# Project3 - Taylor (by Fergus)

September 11, 2018

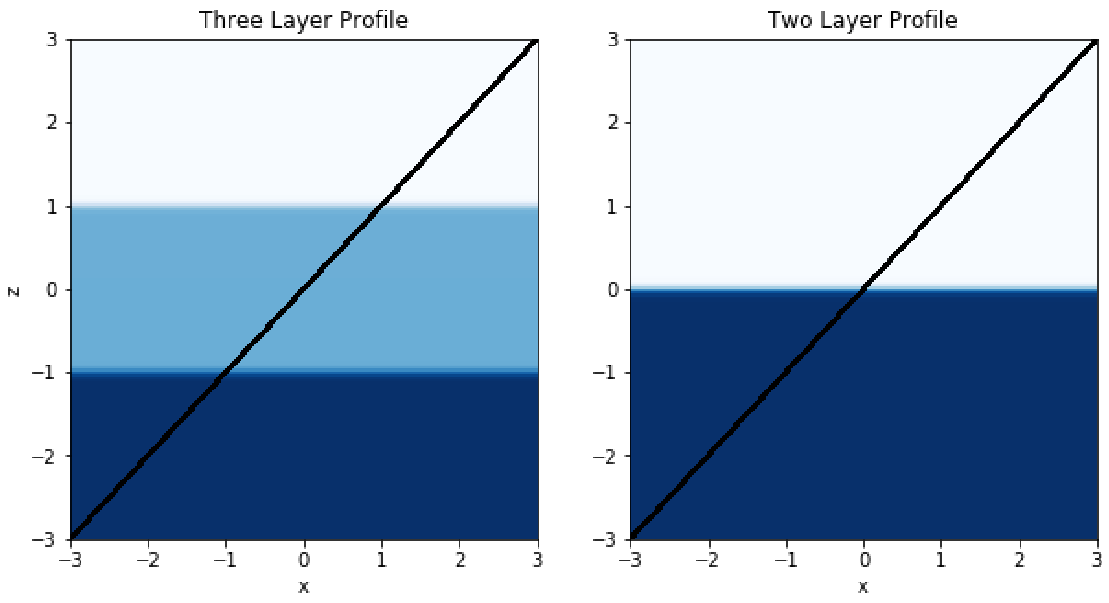
## 1 Introduction - Profile and Equations

The Taylor-instability occurs on a Couette flow in a 3-layer stably stratified fluid. Such conditions satisfy the equations of the Boussinisq approximation - where velocity is only in the $\hat{x}$ direction as a function of $z$ and density is only a function of $z$. In this investigation we also examine the case of Couette flow on a 2-layer stably stratified fluid for mathematical simplicity and understanding of the problem. Plots of each of these cases are given below.

```
In [197]: import numpy as np
          from  matplotlib.pyplot  import *
          %matplotlib inline
          import matplotlib.pyplot as plt
          from matplotlib import gridspec
          from matplotlib import rc

          #functions for plotting eigenvalues are defined at the end of the notebook
```

```
In [198]: nz,  Lz  =  200,  6.0
          nx, Lx = 200, 6
          Lambda=1
          Rho0=10
          RhoS=1
          hh=0.05
          h=1

          z = np.linspace(-Lz/2, Lz/2, nz)
          x = np.linspace(-Lx/2, Lx/2, nx)
          X, Z = np.meshgrid(x, z)

          #Profiles
          U = Lambda*Z
          Rho3 = (Rho0 - (RhoS/2)*(np.tanh((Z+h)/hh)+np.tanh((Z-h)/hh))) * np.exp(0*X)
          Rho2 = (Rho0 - (RhoS)*np.tanh((Z)/hh)) * np.exp(0*X)

          f, axs = plt.subplots(1, 2, figsize=(10, 5))

          gs = gridspec.GridSpec(1, 2)
```

```
axs[0], axs[1]= plt.subplot(gs[0, 0]), plt.subplot(gs[0, 1])
axs[0], axs[1]= plt.subplot(1, 2, 1), plt.subplot(1, 2, 2)

axs[0].pcolormesh(X,Z,Rho3,cmap='Blues')
axs[0].plot(U/Lambda, z/h,'k')
axs[0].set_title("Three Layer Profile", fontsize=12)
axs[0].set_xlabel("x", fontsize=10)
axs[0].set_ylabel("z", fontsize=10)
axs[1].pcolormesh(X,Z,Rho2,cmap='Blues')
axs[1].plot(U/Lambda, z/h,'k')
axs[1].set_title("Two Layer Profile", fontsize=12)
axs[1].set_xlabel("x", fontsize=10);
```



The 3-layer case is described by the following equations:

(1)

$$\vec{u} = \Lambda z \hat{x} \tag{1}$$

(2)

$$\bar{\rho}(z) = \rho_0 - \frac{\rho_s}{2}(\text{sign}(z+1) + \text{sign}(z-1)) \tag{2}$$

$\Lambda$ is the strength of velocity shear in the couette flow profile, with units $s^{-1}$. $\rho_s$ is the magnitude of the density difference at each step in the profile. For simplicity we will begin by considering the step function in density as a general function in $z$ on a background mean density.

(3)

$$\rho(z) = \rho_0 + \bar{\rho}(z) \tag{3}$$

## 2 Analytic Stability Analysis

To analyse the stability of the velocity and density profiles we introduce a small perturbations the the system and look it how it evolves in time. This is done by linearising and reducing to an eigenvalue problem for the growth of the perturbation in the form $\phi(t) = \phi(t_0)e^{\sigma t}$. Here $\sigma$ is the eigenvalue and any positive, real component will result in growth of the perturbation and instability.

(4)
$$\vec{u}(x,z,t) = (\Lambda z + u'(x,z,t))\hat{x} + w'(x,z,t))\hat{z} \tag{4}$$

(5)
$$\rho(x,z,t) = \rho_0 + \bar{\rho}(z) + \rho'(x,z,t) \tag{5}$$

(6)
$$P(x,z,t) = P_0 - g\int_0^z \bar{\rho}(z')dz' + P'(x,z,t) \tag{6}$$

The equations of motion from the Boussinesq approximation are given as:

(7)
$$\partial_t \vec{u} + \vec{u}\cdot\nabla\vec{u} = -\nabla P - g\frac{\rho'}{\rho_0}\hat{z} \tag{7}$$

(8)
$$\partial_t \rho + \vec{u}\cdot\nabla\rho = 0 \tag{8}$$

(9)
$$\nabla\cdot\vec{u} = 0 \tag{9}$$

From here we use the equations and boundary conditions in the system to find the value(s) of $\sigma$ in the system.

First we introduce a streamfunction $\psi$ and eliminate pressure to obtain the following expression for each layer (using eqn's (7) and (9)):

(10)
$$(\partial_t + \Lambda z\partial_x)\nabla^2\psi_j' = -g\frac{\partial_x\rho_j'}{\rho_0}\hat{z} \tag{10}$$

The system is considered in terms of its uniform density layers because at the interfaces between these points quantities either change or must be conserved. We also introduce the perturbation terms into the density equation. In all cases, only perturbation terms to first order are considered.

