



3) Less pumping is required when the well is located upgradient of the lake. More pumping is required when the well si downstream of the lake because the downstream well pulls more from the lake. The upstream well captures less water from the lake because the gradient from the well to the lake is less in the upstream case than the downstream case. The flownet for the upstream well placment resembles the flownet for the lake without the well more than the downstream well placment’s flownet.

Figures:

Lake raised above aquifer average

C:\Users\Jack\AppData\Local\Microsoft\Windows\INetCache\Content.Word\situation1.tif

Well placed upstream:

C:\Users\Jack\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Upstream_well.tif

Well placed downstream:

C:\Users\Jack\AppData\Local\Microsoft\Windows\INetCache\Content.Word\downstream_well.tif

**Code:**

Runfile1:

Qx0 = 0.63;

Phi0= 2576.45;

rl= 100;

Phi1 = 2880

l = 1000

Q= (Phi1 - Phi0 -Qx0\*(l - rl\*rl/l) )\*2\* pi/log(l/rl)

ContourMe\_flow\_net(-1000,1000,500,-500,500,500,@(z)omega\_total1(z,Q,Qx0, Phi0, rl),30);

Omega\_total1:

function [ Omega ] = omega\_total1( z, Q,Qx0, Phi0, rl)

%OMEGA\_TOTAL Summary of this function goes here

% Detailed explanation goes here

rsq=(z)\*conj(z);

if rsq>rl^2

Omega = omega\_uniformflow\_lake( Qx0, z, rl) + omega\_well( z, 0 , rl, Q) +Phi0;

else

Omega = Phi0;

end

end

omega\_uniforflow\_lake:

function [ Omega ] = omega\_uniformflow\_lake( Qx0, z, rl )

%OMEGA\_UNIFORMFLOW\_LAKE Summary of this function goes here

% Detailed explanation goes here

Omega = -Qx0\*(z-rl\*rl/z);

end

omega\_well:

function [ Omega ] = Omega\_well(z,z0,rw,Q)

rsq=(z-z0)\*conj(z-z0);

if rsq>rw^2

Omega=Q/(2\*pi)\*log(z-z0);

else

Omega = 0;

end

Runfile2:

Qx0 = 0.63;

Phi0= 2576.45;

rl= 100;

Phi1=2880;

l=1000;

zw =-l/2;

Q=2\*pi\* (Phi1 - Phi0 + Qx0\*(-l+rl\*rl/l))/real(log((rl/(-l\*conj(-l)))\*(-l-zw)/(-l-rl\*-1/conj(zw))))

ContourMe\_flow\_net(-1000,1000,500,-500,500,500,@(z)omega\_total2(z,zw,Q,Qx0, Phi0, rl),30);

Omega\_total2:

function [ Omega ] = omega\_total2( z, zw, Q,Qx0, Phi0, rl)

rsq=(z)\*conj(z);

if rsq>rl^2

Omega = omega\_uniformflow\_lake(Qx0, z,rl) + omega\_well\_near\_lake(z,zw,rl,Q) + Phi0;

else

Omega = Phi0;

end

end

omega\_well\_near\_lake:

function [ Omega ] = omega\_well\_near\_lake( z,z0,rl,Q )

%OMEGA\_WELL\_NEAR\_LAKE Summary of this function goes here

% Detailed explanation goes here

Omega=Q/(2\*pi)\*log((rl/(z\*conj(z)))\*(z-z0)/(z-rl\*z/conj(z0)));

end