

seismic imaging tutorial

“exploding reflector” modeling/migration

Paul Sava

**Center for Wave Phenomena
Colorado School of Mines
psava@mines.edu**

assignment

**modify the acquisition parameters to
explore the illumination at different
locations in the subsurface**

import packages

```
from rsfproj import *
import sigsbee, fdmod
```

setup main parameters

```
par=sigsbee.paramwin() # Sigsbee2A parameters
par['nt']=3001           # time steps (samples)
par['kt']=100            # wavelet delay (samples)
par['dt']=0.001          # time sampling (ms)
par['nb']=100            # boundary size (grid points)
fdmod.param(par)        # plotting parameters
```

source coordinates

```
# source coordinates (exploding reflectors)
fdmod.boxarray('ss',6,2,1,12,8,1,par)

# plot sources
Plot('ss',fdmod.ssplot('plotfat=1 symbol=.',par))
```

receiver coordinates

```
par['jr']=4      # receiver jump (grid points)
par['nr']=120    # number of receivers
par['fr']=100    # receivers origin (grid points)

# receiver coordinates
fdmod.horizontal('tt',par['oz']+par['dz'],par)
Flow('rr',
     'tt',
     'window n2=%(nr)d j2=%(jr)d f2=%(fr)d %par')

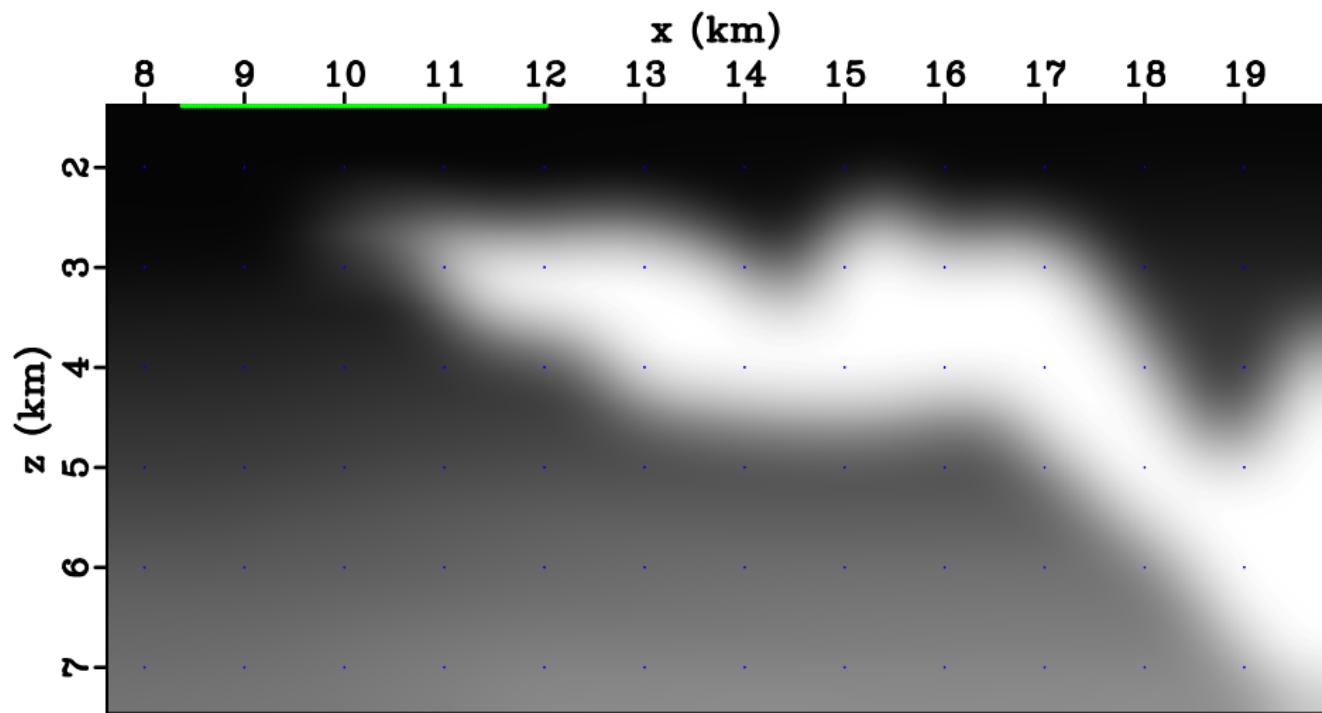
# plot receivers
Plot('rr',fdmod.rrplot(' ',par))
```

velocity/density models

```
# get velocity
sigsbee.getstrvelwin('vstr',par)
Flow(  'velo',
      'vstr',
      'smooth rect1=100 rect2=100 repeat=1')

# plot velocity
Plot(  'velo',fdmod.cgrey('allpos=y bias=1.43',par)
Result('velo',[ 'velo','ss','rr'], 'Overlay')

# plot density
Flow('dens','velo','math output=1')
```

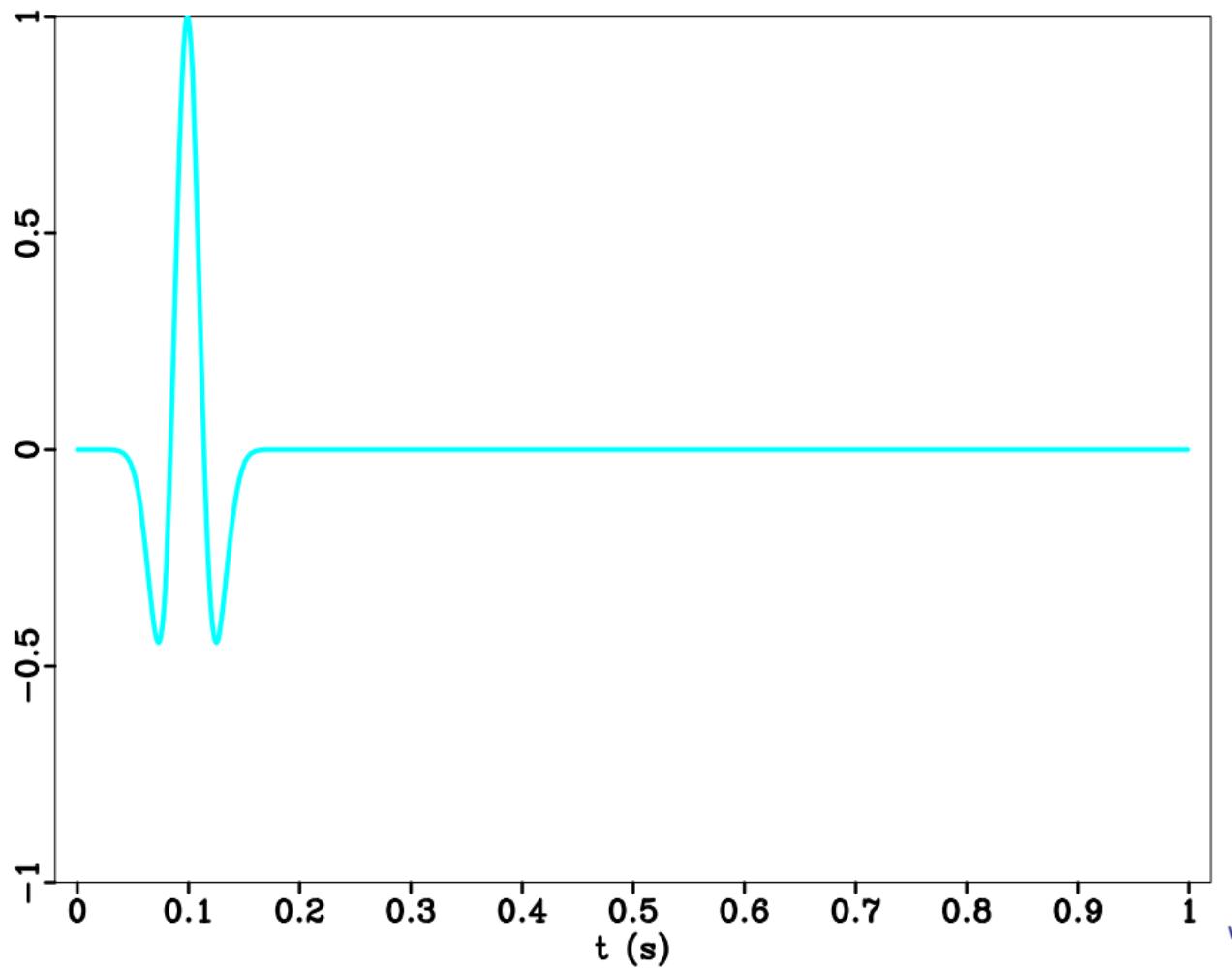


source wavelet

```
# construct wavelet
fdmod.wavelet('wav_',15,par)

# transpose wavelet
Flow('wav','wav_','transp')

# plot wavelet
Result('wav','window n2=1000 | '
+ fdmod.waveplot(' ',par))
```



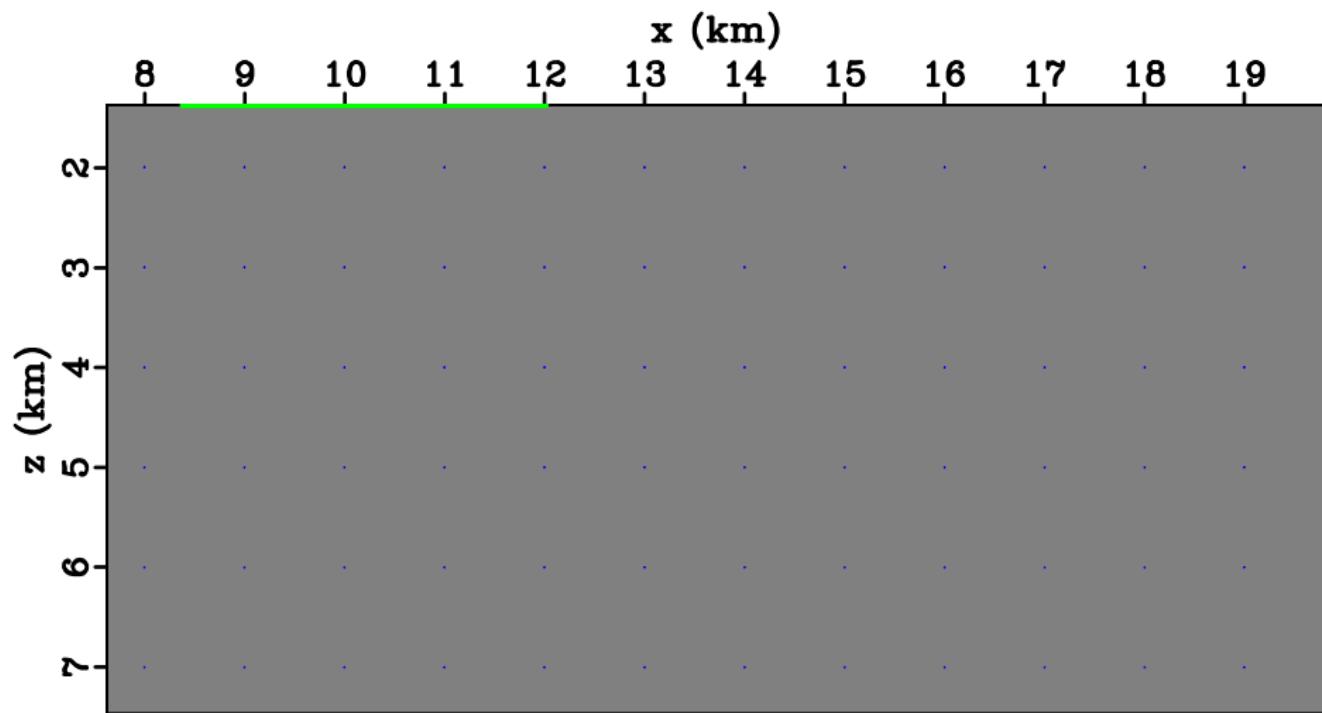
FD modeling

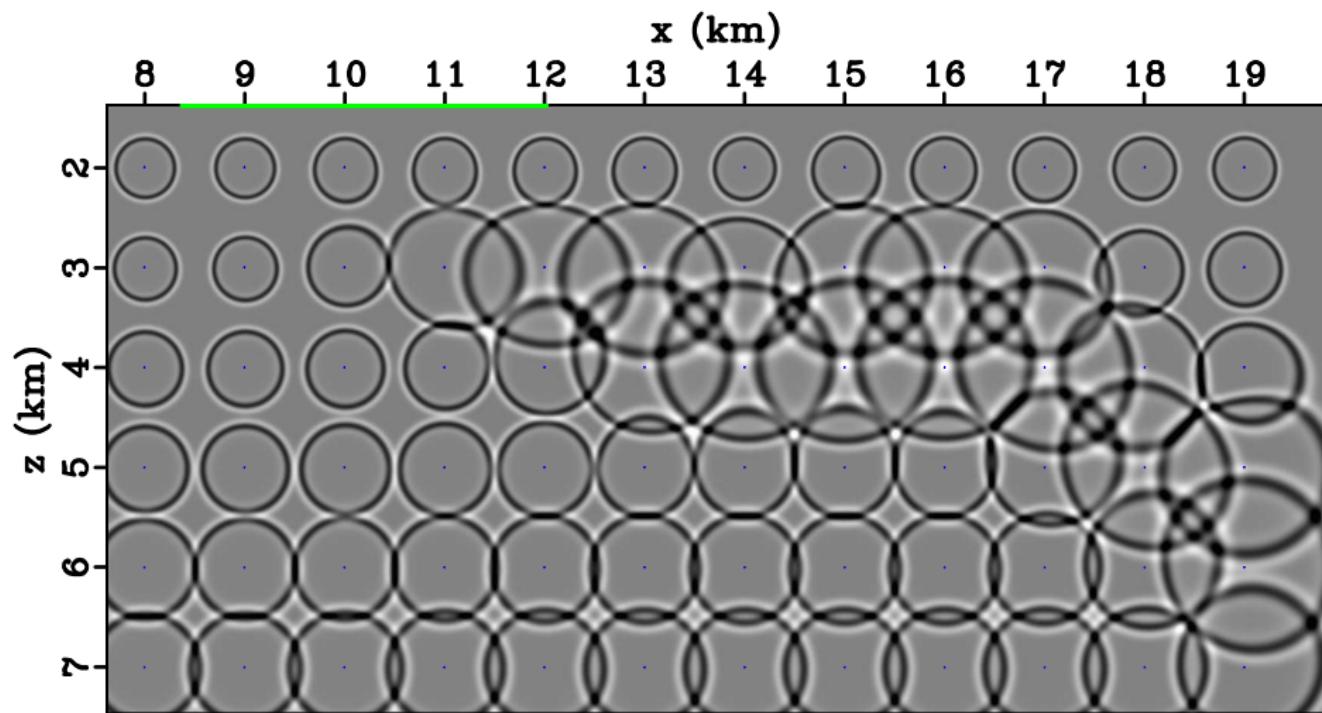
```
# run FD modeling
fdmod.awefd1('temp','wfld',
              'wav','velo','dens',
              'ss','rr',
              'free=n',par)
```

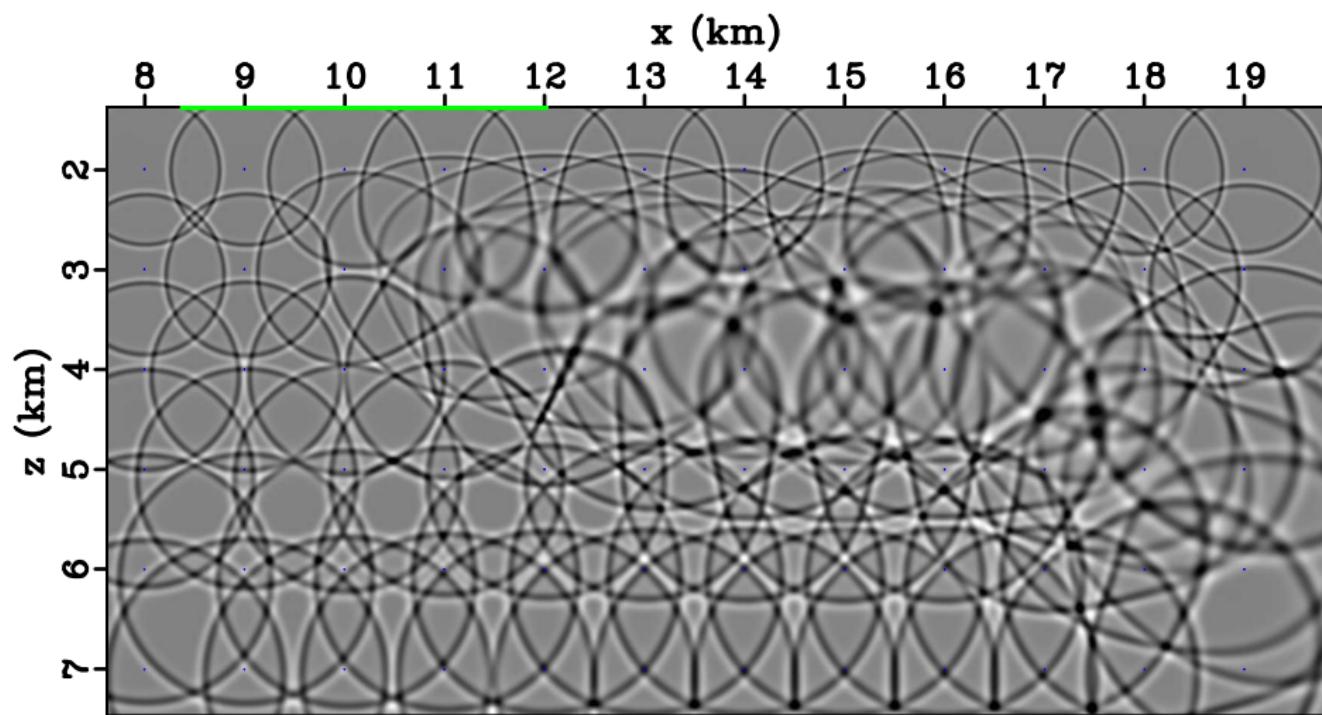
plot wavefield

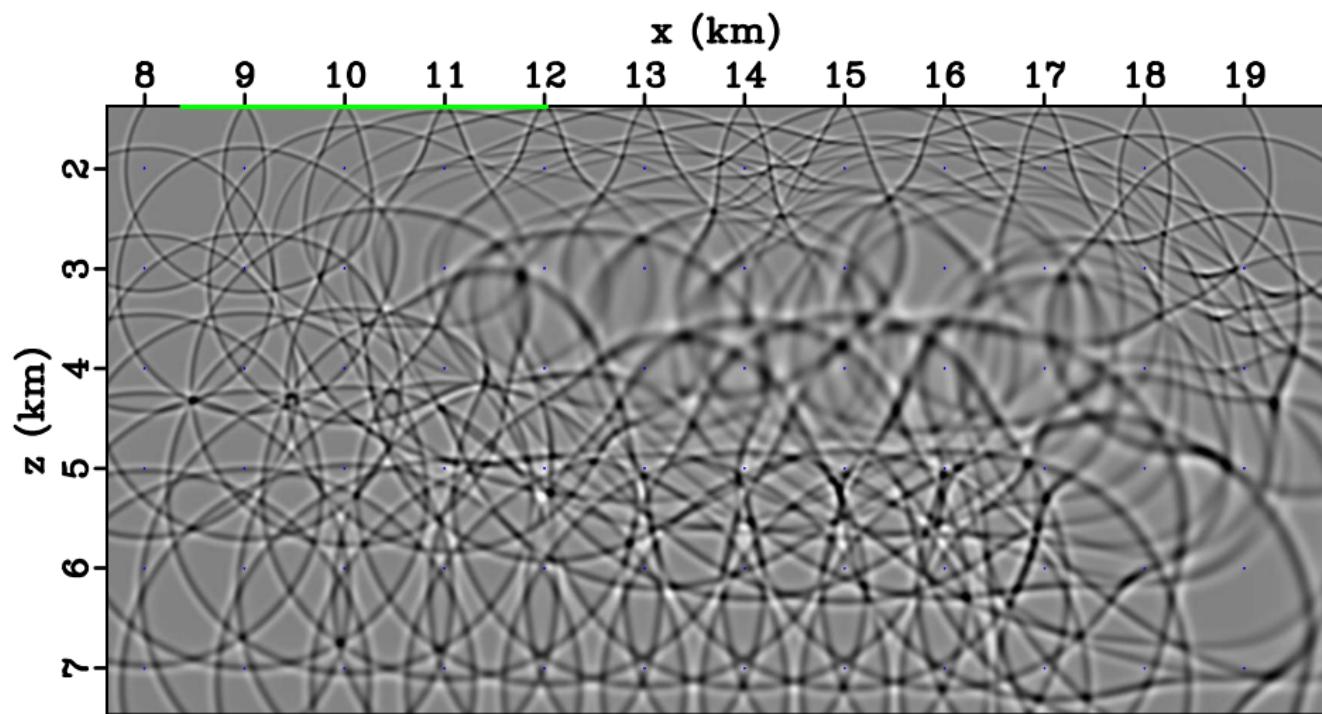
```
# generate wavefield movie
Plot('wfld',fdmod.wgrey('pclip=99',par),view=1)

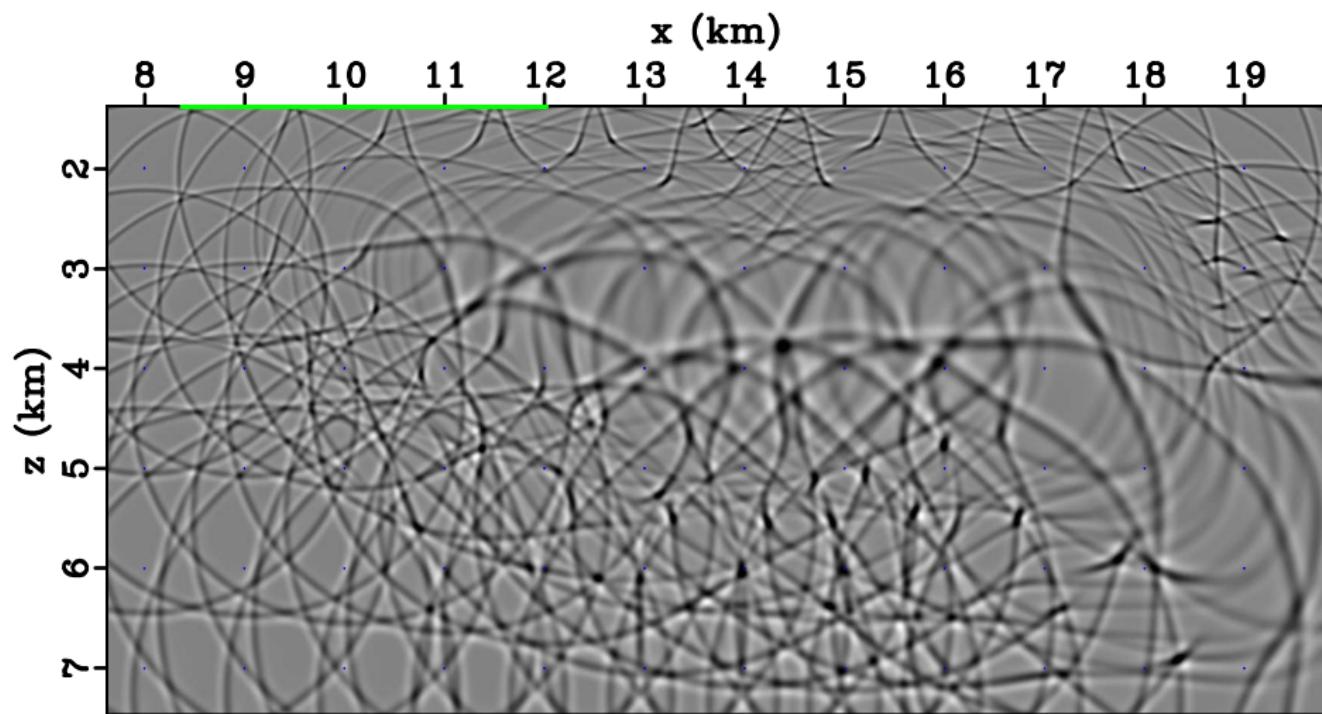
# plot wavefield frames
for i in range(10):
    tag = '-%02d' %(i*3)
    fdmod.wframe('wfld'+tag,
                 'wfld',i*3,'pclip=99',par)
    Result('wfld'+tag,
           ['wfld'+tag,'ss','rr'],'Overlay')
```

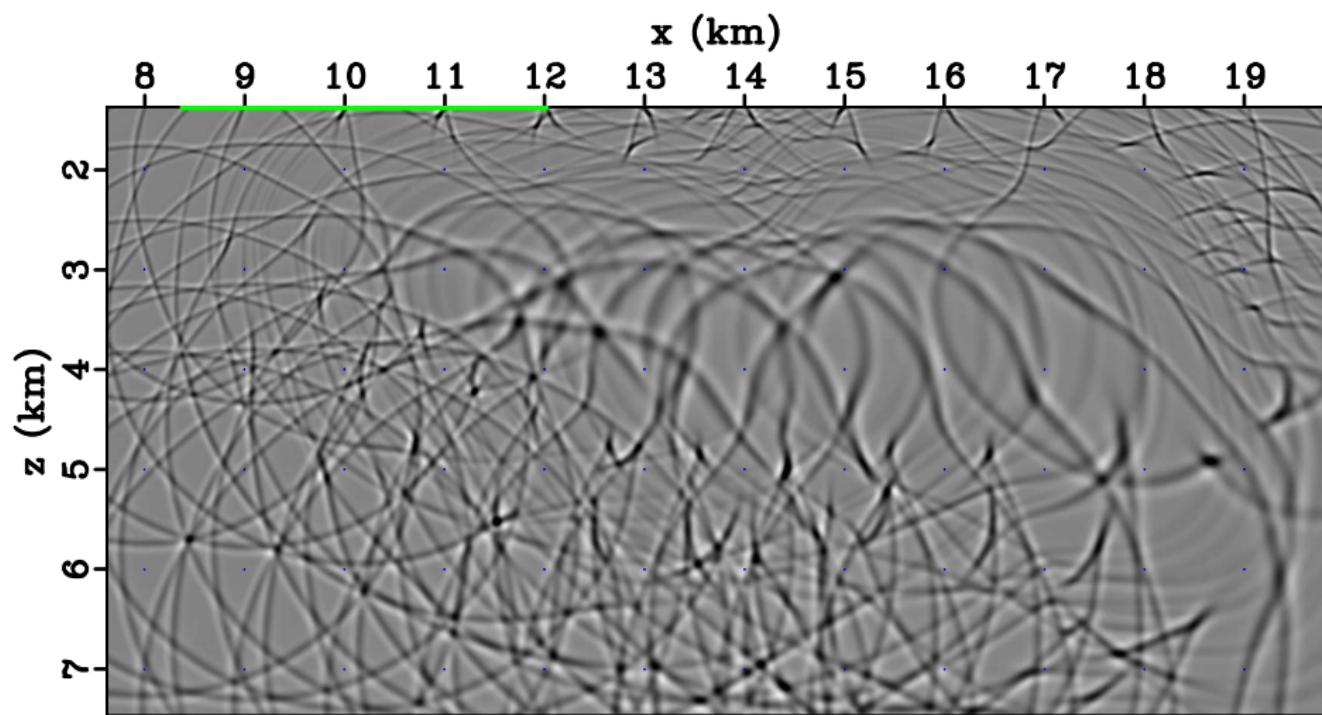


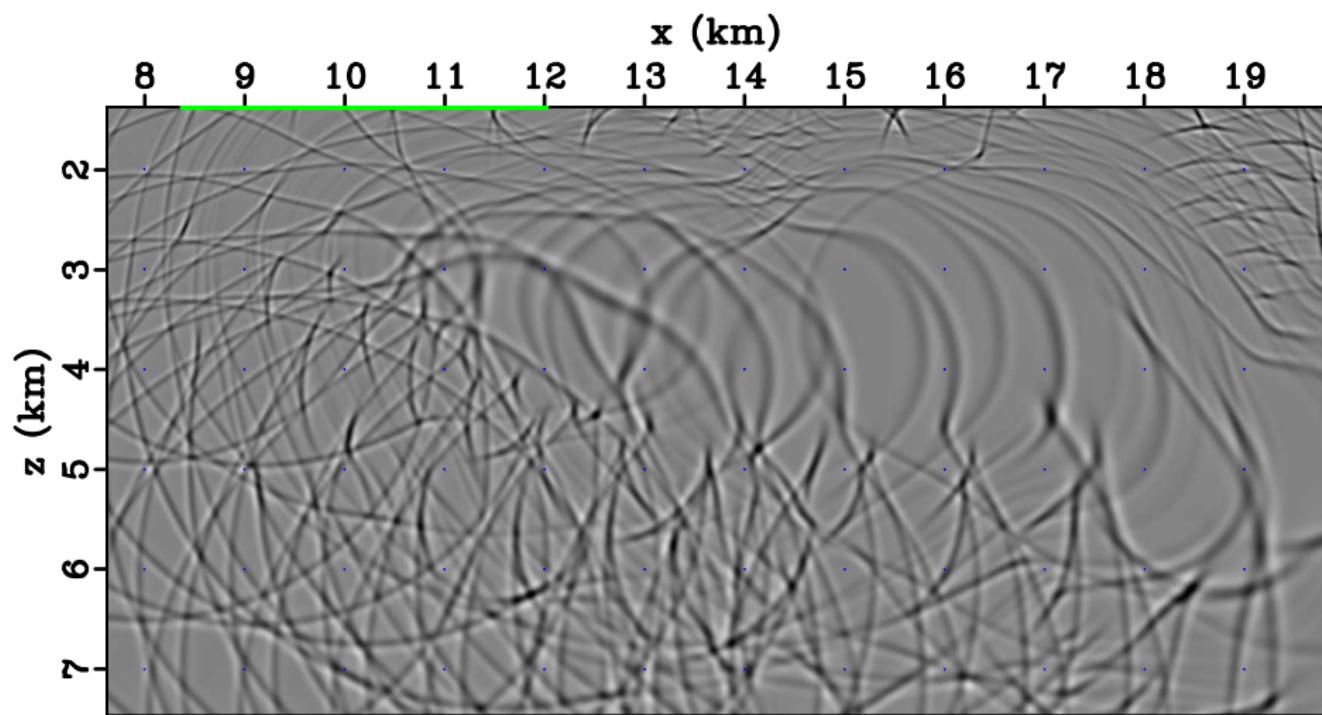


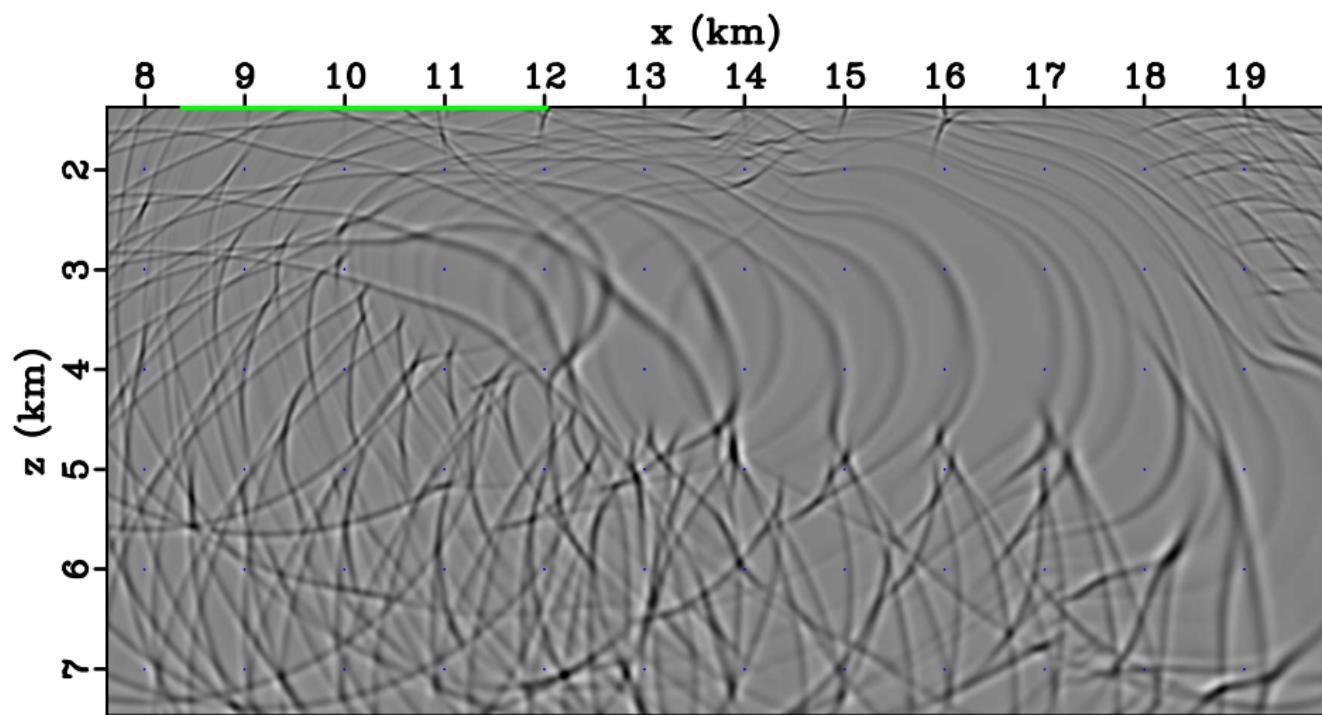


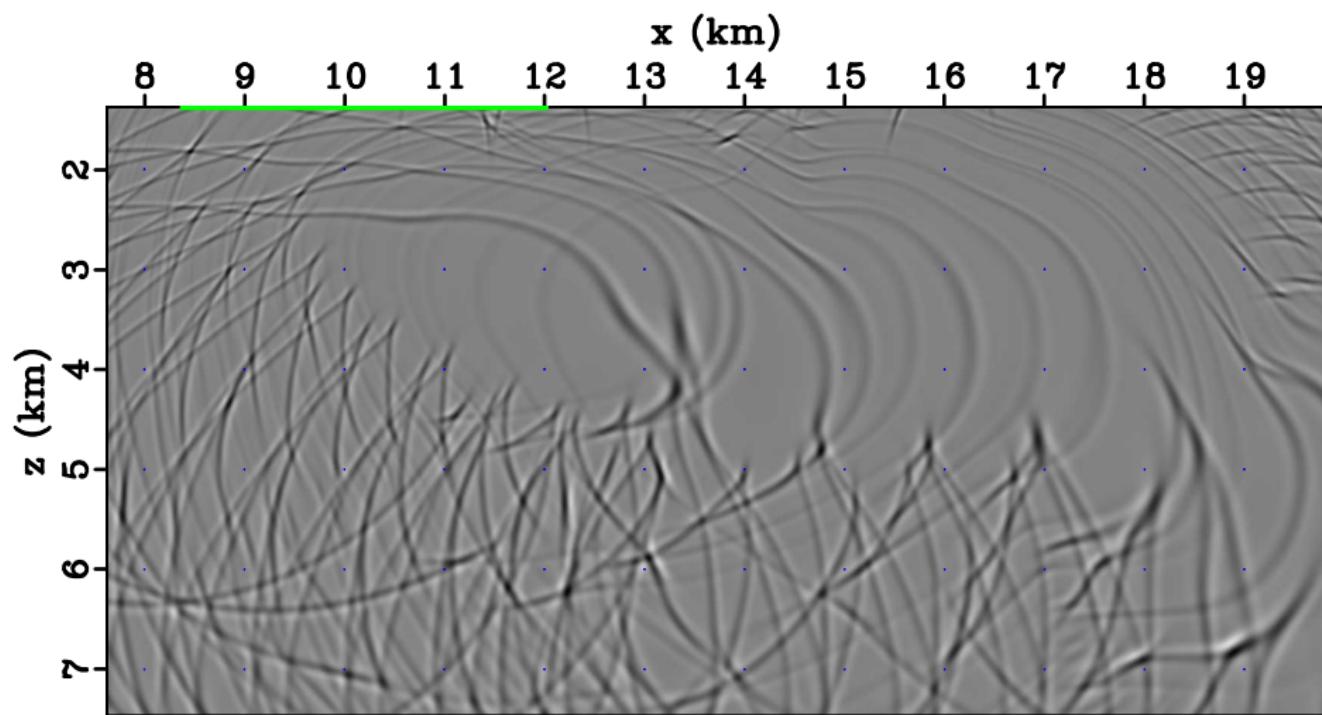


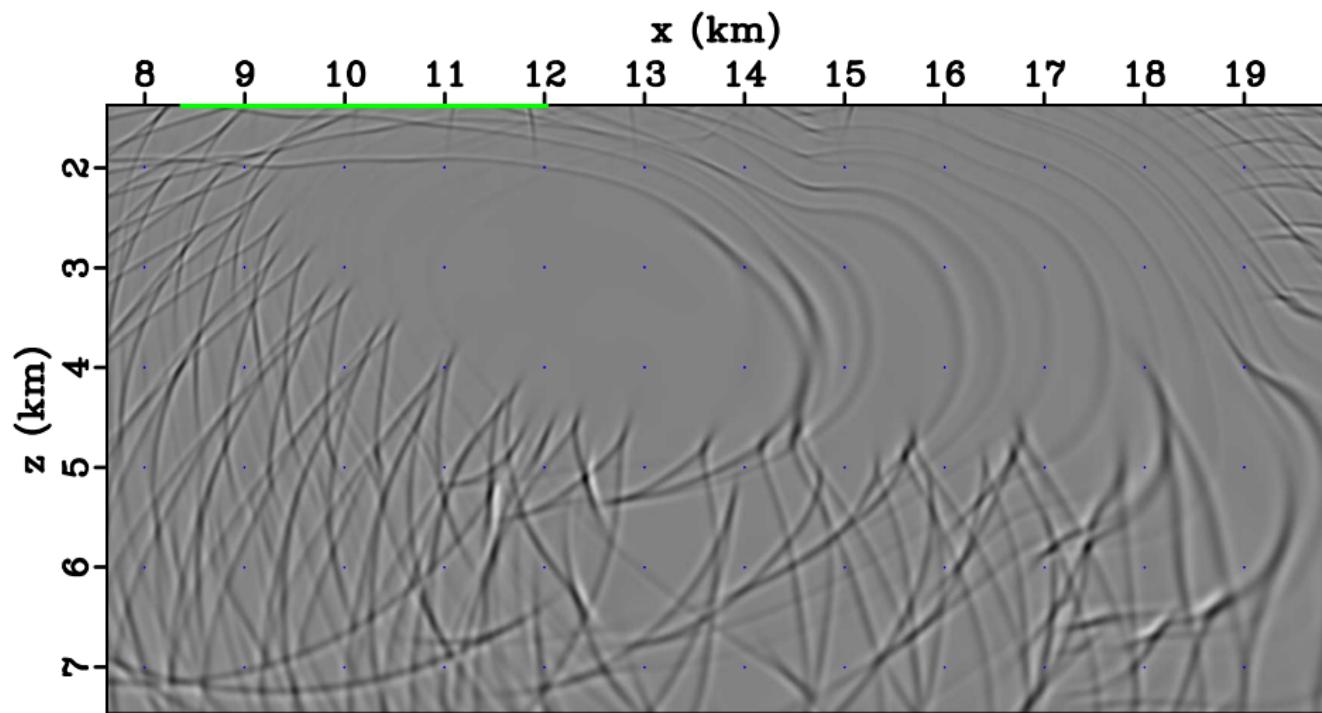








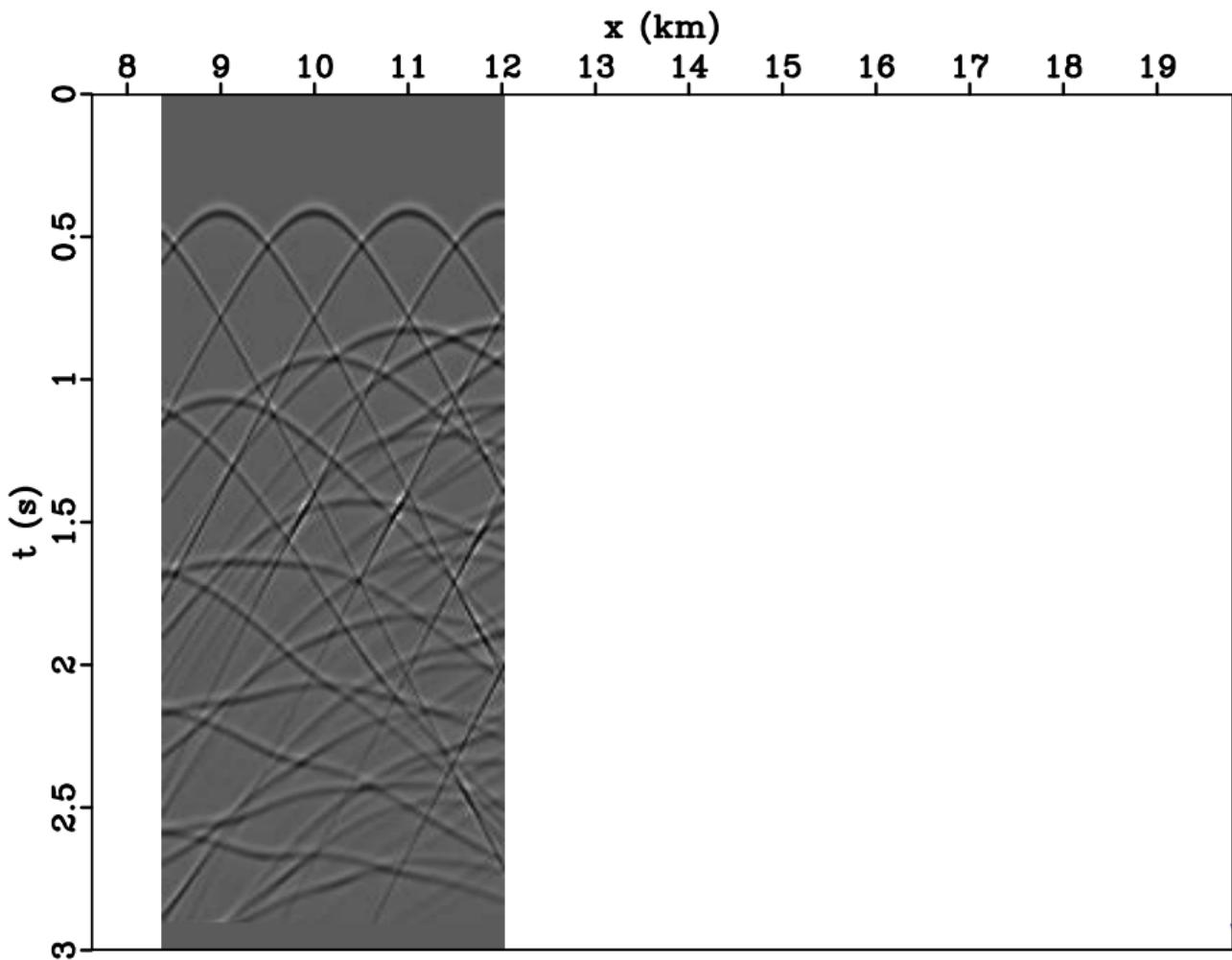




plot data

```
# undo wavelet delay
Flow( 'data' , 'temp' ,
      ,
      window squeeze=n f2=%(kt)d | 
      pad end2=%(kt)d | 
      put o2=%(ot)g
      ''' %par)

# plot data
Result( 'data' , 'window j2=4 | transp | '
      + fdmod.dgrey( ' ', par))
```

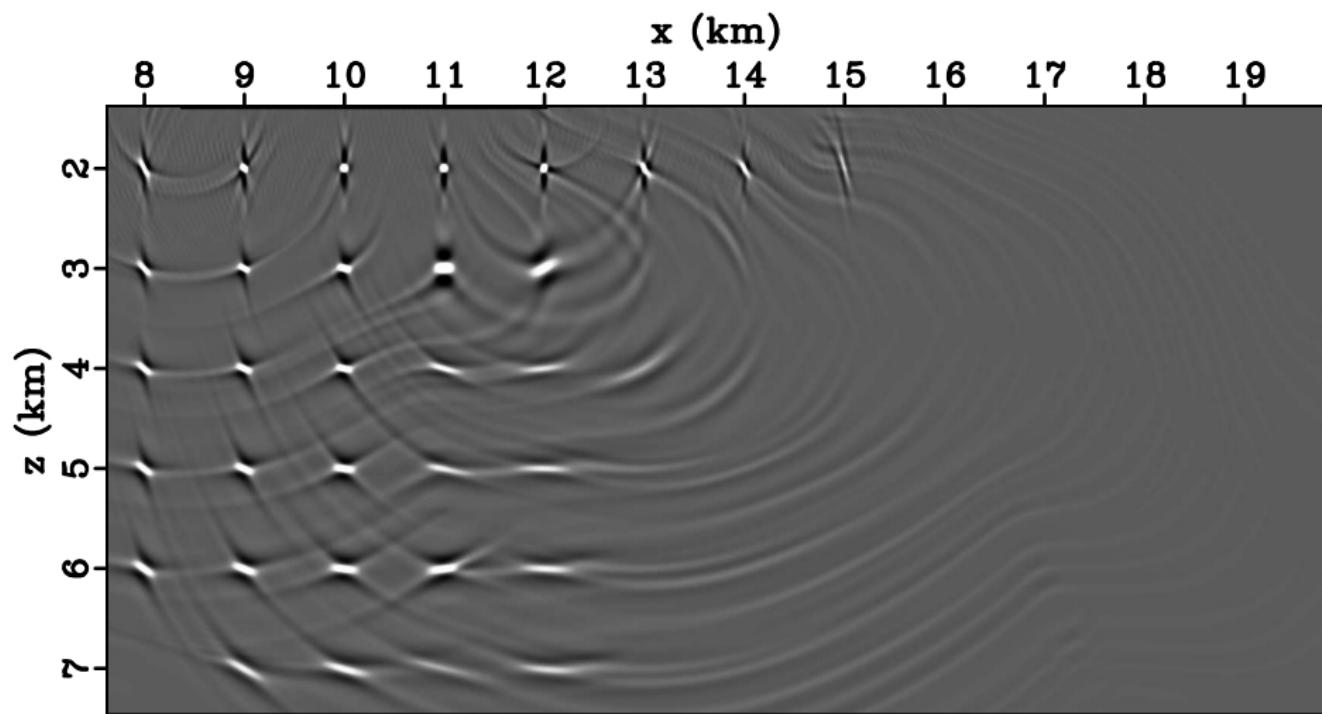


FD migration

```
# run FD migration
fdmod.zom( 'imag' , 'jdat' ,
            'data' , 'velo' , 'dens' ,
            'rr' , 'rr' ,
            'free=n' , par )
```

plot image

```
# plot image
Plot( 'imag' , 'bandpass flo=2 | '
      + fdmod.cgrey('pclip=99.9',par))
Result('imag',[ 'imag' , 'rr' ] , 'Overlay')
```



closing rules

End()

the contest

- ▶ random sources in the subsurface
- ▶ random receivers on the surface

<http://reproducibility.org>

