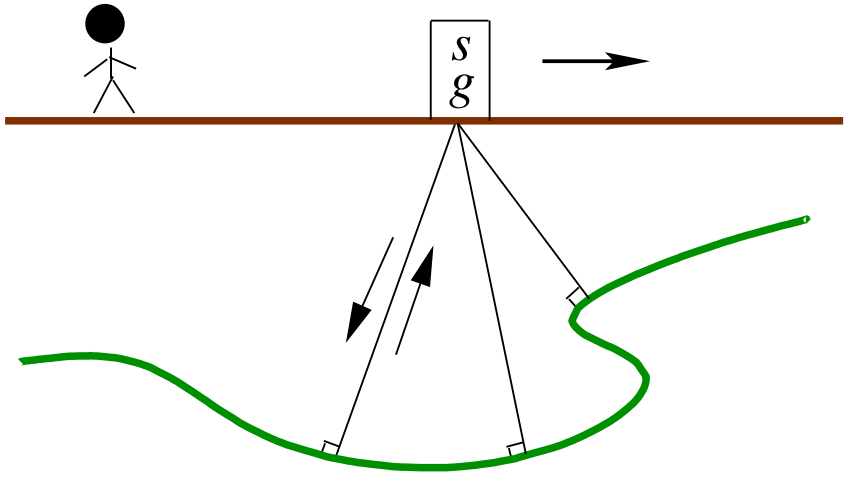
MADAGASCAR
seismic imaging tutorial

Paul Sava

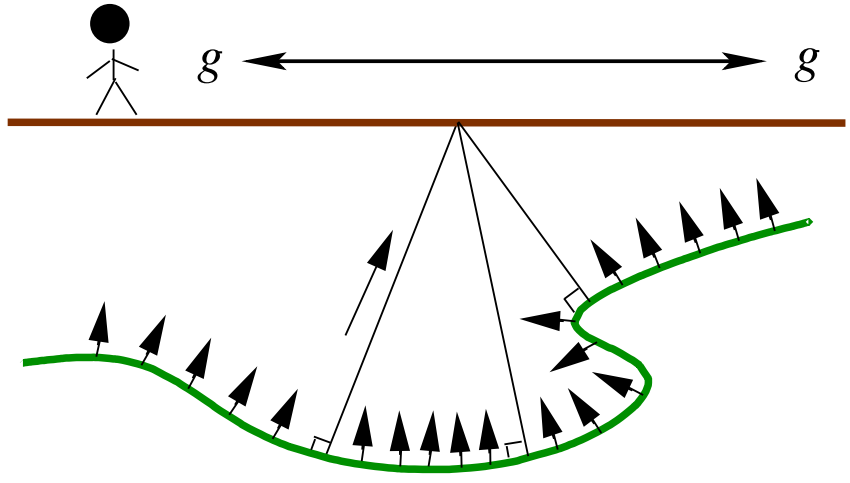
Center for Wave Phenomena
Colorado School of Mines
psava@mines.edu

exploding reflector model

reflectors are treated as collections of simultaneous sources



Zero−offset Section



Exploding Reflectors

acoustic wave propagation

$$\frac{1}{v^2}\frac{\partial^2 W}{\partial t^2} - \rho\nabla\cdot\left(\frac{1}{\rho}\nabla W\right) = f$$

- $W(\mathbf{x}, t)$: acoustic wavefield
- $v(\mathbf{x})$: velocity
- $\rho(\mathbf{x})$: density

---

modeling

forward propagation

migration

backward propagation

example

```
cd $RSF/book/rsf/school/sigsbee
```

```
$EDITOR SConstruct
```

import packages

modules available in `$RSF/book/Recipes`

```
from rsf.proj import *
from rsf.recipes import fdmod
```

`fdmod.py`: *FD modeling and plotting functions*

setup parameter dictionary

```
f2m=0.3048      # convert ft to m
par = dict(
    nx=1601,ox=25.*f2m,dx=0.025*f2m,lx='x',ux='km',
    nz=801, oz=4.5*f2m,dz=0.025*f2m,lz='z',uz='km',
    nt=5001,ot=0.0,    dt=0.0009,   lt='t',ut='s',
    jsnap=200, # snapshot jump
    kt=100,    # wavelet delay (samples)
    nb=100     # boundary (grid points)
    )
```

setup plotting parameters

```
fdmod.param(par)
```

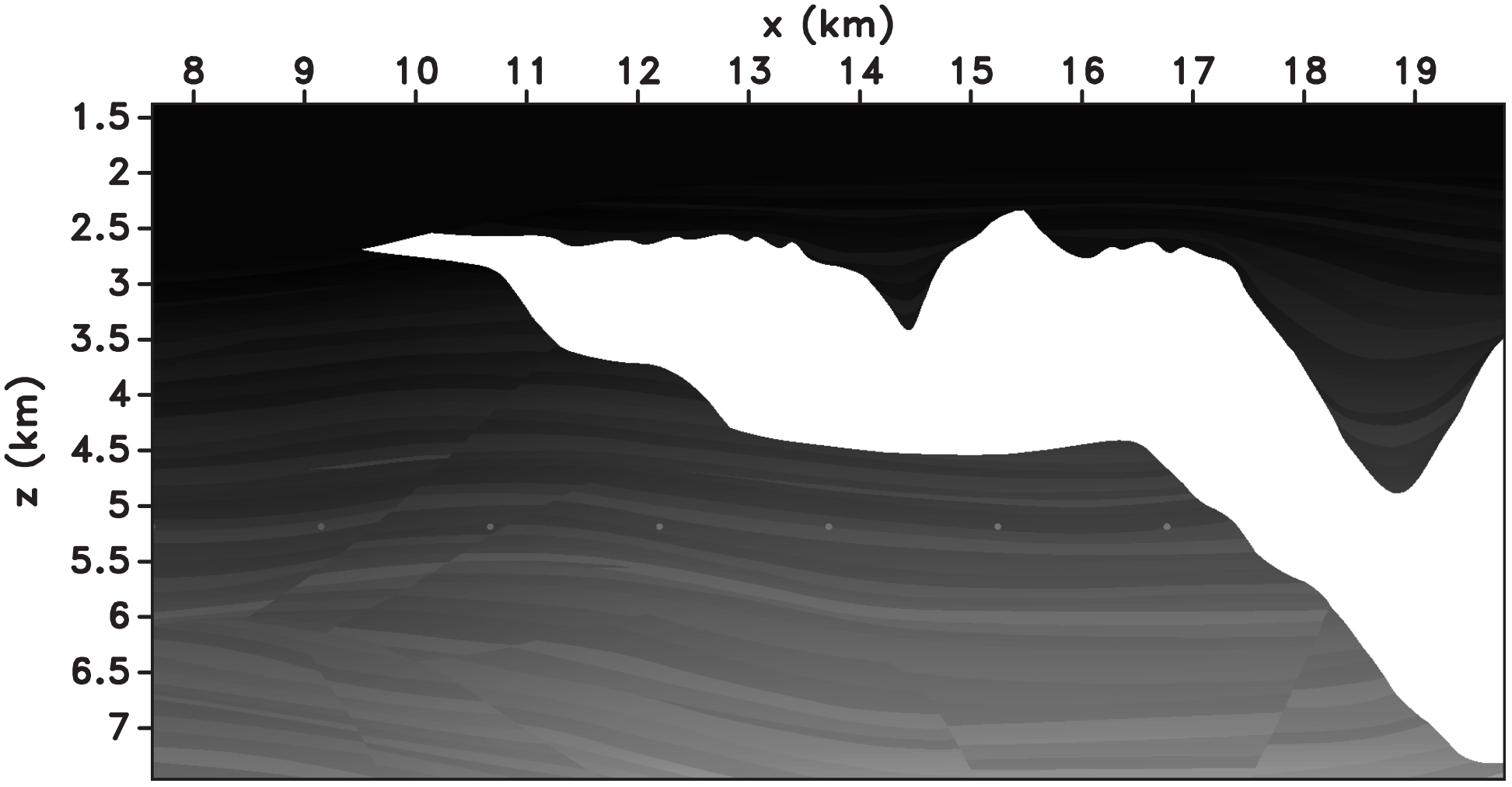
fetch the model

```
strvelfile = 'sigsbee2a_stratigraphy.sgy'
Fetch(strvelfile,'sigsbee')
Flow('vraw',strvelfile,'segyread read=data')
```

setup the model

```
Flow('modl','vraw',
     '''
     scale rscale=%g |
     put o1=%g d1=%g o2=%g d2=%g |
     window n1=%d min1=%g n2=%d min2=%g
     ''' % (0.001*f2m,
            0       ,0.025*f2m,
            10*f2m,0.025*f2m,
            par['nz'],par['oz'],
            par['nx'],par['ox']
            ))
Plot   ('modl',fdmod.cgrey('mean=y',par))
Result('modl',fdmod.cgrey('mean=y',par))
```

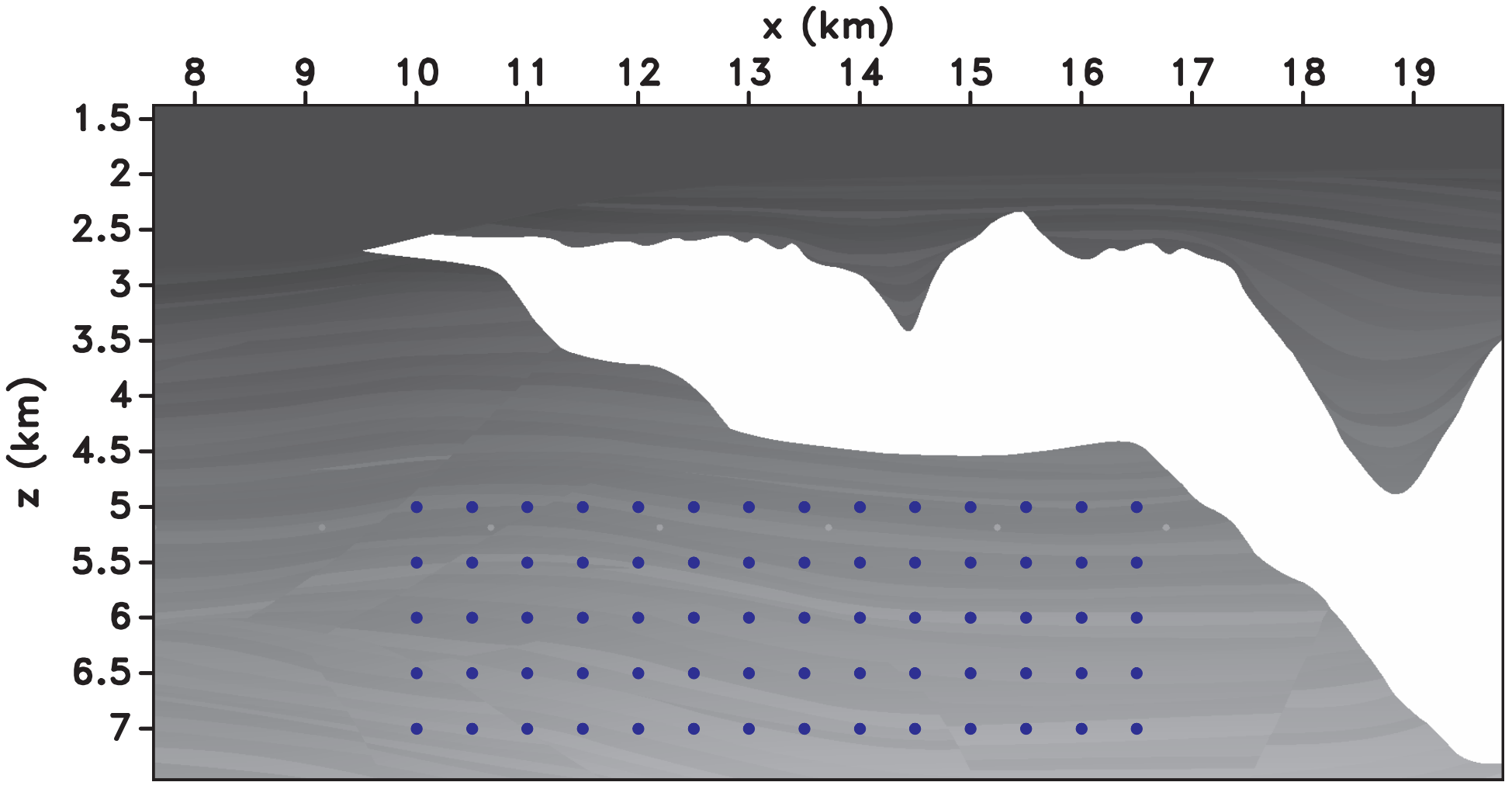
scons modl.view



make the source array

```
fdmod.boxarray('ss',
               5,   # vertical number
               5,   # vertical origin
               0.5, # vertical sampling
               14,  # horizontal number
               10,  # horizontal origin
               0.5, # horizontal sampling
               par)
Plot(  'ss',fdmod.ssplot('',par))
Result('ss',['modl','ss'],'Overlay')
```

scons ss.view



make the horizontal receiver array

```
fdmod.horizontal('tH',1.5,par)
par['jrH']=10    # jump (grid points)
par['orH']=14.0  # origin
par['nrH']=75    # number
```

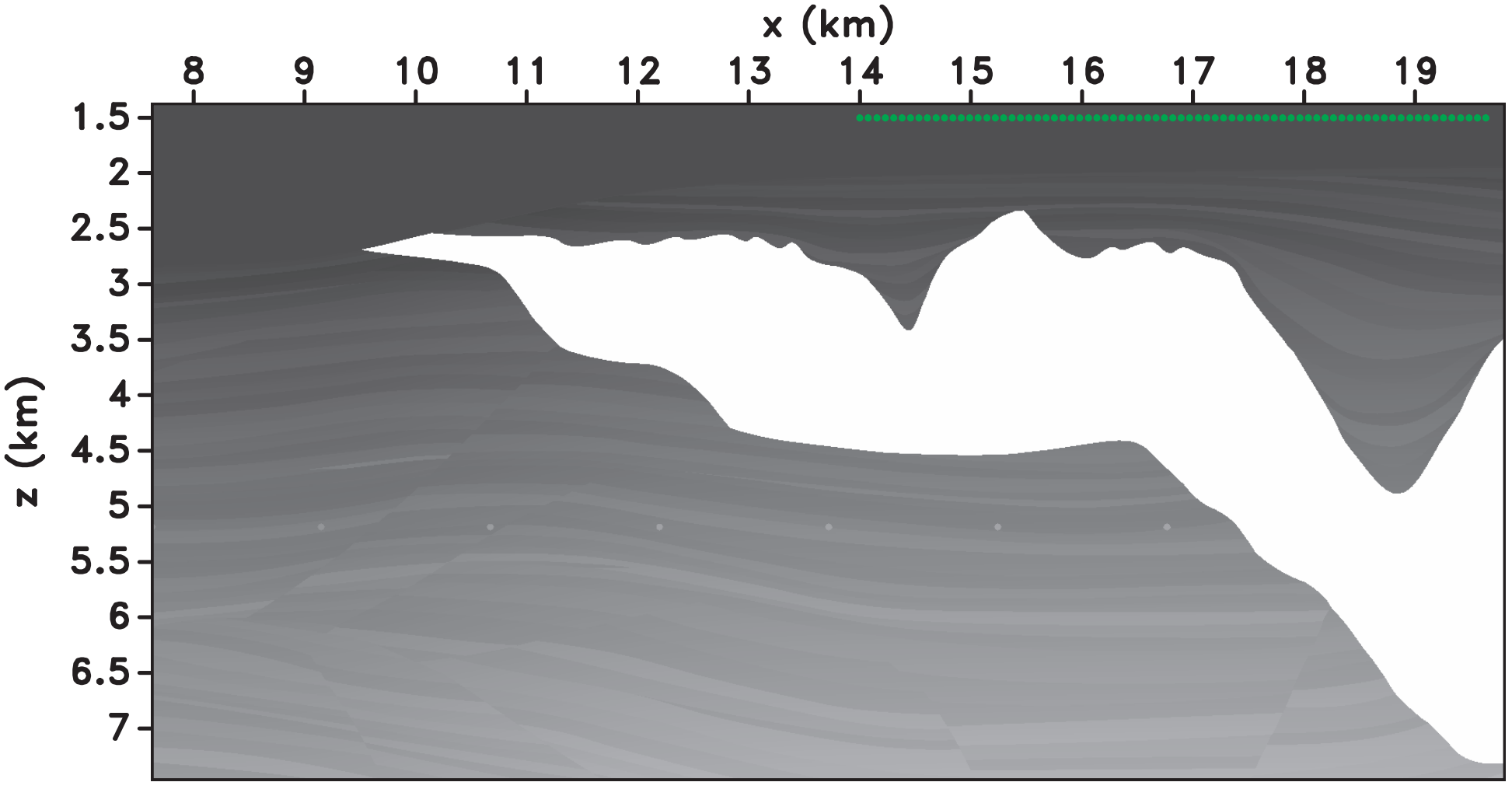
make the vertical receiver array

```
fdmod.vertical('tV',8.5,par)
par['jrV']=20    # jump (grid points)
par['orV']=2.5   # origin
par['nrV']=25    # number
```

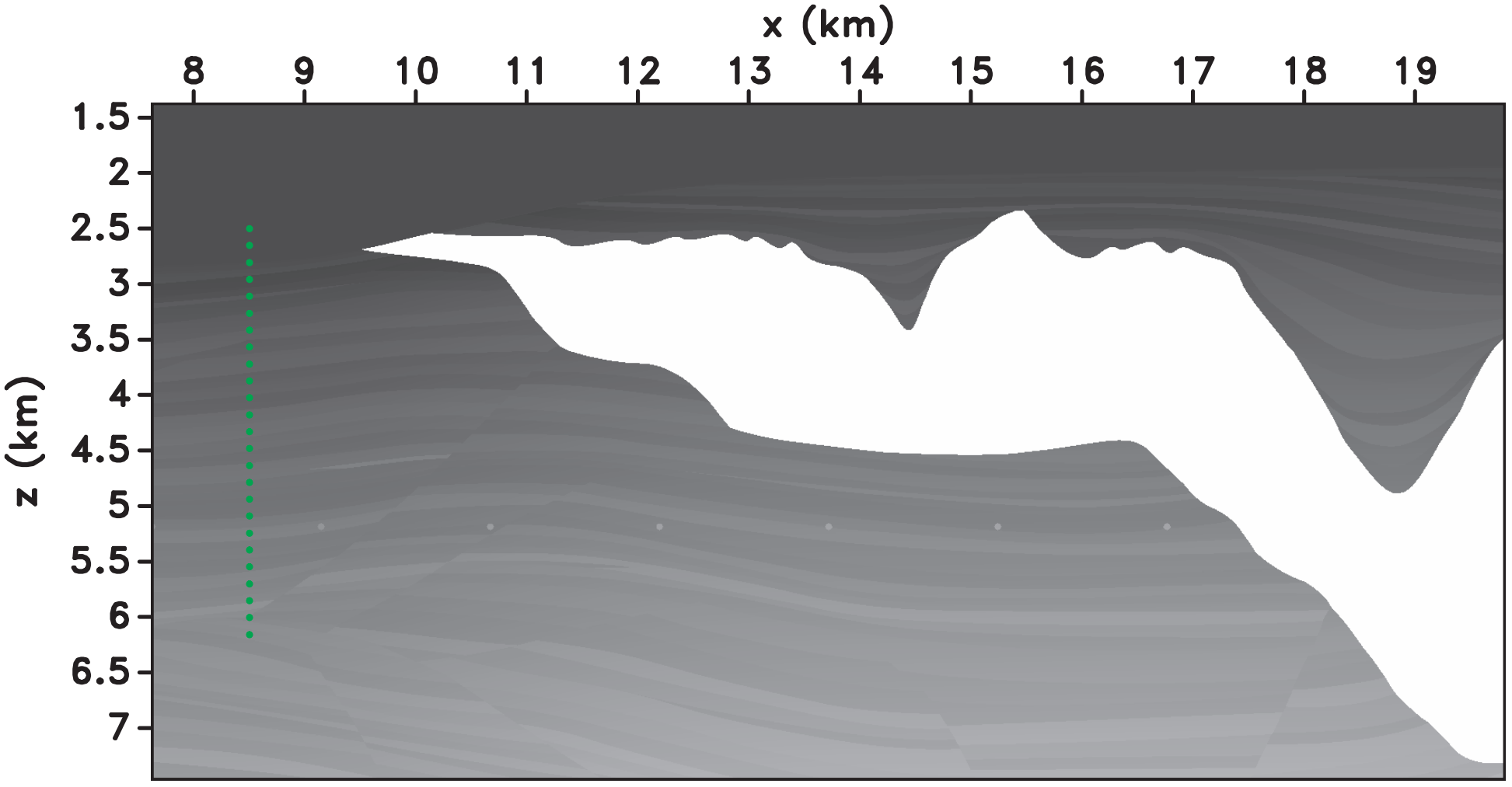
plot the receiver arrays

```
for j in ('H','V'):

    # window array
    Flow('r'+j,'t'+j,
         'window j2=%d min2=%g n2=%d'%
         (par['jr'+j],   # jump
          par['or'+j],   # origin
          par['nr'+j])) # number

    # plot array
    Plot(   'r'+j,fdmod.rrplot('',par))
    Result('r'+j,['modl','r'+j],'Overlay')
```

