sacpsd

1. Introduction

The determination of power spectral density (PSD) of Earth noise is a useful tool for monitoring noise sources and also the performance of seismic instruments (McNamara and Buland, 2004). The purpose for writing this program is to be able to use routine PSD measurements as a tool for verifying the metadata for digital instrumentation and for checking on instrument performance.

Section 2 discusses the determination of a power spectral density from windowed data. After a quick introduction to the command line options of **sacpsd**, Section 3 focuses on real data from three sources. An Appendix follows that highlights the CALPLOT graphics package of Computer Programs in Seismology.

2. Power Spectra

The computation of power spectra follows Ifeachor and Jervis (1993). Given a time series $x(k)$, we first correct that series to have a zero mean, apply a window, $w(k)$, and correct amplitudes for the windowing function. Their equation (10.16) defines the corrected time series, $s(k)$, as

$$s(k) = c_2\, w(k)\, [\, x(k) - c_1\,]$$

where the constants are defined as

$$c_1 \;=\; \Sigma\, w(k)\, x(k)\, /\; \Sigma\, w(k) \quad\text{, and}$$

$$c_2{}^2 \;\;=\; N\, /\, \Sigma\, w^2(k)\text{, and}$$

N is the number of equally spaced observations, with sampling interval Δt, and samples are for $k=0,...,N-1$. In **sacpsd** the user may select either a Hanning window or a 10% sine taper for the windowing function.

Using an N-point discrete Fourier transform (DFT), the transform of $s(k)$ is $S(n)$, which is defined as

$$S(n) = \Sigma\, s(k)\, \exp(\, \text{-}\, j\, 2\pi\, ft)\; \Delta t \;=\; \Sigma\, s(k)\, \exp(\, \text{-}\, j\, 2\pi\, nk/N)\; \Delta t$$

where $n=0,...,N-1$, the summation runs from $k=0,...,N-1$, $t = k\, \Delta t$ and $f = n\, \Delta f$. $\Delta f = 1/N\, \Delta t$. Because of the definitions of the forward and inverse DFT's, both $s(k)$ and $S(n)$ are periodic.

The convention for the powerspectrum is to define it as

$$PSD(f = n\, \Delta f) \;=\; 2\, S^2(n)\, /\, T$$

where the time window $T = N\; \Delta t$. The factor 2 accounts for the negative frequency contribution.

For actual processing, the initial time series is assumed to consist of more than $N=16384$ points. Then using successive N point segments that overlap by $N/2$ points, the individual PSD's are summed and

finally averaged by dividing by the number of segments, NSEG. The stacking and averaging reduces the variance in the PSD estimate, yielding a smoother PSD.

The estimate of the ground noise uses the frequency dependent modulus of acceleration sensitivity of the sensor, $G(counts/m/s^2)$ to give the noise estimate in decibels through the relation

$$N(f) = 10 \log_{10} [\ (1/NSEG) \ \Sigma \ PSD_i / G^2 \ ]$$

For comparison with other determinations, a smoothed noise spectrum is computed from

$$(1/NSEG) \ \Sigma \ PSD_i / G^2 \ ]$$

by applying a simple smoother, currently a 5-point averaging operation in the equi-spaced frequency domain. Other smoothers may be implemented in the future. We note that McNamara and Buland (2004) apply a smoothing operation over period. It will be seen in Figures 3 and 4 that there is a significant difference between the two techniques at certain frequencies. Finally the smoothed noise spectrum is linearly interpolated to yield estimates at periods that are multiples of $2^{1/8}$ and then converted to decibel. To make the output independent of sample rate, the multiples are always with respect to a period of 1.0 seconds. This smoothed version is always plotted as an overlay onto the N(f).
;

3. Using sacpsd

sacpsd is invoked directly from the command line. The command syntax and options can always be seen by invoking the command as

```
rbh> sacpsd -h
Usage: sacpsd -f sacfile -r response -H -S -L listfile
 -f sacfile   (default none) sac binary trace
 -r response  (default none) file giving table of
          frequency-response pairs created by invoking
          evalresp with the -u "acc" flag to give the
          amplitude response in units of COUNTS/M/S**2  vs f(Hz)
 -H          (default false ) use Hanning window
 -S          (default true ) use 10% sin windo
 -L listfile plots according to list which has entries
          psd_file1 kolor1 title1
          psd_file2 kolor2 title2
        where file1 is listing of period-psd(db) pairs
        kolor is a CALPLOT color, e.g., 1000 to 1100 for red ->blue
        title is a string without spaces
 -A alistfile plots according to list which has entries
          psd_file1
          psd_file2
```

```
        The color is automatically generated
 -NT    (default  false ) do not annotate with file name
w
 -5             (default true ) apply 5 point PSD smoothing
 -h             (default false) online help
```

The output of the program is a CALPLOT graphics file, SACPSD.PLT, and file named using the template NNSSSSSCCCLL.YYYY.DDD.HH.MM.psd, where NN is the network code, SSSSS the station code, CCC the channel code, LL the location, YYYY the year, DDD the day of year, and HH and MM are the hour and minute of the reference time in the header of the Sac formatted traces. Because the display graphics is built into the program, different invocations yield differing results, the previous graphics file, SACPSD.PTL is overwritten.

3.1 Simple examples of using sacpsd

If **sacpsd** is invoked with no arguments, then the resulting plot shows the high- and low-noise models (Peterson, 1993). Note that the `rbh>` is the computer prompt and that the command follows the prompt.

```
rbh> sacpsd
```

creates the plot file SACPSD.PLT, which can be converted to an EPS file using the command

```
rbh> plotnps -F7 -W10 -EPS  -K < SACPSD.PLT > j.eps
rbh> gm convert -trim j.eps Fig01.png
```

gm is the GraphicsMagick program (http://www.GraphicsMagick.org/) that convert and/or displays different file formats. The resulting figure is shown in Figure 1.
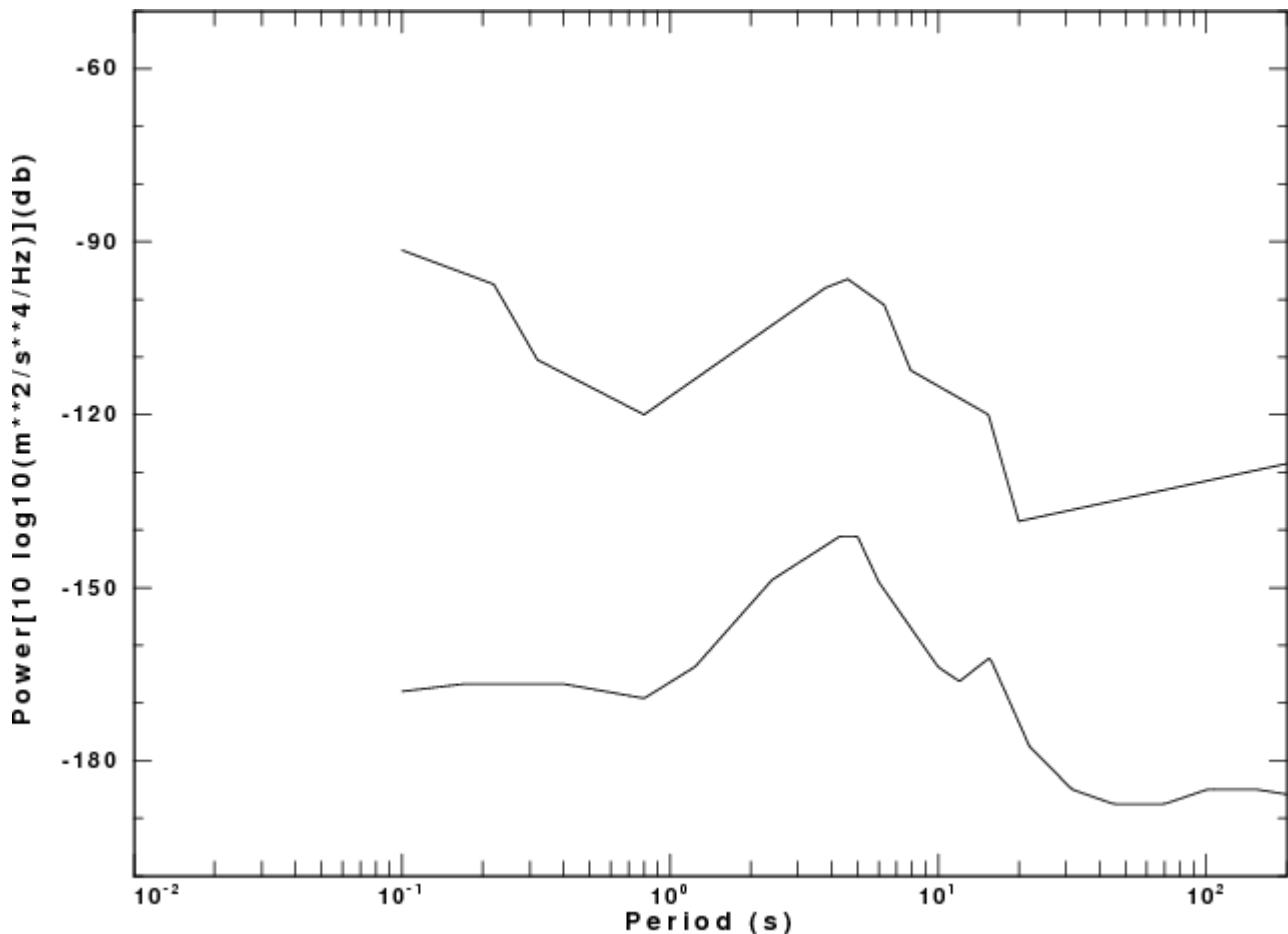
Fig. 1. Default output of **sacpsd** which is just a plot of the low and high noise models

The second example uses a one-hour long BHZ component of the station SLM and the accelereation sensitivity file for that channel:

```
rbh> sacpsd -f NMSLM__BHZ__.SAC -r AMP.NM.SLM..BHZ
rbh> plotnps -F7 -W10 -EPS -L < SACPSD.PLT > j.eps
rbh> gm convert -trim j.eps Fig02.png
```

which creates the SACPSD.PLT file which is converted to a PNG format, and the file

      NMSLM__BHZ__.2009.305.10.59.psd

which is a period – PSD file for the NM network, SLM station, BHZ component for day 305 of 2009 with data set starting at 10:59 UT. Figure 2 shows the graphics output.
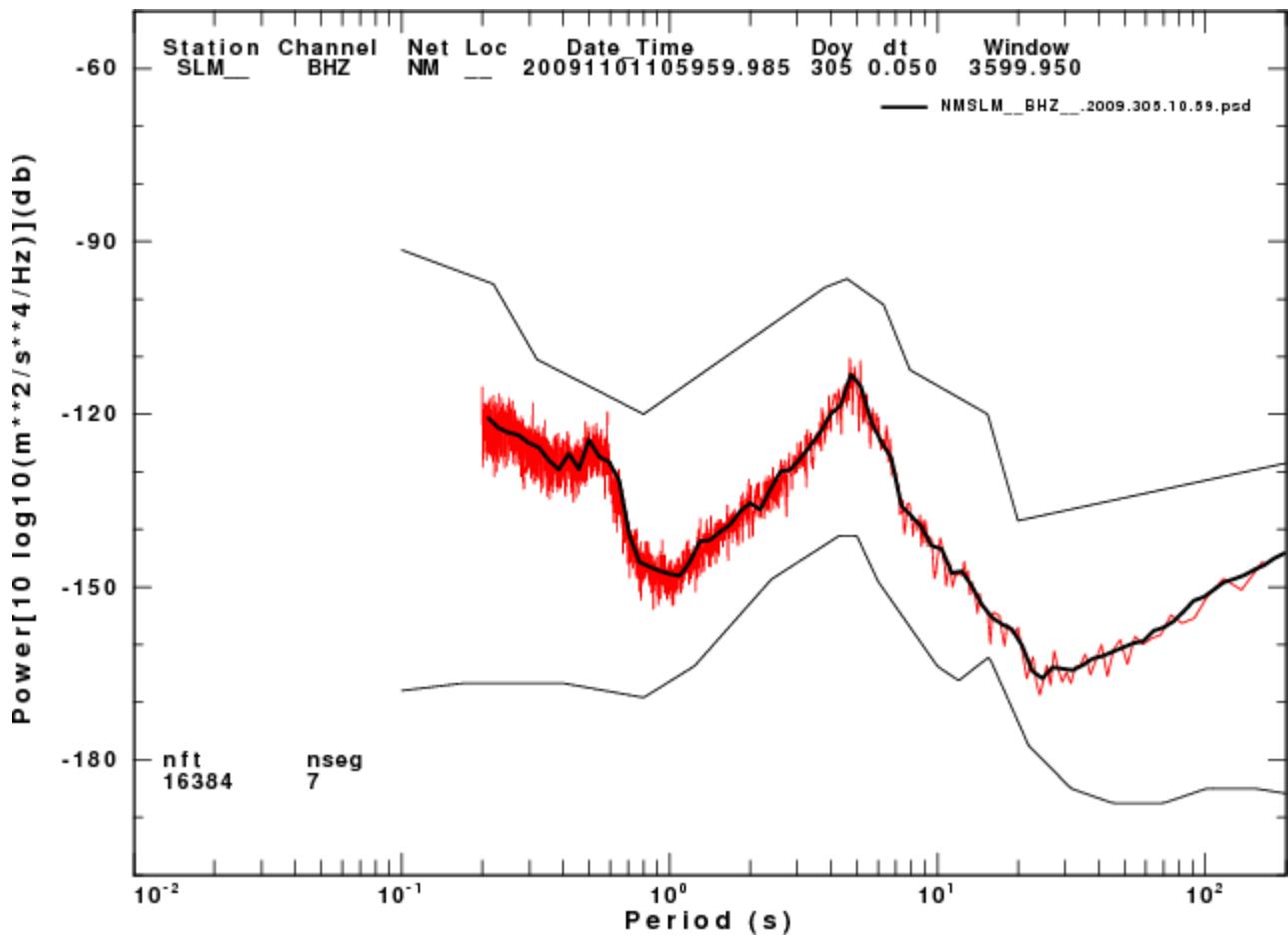
Fig. 2. The computed PSD is plotted in red. The 5-point smoothed PSD is plotted as the thicker black line. The figure is annotated with the reference time contained in the trace header.

The graphics file is annotated with sufficient information to define the trace. **nft** indicates the length of the Fast Fourier transform used, and **nseg** is the number of individual PSD's that were averaged to yield the red curves above. The larger that **nseg** is, the smoother the red curve, and the more meaningful the average.

The next variation overlays another PSD estimate which is in the file *mxy*, with one period – PSD entry per line. The control file *listfile.03* consists of just one line:

```
mxy 1080 McNamara
```

listfile.03 has three entries, the name of the file, an integer representing a CALPLOT color, and a title word. Figure 3 is created using the following command lines:

```
rbh> sacpsd –f NMSLM__BHZ__.SAC –r AMP.NM.SLM..BHZ –L listfile.03
```

Computer Programs in Seismology - sacpsd

```
rbh> plotnps -F7 -W10 -EPS -L < SACPSD.PLT > j.eps
rbh> gm convert -trim j.eps Fig03.png
```

The result of this run is the creation of the PSD file

NMSLM__BHZ__.2009.305.10.59.psd

and the plot shown in Figure 3.
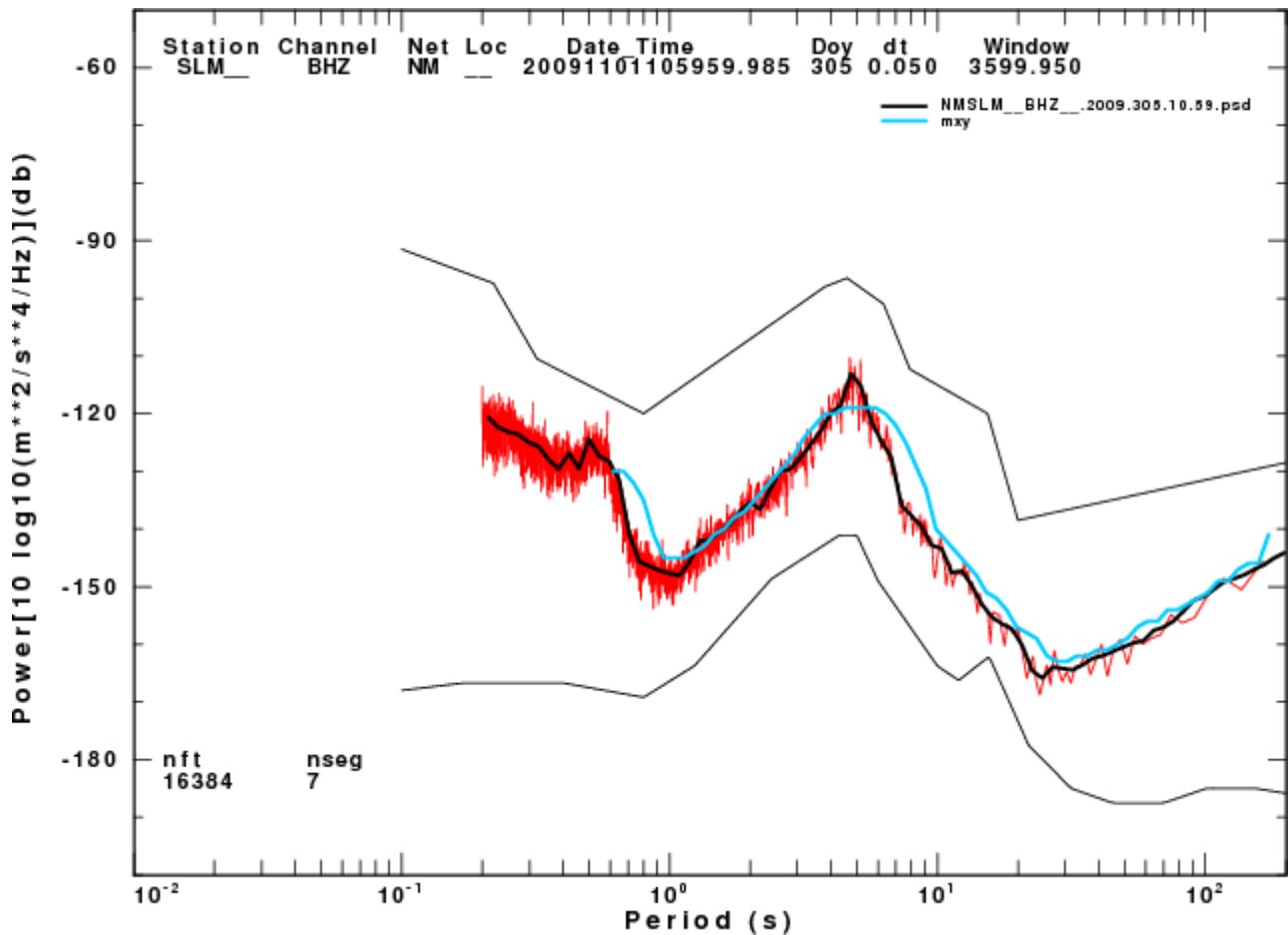
Fig. 3. Overlay of PSD contained in the file *mxy* on top of the computed PSD. The orange curve is from the **sacpsd** computation, while the blue-green curve is the result of running McNamara's program **psd.**

The final example is to use the file list in the file *listfile.04* which contains the two entries:

```
mxy 1080 McNamara
NMSLM__BHZ__.2009.305.10.59.psd 1020 SLM
```

with the command sequence

```
rbh> sacpsd  -L listfile.04
rbh> plotnps -F7 -W10 -EPS -K < SACPSD.PLT > j.eps
rbh> gm convert -trim j.eps Fig04.png
```

Figure 4 shows the result of using this command.
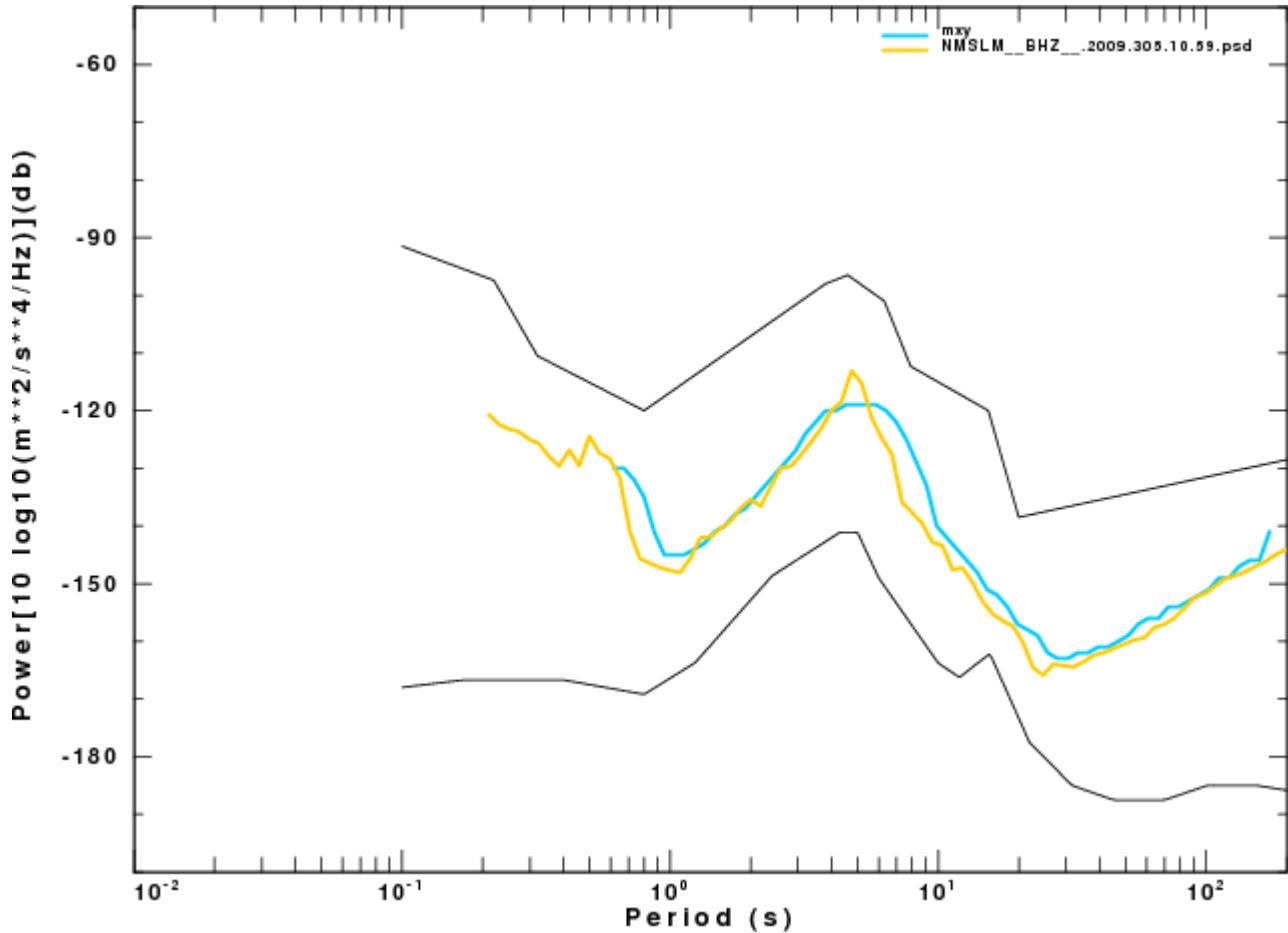
Fig. 4. Example of comparing the contents of PSD files. The orange curve is from the **sacpsd** computation, while the blue-green curve is the result of running the McNamara and Buland (2004) code.

In Figures 3 and 4. we see a major difference between the results of applying the McNamara and Buland (2004) code. To understand the source of this problem, we downloaded an early version from the USGS anonymous ftp site, e.g.,

ftp://ftpext.usgs.gov/pub/cr/co/golden/hazards/web/mcnamara/PDF/pdfsa_src.tar.bz2

 unpacked it, and compiled to create the program **pdf**. Using the same trace and instrument response, debugging statements showed that all output was identical up to the computation of the smoothed

spectrum. **sacpsd** applies a liner average over frequency, while **pdf** averages samples in the period domain between periods T and $2^{1/8}$ T, associating the average with a period $2^{1/16}$ T, This choice of smoothing introduced a bias in regions where the N(f) changes rapidly.

3.2 Examples using CWBQuery

CWBQuery and **mdget** are programs developed by NEIC to access the Continuous Wave Buffer and station metadata at NEIC. In the script shown here, we use CWBQuery to obtain one-hour of data from the BHZ components for the NM network for February 7, 2010 starting at 11:00:00 UT.

The processing script is called ***DOIT*** and is listed here.

```
#!/bin/sh

#####
#       define the time windows for the trace and response
#####
DATETIME="2010/02/07-11:00:00"
MDBEG="2010/02/07-11:00:00"
MDEND="2010/02/07-12:00:00"
WINDOW=3600


#####
#       Clean up previous data files
#####
rm -f *SAC*


#####
#       define a short name for CWBQuery
#####
alias query='java -jar ~/bin/CWBQuery.jar'


#####
#     Use query to get all data for NM network filling gaps with
#     zeros,  returning as a Sac file.
#     Use mdget to get the RESP file for use with the program
#     evalresp.
#     Note that a line terminated with a backslash newline is
#     a continuation for the shell and is use for clarity
#####
query -dccdbg -t sac -o "%N.SAC" -sacpz nm -fill 0 \
     -b "${DATETIME}" -d ${WINDOW} -s "NM.....BHZ.."
mdget -resp -b "${MDBEG}" -e "${MDEND}" -s "NM.....BHZ.."
#####
#    This will yield the following files
#      NMBLO__BHZ___.SAC  NMUSIN_BHZ___.SAC   RESP.NM.PLAL..BHZ
#      NMMGMO_BHZ___.SAC  NMUTMT_BHZ01.SAC    RESP.NM.PVMO..BHZ
#      NMOLIL_BHZ___.SAC  RESP.NM.BLO..BHZ    RESP.NM.SIUC..BHZ
#      NMPBMO_BHZ___.SAC  RESP.NM.FVM..BHZ    RESP.NM.SLM..BHZ
#      NMPVMO_BHZ___.SAC  RESP.NM.MGMO..BHZ   RESP.NM.UALR..BHZ
#      NMSIUC_BHZ___.SAC  RESP.NM.MPH..BHZ    RESP.NM.USIN..BHZ
#      NMSLM__BHZ___.SAC  RESP.NM.OLIL..BHZ   RESP.NM.UTMT.01.BHZ
#      NMUALR_BHZ___.SAC  RESP.NM.PBMO..BHZ   RESP.NM.UTMT..BHZ
```

```
#####
#####
#       perform the PSD on each file terminated with .SAC in the current
#       current directory.
#####

for TRACE in *.SAC
do

#####
#       Extract information from the file using the CPS program saclhdr.
#       We use this program to make header values available to the shell
#       script. We require this to select the proper RESP file.
#####

KSTNM=`saclhdr -KSTNM $TRACE`
KCMPNM=`saclhdr -KCMPNM $TRACE`
DELTA=`saclhdr -DELTA $TRACE`
NPTS=`saclhdr -NPTS $TRACE`
FHH=`echo $DELTA | awk '{print 0.50/$1}' `
FHL=`echo $DELTA | awk '{print 0.25/$1}' `
YEAR=`saclhdr -NZYEAR $TRACE`
DOY=`saclhdr -NZJDAY $TRACE`
KNETWK=`saclhdr -KNETWK $TRACE`
KHOLE=`saclhdr -KHOLE $TRACE`

#####
#       define the lower frequency bound which related to the
#       longest period resolved, e.g.,  1/4 length of record.
#       This will be used by the program evalresp.
#####
FLH=`echo $NPTS $DELTA | awk '{print 1.0/($1*$2)}' `
#####
#       Change defaults of the network and location codes in the
#       trace headers so that we can associate the trace with the
#       proper response file. In a properly created SAC trace file
#       an undefined parameter is always identified as -12345. for
#       a floating point number, -12345 for an integer and
#       -12345 for a string. Here we set an undefined equal to a
#       a zero length string
#
#       KNETWK contains the two character FDSN network code
#       KHOLE  contains the two character FDSN location code
#       We first test for a blank and then for the -12345
#####
        if [ -z "${KNETWK}" ]
        then
                NET=""
        else
                if [ "${KNETWK}" = "-12345" ]
                then
                        NET=""
                else
                        NET="${KNETWK}"
                fi
        fi
        if [ -z "${KHOLE}" ]
        then
                LOC=""
```

```
        else
                if [ "${KHOLE}" = "-12345" ]
                then
                        LOC=""
                else
                        LOC="${KHOLE}"
                fi
        fi
#####
#       we know the response file now
#####
RESPFILE=RESP.${NET}.${KSTNM}.${LOC}.${KCMPNM}
#####
#       create the acceleration response in terms of M/S**2
#       using evalresp for this date.
#####
        evalresp ${KSTNM} ${KCMPNM} ${YEAR} ${DOY} ${FLH} ${FHH} 2049 \
            -u 'acc' -f ${RESPFILE}
        sacpsd -f  ${TRACE} -r AMP.${NET}.${KSTNM}.${LOC}.${KCMPNM}
done
#####
#    As a result of these computations the directory now contains
#    the PSD files
#       NMBLO__BHZ__.2010.038.10.59.psd  NMSIUC_BHZ__.2010.038.10.59.psd
#       NMMGMO_BHZ__.2010.038.10.59.psd  NMSLM__BHZ__.2010.038.10.59.psd
#       NMOLIL_BHZ__.2010.038.10.59.psd  NMUALR_BHZ__.2010.038.10.59.psd
#       NMPBMO_BHZ__.2010.038.10.59.psd  NMUSIN_BHZ__.2010.038.10.59.psd
#       NMPVMO_BHZ__.2010.038.10.59.psd  NMUTMT_BHZ01.2010.038.10.59.psd
#####


#####
#    Finally overlay all of the computed PSD files for an automatic overlay
#    to create Figure 5
#####
ls *psd > afile
sacpsd -A afile
plotnps -K -EPS -F7 -W10 < SACPSD.PLT > t.eps
EPSTOPNG t.eps
mv t.png Fig05.png
rm *.eps
rm SACPSD.PLT
```

Figure 5 shows the result of running the script. The final overlay plot shows the high and low noise model limits, and the individual station PSDs. The color coding and long file name provide a means for identifying a curve.

