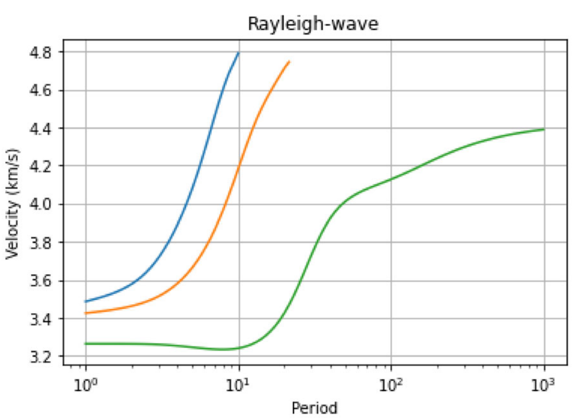
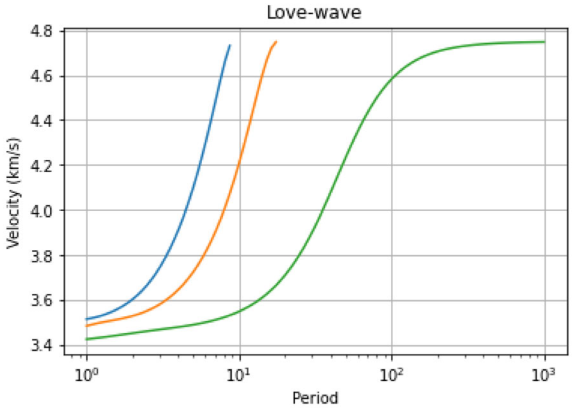
In [1]:
```
#documentation at https://github.com/keurfonluu/disba

import numpy as np
from disba import PhaseDispersion
import matplotlib.pyplot as plt
```

In [2]:
```
import numpy
from disba import PhaseDispersion

# Velocity model
# thickness, Vp, Vs, density
# km, km/s, km/s, g/cm3
velocity_model = numpy.array([
   [10.0, 7.00, 3.50, 2.00],
   [10.0, 6.80, 3.40, 2.00],
   [10.0, 7.00, 3.50, 2.00],
   [10.0, 7.60, 3.80, 2.00],
   [10.0, 8.40, 4.20, 2.00],
   [10.0, 9.00, 4.50, 2.00],
   [10.0, 9.40, 4.70, 2.00],
   [10.0, 9.60, 4.80, 2.00],
   [10.0, 9.50, 4.75, 2.00],
])

# Periods must be sorted starting with low periods
t = numpy.logspace(0.0, 3.0, 100)

# Compute the 3 first Rayleigh- and Love- wave modal dispersion curves
# Fundamental mode corresponds to mode 0
pd = PhaseDispersion(*velocity_model.T)
cpr = [pd(t, mode=i, wave="rayleigh") for i in range(3)]
cpl = [pd(t, mode=i, wave="love") for i in range(3)]

# pd returns a namedtuple (period, velocity, mode, wave, type)
```

In [3]:
```
plt.semilogx(cpr[2][0], cpr[2][1])
plt.semilogx(cpr[1][0], cpr[1][1])
plt.semilogx(cpr[0][0], cpr[0][1])
plt.title("Rayleigh-wave")
plt.xlabel("Period")
plt.ylabel("Velocity (km/s)")
plt.grid()
plt.show()
```



In [4]:
```
plt.semilogx(cpl[2][0], cpl[2][1])
plt.semilogx(cpl[1][0], cpl[1][1])
plt.semilogx(cpl[0][0], cpl[0][1])
plt.title("Love-wave")
plt.grid()
plt.xlabel("Period")
plt.ylabel("Velocity (km/s)")
plt.show()
```



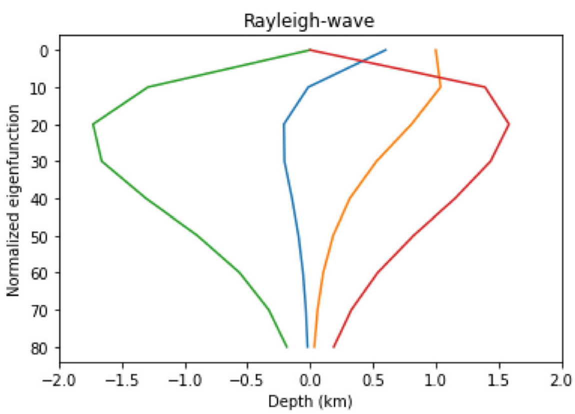
In [6]:

```
from disba import EigenFunction
eigf = EigenFunction(*velocity_model.T)
eigr = eigf(20.0, mode=0, wave="rayleigh")
eigl = eigf(20.0, mode=0, wave="love")

# eigf returns a namedtuple
#  - (depth, ur, uz, tz, tr, period, mode) for Rayleigh-wave
#  - (depth, uu, tt, period, mode) for Love-wave
```

