

# ?gua no solo

May 14, 2019

## 1 Programa de Pós-graduação em Recursos Hídricos e Sanamento - PP-GRHS

### 1.1 Hidrologia

Água no Solo  
Clebson Farias

### 1.2 Questão 1: Em um ensaio granulométrico, uma amostra de solo apresentou as seguintes distribuições: 25% de areia, 65% de silte e 10% de argila. Qual a textura de solo que esta amostra representa?

```
[1]: def textura_solo(p_areia, p_argila, p_silte):  
    if 60 <= p_argila <= 100 and 0 <= p_silte <= 40:  
        if 0 <= p_areia <= 40:  
            return "Muito Argilosa"  
  
    elif 35 <= p_argila <= 100 and 0 <= p_silte <= 65:  
        if 0 <= p_areia <= 65:  
            return "Argilosa"  
  
    elif 0 <= p_argila <= 85 and 0 <= p_silte <= 100:  
        if 70 <= p_areia <= 100:  
            if 0 <= p_argila <= 20:  
                return "Arenosa"  
            elif p_argila > 20:  
                return "Média"  
        elif 0 <= p_areia <= 15:  
            return "Siltosa"  
        else:  
            return "Média"  
  
    elif (0 <= p_argila <= 35 or 85 <= p_argila <= 100) and 50 <= p_silte <= 100:  
        if 0 <= p_areia <= 50:  
            return "Siltosa"
```

```

p_areia = 25
p_argila = 10
p_silte = 65
print("Textura do solo: ", textura_solo(p_areia=p_areia, p_silte=p_silte,
→p_argila=p_argila))

```

Textura do solo: Média

**1.2.1 Questão 2.** Calcule a taxa de infiltração (cm/hr) e o total infiltrado (cm) em intervalos horários (para cada hora) por um período de 5 horas para as seguintes texturas de solo:

(a) Areia;

(b) Silte;

(c) Argila.

**1.2.2** Mostre um gráfico com três curvas de taxa de infiltração e infiltração acumulada. Em cada caso, assuma que a saturação efetiva do solo é de 20% e que o solo fica encharcado instantaneamente.

Classe	$\eta$	$\theta_e$	$\psi$ [cm]	K [cm/hr]
Areia	0.437	0.417	4.95	11.89
Silte	0.501	0.486	16.68	0.65
Argila	0.475	0.385	31.63	0.03

```

[2]: import pandas as pd

classe = {"Areia": {"porosity": 0.437,
                    "e_porosity": 0.417,
                    "t_matricial": 4.95,
                    "c_hidraulica": 11.89
                  },
          "Silte": {"porosity": 0.501,
                    "e_porosity": 0.486,
                    "t_matricial": 16.68,
                    "c_hidraulica": 0.65
                  },
          "Argila": {"porosity": 0.475,
                     "e_porosity": 0.385,
                     "t_matricial": 31.63,
                     "c_hidraulica": 0.03
                   }

```

```

    }
}
df = pd.DataFrame(classe)
df

```

```

[2]:
      Areia  Silte  Argila
c_hidraulica  11.890  0.650  0.030
e_porosity    0.417  0.486  0.385
porosity      0.437  0.501  0.475
t_matricial   4.950  16.680  31.630

```

- Saturação efetiva do solo:  $s_e = 0.2$
- Taxa de infiltração (cm/h):  $f$
- Total infiltrado (cm):  $F$
- Tempo (h): 1:1:5

$$\Delta\theta = \theta_e(1 - s_e) \quad (1)$$

```

[3]: def v_umidade(e_porosity, e_saturation):
      return round(e_porosity*(1-e_saturation), 4)

for i in df:
    e_saturation = 0.2
    e_porosity = df.at['e_porosity', i]
    df.at['v_umidade', i] = v_umidade(e_porosity, e_saturation)
df

```

```

[3]:
      Areia  Silte  Argila
c_hidraulica  11.8900  0.6500  0.030
e_porosity    0.4170  0.4860  0.385
porosity      0.4370  0.5010  0.475
t_matricial   4.9500  16.6800  31.630
v_umidade     0.3336  0.3888  0.308

```

Variação de Umidade no Solo:  $\Delta\theta$

```

[4]: import math

#Cálculo da infiltração acumulada
def infiltracao_acumulada(erro, c_hidraulica, t_matricial, v_umidade, tempo):
    f = 1
    e = 1
    while e >= erro:
        r = ((c_hidraulica * tempo) + t_matricial*v_umidade*math.log(1 + (f/
→(t_matricial*v_umidade)))) - f)
        e = abs(r)
        f = f+r
    return round(f,4)

