Assignment -1

Numerical solution of Orr Sommerfeld equation for plane Poiseuille flow

Submitted by
Vikas Dwivedi

Formulation

Governing equation is

$$(\overline{U} - c)(\partial_{yy} - k^2)\hat{\psi} - \partial_{yy}\overline{U}\hat{\psi} = \frac{1}{ikRe}(\partial_{yy} - k^2)^2\hat{\psi}$$

Base flow is

$$\overline{U} = 1 - y^2$$

BCs are

$$\hat{\psi}(-1) = \hat{\partial_y \psi}(-1) = 0 \tag{0.1}$$

$$\hat{\psi}(-1) = \hat{\partial_y \psi}(-1) = 0 \tag{0.2}$$

Chebyschev polynomials are:

$$T_n(y) = cos(n(cos^{-1}y)) \tag{0.3}$$

These polynomials satisfy the following recursive relationship:

$$T_{n+1}(y) = 2yT_n(y) - T_{n-1}(y) \tag{0.4}$$

$$T_n^k(y) = 2nT_{n-1}^{k-1}(y) - \left(\frac{n}{n-2}\right)T_{n-2}^k(y) \tag{0.5}$$

On substituting $\hat{\psi}$ in terms of Chebyschev polynomials

$$\hat{\psi(y)} = \sum_{n=0}^{N-1} a_n T_n(y) \tag{0.6}$$

$$\hat{\psi(y)}^{(k)} = \sum_{n=0}^{N-1} a_n T_n(y)^{(k)} \tag{0.7}$$

at spectral collocation points

$$yC_j = cos\left(\frac{j\Pi}{N-1}\right) \tag{0.8}$$

and using the following property of Chebyschev functions at boundaries,

$$T_n(+1) = (+1)^n$$

$$T_n(-1) = (-1)^n$$

$$T_n'(+1) = (+1)^{n-1}n^2$$

$$T_n'(-1) = (-1)^{n-1}n^2$$

the Orr Sommerfeld equation reduces to an algebraic system of N equations and N unknown Chebyschev coefficients.

This system can be written as

$$Aa = cBa \tag{0.9}$$

where c is the complex eigen value and a is the vector of unknown Chebyschev coefficients.

"OSE_Poiseuille.cpp" & *"figs.m"* contain the numerical solution of this eigen value poblem. An open source linear algebra library 'Armadillo' has been used in the C++ code. We solved the eigen value problem and finally plotted the eigen values using open source matlab clone: OCTAVE.

Output

In Figure 0.1, there is one slightly unstable mode on the A branch, even though parabolic base flow doesn't contain inflection point. This confirms that effect of viscous terms is destabilizing. This unstable mode is called **Tollmein-Schlichting** wave.

For the same pair (k, Re), other eigenmodes are stable with positive speed and with discrete eigenvalues arranged along well-defined branches in the (C_r, C_i). Eigen modes for these branches are shown in Figure [0.3-0.5].

