CALIFORNIA INSTITUTE OF TECHNOLOGY

EARTHQUAKE ENGINEERING RESEARCH LABORATORY
Center for Research on the Prevention of Natural Disasters

ROUTINE COMPUTER PROCESSING OF STRONG-MOTION ACCELEROGRAMS

by
M. D. Trifunac and V. Lee

EERL 73-03

A REPORT ON RESEARCH CONDUCTED UNDER A
GRANT FROM THE NATIONAL SCIENCE FOUNDATION

PASADENA, CALIFORNIA

October, 1973

CALIFORNIA INSTITUTE OF TECHNOLOGY
EARTHQUAKE ENGINEERING RESEARCH LABORATORY

ROUTINE COMPUTER PROCESSING OF STRONG-MOTION ACCELEROGRAMS

by

M. D. Trifunac and V. Lee

EERL 73-03

A Report on Research Conducted Under a Grant
from the National Science Foundation

Pasadena, California

October, 1973

ABSTRACT

This report contains short descriptions, flow charts and listings of the computer programs that are currently being used for routine processing of strong-motion accelerograms at the Earthquake Engineering Research Laboratory of the California Institute of Technology. All programs are in Fortran IV.

The programs are presented in five groups corresponding to the processing scheme developed at the Earthquake Engineering Research Laboratory for Volume I, II, III, IV, and eventually Volume V reports (Hudson, et al, 1969; Hudson, et al, 1971; Hudson, et al, 1972a; and Hudson, et al, 1972b). The programs belonging to the Volume I operation are used to check the raw digitized data, to perform the elementary corrections for time and fixed base line, and to scale the raw digitized acceleration data to seconds and G/10 (G is the acceleration of gravity = 981 cm/sec^2). The programs belonging to Volume II processing perform the instrument and base line corrections and calculate ground velocity and displacement. The programs for Volume III processing calculate the true velocity spectra, the Fourier amplitude spectra and the pseudo velocity spectra, while the programs for Volume IV compute the Fourier amplitude spectra using the Fast Fourier Transform approach. On the completion of the various stages of processing the output data are plotted, tabulated, punched on cards, and stored on magnetic tapes. The programs for Volume V processing at present plot the RES curves on linear and logarithmic scales by reading

the Volume V tape containing the corrected accelerogram and the RES amplitudes.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

INTRODUCTION

This report has been prepared in response to numerous requests to supply information on routine processing of strong-motion accelerograms. It brings together all routine programs that deal with the scaling and correcting of raw acceleration data; the calculations of corrected acceleration, velocity and displacements; and the computation of response and Fourier amplitude spectra.

The programs presented herein were not prepared by a professional team of computer programmers but rather by a group of research workers in earthquake engineering who have only a modest working knowledge of Fortran IV. For this reason the programming style and the extent and detail presented in the comment statements within the body of each program vary considerably from one program to another. The contributors to this work are therefore somewhat reluctant to publish a report of this kind because, in their opinion, numerous problems might be encountered by a user who is not thoroughly familiar with the limitations of various programs and the details of their usage. On the other hand, it has been obvious that there is a need for a systematic approach to processing the data in strong-motion seismology and earthquake engineering and that the experience gathered at the Earthquake Engineering Research Laboratory during the past several years might be useful to other investigators in the field. Thus it is hoped that this report will serve as a logical addition to the other Earthquake Engineering Research Laboratory publications dealing with the subject of data handling, accuracy, evaluation, and final processing. It must be kept in mind, however, that as new methods and better equipment are developed in the future, there will be an increase in the accuracy and speed of digitization and the amount of information that can be retrieved from each time record. As we learn of better and more efficient ways to process the strong-motion data, it will be necessary to modify and update the programs presented here.

The body of this report is divided into five sections, each section dealing with programs used for processing the data for Volumes I, II, III, and IV published by the Earthquake Engineering Research Laboratory. Volume V, still in the planning stage, is designed to present Response Envelope Spectra (RES), a by-product of the computations leading to the Response Spectra of Volume III (Trifunac, 1971). The summary of all these operations is shown schematically in Figure 1. As seen from this figure, in addition to the printouts and plots required for the reports, all the data are stored on magnetic tapes. This allows quick access to a desired data file and at the same time represents an economical way for preparing archive copies of processed accelerograms, response spectra, and Fourier amplitude spectra.

Each section begins with a brief outline of the key procedures for the processing of the corresponding Volumes. Since the detailed description of the relevant theory has already appeared in the previous reports (IA, IIA, IIIA, and IVA), the emphasis in these introductory sections has been on the description of the output plots, printouts, cards, and tapes. Only the essential background information on the methods of data processing has been extracted

STANDARD ACCELEROGRAM PROCESSING

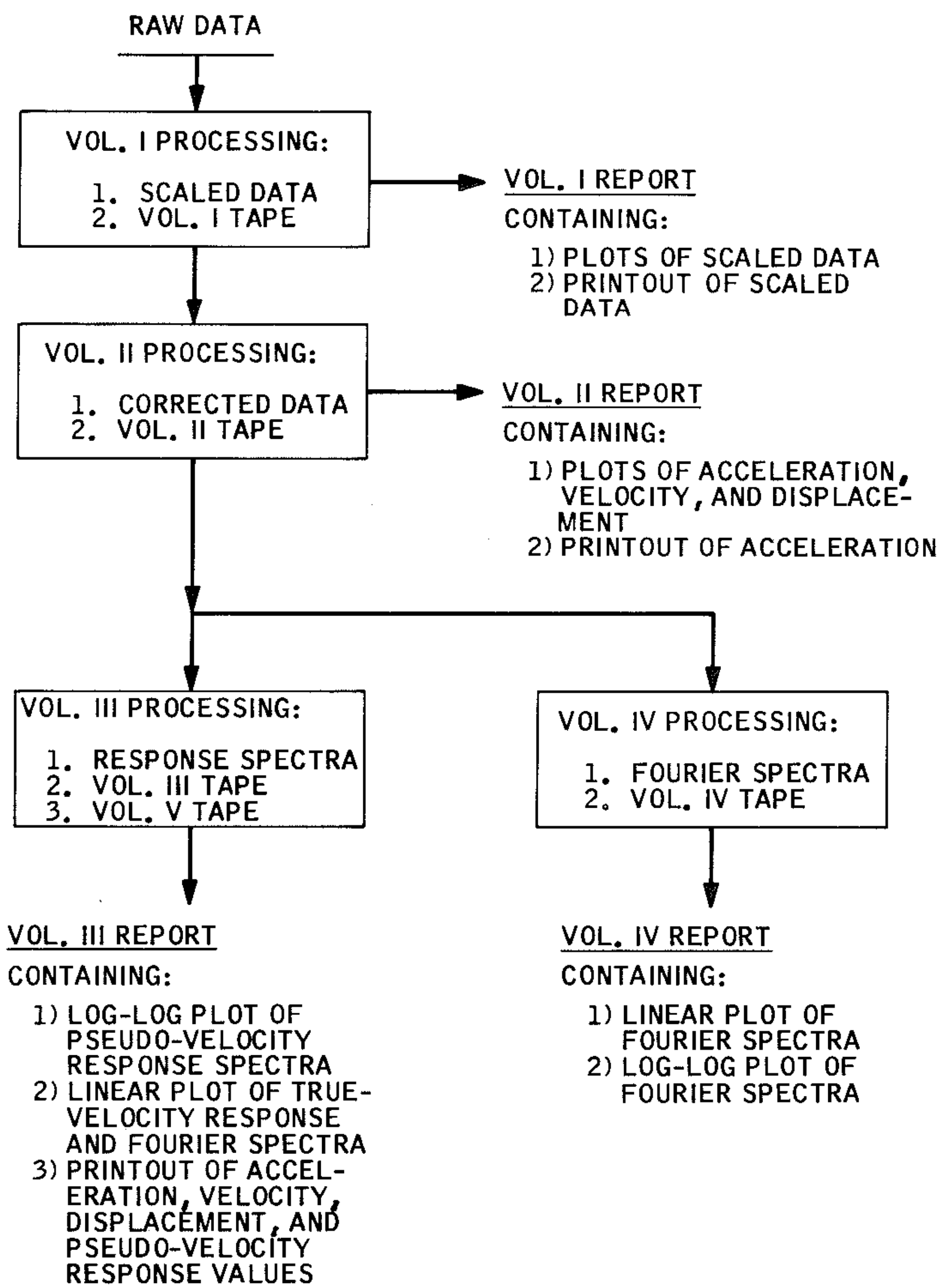


Figure 1

from the introductions of these four volumes (IA, IIA, IIIA, and IVA) to facilitate the use of this report.

Apart from the flow charts and the listings of the input parameters, we are not giving a detailed explanation of the computer programs whose listings are contained in this report. This style of presentation is motivated by the hope that all the users will critically study the programs in detail before using them. In this way we expect that a mechanism can be formed whereby all the programs are critically examined, improved, optimized for their application on different computing facilities, and further developed.

Organizational details of all MAIN programs, which are *presented in this* report, depend on the desired input-output formats. For this reason it is not practical to write their most general versions, which could handle all possible tasks. Thus, no attempt has been made to explain all the details of the particular programs presented. Only some of the key parameters governing the input-output procedures are identified and explained. All other details will have to be interpreted and modified, if necessary, by the individual users to suit their particular processing requirements.

The programs in this report are written for the IBM 370/155 computer that was in use at the time at the Willis H. Booth Computing Center of the California Institute of Technology, since updated to 370/158. While the subroutines that perform data reduction and analysis generally do not depend on the computer used, *the input-output routines presented here are very* much installation-dependent. These subroutines will have to be modified for use on other hardware. To aid the user in such modification we are listing in the Appendix the write-ups and the programs of the input-output subroutines that are, to the best of our knowledge, specific to the system used at Caltech. From this Appendix an experienced programmer should be able to understand the task these subroutines perform and to either modify them for use on other installations or to find the analogous subroutines from those available locally. Then, with only slight modifications, all input-output subroutines in this report can be used at most installations.

The present report completes a phase of a data processing project which has been made possible by the long-range support of the National Science Foundation. The cooperative efforts of many individuals are necessary for such an undertaking, and the Earthquake Engineering Research Laboratory has been unusually fortunate in the devotion of many persons who have played a significant role in the success of this venture. Special gratitude is expressed to Drs. A. G. Brady, F. E. Udwadia and A. Vijayaraghavan, Mr. J. Justiss and Mr. R. C. Dullien with whose able cooperation many programs have been successfully written. Specific contributions are indicated in the text of the report. Professor D. E. Hudson provided continuous help and encouragement during all phases of this work. We are indebted to Barbara A. Zimmerman for critical reading of the manuscript and numerous excellent suggestions that lead to the summary presented in the Appendix. Dr. T. C. Hanks also read the manuscript and offered numerous valuable suggestions. The Earthquake Engineering Research Laboratory group would like to pay a special tribute to the excellent backup that this project received at every stage from the *Willis H. Booth Computing Center* at the California Institute of Technology. Without the fine facilities provided by the Center and the very extensive assistance of the staff, in particular the generous time and effort spent by Mrs. Edith Huang, a project of this scope would have been impossible to carry out.

DATA PROCESSING FOR VOLUME I:
SCALED UNCORRECTED DATA AND VOLUME I TAPE

The Volume I data consists of digitized accelerograms of strong earthquake ground motions as processed from records obtained from strong-motion instruments. No base-line, instrumental corrections, or adjustments have been made so that the data may be regarded as "uncorrected". This digitized data is thus believed to be as close a representation of the original raw information as it is feasible to achieve with a digital process.

As a first step, full size contact film negatives are prepared from the original records which were recorded in the field on photographic paper or film. From these film negatives, contact prints or enlargements (for 70 mm and 35 mm film only) are made on a frosted, translucent, Mylar-based film. Measurements have shown that these prints differ in size from the film negatives by less than 0.1%, and the distortion involved in going from the original paper record to the contact negative is believed to be no larger than this. The translucent film is mechanically strong, dimensionally stable, and affords excellent optical contrast for setting the cross-hairs of the digitizing machine on a back-lighted glass table.

The digitizing is performed on a Benson-Lehner 099D data reducer unit. The cross-hairs are manually set to successive x-y coordinates on the record trace. The coordinates are converted to digital position figures by means of a magnetic readout head, and are stored in a 6-digit accumulator system from which they are automatically read out to a card punch. The maximum resolution of the system is about 800 digital counts per inch, corresponding to a least-time interval on most records of 1/300 seconds. The 24-inch table length on the digitizer can accommodate 10-30 seconds of record depending on the recording speed and the enlargment used. The film record is placed on the digitizing table with the horizontal axis lined up by eye to an estimated zero axis. All traces are digitized without moving the record on the table. Records longer than 24 inches have to be repositioned by displacing them in the horizontal direction, so that a vertical line is required to be drawn at the end of the first segment. In the subsequent set-up this line is positioned to remain vertical to preserve the original coordinate system defined in the first set-up. For those records requiring repositioning, the points in the record at which changes have occurred are noted using an arrow on the plotted accelerogram and an asterisk in the print-out.

The records are digitized on an unequal time basis, since this leads to the best definition of the trace for a given number of data points. All significant peaks, points of inflection, etc., are picked, along with as many intermediate points as are needed, for an accurate definition of shape. The average number of points per second of record in the most rapidly oscillating sections of the accelerograms varies from 30 to 50.

The digitized data are directly punched on cards. The program P1CHECK reads these cards and checks whether the time coordinates monotonically increase. It also searches for possible disproportionate jumps of the amplitude data. If any error is found, the program prints out the message. Small errors are corrected immediately and the corrected cards are punched out. The data are then plotted to the same scale as the digitized record, and the two versions are compared to check the accuracy of digitization. Any portion that is digitized improperly has to be redigitized and replotted until the final plot agrees well with the digitized record.

The program P2SCALE is next used to scale the data to units of seconds and G/10 (G = 981 cm/sec^2). Information about the station, the earthquake, the sensitivity of the digitized record, etc., is also read in with the acceleration data. Once again P2SCALE checks the data for increasing time coordinates. Then the subroutine SCAL is used for the following:

1) The timing marks are smoothed by 1/4, 1/2, 1/4 running average and used to scale the digitized time coordinate.

2) The fixed trace is smoothed and subtracted from the acceleration trace. Most records contain several traces produced by "fixed" mirrors rigidly attached to the accelerograph frame. In some cases these fixed traces depart measurably from straight lines, usually involving long-period components ascribed to paper distortion, motions of the paper in the drive mechanism, etc. For all records on which fixed traces are present, the fixed traces are digitized at intervals of the order of one-half second, smoothed by a weighted averaging over every three consecutive points, and subtracted from the accelerometer traces. The smoothing coefficients differ slightly from the usual 1/4, 1/2, 1/4 for equi-spaced data in that they are weighted according to their actual distances from the mid-point, with the sum of the weights remaining unity.

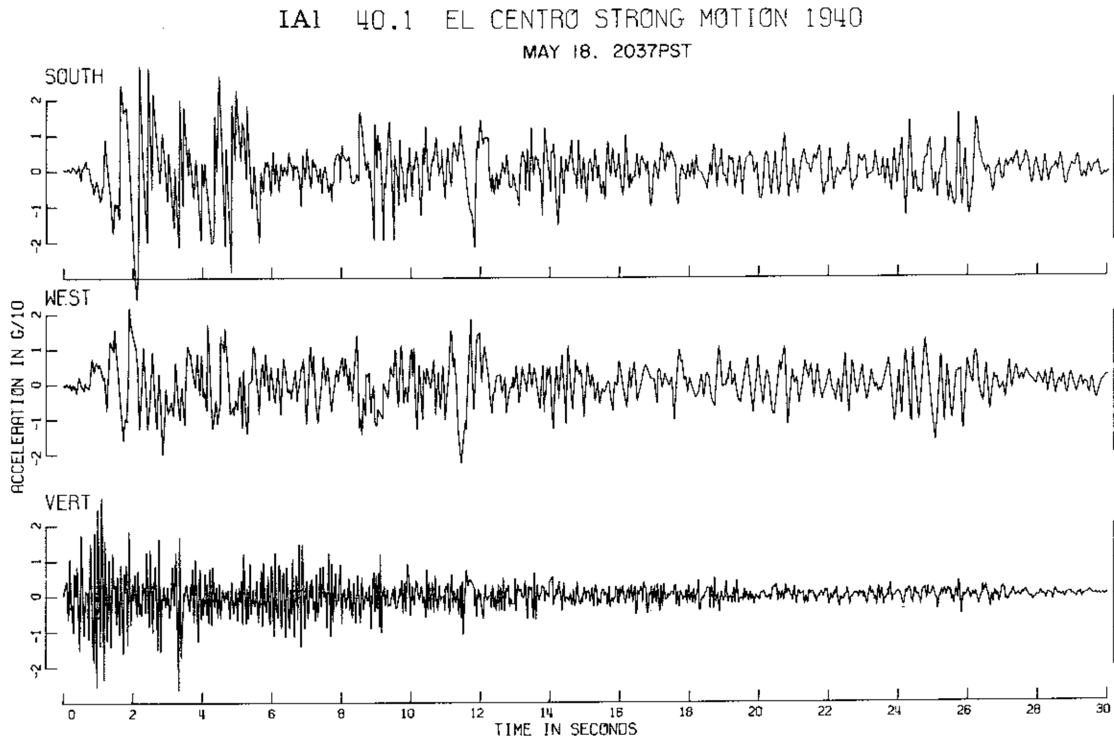
3) The acceleration trace is then scaled and a horizontal line is fixed at the zero mean level as described in the following. To fix the particular values of the digitized ordinates, some more-or-less arbitrary decision must be made as to the position of a straight reference line. When the record is placed on the table of the digitizing machine, it is lined up with the horizontal axis of the machine as closely as can be judged by eye. For this purpose the fixed traces serve as useful guides, as do the zero trace sections at the beginning of the record before the triggering of the instrument. It must be realized, however, that almost imperceptible shifts of the axis in translation or rotation can lead to large deviations in displacement curves. Therefore, some technique which assures a uniform result is needed. For this purpose, the following procedures have been adopted. If the record trace and the fixed mirror traces have been digitized without moving the record on the table of the digitizing machine, then the subtraction of the two traces will correct for any slight rotation of the record on the table, so that only a translation of the zero line is required. The zero axis is translated until the integral of the digitized acceleration curve over the length of the record is zero. This is in principle the same as making the mean zero, or making the sum of the squares of the deviations from the zero line a minimum. Physically this means that the change in ground velocity from beginning to end of the record is zero. For any individual earthquake record this assumption cannot be justified, but it is nevertheless the most logical choice on which to base a standard procedure.

For those records for which fixed trace lines are not available, or for which the record has been moved on the table between the digitizing of the record trace and the fixed trace, the base-line is not only first translated to make the mean zero as above, but then a very small rotation is introduced to make the sum of the squares of the deviations from the zero line a minimum. This removes the effects of any slight rotational misalignment without interfering with the basic data. In all cases the minimum RMS acceleration values are recorded as a significant parameter of the record. It should be noted that this is always the RMS value calculated over the entire length of the digitized record, which may be considerably longer than the strong-motion portion of the record.

It is believed that the above data processing techniques represent the minimum interference with the basic data and that the digitized data so obtained may legitimately be referred to as the basic "uncorrected" data.

At the end, P2SCALE prints out the data (Figure 3; only the first 16.925 seconds of the scaled data for the SOUTH component are reproduced in this figure) and plots the data (Figure 2) in a form suitable for publishing in a Volume I report. The scaled data are also punched out for subsequent loading onto the Volume I tape.

The P3TAPE program loads any set of scaled data onto the Volume I tape. For this operation each accelerogram must be accompanied by its heading data containing all pertinent information on the station, the earthquake, and the instrument characteristics. The heading data are read, reproduced, and augmented by the programs of Volumes II, III, and IV. For this reason ample space in


IA1 40.1 EL CENTRO STRONG MOTION 1940
MAY 18. 2037PST
SOUTH
WEST
VERT
ACCELERATION IN G/10
TIME IN SECONDS

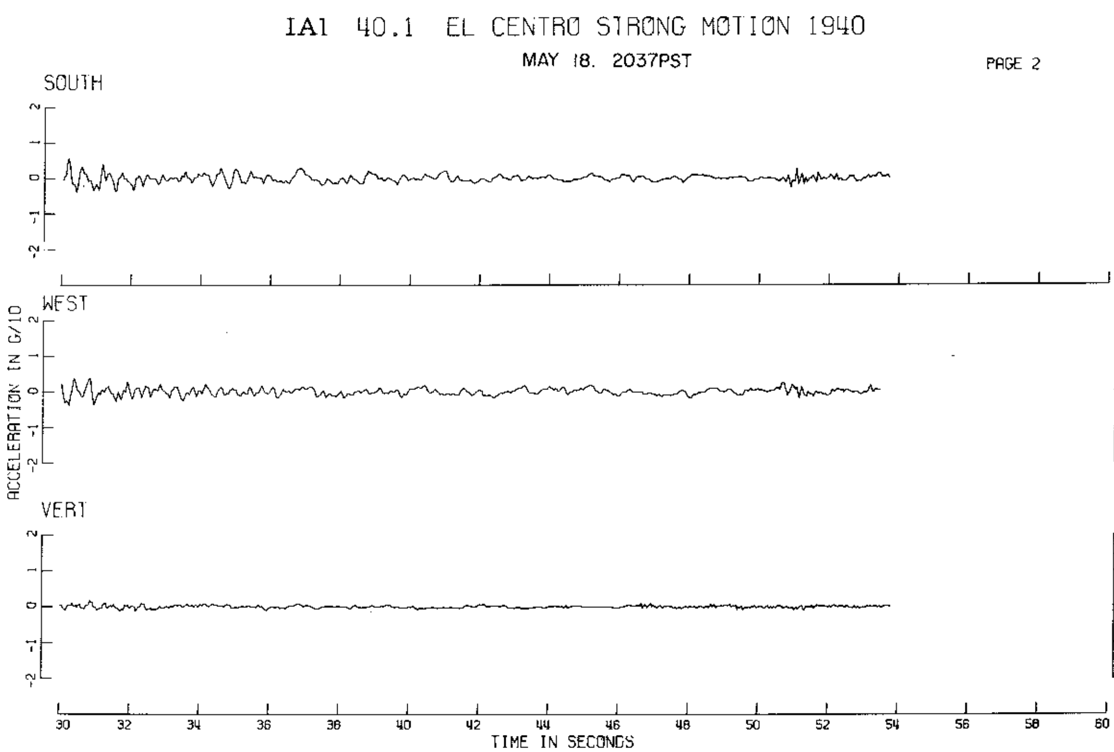

IA1 40.1 EL CENTRO STRONG MOTION 1940
MAY 18. 2037PST
PAGE 2
SOUTH
WEST
VERT
ACCELERATION IN G/10
TIME IN SECONDS

Figure 2

IA 1

1940 EL CENTRO STRONG MOTION. BY TRIFUNAC. 54 SEC. SOUTH

985 POINTS 53.732 SECONDS

SCALED DATA, WITH THE DIGITIZED FIXED TRACE SUBTRACTED
AND A STRAIGHT ZERO LINE FIXED FOR MINIMUM RMS ACCLN

UNITS ARE SEC, G/10.

THE RMS ACCELERATION OF THIS COMPLETE RECORD, MINIMIZED
IN THE ABOVE ALIGNMENT CORRECTION, IS 0.4876 G/10.

TIME	ACCLN	TIME	ACCLN	TIME	ACCLN	TIME	ACCLN
0.0	0.0589	0.093	0.0136	0.144	0.0847	0.215	-0.0035
0.273	0.1462	0.390	-0.0422	0.454	0.1801	0.520	-0.2506
0.554	0.0833	0.593	0.1153	0.653	0.3066	0.680	0.1068
0.744	0.0970	0.795	-0.1932	0.813	-0.3206	0.849	-0.3504
0.894	-0.6194	0.976	-0.2664	1.034	-0.5795	1.078	-0.8412
1.118	-0.6855	1.141	-0.4597	1.170	-0.3943	1.216	0.9063
1.291	0.0428	1.369	-0.6734	1.449	-1.7497	1.504	-0.8830
1.535	-0.9878	1.578	-0.9248	1.629	-1.3280	1.672	2.4222
1.754	1.6539	1.828	1.7918	1.908	0.5823	2.016	-1.6403
2.079	-2.9269	2.137	-3.5905	2.168	-2.9152	2.188	-1.6810
2.208	2.9646	2.315	0.1915	2.435	-2.0050	2.458	2.9107
2.568	-0.5082	2.600	2.1576	2.804	-0.3328	2.871	1.0759
2.944	0.2398	3.023	-0.8342	3.052	0.5117	3.121	-0.4076
3.205	-1.5934	3.223	0.0961	3.347	-2.1231	3.365	2.0124
3.466	-0.9699	3.486	1.7958	3.620	-0.0961	3.634	0.6600
3.716	-0.0510	3.776	-0.7597	3.835	0.3366	3.897	-0.5600
3.980	-1.9299	4.013	0.2259	4.063	-0.4353	4.091	0.2318
4.182	-0.7659	4.249	-1.6054	4.279	-2.0145	4.332	-1.9897
4.363	-1.4131	4.379	1.5346	4.435	-0.0125	4.501	2.6757
4.587	0.6898	4.681	-2.0954	4.708	0.6394	4.855	-2.8201
4.882	1.8694	4.968	0.3269	4.995	2.2638	5.133	0.2942
5.167	1.3868	5.220	1.1336	5.296	-0.2270	5.307	1.8312
5.456	-1.0347	5.479	-0.5805	5.515	0.1566	5.598	-1.0850
5.663	-2.0148	5.694	-0.8222	5.717	-0.0231	5.782	-0.5750
5.830	0.2681	5.859	0.1432	5.901	0.2835	5.944	0.0468
6.001	-0.6691	6.046	0.1929	6.060	0.5627	6.115	-0.3613
6.149	0.2130	6.243	-0.5968	6.257	-0.1974	6.292	0.2175
6.346	-0.1321	6.383	0.1295	6.466	-0.1038	6.518	0.5431
6.541	0.2743	6.617	0.0981	6.638	-0.2479	6.670	0.1896
6.696	0.0694	6.752	0.1549	6.795	-0.0378	6.842	-0.3815
6.877	-0.9701	6.921	0.4543	6.988	-0.2379	7.047	0.6467
7.109	-0.0764	7.159	-0.4784	7.209	0.3598	7.238	0.1313
7.274	0.2217	7.312	0.0100	7.371	-0.2980	7.421	0.1014
7.474	-0.6097	7.512	-0.0391	7.558	0.1368	7.564	-0.0059
7.592	-0.0273	7.670	-0.5851	7.722	-0.3130	7.765	-0.8386
7.777	-0.3630	7.808	-0.3096	7.834	0.2338	7.883	0.4619
7.926	0.5116	7.967	0.4711	7.996	-0.3042	8.049	0.7456
8.074	0.4613	8.108	0.3744	8.187	0.2850	8.234	0.4109
8.274	0.1682	8.325	-0.3135	8.381	-0.3304	8.435	-0.2603
8.469	-0.0749	8.530	-0.2606	8.537	1.0877	8.570	1.6513
8.632	1.1517	8.745	0.0229	8.793	0.2855	8.837	-0.2080
8.866	0.0177	8.928	-1.0217	8.976	-1.9421	8.998	1.3277
9.058	0.0945	9.081	1.0302	9.128	0.5747	9.151	0.7934
9.195	-0.2496	9.245	-1.9537	9.272	-0.3319	9.305	-0.4544
9.388	1.1232	9.418	1.3799	9.502	-0.5870	9.541	-1.9296
9.561	0.4449	9.648	-0.1998	9.679	-0.9610	9.715	0.9011
9.749	0.2079	9.803	-0.0622	9.831	-0.5057	9.876	-0.8697
9.912	-0.4144	9.938	0.1219	9.957	-0.0328	9.994	0.6865
10.067	-0.7096	10.079	-0.0378	10.101	0.1548	10.130	0.0573
10.183	0.5589	10.240	0.1604	10.271	-0.3628	10.322	-1.2475
10.350	-0.4890	10.384	0.6631	10.418	0.1827	10.454	1.2623
10.521	-0.1018	10.567	-0.5572	10.606	0.0621	10.658	0.4080
10.686	0.2855	10.756	0.9620	10.814	0.1449	10.832	-0.1322
10.851	0.0211	10.890	0.1234	10.906	0.1518	10.912	0.5978
10.994	-0.5571	11.016	-0.7687	11.083	0.6652	11.191	-0.4433
11.226	0.4044	11.264	0.8372	11.378	0.2603	11.404	0.6289
11.435	0.8524	11.454	1.2745	11.514	0.6559	11.576	0.2765
11.651	-0.5369	11.719	-1.0696	11.796	-1.2707	11.821	-1.5836
11.864	-2.1448	11.895	-0.4801	11.936	0.8493	11.976	0.6446
12.024	1.4353	12.104	0.7444	12.130	0.8942	12.154	0.9381
12.195	0.9274	12.239	0.9440	12.251	0.8644	12.280	-0.0155
12.289	-0.2724	12.316	-0.2285	12.348	-0.5829	12.372	-0.1049
12.425	-0.6567	12.490	-0.0397	12.529	-0.5153	12.550	-0.2335
12.589	-0.3797	12.610	-0.0873	12.652	0.0588	12.661	0.2420
12.733	-0.0970	12.787	0.4630	12.833	-0.0720	12.854	-0.3990
12.878	-0.4667	12.905	-0.4726	12.947	-0.4240	12.962	-0.0922
12.995	-0.1932	13.046	-0.5440	13.082	-0.5736	13.100	-0.7948
13.127	-0.8270	13.142	-0.9888	13.169	-0.0577	13.211	0.3526
13.244	0.5370	13.256	0.4905	13.286	0.3026	13.313	0.2170
13.337	-0.5618	13.376	0.6249	13.445	-0.5797	13.493	1.1992
13.525	0.2218	13.561	-0.1254	13.586	0.1933	13.614	0.4287
13.639	0.3360	13.673	0.3062	13.701	0.0755	13.728	0.0458
13.748	-0.1861	13.784	-0.3884	13.812	-1.2480	13.879	1.2074
13.932	0.6437	13.983	0.0944	14.022	-0.1625	14.047	-0.5941
14.072	0.5224	14.105	0.5580	14.128	0.7494	14.172	0.2345
14.192	-0.5205	14.236	-1.1935	14.256	-1.5421	14.292	-1.0248
14.314	-0.2222	14.328	-0.0153	14.351	0.3996	14.432	-0.6529
14.459	0.0177	14.473	0.0450	14.487	0.3150	14.539	-0.0418
14.569	0.2030	14.601	0.3528	14.648	0.8711	14.677	0.3264
14.727	-0.0826	14.777	-0.1529	14.806	-0.5121	14.850	0.8553
14.885	0.2489	14.938	-0.5029	15.002	0.3758	15.038	-0.2141
15.086	-0.8932	15.127	-0.1964	15.145	0.2149	15.181	0.3339
15.214	0.1697	15.237	0.0152	15.258	-0.2976	15.324	0.6051
15.401	-0.2177	15.428	0.1664	15.503	-0.6587	15.560	-0.2877
15.617	-0.8120	15.656	0.2166	15.695	0.8422	15.732	0.2773
15.780	-0.1662	15.837	0.6330	15.922	-0.2029	15.970	0.1407
15.988	0.2062	16.013	0.4820	16.057	-0.0721	16.095	-0.4823
16.134	-0.6845	16.172	-0.1590	16.204	1.0111	16.238	0.4866
16.312	-0.4979	16.334	-0.0852	16.375	-0.3683	16.417	-0.1864
16.464	-0.1662	16.530	0.4688	16.604	-0.2269	16.631	0.2595
16.682	0.4164	16.709	0.2190	16.773	0.1596	16.786	-0.0473
16.834	0.2476	16.868	-0.1663	16.911	-0.9153	16.925	-1.0057

Figure 3

various arrays of the heading data is left unused to allow the addition of significant information during the later stages of processing or analysis. The set-up of one file of the Volume I tape is as follows:

<u>Volume I Tape</u>
(one file per one acceleration component)

Each file has:

1. Heading data of alphanumeric type
2. Heading data of integer type
3. Heading data of floating point type
4. Scaled data sequence (time, acceleration)
5. EOF

Tape parameters: 1600 bpi, LRECL=1204, BLKSIZE=3616, RECFM=VBS. The detailed description and a sample of the heading data set are given in the following section.

VOLUME I HEADING DATA

Input Card Number	Heading Data Array (CARMOD(I), I=I1,I2) used in Volume II MAIN Program I1 - I2	Format (20A4)	Description*
1	1 - 20	"	File identification
2	21 - 40	"	Name of the earthquake
3	41 - 60	"	Date & time of the earthquake
4	61 - 80	"	Volume reference and log no. of the accelerogram
5	81 - 100	"	Station no. & coordinates of the accelerograph station
6	101 - 120	"	Location of the station
7	121 - 140	"	Component direction (pendulum motion for the upward deflection of trace on the record)
8	141 - 160	"	Full title of the earthquake
9	161 - 180	"	Epicenter of the earthquake
10	181 - 200	"	Natural period; damping & sensitivity of the instrument
11	201 - 220	"	No. of points & duration of data
12	221 - 240	"	Units of data (G=981 cm/sec^2)
13	241 - 260	"	RMS of data

* The alphanumeric information on each card above ranges from column 1 - 72. Integers in columns 73 and 74 from the 4th card on dictate the number of letters on each card.

	IR(I) I	Format (I4)	
14			
	1	"	File number on tape
	2	"	Volume stage
	3	"	Part number
	4	"	Volume reference number
	5	"	Year of the record
	6	"	Log no. of the record
	7	"	Event no. of the record: 0 for Main Shock; 1, 2, 3, ... for the aftershocks
	8	"	Source of digitization: 1-7: negative, positive, transparency, special, original, enlarged negative, enlarged positive
	9	"	Station number

	IR(I)	Format (I4)	
	10, 11, 12	"	Latitude of station in degrees, minutes & seconds respectively
	13, 14, 15	"	Longitude of station in degrees, minutes & seconds respectively
	16, 17, 18	"	Latitude of epicenter in degrees, minutes & seconds respectively
	19, 20	"	Longitude of epicenter in degrees, minutes & seconds respectively
15	IR(21)-IR(30)		(seconds are in IR(21) on card no. 15)
	22, 23, 24	"	Month, day & year respectively of the earthquake
	25	"	Time of the earthquake
	26	"	Time code: 0, 1, 2, 3 for PST, PDT, MST, CST* respectively
	27	"	Component direction measured in degrees clockwise from north; vert-400, up-500, down-600
	28	"	No. of acceleration data
	29	"	No. of letters for earthquake title (card no. 2)
	30	"	No. of letters for station location (card no. 6)
	FR(1)-FR(5)		
	1	"	Natural period of the transducer in seconds
	2	"	Damping of the transducer, fraction of critical
	3	"	Duration of the data
	4	"	RMS acceleration of the data in G/10
	5	"	Units of acceleration in fractions of G

* Other time codes may be added to this list if necessary.

```
FILE   1  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-A, EERL 70-20
IMPERIAL VALLEY EARTHQUAKE
MAY 18, 1940 - 2037 PST
IA001   40.001.0 S                                                        18
STATION NO. 117   32 47 43N,115 32 55W                                    38
EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT                        50
COMP SOOE                                                                  9
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST                      52
EPICENTER  32 44 00N,115 27 00W                                           31
INSTR PERIOD = 0.0990 SEC  DAMPING = 0.552                                42
NO. OF POINTS = 985   DURATION = 53.73 SEC                                42
UNITS ARE SEC AND G/10.                                                   23
RMS ACCLN OF COMPLETE RECORD = 0.4876 G/10.                               43
   1   1   1   1  40   1   0   4 117  32  47  43-115  32  55  32  44   0-115  27
   0   5  1819402037   0 180 985  26  50   0.099   0.552  53.730   0.488   0.100
```

PROGRAMS FOR PROCESSING VOLUME I DATA

P1CHECK MAIN Program (Lee, Brady, Dullien)

This program reads in raw digitized data. It counts the number of points, checks that the time coordinate increases with time, and corrects the time coordinates if necessary. At the end, this program plots the raw digitized data to the same scale as the original digitized record.

The format (10(1X, F7.0)) (in the line labeled MAIN 2 in the listing of the MAIN program) is used to read the raw digitized data. This format together with the scaling factors 792.0 and 790.8 (in MAIN 40 and MAIN 41) is determined by the 099D Benson-Lehner data reducer in use at Caltech. These parameters, of course, might have to be changed for the reading and scaling of raw data digitized by other installations.

Some of the key parameters in this program are:

NDKS is the number of decks (components of acceleration data) with or without the corresponding fixed traces.

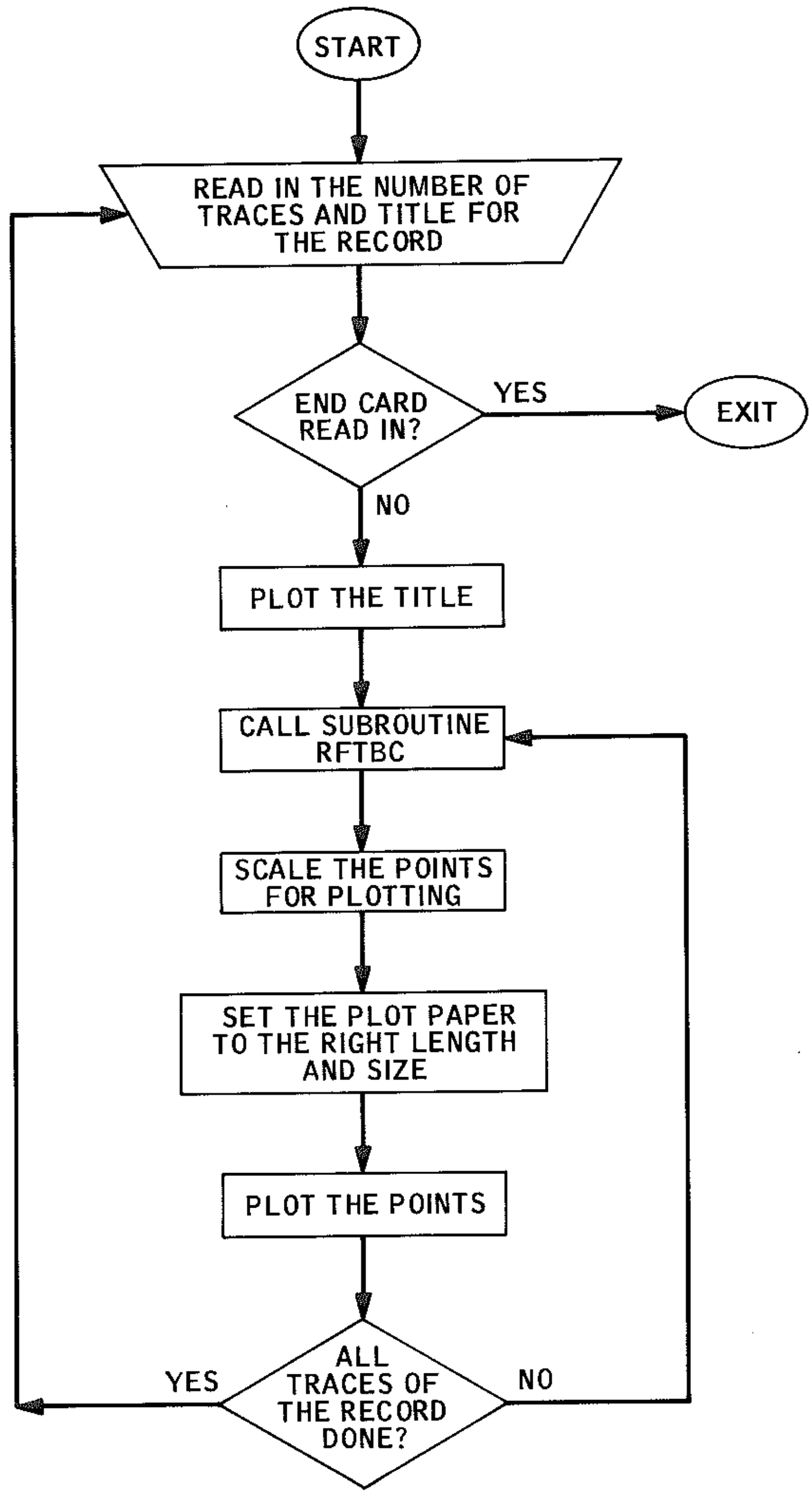
NFTM is selected from the following table:

NFTM	Explanation
-1	no time marks, no fixed traces available for this record.
0	one set of time marks, no fixed traces. Use the time marks in place of one fixed trace.
1	no time marks, one fixed trace available.
2	one set of time marks, one fixed trace available
3	no time marks, one fixed trace available for each acceleration component.
4	one set of time marks, one fixed trace available for each acceleration component.

TITLE is used to identify each record (19 words of 4 characters each).

Each continuous sequence of data, whether time marks, acceleration, or fixed traces, is followed by a card containing 'bb666666'.

PICHECK MAIN FLOW CHART



```
      DIMENSION SCFTS(7),T(8000),A(8000),DD(3),TITLE(19),FMT(4)
      DATA FMT/'(10(','1X,F','7.0)',')    '/,NPPC/5/,NPIG/0/
      DATA SCFTS/9.5,7.0,7.5,4.0,4.0,1.0,1.0/
C  ********   THE NUMBER OF THE POINTS IN A CARD IS FIXED    ********
C  NFTM, NUMBER OF FIXED TRACES(NFT)+NUMBER OF TIMING MARKS DECK(NTM)
C   NFTM     NTM    NFT
C    -1       0      0     NO TM, NO FT
C     0       1      0    ONE TM, NO FT, WITH TM USED AS FT
C     1       0      1     NO TM, ONE FT
C     2       1      1    ONE TM, ONE FT
C     3       0      3     NO TM, FT IN EVERY COMPONENT
C     4       1      3    ONE TM, FT IN EVERY COMPONENT
      DO 4 NRCD=1,99
      READ(5,22,END=99)NDKS,NFTM,TITLE
   22 FORMAT(I1,I2,19A4)
      IF(NDKS.EQ.0)NDKS=3
      WRITE(6,51)NRCD,TITLE
   51 FORMAT(1H1,'RECORD # ',I4,' = ',19A4,///)
      NTM=1-MOD(IABS(NFTM),2)
      NFT=IDIM(NFTM,NTM)
      CALL SYSSYM(1.,9.00,.15,TITLE,76,0.)
      NDKS=2*NDKS+1
    5 DO 1 NDK=1,NDKS
      NREAD=MOD(NDK,2)
      IF(NREAD.EQ.0)GO TO 8
      IF(NDK.GT.1)GO TO 7
      IF(NTM.EQ.0)GO TO 1
      GO TO 8
    7 IF(3*NFT .LT.NDK)GO TO 1
    8 CONTINUE
      WRITE(6,50) NRCD,NDK
   50 FORMAT(/,2X,'RECORD # ',I2,', TRACE #',I2//)
      K1=1
      CALL RFTBC(NP,T,A,NPPC,NDK,FMT,NPIG,K1)
      IF(K1.EQ.2)WRITE(6,52)
   52 FORMAT(' ERROR FOUND NOT CORRECTED. CHECK.')
      WRITE(6,525)NP
      IF(NREAD.EQ.0.OR.NDK.EQ.1)A1=A(1)/790.8
      DO 14 I=1,NP
      T(I)=T(I)/792.
      A(I)=A(I)/790.8+SCFTS(NDK)-A1
   14 CONTINUE
      CALL SYSXMX(T(NP)+15.)
  525 FORMAT(/2X,'NO. OF POINTS =',I6////)
      IF(NDK.NE.1)GO TO 30
      DO 28 I=1,NP
      CALL SYSPLT(T(I),A(I),3)
   28 CALL SYSPLT(T(I),A(I)-.1,2)
      IF(NFTM.NE.0)GO TO 1
      DO 29 I=1,NP
   29 A(I)=A(I)-2.
   30 CALL SYSPLT(T(1),A(1),3)
      DO 31 I=2,NP
   31 CALL SYSPLT(T(I),A(I),2)
    1 IF(NDK.EQ.NDKS)CALL SYSEND(-1,0)
    4 CONTINUE
    2 FORMAT (I5)
   99 STOP
      END
```

Subroutine RFTBC (Lee, Brady)

RFTBC is called by Volume I P1CHECK MAIN program to read in a trace of raw digitized data, count the number of points, check that the points have increasing time and behave well, and make corrections if necessary.

Usage

CALL RFTBC(NP, T, A, NPPC, NDK, FMT, NPIG, K1)

Where

NP = number of points of the trace to be read in

T(I) = time coordinates of the trace to be read in

A(I) = acceleration coordinates of the trace to be read in

NPPC = number of points per card

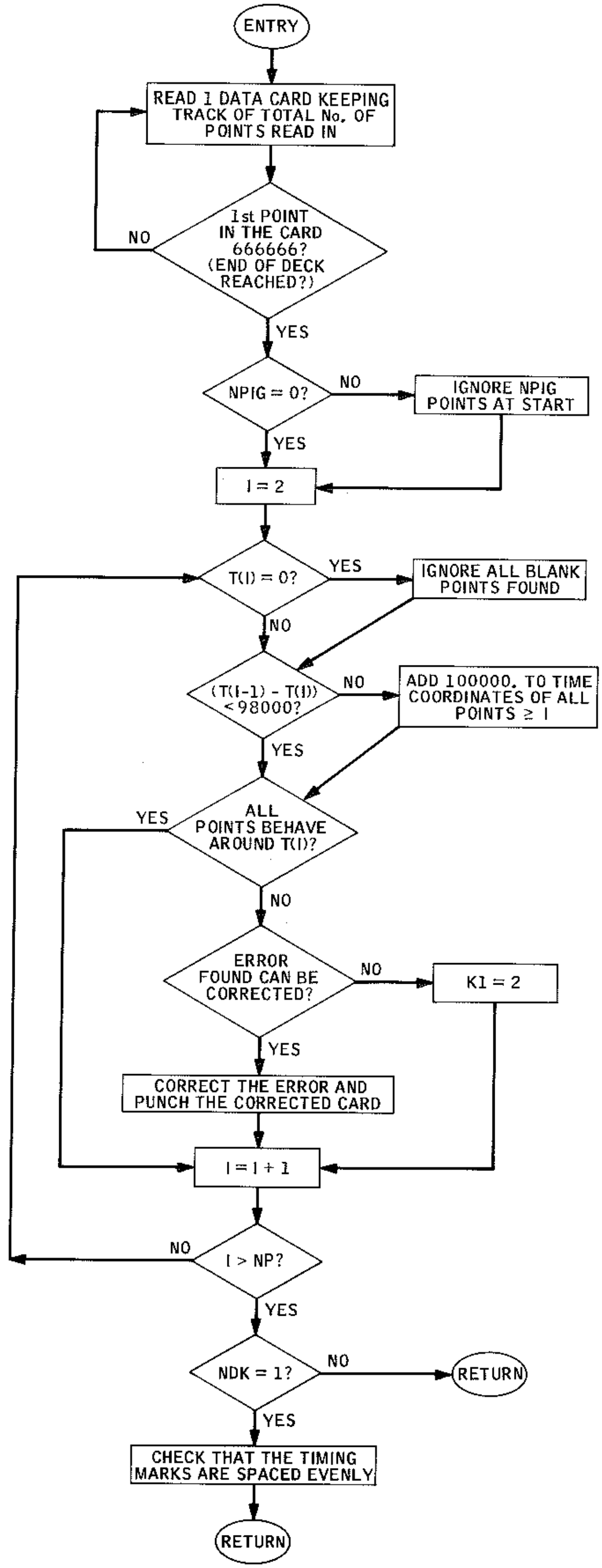
NDK = the number of the trace to be read in

FMT = format to be used in reading the data

NPIG = number of points to be ignored at start

$K1 = \begin{cases} 1 & \text{initially} \\ 2 & \text{if serious error is found} \end{cases}$

SUBROUTINE RFTBC FLOW CHART



```
      SUBROUTINE RFTBC(NP,T,A,NPPC,NDK,FMT,NPIG,K1)
      DIMENSION T(1),A(1),FMT(1)
C     READ IN DATA AND COUNT THE NUMBER OF POINTS
  200 N2=0
  201 NP=N2+1
      N2=NP+NPPC-1
      READ(5,FMT)(T(I),A(I),I=NP,N2)
      IF((T(NP)-666666.))201,203,201
  203 NP=NP-1
      IF(T(NP))206,203,206
  206 IF(NPIG.EQ.0)GO TO 13
      NP=NP-NPIG
      DO 16 I=1,NP
      T(I)=T(I+NPIG)
   16 A(I)=A(I+NPIG)
C     CHECK THAT THE TIME INCREASES, BEHAVING WELL, NEGLECTING BLANK SPACE
   13 CONTINUE
      DO 5 I=2,NP
      IF(T(I).NE.0.)GO TO 15
      DO 4 K=I,NP
      IF(T(K).NE.0.)GO TO 8
    4 CONTINUE
    8 NB=K-I
      NP=NP-NB
      DO 9 L=I,NP
      T(L)=T(L+NB)
    9 A(L)=A(L+NB)
   15 IF ((T(I-1)-T(I)).LT.98000.) GO TO 7
      DO 14 L=I,NP
   14 T(L)=T(L)+100000.
    7 IF(T(I).GT.T(I-1).AND.ABS(A(I)-A(I-1)).LT.2000.)GO TO 5
      WRITE(6,106)T(I),A(I)
      IF(T(I).GT.T(I-1).OR.(T(I-1)-T(I)).GT.20.)K1=2
      IF(T(I-1).GE.T(I).AND.(T(I-1)-T(I)).LE.20.)T(I)=T(I-1)+1
      IBEG=NPPC*((I-1)/NPPC)+1
      IEND=IBEG+NPPC-1
      WRITE(7,FMT)(T(J),A(J),J=IBEG,IEND)
    5 CONTINUE
      IF(NDK.NE.1)RETURN
      IF(NP.LE.3)RETURN
      DO 29 I=3,NP
      IF(ABS((T(I-1)-T(I-2))/(T(I)-T(I-2))-.5).LE..05)GO TO 29
      WRITE(6,106)T(I),A(I)
      K1=2
   29 CONTINUE
  207 FORMAT(5(A1,F7.0,F8.0))
  106 FORMAT(16H THE TIME AROUND 2F10.0,18H IS FAULTY. CHECK.)
      RETURN
      END
```

SAMPLE OF INPUT DATA FOR PICHECK PROGRAM

66.1T 6/27/66 2026PST CHOLAME,SHANDON, ARRAY NO.2 CAL.   2

+4   1.   1.   .5   1

00000	00000	00316	00000	00615	00000	00923	00000	01231	00000
01540	00000	01852	00000	02160	00000	02468	00000	02779	00000
03086	00000	03396	00000	03712	00000	04016	00000	04318	00000
04634	00000	04940	00000	05251	00000	05558	00000	05871	00000
06181	00000	06488	00000	06792	00000	07104	00000	07419	00000
07727	00000	08038	00000	08344	00000	08651	00000	08964	00000
09269	00000	09580	00000	09884	00000	10199	00000	10508	00000
10810	00000	11126	00000	11435	00000	11750	00000	12056	00000
12362	00000	12668	00000	12978	00000	13285	00000	13598	00000
13905	00000	14214	00000	14528	00000	14834	00000	15147	00000
15452	00000	15768	00000	16082	00000	16384	00000	16683	00000
16995	00000	17305	00000	17617	00000	17924	00000	18240	00000
18544	06368	18854	06368	19162	06368	19473	06368	19788	06368
20095	06367	20402	06367	20707	06367	21015	06367	21332	06367
21638	06367	21947	06367	22255	06367	22563	06367	22870	06367
23185	06367	23486	06367	23796	06367	24110	06367	24419	06367
24726	06367	25040	06367	25345	06367	25651	06367	25963	06367
26274	06367	26587	06367	26892	06367	27203	06367		

666666

N65E   7.91   1   1

00000	00000	00138	-00034	00167	00020	00191	-00007	00222	00023
00235	00009	00260	-00009	00287	-00042	00306	-00016	00328	-00004
00350	00053	00380	-00069	00416	00043	00441	00002	00458	00036
00482	-00005	00510	-00017	00534	-00007	00553	00023	00583	-00007
00603	00006	00622	00033	00651	00011	00675	-00017	00695	-00065
00716	-00027	00729	00005	00747	00023	00766	00039	00795	-00003
00816	00026	00835	00040	00851	00020	00875	-00018	00903	-00022
00920	-00058	00934	-00003	00952	00039	00979	-00001	01003	00034
01027	-00015	01044	00018	01066	-00012	01101	00073	01123	00007
01134	00041	01167	-00097	01194	-00023	01215	-00100	01234	-00024
01264	-00123	01282	-00069	01313	-00189	01333	-00071	01351	-00020
01371	00113	01390	00019	01404	-00134	01427	00105	01446	-00042
01470	00205	01474	00279	01504	-00018	01520	00072	01541	-00026
01556	00155	01572	00253	01583	00147	01599	00060	01624	00159
01640	00086	01659	00061	01691	-00223	01719	00089	01741	-00104
01759	-00167	01780	-00313	01808	-00123	01832	-00281	01865	-00026
01895	-00197	01926	-00013	01936	-00073	01959	00040	01978	00166
01998	00298	02016	00131	02030	-00045	02039	-00165	02051	-00215
02067	-00163	02099	-00300	02125	-00259	02150	-00307	02181	-00209
02222	-00319	02242	-00285	02266	-00250	02298	-00318	02316	-00251
02331	-00144	02344	-00027	02360	00207	02374	00388	02396	00667
02411	00859	02430	01097	02452	01252	02475	00941	02502	00469
02518	00142	02531	-00080	02551	-00339	02570	-00650	02591	-00905
02607	-01075	02623	-00629	02632	-00146	02640	00068	02644	00122
02654	00149	02663	00332	02676	00391	02690	00641	02700	00734
02705	00798	02717	00875	02726	00982	02738	01077	02746	01133
02762	00977	02770	00886	02789	00820	02796	00661	02808	00540
02815	00485	02828	00534	02850	00438	02866	00502	02878	00402
02895	00066	02909	-00247	02919	-00429	02942	-00491	02952	-00564
02964	-00640	02979	-00704	02997	-00755	03011	-00798	03025	-00741
03039	-00576	03052	-00389	03059	-00264	03071	-00184	03087	-00298
03097	-00405	03106	-00497	03119	-00566	03131	-00507	03142	-00349
03152	-00213	03154	-00164	03164	-00131	03174	-00067	03190	00004
03197	00053	03216	00113	03233	00045	03246	-00018	03260	-00094
03276	-00176	03293	-00104	03301	-00041	03314	00061	03326	00134
03348	00199	03371	00249	03388	00272	03414	00186	03436	00244
03454	00167	03470	00006	03481	-00059	03496	-00106	03514	-00047
03530	00030	03552	00101	03566	00032	03579	-00025	03602	-00108
03622	-00148	03648	-00139	03666	-00156	03698	-00089	03727	-00155

03756	-00033	03792	-00136	03821	-00085	03838	-00042	03847	00017
03855	00064	03862	00089	03878	00110	03902	00081	03912	00043
03926	00003	03938	00035	03956	-00032	03968	-00079	03986	-00123
04004	-00073	04037	00055	04058	00126	04090	00199	04105	00242
04118	00266	04139	00215	04160	-00024	04172	-00108	04193	-00188
04210	-00121	04227	00042	04246	00138	04264	00114	04287	00105
04306	00132	04319	00177	04335	00201	04359	00170	04378	00130
04392	00030	04407	-00045	04432	-00114	04466	-00084	04503	-00058
04523	-00119	04542	-00184	04568	-00215	04594	-00164	04608	-00103
04628	-00075	04654	-00059	04679	-00088	04698	-00135	04726	-00190
04751	-00238	04774	-00163	04789	-00068	04806	-00017	04842	00075
04876	00107	04907	00165	04930	00209	04958	00201	04967	00187
04987	00216	05036	00159	05074	00231	05091	00188	05118	00081
05141	-00006	05154	-00052	05191	-00059	05234	-00138	05271	-00109
05315	-00200	05334	-00181	05352	-00060	05364	-00000	05375	00028
05402	00042	05434	00022	05463	00030	05494	00005	05511	-00034
05527	-00069	05546	-00086	05571	-00053	05595	00014	05611	00065
05635	00110	05658	00078	05674	00027	05686	-00002	05714	-00010
05747	00035	05755	00062	05779	00092	05804	00063	05831	00050
05863	00027	05891	00008	05911	-00023	05932	-00061	05948	-00094
05966	-00117	05983	-00136	05991	-00125	06015	-00082	06034	-00025
06057	00089	06073	00146	06100	00181	06131	00143	06154	00073
06167	00031	06183	00011	06235	00035	06284	-00041	06355	00007
06416	-00035	06466	-00065	06525	-00098	06599	-00038	06631	00042
06660	00089	06692	00115	06727	00140	06755	00112	06771	00066
06795	00026	06812	-00003	06840	-00020	06858	-00007	06880	-00021
06915	00037	06933	00029	06958	00060	07007	-00005	07057	-00022
07106	-00075	07167	00044	07206	-00015	07247	00065	07283	00111
07316	00095	07346	00027	07376	00011	07417	-00011	07448	-00052
07492	-00090	07544	-00094	07571	-00054	07593	-00002	07612	00032
07627	00041	07647	00008	07664	-00041	07692	-00080	07713	-00099
07743	-00069	07791	-00027	07819	00033	07846	00066	07867	00088
07911	00064	07954	00088	08002	00106	08055	00103	08104	00075
08127	00079	08163	00044	08191	00022	08227	00035	08263	-00007
08285	-00053	08344	-00067	08367	-00108	08384	-00165	08414	-00221
08439	-00156	08458	-00069	08472	00003	08488	00054	08516	00008
08531	-00045	08543	-00118	08560	-00160	08571	-00106	08586	-00039
08598	00048	08611	00115	08630	00153	08656	00070	08668	-00025
08685	-00095	08702	-00054	08722	00036	08738	00112	08762	00146
08785	00123	08804	00089	08820	00039	08832	00017	08859	00016
08886	00037	08906	00068	08919	00083	08951	00045	08968	00018
09007	00011	09035	00040	09071	00008	09097	-00016	09126	-00019
09140	-00002	09175	00021	09204	-00005	09243	-00072	09267	-00060
09308	-00070	09338	-00046	09368	-00004	09394	00015	09448	00008
09498	00035	09535	00057	09570	00048	09630	00106	09691	00018
09752	00050	09781	00034	09810	00045	09835	-00003	09894	-00059
09918	-00026	09954	-00001	10007	-00043	10060	-00073	10107	-00027
10126	00034	10165	00054	10184	00040	10228	00085	10280	00061
10311	00034	10337	00034	10353	00022	10400	-00005	10450	-00057
10496	00021	10535	00005	10570	00021	10593	00011	10617	00034
10672	-00026	10739	00006	10779	-00035	10832	00009	10882	-00051
10938	00018	11012	-00013	11046	00042	11120	00013	11148	00049
11198	00019	11230	00045	11258	00065	11275	00079	11289	00054
11323	00083	11351	00062	11383	00031	11416	00008	11461	00009
11482	00009	11508	-00016	11542	-00050	11595	-00059	11632	-00055
11679	-00047	11731	-00012	11818	-00004	11855	00025	11886	00002
11927	-00017	11979	00026	12024	00007	12063	00016	12106	-00003
12135	-00006	12190	00040	12269	00000	12316	00057	12383	-00042
12424	00065	12454	00093	12480	00054	12513	00009	12542	00038
12576	00061	12605	00026	12640	-00014	12704	00009	12762	-00023
12834	00037	12902	00023	12939	00037	13010	00021	13069	00029
13121	00003	13160	00024	13223	00019	13291	-00014	13371	-00027

13458	-00014	13520	-00006	13557	00023	13620	00006	13678	00033
13787	00013	13859	00029	13912	-00001	13956	00022	14010	00037
14048	00030	14087	00010	14123	-00003	14167	00015	14190	00022
14242	00003	14303	-00011	14357	-00016	14396	-00043	14451	-00028
14498	-00004	14552	00031	14612	00020	14662	00031	14682	00046
14728	00031	14770	00055	14806	00054	14836	00040	14874	00037
14902	00016	14943	00021	14983	00003	15018	-00019	15100	-00059
15158	-00043	15200	-00008	15227	00025	15258	00035	15328	00021
15384	00029	15443	00035	15492	-00013	15523	00013	15571	00006
15607	-00041	15647	00007	15694	00011	15728	00029	15770	00021
15808	00046	15844	00033	15888	00039	15914	00048	15960	00035
16014	00027	16079	00030	16141	00028	16188	00004	16242	-00009
16347	-00036	16415	-00055	16487	-00018	16538	00026	16611	00010
16684	00021	16767	00005	16823	00025	16864	00043	16920	00029
16962	00018	17019	00024	17069	00034	17098	00042	17150	00018
17188	-00005	17246	-00009	17306	-00032	17385	00008	17455	-00006
17505	00026	17566	00003	17625	00010	17676	-00001	17731	00026
17787	00002	17835	00033	17891	00007	17990	00001	18052	-00005
18106	00017	18168	-00001	18214	00019	18243	00033	18299	00025
18368	00019	18422	00026	18459	00009	18511	-00003	18542	-00007
18611	-00023	18654	-00034	18707	-00019	18748	-00001	18804	00017
18846	00018	18878	00022	18922	00039	18968	00039	18996	00026
19034	00012	19062	00002	19091	00011	19151	00001	19203	00015
19273	00023	19324	00032	19382	00002	19421	00010	19442	00020
19485	00010	19570	-00001	19632	-00003	19731	00004	19758	00011
19812	00001	19879	00012	19915	-00004	19979	00020	20011	00036
20044	00037	20088	00028	20125	00019	20173	00026	20218	00013
20259	00006	20314	00006	20359	-00003	20412	00017	20496	00011
20535	00024	20578	00019	20615	00026	20668	00033	20724	00046
20780	00020	20827	00031	20879	00014	20935	00003	21024	-00001
21104	00004	21192	00006	21272	00025	21356	00023	21444	00022
21500	00007	21542	00011	21571	00002	21631	00011	21692	00017
21728	00024	21787	00017	21844	00024	21879	00034	21925	00023
21982	00020	22045	00002	22112	00008	22166	00015	22226	00007
22275	00009	22318	00012	22364	00018	22431	00009	22492	00004
22556	00001	22637	00010	22710	00011	22758	00029	22820	00026
22863	00036	22907	00025	22967	00043	23021	00021	23072	00010
23127	00018	23215	-00000	23286	00003	23359	-00002	23434	00007
23500	00002	23541	00022	23598	00017	23649	00023	23716	00013
23835	00012	23899	00008	23960	00002	24046	00025	24129	00015
24230	00005	24306	00011	24386	00004	24450	00020	24504	00019
24592	00016	24697	00021	24788	00008	24854	00012	24943	00018
25006	00026	25075	00014	25130	00020	25204	00007	25276	00007
25367	-00001	25456	00016	25536	00021	25592	00018	25655	00011
25731	00018	25779	00019	25823	00013	25873	00020	25942	00014
26002	00017	26051	00028	26139	00017	26208	00025	26314	00010
26404	00014	26448	00022	26528	00007	26578	00017	26642	00007
26687	00016	26738	-00001	26844	00009	26916	00020	26970	00011
27034	00021	27110	00011	27196	00017				
666666									
00000	00000	00312	00004	00924	00001	01547	00002	02163	00003
02779	00002	03404	00001	04023	00005	04635	00007	05251	00007
05871	00006	06488	00007	07106	00008	07723	00010	08347	00013
08960	00011	09583	00009	10200	00015	10816	00012	11436	00011
12056	00014	12669	00018	13290	00012	13913	00014	14527	00011
15147	00008	15767	00014	16383	00015	17000	00011	17616	00013
18240	00013	18855	00015	19473	00013	20096	00014	20704	00017
21334	00013	21949	00017	22569	00016	23183	00017	23804	00019
24420	00016	25035	00016	25658	00018	26276	00015	26898	00017
27202	00017								
666666									

DOWN 7.07 1 1

00000	00000	00140	-00084	00149	00004	00160	-00028	00174	00019
00190	00091	00226	-00083	00245	00024	00267	-00063	00289	00060
00309	-00073	00328	00044	00339	00002	00344	00053	00370	-00042
00384	00018	00395	00068	00412	00026	00420	00000	00434	-00034
00453	00057	00467	-00038	00484	00066	00513	-00099	00528	00003
00536	-00005	00556	00100	00578	-00191	00596	00018	00606	-00040
00626	00122	00642	00015	00650	00032	00675	-00158	00690	-00058
00704	00026	00712	00142	00728	-00027	00746	-00193	00760	-00024
00775	00069	00803	-00083	00821	00075	00843	-00067	00859	-00123
00871	00002	00882	00128	00899	00075	00908	-00048	00924	00042
00934	-00004	00946	00072	00960	-00006	00976	-00138	00990	00020
00999	00146	01023	-00128	01039	00060	01054	00254	01069	-00067
01079	-00171	01095	00058	01115	-00352	01130	00064	01143	00405
01148	00326	01155	00355	01170	-00028	01178	-00278	01187	-00422
01198	-00152	01209	-00305	01236	00556	01259	-00201	01282	00424
01299	-00213	01313	-00463	01324	-00081	01335	00267	01352	-00028
01370	-00263	01384	00054	01406	-00232	01430	00214	01447	-00084
01462	00220	01487	-00542	01511	00529	01534	-00521	01556	00263
01576	-00069	01595	00270	01609	00108	01617	00151	01639	-00335
01650	-00032	01664	-00000	01682	00222	01706	-00366	01726	00328
01741	-00157	01748	-00116	01771	-00388	01783	-00097	01791	-00000
01804	00299	01823	-00287	01838	-00175	01847	-00232	01861	00222
01871	00764	01884	00078	01903	-00571	01912	00014	01927	00427
01953	00214	01962	-00062	01971	-00176	01990	00342	02003	-00065
02019	-00257	02036	-00101	02046	-00071	02055	-00007	02067	00043
02086	-00135	02104	00117	02110	00076	02120	00202	02134	00082
02152	-00200	02171	00110	02179	00285	02194	00091	02205	-00189
02214	-00342	02235	00124	02250	-00105	02271	-00219	02288	-00034
02305	00022	02327	00264	02338	00114	02360	00001	02379	00243
02393	00008	02404	-00109	02419	-00010	02441	-00237	02456	-00055
02484	-00282	02506	00110	02527	-00290	02551	00271	02566	-00091
02586	-00350	02603	-00100	02611	00031	02622	00168	02643	00049
02655	00146	02672	-00090	02687	-00271	02700	-00104	02710	00104
02723	00252	02740	00169	02756	-00008	02779	00138	02797	00034
02807	-00091	02819	-00171	02847	00122	02856	-00008	02868	-00181
02880	-00258	02914	-00069	02936	-00221	02961	00061	02986	-00157
03011	00065	03018	00021	03042	00193	03059	00108	03083	00208
03114	00134	03128	00230	03138	00175	03148	00226	03165	00158
03173	00237	03202	00013	03211	00085	03235	-00015	03247	00045
03271	-00029	03284	-00096	03292	-00053	03316	-00131	03328	-00079
03341	-00138	03363	-00280	03384	-00188	03402	-00110	03420	-00014
03429	-00056	03442	-00002	03458	-00044	03487	00038	03514	00010
03530	00056	03548	00022	03573	00094	03592	00033	03604	-00020
03611	00006	03635	-00106	03657	-00052	03680	-00100	03697	-00135
03707	-00095	03724	-00018	03734	-00062	03755	-00003	03771	-00061
03786	00021	03799	00085	03834	00042	03847	00096	03861	00126
03875	00165	03901	00098	03919	00147	03932	00191	03946	00239
03967	00194	03983	00134	03995	00181	04018	00118	04034	00033
04057	-00067	04072	-00022	04095	-00088	04111	-00151	04136	-00088
04164	-00180	04194	-00082	04210	-00130	04233	-00054	04245	-00078
04256	-00090	04284	-00036	04304	-00104	04330	-00029	04355	-00079
04382	00036	04396	-00003	04421	00085	04446	00015	04475	00104
04508	00015	04520	00039	04534	00080	04555	00028	04573	00085
04597	00012	04611	00051	04632	00100	04656	00032	04684	00047
04711	00096	04725	00034	04739	-00017	04768	-00028	04785	00003
04798	00038	04824	-00037	04843	-00002	04859	-00053	04887	-00088
04902	-00111	04922	-00089	04946	-00111	04968	-00053	04982	-00077
05007	-00131	05023	-00061	05036	00011	05050	-00031	05064	-00068
05078	-00030	05091	00019	05107	00055	05122	00021	05139	00054
05159	00004	05171	00057	05183	00105	05199	00126	05226	00077
05244	00097	05270	00059	05290	00112	05323	00011	05351	00062

05376	-00044	05404	00020	05427	-00045	05439	-00006	05459	-00046
05480	00016	05510	-00042	05535	00053	05568	-00037	05587	00002
05604	00021	05623	00054	05663	-00015	05682	00013	05712	-00014
05735	-00047	05758	-00024	05783	-00041	05808	-00112	05833	-00027
05855	-00087	05880	-00019	05904	00020	05924	00046	05946	00007
05959	00047	05974	00088	05992	00048	06016	00083	06039	00028
06053	00062	06081	00002	06090	00042	06115	-00020	06136	00030
06174	-00058	06193	-00005	06211	00042	06234	-00011	06256	00034
06268	00005	06290	00051	06323	00022	06351	00032	06361	00053
06379	00018	06411	00074	06450	00024	06463	00044	06498	-00012
06515	00004	06535	-00064	06559	-00022	06584	-00058	06600	-00015
06618	-00055	06634	-00010	06656	-00074	06675	-00029	06697	-00041
06704	-00033	06723	-00093	06750	00011	06764	-00034	06790	00033
06807	-00010	06824	00074	06847	-00023	06869	00115	06896	-00010
06912	00083	06939	-00078	06960	00073	07003	-00009	07039	00048
07054	00012	07067	00049	07092	-00021	07114	00029	07142	00028
07171	00072	07192	00028	07218	00072	07247	00014	07266	-00073
07291	00013	07328	-00084	07351	-00022	07370	-00054	07388	-00017
07402	-00054	07416	-00015	07430	00003	07447	00039	07474	-00015
07505	00037	07522	00002	07544	00024	07558	00002	07582	00044
07606	-00003	07617	00025	07631	-00001	07651	00039	07668	00036
07710	-00035	07740	00025	07762	-00020	07783	00034	07804	-00008
07823	-00026	07838	-00041	07872	00026	07896	-00032	07926	-00004
07946	00002	07962	00034	08001	-00011	08034	00060	08061	00021
08078	00035	08103	00015	08117	00055	08132	00022	08144	00048
08167	00009	08186	00042	08220	-00004	08247	00020	08280	-00034
08306	00016	08342	-00034	08368	00008	08408	-00018	08438	00015
08461	-00014	08487	00051	08518	-00020	08543	00029	08568	-00004
08594	00038	08635	00070	08662	00035	08689	00007	08720	00024
08744	-00002	08775	00016	08786	00027	08817	-00005	08859	00012
08891	-00023	08908	-00005	08932	-00050	08950	-00018	08975	-00046
08998	-00005	09021	00004	09040	-00031	09062	00004	09087	00047
09117	-00013	09134	00038	09158	-00010	09186	00024	09212	00035
09235	00009	09259	00043	09279	-00002	09310	00058	09338	00015
09353	00051	09388	00006	09406	00044	09439	-00010	09459	00020
09489	00026	09510	00035	09535	-00019	09567	00020	09587	00009
09610	-00027	09627	00004	09655	-00029	09694	-00029	09721	00005
09756	-00015	09795	00022	09819	00017	09834	00032	09871	00004
09905	00035	09955	00018	09995	-00005	10043	00048	10067	00011
10091	00062	10122	00010	10140	00048	10170	00010	10218	-00002
10252	-00018	10274	-00007	10307	-00019	10342	00022	10388	-00005
10410	00038	10430	00010	10460	00042	10494	00058	10512	00018
10539	00060	10578	00038	10611	00018	10658	00002	10673	-00006
10710	00027	10744	-00030	10764	00016	10786	-00030	10808	00024
10839	-00044	10875	-00011	10908	-00031	10944	-00011	10990	00010
11011	-00016	11054	00028	11079	00013	11114	00055	11132	00017
11172	00061	11215	00007	11243	00040	11278	-00004	11299	00027
11369	-00026	11399	00014	11423	-00032	11451	00017	11471	-00020
11489	00008	11512	00033	11560	00018	11591	00034	11612	00046
11642	00020	11658	00041	11677	00014	11696	00043	11730	00013
11760	-00007	11792	00018	11830	-00012	11854	00013	11928	-00028
11950	00018	11970	-00013	12010	00017	12047	00028	12070	-00006
12103	00034	12140	00008	12162	00026	12235	00011	12264	00011
12284	00028	12316	-00004	12350	00030	12387	-00005	12431	00023
12476	00004	12535	-00013	12587	00001	12620	-00018	12656	00007
12706	-00029	12755	00020	12810	00031	12883	00012	12966	00001
13082	00025	13144	00024	13250	00014	13297	00032	13330	00010
13364	00022	13399	00013	13426	00028	13475	-00006	13515	00016
13579	-00007	13646	00016	13743	00005	13796	00008	13826	-00004
13854	00016	13922	-00008	14002	00028	14030	00010	14066	00040
14118	00004	14153	00038	14180	00001	14212	00037	14255	-00006
14284	00023	14331	-00006	14420	00012	14463	-00008	14541	00014

14600	-00023	14663	00003	14701	-00002	14782	00044	14828	00017
14874	00035	14934	00010	15024	00024	15076	00035	15147	00014
15205	00015	15237	-00015	15326	-00011	15394	-00012	15482	00011
15564	00012	15626	-00002	15683	00026	15778	00016	15880	00001
15981	00017	16089	00027	16190	00038	16270	00015	16327	-00005
16412	-00014	16494	-00015	16553	-00015	16623	00006	16710	00014
16768	00036	16852	00028	16918	00022	16969	00021	17004	00040
17086	00019	17164	00002	17281	-00010	17356	-00017	17456	-00012
17543	00006	17641	00037	17735	00016	17828	00003	17943	-00013
18000	-00000	18088	00004	18144	-00004	18221	00007	18334	00016
18420	00026	18498	00017	18597	00002	18691	-00007	18800	00001
18870	00004	18924	00010	19011	00009	19084	00008	19139	00005
19210	-00001	19279	-00009	19327	-00002	19395	00004	19466	00015
19522	00015	19582	00004	19658	-00001	19720	-00002	19799	00003
19858	00011	19917	00007	19988	00009	20055	00009	20120	00009
20167	00010	20237	00000	20308	-00006	20452	00007	20546	00007
20619	00006	20670	00003	20747	00017	20838	00000	20922	-00008
20954	00003	20991	-00011	21033	00003	21100	-00015	21186	00007
21275	00009	21352	00011	21448	00008	21564	00006	21675	00006
21757	00012	21831	00005	21863	00009	21924	-00008	21993	-00004
22112	-00001	22182	-00003	22250	-00008	22366	-00004	22480	-00000
22539	00010	22604	00013	22688	00006	22756	00007	22846	00012
22939	-00003	23030	00002	23094	00001	23136	-00007	23209	-00008
23272	00005	23367	-00001	23464	00010	23596	00004	23679	00005
23822	00005	23958	00008	24076	00004	24203	00000	24340	00007
24446	00003	24543	00010	24616	00010	24680	00009	24791	00000
24899	-00005	24999	00010	25098	00009	25223	00013	25324	00003
25446	-00001	25563	-00006	25666	00000	25770	00005	25890	00010
25931	00003	25994	00008	26063	00004	26146	-00008	26216	-00003
26282	00002	26370	-00000	26430	-00003	26511	00000	26586	00004
26643	-00002	26699	00007	26744	00004	26808	00007	26910	00006
26989	00001	27087	00005	27198	-00003				
666666									
00002	00000	00318	00002	00925	00001	01540	-00002	02164	00004
02780	00002	03396	00005	04027	00004	04634	00007	05253	00009
05867	00008	06487	00008	07100	00009	07725	00012	08343	00012
08968	00011	09575	00010	10202	00012	10819	00010	11433	00013
12056	00014	12672	00013	13282	00013	13907	00015	14531	00016
15151	00014	15770	00013	16383	00013	16994	00014	17622	00015
18238	00012	18851	00012	19476	00017	20098	00018	20715	00017
21327	00018	21948	00019	22563	00016	23182	00020	23802	00018
24419	00017	25034	00015	25654	00015	26274	00018	26897	00013
27196	00013								
666666									

<u>P2SCALE MAIN Program</u> (Lee, Brady, Trifunac)

This program scales the raw digitized data into the units of seconds and G/10, where G is the acceleration of gravity. The digitized timing marks are smoothed and used for the time coordinates. The digitized fixed trace is smoothed and subtracted from the acceleration trace. A horizontal zero line is fixed to have a mean value of zero. A correction is made for position on the digitizer after scaling. The scaled data are printed out and plotted in the format required for a Volume I report. The same data are also punched out for loading onto the Volume I tape.

Some of the key parameters in the program are:

F1 through F0 (in MAIN 9) are formats used to print out the raw digitized data.

W1 through W0 (in MAIN 10) are formats used to print out and punch the scaled data.

YO, YB, XO, XMAX, SL (in MAIN 12) are coordinates used for plotting.

DIGSCL is the number of digitized units per cm.

G, SEC (in MAIN 16) are numbers used to label the acceleration and time axes. (G contains 5 words of 2 characters each; SEC contains 46 words of 2 characters each.)

RECTTL is the record title (14 words of 4 characters each).

NFTM is selected from the following table:

NFTM	Explanation
-3	scaled data (in units of sec and G/10) are read in to be rescaled
-2	one set of time marks, no fixed trace available for this record
-1	no time marks, no fixed trace available for this record
0	one set of time marks, no fixed trace. Use the time marks in place of one fixed trace.
1	no time marks, only one fixed trace available for the whole record
2	one set of time marks, only one fixed trace available for the whole record
3	no time marks, one fixed trace available for each acceleration component
4	one set of time marks, one fixed trace available for each acceleration component

SCFTS1 is the width of the digitized record.

SCFTS2 is the width of the original record.

TMS is the time in seconds between successive time marks.

NMIG - The first NMIG points in the deck containing the digitized time marks are to be ignored.

TMAVE is the average number of digitizer units per sec to be read in when there are no time marks.

DN is the direction of the components (two words: one in A4 and one in A1 format).

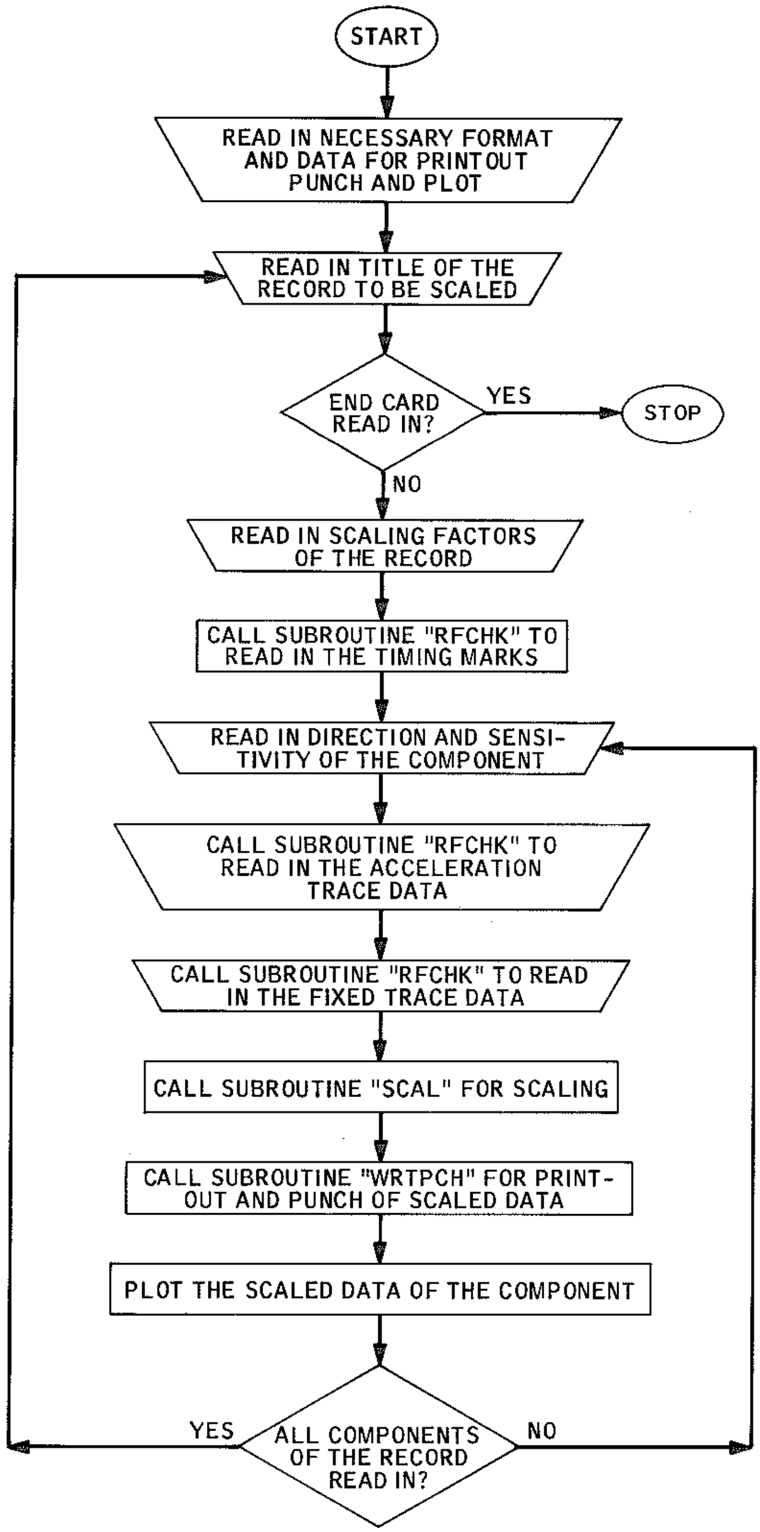
SEN is the component sensitivity in the original record.

NPIG - The first NPIG points in the deck of digitized acceleration data are to be ignored.

NPBIG - The first NPBIG points in the deck of digitized fixed trace data are to be ignored.

Each continuous sequence of data, whether time marks, acceleration (headed with one card containing the corresponding direction and sensitivity), or fixed traces, are followed by a card containing 'bb666666'.

