3. Calculating phase velocities

1. Phase-difference
2. Cross correlation
3. τ-p transform in time domain
4. τ-p transform in Frequency domain（MASW）
5. Spatial Auto-correlation (SPAC)
6. Seismic interferometry

1.Phase difference

Calculate the phase difference between two waves; *f(t)* and *g(t)*.

Fourier transform

$$F(\omega)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} f(t)\cdot \exp^{-i\omega t}dt \qquad G(\omega)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} g(t)\cdot \exp^{-i\omega t}dt$$

Amplitude and phase

$$F(\omega)=A_f(\omega)\cdot \exp^{-i\phi_f(\omega)} \qquad G(\omega)=A_g(\omega)\cdot \exp^{-i\phi_g(\omega)}$$

Phase difference

$$\Delta\phi(\omega)=\phi_f(\omega)-\phi_g(\omega)$$

Phase velocity（$c(\omega)$）

$$c(\omega)=\frac{\omega\cdot\Delta x}{\Delta\phi_f(\omega)}$$

$$\boxed{c(\omega)=\frac{\omega\cdot\Delta x}{\Delta\phi_f(\omega)+2n\pi}}$$

2.Cross-correlation

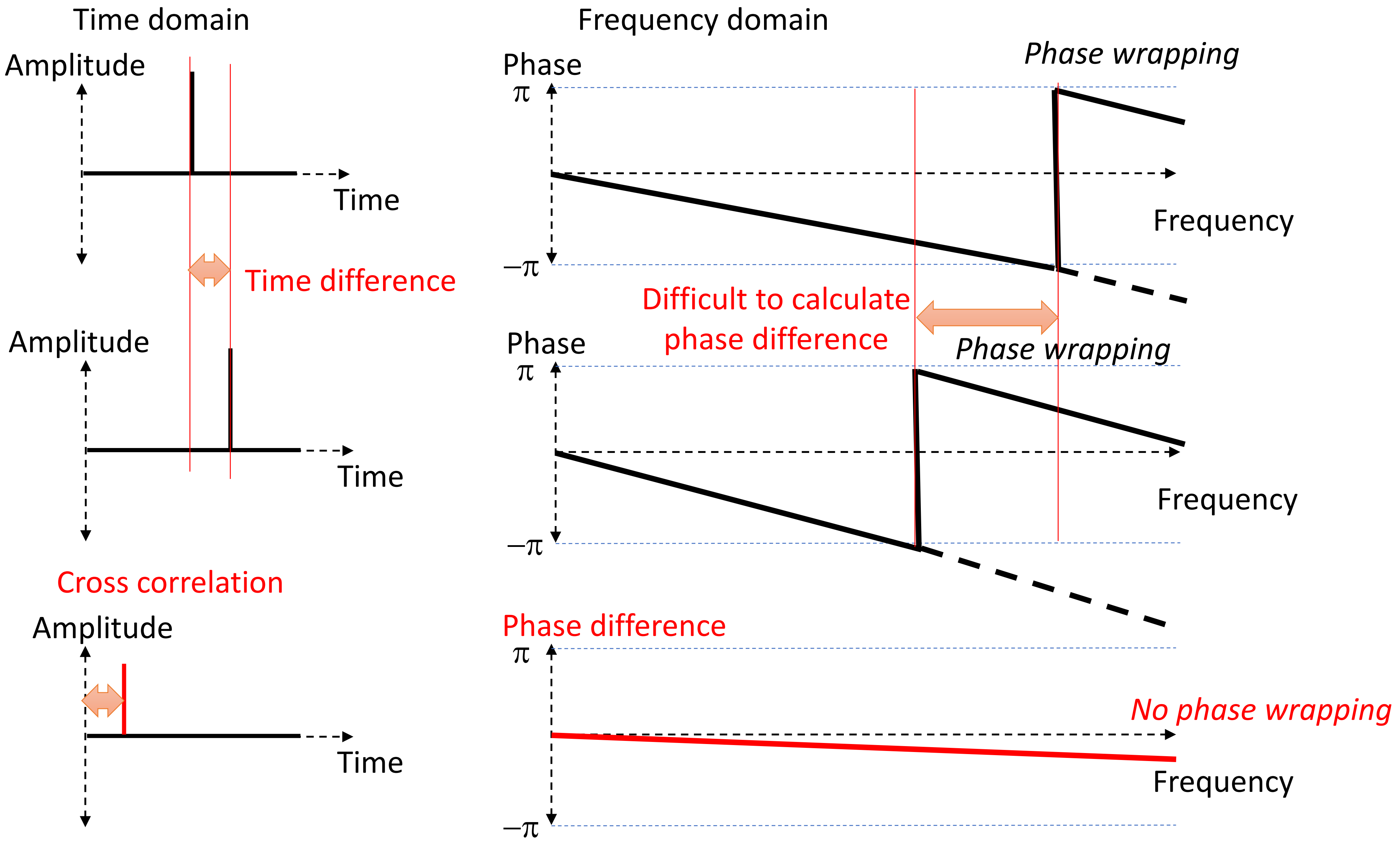
Fourier transform

$$F(\omega)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} f(t)\cdot \exp^{-i\omega t} dt \qquad G(\omega)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} g(t)\cdot \exp^{-i\omega t} dt$$

Cross-correlation

$$CC_{fg}(\omega)=F(\omega)\cdot\overline{G(\omega)}=A_f(\omega)A_g(\omega)\cdot\exp^{i\Delta\phi(\omega)}$$

Phase difference and cross-correlation



Simple example of phase-velocity calculation using cross-correlation

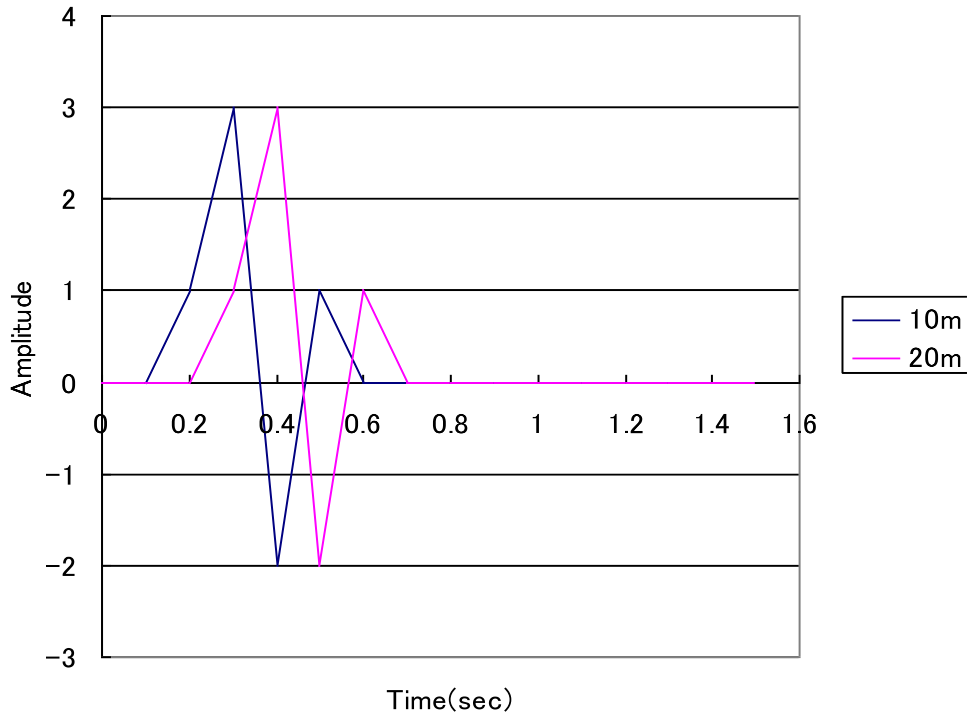
Time domain			Frequency domain					Cross-correlation		Phase-difference
time(sec)	10m	20m	Frequency(Hz)	Re(10m)	Im(10m)	Re(20m)	Im(20m)	Re(CC)	Im(CC)	Phase-difference
0	0	0	0	3	0	3	0	9	0	0
0.1	0	0	0.666666667	1.472474	−2.40262	0.440944	−2.78323	7.336329	3.038807	0.392699
0.2	1	0	1.333333333	−0.82843	−2.41421	−2.29289	−1.12132	4.606602	4.606602	0.785398
0.3	3	1	2	−2.55487	−1.17637	−2.06453	1.91021	3.027483	7.308989	1.178097
0.4	−2	3	2.666666667	−3	2	2	3	0	13	1.570796
0.5	1	−2	3.333333333	1.140652	4.237841	3.478745	−2.67558	−7.37063	17.79427	1.963495
0.6	0	1	4	4.828427	−0.41421	−3.70711	−3.12132	−16.6066	16.6066	2.356194
0.7	0	0	4.666666667	−0.05826	−4.98841	−1.85516	4.630986	−22.9932	9.524088	2.748894
0.8	0	0	5.333333333	−5	0	5	0	−25	0	3.141593
0.9	0	0	6	−0.05826	4.988411	−1.85516	−4.63099	−22.9932	−9.52409	−2.74889
1	0	0	6.666666667	4.828427	0.414214	−3.70711	3.12132	−16.6066	−16.6066	−2.35619
1.1	0	0	7.333333333	1.140652	−4.23784	3.478745	2.675577	−7.37063	−17.7943	−1.9635
1.2	0	0	8	−3	−2	2	−3	0	−13	−1.5708
1.3	0	0	8.666666667	−2.55487	1.176373	−2.06453	−1.91021	3.027483	−7.30899	−1.1781
1.4	0	0	9.333333333	−0.82843	2.414214	−2.29289	1.12132	4.606602	−4.6066	−0.7854
1.5	0	0	10	1.472474	2.402625	0.440944	2.783227	7.336329	−3.03881	−0.3927

Cross-correlation (CC): $CC = (a+bi)(c-di) = ac+bd+(bc-ad)i$

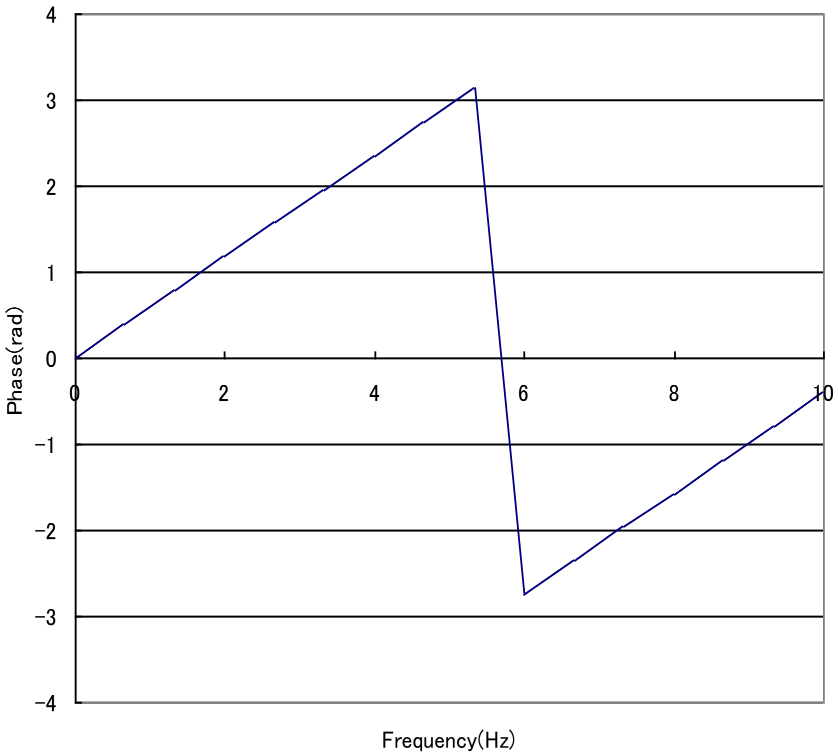
Phase-difference (Δϕ): $\Delta\phi = \tan^{-1}\frac{b}{a}$

Simple example of phase-velocity calculation using cross-correlation

Time-domain waveform



Phase-difference



Simple example of phase-velocity calculation using cross-correlation

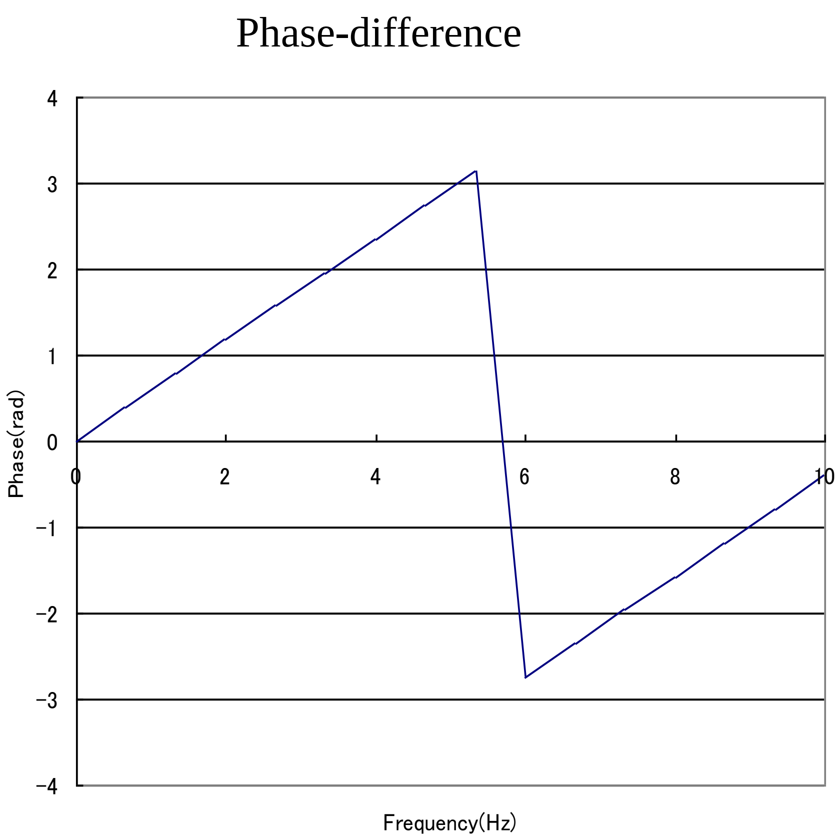
Frequency(Hz)	Phase difference	Phase−velocity(m/s)
0	0	
0.625	0.392699082	100
1.25	0.785398163	100
1.875	1.178097245	100
2.5	1.570796327	100
3.125	1.963495408	100
3.75	2.35619449	100
4.375	2.748893572	100
5	3.141592654	100
5.625	−2.748893572	
6.25	−2.35619449	
6.875	−1.963495408	
7.5	−1.570796327	
8.125	−1.178097245	
8.75	−0.785398163	
9.375	−0.392699082	

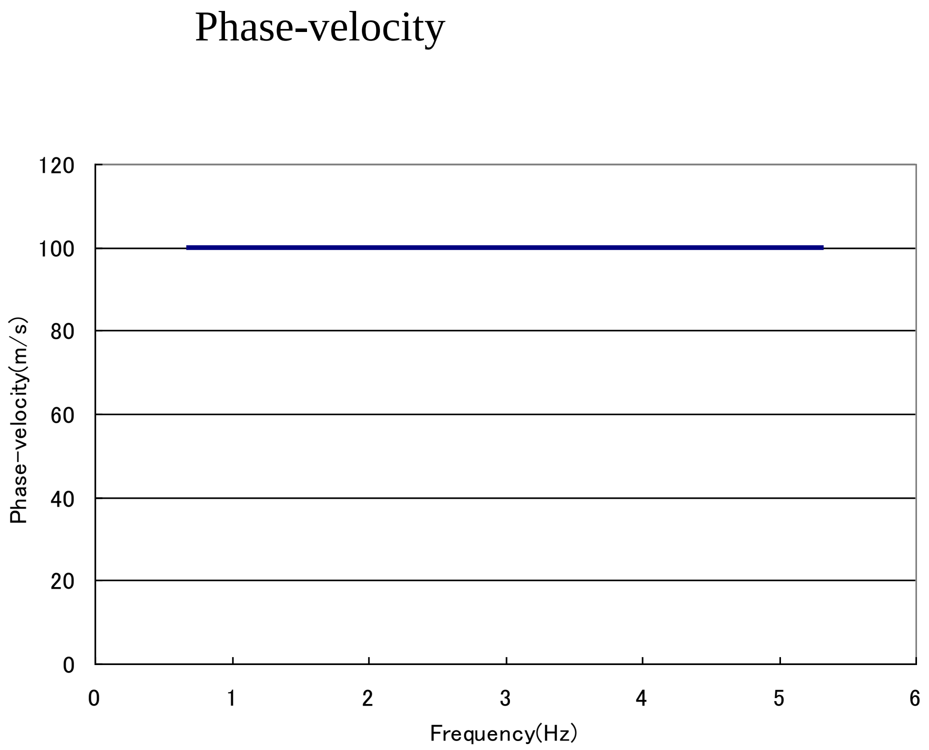
Phase-velocity ($c(\omega)$)

$$c(\omega) = \frac{\omega \cdot \Delta x}{\Delta\phi_f(\omega) + 2n\pi}$$

$$Phase - velocity = \frac{Frequency \cdot 2\pi \cdot \text{Receiver} - spacing}{\text{Phase} - \text{difference}}$$

Simple example of phase-velocity calculation using cross-correlation





Cross-correlation by Python

```
cc=F*np.conj(G)

print(cc)
```

Results

```
[ 9. +0.j 7.33632883 +3.0388069j
 4.60660172 +4.60660172j 3.02748251 +7.30898934j
 0. +13.j -7.37062826+17.79427071j
 -16.60660172+16.60660172j -22.99318308 +9.52408827j
 -25. +0.j -22.99318308 -9.52408827j
 -16.60660172-16.60660172j -7.37062826-17.79427071j
 0. -13.j 3.02748251 -7.30898934j
 4.60660172 -4.60660172j 7.33632883 -3.0388069j ]
```

```
dc=np.angle(cc)

print(dc)
```

Results

```
[ 0. 0.39269908
 0.78539816 1.17809725
 1.57079633 1.96349541
 2.35619449 2.74889357
 3.14159265 -2.74889357
 -2.35619449 -1.96349541
 -1.57079633 -1.17809725
 -0.78539816 -0.39269908]
```

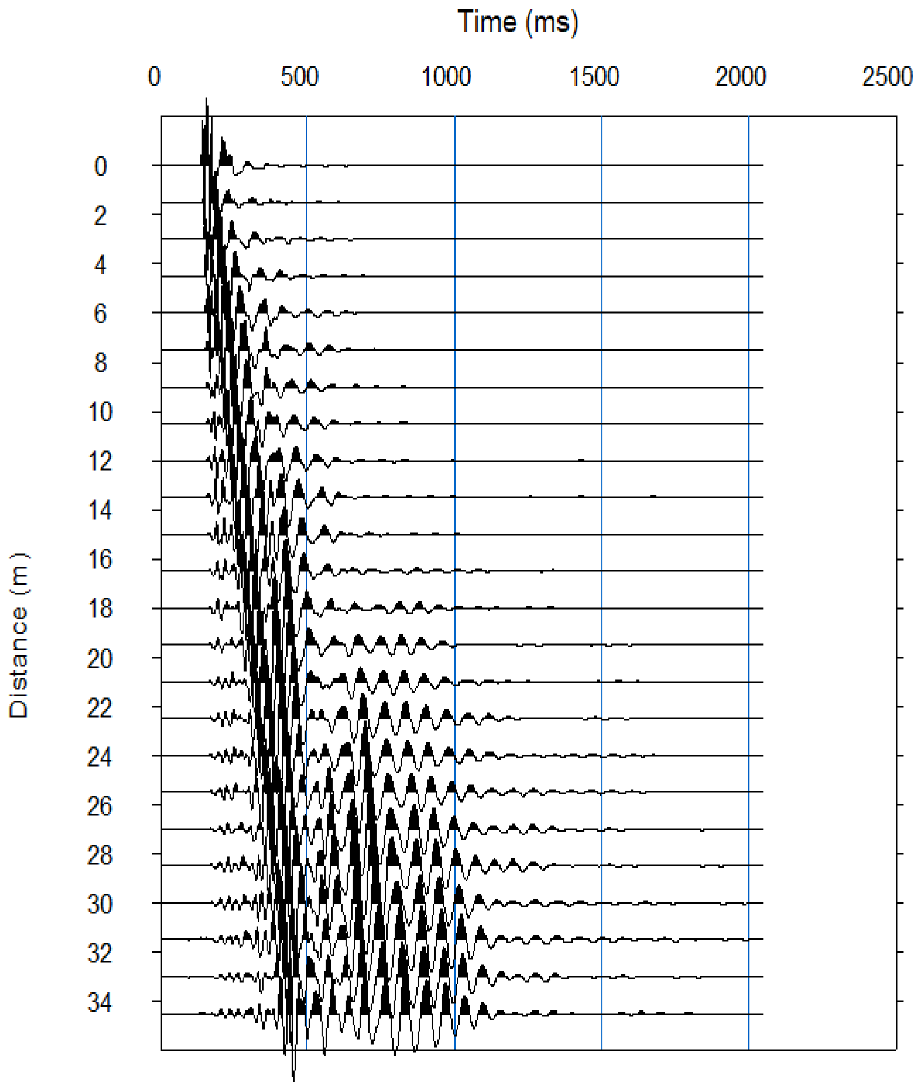
```
pv=freq*2*math.pi*10/dc

print(pv)
```

Results

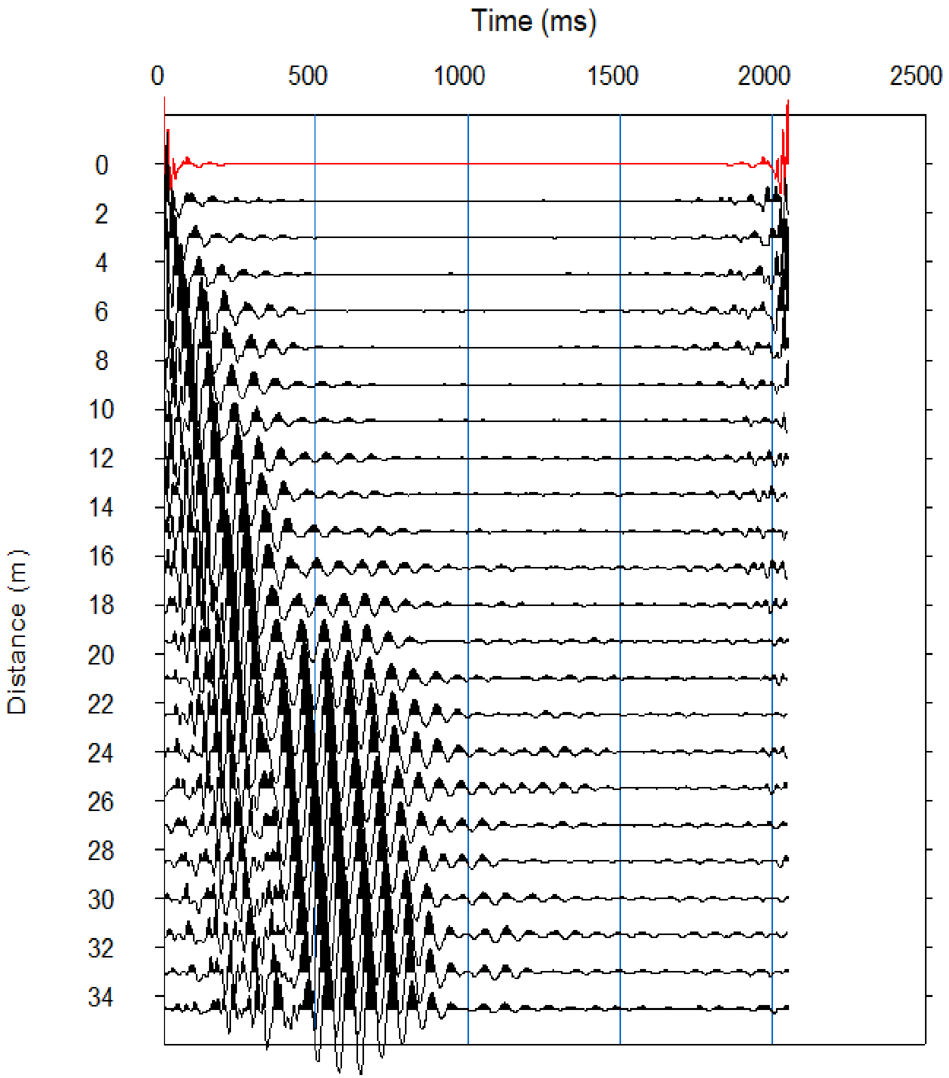
```
[ nan 100. 100. 100. 100. 100. 100. 100.
 -100. 100. 100. 100. 100. 100. 100. 100.]
```

