Lesson 4-5: 3D data processing in SAC

SAC has a file type for three dimensional data.

Base is a grid of X-Y values at constant increments in X and Y separately. Data values are Z at each grid point.

SAC can do arithmetic on the grid, contour gridded data, and display in gray scale or color.

Basic command that yields 3-D data is `SPECTROGRAM`

Spectrograms

- Frequency-time analysis (also known as FTAN)
- Sliding time window in seismogram yields spectrum over a frequency range (amplitude vs frequency)
- Change in spectrum with time window yields 3-D information (amplitude vs frequency vs time)
- Frequency and time are the base grid (discrete values of each)
- Spectral amplitude is value on grid

Example

- Use `SPECTROGRAM` on built-in seismogram

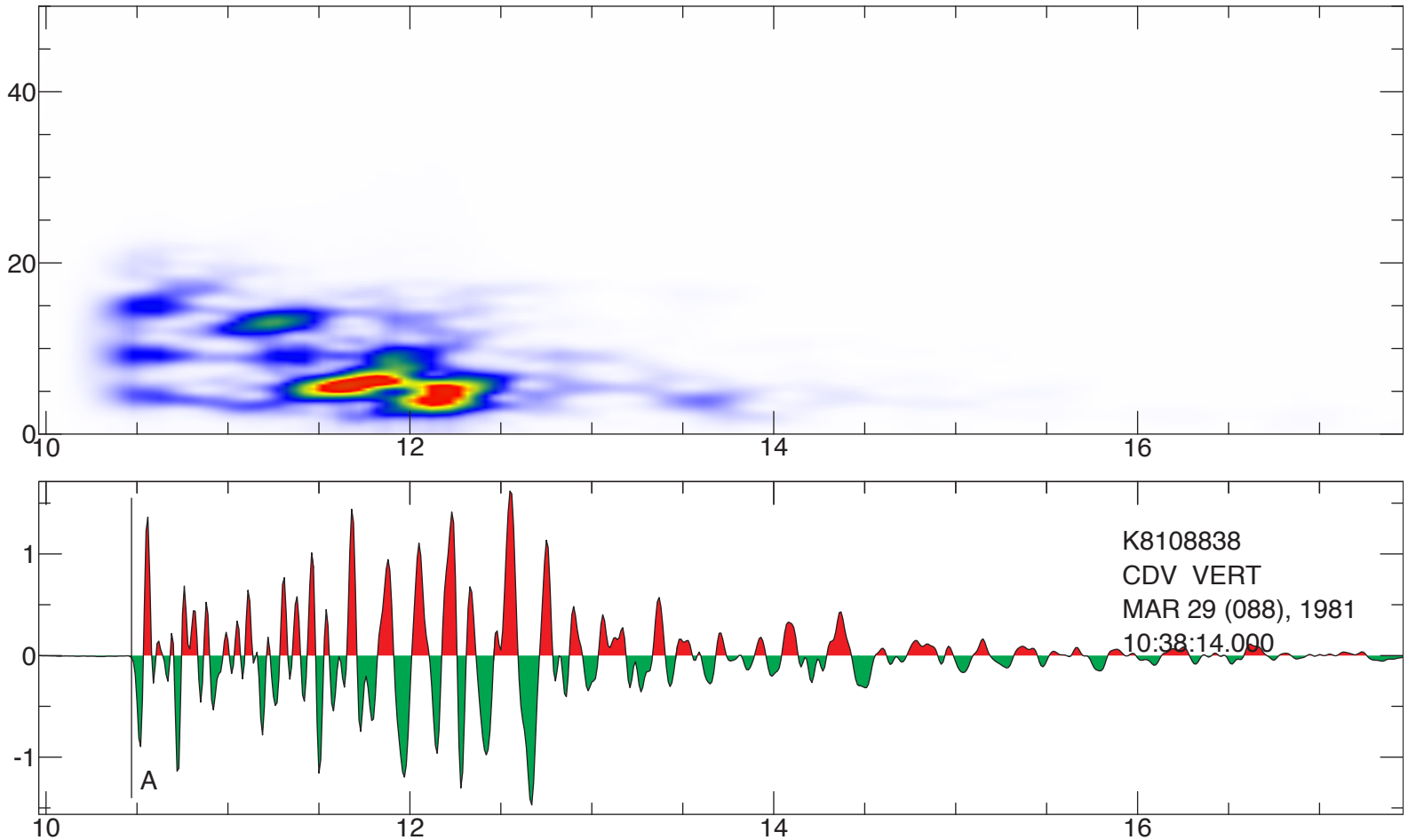
```
SAC> funcgen seismogram; p1   ;* Create and view seismogram

SAC> spectrogram slice 0.05 window 1 ymax 20 cbar on
```

