

Author: Sadaf Arabi

Title: Stability analysis of a Rijke tube by a Green's function approach

Abstract: The objective of this report is to employ a Green's function methodology for conducting stability analysis across distinct control parameters, such as the heat source position and the tube length. The Green's function approach effectively captures the acoustic resonator's behavior. By introducing an interface with a hypothetical point source, this method facilitates the prediction of resonator frequencies and amplitudes, contingent on the occurrence of a temperature jump within the tube. The variation in sound speed within the tube at the interface leads to discontinuities in the derivative of the Green's function amplitude with respect to x . To visualize the impact of bifurcation parameters, namely heat source position and tube length, stability maps are generated. These maps delineate a spectrum of operational states. Unstable states are depicted by black regions, while white regions indicate stable states.

Keywords: Thermoacoustic instabilities, Green's function approach

"This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License."

CC BY-NC-SA 4.0 (link to the license): <https://creativecommons.org/licenses/by-nc-sa/4.0/>

To model the acoustic resonator by a Green's function approach, we consider non-homogenous wave equation, as follow

$$\frac{1}{c^2} \frac{\partial^2 G}{\partial t^2} - \frac{\partial^2 G}{\partial x^2} = \delta(x - x') \delta(t - t')$$

Green's function is defined as a superposition of modes with frequency ω_n and amplitude g_n ,

$$G(x, x', t, t') = H(t - t') \operatorname{Re} \sum_{n=1}^N g_n(x, x') e^{-i\omega_n(t-t')}$$

Due to the pressure field within the tube and considering the upstream and downstream reflection coefficients and temperature jump within the tube, the characteristic equation becomes,

$$F(\omega) = e^{-ik_1 x_q} e^{-ik_2(L-x_q)} - R_L R_{21} e^{-ik_1 x_q} e^{ik_2(L-x_q)} - R_0 R_{12} e^{ik_1 x_q} e^{-ik_2(L-x_q)} - R_0 R_L e^{ik_1 x_q} e^{ik_2(L-x_q)}$$

Here is the code to find the frequencies.

```
clc; clear all;

% parameters

c1 = 350;           % speed of sound in cold region
c2 = 430;           % speed of sound in hot region
ro1 = 1.1493;       % density in cold region
ro2 = 0.7595;       % density in hot region
L=2;                % Length of the duct

R0 = -1;            % reflection coefficient at inlet (x=0)
```

```

R1 = -1;                % reflection coefficient at outlet (x=L)

% reflection and transmission coefficients at interface at x=xq
R12 = (ro2*c2-ro1*c1)/(ro2*c2+ro1*c1);
R21 = -R12;
T12 = (2*ro2*c2)/(ro2*c2+ro1*c1);
T21 = ((ro1*c1)/(ro2*c2))*T12;

n = 1;                % Number of mode
xq = 0.5;            % Heat source position

% Newton-Raphson Method
initial = (n*pi*c2)/(L); % initial guess for iteration
syms w
k1 = w/c1; k2 = w/c2; % Wave numbers

% Characrtistic Equation
fun = exp(-1i*k1*xq)*exp(-1i*k2*(L-xq))-...
      R1*R21*exp(-1i*k1*xq)*exp(1i*k2*(L-xq))-...
      R0*R12*exp(1i*k1*xq)*exp(-1i*k2*(L-xq))-...
      R0*R1*exp(1i*k1*xq)*exp(1i*k2*(L-xq));

w1 = vpasolve(fun == 0,w,initial);

w1 = double(w1); % Output

```

To find the amplitude of Green's function, after many mathematical procedures, the following equations are obtained.

$$\hat{g}(x, x', \omega) = \begin{cases} A(x, \omega) B(x', \omega) & 0 < x < x_q \\ B(x', \omega) C(x, \omega) & x_q < x < x' \\ C(x', \omega) B(x, \omega) & x' < x < L \end{cases}$$

where

$$A(x, \omega) = T_{21} (R_0 e^{ik_1 x} + e^{-ik_1 x})$$

$$B(x, \omega) = R_L e^{ik_2(L-x)} + e^{-ik_2(L-x)}$$

$$C(x, \omega) = e^{ik_2(x-x_q)} (R_{21} e^{-ik_1 x} + R_0 e^{ik_1 x_q}) + e^{-ik_2(x-x_q)} (e^{-ik_1 x_q} - R_0 R_{12} e^{ik_1 x_q})$$

