% Mathlab- code for the Tenti-S6 model. With this code, a spontaneous RB-scattering

%spectrum can be generated. Here, Air is taken as an example. If choosing other molecules,

% the corresponding parameters should be changed. And an Unknown parameters can be set

%as a variables for fitting.

clc; clear all; close all;

%--------------Air as an example-----------------------------------

gas = 'Air';% selecting gas style

M = 28.964; % molar mass, unit: g/mol

p = 1.000 \* 10^5; % pressure, unit pa

T = 20 + 273.15; %temperature, unit K

eta\_0 = 1.716e-5;% unit Pa·s

eta\_s = eta\_0\*(T/273)^1.5\*(273+194)/(T+194);% shear viscosity, unit Pa·s

bulk\_vis = 1.48 \* 10^(-5);% bulk viscosity, unit Pa∙s

thermal\_cond\_0 = 0.0241;% unit W/(m\*K)

thermal\_cond = thermal\_cond\_0\*(T/273)^1.5\*(273+111)/(T+111); % thermal conductivity, unit W/(m\*K)

lambda = 532.22e-9; % The wavelength of the laser, unit m

kb = 1.380649e-23; % Boltzmann constant, unit J/K

R = 8.314462618;% gas constant, unit J/(mol·K)

m\_m = M\*1.66053906660e-27; % Molecular weight, kg

v0 = sqrt(2\*kb\*T/m\_m); % thermal speed，m/sec

n0 = p/(T\*kb); % Numerical density, in 1/m^3

%-------------------scattering angle---------------------------------

scattering\_angle = 55.7; % scattering angle

angle = scattering\_angle/2\*(pi/180); % scattering angle in radians

q = sin(angle)\*4\*pi/lambda; % wave number, in m^-1

y = n0\*kb\*T/(q\*v0\*eta\_s); % Uniformity parameter dimensionless

%--------------------------------------------------------------------

n\_xi = 2001; % Number of frequency points, dimensionless unit

xi\_lef = -2.6; % Minimum frequency point, dimensionless unit

xi\_rgt = 2.6; % Maximum frequency point, dimensionless unit

dxi = (xi\_rgt-xi\_lef)./(2.0.\*(n\_xi-floor(n\_xi./2)-1));

for i = 1:n\_xi

 xi(i) = xi\_lef+dxi.\*(i-1); % Calculate the frequency points,

% x-scale dimensionless unit

end

c\_tr = 3/2; % Translational heat capacity

c\_int = 1; % internal heat capacity, linear molecule c\_int = 1, nonlinear molecule c\_int = 1.5

gamma\_int = c\_int/(c\_tr+c\_int); % Internal heat capacity ratio,

y6 = y;

rlx\_int = 1.5\*bulk\_vis/(eta\_s\*gamma\_int); % Internal relaxation

eukenf = m\_m\*thermal\_cond/(eta\_s\*kb\*(c\_tr+c\_int)); % Euken factor

v1 = xi'.\*q.\*v0./(2.\*pi)./1e9;% The relative frequency, in GHz

%-----------------------------------------------------------------

%% The spectrum calculation from WCU equation

cpxunit = 1i; % Unit imaginary part of complex number

one = 1; % One

zero = 0; % Zero

% Calculate j

j020 = -y;

j100 = -gamma\_int\*y/rlx\_int;

j001 = j100\*c\_tr/c\_int;

j100001 = j100\*sqrt(c\_tr/c\_int);

j110 = j100\*5/6+j020\*2/3;

j011110 = j100\*sqrt(5/(8\*c\_int));

j\_nu = 0.4\*(1.5+c\_int)+(3+c\_int)/(2\*rlx\_int)+9\*eukenf/(16\*rlx\_int^2);

j\_de = -1+(4/15)\*eukenf\*(1.5+c\_int)+(c\_int/3)\*eukenf/rlx\_int;

j\_co = -y\*(2\*gamma\_int/3);

j011 = j\_co\*j\_nu/j\_de;

% Initialization for the area of spontaneous RBS spectrum

% Coherent RBS spectrum

sptarea = 0; % Area of spontaneous RBS spectrum

coharea = 0; % Area of coherent RBS spectrum

% calculate the matrix equation Ax = B. The x here is the target variable.

for i = 1:n\_xi

 % Calculate w

 z = xi(i)+cpxunit.\*y6;

 F = @(x)exp(-x.^2)./(z-x);

 w0 = integral(F,-inf,inf);

 w1 = -sqrt(pi)+z.\*w0;

 w2 = z.\*w1;

 w3 = -0.5.\*sqrt(pi)+z.\*w2;

 w4 = z.\*w3;

 w5 = -3.\*sqrt(pi)./4+z.\*w4;

 w6 = z.\*w5;