- Display shows time on X axis, frequency on Y axis, spectral amplitude as color (see scale).
- Joint display trickier but more interesting:

```
SAC> funcgen seismogram; rmean
SAC> bf; yvport 0.45 0.9                        ;* upper screen
SAC> spectrogram slice 0.05 window 1 ymax 20
SAC> yvport 0.05 0.4; xlim 9.96 17.46; p1; ef ;* lower screen
SAC> yvport 0.15 0.9                          ;* reset full screen
```

- Spectrogram above, trace below. Highest frequencies at onset, then lower frequencies.

My result ...

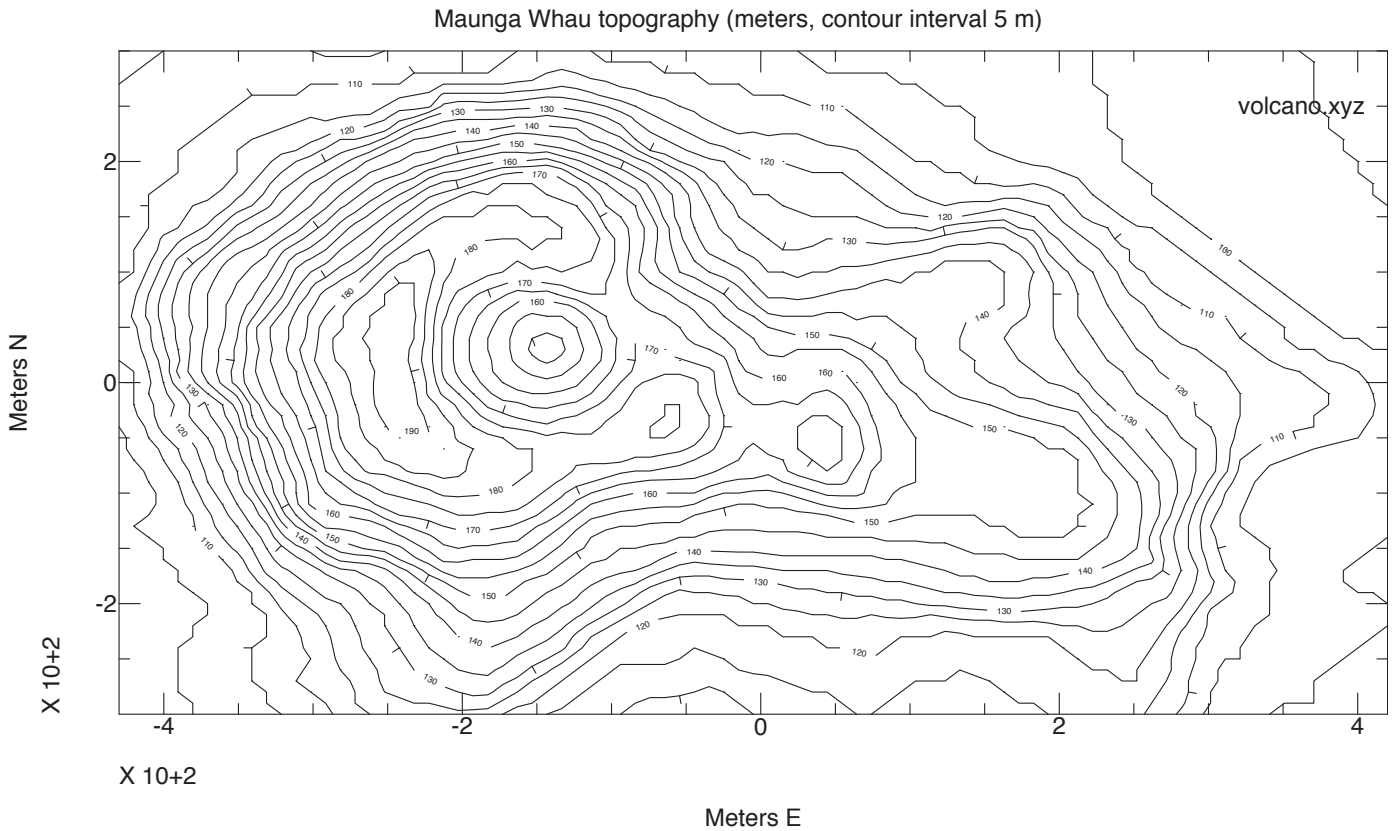


- Top panel is spectrogram, bottom is seismogram
- Peak spectral power at 11.5–12.5 s at 5 Hz

Map data – also is 3D

- Sample contour data in SAC's built-in dataset

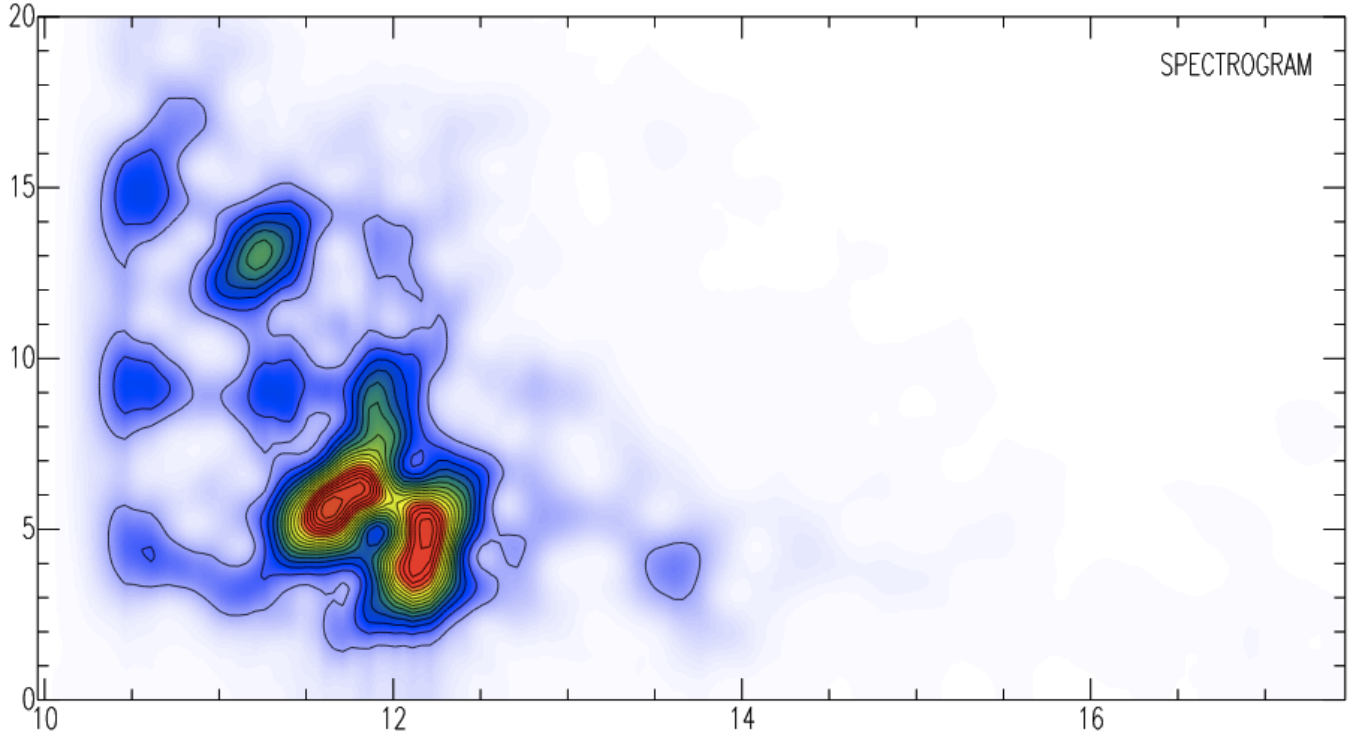
```
SAC> m demo contour complex                ;* Use built-in macro
```



