16: A convectively unstable layer in an inviscid fluid, revisited.

The stability equation is (cf. 2.28):

$$\sigma^2\left(\frac{d^2}{dz^2}-\tilde{k}^2\right)\hat{w}=\tilde{k}^2B_z\hat{w}\ , \tag{2}$$

where

$$B_z=B_{z0}(1-2\mathrm{sech}^2\alpha z).$$

How do you choose values for the constants B_{z0} and α? That problem is sidestepped by scaling. Suppose we have a solution procedure

$$\sigma=\mathscr{F}(z,B_z,\tilde{k})$$

Define scaled variables

$$\sigma=\sqrt{B_{z0}}\sigma_*\ ;\ \ z=z_*/\alpha\ ;\ \ \tilde{k}=\alpha\tilde{k}_*\ ;\ \ B_z=B_{z0}\beta\ .$$

Substitute and get

$$\sigma_*^2\left(\frac{d^2}{dz_*^2}-\tilde{k}_*^2\right)\hat{w}=\tilde{k}_*^2\beta(z_*)\hat{w},$$

with

$$\beta=1-2\mathrm{sech}^2z_*.$$

This is isomorphic to (2), and therefore the solution procedure is

$$\sigma_*=\mathscr{F}(z_*,\beta,\tilde{k}_*)\ .$$

The parameters B_{z0} and α have been removed from the problem.

(**a**) The growth rate curve increases monotonically (figure 1), approaching a limit as $k_*\to\infty$. This is similar to convection in an inviscid fluid with uniform B_z. The limiting growth rate is 1, or $\sigma=\sqrt{B_{z0}}$, consistent with the upper bound found in section 2.2.3.

(**b**) The results match (figure 2).

(**c**) The case done previously does *not* represent the fastest-growing mode.

(**d**) The fastest-growing mode is at $k_*=\infty$. (I did not anticipate this result when I posed the question.) If you chose a reasonably large value of k_*, your eigenfunction was sharply peaked at $z_*=0$.

Here is a function to find convective modes for a general buoyancy profile in an inviscid, nondiffusive fluid:

```
function [sig1,w1]=convect(z,Bz,kt,BC,nmode)
%
% MINIMAL USAGE: [sig]=convect(z,Bz,kt)
%    FULL USAGE:  [sig,w]=convect(z,Bz,kt,nmode)
%
% Stability analysis for inviscid, homogeneous, parallel shear flow
% (Rayleigh equation)
%
% INPUTS:
% z = vertical coordinate vector (evenly spaced)
```

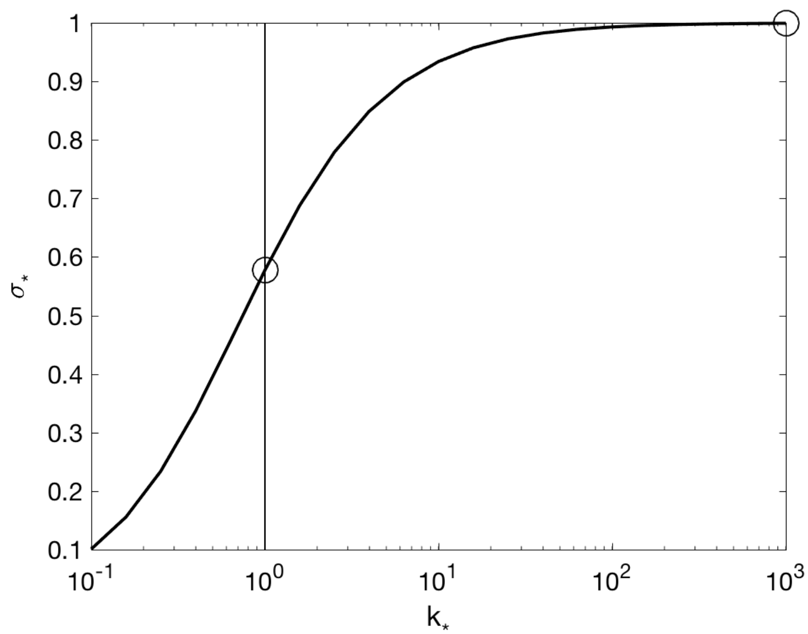

Figure 1: Scaled growth rate versus wavenumber for unstable layer. Circles show the analytical solution for $k_* = 1$ and the fastest-growing mode among the wavenumbers scanned.

```
% Bz = buoyancy gradient profile
% kt = wave vector magnitude
% BC = vector of boundary conditions at z(1) and z(end):
%       1 = impermeable
%       2 = asymptotic
% nmode = mode selection in terms of growth rate:
%       1 = fastest-growing mode [default nmode=1]
%       2 = second-fastest mode, etc.
%
% OUTPUTS:
% sig = complex growth rate
% w = vertical velocity eigfn
%
% W. Smyth, Feb 2018

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

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

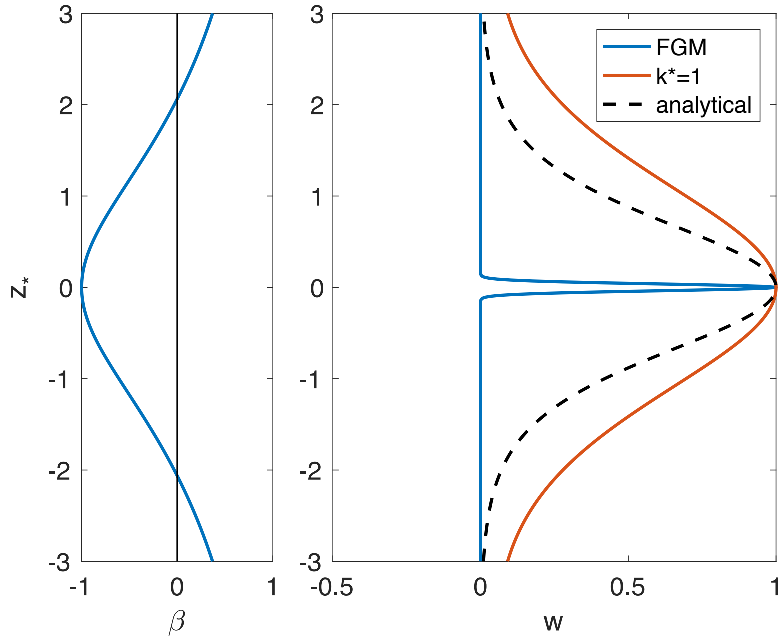

Figure 2: Left: scaled stratification profile. Right: eigenfunctions as indicated. Note that the eigenfunctions are real.

```
    return
end

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

% define constants
del=z(2)-z(1);
N=length(z);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stage 2: Set up derivative matrices
%
D2=ddz2(z); % 2nd derivative matrix with 1-sided boundary terms

% Boundary conditions
% 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 BC(1)==2
```

```
        D2(1,:)=0;D2(1,1)=-2*(1+del*kt)/del^2;D2(1,2)=2/del^2;
end
if BC(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=kt^2*diag(Bz);

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

return
end
```