The result is,

$$g_n(x, x') = \frac{c_2 \hat{g}(x, x', \omega_n)}{\omega_n F'(\omega_n)}$$

Following code shows that g_n is continuous at the point source. However, the gradient $\frac{\partial g_n}{\partial x}$ is not.

```

fun_p = diff(fun);
F = subs(fun_p,w1);
FUN_p = double(F);

```

```

k1 = w1/c1; k2=w1/c2;
xp = xq;

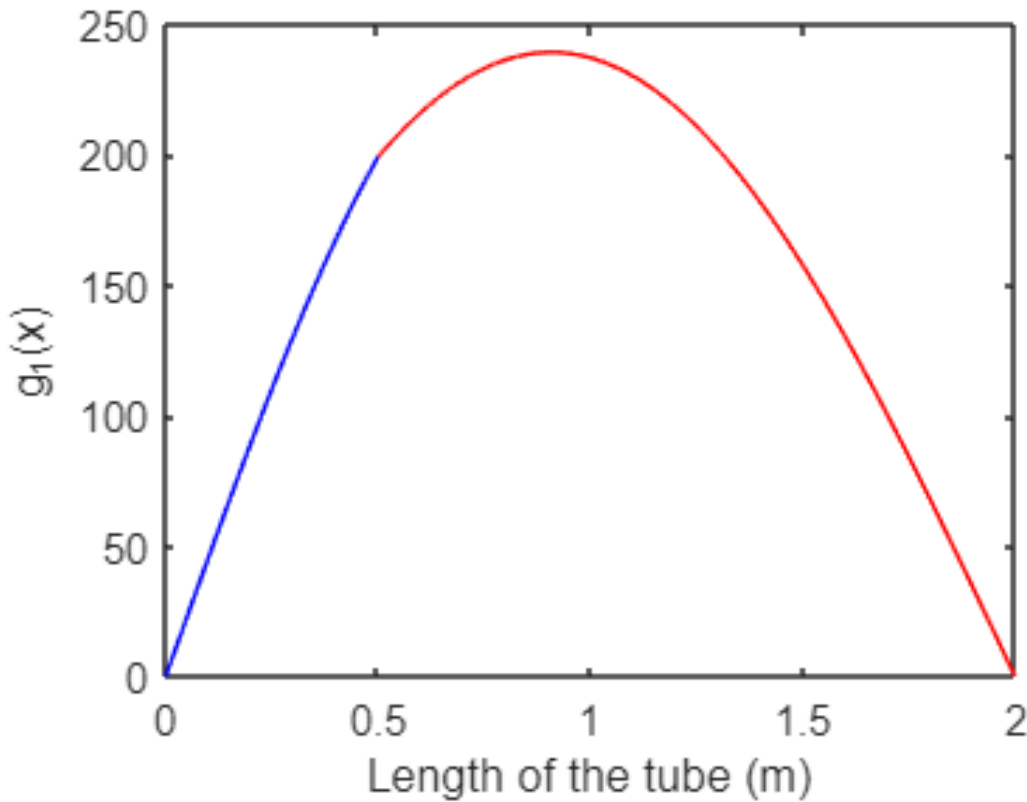
x1=0:0.001:xq;
for i=1:length(x1)
    A(i) = T21*(R0*exp(1i*k1*x1(i)) + exp(-1i*k1*x1(i)));
    Bp = R1*exp(1i*k2*(L-xp)) + exp(-1i*k2*(L-xp));
    ghat_1(i) = A(i) * Bp;
    g_1(i) = (c2*ghat_1(i))/(w1*FUN_p);
end

x2 = xq:0.001:L;
for i=1:length(x2)
    Bp = R1*exp(1i*k2*(L-xp)) + exp(-1i*k2*(L-xp));
    C(i) = exp(1i*k2*(x2(i)-xq))*(R21*exp(-1i*k1*xq) + R0*exp(1i*k1*xq)) + ...
        exp(-1i*k2*(x2(i)-xq))*(exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));
    ghat_2(i) = Bp * C(i);
    g_2(i) = (c2*ghat_2(i))/(w1*FUN_p);
end

x3 = xq:0.001:L;
for i=1:length(x3)
    Cp = exp(1i*k2*(xp-xq))*(R21*exp(-1i*k1*xq) + R0*exp(1i*k1*xq)) + ...
        exp(-1i*k2*(xp-xq))*(exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));
    B(i) = R1*exp(1i*k2*(L-x3(i))) + exp(-1i*k2*(L-x3(i)));
    ghat_3(i) = Cp * B(i);
    g_3(i) = (c2*ghat_3(i))/(w1*FUN_p);
end

figure;
plot(x1,abs(g_1),'b')
hold on
plot(x2,abs(g_2),'-r')
xlabel('Length of the tube, L (m)')
ylabel('g_1(x)')

