Script of analytical calculation of depth of freshwater-seawater interface

December 19, 2020

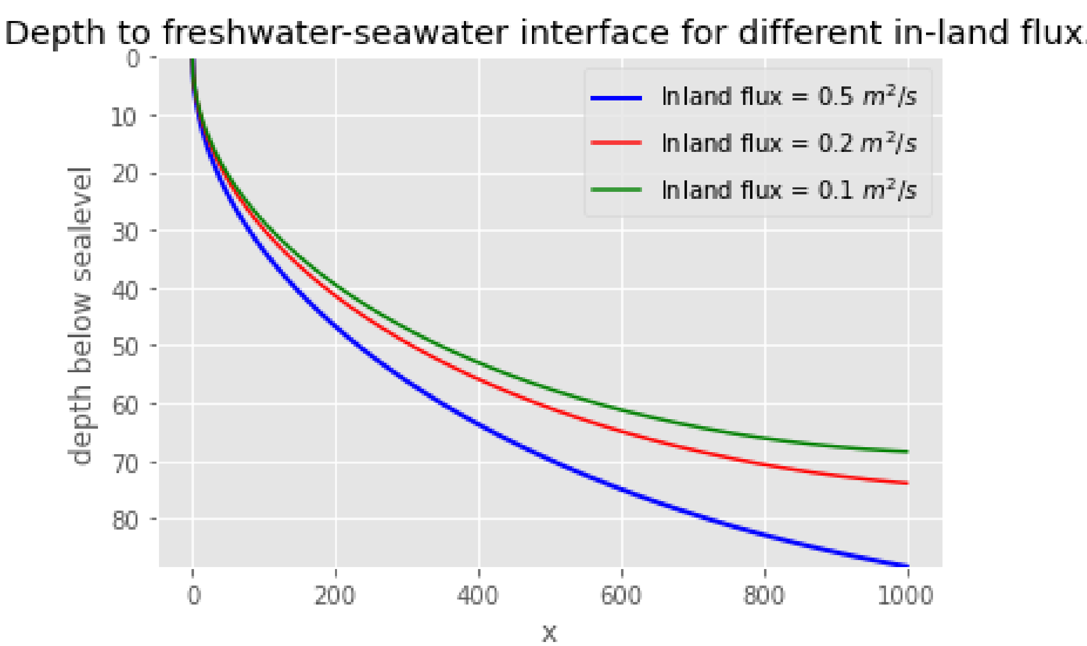
1 Unconfined steady state flow with flux boundary condition

[1]:
```python
import numpy as np

# Depth to freshwater-seawater interface for different inland flux with 0.5, 0.
↪2 and 0.1 $m^2/s$ and a recharge rate of 100 mm/year

H  = 100 # m, aquifer thickness at coast
L  = 1000 # m, length of the studied area
a  = 0.1 * L
va = 1E-6
Dc  = va * a
e  = 40 #  density ratio of freshwater density relative to the difference␣
↪between seawater and freshwater density
es = e * (1 - a / H)**(1 / 6)
K  = 10 # m/d, hydraulic conductivity
Qf = -0.5 # sm per d, the groundwater flow from the basin towards the sea
Qf1 = -0.2 # sm per d, the groundwater flow from the basin towards the sea
Qf2 = -0.1 # sm per d, the groundwater flow from the basin towards the sea
N  = 0.001 # m, per d, recharge rate

# make the linear space of x valus from sea to L
x  = np.arange(0,L,1)

# calculate the depth of the fresh-water seawater interface
# for Qf = -0.5
hs = np.sqrt(((Qf * x - N * L * x + N * ((x**2)/2))*(2 * (e**2)))/(-(1 + e) *␣
↪K))
y  = hs

# calculate the depth of the fresh-water seawater interface
# for Qf1 = -0.2
hs1 = np.sqrt(((Qf1 * x - N * L * x + N * ((x**2)/2))*(2 * (e**2)))/(-(1 + e) *␣
↪K)) # the depth of the fresh-water seawater interface
y1  = hs1
```

```
# calculate the depth of the fresh-water seawater interface
# for Qf2 = -0.1
hs2 = np.sqrt(((Qf2 * x - N * L * x + N * ((x**2)/2))*(2 * (e**2)))/(-(1 + e) *
 ↪K)) # the depth of the fresh-water seawater interface
y2  = hs2
```

