NTNU

Norwegian University of Science and Technology

Applied Computer Methods in Petroleum Science TPG4155

Semester Project Fall 2021

Problem α

Kiana Rezaei

Paniz Ataee

Reza Mozaffari

Department of Geoscience and Petroleum
Norwegian University of Science and Technology
Norway

Contents

Abstract

Reservoir simulation is one of the most significant tasks and needs for reservoir engineers because it helps them to figure out how a reservoir behave in reality. It also allows reservoir engineers to simulate oil and gas flow. This report is provided with information regarding simulation of oil production of a reservoir based on numerical approximation using finite difference methods. The coding in python is done with the help of methods such as stability, convergence, consistency and boundary conditions.

Introduction

In this project oil production is simulated in a three layered, inhomogeneous medium with different permeability. Starting from the surface, the first layer has the highest permeability, the second one has the lowest and the third layer's permeability is less than the first layer but higher than the second one. Also, these three layers are surrounded with impermeable rocks. One well is drilled through these layers and it is perforated in the first and second layer. In this condition, there are two possible ways to analyze the situation and the only difference between these 2 ways, is the permeability differences in second layer. In the first model, the permeability of the second layer is low, in the second model the permeability is still low but it is more than the first model. Due to low permeability of the second layer, it is assumed that there is no fluid flow from the third layer through the first layer. However, this assumption is not accurate when it comes to a real reservoir, and it is almost impossible to see the liquid parameters throughout the reservoir. Therefore, the solution is the drilled well which makes flow rate measurement available. As a result, a conclusion about the subsurface condition can be achieved. Reservoir modelling allows observing a whole picture of the subsurface, and therefore, connecting field observations and procedures in the reservoir.

Theory

Derivation of Equations

Darcy's equation law is the first equation that should be derived. We assume just a flow parallel to the x-axis. Hence, the differential equation is:

$$\boldsymbol{dP = -\frac{q}{A}\frac{\mu}{k}dx}$$

Q is the parameter which represents the flow rate in the porous media and A is the cross-section area, μ is the viscosity and k is the permeability of the porous medium. Integration of the both sides results:

$$\int_{P_1}^{P_2} dP = \int_0^L -\frac{q}{A}\frac{\mu}{k}dx \;\; \rightarrow \;\; (P_2 - P_1) = -\frac{q}{A}\frac{\mu}{k}L$$

We could solve and rearrange the flow rate to:

$$q = -\frac{k}{\mu}\frac{A}{L}(P_2 - P_1)$$

The ΔP is the pressure drop all over the pore volume.to achieve the volume of the flux we have to divide this equation to A, and we have for the x-direction:

$$q_x = \frac{q}{A} = -\frac{k}{\mu}\frac{(P_2 - P_1)}{L} = -\frac{k}{\mu}\frac{\partial P}{\partial x}$$

Then, we define this equation for a 2-dimensional (2D) and rewrite the equation as below:

$$q = -\frac{k}{\mu}\nabla P$$

The volumetric flux is represented by q, which is a vector, k is the permeability, μ is the viscosity and ∇P represents the pressure drop over the porous media length. the material balance equation is defined in order to compensate the physical understanding of the equation. Therefore, the general MBE is defined as below:

According to the mass balance equation and mass balance law we have:

$$\boldsymbol{M_{accumulated} = M_{in} - M_{out}}$$

Then, the continuity equation could be written as:

$$-\nabla.(\rho q) = \frac{\partial}{\partial t}(\varphi\rho)$$

The continuity equation for the left-hand side of the boundary can be written as:

$$\frac{\partial}{\partial t}(\rho\varphi) = \rho\frac{\partial\varphi}{\partial t} + \varphi\frac{\partial\rho}{\partial t}$$

Afterward, the total compressibility, which is composed of the fluid and the rock compressibility is defined as below:

$$c_t = c_\varphi + c_f = \frac{1}{\varphi}\frac{\partial\varphi}{\partial P} + \frac{1}{\rho}\frac{\partial\rho}{\partial P}$$