(11)
$$\partial_t\rho_j' + \Lambda z\partial_x\rho_j' + \partial_x\psi_j'\partial_z\bar{\rho}(z) = 0 \tag{11}$$

Our system, and therefore perturbation terms are invariant in $x$, as such we can assume solutions of the form: $f'(x,z,t) = \tilde{f}(z)e^{ikx+\sigma t}$. Substituting these into equations (10) and (11) gives:

(12)

$$(\sigma + ik\Lambda z)(\partial_{zz} - k^2)\tilde{\psi}_j(z) = -g\frac{ik\tilde{\rho}_j(z)}{\rho_0} \tag{12}$$

(13)

$$(\sigma + ik\Lambda z)\tilde{\rho}_j(z) + ik\tilde{\psi}_j(z)\partial_z\bar{\rho}(z) = 0 \tag{13}$$

From (12) and (13) we eliminate the density terms:

(14)

$$\tilde{\rho}_j(z) = \frac{ik\tilde{\psi}_j(z)\partial_z\bar{\rho}(z)}{\sigma + ik\Lambda z} \tag{14}$$

This gives the following equation in only $\tilde{\psi}_j(z)$:

(15)

$$(\sigma + ik\Lambda z)(\partial_{zz} - k^2)\tilde{\psi}_j(z) = g\frac{k^2\tilde{\psi}_j(\partial_z\bar{\rho}(z)}{(\sigma + ik\Lambda z)\rho_0} \tag{15}$$

(16)

$$(\sigma + ik\Lambda z)^2(\partial_{zz} - k^2)\tilde{\psi}_j(z) - g\frac{k^2\partial_z\bar{\rho}(z)}{\rho_0}\tilde{\psi}_j(z) = 0 \tag{16}$$

Here we can note that $g\frac{\partial_z\bar{\rho}(z)}{\rho_0} = N^2$, the Brunt Vaisala frequency. This is zero within the layers (where there is no density gradient) and non-zero at the interface (density jump) boundaries. As such, we can express our streamfunction equation for within each layer as:

(17)

$$(\sigma + ik\Lambda z)^2(\partial_{zz} - k^2)\tilde{\psi}_j(z) = 0 \tag{17}$$

This equation is solved by solutions of the form $Ae^{kz} + Be^{-kz}$. When we consider the boundary conditions in the top and bottom layers - with perturbations going to zero at $\pm\infty$ we are left with the following system:

$$\tilde{\psi}_1(z) = A_1 e^{-k(z-1)} \tag{18}$$

$$\tilde{\psi}_2(z) = A_2 e^{k(z+1)} + A_3 e^{-k(z-1)} \tag{19}$$

$$\tilde{\psi}_3(z) = A_4 e^{k(z+1)} \tag{20}$$

If we look at the simpler 2-layer case, the system is instead:

$$\tilde{\psi}_1(z) = B_1 e^{-kz} \tag{21}$$

$$\tilde{\psi}_2(z) = B_4 e^{kz} \tag{22}$$

We now look to introduce boundary conditions from the interfaces between layers. The conserved quantites crossing these interfaces are velocity perpendicular to the material surface between layers $\eta$ and total pressure. These are expressed as:

(18)

$$\tilde{\eta} = \frac{ik\tilde{\psi}}{\sigma + ik\Lambda z} \tag{23}$$

(19)

$$(\rho_0 + \bar{\rho}(z))((\sigma + ik\Lambda z)\partial_z \tilde{\psi} - ik\tilde{\psi}U_0 + \frac{k^2\tilde{\psi}}{\sigma + ik\Lambda z}g) \tag{24}$$

(20) is derived from the material derivative of the interface and (19) is derived from conservation of total pressure. With these boundary conditions we can investigate the stability of the system.

## 3 Instability

To solve for $\sigma$ we construct an equation of the form $\mathbb{X}_3 \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = 0$ or $\mathbb{X}_2 \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} = 0$ then solve for $\sigma$ with $\text{Det}(\mathbb{X}_l) = 0$. In essence we solve for $\sigma$ by considering the relative dependencies of the $A_s$'s and $B_i$'s. These systems of equations appear as:

$$\left( \begin{matrix} \frac{ik}{\sigma+ik\Lambda} & \frac{-ike^{2k}}{\sigma+ik\Lambda} & \frac{-ik}{\sigma+ik\Lambda} \\ 0 & \frac{ik}{\sigma+ik\Lambda} & \frac{ike^{2k}}{\sigma+ik\Lambda} \\ (\rho_0 - \rho_s)(-k(\sigma+ik\Lambda) - ik\Lambda + \frac{k^2}{\sigma+ik\Lambda}g) & -\rho_0(k(\sigma+ik\Lambda)e^{2k} - ike^{2k}\Lambda + \frac{k^2e^{2k}}{\sigma+ik\Lambda}g) & -\rho_0(-k(\sigma+ik\Lambda)A_3 - ik\Lambda + \\ 0 & \rho_0((\sigma-ik\Lambda)k - ik\Lambda + \frac{k^2}{\sigma-ik\Lambda}g) & \rho_0((\sigma-ik\Lambda)ke^{2k} - ike^{2k}\Lambda + \end{matrix} \right. \tag{25}$$

In the 2-layer Case:

$$\begin{pmatrix} \frac{1}{\sigma} & -\frac{1}{\sigma} \\ (\rho - \frac{\rho_s}{2})(\sigma k - ik\Lambda + \frac{k^2}{\sigma}) & (\rho + \frac{\rho_s}{2})(\sigma k + ik\Lambda - \frac{k^2}{\sigma}) \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} = 0 \tag{26}$$

We will ficus first on the significantly simpler 2-layer case. Solving $\text{Det}(\mathbb{X}_2) = 0$ we have:

(20)

$$-gk^2\rho_s + ik\rho_s\Lambda\sigma - 2k\rho_0\sigma^2 = 0 \tag{27}$$

With solutions:

(21)

$$\sigma = \frac{i\rho_s\Lambda}{4\rho_0} \pm \sqrt{\frac{-\rho_s^2\Lambda^2 - 8gk\rho_s\rho_0}{16\rho_0^2}} \tag{28}$$

This is purely imaginary for all combinations of shear, gravity and stratification. As such, we determine that the 2-layer system is stable. The isolaed edge wave on the interface does not led to instability.

It can be noted here that if the sign of gravity is reversed, we change the system to a state of unstable stratification. However, there exist purturbation regimes where the flow is stable, even with unstable stratification. This is where $\sigma$ remains purely imaginary.

(22)

$$-\rho_s^2\Lambda^2 + 8gk\rho_s\rho_0 < 0 \tag{29}$$

Assuming $\rho_s \neq 0$

(23)

$$k > \frac{\Lambda^2 \rho_s}{8 g \rho_0} \tag{30}$$

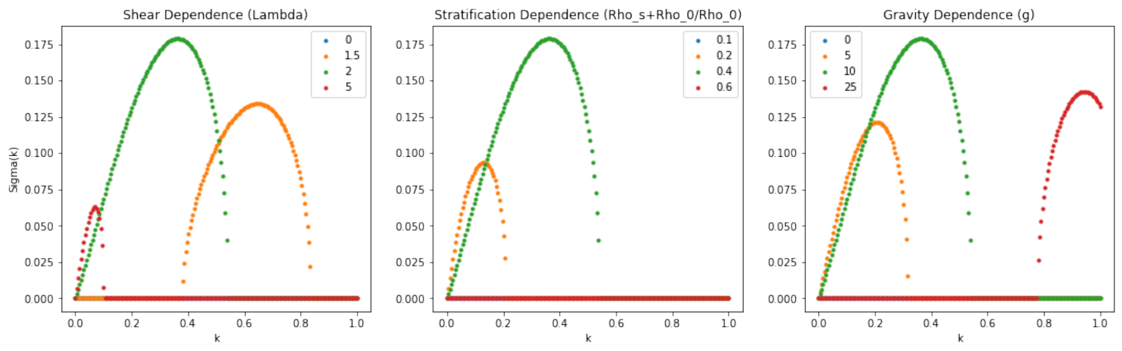
These are cases where the wavenumber (wavelength) is sufficiently large (small) that the perturbation does not grow. It is interesting to note that we have stability without gravity - in this case the density of the fluid is only relevant for momentum. The result is also suggestive that stability can be achieved with large shears or even stronger stratification in the unstably stratified system. The latter is rather unintuitive and in that case the growth rate of instability, if it occurs, increases with the strength of stratification as would be expected.

We now return to the 3-layer case, with is significantly more algebraically complex. Setting $\text{Det}(\mathbb{X}) = 0$ gives a fourth order polynomial in $\sigma$. Plots of $\sigma(k)$ are given in the next section.

## 4 Perturbation growth rate $\sigma(k)$

```
In [294]: kmin=0
          kmax=1
          nk=200
          Rhovec=[0,1,3,5]
          Rhosbase=3
          Rhomean=10
          grav=[0,-5,-10,-25]
          gravbase=-10
          shear =[0,1.5,2,5]
          shearbase=2
          XX = sigmaplots(kmin, kmax, nk, Rhovec, Rhosbase, Rhomean, grav, gravbase, shear, sh

          f, axs = plt.subplots(1, 3, figsize=(18, 5))
          gs = gridspec.GridSpec(1, 3)
          axs[0], axs[1], axs[2]= plt.subplot(gs[0, 0]), plt.subplot(gs[0, 1]), plt.subplot(gs
          axs[0], axs[1], axs[2]= plt.subplot(1, 3, 1), plt.subplot(1, 3, 2), plt.subplot(1, 3

          axs[0].plot(XX[:,0], XX[:,1],'.',label=shear[0])
          axs[0].plot(XX[:,0], XX[:,2],'.',label=shear[1])
          axs[0].plot(XX[:,0], XX[:,3],'.',label=shear[2])
          axs[0].plot(XX[:,0], XX[:,4],'.',label=shear[3])
          axs[0].legend()
          axs[0].set_title('Shear Dependence (Lambda)',fontsize=12)
          axs[0].set_xlabel("k", fontsize=10)
          axs[0].set_ylabel("Sigma(k)", fontsize=10)

          axs[1].plot(XX[:,0], XX[:,5],'.',label=(1+Rhovec[0])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,6],'.',label=(1+Rhovec[1])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,7],'.',label=(1+Rhovec[2])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,8],'.',label=(1+Rhovec[3])/Rhomean)
          axs[1].legend()
```

```
axs[1].set_title("Stratification Dependence (Rho_s+Rho_0/Rho_0)", fontsize=12)
axs[1].set_xlabel("k", fontsize=10)

axs[2].plot(XX[:,0], XX[:,9],'.',label=-grav[0])
axs[2].plot(XX[:,0], XX[:,10],'.',label=-grav[1])
axs[2].plot(XX[:,0], XX[:,11],'.',label=-grav[2])
axs[2].plot(XX[:,0], XX[:,12],'.',label=-grav[3])
axs[2].legend()
axs[2].set_title("Gravity Dependence (g)", fontsize=12)
axs[2].set_xlabel("k", fontsize=10);
```



The above plots of $\sigma(k)$ are generated varying the shear, stratification and gravitational strength in the system. It can be seen that stability or instability is dependent on all of these factors. In general, conditions for instability support a band of wavenumber $k$ and outside this band the system is stable. In each unstable case, $\sigma$ grows from zero at a minimum wavenumber, reaches a peak and then is reduces to zero for all great wavenumbers.

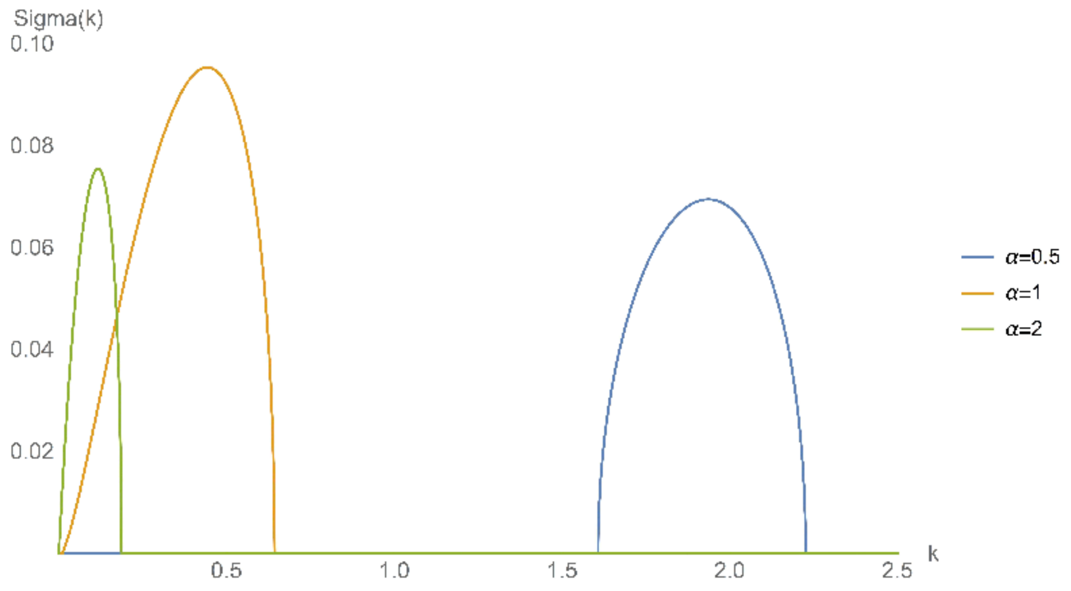
The plot featuring shear dependence shows the band of unstable $k$ shifting towards $k = 0$ and shrinking as $\Lambda$ is increased, for large $k$. While shrinking, the opposite occurs for small $k$, with broadening and a shift to large $k$, indicating an 'optimum' or 'most unstable' shear. This is similar to what we saw in the 2 layer case where increased shear can act to stabilize the flow (even if the stratification is unstable), while it is essential for generating any instability.

