Hydrostab: a universal code for solving hydrodynamic stability problems

Han-Yu Ye, Li-jun Yang, Qing-fei Fu

hanyuye@buaa.edu.cn
yanglijun@buaa.edu.cn
fuqingfei@buaa.edu.cn
School of Astronautics, Beihang University, Beijing, China

Notice: Please cite the following paper if you use ‘hydrostab’.

Han-Yu Ye, Li-Jun Yang, and Qing-Fei Fu. Spatial instability of viscous double-layer liquid sheets. Physics of fluids 28, 102101(2016).

The authors have developed a universal code for solving hydrodynamic stability problems. The code is based on MATLAB and takes advantage of MATLAB’s support for object-oriented programming, especially its support for operator overloading. To solve a custom hydrodynamic stability problem, the user only needs to write an M-file containing descriptions of the governing equations and boundary conditions. The syntax for describing the equations closely resembles these equations themselves. Therefore this code will provide great convenience for user to investigate custom hydrodynamic stability problems. The code performs a normal mode analysis on linearized governing equations and boundary conditions, converting the stability problem into a generalized eigenvalue problem, and takes advantage of MATLAB’s powerful eigenvalue algorithms. Spatial discretization is performed by spectral collocation methods, hence offers high precision. The code supports two-dimensional problems in Cartesian coordinates. The code has been tested on MATLAB 6.5 and later versions of MATLAB.

I. NOTICE

This software relies on the 'MATLAB Differentiation Matrix Suite' developed by Weideman and Reddy. You can find that software on MATLAB Central.

For convenience, the files of the 'MATLAB Differentiation Matrix Suite' are included in this .zip file. The 'MATLAB Differentiation Matrix Suite' includes the following files:

ce0.m cerfa.m cerfb.m cheb2bc.m

cheb4c.m	chebdif.m	chebdifft.m	chebint.m	contents.m
fourdif.m	fourdifft.m	fourint.m	herdif.m	herroots.m
lagdif.m	lagroots.m	legroots.m	matplot.m	
orrsom.m	poldif.m	polint.m	schrod.m	sgrhs.m
sincdif.m	sincdifft.m	sineg.m		

These files are developed by Weideman and Reddy, not the current author, and they are downloaded from

https://cn.mathworks.com/matlabcentral/fileexchange/29-dmsuite

in January 2016. You can search for 'MATLAB Differentiation Matrix Suite' in MATLAB Central for updated versions.

II. INTRODUCTION

Hydrodynamic stability is an important topic in fluid mechanics. One of the basic approaches to analyze hydrodynamic stability is linear stability analysis. The theory of linear stability analysis is pretty mature. For simple problems, such as the stability of Poiseuille flow between two infinite plates (Orszag[1]), it is straightforward for someone to write a computer code performing stability analysis. However, for problems which are a little more complicated, such as those involve multiple regions (e.g., stability of liquid sheets moving in a gas medium, see Fig. 1) or involve extra governing equations (e.g., energy equation), it is not so straightforward for someone to write a computer code from scratch. Motivated by the need of the author's research group, the author develops a universal code for solving hydrodynamic stability problems, based on the MATLAB programming language. The code performs a normal mode analysis on linearized governing equations and boundary conditions, converting the stability problem into a generalized eigenvalue problem, and takes advantage of MATLAB's powerful eigenvalue algorithms. Spatial discretization is performed by spectral collocation methods, hence offers high precision. The name of the code, 'hydrostab', is based on the first few letters of the two words 'hydrodynamic' and 'stability'.

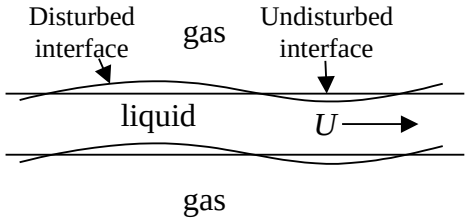


FIG. 1. Example of hydrodynamic stability problems involving multiple regions: stability of liquid sheets moving in a gas medium.

III. A SIMPLE AND DETAILED EXAMPLE

Consider the simple example of inviscid Rayleigh-Taylor instability. (Fig. 2) For the undisturbed state, both fluids are at rest and the interface is located at $y=0$. The gravitational acceleration is g and pointing downward. The surface tension at the interface is σ.

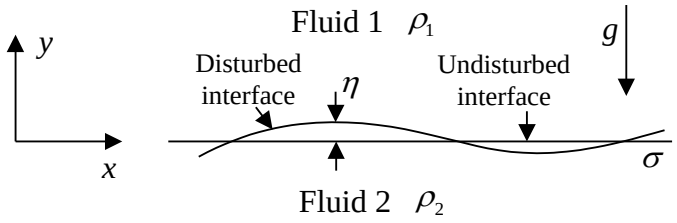


FIG. 2. Rayleigh-Taylor instability: problem schematic.

Since both fluids are inviscid and irrotational initially, they are also irrotational at later times. Therefore, the disturbance to the upper and lower fluids can be represented by disturbance potential functions φ_1 and φ_2, respectively. The disturbance to the interface is represented by η. The linearized governing equations and boundary conditions for linear stability analysis are (The details of the linearizing process is omitted; the reader can found them in many textbooks on fluid mechanics or hydrodynamic stability.)

Linearized governing equations:

$$\frac{\partial^2\varphi_1}{\partial x^2}+\frac{\partial^2\varphi_1}{\partial y^2}=0,\quad y>0, \tag{1}$$

$$\frac{\partial^2\varphi_2}{\partial x^2}+\frac{\partial^2\varphi_2}{\partial y^2}=0,\quad y<0, \tag{2}$$

Linearized boundary conditions:

$$\text{Kinematic boundary conditions}\quad \left.\frac{\partial\varphi_1}{\partial y}\right|_{y=0}=\frac{\partial\eta}{\partial t},\quad \left.\frac{\partial\varphi_2}{\partial y}\right|_{y=0}=\frac{\partial\eta}{\partial t}, \tag{3}$$

$$\text{Dynamic boundary condition}\quad \rho_1\left(\left.\frac{\partial\varphi_1}{\partial t}\right|_{y=0}+g\eta\right)=\rho_2\left(\left.\frac{\partial\varphi_2}{\partial t}\right|_{y=0}+g\eta\right)-\sigma\frac{\partial^2\eta}{\partial x^2}. \tag{4}$$

There are also conditions that the disturbances decay to zero at $y\to+\infty$ and $y\to-\infty$. But they need not to be specified in the computer program since the Laguerre spectral method assumes that the variables decay to zero at infinity. To solve Equations (1)-(4), the user can simply write an MATLAB script as follows. The script should be placed in the folder ‘temporal’ since we are performing a temporal stability analysis.

Program 1: inviscid Rayleigh-Taylor instability

```
1   clear
2   sigma=0.037;
3   rho1=1000;
4   rho2=800;
5   g=9.8;
6   hs=hydrostab(2);
7   hs=set_interval(hs,1,0,inf,50);
8   hs=set_interval(hs,2,-inf,0,50);
9   hs=setscale(hs,1,10);
10  hs=setscale(hs,2,10);
11  n=50;
12  arrk=linspace(0,280,n);
13  for i=1:n
14      hs=setk(hs,arrk(i));
15      hs=addvar(hs,1,'phi1');
16      hs=addvar(hs,2,'phi2');
17      hs=addvarxt(hs,'eta');
18      hs=addeq(hs,dx2(phi1)+dy2(phi1)==0,'ignorefirstpoint');
19      hs=addeq(hs,dx2(phi2)+dy2(phi2)==0,'ignorelastpoint');
20      hs=addeq(hs,dyat(phi1,0)==dt(eta));
21      hs=addeq(hs,dyat(phi2,0)==dt(eta));
22      hs=addeq(hs,rho1*(dtat(phi1,0)+g*eta)==rho2*(dtat(phi2,0)+g*eta)-sigma*dx2(eta));
23      hs=compute(hs);
24      eige=eigenvalue(hs,1e8);
25      plot(arrk(i)*ones(length(eige),1),imag(eige),'k+');
26      hold on
27  end
28  axis([0 280 0 10]);
```