Which c_φ is the rock compressibility and c_f is the reservoir fluid compressibility. This equation could be expanded using the chain rule and the total compressibility:

$$\begin{aligned}\rho\frac{\partial\phi}{\partial t}+\phi\frac{\partial\rho}{\partial t}&=\rho\frac{\partial\phi}{\partial p}\frac{\partial p}{\partial t}+\phi\frac{\partial\rho}{\partial p}\frac{\partial p}{\partial t}\\&=\rho\phi[\frac{1}{\phi}\frac{\partial\phi}{\partial p}+\frac{1}{\rho}\frac{\partial\rho}{\partial p}]\frac{\partial p}{\partial t}\\&=\rho\phi[c_\phi+c_f]\frac{\partial p}{\partial t}\\&=\rho\phi c_t\frac{\partial p}{\partial t}\end{aligned}$$

Combination of these equations with Darcy's law and continuity equations gives:

$$-\boldsymbol{\nabla}.\left(\boldsymbol{\rho}\left(-\frac{\boldsymbol{k}}{\boldsymbol{\mu}}\boldsymbol{\nabla P}\right)\right)=\boldsymbol{\rho\varphi c_t}\frac{\boldsymbol{\partial P}}{\boldsymbol{\partial t}}$$

By rearrangement of this equation, the material balance equation is achieved:

$$\frac{\boldsymbol{\partial P}}{\boldsymbol{\partial t}}=-\frac{\boldsymbol{1}}{\boldsymbol{\rho\varphi c_t}}\boldsymbol{\nabla}.(\boldsymbol{\rho q})$$

We could assume constant permeability (k), density through the reservoir (ρ) and the viscosity (μ) in some cases, which then we could use the diffusivity equation in order to achieve the pressure change.

The diffusivity equation is:

$$\frac{\boldsymbol{\partial P}}{\boldsymbol{\partial t}}=\frac{\boldsymbol{k}}{\boldsymbol{\varphi\mu c_t}}\boldsymbol{\nabla^2 P}$$

But, in diffusivity equation we assume that the reservoir is homogeneous, while in reality and specifically in this case, the reservoir is not homogeneous and each layer has its specific permeability. Therefore, the diffusivity equation can not be used in this case.

Finite Differences

To solve problem including partial derivatives we often use finite difference method. Then, using the approximation of the derivatives, the finite difference method simplifies the equation in some times. There are lots of methods which can be used to solve this equation, including backward, central and forward difference methods. The forward difference method is an explicit differential method which uses the value in the next step to approximate the derivative at the given point. The backward difference is an implicit differential method, which means that it uses the value from the previous step to approximate the derivative. Central difference method is called the central difference method.

Forward difference method:

$$\boldsymbol{f'(x) = D_+ f(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}, O(\Delta x)}$$

Backward difference method:

$$\boldsymbol{f'(x) = D_- f(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x}, O(\Delta x)}$$

$O(\Delta x)$ is often used to show how far is the truncation of infinite series from the approximation given.

Boundary Conditions

We consider two boundary conditions, one regarding to the perforation zone, and the other with the no-flow region. Dirichlet and Newman boundary conditions, which came below:

Dirichlet Method:

$$\boldsymbol{P(x,t) = P_w}\,, \qquad \forall \boldsymbol{x} \in \boldsymbol{\Gamma_D}$$

Newman Method:

$$\boldsymbol{\frac{\partial P(x,t)}{\partial n} = 0}\,, \qquad \forall \boldsymbol{x} \in \boldsymbol{\Gamma_N}$$

P represents pressure, x is the position in x-axis, t is the time, the well pressure is specified with P_w while n is the normal vector to the surface. In Dirichlet condition, the pressure will be constant and equal to well pressure:

$$P = P_w$$

However, in Newman condition the pressure change in n direction is 0.

$$\boldsymbol{\frac{\partial P(x,t)}{\partial n} = 0}$$

In this case, there will not be any influx (is or out) in the normal direction to the boundary. There would not be any flow in y-direction and just in x-direction at the top layer of the reservoir. In contrast, there is will be a flow just in the y-direction and not x-direction a the end of the reservoir.