The positive thing about this plot is all systems, except PBMO, show the same noise levels between 6 and 20 seconds. We can examine the levels at a period of 12.8 seconds by using the following **grep** command on the PSD files:

```
rbh> g rep 12.337 *.psd | sort -n -k3
NMUALR_BHZ__.2010.038.10.59.psd:    12.337687    -145.427
NMPVMO_BHZ__.2010.038.10.59.psd:    12.337687    -144.736
```

```
NMOLIL_BHZ__.2010.038.10.59.psd:    12.337687    -144.315
NMUSIN_BHZ__.2010.038.10.59.psd:    12.337687    -144.178
NMSLM__BHZ__.2010.038.10.59.psd:    12.337687    -144.129
NMSIUC_BHZ__.2010.038.10.59.psd:    12.337687    -144.123
NMMGMO_BHZ__.2010.038.10.59.psd:    12.337687    -143.638
NMBLO__BHZ__.2010.038.10.59.psd:    12.337687    -143.049
NMUTMT_BHZ01.2010.038.10.59.psd:    12.337687    -141.430
NMPBMO_BHZ__.2010.038.10.59.psd:    12.337687    -136.830
```

Clearly PBMO behaves differently having a gain about 6db (a factor of 2) higher than the others.

At long periods, the PBMO response differs greatly, indicating a sensor problem.

The short-period noise at PVMO is very high, as is expected since this is a deep sediment site, just a short distance (200 meters) from a railroad. A visual examination the hour-long segment shows that there is a significant noise burst, presumably from a train.

At a period of 0.1 sec, the curve for PBMO is very high. We know that this station has a sample rate of 40Hz, but this behavior is indicative of using the response for a 20Hz sampling interval, which is indicated in the RESP file, e.g.,

```
rbh> grep date RESP*PBMO*
B052F22     Start date:  2007,173,00:00:00.0000
B052F23     End date:    No Ending Time
```

which was not expected since the station was upgraded in November, 2009. We then verified that the correct metadata had not yet been loaded into the NEIC metadata server.

Figure 6 shows the result of changing the script to work with the N component at these stations. At a 12.8 second period we see, using **grep,** amplitudes varying between -140.2 db at OLIL to -144.9 db at SIUC. Not surprisingly, the metadata problem at short periods at PBMO and the short-period noise at PVMO are still there. The variability of the long period levels may be due to sensor emplacement.

Figure 7 shows similar plots for the E component. The PVMO E component sensor has problems at long periods
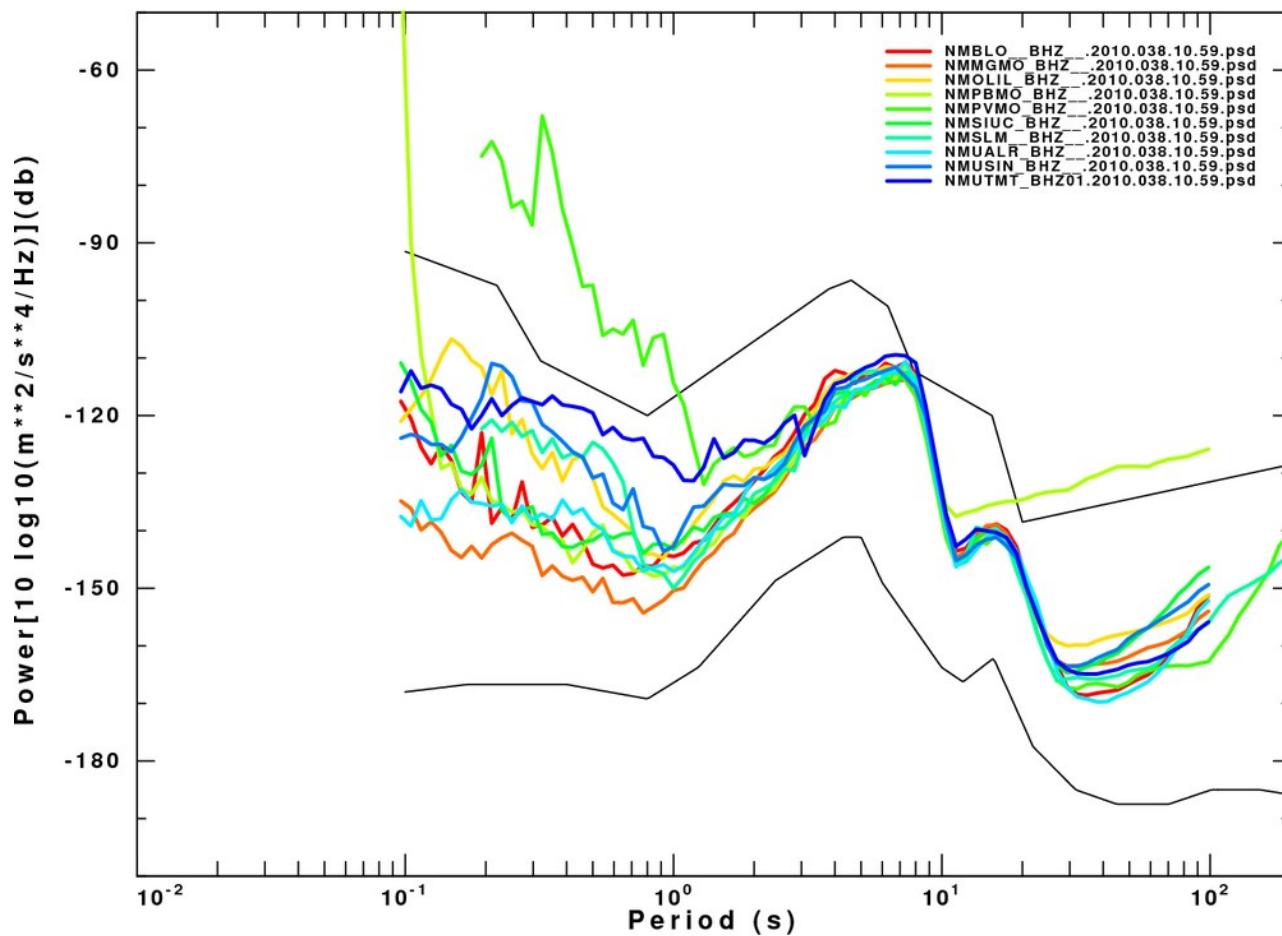
Fig. 5. Result of running the script for the Z component. The reason that different components cover different period ranges is due to the different sample rates. PVMO has a rate of 20 Hz while the others have a rate of 40 Hz.

NOTE: Use overlay plots to focus on outliers, which may reflect a) incorrect responses, b) improper instrument performance, or c) major noise differences.
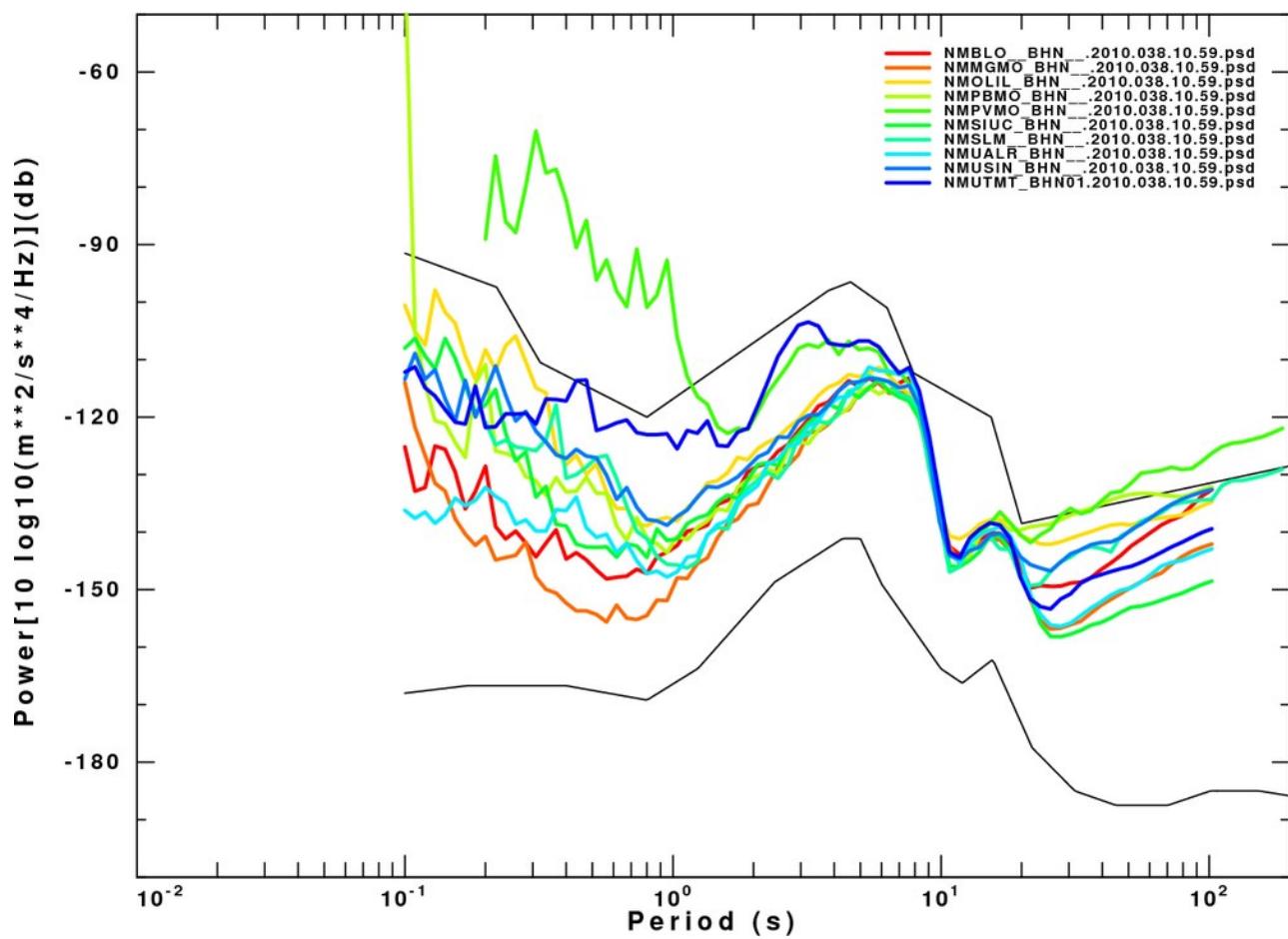
Fig. 6. Running the script used to create Figure 5, but for the N components.
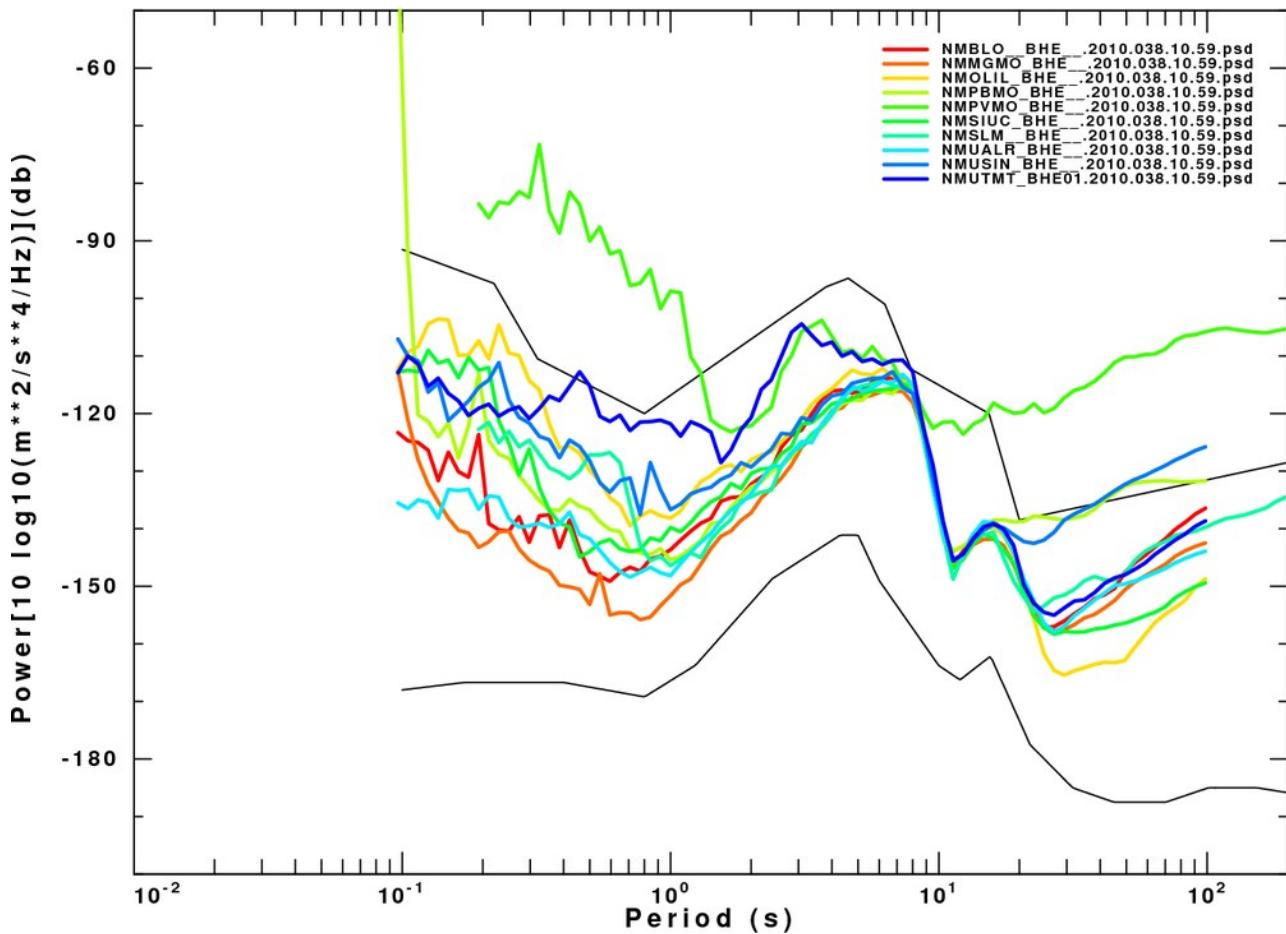
Fig. 7. Running the script used to create Figure 5, but for the E components.