Stratification dependence of the instability is very much as expected - with no stratification there is stable couette flow and with stratification too strong an instability cannot form resulting in a range of unstable $k$ with a $\sigma(k)$ maximum. The effect of gravity in the system are intrinsically linked to stratification so we see the same structure of dependence.

In this system, the instability is generated from the interaction of edge waves formed at each density jump interface. We have seen that the 2-layer, or single-step, case is stable - the edge wave at the interface does not lead to instability. However, with a second jump sufficiently close that the waves can interact there is the potential for instability. The two edge waves require sufficient cross-stream extent to interact. This occurs only for sufficiently long wavelengths because the cross-stream decay scale is proportional to the along-stream wavelength (Vallis, chapter 9). For high $k$, the along stream wavelength $\frac{2\pi}{k}$ is short. This means the cross-stream decay scale is short and the edge waves do not interact. This is a reason that the growth rates are only positive over a range of small $k$ - the waves need to be able to "see" each other in order to interact and grow. By this argument, if we make the space between the density jumps larger, smaller values of $k$

(large wavelengths) will produce instability and vice-versa for a decrease in the distance between density jumps. This is shown in the figure below (produced in mathematica) for density jumps at $z = \pm\alpha$ where $\alpha = 0.5, 1, 2$.

```
In [313]: from scipy import misc
          o=misc.imread('sigmaplot.jpg')
          plt.figure(figsize=(14,14))
          plt.axis("off")
          plt.imshow(o)
          plt.show()
```



We will now look at a numerical representation of the system and examine the equivalent stability of the system.

## 5 Numerical

To investigate stability of the system numerically we will use the following equations:

(12)

$$(\sigma + ik\Lambda z)(\partial_{zz} - k^2)\tilde{\psi}_j(z) = -g\frac{ik\tilde{\rho}_j(z)}{\rho_0} \tag{31}$$

(13)

$$(\sigma + ik\Lambda z)\tilde{\rho}_j(z) + ik\tilde{\psi}_j(z)\partial_z\bar{\rho}(z) = 0 \tag{32}$$

We set up a system fo equations of the form:

$$\partial_t \begin{pmatrix} \blacksquare \\ æ \end{pmatrix} = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} \blacksquare \\ æ \end{pmatrix} \tag{33}$$

The state vector includes both the density and streamfunction because of the interaction between them.

$$\partial_t \tilde{\psi} = \nabla^{-2}(-ik\Lambda z \nabla^2 \tilde{\psi} - \frac{gik}{\rho_0}\tilde{\rho}) \tag{34}$$

$$\partial_t \tilde{\rho} = -ik\Lambda z \tilde{\rho} - ik(\partial_z \bar{\rho}(z))\tilde{\psi} \tag{35}$$

Now we can build the components of the $\mathbb{A}$ matrix.

$$A_{1,1} = \nabla^{-2}(-ik\Lambda)\begin{pmatrix} z_j & & \\ & \ddots & \\ & & z_j \end{pmatrix}\nabla^2 \tag{36}$$

$$A_{1,2} = \nabla^{-2}(\frac{-gik}{\rho_0})\mathbb{I} \tag{37}$$

$$A_{2,1} = -ik\begin{pmatrix} \partial_z \bar{\rho}(z_j) & & \\ & \ddots & \\ & & \partial_z \bar{\rho}(z_j) \end{pmatrix} \tag{38}$$

$$A_{2,2} = -ik\Lambda\begin{pmatrix} z_j & & \\ & \ddots & \\ & & z_j \end{pmatrix} \tag{39}$$

In the numerical analysis there is no need to consider the boundary conditions of the problem because they are contained in the initial state. We will first look at the 2-layer case numerically. A sharp $\tanh(z)$ function is sampled to capture the density jump.

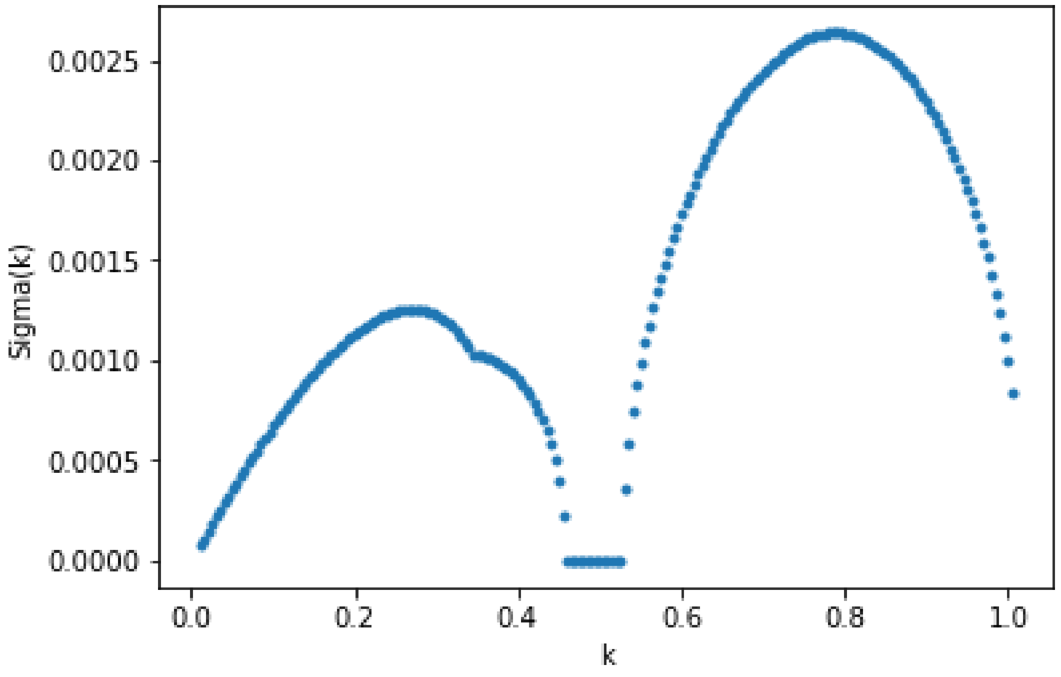
```
In [314]: nz,  Lz  =  200,  6.0
          nx, Lx = 200, 4*np.pi

          z = np.linspace(-Lz/2, Lz/2, nz)
          x = np.linspace(-Lx/2, Lx/2, nx)
          dz = z[1]-z[0]
          dx = x[1]-x[0]

          print("dz =", dz)
          print("Lz/(nz-1) =", Lz/(nz-1))
          print("dx =", dx)
          print("Lx/(nx-1) =", Lx/(nx-1))
```

```
dz = 0.0301507537688
Lz/(nz-1) = 0.030150753768844
dx = 0.0631475910269
Lx/(nx-1) = 0.06314759102693052
```