[2]:
```
# Plot the results

import matplotlib.pyplot as plt
plt.style.use('ggplot')

plt.plot(x,y, label = 'Inland flux = 0.5 $m^2/s$' , c = 'b' , linewidth = 2)
plt.plot(x,y1, label = 'Inland flux = 0.2 $m^2/s$' , c = 'r')
plt.plot(x,y2, label = 'Inland flux = 0.1 $m^2/s$' , c = 'g')

plt.ylim(y.max(), 0)
plt.title('Depth to freshwater-seawater interface for different in-land flux.')
plt.ylabel("depth below sealevel")
plt.xlabel('x')
plt.savefig('plot-unnamed-chunk-2-1.png')
plt.legend()
```

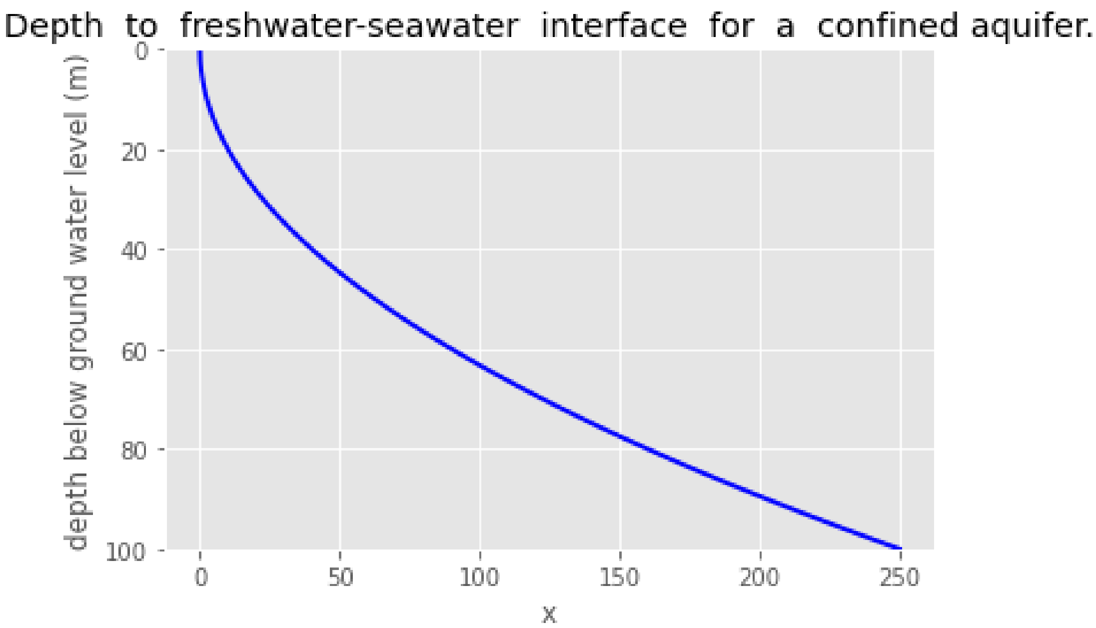
[2]: <matplotlib.legend.Legend at 0x7fe20d1f81f0>



2 Confined steady state flow with flux boundary condition

[3]:
```
# Depth to freshwater-seawater interface for a confined aquifer with a flux of␣
↪$0.5 m^2/d$,
# aquifer thickness of 100 m and a hydraulic conductivity of 1 m/d.

H  = 100 # m, aquifer thickness at coast
e  = 40 #
K  = 1 # m/d, hydraulic conductivity
L  = 100 # m
Qf = 0.5 # sm per d
h  = np.arange(0, L + 1, 1)
x =  K / (e * Qf) * ((h**2)/2)
y  = h
```

