ESN-583 Petrophysics and Seismic Rock Characterization

Report on Chalk Data Tutorial

Name: Sobhit Soni **Enrolment: 21411033** **Branch: GPT, 4th Year**

1). Calculate porosity using bulk density, grain density, and fluid density. (Use mineral fraction information for calculating exact grain density) otherwise use the grain density of the dominant mineral.

Methodology: By using these following formulas, we can calculate porosity using grain density, Bulk density, and fluid density.

The porosity using density is given by

$$\phi = \frac{\rho_g - \rho_b}{\rho_g - \rho_f}$$

Where

ρ_g is grain density

ρ_b = bulk density

ρ_f is fluid density.

Note:

- **Rho_fluid** is assumed to be 1gm/cc (Water).
- **Rho_Grain** is calculated by using mineral fraction data provided.

Formula for Bulk Density from Composite Minerals

$$\rho_{\text{bulk}} = \sum_{i=1}^{n} V_i \cdot \rho_i$$

Where:

- ρ_{bulk} = bulk density of the rock (g/cm³ or kg/m³)
- V_i = volume fraction of the i-th mineral (must sum to 1)
- ρ_i = density of the i-th mineral (same units)

Code Snippet:

```
# Calculate Grain Density by using mineral fraction data
Carb=(df["Carb"].iloc[1:])/100
Q=(df["Q"].iloc[1:])/100
Cl=(df["Cl"].iloc[1:])/100
# Constants are used from the Header
Rho_Grain=(Carb*2.710+Q*2.650+Cl*2.800)
print(Rho_Grain)
```

```
# calculate Porosity using calculated Rho_grain
import numpy as np
Porosity_Cal=(Rho_Grain-df["Sat. Blk. Den"].iloc[1:])/(Rho_Grain-1)
Porosity_Cal=np.where(Porosity_Cal>=0,Porosity_Cal,np.nan)  #replace the negative porosity with NaN
Porosity=(df["Porosity  "].iloc[1:])/100  #But we will use the given porosity throughout this work
print(Porosity)
```

2). Calculate Bulk modulus: dry and saturated, Shear Modulus: dry and saturated, Poisson's Ratio: Dry and Saturated

Methodology:

For Bulk Modulus(K):

Dry: -

Formula:

$$K_{\text{dry}} = \rho_{\text{dry}} \left(V_{p\_dry}^2 - \frac{4}{3} V_{s\_dry}^2 \right)$$

- Where:
- K_{dry} = dry bulk modulus (Pa or GPa)
- ρ_{dry} = bulk density of the dry rock (kg/m³)
- V_{p_dry} = dry P-wave velocity (m/s)
- V_{s_dry} = dry S-wave velocity (m/s)

Saturated: -

Gassmann's Relation

$$K_{sat} = K_{frame} + \frac{\left(1 - \frac{K_{frame}}{K_{matrix}}\right)^2}{\frac{\phi}{K_{fl}} + \frac{(1-\phi)}{K_{matrix}} - \frac{K_{frame}}{K_{matrix}^2}}$$

K_{sat} Bulk modulus of saturated rock
K_{frame} Bulk modulus of dry rock frame
K_{matrix} Bulk modulus of the mineral
K_{fl} Bulk modulus of the fluid
G_{dry} Shear modulus of the dry rock

Note: -

- Kdry=Kframe & Kmatrix=Kmineral

For Shear Modulus(G):

Dry: -

Formula:

$$\mu = \rho_{\text{dry}} \cdot V_{s\_dry}^2$$

Where:

- μ or G = shear modulus (Pa or GPa)
- ρ_{dry} = dry bulk density of the rock (kg/m^3)
- V_{s_dry} = dry S-wave velocity (m/s)

Saturated: -

$$G_{sat} = G_{dry}$$

Assumption: - Fluid shows no yield in shear modulus

Code Snippet:

```
#Calculate Dry Bulk Modulus and Shear Modulus
Rho_dry=(df["Dry Blk. Den"].iloc[1:])
Vp_dry=df["Dry Ver. Vp"].iloc[1:]
Vs_dry=df["Dry Ver.Vs "].iloc[1:]
K_Dry=Rho_dry*((Vp_dry)**2-(4/3)*(Vs_dry)**2)  #in GPa
G_Dry=Rho_dry*((Vs_dry)**2) #in GPa
print(G_Dry)
```

```
#Calculate Sat Bulk Modulus and Shear Modulus using Gassman's Point prediction Method
Biot_Coeff=(1-K_Dry/K_VRH)
K_Sat=K_Dry+((Biot_Coeff**2)/((Porosity/2.25-Porosity/K_VRH)+Biot_Coeff/K_VRH))
G_Sat=G_Dry
print(K_Sat)
```

For <u>Poisson's Ratio(v): -</u>

3. Relation between Bulk Modulus (K), Shear Modulus (G), and Poisson's Ratio (ν):

$$K = \frac{2G(1+\nu)}{3(1-2\nu)}$$

Or rearranged:

$$\nu = \frac{3K-2G}{2(3K+G)}$$

Code Snippet: -

```
# calculate Sat and Dry Poissons Ratio

Poisson_R_Dry=(3*K_Dry-2*G_Dry)/(2*(3*K_Dry+G_Dry))
Poisson_R_Sat=(3*K_Sat-2*G_Sat)/(2*(3*K_Sat+G_Sat))
```

3) Calculate Biot's coefficient using a critical porosity (PHIc) of 50%.

Methodology: -

◆ **Empirical Approximation Using Critical Porosity**

One common approximation (based on the critical porosity concept in rock physics) is:

$$\alpha \approx \frac{\phi}{\phi_c}$$

Where:

- ϕ = porosity of the rock
- ϕ_c = critical porosity (typically ~0.4 for sandstones)

Note: here alpha is Biot's coefficient

Code snippet: -

```
# Calculate Biot's Coefficient using PHIC=50%
Biot_Coeff_Critical=Porosity/0.5
```

4) Calculate Vertical Overburden Stress, Terzagh's Stress (Differential), and Biot's Stress (Effective).