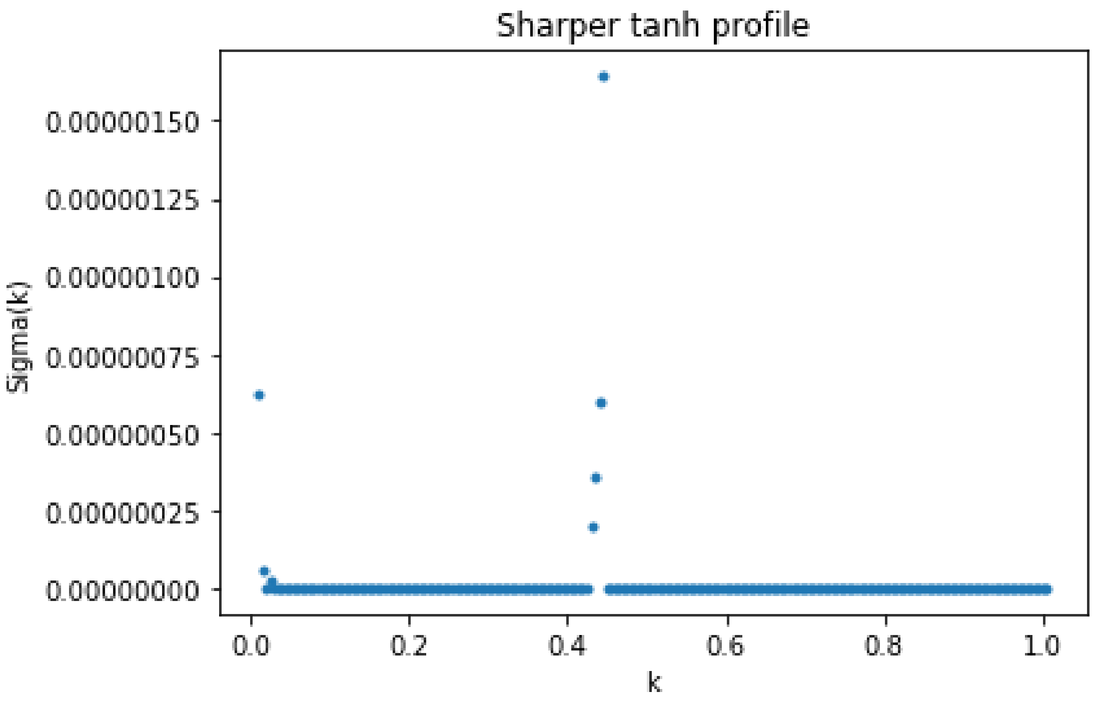
```
In [316]: #building the A matrix
          def create_diffz(nz, dz):
              Dz = np.diag(np.linspace(1, 1, nz-1), +1) - np.diag(np.linspace(1, 1, nz-1), -1)
              Dz[0, 0], Dz[0, 1]= -2, 2
              Dz[-1, -2], Dz[-1, -1] = -2, 2
              Dz = Dz/(2*dz)

              D2z = np.diag(np.linspace(1, 1, nz-1), +1) - 2*np.diag(np.linspace(1, 1, nz)) +
              D2z[0, :] = D2z[1, :]
              D2z[-1, :] = D2z[-2, :]
              D2z = D2z/(dz**2)
              return Dz, D2z

          Dz, D2z = create_diffz(nz, 1)

In [319]: def construct_Akop(k,L0,g):
              I=np.eye((nz))
              laplacian = D2z - k**2*I
              invlaplacian = np.linalg.inv(laplacian)
              Rhovec=Rho[:,0]
              DzRhovec=Dz @ Rhovec
              DzRhodiag=np.diag(DzRhovec)
              Zdiag=np.diag(Z[:,0])


              A11=-1j*k*L0*invlaplacian @ Zdiag @ laplacian
              A12=(-1j*g*k/Rho0)*invlaplacian
              A21=-1j*k*DzRhodiag
              A22=-1j*k*L0*Zdiag

              Top=np.append(A11,A12,axis=1)
              #print(len(Top))
              Bottom=np.append(A21,A22,axis=1)

              Ak=np.append(Top,Bottom,axis=0)
              #print(len(Ak))
              return Ak

In [330]: Acomplete=construct_Akop(1,1,10) #k, L0, g
          eigenValues, eigenVectors = np.linalg.eig(Acomplete)
          print(max(np.real(eigenValues)))

0.000993271014508
```

The maximum eigenvalue for $k = 1$ is very small, so we can say that the system is stable for that k-value. Interestingly with the same assumptions we can plot $\sigma(k)$ for the numerical system and see some small unstable bands of $k$ (below). This is possibly a consequence of using a tanh profile to sample the density jump with numerical differentiation. In the next plot the density profile far more closely approximates a step function and the result is stability.

```
In [368]: nk=200
          dk=0.01
          s=np.zeros((nk))
          kvec=np.zeros((nk))
          Rho = (Rho0 - (RhoS)*(np.tanh((Z)/hh)) * np.exp(0*X)

          Lambda=1 #Shear
          g=10 #gravity
          RhoS=0.5 #Stratification
          RHo0=10 #Base Density

          mn=0.01

          for i in range(nk):
              kvec[i]=i/200 + mn
              Acomplete=construct_Akop(kvec[i],Lambda,g) #k, L0, g
              eigenValues, eigenVectors = np.linalg.eig(Acomplete)
              s[i]=max(np.real(eigenValues))
          plt.plot(kvec,s,'.')
          plt.xlabel("k")
          plt.ylabel("Sigma(k)")
          plt.show()
```

