APPENDIX C
ESTIMATES OF AVERAGE SPECTRAL AMPLITUDES AT FOAKE SITES

David M. Boore
U.S. Geological Survey
Menlo Park, CA 94025
(415)-329-5616

Introduction

In my role as consultant to the panel of experts evaluating the equipment qualification work for the NRC, I was asked to estimate ground motions at selected FOAKE sites. This report presents my estimates and the method used to arrive at the estimates. After a brief description of the method, I present the results with a short description of particular considerations for each site, if needed. For clarity of presentation, tables giving the details of the estimates are gathered together in an appendix. Another appendix contains plots of the acceleration response spectra for each station used in the estimation process, with the average level from 3 to 8 Hz (the measure of ground motion used in the report) given by horizontal lines.

Method

The method for estimating the mean ground motion from a particular earthquake at a specified site required finding nearby strong motion recordings, computing the ground motion measure of interest, and correcting these recordings for differences in site geology and for differences in the distance from the sites to the earthquake. In addition, uncertainty bounds are computed that account for the distance between the reference site and the recording site.

In somewhat more detail, these steps are as follow:

1. Search strong-motion database for all recordings within a radius of 10 km.

2. Determine the distance from the reference site to each strong-motion station identified in step 1.

3. Pick one or several recordings from this set, depending on proximity to the reference site and similarity of site geology.

4. Compute the response spectra for each site, in most cases using uncorrected

acceleration data with a least-square fitted straight line removed (no instrument correction or high- and low-cut filtering was done).

5. For each horizontal component, compute the average acceleration response spectra (S_a) between 3 and 8 Hz, according to

$$S_a = \frac{1}{5} \int_3^8 (2\pi f)^2 \, S_d \, df,$$

where S_d is the relative displacement of a 5 percent damped oscillator with natural frequency f. Find the arithmetic average of S_a for each horizontal component. Plots of all spectra used are given in Appendix C.1.

6. Determine the shortest distance from each strong-motion recording station and the reference site to the surface projection of the rupture surface (the boundaries of the rupture surface were extracted from published studies of each earthquake, using my judgment as to the best estimate of the rupture surface).

7. Assign a shear-wave velocity to each station and to the reference site. This is the time-averaged velocity over the first 30 m of depth, computed as 30 m divided by the travel time from the surface to 30 m. In some cases velocities from a nearby borehole were available, but in most cases the velocities were estimated from boreholes in geologic materials similar to those under the site; Tom Fumal, who has had years of experience in making these assessments, helped me in assigning the velocities.

8. For each recording to be used in the estimation, correct for differences in site response and distance to the earthquake by multiplying average spectral acceleration by the correction factor

$$psv(m, d_{ref}, \upsilon_{ref})/psv(m, d_{sta}, \upsilon_{sta}),$$

where psv is the response spectrum predicted from the regression equations of Boore, Joyner, and Fumal (1993 and 1994), and d_{ref}, d_{sta} and υ_{sta} are the earthquake-to-site distances and average sub-site shear velocity for the reference and recording site, respectively (I have included in Appendix C.3 a listing of the Fortran program used in the analysis).

9. Compute the geometric mean of the corrected estimates (i.e., average the logs of the corrected estimates and raise 10 to this average of the logs).

10. Approximate the plus and minus one sigma uncertainty ranges by multiplying and dividing the averaged corrected spectral estimate by the factor

$$10^{0.182\sqrt{1 + \frac{1}{N}\left(1 - \exp - \sqrt{0.6\Delta}\right)}} .$$

The basis for this equation is given in the next section.

Uncertainty in Estimates

Analysis of scatter about regression curves yields the uncertainty in the prediction of any one value of ground motion. The analyses that I have been associated with have regressed on the common log of the ground motion, and all of my discussion here will refer to logs to the base 10. We found from our regression work that the within-earthquake $\sigma_{\log pga}$ was 0.188 and 0.182 for the larger and random horizontal peak acceleration, respectively, for earthquakes with magnitudes between 6.0 and 6.9. (I am assuming that the uncertainty of the 3-to-8 Hz averaged spectral acceleration will be similar to that for the peak acceleration.) In the application in this report, nearby records provide an estimate of the actual mean motion at the reference site, but because there is a spatial variation in ground motion, the reference site motion will be uncertain even if the true value of the mean of the motions within a small region surrounding the site has been determined. Clearly, this additional uncertainty reduces to zero if the recording site is at the exact location of the reference site. On the other hand, for a great enough separation distance, the spatial correlation reduces to zero and the additional uncertainty reaches that for an individual observation. This discussion suggests the following equation for the variance of the estimated motion at the reference site (because ground motions are well-approximated by a lognormal distribution, the standard deviations in the following discussion are those of the log of the ground motion; uncertainty ranges for the ground motion are given by respectively multiplying and dividing the ground motion by 10 raised to a power equal to the standard deviation):

$$\sigma_{ref}^2 = \sigma_{sta}^2 \left(1 + \frac{1}{N}\right) F(\Delta)^2,$$

where σ_{sta} is the standard deviation of an individual observation (e.g., 0.182 for the random horizontal component of peak acceleration), and N is the number of recordings used in the average (the term in N accounts for the uncertainty in the estimate of the mean motion). $F(\Delta)$ is a function that accounts for the spatial correlation of the motion, where Δ is the average separation between recording station and reference site; F takes on values of 0.0 and 1.0 for $\Delta = 0$ and $\Delta = \infty$, respectively.

I estimated $F(\Delta)$ by studying larger peak horizontal accelerations from the 1994 Northridge mainshock (the most complete data set available to me), supplemented by studies of spatial variability in small arrays (Abrahamson and Sykora, 1993), the SMART 1 array in Taiwan (Abrahamson, written commun, 1995), and local regions in the 1971 San Fernando earthquake (McCann and Boore, 1983). The analysis for the Northridge data followed these steps:

1. Compute Δ for all pairs of stations, keeping only those for which the separation was less than 10 km (over 600 pairs).

2. For each pair, compute the difference of the larger peak horizontal acceleration after correcting for differences in distance from the station to the earthquake (the distance attenuation used for this correction was derived in the course of the analysis as corrections to the average attenuation of Boore, Joyner, and Fumal, 1993).

3. Divide the range of Δ into bins such that 15 station pairs are within each bin. This was done so that a reasonable estimate of the variance of the residuals could be obtained for each bin.

4. Compute the standard deviation of the residuals within each Δ bin.

5. Plot the standard deviations against the median distance for each bin, and fit a function to this plot, guided also by the Abrahamson and Boore and McCann studies. The results are shown in Figure C.1. This procedure yielded the following equation for $F(\Delta)$:

$$F = (1 - \exp - \sqrt{0.6\Delta}).$$

Listings of the computer programs used in the analysis are included in Appendix C.3.

I am aware that a whole computational structure ("kriging") has been built up to deal with spatial estimation problems (e.g., Journel, 1989). I did not have time to learn about this structure, so I devised a simplified procedure that should give reasonable results (I have presented the uncertainty ranges to only one decimal place to emphasize the imprecision of the estimates).

Results

The results are summarized in Table C.1; details are given in tables gathered together in Appendix C.2. The detailed tables contain all the information used in the processing. In addition to the corrected values summarized in Table C.1, the Appendix tables give values uncorrected for distance and site differences. Although not annotated, the entries in the tables should be self explanatory.

There were many recordings for the Whittier Narrows earthquake, including a large number from the USC array. I have these data, but I have not yet entered them into my database. In view of the proximity of the Commerce Refuse reference site to the Bulk Mail facility (0.8 km) and the limited time available to me, I did not do a search for nearby stations that recorded the Whittier Narrows earthquake; I simply used the recording at the Bulk Mail facility. According to Ed Etheridge (personal communication, 1995) and the notes in the station files in the strong motion lab at the U.S.G.S., the Bulk Mail site is located within a very large warehouse with a slab foundation of considerable horizontal extent. It is very likely that the motions at the recording instrument were reduced by the slab, particularly for the higher frequencies of interest to the FOAKE study. This will mean that the motions estimated from that record will be conservative for purposes of FOAKE.

Note that for the Northridge earthquake two estimates are given for the Placerita cogen reference site and three for the Sylmar Converter Station reference site. For Placerita the nearest site is at Newhall ($\Delta = 3.5$ km), but there were a number of additional sites at $\Delta \approx 7.5$ km. The Newhall site is not so close that it is obvious that it alone should be used in the estimate. Note that the two estimates of the median motions are well within the uncertainty ranges.

For the Sylmar Converter Station, the VG1-6 (Valve Group 1-6) record was obtained in the basement of the terminal building containing the equipment of interest. I assume that the reference site coincides with that building. Logically, the VG1-6 record should be used solely for the estimate of the motions of equipment in that structure. On the other hand, the VG1-6 spectrum is quite different from the nearby free-field recording near Valve Group 7 (VG7FF). I wonder whether the VG1-6 record is contaminated by building response and embedment depth effects. (The differences could, of course, also be due to variations in local geology or to the soil failure that was observed in the vicinity). I was instructed by the Panel to estimate free-field motions, which I have attempted to do. Modifications of the motion due to structural effects are the responsibility of others more qualified than I to do so. I do not have the expertise to evaluate the possible modifications of the VG1-6 record due to embedment and structure. If the modifications are small, then I would recommend using estimate 1 for equipment in the terminal building (and I note that during our meeting on March 29, 1995, the Panel instructed me to use only the VG1-6 record). In view of possible structural effects at VG1-6, for the Sylmar Converter Station reference site I think it might be most appropriate to use my second estimate, which combines the VG1-6 and VG7FF. For completeness, Table 1 also contains an estimate from VG7FF alone.

I am assuming that most of the equipment at the Sylmar Converter Station is in the terminal building, but I do recall that we walked through Valve Group 7. If there is equipment in that structure, it should be considered a separate reference site. For completeness, I include in the summary table and in Appendix C.2 estimates for the Valve Group 7 building, using the average of the free field and floor spectra.

References

Abrahamson, N. and D. Sykora (1993). Variation of ground motions across individual sites, *Fourth DOE Natural Phenomena Hazards Mitigation Conference*, 1993.

Boore, D. M., W. B. Joyner, and T. E. Fumal (1993). Estimation of response spectra and peak accelerations from western North American earthquakes: An interim report, *U. S. Geol. Surv. Open-File Rept. 93-509*, 72 pp.

Boore, D. M., W. B. Joyner, and T. E. Fumal (1994). Estimation of response spectra and peak accelerations from western North American earthquakes: An interim report, Part 2 *U. S. Geol. Surv. Open-File Rept. 94-127*, 40 pp.

Journel, A. G. (1989). Fundamentals of geostatistics in five lessons, *American Geophysical Union Short Course in Geology: Volume 8*, 40 pp.

McCann, Jr., M. W. and D. M. Boore (1983). Variability in ground motions: root mean square acceleration and peak acceleration for the 1971 San Fernando, California, earthquake, *Bull. Seism. Soc. Am.* **73**, 615-632.

Table C.1
Summary of Results - SA Averaged from 3 to 8 Hz, in g.

Site	FOAKE	Boore		Comments
Altwind, NPS86	1.39	1.23	(0.8, 1.8)	
Buckwind, NPS86	1.39	1.37	(1.0, 1.9)	
Devers, NPS86	1.33	1.48	(1.1, 2.1)	
Garnet Sub, NPS86	1.39	1.16	(0.8, 1.7)	
Renwind, NPS86	1.39	1.28	(0.8, 2.0)	
Sanwind, NPS86	1.39	1.47	(1.0, 2.2)	
Terrawind, NPS86	1.39	1.35	(0.9, 1.9)	
Venwind, NPS86	1.39	1.53	(1.0, 2.3)	
Whitewater, NPS86	1.39	1.45	(0.9, 2.2)	
Commerce Refuse, W87	1.03	1.11	(0.8, 1.5)	
SC Telephone, LP89	1.30	1.10	(0.7, 1.7)	
SC Water, LP89	1.26	1.18	(0.8, 1.8)	
Soquel Water, LP89	1.30	1.47	(1.0, 2.1)	
UCSC cogen, LP89	1.23	1.30	(1.2, 1.4)	
Centerville, P92	0.90	1.00	(0.9, 1.1)	
PALCO cogen, P92	0.93	0.93	(0.6, 1.4)	
Financial Center, NR94	1.22	1.52	(1.0, 2.3)	
Olive View cogen, NR94	1.20	1.18	(1.0, 1.4)	
Placerita cogen, NR94: est. 1	1.33	1.26	(0.8, 2.0)	Using closest station
Placerita cogen, NR94: est. 2	1.33	1.10	(0.7, 1.6)	Using 4 stations
Rinaldi, NR94	1.20	1.33	(1.1, 1.6)	
Sylmar CS, NR94: est. 1	1.20	0.62	(0.6, 0.6)	Using VG1-6
Sylmar CS, NR94: est. 2	1.20	0.82	(0.7, 0.9)	Using VG1-6 & VG7 FF
Sylmar CS, NR94: est. 3	1.20	1.09	(0.9, 1.3)	Using VG7 FF
Sylmar CS, VG7, NR94	1.20	1.05	(1.0, 1.1)	Using VG7 FF & Bldg

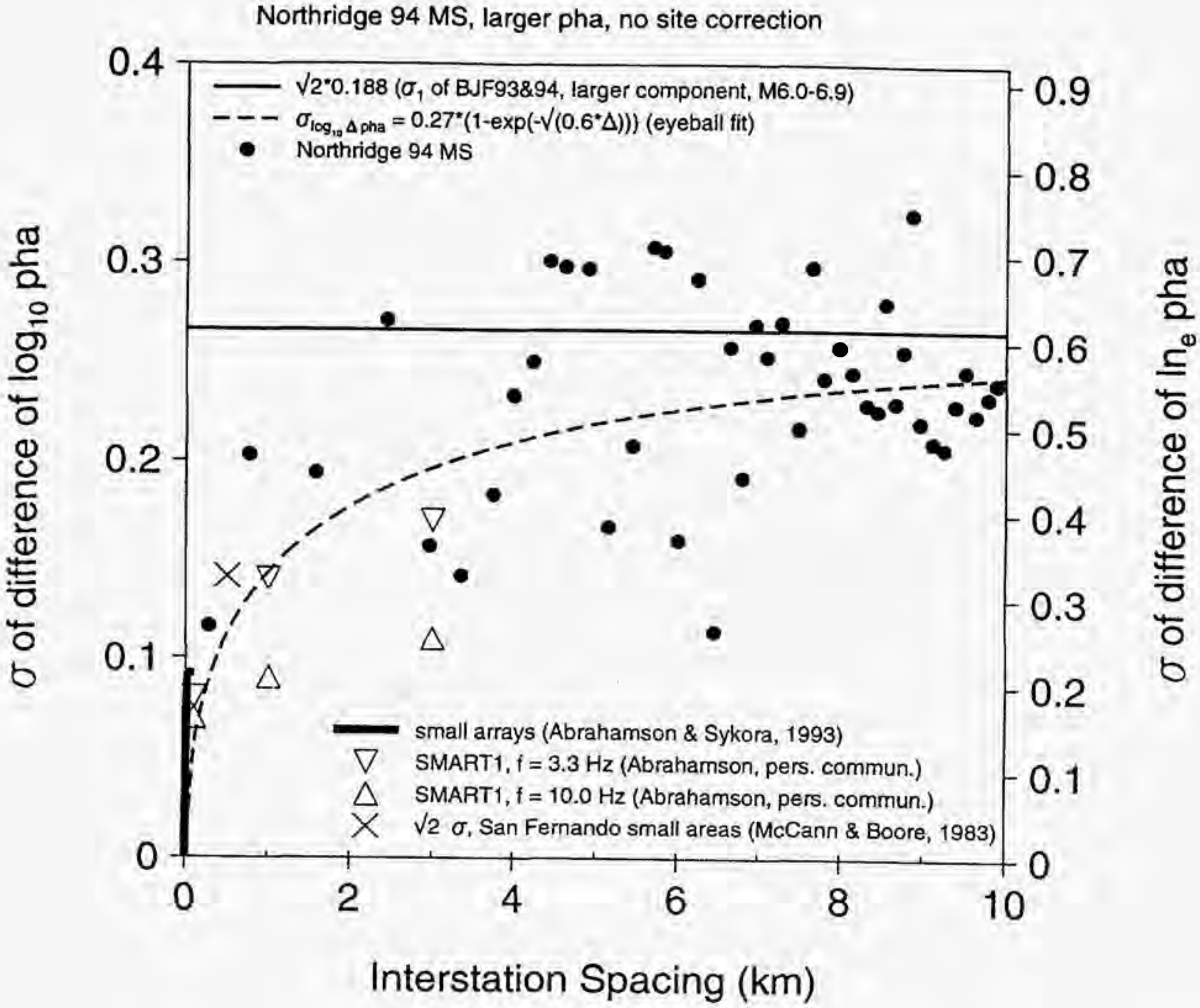
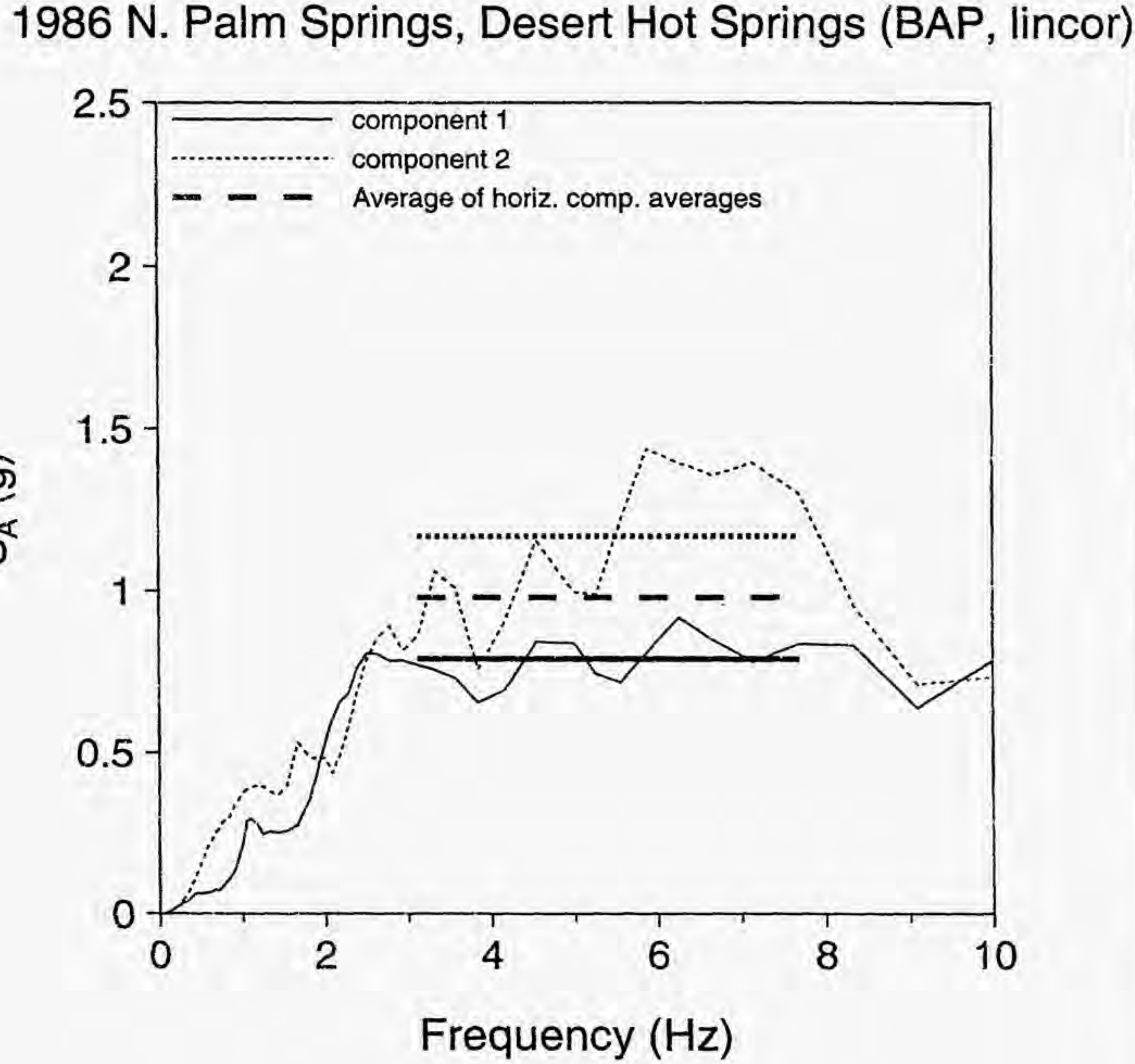