```



```

x1=0:0.001:xq;
for i=1:length(x1)
    A_dx(i) = T21*(1i*k1)*(R0*exp(1i*k1*x1(i)) - exp(-1i*k1*x1(i)));
    Bp = R1*exp(1i*k2*(L-xp)) + exp(-1i*k2*(L-xp));
    ghat_1_dx(i) = A_dx(i) * Bp;
    g_1_dx(i) = (c2*ghat_1_dx(i))/(w1*FUN_p);
end

x2 = xq:0.001:L;
for i=1:length(x2)
    Bp = R1*exp(1i*k2*(L-xp)) + exp(-1i*k2*(L-xp));
    C_dx(i) = (1i*k2)*exp(1i*k2*(x2(i)-xq))*(R21*exp(-1i*k1*xq) + ...
        R0*exp(1i*k1*xq)) - (1i*k2)*(exp(-1i*k2*(x2(i)-xq)))*...
        (exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));
    ghat_2_dx(i) = Bp * C_dx(i);
    g_2_dx(i) = (c2*ghat_2_dx(i))/(w1*FUN_p);
end

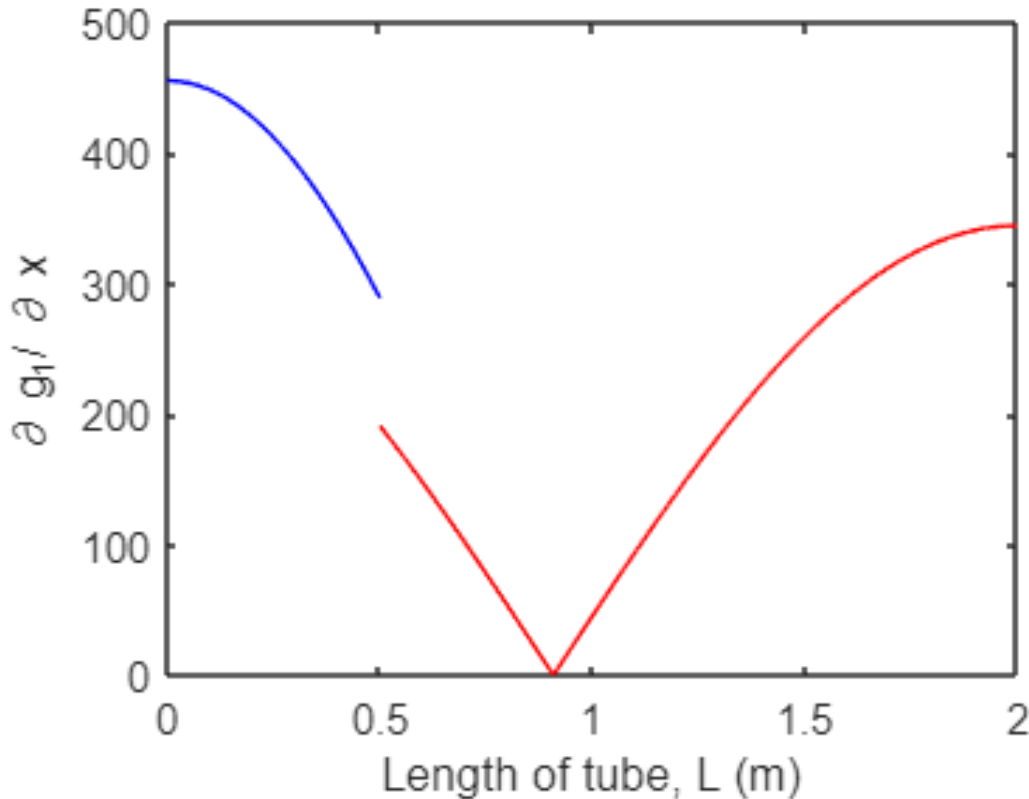
x3 = xq:0.001:L;
for i=1:length(x3)
    Cp = exp(1i*k2*(xp-xq))*(R21*exp(-1i*k1*xq) + R0*exp(1i*k1*xq)) + ...
        exp(-1i*k2*(xp-xq))*(exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));
    B_dx(i) = (1i*k2)*(-R1*exp(1i*k2*(L-x3(i))) + exp(-1i*k2*(L-x3(i))));
    ghat_3_dx(i) = Cp * B_dx(i);
    g_3_dx(i) = (c2*ghat_3_dx(i))/(w1*FUN_p);
end