Real data example

Raw data

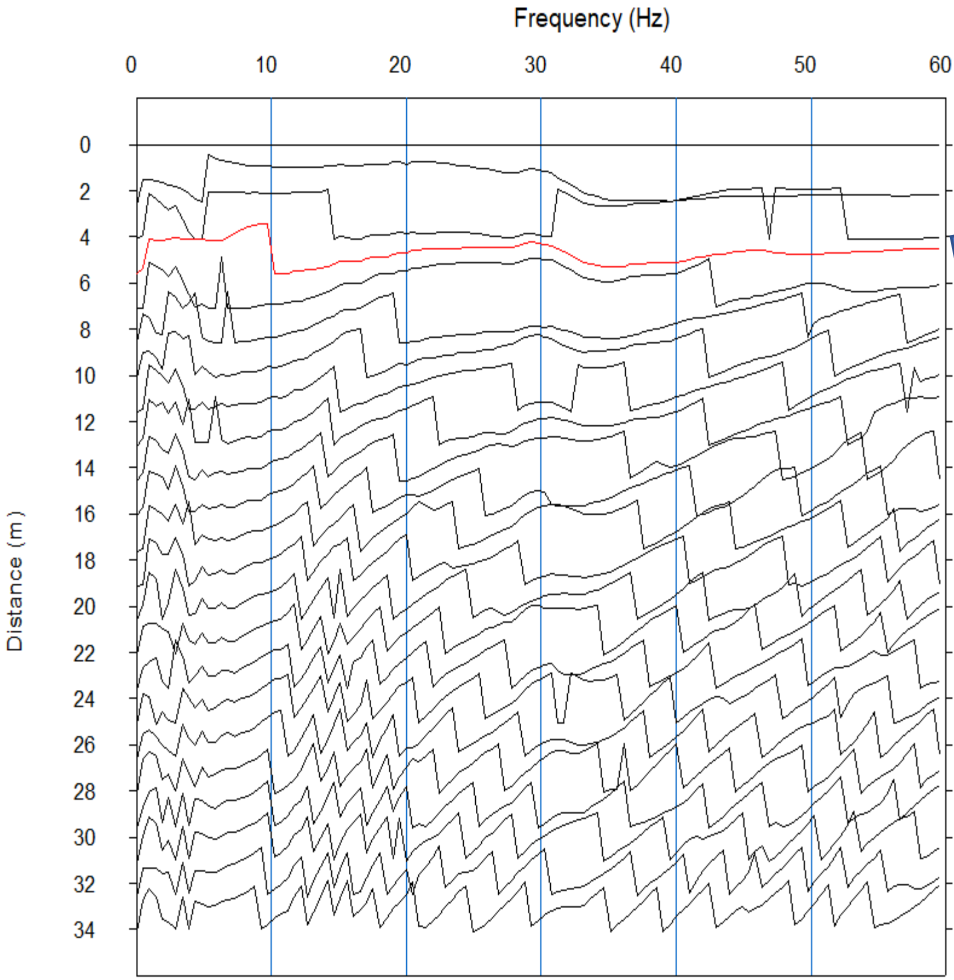


Cross-correlation

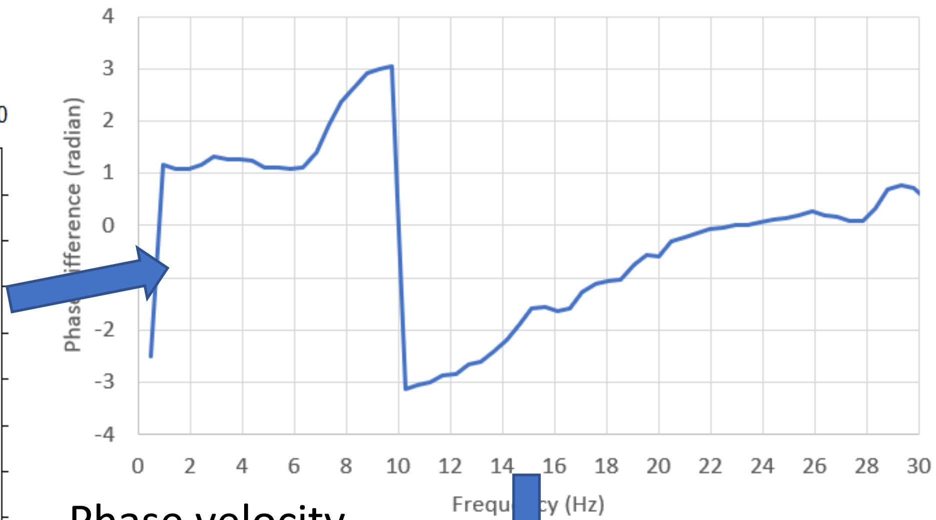


Real data example

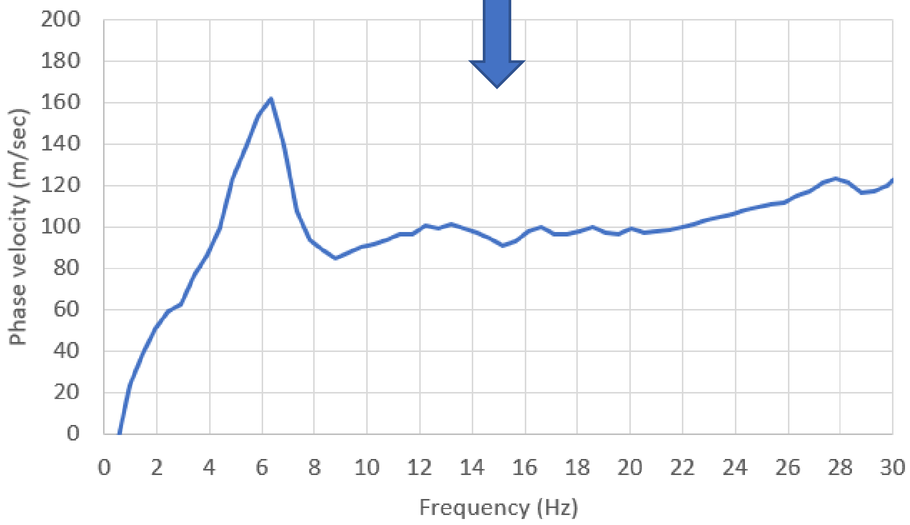
Phase spectrum of cross-correlation



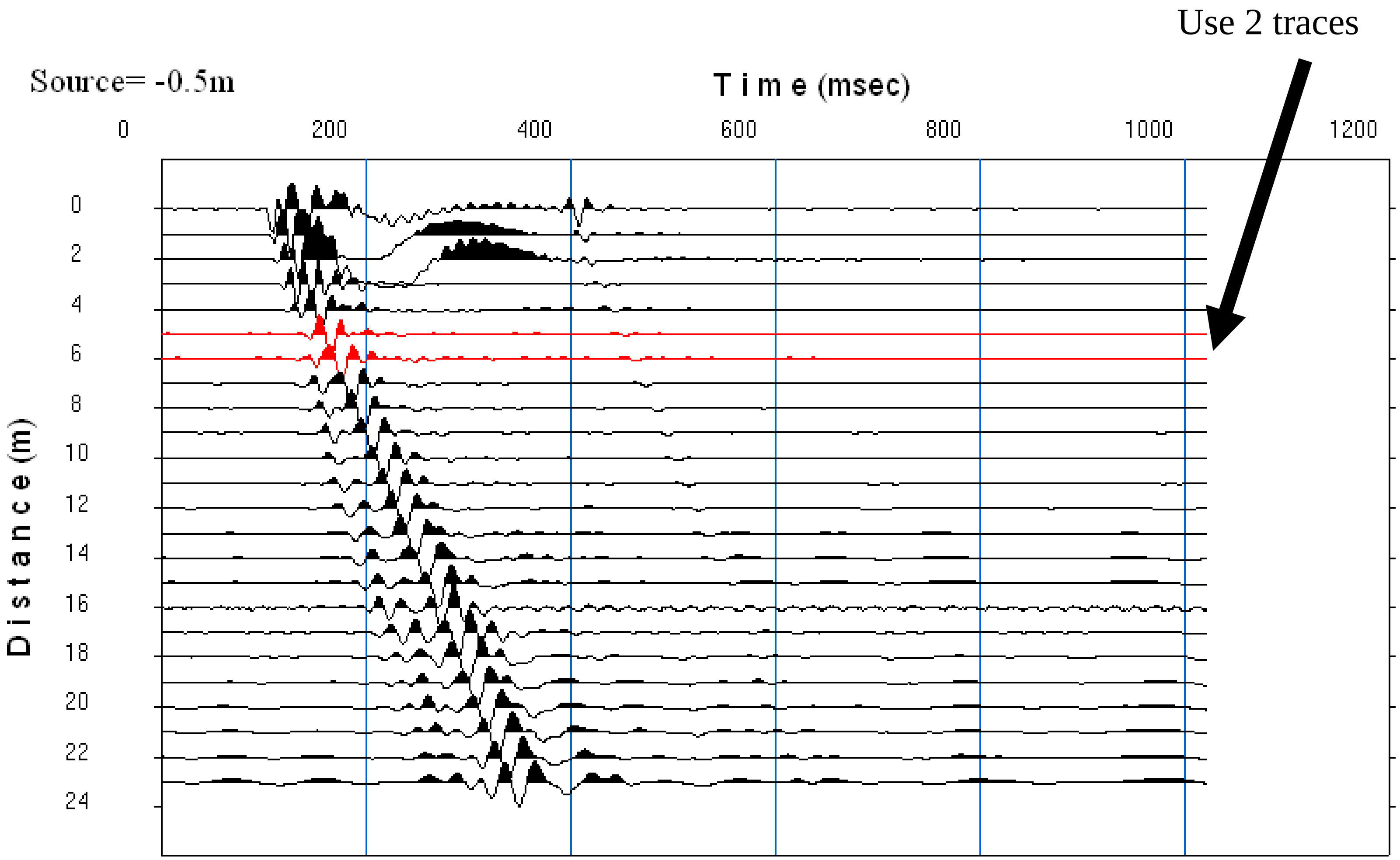
Phase spectrum of cross-correlation



Phase velocity

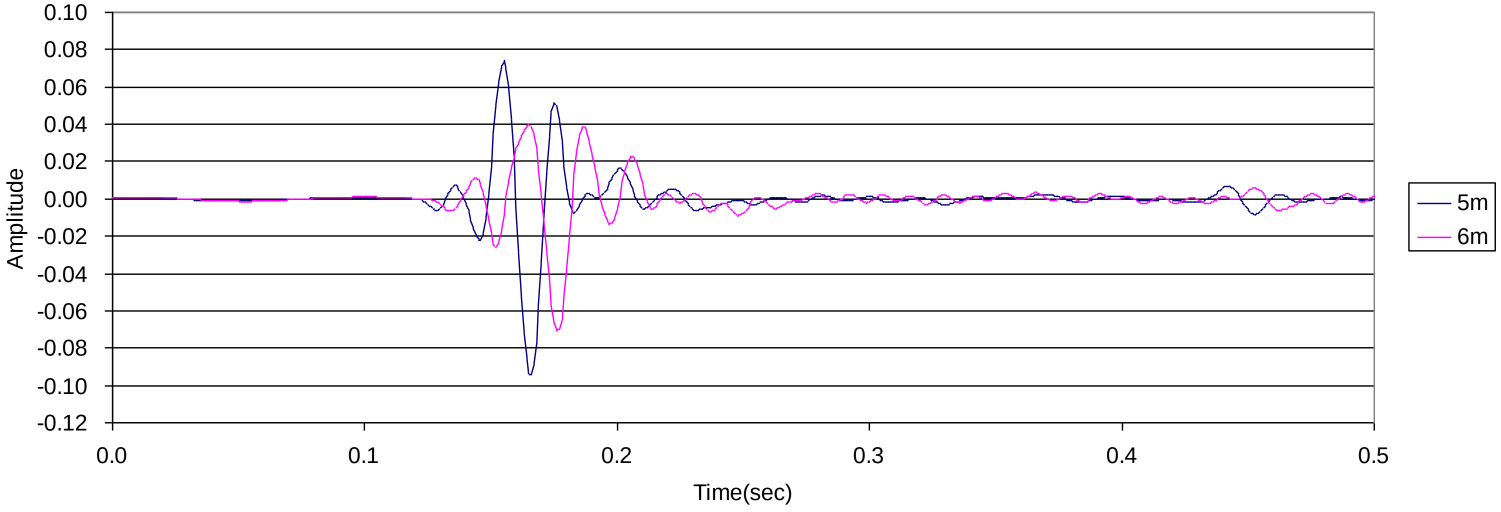


Real data example

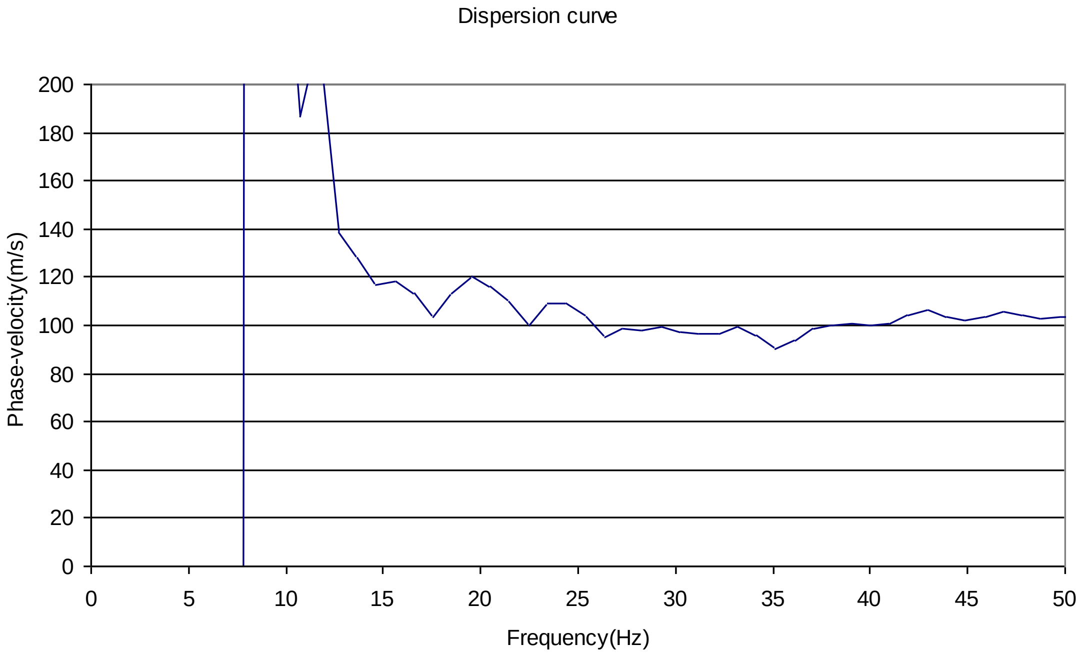


Real data example

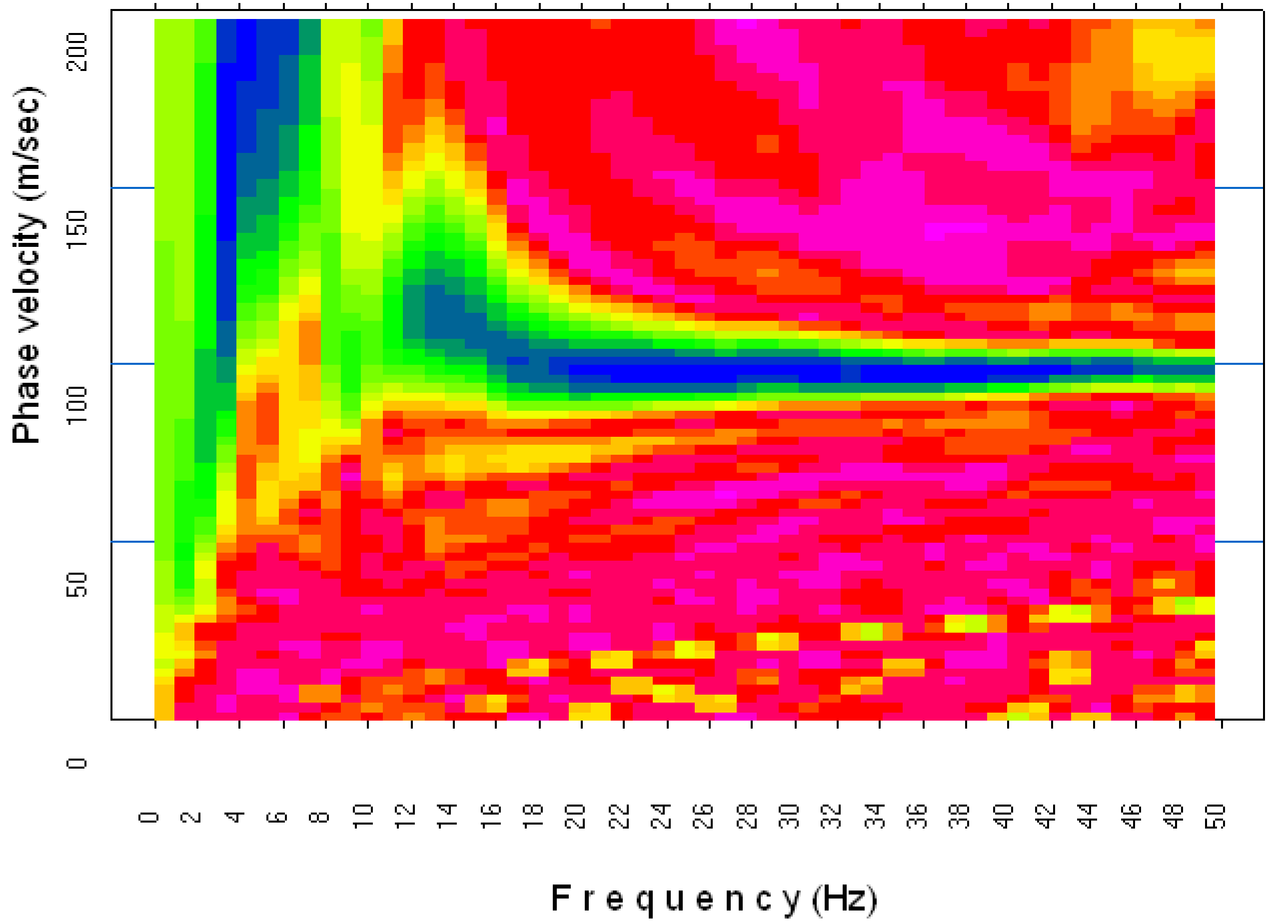
Waveform data



Dispersion curve



Real data example
(Use all traces and calculated by MASW)



3. τ-p transform (in time domain)

τ-p transform

$$F(\tau, p) = \int_{-\infty}^{+\infty} f(x, t + xp)\,dx$$

$$p = \frac{1}{c} \qquad \tau = t$$

τ :intercept time
p:ray parameter
c:(phase) velocity

Fourier transform

$$F\left(\omega, \frac{1}{c}\right) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F\left(\tau, \frac{1}{c}\right) \cdot e^{-i\omega\tau}\, d\tau$$

3.τ-p transform (in time domain)

τ-p transform

$$f(\tau, p) = \int_{-\infty}^{+\infty} f(x, t + xp)\,dx$$

$$p = \frac{1}{c} \qquad \tau = t$$

τ :intercept time
p:ray parameter
c:(phase) velocity

Fourier transform

$$F\left(\omega, \frac{1}{c}\right) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F\left(\tau, \frac{1}{c}\right) \cdot e^{-i\omega\tau}\,d\tau$$

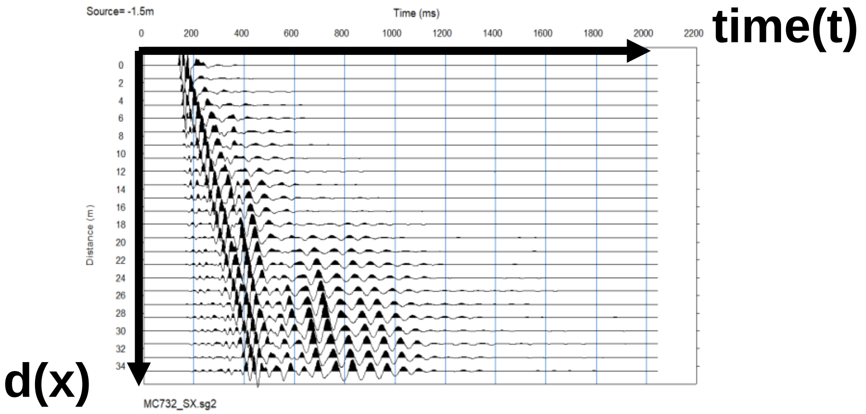
$$f(x, t)$$

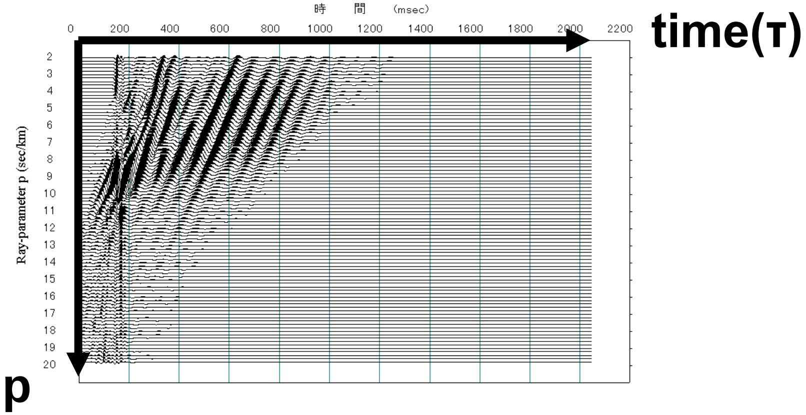
$$\downarrow$$

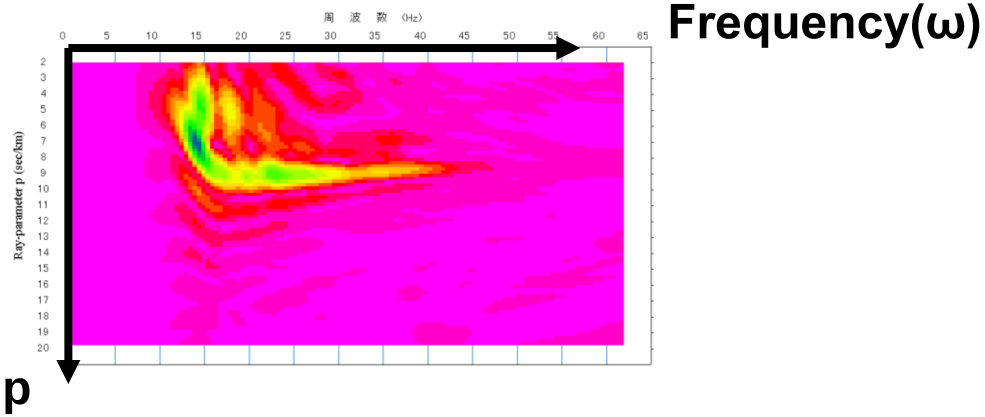
$$f(\tau, p)$$

$$\downarrow$$

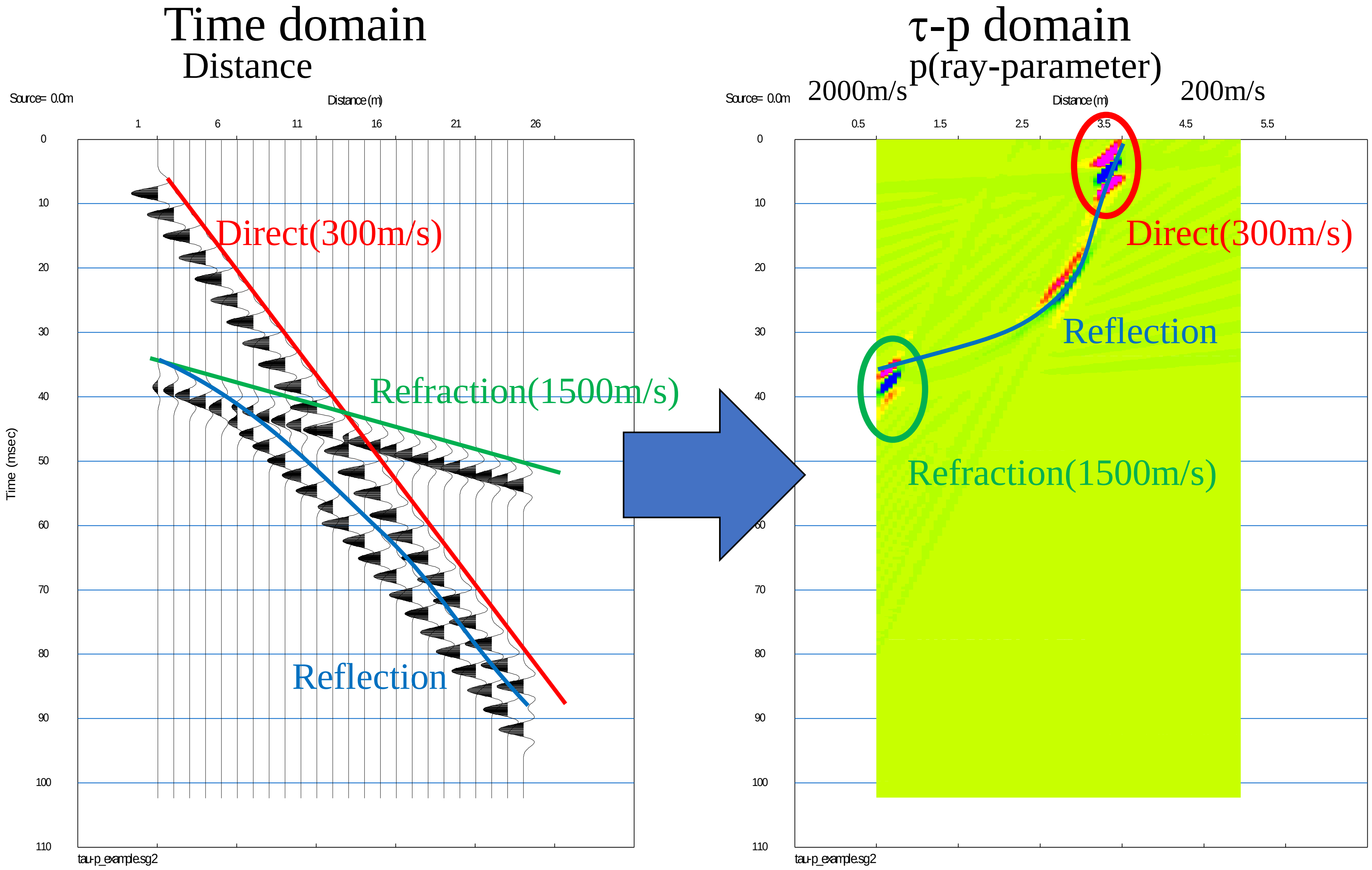
$$F\left(\omega, \frac{1}{c}\right)$$





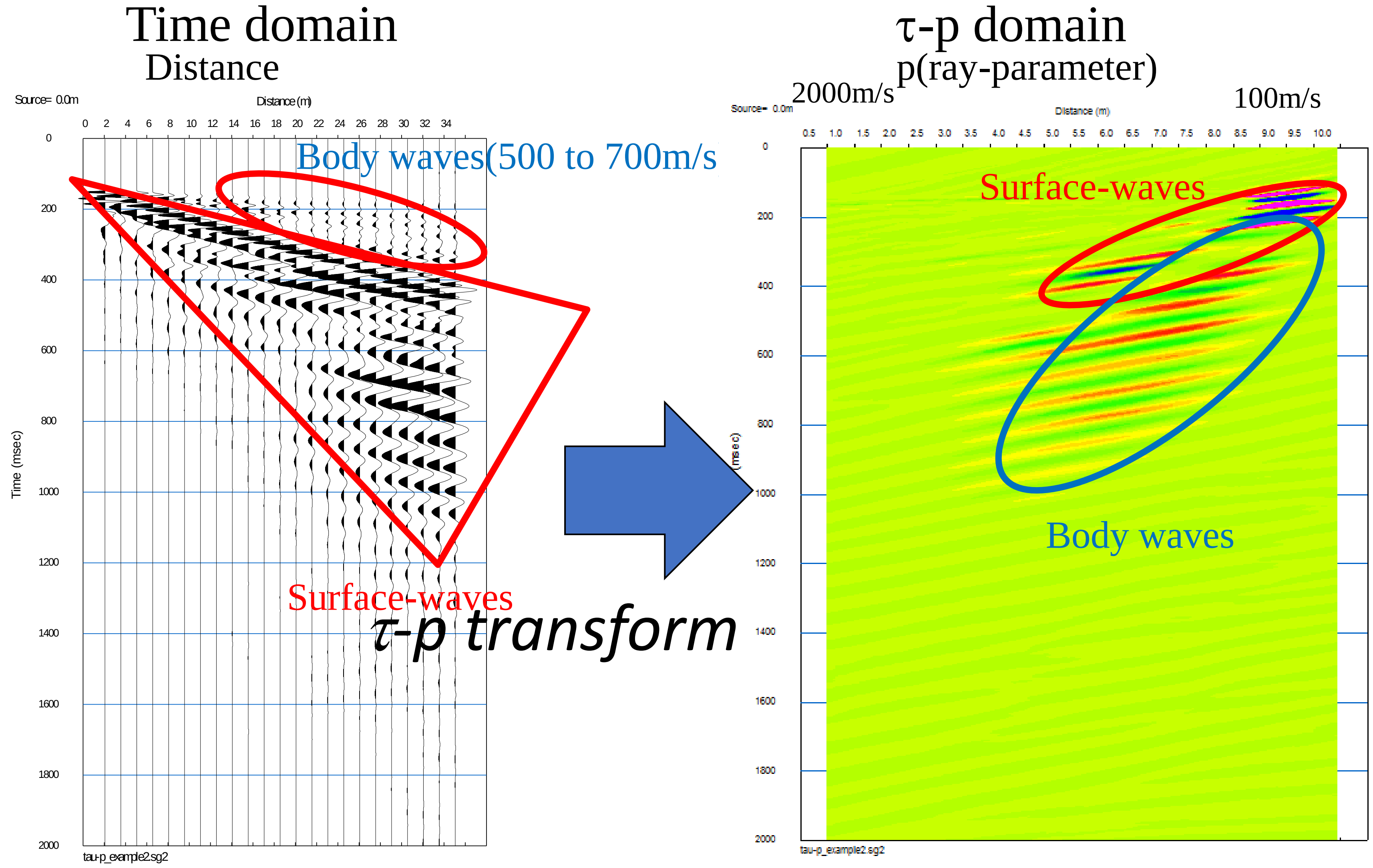


τ-p transform for non-dispersive data



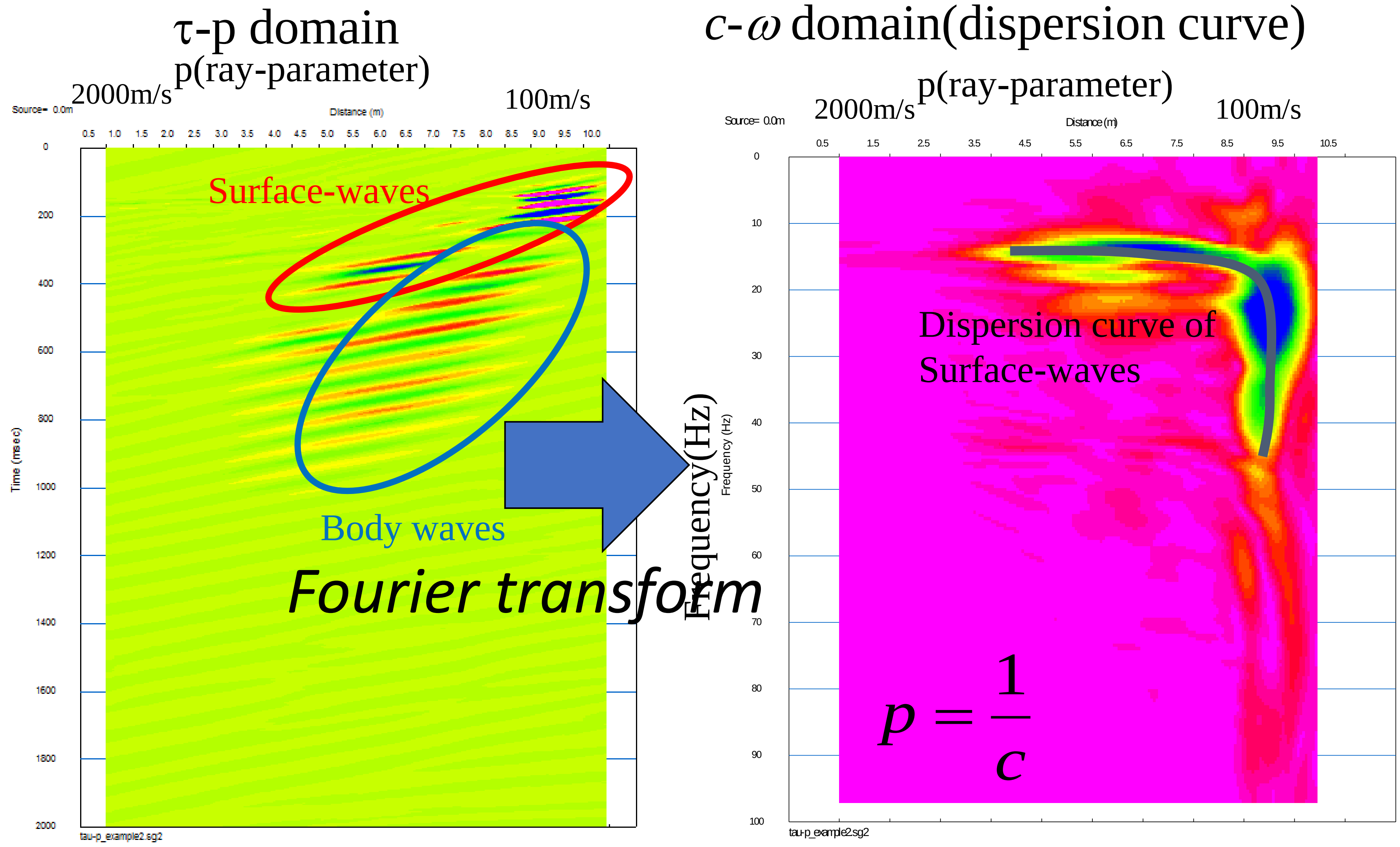
τ-p transform for dispersive data

Time-distance to τ*-p*



τ-p transform for dispersive data

τ-p to c-ω(dispersion curve)



$$p = \frac{1}{c}$$

4. τ-p transform in frequency domain (phase shift and stack : MASW)

Observed waveform

$$F(\omega)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} f(t)\cdot e^{-i\omega t}\,dt$$

$$f(x,t) \longrightarrow F(x,\omega)$$

Fourier transform

$$F(\tau,p)=\int_{-\infty}^{+\infty} f(x,t+xp)\,dx$$

τ-p transform
(Slant stack)

$$F(c,\omega)=\int_{-\infty}^{+\infty} F(x,\omega)\cdot e^{i\omega\frac{x}{c}}\,dx$$

