Basic command-line usage

Paul Sava

Center for Wave Phenomena
Colorado School of Mines

psava@mines.edu

Presentation goal

- Describe basic MADAGASCAR program usage
- Describe file format
- Present simple processing flows

Madagascar

Madagascar is an open-source software package for geophysical data processing and reproducible numerical experiments. Its mission is to provide

- a convenient and powerful environment
- a convenient technology transfer tool

for researchers working with digital image and data processing.

Reproducible research

The technology developed using the MADAGASCAR project management system is transferred in the form of recorded processing histories, which become "computational recipes" to be verified, exchanged, and modified by users of the system.

Programs

- "sf" prefix
- program count: 392 on 4/16/2007
- documented by examples ("books")

Program list

sfdoc -k .

```
sfofpwd: Objective function of dip estimation with PWD filters.
sfinfill: Shot interpolation.
sfslice: Extract a slice using picked surface (usually from a stack or a semblance).
sfin: Display basic information about RSF files.
sfdmo: Kirchhoff DMO with antialiasing by reparameterization.
sfradstretch: Stretch of the time axis.
sflpef: Find PEF on aliased traces.
sfrefer: Subtract a reference from a grid.
sflevint: Leveler inverse interpolation in 1-D.
sfnoise: Add random noise to the data.
sfenvcorr: Local correlation with the envelope.
sfmsmiss: Multiscale missing data interpolation (N-dimensional).
sfconv: 1-D convolution.
sfdottest: Generic dot-product test for linear operators with adjoints
sfcut: Zero a portion of the dataset.
sfwiggle: Plot data with wiggly traces.
sfplanemis2: Missing data interpolation in 2-D using plane-wave destruction.
sfgraph: Graph plot.
sfpldb: Plot Debugger - convert vplot to ascii.
sfflat3: 3-D flattening (without picking).
sfexgr: Exact group velocity in VTI media
sfmodrefl: Normal reflectivity modeling.
sfmisif: Find MISSing Input values and Filter in 1-D.
sfspectra: Frequency spectra.
sfintbin: Data binning.
...
```

Self documentation

- Run program without arguments
- Find program purpose
- Find execution parameters
- Find execution examples

Example

sfspike

```
NAME
        sfspike
DESCRIPTION
        Generate simple data: spikes, boxes, planes, constants.
SYNOPSIS
        sfspike > spike.rsf mag= nsp=1 k#=[0,...] l#=[k1,k2,...] p#=[0,...] n#=
o#=(0,...) d#=(0.004,0.1,0.1,...) label#=(Time,Distance,Distance,...) unit#=[s,km,km,...] title=
PARAMETERS
        float   d#=(0.004,0.1,0.1,...)  sampling on #-th axis
        ints    k#=[0,...]      spike starting position  [nsp]
        ints    l#=[k1,k2,...]  spike ending position  [nsp]
        string  label#=(Time,Distance,Distance,...)     label on #-th axis
        floats  mag=    spike magnitudes  [nsp]
        int     n#=     dimension of #-th axis
        int     nsp=1   Number of spikes
        float   o#=(0,...)      origin on #-th axis
        floats  p#=[0,...]      spike inclination (in samples)  [nsp]
        string  title=  title for plots
        string  unit#=[s,km,km,...]     unit on #-th axis
USED IN
        bei/conj/causint
        bei/dpmv/matt
        bei/dpmv/yalei
        bei/dwnc/vofz
        bei/dwnc/phasemod
        bei/fdm/kjartjac
        bei/ft1/autocor
        bei/ft1/ft2d
...
```

psava@mines.edu

Program execution
single input, single output

< input.rsf **sfprog** *arguments* > output.rsf

- sfprog = MADAGASCAR program
- *arguments* = program arguments
- input from `stdin` (<)
- output to `stdout` (>)

psava@mines.edu

Demo

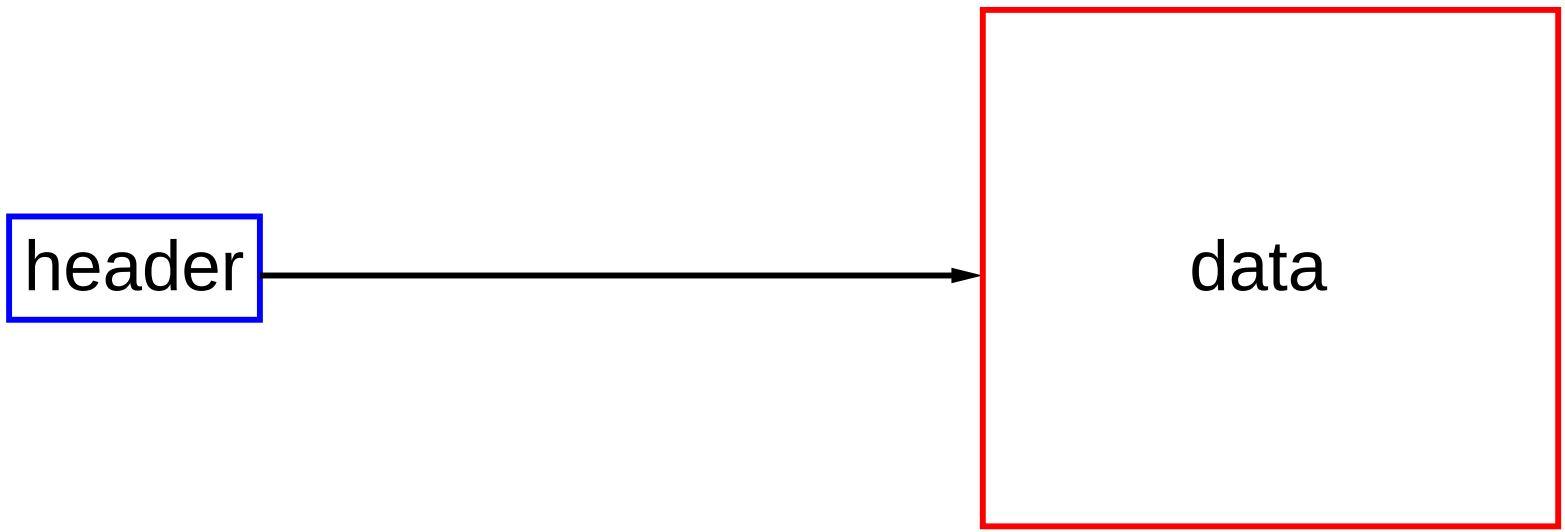
sfspike
n1=100 o1=0 d1=0.01
n2=50 o2=1000 d2=10
> file1.rsf