[4]:
```
# plot the results

import matplotlib.pyplot as plt
plt.style.use('ggplot')

plt.plot(x,y, linewidth = 2, c='b')

plt.ylim(y.max(), 0)

plt.ylabel("depth below ground water level (m)")
plt.xlabel('x')

plt.title('Depth  to  freshwater-seawater  interface  for  a  confined aquifer.
↪')

plt.savefig('plot-confined_K-1.png')
```



3 Pumped wells

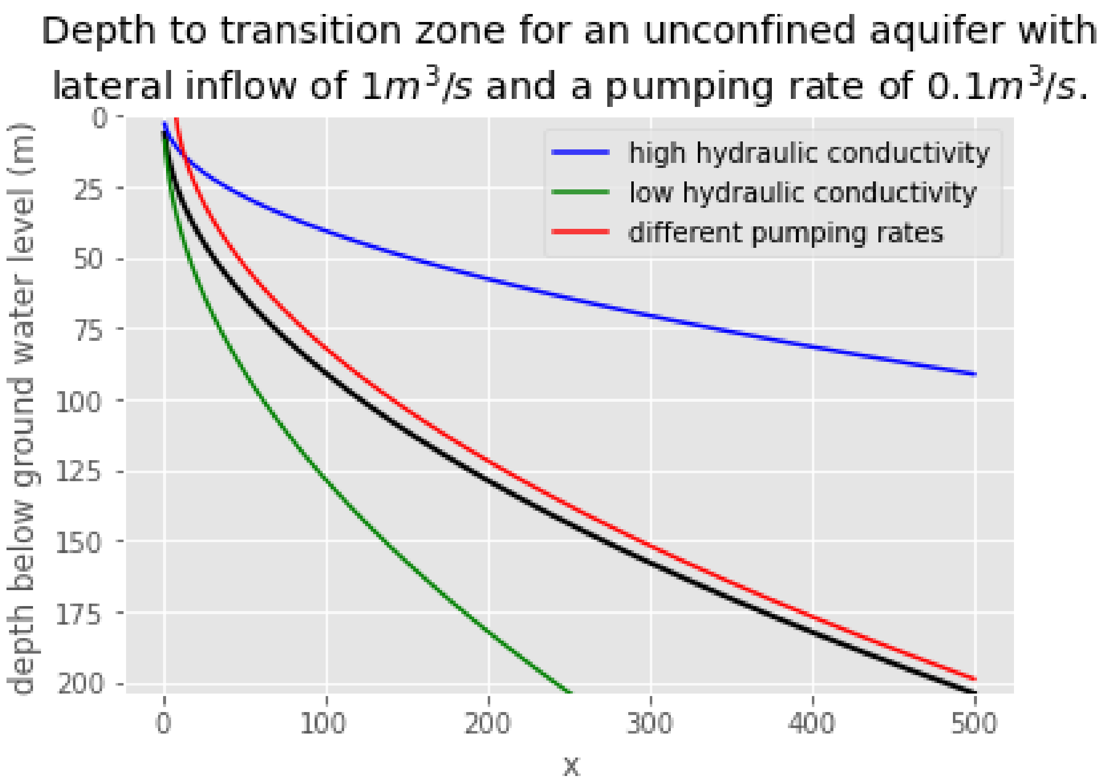
3.1 Unconfined aquifer

[5]:
```
# Depth to transition zone for an unconfined aquifer with lateral inflow of 1␣
↪$m^3/s$ and
# a pumping rate of 0.1 $m^3/s$ and scenarios for high hydraulic conductivity ,
# low hydraulic conductivity and different pumping rates  based on the␣
↪analytical solution
# of Strack1976.

# pumping_unconfined

rs = 1.024
rh = 1.000
e  = (rs - rh) / rh #
H  = 100 # aquifer thickness at coast
K  = 1 # m/s
K1 = 5
K2 = 0.5
Qw = 1 # qm/s
Qw2= 25
Qx0= 1 # qm/s
p  = np.pi
x  = np.arange(1, 501, 1)
```

```
# calculate for Qw = 1 qm/s, Qx0= 1 qm/s and K = 1
mu = Qw / (Qx0 * x)
lamda = 2*(1 - mu / p)**(1/2) + mu / p * np.log((1 - (1 - mu / p)**(1/2)) / (1␣
↪+ (1 - mu / p)**(1/2)))
y = np.sqrt((lamda * Qx0 * x)/(K * e))

# calculate for Qw = 1 qm/s, Qx0= 1 qm/s and K1 = 5
mu1 = Qw / (Qx0 * x)
lamda1 = 2*(1 - mu1 / p)**(1/2) + mu1 / p * np.log((1 - (1 - mu1 / p)**(1/2)) /␣
