SAC Manual

New Users

Overview

SAC was designed as an aid to research seismologists in the study of seismic events. As such, it is used for quick preliminary analyses, for routine processing, for testing new techniques, for detailed research, and for creating publication quality graphics. It is used by both computer novices and experts. In order to make SAC quick to learn and easy to use, default values for all operational parameters were carefully chosen. At the same time, almost all of these parameters are under direct user control. This design combines ease of use with significant flexibility.

User Interface

SAC is an interactive command driven program. Commands may be typed at the terminal or placed in a macro file. SAC commands fall into three main categories: parameter-setting, action-producing and data-set manipulation. The parameter-setting commands change values of internal SAC parameters. Action-producing commands perform some operation on the signals currently in selected memory based upon the values of these parameters. Data-set commands determine which files are in active (selected) memory and therefore will be acted upon (data-set commands are not currently operational). The effect of a parameter-setting command remains in effect until it is reset. The effect of an action-producing command is immediate and transitory. Action-producing commands also have options which normally remain in effect until reset. These options, however, apply only to that particular command. The underlying assumption is that you are more likely than not to want to use the same values the next time you execute the same command. When you start up SAC, default values are defined for all of these parameters. SAC can be reinitialized to this default state at any time by executing the INICM command.

Mode of Operation

Each signal is stored in a separate data file. Each data file contains a header that describes the contents of that file. See the section on Data File Format for details. Signals are read from disk into memory using the READ command. CSS 3.0 formated flat files can be read using the READCSS command. SAC can process up to 200 signals of arbitrary size at a time. Once data is in memory other commands are typed at the terminal (or read from a macro file) to perform operations on these signals. All operations work concurrently on all signals in memory. You can look at the results at any time using the plot commands. There are several plot formats to choose from. You have control over titles and labels, plot limits, file identifications, axes and tick mark locations, etc. You can also save the results of these operations at any time using the WRITE command. All of the commands are described briefly in the sections on Analysis Capabilities and Graphics Capabilities of this manual and documented in detail in the Commands Reference Manual.

How SAC Handles Time

The SAC header contains a reference or zero time, stored as six integers (NZYEAR, NZJDAY, NZHOUR, NZMIN, NZSEC, NZMSEC), but normally printed in an equivalent alphanumeric format (KZDATE and KZTIME). This can be set to any reference time you wish. It is often the time of the first data point, but can also be the origin time of the event, midnight, your birthday, etc. It does not even have to be a time encompassed by the data itself. All other times are offsets in seconds from this reference time and are stored as floating point values in the header:

B Begin time of the file. E End time of the file. O Event origin time. A First arrival time. F Fini (end of signal.) Tn Time markers, where n is an integer from 0 to .9

Many SAC commands work with these header variables. They are discussed in the two sections on Analysis and Graphics Capabilities. The CUT command is one of the most important of these commands. It lets you select portions of data files using the offset time header fields to be read in by subsequent READ commands. Examples of the use of CUT and READ are shown in the Tutorial Guide and also in the Command Reference Manual.

Getting Started

The second best way to get started (after you have finished reading this section) is to sit down at a terminal or workstation with a copy of the Tutorial Guide For New Users, and try the examples as you read the manual. (The best way is corner an experienced and patient user and make him or her show you how things work!) The examples in the tutorial cover the basic commands you need to know in order to proceed further. The next step is to browse through the lists (one alphabetical and the other functional) of SAC commands in the Command Reference Manual, find a few that sound interesting, and try them. SAC is fairly careful in checking for errors so you can't do much harm in experimenting. After you have become somewhat comfortable using SAC, you might want to return to this manual to get a better overview of the program. Be sure and read the section on SAC macros. They will definitely save you time when you get ready to do fairly complicated or repetitive analysis on large data sets. Eventually you will probably want to sit down and wade through the complete Command Reference Manual to get a complete picture of what you can do with SAC. It's better to defer that step until you have used the code for awhile.

SAC Tutorial Guide for New Users

Overview

SAC was designed as an aid to research seismologists in the study of seismic events. As such, it is used for quick preliminary analyses, for routine processing, for testing new techniques, for detailed research, and for creating publication quality graphics. It is used by both computer novices and experts. In order to make SAC quick to learn and easy to use, default values for all operational parameters were carefully chosen. At the same time, almost all of these parameters are under direct user control. This design combines ease of use with significant flexibility.

README

The first step is to study the README file that is in the top directory of the distribution: sac. It gives detailed instructions about setting up the environmental variables necessary to run SAC and other pieces of useful information.

