Project 2:

Read Chapter 2: "The flow of a homogenous layer with a free surface" (Baines)

1. Recreate Figure 2.6 (Baines), Using the method of characteristics calculating one-dimensional non-linear, hydrostatic flow over a bump
2. Using MITgcm to recreate one-dimensional non-linear, hydrostatic flow over a bump. A reduced gravity simulation (two layers, very thick upper layer) will be used to approximate a surface wave flow.
3. Analyze the jump condition, looking at the rate of energy dissipation and speed (equations 2.3.12 and 2.3.13 (Baines), respectively).
4. Examining the 5 flow regimes in Figure 2.11 (Baines), varying the Froude number and obstacle height.

*** Note: Codes are included in this pdf, but are also available in the folder.

<u>PART 1</u>

One-dimensional non-linear hydrostatic flow:

$$u_t + uu_x = -g\eta_x \qquad \text{where,} \qquad d = d_o + \eta - h$$
$$d_t + (du)_x = 0$$

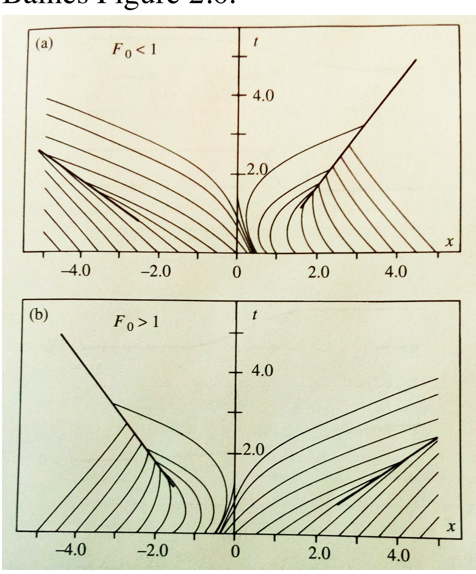
The characteristic form is:

$$\frac{d}{dt}\left(u \pm 2\sqrt{gd}\right) = -g\frac{dh}{dx}$$

with initial conditions $u = u_o$, $d = d_o$, at $t = 0$.

Where u and d are related by the Froude number: $Fr = \frac{u}{\sqrt{gd}}$.

Baines Figure 2.6:



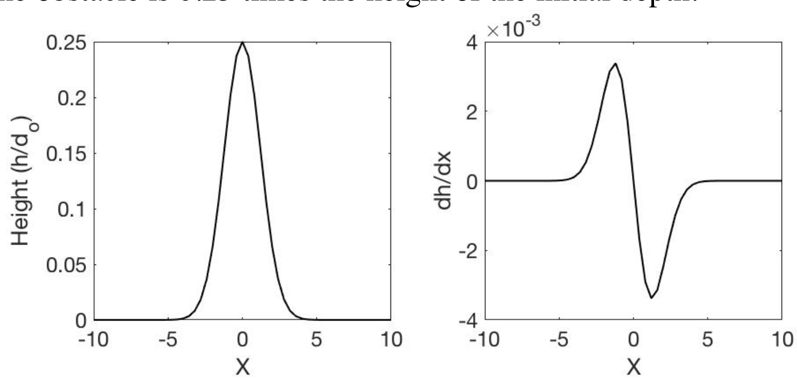
Upstream-characteristics for one-dimensional non-linear, hydrostatic flow over a obstacle. The obstacle is located at x=0, and falls off to zero by x = ±3. The hydraulic jumps are denoted by the heavy lines.

Here is the Matlab code (project2figure26.m) I used to recreate Baines' Figure 2.6:

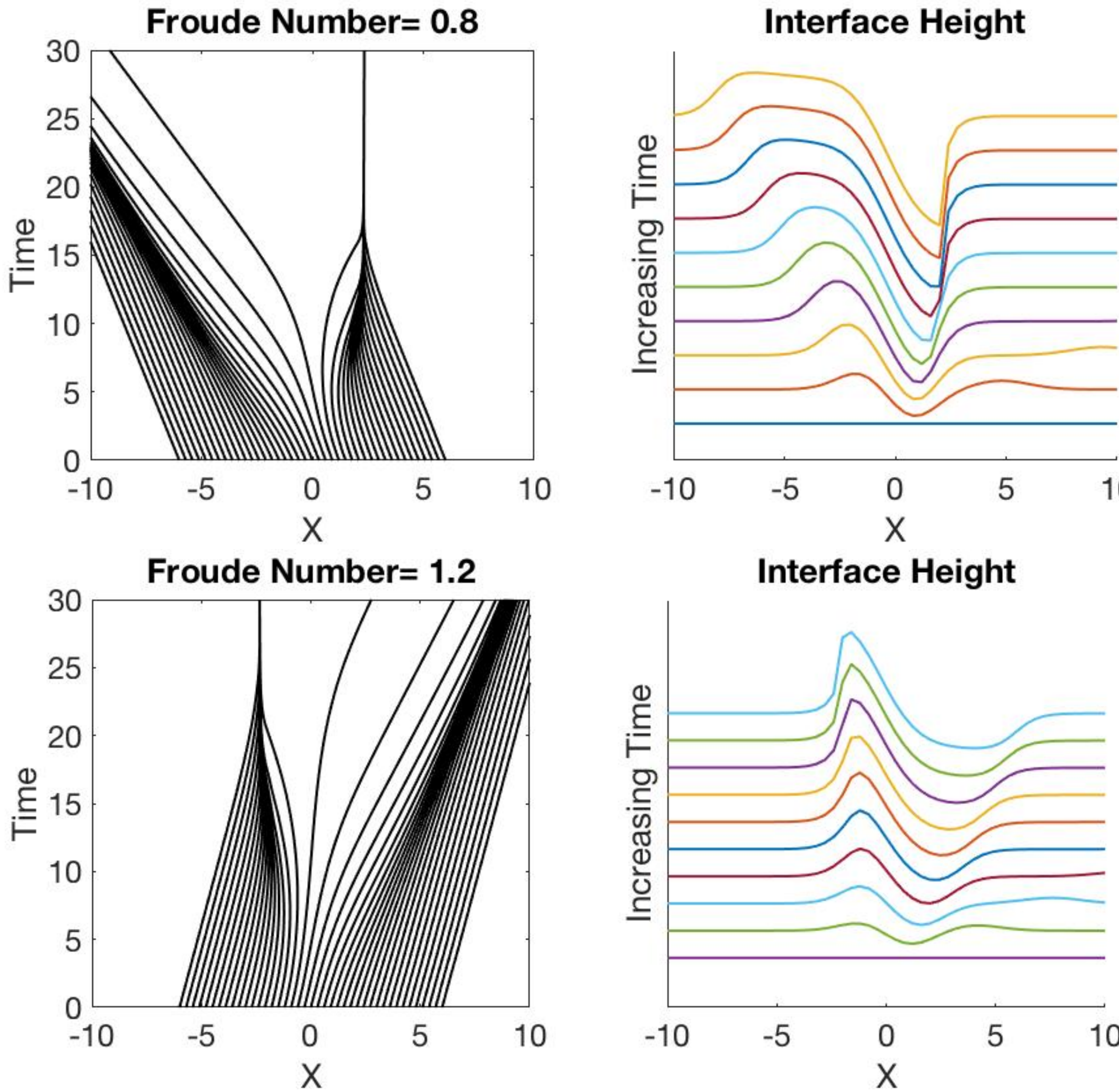
```
% Recreating Figure 2.6 (Baines) using the method of characteristics

% 1d nonlinear hydrostatic flow over an obstacle centered at 0 and
% approaching a height of 0 of x= +/- 3

%% Setup

clear

x3=[-20:0.4:20];        % x-position

dt=0.2;                 % delta T
t_o=0;                  % initial time
t=[t_o:dt:30];          % time array

NT= length(t);          % number of time steps
NX=length(x3);          % number of x steps

g=9.81;                 % gravity

Fr=0.8; % 1.2; %        % Froude number
u_o=1;                  % initial velocity
d_o=((u_o/Fr).^2)/g;    % initial depth

xchar= [-6:0.3:6];      % x- characteristics

H=0.25*d_o;             % height of obstacle
h=exp(((-x3.^2)./3))*H; % obstacle
h_x=gradient(h);        % gradient of obstacle dh/dx
%h_x=-2*a*exp(-x3.^2).*x3;

method='linear';        % for interpolation

%% Initializing

Ui = zeros(NT,NX);
Ui(1,:) = u_o;

D = zeros(NT,NX);
D(1,:) = d_o;

char= zeros(NT,length(xchar));
char(1,:)=xchar;

%% Calculating characteristics for all x and t

for ii=2:NT;

    for kk=1:NX;

        xx=x3(kk);

        u3=Ui(ii-1,kk);
        d3=D(ii-1,kk);

        x1 = xx - (u3 + sqrt(g*d3))*dt;
        x2 = xx - (u3 - sqrt(g*d3))*dt;

        d1 = interp1(x3, D(ii-1,:),x1,method,'extrap');
        d2 = interp1(x3, D(ii-1,:),x2,method,'extrap');

        u1 = interp1(x3, Ui(ii-1,:),x1,method,'extrap');
        u2 = interp1(x3, Ui(ii-1,:),x2,method,'extrap');

        zeta = u1 + 2*sqrt(g*d1);
        eta = u2 - 2*sqrt(g*d2);

        dhdx1 = interp1(x3, h_x,x1,method,'extrap');
        dhdx2 = interp1(x3, h_x,x2,method,'extrap');
```

```
        Ui(ii, kk) = (-g*(dhdx1 + dhdx2)*dt + zeta + eta)/2;
        Ci= ( -g*(dhdx1 - dhdx2)*dt +zeta - eta)/4;
        D(ii, kk) = (Ci.^2)./g;

    end

    q = char(ii-1 , :);
    uq=interp1(x3, Ui(ii, :), q,method,'extrap');
    dq = interp1(x3, D(ii, :), q,method,'extrap');
    char(ii, :) = (uq - sqrt(g*dq))*dt + q;

end

%% Figures

figure(1)
subplot(1,2,1)
plot(x3, h/d_o, 'k','LineWidth',2)
xlabel('X')
ylabel('Height (h/d_o)')
set(gca,'FontSize',24)
xlim([-10 10])

subplot(1,2,2)
plot(x3, h_x, 'k','LineWidth',2)
xlabel('X')
ylabel('dh/dx')
set(gca,'FontSize',24)
xlim([-10 10])

figure(2)

subplot(1,2,1)
plot(char, t, 'k','LineWidth',2)
xlabel('X')
ylabel('Time')
set(gca,'FontSize',24)
title('Froude Number= 0.8')
ylim([0 30])
xlim([-10 10])

subplot(1,2,2)
hold on
nn=-1;
for jj = 1:10:100
 plot(x3,D(jj,:)/d_o+nn,'LineWidth',2)
 nn=nn+0.1;
end
xlabel('X')
ylabel('Increasing Time')
title('Interface Height')
xlim([-10 10])
set(gca,'FontSize',24)
set(gca,'YTick',[]);
```

The obstacle is 0.25 times the height of the initial depth.