Execution

Various ways of execution are used. we have started with two loops in each other and improved it by the analytical method.

Loops

One of the ways is to use (for) loops in each other. Probably it is a faster and more explicit way to write this problem into the referred programming language. It works properly for vectors and matrices, except it is slow for the more extended matrices. In the nested loops, each loop each loop implements one of the parameters which are used in the pressure calculation. The first loop represents the x-direction and the second one is for the y-direction.

At first, we should calculate the rate (q) for both x and y directions for the two models. This calculation and the forward and backward difference method are executed by staggered grid method. Afterward, we calculate the pressure gradient (∇f) using the q arrays. The assumption is that the work of this method is complete, but because the time that is needed is too much, so the loops ended without the desirable result.

Analytical Solution

As It is mentioned in the question guide, the analytical solution for a perforated at one side and isolated on the other side homogeneous layer is:

$$\boldsymbol{P(x,y,t) = P_w + \sum_{i=-\infty}^{\infty} \frac{P_r - P_w}{\pi/2 + i\pi} e^{-\mathcal{K} t c_i^2} \sin(c_i x)}$$

While

$$\boldsymbol{c_i = \left(\pi/2 + i\pi\right)/L}$$

$$\boldsymbol{\mathcal{K} = k/(\mu \varphi c_t)}$$

It is essential to truncate the infinite sum in the analytical solution implementation. We should do that because it is not possible to compute the infinite sum with an analytical solution in Python. Hence, we must choose a restricted number of terms. A comparison is made regarding how the accuracy changes versus the number of terms and this is done for only 4 hours elapsed time. In order to achieve a result to compare, the first 10,000 terms implemented. Then, we reduced the number of the terms to analyze that how many terms is needed to achieve a result

with the same accuracy. We achieved the same accuracy of 10,000 terms by passing just 300 terms. Hence, we were resulted that this number of terms is enough. A bad solution would be achieved by using 100 terms which is not a sufficient representative for the analytical solution method.

One of the ways to perceive a comparison between the analytical solution and the numerical solution is to analyze the pressure difference resulted by both solutions. We found that the most interesting way to do that is to compare the pressure difference just in the top and the lower layers, and that should be done separately in each model, because no production is due from the middle layer (Layer 02).

Results

By plotting model 1 and 2 that are shown in figure …. We can see the difference between 2 models. This difference is in their permeability in layer 2 of each model. Permeability in model 2 is much higher than model 1. Both models have higher permeability in layer 1 and low permeability in layer 3 while layer 2 is almost impermeable.

After 4 hours a notable change can be seen in layer 1 in comparison with layer 3. Upper layer has faster flow than lower layer and pressure in middle layer model 1 is nearly same as before whereas in model 2 is changing slowly. After 4 days, as you can see in figure … the middle layer in model 1 started extracting while both layers 1 and 3 show similar pressure. Extraction in model 2 is significantly high.