```

```

figure;
plot(x1,abs(g_1_dx),'b')
hold on
plot(x2,abs(g_2_dx),'-r')
xlabel('Length of tube, L (m)')
ylabel('\partial g_1/ \partial x')

```



By using a Green's function approach, stability map is produced for different control parameters. Here, we start with heat source position as bifurcation parameter.

```

clc; clear all;

% parameters

c1 = 350;           % speed of sound in cold region
c2 = 430;           % speed of sound in hot region
ro1 = 1.1493;       % density in cold region
ro2 = 0.7595;       % density in hot region
L=2;                % Length of the duct

R0 = -1;            % reflection coefficient at inlet (x=0)
R1 = -1;            % reflection coefficient at outlet (x=L)

% reflection and transmission coefficients at interface at x=xq
R12 = (ro2*c2-ro1*c1)/(ro2*c2+ro1*c1);

```

```

R21 = -R12;
T12 = (2*ro2*c2)/(ro2*c2+ro1*c1);
T21 = ((ro1*c1)/(ro2*c2))*T12;

tau0 = 5*10^-3;           % Time-delay coefficient 1
tau2 = 4.4*10^-3;       % Time-delay coefficient 2
g0 = 1.4;                % Heat release coefficient 1
g1 = 0.3;                % Heat release coefficient 1
K = 3*10^5;              % Heater power
gamma = 1.4;             % Specific heat ratio
Be = -(gamma-1)/(c2*c2); % Heat release constant

n = 1;                   % Number of mode
xq = 0:0.001:2;         % Heat source position

% Newton-Raphson Method
initial = (n*pi*c2)/(L); % initial guess for iteration

for i=1:length(xq)
syms w
k1 = w/c1; k2 = w/c2;   % Wave numbers

% Characteristic Equation
fun = exp(-1i*k1*xq(i))*exp(-1i*k2*(L-xq(i)))-...
      R1*R21*exp(-1i*k1*xq(i))*exp(1i*k2*(L-xq(i)))-...
      R0*R12*exp(1i*k1*xq(i))*exp(-1i*k2*(L-xq(i)))-...
      R0*R1*exp(1i*k1*xq(i))*exp(1i*k2*(L-xq(i)));

wn = vpasolve(fun == 0,w,initial);

w1(i) = double(wn);     % Output
initial=w1(i);

fun_p = diff(fun);
F = subs(fun_p,wn);
FUN_p = double(F);

k1 = w1(i)/c1; k2=w1(i)/c2;
xp(i)=xq(i);

A(i) = T21*(R0*exp(1i*k1*xq(i)) + exp(-1i*k1*xq(i)));
B(i) = R1*exp(1i*k2*(L-xq(i))) + exp(-1i*k2*(L-xq(i)));
C(i) = exp(1i*k2*(xq(i)-xq(i)))*(R21*exp(-1i*k1*xq(i)) + R0*exp(1i*k1*xq(i)))+ ...
      exp(-1i*k2*(xq(i)-xq(i)))*(exp(-1i*k1*xq(i)) - R0*R12*exp(1i*k1*xq(i)));

Ap(i) = T21*(R0*exp(1i*k1*xp(i)) + exp(-1i*k1*xp(i)));
Bp(i) = R1*exp(1i*k2*(L-xp(i))) + exp(-1i*k2*(L-xp(i)));
Cp(i) = exp(1i*k2*(xp(i)-xq(i)))*(R21*exp(-1i*k1*xq(i)) + R0*exp(1i*k1*xq(i)))+ ...
      exp(-1i*k2*(xp(i)-xq(i)))*(exp(-1i*k1*xq(i)) - R0*R12*exp(1i*k1*xq(i)));

A_dx(i) = T21*(1i*k1)*(R0*exp(1i*k1*xq(i)) - exp(-1i*k1*xq(i)));
B_dx(i) = (1i*k2)*(-R1*exp(1i*k2*(L-xq(i))) + exp(-1i*k2*(L-xq(i))));
C_dx(i) = (1i*k2)*exp(1i*k2*(xq(i)-xq(i)))*(R21*exp(-1i*k1*xq(i)) +...
      R0*exp(1i*k1*xq(i))) - (1i*k2)*(exp(-1i*k2*(xq(i)-xq(i))))*...