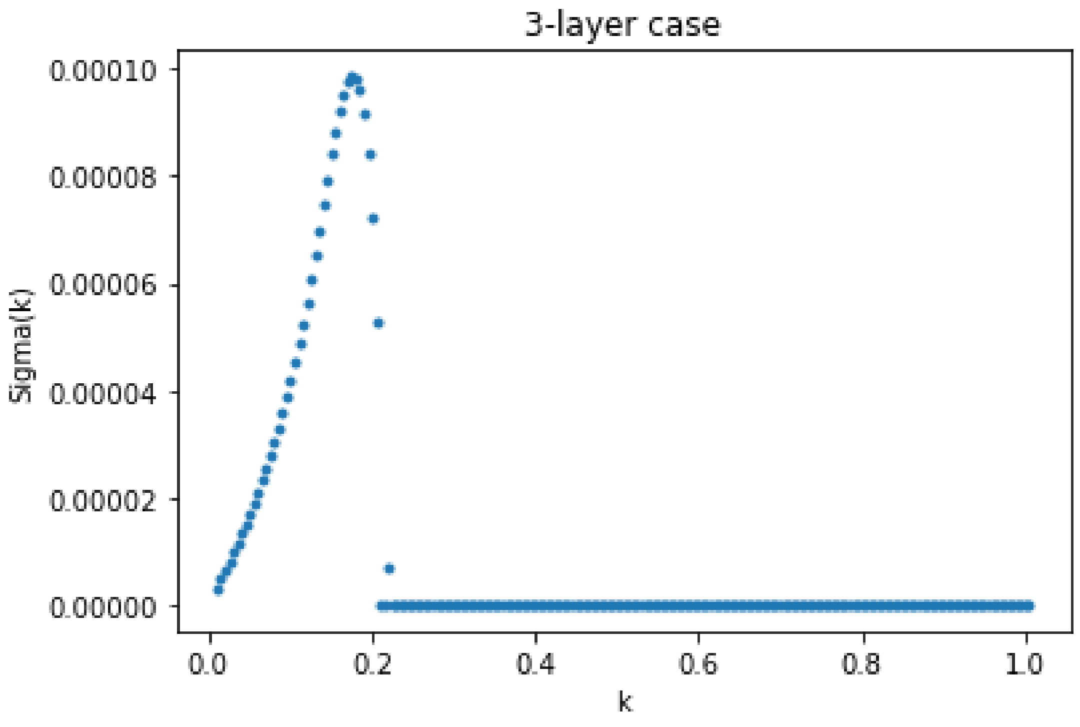


```
In [342]: Rho= (Rho0 - (RhoS)*(np.tanh((Z)/0.01))) * np.exp(0*X)
          nk=200
          dk=0.01
          s=np.zeros((nk))
          kvec=np.zeros((nk))


          Lambda=1 #Shear
          g=10 #gravity
          RhoS=0.5 #Stratification
          RHo0=10 #Base Density

          mn=0.01

          for i in range(nk):
              kvec[i]=i/200 + mn
              Acomplete=construct_Akop(kvec[i],Lambda,g) #k, L0, g
              eigenValues, eigenVectors = np.linalg.eig(Acomplete)
              s[i]=max(np.real(eigenValues))
          plt.plot(kvec,s,'.')
          plt.title("Sharper tanh profile")
          plt.xlabel("k")
          plt.ylabel("Sigma(k)")
          plt.show()
```

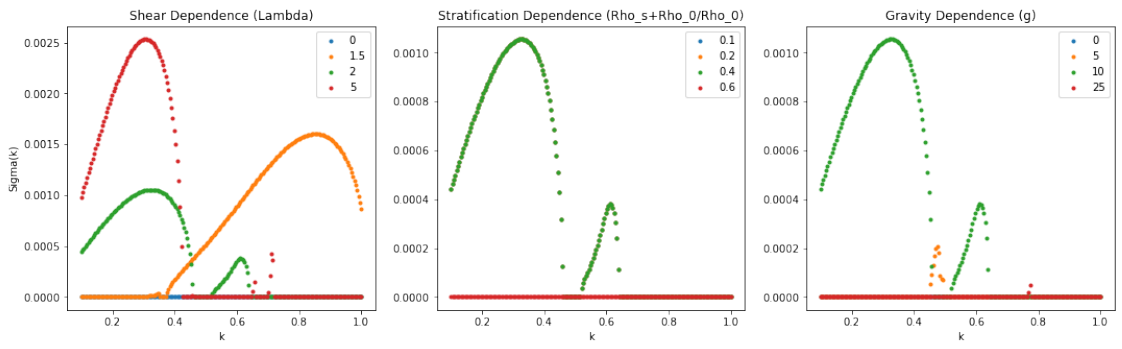


We will now look at the numerical results for the 3-layer system. The steeper tanh approximation is be used because it provided a stable result for the 2-layer system.

```
In [353]: nk=200
          dk=0.5
          s=np.zeros((nk))
          kvec=np.zeros((nk))

          Rho= (Rho0 - (RhoS/2)*(np.tanh((Z+1)/0.01)+np.tanh((Z-1)/0.01))) * np.exp(0*X)

          Lambda=1 #Shear
          g=10 #gravity
          RhoS=0.5 #Stratification
          RHo0=10 #Base Density

          mn=0.01

          for i in range(nk):
              kvec[i]=i/200 + mn
              Acomplete=construct_Akop(kvec[i],Lambda,g) #k, L0, g
              eigenValues, eigenVectors = np.linalg.eig(Acomplete)
              s[i]=max(np.real(eigenValues))
          plt.plot(kvec,s,'.')
          plt.title("3-layer case")
          plt.xlabel("k")
          plt.ylabel("Sigma(k)")
          plt.show()
```