My results are:



- My results are similar to Figure 2.6. For *Fr = 0.8*, the characteristics slowly converge and upstream jump. A downstream hydraulic jump also forms. For *Fr =1.2* it is reversed, with a upstream hydraulic jump forming.
- For both *Fr = 0.8, 1.2* the initial waves are similar, with a small decrease downstream and small increase upstream. However with time the downstream depression becomes much larger for *Fr = 0.8* compared to *Fr = 1.2* and the increased height upstream becomes much larger for *Fr = 1.2* compared to *Fr = 0.8*. As the thickness changes so does the speed of the water. A downstream jump forms when *Fr = 0.8* and an upstream jump forms when *Fr = 1.2*.

PART 2

Next I repeated this flow set up, but using MITgcm. I used I reduced gravity simulation, with two layers. The bottom layer is much smaller compared to the upper layer; this way the interface represents the free surface.

My gendata.py file is:

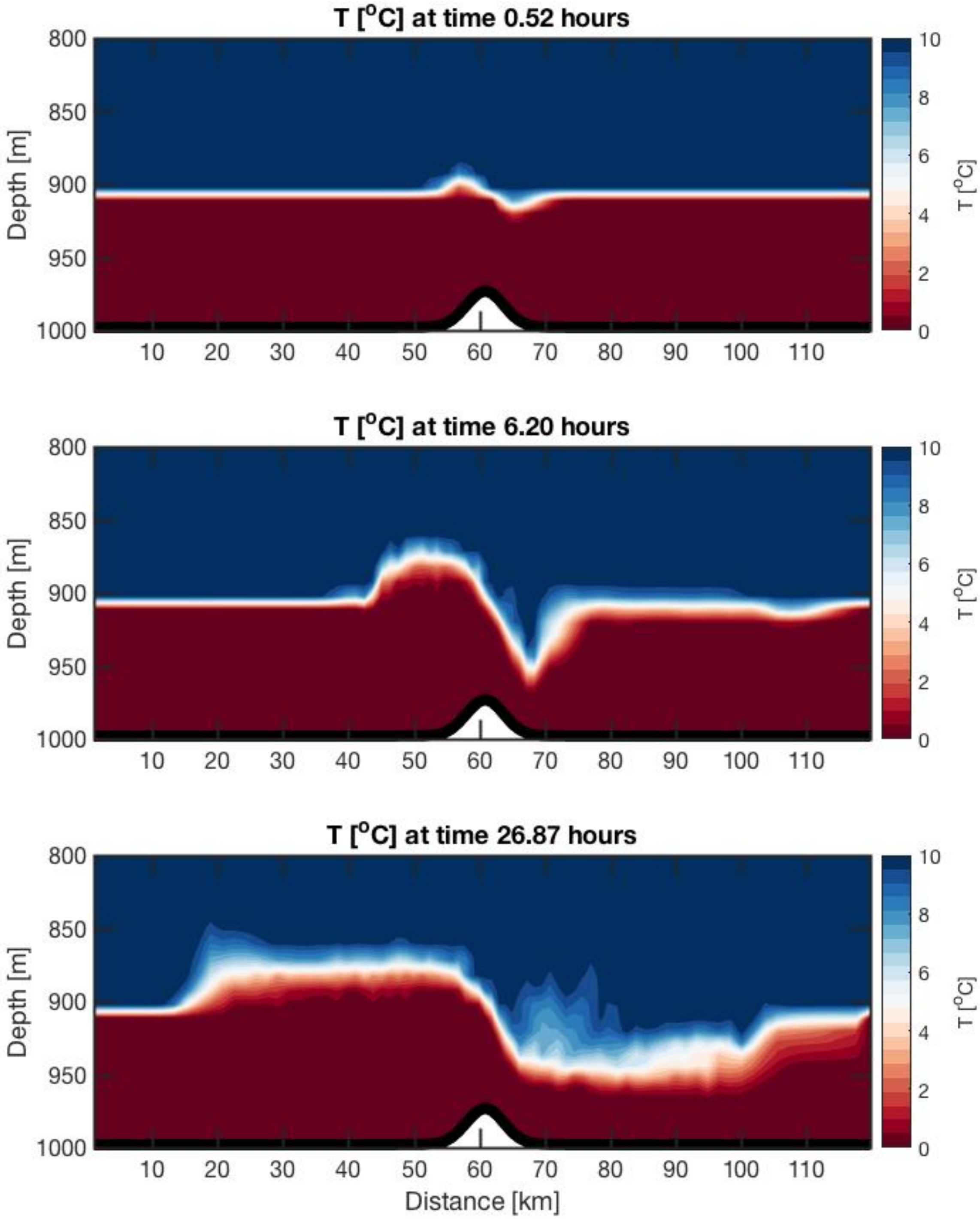
```
# 1d nonlinear hydrostatic flow over a gaussian bump - simulated by reduced gravity boundary as
the free surface
# this reproduces the 5 regimes shown in Figure 2.11 of Baines and two other simulations with Fr=
0.8 and 1.2
# the 5 regimes are Supercritical flow. Subcritical flow. Partially blocked no lee jump.
Partially blocked with lee jump. Complete blocking.
# need to vary Fr and H_m=h_m/d_o

from numpy import *
from scipy import *
from pylab import *
import numpy.matlib as matlib
from shutil import copy
from os import mkdir
import os


Fr=2                                                                   #Variable
H = 1000
d0=100
H_m=0.25                                                               #Variable
h_m=H_m*d0
g=9.81

outdir='../runs/Run_trial_dT_2'

# 5 different regimes + Fr=0.8, Fr=1.2
#Supercritical, Fr=2 H_m=0.25
#Subcritical,   Fr=0.25 H_m=0.25
#Complete_blocking,   Fr=0.25 H_m=2
#Partially_blocked_with_lee_jump, Fr=0.25 H_m=1
#Partially_blocked_no_lee_jump,   Fr=1.2 H_m=1
#Fr0p8H0p25,  Fr=0.8 H_m=0.25
#Fr1p2H0p25   Fr=1.2 H_m=0.25

try:
  mkdir(outdir)
except:
  print outdir+' Exists'
try:
  mkdir(outdir+'/figs')
except:
  print outdir+'/figs Exists'
copy('./gendata.py',outdir)

# These must match ../code/SIZE.h
ny = 1
nx = 4*20
nz = 50

# y direction:
dy = 1000
# x direction
xt = 410e3

# dx
dx=zeros(nx)+1000
p1=np.arange(nx/2)
p2=np.flipud(p1)
p3=np.concatenate((p2,p1), axis=0)
dx=dx*1.02**p3
```

```
x=cumsum(dx)
x = x-x[nx/2]

with open(outdir+"/delXvar.bin", "wb") as f:
       dx.tofile(f)
f.close()

# plot
if 1:
    plot(x/1000.,dx)
    #xlim([-10,10])
    savefig(outdir+'/figs/dx.pdf')

# topo
sigma = 4000. # m
topo = h_m*exp(-x*x/(sigma**2))-H #correct?
print shape(topo)
topo[topo>0.]=0.

with open(outdir+"/topo.bin", "wb") as f:
       topo.tofile(f)
f.close()

# plot
if 1:
    clf()
    plot(x/1.e3,topo)
    # xlim([-20.,20.])
    savefig(outdir+'/figs/topo.pdf')

# dz
dz=zeros(nz)+4.78
p4=np.arange(nz)
p5=np.flipud(p4)
dz=dz*1.05**p5
z=cumsum(dz)
z = [(z[:-1]+z[1:])/2]

with open(outdir+"/delZvar.bin", "wb") as f:
       dz.tofile(f)
f.close()

# Temperature profile ... two layer
gravity=9.81
alpha = 2e-4
g_reduced=0.4E-3*gravity
dT=g_reduced/(gravity*alpha)
u0=Fr*sqrt(g_reduced*d0)

Tref=np.ones(nz)*dT
dz_flip=np.flipud(dz)
CumlativeSum=cumsum(dz_flip)
CumlativeSum_flip=np.flipud(CumlativeSum)

for j in range(0, nz):
    if CumlativeSum_flip[j] < d0:
        Tref[j]=0

#T0 = Tref[np.newaxis, :]*np.ones((nx,nz))
T0 = Tref[:,np.newaxis]*np.ones((nz,nx))

with open(outdir+"/TRef.bin", "wb") as f:
       Tref.tofile(f)
f.close()

# save T0 over whole domain

with open(outdir+"/T0.bin", "wb") as f:
       T0.tofile(f)

#initial velocity
U_initial=u0*np.ones(nz)
for j in range(0, nz):
```