```

```

    (exp(-1i*k1*xq(i)) - R0*R12*exp(1i*k1*xq(i)));

ghat_1_dx(i) = A_dx(i) * Bp(i);
ghat_2_dx(i) = Bp(i) * C_dx(i);
ghat_3_dx(i) = Cp(i) * B_dx(i);

G1(i) = (c2*ghat_1_dx(i))/(w1(i)*FUN_p);
G2(i) = (c2*ghat_2_dx(i))/(w1(i)*FUN_p);
G3(i) = (c2*ghat_3_dx(i))/(w1(i)*FUN_p);

end

A = linspace(0,2,length(xq));
% A=0.01;
for j = 1:length(A)
    tau(j) = tau0 + tau2*(A(j)^2);
    gmax(j) = g0 - g1*A(j);
    n0(j) = 0.5 * (gmax(j)-1);
    n1(j) = 0.5 * (gmax(j)+1);
end

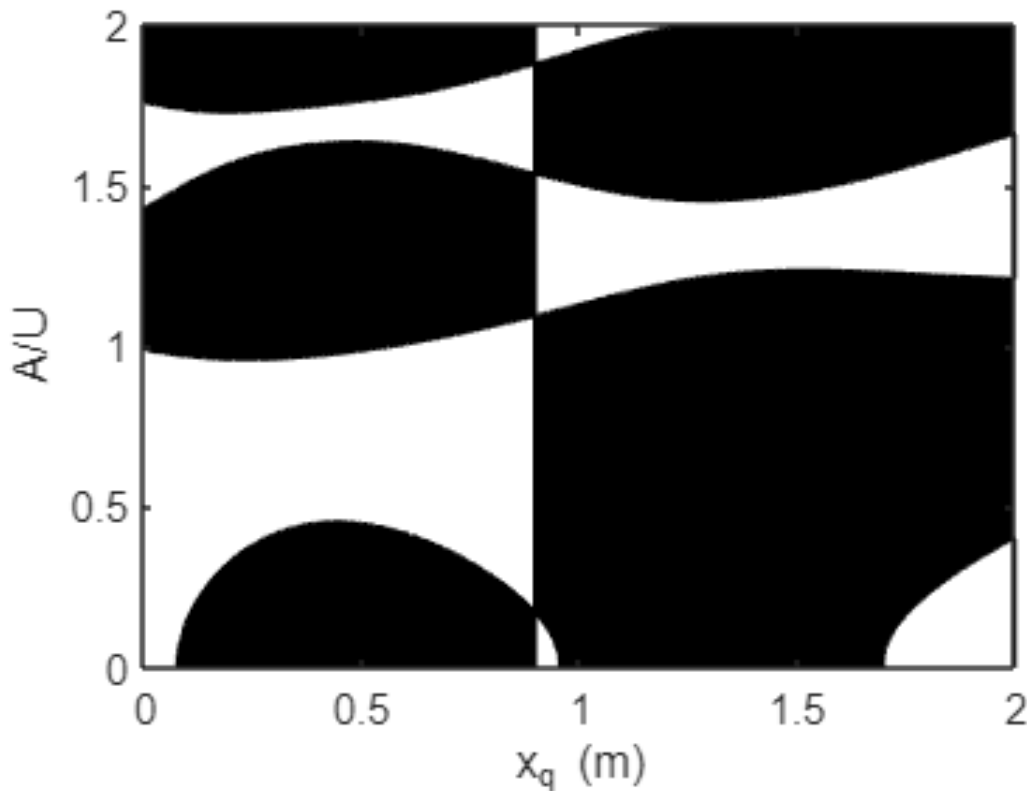
for i = 1:length(xq)
    for j = 1:length(A)
        b0 = -K*n1(j)*Be*imag(w1(i)*conj(G2(i)));
        b1 = K*n1(j)*Be*real(G2(i));
        c0 = abs(w1(i))^2-K*n0(j)*Be*imag(w1(i)*conj(G2(i)));
        c11 = -2*imag(w1(i))+K*n0(j)*Be*real(G2(i));

        a1lin = c11+b0*((sin(w1(i)*tau(j)))/(w1(i)))-b1*cos(w1(i)*tau(j));
        a0lin = c0-b0*cos(w1(i)*tau(j))-b1*w1(i)*sin(w1(i)*tau(j));
        omlin = sqrt(a0lin-0.5*(a1lin*a1lin));
        a1non = c11+b0*((sin(omlin*tau(j)))/(omlin))-b1*cos(omlin*tau(j));
        a0non = c0-b0*cos(omlin*tau(j))-b1*omlin*sin(omlin*tau(j));
        growth = -0.5*(c11+b0*((sin(sqrt(a0non)*tau(j)))/(sqrt(a0non)))-...
            b1*cos(sqrt(a0non)*tau(j)));

        if growth >= 0
            stable1(i,j) = -1;
        else
            stable1(i,j) = 1;
        end
    end
end

end
figure;
contourf(xq,A,stable1')
colormap gray
xlabel('x_q (m)')
ylabel('A/U')

