OCE680 Assignment 3, Instructor's notes

1: Numerical analysis of shear instability [10]

Here is my function that solves the Rayleigh equation in matrix form. You can choose impermeable or asymptotic boundary conditions. The function also outputs growth rate, vertical velocity eigenfunction, and various functions from the energy budget (needed later). These outputs pertain to either the fastest-growing mode or some other mode as selected via the input parameter `nmode`.

```
function [sig1,w1,uw1,SP1,EF1,K1,p1]=Ray(z,U,k,l,iBC,nmode)
%
% MINIMAL USAGE: [sig1]=Ray(z,U,k)
%    FULL USAGE: [sig1,w1,uw1,SP1,EF1,K1]=Ray(z,U,k,l,iBC,nmode)
%
% Stability analysis for inviscid, homogeneous, parallel shear flow
% (Rayleigh equation)
%
% INPUTS:
% z = vertical coordinate vector (evenly spaced)
% U = velocity profile
% k,l = wave vector components [default l=0]
% iBC = [1BC(1) iBC(2)]: boundary condition choice at z(0) and z(N+1), resp.
%   1 for impermeable boundary; 2 for asymptotic boundary; [default iBC=[1 1]]
% nmode = mode selection in terms of growth rate:
%       1 = fastest-growing mode [default nmode=1]
%       2 = second-fastest mode, etc.
%
% OUTPUTS:
% sig1 = complex growth rate
% w1 = vertical velocity eigfn
% uw1, SP1 = Reynolds stress and shear production
% EF1 = energy flux
% K1 = perturbation kinetic energy
%
% W. Smyth, Sep 2004, Jan 2012, Feb 2016

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 1: Preliminaries
%

% check for equal spacing
if abs(std(diff(z))/mean(diff(z)))>.000001
    disp(['ddz2: values not evenly spaced!'])
    sig1=NaN;
    return
end

% defaults
if nargin<4; l=0;end        % l=0
if nargin<5; iBC=[1 1];end      % impermeable BCs
if nargin<6; nmode=1;end    % choose FGM

% define constants
ii=complex(0.,1.);
del=z(2)-z(1);
N=length(z);
kt=sqrt(k^2+l^2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 2: Set up derivative matrices
```

```
%
D2=ddz2(z); % 2nd derivative matrix with 1-sided boundary terms
Uzz=D2*U;   % Uzz is computed BEFORE BCs are applied.

% Boundary conditions
% First assume impermeable boundaries
D2(1,:)=0;D2(1,1)=-2/del^2;D2(1,2)=1/del^2;
D2(N,:)=0;D2(N,N)=-2/del^2;D2(N,N-1)=1/del^2;
% Change to asymptotic boundaries if requested
if iBC(1)==2
        D2(1,:)=0;D2(1,1)=-2*(1+del*kt)/del^2;D2(1,2)=2/del^2;
end
if iBC(2)==2
    D2(N,:)=0;D2(N,N)=-2*(1+del*kt)/del^2;D2(N,N-1)=2/del^2;
end
% 2nd derivative matrix complete

% Laplacian matrix
L=D2-kt^2*eye(N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 3: Set up stability matrices,
%          solve eigval problem, sort results
%

% Set up arrays for eigenvalue analysis.
A=L;
B=-ii*k*diag(U)*A+ii*k*diag(Uzz);

% Solve eigenvalue problem.
[w,S]=eig(B,A);
s=diag(S);

% Sort eigvals and eigvecs by real growth rate
[sr,ind]=sort(real(s),1,'descend');
sigma=s(ind);
w=w(:,ind);

% Save the mode selected via nmode
sig1=sigma(nmode);w1=w(:,nmode);

% Normalize by the value at the max of abs(w). This works better than using
% the value at z=0, which may turn out to be zero.
cnorm=w1(abs(w1)==max(abs(w1)));
w1=w1/cnorm;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 4 (optional): Compute extra profiles if requested (1-dimensional!)
%

if nargout<3;return;end
% Compute horizontal velocity eigenfunction and perturbation kinetic energy
wz1=ddz(z)*w1;
u1=(ii/k)*wz1;
K1=(u1.*conj(u1)+w1.*conj(w1))/4;

% Compute shear production terms
uw1=real(u1.*conj(w1))/2;
Uz=ddz(z)*U;
SP1=-uw1.*Uz;
```

```
% Compute pressure eigfn and energy flux
p1=-(sig1+ii*k*U).*wz1/k^2 +(ii/k)*Uz.*w1;
EF1=real(p1.*conj(w1))/2;
return
end
```

Here is the “wrapper” that defines the problem and calls the Rayleigh subroutine.