Phase shift and stack

$$F\left(\omega,\frac{1}{c}\right)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\tau,p)\cdot e^{-i\omega\tau}\,d\tau$$

$$F(\tau,p) \longrightarrow F(c,\omega)$$

Fourier transform

Phase-velocity

4.τ-p transform in frequency domain *(MASW)*

$$f(x,t)$$

Fourier transform

$$F(\omega)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} f(t)\cdot e^{-i\omega t}dt$$

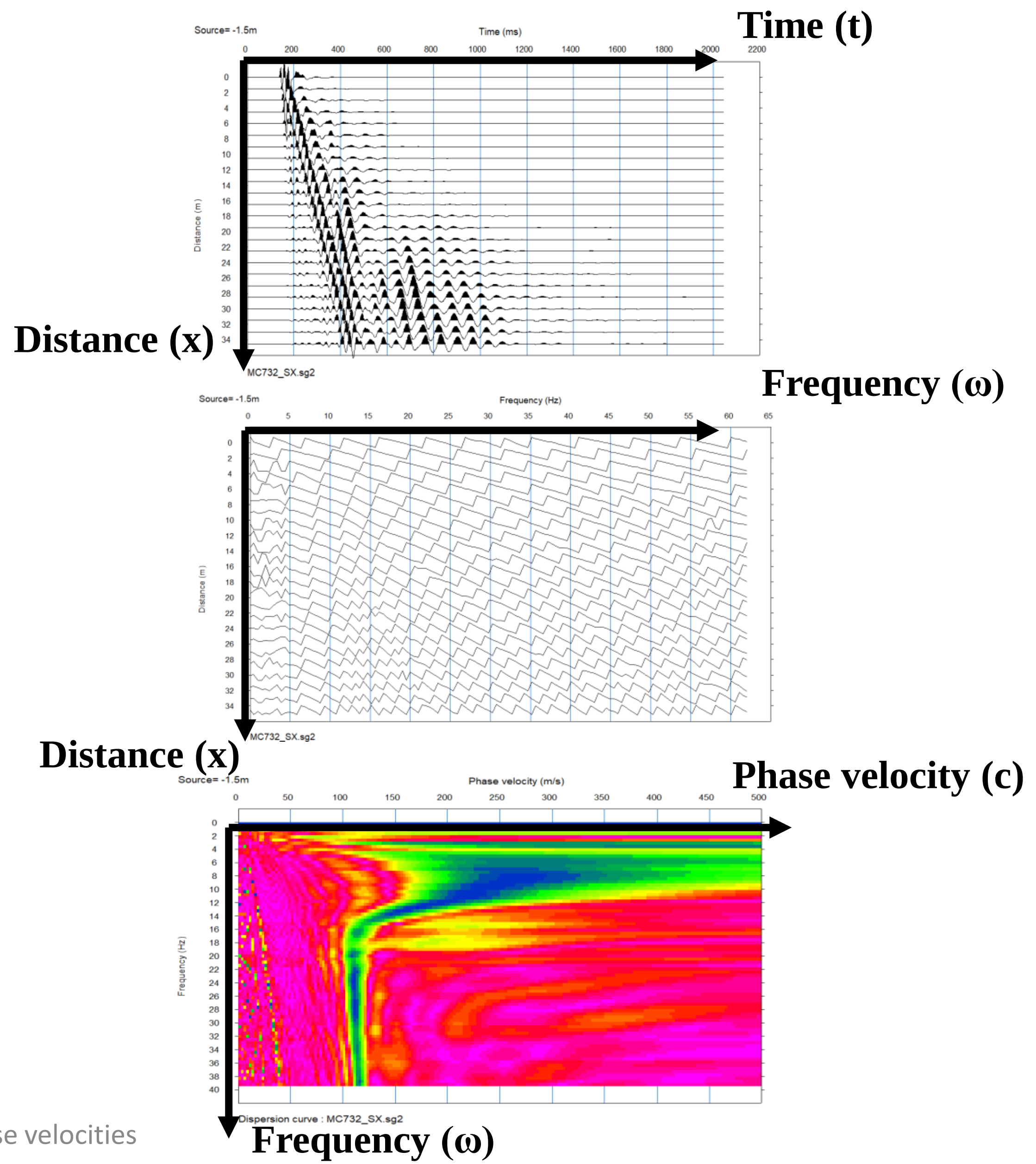
$$F(x,\omega)$$

Phase shift+Stack

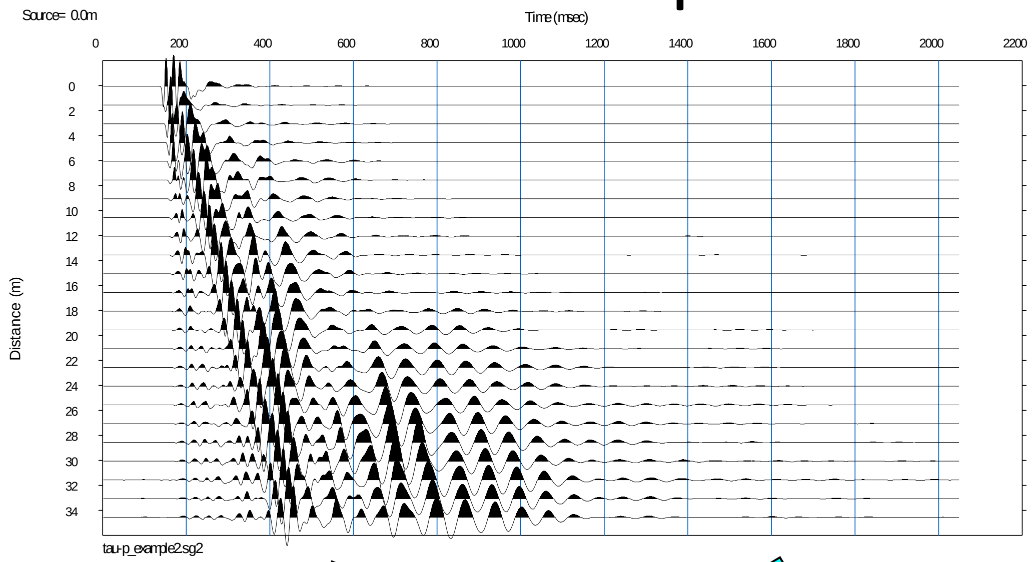
$$F(c,\omega)=\int_{-\infty}^{+\infty} F(x,\omega)\cdot e^{i\omega\frac{x}{c}}dx$$

$$F(c,\omega)$$

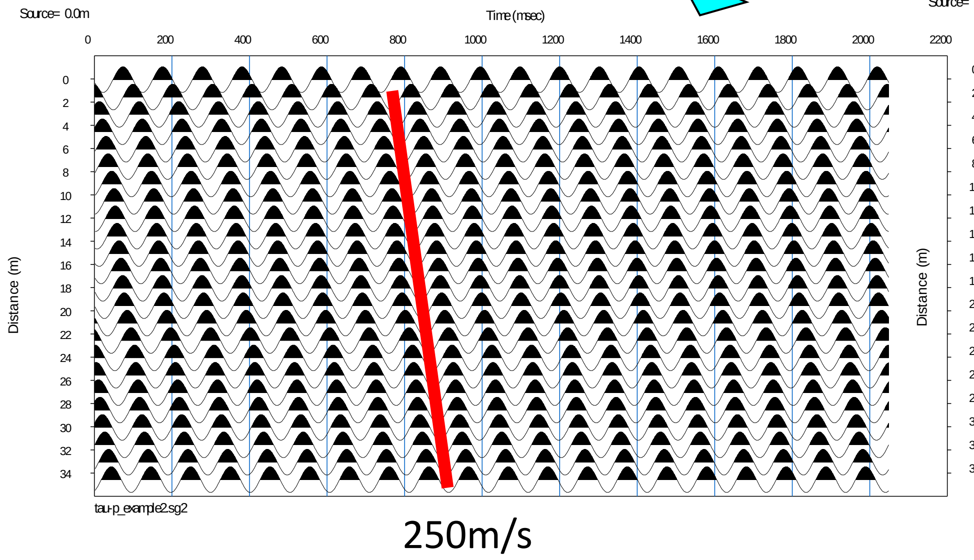
Park et al. (1999a)



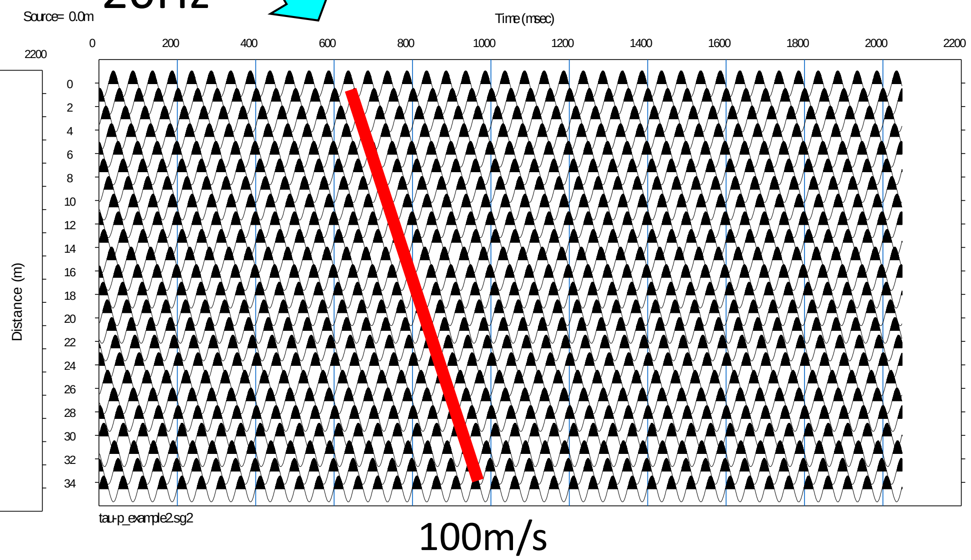
MASW (Fourier decomposition)



10Hz



20Hz

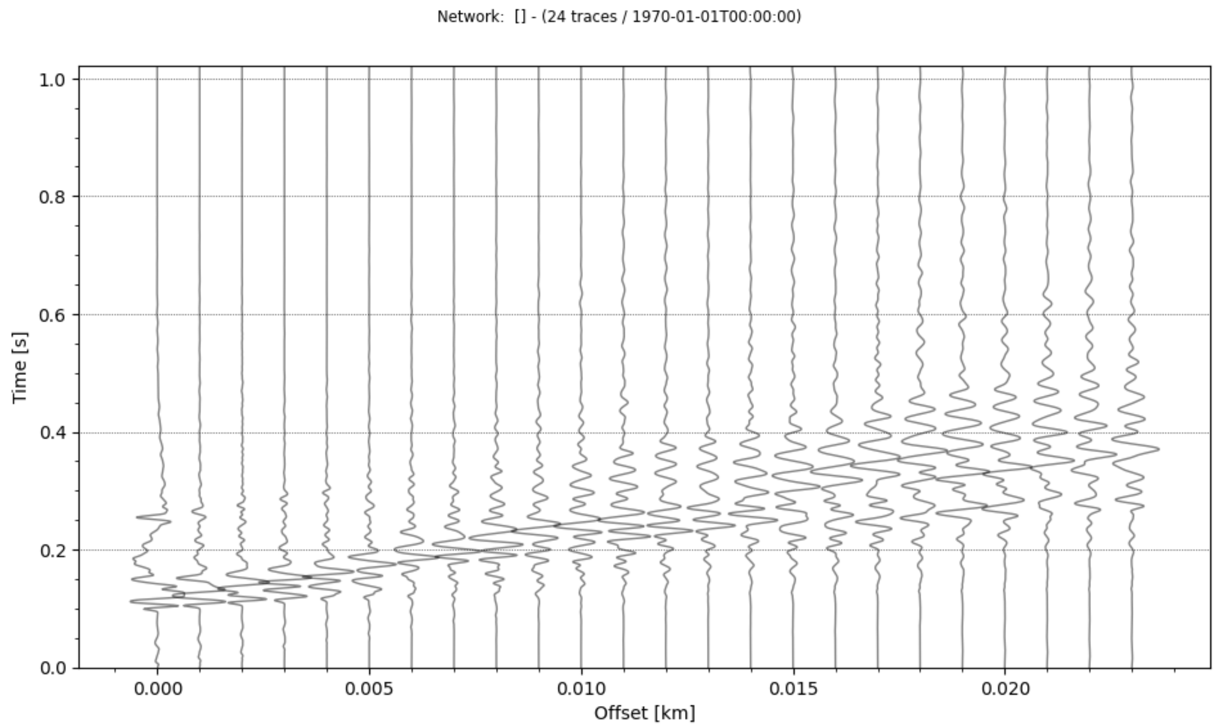


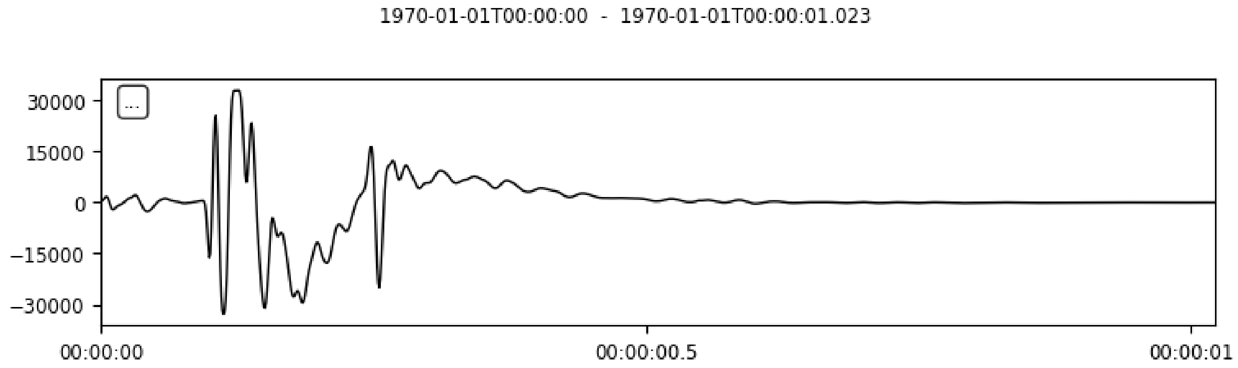
Read a SEG2 file using ObsPy

MASW.ipynb

```
pip install obspy

import obspy.io.seg2.seg2
import os

_colab_dir = "/content/drive/MyDrive/python"
os.chdir(_colab_dir)

obspy.io.seg2.seg2._is_seg2("sxdx2681.sg2")
st=obspy.io.seg2.seg2._read_seg2("sxdx2681.sg2")

i=0
for tr in st:
  tr.stats.distance=i
  i+=3

st.plot(type='section')

tr=st[0]
tr.plot()
tr.data
```

Results





array([435., 493., 800., ..., -103., -104., -105.], dtype=float32)

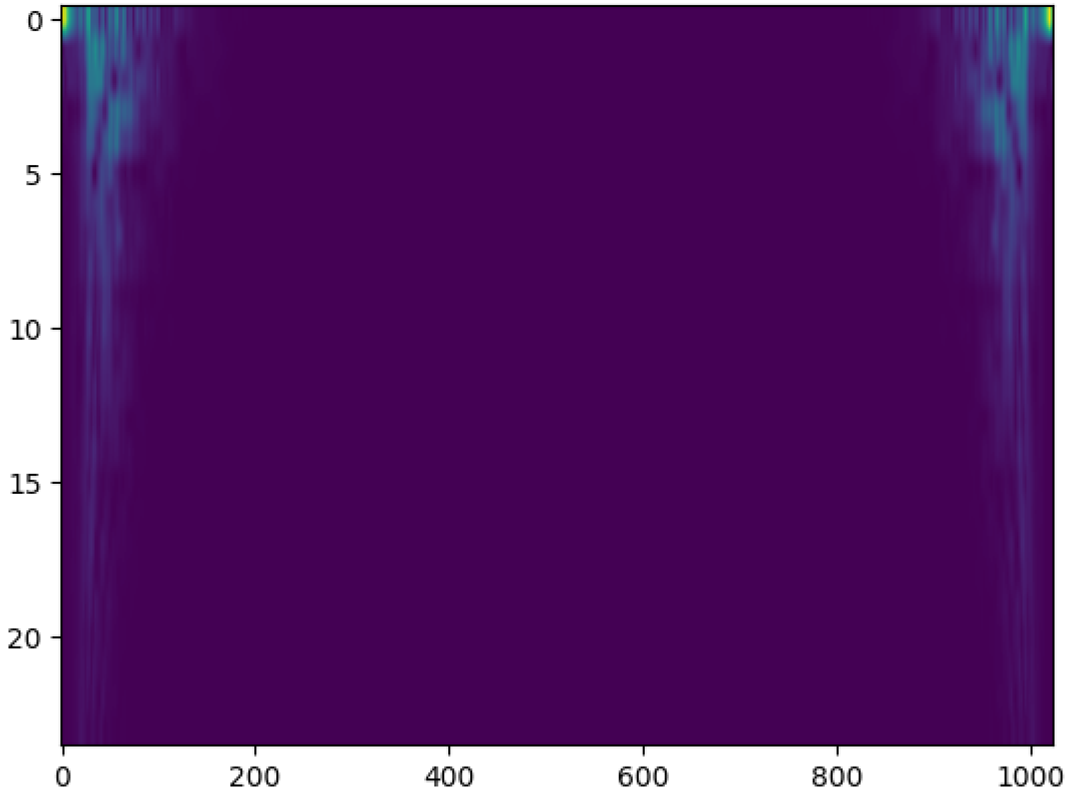
Convert to distance-frequency domain

```
import numpy as np
import matplotlib.pyplot as plt

spl=[] # complex number

for tr in st:
  # convert to frequency domain
  spl.append(np.fft.fft(tr.data))

print(spl[0])

dl=tr.stats.endtime-tr.stats.starttime # data length (sec)
df=1/dl # frequency sampling (Hz)

spa=np.array(spl)

plt.imshow(abs(spa),aspect='auto')
```

Results

```
[ 13490. +0.j -931176.54037123 +66977.21680847j
397914.86873977+1392909.39828941j ...
1514227.04638442 +390369.4001934j
397914.86873977-1392909.39828941j -
931176.54037123 -66977.21680847j]
```

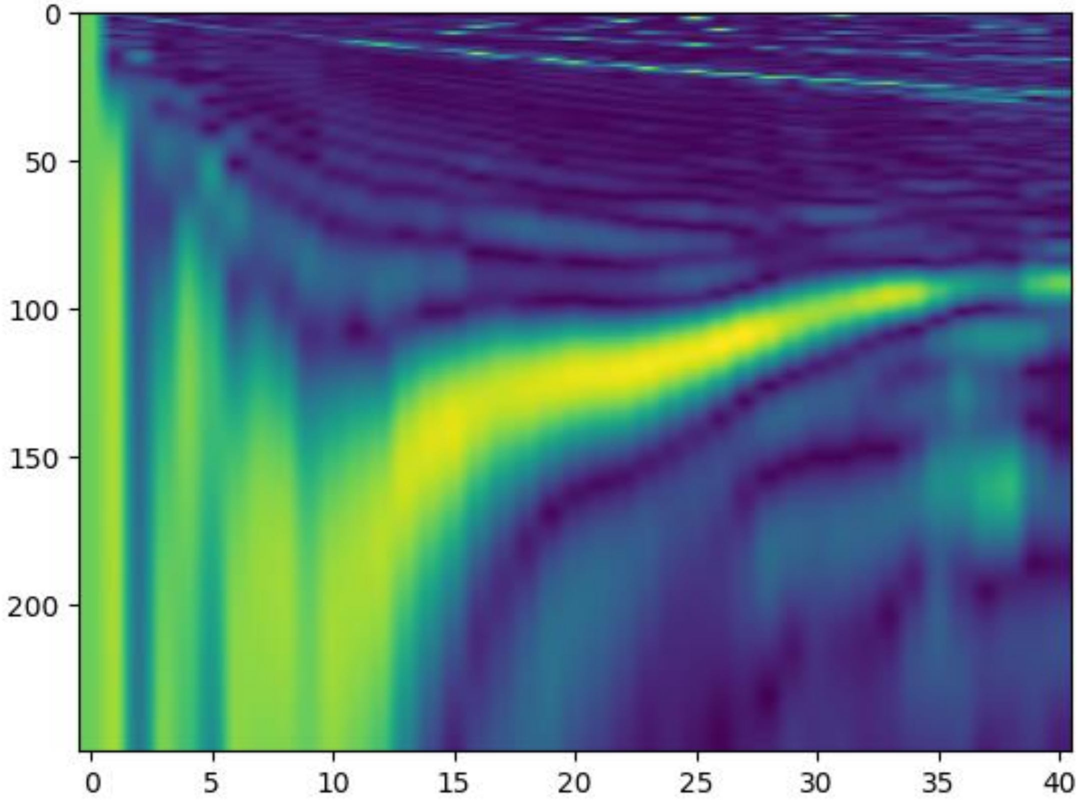
Results



Convert to phase velocity-frequency domain

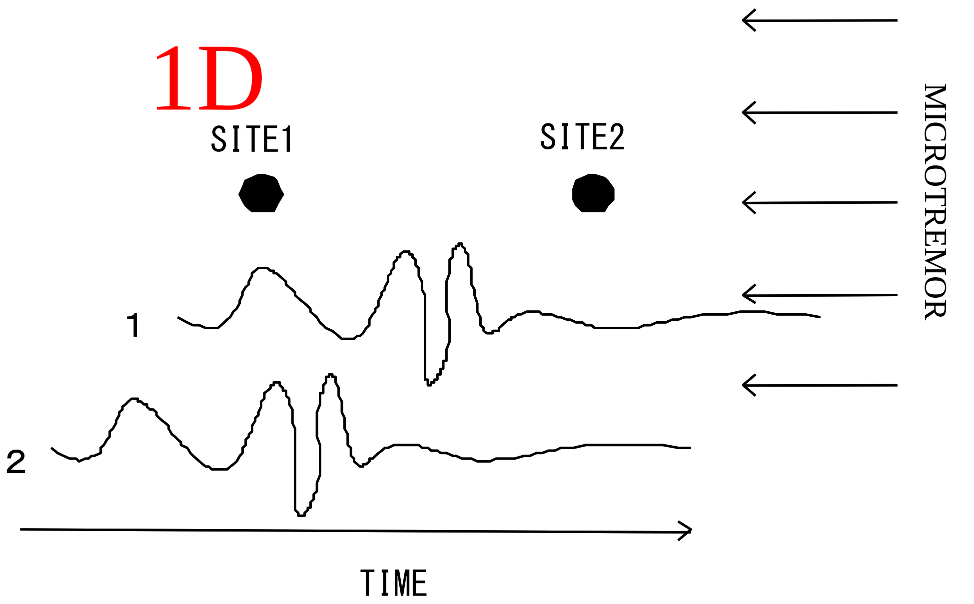
```
import math
import cmath

nf=41       # frequency
nc=250      # phase velocity
dc=1        # phase velocity sampling (m/sec)
nt=len(spl) # number of traces
pvi=[[0] * nf for i in range(nc)]

for i in range(nc):     # phase velocity loop
  cv=dc*i               # phase velocity (m/sec), real scalar
  cv=max(0.1,cv)
  for j in range(nf):   # frequency loop
    all=0j              # complex number
    f=df*j              # frequency (Hz)
    for k in range(nt): # trace loop
      d=abs(dist[k]-dist[0])/cv;  # time shift (sec)
      amp=abs(spl[k][j])          # absolute amplitude (complex to real number)
      a=spl[k][j]/amp             # normalized by amplitude (complex/real)
      w=complex(0,f*math.pi*2*d)  # phase shift (complex) calculated from time shift
      w=cmath.exp(w)              # complex number
      a*=w              # apply phase shift to waveform data (complex)
      all+=a            # stack alltraces with time shift (complex number)
    pvi[i][j]=abs(all)  # take absolute value (complex to real number)
phase_velocity_image=np.array(pvi)
plt.imshow(phase_velocity_image,aspect='auto')
```

Results

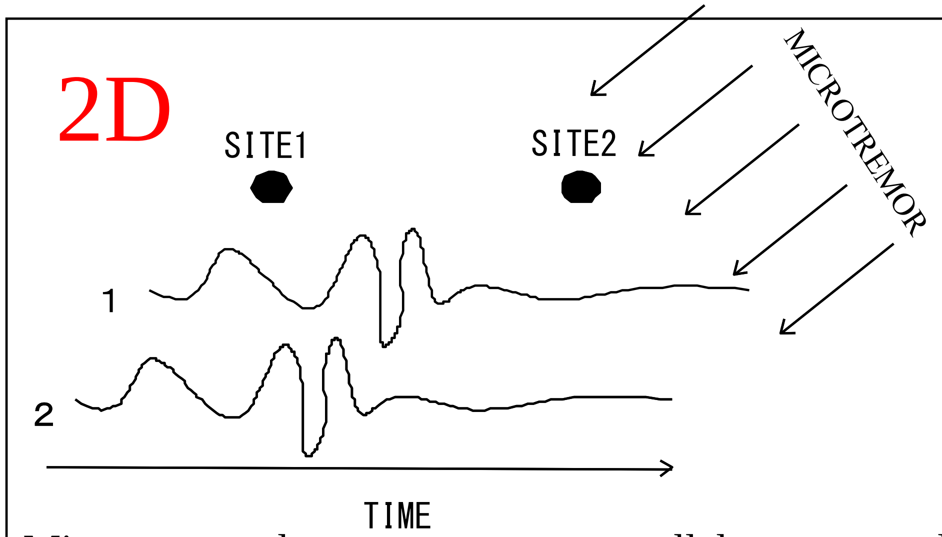


5.Spatial Auto-correlation (SPAC)

Calculating phase velocity from micro-tremors (passive method)

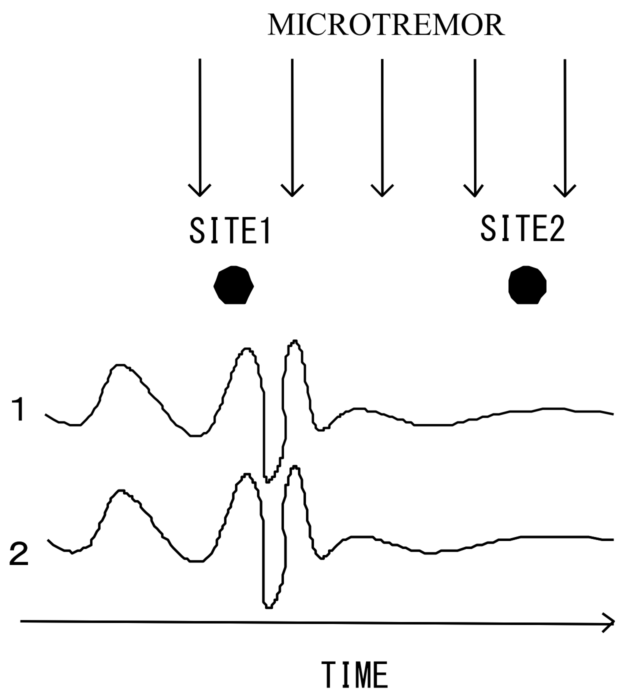


Micro-tremor propagates parallel to survey line.
Phase velocity can be calculated directly.



Micro-tremor does not propagate parallel to a survey line.
Propagation direction is known
-> phase velocity can be calculated.
Propagation direction is unknown
-> apparent velocity is higher than true phase velocity

Spatial auto correlation (2D)

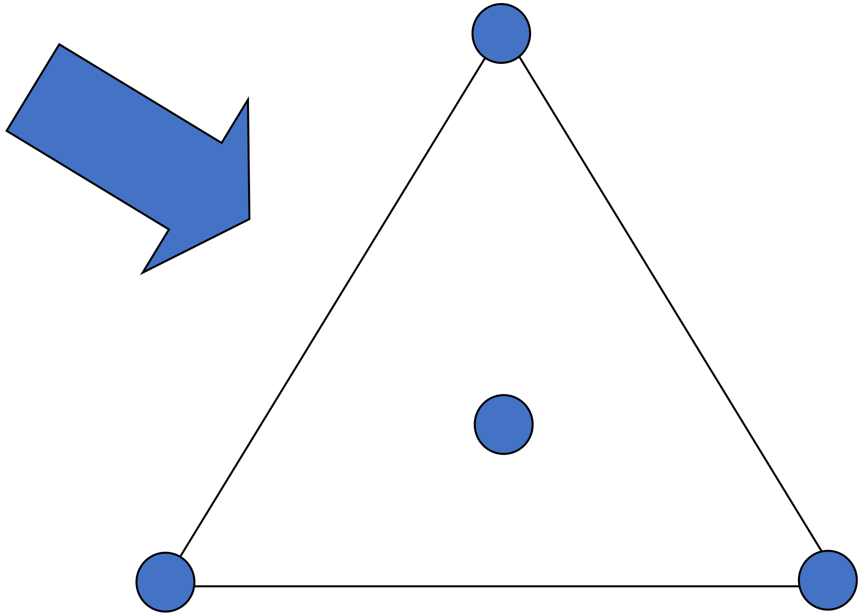


Microtremors propagate perpendicular to survey line.
Phase velocity can't be calculated.

We do not know the direction of microtremors propagation before measuring.

And the sources of microtremors is generally unstable.

