Seismic field data processing example

-- Tutorial of Madagascar --

http://www.ahay.org

Yang Liu (刘洋)

College of Geo-exploration Science and Technology,

Jilin University

07/22/2011

yangliu1979@gmail.com

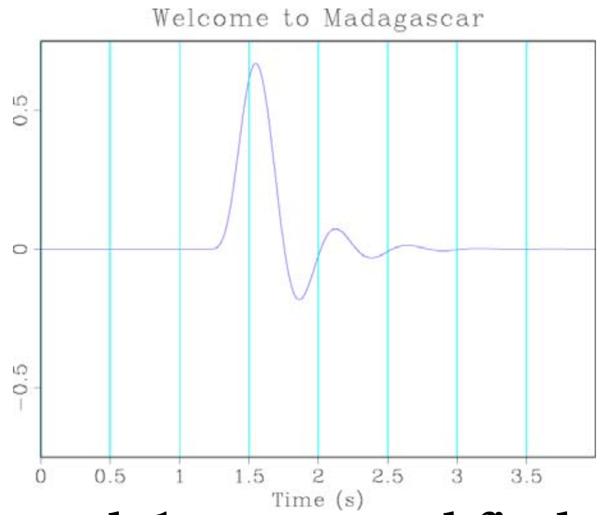
Outline

- **Dive into Madagascar**
- **Assignment**
- **Simple workflow**

Outline

- **Dive into Madagascar**
- **Assignment**
- **Simple workflow**

Dive into Madagascar



- **Using Self-document**
  - Find which module you want to use by "sfdoc –k keyword"
  - Read self-document by just running module name and find parameters, examples, and source code.
  - Command-line Test

    sfspike n1=1000 k1=300 | sfbandpass fhi=2 phase=1 | \
    sfwiggle clip=0.02 title="Welcome to Madagascar" | sfpen

- **Following examples is the best way to learn.**
  - Everything is reproducible.

Outline

- Dive into Madagascar
- **Assignment**
- Simple workflow

Assignment

Field seismic data are always complicated!

- Crucial Problems for different types of data:
  - Land data: Static correction, SNR, etc.
  - Marine data: Interpolation, Multiples, migration, etc.

- The tutorial is given to set up a processing workflow for a simple deep water data.

Outline

- Dive into Madagascar
- Assignment
- **Simple workflow**

Tutorial workflow

- Get data and convert to RSF format
- Initial data checking
- Initial signal analysis
- Subsampling with anti-aliasing filter
- Convert from CSP to CMP
- Quality control
- Velocity analysis
- NMO and brute stacking
- Poststack time migration

The first step to do:

- **Open a new text file and name it as "SConstruct".**

- **Begin to write your script for workflow.**

Get data and convert to RSF format

Always start here!

```
from rsf.proj import *

# Download data
Fetch('Nshots.su','nankai')

# Convert to RSF
Flow('shots tshots','Nshots.su',
     'suread suxdr=y tfile=${TARGETS[1]} | window min1=5')
```

If you have had the data in the directory, comment this line.

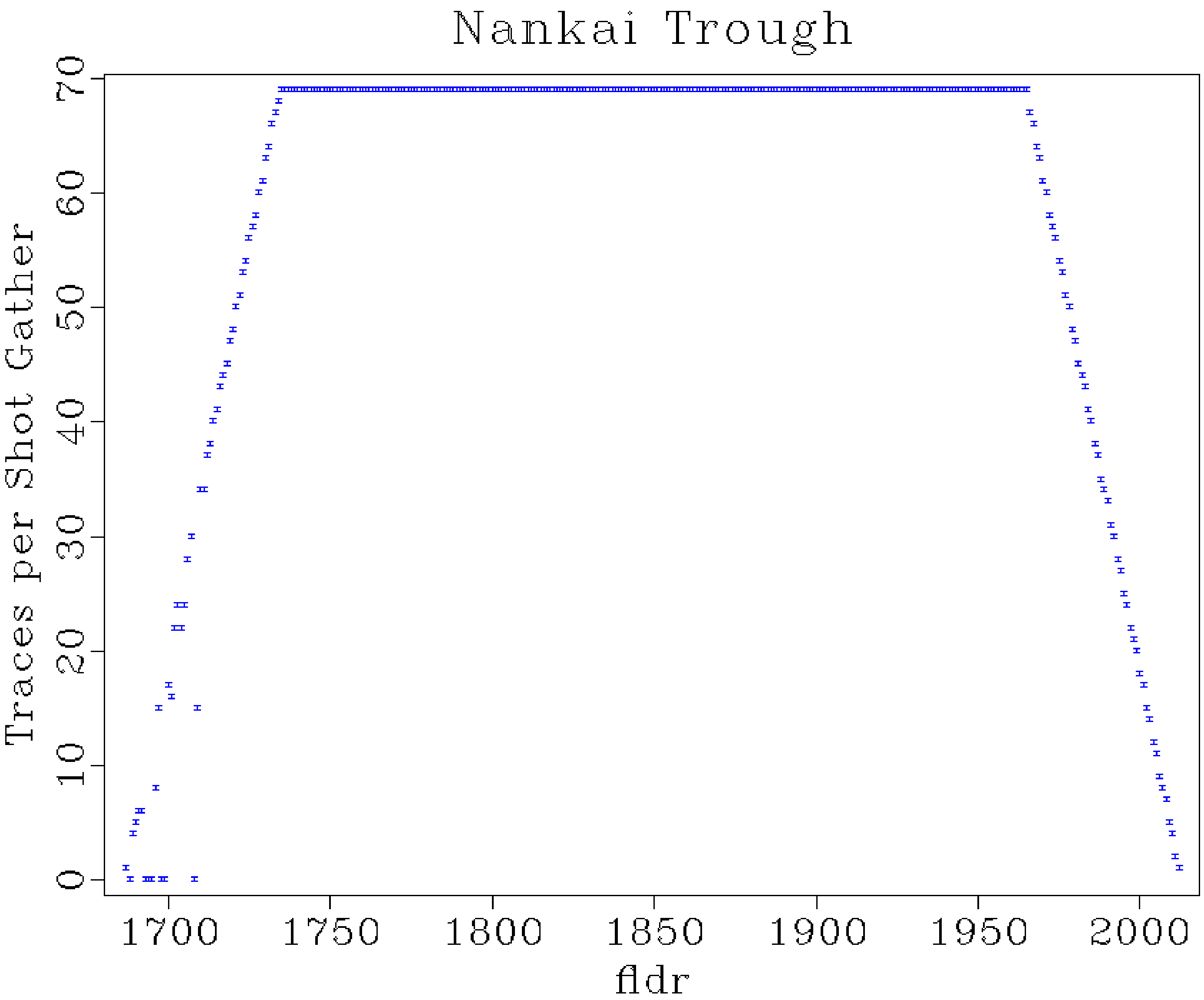
```
$scons  tshots.rsf
$sfheaderattr  < tshots.rsf
******************************************
91 headers, 19057 traces
key[0]="tracl"      min[14770]=39069   max[19056]=58125   mean=48597
key[1]="tracr"      min[14770]=1       max[19056]=19057   mean=9529
key[2]="fldr"       min[14770]=1687    max[19056]=2012    mean=1850.61
key[3]="tracf"      min[14770]=28      max[19056]=96      mean=61.8195
key[5]="cdp"        min[14770]=900     max[19056]=1300    mean=1101.39
key[6]="cdpt"       min[14770]=1       max[19056]=69      mean=33.1667
key[7]="trid"       min[14770]=1       max[19056]=2       mean=1.0001
key[11]="offset"    min[14770]=-2435   max[19056]=-170    mean=-1308.69
key[19]="scalel"    min[14770]=-10000  max[19056]=-10000  mean=-10000
key[20]="scalco"    min[14770]=-10000  max[19056]=-10000  mean=-10000
key[25]="counit"    min[14770]=1       max[19056]=1       mean=1
key[36]="muts"      min[14770]=0       max[19056]=11000   mean=1.15443
key[38]="ns"        min[14770]=5500    max[19056]=5500    mean=5500
key[39]="dt"        min[14770]=2000    max[19056]=2000    mean=2000
key[60]="day"       min[14770]=206     max[19056]=206     mean=206
key[61]="hour"      min[14770]=21      max[19056]=22      mean=21.225
key[62]="minute"    min[14770]=0       max[19056]=59      mean=28.3935
key[63]="sec"       min[14770]=1       max[19056]=59      mean=30.0732
******************************************
```