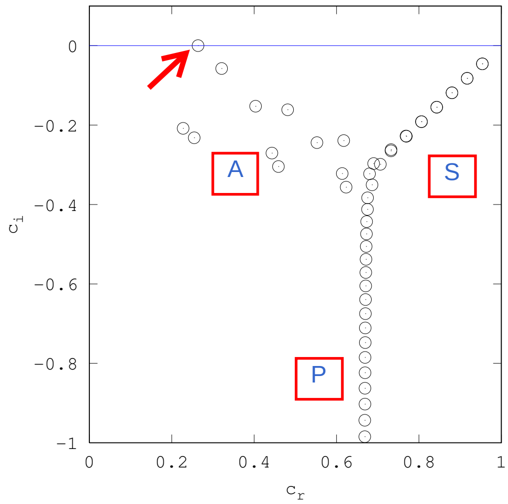

Figure 0.1: Orr Sommerfeld spectrum for plane Poiseuille flow at k=1.02, Re=5772

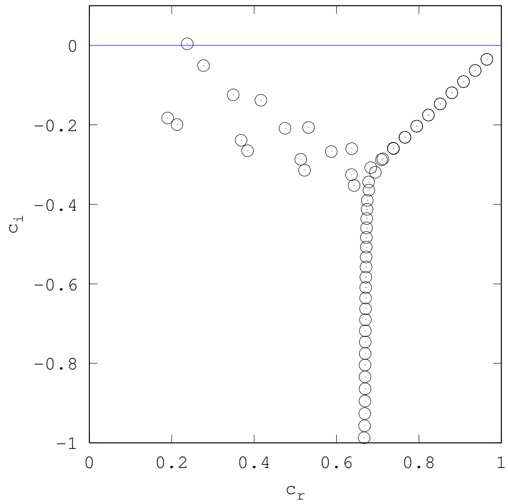

Figure 0.2: Orr Sommerfeld spectrum for plane Poiseuille flow at k=1, Re=10,000

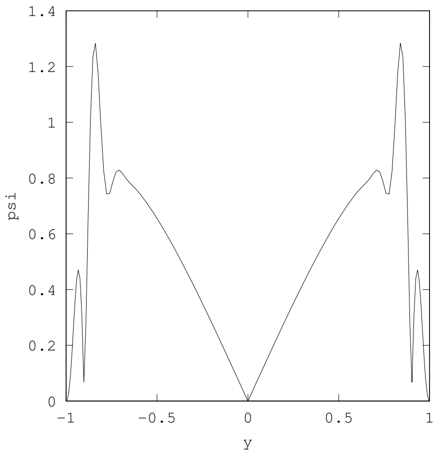

Figure 0.3: A- mode at k=1, Re=10,000

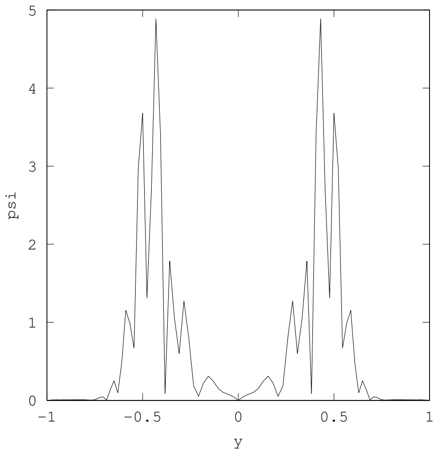

Figure 0.4: S- mode at k=1, Re=10,000

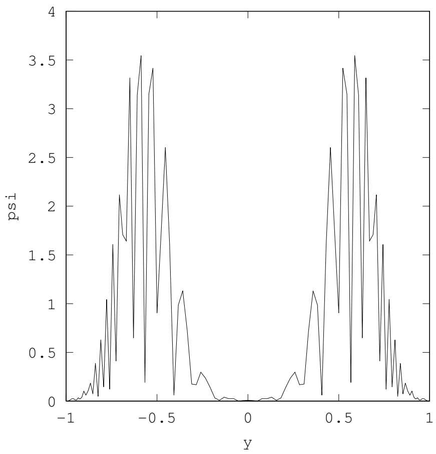

Figure 0.5: P- mode at k=1, Re=10,000

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <armadillo>
using namespace std;
using namespace arma;
#define pi 3.14159265
int main ()
{
char const* ch1="Ar";
char const* ch2="Ai";
char const* ch3="Br";
char const* ch4="Bi";
int    N=121;
double R=10000.0;
double K=1.0;
double  yC;
double   U;
double  dU;
double d2U;


mat cheb    = zeros<mat>(5,N)   ;// contains Chebyschev polynomials and derivs
cx_mat A    = zeros<cx_mat>(N,N);
cx_mat B    = zeros<cx_mat>(N,N);
cx_mat iota = zeros<cx_mat>(1,1);
iota=cx_double(0,1);

// Spectral collocation points yC=cos(m*pi/(N-1)) for m=0..N-1
for (int m=N-2;m>1;m--)
{
 yC = cos(m*pi/(N-1));


 // Cheb Poly @ yC

 cheb(0,0)=1.0;
 cheb(0,1)=yC;

 for (int jj=1; jj <N-1;jj++)
     {
      cheb(0,jj+1)=2.0*yC*cheb(0,jj)-cheb(0,jj-1);
     }

// Cheb derivs @ yC
 for (int ii=1; ii<5; ii++)
     {
      cheb(ii,0)=0.0;
      cheb(ii,1)=cheb(ii-1,0);
      cheb(ii,2)=4.0*cheb(ii-1,1);

      for (int jj=3;jj<N;jj++)
          {
           cheb(ii,jj)=2.0*jj*cheb(ii-1,jj-1)+(jj)/(jj-2.0)*cheb(ii,jj-2);
          }
     }

// Base flow @ yC
U   = 1.0-pow(yC,2);
dU  =-2.0*yC;
d2U =-2.0;
// A and B of Aa=cBa
for (int jj=0;jj<N;jj++)
    {
      A(N-m-1,jj)=U*(cheb(2,jj)-(K*K)*cheb(0,jj))-d2U*cheb(0,jj)+(iota(0,0)/(K*R))*(cheb
(4,jj)-2.0*K*K*cheb(2,jj)+(pow(K,4))*cheb(0,jj));
      B(N-m-1,jj)=   cheb(2,jj)-(K*K)*cheb(0,jj);
    }
}
// BCs
for (int j=0;j<N;j++ )
{
```