P2SCALE FLOW CHART



```
      DIMENSION XO(3,2),YO(3,3,2,2),YB(3,2,2),PLT(2),SL(2),G(5),SEC(46),
     *FO(300),F1(16),F2(11),TA(10),YZERO(3),YTOP(2)
      COMMON/RAWDTA/M,NPB,NP,TM(700),TB(1500),AB(1500),T(9000),A(9000),
     *STAR(9000)/RAWFMT/F3(13),F4(10),F5(12),F6(18),F7(20),F8(20),F9(20)
     */WRTFMT/W1(20),W2(60),W3(80),W4(40),W5(16),W6(18),W7(6),W8(7),W9
     *(13),WO(15),WF1(5),WF2(9),WF3(4),NEXTPG(2)/TITLE/RECTTL(14),DN(2)
      INTEGER STAR,S1
      DATA S1/'*'/
      READ(5,111)F1,F2,F3,F4,F5,F6,F7,F8,F9,FO
      READ(5,111)W1,W2,W3,W4,W5,WF3,NEXTPG,W6,W7,WF1,WF2,W8,W9,WO
  111 FORMAT(20A4)
      READ(5,1)YO,YB,XO,XMAX,        SL,DIGSCL,YZERO,YTOP,PLOT
    1 FORMAT(16F5.2)
      IPLOT=PLOT+.2
      CALL SYSPSZ(IPLOT)
      READ(5,2)G,SEC
    2 FORMAT(40A2)
C     IPLOT =0, NARROW PAPER USED.
C           =1, WIDE PAPER USED FOR PLOTTING
      IF(IPLOT.EQ.0)GO TO 1973
      XO(2,2)=XO(2,2)-17.
      XO(3,2)=XO(3,2)-17.
      XMAX=XMAX-17.
      DO 1970 LSCL=1,2
      DO 1970 NDK=1,3
 1970 YO(1,NDK,2,LSCL)=YO(1,NDK,2,LSCL)+11.5
 1973 CONTINUE
      NFTM=0
      NTM=1
      DO 35 NRCD=1,99
      READ(5,3,END=5)RECTTL,NTTL,P2L,NDKS,PLOT,WRT,PUN,PLT,NTHSL,NARROW
    3 FORMAT(14A4,I2,F1.0,I1,5F1.0,I1,12X,I2)
C    RECTTL,NTTL   RECORD TITLE & NUMBER OF COLUMNS USED FOR IT(OPTIONAL)
C   PLT(1),PLT(2),PUN =0. RESPECTIVELY FOR SMALL, BIG PLOT AND PUNCH.
C    NDKS=NUMBER OF COMPONENTS READ IN
C    PLOT=1, FOR SCALED DATA READ IN FOR PLOT(IF NO PUNCH, PUT  PUN=1)
C     WRT=1, FOR SCALED DATA READ IN FOR PRINTOUT
C     P2L =1, FOR P2L(ADJUSTING MINIMUM RMS), FOR SCALED DATA, PUT WRT=1
C    NARROW  NO. OF ARROWS ON PLOT
      IF(NARROW.NE.0.AND.PLOT.NE.0.)READ(5,33)(TA(I),I=1,NARROW)
   33 FORMAT(10F8.2)
      IF(NTTL.EQ.0)NTTL=56
      IF(NDKS.EQ.0)NDKS=3
      IF(PLOT.EQ.1..OR.WRT.EQ.1.)GO TO 42
      READ(5,13)NFTM,SCFTS1,SCFTS2,TMS,SHIFT,NMIG,M,TMCHK,K1,K2
   13 FORMAT(I2,8X,3F10.2,F1.0,I3,I4,2X,F1.0,2I1)
C   NFTM, NUMBER OF FIXED TRACES(NFT)+NUMBER OF TIMING MARKS DECK(NTM)
C    NFTM     NTM    NFT
C     -3       0      0     NO TM,  NO FT, SCALED DATA READ IN TO BE RESCALED
C     -2       1      0     ONE TM, NO FT
C     -1       0      0      NO TM, NO FT
C      0       1      0     ONE TM, NO FT, WITH TM USED AS 1 FT
C      1       0      1      NO TM, ONE FT
C      2       1      1     ONE TM, ONE FT
C      3       0      3      NO TM, FT IN EVERY COMPONENT
C      4       1      3     ONE TM, FT IN EVERY COMPONENT
C    TMS=TIME IN SEC. BETWEEN EACH INTERVAL OF TIMING MARKS
C    SCFTS1,  WIDTH OF DIGITIZED RECORD
C    SCFTS2,  WIDTH OF ORIGINAL RECORD
C    SHIFT =1,EVERY TACE TO BE SHIFTED TO ORIGIN(SOME 35 MM FILM).
C    TMCHK   =1, WHEN THE TIMING MARKS ARE NOT TO BE CHECKED FOR SPACING.
```

```
      NTM=1-MOD(IABS(NFTM),2)
      NFT=IDIM(NFTM,NTM)
      RCMAG=SCFTS1/SCFTS2
      IF (NFTM.LE.-3)GO TO 42
      IF(NTM.EQ.1)CALL RFCHK(M,TM,STAR,AB,1.-TMCHK,NMIG,F1,SHIFT,K1,0)
      IF(NTM.EQ.1)TMAVE=(TM(M )-TM(1))/(M-1)/TMS
      IF(NTM.EQ.0)READ(5,33)TMAVE
      IF(NFTM.NE.0)GO TO 42
      NPB=M
      DO 41 I=1,NPB
   41 TB(I)=TM(I)
   42 DO 31 NDK=1,NDKS
      IF(K1.EQ.2)K2=2
C   DN,  DIRECTION OF THE COMPONENT (2 WORDS TOTALING 5 CHARACTERS).
C   SEN,  SENSITIVITY OF ORIGINAL RECORD.
      READ(5,93)DN,NP,SEN,RMS,NPIG,NPF,NPBIG
   93 FORMAT(A4,A1,I4,1X,F5.2,F5.4,3I5)
      IF(SEN.NE.0.)GO TO 43
      READ(5,303)(T(I),STAR(I),A(I),I=1,NP)
      READ(5,302)
  303 FORMAT(///(5(F7.3,A1,F6.3),10X))
  302 FORMAT(80X)
      IF(NFTM.GT.-3.OR.PLOT.NE.0.)GO TO 100
      WRT=1.
      DO 95 I=1,NP
   95 A(I)=A(I)*RCMAG
      GO TO 100
   43 RECSEN=SEN*RCMAG
      DSEN=RECSEN*DIGSCL
      WRITE(6,FO)RECTTL,DN,TMAVE,SEN,RCMAG,DIGSCL,RECSEN,DSEN,TMAVE,DSEN
      CALL RFCHK(NP,T,STAR,A,SEN,NPIG,F1,SHIFT,K1,NDK)
      IF(SEN.EQ.0.)GO TO 100
      IF(NFT.GT.K2)NPB=NPF
      IF(NFT.GT.K2)CALL RFCHK(NPB,TB,STAR,AB,SEN,NPBIG,F2,SHIFT,K1,NDK)
      IF(K1.EQ.2)GO TO 31
  100 IF(SEN.EQ.0..AND.WRT.EQ.0.)GO TC 200
      IF(RMS.NE.0.)GO TO 150
      CALL SCAL(  DSEN,TMAVE,NFT,NTM,TMS, P2L, WRT,RMS,K2,NDK)
  150 CALL WRTPCH(NP,T,STAR,A,NTM,56,PUN,RMS)
      IF(NARROW.EQ.0.OR.NDK.NE.NDKS)GO TO 200
      IF(WRT.EQ.0.)GO TO 151
      N=1
      DO 46 I=1,NP
      IF(ABS(T(I)-TA(N)).GT..05)GO TC 46
      STAR(I)=S1
      N=N+1
      IF((N-NARROW))46,46,200
   46 CONTINUE
  151 I=0
      DO 45 N=1,NARROW
   44 I=I+1
      IF(STAR(I).EQ.S1)GO TO 45
      GO TO 44
   45 TA(N)=T(I-1)
  200 IF(PLT(1)*PLT(2).NE.0.)GO TO 31
      IF(NDK.NE.1)GO TO 101
C   NTHSL, NO. OF 30 SECONDS' LENGTH
      IF(NTHSL.EQ.0)NTHSL=1.+(T(NP)-DIM(T(NP),90.))/32.
      LSCL=1+NTHSL/3
      CALL SYSXMX(XMAX)
  101 NBEG=1
      DO 102 K=1,NTHSL
```

```
      TBEG=30.*(K-1)
      TEND=TBEG+30.
      NI=4-2*(LSCL-1)*(K/2)
      NI1=NI+1
      N30=0
      DO 4 I=NBEG,NP
      IF(T(I).GT.TEND)GO TO 15
    4 N30=N30+1
   15 NEND=NBEG+N30-1
C    NSCL =1,SMALL PLOT,SCALE:.2 IN/SEC: =2,BIG PLOT,.5 IN/SEC..
C    (X0,Y0)  ORIGIN OF 1 COMPONENT OF 1 30 SEC. INTERVAL.
C    (X0,YL)  ORIGIN OF THE CORRESPONDING HORIZONTAL LABELS(TIME AXIS).
C    NI, NO. OF INTERVALS IN VERTICAL LABELS(0-60 SEC.,4; 60-90 SEC.,2)
      DO 99 NSCL =1,2
      IF(PLT(NSCL).NE.0.)GO TO 99
      X0=X0(K,NSCL)
      Y0=Y0(K,NDK,NSCL,LSCL)
      IF(NSCL.EQ.2.AND.NTHSL.EQ.1.AND.IPLOT.EQ.1)Y0=YZERO(NDK)
      YL=Y0-  YB(K,NSCL,LSCL)
      XSCL= SL(NSCL)
C    PLOT THE RECORD TITLE
      YTTL=Y0+YTOP(NSCL)-(NTHSL/3)*(K/2)*(NSCL/2)*1.25
      IF(NDK.EQ.1.AND.K.LE.NSCL) CALL SYSSYM
     1(X0+15.*XSCL*(1.-NTTL/70.),YTTL          ,.5*XSCL,RECTTL,NTTL,0.)
      IF(IPLOT.EQ.0.AND.(NSCL*K)/NDK.EQ.4)
     *                         CALL SYSSYM(X0+14.,9.25,.15,'PAGE 2',6,0.)
C    PLOT THE VERTICAL LABELS (ACCELERATION AXIS)
C    SSCL,STTL,SLBL,  SIZE OF TICK MARKS,TITLE AND NUMERIC LABELS
      SLBL=.04*(NSCL+1)
      STTL=.08*NSCL
      SSCL=.05*(NSCL+1)
      XL=X0-.05*(NSCL+3)
      XR=X0+30.2*XSCL
      DO 7 I=1,NI1
      YY=Y0+XSCL*(I-3+2/NI)
      CALL SYSSYM(XL-2.*SLBL,YY-SLBL/2.,SLBL,G(I+2/NI),2,0.)
      CALL SYSSYM(XL+SSCL/2.,YY,SSCL,13,-1,90.)
    7 CALL SYSSYM(XR+SSCL/2.,YY,SSCL,13,-1,90.)
      CALL SYSSYM(XL,Y0,XSCL*NI,13,-1,0.)
      CALL SYSSYM(XR,Y0,XSCL*NI,13,-1,0.)
      IF(NDK.EQ.3.AND.(K.EQ.1.OR.K.EQ.NTHSL))CALL SYSSYM
     1(X0-4.*SSCL,YL+6.*XSCL,STTL,'ACCELERATION IN G/10',20,90.)
C    PLOT THE DIRECTION OF THE COMPONENT
      CALL SYSSYM(XL,Y0+.56*XSCL*NI,.4*XSCL-.05*(NSCL-1)*(2/NI),DN,5,0.)
C    PLOT THE HORIZONTAL LABELS (TIME AXIS)
      IF(NDK.NE.NDKS)GO TO 30
      YR=Y0+9.*XSCL
      DO 17 I=1,16
      XX=X0+2.*XSCL*(I-1)
      IBEG=TBEG/2.+.02
      CALL SYSSYM(XX,YL+SSCL/2.,SSCL,13,-1,0.)
      IF(NI.EQ.4.AND.NDK.EQ.3)CALL SYSSYM(XX,YR+SSCL/2.,SSCL,13,-1,0.)
   17 CALL SYSSYM(XX-SLBL,YL-3.*SLBL/2.,SLBL,SEC(IBEG+I),2,0.)
      CALL SYSSYM(X0+12.5*XSCL,YL-4.*SLBL,STTL,'TIME IN SECONDS',15,0.)
      CALL SYSSYM(X0+15.*XSCL,YL,30.*XSCL,13,-1,90.)
      IF(NI*NDK.EQ.12)CALL SYSSYM(X0+15.*XSCL,YR,30.*XSCL,13,-1,90.)
C    PLOTTING THE ARROWS INDICATING SHIFTING POSITIONS (OPTIONAL).
      IF(NARROW.EQ.0)GO TO 30
      YP=Y0-(NI-1)*XSCL/2.+XSCL/4.*(NSCL/2)*(K/3)
      YQ=YP-.1*XSCL
      DO 8 I=1,NARROW
      IF(TA(I)     .GT.TEND)GO TO 30
```

```
      IF(TA(I)     .LE.TBEG)GO TO 8
      TP=XO+(TA(I)     -TBEG)*XSCL
      CALL SYSSYM(TP,(YL+YP)/2.,YP-YL,13,-1,0.)
      CALL SYSSYM(TP,YQ,.05,2,-1,0.)
    8 CONTINUE
C     PLOT THE ACCELERATION TRACE IN THE INTERVAL (TBEG,TEND), N30 POINTS.
   30 IF(N30.EQ.0)GO TO 99
      CALL SYSPLT(XO+XSCL*(T(NBEG)-TBEG),YO+XSCL*A(NBEG),3)
      DO 6 I=NBEG,NEND
    6 CALL SYSPLT(XO+XSCL*(T(I)-TBEG),YO+XSCL*A(I),2)
   99 CONTINUE
  102 NBEG=NEND+1
      IF(NDK.EQ.NDKS.AND.NARROW.NE.0)WRITE(6,83)(TA(I)     ,I=1,NARROW)
   83 FORMAT(1H1//10X,'ARROW POSITIONS (IN SEC)'//(13X,6(F9.3,',')))
   31 IF(PLOT.EQ.0.)WRITE(6,NEXTPG)
      IF(PLT(1)*PLT(2).NE.0.)GO TO 35
      CALL SYSEND(-1,0)
      CALL SYSPSZ(IPLOT)
   35 CONTINUE
    5 STOP
      END
```

Subroutine RFCHK (Lee, Brady)

RFCHK is called by Volume I P2SCALE MAIN program to read in a set of digitized data, to count the number of points, to check that all points have increasing time coordinates, and to correct if necessary.

Usage

CALL RFCHK(NP, T, STAR, A, SEN, NPIG, FMT, SHIFT, K1, NDK)

Where

NP = number of points to be read in

T(I) = time coordinates

STAR(I) = asterisks besides time-ordinates indicating shifting position of the digitized record

A(I) = acceleration coordinates

SEN = sensitivity of acceleration in cm/G

NPIG = number of points to be ignored at start

FMT = format used to printout the raw data

SHIFT = 1, if the time coordinates are to be shifted to the origin (this is required for some 35 mm film records).

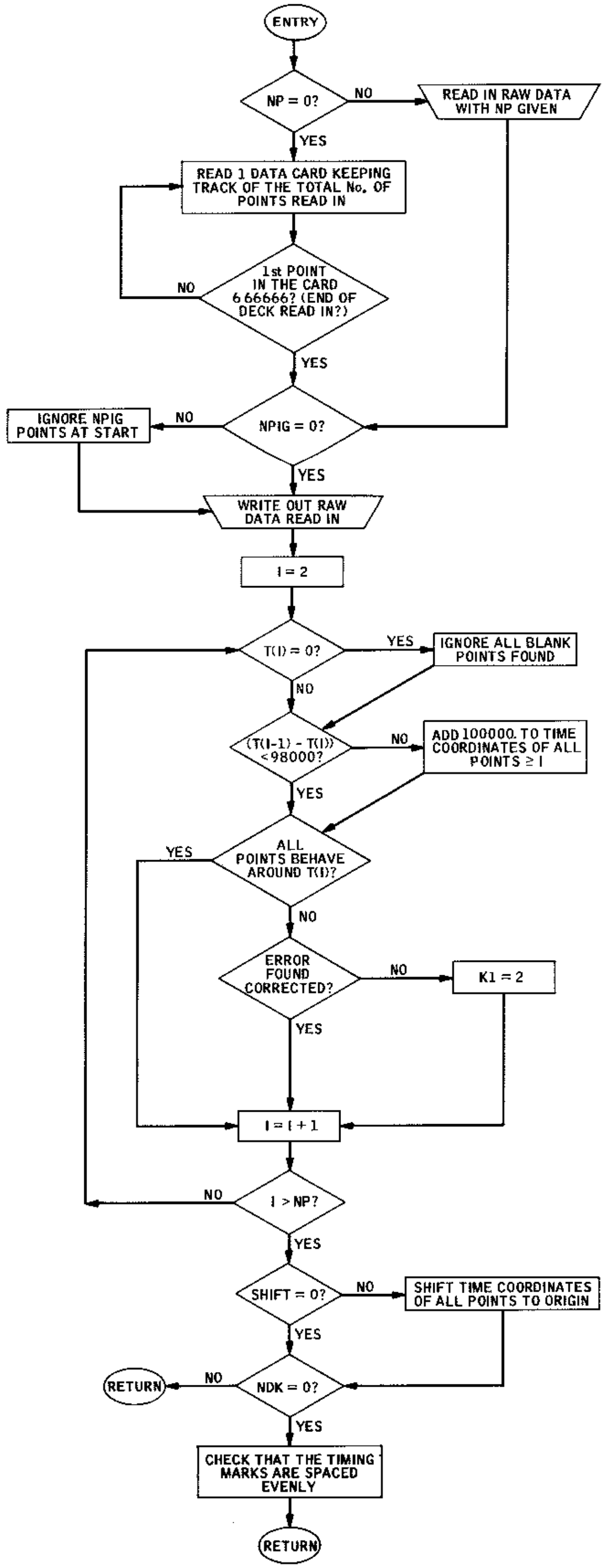
0, otherwise.

K1 = 2, if error is found in the data

1, otherwise

NDK = the number of the components read in

SUBROUTINE RFCHK FLOW CHART



```
      SUBROUTINE RFCHK(NP,T,STAR,A,SEN,NPIG,FMT,SHIFT,K1,NDK)
      DIMENSION T(1),A(1),FMT(1),STAR(1)
C    READ IN DATA AND COUNT THE NUMBER OF POINTS
      IF(NP.EQ.0)GO TO 200
      READ(5,207)(STAR(I),T(I),A(I),I=1,NP)
      GO TO 206
  200 N2=0
  201 NP=N2+1
      N2=NP+4
      READ(5,207)(STAR(I),T(I),A(I),I=NP,N2)
      IF((T(NP)-666666.))201,203,201
  203 NP=NP-1
      IF(T(NP))206,203,206
  206 IF(NPIG.EQ.0)GO TO 13
      NP=NP-NPIG
      DO 16 I=1,NP
      T(I)=T(I+NPIG)
   16 A(I)=A(I+NPIG)
C    CHECK THAT THE TIME INCREASES, BEHAVING WELL, NEGLECTING BLANK SPACE
   13 IF(NDK*SEN.NE.0.)WRITE(6,FMT)(T(I),A(I),I=1,NP)
      DO 5 I=2,NP
      IF(T(I).NE.0.)GO TO 15
      DO 4 K=I,NP
      IF(T(K).NE.0.)GO TO 8
    4 CONTINUE
    8 NB=K-I
      NP=NP-NB
      DO 9 L=I,NP
      T(L)=T(L+NB)
    9 A(L)=A(L+NB)
   15 IF ((T(I-1)-T(I)).LT.98000.) GO TO 7
      DO 14 L=I,NP
   14 T(L)=T(L)+100000.
    7 IF(T(I).GT.T(I-1).AND.ABS(A(I)-A(I-1)).LT.2000.)GO TO 5
      IF(T(I).GT.T(I-1).OR.(T(I-1)-T(I)).GT.20.)K1=2
      IF(T(I-1).GE.T(I).AND.(T(I-1)-T(I)).LE.20.)T(I)=T(I-1)+1
      WRITE(6,106)T(I),A(I)
    5 CONTINUE
      IF(SHIFT.EQ.0.)GO TO 51
      X1=T(1)
      DO 6 I=1,NP
    6 T(I)=T(I)-X1
   51 IF(NDK.NE.0.OR.SEN.EQ.0.)RETURN
      DO 29 I=3,NP
      IF(ABS((T(I-1)-T(I-2))/(T(I)-T(I-2))-.5).LE..05)GO TO 29
      WRITE(6,106)T(I),A(I)
      K1=2
   29 CONTINUE
  207 FORMAT(5(A1,F7.0,F8.0))
  106 FORMAT(16H THE TIME AROUND 2F10.0,18H IS FAULTY. CHECK.)
      RETURN
      END
```

Subroutine SCAL (Lee, Brady, Trifunac)

SCAL is called by Volume I P2SCALE MAIN program to

(1) smooth the timing marks used for the time coordinates

(2) smooth and subtract the fixed trace

(3) scale the acceleration data to units of sec and G/10

(4) fix a horizontal zero line for zero mean value

(5) calculate the RMS of the data and adjust it to minimum when requested.

Usage

```
CALL SCAL(SEN,TMAVE,NFT,NTM,TMS,P2L,WRT,RMS,K2,J)
COMMON/RAWDTA/M, NPB, NP, TM(100), TB(1500), AB(1500),
*        T(9000), A(9000), STAR(9000)
COMMON/RAWFMT/F1(16), F2(11), F3(13), F4(10), F5(12), F6(8),
*        F7(20), F8(20), F9(20)
```

Where

SEN = sensitivity of acceleration in digitizer units/G

TMAVE = average number of digitizer units per second

NFT = 1, if there is only 1 fixed trace for the record
      3, if there is 1 fixed trace for each component

NTM = 0, if there is no timing marks for the record
      1, otherwise

TMS = length in seconds of each interval of digitized timing marks

P2L = 1, if the RMS of the trace is to be adjusted to minimum
      0, otherwise

WRT = 1, if scaled data is read in for printout
      0, otherwise

RMS = RMS of the scaled data

K2 = 2 if the timing marks have been smoothed

J = the number of component to be scaled

COMMON/RAWDTA/M = number of digitized time marks

NPB = number of points in the digitized fixed trace

NP = number of points in the digitized acceleration trace

TM(I) = the timing mark data

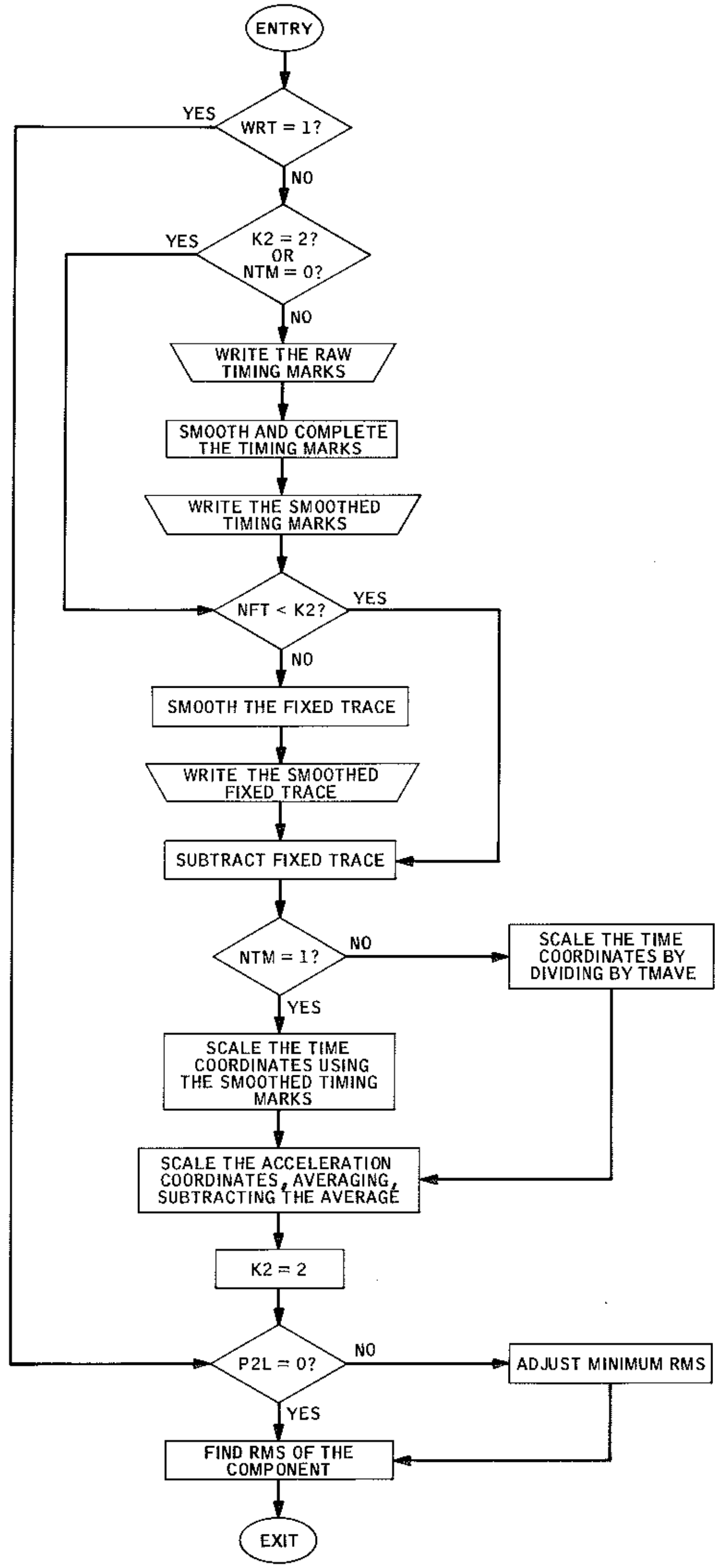
TB(I), AB(I) = fixed trace data

T(I), A(I) = acceleration trace data

STAR(I) = asterisks beside the time coordinates indicating the shifting position of the digitized record (optional)

COMMON/RAWFMT/Format used for printout of the raw data

SUBROUTINE SCAL FLOW CHART



```
      SUBROUTINE SCAL(SEN,TMAVE,NFT,NTM,TMS, P2L, WRT,RMS,K2,J)
C   K2=2 AFTER THE TIMING MARKS HAVE BEEN SMOOTH AND COMPLETED
      COMMON/RAWDTA/M,NPB,NP,TM(700),TB(1500),AB(1500),T(9000),A(9000),
     *STAR(9000)/RAWFMT/F3(13),F4(10),F5(12),F6(18),F7(20),F8(20),F9(20)
      IF( WRT.EQ.1.)GO TO 2
C   SMOOTH & COMPLETE THE TIMING MARKS
      IF(K2.EQ.2.OR.NTM.EQ.0)GO TO 99
      WRITE(6,F4)(TM(I),I=1,M)
      DIGTM=TMAVE*TMS
      NTMMF =DIM(T(NP),TM(M))/DIGTM+1.
      DO 105 I=1,NTMMF
  105 TM(M+I)=TM(M)+I*DIGTM
      M=M+NTMMF
      NTMMS =DIM(TM(1),T(1))/DIGTM+1.
      M1=M+NTMMS
      TM(M1)=TM(M)
      X1=TM(M)
      NSTPS=M-2
      DO 104 I=1,NSTPS
      TM(M1-I)=.5*TM(M-I)+.25*(X1+TM(M-I-1))
  104 X1=TM(M-I)
      TM(NTMMS+1)=TM(1)
      DO 106 I=1,NTMMS
  106 TM(I)=TM(NTMMS +1)-(NTMMS -I+1)*DIGTM
      M=M+NTMMS
      WRITE(6,F6)(TM(I),I=1,M)
C    SMOOTHING THE FIXED TRACE
   99 IF(NPB.NE.0)GO TO 10
      WRITE(6,F8)
      GO TO 110
   10 IF(NFT.LT.K2)GO TO 101
      NSTPS=NPB-1
      X1=AB(1)
      AB(1)=.5*(X1+AB(2))
      X0 =.5*(AB(NPB-1)+AB(NPB))
      DO 1 I=2,NSTPS
      XI=AB(I)
      AB(I)=XI/2.+(X1*(TB(I+1)-TB(I))+AB(I+1)*(TB(I)-TB(I-1)))/2.
     1/(TB(I+1)-TB(I-1))
    1 X1=XI
      AB(NPB)= X0
      WRITE(6,F3)(TB(I),AB(I),I=1,NPB)
      IF(NPB.LE.5)WRITE(6,F8)
C   SUBTRACTING THE FIXED TRACE FROM THE ACCELERATION TRACE
  101 K=2
      DO 5 I=1,NP
    3 IF(T(I).LE.TB(K))GO TO 5
      K=K+1
      IF((K-NPB))3,3,4
    4 K=NPB
    5 A(I)=A(I)-AB(K-1)-(T(I)-TB(K-1))/(TB(K)-TB(K-1))*(AB(K)-AB(K-1))
C   SCALING THE TIME CO-ORDINATES OF THE ACCELERATION TRACE
  110 IF(NTM.EQ.1)GO TO 200
      X1=T(1)
      DO 140 I=1,NP
  140 T(I)=(T(I)-X1)/TMAVE
      GO TO 300
  200 K=2
      DO 32 I=1,NP
   30 IF(T(I).LE.TM(K))GO TO 32
      K=K+1
```

```
      GO TO 30
   32 T (I)=(FLOAT(K-2)+(T (I)-TM(K-1))/(TM(K)-TM(K-1)))*TMS
      X1=T(1)
      DO 335 I=1,NP
  335 T(I)=T(I)-X1
C    SCALING ITS Y CO-ORDINATES, AVERGING IT AND SUBTRACTING THE AVERAGE.
  300 SUM=0.
      DO 6 I=2,NP
    6 SUM=SUM+(A(I-1)+A(I))*(T(I)-T(I-1))
      AV=SUM/2./T(NP)
      DO 7 I=1,NP
    7 A(I)=(A(I)-AV)/SEN*10.
      K2=2
C    FINDING THE RMS, ADJUSTING IT TO MINIMUM WHEN NECESSARY
    2 IF(P2L.EQ.0.)GO TO 9
      V1=0.
      X1=0.
      DO 301 I=2,NP
      DLT=T(I)-T(I-1)
      X1=X1+V1*DLT+DLT**2/6.*(2.*A(I-1)+A(I))
  301 V1=V1+DLT/2.*(A(I-1)+A(I))
      X0=6./T(NP)*X1/T(NP)-2./T(NP)*V1
      XI=6./T(NP)*V1/T(NP)-2./T(NP)*6./T(NP)*X1/T(NP)
      WRITE(6,F7 )X0,XI
      DO 303 I=1,NP
  303 A(I)=A(I)-X0-XI*T(I)
    9 NSTPS=NP-1
      SUM= (A(NP-1)**2+A(NP)*A(NP-1)+A(NP)**2)*T(NP)
      DO 8 I=2,NSTPS
    8 SUM=SUM+T(I)*(A(I-1)-A(I+1)) * (A(I-1)+A(I)+A(I+1))
      RMS = SQRT(SUM/3./T(NP))
      RETURN
      END
```

Subroutine WRTPCH (Lee, Brady)

WRTPCH is called by Volume I P2SCALE MAIN program to

(1) printout scaled data in the format of the Volume I report

(2) printout scaled data in a format suitable for Xerox copying

(3) punch scaled data.

Usage

```
CALL  WRTPCH(NP, T, STAR, A, NTM, NI, PUN, RMS)
COMMON/TITLE/RECTTL(14), DN(2)
COMMON/WRTFMT/W1(20), W2(60), W3(80), W4(40), W5(16),
*       W6(18), W7(60), W8(7), W9(13), W0(15), WF1(5), WF2(9),
*       WF3(4), NEXTPG(2)
```

Where

NP = number of data points

T(I) = time coordinates of scaled data (in sec)

STAR(I) = asterisks indicating shifting position during the digitization process (optional)

A(I) = acceleration coordinates of scaled data (in G/10)

NTM = 1, if record has timing marks
      0, otherwise

NI = number of lines per page of printout

PUN = 0, if scaled data is to be punched
      1, otherwise

RMS = RMS of the scaled data

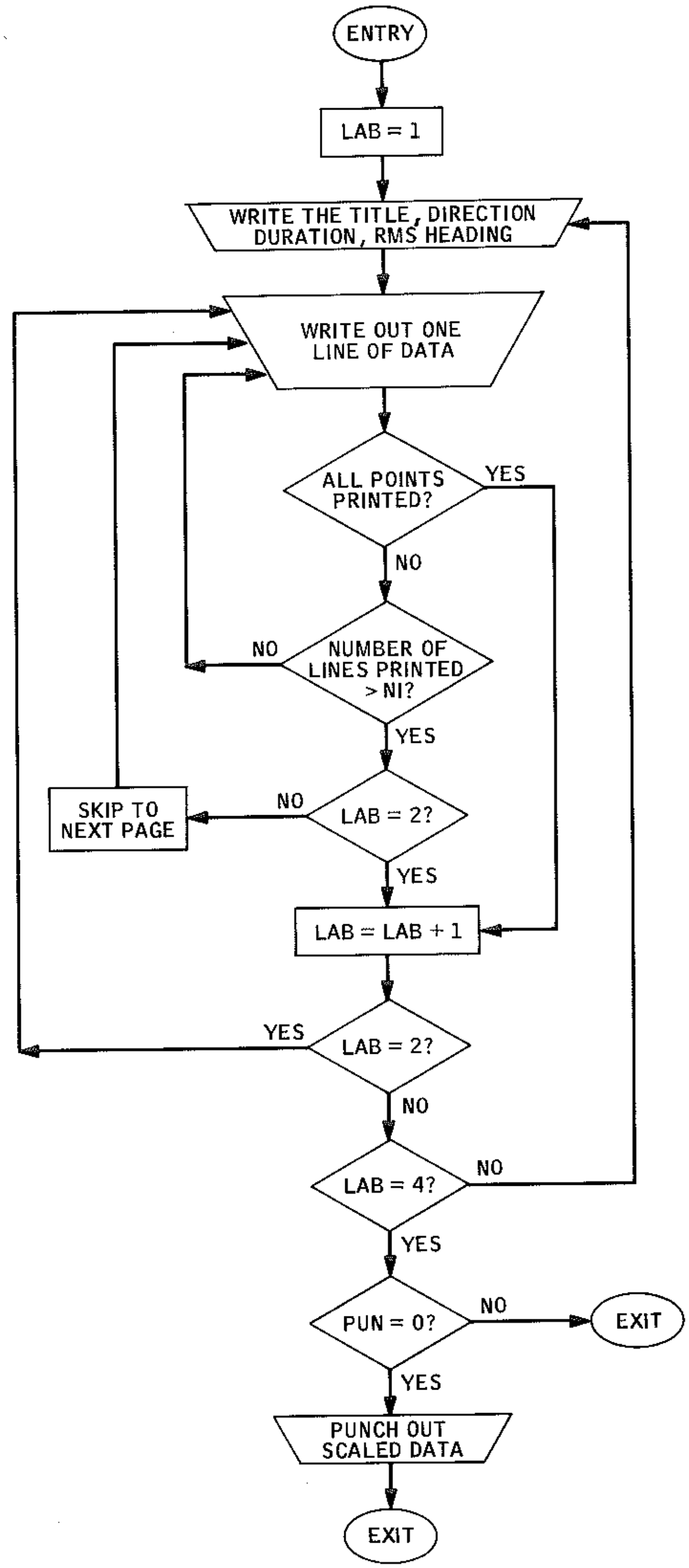
COMMON/TITLE

RECTTL(14) = main title with earthquake time and location

DN(2) = direction of the component

WRTFMT = format used for printout and punched output

SUBROUTINE WRTPCH FLOW CHART



```
      SUBROUTINE WRTPCH(NP,T,STAR,A,NTM,NI,PUN,RMS)
      DIMENSION T(1),A(1),STAR(1)
      COMMON/TITLE/RECTTL(14),DN(2)
      COMMON/WRTFMT/W1(20),W2(60),W3(80),W4(40),W5(16),W6(18),W7(6)
     1,W8(7),W9(13),W0(15),WF1(5),WF2(9),WF3(4),NEXTPG(2)
C     LAB=1,  WRITE-OUT OF SCALED DATA OF REPORT SIZE
C     LAB=2,   CONTINUATION AND LAST PAGE OF THE ABOVE WRITE-OUT
C     LAB=3,  WRITE-OUT OF SCALED DATA IN XEROX COPY SIZE
      DO 14 LAB=1,3
      IF(LAB.EQ.2)GO TO 2
      WRITE(6,W0)RECTTL,DN,NP,T(NP)
      IF(NTM.EQ.0)WRITE(6,W2 )
      IF(NTM.EQ.1)WRITE(6,W3 )
      WRITE(6,W4) RMS
      NL=17
      I4=0
      IF(LAB.EQ.1)WRITE(6,W9)
      IF(LAB.EQ.3)WRITE(6,W8)
    1 I1=I4+1
    2 I3=I1+4*(1/LAB)*(NI-NL+1)
      IF(I3.GT.NP)GO TO 14
   10 I2=I1+3
   11 I4=MINO(I3+3,NP)
      IF(LAB.EQ.1)WRITE(6,WF2)(STAR(I),T(I),A(I),I=I1,I2),
     *                        (STAR(I),T(I),A(I),I=I3,I4)
      IF(LAB.EQ.2)WRITE(6,WF2)(STAR(I),T(I),A(I),I=I3,I4)
      IF(LAB.EQ.3)WRITE(6,WF1)(STAR(I),T(I),A(I),I=I3,I4)
      NL=NL+1
      I1=I2+1
      I3=I4+1
      IF(I3.GT.NP)GO TO 14
      IF(NL.LE.NI)GO TO 10
      IF(LAB.NE.2)WRITE(6,NEXTPG)
      NL=1
      GO TO(1,14,11),LAB
   14 CONTINUE
   16 IF(PUN.NE.0.)RETURN
      IF(NTM.EQ.0)PUNCH W5, RECTTL,DN
      IF(NTM.EQ.1)PUNCH W6, RECTTL,DN
      PUNCH W7
      NL=0
      DO 20 I1=1,NP,5
      I2=I1+4
      NL=NL+1
   20 PUNCH WF3,(T(I),A(I),I=I1,I2),NL
      PUNCH W1,NL,NP,T(NP)
      RETURN
      END
```

SAMPLE OF INPUT DATA FOR P2SCALE PROGRAM

2+4 66.1 CHOLAME-SHANDON #2 - PICHECK OF RAW DIGITIZED DATA

00000	00000	00316	00000	00615	00000	00923	00000	01231	00000
01540	00000	01852	00000	02160	00000	02468	00000	02779	00000
03086	00000	03396	00000	03712	00000	04016	00000	04318	00000
04634	00000	04940	00000	05251	00000	05558	00000	05871	00000
06181	00000	06488	00000	06792	00000	07104	00000	07419	00000
07727	00000	08038	00000	08344	00000	08651	00000	08964	00000
09269	00000	09580	00000	09884	00000	10199	00000	10508	00000
10810	00000	11126	00000	11435	00000	11750	00000	12056	00000
12362	00000	12668	00000	12978	00000	13285	00000	13598	00000
13905	00000	14214	00000	14528	00000	14834	00000	15147	00000
15452	00000	15768	00000	16082	00000	16384	00000	16683	00000
16995	00000	17305	00000	17617	00000	17924	00000	18240	00000
18544	06368	18854	06368	19162	06368	19473	06368	19788	06368
20095	06367	20402	06367	20707	06367	21015	06367	21332	06367
21638	06367	21947	06367	22255	06367	22563	06367	22870	06367
23185	06367	23486	06367	23796	06367	24110	06367	24419	06367
24726	06367	25040	06367	25345	06367	25651	06367	25963	06367
26274	06367	26587	06367	26892	06367	27203	06367		
666666									
00000	00000	00138	-00034	00167	00020	00191	-00007	00222	00023
00235	00009	00260	-00009	00287	-00042	00306	-00016	00328	-00004
00350	00053	00380	-00069	00416	00043	00441	00002	00458	00036
00482	-00003	00510	-00017	00534	-00007	00553	00023	00583	-00007
00603	00006	00622	00033	00651	00011	00675	-00017	00695	-00065
00716	-00027	00729	00005	00747	00023	00766	00039	00795	-00003
00816	00026	00835	00040	00851	00020	00875	-00018	00903	-00022
00920	-00058	00934	-00003	00952	00039	00979	-00001	01003	00034
01027	-00015	01044	00018	01066	-00012	01101	00073	01123	00007
01134	00041	01167	-00097	01194	-00023	01215	-00100	01234	-00024
01264	-00123	01282	-00069	01313	-00189	01333	-00071	01351	-00020
01371	00113	01390	00019	01404	-00134	01427	00105	01446	-00042
01470	00205	01474	00279	01504	-00018	01520	00072	01541	-00026
01556	00155	01572	00253	01583	00147	01599	00060	01624	00159
01640	00086	01659	00061	01691	-00223	01719	00089	01741	-00104
01759	-00167	01780	-00313	01808	-00123	01832	-00281	01865	-00026
01895	-00197	01926	-00013	01936	-00073	01959	00040	01978	00166
01998	00298	02016	00131	02030	-00045	02039	-00165	02051	-00215
02067	-00163	02099	-00300	02125	-00259	02150	-00307	02181	-00209
02222	-00319	02242	-00285	02266	-00250	02298	-00318	02316	-00251
02331	-00144	02344	-00027	02360	00207	02374	00388	02396	00667
02411	00859	02430	01097	02452	01252	02475	00941	02502	00469
02518	00142	02531	-00080	02551	-00339	02570	-00650	02591	-00905
02607	-01075	02623	-00629	02632	-00146	02640	00068	02644	00122
02654	00149	02663	00332	02676	00391	02690	00641	02700	00734
02705	00798	02717	00875	02726	00982	02738	01077	02746	01133
02762	00977	02770	00886	02789	00820	02796	00661	02808	00540
02815	00485	02828	00534	02850	00438	02866	00502	02878	00402
02895	00066	02909	-00247	02919	-00429	02942	-00491	02952	-00564
02964	-00640	02979	-00704	02997	-00755	03011	-00798	03025	-00741
03039	-00576	03052	-00389	03059	-00264	03071	-00184	03087	-00298
03097	-00405	03106	-00497	03119	-00566	03131	-00507	03142	-00349
03152	-00213	03154	-00164	03164	-00131	03174	-00067	03190	00004
03197	00053	03216	00113	03233	00045	03246	-00018	03260	-00094
03276	-00176	03293	-00104	03301	-00041	03314	00061	03326	00134
03348	00199	03371	00249	03388	00272	03414	00186	03436	00244
03454	00167	03470	00006	03481	-00059	03496	-00106	03514	-00047
03530	00030	03552	00101	03566	00032	03579	-00025	03602	-00108
03622	-00148	03648	-00139	03666	-00156	03698	-00089	03727	-00155
03756	-00033	03792	-00136	03821	-00085	03838	-00042	03847	00017
03855	00064	03862	00089	03878	00110	03902	00081	03912	00043

03926	00003	03938	00035	03956	-00032	03968	-00079	03986	-00123
04004	-00073	04037	00055	04058	00126	04090	00199	04105	00242
04118	00266	04139	00215	04160	-00024	04172	-00108	04193	-00188
04210	-00121	04227	00042	04246	00138	04264	00114	04287	00105
04306	00132	04319	00177	04335	00201	04359	00170	04378	00130
04392	00030	04407	-00045	04432	-00114	04466	-00084	04503	-00058
04523	-00119	04542	-00184	04568	-00215	04594	-00164	04608	-00103
04628	-00075	04654	-00059	04679	-00088	04698	-00135	04726	-00190
04751	-00238	04774	-00163	04789	-00068	04806	-00017	04842	00075
04876	00107	04907	00165	04930	00209	04958	00201	04967	00187
04987	00216	05036	00159	05074	00231	05091	00188	05118	00081
05141	-00006	05154	-00052	05191	-00059	05234	-00138	05271	-00109
05315	-00200	05334	-00181	05352	-00060	05364	-00000	05375	00028
05402	00042	05434	00022	05463	00030	05494	00005	05511	-00034
05527	-00069	05546	-00086	05571	-00053	05595	00014	05611	00065
05635	00110	05658	00078	05674	00027	05686	-00002	05714	-00010
05747	00035	05755	00062	05779	00092	05804	00063	05831	00050
05863	00027	05891	00008	05911	-00023	05932	-00061	05948	-00094
05966	-00117	05983	-00136	05991	-00125	06015	-00082	06034	-00025
06057	00089	06073	00146	06100	00181	06131	00143	06154	00073
06167	00031	06183	00011	06235	00035	06284	-00041	06355	00007
06416	-00035	06466	-00065	06525	-00098	06599	-00038	06631	00042
06660	00089	06692	00115	06727	00140	06755	00112	06771	00066
06795	00026	06812	-00003	06840	-00020	06858	-00007	06880	-00021
06915	00037	06933	00029	06958	00060	07007	-00005	07057	-00022
07106	-00075	07167	00044	07206	-00015	07247	00065	07283	00111
07316	00095	07346	00027	07376	00011	07417	-00011	07448	-00052
07492	-00090	07544	-00094	07571	-00054	07593	-00002	07612	00032
07627	00041	07647	00008	07664	-00041	07692	-00080	07713	-00099
07743	-00069	07791	-00027	07819	00033	07846	00066	07867	00088
07911	00064	07954	00088	08002	00106	08055	00103	08104	00075
08127	00079	08163	00044	08191	00022	08227	00035	08263	-00007
08285	-00053	08344	-00067	08367	-00108	08384	-00165	08414	-00221
08439	-00156	08458	-00069	08472	00003	08488	00054	08516	00008
08531	-00045	08543	-00118	08560	-00160	08571	-00106	08586	-00039
08598	00048	08611	00115	08630	00153	08656	00070	08668	-00025
08685	-00095	08702	-00054	08722	00036	08738	00112	08762	00146
08785	00123	08804	00089	08820	00039	08832	00017	08859	00016
08886	00037	08906	00068	08919	00083	08951	00045	08968	00018
09007	00011	09035	00040	09071	00008	09097	-00016	09126	-00019
09140	-00002	09175	00021	09204	-00005	09243	-00072	09267	-00060
09308	-00070	09338	-00046	09368	-00004	09394	00015	09448	00008
09498	00035	09535	00057	09570	00048	09630	00106	09691	00018
09752	00050	09781	00034	09810	00045	09835	-00003	09894	-00059
09918	-00026	09954	-00001	10007	-00043	10060	-00073	10107	-00027
10126	00034	10165	00054	10184	00040	10228	00085	10280	00061
10311	00034	10337	00034	10353	00022	10400	-00005	10450	-00057
10496	00021	10535	00005	10570	00021	10593	00011	10617	00034
10672	-00026	10739	00006	10779	-00035	10832	00009	10882	-00051
10938	00018	11012	-00013	11046	00042	11120	00013	11148	00049
11198	00019	11230	00045	11258	00065	11275	00079	11289	00054
11323	00083	11351	00062	11383	00031	11416	00008	11461	00009
11482	00009	11508	-00016	11542	-00050	11595	-00059	11632	-00055
11679	-00047	11731	-00012	11818	-00004	11855	00025	11886	00002
11927	-00017	11979	00026	12024	00007	12063	00016	12106	-00003
12135	-00006	12190	00040	12269	00000	12316	00057	12383	-00042
12424	00065	12454	00093	12480	00054	12513	00009	12542	00038
12576	00061	12605	00026	12640	-00014	12704	00009	12762	-00023
12834	00037	12902	00023	12939	00037	13010	00021	13069	00029
13121	00003	13160	00024	13223	00019	13291	-00014	13371	-00027
13458	-00014	13520	-00006	13557	00023	13620	00006	13678	00033
13787	00013	13859	00029	13912	-00001	13956	00022	14010	00037

14048	00030	14087	00010	14123	-00003	14167	00015	14190	00022
14242	00003	14303	-00011	14357	-00016	14396	-00043	14451	-00028
14498	-00004	14552	00031	14612	00020	14662	00031	14682	00046
14728	00031	14770	00055	14806	00054	14836	00040	14874	00037
14902	00016	14943	00021	14983	00003	15018	-00019	15100	-00059
15158	-00043	15200	-00008	15227	00025	15258	00035	15328	00021
15384	00029	15443	00035	15492	-00013	15523	00013	15571	00006
15607	-00041	15647	00007	15694	00011	15728	00029	15770	00021
15808	00046	15844	00033	15888	00039	15914	00048	15960	00035
16014	00027	16079	00030	16141	00028	16188	00004	16242	-00009
16347	-00036	16415	-00055	16487	-00018	16538	00026	16611	00010
16684	00021	16767	00005	16823	00025	16864	00043	16920	00029
16962	00018	17019	00024	17069	00034	17098	00042	17150	00018
17188	-00005	17246	-00009	17306	-00032	17385	00008	17455	-00006
17505	00026	17566	00003	17625	00010	17676	-00001	17731	00026
17787	00002	17835	00033	17891	00007	17990	00001	18052	-00005
18106	00017	18168	-00001	18214	00019	18243	00033	18299	00025
18368	00019	18422	00026	18459	00009	18511	-00003	18542	-00007
18611	-00023	18654	-00034	18707	-00019	18748	-00001	18804	00017
18846	00018	18878	00022	18922	00039	18968	00039	18996	00026
19034	00012	19062	00002	19091	00011	19151	00001	19203	00015
19273	00023	19324	00032	19382	00002	19421	00010	19442	00020
19485	00010	19570	-00001	19632	-00003	19731	00004	19758	00011
19812	00001	19879	00012	19915	-00004	19979	00020	20011	00036
20044	00037	20088	00028	20125	00019	20173	00026	20218	00013
20259	00006	20314	00006	20359	-00003	20412	00017	20496	00011
20535	00024	20578	00019	20615	00026	20668	00033	20724	00046
20780	00020	20827	00031	20879	00014	20935	00003	21024	-00001
21104	00004	21192	00006	21272	00025	21356	00023	21444	00022
21500	00007	21542	00011	21571	00002	21631	00011	21692	00017
21728	00024	21787	00017	21844	00024	21879	00034	21925	00023
21982	00020	22045	00002	22112	00008	22166	00015	22226	00007
22275	00009	22318	00012	22364	00018	22431	00009	22492	00004
22556	00001	22637	00010	22710	00011	22758	00029	22820	00026
22863	00036	22907	00025	22967	00043	23021	00021	23072	00010
23127	00018	23215	-00000	23286	00003	23359	-00002	23434	00007
23500	00002	23541	00022	23598	00017	23649	00023	23716	00013
23835	00012	23899	00008	23960	00002	24046	00025	24129	00015
24230	00005	24306	00011	24386	00004	24450	00020	24504	00019
24592	00016	24697	00021	24788	00008	24854	00012	24943	00018
25006	00026	25075	00014	25130	00020	25204	00007	25276	00007
25367	-00001	25456	00016	25536	00021	25592	00018	25655	00011
25731	00018	25779	00019	25823	00013	25873	00020	25942	00014
26002	00017	26051	00028	26139	00017	26208	00025	26314	00010
26404	00014	26448	00022	26528	00007	26578	00017	26642	00007
26687	00016	26738	-00001	26844	00009	26916	00020	26970	00011
27034	00021	27110	00011	27196	00017				
666666									
00000	00000	00312	00004	00924	00001	01547	00002	02163	00003
02779	00002	03404	00001	04023	00005	04635	00007	05251	00007
05871	00006	06488	00007	07106	00008	07723	00010	08347	00013
08960	00011	09583	00009	10200	00015	10816	00012	11436	00011
12056	00014	12669	00018	13290	00012	13913	00014	14527	00011
15147	00008	15767	00014	16383	00015	17000	00011	17616	00013
18240	00013	18855	00015	19473	00013	20096	00014	20704	00017
21334	00013	21949	00017	22569	00016	23183	00017	23804	00019
24420	00016	25035	00016	25658	00018	26276	00015	26898	00017
27202	00017								
666666									
00000	00000	00140	-00084	00149	00004	00160	-00028	00174	00019
00190	00091	00226	-00083	00245	00024	00267	-00063	00289	00060

00309	-00073	00328	00044	00339	00002	00344	00053	00370	-00042
00384	00018	00395	00068	00412	00026	00420	00000	00434	-00034
00453	00057	00467	-00038	00484	00066	00513	-00099	00528	00003
00536	-00005	00556	00100	00578	-00191	00596	00018	00606	-00040
00626	00122	00642	00015	00650	00032	00675	-00158	00690	-00058
00704	00026	00712	00142	00728	-00027	00746	-00193	00760	-00024
00775	00069	00803	-00083	00821	00075	00843	-00067	00859	-00123
00871	00002	00882	00128	00899	00075	00908	-00048	00924	00042
00934	-00004	00946	00072	00960	-00006	00976	-00138	00990	00020
00999	00146	01023	-00128	01039	00060	01054	00254	01069	-00067
01079	-00171	01095	00058	01115	-00352	01130	00064	01143	00405
01148	00326	01155	00355	01170	-00028	01178	-00278	01187	-00422
01198	-00152	01209	-00305	01236	00556	01259	-00201	01282	00424
01299	-00213	01313	-00463	01324	-00081	01335	00267	01352	-00028
01370	-00263	01384	00054	01406	-00232	01430	00214	01447	-00084
01462	00220	01487	-00542	01511	00529	01534	-00521	01556	00263
01576	-00069	01595	00270	01609	00108	01617	00151	01639	-00335
01650	-00032	01664	-00000	01682	00222	01706	-00366	01726	00328
01741	-00157	01748	-00116	01771	-00388	01783	-00097	01791	-00000
01804	00299	01823	-00287	01838	-00175	01847	-00232	01861	00222
01871	00764	01884	00078	01903	-00571	01912	00014	01927	00427
01953	00214	01962	-00062	01971	-00176	01990	00342	02003	-00065
02019	-00257	02036	-00101	02046	-00071	02055	-00007	02067	00043
02086	-00135	02104	00117	02110	00076	02120	00202	02134	00082
02152	-00200	02171	00110	02179	00285	02194	00091	02205	-00189
02214	-00342	02235	00124	02250	-00105	02271	-00219	02288	-00034
02305	00022	02327	00264	02338	00114	02360	00001	02379	00243
02393	00008	02404	-00109	02419	-00010	02441	-00237	02456	-00055
02484	-00282	02506	00110	02527	-00290	02551	00271	02566	-00091
02586	-00350	02603	-00100	02611	00031	02622	00168	02643	00049
02655	00146	02672	-00090	02687	-00271	02700	-00104	02710	00104
02723	00252	02740	00169	02756	-00008	02779	00138	02797	00034
02807	-00091	02819	-00171	02847	00122	02856	-00008	02868	-00181
02880	-00258	02914	-00069	02936	-00221	02961	00061	02986	-00157
03011	00065	03018	00021	03042	00193	03059	00108	03083	00208
03114	00134	03128	00230	03138	00175	03148	00226	03165	00158
03173	00237	03202	00013	03211	00085	03235	-00015	03247	00045
03271	-00029	03284	-00096	03292	-00053	03316	-00131	03328	-00079
03341	-00138	03363	-00280	03384	-00188	03402	-00110	03420	-00014
03429	-00056	03442	-00002	03458	-00044	03487	00038	03514	00010
03530	00056	03548	00022	03573	00094	03592	00033	03604	-00020
03611	00006	03635	-00106	03657	-00052	03680	-00100	03697	-00135
03707	-00095	03724	-00018	03734	-00062	03755	-00003	03771	-00061
03786	00021	03799	00085	03834	00042	03847	00096	03861	00126
03875	00165	03901	00098	03919	00147	03932	00191	03946	00239
03967	00194	03983	00134	03995	00181	04018	00118	04034	00033
04057	-00067	04072	-00022	04095	-00088	04111	-00151	04136	-00088
04164	-00180	04194	-00082	04210	-00130	04233	-00054	04245	-00078
04256	-00090	04284	-00036	04304	-00104	04330	-00029	04355	-00079
04382	00036	04396	-00003	04421	00085	04446	00015	04475	00104
04508	00015	04520	00039	04534	00080	04555	00028	04573	00085
04597	00012	04611	00051	04632	00100	04656	00032	04684	00047
04711	00096	04725	00034	04739	-00017	04768	-00028	04785	00003
04798	00038	04824	-00037	04843	-00002	04859	-00053	04887	-00088
04902	-00111	04922	-00089	04946	-00111	04968	-00053	04982	-00077
05007	-00131	05023	-00061	05036	00011	05050	-00031	05064	-00068
05078	-00030	05091	00019	05107	00055	05122	00021	05139	00054
05159	00004	05171	00057	05183	00105	05199	00126	05226	00077
05244	00097	05270	00059	05290	00112	05323	00011	05351	00062
05376	-00044	05404	00020	05427	-00045	05439	-00006	05459	-00046
05480	00016	05510	-00042	05535	00053	05568	-00037	05587	00002
05604	00021	05623	00054	05663	-00015	05682	00013	05712	-00014

05735	-00047	05758	-00024	05783	-00041	05808	-00112	05833	-00027
05855	-00087	05880	-00019	05904	00020	05924	00046	05946	00007
05959	00047	05974	00088	05992	00048	06016	00083	06039	00028
06053	00062	06081	00002	06090	00042	06115	-00020	06136	00030
06174	-00058	06193	-00005	06211	00042	06234	-00011	06256	00034
06268	00005	06290	00051	06323	00022	06351	00032	06361	00053
06379	00018	06411	00074	06450	00024	06463	00044	06498	-00012
06515	00004	06535	-00064	06559	-00022	06584	-00058	06600	-00015
06618	-00055	06634	-00010	06656	-00074	06675	-00029	06697	-00041
06704	-00033	06723	-00093	06750	00011	06764	-00034	06790	00033
06807	-00010	06824	00074	06847	-00023	06869	00115	06896	-00010
06912	00083	06939	-00078	06960	00073	07003	-00009	07039	00048
07054	00012	07067	00049	07092	-00021	07114	00029	07142	00028
07171	00072	07192	00028	07218	00072	07247	00014	07266	-00073
07291	00013	07328	-00084	07351	-00022	07370	-00054	07388	-00017
07402	-00054	07416	-00015	07430	00003	07447	00039	07474	-00015
07505	00037	07522	00002	07544	00024	07558	00002	07582	00044
07606	-00003	07617	00025	07631	-00001	07651	00039	07668	00036
07710	-00035	07740	00025	07762	-00020	07783	00034	07804	-00008
07823	-00026	07838	-00041	07872	00026	07896	-00032	07926	-00004
07946	00002	07962	00034	08001	-00011	08034	00060	08061	00021
08078	00035	08103	00015	08117	00055	08132	00022	08144	00048
08167	00009	08186	00042	08220	-00004	08247	00020	08280	-00034
08306	00016	08342	-00034	08368	00008	08408	-00018	08438	00015
08461	-00014	08487	00051	08518	-00020	08543	00029	08568	-00004
08594	00038	08635	00070	08662	00035	08689	00007	08720	00024
08744	-00002	08775	00016	08786	00027	08817	-00005	08859	00012
08891	-00023	08908	-00005	08932	-00050	08950	-00018	08975	-00046
08998	-00005	09021	00004	09040	-00031	09062	00004	09087	00047
09117	-00013	09134	00038	09158	-00010	09186	00024	09212	00035
09235	00009	09259	00043	09279	-00002	09310	00058	09338	00015
09353	00051	09388	00006	09406	00044	09439	-00010	09459	00020
09489	00026	09510	00035	09535	-00019	09567	00020	09587	00009
09610	-00027	09627	00004	09655	-00029	09694	-00029	09721	00005
09756	-00015	09795	00022	09819	00017	09834	00032	09871	00004
09905	00035	09955	00018	09995	-00005	10043	00048	10067	00011
10091	00062	10122	00010	10140	00048	10170	00010	10218	-00002
10252	-00018	10274	-00007	10307	-00019	10342	00022	10388	-00005
10410	00038	10430	00010	10460	00042	10494	00058	10512	00018
10539	00060	10578	00038	10611	00018	10658	00002	10673	-00006
10710	00027	10744	-00030	10764	00016	10786	-00030	10808	00024
10839	-00044	10875	-00011	10908	-00031	10944	-00011	10990	00010
11011	-00016	11054	00028	11079	00013	11114	00055	11132	00017
11172	00061	11215	00007	11243	00040	11278	-00004	11299	00027
11369	-00026	11399	00014	11423	-00032	11451	00017	11471	-00020
11489	00008	11512	00033	11560	00018	11591	00034	11612	00046
11642	00020	11658	00041	11677	00014	11696	00043	11730	00013
11760	-00007	11792	00018	11830	-00012	11854	00013	11928	-00028
11950	00018	11970	-00013	12010	00017	12047	00028	12070	-00006
12103	00034	12140	00008	12162	00026	12235	00011	12264	00011
12284	00028	12316	-00004	12350	00030	12387	-00005	12431	00023
12476	00004	12535	-00013	12587	00001	12620	-00018	12656	00007
12706	-00029	12755	00020	12810	00031	12883	00012	12966	00001
13082	00025	13144	00024	13250	00014	13297	00032	13330	00010
13364	00022	13399	00013	13426	00028	13475	-00006	13515	00016
13579	-00007	13646	00016	13743	00005	13796	00008	13826	-00004
13854	00016	13922	-00008	14002	00028	14030	00010	14066	00040
14118	00004	14153	00038	14180	00001	14212	00037	14255	-00006
14284	00023	14331	-00006	14420	00012	14463	-00008	14541	00014
14600	-00023	14663	00003	14701	-00002	14782	00044	14828	00017
14874	00035	14934	00010	15024	00024	15076	00035	15147	00014
15205	00015	15237	-00015	15326	-00011	15394	-00012	15482	00011

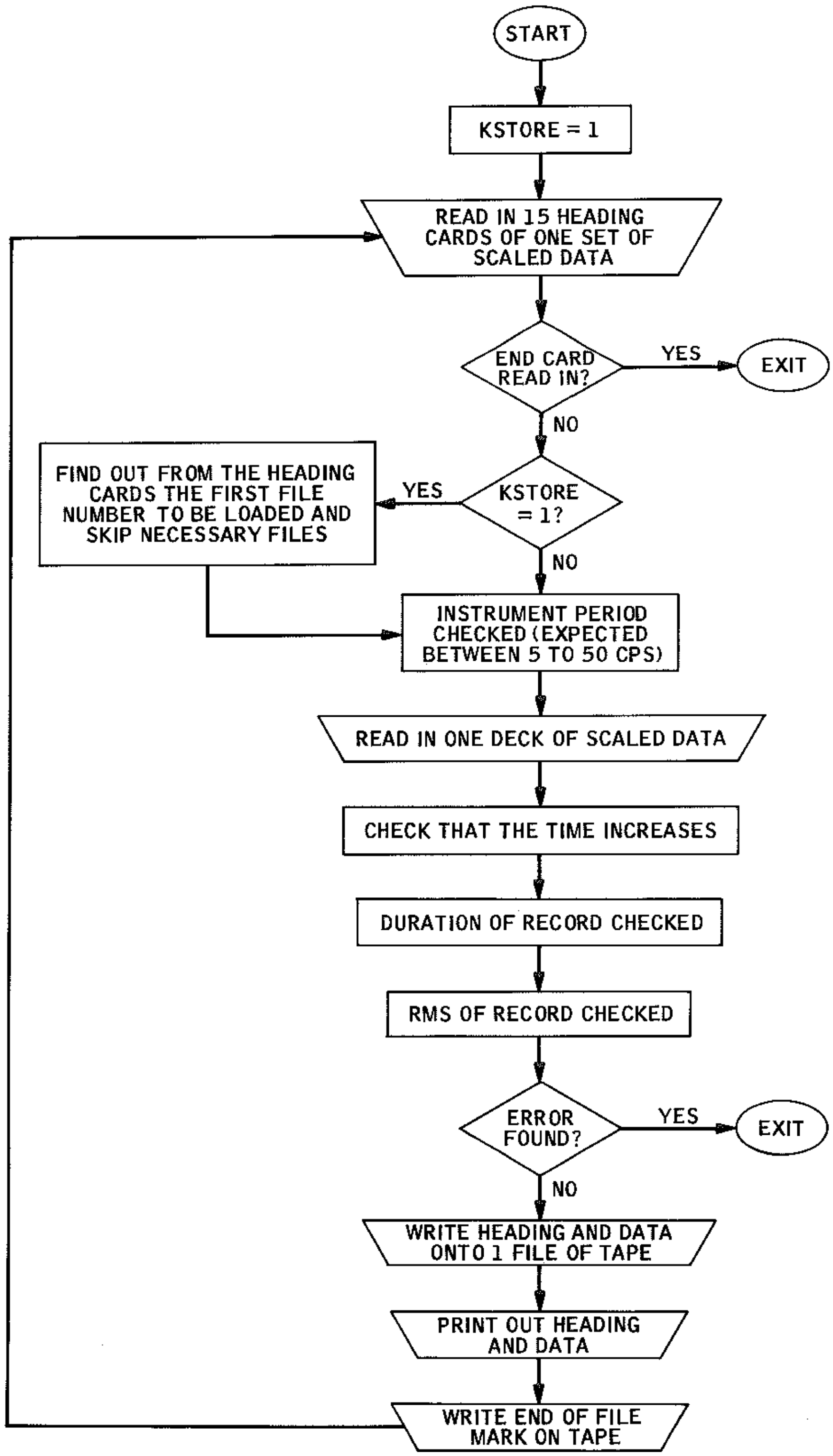
15564	00012	15626	-00002	15683	00026	15778	00016	15880	00001
15981	00017	16089	00027	16190	00038	16270	00015	16327	-00005
16412	-00014	16494	-00015	16553	-00015	16623	00006	16710	00014
16768	00036	16852	00028	16918	00022	16969	00021	17004	00040
17086	00019	17164	00002	17281	-00010	17356	-00017	17456	-00012
17543	00006	17641	00037	17735	00016	17828	00003	17943	-00013
18000	-00000	18088	00004	18144	-00004	18221	00007	18334	00016
18420	00026	18498	00017	18597	00002	18691	-00007	18800	00001
18870	00004	18924	00010	19011	00009	19084	00008	19139	00005
19210	-00001	19279	-00009	19327	-00002	19395	00004	19466	00015
19522	00015	19582	00004	19658	-00001	19720	-00002	19799	00003
19858	00011	19917	00007	19988	00009	20055	00009	20120	00009
20167	00010	20237	00000	20308	-00006	20452	00007	20546	00007
20619	00006	20670	00003	20747	00017	20838	00000	20922	-00008
20954	00003	20991	-00011	21033	00003	21100	-00015	21186	00007
21275	00009	21352	00011	21448	00008	21564	00006	21675	00006
21757	00012	21831	00005	21863	00009	21924	-00008	21993	-00004
22112	-00001	22182	-00003	22250	-00008	22366	-00004	22480	-00000
22539	00010	22604	00013	22688	00006	22756	00007	22846	00012
22939	-00003	23030	00002	23094	00001	23136	-00007	23209	-00008
23272	00005	23367	-00001	23464	00010	23596	00004	23679	00005
23822	00005	23958	00008	24076	00004	24203	00000	24340	00007
24446	00003	24543	00010	24616	00010	24680	00009	24791	00000
24899	-00005	24999	00010	25098	00009	25223	00013	25324	00003
25446	-00001	25563	-00006	25666	00000	25770	00005	25890	00010
25931	00003	25994	00008	26063	00004	26146	-00008	26216	-00003
26282	00002	26370	-00000	26430	-00003	26511	00000	26586	00004
26643	-00002	26699	00007	26744	00004	26808	00007	26910	00006
26989	00001	27087	00005	27198	-00003				
666666									
00002	00000	00318	00002	00925	00001	01540	-00002	02164	00004
02780	00002	03396	00005	04027	00004	04634	00007	05253	00009
05867	00008	06487	00008	07100	00009	07725	00012	08343	00012
08968	00011	09575	00010	10202	00012	10819	00010	11433	00013
12056	00014	12672	00013	13282	00013	13907	00015	14531	00016
15151	00014	15770	00013	16383	00013	16994	00014	17622	00015
18238	00012	18851	00012	19476	00017	20098	00018	20715	00017
21327	00018	21948	00019	22563	00016	23182	00020	23802	00018
24419	00017	25034	00015	25654	00015	26274	00018	26897	00013
27196	00013								
666666									

P3TAPE MAIN Program (Vijayaraghavan, Justiss)

This program reads in the decks of scaled data preceded by the heading cards containing all the pertinent information and loads them onto an assigned magnetic tape.

The input data for this program consists of a sequence (at most 30) of scaled acceleration data which are punched by the P2SCALE program. The heading cards for each component are described at the beginning of this section.

P3 TAPE FLOW CHART



```
C      K1=1  FILE O.K.,  K1=2  ERROR FOUND, PROGRAM STOPS.
       DIMENSION CARMOD(300),IR(100),FR(50)
       DIMENSION CARD(260),FCR(5),ICR(20),IDR(10)
       DIMENSION TX(10000),X(10000)
       DIMENSION ET1S(20)
       NLPAGE=56
       EQUIVALENCE (CARD(1),CARMOD(1)),(IR(1),ICR(1)),
     2  (IR(21),IDR(1)),(FR(1),FCR(1))
       DATA CARMOD/300*0.0/,IR/100*0/,FR/50*0.0/
       K1=1
       JST=1
       DO 100 KSTORE=1,30
       IF(JST.NE.1)CALL WRTNF(10)
       WRITE (6,101)
   101 FORMAT (1H1)
       READ(5,1,END=950)CARD
     1 FORMAT(20A4)
       READ(5,2)ICR
     2 FORMAT(20I4)
       READ(5,3)IDR,FCR
     3 FORMAT(10I4,5F8.4)
       WRITE (6,154) (CARD(KK),KK=81,100)
   154 FORMAT (1H0,' STATION NO. :  ',20A4,//)
       WRITE (6,155) (CARD (KK),KK=101,120)
   155 FORMAT (//1H0,'   ADDRESS IS :   ',20A4,//)
       IF(KSTORE.NE.1)GO TO 102
C    SKIP THE NECESARY FILES
       NFILE=IR(1)-1
       IF(NFILE.EQ.0)GO TO 102
       KSKFL=NFILE-1
    66 READ(10,END=68)
       GO TO 66
    68 CALL READNF(10,KSKFL)
   102 CONTINUE
       NFILE=NFILE+1
C             FILE NUMBER CHECK.
       IF (IR(1).EQ.NFILE ) GO TO 6
       WRITE (6,151) IR(1),NFILE
   151 FORMAT(//'  FILE NUMBER',I4,' SHOULD BE FILE',I4)
       K1=2
     6 CONTINUE
C             INSTRUMENT PERIOD CHECK.
C   EXPECTED NATURAL INSTR. PERIOD :  BETWEEN 5 TO 50 CPS
       IF (FR(1).LT.0.2.AND.FR(1).GT.0.02 ) GO TO 7
       WRITE (6,152) FR(1),NFILE
       K1=2
   152 FORMAT (/// 1H0,' INSTRUMENT PERIOD = ',E12.5,5X' ON FILE  ',I5,/)
     7 CONTINUE
       NDATA=IR(28)
       IF(NDATA.EQ.0)GO TO 999
       READ(5,5)(TX(J),X(J),J=1,NDATA)
     5 FORMAT(10F7.3)
C         CHECK THAT THE TIME INCREASES
       DO 937 J=2,NDATA
       IF(TX(J-1).LE.TX(J))GO TO 937
       K1=2
       WRITE(6,936)TX(J)
   936 FORMAT('  THE TIME AROUND',F7.3,' DOES NOT INCREASE.')
   937 CONTINUE
C             DURATION OF RECORD CHECK.
       CHECK=ABS(TX(NDATA)-FR(3))
```

```
      IF (CHECK.LT.0.001) GO TO 8
      WRITE (6,153) TX(NDATA),FR(3)
      K1=2
  153 FORMAT (////,1H0,'  RECORD DURATION = ',F12.5,5X ,
     A   ' BUT THE EXPECTED DURATION IS  ',F12.5,//)
    8 CONTINUE
C     RMS OF RECORD CHECKED
      NSTPS=NDATA-1
      SUM=(X(NSTPS)**2+X(NDATA)*X(NSTPS)+X(NDATA)**2)*TX(NDATA)
      DO 357 J=2,NSTPS
  357 SUM=SUM+TX(J)*(X(J-1)-X(J+1))*(X(J-1)+X(J)+X(J+1))
      RMS=SQRT(SUM/3./TX(NDATA))
      IF(ABS(RMS-FR(4)).LT.0.001)GO TO 999
      WRITE(6,358)RMS,FR(4)
      K1=2
  358 FORMAT(//'  RMS CALCULATED :',F7.4,'  RMS RECORDED :',F7.4)
  999 CONTINUE
      IF(K1.EQ.2)GO TO 951
      WRITE(10)CARMOD,IR,FR
      IF(NDATA.EQ.0)GO TO 100
      WRITE(10)(TX(J),X(J),J=1,NDATA)
      WRITE (6,156) NFILE
  156 FORMAT (////1H0,'*** COMPLETED FILE ',1XI5)
      JST=JST+1
      WRITE(6,20)
   20 FORMAT(1H1,12(/))
      WRITE(6,21)(CARMOD(K),K=1,20)
   21 FORMAT(2X,20A4,/)
      WRITE(6,22)(CARMOD(K),K=21,60)
   22 FORMAT(2X,20A4)
      WRITE(6,24)
   24 FORMAT(///)
      WRITE(6,22)(CARMOD(K),K=61,300)
      WRITE(6,24)
      WRITE(6,26)(IR(K),K=1,30)
   26 FORMAT(7X,10I6)
      WRITE(6,23)
   23 FORMAT(//)
      WRITE(6,30)(FR(K),K=1,5)
   30 FORMAT(10X,5(F7.3,3X))
      IDTA=160
      WRITE(6,32)
   32 FORMAT(1H1,10(/))
      WRITE(6,33)
      WRITE(6,34)(TX(K),X(K),K=1,IDTA)
      JB=IDTA+1
   40 CONTINUE
      IF(JB.GT.NDATA)JB=NDATA
      JE=JB+NLPAGE*4-1
      IF(JE.GT.NDATA)JE=NDATA
      WRITE(6,36)
   36 FORMAT(1H1,/)
      WRITE(6,33)
   33 FORMAT(6X,4HTIME,4X,5HACCLN,5X,4HTIME,3X,5HACCLN,6X,
     2  4HTIME,4X,5HACCLN,4X,4HTIME,4X,5HACCLN,/)
      WRITE(6,34)(TX(K),X(K),K=JB,JE)
   34 FORMAT(4X,F7.3,1X,F7.3,2X,F7.3,1X,F7.3,4X,
     2  F7.3,1X,F7.3,2X,F7.3,1X,F7.3)
      IF(JE-NDATA)38,100,100
   38 JB=JE+1
      GO TO 40
  100 CONTINUE
```

```
950 CONTINUE
    CALL ENDMF(10)
    WRITE(6,150)NFILE,NDTAOT
150 FORMAT (1H1,////,20X,' UP THROUGH FILE NO. ',I2,10X,
   A    'TOTAL DATA = ',I9)
951 CONTINUE
    IF(K1.EQ.2)WRITE(6,166)
166 FORMAT (//,' **********   ERROR   **********',/)
    STOP
    END
```

SAMPLE OF INPUT DATA FOR P3TAPE PROGRAM

```
FILE  37  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-B, EERL 70-21
PARKFIELD, CALIFORNIA EARTHQUAKE
JUN 27, 1966 - 2026 PST
IB033    66.001.0 T                                                       18
STATION NO. 013    35 43 35N,120 17 13W                                   38
CHOLAME,SHANDON, CALIFORNIA ARRAY NO. 2                                   39
COMP N65E                                                                 9
PARKFIELD, CALIFORNIA EARTHQUAKE   JUNE 27, 1966 - 2026 PST               59
EPICENTER  35 54 00N,120 54 00W                                           31
INSTR PERIOD = 0.0640 SEC  DAMPING = 0.574                                42
NO. OF POINTS = 747   DURATION = 43.77 SEC                                42
UNITS ARE SEC AND G/10.                                                   23
RMS ACCLN OF COMPLETE RECORD = 0.5221 G/10.                               43
  37   1   2  33  66   1   0   3  13  35  43  35-120  17  13  35  54  0-120  54
   0   6  2719662026   0  65 747  32  39  0.0640  0.5740 43.7700  0.5221  0.1000
  0.0   -0.138  0.047  0.081  0.086 -0.028  0.137  0.093  0.158  0.036      1
  0.199 -0.037  0.243 -0.171  0.274 -0.065  0.310 -0.016  0.346  0.215      2
  0.396 -0.281  0.455  0.175  0.496  0.008  0.524  0.146  0.564 -0.012      3
  0.610 -0.069  0.649 -0.028  0.680  0.093  0.730 -0.028  0.763  0.024      4
  0.794  0.134  0.841  0.045  0.881 -0.069  0.913 -0.264  0.948 -0.110      5
  0.969  0.021  0.998  0.094  1.029  0.159  1.077 -0.012  1.111  0.106      6
  1.142  0.163  1.168  0.082  1.208 -0.073  1.253 -0.089  1.281 -0.235      7
  1.304 -0.012  1.333  0.159  1.377 -0.004  1.416  0.139  1.455 -0.061      8
  1.482  0.074  1.516 -0.048  1.575  0.297  1.611  0.029  1.628  0.167      9
  1.682 -0.394  1.726 -0.093  1.760 -0.406  1.791 -0.097  1.839 -0.499     10
  1.868 -0.280  1.918 -0.768  1.951 -0.288  1.980 -0.081  2.012  0.460     11
  2.043  0.078  2.065 -0.544  2.102  0.427  2.133 -0.170  2.172  0.833     12
  2.178  1.134  2.227 -0.073  2.253  0.293  2.287 -0.105  2.311  0.630     13
  2.337  1.028  2.354  0.598  2.380  0.244  2.420  0.646  2.446  0.350     14
  2.477  0.248  2.528 -0.906  2.573  0.362  2.609 -0.423  2.638 -0.679     15
  2.672 -1.272  2.717 -0.500  2.756 -1.142  2.809 -0.106  2.857 -0.801     16
  2.908 -0.054  2.924 -0.297  2.961  0.162  2.992  0.674  3.024  1.210     17
  3.053  0.531  3.076 -0.184  3.090 -0.672  3.110 -0.875  3.136 -0.664     18
  3.187 -1.220  3.230 -1.054  3.270 -1.249  3.320 -0.851  3.387 -1.298     19
  3.419 -1.159  3.458 -1.017  3.510 -1.293  3.539 -1.021  3.563 -0.586     20
  3.584 -0.111  3.610  0.840  3.633  1.576  3.668  2.709  3.693  3.490     21
  3.723  4.457  3.759  5.087  3.796  3.823  3.840  1.905  3.866  0.577     22
  3.887 -0.325  3.919 -1.378  3.950 -2.641  3.984 -3.678  4.010 -4.368     23
  4.036 -2.556  4.050 -0.593  4.063  0.276  4.070  0.496  4.086  0.605     24
  4.100  1.349  4.121  1.589  4.144  2.605  4.160  2.983  4.168  3.243     25
  4.188  3.556  4.202  3.990  4.222  4.377  4.234  4.604  4.260  3.970     26
  4.273  3.601  4.304  3.332  4.315  2.686  4.335  2.195  4.346  1.971     27
  4.367  2.170  4.403  1.780  4.429  2.040  4.448  1.634  4.476  0.268     28
  4.498 -1.003  4.515 -1.743  4.552 -1.995  4.568 -2.292  4.587 -2.600     29
  4.612 -2.861  4.641 -3.068  4.664 -3.243  4.686 -3.011  4.709 -2.341     30
  4.730 -1.581  4.741 -1.073  4.761 -0.748  4.787 -1.211  4.803 -1.646     31
  4.817 -2.020  4.838 -2.300  4.857 -2.060  4.875 -1.418  4.891 -0.866     32
  4.894 -0.667  4.910 -0.533  4.927 -0.272  4.952  0.016  4.964  0.215     33
  4.994  0.459  5.022  0.183  5.042 -0.073  5.065 -0.382  5.091 -0.716     34
  5.118 -0.423  5.131 -0.167  5.152  0.247  5.171  0.544  5.207  0.808     35
  5.244  1.011  5.271  1.105  5.313  0.755  5.348  0.990  5.377  0.677     36
  5.403  0.023  5.420 -0.241  5.444 -0.433  5.473 -0.193  5.499  0.119     37
  5.534  0.408  5.557  0.127  5.578 -0.105  5.614 -0.442  5.647 -0.605     38
  5.688 -0.569  5.717 -0.638  5.769 -0.367  5.816 -0.635  5.863 -0.140     39
  5.922 -0.559  5.969 -0.352  5.997 -0.178  6.011  0.062  6.024  0.253     40
  6.036  0.354  6.062  0.439  6.101  0.321  6.117  0.167  6.140  0.004     41
  6.160  0.134  6.189 -0.139  6.209 -0.330  6.238 -0.509  6.267 -0.306     42
  6.321  0.214  6.356  0.502  6.408  0.798  6.432  0.972  6.454  1.070     43
  6.488  0.862  6.522 -0.109  6.542 -0.451  6.576 -0.776  6.604 -0.504     44
  6.632  0.158  6.663  0.548  6.692  0.450  6.730  0.413  6.761  0.523     45
  6.782  0.706  6.808  0.803  6.847  0.677  6.877  0.514  6.900  0.107     46
```

6.924	-0.198	6.964	-0.478	7.019	-0.357	7.079	-0.252	7.111	-0.500	47
7.142	-0.764	7.184	-0.891	7.226	-0.684	7.248	-0.436	7.281	-0.323	48
7.323	-0.258	7.363	-0.375	7.394	-0.567	7.439	-0.790	7.479	-0.985	49
7.516	-0.680	7.540	-0.294	7.568	-0.087	7.626	0.287	7.681	0.417	50
7.731	0.652	7.768	0.831	7.813	0.798	7.828	0.741	7.860	0.859	51
7.940	0.628	8.001	0.920	8.029	0.745	8.072	0.310	8.110	-0.043	52
8.131	-0.230	8.191	-0.259	8.260	-0.580	8.320	-0.462	8.391	-0.831	53
8.422	-0.754	8.451	-0.263	8.470	-0.019	8.488	0.095	8.532	0.152	54
8.583	0.071	8.630	0.103	8.680	0.002	8.708	-0.157	8.734	-0.299	55
8.764	-0.368	8.805	-0.234	8.843	0.039	8.869	0.246	8.908	0.429	56
8.945	0.299	8.970	0.092	8.990	-0.026	9.035	-0.059	9.088	0.124	57
9.101	0.234	9.139	0.356	9.180	0.238	9.223	0.185	9.275	0.092	58
9.320	0.015	9.352	-0.111	9.386	-0.266	9.412	-0.400	9.441	-0.494	59
9.468	-0.571	9.481	-0.526	9.520	-0.352	9.550	-0.120	9.587	0.343	60
9.613	0.575	9.657	0.717	9.707	0.562	9.744	0.278	9.765	0.107	61
9.791	0.026	9.875	0.123	9.955	-0.186	10.071	0.009	10.170	-0.162	62
10.252	-0.284	10.348	-0.418	10.468	-0.175	10.520	0.150	10.568	0.340	63
10.620	0.446	10.677	0.547	10.723	0.433	10.749	0.246	10.788	0.083	64
10.815	-0.035	10.860	-0.104	10.889	-0.051	10.925	-0.108	10.981	0.127	65
11.010	0.094	11.050	0.220	11.129	-0.044	11.209	-0.114	11.288	-0.330	66
11.386	0.153	11.448	-0.087	11.514	0.237	11.571	0.424	11.624	0.358	67
11.672	0.082	11.720	0.016	11.786	-0.074	11.836	-0.241	11.907	-0.396	68
11.990	-0.413	12.034	-0.250	12.069	-0.039	12.100	0.098	12.124	0.135	69
12.156	0.000	12.184	-0.199	12.229	-0.358	12.262	-0.435	12.311	-0.314	70
12.389	-0.143	12.434	0.100	12.478	0.234	12.512	0.323	12.583	0.225	71
12.652	0.322	12.730	0.395	12.816	0.382	12.896	0.268	12.933	0.284	72
12.991	0.141	13.037	0.052	13.096	0.104	13.154	-0.067	13.190	-0.254	73
13.286	-0.311	13.323	-0.478	13.351	-0.709	13.399	-0.937	13.440	-0.673	74
13.471	-0.319	13.493	-0.026	13.519	0.181	13.565	-0.006	13.589	-0.221	75
13.609	-0.518	13.636	-0.688	13.654	-0.469	13.678	-0.196	13.698	0.157	76
13.719	0.429	13.750	0.584	13.792	0.247	13.811	-0.139	13.839	-0.424	77
13.866	-0.257	13.898	0.109	13.924	0.418	13.963	0.556	14.000	0.463	78
14.031	0.325	14.057	0.122	14.076	0.032	14.120	0.028	14.163	0.114	79
14.196	0.240	14.217	0.301	14.268	0.147	14.296	0.037	14.359	0.009	80
14.404	0.126	14.463	-0.004	14.505	-0.101	14.552	-0.113	14.575	-0.044	81
14.631	0.049	14.678	-0.056	14.742	-0.329	14.781	-0.280	14.847	-0.321	82
14.896	-0.223	14.945	-0.052	14.987	0.025	15.075	-0.004	15.156	0.106	83
15.216	0.195	15.273	0.159	15.370	0.394	15.469	0.036	15.568	0.165	84
15.615	0.100	15.662	0.144	15.702	-0.051	15.798	-0.279	15.836	-0.146	85
15.894	-0.044	15.980	-0.216	16.065	-0.338	16.141	-0.152	16.171	0.096	86
16.234	0.177	16.264	0.119	16.336	0.302	16.420	0.205	16.470	0.095	87
16.512	0.095	16.538	0.046	16.614	-0.063	16.695	-0.274	16.770	0.043	88
16.833	-0.022	16.890	0.043	16.927	0.002	16.966	0.096	17.056	-0.148	89
17.165	-0.018	17.230	-0.185	17.316	-0.006	17.396	-0.249	17.486	0.031	90
17.606	-0.094	17.660	0.129	17.779	0.012	17.824	0.158	17.904	0.036	91
17.956	0.142	18.000	0.223	18.028	0.280	18.050	0.179	18.104	0.297	92
18.149	0.211	18.201	0.086	18.253	-0.008	18.326	-0.004	18.359	-0.004	93
18.401	-0.106	18.456	-0.245	18.541	-0.282	18.600	-0.267	18.676	-0.235	94
18.759	-0.093	18.900	-0.062	18.960	0.055	19.010	-0.039	19.077	-0.117	95
19.161	0.057	19.234	-0.020	19.298	0.016	19.368	-0.062	19.415	-0.074	96
19.505	0.112	19.634	-0.051	19.711	0.180	19.820	-0.223	19.887	0.212	97
19.936	0.325	19.978	0.167	20.032	-0.017	20.079	0.101	20.135	0.194	98
20.182	0.052	20.239	-0.111	20.343	-0.018	20.437	-0.147	20.554	0.098	99
20.664	0.041	20.724	0.099	20.839	0.034	20.935	0.067	21.019	-0.038	100
21.082	0.048	21.184	0.028	21.293	-0.105	21.422	-0.157	21.563	-0.104	101
21.663	-0.071	21.722	0.047	21.824	-0.021	21.918	0.089	22.094	0.009	102
22.211	0.074	22.297	-0.047	22.368	0.047	22.455	0.108	22.516	0.080	103
22.579	-0.001	22.637	-0.053	22.708	0.021	22.745	0.049	22.829	-0.027	104
22.927	-0.083	23.014	-0.103	23.077	-0.212	23.165	-0.151	23.241	-0.053	105
23.328	0.090	23.425	0.046	23.506	0.090	23.538	0.152	23.612	0.091	106
23.680	0.189	23.738	0.185	23.787	0.128	23.848	0.116	23.893	0.031	107
23.960	0.051	24.024	-0.022	24.081	-0.111	24.213	-0.273	24.307	-0.208	108