In [7]:

```
plt.gca().invert_yaxis()
plt.plot(eigr[1],eigr[0])
plt.plot(eigr[2],eigr[0])
plt.plot(eigr[3],eigr[0])
plt.plot(eigr[4],eigr[0])
plt.xlabel("Depth (km)")
plt.ylabel("Normalized eigenfunction")
plt.xlim(-2,2)
plt.title("Rayleigh-wave")
```

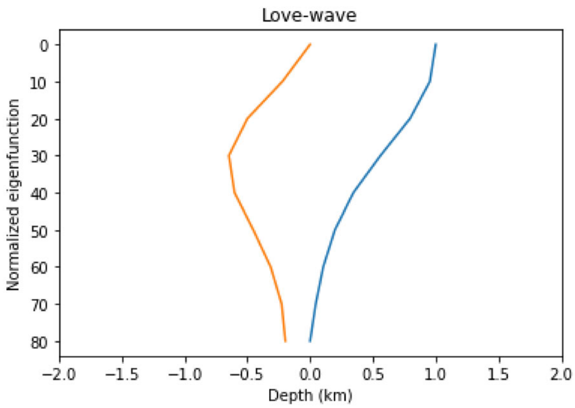
Out[7]: `Text(0.5, 1.0, 'Rayleigh-wave')`



In [8]:

```
plt.gca().invert_yaxis()
plt.plot(eigl[1],eigl[0])
plt.plot(eigl[2],eigl[0])
plt.xlabel("Depth (km)")
plt.ylabel("Normalized eigenfunction")
plt.xlim(-2,2)
plt.title("Love-wave")
```

Out[8]: `Text(0.5, 1.0, 'Love-wave')`

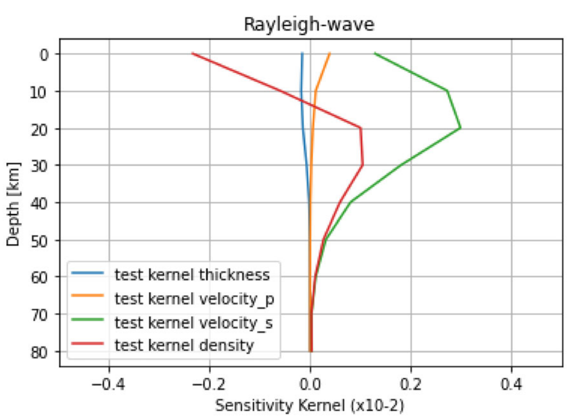


In [9]:
```python
from disba import PhaseSensitivity

ps = PhaseSensitivity(*velocity_model.T)
parameters = ["thickness", "velocity_p", "velocity_s", "density"]
skr = [ps(20.0, mode=0, wave="rayleigh", parameter=parameter) for parameter in parameters]
skl = [ps(20.0, mode=0, wave="love", parameter=parameter) for parameter in parameters]

# ps returns a namedtuple (depth, kernel, period, velocity, mode,wave, type, parameter)
```

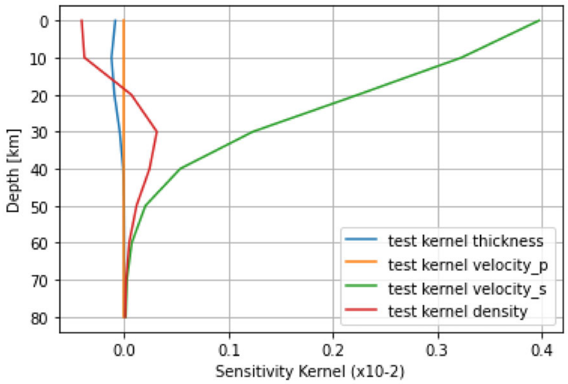
In [10]:
```python
skr[2]
```

Out[10]:
```
SensitivityKernel(depth=array([ 0., 10., 20., 30., 40., 50., 60., 70., 80.]), kernel=array([0.12909738, 0.2727874 , 0.2992
6339, 0.18128212, 0.08069778,
       0.03154188, 0.01120242, 0.00279439, 0.00328742]), period=20.0, velocity=3.423702611993072, mode=0, wave='rayleigh',
type='phase', parameter='velocity_s')
```