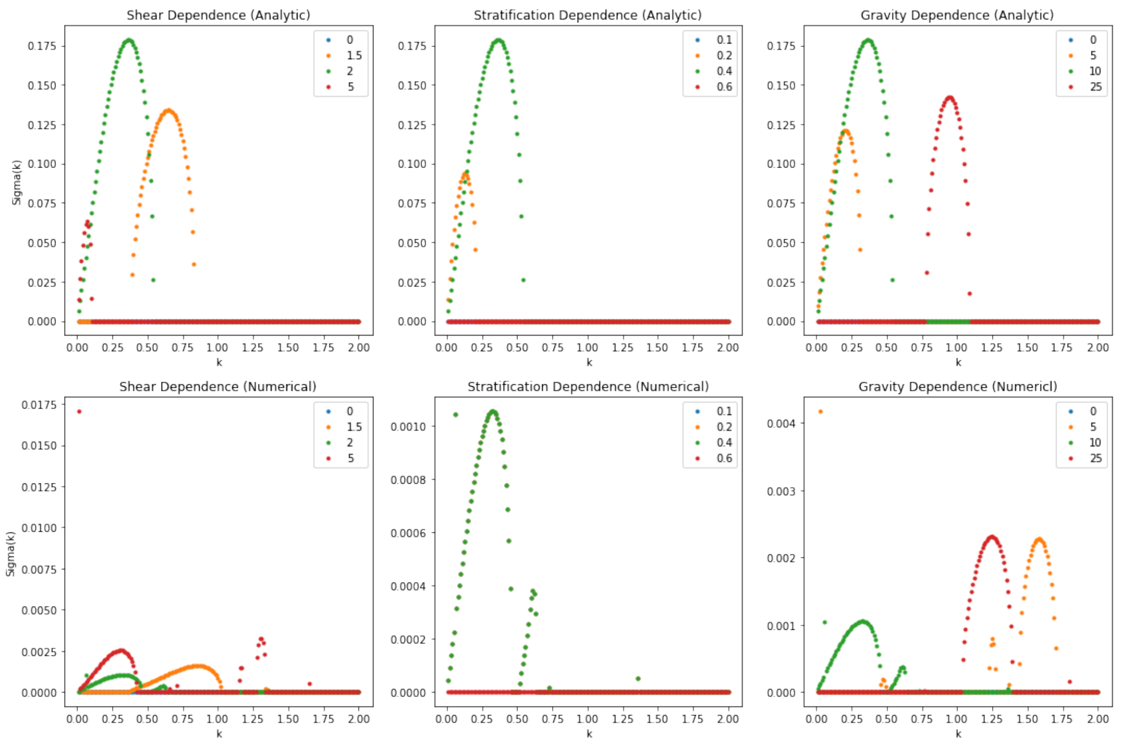
```
In [360]: kmin=0.1
          kmax=1
          nk=200 #needs to stay as 200 to fit the A matrix construction
          Rhovec=[0,1,3,5]
          Rhosbase=3
          Rhomean=10
          grav=[0,5,10,25]
          gravbase=10
          shear =[0,1.5,2,5]
          shearbase=2
          XX = nsigmaplots(kmin, kmax, nk, Rhovec, Rhosbase, Rhomean, grav, gravbase, shear, sh

          f, axs = plt.subplots(1, 3, figsize=(18, 5))
          gs = gridspec.GridSpec(1, 3)
          axs[0], axs[1], axs[2]= plt.subplot(gs[0, 0]), plt.subplot(gs[0, 1]), plt.subplot(gs
          axs[0], axs[1], axs[2]= plt.subplot(1, 3, 1), plt.subplot(1, 3, 2), plt.subplot(1, 3

          axs[0].plot(XX[:,0], XX[:,1],'.',label=shear[0])
          axs[0].plot(XX[:,0], XX[:,2],'.',label=shear[1])
          axs[0].plot(XX[:,0], XX[:,3],'.',label=shear[2])
          axs[0].plot(XX[:,0], XX[:,4],'.',label=shear[3])
          axs[0].legend()
          axs[0].set_title('Shear Dependence (Lambda)',fontsize=12)
          axs[0].set_xlabel("k", fontsize=10)
          axs[0].set_ylabel("Sigma(k)", fontsize=10)

          axs[1].plot(XX[:,0], XX[:,5],'.',label=(1+Rhovec[0])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,6],'.',label=(1+Rhovec[1])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,7],'.',label=(1+Rhovec[2])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,8],'.',label=(1+Rhovec[3])/Rhomean)
          axs[1].legend()
          axs[1].set_title("Stratification Dependence (Rho_s+Rho_0/Rho_0)", fontsize=12)
          axs[1].set_xlabel("k", fontsize=10)

          axs[2].plot(XX[:,0], XX[:,9],'.',label=grav[0])
          axs[2].plot(XX[:,0], XX[:,10],'.',label=grav[1])
          axs[2].plot(XX[:,0], XX[:,11],'.',label=grav[2])
          axs[2].plot(XX[:,0], XX[:,12],'.',label=grav[3])
          axs[2].legend()
          axs[2].set_title("Gravity Dependence (g)", fontsize=12)
          axs[2].set_xlabel("k", fontsize=10);
```



Above are plots of the same conditions that were analysed analytically on scales to $k = 1$ and $k = 2$. The most immediate difference to notice is that the growth rates for the numerical instability are significantly smaller. There are also weak hints of bimodality in the numerical results and appears to be more sensitivity to the strength of stratification, with only one of the numerical $\rho_s$ values resulting in instability. (Plotted together below)

```
In [365]: kmin=0.01
          kmax=2
          nk=200 #needs to stay as 200 to fit the A matrix construction
          Rhovec=[0,1,3,5]
          Rhosbase=3
          Rhomean=10
          grav=[0,5,10,25]
          gravbase=10
          shear =[0,1.5,2,5]
          shearbase=2
          XX = sigmaplots(kmin, kmax, nk, Rhovec, Rhosbase, Rhomean, grav, gravbase, shear, she
          XX2 = nsigmaplots(kmin, kmax, nk, Rhovec, Rhosbase, Rhomean, grav, gravbase, shear, s

          f, axs = plt.subplots(1, 6, figsize=(18, 12))
          gs = gridspec.GridSpec(2, 3)
          axs[0], axs[1], axs[2], axs[3], axs[4], axs[5]= plt.subplot(gs[0, 0]), plt.subplot(gs
          axs[0], axs[1], axs[2], axs[3], axs[4], axs[5]= plt.subplot(2, 3, 1), plt.subplot(2,

          axs[0].plot(XX[:,0], XX[:,1],'.',label=shear[0])
          axs[0].plot(XX[:,0], XX[:,2],'.',label=shear[1])
          axs[0].plot(XX[:,0], XX[:,3],'.',label=shear[2])
          axs[0].plot(XX[:,0], XX[:,4],'.',label=shear[3])
          axs[0].legend()
          axs[0].set_title('Shear Dependence (Analytic)',fontsize=12)
          axs[0].set_xlabel("k", fontsize=10)
          axs[0].set_ylabel("Sigma(k)", fontsize=10)

          axs[1].plot(XX[:,0], XX[:,5],'.',label=(1+Rhovec[0])/Rhomean)
          axs[1].plot(XX[:,0], XX[:,6],'.',label=(1+Rhovec[1])/Rhomean)
```

```
axs[1].plot(XX[:,0], XX[:,7],'.',label=(1+Rhovec[2])/Rhomean)
axs[1].plot(XX[:,0], XX[:,8],'.',label=(1+Rhovec[3])/Rhomean)
axs[1].legend()
axs[1].set_title("Stratification Dependence (Analytic)", fontsize=12)
axs[1].set_xlabel("k", fontsize=10)

axs[2].plot(XX[:,0], XX[:,9],'.',label=grav[0])
axs[2].plot(XX[:,0], XX[:,10],'.',label=grav[1])
axs[2].plot(XX[:,0], XX[:,11],'.',label=grav[2])
axs[2].plot(XX[:,0], XX[:,12],'.',label=grav[3])
axs[2].legend()
axs[2].set_title("Gravity Dependence (Analytic)", fontsize=12)
axs[2].set_xlabel("k", fontsize=10)

axs[3].plot(XX2[:,0], XX2[:,1],'.',label=shear[0])
axs[3].plot(XX2[:,0], XX2[:,2],'.',label=shear[1])
axs[3].plot(XX2[:,0], XX2[:,3],'.',label=shear[2])
axs[3].plot(XX2[:,0], XX2[:,4],'.',label=shear[3])
axs[3].legend()
axs[3].set_title('Shear Dependence (Numerical)',fontsize=12)
axs[3].set_xlabel("k", fontsize=10)
axs[3].set_ylabel("Sigma(k)", fontsize=10)

axs[4].plot(XX2[:,0], XX2[:,5],'.',label=(1+Rhovec[0])/Rhomean)
axs[4].plot(XX2[:,0], XX2[:,6],'.',label=(1+Rhovec[1])/Rhomean)
axs[4].plot(XX2[:,0], XX2[:,7],'.',label=(1+Rhovec[2])/Rhomean)
axs[4].plot(XX2[:,0], XX2[:,8],'.',label=(1+Rhovec[3])/Rhomean)
axs[4].legend()
axs[4].set_title("Stratification Dependence (Numerical)", fontsize=12)
axs[4].set_xlabel("k", fontsize=10)

axs[5].plot(XX2[:,0], XX2[:,9],'.',label=grav[0])
axs[5].plot(XX2[:,0], XX2[:,10],'.',label=grav[1])
axs[5].plot(XX2[:,0], XX2[:,11],'.',label=grav[2])
axs[5].plot(XX2[:,0], XX2[:,12],'.',label=grav[3])
axs[5].legend()
axs[5].set_title("Gravity Dependence (Numericl)", fontsize=12)
axs[5].set_xlabel("k", fontsize=10);
```



The instability of the profile relies on the interaction of perturbations on each density boundary. This is not captured as well with numerical methods as solving for the growth rates analytically. This is likely due to the resolution and profile approximations in the numerical method. The importance of stratification in the problem is coupled strongly with gravity determining the stability of the initial profile and the growth of a perturbation. Both factors are necessary for instability to occur but with increased strength they lead to stability. Strength of the velocity shear in the profile displayed the same characteristics - necessary but leading to stability when too strong.