3.3 Examples using rdseed

This example focuses on the use of data from the IRIS DMC . I will use the *breq_fast* request method to get data.

First go to

http://www.iris.edu/dms/sq.htm

and then click on the   **breq_fast**   tab to take you to the BREQ_FAST Request Query Form at

http://www.iris.edu/dms/sq.htm

Here I ask for the network NM, station SIUC with a start time of 2010 01 02 11 00 00.0 and an end time of 2010 01 02 12 00 00.0. I then hit the

Start Query**

which will provide an option to display the search results. Select View BREQ_FAST File, which then provides a prototype Email message for requesting the data:

```
.NAME YOUR_NAME
.INST YOUR_INSTITUTION
.MAIL YOUR_ADDRESS
.EMAIL YOUR_EMAIL
.PHONE YOUR_PHONE
.FAX YOUR_FAX
.MEDIA: YOUR_MEDIA
.ALTERNATE MEDIA: YOUR_ALT1MEDIA
.ALTERNATE MEDIA: YOUR_ALT2MEDIA
.LABEL YOUR_LABEL
.END
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 BHE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 BHN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 BHZ
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HHE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HHN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HHZ
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HLE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HLN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HLZ
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 LHE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 LHN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 LHZ
```

At this point modify the entries identifying the user, and perhaps modify the waveform request. For this example, I send the following text to **breq_fast@iris.washington.edu** :

```
.NAME RBHerrmann
.INST Dept Earth and Atmos Sciences Saint Louis University
.MAIL 3642 Lindell Blvd, St Louis MO 63108
.EMAIL rbh@eas.slu.edu
.LABEL 20100102SIUC
.END
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 BHE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 BHN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 BHZ
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HHE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HHN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HHZ
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HLE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HLN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 HLZ
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 LHE
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 LHN
SIUC NM 2010 01 02 11 00 00.0 2010 01 02 12 00 00.0 1 LHZ
```

I then received an Email from IRIS stating that they received the request. When the request is completed, an other Email is sent stating that the file is placed in the ftp directory

ftp://ftp.iris.washington.edu/pub/userdata/RBHerrmann

 If I place this URL into my browser, I will see a page with entries of the form:

Index of ftp://ftp.iris.washington.edu/pub/userdata/RBHerrmann/

Up to higher level directory

Name	Size	Last Modified
20050110.12736	5108 KB	02/11/2010 09:38:00 AM
20100102MGMO.3924	2560 KB	02/17/2010 09:35:00 AM
20100102SIUC.831	5024 KB	02/17/2010 09:34:00 AM

 tells me that the desired file location is given by the URL

> ftp://ftp.iris.washington.edu/pub/userdata/RBHerrmann/20100102SIUC.831

which I can access using either from the browser, or from the command line, by

```
wget  ftp://ftp.iris.washington.edu/pub/userdata/RBHerrmann/20100102SIUC.831
```

which permits this to be placed withing a shell script. The command **wget** is available under LINUX, CYGWIN and MacOS-X.

The resultant transfer will place the file `20100102SIUC.831` in the current directory. The files and response data can then be extracted using **rdseed**:

```
rdseed −f 20100102SIUC.081 −R −p −d −o 1
```

which creates the SAC trace files, the RESP file and the pole-zero file for use with **sac2000** or **gsac**. The contents of the current directory after this operation is

```
2010.002.10.20.24.0695.NM.SIUC..LHZ.D.SAC  rdseed.err_log.02.18.10  SAC_PZs_NM_SIUC_BHE__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.24.19.0695.NM.SIUC..LHE.D.SAC  RESP.NM.SIUC..BHE        SAC_PZs_NM_SIUC_BHN__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.47.50.0695.NM.SIUC..LHN.D.SAC  RESP.NM.SIUC..BHN        SAC_PZs_NM_SIUC_BHZ__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.59.16.6195.NM.SIUC..BHZ.D.SAC  RESP.NM.SIUC..BHZ        SAC_PZs_NM_SIUC_HHE__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.59.19.4945.NM.SIUC..BHE.D.SAC  RESP.NM.SIUC..HHE        SAC_PZs_NM_SIUC_HHN__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.59.42.7445.NM.SIUC..BHN.D.SAC  RESP.NM.SIUC..HHN        SAC_PZs_NM_SIUC_HHZ__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.59.47.8483.NM.SIUC..HHZ.D.SAC  RESP.NM.SIUC..HHZ        SAC_PZs_NM_SIUC_HLE__2004.174.21.05.00.0000_2010.014.24.60.60.99999
2010.002.10.59.50.0583.NM.SIUC..HLZ.D.SAC  RESP.NM.SIUC..HLE        SAC_PZs_NM_SIUC_HLN__2004.174.21.05.00.0000_2010.014.24.60.60.99999
2010.002.10.59.50.5983.NM.SIUC..HHN.D.SAC  RESP.NM.SIUC..HLN        SAC_PZs_NM_SIUC_HLZ__2004.174.21.05.00.0000_2010.014.24.60.60.99999
2010.002.10.59.58.0083.NM.SIUC..HHE.D.SAC  RESP.NM.SIUC..HLZ        SAC_PZs_NM_SIUC_LHE__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.59.58.4483.NM.SIUC..HLN.D.SAC  RESP.NM.SIUC..LHE        SAC_PZs_NM_SIUC_LHN__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
2010.002.10.59.59.1283.NM.SIUC..HLE.D.SAC  RESP.NM.SIUC..LHN        SAC_PZs_NM_SIUC_LHZ__2004.174.14.54.00.0000_99999.9999.24.60.60.99999
20100102SIUC.831                           RESP.NM.SIUC..LHZ
```

You will note that the SAC file does not begin exactly at 11:00:00 as requested because the data blocks created by the data acquisition system do not start at this time. If there are no data gaps, the trace data will begin before 11:00:00 and terminated after 12:00:00. The pole-zero files, e.g., SAC_PZs_, are annotated with the time windows for the response, which can change change with time as sensors and data acquisition systems are replaced.

The complete script for retrieving and unpacking waveforms and for performing the noise analysis is

now listed.

```sh
#!/bin/sh

DOWNLOAD="ftp://ftp.iris.washington.edu/pub/userdata/RBHerrmann/20100102SIUC.831"
wget ${DOWNLOAD}

#####
#       do not change after here
#####

#####
#       clean up
#####
rm -f *.SAC
#####
#       use basename to get the actual seed file on the local machine
#       which will be 20100102SIUC.831 in this example
#####
SEEDFILE=`basename ${DOWNLOAD} `

rdseed -f ${SEEDFILE} -R -d -o 1

#####
#       perform the PSD
#####

for TRACE in *.SAC
do

#####
#       get information from the file
#####

KSTNM=`saclhdr -KSTNM $TRACE`
KCMPNM=`saclhdr -KCMPNM $TRACE`
DELTA=`saclhdr -DELTA $TRACE`
NPTS=`saclhdr -NPTS $TRACE`
FHH=`echo $DELTA | awk '{print 0.50/$1}' `
FHL=`echo $DELTA | awk '{print 0.25/$1}' `
YEAR=`saclhdr -NZYEAR $TRACE`
DOY=`saclhdr -NZJDAY $TRACE`
KNETWK=`saclhdr -KNETWK $TRACE`
KHOLE=`saclhdr -KHOLE $TRACE`

#####
#       define the lower frequency bound which is 1/4 length of record
#####
FLH=`echo $NPTS $DELTA | awk '{print 1.0/($1*$2)}' `
#####
#       we will not rewrite the KNETWK and KHOLE in the headers with new values
#       but we need to match patterns in the RESP file
#####
        if [ -z "${KNETWK}" ]
        then
                NET=""
        else
                if [ "${KNETWK}" = "-12345" ]
                then
```

```
                                NET=""
                else
                                NET="${KNETWK}"
                fi
        fi
        if [ -z "${KHOLE}" ]
        then
                LOC=""
        else
                if [ "${KHOLE}" = "-12345" ]
                then
                                LOC=""
                else
                                LOC="${KHOLE}"
                fi
        fi


RESPFILE=RESP.${NET}.${KSTNM}.${LOC}.${KCMPNM}



#####
#       create the acceleration response in terms of M/S**2
#####
evalresp ${KSTNM} ${KCMPNM} ${YEAR} ${DOY} ${FLH} ${FHH} 2049 -u 'acc' -f ${RESPFILE}


sacpsd -f  ${TRACE} -r AMP.${NET}.${KSTNM}.${LOC}.${KCMPNM} -5

#####
#       rename the plot file for later use
#####
mv SACPSD.PLT ${KNETWK}${KSTNM}.${KCMPNM}.${YEAR}.${DOY}.PLT

done

#####
#       now make the composite plot of individual components
#####
for COMP in Z N E
do
        case ${COMP} in
                Z) FN=08;;
                N) FN=09;;
                E) FN=10;;
        esac
        ls *${COMP}_*.psd > afile

        sacpsd -A afile
        plotnps -K -EPS -F7 -W10 < SACPSD.PLT > t.eps
        EPSTOPNG t.eps
        mv t.png Fig${FN}.png
        rm *.eps
        rm SACPSD.PLT afile
done

#####
#       for a special study we want the original plots for the
#       Z component
#####
```

```
for i in `ls *Z.${YEAR}.${DOY}.PLT`
do
        B=`basename $i .PLT`
        plotnps -F7 -W10 -EPS -K < $i > t.eps
        gm convert -trim t.eps ${B}.png
        rm t.eps
done


#####
#       clean up
#####
        rm -f sacpsd.out RESP* SAC_PZs*
        rm -f AMP* PHASE*
```

This script creates six PNG files: Fig08.png Fig09.png Fig10.png NMSIUC.BHZ.2010.002.png NMSIUC.HHZ.2010.002.png NMSIUC.HLZ.2010.002.png NMSIUC.LHZ.2010.002.png .

The script is essentially the same as the CWBQuery script of Section 3.2, differing in the data source, and in the tailored graphics at the end.

The Figure 8-10 compare the smoothed noise PSD for all data streams for an individual component of motion.
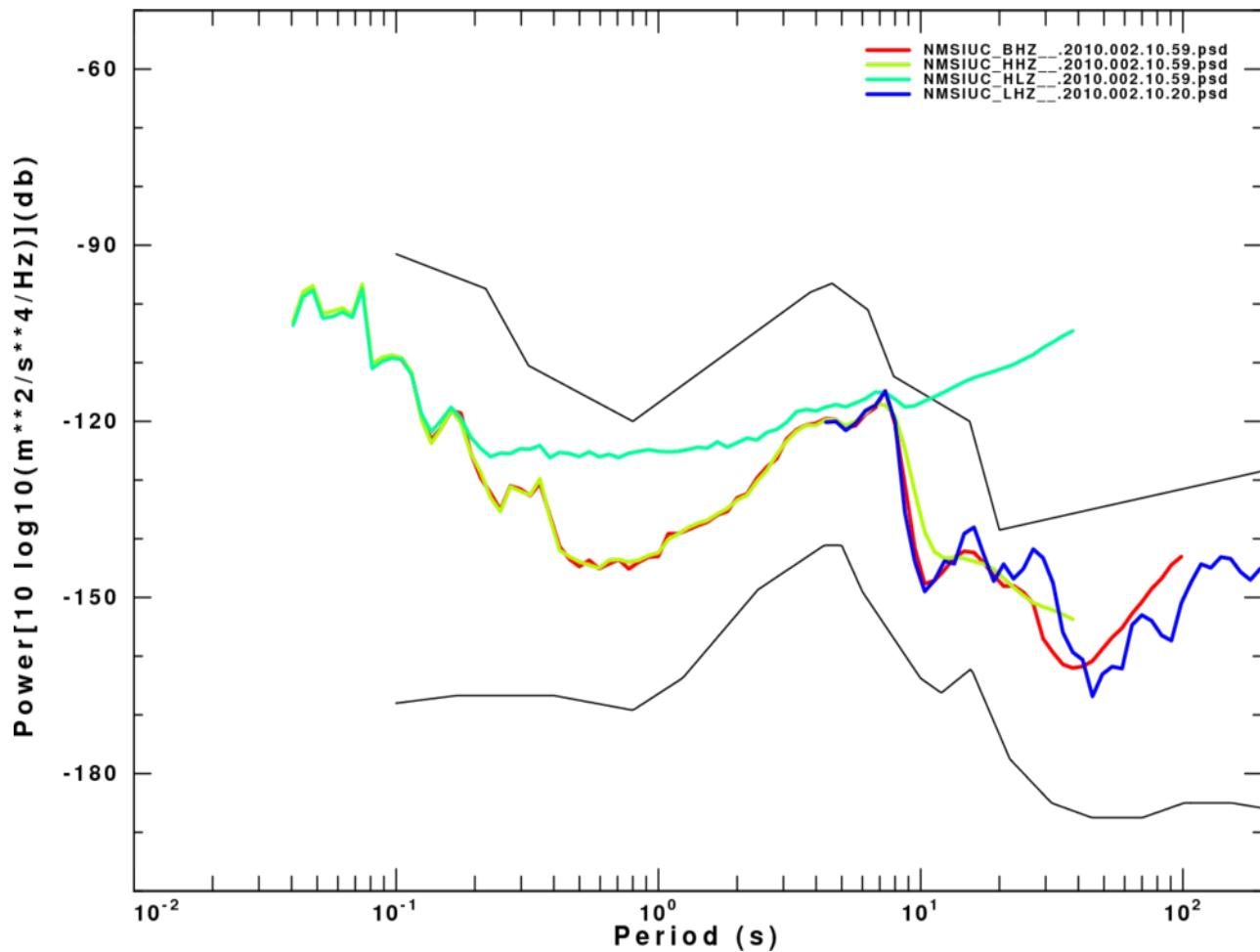
Fig. 8. Comparison of noise PSD form the Z channel data streams at SIUC.

