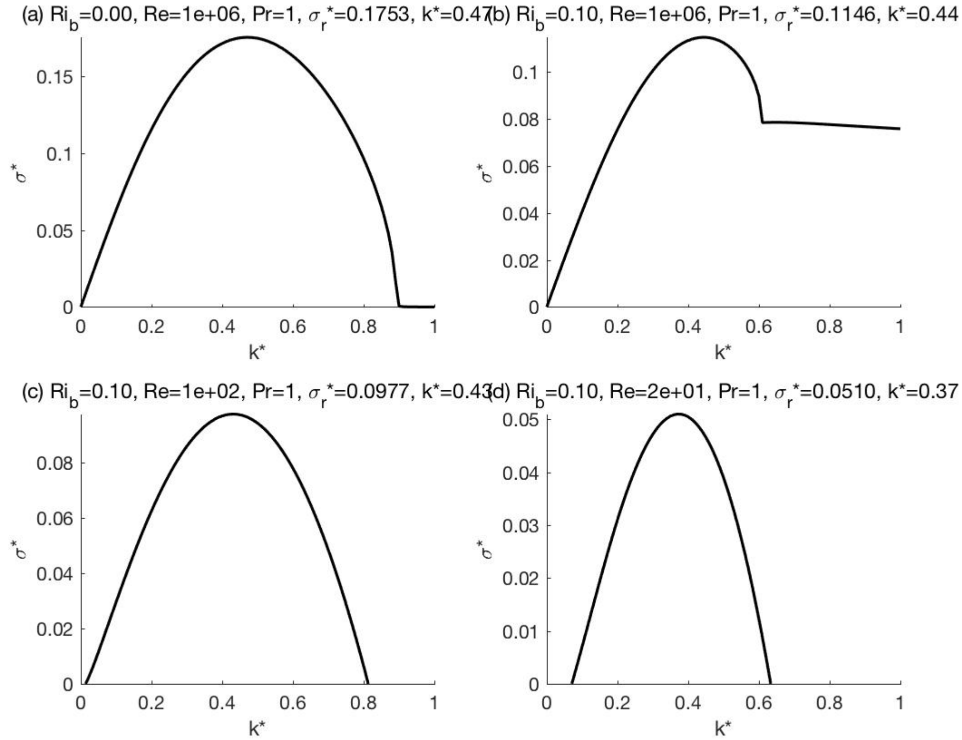

Figure 1: Growth rate versus wavenumber for four cases of a stratified shear flow. $U^* = -\tanh z^*$, $B^* = Ri_b \tanh z^*$. Impermeable, fixed-buoyancy boundaries at $z^* = \pm 4$. Grid spacing $\Delta^* = 0.2$. Note under-resolution and boundary effects at high and low k^*.

18: Instabilities in a plunging downslope flow

See figure 1.

(**a**) We get $\sigma^* = 0.1753$ at $k^* = 0.47$ The growth rate is much greater than $1/Re$, so the frozen flow hypothesis is valid. $S_{mx} = 1$, so the scaled growth rate is 0.1753. In contrast, the boundary layer instability in problem 1 gave a scaled growth rate of 0.06, considerably smaller than this. The main difference is the proximity of the lower boundary in problem 1, which tends to damp instabilities.

(**b**) We get $\sigma^* = 0.1146$ at $k^* = 0.44$. The growth rate is much greater than $1/Re$, so the frozen flow hypothesis is valid. The behavior at large k^* is an artifact of coarse resolution. It drops away as resolution is increased (see comparison below), while behavior near the FGM is not affected.

(**c**) We get $\sigma^* = 0.0977$ at $k^* = 0.43$. The growth rate is a factor of ten greater than $1/Re$, so the frozen flow hypothesis is marginally valid. Note that decreasing Re has an effect similar to increasing resolution. It smooths the eigenfunctions, making them easier to resolve on a coarse grid. Thus, the anomalous behavior at high k^* is removed even when $\Delta^* = 0.2$.

(**d**) We get $\sigma^* = 0.0510$ at $k^* = 0.37$. The growth rate is similar to $1/Re$, so the frozen flow hypothesis is not valid.

A typical wavenumber from cases a-c is $k^* = 0.45$, which yields a wavelength of $14h$, or 7 times the thickness of the shear layer ($2h$). Correcting for the aspect ratio of the image (4), this wavelength would appear about

7/4 times the thickness of the shear layer (figure 3). On the figure, the vertical line indicates an estimate of the initial thickness of the shear layer. The horizontal line is longer by a factor 7/4. This actually agrees pretty well with the typical scale of the earliest deformations of the shear layer (which may be pure luck!) Further downstream, larger structures emerge, possibly as a result of merging of the primary billows.

```
% script SSF_plunging
% Stability analysis of viscous stratified shear flow
% Calls functions: SSF, ddz, ddz2
% Used for problem A.18 of
%   Instability in Geophysical Flows
%   W.D. Smyth & J.R. Carpenter
%   Cambridge University Press, 2019
%
% Written using Matlab R2017a
%   W. Smyth, Oregon State University, 2018
%   smythw@oregonstate.edu

clear

% set parameters
Lz_st=8;
Ri_bs=[0 0.1 0.1 0.1];
Pr=1;
Res=[1e6 1e6 100 20];

% define z values
del=0.2;
z_st=[-Lz_st/2+del:del:Lz_st/2-del]'; % exclude boundaries
nz=length(z_st)
labs=['(a) ';'(b) ';'(c) ';'(d) '];

figure

% loop over Ri, Re
for j=1:4
    Ri_b=Ri_bs(j);
    Re=Res(j);

    % define profiles
    U_st=-tanh(z_st);

    B_st=Ri_b*tanh(z_st);
    Bz_st=Ri_b*sech(z_st).^2;

    % loop over k
    ks=[0.0:.01:1];nks=length(ks)
    l_st=0;
    for i=1:nks
```

```
        k_st=ks(i)
        [s_st,w_st,beta_st]=SSF(z_st,U_st,Bz_st,k_st,l_st,1/Re,1/(Re*Pr));
        sig_st(i)=real(s_st(1));
    end

    % plot results
    subplot(2,2,j)
    hold on
    plot(ks,sig_st,'k','linewidth',1.5)
    xlabel('k*')
    ylabel('\sigma*')
    title('LSA of stratified shear flow')
    axis tight
    yl=ylim;
    xlim([0 1])
    ylim([0 yl(2)])

    srmax=max(sig_st);
    kmax=ks(sig_st==srmax);
    ttle=sprintf('Ri_b=%.2f, Re=%.0e, Pr=%.0f, \\sigma_r*=%.4f, k*=%.2f',Ri_b,Re,Pr,srmax,kmax)
    title([labs(j,:) ttle],'fontweight','normal')
end
```

