M Hari Prasath

22101021

Turbulence: AE-621A

Assignment-1

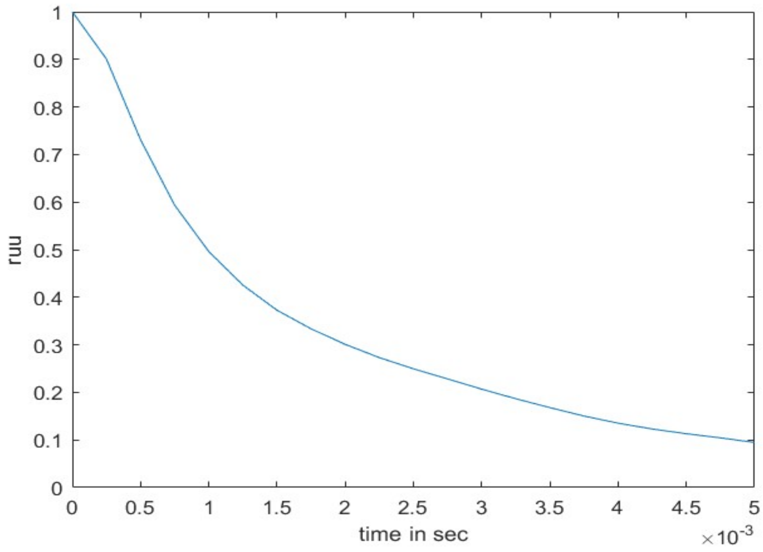
1. The streamwise instantaneous velocity data, as provided in a file, '`hotwire.dat`', were acquired in a turbulent boundary layer using a single hotwire probe. The first column of the data file is time in second (s), and the second column of this file is instantaneous velocity ($\tilde{u}$). Using these data, find the fluctuating velocity, and then plot the correlation function, $R_{uu} = \frac{\overline{u(t)u(t+\Delta t)}}{u_{rms}^2}$, against the time separation and estimate the integral time-scale and the Taylor micro-scale from this plot.

MATLAB Code:

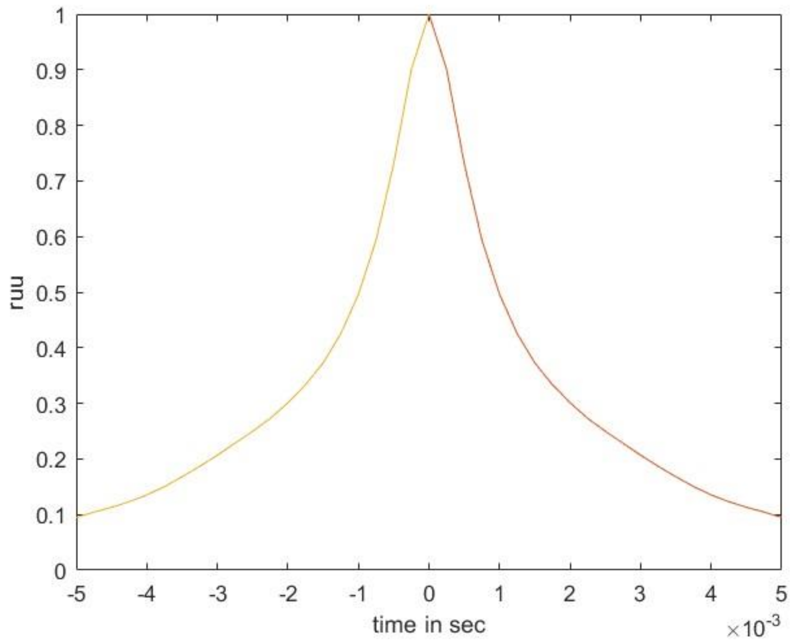
```
%importing the data
hotwire=importdata("hotwire.dat");
%% Getting the velocoties
velocity=hotwire(:,2);
%% finding mean and delt
mu=mean(velocity);
delt=hotwire(2,1)-hotwire(1,1);
%% fluctuating velocity
vdash=velocity-mu;
%% delt between a and b arrays
m=1;
for t=(0:20);
    %specify no of data points n
    n=10000;
    a=vdash(1:n);
    b=vdash(1+t:n+t);
    c=corrcoef(a,b);
    cor(m)=c(1,2);
    x(m)=t*delt;
    m=m+1;
end
%plotting
hold on
plot(x,cor)%% Importing the hot wire data
% integral time scale
intTscal=trapz(x,cor)
%plotting mirror image
plot(-x,cor)
%fitting parabola
%xpara1=[-0.000987 -0.0005 -0.00025 0 0.00025 0.0005 0.000987];
xpara1=[-0.0005 -0.00025 0 0.00025 0.0005];
ypara1=[0.731075 0.900649 1 0.900949 0.731075];
p1=polyfit(xpara1,ypara1,2);
%by solving p1 for y=0 we get taylor microscale in seconds x=0.000987
xpara=[-0.000987 -0.0005 -0.00025 0 0.00025 0.0005 0.000987];
ypara=polyval(p1,xpara);
plot(xpara,ypara)
hold off
taylorscale=0.00987*mu
```