In [12]:
```python
plt.gca().invert_yaxis()
plt.plot(skr[0][1], skr[0][0],label='test kernel thickness')
plt.plot(skr[1][1], skr[1][0],label='test kernel velocity_p')
plt.plot(skr[2][1], skr[2][0],label='test kernel velocity_s')
plt.plot(skr[3][1], skr[3][0],label='test kernel density')
plt.xlim(-0.5,0.5)
plt.grid()
plt.legend()
plt.title("Rayleigh-wave")
plt.xlabel("Sensitivity Kernel (x10-2)")
plt.ylabel("Depth [km]")
plt.show()
```



In [13]:
```python
plt.gca().invert_yaxis()

plt.plot(skl[0][1], skl[0][0],label='test kernel thickness')
plt.plot(skl[1][1], skl[1][0],label='test kernel velocity_p')
plt.plot(skl[2][1], skl[2][0],label='test kernel velocity_s')
plt.plot(skl[3][1], skl[3][0],label='test kernel density')
#plt.xlim(-0.5,0.5)
plt.grid()
plt.legend()
#plt.ylim(0.035,0)
plt.xlabel("Sensitivity Kernel (x10-2)")
plt.ylabel("Depth [km]")
plt.show()
```



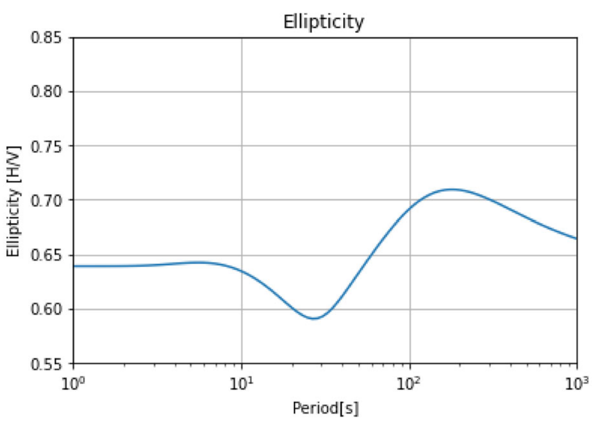
In [16]:

```
from disba import Ellipticity, EllipticitySensitivity

ell = Ellipticity(*velocity_model.T)
rel = ell(t, mode=0)

# ell returns a namedtuple (period, ellipticity, mode)

es = EllipticitySensitivity(*velocity_model.T)
ek = [es(20.0, mode=0, parameter=parameter) for parameter in parameters]

# es returns a namedtuple (depth, kernel, period, velocity, mode, wave, type, parameter)
```

In [27]:

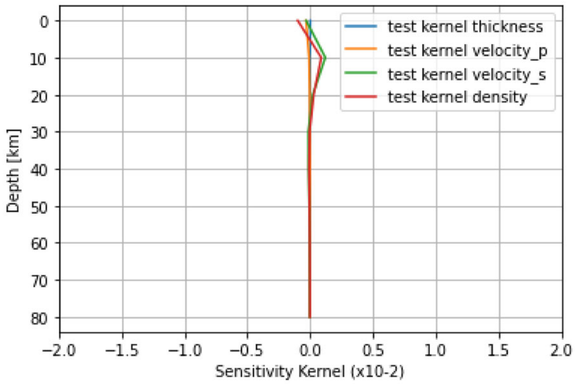
```
plt.semilogx(rel[0],rel[1])
plt.ylim(0.55,0.85)
plt.xlim(1,1000)
plt.grid()
plt.title("Ellipticity")
plt.xlabel('Period[s]')
plt.ylabel("Ellipticity [H/V]")
```

Out[27]: Text(0, 0.5, 'Ellipticity [H/V]')



In [22]:

```
plt.gca().invert_yaxis()
plt.plot(ek[0][1], ek[0][0],label='test kernel thickness')
plt.plot(ek[1][1], ek[1][0],label='test kernel velocity_p')
plt.plot(ek[2][1], ek[2][0],label='test kernel velocity_s')
plt.plot(ek[3][1], ek[3][0],label='test kernel density')
plt.xlim(-2,2)
plt.grid()
plt.legend()
plt.xlabel("Sensitivity Kernel (x10-2)")
plt.ylabel("Depth [km]")
plt.show()
```


test kernel thickness
test kernel velocity_p
test kernel velocity_s
test kernel density
Depth [km]
0
10
20
30
40
50
60
70
80
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Sensitivity Kernel (x10-2)

In [ ]: