TEA018 - Hidrologia Ambiental

Método de Newton-Raphson

Modelo de Green-Ampt para cálculo de infiltração

Junho de 2023

Emílio Graciliano Ferreira Mercuri
Departamento de Engenharia Ambiental
Universidade Federal do Paraná (UFPR)

Outline

Agenda de atividades

- Método de Newton-Raphson
- Solução do Modelo de Green-Ampt
    - Método de sucessivas substituições
    - Método de Newton-Raphson

Método de Newton-Raphson

Método de Newton-Raphson

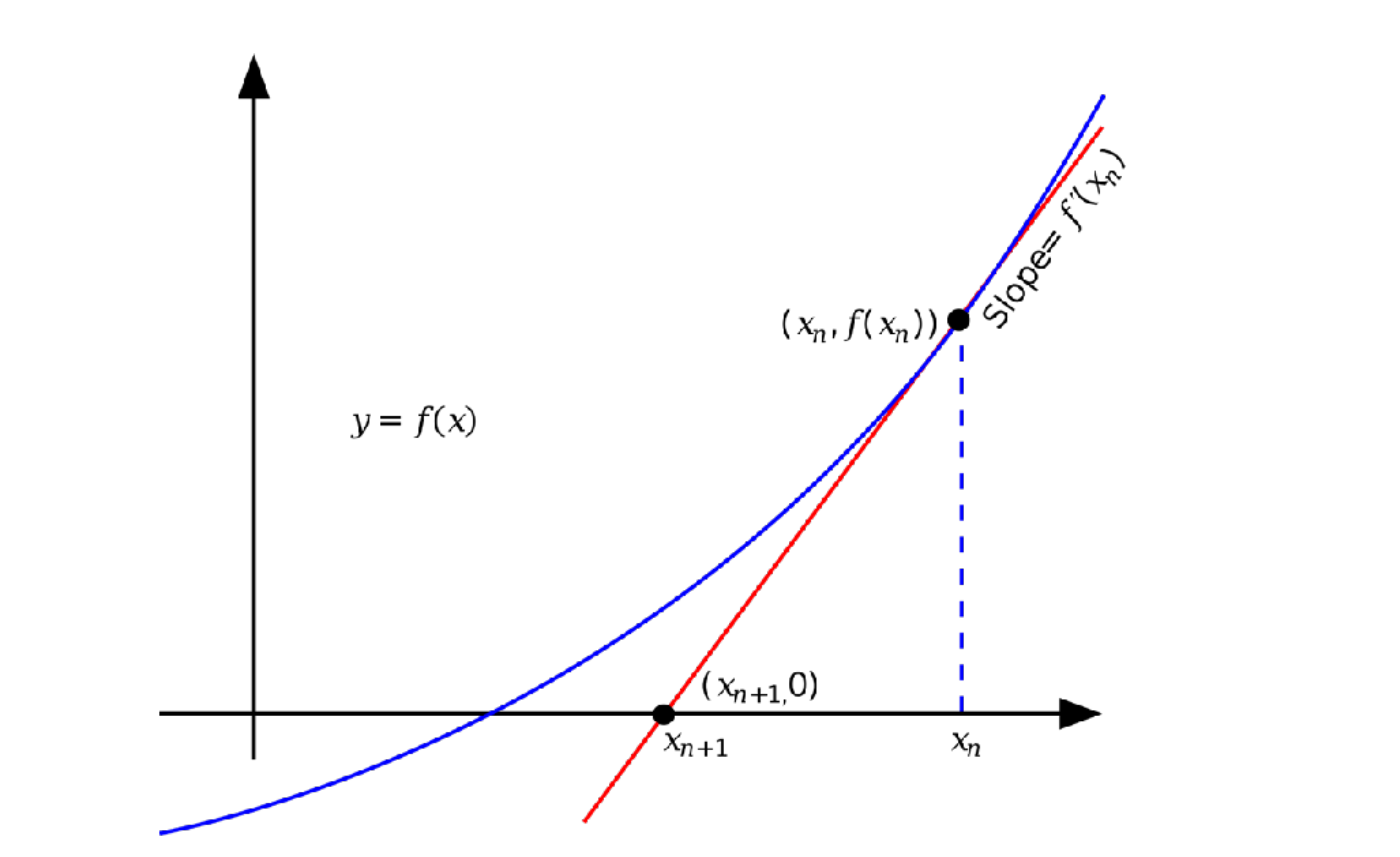
Objetivo: calcular o zero (as raízes) de funções reais de uma variável real.

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n+1})}{x_n - x_{n+1}}$$

$$f'(x_n) = \frac{f(x_n) - 0}{x_n - x_{n+1}}$$

Resolvendo para x_{n+1}:

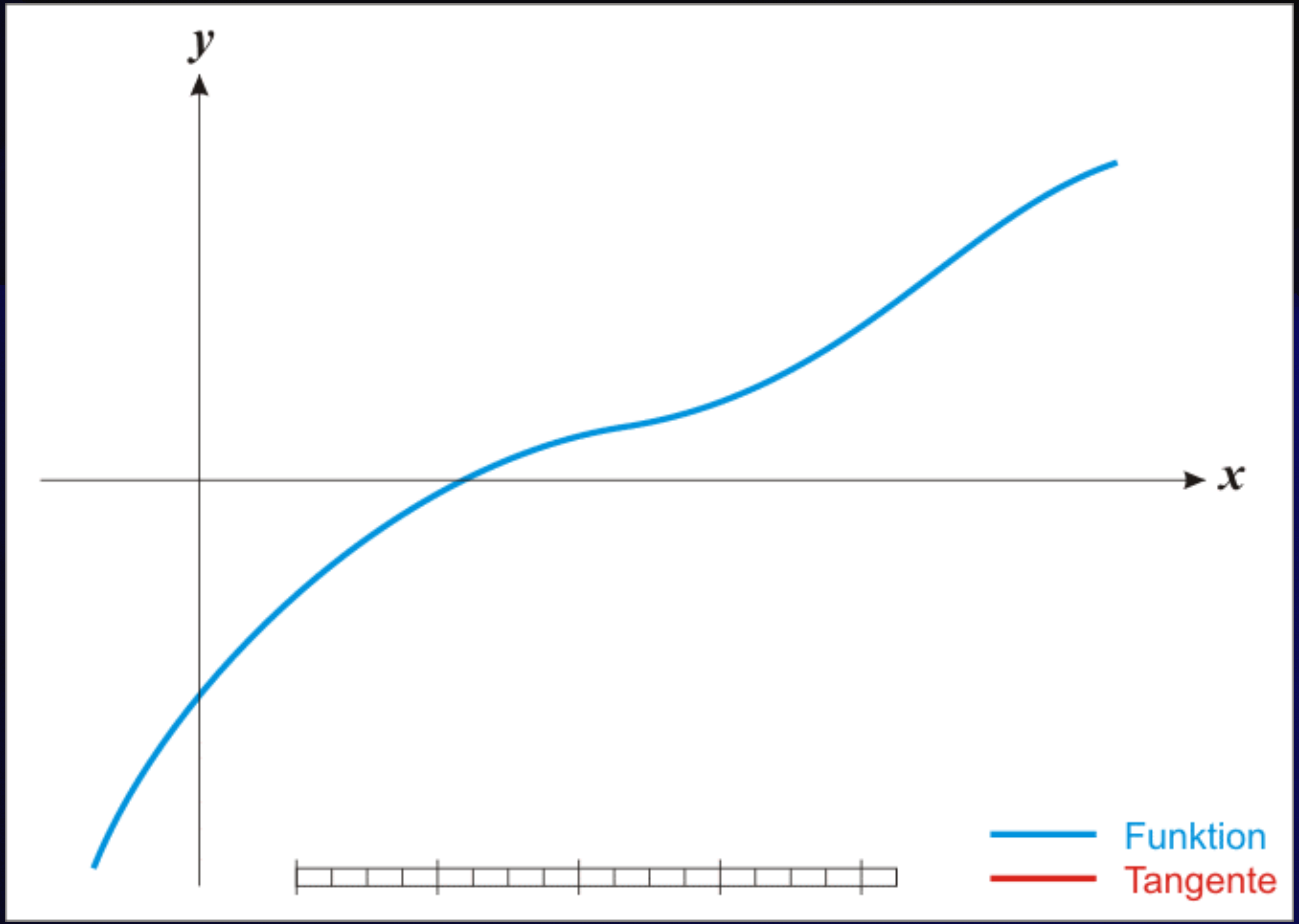
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



Método de Newton-Raphson

Objetivo: calcular o zero (as raízes) de funções reais de uma variável real.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



Modelo de Green-Ampt

Problema: Calcule a Infiltração acumulada $F(t)$ e a taxa de infiltração $f(t)$ após 1 hora de infiltração no solo mencionado.

Comparação Métodos: Sucessivas Substituições e Newton-Raphson

Tabela dos parâmetros de infiltração do modelo Green-Ampt para diferentes solos

Classe do solo	Porosidade η	Porosidade efetiva θ_e	Carga de sucção na frente de molhamento ψ (cm)	Condutividade Hidráulica Saturada K (cm/h)
Areia	0.437 (0.374 – 0.500)	0.417 (0.354 – 0.480)	4.95 (0.97 – 25.36)	11.78
Areia argilosa	0.437 (0.363 – 0.506)	0.401 (0.329 – 0.473)	6.13 (1.35 – 27.94)	2.99
Franco arenoso	0.453 (0.351 – 0.555)	0.412 (0.283 – 0.541)	11.01 (2.67 – 45.47)	1.09
Franco	0.463 (0.375 – 0.551)	0.434 (0.334 – 0.534)	8.89 (1.33 – 59.38)	0.34
Franco siltoso	0.501 (0.420 – 0.582)	0.486 (0.394 – 0.578)	16.68 (2.92 – 95.39)	0.65
Argila siltosa	0.479 (0.425 – 0.533)	0.423 (0.334 – 0.512)	29.22 (6.13 – 139.4)	0.05
Argila	0.475 (0.427 – 0.523)	0.385 (0.269 – 0.501)	31.63 (6.39 – 156.5)	0.03

Os números entre parênteses abaixo de cada parâmetro são um desvio padrão em torno do valor da média. **Fonte**: Rawls, Brakensiek, and Miller, 1983.

Franco Siltoso com saturação efetiva de 30% ($s_e = 30\,\%$)

Das informações dadas (tipo do solo) e da tabela:

- $s_e = 30\,\%$
- $\eta = 0.501$
- $\theta_e = 0.486$
- $\psi = 16.68$ cm
- $K = 0.65$ cm/h

Resolver iterativamente o **modelo de Green-Ampt** para a Infiltração Acumulada $F(t)$ utilizando os dois métodos citados.

$$F = Kt + \psi\Delta\theta \ln\left(1 + \frac{F}{\psi\Delta\theta}\right)$$

Já conhecemos:

$$\Delta\theta = \left(1 - s_e\right)\theta_e = (1 - 0.3)(0.486) = 0.3402$$

Portanto: $\psi\Delta\theta = 5.674536$

Para t = 1 hora a equação fica:

$$F = 0.65 + 5.674536 \ln\left(1 + \frac{F}{5.674536}\right)$$

Equações estudadas em aula:

$$\Delta\theta = \left(1 - s_e\right)\theta_e \qquad s_e = \frac{\theta - \theta_r}{\eta - \theta_r} \qquad \Delta\theta = \eta - \theta_i$$

Modelo de Green-Ampt

$$f = K\left(\frac{\psi\Delta\theta}{F} + 1\right) \qquad F - \psi\Delta\theta \ln\left(1 + \frac{F}{\psi\Delta\theta}\right) = Kt$$

Modelo de Green-Ampt

Problema: Calcule a Infiltração acumulada $F(t)$ e a taxa de infiltração $f(t)$ após 1 hora de infiltração no solo mencionado.

Método: Sucessivas Substituições

$$F = 0.65 + 5.674536 \ln\left(1 + \frac{F}{5.674536}\right)$$

Google Colaboratory: https://colab.research.google.com/drive/1pF0LoKkLDPcudpyl0Pd9O6r6J9c36Ckx?usp=sharing

```python
import numpy
```

```python
## Definição de função para cálculo de F
def funcao(F):
  return 0.65 + 5.674536*numpy.log(1+F/5.674536)
```

```python
tolerancia = 10**(-6)
erro = 100

# chute inicial: F = K*t
F = 0.65

i = 0
while erro > tolerancia:
  F_novo = funcao(F)
  erro = abs(F_novo-F)
  print(i, F_novo)
  F = F_novo
  i += 1
```

34 iterações:

```
0 1.2653913670011807
1 1.7922983438787963
2 2.2075598561393894
3 2.5146811416698487
4 2.7315872506443513
5 2.879931495648853
6 2.9791976400493545
7 3.044665907612381
8 3.0874340165127876
9 3.1151998503900744
10 3.1331534734206925
11 3.144732271306316
12 3.1521872520058225
13 3.156981945329756
14 3.160063527491899
15 3.1620431983641546
16 3.163314614816225
17 3.164131014385203
18 3.1646551774409613
19 3.164991686773525
20 3.1652077130972502
21 3.1653463895306007
22 3.1654354100120115
23 3.1654925541413963
24 3.1655292358631204
25 3.1655527823163165
26 3.165567897025859
27 3.1655775992915514
28 3.1655838272529397
29 3.165587825027206
30 3.1655903912264547
31 3.1655920384870813
32 3.1655930958745486
33 3.165593774618438
```

```python
# Taxa de infiltração
f = K*(psi*delta_theta/F + 1)
print('F = ',F, 'cm')
print('f = ',f, 'cm/h')
```

```
F =  3.165593774618438 cm
f =  1.8151679471869644 cm/h
```

Modelo de Green-Ampt

Problema: Calcule a Infiltração acumulada $F(t)$ e a taxa de infiltração $f(t)$ após 1 hora de infiltração no solo mencionado.

Método: Newton-Raphson

$$F = 0.65 + 5.674536 \ln\left(1 + \frac{F}{5.674536}\right)$$

Google Colaboratory: https://colab.research.google.com/drive/1M6BbQTrOGVqAVfRjqC6LLsFC2hd-la-X?usp=sharing

```
import numpy
```

```
## Definição de função para cálculo de F
def funcao(F):
  return 0.65 + 5.674536*numpy.log(1+F/5.674536) - F
```

```
## Definição de função para cálculo de F
def derivada(F):
  return 5.674536/(5.674536+F) - 1
```

```
tolerancia = 10**(-6)
erro = 100

# chute inicial: F = K*t
F = 0.65

i = 0
while erro > tolerancia:
  F_novo = F - funcao(F)/derivada(F)
  erro = abs(F_novo-F)
  print(i, F_novo)
  F = F_novo
  i += 1
```

6 iterações:

```
0 6.637792084135656
1 3.684397178748581
2 3.188595866709142
3 3.1656481965726386
4 3.1655949915986064
5 3.1655949913116053
```

```
# Taxa de infiltração
f = K*(psi*delta_theta/F + 1)
print('F = ',F, 'cm')
print('f = ',f, 'cm/h')
```

```
F =  3.1655949913116053 cm
f =  1.8151674993558666 cm/h
```

Modelo de Green-Ampt

Explicação do método de Newton-Raphson

$$F = 0.65 + 5.674536 \ln\left(1 + \frac{F}{5.674536}\right)$$

A função a ser encontrada as raízes é:

$$g = 0.65 + 5.674536 \ln\left(1 + \frac{F}{5.674536}\right) - F$$

$$g = 0.65 + 5.674536 \ln\left(\frac{5.674536 + F}{5.674536}\right) - F$$

Derivada do $\ln x$:

$$g(x) = \ln x$$

$$\frac{dg}{dx} = \frac{1}{x}$$

Portanto, usando a regra da cadeia:

$$\frac{dg}{dF} = 5.674536 \frac{5.674536}{5.674536 + F} \frac{d}{dF}\left(\frac{5.674536 + F}{5.674536}\right) - 1$$

$$\frac{dg}{dF} = \frac{5.674536}{5.674536 + F} - 1$$

Equação do método:

$$F_{n+1} = F_n - \frac{g(x_n)}{g'(x_n)}$$

```python
import numpy
```

```python
## Definição de função para cálculo de F
def funcao(F):
  return 0.65 + 5.674536*numpy.log(1+F/5.674536) - F
```

```python
## Definição de função para cálculo de F
def derivada(F):
  return 5.674536/(5.674536+F) - 1
```

```python
tolerancia = 10**(-6)
erro = 100

# chute inicial: F = K*t
F = 0.65

i = 0
while erro > tolerancia:
  F_novo = F - funcao(F)/derivada(F)
  erro = abs(F_novo-F)
  print(i, F_novo)
  F = F_novo
  i += 1
```

Obrigado

Final de atividades

- Praticar exercícios
- Refazer os exemplos no Google Colaboratory
- Refazer usando a calculadora