scons rH.view



scons rV.view



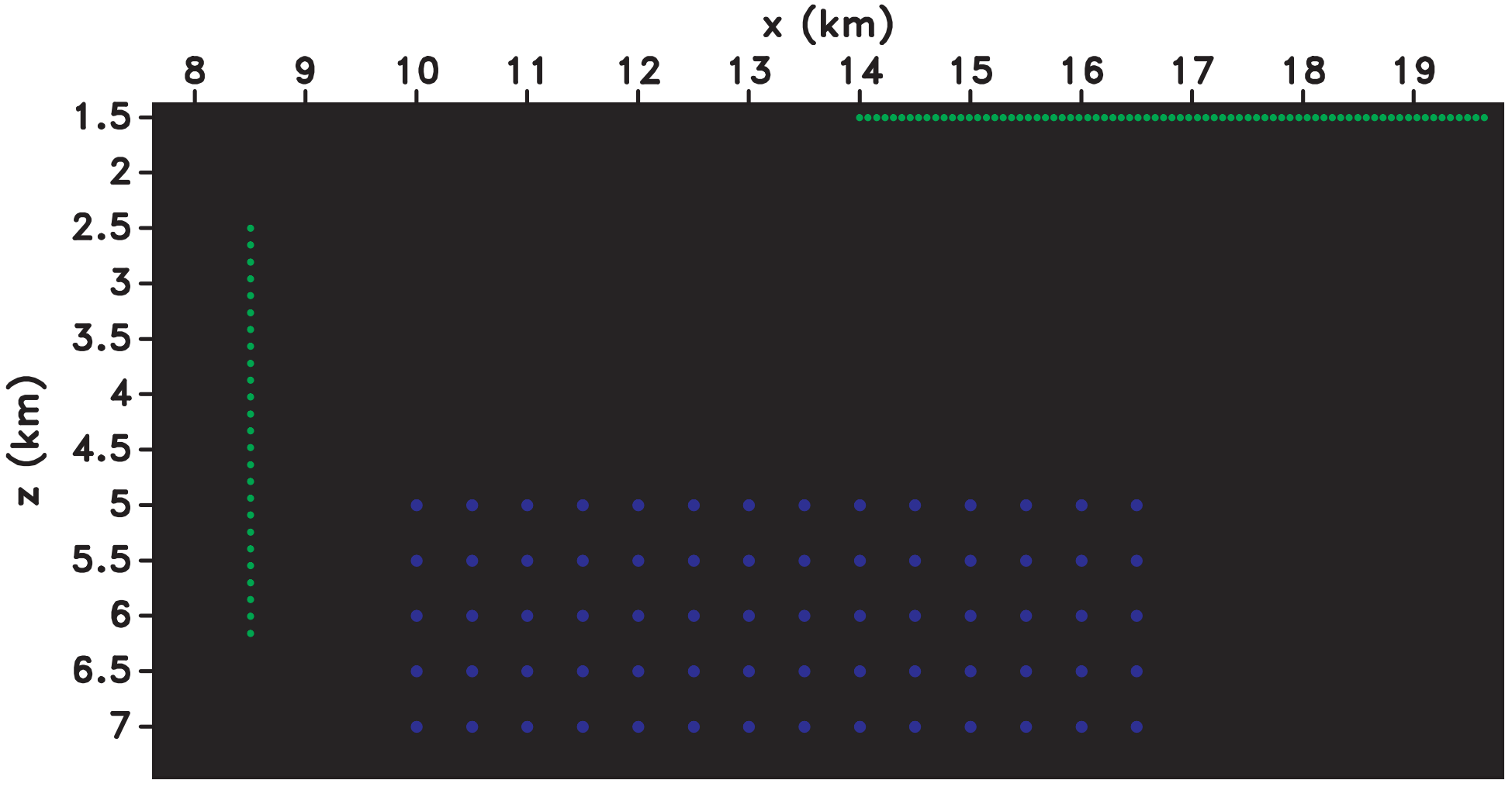
group the receivers in one array

```
Flow('rA',['rH','rV'],
     'cat axis=2 space=n ${SOURCES[1]}')
Plot('rA',['rH','rV'],'Overlay')
```

make the density

```
Flow('dens','modl',
     'math output=1')
Plot(  'dens',fdmod.cgrey('mean=y',par))
Result('dens',['dens','ss','rH','rV'],'Overlay')
```

scons dens.view



make the stratigraphic velocity

```
Flow('vstr','modl',
     'window')
```

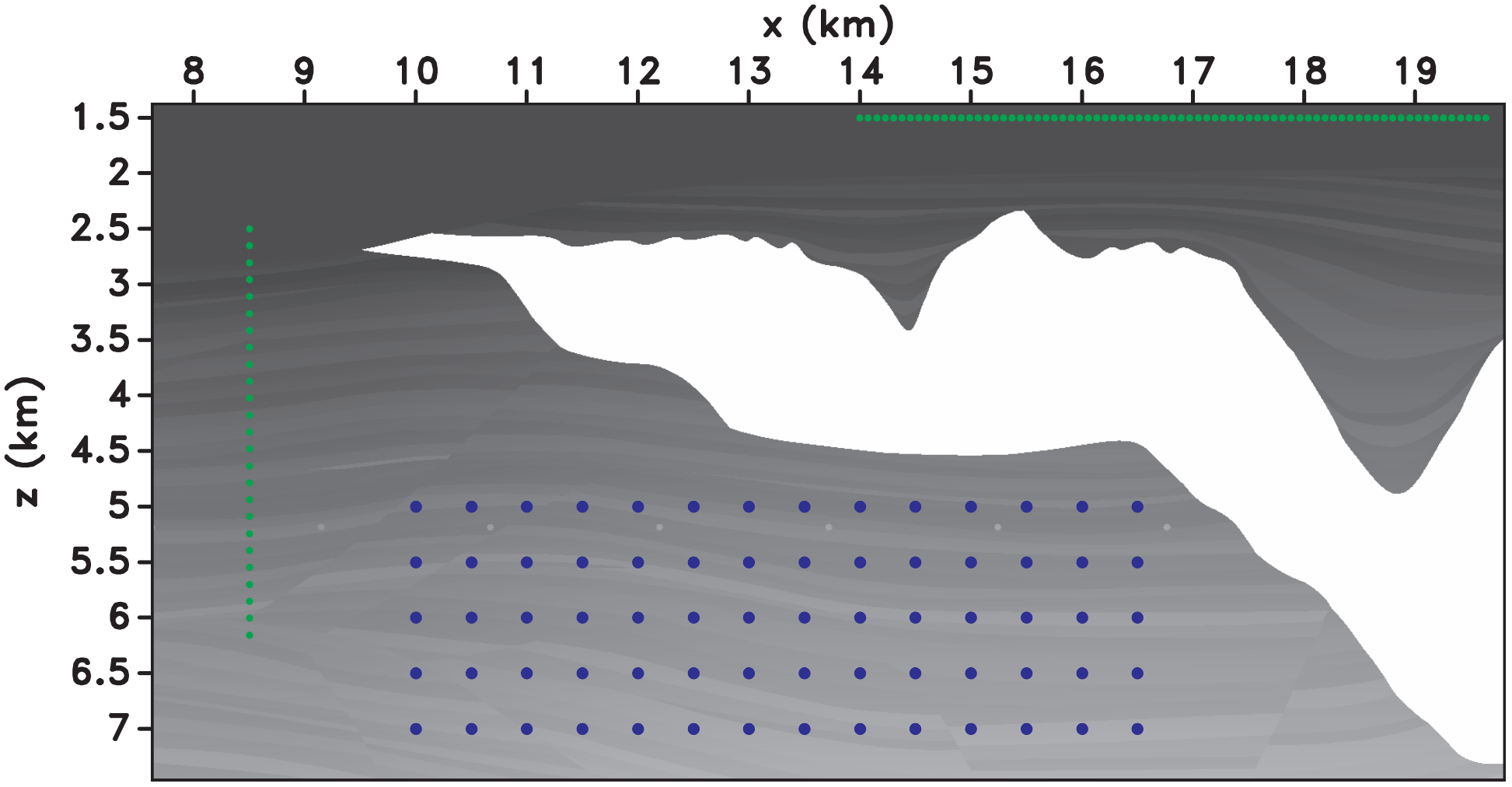
make the smooth velocity

```
Flow('vsmo','modl',
     'smooth rect1=25 rect2=25 repeat=3')
```

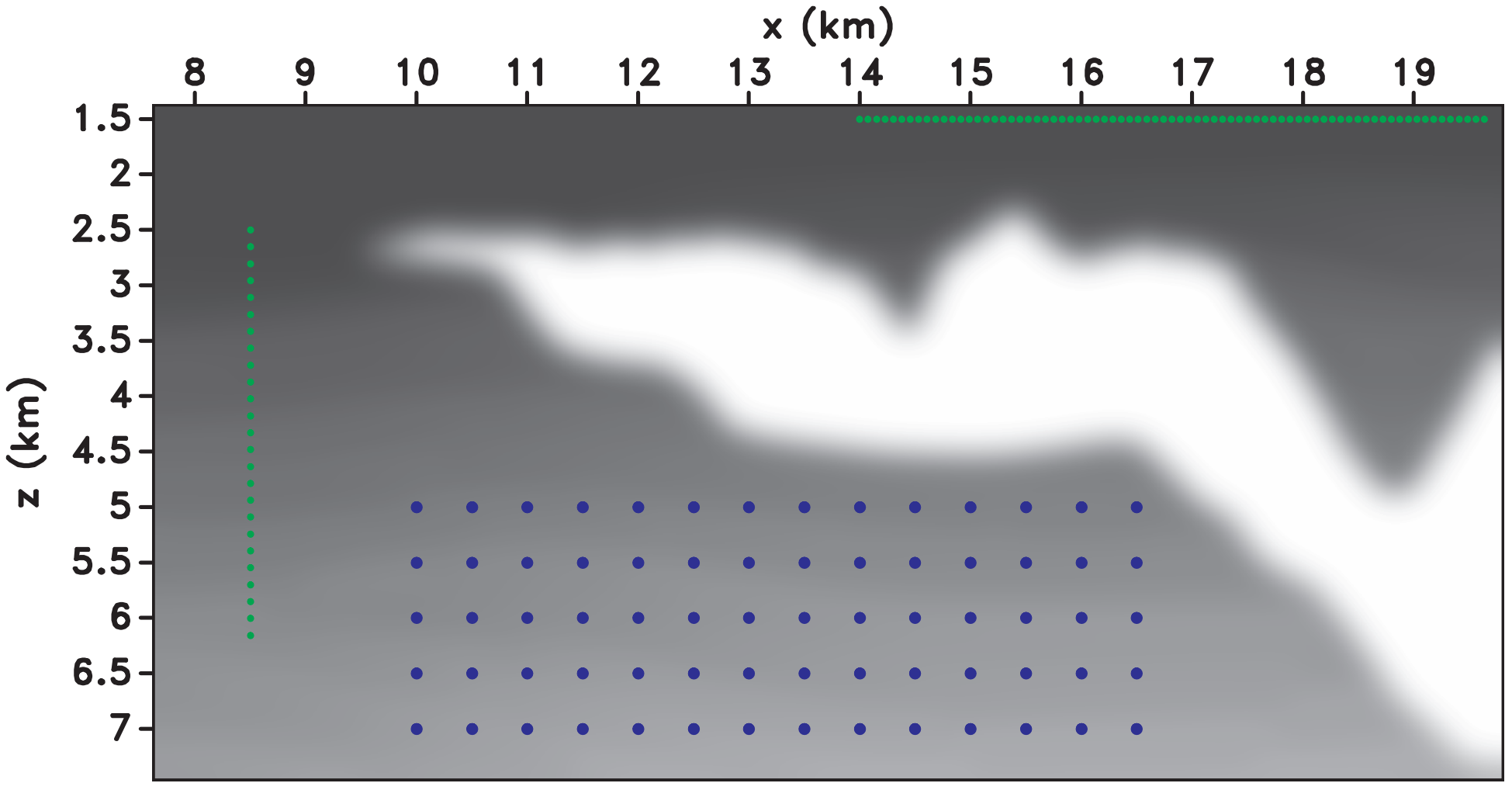
plot the velocities

```
for v in ('vstr','vsmo'):
    # plot velocities
    Plot(  v,fdmod.cgrey('mean=y',par))

    # overlay sources and receivers
    Result(v,[v,'ss','rH','rV'],'Overlay')
```