```



Varying the length of the tube, leads to following stability map:

```

clc; clear all;

% parameters

c1 = 350;           % speed of sound in cold region
c2 = 430;           % speed of sound in hot region
ro1 = 1.1493;       % density in cold region
ro2 = 0.7595;       % density in hot region
L=0.4:0.01:2;      % Length of the duct

R0 = -1;           % reflection coefficient at inlet (x=0)
R1 = -1;           % reflection coefficient at outlet (x=L)

% reflection and transmission coefficients at interface at x=xq
R12 = (ro2*c2-ro1*c1)/(ro2*c2+ro1*c1);
R21 = -R12;
T12 = (2*ro2*c2)/(ro2*c2+ro1*c1);
T21 = ((ro1*c1)/(ro2*c2))*T12;

tau0 = 5*10^-3;    % Time-delay coefficient 1
tau2 = 4.4*10^-3;  % Time-delay coefficient 2
g0 = 1.4;          % Heat release coefficient 1
g1 = 0.3;          % Heat release coefficient 1

```



```

K = 3*10^5; % Heater power
gamma = 1.4; % Specific heat ratio
Be = -(gamma-1)/(c2*c2); % Heat release constant

n = 1; % Number of mode
xq = 0.1; % Heat source position

% Newton-Raphson Method
initial = (n*pi*c2)/(L(1)); % initial guess for iteration

for i=1:length(L)
syms w
k1 = w/c1; k2 = w/c2; % Wave numbers

% Characteristic Equation
fun = exp(-1i*k1*xq)*exp(-1i*k2*(L(i)-xq))-...
      R1*R21*exp(-1i*k1*xq)*exp(1i*k2*(L(i)-xq))-...
      R0*R12*exp(1i*k1*xq)*exp(-1i*k2*(L(i)-xq))-...
      R0*R1*exp(1i*k1*xq)*exp(1i*k2*(L(i)-xq));

wn = vpasolve(fun == 0,w,initial);

w1(i) = double(wn); % Output
initial=w1(i);

fun_p = diff(fun);
F = subs(fun_p,wn);
FUN_p = double(F);

k1 = w1(i)/c1; k2=w1(i)/c2;
xp = xq;

A(i) = T21*(R0*exp(1i*k1*xq) + exp(-1i*k1*xq));
B(i) = R1*exp(1i*k2*(L(i)-xq)) + exp(-1i*k2*(L(i)-xq));
C(i) = exp(1i*k2*(xq-xq))*(R21*exp(-1i*k1*xq) + R0*exp(1i*k1*xq)) + ...
      exp(-1i*k2*(xq-xq))*(exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));

Ap(i) = T21*(R0*exp(1i*k1*xp) + exp(-1i*k1*xp));