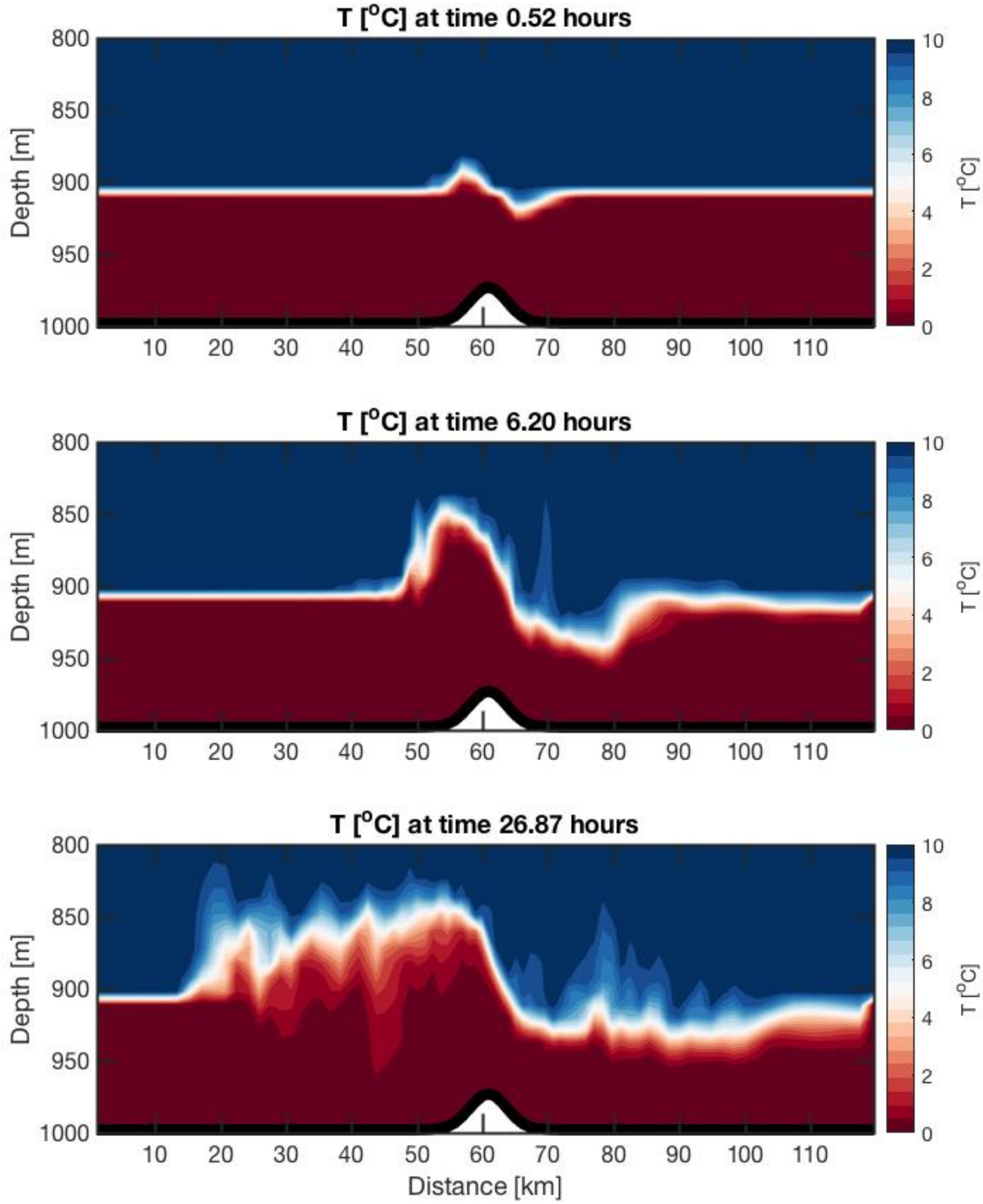
```
    if CumlativeSum_flip[j] > d0:
        U_initial[j]=0

#U0=U_initial[np.newaxis, :]*np.ones((nx, nz))
U0=U_initial[:,np.newaxis]*np.ones((nz,nx))

with open(outdir+"/Uin.bin","wb") as f:
  U_initial.tofile(f)

with open(outdir+"/U0.bin", "wb") as f:
  U0.tofile(f)

# plot:
if 1:
    clf()
    plot(Tref,CumlativeSum)
    savefig(outdir+'/figs/Tref.pdf')


with open(outdir+"/Ue.bin","wb") as f:
  U_initial.tofile(f)

with open(outdir+"/Uw.bin", "wb") as f:
  U_initial.tofile(f)

with open(outdir+"/Te.bin", "wb") as f:
       Tref.tofile(f)
f.close()
with open(outdir+"/Tw.bin", "wb") as f:
       Tref.tofile(f)
f.close()

## Copy some other files
import shutil
shutil.copy('data', outdir+'/data')
shutil.copy('eedata', outdir)
shutil.copy('data.kl10', outdir)
shutil.copy('data.mnc', outdir)
shutil.copy('data.obcs', outdir)
shutil.copy('data.diagnostics', outdir)
shutil.copy('data.pkg', outdir+'/data.pkg')
# also store these.  They are small and helpful to document what we did
for nm in {'input','code','build_options','analysis'}:
    to_path = outdir+'/'+nm
    if os.path.exists(to_path):
        shutil.rmtree(to_path)
    shutil.copytree('../'+nm, outdir+'/'+nm)
```