Spectrogram data is 3D – can be contoured

```
SAC> fg seismogram; rmean                ;* Data, de-mean
SAC> spectrogram xyz                     ;* 3D file spectrogram
SAC> bf; ylim 0 20; grayscale color; contour; ef
```

- Overlay color 3D plot and contour it



3D data – closer look

```
SAC> DATAGEN SUB XYZ VOLCANO.XYZ   ;* Retrieve built-in XYZ data
SAC> lh                            ;* List header


  FILE: VOLCANO.XYZ
  -----------------
        NPTS = 5307                     B = 0.0
           E = 5306.0             IFTYPE = GENERAL XYZ (3-D) FILE
       LEVEN = TRUE                DELTA = 1.0
      DEPMIN = 94.0              DEPMAX = 195.0
      DEPMEN = 130.188           LOVROK = TRUE
      NXSIZE = 87               XMINIMUM = -430.0
    XMAXIMUM = 420.0             NYSIZE = 61
    YMINIMUM = -300.0          YMAXIMUM = 300.0
SAC>
```

- File header details
  - `DEPMIN`, `DEPMAX` – range of elevation data
  - `NXSIZE`, `NYSIZE` – # X & Y pixels
  - `XMINIMUM`, `XMAXIMUM` – low, high X values
  - `YMINIMUM`, `YMAXIMUM` – low, high Y values

Simple 3D data file writing

- In your program (example: 3dprog.f)
  - Calculate data on grid
  - Create basic SAC file header
  - Modify header to describe file geometry
  - Write data into file
- Compile with SAC library routines

```
gfortran -m32 -o 3dprog 3dprog.f /usr/local/lib/sacio.a
```

- Run program to write file

```
3dprog
```

The program

```
      program plot3D
C     Grid extends symmetrically from zero
C        for nxgr cells in X and nygr in Y
      parameter (nxgr=50,nygr=50)
      parameter (nx=1+2*nxgr, ny=1+2*nygr)
C     Parameters to control depth of valley (fscl) and X and Y
      parameter (fscl=1, xscl=4, yscl=2)
C     Data for grid.  Grid layout is
C        (*, 1) - lowest streak of y values
C        (*,ny) - highest streak of y values
C        (1, *) - lowest x value at any y position
C        (nx,*) - highest x value at any y position
      real data(nx,ny)

C     Create data:  Curved valley defined by a Rosenbrock fcn
      f(x,y) = (1-x)**2 + fscl*(y-x**2)**2

C     Run over grid and insert elevation information into it.

      do j=1,ny
         y = yscl*float(j-1-nygr)/nygr
         do i=1,nx
            x = xscl*float(i-1-nxgr)/nxgr
            data(i,j) = f(x,y)
         enddo
      enddo
```

The program (cont.)

```
C     Create default header for SAC file, then make it 3-D data
      call newhdr

C     File type to XYZ for 3D data
      call setihv('IFTYPE', 'IXYZ', nerr)

C     Grid dimensions: NXSIZE, NYSIZE
      call setnhv('NXSIZE', nx, nerr)
      call setnhv('NYSIZE', ny, nerr)

C     Grid scale:  XMINIMUM, XMAXIMUM, YMINIMUM, YMAXIMUM
      call setfhv('XMINIMUM', -xscl, nerr)
      call setfhv('XMAXIMUM',  xscl, nerr)
      call setfhv('YMINIMUM', -yscl, nerr)
      call setfhv('YMAXIMUM',  yscl, nerr)

C     Number of data points, begin, delta (req but not relevant)
      call setfhv('B', 0.0, nerr)
      call setfhv('DELTA', 1.0, nerr)
      call setnhv('NPTS', nx*ny, nerr)

C     Write data to file using present header values
      call wsac0('/tmp/data3D.xyz',data,data,nerr)
      end
```