Methodology: -

- Confining pressure (Overburden pressure) = load or pressure exerted by the overlying layers and fluids (Pc)
- Pore pressure (Formation pressure) = pressure exerted by pore fluids (Pp)
- Differential pressure = Pressure acting on frame (Pd)

  Pd = Pc - Pp
- Effective pressure = corrected pressure (Pe)

  Pe = Pc - nPp

Note:

- Avg Overburden Pressure Gradient (Pc)=1Psi/ft and Avg Pore Pressure Gradient (Pp)=0.45Psi/ft is assumed. (here n is Biot's Coefficient)

Code snippet:

```
# Stress Calculation
VOP=1*df["TVD.1"].iloc[1:] #Vertical Overburden Pressure in Psi
PP=0.45*df["TVD.1"].iloc[1:] #Pore Pressure in Psi
Diff_Pressure=VOP-PP #Differential or Terzhag's Pressure
EFF_Pressure=VOP-Biot_Coeff*PP #Effective Pressure
print(EFF_Pressure)
```

5) Calculate the upper and lower bound for the given system using both Voigt-Reuss and Hashin-Shtrikman Method. Use hint from point no. 1.

Methodology: -

(i) For <u>Voigt Bound (or Upper Bound): -</u>

The effective properties were estimated to be **highest in the direction parallel to the fibers** and represented as

$$E_c = fE_f + (1-f)E_{matrix}$$

Where, E_c, E_f and E_{matrix} is the material property of the composite, fiber, and matrix, and f is the volume fraction of fiber

Note: here Ec(Effective Composite Bulk Modulus(K)), f is the porosity, Ef and Ematrix are the Bulk Modulus of the Dominant minerals of fluid and Matrix respectively. (here water and Calcite are the dominant minerals for fluid part and matrix part respectively).

(ii) For Reuss Bound (or Lower Bound): -

The **inverse rule of mixtures** states that in the direction perpendicular to the fibers, the elastic modulus of a composite can will be its lowest and represented as:

$$E_c = \left(\frac{f}{E_f} + \frac{(1-f)}{E_{matrix}}\right)^{-1}$$

https://www.doitpoms.ac.uk/tlplib/fibre_composites/stiffness.php

Code snippet: -

```
# Voigt and Reuss Bounds using Dominant minerals
import numpy as np
por=np.arange(0,1,0.01)  #Porosity values ranging [0:1] with a step of 0.01

K_VB=por*2.25+(1-por)*71            #Composite Voigt Upper Bound using Dominant minerals for K
K_RB=(por/2.25+(1-por)/71)**(-1)    #Composite Reuss Lower Bound using Dominant minerals for K
G_VB=(1-por)*30      #Composite Voigt Upper Bound using Dominant minerals for G
G_RB=0*por           #Composite Reuss Lower Bound using Dominant minerals for G
K_VRHB=(K_VB+K_RB)/2 #To be used in Hassian-Shritkman's Bound
G_VRHB=(G_VB+G_RB)/2 #To be used in Hassian-Shritkman's Bound
```

(iii) For Hashin-Shtrikman Bounds: -

For a two phase medium the resultant effective moduli are given by:

$$K^{HS\pm} = K_1 + \frac{f_2}{(K_2 - K_1)^{-1} + f_1\left(K_1 + \frac{4}{3}\mu_1\right)^{-1}}$$

$$\mu^{HS\pm} = \mu_1 + \frac{f_2}{(\mu_2 - \mu_1)^{-1} + \frac{2f_1(K_1+2\mu_1)}{5\mu_1(K_1+\frac{4}{3}\mu_1)}}$$

These give upper bounds when stiff material is K1, μ1 (shell) and lower bounds when soft material is K1, μ1.
f_1, f_2 are the fraction of individual component (phases),
K_1, K_2 are the bulk moduli of individual phases
μ_1, μ_2 are the shear moduli of individual phases

Code snippet: -

```
# Hassian Shritkman Bounds
K_HS_U=K_VB+(por/((2.25-K_VB)**(-1)+(1-por)*(K_VB+(4/3)*G_VB)**(-1)))
K_HS_L=2.25+(1-por)/((K_VRHB-2.25)**(-1)+por*(2.25)**(-1))
```

6) Calculate Gassmann's response for each of the samples using the given information about dry state of the samples, and the mineral fractions. Use the same fluid as in point no.1.

Methodology: -

- Kdry is calculated first from the Dry density, Vp_dry and Vs_dry.
- Kmatrix is calculated by taking average of K Voigt and K Reuss (i.e. KVRH) using mineral fraction data.
- K Voigt and K Reuss now are in Stiffer Domain for accurate value.
- KVRH is the average taken as we don't have the information about orientation of formation.
- Water is assumed to be only fluid. (i.e. Kfl=2.25 Gpa)
- Gassmann's relation adds the effect of frame (i.e. Kdry or Kframe) with fluid (i.e. Kfl).

Gassmann's Relation

$$K_{sat} = K_{frame} + \frac{\left(1 - \frac{K_{frame}}{K_{matrix}}\right)^2}{\frac{\phi}{K_{fl}} + \frac{(1-\phi)}{K_{matrix}} - \frac{K_{frame}}{K_{matrix}^2}}$$

K_{sat}	Bulk modulus of saturated rock
K_{frame}	Bulk modulus of dry rock frame
K_{matrix}	Bulk modulus of the mineral
K_{fl}	Bulk modulus of the fluid
G_{dry}	Shear modulus of the dry rock

Code snippet: -

```
#Calculate Sat Bulk Modulus and Shear Modulus using Gassman's Point prediction Method
Biot_Coeff=(1-K_Dry/K_VRH)
K_Sat=K_Dry+((Biot_Coeff**2)/((Porosity/2.25-Porosity/K_VRH)+Biot_Coeff/K_VRH))
G_Sat=G_Dry
print(K_Sat)
```

7) Recalculate the compressive- and shear- wave velocity of the samples using Gassmann's output.

