2-3

Basic array data handling

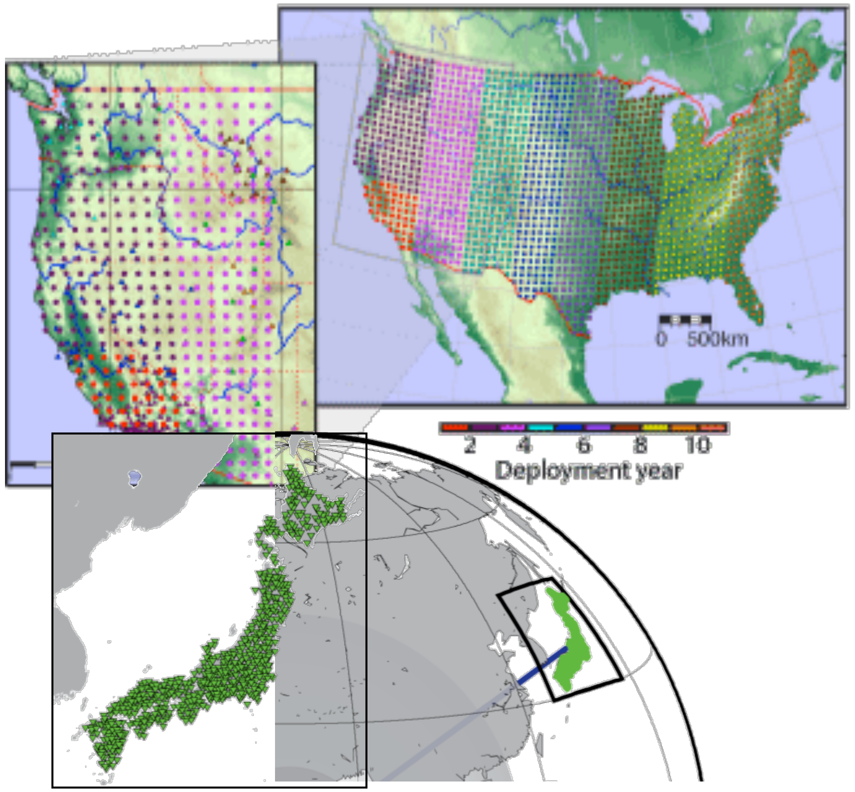
Array processing

- Array data (i.e., using data from multiple instruments) is increasingly being used in global seismology and is the norm in exploration seismology

- New array facilities like USArray and Japan's hi-net are opening new vistas of understanding

- Multiple instruments allow, among other things
  - Measurement of very subtle signals not visible in a single seismogram
  - Measurement of angles of arrival of waves – improves identification of phases

- SAC's capabilities for handling array data are fairly basic, but useful functionality is still available



The SSS subprocess

- Array processing and plotting is done in the SSS subprogram (start with command sss).
- The traces in current memory in SSS is called the *stack*
- All SSS operations operate on the files currently in the stack
- Any files in memory are automatically added to the stack when SSS is started
- Traces may be added to the stack from file, have properties updated, or be removed from the stack
- Traces to be added to the stack should have their station-event geometry set correctly, and have the same sampling rate

Adding traces to the stack

- Traces are added from files with command addstack.
- Note, this can only add *one file at a time* so wildcards are not allowed

```
SAC/SSS> help addstack

 SUMMARY:
 Add a new file to the stack file list.

 SYNTAX:
 [A]DD[S]TACK filename [property ...]
```

- properties allow control of the stack. Some useful ones are:
  - delay *t* : add the file specified with a timeshift *t*
  - weight *w* : weight trace by *w* in the stack (useful for down-weighting noisy traces for example)
  - n or r : specify normal or flipped polarity

Changing traces in the stack

- Properties like delay and weight for traces currently in the stack can be modified with the changestack command.

```
SAC/SSS> help changestack

 SUMMARY:
 Change properties of files currently in the stack file list.

 SYNTAX:
 [C]HANGE[S]TACK filename|filenumber property {property}
```

- In this filename must be the name of a file currently in memory, or the number of the file (in order of addition)

Deleting traces from the stack

- Remove traces from the stack with the deletestack command.

```
SAC/SSS> help deletestack

 SUMMARY:
 Deletes one or more files from the stack file list.

 SYNTAX:
 [D]ELETE[S]TACK filename|filenumber {filename|filenumber...}
```

- The command zerostack empties the current stack entirely

Distance and time windows

- Once the stack is populated, in order to plot record sections or make stack calculations time and distance windows are required
- The distance window is set by default (to the span of the data in km), however the time window is not set
- These are set by the timewindow (tw) and distancewindow (dw) commands respectively

```
SAC/SSS> tw 1200 1400
```

- The dw command can be made to display a fixed width, and can use degrees instead of km, e.g.,

```
SAC/SSS> dw fixed 70 80 units degrees
```

Plotting record sections

- A record section of the current stack can be plotted with the plotrecordsection command.

```
SAC/SSS> help prs

 SUMMARY:
    Plots a record section of the files in the stack file list.

 SYNTAX:
    [P]LOT[R]ECORD[S]ECTION [ options ]
```