Viewing 3D data

- `READ` command reads in 3D data files
- `GRAYSCALE` command plots 3D data converting elevation into color level (or gray scale, thus the name)
- `CONTOUR` command contours 3D data, with optional control of
  - contour levels (scaled to data, or explicit)
  - labels on contour lines
  - line types (full, dashed, thick, thin)
  - tic marks on uphill/downhill side of contour

```
SAC> read /tmp/data3D.xyz    ;* Read 3D data file

SAC> grayscale color         ;* Color plot

SAC> contour                 ;* Contour plot, default options
```

Integration into SAC processing

- Assignment: Want to vary depth of valley under control of SAC and display results

- Strategy: Build a SAC macro that
  - takes a “valley depth value”
  - supplies the program with that value and runs it
  - reads the output file
  - plots it

Building the SAC macro

```
* SAC macro to make 3D plot of Rosenbrock's function with
* varying valley depth.  Keyword parameter SCALE is valley
* scale factor.
$KEYS scale                        ;* keyword for macro parameter
$DEFAULT scale 1                   ;* supply default value

* Pass SCALE keyword value to program using $RUN / $ENDRUN
*    Will need to modify existing 3dprog to read scale factor
*    from input stream.
$RUN 3dprog-new
$scale$
$ENDRUN

* Read file that 3dprog-new wrote; assume it is /tmp/data3D.xyz

read /tmp/data3D.xyz

* Set up contouring parameters for plot

zlevels list 0 1 10 20 50 100 200 300    ;* contour levels
zlabels on list 0 1 10 20 50 100 200 300 ;* labels for each line

beginframe                         ;* begin plot
   grayscale color                 ;* plot color background
   contour                         ;* overlay contours
endframe                           ;* end plot
zlabels previous; zlevels previous  ;* Restore contour params.
```

Modifying the program

- Hack previous program to read in scale value

```
      program new3D
C     Grid extends symmetrically from zero
C        for nxgr cells in X and nygr in Y
      parameter (nxgr=50,nygr=50)
      parameter (nx=1+2*nxgr, ny=1+2*nygr)
C     Parameters to control depth of valley (fscl) and X and Y
      parameter (fscl=1, xscl=4, yscl=2)
C     Data for grid.  Grid layout is
C        (*, 1) - lowest streak of y values
C        (*,ny) - highest streak of y values
C        (1, *) - lowest x value at any y position
C        (nx,*) - highest x value at any y position
      real data(nx,ny)

C     Create data:  Curved valley defined by a Rosenbrock fcn
      f(x,y) = (1-x)**2 + fscl*(y-x**2)**2

      read(*,*) fscl
      ...
```

Compile and install the program

```
gfortran -m32 -o 3dprog-new 3dprog-new.f /usr/local/lib/sacio.a
```

- Make sure 3dprog-new is in your search path so SAC macro can find it!
- Typical places for your own program executables is ~/bin
- PATH variable should have ~/bin in it

```
install 3dprog-new ~/bin          # Install binary in path
```

Test the program+macro

```
#vi 3dmacro                                # Create macro
#sac ~/seismo/sacinit
 SEISMIC ANALYSIS CODE [August 11, 1992 (Version 10.6f-grh106)]
 Copyright 1992 Regents of the University of California

SAC> echo on                              ;* Use ECHO ON for testing
SAC> m 3dmacro scale 1
 m 3dmacro scale 1
 $RUN 3dprog-new
 $scale$
 ==> 1
 $ENDRUN

 * Read file that 3dprog-new wrote; assume it is /tmp/data3d.xyz

 read /tmp/data3d.xyz

 * Set up contouring parameters for plot

 zlevels list 0 1 10 20 50 100 200 300    ;* contour levels
 zlabels on list 0 1 10 20 50 100 200 300 ;* labels for each line

 beginframe                               ;* begin plot
    grayscale color                       ;* plot color background
    contour                               ;* overlay contours
 endframe                                 ;* end plot
 zlabels previous; zlevels previous  ;* Restore contour params.
SAC>
```

Testing macro+program

- Works!
- Turn ECHO OFF
- Test with other values of SCALE (up to 100) -- what happens?
  - Fix?

- Improvements:
  - Read 3D data file name from input -- possibly will be a clash if more than one user is running on same computer! Use random temporary file name specific to macro, then destroy it after use.
  - Display subset of grid?