Line 2 to 5 defines the parameters of the problem. The parameter values $\rho_1 = 1000\text{kg}\cdot\text{m}^{-3}$, $\rho_2 = 800\text{kg}\cdot\text{m}^{-3}$ and $\sigma = 0.037\text{N}\cdot\text{m}^{-1}$ approximately correspond to a system consisting water (upper) and oil (lower). Line 6 creates a hydrostab object named ‘hs’. This will invoke the code developed by the author. The parameter ‘2’ means that the problem involves two regions. Line 7 sets the range of Region 1 to be $y \in [0, +\infty)$. The fourth parameter ‘50’ means that there are 50 collocation points for Region 1. Similarly, Line 8 sets the range of Region 2 to be $y \in (-\infty, 0]$ and the number of collocation points 50. In this example, both regions are semi-infinite. For problems posed on semi-infinite intervals, the code will automatically choose the Laguerre spectral method. Although the intervals are infinite, the number of collocation points is finite. Therefore, the user should give an additional parameter to determine the position of the farthest collocation point. This is done in Lines 9 and 10. The parameter ‘10’ in these two lines means that the distance between the first and the last collocation point is 10. This is illustrated in Fig. 3. As demonstrated later, this parameter has great impact on the accuracy of the result.

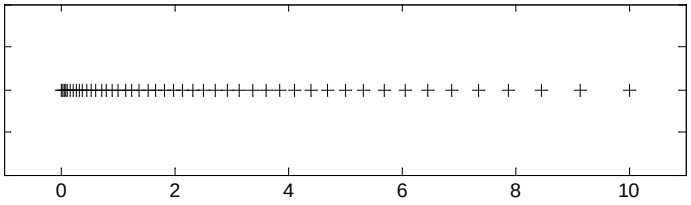


FIG. 3. Positions of collocation points for Region 1.

Lines 13 to 27 is a for-loop, in which the wavenumber k varies between 0 and 280. The normal mode employed in hydrostab is $\exp(\mathrm{i}kx-\mathrm{i}\omega t)$, therefore the imaginary part of the frequency ω represents temporal growth rate. At the beginning of each loop, the wavenumber is set (Line 14). (The user could also set a complex wavenumber with nonzero real part as well as nonzero imaginary part, which is useful when performing spatial-temporal stability analysis.) Then, Line 15 adds the variables in Region 1. All variables appear in the governing equations and boundary conditions with the normal mode factor $\exp(\mathrm{i}kx-\mathrm{i}\omega t)$ should be added. For this simple example, in Region 1 there is only one variable with the normal mode factor, i.e. φ_1. Similarly, Line 16 adds the variables in Region 2.

Line 17 adds the variables with the normal mode factor but only functions of x and t, i.e. not functions of y. In this case, the interface displacement η belongs to this class. Lines 18 and 19 add the linearized governing equations. It can be seen that the syntax for describing the equations closely resembles these equations themselves. The only thing the user should take care is that the equal sign in the equation should be represented by the comparison operator '=='. The option 'ignorefirstpoint' in Line 18 tells hydrostab to ignore the governing equation at the first collocation point in the corresponding region. In this case, it means that the governing equation at $y=0$ is ignored. This is a technique to keep the total number of equations equal to the total number of unknowns. Similarly, in Line 19 the option 'ignorelastpoint' is used, and it also means the governing equation at $y=0$ is ignored. (In hydrostab, the collocation points are always sorted in ascending oreder.)

Lines 20-22 add the linearized boundary conditions. Again, the syntax for describing the boundary conditions closely resembles the boundary conditions themselves. To represent the value of something at a value of y, the user simply needs to add the 'at' suffix. For instance, "dyat(phi1,0)" means

$$\left.\frac{\partial\varphi_1}{\partial y}\right|_{y=0}$$

Line 23 invokes the function compute() to compute the eigenvalues, namely the frequency ω in the normal mode. Line 24 gets the results by invoking the eigenvalue() function. The parameter '1e8' is a threshold to reject the unreasonable eigenvalues.

The necessity of a threshold is explained as follows. By the normal mode analysis, the stability problem is converted into a generalized eigenvalue problem

$$\mathbf{A}\mathbf{x} = \omega\mathbf{B}\mathbf{x} \tag{5}$$

where $\mathbf{A}$ and $\mathbf{B}$ are square matrices and $\mathbf{x}$ is the eigenvector. Since the normal mode is $\exp(\mathrm{i}kx - \mathrm{i}\omega t)$, an equation with time derivative $\partial(\ )/\partial t$ will result in a nonzero row in the matrix $\mathbf{B}$. However, not all equations contains time derivative. Therefore, the matrix $\mathbf{B}$ is rank deficient and the result of the eigenvalue problem contains infinite eigenvalues. In numerical computation, some of the infinite eigenvalues will not exactly be infinity but some very large values. Therefore, we need a threshold to reject these unreasonable eigenvalues. The eigenvalue of which the absolute value is greater than the threshold will be rejected.

Line 25 plots the imaginary part of ω. Line 26 holds the current figure so that the current figure will not be replaced by the figure produced at the next wavenumber. As a result, the figures from all wavenumbers superimpose to form the dispersion curve. Line 28 sets the range of the coordinate axes so that only the unstable eigenvalues ($\mathrm{Im}\,\omega > 0$) are displayed.

The result is shown in Fig. 4. For this problem, analytical result exists: (See, e.g. Charru[2])

$$\omega = k\left(\frac{g}{k}\frac{\rho_2 - \rho_1}{\rho_1 + \rho_2} + \frac{\sigma\kappa}{\rho_1 + \rho_2}\right)^{1/2}, \tag{6}$$

hence in Fig. 4 the analytical result is superimposed on the numerical result to verify it. The user can plot the analytical result by typing the following statements in the MATLAB command window:

```
k=linspace(1e-6,280,1000);
omega=k.*sqrt(g./k*(rho2-rho1)/(rho1+rho2)+sigma*k/(rho1+rho2));
plot(k,imag(omega),'k-');
```

The minimal wavenumber is 1e-6 in order to avoid a divide-by-zero error. It can be seen that the result computed by hydrostab agrees well with the analytical result. With the increase of wavenumber, the growth rate first increases then decreases, and there exists a cutoff wavenumer

$$k_c = \sqrt{\frac{g(\rho_1 - \rho_2)}{\sigma}}, \tag{7}$$

The growth rate goes to zero at $k = k_c$. The mechanism is that surface tension damps the high wavenumber disturbances; if the surface tension is zero, the growth rate will be positive for any wavenumber.

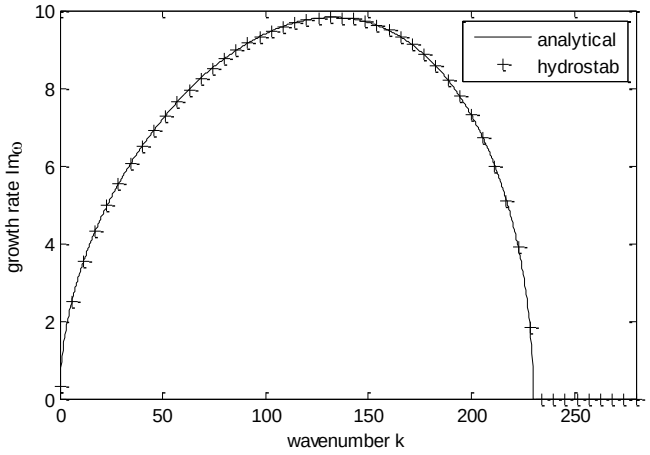
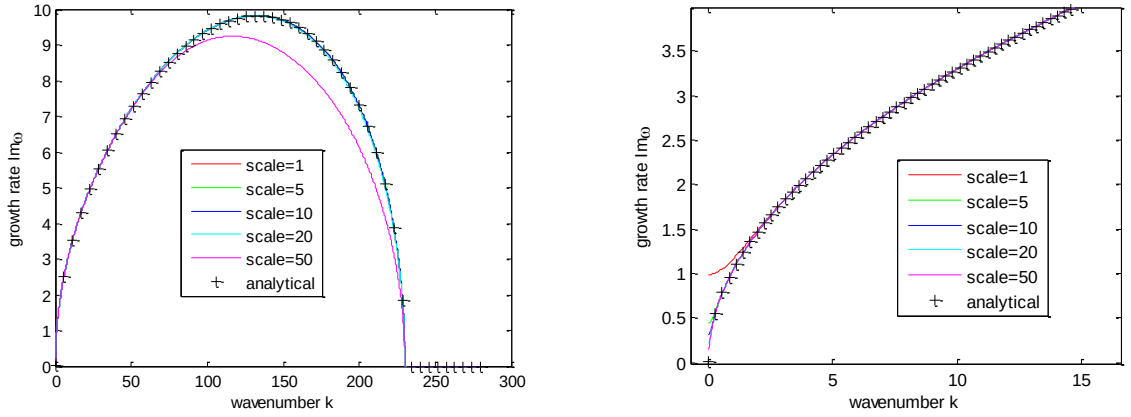