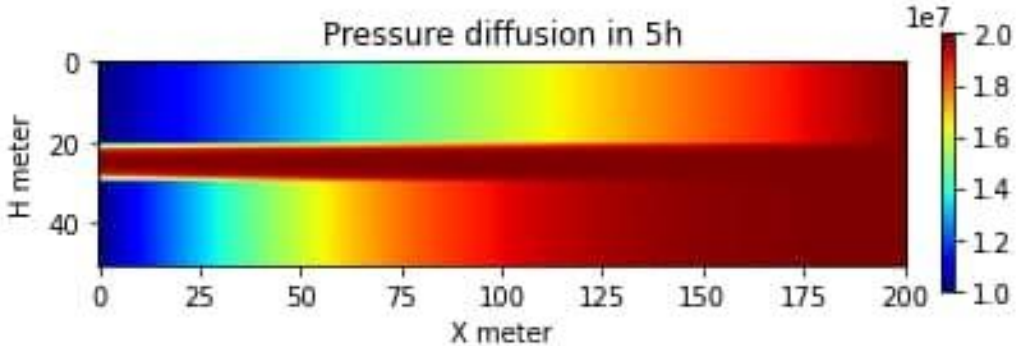

Figure 1, Plot of Crossflow in Layer 02 for Model 01 after 5 hours

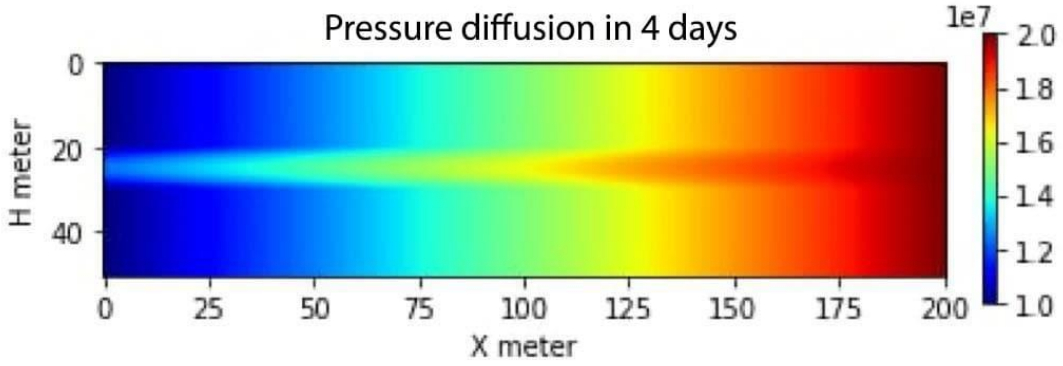

Figure 2, Plot of Crossflow in Layer 02 for Model 01 after 4 days

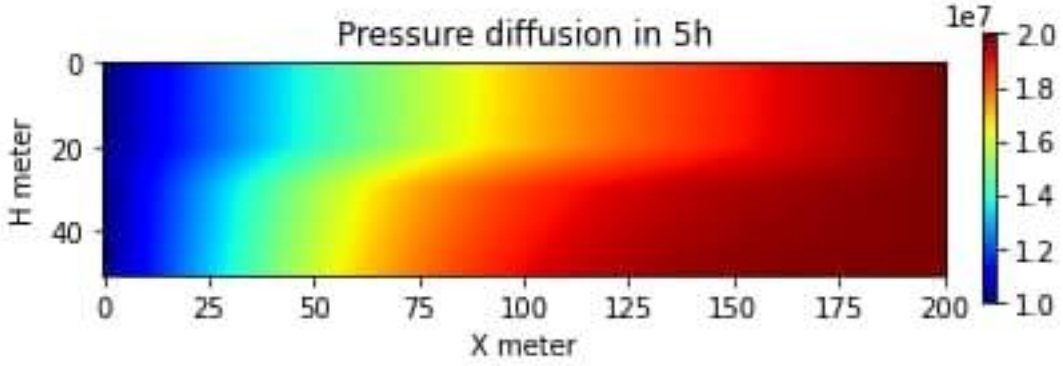

Figure 3, Plot of Crossflow in Layer 02 for Model 02 after 5 hours

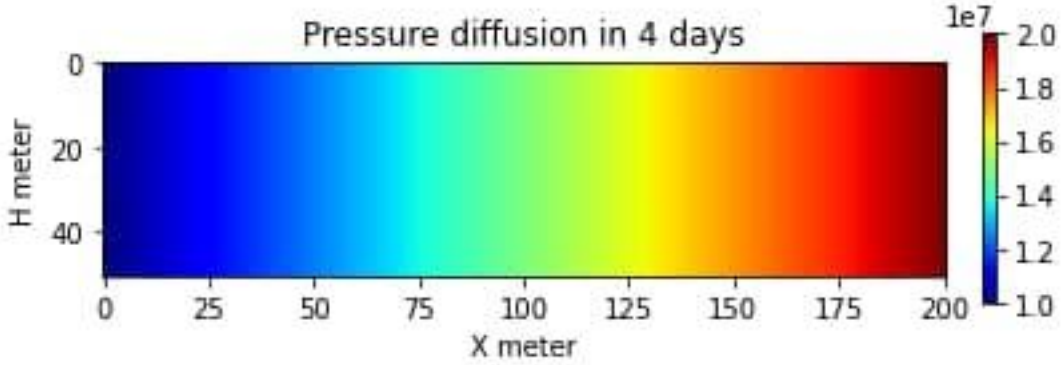

Figure 4, Plot of Crossflow in Layer 02 for Model 02 after 4 days

Since this method considers only horizontal flow, after 4 days we see a bigger difference of extraction in layer 2 which causes top and bottom layers' depletion too. This variation is almost

negligible in 4 hours. The plots for model 1 and 2 using analytical method are shown respectively in figure 5 to 8.