NOTE: Accelerometer noise estimate at long periods has no value since there is no resolution.

NOTE: Differences between LH and BH channels at long periods may be due to the small number of segments used for averaging with LH channels.

NOTE: The shorter HH cutoff at long periods is due to the 16384 point long FFT used.

NOTE: For this time window and station, DO NOT try to compare deconvolved ground motions between the HL (accelerometer) and HH (broadband) streams except at periods less than 0.2 sec

---
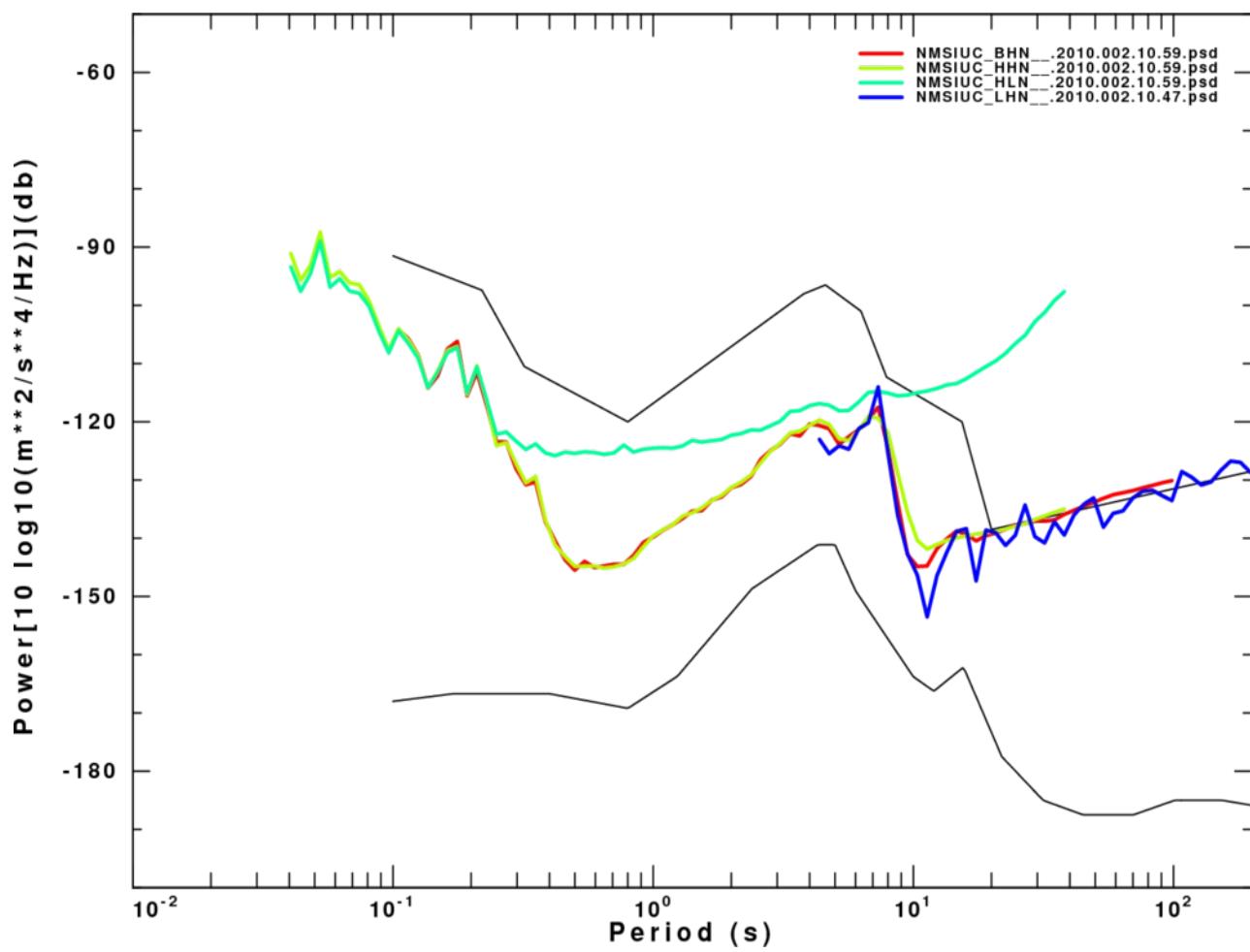
Fig. 9. Comparison of noise PSD form the N channel data streams at SIUC.
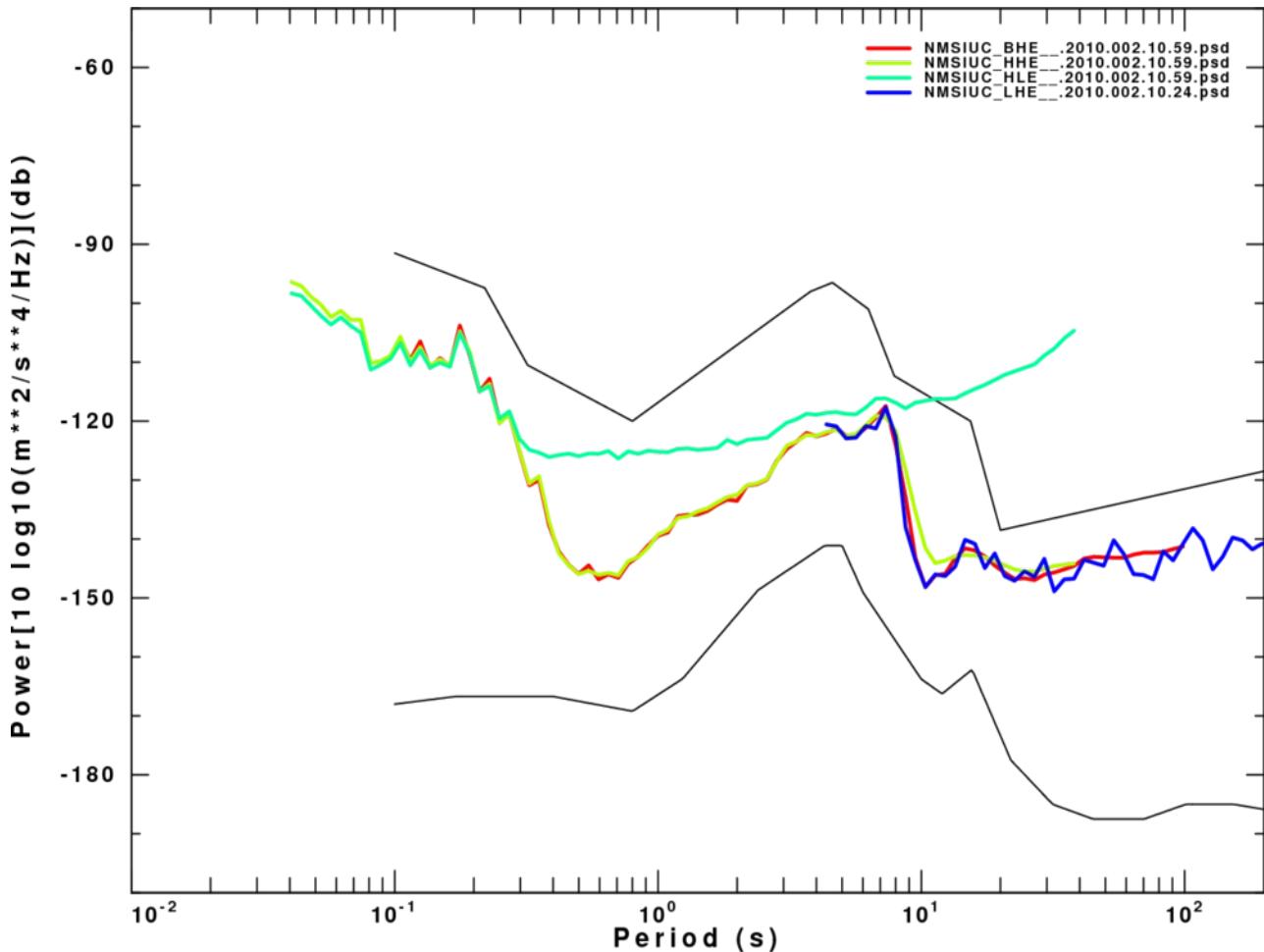
Fig. 10. Comparison of noise PSD form the E channel data streams at SIUC.

At times it may be useful to look at the raw rather than smoothed PSD to further understand the station noise. The current SIUC site is in the basement of a university building. At high frequencies we can see the cycling of the air handler system in the daily seismograms. This is also very apparent in the PSD's

Figures 11 and 12 present contents of the CALPLOT files *NMSIUC.HLZ.2010.002.PLT* and *NMSIUC.HHZ.2010.002.PLT.*
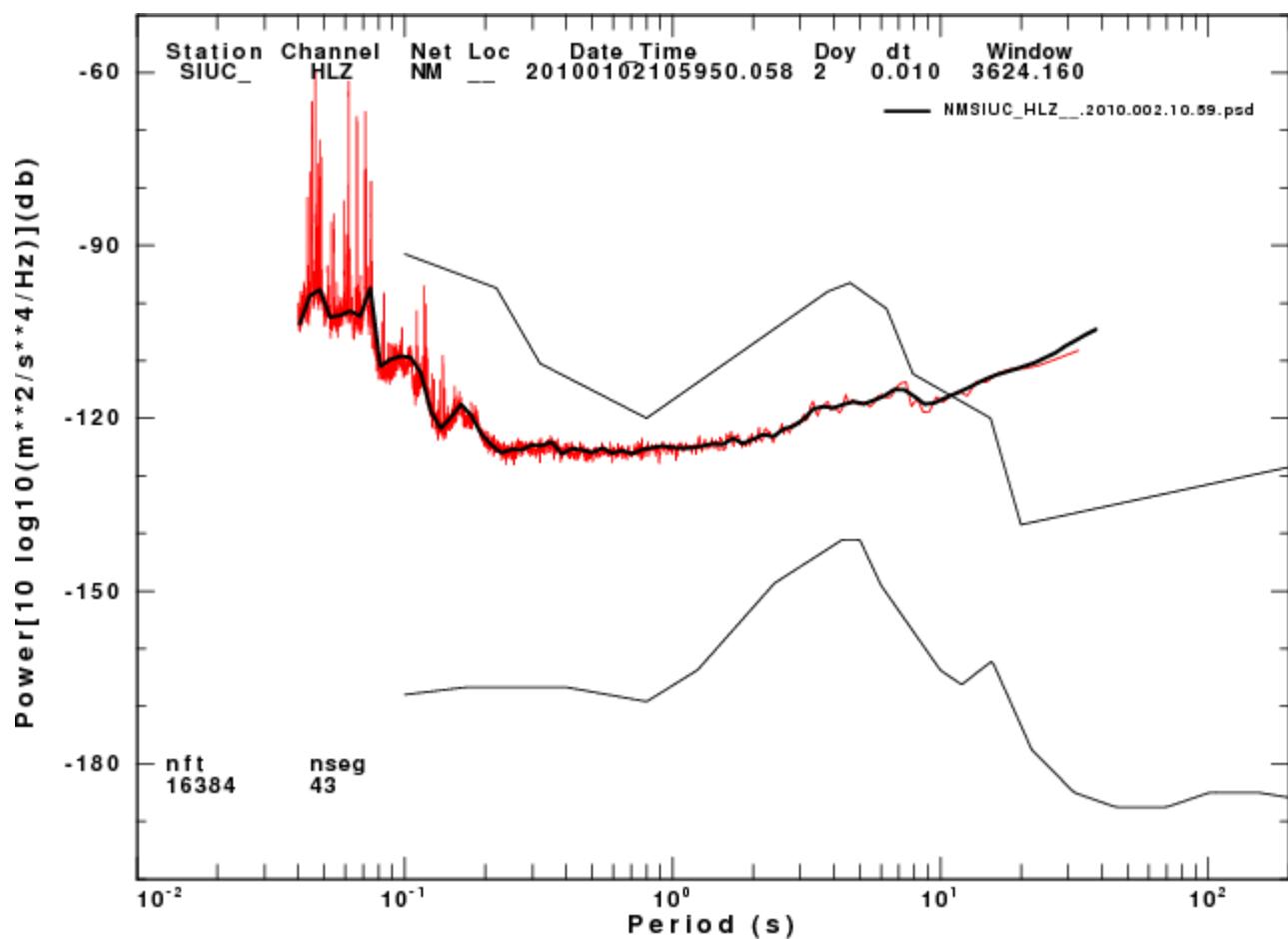
Fig. 11. Noise PSD for the HLZ component at SIUC.
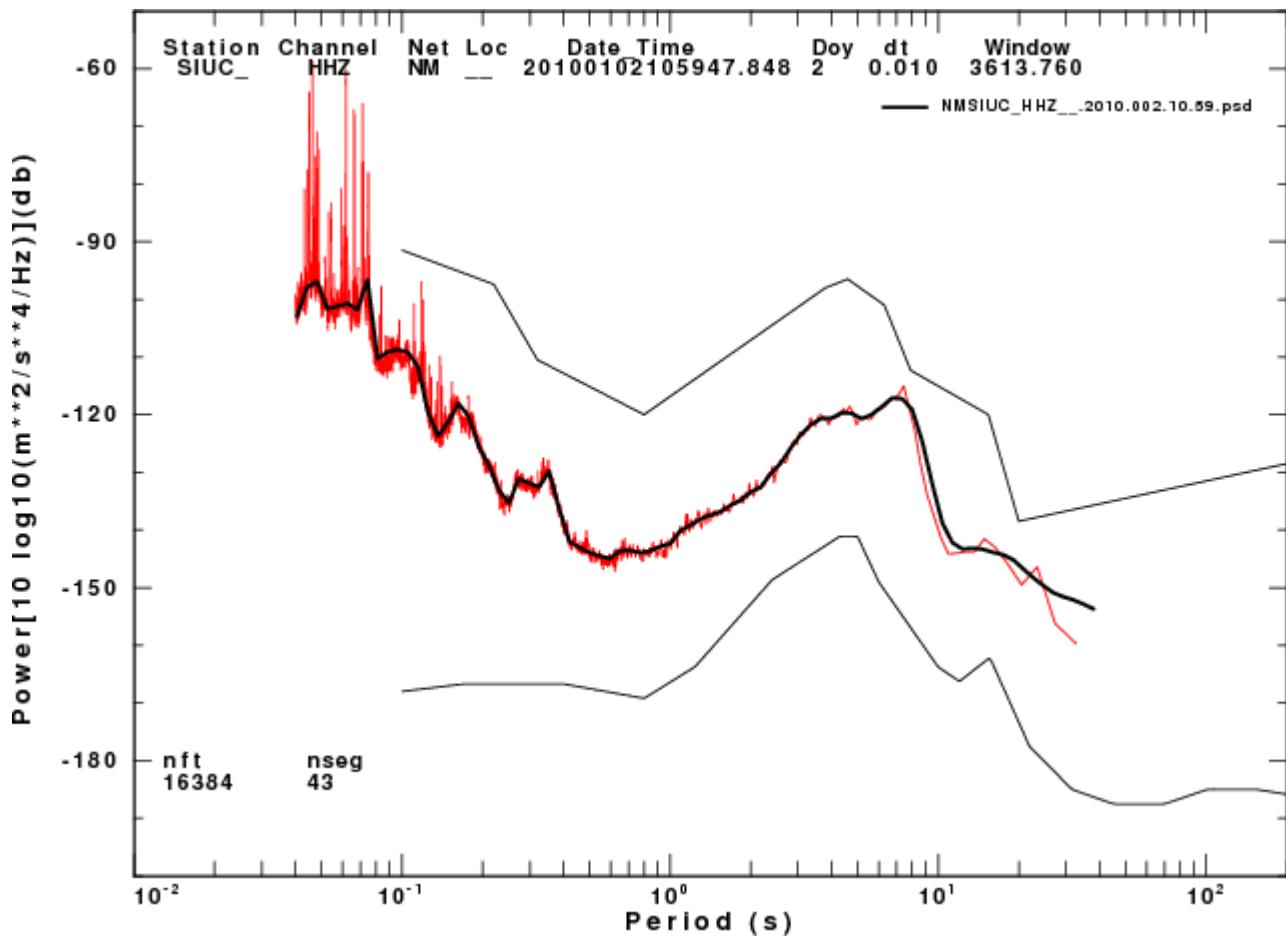
Fig. 12. Noise PSD for the HHZ component at SIUC.