FIG. 4. Results of Program 1. The analytical result is also superimposed.

The argument in the function setscale() controls the distribution of the collocation points, and it can affect the accuracy of the result. Fig. 5 shows the results with different values for the argument 'scale' in setscale(). (As mentioned above, this argument represents the distance between the first and the last collocation point.) It can be seen that accuracy is lost when the argument 'scale' is too large or too small. More specifically, for large wavenumbers, the accuracy is lost when the 'scale' is too large; for wavenumbers near $k=0$, accuracy is lost when the 'scale' is too small.



(a) (b)

FIG. 5. Results with different values for the argument in setscale(). (a) global view; (b) at the vicinity of $k=0$.

The reason can be explained by plotting the disturbance potential functions. The disturbance potential functions are represented by the normal mode as

$$\varphi_1=\hat{\varphi}_1\exp\left(\mathrm{i}kx-\mathrm{i}\omega t\right),\quad \varphi_2=\hat{\varphi}_2\exp\left(\mathrm{i}kx-\mathrm{i}\omega t\right), \tag{8}$$

and we can slightly modify Program 1 to plot the complex amplitude $\hat{\varphi}_1$:

Program 2: inviscid Rayleigh-Taylor instability; plot the complex amplitude

```
clear
sigma=0.037;
rho1=1000;
rho2=800;
g=9.8;
hs=hydrostab(2);
hs=set_interval(hs,1,0,inf,50);
hs=set_interval(hs,2,-inf,0,50);
hs=setscale(hs,1,10);
hs=setscale(hs,2,10);
arrk=[160 2];
for i=1:2
    hs=setk(hs,arrk(i));
    hs=addvar(hs,1,'phi1');
    hs=addvar(hs,2,'phi2');
    hs=addvarxt(hs,'eta');
    hs=addeq(hs,dx2(phi1)+dy2(phi1)==0,'ignorefirstpoint');
    hs=addeq(hs,dx2(phi2)+dy2(phi2)==0,'ignorelastpoint');
    hs=addeq(hs,dyat(phi1,0)==dt(eta));
    hs=addeq(hs,dyat(phi2,0)==dt(eta));
    hs=addeq(hs,rho1*(dtat(phi1,0)+g*eta)==rho2*(dtat(phi2,0)+g*eta)-sigma*dx2(eta));
    hs=compute(hs);
    eige=eigenvalue(hs,1e8);
    subplot(2,1,i);
    index=find(imag(eige)>0);
    index=index(1);
    [arrphi,arry]=retrieve(hs,phi1,eige(index));
    plot(arry,abs(arrphi),'b-+');
end
```

The statements in Lines 25 and 26 identify the eigenvalue with positive growth rate. Line 27 invokes the retrieve() function to obtain the results of φ_1 for that eigenvalue. The number of output arguments of the retrieve() function depends on which variable the user is intended to obtain results. If the variable is only functions of x and t, the retrieve() function will return a single value; if the variable is functions of x, y and t, the retrieve() function return two arrays, the first array stores the results of that variable, and the second array stores the collocation points of the corresponding interval. In this case, since φ_1 is functions of x, y and t, the retrieve() function returns two arrays, 'arrphi' and 'arry'. 'arry' stores the collocation points in the interval $[0,+\infty)$. 'arrphi' stores the values of the complex amplitude $\hat{\varphi}_1$ at these collocation points. Line 28 plots the absolute value of $\hat{\varphi}_1$.

The result is shown in Fig. 6. At small wavenumber ($k=2$), the disturbance decays rather slowly. In contrast, at large wavenumber ($k=160$), the disturbance decays quickly, and becomes negligible at a small value of y, i.e., positions very near the interface. Therefore, for better accuracy, the range of the collocation points should be larger at small wavenumber but should be smaller at large wavenumber. And this is the reason for the phenomenon observed in Fig. 5.

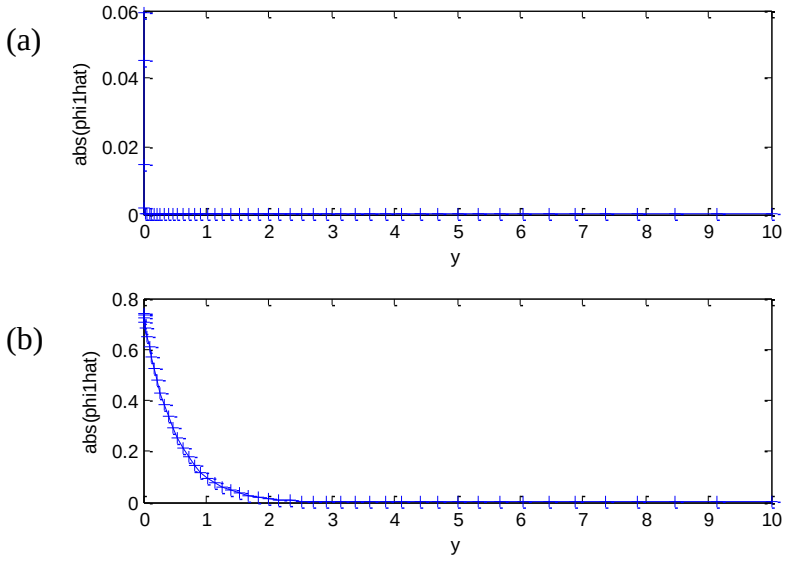


FIG. 6. Absolute value of the complex amplitude $\hat{\varphi}_1$. (a) $k=160$; (b) $k=2$.

IV. MORE EXAMPLES

In this section, more examples are presented, but they are not presented as detailed as the example above. Examples A to E are temporal stability analysis, hence the MATLAB scripts should be placed in the folder 'temporal'. Example F is a spatial stability analysis and the MATLAB script should be placed in the folder 'spatial'.

A. Stability of plane Poiseuille flow

This is a classic problem which has been investigated by many authors. For instance, Orszag[1] investigated it by the Chebyshev spectral method and obtained an accurate critical Reynolds number, $Re = 5722.22$. Here the author presents this example in order to demonstrate the ability of hydrostab to deal with a non-uniform velocity profile. The flow is between two infinite plates located at $y = \pm 1$. (Fig. 7) The velocity profile for the undisturbed flow is

$$U = 1 - y^2 \tag{9}$$

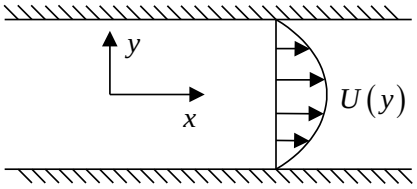


FIG. 7. Plane poiseuille flow.