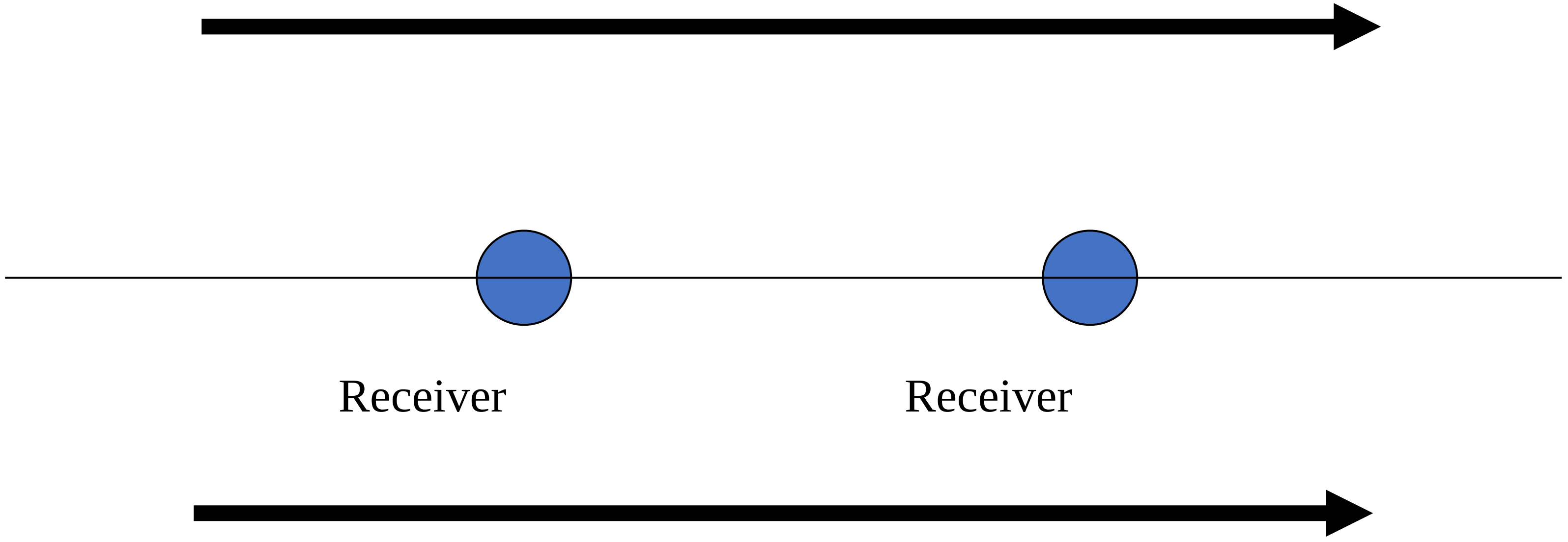
Microtremors random and stochastic phenomena.



Isotropic sensor arrangement is need.
Like triangle array

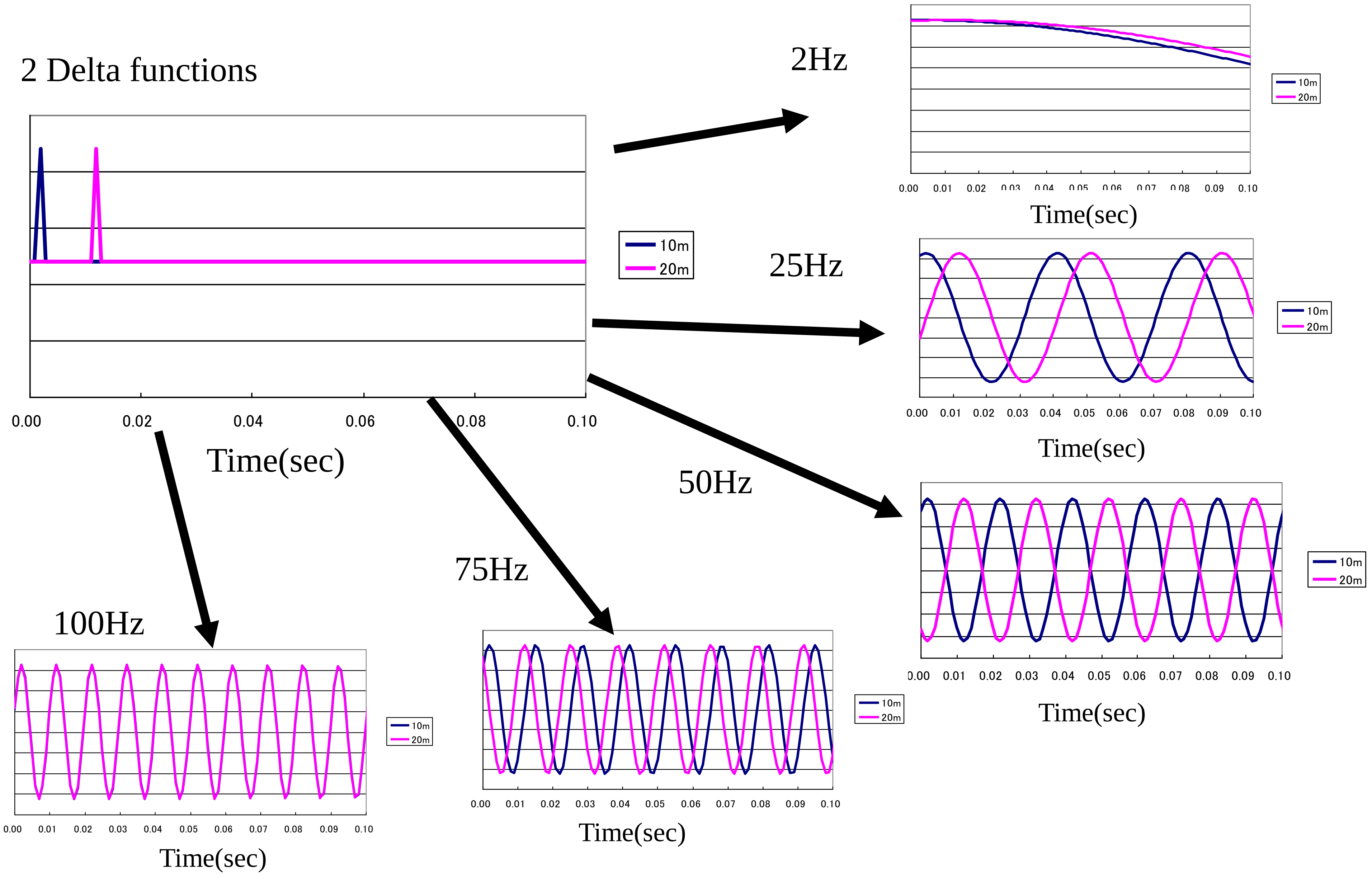
How to calculate phase velocity from ambient noise based on Spatial Auto-correlation (SPAC) ?

1D wave propagation

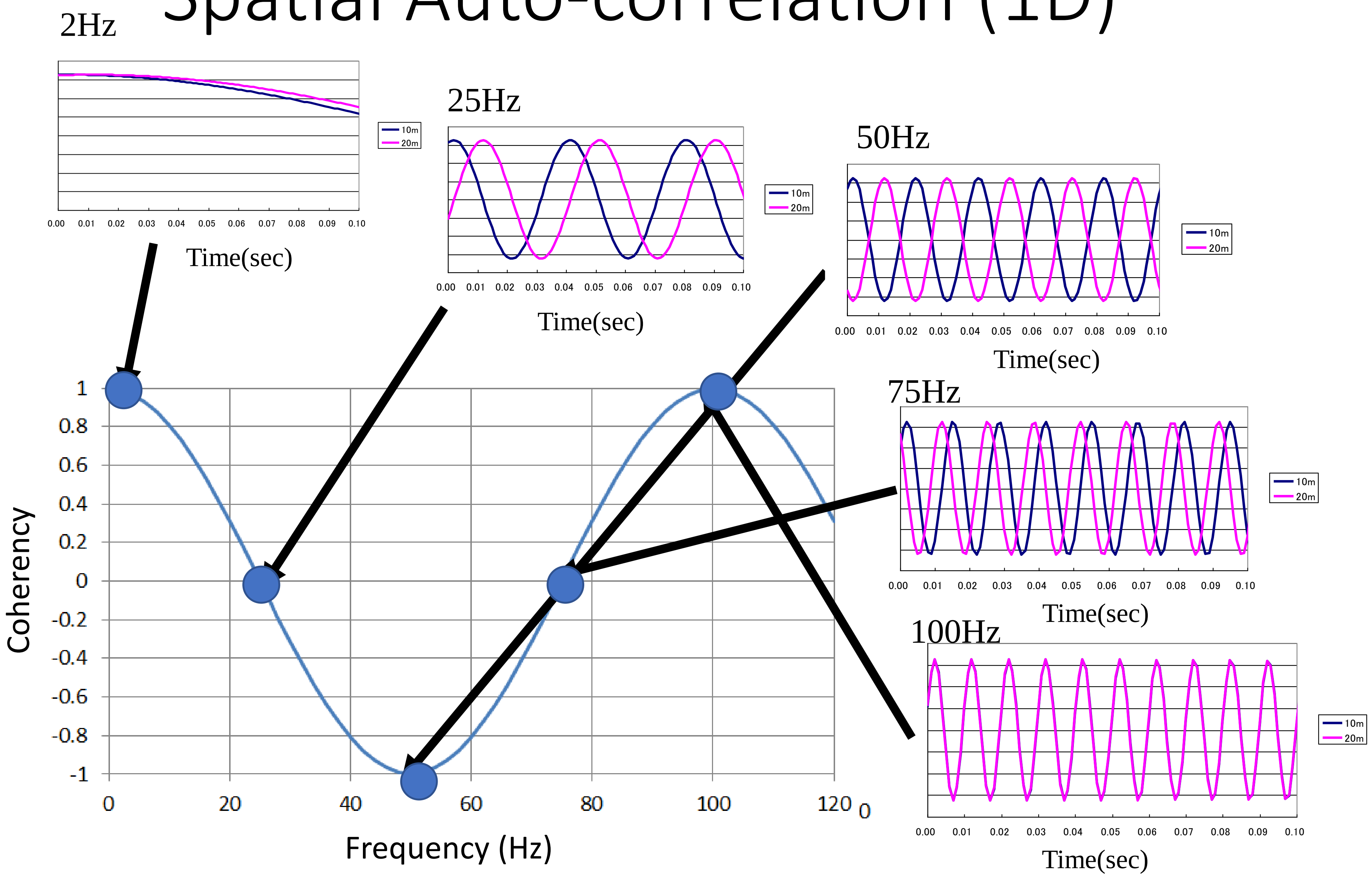


Waves propagating parallel to the receiver array

Spatial Auto-correlation (1D)



Spatial Auto-correlation (1D)



Spatial Auto-correlation (1D)

Aki (1957)

Cross-correlation

$$CC_{fg}(\omega) = F(\omega)\cdot\overline{G(\omega)} = A_f(\omega)A_g(\omega)\cdot\exp^{i\Delta\phi(\omega)}$$

Phase velocity $(c(\omega))$

Substitute

$$c(\omega) = \frac{\omega\cdot\Delta x}{\Delta\phi(\omega)} \quad\longrightarrow\quad \Delta\phi(\omega) = \frac{\omega\cdot\Delta x}{c(\omega)}$$

$$CC_{fg}(\omega) = F(\omega)\cdot\overline{G(\omega)} = A_f(\omega)A_g(\omega)\cdot\exp^{i\frac{\omega\cdot\Delta x}{c(\omega)}}$$

Take real part

$$\mathrm{Re}(CC_{fg}(\omega)) = A_f(\omega)A_g(\omega)\cos\left(\frac{\omega\cdot\Delta x}{c(\omega)}\right)$$

Coherence

$$COH_{fg}(\omega) = \frac{\mathrm{Re}(CC_{fg}(\omega))}{A_f(\omega)A_g(\omega)} = \cos\left(\frac{\omega\cdot\Delta x}{c(\omega)}\right)$$

Spatial auto correlation (1D)

Spatial auto correlation

Time domain

$$cc(\Delta x, t) = f(x,t) * \overline{f(x+\Delta x, t)}$$

Two traces with Δx separation

Frequency domain

$$CC(\Delta x, \omega) = F(x,\omega) \cdot \overline{G(x+\Delta x, \omega)}$$

Coherence

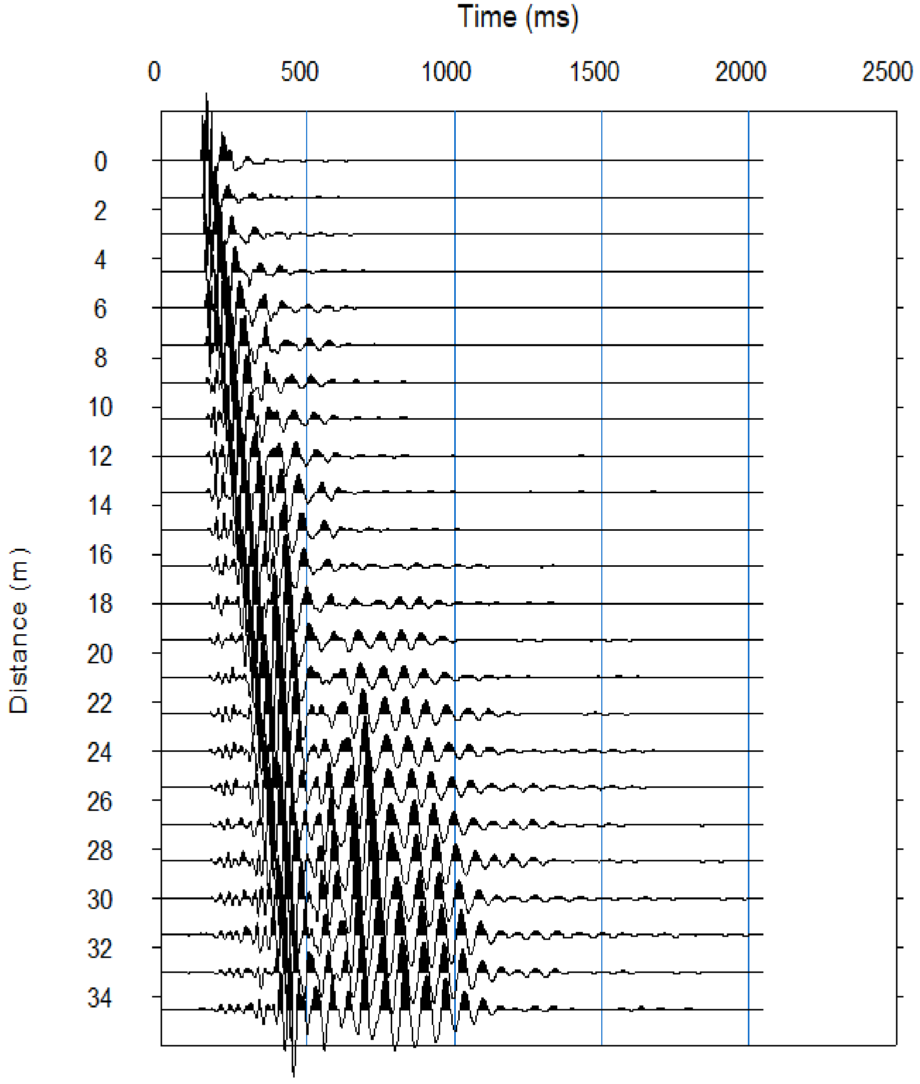
$$COH(\Delta x, \omega) = \frac{\mathrm{Re}(CC(\Delta x, \omega))}{AC(x,\omega) AC(x+\Delta x, \omega)}$$

$$COH(\Delta x, \omega) = \boxed{\cos\left(\frac{\omega}{c(\omega)} \Delta x\right)}$$

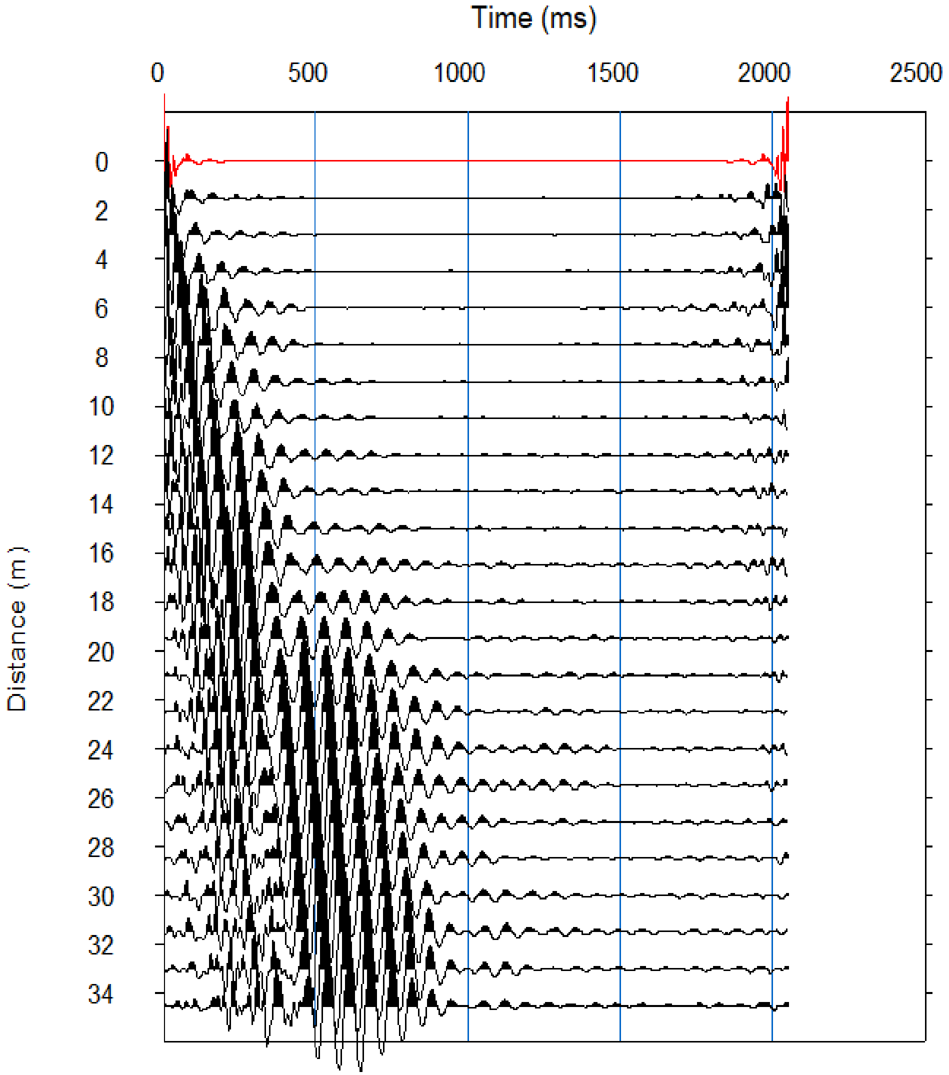
→ **Cosine function**

1D real data example

Raw data

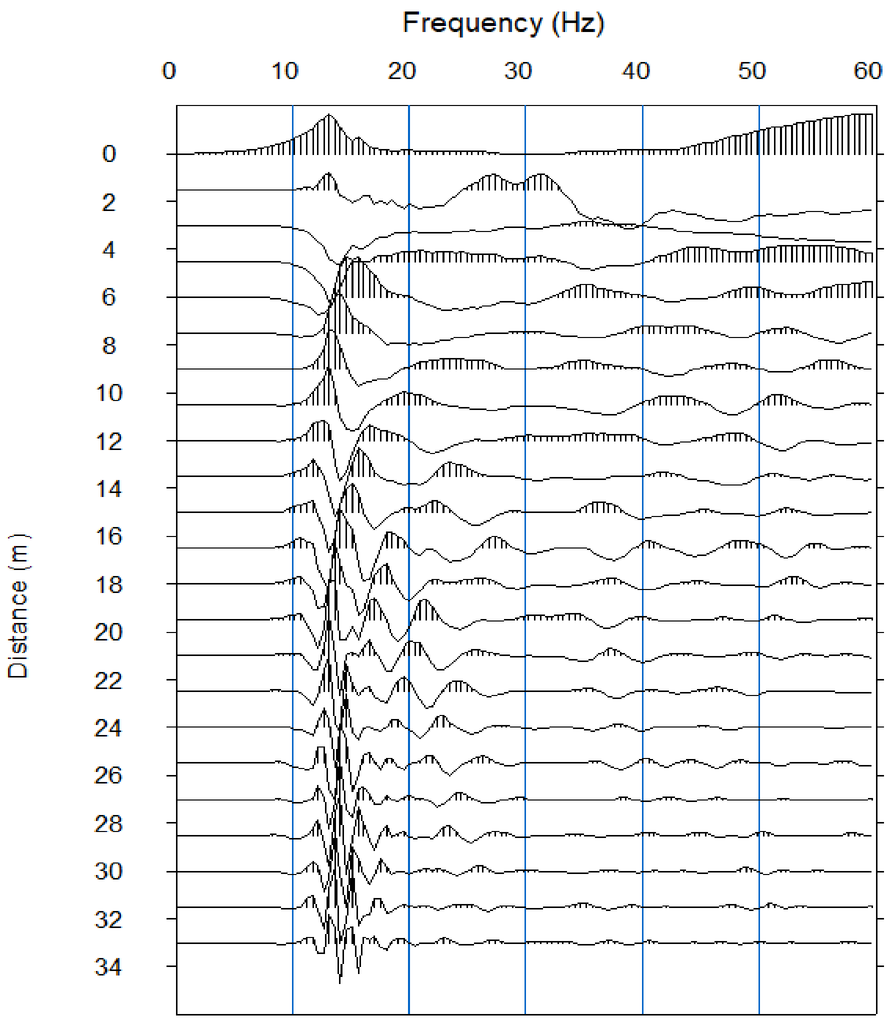


Cross-correlation

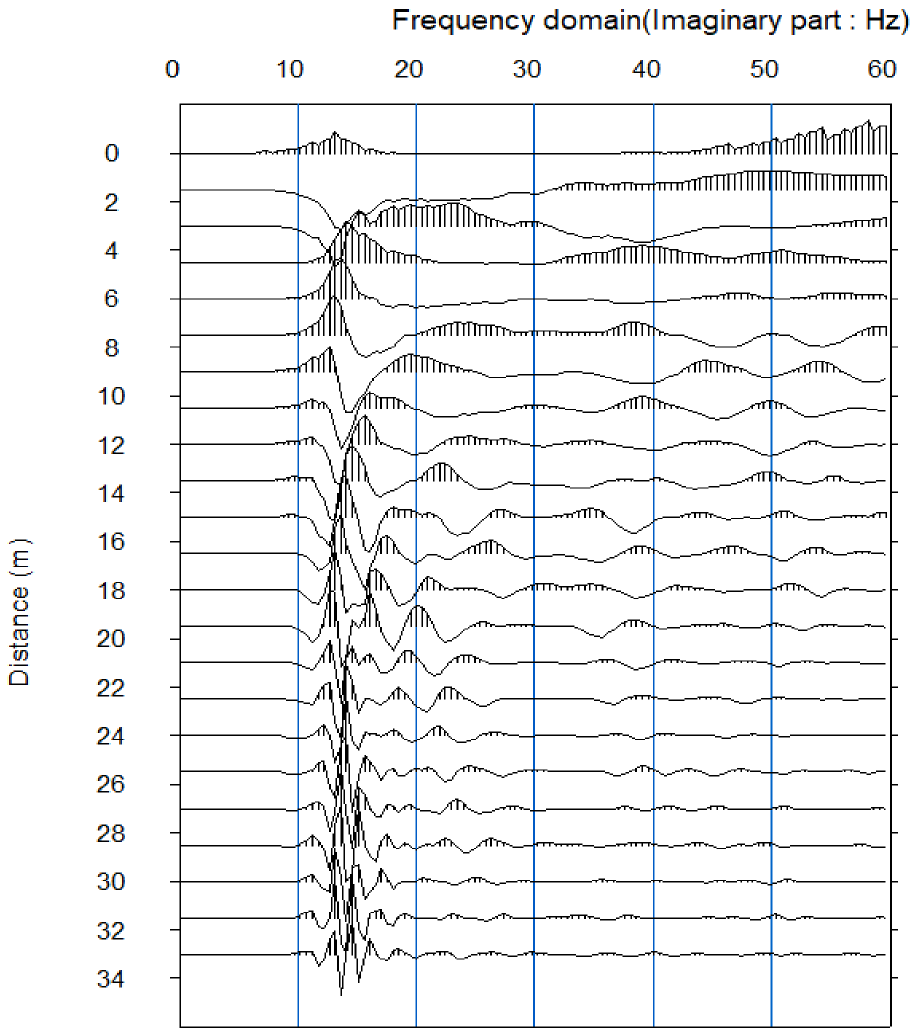


1D real data example

Real part (cos)

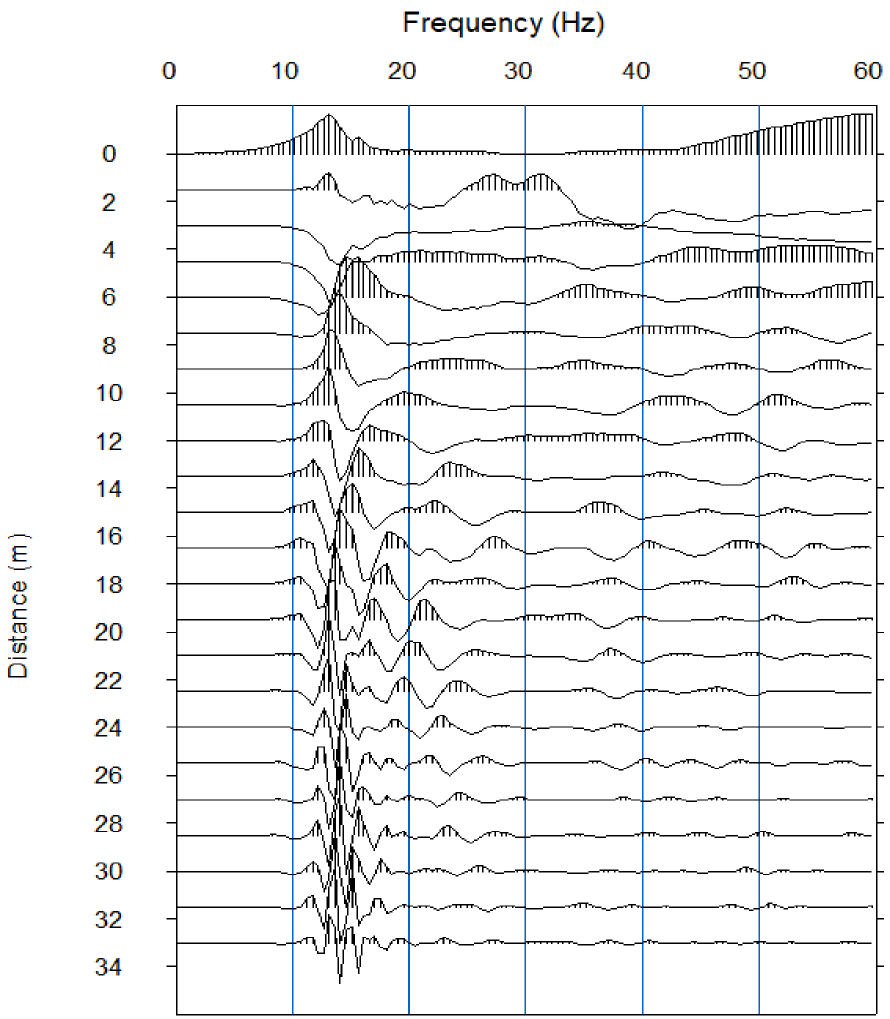


Imaginary part (sin)

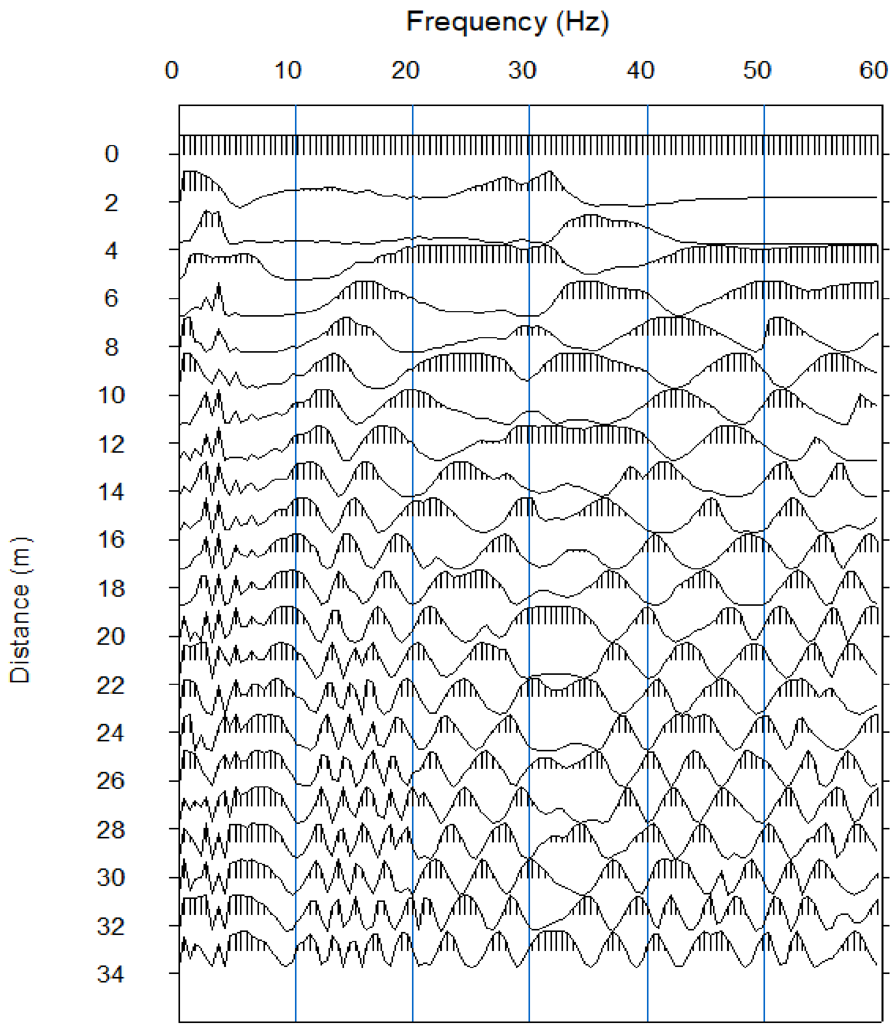


1D real data example

Real part (cos)