```
A(0,j)  = pow(+1,j);
A(1,j)  = pow(+1,j-1)*pow(j,2);
A(N-2,j)= pow(-1,j-1)*pow(j,2);
A(N-1,j)= pow(-1,j);
B(0,j)  = 0.0;
B(1,j)  = 0.0;
B(N-2,j)= 0.0;
B(N-1,j)= 0.0;
}
/*
cx_vec eigval = eig_pair( A, B );
cout<<eigval<<endl;
*/
char fname1[50];
sprintf(fname1, "/home/vikas/Desktop/hw_5430/%s",ch1);
ofstream fout1;
fout1.open(fname1);

char fname2[50];
sprintf(fname2, "/home/vikas/Desktop/hw_5430/%s",ch2);
ofstream fout2;
fout2.open(fname2);

char fname3[50];
sprintf(fname3, "/home/vikas/Desktop/hw_5430/%s",ch3);
ofstream fout3;
fout3.open(fname3);

char fname4[50];
sprintf(fname4, "/home/vikas/Desktop/hw_5430/%s",ch4);
ofstream fout4;
fout4.open(fname4);


 for(int i = 0; i < N; i++)//row wise
    {
     for(int j = 0; j < N; j++)
         {

          fout1.precision(18);
          fout1 << real(A(i,j));
          fout1 << "\n";

          fout2.precision(18);
          fout2 << imag(A(i,j));
          fout2 << "\n";

          fout3.precision(18);
          fout3 << real(B(i,j));;
          fout3 << "\n";

          fout4.precision(18);
          fout4 << imag(B(i,j));
          fout4 << "\n";

         }

    }

fout1.close();
fout2.close();
fout3.close();
fout4.close();


return 0;

}
```

```
% Read data:
file1  = '/home/vikas/Desktop/hw_5430/Ar';
file2  = '/home/vikas/Desktop/hw_5430/Ai';
file3  = '/home/vikas/Desktop/hw_5430/Br';
file4  = '/home/vikas/Desktop/hw_5430/Bi';


Ar = load(file1,'-ascii');
Ai = load(file2,'-ascii');
Br = load(file3,'-ascii');
Bi = load(file4,'-ascii');

N  = sqrt(length(Ar));

AR =zeros(N,N);
AI =zeros(N,N);
BR =zeros(N,N);
BI =zeros(N,N);
for i=1:N
    for j=1:N
        AR(i,j)=Ar((i-1)*N+j);
        AI(i,j)=Ai((i-1)*N+j);
        BR(i,j)=Br((i-1)*N+j);
        BI(i,j)=Bi((i-1)*N+j);
    end
end
i=sqrt(-1);
A = AR+i*AI
B = BR+i*BI

% find eigenvalues
[V,D]=eig(A,B);
for j=1:N;
real_c(j)=real(D(j,j));
imag_c(j)=imag(D(j,j));
end
%--Eigen Modes--%
psi_hat=zeros(1,N);
y = linspace(-1,1,N);
for m=N-2:-1:2;
yC=cos(m*pi/(N-1));
cheb=zeros(1,N);% contains Chebyschev polynomial and derivs
% Cheb Poly @ yC
cheb(1,1)=1.0;
cheb(1,2)=yC;
for jj=2:N-1;
cheb(1,jj+1)=2.0*yC*cheb(1,jj)-cheb(1,jj-1);
end
s=0;
for j=1:N
s = s + cheb(1,j)*real(V(j,58));% P:56, A:58, S:102
end
psi_hat(m)=s;
end

% plot
f1 = figure(1);
W = 4; H = 4;
set(f1,'PaperUnits','inches');set(f1,'PaperOrientation','portrait');
set(f1,'PaperSize',[H,W])    ;set(f1,'PaperPosition',[0,0,W,H]);
plot(real_c,imag_c,'ko',[0 1],[0 0]);axis([0 1 -1 0.1])
xlabel('c_{r}');ylabel('c_{i}');
print(f1,'-deps','-color','eig_vals.eps');

f2 = figure(2);
W = 4; H = 4;
set(f2,'PaperUnits','inches');set(f2,'PaperOrientation','portrait');
set(f2,'PaperSize',[H,W])    ;set(f2,'PaperPosition',[0,0,W,H]);
plot(y,abs(psi_hat),'k');% axis([0 1 -1 0.1])
xlabel('y');ylabel('psi');
print(f2,'-deps','-color','eig_modes.eps');
```