Initial data checking

```
Flow('shots2 smask','shots',
     'intbin mask=${TARGETS[1]} xk=tracf yk=fldr')

Result('smask',
       '''
       dd type=float |
       stack axis=1 norm=n |
       graph symbol=x title="Nankai Trough"
       label2="Traces per Shot Gather"
       ''')
```

$scons smask.view



```
Flow('offset','tshots',
     '''
     window n1=1 f1=11 squeeze=n | dd type=float |
     intbin head=$SOURCE xk=tracf yk=fldr
     ''')
```

```
$scons  offset.rsf
$sfin offset.rsf
offset.rsf:
   in="/your directory/offset.rsf@"
   esize=4 type=float form=native
   n1=1          d1=1          o1=11
   n2=69         d2=1          o2=28
   n3=326        d3=1          o3=1687
      22494 elements 89976 bytes
```

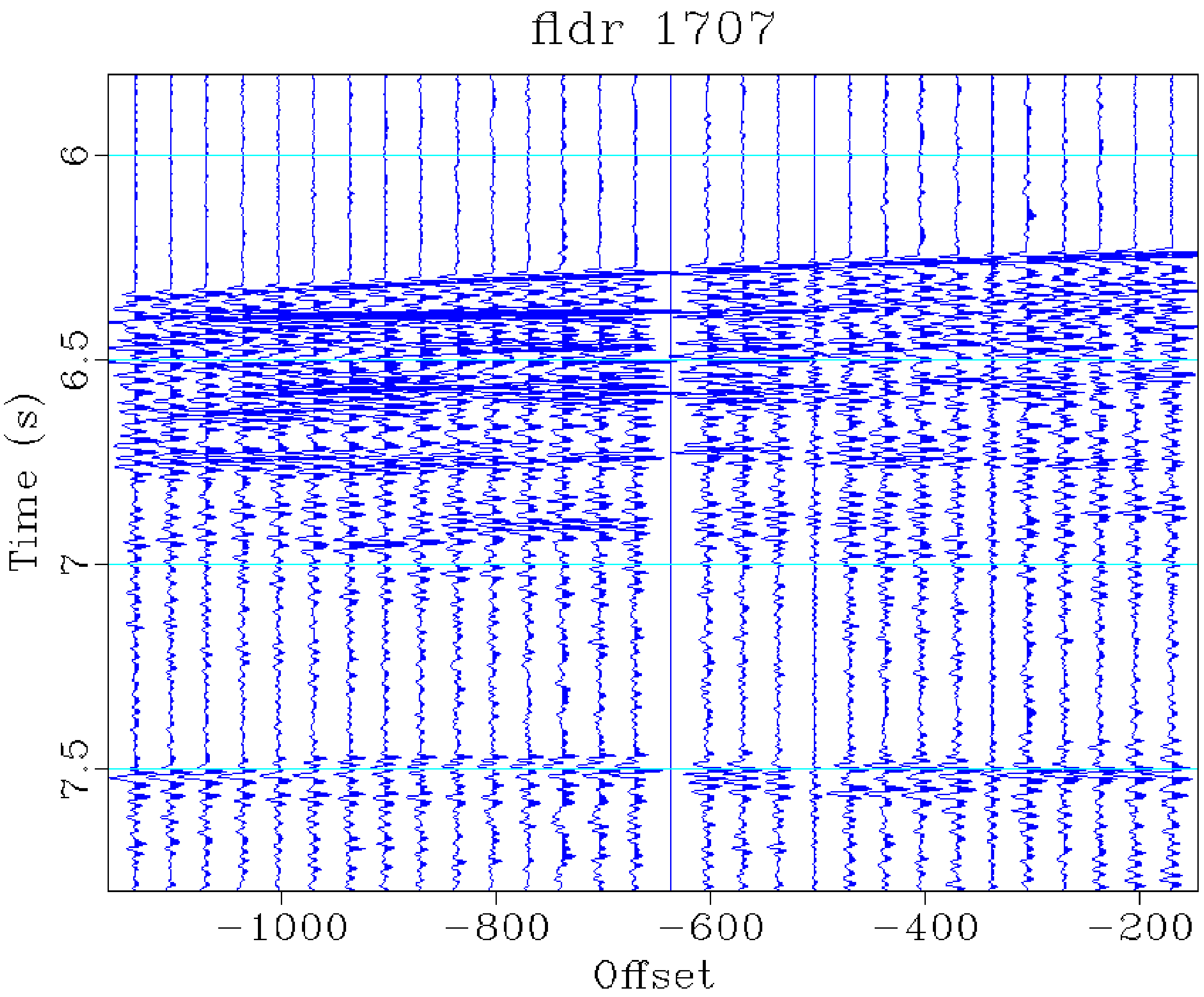
Display individual shots

```
Flow('mask1707','smask','window n2=1 min2=1707')
Flow('shot1707','shots2 mask1707',
        '''
        window n3=1 min3=1707 squeeze=n |
        headerwindow mask=${SOURCES[1]}
        ''')
Flow('offset1707','offset mask1707',
        '''
        window n3=1 min3=1707 squeeze=n |
        headerwindow mask=${SOURCES[1]}
        ''')
Result('shot1707','shot1707 offset1707',
          '''
          window min1=5.8 max1=7.8 |
          wiggle xpos=${SOURCES[1]}  yreverse=y transp=y poly=y
          title="fldr 1707" label2=Offset
          ''')
```

Three kinds of information:

1. Data
2. Offset
3. mask

$scons shot1707.view

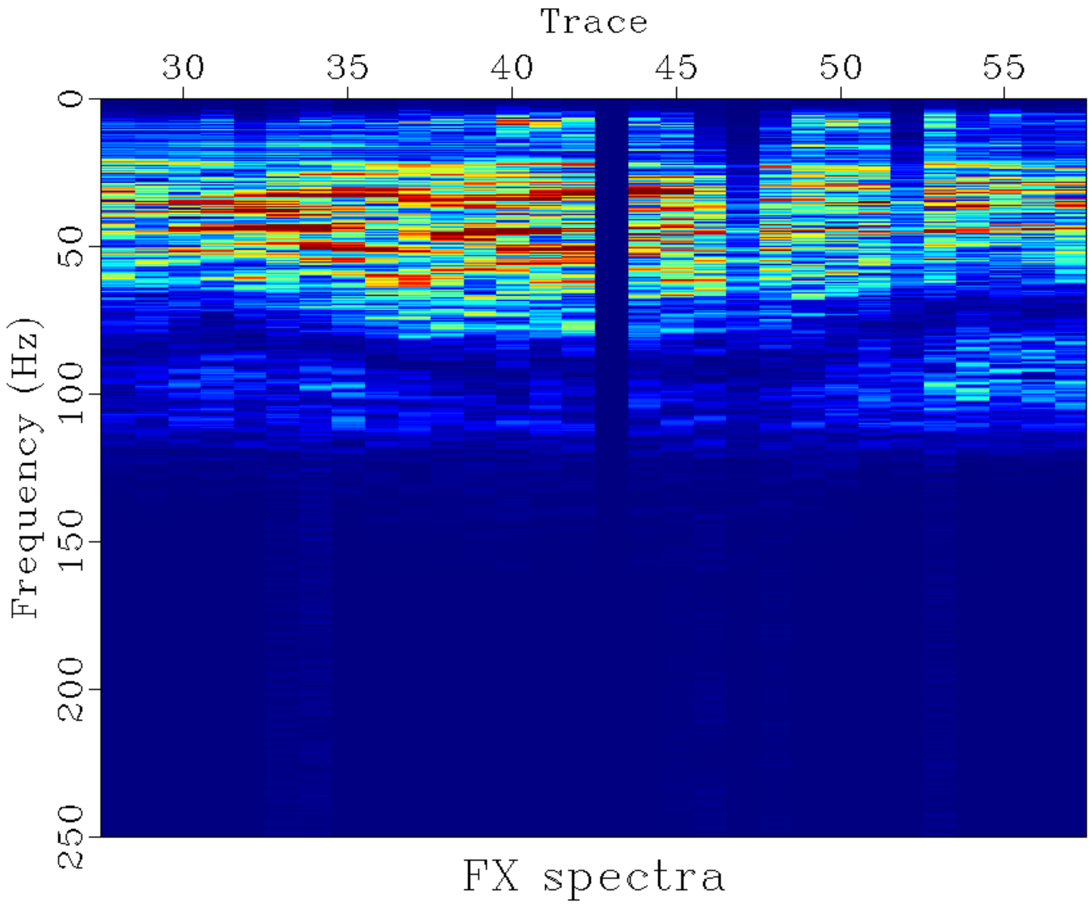


Initial signal analysis

Flow('fx1707','shot1707','spectra')

Result('fx1707', 'grey color=j allpos=y title="FX spectra" ')

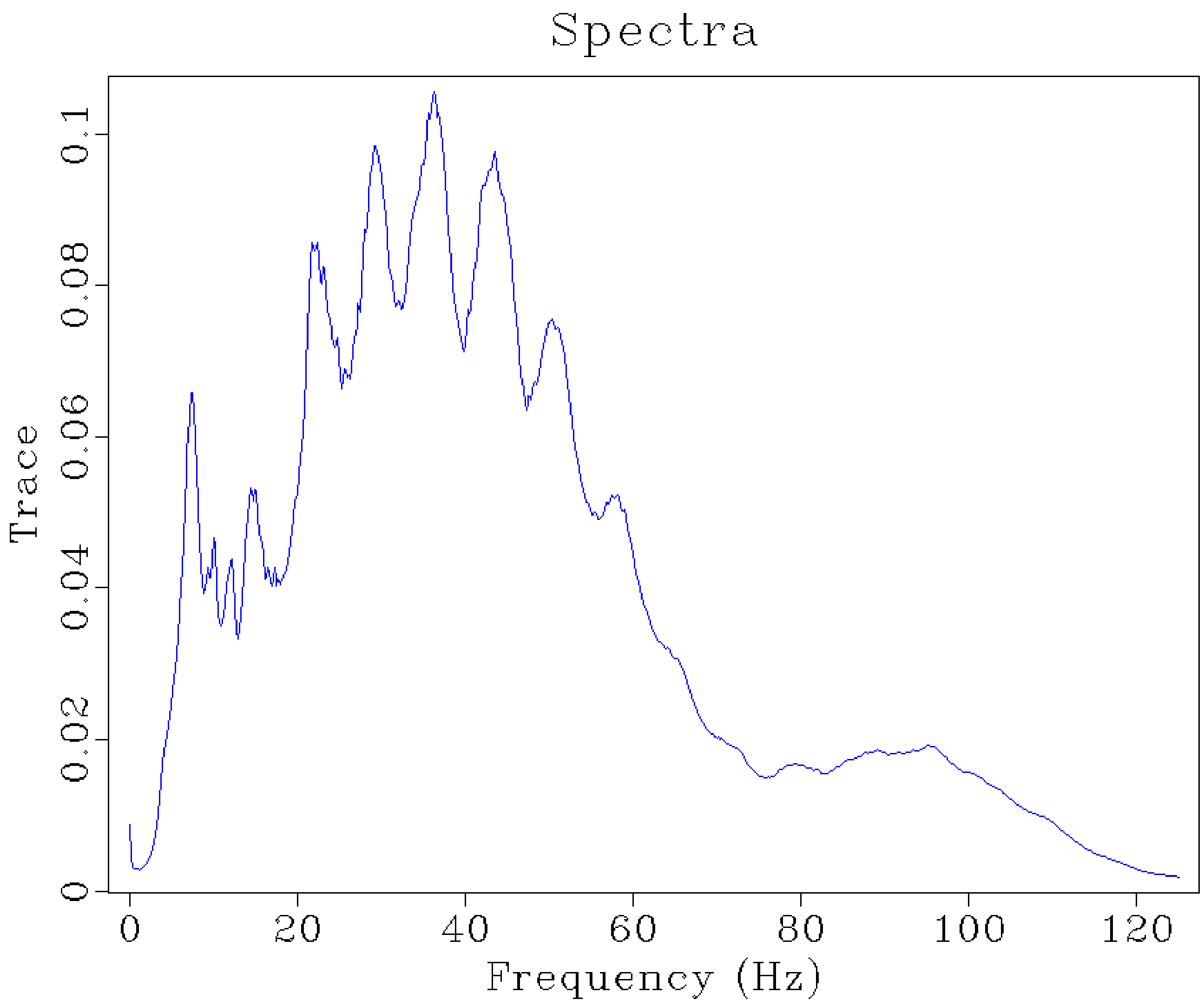
$scons shot1707.view



Subsampling with anti-aliasing filter

```
# Resample to 4 ms
Flow('subsamples','shots',
        '''
        bandpass fhi=125 | window j1=2
        ''')
Result('spectra','subsamples',
         'spectra all=y | graph title="Average Spectra" ')
```

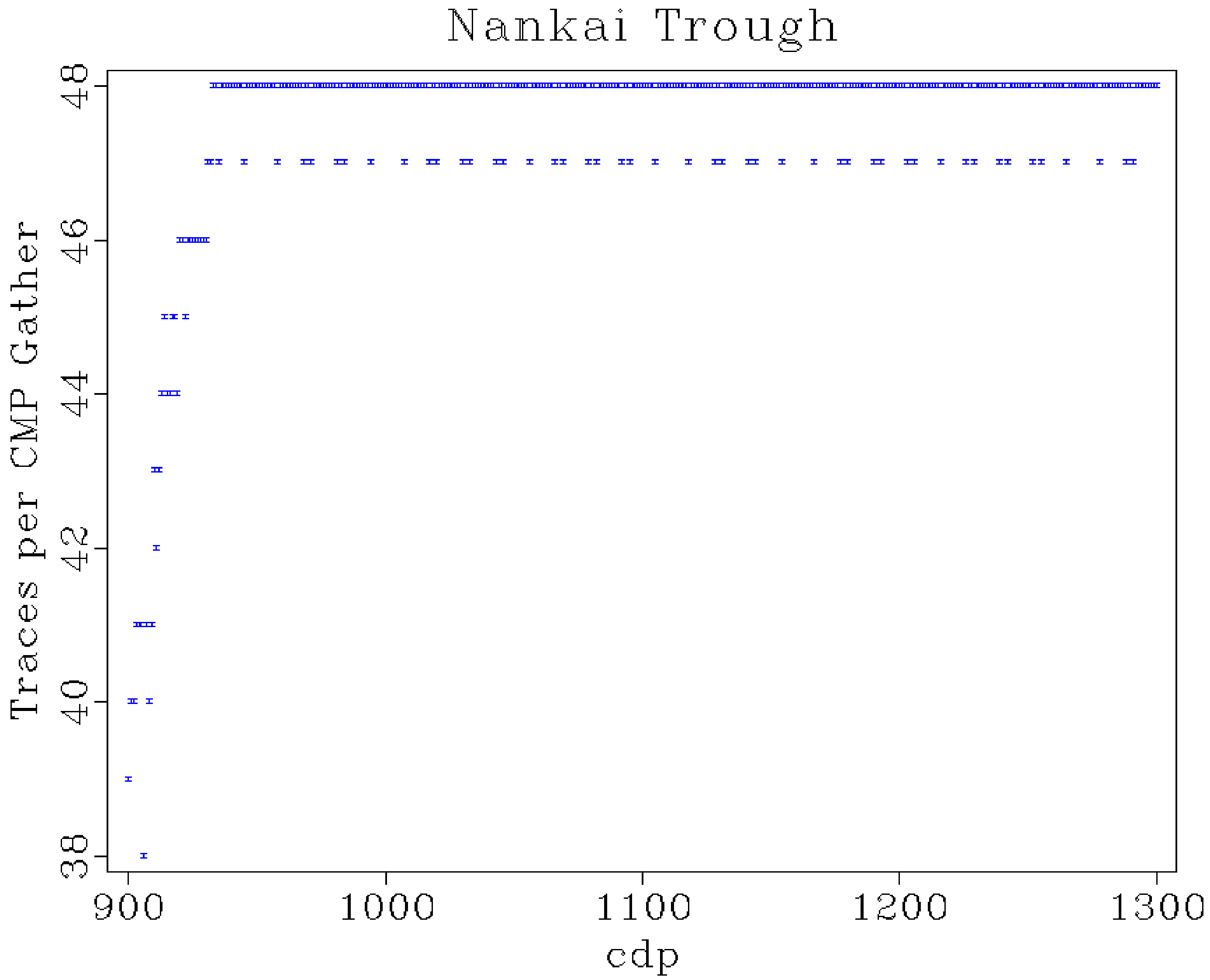
$scons spectra.view



Convert from CSP to CMP

```
# Extract CMPs
Flow('cmps cmask', 'subsamples',
        '''
        intbin mask=${TARGETS[1]} xk=tracf yk=cdp
        ''')
Flow('offs','tshots',
        '''
        window n1=1 f1=11 squeeze=n | dd type=float |
        intbin xk=tracf yk=cdp head=$SOURCE
        ''')
Result('cmask',
         '''
         dd type=float | stack axis=1 norm=n |
         graph symbol=x title="Nankai Trough"
         label2="Traces per CMP Gather"
         ''')
```

$scons cmask.view



Quality control

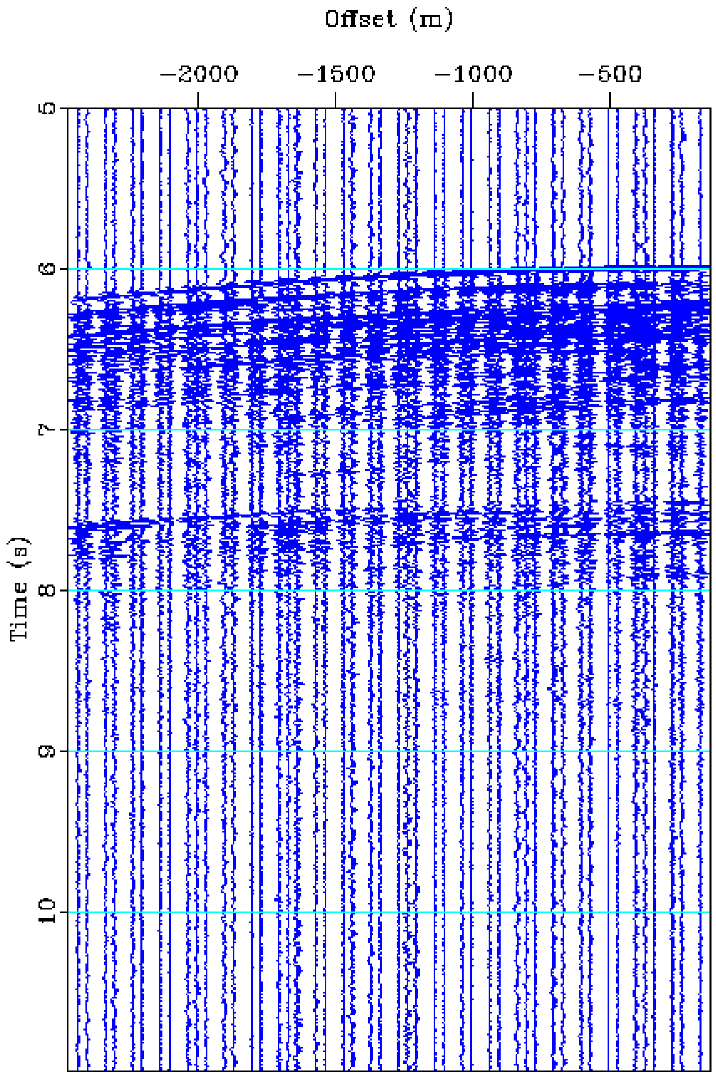
```
# Apply t^2 gain
Flow('cmpt','cmps', 'pow pow1=2')
Flow('mask','cmask','window n2=1 min2=1280')
Flow('cmp1','cmps mask',
    '''
    window n3=1 min3=1280 |  headerwindow mask=${SOURCES[1]}
    ''')

Flow('cmpt1','cmpt mask',
    '''
    window n3=1 min3=1280 |  headerwindow mask=${SOURCES[1]}
    ''')
Flow('off','offs mask',
    '''
    window n3=1 min3=1280 squeeze=n |  headerwindow mask=${SOURCES[1]}
    ''')
```

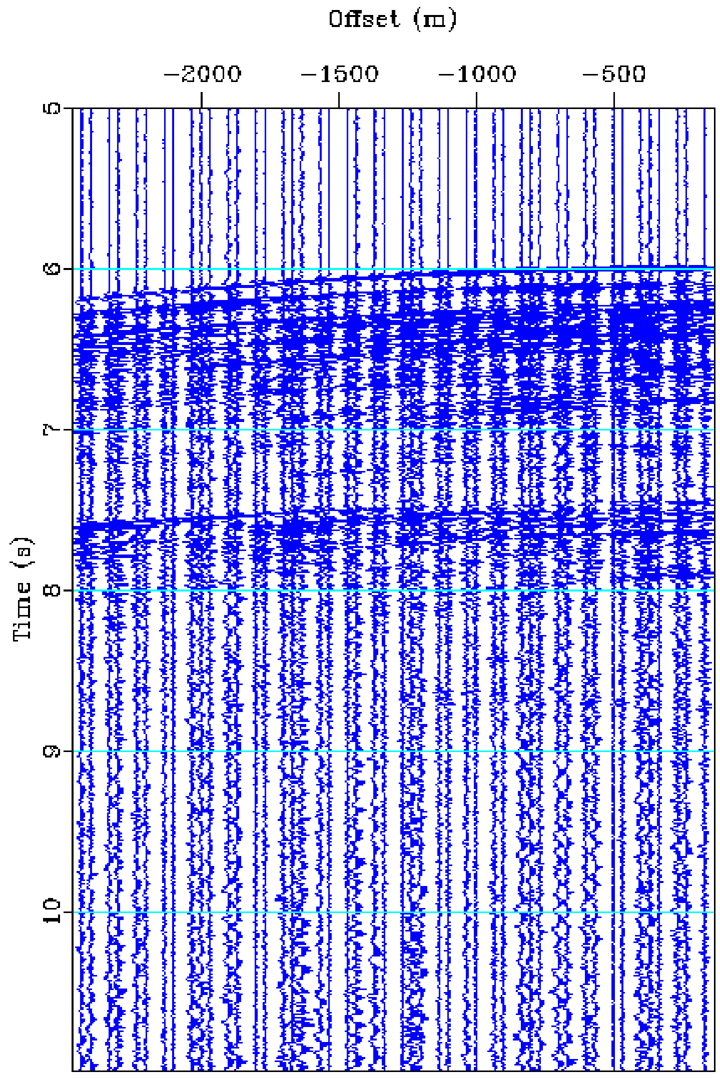
Quality control

```
Plot('cmp1','cmp1 off',
     '''
     wiggle xpos=${SOURCES[1]} title="CMP 1280"
     yreverse=y transp=y poly=y label2=Offset
     unit2=m  wherexlabel=t wheretitle=b
     ''')
Plot('cmpt1','cmpt1 off',
     '''
     wiggle xpos=${SOURCES[1]} title="CMP 1280 (Gained)"
     yreverse=y transp=y poly=y label2=Offset unit2=m
     wherexlabel=t wheretitle=b
     ''')
Result('cmp','cmp1 cmpt1','SideBySideAniso')
```

$scons cmp.view



CMP 1280



CMP 1280 (Gained)

Initial velocity analysis

```
# Velocity analysis and NMO
Flow('vscan','cmpt1 off mask',
        '''
        vscan half=n offset=${SOURCES[1]} mask=${SOURCES[2]}
        v0=1400 nv=101 dv=10 semblance=y
        ''')
Plot('vscan','grey color=j allpos=y title="Velocity Scan" unit2=m/s')

Flow('pick','vscan',
        'mutter inner=y half=n t0=5 x0=1400 v0=75 | pick v0=1500 rect1=25')
Plot('pick',
       '''
       graph transp=y yreverse=y plotcol=7 plotfat=3
       pad=n min2=1400 max2=2400 wanttitle=n wantaxis=n
       ''')
Plot('vscanp','vscan pick','Overlay')
```

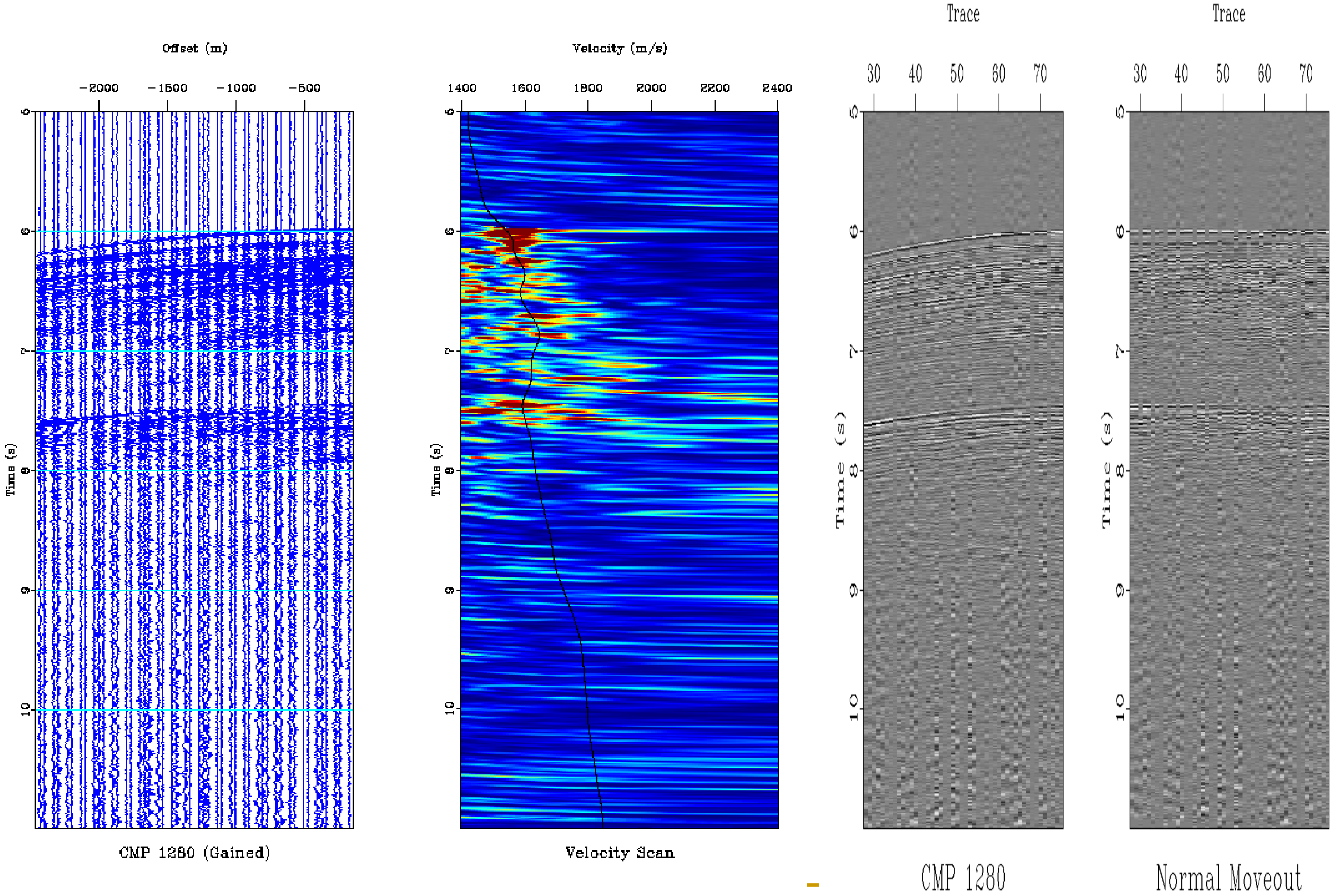
Initial velocity analysis

```
Flow('nmo','cmpt1 off mask pick',
	'''
	nmo half=n offset=${SOURCES[1]} mask=${SOURCES[2]}
	velocity=${SOURCES[3]}
	''')

Plot('tpowg','cmpt1','grey title="CMP 1280" labelsz=12 titlesz=18')
Plot('nmog','nmo','grey title="Normal Moveout" labelsz=12 titlesz=18')
Plot('nmo','tpowg nmog','SideBySideAniso')

Result('vscan','cmpt1 vscanp nmo','SideBySideAniso')
```

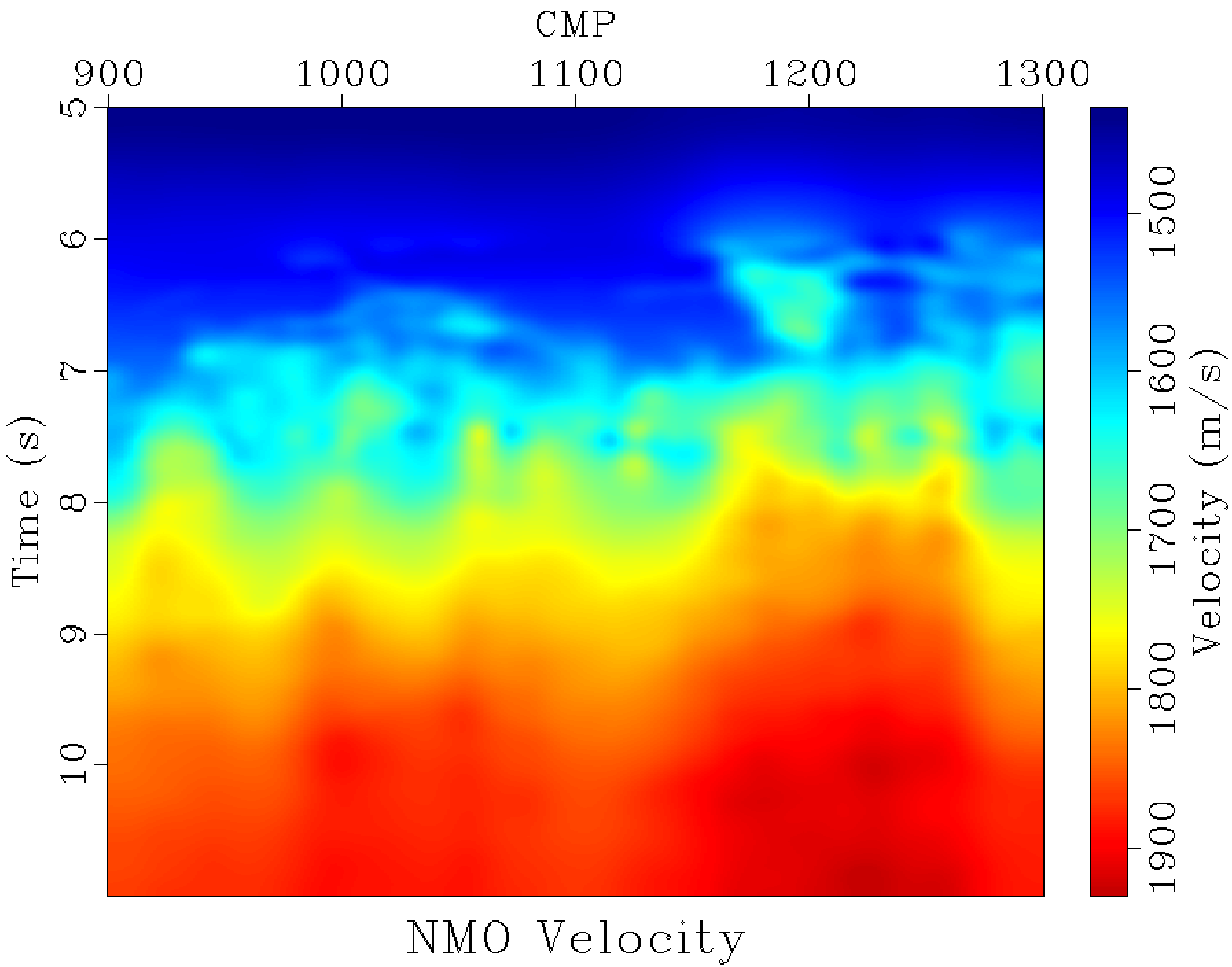
$scons vscan.view



Velocity analysis

```
# Apply to all CMPs
Flow('vscans','cmpt offs cmask',
        '''
        vscan half=n offset=${SOURCES[1]} mask=${SOURCES[2]}
        v0=1400 nv=101 dv=10 semblance=y nb=5
        ''')
Flow('picks','vscans',
        '''
       mutter inner=y half=n t0=5 x0=1400 v0=75 |
       pick v0=1500 rect1=25 rect2=10
       ''')
Result('picks',
          '''
          window | grey color=j mean=y scalebar=y title="NMO Velocity"
          label2=CMP barreverse=y barlabel=Velocity barunit=m/s
          ''')
```

$scons picks.view



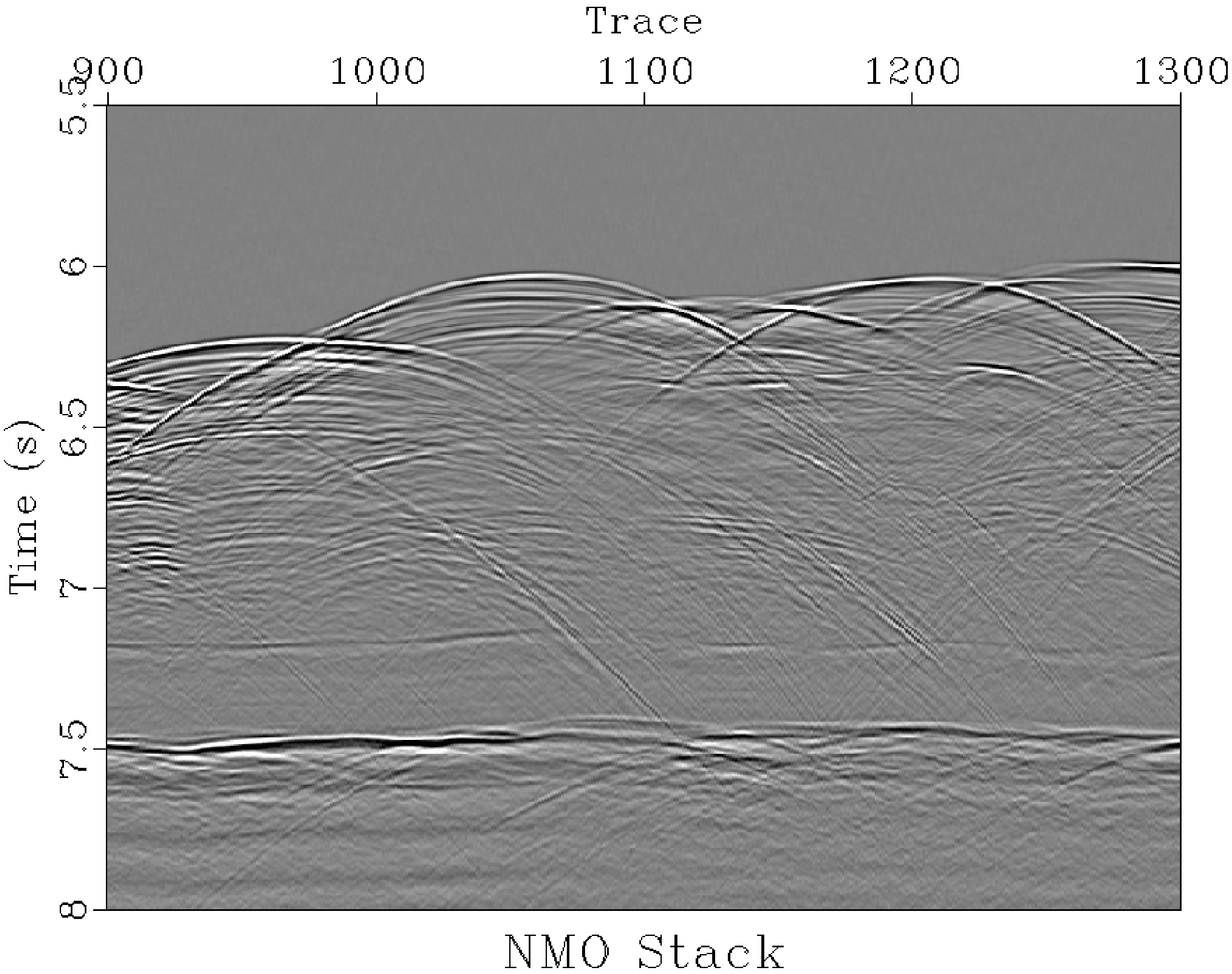
NMO and brute stacking

```
Flow('nmos','cmpt offs cmask picks',
     '''
     nmo half=n offset=${SOURCES[1]} mask=${SOURCES[2]}
     velocity=${SOURCES[3]}
     ''')

Flow('stack','nmos','stack')

Result('stack',
       'window min1=5.5  max1=8.0 |  grey title="NMO Stack" ')
```

$scons stack.view

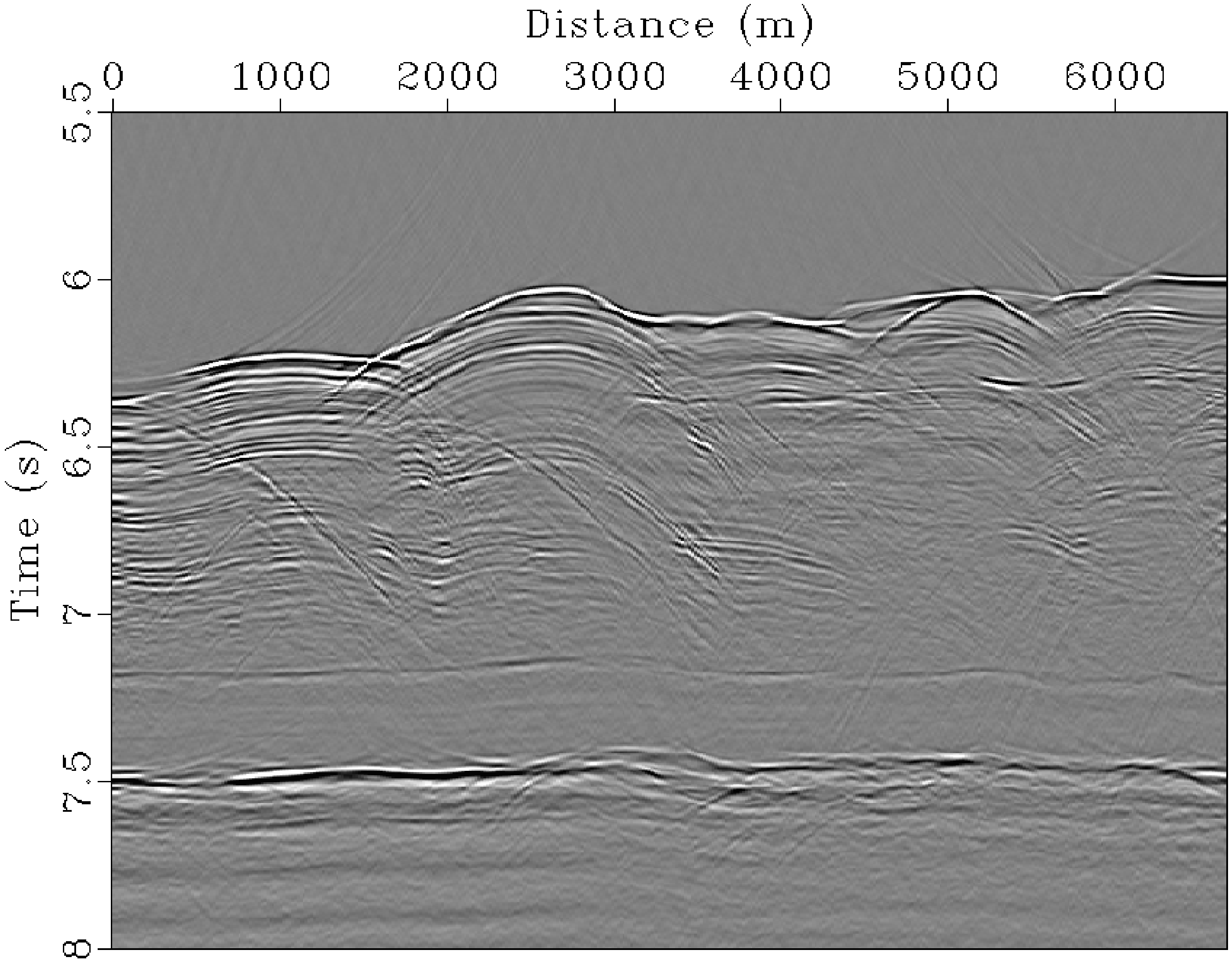


Poststack time migration

```
# Velocity continuation
Flow('first','stack',
      '''
     put o2=0 d2=16.667 label2=Distance unit2=m |
     cosft sign2=1 | pad beg1=1250 |
     stolt vel=1400
     ''')

Result('first',
       '''
       window min1=5.5 max1=8.0 |
       cosft sign2=-1 |
       grey title="Migration with 1400 m/s"
       ''')
```

$scons first.view



Migration with 1400 m/s

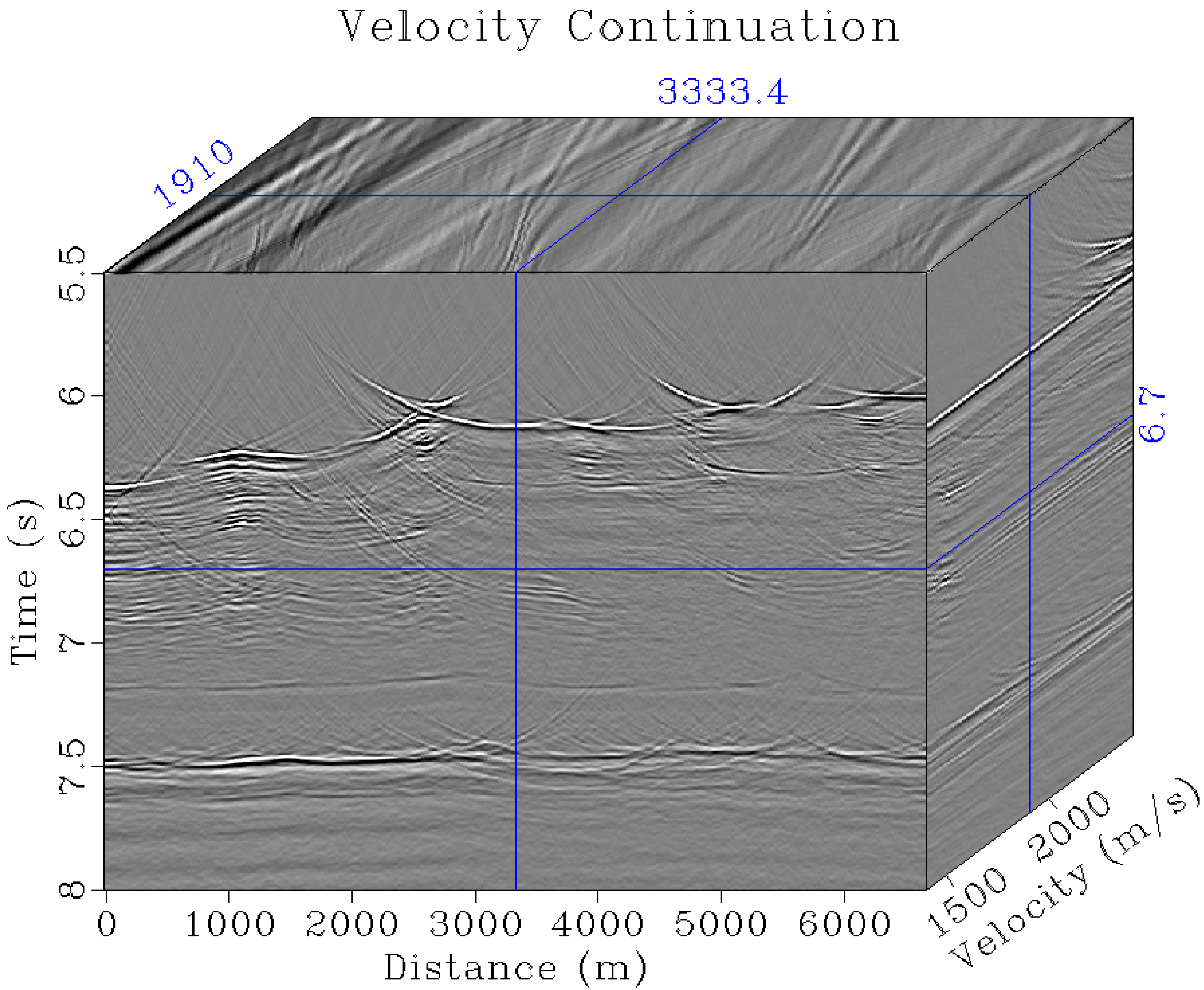
Poststack time migration

```
Flow('velcon','first',
     '''
     spray axis=2 n=1 o=0 d=1 |
     put o4=0 |
     fourvc nv=101 dv=10 v0=1400 verb=y |
     window min1=5.5 max1=8.0 |
     transp plane=23 memsize=1000 |
     cosft sign2=-1
     ''')
Result('velcon',
       '''
       byte gainpanel=a |
       grey3 frame1=300 frame2=200 frame3=50
       point1=0.8 point2=0.8 flat=n
       title="Velocity Continuation"
       ''')
Result('movie','velcon','grey title="velocity continuation"')
```

