This program intends to calculate the dielectric function tensor of a polar crystal at frequencies near the phonon frequency. The formula used in this program is:

$$\varepsilon_{\beta\sigma} = \varepsilon_\infty + \frac{e^2}{\varepsilon_0 \Omega} \sum_{\mu j' k} \frac{1}{\left[\omega_{pk}(\vec{q})\right]^2 - \omega_L^2 - i\gamma\omega_L} \frac{Z_{\beta\mu}(j')}{\sqrt{m_{j'}}} d_k(\mu j', \vec{q}, t) * \sum_{\alpha j} \frac{Z_{\sigma\alpha}(j)}{\sqrt{m_j}} d_k^*(\alpha j, \vec{q}, t)$$

It is important to note that in this formula, the dielectric function depends not only on the polarization direction of the electric field but also on the wave vector direction of the electromagnetic wave (or phonon polaritons). For detailed theoretical background, please refer to: [Yue Fang, Huanjun Chen, Zhibing Li and Weiliang Wang, Hyperbolic phonon polaritons and wave vector direction dependent dielectric tensors in anisotropic crystals, *Journal of Physical Chemistry C*, **128** (2024) 7359.]
https://doi.org/10.1021/acs.jpcc.4c01183
Please cite this paper when publishing results obtained with this program.

We do not need to install the program; just put all the M files into one folder. We need to prepare 5 input files:
OUTCAR.txt
qpoints.yaml
band.yaml
phonopy.yaml
lifetime.dat

Methods for preparing input files:

OUTCAR.txt
Perform a VASP static calculation to obtain the Born effective charges for the unit cell, resulting in an OUTCAR. Save the following content from it as OUTCAR.txt. (Note: Remove content such as PIEZOELECTRIC TENSOR, etc.)

```
MACROSCOPIC STATIC DIELECTRIC TENSOR (including local field effects)
 -------------------------------------
             5.447      -0.000       0.000
            -0.000       6.208       0.000
             0.000       0.000       4.248
 -------------------------------------


 BORN EFFECTIVE CHARGES (including local field effects)
 -----------------------------------------------------
ion     1
    1      6.51576      -0.00000      -0.28531
    2     -0.00000       7.83547       0.00000
    3     -0.54889       0.00000       4.47872
```

ion	2		
1	6.51576	-0.00000	-0.28531
2	-0.00000	7.83547	0.00000
3	-0.54889	0.00000	4.47872
ion	3		
1	6.51576	-0.00000	0.28531
2	-0.00000	7.83547	0.00000
3	0.54889	0.00000	4.47872
ion	4		
1	6.51576	-0.00000	0.28531
2	-0.00000	7.83547	0.00000
3	0.54889	0.00000	4.47872
ion	5		
1	-1.14288	0.00000	0.35376
2	0.00000	-6.11201	0.00000
3	0.19852	0.00000	-1.59655
ion	6		
1	-1.14288	0.00000	0.35376
2	0.00000	-6.11201	0.00000
3	0.19852	0.00000	-1.59655
ion	7		
1	-1.14288	0.00000	-0.35376
2	0.00000	-6.11201	0.00000
3	-0.19852	0.00000	-1.59655
ion	8		
1	-1.14288	0.00000	-0.35376
2	0.00000	-6.11201	0.00000
3	-0.19852	0.00000	-1.59655
ion	9		
1	-4.76414	0.00000	0.32873
2	0.00000	-1.16597	0.00000
3	0.47715	0.00000	-0.66670
ion	10		
1	-4.76414	0.00000	0.32873
2	0.00000	-1.16597	0.00000
3	0.47715	0.00000	-0.66670
ion	11		
1	-4.76414	0.00000	-0.32873
2	0.00000	-1.16597	0.00000
3	-0.47715	0.00000	-0.66670
ion	12		
1	-4.76414	0.00000	-0.32873
2	0.00000	-1.16597	0.00000
3	-0.47715	0.00000	-0.66670

```
ion     13
     1      -0.60880        0.00000        0.36885
     2       0.00000       -0.55743        0.00000
     3       0.30126        0.00000       -2.21650
ion     14
     1      -0.60880        0.00000        0.36885
     2       0.00000       -0.55743        0.00000
     3       0.30126        0.00000       -2.21650
ion     15
     1      -0.60880        0.00000       -0.36885
     2       0.00000       -0.55743        0.00000
     3      -0.30126        0.00000       -2.21650
ion     16
     1      -0.60880        0.00000       -0.36885
     2       0.00000       -0.55743        0.00000
     3      -0.30126        0.00000       -2.21650
```

band.yaml, phonopy.yaml

Use the following band.conf:

```
ATOM_NAME = ***
DIM = ***
FORCE_SETS = READ
EIGENVECTORS = .TRUE.
BAND_POINTS = 2
BAND = 0.0 0.0 0.0    0.5 0.0 0.0
```

This sets the direction of the light wave vector along the first reciprocal lattice vector. If q is along the second reciprocal lattice vector, change the last line to:

```
BAND = 0.0 0.0 0.0    0.0 0.5 0.0
```

Similarly for other directions.

Execute phonopy -c POSCAR -p -s band.conf --nac

to obtain band.yaml and phonopy.yaml.

qpoints.yaml

Use the following band.conf:

```
ATOM_NAME = ***
DIM = ***
FORCE_SETS = READ
EIGENVECTORS = .TRUE.
WRITEDM = .TRUE.
QPOINTS = .TRUE.
```

Use the following QPOINTS file:

```
2
0.0    0.0    0.0
```

0.5 0.0 0.0

This sets q along the first reciprocal lattice vector. If q is along the second reciprocal lattice vector, change the third line to 0.0 0.5 0.0, similarly for other directions.
Execute phonopy -c POSCAR -p -s band.conf --nac
to obtain qpoints.yaml.

lifetime.dat
Rename the phono3py output file gammas-***.dat to lifetime.dat. The direction of q used in phono3py should be consistent with the previous two steps.

Execute calculate_dielectric.m with MATLAB
Then we get the results in dielectric.txt. The first column is frequency, the second column is wavelength, the third column is wavenumber, the 4th, 7th, and 9th columns are the real parts of the dielectric function for polarization along the X, Y, and Z directions, respectively, and the 10th, 13th, and 15th columns are the imaginary parts of the dielectric function for polarization along the X, Y, and Z directions, respectively. In the principal axis coordinate system, the 5th, 6th, 8th, 11th, 12th, and 14th columns should be much smaller than the aforementioned dielectric function columns. If you want to plot the imaginary part of the dielectric function, you need to rotate the coordinate system to the principal axis coordinate system.