- Standard in: none
- Standart out: file1.rsf

File format

home file system

scratch file system



File format

Header:

- Text file (description of data)
- Description of regularly-sampled format
- Small, can be archived

File format

Binary:

- Binary file (actual data)
- Regularly-sampled data
  - native binary
  - XDR binary
- Large, can be stored on a different file system
- Path to binary set with environment variable `DATAPATH`

Example

sfin

```
NAME
        sfin
DESCRIPTION
        Display basic information about RSF files.
SYNOPSIS
        sfin info=true check=2. trail=true file1.rsf file2.rsf ...
COMMENTS
        n1,n2,... are data dimensions
        o1,o2,... are axis origins
        d1,d2,... are axis sampling intervals
        label1,label2,... are axis labels
        unit1,unit2,... are axis units

PARAMETERS
        float   check=2.        Portion of the data (in Mb) to check for zero values.
        bool    info=y [y/n]    If n, only display the name of the data file.
        bool    trail=y [y/n]   If n, skip trailing dimensions of  one
USED IN
        data/sigsbee/fs2B
        data/sigsbee/nfs2B
...
SOURCE
        filt/main/in.c
```

psava@mines.edu

Demo

sfin file1.rsf

```
file1.rsf:
    in="/scratch/file1.rsf@"
    esize=4 type=float form=native
    n1=100          d1=0.01         o1=0          label1="Time" unit1="s"
    n2=50           d2=10           o2=1000       label2="Distance" unit2="km"
        5000 elements 20000 bytes
```

Axes

Dataset

- header: *file1.rsf*
- binary: *in* = "*/scratch/file1.rsf@*"

Axis described by:

- *n*: number of samples
- *o*: sampling origin
- *d*: sampling rate (delta)
- *label*: axis label
- *unit*: axis unit

Demo

< file1.rsf **sfattr**

```
*****************************************
rms = 1
mean value = 1
norm value = 70.7107
variance = 0
standard deviation = 0
maximum value = 1 at 1 1
minimum value = 1 at 1 1
number of nonzero samples = 5000
total number of samples = 5000
*****************************************
```

Compatibility

- SEPlib: identical format
  In file1.rsf

- SU: use converters
  sfsegyread tape=file1.su su=y tfile=tfile.rsf
  endian=0 > file1.rsf

  sfsegywrite tape=file1.su su=y tfile=tfile.rsf
  endian=0 < file1.rsf

Demo

< file1.rsf **sfwindow** n2=25 min2=1200 > file2.rsf

sfin file1.rsf

```
file1.rsf:
    in="/scratch/file1.rsf@"
    esize=4 type=float form=native
    n1=100          d1=0.01         o1=0          label1="Time" unit1="s"
    n2=50           d2=10           o2=1000       label2="Distance" unit2="km"
        5000 elements 20000 bytes
```

sfin file2.rsf

```
file2.rsf:
    in="/scratch/file2.rsf@"
    esize=4 type=float form=native
    n1=100          d1=0.01         o1=0          label1="Time" unit1="s"
    n2=25           d2=10           o2=1200       label2="Distance" unit2="km"
        2500 elements 10000 bytes
```

Program execution
Multiple inputs, multiple outputs

< input.rsf **sfprog** *arguments*
label1=file1.rsf label2=file2.rsf ...
> output.rsf

- sfprog = MADAGASCAR program
- Input from `stdin` (<)
- Output to `stdout` (>)
- file1.rsf can be open for input and/or output
- file2.rsf can be open for input and/or output

Example

sfafmod

```
NAME
        sfafmod
DESCRIPTION
        Time-domain acoustic FD modeling.
SYNOPSIS
        sfafmod < Fw.rsf vel=Fv.rsf den=Fe.rsf > Fd.rsf sou=Fs.rsf rec=Fr.rsf wfl=Fu.rsf
verb=false abc=false snap=false free=false dens=false jsnap=nt nbz=nop nbx=nop tz=0.025 tx=0.025

PARAMETERS
        bool    abc=n [y/n]
        bool    dens=n [y/n]
        bool    free=n [y/n]
        int     jsnap=nt
        int     nbx=nop
        int     nbz=nop
        bool    snap=n [y/n]
        float   tx=0.025
        float   tz=0.025
        bool    verb=n [y/n]
USED IN
        gti/fdmod/dens
        gti/fdmod/scat
...
```

Demo

```
< wavelet.rsf sfafmod
   vel=velocity.rsf
   den=density.rsf
   sou=sources.rsf
   rec=receivers.rsf
   wfl=wavefield.rsf
> data.rsf
```

Pipes

- MADAGASCAR programs can be piped
- Stdout from one program is stdin for the next
- No intrinsic limit for the number of pipes
- Different from SEPlib's pipes

Demo

< file1.rsf
sfwindow n2=25 min2=1200 |
sftransp
> file3.rsf
sfin file1.rsf file3.rsf

```
file1.rsf:
    in="/scratch/file1.rsf@"
    esize=4 type=float form=native
    n1=100          d1=0.01         o1=0          label1="Time" unit1="s"
    n2=50           d2=10           o2=1000       label2="Distance" unit2="km"
        5000 elements 20000 bytes
```

```
file3.rsf:
    in="/scratch/file3.rsf@"
    esize=4 type=float form=native
    n1=25           d1=10           o1=1200       label1="Distance" unit1="km"
    n2=100          d2=0.01         o2=0          label2="Time" unit2="s"
        2500 elements 10000 bytes
```

Useful utilities

- simple math operations
- basic 1D, 2D, 3D plotting

Example

sfmath

```
NAME
        sfmath
DESCRIPTION
        Mathematical operations on data files.
SYNOPSIS
        sfmath > out.rsf type= unit= output=
COMMENTS

        Known functions: cos,  sin,  tan,  acos,  asin,  atan,
                         cosh, sinh, tanh, acosh, asinh, atanh,
                         exp,  log,  sqrt, abs, conj (for complex data).

        sfmath will work on float or complex data, but all the input and output
        files must be of the same data type.

        Examples:

        sfmath x=file1.rsf y=file2.rsf power=file3.rsf output='sin((x+2*y)^power)' > out.rsf
        sfmath < file1.rsf tau=file2.rsf output='exp(tau*input)' > out.rsf
        sfmath n1=100 type=complex output="exp(I*x1)"

        See also: sfheadermath.

PARAMETERS
        string  output=         Mathematical description of the output
        string  type=   output data type [float,complex]
        string  unit=
USED IN
        bei/dpmv/matt
        bei/dwnc/sigmoid
```

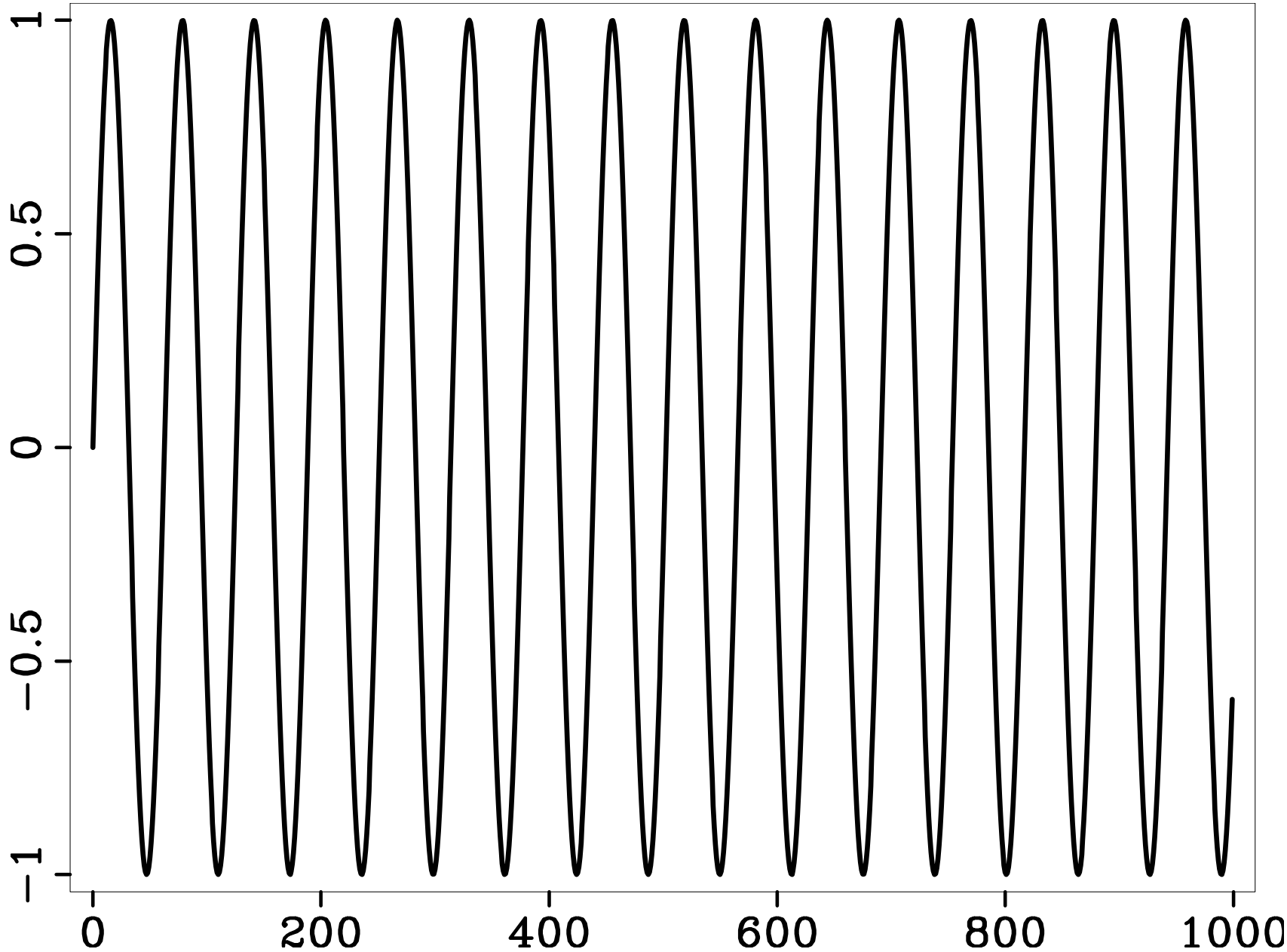
psava@mines.edu

Demo

sfmath n1=1000
output='sin(0.1*x1)'
> sin1.rsf

sfin sin1.rsf

```
sin1.rsf:
    in="/scratch/sin1.rsf@"
    esize=4 type=float form=native
    n1=1000         d1=1            o1=0
        1000 elements 4000 bytes
```