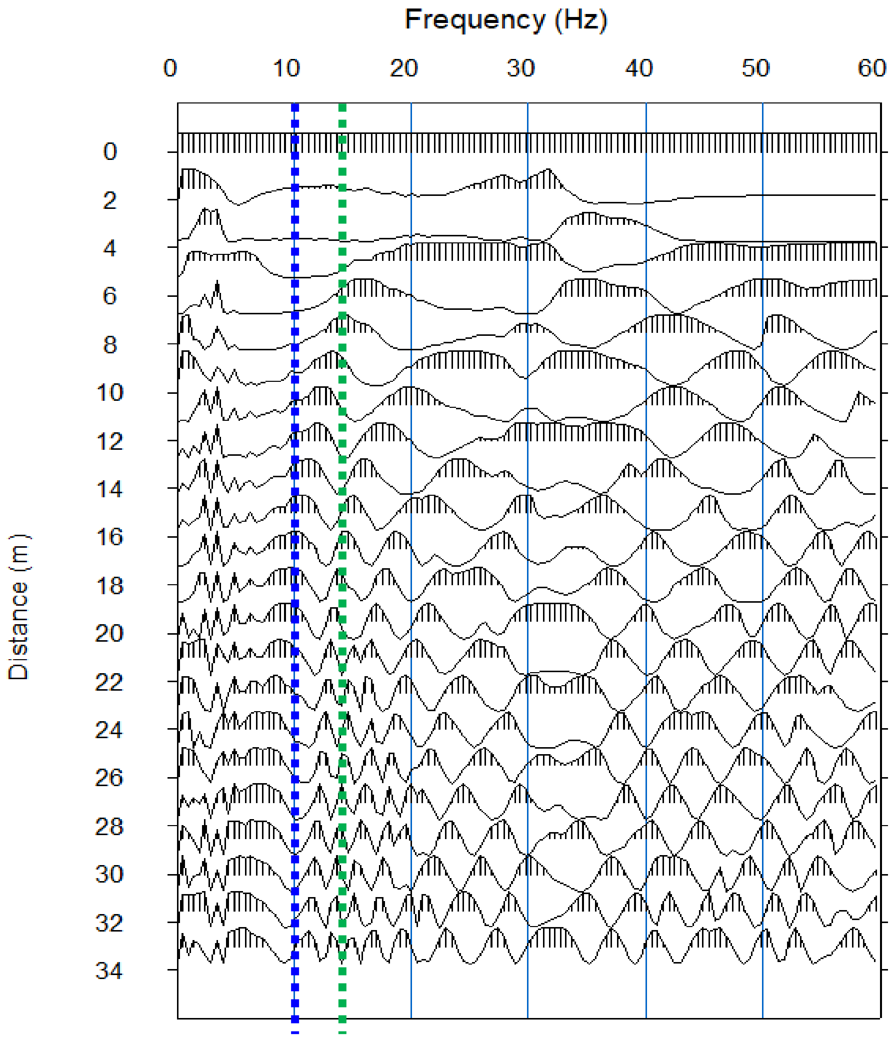
Normalized real part (cos) = Coherency

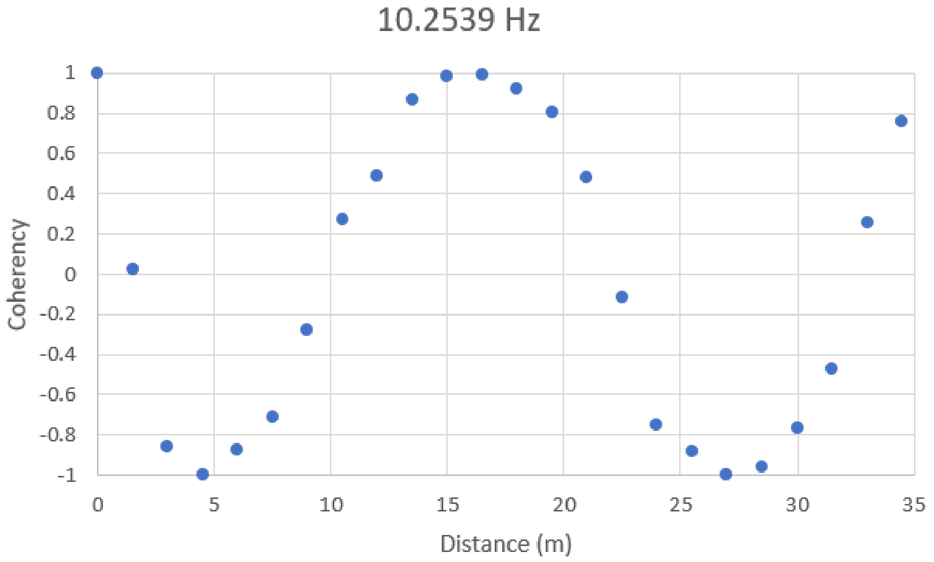


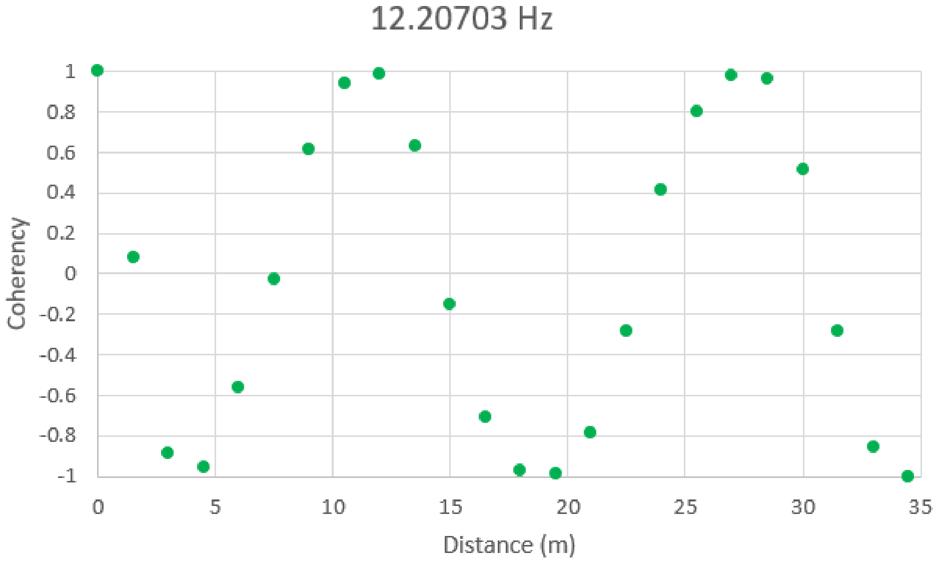
1D real data example

$$COH(\Delta x,\omega) = \cos\left(\frac{\omega}{c(\omega)}\Delta x\right)$$

Normalized real part (cos) = Coherency

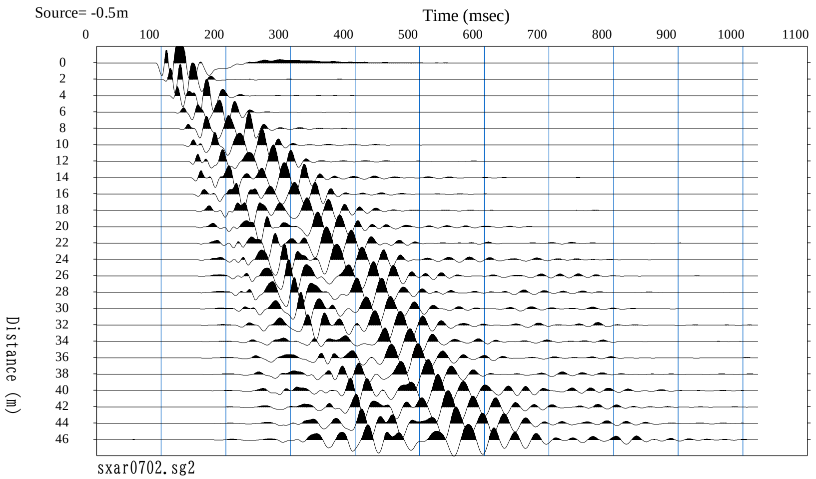




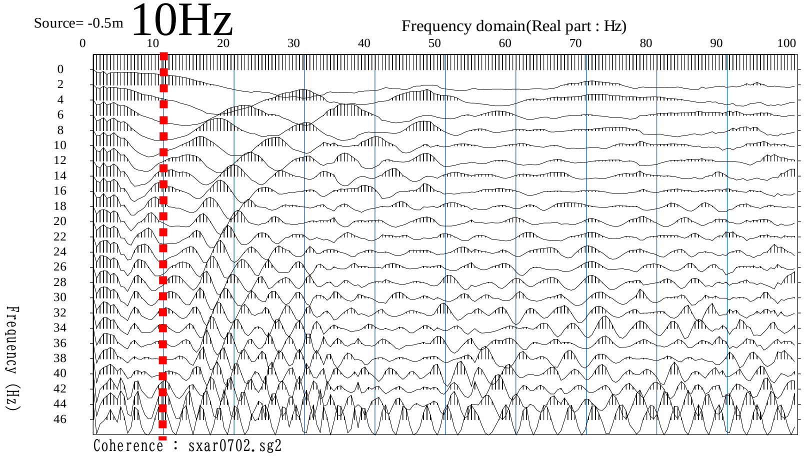


1D example

Time domain

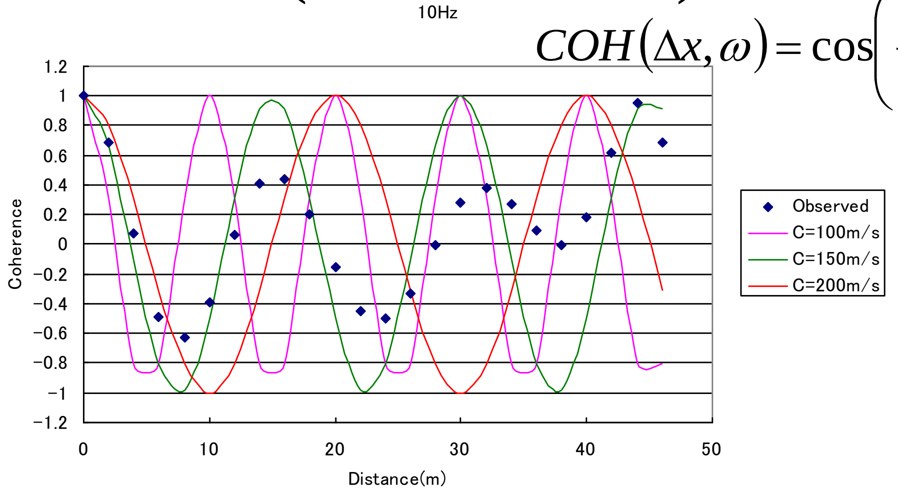


Coherence

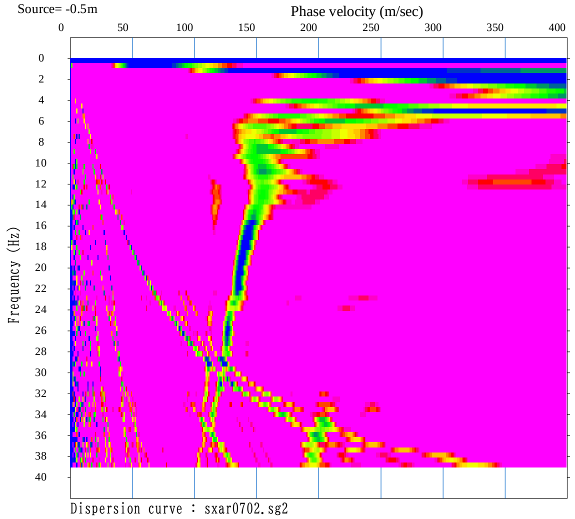


Coherence (function of Δx)

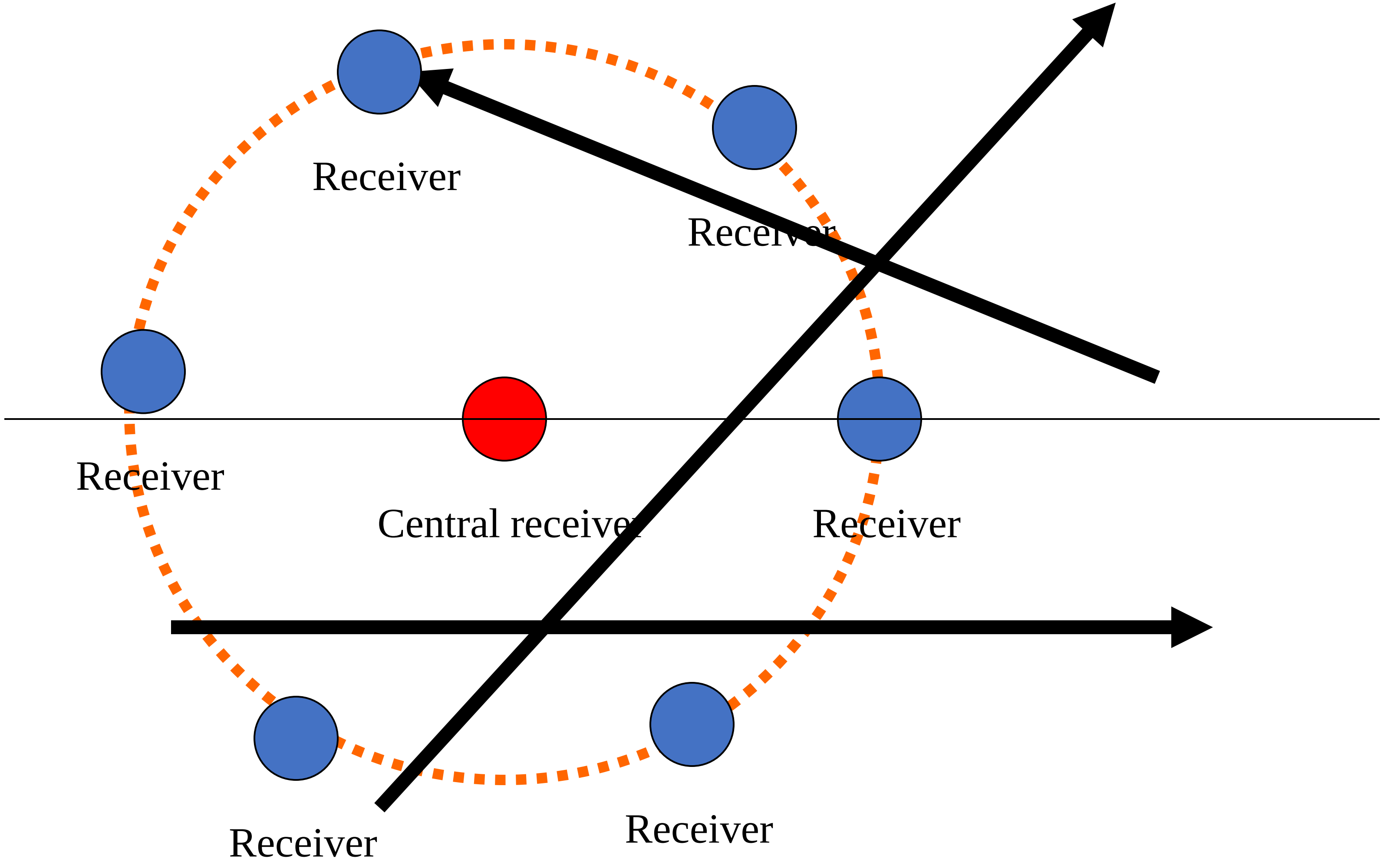
$$COH(\Delta x, \omega) = \cos\left(\frac{\omega}{c(\omega)}\Delta x\right)$$



Phase-velocity image

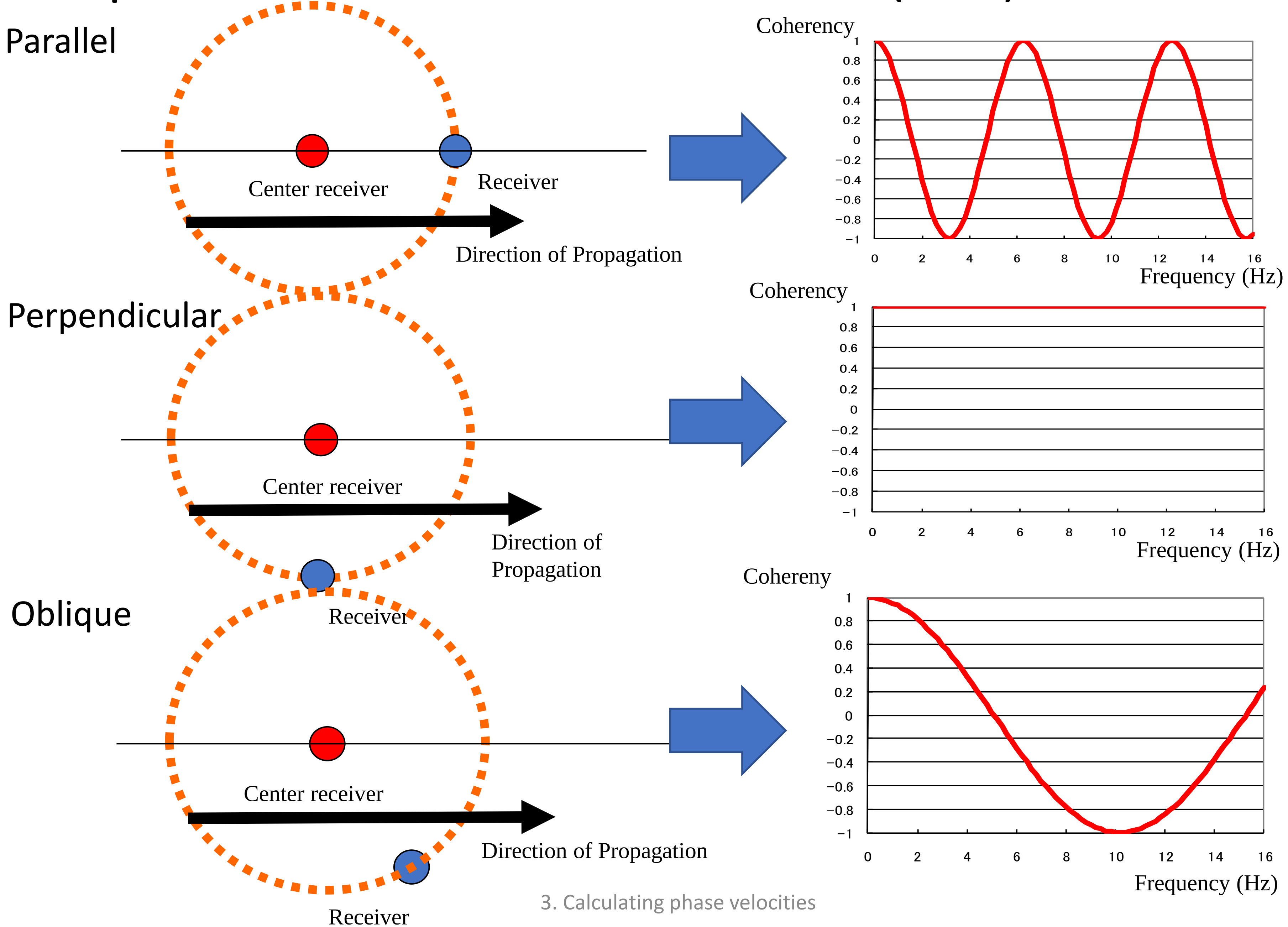


Spatial Auto-correlation (2D)

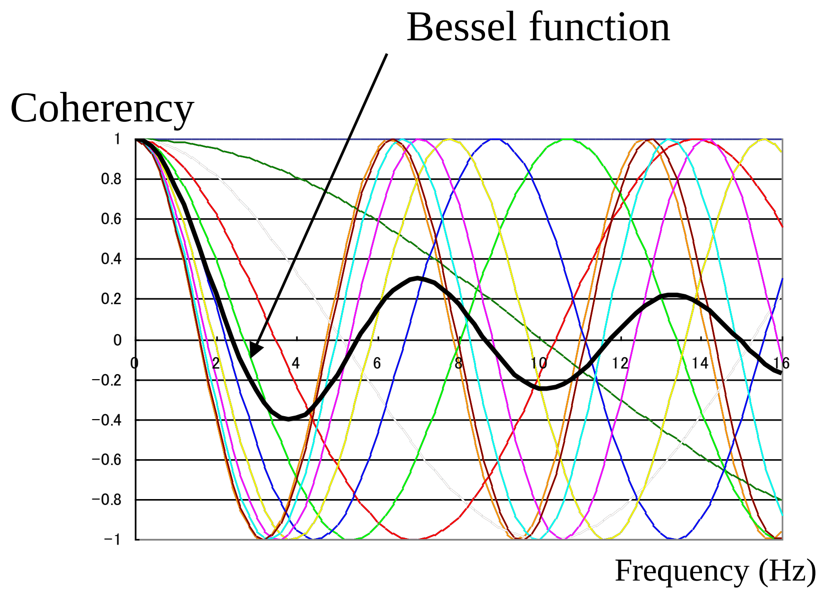


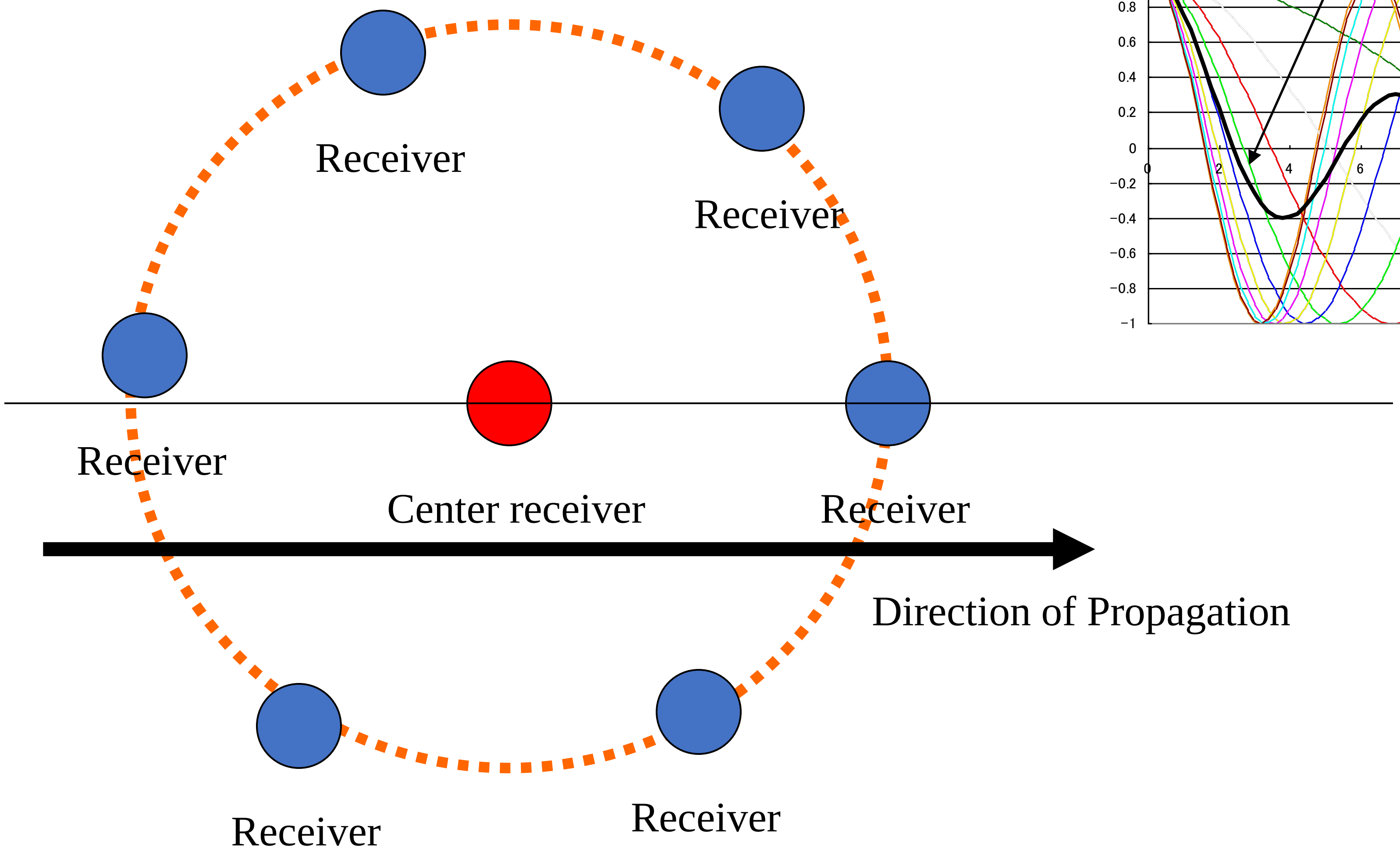
Micro-tremors does not propagate with constant direction

Spatial Auto-correlation (2D)



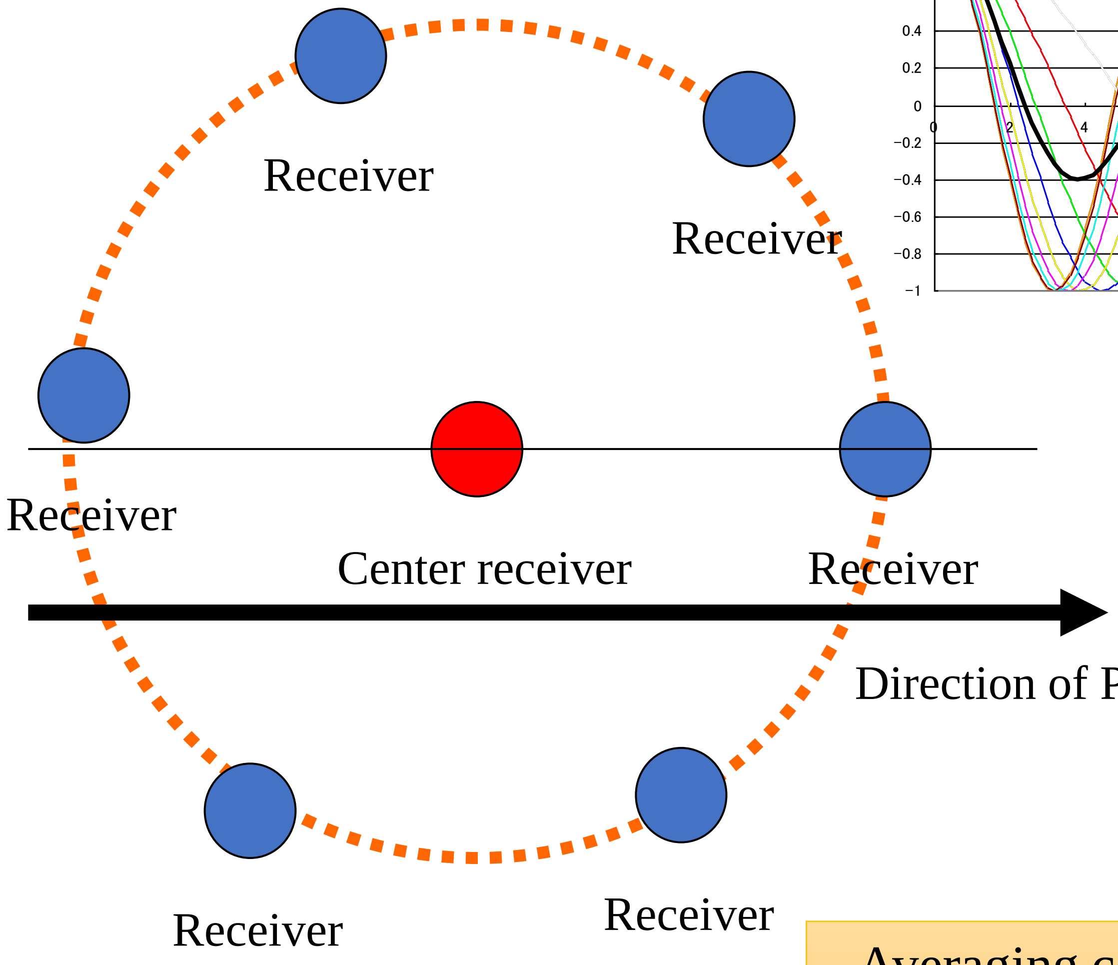
Spatial Auto-correlation (2D)