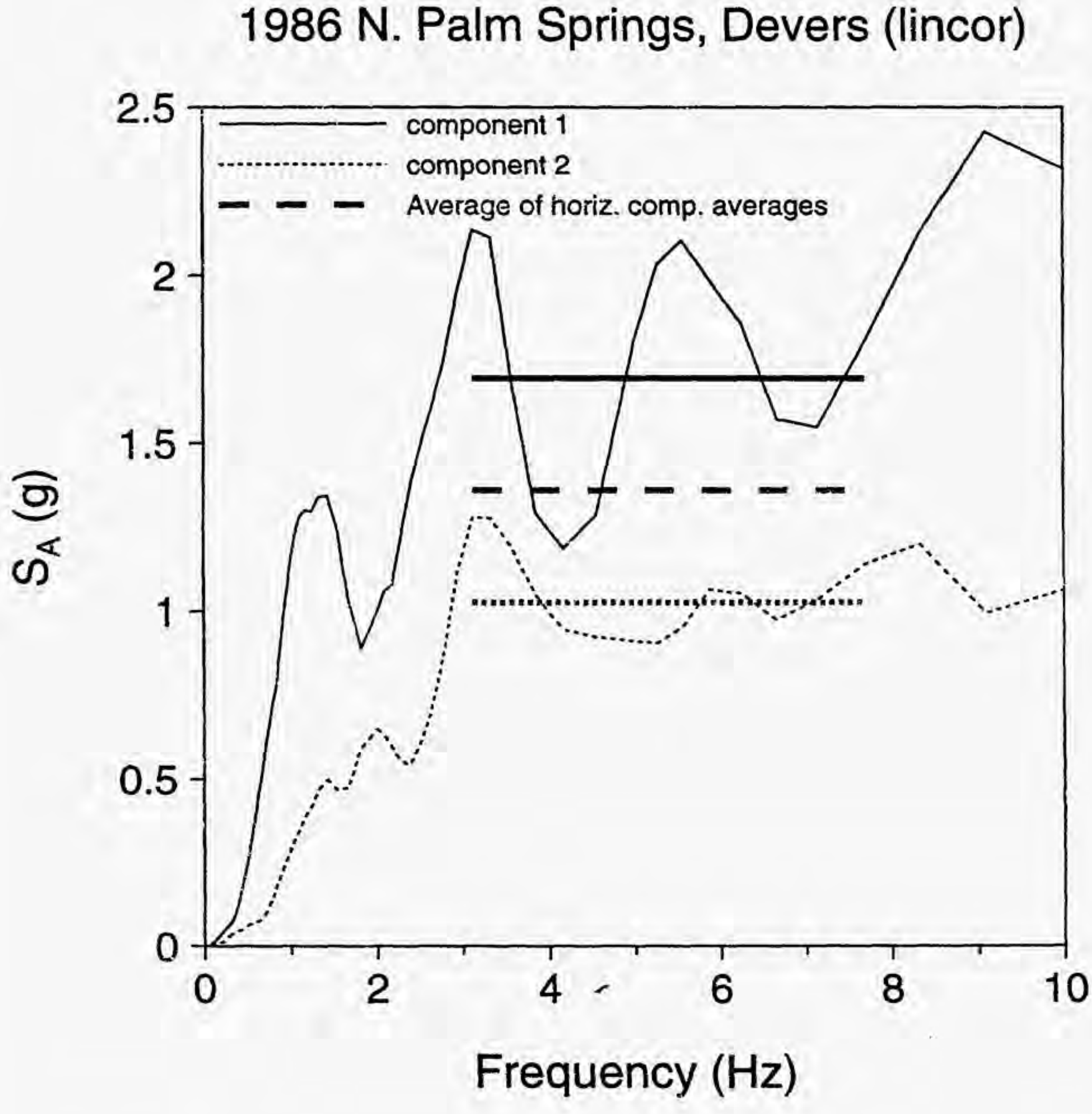
Figure C.1 - Standard deviation of difference of log of the larger peak horizontal acceleration as a function of interstation spacing. This provides the function F(Δ) referred to in the text. As an example of use, assume that a recording of 0.6 g was obtained 2 km from a reference site, and that the parameter of interest is larger peak horizontal acceleration (I assume that F(Δ) is independent of whether larger or random motions are being estimated--- those differences are accounted for in the leading term; see the equation in the text). If both the recording and reference sites are on the same geology and are both at the same distances from the earthquake, then the best estimate of the motion at the reference site is 0.6 g, with an uncertainty range given by $0.6/10^{0.18} = 0.4$ and $0.6 \times 10^{0.18} = 0.9$; I would report this as 0.6 (0.4, 0.9). (The factor 0.18 came from the value of the dashed curve at an interstation spacing of 2 km.)

APPENDIX C.1

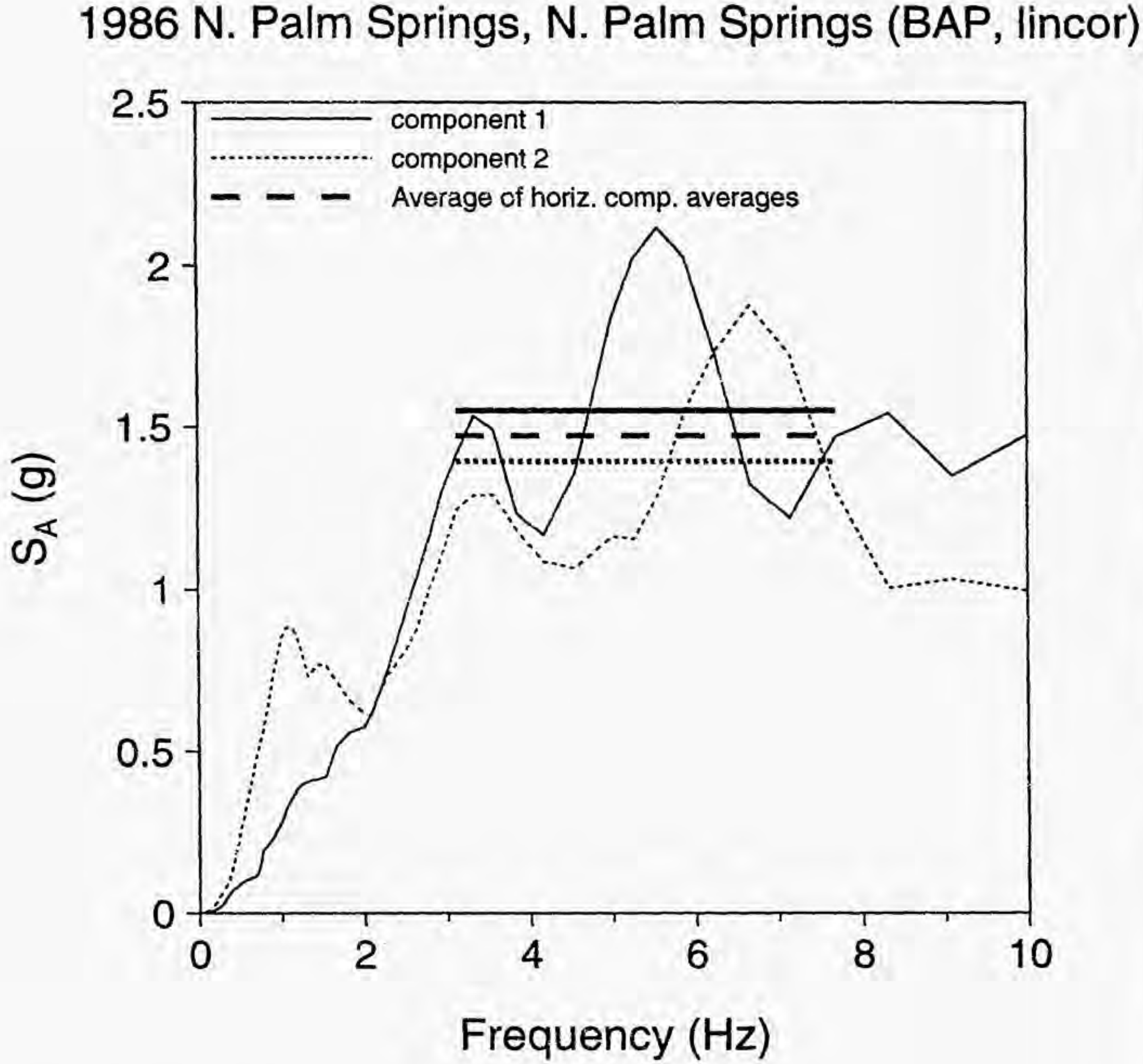
FIGURES OF RESPONSE SPECTRA



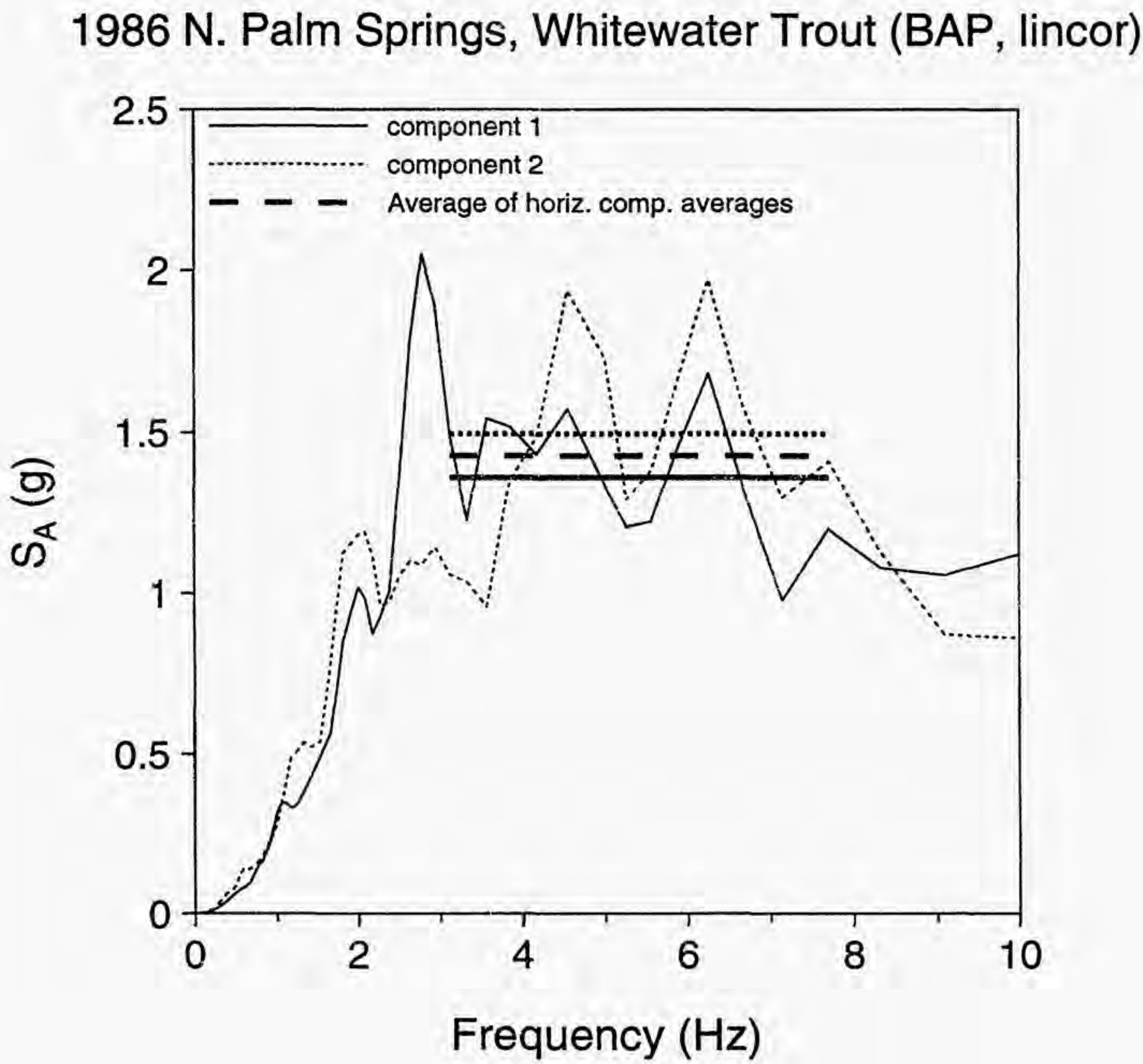
Apr 11, 1995 1:36:06 pm dsp_bap.gra



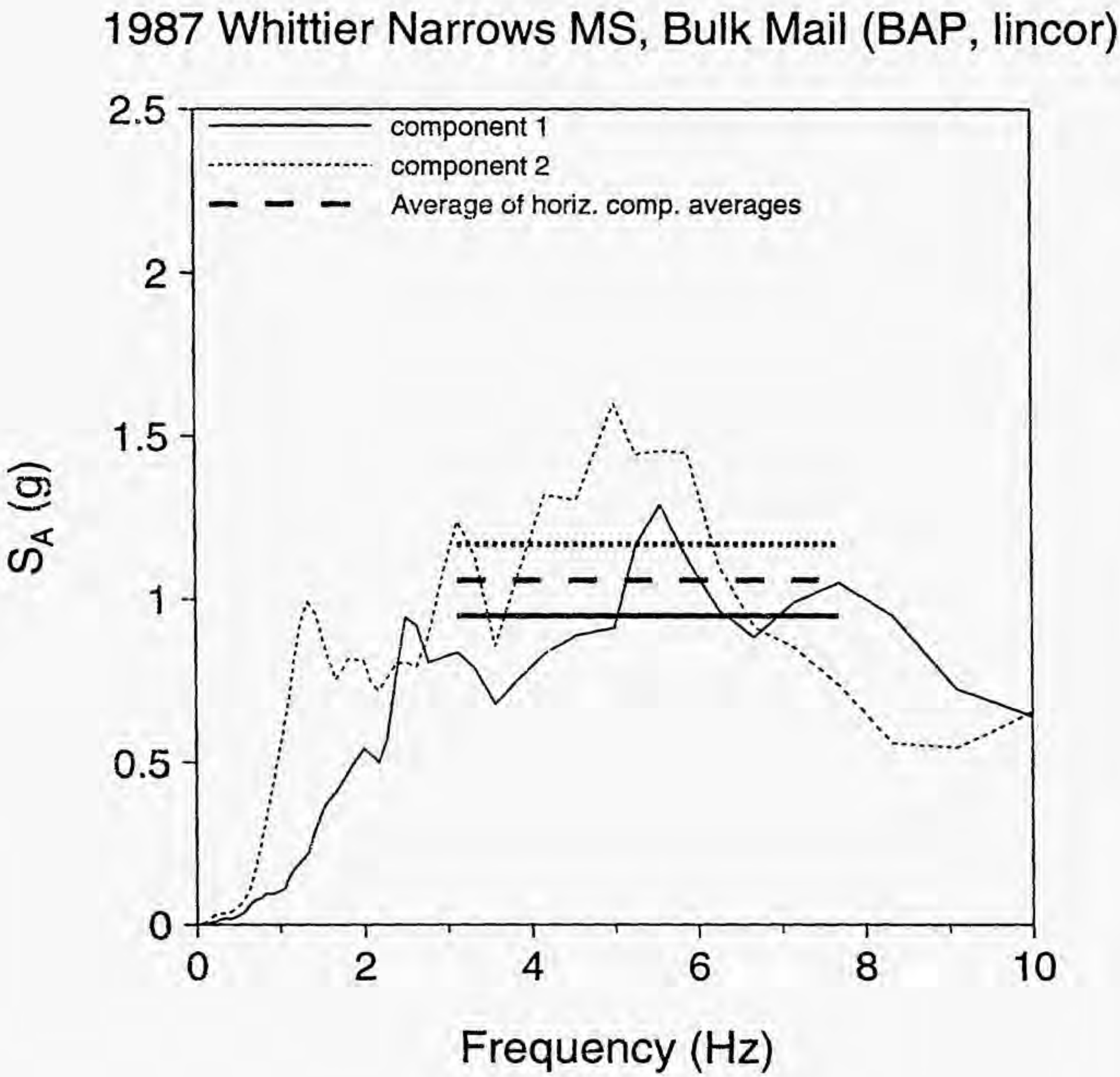
Apr 11, 1995 1:29:00 pm devers.gra



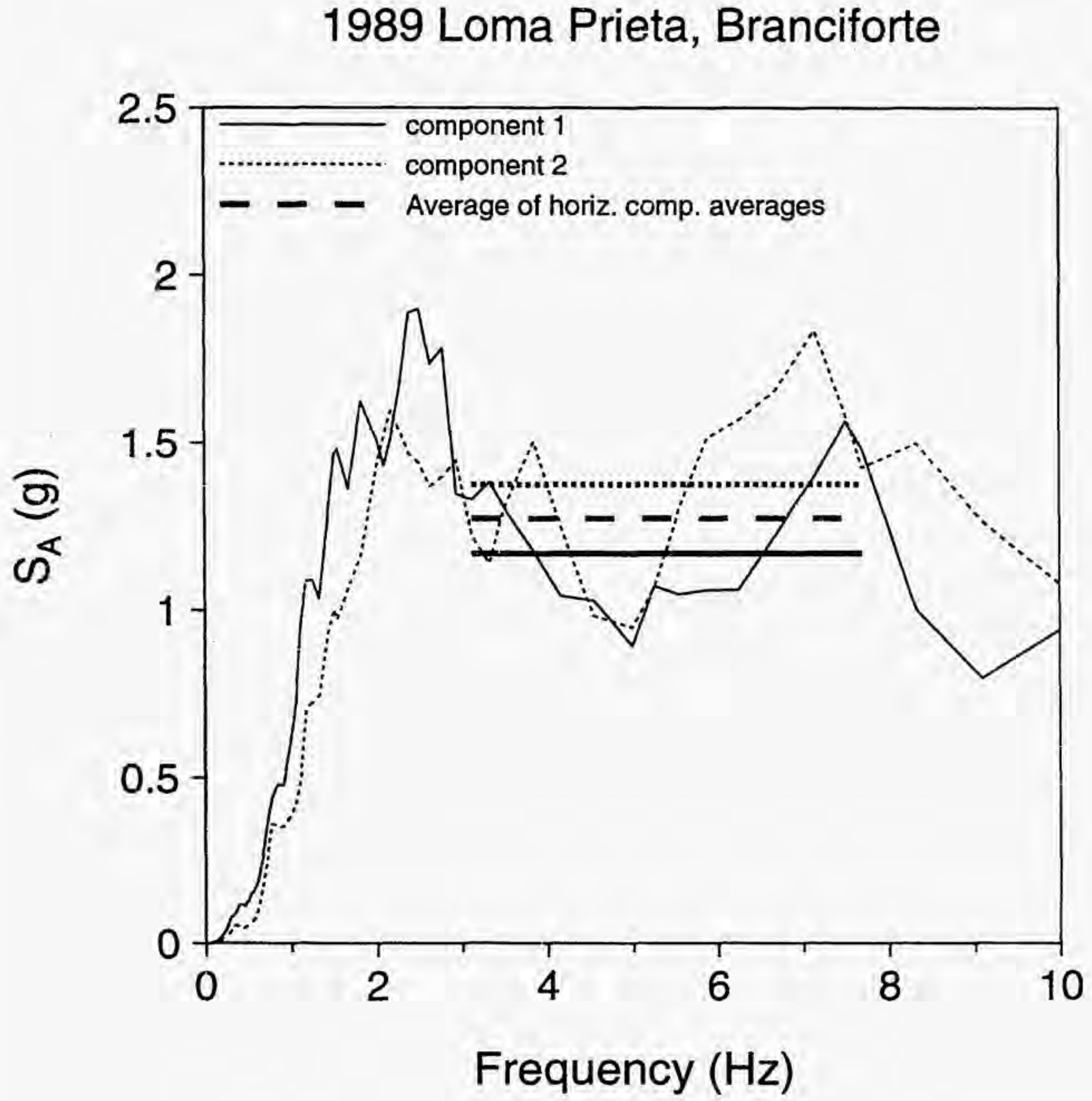
Apr 11, 1995 1:34:05 pm nps_bap.gra



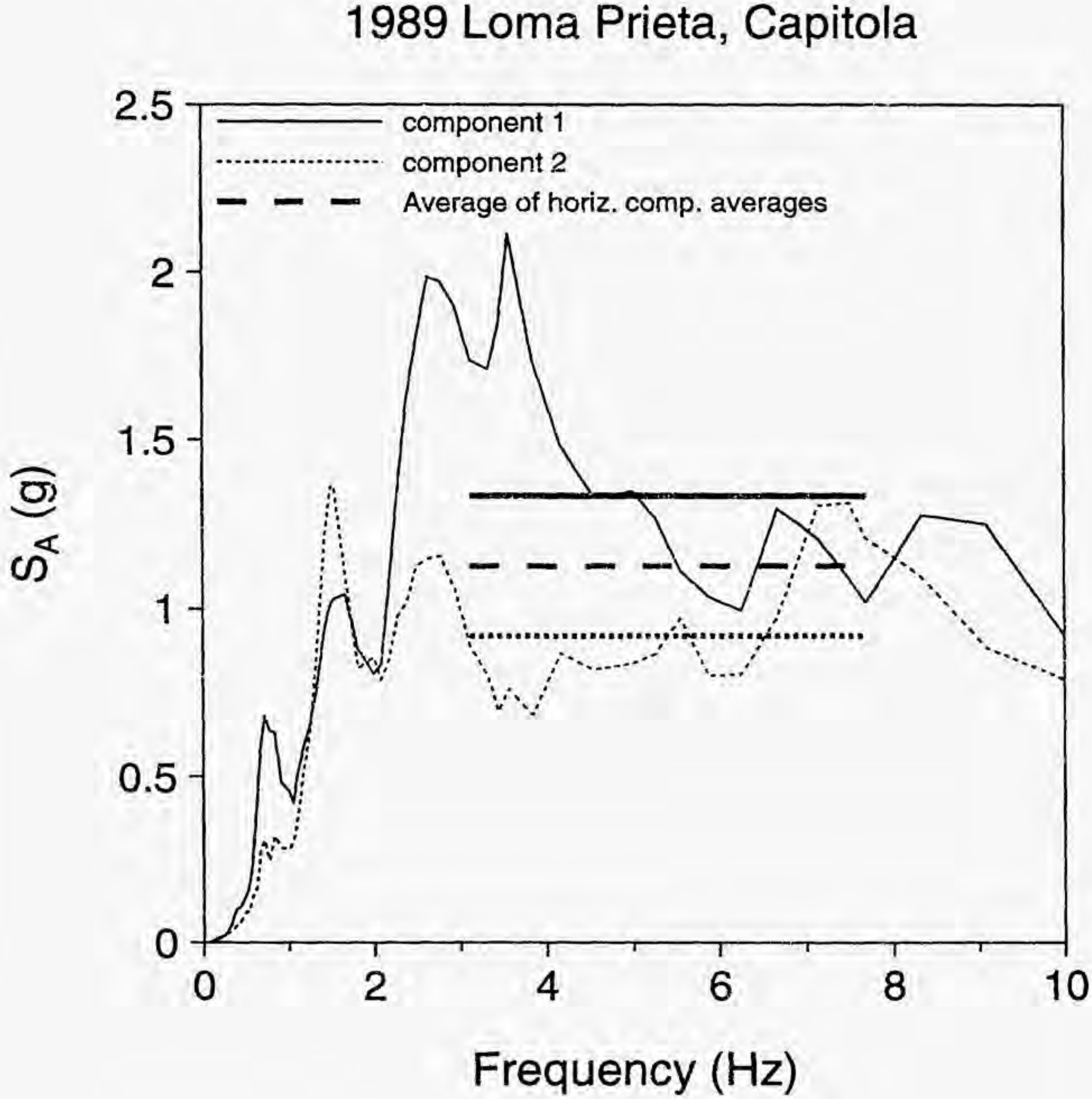
Apr 11, 1995 1:27:01 pm wwt_bap.gra



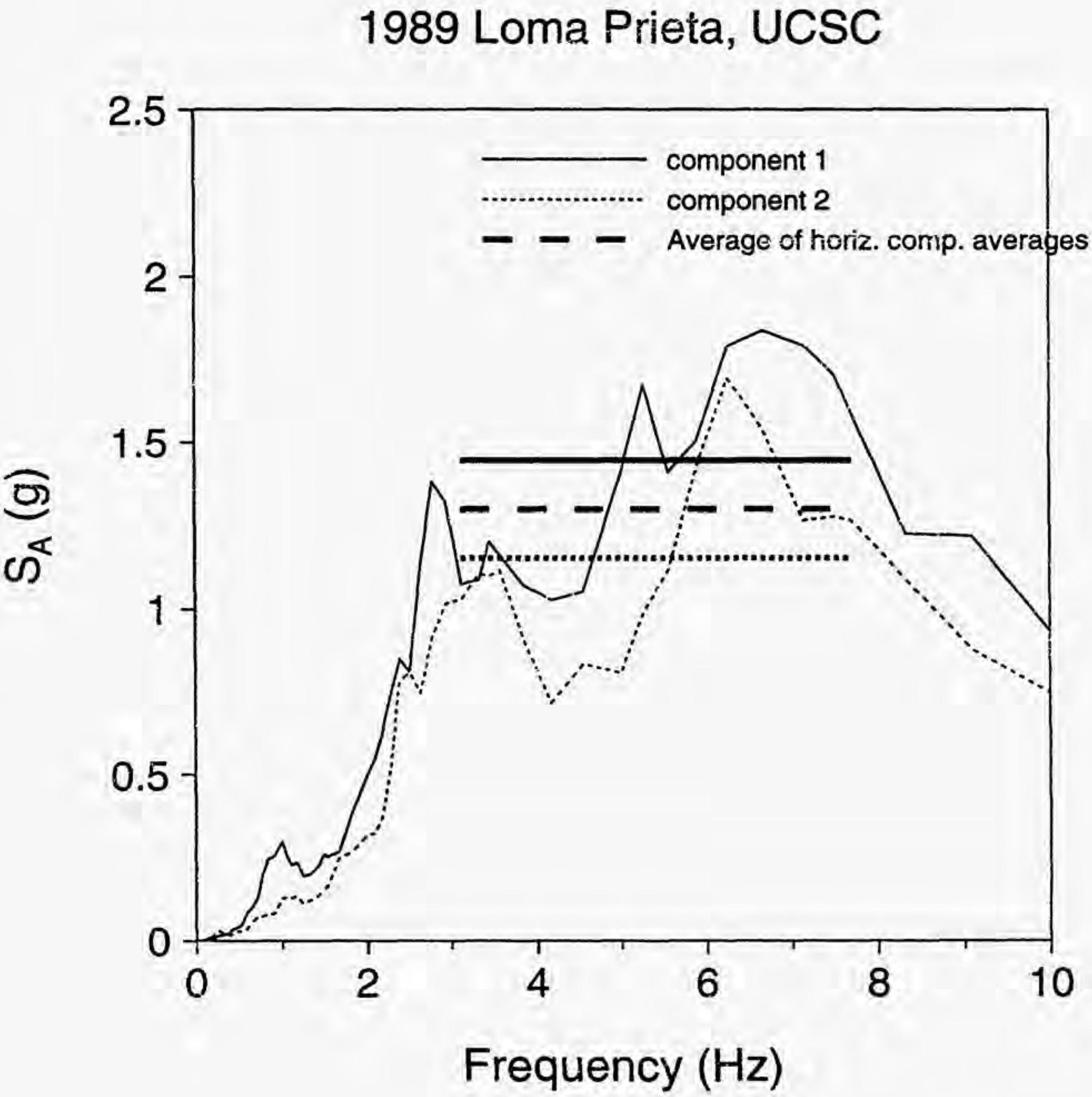
Apr 11, 1995 3:14:22 pm bulk_bap.gra



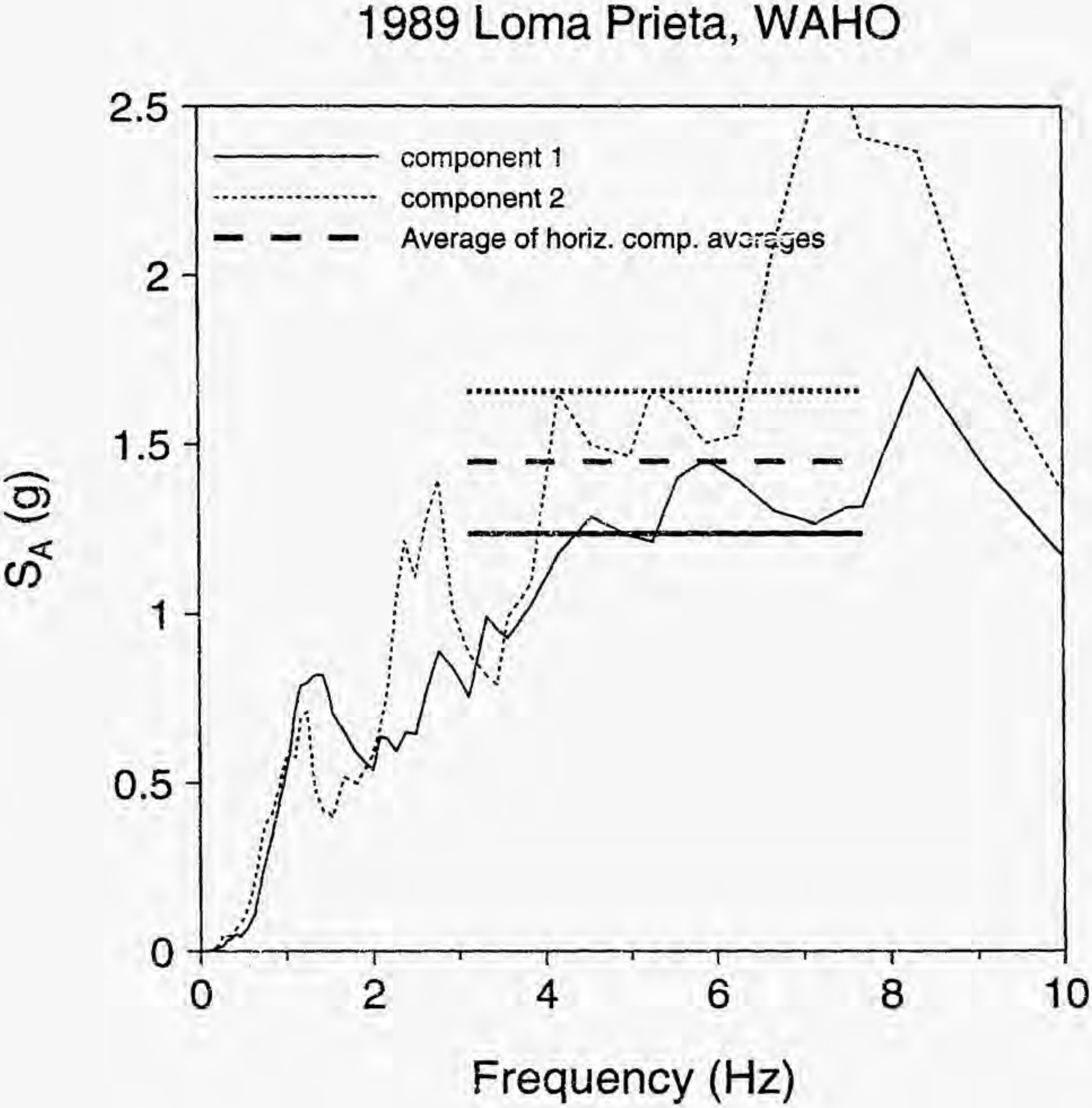
Apr 11, 1995 3:37:38 pm bran.gra



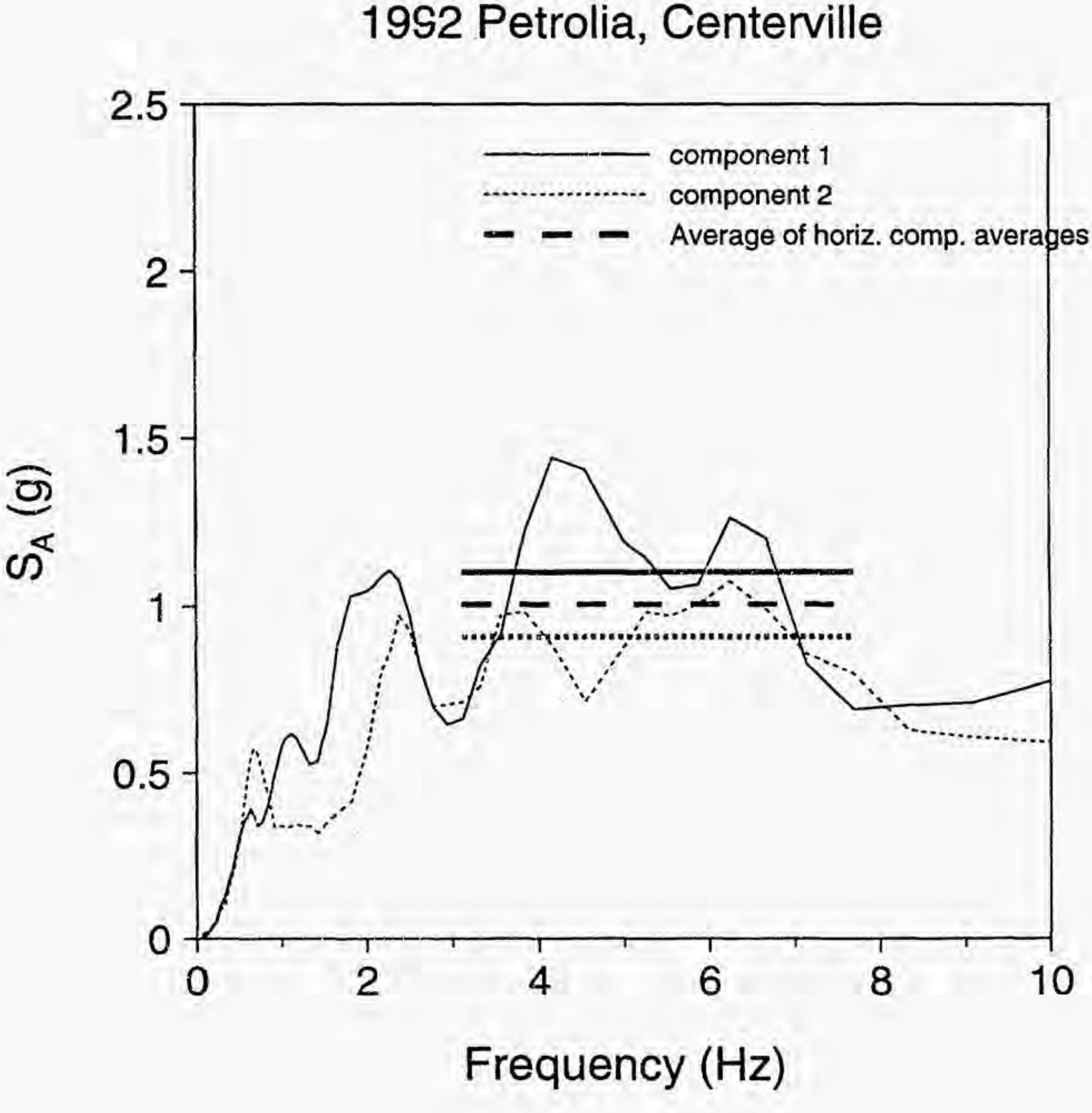
Apr 11, 1995 3:37:51 pm capitola.gra



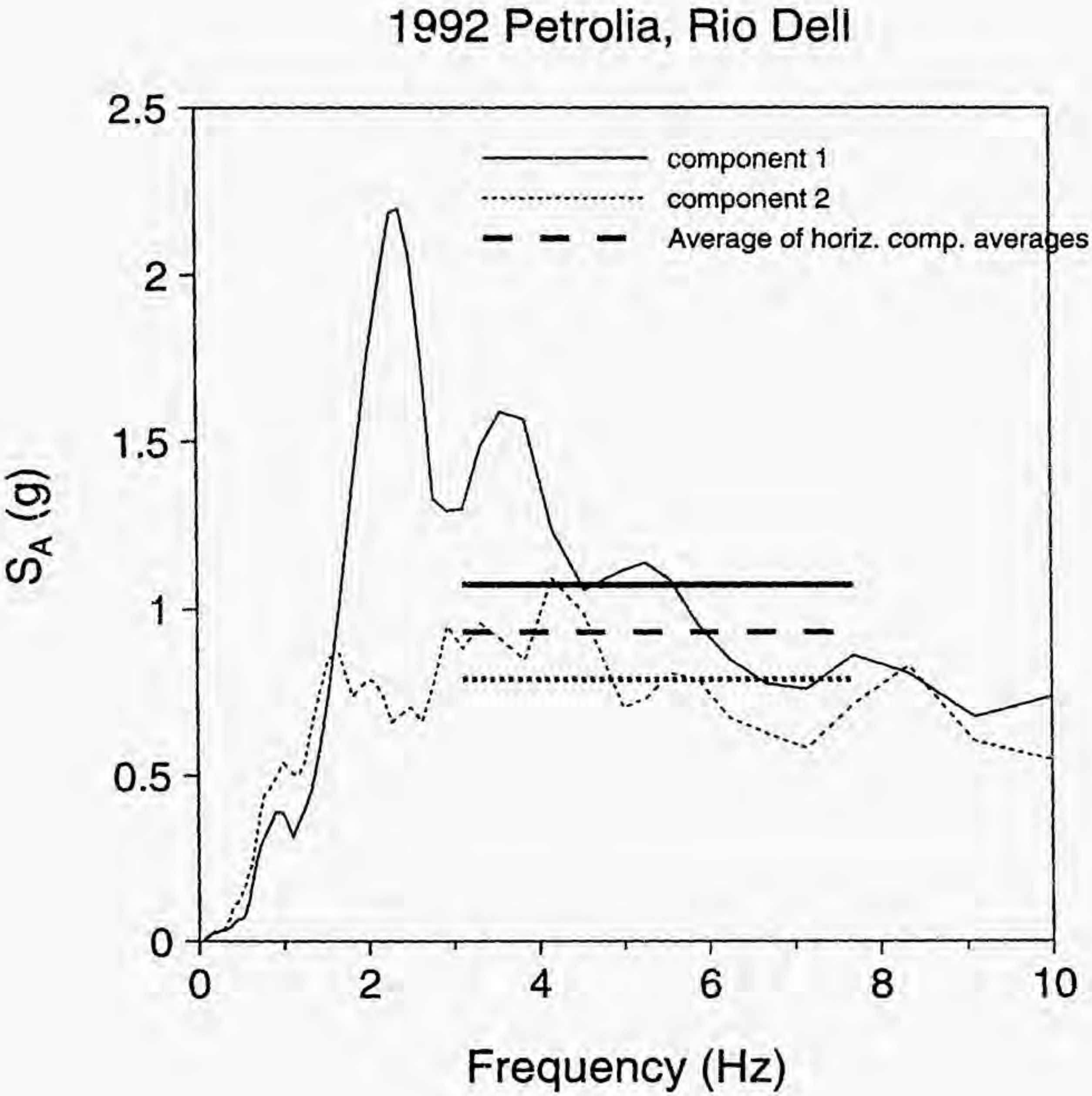
Apr 11, 1995 3:38:03 pm ucsc.gra



Apr 11, 1995 3:38:17 pm waho.gra

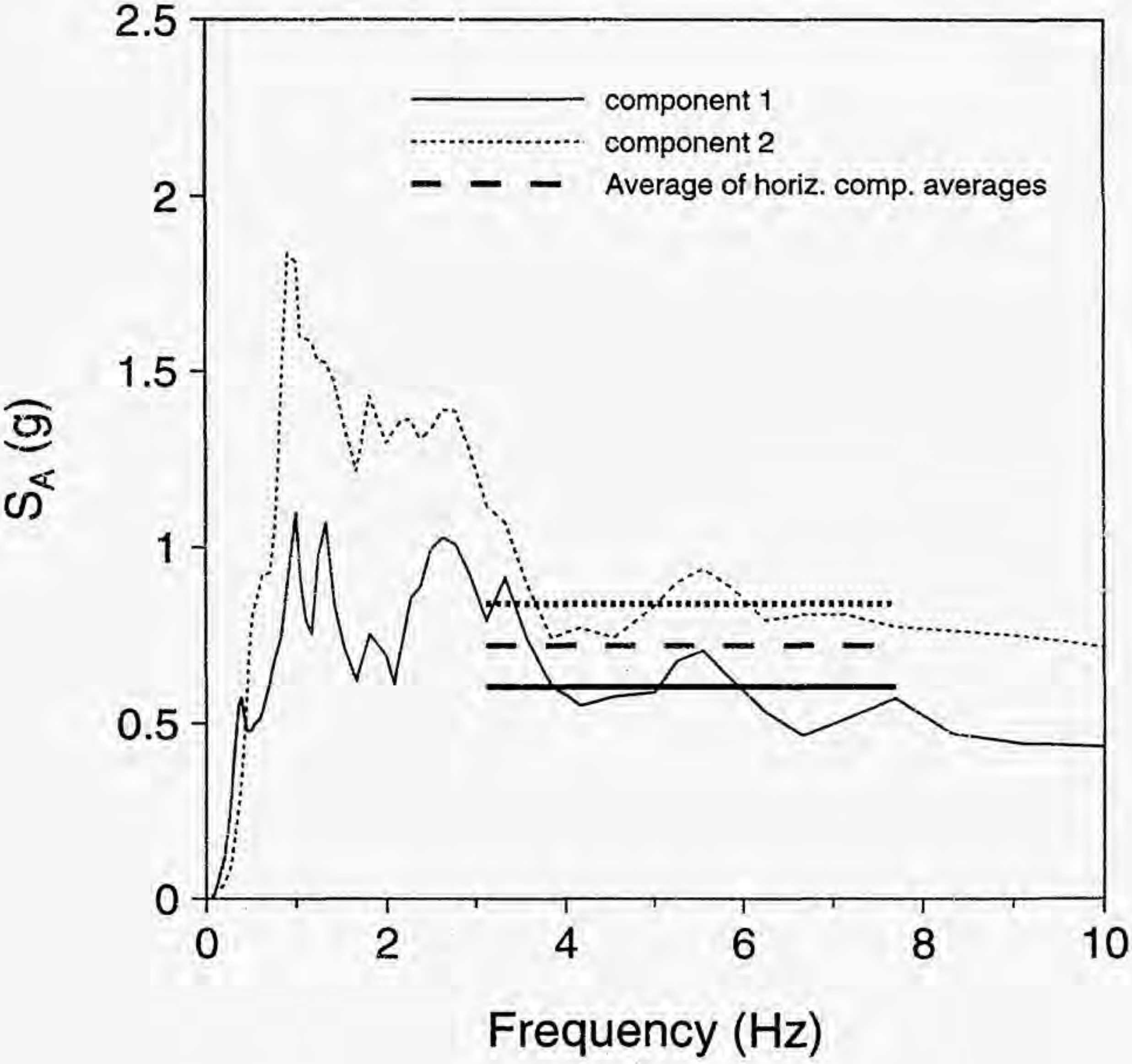


Apr 11, 1995 3:30:21 pm centerv.gra

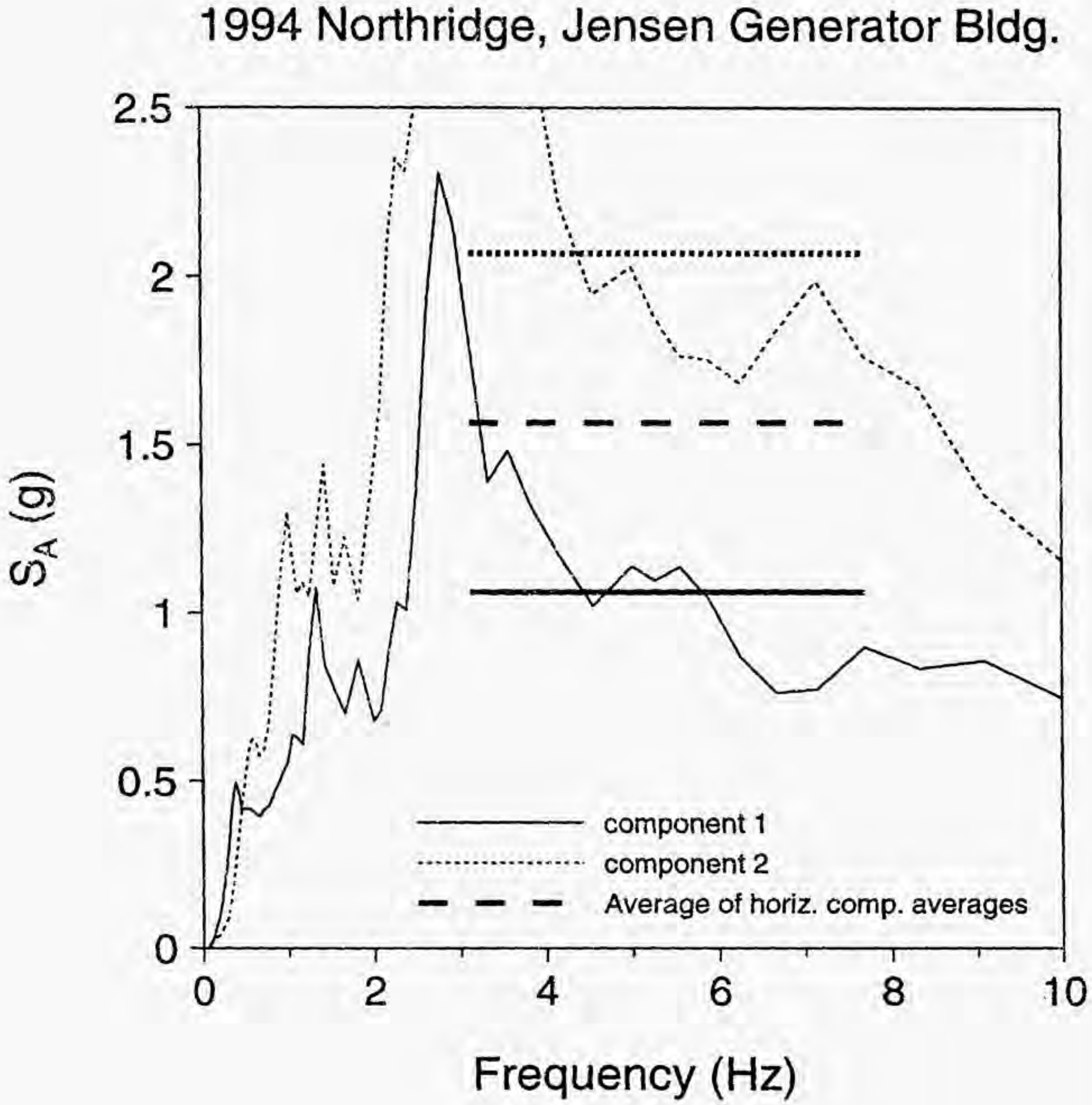


Apr 11, 1995 3:31:02 pm riodell.gra

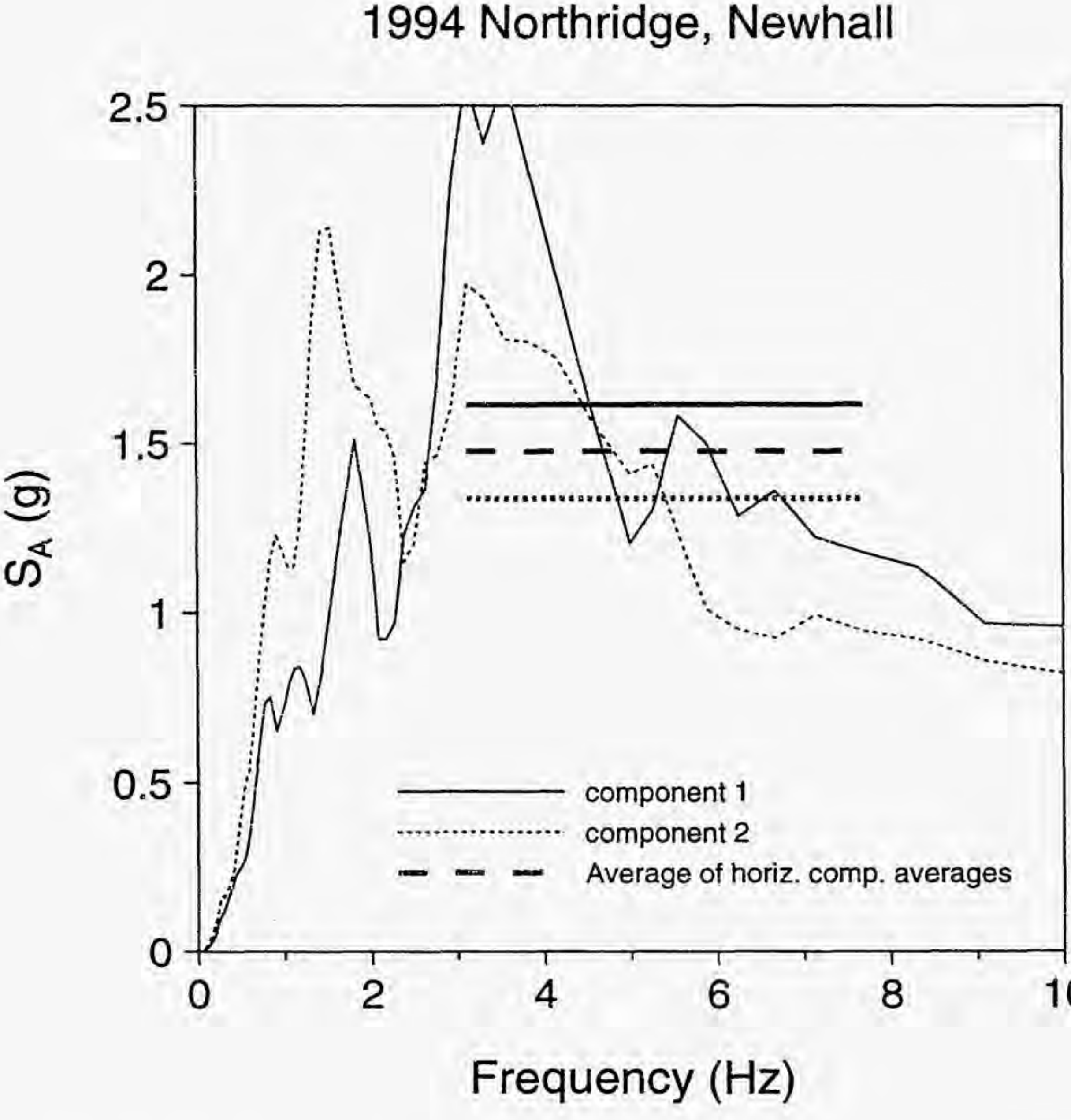
1994 Northridge, Jensen Admin Bldg.