My results are:

For *Fr = 0.8*



For *Fr = 1.2*



- I have plotted the temperature of the layers. The upper layer was initially 10ºC and the bottom layer was initially 0ºC. The interface between the layers evolves similarly to the method of characteristics. The downstream depression is much larger for *Fr = 0.8*, and the upstream is greater for *Fr = 1.2*. Note the depth scale differences.

PART 3

Approximate the jump condition:
The rate of energy dissipation within the jump is (equation 2.3.12 (Baines)):

$$\frac{dE_J}{dt} = \frac{\rho_o g Q}{4}\frac{(d_d - d_u)^3}{d_d d_u} \qquad \text{where,} \quad Q = u_u d_u = u_d d_d$$

The speed of the jump is (equation 2.3.13 (Baines)):

$$c_J^2 = \frac{g d_d}{2}\left(1 + \frac{d_d}{d_u}\right)$$

For $Fr = 0.8$

- $u_o = 1\ m/s$
- $d_o = 0.1593\ m$
- $g = 9.81 \div 1000\ m/s^2$
- $Q = 0.1593\ m^2/s$

Upstream:

- $d_u = 100\ m$
- $d_d = 118\ m$
- $dE_J/dt = (\rho_o g Q/4)\cdot 0.49$
- $c_J = 1.1\ m/s$
- from the MITgcm simulation the upstream speed is $\sim 0.8\ m/s$

Downstream:

- $d_u = 55\ m$
- $d_d = 100\ m$
- $dE_J/dt = (\rho_o g Q/4)\cdot 16.57$
- $c_J = 1.2\ m/s$
- from the MITgcm simulation the downstream speed is $\sim 1.3\ m/s$

For $Fr = 1.2$

- $u_o = 1\ m/s$
- $d_o = 0.0708\ m$
- $g = 9.81 \div 1000\ m/s^2$
- $Q = 0.0708\ m^2/s$

Upstream:

- $d_u = 100\ m$
- $d_d = 135\ m$
- $dE_J/dt = (\rho_o g Q/4)\cdot 3.18$
- $c_J = 1.25\ m/s$
- from the MITgcm simulation the upstream speed is $\sim 1.3\ m/s$

Downstream:

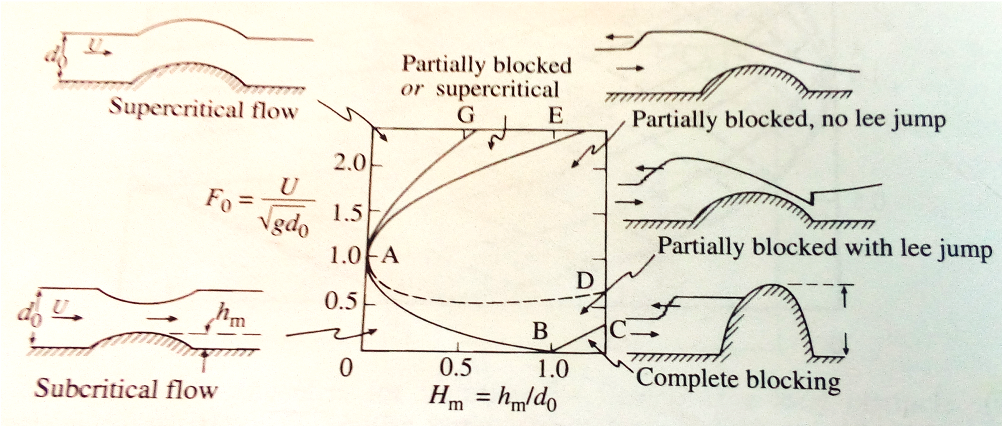
- $d_u = 68\ m$
- $d_d = 100\ m$
- $dE_J/dt = (\rho_o g Q/4)\cdot 1.95$
- $c_J = 1.1\ m/s$
- from the MITgcm simulation the downstream speed is $\sim 1.9\ m/s$

- For $Fr = 0.8$, the energy dissipation is much larger in the downstream hydraulic jump compared to the upstream counterpart. It is the opposite for $Fr = 1.2$, where the upstream hydraulic jump dissipates more energy than downstream.

- The speed calculations are estimates, where d_d and d_u were obtained from the above MITgcm simulations. Two of the predicted speeds are similar to the values from the simulation, however the other two are not similar. Velocities in the simulation are not uniform as assumed in equation 2.3.13. Furthermore the upstream and downstream depths are not fully constant either.

PART 4

Examining the 5 flow regimes in Figure 2.11 (Baines), varying the Froude number and obstacle height:



In my MITgcm simulations I used the followed H_m and F_o values:

- Supercritical $H_m = 0.25$ $F_o = 2$
- Subcritical $H_m = 0.25$ $F_o = 0.25$
- Partially blocked, no lee jump $H_m = 1$ $F_o = 1.2$
- Partially blocked, with lee jump $H_m = 1$ $F_o = 0.25$
- Complete blocking $H_m = 2$ $F_o = 0.25$

The Matlab (MITgcm_Run_reduced_gravity) code used to analyze the data generated by gendata.py is:

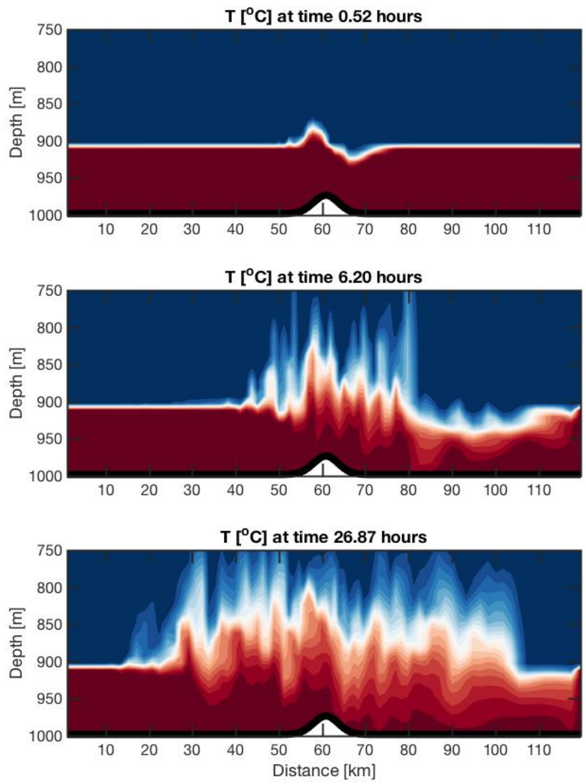
```
% Nonlinear hysdrostatic flow over a gaussian bump
% MITgcm reduced gravity simulation
% Fr= 1.2
% H_m= 0.25

%% Read Data

U_t1 = rdmds('U',150, 'n'); % Velocity [m/s]
U_t2 = rdmds('U',1800, 'n');
U_t3 = rdmds('U',7800, 'n');

T_t1 = rdmds('T',150, 'n'); % Temperature [^oC]
T_t2 = rdmds('T',1800, 'n');
T_t3 = rdmds('T',7800, 'n');

Depth = rdmds('Depth', 'n');

XC = rdmds('XC', 'n'); % X [m]
XG = rdmds('XG', 'n');

RC = rdmds('RC', 'n'); % Z [m]
RF = rdmds('RF', 'n');

XC= XC./1000; %distance in km

%% Time
% the time [s] is the file name # multiplied by the time step = 12.4 seconds

Time1=150*12.4/60/60; %Time in hours
Time2=1800*12.4/60/60;
```

```
Time3=7800*12.4/60/60;

Str1=sprintf('U [m/s] at time %0.2f hours',Time1);
Str2=sprintf('U [m/s] at time %0.2f hours',Time2);
Str3=sprintf('U [m/s] at time %0.2f hours',Time3);
Str4=sprintf('T [^oC] at time %0.2f hours',Time1);
Str5=sprintf('T [^oC] at time %0.2f hours',Time2);
Str6=sprintf('T [^oC] at time %0.2f hours',Time3);

%% Restructuring U, T and Z

U_1=U_t1(:,1,1);
U_2=U_t2(:,1,1);
U_3=U_t3(:,1,1);

T_1=T_t1(:,1,1);
T_2=T_t2(:,1,1);
T_3=T_t3(:,1,1);

Z=RC(:,1,1);

for i=2:50;

A=U_t1(:,1,i);  D=U_t2(:,1,i);  E=U_t3(:,1,i);
F=T_t1(:,1,i);  I=T_t2(:,1,i);  J=T_t3(:,1,i);
K=RC(:,1,i);
U_1=[U_1,A];    U_2=[U_2,D];    U_3=[U_3,E];
T_1=[T_1,F];    T_2=[T_2,I];    T_3=[T_3,J];
Z=[Z,K];

end

U_1=U_1.';
U_2=U_2.';
U_3=U_3.';

T_1=T_1.';
T_2=T_2.';
T_3=T_3.';

%% Removing values beneath seafloor

distance= XC;
range=-1*Z;

for ii=1:80;
    w=find(abs(Depth(ii)-range')<5);

    for nn=1:50
        if nn>w(1)
            U_1(nn,ii)=NaN;
            U_2(nn,ii)=NaN;
            U_3(nn,ii)=NaN;
            T_1(nn,ii)=NaN;
            T_2(nn,ii)=NaN;
            T_3(nn,ii)=NaN;
        end
    end
end

%% Figures

figure(1)
colormap(brewermap(21,'RdBu'))

subplot(3,1,1);
k=[-3:0.05:3];
[C h]  = contourf(distance,range,U_1,k);
set(h,'LineColor','none');
hold on
ylabel('Depth [m]'); ylim([800 1000])
c=colorbar; c.Label.String='U [m/s]'; caxis([-2 2]);
hold on
```