24.375	-0.066	24.419	0.067	24.469	0.107	24.582	0.049	24.672	0.081	109
24.767	0.104	24.846	-0.092	24.895	0.014	24.972	-0.016	25.030	-0.207	110
25.094	-0.013	25.169	0.003	25.223	0.075	25.290	0.042	25.351	0.143	111
25.409	0.090	25.480	0.114	25.521	0.151	25.595	0.098	25.682	0.065	112
25.786	0.077	25.888	0.068	25.965	-0.030	26.054	-0.083	26.227	-0.193	113
26.339	-0.271	26.457	-0.120	26.542	0.060	26.662	-0.005	26.782	0.041	114
26.917	-0.024	27.008	0.058	27.074	0.131	27.165	0.075	27.233	0.031	115
27.326	0.055	27.406	0.096	27.453	0.128	27.536	0.031	27.597	-0.063	116
27.691	-0.079	27.787	-0.172	27.914	-0.010	28.027	-0.067	28.108	0.063	117
28.206	-0.030	28.301	-0.002	28.383	-0.047	28.472	0.063	28.562	-0.035	118
28.639	0.090	28.729	-0.016	28.889	-0.041	28.989	-0.065	29.075	0.024	119
29.175	-0.050	29.249	0.031	29.296	0.088	29.387	0.055	29.498	0.030	120
29.586	0.059	29.646	-0.010	29.730	-0.059	29.781	-0.076	29.893	-0.141	121
29.962	-0.186	30.048	-0.125	30.115	-0.052	30.206	0.021	30.274	0.025	122
30.326	0.041	30.397	0.110	30.471	0.110	30.517	0.058	30.578	0.001	123
30.623	-0.040	30.670	-0.003	30.767	-0.044	30.851	0.013	30.963	0.046	124
31.045	0.082	31.138	-0.039	31.201	-0.007	31.235	0.034	31.304	-0.007	125
31.440	-0.052	31.539	-0.060	31.698	-0.032	31.741	-0.004	31.828	-0.045	126
31.937	-0.001	31.995	-0.066	32.098	0.031	32.150	0.096	32.204	0.100	127
32.275	0.063	32.335	0.027	32.413	0.055	32.487	0.002	32.554	-0.027	128
32.643	-0.027	32.717	-0.064	32.803	0.017	32.940	-0.008	33.004	0.045	129
33.074	0.024	33.135	0.053	33.221	0.081	33.312	0.134	33.403	0.028	130
33.479	0.073	33.563	0.004	33.653	-0.041	33.797	-0.057	33.925	-0.037	131
34.066	-0.028	34.194	0.049	34.329	0.041	34.472	0.036	34.562	-0.025	132
34.630	-0.009	34.677	-0.045	34.774	-0.009	34.873	0.015	34.931	0.043	133
35.027	0.014	35.119	0.043	35.176	0.083	35.251	0.038	35.343	0.026	134
35.446	-0.048	35.554	-0.024	35.642	0.004	35.739	-0.028	35.819	-0.020	135
35.889	-0.008	35.963	0.016	36.072	-0.021	36.171	-0.042	36.275	-0.054	136
36.406	-0.018	36.524	-0.014	36.602	0.058	36.702	0.046	36.772	0.086	137
36.843	0.041	36.940	0.114	37.027	0.025	37.109	-0.020	37.198	0.012	138
37.341	-0.062	37.457	-0.050	37.576	-0.070	37.698	-0.034	37.805	-0.054	139
37.872	0.027	37.964	0.006	38.046	0.030	38.155	-0.010	38.347	-0.015	140
38.450	-0.030	38.548	-0.054	38.686	0.040	38.819	-0.000	38.982	-0.040	141
39.105	-0.016	39.234	-0.043	39.337	0.022	39.424	0.018	39.567	0.006	142
39.736	0.026	39.883	-0.026	39.990	-0.010	40.133	0.015	40.235	0.047	143
40.347	-0.002	40.436	0.023	40.557	-0.030	40.674	-0.030	40.822	-0.063	144
40.966	0.006	41.097	0.026	41.188	0.014	41.290	-0.015	41.413	0.014	145
41.490	0.018	41.561	-0.006	41.642	0.022	41.753	-0.002	41.849	0.011	146
41.928	0.056	42.069	0.011	42.180	0.044	42.350	-0.017	42.495	-0.001	147
42.566	0.031	42.694	-0.030	42.775	0.011	42.879	-0.030	42.951	0.006	148
43.034	-0.063	43.206	-0.022	43.323	0.022	43.410	-0.015	43.514	0.026	149
43.636	-0.015	43.776	0.009	0.0	0.0	0.0	0.0	0.0	0.0	150

FILE 38 OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-B, EERL 70-21
PARKFIELD, CALIFORNIA EARTHQUAKE
JUN 27, 1966 - 2026 PST
IB033 66.001.0 T 18
STATION NO. 013 35 43 35N,120 17 13W 38
CHOLAME,SHANDON, CALIFORNIA ARRAY NO. 2 39
COMP N65E 9
PARKFIELD, CALIFORNIA EARTHQUAKE JUNE 27, 1966 - 2026 PST 59
EPICENTER 35 54 00N,120 54 00W 31
INSTR NOT PLACED USELESS FILE USELESS FILE 51
INSTR NOT PLACED USELESS FILE USELESS FILE 51
UNITS ARE SEC AND G/10. 23
RMS ACCLN OF COMPLETE RECORD = USELESS FILE 43

38	1	2	33	66	1	0	3	13	35	43	35-120	17	13	35	54	0-120	54	
0	6	2719662026	01000	0	32	39	0.0	0.0	0.0	0.0	0.1000							

FILE 39 OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-B, EERL 70-21
PARKFIELD, CALIFORNIA EARTHQUAKE
JUN 27, 1966 - 2026 PST
IB033 66.001.0 T 18
STATION NO. 013 35 43 35N,120 17 13W 38

```
CHOLAME,SHANDON, CALIFORNIA ARRAY NO. 2                                    39
COMP DOWN                                                                  9
PARKFIELD, CALIFORNIA EARTHQUAKE   JUNE 27, 1966 - 2026 PST                59
EPICENTER  35 54 00N,120 54 00W                                            31
INSTR PERIOD = 0.0560 SEC  DAMPING = 0.540                                 42
NO. OF POINTS = 772   DURATION = 43.77 SEC                                 42
UNITS ARE SEC AND G/10.                                                    23
RMS ACCLN OF COMPLETE RECORD = 0.2771 G/10.                                43
 39   1   2  33  66   1   0   3  13  35  43  35-120  17  13  35  54  0-120  54
  0   6  2719662026   0 600 772  32  39  0.0560  0.5400 43.7700  0.2771  0.1000
```

0.0	-0.355	0.015	0.044	0.033	-0.101	0.055	0.112	0.082	0.438	1
0.140	-0.351	0.171	0.134	0.207	-0.260	0.243	0.298	0.276	-0.306	2
0.307	0.225	0.325	0.035	0.333	0.266	0.376	-0.165	0.399	0.107	3
0.417	0.334	0.445	0.144	0.458	0.026	0.481	-0.128	0.513	0.285	4
0.536	-0.146	0.564	0.326	0.611	-0.423	0.636	0.040	0.649	0.004	5
0.682	0.480	0.718	-0.839	0.748	0.108	0.764	-0.155	0.797	0.580	6
0.823	0.095	0.836	0.172	0.877	-0.689	0.902	-0.236	0.925	0.145	7
0.938	0.671	0.964	-0.095	0.993	-0.848	1.016	-0.081	1.041	0.341	8
1.087	-0.349	1.116	0.368	1.152	-0.276	1.178	-0.530	1.198	0.037	9
1.216	0.609	1.244	0.368	1.258	-0.189	1.285	0.219	1.301	0.010	10
1.320	0.355	1.343	0.001	1.369	-0.597	1.392	0.119	1.406	0.691	11
1.445	-0.552	1.471	0.301	1.495	1.181	1.520	-0.275	1.536	-0.747	12
1.562	0.292	1.594	-1.568	1.619	0.319	1.640	1.866	1.648	1.507	13
1.659	1.639	1.684	-0.098	1.697	-1.232	1.711	-1.885	1.729	-0.660	14
1.747	-1.354	1.791	2.551	1.828	-0.882	1.865	1.952	1.892	-0.937	15
1.915	-2.071	1.933	-0.338	1.951	1.240	1.978	-0.098	2.007	-1.163	16
2.030	0.274	2.065	-1.023	2.104	1.000	2.131	-0.351	2.156	1.027	17
2.196	-2.428	2.235	2.429	2.272	-2.333	2.307	1.222	2.340	-0.284	18
2.370	1.253	2.393	0.519	2.406	0.713	2.441	-1.491	2.459	-0.117	19
2.482	0.028	2.511	1.035	2.549	-1.632	2.581	1.515	2.606	-0.685	20
2.617	-0.499	2.654	-1.733	2.673	-0.413	2.686	0.026	2.707	1.382	21
2.738	-1.276	2.762	-0.768	2.776	-1.026	2.799	1.032	2.815	3.490	22
2.836	0.379	2.867	-2.565	2.882	0.088	2.906	1.961	2.948	0.995	23
2.963	-0.257	2.977	-0.774	3.008	1.575	3.029	-0.271	3.055	-1.142	24
3.082	-0.435	3.098	-0.299	3.113	-0.009	3.132	0.218	3.163	-0.590	25
3.192	0.553	3.202	0.367	3.218	0.938	3.241	0.394	3.270	-0.885	26
3.301	0.520	3.314	1.314	3.338	0.434	3.356	-0.836	3.370	-1.530	27
3.404	0.583	3.429	-0.455	3.463	-0.973	3.490	-0.134	3.518	0.120	28
3.553	1.217	3.571	0.537	3.607	0.024	3.638	1.122	3.660	0.056	29
3.678	-0.475	3.702	-0.026	3.738	-1.056	3.762	-0.230	3.808	-1.260	30
3.843	0.517	3.877	-1.297	3.916	1.247	3.940	-0.395	3.972	-1.569	31
4.000	-0.436	4.013	0.158	4.031	0.779	4.065	0.240	4.084	0.679	32
4.112	-0.391	4.136	-1.212	4.157	-0.455	4.173	0.488	4.194	1.159	33
4.221	0.783	4.247	-0.020	4.285	0.642	4.314	0.170	4.330	-0.397	34
4.349	-0.760	4.395	0.569	4.409	-0.021	4.429	-0.805	4.448	-1.155	35
4.503	-0.298	4.539	-0.987	4.579	0.292	4.620	-0.697	4.660	0.310	36
4.672	0.110	4.710	0.890	4.738	0.504	4.777	0.958	4.827	0.622	37
4.849	1.057	4.865	0.808	4.881	1.039	4.909	0.730	4.922	1.089	38
4.968	0.073	4.983	0.399	5.021	-0.054	5.041	0.218	5.079	-0.118	39
5.100	-0.422	5.113	-0.227	5.152	-0.581	5.171	-0.345	5.192	-0.613	40
5.227	-1.257	5.261	-0.840	5.290	-0.486	5.319	-0.051	5.333	-0.242	41
5.354	0.003	5.380	-0.187	5.427	0.184	5.470	0.057	5.496	0.266	42
5.524	0.111	5.565	0.438	5.595	0.161	5.614	-0.080	5.626	0.038	43
5.664	-0.470	5.699	-0.225	5.736	-0.443	5.764	-0.602	5.780	-0.420	44
5.807	-0.071	5.824	-0.271	5.858	-0.004	5.884	-0.267	5.909	0.105	45
5.930	0.395	5.987	0.200	6.008	0.445	6.031	0.581	6.054	0.757	46
6.096	0.453	6.126	0.676	6.147	0.875	6.170	1.093	6.204	0.888	47
6.230	0.616	6.249	0.829	6.287	0.543	6.313	0.158	6.351	-0.296	48
6.375	-0.092	6.413	-0.392	6.439	-0.678	6.480	-0.392	6.526	-0.810	49
6.575	-0.366	6.601	-0.584	6.638	-0.240	6.658	-0.349	6.676	-0.403	50
6.722	-0.159	6.754	-0.467	6.797	-0.127	6.837	-0.354	6.881	0.167	51
6.903	-0.010	6.943	0.388	6.984	0.071	7.031	0.474	7.084	0.070	52

7.103	0.179	7.126	0.364	7.160	0.128	7.189	0.386	7.227	0.055	53
7.250	0.232	7.284	0.454	7.322	0.145	7.368	0.213	7.411	0.435	54
7.434	0.153	7.456	-0.078	7.503	-0.128	7.531	0.012	7.552	0.171	55
7.594	-0.170	7.624	-0.011	7.650	-0.243	7.695	-0.402	7.720	-0.506	56
7.752	-0.407	7.791	-0.507	7.826	-0.244	7.849	-0.353	7.889	-0.598	57
7.915	-0.281	7.936	0.046	7.959	-0.145	7.982	-0.313	8.004	-0.141	58
8.025	0.081	8.051	0.244	8.076	0.090	8.103	0.240	8.136	0.013	59
8.155	0.253	8.174	0.470	8.200	0.565	8.244	0.343	8.273	0.433	60
8.315	0.261	8.348	0.501	8.401	0.043	8.446	0.275	8.486	-0.206	61
8.532	0.084	8.569	-0.211	8.588	-0.034	8.621	-0.215	8.654	0.066	62
8.703	-0.197	8.743	0.234	8.797	-0.174	8.827	0.002	8.855	0.089	63
8.885	0.238	8.949	-0.075	8.980	0.052	9.028	-0.070	9.065	-0.220	64
9.102	-0.115	9.143	-0.193	9.183	-0.515	9.223	-0.129	9.258	-0.401	65
9.299	-0.093	9.337	0.084	9.370	0.202	9.405	0.025	9.426	0.206	66
9.450	0.392	9.479	0.211	9.518	0.370	9.555	0.120	9.578	0.275	67
9.623	0.002	9.637	0.184	9.678	-0.097	9.712	0.129	9.773	-0.270	68
9.804	-0.029	9.833	0.184	9.870	-0.057	9.906	0.148	9.926	0.016	69
9.962	0.225	10.015	0.093	10.061	0.138	10.077	0.234	10.107	0.075	70
10.159	0.329	10.222	0.102	10.243	0.193	10.300	-0.061	10.328	0.011	71
10.361	-0.297	10.400	-0.107	10.441	-0.271	10.467	-0.076	10.496	-0.257	72
10.522	-0.053	10.558	-0.344	10.589	-0.140	10.625	-0.195	10.636	-0.158	73
10.667	-0.431	10.711	0.041	10.734	-0.163	10.776	0.140	10.804	-0.055	74
10.831	0.326	10.868	-0.114	10.904	0.511	10.947	-0.056	10.973	0.366	75
11.016	-0.365	11.050	0.320	11.119	-0.052	11.177	0.206	11.201	0.043	76
11.222	0.210	11.262	-0.107	11.298	0.119	11.343	0.114	11.389	0.313	77
11.423	0.114	11.464	0.313	11.511	0.049	11.541	-0.345	11.581	0.044	78
11.640	-0.396	11.677	-0.115	11.707	-0.261	11.736	-0.093	11.759	-0.261	79
11.781	-0.084	11.803	-0.003	11.831	0.160	11.874	-0.085	11.924	0.150	80
11.952	-0.009	11.987	0.091	12.010	-0.009	12.048	0.181	12.087	-0.032	81
12.105	0.094	12.127	-0.024	12.159	0.158	12.187	0.144	12.254	-0.179	82
12.303	0.093	12.338	-0.111	12.372	0.134	12.406	-0.057	12.437	-0.139	83
12.461	-0.207	12.516	0.097	12.555	-0.166	12.604	-0.039	12.636	-0.012	84
12.662	0.133	12.725	-0.071	12.778	0.251	12.822	0.074	12.850	0.137	85
12.891	0.046	12.913	0.228	12.938	0.078	12.957	0.196	12.995	0.019	86
13.026	0.169	13.081	-0.040	13.125	0.069	13.178	-0.176	13.221	0.050	87
13.279	-0.177	13.321	0.014	13.386	-0.104	13.435	0.046	13.472	-0.085	88
13.514	0.210	13.565	-0.112	13.605	0.110	13.646	-0.039	13.688	0.151	89
13.755	0.297	13.798	0.138	13.842	0.011	13.892	0.088	13.931	-0.029	90
13.981	0.052	13.999	0.102	14.049	-0.043	14.117	0.035	14.168	-0.124	91
14.196	-0.042	14.234	-0.246	14.264	-0.101	14.304	-0.228	14.341	-0.042	92
14.379	-0.001	14.409	-0.160	14.445	-0.001	14.486	0.194	14.534	-0.078	93
14.562	0.154	14.601	-0.064	14.646	0.090	14.688	0.140	14.725	0.022	94
14.764	0.176	14.797	-0.028	14.847	0.245	14.893	0.050	14.917	0.213	95
14.974	0.009	15.003	0.181	15.057	-0.064	15.089	0.073	15.138	0.100	96
15.172	0.141	15.213	-0.104	15.265	0.073	15.297	0.023	15.334	-0.140	97
15.362	0.000	15.407	-0.150	15.471	-0.150	15.514	0.004	15.571	-0.086	98
15.634	0.081	15.673	0.059	15.697	0.127	15.757	-0.000	15.812	0.140	99
15.893	0.063	15.957	-0.041	16.034	0.199	16.073	0.031	16.112	0.262	100
16.161	0.026	16.190	0.199	16.239	0.026	16.316	-0.028	16.371	-0.101	101
16.407	-0.051	16.460	-0.105	16.517	0.080	16.591	-0.042	16.627	0.153	102
16.659	0.026	16.708	0.171	16.763	0.243	16.792	0.062	16.836	0.252	103
16.900	0.153	16.953	0.062	17.030	-0.011	17.054	-0.047	17.114	0.102	104
17.170	-0.156	17.202	0.052	17.238	-0.156	17.274	0.089	17.324	-0.220	105
17.382	-0.071	17.435	-0.162	17.493	-0.071	17.567	0.024	17.601	-0.095	106
17.670	0.105	17.710	0.036	17.766	0.226	17.795	0.054	17.859	0.253	107
17.928	0.008	17.973	0.157	18.029	-0.043	18.063	0.098	18.175	-0.143	108
18.223	0.038	18.261	-0.171	18.306	0.051	18.338	-0.117	18.367	0.010	109
18.404	0.123	18.481	0.055	18.531	0.127	18.565	0.181	18.613	0.063	110
18.639	0.158	18.669	0.036	18.700	0.167	18.754	0.031	18.803	-0.060	111
18.855	0.053	18.916	-0.083	18.955	0.030	19.075	-0.156	19.111	0.052	112
19.143	-0.089	19.208	0.047	19.268	0.097	19.306	-0.058	19.360	0.124	113
19.420	0.006	19.456	0.088	19.575	0.020	19.623	0.020	19.655	0.097	114

19.708	-0.048	19.763	0.106	19.824	-0.053	19.895	0.075	19.969	-0.012	115
20.065	-0.089	20.149	-0.025	20.203	-0.111	20.262	0.002	20.343	-0.161	116
20.422	0.061	20.512	0.111	20.630	0.025	20.765	-0.025	20.952	0.083	117
21.053	0.079	21.224	0.033	21.300	0.115	21.353	0.014	21.408	0.069	118
21.464	0.027	21.508	0.095	21.587	-0.059	21.651	0.040	21.755	-0.065	119
21.863	0.039	22.020	-0.012	22.106	0.001	22.154	-0.054	22.200	0.037	120
22.310	-0.072	22.439	0.091	22.484	0.009	22.542	0.145	22.626	-0.019	121
22.682	0.135	22.726	-0.033	22.778	0.131	22.847	-0.065	22.894	0.067	122
22.969	-0.065	23.112	0.016	23.182	-0.074	23.307	0.025	23.402	-0.142	123
23.504	-0.024	23.565	-0.046	23.696	0.163	23.770	0.041	23.845	0.123	124
23.942	0.010	24.087	0.074	24.171	0.124	24.286	0.030	24.380	0.035	125
24.431	-0.101	24.575	-0.082	24.685	-0.086	24.826	0.018	24.958	0.024	126
25.057	-0.039	25.148	0.088	25.300	0.043	25.463	-0.025	25.626	0.048	127
25.799	0.093	25.965	0.143	26.097	0.039	26.190	-0.052	26.330	-0.093	128
26.466	-0.098	26.563	-0.098	26.679	-0.003	26.821	0.032	26.916	0.132	129
27.052	0.095	27.159	0.067	27.242	0.063	27.298	0.149	27.430	0.053	130
27.555	-0.024	27.744	-0.078	27.864	-0.110	28.025	-0.087	28.166	-0.005	131
28.324	0.135	28.475	0.041	28.625	-0.017	28.810	-0.089	28.902	-0.029	132
29.043	-0.010	29.133	-0.046	29.257	0.005	29.440	0.045	29.579	0.090	133
29.706	0.049	29.867	-0.019	30.019	-0.060	30.196	-0.025	30.310	-0.012	134
30.397	0.015	30.538	0.008	30.656	0.002	30.745	-0.012	30.859	-0.041	135
30.970	-0.079	31.047	-0.048	31.156	-0.022	31.270	0.026	31.360	0.026	136
31.456	-0.025	31.578	-0.048	31.677	-0.054	31.804	-0.032	31.899	0.004	137
31.995	-0.015	32.110	-0.007	32.218	-0.007	32.324	-0.008	32.400	-0.003	138
32.515	-0.049	32.630	-0.076	32.865	-0.017	33.019	-0.017	33.138	-0.021	139
33.221	-0.035	33.346	0.028	33.493	-0.049	33.629	-0.086	33.681	-0.036	140
33.740	-0.100	33.808	-0.036	33.916	-0.118	34.053	-0.019	34.196	-0.010	141
34.320	-0.001	34.475	-0.015	34.662	-0.024	34.842	-0.024	34.975	0.004	142
35.095	-0.028	35.147	-0.010	35.246	-0.087	35.358	-0.069	35.551	-0.055	143
35.664	-0.064	35.775	-0.087	35.963	-0.068	36.148	-0.050	36.244	-0.004	144
36.350	0.009	36.485	-0.023	36.595	-0.019	36.741	0.003	36.891	-0.065	145
37.038	-0.043	37.142	-0.048	37.209	-0.085	37.328	-0.089	37.430	-0.030	146
37.585	-0.057	37.743	-0.007	37.957	-0.034	38.092	-0.030	38.323	-0.029	147
38.541	-0.014	38.730	-0.031	38.935	-0.048	39.156	-0.014	39.327	-0.031	148
39.484	0.001	39.602	0.002	39.706	-0.002	39.885	-0.042	40.059	-0.063	149
40.220	0.005	40.381	0.001	40.584	0.019	40.748	-0.026	40.947	-0.045	150
41.137	-0.068	41.305	-0.041	41.472	-0.018	41.666	0.004	41.732	-0.028	151
41.833	-0.005	41.944	-0.023	42.077	-0.078	42.189	-0.055	42.295	-0.033	152
42.437	-0.040	42.533	-0.053	42.664	-0.038	42.785	-0.019	42.877	-0.045	153
42.968	-0.003	43.041	-0.016	43.144	-0.002	43.310	-0.004	43.437	-0.026	154
43.596	-0.007	43.776	-0.042	0.0	0.0	0.0	0.0	0.0	0.0	155

DATA PROCESSING FOR VOLUME II: CORRECTED ACCELERATION, VELOCITY AND DISPLACEMENT CURVES AND VOLUME II TAPE

The scaled, uncorrected accelerogram data (Figure 1), obtained from the first stage of processing and presented in the various parts of Volume I, are retrieved from the Volume I magnetic tape storage and corrected for instrument frequency response and base-line adjustment as outlined in Volume II, Part A, Report No. EERL 71-50 (Hudson, et al, 1971). This section presents the Fortran IV programs that perform those operations.

The processing of corrected accelerograms (Vol. II MAIN program) involves first instrument correction and then base-line correction. From the uncorrected accelerograms, digitized at unequally spaced points, equally spaced data with 50 points per second are interpolated. To improve accuracy this interpolation is done as follows. First, 100 equally spaced points per second are interpolated from the uncorrected data, assuming a straight line between points. The equally spaced data are then low-pass filtered using an Ormsby filter having a cut-off frequency $f_c = 25$ cps and a roll-off termination frequency $f_T = 27$ cps. Subsequent to low-pass filtering the data are decimated by keeping every second point. The resulting sequence has 50 points per second which correspond to a Nyquist frequency of 25 cps. Instrument correction is performed by using the instrument constants ω_o and ζ_o determined from calibration tests of each accelerograph component. The above operations are performed by the ICR subroutine.

Accelerograms corrected for instrument response are next base-line corrected by high-pass filtering with an Ormsby filter having a cut-off frequency of 0.07 cps and a roll-off termination frequency of 0.05 cps. The output of this operation, carried out by the BAS subroutine, is the corrected accelerogram which accurately represents the absolute acceleration of the instrument base in the frequency band between 0.07 cps and 25 cps.

To avoid long period errors, for periods longer than 16 seconds, resulting from the uncertainties involved in estimating the the initial velocity and displacement of ground motion (Hudson, et al, 1971), the computed velocity and displacement curves are high-pass filtered using the same Ormsby filter as in the BAS subroutine. This is done by the HYPSVD Subroutine.

The corrected acceleration, velocity and displacement for each component of an earthquake record are plotted to appropriate scales by the subroutine TRILOT; however, only the accelerogram is printed out. Details concerning identification and peak values of acceleration, velocity, and displacement are given at the top of each plot (Figure 4). The first line gives the name, date, and time of occurrence of the earthquake. The second line is comprised of two labels, the observation station and component processed. The Roman numeral "II" in the first identification label indicates that the results pertain to the second stage of data processing, i.e., Volume II of corrected accelerogram records. The letter "A" following the Roman numerals implies that the processed record belongs to Part A of Volume II. Volume II, Part A consists of the corrected acceleration data for

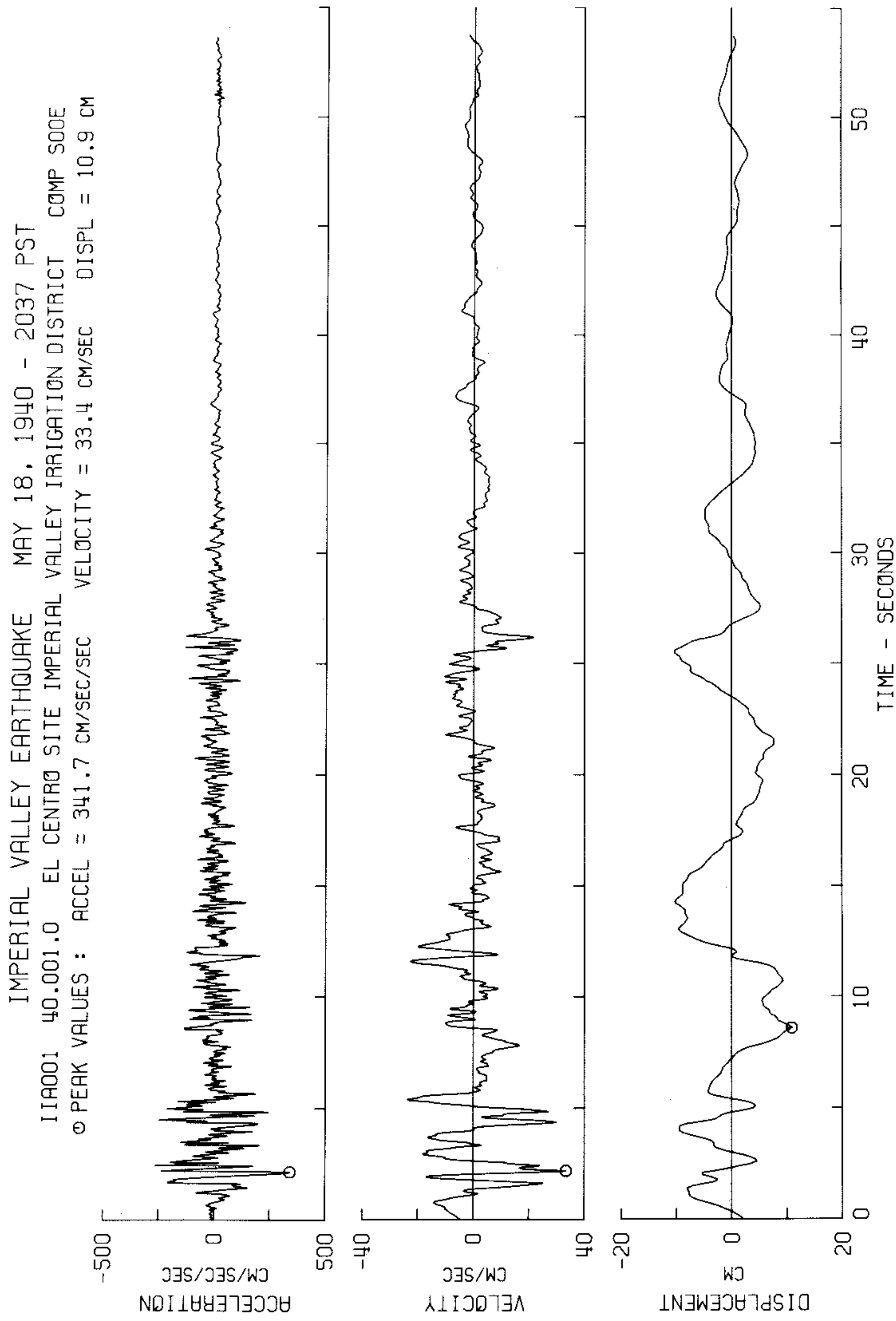

IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST
IIA001 40.001.0   EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT   COMP SOOE
PEAK VALUES : ACCEL = 341.7 CM/SEC/SEC   VELOCITY = 33.4 CM/SEC   DISPL = 10.9 CM
ACCELERATION
CM/SEC/SEC
-500
0
500
VELOCITY
CM/SEC
-40
0
40
DISPLACEMENT
CM
-20
0
20
0
10
20
30
40
50
TIME - SECONDS

Figure 4

the same records for which the uncorrected data have been published in Volume I, Part A (Hudson, et al, 1969). This is valid for all succeeding Parts. The three-digit number completing the first label is the Caltech Reference Number for the given earthquake record in Volume I, right-adjusted in a three-digit numerical field. The record label (for instance 57.002.2) is a string of three numbers separated by periods. The first number gives the year in which the earthquake occurred. The second is the serial number of the record as it was received at the Caltech Earthquake Engineering Research Laboratory during that year. The last number indicates whether it was a main event or an aftershock, sequentially numbered, the main event starting from zero.

It will be observed that, contrary to general practice, the y-axis in Figure 4 is marked positive downwards. Since the accelerograph registers the negative of the true acceleration, it is necessary to either invert the recorded accelerogram about the x-axis or change the positive direction of the y-axis. Since investigators are often familiar with typical accelerograms in the form as registered by the instrument, it is preferred to label the y-axis suitably and retain the familiar pattern of the accelerograms.

In the Volume II reports the corrected accelerogram is printed out at equal time intervals of 0.02 seconds. Figure 5 shows the identification labels and the beginning of the printout for the component plotted in Figure 4. For each component in a record, the identification labels are given in the first line together with the

IIA001 40.001.0 IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST EPICENTER 32 44 00N,115 27 00W
STATION NO. 117 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E 32 47 43N,115 32 55W
INSTR PERIOD = 0.0990 SEC DAMPING = 0.552 ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN 0.070 AND 25 CYC/SEC.

PEAK VALS ACLN = 341.7 CM/SEC/SEC AT 2.12 SEC VELO = 33.4 CM/SEC AT 2.18 SEC DISP = 10.9 CM AT 8.58 SEC

INITIAL VELO = -4.66421 CM/SEC INITIAL DISP = 2.15852 CM

2688 INSTRUMENT AND BASELINE CORRECTED DATA IN MM/SEC/SEC AT EQUALLY-SPACED INTERVALS OF 0.02 SEC

-14	-108	-101	-88	-95	-120	-142	-128	-110	-85	-85	-131	-176	-194	-162	-144	-108	-82	-42	-66
-131	-190	-196	-66	30	141	-49	-128	-144	-203	-260	-325	-306	-172	-197	-163	-164	-67	25	150
236	252	336	463	492	419	359	271	235	339	412	530	639	732	652	599	400	400	63	-515
-787	-603	-484	-250	-59	134	308	499	710	995	1219	1529	1449	1155	935	892	926	839	901	993
1209	328	-1475	-2066	-1989	-2034	-1816	-1725	-1752	-1753	-1805	-1630	-1347	-1087	-782	-429	-17	360	785	1164
1598	1960	2412	2729	3036	3200	3417	2821	2324	-1198	-2373	-1640	-1865	-1095	-753	-173	113	533	895	1186
1757	576	-2631	-1547	-1729	-1012	-579	237	-670	-1980	-1641	-1685	-1481	-1231	-1001	-751	-523	-271	-44	188
-95	-433	-838	-951	-716	-599	-334	-108	185	420	673	-97	-372	-40	11	344	565	883	1130	1363
219	241	683	689	1318	1353	2040	-931	-1308	-692	-546	72	675	-1067	-1488	-1071	-1162	-762	-559	-215
-126	-674	-324	-337	-109	17	299	488	608	222	-32	-245	77	211	568	826	1206	1478	1737	421
29	259	293	-55	-147	143	206	499	645	957	1128	1447	1629	1945	1856	1984	1769	1250	-1207	-542
-384	-311	-1118	-1661	-2464	-2025	-1835	-1317	-960	-325	154	816	1319	1818	-58	-169	285	447	983	1424
1853	2456	1685	-1380	-999	-1089	-907	-469	-1250	-2111	-1617	-1692	-1306	-1111	-773	-510	-544	-1200	-1209	-1158
-1145	-717	-546	64	-804	-1634	-859	-961	-396	-147	319	648	876	472	198	-27	292	445	785	1033
1352	1606	1861	1281	640	204	314	373	496	235	-84	-168	-113	-229	-248	-157	-69	147	379	579
255	-41	-428	-133	95	230	-129	-50	80	210	380	510	157	-32	-111	5	76	35	-95	-36
-16	38	85	-56	-304	-421	-244	-236	-177	-129	-18	203	-108	-91	-34	-106	-111	-99	-2	73
235	355	705	779	184	-263	-124	-42	159	48	-219	-467	-428	-216	-43	159	320	419	123	-160
-204	-82	-206	-137	-55	53	134	266	232	79	-8	200	435	492	191	92	-22	-21	52	93
255	368	525	541	425	398	559	756	365	411	98	-204	-249	-405	-413	-471	-433	-458	-57	178
-208	-492	-530	-362	-405	-308	-316	-265	-265	-269	-345	-309	-217	-78	87	281	310	358	341	358
287	305	112	214	136	384	-861	-1349	-1342	-1354	-1193	-1042	-829	-651	-444	-258	-60	-91	-182	-147
85	163	50	264	582	867	1200	1695	1111	-1100	-366	-445	-236	-960	-656	-597	-670	-552	-27	378
1072	1669	947	408	667	132	-95	-520	-827	-1152	-1150	-803	-369	29	545	1178	1610	-270	34	-56
20	146	537	798	-205	-590	-169	-175	-28	74	382	567	753	801	592	304	23	64	-406	-451
-79	168	567	93	-55	44	-123	-282	-437	-352	-255	-111	205	519	854	1144	733	237	-368	-271
-217	-873	-973	-589	-336	77	259	508	361	81	-56	-209	-317	-238	-376	-550	-722	-803	-523	-340
-11	65	-37	-5	-168	-410	-80	79	374	615	665	254	-57	-474	-356	-243	-48	126	379	241
-227	-428	-679	-661	-590	-513	-408	-309	-266	-541	-628	-908	-1107	-881	-770	-582	-473	-333	-199	20
211	432	613	767	933	1066	1130	1187	1247	1334	1594	1797	2037	1236	442	-140	-666	-555	-693	-984
-1246	-1179	-1050	-920	-743	-809	-850	-860	-863	-873	-868	-885	-537	52	215	245	580	314	236	485
589	525	355	197	199	492	343	288	432	239	88	77	-148	-77	-19	75	44	-145	-316	-241
-28	182	426	439	512	466	479	193	222	274	393	504	577	588	822	797	949	345	45	-123
-347	-426	-416	-275	-270	74	428	-231	-387	-83	139	445	27	-697	-796	-251	-135	79	-115	-251
-333	-269	-301	-200	-67	-38	105	296	344	957	898	179	-362	-994	-807	-744	-539	-330	-128	31
148	508	-22	-489	-358	-691	-516	-371	88	632	841	1276	1388	1193	751	225	-88	-227	74	181
544	399	45	-82	-185	-20	6	-117	-210	-303	-512	-727	-579	-266	-178	40	98	137	221	437
91	-548	-555	-243	-81	250	410	182	-27	-243	-15	247	482	783	622	331	-14	-195	-247	-212
-110	50	241	-34	-216	-471	-363	-195	-18	170	-80	5	230	374	601	516	432	344	505	653
683	172	-170	-527	-664	-387	-222	-33	119	-128	-351	-514	-335	-218	-12	142	70	-63	-120	-322

Figure 5

latitude and longitude of the earthquake. In the second line the station number* of the observation station, as well as its latitude and longitude, are given for the convenience of those who would like to work on seismic waves. The accelerograph characteristics used for instrument correction are also given in each case.

The data are printed out in integer form with units of mm/sec^2, so as to utilize the printing space more efficiently. There are 20 successive values of acceleration at 0.02 sec intervals printed along each row with blank spaces after every set of 5 entries for each reading. Similarly at the end of 10 lines, one row is skipped before printing the succeeding values. This printing is performed by IIXWRT program which reads the Volume II tape.

Since the initial values of velocity and displacement are given, with the corresponding acceleration data, it is possible to compute the velocity and displacement curves if numerical values are desired. However, it must be remembered that the integrated curves so obtained might need further filtering to eliminate minor long period errors resulting from small errors in the initial values. This question is discussed in greater detail in Volume II, Part A, Report No. EERL 71-50 (Hudson, et al, 1971).

The corrected acceleration, velocity and displacement data are stored on punched cards and on magnetic tapes (Volume II tape),

---

* Permanent Identification Number in Annual List of Stations issued by Seismic Engineering Branch, USGS (formerly Seismological Field Survey, NOAA).

copies of which can be supplied at cost from the National Information Service for Earthquake Engineering at the California Institute of Technology. The set-up of one file of the Volume II tape is as follows:

<u>Volume II Tape</u>
(one file per one acceleration component)

Each file has:

1. Heading data of alphanumeric type
2. Heading data of integer type
3. Heading data of floating point type
4. Corrected acceleration points equally spaced at 50 points/second
5. Velocity points equally spaced at 50 points/second
6. Displacement points equally spaced at 50 points/second
7. EOF

Tape parameters: 1600 bpi, LRECL=1204, BLKSIZE=3616, RECFM=VBS. The detailed description and a sample of the heading data set are given in the following section.

VOLUME II HEADING DATA

Punched Output Card No.	Heading Data Array (CORTIL(I), I=I1, I2) I1 - I2	Description
1	1 - 20	Volume II MAIN title
2-14	21 - 280	Same as (CARMOD(I), I=1, 260) in the cards no. 1-13 of Vol. I heading data
15	281 - 300	Frequency limits for the band-pass filtering
16	301 - 320	No. of corrected acceleration data
17	321 - 340	Time spacing of interpolated data
18	341 - 360	Value & time of peak acceleration
19	361 - 380	Value & time of peak velocity
20	381 - 400	Value & time of peak displacement
21	401 - 420	Initial velocity & displacement
22, 23	421 - 460	Earthquake title; same as card no. 9
24, 25	461 - 500	Volume II earthquake title

Punched Output Card No.	ICOR(I) I	Description
26	1 - 20	Same as IR(1)-IR(20) of card no. 14 in Volume I heading data
27	21 - 30	Same as IR(21)-IR(30) of card no. 15 in Volume I heading data
	31 - 40	Blank
28	41 - 50	Blank
	51	NDATAB, Same as NDATA, ICOR(28)
	52	NDATA2*, Same as NDATA, ICOR(28)
	53	NDATAA, No. of corrected data of acceleration
	54	NSKIP1, No. of points to be skipped in the SMU decimation process
	55	NSKIP2, No. of points to be skipped in the BAS decimation process
	56	IPRO, No. of points in the digital filter in BAS filtering process
	57	NITR, No. of iterations used in the BAS filtering process
	58	IEXP, Defined in the BAS comment cards
	59	NLINE1, No. of letters in the earthquake title (used by the subroutine TRILOT)
	60	NLINE2, No. of letters in the second earthquake title (used by the subroutine TRILOT)

* See end of table.

29	61	NSKVEL, No. of velocity data to be skipped in the HYPSVD decimation process
	62	NSKDIS, No. of displacement data to be skipped in the HYPSVD decimation process
	63	NSKV, No. of velocity data to be skipped in TRILOT plotting process
	64	NDATV, No. of velocity data
	65	NSKD, No. of displacement data to be skipped in the TRILOT plotting process
	66	NDATD, No. of displacement data
	67 - 80	Blank
30	81 - 100	Blank

	FCOR(I) I	
31	1 - 5	Same as FR(1)-FR(5) in card no. 15 of Volume I heading data
	6 - 10	0.0
32 - 35	11 - 50	0.0
36	51	SCALE1, Scaling factor for the time coordinates
	52	SCALE2**, Scaling factor for converting input from G/10 to cm/sec^2
	53	DDT1, Time interval for equally spaced input data for ICR
	54	TMX1, Duration of uncorrected data
	55	WNO, Natural frequency of the instrument
	56	SCAL1, Scaling factors for the time coordinates (see BAS for details)
	57	SCAL2, Scaling factors for the acceleration coordinates (see BAS for details)
	58	FN2, Roll-off termination frequency for Ormsby low-pass filter in cps (used in ICR)
	59	DF2, Roll-off width for Ormsby low-pass filter in cps (used in ICR)
	60	TMX2, Maximum value of the time coordinate for plotting
37	61	DDT2, Time spacing of interpolated acceleration data
	62	FN1, Roll-off termination frequency for Ormsby low-pass filter in cps (used in BAS)
	63	DF1, Roll-off width for Ormsby low-pass filter (used in BAS)
	64	TBEG, Initial time, usually 0.0 (see BAS comment for special cases)

** See end of table.

	65	TVAL(1), Time of peak acceleration in sec	
	66	PVAL(1), Value of peak acceleration in cm/sec^2	
	67	TVAL(2), Time of peak velocity in sec	
	68	PVAL(2), Value of peak velocity in cm/sec	
	69	TVAL(3), Time of peak displacement in sec	
	70	PVAL(3), Value of peak displacement in cm	
38	71	VO, Initial velocity	
	72	BAND1, Ormsby filter (high-pass) roll-off frequency in cps (used in BAS)	
	73	BAND2, Ormsby filter (low-pass) roll-off frequency in cps (used in ICR)	
	74	DTV, Time spacing of velocity data used in calculating weights of Ormsby filter in HYPSVD	
	75	DDIS, Time spacing of displacement data used in calculating weights of Ormsby filter in HYPSVD	
	76	X0IN, Initial displacement	
	77	TLVAL(2), Time of peak velocity in sec in TRILOT plotting process	
	78	PLVAL(2), Value of peak velocity in cm/sec in TRILOT plotting process	
	79	TLVAL(3), Time of peak displacement in sec in TRILOT plotting process	***
	80	PLVAL(3), Value of peak displacement in cm in TRILOT plotting process	
39,40	81 - 100	0.0	

---

*ICOR(52), NDATA2; In older versions of the program, NDATA2=NDATA+2 because it was felt desirable to add to the input data 2 zero points at the end, so that the last point of the new equally spaced sequence has time coordinates no less than the total duration of the record. The versions used now have NDATA2=NDATA.

**FCOR(52), SCALE2; In older versions of the program, SCALE2 was equal to 1/98.0655 instead of 98.0655 and card no. 186 of the Volume II MAIN program read X(I)=X(I)/SCALE2, instead of X(I)=X(I)*SCALE2 as it now does.

***Due to the decimation of velocity and displacement data the coordinates of the plotted peak velocity and displacement points do not agree exactly with the peak values calculated before the decimation. While for the early parts of Volume II data this fine point was ignored by putting zeroes in those slots, the latest version provides the four numbers.

```
CORRECTED ACCELEROGRAM  IIA001  40.001.0    COMP SOOE   FILE  1 CORRESPONDING TO
FILE   1  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-A, EERL 70-20
IMPERIAL VALLEY EARTHQUAKE
MAY 18, 1940 - 2037 PST
IA001    40.001.0 S                                                        18
STATION NO. 117    32 47 43N,115 32 55W                                    38
EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT                         50
COMP SOOE                                                                  9
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST                       52
EPICENTER  32 44 OON,115 27 OOW                                            31
INSTR PERIOD = 0.0990 SEC  DAMPING = 0.552                                 42
NO. OF POINTS = 985   DURATION = 53.73 SEC                                 42
UNITS ARE SEC AND G/10.                                                    23
RMS ACCLN OF COMPLETE RECORD = 0.4876 G/10.                                43
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  2688  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF    0.02    SEC.
PEAK ACCELERATION =    341.69531  CMS/SEC/SEC AT    2.1200 SEC
PEAK VELOCITY     =     33.44914  CMS/SEC      AT    2.1800 SEC
PEAK DISPLACEMENT =     10.86678  CMS          AT    8.5800 SEC
INITIAL VELOCITY  =     -4.66421  CMS/SEC         INITIAL DISP. =      2.15852 CMS
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST

IIA001  40.001.0   EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT   COMP SOO
E
   1   1    1   1  40   1   0   4 117   32   47   43-115   32   55   32   44    0-115   27
   0   5   1819402037   0 180 985  26   50    0    0    0    0    0    0    0    0    0    0
   0   0    0   0   0   0   0   0   0    0 985 9872688    2   10   10    1    0   52   81
  10  10   21344    5 538   0   0   0    0    0    0    0    0    0    0    0    0    0
   0   0    0   0   0   0   0   0   0    0    0    0    0    0    0    0    0    0    0    0
   0.099    0.552   53.730    0.488    0.100    0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   1.000    0.010    0.010   53.732   63.466    1.000    1.000   27.000    2.000   53.740
   0.020    0.070    0.020    0.0      2.120  341.695    2.180   33.449    8.580   10.867
  -4.664    0.070   25.000    0.200    0.200    2.159    0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
```

PROGRAMS FOR PROCESSING VOLUME II DATA

Volume II MAIN Program (Trifunac, Vijayaraghavan)

Volume I accelerograms, which are read from Volume I tape, are corrected using this program for instrument effects, high-frequency digitization errors, base-line effects, and low frequency digitization errors. Velocity and displacement are calculated from the corrected, equally-spaced acceleration. The acceleration, velocity, and displacement curves are next plotted and written on tape with explanatory titles.

Usage

The program reads two cards containing the following data:

LFILE, MFILE, NREAD, NWRITE

and

BAND1, BAND2, DF1, DF2, ISHORT, INCARD, TLIMIT

where

LFILE = first file number of the Vol. I tape to be read in

MFILE = last file number of the Vol. I tape to be read in

NREAD = unit number for the Vol. I tape to be read in

NWRITE = unit number for the Vol. II tape to be written on

BAND1 = Ormsby filter (high-pass) roll-off frequency in cps (used in BAS)

BAND2 = Ormsby filter (low-pass) roll-off frequency in cps (used in ICR)

DF1 = Roll-off width for Ormsby low-pass filter (used in BAS)

DF2 = Roll-off width for Ormsby low-pass filter (used in ICR)

ISHORT = 1, bypasses the ICR subroutine

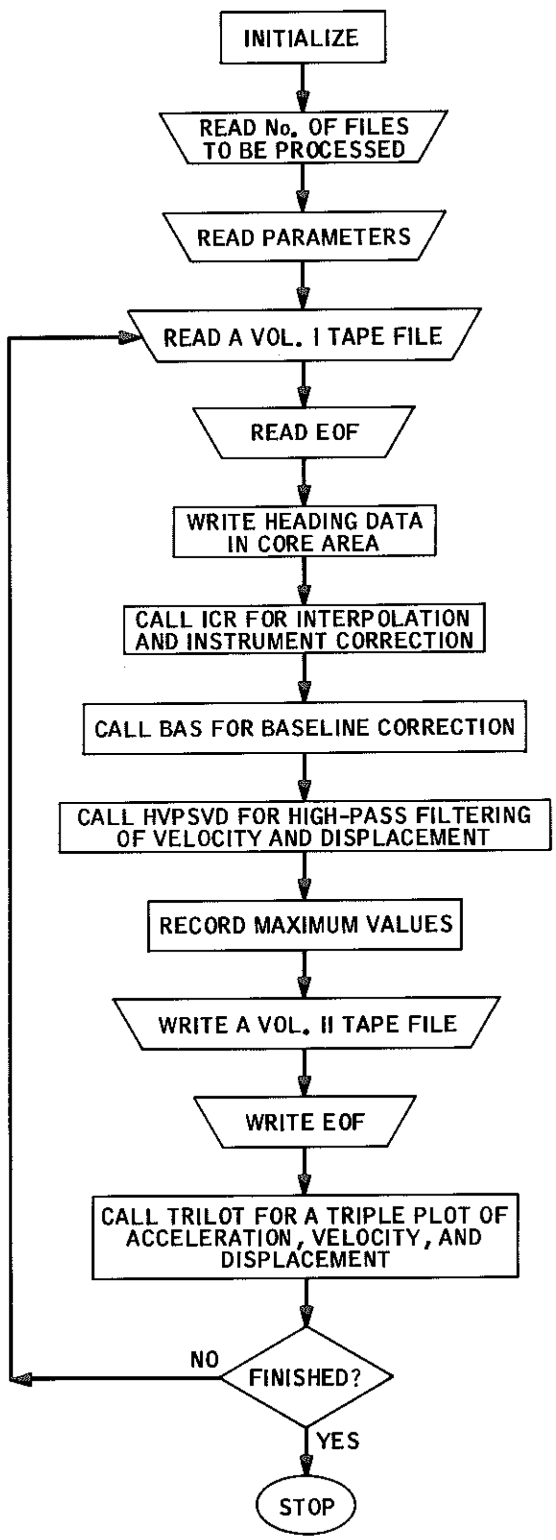
0, in normal usage

INCARD > 0, if Volume I acceleration data is being read from cards, while the heading data is read from tape

= 0, otherwise

TLIMIT = 0, if |last time coordinate read in - the total duration of the record read in from the heading data| ≥ TLIMIT, program stops and gives the message (MAIN 399). In normal usage TLIMIT = 2×10^{-5}.

VOL. II MAIN FLOW CHART



```
      INTEGER ICOR(100),IR(100)
      REAL FCOR(100),FR(50)
      REAL*8 TIME
      DIMENSION VVV(2),VOL(3),CHAR(10)
      INTEGER   ALINE1(33),ALINE2(33),SITE(72),TITLE(72)
      DIMENSION FOM1(3),FOM2(10),FMT131(7),FMT132(12)
      DIMENSION PLVAL (3),TLVAL(3)
      COMMON DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
     *     TX(10000),X(10000),T(10000)
      COMMON /B1/ PVAL(3),TVAL(3),PP(3),XD(5000),XDD(5000)
      INTEGER CORTIL(500),CARMOD(300),VOLREF,IVOL
      EQUIVALENCE (NWR,NWRITE)
      EQUIVALENCE (CORTIL(421),ALINE1(1)),(CORTIL(461),ALINE2(1))
      EQUIVALENCE (SITE(1),TITLE(1))
      EQUIVALENCE (ICOR(1),IR(1)),(FCOR(1),FR(1))
      EQUIVALENCE (VVV(1),VOLREF),(VVV(2),VOL(1))
      EQUIVALENCE (FCOR(51),SCALE1),(FCOR(52),SCALE2),(FCOR(53),DDT1),
     2 (FCOR(54),TMX1),(FCOR(55),WNO),(FCOR(56),SCAL1),(FCOR(57),SCAL2)
     3 ,(FCOR(58),FN2),(FCOR(59),DF2),(FCOR(60),TMX2),(FCOR(61),DDT2),
     4 (FCOR(62),FN1),(FCOR(63),DF1),(FCOR(64),TBEG),(FCOR(71),V0),
     5 (FCOR(72),BAND1),(FCOR(73),BAND2)
      EQUIVALENCE (ICOR(51),NDATAB),(ICOR(52),NDATA2),(ICOR(53),NDATAA),
     2 (ICOR(54),NSKIP1),(ICOR(55),NSKIP2),(ICOR(56),IPRO),
     3 (ICOR(57),NITR),(ICOR(58),IEXP),(ICOR(59),NLINE1),(ICOR(60),
     4    NLINE2)
      EQUIVALENCE (ICOR(61),NSKVEL),(ICOR(62),NSKDIS)
      EQUIVALENCE (ICOR(63),NSKV),(ICOR(64),NDATV)
      EQUIVALENCE (ICOR(65),NSKD),(ICOR(66),NDATD)
      EQUIVALENCE (FCOR(74),DTV),(FCOR(75),DDIS),(FCOR(76),XOIN)
      EQUIVALENCE (CARMOD(1),CORTIL(21))
      DATA ICOR/100*0/,FCOR/100*0.0/
      DATA CORTIL/'CORR','ECTE','D AC','CELE','ROGR','AM  ',10*'    ',
     *     'DERI','VED ','FROM',':   ',255*'    ',45*'    ',
     *     'AT F','QUAL','LY-S','PACE','D IN','TERV','ALS ','OF  ',
     *     2*'    ','SEC.',
     *     9*'    ','PEAK',' ACC','ELER','ATIO','N = ',9*'    ',' SEC',
     *     5*'    ','PEAK',' VEL','OCIT','Y   ','  = ',9*'    ',' SEC',
     *     5*'    ','PEAK',' DIS','PLAC','EMEN','T = ',9*'    ',' SEC',
     *     5*'    ','INIT','IAL ','VELO','CITY','  = ',3*'    ',' CMS',
     *     '/SEC',
     *     2*'    ','INIT','IAL ','DISP','. = ',3*'    ',' CMS',
     *     80*'    '/
      DATA FOM1/'(','72','A1)'/
      DATA CHAR/'0','1','2','3','4','5','6','7','8','9'/
      DATA FMTI2/'A3,'/,FMTI3/'A4,'/,BLANK/' '/
      DATA FOM2/'(','A4,','3A1,','2X,','2A4,','3X,','72','A1,','3X,',
     *     '3A4)'/
      DATA FMT131/4H(1H1,1H,,4H4(/),1H,,2H10,4HX,72,3HA1)/
      DATA FMT132/'(//,','10','X,','A4,','3A1,','2X,','2A4,','3X,','10',
     *     3HA1,,3H3X,,4H3A4)/
      EXTERNAL ERDUMP
      CALL ERRSET(209,1,1,2,ERDUMP)
      CALL ERRSET(218,1,1,2,ERDUMP)
      READ (5,2) LFILE,MFILE,NREAD,NWRITE
    2 FORMAT (6X,20I2)
      LFWRT=LFILE
      READ (5,139) BAND1,BAND2,DF1,DF2,ISHORT,INCARD,TLIMIT
  139 FORMAT (4F8.4,2I4,5F8.4)
      JFLS=0
      IF(1-LFILE)45,50,50
   45 KSKFL=LFILE-2
```

```
      IF(KSKFL)182,44,44
 44   CONTINUE
 42   READ(NREAD,END=46)
      GO TO 42
 46   CALL READNF(NREAD,KSKFL)
 50   CONTINUE
      IF (1-LFWRT) 65,60,60
 65   KSKFL=LFWRT-2
      IF (KSKFL) 182,66,66
 66   READ (NWRITE,END=67)
      GO TO 66
 67   CALL READNF (NWRITE,KSKFL)
 60   CONTINUE
      DO 100 NFILE=LFILE,MFILE
      DO 250 KS=1,33
      ALINE1(KS)=CORTIL(319)
      ALINE2(KS)=CORTIL(319)
250   CONTINUE
      READ(NREAD)CARMOD,IR,FR
      IFILE=IR(1)
      NDATA=IR(28)
      IF (NDATA.GT.10000) GO TO 187
      NDATAB=NDATA
      IF(IFILE-NFILE)180,32,180
 32   CONTINUE
      PERN=FR(1)
      IF (PERN-0.2) 966,966,967
966   CONTINUE
      ZTO=FR(2)
      DURN=FR(3)
      IF (INCARD) 600,600,34
600   CONTINUE
      IF (NDATA.NE.0) READ(NREAD) (TX(K),X(K),K=1,NDATA)
601   READ(NREAD,END=636)
      GO TO 601
636   CALL READNF(NREAD)
      GO TO 700
 34   READ(NREAD,END=36)
      GO TO 34
 36   CALL READNF(NREAD)
      READ(5,351)(TX(K),X(K),K=1,NDATA)
351   FORMAT(10F7.3)
700   CONTINUE
      JFLS=JFLS+1
      IF(JFLS.NE.1)CALL WRTNF(NWRITE)
      CALL INCORE(CARMOD(61),8)
      READ (5,80) VV,BB,IVOL
 80   FORMAT (A2,A1,I3)
      CALL INCORE
      I3=0
      IF (BB.NE.BLANK) GO TO 81
      CALL OUTCOR (VOLREF,NBYT)
      WRITE (6,83) VV
 83   FORMAT (1H1,A2)
      CALL OUTCOR
      GO TO 82
 81   CONTINUE
      I3=1
      CALL INCORE(CARMOD(61),8)
      READ (5,84) VV,BB,IVOL
 84   FORMAT (A3,A1,I3)
      CALL INCORE
```

```
      CALL OUTCOR (VOLREF,NBYT)
      WRITE (6,85) VV
   85 FORMAT (1H1,A3)
      CALL OUTCOR
   82 CONTINUE
      FMT=FMTI2
      IF (I3.EQ.1) FMT=FMTI3
      MS=IVOL/100
      KS=IVOL/10-10*MS
      VOL(1)=CHAR(MS+1)
      VOL(2)=CHAR(KS+1)
      VOL(3)=CHAR(IVOL-10*KS-100*MS+1)
      IF (NDATA.LE.0) GO TO 3037
      TDURN=TX(NDATA)
      IF (ABS(TDURN-DURN)-TLIMIT) 702,186,186
  702 CONTINUE
      KDATA=NDATA
  706 IF(TX(KDATA).LE.99.0)  GO TO 704
      KDATA=KDATA-1
      GO TO 706
  704 IF(KDATA .NE. NDATA)  WRITE(6,153)
  153 FORMAT(1H1,((' IN THIS FILE WE HAVE HAD TO RESTRICT THE DURATION
     .TO 100 SEC.')))
      NDATA=KDATA
      CALL INCORE(CARMOD(141),74)
      READ(5,52)TITLE,NLINE1
   52 FORMAT(72A1,I2)
      CALL INCORE
      FOM1(2)=ABCD(NLINE1)
      CALL OUTCOR(ALINE1,NBYT)
      WRITE(6,FOM1)(TITLE(KS),KS=1,NLINE1)
      CALL OUTCOR
      KLFT=(132-NLINE1)/2
      FMT131(5)=ABCD(KLFT)
      WRITE(6,FMT131)TITLE
      CALL INCORE(CARMOD(101),74)
      READ(5,52)SITE,NSITE
      CALL INCORE
      FOM2(2)=FMT
      FOM2(7)=ABCD(NSITE)
      CALL OUTCOR(ALINE2,NBT)
      WRITE (6,FOM2) VOLREF,VOL,CARMOD(63),CARMOD(64),
     2 (SITE(KS),KS=1,NSITE),(CARMOD(MS),MS=121,123)
      CALL OUTCOR
      NLINE2=31+NSITE
      KLFT=(132-NLINE2)/2
      FMT132(2)=ABCD(KLFT)
      FMT132(4)=FMT
      FMT132(9)=ABCD(NSITE)
      WRITE (6,FMT132) VOLREF,VOL,CARMOD(63),CARMOD(64),
     2  (SITE(KS),KS=1,NSITE),(CARMOD(MS),MS=121,123)
      WRITE(6,134)IFILE,NDATA,PERN,ZTO,DURN,TX(NDATA)
  134 FORMAT(//,43X,I2,2X,I5,4X,F6.4,2X,F6.4,3X,F7.3,2X,F7.3,//)
      SCALE1=1.0
      SCALE2=98.0665
      NSKIP=2
      NSKIP1=NSKIP
      DDT=0.01000
      DDT1=DDT
      IFEQ=0
      IFFP=0
      DO 12 I=1,NDATA
```

```
   12 X(I)=X(I)*SCALE2
      TMX1=TX(NDATA)
      NDATA2=NDATA
      PI=3.1415926535
      SCAL1=1.0
      SCAL2=1.0
      FN=BAND2+DF2
      FN2=FN
      DF=DF2
      IF (ISHORT.EQ.1) GO TO 20
      WNO=2*PI/PERN
      CALL ICR(NDATA,NSKIP,IFEQ,IFFP,DDT,SCAL1,SCAL2,WNO,ZTO,DF,FN)
      GO TO 89
   20 CONTINUE
      IFEQ=0
      IP(1)=1
      Q(1,1)=1.0
      Q(2,1)=0.0
      NK=1
      IFLO=1
      IFAS=0
      DDT=0.02
      IFSYM=1
      CALL SMU(NDATA,NK,IFLO,0,1,IFEQ,ZMIN,ZMAX,TX(NDATA),DDT,0,0,IFAS,
     *      IFSYM)
   89 CONTINUE
      WRITE(6,136)NDATA2,NDATA,TX(NDATA),DDT
  136 FORMAT(48X,I5,2X,I5,4X,F8.4,2X,F10.8,////)
      NDATAA=NDATA
      TMX2=TX(NDATA)
      NSKIP=10
      NSKIP2=NSKIP
      IFEQ=1
      IPRO=10
      NITR=1
      IEXP=0
      IFFP=0
      IFPSD=0
      IFPL1=0
      DDT=0.02000
      DDT2=DDT
      TMAX=TMX2
      FN=BAND1
      FN1=FN
      DF=DF1
      TBEG=0.0
      IFPL2=0
      CALL BAS(NDATA,NSKIP,IFEQ,IPRO,NITR,IEXP,IFFP,IFPSD,IFPL1,TMAX,DDT
     2,AMP,AMQ,AMR,SCAL1,SCAL2,FN,DF,TBEG,CO,IFPL2,AMPT,AMQT,AMRT)
      TMAX=TX(NDATA)
      NSKVEL=10
      NSKDIS=10
      DTA=0.02
      DTV=0.2
      DDIS=0.2
      FN=BAND1
      DF=DF1
      NSKV=2
      NSKD=5
      CALL HYPSVD(NDATA,NSKVEL,NSKDIS,DTA,DTV,DDIS,DF,FN)
      DDT=0.02
      TIME=0.0
```

```
      DO 2001 NGNR=1,NDATA
      TX(NGNR)=TIME
      TIME=TIME+DDT
 2001 CONTINUE
      AMP=0.0
      DO 3001 KM=1,NDATA
      XTMPX=ABS(X(KM))
      IF(AMP-XTMPX)3005,3001,3001
 3005 AMP=XTMPX
      PVAL(1)=X(KM)
      TVAL(1)=TX(KM)
 3001 CONTINUE
      AMQ=0.0
      DO 3010 KM=1,NDATA
      XTMPX=ABS(XD(KM))
      IF(AMQ-XTMPX)3015,3010,3010
 3015 AMQ=XTMPX
      PVAL(2)=XD(KM)
      TVAL(2)=TX(KM)
 3010 CONTINUE
      AMR=0.0
      DO 3020 KM=1,NDATA
      XTMPX=ABS(XDD(KM))
      IF(AMR-XTMPX)3025,3020,3020
 3025 AMR=XTMPX
      PVAL(3)=XDD(KM)
      TVAL(3)=TX(KM)
 3020 CONTINUE
      DO 3027 KM=1,3
      PLVAL(KM)=PVAL(KM)
      TLVAL(KM)=TVAL(KM)
 3027 CONTINUE
      NDATV=NDATA
      NDATD=NDATA
      IF(NSKV.EQ.1)GO TO 3037
      NDATV=0
      AMQPL=0.0
      DO 3030 KM=1,NDATA,NSKV
      NDATV=NDATV+1
      XTMPX=ABS(XD(KM))
      IF(AMQPL-XTMPX)3035,3030,3030
 3035 AMQPL=XTMPX
      PLVAL(2)=XD(KM)
      TLVAL(2)=TX(KM)
 3030 CONTINUE
 3037 CONTINUE
      IF(NSKD.EQ.1)GO TO 3047
      AMRPL=0.0
          NDATD=0
      DO 3040 KM=1,NDATA,NSKD
      NDATD=NDATD+1
      XTMPX=ABS(XDD(KM))
      IF(AMRPL-XTMPX)3045,3040,3040
 3045 AMRPL=XTMPX
      PLVAL(3)=XDD(KM)
      TLVAL(3)=TX(KM)
 3040 CONTINUE
 3047 CONTINUE
      WRITE(6,137)
  137 FORMAT(////)
      DO 22 KV=1,3
      PLVAL(KV)=-PLVAL(KV)
```

```
 22 PVAL(KV)=-PVAL(KV)
    FCOR(65)=TVAL(1)
    FCOR(66)=PVAL(1)
    FCOR(67)=TVAL(2)
    FCOR(68)=PVAL(2)
    FCOR(69)=TVAL(3)
    FCOR(70)=PVAL(3)
    FCOR(77)=TLVAL(2)
    FCOR(78)=PLVAL(2)
    FCOR(79)=TLVAL(3)
    FCOR(80)=PLVAL(3)
    WRITE(6,138)PLVAL
    WRITE(6,138)TLVAL
138 FORMAT(40X,3(4X,F12.6),/)
    VO=-XD(1)
    XOIN=-XDD(1)
    CALL OUTCOR(CORTIL(7),NBT)
    IF (FMT.EQ.FMTI3) GO TO 200
    WRITE (6,140) VOLREF,VOL,CARMOD(63),CARMOD(64),
   2 (CARMOD(KW),KW=121,123),NFILE
140 FORMAT (A3,3A1,2X2A4,4X,3A4,'FILE',1X,I2)
    GO TO 201
200 CONTINUE
    WRITE (6,240) VOLREF,VOL,CARMOD(63),CARMOD(64),
   * (CARMOD(KW),KW=121,123),NFILE
240 FORMAT (A4,3A1,2X2A4,4X,3A4,'FILE',1X,I2)
201 CONTINUE
    CALL OUTCOR
    CALL OUTCOR(CORTIL(281),NBT)
    WRITE(6,141)BAND1,BAND2
141 FORMAT('ACCFLEROGRAM IS BAND-PASS FILTERED BETWEEN  ',F6.3,
   22X,'AND ',1X,F6.3,1X,' CYC/SEC')
    CALL OUTCOR
    CALL OUTCOR(CORTIL(301),NBT)
    WRITE(6,144)NDATA
144 FORMAT(I6,2X,'INSTRUMENT AND BASELINE CORRECTED DATA')
    CALL OUTCOR
    CALL OUTCOR(CORTIL(329),NBT)
    WRITE(6,145)DDT
145 FORMAT(1X,F4.2,3X)
    CALL OUTCOR
    CALL OUTCOR(CORTIL(346),NBT)
    WRITE(6,146)PVAL(1),TVAL(1)
146 FORMAT(F11.5,1X,' CMS/SEC/SEC AT ',F8.4)
    CALL OUTCOR
    CALL OUTCOR(CORTIL(366),NBT)
    WRITE(6,147)PVAL(2),TVAL(2)
147 FORMAT(F11.5,1X,' CMS/SEC     AT ',F8.4)
    CALL OUTCOR
    CALL OUTCOR(CORTIL(386),NBT)
    WRITE(6,148)PVAL(3),TVAL(3)
148 FORMAT(F11.5,1X,' CMS',8X,' AT ',F8.4)
    CALL OUTCOR
    CALL OUTCOR(CORTIL(406),NBT)
    WRITE(6,149)VO
149 FORMAT(F11.5,1X)
    CALL OUTCOR
    CALL OUTCOR(CORTIL(417),NBYT)
    WRITE(6,149)XOIN
    CALL OUTCOR
    WRITE(6,150)(CORTIL(KP),KP=281,420)
150 FORMAT(26X,20A4)
```

```
      WRITE(NWRITE)CORTIL,ICOR,FCOR
      WRITE(NWR)(X(K),K=1,NDATA),(XD(J),J=1,NDATA),(XDD(L),L=1,NDATA)
      TVAL(2)=TLVAL(2)
      TVAL(3)=TLVAL(3)
      PVAL(2)=PLVAL(2)
      PVAL(3)=PLVAL(3)
      CALL TRILOT(TMAX,ALINE1,NLINE1,ALINE2,NLINE2,
     2 NDATA,NDATV,NDATD,NSKV,NSKD)
      IF (MOD(NFILE,3).EQ.0) IVOL=IVOL+1
      IF (IVOL.NE.100) GO TO 100
      VOLREF=VOLREF+1
      IVOL=0
  100 CONTINUE
      GO TO 184
  180 WRITE(6,181)
  181 FORMAT(10(/),25X,'PROG STOPS BECAUSE OF CLASH OF FILES.')
      GO TO 184
  182 WRITE(6,183)KSKFL
  183 FORMAT(10(/),25X, 'KSKFL =',I3,'LFILE SPECIFICATION IS WRONG AND H
     2ENCE STOP.')
      GO TO 184
  967 WRITE (6,968) PERN
      GO TO 184
  187 WRITE (6,189) NDATA
  189 FORMAT (///,1H0,10X,'NDATA IS',I12,'  --- TOO LARGE.')
      GO TO 184
  968 FORMAT (///1H0,'INSTRUMENT PERIOD = ',F10.4,5X'PROGRAM STOPS')
  186 WRITE (6,188) DURN,TDURN
  188 FORMAT(10(/),10X,'DURN = ',F9.4,2X,'TDURN = ',F9.4,
     2 'PROG STOPS AS TAPE STORAGE IS DOUBTFUL.',//)
  184 CONTINUE
      STOP
      END
```

Subroutine ICR (instrument correction) (Trifunac)

ICR is called by Volume II MAIN correction program

ICR corrects the acceleration for instrument response and high-frequency digitization errors.

Usage

```
CALL  ICR(NDATA,NSKIP,IFEQ,IFFP,DDT,SCALE1,SCALE2,
*        WNO,ZTO,DF,FN)
COMMON  DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
*        TX(10000),X(10000),T(10000).
```

Where

NDATA is no. of data points in the input accelerogram, TX(I), X(I).

NSKIP is no. of points to be skipped in the SMU decimation process.

IFEQ = 1, means equally spaced input data
0, means unequally spaced input data

IFFP = 0, means do not plot the Ormsby filter transfer function.

DDT is the time interval for equally-spaced data in SMU.

SCALE1 is the scaling factor for the time coordinate.

SCALE2 is the scaling factor for the acceleration coordinate.

WNO is the natural frequency of the instrument

ZTO is the fraction of critical damping of the instrument.

DF is the roll-off interval for the Ormsby filter in cps.

FN is the cut-off frequency for the Ormsby filter in cps.

DD is used by XYPLOT (system plot routine).

IP is used by SMU.

FMT1 is used by SMU in the punch options.

Q is the filter weight array input to SMU.

Z is the filtered output from SMU.

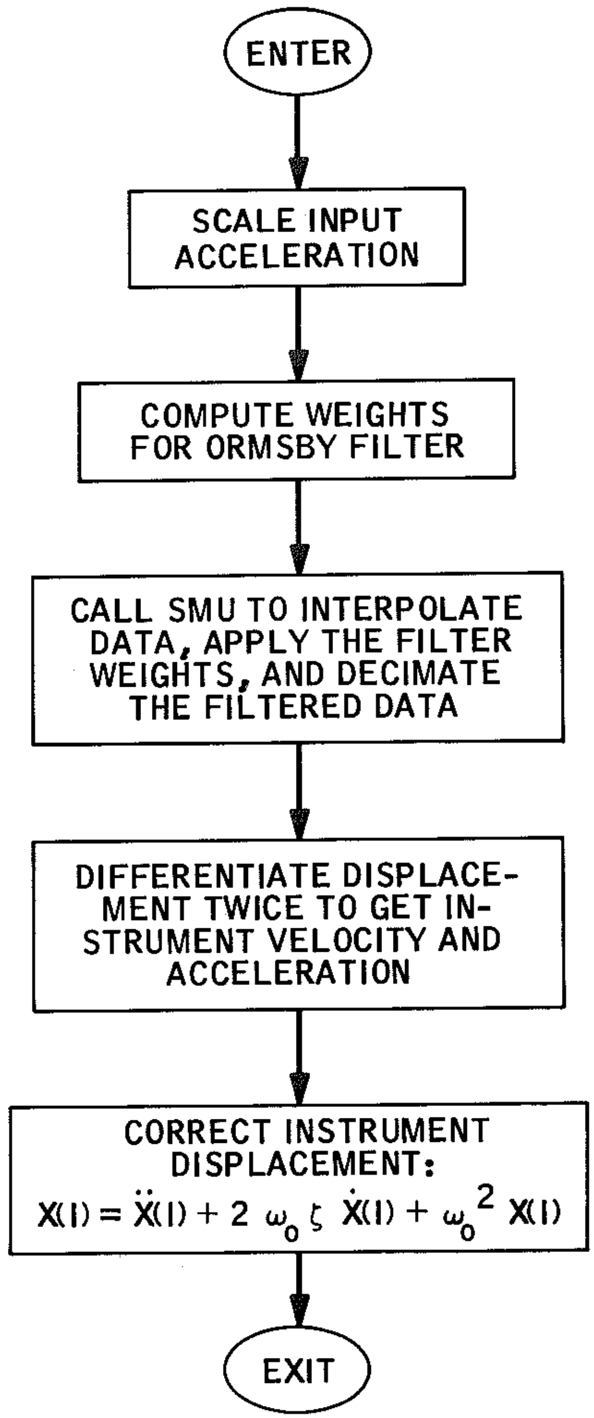
OBJAS is used by SMU in the punch options.

TX is the time coordinate array input and output for ICR.

X is the acceleration coordinate array input and output for ICR.

T is used by SMU for working space.

ICR FLOW CHART



```
      SUBROUTINE ICR(NDATA,NSKIP,IFEQ,IFFP,DDT,SCALE1,SCALE2
     1,WNO,ZTO,DF,FN)
C*****INSTRUMENT CORRECTION****
C
C     FOR THE METHOD REFER TO : HIGH FREQUENCY ERRORS AND INSTRUMENT
C     CORRECTIONS OF STRONG MOTION ACCELEROGRAMS BY M.D.TRIFUNAC,
C     F.E.UDWADIA AND A.G.BRADY,EARTHQUAKE ENGINEERING RESEARCH LABORATO
C     RY, EERL 71-05, CALIFORNIA INSTITUTE OF TECHNOLOGY
C
C     NDATA=NO. OF DATA IN THE INPUT ARRAY TX(I),X(I) I=1,NDATA
C     NSKIP=NO. OF POINTS TO BE SKIPPED IN THE SMU DECIMATION PROCESS
C     IFEQ=1 : INPUT DATA ARE EQUALLY SPACED
C         =0 : INPUT DATA ARE NOT EQUALLY SPACED
C     IFFIT=1 PLOT THE ORMSBY FILTER TRANSFER FUNCTION
C          =0 DO NOT PLOT THE ABOVE
C     DDT - TIME INTERVAL FOR EQUALLY SPACED OUTPUT DATA IF NSKIP=1
C     SCALE1=SCALING FACTOR FOR THE TIME COORDINATE
C     SCALE2=SCALING FACTOR FOR THE ACCELERATION COORDINATE
C     WNO=NATURAL FREQUENCY OF THE INSTRUMENT
C     ZTO=FRACTION OF CRITICAL DAMPING FOR THE SAME INSTRUMENT
C     FN=ROLL-OFF TERMINATION FREQUENCY FOR THE ORMSBY LOW-PASS FILTER
C     DF : FN-DF IS THE CUT-OFF FREQUENCY FOR THE SAME FILTER
      DIMENSION PP(3)
      COMMON DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
     *      TX(10000),X(10000) ,T(10000)
      REAL*8 TIME
      PP(1)=0
      PP(3)=1
      NK=1
      IFLO=1
      IFPUN=0
      IFPL=0
      IFPD=0
      NFF=200
      ZMAX=0
      DO 4 I=1,NDATA
      TX(I)=TX(I)*SCALE1
      X(I)=X(I)*SCALE2
      ZMAX=AMAX1(ZMAX,ABS(X(I)))
    4 CONTINUE
      ZMIN=-ZMAX
      TMAX=TX(NDATA)
C     IF THE NYQUIST  FREQUENCY FNDQ IS LARGER THAN FN THEN DO NOT LOW-
C     PASS FILTER
      FNDQ=1.0/(2.*DDT)
      IF(FN .GE. FNDQ) GO TO 102
C     CALCULATE THE ORMSBY FILTER WEIGHTS
      ALR=DF*DDT
      NN=1./ALR
      IF(NN .GE. NDATA) NN=NDATA
      ALC=(FN-DF)*DDT
      ALT=ALC+ALR
      Q(1,1)=ALT+ALC
      B1=2.*ALR
      SUM=0.5*Q(1,1)
      PI=3.1415926535
      DO 21 I=2,NN
      AN=(I-1)*PI
      AR1=2*AN*ALC
      AR2=2*AN*ALT
      AR4=B1*AN*AN
```

```
      Q(I,1)=(COS(AR1)-COS(AR2))/AR4
      SUM=SUM+Q(I,1)
   21 CONTINUE
      SUM=1.0/(2*SUM)
      TIME=0
      DO 22 I=1,NN
      Q(I,1)=Q(I,1)*SUM
      Z(I)=Q(I,1)
      T(I)=TIME
      TIME=TIME+DDT
   22 CONTINUE
      IP(1)=NN
      IF(IFFP .EQ. 0)  GO TO 101
      T(1)=0.
      TM=T(NN)
C     PLOT THE ORMSBY FILTER WEIGHTS
      CALL XYPLOT(NN,T,Z,0.0,TM,-1.0,1.0,PP,1)
      DIMENSION H(200),F(200)
C     CALCULATE THE TRANSFER FUNCTION FOR THE ABOVE FILTER WEIGHTS
      ANFF=NFF
      FDD=(1.5*FN)/ANFF
      AR5=6.28318*DDT
      DO 24 J=1,NFF
      AJ=J-1
      F(J)=FDD*AJ
      H(J)=Q(1,1)
      DO 23 I=2,NN
      WG=COS(AR5*F(J)*(I-1))
      H(J)=H(J) + 2.*Q(I,1)*WG
   23 CONTINUE
   24 CONTINUE
      FMAX=1.5*FN
      FMIN=-0.5*FN
C     PLOT THE TRANSFER FUNCTION
      CALL XYPLOT(NFF,F,H,FMIN,FMAX,-0.5,1.5,PP,1)
  101 IFAS=0
      NK=1
      IFSYM=1
C STEP #1 AND #2 FROM THE ABOVE REFERENCE
      CALL SMU(NDATA,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,ZMAX,TMAX,DDT,IFPL,
     1 IFPD ,IFAS,IFSYM)
      GO TO 103
  102 DO 104 I=1,NDATA
      Z(I)=X(I)
  104 T(I)=TX(I)
  103 NXM=NDATA-1
C STEP #3 AND #4 FROM THE ABOVE REFERENCE
      WNOSQ=WNO*WNO
      WNSQ=1.0/WNOSQ
      X(1)=Z(1)
      X(NDATA)=Z(NDATA)
      DO 71 I=1,NDATA
   71 Z(I)=Z(I)*WNSQ
      DINV2=1.0/(2.0*DDT)
      DINVSQ=1.0/(DDT*DDT)
      DO 12 I=2,NXM
      Z1D=(Z(I+1)-Z(I-1))*DINV2
      Z2D=(Z(I+1)-2*Z(I)+Z(I-1))*DINVSQ
      X(I)=Z2D+2.0*WNO*ZTO*Z1D+WNOSQ*Z(I)
      TX(I)=T(I)
   12 CONTINUE
      RETURN
```

END

Subroutine BAS (base-line correction) (Trifunac)

BAS is called by Volume II MAIN correction program.

BAS corrects the acceleration for base-line effects and for low-frequency digitization errors. BAS also calculates velocity and displacement.

Usage

```
CALL  BAS  (NDATA,NSKIP,IFEQ,IPRO,NITR,IEXP,IFFP,
*        IFPSD,IFPL1,TMAX,DDT,AMP,AMQ,AMR,SCALE1,
*        SCALE2,FN,DF,TBEG,CO,IFPL2,AMPT,AMQT,AMRT)
COMMON  DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
*        TX(10000),X(10000),T(10000)
COMMON  (B1/PVAL(3),TVAL(3),XD(5000),XDD(5000)
```

Where

NDATA is the no. of acceleration data points

NSKIP is the no. of points to be skipped in step #7, the decimation step (see the report by Trifunac, 1970).

IFEQ = 1, if the input data Z(I) to SMU are equally spaced

0, if the input data X(I) to SMU are unequally spaced

IPRO is [1+(1/2) (the no. of points in the running mean average)].

NITR is the no. of iterations (see the report).

IEXP = 1, will cause the following when NIT is larger than 1: after first iteration, go to step 1 instead of 16. This improves end effects by making a more accurate choice of a straight line on acceleration.

IFFP = 1, means plot the transfer functions of filters in steps 6 and 8 and plot the weighting function in step 8

IFPSD = 1, means plot the smoothed functions after steps 6 and 8

IFPL1 is an unused artifact.

TMAX is the maximum time coordinate for plot scaling.

DDT is the spacing of equally-spaced data.