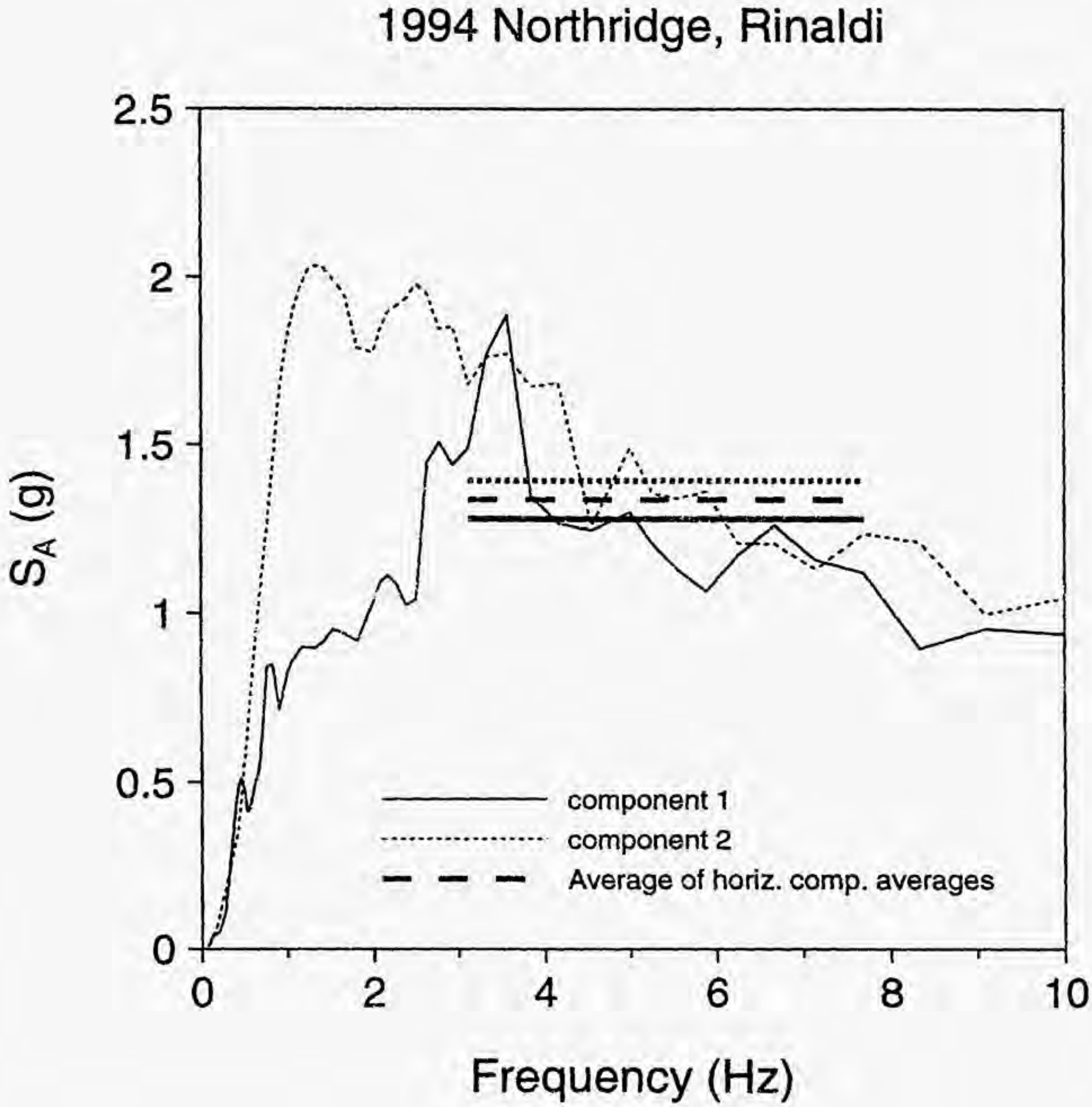
Apr 11, 1995 5:50:09 pm jenadmn.gra



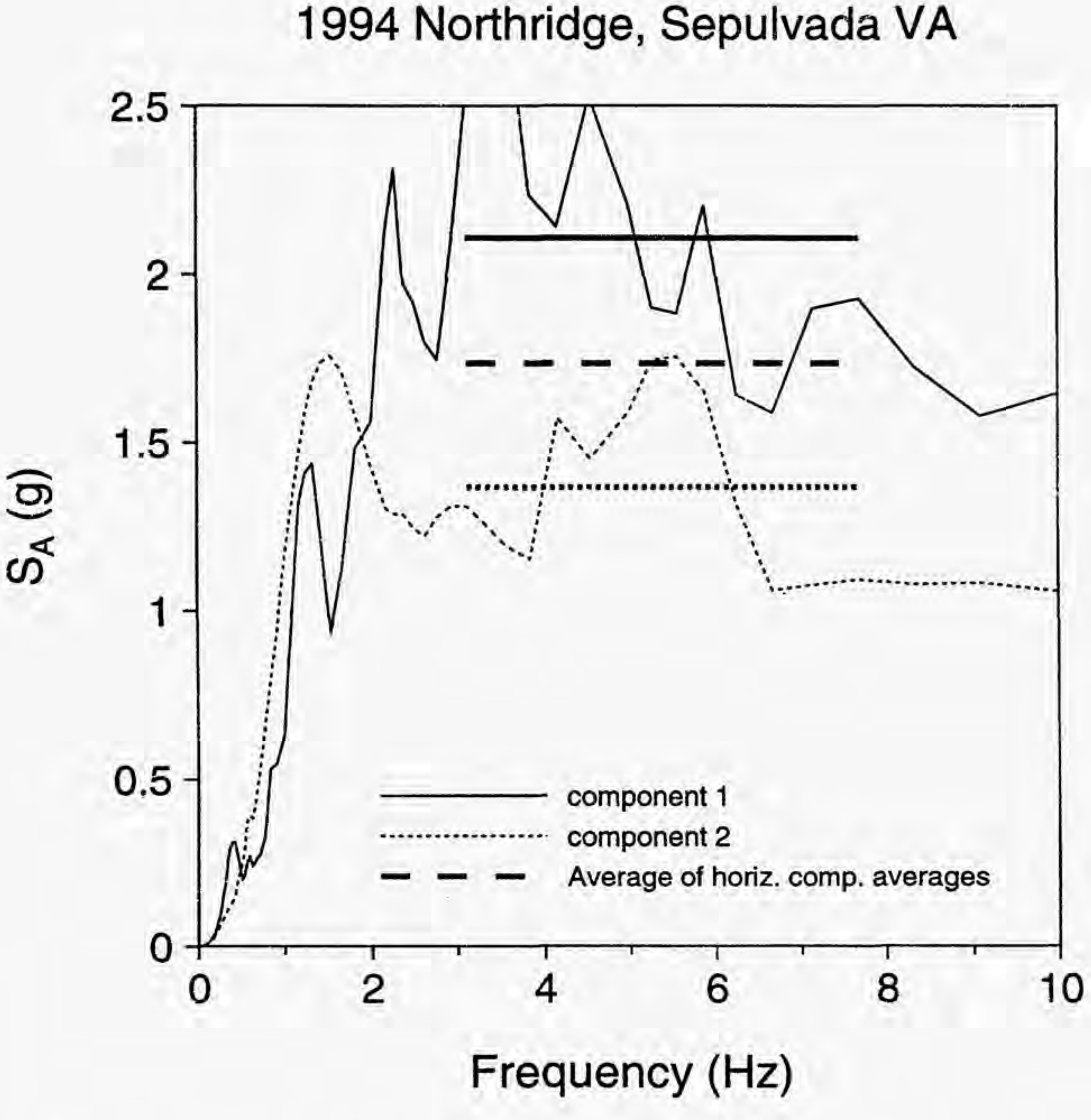
Apr 11, 1995 5:51:36 pm jengen.gra



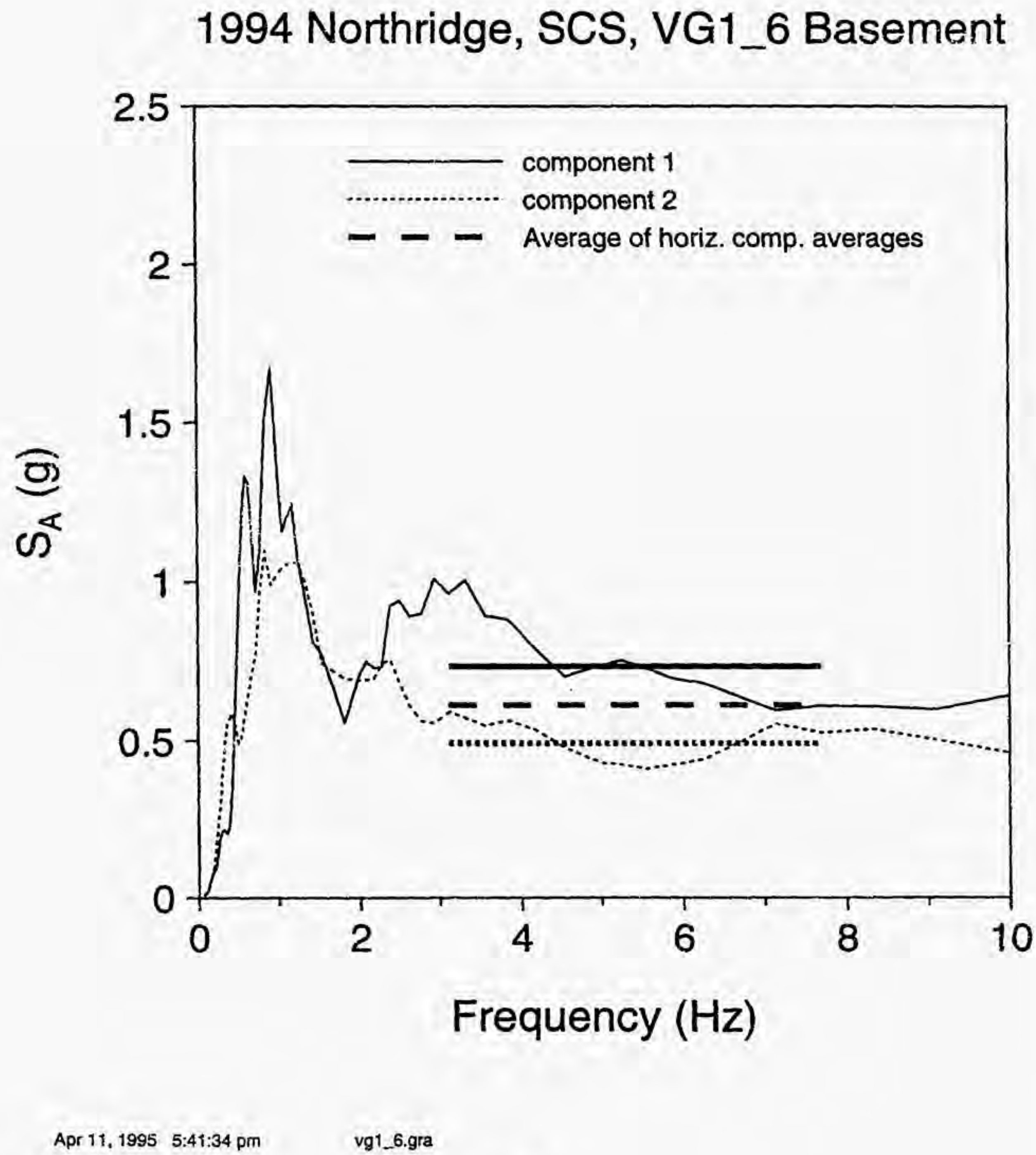
Apr 11, 1995 5:37:17 pm newh.gra

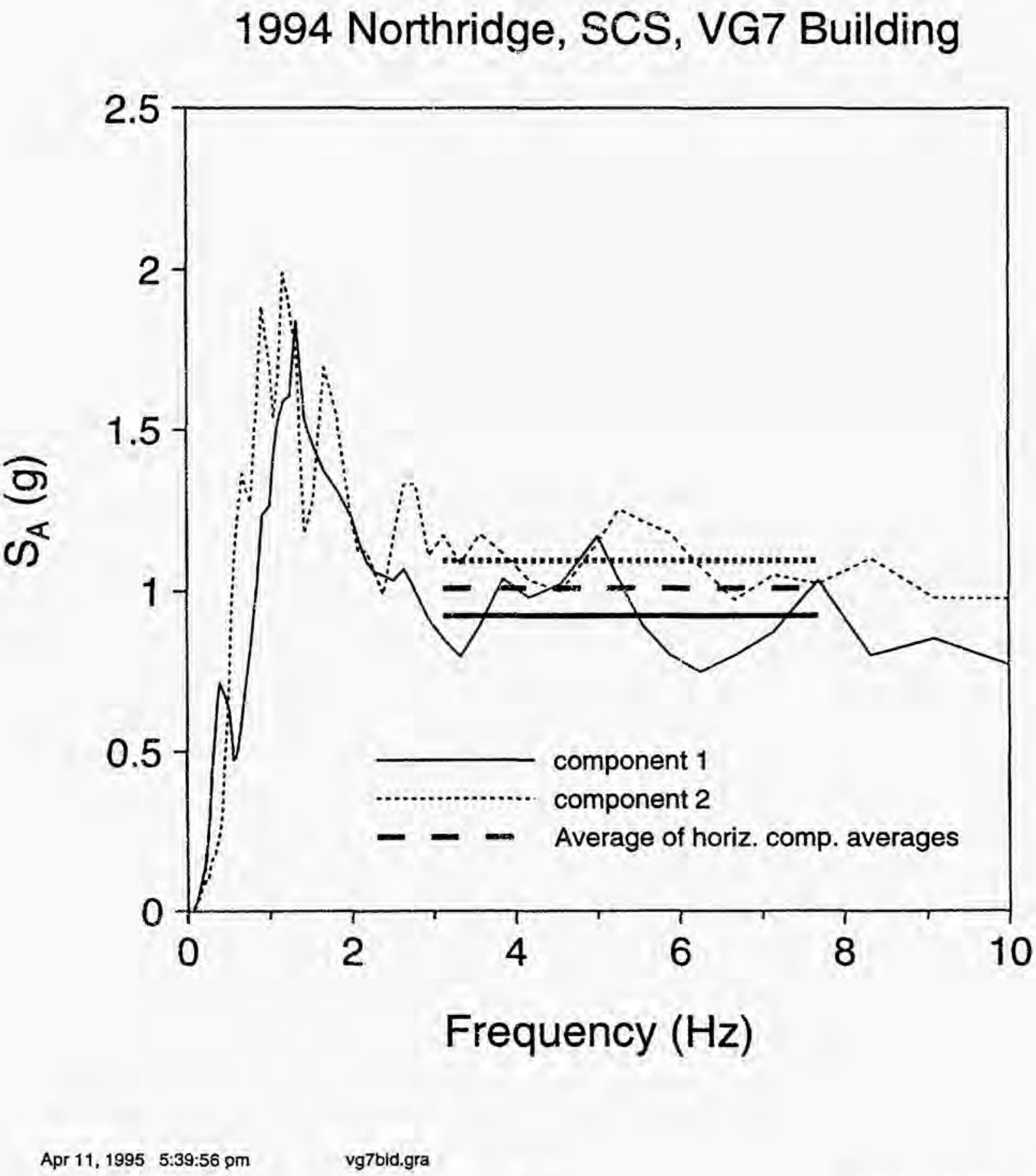


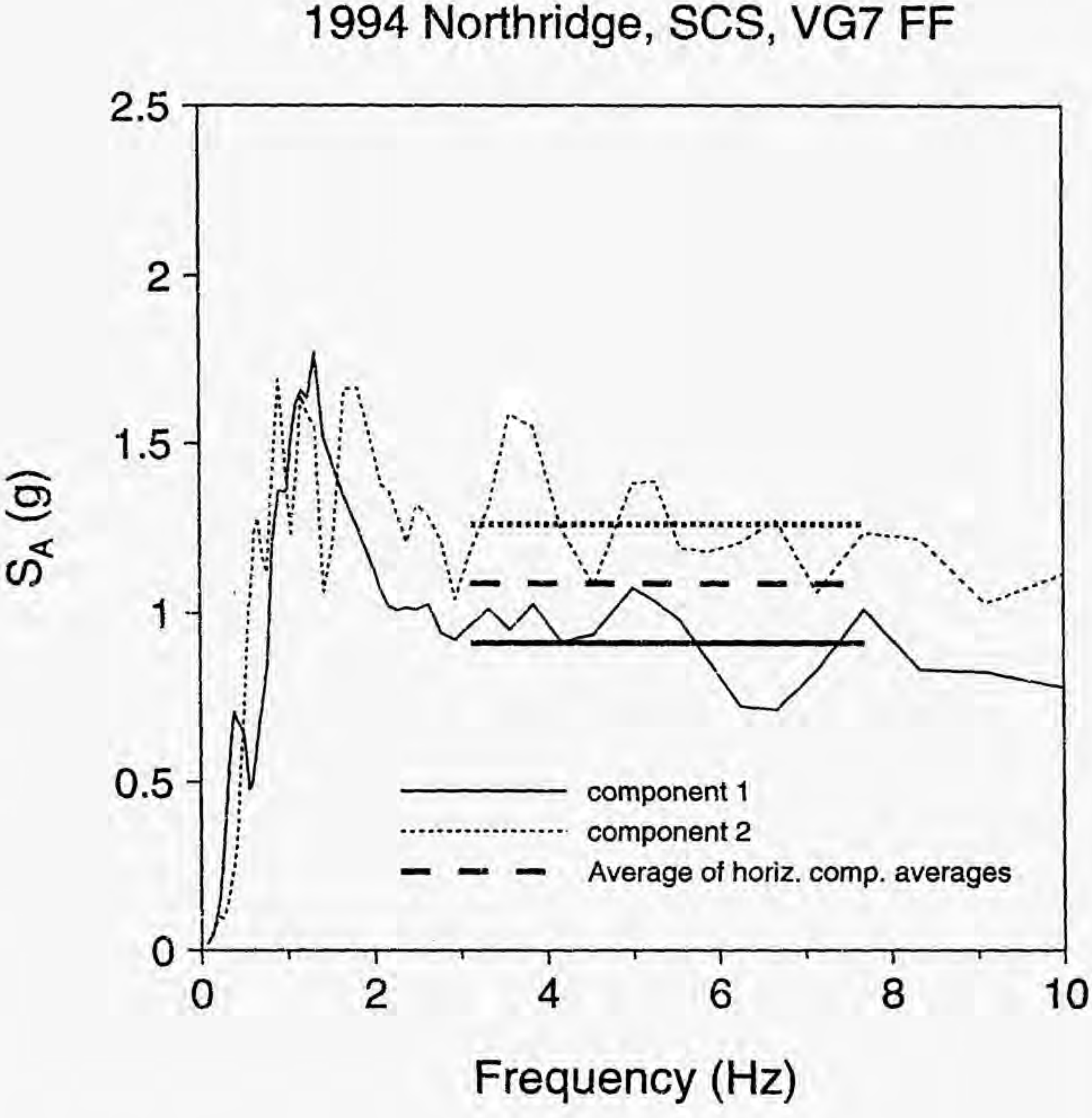
Apr 11, 1995 5:42:38 pm rinald.gra



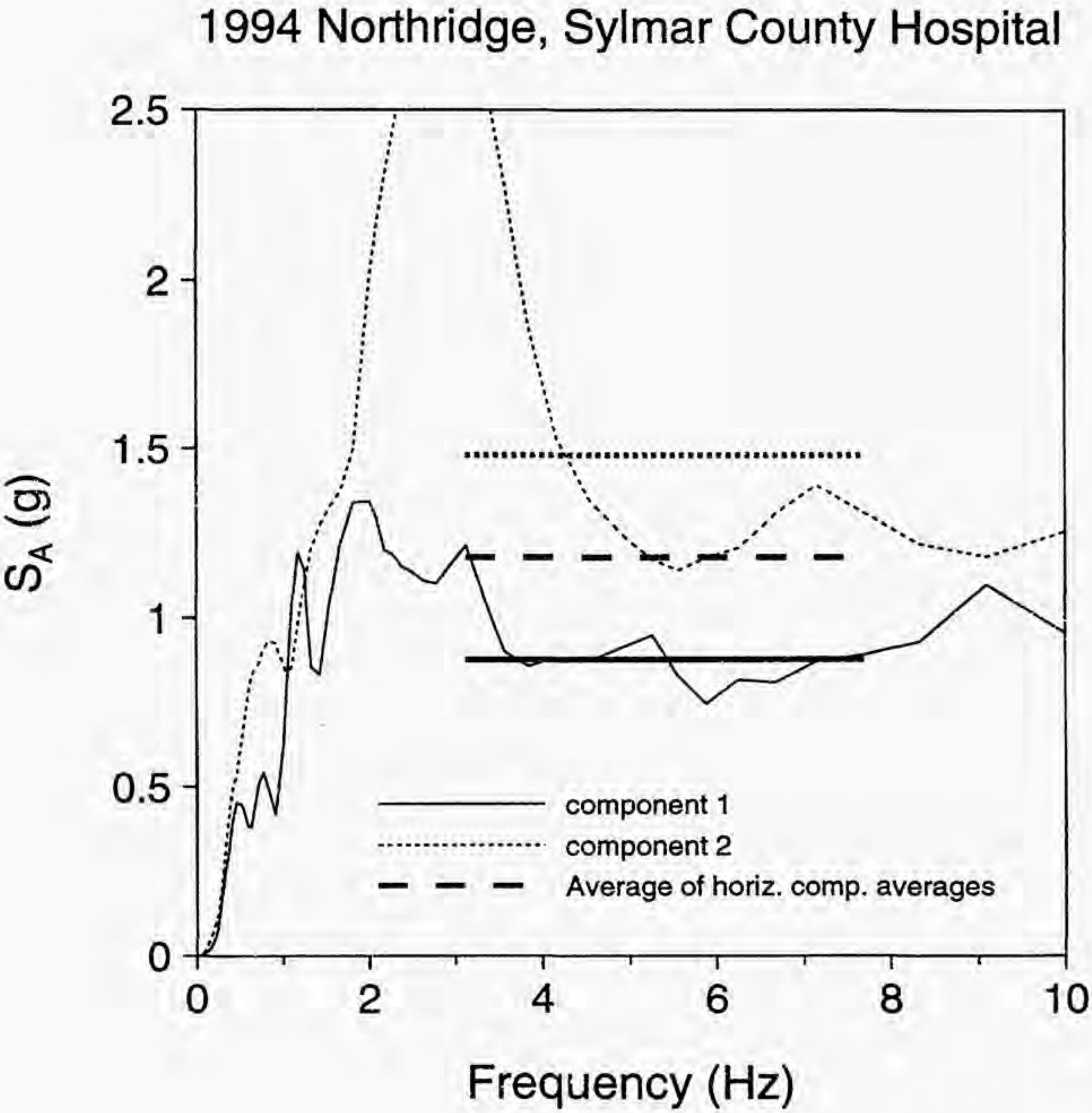
Apr 11, 1995 5:47:01 pm olive.gra


1994 Northridge, SCS, VG1_6 Basement
2.5
2
1.5
1
0.5
0
S_A (g)
component 1
component 2
Average of horiz. comp. averages
0
2
4
6
8
10
Frequency (Hz)
Apr 11, 1995 5:41:34 pm
vg1_6.gra

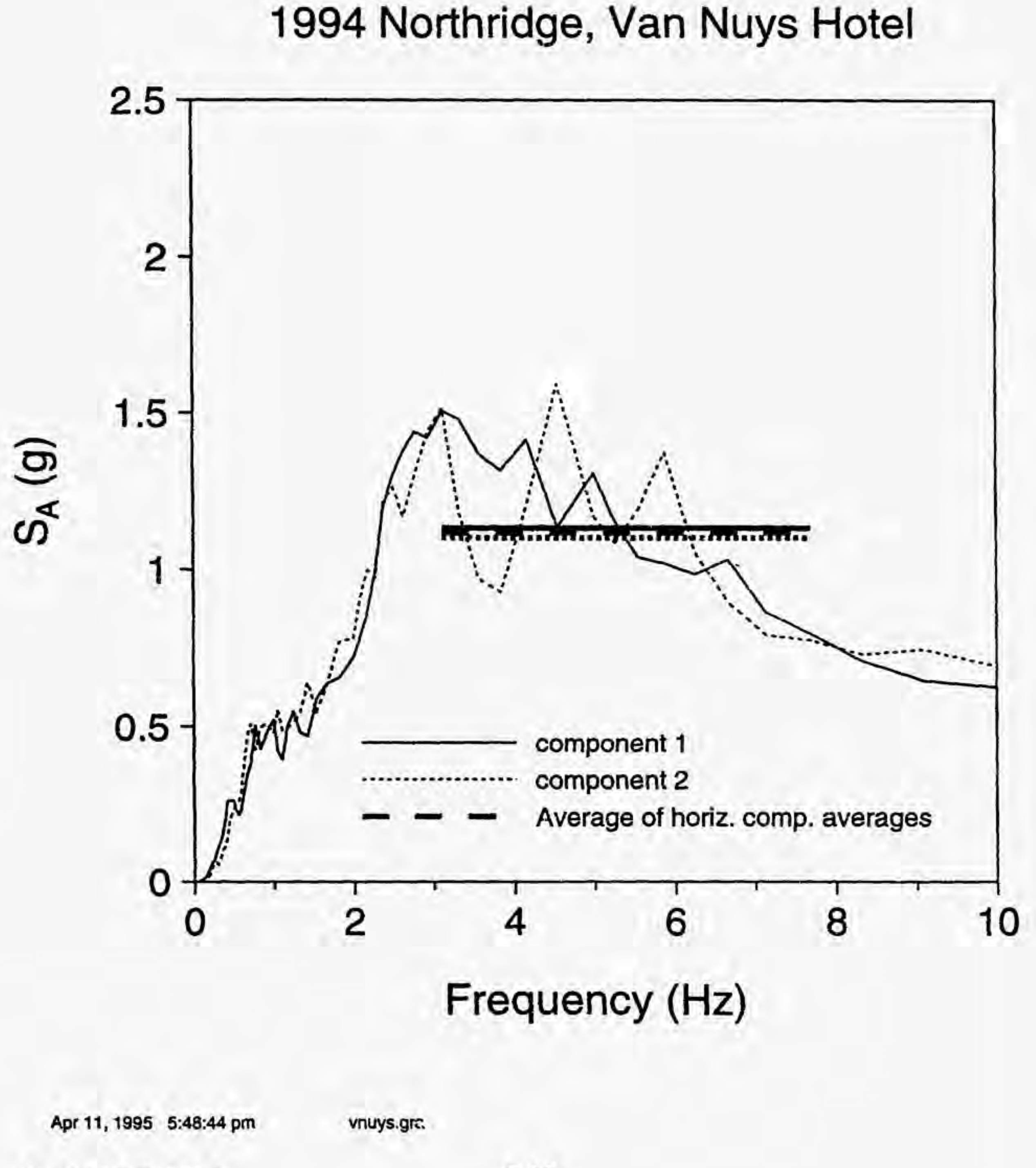

1994 Northridge, SCS, VG7 Building
2.5
2
1.5
1
0.5
0
S_A (g)
0
2
4
6
8
10
Frequency (Hz)
component 1
component 2
Average of horiz. comp. averages
Apr 11, 1995 5:39:56 pm
vg7bld.gra



Apr 11, 1995 5:38:51 pm vg7ff.gra



Apr 11, 1995 5:45:58 pm olive.gra


1994 Northridge, Van Nuys Hotel
2.5
2
1.5
1
0.5
0
S_A (g)
0
2
4
6
8
10
Frequency (Hz)
component 1
component 2
Average of horiz. comp. averages
Apr 11, 1995 5:48:44 pm
vnuys.grc

APPENDIX C.2

TABLES OF RESULTS

```
Summary of processing, file altwind.in

Contents of input file:

  |Nsta |DRef2O |AvgVel |SummaryFile
   2       6.2    520.0  altwind.sum

  |RS_Source |FileComp1    |FileComp2    |DSta2O |DSta2Ref |AvgVel |File4Plots
   bap        devers_a.rs2 devers_c.rs2 3.11    03.35     0520.0  junk.col
   bap        npalms_a.rs2 npalms_c.rs2 4.71    02.94     0520.0  junk.col

Results of Processing:

 For each station:

     FileComp1    FileComp2     SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   devers_a.rs2 devers_c.rs2  1.693  1.025  1.359        1.403   .848  1.125
   npalms_a.rs2 npalms_c.rs2  1.550  1.392  1.471        1.408  1.264  1.336

 Averaged over stations:

   AvgDSta2Ref  Sig  10*Sig   AvgOverStations  AvgCorrOverStations
      3.14      .17    1.47   1.41( 1.0, 2.1)   1.23(  .8, 1.8)
```

ALTWIND.SUM 4-11-95 1:59p

Page 1 of 1

```
Summary of processing, file buckwind.in

Contents of input file:

  |Nsta |DRef2Q |AvgVel |SummaryFile
   2      4.5    520.0  buckwind.sum

  |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
  bap         devers_a.rs2 devers_c.rs2 3.11    01.43     0520.0  junk.col
  bap         npalms_a.rs2 npalms_c.rs2 4.71    02.99     0520.0  junk.col

Results of Processing:

 For each station:

     FileComp1    FileComp2    SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
  devers_a.rs2 devers_c.rs2 1.693 1.025  1.359        1.566   .948  1.257
  npalms_a.rs2 npalms_c.rs2 1.550 1.392  1.471        1.572  1.411  1.492

 Averaged over stations:

  AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
     2.21      .15   1.42   1.41( 1.0, 2.0)    1.37( 1.0, 1.9)
```

BUCKWIND.SUM 4-11-95 1:59p   Page 1 of 1

```
Summary of processing, file devers.in

Contents of input file:

   |Nsta |DRef2O |AvgVel |SummaryFile
    2      3.1    520.0  devers.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2O |DSta2Ref |AvgVel |File4Plots
    bap        devers_a.rs2 devers_c.rs2 3.11    00.00     0520.0  junk.col
    bap        npalms_a.rs2 npalms_c.rs2 4.71    03.55     0520.0  junk.col

Results of Processing:

  For each station:

     FileComp1     FileComp2    SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   devers_a.rs2 devers_c.rs2  1.693  1.025  1.359       1.693  1.025  1.359
   npalms_a.rs2 npalms_c.rs2  1.550  1.392  1.471       1.698  1.524  1.611

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
      1.77      .14    1.39   1.41( 1.0, 2.0)   1.48( 1.1, 2.1)
```

DEVERS.SUM 4-11-95 1:59p

Page 1 of 1

Summary of processing, file garnet.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
2	7.0	520.0	garnet.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	devers_a.rs2	devers_c.rs2	3.11	04.77	0520.0	junk.col
hop	npalms_a.rs2	npalms_c.rs2	4.71	02.51	0520.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
devers_a.rs2	devers_c.rs2	1.693	1.025	1.359	1.331	.805	1.068
npalms_a.rs2	npalms_c.rs2	1.550	1.392	1.471	1.337	1.200	1.269

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
3.64	.17	1.49	1.41(1.0, 2.1)	1.16(.8, 1.7)

GARNET.SUM 4-11-95 1:59p

Page 1 of 1

```
Summary of processing, file renwind.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    3      5.5    520.0  renwind.sum

   |RS_Source |FileComp1     |FileComp2     |DSta2Q |DSta2Ref |AvgVel |File4Plots
    bap        devers_a.rs2 devers_c.rs2 3.11    06.21     0520.0  junk.col
    bap        wwater_a.rs2 wwater_c.rs2 0.00    08.34     0765.0  junk.col
    bap        npalms_a.rs2 npalms_c.rs2 4.71    09.29     0520.0  junk.col

Results of Processing:

  For each station:

     FileComp1     FileComp2     SA1    SA2  Avg1&2  Corr: SA1    SA2  Avg1&2
   devers_a.rs2 devers_c.rs2  1.693  1.025  1.359       1.471   .890  1.180
   wwater_a.rs2 wwater_c.rs2  1.359  1.495  1.427       1.213  1.333  1.273
   npalms_a.rs2 npalms_c.rs2  1.550  1.392  1.471       1.477  1.326  1.401

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig  AvgOverStations  AvgCorrOverStations
      7.95      .19   1.54   1.42(  .9, 2.2)   1.28(  .8, 2.0)
```

RENWIND.SUM 4-11-95 1:58p

Page 1 of 1

Summary of processing, file sanwind.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
2	3.3	520.0	sanwind.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	devers_a.rs2	devers_c.rs2	3.11	02.99	0520.0	junk.cot
bap	npalms_a.rs2	npalms_c.rs2	4.71	06.35	0520.0	junk.col

Results of Processing.