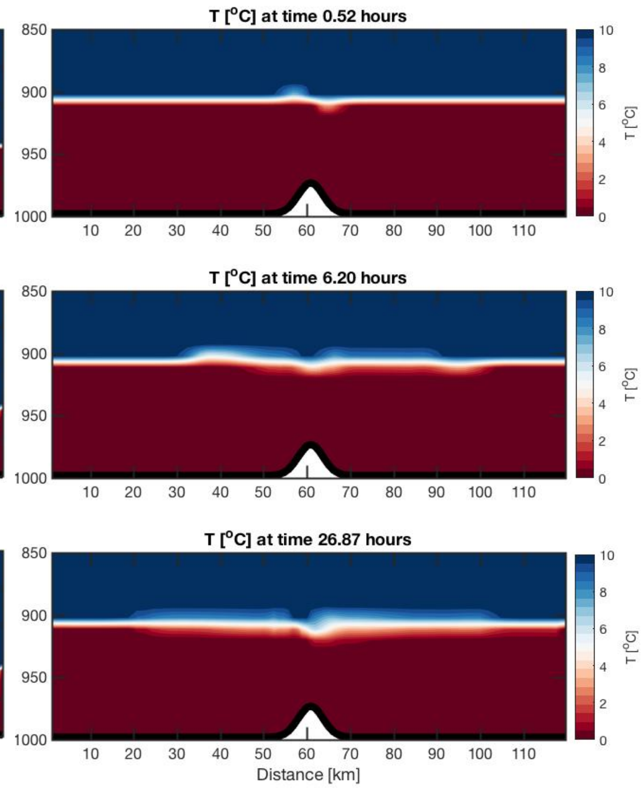
```
p=plot(distance,Depth,'k-','LineWidth',4)
title(Str1);
axis ij
set(gca,'LineWidth',2,'TickLength',[0.025 0.025]); set(gca,'fontsize',16);
subplot(3,1,2)
k=[-3:0.05:3];
[C h]  = contourf(distance,range,U_2,k);
set(h,'LineColor','none');
hold on
ylabel('Depth [m]'); ylim([800 1000])
c=colorbar; c.Label.String='U [m/s]'; caxis([-2 2]);
hold on
p=plot(distance,Depth,'k-','LineWidth',4)
title(Str2);
axis ij
set(gca,'LineWidth',2,'TickLength',[0.025 0.025]); set(gca,'fontsize',16);
subplot(3,1,3)
k=[-3:0.05:3];
[C h]  = contourf(distance,range,U_3,k);
set(h,'LineColor','none');
hold on
xlabel('Distance [km]'); ylabel('Depth [m]'); ylim([800 1000])
c=colorbar; c.Label.String='U [m/s]'; caxis([-2 2]);
hold on
p=plot(distance,Depth,'k-','LineWidth',4)
title(Str3);
axis ij
set(gca,'LineWidth',2,'TickLength',[0.025 0.025]); set(gca,'fontsize',16);

figure(2)
colormap(brewermap(21,'RdBu'))

subplot(3,1,1);
k=[0:0.05:10];
[C h]  = contourf(distance,range,T_1,k);
set(h,'LineColor','none');
hold on
ylabel('Depth [m]'); ylim([800 1000])
c=colorbar; c.Label.String='T [^oC]'; caxis([0 10])
hold on
p=plot(distance,Depth-2,'k-','LineWidth',8)
title(Str4);
axis ij
set(gca,'LineWidth',2,'TickLength',[0.025 0.025]); set(gca,'fontsize',16)
subplot(3,1,2)
k=[0:0.05:10];
[C h]  = contourf(distance,range,T_2,k);
set(h,'LineColor','none');
hold on
ylabel('Depth [m]'); ylim([800 1000])
c=colorbar; c.Label.String='T [^oC]'; caxis([0 10])
hold on
p=plot(distance,Depth-2,'k-','LineWidth',8)
title(Str5);
axis ij
set(gca,'LineWidth',2,'TickLength',[0.025 0.025]); set(gca,'fontsize',16)
subplot(3,1,3)
k=[0:0.05:10];
[C h]  = contourf(distance,range,T_3,k);
set(h,'LineColor','none');
hold on
xlabel('Distance [km]'); ylabel('Depth [m]'); ylim([800 1000])
c=colorbar; c.Label.String='T [^oC]'; caxis([0 10])
hold on
p=plot(distance,Depth-2,'k-','LineWidth',8)
title(Str6);
axis ij
set(gca,'LineWidth',2,'TickLength',[0.025 0.025]); set(gca,'fontsize',16)
```