AMP is peak acceleration.

AMQ is peak velocity.

AMR is peak displacement.

SCALE1 is scaling constant for time coordinates:

TX(I)=TX(I)*SCALE1.

SCALE2 is scaling constant for acceleration: X(I)=X(I)*SCALE2

FN is Ormsby filter roll-off frequency in cps.

DF is roll-off interval in cps.

TBEG is the first time coordinate to be included in the least-squares fit.

CO is the filtered initial velocity.

IFPL2 = 1, means plot the final acceleration, velocity, and displacement

AMPT is the time of peak acceleration.

AMQT is the time of peak velocity.

AMRT is the time of peak displacement.

DD is used by Caltech's XYPLOT Subroutine (SYSTEM PLOT SUBROUTINE).

IP is used by SMU (FILTER APPLICATION SUBROUTINE).

FMT1 is used by SMU under the punch options.

Q is the filter weights applied by SMU.

Z is the equally-spaced input to SMU.

OBJAS is used by SMU under the punch options.

TX is the time coordinate input to BAS and output from BAS.

X is the acceleration coordinate input to BAS and output from BAS.

T is used by SMU and BAS as working space.

PVAL contains max values AMP, AMQ, and AMR.

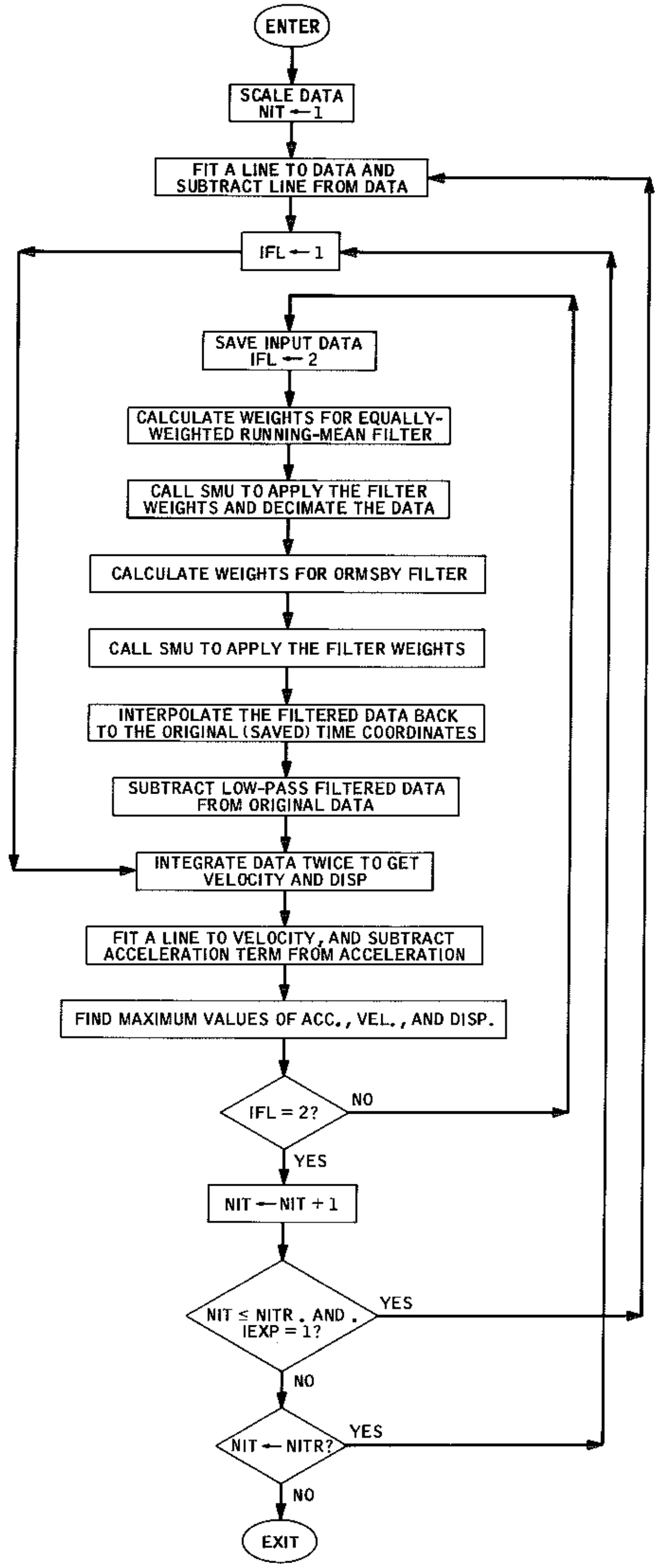
TVAL contains times AMPT, AMQT, and AMRT.

PP is used by XYPLOT.

XD is velocity on exit from BAS.

XDD is displacement on exit from BAS.

BAS FLOW CHART



```
      SUBROUTINE BAS(NDATA,NSKIP,IFEQ,IPRO,NITR,IEXP,IFFP,IFPSD,IFPL1,
     1TMAX,DDT,AMP,AMQ,AMR,SCALE1,SCALE2,FN,DF,TBEG,CO,IFPL2,
     2 AMPT,AMQT,AMRT)
C*****ZERO BASELINE CORRECTION *****
C
C     FOR THE METHOD USED IN THIS PROGRAM REFER TO : LOW FREQUENCY
C     ERRORS AND A NEW METHOD FOR ZERO BASELINE CORRECTION OF STRONG
C     MOTION ACCELEROGRAMS BY M.D.TRIFUNAC,EARTHQUAKE ENGINEERING
C     RESEARCH LABORATORY EERL 70-07,CALIFORNIA INSTITUTE OF TECHNOLOGY
C
C     NDATA=NO. OF POINTS IN THE INPUT ARRAY TX(I),X(I),I=1,NDATA
C     NSKIP=NO. OF POINTS TO BE SKIPPED IN THE DECIMATION STEP#7
C     IFEQ=1 INPUT DATA Z(I) TO SMU ARE EQUALLY SPACED,NOTE:DDT AND THE
C     TIME COORDINATES RETURNED BY SMU ARE THEN MEANINGLESS
C         =0 INPUT DATA X(I) TO SMU ARE NOT EQUALLY SPACED
C     IPRO= ONE HALF + 1 NO. OF POINTS IN THE RUNNING MEAN AVERAGE
C     FILTER,STEP #6, FOR INPUT DATA WITH DDT SPACING
C     NIT=NO. OF ITERATIONS AS DESCRIBED IN THE ABOVE REFERENCE
C     IEXP=1 A NEW FEATURE ADDED TO THE SCHEEME OF THE ABOVE REFERENCE,
C     NAMELY INSTEAD OF GOING TO #16 AFTER THE FIRST ITERATION WE GO TO
C     #1 AGAIN TO IMPROVE THE END EFFECTS, DUE TO INACURATE FIRST CHOISE
C     OF A STRAIGHT LINE ON ACCELERATION
C         =0: STANDARD PROCEDURE IS FOLLOWED AS IN THE ABOVE REFERENCE
C     IFFP=1  PLOT THE TRANSFER FUNCTION OF FILTERS IN THE STEPS #6 AND
C     #8 AND THE WEIGHTING FUNCTION IN THE STEP #8
C         =0 DO NOT PLOT THE ABOVE
C     IFPSD=1 PLOT THE SMOOTHED FUNCTIONS AFTER THE STEP #6 AND #8
C          =0 DO NOT PLOT THE ABOVE
C     IFPL1=1 PLOT THE VELOCITY AND DISPLACEMENT CURVES AFTER THE STEP
C     #3 AND #4'
C     IFPL2=1 PLOT THE FINAL ACCELERATION,VELOCITY AND DISPLACEMENT
C     IFPL2=0 DO NOT PLOT THE ABOVE
C          =0 DO NOT PLOT THE ABOVE
C     TMAX=MAXIMUM VALUE OF TX COORDINATE IN THE PLOTS
C     DDT=SPACING OF THE INTERPOLATED DATA FOR THE FITTING PURPOSES
C     AMP=PEAK ACCELERATION
C     AMQ=PEAK VELOCITY
C     AMR=PEAK DISPLACEMENT
C     SCALE1=SCALING CONSTANT FOR X COORDINATES TX(I)=TX(I)*SCALE1
C     SCALE2=SCALING CONSTANT FOR Y COORDINATES X(I)=X(I)*SCALE2
C     FN=ORMSBY FILTER ROLL-OFF TERMINATION FREQUENCY IN CPS
C     DF=FN-FC WHERE FC IS THE CUT-OFF FREQUENCY FOR THE SAME FILTER
C     TBEG-IF IT IS REQUIRED THAT THE LEAST SQUARE FITTING IS NOT PERFOR
C     MED ON THE WHOLE RECORD BUT FROM TBEG TO THE TX(NDATA) THEN TBEG
C     SHOULD BE GIVEN IN UNITS OF THE ORIGINAL INPUT DATA BEFORE THE
C     SCALING OF THE INPUT DATA WITH SCALE1
C
      COMMON DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
     *     TX(10000),X(10000) ,T(10000)
      COMMON /B1/ PVAL(3),TVAL(3),PP(3),XD(5000),XDD(5000)
      DIMENSION H(200),F(200),H1(200)
      DIMENSION ZIN(5000),TI(5000)
      REAL*8 TIME
      EQUIVALENCE (TI(1),XD(1)),(ZIN(1),XDD(1))
      TMX=TMAX
      PP(1)=0
      PP(3)=1
      NK=1
      IFLO=1
      IFPUN=0
      IFPL=0
```

```
      IFPD=0
      NFF=200
      IFAS=0
      IFSYM=1
      DO 89 I=1,NDATA
      DIC=TX(I)-TBEG
      IF(DIC .GT. 0.)  GO TO 90
   89 CONTINUE
   90 NBEG=I
      AM1=AMQ
      AM2=AMR
      IFFQ=IFEQ
      NSKK=NSKIP
      NIT=1
      DDTP=DDT
      DO 4 I=1,NDATA
      TX(I)=TX(I)*SCALE1
      X(I)=X(I)*SCALE2
    4 CONTINUE
C STEP#1 FITTING A STRAIGHT LINE C0 + C1*TX(I) TO THE INPUT ACCELEROGRAM
  337 V1=0
      X1=0
      NSTPS=NDATA-1
      DO 301 I=   1,NSTPS
      IF(I .EQ. NBEG) VBEG=V1
      IF(I .EQ. NBEG) DBEG=X1
      DLT=TX(I+1)-TX(I)
      X1=X1+V1*DLT+DLT**2/6.*(2.*X(I)+X(I+1))
  301 V1=V1+DLT/2.*(X(I)+X(I+1))
      T1=TX(NDATA) - TBEG
      T2=TX(NDATA)**2 - TBEG**2
      T3=TX(NDATA)**3 - TBEG**3
      A1=V1-VBEG
      A2=V1*TX(NDATA) - VBEG*TBEG - X1 + DBEG
      DDD=(4./3.)*T1*T3 - T2*T2
      DDC0=(4./3.)*A1*T3 - 2.*A2*T2
      DDC1= 4.    *T1*A2 - 2.*A1*T2
      C0=DDC0/DDD
      C1=DDC1/DDD
      DO 300 I=1,NDATA
  300 X(I)=X(I) -C0 -C1* TX(I)
   47 NDZ=NDATA
      IFL=1
      GO TO 200
  201 IFL=2
      DO 202 I=1,NDATA
      TI(I)=TX(I)
      ZIN(I)=X(I)
      Z(I)=X(I)
  202 CONTINUE
      ZMAX=AMP*1.2
      ZMIN=-ZMAX
      AMQ=AM1
      AMR=AM2
C     WRITE ALL INPUT PARAMETERS
      WRITE(6,36) NDATA,NK,IFLO,IFPUN,NSKIP,IFEQ,IFPL,IPRO,IFPD,NFF,IFAS
     1 ,NITR,IFXP,IFFP,IFPSD,IFPL,IFSYM
   36 FORMAT(1H ,20I5)
      WRITE(6,37) TMAX,DDT,AMP,AMQ,AMR,SCALE1,SCALE2, FN,DF,TBEG
   37 FORMAT(1H ,10E12.5)
C     CALCULATE THE WEIGHTS FOR THE EQUALLY WEIGHTED RUNNING MEAN FILTER
C     IN THE STEP #6
```

```
      SS=1.0/(2*IPRO-1)
      DO 7 I=1,IPRO
      Q(I,1)=SS
    7 CONTINUE
      IP(1)=IPRO
      DDT1=DDT
      IF(NIT .GT. 1 .OR. IFFP .EQ. 0) GO TO 102
      ANFF=NFF
      FDD=(1.5*FN)/ANFF
      AR6=6.28318*DDT1
C     CALCULATE THE TRANSFER FUNCTION H1(J) FOR THE RUNNING MEAN FILTER
C     IN THE STEP #6
      DO 30 J=1,NFF
      H1(J)=Q(1,1)
      DO 31 I=2,IPRO
      WG=COS(FDD*AR6*(J-1)*(I-1))
      H1(J)=H1(J)+ 2.*Q(I,1)*WG
   31 CONTINUE
   30 CONTINUE
C STEP #6 LOW-PASS FILTER WITH RUNNING MEAN FILTER
  102 CALL SMU(NDATA,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,ZMAX,TMAX,DDT,IFPL,
     1 IFPD ,IFAS,IFSYM)
      IF(IFPSD  .EQ. 0)  GO TO 104
      WRITE(6,33) X(1),X(NDATA),DDT
      SM=0.
      DO 28 I=1,NDATA
      SMM=ABS(Z(I))
      IF(SMM .GE. SM) SM=SMM
   28 CONTINUE
C STEP #7 PLOT LOW-PASS FILTERED DATA
      CALL XYPLOT(NDATA,T,Z,0.0,TMAX,-SM,SM,PP,1)
  104 ALR=DF*DDT
C     CALCULATE THE WEIGHTS FOR THE ORMSBY FILTER
      NN=1./ALR
      ALC=(FN-DF)*DDT
      ALT=ALC+ALR
      Q(1,1)=ALT+ALC
      B1=2.*ALR
      SUM=0.5*Q(1,1)
      PI=3.1415926535
      DO 21 I=2,NN
      AN=(I-1)*PI
      AR1=2*AN*ALC
      AR2=2*AN*ALT
      AR4=B1*AN*AN
      Q(I,1)=(COS(AR1)-COS(AR2))/AR4
      SUM=SUM+Q(I,1)
   21 CONTINUE
      SUM=1.0/(2*SUM)
      TIME=0.0
      DO 22 I=1,NN
      Q(I,1)=Q(I,1)*SUM
      Z(I)=Q(I,1)
      T(I)=TIME
      TIME=TIME+DDT
   22 CONTINUE
      IF(IFFP .EQ. 0)  GO TO 103
      IP(1)=NN
      T(1)=0.
      TM=T(NN)
      IF(NIT .GT. 1)  GO TO 103
C     PLOT THE WEIGHTS FOR THE ORMSBY FILTER
```

```
      CALL XYPLOT(NN,T,Z,0.0,TM,-1.0,1.0,PP,1)
C     CALCULATE THE TRANSFER FUNCTION H FOR THE ORMSBY FILTER
      AR5=6.28318*DDT
      DO 24 J=1,NFF
      AJ=J-1
      F(J)=FDD*AJ
      H(J)=Q(1,1)
      DO 23 I=2,NN
      WG=COS(AR5*F(J)*(I-1))
      H(J)=H(J) + 2.*Q(I,1)*WG
   23 CONTINUE
   24 CONTINUE
C     PLOT THE TRANSFER FUNCTION H1 FOR THE RUNNING MEAN FILTER AND H
C     FOR THE ORMSBY FILTER
      FMAX=1.5*FN
      FMIN=-0.5*FN
      CALL XYPLOT(NFF,F,H1,FMIN,FMAX,-0.5,1.5,PP,1)
      CALL XYPLOT(NFF,F,H,FMIN,FMAX,-0.5,1.5,PP,1)
      DO 32 I=1,NFF
   32 H(I)=H(I)*H1(I)
C     PLOT THE RESULTING TRANSFER FUNCTION H1*H
      CALL XYPLOT(NFF,F,H ,FMIN,FMAX,-0.5,1.5,PP,1)
  103 IP(1)=NN
      NSKIP=1
      IFEQ=1
      DO 27 I=1,NDATA
      T(I)=TX(I)
      Z(I)=X(I)
   27 CONTINUE
C STEP #8 LOW-PASS FILTER DATA FROM THE STEP #7 WITH ORMSBY FILTER
      CALL SMU(NDATA,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,ZMAX,TMAX,DDT,IFPL,
     1 IFPD ,IFAS,IFSYM)
      IF(IFPSD .EQ. 0)  GO TO 105
      WRITE(6,33) X(1),X(NDATA),DDT
   33 FORMAT(1H ,9E12.5)
      SM=0.
      DO 25 I=1,NDATA
      SMM=ABS(Z(I))
      IF(SMM .GE. SM) SM=SMM
   25 CONTINUE
C     PLOT THE LOW- PASS FILTERED ACCELEROGRAM - ZERO BASELINE
      CALL XYPLOT(NDATA,T,Z,0.0,TMAX,-SM,SM,PP,1)
C STEP #9 INTERPOLATE Z(I) TO THE ZERO BASELINE X(J) TO GET THE BASELINE
C     DATA AT THE ORIGINAL TIME COORDINATES TX(I)
  105 CONTINUE
      T(1)=TX(1)
      Z(1)=X(1)
      NPOINT=NDATA-1
      I=2
      J=1
  182 T(I)=TI(I)
  183 IF (T(I)-TX(J+1)) 186,184,184
  184 J=J+1
      IF (J-NPOINT) 183,183,185
  186 CONTINUE
      Z(I)=X(J)+(X(J+1)-X(J))*((T(I)-TX(J))/(TX(J+1)-TX(J)))
      I=I+1
      IF (I-NDZ) 182,182,188
  185 CONTINUE
C
C           EXTRAPOLATE UP TO NSKIP POINTS ON THE LAST STRAIGHT LINE.
C
```

```
      J=J-1
      II=I
      DO 187 I=II,NDZ
      T(I)=TI(I)
      Z(I)=X(J)+(X(J+1)-X(J))*((T(I)-TX(J))/(TX(J+1)-TX(J)))
  187 CONTINUE
  188 NDATA=NDZ
C STEP #10 SUBSTRACT THE ZERO BASELINE Z(I) FROM THE ACCELEROGRAM ZIN(I)
      DO 26 I=1,NDATA
      ZIN(I)=ZIN(I)-Z(I)
      X(I)=ZIN(I)
      TX(I)=TI(I)
   26 CONTINUE
C STEP #2 AND #11 COMPUTE THE VELOCITY XD(I) ASSUMING XD(1)=0.
  200 XD(1)=0.
      XDD(1)=0.
      NBETA=NDATA-1
      DO 8 I=1,NBETA
      DLT=TX(I+1)-TX(I)
      XD(I+1)=XD(I)+DLT*(X(I)+X(I+1))*0.5
      XDD(I+1)=XDD(I)+XD(I)*DLT+(1./6.)*DLT*DLT*(2.*X(I)+X(I+1))
    8 CONTINUE
C STEP #3 AND #12 COMPUTE CO AND C1 BY FITTING CO + C1*TX(I) TO THE
C       VELOCITY COMPUTED IN THE STEP #2 OR #11
      V1=0
      X1=0
      NSTPS=NDATA-1
      DO 303  I=    1,NSTPS
      IF(I .EQ. NBEG) VBEG=V1
      IF(I .EQ. NBEG) DBEG=X1
      DLT=TX(I+1) -TX(I)
      X1=X1+V1*DLT + DLT**2/6.*(2.*XD(I) + XD(I+1))
  303 V1=V1 + DLT/2.*(XD(I) + XD(I+1))
      T1=TX(NDATA) - TBEG
      T2=TX(NDATA)**2 - TBEG**2
      T3=TX(NDATA)**3 - TBEG**3
      A1=V1-VBEG
      A2=V1*TX(NDATA) - VBEG*TBEG - X1 + DBEG
      DDD=(4./3.)*T1*T3 - T2*T2
      DDC0=(4./3.)*A1*T3 - 2.*A2*T2
      DDC1= 4.   *T1*A2 - 2.*A1*T2
      C0=DDC0/DDD
      C1=DDC1/DDD
C STEPS #4, #13, #14, #4' AND #15
      AMP=0.
      AMQ=0.
      AMR=0.
      DO 29 I=1,NDATA
      X(I)=X(I)-C1
      Z(I)=X(I)
      T(I)=TX(I)
      XD(I)=XD(I)-C0-C1*TX(I)
      XDD(I)=XDD(I)-C0*TX(I)-(C1*TX(I)*TX(I))/2.
      XM1=ABS(X(I))
      IF(XM1-AMP)2902,2902,2901
 2901 AMP=XM1
      AMPT=TX(I)
      TVAL(1)=AMPT
      PVAL(1)=X(I)
 2902 CONTINUE
      XM=ABS(XD(I))
      IF(XM-AMQ)2904,2904,2903
```

```
 2903 AMQ=XM
      AMQT=TX(I)
      TVAL(2)=AMQT
      PVAL(2)=XD(I)
 2904 CONTINUE
      XMM=ABS(XDD(I))
      IF(XMM-AMR)29,29,2905
 2905 AMR=XMM
      AMRT=TX(I)
      TVAL(3)=AMRT
      PVAL(3)=XDD(I)
   29 CONTINUE
C     GO TO THE STEP #5 VIA THE STATEMENT 201
      IF(IFL .EQ. 1) GO TO 201
      IF(IFPL2 .EQ. 0)  GO TO 453
C     PLOT THE CORRECTED ACCELEROGRAM ,VELOCITY AND DISPLACEMENT
      CALL XYPLOT(NDATA,TX,X  ,0.0,TMX,-AMP,AMP,PP,1)
      CALL XYPLOT(NDATA,TX,XD ,0.0,TMX,-AMQ,AMQ,PP,1)
      CALL XYPLOT(NDATA,TX,XDD,0.0,TMX,-AMR,AMR,PP,1)
C     FINAL ESTIMATE OF THE INITIAL VELOCITY IS CO
  453 WRITE(6,34) CO,C1
   34 FORMAT(1H ,2E12.5)
      NIT=NIT+1
      SCALE1=1.
      SCALE2=1.
      IFEQ=IFFQ
      DDT=DDTP
      NSKIP=NSKK
      IF(NIT .LE. NITR  .AND. IFXP .EQ. 1)  GO TO 337
      IF(NIT  .LE. NITR) GO TO 47
      RETURN
      END
```

Subroutine HYPSVD (high-pass velocity and displacement)(Trifunac)

HYPSVD is called by the Vol. II correction program (MAIN).

HYPSVD filters long periods out of the computed velocity and displacement.

Usage

```
CALL  HYPSVD (NDATA, NSKVEL, NSKDIS, DTA, DTV, DDIS,
*     DF, FN)
COMMON DD(3), IP(5), FMT1(9), Q(1000, 1), Z(10000), OBJAS(18),
*     TX(10000), X(10000), T(10000)
COMMON /B1/PVAL(3), TVAL(3), PP(3), XD(5000), XDD(5000)
```

Where

NDATA is no. of points in acceleration, velocity, and displacement.

NSKVEL is no. of points to be skipped in decimation after filtering the velocity.

NSKDIS is no. of points to be skipped in decimation after filtering the displacement.

DTA is the time interval of acceleration (usually 0.02 sec).

DTV is the time interval of velocity for computing filter weights.

DDIS is the time interval of displacement for computing filter weights.

DF is the frequency interval from roll-off point to cut-off point.

FN is the filter roll-off frequency (high-pass).

The relevant elements of the catch-all COMMON are:

/B1/

XD is velocity

XDD is displacement

/BLANK/

X is acceleration

Z is unfiltered input to SMU (the filtering subroutine)

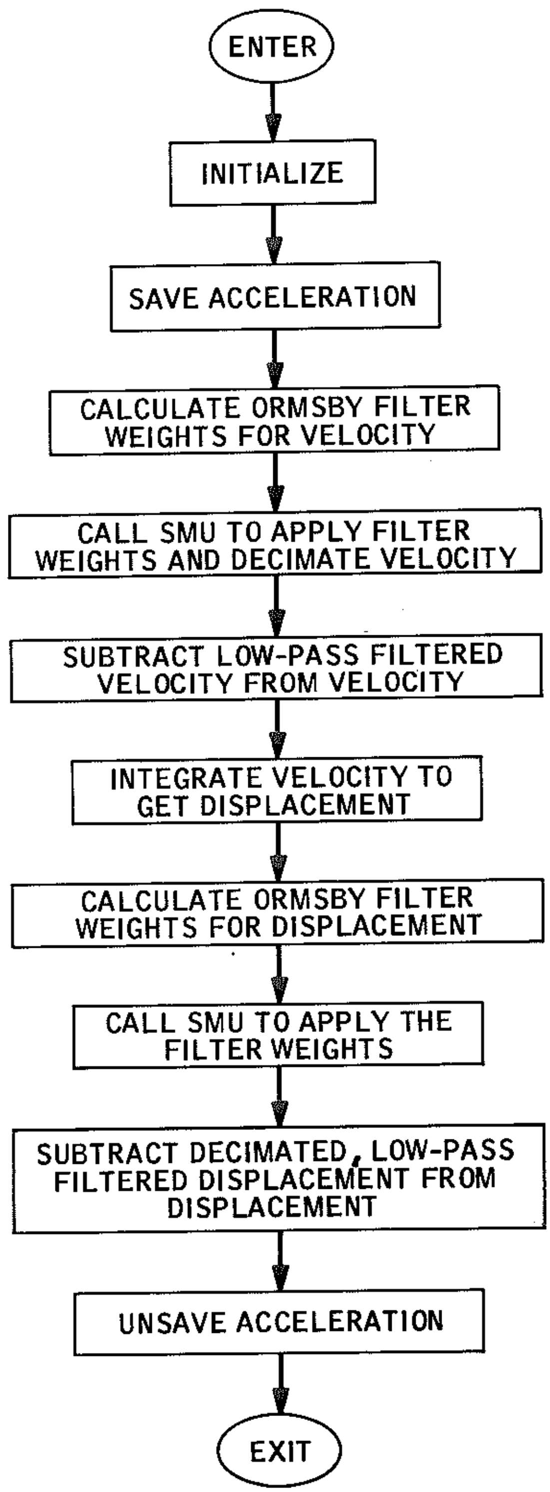
Q is the filter weight input to SMU

TX is the times of acceleration

T is a working space

T, Z, and Q are used during HYPSVD, so they must not contain useful information upon entry to HYPSVD.

HYPSVD FLOW CHART



```
      SUBROUTINE HYPSVD(NDATA,NSKVEL,NSKDIS,DTA,DTV,DDIS,DF,FN)
      COMMON DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
     *      TX(10000),X(10000) ,T(10000)
      COMMON /B1/ PVAL(3),TVAL(3),PP(3),XD(5000),XDD(5000)
      EQUIVALENCE (FNSKV,FNSKD)
      DO 160 I=1,NDATA
      XDD(I)=XD(I)
  160 T(I)=X(I)
      II=1
      DO 167 I=1,NDATA,NSKVEL
      Z(II)=XD(I)
      II=II+1
  167 CONTINUE
      NDATD=II-1
      IFEQ=1
      NK=1
      IFLO=1
      IFPUN=0
      NSKIP=1
      DDT=DTV
      IFPL=0
      IFPD=0
      IFAS=0
      IFSYM=1
C     CALCULATE THE WEIGHTS FOR THE ORMSBY FILTER
      ALR=DF*DDT
      NN=1./ALR
      ALC=(FN-DF)*DDT
      ALT=ALC+ALR
      Q(1,1)=ALT+ALC
      B1=2.*ALR
      SUM=0.5*Q(1,1)
      PI=3.1415926535
      DO 24 I=2,NN
      AN=(I-1)*PI
      AR1=2*AN*ALC
      AR2=2*AN*ALT
      AR4=B1*AN*AN
      Q(I,1)=(COS(AR1)-COS(AR2))/AR4
      SUM=SUM+Q(I,1)
   24 CONTINUE
      SUM=1.0/(2*SUM)
      DO 25 I=1,NN
      Q(I,1)=Q(I,1)*SUM
   25 CONTINUE
      IP(1)=NN
      CALL        SMU(NDATD,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,ZMAX,TMAX,DDT,
     1 IFPL,IFPD,IFAS,IFSYM)
      Z(NDATD+1)=Z(NDATD)*2.-Z(NDATD-1)
      IZ=1
      FNSKV=1.0/NSKVEL
      DTVV=1.0/DTV
      DO 165 I=1,NDATA,NSKVEL
      ZZZ=(Z(IZ+1)-Z(IZ))
      ZFF=ZZZ*FNSKV
      ZDF=ZZZ*DTVV
      DO 166 J=1,NSKVEL
      L=I+J-1
      AL=J-1
      XD(L)=-ZFF*AL-Z(IZ)+XDD(L)
      T(L)=T(L)-ZDF
```

```
  166 CONTINUE
      IZ=IZ+1
  165 CONTINUE
C     INTEGRATE XD(I) TO GET XDD(I)
      V0=XD(1)
      XDD(1)=0.
      DERV=DTA*DTA/6
      DO 170 I=1,NDATA
      XDD(I+1)=XDD(I)+XD(I)*DTA+DERV*(2.0*T(I)+T(I+1))
  170 CONTINUE
      DERV=XDD(NDATA)/(NDATA-1)
      DO 175 I=1,NDATA
  175 XDD(I)=XDD(I)-(I-1)*DERV
      II=1
      DO 161 I=1,NDATA,NSKDIS
      Z(II)=XDD(I)
      II=II+1
  161 CONTINUE
      NDATD=II-1
      IFEQ=1
      NK=1
      IFLO=1
      IFPUN=0
      NSKIP=1
      DDT=DDIS
      IFPL=0
      IFPD=0
      IFAS=0
      IFSYM=0
      CALCULATE THE WEIGHTS FOR THE ORMSBY FILTER
      ALR=DF*DDT
      NN=1./ALR
      ALC=(FN-DF)*DDT
      ALT=ALC+ALR
      Q(1,1)=ALT+ALC
      B1=2.*ALR
      SUM=0.5*Q(1,1)
      DO 21 I=2,NN
      AN=(I-1)*PI
      AR1=2*AN*ALC
      AR2=2*AN*ALT
      AR4=B1*AN*AN
      Q(I,1)=(COS(AR1)-COS(AR2))/AR4
      SUM=SUM+Q(I,1)
   21 CONTINUE
      SUM=1.0/(2*SUM)
      DO 23 I=1,NN
      Q(I,1)=Q(I,1)*SUM
   23 CONTINUE
      IP(1)=NN
      CALL        SMU(NDATD,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,ZMAX,TMAX,DDT,
     1 IFPL,IFPD,IFAS,IFSYM)
      IZ=1
      Z(NDATD+1)=2.0*Z(NDATD)-Z(NDATD-1)
      FNSKD=1.0/NSKDIS
      DO 173 I=1,NDATA,NSKDIS
      ZDF=(Z(IZ+1)-Z(IZ))*FNSKD
      DO 174 J=1,NSKDIS
      L=I+J-1
      AL=J-1
      XDD(L)=XDD(L)-ZDF*AL-Z(IZ)
  174 CONTINUE
```

```
      IZ=IZ+1
  173 CONTINUE
      DO 163 I=1,NDATA
  163 X(I)=T(I)
      RETURN
      END
```

Subroutine SMU (smoothing) (Trifunac)

SMU applies filter weights for any digital filter. SMU is called by ICR, BAS, and HYPSVD.

Usage

```
CALL SMU (NDATA,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,
*       ZMAX,TMAX,DDT,IFPL,IFPD,IFAS,IFSYM)
COMMON DD(3),ID(5),FMT1(9),W(1000,1),Z(10000),OBJAS(18),
*       VA(10000),ZA(10000),T(10000)
```

Where

NDATA is no. of points in array to be filtered.

NK is no. of filter dimensions.

IFLO = 1, means low-pass filter the data

IFPUN = 0, means do not punch the filtered data

NSKIP is the no. of points to skip during decimation (after filtering).

IFEQ = 1, means data are equally spaced in Z(I).

ZMIN is minimum value of Z for the plot option.

ZMAX is maximum value of Z for the plot option.

TMAX is maximum value of time for the plot option.

DDT is the time interval to interpolate unequally-spaced data.

IFPL = 1, means plot the filtered data

IFPD = 1, means punch the interpolated data

IFAS = 0, means do not apply the filter to an assymmetric extension of the data

IFSYM = 0, means do not apply the filter to a symmetric extension

DD is used by Caltech's plotting routines.

ID is the no. of filter weights in each filter dimension.

FMT1 is used in the punch options.

W is the filter weights.

Z is the equally-spaced input data and filtered output.

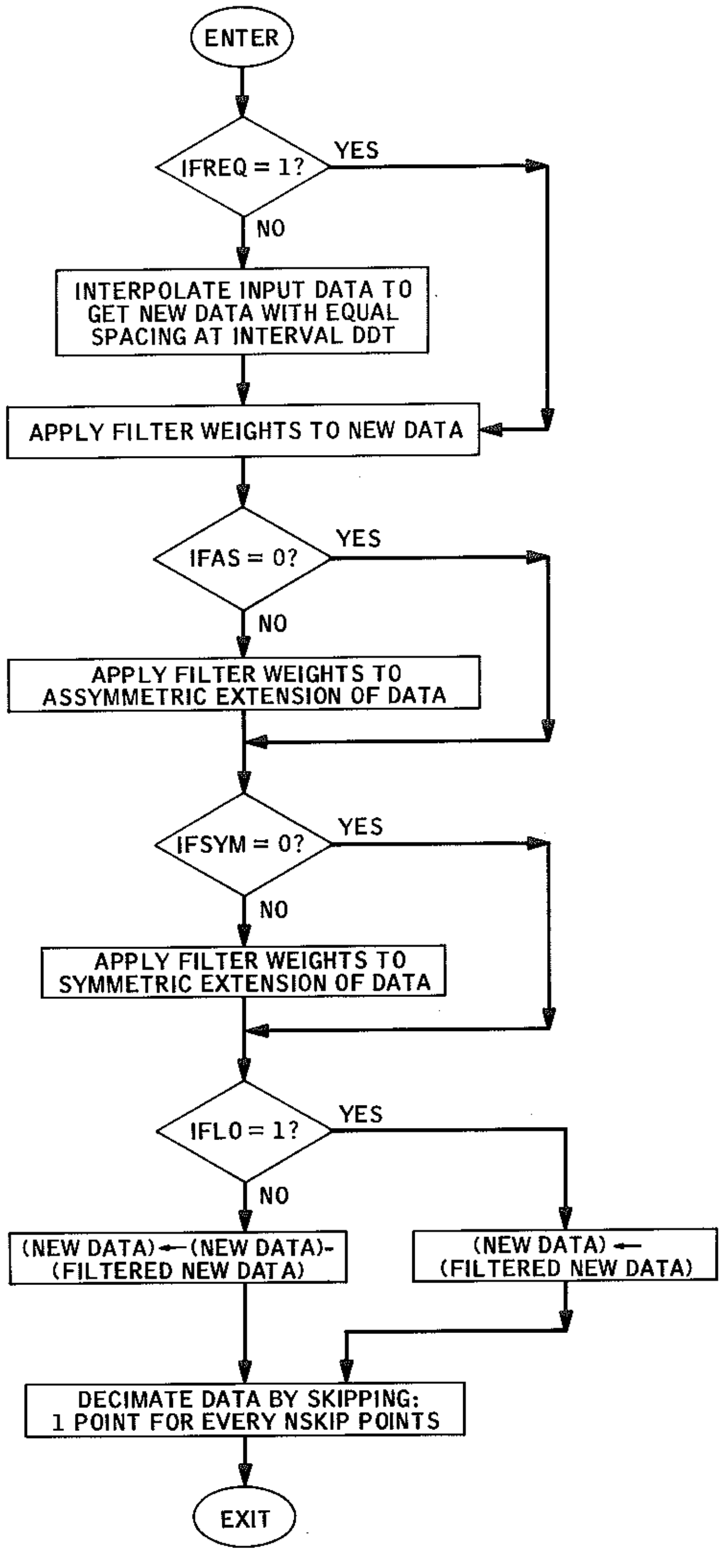
OBJAS is used in the punch options.

VA is the input unequally-spaced time data.

ZA is the input unequally-spaced function values.

T is the equally-spaced time input data and output data.

SMU FLOW CHART



```
      SUBROUTINE SMU(NDATA,NK,IFLO,IFPUN,NSKIP,IFEQ,ZMIN,ZMAX,TMAX,DDT,
     1 IFPL,IFPD,IFAS,IFSYM)
      COMMON DD(3),ID(5),FMT1(9),W(1000,1),Z(10000),OBJAS(18),
     *       VA(10000),ZA(10000),T(10000)
      REAL*8 TIME
      IF (IFEQ .EQ. 1) GO TO 8
      T(1)=VA(1)
      Z(1)=ZA(1)
      NPOINT=NDATA-1
      I=2
      J=1
      TIME=DDT
  182 T(I) =T(1)+TIME
  183 IF (T(I)-VA(J+1)) 186,184,184
  184 J=J+1
      IF (J-NPOINT) 183,183,185
  186 CONTINUE
      Z(I)=ZA(J)+(ZA(J+1)-ZA(J))*((T(I)-VA(J))/(VA(J+1)-VA(J)))
      I=I+1
      TIME=TIME+DDT
      GO TO 182
  185 NDATA=I-1
      IF(IFPD .EQ. 1) PUNCH 49, OBJAS
      IF(IFPD .EQ. 1) PUNCH FMT1, (T(I),Z(I),I=1,NDATA)
    8 DO 88 K=1,NK
      IPRO=ID(K)
      DO 897 I=1,NDATA
      IPPP=I-1+IPRO
      IF(IPPP .GT. NDATA) IPPP=NDATA
      ZA(I)=0.
      IF(I-1) 772,772,773
  773 II=I-1
      IP=1+I-IPRO
      IF(IP .LT. 1)  IP=1
      DO 896 J=IP,II
      M=-J+1+I
      ZA(I)=ZA(I)+Z(J)*W(M,K)
  896 CONTINUE
  772 DO 898 J=I,IPPP
      M=J-I+1
      ZA(I)=ZA(I)+Z(J)*W(M,K)
  898 CONTINUE
  897 CONTINUE
      IF(IFAS .EQ. 0) GO TO 25
      IF (IPRO.GT.NDATA) IPRO=NDATA
      DO 1 J=1,IPRO
      IJK=IPRO-J+1
      DO 2 I=1,IJK
      M=IPRO-IJK+I
      ZA(J)=ZA(J)-Z(I)*W(M,K)
    2 CONTINUE
    1 CONTINUE
      DO 3 J=1,IPRO
      ILK=NDATA-IPRO+J
      ILL=NDATA-J+1
      DO 4 I=ILK,NDATA
      M=IPRO-I+ILK
      ZA(ILL)=ZA(ILL)-Z(I)*W(M,K)
    4 CONTINUE
    3 CONTINUE
   25 IF(IFSYM  .EQ. 0)  GO TO 24
```

```
      IF (IPRO.GT.NDATA) IPRO=NDATA
      IPROM=IPRO-1
      IF(IPROM .GE. NDATA)  IPROM=NDATA
      DO 6 J=1,IPROM
      IJK=IPRO-J+1
      DO 7 I=2,IJK
      M=IPRO-IJK+I
      ZA(J)=ZA(J) + Z(I)*W(M,K)
    7 CONTINUE
    6 CONTINUE
      NDMF=NDATA-1
      DO 18 J=1,IPROM
      ILK=NDATA-IPRO+J
      ILL=NDATA-J+1
      DO 9 I=ILK,NDMF
      M=IPRO-I+ILK
      ZA(ILL)=ZA(ILL) + Z(I)*W(M,K)
    9 CONTINUE
   18 CONTINUE
   24 IF(IFLO .EQ. 1) GO TO 21
      DO 5 L=1,NDATA
      Z(L)=Z(L)-ZA(L)
    5 CONTINUE
      GO TO 88
   21 DO 23 L=1,NDATA
      Z(L)=ZA(L)
   23 CONTINUE
   88 CONTINUE
      IF(IFPL .EQ. 1) CALL XYPLOT(NDATA,T,Z,0.0,TMAX,ZMIN,ZMAX,DD,1)
      IF (IFPUN .EQ. 0) GO TO 48
      PUNCH 49, OBJAS
   49 FORMAT(20A4)
      PUNCH FMT1, (T(I),Z(I),I=1,NDATA,NSKIP)
   48 I=1
      DO 50 J=1,NDATA,NSKIP
      ZA(I)=Z(J)
      VA(I)=T(J)
      I=I+1
   50 CONTINUE
      NDATA=I-1
      DO 51 I=1,NDATA
      Z(I)=ZA(I)
      T(I)=VA(I)
   51 CONTINUE
      DDT=(T(NDATA)-T(1))/(NDATA-1)
      RETURN
      END
```

Subroutine TRILOT (Vijayaraghavan, Justiss)

TRILOT is called by the Volume II MAIN correction program. It produces a plot of acceleration, velocity, and displacement.

Usage

```
CALL TRILOT (TMAX,ALINE1,NLINE1,ALINE2,NLINE2,
*       NDATA,NDATVL,NDATDS,NSKVEL,NSKDIS)
COMMON DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
*       TX(10000),X(10000),T(10000)
COMMON /B1/PVAL(3),TVAL(3),PP(3),XD(5000),XDD(5000)
```

Where

TMAX is the time length of the accelerogram.

ALINE1 is the earthquake title.

NLINE1 is the no. of characters in ALINE1.

ALINE2 is the accelerogram title.

NLINE2 is the no. of characters in ALINE2.

NDATA is the no. of data points in acceleration.

NDATVL is the no. of data points in velocity (unused).

NDATDS is the no. of data points in displacement (unused).

NSKVEL is the no. of data points to skip for velocity.

NSKDIS is the no. of data points to skip for displacement.

COMMON /BLANK/ has the following information:

TX is the equally-spaced time coordinates.

X is the equally-spaced acceleration coordinates.

DD is used by Caltech's XYPLOT subroutine.

COMMON /B1/ has

PVAL is the maximum values of acceleration, velocity, and displacement.

TVAL is the times of the maximum values in PVAL.

XD is the displacement coordinates.

Subroutines VDBASE, XTICK, YLABEL, and VDLAST are used inside TRILOT. Their functions are tied to Caltech plotting routines and so they are of little use at other installations.

```
      SUBROUTINE TRILOT(TMAX,ALINE1,NLINE1,ALINE2,NLINE2,
     2   NDATA,NDATVL,NDATDS,NSKVEL,NSKDIS)
      DIMENSION UNITV(3)
      DIMENSION BCDW(33)
      DIMENSION INTFMT(3),XTIL(20),NRIP(3),NRFMT(3)
      DIMENSION FLFMT(5)
      DIMENSION ALINE1(33),ALINE2(33)
      DIMENSION AXLEND(30)
      DIMENSION CC(15)
      DIMENSION AXLTRA(7)
      DIMENSION TNEWS(2000)
      COMMON DD(3),IP(5),FMT1(9),Q(1000,1),Z(10000),OBJAS(18),
     *     TX(10000),X(10000),T(10000)
      COMMON /B1/ PVAL(3),TVAL(3),PP(3),XD(5000),XDD(5000)
      EQUIVALENCE (TNEWS(1),X(1))
      EQUIVALENCE (SU,SL3UNT),(S,SLINE3),(Y,YLINE3)
      DATA INTFMT/2H(I,1H3,1H)/
      DATA FLFMT/2H(F,1H6,1H.,1H1,1H)/
      DATA AXLTRA/100.0,250.0,500.0,750.0,1000.0,1250.0,1500.0/
      NPRT(X)=2+ALOG10(ABS(X)+5.0E-6)
      INTFIX(X)=INT(X+5.0E-6*SIGN(X))
      DD(1)=0.0
      DD(2)=0.0
      DD(3)=0.0
      NAXEND=26
      DO 8 N=1,19
    8 AXLEND(N)=N*5
      DO 9 N=20,26
    9 AXLEND(N)=AXLTRA(N-19)
      FRAC=(6.0/7.0)
      SSYM=0.07000
      STIME=0.10
      SLTIM=0.1
      SLTIM2=0.085
      SLINE1=0.12
      SLINE2=0.10
      SLINE3=0.1
      SL3UNT=0.085
      WSYM=FRAC*SSYM
      WTIME=STIME*FRAC
      WLTIM=SLTIM*FRAC
      WLTIM2=FRAC*SLTIM2
      WLINE1=FRAC*SLINE1
      WLINE2=SLINE2*FRAC
      WLINE3=FRAC*SLINE3
      WL3UNT=SL3UNT*FRAC
      CTK=0.07
      XSHEET=15.0
      YSHEET=10.0
      YPAP=8.5
      XPAP=11.0
      XOS=0.5*(XSHEET-XPAP)
      YOS=0.5*(YSHEET-YPAP)
      CALL SYSPLT(XOS,YOS,3)
      CALL SYSPLT(XOS+XPAP,YOS,2)
      CALL SYSPLT(XOS+XPAP,YOS+YPAP,2)
      CALL SYSPLT(XOS,YOS+YPAP,2)
      CALL SYSPLT(XOS,YOS,2)
      XL=8.2500
      XRITE=1.1000
      XORG=XOS+XPAP-(XL+XRITE)
```

```
      SLOCL=0.075
      WRLOW=STIME+SLTIM+2.0*SLOCL
      CLBGR=0.25
      YBOT=1.0
      YDIS=YOS+YBOT+WRLOW
      YLDIS=1.5
      YVEL=YDIS+YLDIS+CLBGR
      YLVEL=1.5
      YACC=YVEL+YLVEL+CLBGR
      YLACC=1.5
C     CHOICE OF TIME AXIS PARAMETERS
      TMP=TMAX/5.0
      ITMP=INT(TMP+5.0E-6)
      TMP2=ITMP*5.0
      IF(TMP2-TMAX)10,15,15
   10 ITMP=ITMP+1
   15 TLAST=ITMP*5.0
      ITMTIK=ITMP
      IF(TLAST-100.0)151,151,155
  151 ITMARK=ITMTIK/2
      DTIME=10.0
      IF(ITMARK-4)152,152,156
  152 ITMARK=ITMTIK
      DTIME=5.0
      GO TO 156
  155 CONTINUE
      ITMARK= INT((TLAST/20.0)+5.0E-6)
      DTIME=20.0
  156 CONTINUE
C     SET-UP AND WRITE FIRST 3 LINES
      NUMTOT=3
      DO 5 N=1,3
      TMP=PVAL(N)
      NCHAR=ALOG10(ABS(TMP)+0.05)+3
C     THIS PART HAS BEEN MODIFIED TO ACCOMMODATE FLOATING POINT FIELDS.
      IF (TMP) 2,3,3
    2 NRIP(N)=NCHAR+1
      GO TO 555
    3 NRIP(N)=NCHAR
  555 NREMT(N)=NCHAR+1
    5 NUMTOT=NUMTOT+NRIP(N)
      CL3LG=0.11
      CL2L3=0.10
      CL1L2=0.10
      YLINE3=YACC+YLACC+CL3LG
      YLINE2=YLINE3+SLINE3+CL2L3
      YLINE1=YLINE2+SLINE2+CL1L2
      XL1=XORG+0.5*(XL-FLOAT(NLINE1)*WLINE1)
      CALL SYSSYM(XL1,YLINE1,SLINE1,ALINE1,NLINE1,0.0)
      XL2=XORG+0.5*(XL-FLOAT(NLINE2)*WLINE2)
      CALL SYSSYM(XL2,YLINE2,SLINE2,ALINE2,NLINE2,0.0)
      WL3=(46.0+FLOAT(NUMTOT))*WLINE3+24.0*WL3UNT
      WL3=WL3+WLINE3+WSYM-2.0*WL3UNT
      CC(1)=0.5*(XL-WL3)+WSYM+WLINE3
      CC(2)=12.0*WLINE3
      CC(3)=WLINE3
      CC(4)=10.0*WLINE3
      CC(5)=(NRIP(1)+1)*WLINE3
      CC(6)=12.0*WL3UNT
      CC(7)=13.0*WLINE3
      CC(8)=(NRIP(2)+1)*WLINE3
      CC(9)=8.0*WL3UNT
```

```
      CC(10)=10.0*WLINE3
      CC(11)=(NRIP(3)+1)*WLINE3
      XTIL(1)=XORG+CC(1)
      DO 12 J=2,11
   12 XTIL(J)=XTIL(J-1)+CC(J)
      XTIL(2)=XTIL(2)-0.025
      Y1=Y+0.02
      Y2=Y+S-0.035
      XSYM=XTIL(1)-WLINE3-0.5*WSYM
      YSYM=YLINE3+0.5*SLINE3
      ISYM=1
      CALL SYSSYM(XSYM,YSYM,SSYM,ISYM,-1,0.0)
      CALL SYSSYM(XTIL(1),Y,S,'PEAK VALUES ',12,0.0)
      CALL SYSSYM(XTIL(2),Y1,S,'.',1,0.0)
      CALL SYSSYM(XTIL(2),Y2,S,'.',1,0.0)
      CALL SYSSYM(XTIL(3),Y,S,'  ACCEL = ',10,0.0)
      CALL SYSSYM(XTIL(5),Y,SU,'CM/SEC/SEC  ',12,0.0)
      CALL SYSSYM(XTIL(6),Y,S,'  VELOCITY = ',13,0.0)
      CALL SYSSYM(XTIL(8),Y,SU,'CM/SEC  ',8,0.0)
      CALL SYSSYM(XTIL(9),Y,S,'  DISPL = ',10,0.0)
      CALL SYSSYM(XTIL(11),Y,SU,'CM',2,0.0)
      DO 14 J=1,3
      FLFMT(2)=ABCD(NRFMT(J))
      C1=WLINE3
      CALL OUTCOR(BCDW,NW)
      WRITE(6,FLFMT)PVAL(J)
      CALL OUTCOR
      IF(PVAL(J))141,142,142
  141 C1=0.0
  142 XTMP=XTIL(4+(J-1)*3)-C1
      CALL SYSSYM(XTMP,Y,S,BCDW,NW*4,0.0)
   14 CONTINUE
      AXLAST=AXLEND(1)
      DO 20 N=1,NAXEND
      IF(AXLEND(N)-ABS(PVAL(1)))16,25,25
   16 AXLAST=AXLEND(N+1)
   20 CONTINUE
   25 CONTINUE
      CALL SYSPLT(XORG,YACC,3)
      YTMP=YACC+YLACC
      CALL SYSPLT(XORG,YTMP,2)
      CALL SYSPLT(XORG+CTK,YTMP,2)
      XTMP=XORG+CTK
      NTKTOT=3
      NTOT=NTKTOT+1
      DO 30 N=1,NTKTOT
      J=NTOT-N
      YTMP=YACC+YLACC*FLOAT(J)/FLOAT(NTOT)
      CALL SYSPLT(XORG,YTMP,3)
      CALL SYSPLT(XTMP,YTMP,2)
   30 CONTINUE
      CALL SYSPLT(XORG,YACC,3)
      CALL SYSPLT(XORG+XL,YACC,2)
      CALL YLABEL(XORG,YACC,YLACC,SLTIM,WLTIM,AXLAST)
      CALL XTICK(XORG,YACC,XL,ITMTIK,CTK)
      CL1=0.1
      CL2=0.08
      NFLD=NPRT(AXLAST)
      XYLABL=XORG-(FLOAT(NFLD)+0.5)*WLTIM-0.03
      XLINE1=XYLABL-CL1-SLTIM2-CL2
      XLINE1=XLINE1+0.15
      WACC=12.0*WTIME
```

```
      YTMP=YACC+0.5*(YLACC-WACC)
      CALL SYSSYM(XLINE1,YTMP,STIME,'ACCELERATION',12,90.0)
      XLINE2=XLINE1+CL1+SLTIM2
      WACU=10.0*WLTIM2
      YTMP=YACC+0.5*(YLACC-WACU)
      CALL SYSSYM(XLINE2,YTMP,SLTIM2,'CM/SEC/SEC',10,90.0)
      TRATE=TLAST/XL
      YRATE=AXLAST/(0.5*YLACC)
      YTMP=YACC+0.5*YLACC
      TLFT=-XORG*TRATE
      TRITE=(XSHEET-XORG)*TRATE
      ACHI=(YSHEET-YTMP)*YRATE
      ACLO=-YTMP*YRATE
      CALL XYPLOT(NDATA,TX,X,TLFT,TRITE,ACLO,ACHI,DD,0)
      XSYM=XORG+TVAL(1)/TRATE
      YSYM=YTMP-PVAL(1)/YRATE
      ISYM=1
      CALL SYSSYM(XSYM,YSYM,SSYM,ISYM,-1,0.0)
C     VELOCITY AND DISPLACEMENT PLOTS
      VELMAX=ABS(PVAL(2))
      DISMAX=ABS(PVAL(3))
      CALL VDLAST(VELMAX,UNITV,VELAST)
      CALL VDLAST(DISMAX,UNITV,DILAST)
C      VELOCITY PLOT
      CALL VDBASE(XORG,YVEL,YLVEL,XL,ITMTIK,CTK)
      CALL YLABEL(XORG,YVEL,YLVEL,SLTIM,WLTIM,VELAST)
C       WRITE Y-AXIS TITLE
      WVEL=8.0*WTIME
      YTMP=YVEL+0.5*(YLVEL-WVEL)
      CALL SYSSYM(XLINE1,YTMP,STIME,'VELOCITY',8,90.0)
      WVELU=6.0*WLTIM2
      YTMP=YVEL+0.5*(YLVEL-WVELU)
      CALL SYSSYM(XLINE2,YTMP,SLTIM2,'CM/SEC',6,90.0)
      YTMP=YVEL+0.5*YLVEL
      YRATE=VELAST/(0.5*YLVEL)
      VEHI=(YSHEET-YTMP)*YRATE
      VELO=-YTMP*YRATE
      ISK=1
      DO 1331 KTRAN=1,NDATA,NSKVEL
      XD(ISK)=XD(KTRAN)
      TNEWS(ISK)=TX(KTRAN)
      ISK=ISK+1
 1331 CONTINUE
      NDATV=ISK-1
      CALL XYPLOT(NDATV,TNEWS,XD,TLFT,TRITE,VELO,VEHI,DD,0)
      XSYM=XORG+TVAL(2)/TRATE
      YSYM=YTMP-PVAL(2)/YRATE
      ISYM=1
      CALL SYSSYM(XSYM,YSYM,SSYM,ISYM,-1,0.0)
C     DISPLACEMENT PLOT
      CALL VDBASE(XORG,YDIS,YLDIS,XL,ITMTIK,CTK)
      CALL YLABEL(XORG,YDIS,YLDIS,SLTIM,WLTIM,DILAST)
       WDIS=12.0*WTIME
      YTMP=YDIS+0.5*(YLDIS-WDIS)
      CALL SYSSYM(XLINE1,YTMP,STIME,'DISPLACEMENT',12,90.0)
      WDISU=2.0*WLTIM2
      YTMP=YDIS+0.5*(YLDIS-WDISU)
      CALL SYSSYM(XLINE2,YTMP,SLTIM2,'CM',2,90.0)
      YTMP=YDIS+0.5*YLDIS
      YRATE=DILAST/(0.5*YLDIS)
      DIHI=(YSHEET-YTMP)*YRATE
      DILO=-YTMP*YRATE
```

```
      XSYM=XORG+TVAL(3)/TRATE
      YSYM=YTMP-PVAL(3)/YRATE
      ISYM=1
      CALL SYSSYM(XSYM,YSYM,SSYM,ISYM,-1,0.0)
C     WRITING THE ZERO ON THE LOWEST TIME AXIS
      YXLABL=YDIS-(SLTIM+SLOCL)
      XTMP=XORG
      CALL SYSSYM(XTMP,YXLABL,SLTIM,'0',1,0.0)
      DO 50 NT=1,ITMARK
      IVAL=NT*DTIME
      IF(IVAL-9)420,420,440
  420 INTFMT(2)=IABCD(2)
      C2=1.5*WLTIM
      GO TO 46
  440 IF(IVAL-99)42,42,44
   42 INTFMT(2)=IABCD(3)
      C2=2.0*WLTIM
      GO TO 46
   44 C2=2.5*WLTIM
      INTFMT(2)=IABCD(4)
   46 XIN=XORG+FLOAT(IVAL)/TRATE
      CALL OUTCOR(BCDW,NW)
      WRITE(6,INTFMT)IVAL
      CALL OUTCOR
      NPT=NW*4
      CALL SYSSYM(XIN-C2,YXLABL,SLTIM,BCDW,NPT,0.0)
   50 CONTINUE
      YXLABL=YXLABL-(STIME+SLOCL)
      XTMP=XORG+0.5*(XL-14.0*STIME)
      CALL SYSSYM(XTMP,YXLABL,STIME,'TIME - SECONDS',14,0.0)
      DD(3)=1.0
      ISK=1
      DO 1332 KTRAN=1,NDATA,NSKDIS
      XDD(ISK)=XDD(KTRAN)
      TNEWS(ISK)=TX(KTRAN)
      ISK=ISK+1
 1332 CONTINUE
      NDATD=ISK-1
      CALL XYPLOT(NDATD,TNEWS,XDD,TLFT,TRITE,DILO,DIHI,DD,1)
      RETURN
      END
```

```
SUBROUTINE VDBASE(XORG,YVEL,YLVEL,XL,ITMTIK,CTK)
YTMP=YVEL+YLVEL
XTMP=XORG+CTK
CALL SYSPLT(XORG,YVEL,3)
CALL SYSPLT(XORG,YTMP,2)
CALL SYSPLT(XTMP,YTMP,2)
DYTMP=0.25*YLVEL
YTMP=YTMP-DYTMP
CALL SYSPLT(XORG,YTMP,3)
CALL SYSPLT(XTMP,YTMP,2)
XTMPL=XORG+XL
YTMP=YVEL+0.5*YLVEL
CALL SYSPLT(XORG,YTMP,3)
CALL SYSPLT(XTMPL,YTMP,2)
YTMP=YTMP-DYTMP
CALL SYSPLT(XORG,YTMP,3)
CALL SYSPLT(XTMP,YTMP,2)
CALL SYSPLT(XORG,YVEL,3)
CALL SYSPLT(XTMPL,YVEL,2)
 CALL XTICK(XORG,YVEL,XL,ITMTIK,CTK)
RETURN
END
```

```
      SUBROUTINE XTICK(XORG,YACC,XL,ITMTIK,CTK)
      DXINTK=XL/FLOAT(ITMTIK)
      YTMP=YACC+CTK
      XTMP=XORG+XL
      DO 34 NT=1,ITMTIK
      XTK=XTMP-(NT-1)*DXINTK
      CALL SYSPLT(XTK,YACC,3)
      CALL SYSPLT(XTK,YTMP,2)
   34 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE YLABEL(XORG,YACC,YLACC,SLTIM,WLTIM,AXLAST)
      DIMENSION INTFMT(3),BCDW(20)
      DATA INTFMT/2H(I,1H3,1H)/
      IAXL=(AXLAST+5.0E-6)
      NFLD=2+ALOG10(ABS(AXLAST)+5.0E-6)
      INTFMT(2)=IABCD(NFLD)
      XYLABL=XORG-(FLOAT(NFLD)+0.5)*WLTIM-0.03
      IAXL=-1*IAXL
      CALL OUTCOR(BCDW,NW)
      WRITE(6,INTFMT)IAXL
      CALL OUTCOR
      NPT=NW*4
      YTMP=YACC+YLACC-0.5*SLTIM
      CALL SYSSYM(XYLABL,YTMP,SLTIM,BCDW,NPT,0.0)
      IZERO=0
      CALL OUTCOR(BCDW,NW)
      WRITE(6,32)IZERO
   32 FORMAT(I2)
      CALL OUTCOR
      NPT=NW*4
      XTMP=XORG-2.5*WLTIM-0.03
      YZ=YTMP-0.5*YLACC
      CALL SYSSYM(XTMP,YZ,SLTIM,BCDW,NPT,0.0)
      IAXL=-1*IAXL
      CALL OUTCOR(BCDW,NW)
      WRITE(6,INTFMT)IAXL
      CALL OUTCOR
      NPT=NW*4
      YTMP=YACC-0.5*SLTIM
       CALL SYSSYM(XYLABL,YTMP,SLTIM,BCDW,NPT,0.0)
      RETURN
      END
```

```
      SUBROUTINE VDLAST (VMAX,UNITV,VLAST)
      DIMENSION UNITV(3)
      UNITV(1)=2.0
      UNITV(2)=10.0
      UNITV(3)=25.0
      NCHAR=1+AMAX1(0.0,ALOG10(VMAX+5.0E-6))
      UNIT=UNITV(NCHAR)
      INTMP=VMAX/UNIT
      TMP=FLOAT(INTMP)*UNIT
      IF(TMP-VMAX)38,40,40
   38 VLAST=(INTMP+1)*UNIT
      GO TO 42
   40 VLAST=TMP
   42 RETURN
      END
```

IIXWRT MAIN Program (Vijayaraghavan)

This program reads in a specified number of files from the Volume II tape and prints out each file of heading and acceleration data in the format of the Volume II report (Figure 5).

Usage

The program reads in the following data:

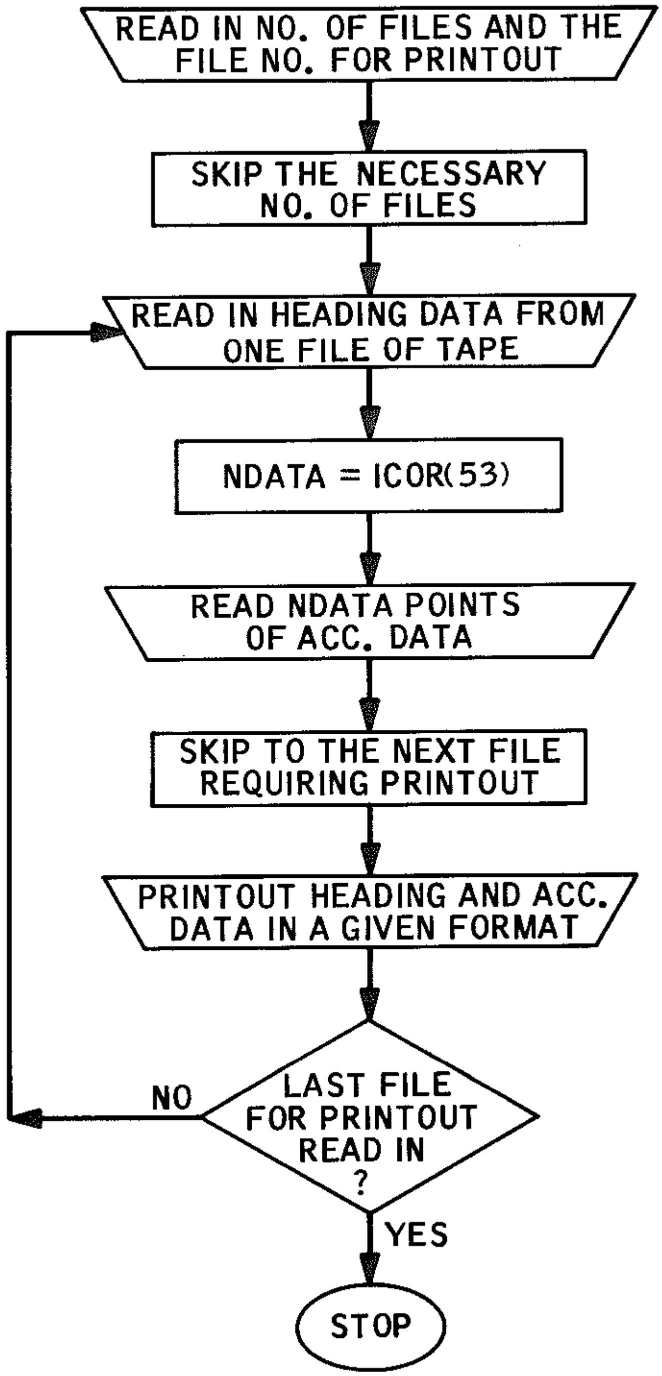
NFILES, (IFILE(K), K=1, NFILES)

Where

NFILES is the total number of files to be read in for printout.

IFILE(K) is the particular file number to be read in.

IIXWRT MAIN PROGRAM FLOW CHART



```
      INTEGER TITL(72),ADDR(72),IFILE(60)
      DIMENSION ADD(18),Y(2)
      DIMENSION ICOR(100),FCOR(100),CORTIL(500),X(5000),IX(5000)
      DIMENSION FORM(10),FOR1(11),COMP(9),STA(15),FOR2(9)
      DIMENSION TIT(42)
      EQUIVALENCE (TIT(1),TITL(1))
      EQUIVALENCE (IX(1),X(1))
      DATA FORM/'(1H ',',','4A4,',','10','X,',','72','A1,',','10','X,',','8A4)'/
      DATA FOR1/'(1X1','5A1,','10','X,','72','A1,','10','X,','9A1,',
     *     '5A4,','2A1)'/
      DATA FOR2/'(1X,','42','A1,','21X,','14A4',',I2,','2A4,','A1,',
     *     '1H.)'/
      NRD=10
      READ (5,1) NFILES,(IFILE(K),K=1,NFILES)
    1 FORMAT (40I2)
      IF (1-IFILE(1)) 2,3,3
    2 KSKIPF=IFILE(1)-2
    4 READ (NRD,END=5)
      GO TO 4
    5 CALL READNF(NRD,KSKIPF)
    3 CONTINUE
      DO 100 NF=1,NFILES
      READ (NRD) CORTIL,ICOR,FCOR
      NDATA=ICOR(53)
      READ (NRD) (X(K),K=1,NDATA)
    6 READ (NRD,END=7)
      GO TO 6
    7 IF (NF.EQ.NFILES) GO TO 10
      IF (IFILE(NF)+1 -IFILE(NF+1)) 8,9,9
    8 KSKIPF=IFILE(NF+1)-IFILE(NF)-1
      CALL READNF(NRD,KSKIPF)
      GO TO 10
    9 CALL READNF(NRD)
   10 CONTINUE
      WRITE (6,56)
   56 FORMAT (1H1)
C
C          WRITE FIRST LINE OF TITLE.
C
      CALL INCORE(CORTIL(161),74)
      READ (5,11) TITL,NTIT
   11 FORMAT (72A1,I2)
      CALL INCORE
      NT=(115-NTIT)/2
      FORM(4)=ABCD(NT-16)
      FORM(6)=ABCD(NTIT)
      FORM(8)=ABCD(100-NT-NTIT)
      WRITE(6,FORM) (CORTIL(K),K=7,10),(TITL(K),K=1,NTIT),(CORTIL(K),
     *     K=181,188)
C
C          SECOND LINE.
C
      CALL INCORE(CORTIL(121),74)
      READ (5,11) ADDR,NADD
      CALL INCORE
      CALL INCORE (CORTIL(141),9)
      READ (5,11) COMP
      CALL INCORE
      CALL INCORE (CORTIL(101),15)
      READ (5,11) STA
      CALL INCORE
```

```
      CALL INCORE(CORTIL(110),2)
      READ (5,11) Y
      CALL INCORE
      NT=(115-NADD)/2
      FOR1(3)=ABCD(NT-15)
      FOR1(7)=ABCD(100-NT-NADD)
      FOR1(5)=ABCD(NADD)
      WRITE (6,FOR1) STA,(ADDR(K),K=1,NADD),COMP,(CORTIL(K),K=105,109),Y
C
C           THIRD LINE.
C
      CALL INCORE(CORTIL(201),74)
      READ (5,11) TITL,NTIT
      CALL INCORE
      CALL INCORE(CORTIL(281),72)
      READ (5,12) ADD
   12 FORMAT (11A4,1X,3A4,I2,4X,2A4,A1)
      CALL INCORE
      WRITE (6,FOR2) TIT,ADD
C
C           FOURTH, FIFTH, SIXTH LINES AND THE REST.
C
      WRITE (6,50) FCOR(66),FCOR(65),FCOR(58),FCOR(67),FCOR(70),FCOR(69)
   50 FORMAT (1H0,'PEAK VALS',6X,'ACLN =',F7.1,2X,'CM/SEC/SEC AT',
     *     F6.2,' SEC',       6X,'VELO =',F7.1,2X,'CM/SEC AT',
     *     F6.2,' SEC',       6X,'DISP =',F7.1,2X,'CM AT',
     *     F6.2,' SEC')
      WRITE (6,51) (CORTIL(K),K=401,403),FCOR(71),(CORTIL(K),
     *     K=413,415),FCOR(76)
   51 FORMAT (1H0,30X,3A4,' =',F11.5,'  CM/SEC',8X,
     *                3A4,' =',F11.5,'  CM')
      WRITE (6,52) (CORTIL(K),K=301,312),(CORTIL(K),K=321,331)
   52 FORMAT (1H0,9X,12A4,'IN  MM/SEC/SEC    ',11A4,//)
      DO 20 I=1,NDATA
   20 IX(I)=-(10.0*X(I)+SIGN(0.5,X(I)))
      IFIN=800
      IF (IFIN.GT.NDATA) IFIN=NDATA
      DO 53 I=1,IFIN,200
      IFN=I+199
      IF (IFN.GT.IFIN) IFN=IFIN
      WRITE (6,54) (IX(K),K=I,IFN)
      WRITE (6,55)
   53 CONTINUE
      WRITE (6,58)
   58 FORMAT (1H ,/)
      IF (IFIN.GE.NDATA) GO TO 60
      IFIN=IFIN+1
      DO 57 I=IFIN,NDATA,1000
      IFN=I+999
      IF (IFN.GT.NDATA) IFN=NDATA
      DO 59 J=I,IFN,200
      JFIN=J+199
      IF (JFIN.GT.NDATA) JFIN=NDATA
      WRITE (6,54) (IX(K),K=J,JFIN)
      WRITE (6,55)
   59 CONTINUE
      WRITE (6,58)
   57 CONTINUE
   60 CONTINUE
   55 FORMAT (1H )
   54 FORMAT (1H ,5I6,3X,5I6,4X,5I6,3X,5I6)
  100 CONTINUE
```

```
      STOP                                                          IIXW 124
      END                                                           IIXW 125
```

DATA PROCESSING FOR VOLUME III: RESPONSE SPECTRA, VOLUME III TAPE AND VOLUME V TAPE

The response spectra processed by the programs presented in this section are calculated from the corrected data published in the various parts of Volume II of the report "Strong Motion Earthquake Accelerograms" (Hudson, et al, 1971) and stored on the Volume II tapes. For the standard corrected accelerograms which are available at equally-spaced time intervals with $\Delta t = 0.02$ sec (on Volume II tape), an approach based on the exact analytical solution of the Duhamel integral for successive linear segments of excitation appears to be the most practical (Iwan, 1960). This approach has been described in detail by Nigam and Jennings (1968). Its basic advantage is that if the relative velocity $\dot{x}_i$ and displacement x_i of an oscillator are known at some time t_i, the complete response can be computed by a step-by-step application of recursive equations of the form

$$\begin{Bmatrix} x_{i+1} \\ \dot{x}_{i+1} \end{Bmatrix} = \begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} x_i \\ \dot{x}_i \end{Bmatrix} + \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} a_i \\ a_{i+1} \end{Bmatrix} \qquad (1)$$

where a_i is the acceleration amplitude at t_i. The constant matrices A and B depend only on ζ the fraction of critical damping and ω_n the natural frequency of the oscillator. The step-by-step integration of the differential equation of motion, equation (1), is carried out by the SPCTRA and PCNO3 subroutines while the matrices A and B are calculated by the PCNO4 subroutine.

For each acceleration component two figures showing response spectra and two pages of tables containing most of the spectral ordinates are obtained. The first plot is that of the true relative velocity response spectrum, SV (Figure 6). The name of the earthquake, the instrument location, and the damping values appear in the descriptive titles. The first title gives the name, date and time of occurrence of the earthquake; the second is comprised of two labels, the instrument location or observation station and the particular component. The first identification label, for example IIIA001, indicates that the processed record belongs to Part A of Volume II, this volume containing corrected data, and that this record is the first of the complete series. This record's uncorrected data appear in Volume I, Part A; its response spectra in Volume III, Part A; and its Fourier spectra in Volume IV, Part A. The five continuous line plots correspond to damping values of 0, 2, 5, 10, and 20 percent of critical, and these curves will usually be free of any confusion with the zero damped curve having the greatest ordinates. The dashed curve on this plot is the Fourier amplitude spectrum, FS, calculated at the same periods as for the relative velocity response spectra. The ordinates for the spectra are in units of in/sec, in accordance with engineering practice, while the scale is chosen to fill the available space. The periods extend to 15 seconds, close to the long period cut-off point in the corrected data of Volume II, but the axis is divided into two linear portions. From 0 to 3 seconds takes up three-quarters of the axis, and from 3 to 15 seconds takes up the remainder. The subroutine used for this plotting is RVPLOT.