scons vstr.view



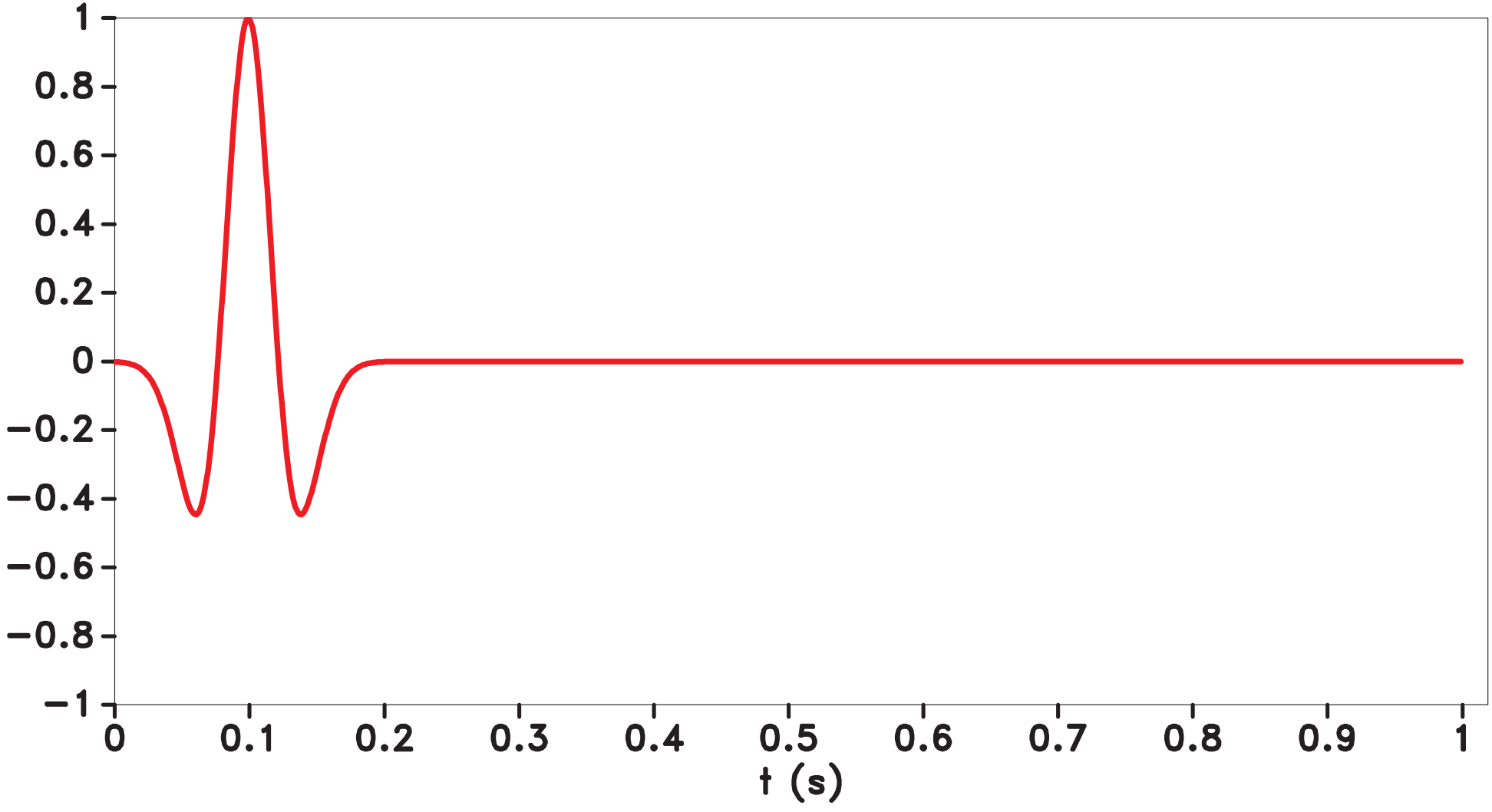
scons vsmo.view



make the wavelet

```
fdmod.wavelet('wav_',10,par)

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

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

scons wav.view



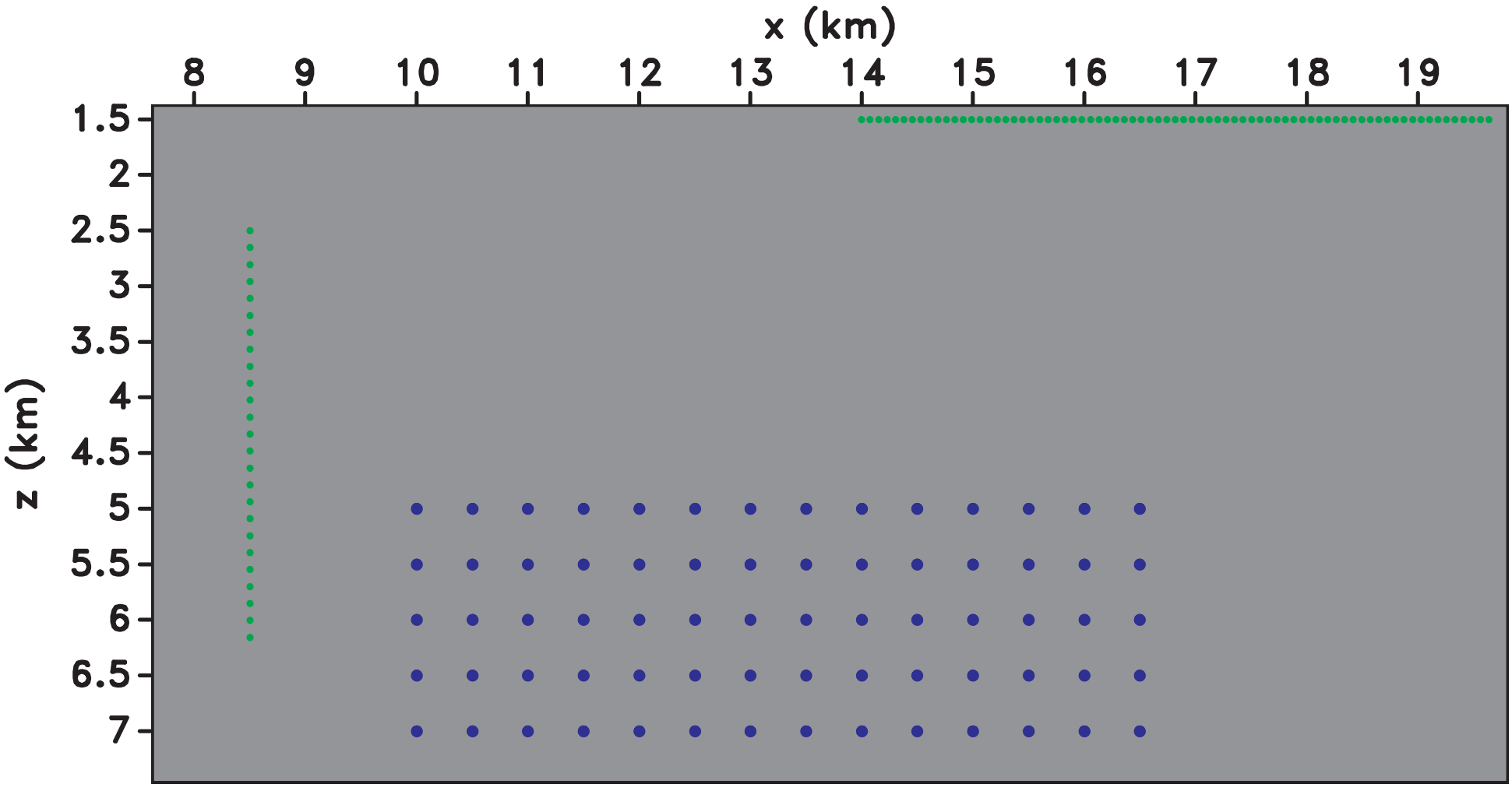
run FD modeling

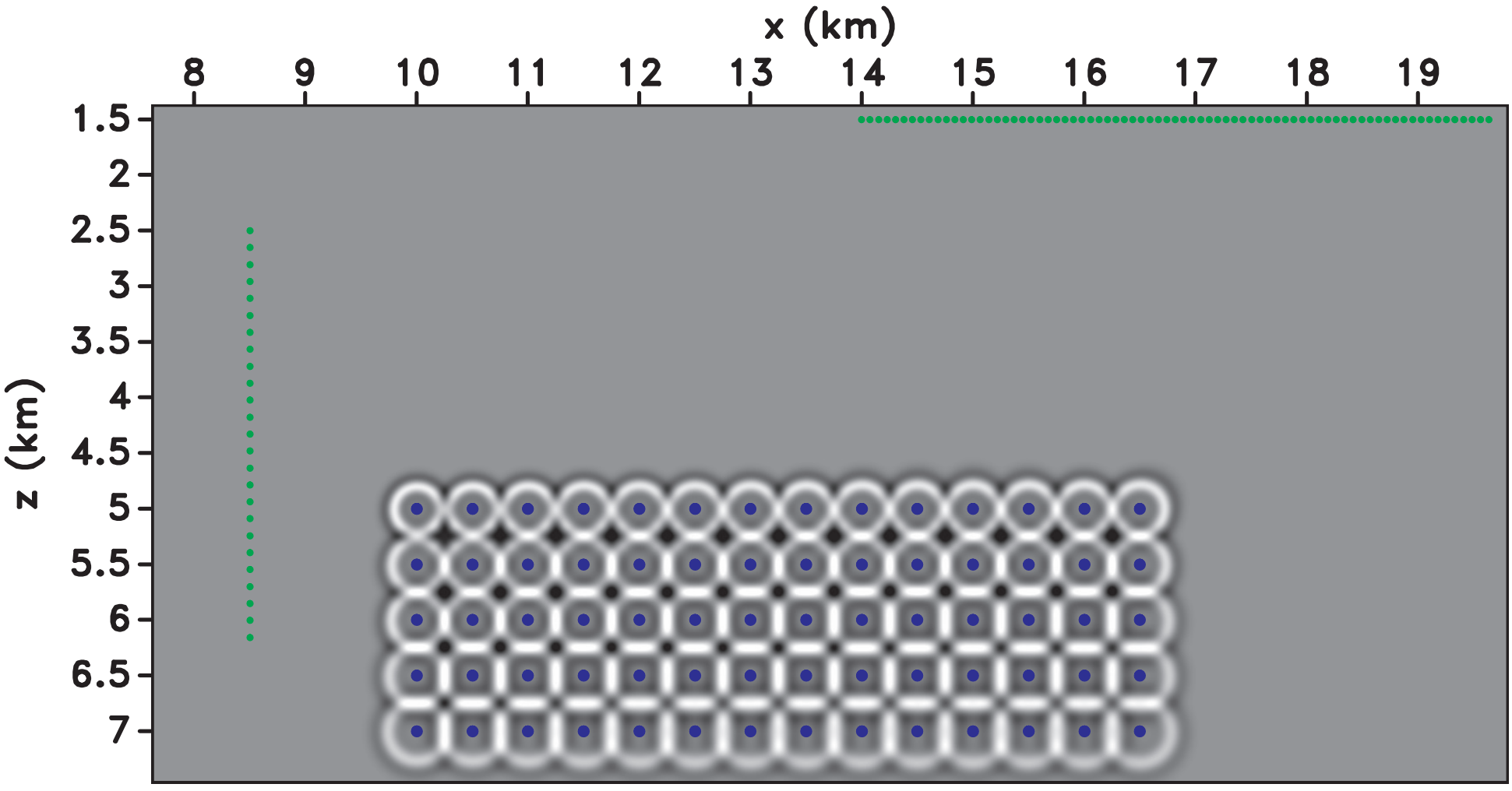
```
fdmod.awefd1('tmpA',   # data file (all receivers)
             'wfld',   # wavefield snapshots
             'wav',    # source wavelet
             'vsmo',   # velocity
             'dens',   # density
             'ss',     # source coordinates
             'rA',     # receiver coordinates
             'fsrf=n',# optional flags
             par)
```

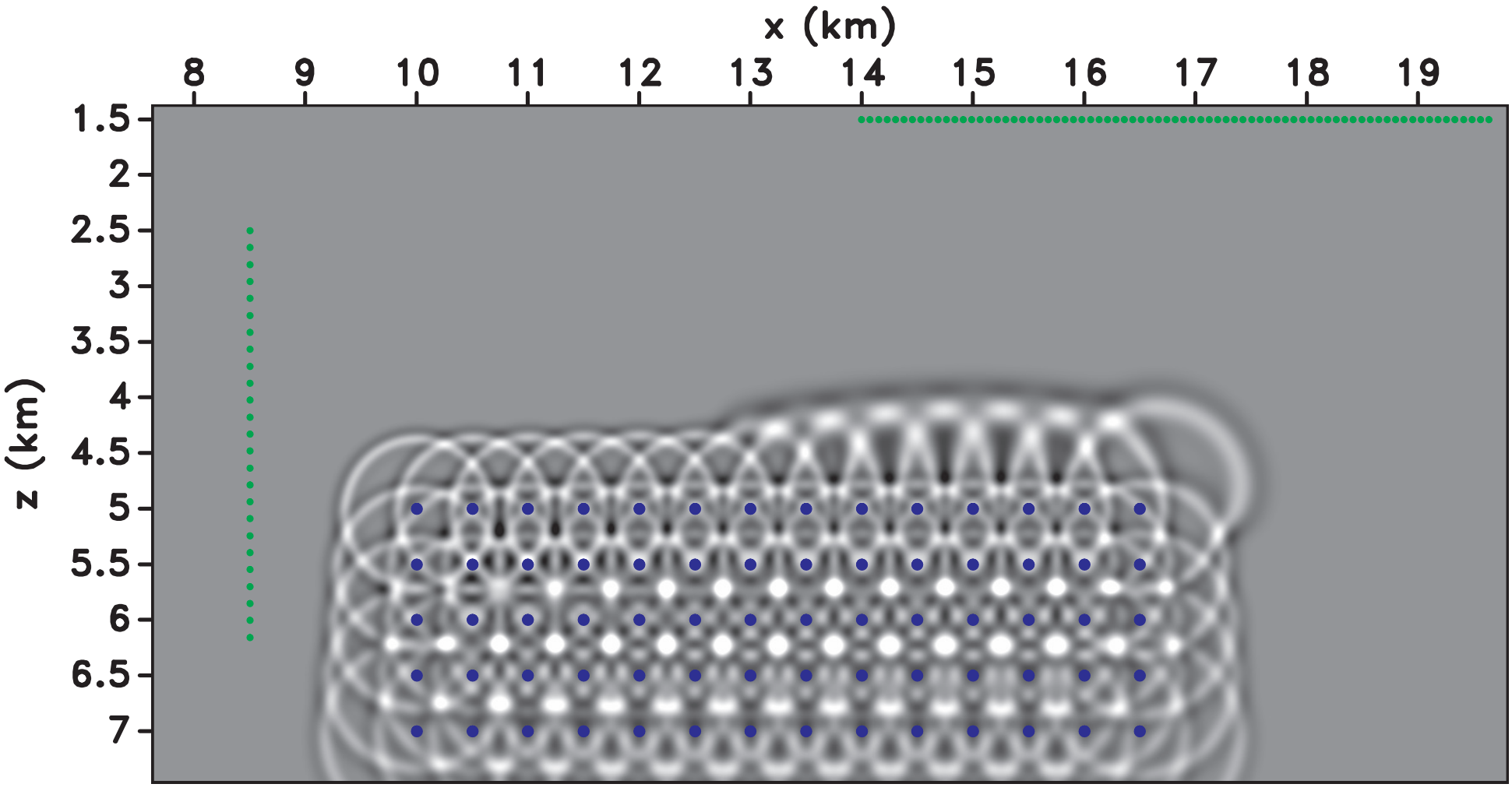
plot the wavefield

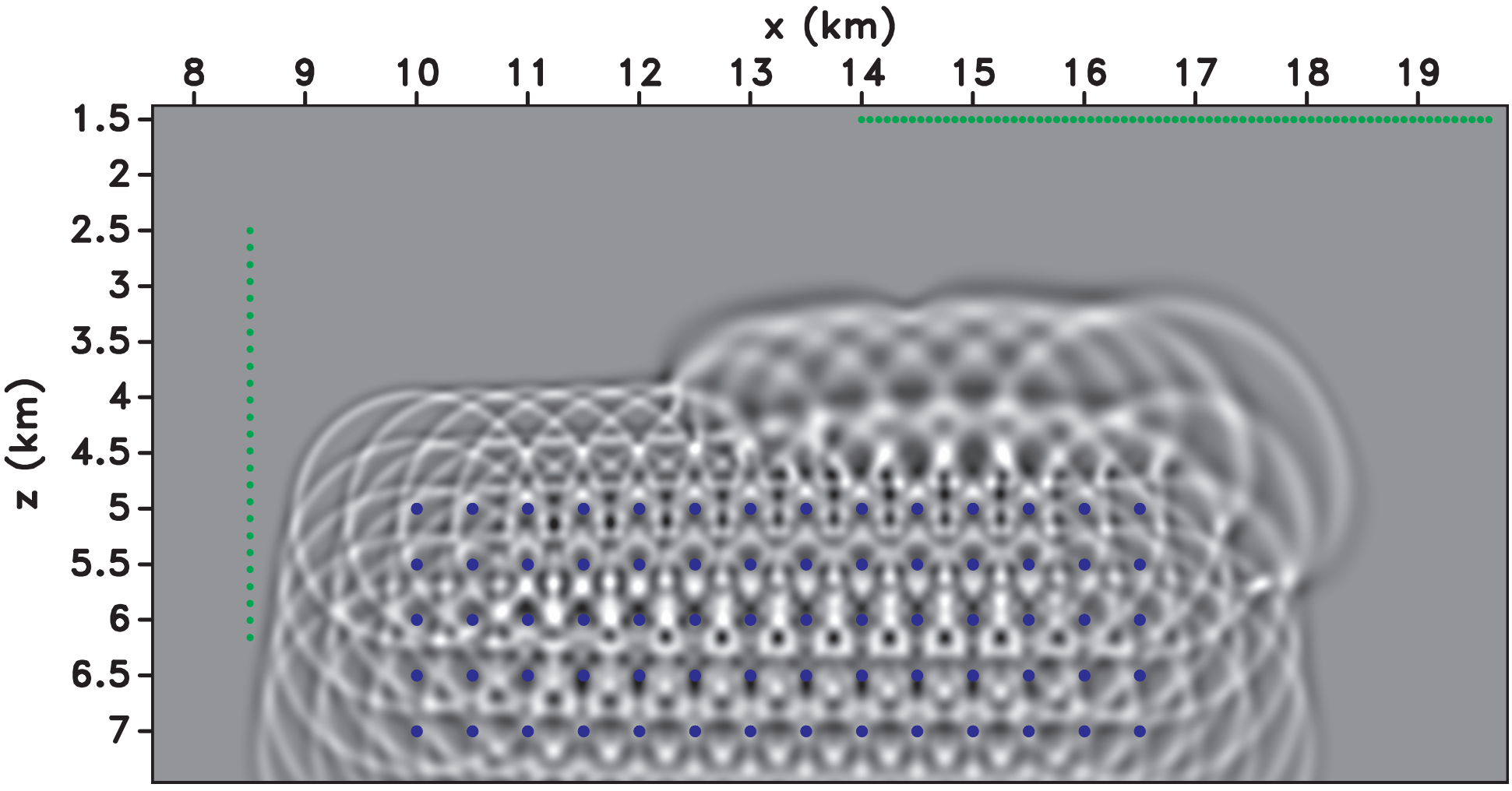
```
par['nfrm']=int((par['nt']-1)/par['jsnap'])
for i in range(par['nfrm']):
    tag = '-%02d'%i