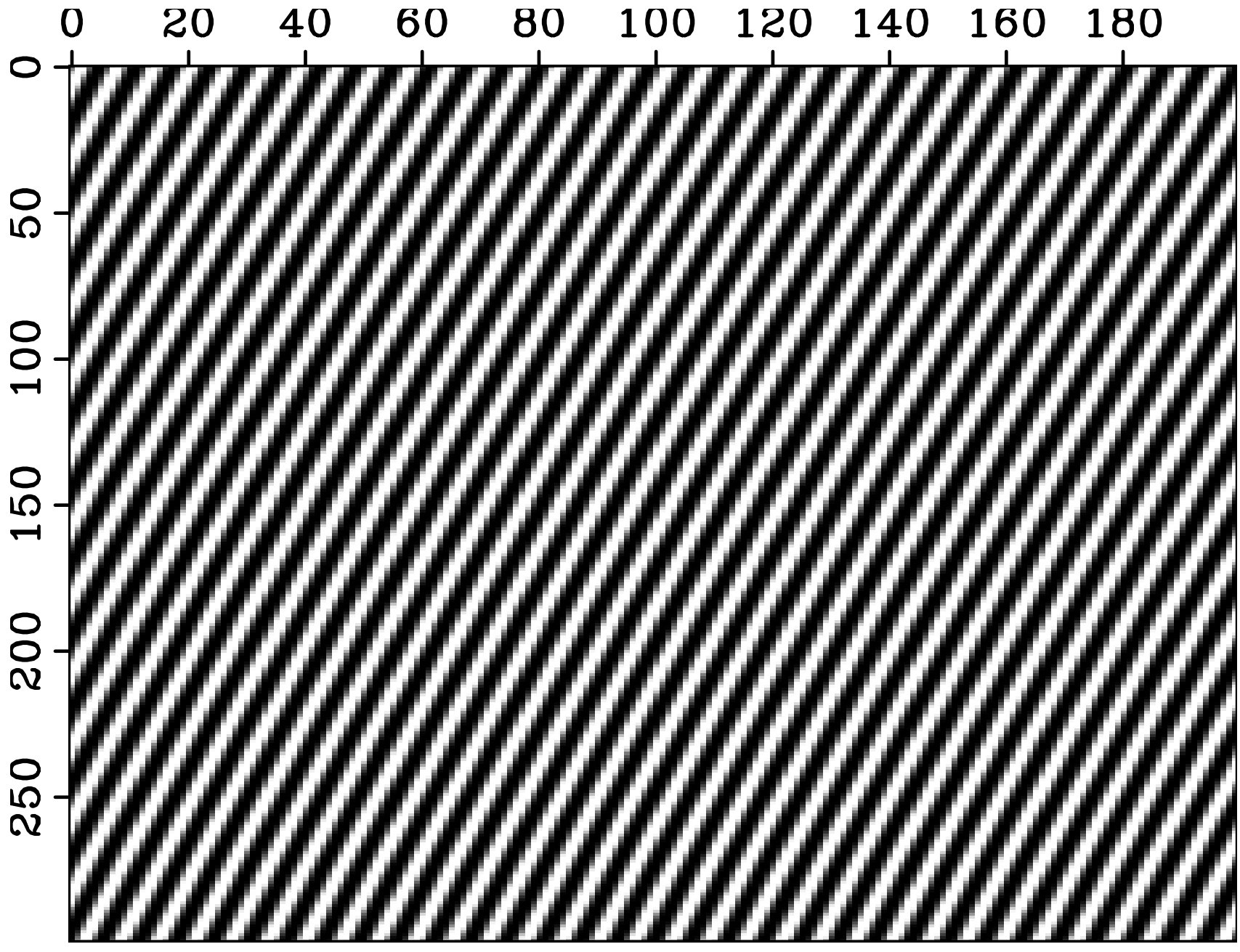

1
0.5
0
−0.5
−1
0
200
400
600
800
1000

Demo

sfmath n1=300 n2=200
output='sin(0.25*x1+1*x2)'
> sin2.rsf

sfin sin2.rsf

```
sin2.rsf:
    in="/scratch/sin2.rsf@"
    esize=4 type=float form=native
    n1=300          d1=1            o1=0
    n2=200          d2=1            o2=0
        60000 elements 240000 bytes
```

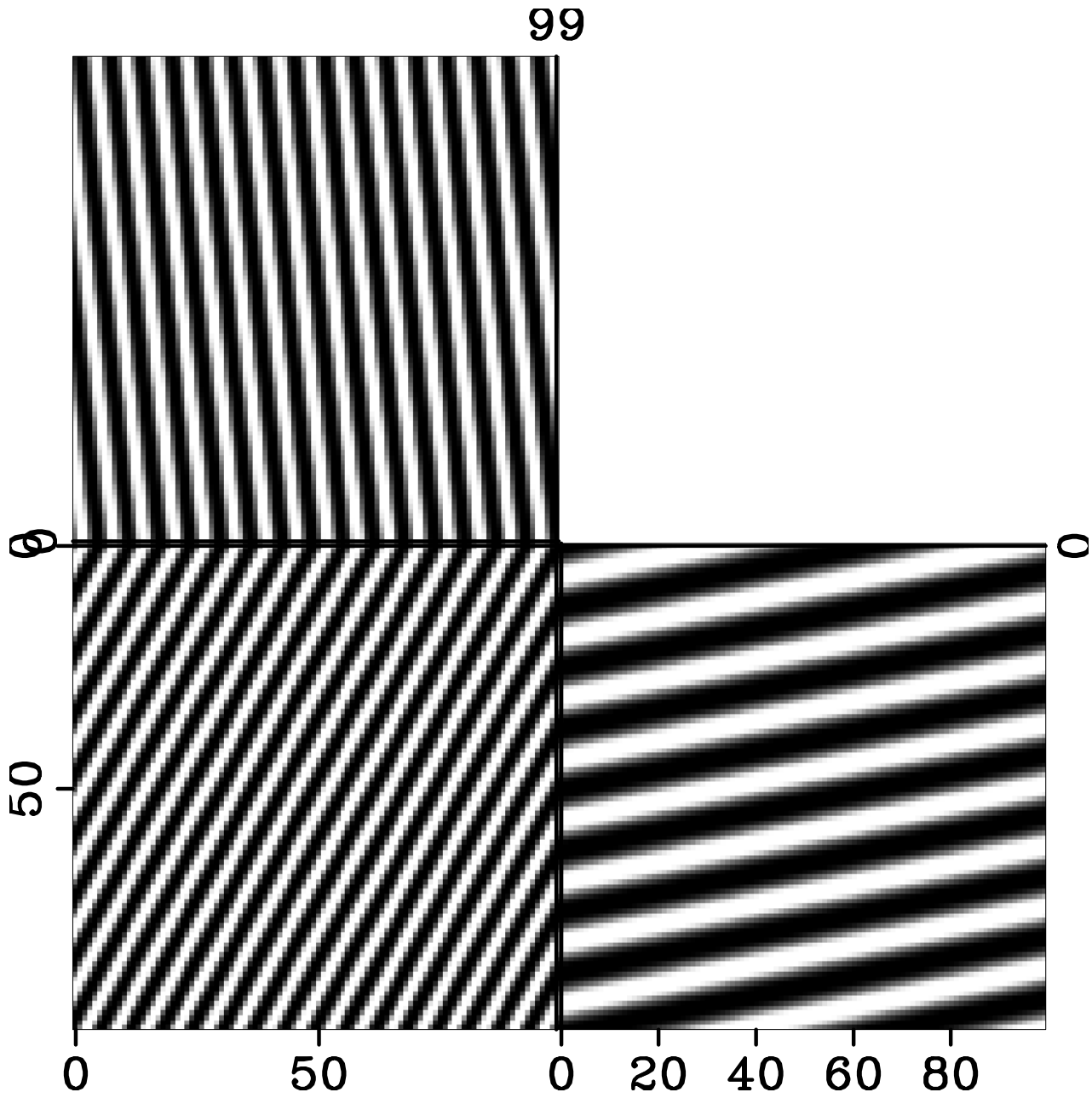

0
20
40
60
80
100
120
140
160
180
0
50
100
150
200
250

Demo

sfmath n1=100 n2=100 n3=100
output='sin(0.5*x1+1.0*x2+0.1*x3)'
> sin3.rsf

sfin sin3.rsf

```
sin2.rsf:
    in="/scratch/sin3.rsf@"
    esize=4 type=float form=native
    n1=100          d1=1            o1=0
    n2=100          d2=1            o2=0
    n3=100          d3=1            o3=0
        1000000 elements 4000000 bytes
```


99
0
0
50
0
50
0
20
40
60
80

Plotting

- **sfgraph**: 1D graphs
- **sfgrey**: 2D/3D grayscale graphs
- **contour**: contour plots
- **sfgrey3**: cube plots
- ...

Demo

sfin sin1.rsf

```
sin1.rsf:
    in="/scratch/sin1.rsf@"
    esize=4 type=float form=native
    n1=1000         d1=1            o1=0
        1000 elements 4000 bytes
```

< sin1.rsf **sfgraph** title="1D plot" | **xtpen**

Demo

sfin sin2.rsf

```
sin2.rsf:
    in="/scratch/sin2.rsf@"
    esize=4 type=float form=native
    n1=300          d1=1            o1=0
    n2=200          d2=1            o2=0
        60000 elements 240000 bytes
```

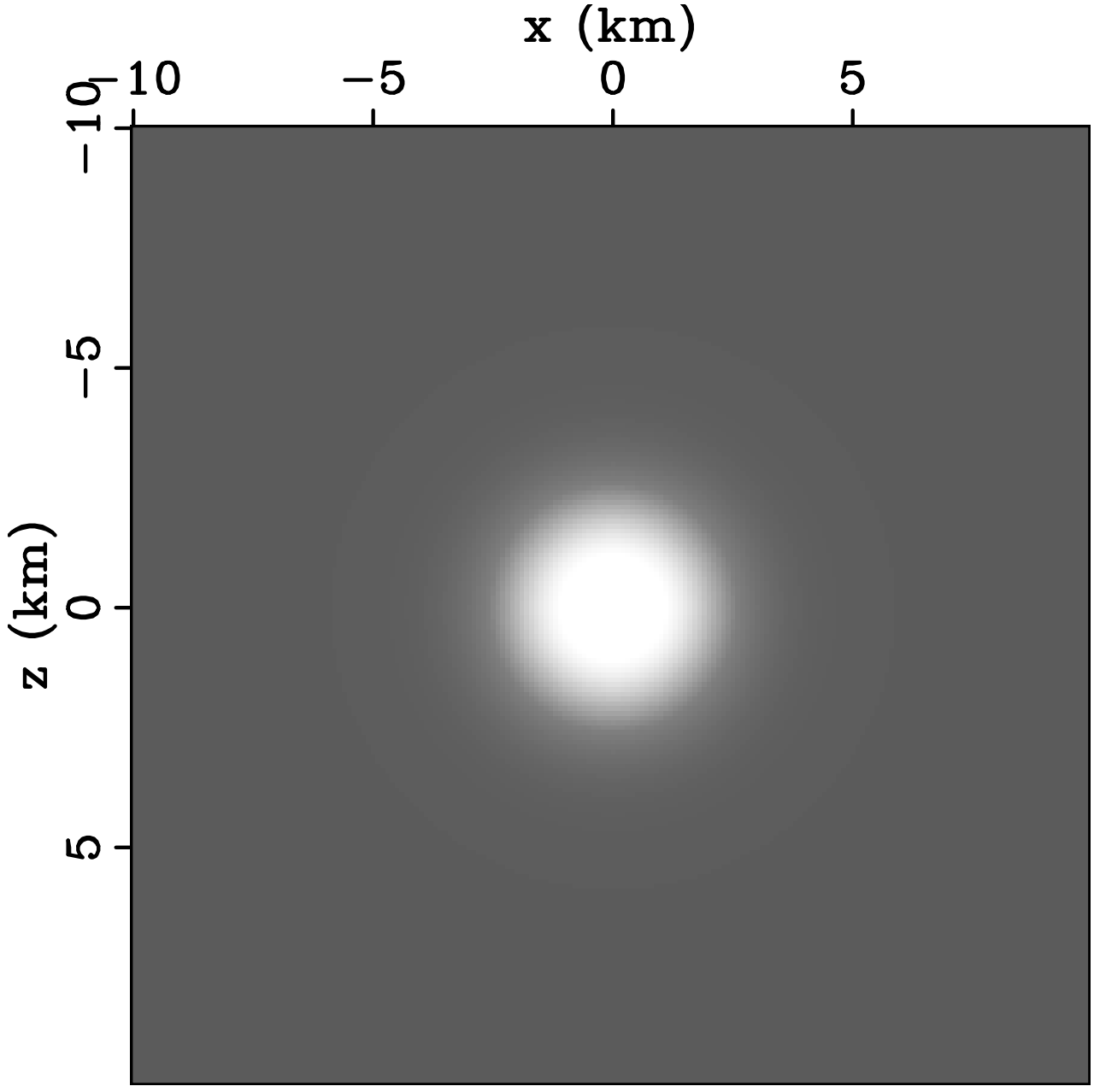
< sin2.rsf **sfgrey** title="2D plot" | **xtpen**