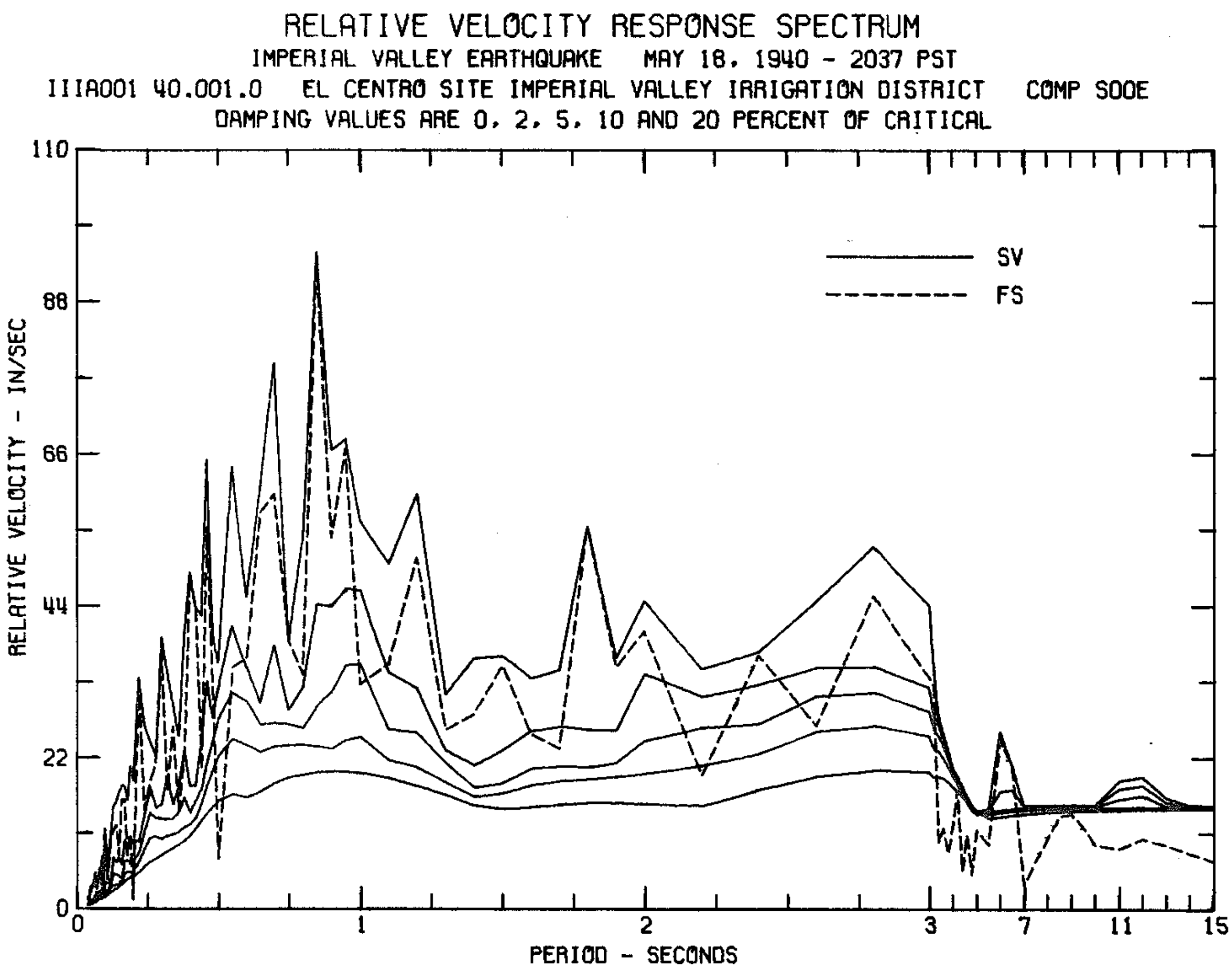

RELATIVE VELOCITY RESPONSE SPECTRUM
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
IIIA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE
DAMPING VALUES ARE 0, 2, 5, 10 AND 20 PERCENT OF CRITICAL
SV
FS
RELATIVE VELOCITY - IN/SEC
0
22
44
66
88
110
0
1
2
3
7
11
15
PERIOD - SECONDS

Figure 6

The second plot is that of the pseudo velocity response spectrum, PSV, together with the relative displacement spectrum, SD, and the pseudo acceleration spectrum, PSA, in the tripartite logarithmic plot versus period (Figure 7). This convenient plot is made possible by the relationships between PSV, SD, and PSA [see Equations (14) and (15) in Report No. EERL 72-80, Hudson, et al, 1972a]. The units used are, once again, the normal engineering units of in/sec, in, and g, respectively. The descriptive titles are the same as on the true relative velocity spectrum (Figure 6). To save computer time only the five response spectrum curves and three title lines are plotted using subroutines THLN and XYPLOT. The main title, the axes labels and the grid for the tripartite plot are reproduced by placing a transparent overlay, having two reference crosses for accurate positioning and the overlay number, over the computer plot. The overlay number and the reference crosses are, of course, located outside the final 8-1/2 x 11 inches print of the response spectrum. The overlay number is selected by the subroutine SPCTRA as follows:

for the overlay no.	SPCTRA selects the vertical axis range (in/sec)
1	1 - 4,000
2	0.1 - 400
3	0.01 - 40
4	0.001 - 4

The two pages of tables (Figures 8a, b) contain values of the ordinates for the previous plots. After the titles, there are arranged

RESPONSE SPECTRUM

IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST

IIIA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E

DAMPING VALUES ARE 0, 2, 5, 10 AND 20 PERCENT OF CRITICAL

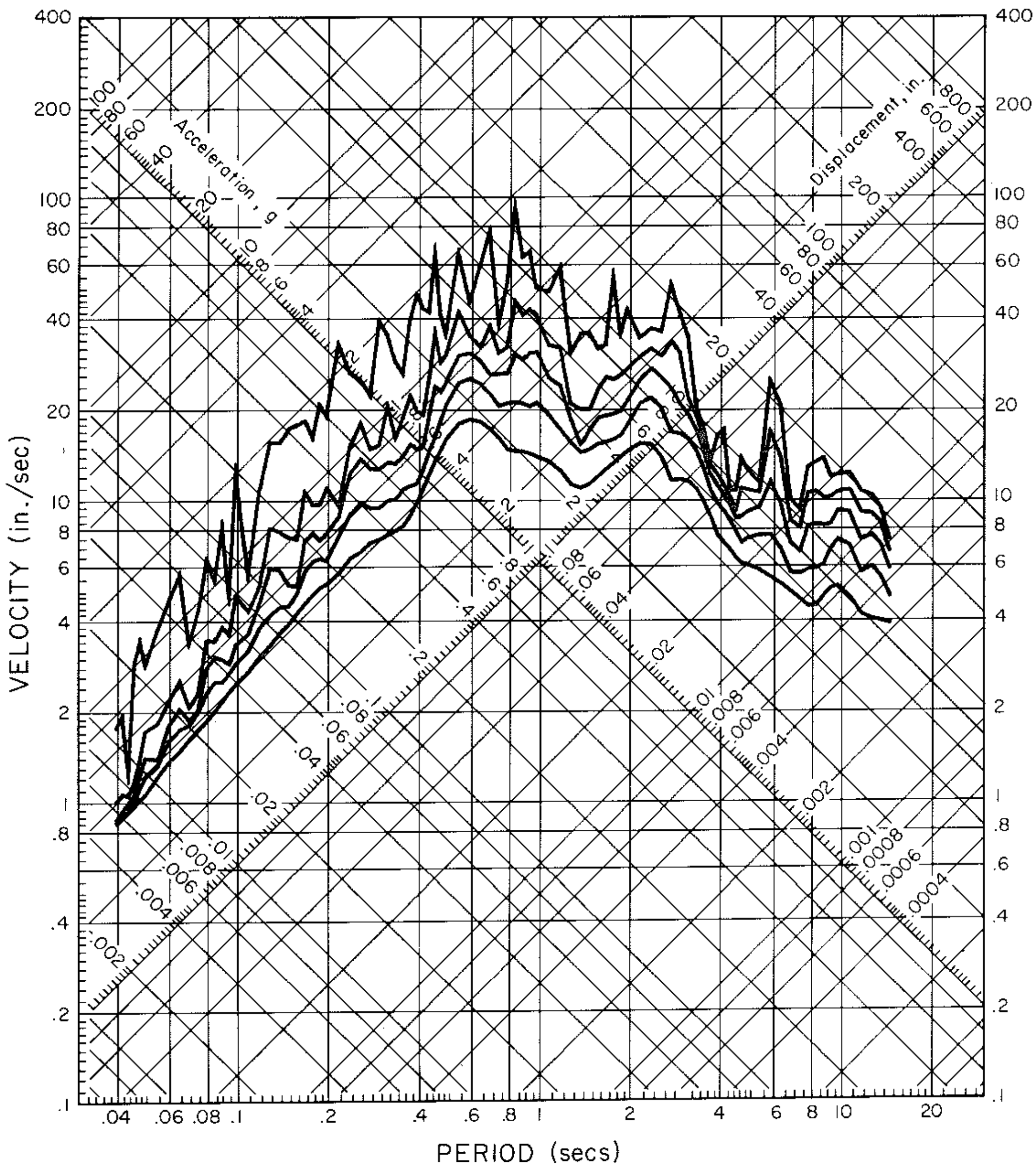


Figure 7

IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST

IIIA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE

THE UNITS USED ARE SEC FOR PER, IN/SEC FOR FS, SV AND PSV, IN FOR SD AND G FOR SA

		DAMPING = 0 PERCENT				DAMPING = 2 PERCENT				DAMPING = 5 PERCENT				DAMPING = 10 PERCENT				DAMPING = 20 PERCENT			
PER	FS	SD	SV	SA	PSV	SD	SV	SA	PSV	SD	SV	SA	PSV	SD	SV	SA	PSV	SD	SV	SA	PSV
0.040	155-2	114-4	162-2	731-3	179-2	647-5	666-3	416-3	101-2	566-5	550-3	363-3	890-3	552-5	502-3	356-3	868-3	547-5	420-3	351-3	859-3
0.042	131-2	131-4	165-2	765-3	197-2	727-5	752-3	421-3	108-2	635-5	617-3	374-3	957-3	610-5	554-3	360-3	913-3	601-5	457-3	350-3	900-3
0.044	885-3	856-5	959-3	473-3	128-2	747-5	797-3	393-3	106-2	708-5	713-3	371-3	101-2	670-5	558-3	361-3	957-3	659-5	490-3	352-3	942-3
0.046	219-2	217-4	262-2	105-2	297-2	883-5	920-3	427-3	120-2	793-5	772-3	384-3	108-2	753-5	638-3	371-3	102-2	720-5	520-3	358-3	983-3
0.048	293-2	270-4	329-2	119-2	353-2	107-4	555-3	479-3	140-2	954-5	767-3	424-3	124-2	864-5	663-3	391-3	113-2	794-5	557-3	363-3	103-2
0.050	241-2	232-4	258-2	951-3	292-2	139-4	116-2	571-3	175-2	113-4	826-3	467-3	142-2	982-5	668-3	407-3	123-2	860-5	585-3	365-3	108-2
0.055	309-2	338-4	339-2	114-2	386-2	163-4	128-2	557-3	187-2	123-4	879-3	416-3	140-2	117-4	763-3	406-3	134-2	108-4	684-3	382-3	123-2
0.060	314-2	459-4	409-2	130-2	481-2	212-4	155-2	603-3	222-2	175-4	114-2	507-3	184-2	156-4	891-3	441-3	163-2	132-4	774-3	389-3	139-2
0.065	453-2	607-4	531-2	147-2	587-2	268-4	193-2	649-3	259-2	216-4	133-2	523-3	208-2	184-4	104-2	459-3	178-2	155-4	882-3	357-3	150-2
0.070	140-2	385-4	282-2	803-3	345-2	235-4	172-2	493-3	211-2	211-4	128-2	446-3	189-2	205-4	110-2	424-3	184-2	183-4	979-3	358-3	164-2
0.075	340-2	529-4	431-2	962-3	443-2	282-4	179-2	514-3	236-2	248-4	142-2	458-3	268-2	243-4	119-2	442-3	204-2	212-4	109-2	395-3	177-2
0.080	552-2	830-4	602-2	132-2	652-2	446-4	258-2	712-3	350-2	361-4	187-2	579-3	284-2	299-4	149-2	488-3	235-2	242-4	125-2	407-3	190-2
0.085	456-2	757-4	525-2	107-2	559-2	474-4	256-2	668-3	350-2	420-4	223-2	591-3	310-2	346-4	178-2	490-3	255-2	276-4	139-2	408-3	204-2
0.090	243-2	120-3	776-2	151-2	839-2	559-4	305-2	707-3	390-2	435-4	217-2	545-3	304-2	365-4	180-2	472-3	254-2	315-4	150-2	418-3	220-2
0.095	244-2	762-4	430-2	864-3	504-2	552-4	268-2	623-3	365-2	445-4	219-2	508-3	294-2	414-4	187-2	474-3	274-2	358-4	158-2	425-3	237-2
0.100	116-1	202-3	116-1	207-2	127-1	782-4	392-2	805-3	491-2	544-4	250-2	567-3	342-2	469-4	188-2	480-3	294-2	400-4	169-2	419-3	251-2
0.110	422-2	102-3	522-2	866-3	585-2	768-4	367-2	653-3	438-2	648-4	257-2	556-3	370-2	572-4	224-2	501-3	327-2	485-4	198-2	437-3	277-2
0.120	100-1	211-3	108-1	149-2	110-1	101-3	409-2	717-3	532-2	856-4	366-2	632-3	465-2	740-4	305-2	526-3	387-2	596-4	231-2	443-3	312-2
0.130	113-1	329-3	149-1	199-2	159-1	170-3	742-2	103-2	823-2	125-3	519-2	772-3	607-2	682-4	342-2	541-3	426-2	701-4	264-2	445-3	339-2
0.140	121-1	356-3	157-1	185-2	159-1	179-3	689-2	931-3	805-2	134-3	493-2	705-3	601-2	101-3	345-2	530-3	454-2	830-4	293-2	454-3	372-2
0.150	770-2	422-3	171-1	192-2	176-1	184-3	737-2	837-3	772-2	127-3	481-2	579-3	536-2	109-3	346-2	504-3	458-2	952-4	320-2	454-3	399-2
0.160	158-1	465-3	180-1	186-2	182-1	194-3	655-2	775-3	762-2	135-3	447-2	538-3	532-2	129-3	423-2	528-3	509-2	110-3	351-2	463-3	434-2
0.170	159-1	511-3	178-1	181-2	189-1	295-3	100-1	104-2	109-1	203-3	691-2	715-3	752-2	162-3	503-2	579-3	599-2	126-3	394-2	473-3	469-2
0.180	772-2	472-3	158-1	149-2	164-1	282-3	890-2	890-3	987-2	229-3	702-2	727-3	800-2	184-3	540-2	597-3	644-2	142-3	431-2	474-3	498-2
0.190	108-1	643-3	209-1	182-2	212-1	300-3	932-2	852-3	995-2	227-3	661-2	652-3	751-2	199-3	541-2	569-3	658-2	158-3	458-2	476-3	524-2
0.200	124-2	619-3	187-1	158-2	194-1	357-3	597-2	914-3	112-1	253-3	690-2	644-3	797-2	205-3	536-2	542-3	645-2	172-3	462-2	463-3	541-2
0.220	325-1	118-2	337-1	250-2	337-1	342-3	978-2	728-3	977-2	317-3	817-2	667-3	905-2	272-3	688-2	576-3	778-2	205-3	530-2	471-3	587-2
0.240	143-1	103-2	266-1	183-2	270-1	595-3	139-1	105-2	155-1	478-3	113-1	861-3	125-1	351-3	875-2	654-3	919-2	251-3	614-2	473-3	659-2
0.260	184-1	106-2	242-1	160-2	256-1	766-3	176-1	115-2	185-1	582-3	140-1	902-3	140-1	414-3	103-1	653-3	100-1	286-3	682-2	469-3	691-2
0.280	211-1	101-2	221-1	131-2	226-1	676-3	146-1	682-3	151-1	577-3	132-1	746-3	129-1	428-3	105-1	578-3	961-2	330-3	723-2	463-3	741-2
0.300	374-1	191-2	395-1	217-2	400-1	747-3	152-1	847-3	156-1	622-3	130-1	705-3	130-1	462-3	101-1	540-3	968-2	363-3	780-2	442-3	760-2
0.320	171-1	182-2	343-1	182-2	358-1	107-2	194-1	107-2	210-1	656-3	131-1	703-3	136-1	518-3	106-1	527-3	101-1	400-3	835-2	432-3	786-2
0.340	265-1	158-2	296-1	140-2	293-1	892-3	151-1	789-3	165-1	729-3	131-1	641-3	134-1	556-3	109-1	500-3	102-1	439-3	884-2	422-3	812-2
0.360	140-1	153-2	251-1	121-2	268-1	111-2	179-1	877-3	193-1	826-3	140-1	655-3	144-1	637-3	111-1	511-3	111-1	481-3	932-2	408-3	840-2
0.380	224-1	243-2	402-1	172-2	403-1	137-2	231-1	972-3	227-1	949-3	162-1	678-3	156-1	689-3	119-1	502-3	114-1	550-3	991-2	403-3	910-2
0.400	484-1	312-2	490-1	199-2	491-1	128-2	178-1	827-3	202-1	957-3	139-1	615-3	150-1	740-3	123-1	481-3	116-1	639-3	107-1	418-3	100-1
0.420	436-1	293-2	437-1	169-2	438-1	131-2	180-1	765-3	196-1	102-2	158-1	601-3	153-1	870-3	132-1	512-3	130-1	745-3	117-1	446-3	112-1
0.440	202-1	299-2	426-1	158-2	428-1	183-2	245-1	966-3	261-1	137-2	175-1	731-3	196-1	104-2	155-1	558-3	149-1	877-3	129-1	478-3	125-1
0.460	553-1	491-2	652-1	237-2	671-1	269-2	331-1	130-2	368-1	179-2	214-1	865-3	244-1	121-2	180-1	595-3	166-1	101-2	140-1	510-3	138-1
0.480	320-1	318-2	399-1	141-2	416-1	224-2	278-1	996-3	293-1	177-2	245-1	797-3	232-1	144-2	203-1	651-3	189-1	115-2	149-1	537-3	150-1
0.500	725-2	288-2	357-1	117-2	362-1	248-2	319-1	102-2	312-1	201-2	276-1	836-3	253-1	169-2	222-1	699-3	212-1	130-2	157-1	559-3	163-1
0.550	351-1	588-2	643-1	199-2	672-1	375-2	412-1	126-2	428-1	268-2	315-1	917-3	307-1	219-2	247-1	759-3	250-1	160-2	167-1	588-3	183-1
0.600	365-1	441-2	453-1	125-2	461-1	342-2	352-1	971-3	358-1	300-2	302-1	859-3	314-1	247-2	238-1	722-3	259-1	181-2	162-1	570-3	189-1
0.650	575-1	628-2	614-1	152-2	607-1	339-2	300-1	824-3	328-1	305-2	268-1	738-3	294-1	256-2	228-1	637-3	248-1	192-2	171-1	521-3	185-1
0.700	603-1	884-2	792-1	184-2	794-1	430-2	384-1	900-3	386-1	297-2	271-1	622-3	266-1	255-2	236-1	546-3	229-1	195-2	183-1	459-3	175-1

Figure 8a

IIIA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT CCMP S00E

THE UNITS USED ARE SEC FOR PER, IN/SEC FOR FS, SV AND PSV, IN FOR SD AND G FOR SA

		CAMPING = 0 PERCENT				CAMPING = 2 PERCENT				CAMPING = 5 PERCENT				CAMPING = 10 PERCENT				CAMPING = 20 PERCENT			
PER	FS	SC	SV	SA	PSV	SD	SV	SA	PSV	SD	SV	SA	PSV	SD	SV	SA	PSV	SD	SV	SA	PSV
0.75	389-1	467-2	389-1	849-3	391-1	375-2	289-1	682-3	314-1	320-2	269-1	584-3	268-1	249-2	238-1	466-3	209-1	194-2	192-1	401-3	163-1
0.80	341-1	681-2	540-1	108-2	535-1	420-2	324-1	670-3	329-1	343-2	263-1	549-3	269-1	273-2	239-1	444-3	214-1	191-2	196-1	347-3	150-1
0.85	940-1	129-1	953-1	182-2	954-1	625-2	443-1	887-3	462-1	420-2	297-1	598-3	310-1	291-2	236-1	421-3	215-1	200-2	199-1	304-3	147-1
0.90	538-1	932-2	665-1	117-2	650-1	597-2	439-1	755-3	416-1	424-2	316-1	539-3	296-1	305-2	233-1	393-3	213-1	211-2	200-1	289-3	147-1
0.95	669-1	102-1	682-1	116-2	680-1	661-2	466-1	751-3	437-1	476-2	355-1	542-3	315-1	315-2	246-1	363-3	208-1	219-2	200-1	270-3	145-1
1.00	327-1	811-2	563-1	830-3	509-1	661-2	463-1	677-3	415-1	503-2	357-1	518-3	316-1	342-2	251-1	359-3	215-1	226-2	198-1	249-3	142-1
1.10	356-1	872-2	502-1	737-3	498-1	582-2	343-1	493-3	332-1	453-2	261-1	387-3	259-1	339-2	216-1	292-3	193-1	239-2	190-1	214-3	136-1
1.20	511-1	115-1	603-1	818-3	603-1	620-2	320-1	441-3	324-1	464-2	256-1	331-3	243-1	332-2	206-1	241-3	173-1	243-2	179-1	186-3	127-1
1.30	261-1	648-2	312-1	392-3	313-1	437-2	231-1	265-3	211-1	390-2	213-1	237-3	188-1	325-2	184-1	200-3	157-1	237-2	166-1	157-3	114-1
1.40	283-1	806-2	365-1	420-3	362-1	454-2	209-1	237-3	203-1	346-2	177-1	181-3	155-1	325-2	163-1	173-3	146-1	248-2	150-1	138-3	111-1
1.50	354-1	867-2	368-1	394-3	363-1	484-2	231-1	220-3	202-1	417-2	184-1	190-3	174-1	359-2	169-1	167-3	150-1	275-2	146-1	136-3	115-1
1.60	254-1	820-2	335-1	327-3	322-1	607-2	259-1	243-3	238-1	486-2	205-1	195-3	191-1	398-2	180-1	162-3	156-1	311-2	148-1	136-3	122-1
1.70	234-1	905-2	348-1	320-3	334-1	703-2	266-1	249-3	260-1	526-2	206-1	187-3	194-1	429-2	187-1	154-3	158-1	348-2	152-1	136-3	128-1
1.80	556-1	159-1	556-1	503-3	556-1	730-2	261-1	230-3	254-1	564-2	206-1	179-3	196-1	463-2	189-1	149-3	161-1	387-2	155-1	136-3	135-1
1.90	351-1	109-1	365-1	309-3	361-1	801-2	260-1	227-3	265-1	608-2	214-1	173-3	201-1	505-2	193-1	147-3	167-1	428-2	155-1	135-3	141-1
2.00	404-1	138-1	448-1	353-3	434-1	884-2	342-1	226-3	277-1	695-2	246-1	178-3	218-1	579-2	197-1	152-3	182-1	471-2	153-1	135-3	148-1
2.20	194-1	122-1	349-1	259-3	350-1	106-1	309-1	224-3	303-1	891-2	264-1	189-3	254-1	744-2	209-1	161-3	212-1	550-2	151-1	130-3	157-1
2.40	369-1	143-1	375-1	255-3	376-1	123-1	327-1	219-3	322-1	106-1	270-1	190-3	277-1	849-2	227-1	156-3	222-1	593-2	174-1	119-3	155-1
2.60	266-1	151-1	448-1	228-3	365-1	125-1	351-1	190-3	303-1	106-1	310-1	162-3	258-1	841-2	259-1	132-3	203-1	581-2	194-1	103-3	140-1
2.80	455-1	234-1	527-1	305-3	525-1	152-1	353-1	198-3	341-1	106-1	315-1	140-3	238-1	759-2	267-1	101-3	170-1	526-2	202-1	874-4	118-1
3.00	337-1	202-1	441-1	229-3	423-1	148-1	322-1	168-3	310-1	100-1	287-1	115-3	210-1	802-2	251-1	951-4	168-1	568-2	199-1	754-4	118-1
3.20	326-1	178-1	351-1	178-3	350-1	111-1	281-1	111-3	218-1	987-2	261-1	999-4	193-1	820-2	233-1	856-4	161-1	593-2	192-1	695-4	116-1
3.40	986-2	112-1	278-1	991-4	207-1	103-1	267-1	921-4	192-1	934-2	251-1	839-4	172-1	792-2	229-1	741-4	146-1	592-2	193-1	638-4	109-1
3.60	118-1	991-2	254-1	782-4	173-1	927-2	247-1	733-4	161-1	844-2	236-1	676-4	147-1	732-2	219-1	622-4	127-1	570-2	191-1	569-4	595-2
3.80	812-2	838-2	229-1	594-4	138-1	795-2	225-1	564-4	131-1	736-2	218-1	527-4	121-1	655-2	206-1	507-4	108-1	534-2	185-1	456-4	883-2
4.00	120-1	105-1	204-1	677-4	166-1	779-2	203-1	498-4	122-1	713-2	200-1	462-4	112-1	623-2	193-1	414-4	979-2	491-2	179-1	432-4	771-2
4.20	174-1	116-1	192-1	675-4	174-1	753-2	183-1	438-4	112-1	658-2	182-1	412-4	104-1	616-2	180-1	377-4	923-2	487-2	171-1	373-4	729-2
4.40	557-2	758-2	179-1	400-4	108-1	721-2	172-1	382-4	103-1	673-2	167-1	364-4	961-2	601-2	168-1	340-4	659-2	484-2	164-1	322-4	692-2
4.60	109-1	798-2	165-1	385-4	109-1	686-2	160-1	332-4	937-2	642-2	154-1	318-4	877-2	579-2	157-1	304-4	791-2	476-2	157-1	307-4	650-2
4.80	510-2	106-1	151-1	472-4	139-1	843-2	149-1	374-4	110-1	682-2	144-1	304-4	893-2	557-2	147-1	274-4	730-2	478-2	150-1	294-4	626-2
5.00	115-1	101-1	143-1	415-4	127-1	866-2	141-1	355-4	108-1	735-2	138-1	303-4	924-2	603-2	139-1	259-4	758-2	491-2	144-1	281-4	617-2
5.50	948-2	100-1	146-1	339-4	114-1	936-2	144-1	317-4	107-1	838-2	142-1	286-4	957-2	677-2	138-1	240-4	774-2	518-2	132-1	253-4	592-2
6.00	246-1	240-1	259-1	682-4	251-1	162-1	172-1	462-4	170-1	111-1	144-1	317-4	116-1	741-2	141-1	218-4	776-2	537-2	134-1	228-4	562-2
6.50	208-1	215-1	208-1	522-4	208-1	144-1	174-1	349-4	139-1	100-1	146-1	244-4	968-2	714-2	143-1	176-4	691-2	550-2	137-1	207-4	532-2
7.00	349-2	112-1	151-1	234-4	100-1	953-2	149-1	199-4	856-2	808-2	147-1	171-4	725-2	647-2	144-1	154-4	581-2	560-2	139-1	188-4	503-2
7.50	720-2	112-1	151-1	204-4	940-2	976-2	150-1	178-4	818-2	815-2	148-1	150-4	683-2	689-2	145-1	138-4	577-2	567-2	140-1	172-4	475-2
8.00	103-1	165-1	151-1	264-4	130-1	135-1	150-1	216-4	106-1	105-1	149-1	169-4	824-2	760-2	146-1	124-4	597-2	573-2	141-1	158-4	450-2
8.50	135-1	183-1	152-1	259-4	135-1	145-1	151-1	206-4	107-1	113-1	149-1	161-4	837-2	812-2	147-1	119-4	600-2	612-2	142-1	146-4	453-2
9.00	139-1	199-1	152-1	252-4	139-1	147-1	151-1	185-4	102-1	118-1	149-1	151-4	829-2	894-2	147-1	123-4	624-2	702-2	143-1	136-4	490-2
9.50	120-1	182-1	152-1	206-4	120-1	157-1	151-1	178-4	103-1	128-1	150-1	146-4	846-2	105-1	148-1	124-4	701-2	781-2	144-1	127-4	516-2
10.00	934-2	195-1	152-1	200-4	122-1	171-1	151-1	175-4	107-1	148-1	150-1	153-4	930-2	119-1	148-1	127-4	748-2	837-2	144-1	120-4	526-2
11.00	883-2	216-1	188-1	183-4	123-1	191-1	175-1	162-4	109-1	161-1	160-1	137-4	919-2	124-1	148-1	112-4	713-2	843-2	145-1	107-4	481-2
12.00	102-1	205-1	193-1	146-4	107-1	175-1	181-1	125-4	920-2	142-1	165-1	102-4	747-2	111-1	148-1	834-5	584-2	806-2	146-1	965-5	422-2
13.00	935-2	216-1	162-1	131-4	104-1	188-1	156-1	114-4	911-2	160-1	150-1	994-5	777-2	126-1	149-1	816-5	610-2	843-2	146-1	676-5	407-2
14.00	816-2	209-1	151-1	109-4	939-2	186-1	151-1	980-5	836-2	158-1	150-1	844-5	712-2	124-1	149-1	707-5	558-2	893-2	146-1	806-5	400-2
15.00	703-2	177-1	151-1	807-5	744-2	161-1	151-1	741-5	678-2	141-1	150-1	671-5	593-2	115-1	149-1	575-5	482-2	938-2	147-1	744-5	393-2

Figure 8b

in columns, the periods (PER), Fourier amplitude spectrum (FS), and then sets of four columns containing SD, SV, SA, and PSV for all of the five damping values (SD, SV, SA and PSV are defined in the Volume III, Part A, Report No. 72-80). The units are indicated and each value is followed by a multiplicative power of 10, e.g., 731-1 is 73.1.

The ordinates are also stored on computer cards containing the Fourier amplitude spectrum and, for all of the five damping values, the pseudo velocity and relative velocity response spectra. A simplified economic format, 13F6.3, is made possible by using the logarithms of these spectra. The last two columns are for identification purposes. These cards are available on request from the National Information Service for Earthquake Engineering, at the California Institute of Technology.

The spectral amplitudes are stored on the magnetic tapes as follows:

<u>Volume III Tape</u>
(one file per one acceleration component)

Each file has:

1. Heading data of alphanumeric type
2. Heading data of integer type
3. Heading data of floating point type
4. Response values for 91 periods and five dampings (acceleration, velocity, displacement, and pseudo velocity response amplitudes)
5. Times of maximum responses above*

Tape parameters: 1600 bpi, LRECL=1204, BLKSIZE=3616, RECFM=VBS.

*These times are available only on the most recent Volume III tapes. Old version contains zeroes in these locations.

Volume V Tape
(one file per one acceleration component)

Each file has:

1. Heading data of alphanumeric type
2. Heading data of integer type
3. Heading data of floating point type
4. Equally spaced corrected accelerograms from Volume II tape
5. Response envelope spectrum ordinates
6. EOF

Tape parameters: 1600 bpi, LRECL=1204, BLKSIZE=3616, RECFM=VBS. The detailed description and a sample of the heading data set are given in the following section.

VOLUME III HEADING DATA

Punched Output Card No.	Heading Data Array CORTIL(I), I=I1, I2 I1, I2	Description
1	1 - 20	Volume III main title
2 - 26	21 - 521	Same as CORTIL(1)-CORTIL(500) of cards no. 1-25 in Vol. II heading data
27 - 28	521 - 560	Volume III earthquake title of the file
29 - 30	561 - 600	Output and input units of response & Fourier spectra
31 - 35	ICOR(I), I=1, 100	Same as ICOR in Vol. II heading data
36 - 45	FCOR(I), I=1, 100	Same as FCOR in Vol. II heading data
46	DMP(I), I=1, 5	Five damping values, fraction of critical damping
47 - 56	PD(I), I=1, 100	Period } only the first 91 entries are used, the rest being 0.0
57 - 66	FS(I), I=1, 100	Fourier spectra } only the first 91 entries are used, the rest being 0.0

```
RESPONSE AND FOUR. AMPL. SPECTRA (ZETA=0) FOR 91 PERIODS CORRESPONDING TO THE
CORRECTED ACCELEROGRAM  IIA001  40.001.0    COMP S00E    FILE  1 CORRESPONDING TO
FILE   1  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-A, EERL 70-20
IMPERIAL VALLEY EARTHQUAKE
MAY 18, 1940 - 2037 PST
IA001    40.001.0 S                                                            18
STATION NO. 117    32 47 43N,115 32 55W                                        38
EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT                             50
COMP S00E                                                                       9
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST                           52
EPICENTER  32 44 00N,115 27 00W                                                31
INSTR PERIOD = 0.0990 SEC  DAMPING = 0.552                                     42
NO. OF POINTS = 985    DURATION = 53.73 SEC                                    42
UNITS ARE SEC AND G/10.                                                        23
RMS ACCLN OF COMPLETE RECORD = 0.4876 G/10.                                    43
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  2688  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF    0.02   SEC.
PEAK ACCELERATION =    341.69531  CMS/SEC/SEC AT    2.1200 SEC
PEAK VELOCITY     =     33.44914  CMS/SEC      AT    2.1800 SEC
PEAK DISPLACEMENT =     10.86678  CMS          AT    8.5800 SEC
INITIAL VELOCITY  =     -4.66421  CMS/SEC         INITIAL DISP. =      2.15852 CMS
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST

IIA001  40.001.0   EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT   COMP S00
E
IIIA001  40.001.0   EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT   COMP S0
0E
UNITS FOR RESPONSE AND FOUR. AMPL. SPECTRA HERE ARE INCHES AND SEC.
NOTE.  INPUT IS IN CM/SEC/SEC.   OUTPUT IS  IN INCHES AND SEC UNITS.
   1   1   1   1  40   1   0   4 117  32  47  43-115  32  55  32  44   0-115  27
   0   5  1819402037   0 180 985  26  50   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0 985 9872688   2  10  10   1   0  52  81
  10  10   21344   5 538  82   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0.099    0.552   53.730    0.488    0.100    0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   1.000    0.010    0.010   53.732   63.466    1.000    1.000   27.000    2.000   53.740
   0.020    0.070    0.020    0.0      2.120  341.695    2.180   33.449    8.580   10.867
  -4.664    0.070   25.000    0.200    0.200    2.159    0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   0.0      0.020    0.050    0.100    0.200
   0.040    0.042    0.044    0.046    0.048    0.050    0.055    0.060    0.065    0.070
   0.075    0.080    0.085    0.090    0.095    0.100    0.110    0.120    0.130    0.140
   0.150    0.160    0.170    0.180    0.190    0.200    0.220    0.240    0.260    0.280
   0.300    0.320    0.340    0.360    0.380    0.400    0.420    0.440    0.460    0.480
   0.500    0.550    0.600    0.650    0.700    0.750    0.800    0.850    0.900    0.950
   1.000    1.100    1.200    1.300    1.400    1.500    1.600    1.700    1.800    1.900
   2.000    2.200    2.400    2.600    2.800    3.000    3.200    3.400    3.600    3.800
   4.000    4.200    4.400    4.600    4.800    5.000    5.500    6.000    6.500    7.000
   7.500    8.000    8.500    9.000    9.500   10.000   11.000   12.000   13.000   14.000
  15.000    0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   1.560    1.319    0.887    2.200    2.935    2.412    3.098    3.145    4.534    1.403
   3.405    5.528    4.572    2.441    2.446   11.680    4.230   10.097   11.319   12.197
   7.713   15.882   15.971    7.732   10.901    1.246   32.538   14.336   18.485   21.160
  37.449   17.187   26.588   14.088   22.424   48.470   43.660   20.211   55.413   32.015
   7.259   35.158   36.536   57.566   60.382   38.968   34.181   94.070   53.891   66.910
```

32.734	35.665	51.163	26.137	28.403	35.474	25.477	23.427	55.684	35.178
40.508	19.499	36.985	26.684	45.574	33.750	32.659	9.871	11.812	8.131
12.092	17.425	5.573	10.906	5.110	11.545	9.483	24.624	20.855	3.497
7.203	10.329	13.544	13.946	12.012	9.351	8.835	10.268	9.361	8.168
7.037	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

PROGRAMS FOR PROCESSING VOLUME III DATA

Volume III MAIN Program (Trifunac, Vijayaraghavan)

Volume III MAIN computes response spectra from Volume II accelerograms for 5 values of damping. It plots the response spectra of velocity on a log-log scale and on a linear scale. It prints out results and writes them on tape. Also, it writes a second tape containing the relative oscillator displacements for use in plotting the Response Envelope Spectra (RES).

Usage

The program reads in the following data:

LFILE, MFILE, JFILE, NREAD, NWRITE

Where

LFILE = first file number of the Vol. II tape to be read in

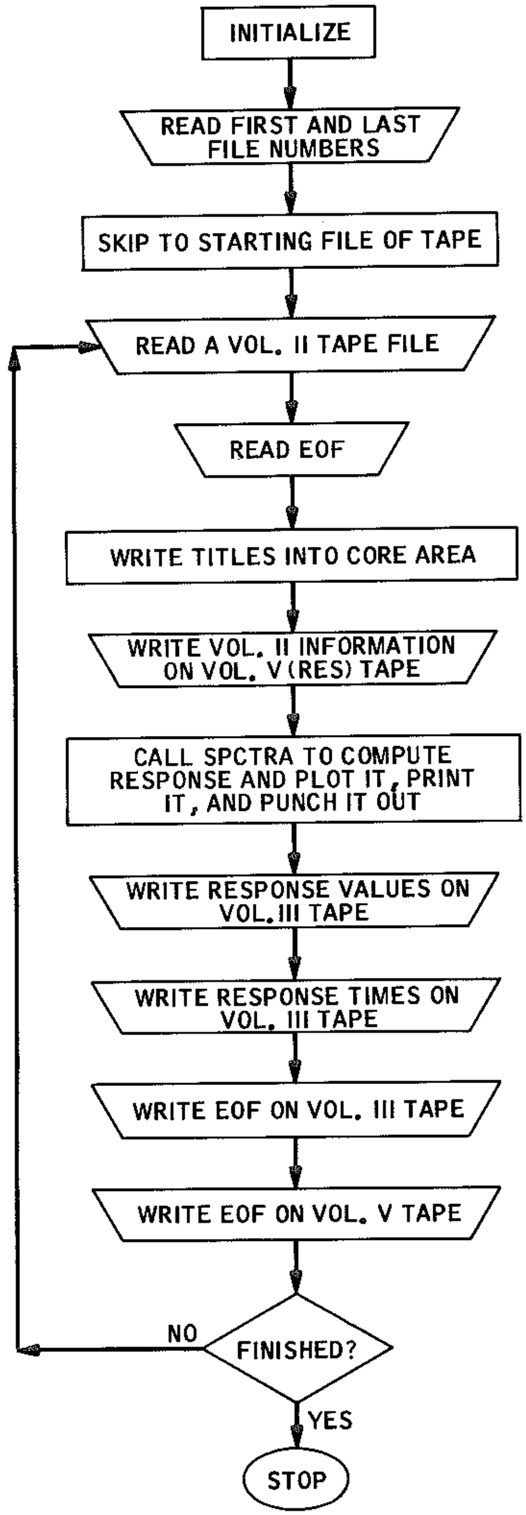
MFILE = last file number of the Vol. II tape to be read in

JFILE = 0

NREAD = unit number for the Vol. II tape to be read in

NWRITE = unit number for the Vol. III tape to be written onto.

VOL. III MAIN FLOW CHART



```
      DIMENSION CORESP(600),ICOR(100),FCOR(100)
      DIMENSION CORTIL(500)
      DIMENSION TITLE1(33),TITLE2(33)
      COMMON DMP(5),SD(5,100),SA(5,100),GA(10000),PSSV(5,100)
      COMMON/BLCKT/TTSD(5,100),TTSV(5,100),TTSA(5,100)
      COMMON /BT2/PD(100),SV(5,100),FS(100)
      EQUIVALENCE (ICOR(67),NT2),(ICOR(68),NPERID)
      EQUIVALENCE (CORESP(21),CORTIL(1))
      EQUIVALENCE (CORESP(521),TITLE2(1))
      EQUIVALENCE (CORTIL(421),TITLE1(1))
      DATA CORESP/'RESP','ONSE',' AND',' FOU','R. A','MPL.',' SPE',
     *     'CTRA',' (ZE','TA=0',') FO','R 91',' PER','IODS',' COR',
     *     'RESP','ONDI','NG T','O TH','E   ',255*'    ',255*'    ',
     *     30*'    ',
     *     'UNIT','S FO','R RE','SPON','SE A','ND F','OUR.',' AMP',
     *     'L. S','PECT','RA H','ERE ','ARE ','INCH','ES A','ND S',
     *     'EC. ',3*'    ',
     *     'NOTE',':  I','NPUT',' IS ','IN C','M/SE','C/SE','C.  ',
     *     ' OUT','PUT ','IS I','N IN','CHES',' AND',' SEC',' UNI',
     *     'TS. ',3*'    '/
      NWR2=12
      CALL ERRSET (218,1,0,0)
      CALL ERRSET (213,1,0,0)
      ICOR(68)=91
      DO 1000 I=1,5
      DO 1000 J=92,100
      SD(I,J)=0.0
      SV(I,J)=0.0
      SA(I,J)=0.0
      PSSV(I,J)=0.0
      PD(J)=0.0
      FS(J)=0.0
 1000 CONTINUE
      READ(5,1)LFILE,MFILE,JFILE,NREAD,NWRITE
    1 FORMAT(6X,5I2)
      SF=1.0/2.537
      DEL=0.02
      IF(1-LFILE)10,20,20
   10 KSKFL=LFILE-2
   11 READ(NREAD,END=14)
      GO TO 11
   14 CALL READNF(NREAD,KSKFL)
  543 READ (NWRITE,END=544)
      GO TO 543
  544 CALL READNF (NWRITE,KSKFL)
  545 READ (NWR2,END=546)
      GO TO 545
  546 CALL READNF (NWR2,KSKFL)
   20 CONTINUE
      JST=JFILE
      DO 100 NFILE=LFILE,MFILE
      JST=JST+1
      IF (JST.NE.1) CALL WRTNF(NWR2)
      IF(JST.NE.1)CALL WRTNF(NWRITE)
      READ(NREAD)CORTIL,ICOR,FCOR
      NDATA=ICOR(53)
      NLINE1=ICOR(59)
      NT1=NLINE1
      NLINE2=ICOR(60)
      NT2=NLINE2+1
C     FOR NT2 FOR VOL III HAS AN ADDITIONAL I OVER II IN VOL II
```

```
      CALL OUTCOR(CORESP(521),NBYT)
      WRITE (6,30) (CORTIL(K),K=461,493)
   30 FORMAT (1HI,33A4)
      CALL OUTCOR
      READ(NREAD)(GA(K),K=1,NDATA)
   32 READ(NREAD,END=36)
      GO TO 32
   36 CALL READNF(NREAD)
      WRITE (NWR2) CORESP,ICOR,FCOR
      CALL WRTR (NDATA,GA)
      WRITE (6,966)
  966 FORMAT (1HI)
      CALL SPCTRA(NDATA,TITLE1,NT1,TITLE2,NT2,SF,DEL)
      WRITE(NWRITE)CORESP,ICOR,FCOR,DMP,PD,FS
      WRITE (NWRITE) ((SD(L,K),K=1,100),(SV(L,K),K=1,100),
     *     (SA(L,K),K=1,100),(PSSV(L,K),K=1,100),L=1,5)
      WRITE (NWRITE) ((TTSD(L,K),K=1,100),(TTSV(L,K),K=1,100),
     *     (TTSA(L,K),K=1,100),L=1,5)
  100 CONTINUE
      CALL ENDMF(NWR2)
      CALL ENDMF(NWRITE)
      STOP
      END
```

Subroutine SPCTRA (Trifunac)

SPCTRA is called by the Volume III MAIN program.

SPCTRA calls PCNO3 to compute response; then it plots, prints, and punches response values.

Usage

```
CALL SPCTRA (NDATA,TITLE1,NT1,TITLE2,NT2,SF,DEL)
COMMON DMP(5),SD(5,100),SA(5,100),GA(10000),PSSV(5,100)
COMMON /BT1/V(100),H(100)
COMMON /BT2/PD(100),SV(5,100),FS(100)
COMMON /BT3/RVRS(9),TITLE3(15)
```

Where

NDATA is no. of acceleration data points.

TITLE1 contains earthquake information.

NT1 is no. of characters in TITLE1.

TITLE2 contains accelerogram information.

NT2 is no. of characters in TITLE2.

SF is a scaling factor (cm to inches implies SF = 1.0/2.537).

DEL is the equi-spacing interval of acceleration.

DMP is the response damping values of the oscillators.

SD is the peak displacement.

SA is the peak acceleration.

GA is input acceleration.

PSSV is pseudo-velocity: (2 * PI/PD)*SD of response, where PD is the undamped natural period of an oscillator and PI = 3.14159.

V is working space for the subroutine THLN.

H is working space for the subroutine THLN.

PD is an array containing the periods of the response oscillators whose response is computed.

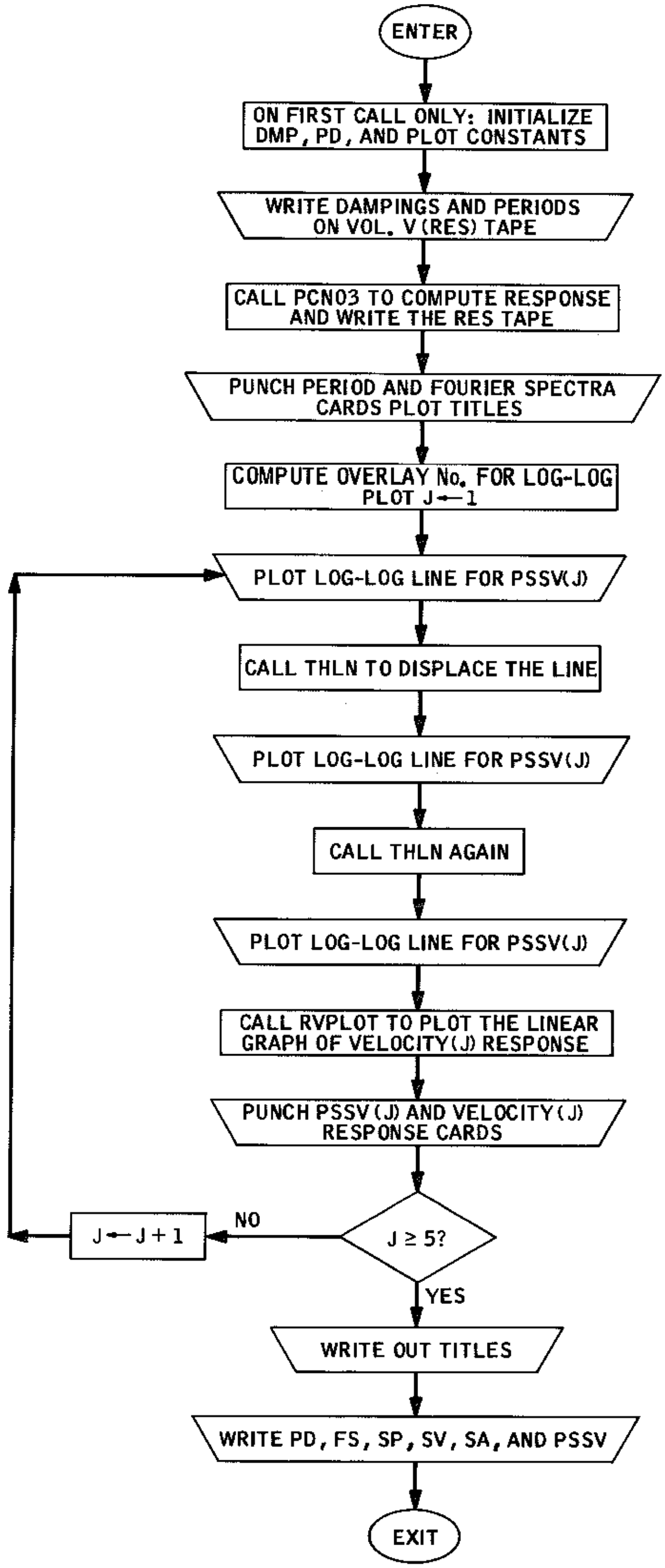
SV is the peak velocity response.

FS is Fourier spectra.

RVRS is a plotting title.

TITLE3 is a plotting title.

SPCTRA FLOW CHART



```
      SUBROUTINE SPCTRA(NDATA,TITLE1,NT1,TITLE2,NT2,SF,DEL)
C
C     THIS SUBROUTINE COMPUTES SA, SV, SD, (6.28/T)*SD AND FS SPECTRA
C     FOR 91 PERIODS BETWEEN 0.04 AND 15 SECONDS FOR THE INPUT ACCELERA-
C     TION WITH EQUALLY SPACED DATA POINTS
C     THE OUTPUT CONSISTS OF:
C        1.LOG-LOG PLOT OF (6.28/T)*SD SPECTRUM WITH TITLE1 AND TITLE2
C        2.PLOT OF SV AND SF VALUES VERSUS BROKEN LINEAR SCALE
C        3.PRINT OF TITLE1, TITLE2, ALL PERIODS, FS, SA, SV, SD AND
C          (6.28/T)*SD SPECTRA.
C        4.PUNCHED CARDS WITH PERIODS, LOGARITHMS OF FS AND REAL VALUES
C          OF SV AND (6.28/T)*SD SPECTRA FOR DAMPING VALUES 2, 5, 10 AND
C          20 PERCENT
C
C***********************************************************************
C
C     INPUT PARAMETERS:
C     NDATA=NO. OF DATA IN THE ACCELEROGRAM
C     TITLE1=TITLE CONTAINING EARTHQUAKE INFORMATION
C     NT1=NO. OF ALPHANUMERIC CHARACTERS IN TITLE1
C     TITLE2=TITLE CONTAINING ACCELEROGRAM INFORMATION
C     NT2=NO. OF ALPHANUMERIC CHARACTERS IN TITLE2
C     SF=SCALING FACTOR FOR INPUT DATA IN UNITS OF G
C     DEL=DT INTERVAL FOR INPUT DATA
C
      COMMON DMP(5),SD(5,100),SA(5,100),GA(10000),PSSV(5,100)
      COMMON /BT1/ V(100),H(100)
      COMMON  /BT2/ PD(100),SV(5,100),FS(100)
      COMMON /BT3/RVRS(9),TITLE3(15)
      DIMENSION TITLE1(33), TITLE2(33), EX(2),EY(2),DD(3),IRG(21),IE(21)
      DIMENSION TITLE4(1),TITLE5(1),ST(21),FMTTL(5),FMTSTN(5)
      DIMENSION IDMP(5)
      DATA IDMP/0,2,5,10,20/
      DATA TITLE4/' SV '/
      DATA TITLE5/' FS '/
C     8 IN FMTTL AND FMTSTN IS DUMMY
      DATA FMTTL /1H(,1H8,2HX,,1H8,3HA4)/
      DATA FMTSTN/1H(,1H8,2HX,,1H8,3HA4)/
      EQUIVALENCE (FSSMAX,ISSMAX),(DS,IDS),(IP,XIP)
      INTEGER SCALSW
      DATA NFIRST/1/
      IF (NFIRST.NE.1) GO TO 50
      IP=91
      ID=5
      DD(1)=0.
      DD(3)=0.
C     FIVE DAMPING VALUES 0.0 0.02 0.05 0.1 0.2
      DMP(1)=0.
      DMP(2)=0.02
      DMP(3)=0.05
      DMP(4)=0.10
      DMP(5)=0.20
C     GENERATING PERIODS FOR THE SPECTRA
      DO 1 I=1,6
      AI=I-1
    1 PD(I)=0.04 + 0.002*AI
      DO 2 I=7,16
      AI=I-6
    2 PD(I)=0.05 + 0.005*AI
      DO 3 I=17,26
      AI=I-16
```

```
    3 PD(I)=0.1 + 0.01*AI
      DO 4 I=27,41
      AI=I-26
    4 PD(I)=0.2 + AI*0.02
      DO 5 I=42,51
      AI=I-41
    5 PD(I)=0.5 + 0.05*AI
      DO 6 I=52,61
      AI=I-51
    6 PD(I)=1.0 + AI*0.1
      DO 7 I=62,76
      AI=I-61
    7 PD(I)=2.0 + AI*0.2
      DO 8 I=77,86
      AI=I-76
    8 PD(I)=5.0 + AI*0.5
      DO 9 I=87,91
      AI=I-86
    9 PD(I)=10.0 + AI
      EX(1)=0.32
      EX(2)=14.93125
      EY(1)=-0.16
      EY(2)=10.16
      X1=3.0
      X2=3.40
      X3=3.75
      X4=0.10125
      SZ1=0.15
      SZ2=0.1
      SZ3=0.172
      WDTH=6./7.
      NFIRST=0
   50 CONTINUE
      Y1=5.0-SZ1*NT1/2.0*WDTH
      Y2=5.0-SZ2*NT2/2.0*WDTH
      Y3=5.0-SZ2*28*WDTH
      Y4=5.0-SZ3*6*WDTH
      THETA=90.
C     COMPUTE SPECTRA
      NWR2=12
      WRITE (NWR2) ID
      CALL WRTR (ID,DMP)
      WRITE (NWR2) IP
      CALL WRTR (IP,PD)
      CALL PCN03 (IP,NDATA,DEL,SF)
      IF (NDATA.EQ.0) GO TO 51
      PUNCH 100, TITLE1
      PUNCH 100, TITLE2
  100 FORMAT(20A4)
      PUNCH 101
  101 FORMAT('PERIOD COORDINATES IN SECONDS')
      DO 103 L=1,7
      J2=13*L
      J1=J2-12
      PUNCH 102,  (PD(K),K=J1,J2),L
  103 CONTINUE
  102 FORMAT(13F6.3,'P',I1)
      PUNCH 104
  104 FORMAT('LOGARITHM OF FOURIER AMPLITUDE SPECTRA IN IN/SEC')
      DO 106 K=1,IP
      FSSS=FS(K)
      V(K)=ALOG10(FSSS)
```

```
  106 CONTINUE
      DO 105 L=1,7
      J2=13*L
      J1=J2-12
      PUNCH 122, (V(K),K=J1,J2),L
  105 CONTINUE
  122 FORMAT(13F6.3,'F',I1)
      PUNCH 108
  108 FORMAT('LOGARITHM OF 6.28*SD/T AND SV SPECTRA IN IN/SEC')
      CALL SYSSYM(X1,Y1,SZ1,TITLE1,NT1,THETA)
      CALL SYSSYM(X2,Y2,SZ2,TITLE2,NT2,THETA)
      CALL SYSSYM(X3,Y3,SZ2,TITLE3,57,THETA)
      FSSMIN=10.0E10
      FSSMAX=0
      DO 255 J=1,ID
      DO 26 I=1,IP
      SA(J,I)=SA(J,I)/386.0
      PSD=6.28318*SD(J,I)/PD(I)
      FSSMIN=AMIN1(FSSMIN,PSD)
      FSSMAX=AMAX1(FSSMAX,PSD)
      PSSV(J,I)=PSD
   26 CONTINUE
  255 CONTINUE
      SCALSW=4
      IF (FSSMAX.GE.4.0) SCALSW=3
      IF (FSSMAX.GE.40.0) SCALSW=2
      IF (FSSMAX.GE.400.0) SCALSW=1
      IF (FSSMAX.LE.4000.0) GO TO 1060
      WRITE (6,1061) FSSMAX
 1061 FORMAT (////1H0,'******   ERROR   ******',10X,'MAX. VALUE = ',
     A      E16.5)
      STOP
 1060 CONTINUE
      ISCSW=1
      IF (FSSMIN.LE.1.0) ISCSW=2
      IF (FSSMIN.LE.0.1) ISCSW=3
      IF (FSSMIN.LE.0.01) ISCSW=4
      IF (FSSMIN.GE.0.0001) GO TO 1160
 1161 WRITE (6,1162) FSSMIN
 1162 FORMAT (////1H0,'******   ERROR   ******',10X,'MIN. VALUE = ',E16.
     *5)
 1160 SCALSW=MIN0(SCALSW,ISCSW)
      GO TO (1062,1063,1064,1065),SCALSW
 1062 CALL SYSSYM (X4,Y4,SZ3,'OVERLAY NO.1',12,THETA)
      GO TO 1066
 1063 CALL SYSSYM (X4,Y4,SZ3,'OVERLAY NO.2',12,THETA)
      GO TO 1066
 1064 CALL SYSSYM (X4,Y4,SZ3,'OVERLAY NO.3',12,THETA)
      GO TO 1066
 1065 CALL SYSSYM (X4,Y4,SZ3,'OVERLAY NO.4',12,THETA)
 1066 CONTINUE
      PXMIN=-6.42+SCALSW
      PXMAX=SCALSW
      PYMIN=-2.256
      PYMAX=2.03
      DO 25 J=1,ID
      DO 266 I=1,IP
      H(I)=ALOG10(PD(I))
  266 V(I)=-ALOG10(PSSV(J,I))
      CALL XYPLOT (IP,V,H,PXMIN,PXMAX,PYMIN,PYMAX,DD,0)
C     THICKENING THE LINE
      DS=-0.003
```

```
      CALL THLN(IP,DS)                                                  SPCT 186
      CALL XYPLOT (IP,V,H,PXMIN,PXMAX,PYMIN,PYMAX,DD,0)                 SPCT 187
      DS=0.006                                                          SPCT 188
      CALL THLN(IP,DS)                                                  SPCT 189
      CALL XYPLOT (IP,V,H,PXMIN,PXMAX,PYMIN,PYMAX,DD,0)                 SPCT 190
      DO 107 K=1,IP                                                     SPCT 191
      V(K)=ALOG10(PSSV(J,K))                                            SPCT 192
  107 H(K)=ALOG10(SV(J,K))                                              SPCT 193
      PUNCH 109 , DMP(J)                                                SPCT 194
  109 FORMAT('DAMPING = ',F4.2,' 6.28*SD/T SPECTRUM')                   SPCT 195
      JM2=2*J-2                                                         SPCT 196
      DO 110 L=1,7                                                      SPCT 197
      J2=13*L                                                           SPCT 198
      J1=J2-12                                                          SPCT 199
  110 PUNCH 132, (V(K),K=J1,J2),JM2,L                                   SPCT 200
      PUNCH 111, DMP(J)                                                 SPCT 201
  111 FORMAT('DAMPING = ',F4.2,' SV SPECTRUM')                          SPCT 202
      JM1=2*J-1                                                         SPCT 203
      DO 112 L=1,7                                                      SPCT 204
      J2=13*L                                                           SPCT 205
      J1=J2-12                                                          SPCT 206
  112 PUNCH 132, (H(K),K=J1,J2),JM1,L                                   SPCT 207
   25 CONTINUE                                                          SPCT 208
  132 FORMAT(13F6.3,2I1)                                                SPCT 209
      CALL PLOTXY(2,EX,EY,0.0,15.0,0.0,10.0,1,0,4,1,DD)                 SPCT 210
      RATIO=0.75                                                        SPCT 211
      CALL RVPLOT(TITLE1,TITLE2,TITLE4,TITLE5,NT1,NT2,RATIO,IP)         SPCT 212
C                                                                       SPCT 213
      WRITE(6,84)                                                       SPCT 214
   84 FORMAT(1H1)                                                       SPCT 215
      WRITE(6,86)                                                       SPCT 216
   86 FORMAT(1H )                                                       SPCT 217
      FMTTL(4)=ABCD((NT1/4)+1)                                          SPCT 218
      FMTSTN(4)=ABCD((NT2/4)+1)                                         SPCT 219
      FMTTL(2)=ABCD(MAX0((( 132-NT1)/2),1))                             SPCT 220
      FMTSTN(2)=ABCD(MAX0(((132-NT2)/2),1))                             SPCT 221
      INT1=NT1/4+1                                                      SPCT 222
      INT2=NT2/4+1                                                      SPCT 223
      WRITE(6,FMTTL)(TITLE1(I),I=1,INT1)                                SPCT 224
      WRITE(6,86)                                                       SPCT 225
      WRITE(6,FMTSTN)(TITLE2(I),I=1,INT2)                               SPCT 226
      WRITE(6,98)                                                       SPCT 227
   98 FORMAT (//1H ,25X'THE UNITS USED ARE SEC FOR PER, IN/SEC FOR FS, SSPCT 228
     AV, AND PSV, IN FOR SD AND G FOR SA')                              SPCT 229
      WRITE (6,87) IDMP                                                 SPCT 230
   87 FORMAT (/1H ,12X,5('DAMPING = ',I2,' PER CENT    '))              SPCT 231
      WRITE(6,89)                                                       SPCT 232
   89 FORMAT (/1H ,' PER',3X'FS',1X,5(3X'SD',4X'SV',4X'SA',4X'PSV'),/)  SPCT 233
      DO 204 I=1,45                                                     SPCT 234
       E1=ALOG10(FS(I))                                                 SPCT 235
      IE1=E1                                                            SPCT 236
      IF(IE1 .LE. 0 .AND. E1 .LT. 0.)  IE1=IE1-1                        SPCT 237
      IRG(1)=FS(I)*10.0**(-IE1)*100.                                    SPCT 238
      IF(1)=IE1-2                                                       SPCT 239
      DO 200 J=1,5                                                      SPCT 240
      LL=4*(J-1)                                                        SPCT 241
      ST(LL+1)=SD(J,I)                                                  SPCT 242
      ST(LL+2)=SV(J,I)                                                  SPCT 243
      ST(LL+3)=SA(J,I)                                                  SPCT 244
      ST(LL+4)=PSSV(J,I)                                                SPCT 245
  200 CONTINUE                                                          SPCT 246
      DO 201 J=1,20                                                     SPCT 247
```

```
      E1=ALOG10(ST(J))
      IE1=E1
      IF(IE1 .LE. 0 .AND. E1 .LT. 0.)  IE1=IE1-1
      IRG(J+1)=ST(J)*10.0**(-IE1)*100.
      IE(J+1)=IE1-2
  201 CONTINUE
      WRITE(6,202)  PD(I),(IRG(J),IE(J),J=1,21)
  204 CONTINUE
  202 FORMAT(1H ,F5.3,21(I4,I2))
      WRITE(6,84)
      WRITE(6,86)
      WRITE(6,86)
      WRITE(6,FMTSTN)(TITLE2(I),I=1,INT2)
      WRITE(6,98)
      WRITE (6,87) IDMP
      WRITE(6,89)
      DO 205  I=46,91
       E1=ALOG10(FS(I))
      IE1=E1
      IF(IE1 .LE. 0 .AND. E1 .LT. 0.)  IE1=IE1-1
      IRG(1)=FS(I)*10.0**(-IE1)*100.
      IE(1)=IE1-2
      DO 206 J=1,5
      LL=4*(J-1)
      ST(LL+1)=SD(J,I)
      ST(LL+2)=SV(J,I)
      ST(LL+3)=SA(J,I)
      ST(LL+4)=PSSV(J,I)
  206 CONTINUE
      DO 207 J=1,20
       E1=ALOG10(ST(J))
      IE1=E1
      IF(IE1 .LE. 0 .AND. E1 .LT. 0.)  IE1=IE1-1
      IRG(J+1)=ST(J)*10.0**(-IE1)*100.
      IE(J+1)=IE1-2
  207 CONTINUE
      WRITE(6,208)  PD(I),(IRG(J),IE(J),J=1,21)
  205 CONTINUE
  208 FORMAT(1H ,F5.2,21(I4,I2))
   51 CONTINUE
      RETURN
      END
```

```
      BLOCK DATA
      COMMON /BT3/RVRS(9),TITLE3(15),PS(4),RV(7)
      DATA TITLE3/'DAMP','ING ','VALU','ES A','RE 0',', 2,',' 5, ','10 A
     1',' ND 2','0 PE','RCEN','T OF',' CRI','TICA','L   '/
      DATA RVRS/'RELA','TIVE',' VEL','OCIT','Y RE','SPON','SE S','PECT',
     1'RUM '/
      DATA  PS/'PERI','OD -',' SEC','ONDS'/
      DATA RV/'RELA','TIVE',' VEL','OCIT','Y - ','IN/S','EC  '/
      END
```

Subroutine PCNO3 (Nigam, Jennings, Trifunac, Justiss)

PCNO3 computes the response of a single-degree of freedom oscillator for 5 damping values. It also computes response envelope spectra (RES) and writes them on tape.

Usage

CALL PCNO3 (IP, N, DEL, SF)

COMMON DMP(5), SD(5, 100), SA(5, 100), GA(10000)

COMMON /BT2/PD(100), SV(4, 100), FS(100)

COMMON /BLCKT/TTSD(5, 100), TTSV(5, 100), TTSA(5, 100)

Where

IP is the no. of periods at which to compute response.

N is the no. of acceleration data points.

DEL is the acceleration time interval.

SF is the scaling factor.

DMP is the 5 oscillator dampings.

SD is the response displacement.

SA is the response acceleration.

GA is the acceleration.

PD is the oscillator periods.

SV is the response velocity.

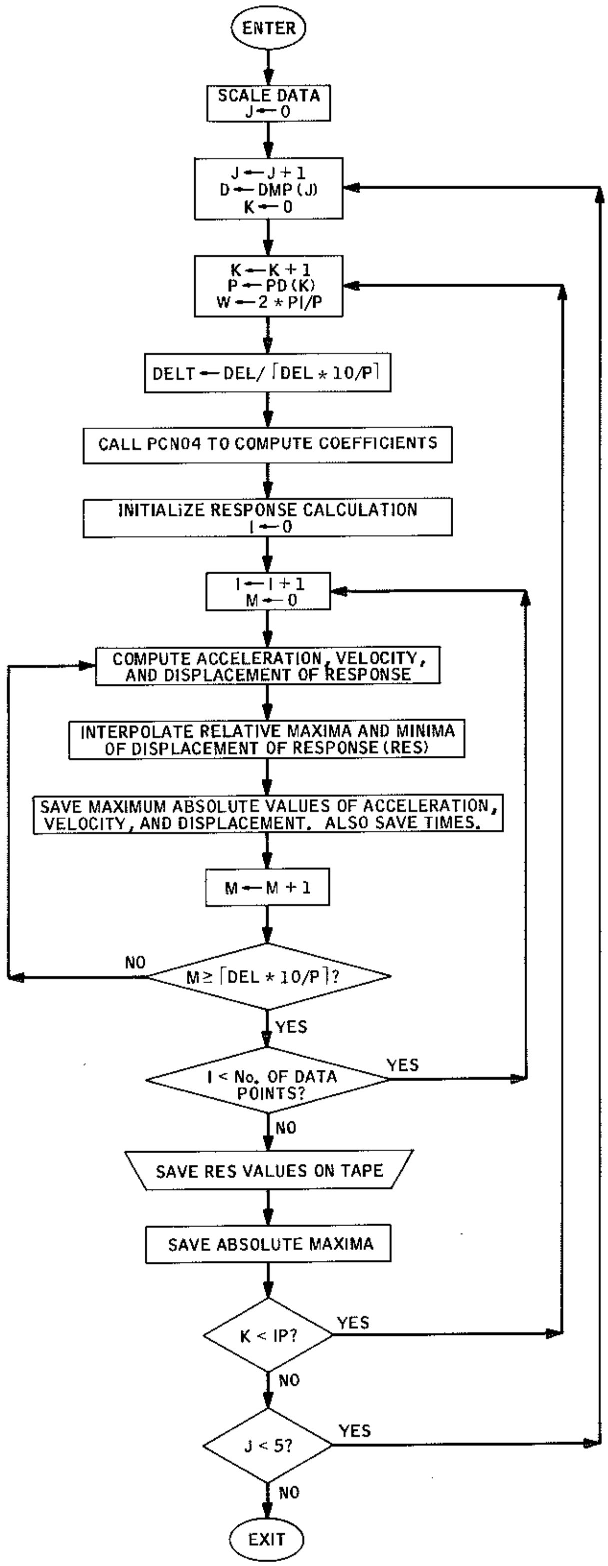
FS is the Fourier spectra.

TTSD is the times of maximum response displacement.

TTSV is the times of maximum response velocity.

TTSA is the times of maximum response acceleration.

PCN03 FLOW CHART



```
      SUBROUTINE PCN03 (IP,N,DEL,SF)
      COMMON DMP(5),SD(5,100),SA(5,100),GA(10000)
      COMMON /BT2/ PD(100),SV(5,100),FS(100)
      COMMON /BLCKT/ TTSD(5,100),TTSV(5,100),TTSA(5,100)
      REAL RTIME*8,TIMEP*8,DW,W2,AMAX,VMAX,DMAX,X,XX,VERTL,BEB,
     *     BB7,SL,G,TY,TYD,TYDD,DELT,DELP,DTPLOT,VEND,D,P,W,TD,AMD,
     *     TDL,AMDL,
     *     TTDMAX,TTVMAX,TTAMAX
      INTEGER L,IA,LSUM,I,J,M,K,NWR2,LP,LLP
      REAL A(2,2),B(2,2),SAVPT(750)
      REAL    AAAA,AAA,AA,AE,BBBB,BBB,BB,BE
      EQUIVALENCE (A(2,2),AAAA),(A(2,1),AAA),(A(1,2),AA),(A(1,1),AE)
      EQUIVALENCE (B(2,2),BBBB),(B(2,1),BBB),(B(1,2),BB),(B(1,1),BE)
      EQUIVALENCE (AMD,IAMD),(TD,ITD),(AMDL,IAMDL),(TDL,ITDL)
      NWR2=12
      DTPLOT=0.2
      LSUM=0
      LLP=750
      DO 200 J=1,N
      GA(J)=GA(J)*SF
  200 CONTINUE
      DO 4 J=1,5
      D=DMP(J)
      DO 5 K=1,IP
      P=PD(K)
      W=6.2831853071795860O/P
C
C      ****** CHOICE OF INTERVAL OF INTEGRATION ******
C
      DELP=P/10.
      L=DEL/DELP+1.-1.E-05
      VERTL=1.0/L
      DELT=DEL*VERTL
C
C      ****** COMPUTATION OF MATRICES A AND B ******
C
      CALL PCN04(D,W,DELT,A,B)
C
C      ****** INITIATION ******
C
      RTIME=0
      TIMEP=0
      AMD=0.0
      TD=1.0
      LP=1
      X=0
      XX=0
      DMAX=0
      VMAX=0
      AMAX=0
      I=1
      DW=-2.0*W*D
      W2=-W*W
C          NOTE NEGATIVES.
      BEB=-(BE+BB)
      BB7=-(BBB+BBBB)
      IA=2.0*P/DELT + 0.5
C
C      ****** COMPUTATION OF RESPONSE ******
C
    7 SL=(GA(I+1)-GA(I))*VERTL
```

```
      M=0
    6 G=GA(I)+SL*M
      TY=AA*XX-SL*BB+AE*X+BEB*G
      TYD=AAAA*XX-SL*BBBB+AAA*X+BB7*G
      TYDD = ABS(DW*TYD + W2*TY)
C
C       ***   MONITORING AND SAVING THE PEAK VALUES
C
      IF (LP-LLP) 153,153,112
  153 CONTINUE
      IF (TYD*XX) 154,154,112
  154 AMDL=AMD
      TDL=TD
      AMD=ABS(X)
      TD=RTIME
C
C            INTERPOLATION -- ONE POINT EVERY DTPLOT SECONDS.
C
  111 IF (TIMEP-RTIME)113,112,112
  113 SAVPT(LP)=AMDL+(AMD-AMDL)*(TIMEP-TDL)/(RTIME-TDL)
      LP=LP+1
      TIMEP=TIMEP+DTPLOT
      IF (LP-LLP) 111,111,112
  112 CONTINUE
C
C       ****** MONITORING THE MAX. VALUES ******
C
      IF (DMAX-ABS(TY)) 140,14,14
  140 DMAX=ABS(TY)
      TTDMAX=RTIME
   14 X=TY
      IF (VMAX-ABS(TYD)) 150,15,15
  150 VMAX=ABS(TYD)
      TTVMAX=RTIME
   15 XX=TYD
      IF (AMAX-TYDD) 160,16,16
  160 AMAX=TYDD
      TTAMAX=RTIME
   16 CONTINUE
      RTIME=RTIME+DELT
      M=M+1
      IF (M-L) 6,162,162
  162 CONTINUE
C
C       ****** TEST FOR END OF INTEGRATION ******
C
      I=I+1
      IF (I-N) 7,18,19
   19 IF (I-(N+IA)) 17,8,8
   18 VEND = SQRT(XX*XX - W2*X*X)
   17 GA(I+1)=0
      GO TO 7
    8 IF (K.NE.1.OR.J.NE.1) GO TO 163
C            ONLY AT THE HIGHEST FREQUENCY.
      LLP=LP-1
      WRITE (NWR2) LLP
  163 IF ((LP-1).LT.LLP) GO TO 17
      CALL WRTR (LLP,SAVPT)
      LSUM=LSUM+LLP
      IF (D.LT.1.0E-03) FS(K)=VEND
      SD(J,K)=DMAX
      SV(J,K)=VMAX
```

```
      SA(J,K)=AMAX
      TTSD(J,K)=TTDMAX
      TTSV(J,K)=TTVMAX
      TTSA(J,K)=TTAMAX
    5 CONTINUE
    4 CONTINUE
      WRITE (6,155) LP,P,IA,RTIME,LLP,D,LSUM,L,N
  155 FORMAT (/1H0,3(2XI5,2XF12.4),4(2XI10),////)
      RETURN
      END
```

```
SUBROUTINE WRTR (LLP,SAVPT)
DIMENSION SAVPT (LLP)
WRITE (12) SAVPT
RETURN
END
```

Subroutine PCNO4 (Nigam, Jennings)

PCNO4 is called by PCNO3. PCNO4 computes coefficients of the A and B matrices.

Usage

CALL PCNO4 (D, W, DELT, A, B)

Where

D is the oscillator damping.

W is the oscillator frequency.

DELT is the interval of response computation.

A is the A matrix, dimension (2, 2).*

B is the B matrix, dimension (2, 2).*

```
      SUBROUTINE PCN04 (D,W,DELT,A,B)
C
C          SUBROUTINE FOR COMPUTATION OF MATRICES A AND B
C                         PCN04
C
      DIMENSION A(2,2),B(2,2)
      DW=D*W
      D2=D*D
      A0=EXP(-DW*DELT)
      A1=W*SQRT(1.-D2)
      AD1=A1*DELT
      A2=SIN(AD1)
      A3=COS(AD1)
      A7=1.0/(W*W)
      A4=(2.0*D2-1.0)*A7
      A5=D/W
      A6=2.0*A5*A7
      A8=1.0/A1
      A9=-(A1*A2+DW*A3)*A0
      A10=(A3-DW*A2*A8)*A0
      A11=A2*A8
      A12=A11*A0
      A13=A0*A3
      A14=A10*A4
      A15=A12*A4
      A16=A6*A13
      A17=A9*A6
      A(1,1)=A0*(DW*A11+A3)
      A(1,2)=A12
      A(2,1)=A10*DW+A9
      A(2,2)=A10
      DINV=1.0/DELT
      B(1,1)=(-A15-A16+A6)*DINV-A12*A5-A7*A13
      B(1,2)=(A15+A16-A6)*DINV+A7
      B(2,1)=(-A14-A17-A7)*DINV-A10*A5-A9*A7
      B(2,2)=(A14+A17+A7)*DINV
      RETURN
      END
```

Subroutine THLN (Trifunac)

THLN (thickens line) is used to displace each segment of a line with coordinates (PD(I), VM(I)) perpendicular to itself.

Usage

CALL THLN (IP, DS)

COMMON /BT1/VM(100), PD(100)

Where

IP is the no. of points in the line.

DS is the desired displacement.

VM is the Y-value of a line coordinate.

PD is the X-value of a line coordinate.

```
      SUBROUTINE THLN(IP,DS)
      COMMON /BT1/ VM(100),PD(100)
C     ROUTINE FOR THICKENNING THE LINE
      H1=PD(2)-PD(1)
      V1=VM(2)-VM(1)
      HP1=SQRT(H1*H1+V1*V1)
      C1=H1/HP1
      T1=V1/H1
      VMM=VM(1)
      VM(1)=VM(1)-DS*C1
      IPM=IP-1
      XAM=PD(1)+DS*V1/HP1
      DO 17 I=2,IPM
      H2=PD(I+1)-PD(I)
      V2=VM(I+1)-VM(I)
      HP2=SQRT(H2*H2+V2*V2)
      C2=H2/HP2
      T2=V2/H2
      AR2=T1-T2
      IF (AR2.NE.0.0) GO TO 20
      PD(I-1)=XAM
      XAM=PD(I)+DS*V2/HP2
      VMM=VM(I)
      VM(I)=VM(I)-DS*C2
      GO TO 21
   20 CONTINUE
      AR1=VM(I)-VMM + DS/C1 - DS/C2 + T1*PD(I-1) - T2*PD(I)
      VMM=VM(I)
      PD(I-1)=XAM
      XAM=AR1/AR2
      VM(I)=VM(I) - DS/C2 + T2*(XAM-PD(I))
   21 CONTINUE
      C1=C2
      T1=T2
   17 CONTINUE
      PD(IPM)=XAM
      VM(IP)=VM(IP)-DS*C2
      PD(IP)=PD(IP)+DS*V2/HP2
      RETURN
      END
```

Subroutine RVPLOT (Lee)

RVPLOT is called by SPCTRA in Volume III processing. RVPLOT produces a plot for five dampings of velocity response spectra vs. period on a linear scale. The Fourier spectra are also plotted with a dashed line on the same page. The horizontal scale (period) is divided into two sections: 0 - 3 sec takes up 3/4 of the total length of the horizontal axis and 3 - 15 sec takes up the remainder.

Usage

```
CALL RVPLOT (TTLE1, TTLE2, TTLE4, TTLE5, NT1, NT2,
*       RATIO, NX)
COMMON /BT2/X(100), Y(5, 100), FS(100)
COMMON /BT3/RVRS(9), DV(15), PS(4), RV(7)
```

Where

TTLE1 is the earthquake title.

TTLE2 is the accelerogram title.

TTLE4 is the title for the solid lines (4 characters).

TTLE5 is the title for the dashed line (4 characters).

NT1 is the no. of characters in TTLE1.

NT2 is the no. of characters in TTLE2.

RATIO is the ratio of the uncompressed horizontal scale to the total horizontal scale.

NX is the no. of periods.

COMMON /BT2/ has

X is the period coordinates.

Y is the velocity response for 5 dampings.

FS is the Fourier spectra.

COMMON /BT3/ has titles and axes labels.

They are initialized in BLOCK DATA.

Subroutine SQBOUN is used inside RVPLOT to draw a square boundary around the plot. SQBOUN uses a Caltech plotting routine, SYSPLT, to draw the lines.

```
      SUBROUTINE RVPLOT(TTLE1, TTLE2, TTLE4, TTLE5, NT1, NT2, RATIO, NX)
C     TITLE5 IS FOR DASH-DASH LINE
C     TITLE4 IS FOR FUL LINE
      COMMON  /BT2/ X(100),Y(5,100),FS(100)
      COMMON /BT3/RVRS(9),DV(15),PS(4),RV(7)
      DIMENSION XMAX(2),XMIN(2)
      DIMENSION NXS(2)
      DIMENSION YL(6),YY(5),LT(6)
      DIMENSION XH(4),LH(4),A(4)
      DIMENSION XX(25)
      DIMENSION TTLE1(1),TTLE2(1),TTLE4(1),TTLE5(1)
      EQUIVALENCE (XLEN1,IXLEN1),(XLEN2,IXLEN2),(NP,XNP),(IPEN,XIPEN)
      EQUIVALENCE (SX,ISX),(XV,IXV),(CH,ICH),(A(1),IAA),(XI,IXI)
      EQUIVALENCE (D3, ID3),(DD1,IDD1),(DD2,IDD2),(DI,IDI),(D2,ID2)
      EQUIVALENCE (D1,ID1)
      CALL SYSOFF (0.0,1.0,0.0,1.0)
      XR=13.5
      YR=8.2
      XL=2.5
      YLL=-0.3
      IDIV=6
      CALL SQBOUN (XR,YR,IDIV,XL,YLL)
C PLOT THE TITLES
      X1=8.-3.*.14*35./7.
      X2= 8.-3.*.105*FLOAT(NT1)/7.
      X3= 8.-3.*.105*FLOAT(NT2)/7.
      X4= 8.-3.*.105*57./7.
      CALL SYSSYM(X1, 7.25, .14, RVRS, 35, 0.)
      CALL SYSSYM(X2, 7.05, .105, TTLE1, NT1, 0.)
      CALL SYSSYM(X3, 6.85, .105, TTLE2, NT2, 0.)
      CALL SYSSYM(X4, 6.65, .105, DV, 57, 0.)
      CALL SYSPLT(9.45, 5.8, 3)
      CALL SYSPLT(10.41, 5.8, 2)
      CALL SYSSYM(10.5, 5.75, .105, TTLE4, 4, 0.)
      XT5=9.45
      YT5=5.55
      DO 100 I=1,8
      CALL SYSPLT(XT5, YT5, 3)
      CALL SYSPLT(XT5+.08, YT5, 2)
  100 XT5=XT5+.12
      CALL SYSSYM(10.5, 5.5, .105, TTLE5, 4, 0.)
C PLOT THE PLOT BOUNDARY
      XR=12.0
      YR=6.5
      XL=4.5
      YLL=1.5
      CALL SQBOUN (XR,YR,IDIV,XL,YLL)
C     PLOT THE HORIZONTAL AXIS
      XLEN=7.5
      XLEN1=XLEN*RATIO
      XLEN2=XLEN-XLEN1
      X0=4.5
      Y0=1.5
      X1=X0+XLEN1
      YLEN=5.
      YTOP=Y0+YLEN
      CALL LABEL(X0, Y0, 0., 3., XLEN1, -3, RV, 0, 0)
      CALL LABEL(X1, Y0, 3., 15., XLEN2, -3, RV, 0, 0)
      CALL SYSSYM(7.5, 1.15, .1, PS, 16, 0.)
      XX(25)=X0+XLEN
      DO 1 IK=1,24
```

```
      I=25-IK
      XI=XLEN1/12.
      IF (I.GE.13) XI=XLEN2/12.0
      XX(I)=XX(I+1)-XI
      CALL SYSPLT(XX(I), Y0, 3)
      CALL SYSPLT(XX(I), Y0+0.1, 2)
    1 CONTINUE
      DO 8 I=1,4
      XH(I)=X0+FLOAT(I-1)*XLEN1/3.
      LH(I)=I-1
      CALL OUTCOR(A,NUM)
      WRITE(6,9)LH(I)
      CALL OUTCOR
    9 FORMAT(I1)
      CALL SYSSYM(XH(I)-.025, Y0-.15, .1, A, NUM*4, 0.)
    8 CONTINUE
      DO 20 I=1,3
      CH=.06
      IF(I.EQ.1)CH=.115
      XH(I)=X1+FLOAT(I)*XLEN2/3.
      LH(I)=7+(I-1)*4
      CALL OUTCOR(A,NUM)
      WRITE (6,21)LH(I)
      CALL OUTCOR
   21 FORMAT(I2)
      CALL SYSSYM(XH(I)-CH , Y0-.15, .1, A, NUM*4, 0.)
   20 CONTINUE
C     PLOT THE VERTICAL AXIS
      YMAX=Y(1,1)
      I=2
    2 CONTINUE
      IF(YMAX.LT.Y(1,I))YMAX=Y(1,I)
      I=I+1
      IF(I.LE.NX)GO TO 2
      LIM=INT(YMAX*1.1)
      LIMY=LIM+10-MOD(LIM,10)
      YL(1)=1.5
      LT(1)=0
      XR=X0+XLEN
      DO 3 I=1,5
      YL(I+1)=YL(I)+1.
      YY(I)=YL(I)+.5
      LT(I+1)=LT(I)+LIMY/5
      CALL SYSPLT(XR,YY(I),3)
      CALL SYSPLT(XR-.1,YY(I),2)
      CALL SYSPLT(XR,YL(I+1),3)
      CALL SYSPLT(XR-.15,YL(I+1),2)
    3 CONTINUE
      DO 50 IK=1,24
      I=25-IK
      CALL SYSPLT (XX(I),YTOP,3)
      CALL SYSPLT (XX(I),YTOP-0.1,2)
   50 CONTINUE
      DO 51 IK=1,5
      I=6-IK
      CALL SYSPLT(X0,YL(I+1),3)
      CALL SYSPLT(X0+.15,YL(I+1),2)
      CALL SYSPLT(X0,YY(I),3)
      CALL SYSPLT(X0+.1,YY(I),2)
   51 CONTINUE
      C0=.3
      C1=.05
```

```
      DO 4 I=1,6
      CALL OUTCOR(A,NUM)
      WRITE(6,5)LT(I)
      CALL OUTCOR
    5 FORMAT(I3)
    4 CALL SYSSYM(XO-CO, YL(I)-C1, .1 , A, NUM*4, 0.)
      Y1=4.-3.*.08*24./7.
      CALL SYSSYM(4.15, Y1, .1, RV, 26, 90.)
      XV=XO
      DO 30 I=1,5
      SX=XLEN1/3.
      IF(I.GT.3)SX=XLEN2/3.
      XV=XV+SX
      CALL SYSPLT(XV, YTOP, 3)
   30 CALL SYSPLT(XV, YTOP-.15, 2)
C PLOT THE GRAPH
      NXS(1)=0
      NXS(2)=0
      DO 6 I=1,NX
      IF(X(I).LE.3.)NXS(1)=NXS(1)+1
      IF(X(I).GT.3.)NXS(2)=NXS(2)+1
    6 CONTINUE
      SY=5./FLOAT(LIMY)
      YMIN=-1.5/SY
      YMAX=8.5/SY
      SX1=XLEN1/3.
      SX2=XLEN2/12.
      XMIN(1)=-4.5/SX1
      XMAX(1)=10.5/SX1
      XMIN(2)=3.-X1/SX2
      XMAX(2)=3.+(15.-X1)/SX2
      DO 7 J=1,5
      DO 7 I=1,2
      NP=0
      IF(I.EQ.2)NP=NXS(1)-1
      IF(NXS(I).EQ.0)GO TO 7
      IPEN=3
      NI=NXS(I)
      IF(I.EQ.2)NI=NXS(I)+1
      DO 7 K=1,NI
      XXP=(X(NP+K)-XMIN(I))*15./(XMAX(I)-XMIN(I))
      YYP=(Y(J,NP+K)-YMIN)*10./(YMAX-YMIN)
      CALL SYSPLT(XXP, YYP, IPEN)
      IPEN=2
    7 CONTINUE
C PLOT THE FS GRAPH
      DI=.04
      D1=.08
      J = 1
      DD1=0.
      DD2=0.
      I=1
   11 X1=(X(I)-XMIN(J))*15./(XMAX(J)-XMIN(J))
      Y1=(FS(I)-YMIN)*10./(YMAX-YMIN)
      X2=(X(I+1)-XMIN(J))*15./(XMAX(J)-XMIN(J))
      Y2=(FS(I+1)-YMIN)*10./(YMAX-YMIN)
   12 DO=SQRT((X2-X1)**2+(Y2-Y1)**2)
      D3=D1
      IF(DD1.NE.0.)D3=DD1
      DD1=0.
      IF(DD2.EQ.0.)GO TO 13
      X1=(X2-X1)*DD2/DO+X1
```

```
      Y1=(Y2-Y1)*DD2/D0+Y1
      D0=SQRT((X2-X1)**2+(Y2-Y1)**2)
      DD2=0.
   13 CALL SYSPLT(X1,Y1,3)
      IF (D3.GT.D0)GO TO 14
      XD1=(X2-X1)*D3/D0+X1
      YD1=(Y2-Y1)*D3/D0+Y1
      CALL SYSPLT(XD1,YD1,2)
      D2=D3+DI
      IF(D2.GT.D0)GO TO 15
      X1=(X2-X1)*D2/D0+X1
      Y1=(Y2-Y1)*D2/D0+Y1
      GO TO 12
   14 CALL SYSPLT(X2,Y2,2)
      DD1=D3-D0
      GO TO 16
   15 DD2=D2-D0
   16 I=I+1
      IF(I.GE.NX)GO TO 17
      IF(I.GE.NXS(1))J=2
      GO TO 11
   17 CONTINUE
      CALL SYSEND (1,0.0)
      RETURN
      END
```

```
      SUBROUTINE SQBOUN (XTOP,YTOP,IDIV,XBOT,YBOT)
      REAL*8 DIV
      DIV=1.0D0/IDIV
      XINC=(XTOP-XBOT)*DIV
      YINC=(YTOP-YBOT)*DIV
      CALL SYSPLT (XBOT,YBOT,13)
      YCORD=YBOT
      DO 1 I=1,IDIV
      XCORD=XINC*I + XBOT
    1 CALL SYSPLT (XCORD,YCORD,12)
      DO 2 I=1,IDIV
      YCORD=YINC*I + YBOT
    2 CALL SYSPLT (XCORD,YCORD,12)
      DO 3 I=1,IDIV
      XCORD = XTOP-XINC*I
    3 CALL SYSPLT (XCORD,YCORD,12)
      DO 4 I=1,IDIV
      YCORD=YTOP-YINC*I
    4 CALL SYSPLT (XCORD,YCORD,12)
      RETURN
      END
```

DATA PROCESSING FOR VOLUME IV: FOURIER AMPLITUDE SPECTRA, VOLUME IV TAPE

The corrected accelerogram data obtained from the second stage of processing and published in the different parts of Volume II (Hudson, et al, 1971a) are retrieved from the Volume II magnetic tape storage and, by means of the Cooley-Tukey algorithm the Fourier amplitude spectra are calculated. The Fourier amplitude spectrum is calculated and smoothed by the subroutine MDTRHA using the 1/4, 1/2, 1/4 weights. MDTRHA calls RHARM which calls HARM (a coded version of the Cooley-Tukey algorithm). Volume IV, Part A, Report No. EERL 72-100 (Hudson, et al, 1972b) summarizes the method used for these calculations and the meaning of the 95% confidence level.

Details concerning identification are given at the top on each plot (Figures 9 and 10; plotted by the subroutines MDTPLT and LOGLOG respectively). The second line gives the name, data and time of occurrence of the earthquake; the third line is comprised of two labels, the observation station and the component processed. The Roman numeral "IV" in the first identification label indicates that the results pertain to the fourth stage of data processing, i.e., Volume IV contains the Fourier amplitude spectra of accelerogram records already corrected for base line adjustment and instrument response. The letter "A" following the Roman numberals implies that the processed record belongs to Part A of Volume IV. Volume IV, Part A contains the Fourier spectra of acceleration for the same records for which the corrected accelerograms are published

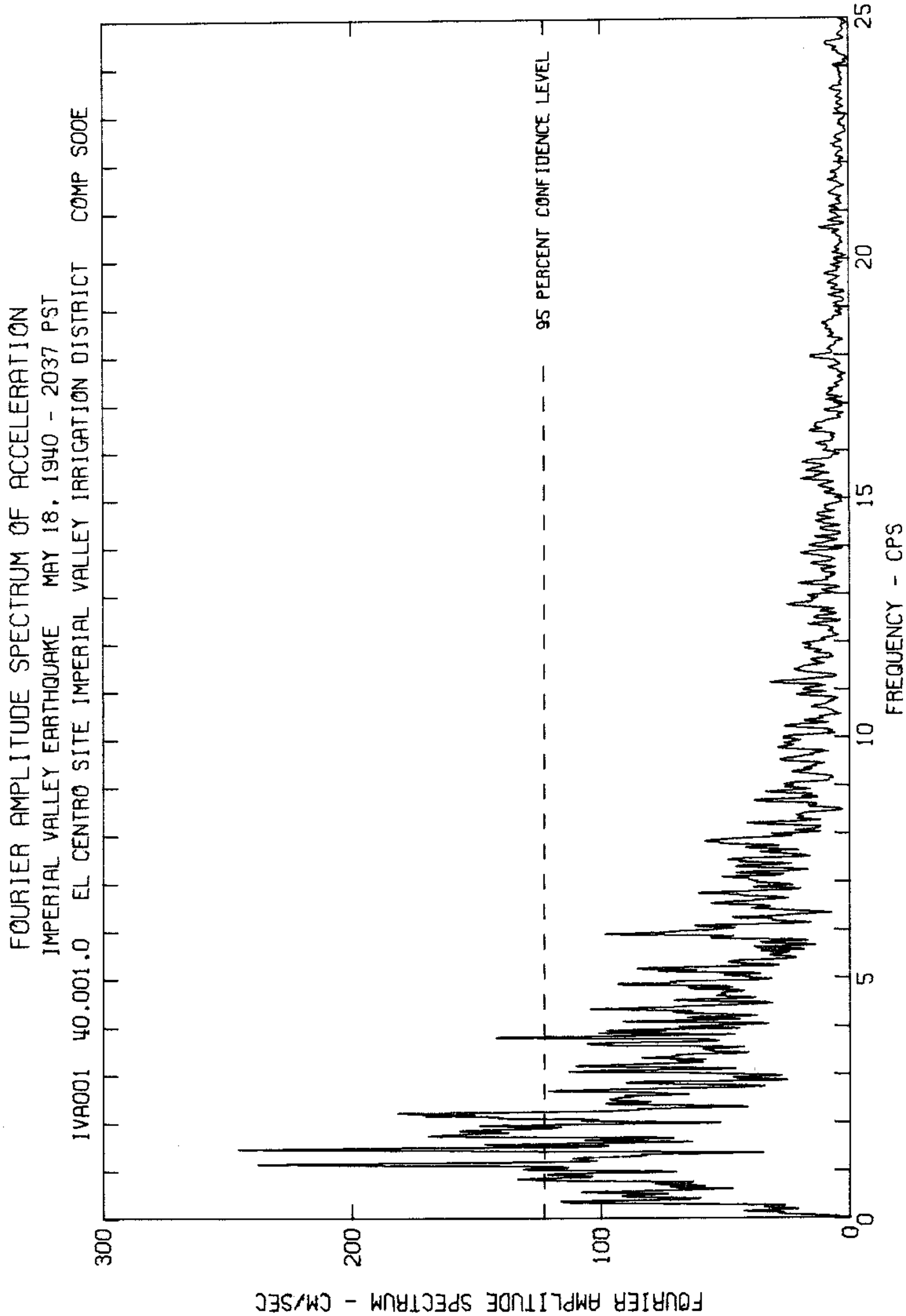

FOURIER AMPLITUDE SPECTRUM OF ACCELERATION
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
IVA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE
95 PERCENT CONFIDENCE LEVEL
FOURIER AMPLITUDE SPECTRUM - CM/SEC
300
200
100
0
0
5
10
15
20
25
FREQUENCY - CPS

Figure 9

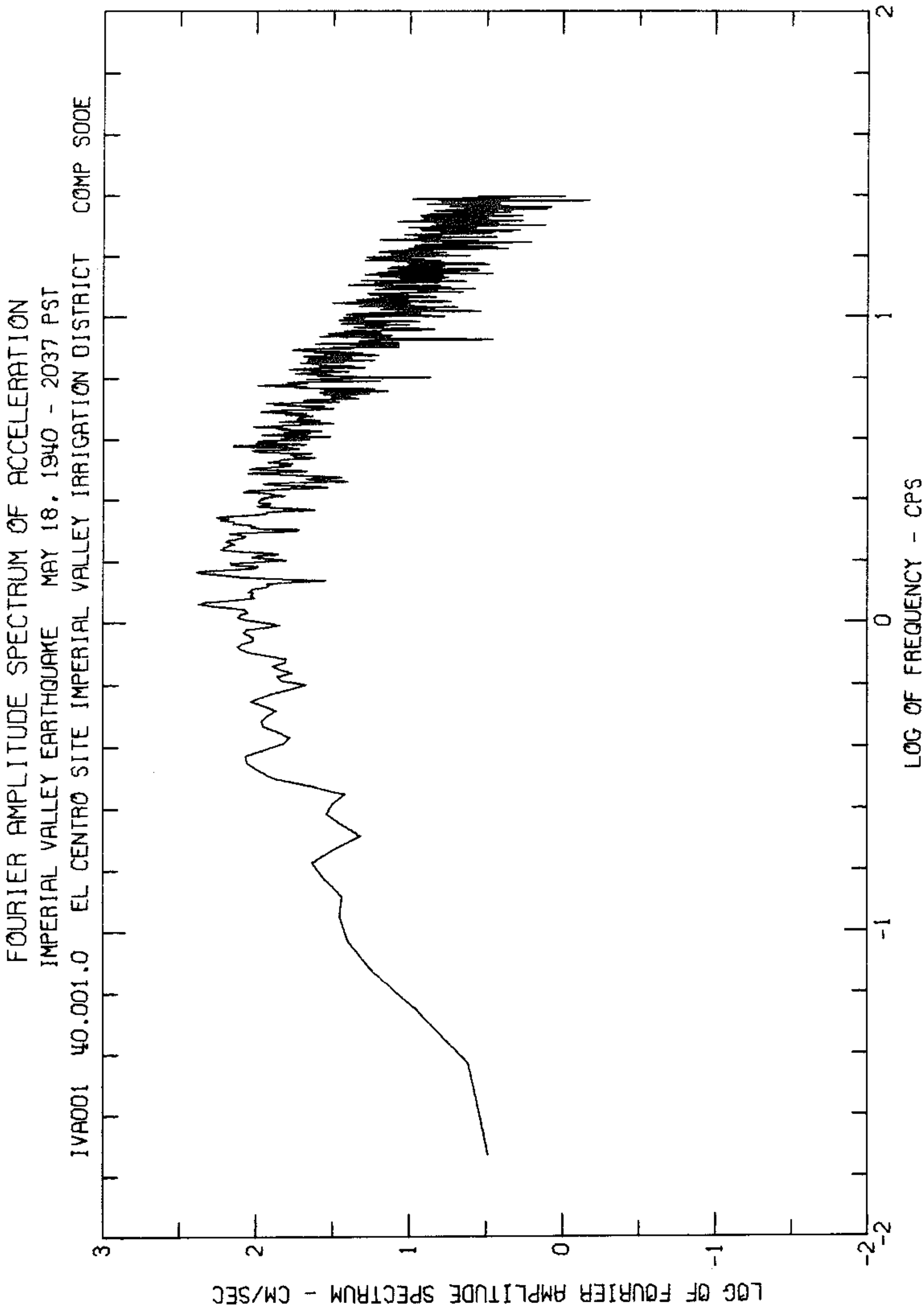

FOURIER AMPLITUDE SPECTRUM OF ACCELERATION
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
IVA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E
LOG OF FOURIER AMPLITUDE SPECTRUM - CM/SEC
LOG OF FREQUENCY - CPS
3
2
1
0
-1
-2
-2
-1
0
1
2

Figure 10

in Volume II, Part A (Hudson, et al, 1971). The three-digit number completing the first label is the Caltech Reference Number for the given earthquake record in Volume I. The second label is a string of three numbers separated by periods: the first number gives the year in which the earthquake occurred, the second is the serial number of the record as it was received at the Caltech Earthquake Engineering Research Laboratory during that year, and the last number indicates whether it was a main event or an aftershock (sequentially numbered, the main event starting from zero). On the linear spectrum plot (Figure 9), the data lying above the 95 percent confidence level (Hudson, et al, 1972b) may be considered relevant to that degree. The spectra have been plotted up to a frequency of 25 cps on linear and logarithmic scales, corresponding to the capabilities of the instrumentation and data processing methods used.