NOTE: High noise levels may be reduced by moving the sensor.

3.4 Examples using SLU internal seiscomp files

SLU uses either the older DACOMMO/COMSERVE code to communicate with the Quanterra Q380 and Q680's, and seiscomp to communicate with the Quanterra Q330's. These communication programs retrieve field data and reorganize the data streams in terms of Station/Component/miniSEED. The resultant miniSEED is then QC'd for forwarding to IRIS for long-term archiving. We also preserve copies of the miniSEED. In the future, we will also convert the data streams from the Guralp CMG5-TD accelographs to miniSEED.

Since the USGS **CWBQuery** has an easy to use syntax, we decided to create a **SLUQuery** to access out data. This was accomplished in a shell script that uses **getopts**, **grep**, **qmerge** and the CPS version of **rdseed** and **evalresp.**

The **DOIT** script is the same as in Section 3.2, except that the lines

```
query –dccdbg –t sac –o "%N.SAC" –sacpz nm –fill 0 \
      –b "${DATETIME}" –d ${WINDOW} –s "NM.....BHZ.."
mdget –resp –b "${MDBEG}" –e ${MDEND} –s "NM.....BHZ.."
```

are replaced by the single line

```
SLUQuery  –b "${DATETIME}" –d ${WINDOW} –s "NM.......E.."
```

Note only the time window and search pattern are used. At present we cannot zero fill gaps, **SLUQuery** always returns the waveforms as sac files, and the use of rdseed is scripted to return waveforms as Sac files. **SLUQuery** is currently a 200 line shell script that is well annotated. As our seismic waveform storage system evolves, this script will be adapted as necessary.

Figure 13 shows the result of running the modified DOIT script with the parameters

```
DATETIME="2010/02/07–11:00:00"
```

```
SLUQuery  –b "${DATETIME}" –d ${WINDOW} –s "NM.......Z.."
```
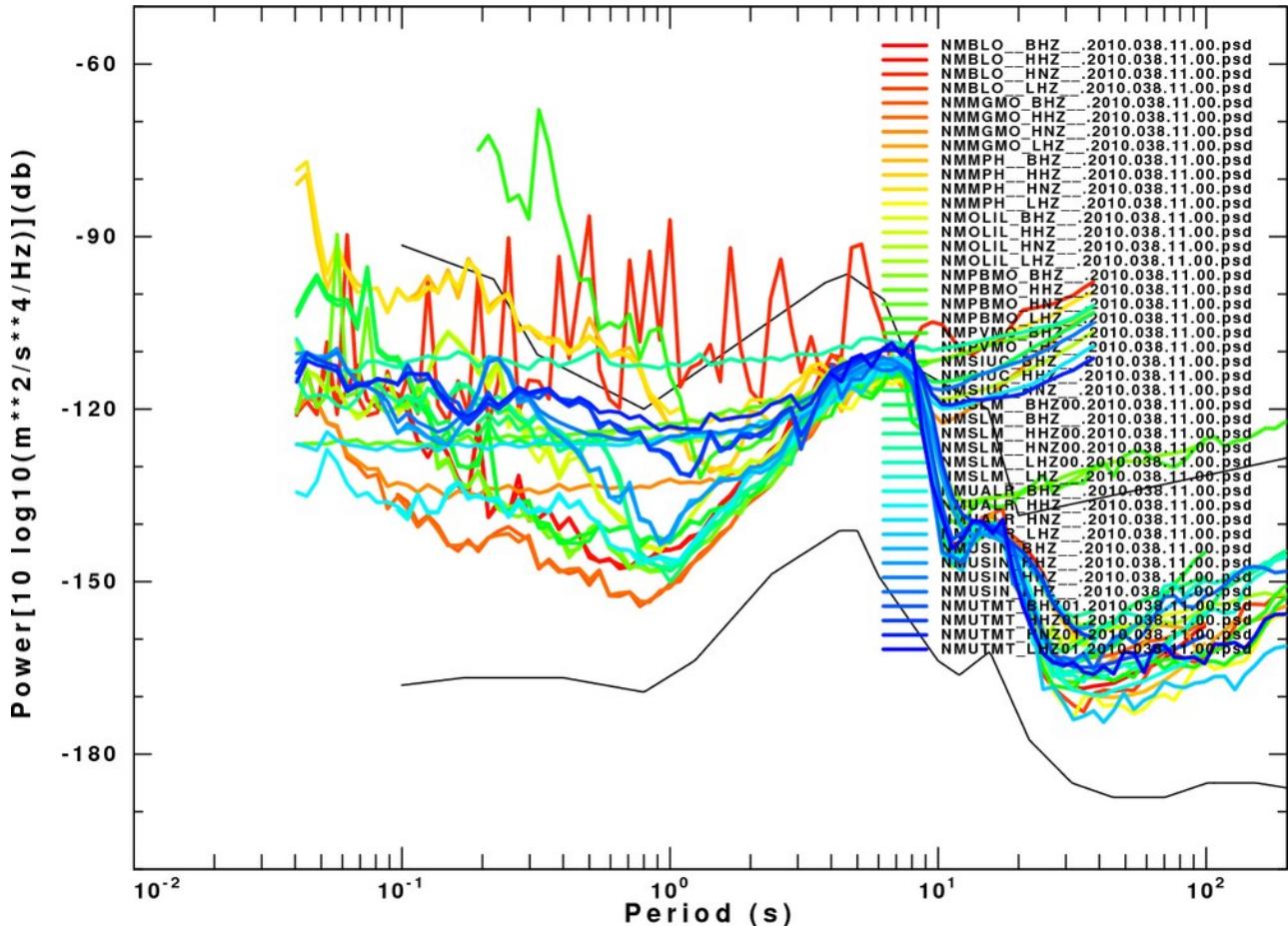

Fig. 13. Noise analysis on the vertical components of the SLU component of the NM network for one hour starting 11:00UT on February 7, 2010.

This figure shows that the script and processing works. Detailed analysis of the individual PSD files is required. The scatter is not surprising since accelerometer and broadband data streams are compared.

4.0 References

Ileachor, E. C., and B. W. Jervis (1993). *Digital Signal Processing: A Practical Approach*, Addison Wesley, Wokingham, England, 760pp

McNamara, D. E., and R. P Buland (2004). Ambient noise levels in the continental United States, *Bull. Seism. Soc. Am.* **94,** 1517-1527.

Peterson, J. (1993). Observation and modeling of seismic background noise, *U. S. Geol. Surv. Tech. Rept. 93-322,* 1-95.

Appendix – Calplot graphics

A.1 Introduction

Computer Programs in Seismology is distributed with its own graphics package to make the installation of the distributed software easier. Each non-interactive graphics program will create a binary, device-independent metafile of plotting commands, which must be converted for use by a specific hardware device. At the simplest level, the low level plotting commands are a sequence of pen up, pen move and pen down commands. Some of the early plotting devices supported were Calcomp mechanical plotters, Versatec electrostatic printers and Tektronix graphics terminals. Today graphics output is supported for X-Windows, and PostScript printers. Only a small subset of output devices are currently supported, primarily because of the existence of excellent conversion software; one example of which is *ghostscript* which converts PostScript to many devices. The ImageMagick and GraphicsMagick packages are very useful from converting Encapsulated PostScript to graphics bitmaps, such as PNG or JPEG, for inclusion into MS Word and PowerPoint documents.

If a CALPLOT device filter is named *plotdev*, one uses the program as follows:

       plotdev [options] < PLOTFILE   *for a screen device*

       plotdev [options] < PLOTFILE > temp_file *(create temp file***)**
       print temp_file   *(output to the actual printer***)**

Some common options are

       -S*scaling_factor*
              Multiply all plot moves by the ***scaling_factor (default 1.0)***

-R

        Rotate the plot by 90°

-Ffont

        Change the default font to number font. The default is Times Roman. A

        -F7 will invoke Helvetica Bold on a PostScript printer

Other commands are specific to the hardware device.

A.2. PostScript Output

```
plotnps [options]
-V                              Program Version
-Sscalefac   (default=1.0)      Plot magnifier
-Ppipe       (default stdout)   Pipe output to process pipe
-R           (default off)      Rotate plot 90 degrees
-N           (default off)      Turn off shading
-Ffont       (default 0)        Default font
                                 0 Times-Roman
                                 1 Times-Roman
                                 2 Times-Italic
                                 3 Times-Bold
                                 4 Symbol (Greek)
                                 5 Helvetica
                                 6 Helvetica-Oblique
                                 7 Helvetica-Bold
                                 8 Symbol (Greek)
                                 9 Courier
                                10 Courier-Oblique
                                11 Courier-Bold
                                12 Symbol (Greek)
-H30         (default H60)      Halftone for shading larger dots
-H60         (default H60)      Halftone for shading
-K           (default gray-)    Color PostScript output
-KW          (default gray-)    Color PostScript output, whitened spectrum
-KR          (default gray-)    Color output Red->White->Blue
-KB          (default gray-)    Color output Blue->White->Red
-G           (default gray-)    Gray PostScript output
             (default is gray shading but all black colored lines)
-B           (default 8.5x11)   Paper is 11 x 14
-L           (default 8.5x11)   Paper is 8.5x14
-A3          (default 8.5x11)   Paper is A3
-A4          (default 8.5x11)   Paper is A4
-W           (default 0)        Line width in units of 0.001 in or 0.0025cm)
-EPS                            EPS output
-Ttitle      (default off)      Title at bottom left of page
-X0xoff      (default 0)        x-offset in CALPLOT units
-Y0yoff      (default 0)        y-offset in CALPLOT units
-h                              Do not execute, show options
-?                              Do not execute, show options
```

Standards

To be compatible with PostScript display software and with word processing software that permits

inclusion of PostScript files, PostScript Document Structure Convention 3.0 (DSC 3.0) is followed.

Plotspace Mapping

The CALPLOT definition of axes is such that the X-axis is horizontal and the Y-axis is vertical. This is then mapped onto a printed page of dimension 8.5" x 11". In the default case the X-axis is mapped onto the long dimension of the paper. The plot space on the paper is demonstrated in Figure 1.
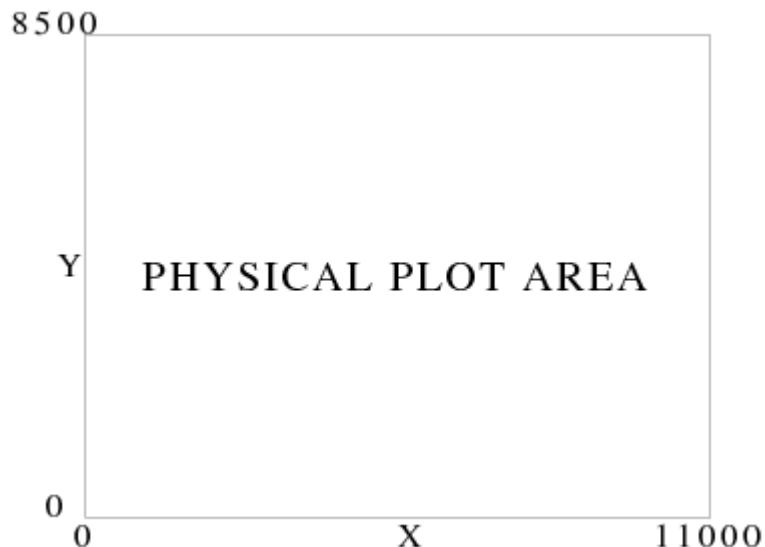
Fig. A.1. Default mapping of CALPLOT plot space onto physical page showing the CALPLOT units.