Methodology: -

✅ Formulas:

- **P-wave velocity (saturated):**

$$\boxed{V_{p\_sat} = \sqrt{\frac{K_{\text{sat}} + \frac{4}{3}G_{\text{sat}}}{\rho_{\text{sat}}}}}$$

- **S-wave velocity (saturated):**

$$\boxed{V_{s\_sat} = \sqrt{\frac{G_{\text{sat}}}{\rho_{\text{sat}}}}}$$

Note:

- here **Ksat** is calculated from Gassmann's relation

Code snippet: -

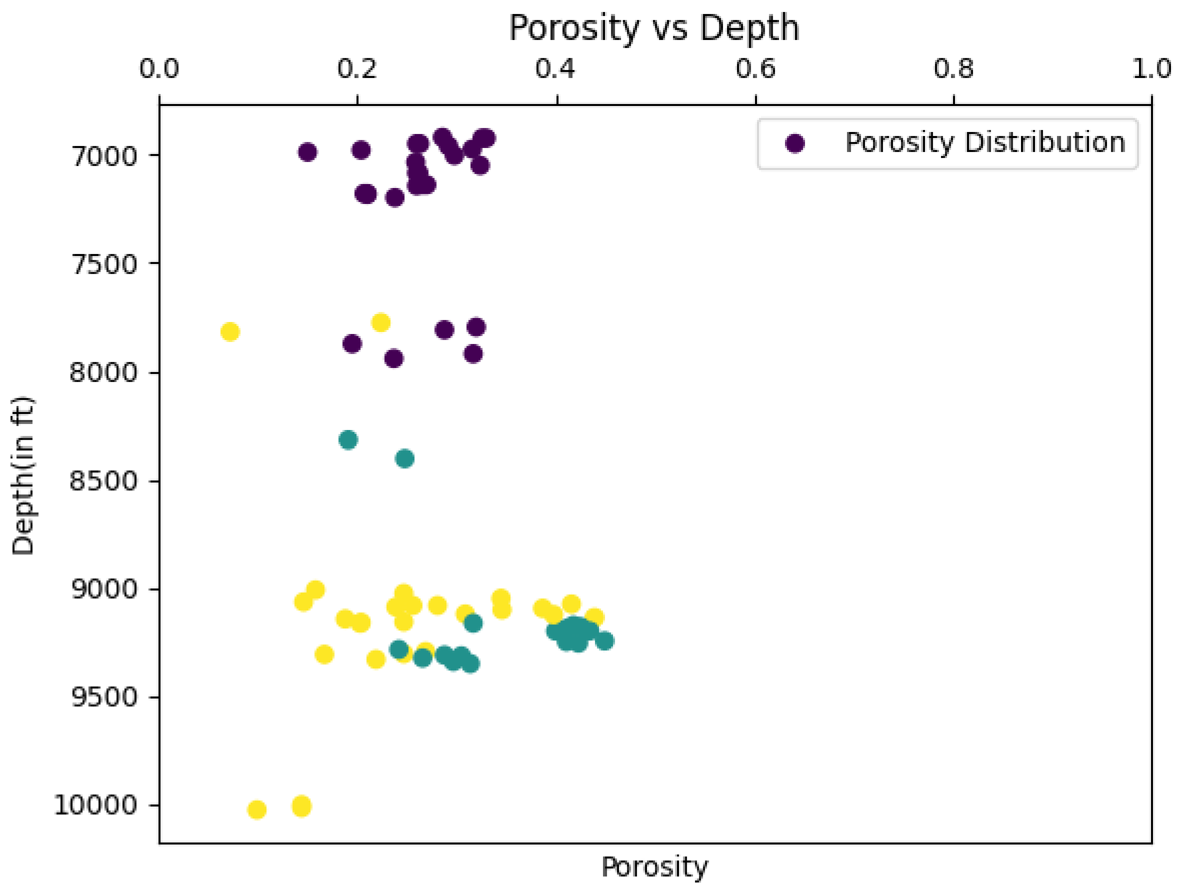
```
# Vp and Vs Sat and dry from Gassman's Output

Rho_Sat=df["Sat. Blk. Den"].iloc[1:]
Vp_Sat_Gassman=((K_Sat+(4/3)*G_Sat)/Rho_Sat)**0.5
Vp_Dry_Gassman=((K_Dry+(4/3)*G_Dry)/Rho_dry)**0.5

Vs_Sat_Gassman=((G_Sat)/Rho_Sat)**0.5
Vs_Dry_Gassman=((G_Dry)/Rho_dry)**0.5
```

8) Plot the porosity trend with depth.

Code snippet: -

```
# Plot Porosity trend vs Depth
color_bar=df["Formation"].iloc[1:]
color_bar[color_bar=="chalk"]=1
color_bar[color_bar=="Tor"]=2
color_bar[color_bar=="Ekofisk"]=3
plt.scatter(Porosity,df["TVD.1"].iloc[1:],c=color_bar,cmap='viridis',label='Porosity Distribution')
plt.xlim(0,1)
plt.ylabel("Depth(in ft)")
plt.xlabel("Porosity")
plt.gca().invert_yaxis()
plt.gca().xaxis.tick_top()
plt.title("Porosity vs Depth")
plt.legend(loc='best')
```

Plot: -



Comments:

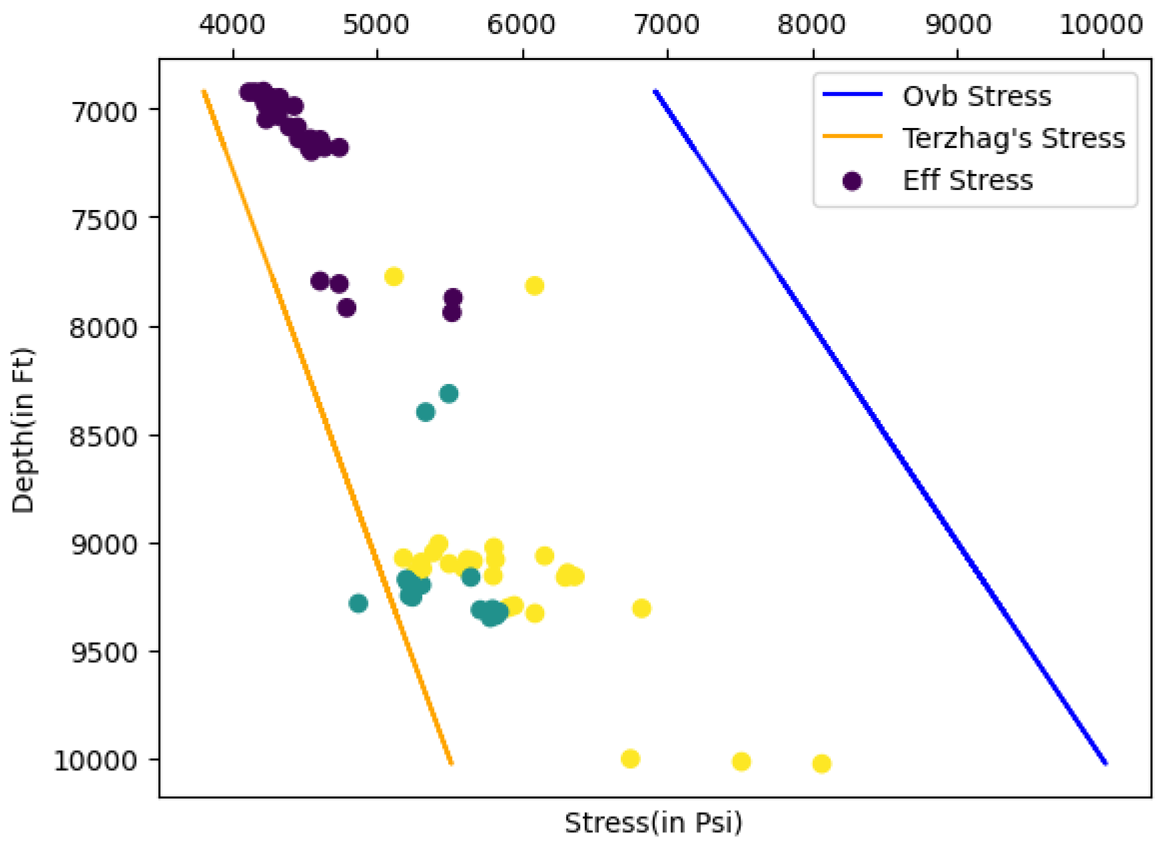
- Porosity of Chalk usually ranges from 35-47% but here you can see at 7000 ft depth, chalk formation having mainly less than 30% indicating the Diagenesis process would have led to compaction and dissolution.
- At all depths, large range of porosity distribution can be seen making it a multimodal case (i.e. Heterogeneous formation).
- High porosity (nearly 40%) can be observed at 9000 ft depth, indicating a porous formation.

9) Plot the Stress trend with depth

Code snippet: -

```
# plot stress trend with Depth
plt.plot(VOP,df["TVD.1"].iloc[1:],color='blue',label='Ovb Stress')
plt.plot(Diff_Pressure,df["TVD.1"].iloc[1:],color='orange',label="Terzhag's Stress")
plt.scatter(EFF_Pressure,df["TVD.1"].iloc[1:],c=color_bar,cmap='viridis',label='Eff Stress')
plt.gca().invert_yaxis()
plt.gca().xaxis.tick_top()
plt.xlabel("Stress(in Psi)")
plt.ylabel("Depth(in Ft)")
plt.legend(loc='best')
```

Plot: -



Comments:

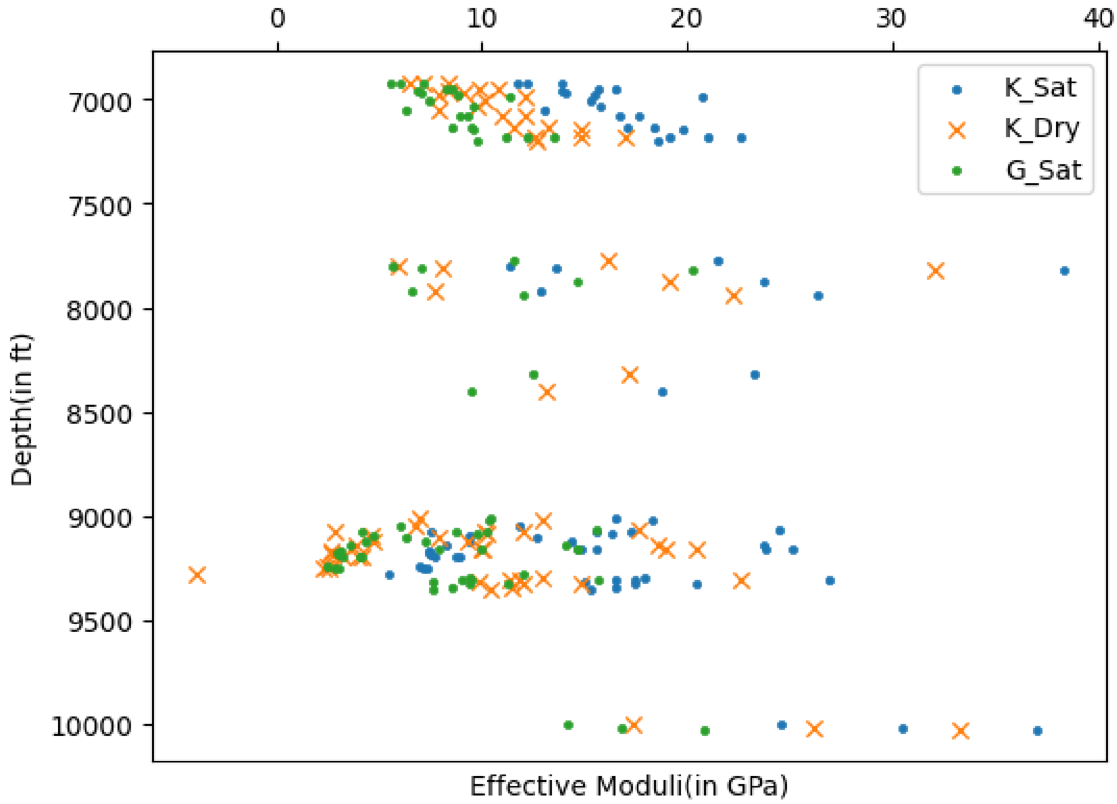
- Effective Pressure showing nearly linear trend at Depth ~7000ft (i.e. chalk formation) implying the good, connected pore network and close to differential pressure indicating that the pore pressure is significantly felt in vertical direction resulting in normal zone.
- Effective Pressure is distributed over large range at deeper formation and showing offset with differential pressure indicating heterogeneous and underpressure formations.