The spectral data are also stored on magnetic tapes, copies of which will be available on request from the National Information Service for Earthquake Engineering at the California Institute of Technology. The set-up of one file of the Volume IV tape is as follows:

Volume IV Tape
(one file per one acceleration component)

Each file has:

1. Heading data of alphanumeric type
2. Heading data of integer type
3. Heading data of floating point type
4. Pairs of real and imaginary parts of the Fourier amplitude spectra calculated from the corrected accelerograms which is read from the Volume II tape

Tape parameters: 1600 bpi, LRECL=1204, BLKSIZE=3616, RECFM=VBS. The detailed description and a sample of the heading data set are given in the following section.

VOLUME IV HEADING DATA

Punched Output Card No.	Heading Data Array	Description
1 - 10	HEDER(I), I=1, 200	Volume IV identification & description
11 - 35	CORTIL(I), I=1, 500	Same as CORTIL of Volume II heading data
36 - 40	ICOR(I), I=1, 100	Same as ICOR of Volume II heading data
41 - 50	FCOR(I), I=1, 100	Same as FCOR of Volume II heading data
51	(FLN(I), I=1, 4), NFILE	File number
52 - 53	TITLE2(I), I=1, 40	Earthquake title*
54 - 55	TITLE3(I), I=1, 40	Volume IV earthquake title of the file
56 - 57	(TTLN(I), I=1, 20), NT2, NT3	Number of letters in the above 2 titles

* In earlier parts of Volume IV data tapes only 72 locations were allotted for this title. The most recent version contains 40 A4 words, i.e., 160 letters.

HEADER DATA FOR TAPE FILES 1 THROUGH 60 FOR VOLUME IVA

THIS TAPE CONTAINS THE REAL AND IMAGINARY COMPONENTS OF THE FOURIER AMPLITUDE SPECTRUM OF THE ACCELEROGRAMS CONTAINED IN VOLUME IIA.
THE TIME LENGTHS OF THE ACCELEROGRAMS USED ARE EQUAL TO THOSE OBTAINED FROM DATA IN VOLUMEIIA. FOR TIME LENGTHS GREATER THAN 80 SECS. M = 12,FOR TIME LENGTHS LESS THAN 40 SECS M = 10,FOR TIME LENGTHS BETWEEN 40 AND 80 SECS M = 11
THE SPECTRAL AMPLITUDES ARE GIVEN IN ARRAY ZI,WHERE,Z(2*J) IS THE SINE TRANSFORM AND ZI(2*J-1) IS THE COSINE TRANSFORM ----ALL AMPLITUDES IN CMS/SEC
THE TRANSFORM CALCULATED OF THE SIGNAL X(T)ISTHE INTEGRAL OF X(T)*EXP(-IWT)
THE FOLLOWING IS DATA OBTAINED FROM VOL IIA

```
CORRECTED ACCELEROGRAM  IIA001  40.001.0    COMP SOOE   FILE  1 CORRESPONDING TO
FILE   1  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-A, EERL 70-20
IMPERIAL VALLEY EARTHQUAKE
MAY 18, 1940 - 2037 PST
IA001   40.001.0 S                                                            18
STATION NO. 117   32 47 43N,115 32 55W                                        38
EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT                            50
COMP SOOE                                                                     9
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST                          52
EPICENTER  32 44 00N,115 27 00W                                               31
INSTR PERIOD = 0.0990 SEC  DAMPING = 0.552                                    42
NO. OF POINTS = 985   DURATION = 53.73 SEC                                    42
UNITS ARE SEC AND G/10.                                                       23
RMS ACCLN OF COMPLETE RECORD = 0.4876 G/10.                                   43
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  2688  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF    0.02   SEC.
PEAK ACCELERATION =    341.69531  CMS/SEC/SEC AT    2.1200 SEC
PEAK VELOCITY     =     33.44914  CMS/SEC     AT    2.1800 SEC
PEAK DISPLACEMENT =     10.86678  CMS         AT    8.5800 SEC
INITIAL VELOCITY  =     -4.66421  CMS/SEC        INITIAL DISP. =     2.15852 CMS
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST

IIA001  40.001.0    EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT   COMP SOO
E
   1   1    1   1  40   1    0    4 117   32   47   43-115   32   55   32   44    0-115   27
   0   5   1819402037   0 180 985   26   50    0    0    0    0    0    0    0    0    0    0
   0   0    0   0   0    0    0    0    0    0 985 9872688    2   10   10    1    0   52   81
  10  10    21344   5 538   0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0   0    0   0   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0.099     0.552    53.730     0.488     0.100     0.0       0.0       0.0       0.0       0.0
   0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
   0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
   0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
   0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
   1.000     0.010     0.010    53.732    63.466     1.000     1.000    27.000     2.000    53.740
   0.020     0.070     0.020     0.0       2.120   341.695     2.180    33.449     8.580    10.867
  -4.664     0.070    25.000     0.200     0.200     2.159     0.0       0.0       0.0       0.0
   0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
   0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
FILE NUMBER  =      1
IMPERIAL VALLEY EARTHQUAKE   MAY 18, 1940 - 2037 PST

IVA001  40.001.0    EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT   COMP SOO
E
THE FOLLOWING TWO NUMBERS GIVE THE LENGTHS OF TITLE2 AND TITLE3 RESPECTIVELY
  52  81
```

PROGRAMS FOR PROCESSING VOLUME IV DATA

Volume IV MAIN Program (Trifunac, Vijayaraghavan)

The Volume IV MAIN program reads a Volume II tape and Fourier transforms the acceleration. The resulting Fourier amplitude spectrum is plotted. The Fourier coefficients are written on tape.

Usage

This programs reads in the following data:

NS, NFILE, HEDER(18), INW

and

HEDER(14), LPART, TITLE3(1), HEDER(84)

Where

NW = first file number of a Vol. II tape to be read in

NFILE = last file number of a Vol. II tape to be read in

HEDER(18) = alphanumeric field (A4) containing 'IVxb' where x is a letter identifying the volume part, e.g., IVBb

INW = three-digit Caltech reference number for the first record in the corresponding part of Volume II

HEDER(14) = alphanumeric field (A4) containing 'Hbww' where ww is the total number of files in this part

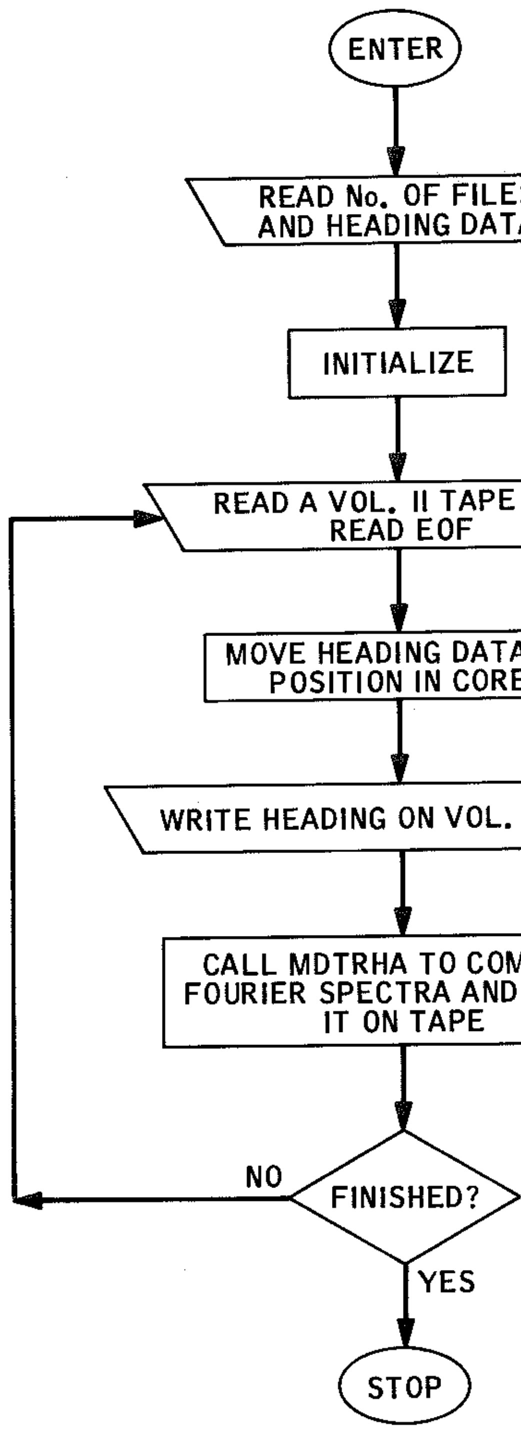
LPART = alphanumeric field (A4) containing 'pbbb' where 'p' is the letter identifying this part, e.g., 'B'

TITLE3(1) = alphanumeric field (A4) containing 'IVq0' where 'q' is the letter identifying this part, e.g., 'B'

HEDER(84) = alphanumeric field (A4) containing 'Iq.b' where

'q' is the letter identifying this part, e.g., 'B'.

VOL. IV MAIN PROGRAM



```
      COMMON TI(8200),ZI(8200),INV(1400),
     *      S(1400)
      COMMON / BM1/ FREQ(4200),AMB(5810)
      DIMENSION Z(5005)
      DIMENSION T(5005)
      DIMENSION HEDER(200),TITLN(20),HED2(100)
      EQUIVALENCE (HED2(1),HEDER(101))
      EQUIVALENCE (T(1),FREQ(1))
      EQUIVALENCE (Z(1),AMB(806))
      INTEGER TITLE2,TITLE3,HEDER,LPART,CORTIL
       DIMENSION FLN(4)
      DATA HEDER / 4*'    ',' HEA','DER ','DATA',' FOR',' TAP','E FI',
     *      'LES ','I TH','ROUG','    ',' FOR',' VOL','UME ','    ',
     *      2*'    ',
     *      'THIS',' TAP','E CO','NTAI','NS T','HE R','EAL ','AND ',
     *      'IMAG','INAR','Y CO','MPON','ENTS',' OF ','THE ','FOUR',
     *      'IER ','AMPL','ITUD','E   ',
     *      'SPEC','TRUM',' OF ','ACCE','LERO','GRAM','S CO','NTAI',
     *      'NED ','IN V','OLUM','E II','    ',7*'    ',
     *      'THE ','TIME',' LEN','GTHS',' OF ','THE ','ACCE','LERO',
     *      'GRAM','S US','ED A','RE E','QUAL',' TO ','THOS','E OB',
     *      'TAIN','ED F','ROM ','DATA',' IN ','VOLU','ME I','    ',
     *      ' FOR',' TIM','E LE','NGTH','S GR','EATE','R TH','AN 8',
     *      '0 SE','C, M','=12,',' FOR',' TIM','E LE','NGTH','S   '/
      DATA HED2/
     *      'LESS',' THA','N 40',' SEC',', M ','= 10',', FO','R TI',
     *      'ME L','ENGT','HS B','ETWE','EN 4','0 AN','D 80',' SEC',
     *      ', M ','= 11',2*'    ',
     *      'THE ','SPEC','TRAL',' AMP','LITU','DES ','ARE ','GIVE',
     *      'N IN',' ARR','AY Z','I, W','HERE',' ZI(','2*J)',' IS ',
     *      'THE ','SINE',2*'    ',
     *      'TRAN','SFOR','M AN','D ZI','(2*J','-1) ','IS T','HE C',
     *      'OSIN','E TR','ANSF','ORM ',' ---',' ALL',' AMP','LITU',
     *      'DES ','IN C','MS/S','EC. ',
     *      'THE ','TRAN','SFOR','M CA','LCUL','ATED',' FRO','M TH',
     *      'E SI','GNAL',' X(T',') IS',' THE',' INT','EGRA','L OF',
     *      ' X(T',')*EX','P(-I','WT).',
     *      '  TH','E FO','LLOW','ING ','IS D','ATA ','OBTA','INED',
     *      ' FRO','M VO','L II','    ',8*'    '/
      DATA TITLN/'THE ','FOLL','OWIN','G TW','O NU','MBER','S GI',
     *      'VE T','HE L','ENGT','HS O','F TI','TLE2',' AND',' TIT',
     *      'LE3 ','RESP','ECTI','VELY','    '/
       DATA FLN/'FILE',' NUM','BER ',' =  '/
      DIMENSION TITLE2(40),TITLE3(40)
       DIMENSION CORTIL(500),ICOR(100),FCOR(100)
      EQUIVALENCE (TITLE2(1),CORTIL(421))
      READ (5,1) NW,NFILE,HEDER(18),INW
    1 FORMAT (6X,2I2,A4,I3)
      READ (5,2) HEDER(14),LPART,TITLE3(1),HEDER(84)
    2 FORMAT (6X,10A4)
      HEDER(53)=LPART
      HEDER(192)=LPART
      IF (1-NW) 20,21,21
   20 KSKFL=NW-2
   22 READ (35,END=23)
      GO TO 22
   23 CALL READNF (35,KSKFL)
   24 READ (50,END=25)
      GO TO 24
   25 CALL READNF (50,KSKFL)
   21 CONTINUE
```

```
      DO 181 JFILE = NW,NFILE
      TITLE3(1)=TITLE3(1)+(INW+(JFILE-1)/3)/100
      READ(35) CORTIL,ICOR,FCOR
      NDATA = ICOR(53)
      READ (35) (Z(K),K=1,NDATA)
  100 READ (35,END=230)
      GO TO 100
  230 CALL READNF(35)
      WRITE (6,600) HEDER,TITLE3(1)
      WRITE(6,600) CORTIL
  600 FORMAT(2X,20A4)
      WRITE(6,601) ICOR
  601 FORMAT(1X,'ICOR'/(1X,10I10))
      WRITE(6,602) FCOR
  602 FORMAT(1X,'FCOR'/(1X,10E13.6))
      NT2 = ICOR(59)
      NT3 = ICOR(60)
      DO 14  J = 2,40
   14 TITLE3(J) = CORTIL(460+J)
      M = 12
      NSKIP=1
      IEX=0
      WO=25.0*6.28318
      SCALE2=1.
      DDTG=0.02
      DO 5 I=1,NDATA
      T(I)=(I-1)*DDTG
    5 CONTINUE
      TEND = T(NDATA)
      IF(TEND.LT.80.) M = 11
      IF(TEND.LT.40.) M = 10
      WRITE(50) HEDER,CORTIL,ICOR,FCOR,FLN,JFILE
      WRITE(50) TITLE2,TITLE3,TITLN,NT2,NT3
      CALL      MDTRHA(NDATA,M,WO,TITLE2,NT2,TITLE3,NT3,NTR,NSKIP,IEX,
     1 ENF,DFRQ,IFERR,YGP)
      WRITE(6,6) ENF,DFRQ,NTR,IFERR,YGP
    6 FORMAT(1H ,2E12.5,2I5,E12.5)
      WRITE(6,441)         JFILE
  441 FORMAT(25X,'NUMBER OF THIS FILE WRITTEN= ', I10)
      WRITE (6,442)
  442 FORMAT (1H1)
  181 CONTINUE
      STOP
      END
```

<u>Subroutine MDTRHA</u> (Trifunac, Udwadia)

MDTRHA is called by the Volume IV MAIN program to Fourier transform the acceleration data of Volume II and plot it and write it on tape.

<u>Usage</u>

```
CALL MDTRHA (NDATA, M, WO, TITLE2, NT2, TITLE3, NT3,
*       NTR, NSKIP, IEX, ENF, DFRQ, IFERR, YGP)
COMMON TI(8200), ZI(8200), INV(1400), S(1400)
COMMON /BM1/FREQ(4200), AMB(5810)
```

<u>Where</u>

NDATA is the no. of data points of acceleration.

M is the smallest integer such that NDATA is less than 2*2**M.

WO is highest frequency desired in the analysis.

TITLE2 is the title containing earthquake information.

NT2 is the no. of characters in TITLE2.

TITLE3 is the title containing accelerogram information.

NT3 is the no. of characters in TITLE3.

NTR is the no. of plotted frequencies.

NSKIP is the no. of points for decimation in plotting.

IEX is the power of the frequency to be multiplied by the amplitude when plotting. AMP(I) = AMP(I)*FREQ(I)**IEX. Normally IEX = 0.

ENF is the energy integral; integral from 0 to T of ((F(I)/W)**2)DW.

DFRQ is the frequency interval; DFRQ = 2*PI/T, where PI = 3.14159.

IFERR is error return from RHARM. IFERR is 0 unless $M < 3$ or $M > 20$. Then IFERR is 1.

YGP is 95 percent confidence level of amplitude spectrum peaks.

TI is working space for interpolation of acceleration times.

ZI is working space for acceleration interpolation and Fourier coefficients.

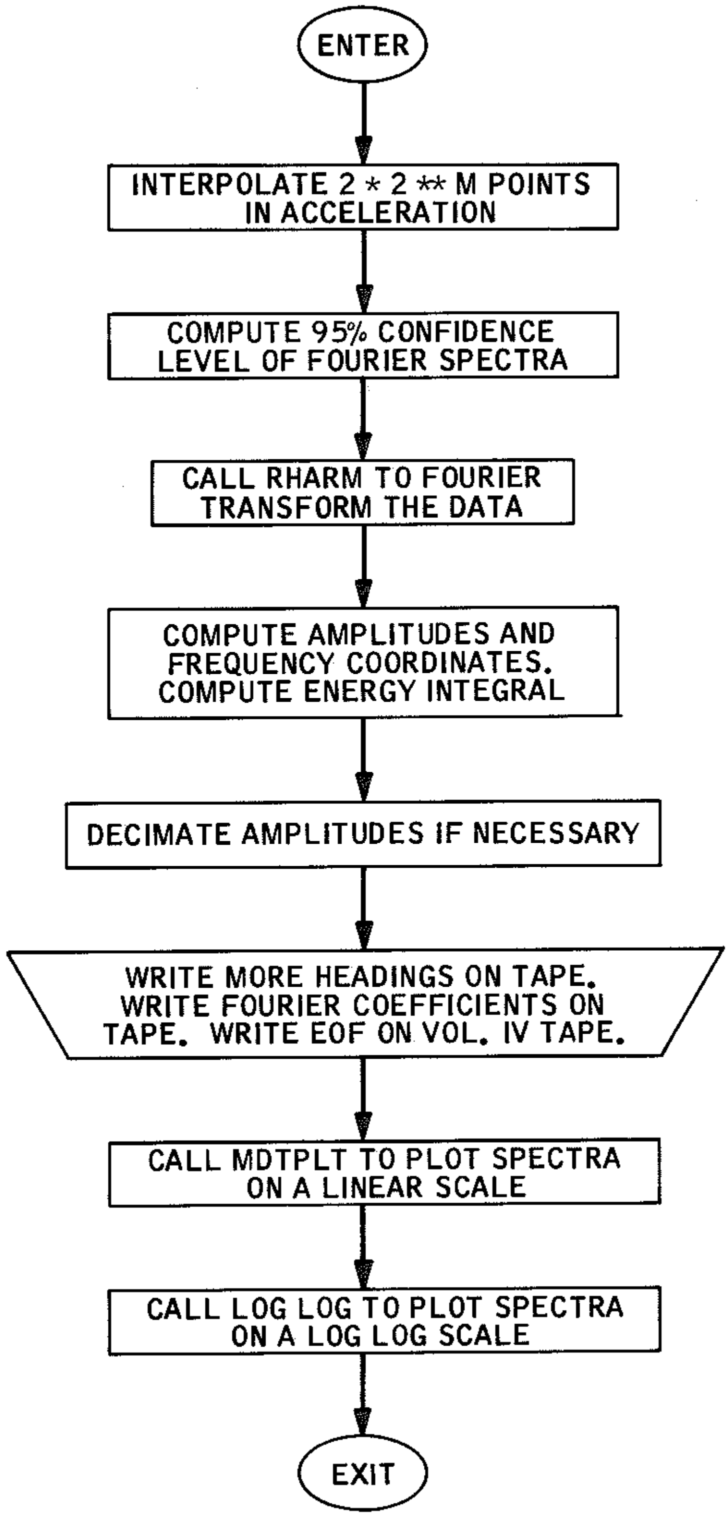
INV is working space for the fast Fourier transform routine.

S is working space for the fast Fourier transform routine.

FREQ is the frequency coordinates for plotting. On entry FREQ has the acceleration times.

AMP is the amplitude coordinates for plotting. On entry AMP(806) and up has the acceleration amplitudes.

MDTRHA FLOW CHART



```
      SUBROUTINE MDTRHA(NDATA,M,WO,TITLE2,NT2,TITLE3,NT3,NTR,NSKIP,IEX,
     1 ENF,DFRQ,IFERR,YGP)
C***********************************************************************
C
C     THIS SUBROUTINE COMPUTES THE FOURIER AMPLITUDE SPECTRA BY USING
C     THE FAST FOURIER ALGORITHM WRITTEN BY COOLY AND TUKEY. MDTRHA
C     CALLS RHARM WHICH CALLS HARM. BOTH RHARM AND HARM ARE CURRENTLY
C     ON THE CALTECH LIBRARY. THE INPUT CONSISTS OF TWO ARRAYS WHICH
C     CONTAIN TIME T(I) AND ACCELERATION Z(I) COORDINATES. EACH ARRAY
C     HAS NDATA POINTS WHICH DO NOT HAVE TO BE EQUALLY SPACED.
C     THE OUTPUT IS A PLOT OF FOURIER AMPLITUDES, SMOOTHED BY 1/4 1/2
C     1/4, VERSUS FREQUENCY OVER THE RANGE FROM ZERO TO 25 CPS. THE PLOT
C     ALSO CONTAINS THE 95 PERCENT CONFIDENCE LEVEL FOR THE PERIODIC
C     CONTENT OF INPUT FUNCTION(A.A.NOWROOZI(1967). TABLE FOR FISHER'S
C     TEST OF SIGNIFICANCE IN HARMONIC ANALYSIS,GEOPHYS. J. R. ASTR. SOC
C     12,517-520)
C
C***********************************************************************
C
C     NDATA=NO. OF DATA POINTS IN T(I) AND Z(I)
C     M - MDTRHA INTERPOLATES 2*2**M EQUALLY SPACED POINTS FROM Z(I).
C     M MUST BE <11
C     TITLE2=TITLE CONTAINING EARTHQUAKE INFORMATION
C     NT2=NO. OF ALPHANUMERIC CHARACTERS IN TITLE2
C     TITLE3=TITLE CONTAINING ACCELEROGRAM INFORMATION
C     NT3=NO. OF ALPHANUMERIC CHARACTERS IN TITLE3
C     WO=HIGHEST FREQUENCY TO BE PLOTTED. FOR THE STANDARD PLOT WO MUST
C     BE LESS THAN 25.0*6.28
C     NFR=NO. OF PLOTTED FREQUENCIES
C     NSKIP=NORMALLY NSKIP=1, WHEN GREATER THAN 1 ONLY EVERY NSKIP'TH
C     POINT IN SPECTRUM IS PLOTTED
C     IEX - BEFORE PLOTTING THE SPECTRUM IS MULTIPLIED BY FREQ(I)**IEX
C     IEX IS NORMALLY SET TO ZERO
C     ENF=ENERGY INTEGRAL: INTEGRAL FROM 0 TO T OF (F(W)/W)**2*DW
C     DFRQ=FREQUENCY INTERVAL DFRQ=2*3.14/T
C     IFERR=ERROR RETURN FROM RHARM. IFERR IS SET 0 BY THE ROUTINE
C     UNLESS M<3 OR M>20 IN WHICH CASE IFERR=1
C     YGP- ALL SPECTRAL PEAKS WHICH HAVE AMPLITUDES ABOVE THE YGP LEVEL
C     ARE SIGNIFICANT AT 95 PERCENT CONFIDENCE LEVEL
C     PLOTTING HINT: SPECTRAL PEACKS SHOULD BE SMALLER THAN THREE DIGIT
C     NUMBER 999. IF NOT THE ERROR MESSAGE WILL BERETURNED FROM PLOTTING
C     ROUTINE.
C
       DIMENSION SPN(6),FRS(5),ENFL(5),UNITS(3),   YGLN(7),UN(1),UNI(4)
       DIMENSION TL(8)       ,TOT(1)
      COMMON TI(8200),ZI(8200),INV(1400),
     1S(1400)
      COMMON / BM1/ FREQ(4200),AMB(5810)
      DIMENSION Z(5005)
      DIMENSION T(5005)
      EQUIVALENCE (T(1),FREQ(1))
      EQUIVALENCE (Z(1),AMB(806))
      EQUIVALENCE (AMP1,IAMP1),(AMP2,IAMP2),(AMP3,IAMP3)
      DIMENSION HTTL(4),VTTL(9),TITLE1(11),TITLE2(40),TITLE3(40)
       DATA SPN/'NUMB','ER O','F FR','EQUE','NCIE','S = '/
       DATA FRS/'FREQ','UENC','Y SP','ACIN','G = '/
       DATA UN/'CPS.'/
        DATA ENFL/'ENER','GY I','N SI','GNAL',' =  '/
       DATA UNI/'CMS.',' PER',' CYC','LE  '/
       DATA UNITS/'CMS.',' PER',' SEC'/
       DATA YGLN/'VALU','E OF','CONF','IDEN','CE L','EVEL',' =  '/
```

```
      DATA TL/'TIME',' LEN','GTH ','OF S','IGNA','L AN','ALYS','ED ='/
      DATA TOT/'SECS'/
      DATA TITLE1/'FOUR','IER ','AMPL','ITUD','E SP','ECTR','UM O','F AC
     1','CELE','RATI','ON  '/
      DATA HTTL/'FREQ','UENC','Y - ','CPS '/
      DATA VTTL/'FOUR','IER ','AMPL','ITUD','E SP','ECTR','UM -',' CM/',
     1'SEC '/
      DATA  NT1/42/,LH/15/,LV/35/
      REAL*8 TIME
      NPOINT=NDATA-1
      NTOT=2**M
      ATOT=NTOT
      NTOT2=2*NTOT
      ATOT2=NTOT2 -1
      TINT=T(NPOINT)-T(1)
      TRIF=TINT/2.
      DDT=TINT/ATOT2
      TIME=DDT
      TI(1)=T(1)
      ZI(1)=Z(1)
      I=2
      J=1
  382 TI(I)=T(1)+TIME
  383 IF (TI(I)-T(J+1)) 386,384,384
  384 J=J+1
      IF (J-NPOINT) 383,383,385
  386 CONTINUE
      ZI(I)=Z(J)+(Z(J+1)-Z(J))*((TI(I)-T(J))/(T(J+1)-T(J)))
      I=I+1
      TIME=TIME+DDT
      IF (J-NPOINT) 382,382,385
  385 IKRAJ=I-1
      ZI(1)=(ZI(1)+ZI(IKRAJ))/2.
      ZI(IKRAJ)=ZI(1)
C     COMPUTING 95 PERCENT CONFIDENCE LEVEL
      SUM=0.
      NEND=NTOT2
      ANT=NEND
      DO 75 I=1,NEND
   75 SUM=SUM+ZI(I)*ZI(I)
      AM=NTOT
      AM1=1./(AM-1.)
      GG=1.- (0.05/AM)**AM1
      YGP2=(GG*2.0/ANT)*SUM
      YGP=SQRT(YGP2)   *TRIF
      CALL RHARM(ZI,M,INV,S,IFERR)
      ENSP=0.
      NNTT=NTOT-1
      AMP1=SQRT(ZI(2)**2+ZI(1)**2)*TRIF
      AMP2=SQRT(ZI(4)**2+ZI(3)**2)*TRIF
      AMB(1)=AMP1
      FREQ(1)=0.0
      PI=3.1415926535
      TRIFIN=PI/TRIF
      DFRQ=1.0/TINT
      DO 64 J=2,NNTT
      JJ=J+1
      AMP3=SQRT(ZI(2*JJ)**2+ZI(2*JJ-1)**2)*TRIF
      AMB(J)=0.5*AMP2+0.25*(AMP1+AMP3)
      FREQ(J)=(J-1)*TRIFIN
      ENSP=ENSP +(AMB(J)/FREQ(J) )**2
      AMP1=AMP2
```

```
      AMP2=AMP3
   64 CONTINUE
      ENF=ENSP*FREQ(2)
      NTOD=NTOT
      AMB(NTOD)=AMP3
      FREQ(NTOD)=NNTT*TRIFIN
      NTK=NTOT
      NTR=NTK*(WO/FREQ(NTK))
      IF(NTR .GE. NTK) NTR=NTK
      TWOPIN=1.0/(2*PI)
      K=1
      NST=2
      DO 67 I=NST,NTOT,NSKIP
      AMB(K)=AMB(I) *FREQ(I)**IEX
      FREQ(K)=FREQ(I)*TWOPIN
      K=K+1
   67 CONTINUE
      NTJ=K-1
       NNNN=2*NTR
       NTQ =NTR
       DO 667 KO = 1,NNNN
       ZI(KO) = ZI(KO)*TRIF
  667  CONTINUE
       WRITE(50) TL,TINT,TOT
       WRITE(50) SPN,NTQ,FRS,DFRQ,UN,ENFL,ENF,UNI,YGLN,YGP,UNITS
       WRITE(50)  (ZI(KO),KO = 1,NNNN)
       CALL WRTNF(50)
  442  FORMAT(10(1X,F11.3))
      IF (NDATA.EQ.0) RETURN
      CALL MDTPLT(NTJ,TITLE1,NT1,TITLE2,NT2,TITLE3,NT3,HTTL,LH,VTTL,LV,
     1 YGP)
      CALL LOGLOG(NTQ,TITLE1,NT1,TITLE2,NT2,TITLE3,NT3,HTTL,LH,VTTL,LV)
      RETURN
      END
```

Subroutine RHARM*

RHARM is called by MDTRHA to calculate the Fourier coefficients $a_o/2$, $b_o=0, a_1, b_1, \ldots, a_N/2, b_N=0$ where $N = 2^M$.

For computational efficiency, RHARM treats the real input as a pseudo complex function (the odd points forming the real part; the even ones, the imaginary part) and uses a subprogram HARM to evaluate the complex coefficients.

$$A_k = \frac{1}{N} \sum_{\nu=0}^{N-1} (x_{2\nu} + ix_{2\nu+1})e^{-2\pi i\nu k/N}, \quad k = 0, 1, \ldots, N-1.$$

Usage

CALL RHARM (A, M, INV, S, IERR)

Where

A is input, dimensioned at least 2N+4 where $N=2^M$. The values of 2N equispaced samples $x(0), \ldots x[(2N-1)T]$ are placed in the array A(1), A(2), ..., A(2N) (the input wave form is assumed periodic of period 2NT). As output, A contains the Fourier coefficients $a_o/2$, $b_o=0, a_1, b_1, \ldots, a_N/2, b_N=0$ in A(1), A(2), ..., A(2N+2) respectively.

M is an integer such that $N=2^M$.

---

* This description of RHARM has been extracted from the DFT report C268-239-370 issued by the Booth Computing Center at Caltech (October, 1971). It is based on the "System/360 Scientific Subroutine Package (360A-CM-03X) Version III".

INV and S are vector work areas each of dimension N/4 or 4, whichever is greater.

IERR is set to 0 by the routine unless $M < 3$ or $M > 20$, in which case, IERR = 1.

```
      SUBROUTINE RHARM(A,M,INV,S,IFERR)
C    RHARM DATE 05-30-73
C
C     ..................................................................
C
C        SUBROUTINE RHARM
C
C        PURPOSE
C           FINDS THE FOURIER COEFFICIENTS OF ONE DIMENSIONAL REAL DATA
C
C        USAGE
C           CALL RHARM (A,M,INV,S,IFERR)
C
C        DESCRIPTION OF PARAMETERS
C           A      - AS INPUT, CONTAINS ONE DIMENSIONAL REAL DATA. A IS
C                    2*N+4 CORE LOCATIONS, WHERE N = 2**M. 2*N REAL
C                    NUMBERS ARE PUT INTO THE FIRST 2*N CORE LOCATIONS
C                    OF A
C                    AS OUTPUT, A CONTAINS THE FOURIER COEFFICIENTS
C                    A0/2,B0=0,A1,B1,A2,B2,...,AN/2,BN=0 RESPECTIVELY IN
C                    THE FIRST 2N+2 CORE LOCATIONS OF A
C           M      - AN INTEGER WHICH DETERMINES THE SIZE OF THE VECTOR
C                    A. THE SIZE OF A IS 2*(2**M) + 4
C           INV    - A VECTOR WORK AREA FOR BIT AND INDEX MANIPULATION OF
C                    DIMENSION ONE EIGHTH THE NUMBER OF REAL INPUT, VIZ.,
C                    (1/8)*2*(2**M),.GE.4   (MINIMUM FOR M=3 IS 4)
C           S      - A VECTOR WORK AREA FOR SINE TABLES WITH DIMENSION
C                    THE SAME AS INV
C           IFERR  - A RETURNED VALUE OF 1 MEANS THAT M IS LESS THAN 3 OR
C                    GREATER THAN 20. OTHERWISE IFERR IS SET = 0
C
C        REMARKS
C           THIS SUBROUTINE GIVES THE FOURIER COEFFICIENTS OF 2*(2**M)
C           REAL POINTS. SEE SUBROUTINE HARM FOR THREE DIMENSIONAL,
C           COMPLEX FOURIER TRANSFORMS
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           HARM
C
C        METHOD
C           THE FOURIER COEFFICIENTS A0,B0=0,A1,B1,...,AN,BN=0 ARE
C           OBTAINED FOR INPUT XJ, J=0,1,2,...,2N-1 FOR THE FOLLOWING
C           EQUATION (PI = 3.14159...)
C
C                    N-1                                              J
C     XJ=(1/2)A0+SUM (AK*COS(PI*J*K/N)+BK*SIN(PI*J*K/N))+(1/2)AN(-1)
C                    K=1
C
C        SEE REFERENCE UNDER SUBROUTINE HARM
C
C     ..................................................................
C
      DIMENSION A(1),L(3),INV(1),S(1)
      IFSET=1
      L(1)=M
      L(2)=0
      L(3)=0
      NTOT=2**M
      NTOT2 = 2*NTOT
      FN = NTOT
      DO   3 I = 2,NTOT2,2
```

```
    3  A(I) = -A(I)
       DO   6 I = 1,NTOT2
    6  A(I) = A(I)/FN
       CALL HARM(A,L,INV,S ,IFSET,IFERR)
C
C      MOVE LAST HALF OF A(J)S DOWN ONE SLOT AND ADD A(N) AT BOTTOM TO
C      GIVE ARRAY FOR A1PRIME AND A2PRIME CALCULATION
C
   21  DO  52 I=1,NTOT,2
       J0=NTOT2+2-I
       A(J0)=A(J0-2)
   52  A(J0+1)=A(J0-1)
       A(NTOT2+3)=A(1)
       A(NTOT2+4)=A(2)
C
C      CALCULATE A1PRIMES AND STORE IN FIRST N SLOTS
C      CALCULATE A2PRIMES AND STORE IN SECOND N SLOTS IN REVERSE ORDER
       K0=NTOT+1
       DO 104 I=1,K0,2
       K1=NTOT2-I+4
       AP1RE=.5*(A(I)+A(K1))
       AP2RE=-.5*(A(I+1)+A(K1+1))
       AP1IM=.5*(-A(I+1)+A(K1+1))
       AP2IM=-.5*(A(I)-A(K1))
       A(I)=AP1RE
       A(I+1)=AP1IM
       A(K1)=AP2RE
  104  A(K1+1)=AP2IM
       NTO = NTOT/2
  110  NT=NTO+1
       DEL=3.1415927/FLOAT(NTOT)
       SS=SIN(DEL)
       SC=COS(DEL)
       SI=0.0
       CO=1.0
C
C      COMPUTE C(J)S FOR J=0 THRU J=N
  114  DO 116 I=1,NT
       K6=NTOT2-2*I+5
       AP2RE=A(K6)*CO+A(K6+1)*SI
       AP2IM=-A(K6)*SI+A(K6+1)*CO
       C1RE=.5*(A(2*I-1)+AP2RE)
       C1IM=.5*(A(2*I)+AP2IM)
       CNIRE=.5*(A(2*I-1)-AP2RE)
       CNIIM=.5*(A(2*I)-AP2IM)
       A(2*I-1)=C1RE
       A(2*I)=C1IM
       A(K6)=CNIRE
       A(K6+1)=-CNIIM
       SIS=SI
       SI=SI*SC+CO*SS
  116  CO=CO*SC-SIS*SS
C
C      SHIFT C(J)S FOR J=N/2+1 TO J=N UP ONE SLOT
       DO 117 I=1,NTOT,2
       K8=NTOT+4+I
       A(K8-2)=A(K8)
  117  A(K8-1)=A(K8+1)
       DO 500 I=3,NTOT2,2
       A(I) = 2. * A(I)
  500  A(I + 1) = -2. * A(I + 1)
       RETURN
```

END RHAR0123

```
      SUBROUTINE HARM(A,M,INV,S,IFSET, IFERR)
C     HARM DATE 05-30-73
C
C     ...............................................................
C
C        SUBROUTINE HARM
C
C        PURPOSE
C           PERFORMS DISCRETE COMPLEX FOURIER TRANSFORMS ON A COMPLEX
C           THREE DIMENSIONAL ARRAY
C
C        USAGE
C           CALL HARM (A,M,INV,S,IFSET,IFERR)
C
C        DESCRIPTION OF PARAMETERS
C           A      - AS INPUT, A CONTAINS THE COMPLEX, 3-DIMENSIONAL
C                    ARRAY TO BE TRANSFORMED.  THE REAL PART OF
C                    A(I1,I2,I3) IS STORED IN VECTOR FASHION IN A CELL
C                    WITH INDEX 2*(I3*N1*N2 + I2*N1 + I1) + 1 WHERE
C                    NI = 2**M(I), I=1,2,3 AND I1 = 0,1,...,N1-1 ETC.
C                    THE IMAGINARY PART IS IN THE CELL IMMEDIATELY
C                    FOLLOWING.  NOTE THAT THE SUBSCRIPT I1 INCREASES
C                    MOST RAPIDLY AND I3 INCREASES LEAST RAPIDLY.
C                    AS OUTPUT, A CONTAINS THE COMPLEX FOURIER
C                    TRANSFORM.  THE NUMBER OF CORE LOCATIONS OF
C                    ARRAY A IS 2*(N1*N2*N3)
C           M      - A THREE CELL VECTOR WHICH DETERMINES THE SIZES
C                    OF THE 3 DIMENSIONS OF THE ARRAY A.   THE SIZE,
C                    NI, OF THE I DIMENSION OF A IS 2**M(I), I = 1,2,3
C           INV    - A VECTOR WORK AREA FOR BIT AND INDEX MANIPULATION
C                    OF DIMENSION ONE EIGHTH THE NUMBER OF CORE
C                    LOCATIONS OF A, VIZ., (1/8)*2*N1*N2*N3
C           S      - A VECTOR WORK AREA FOR SINE TABLES WITH DIMENSION
C                    THE SAME AS INV
C           IFSET  - AN OPTION PARAMETER WITH THE FOLLOWING SETTINGS
C                       0     SET UP SINE AND INV TABLES ONLY
C                       1     SET UP SINE AND INV TABLES ONLY AND
C                             CALCULATE FOURIER TRANSFORM
C                      -1     SET UP SINE AND INV TABLES ONLY AND
C                             CALCULATE INVERSE FOURIER TRANSFORM (FOR
C                             THE MEANING OF INVERSE SEE THE EQUATIONS
C                             UNDER METHOD BELOW)
C                       2     CALCULATE FOURIER TRANSFORM ONLY (ASSUME
C                             SINE AND INV TABLES EXIST)
C                      -2     CALCULATE INVERSE FOURIER TRANSFORM ONLY
C                             (ASSUME SINE AND INV TABLES EXIST)
C           IFERR  - ERROR INDICATOR.   WHEN IFSET IS 0,+1,-1,
C                    IFERR = 1 MEANS THE MAXIMUM M(I) IS GREATER THAN
C                  20 , I=1,2,3   WHEN IFSET IS 2,-2 , IFERR = 1
C                    MEANS THAT THE SINE AND INV TABLES ARE NOT LARGE
C                    ENOUGH OR HAVE NOT BEEN COMPUTED .
C                    IF ON RETURN IFERR = 0 THEN NONE OF THE ABOVE
C                    CONDITIONS ARE PRESENT
C
C        REMARKS
C           THIS SUBROUTINE IS TO BE USED FOR COMPLEX, 3-DIMENSIONAL
C           ARRAYS IN WHICH EACH DIMENSION IS A POWER OF 2.  THE
C           MAXIMUM M(I) MUST NOT BE LESS THAN 3 OR GREATER THAN 20,
C           I = 1,2,3
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
```

```
C           NONE
C
C        METHOD
C           FOR IFSET = +1, OR +2, THE FOURIER TRANSFORM OF COMPLEX
C           ARRAY A IS OBTAINED.
C
C                    N1-1    N2-1    N3-1                    L1    L2    L3
C       X(J1,J2,J3)=SUM     SUM     SUM    A(K1,K2,K3)*W1   *W2   *W3
C                    K1=0    K2=0    K3=0
C
C                    WHERE WI IS THE N(I) ROOT OF UNITY AND L1=K1*J1,
C                          L2=K2*J2, L3=K3*J3
C
C
C           FOR IFSET = -1, OR -2, THE INVERSE FOURIER TRANSFORM A OF
C           COMPLEX ARRAY X IS OBTAINED.
C
C       A(K1,K2,K3)=
C                 1       N1-1    N2-1    N3-1                   -L1   -L2   -L3
C           -------- *SUM     SUM     SUM    X(J1,J2,J3)*W1   *W2   *W3
C           N1*N2*N3    J1=0    J2=0    J3=0
C
C
C           SEE J.W. COOLEY AND J.W. TUKEY, 'AN ALGORITHM FOR THE
C           MACHINE CALCULATION OF COMPLEX FOURIER SERIES',
C           MATHEMATICS OF COMPUTATIONS, VOL. 19 (APR. 1965), P. 297.
C
C       ..................................................................
C
      DIMENSION A(1),INV(1),S(1),N(3),M(3),NP(3),W(2),W2(2),W3(2)
      EQUIVALENCE (N1,N(1)),(N2,N(2)),(N3,N(3))
   10 IF( IABS(IFSET) - 1) 900,900,12
   12 MTT=MAX0(M(1),M(2),M(3)) -2
      ROOT2 = SQRT(2.)
      IF (MTT-MT ) 14,14,13
   13 IFERR=1
      RETURN
   14 IFERR=0
      M1=M(1)
      M2=M(2)
      M3=M(3)
      N1=2**M1
      N2=2**M2
      N3=2**M3
   16 IF(IFSET) 18,18,20
   18 NX= N1*N2*N3
      FN = NX
      DO 19 I = 1,NX
      A(2*I-1) = A(2*I-1)/FN
   19 A(2*I) = -A(2*I)/FN
   20 NP(1)=N1*2
      NP(2)= NP(1)*N2
      NP(3)=NP(2)*N3
      DO 250 ID=1,3
      IL = NP(3)-NP(ID)
      IL1 = IL+1
      MI = M(ID)
      IF (MI)250,250,30
   30 IDIF=NP(ID)
      KBIT=NP(ID)
      MEV = 2*(MI/2)
      IF (MI - MEV )60,60,40
```

```
C
C     M IS ODD. DO L=1 CASE
   40 KBIT=KBIT/2
      KL=KBIT-2
      DO 50 I=1,IL1,IDIF
      KLAST=KL+I
      DO 50 K=I,KLAST,2
      KD=K+KBIT
C
C     DO ONE STEP WITH L=1,J=0
C     A(K)=A(K)+A(KD)
C     A(KD)=A(K)-A(KD)
C
      T=A(KD)
      A(KD)=A(K)-T
      A(K)=A(K)+T
      T=A(KD+1)
      A(KD+1)=A(K+1)-T
   50 A(K+1)=A(K+1)+T
      IF (MI - 1)250,250,52
   52 LFIRST =3
C
C     DEF - JLAST = 2**(L-2) -1
      JLAST=1
      GO TO 70
C
C     M IS EVEN
   60 LFIRST = 2
      JLAST=0
   70 DO 240 L=LFIRST,MI,2
      JJDIF=KBIT
      KBIT=KBIT/4
      KL=KBIT-2
C
C     DO FOR J=0
      DO 80 I=1,IL1,IDIF
      KLAST=I+KL
      DO 80 K=I,KLAST,2
      K1=K+KBIT
      K2=K1+KBIT
      K3=K2+KBIT
C
C     DO TWO STEPS WITH J=0
C     A(K)=A(K)+A(K2)
C     A(K2)=A(K)-A(K2)
C     A(K1)=A(K1)+A(K3)
C     A(K3)=A(K1)-A(K3)
C
C     A(K)=A(K)+A(K1)
C     A(K1)=A(K)-A(K1)
C     A(K2)=A(K2)+A(K3)*I
C     A(K3)=A(K2)-A(K3)*I
C
      T=A(K2)
      A(K2)=A(K)-T
      A(K)=A(K)+T
      T=A(K2+1)
      A(K2+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
C
      T=A(K3)
      A(K3)=A(K1)-T
```

```
      A(K1)=A(K1)+T                                                     HARM0185
      T=A(K3+1)                                                         HARM0186
      A(K3+1)=A(K1+1)-T                                                 HARM0187
      A(K1+1)=A(K1+1)+T                                                 HARM0188
C                                                                       HARM0189
      T=A(K1)                                                           HARM0190
      A(K1)=A(K)-T                                                      HARM0191
      A(K)=A(K)+T                                                       HARM0192
      T=A(K1+1)                                                         HARM0193
      A(K1+1)=A(K+1)-T                                                  HARM0194
      A(K+1)=A(K+1)+T                                                   HARM0195
C                                                                       HARM0196
      R=-A(K3+1)                                                        HARM0197
      T = A(K3)                                                         HARM0198
      A(K3)=A(K2)-R                                                     HARM0199
      A(K2)=A(K2)+R                                                     HARM0200
      A(K3+1)=A(K2+1)-T                                                 HARM0201
   80 A(K2+1)=A(K2+1)+T                                                 HARM0202
      IF (JLAST) 235,235,82                                             HARM0203
   82 JJ=JJDIF   +1                                                     HARM0204
C                                                                       HARM0205
C     DO FOR J=1                                                        HARM0206
      ILAST= IL +JJ                                                     HARM0207
      DO 85 I = JJ,ILAST,IDIF                                           HARM0208
      KLAST = KL+I                                                      HARM0209
      DO 85 K=I,KLAST,2                                                 HARM0210
      K1 = K+KBIT                                                       HARM0211
      K2 = K1+KBIT                                                      HARM0212
      K3 = K2+KBIT                                                      HARM0213
C                                                                       HARM0214
C     LETTING W=(1+I)/ROOT2,W3=(-1+I)/ROOT2,W2=I,                       HARM0215
C     A(K)=A(K)+A(K2)*I                                                 HARM0216
C     A(K2)=A(K)-A(K2)*I                                                HARM0217
C     A(K1)=A(K1)*W+A(K3)*W3                                            HARM0218
C     A(K3)=A(K1)*W-A(K3)*W3                                            HARM0219
C                                                                       HARM0220
C     A(K)=A(K)+A(K1)                                                   HARM0221
C     A(K1)=A(K)-A(K1)                                                  HARM0222
C     A(K2)=A(K2)+A(K3)*I                                               HARM0223
C     A(K3)=A(K2)-A(K3)*I                                               HARM0224
C                                                                       HARM0225
      R =-A(K2+1)                                                       HARM0226
      T = A(K2)                                                         HARM0227
      A(K2) = A(K)-R                                                    HARM0228
      A(K) = A(K)+R                                                     HARM0229
      A(K2+1)=A(K+1)-T                                                  HARM0230
      A(K+1)=A(K+1)+T                                                   HARM0231
C                                                                       HARM0232
      AWR=A(K1)-A(K1+1)                                                 HARM0233
      AWI = A(K1+1)+A(K1)                                               HARM0234
      R=-A(K3)-A(K3+1)                                                  HARM0235
      T=A(K3)-A(K3+1)                                                   HARM0236
      A(K3)=(AWR-R)/ROOT2                                               HARM0237
      A(K3+1)=(AWI-T)/ROOT2                                             HARM0238
      A(K1)=(AWR+R)/ROOT2                                               HARM0239
      A(K1+1)=(AWI+T)/ROOT2                                             HARM0240
      T= A(K1)                                                          HARM0241
      A(K1)=A(K)-T                                                      HARM0242
      A(K)=A(K)+T                                                       HARM0243
      T=A(K1+1)                                                         HARM0244
      A(K1+1)=A(K+1)-T                                                  HARM0245
      A(K+1)=A(K+1)+T                                                   HARM0246
```

```
      R=-A(K3+1)
      T=A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
   85 A(K2+1)=A(K2+1)+T
      IF(JLAST-1) 235,235,90
   90 JJ= JJ + JJDIF
C
C     NOW DO THE REMAINING J'S
      DO 230 J=2,JLAST
C
C     FETCH W'S
C     DEF- W=W**INV(J), W2=W**2, W3=W**3
   96 I=INV(J+1)
   98 IC=NT-I
      W(1)=S(IC)
      W(2)=S(I)
      I2=2*I
      I2C=NT-I2
      IF(I2C)120,110,100
C
C     2*I IS IN FIRST QUADRANT
  100 W2(1)=S(I2C)
      W2(2)=S(I2)
      GO TO 130
  110 W2(1)=0.
      W2(2)=1.
      GO TO 130
C
C     2*I IS IN SECOND QUADRANT
  120 I2CC = I2C+NT
      I2C=-I2C
      W2(1)=-S(I2C)
      W2(2)=S(I2CC)
  130 I3=I+I2
      I3C=NT-I3
      IF(I3C)160,150,140
C
C     I3 IN FIRST QUADRANT
  140 W3(1)=S(I3C)
      W3(2)=S(I3)
      GO TO 200
  150 W3(1)=0.
      W3(2)=1.
      GO TO 200
C
  160 I3CC=I3C+NT
      IF(I3CC)190,180,170
C
C     I3 IN SECOND QUADRANT
  170 I3C=-I3C
      W3(1)=-S(I3C)
      W3(2)=S(I3CC)
      GO TO 200
  180 W3(1)=-1.
      W3(2)=0.
      GO TO 200
C
C     3*I IN THIRD QUADRANT
  190 I3CCC=NT+I3CC
      I3CC = -I3CC
```

```
      W3(1)=-S(I3CCC)
      W3(2)=-S(I3CC)
  200 ILAST=IL+JJ
      DO 220 I=JJ,ILAST,IDIF
      KLAST=KL+I
      DO 220 K=I,KLAST,2
      K1=K+KBIT
      K2=K1+KBIT
      K3=K2+KBIT
C
C     DO TWO STEPS WITH J NOT 0
C     A(K)=A(K)+A(K2)*W2
C     A(K2)=A(K)-A(K2)*W2
C     A(K1)=A(K1)*W+A(K3)*W3
C     A(K3)=A(K1)*W-A(K3)*W3
C
C     A(K)=A(K)+A(K1)
C     A(K1)=A(K)-A(K1)
C     A(K2)=A(K2)+A(K3)*I
C     A(K3)=A(K2)-A(K3)*I
C
      R=A(K2)*W2(1)-A(K2+1)*W2(2)
      T=A(K2)*W2(2)+A(K2+1)*W2(1)
      A(K2)=A(K)-R
      A(K)=A(K)+R
      A(K2+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
C
      R=A(K3)*W3(1)-A(K3+1)*W3(2)
      T=A(K3)*W3(2)+A(K3+1)*W3(1)
      AWR=A(K1)*W(1)-A(K1+1)*W(2)
      AWI=A(K1)*W(2)+A(K1+1)*W(1)
      A(K3)=AWR-R
      A(K3+1)=AWI-T
      A(K1)=AWR+R
      A(K1+1)=AWI+T
      T=A(K1)
      A(K1)=A(K)-T
      A(K)=A(K)+T
      T=A(K1+1)
      A(K1+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
      R=-A(K3+1)
      T=A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
  220 A(K2+1)=A(K2+1)+T
C     END OF I AND K LOOPS
C
  230 JJ=JJDIF+JJ
C     END OF J-LOOP
C
  235 JLAST=4*JLAST+3
  240 CONTINUE
C     END OF  L  LOOP
C
  250 CONTINUE
C     END OF  ID  LOOP
C
C     WE NOW HAVE THE COMPLEX FOURIER SUMS BUT THEIR ADDRESSES ARE
C     BIT-REVERSED.  THE FOLLOWING ROUTINE PUTS THEM IN ORDER
```

```
      NTSQ=NT*NT
      M3MT=M3-MT
  350 IF(M3MT) 370,360,360
C
C     M3 GR. OR EQ. MT
  360 IGO3=1
      N3VNT=N3/NT
      MINN3=NT
      GO TO 380
C
C     M3 LESS THAN MT
  370 IGO3=2
      N3VNT=1
      NTVN3=NT/N3
      MINN3=N3
  380 JJD3 = NTSQ/N3
      M2MT=M2-MT
  450 IF (M2MT)470,460,460
C
C     M2 GR. OR EQ. MT
  460 IGO2=1
      N2VNT=N2/NT
      MINN2=NT
      GO TO 480
C
C     M2 LESS THAN MT
  470 IGO2 = 2
      N2VNT=1
      NTVN2=NT/N2
      MINN2=N2
  480 JJD2=NTSQ/N2
      M1MT=M1-MT
  550 IF(M1MT)570,560,560
C
C     M1 GR. OR EQ. MT
  560 IGO1=1
      N1VNT=N1/NT
      MINN1=NT
      GO TO 580
C
C     M1 LESS THAN MT
  570 IGO1=2
      N1VNT=1
      NTVN1=NT/N1
      MINN1=N1
  580 JJD1=NTSQ/N1
  600 JJ3=1
      J=1
      DO 880 JPP3=1,N3VNT
      IPP3=INV(JJ3)
      DO 870 JP3=1,MINN3
      GO TO (610,620),IGO3
  610 IP3=INV(JP3)*N3VNT
      GO TO 630
  620 IP3=INV(JP3)/NTVN3
  630 I3=(IPP3+IP3)*N2
  700 JJ2=1
      DO 870 JPP2=1,N2VNT
      IPP2=INV(JJ2)+I3
      DO 860 JP2=1,MINN2
      GO TO (710,720),IGO2
  710 IP2=INV(JP2)*N2VNT
```

```
      GO TO 730
  720 IP2=INV(JP2)/NTVN2
  730 I2=(IPP2+IP2)*N1
  800 JJ1=1
      DO 860 JPP1=1,N1VNT
      IPP1=INV(JJ1)+I2
      DO 850 JP1=1,MINN1
      GO TO (810,820),IGO1
  810 IP1=INV(JP1)*N1VNT
      GO TO 830
  820 IP1=INV(JP1)/NTVN1
  830 I=2*(IPP1+IP1)+1
      IF (J-I) 840,845,845
  840 T=A(I)
      A(I)=A(J)
      A(J)=T
      T=A(I+1)
      A(I+1)=A(J+1)
      A(J+1)=T
  845 CONTINUE
  850 J=J+2
  860 JJ1=JJ1+JJD1
C     END OF JPP1 AND JP2
C
  870 JJ2=JJ2+JJD2
C     END OF JPP2 AND JP3 LOOPS
C
  880 JJ3 = JJ3+JJD3
C     END OF JPP3 LOOP
C
  890 IF(IFSET)891,895,895
  891 DO 892 I = 1,NX
  892 A(2*I) = -A(2*I)
  895 RETURN
C
C     THE FOLLOWING PROGRAM COMPUTES THE SIN AND INV TABLES.
C
  900 MT=MAX0(M(1),M(2),M(3)) -2
      MT = MAX0(2,MT)
 904   IF (MT-18) 906,906,905
  905 IFERR = 1
      GO TO 895
  906 IFERR=0
      NT=2**MT
      NTV2=NT/2
C
C     SET UP SIN TABLE
C     THETA=PIE/2**(L+1) FOR L=1
  910 THETA=.7853981634
C
C     JSTEP=2**(MT-L+1) FOR L=1
      JSTEP=NT
C
C     JDIF=2**(MT-L) FOR L=1
C     JDIF=2**(MT-L) FOR L=1
      JDIF=NTV2
      S(JDIF)=SIN(THETA)
      DO 950 L=2,MT
      THETA=THETA/2.
      JSTEP2=JSTEP
      JSTEP=JDIF
      JDIF=JSTEP/2
```

```
      S(JDIF)=SIN(THETA)
      JC1=NT-JDIF
      S(JC1)=COS(THETA)
      JLAST=NT-JSTEP2
      IF(JLAST - JSTEP) 950,920,920
  920 DO 940 J=JSTEP,JLAST,JSTEP
      JC=NT-J
      JD=J+JDIF
  940 S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)
  950 CONTINUE
C
C     SET UP INV(J) TABLE
C
  960 MTLEXP=NTV2
C
C     MTLEXP=2**(MT-L). FOR L=1
      LM1EXP=1
C
C     LM1EXP=2**(L-1). FOR L=1
      INV(1)=0
      DO 980 L=1,MT
      INV(LM1EXP+1) = MTLEXP
      DO 970 J=2,LM1EXP
      JJ=J+LM1EXP
  970 INV(JJ)=INV(J)+MTLEXP
      MTLEXP=MTLEXP/2
  980 LM1EXP=LM1EXP*2
  982 IF(IFSET)12,995,12
  995 RETURN
      END
```

Subroutine MDTPLT (Trifunac)

MDTPLT is used in MDTRHA to produce a linear plot of Fourier Amplitude Spectra. Also it plots a dashed level line at the value of the 95% confidence level.

Usage

```
CALL MDTPLT (N,TTL1,L1,TTL2,L2,TTL3,L3,HTTL,LH,
*       VTTL,LV,YGP)
COMMON /BM1/X(4200),Y(4200)
COMMON /COMPLO/ITEST,XLNG,YLNG
```

Where

N is the no. of points to be plotted.

TTL1 is the plot title.

L1 is the no. of characters in TTL1.

TTL2 is the earthquake title.

L2 is the no. of characters in TTL2.

TTL3 is the accelerogram title.

L3 is the no. of characters in TTL3.

HTTL is the horizontal axis title.

LH is the no. of characters in HTTL.

VTTL is the vertical axis title.

LV is the no. of characters in VTTL.

YGP is the 95% confidence level.

COMMON /BM1/ has:

X is the frequency coordinates. 25 cps is the maximum.

Y is the amplitude coordinate.

COMMON /COMPLO/ is used by XYPLOT as described in the Caltech write-up.

Subroutine SQBOUN is used in MDTPLT to draw a square boundary around the plot. SQBOUN uses a Caltech plotting routine, SYSPLT, to draw the lines. SQBOUN is used in LOGLOG also.

Subroutine XYPLOT

XYPLOT is the Caltech array-plotting subroutine. However, the version that is used in Volume IV programs is slightly modified. It uses the option of SYSPLT that offsets the origin by some factor. Otherwise it is identical to the Caltech program.

```
      SUBROUTINE MDTPLT    (N,TTL1,L1,TTL2,L2,TTL3,L3,HTTL,LH,VTTL,LV,
     1 YGP)
      DIMENSION    DD(3), TTL1(1),TTL2(1),TTL3(1),HTTL(1),VTTL(1),A(2)
      COMMON /BM1/ X(4200),Y(4200)
      COMMON/COMPLO/ITEST,XLNG,YLNG
      DIMENSION XL(2),YL(2),CONF(7),XR(2),YR(2)
      DATA DD/3*0./
      DATA CONF/'95 P','ERCE','NT C','ONFI','DENC','E LE','VEL '/
      CALL SYSOFF (0.0,1.0,0.0,1.0)
      XL(1)=0.
      YL(1)=YGP
      XL(2)=18.25
      YL(2)=YGP
       XR(1) = 24.7
      YR(1)=YGP
      XR(2)=25.
      YR(2)=YGP
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      DO 6 I=2,N
      IF (YMIN-Y(I)) 2,3,1
    1 YMIN=Y(I)
      GO TO 3
    2 IF (YMAX.GE.Y(I)) GO TO 3
      YMAX=Y(I)
    3 IF (XMIN-X(I)) 5,6,4
    4 XMIN=X(I)
      GO TO 6
    5 IF (XMAX.GE.X(I)) GO TO 6
      XMAX=X(I)
    6 CONTINUE
       XMIN = 0.
       XMAX = 25.
      XCO=11.0
      YCO=8.5
      XCORD=0.0
      YCORD=0.0
      IDIV=6
      CALL SQBOUN (XCO,YCO,IDIV,XCORD,YCORD)
      CALL SYSOFF(1.75,1.,1.6,1.)
      XCO=8.0
      YCO=5.0
      CALL SQBOUN (XCO,YCO,IDIV,XCORD,YCORD)
      CALL SYSOFF(1.75,.32,1.6,1.)
      DO 7 I=1,6
      LABEL=5*(I-1)
      IF(I .EQ. 6)  GO TO 16
      DO 15 J=1,5
      IF (I.EQ.1.AND.J.EQ.1) GO TO 15
      XLAB=LABEL+J-1
      CALL SYSPLT (XLAB,0.1,13)
      CALL SYSPLT (XLAB,0.0,12)
   15 CONTINUE
   16 CALL OUTCOR(A,NX)
      WRITE (6,100) LABEL
      CALL OUTCOR
      OFFSET=1.75+1.6*(I-1)
      G=.13
      IF (LABEL.GE.10) G=.2/3.
```

```
 7 CALL SYSSYM(OFFSET-G,1.45,.1,A,2,0.)
   OFFSET=5.75-LH*3./70.
   CALL SYSSYM(OFFSET,1.25,.1 ,HTTL,LH,0.)
   LOGY=ALOG10(YMAX)
   IF (LOGY.LT.0) LOGY=0
   MAXY=10**LOGY
   IF (LOGY.GT.4) GO TO 11
   DO 8 IDIV=1,9
   IF (IDIV*MAXY.GE.YMAX) GO TO 9
 8 CONTINUE
   IF (LOGY.GE.4) GO TO 11
   IDIV=10
 9 CALL SYSSYM(1.45,1.55,.1,'  0',3,0.)
   CALL SYSOFF(1.75,1.,1.6 ,5./IDIV)
   DO 27 I=1,IDIV
   CALL SYSPLT (8.0,FLOAT(I),13)
   CALL SYSPLT (7.9,FLOAT(I),12)
27 CONTINUE
   CALL SYSOFF  (1.75,0.32,1.6,1.0)
   DO 26 IK=1,5
   I=6-IK
   LABEL=5*(I-1)
   DO 26 JK=1,5
   J=6-JK
   IF (I.EQ.1.AND.J.EQ.1) GO TO 26
   XLAB=LABEL+J-1
   CALL SYSPLT (XLAB,5.0,13)
   CALL SYSPLT (XLAB,4.9,12)
26 CONTINUE
   CALL SYSOFF  (1.75,1.0,1.6,5.0/IDIV)
   DO 10 IK=1,IDIV
   I=IDIV-IK+1
   CALL SYSPLT(0.,FLOAT(I),13)
   CALL SYSPLT(.1,FLOAT(I),12)
   LABEL=I*MAXY
   CALL OUTCOR(A,NX)
   WRITE (6,200) LABEL
   CALL OUTCOR
   OFFSET=1.55+5.*I/IDIV
   CALL SYSSYM (1.32,OFFSET,0.1,A,5,0.0)
10 CONTINUE
   OFFSET=4.1 -LV*3./70.
   CALL SYSSYM(1.25,OFFSET,.1 ,VTTL,LV,90.)
   OFFSET=5.75-L1*3./58.
   CALL SYSSYM(OFFSET,7.1 ,.12,TTL1,L1,0.)
   OFFSET=5.75-L2*3./70.
   CALL SYSSYM(OFFSET,6.9 ,.1,TTL2,L2,0.)
   OFFSET=5.75-L3*3./70.
   CALL SYSSYM(OFFSET,6.7 ,.1,TTL3,L3,0.)
   CALL SYSOFF(1.75,1.,1.6 ,1.)
   YBIG=FLOAT(IDIV*MAXY)
   YCNF=(YGP/YBIG)*5. +  1.6 - 0.04
   XCNF=7.7
    YOT = YGP/YBIG*5.
   CALL SYSSYM(XCNF,YCNF,0.08,CONF,27,0.0)
    XLR = 5.8
    NNN = (XLR-XL(1))/0.1-1.
   CALL SYSPLT( XL(1),YOT,13 )
    DO 12  I = 1,NNN,2
    XXXX = XL(1)+0.1*I
    CALL SYSPLT(XXXX,YOT,12)
    XXXX = XXXX+0.1
```

```
 12   CALL SYSPLT(XXXX,YOT,13)
      IF(MOD(NNN,2).EQ.0) CALL SYSPLT(XLR,YOT,12)
     ITEST=1
     XLNG=8.
     YLNG=5.
     CALL XYPLOT(2,XR,YR,.0,25.0,0.0,YBIG,DD,0)
     DD(3)=1.0
     CALL XYPLOT (N,X,Y,0.0,25.0,0.0,FLOAT (IDIV*MAXY),DD,1)
     CALL SYSOFF(0.,1.,0.,1.)
     RETURN
  11 WRITE (6,300) YMAX
     STOP
 100 FORMAT(I2)
 200 FORMAT (I5)
 300 FORMAT('-YMAX TOO LARGE.  YMAX = ',E16.8)
     END
```

```
      SUBROUTINE SQBOUN (XTOP,YTOP,IDIV,XBOT,YBOT)
      REAL*8 DIV
      DIV=1.0D0/IDIV
      XINC=(XTOP-XBOT)*DIV
      YINC=(YTOP-YBOT)*DIV
      CALL SYSPLT (XBOT,YBOT,13)
      YCORD=YBOT
      DO 1 I=1,IDIV
      XCORD=XINC*I + XBOT
    1 CALL SYSPLT (XCORD,YCORD,12)
      DO 2 I=1,IDIV
      YCORD=YINC*I + YBOT
    2 CALL SYSPLT (XCORD,YCORD,12)
      DO 3 I=1,IDIV
      XCORD = XTOP-XINC*I
    3 CALL SYSPLT (XCORD,YCORD,12)
      DO 4 I=1,IDIV
      YCORD=YTOP-YINC*I
    4 CALL SYSPLT (XCORD,YCORD,12)
      RETURN
      END
```

```
      SUBROUTINE XYPLOT(N,X,Y,XMN,XMX,YMN,YMX,DD,LAB)
C     N = TOTAL NO. OF POINTS TO BE PLOTTED.
C     X = ARRAY OF ABSCISSA.
C     Y = ARRAY OF ORDINATES.
C     XMN, XMX = RANGE OF X
C     YMN, YMX = RANGE OF Y.
C     LAB = 0, PLOT ON SAME SHEET OF PAPER.
C     LAB .GT. 0, PLOT TERMINATES CURRENT SHEET OF PAPER.
C     LAB = -1, PRINTING OF JOB SEQUENCE NUMBER IS SUPPRESSED.
C     DD(1),DD(2) = BCD TITLE TO BE PLOTTED ON THE UPPER RIGHT CORNER
C                   AT THE END OF EACH PLOT AS AN IDENTIFICATION. IF
C                   DD(1) = 0 THIS PLOTTING WILL BE SUPPRESSED.
C     DD(3) = 0, PRINTING OF PLOTTER INFORMATION LIKE SCALE, LABEL ETC
C             ON USER'S OUTPUT WILL BE SUPPRESSED.
C     DD(3) .NE. 0, THE ABOVE INFORMATION WILL BE PRINTED.
C
      COMMON/COMPLC/ITEST,XLNG,YLNG
       REAL LBOUND
      DIMENSION X(1),Y(1),DD(1)
      INTEGER PEN
      DATA XC,YC/1HX,1HY/
       RBOUND = XMX
       IF (XMN .LE. XMX) GO TO 5
       RBOUND = XMN
   5   CONTINUE
       IF (YMN.LE.YMX) GO TO 10
       TBOUND = YMN
       BBOUND = YMX
  10   CONTINUE
      IF(ITEST .EQ. 1) GO TO 12
      XLNGTH = 15.0
      YLNGTH = 10.0
      GO TO 13
   12 XLNGTH = XLNG
      YLNGTH = YLNG
   13 IF(N .GT. 0) GO TO 18
   14 WRITE (6,16) N,XMN,XMX,YMN,YMX
   16 FORMAT(//' ERROR RETURN FROM ''XYPLOT''--ONE OF THE FOLLOWING ARGU
     1MENTS HAS WRONG VALUE'/10X,'N,XMN,XMX,YMN,YMX =',I10,4E18.6)
      RETURN
   18 IF(XMX .EQ. XMN .OR. YMX .EQ. YMN) GO TO 14
      SX = XLNGTH/(XMX-XMN)
      SY = YLNGTH/(YMX-YMN)
      PEN=13
      DO 30 I = 1,N
       IF (X(I).GT.RBOUND)GO TO 28
      XX = (X(I) - XMN)*SX
      YY = (Y(I) - YMN)*SY
      CALL SYSPLT(XX,YY,PEN)
      PEN=12
      GO TO 30
   28 PEN=13
   30 CONTINUE
      IF(DD(3) .EQ. 0.0) GO TO 35
      WRITE (6,331)
  331 FORMAT('OXYPLOT COMPLETED.')
      WRITE (6,34) XC,XMN,XMX,YC,YMN,YMX
   34 FORMAT(5X5H THE A1,27H COORDINATE IS SCALED FROM 1PE10.3,4H TO
     1 E10.3)
   35 IF(LAB .EQ. 0) RETURN
      IF(DD(1) .EQ. 0.0) GO TO 32
```

```
      CALL SYSSYM(13.0,9.8,0.2,DD,8,0)
   32 CALL SYSEND(LAB,DD(3))
      RETURN
       END
       BLOCK DATA
C XYPLOT
       COMMON /COMPLO/ IPLO,XPLO,YPLO
       DATA IPLO,XPLO,YPLO /0,15.,10./
C
       END
```

Subroutine LOGLOG (Lee)

LOGLOG is used by MDTRHA to produce a log-log plot of Fourier amplitude spectra. It has a maximum range of 10^{-2} cps to 10^{2} cps on the frequency scale.

Usage

```
CALL LOGLOG (N, TTL1, L1, TTL2, L2, TTL3, L3, TTLH, LH,
*        TTLV, LV)
COMMON/BM1/X(4200), Y(4200)
```

Where

N is the no. of points to be plotted.

TTL1 is the plot title.

L1 is the no. of characters in TTL1.

TTL2 is the earthquake title.

L2 is the no. of characters in TTL2.

TTL3 is the accelerogram title.

L3 is the no. of characters in TTL3.

TTLH is the horizontal (frequency) axis title.

LH is the no. of characters in TTLH.

TTLV is the vertical (amplitude) axis title.

LV is the no. of characters in TTLV.

COMMON/BM1/ has

X is the frequency coordinates.

Y is the amplitude values.

Subroutine SQBOUN is used inside LOGLOG to draw a square boundary around the plot. SQBOUN uses a Caltech plotting routine, SYSPLT, to draw the lines. SQBOUN is also used by MDTPLT.

```
      SUBROUTINE LOGLOG(N,TTL1,L1,TTL2,L2,TTL3,L3,TTLH,LH,TTLV,LV)
      DIMENSION TTL1(1),TTL2(1),TTL3(1)
      DIMENSION TTLV(1),TTLH(1)
      COMMON/BM1/X(4200),Y(4200)
C    PLOT THE GRAPH BOUNDARY
      CALL SYSOFF (0.0,1.0,0.0,1.0)
      IDIV=6
      XR=11.0
      YR=8.5
      YL=0.0
      XL=0.0
      CALL SQBOUN (XR,YR,IDIV,XL,YL)
C    PLOT THE TITLE
      CALL SYSSYM(5.75-L1*3./58.,7.1,.12,TTL1,L1,0.)
      CALL SYSSYM(5.75-L2*3./70.,6.9,.1,TTL2,L2,0.)
      CALL SYSSYM(5.75-L3*3./70.,6.7,.1,TTL3,L3,0.)
      XL=1.75
      YL=1.6
      XR=9.75
      YR=6.6
      CALL SQBOUN (XR,YR,IDIV,XL,YL)
      NHI=4
      NVI=5
      SX=(XR-XL)/NHI
      SY=(YR-YL)/NVI
      NH1=NHI+1
      NV1=NVI+1
C    PLOT THE HORIZONTAL LABEL
      YLBL=YL-.15
      DO 2 NSTP=1,NH1
      XX=XL+(NSTP-1)*SX
      IF(NSTP.EQ.NH1)GO TO 1
      CALL SYSSYM(XX,YL+.075,.15,13,-1,0.)
      DO 11 I=1,4
      CALL SYSSYM(XX+SX/5.*I,YL+.05,.1,13,-1,0.)
   11 CONTINUE
    1 CALL OUTCOR(AN,NUM)
      LABEL=NSTP-3
      WRITE(6,100)LABEL
      CALL OUTCOR
      XLBL=XX-.115
      IF(LABEL.LT.0)XLBL=XX-.07
    2 CALL SYSSYM(XLBL,YLBL,.1,AN,2,0.)
      STTL=.1
      NTTL=LH+7
      XTTL=XL+NHI/2.*SX-NTTL/2.*STTL*6./7.
      YTTL=YLBL-2.*STTL
      NTTL=7
      CALL SYSSYM(XTTL,YTTL,STTL,'LOG OF ',NTTL,0.0)
      XTTL=XTTL+NTTL*STTL*6.0/7.0
      CALL SYSSYM(XTTL,YTTL,STTL,TTLH,LH,0.0)
C    PLOT THE VERTICAL LABEL
C    TAKE THE LOG; LOG(X(I)) SET TO -2. IF X(I).LT. .01
      DO 50 I=1,N
      X(I)=ALOG10(X(I))
      IF(X(I).LT.-2.)X(I)=-2.
      IF (Y(I).LT.5.0E-6) GO TO 49
      Y(I)=ALOG10(Y(I))
      GO TO 50
   49 Y(I)=-100.
   50 CONTINUE
```

```
      CALL MAXMIN(Y,N,YMAX,YMIN)
      LMAX=INT(YMAX)+1+        INT((YMAX-INT(YMAX))/.5)
      LMIN=LMAX-NVI
      YMIN=LMIN
      XLBL=XL-.2
      DO 60 NSTP=1,NV1
      YY=YL+(NSTP-1)*SY
      IF (NSTP.EQ.NV1) GO TO 60
      CALL SYSSYM (XR-0.075,YY,0.15,13,-1,90.0)
      CALL SYSSYM (XR-0.05,YY+SY/2.0,0.1,13,-1,90.0)
   60 CONTINUE
      DO 61 NSTP=1,NH1
      INSTP=NH1-NSTP+1
      XX=XL+(INSTP-1)*SX
      IF (NSTP.EQ.1) GO TO 61
      CALL SYSSYM (XX,YR-0.075,0.15,13,-1,0.0)
      DO 62 I=1,4
      CALL SYSSYM (XX+SX/5.0*I,YR-0.05,0.1,13,-1,0.0)
   62 CONTINUE
   61 CONTINUE
      DO 4 INSTP=1,NV1
      NSTP=NV1-INSTP+1
      YY=YL+(NSTP-1)*SY
      IF (NSTP.EQ.NV1) GO TO 3
      CALL SYSSYM(XL+.075,YY,.15,13,-1,90.)
      CALL SYSSYM(XL+.05,YY+SY/2.,.1,13,-1,90.)
    3 CALL OUTCOR(AN,NUM)
      LABEL=LMIN+(NSTP-1)
      WRITE(6,100)LABEL
      CALL OUTCOR
      YLBL=YY-.045
    4 CALL SYSSYM(XLBL,YLBL,.1,AN,2,0.)
      STTL=.1
      NTTL=LV+7
      XTTL=XLBL-1.25*STTL
      YTTL=YL+NVI/2.*SY-NTTL/2.*STTL*6./7.
      NTTL=7
      CALL SYSSYM(XTTL,YTTL,STTL,'LOG OF ',NTTL,90.0)
      YTTL=YTTL+NTTL*STTL*6.0/7.0
      CALL SYSSYM (XTTL,YTTL,STTL,TTLV,LV,90.0)
    PLOT THE GRAPH
      DO 5 I=1,N
      IF(Y(I).LT.YMIN)Y(I)=YMIN
    5 CONTINUE
      XO=(XL+XR)/2.
      YO=YL+(-LMIN)*SY
      I1=1
      IF(X(1).GE.-2.)GO TO 20
      DO 12 I=2,N
      IF(X(I).GE.-2.)GO TO 13
   12 I1=I1+1
   13 I2=I1+1
      Y(I1)=Y(I1)+(Y(I2)-Y(I1))/(X(I2)-X(I1))*(-2.-X(I1))
      X(I1)=-2.
   20 CALL SYSPLT(XO+SX*X(I1),YO+SY*Y(I1),3)
      DO 6 I=I1,N
      CALL SYSPLT(XO+SX*X(I),YO+SY*Y(I),2)
    6 CONTINUE
      CALL SYSEND (1,1.0)
  100 FORMAT(I2)
      RETURN
      END
```

DATA PROCESSING FOR VOLUME V: RESPONSE ENVELOPE SPECTRA

The response spectrum curves alone do not give a complete picture of the effects of duration of an acceleration history on the structural response. This is clearly illustrated by comparing the spectra for the Parkfield earthquake (Housner and Trifunac, 1967) and the Imperial Valley, 1940, earthquake (Hudson, et al, 1971). Although the two earthquakes have spectra with similar amplitudes, the short impulsive motion recorded during the Parkfield earthquake caused little or no damage. On the other hand the motion recorded during the Imperial Valley, 1940, earthquake represents an example of the relatively long and damaging shaking produced by the multiple events successively occurring along a 40 mile long fault (Trifunac and Brune, 1970). The distribution of shaking and the details of the changing response amplitudes in time are thus important factors that should not be neglected in the analysis and the design of earthquake resistant structures.

A method of displaying the effects of the time variations in response analysis consists of plotting the Response Envelope Spectrum (RES) (Trifunac, 1971). This three-dimensional plot of response amplitudes gives information on the amplitude of the response envelope at a given time and frequency and thus gives a detailed picture of response fluctuations in time. The response amplitude here can be taken to be the true relative velocity, the pseudo relative velocity, the absolute acceleration or the relative displacement. The choice, of course, depends on the particular needs of the analysis. The usefulness of the RES is further increased by the fact that it represents a form of the multiple filter analysis of the acceleration time series, and as such it becomes an important tool in analyzing the arrivals of different phases of strong ground motion in dispersion analysis and in the studies of the effects of local geology on the recorded accelerations (Trifunac, 1971).

The RES data are computed in the following way. The recorded ground acceleration is substituted into the forcing function term of the differential equation of motion of the single-degree-of-freedom viscously damped oscillator. This equation is then integrated by using the Duhamel integral solution, which is coded in the PCNO3 and PCNO4 subroutines. The envelope of the relative response $x(t)$ is next approximated by connecting the successive peaks of $|x(t)|$ by a straight line. This is illustrated in Figure 11 where the response and its envelope are computed for the NS component of the Imperial Valley, 1940, accelerogram (IIA001) for an oscillator with a natural frequency of 1.59 cps and fraction of critical damping equal to 0.10. Such calculations are performed for all 91 periods used in the computation of response spectra in Volume III (see subroutine SPCTRA) for 0, 2, 5, 10, and 20 percent of critical damping. The envelopes of the relative displacement responses, interpolated at the rate of 5 points per second, are stored on the Volume V tape and plotted by the Volume V programs using linear or logarithmic ordinate scales. The maxima of RES for $SD*\frac{2\pi}{T}$ (Hudson, et al, 1972) versus frequency are also plotted using a heavy line. Their amplitudes correspond to the pseudo velocity spectra presented in Volume III and here give the time of the maximum response.