My results are:

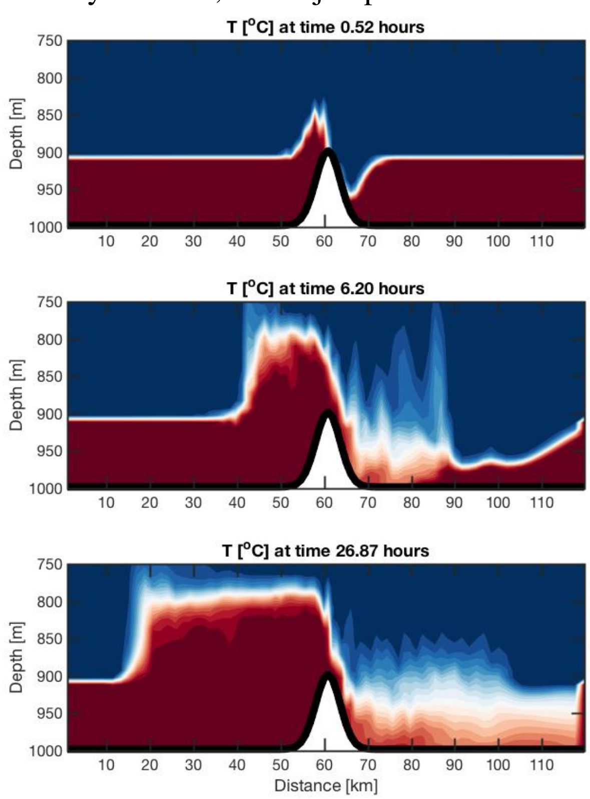
Supercritical:



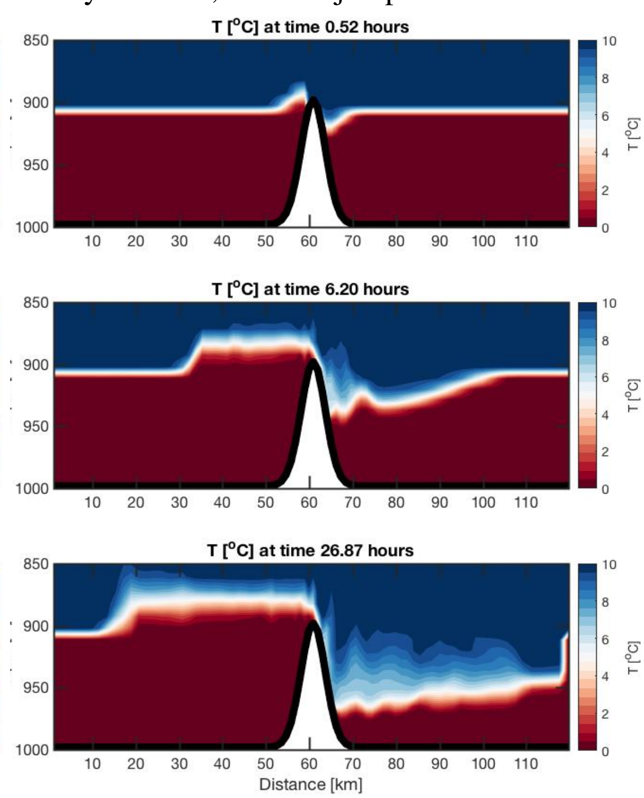
Subcritical:



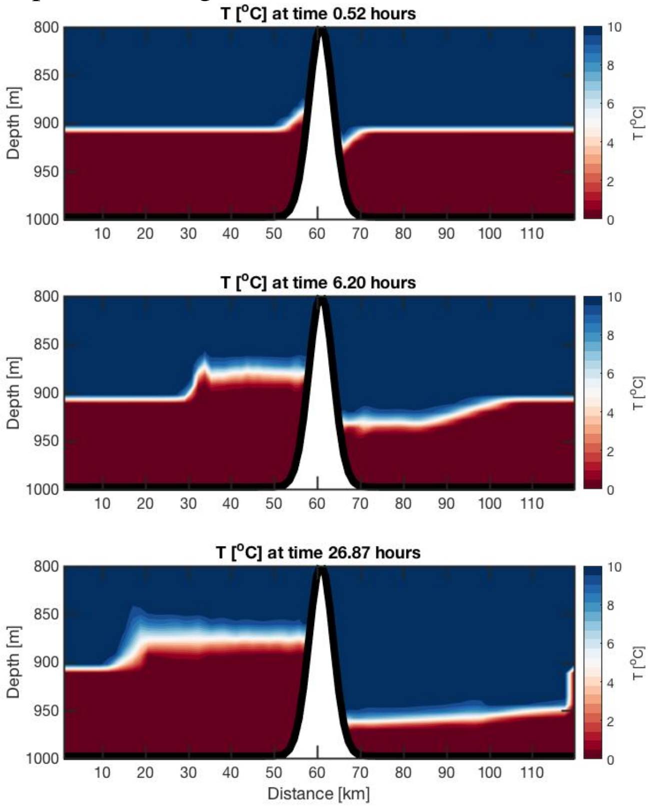
Partially blocked, no lee jump:



Partially blocked, with lee jump:



Complete blocking:



- Supercritical: a huge (erratic) bulge is formed over the bump, however clear instabilities have formed.

- Subcritical: a very small depression forms after 24 hours.

- Partially blocked, no lee jump and with lee jump look similar at 24 hours, with the exception that with no lee jump has a much greater height upstream the obstacle and a much smaller height downstream the bump compared to the with a lee jump.

- Complete blocking: The flow is completely blocked, and moves back upstream.

- The general behavior of the flows in each regime compare well with Figure 2.11 (Baines).