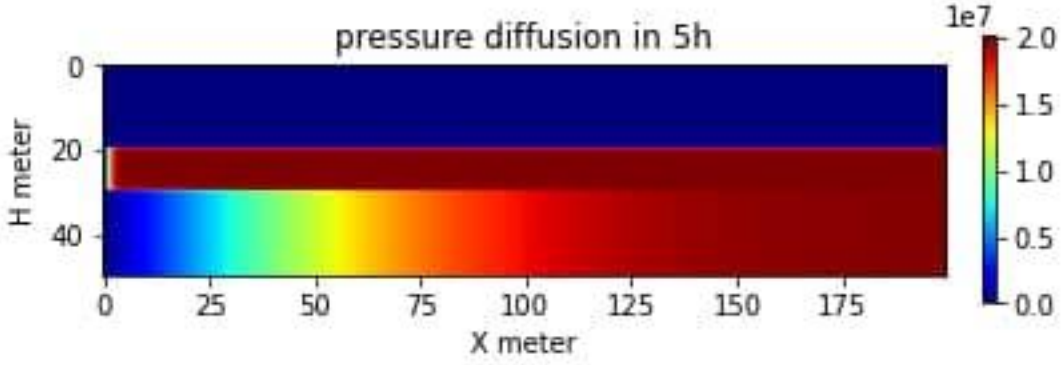

Figure 5, Analytical Method in Layer 02 for Model 01 after 5 hours

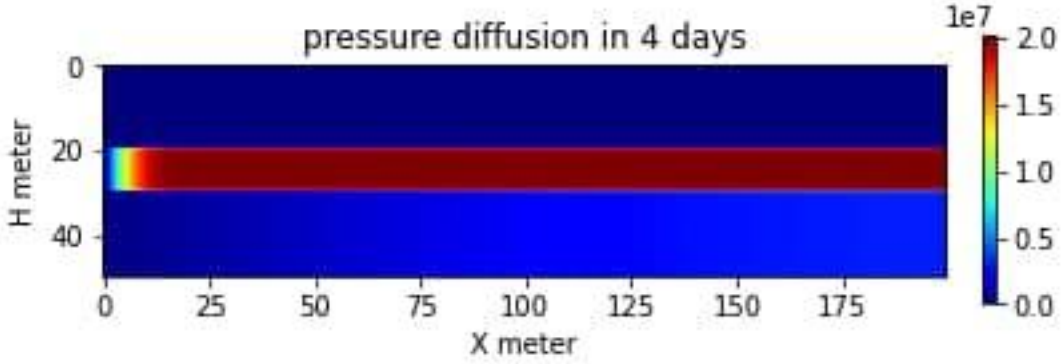

Figure 6, Analytical method in layer 02 for Model 01 after 4 days

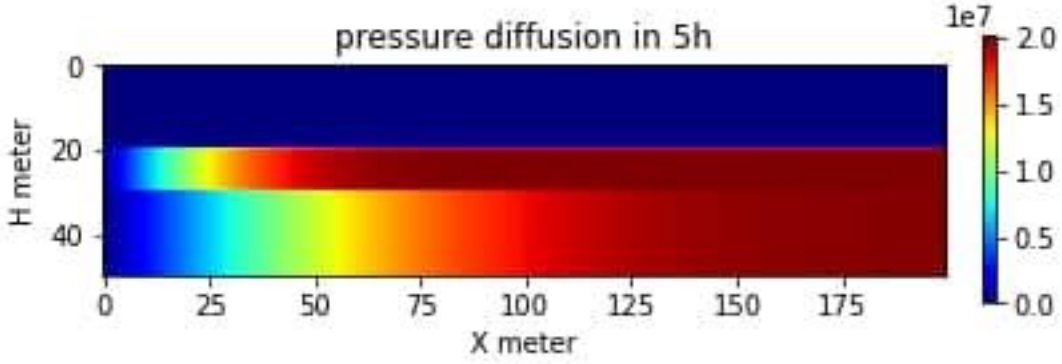

Figure 7, Analytical method in layer 02 for Model 02 after 5 hours

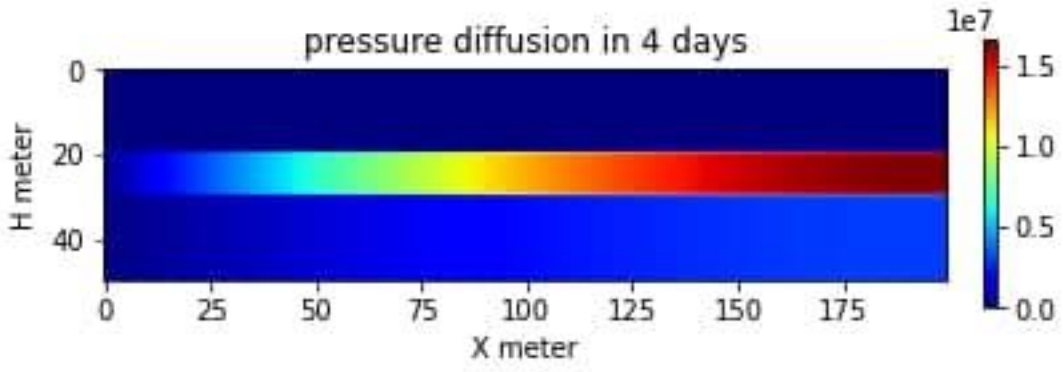

Figure 8, Analytical method in layer 02 for Model 02 after 4 days

Conclusion

Reservoir simulation is an area in which computer models are used to predict the flow of fluids. Two different models simulated by two different methods throughout this project. The general form of the diffusion equation with finite difference method is used to make reservoir model of the subsurface based on reservoir parameters. In this project, model 1 differs model 2 in the permeability of middle layer. For looking over the variation in pressure distribution in these 2 models, we simulated both in short and long elapsed time. Implementation of the numerical method is needed to optimize the run time and subsequently solve the numerical problem with an improved algorithm. Also, in each model an analytical solution has been used and compared with numerical method. All vital criteria such as stability, convergence, consistency and boundary conditions have been considered.

$$\boldsymbol{q} = -\frac{\boldsymbol{k}}{\boldsymbol{\mu}}\boldsymbol{\nabla P}$$

$$\frac{\boldsymbol{\partial P}}{\boldsymbol{\partial t}} = -\frac{\mathbf{1}}{\boldsymbol{\rho\varphi c_t}}\boldsymbol{\nabla}.(\boldsymbol{\rho q})$$