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
devers_a.rs2	devers_c.rs2	1.693	1.025	1.359	1.677	1.015	1.346
npalms_a.rs2	npalms_c.rs2	1.550	1.392	1.471	1.683	1.510	1.597

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
4.67	.18	1.52	1.41(.9, 2.1)	1.47(1.0, 2.2)

Summary of processing, file terawind.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
3	2.5	520.0	terawind.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	devers_a.rs2	devers_c.rs2	3.11	01.62	0520.0	junk.col
bap	npalms_a.rs2	npalms_c.rs2	4.71	02.47	0520.0	junk.col
bap	dsp_a.rs2	dsp_c.rs2	4.16	05.66	0520.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
devers_a.rs2	devers_c.rs2	1.693	1.025	1.359	1.745	1.056	1.401
npalms_a.rs2	npalms_c.rs2	1.550	1.392	1.471	1.750	1.571	1.661
dsp_a.rs2	dsp_c.rs2	.787	1.167	.977	.860	1.274	1.067

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
3.25	.16	1.44	1.25(.9, 1.8)	1.35(.9, 1.9)

TERAWIND.SUM 4-11-95 1:58p

Page 1 of 1

```
Summary of processing, file venwind.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    3      2.3    520.0  venwind.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
   bap         devers_a.rs2 devers_c.rs2 3.11    03.47     0520.0  junk.col
   bap         wwater_a.rs2 wwater_c.rs2 0.00    06.58     0765.0  junk.col
   bap         npalms_a.rs2 npalms_c.rs2 4.71    06.92     0520.0  junk.col

Results of Processing:

  For each station:

     FileComp1    FileComp2     SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   devers_a.rs2 devers_c.rs2  1.693  1.025  1.359       1.760  1.066  1.413
   wwater_a.rs2 wwater_c.rs2  1.359  1.495  1.427       1.452  1.593  1.523
   npalms_a.rs2 npalms_c.rs2  1.550  1.392  1.471       1.766  1.585  1.675

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig  AvgOverStations  AvgCorrOverStations
      5.66      .18    1.50  1.42(  .9, 2.1)   1.53( 1.0, 2.3)
```

```
Summary of processing, file wwhydro.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    2      2.6    520.0  wwhydro.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
   bap         devers_a.rs2 devers_c.rs2 3.11    05.75     0520.0  junk.col
   bap         wwater_a.rs2 wwater_c.rs2 0.00    05.38     0765.0  junk.col

Results of Processing:

  For each station:

     FileComp1    FileComp2     SA1     SA2 Avg1&2  Corr: SA1     SA2 Avg1&2
   devers_a.rs2 devers_c.rs2  1.693  1.025  1.359        1.737  1.052  1.395
   wwater_a.rs2 wwater_c.rs2  1.359  1.495  1.427        1.433  1.573  1.503

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
      5.57      .19    1.54   1.39(  .9, 2.1)   1.45(  .9, 2.2)
```

WWHYDRO.SUM 4-11-95 1:59p

Page 1 of 1

Summary of processing, file commerce.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
1	5.5	255.0	commerce.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	bulk_1.rs2	bulk_3.rs2	6.17	.83	0255.0	bulkmail.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
bulk_1.rs2	bulk_3.rs2	.948	1.168	1.058	.992	1.221	1.107

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
.83	.13	1.35	1.06(.8, 1.4)	1.11(.8, 1.5)

Summary of processing, file sctele.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
2	12.7	245.0	sctele.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
sil	cap000.050	cap090.050	8.57	6.74	289.0	junk.col
sil	wah000.050	wah090.050	9.69	3.24	340.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
cap000.050	cap090.050	1.333	.918	1.126	1.086	.747	.916
wah000.050	wah090.050	1.236	1.656	1.446	1.132	1.512	1.322

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
4.99	.18	1.52	1.28(.8, 1.9)	1.10(.7, 1.7)

SCTELE.SUM 4-11-95 4:39p

Page 1 of 1

```
Summary of processing, file scwater.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    2     11.0    340.0  scwater.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
   sil         lob000.050    lob090.050    12.53   2.51      612.0   junk.col
   sil         brn000.050    brn090.050    4.32    6.64      340.0   junk.col

Results of Processing:

  For each station:

     FileComp1     FileComp2     SA1     SA2  Avg1&2  Corr: SA1    SA2  Avg1&2
   lob000.050   lob090.050    1.443  1.151  1.297       1.865  1.489  1.677
   brn000.050   brn090.050    1.168  1.373  1.271        .760   .894   .827

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
      4.57      .18    1.51   1.28(  .8, 1.9)   1.16(  .8, 1.8)
```

SCWATER.SUM 4-11-95 4:39p

Page 1 of 1

```
Summary of processing, file soquel.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    2      7.3    289.0  soquel.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
   sil         cap000.050   cap090.050    8.57    1.48      289.0   junk.col
   sil         wah000.050   wah090.050    9.69    4.03      340.0   junk.col

Results of Processing:

  For each station:

     FileComp1     FileComp2     SA1     SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   cap000.050   cap090.050     1.333    .918  1.126      1.447   .996  1.222
   wah000.050   wah090.050     1.236   1.656  1.446      1.508  2.017  1.763

  Averaged over stations:

   AvgDSta2Ref   Sig  10^Sig   AvgOverStations   AvgCorrOverStations
      2.76       .16    1.45   1.28(  .9, 1.8)    1.47( 1.0, 2.1)
```

SOQUEL.SUM 4-11-95 4:39p

Page 1 of 1

```
Summary of processing, file ucsc.in

Contents of input file:

  |Nsta |DRef2Q |AvgVel |SummaryFile
   1     12.5    612.0  ucsc.sum

  |RS_Source |FileComp1   |FileComp2   |DSta2Q |DSta2Ref |AvgVel |File4Plots
   sil        lob000.050   lob090.050   12.53   0.03      612.0   junk.col

Results of Processing:

 For each station:

     FileComp1    FileComp2     SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   lob000.050   lob090.050    1.443  1.151  1.297      1.443  1.151  1.297

 Averaged over stations:

  AvgDSta2Ref  Sig  10*Sig   AvgOverStations   AvgCorrOverStations
      .03      .03    1.08   1.30( 1.2, 1.4)   1.30( 1.2, 1.4)
```

UCSC.SUM 4-11-95 4:39p

Page 1 of 1

```
Summary of processing, file centerv.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    1      9.8    520.0  centerv.sum

   |RS_Source |FileComp1     |FileComp2     |DSta2Q |DSta2Ref |AvgVel |File4Plots
   bap         centrv_a.rs2 centrv_c.rs2 9.8      0.1        0520.0  centerv.col

Results of Processing:

  For each station:

      FileComp1     FileComp2     SA1     SA2 Avg1&2  Corr: SA1     SA2 Avg1&2
   centrv_a.rs2 centrv_c.rs2  1.101    .908  1.005        1.101    .908  1.005

  Averaged over stations:

   AvgDSta2Ref  Sig  10*Sig   AvgOverStations  AvgCorrOverStations
       .10      .06    1.14   1.00(  .9, 1.1)   1.00(  .9, 1.1)
```

```
Summary of processing, file riodell.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    1     12.3    520.0  riodell.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
   bjf         349x0567.002 349x0567.272 12.3    2.5        0520.0  riodell.col

Results of Processing:

  For each station:

     FileComp1    FileComp2    SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   349x0567.002 349x0567.272  1.073   .788   .930       1.073   .788   .930

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig   AvgOverStations   AvgCorrOverStations
      2.50      .18    1.52    .93(  .6, 1.4)    .93(  .6, 1.4)
```

RIODELL.SUM 4-11-95 3:29p

Page 1 of 1

Summary of processing, file finance.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
3	.0	255.0	finance.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	sepulv_1.rs2	sepulv_3.rs2	0.41	7.98	400.0	junk.col
bap	vnuys_n.rs2	vnuys_w.rs2	2.09	8.41	366.0	junk.col
bap	rinald_1.rs2	rinald_3.rs2	0.00	9.06	282.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
sepulv_1.rs2	sepulv_3.rs2	2.106	1.363	1.734	2.413	1.557	1.985
vnuys_n.rs2	vnuys_w.rs2	1.132	1.101	1.117	1.315	1.277	1.296
rinald_1.rs2	rinald_3.rs2	1.278	1.392	1.335	1.316	1.434	1.375

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
8.48	.19	1.54	1.37(.9, 2.1)	1.52(1.0, 2.3)

FINANCE.SUM 4-13-95 7:27p

Page 1 of 1

```
Summary of processing, file olivcogn.in

Contents of input file:

  |Nsta |DRef2Q |AvgVel |SummaryFile
   1      3.6    385.0  olivcogn.sum

  |RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
  bap         olive_1.rs2  olive_3.rs2  3.59    0.200     0385.0  junk.cot

Results of Processing:

 For each station:

    FileComp1     FileComp2     SA1     SA2 Avg1&2  Corr: SA1     SA2 Avg1&2
  olive_1.rs2  olive_3.rs2    .876  1.479  1.178          .876  1.479  1.178

 Averaged over stations:

  AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
      .20      .08    1.19   1.18( 1.0, 1.4)   1.18( 1.0, 1.4)
```

```
Summary of processing, file placcgn1.in

Contents of input file:

  |Nsta |DRef2Q |AvgVel |SummaryFile
   1      4.9    385.0  placcgn1.sum

  |RS_Source |FileComp1   |FileComp2   |DSta2Q |DSta2Ref |AvgVel |File4Plots
  bap        newh_1.rs2   newh_3.rs2   4.53    3.45      0245.0  junk.col

Results of Processing:

  For each station:

     FileComp1    FileComp2    SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   newh_1.rs2   newh_3.rs2   1.615  1.334  1.475       1.377  1.137  1.257

  Averaged over stations:

  AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
     3.45      .20    1.57   1.47(  .9, 2.3)   1.26(  .8, 2.0)
```

PLACCGN1.SUM 4-13-95 7:27p

Page 1 of 1

Summary of processing, file placcgn2.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
4	4.9	385.0	placcgn2.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	newh_1.rs2	newh_3.rs2	4.53	3.45	0245.0	junk.col
bap	jengena.rs2	jengenc.rs2	0.00	7.42	0385.0	junk.col
bap	vg7ff_1.rs2	vg7ff_3.rs2	0.00	7.59	0282.0	junk.col
bap	olive_1.rs2	olive_3.rs2	3.59	7.76	0385.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
newh_1.rs2	newh_3.rs2	1.615	1.334	1.475	1.377	1.137	1.257
jengena.rs2	jengenc.rs2	1.061	2.067	1.564	.876	1.706	1.291
vg7ff_1.rs2	vg7ff_3.rs2	.911	1.260	1.086	.688	.952	.820
olive_1.rs2	olive_3.rs2	.876	1.479	1.178	.811	1.367	1.089

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
6.55	.18	1.50	1.31(.9, 2.0)	1.10(.7, 1.6)

PLACCGN2.SUM 4-14-95 12:51p

Page 1 of 1

Summary of processing, file rinaldi.in

Contents of input file:

Nsta	DRef2O	AvgVel	SummaryFile
1	.0	282.0	rinaldi.sum

RS_Source	FileComp1	FileComp2	DSta2O	DSta2Ref	AvgVel	File4Plots
bap	rinald_1.rs2	rinald_3.rs2	0.0	0.200	0282.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
rinald_1.rs2	rinald_3.rs2	1.278	1.392	1.335	1.278	1.392	1.335

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
.20	.08	1.19	1.33(1.1, 1.6)	1.33(1.1, 1.6)

RINALDI.SUM 4-13-95 7:29p

Page 1 of 1

```
Summary of processing, file scs_1.in

Contents of input file:

   |Nsta |DRef2Q |AvgVel |SummaryFile
    1       .0    282.0  scs_1.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2O |DSta2Ref |AvgVel |File4Plots
   bap          vg1_6_1.rs2  vg1_6_3.rs2  0.00     0.010    0282.0  junk.col

Results of Processing:

  For each station:

     FileComp1    FileComp2     SA1    SA2  Avg1&2  Corr: SA1    SA2  Avg1&2
   vg1_6_1.rs2  vg1_6_3.rs2    .736   .494   .615         .736   .494   .615

  Averaged over stations:

   AvgDSta2Ref  Sig  10*Sig   AvgOverStations   AvgCorrOverStations
       .01      .02   1.05     .62(  .6,  .6)     .62(  .6,  .6)
```

```
Summary of processing, file scs_2.in

Contents of input file:

|Nsta |DRef2Q |AvgVel |SummaryFile
 2      .0    282.0   scs_2.sum

|RS_Source |FileComp1    |FileComp2    |DSta2Q |DSta2Ref |AvgVel |File4Plots
 bap        vg1_6_1.rs2   vg1_6_3.rs2   0.00    0.010     0282.0  junk.col
 bap        vg7ff_1.rs2   vg7ff_3.rs2   0.00    0.300     0252.0  junk.col

Results of Processing:

For each station:

  FileComp1    FileComp2    SA1   SA2   Avg1&2  Corr: SA1   SA2   Avg1&2
 vg1_6_1.rs2  vg1_6_3.rs2   .736  .494  .615          .736  .494  .615
 vg7ff_1.rs2  vg7ff_3.rs2   .911 1.260 1.086          .911 1.260 1.086

Averaged over stations:

AvgDSta2Ref  Sig  10^Sig  AvgOverStations  AvgCorrOverStations
   .16       .06   1.14    .82( .7, .9)     .82( .7, .9)
```

SCS_2.SUM 4-13-95 9:07p

Page 1 of 1

Summary of processing, file scs_3.in

Contents of input file:

Nsta	DRef2Q	AvgVel	SummaryFile
1	.0	282.0	scs_3.sum

RS_Source	FileComp1	FileComp2	DSta2Q	DSta2Ref	AvgVel	File4Plots
bap	vg7ff_1.rs2	vg7ff_3.rs2	0.00	0.300	0282.0	junk.col

Results of Processing:

For each station:

FileComp1	FileComp2	SA1	SA2	Avg1&2	Corr: SA1	SA2	Avg1&2
vg7ff_1.rs2	vg7ff_3.rs2	.911	1.260	1.086	.911	1.260	1.086

Averaged over stations:

AvgDSta2Ref	Sig	10^Sig	AvgOverStations	AvgCorrOverStations
.30	.09	1.23	1.09(.9, 1.3)	1.09(.9, 1.3)

SCS_3.SUM 4-14-95 10:15a

Page 1 of 1

```
Summary of processing, file scs_vg7.in

Contents of input file:

   |Nsta |DRef2O |AvgVel |SummaryFile
    2       .0    282.0  scs_vg7.sum

   |RS_Source |FileComp1    |FileComp2    |DSta2O |DSta2Ref |AvgVel |File4Plots
   bap         vg7ff_1.rs2  vg7ff_3.rs2  0.00    0.030     0282.0  junk.col
   bap         vg7bld_1.rs2 vg7bld_3.rs2 0.00    0.008     0282.0  junk.col

Results of Processing:

  For each station:

      FileComp1    FileComp2     SA1    SA2 Avg1&2  Corr: SA1    SA2 Avg1&2
   vg7ff_1.rs2  vg7ff_3.rs2     .911  1.260  1.086        .911  1.260  1.086
   vg7bld_1.rs2 vg7bld_3.rs2    .922  1.094  1.008        .922  1.094  1.008

  Averaged over stations:

   AvgDSta2Ref  Sig  10^Sig   AvgOverStations  AvgCorrOverStations
       .02      .02    1.05   1.05( 1.0, 1.1)   1.05( 1.0, 1.1)
```

SCS_VG7.SUM 4-14-95 10:37a

Page 1 of 1

APPENDIX C.3

LISTINGS OF PROGRAMS

```
      Program GetDist

* Gets interstation distances.  Use as a preprocessor for
* CorrelPA.

* Note: Uses input file made by Paradox, report in RECS_IN, sorted by distance.
* Note that the report file must be cleaned up, making sure that the
* data start on line 11, eliminating page breaks, deleting the
* very last line (some distance below the
* last actual entry), and putting "stop" after the last entry (starting
* in column 2)

* Dates:
*  03/20/95 - Written by Dave Boore

      real lat(300), long(300)
      real temp_delta, delta(3000), delta_max
      integer indx(3000), ndelta, max_ndelta,
     :  temp_ista_1, temp_ista_2, ista_1(3000), ista_2(3000)
      character buffer*150

      delta_max = 15.0
      max_ndelta = 1500

* Read in data:

      nu_in = 10

      open(unit= nu_in, file = 'nr94pga.prn', recl = 150,
     :    status = 'unknown')

      do i = 1, 10
        read(nu_in, *)
      end do

      nsta = 0

100   continue
      buffer = ' '
      read(nu_in, '(a)') buffer

      if (buffer(2:5) .eq. 'stop') goto 1000
      nsta = nsta + 1
      read(buffer(93:98), '(f6.3)') lat(nsta)
      read(buffer(100:106), '(f7.3)') long(nsta)
      goto 100

1000  continue
      close(unit=nu_in)

*<<<<<<<<<<<<<<<<<<<<<<<<<<<<< START COMPUTATION OF DISTANCES >>>>>>>>>>>>>>>

      ndelta = 0
      do i = 1, nsta-1
        do j = i+1, nsta
          call distaz(+1.0, lat(i), long(i), lat(j), long(j),
     :         rdeg, temp_delta, az, baz)
          temp_ista_1 = i
          temp_ista_2 = j
          if (temp_delta .le. delta_max) then
            ndelta = ndelta + 1

            if (ndelta .gt. max_ndelta) then
              ndelta = max_ndelta
              go to 9999
            end if

            delta(ndelta) = temp_delta
            ista_1(ndelta) = temp_ista_1
            ista_2(ndelta) = temp_ista_2

          end if
        end do
      end do

9999  continue

* Now sort and write the information:

      call indexx( ndelta, delta, indx)

      nu_out = 30
      open(unit = nu_out, file = 'delta.out', status='unknown')

      write(nu_out, '(a,f6.1)') ' Max delta = ', delta_max
      write(nu_out, '(a)') ' ista_1 ista_2  delta'

      do i = 1, ndelta
        write(nu_out, '(4x,i3,4x,i3,1x,f6.2)')
     :          ista_1(indx(i)), ista_2(indx(i)), delta(indx(i))
      end do

      close(unit=nu_out)

      stop
      end


      subroutine distaz( wlongsign, alat, along, blat, blong,
     * rdeg, rkm, az, baz)
c
c compute distances, azimuths using formulas from
c Bruce Julian.
c
c latest modification: 1/27/84
c
      pi = 4.0 * atan( 1. )
      dtor = pi/ 180.
c
c convert from degrees to radians and correct sign of
c longitude so that east longitude is positive.
c
      alatr = dtor * alat
      alongr = -dtor * along * wlongsign
      blatr = dtor * blat
      blongr = -dtor * blong * wlongsign
c
c compute geocentric latitudes.
c
      alatr = atan( 0.993305 * tan( alatr ) )
      blatr = atan( 0.993305 * tan( blatr ) )
c
c compute latitude dependent quantities
c
      ca = cos( alatr )
      cb = cos( blatr )
      sa = sin( alatr )
      sb = sin( blatr )
c
c now compute other quantities
c
      a = cb * sin( blongr - alongr )
```