```
% Test Rayleigh equation solver in 2nd order FD form
clear
close all
direc='/Users/smyth/Dropbox/HOME/Projects/Book_Instability/figures/';

% plotting parameters
fs=18;
lw=2;
ms=18;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OC680 HMWK 3
% Project 1: tanh test

% define wave vector
k=.47;l=0;

% define grid
dz=0.2;
ztop=4;
zbot=-ztop;
z=[zbot+dz:dz:ztop-dz]'; % NOTE: BOUNDARIES ARE EXCLUDED

% define background flow
U=tanh(z);

% compute growth rate, eigenfunction
[sg,w]=Ray(z,U,k,l);

% plot results
figure
subplot(1,3,1)
plot(real(w),z,'linewidth',lw)
hold on
plot(imag(w),z,'linewidth',lw)
legend('w_r','w_i')
legend boxoff
xlabel('w','fontsize',fs)
ylabel('z/h','fontsize',fs)
title(...
   sprintf('FGM: \\sigma = %.4f u_0/h',sg),...
   'fontsize',fs-2,'fontweight','normal')
set(gca,'fontsize',fs-2)

subplot(1,3,2)
plot(abs(w),z,'linewidth',lw)
xlabel('|w|','fontsize',fs)
set(gca,'yticklabel','','fontsize',fs-2)

subplot(1,3,3)
```

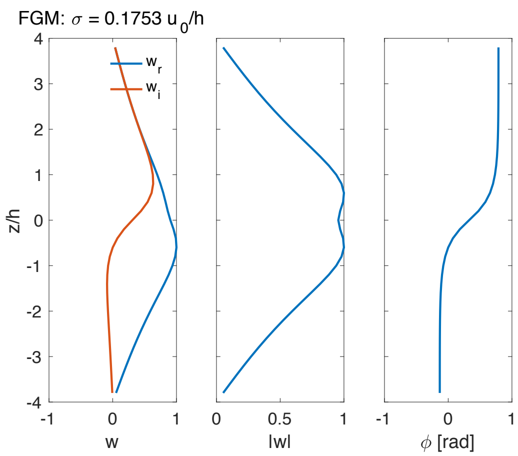
```
plot(phase(w),z,'linewidth',lw)
xlabel('\phi [rad]','fontsize',fs)
set(gca,'yticklabel','','fontsize',fs-2)

print('-dpdf',[direc 'hmwk3_1'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 14.11 shows the growth rate and eigenfunction I got for the given parameter values.

Figure 14.11: Growth rate and eigenfunction for a shear instability. (a) Real and imaginary parts. (b) Magnitude. (c) Phase.



2: The piecewise-linear shear layer: numerical solution [20]

(a) [10] Reproduce analytical solution as in lecture or text with algebraic details included.

(b) [7], **(c)** [3]

Here are results for four numerical solutions of the piecewise linear shear layer, along with the analytical solution (yellow). By using the asymptotic boundary conditions (red dashes) you reproduce the analytical solution very closely. With impermeable boundary conditions, you get terrible results (red) unless you move the boundaries far away from the shear layer (blue, black).

Below is my plot of eigenfunctions. The red solid curve is the analytical result; the blue dots are the numerical result. Note that both are normalized so that $\hat{w}(0) = 1$. Remember that eigenfunctions are only defined up to a multiplicative constant, so two results can both be right but look totally different. Normalizing solves that.

Here is the code I used:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Project 2: PL shear layer

% wave vector range
ks=[0:.025:.7]; l=0;

figure
% plot analytical formula
s_an=sqrt(-( (ks-.5).^2-.25*exp(-4*ks)) );
plot(ks,s_an,'y-','linewidth',6);hold on
xlabel('k*','fontsize',fs+2)
ylabel('\sigma*','fontsize',fs+2)
title('PL Shear Layer','fontsize',fs,'fontweight','normal')

%%%%%%%%%
% case 1: ztop=3; impermeable bcs
% set up z grid, BCs, U
ztop=3; zbot=-ztop;
dz=.1;
```

```
z=[zbot+dz:dz:ztop-dz]'; % exclude boundary points
iBC=[1 1]; % impermeable boundaries

% background velocity profile
U=z; U(U>1)=1; U(U<-1)=-1;

% stability analysis over range of k
clear sig
for i=1:length(ks)
    [sig(i)]=Ray(z,U,ks(i),l,iBC);
end
plot(ks,real(sig),'r.','markersize',ms);hold on

%%%%%%%%%
% case 2: ztop=6; impermeable bcs
% set up z grid, BCs, U
ztop=6; zbot=-ztop;
dz=.2;
z=[zbot+dz:dz:ztop-dz]'; % exclude boundary points
iBC=[1 1]; % impermeable boundaries

% background velocity profile
U=z; U(U>1)=1; U(U<-1)=-1;

% stability analysis over range of k
clear sig
for i=1:length(ks)
    [sig(i)]=Ray(z,U,ks(i),l,iBC);
end
plot(ks,real(sig),'b.','markersize',ms);hold on

%%%%%%%%%
% case 3: ztop=10; impermeable bcs
% set up z grid, BCs, U
ztop=10; zbot=-ztop;
dz=.2;
z=[zbot+dz:dz:ztop-dz]'; % exclude boundary points
iBC=[1 1]; % impermeable boundaries

% background velocity profile
U=z; U(U>1)=1; U(U<-1)=-1;

% stability analysis over range of k
clear sig
for i=1:length(ks)
    k=ks(i);
    [sig(i)]=Ray(z,U,k,l,iBC);
end
plot(ks,real(sig),'k.','markersize',ms);

%%%%%%%%%
% case 4: ztop=3; asymptotic bcs
ztop=3;
zbot=-ztop;
dz=.2;
z=[zbot+dz:dz:ztop-dz]'; % exclude boundary points
iBC=[2 2]; % asymptotic boundaries

% background velocity profile
U=z; U(U>1)=1; U(U<-1)=-1;
```

```
% stability analysis over range of k
clear sig
for i=1:length(ks)
    [sig(i)]=Ray(z,U,ks(i),l,iBC);
end
plot(ks,real(sig),'r-','linewidth',2);hold on
ylim([0 .21])
set(gca,'fontsize',fs-2)

h=legend('analytical','z_T=3  Impermeable','z_T=6  Impermeable','z_T=10 Impermeable','z_T=3  Asymptotic','locatio
legend boxoff
set(h,'fontsize',fs-2)


print('-dpdf',[direc 'hmwk3_2'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% compare eigenfunctions

% Numerical eigfn from asymptotic case
% choose wavenumber of FGM
kt_mx=ks(real(sig)==max(real(sig))); % FGM
k=kt_mx;l=0;

iBC=[2 2];
[~,w]=Ray(z,U,k,l,iBC);

% normalize numerical eigenfunctions so that w(0)=1.
wnorm=w(z==0);
w=w/wnorm;
%u=1i*ddz(z)*w/k;

% Theoretical eigfn (I cheated and solved numerically for the
% coefficients B,C,D,E).
kt_mx=0.40;
k=kt_mx;

s_an=sqrt(-( (k-.5).^2-.25*exp(-4*k)) );
cmx_an=1i*s_an/k;
%
% option 1
x=exp(k);xi=1/x;c=cmx_an
M=[xi -x -xi 0; ...
    0 xi x -xi;...
    (1/(k*(1-c))-1)*xi -x xi 0;...
    0 xi -x (1/(k*(1+c))-1)*xi]
V=null(M);
w_an=V(2)*exp(k*z)+V(3)*exp(-k*z);
w_an(z>1)=V(1)*exp(-k*z(z>1));
w_an(z<-1)=V(4)*exp(k*z(z<-1));

% option 2
x=exp(-2*k);
MM=[-2*(1-c)+1/k  x/k; ...
    -x/k    2*(1+c)-1/k];
V=null(MM);
w_an=V(1)*exp(-k*abs(z-1))+V(2)*exp(-k*abs(z+1));
```

```
% normalize analytical eigenfunctions so that w(0)=1.