Most general stratified shear flow routine:

```
function [sig,w,b]=SSF(z,U,Bz,k,l,nu,kappa,bcw,bcb,imode)
%
% USAGE: [sig,w,b]=SSF(z,U,Bz,k,l,nu,kappa,bcw,bcb,imode)
%
% Stability analysis for a viscous, diffusive, stratified, parallel shear flow
% INPUTS:
% z = vertical coordinate vector (evenly spaced)
% U = velocity profile
% Bz = buoyancy gradient profile (Bz=squared BV frequency)
% k,l = wave vector
% nu, kappa = viscosity, diffusivity
% bcw = boundary conditions on w at (1) z=z(0) and (2) z=z(N+1)
%       r=rigid, f=frictionless (default)
% bcb = boundary conditions on b at (1) z=z(0) and (2) z=z(N+1)
%       f=insulating, c=fixed-buoyancy (default)
% imode = mode choice (default imode=1)
%         imode=0: output all modes, sorted by growth rate
%
% OUTPUTS:
% sig = growth rate of FGM
% w = vertical velocity eigenfunction
% b = buoyancy eigenfunction
```

```
%
% CALLS:
% ddz2, ddz4
%
% CITATION:
%   Instability in Geophysical Flows
%   W.D. Smyth & J.R. Carpenter
%   Cambridge University Press, 2019
%
% W. Smyth, Oregon State University, February 2016

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

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

% defaults
if(~exist('bcw'))
    % bcs for w are both frictionless
    bcw='ff';
end
if(~exist('bcb'))
    %boundary conditions for b are both constant-buoyancy
    bcb='cc';
end
if ~exist('imode');imode=1;end

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

% differentate U
Uzz=ddz2(z)*U;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 2: Derivative matrices and BCs

% D2: 2nd derivative with impermeable boundary
D2=ddz2(z);
D2(1,:)=0;D2(1,1:2)=[-2 1]/del^2;
D2(N,:)=0;D2(N,N-1:N)=[1 -2]/del^2;
```

```
% Fourth derivative with frictionless or rigid boundaries
D4=ddz4(z);
if bcw(1)=='f';    % frictionless
    D4(1,:)=0;D4(1,1:3)=[5 -4 1]/del^4;
elseif bcw(1)=='r';     % rigid
    D4(1,:)=0;D4(1,1:3)=[7 -4 1]/del^4;
else
    display(['SSF: Boundary condition 1 on w not understood'])
    sig=nan;
    return
end
D4(2,:)=0;D4(2,1:4)=[-4 6 -4 1]/del^4;
if bcw(2)=='f'; % frictionless
    D4(N,:)=0;D4(N,N-2:N)=[1 -4 5]/del^4;
elseif bcw(2)=='r'
    D4(N,:)=0;D4(N,N-2:N)=[1 -4 7]/del^4;
else
    display(['SSF: Boundary condition 2 on w not understood'])
    sig=nan;
    return
end
D4(N-1,:)=0;D4(N-1,N-3:N)=[1 -4 6 -4]/del^4;

%
% 2nd derivative matrix for buoyancy

D2b=ddz2(z);
if bcb(1)=='c'
    % Dirichlet bcs
    D2b(1,:)=0;D2b(1,1:2)=[-2 1]/del^2;
elseif bcb(1)=='f'
    % Neumann bcs
    D2b(1,:)=0;D2b(1,1:2)=[1 1]*2/3/del^2;
else
    display(['ddz2: Boundary condition 1 on buoyancy not understood'])
    d=NaN;
    return
end

if bcb(2)=='c'
    % Dirichlet bcs
    D2b(N,:)=0;D2b(N,N-1:N)=[1 -2]/del^2;
elseif bcb(2)=='f'
    % Neumann bcs
    D2b(N,:)=0;D2b(N,N-1:N)=[1 -1]*2/3/del^2;
else
    display(['ddz2: Boundary condition 2 not understood'])
```

```
    d=NaN;
    return
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 3: Assemble stability matrices
%
% Laplacian and squared Laplacian matrices
Id=eye(N);
L=D2-kt^2*Id;
Lb=D2b-kt^2*Id;
LL=D4-2*kt^2*D2+kt^4*Id;

N2=2*N;
NP=N+1;
A=zeros(N2,N2);B=zeros(N2,N2);

% assemble matrix A
A=[L Id*0 ; Id*0 Id];

% Compute submatrices of B using Levi's syntax
b11=-ii*k*diag(U)*L+ii*k*diag(Uzz)+nu*LL;
b21=-diag(Bz);
b12=-kt^2*Id;
b22=-ii*k*diag(U)+kappa*Lb;

% assemble matrix B
B=[b11 b12 ; b21 b22];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 4: Solve eigenvalue problem and extract results
%
% Solve generalized eigenvalue problem
[v,e]=eig(B,A);sigma=diag(e);

% Sort eigvals
[sr,ind]=sort(real(sigma),1,'descend');
sigma=sigma(ind);
v=v(:,ind);


% Extract the selected mode(s)
if imode==0
    sig=sigma;
    w=v(1:N,:);
    b=v(NP:N2,:);
elseif imode>0
    sig=sigma(imode);
```

```
    w=v(1:N,imode);
    b=v(NP:N2,imode);
end

return
```