The linearized governing equations are

$$\frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} + v\frac{\mathrm{d}U}{\mathrm{d}y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad -1<y<1, \tag{10}$$

$$\frac{\partial v}{\partial t} + U\frac{\partial v}{\partial x} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \quad -1<y<1, \tag{11}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad -1<y<1. \tag{12}$$

where u and v are disturbance velocities and p is disturbance pressure. The linearized boundary conditions are

$$u\big|_{y=1} = 0, \quad v\big|_{y=1} = 0, \quad u\big|_{y=-1} = 0, \quad v\big|_{y=-1} = 0. \tag{13}$$

The MATLAB script the user needs to write is listed as follows. The Reynolds number is set to be $Re = 5722.22$, the critical Reynolds number.

Program 3: Stability of plane Poiseuille flow

```
1  clear
2  Re=5772.22;
```

```
3   hs=hydrostab(1);
4   hs=set_interval(hs,1,-1,1,50);
5   nk=100;
6   arrk=linspace(0,1.7,nk);
7   U=func('1-y^2');
8   dU=func('-2*y');
9   for i=1:nk
10      hs=setk(hs,arrk(i));
11      hs=addvar(hs,1,'u','v','p');
12      hs=addeq(hs,dt(u)+U*dx(u)+v*dU==-dx(p)+1/Re*(dx2(u)+dy2(u)),'ignoreendpoints');
13      hs=addeq(hs,dt(v)+U*dx(v)==-dy(p)+1/Re*(dx2(v)+dy2(v)),'ignoreendpoints');
14      hs=addeq(hs,dx(u)+dy(v)==0);
15      hs=addeq(hs,at(u,1)==0);
16      hs=addeq(hs,at(v,1)==0);
17      hs=addeq(hs,at(u,-1)==0);
18      hs=addeq(hs,at(v,-1)==0);
19      hs=compute(hs);
20      eige=eigenvalue(hs,1e8);
21      plot(arrk(i)*ones(length(eige),1),imag(eige),'k+');
22      hold on
23  end
24  axis([0 1.7 -0.06 0.02]);
25  grid on
```

Note that the user does not need to rewrite the linearized governing equations using the streamfunction representation. (The classical treatment is rewriting the governing equations using the streamfunction representation, and then, under the normal mode assumed, the Orr-Sommerfeld equation is obtained. The advantage of that treatment is that the resulting governing equation only contains one variable. However, the current code is capable of solving equations with multiple variables; hence the user does not need to follow the classical treatment.) The major new feature of hydrostab involved in this example is Lines 7 and 8, where the function func() is invoked to create a velocity profile.

The result of Program 3 is shown in Fig. 8. It can be seen that the maximum growth rate is exactly zero; therefore the correctness of the program is verified.

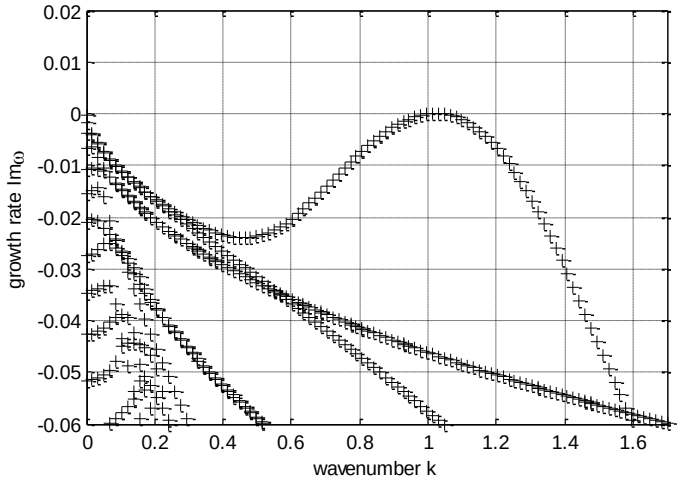


FIG. 8. Stability of plane Poiseuille flow. Growth rate vs wavenumber. $Re = 5772.22$.

It is worth point out that there is an alternative way to specify a velocity profile, i.e. using MATLAB function handles. To do this, the user should create two function M-files as follows (They must be stored at the current folder)

velocity.m

```
function U=velocity(y)
U=1-y^2;
```

dvelocity.m

```
function dU=dvelocity(y)
dU=-2*y;
```

and then modified Lines 7 and 8 of Program 3 as follows

```
U=func(@velocity);
dU=func(@dvelocity);
```

This method is somewhat more complicated, but the user must use this method if his/her profile contains parameters, as shown in the following example:

velocity.m

```
function U=velocity(y,a,b,c)
U=a*y^2+b*y+c;
```

dvelocity.m

```
function dU=dvelocity(y,a,b,c)
dU=2*a*y+b;
```

Lines 7 to 8 of Program 3 modified as

```
a=-1;
b=0;
c=1;
U=func(@velocity,a,b,c);
dU=func(@dvelocity,a,b,c);
```

It is also worth point out that, if the user-defined function can be vectorized, the user can specify an extra option in the constructor of hydrostab to enhance performance. For instance, the user can modify Line 3 of Program 3 as

```
hs=hydrostab(1,'vectorizedfunc');
```

and Line 7 as

```
U=func('1-y.^2');
```

B. Stability of a viscous sheet moving in an inviscid gas

This example demonstrates the ability of hydrostab to deal with multiple regions and multiple unstable modes. The physical model is shown in Fig. 9. This problem is first studied by Lin[3]. For the undisturbed state, the liquid sheet is moving at a speed of U and the gas is stationary. The Reynolds number is defined as $Re=\rho_l Ua/\mu$, the Weber number $We=\rho_l U^2 a/\sigma$ and the gas-to-liquid density ratio $\rho=\rho_g/\rho_l$, where ρ_l is liquid density, ρ_g gas density, a the half-thickness of the liquid sheet, μ the liquid viscosity, σ the surface tension. The dimensionless linearized governing equations and boundary conditions are (see, e.g. Yang *et al.*[4])

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right),\quad -1<y<1 \tag{14}$$

$$\frac{\partial v}{\partial t}+\frac{\partial v}{\partial x}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right),\quad -1<y<1 \tag{15}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0,\quad -1<y<1 \tag{16}$$

$$\frac{\partial^2 \varphi_{g1}}{\partial x^2}+\frac{\partial^2 \varphi_{g1}}{\partial y^2}=0,\quad y>1 \tag{17}$$

$$\frac{\partial^2 \varphi_{g2}}{\partial x^2}+\frac{\partial^2 \varphi_{g2}}{\partial y^2}=0,\quad y<-1 \tag{18}$$

$$v\big|_{y=1}=\frac{\partial \eta_1}{\partial t}+\frac{\partial \eta_1}{\partial x} \tag{19}$$

$$v\big|_{y=-1} = \frac{\partial \eta_2}{\partial t} + \frac{\partial \eta_2}{\partial x} \tag{20}$$

$$\left.\frac{\partial \varphi_{g1}}{\partial y}\right|_{y=1} = \frac{\partial \eta_1}{\partial t} \tag{21}$$

$$\left.\frac{\partial \varphi_{g2}}{\partial y}\right|_{y=-1} = \frac{\partial \eta_2}{\partial t} \tag{22}$$

$$-p\big|_{y=1} + \frac{2}{Re}\left.\frac{\partial v}{\partial y}\right|_{y=1} - \frac{1}{We}\frac{\partial^2 \eta_1}{\partial x^2} - \rho\left.\frac{\partial \varphi_{g1}}{\partial t}\right|_{y=1} = 0 \tag{23}$$

$$-p\big|_{y=-1} + \frac{2}{Re}\left.\frac{\partial v}{\partial y}\right|_{y=-1} + \frac{1}{We}\frac{\partial^2 \eta_2}{\partial x^2} - \rho\left.\frac{\partial \varphi_{g2}}{\partial t}\right|_{y=-1} = 0 \tag{24}$$

$$\left.\frac{\partial u}{\partial y}\right|_{y=1} + \left.\frac{\partial v}{\partial x}\right|_{y=1} = 0 \tag{25}$$

$$\left.\frac{\partial u}{\partial y}\right|_{y=-1} + \left.\frac{\partial v}{\partial x}\right|_{y=-1} = 0 \tag{26}$$

where u, v, p are disturbance velocities and pressure in liquid phase, $\varphi_{g1}, \varphi_{g2}$ are disturbance potential function in gas phase, and η_1, η_2 are interface displacements.