    # plot wavefield frames
    fdmod.wframe('wfld'+tag,
                 'wfld',i,'pclip=99',par)
    Plot(  'wovl'+tag,['wfld'+tag,'ss','rH','rV'],
           'Overlay')
    Result('wfld'+tag,['wfld'+tag,'ss','rH','rV'],
           'Overlay')
Result('wfld',
       ['wovl-%02d'%i for i in range(par['nfrm'])],
       'Movie')
```

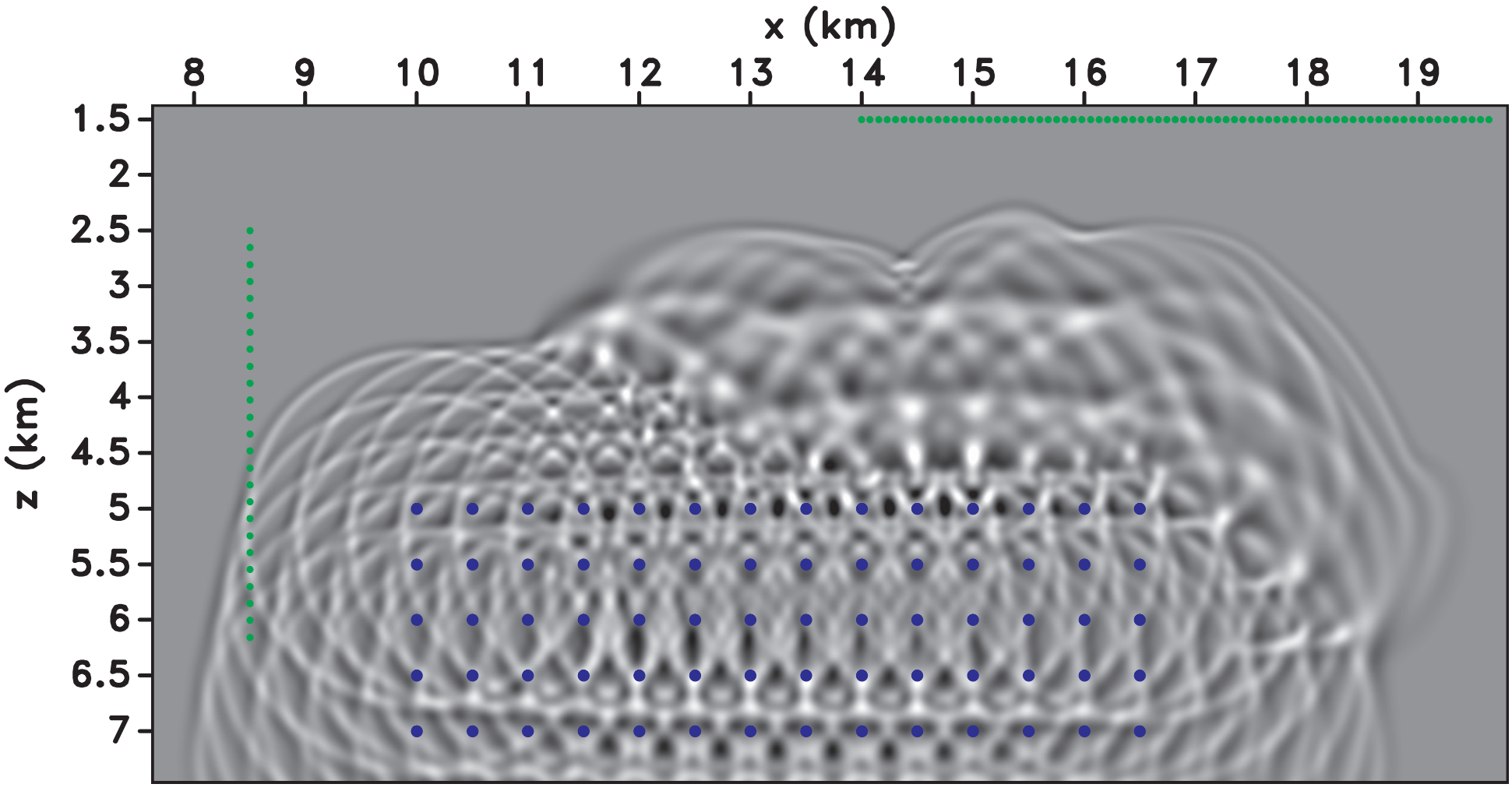
```
scons wfld.view
```

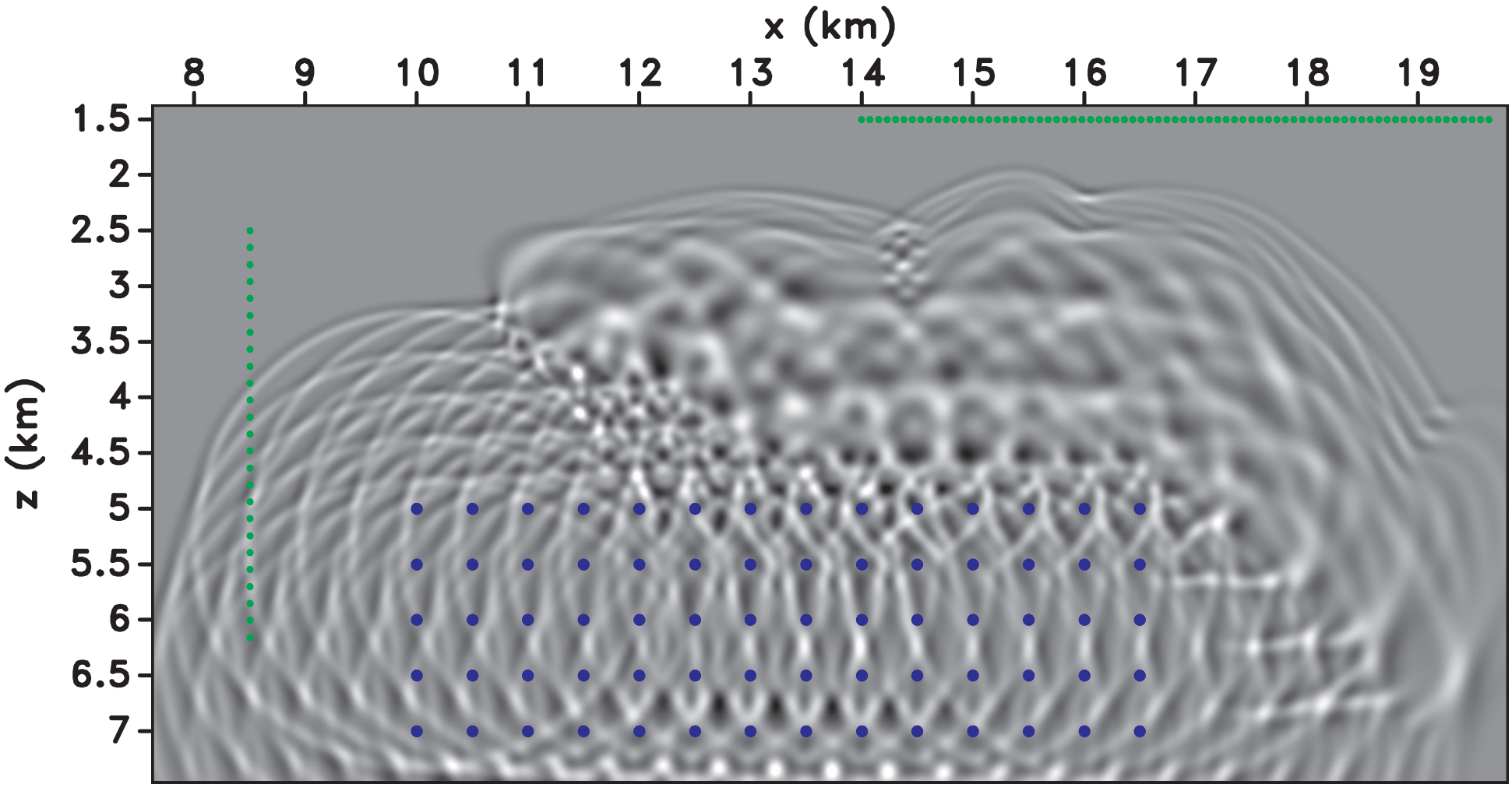

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

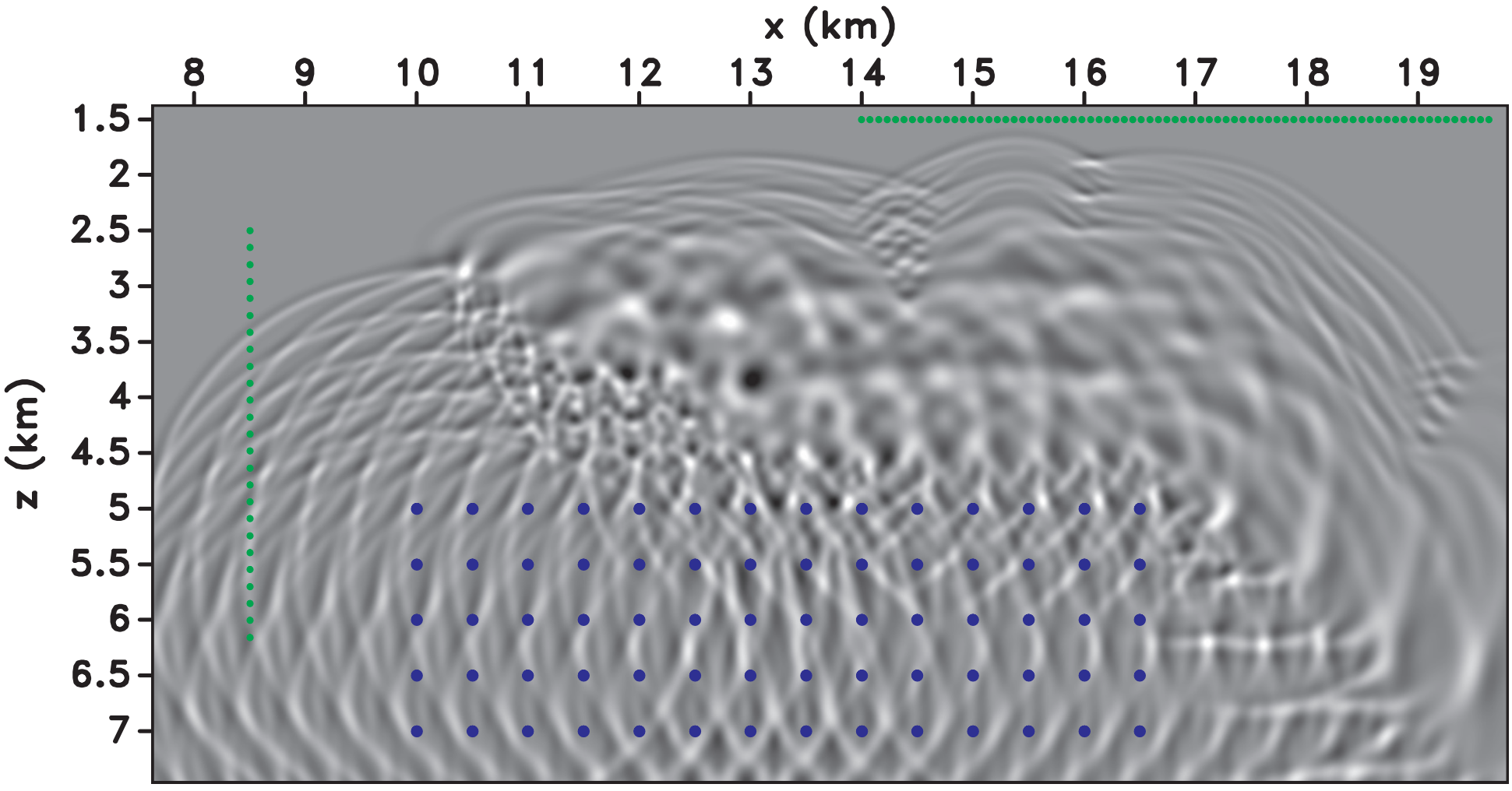

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

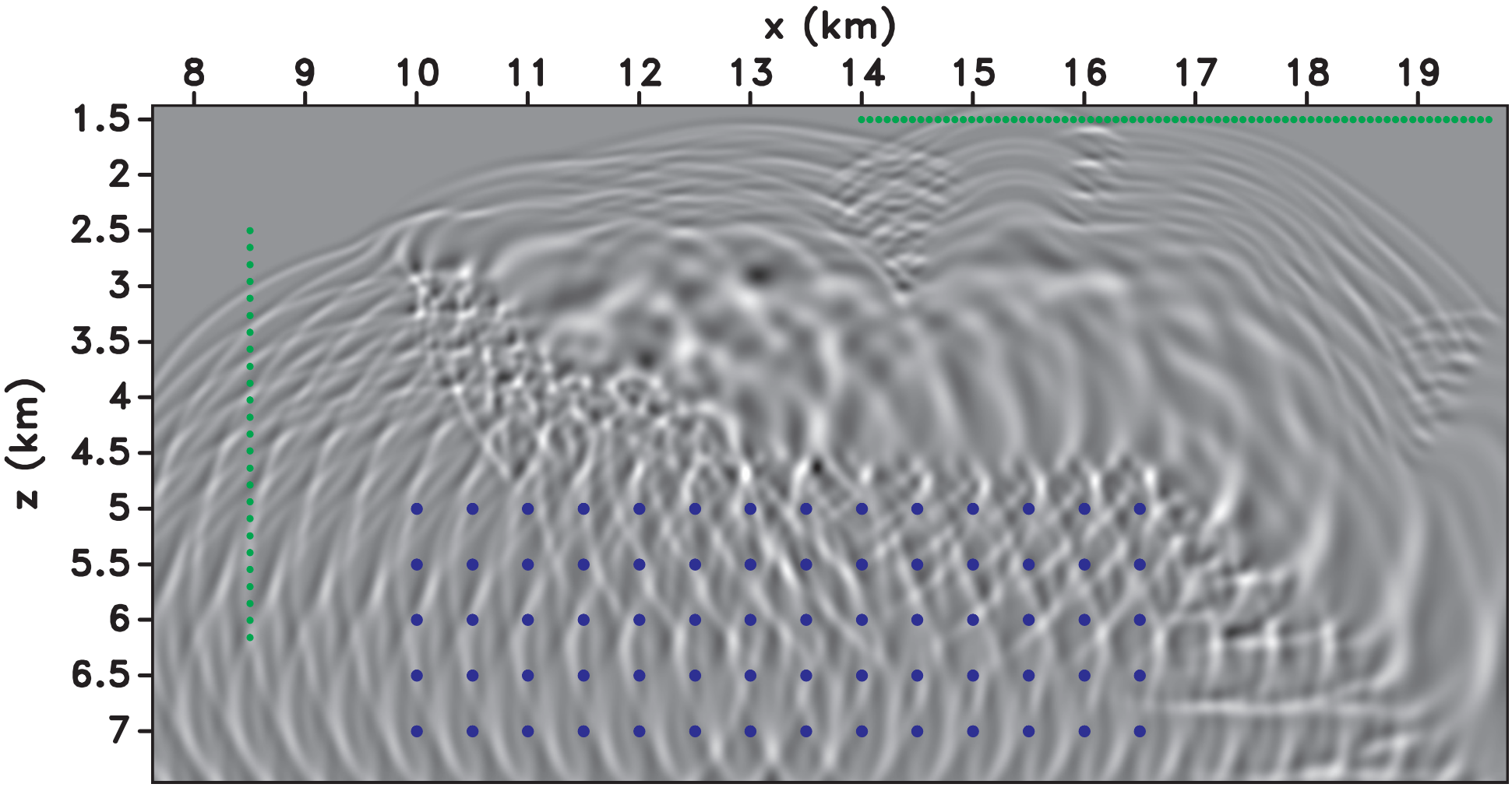

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

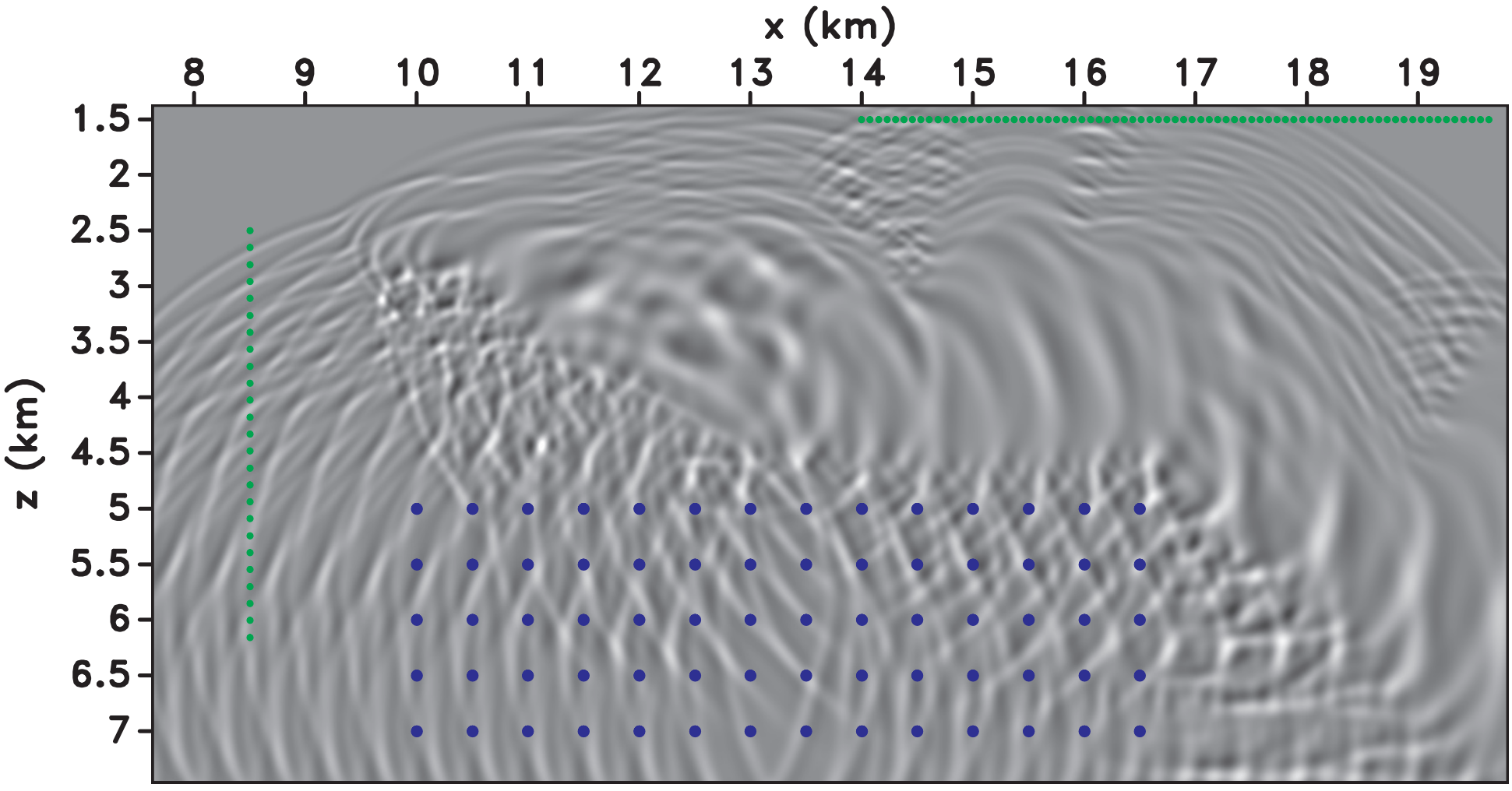

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

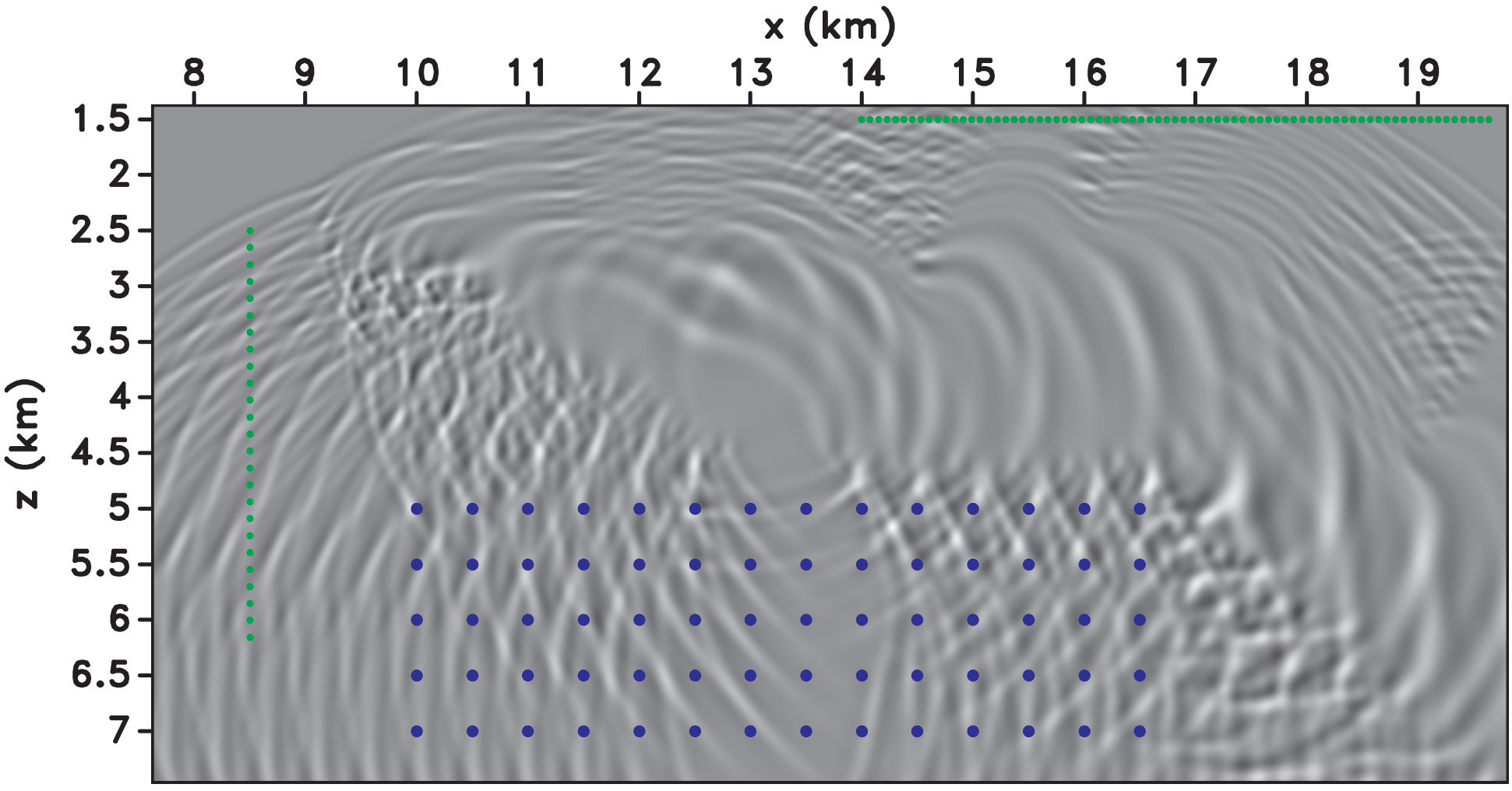

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

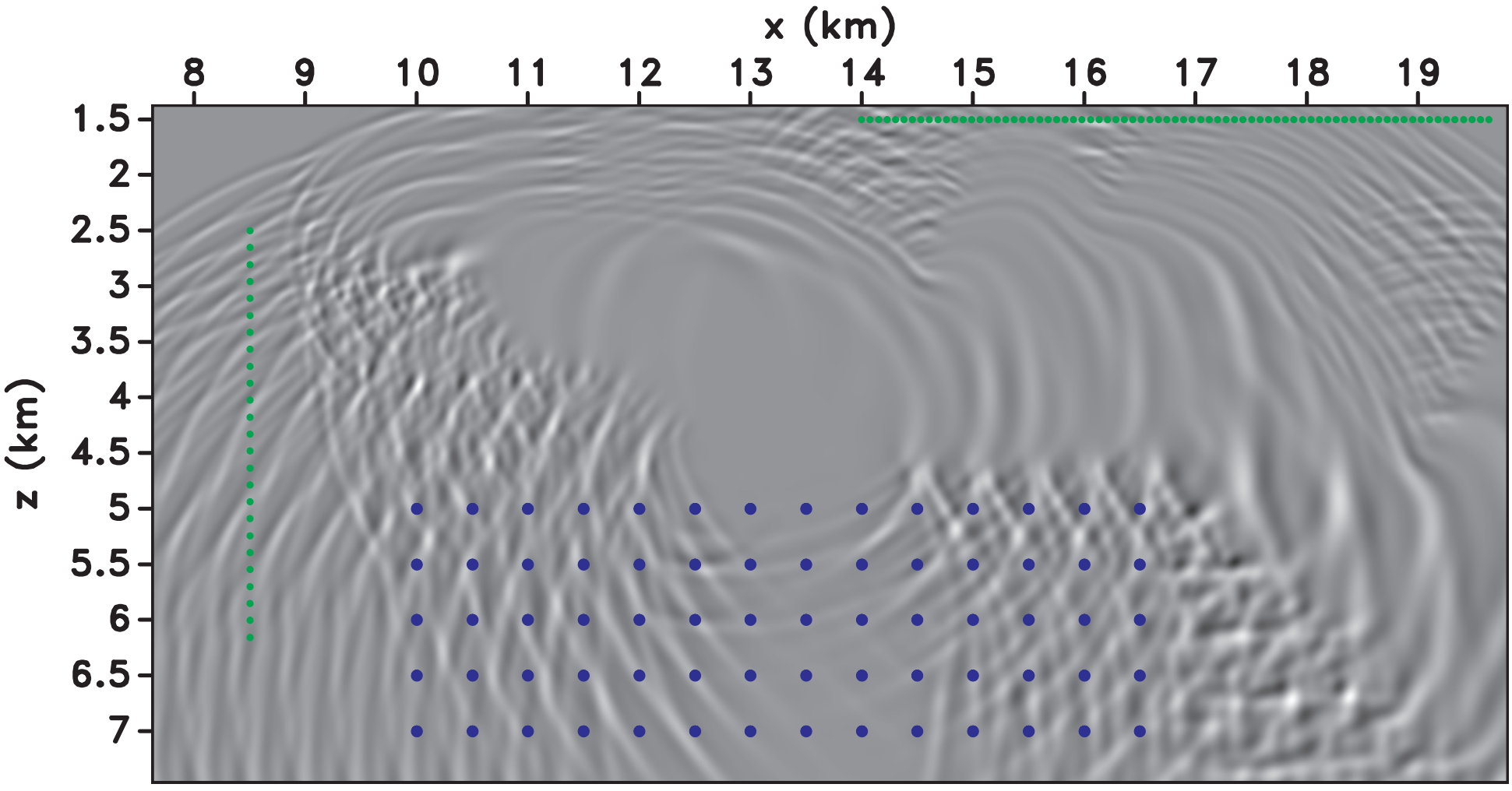