```

$$F(t) = kt + \psi \Delta \theta \ln(1 + \frac{F(t)}{\psi \Delta \theta}) \quad (2)$$

```
[5]: #Cálculo taxa de infiltração
def taxa_infiltracao(c_hidraulica, t_matricial, v_umidade, inf_acum):
    return round(c_hidraulica*(1+(t_matricial*v_umidade)/inf_acum), 4)
```

$$f = k(1 + \frac{\psi \Delta \theta}{F(t)}) \quad (3)$$

```
[6]: inf_acum_df = pd.DataFrame()

for i in df:
    c_hidraulica = df.at['c_hidraulica', i]
    t_matricial = df.at['t_matricial', i]
    v_umidade = df.at['v_umidade', i]
    for time in range(1,6):
        inf_acum_df.at[time, i] = infiltracao_acumulada(0.000001, c_hidraulica,
        →t_matricial, v_umidade, time)

inf_acum_df
```

```
[6]:      Areia  Silte  Argila
1  15.7818  3.3521  0.7847
2  28.5809  5.0150  1.1216
3  41.0408  6.4049  1.3849
4  53.3491  7.6554  1.6101
5  65.5705  8.8176  1.8110
```

Infiltração acumulada  $F$

```
[7]: inf_df = pd.DataFrame()

for i in df:
    c_hidraulica = df.at['c_hidraulica', i]
    t_matricial = df.at['t_matricial', i]
    v_umidade = df.at['v_umidade', i]
    for inf_acum in inf_acum_df[i].index:
        inf_df.at[inf_acum, i] = taxa_infiltracao(c_hidraulica, t_matricial,
        →v_umidade, inf_acum_df.at[inf_acum, i])

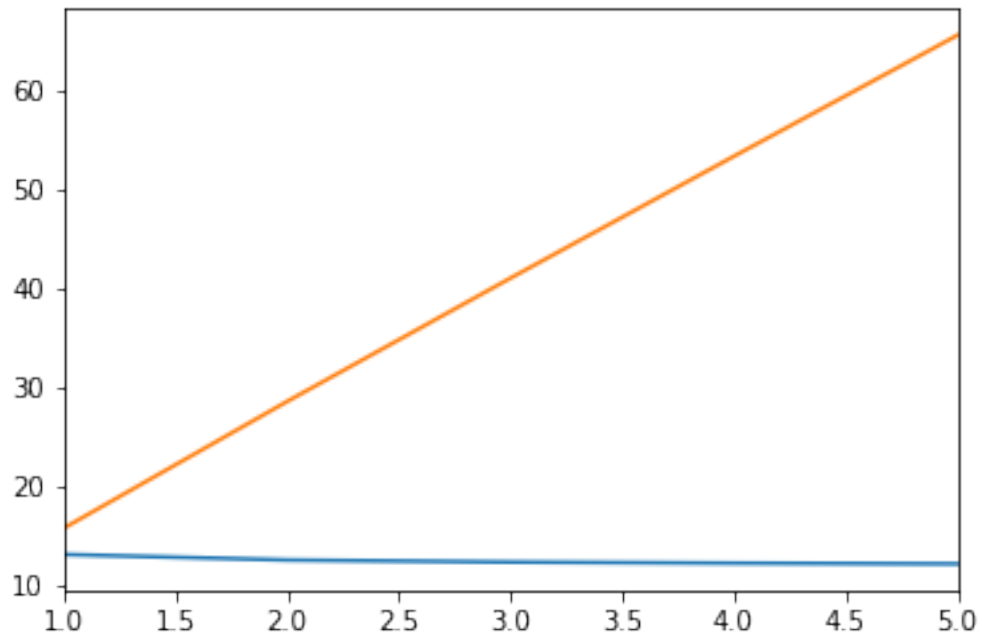
inf_df
```

```
[7]:      Areia  Silte  Argila
1  13.1341  1.9075  0.4024
2  12.5770  1.4906  0.2906
3  12.3684  1.3081  0.2410
4  12.2580  1.2006  0.2115
5  12.1894  1.1281  0.1914
```