Exercise (1)
create 2D Gaussian function

```
sfmath
   output="exp(-(x1*x1+x2*x2)/(2*1.5*1.5))"
   n1=200 d1=0.1 o1=-10.
   n2=200 d2=0.1 o2=-10. |
sfput
   label1=z unit1=km
   label2=x unit2=km > gg.rsf
```

```
< gg.rsf
   sfgrey pclip=100 screenratio=1 |
   xtpen
```

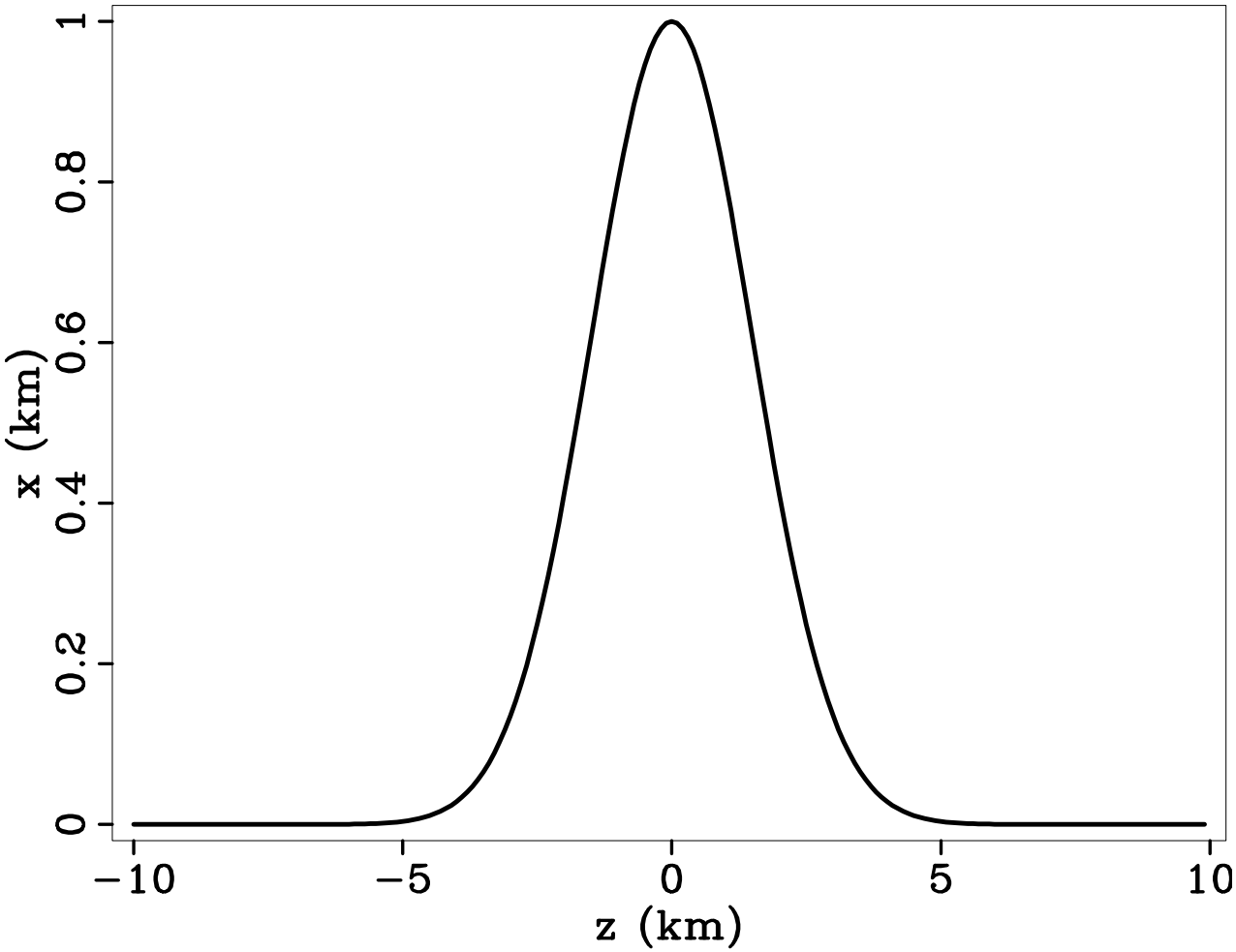
psava@mines.edu


x (km)
−10
−5
0
5
z (km)
−10
−5
0
5

Exercise (2)
extract 1D subset from the 2D Gaussian function

```
< gg.rsf
  sfwindow n2=1 f2=100 |
  sfgraph |
  xtpen
```