10) Plot the Modulus trend with depth

code snippet: -

```
# Effective Moduli vs Depth

plt.scatter(K_Sat,df["TVD.1"].iloc[1:],label='K_Sat',linewidths=0.5,marker='.')
plt.scatter(K_Dry,df["TVD.1"].iloc[1:],label='K_Dry',linewidths=1,marker='x')
plt.scatter(G_Sat,df["TVD.1"].iloc[1:],label='G_Sat',linewidths=0.5,marker='.')
plt.gca().invert_yaxis()
plt.gca().xaxis.tick_top()
plt.xlabel("Effective Moduli(in GPa)")
plt.ylabel("Depth(in ft)")
plt.legend(loc='best')
```

Plot: -



Comments:

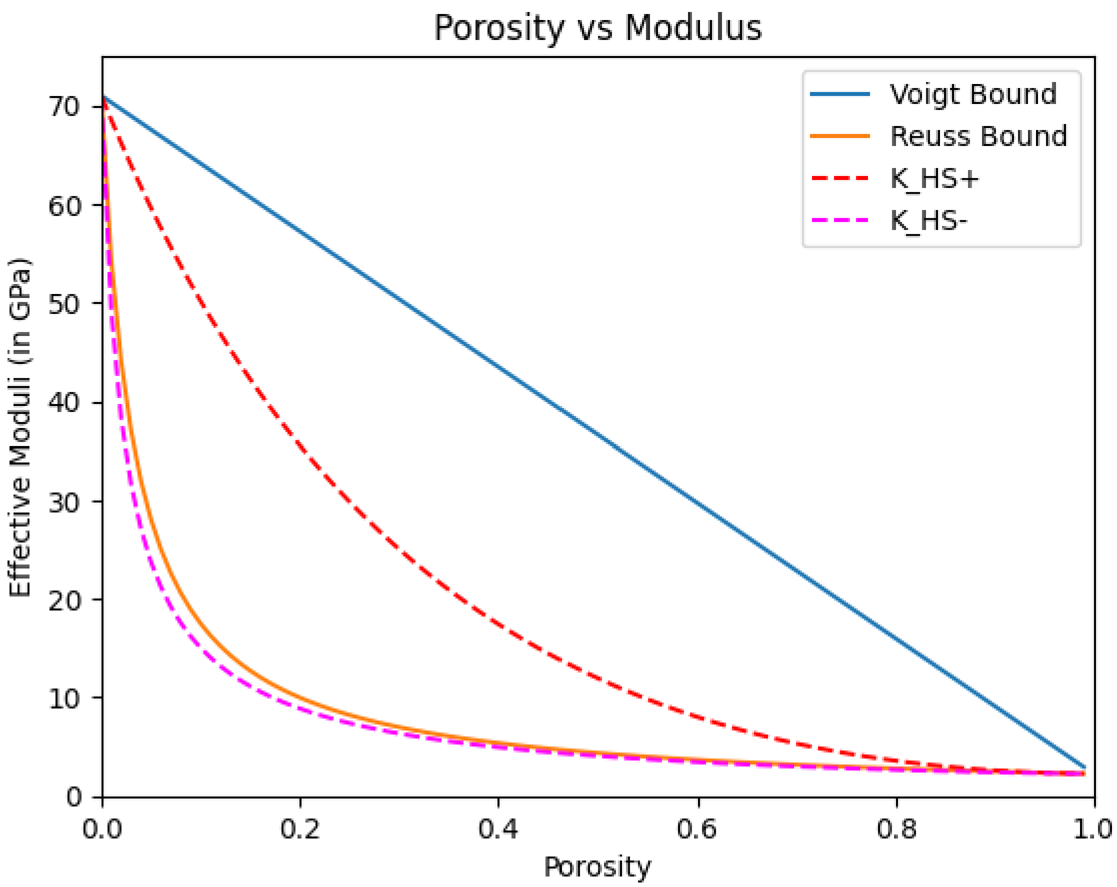
- Large variability at all depths.
- Chalk formation's Ksat usually ranges between 2-10 GPa for high to moderate porosity but here Ksat goes beyond 20 GPa implying the non-compliant frame formed by cementation, low porosity validated by diagenesis.

11) Plot the normal V-R and the normal Ha-Sh bounds on the same template

Code snippet: -

```
# V-R Bounds and Hassian Shritkman Bounds

plt.plot(por,K_VB,label='Voigt Bound')
plt.plot(por,K_RB,label='Reuss Bound')
plt.xlim(0,1)
plt.ylim(0,75)
plt.xlabel("Porosity")
plt.ylabel("Effective Moduli (in GPa)")
plt.title("Porosity vs Modulus")


K_HS_U=K_VB+(por/((2.25-K_VB)**(-1)+(1-por)*(K_VB+(4/3)*G_VB)**(-1)))
K_HS_L=2.25+(1-por)/((K_VRHB-2.25)**(-1)+por*(2.25)**(-1))
plt.plot(por,K_HS_U,linestyle='--',color='red',label='K_HS+')
plt.plot(por,K_HS_L,linestyle='--',color='magenta',label='K_HS-')
plt.legend(loc='best')
```

Plot: -



Comments:

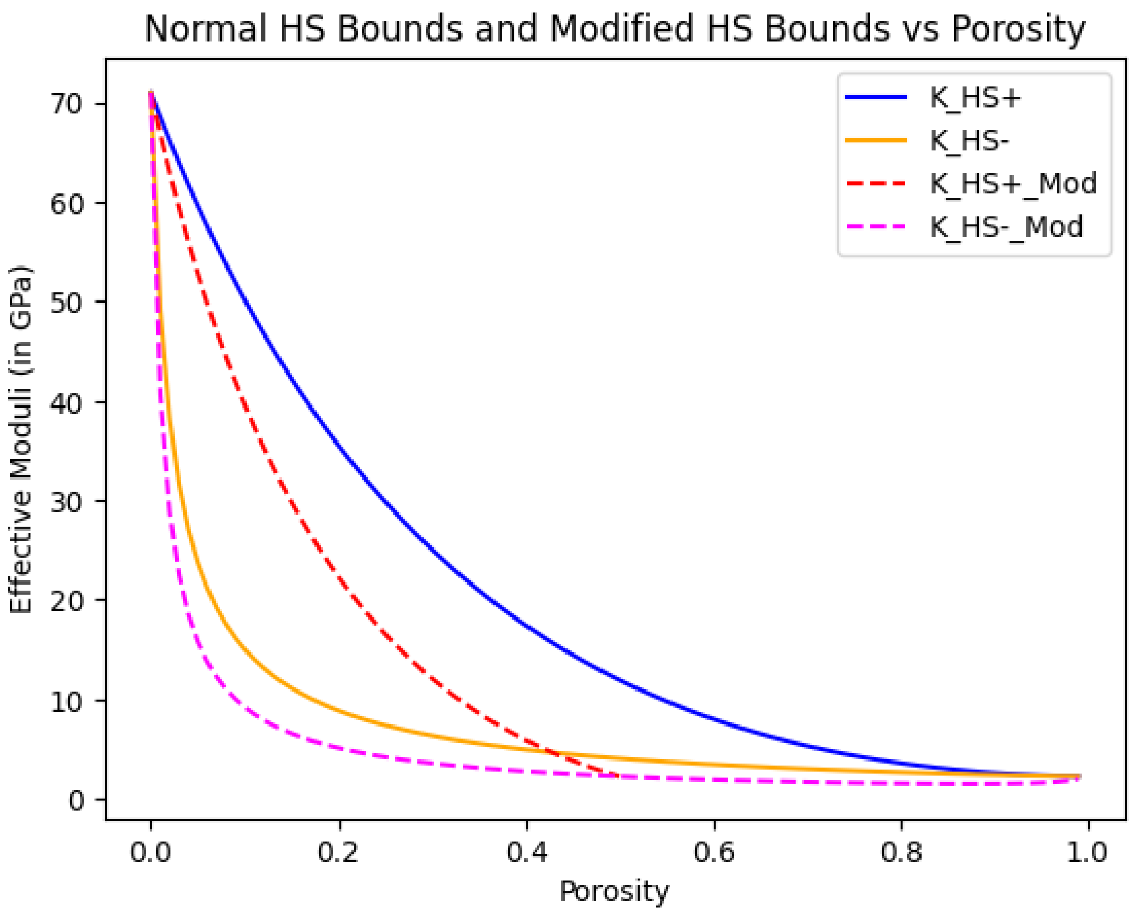
- K_HS+ gives the more accurate physical bound by accounting the KVRH (Voigt Reuss Hill Average).
- K_HS- and KR are nearly the same.

12) Plot the normal Ha-Sh bounds with its modified bounds.

Code snippet: -

```
# Modified Hassian-Shritkman
por_critical=0.50  #critical porosity = 50% is asked
por_new=por/por_critical
K_HS_U_Mod=K_VB+por_new/((2.25-K_VB)**(-1)+(1-por_new)*(K_VB+(4/3)*G_VB)**(-1))
K_HS_L_Mod=2.25+(1-por_new)/((K_VRHB-2.25)**(-1)+por_new*(2.25)**(-1))
#plt.plot(por,K_VB,color='blue',linestyle='-',label='K_VB')
#plt.plot(por,K_RB,color='green',linestyle='-',label='K_RB')
plt.plot(por,K_HS_U,color='blue',linestyle='-',label='K_HS+')
plt.plot(por,K_HS_L,color='orange',linestyle='-',label='K_HS-')
plt.plot(por[por<=0.5],K_HS_U_Mod[por<=0.5],color='red',linestyle='--',label='K_HS+_Mod')
plt.plot(por,K_HS_L_Mod,color='magenta',linestyle='--',label='K_HS-_Mod')
color_bar=df["Formation"].iloc[1:]
color_bar[color_bar=="chalk"]=1
color_bar[color_bar=="Tor"]=2
color_bar[color_bar=="Ekofisk"]=3
#plt.scatter(Porosity,K_Sat,c=color_bar,cmap='viridis',marker='.',linewidths=1,label='K_Sat')
plt.xlabel("Porosity")
plt.ylabel("Effective Moduli (in GPa)")
plt.title("Normal HS Bounds and Modified HS Bounds vs Porosity")
plt.legend(loc='best')
```

Plot: -



Comments:

- In all Physical bounds, the effect of Critical porosity was not in place resulting in unnecessary extension of bound at higher porosity, the Modified K_HS+ sets the bounds considering the effect of Critical porosity on Effective Moduli by truncating up to PHIC.
- At higher values than Critical porosity, the load bearing phase is fluid (i.e. suspension).

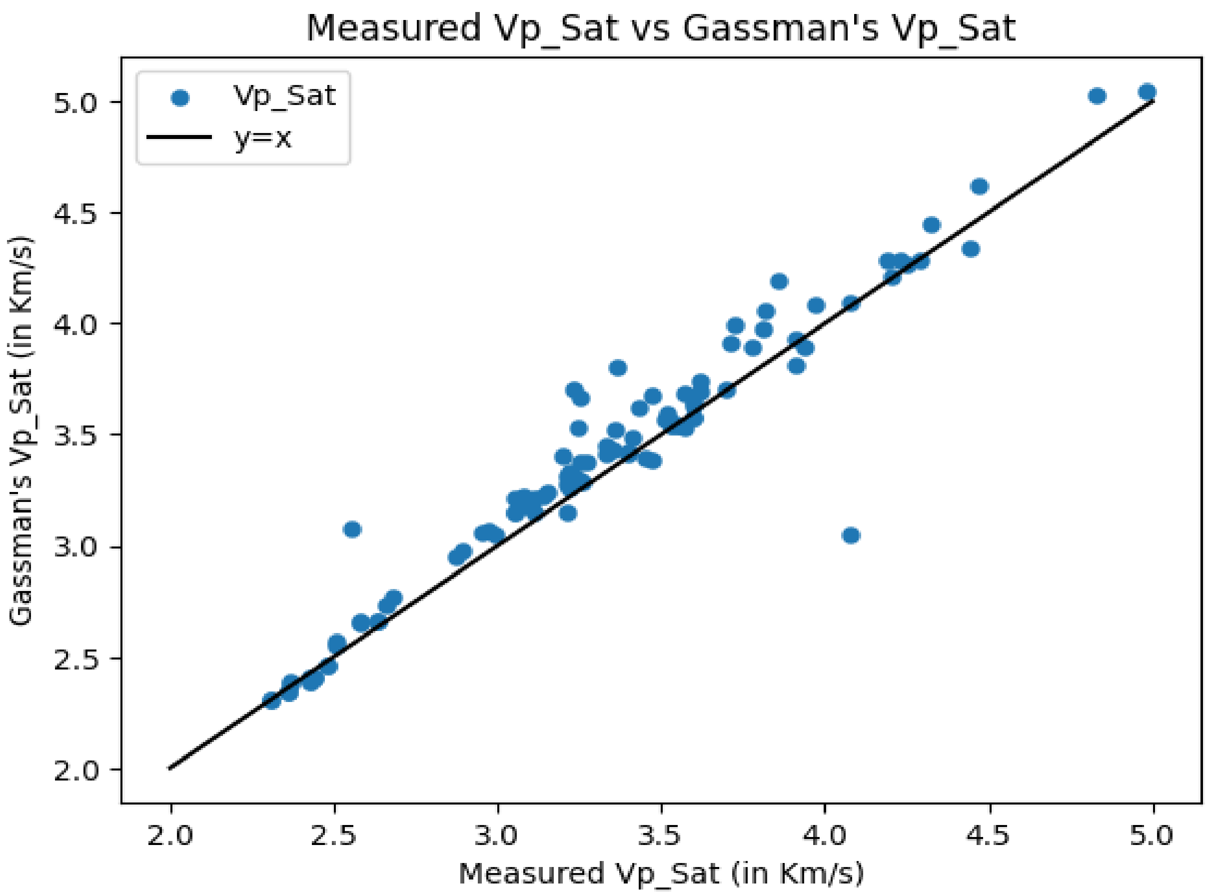
13) Make a cross plot between measured saturated and Gassmann saturated values of compressive- and shear-wave velocities.