Taxa de Infiltração  $f$

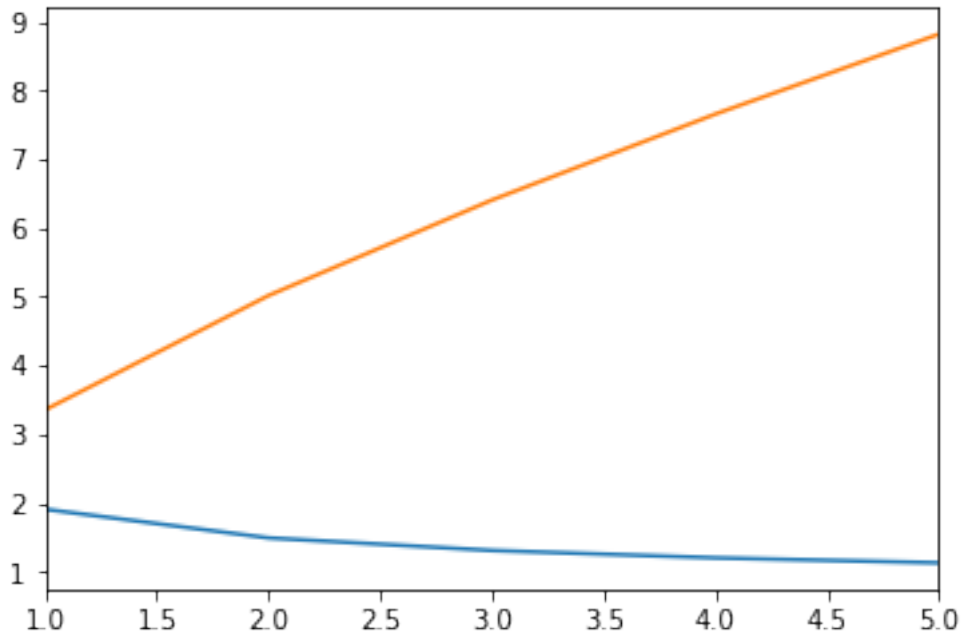
```
[12]: inf_df['Areia'].plot()  
      inf_acum_df['Areia'].plot()
```

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f299142cd30>
```



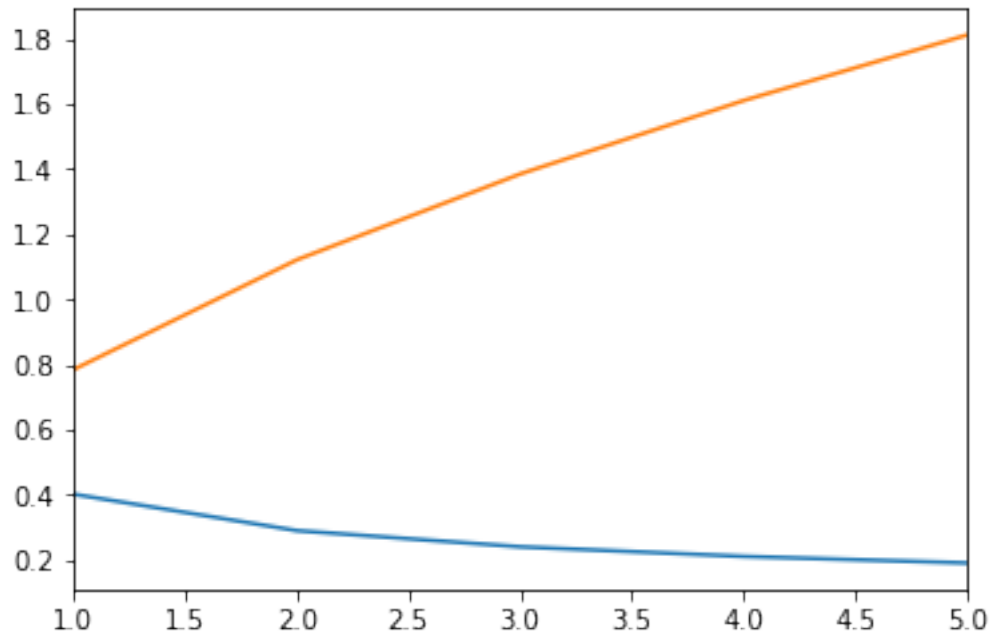
```
[9]: inf_df['Silte'].plot()  
      inf_acum_df['Silte'].plot()
```

```
[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2991549278>
```



```
[10]: inf_df['Argila'].plot()  
      inf_acum_df['Argila'].plot()
```

```
[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f299149c5f8>
```



**1.2.3 Questão 3. Determine quanto tempo leva até que os três tipos de solo fiquem encharcados quando submetidos a uma chuva de intensidade constante e igual a**

- (a) 1 cm/hr;
- (b) 10 cm/hr. ### Mostre estes resultados em uma tabela (incluindo os que não encharcaram também, se houver).

$$t_e = \frac{k\psi\Delta\theta}{i(i-k)} \quad (4)$$

```
[11]: def tempo_encha(c_hidraulica, t_matricial, v_umidade, i):  
        return (c_hidraulica * t_matricial * v_umidade)/(i*(i-c_hidraulica))  
  
df_tempo = pd.DataFrame()  
for i in df:  
    c_hidraulica = df.at['c_hidraulica', i]  
    t_matricial = df.at['t_matricial', i]  
    v_umidade = df.at['v_umidade', i]  
    for j in [1, 10]:  
        df_tempo.at[j, i] = tempo_encha(c_hidraulica, t_matricial, v_umidade, j)  
  
df_tempo
```

```
[11]:      Areia      Silte      Argila  
1  -1.802956  12.043913  0.301300  
10 -1.038846   0.045084  0.002931
```