MADAGASCAR
reproducible experiments

Paul Sava

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

scons

- build system (software construction)
- written in python
- configuration files are Python scripts
- provides dependency analysis
- allows parallel builds
- works cross-platform

python

- dynamic programming language
- clear, readable syntax
- provides full modularity
- integrates with other languages
- free and open-source

python primer

- strings
- lists
- tuples
- dictionaries
- loops
- conditionals
- functions
- modules

python primer: strings

```
a='StPetersburg'
len(a)
a[0]
a[4:7]

b=a+' '+'workshop'
print b

c=b+2014
c=b+' '+str(2014)
print c
```

python primer: lists

```
d = ['StPetersburg', 'workshop']
len(d)
print d[0]
print d[1]

d.append('2014')
print d
```

python primer: tuples

```
t = ('StPetersburg', 'workshop')

t = t + (2014,)
print t
```

python primer: dictionaries

```
e={'what':'workshop','where':'StPetersburg','when':2014
print e
print e['where']+' '+e['what']+' '+str(e['when'])

f=dict(what='workshop',where='Melbourne',when=2013)
print f
print f['where']+' '+f['what']+' '+str(f['when'])
```

python primer: loops

```
for i in range(len(a)):
    print a[i]

for i in range(len(d)):
    print d[i]

for i in t:
    print i

for key in e.keys():
    print key,e[key]
```

python primer: conditionals

```
for k in range(5):
    if k < 2:
        print k, '<2'
    else:
        print k, '>=2'

try:
    b+2014
except:
    print "error!"
```

python primer: functions

```
def m8rschool(year):
    workshops=dict(StPetersburg=2014,Melbourne=2013)
    for key in workshops.keys():
        if workshops[key]==year:
            return key

print m8rschool(2014)
```

python primer: modules

```
def increment(a,b=5):
    return a+b
```

```
import math
x=math.sqrt(increment(4))
print x
```

MADAGASCAR functions

`Fetch(rsfFile,dirName)`

download files from a server

MADAGASCAR functions

```
Flow(outFile,inFile,command)
```

make output from input using the command

MADAGASCAR functions

```
Plot(vplFile,rsfFile,command)
```

make plot in local directory

MADAGASCAR functions

```
Result(vplFile,rsfFile,command)
```

make plot in `Fig` directory

MADAGASCAR commands

```
scons

scons blah.view

scons view

scons blah.lock

scons lock

scons -c

scons -n
```

example

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

```
$EDITOR SConstruct
```

import module

```
from rsf.proj import *
```

setup parameter dictionary

```
par = dict(
    nx=200,   ox=-10, dx=0.1,   lx='x', ux='km',
    nz=200,   oz=-10, dz=0.1,   lz='x', uz='km',
    nt=1000,  ot=0,   dt=0.01,  # traveltime
    ng=1801,  og=-90, dg=0.1,   # angle
    sig=1.5,                    # stdev
    xsou=0,  zsou=-10,          # coords
    labelattr="titlefat=3 labelsz=5 labelfat=3"
    )
```

expand parameter dictionary

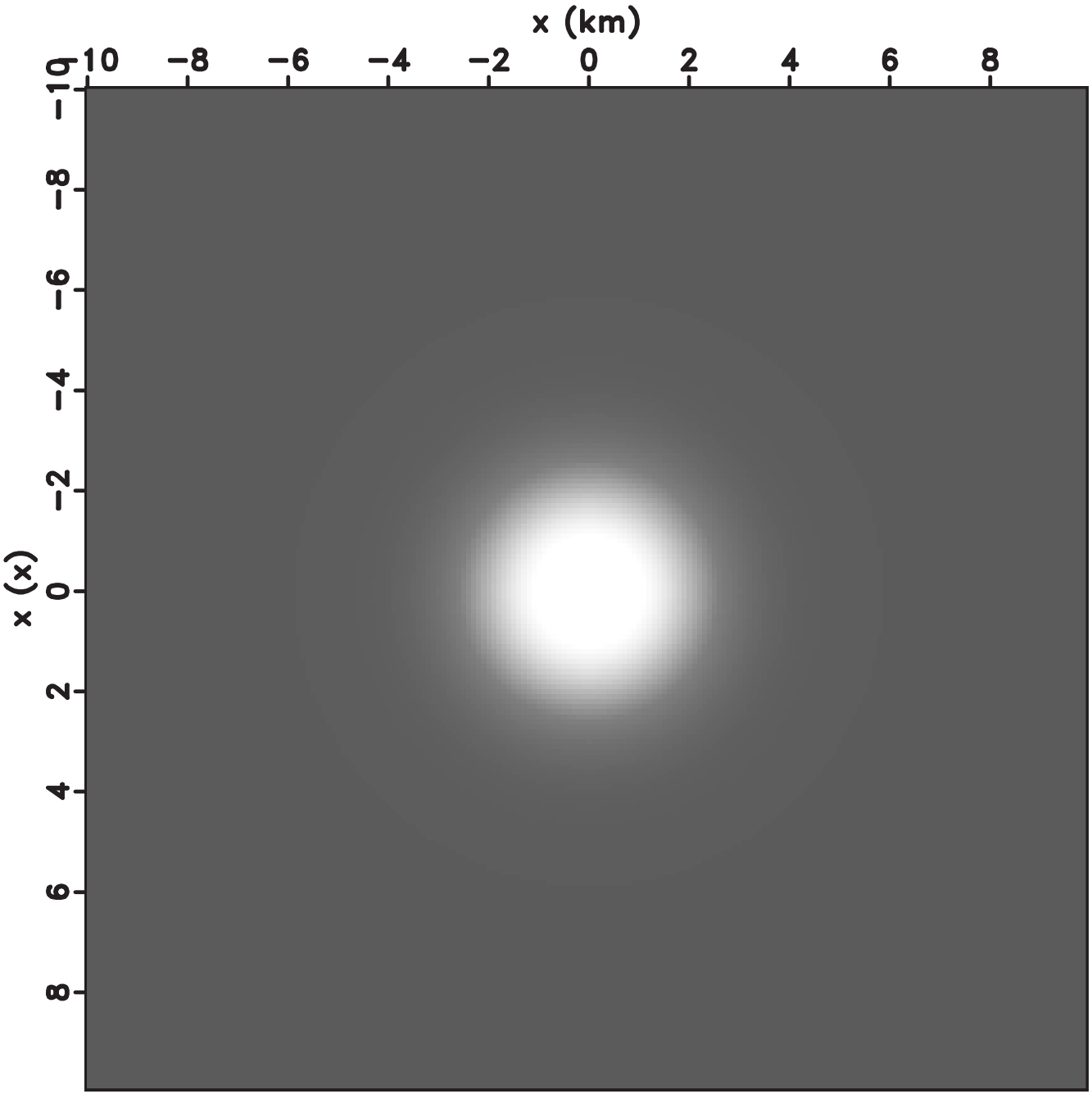
```
par['xmin']=par['ox']
par['xmax']=par['ox'] + (par['nx']-1)*par['dx']
par['zmin']=par['oz']
par['zmax']=par['oz'] + (par['nz']-1)*par['dz']
par['ratio']=(par['nz']-1)*par['dz']/(par['nx']-1)/par[
```

make Gaussian function

```
Flow('gg',None,
     '''
     math output="exp(-(x1^2+x2^2)/(2*%(sig)g^2))"
     n1=%(nz)d d1=%(dz)g o1=%(oz)g
     n2=%(nx)d d2=%(dx)g o2=%(ox)g |
     put label1=%(lz)s unit1=%(lz)s
         label2=%(lx)s unit2=%(ux)s
     '''%par)
```

plot 2D Gaussian function

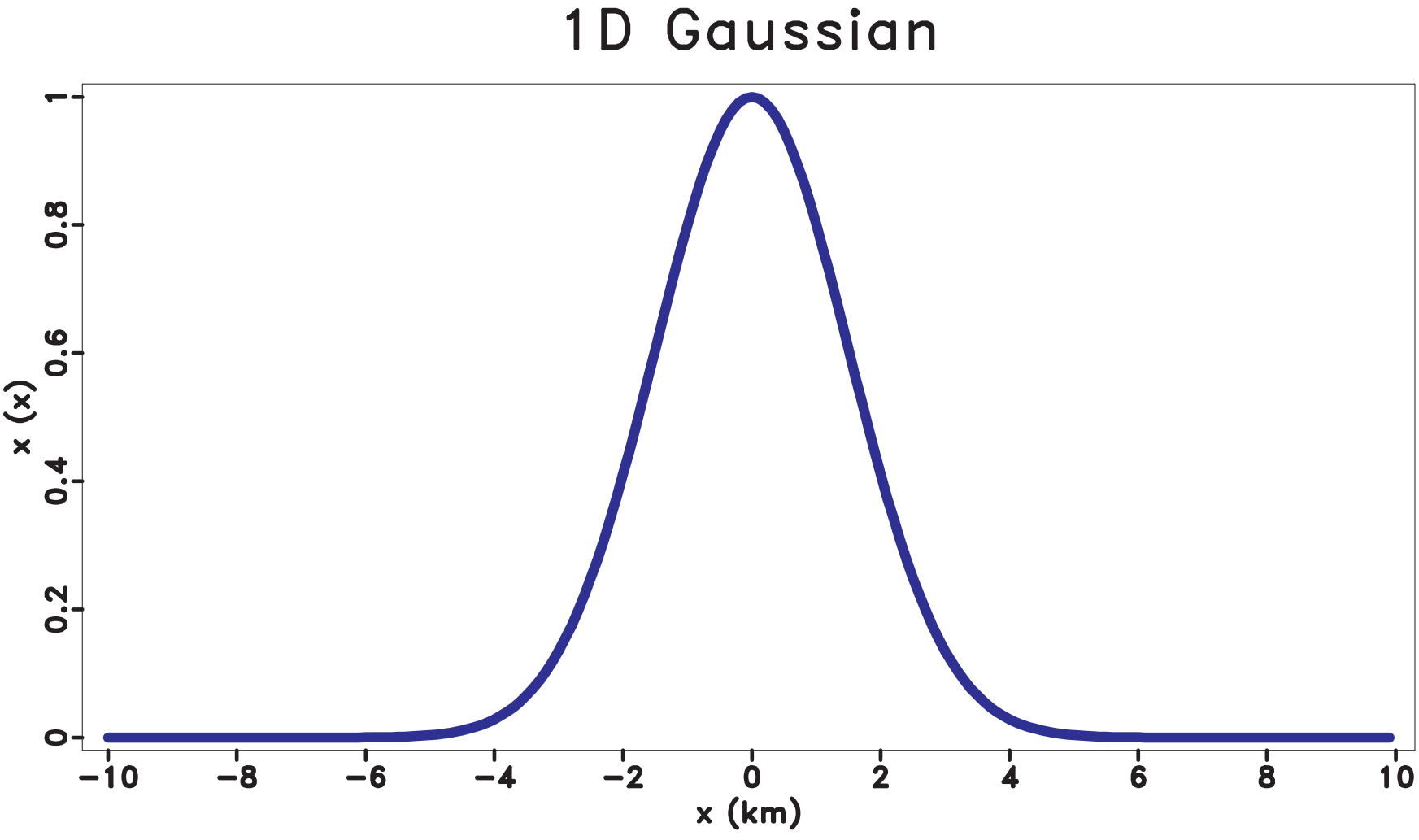
```
Result('gg',
       '''
       grey title="2D Gaussian"
       pclip=100 screenratio=%(ratio)g
       %(labelattr)s
       '''%par)
```



2D Gaussian

plot 1D Gaussian function

```
Result('gg0','gg',
       '''
       window n1=1 min1=0 |
       graph title="1D Gaussian"
       plotfat=10 screenratio=0.5 screenht=7
       %(labelattr)s
       '''%par)
```


1D Gaussian
x (x)
0
0.2
0.4
0.6
0.8
1
−10
−8
−6
−4
−2
0
2
4
6
8
10
x (km)

make velocity

```
Flow('vel','gg',
     'math output="3.0-input"')
```

plot velocity

```
Plot('vel',
     '''
     grey title="" pclip=100 color=g
     screenratio=%(ratio)g mean=y %(labelattr)s
     '''%par)
```

make traveltimes

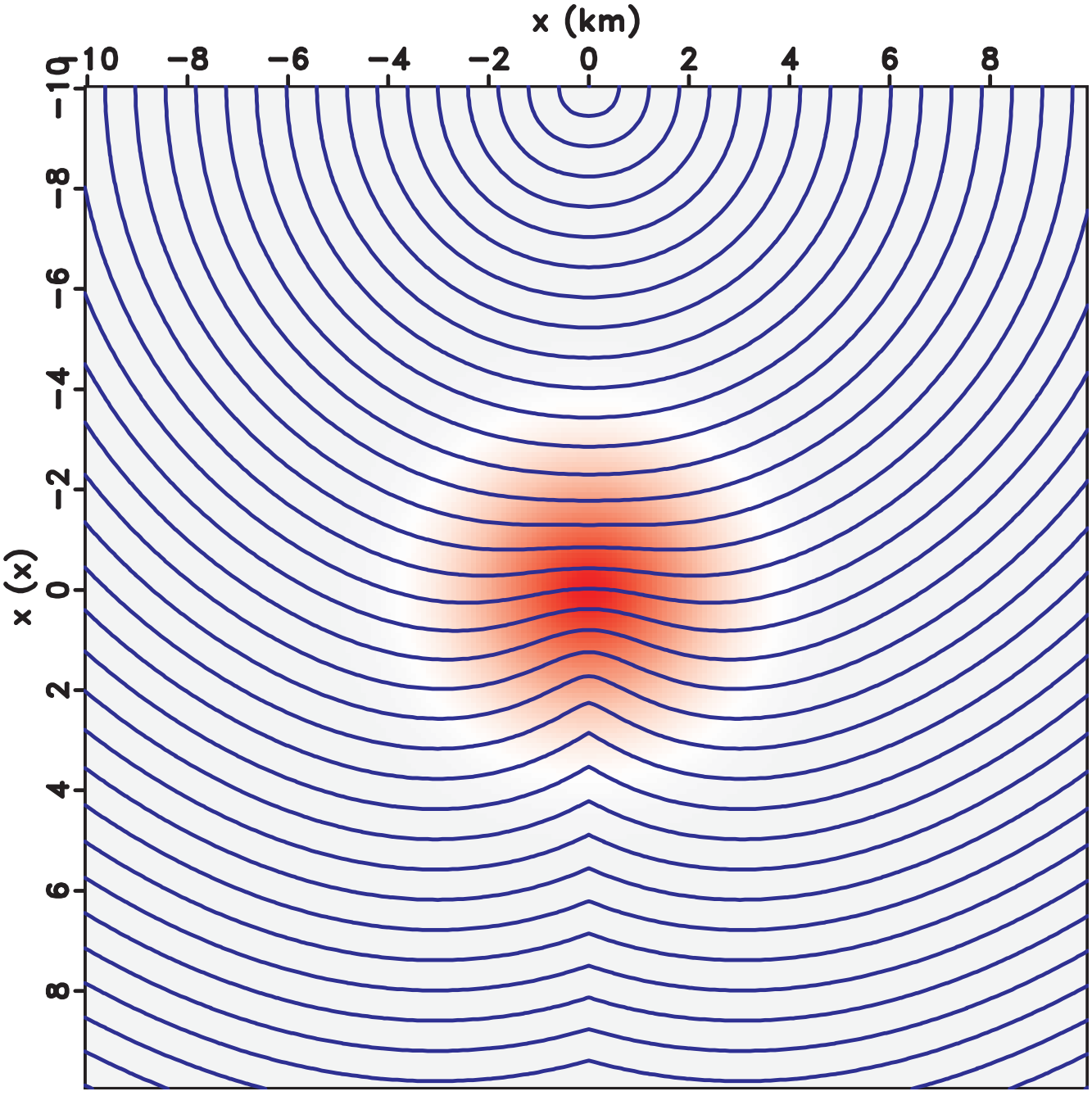
```
Flow(    'fme','vel',
         'eikonal zshot=%(zsou)g yshot=%(xsou)g'%par)
```

plot traveltimes

```
Plot(    'fme',fmeplot('',par))
Result('fme',['vel','fme'],'Overlay')
```

traveltime plot function

```
def fmeplot(custom,par):
    return '''
    contour
    title="" screenratio=%(ratio)g
    nc=100 plotcol=6 plotfat=3
    labelrot=n wantaxis=n wanttitle=n
    %(labelattr)s
    '''%par + custom
```


x (km)
−10
−8
−6
−4
−2
0
2
4
6
8
x (x)
−10
−8
−6
−4
−2
0
2
4
6
8

make rays/wavefronts

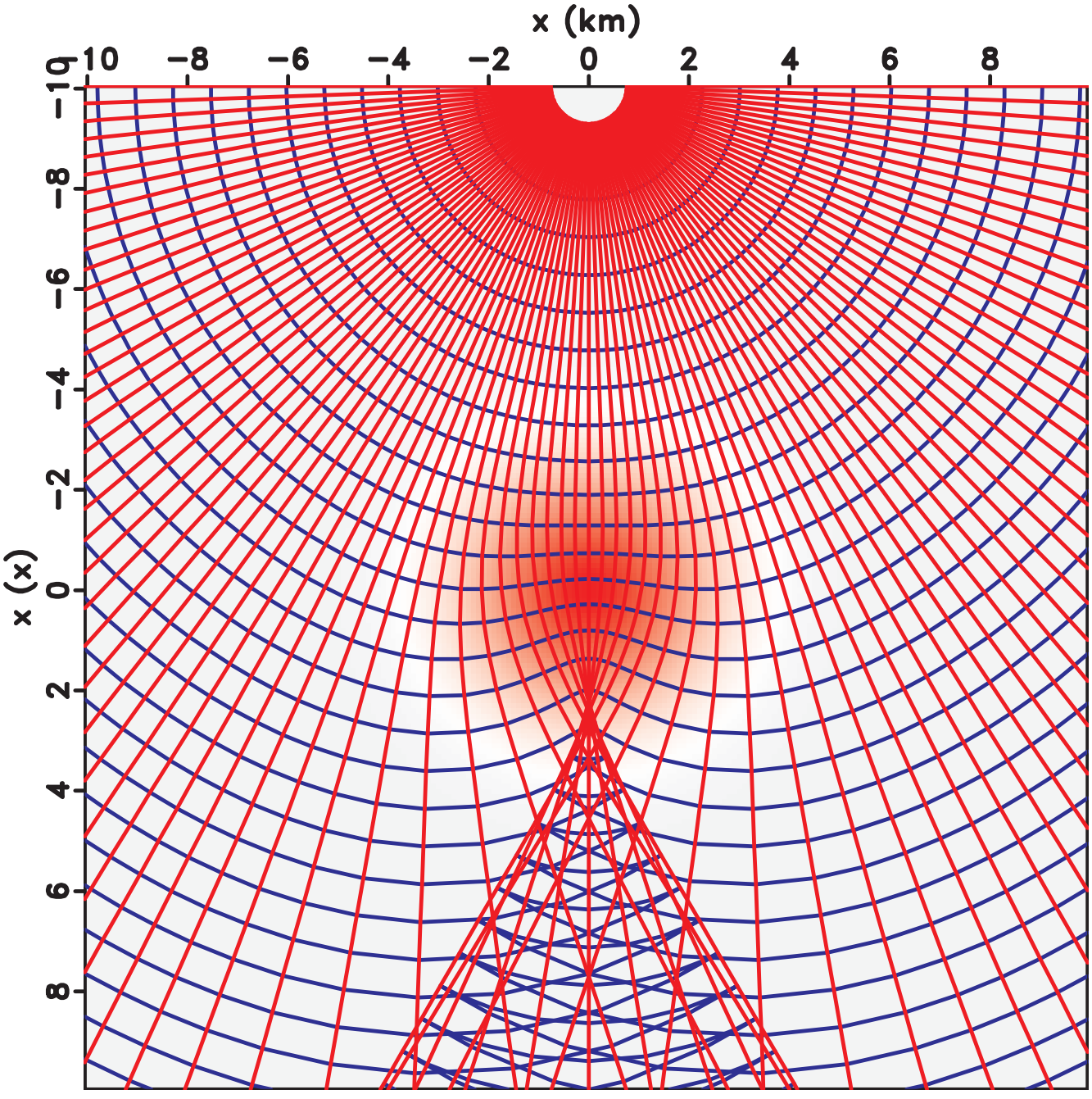
```
Flow('hwt','vel',
     '''
     hwt2d xsou=%(xsou)g zsou=%(zsou)g
     nt=1000 ot=0    dt=0.01
     ng=1801 og=-90 dg=0.1
     '''%par)
```

plot rays/wavefronts

```
Plot('ray','hwt',
     'transp | window f1=25 j1=25 j2=20 |'
    + hwtplot('plotcol=5',par))
Plot('wft','hwt',
     'window f2=50 j2=25 j1=20 |'
    + hwtplot('plotcol=6',par))
Result('hwt',['vel','wft','ray'],'Overlay')
```

rays/wavefronts plot function

```
def hwtplot(custom,par):
    return '''
    graph
    title="" screenratio=%(ratio)g
    plotcol=0 plotfat=3 wantaxis=n yreverse=y
    min1=%(zmin)g max1=%(zmax)g
    min2=%(xmin)g max2=%(xmax)g
    %(labelattr)s
    '''%par + custom
```


x (km)
−10
−8
−6
−4
−2
0
2
4
6
8
x (x)
−10
−8
−6
−4
−2
0
2
4
6
8

http://www.ahay.org

$RSF/book/rsf/school