For Vp Saturated: -

Code snippet: -

```
# Measured Sat and Gassman's Sat Vp and Vs
plt.scatter(df["Wet Ver. Vp"].iloc[1:],Vp_Sat_Gassman,label='Vp_Sat',linewidths=0.1)
plt.xlabel("Measured Vp_Sat (in Km/s)")
plt.ylabel("Gassman's Vp_Sat (in Km/s)")
plt.title("Measured Vp_Sat vs Gassman's Vp_Sat")

x=np.arange(2,6,1)
plt.plot(x,x,label='y=x',color='black')
plt.legend(loc='best')
```

Plot: -



Comments:

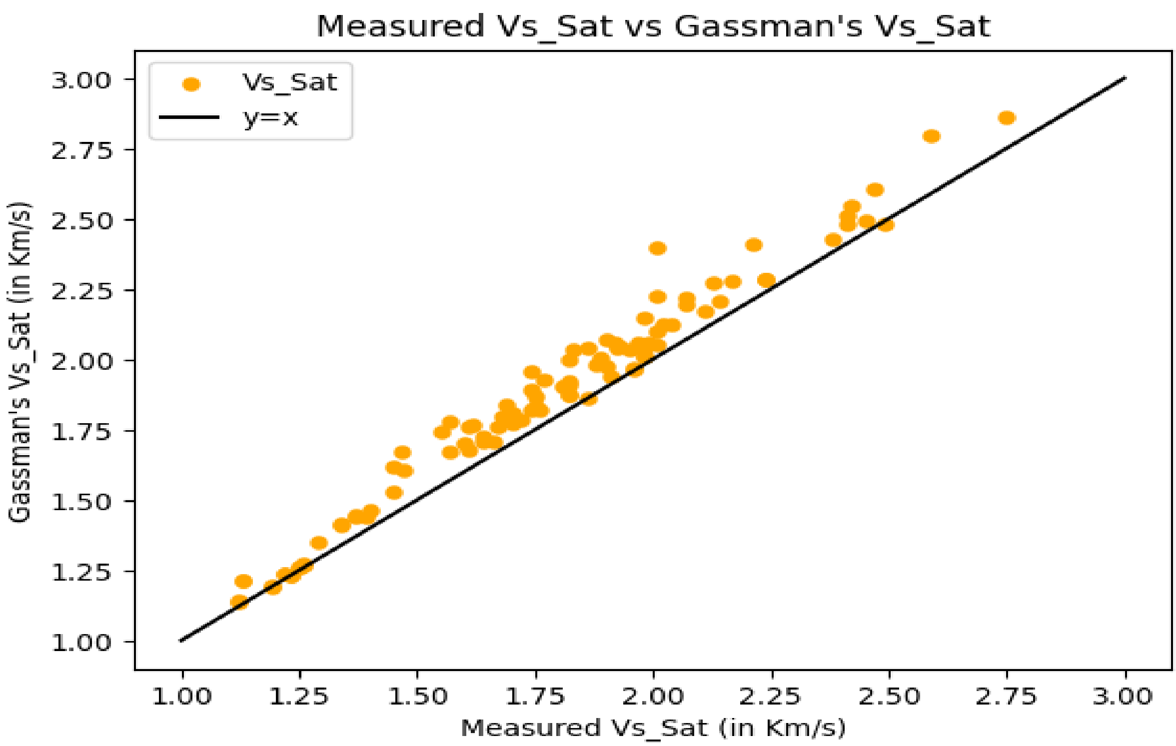
- Good correlation between the values of Measured Vp_Sat and Gassmann's Predicted Vp_Sat.
- Showing the applicability and accuracy of Gassmann's Point Prediction Method.

For Vs Saturated: -

code snippet: -

```
# Measured Sat and Gassman's Sat Vs
plt.scatter(df["Wet Ver. Vs"].iloc[1:],Vs_Sat_Gassman,color='orange',label='Vs_Sat',linewidths=0.1)
plt.xlabel("Measured Vs_Sat (in Km/s)")
plt.ylabel("Gassman's Vs_Sat (in Km/s)")
plt.title("Measured Vs_Sat vs Gassman's Vs_Sat")

y=np.arange(1,4,1)
plt.plot(y,y,label='y=x',color='black')
plt.legend(loc='best')
```

Plot: -



Comments:

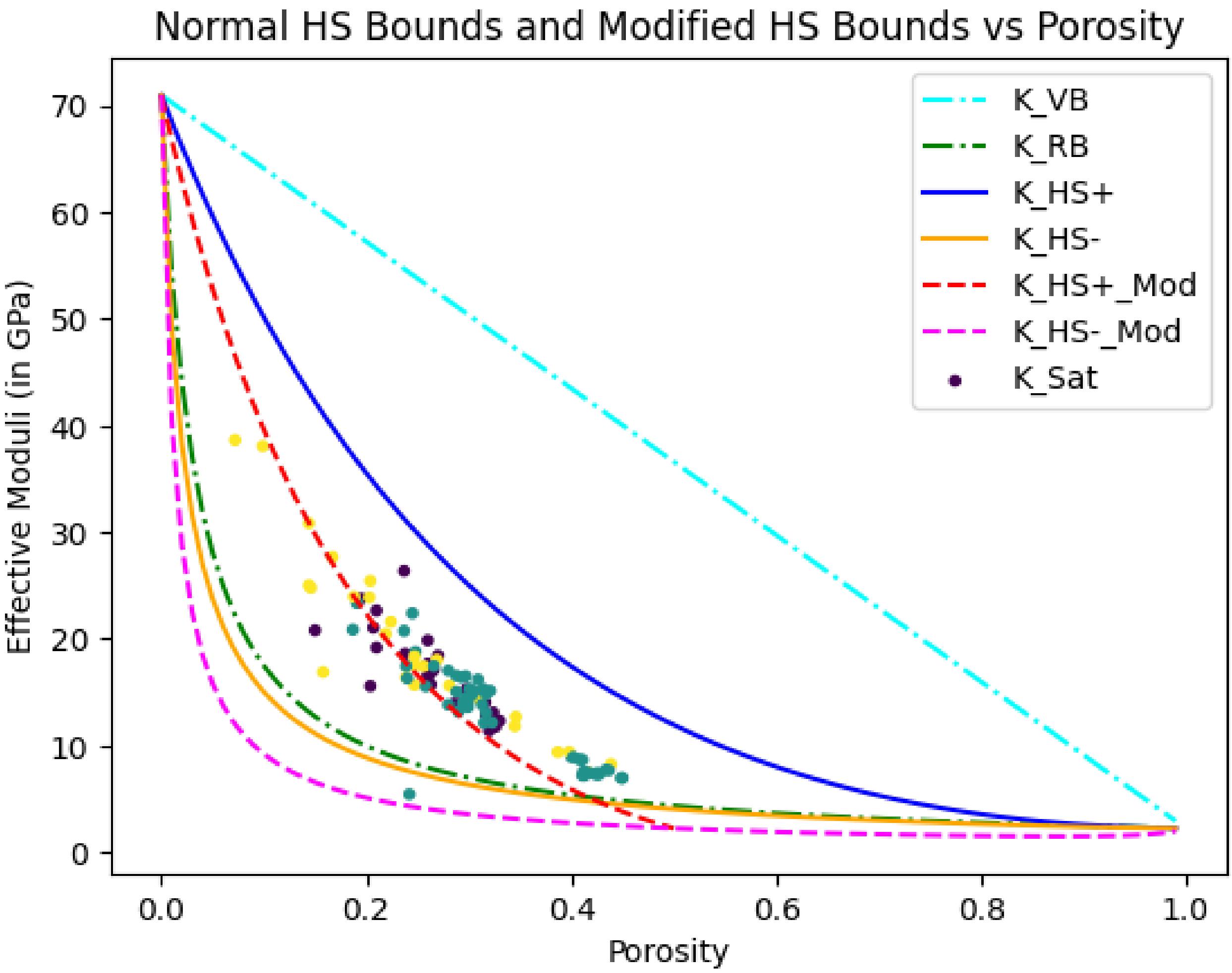
- Good correlation between the values of Measured Vp_Sat and Gassmann's Predicted Vp_Sat.
- Showing the applicability and accuracy of Gassmann's Point Prediction Method.

14) Plot the given core data on the bounds template for bulk and shear moduli and comment on the condition of the reservoir formations/samples.

Code snippet: -

```
# Modified Hassian-Shritkman and Core Data
por_critical=0.50  #critical porosity = 50% is asked
por_new=por/por_critical
K_HS_U_Mod=K_VB+por_new/((2.25-K_VB)**(-1)+(1-por_new)*(K_VB+(4/3)*G_VB)**(-1))
K_HS_L_Mod=2.25+(1-por_new)/((K_VRHB-2.25)**(-1)+por_new*(2.25)**(-1))
plt.plot(por,K_VB,color='cyan',linestyle='-.',label='K_VB')
plt.plot(por,K_RB,color='green',linestyle='-.',label='K_RB')
plt.plot(por,K_HS_U,color='blue',linestyle='-',label='K_HS+')
plt.plot(por,K_HS_L,color='orange',linestyle='-',label='K_HS-')
plt.plot(por[por<=0.5],K_HS_U_Mod[por<=0.5],color='red',linestyle='--',label='K_HS+_Mod')
plt.plot(por,K_HS_L_Mod,color='magenta',linestyle='--',label='K_HS-_Mod')
color_bar=df["Formation"].iloc[1:]
color_bar[color_bar=="chalk"]=1
color_bar[color_bar=="Tor"]=2
color_bar[color_bar=="Ekofisk"]=3
plt.scatter(Porosity,K_Sat,c=color_bar,cmap='viridis',marker='.',linewidths=1,label='K_Sat')
plt.xlabel("Porosity")
plt.ylabel("Effective Moduli (in GPa)")
plt.title("All Physical Effective Model Bounds vs Porosity ")
plt.title("Normal HS Bounds and Modified HS Bounds vs Porosity")
plt.legend(loc='best')
```

Plot: -



Comments:

- All Gassmann's predicted Ksat point values completely enclosed within the Hashin-Shtrikman's Bounds showing the applicability of this bound by removing the level of uncertainty in case of Voigt-Reuss Bounds.
- Nearly all Gassmann's predicted Ksat point values are good fitted over the Modified K_HS+ bound possibly indicating that the critical porosity of the formations is close to 50%.
- The best fitted Modified K_HS+ bound was observed for the critical porosity value equal to 60%, which is the usual critical porosity value for the carbonates (chalks).

END

Thank You