GETDIST.FOR 3-20-95 9:39p

Page 1 of 2

```
      b = sa * sb - sa * cb * cos( blongr - alongr )
      cd = ca * cb * cos( blongr - alongr ) + sa * sb
      sd = sqrt( a*a + b*b )
c
c compute distances
c
      rdeg = atan2( sd, cd )/ dtor
      rkm = 111.19 * rdeg
c
c compute azimuth (from a to b) and make it positive.
c
      az = atan2( a, b )/ dtor
      if ( az .lt. 0.0 ) az = az + 360.0
c
c compute back azimuth (from b to a) and make it positive.
c
      a = ca * sin( alongr - blongr )
      b = cb * sa - sb * ca * cos( alongr - blongr )
      baz = atan2( a, b)/ dtor
      if ( baz .lt. 0.0 ) baz = baz + 360.0
c
      return
      end


      SUBROUTINE indexx(n,arr,indx)
      INTEGER n,indx(n),M,NSTACK
      REAL arr(n)
      PARAMETER (M=7,NSTACK=50)
      INTEGER i,indxt,ir,itemp,j,jstack,k,l,istack(NSTACK)
      REAL a
      do 11 j=1,n
        indx(j)=j
11    continue
      jstack=0
      l=1
      ir=n
1     if(ir-l.lt.M)then
        do 13 j=l+1,ir
          indxt=indx(j)
          a=arr(indxt)
          do 12 i=j-1,1,-1
            if(arr(indx(i)).le.a)goto 2
            indx(i+1)=indx(i)
12        continue
          i=0
2         indx(i+1)=indxt
13      continue
        if(jstack.eq.0)return
        ir=istack(jstack)
        l=istack(jstack-1)
        jstack=jstack-2
      else
        k=(l+ir)/2
        itemp=indx(k)
        indx(k)=indx(l+1)
        indx(l+1)=itemp
        if(arr(indx(l+1)).gt.arr(indx(ir)))then
          itemp=indx(l+1)
          indx(l+1)=indx(ir)
          indx(ir)=itemp
        endif
        if(arr(indx(l)).gt.arr(indx(ir)))then
          itemp=indx(l)
          indx(l)=indx(ir)
          indx(ir)=itemp
        endif
        if(arr(indx(l+1)).gt.arr(indx(l)))then
          itemp=indx(l+1)
          indx(l+1)=indx(l)
          indx(l)=itemp
        endif
        i=l+1
        j=ir
        indxt=indx(l)
        a=arr(indxt)
3       continue
          i=i+1
        if(arr(indx(i)).lt.a)goto 3
4       continue
          j=j-1
        if(arr(indx(j)).gt.a)goto 4
        if(j.lt.i)goto 5
        itemp=indx(i)
        indx(i)=indx(j)
        indx(j)=itemp
        goto 3
5       indx(l)=indx(j)
        indx(j)=indxt
        jstack=jstack+2
        if(jstack.gt.NSTACK)pause 'NSTACK too small in indexx'
        if(ir-i+1.ge.j-l)then
          istack(jstack)=ir
          istack(jstack-1)=i
          ir=i-1
        else
          istack(jstack)=j-1
          istack(jstack-1)=l
          l=i
        endif
      endif
      goto 1
      END
C  (C) Copr. 1986-92 Numerical Recipes Software $!6)$-"11j.
```

```
      Program CorrelPA

* Computes correlation of peak accelerations as a function of distance

* Note: Uses input file made by Paradox, report in RECS_IN, sorted by distance.
* Note that the report file must be cleaned up, making sure that the
* data start on line 11, deleting the very last line (some distance below the
* last actual entry), and putting "stop" after the last entry (starting
* in column 2)

* Dates:
*  03/18/95 - Written by Dave Boore

      real mag(300), dist(300), pgah1(300), lat(300), long(300)
      real resid(300), d_bin(100), ave_bin(100)
      real bjf93(300), resid_old(300)
      integer ibin(300), num_not_empty_bins, bin_num_not_empty(100),
     :  istart_not_empty_bin(100), istop_not_empty_bin(100)
      character buffer*150, stem_name*8, f_out*12

      real temp_delta, delta(1000), delta_max
      integer indx(1000), ndelta, max_ndelta,
     :  temp_ista_1, temp_ista_2, ista_1(1000), ista_2(1000)

      real diff_old(1000), diff_new(1000)

* !!!!!!!!!! SET SOME PARAMETERS !!!!!!!!!!!!!!!!

      max_ndelta = 1000
      nu_in = 10
      nu_atn = 20
      nu_ald = 30
      nu_std = 40
      nu_als = 50

      d_min = 0.1
      d_max = 300

*<<<<<<<<<<<<<<<< OBTAIN DATA AND PARAMETERS >>>>>>>>>>>>>>>>

      write(*, '(a\)') ' Enter stem name for files: '
      stem_name = ' '
      read(*, '(a8)') stem_name

*      write(*, '(a\)') ' Enter d_min: '
*      read(*, '(f5.0)') d_min

*      write(*, '(a\)') ' Enter d_max: '
*      read(*, '(f5.0)') d_max

*      write(*, '(a\)') ' Enter n_bin: '
*      read(*, '(i3)') n_bin

      write(*, '(a\)') ' Enter num_pnts_per_dist_bin: '
      read(*, '(i2)') num_pnts_per_dist_bin

      write(*, '(a\)') ' Enter delta_max: '
      read(*, '(f5.0)') delta_max

      write(*, '(a\)') ' Enter num_diff_per_dlta_bin: '
      read(*, '(i2)') num_diff_per_dlta_bin

      write(*, '(a\)') ' Enter num_diff_2_print: '
      read(*, '(i4)') num_diff_2_print

* Read in data:
      open(unit= nu_in, file = 'nr94pga.prn', recl = 150,
     :     status = 'unknown')

      do i = 1, 10
        read(nu_in, *)
      end do

      nsta = 0

100   continue
      buffer = ' '
      read(nu_in, '(a)') buffer

      if (buffer(2:5) .eq. 'stop') goto 1000
      nsta = nsta + 1
      read(buffer(15:18), '(f4.2)') mag(nsta)
      read(buffer(20:24), '(f5.1)') dist(nsta)
      read(buffer(32:35), '(f4.2)') pgah1(nsta)
      read(buffer(93:98), '(f6.3)') lat(nsta)
      read(buffer(100:106), '(f7.3)') long(nsta)
      goto 100

1000  continue

      close(unit=nu_in)

*<<<<<<<<<<<<<<<< ATTENUATION ANALYSIS >>>>>>>>>>>>>>>>

* Debug
      i = 1
      write(*, '(a,i4,f5.2,f6.1,f5.2,f7.3,f8.3)')
     :  ' i=', i, mag(i), dist(i), pgah1(i), lat(i), long(i)
      i = nsta
      write(*, '(a,i4,f5.2,f6.1,f5.2,f7.3,f8.3)')
     :  ' i=', i, mag(i), dist(i), pgah1(i), lat(i), long(i)
* Debug

* Compute residuals, relative to BJF93:

      do i = 1, nsta
        r = sqrt(dist(i)**2 + (5.48)**2)
        bjf93(i) = -0.038+0.216*(mag(i)-6.0)-0.777*alog10(r)+0.254
        resid(i) = alog10(pgah1(i)) - bjf93(i)
      end do

* Data are already sorted by distance (this was done by the Paradox report),
* so now find the bins:

* Skip over this old coding:

      goto 9119

6116  continue

* Define bins (this was originally inside bin_data.  Put
* it outside to increase flexibility):
      dlogd = (alog10(d_max/d_min))/n_bin
      do i = 1, n_bin
        d_bin(i) = d_min * 10.0**((i-1)*dlogd)
      enddo

* Reset all distances less than d_min to slightly more than d_min:
      do i = 1, nsta
        if (dist(i) .le. d_min) dist(i) = 1.001* d_min
```

```
      end do

      call bin_data(d_bin, n_bin,
     :   dist, nsta, ibin,
     :   num_not_empty_bins, bin_num_not_empty,
     :   istart_not_empty_bin, istop_not_empty_bin)

* Debug
      open(unit=30, file='temp.out', status='unknown')
      do i = 1, nsta
        write(30, '(a,i4,f5.2, f6.1,i4,f6.1,f6.1)')
     :    ' ista, resid, dist, ibin, d_bin_low, d_bin_high =',
     :    i, resid(i), dist(i),
     :    ibin(i), d_bin(ibin(i)), d_bin(ibin(i))*10.0**dlogd
      end do
* Debug

* Now compute the mean residual in each bin:
* (requires that resid be sorted by distance)

      do i = 1, num_not_empty_bins
        call momntdmb(resid,
     :    istart_not_empty_bin(i), istop_not_empty_bin(i),
     :    ave_bin(bin_num_not_empty(i)), adev, sdev, var, skew, curt)


* Debug
        istart = istart_not_empty_bin(i)
        istop =  istop_not_empty_bin(i)
        write(30, '(2a, i3,i4,i4,i4,i4,f6.1,f5.2)')
     :    ' n,bn_#_nt_mpty,#_in_bn,istrt,istp,',
     :    'avdst,resid = ',
     :    i, bin_num_not_empty(i),
     :    istop - istart + 1, istart, istop,
     :    d_bin(bin_num_not_empty(i))*10.0**(dlogd/2.0),
     :    ave_bin(bin_num_not_empty(i))
* Debug

      end do

* Find the new residual for each station by subtracting the average found above
* for each bin:

      do i = 1, nsta
        resid_old(i) = resid(i)
        resid(i) = resid_old(i) - ave_bin(ibin(i))

        write(30, '(a,i4, i4,3f6.2)')
     :    ' nsta, ibin, resid_old, ave_bin, resid_new= ',
     :      i, ibin(i), resid_old, ave_bin(ibin(i)), resid(i)

      end do

9119  continue

      f_out = ' '
      f_out = stem_name//'.atn'

      open(unit=nu_atn, file = f_out, status='unknown')
      write(nu_atn, 721)
721   format(t2,'ibin', t7,'istrt', t13,'istp', t18,'d_strt',
     :        t25,'d_stp', t31,'avdst', t38,'resid')

      f_out = ' '
      f_out = stem_name//'.ald'
      open(unit=nu_ald, file = f_out, status='unknown')
      write(nu_ald, 722)
722   format( t2,'ista', t7,'ibin', t13,'dist',
     :        t18,'res_old', t26,'res_new',
     :        t34,'bjf+ave_bin',
     :        t46,'ave_bin', t54,'bjf93')

      number_dist_bins = nsta/num_pnts_per_dist_bin

      istrt = -num_pnts_per_dist_bin + 1
      do i = 1, number_dist_bins
        istrt = istrt + num_pnts_per_dist_bin
        istp = istrt + num_pnts_per_dist_bin - 1
        call momntdmb(resid, istrt, istp, ave_bin(i),
     :     adev, sdev, var, skew, curt)
        write(nu_atn, '(t3,i3, t9,i3, t14,i3, t19,f5.1,
     :        t25,f5.1, t31,f5.1, t37,f6.3)')
     :        i, istrt, istp, dist(istrt), dist(istp),
     :        sqrt(dist(istrt)*dist(istp)), ave_bin(i)

        do j = istrt, istp
           resid_old(j) = resid(j)
           resid(j) = resid_old(j) - ave_bin(i)
           write(nu_ald, '(t3,i3, t8,i3, t12,f5.1,
     :             t20,f5.2, t28,f5.2, t40,f5.2,
     :             t48,f5.2, t54,f5.2)')
     :       j, i, dist(j), resid_old(j), resid(j),
     :       bjf93(j) + ave_bin(i), ave_bin(i), bjf93(j)
        end do

      end do

      close(unit=nu_ald)
      close(unit=nu_atn)

*<<<<<<<<<<<<<<<< INTERSTATION ANALYSIS >>>>>>>>>>>>>>>

      ndelta = 0
      do i = 1, nsta-1
        do j = i+1, nsta
          call distaz(+1.0, lat(i), long(i), lat(j), long(j),
     :         rdeg, temp_delta, az, baz)
          temp_ista_1 = i
          temp_ista_2 = j
          if (temp_delta .le. delta_max) then
            ndelta = ndelta + 1

            if (ndelta .gt. max_ndelta) then
              ndelta = max_ndelta
              go to 9999
            end if

            delta(ndelta) = temp_delta
            ista_1(ndelta) = temp_ista_1
            ista_2(ndelta) = temp_ista_2

          end if
        end do
      end do

9999  continue

      write(*, '(a, i5)') ' ndelta = ', ndelta

* Now sort and fill difference array:

      call indexx( ndelta, delta, indx)
```

```
      do i = 1, ndelta
        resid1 = resid(ista_1(indx(i)))
        resid2 = resid(ista_2(indx(i)))
        diff_new(i) = resid2 - resid1
        resid1 = resid_old(ista_1(indx(i)))
        resid2 = resid_old(ista_2(indx(i)))
        diff_old(i) = resid2 - resid1
      end do

* Write the first num_diff_2_print values:

      f_out = ' '
      f_out = stem_name//'.als'

      open(unit=nu_als, file = f_out, status='unknown')
      write(nu_als, 723)
723   format(t2,'indx1', t8,'indx2', t14,'delta',
     :       t20,'resid1', t27,'resid2',
     :       t34,'diff_new', t43,'diff_old')

      imax = ndelta
      if (imax .gt. num_diff_2_print) imax = num_diff_2_print

      do i = 1, imax
        resid1 = resid(ista_1(indx(i)))
        resid2 = resid(ista_2(indx(i)))
        write(nu_als, '(t3,i4, t10,i4, t14,f5.2, t21,f5.2,
     :                t28,f5.2, t37,f5.2, t46,f5.2)')
     :        ista_1(indx(i)), ista_2(indx(i)), delta(indx(i)),
     :        resid1, resid2, diff_new(i), diff_old(i)
      end do

      close(unit=nu_als)

* Now set up bins for interstation spacing and compute sdev:

      f_out = ' '
      f_out = stem_name//'.std'

      open(unit=nu_std, file = f_out, status='unknown')
      write(nu_std, 724)
724   format(t2,'i_dlta_bin', t13,'istrt', t19,'istp',
     :       t24,'avg_dst', t32,'avg_new', t40,'std_new',
     :       t48,'avg_old', t56,'std_old')

      number_delta_bins = ndelta/num_diff_per_dlta_bin

      istrt = -num_diff_per_dlta_bin + 1
      do i = 1, number_delta_bins
        istrt = istrt + num_diff_per_dlta_bin
        istp = istrt + num_diff_per_dlta_bin - 1
        call momntdmb(diff_new, istrt, istp, ave_new,
     :                adev, sdev_new, var, skew, curt)
        call momntdmb(diff_old, istrt, istp, ave_old,
     :                adev, sdev_old, var, skew, curt)

        avgdist = 0.5*(delta(indx(istrt))+delta(indx(istp)))
        write(nu_std, '(t8,i3, t14,i4, t19,i4, t26,f5.2,
     :                 t34,f5.2, t41,f6.3, t50,f5.2, t57,f6.3)')
     :     i, istrt, istp, avgdist,
     :     ave_new, sdev_new, ave_old, sdev_old

      end do

      stop
      end

      subroutine bin_data(d_bin, n_bin,
     :   dist, nsta, ibin,
     :   num_not_empty_bins, bin_num_not_empty,
     :   istart_not_empty_bin, istop_not_empty_bin)

       real d_bin(*), dist(*)
       integer ibin(*)
       integer num_not_empty_bins, bin_num_not_empty(*),
     :   istart_not_empty_bin(*), istop_not_empty_bin(*)

* assign distances to bins:

      do i = 1, nsta
        call locate(d_bin, n_bin, dist(i), ibin(i))
      end do

* Find indices at start and stop of each bin:

      num_not_empty_bins = 1
      istart_not_empty_bin(1) = 1
      istop_not_empty_bin(1) = 1
      bin_num_not_empty(1) = ibin(1)

      do i = 1, nsta-1
        if (ibin(i+1) .eq. ibin(i)) then
          istop_not_empty_bin(num_not_empty_bins) = i+1
        else
          num_not_empty_bins = num_not_empty_bins + 1
          bin_num_not_empty(num_not_empty_bins) = ibin(i+1)
          istart_not_empty_bin(num_not_empty_bins) = i+1
          istop_not_empty_bin(num_not_empty_bins) = i+1
        end if
      end do

      return
      end


      subroutine distaz( wlongsign, alat, along, blat, blong,
     * rdeg, rkm, az, baz)
c
c compute distances, azimuths using formulas from
c Bruce Julian.
c
c latest modification: 1/27/84
c
      pi = 4.0 * atan( 1. )
      dtor = pi/ 180.
c
c convert from degrees to radians and correct sign of
c longitude so that east longitude is positive.
c
      alatr = dtor * alat
      alongr = -dtor * along * wlongsign
      blatr = dtor * blat
      blongr = -dtor * blong * wlongsign
c
c compute geocentric latitudes.
c
      alatr = atan( 0.993305 * tan( alatr ) )
      blatr = atan( 0.993305 * tan( blatr ) )
c
```

```
c compute latitude dependent quantities
c
      ca = cos( alatr )
      cb = cos( blatr )
      sa = sin( alatr )
      sb = sin( blatr )
c
c now compute other quantities
c
      a = cb * sin( blongr - alongr )
      b = ca * sb - sa * cb * cos( blongr - alongr )
      cd = ca * cb * cos( blongr - alongr ) + sa * sb
      sd = sqrt( a*a + b*b )
c
c compute distances
c
      rdeg = atan2( sd, cd )/ dtor
      rkm = 111.19 * rdeg
c
c compute azimuth (from a to b) and make it positive.
c
      az = atan2( a, b )/ dtor
      if ( az .lt. 0.0 ) az = az + 360.0
c
c compute back azimuth (from b to a) and make it positive.
c
      a = ca * sin( alongr - blongr )
      b = cb * sa - sb * ca * cos( alongr - blongr )
      baz = atan2( a, b)/ dtor
      if ( baz .lt. 0.0 ) baz = baz + 360.0
c
      return
      end



      SUBROUTINE locate(xx,n,x,j)
      INTEGER j,n
      REAL x,xx(n)
      INTEGER jl,jm,ju
      jl=0
      ju=n+1
10    if(ju-jl.gt.1)then
        jm=(ju+jl)/2
        if((xx(n).gt.xx(1)).eqv.(x.gt.xx(jm)))then
          jl=jm
        else
          ju=jm
        endif
      goto 10
      endif
      j=jl
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software $!6)$-"11j.

      SUBROUTINE indexx(n,arr,indx)
      INTEGER n,indx(n),M,NSTACK
      REAL arr(n)
      PARAMETER (M=7,NSTACK=50)
      INTEGER i,indxt,ir,itemp,j,jstack,k,l,istack(NSTACK)
      REAL a
      do 11 j=1,n
        indx(j)=j
11    continue
      jstack=0
      l=1
      ir=n
1     if(ir-l.lt.M)then
        do 13 j=l+1,ir
          indxt=indx(j)
          a=arr(indxt)
          do 12 i=j-1,1,-1
            if(arr(indx(i)).le.a)goto 2
            indx(i+1)=indx(i)
12        continue
          i=0
2         indx(i+1)=indxt
13      continue
        if(jstack.eq.0)return
        ir=istack(jstack)
        l=istack(jstack-1)
        jstack=jstack-2
      else
        k=(l+ir)/2
        itemp=indx(k)
        indx(k)=indx(l+1)
        indx(l+1)=itemp
        if(arr(indx(l+1)).gt.arr(indx(ir)))then
          itemp=indx(l+1)
          indx(l+1)=indx(ir)
          indx(ir)=itemp
        endif
        if(arr(indx(l)).gt.arr(indx(ir)))then
          itemp=indx(l)
          indx(l)=indx(ir)
          indx(ir)=itemp
        endif
        if(arr(indx(l+1)).gt.arr(indx(l)))then
          itemp=indx(l+1)
          indx(l+1)=indx(l)
          indx(l)=itemp
        endif
        i=l+1
        j=ir
        indxt=indx(l)
        a=arr(indxt)
3       continue
          i=i+1
        if(arr(indx(i)).lt.a)goto 3
4       continue
          j=j-1
        if(arr(indx(j)).gt.a)goto 4
        if(j.lt.i)goto 5
        itemp=indx(i)
        indx(i)=indx(j)
        indx(j)=itemp
        goto 3
5       indx(l)=indx(j)
        indx(j)=indxt
        jstack=jstack+2
        if(jstack.gt.NSTACK)pause 'NSTACK too small in indexx'
        if(ir-i+1.ge.j-l)then
          istack(jstack)=ir
          istack(jstack-1)=i
          ir=j-1
        else
          istack(jstack)=j-1
          istack(jstack-1)=l
          l=i
        endif
      endif
```

```
      goto 1
      END
C  (C) Copr. 1986-92 Numerical Recipes Software $16)$-"11j.

      SUBROUTINE momntdmb(data,nstart,nstop,ave,adev,
     :                    sdev,var,skew,curt)

* Modified by Dave Boore on 03/18/95 so that it will
* compute the moment for array entries from nstart to nstop

      INTEGER n, nstart, nstop
      REAL adev,ave,curt,sdev,skew,var,data(*)
      INTEGER j
      REAL p,s,ep
*      if(n.le.1)pause 'n must be at least 2 in moment'
      s=0.
      do 11 j=nstart,nstop
        s=s+data(j)
11    continue
      n = nstop - nstart + 1
      ave=s/n
      adev=0.
      var=0.
      skew=0.
      curt=0.
      ep=0.
      do 12 j=nstart,nstop
        s=data(j)-ave
        ep=ep+s
        adev=adev+abs(s)
        p=s*s
        var=var+p
        p=p*s
        skew=skew+p
        p=p*s
        curt=curt+p
12    continue
      adev=adev/n
      if ( n .eq. 1) then
        var = 0.0
        sdev = 0.0
      else
        var=(var-ep**2/n)/(n-1)
        sdev=sqrt(var)
      end if
      if(var.ne.0.)then
        skew=skew/(n*sdev**3)
        curt=curt/(n*var**2)-3.
      else
*         pause 'no skew or kurtosis when zero variance in moment'
        skew = 0.0
        curt = 0.0
      endif
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software $16)$-"11j.
```

```
      Program GetAvgSA

* Reads the psv values from various sources and then compute Sa for
* each the PSV at each period and
* find the average over frequency.  I do this for two components, and
* average the components.

* Dates:  03/22/95 - written by D. Boore for use in Equipment Qualification
*                    project (done for BNL)
*         03/28/95 - extensive revision
*         04/03/95 - added computation of sigma (this required changing
*                    the *.in file) and improved summary file.
*         04/11/95 - minor changes in output format

      real sd(120), sv(120), sa(120,2), per(120), freq(120)
      real sa_corr(120,2), correct(120),
     :    avg_of_2(20), avg_of_2_corr(20)
      real avg(2,20), avg_corr(2,20), freqavg(2), m, delta(20)
      character f_in*60, f_out*12, f_rs(2,20)*12
      character f_sum*12, rs_fmt*3
      character header1*77, header2*77, buffer*77

      pi = 4.0*atan(1.0)

      nu_in = 18
      nu_out = 20
      nu_sum = 30

* Get name of file with input stuff:

      f_in = ' '
      write(*,'(a\)') ' Enter name of input file: '
      read(*, '(a)') f_in

* Open the file and start processing:

      open(nu_in, file=f_in, status='unknown')

      header1 = ' '
      read(nu_in, '(a)') header1

      f_sum = ' '
      read(nu_in, '(t2,i2, t8,f7.1, t16,f7.1, t24,a12)')
     :     nsta, dref, velref, f_sum

      read(nu_in, *)

      header2 = ' '
      read(nu_in, '(a)') header2

* Open summary file:

      open(nu_sum, file=f_sum, status='unknown')

      write(nu_sum, '(2a)') ' Summary of processing, file ',
     :   f_in

      write(nu_sum, *)

      write(nu_sum, '(a)') ' Contents of input file: '
      write(nu_sum,*)
      write(nu_sum, '(3x,a)') header1
      write(nu_sum, '(t5,i2, t11,f5.1, t19,f6.1, t27,a12)')
     :     nsta, dref, velref, f_sum
      write(nu_sum, *)
      write(nu_sum, '(3x,a)') header2

* Loop over stations:


       do ista = 1, nsta                                     ! LOOP A
         buffer = ' '
         read(nu_in, '(a)') buffer
         write(nu_sum, '(3x,a)') buffer
         f_rs(1,ista) = ' '
         f_rs(2,ista) = ' '
         rs_fmt = ' '
         f_out = ' '
         read(buffer, '(t2,a3, t13,a12, t26,a12, t39,f7.1,
     :                  t47,f9.2, t57,f7.1, t65,a12)')
     :       rs_fmt, (f_rs(i,ista),i=1,2), dsta, delta(ista),
     :       velsta, f_out

         write(*,'(a,2i5,3a)')
     :    ' begin ista loop: nu_out, ista, f_in, f_sum, f_out = ',
     :   nu_out, ista, f_in, f_sum, f_out

         do icomp = 1, 2                                      ! LOOP B


           if (rs_fmt .eq. 'BAP' .or. rs_fmt .eq. 'bap') then
             call read_bap(f_rs(icomp,ista), freq, per, nper, sd,
     :                   sv, sa(1, icomp))
           else if (rs_fmt .eq. 'BJF' .or. rs_fmt .eq. 'bjf') then
             call read_bjf(f_rs(icomp,ista), freq, per, nper, sd,
     :                   sv, sa(1, icomp))
           else if (rs_fmt .eq. 'SIL' .or. rs_fmt .eq. 'sil') then
             call read_sil(f_rs(icomp,ista), freq, per, nper, sd,
     :                   sv, sa(1, icomp))
           else
             write(*, '(3a)') ' rs_fmt = ', rs_fmt,
     :         ' and not bap or bjf or sil; quitting.'
             stop
           end if

* Change units of sa to g:

           do i = 1, nper
             sa(i, icomp) = sa(i, icomp)/980.0
           end do

* Reverse order, if needed, so that frequency increases:

           if (freq(2) .lt. freq(1) ) then
             call reorder(freq, nper)
             call reorder(per, nper)
             call reorder(sa(1, icomp), nper) ! I hope this picks out right array
           endif

* Get limits:

           call locate(freq, nper, 3.0, nlowm1)
           call locate(freq, nper, 8.0, nhigh)
           nlow = nlowm1 + 1

* Fill sa_corr with corrected sa (because of the cubic polynomial used by bjf,
* set values outside 2 to 0.1 sec to garbage that will not plot).

           do i = 1, nper
             if (per(i) .lt. 0.1 .or. per(i) .gt. 2.0) then
               correct(i) = 10000.0
             else
```

```
          m = 6.0  ! can be anything, since the correction is for same quake
          correct(i) = 10.0**(psvper_f(per(i), m, dref, velref)
     :                 - psvper_f(per(i), m, dsta, velsta))
        end if
      end do

      do i = 1, nper
        sa_corr(i, icomp) = correct(i)* sa(i, icomp)
      end do

* Now compute the averages:

        call find_avg(freq, sa(1, icomp), nlow, nhigh,
     :           avg(icomp, ista))
        call find_avg(freq, sa_corr(1, icomp), nlow, nhigh,
     :           avg_corr(icomp, ista))
        freqavg(1) = freq(nlow)
        freqavg(2) = freq(nhigh)

* Then loop back for another component

      end do                                   ! LOOP B (over components)

* Then compute average of the average and write out a column
* file that has freq, per, sa1, sa2, freq, avg1, avg2, avgavg
* that I can use in coplot.

      avg_of_2(ista) = 0.5 * (avg(1, ista) + avg(2, ista))
      avg_of_2_corr(ista) =
     :           0.5 * (avg_corr(1, ista) + avg_corr(2, ista))

      write(*, '(a, i5, 1p2e10.3)')
     :  ' ista, avg_corr1, avg_corr2 = ',
     :    ista, avg_corr(1,ista), avg_corr(2, ista)

      write(*, '(a, i5, 1p2e10.3)')
     :  ' ista, avg_of_2, avg_of_2_corr = ',
     :   ista, avg_of_2(ista), avg_of_2_corr(ista)

      open(unit=nu_out, file=f_out, recl=155, status='unknown')

      write(nu_out, 999)
999     format(t4,'freq', t12,'per', t24,'sa1', t35,'sa2',
     :        t43,'sa1_corr', t55,'sa2_corr',
     :        t64,'freqavg', t72,'peravg',
     :        t84,'avg1', t95,'avg2', t103,'avgavg',
     :        t112,'avg1corr', t123,'avg2corr', t132,'avgavgcorr')

        do i = 1,2
          peravg = 1.0/freqavg(i)
          write(nu_out, '(t2,f6.3, t9,f6.3, t16,e11.4,
     :          t27,e11.4, t40,e11.4, t52,e11.4,
     :          t65,f6.3, t72,f6.3,1p,
     :          t78,e10.3, t89,e10.3, t99,e10.3,
     :          t110,e10.3, t121,e10.3, t132,e10.3)')
     :    freq(i), per(i), (sa(i,j), j=1,2), (sa_corr(i,j), j=1,2),
     :    freqavg(i), peravg, (avg(j, ista), j=1,2), avg_of_2(ista),
     :    (avg_corr(j, ista), j=1,2), avg_of_2_corr(ista)
        end do

        do i = 3, nper
          write(nu_out, '(t2,f6.3, t9,f6.3, t16,e11.4,
     :          t27,e11.4, t40,e11.4, t52,e11.4,
     :          t64,f6.3, t71,f6.3,
     :          t77,e11.4, t88,e11.4, t99,e11.4,
     :          3(1x,e11.4))')
     :    freq(i), per(i), (sa(i,j), j=1,2), (sa_corr(i,j), j=1,2)
        end do

        write(*,'(a,i5,3a)')
     :      ' end ista loop:ista, f_in, f_sum, f_out = ',
     :      ista, f_in, f_sum, f_out

        write(*, '(a,3i5)')
     :       ' nu_in, nu_sum, nu_out = ',
     :       nu_in, nu_sum, nu_out


        close(unit=nu_out)

        end do                                   ! LOOP A (over stations)

* Write out SA for each station and component

        write(nu_sum,*)
        write(nu_sum, '(a)') ' Results of Processing:'
        write(nu_sum,*)
        write(nu_sum, '(a)') '   For each station:'
        write(nu_sum,*)
        write(nu_sum, 9669)
9669    format(7x,'FileComp1', 4x,'FileComp2',
     :     4x,'SA1', 4x,'SA2', 1x,'Avg1&2',
     :     2x, 'Corr: SA1', 4x,'SA2', 1x,'Avg1&2' )
        do i = 1, nsta
          write(nu_sum, '(4x,a,1x,a,
     :      1x,f6.3,1x,f6.3,1x,f6.3,
     :      5x,f6.3,1x,f6.3,1x,f6.3)')
     :      f_rs(1,i), f_rs(2,i),
     :      (avg(j,i),j=1,2), avg_of_2(i),
     :      (avg_corr(j,i),j=1,2), avg_of_2_corr(i)
        end do

* Compute geometric average of corrected averages
* over stations and print out various averages

        cumdelta = 0.0
        cum = 0.0
        cumcorr = 0.0
        do i = 1, nsta
          cumdelta = cumdelta + delta(i)
          cum = cum + alog10(avg_of_2(i))
          cumcorr = cumcorr + alog10(avg_of_2_corr(i))
        end do
        avg_delta_over_sta = cumdelta/nsta
        cum = cum / nsta
        avg_over_sta = 10.0**cum
        cumcorr = cumcorr / nsta
        avg_corr_over_sta = 10.0**cumcorr

        call inter_interstation_sigma(
     :      avg_delta_over_sta, nsta, sigma)

        write(nu_sum,*)
        write(nu_sum, '(a)') '   Averaged over stations:'
        write(nu_sum,*)
        write(nu_sum, 309)
309     format(3x,' AvgDSta2Ref  Sig  10^Sig',
     :      '   AvgOverStations  AvgCorrOverStations')

      ten2sig = 10.0**sigma
      write(nu_sum, 948) avg_delta_over_sta,
     : sigma, ten2sig,
```

```
     : avg_over_sta,
     : avg_over_sta/ten2sig, avg_over_sta*ten2sig,
     : avg_corr_over_sta,
     : avg_corr_over_sta/ten2sig, avg_corr_over_sta*ten2sig

948    format(6x, f5.2,
     :        5x, f4.2, 4x, f4.2,
     :        2x, f5.2,
     :        '(', f4.1, ',', f4.1, ')',
     :        2x, f5.2,
     :        '(', f4.1, ',', f4.1, ')' )

        close(unit=nu_in)
        close(unit=nu_sum)

        stop
        end

      subroutine inter_interstation_sigma(delta, nsta, sigma)
        sig_1 = 0.1817       !  BJF, random comp, M 6.0-6.9
        sigma = sig_1 * sqrt(1.0+1.0/nsta) *
     :          (1.0 - exp(-sqrt(0.6*delta)))

        return
        end

      subroutine reorder(a, n)
        real a(*)
        do i = 1, n/2
          dum = a(n+1-i)
          a(n+1-i) = a(i)
          a(i) = dum
        end do
        return
      end

      subroutine find_avg(x, y, nlow, nhigh, avg)
        real x(*), y(*)
        area = 0.0
        do i = nlow, nhigh-1
          area = area + 0.5*(y(i)+y(i+1))*(x(i+1)-x(i))
        end do
        avg = area/(x(nhigh)-x(nlow))
        return
      end

      SUBROUTINE locate(xx,n,x,j)
      INTEGER j,n
      REAL x,xx(n)
      INTEGER jl,jm,ju
      jl=0
      ju=n+1
10    if(ju-jl.gt.1)then
        jm=(ju+jl)/2
        if((xx(n).gt.xx(1)).eqv.(x.gt.xx(jm)))then
          jl=jm
        else
          ju=jm
        endif
      goto 10
      endif
      j=jl
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software $!6)$-"11j.

      subroutine read_bap(fil_name, freq, per, nper, sd, sv, sa)

* Read response spectra file made by BAP.

* NOTE: This version assumes that the spectra were computed for
* only one damping

* Also note that the array of values is inverted in order so that
* frequency increases.

* Dates: 03/27/95 - Written by D. Boore
*        03/29/95 - allow for comment lines

      real freq(*), per(*), sd(*), sv(*), sa(*), rhead(50)
      integer ihead(48)
      character fil_name*(*)

      open(unit=10, file=fil_name, status='unknown')

      call skip(10, 11)
      read(10, '(8i10)') (ihead(i), i=1,48)
      read(10, '(5e15.7)') (rhead(i), i=1,50)
      nskip = ihead(16)
      call skip(10, nskip)
      call skip(10, 1)

      read(10, '(3i5)') ndamp, nper, iflag
      read(10, '(5e10.5)') damp

      read(10, '(7e11.4)') (per(i), i=1, nper)

      read(10, *)

      read(10, '(7e11.4)') (sd(i), i=1, nper)

      close(unit=10)

      pi = 4.0*atan(1.0)

      do i = 1, nper
        freq(i) = 1.0/per(i)
        sv(i) = 2.0*pi*freq(i)*sd(i)
        sa(i) = 2.0*pi*freq(i)*sv(i)
      end do

      return
      end


      subroutine read_bjf(fil_name, freq, per, nper, sd, sv, sa)

* Read response spectra file in format used in BJF93 study.

* NOTE: This version assumes that the spectra were computed for
* only one damping

* Dates: 03/27/95 - Written by D. Boore

      real freq(*), per(*), sd(*), sv(*), sa(*)
      character fil_name*(*)


* Read the periods:

      open(unit=12,file='\psv\progs\csmip.per')
```