Bp(i) = R1*exp(1i*k2*(L(i)-xp)) + exp(-1i*k2*(L(i)-xp));
Cp(i) = exp(1i*k2*(xp-xq))*(R21*exp(-1i*k1*xq) + R0*exp(1i*k1*xq)) + ...
      exp(-1i*k2*(xp-xq))*(exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));

A_dx(i) = T21*(1i*k1)*(R0*exp(1i*k1*xq) - exp(-1i*k1*xq));
B_dx(i) = (1i*k2)*(-R1*exp(1i*k2*(L(i)-xq)) + exp(-1i*k2*(L(i)-xq)));
C_dx(i) = (1i*k2)*exp(1i*k2*(xq-xq))*(R21*exp(-1i*k1*xq) + R0*exp(1i*k1*xq)) - ...
      (1i*k2)*(exp(-1i*k2*(xq-xq)))*(exp(-1i*k1*xq) - R0*R12*exp(1i*k1*xq));

ghat_1_dx(i) = A_dx(i) * Bp(i);
ghat_2_dx(i) = Bp(i) * C_dx(i);
ghat_3_dx(i) = Cp(i) * B_dx(i);

G1(i) = (c2*ghat_1_dx(i))/(w1(i)*FUN_p);
G2(i) = (c2*ghat_2_dx(i))/(w1(i)*FUN_p);
G3(i) = (c2*ghat_3_dx(i))/(w1(i)*FUN_p);

```

```

end

A = linspace(0,2,length(L));
% A=0.01;
for j = 1:length(A)
    tau(j) = tau0 + tau2*(A(j)^2);
    gmax(j) = g0 - g1*A(j);
    n0(j) = 0.5 * (gmax(j)-1);
    n1(j) = 0.5 * (gmax(j)+1);
end

for i = 1:length(L)
    for j = 1:length(A)
        b0 = -K*n1(j)*Be*imag(w1(i)*conj(G2(i)));
        b1 = K*n1(j)*Be*real(G2(i));
        c0 = abs(w1(i))^2-K*n0(j)*Be*imag(w1(i)*conj(G2(i)));
        c11 = -2*imag(w1(i))+K*n0(j)*Be*real(G2(i));

        a1lin = c11+b0*((sin(w1(i)*tau(j)))/(w1(i)))-b1*cos(w1(i)*tau(j));
        a0lin = c0-b0*cos(w1(i)*tau(j))-b1*w1(i)*sin(w1(i)*tau(j));
        omlin = sqrt(a0lin-0.5*(a1lin*a1lin));
        a1non = c11+b0*((sin(omlin*tau(j)))/(omlin))-b1*cos(omlin*tau(j));
        a0non = c0-b0*cos(omlin*tau(j))-b1*omlin*sin(omlin*tau(j));
        growth = -0.5*(c11+b0*((sin(sqrt(a0non)*tau(j)))/(sqrt(a0non)))-...
            b1*cos(sqrt(a0non)*tau(j)));

        if growth >= 0
            stable1(i,j) = -1;
        else
            stable1(i,j) = 1;
        end
    end
end

end
figure;
contourf(L,A,stable1')
colormap gray
xlabel('L (m)')
ylabel('A/U')

```