- This command has a great deal of functionality (including an interactive picking interface), however some basic options include:
  - labels *on/off/header* : control trace labelling
  - reference *on/off* : whether a zero line is drawn
  - size *s* : this defines the scale at which the traces are drawn
- Permanent plots can be produced using the sgf device, as before

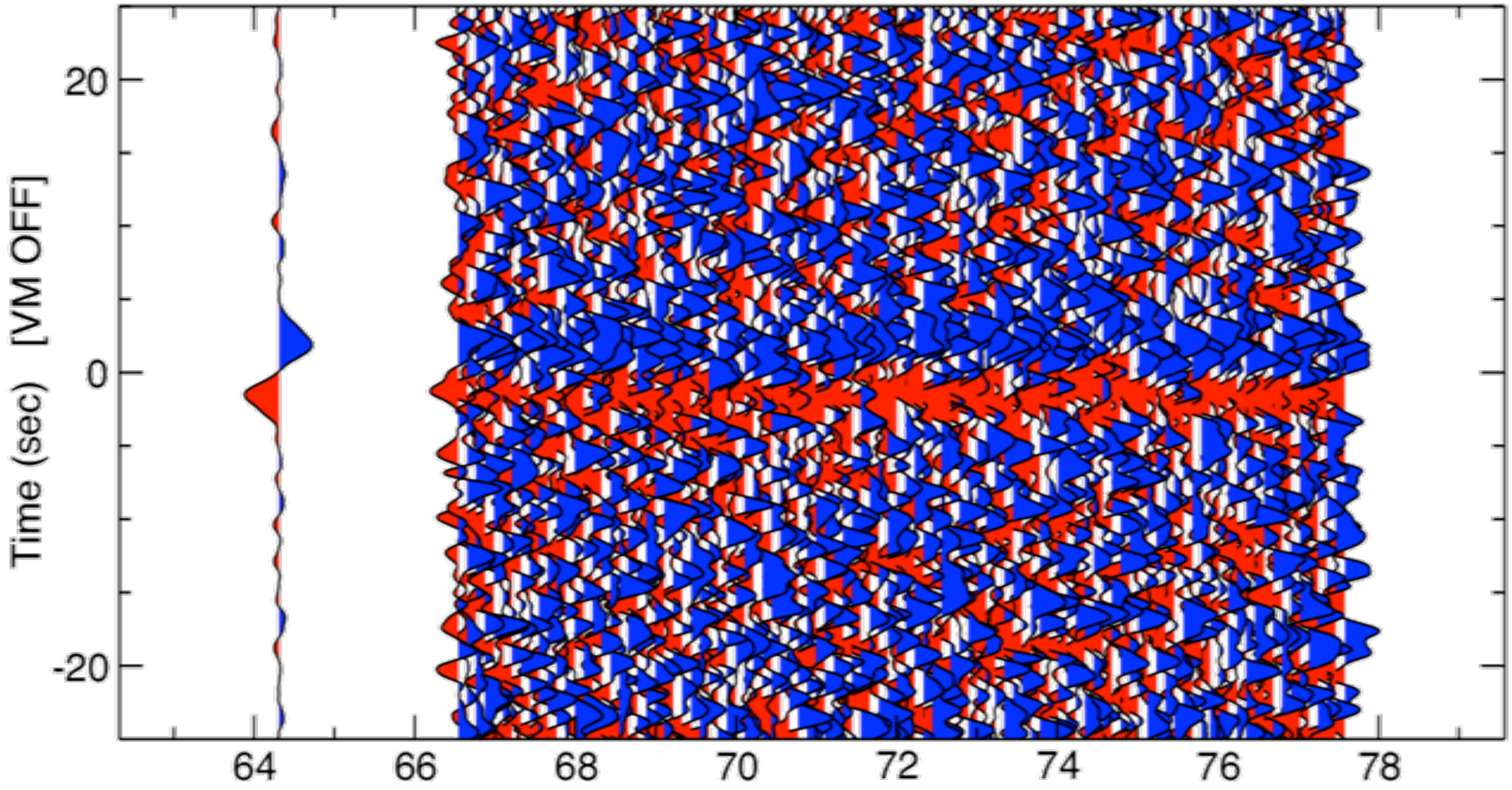
Stacking

- Stacking is a standard technique in seismology, and is used extensively in global and particularly exploration contexts

- Stacking is (at its simplest) merely summing many traces together

- The motivation for this is to enhance signal which coherent across the traces, and suppress noise which is not (constructive vs. destructive interference)

Stacking example



Generating stacks

- A linear sum of the current stack can be generated with the sumstack command.

```
SAC/SSS> help sumstack

 SUMMARY:
 Sums the files in the stack file list.

 SYNTAX:
    [S]UM[S]STACK  [[N]ORMALIZATION ON|OFF]
```

- This will plot a stacked trace, which can be written to disk with the command writestack

```
SAC/SSS> help writestack

 SUMMARY:
 Writes the stack summation to disk.

 SYNTAX:
    [W]RITE[S]TACK [filename]
```

- This trace is then a standard SAC trace