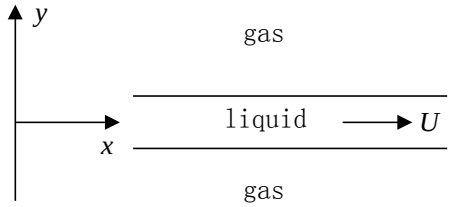


FIG. 9. A viscous sheet moving in an inviscid gas.

The MATLAB script the user needs to write is listed as follows.

Program 4: Stability of a viscous sheet moving in an inviscid gas

```
clear
Re=1000;
We=400;
rho=0.001;
hs=hydrostab(3);
hs=set_interval(hs,1,-inf,-1,20);
hs=set_interval(hs,2,-1,1,20);
hs=set_interval(hs,3,1,inf,20);
hs=setscale(hs,1,100);
hs=setscale(hs,3,100);
```

```
11  nk=50;
12  arrk=linspace(0,0.5,nk);
13  for i=1:nk
14      hs=setk(hs,arrk(i)); % wavenumber
15      hs=addvar(hs,2,'u','v','p');
16      hs=addvar(hs,3,'phig1');
17      hs=addvar(hs,1,'phig2');
18      hs=addvarxt(hs,'eta1','eta2');
19      % liquid phase
20      hs=addeq(hs,dt(u)+dx(u)==-dx(p)+1/Re*(dx2(u)+dy2(u)),'ignorepossible');
21      hs=addeq(hs,dt(v)+dx(v)==-dy(p)+1/Re*(dx2(v)+dy2(v)),'ignorepossible');
22      hs=addeq(hs,dx(u)+dy(v)==0);
23      % gas phase
24      hs=addeq(hs,dx2(phig1)+dy2(phig1)==0,'ignorepossible');
25      hs=addeq(hs,dx2(phig2)+dy2(phig2)==0,'ignorepossible');
26      % boundary conditions
27      hs=addeq(hs,at(v,1)==dt(eta1)+dx(eta1));
28      hs=addeq(hs,at(v,-1)==dt(eta2)+dx(eta2));
29      hs=addeq(hs,dyat(phig1,1)==dt(eta1));
30      hs=addeq(hs,dyat(phig2,-1)==dt(eta2));
31      hs=addeq(hs,-at(p,1)+2/Re*dyat(v,1)-1/We*dx2(eta1)-rho*dtat(phig1,1)==0);
32      hs=addeq(hs,-at(p,-1)+2/Re*dyat(v,-1)+1/We*dx2(eta2)-rho*dtat(phig2,-1)==0);
33      hs=addeq(hs,dyat(u,1)+dxat(v,1)==0);
34      hs=addeq(hs,dyat(u,-1)+dxat(v,-1)==0);
35      % compute eigenvalues
36      hs=compute(hs);
37      eige=eigenvalue(hs,1e8);
38      for j=1:length(eige)
39          if real(retrieve(hs,eta1,eige(j))/retrieve(hs,eta2,eige(j)))>0 % sinuous
40              plot(arrk(i),imag(eige(j)),'r+');
41          else % varicose
42              plot(arrk(i),imag(eige(j)),'b+');
43          end
44          hold on
45      end
46  end
47  axis([0 0.5 0 0.011]);
```

As specified in Lines 6 to 8, this problem involves three regions, two of which are semi-infinite, and the other is finite. hydrostab will deal with these kinds of mixed intervals properly—Laguerre spectral method will be adopted in the semi-infinite intervals while Chebyshev spectral method will be used in the finite interval.

For instability of plane liquid sheets, it is well known that there are two unstable modes, i.e. sinuous mode and varicose mode. For sinuous mode, displacements of the two interfaces are in-phase; for varicose mode, displacements of the two interfaces are anti-phase. (see Fig. 10)

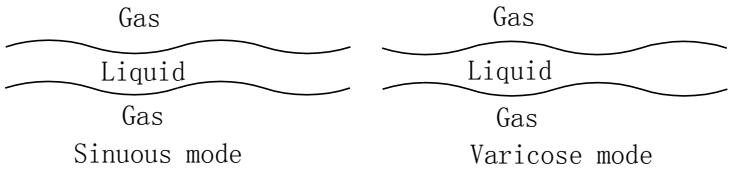


FIG. 10. Two different unstable modes in the instability of liquid sheets.

In order to distinguish these two modes, the statements for plotting are somewhat more complicated (Lines 38 to 45). The eigenvalue() function returns all the eigenvalues, eigenvalue for both sinuous mode and varicose mode are included. Hence a for-loop (Line 38) is designed to loop over all the eigenvalues. Each time through the loop, the displacements of the upper interface and the lower interface are obtained by invoking the retrieve() function (Line 39). If the ratio between them is positive, then that eigenvalue corresponds to the sinuous mode; otherwise, that eigenvalue corresponds to the varicose mode. Then, different colors are employed to plot the growth rate for the sinuous and varicose mode. (Note that the value returned by retrieve() is the complex amplitude $\hat{\eta}_1, \hat{\eta}_2$, where $\eta_1 = \hat{\eta}_1 \exp(\mathrm{i}kx - \mathrm{i}\omega t), \eta_2 = \hat{\eta}_2 \exp(\mathrm{i}kx - \mathrm{i}\omega t)$. Therefore we use the real() function to take its real part.)

The option 'ignorepossible' in Lines 20, 21, 24 and 25 means ignoring endpoints whenever possible. For a finite interval, this option means ignoring both endpoints. For a semi-infinite interval of $[a, +\infty)$ type, this options means ignoring the first point. For a semi-infinite interval of $(-\infty, a]$ type, this option means ignoring the last point. Therefore, sometimes this option is more convenient than explicitly writing 'ignoreendpoints', 'ignorefirstpoint' and 'ignorelastpoint'.

The result of Program 4 is shown in Fig. 11. It can be seen that both unstable modes have been successfully identified. The growth rate of sinuous mode is greater than that of varicose mode, which agrees with conclusions drawn in the literature. Similar to the previous example of Rayleigh-Taylor instability, there exists a cutoff wavenumber.

The analytical result for sinuous mode is (see, e.g. Yang *et al.*[4])

$$-\rho\omega^2 + \frac{k^3}{We} + \frac{\left(l^2 + k^2\right)^2}{Re^2} \tanh k - \left(\frac{2}{Re}\right)^2 lk^3 \tanh l \tag{27}$$

where

$$l^2 = k^2 + i\left(k - \omega\right) Re \tag{28}$$

In Fig. 11, the analytical result is also superimposed, and it can be observed that the numerical result agrees well with the analytical result.

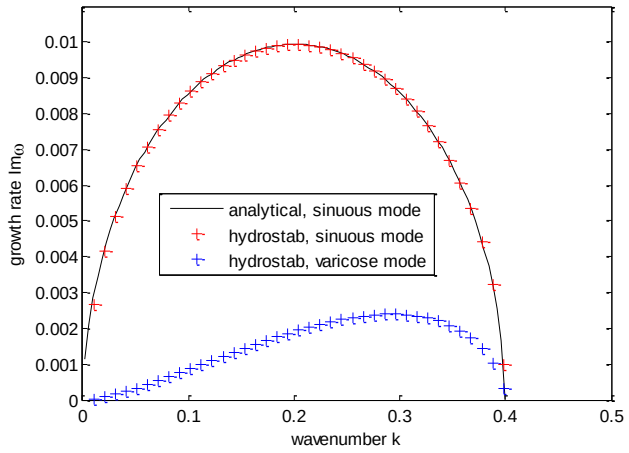


FIG. 11. Results of Program 4. The analytical result is also superimposed.

C. Two dimensional Rayleigh–Bénard convection

This example demonstrates how to deal with extra governing equations and plot the marginal stability curve. The schematic is shown in Fig. 12. The distance between two infinite plates is d. The temperatures of the upper plate and the lower plate are T_2 and T_1, respectively. ($T_1 > T_2$) The gravitational acceleration is g and pointing downward. The dimensionless Rayleigh number is defined as

$$Ra = \frac{g\beta\Delta T d^3}{\nu\kappa} \tag{29}$$

where β is the thermal expansion coefficient, ν is the kinematic viscosity, κ is the thermal diffusivity, and $\Delta T = T_1 - T_2$. The Prandtl number is defined as $Pr = \nu/\kappa$. The dimensionless linearized governing equations and boundary conditions are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{30}$$

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} + Pr\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{31}$$

$$\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial y} + Pr\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + Pr \cdot Ra \cdot \theta \tag{32}$$

$$\frac{\partial \theta}{\partial t} - v = \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \tag{33}$$

$$u\big|_{y=1} = 0,\ \ v\big|_{y=1} = 0,\ \ u\big|_{y=0} = 0,\ \ v\big|_{y=0} = 0, \tag{34}$$

$$\theta\big|_{y=1} = 0\,,\quad \theta\big|_{y=0} = 0\,. \tag{35}$$

where u and v are dimensionless disturbance velocities, p is dimensionless disturbance pressure and θ is dimensionless disturbance temperature. This example differs from the previous ones in that it contains an extra governing equation, i.e. the energy equation, Eq. (33).

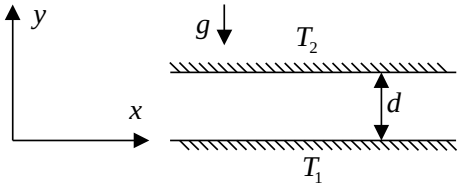


FIG. 12. Two dimensional Rayleigh–Bénard convection

The MATLAB script the user needs to write is listed as follows.

Program 5: Two dimensional Rayleigh–Bénard convection

```
clear
Pr=7;
nk=20;
arrk=linspace(0.1,8,nk);
nRa=20;
arrRa=linspace(0,8000,nRa);
[gridk,gridRa]=meshgrid(arrk,arrRa);
gridgrowth=zeros(nRa,nk);
hs=hydrostab(1);
hs=set_interval(hs,1,0,1,40);
for i=1:nk
    fprintf('%d of %d\n',i,nk);
    for j=1:nRa
        hs=setk(hs,arrk(i));
        hs=addvar(hs,1,'u','v','p','theta');
        hs=addeq(hs,dx(u)+dy(v)==0);
        hs=addeq(hs,dt(u)==-dx(p)+Pr*(dx2(u)+dy2(u)),'ignoreendpoints');
        hs=addeq(hs,dt(v)==-dy(p)+Pr*(dx2(v)+dy2(v))+Pr*arrRa(j)*theta,'ignoreendpoints');
        hs=addeq(hs,dt(theta)-v==dx2(theta)+dy2(theta),'ignoreendpoints');
        hs=addeq(hs,at(u,1)==0);
        hs=addeq(hs,at(v,1)==0);
        hs=addeq(hs,at(u,0)==0);
        hs=addeq(hs,at(v,0)==0);
        hs=addeq(hs,at(theta,1)==0);
        hs=addeq(hs,at(theta,0)==0);
        hs=compute(hs);
        eige=eigenvalue(hs,1e8);
        gridgrowth(j,i)=max(imag(eige));
    end
end
contour(gridk,gridRa,gridgrowth,[0 0],'k');
```

To create the marginal stability curve, the basic idea is to create a grid on the wavenumber-Rayleigh number plane, and then compute the growth rate at each grid point, and finally draw a contour plot with only one iso-line, i.e. the iso-line on which growth rate equals zero.

Note that the setk() (Line 14) must be placed in the inner for-loop, although the wavenumber only changes in the outer for-loop. This is because, in hydrostab, setk() not only sets the value of the wavenumber, but also clear the data of the previous computation. This operation is necessary at the beginning of each new computation.

The result is shown in Fig. 13, and it can be seen that it agrees with the existing results in the literature (See, e.g. Charru[2]), where the critical Rayleigh number is $Ra_c = 1708$, and the wave number at the critical point is $k_c = 3.117$.

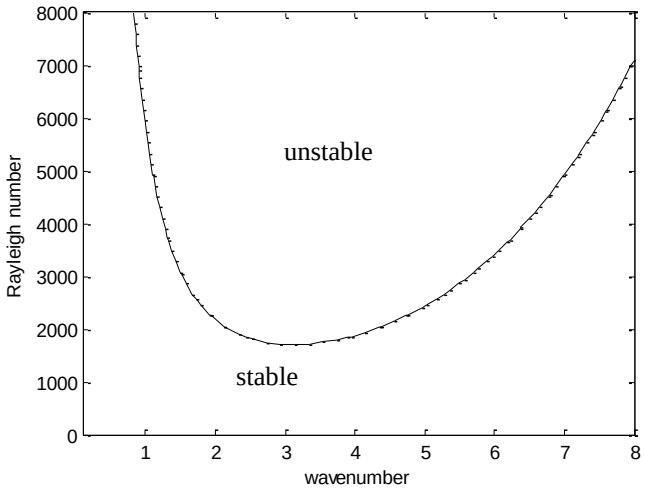


FIG. 13. Result of Program 5.

D. Instability of a shear layer

This example demonstrates application of hydrostab in infinite intervals. The problem is taken from Betchov and Szewczyk[5]. The schematic is shown in Fig. 14, where the undisturbed velocity profile is described by the hyperbolic tangent function:

$$U(y) = \tanh(y) \tag{36}$$

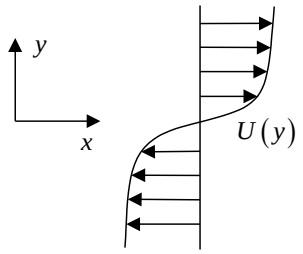


FIG. 14. Instability of a shear layer

The linearized governing equations are the same as those in the example "Stability of plane Poiseuille flow" (See Section A). Since the problem is posed on an infinite interval $y \in (-\infty, +\infty)$, there is no boundary conditions. The MATLAB script the user needs to write is listed as follows.

Program 6: Instability of a shear layer

```
clear
Re=20;
U=func('tanh(y)');
dU=func('1 - tanh(y).^2');
nscale=4;
arrscale=logspace(0.7,1.9,nscale);
arrcolor={'r','g','b','m'};
nk=50;
arrk=linspace(0,1,nk);
arrgrowthrate=zeros(1,nk);
for j=1:nscale
    fprintf('%d of %d\n',j,nscale);
    hs=hydrostab(1,'vectorizedfunc');
    hs=set_interval(hs,1,-inf,inf,70);
    hs=setscale(hs,1,arrscale(j));
    for i=1:nk
        hs=setk(hs,arrk(i));
        hs=addvar(hs,1,'u','v','p');
        hs=addeq(hs,dt(u)+U*dx(u)+v*dU==-dx(p)+1/Re*(dx2(u)+dy2(u)));
        hs=addeq(hs,dt(v)+U*dx(v)==-dy(p)+1/Re*(dx2(v)+dy2(v)));
        hs=addeq(hs,dx(u)+dy(v)==0);
        hs=compute(hs);
        eige=eigenvalue(hs,1e8);
        arrgrowthrate(i)=max(imag(eige));
    end
    plot(arrk,arrgrowthrate,arrcolor{j});
    hold on
end
```

For infinite intervals, hydrostab will automatically choose the Hermite spectral method. For Hermite spectral method, the argument in setscale() means the distance between the farthest collocation point and the origin, i.e. half the distance between the first and the last collocation point. (The collocation points always distribute symmetrically around the origin.)

As mentioned in Section III, the argument in setscale() can affect the accuracy of the result. In this example, for different wavenumbers, the appropriate value of the argument in setscale() is so different that we have to simultaneously plot results for different values of this argument. (The for-loop from Line 11 to Line 28 loops over four values of this argument.) The result is shown in Fig. 15, with the results of Betchov and Szewczyk[5] superimposed. It can be seen that a small value of this argument works well at high wavenumber while a large number of this argument works well at low wavenumber, similar to the previous example of Rayleigh-Taylor instability.