wnorm=w_an(z==0);
w_an=w_an/wnorm;

% plot w

figure
subplot(1,10,1:2)
plot(U,z,'linewidth',lw);
axis tight;xlim([-1.1 1.1])
title('(a) U_ ','fontsize',fs,'fontweight','normal')
ylabel('z','fontsize',fs)
set(gca,'fontsize',fs-4)

subplot(1,10,3:4)
plot(real(w_an),z,'r','linewidth',lw)
hold on
plot(real(w),z,'.','markersize',ms);
axis tight;
title('(b) w_r','fontsize',fs','fontweight','normal')
set(gca,'yticklabel','')
set(gca,'fontsize',fs-4)

subplot(1,10,5:6)
plot(imag(w_an),z,'r','linewidth',lw)
hold on
plot(imag(w),z,'.','markersize',ms);
axis tight;
title('(c) w_i','fontsize',fs','fontweight','normal')
set(gca,'yticklabel','')
set(gca,'fontsize',fs-4)

subplot(1,10,7:8)
[Phase, Mag]=cart2pol(real(w),imag(w));
[Phase_an, Mag_an]=cart2pol(real(w_an),imag(w_an));
plot(Mag_an,z,'r','linewidth',lw)
hold on
plot(Mag,z,'.','markersize',ms);
axis tight;
title('(d) |w|_ ','fontsize',fs','fontweight','normal')
set(gca,'yticklabel','')
set(gca,'fontsize',fs-4)

subplot(1,10,9:10)
plot(Phase_an,z,'r','linewidth',lw)
hold on
plot(Phase,z,'.','markersize',ms);
axis tight;
title('(e)_ \phi_w','fontsize',fs','fontweight','normal')
set(gca,'yticklabel','')
set(gca,'fontsize',fs-4)

h=legend('num','an','location','northeast');legend boxoff
set(h,'orientation','vertical')

print('-dpdf',[direc 'hmwk3_3'])
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

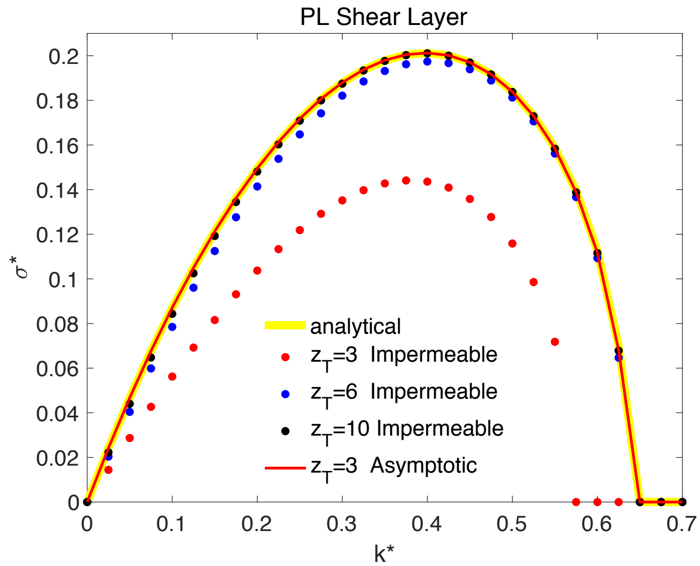


Figure 14.12: Scaled growth rate versus wavenumber for the piecewise-linear shear layer. Computation was done with impermeable boundaries in three locations (red, blue and black circles), and once with asymptotic boundary conditions (red curve). The analytical result (3.4.12) is shown in yellow.

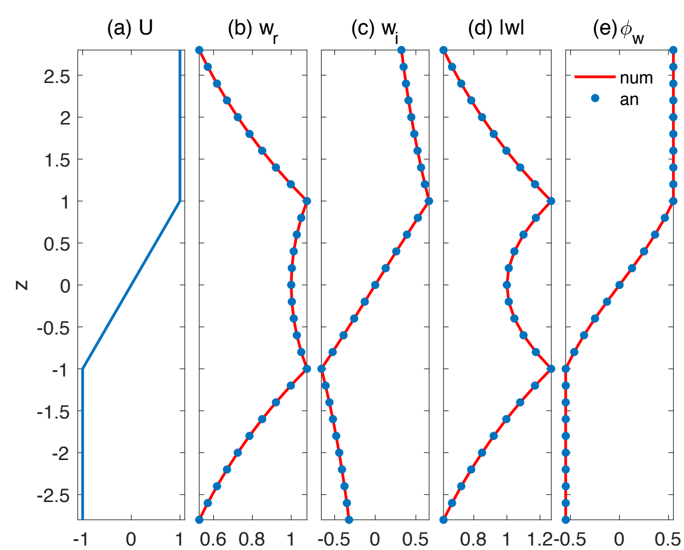


Figure 14.13: Eigenfunction of $\hat{w}$ for the fastest-growing mode of the piecewise-linear shear layer, numerical (red curve) and analytical (blue dots). At left is the velocity profile followed by the real part, imaginary part, magnitude and phase of $\hat{w}$.