↪(1 + (1 - mu1 / p)**(1/2)))
y1 = np.sqrt((lamda1 * Qx0 * x)/(K1 * e))

# calculate for Qw = 1 qm/s, Qx0= 1 qm/s and K2 = 0.5
mu2 = Qw / (Qx0 * x)
lamda2 = 2*(1 - mu2 / p)**(1/2) + mu2 / p * np.log((1 - (1 - mu2 / p)**(1/2)) /␣
↪(1 + (1 - mu2 / p)**(1/2)))
y2 = np.sqrt((lamda2 * Qx0 * x)/(K2 * e))

# calculate for Qw2= 25 qm/s, Qx0= 1 qm/s and K = 1
mu3 = Qw2 / (Qx0 * x)
lamda3 = 2*(1 - mu3 / p)**(1/2) + mu3 / p * np.log((1 - (1 - mu3 / p)**(1/2)) /␣
↪(1 + (1 - mu3 / p)**(1/2)))
y3 = np.sqrt((lamda3 * Qx0 * x)/(K * e))
```

```
<ipython-input-5-2512d590fe55>:38: RuntimeWarning: invalid value encountered in
sqrt
  lamda3 = 2*(1 - mu3 / p)**(1/2) + mu3 / p * np.log((1 - (1 - mu3 / p)**(1/2))
/ (1 + (1 - mu3 / p)**(1/2)))
```

[6]:
```
import matplotlib.pyplot as plt
plt.style.use('ggplot')

plt.plot(x,y, linewidth = 2, c='black')
plt.plot(x,y1,  c = 'b', label = 'high hydraulic conductivity')
plt.plot(x,y2, c = 'g', label = 'low hydraulic conductivity')
plt.plot(x,y3,  c = 'r', label = 'different pumping rates')

plt.ylim(y.max(), 0)
plt.title('Depth to transition zone for an unconfined aquifer with\nlateral␣
↪inflow of $1 m^3/s$ and a pumping rate of $0.1 m^3/s$.')
plt.ylabel("depth below ground water level (m)")

plt.xlabel('x')
plt.legend()

plt.savefig('plot_pumping_unconfined.png')
```