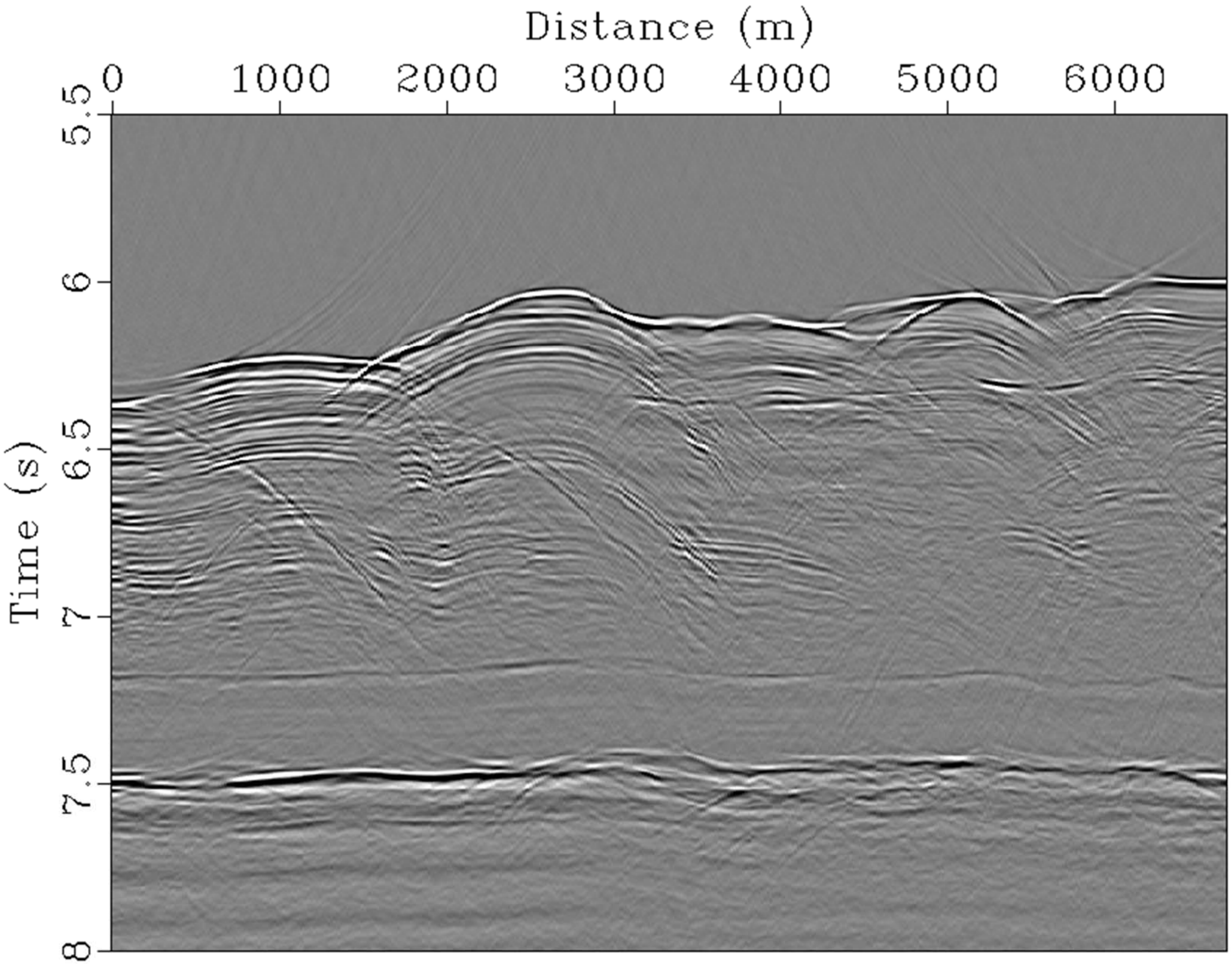
Reference:

Fomel S., 2003, Time-migration velocity analysis by velocity continuation, Geophysics, 68 (5): 1662-1672

$scons velcon.view



$scons velcon.view



velocity continuation

1410

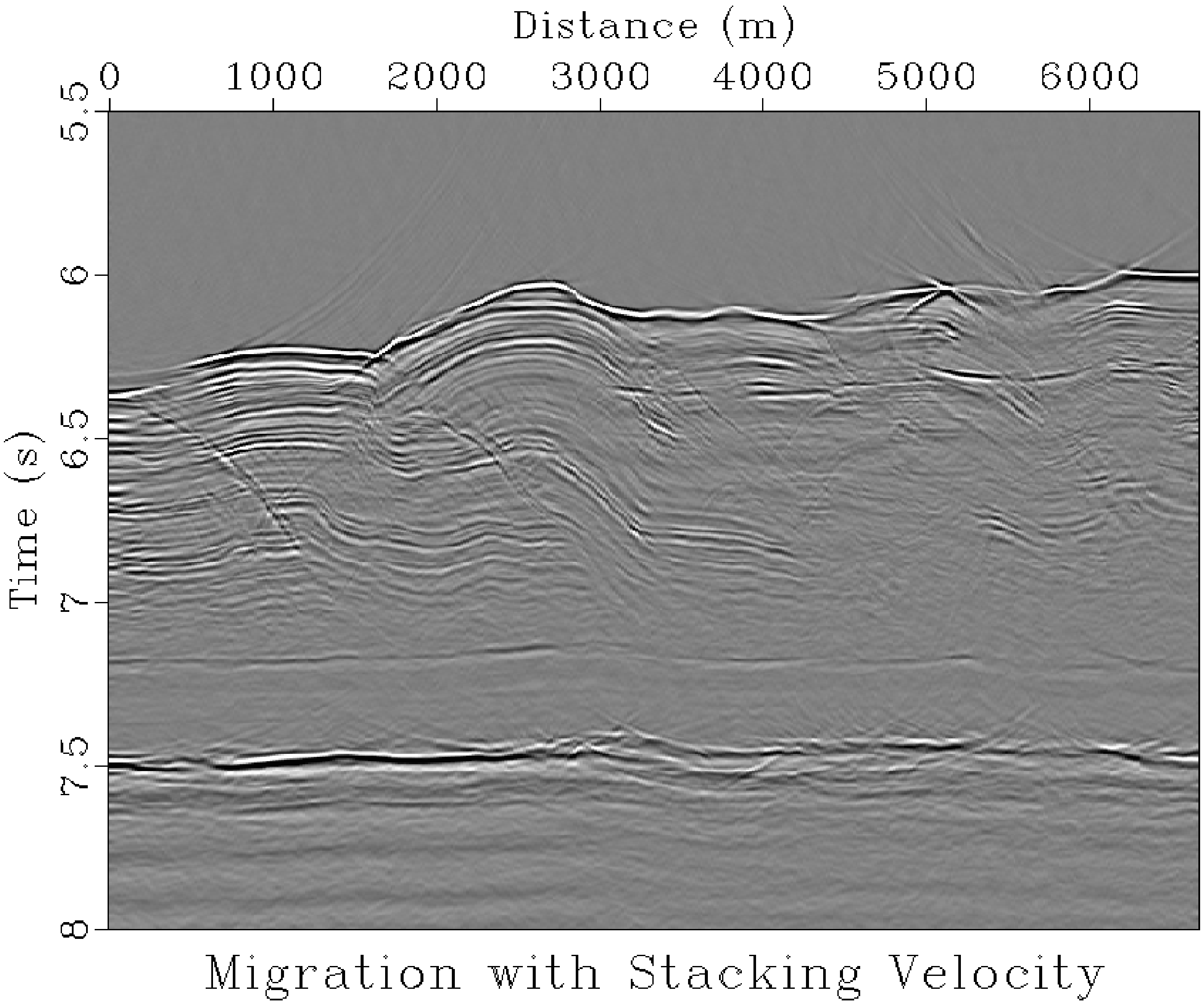
Poststack time migration

```
# Migration with the stacking velocity
Flow('mpick','picks','window min1=5.5 max1=8.0')
Flow('mstack','velcon mpick',
        '''
        transp plane=23 memsize=1000 |
        slice pick=${SOURCES[1]}
        ''')
Result('mstack',
         'grey title= "Migration with Stacking Velocity" ')

End()
```

Always stop here!

$scons mstack.view



Try “scons view” to see all figures one by one.

Add more modules as you like to get a better result.

Beyond seismic and Enjoy!