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

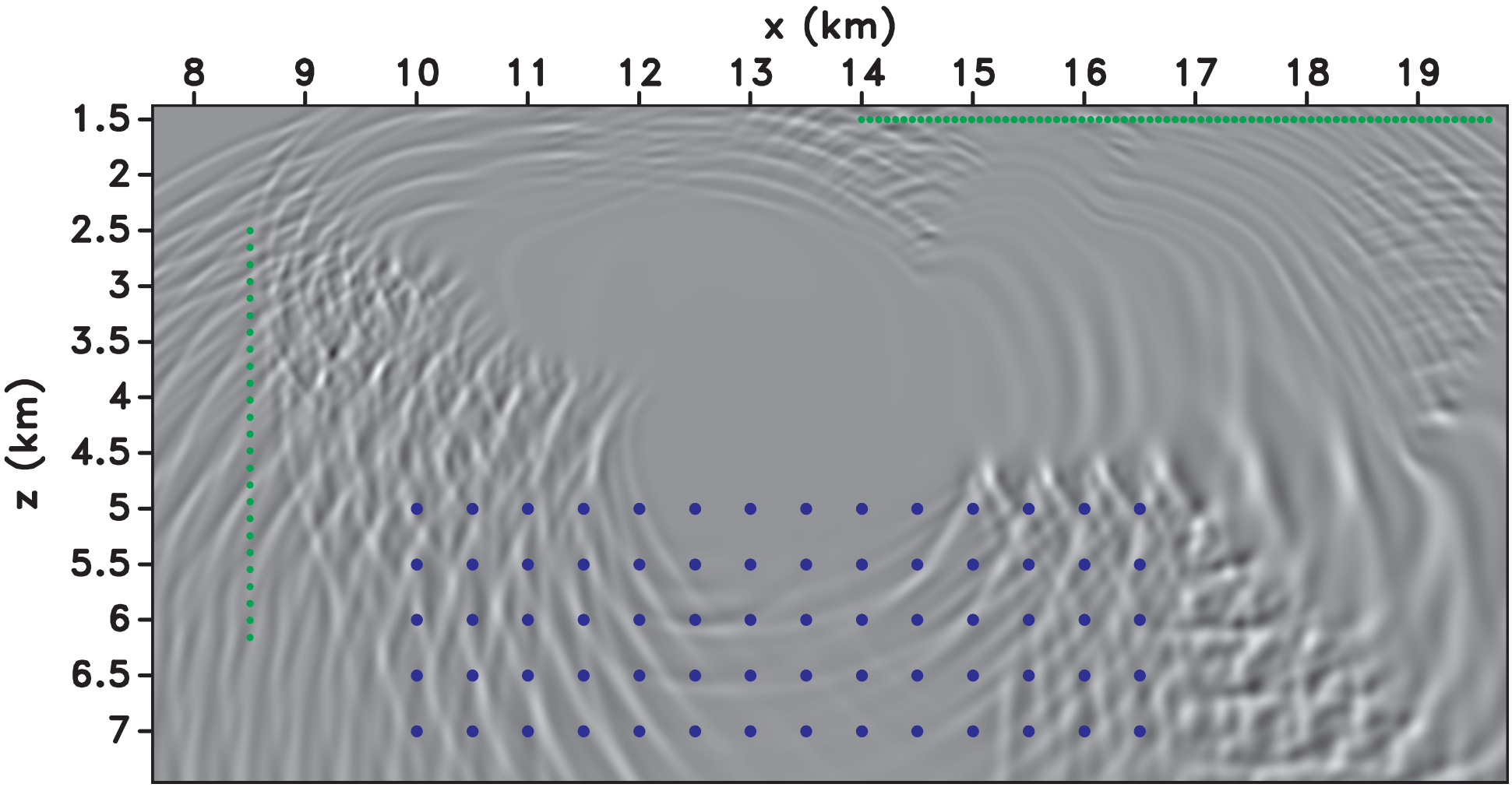

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

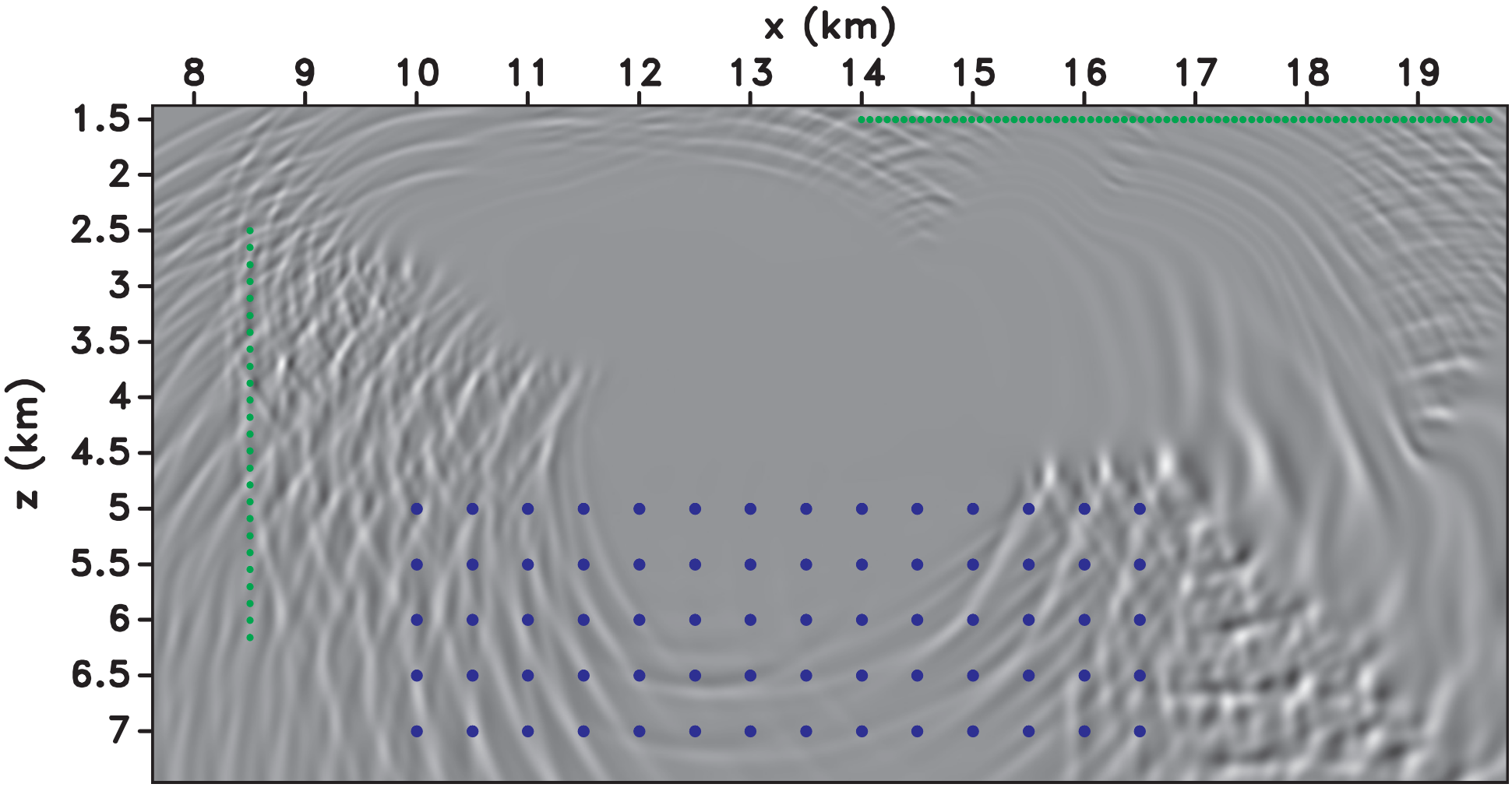

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

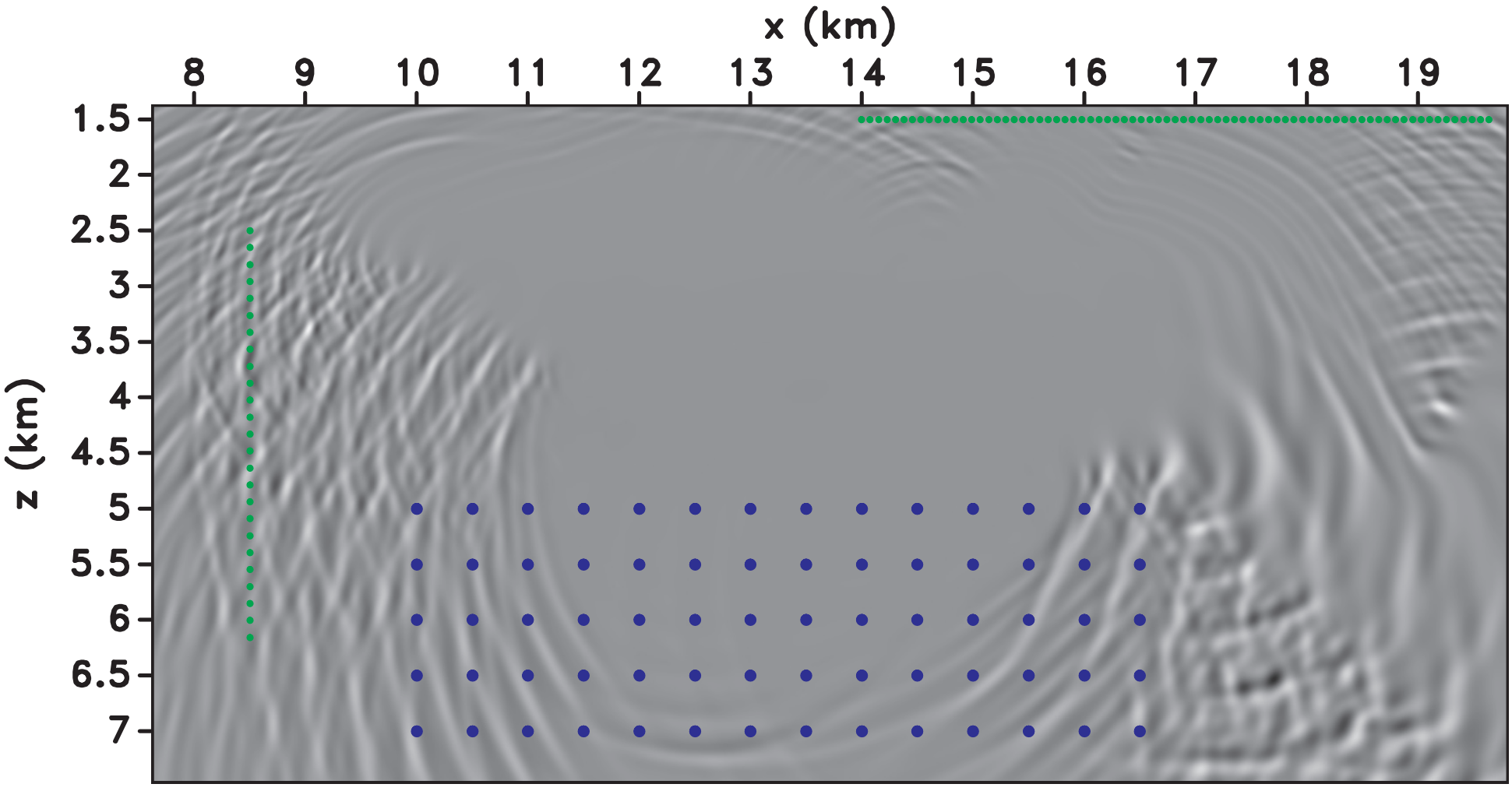

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

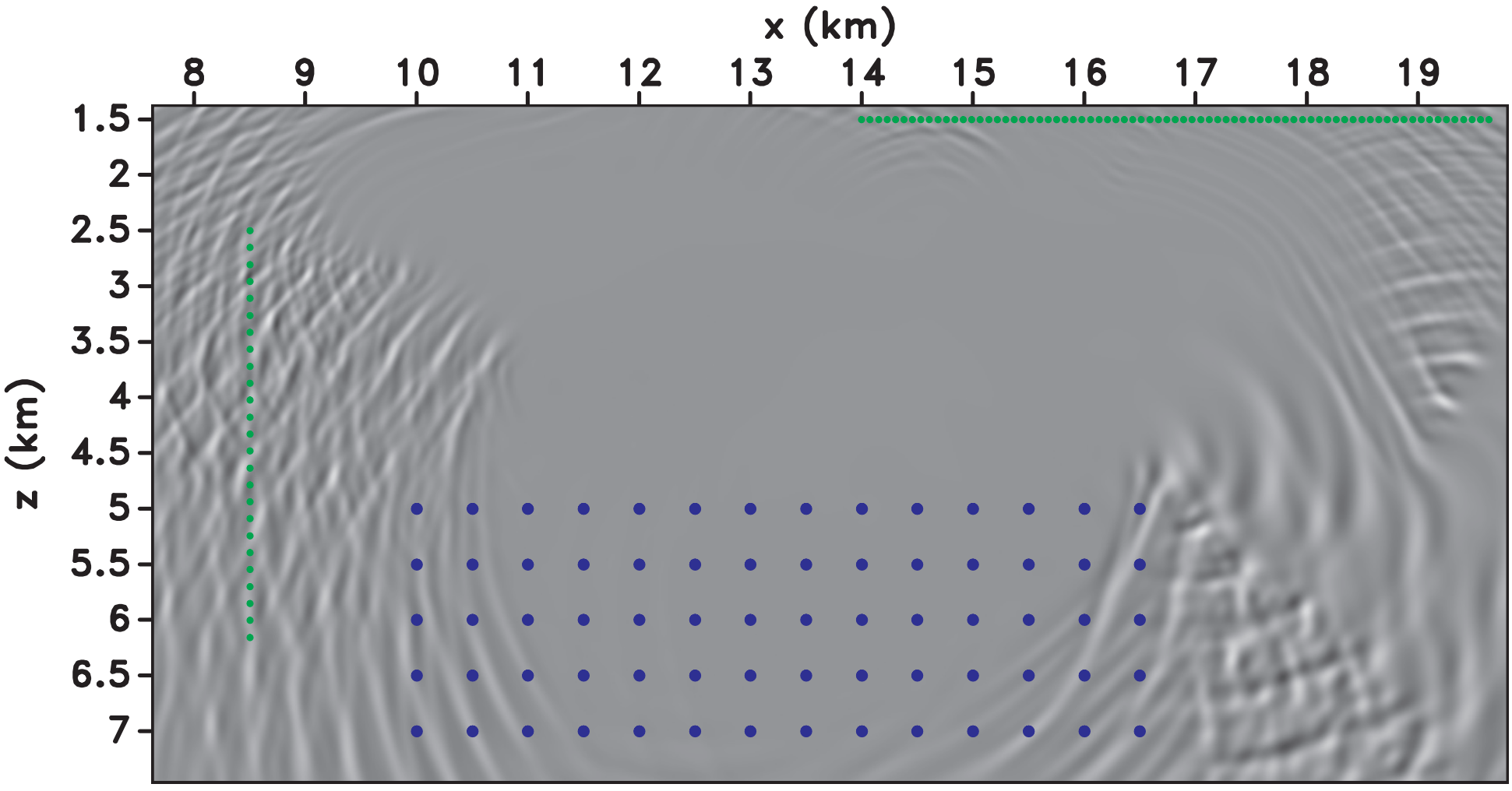

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

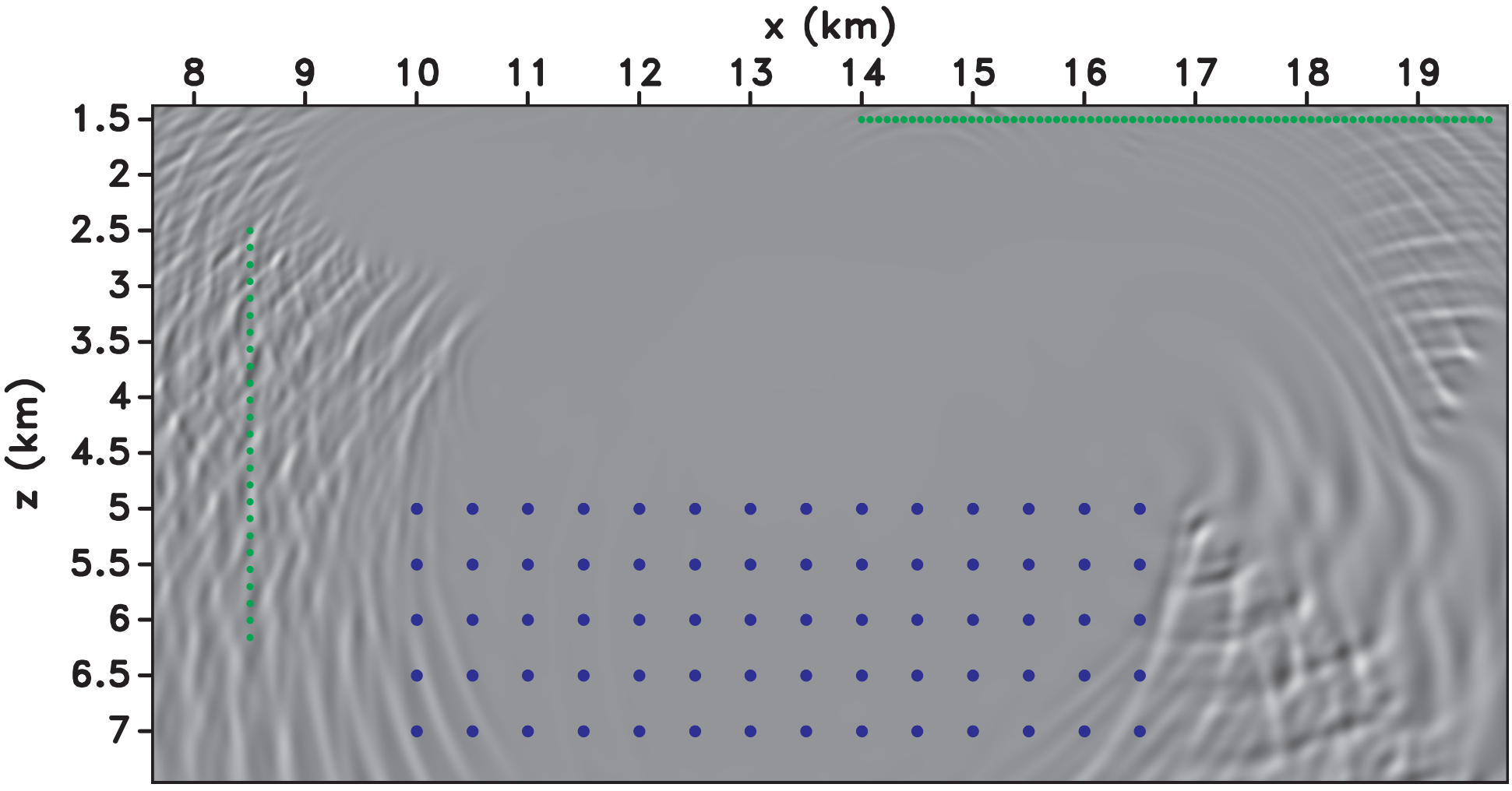

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

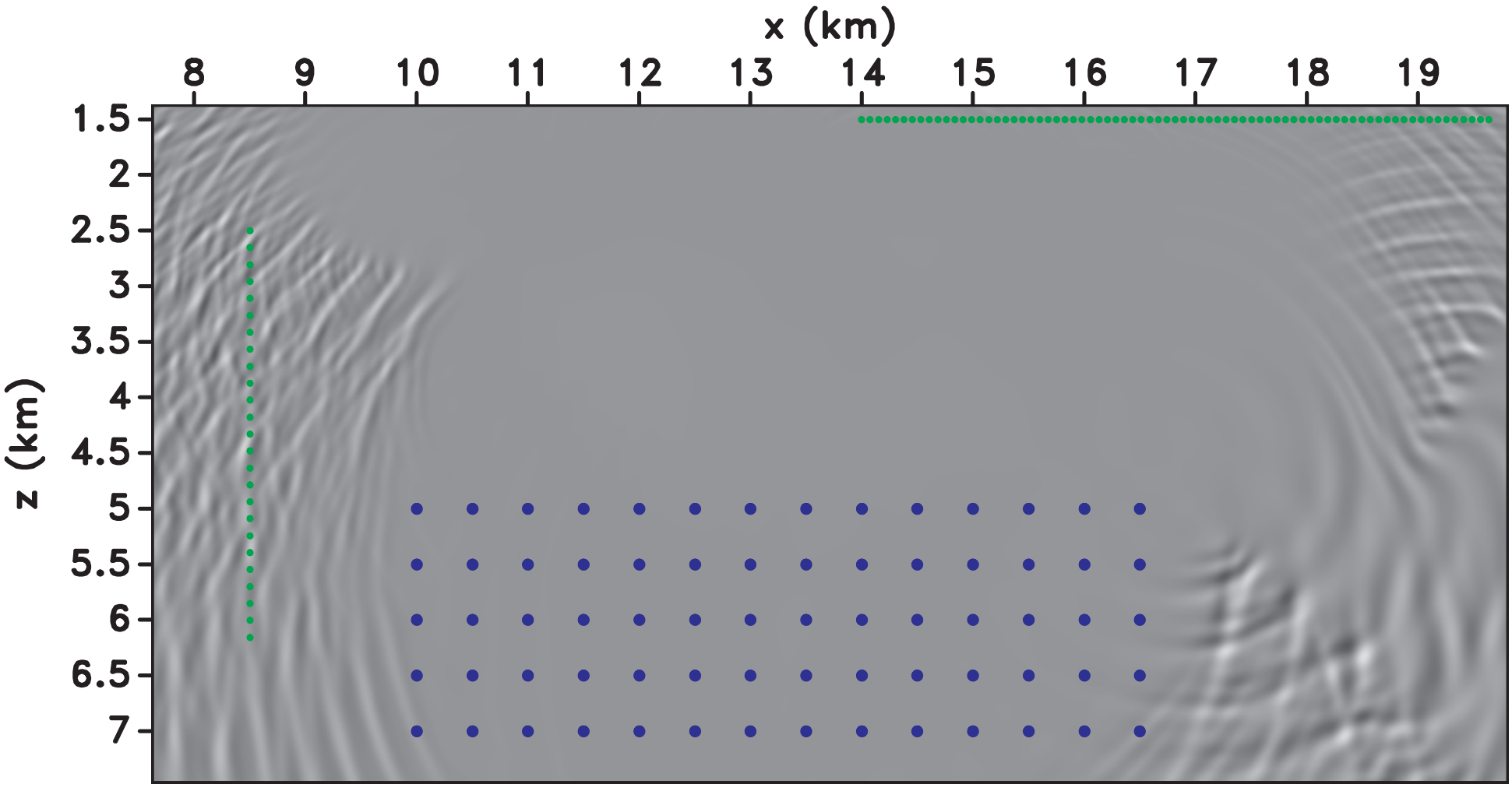

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

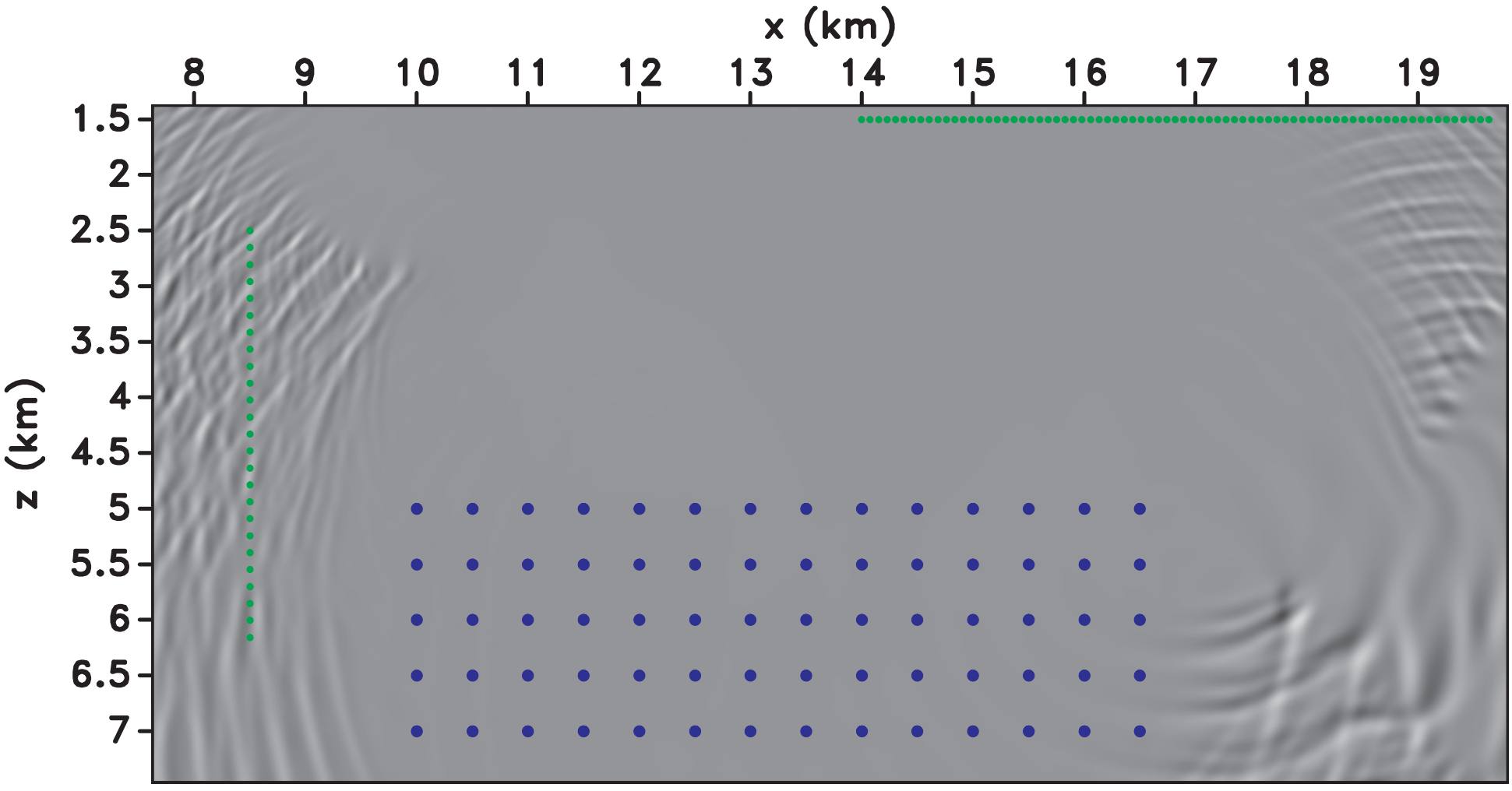

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

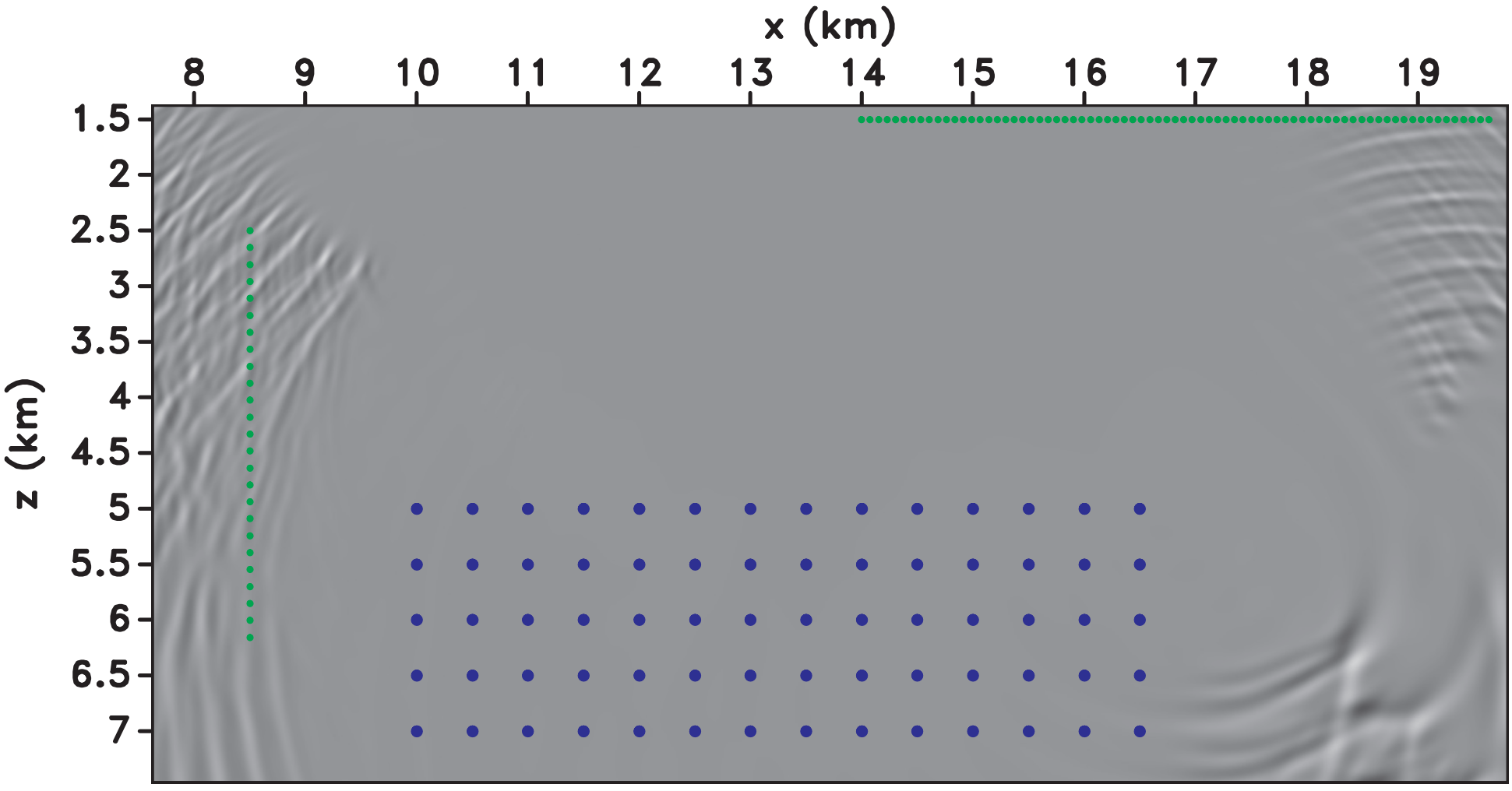

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

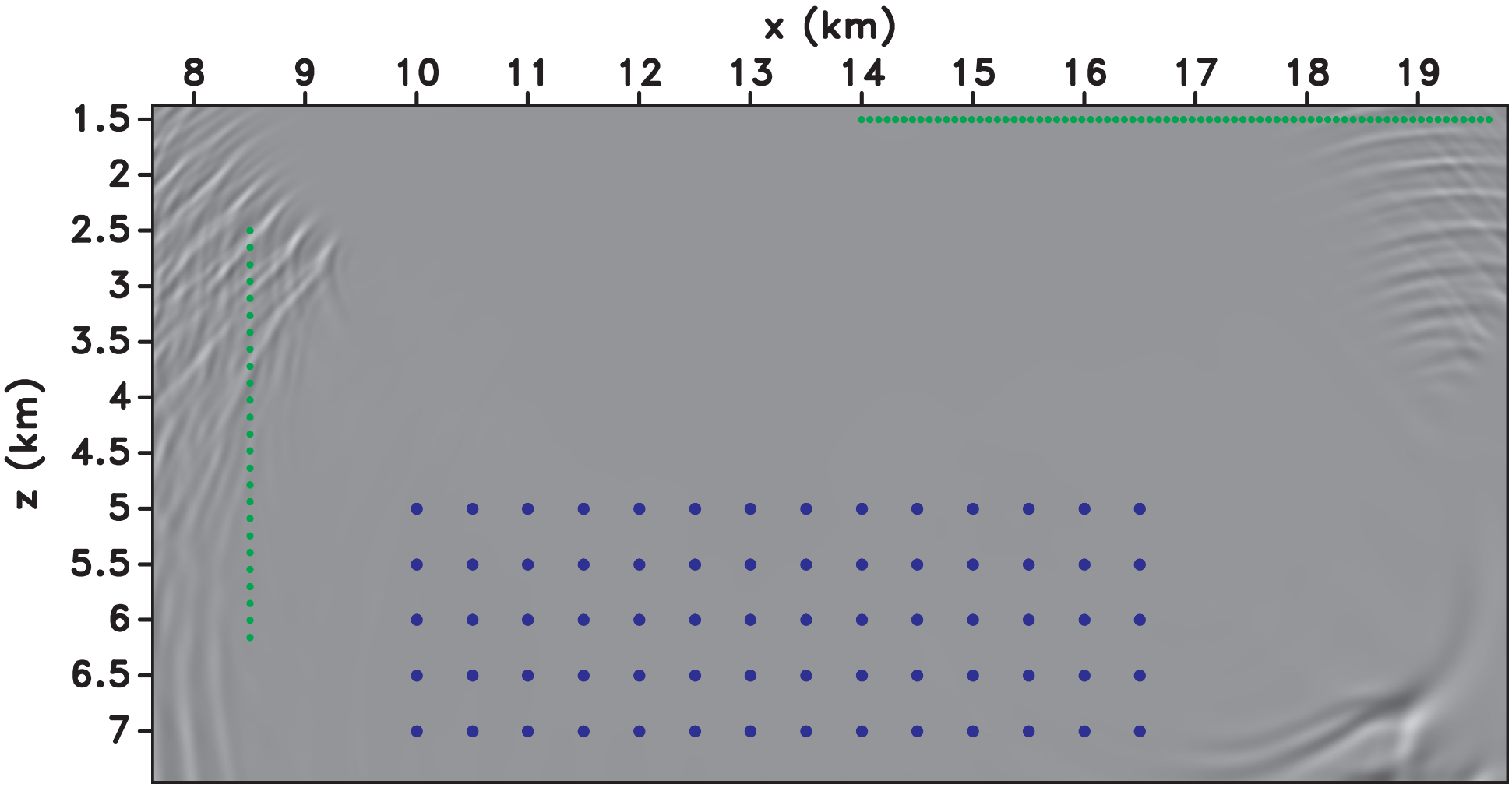

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