## 6 Plotting Functions

```
In [364]: def sigmaplots(kmin, kmax, nk,Rhovec, Rhosbase, Rhomean, grav, gravbase, shear, shea
          #kmin 1,kmax 1,dk 1, Rhovec 4, Rhosbase 1, Rhomean 1, grav 4, gravbase1 , shear 4, s

              k=np.linspace(kmin,kmax,nk)

              sigmavalues=np.zeros((nk,13))
              sigmavalues[:,0]=k
              #plot 1 is the shear dependence
              for ii in range(4):
                  Lambda=shear[ii]
                  g=-gravbase
                  Rhos=Rhosbase

                  for j in range(nk):
```

```
        a4 = k[j]**2*((-1 + np.exp(4*k[j]))*Rhos**2 - 4*np.exp(4*k[j])*Rhomean**
        a3 = 4*1j*np.exp(4*k[j])*k[j]**2*Rhos*Lambda*Rhomean
        a2 = k[j]**2*((-1 + np.exp(4*k[j]))*(1 + 2*k[j] + 2*k[j]**2)* Rhos**2*La
        a1 = -2*1j*k[j]**3*Rhos*Lambda*((-1 + np.exp(4*k[j]))*g*(1 + 2*k[j])*Rho
        a0 = k[j]**4*(-(-1 + np.exp(4*k[j]))*g**2*Rhos**2 - 4*np.exp(4*k[j])*g*k

        root = max(np.real(np.roots([a4, a3, a2, a1, a0])),default=0)
        sigmavalues[j,1+ii]=root


#plot 2 is the stratification dependence
for ii in range(3):
    Lambda=shearbase
    g=-gravbase
    Rhos=Rhovec[ii]

    for j in range(nk):

        a4 = k[j]**2*((-1 + np.exp(4*k[j]))*Rhos**2 - 4*np.exp(4*k[j])*Rhomean**
        a3 = 4*1j*np.exp(4*k[j])*k[j]**2*Rhos*Lambda*Rhomean
        a2 = k[j]**2*((-1 + np.exp(4*k[j]))*(1 + 2*k[j] + 2*k[j]**2)* Rhos**2*La
        a1 = -2*1j*k[j]**3*Rhos*Lambda*((-1 + np.exp(4*k[j]))*g*(1 + 2*k[j])*Rho
        a0 = k[j]**4*(-(-1 + np.exp(4*k[j]))*g**2*Rhos**2 - 4*np.exp(4*k[j])*g*k

        root = max(np.real(np.roots([a4, a3, a2, a1, a0])),default=0)
        sigmavalues[j,5+ii]=root


#plot 3 is the stratification dependence
for ii in range(4):
    Lambda=shearbase
    g=-grav[ii]
    Rhos=Rhosbase

    for j in range(nk):

        a4 = k[j]**2*((-1 + np.exp(4*k[j]))*Rhos**2 - 4*np.exp(4*k[j])*Rhomean**
        a3 = 4*1j*np.exp(4*k[j])*k[j]**2*Rhos*Lambda*Rhomean
        a2 = k[j]**2*((-1 + np.exp(4*k[j]))*(1 + 2*k[j] + 2*k[j]**2)* Rhos**2*La
        a1 = -2*1j*k[j]**3*Rhos*Lambda*((-1 + np.exp(4*k[j]))*g*(1 + 2*k[j])*Rho
        a0 = k[j]**4*(-(-1 + np.exp(4*k[j]))*g**2*Rhos**2 - 4*np.exp(4*k[j])*g*k

        root = max(np.real(np.roots([a4, a3, a2, a1, a0])),default=0)
        sigmavalues[j,9+ii]=root

return sigmavalues
```

```
In [359]: def nsigmaplots(kmin, kmax, nk,Rhovec, Rhosbase, Rhomean, grav, gravbase, shear, shea
          #kmin 1,kmax 1,dk 1, Rhovec 4, Rhosbase 1, Rhomean 1, grav 4, gravbase1 , shear 4, s

              k=np.linspace(kmin,kmax,nk)

              nsigmavalues=np.zeros((nk,13))
              nsigmavalues[:,0]=k
              #plot 1 is the shear dependence
              for ii in range(4):
                  Lambda=shear[ii]
                  g=gravbase
                  Rhos=Rhosbase
                  Rho0=Rhomean
                  Rho= (Rho0 - (RhoS/2)*(np.tanh((Z+1)/0.01)+np.tanh((Z-1)/0.01))) * np.exp(0*
                  for j in range(nk):

                      Acomplete=construct_Akop(k[j],Lambda,g)
                      eigenValues, eigenVectors = np.linalg.eig(Acomplete)
                      nsigmavalues[j,1+ii]=max(np.real(eigenValues))


              #plot 2 is the stratification dependence
              for ii in range(3):
                  Lambda=shearbase
                  g=gravbase
                  Rhos=Rhovec[ii]
                  Rho0=Rhomean
                  Rho= (Rho0 - (RhoS/2)*(np.tanh((Z+1)/0.01)+np.tanh((Z-1)/0.01))) * np.exp(0*
                  for j in range(nk):
                      Acomplete=construct_Akop(k[j],Lambda,g)
                      eigenValues, eigenVectors = np.linalg.eig(Acomplete)
                      nsigmavalues[j,5+ii]=max(np.real(eigenValues))


              #plot 3 is the stratification dependence
              for ii in range(4):
                  Lambda=shearbase
                  g=grav[ii]
                  Rhos=Rhosbase
                  Rho0=Rhomean
                  Rho= (Rho0 - (RhoS/2)*(np.tanh((Z+1)/0.01)+np.tanh((Z-1)/0.01))) * np.exp(0*
                  for j in range(nk):

                      Acomplete=construct_Akop(k[j],Lambda,g)
                      eigenValues, eigenVectors = np.linalg.eig(Acomplete)
                      nsigmavalues[j,9+ii]=max(np.real(eigenValues))

              return nsigmavalues
```