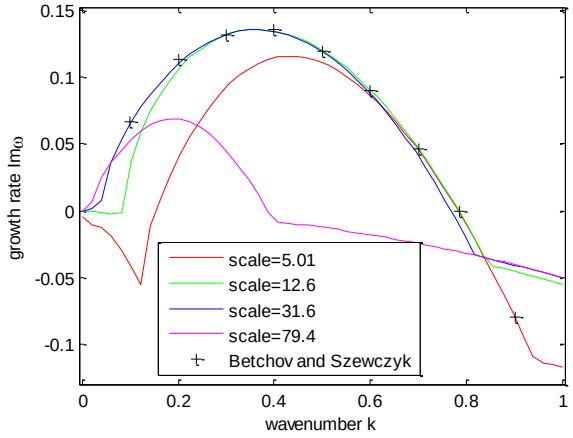


FIG. 15. Instability of a shear layer: growth rate vs wavenumber.

E. Stability of a viscoelastic liquid sheet moving in an inviscid gas

This example demonstrates how to use mixed partial derivatives. The problem is first studied by Liu *et al.*[6]. The physical model is almost the same as Fig. 9, except that the liquid is non-Newtonian and obey the Oldroyd-B viscoelastic model. The dimensionless linearized governing equations and boundary conditions are (Wang *et al.*[7])

$$\tau_{xx}+\lambda_1\left(\frac{\partial\tau_{xx}}{\partial t}+\frac{\partial\tau_{xx}}{\partial x}\right)=\frac{2}{Re}\left[\frac{\partial u}{\partial x}+\lambda_2\left(\frac{\partial^2 u}{\partial x\partial t}+\frac{\partial^2 u}{\partial x^2}\right)\right],\ \text{-1<y<1} \tag{37}$$

$$\tau_{xy}+\lambda_1\left(\frac{\partial\tau_{xy}}{\partial t}+\frac{\partial\tau_{xy}}{\partial x}\right)=\frac{1}{Re}\left[\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}+\lambda_2\left(\frac{\partial^2 u}{\partial y\partial t}+\frac{\partial^2 v}{\partial x\partial t}+\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 v}{\partial x^2}\right)\right],\ \text{-1<y<1} \tag{38}$$

$$\tau_{yy}+\lambda_1\left(\frac{\partial\tau_{yy}}{\partial t}+\frac{\partial\tau_{yy}}{\partial x}\right)=\frac{2}{Re}\left[\frac{\partial v}{\partial y}+\lambda_2\left(\frac{\partial^2 v}{\partial y\partial t}+\frac{\partial^2 v}{\partial x\partial y}\right)\right],\ \text{-1<y<1} \tag{39}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0,\quad \text{-1<y<1} \tag{40}$$

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=-\frac{\partial p}{\partial x}+\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y},\quad \text{-1<y<1} \tag{41}$$

$$\frac{\partial v}{\partial t}+\frac{\partial v}{\partial x}=-\frac{\partial p}{\partial y}+\frac{\partial \tau_{xy}}{\partial x}+\frac{\tau_{yy}}{\partial y},\quad \text{-1<y<1} \tag{42}$$

$$\frac{\partial^2 \varphi_{g1}}{\partial x^2}+\frac{\partial^2 \varphi_{g1}}{\partial y^2}=0,\quad \text{y>1} \tag{43}$$

$$\frac{\partial^2 \varphi_{g2}}{\partial x^2}+\frac{\partial^2 \varphi_{g2}}{\partial y^2}=0,\quad \text{y<-1} \tag{44}$$

$$v\big|_{y=1}=\frac{\partial \eta_1}{\partial t}+\frac{\partial \eta_1}{\partial x} \tag{45}$$

$$v\big|_{y=-1}=\frac{\partial \eta_2}{\partial t}+\frac{\partial \eta_2}{\partial x} \tag{46}$$

$$\left.\frac{\partial \varphi_{g1}}{\partial y}\right|_{y=1}=\frac{\partial \eta_1}{\partial t} \tag{47}$$

$$\left.\frac{\partial \varphi_{g2}}{\partial y}\right|_{y=-1}=\frac{\partial \eta_2}{\partial t} \tag{48}$$

$$-p\big|_{y=1}+\tau_{yy}\big|_{y=1}-\frac{1}{We}\frac{\partial^2 \eta_1}{\partial x^2}-\rho\left.\frac{\partial \varphi_{g1}}{\partial t}\right|_{y=1}=0 \tag{49}$$

$$-p\big|_{y=-1}+\tau_{yy}\big|_{y=-1}+\frac{1}{We}\frac{\partial^2 \eta_2}{\partial x^2}-\rho\left.\frac{\partial \varphi_{g2}}{\partial t}\right|_{y=-1}=0 \tag{50}$$

$$\tau_{xy}\big|_{y=1}=0 \tag{51}$$

$$\tau_{xy}\big|_{y=-1}=0 \tag{52}$$

where u, v, p are disturbance velocities and pressure in liquid phase, $\varphi_{g1}, \varphi_{g2}$ are disturbance potential function in gas phase, and η_1, η_2 are interface displacements. λ_1 and λ_2 are the dimensionless stress relaxation time and dimensionless deformation retardation time, respectively, which are parameters in the Oldroyd-B model. The MATLAB script the user needs to write is listed as follows.

Program 7: Stability of a viscoelastic liquid sheet moving in an inviscid gas

```
1  clear
```

```
Re=10;
We=100;
rho=0.01;
lambda1=100;
lambda2=50;
hs=hydrostab(3);
hs=set_interval(hs,1,-inf,-1,20);
hs=set_interval(hs,2,-1,1,20);
hs=set_interval(hs,3,1,inf,20);
hs=setscale(hs,1,100);
hs=setscale(hs,3,100);
nk=50;
arrk=linspace(0,1,nk);
for i=1:nk
    hs=setk(hs,arrk(i)); % wavenumber
    hs=addvar(hs,2,'u','v','p','txx','txy','tyy');
    hs=addvar(hs,3,'phig1');
    hs=addvar(hs,1,'phig2');
    hs=addvarxt(hs,'eta1','eta2');
    % liquid phase
    hs=addeq(hs,txx+lambda1*(dt(txx)+dx(txx))==2/Re*(dx(u)+lambda2*(dxdt(u)+dx2(u))));
    hs=addeq(hs,txy+lambda1*(dt(txy)+dx(txy))== ...
        1/Re*(dy(u)+dx(v)+lambda2*(dydt(u)+dxdt(v)+dxdy(u)+dx2(v))));
    hs=addeq(hs,tyy+lambda1*(dt(tyy)+dx(tyy))==2/Re*(dy(v)+lambda2*(dydt(v)+dxdy(v))));
    hs=addeq(hs,dt(u)+dx(u)==-dx(p)+dx(txx)+dy(txy),'ignorepossible');
    hs=addeq(hs,dt(v)+dx(v)==-dy(p)+dx(txy)+dy(tyy),'ignorepossible');
    hs=addeq(hs,dx(u)+dy(v)==0);
    % gas phase
    hs=addeq(hs,dx2(phig1)+dy2(phig1)==0,'ignorepossible');
    hs=addeq(hs,dx2(phig2)+dy2(phig2)==0,'ignorepossible');
    % boundary conditions
    hs=addeq(hs,at(v,1)==dt(eta1)+dx(eta1));
    hs=addeq(hs,at(v,-1)==dt(eta2)+dx(eta2));
    hs=addeq(hs,dyat(phig1,1)==dt(eta1));
    hs=addeq(hs,dyat(phig2,-1)==dt(eta2));
    hs=addeq(hs,-at(p,1)+at(tyy,1)-1/We*dx2(eta1)-rho*dtat(phig1,1)==0);
    hs=addeq(hs,-at(p,-1)+at(tyy,-1)+1/We*dx2(eta2)-rho*dtat(phig2,-1)==0);
    hs=addeq(hs,at(txy,1)==0);
    hs=addeq(hs,at(txy,-1)==0);
    % compute eigenvalues
    hs=compute(hs);
    eige=eigenvalue(hs,1e8);
    for j=1:length(eige)
        if real(retrieve(hs,eta1,eige(j))/retrieve(hs,eta2,eige(j)))>0 % sinuous
            plot(arrk(i),imag(eige(j)),'k+');
        else % varicose
            plot(arrk(i),imag(eige(j)),'m+');
        end
        hold on
    end
```

```
52    end
53    axis([0 1 0 0.05]);
```

The script is very similar to that in the example "Stability of a viscous sheet moving in an inviscid gas". The only thing to note is that the mixed partial derivatives should be written in certain format. For instance, the derivative $\partial^2(\ )/\partial x\partial t$ should be written as 'dxdt' but not 'dtdx'. The user can browse the @equ folder to obtain the formats.