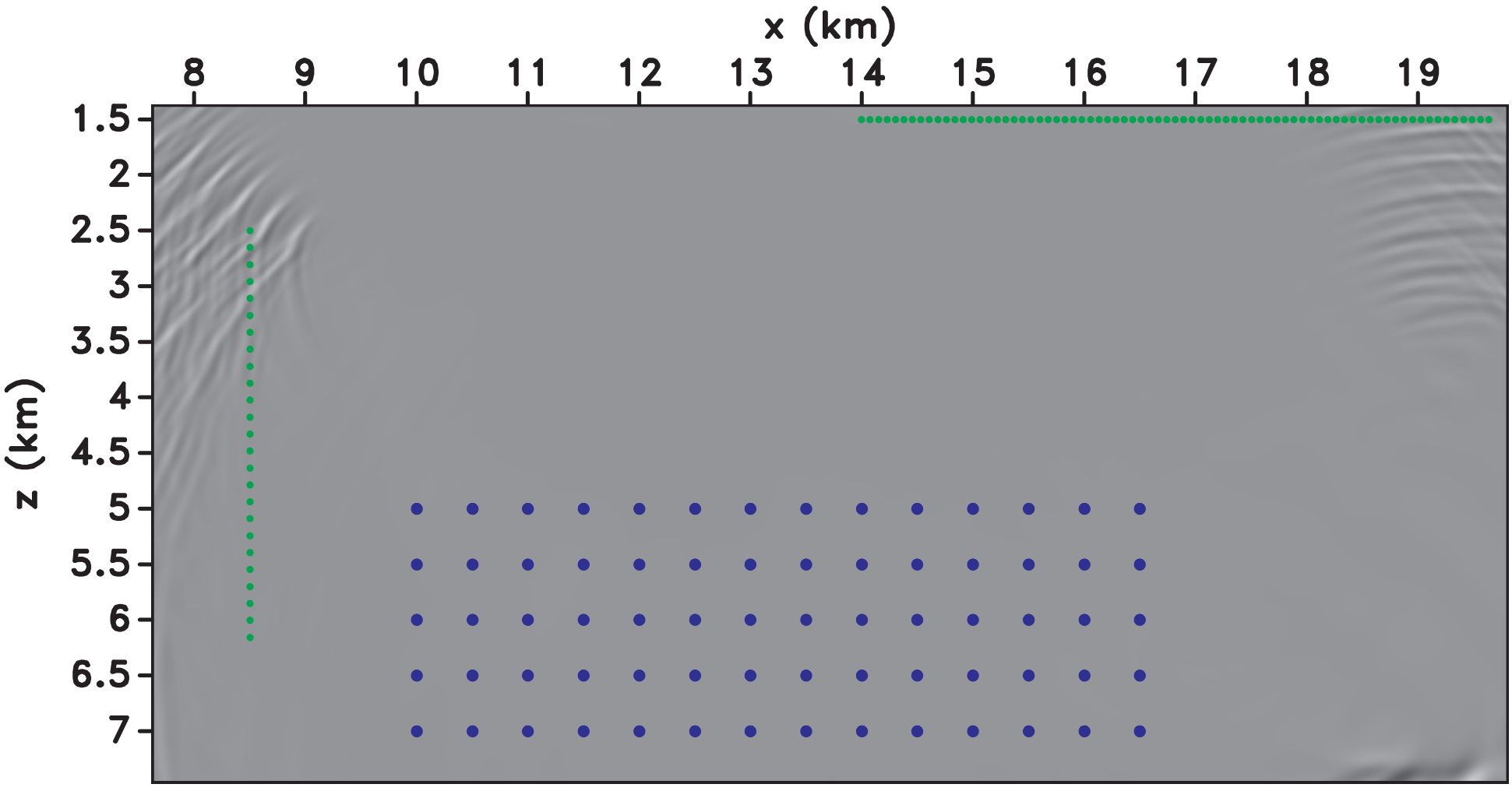

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

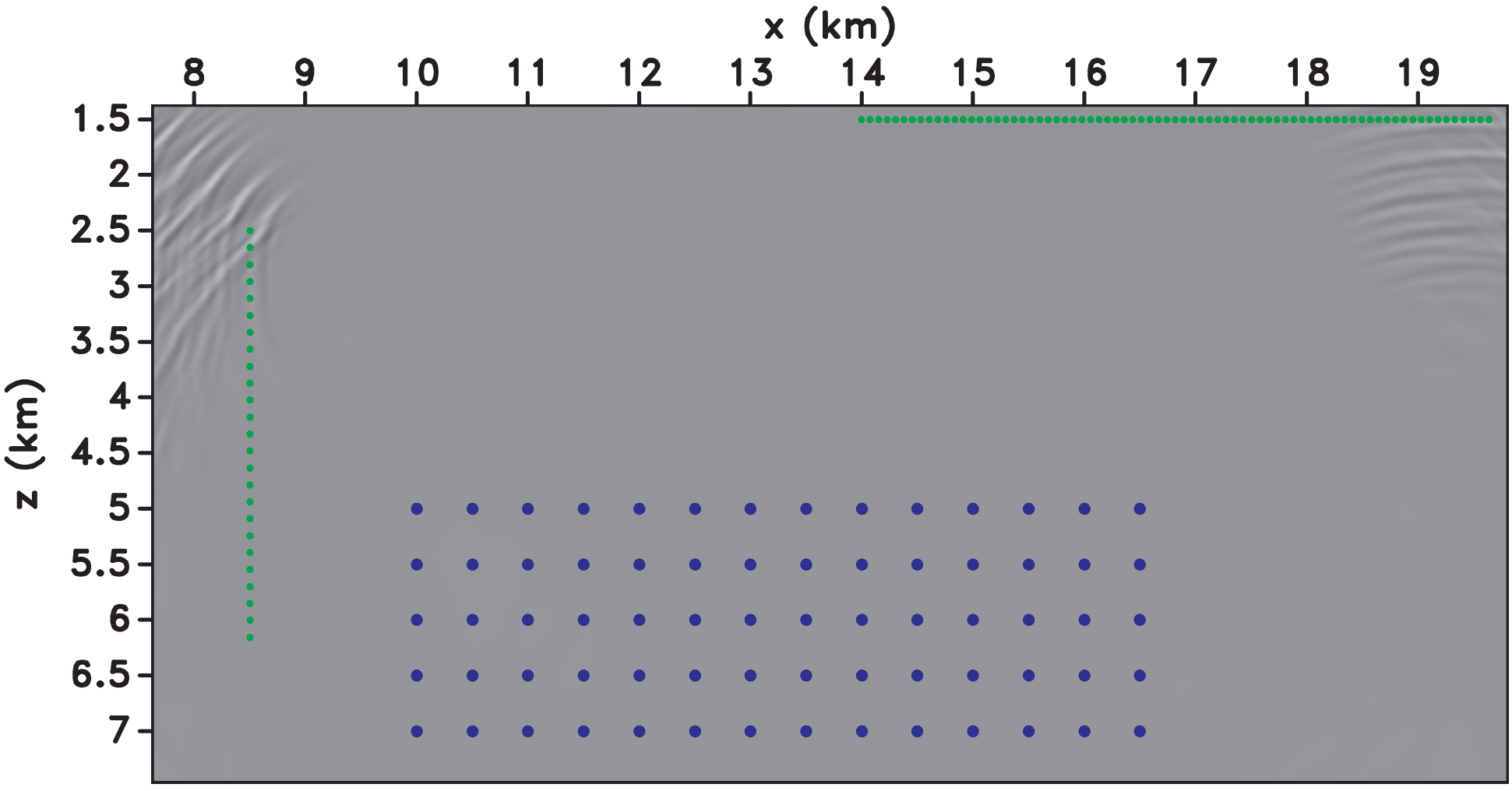

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

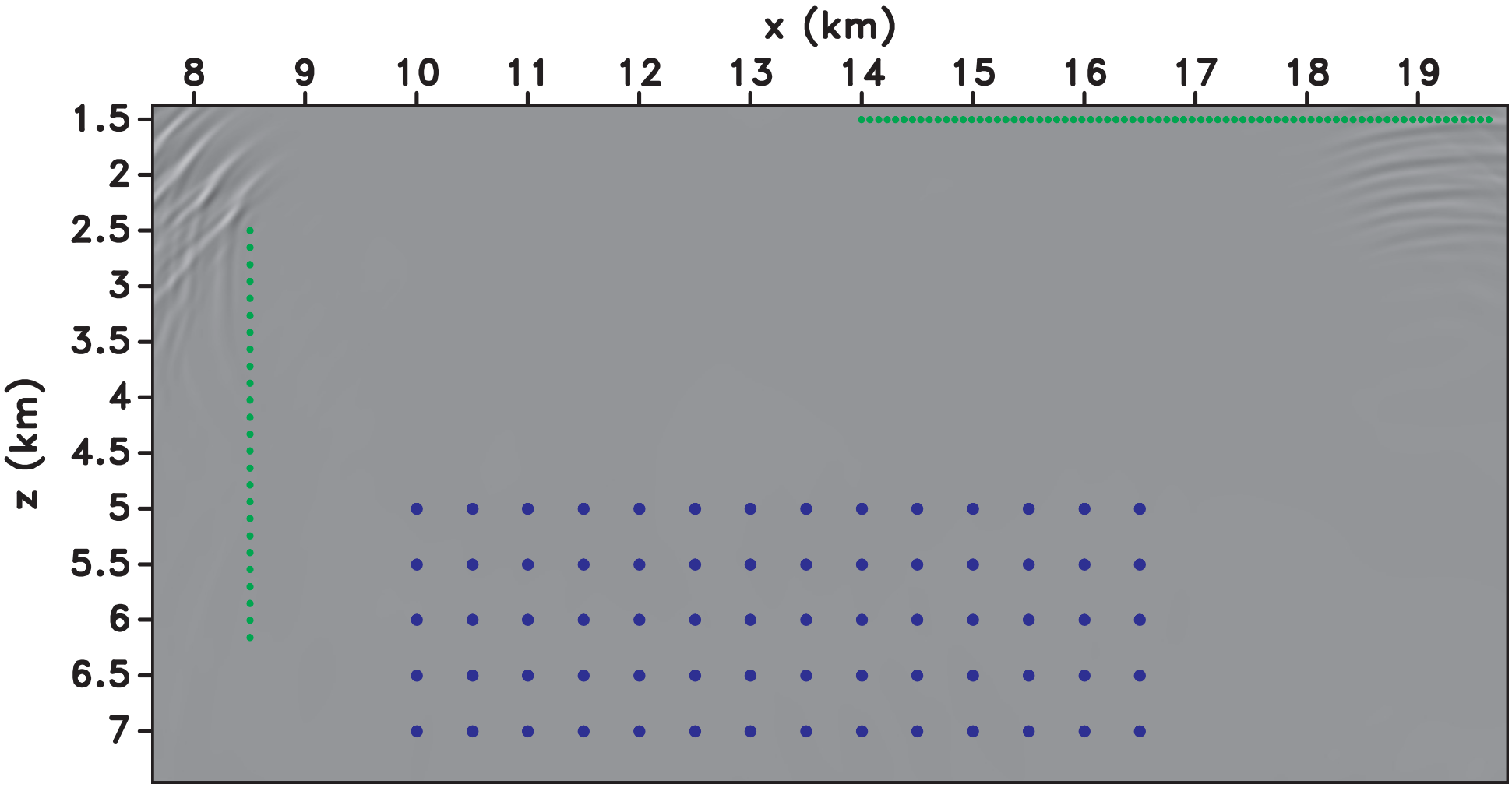

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

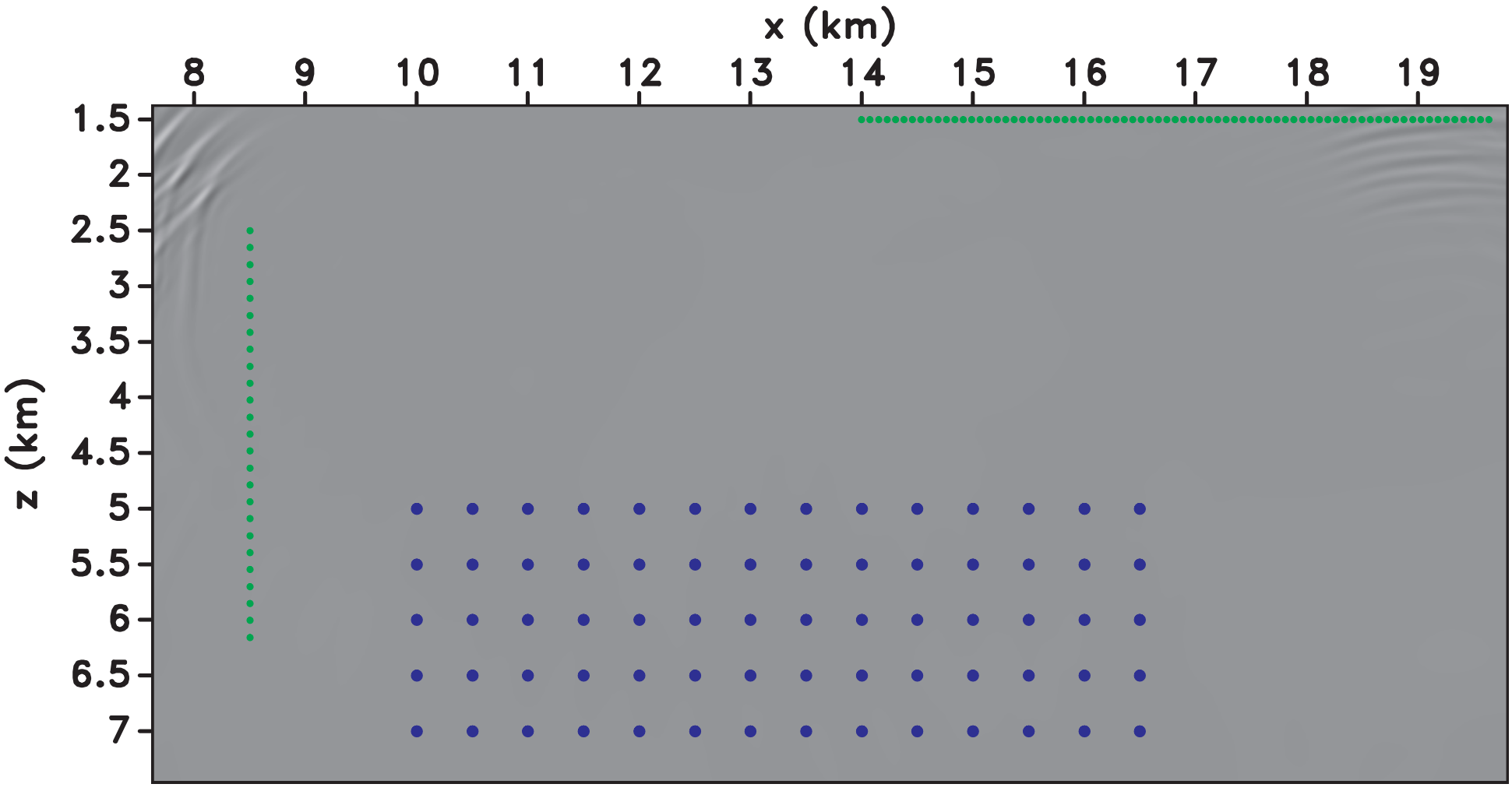

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

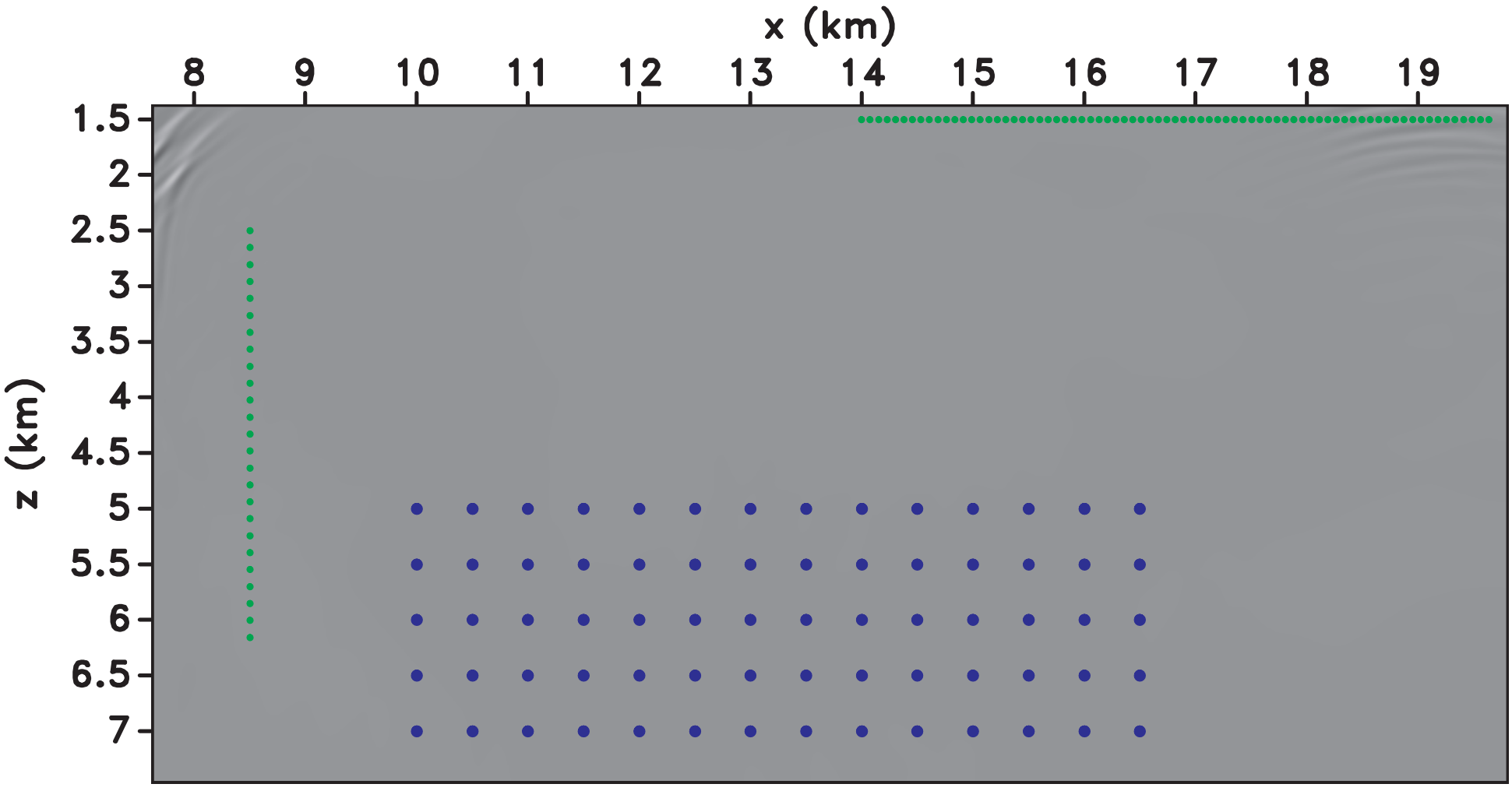

x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7


x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7


x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7


x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7


x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7


x (km)
8
9
10
11
12
13
14
15
16
17
18
19
z (km)
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

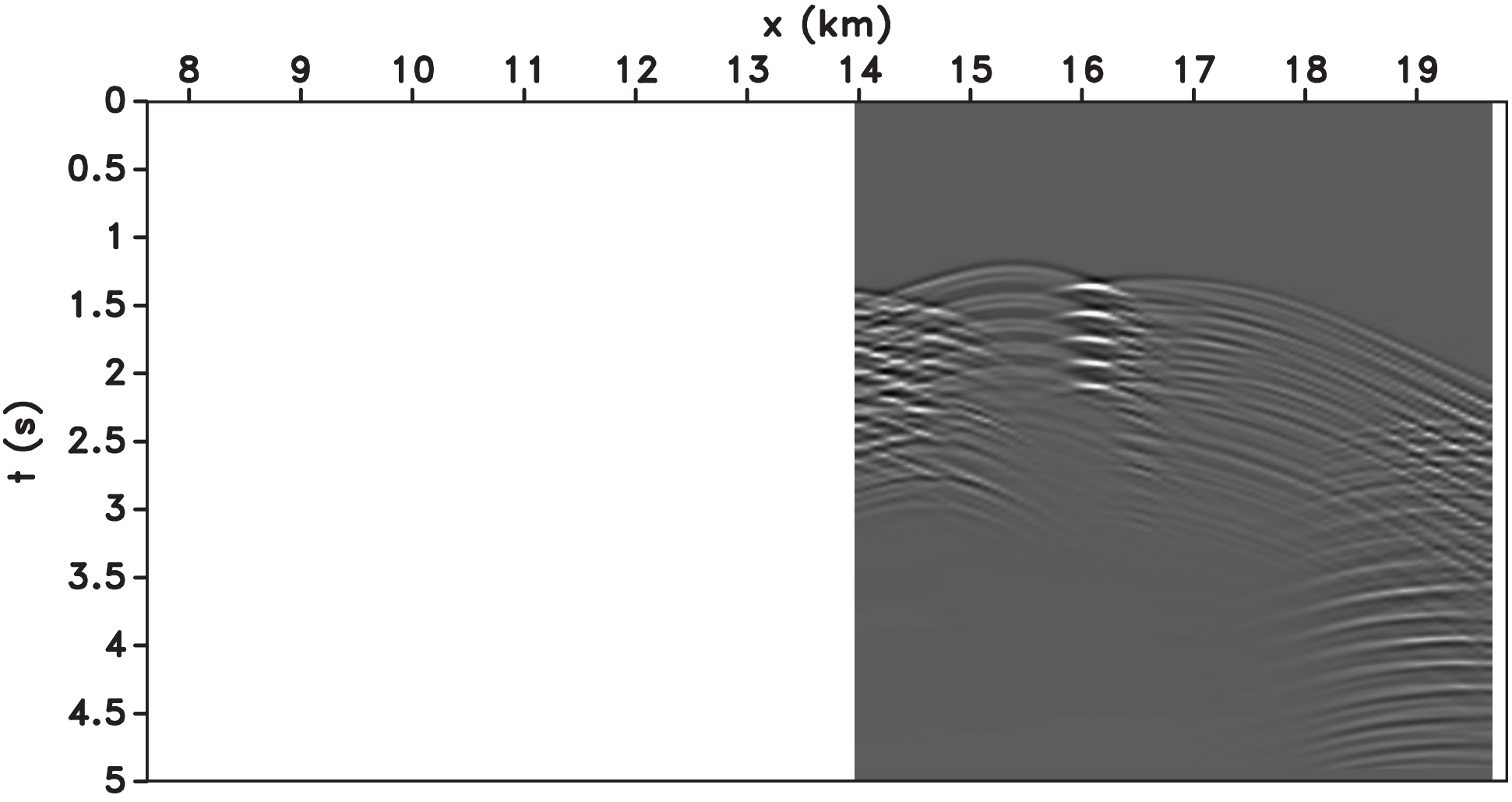
remove the wavelet delay