User Interface

SAC is an interactive command driven program. Commands may be typed at the terminal or placed in a macro file. SAC commands fall into three main categories: parameter-setting, action-producing and data-set manipulation. The parameter-setting commands change values of internal SAC parameters. Action-producing commands perform some operation on the signals currently in selected memory based upon the values of these parameters. Data-set commands determine which files are in active (selected) memory and therefore will be acted upon (data-set commands are not currently operational). The effect of a parameter-setting command remains in effect until it is reset. The effect of an action-producing command is immediate and transitory. Action-producing commands also have options which normally remain in effect until reset. These options, however, apply only to that particular command. The underlying assumption is that you are more likely than not to want to use the same values the next time you execute the same command. When you start up SAC, default values are defined for all of these parameters. SAC can be reinitialized to this default state at any time by executing the INICM command.

Mode of Operation

Each signal is stored in a separate data file. Each data file contains a header that describes the contents of that file. See the section on Data File Format for details. Signals are read from disk into memory using the READ command. CSS 3.0 formated flat files can be read using the READCSS command. SAC can process up to 200 signals of arbitrary size at a time. Once data is in memory other commands are typed at the terminal (or read from a macro file) to perform operations on these signals. All operations work concurrently on all signals in memory. You can look at the results at any time using the plot commands. There are several plot formats to choose from. You have control over titles and labels,

plot limits, file identifications, axes and tick mark locations, etc. You can also save the results of these operations at any time using the WRITE command. All of the commands are described briefly in the sections on Analysis Capabilities and Graphics Capabilities of this manual and documented in detail in the Commands Reference Manual.

How SAC Handles Time

The SAC header contains a reference or zero time, stored as six integers (NZYEAR, NZJDAY, NZHOUR, NZMIN, NZSEC, NZMSEC), but normally printed in an equivalent alphanumeric format (KZDATE and KZTIME). This can be set to any reference time you wish. It is often the time of the first data point, but can also be the origin time of the event, midnight, your birthday, etc. It does not even have to be a time encompassed by the data itself. All other times are offsets in seconds from this reference time and are stored as floating point values in the header:

B Begin time of the file. E End time of the file. O Event origin time. A First arrival time. F Fini (end of signal.) Tn Time markers, where n is an integer from 0 to .9

Many SAC commands work with these header variables. They are discussed in the two sections on Analysis and Graphics Capabilities. The CUT command is one of the most important of these commands. It lets you select portions of data files using the offset time header fields to be read in by subsequent READ commands. Examples of the use of CUT and READ are shown in the Tutorial Guide and also in the Command Reference Manual.

Getting Started

The second best way to get started (after you have finished reading this section) is to sit down at a terminal or workstation with a copy of the Tutorial Guide For New Users, and try the examples as you read the manual. (The best way is corner an experienced and patient user and make him or her show you how things work!) The examples in the tutorial cover the basic commands you need to know in order to proceed further. The next step is to browse through the lists (one alphabetical and the other functional) of SAC commands in the Command Reference Manual, find a few that sound interesting, and try them. SAC is fairly careful in checking for errors so you can't do much harm in experimenting. After you have become somewhat comfortable using SAC, you might want to return to this manual to get a better overview of the program. Be sure and read the section on SAC macros. They will definitely save you time when you get ready to do fairly complicated or repetitive analysis on large data sets. Eventually you will probably want to sit down and wade through the complete Command Reference Manual to get a complete picture of what you can do with SAC. It's better to defer that step until you have used the code for awhile.

Assuming the environmental variables have been set so that SAC is in your page, to get started one simply types "sac".

SAC will then print a short headline including the number and date of the version you have on your system. It may also print a bulletin giving some current information. SAC will then ask you for input by sending the prompt "SAC>".:

```
% sac
 SEISMIC ANALYSIS CODE [08/15/2006 (Version 100.1)]
 Copyright 1995 Regents of the University of California

SAC>
```

Interaction

SAC is an interactive command driven program. This means that you must type a command to get SAC to do something. It does not prompt you for input. Commands may be typed at the terminal or placed in a command file. Symbols within a command are separated by spaces and commands within a given line may be separated by a semicolon.

We'll start by creating a simple function:

```
SAC> FUNCGEN impluse
```

This generates an impulse function and stores it in SAC's memory. To see what this function looks like on your screen type:

```
SAC> BEGINDEVICES xwindows
SAC> PLOT
```

In this example device is the name of the graphics device you are using. If you don't use the BEGINDEVICES command, SAC will use the default device, which is X windows on most.

Abbreviations

There are abbreviations for the most used SAC commands. For example, fg , bd , and p are the abbreviations for FUNCGEN, BEGINDEVICES, and PLOT

perform some operation on the data files currently in memory based upon the values of these same parameters. The effect of a parameter-setting command remains in effect until it is reset. The effect of an action-producing command, however, is immediate and transitory. For example, the parameter-setting command, YLOG, tells SAC to use logarithmic interpolation for the y axis in subsequent plots. The action-producing command, PLOT, does the actual plotting. Options to action-producing commands also remain in effect until reset just like parameter-setting commands. The underlying assumption is that you are more likely than not to want to use the same values the next time you execute the same command.

Default Values

All commands have "nice" default values for most options. The use of current and default values for command options can save you a lot of typing. For example, let's look at the BANDPASS command. This command applies a bandpass filter to the data currently in memory:

```
SAC> fg impulse npts 100 delta 0.01
SAC> bandpass bessel corner 0.1 0.3 npole 4
```

These two commands generate an impulse function and then apply a bandpass filter to that impulse. The filter is a four-pole Bessel filter with corner freqencies at 0.1 and 0.3 Hz. You can see the result in the time domain by typing PLOT or you can see the amplitude reponse by taking the Fourier transform and using the PLOTSP command:

```
SAC> fft
SAC> plotsp am
```

You can now try a different set of corner frequencies very easily:

```
SAC> fg
SAC> bandpass corner 0.2 0.5
```

SAC generates the same impulse function and applies the same Bessel filter except for the new corner frequencies.

SAC Data Files

SAC is a program to examine, analyze, and plot data. This data is stored on disk as SAC data files. Each data file contains a single data set. For seismic data this means a single data component recorded at a single seismic station. SAC does not currently work on multiplexed data. The data will generally be evenly spaced time series data. SAC can also handle unevenly spaced data and spectral data. The spectral data can be in either real-imaginary or amplitude-phase format.

SAC Header

Each data file also contains a header record which describes the contents of that file. Certain header entries are always present (e.g., the number of data points, the file type.) Others are always present for certain file types (e.g., sampling interval, begin time, etc. for evenly spaced time series files.) Other header variables provide information needed by a particular operation (e.g., seismic component orientation used by the ROTATE command.) Still others are not used by SAC at all. They are simply informational. The LISTHDR command displays the contents of the headers for the data files currently in memory. You may wish to examine the header from the sample seismogram mentioned earlier:

```
SAC> FG seismogram
SAC> LH
```

If a particular header variable does not have a value for a particular file, then that variable is said to be "undefined" for that file. The LISTHDR command does not list undefined header variables, usless it is invoked with the INC or INCLUSIVE option (which includes undefined header variables).

Header Variables

Each header variable is described in the Users Manual. The most important ones are also listed below:

NPTS Number of points in data set. B Beginning value of the independent variable. E Ending value of the independent variable. IFTYPE Type of file. LEVEN TRUE if data set is evenly spaced. DELTA Increment between evenly spaced samples. IDEP Type of dependent variable. KZDATE Alphanumeric form of GMT reference date. KZTIME Alphanumeric form of GMT reference time. A First arrival time (seconds relative to reference time.) T n User defined time picks or markers, n=0,9.

Reading Data Files

SAC commands work on data already in SAC's working memory, not data on disk. The READ command is used to transfer data from disk to memory. Up to 100 data files can be in memory at the same time, and this limitation should be removed in upcoming versions. These can be of any size up the maximum size of SAC's working memory. You can use wildcard characters in the READ command to represent groups of files which have a similiar set of characters in their names. Each time you use the READ command to transfer data from disk to memory the data currently in memory is destroyed. If you want this data saved, you must write it to diskbefore reading more data into memory. There is an option called MORE in the READ command that lets you read data into memory without destroying the old data. See the Command_Reference for details.

Writing Data Files

At any time during your analysis, you may transfer this modified data back to disk using the WRITE command. You may overwrite the old data files on disk using the OVER option or create new ones by specifying their file names. Action commands (such as ADD, DECIMATE, and FFT) modify the data that is currently in memory. The data files on disk are not modified.

Reading and Writing Examples

The examples below demonstrates several uses of the READ and WRITE commands.