The -R option rotates the mapping, such that the Y-axis is mapped onto the long dimension of the paper. This plot space on the paper is shown in Figure 2.
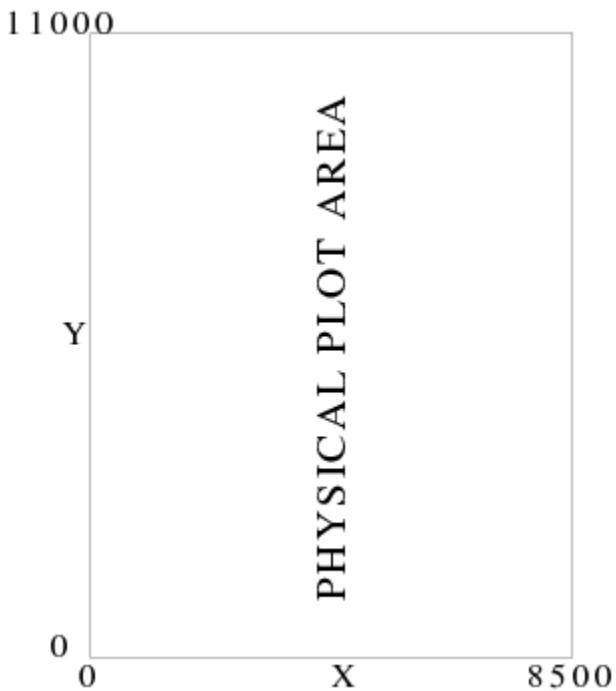
Fig. A.2. Mapping of CALPLOT plot space onto physical page using -R

.

Note that the CALPLOT plot space is mapped onto a rectangular page with no distortion of the unit lengths of the X- or Y-axes.

EPS EXAMPLES

The PostScript plot space is assumed to be that the X-axis is horizontal with a length of 8.5" and the Y-axis is vertical with a length of 11.0". The use of the **-EPS** or **-LEPS** options permits a plot to be able to fit within these limits. For the **-EPS** option, the CALPLOT X-axis will still be horizontal, which is required for inclusion in **groff** or **latex** documents. The default and **-R** option changes the lengths of the plotted axes in the manner consistent with Figures 1 and 2. The **-LEPS** option make the X-axis parallel to the long direction of the page.

The following examples use the second page of the graphics test file PLTTST to illustrate the result of using this program with different options.
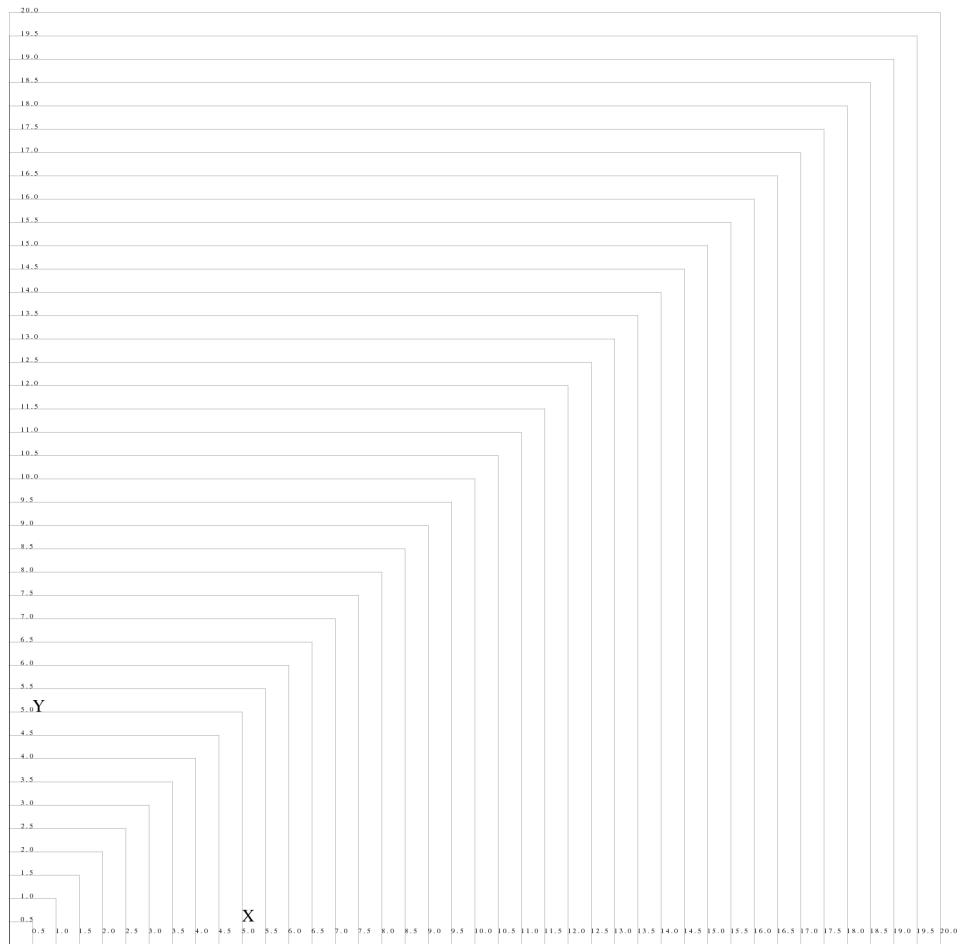
Fig. A.3. This is the result of using plotnps -EPS < plt > plttst.eps
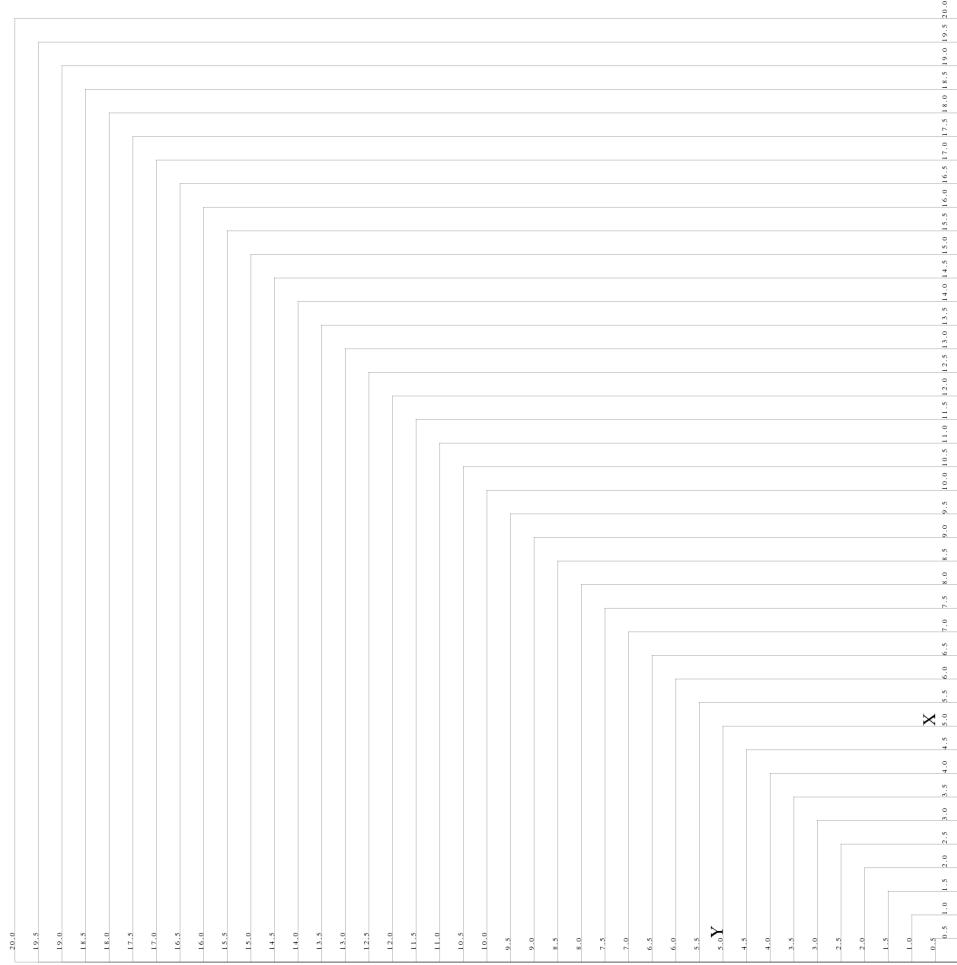
Fig. A.4. This is the result of using plotnps -EPS -R < plt > plttst.rps

A.3. X11 Output

The program plotxvig is a native X11 program for use under the X11 windowing system. It is based on the XviG Version 1.1 package (Antoon Demarrée, IMEC, ©1993). This program supports 35 unique colors in its palette. If these colors are not available, dithering is used to create the apparent set. This package is also the basis of interactive X11 software.

Program control is through the command line:

```
plotxvig [options]
    -V                    Program Version
    -Sscalefac (default=1.0)  Plot magnifier
```

```
-R            (default off)  Rotate plot 90 degrees
N             (default off)  Turn off shading
-I            (default off)  Invert background (e.g., black)
-Ffont        (default 0)    Default font
                             0 Roman
                             1 Roman
                             2 Italic
                             3 Bold
                             4 Symbol (Greek)
-K            (default color) Color output Red->Green->Blue
-KR           (default -K  ) Color output Red->White->Blue
-KB           (default -K  ) Color output Blue->White->Red
-Kr           (default -K  ) Color output White -> Red->BLue->White
-Kb           (default -K  ) Color output White->Blue->Red->White
-G            (default -K  ) Gray output
-W            (default 0)    Line width in units of 0.001 in or 0.0025cm)
-geometry     (default off)  Parse X11 geometry string
 -geometry WIDTHxHEIGHT{+-}XOFF{+-}YOFF
-p             Put up positioning grid every 1000 CALPLOT units, same as -p1
-p2 every 500, -p4 every 250, -p10 every 100
-X0xoff       (default 0)    x-offset in CALPLOT units
-Y0yoff       (default 0)    y-offset in CALPLOT units
-C            (default off)  Implement crosshairs for the mouse cursor
-h                           Do not execute, show options
-?                           Do not execute, show options
```

To provide additional user control, the command line arguments can be placed in the environment by separating them by colons (:) with no intervening spaces. This is the only way to change display options when using the graphics libraries for interactive plots, such as in the programs **gsac**, **do_mft** or **do_pom**. To force a scale factor of 0.5, and the images size of 800x600 one would set the environment parameter PLOTXVIG

```
PLOTXVIG=:-S0.5:-g:800x600:
export PLOTXVIG      (under sh or bash)
```

or

```
setenv PLOTXVIG=:-S0.5:-g:800x600:   (under csh)
```

The X11 screen is viewed as a piece of paper exactly 10.0" wide and 8.0" high (approximately 25.4 cm wide by 20.32 cm high). The default screen has dimensions of 800 x 640 pixels, which can be changed through the window manager when the program begins. The following screen would appear:
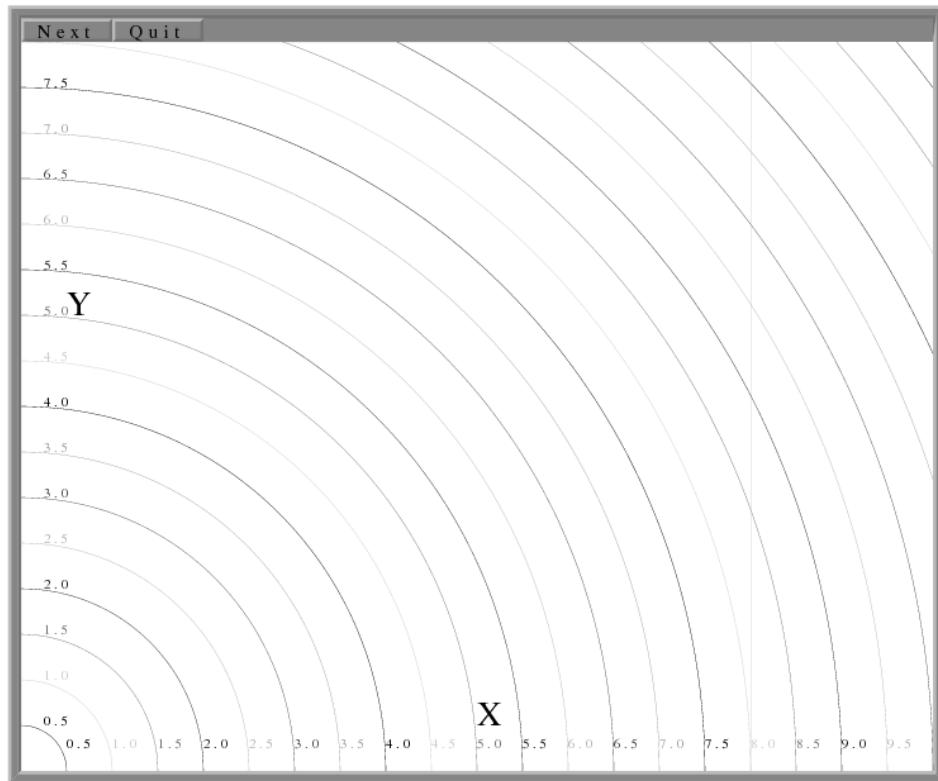
When the page is completely drawn, a cursor will appear. One can use this to point to a feature of interest. The following actions can be performed:

- Pressing and releasing the Left Mouse Button will advance the page.
- Pressing the menu button Next will advance the page.
- Pressing the menu button Quit terminates the plot.

To be consistent with X11, the geometry of the plot window can be specified by an entry in the .Xdefaults file:

```
plotxvig.calxvig.plotxvig.geometry: 1000x800+100+50
```

In order of importance, an entry such as this overrides a command line or environment option. For the other options, the command line overrides the PLOTXVIG environment control.

Problems:

Resizing a window after plotting begins will truncate the plot, if the window is smaller, or will have unused areas. Because of the size of the binary plotfiles, there is no way to rewind and redraw a plot.

Instead a backup image is used.

Finally **plotxvig** works by starting two UNIX processes: one do the drawing, the other to handle events and to place the drawing on the screen - an interesting use of interprocess communication. You may find yourself with a display that is not responsive - this usually happens because one, but not both processes have terminated. Use ps to list the processes, select the process ID number, PID, and and then kill PID to get rid of **calxvig** and **plotxvig**.

A.4. Figure Manipulation

The program **reframe** permits manipulation of a CALPLOT figure, either by changing the position on the page or by imposing a primitive clipping. Options exist to select one figure of a multi-page plot file, and to merge plot files. The output of this program is another plot file. The program input is from the last argument on the command line, if that argument is not a command flag, or the standard input

Program control is through the command line:

```
reframe  [flags], where the command flags are
     -O
          Redirect the output to the standard output. Otherwise      a plotXXXXX
file            will be created, where XXXXX is a unique identification number.
     -P
          Force the output to be a plot file. This the default.
     -Mmergefile
          This is the file that will be superimposed onto the original file.
     -XLx_low_clip
          (default = -100000000)
     -XHx_high_clip
          (default =  100000000)
     -YLy_low_clip
          (default = -100000000)
     -YHy_high_clip
          (default =  100000000)
          A selected position of the input figure can be passed through to the
           output. The selected region is bounded by these coordinates.
     -X0x_origin
     -Y0y_origin
          These values are added to the (x,y) coordinates of all input values
          within the clipping window to shift the resulting figure on the page.
```

The sequence of operations is that first the image is clipped, and then the origin is shifted.