$\boldsymbol{q} = volumetric\ flux$

$\boldsymbol{P} = pressure$

$\boldsymbol{\rho} = Fluid\ density$

$\boldsymbol{\varphi} = Porosity$

$\boldsymbol{c_t} = Total\ compressibility\ of\ the\ formation\ and\ fluid$

$\boldsymbol{k} = Permeability$

$\boldsymbol{\mu} = The\ pressure\ independent\ viscosity\ of\ the\ fluid$

$\boldsymbol{\mathcal{K}} = {}^{k}/_{(\mu\varphi c_t)}$

$\boldsymbol{n} = \textit{The normal to the boundary}$

$P_w = 10\ MPa$

$P_r = 20\ MPa$

$L = 200\ m$

$\boldsymbol{Q} = \int_{\boldsymbol{\Gamma_D}} \boldsymbol{q}.\boldsymbol{n}\ \boldsymbol{d\Gamma}$

Medium parameters for reservoir modeling:

	Model 1			Model 2		
	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3
Height, m	20	10	20	20	10	20
Density, kg/m^3	930	930	930	930	930	930
Viscosity, Pa s	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Total compressibility, Pa^{-1}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}
Porosity	0.2	0.2	0.2	0.2	0.2	0.2
Permeability, m^2	10^{-13}	10^{-17}	$2 \cdot 10^{-14}$	10^{-13}	$4 \cdot 10^{-15}$	$2 \cdot 10^{-14}$

References

[1] Jon Kleppe: Lecture note 4. http://www.ipt.ntnu.no/~kleppe/ TPG4160/note4.pdf. Accessed: 2019-11-15.

[2] Numerical Methods. http://ksuweb.kennesaw.edu/~plaval/ /math4310/num_derivapp.pdf. Accessed: 2019-11-15.

[3] Wiki Pengtools: Darcy's law. https://wiki.pengtools.com/ index.php?title=Darcy%27s_law. Accessed: 2019-11-15.

[4] John H. Mathews, Kurtis D. Fink, 2004: Numerical Methods Using MATLAB.