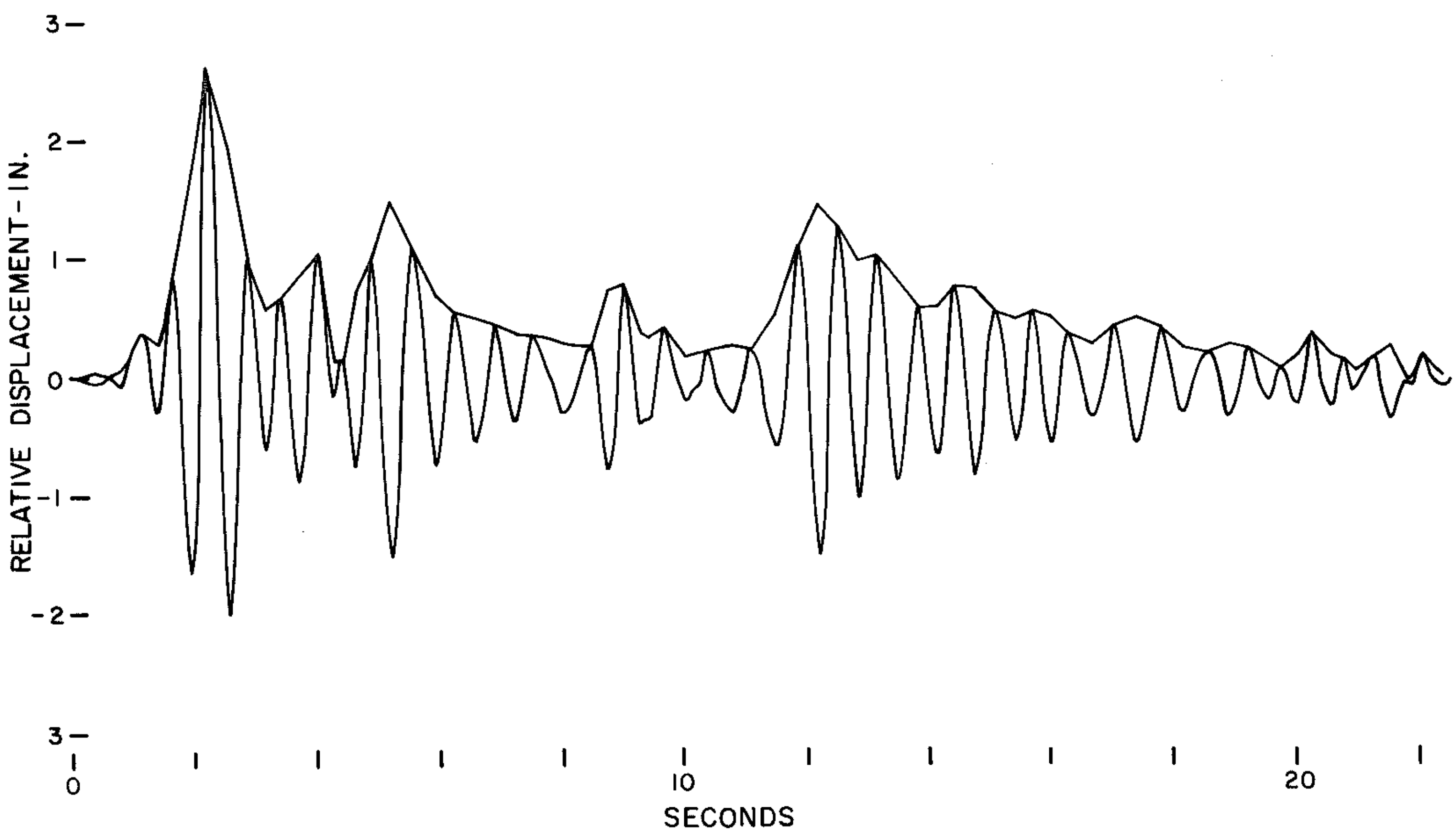


Figure 11. The relative displacement and its envelope for an oscillator with natural frequency $f_n = 1.59$ cps and fraction of critical damping $\zeta = 0.10$, for the NS component of the El Centro, 1940, strong motion accelerogram.

The information about the earthquake and the characteristics of the RES spectra appear in the descriptive titles. The second line of the long title, above the frequency label of the RES plot, gives the name, date, and time of occurrence of the earthquake. The third line consists of two labels, the instrument location or observation station and the particular component. The first identification label (Figure 12), e.g., VA001, indicates that this spectrum results from an accelerogram that belongs to Part A of Volume II containing the corrected data, and that this record is the first of the complete series. The response spectra for this record appear in Volume III, Part A. The peak value and the fraction of critical damping for which it was calculated appear on the right hand side of each RES plot together with the table giving the correspondence between the plotted contour levels and the actual spectral amplitudes.

For each component of acceleration five RES plots are generated corresponding to the five damping values (0, 2, 5, 10, and 20 percent) used in the Volume III spectral calculations. Their plots are shown in Figures 12 through 16 where the pseudo velocity ($\frac{2\pi}{T}$ *SD) spectra are plotted for the NS component of the El Centro accelerogram. The ordinates for the spectra are in units of in/sec in accordance with both engineering practice and the pseudo velocity spectra in Volume III.

While the linear scale for the RES of the pseudo relative velocity ordinates should be adequate for the analysis of typical engineering structures, for those structures that have unusually high- or low-frequency characteristics, RES ordinates can be plotted on any nonlinear scale. An example of using the logarithmic scale is shown in Figures 17 through 21.

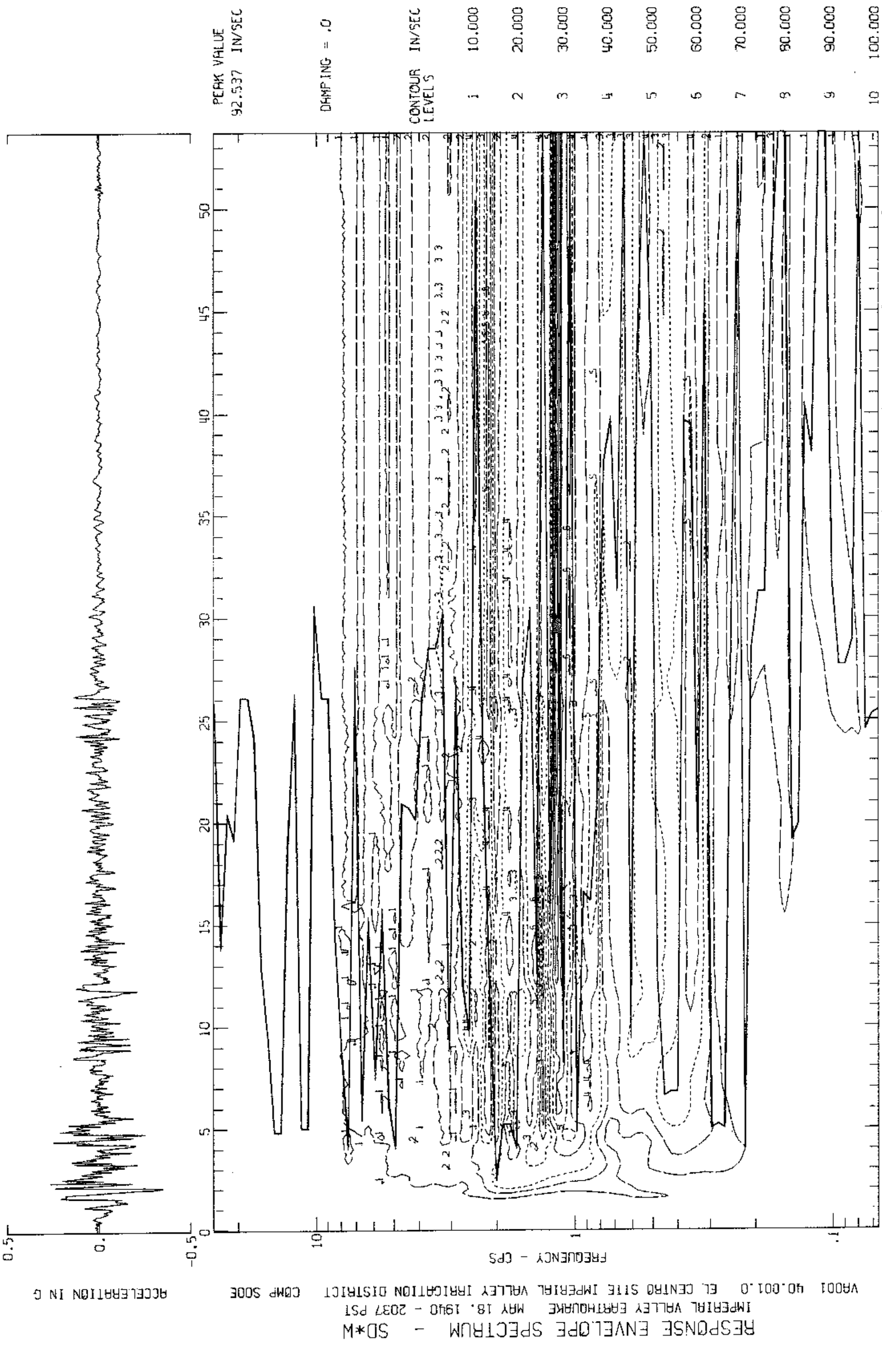

PEAK VALUE
92.537 IN/SEC
DAMPING = .0
CONTOUR LEVELS IN/SEC
1 10.000
2 20.000
3 30.000
4 40.000
5 50.000
6 60.000
7 70.000
8 80.000
9 90.000
10 100.000
TIME - SECONDS
FREQUENCY - CPS
ACCELERATION IN G
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
RESPONSE ENVELOPE SPECTRUM - SD*W

Figure 12

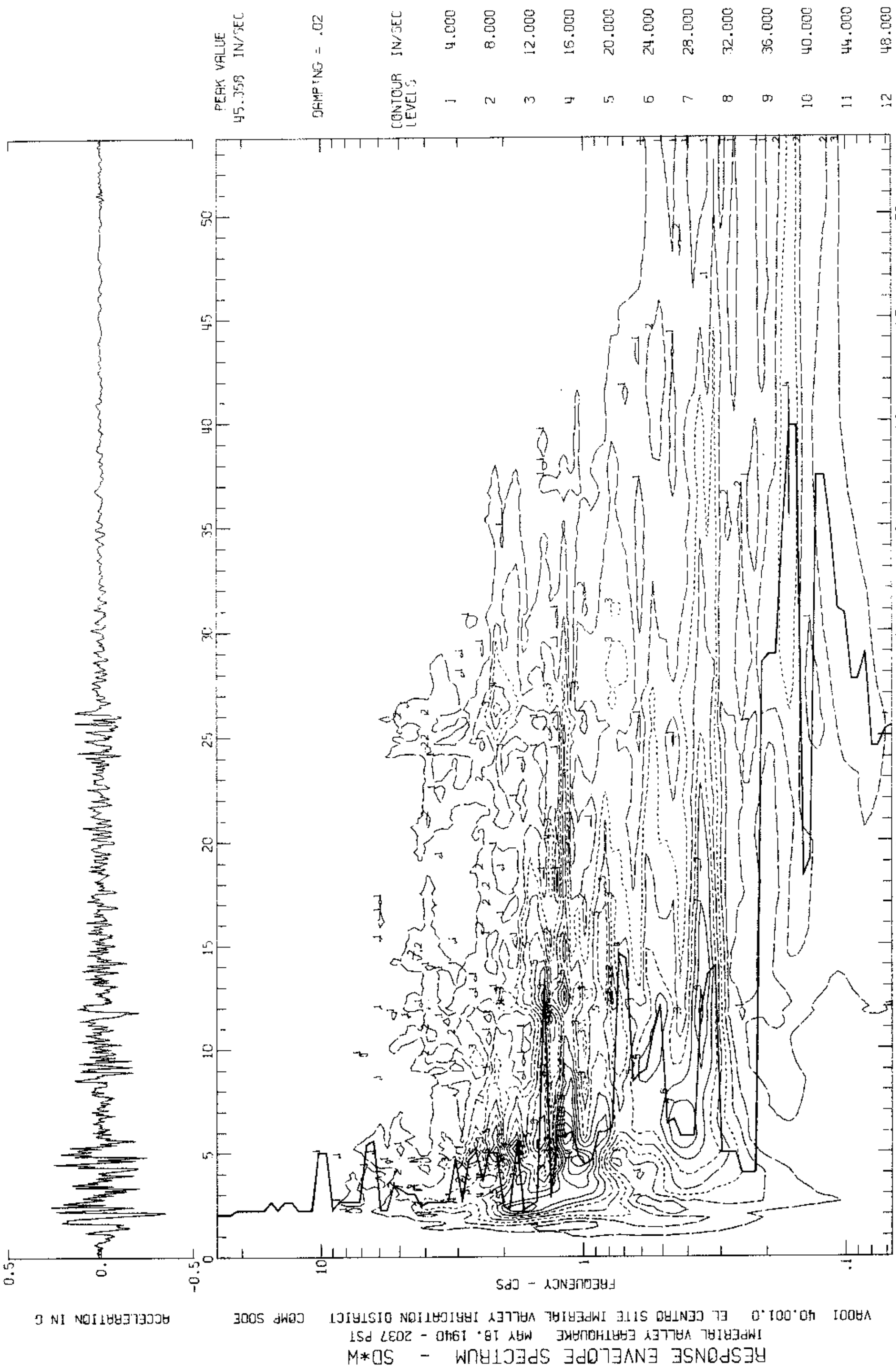

RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE
ACCELERATION IN G
FREQUENCY - CPS
TIME - SECONDS
PEAK VALUE
45.358 IN/SEC
DAMPING = .02
CONTOUR LEVELS IN/SEC
1 4.000
2 8.000
3 12.000
4 16.000
5 20.000
6 24.000
7 28.000
8 32.000
9 36.000
10 40.000
11 44.000
12 48.000

Figure 13

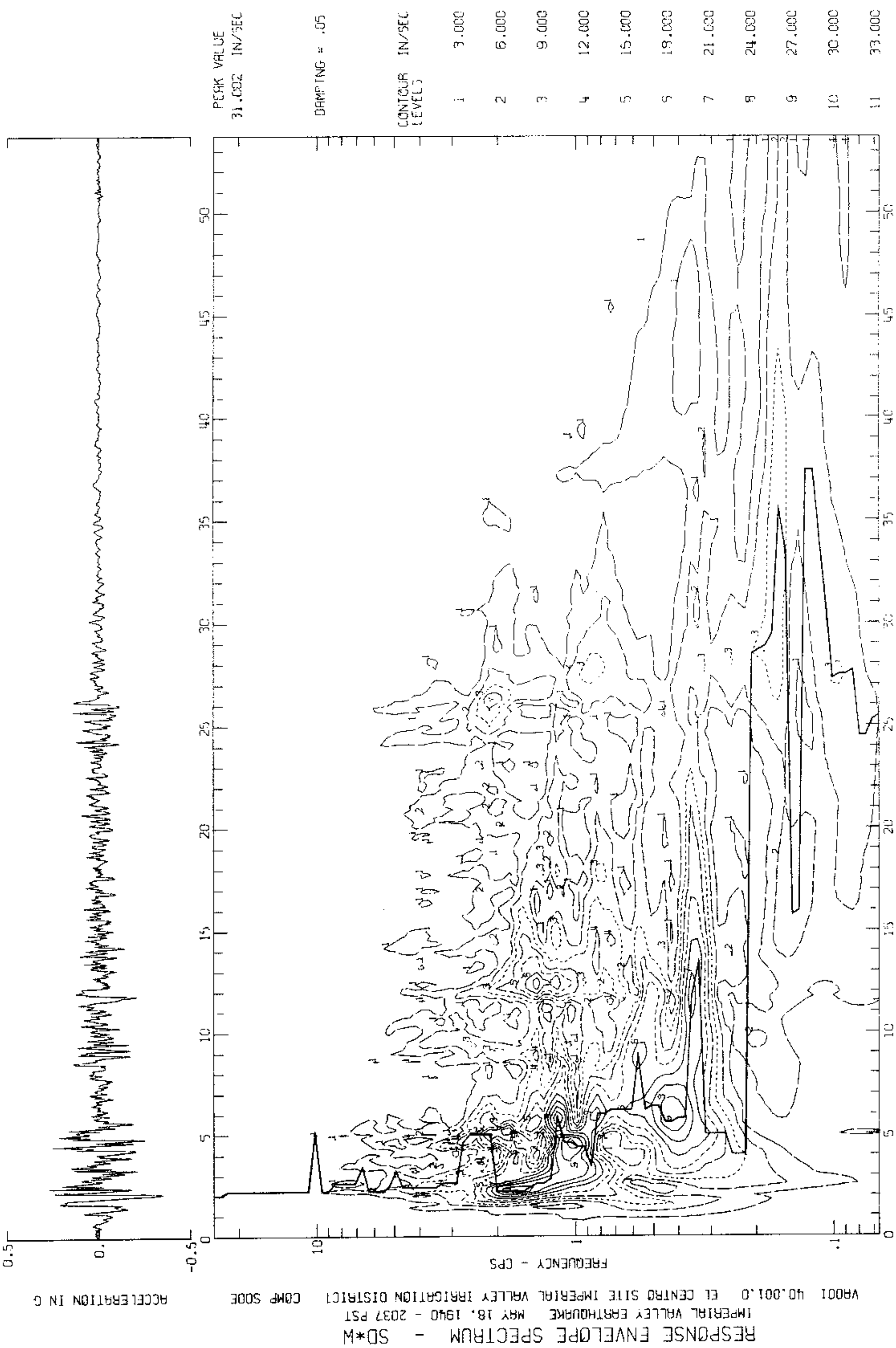

RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E
ACCELERATION IN G
FREQUENCY - CPS
TIME - SECONDS
PEAK VALUE
31.002 IN/SEC
DAMPING = .05
CONTOUR LEVEL IN/SEC
1 3.000
2 6.000
3 9.000
4 12.000
5 15.000
6 18.000
7 21.000
8 24.000
9 27.000
10 30.000
11 33.000

Figure 14

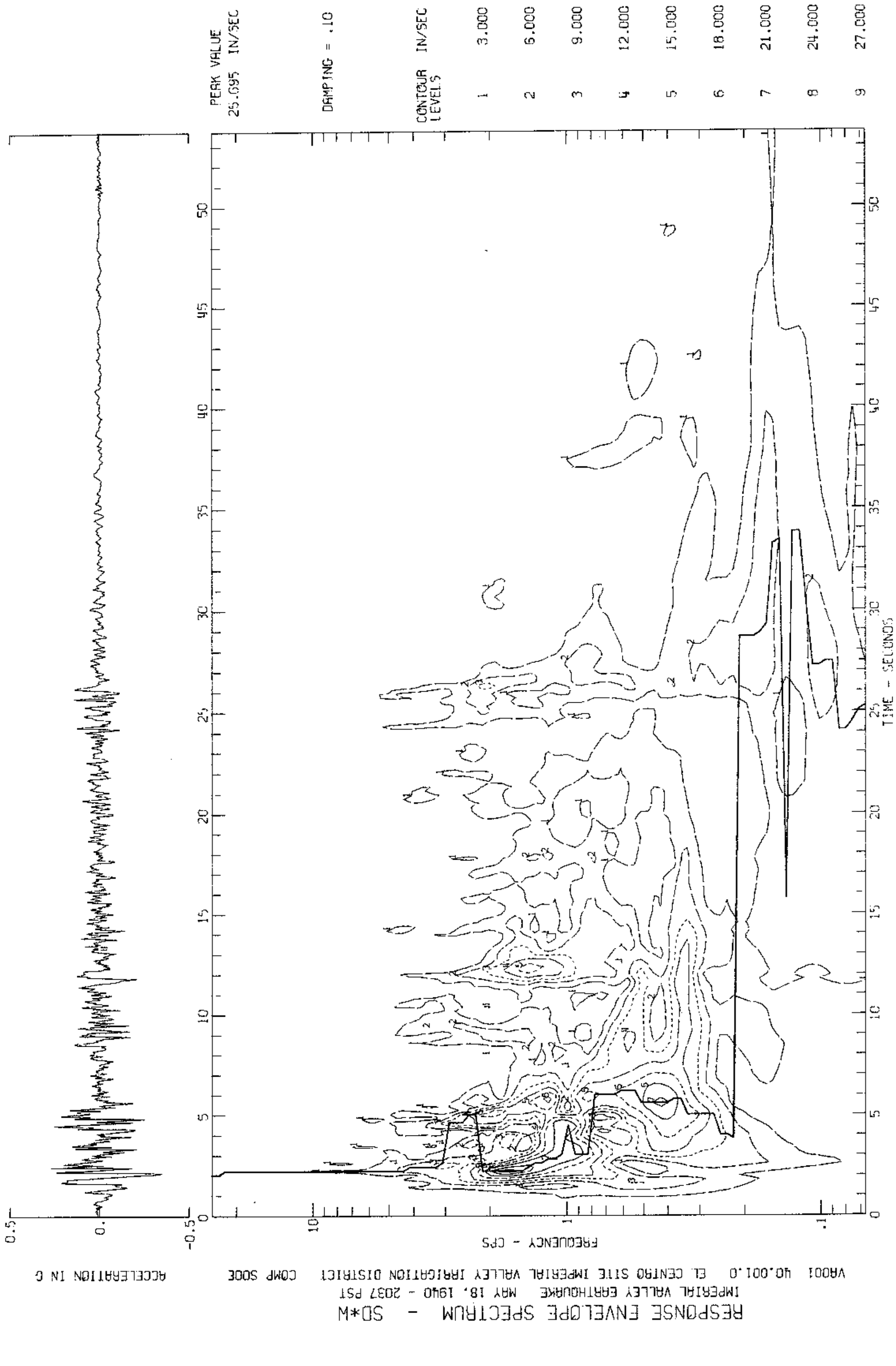

PEAK VALUE
25.095 IN/SEC
DAMPING = .10
CONTOUR LEVELS IN/SEC
1 3.000
2 6.000
3 9.000
4 12.000
5 15.000
6 18.000
7 21.000
8 24.000
9 27.000
TIME - SECONDS
FREQUENCY - CPS
ACCELERATION IN G
RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE

Figure 15

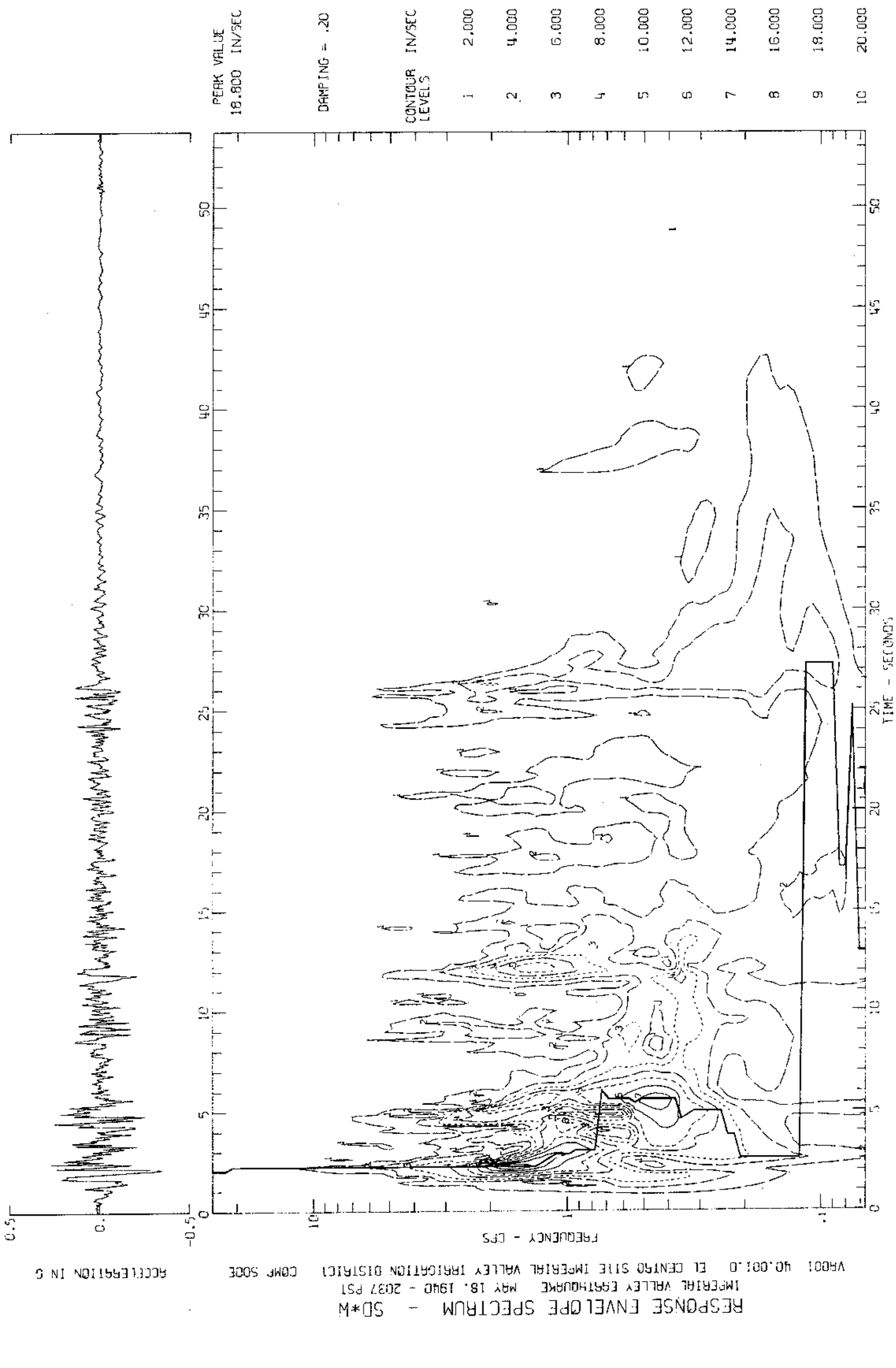

PEAK VALUE
18.800 IN/SEC
DAMPING = .20
CONTOUR IN/SEC
LEVELS
1 2.000
2 4.000
3 6.000
4 8.000
5 10.000
6 12.000
7 14.000
8 16.000
9 18.000
10 20.000
TIME - SECONDS
FREQUENCY - CPS
ACCELERATION IN G
RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE

Figure 16

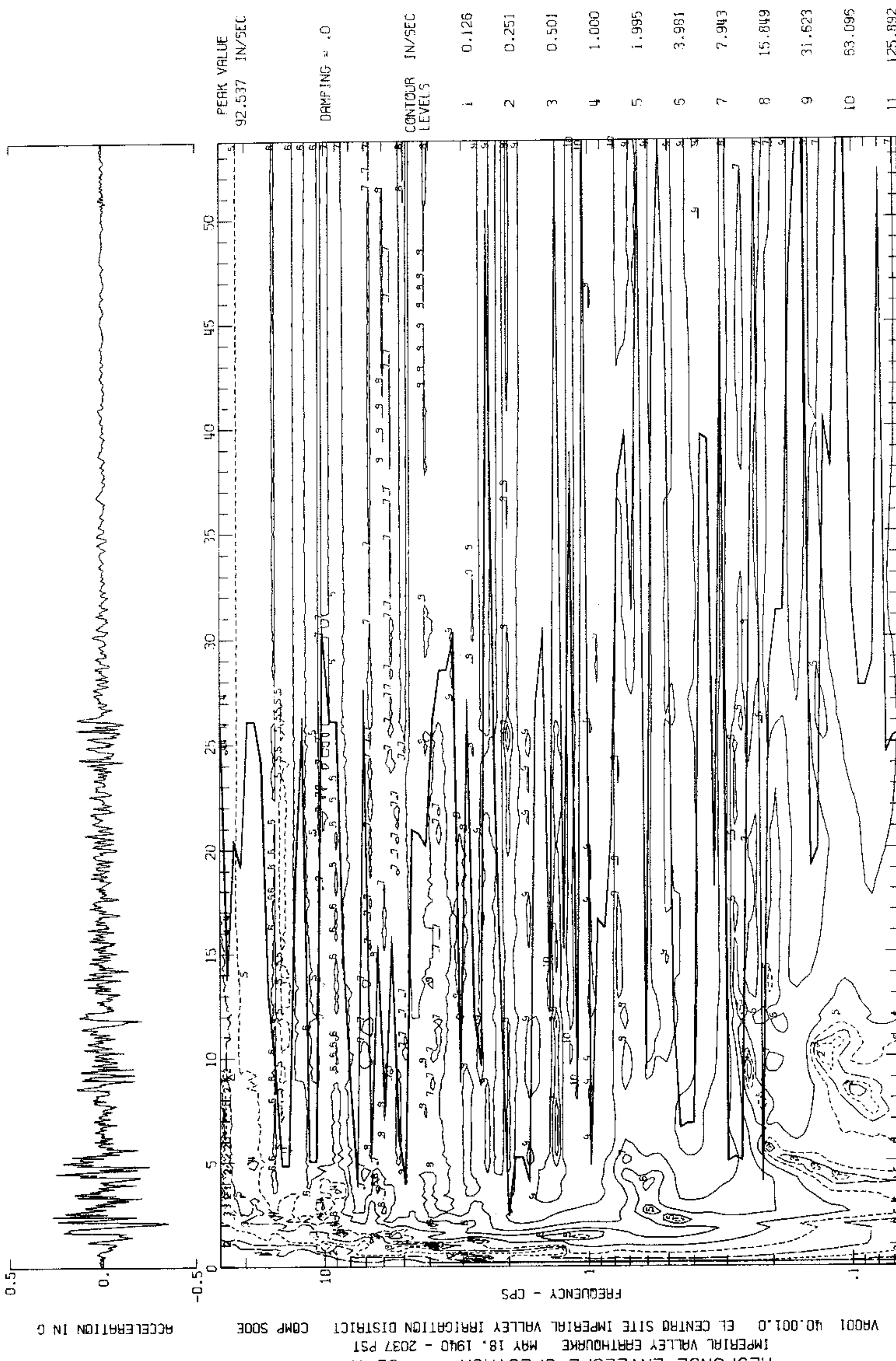

RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E
ACCELERATION IN G
FREQUENCY - CPS
TIME - SECONDS
PEAK VALUE
92.537 IN/SEC
DAMPING = .0
CONTOUR LEVELS IN/SEC
1 0.126
2 0.251
3 0.501
4 1.000
5 1.995
6 3.981
7 7.943
8 15.849
9 31.623
10 63.096
11 125.892

Figure 17

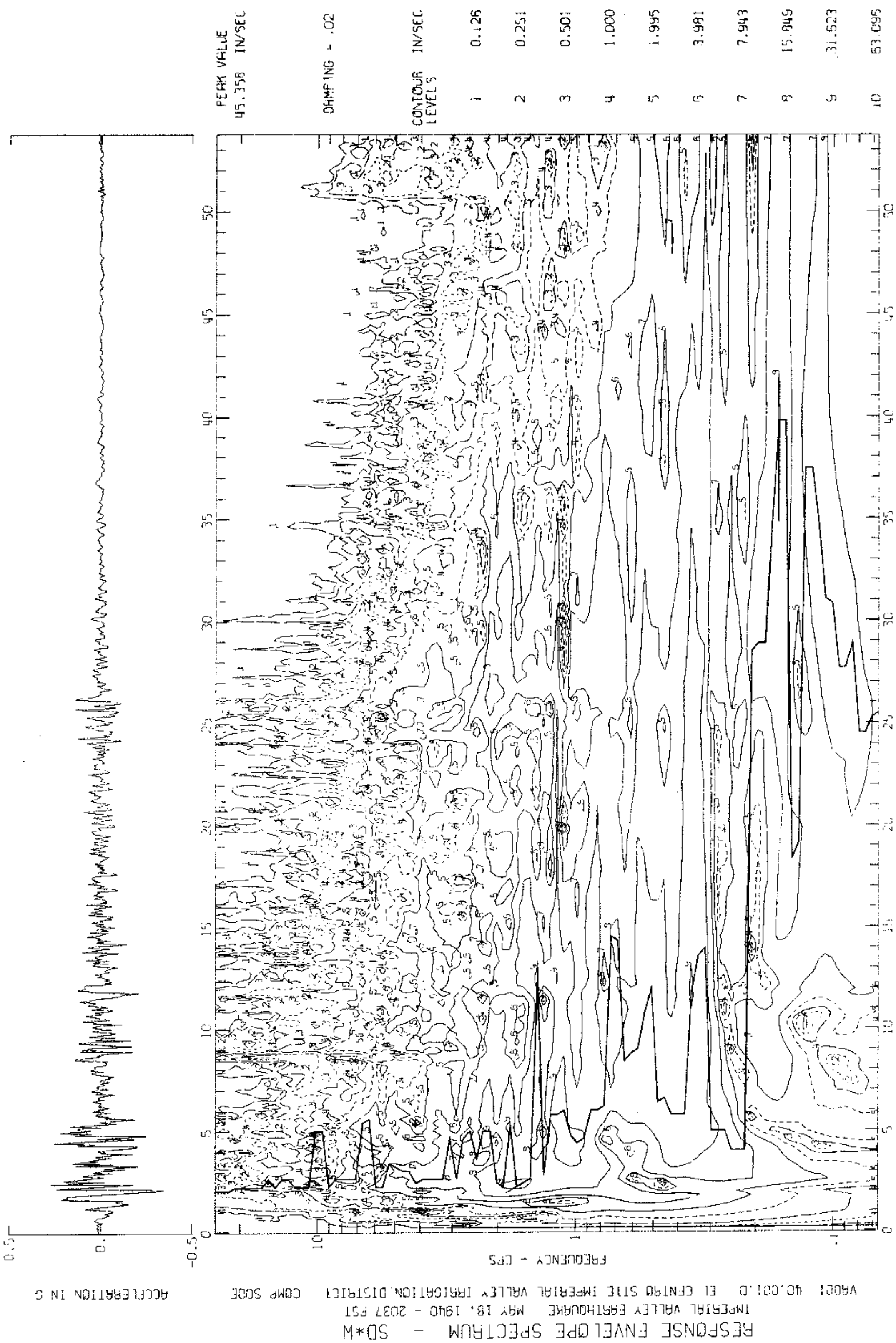

PEAK VALUE
45.358 IN/SEC
DAMPING - .02
CONTOUR LEVELS IN/SEC
1 0.126
2 0.251
3 0.501
4 1.000
5 1.995
6 3.981
7 7.943
8 15.849
9 31.623
10 63.096
TIME - SECONDS
FREQUENCY - CPS
ACCELERATION IN G
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
RESPONSE ENVELOPE SPECTRUM - SD*W

Figure 18

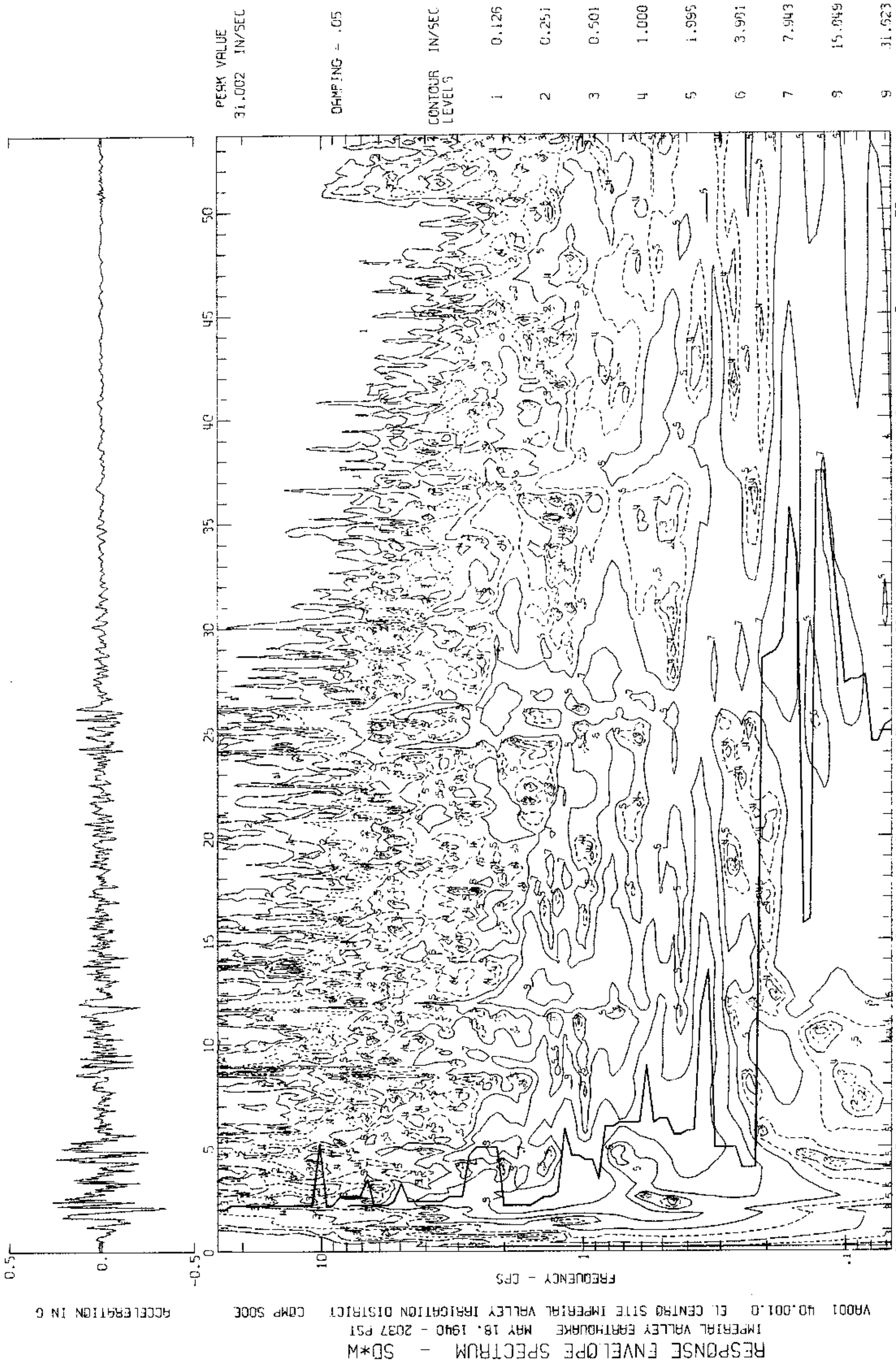

PEAK VALUE
31.002 IN/SEC
DAMPING - .05
CONTOUR IN/SEC
LEVELS
1 0.126
2 0.251
3 0.501
4 1.000
5 1.995
6 3.981
7 7.943
8 15.849
9 31.623
TIME - SECONDS
FREQUENCY - CPS
ACCELERATION IN G
RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E

Figure 19

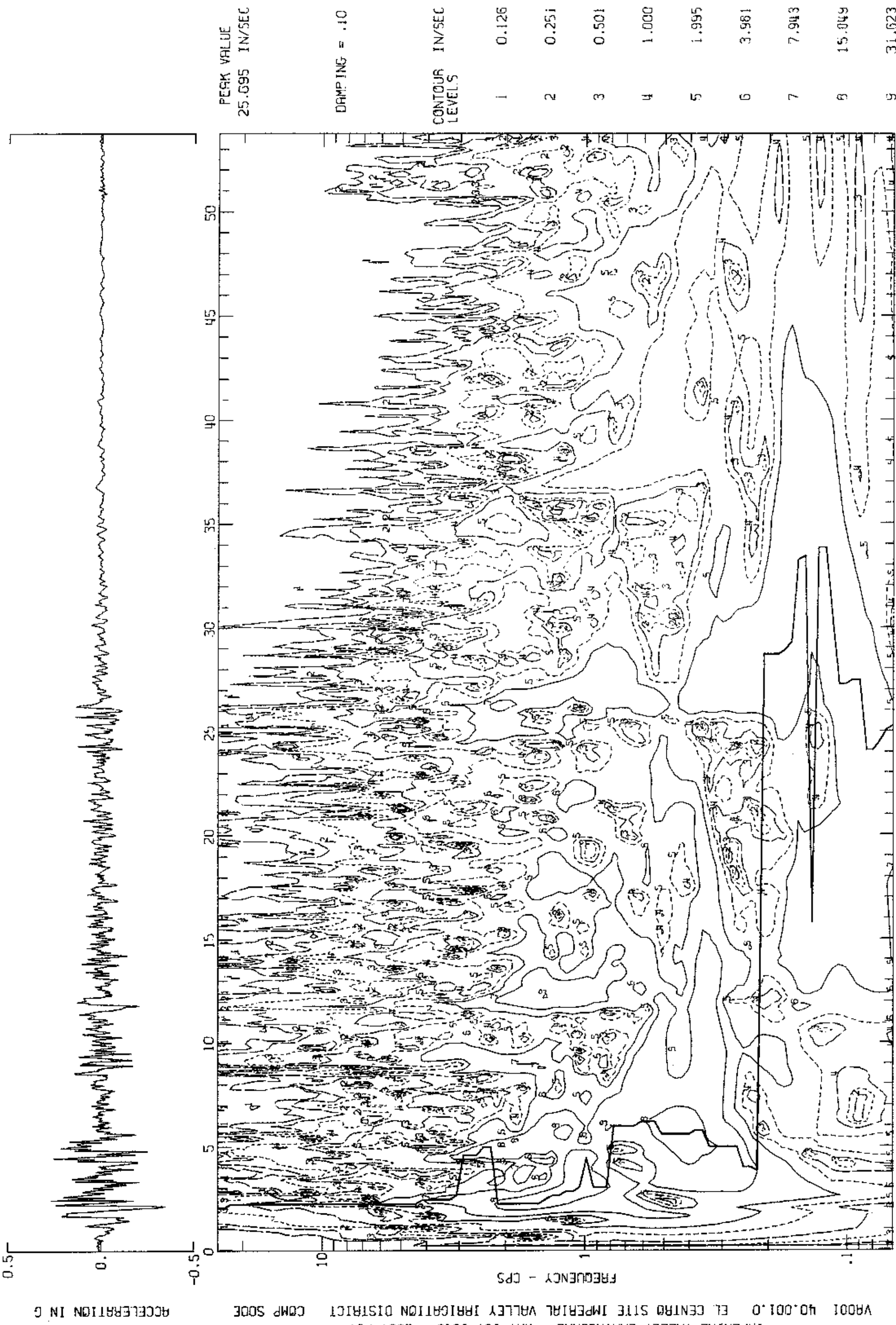

RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP SOOE
PEAK VALUE 25.095 IN/SEC
DAMPING = .10
CONTOUR LEVELS IN/SEC
1 0.126
2 0.251
3 0.501
4 1.000
5 1.995
6 3.981
7 7.943
8 15.849
9 31.623
ACCELERATION IN G
FREQUENCY - CPS
TIME - SECONDS

Figure 20

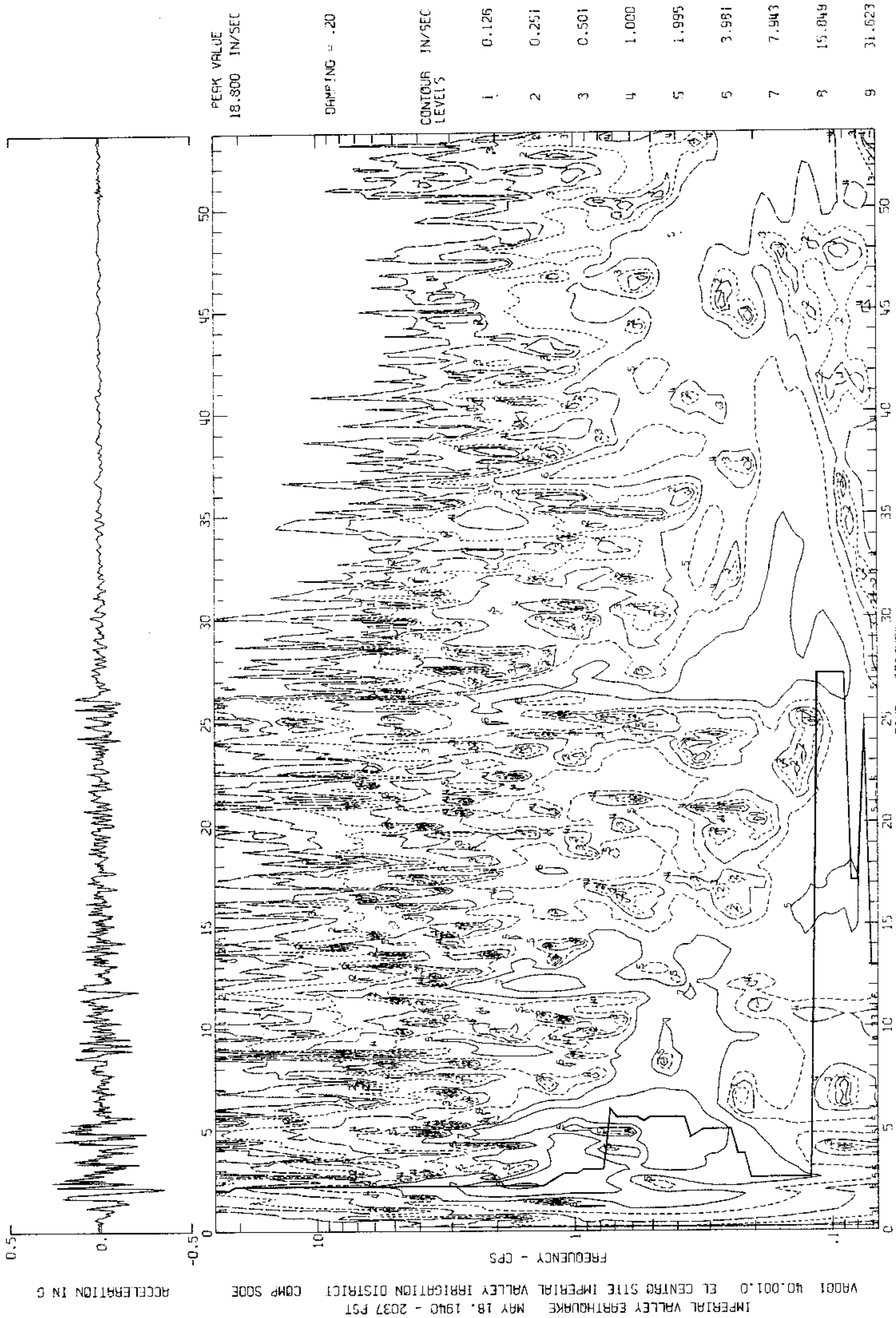

RESPONSE ENVELOPE SPECTRUM - SD*W
IMPERIAL VALLEY EARTHQUAKE MAY 18, 1940 - 2037 PST
VA001 40.001.0 EL CENTRO SITE IMPERIAL VALLEY IRRIGATION DISTRICT COMP S00E
ACCELERATION IN G
FREQUENCY - CPS
TIME - SECONDS
PEAK VALUE 18.800 IN/SEC
DAMPING = .20
CONTOUR LEVELS IN/SEC
1 0.126
2 0.251
3 0.501
4 1.000
5 1.995
6 3.981
7 7.943
8 15.849
9 31.623

Figure 21

PROGRAMS FOR PLOTTING RES DATA

Volume V MAIN Program (RESPLT) (Lee)

This program reads in:

(1) One file of corrected acceleration data from the Volume V tape.

(2) Corresponding RES spectral values from the same file of the Volume V tape generated by the Volume III programs.

It then plots a contour map representing the RES spectra.

Usage

The program reads in the following data:

```
TTL1, VTTL, HTTL, TTL4, TTL5
L1, LV, LH, L4, L5
NML
DIS
IDMP
NY, LSCL, NPLT, LPLT, IEXP, IPEAK, NFMAX, NL
DI, (DD(I), I=1, NL)
WIDTH, RATIO, AMIN, FSCALE, TSPACE, FSPACE, XLNGTH,
*       YLNGTH
LFILE, MFILE
```

Where

TTL1 is the main title of the contour plot (20 words of 4 characters each).

VTTL, HTTL are the Y and X axes titles (frequency and time) (20 words of 4 characters each).

TTL4, TTL5 are the titles for contour level labels (20 words of 4 characters each).

L1, LV, LH, L4, L5 are the corresponding numbers of characters used in the above titles.

NML are the numeric labels of the contour levels (30 words of 2 characters each).

(DIS(I), I=1, 8) is the list of spacings between the contour levels to be chosen.

For N = 1 to 5:

IDMP(N) = 1, if the RES of the Nth damping ratio is not to be plotted.

2, if it is to be plotted.

NY is the number of grid lines in the Y-direction (including the boundary lines).

LSCL = 1, vertical (frequency) axis in linear scale

2, vertical (frequency) axis in log scale

NPLT = 0, if the corresponding acceleration trace is not plotted

1, if the corresponding acceleration trace is plotted.

LPLT = 1, contour values given in linear scale.

2, contour values given in log scale.

IEXP = 0, RES for SD

1, RES for SD*W

2, RES for SD*W**2

here $W = 2\pi f$ and f is the frequency.

IPEAK = 2, if RES peak values at each frequency are joined together (by a thick line)

1, if RES peak values at each frequency are not joined together

NFMAX is the maximum number of contour levels allowed.

NL - the first NL contour levels are plotted with broken lines.

DI is the gap length for the broken lines of the first NL contour levels.

(DD(I), I=1, NL) are the lengths of broken lines for the first NL contour levels.

WIDTH is the width of the thick line joining the RES peak values versus frequency (if IPEAK = 2).

RATIO is the ratio of the size of the numeric contour labels to those in the main titles.

AMIN is the minimum value allowed for the first contour level (same unit as the contour levels).

FSCALE is the scale of the RES values (1 for units in in/sec, 2.54 for units in cm/sec).

TSPACE is the spacing of numeric time labels (in seconds).

FSPACE is the spacing of numeric frequency labels (in radians/sec).

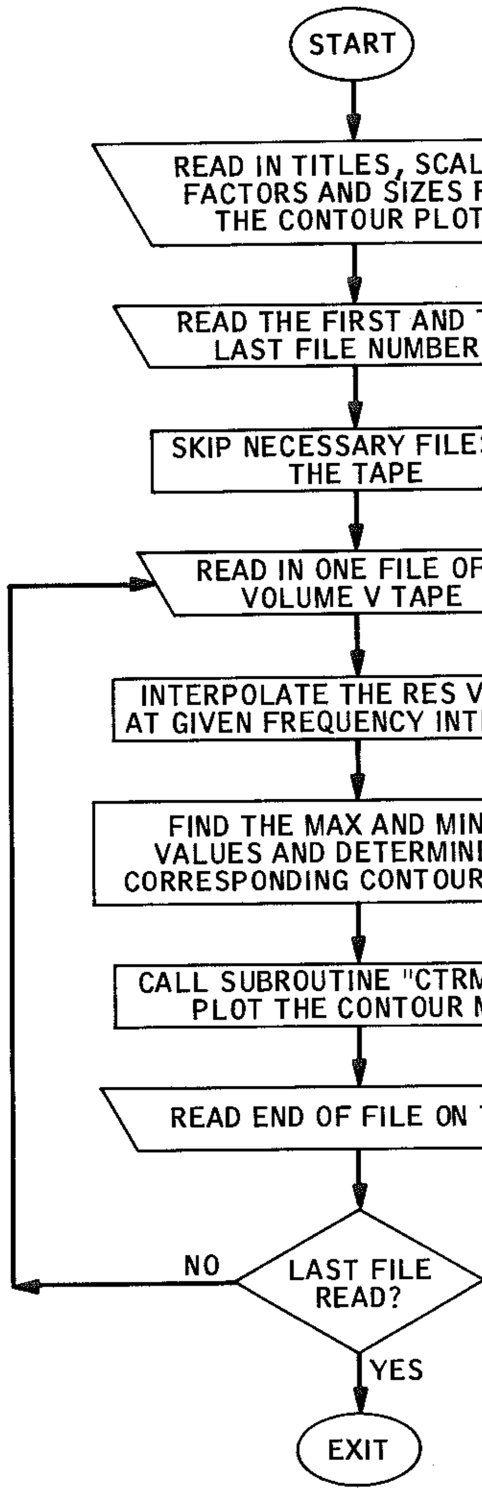
XLNGTH is the horizontal length of the whole contour map.

YLNGTH is the vertical length of the whole contour map.

LFILE is the first file number of the Vol. V tape to be read in.

MFILE is the last file number of the Vol. V tape to be read in.

RESPLT MAIN PROGRAM FLOW CHART



```
      DIMENSION DMP(5),PD(100),FR(100),A(510),B(510),F(100,350),FS(100)
      DIMENSION CORTIL(600),ICOR(100),FCOR(100),TX(5010),DIS(8),IDMP(5)
      EQUIVALENCE (FR(1),PD(1)),(NX,LLP),(VOLREF,CORTIL(481))
      COMMON/TITLE/L1,L2,L3,L4,L5,LH,LV,TTL1(20),TTL2(20),TTL3(40),
     *TTL4(20),TTL5(20),HTTL(20),VTTL(20)
      COMMON/CONCOM/KFLAG1,KFLAG2,FUNCTN,NF,TMIN,TMAX,FMIN,FMAX,
     *NDIM,NX,NY,X,Y,EXCLUD
      COMMON/BUFA/BUFA(5000)/BUFB/BUFB(5000)/BUFC/BUFC(5000)
      COMMON/BUFZ/BUFZ(510)
      COMMON/BT1/XPK(100),YPK(100),IPEAK,WIDTH
      COMMON/CPLOT2/TSCL,YSCL,XL,YL,XR,YR,SLBL
      COMMON/SPLT/NL,DI,DD1,DD2,DD(20),NML(30)
      NDIM=5000
      NY=100
      READ(5,100)TTL1,VTTL,HTTL,TTL4,TTL5
      READ(5,110)L1,LV,LH,L4,L5
      READ(5,106)NML
      READ(5,102)DIS
      READ(5,1)IDMP
      READ(5,110)NY,LSCL,NPLT,LPLT,IEXP,IPEAK,NFMAX,NL
C     IPEAK  =2,  PEAK VALUES OF EACH FREQUENCY JOINED TOGETHER
C            =1,  NO.
C    IDMP(N) =1,  NTH DAMPING RATIO RES NOT PLOTTED
C            =2,   PLOTTED.
C   IEXP =0,    RES - SD
C   IEXP =1,    RES - SD*W
C   IEXP =2,    RES - SD*W**2
C    NFMAX,    MAXIMUM NO. OF CONTOUR LEVELS ALLOWED.
C   DIS(I),I<=8  LIST OF CONTOUR SPACING IN SUITABLE UNITS TO BE CHOSEN.
C   AMIN, LOWER BOUND FOR 1ST CONTOUR LEVEL (SAME UNIT AS CONTOUR VALUES)
      READ(5,102)DI,(DD(I),I=1,NL)
      READ(5,102)WIDTH,RATIO,AMIN,FSCALE,TSPACE,FSPACE,XLNGTH,YLNGTH
  100 FORMAT(20A4)
  102 FORMAT(8F10.3)
  106 FORMAT(40A2)
  110 FORMAT(16I5)
      READ(5,1)LFILE,MFILE
    1 FORMAT(40I2)
      IF(LFILE.EQ.1)GO TO 50
      KSKFL=LFILE-2
    8 READ(10,END=9)
      GO TO 8
    9 CALL READNF(10,KSKFL)
   50 CONTINUE
      PI=3.1415962535
      DO 1000 NFILE=LFILE,MFILE
      READ(10)CORTIL,ICOR,FCOR
      L2=ICOR(59)
      L3=ICOR(60)
      DO 13 I=1,18
   13 TTL2(I)=CORTIL(180+I)
      CALL INCORE(VOLREF,4)
      READ(5,1001)RTTL
 1001 FORMAT(2X,A2)
      CALL INCORE
      CALL OUTCOR(VOLREF,N4)
      WRITE(6,1002)RTTL
 1002 FORMAT(' V',A2)
      CALL OUTCOR
      DO 14 I=1,40
   14 TTL3(I)=CORTIL(480+I)
```

```
      DELT=FCOR(61)
      NDATA=ICOR(53)
      READ(10)(TX(K),K=1,NDATA)
      SCA=1.0/980.665
      DO 51 I=1,NDATA
   51 TX(I)=TX(I)*SCA
      READ(10)ID
      READ(10)(DMP(I),I=1,ID)
      READ(10)IP
      READ(10)(PD(I),I=1,IP)
      READ(10)LLP
      IF(LLP.LE.350)GO TO 52
      WRITE(6,53)
   53 FORMAT(1H1,' NO. OF POINTS IN X-DIRECTION EXCEEDS. PROG. STOPS')
      STOP
   52 CONTINUE
      DO 10 I=1,IP
   10 FR(I)=1/PD(I)
      TMIN=0.
      TMAX=DELT*FLOAT(NDATA-1)
      GO TO (535,536),LSCL
  535 CONTINUE
      FMIN=FR(IP)
      FMAX=FR(1)
      FSTP=(FMAX-FMIN)/FLOAT(NY-1)
      DO 1041 I=1,NY
      FS(I)=FMIN+FSTP*FLOAT(I-1)
 1041 CONTINUE
      GO TO 537
  536 CONTINUE
      FMIN=ALOG10(FR(IP))
      FMAX=ALOG10(FR(1))
      FSTP=(FMAX-FMIN)/FLOAT(NY-1)
      DO 104 I=1,NY
      FS(I)=FMIN+FSTP*FLOAT(I-1)
  104 FS(I)=10.**FS(I)
  537 CONTINUE
      FZERO=0.
      IF(LPLT.EQ.2)FZERO=-5.
      DO 54 JSTP=1,NY
   54 F(JSTP,1)=FZERO
      DO 999 NDMP=1,ID
      KDMP=IDMP(NDMP)
      GO TO (55,56),KDMP
   55 DO 555 ISTP=1,IP
  555 READ(10)(A(I),I=1,LLP)
      GO TO 999
   56 CONTINUE
      WRITE (6,222)
  222 FORMAT (1H1)
      FMM=0.
      FMN=10.
      FMIN=FR(IP)
      FMAX=FR(1)
      FS(NY)=FMAX
      ISTP=1
      JSTP=NY
      READ(10)(A(I),I=1,LLP)
      READ(10)(B(I),I=1,LLP)
      WRITE(6,101)ISTP,FR(ISTP),A(2),A(LLP)
  101 FORMAT(10X,I5,5X,F10.5,10X,2F10.5)
  105 CONTINUE
```

```
      FREQ=FS(JSTP)
      IF(FREQ.GT.FR(ISTP+1))GO TO 20
      DO 11 I=1,LLP
   11 A(I)=B(I)
      ISTP=ISTP+1
      WRITE(6,101)ISTP,FR(ISTP),A(2),A(LLP)
      READ(10)(B(I),I=1,LLP)
      IF(ISTP.EQ.(IP-1))GO TO 20
      GO TO 105
   20 CONTINUE
      FRACT=(FREQ-FR(ISTP+1))/(FR(ISTP)-FR(ISTP+1))
      OMEGA=2.*PI*FREQ
      DO 21 I=2,LLP
      F(JSTP,I)=B(I)+FRACT*(A(I)-B(I))
      F(JSTP,I)=F(JSTP,I)*(OMEGA**IEXP)*FSCALE
      IF(F(JSTP,I).GT.FMM)FMM=F(JSTP,I)
      IF(F(JSTP,I).LT.FMN)FMN=F(JSTP,I)
   21 CONTINUE
      WRITE(6,31)JSTP,FREQ,F(JSTP,2),F(JSTP,LLP)
   31 FORMAT(5X,I5,5X,F10.5,2X,2F10.5)
      JSTP=JSTP-1
      IF(JSTP.GT.0)GO TO 105
      WRITE(6,102)FMN,FMM
      GO TO (316,315),LPLT
  315 FMM=ALOG10(FMM)
      FMN=ALOG10(FMN)
      DO 3155 I=2,LLP
      DO 3155 JSTP=1,NY
 3155 F(JSTP,I)=ALOG10(F(JSTP,I))
  316 CONTINUE
      LSTP=1
   33 DA=DIS(LSTP)
      AMN=DA*INT(FMN/DA)
   34 IF(AMN.GT.AMIN )GO TO 35
      AMN=AMN+DA
      GO TO 34
   35 CONTINUE
      AMM=DA*(INT(FMM/DA)+1.)
      NF=INT((AMM-AMN)/DA+.02)+1
      WRITE(6,61)DA,NF
   61 FORMAT(1X,'WITH CONTOUR SPACING OF',F6.2,' UNITS, REQUIRED NO. OF
     *LEVELS IS',I4)
      IF(NF.LE.NFMAX)GO TO 62
      LSTP=LSTP+1
      IF(LSTP.LE.8)GO TO 33
   62 CONTINUE
      IF(NF.LE.NFMAX)GO TO 63
      WRITE(6,625)
  625 FORMAT(' THE MAXIMUM NUMBER OF CONTOUR LEVELS ALLOWED IS EXCEEDED,
     * CHANGE THE SPACING OF CONTOUR LEVELS.')
      STOP
   63 CONTINUE
      BUFA(1)=AMN
      DO 32 I=2,NF
      BUFA(I)=BUFA(I-1)+DA
   32 CONTINUE
      WRITE(6,64)NF
   64 FORMAT(  /10X,'******' /1X,'NUMBER OF CONTOUR LEVELS =',I5/1X,
     *'THE CONTOUR LEVELS ARE :')
      WRITE(6,102)(BUFA(I),I=1,NF)
      CALL CTRMAP(XLNGTH,YLNGTH,TSPACE,FSPACE,NDATA,DELT,TX,F,FMM,LSCL,
     *LPLT,NPLT,DMP(NDMP),RATIO)
```

```
  999 CONTINUE
   71 READ(10,END=72)
      GO TO 71
   72 CALL READNF(10)
 1000 CONTINUE
      STOP
      END
```

Subroutine CTRMAP (Lee)

CTRMAP is called by Volume V MAIN program to plot a contour map of RES values. It is a program calling the contour plotting package "CONTUR", "LEGEND", "PLTCTR" available at Caltech for the Calcomp Plotter.

Usage

```
      CALL CTRMAP(XLNGTH, YLNGTH, TSPACE, FSPACE, NP,
      *      DELT, A, F, FMM, LSCL, LPLT, NPLT, DMPR, RATIO)
      COMMON /TITLE/L1, L2, L3, L4, L5, LH, LV, TTL1(20),
      *      TTL2(20), TTL3(40), TTL4(20), TTL5(20), HTTL(20),
      *      VTTL(20)
      COMMON /CONCOM/KFLAG1, KFLAG2, FUNCTN, NF, TMIN,
      *      TMAX, FMIN, FMAX, NDIM, NX, NY, X, Y, EXCLUD
and   COMMON /BUFA/BUFA(N)/BUFB(N)/BUFC/BUFC(N)/BUFZ/
      *      BUFZ(M)
      COMMON /SPLT/NL, DI, DD1, DD2, DD(20), NML(30)
      COMMON /CPLOT2/TSCL, YSCL, XL, YL, XR, YR, SLBL
```

Where

XLNGTH = horizontal length of the contour plot (in inches)

YLNGTH = vertical length of the contour plot (in inches)

TSPACE = spacing of horizontal labels (in sec. for time axis)

FSPACE = spacing of vertical labels (in radians/sec. for frequency)

NP = number of points of the acceleration trace

DELT = equally spaced time intervals of the acceleration data

(A(I), I=1, NP) = acceleration in G at equally spaced time intervals of DELT sec

F = (NY x NX) matrix of contour values

FMM = maximum value of contour values

LSCL = 1, vertical (frequency) axis in linear scale

2, vertical (frequency) axis in log scale

LPLT = 1, contour values given in linear scale

2, contour values given in log scale

NPLT = 0, if the corresponding acceleration trace is not plotted

1, if the corresponding acceleration trace is plotted

DMPR = damping ratio of the contour values to be plotted

RATIO = size of numeric contour labels/size of main titles

COMMON/TITLE/

TTL1, TTL2, TTL3 are the three main titles of length L1, L2, L3

TTL4, TTL5 are the titles for contour levels labels of length L4, L5

HTTL, VTTL are the X and Y axis titles (time and frequency) of length LH, LV

CONCOM = the common region used to pass control data to CONTUR

KFLAG1, KFLAG2 = branching flags. KFLAG1 must be initially set by the calling program to 1 to start a new case and is otherwise controlled by CONTUR. KFLAG2 is set by CONTUR to control branching in the calling program.

If KFLAG1 = 1, Begin new case

KFLAG1 = 2, Go to next to continue processing

KFLAG2 is set by CONTUR to be used in a computed GO TO in the user's program.

If KFLAG2 = 1, Request for function evaluation

KFLAG2 = 2, Process interrupted. Available space used up. User should plot and/or print existing contour lists and then re-enter CONTUR for further processing.

KFLAG2 = 3, Process completed. User should plot and/or print existing contour lists.

KFLAG2 = 4, Catastrophic shortage of available space. The case cannot be completed.

FUNCTN = the value of the function at the current grid point. Supplied by the calling program by computation, interpolation, or table look-up when requested by CONTUR.

NF = the desired number of contour values. NF = 0 gives no contours but simply an evaluation of function at the grid points.

TMIN, TMAX = left and right limits for x axis

FMIN, FMAX = bottom and top limits for y axis

NDIM = dimension of A, B, and C ($\leq$ N above). It should be large enough to avoid excessive interruption; N should be at least 100, and larger if storage is available.

NX, NY = number of grid lines (including the boundary lines) in X and Y directions.

X, Y = point at which the user must supply function value on request from CONTUR.

EXCLUD = a floating point number set by the user to be used by CONTUR in its "exclusion" test. This permits the user to define regions of the basic rectangular grid to be excluded from contour plotting. Each time that CONTUR receives a new value of FUNCTN, it compares FUNCTN with EXCLUD. If equality holds, then FUNCTN will not be regarded by CONTUR as a function value but rather as a flag indicating that the grid line segments immediately adjacent to the current X, Y grid point are to be excluded from the basic grid over which contouring is done.

BUFA, BUFB, BUFC = contour list buffers. The desired contour values must be stored initially by the calling program in the first NF locations of BUFA. The dimension, N, of these three arrays must be at least NDIM. The main program should be loaded first to force the loader to use the dimensions given in the main program to allocate storage for these three arrays.

BUFZ(M) = working space for CONTUR. Dimension, M, must be at least NX + 2.

COMMON/SPLT/

NL, first NL contour levels are plotted with broken lines

DI, size of the gap between broken lines

(DD(I), I=1, NL), size of the broken lines at each level

(NML(I), I=1, NF), numeric labels of each of the NF contour levels

COMMON/CPLOT2/

TSCL, FSCL, scale of X and Y coordinates

XL, YL, XR, YR, X and Y coordinates of the left lower and the right upper corner of the box for the contour map

SLBL, size of the numeric labels

COMMON/BT1/XPK(100), YPK(100), IPEAK, WIDTH

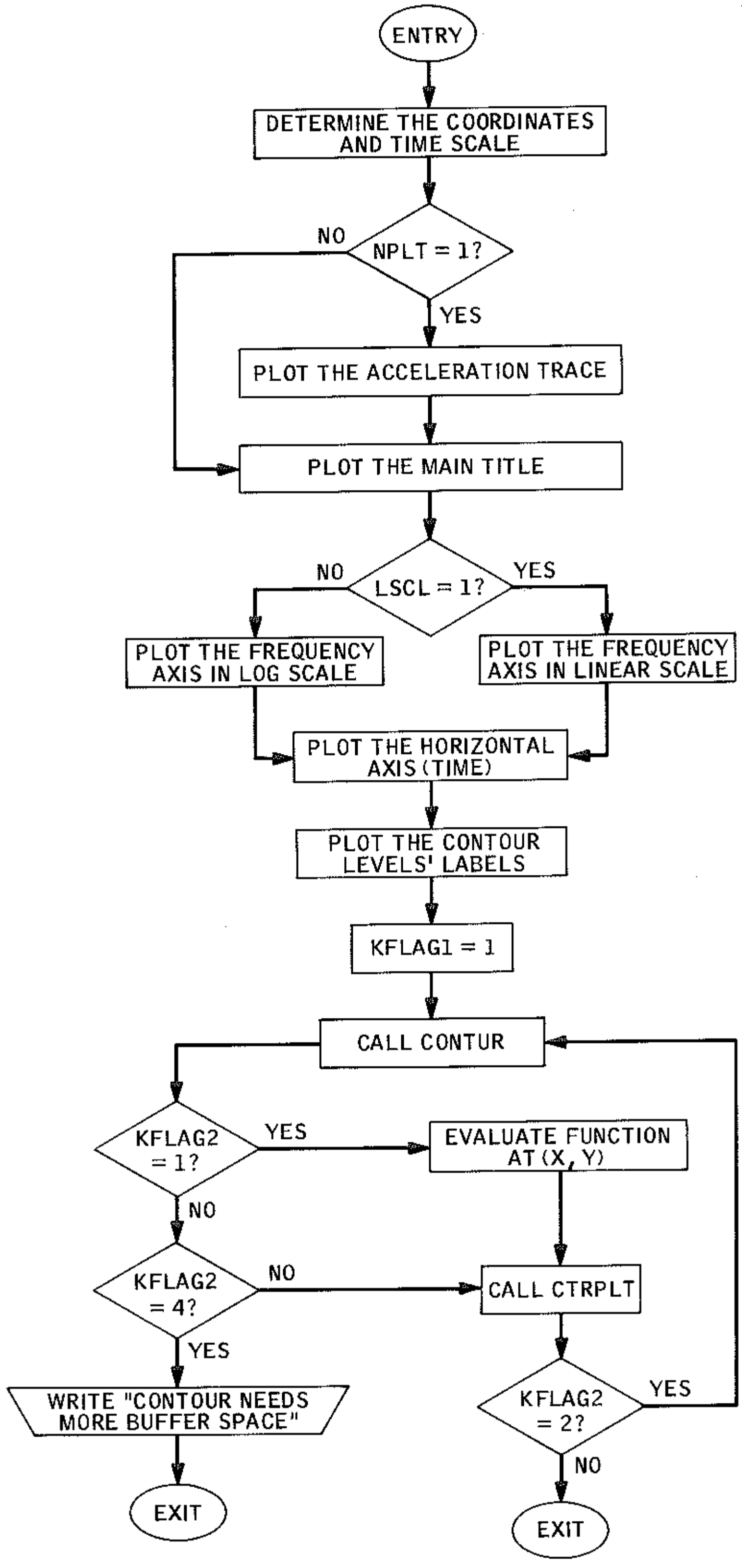
(XPK(I), YPK(I), I=1, NY), time and frequency coordinates of the peak RES amplitudes

IPEAK = 2, the above peaks joined together by a thick line

1, the above are not to be plotted.

WIDTH, "width" of the thick line

SUBROUTINE CTRMAP FLOW CHART



```
      SUBROUTINE CTRMAP(XLNGTH,YLNGTH,TSPACE,FSPACE,NP,DELT,A,F,FMM,
     *LSCL,LPLT,NPLT,DMPR,RATIO)
C  XLNGTH,  LENGTH OF HORIZONTAL INTERVAL (TIME INTERVAL).
C  YLNGTH,  TOTAL LENGTH OF Y-AXIS (FREQUENCY AND ACCELERATION AXIS).
C  TSPACE  SPACING OF TIME LABELS (IN SEC.)
C  FSPACE  SPACING OF FREQUENCY LABELS(FOR LINEAR SCALE: RADIANS/SEC)
C  NDATA,  NO. OF POINTS OF THE ACCELERATION TRACE
C  T(I),A(I)  CO-ORDUNATES OF THE ACCELERATICN TRACE
C  F,  NY BY NX MATRIX OF CONTOUR VALUES WITH MAXIMUM FMM
C  LSCL  =1, FREQUENCY AXIS IN LINEAR SCALE
C        =2, FREQUENCY AXIS IN LOG SCALE
C   LPLT=1, CONTOUR VALUES IN LINEAR SCALE
C       =2,                IN LOG SCALE
C  NPLT  =0, ACCELERATION TRACE NOT PLOTTED
C        =1, PLOTTED
C     RATIO  SIZE OF NUMERIC CONTOUR LABELS/SIZE OF MAIN TITLES
C  THE FOLLOWING IS A LIST OF LABELLED COMMON STATEMENTS TO BE SUPPLIED
C  /TITLE/  TTL1,TTL2,TTL3  ARE THE 3 MAIN TITLES OF LENGTH L1,L2,L3
C        TTL4,TTL5,  TITLES OF CONTOUR LEVELS' LABELS, OF LENGTH L4,L5
C       HTTL,VTTL, TIME & FREQ. AXIS TITLE, OF LENGTH LH,LV
C  /CONCOM/  NF, NUMBER OF CONTOUR LEVELS
C           TMIN,TMAX,  MIN. AND MAX. OF X- AXIS (TIME AXIS).
C           FMIN,FMAX,  MIN. AND MAX. OF Y- AXIS (FREQUENCY AXIS).
C           NDIM, DIMENSION OF BUFFER SPACES BUFA,BUFB,BUFC, (5000)
C           NF,  THE DESIRED NUMBER OF CONTOUR VALUES.
C           NX,NY,  NUMBER OF GRID LINES IN THE X AND Y DIRECTIONS.
C   THE DIMENSION OF BUFA,BUFB,BUFC MUST BE < OR = NDIM
C    THE DIMENSION OF BUFZ MUST BE AT LEAST NX+2
C         NL    FIRST NL CONTOUR LEVELS ARE PLOTTED WITH BROKEN LINES.
C         DI,  SIZE OF GAP BETWEEN THE BROKEN LINES.
C        DD(I),I=1,NL   SIZE OF BROKEN LINES AT EACH LEVEL.
C      NML(I),I=1,NF    NUMERIC LABELS OF EACH OF THE NF CONTOUR LEVELS.
      DIMENSION A(1),F(100,1)
      COMMON/TITLE/L1,L2,L3,L4,L5,LH,LV,TTL1(20),TTL2(20),TTL3(40),
     *TTL4(20),TTL5(20),HTTL(20),VTTL(20)
      COMMON/CONCOM/KFLAG1,KFLAG2,FUNCTN,NF,TMIN,TMAX,FMIN,FMAX,
     *NDIM,NX,NY,X,Y,EXCLUD
      COMMON/BUFA/BUFA(1)/BUFB/BUFB(1)/BUFC/BUFC(1)/BUFZ/BUFZ(1)
      COMMON/SPLT/NL,DI,DD1,DD2,DD(20),NML(30)
      COMMON/CPLOT2/TSCL,YSCL,XL,YL,XR,YR,SLPL
      COMMON/BT1/XPK(100),YPK(100),IPEAK,WIDTH
C   DETERMINE THE CO-ORDINATES AND TIME SCALE
      IPLT=0
      IF(YLNGTH.GT.10.)IPLT=1
      CALL SYSPSZ(IPLT)
      SCALE=YLNGTH/10.*XLNGTH/15.
      XL=.1*XLNGTH
      YL=.5*SCALE
      XR=XL+.8*XLNGTH
      YR=YL+.725*YLNGTH
      YT=YL+.95*YLNGTH
      YO=YT-.1*YLNGTH
      TSCL=(XR-XL)/(TMAX-TMIN)
      DIMENSION G(6),AN(2),SIGN(2)
      SIGN(1)=1.
      SIGN(2)=-1.
      STICK=.15*SCALE
      TICK=.1*SCALE
      IF(NPLT.EQ.0)GO TO 12
C   PLOT THE ACCELERATION TRACE
      CALL MAXMIN(A,NP,AMAX,AMIN)
```

```
      DATA G/.1,.2,.5,1.,1.5,2./
      AMX=AMAX1(AMAX,-AMIN)
      I=6
    1 IF(AMX.GT.G(I-1))GO TO 10
      I=I-1
      IF(I-1)10,10,1
   10 AMAX=G(I)
      ASCL=(YT-YO)/AMAX
      SLBL=.1*SCALE
      XO=XL
      DO 9 I=1,3
      YY=YT-(I-1)*.1*YLNGTH
      CALL SYSPLT(XO,YY,3)
      CALL SYSPLT(XO+TICK,YY,2)
      CALL OUTCOR(AN,NUM)
      ACC=AMAX-AMAX*(I-1)
      IF(I.NE.2)WRITE(6,1010)ACC
      IF(I.EQ.2)WRITE(6,1011)ACC
      CALL OUTCOR
      XLBL=XO-3.6*SLBL
      CALL SYSSYM(XLBL,YY-SLBL/2.,SLBL,AN,NUM*4,0.)
    9 CONTINUE
      CALL SYSSYM(XO,YO,.2*YLNGTH,13,-1,0.)
 1010 FORMAT(F4.1)
 1011 FORMAT(F4.0)
      TM=TMIN
      CALL SYSPLT(XL+TM*TSCL,YO+A(1)*ASCL,3)
      DO 11 I=2,NP
      TM=TM+DELT
      IF(TM.GT.TMAX)GO TO 12
   11 CALL SYSPLT(XL+TM*TSCL,YO+A(I)*ASCL,2)
      DO 13 I=1,3
      YY=YT-FLOAT(I-1)*.1*YLNGTH
      CALL SYSPLT(XR,YY,3)
      CALL SYSPLT(XR-TICK,YY,2)
   13 CONTINUE
      CALL SYSSYM(XR,YO,.2*YLNGTH,13,-1,0.)
   12 CONTINUE
C     PLOT THE TITLE
      YM=(YL+YR)/2.
      STL1=.15*SCALE
      XTL1=XL-1.*SCALE
      YTL1= YM-L1/2.*STL1*6./7.
      CALL SYSSYM(XTL1,YTL1,STL1,TTL1,L1,90.)
      STTL=.1*SCALE
      XTL2=XTL1+2.*STTL
      YTL2=YM-L2/2.*STTL*6./7.
      CALL SYSSYM(XTL2,YTL2,STTL,TTL2,L2,90.)
      XTL3=XTL2+2.*STTL
      YTL3=YM-L3/2.*STTL*6./7.
      CALL SYSSYM(XTL3,YTL3,STTL,TTL3,L3,90.)
      CALL SYSSYM(XTL3,YO-STTL*51./7.,STTL,'ACCELERATION IN G',17,90.)
      XTL4=XL-SCALE/4.
      YTL4=YM-LV/2.*STTL*6./7.
      CALL SYSSYM(XTL4,YTL4,STTL,VTTL,LV,90.)
C     PLOT THE PLOT BOUNDARY
      CALL SYSPLT(XL,YL,3)
      CALL SYSPLT(XL,YR,2)
      CALL SYSPLT(XR,YR,2)
      CALL SYSPLT(XR,YL,2)
      CALL SYSPLT(XL,YL,2)
C     PLOT THE FREQUENCY LABELS
```

```
      GO TO (150,200),LSCL
C     PLOT THE FREQUENCY LABELS IN LINEAR SCALE
  150 CONTINUE
      YSCL=(YR-YL)/(FMAX-FMIN)
      YSTP=YSCL*FSPACE
      I1=0
  171 I1=I1+1
      IF(FSPACE*I1.LT.FMIN)GO TO 171
      F1=FSPACE*FLOAT(I1-1)
      I2=I1
  181 I2=I2+1
      IF(FSPACE*I2.LE.FMAX)GO TO 181
      F2=FSPACE*FLOAT(I2)
      Y1=YL-(FMIN-F1)*YSCL
      Y2=YR+(F2-FMAX)*YSCL
      SLBL=.1*SCALE
      XLBL=XL-2.*SLBL
      NSTPS=I2-I1+1
      ISIGN=1
      XS=XL
  159 CONTINUE
      DO 160 I=1,NSTPS
      YY=Y1+(I-1)*YSTP
      IF(I.EQ.1.AND.F1.LT.FMIN)GO TO 300
      CALL SYSPLT(XS,YY,3)
      CALL SYSPLT(XS+STICK*SIGN(ISIGN),YY,2)
      IF(ISIGN.GT.1)GO TO 300
  161 CALL OUTCOR(AN,NUM)
      LABEL=(I1+I-2)*INT(FSPACE)
      WRITE(6,1000)LABEL
      CALL OUTCOR
      YLBL=YY-SLBL/2.
      CALL SYSSYM(XLBL,YLBL,SLBL,AN,NUM*4,0.)
  300 YS=YY
  151 YS=YS+YSCL
      IF(YS.GE.YY+YSTP.OR.YS.GE.YR)GO TO 160
      IF(YS.LE.YL)GO TO 151
      CALL SYSPLT(XS,YS,3)
      CALL SYSPLT(XS+TICK*SIGN(ISIGN),YS,2)
      GO TO 151
  160 CONTINUE
      ISIGN=ISIGN+1
      XS=XR
      GO TO (159,159,320),ISIGN
C     PLOT THE FREQUENCY LABEL IN LOG SCALE
  200 CONTINUE
      YSCL= (YR-YL)/(ALOG10(FMAX)-ALOG10(FMIN))
      I1=2
  201 I1=I1-1
      IF(FMIN.LE.10.**I1)GO TO 201
      F1=10.**I1
      I2=1
  211 I2=I2+1
      IF(FMAX.GE.10.**I2)GO TO 211
      F2=10.**I2
      Y1=YL-(ALOG10(FMIN)-I1)*YSCL
      Y2=YR+(I2-ALOG10(FMAX))*YSCL
      SLBL=.1*SCALE
      NSTPS=I2-I1
      ISIGN=1
      XS=XL
  279 CONTINUE
```

```
      DO 280 NSTP=1,NSTPS
      YY=Y1+(NSTP-1)*YSCL
      IF(NSTP.EQ.1.AND.F1.LT.FMIN)GO TO 258
      CALL SYSPLT(XS,YY,3)
      CALL SYSPLT(XS+STICK*SIGN(ISIGN),YY,2)
      IF(ISIGN.GT.1)GO TO 258
      CALL OUTCOR(AN,NUM)
      LABEL=10**(I1+NSTP-1)
      FLABEL=10.**(I1+NSTP-1)
      IF(FLABEL.LE..1)GO TO 261
      WRITE(6,1100)LABEL
      XLBL=XL-5.1*SLBL*6./7.
 1100 FORMAT(I5)
      GO TO 265
  261 CONTINUE
      II=I1+NSTP-1
      IF(II.EQ.-1)WRITE(6,1101)FLABEL
      IF(II.EQ.-2)WRITE(6,1102)FLABEL
      IF(II.EQ.-3)WRITE(6,1103)FLABEL
      XLBL=XL-(1-II)*SLBL*6.5/7.
 1101 FORMAT(F2.1)
 1102 FORMAT(F3.2)
 1103 FORMAT(F4.3)
  265 CALL OUTCOR
      YLBL=YY-SLBL/2.
      CALL SYSSYM(XLBL,YLBL,SLBL,AN,NUM*4,0.)
  258 DO 259 I=2,9
      YS=YY+ALOG10(FLOAT(I))*YSCL
      IF(YS.LE.YL.OR.YS.GE.YR)GO TO 259
      CALL SYSPLT(XS,YS,3)
      CALL SYSPLT(XS+TICK*SIGN(ISIGN),YS,2)
      IF(I.EQ.5)CALL SYSPLT(XS+1.25*TICK*SIGN(ISIGN),YS,2)
  259 CONTINUE
  280 CONTINUE
      ISIGN=ISIGN+1
      XS=XR
      GO TO (279,279,319),ISIGN
  319 FMIN=ALOG10(FMIN)
      FMAX=ALOG10(FMAX)
C   PLOT THE HORIZONTAL LABELS (TIME AXIS)
  320 CONTINUE
      XLBL=(XR+XL)/2.-LH*3./7.*SLBL
      YLBL=YL-3.*SLBL
      CALL SYSSYM(XLBL,YLBL,SLBL,HTTL,LH,0.)
      XSTP=TSCL*TSPACE
      SLBL=.1*SCALE
      I1=0
  311 I1=I1+1
      IF(TSPACE*I1.LT.TMIN)GO TO 311
      T1=TSPACE*FLOAT(I1-1)
      I2=I1
  312 I2=I2+1
      IF(TSPACE*I2.LE.TMAX)GO TO 312
      T2=TSPACE*I2
      X1=XL-(TMIN-T1)*TSCL
      X2=XR+(T2-TMAX)*TSCL
      NSTPS=I2-I1+1
      YS=YL
      ISIGN=1
  305 CONTINUE
      DO 310 I=1,NSTPS
      XX=X1+(I-1)*XSTP
```

```
      IF(I.EQ.1.AND.T1.LT.TMIN)GO TO 301                                CTRM 248
      CALL SYSPLT(XX,YS,3)                                              CTRM 249
      CALL SYSPLT(XX,YS+STICK*SIGN(ISIGN),2)                            CTRM 250
      CALL OUTCOR(AN,NUM)                                               CTRM 251
      LABEL=(I1+I-2)*INT(TSPACE)                                        CTRM 252
      WRITE(6,1002)LABEL                                                CTRM 253
      CALL OUTCOR                                                       CTRM 254
      XLBL=XX-SLBL*6./7.*2.5                                            CTRM 255
      IF(LABEL.GE.10.)XLBL=XX-SLBL*6./7.*2.                             CTRM 256
      IF(LABEL.GE.100.)XLBL=XX-SLBL*6./7.*1.5                           CTRM 257
      IF(ISIGN.EQ.1)CALL SYSSYM(XLBL,YL-1.5*SLBL,SLBL,AN,NUM*4,0.)      CTRM 258
      IF(ISIGN.EQ.2)CALL SYSSYM(XLBL,YR+.5*SLBL,SLBL,AN,NUM*4,0.)       CTRM 259
  301 XS=XX                                                             CTRM 260
  309 XS=XS+TSCL                                                        CTRM 261
      IF(XS.GE.XX+XSTP.OR.XS.GE.XR)GO TO 310                            CTRM 262
      IF(XS.LE.XL)GO TO 309                                             CTRM 263
      CALL SYSPLT(XS,YS,3)                                              CTRM 264
      CALL SYSPLT(XS,YS+TICK*SIGN(ISIGN),2)                             CTRM 265
      GO TO 309                                                         CTRM 266
  310 CONTINUE                                                          CTRM 267
      ISIGN=ISIGN+1                                                     CTRM 268
      YS=YR                                                             CTRM 269
      GO TO(305,305,315),ISIGN                                          CTRM 270
  315 CONTINUE                                                          CTRM 271
 1000 FORMAT(I2)                                                        CTRM 272
 1002 FORMAT(I3)                                                        CTRM 273
C    PLOT THE CONTOUR LEVELS' LABELS                                    CTRM 274
      YI=(YR-YL-4.5*STTL)/FLOAT(NF+4)                                   CTRM 275
      XM=XP+(XLNGTH-XF)/2.                                              CTRM 276
      XTTL=XM- 5.*STTL*6./7.                                            CTRM 277
      YTTL=YR-STTL                                                      CTRM 278
      CALL SYSSYM(XTTL,YTTL,STTL,'PEAK VALUE',10,0.)                    CTRM 279
      CALL OUTCOR(AN,NUM)                                               CTRM 280
      FMX=FMM                                                           CTRM 281
      IF(LPLT.EQ.2)FMX=10.**FMM                                         CTRM 282
      IF(FMX.LT.100.)WRITE(6,1012)FMX                                   CTRM 283
      IF(FMX.GE.100.)WRITE(6,1015)FMX                                   CTRM 284
 1012 FORMAT(F6.3)                                                      CTRM 285
 1015 FORMAT(F6.2)                                                      CTRM 286
      CALL OUTCOR                                                       CTRM 287
      XTTL=XM-6.5*STTL*6./7.                                            CTRM 288
      YTTL=YTTL-2.*STTL                                                 CTRM 289
      CALL SYSSYM(XTTL,YTTL,STTL,AN,8,0.)                               CTRM 290
      XTTL=XTTL+ 6.*STTL                                                CTRM 291
      DIMENSION UNIT(3)                                                 CTRM 292
      EQUIVALENCE (UNIT(1),TTL4(3))                                     CTRM 293
      CALL SYSSYM(XTTL,YTTL,STTL,UNIT,12,0.)                            CTRM 294
      YTTL=YTTL-2.*YI                                                   CTRM 295
      XTTL=XM-36./7.*STTL                                               CTRM 296
      CALL SYSSYM(XTTL,YTTL,STTL,'DAMPING = ',10,0.)                    CTRM 297
      CALL OUTCOR(AN,NUM)                                               CTRM 298
      WRITE(6,1013)DMPR                                                 CTRM 299
 1013 FORMAT(F3.2)                                                      CTRM 300
      CALL OUTCOR                                                       CTRM 301
      XTTL=XTTL+10.*STTL*6./7.                                          CTRM 302
      CALL SYSSYM(XTTL,YTTL,STTL,AN,NUM*4,0.)                           CTRM 303
      YTTL=YTTL-2.*YI                                                   CTRM 304
      XTTL=XM-3./7.*STTL*FLOAT(L4)                                      CTRM 305
      CALL SYSSYM(XTTL,YTTL,STTL,TTL4,L4,0.)                            CTRM 306
      YTTL=YTTL-1.5*STTL                                                CTRM 307
      XTTL=XM-3./7.*STTL*(FLOAT(L5)-.5)                                 CTRM 308
      CALL SYSSYM(XTTL,YTTL,STTL,TTL5,L5,0.)                            CTRM 309
```

```
      YI=(YTTL-YL)/FLOAT(NF)
      XTTL=XM-6.5*STTL*6./7.
      DO 500 I=1,NF
      YTTL=YTTL-YI
      CALL OUTCOR(AN,NUM)
      BUFFER=BUFA(I)
      IF(LPLT.EQ.2)BUFFER=10.**BUFA(I)
      WRITE(6,1014)I,BUFFER
      CALL OUTCOR
      CALL SYSSYM(XTTL,YTTL,STTL,AN,NUM*4,0.)
  500 CONTINUE
 1014 FORMAT(I3,F11.3)
C    PLOT THE THICK LINE JOINING PEAKS OF DIFFERENT FREQUENCY
C    IPEAK  = 2, PEAKS OF DIFFERENT FREQUENCY JOINED TOGETHER.
C           = 1,  NOT PLOTTED.
      GO TO (999,899),IPEAK
  899 CONTINUE
      XCOUNT=FLOAT(NX-1)
      YCOUNT=FLOAT(NY-1)
      DO 910 JSTP=1,NY
      PEAK=F(JSTP,1)
      DO 900 I=2,NX
      IF(F(JSTP,I).LE.PEAK)GO TO 900
      PEAK=F(JSTP,I)
      MAXI=I
  900 CONTINUE
      XPK(JSTP)=XL+(XR-XL)*FLOAT(MAXI-1)/XCOUNT
      YPK(JSTP)=YL+(YR-YL)*FLOAT(JSTP-1)/YCOUNT
  910 CONTINUE
      DS=WIDTH*SCALE
      DT=-3.*DS
      DO 950 ICOUNT=1,3
      IF(ICOUNT.GT.1)CALL THLN(NY,DT)
      CALL SYSPLT(XPK(1),YPK(1),3)
      DO 920 I=2,NY
      XX=AMAX1(XL,XPK(I))
      XX=AMIN1(XX,XR)
  920 CALL SYSPLT(XX,YPK(I),2)
  950 DT=DT+DS*2.
  999 CONTINUE
C    PLOT THE CONTOUR
      SLBL=RATIO*SLBL
      IY=1
      JX=1
      KFLAG1=1
  111 CONTINUE
      CALL CONTUR
      GO TO (21,22,22,24),KFLAG2
   21 FUNCTN=F(IY,JX)
      JX=JX+1
      IF(JX  -NX)111,111,2111
 2111 IY=IY+1
      JX=1
      IF(IY  -NY)111,111,22
   22 CONTINUE
      CALL CTRPLT
      GO TO (99,111,99),KFLAG2
   24 WRITE(6,1030)
 1030  FORMAT(31HOCONTUR NEEDS MORE BUFFER SPACE)
   99 CONTINUE
      RETURN
      END
```

Subroutine CTRPLT (Lee)

This subroutine is called by Volume V subroutine CTRMAP to plot and label contours generated by CONTUR. It is a modification of "PLTCTR", one of the subroutines of the contour plotting package available at Caltech for the Calcomp and Stromberg-Carlson Plotters ("LEGEND", "CONTUR", PLTCTR").