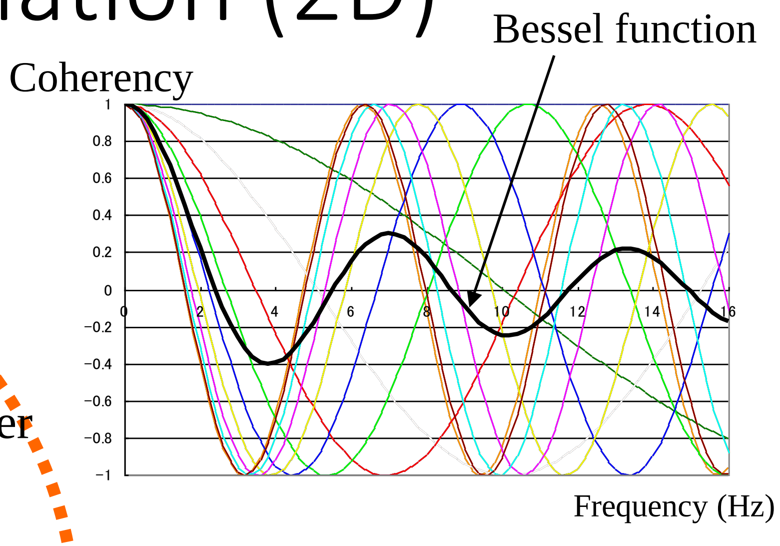


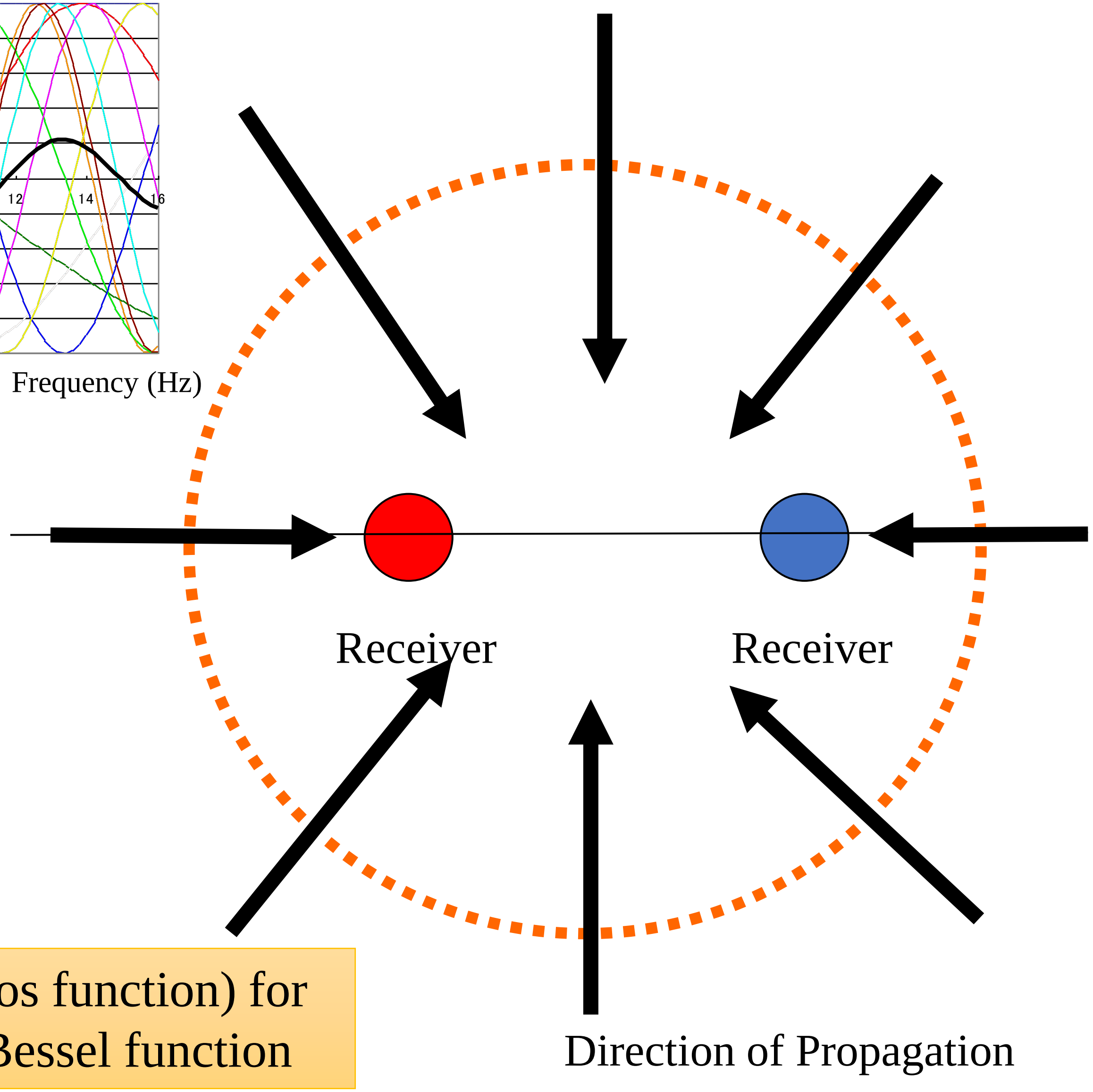


Averaging all direction coherence (cos function） results in Bessel function

Spatial Auto-correlation (2D)







Averaging coherency (cos function) for all direction results in Bessel function

Spatial Auto-correlation (2D)

Spatial auto correlation in 1D

$$\mathrm{Re}\left(CC_{fg}(\omega)\right) = A_f(\omega)A_g(\omega)\cos\left(\frac{\omega \cdot \Delta x}{c(\omega)}\right)$$

Angular averaging(Integrating over 360 degree)

$$\int_{\varphi=0}^{\varphi=2\pi} \mathrm{Re}\left(CC_{fg}(\omega)\right) d\phi = \int_{\phi=0}^{\phi=2\pi} A_f(\omega)A_g(\omega)\cos\left(\frac{\omega \cdot \Delta x}{c(\omega)}\right) d\phi$$

$$\int_{\phi=0}^{\phi=2\pi} \mathrm{Re}\left(CC_{fg}(\omega)\right) d\phi = \int_{\phi=0}^{\phi=2\pi} A_f(\omega)A_g(\omega) d\phi \cdot \int_{\phi=0}^{\phi=2\pi} \cos\left(\frac{\omega \cdot \Delta x}{c(\omega)}\right) d\phi$$

$$\int_{\phi=0}^{\phi=2\pi} \mathrm{Re}\left(CC_{fg}(\omega)\right) d\phi = \int_{\phi=0}^{\phi=2\pi} A_f(\omega)A_g(\omega) d\phi \cdot j_0\left(\frac{\omega \cdot \Delta x}{c(\omega)}\right)$$

$$\int_{\phi=0}^{\phi=2\pi} COH_{fg}(\omega) d\phi = \int_{\phi=0}^{\phi=2\pi} \frac{\mathrm{Re}\left(CC_{fg}(\omega)\right)}{A_f(\omega)A_g(\omega)} d\phi = j_0\left(\frac{\omega \cdot \Delta x}{c(\omega)}\right)$$

Spatial Auto-correlation (2D)

$$cc(\Delta x, \Delta y, t) = f(x, y, t) * \overline{f(x + \Delta x, y + \Delta y, t)}$$

$$CC(\Delta x, \Delta y, \omega) = F(x, y, \omega) \cdot \overline{G(x + \Delta x, y + \Delta y, \omega)}$$

$$\Delta x = r \cos \varphi$$
$$\Delta y = r \sin \varphi$$

$$CC(r, \omega) = F(x, y, \omega) \cdot \overline{G(x + \Delta x, y + \Delta y, \omega)}$$

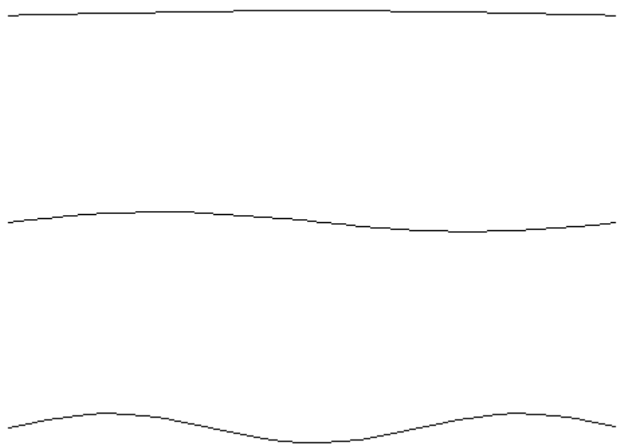
$$COH(r, \omega) = \frac{CC(r, \omega)}{AC(x, y, \omega) AC(x + \Delta x, y + \Delta y, \omega)}$$

$$\int_{\phi=0}^{\phi=2\pi} COH(r, \omega) = J_0\left(\frac{\omega}{c(\omega)} r\right)$$

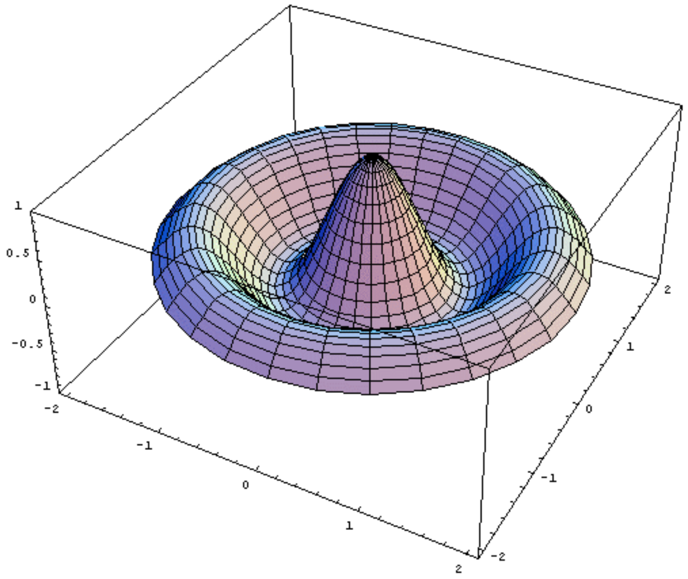
→ **Bessel function**

Trigonometric and Bessel functions

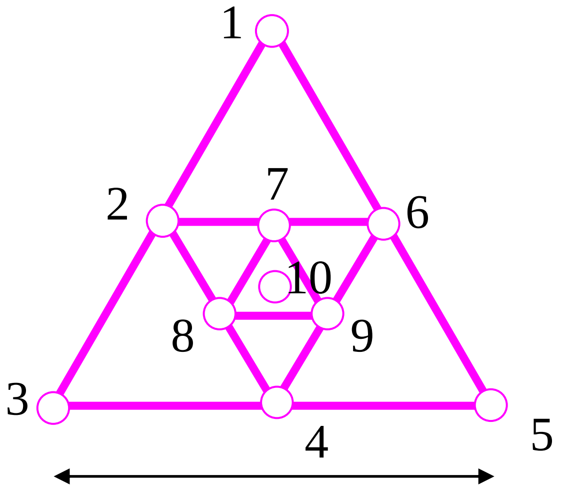
Active method
Trigonometric functions



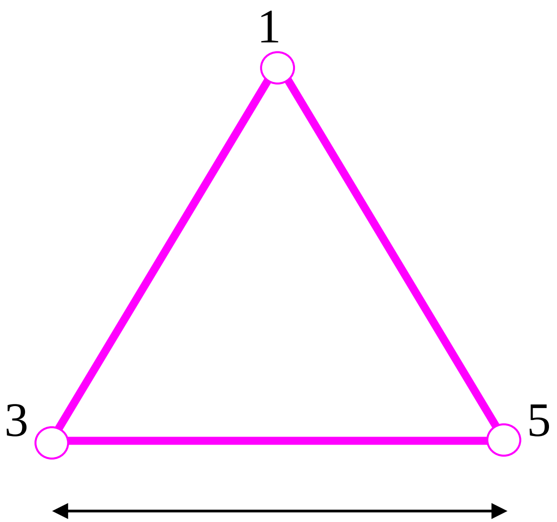
Passive method
Bessel functions



2D example

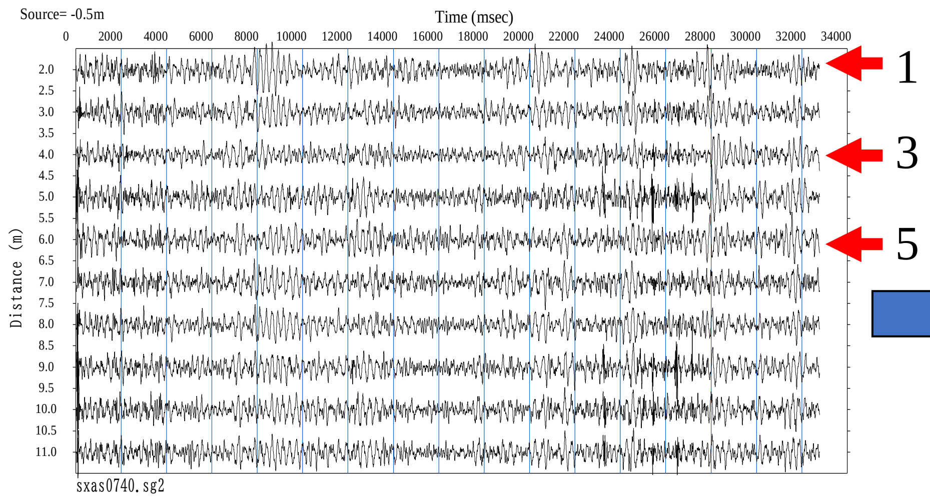


40m



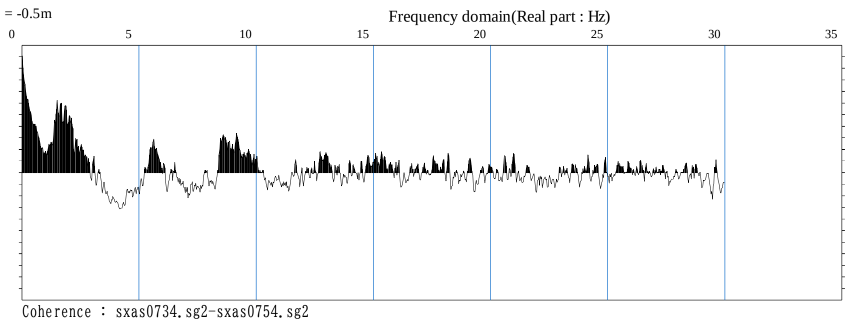
40m

Time domain



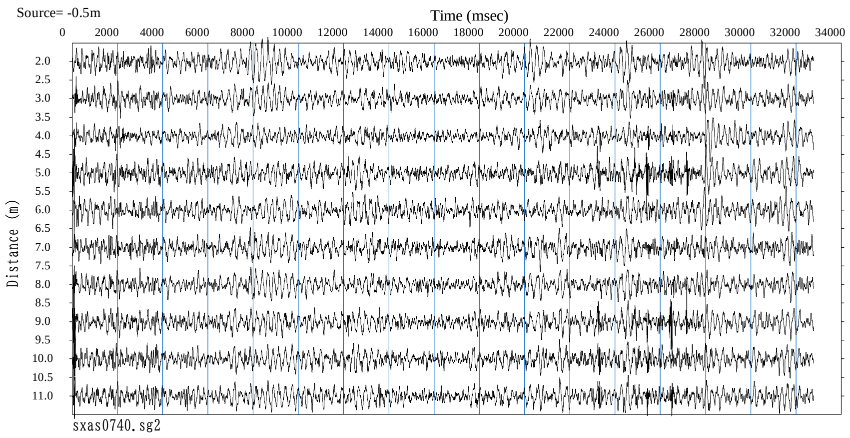
$$COH_{40m} = \frac{COH_{1-3} + COH_{1-5} + COH_{3-5}}{3}$$

Coherence for 40m



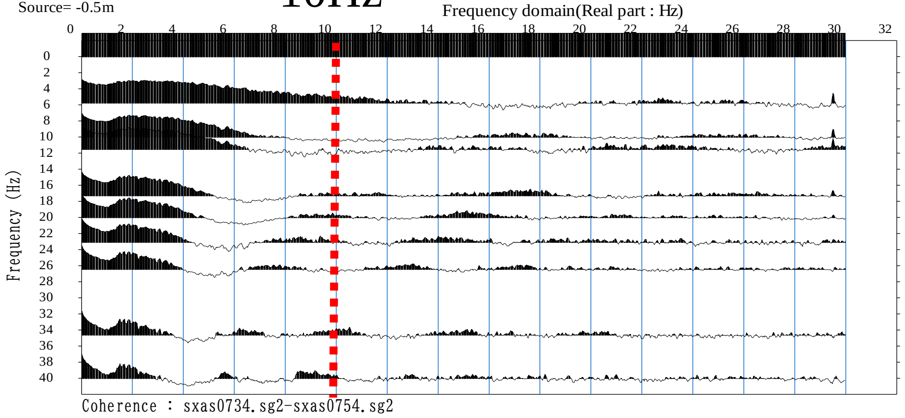
2D example

Time domain



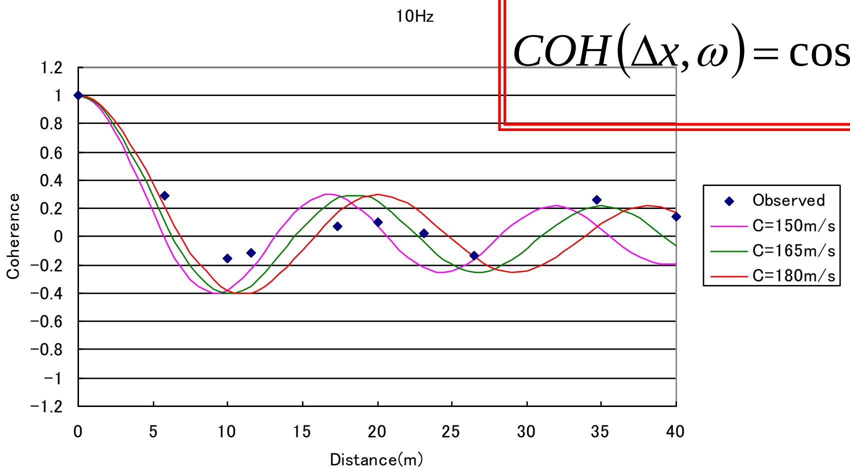
Coherence

10Hz

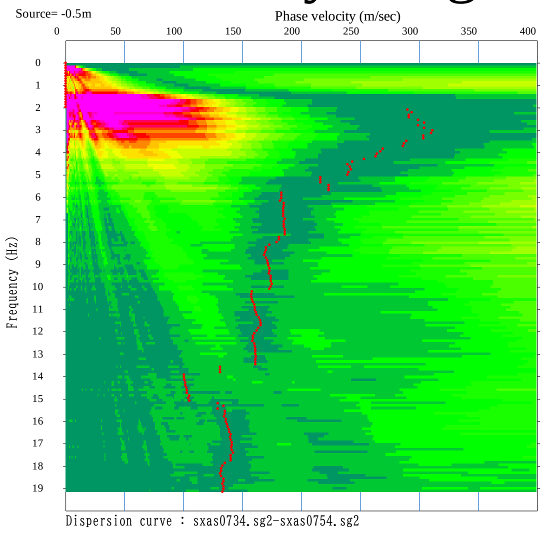


Coherence (function of Δx)