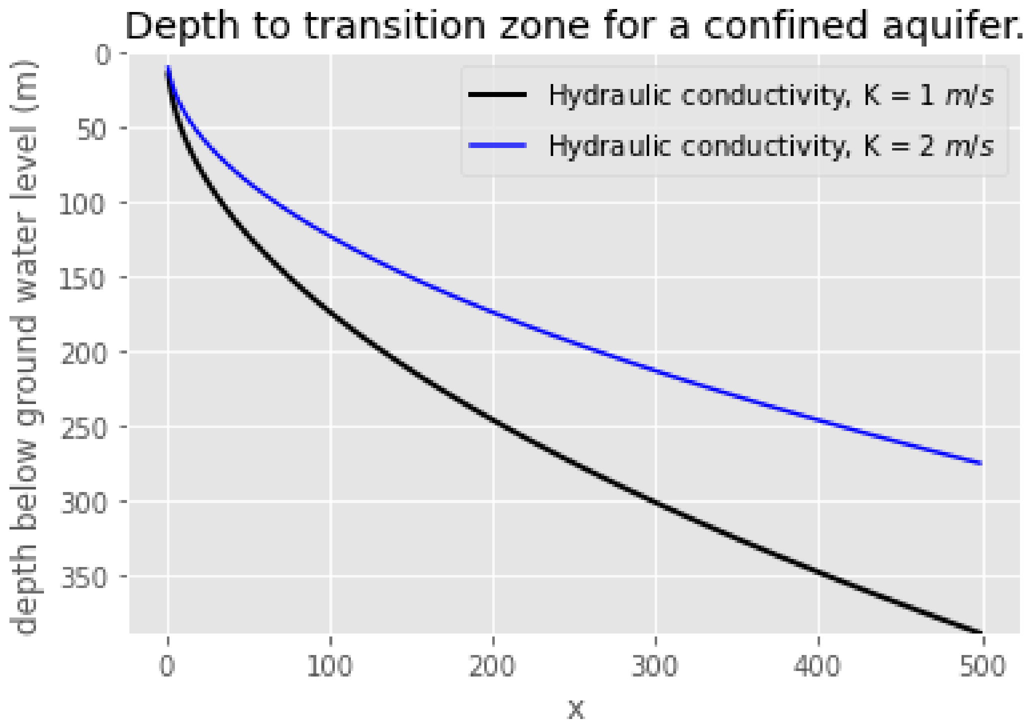
[7]:
```
## Confined aquifer
```

[8]:
```
# For the confined aquifer dispersion and the range of the transition zone
# has been implemented based on a method described in Beebe 2016.

# pumping_confined

rs = 1.024
rh = 1.000
e  = (rs / rh**2) * (rs - rh) #
a  = 0.1 * 10
H  = 100 # aquifer thickness at coast
K  = 1 # m/s
K1  = 2
Qw = 1 # qm/s
Qx0= 2 # qm/s
p  = np.pi
x  = np.arange(1,500,1)

# calculate for Qw = 1 qm/s, Qx0= 2 qm/s and K = 1
mu = Qw / (Qx0 * x)
```

```
lamda = 2 * (1 - mu / p)**(1/2) + mu / p * np.log((1 - (1 - mu / p)**(1/2))/(1␣
↪+ (1 - mu / p)**(1/2)))
es = e * (1 - (a/H)**(1/6))
y = np.sqrt((lamda * Qx0 * x) / (K * es))

# calculate for Qw = 1 qm/s, Qx0= 2 qm/s and K = 2
mu1 = Qw / (Qx0 * x)
lamda1 = 2 * (1 - mu1 / p)**(1/2) + mu1 / p * np.log((1 - (1 - mu1 / p)**(1/2))/
↪(1 + (1 - mu1 / p)**(1/2)))

y1 = np.sqrt((lamda1 * Qx0 * x) / (K1 * es))
```

[9]:

```
# Plot the results

import matplotlib.pyplot as plt
plt.style.use('ggplot')

plt.plot(x,y, linewidth = 2, c='black', label = 'Hydraulic conductivity, K = 1␣
↪$m/s$')
plt.plot(x,y1,  c = 'b', label = 'Hydraulic conductivity, K = 2 $m/s$')

plt.ylim(y.max(), 0)
plt.title('Depth to transition zone for a confined aquifer.')
plt.ylabel("depth below ground water level (m)")

plt.xlabel('x')
plt.legend()

plt.savefig('plot_pumping_confined.png')
```



Depth to transition zone for a confined aquifer.
Hydraulic conductivity, K = 1 m/s
Hydraulic conductivity, K = 2 m/s
depth below ground water level (m)
0
50
100
150
200
250
300
350
0
100
200
300
400
500
x