To illustrate the usage of the program, consider the following two examples:

To merge the second frames of two plot files, one need only do

```
reframe -N2 -O -MPLOTrhwvint < PLOTrefplt > PLOTrefplt2
```

Note that the page number flag applies to both input files. It may be necessary to run the program three times to select the desired pages, first two runs, and then to merge the output using the temporary files.

To select the first three pages of a multi-page plotfile, and then to combine them on to a single page,

```
reframe -V  -XH8500 -YH11000  -N1 -O < TABL > hunk1
    ( retrieve page 1 and save in the file hunk1)
reframe -V  -XH8500 -YH11000  -X0+8750  -N2 -O < TABL > hunk2
    ( retrieve page 2, move plot 8750 units to right and
        save in the file hunk2)
reframe -V  -XH8500 -YH11000  -X0+17500 -N3 -O < TABL > hunk3
    ( retrieve page 3, move plot 11000 units to right and
        save in the file hunk3)
reframe -V -N1 -O -Mhunk2 hunk1 > munk1
    ( merge the files hunk2 and hunk1 into the file munk1)
reframe -V -X0+1000 -Y0+1000 -N1 -O -Mhunk3 munk1 > PLOTreframe2
    ( merge files hunk3 and munk1, and shift the origin 1000 units to
        the right and upward)
```

The CALPLOT programs are used, 1000 units correspond to 1.000 inches on the hardcopy plot.

The results of another example are shown in Figures A.5 and A.6. The object is to cut Figure A.5 into four quadrants centered at (4.0,4.0) and to exchange the upper right with the lower left quadrant and the upper left with the lower right quadrant. The commands used are as follow:

```
reframe -N1 -O < PLTTST > p
reframe -N1 -O -X0+4000 -Y0+4000 -XL0000 -XH4000 -YL0000 -YH4000 < p1 > g1
reframe -N1 -O -X0-4000 -Y0+4000 -XL4000 -XH8000 -YL0000 -YH4000 < p1 > g2
reframe -N1 -O -X0+4000 -Y0-4000 -XL0000 -XH4000 -YL4000 -YH8000 < p1 > g3
reframe -N1 -O -X0-4000 -Y0-4000 -XL4000 -XH8000 -YL4000 -YH8000 < p1 > g4
reframe -N1 -O -Mg1 < g2 > g5
reframe -N1 -O -Mg3 < g4 > g6
reframe -N1 -O -Mg5 -X0+1000 -Y0+1000 < g6 > g7
plotnps -F7 -W10 -G -EPS < g7 > g7.eps
rm p1 g1 g2 g3 g4 g5 g6 g7
```
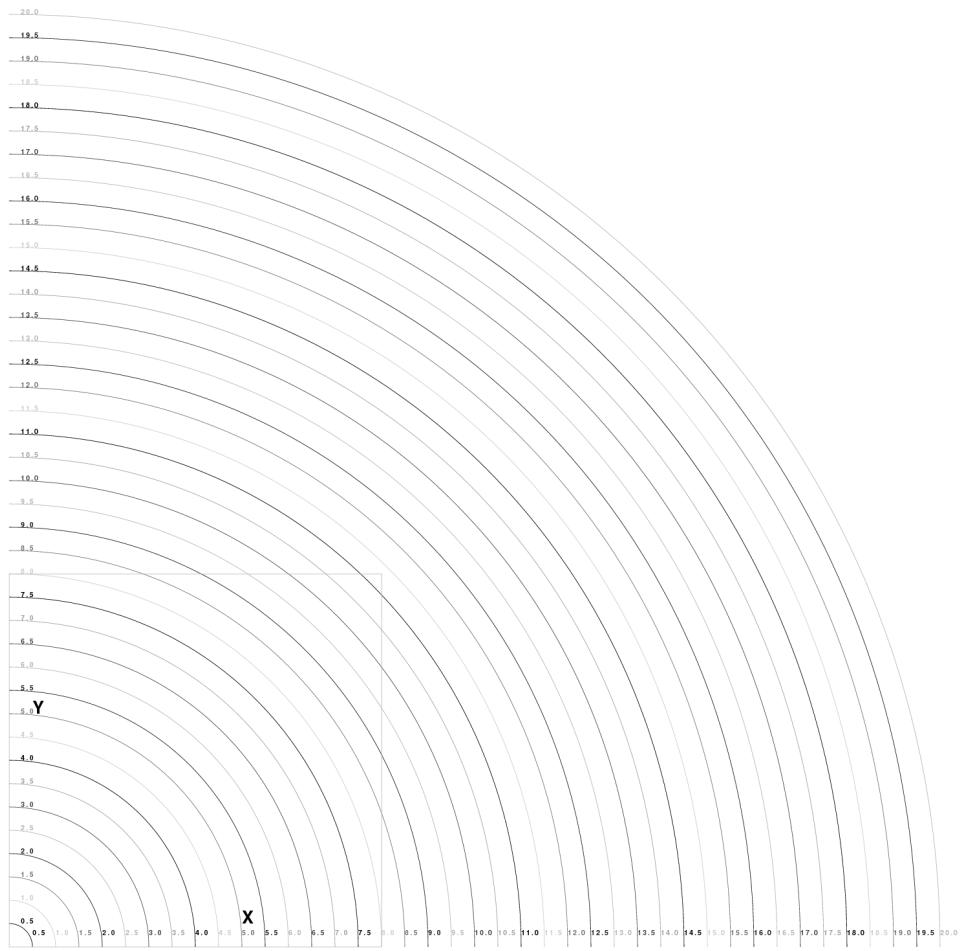
Fig. A.5 . Initial plot to be sectioned
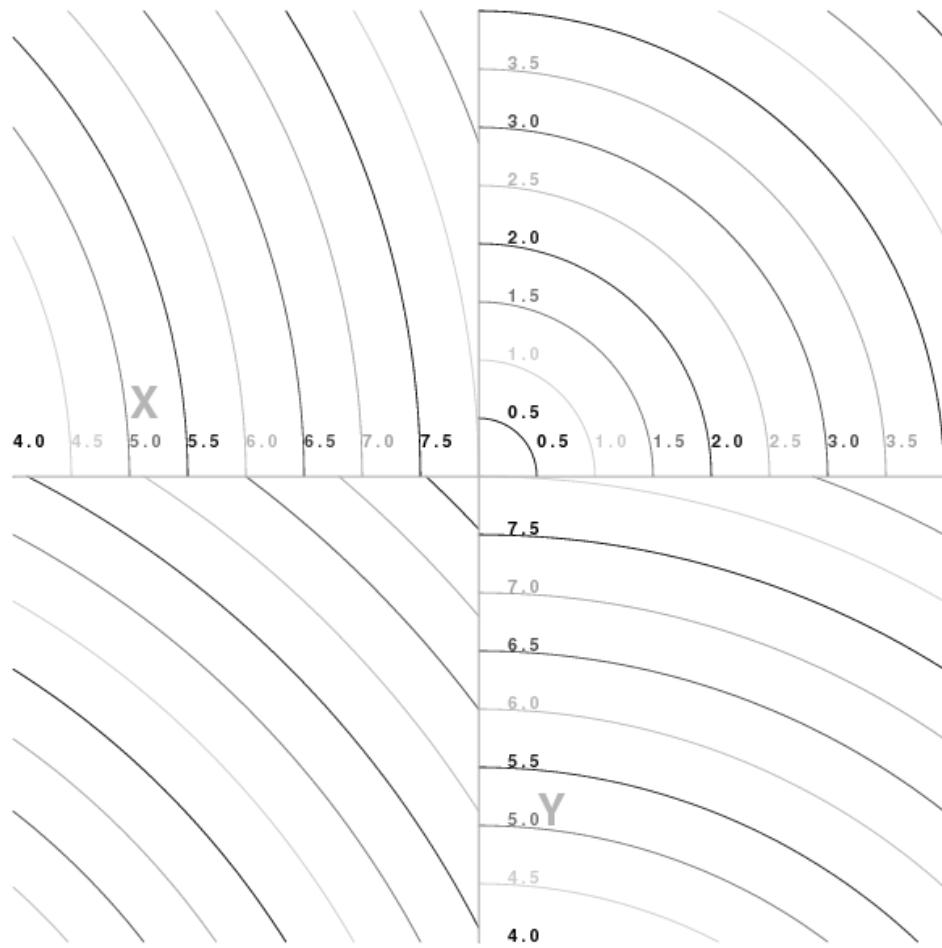
Fig. A.6. Result of clipping and shifting

A.5. CALPLOT Colors

Many programs permit the user definition of colors for curves. These are invoked using

> -Kflag

The CALPLOT graphics uses a set of predefined colors that take on slightly different meanings depending upon whether the plot program (plotxvig, plotnps, and plotgif), is invoked with the -G, -K ,

-KR or -KB flags. Values of *kolor* between 0 and 999 are repeated as a specified sequence of 7 colors. Values in the range 1000 - 1100 select a palette of continuous color tones selected by the use of these flags. The table below defines some of these values as do the figures, which are best viewed on a color terminal screen using GhostView or Acroread.

Kolor	-G	-K	-KR	-KB
0	Background	Background	Background	Background
1	Foreground	Foreground	Foreground	Foreground
2	(see below)	Red	(see below)	(see below)
3		Green		
4		Blue		
5		Orange		
6		Blue-Green		
7		Yellow		
8	Foreground	Foreground	Foreground	Foreground
9		Red		
999				
1000	Lt. Gray	Red	Red	Blue
1025		Orange	Lt. Red	Lt. Blue
1050	Med. Gray	Green	White	White
1075		Blue-Green	Lt. Blue	Lt. Red
1100	Dark Gray	Blue	Blue	Red

Normal plotting uses the -G and -K flags. Displays of continuous color maps can use the -KR and -KB modes if the color indices are programed to represent a range of negative - positive values with white representing a median value. The following figures show the resulting colors for a given choice of the kolor index.

Computer Programs in Seismology - sacpsd
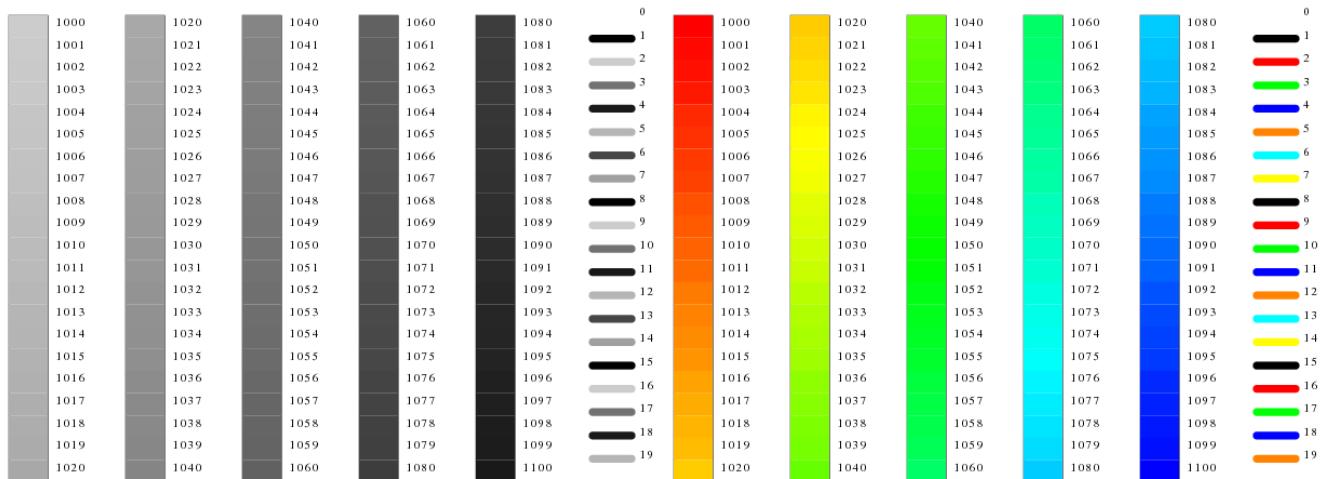
plotnps -G < GRAYSC

plotnps -K < GRAYSC
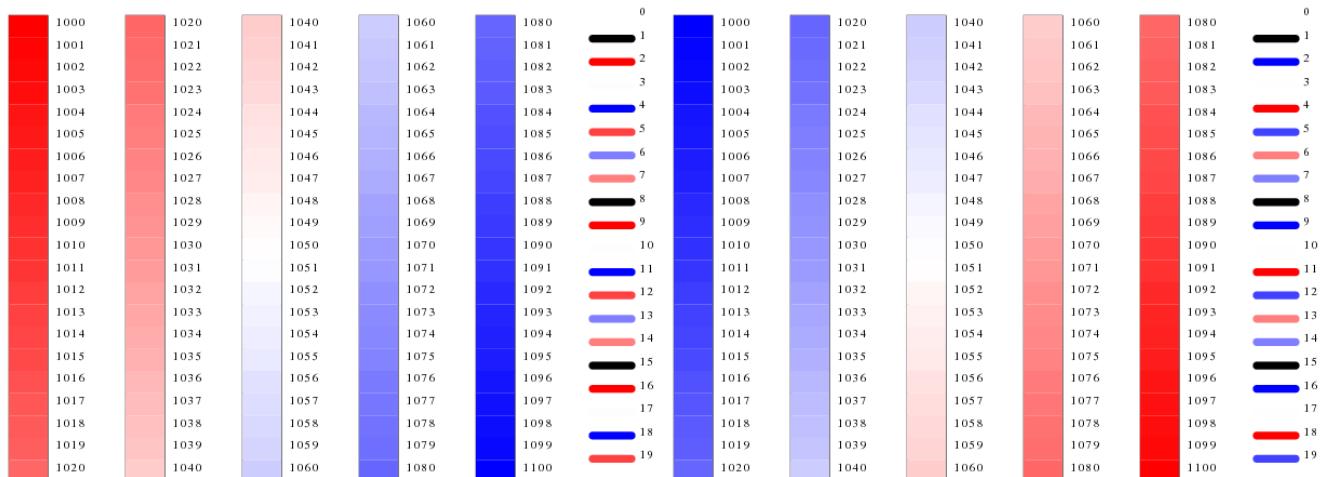
plotnps -KR < GRAYSC

plotnps -KB < GRAYSC