$$COH(\Delta x, \omega) = \cos\left(\frac{\omega}{c(\omega)}\Delta x\right)$$

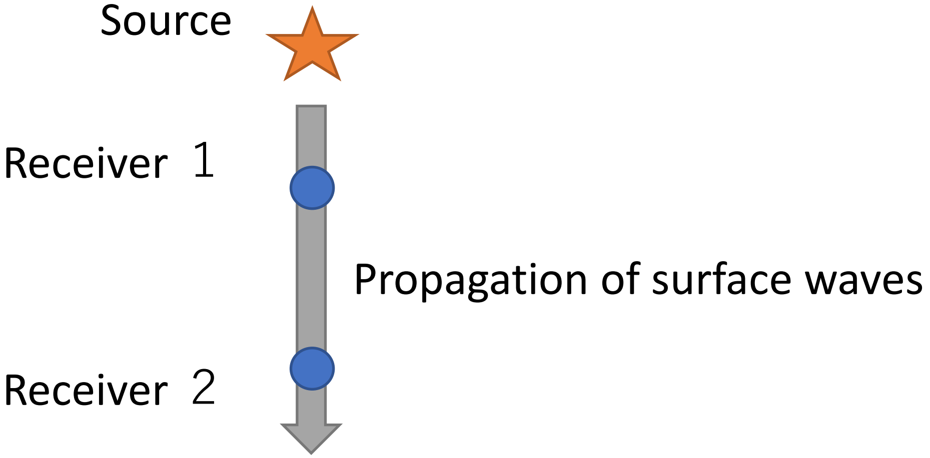


Phase-velocity image

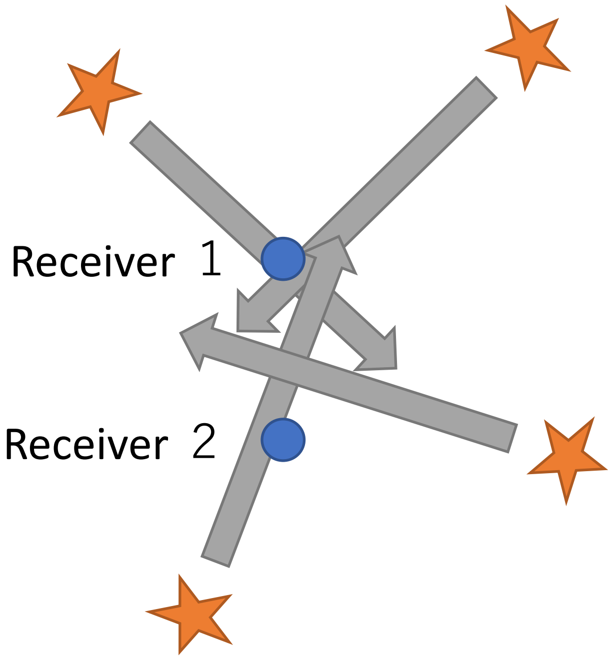


6. Seismic interferometry (SI)

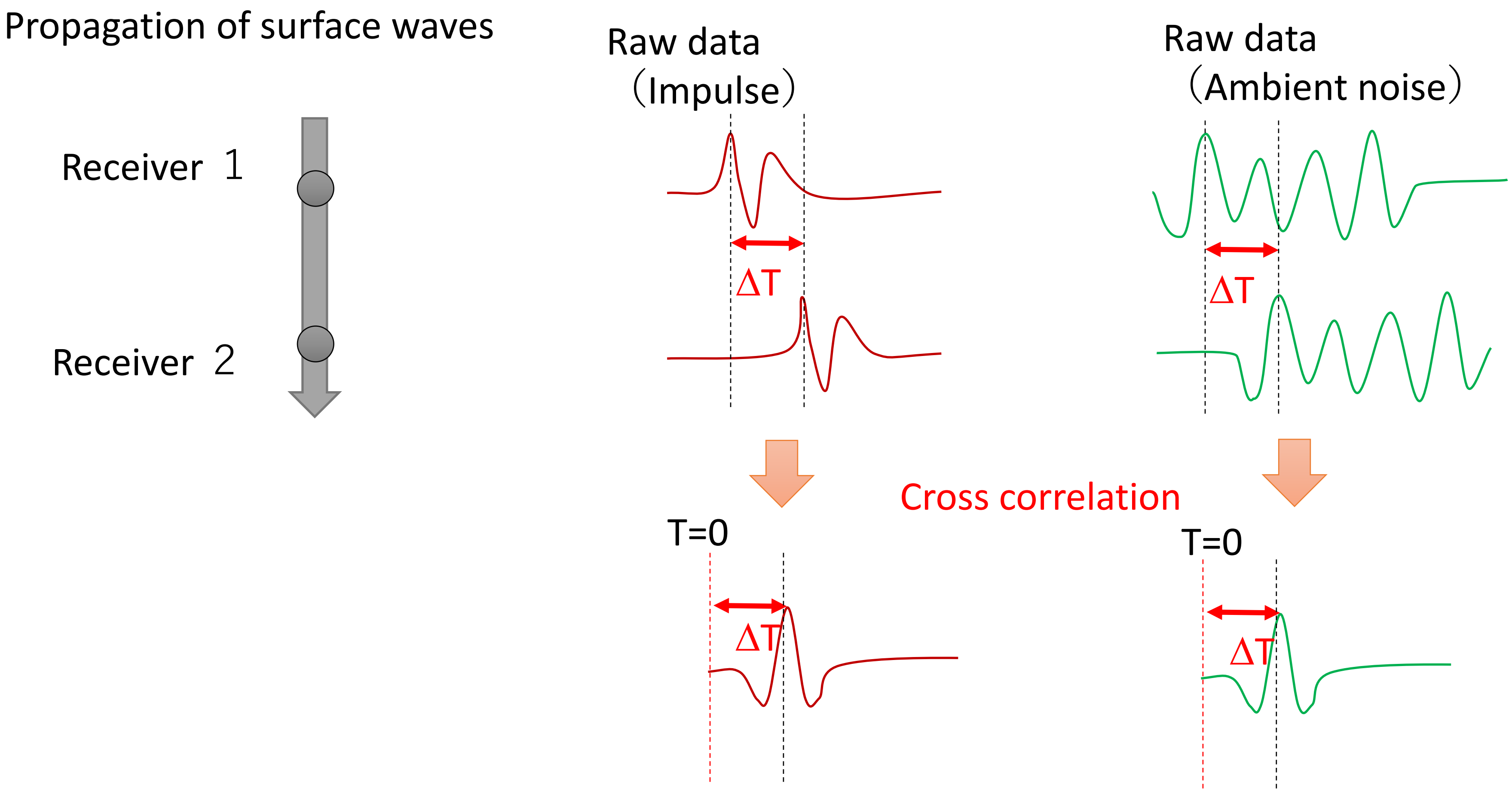
Active methods



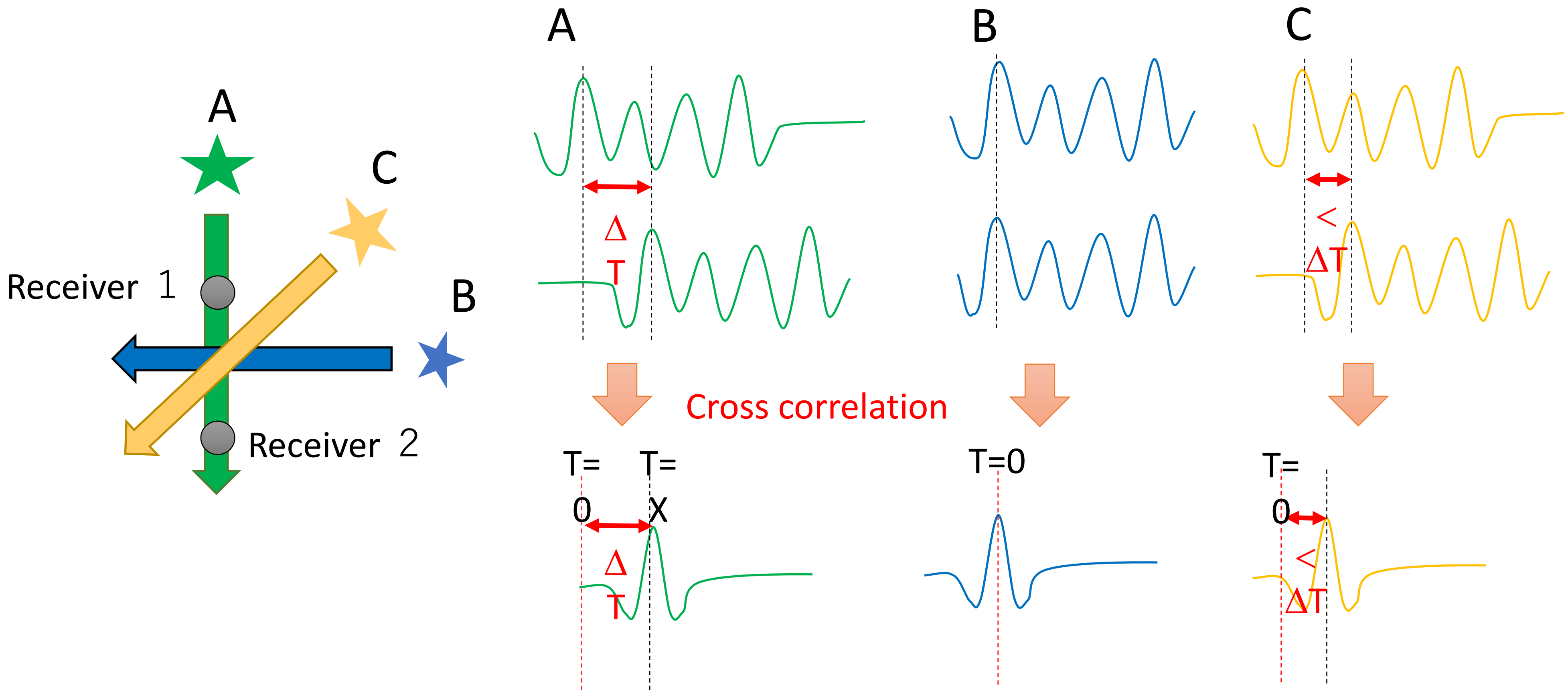
Passive methods



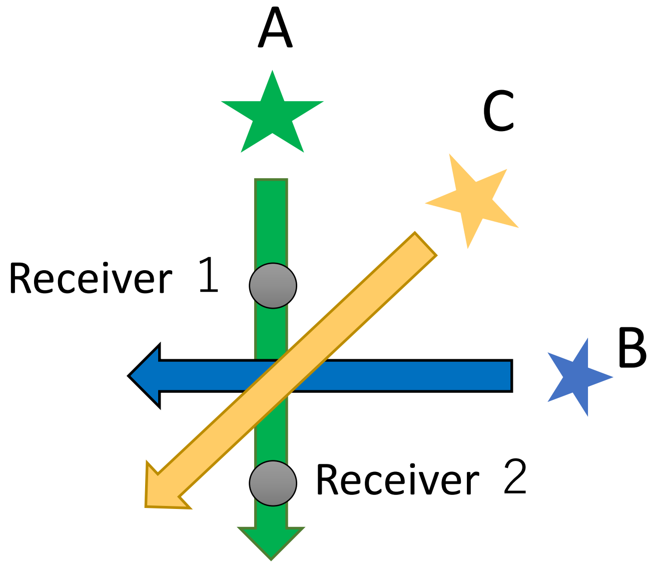
Cross correlation

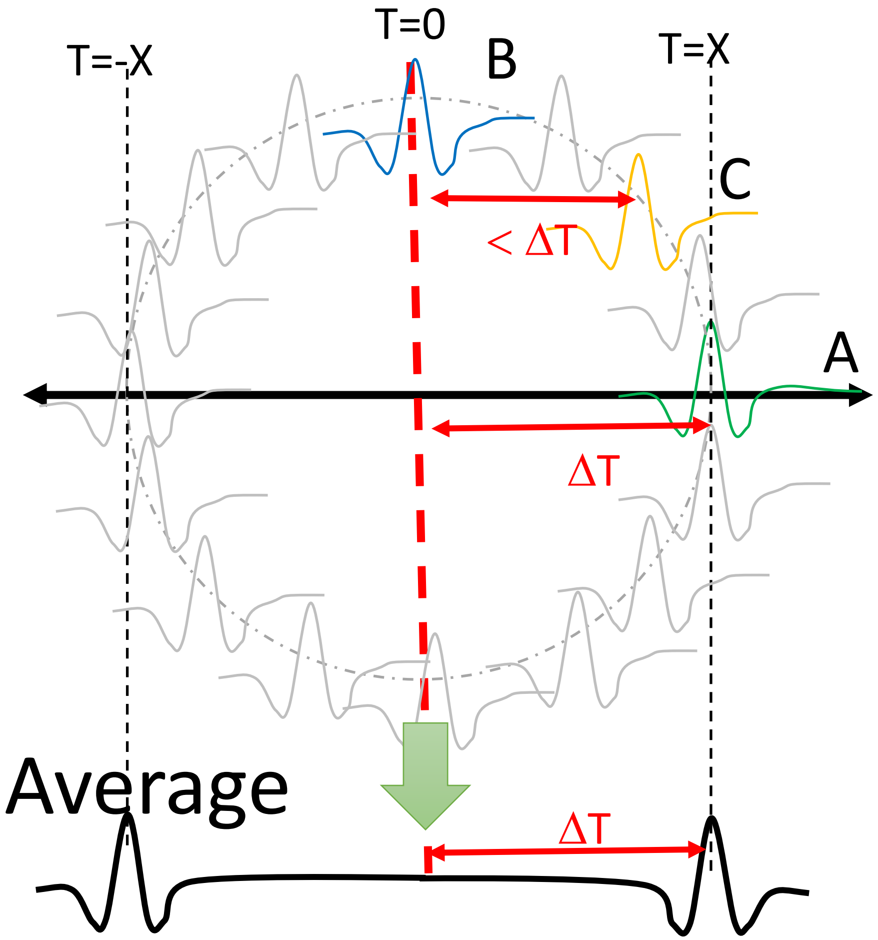


Cross correlation of ambient noise



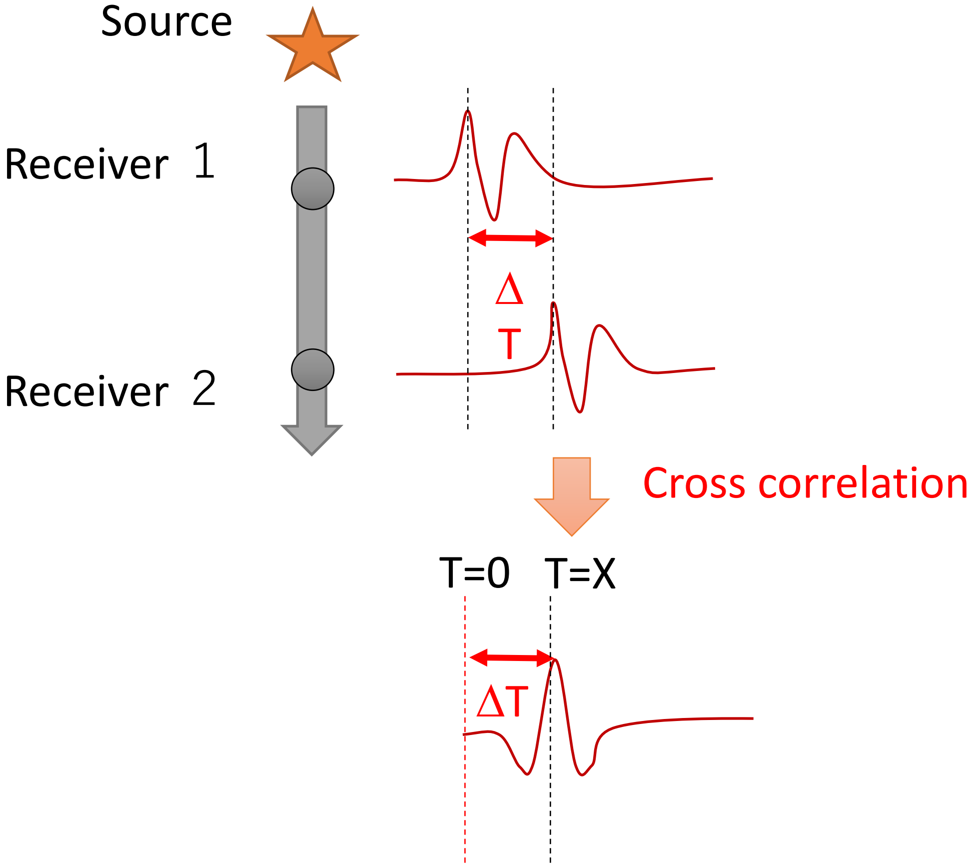
Cross correlation of ambient noise



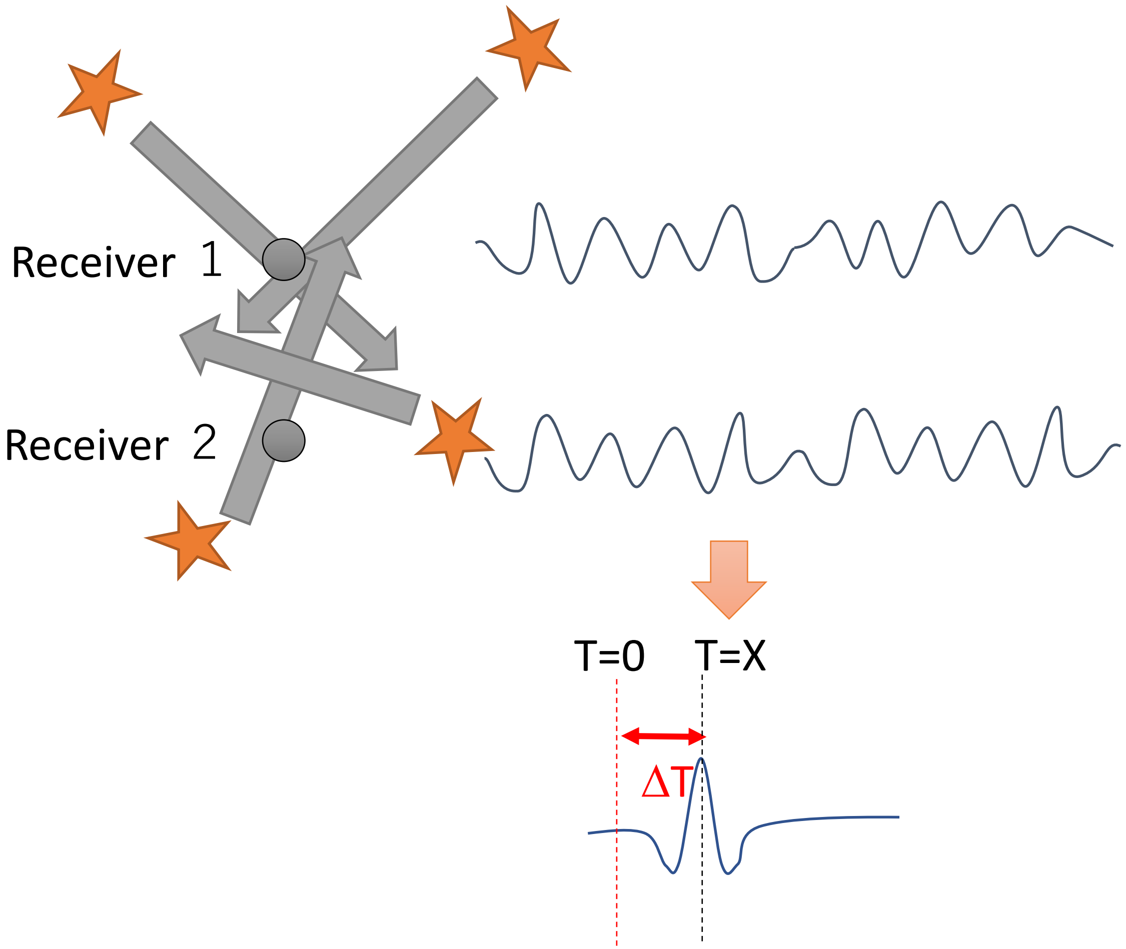


Active and passive surface-wave methods

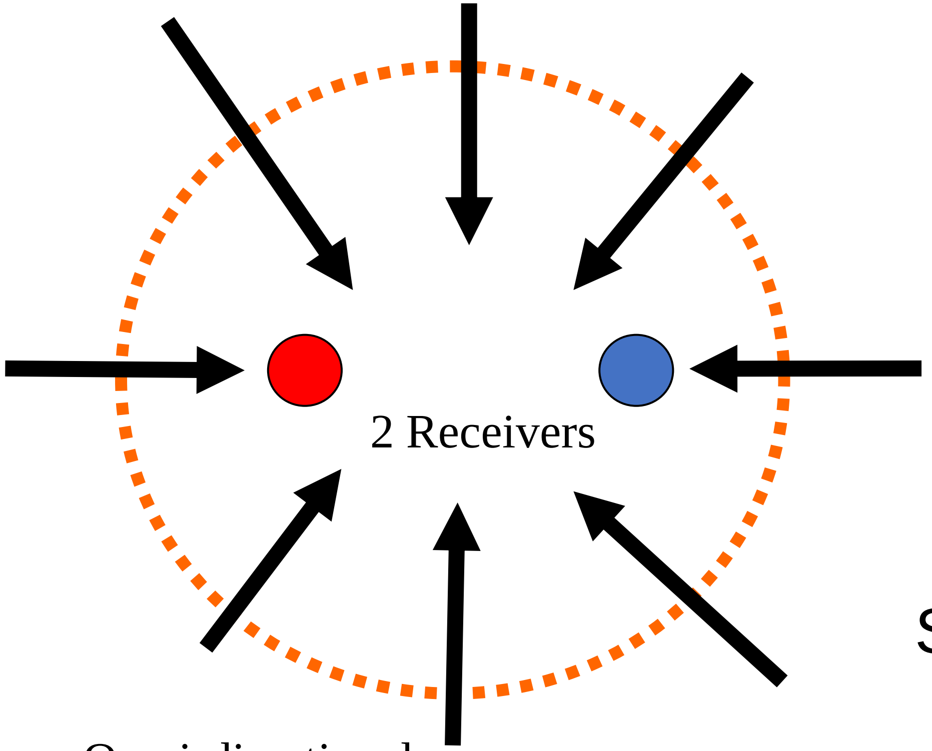
Active methods



Passive methods



SPAC and seismic interferometry (SI)



Omni-directional Propagation

Seismic interferometry (SI)

Green's function

$$F_1(\omega)\cdot\overline{F_2(\omega)} = CC_{f1f2}(\omega) = G(\omega)$$

Time domain

Cross-correlation

$$F_1(\omega)\cdot\overline{F_2(\omega)} = CC_{f1f2}(\omega)$$

Frequency domain
(Real part)

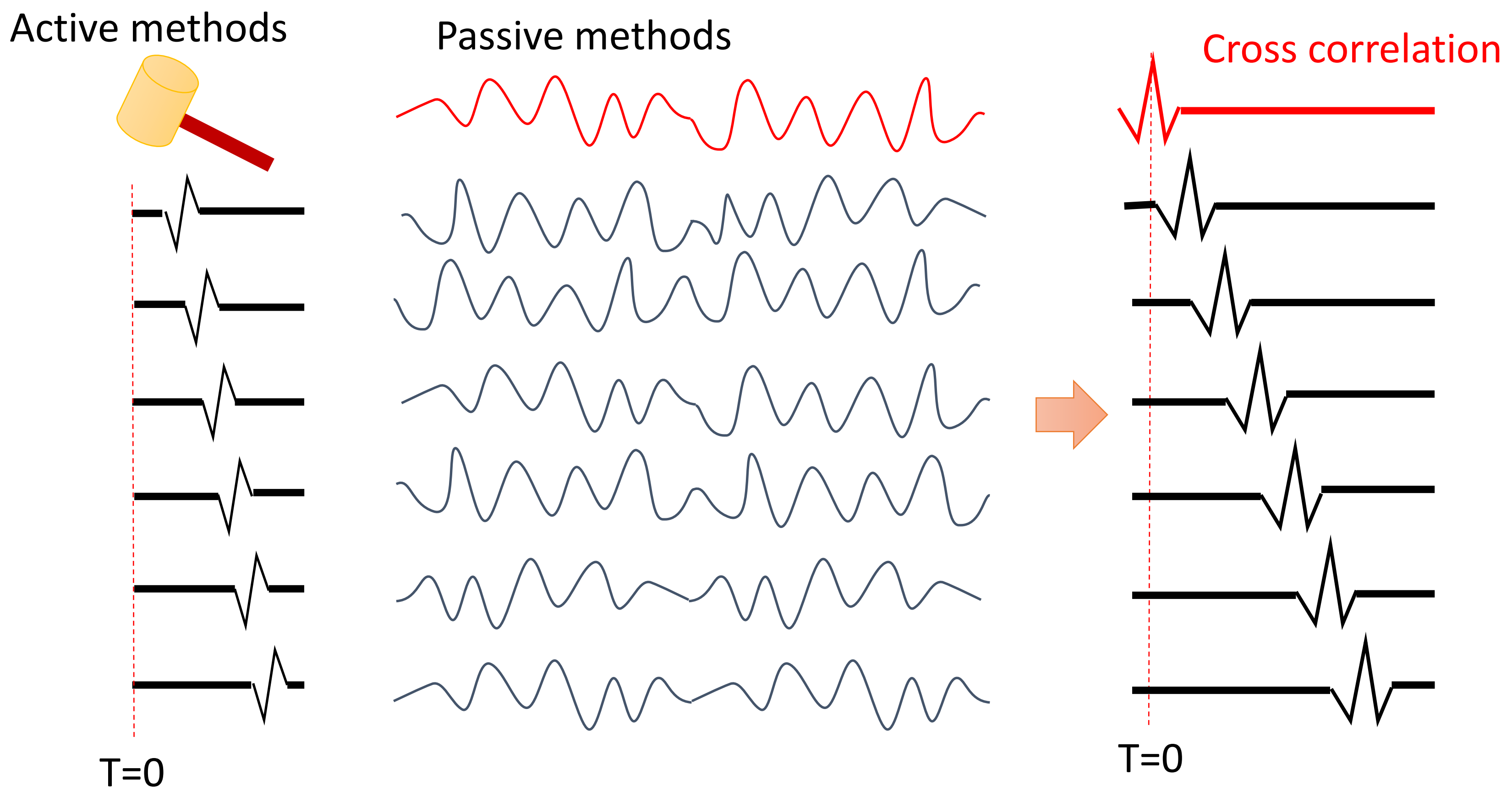
Spatial autocorrelation (SPAC)

Coherency

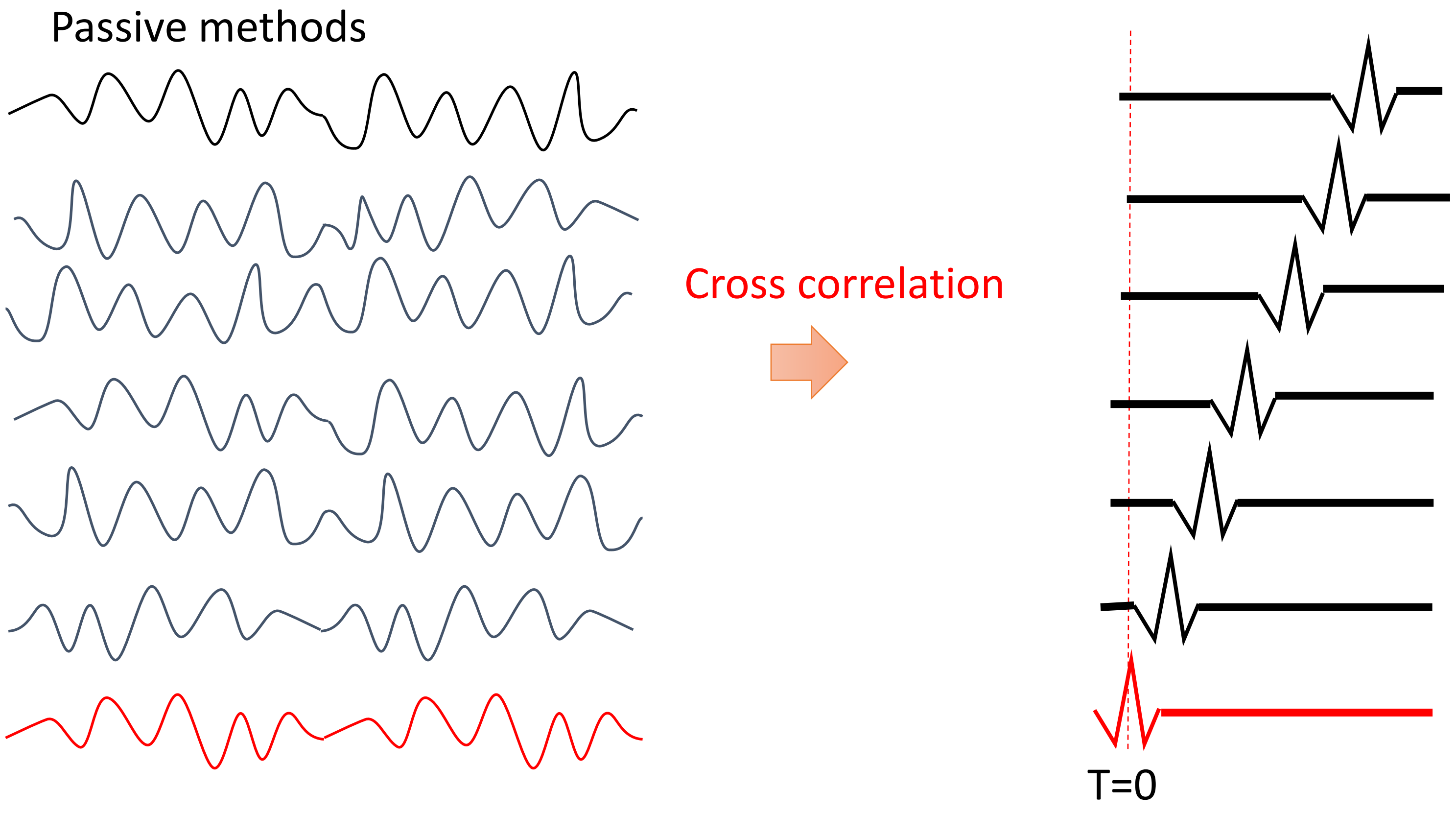
Bessel function

$$\frac{\mathrm{Re}\left(CC_{f1f2}(\omega)\right)}{A_{f1}(\omega)A_{f2}(\omega)} = j_0\left(\frac{\omega\cdot\Delta x}{c(\omega)}\right)$$

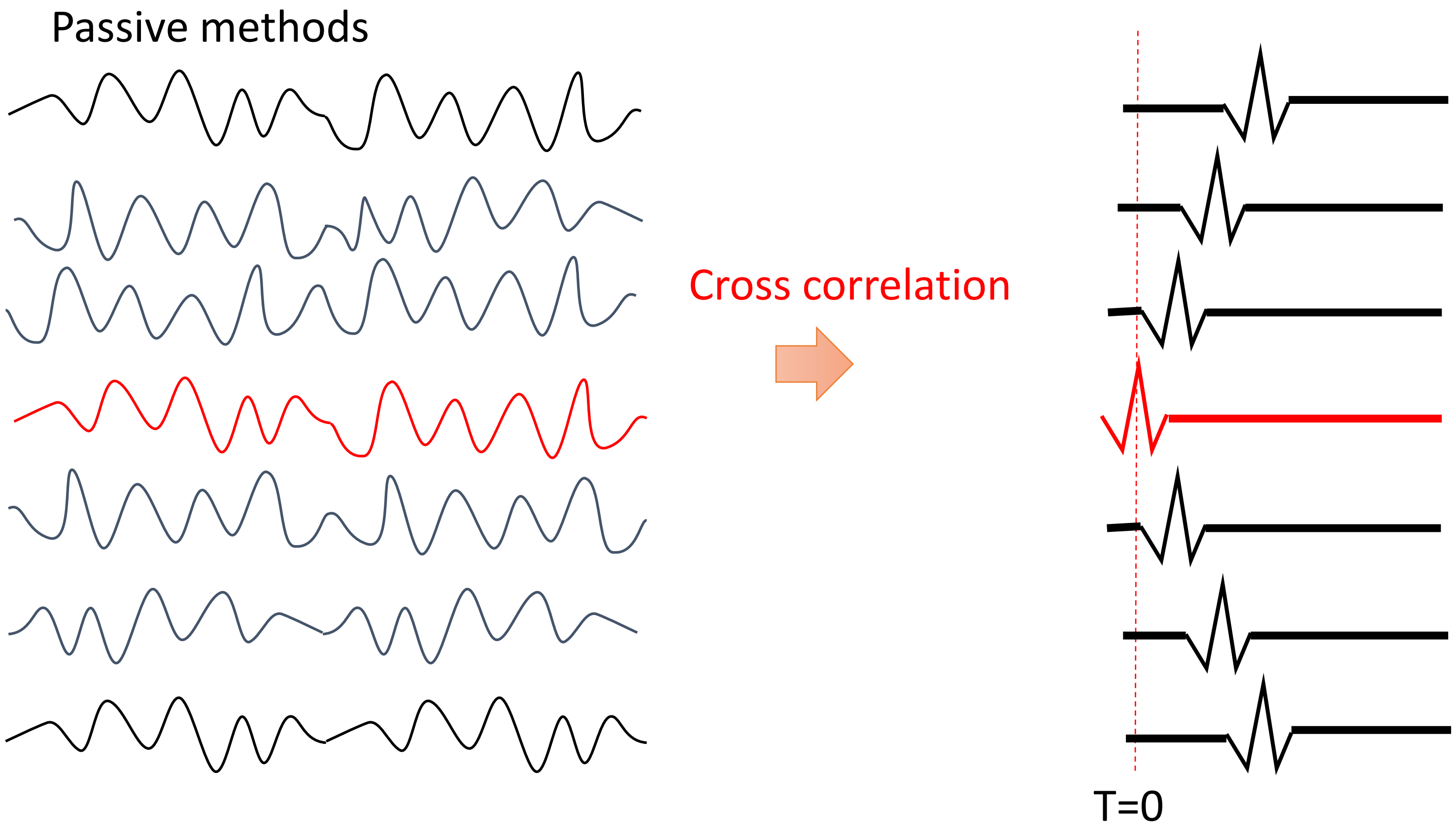
Active and passive surface-wave methods



Active and passive surface-wave methods

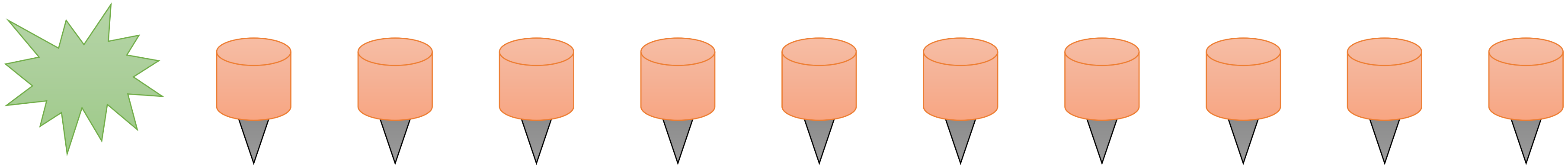


Active and passive surface-wave methods



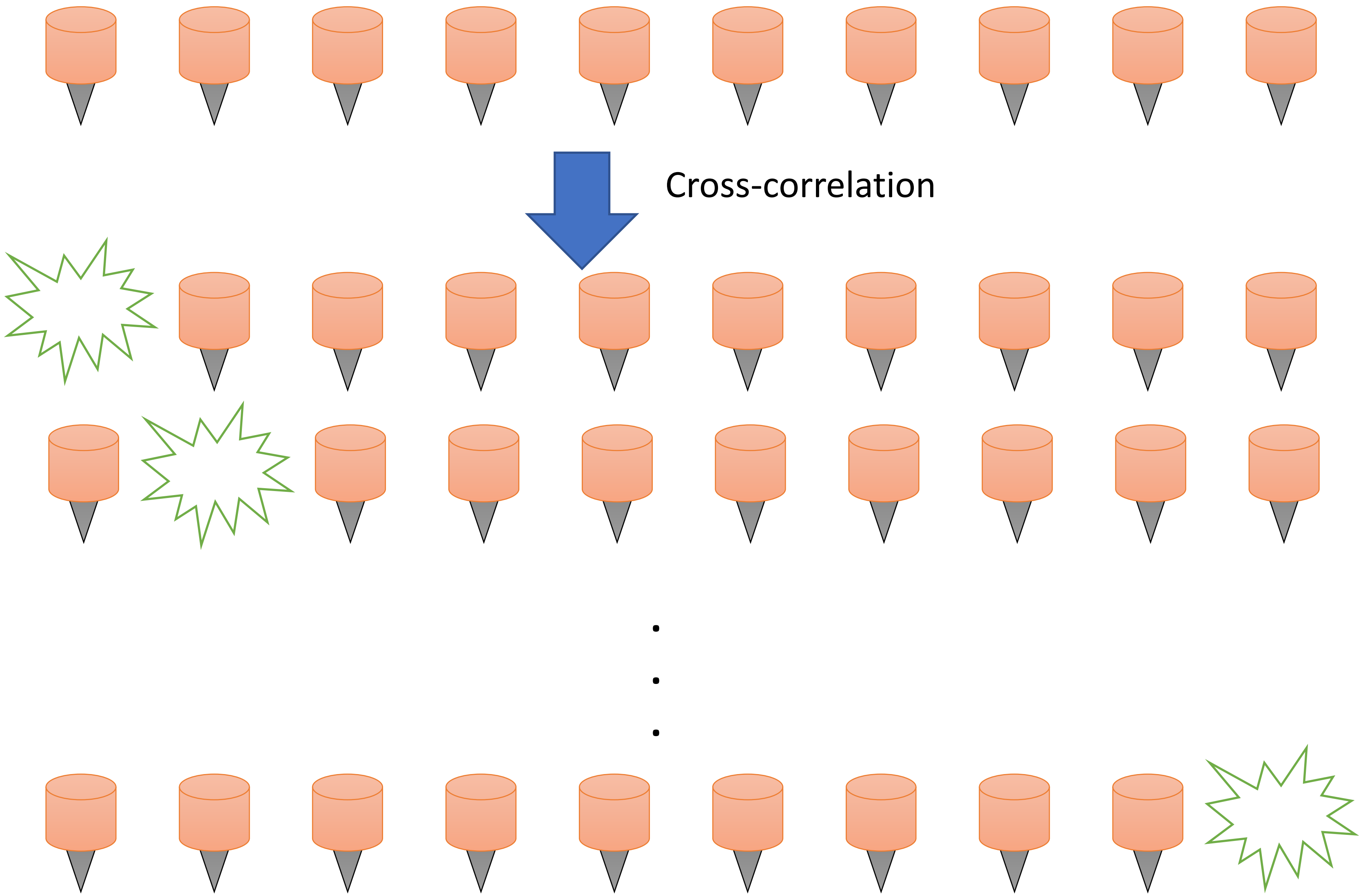
Number of Sources can be Easily Increased in Passive Methods based on the Theory of Seismic Interferometry

Active methods



One shot gives us one shot record.

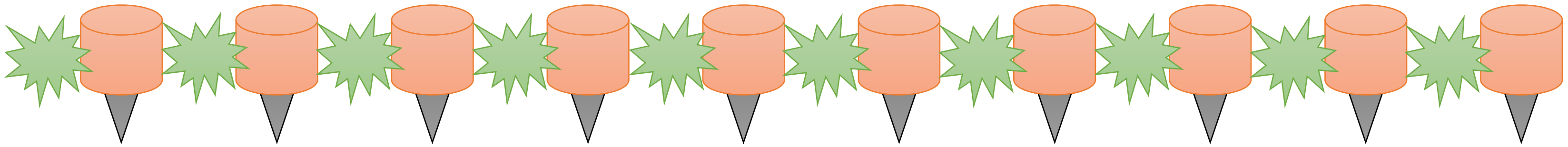
Passive methods



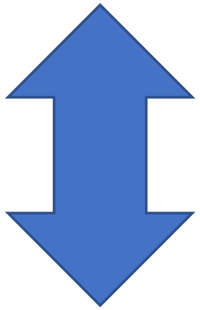
10 min. of ambient noise give us 10 virtual shot records.

Assume we need 10 receivers and 10 shots

Active methods

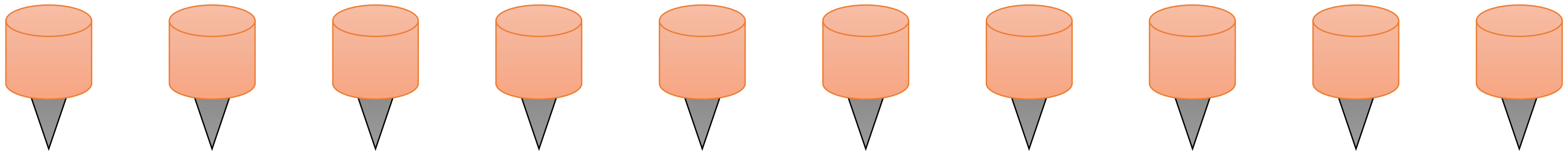


10 shots take 10 min.



Same information

Passive methods



10 min. of ambient noise give us 10 virtual shots

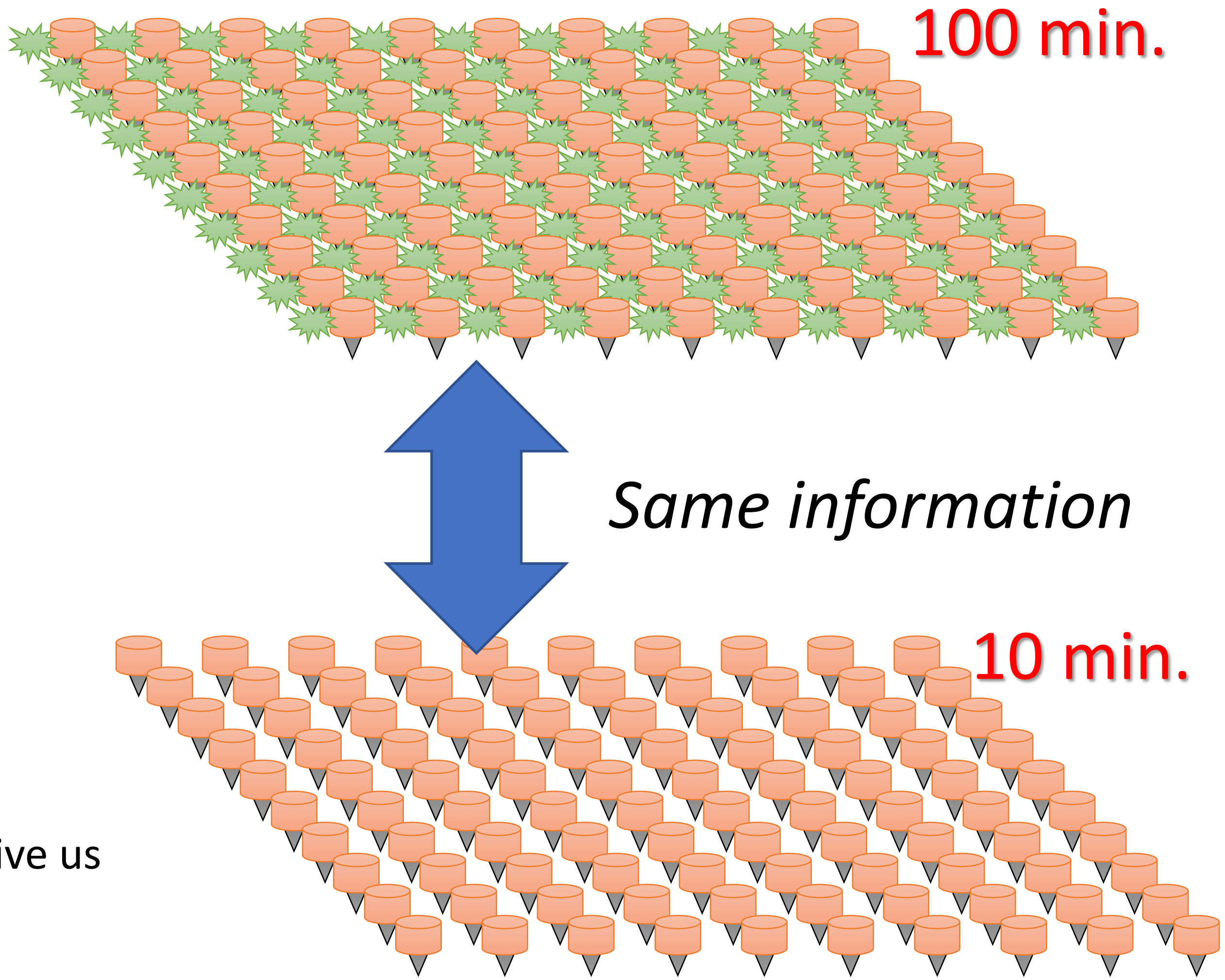
Assume we need 100 receivers and 100 shots

Active methods

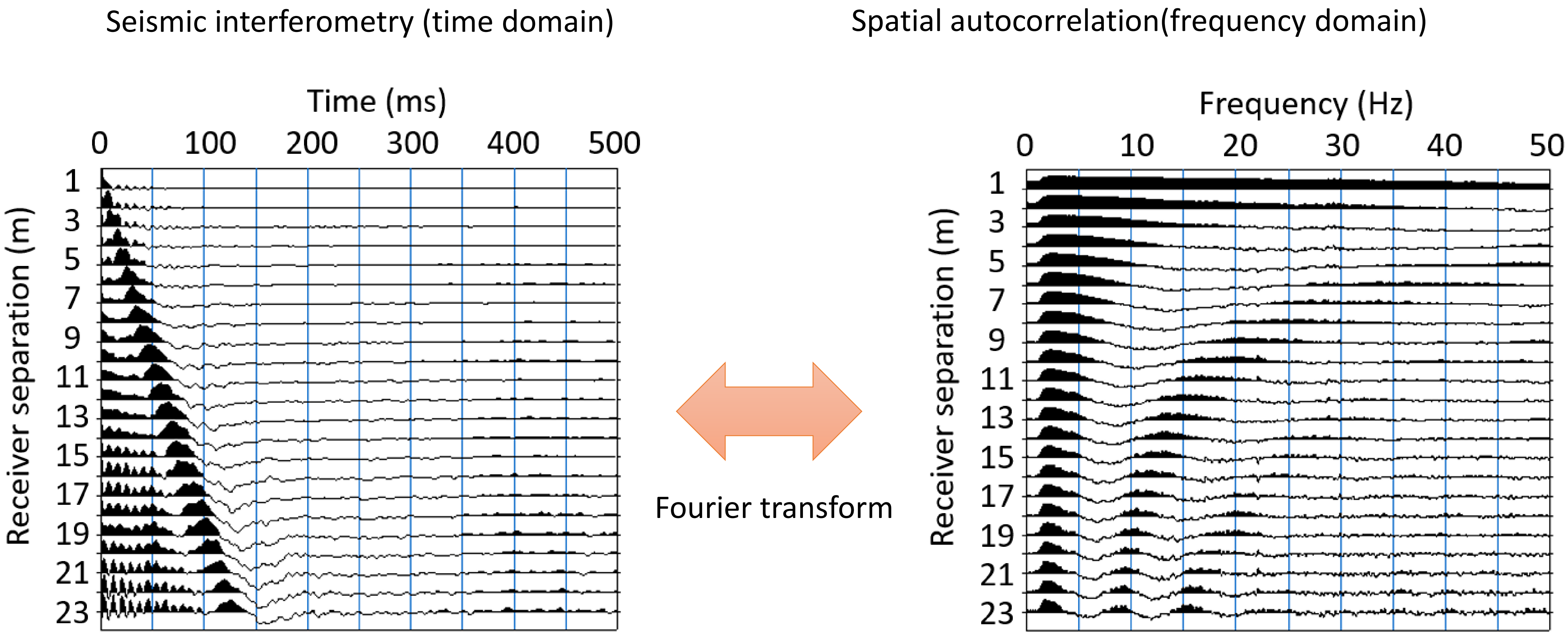
100 shots take 100 min.

Passive methods

10 min. of ambient noise give us
100 virtual shots

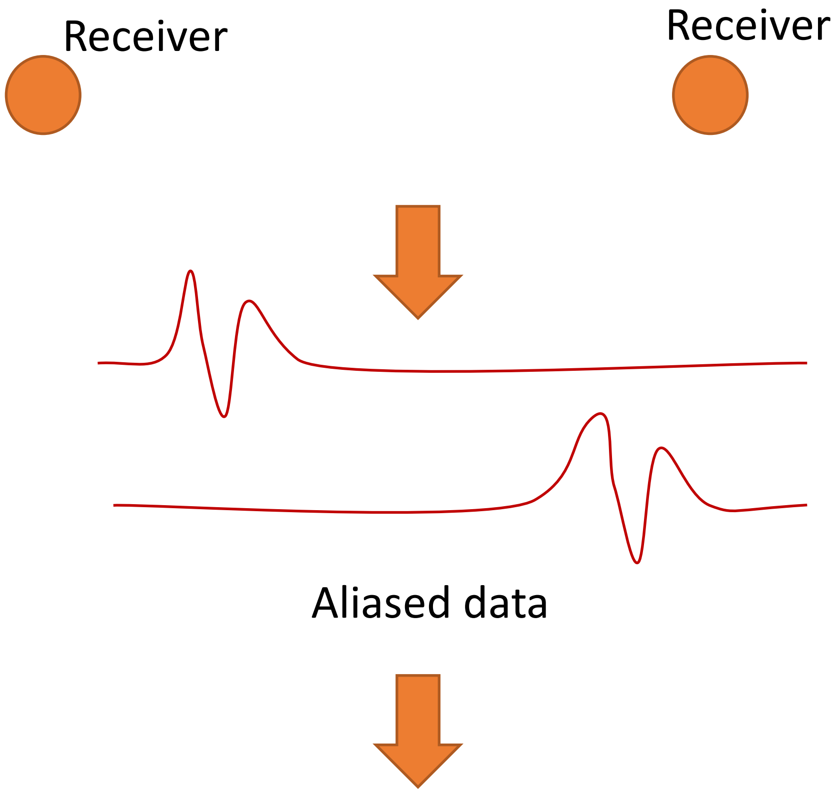


Seismic interferometry (SI) Vs Spatial autocorrelation (SPAC)



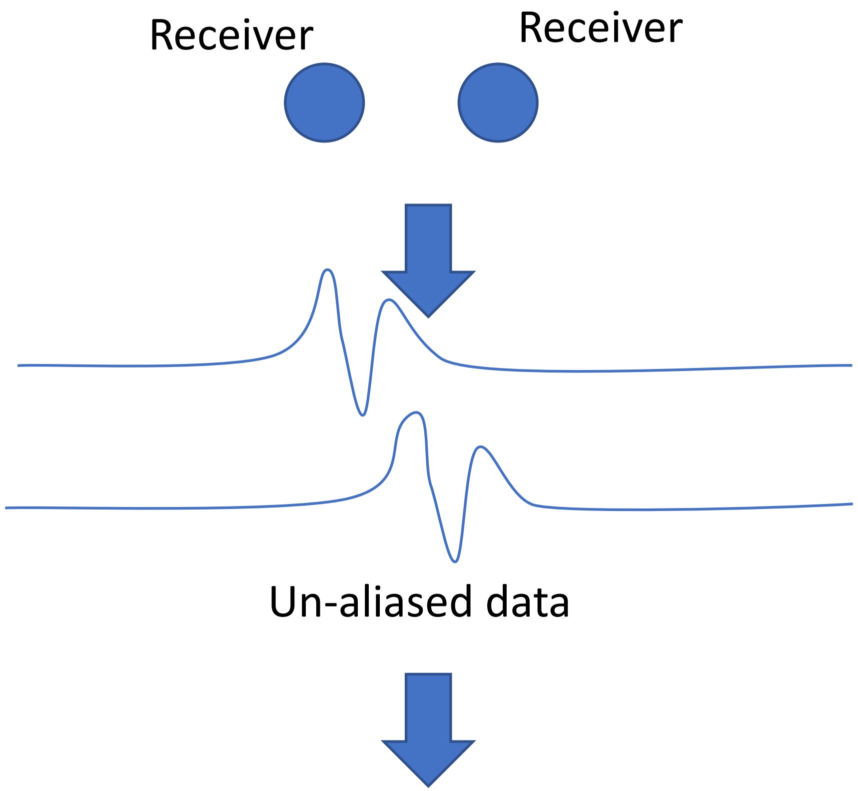
Seismic interferometry (SI) Vs Spatial autocorrelation (SPAC)

Seismic interferometry (SI)



Calculate group velocity in time domain

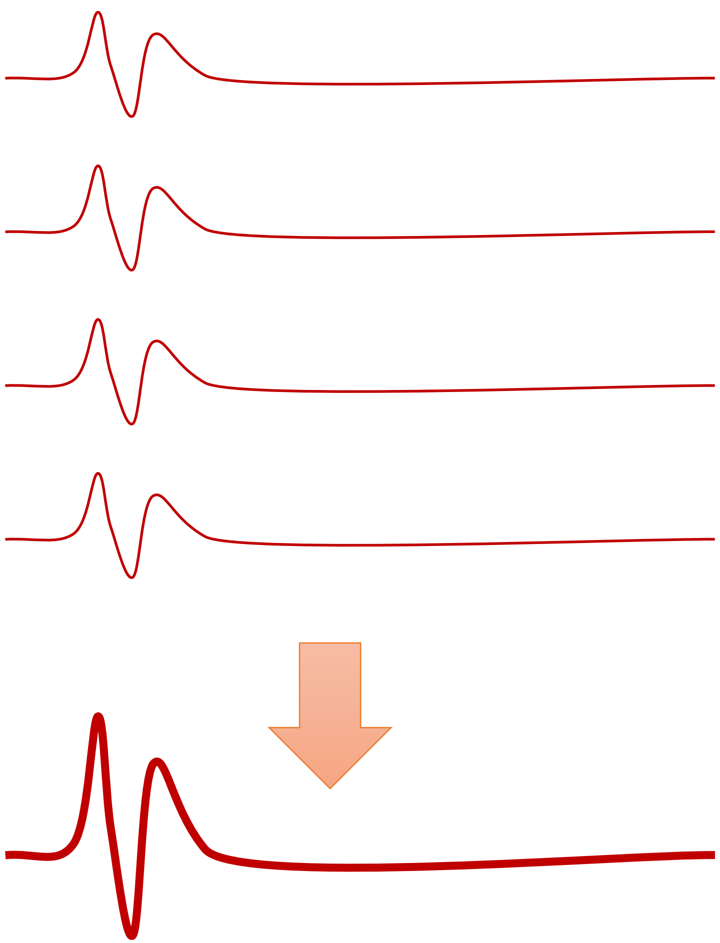
Spatial autocorrelation (SPAC)



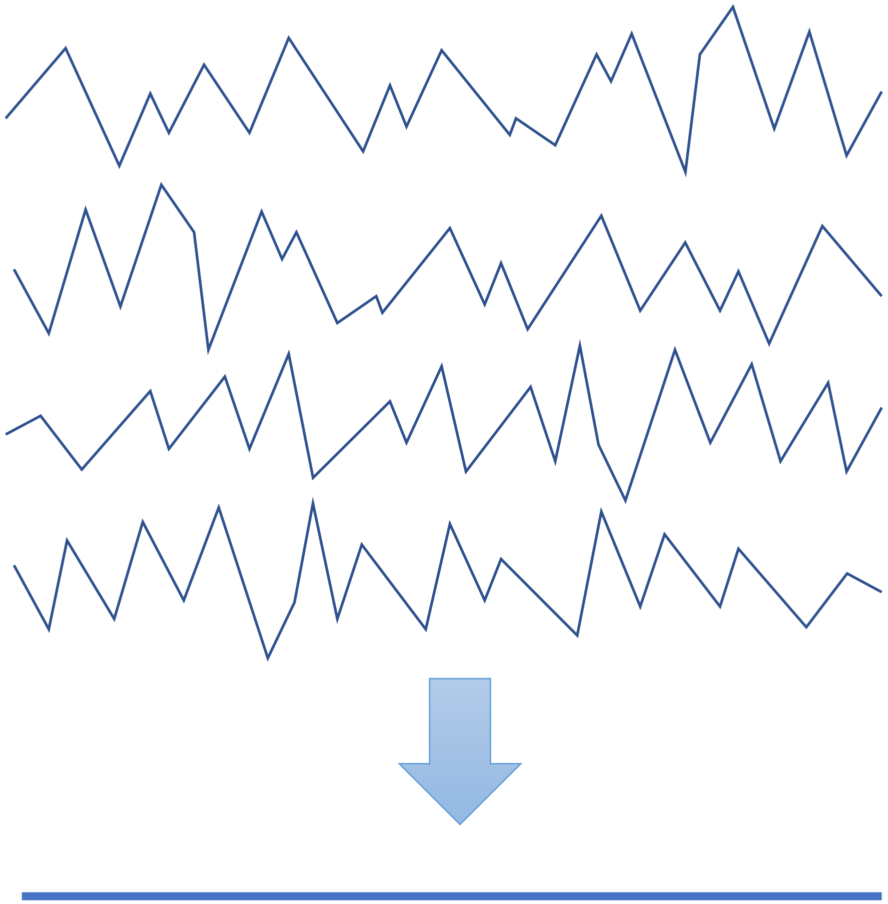
Calculate phase velocity in frequency domain

Active and passive data processing in terms of stacking

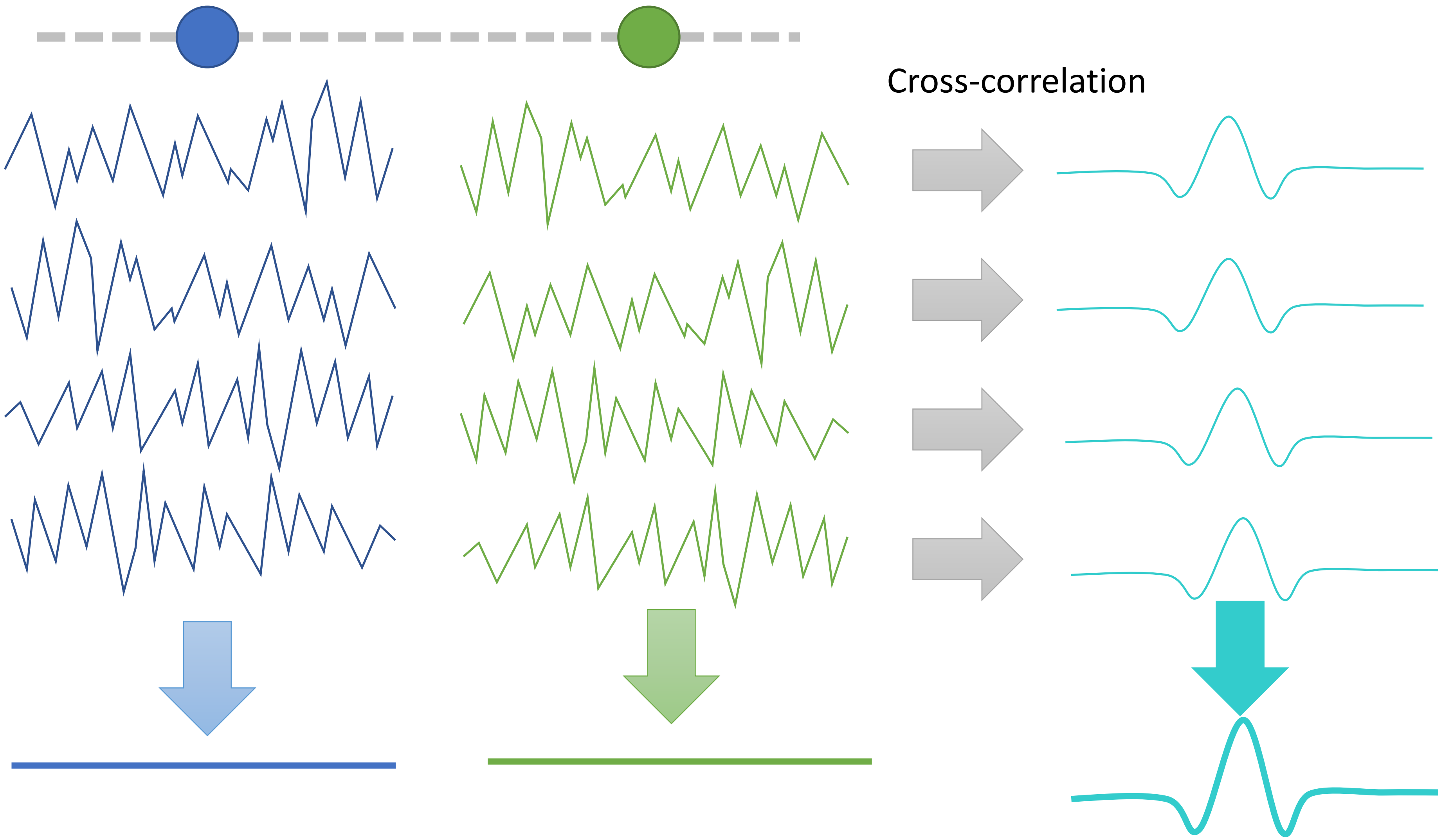
Active data



Passive data (random stochastic data)



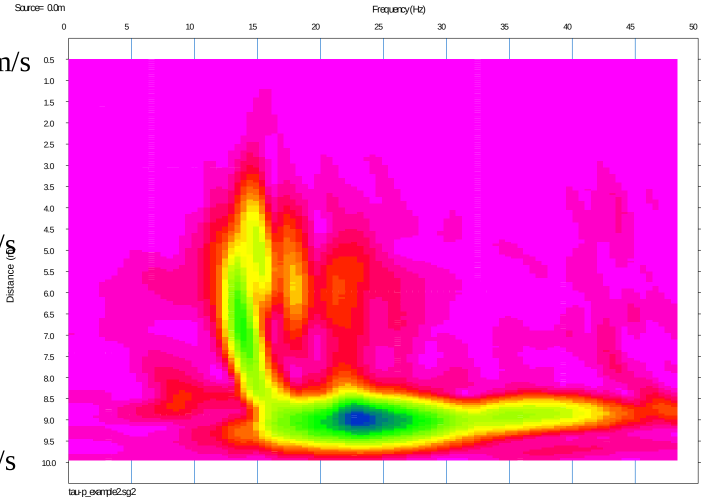
How to stack passive data ?



Comparison of dispersion curves

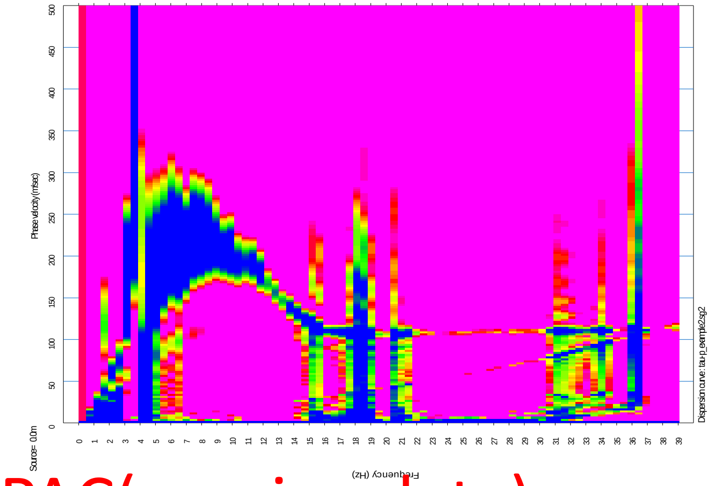
τ-p transform

p (ray parameter)
2000m/s
200m/s
100m/s



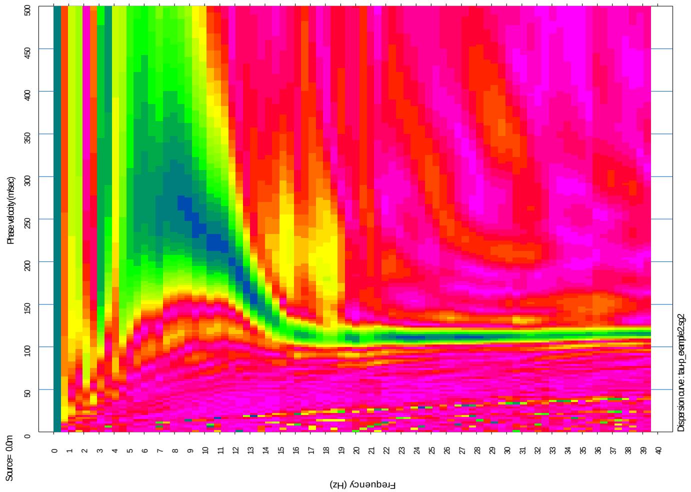
1D SPAC

c (phase-velocity)
400m/s
200m/s
100m/s



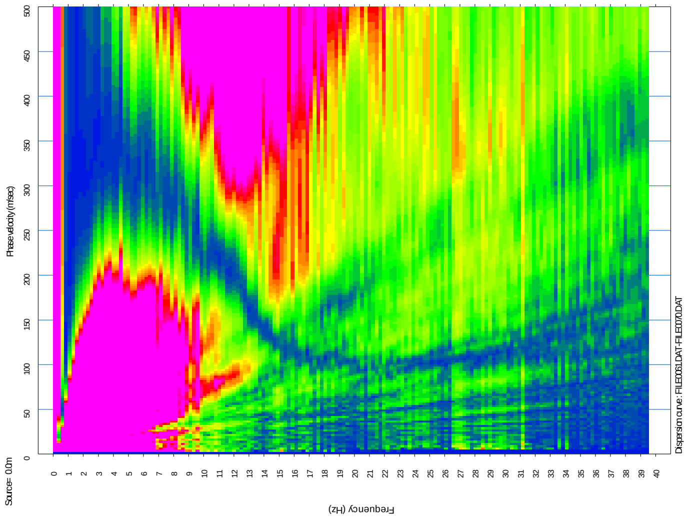
MASW

c (phase-velocity)
400m/s
200m/s
100m/s



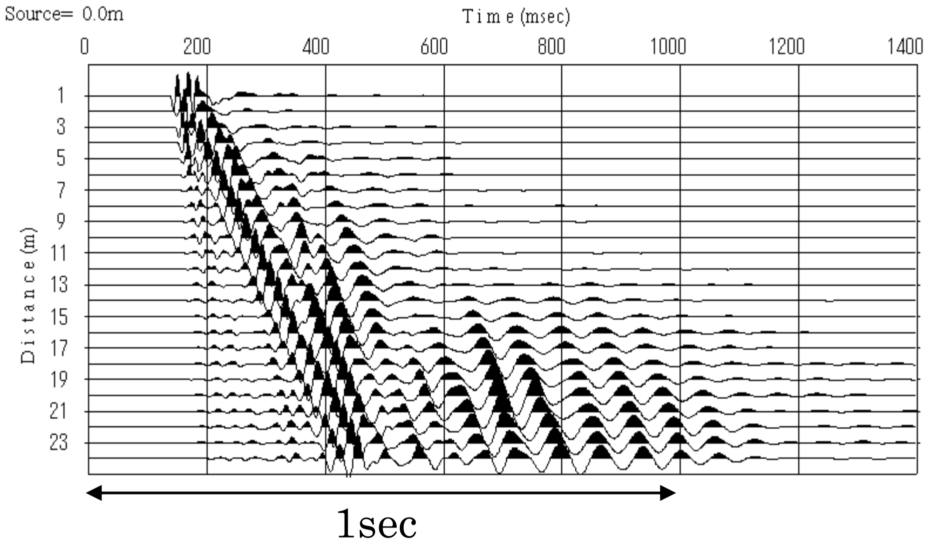
2D SPAC(passive data)

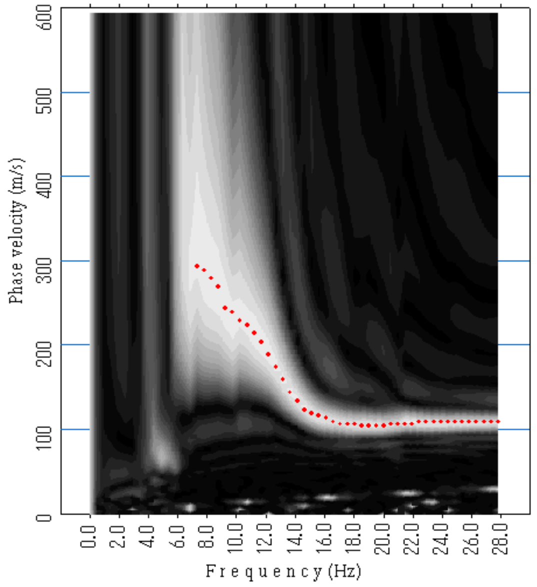
c (phase-velocity)
400m/s
200m/s
100m/s



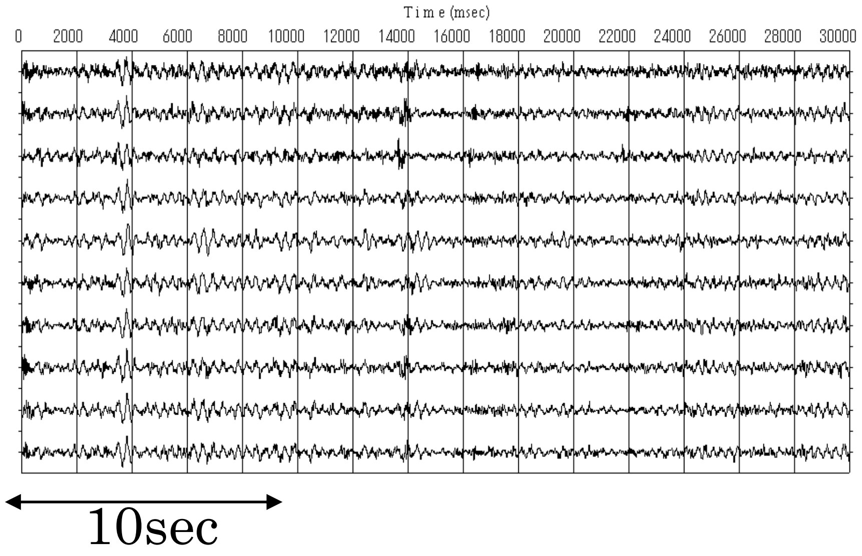
Comparison of dispersion curves

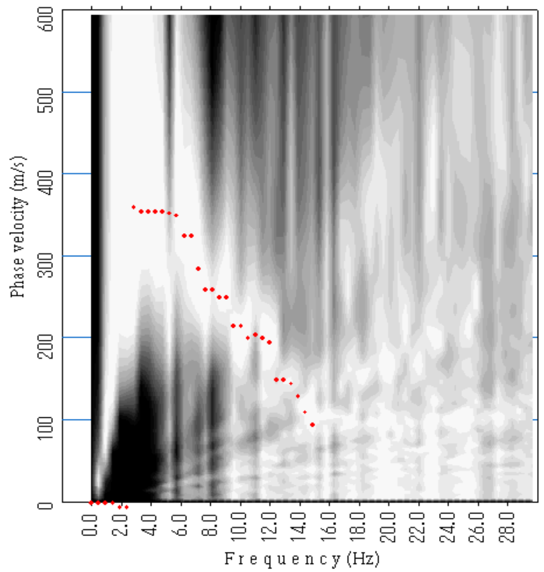
Active





Passive





Comparison of dispersion curves