.4	220	-	-
2	0.8	440	-	-
3	1.6	882	-	-
4	2.4	1321	-	-
5	3.2	1540	-	-
6	4.0	1760	-	-

In [11]:
```
import numpy as np
import matplotlib.pyplot as plt

Rate = DataQ4Summary['Rate [bbl/min]']
Pressure = DataQ4Summary['Max Pressure [psi]']
# Segment of the data set corresponding to the first linear trend
Rate1 = Rate.iloc[0:4]
Pressure1 = Pressure.iloc[0:4]
# Segment of the data set corresponding to the second linear trend
Rate2 = Rate.iloc[4:7]
Pressure2 = Pressure.iloc[4:7]

# Plot data
plt.plot(Rate1,Pressure1,'ok')
plt.plot(Rate2,Pressure2,'ob')
# Calculate the simple linear regression fit for the first linear trend
coefficients = np.polyfit(Rate1, Pressure1, 1)
yy = np.poly1d(coefficients)
# Plot trendline for first linear trend
Rate1 = np.arange(0,3.5)
plt.plot(Rate1,yy(Rate1),"-k",linewidth=2)
# Calculate the simple linear regression fit for the first linear trend
coefficients = np.polyfit(Rate2, Pressure2, 1)
yy = np.poly1d(coefficients)
# Plot trendline for first linear trend
Rate2 = np.arange(2,6,1)
plt.plot(Rate2,yy(Rate2),"-b",linewidth=2)
# Plot labels
plt.xlabel('Rate [bbl/min]')
plt.ylabel('Pressure [psi]')
# Axis range
plt.xlim([0, 5])
plt.ylim([0, 2000])
plt.show()
# Get current figure size
fig_size = plt.rcParams["figure.figsize"]
# print ("Current size:", fig_size)
# Set figure size
fig_size[0] = 12
fig_size[1] = 9
plt.rcParams["figure.figsize"] = fig_size
# Set font size
plt.rcParams.update({'font.size': 16})
```



From above graph, the formation parting pressure is 1320 [psi], the intersection between the two trends.