The result of Program 7 is shown in Fig. 16. The analytical result taken from Wang *et al.*[7] is also superimposed. It can be seen that they agrees well. The analytical result for Newtonian sheet is also superimposed, which is obtained by setting both λ_1 and λ_2 to zero. It can be seen that the growth rate of a viscoelastic sheet is slightly greater than a Newtonian sheet.

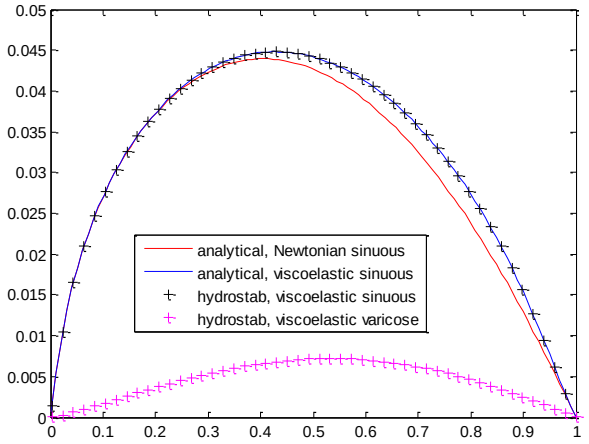


FIG. 16. Stability of a viscoelastic liquid sheet moving in an inviscid gas.

F. Spatial stability of a viscous sheet moving in an inviscid gas

This example demonstrates how to perform a spatial stability analysis. The physical model, governing equations and boundary conditions are the same as those in Example B. The difference is that Example B is a temporal stability analysis while this example is a spatial stability analysis. This difference is reflected on the known quantity and unknown quantity: in Example B, the wavenumber is known and the frequency is unknown; in this example, the frequency is known and the wavenumber is unknown. The frequency ω is a real number and the wavenumber k is a complex number. The opposite number of the imaginary part of k represents the spatial growth rate.

The MATLAB script the user needs to write is listed as follows. The MATLAB script should be placed in the folder 'spatial' because this is a spatial stability analysis. Note that if you have run the previous examples before, you may need to restart MATLAB to switch the classes from those in the folder 'temporal' to those in the folder 'spatial'.

Program 8: Spatial stability of a viscous sheet moving in an inviscid gas

```
clear
Re=1000;
We=400;
rho=0.001;
hs=hydrostab(3);
hs=set_interval(hs,1,-inf,-1,20);
hs=set_interval(hs,2,-1,1,20);
hs=set_interval(hs,3,1,inf,20);
hs=setscale(hs,1,100);
hs=setscale(hs,3,100);
nomega=50;
arromega=linspace(0,0.5,nomega);
for i=1:nomega
    hs=setomega(hs,arromega(i)); % frequency
    hs=addvar(hs,2,'u','v','p');
    hs=addvar(hs,3,'phig1');
    hs=addvar(hs,1,'phig2');
    hs=addvarxt(hs,'eta1','eta2');
    % liquid phase
    hs=addeq(hs,dt(u)+dx(u)==-dx(p)+1/Re*(dx2(u)+dy2(u)),'ignorepossible');
    hs=addeq(hs,dt(v)+dx(v)==-dy(p)+1/Re*(dx2(v)+dy2(v)),'ignorepossible');
    hs=addeq(hs,dx(u)+dy(v)==0);
    % gas phase
    hs=addeq(hs,dx2(phig1)+dy2(phig1)==0,'ignorepossible');
    hs=addeq(hs,dx2(phig2)+dy2(phig2)==0,'ignorepossible');
    % boundary conditions
    hs=addeq(hs,at(v,1)==dt(eta1)+dx(eta1));
    hs=addeq(hs,at(v,-1)==dt(eta2)+dx(eta2));
    hs=addeq(hs,dyat(phig1,1)==dt(eta1));
    hs=addeq(hs,dyat(phig2,-1)==dt(eta2));
    hs=addeq(hs,-at(p,1)+2/Re*dyat(v,1)-1/We*dx2(eta1)-rho*dtat(phig1,1)==0);
    hs=addeq(hs,-at(p,-1)+2/Re*dyat(v,-1)+1/We*dx2(eta2)-rho*dtat(phig2,-1)==0);
    hs=addeq(hs,dyat(u,1)+dxat(v,1)==0);
    hs=addeq(hs,dyat(u,-1)+dxat(v,-1)==0);
    % compute eigenvalues
    hs=compute(hs);
    eige=eigenvalue(hs,1e8);
    for j=1:length(eige)
        if real(retrieve(hs,eta1,eige(j))/retrieve(hs,eta2,eige(j)))>0 % sinuous
            plot(arromega(i),-imag(eige(j)),'r+');
        else % varicose
            plot(arromega(i),-imag(eige(j)),'b+');
        end
        hold on
    end
end
axis([0 0.5 0 0.011]);
```

This program is almost the same as Program 4, except for Lines 11, 12, 13, 14, 40 and 42. The result is shown in Fig. 17, with the analytical result superimposed. It can be seen that the numerical result agrees well with the analytical result.

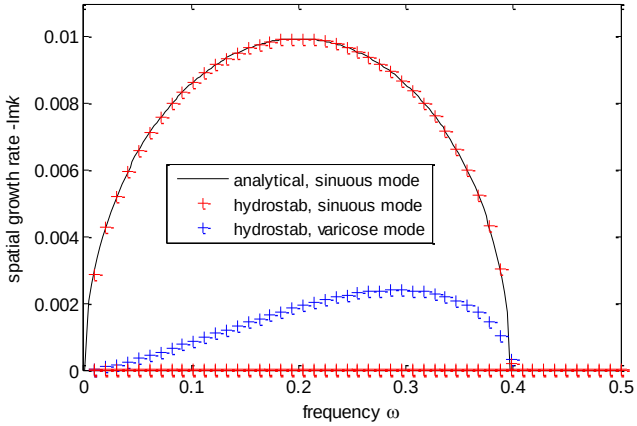


FIG. 17. Results of Program 8. The analytical result is also superimposed.

REFERENCES

[1] S. A. Orszag, "Accurate solution of the Orr-Sommerfeld stability equation," J. Fluid Mech. **50**, 689-703 (1971).

[2] F. Charru, *Hydrodynamic instabilities* (Cambridge University Press, 2011), 53-61 and 68-74. (Translated by Patricia de Forcrand-Millard)

[3] S. P. Lin, "Stability of a viscous liquid curtain," J. Fluid Mech. **104**, 111-118 (1981).

[4] L. Yang, C. Wang, Q. Fu, M. Du and M. Tong, "Weakly nonlinear instability of planar viscous sheets," J. Fluid Mech. **735**, 249-287 (2013)

[5] R. Betchov and A. Szewczyk, "Stability of a shear layer between parallel streams," Physics of Fluids **6**, 1391 (1963)

[6] Z. Liu, G. Brenn and F. Durst, "Linear analysis of the instability of two-dimensional non-Newtonian liquid sheets," J. Non-Newtonian Fluid Mech. **78**, 133-166 (1998)

[7] C. Wang, L. Yang, L. Xie and P. Chen, “Weakly nonlinear instability of planar viscoelastic sheets,” Physics of Fluids **27**, 013103 (2015)