psava@mines.edu

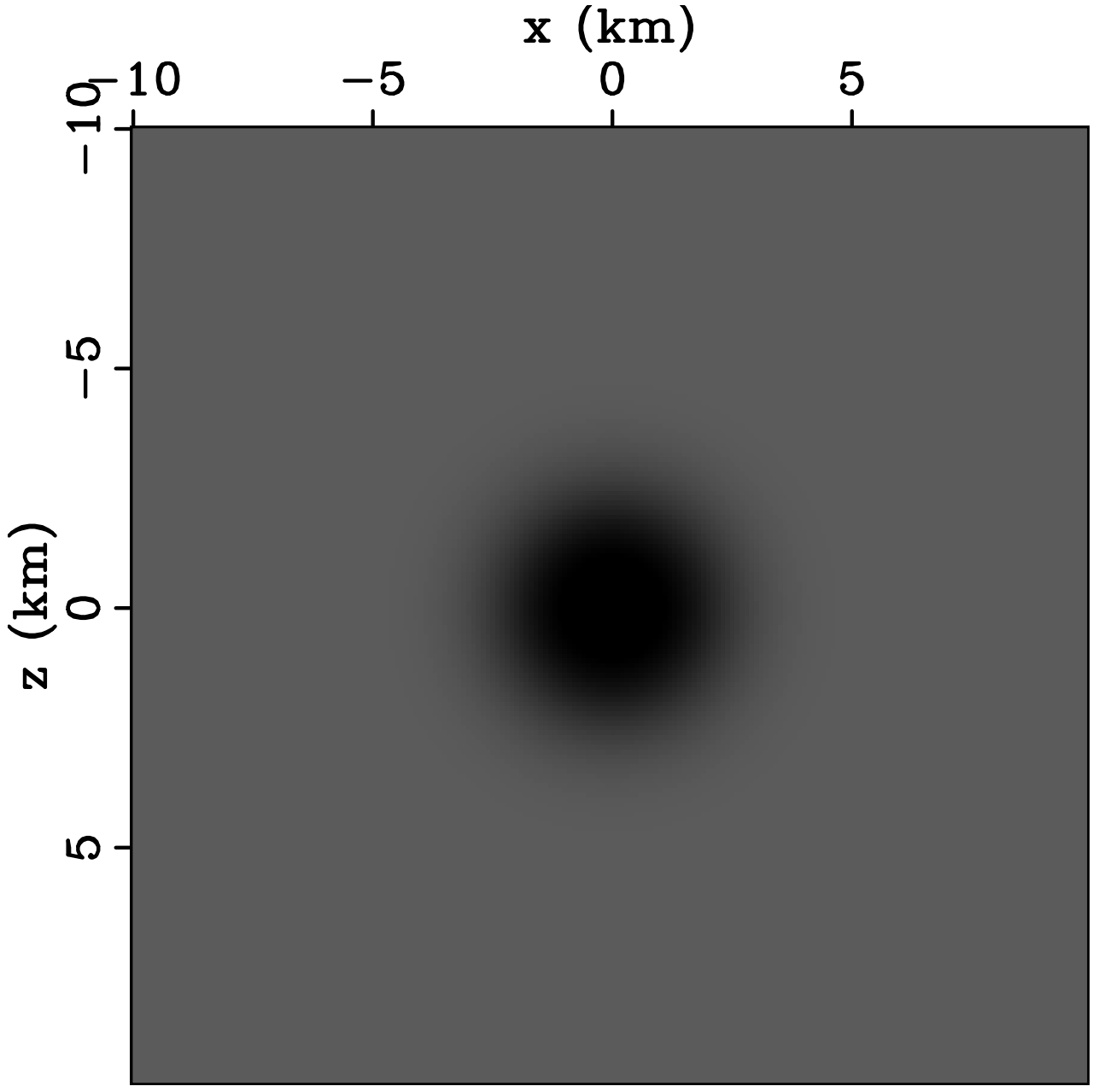

x (km)
0
0.2
0.4
0.6
0.8
1
−10
−5
0
5
10
z (km)

Exercise (3)
create a velocity model

```
< gg.rsf
  sfscale rscale=-1. |
  sfadd add=3 > vel.rsf
```

```
< vel.rsf
  sfgrey title="" pclip=100 screenratio=1 bias=3 |
  xtpen
```