[Note: here I have taken delt from 0 to 20 to get bell shaped graph]

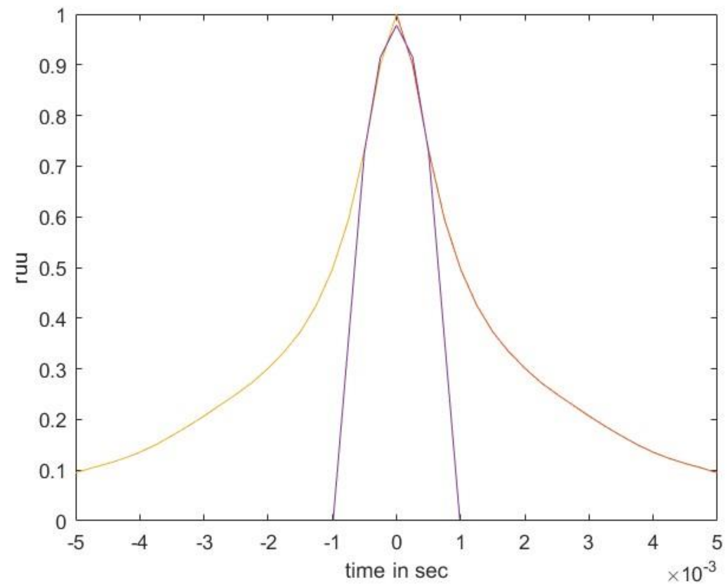
Plots:



ruu vs time for delt=(0 to 20)



After mirroring



After fitting parabola

Results:

Integral time scale=0.0017 sec

Taylor microscale=0.0316 m

2. Instantaneous velocities in the wall-normal and the streamwise plane (400 snapshots) are acquired and are provided in a file, '`PIVdata.dat`'. The first two columns of this file indicate the x and y co-ordinates and the third and the fourth column indicate the corresponding instantaneous velocities, $\tilde{u}$ and $\tilde{v}$, respectively, at those x and y co-ordinates. Using the ensemble average technique, find the components of the Reynolds stresses and show as line plots of these quantities at $x = 15.3$ mm location. Estimate $R_{uu} = \frac{\overline{u(x_0,y_0)u(x_0+\Delta x,y_0+\Delta x)}}{u_{rms}^2}$ considering $x_0 \approx 15.3$ mm and $y_0 \approx 3.9$ mm and plot the correlation contours.

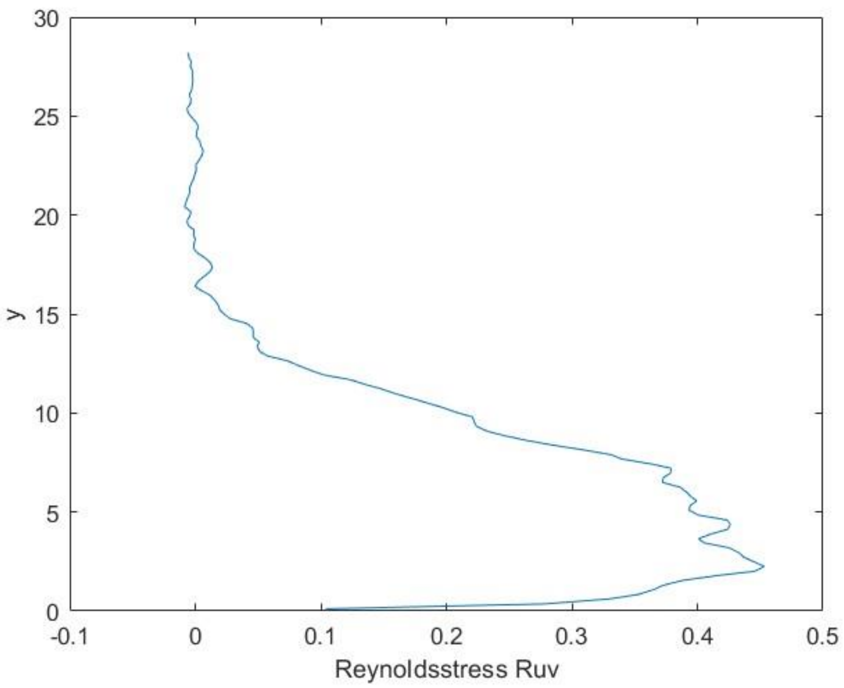
Part A:

MATLAB Code:

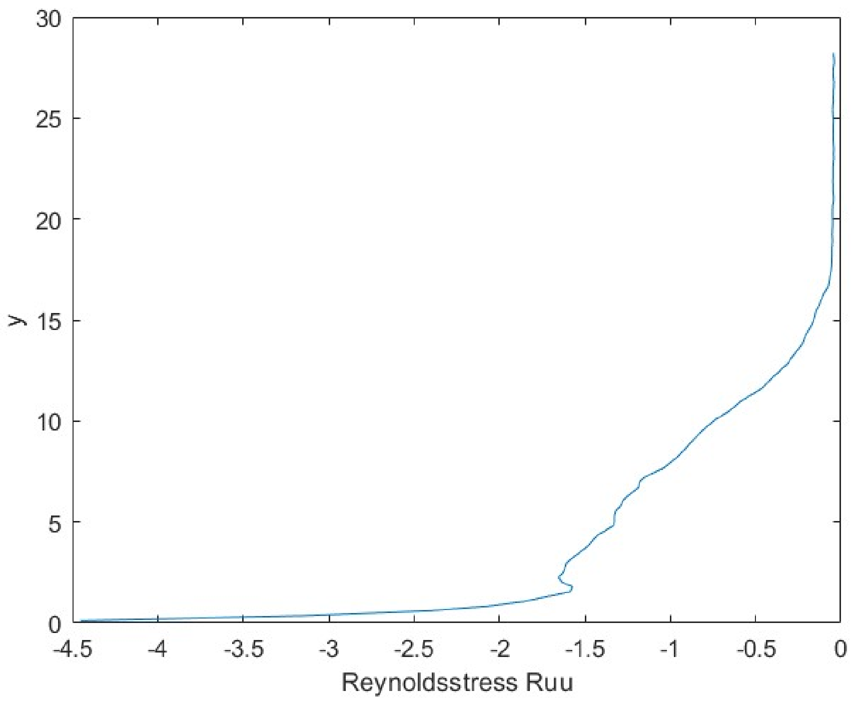
```
pivdata=importdata("PIVdata.dat");
%%geting points on x=15.3mm
x153=zeros([48000 4]);
n=0;
for i=(1:length(pivdata))
    if pivdata(i,1)== 15.30514700
       n=n+1;
       x153(n,:)=pivdata(i,:);

    end
end
x153;
n;
%% Trying to put for loop for whole thing and ploting rey stress for x=15.3 line
for l=(1:120)
%ensemble average
% use 1st and 121th data like wise
x153y1=zeros([400 4]);
m=0;
q=l;
    for j=(1:400)
    m=m+1;
    x153y1(m,:)=x153(q,:);
    q=q+120;
    end
     x153y1;
%      finding fluctuating components of u,v
    x153y1(:,3)=x153y1(:,3)-mean(x153y1(:,3));
    x153y1(:,4)=x153y1(:,4)-mean(x153y1(:,4));

    %now try to find reynilds stress at this point

    for k=(1:400)
        %for uv component
        c(k)=-x153y1(k,3)*x153y1(k,4);
        %now for uu comonent
        c1(k)=-x153y1(k,3)*x153y1(k,3);
```

```
        %now for vv component
        c2(k)=-x153y1(k,4)*x153y1(k,4);
    end
%    c;
    mean(c);
    reynoldsstressuv=mean(c);
    reynoldsstressuu=mean(c1);
    reynoldsstressvv=mean(c2);
    %next code is for l loop
    reystresuv(l)=reynoldsstressuv;
    reystresuu(l)=reynoldsstressuu;
    reystresvv(l)=reynoldsstressvv;
    y(l)=x153y1(1,2);
    q=0;

end
reystresflipuv=flip(reystresuv);
reystresflipuu=flip(reystresuu);
reystresflipvv=flip(reystresvv);
yfliped=flip(y);
% hold on
figure(1);
plot(reystresflipuv,yfliped)
xlabel("Reynoldsstress Ruv")
ylabel("y")
figure(2);
plot(reystresflipuu,yfliped)
xlabel("Reynoldsstress Ruu")
ylabel("y")
figure(3);
plot(reystresflipvv,yfliped)
xlabel("Reynoldsstress Rvv")
ylabel("y")
% hold off
q;
```