Scaling Example

The first example reads two files into memory, multiplies each data point in each file by a constant, and then writes the results to disk in two new files:

```
SAC> R file1 file2
SAC> MUL 10 20
SAC> W file3 file4
```

Decimation Example

The next example reads a single file into memory, desamples the data by a factor of five (DECIMATE also applies an anti-aliasing filter), and then writes the results back to disk using the same file name:

```
SAC> R file1 file2 file3 file4
SAC> DECIMATE 5
SAC> WRITE OVER
```

Sample Data Files

You're going to need some data files for use in the next section on plotting. You'll also need them if you want to try any of the other commands discussed later in this guide. If you don't have any sample SAC data files around to play with, you can use FUNCGEN to generate some. This is shown in the example below:

```
SAC> fg trianlge npts 200 delta 1.0
SAC> write file1
SAC> fg boxcar
SAC> write file2
SAC> fg step
SAC> write file3
```

This results in you having three files in your directory called file1, file2, file3 which contain the triangle and boxcar, and step functions respectively. Each will have 200 data points in them and be sampled at 1 sample per second. If you already had files in your directory by those names, they would be replaced by these new ones.

Displaying the Results

After reading data into SAC you can see it on your screen in several different formats using the various plot commands. Default values for each of the graphics display commands have been chosen to make it as easy as possible to display your data. By changing these default values before plotting, you also have complete control over the details of how each plot will look.

You've already used PLOT to display data files. With this command, each data file is plotted one at a time. SAC pauses between files to give you a chance to examine the data. This is shown in the following example.:

```
SAC> read file1 file2 file3
SAC> plot
Waiting [press return]
Waiting [press return]
Waiting [press return]
SAC>
```

Typing a "q" and then return will exit the plot command and not plot the remainder of the files in memory.

More Plot Commands

Several other canned plot formats are available. PLOT1 plots each file along a common x axis but with a separate y axes. By default all files are placed on the same plot. Try this with the three files from the example above. PLOT2 is an overlay plot. Again all files are plotted together, this time using both a common x and a common y axis. PLOTPK uses a format similiar to PLOT1. It lets you use the cursor to blow up parts of the plot, determine values of selected data points, pick phase arrival times, etc.

Display Options

By default, all SAC plots are self-scaling. SAC determines what limits to use for the x and y axes. If you want to set these limits yourself, you may do so using the XLIM and YLIM commands. If you wish, you may also change the location of annotated axes, change the linestyle, select a symbol to be plotted at each data point, create titles and labels, make logarithmic plots, change the size and type of text, and control a number of other even more exotic aspects of the plot. These commands are part of the Graphics Environment Module, and are defined in links frmon the Command_Reference Manual.

Analysis Capabilities

Fortunately SAC does a lot more than just reading, writing, and plotting data files! Some of SAC's analysis capabilities are briefly discussed below.

Filtering

FFT and IFFT take the Fourier and inverse Fourier transform time series data.

- LOWPASS
- HIGHPASS
- BANDPASS
- BANDREJ

are a set of Infinite Impulse Response (IIR) filters. You may choose from Butterworth, Bessel, and Chebyshev Type I and II filters.

- WIENER applies an adaptive Wiener filter.
- FIR applies a Finite Impulse Response filter.
- DECIMATE applies an anti-aliasing lowpass filter as it desamples data.
- UNWRAP computes a spectral amplitude and an unwrapped phase.

Unary and Binary Operations

You may add a constant to each data point in a file using the ADD commands This is called a unary operation in SAC.

- ADD Add a constant
- SUB Subtract a constant
- MUL Multiply by a constant
- DIV Divide by a constant
- SQR Square each data point
- SQRT Take the Square Root of cach data point
- EXP Take the exponential of each data point
- LOG Take the logarithm of each data point

You may also add two data files together using the ADDF command. This is called a binary operation in SAC. Other binary commands include

- ADDF Add one file to another

- SUBF Subtract one file from another
- MULF Multiple one file by another
- DIVF Divide one file by another
- MERGE Merge two different files

Correcting Signals

There are a number of commands available to correct or modify seismic signals.

- RQ removes the seismic Q factor from spectral data.
- RTREND and RMEAN remove the linear trend and the mean, respectively, from time series data.
- TAPER applies a symmetric taper to each end of the data.
- ROTATE rotates a pair of data components through an specified angle in the plane of the components.

Phase Picking

- APK applies an automatic event picking algorithm to seismic data. Output can be written to a HYPO formatted disk file or to a more general alphanumeric pick file.
- PLOTPK can also be used to pick and write phase information into these files.

Summary

This is only a partial list of SAC's analysis capabilities. The complete list is given and each command is described fully in command_reference. If you have ideas on commands or features that you think are missing, send an e-mail message to the sac-help mailing list at sac-help@iris.washington.edu.