```
Flow('datA','tmpA',
     '''
     window squeeze=n f2=%(kt)d |
     pad end2=%(kt)d |
     put o2=%(ot)g
     ''' %par)
```

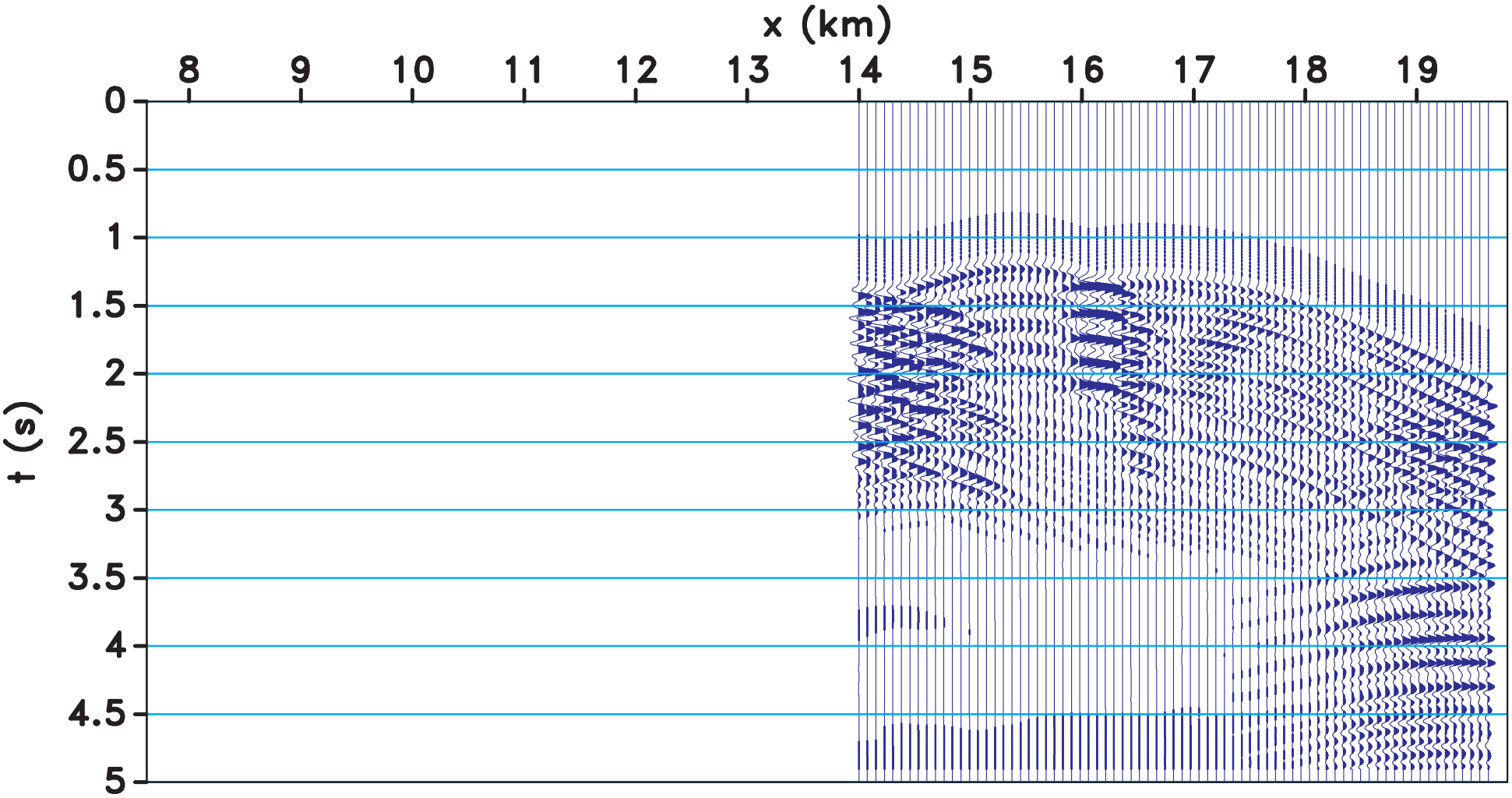
setup the horizontal array data