psava@mines.edu

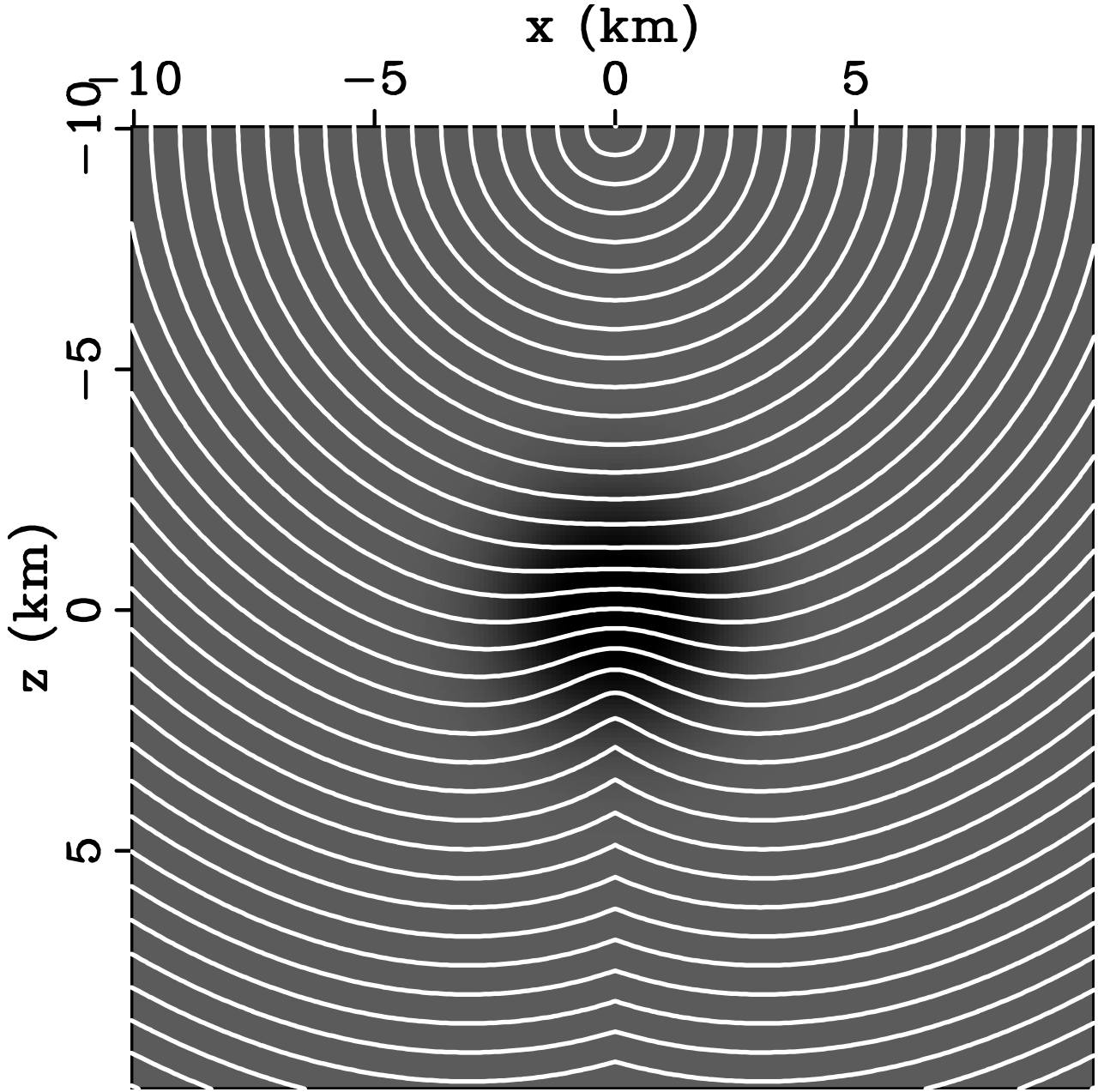

x (km)
−10
−5
0
5
z (km)
−10
−5
0
5

Exercise (4)
compute traveltimes with an eikonal solver

```
< vel.rsf
  sfeikonal zshot=-10 yshot=0
> fme.rsf
```

```
< fme.rsf
  sfcontour title="" nc=200 screenratio=1 |
  xtpen
```

psava@mines.edu



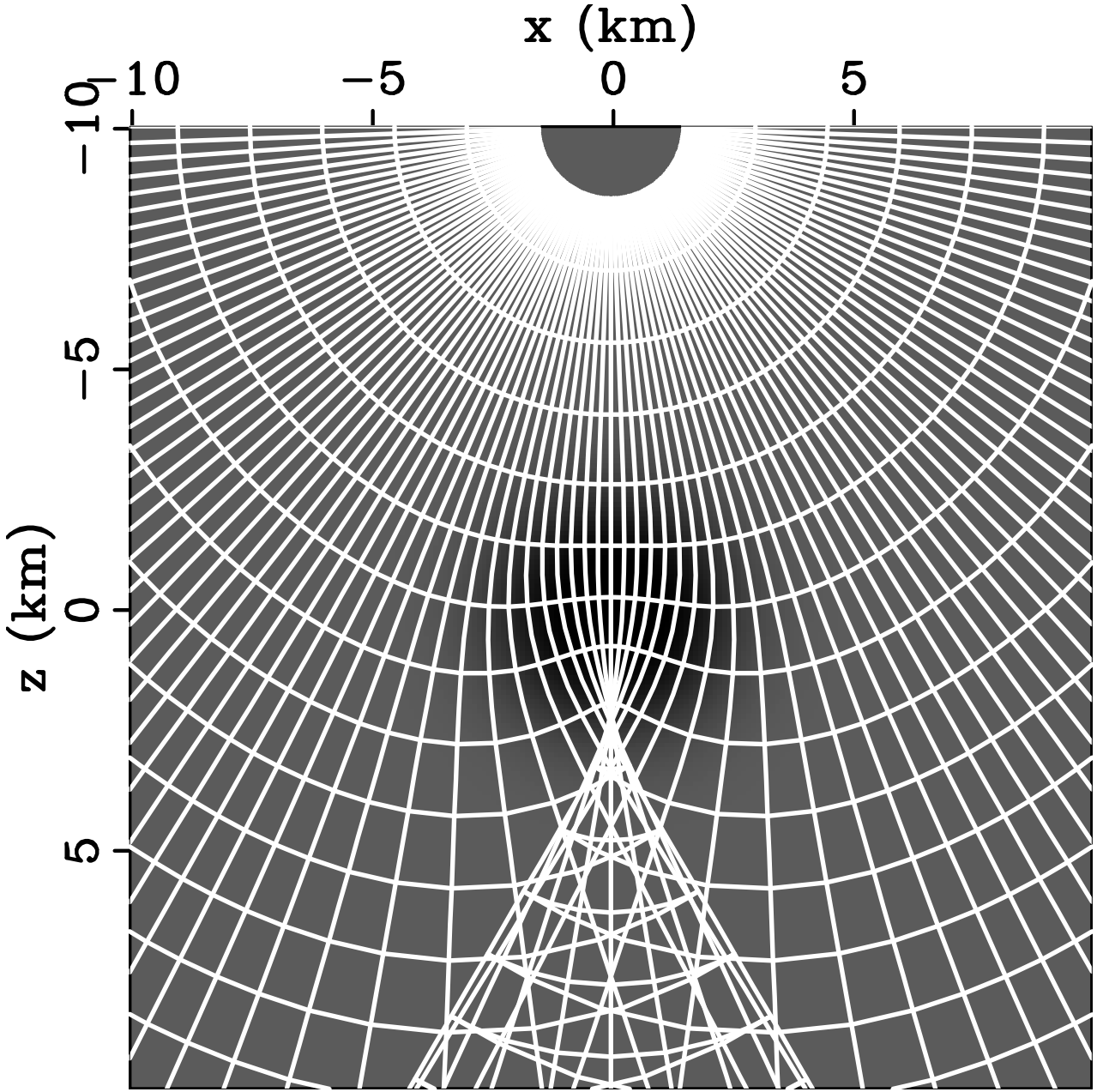
psava@mines.edu

Exercise (5)
compute rays and wavefronts

```
< vel.rsf
  sfhwt2d xsou=0 zsou=-10
  nt=1000 ot=0 dt=0.01
  ng=1801 og=-90 dg=0.1
> hwt.rsf
```

```
< hwt.rsf
  sfwindow j1=20 j2=20 |
  sfgraph title="" yreverse=y screenratio=1
  min1=-10 max1=+10 min2=-10 max2=+10 |
  xtpen
```

psava@mines.edu


x (km)
−10
−5
0
5
z (km)
−10
−5
0
5

Resources

- Introduction to MADAGASCAR
  `http://rsf.sourceforge.net/wiki/index.php/Introduction`
- Guide to MADAGASCAR programs
  `http://rsf.sourceforge.net/wiki/index.php/Programs`
- Guide to MADAGASCAR file format
  `http://rsf.sourceforge.net/wiki/index.php/Format`
- Guide to MADAGASCAR API
  `http://rsf.sourceforge.net/wiki/index.php/API`

psava@mines.edu