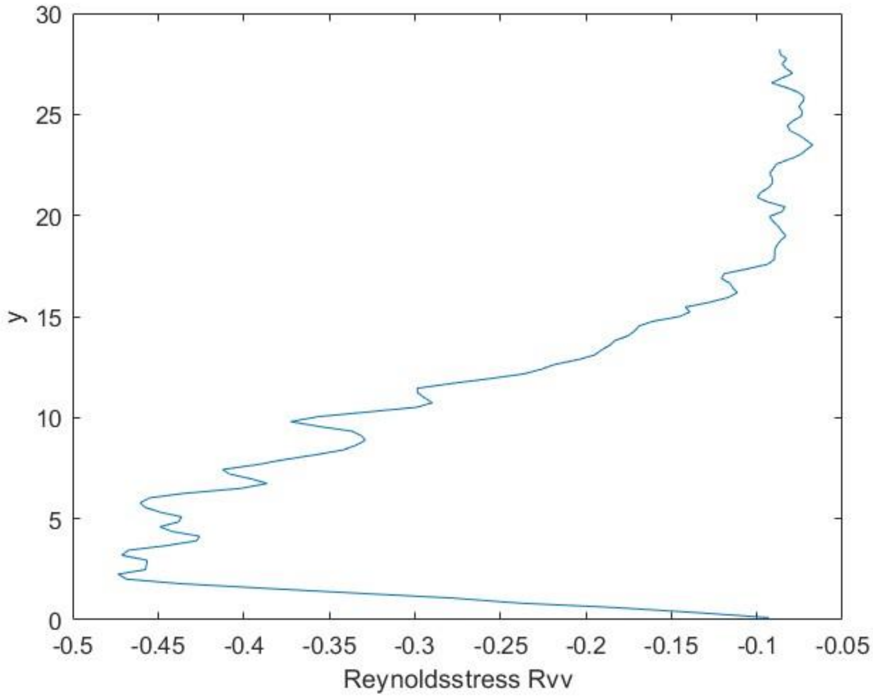
Plots:



Reynolds stress Ruv vs y



Reynolds stress Ruu vs y



Reynolds stress Rvv vs y

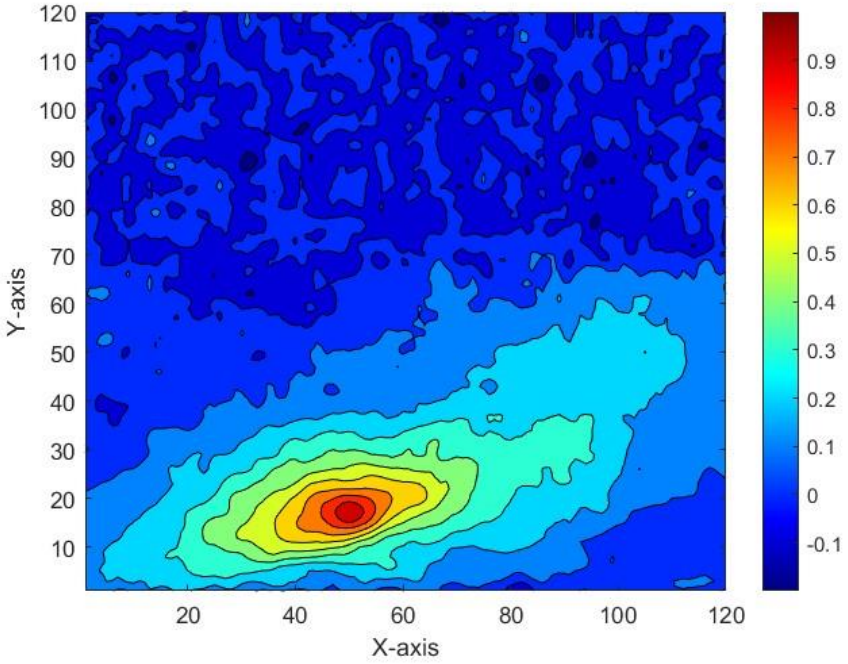
Part B:

MATLAB CODE:

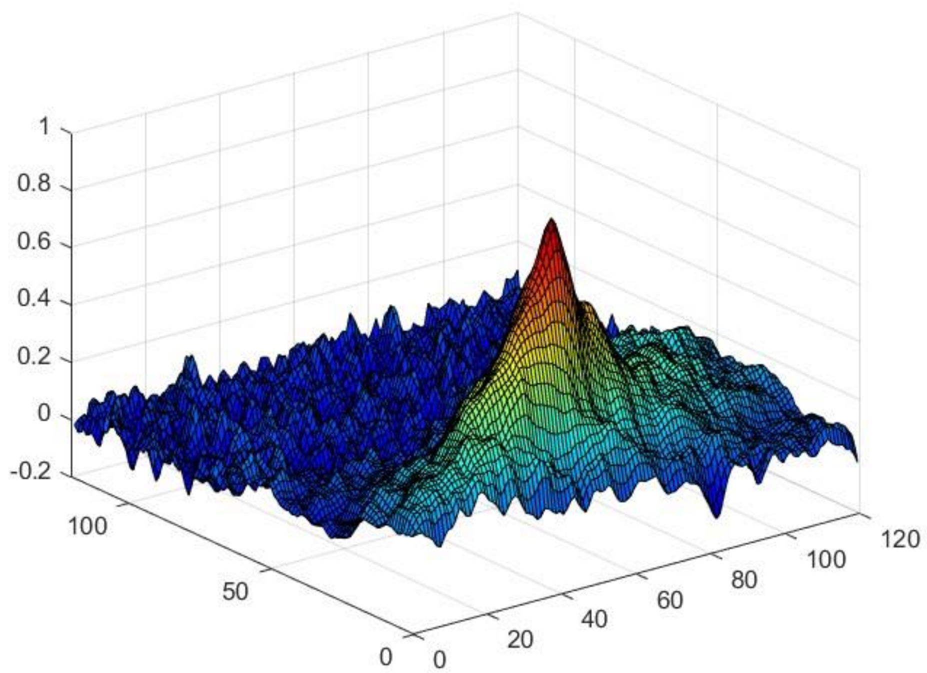
```
clear;
% Try to find 121*120 array of ruu
pivdata=load("PIVdata.dat");
%% getting all datas on x=15.3
x153=zeros([48000 4]);
n=0;
for i=(1:length(pivdata))
    if pivdata(i,1)== 15.30514700
       n=n+1;
       x153(n,:)=pivdata(i,:);

    end
end
x153;
n;
%% we need 400 snaps at x,y=15.3,3.9
x153y39=zeros([400 4]);
n1=0;
for i1=(1:length(x153))
    if x153(i1,2)==3.908869000000000
       n1=n1+1;
       x153y39(n1,:)=x153(i1,:);

    end
end
x153y39;
%% get 14400 ruu
for j=(1:14400)
    p=j;
    for j1=(1:400)
        x(j1,:)=pivdata(p,:);
        p=p+14400;
    end
    cor=corrcoef(x153y39(:,3),x(:,3));
    ruu(j)=cor(1,2);
end
ruu;
transruu=transpose(ruu);
revtransruu=flip(transruu);
matruu=reshape(revtransruu,120,[]);
invmatruu=transpose(matruu);
flipinvmatruu=flip(invmatruu,2);
contourf(flipinvmatruu)
colormap jet
colorbar
xlabel("X-axis");
ylabel("Y-axis");
surf(flipinvmatruu)
```

Plots:

Contour plot:



Surface Plot:



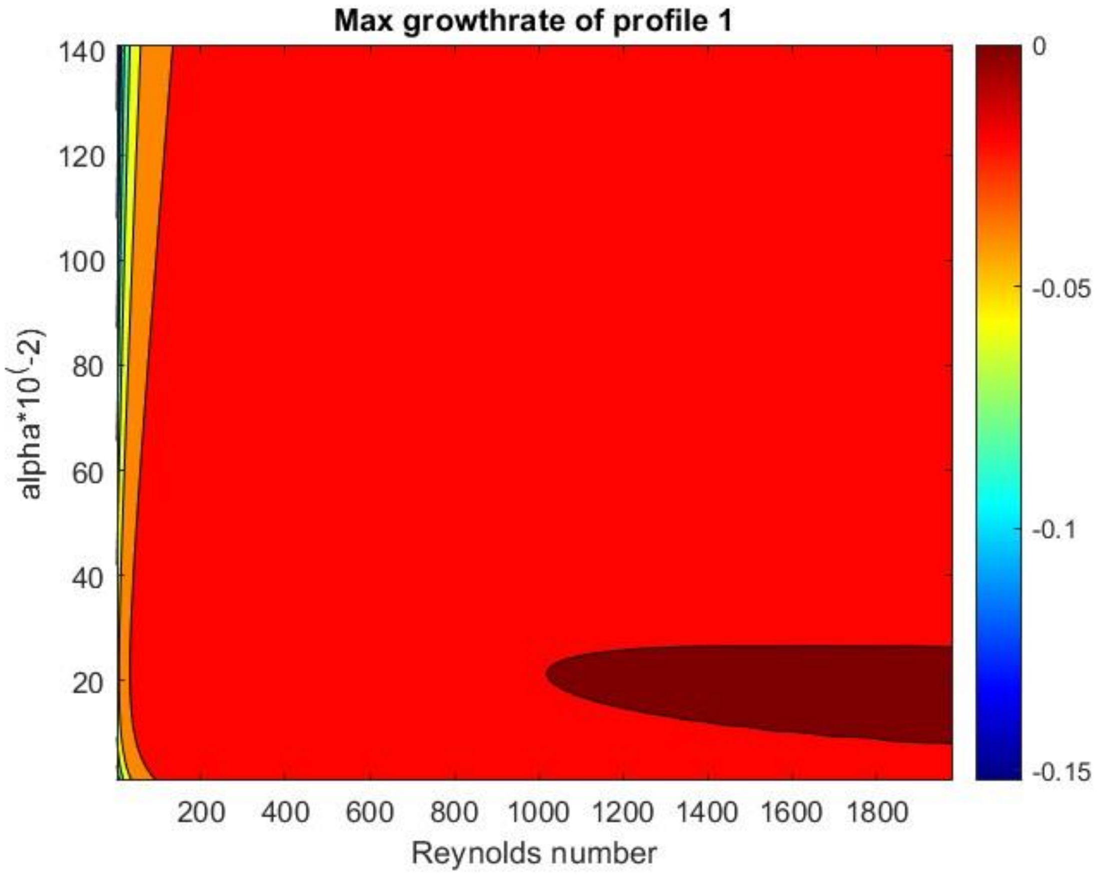
3. Using the data provided in the files, `velocity1.dat` and `velocity2.dat`, find out the most unstable eigenvalues for $\alpha = 0.20$ and $Re = 550$, and compare the absolute value of the eigenfunctions, $|\phi(y)|$, corresponding to the wall-normal disturbance velocity for both of these velocity profiles. Also compare the eigenfunctions corresponding to the streamwise disturbance velocity for these velocity profiles. Plot the neutral stability curves in the plane of α and Re, for these velocity profiles.