```
Flow('datH','datA',
     '''
     window squeeze=n n1=%d |
     put o1=%g d1=%g
     '''%(par['nrH'],
          par['orH'],
          par['jrH']*par['dx']))
Result('datH',          'window j2=4 | transp|'
       + fdmod.dgrey(''%par,par))
Result('wigH','datH','window j2=4 | transp|'
       + fdmod.dwigl('pclip=98'%par,par))
```

```
scons datH.view
```



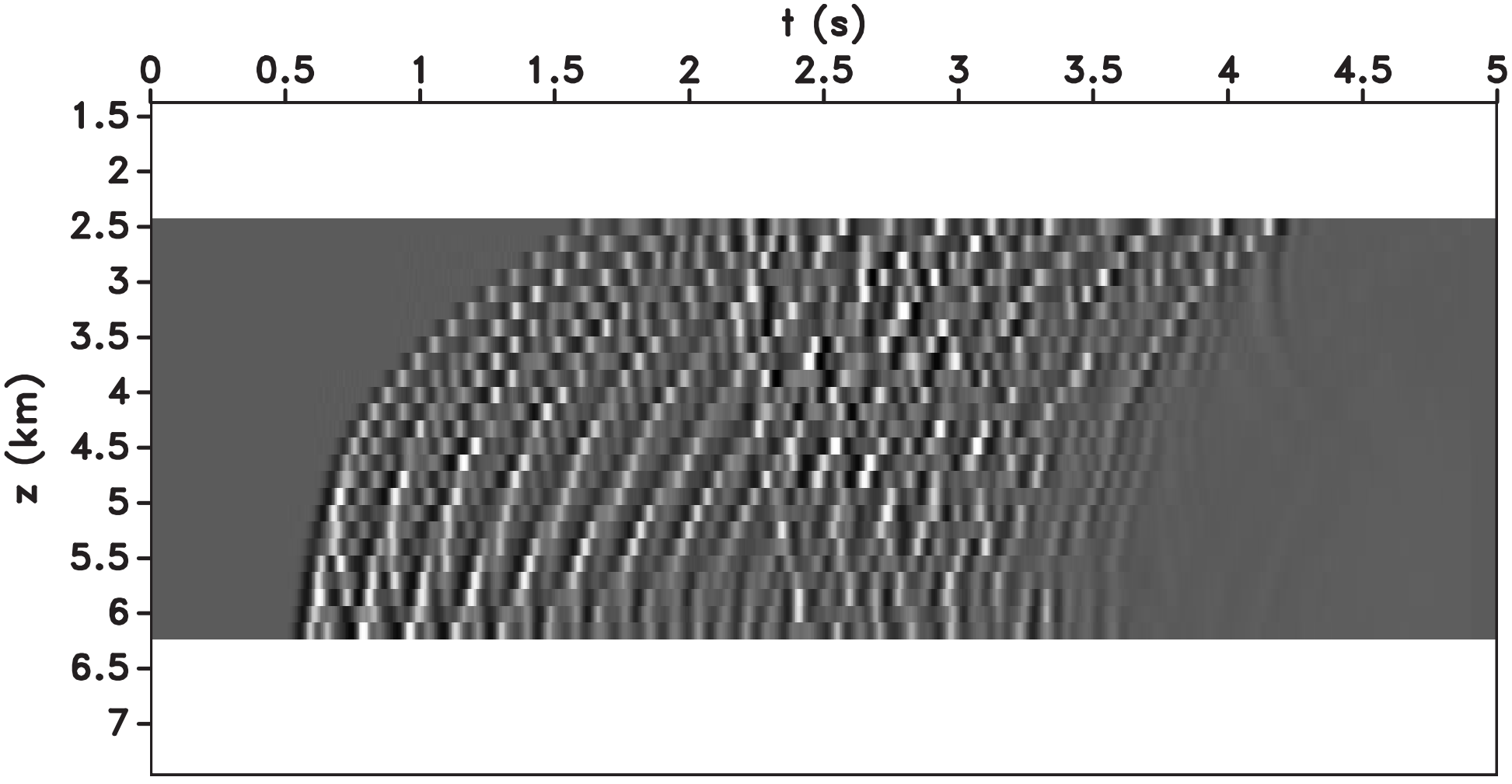
scons wigH.view



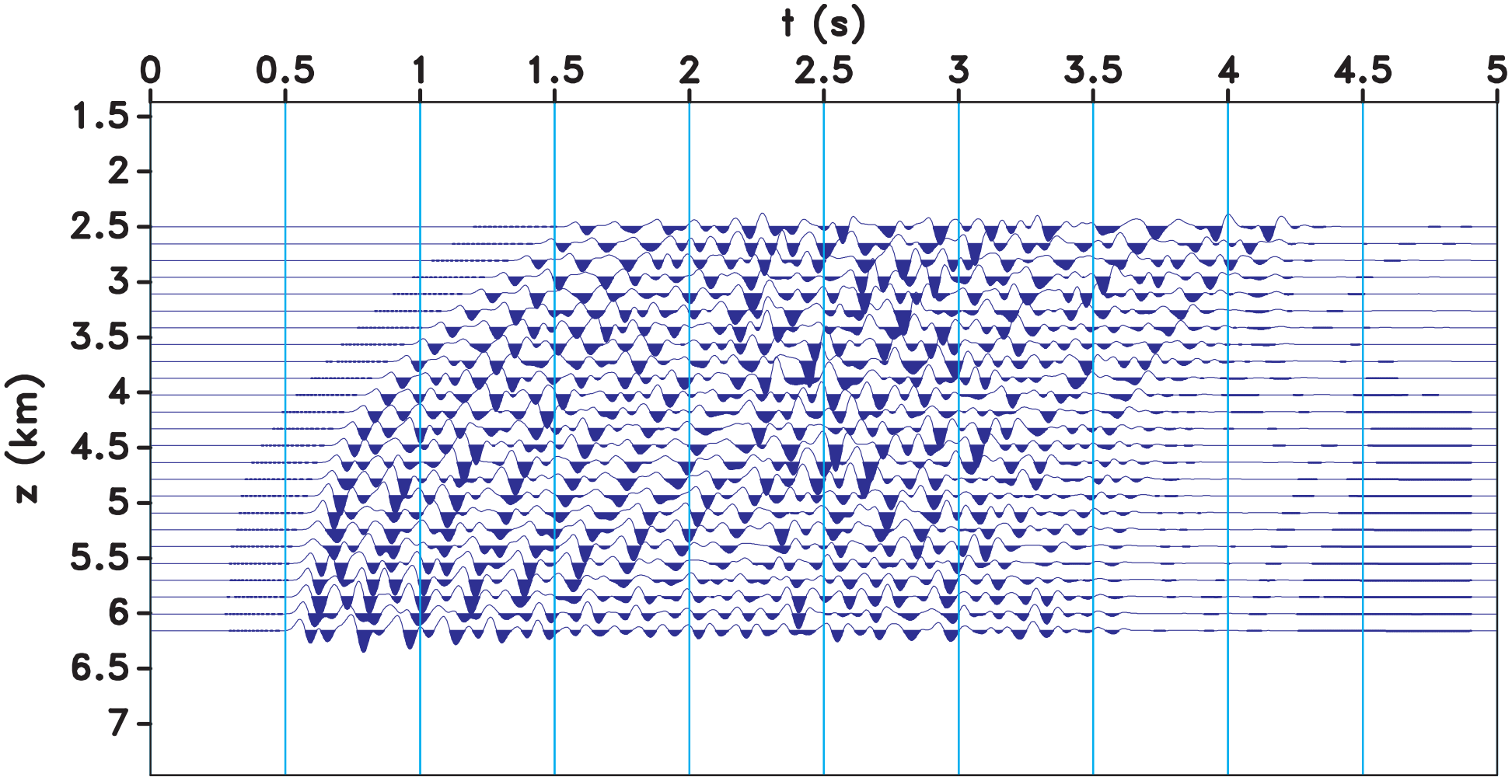
setup the vertical array data

```
Flow('datV','datA',
     '''
     window squeeze=n f1=%d |
     put o1=%g d1=%g
     '''%(par['nrH'],
          par['orV'],
          par['jrV']*par['dz']))
Result('datV',          'window j2=4 |'
       + fdmod.egrey(''%par,par))
Result('wigV','datV','window j2=4 |'
       + fdmod.ewigl('pclip=98'%par,par))
```

scons datV.view



scons wigV.view



run FD migration

```
for j in ('H','V','A'):
```

```
    fdmod.zom('img'+j,   # image
              'dat'+j,   # data
              'vsmo',    # velocity
              'dens',    # density
              'r'+j,     # receiver coordinates
              'fsrf=n',  # optional flags
              par)
```

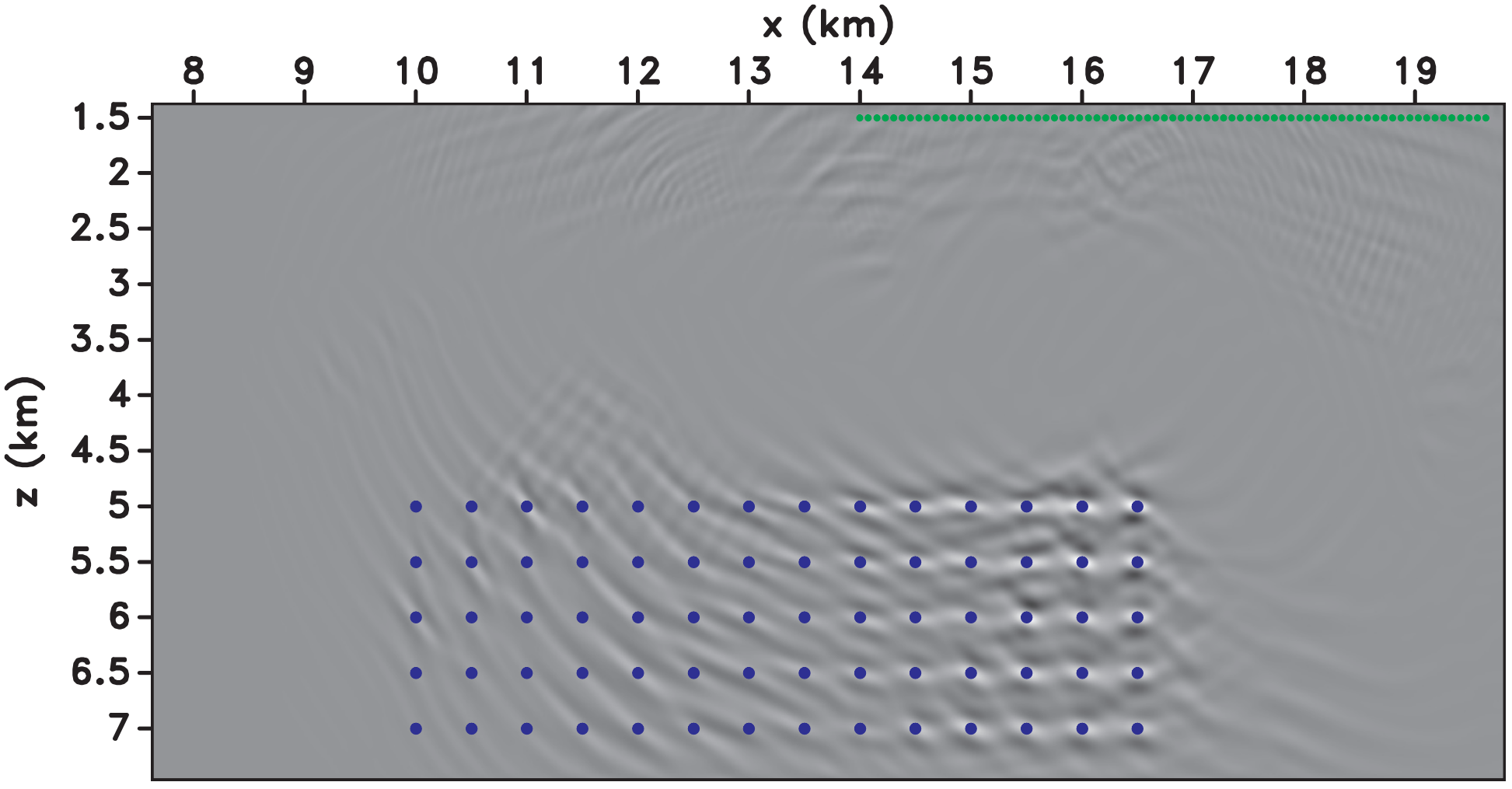
plot the image

```
for j in ('H','V','A'):
```

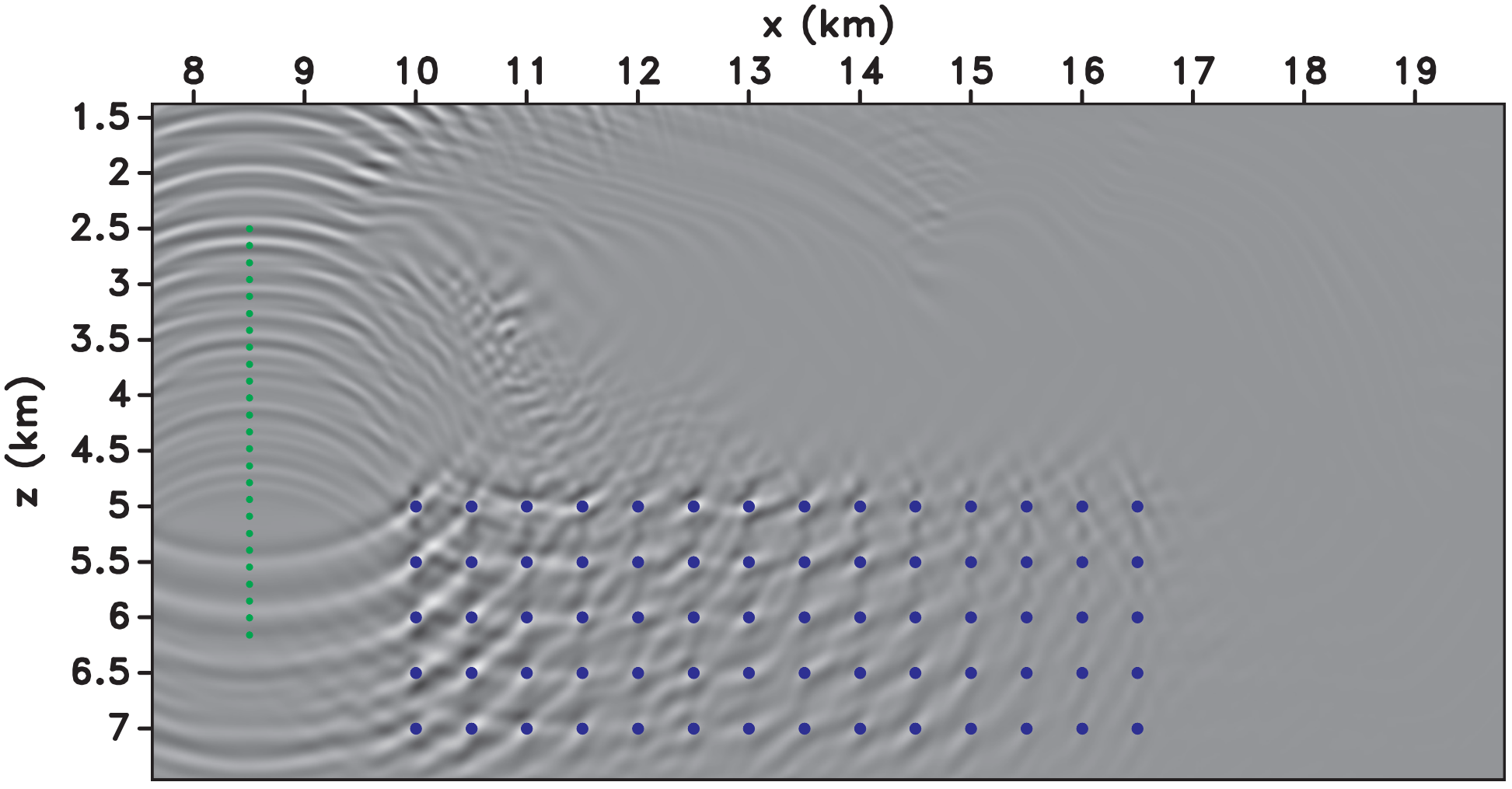
```
    Plot(   'img'+j,'bandpass flo=2 |'
          + fdmod.cgrey('pclip=99.99',par))
```

```
    Result('img'+j,['img'+j,'ss','r'+j],'Overlay')
```

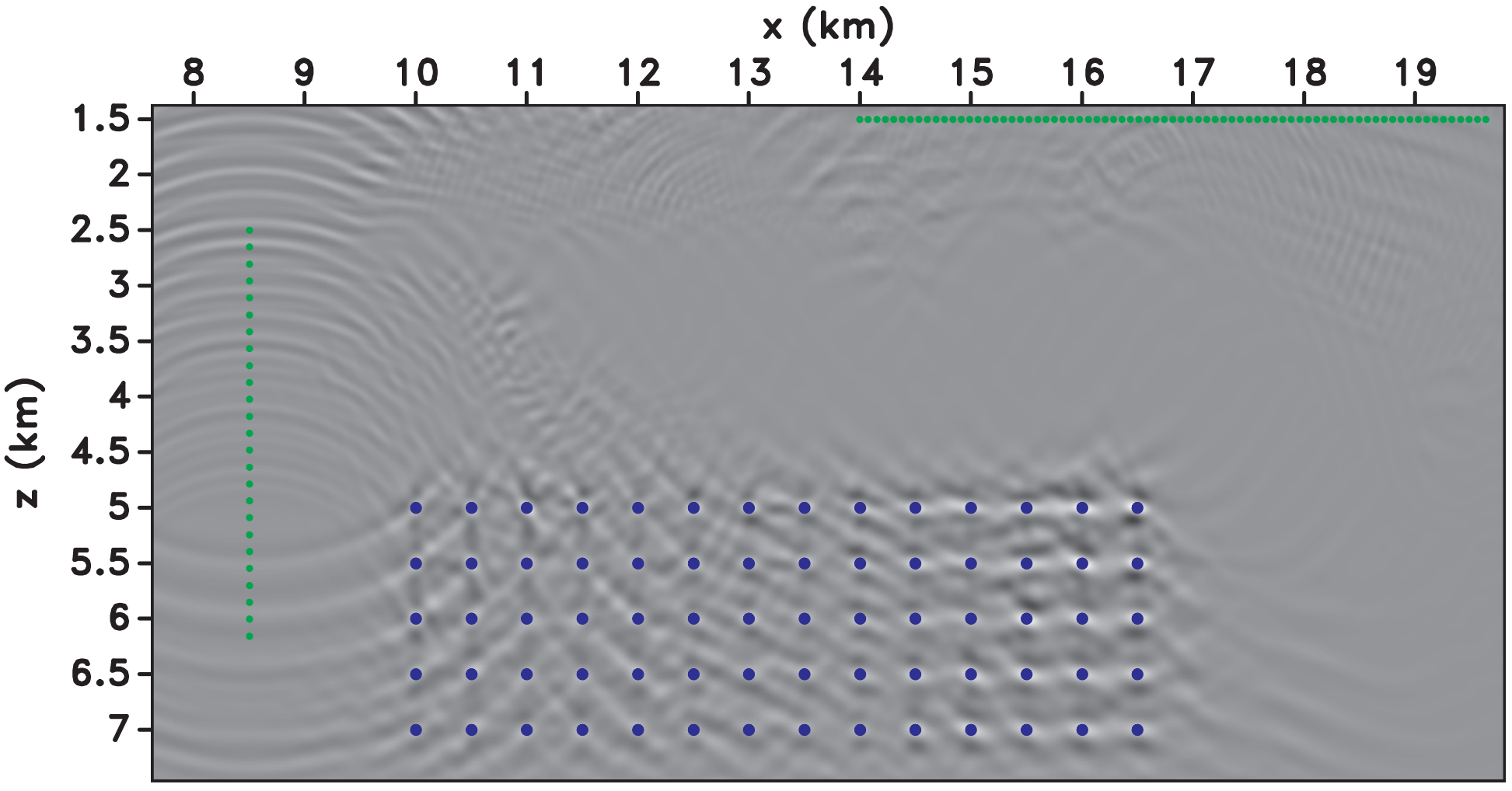
scons imgH.view



scons imgV.view



scons imgA.view



close

```
End()
```

http://www.ahay.org

$RSF/book/rsf/school

assignment

Modify the acquisition parameters
to explore the seismic illumination
at different locations in the subsurface.