 % Calculate i

 i0000 = w0./(sqrt(pi));

 i0100 = (z.\*w0-sqrt(pi)).\*sqrt(2./pi);

 i0001 = i0100;

 i0010 = (2.\*w2-w0)./(sqrt(6.\*pi));

 i1000 = i0010;

 i0011 = (2.\*w3-3.\*w1)./(sqrt(5.\*pi));

 i1100 = i0011;

 i0101 = 2.\*w2./sqrt(pi);

 i0110 = (-w1+2\*w3)./sqrt(3.\*pi);

 i1001 = i0110;

 i0111 = (-3.\*w2+2.\*w4).\*sqrt(2./(5.\*pi));

 i1101 = i0111;

 i1111 = (13.\*w2-12.\*w4+4.\*w6)./(5.\*sqrt(pi));

 i0002 = (-w0+2.\*w2)./sqrt(3.\*pi);

 i0200 = i0002;

 i0211 = (-w1+8.\*w3-4.\*w5)/sqrt(15.\*pi);

 i1102 = i0211;

 i0202 = 2.\*(w0-2.\*w2+2.\*w4)./(3.\*sqrt(pi));

 i0210 = (w0+4.\*w2-4.\*w4)./(3.\*sqrt(2.\*pi));

 i1002 = i0210;

 i0102 = (-w1+2.\*w3).\*sqrt(2./(3.\*pi));

 i0201 = i0102;

 i1010 = (5.\*w0-4.\*w2+4.\*w4)./(6.\*sqrt(pi));

 i1110 = (7.\*w1-8.\*w3+4.\*w5)./sqrt(30.\*pi);

 i1011 = i1110;

 % Calculate matrix A

 a\_factor = one;

 A(1,1) = -j020.\*i0000+cpxunit;

 A(2,1) = -j020.\*i0001;

 A(3,1) = -j020.\*i0011;

 A(4,1) = -j020.\*i0010;

 A(5,1) = zero;

 A(6,1) = zero;

 A(1,2) = -j020.\*i0100;

 A(2,2) = -j020.\*i0101+cpxunit;

 A(3,2) = -j020.\*i0111;

 A(4,2) = -j020.\*i0110;

 A(5,2) = zero;

 A(6,2) = zero;

 A(1,3) = (j020-j110).\*i1100;

 A(2,3) = (j020-j110).\*i1101;

 A(3,3) = (j020-j110).\*i1111-cpxunit;

 A(4,3) = (j020-j110).\*i1110;

 A(5,3) = -j011110.\*i0100;

 A(6,3) = -j011110.\*i0101;

 A(1,4) = (j020-j100).\*i1000;

 A(2,4) = (j020-j100).\*i1001;

 A(3,4) = (j020-j100).\*i1011;

 A(4,4) = (j020-j100).\*i1010-cpxunit;

 A(5,4) = -j100001.\*i0000;

 A(6,4) = -j100001.\*i0001;

 A(1,5) = j100001.\*i1000;

 A(2,5) = j100001.\*i1001;

 A(3,5) = j100001.\*i1011;

 A(4,5) = j100001.\*i1010;

 A(5,5) = (j001-j020).\*i0000+cpxunit;

 A(6,5) = (j001-j020).\*i0001;

 A(1,6) = j011110.\*i1100;

 A(2,6) = j011110.\*i1101;

 A(3,6) = j011110.\*i1111;

 A(4,6) = j011110.\*i1110;

 A(5,6) = (j011-j020).\*i0100;

 A(6,6) = (j011-j020).\*i0101+cpxunit;

 % calculate matrix B

 % For coherent RBS

 B(1,1) = -i0100;

 B(2,1) = -i0101;

 B(3,1) = -i0111;

 B(4,1) = -i0110;

 B(5,1) = zero;

 B(6,1) = zero;

 % For spontaneous RBS

 B(1,2) = -i0000;

 B(2,2) = -i0001;

 B(3,2) = -i0011;

 B(4,2) = -i0010;

 B(5,2) = zero;

 B(6,2) = zero;

 % solve the equation for X

 X=A\B;

% cohsig(i) = X(1,1).\*conj(X(1,1));

 sptsig(i) = 2.\*real(X(1,2));

end

% area normalization

TentiSpectrum = sptsig./polyarea([min(v1) v1' max(v1)],[0 sptsig 0]);

FSR = 2.9929;% the value of FSR, unit GHz

FWHM = 0.058;% the linewidth of the instrument function, unit GHz

Airy = 1./(1+(2.\*FSR./(pi.\*FWHM)).^2.\*sin(pi./FSR.\*v1).^2); % Calculate the instrument function

% convolve with instrument function

TentiSpectrum\_conv = conv(Airy,TentiSpectrum,'same');

% area normalization

TentiSpectrum\_conv = TentiSpectrum\_conv./polyarea([min(v1) v1' max(v1)],[0 ...

 TentiSpectrum\_conv' 0]); %

% plot figure;

figure;

plot(v1,TentiSpectrum\_conv)