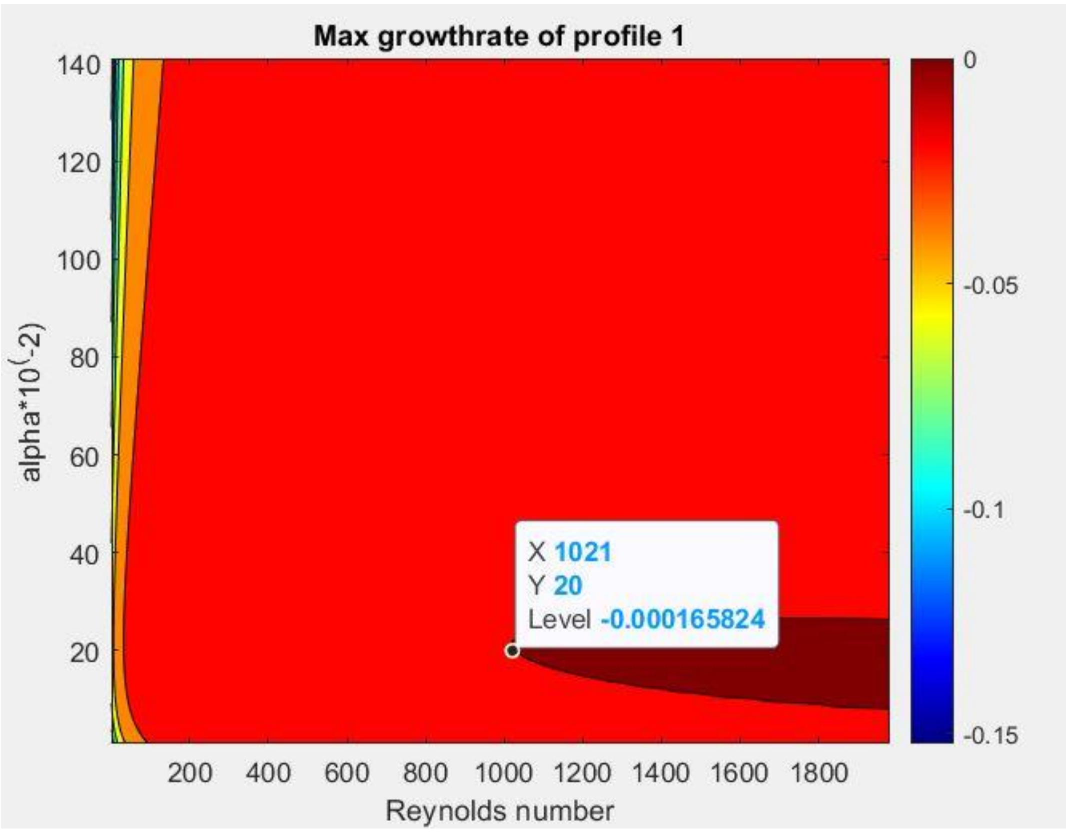
MATLAB Code: Velocity profile 1 and 2

```
clear all;
% File which includes three columns, y U U''
data= load ('velocity1.dat');  % load your velocity profile
%----------------------------------------

% Number of Chebyshev modes
N= 99; N1 = N+1;
j = (0:N)'; X = cos(pi*j/N);
%-------------------------------------

% Scaling for mapping from -1 to 1
L = data(end,1);
scal=L/2; % Physical domain => [0, L]
Y=X*scal + scal; % Mapping from Physical to Chebyshev domain
%----------------------------------

% Velocity and its second derivative are being evaluated at scaled Y
U = interp1(data(:,1), data(:,2), Y);
U2 = interp1(data(:,1), data(:,3), Y);
fac = 1/scal;
%-------------------------------------

%%For singel value of alpha and Re, for example,
% alpha = 0.3;
% length(alpha)
% Rey = 600;
%%-------------------------------

% For a range of alpha and Reynolds number, for example,
 alpha = 0.1:0.01:1.5;
 length(alpha)
 Rey = 10:0.5:1000;
%-----------------------------------------

% Main program; try to understand before you run it
ci = zeros(length(alpha),length(Rey));
cr = zeros(length(alpha),length(Rey));
for ii = 1:length(alpha)
    ii
    for jj = 1:length(Rey)
        al = alpha(ii);
        R = Rey(jj);
        zi = sqrt(-1); a2 = al^2; a4 = a2^2; er = -200*zi;
```

```
        [D0,D1,D2,D3,D4] = Dmat(N); % Read about it in the book.
        D1 = fac*D1;
        D2 = (fac^2)*D2;
        D4 = (fac^4)*D4;
        B = (D2-a2*D0);
        A = (U*ones(1,N1)).*B-(U2*ones(1,N1)).*D0-(D4-2*a2*D2+a4*D0)/(zi*al*R);
        A = [er*D0(1,:); er*D1(1,:); A(3:N-1,:); er*D1(N1,:); er*D0(N1,:)];
        B = [D0(1,:); D1(1,:); B(3:N-1,:); D1(N1,:); D0(N1,:)];
        d = (inv(B)*A);
        [vv, c] = eig(d);
        % eigenvalues are being evaluated using eig function
        cdiag=diag(c);
        mxci = max(imag(cdiag));  % useful for contour plots
         ci(ii,jj)= mxci;
%          cr(ii,jj) =real(c(I));

    end;
end
figure(1)
contourf(ci)
colormap jet
colorbar
xlabel("Reynolds number")
ylabel("alpha*10^(-2)")
figure(2)
surf(ci)
```