GETAVGSA.FOR 6-14-95 12:21p   Page 3 of 5

```
      read(12, '(8f10.3)') (per(i), i=1, 91)
      close(12)

      nper = 91

      open(unit=10, file=fil_name, status='unknown')

* Skip 32 lines:

        do i = 1, 32
          read(10, *)
        end do

* Read the psv values:

        read(10, '(7e11.4)') (sv(i), i = 1, 91)

      close(unit=10)

      pi = 4.0*atan(1.0)

      do i = 1, nper
        freq(i) = 1.0/per(i)
        sd(i) = sv(i)/(2.0*pi*freq(i))
        sa(i) = 2.0*pi*freq(i)*sv(i)
      end do

      return
      end

      subroutine read_sil(fil_name, freq, per, nper, sd, sv, sa)

* Read response spectra file made by Walt Silva.

* NOTE: This version assumes that the spectra were computed for
* only one damping

* Dates: 03/27/95 - Written by D. Boore

      real freq(*), per(*), sd(*), sv(*), sa(*)
      character fil_name*(*)

      open(unit=10, file=fil_name, status='unknown')

      do i = 1, 3
        read(10, *)
      end do

      read(10, '(t3,i3)') nper

      do i = 1, nper
        read(10, '(3x, 8(3x, e12.7))') freq(i), sd(i)
      end do

      close(unit=10)

      pi = 4.0*atan(1.0)

      do i = 1, nper
        per(i) = 1.0/freq(i)
        sv(i) = 2.0*pi*freq(i)*sd(i)
        sa(i) = 2.0*pi*freq(i)*sv(i)
      end do

      return
      end

      function psvper_f(t, m, d, v)

* Returns BJF93, 94 value for random value, 5 % damping

*     t = period
*     m = moment magnitude
*     d = distance
*     v = average shear-wave velocity

* This routine uses the cubic polynomial results for the regression
* coefficients, from Table 8 in BJF94.

* Dates:
*         03/28/95 - Written by D. Boore

      real b1_c(4), b2_c(4), b3_c(4), h_c(4), b5_c(4), bv_c(4),
     :     logva_c(4), sig1_c(4), sig2_c(4), sig4_c(4)
      real b1, b2, b3, h, b5, bv,
     :     logva, sig1, sig2, sig4
      real m, d

      data b1_c    /  1.65301,  1.87615,  -3.17713,  1.37157/
      data b2_c    /  0.32667, -0.22536,   0.64842, -0.29982/
      data b3_c    / -0.09803, -0.06168,   0.35352, -0.20739/
      data h_c     /  6.26923, 10.59215, -32.48153, 18.51690/
      data b5_c    / -0.93430, -0.09835,   0.52386, -0.28909/
      data bv_c    / -0.21172,  0.06619,  -1.35085,  0.79809/
      data logva_c/  3.04586,  1.69975,  -2.97445,  1.37668/
      data sig1_c /  0.19117, -0.05830,   0.13415, -0.05913/
      data sig2_c /  0.00266,  0.05649,   0.07367, -0.03324/
      data sig4_c /  0.08263,  0.11264,  -0.09145,  0.03751/

* Evaluate coefficients:

      call get_coeff(b1, b1_c, t)
      call get_coeff(b2, b2_c, t)
      call get_coeff(b3, b3_c, t)
      call get_coeff(h, h_c, t)
      call get_coeff(b5, b5_c, t)
      call get_coeff(bv, bv_c, t)
      call get_coeff(logva, logva_c, t)
      call get_coeff(sig1, sig1_c, t)
      call get_coeff(sig2, sig2_c, t)
      call get_coeff(sig4, sig4_c, t)

* Check for sig less than 0... this is possible because of the smoothing.
      if (sig1 .lt. 0.0) sig1 = 0.0
      if (sig2 .lt. 0.0) sig2 = 0.0
      if (sig4 .lt. 0.0) sig4 = 0.0

      sigc = sig4
      sige = sig2

      sigr = sqrt(sig1**2+ sigc**2)
      sloga = sqrt(sigr**2+sige**2)

      r = sqrt(d**2.0+ h**2.0)

      b4 = 0.0    ! in BJF93

      psvper_f = b1 + b2*(m-6.0)+b3*(m-6.0)**2.0
     :             + b4*r + b5*alog10(r)
     :             + bv*(alog10(v)-logva)

      return
```

```
end

subroutine get_coeff(b, b_c, t)

  real b_c(*), logtnrm

  logtnrm = alog10(t/0.1)

  b = 0.0
  do i = 1, 4
    b = b + b_c(i)*(logtnrm)**(i-1)
  end do

  return
end

subroutine SKIP(lunit, nlines)
  if (nlines .eq. 0) return
  do i = 1, nlines
     read(lunit, *)
  end do
  return
end
```

GETAVGSA.FOR 4-14-95 12:21p

Page 5 of 5

NRC FORM 335 (2-89) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)

NUREG/CR-6464
BNL-NUREG-52500

2. TITLE AND SUBTITLE

An Evaluation of Methodology for Seismic Qualification of Equipment, Cable Trays, and Ducts in ALWR Plants by Use of Experience Data

3. DATE REPORT PUBLISHED

MONTH	YEAR
July	1997

4. FIN OR GRANT NUMBER

L2222

5. AUTHOR(S)

K. K. Bandyopadhyay, D. D. Kana, R. P. Kennedy, A. J. Schiff

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION - NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*

Brookhaven National Laboratory
Upton, NY 11973-5000

9. SPONSORING ORGANIZATION - NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*

Division of Engineering Technology
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

R. Kenneally, NRC Project Manager

11. ABSTRACT *(200 words or less)*

Advanced Reactor Corporation (ARC) has developed a methodology for seismic qualification of equipment, cable trays, and ducts in Advanced Light Water Reactor plants. A Panel (members of which acted as individuals) supported by the Office of Nuclear Regulatory Research of the Nuclear Regulatory Commission has evaluated this methodology. The review approach and observations are included in this report. In general, the Panel supports the ARC methodology with some exceptions and provides recommendations for further improvements.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

Data
Ducts
Electric Cables
Electrical Equipment - Performance Testing
Electrical Equipment - Seismic Effects
Evaluation
Nuclear Power Plants - Electrical Equipment
Nuclear Power Plants - Seismic Effects
USNRC

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page) unclassified

(This Report) unclassified

15. NUMBER OF PAGES

16. PRICE