Appendix

Finite Difference PYTHON Code

```
import numpy as np
import pylab as pl
from pylab import draw, ion

def dxplus(A,dx):
   nx=A.shape[0]
   ny=A.shape[1]
   dA=np.zeros((nx,ny))
   for j in range(0,ny):
      for i in range(0,nx-1):
         dA[i,j]=(A[i+1,j]-A[i,j])/dx
         return dA

def dxminus(A,dx):
   nx=A.shape[0]
   ny=A.shape[1]
   dA=np.zeros((nx,ny))
   for j in range(0,ny):
      for i in range(1,nx):
         dA[i,j]=(A[i,j]-A[i-1,j])/dx
         return dA

def dyplus(A,dx):
   nx=A.shape[0]
   ny=A.shape[1]
   dA=np.zeros((nx,ny))
   for i in range(0,nx):
      for j in range(0,ny-1):
         dA[i,j]=(A[i,j+1]-A[i,j])/dx
         return dA

def dyminus(A,dx):
   nx=A.shape[0]
   ny=A.shape[1]
   dA=np.zeros((nx,ny))
   for i in range(0,nx):
      for j in range(1,ny):
         dA[i,j]=(A[i,j]-A[i,j-1])/dx
         return dA

#Parameters
#Reservoir Length
L=200
#Reservoir Height
```

```
H=50
#Griding
dx=1
dt=0.5
#for 5 hours
Tmax=3600*5*24
Nt=int(Tmax/dt)+1
#X-direction Grids
Nx=int(L/dx)+1
#Y-direction Grids
Ny=int(H/dx)+1

kappa=np.zeros((Nx,Ny))
ro=np.zeros((Nx,Ny))
mu=np.zeros((Nx,Ny))
ct=np.zeros((Nx,Ny))
por=np.zeros((Nx,Ny))

#Reservoir Properties
#Model 01 Permeability
kappa[:,:20]=1.0e-13
kappa[:,:]=1e-17
kappa[:,30:]=2*1e-14
#Model 02 Permeability
kappa[:,:20]=1.0e-13
kappa[:,:]=4*1e-15
kappa[:,30:]=2*1e-14
ro[:,:]=930.0
mu[:,:]=1.0e-3
ct[:,:]=1.0e-9
por[:,:]=0.2
Pinit=2.0e+07
Pwf=1.0e+07

#Present Pressure
P0=np.zeros((Nx,Ny))
#Initial Pressure
P0[:,:]=Pinit
#Future Pressure
P1=np.zeros((Nx,Ny))
#Darcy Flow in X-direction
qx=np.zeros((Nx,Ny))
#Darcy Flow in Y-direction
qy=np.zeros((Nx,Ny))


#Stability
for i in range(0,Nx):
```

```
    for j in range(0,Ny):
        alpha=((2.0*dt/dx*dx)*kappa[i,j]/mu[i,j]*por[i,j]*ct[i,j])

        if (alpha>0.5):
            print("It Is Not Stable")
print("Alpha: ",alpha)

#Boundary Condition (Left Hand Side)
P0[0,:]=Pwf

#Main Time Loop
for k in range(0,Nt):
    if(k%100==0):
        print("Timestep= ",k)

    qx=dxplus(P0,dx)
    qx=(-kappa/mu)*qx
    qx=dxminus(qx,dx)
    qx[200:,:]=0

    qy=dyplus(P0,dx)
    qy=(-kappa/mu)*qy
    qy=dyminus(qy,dx)

    #Neuman
    qy[:,:0]=0
    qy[:,50:]=0

    P1=P0+(-dt/(por*ct))*(qx+qy)
    P0=P1

    #Boundary (Right-Hand Side)
    #Dirichlet
    P0[0,:20]=Pwf
    P0[0,30:]=Pwf

    #Plotting
    plt=np.transpose(P1)
    im=pl.imshow(plt)
    cbar=pl.colorbar(im,fraction=0.015,pad=0.04)
    pl.title('Crossflow (5 Hours)')
    pl.xlabel('x (m)')
    pl.ylabel('Depth (m)')
    pl.savefig('Finite Difference (Crossflow) in 5 Hours.pdf')
    pl.show()
```

Analytical Solution PYTHON Code

```
import numpy as np
import math
import matplotlib.pyplot as pl


Pw=1e7;
k=[1e-3,4*1e-15,2*1e-14]
ny=50
nx=200
P=np.zeros((ny,nx))
l=200
maxt=3600*5
n=300


def sum (n,t,x,k):
    Pw=1e7
    Pr=2e7
    Phi=0.2
    Mu=1e-3
    c=1e-9
    K=k/(Mu*c*Phi)
    s=0
    for i in range (-n,n):
        c=(np.pi/2+i*np.pi)/l
        s=s+((Pr-Pw)/c)*(math.e**(-K*t*c**2))*np.sin(c*x)
    return s
for k in (k):
    if k==1e-13:
        for i in range (20):
```

```
            for j in range (nx-1):
                P[i,j]=(Pw+sum(n,maxt,j,k))/100
    else:
        if k==4*1e-15:
            for i in range (20,30):
                for j in range (nx):
                    P[i,j]=(2*1e7+sum(n,maxt,j,k))/100
        if k==2*1e-14:
            for i in range (30,50):
                for j in range (nx):
                    P[i,j]=(Pw+sum(n,maxt,j,k))/100

im=pl.imshow(P,cmap='jet')
cbar=pl.colorbar(im,fraction=0.015,pad=0.04)
pl.title('pressure diffusion in 5h')
pl.xlabel('X meter')
pl.ylabel('H meter')
pl.savefig('model2 analytical.jpg')
```