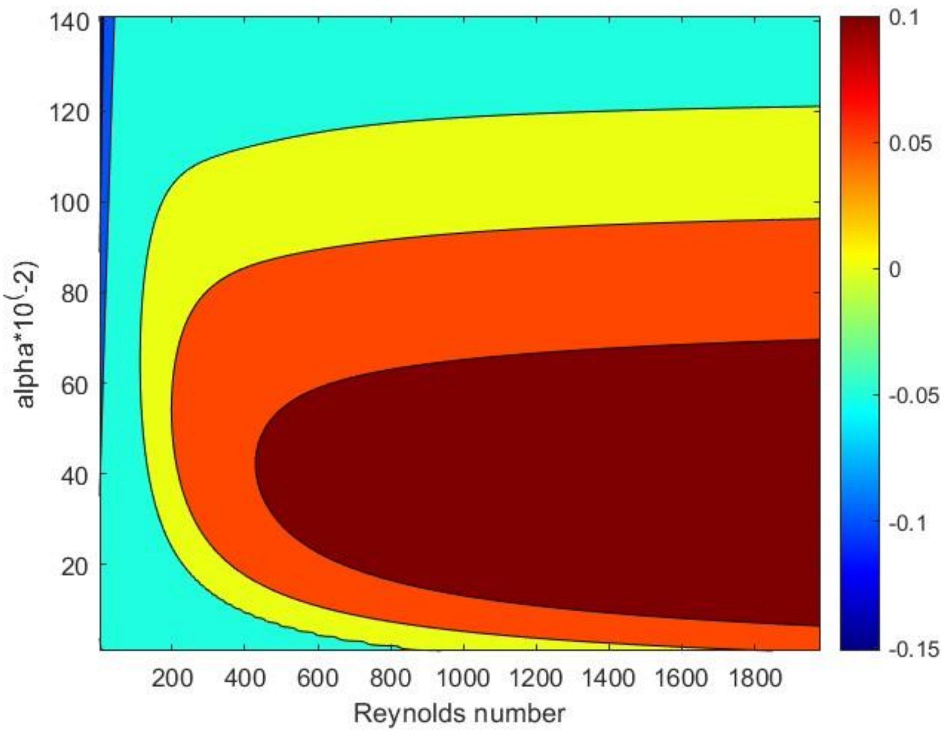
For velocity profile1



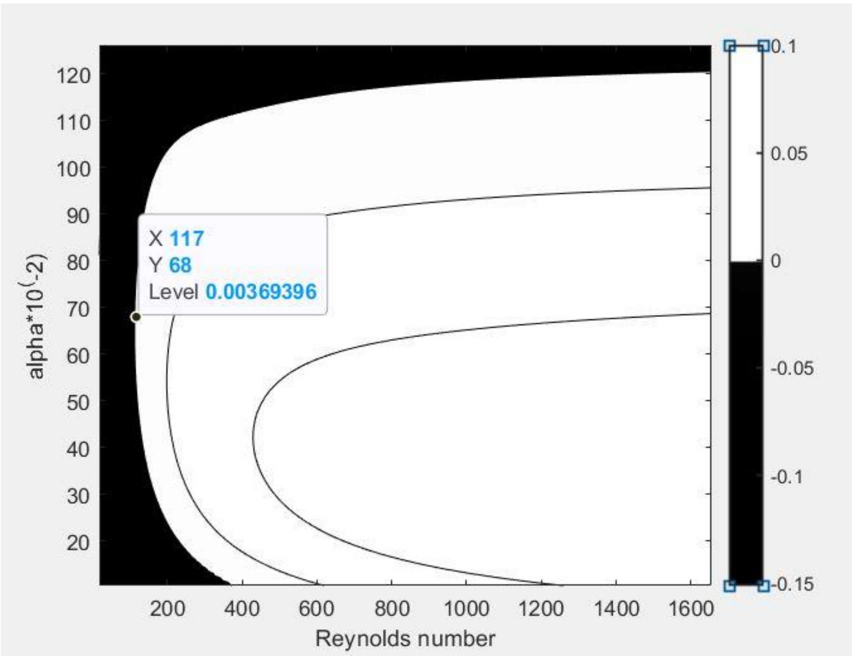


Re critical=1021

For velocity profile2:



Reynolds number vs alpha



Re critical = 117

Results:

For velocity profile 1 at Re=550, alpha=0.2:

Unstable eigenvalue =-0.0012

For velocity profile 2 at Re=550, alpha=0.2:

Unstable eigenvalue =0.0941