Method

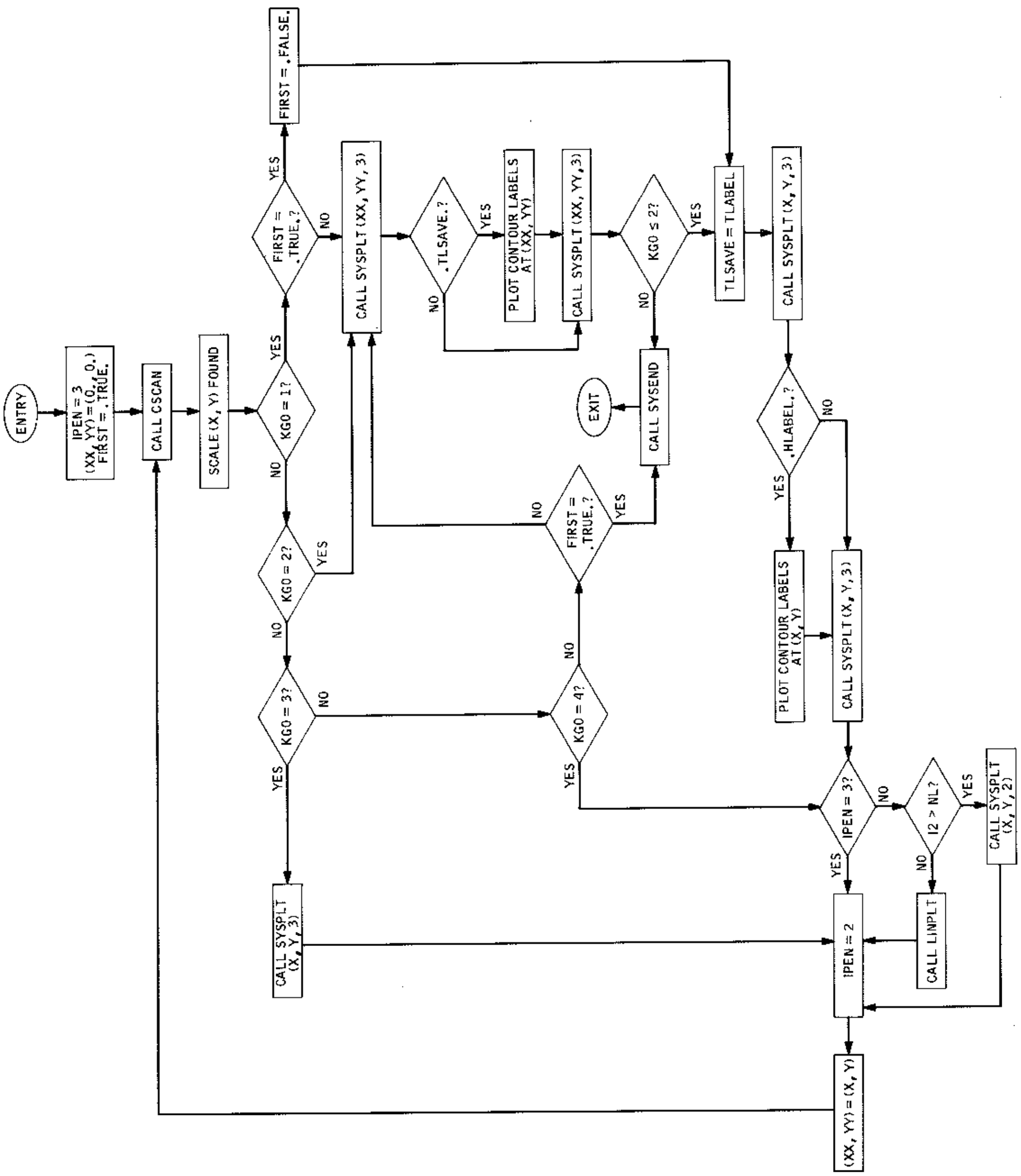
The subroutine uses CSCAN to obtain coordinate pairs (X, Y) one pair at a time from the list constructed by CONTUR. It plots the contours, which are labeled according to the control flags set by CSCAN. Before labeling, the program checks to make sure that no overlapping of labels occur. If this happens, the labeling is cancelled.

Usage

```
CALL CTRPLT
COMMON/CONCOM/...
COMMON/CPLOT2/...
COMMON/SPLT/...
COMMON/SCOM/KSTART,KGO,HLABEL,TLABEL,IZ,Z,
*      IARC,X,Y
```

For the first three common regions, refer to subroutine "CTRMAP". The common region SCOM is used in the subroutine "CSCAN" available at Caltech for the Calcomp and Stromberg-Carlson Plotters.

SUBROUTINE CTRPLT FLOW CHART



```
      SUBROUTINE CTRPLT                                                 CTRP   1
      COMMON/CONCOM/KFLAG1,KFLAG2,FUNCTN,NF,XMIN,XMAX,YMIN,YMAX,NDIM,   CTRP   2
     1 NX,NY,XX,YY,EXCLUD                                               CTRP   3
      COMMON /SCOM/KSTART,KGO,HLABEL,TLABEL,IZ,   Z,IARC,X,Y            CTRP   4
      COMMON/CPLOT2/TSCL,YSCL,XL,YL,XR,YR,SLBL                          CTRP   5
      COMMON/SPLT/NL,DI,DD1,DD2,DD(20),NUM(30)                          CTRP   6
      DIMENSION XXLBL(500),YYLBL(500)                                   CTRP   7
      NLBL=0                                                            CTRP   8
      SSLBL=1.3*SLBL                                                    CTRP   9
      IPEN=3                                                            CTRP  10
      XX=0.                                                             CTRP  11
      YY=0.                                                             CTRP  12
      DD1=0.                                                            CTRP  13
      DD2=0.                                                            CTRP  14
      LOGICAL FIRST,HLABEL,TLABEL,TLSAVE                                CTRP  15
      KSTART = 1                                                        CTRP  16
      FIRST = .TRUE.                                                    CTRP  17
   10 CALL CSCAN                                                        CTRP  18
      X=XL+(X-XMIN)*TSCL                                                CTRP  19
      Y=YL+(Y-YMIN)*YSCL                                                CTRP  20
      GO TO (12,24,15,32,20),KGO                                        CTRP  21
   12 IF(.NOT.FIRST) GO TO 24                                           CTRP  22
      FIRST = .FALSE.                                                   CTRP  23
      GO TO 30                                                          CTRP  24
   15 CONTINUE                                                          CTRP  25
      CALL SYSPLT(X,Y,3)                                                CTRP  26
      GO TO 33                                                          CTRP  27
   20 IF(FIRST) GO TO 40                                                CTRP  28
   24 CONTINUE                                                          CTRP  29
      CALL SYSPLT(XX,YY,3)                                              CTRP  30
      IF(.NOT.TLSAVE)GO TO 27                                           CTRP  31
      XLBL=XX                                                           CTRP  32
      YLBL=YY                                                           CTRP  33
      IF(XX.GE.XR-SLBL)XLBL=XR-SLBL                                     CTRP  34
      IF(YY.GE.YR-SLBL)YLBL=YR-1.5*SLBL                                 CTRP  35
      IF(NLBL.EQ.0)GO TO 255                                            CTRP  36
      DO 25 I=1,NLBL                                                    CTRP  37
      DX=ABS(XXLBL(I)-XLBL)                                             CTRP  38
      DY=ABS(YYLBL(I)-YLBL)                                             CTRP  39
      IF(DX.GE.SLBL)GO TO 25                                            CTRP  40
      IF(DY.LT.SSLBL)GO TO 27                                           CTRP  41
   25 CONTINUE                                                          CTRP  42
  255 NLBL=NLBL+1                                                       CTRP  43
      IF(NLBL.GT.500)GO TO 40                                           CTRP  44
      XXLBL(NLBL)=XLBL                                                  CTRP  45
      YYLBL(NLBL)=YLBL                                                  CTRP  46
   26 CALL SYSSYM(XLBL,YLBL,SLBL,NUM(IZ   ),2,0)                        CTRP  47
   27 CONTINUE                                                          CTRP  48
      DD1=0.                                                            CTRP  49
      DD2=0.                                                            CTRP  50
       CALL SYSPLT(XX,YY,3)                                             CTRP  51
      GO TO (30,30,99,99,36),KGO                                        CTRP  52
   30 TLSAVE = TLABEL                                                   CTRP  53
      CALL SYSPLT(X,Y,3)                                                CTRP  54
      IF(.NOT.HLABEL)GO TO 31                                           CTRP  55
      XLBL=X                                                            CTRP  56
      YLBL=Y                                                            CTRP  57
      IF(X.GE.XR-SLBL)XLBL=XR-SLBL                                      CTRP  58
      IF(Y.GE.YR-SLBL)YLBL=YR-1.5*SLBL                                  CTRP  59
      IF(NLBL.EQ.0)GO TO 3055                                           CTRP  60
      DO 305 I=1,NLBL                                                   CTRP  61
```

```
      DX=ABS(XXLBL(I)-XLBL)
      DY=ABS(YYLBL(I)-YLBL)
      IF(DX.GE.SLBL)GO TO 305
      IF(DY.LT.SSLBL)GO TO 31
  305 CONTINUE
 3055 NLBL=NLBL+1
      IF(NLBL.GT.500)GO TO 40
      XXLBL(NLBL)=XLBL
      YYLBL(NLBL)=YLBL
  306 CALL SYSSYM(XLBL,YLBL,SLBL,NUM(IZ  ),2,0)
   31 CONTINUE
      CALL SYSPLT(        X,Y,3)
   32 CONTINUE
      IF(IPEN.EQ.3)GO TO 33
      IF(KGO.EQ.4.AND.IZ.LE.NL)CALL LINPLT(XX,YY,X,Y,DD(IZ))
      IF(IZ.GT.NL)CALL SYSPLT(X,Y,IPEN)
   33 CONTINUE
      IPEN=2
      XX = X
      YY = Y
      GO TO 10
   36 CONTINUE
      IF(KFLAG2.EQ.2) RETURN
   99 CONTINUE
   40 CONTINUE
      IF(NLBL.GT.500)WRITE(6,101)
  101 FORMAT(' MORE STORAGE SPACE FOR CONTOUR LABEL CO-ORDINATES.')
      CALL SYSEND  (1,1)
      RETURN
      END
```

Subroutine LINPLT (Lee)

This subroutine is called by the Volume V subroutine CTRPLT to plot a given contour level with a broken line from (X1, Y1) to (X2, Y2).

Usage

CALL LINPLT(X1, Y1, X2, Y2, D1)

COMMON/SPLT/...

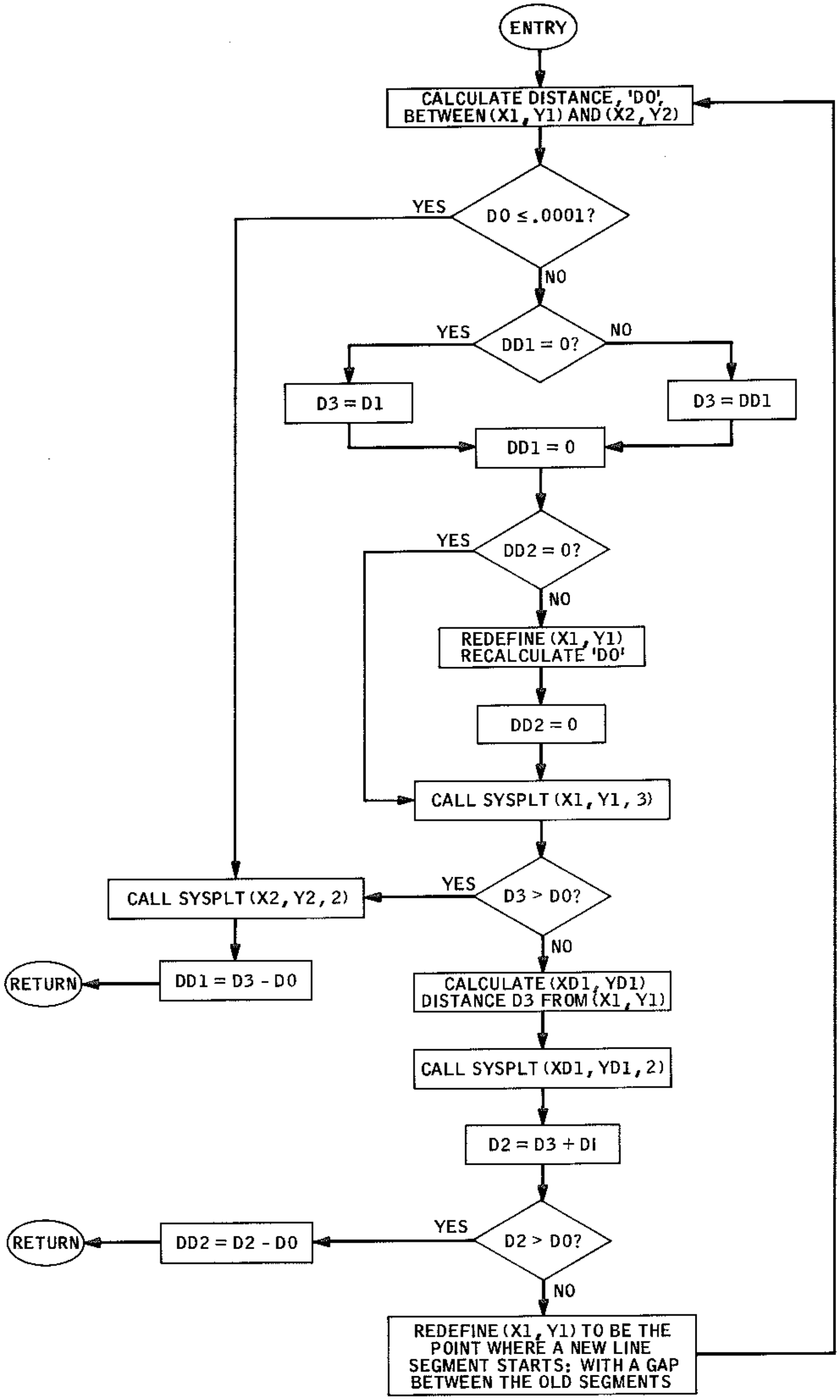
Refer to the subroutine "CTRMAP" for description of the common region SPLT

(X1, Y1) = coordinates of the first point where the contour line starts.

(X2, Y2) = coordinates of the terminal point where the contour line ends.

D1 = size of each segment of broken lines.

SUBROUTINE LINPLT FLOW CHART



```
      SUBROUTINE LINPLT(X1,Y1,X2,Y2,D1)
      COMMON/SPLT/NL,DI,DD1,DD2,DD(20),NUM(30)
   12 D0=SQRT((X2-X1)**2+(Y2-Y1)**2)
      IF(D0.LE..0001)GO TO 14
      D3=D1
      IF(DD1.NE.0.)D3=DD1
      DD1=0.
      IF(DD2.EQ.0.)GO TO 13
      X1=(X2-X1)*DD2/D0+X1
      Y1=(Y2-Y1)*DD2/D0+Y1
      D0=SQRT((X2-X1)**2+(Y2-Y1)**2)
      DD2=0.
   13 CALL SYSPLT(X1,Y1,3)
      IF (D3.GT.D0)GO TO 14
      XD1=(X2-X1)*D3/D0+X1
      YD1=(Y2-Y1)*D3/D0+Y1
      CALL SYSPLT(XD1,YD1,2)
      D2=D3+DI
      IF(D2.GT.D0)GO TO 15
      X1=(X2-X1)*D2/D0+X1
      Y1=(Y2-Y1)*D2/D0+Y1
      GO TO 12
   14 CALL SYSPLT(X2,Y2,2)
      DD1=D3-D0
      GO TO 16
   15 DD2=D2-D0
   16 RETURN
      END
```

```
      SUBROUTINE THLN(IP,DS)
      COMMON /BT1/ VM(100),PD(100)
C     ROUTINE FOR THICKENNING THE LINE
      H1=PD(2)-PD(1)
      V1=VM(2)-VM(1)
      HP1=SQRT(H1*H1+V1*V1)
      C1=H1/HP1
      T1=V1/H1
      VMM=VM(1)
      VM(1)=VM(1)-DS*C1
      IPM=IP-1
      XAM=PD(1)+DS*V1/HP1
      DO 17 I=2,IPM
      H2=PD(I+1)-PD(I)
      V2=VM(I+1)-VM(I)
      HP2=SQRT(H2*H2+V2*V2)
      C2=H2/HP2
      T2=V2/H2
      AR2=T1-T2
      IF (AR2.NE.0.0) GO TO 20
      PD(I-1)=XAM
      XAM=PD(I)+DS*V2/HP2
      VMM=VM(I)
      VM(I)=VM(I)-DS*C2
      GO TO 21
   20 CONTINUE
      AR1=VM(I)-VMM + DS/C1 - DS/C2 + T1*PD(I-1) - T2*PD(I)
      VMM=VM(I)
      PD(I-1)=XAM
      XAM=AR1/AR2
      VM(I)=VM(I) - DS/C2 + T2*(XAM-PD(I))
   21 CONTINUE
      C1=C2
      T1=T2
   17 CONTINUE
      PD(IPM)=XAM
      VM(IP)=VM(IP)-DS*C2
      PD(IP)=PD(IP)+DS*V2/HP2
      RETURN
      END
```

REFERENCES

Cooley, J. W. and J. W. Tukey (1965). An Algorithm for the Machine Calculation of Complex Fourier Series, Mathematics of Computations, Vol. 19, p. 297.

Housner, G. W. and M. D. Trifunac (1967). Analysis of Accelerograms - Parkfield Earthquake, Bull. Seis. Soc. Amer., 57, 1193-1220.

Hudson, D. E., A. G. Brady, and M. D. Trifunac (1969). Strong Motion Earthquake Accelerograms, Digitized and Plotted Data, Vol. I, Part A, Earthquake Engineering Research Laboratory, Report No. EERL 69-20, Calif. Inst. of Tech., Pasadena.

Hudson, D. E., A. G. Brady, M. D. Trifunac, and A. Vijayaraghavan (1971). Strong-Motion Earthquake Accelerograms, Corrected Accelerograms and Integrated Velocity and Displacement Curves, Vol. II, Part A, Earthquake Engineering Research Laboratory, Report No. EERL 71-50, Calif. Inst. of Tech., Pasadena.

Hudson, D. E., M. D. Trifunac, and A. G. Brady (1972a). Analyses of Strong-Motion Earthquake Accelerograms, Response Spectra, Vol. III, Part A, Earthquake Engineering Research Laboratory, Report No. EERL 72-80, Calif. Inst. of Tech., Pasadena.

Hudson, D. E., M. D. Trifunac, F. E. Udwadia, A. Vijayaraghavan and A. G. Brady (1972b). Analyses of Strong-Motion Earthquake Accelerograms, Fourier Amplitude Spectra, Vol. IV, Part A, Earthquake Engineering Research Laboratory, Report No. EERL 72-100, Calif. Inst. of Tech., Pasadena.

Iwan, W. D. (1960). Digital Calculation of Response Spectra and Fourier Spectra, Unpublished Note, Calif. Inst. of Tech., Pasadena.

Nigam, N. C. and P. C. Jennings (1968). Digital Calculation of Response Spectra from Strong-Motion Earthquake Records, Earthquake Engineering Research Laboratory, Calif. Inst. of Tech., Pasadena.

Nowroozi, A. A. (1967). Table for Fisher's Test of Significance in Harmonic Analysis, Geophys. J. R. Astr. Soc., 12, 517-520.

Trifunac, M. D. (1970). Low Frequency Digitization Errors and a New Method for Zero Baseline Correction of Strong-Motion Accelerograms, Earthquake Engineering Research Laboratory, Report No. EERL 70-07, Calif. Inst. of Tech., Pasadena.

Trifunac, M. D. and J. N. Brune (1970). Complexity of Energy Release during the Imperial Valley, California, Earthquake of 1940, Bull. Seis. Soc. Amer., 60, 137-160.

Trifunac, M. D. (1971). Response Envelope Spectrum and Interpretation of Strong Earthquake Ground Motion, Bull. Seis. Soc. Amer., 61, 343-356.

Trifunac, M. D., F. E. Udwadia, and A. G. Brady (1971). High Frequency Errors and Instrument Corrections of Strong-Motion Accelerograms, Earthquake Engineering Research Laboratory, Report No. EERL 71-05, Calif. Inst. of Tech., Pasadena.

APPENDIX

This appendix lists the write-ups and the programs that are part of the routine library available at the Willis H. Booth Computing Center of the California Institute of Technology. The contents of this appendix are reproduced with permission from the Computing Center Staff.

IDENTIFICATION

ABCD/Change an Integer to EBCDIC for Printing* -

ASSEMBLER coded

November, 1971

USAGE

There are three entry points - each returns high order zeros for N less than zero. If N is greater than or equal to zero, high order blanks are returned.

The three entry points are:

(a) X = ABCD(N) (for a real variable X)

(b) K = IABCD(N) (for an integer variable K)

(c) Double Precision D, ABCD8

D=ABCD8(N)

For cases (a) and (b) use A4 format for 370. (A4 for PDP-10.) For case (c) use A8 for 370. (A10 for PDP-10.)

EXAMPLES:

```
      N=1
      X=ABCD(N)
C     X contains bbbl in A4 FORMAT (where b is a blank)
      N=-1
      X=ABCD(N)
C     X contains 0001 in A4 FORMAT
```

STORAGE: $(216)_{10}$

---

* Willis H. Booth Computing Center Report No. C868-253-370-10.

<u>NOTE:</u>

These routines are on the 370/155 FORTRAN library and PREST library.

These routines are also available on the PDP-10.

```
./  PRINT  LIST
BCD       TITLE 'IABCD && ABCD ROUTINES'
* ABCD(IABCD)                                   DATE OF OBJECT DECK 08-05-68
          SPACE 2
***   ROUTINE TO CONVERT INTEGER TO ZONED FOR PRINTING
*     BY A FORTRAN PROGRAM USING 'A' FORMAT.
*
*     CALLING SEQUENCE...
*
*      Y=ABCD(N)    OR
*      K=IABCD(N)
*
*     N + --> H.O. BLANKS
*       - --> H.O. ZEROS
*
*     REGISTER ASSIGNMENTS
*
ABCD      START X'10000'
XBASE     EQU   12                     BASE REG
FRO       EQU   0                      FLOATING REG 0
          EJECT
          SPACE 2
***   ENTRY POINTS FOR IABCD & ABCD
*
*
          ENTRY IABCD
ABCD      CSECT
IABCD     SAVE  (14,12),,*             SAVE REGS
          LR    XBASE,15
          USING ABCD,XBASE       #XBASE
          LA    15,SAREA
          ST    13,4(0,15)             CHAIN SAVE AREAS
          ST    15,8(0,13)
          LR    13,15
          L     1,0(0,1)               GET CALLER'S N VALUE
          L     1,0(0,1)
          ST    1,N                    SAVE N
          LPR   1,1                    FORCE N POSITIVE
          CVD   1,PACKED               CHANGE TO PACKED
          UNPK  ZONED,PACKED           THEN TO ZONED
          BZ    BCDO
          TM    N,X'80'                H.O. BLANKS WANTED?
          BO    BCD1                   NO
BCDO      CLI   ZONED,C'0'             YES - PUT THEM IN
          BNE   BCD1
          MVI   ZONED,C' '
          CLI   ZONED+1,C'0'
          BNE   BCD1
          MVI   ZONED+1,C' '
          CLI   ZONED+2,C'0'
```

```
         BNE    BCD1
         MVI    ZONED+2,C' '
BCD1     OI     ZONED+3,X'F0'          ZONE OUT + SIGN
         LE     FR0,ZONED              RESULT TO FR0
         L      1,ZONED                AND TO GEN REG 0
         L      13,4(0,13)
         ST     1,20(0,13)
         RETURN  (14,12)
         SPACE 2
***   STORAGE AREA
*
*
PACKED   DS     D
ZONED    DS     CL4
SAREA    DS     18F
N        DS     F
         END    ABCD
```

IDENTIFICATION

CONTUR*† (Stromberg-Carlson)

CONTUR (Calcomp)

Contour plotting package with user's choice of Stromberg-Carlson plots or Calcomp plots, respectively

JPL Section 314, Report No. 106, May, 1965

Modified by David Cartwright, C.I.T., August, 1967

I. IDENTIFICATION

CONTUR/Contour plotting routine

PURPOSE

To construct strings of (x, y) values representing contours along which a function f = f(x, y) has specified constant values.

METHOD

The rectangle of definition of the function is overlaid with a grid of a specified fineness, and each square of the grid is processed for all contour values given. Thus, each square is processed by the program only once, and each grid value of f is computed but once. List processing methods are applied to keep track of which contours passed through which squares, i.e., the lists are threaded by pointers.

---

* This is an abbreviated version of C967-229-370 Caltech write-up containing CONTUR, LEGENS, PLTCTS, LEGEND and PLTCTR. (Dated April, 1972)

† Willis H. Booth Computing Center Report No. C967-229-370.

USAGE

Calling Sequence: CALL CONTUR

The necessary buffer regions and control data are stored by the calling program in labeled common.

The main program calling CONTUR must have the following:

```
COMMON/CONCOM/KFLAG1,KFLAG2,FUNCTN,NF,XMIN,
*XMAX,YMIN,YMAX,NDIM,NX,NY,X,Y,EXCLUD
COMMON/BUFA/A(N)
COMMON/BUFB/B(N)
COMMON/BUFC/C(N)
COMMON/BUFZ/Z(M)
```

CONCOM = the common region used to pass control data to CONTUR

KFLAG1, KFLAG2 = branching flags. KFLAG1 must be initially set by the calling program to 1 to start a new case and is other-wise controlled by CONTUR. KFLAG2 is set by CONTUR to control branching in the calling program.

KFLAG1 must be initially set = 1 by the user. CONTUR resets KFLAG1=2.

If KFLAG1 = 1, Begin new case

KFLAG1 = 2, Go to next to continue processing.

KFLAG2 is set by CONTUR to be used in a computed GO TO in the user's program.

If KFLAG2 = 1, Request for function evaluation

KFLAG2 = 2, Process interrupted. Available space used up. User should plot and/or print existing contour lists and then re-enter CONTUR for further processing.

KFLAG2 = 3, Process completed. User should plot and/or print existing contour lists.

KFLAG2 = 4, Catastrophic shortage of available space. The case cannot be completed.

FUNCTN = the value of the function at the current grid point. Supplied by the calling program by computation, interpolation, or table look-up when requested by CONTUR.

NF = the desired number of contour values. NF = 0 gives no contours, but simply an evaluation of the function at the grid points.

XMIN, XMAX
YMIN, YMAX = the X and Y limits.

NDIM = dimension of A, B, and C. (≤ N above). It should be large enough to avoid excessive interruption; N should be at least 100, and larger if storage is available.

NX, NY = number of grid lines (including the boundary lines) in X and Y directions.

X, Y = point at which the user must supply function value on request from CONTUR.

EXCLUD = a floating point number set by the user to be used by CONTUR in its "exclusion" test. This permits the user to define regions of the basic rectangular grid to be excluded from contour plotting. Each time that CONTUR receives a new value of FUNCTN, it compares FUNCTN with EXCLUD. If equality holds, then

FUNCTN will not be regarded by CONTUR as a function value but rather as a flag indicating that the grid line segments immediately adjacent to the current X, Y grid point are to be excluded from the basic grid over which contouring is done.

A, B, C = contour list buffers. The desired contour values must be stored initially by the calling program in the first NF locations of BUFA. The dimension, N, of these three arrays must be at least NDIM. The main program should be loaded first to force the loader to use the dimensions given in the main program to allocate storage for these three arrays.

Z(M) = working space for CONTUR. Dimension, M, must be at least NX + 2.

<u>EXAMPLE:</u>

```
      CALL LEGEND (AA, TITLE, NF, XMIN, XMAX, YMIN, YMAX, CX)
          or
      CALL LEGENS (See write-up of LEGEND for explanation).
      KFLAG1 = 1
11    CALL CONTUR
      GO TO (21, 22, 23, 24), KFLAG2
21    CONTINUE      (Evaluate function at (X, Y) and store value in
                    FUNCTN.)
      GO TO 11
22    CONTINUE      (Available space temporarily used up.  Plot
                    or print contour lists.)
      GO TO 11
```

23 CONTINUE (Processing completed. Plot or print contour lists.)

GO TO next case

24 CONTINUE (Available space inadequate for job. Discontinue processing.)

At 22 and 23, the subroutines PLTCTR or PLTCTS may be called to plot the contour lists. The function evaluation at 21 may be by direct computation of a closed formula, by interpolation in a table or direct lookup, or by reading values from prepared input units, At 24, processing of job requires increasing NDIM, which in general can be done only by recompiling the main program to increase the dimensions of A, B, and C.

Subroutines used by CALCOMP CONTUR

CSCAN

CTRIM

LABEL

LEGEND

OUTCOR

PLTCTR

SCALE

+ System CALCOMP plotting routines SYSPIO, SYSEND

Stromberg-Carlson 4060 Subroutines contained in SSS.SC4060, CITSX2

1. ARCPLG
2. CIRARG
3. CIRPLG
4. CLOSE
5. CONVTG
6. DATIME
7. ERRZZ
8. EXITG
9. FAIRG
10. FMTSG
11. FONT2
12. GETCZZ
13. GETSMG
14. GRAFG
15. GRAPHG
16. GRIDG
17. IFMZZ
18. JOBZZ
19. KADZ
20. KWKBRG
21. KWPOLG
22. LABELG
23. LEGNDG
24. LINESG
25. MAKBRG
26. METAZZ
27. MLTPLG
28. MODESG
29. MOVECH
30. NUMBRG
31. NVECZ
32. OBJECTG
33. PACKZZ
34. PAGEG
35. POINTG
36. POLGDG
37. POLRCT
38. PUTCZZ
39. PZZ
40. RCTPOL
41. RSETSMG
42. SAMPLE
43. SCALZZ
44. SEGMTG
45. SETSMG
46. SETVPG
47. SUBJEG
48. TABSG
49. TEXTG
50. TEXTRG
51. TITLEG
52. UNSCZZ
53. VECIG
54. VECSZZ
55. VECTZZ
56. XMODZ
57. XNORMZ
58. YMODZ
59. YNORMZ

```
./  PRINT  LIST
      SUBROUTINE CONTUR                                                 CN
C CONTUR                                DATE OF OBJECT DECK 04-28-69
C       C.L.LAWSON,N.BLOCK,R.D.GARRETT   JPL    1965 APR 8              CN
C         C.LAWSON,N.BLOCK,L.SCHMELE   JPL    1966 JAN 18               CN
      COMMON/CONCOM/KFLAG1,KFLAG2,FUNCTN,NF,XMIN,XMAX,YMIN,YMAX,NDIM,   CN
     *NX,NY,XX,YY,EXCLUD                                                CN
      COMMON /BUFA/BUFA(1)                                              CN
      COMMON /BUFB/BUFB(1)                                              CN
      COMMON /BUFC/BUFC(1)                                              CN
      COMMON /BUFZ/Z(1)                                                 CN
      COMMON /AVLCOM/AVAIL                                              CN
C     KFLAG1 IS INITIALLY SET=1 BY USER. CONTUR RESETS KFLAG1=2 OR 3,   CN
C     HOWEVER USER IS PERMITTED TO RESET KFLAG1=3. GOOD LUCK.           CN
C     KFLAG1=1 BEGIN NEW CASE                                           CN
C     KFLAG1=2 GO TO NEXT TO CONTINUE PROCESSING                        CN
C     KFLAG1=3 CALL CTRIM TO TRIM LISTS, THEN GO TO NEXT TO             CN
C              CONTINUE PROCESSING                                      CN
C     KFLAG2   IS SET BY CONTUR TO BE USED IN A COMPUTED GO-TO          CN
C     IN THE USER PROGRAM.                                              CN
C     KFLAG2=1 REQUESTING FUNCTION EVALUATION                           CN
C     KFLAG2=2 PROCESSING INTERRUPTED. AVAILABLE SPACE USED UP.         CN
C              USER SHOULD PLOT AND/OR PRINT EXISTING                   CN
C              CONTOUR LISTS AND THEN RE-ENTER CONTUR                   CN
C              FOR FURTHER PROCESSING.                                  CN
C     KFLAG2=3 PROCESSING COMPLETED. USER SHOULD PLOT                   CN
C              AND/OR PRINT EXISTING CONTOUR LISTS.                     CN
C     KFLAG2=4 CATASTROPHIC SHORTAGE OF AVAILABLE SPACE.                CN
C              THE CASE CAN NOT BE COMPLETED.                           CN
      REAL X(5), Y(5), XCOOR(2), YCOOR(2),ZLO(5),ZHI(5),LAMBDA          CN
      INTEGER SAVE0,SAVE1,SAVE2,SIGN1,SIGN2,AVAIL,AVAILN                CN
      INTEGER      POINT1,POINT2,SWTCH1,SWTCH2                          CN
      INTEGER KZ1(4),KZ2(4),KX(4),KY(4),LAB(3)                          CN
      INTEGER               IBUFA(1), IBUFB(1), IBUFC(1)                CN
      INTEGER RULE(4),MODE(5)     ,POINT,POINTR, SWITCH                 CN
      EQUIVALENCE (BUFA(1),IBUFA(1)), (BUFB(1), IBUFB(1)),              CN
     *(BUFC(1), IBUFC(1))                                               CN
      EQUIVALENCE (IZ,JCOL)                                             CN
      DATA KX          /1,1,2,1/
      DATA KY          /1,1,1,2/
      DATA KZ1/2,2,3,0/                                                 CN
      DATA KZ2/0,3,1,1/                                                 CN
      DATA LABF/102 400 000/,LABT/0/                                    CN
      DATA BIG/.17E37/                                                  CN
C     ******************************************************************CN
C                                   BRANCH ON KFLAG1                    CN
      GO TO (104,102,101),KFLAG1                                        CN
C                                   TRIM LISTS TO RETRIEVE AVAILABLE SPACECN
 101  CALL CTRIM(AVAIL)                                                 CN
      IF(IBUFA(AVAIL) .GT. 0) GO TO 102                                 CN
```

```
C                                          CATASTROPHIC SPACE SHORTAGE
      KFLAG2=4
      RETURN
  102 GO TO NEXT,(115,141,18,20)
  104 CONTINUE
C     ******************************************************************
      IF (NF .NE. 0) GO TO 106
C                                          SET BRANCHES FOR NF .EQ. 0
      ASSIGN 1165 TO LINK1
      ASSIGN 145 TO LINK2
      ASSIGN 831 TO LINK4
      ASSIGN 855 TO LINK5
      IBUFB(1)=0
      IBUFC(1)=0
      GO TO 111
C                                          SET BRANCHES FOR NF .GT. 0
  106 ASSIGN 116 TO LINK1
      ASSIGN 143 TO LINK2
      ASSIGN 185 TO LINK4
      ASSIGN 845 TO LINK5
C     INITIALIZE AVAILABLE SPACE
      NF1=NF+1
      NDIMM5=NDIM-5
      DO 110 I=NF1,NDIM
      IBUFA(I)=NDIMM5-I
  110 IBUFC(I)=I+1
      IBUFC(NDIM)=0
      AVAIL = NF1
C     ******************************************************************
C     INITIALIZE CVL (CONTOUR VALUE LIST)
      DO 108 I = 1,NF
      IBUFB(I) = 0
  108 IBUFC(I) = I+1
      IBUFC(NF) = 0
C     ******************************************************************
C     INITIALIZE RULE ARRAY
C     RULE(I)=RULE TO BE USED FOR SIDE I
C     WHERE    -1=INACTIVE
C               0=SEARCH
C              +1=ACTIVE
      RULE(1)=-1
      RULE(2)=-1
      RULE(3)=+1
      RULE(4)=+1
C     ******************************************************************
C     SET UP X AND Y ARRAYS
  111 CONTINUE
      NX1 = NX - 1
      NY1 = NY - 1
      STEPX = (XMAX - XMIN)/FLOAT (NX1)
```

```
      STEPY = (YMAX - YMIN)/FLOAT (NY1)
      XCOOR(1) = XMIN
      YCOOR(1) = YMIN
      XCOOR(2) = XMIN + STEPX
      YCOOR(2) = YMIN + STEPY
C     SET UP INITIAL Z ARRAY
      NZMAX=NX1+3
      K = 1
      XX = XCOOR(1)
      YY=YCOOR(1)
      ASSIGN 115 TO NEXT
C     *****************************************************************
C                                          RETURN FOR FUNCTION EVALUATION
 112  KFLAG2 = 1
      KFLAG1=2
      RETURN
C     *****************************************************************
 115  CONTINUE
      GO TO LINK1,(116,1165)
 116  CONTINUE
      Z(K+2)=FUNCTN
 1165 CONTINUE
      XX = XX + STEPX
      K = K + 1
      IF(K .LE. NX) GO TO 112
C     *****************************************************************
C     BEGIN LOOP ON ROWS
      IROW = 1
C     RETURN HERE FROM 85+ FOR NEXT ROW OF MATRIX
 118  CONTINUE
      XX = XCOOR(1)
      YY = YCOOR(2)
      ASSIGN 141 TO NEXT
      GO TO 112
C     *****************************************************************
 141  CONTINUE
      GO TO LINK2,(143,145)
 143  CONTINUE
      Z(1) = FUNCTN
 145  CONTINUE
C     *****************************************************************
C     SET UP LOOP ON COLUMNS
      JCOL = 1
C     RETURN HERE FROM 83+ FOR NEXT COLUMN OF MATRIX
 149  CONTINUE
      XX = XCOOR(2)
      YY = YCOOR(2)
      ASSIGN 18 TO NEXT
      GO TO 112
C     *****************************************************************
```

```
 18     CONTINUE
        GO TO LINK4,(185,831)
 185    CONTINUE
        Z(IZ+1) = FUNCTN
C       *****************************************************************
C                SET ZLO( ),ZHI( ) FOR NEW SQUARE
        IEXC=0
        ZLO(5)=BIG
        ZHI(5)=-BIG
        DO 188 ISIDE=1,4
        I1=KZ1(ISIDE)+IZ
        I2=KZ2(ISIDE)+IZ
        IF(Z(I1).EQ.EXCLUD.OR.Z(I2).EQ.EXCLUD) GO TO 186
        ZLO(ISIDE)=AMIN1(Z(I1),Z(I2))
        ZHI(ISIDE)=AMAX1(Z(I1),Z(I2))
        ZLO(5)=AMIN1(ZLO(5),ZLO(ISIDE))
        ZHI(5)=AMAX1(ZHI(5),ZHI(ISIDE))
        GO TO 188
  186   ZLO(ISIDE)=EXCLUD
        ZHI(ISIDE)=EXCLUD
        IEXC=IEXC+1
  188   CONTINUE
        IF(IEXC.EQ.4) GO TO 831
C       SET UP LOOP ON CVL
  189   ICNTR=1
C       *****************************************************************
C       RETURN HERE FROM 83+ FOR NEXT CONTOUR VALUE
 19     CONTINUE
        VALUE = BUFA(ICNTR)
        IF((ZLO(5) .GT. VALUE) .OR. (VALUE .GE. ZHI(5))) GO TO 83
C       *****************************************************************
C                            TEST AVAILABLE SPACE
  195   IF(IBUFA(AVAIL).GT.0) GO TO 20
C       INSUFFICIENT AVAILABLE SPACE IN BUFFERS
        KFLAG1=3
        KFLAG2=2
        ASSIGN 20 TO NEXT
        RETURN
 20     CONTINUE
        IS=1
        DO 36 ISIDE=1,4
        IF((ZLO(ISIDE) .GT. VALUE) .OR. (VALUE .GE. ZHI(ISIDE))) GO TO 36
        IX = KX(ISIDE)
        IY = KY(ISIDE)
        I1=KZ1(ISIDE)+IZ
        I2=KZ2(ISIDE)+IZ
        DELZ = Z(I2) - Z(I1)
        FRACTN = (VALUE - Z(I1))/DELZ
        GO TO (32,33,32,33), ISIDE
 32     X(IS) = XCOOR(IX)
```

```
      Y(IS) = YCOOR(IY) + FRACTN*STEPY
      GO TO 35
 33   X(IS) = XCOOR(IX) + FRACTN*STEPX
      Y(IS) = YCOOR(IY)
 35   MODE(IS)=RULE(ISIDE)
      IS = IS + 1
 36   CONTINUE
C     ******************************************************************
C     TEST FOR NUMBER OF CROSSINGS
  361 GO TO (83,362,37,363,364),IS
C     ******************************************************************
C     HERE FOR 1 CROSSING - SHOULD ONLY HAPPEN WITH EXCLUDED MESHPOINTS
  362 MODE(2)=-1
      X(2)=X(1)
      Y(2)=Y(1)
      IS=3
      GO TO 37
C     ******************************************************************
C     HERE FOR ERRONEOUS CROSSING COUNT
 363  WRITE(6,3631)
 3631 FORMAT(2X,25HINDICATION OF 3 CROSSINGS)
C     CALL FXEM(1000)
C     ******************************************************************
C     HERE FOR 4 CROSSINGS
 364  IF(X(2) .LE. X(4)) GO TO 37
      MODE(5)=MODE(4)
      X   (5)=X   (4)
      Y   (5)=Y   (4)
      MODE(4)=MODE(2)
      X   (4)=X   (2)
      Y   (4)=Y   (2)
      MODE(2)=MODE(5)
      X   (2)=X   (5)
      Y   (2)=Y   (5)
C     ******************************************************************
C     RETURN TO 37 FROM 71+ FOR SECOND PAIR OF CROSSINGS (IF ANY)
 37   CONTINUE
      LAB(1)=LABT
      LAB(2)=LABT
C     RETURN TO 38 FROM 545- IF SEARCH FOR P1 FAILS
 38   CONTINUE
C               BRANCH ACCORDING TO MODE(1) AND MODE(2)
      KEY=3*MODE(1)+MODE(2)+5
      GO TO (60,42,60,43,44,43,60,42,60), KEY
C     ******************************************************************
C     HERE IF KEY=2,8
C     (I,S),(A,S)
C     EXCHANGE ROLES OF POINTS 1 AND2 AND USE 43
 42   MODE(5)=MODE(1)
      X(5)=X(1)
```

```
      Y(5)=Y(1)
      LAB(3)=LAB(1)
      MODE(1)=MODE(2)
      X(1)=X(2)
      Y(1)=Y(2)
      LAB(1)=LAB(2)
      MODE(2)=MODE(5)
      X(2)=X(5)
      Y(2)=Y(5)
      LAB(2)=LAB(3)
      GO TO 43
C     ******************************************************************
C     HERE IF KEY=4,6
C     (S,I),(S,A)
C     SEARCH ON P1 AND ATTACH P2
 43   POINT=0
      ASSIGN 435 TO KRTN
      GO TO 49
 435  CONTINUE
C                                    ATTACH P2 TO EXISTING ARC
      BUFA(AVAIL)=X(2)
      BUFB(AVAIL)=Y(2)
      AVAILN = IBUFC(AVAIL)
C     IF SWITCH=0--ATTACH NEW POINT TO HEAD OF ARC LIST
C     IF SWITCH=1--ATTACH NEW POINT TO TAIL OF ARC LIST
      IF (SWITCH .NE. 0) GO TO 437
C     HERE TO ATTACH TO HEAD LIST
      IBUFC(AVAIL) = PCINTR
      IBUFA(IXADL)=ISIGN(AVAIL,MODE(2))-LAB(2)
      GO TO 439
C     HERE TO ATTACH TO TAIL LIST
 437  IBUFC(AVAIL)=0
      IBUFC(POINTR) = AVAIL
      IBUFB(IXADL)=ISIGN(AVAIL,MODE(2))-LAB(2)
 439  AVAIL=AVAILN
      GO TO 70
C     ******************************************************************
C     HERE IF KEY=5
C     (S,S)
C     SEARCH FOR P1 AND P2 -- JOIN THE ARCS
 44   POINT=0
      ASSIGN 445 TO KRTN
      GO TO 49
 445  CONTINUE
      POINT1=POINTR
      SWTCH1=SWITCH
      IXADL1=IXADL
      POINT=1
      ASSIGN 475 TO KRTN
      GO TO 49
```

```
 475   CONTINUE
       POINT2 = POINTR
       SWTCH2 = SWITCH
       IXADL2 = IXADL
       IF(IXADL1.NE.IXADL2) GO TO 4753
C                                  COMPLETE A CLOSED CURVE
C                                  SET FLAGS TO LABEL HEAD BUT NOT TAIL
       IHEAD=IBUFA(IXADL1)
       IBUFA(IXADL1)=-IHEAD
       ITAIL=IBUFB(IXADL1)
       IBUFC(ITAIL)=AVAIL
       AVAILN=IBUFC(AVAIL)
       IBUFB(IXADL1)=-AVAIL-LABF
       BUFA(AVAIL)=BUFA(IHEAD)
       BUFB(AVAIL )=BUFB(IHEAD)
       IBUFC(AVAIL)=0
       AVAIL=AVAILN
       GO  TO 70
 4753  IF(SWTCH1 .EQ. SWTCH2) GO TO 477
       IF(SWTCH1 .NE. 0) GO TO 476
C      HERE IF P1 IS HEAD, P2 IS TAIL
       IBUFC(POINT2) = POINT1
       IBUFB(IXADL2) = IBUFB(IXADL1)
       IREMOV = IXADL1
       GO TO 65
C      HERE IF P1 IS TAIL, P2 IS HEAD
 476   IBUFC(POINT1) = POINT2
       IBUFB(IXADL1) = IBUFB(IXADL2)
       IREMOV = IXADL2
       GO TO 65
C                     HERE MUST CONNECT ARCS HEAD TO HEAD OR TAIL TO TAIL
 477   IF(SWTCH1 .EQ. 0) GO TO 478
C      HERE IF P1 AND P2 ARE TAILS
       IBUFB(IXADL2) = IBUFA(IXADL1)
       IBUFC(POINT2) = POINT1
C                                  UNPACK IBUFA(IXADL1)
       SAVE0=IABS(IBUFA(IXADL1))
       SAVE1=SAVE0-LABF
       IF(SAVE1 .GT. 0) SAVE0=SAVE1
       SIGN1=IBUFC(SAVE0)
       SAVE1=IABS(SIGN1)
       IBUFC(SAVE0)=0
       GO TO 479
C      HERE IF P1 AND P2 ARE HEADS
 478   IBUFA(IXADL2) = IBUFB(IXADL1)
       SAVE0=POINT1
       SIGN1 = IBUFC(SAVE0)
       SAVE1=IABS(SIGN1)
       IBUFC(SAVE0)=POINT2
C               REVERSE THE NO. 1 LIST
```

```
 479   SIGN2=IBUFC(SAVE1)
       SAVE2=IABS(SIGN2)
       IBUFC(SAVE1)=ISIGN(SAVE0,SIGN1)
       IF(SAVE2 .EQ. 0) GO TO 4791
       SAVE0=SAVE1
       SAVE1=SAVE2
       SIGN1=SIGN2
       GO TO 479
 4791  IREMOV = IXADL1
       GO TO 65
C      ******************************************************************
C      SEARCH FOR POINT ON APL (ARC POINT LIST)
 49    IXADL = IBUFB(ICNTR)
       XNEW=X(POINT+1)
       YNEW=Y(POINT+1)
 50    IF(IXADL .EQ. 0) GO TO 54
       IHEAD = IBUFA(IXADL)
       ITAIL = IBUFB(IXADL)
       POINTR = IHEAD
       SWITCH = 0
       GO TO 52
 51    POINTR = ITAIL
       SWITCH = 1
C      EXAMINE ONLY ACTIVE ENDS OF ARCS
 52    IF(POINTR .LE. 0) GO TO 53
       XOLD = BUFA(POINTR)
       YOLD = BUFB(POINTR)
C      COMPARE POINTS (XNEW, YNEW,)AND      (XOLD, YOLD)
       IF((XNEW .NE. XOLD) .OR.  (YNEW .NE. YOLD)) GO TO 53
       IF(POINT.EQ.0) GO TO 525
       IF(POINT1.EQ.POINTR) GO TO 53
 525   KFOUND = 1
       GO TO 55
 53    IF(SWITCH .EQ. 0) GO TO 51
       IXADL=IBUFC(IXADL)
       GO TO 50
 54    KFOUND = 0
C      ******************************************************************
C      SEARCH FAILED
       IF(POINT.NE.0) GO TO 545
       MODE(1)=-1
       LAB(1)=LABF
       GO TO 38
 545   MODE(2)=-1
       LAB(2)=LABF
       KEY=4
       SWITCH=SWTCH1
       POINTR=POINT1
       IXADL=IXADL1
       GO TO 435
```

```
 55     CONTINUE
        GO TO KRTN, (435,445,475)
C       ***********************************************************************
C       HERE IF KEY=1,3,7,9
C       (I,I),(I,A),(A,I),(A,A)
C       HERE TO START NEW ARC
C                FIRST OBTAIN 3 TRIPLE WORDS FROM AVAILABLE LIST
 60     IXADL=AVAIL
        IHEAD=IBUFC(IXADL)
        ITAIL=IBUFC(IHEAD)
        AVAIL=IBUFC(ITAIL)
C                THEN CONSTRUCT NEW ARC
        BUFA(IHEAD) = X(1)
        BUFB(IHEAD) = Y(1)
        IBUFC(IHEAD) = ITAIL
        BUFA(ITAIL) = X(2)
        BUFB(ITAIL) = Y(2)
        IBUFC(ITAIL) = 0
        IBUFA(IXADL)=ISIGN(IHEAD,MODE(1))-LAB(1)
        IBUFB(IXADL)=ISIGN(ITAIL,MODE(2))-LAB(2)
        IBUFC(IXADL) = IBUFB(ICNTR)
        IBUFB(ICNTR) = IXADL
        GO TO 70
C       ***********************************************************************
C       HERE TO RESTORE A WORD TO AVAILABLE SPACE
 65     CONTINUE
C       FIRST, EXAMINE CVL LIST
        IXADL=IBUFB(ICNTR)
        IF(IXADL.NE.IREMOV) GO TO 653
        IBUFB(ICNTR)=IBUFC(IREMOV)
        GO TO 655
 652    IXADL=IBUFC(IXADL)
 653    IF(IBUFC(IXADL).NE.IREMOV) GO TO 652
 654    IBUFC(IXADL)=IBUFC(IREMOV)
 655    IBUFC(IREMOV)=AVAIL
        IBUFA(IREMOV)=IBUFA(AVAIL)+1
        AVAIL = IREMOV
C       ***********************************************************************
 70     CONTINUE
        IF(IS.LT.5) GO TO 83
        DO 71 I = 1,2
        MODE(I)=MODE(I+2)
        X(I) = X(I+2)
 71     Y(I) = Y(I+2)
        IS=3
        GO TO 37
C       ***********************************************************************
C       HERE AFTER PROCESSING ONE SQUARE FOR A SINGLE CONTOUR VALUE
 83     CONTINUE
        ICNTR = ICNTR + 1
```

```
      IF(ICNTR .LE. NF) GO TO 19
 831  CONTINUE
      XCOOR(1) = XCOOR(2)
      XCOOR(2) = XCOOR(2) + STEPX
      JCOL = JCOL + 1
      RULE(1)=0
      IF(JCOL-NX1) 149,833,84
 833  RULE(3)=-1
      GO TO 149
C     ***************************************************************
C     HERE AFTER ALL COLUMNS OF MATRIX HAVE BEEN PROCESSED
C     MOVE Z-VALUES DOWN 2 POSITIONS AND INCREMENT YCOOR(1),
C     YCOOR(2), FOR NEXT ROW
 84   CONTINUE
      GO TO LINK5,(845,855)
 845  CONTINUE
      DO 85 K = 3, NZMAX
      KBACK=3+NZMAX-K
 85   Z(KBACK)=Z(KBACK-2)
 855  CONTINUE
      YCOOR(1) = YCOOR(2)
      YCOOR(2) = YCOOR(2) + STEPY
      XCOOR(1)=XMIN
      XCOOR(2)=XMIN+STEPX
      IROW = IROW + 1
      RULE(1)=-1
      RULE(2)=0
      RULE(3)=+1
      IF(IROW-NY1) 118,853,86
 853  RULE(4)=-1
      GO TO 118
C     ***************************************************************
C     HERE WHEN FINISHED WITH ALL CONTOURS FOR ENTIRE MATRIX
 86   CONTINUE
      KFLAG2 = 3
      KFLAG1 = 1
      RETURN
      END
      SUBROUTINE CSCAN
C CSCAN                               DATE OF OBJECT DECK 04-28-69
C       C.L.LAWSON,N.BLOCK,R.D.GARRETT   JPL    1965 APR 5
      COMMON /BUFA/BUFA(1)
      COMMON /BUFB/BUFB(1)
      COMMON /BUFC/BUFC(1)
      COMMON /SCOM/KSTART,KGO,HLABEL,TLABEL,IZ,Z,IARC,X,Y
C     KSTART IS SET=1 BY USER TO INITIATE NEW CASE.
C     KGO IS SET BY CSCAN
C     KGO = 1  NEW CONTOUR VALUE,(AND NEW ARC,POINT,AND LABEL FLAGS)
C           2  NEW ARC,(AND NEW POINT AND LABEL FLAGS)
C           3 NEW SUB-ARC AFTER GAP,(AND NEW POINT)
```

```
C           4  NEW POINT
C           5  FINISHED,(NO NEW POINT)
      LOGICAL HLABEL,TLABEL
      INTEGER IBUFA(1),IBUFB(1),IBUFC(1)
      INTEGER KSTART,KGO,IZ,IARC
      INTEGER JCVL,JADL,JAPL
      REAL X,Y,Z
      REAL BUFA,BUFB,BUFC
      DATA LABF/102 400 000/
      EQUIVALENCE(IBUFA,BUFA),(IBUFB,BUFB),(IBUFC,BUFC)
      GO TO (11,15),KSTART
 11   KSTART=2
      JCVL=1
      IZ=1
 12   JADL=IBUFB(JCVL)
      IF(JADL.EQ.0)GO TO 16
      IARC=1
      KGO=1
      Z=BUFA(JCVL)
C                                   HERE AT BEGINNING OF NEW ARC
 13   JAPL=IABS(IBUFA(JADL))
      JTRY=JAPL-LABF
      IF(JTRY .GT. 0) JAPL=JTRY
      HLABEL=(IBUFA(JADL).LT.0).AND.(IBUFA(JADL).GT.(-LABF))
      TLABEL=(IBUFB(JADL).LT.0).AND.(IBUFB(JADL).GT.(-LABF))
 148  X=BUFA(JAPL)
      Y=BUFB(JAPL)
      RETURN
 15   JAPL=IBUFC(JAPL)
      IF(JAPL)151,155,152
C                                   CROSSING A GAP
 151  JAPL=-JAPL
      KGO=3
      GO TO 148
C                                   ORDINARY NEW POINT
 152  KGO=4
      IF(X.EQ.BUFA(JAPL).AND.Y.EQ.BUFB(JAPL)) GO TO 15
      GO TO 148
 155  KGO=2
      JADL=IBUFC(JADL)
      IARC=IARC+1
      IF(JADL.GT.0)GO TO 13
 16   JCVL=IBUFC(JCVL)
      IZ=IZ+1
      IF(JCVL.NE.0) GO TO 12
      KGO=5
      RETURN
      END
      SUBROUTINE CTRIM(AVAIL1)
C      CTRIM       TRIM CONTUR LISTS RESTORING AVAILABLE SPACE
```

```
C CTRIM                                         DATE OF OBJECT DECK 04-28-69
C        C.L.LAWSON,N.BLOCK,R.D.GARRETT    JPL    1965 APR 5
      INTEGER  AVAIL1
      INTEGER  LABF,AVAIL,ICVL,IADL,IADLX,IHEAD,ITAIL,IADL1
      INTEGER           JUNK,JUNK1
      INTEGER      A,      B,      C
      LOGICAL DBLPLS
      COMMON   /BUFA/A(1)
      COMMON   /BUFB/B(1)
      COMMON   /BUFC/C(1)
      DATA     LABF/102400000/
C     ******************************************************************
      AVAIL=AVAIL1
      ICVL=1
 11   IADL=B(ICVL)
      IADLX=0
C                    RETURN HERE FOR EACH NEW ARC
 12   IF(IADL.NE.0)GO TO 15
      ICVL=C(ICVL)
      IF(ICVL.NE.0)GO TO 11
      IF(A(AVAIL) .LE. 0) WRITE(6,1000)
      AVAIL1=AVAIL
C     ******************************************************************
      RETURN
C     ******************************************************************
 15   IHEAD=A(IADL)
      ITAIL=B(IADL)
      IADL1=C(IADL)
      DBLPLS=(IHEAD .GT. 0) .AND. (ITAIL .GT. 0)
      IF(DBLPLS) GO TO 16
C     ******************************************************************
C     HERE IF EITHER HEAD OR TAIL OF LIST IS INACTIVE
C     ENTIRE ARC LIST MAY BE ELIMINATED
      IF(IHEAD.LT.(-LABF)) IHEAD=IHEAD+LABF
      C(IADL)=IABS(IHEAD)
      JUNK=IADL
      IF(IADLX.NE.0)GO TO 17
C     ******************************************************************
C     HERE IF ARC IS FIRST ARC FOR CURRENT CONTOUR VALUE
      B(ICVL)=IADL1
      GO TO 18
C     ******************************************************************
C     ******************************************************************
C     HERE IF ARC IS NOT FIRST ARC FOR CURRENT CONTOUR VALUE
 17   C(IADLX)=IADL1
      GO TO 18
C     ******************************************************************
C     HERE IF BOTH HEAD AND TAIL ARE ACTIVE (DBLPLS=.TRUE.)
C     END-POINTS OF LIST MUST BE PRESERVED
 16   JUNK=IABS(C(IHEAD))
```

```
      C(IHEAD)=-ITAIL
      IADLX=IADL
C     SAVE ADL POINTER FOR LATER USE AT STATEMENT 12
 18   IADL=IADL1
C     *****************************************************************
C     HERE TO RETURN WORDS TO AVAILABLE STORAGE
 19   JUNK1=IABS(C(JUNK))
      IF((JUNK1.EQ.0).AND. DBLPLS )GO TO 12
C     *****************************************************************
      A(JUNK)=A(AVAIL)+1
      C(JUNK)=AVAIL
      AVAIL=JUNK
C     WHEN LAST WORD ON CURRENT ARC LIST HAS BEEN PROCESSED,START WITH
C     NEXT ARC
      IF(JUNK1.EQ.0)GO TO 12
C     OTHERWISE,PROCEED TO NEXT POINT ON CURRENT ARC LIST
      JUNK=JUNK1
      GO TO 19
C     *****************************************************************
 1000 FORMAT( 41H0AVAILABLE SPACE INSUFFICIENT FOR FURTHER
     * , 30H CONSTRUCTION OF CONTOUR LISTS/1X)
      END
```

IDENTIFICATION

ERDUMP/Subroutine causing an abend with any program interrupt*

May, 1971

DESCRIPTION

ERDUMP causes a program to terminate abnormally whenever program interrupts, such as underflow, overflow, divide check, etc. occurs. If a core dump is needed at the time of these interrupts, it is sufficient to call ERDUMP and to add a SYSUDUMP DD card.

USAGE

At the beginning of the program, insert the following card:

CALL ERDUMP

In addition include the following DD card before any data, observing the rules for user-provided DD cards:

//SYSUDUMP DD SYSOUT=A for FORTG procedure

NOTE:

The dump routine takes approximately 6 K bytes of core. Care must be taken that the REGION on the EXEC card parameter is big enough.

If a dump is required for a particular program interrupt, other than the first one that occurs, then the "Extended Error Message Facility" described in the FORTRAN IV Programmer's Guide must be used.

---

* Willis H. Booth Computing Center Report No. C1169-320-360

```
./  PRINT  LIST
        ENTRY ERDUMP
ERDUMP  STM   14,12,12(13)
        BALR  12,0
        USING *,12
        SPIE
        L     1,=X'02000000'
        SPM   1
        L     14,12(0,13)
        LM    2,12,28(13)
        BR    14
*
        END
```

IDENTIFICATION

INCORE/Input from core storage - ASSEMBLER coded*

John Hughes

October, 1971

PURPOSE

To read literal data from core storage with a FORMAT.

METHOD

A. To disconnect the reader use:

CALL INCORE(FWA,N)

where

FWA is the name of a dimensioned array containing the literal data to be read.

N is an INTEGER*4 variable or constant equal to the number of BYTES or characters in the array FWA.

After the reader is disconnected, use a READ on unit 5 with a FORMAT as usual. This READ will not read a card but will read the literal data in array FWA instead of the usual card image. The FORMAT must specify only one record; that is, it must contain no slashes. Also, there must be at least as many format codes as there are variables to be read. No END= or ERR= may be used.

---

* Willis H. Booth Computing Center Report No. C1068-266-370

B. To connect the reader again use the statement:

CALL INCORE

This call must be issued right after the READ statement.

USER ABEND CODES

2001 - invalid INCORE CALL

2002 - not formatted READ from unit 5 or a multiple record usage

2003 - improper FIOCS module loaded.

STORAGE

INCORE = $(368)_{10}$ bytes

Example of usage of INCORE:

```
      DIMENSION A(9), E(6)
      DATA A/' B =', ' 3.1', '4159', 'C ='.
  1   ' 6.2', '831 ', 'D = ', '12.5', '462 '/
      NBYTES=36
      CALL INCORE (A,NBYTES)
  5   READ (5,500) E
      CALL INCORE
 10   WRITE (6,500) E
500   FORMAT (IX,A4,F7.5,IX,A4,F6.4,IX,A4,F7.4)
      STOP
      END
```

NOTES:

Statement 5 does not read from unit 5, but stores into E the following values with corresponding format:

E(1)	B =	A4
E(2)	3.14159	F7.5
E(3)	C =	A4
E(4)	6.2831	F6.4
E(5)	D =	A4
E(6)	12.5462	F7.4

OUTPUT created by statement 10:

B = 3.14159 C = 6.2831 D = 12.5462

INCORE is available only on the 370/155 FORTRAN library.

```
./  PRINT  LIST
ICOR      TITLE 'INCORE.  FORTRAN CORE <--> CORE READ.'
* INCORE                                        DATE OF OBJECT DECK 03-18-72
          SPACE 2
***   INTERFACE AND ASSEMBLY PARAMETERS.
*
*         CALL   INCORE(A,N)
*         READ (5,F) LIST
*         CALL   INCORE
*
*WILL CAUSE THE EBCDIC INPUT OF THE READ TO BE TAKEN FROM CORE
*STARTING AT LOCATION A.  N IS THE NUMBER OF CHARS IN THE ARRAY 'A'.
*
****************
*
INCORE    START X'10000'
*
*
INUNIT    EQU    5                          NUMBER OF INPUT UNIT
FOMASK    EQU    X'F0'                      CODE FOR FORMATTED INPUT
INITOPT   EQU    0                          OPCODE FOR INITIAL ENTRY.
*
*
XBASE     EQU    12                         PROGRAM BASE REGISTER.
XFIO      EQU    11                         PTR TO FIOCS.
XOPT      EQU    10                         PTR TO OPCODE AND OPTION BYTES.
*
*
XBLEN     EQU    3                          RETURN REG FOR BUFFER LENGTH.
XBUF      EQU    2                          RETURN REG FOR BUFFER ADDRESS.
XDSRN     EQU    2                          INPUT REG FOR UNIT NUMBER PTR.
*
*
          EJECT
          SPACE 2
***   SET/RESET FIOCS FOR CORE-CORE READ.
*
*         CALL INCORE<(A,M)>
*
*
INCORE    CSECT
          SAVE  (14,12),,*                  SAVE USER'S GPR'S.
          LR    XBASE,15                    SET PROGRAM BASE.
          USING INCORE,XBASE          #XBASE
          LA    15,SVA                      SET NEW SAVE AREA.
          ST    13,4(,15)
          ST    15,8(,13)
          LR    13,15
          L     XFIO,=V(FIOCS#)             (XFIO)= PTR TO FIOCS MODULE.
          XI    SETSW,X'FF'                 INVERT INCORE STATUS.
```

```
         BNZ   SETFIOCS                BRANCH IF ACTIVATING CALL.
         MVC   0(PLUGSIZE,XFIO),UNPLUG RESTORE FIOCS.
RETURN   L     13,4(,13)               RESTORE OLD SAVE AREA PTR.
         RETURN (14,12)                RETURN TO CALLER.
SETFIOCS LA    0,3                     GET SET TO CHECK (1).
         NR    0,1                     IS (1) A VALID PARAM LIST PTR ?
         BNZ   ERROR1                  BRANCH IF NOT.
         TM    4(1),X'80'              HAVE WE GOT 2 PARAMETERS ?
         BNO   ERROR1                  BRANCH IF NOT.
         TM    7(1),X'03'              IS 2ND PARAM A FULL-WORD ?
         BNZ   ERROR1                  BRANCH IF NOT.
         MVC   PLIST(8),0(1)           MOVE PTR'S TO AREA & WORD COUNT.
         MVC   UNPLUG(PLUGSIZE),0(XFIO) SAVE OVERLAYED FIOCS CODE.
*        CLC   UNPLUG(PLUGSIZE),TEMPLATE
*        BE    PROCEED
         B     PROCEED
         ABEND 2003,DUMP,STEP
PROCEED  MVC   0(PLUGSIZE,XFIO),PLUG   OVERLAY FIOCS WITH LINKAGE TO   *
                                       THE CORE-CORE READER.
         B     RETURN                  RETURN TO CALLER.
ERROR1   ABEND 2001,DUMP,STEP          INVALID INCORE CALL
         DROP  XBASE              #XBASE
         EJECT
         SPACE 2
***   CORE <--> CORE READER.
*
*        (0) = RETURN ADDR - 2
*            = PTR TO AL1(OPCODE,OPTION)
*        (1) = A(INCOREX)
*        (2) = PTR TO UNIT NUMBER (OPCODE = INIT)
*            = PTR TO BUFFER ADDRESS (OPCODE = READ)
*        (3) = LENGTH OF BUFFER
*
*
INCOREX  DS    0H
         USING INCOREX,1          #1
         STM   0,15,SVA                SAVE USER'S GPR'S.
         DROP  1                  #1
         LR    XBASE,1                 SET PROGRAM BASE.
         USING INCOREX,XBASE      #XBASE
         LR    XOPT,0                  (XOPT)= A(OPTION BYTES).
         CLI   0(XOPT),INITOPT         INITIAL ENTRY ?
         BE    INIT                    BRANCH IF SO.
         B     ERROR2                  BRANCH IF NOT.
RETX     L     XBUF,PLIST              BUFFER ADDRESS
         L     XBLEN,PLIST+4           A(LENGTH)
         L     XBLEN,0(0,XBLEN)        LENGTH
         L     0,SVA                   RESTORE GPR 0.
         LR    1,0                     (1)= RETURN ADDRESS - 2.
         LM    4,15,SVA+16             RESTORE REMAINING GPR'S.
```

```
         B      6(,1)                              RETURN TO CALLER. (IBCOM)
*
*
INIT     L      1,0(,XDSRN)                        (1)= UNIT NUMBER OR PTR.
         TM     0(XDSRN),X'01'                     (1)= PTR ?
         BZ     *+8                                SKIP IF NOT.
         L      1,0(,1)                            (1)= UNIT NUMBER.
         CH     1,=Y(INUNIT)                       IS THIS FOR INPUT  UNIT ?
         BNE    ERROR2                             BRANCH IF NOT.
         CLI    1(XOPT),FOMASK                     FORMATTED INPUT ?
         BE     RETX                               RETURN IF SO.
ERROR2   ABEND 2002,DUMP,STEP                      NOT FORMATTED INPUT FROM UNIT 5
*                                                  OR MULTIPLE RECORD USAGE
         EJECT
         SPACE 2
***  CONSTANTS AND WORK AREAS.
*
*
PLIST    DC     2A(0)                              AREA, WORD COUNT PTR.
PLUG     DS     0F                                 OVERLAY FOR FIOCS.
         L      1,8(,1)                                 (1)= A(CORE-CORE READER).
         BR     1                                       LINK TO CORE-CORE READER.
SETSW    DC     AL2(0)                                  INCORE STATUS SWITCH.
         DC     A(INCOREX)
PLUGSIZE EQU    *-PLUG                             SIZE OF FIOCS OVERLAY.
UNPLUG   DS     CL(PLUGSIZE)                       OVERLAYED FIOCS CODE.
TEMPLATE DC     X'071047F01022071058101F9A'
SVA      DS     18F                                REGISTER SAVE AREA.
         LTORG                                     LITERAL POOL.
         SPACE 2
         END  INCORE
```

IDENTIFICATION

LABEL/VLABEL/Calcomp plotter routine to label and draw axis with linear scale.*

James Lo

Program date (latest version) - April 13, 1973

Write-up date - May, 1973

METHOD

This routine draws a horizontal or vertical axis with tick marks, labels at tick marks, and title.

USAGE

Calling sequence:

```
      CALL LABEL(X,Y,XMN,XMX,SIZE,NI,TITLE,NCT,
     *   ND)
```

where:

X, Y = starting point of axis, in inches

XMN, XMX = range of axis (first and last values, printed at the beginning and end of axis)

SIZE = length of axis, in inches

NI = number of intervals along axis

If this value is too large to allow printed label, or is negative, only tick marks will be made.

TITLE = Hollerith title to be printed along axis

NCT = number of characters including blanks, of TITLE

---

* Willis H. Booth Computing Center Report No. C167-210-370

(NCT > 0) - Title will be printed below the X-axis or to the left of the Y-axis

(NCT < 0) - Title will be printed above the X-axis or to the right of the Y-axis

(NCT = 0) - No title will be printed

ND = 0, axis will be drawn horizontally

1, axis will be drawn vertically

or:

```
CALL VLABEL (X, Y, XMN, XMX, SIZE, NI, TITLE, NCT, ND,
*       FMT, LF)
```

Normally, the magnitude of the numeric labels along the axis drawn is scaled to a value which lies between 0 and 9.999 and then the format F6.3 is used to plot the label. To some users, neither the scaling nor the format adopted may prove desirable. Two more arguments are therefore added to provide more flexibility in labeling:

FMT = a one-dimensional EBCDIC array where the format is stored. The format will be used in labeling. FMT is of the following form: (F n. m) FMT <u>must</u> be dimensioned if there are more than 4 characters, counting parentheses. Note: Be sure to include parentheses.

where:

n - is the field length, which must include a position for the sign and a position for the decimal point (m may be zero).

m - is the number of decimal places after the decimal point (m may be zero).

LF = an integer specifying the field length of the format provided (LF = n). However, if the user wishes to print out an integer value, without the decimal point, LF should be set equal to n-1, and the format should be of the form (Fn.0).

NOTE:

When FMT and LF are provided in the calling sequence the numeric labels will not be scaled, therefore the user must provide a format with sufficient field length.

If the space between intervals is not large enough for the printing of the values, there may be overlapping of the first and second prints.

To avoid this, the following formula must be true:

$$\text{Space} > (n + I) * 0.12$$

where:

Space = value of the interval in inches

n = field length in the format statement (i.e., Fn.m)

I = number of digits to the left of the decimal point in the second value to be printed.

NOTE:

Letter sizes have been set as follows:

Numeric label along axis	=	0.12
Title	=	0.16
Scale factor	=	0.10

If the user wishes to alter any or all of the sizes, he may make use of the labeled common statement.

COMMON/LBLCOM/ITEST, SLBL, STTL, SSCL

where:

ITEST must be set = 1

SLBL = size of numeric label

STTL = size of title

SSCL = size of scale

These are real values in inches and must all be defined even if only one or two values are to be altered. User must be cautioned not to exceed boundaries of paper.

DD Card

Users calling any plot subroutines must provide the appropriate DD Card as follows:

```
//SYSPLTDN      DD      SYSOUT=N     (for narrow paper plotting)
     or
//SYSPLTDW      DD      SYSOUT=W     (for wide paper plotting)
```

The GO. step should be included if the standard FORTGCLG procedure is used.

H Compilation

Length = LABEL (5488 bytes), CONST (1208 bytes)

Timing = 100 ms for 15'' labeled axis

NOTE:

LABEL uses OUTCOR, SYSSYM, SYSPLT, CONST, and Character Manipulation programs: INDEX and DELETS

For general plotting information, see "CIT User's Guide to Calcomp Plotter."

A call to LABEL should precede calls to XYPLT, XYPLOT, CPLT, CPLOT, PLOTXY, or PLOTY because the latter six routines have 'LAB' as an argument, which is an option to terminate the plotting page. Otherwise, a call to SYSEND is required.

This program was revised in October, 1972 to create improved spacing for the printing along the axis, using CONST and the Character Manipulation programs.

NOTE: The name INDEX must not be used for a subprogram when LABEL is used.

```
./  PRINT  LIST
./  PRINT  LIST
./  PRINT  LIST
      SUBROUTINE LABEL(X,Y,XMN,XMX,FL   ,NV,TITLE,NCT,ND)                  02
C   LABEL REVISED 04-13-73
       INTEGER DELETS
       LOGICAL*1 NUMBER(10)
        LOGICAL*1 IFMT(26)
       LOGICAL IFMTL,NUML
       EQUIVALENCE(IFMTL,IFMTT),(NUML,NUMM)
       DATA NUMBER/'1','2','3','4','5','6','7','8','9','0' /
C     THE SUBROUTINE 'LABEL' PRINTS (1) TICK MARKS, (2) LABEL AT EACH TILA
C     MARK AND (3) TITLE, ALONG THE AXIS IN EITHER X OR Y DIRECTION.    LA
C     THE ARGUMENTS ARE,                                                LA
C     X,Y--STARTING POINT OF THE AXIS, IN INCHES.                       LA
C     XMN,XMX--RANGE OF THE AXIS.                                       LA
C     FL  --LENGTH OF THE AXIS, IN INCHES.                              LA
C     NV--NO. OF INTERVALS ALONG THE AXIS.                              LA
C     TITLE--BCD TITLE TO BE PRINTED ALONG THE AXIS.                    LA
C     NCT--NO. OF CHARACTERS, INCLUDING BLANKS, OF THE TITLE.           LA
C     (1) NCT = POSITIVE, TITLE WILL BE PRINTED BELOW THE X-AXIS OR TO  LA
C               THE LEFT OF Y-AXIS.                                     LA
C     (2) NCT = NEGATIVE, TITLE WILL BE PRINTED ABOVE THE X-AXIS OR TO TLA
C               RIGHT OF Y-AXIS.                                        LA
C     (3) NCT = 0, NO TITLE WILL BE PRINTED.                            LA
C     ND--DIRECTION OF THE AXIS.                                        LA
C     FMT--OPTIONAL F FORMAT PROVIDED BY THE USER.                      LA
C     NCT--FIELD LENGTH OF THE F FORMAT.                                LA
C         ND = 0, THE AXIS IS TO BE DRAWN HORIZONTALLY.                 LA
C         ND = 1, THE AXIS IS TO BE DRAWN VERTICALLY.                   LA
C      LETTER SIZES (HEIGHTS) HAVE BEEN SET AS FOLLOWS -                LA
C             NUMERIC LABEL ALONG AXIS = 0.12                           LA
C             TITLE                    = 0.16                           LA
C             SCALE FACTOR             = 0.10                           LA
C      IF USER WISHES TO ALTER ANY OR ALL OF THE SIZES, HE SHOULD       LA
C      USE A LABELED COMMON STATEMENT -                                 LA
C      COMMON / LBLCOM / ITEST, SLBL, STTL, SSCL                        LA
C      WHERE ITEST MUST BE SET TO 1 AND                                 LA
C            SLBL = SIZE OF NUMERIC LABEL                               LA
C            STTL = SIZE OF TITLE                                       LA
C            SSCL = SIZE OF SCALE                                       LA
C      SLBL, STTL, SSCL ARE ALL REAL NUMBERS IN INCHES AND MUST ALL BE  LA
C      DEFINED EVEN IF ONLY ONE OR TWO VALUES ARE TO BE ALTERED.        LA
C                                                                       LA
C                                                                       LA
      DIMENSION TITLE(1)                                                LA
      DIMENSION SCL(5),XL(100),BCDW(22),FMT(6)
       DIMENSION FMX(1)
      COMMON /LBLCOM/ ITEST,SLBL,STTL,SSCL                              LA
      DATA SCL/'SCAL','E FA','CTOR',' = 1','0'/                         LA
```

```
      EQUIVALENCE(IFMT(1),FMT(1))
       ICHKNV = IABS(INT(FL)) * 40                                      LA
       IF (IABS(NV) .GT. ICHKNV) GO TO 50                               LA
      SGLBL = 0                                                         LA
      SGFMT = 0                                                         LA
      GO TO 2                                                           LA
C     CALL NARGS(NAG)                                                   LA
C     IF(NAG .EQ. 9) GO TO 2                                            LA
      ENTRY VLABEL(X,Y,XMN,XMX,FL,NV,TITLE,NCT,ND,FMX,NCF)
       ICHKNV = IABS(INT(FL)) * 40                                      LA
       IF (IABS(NV) .GT. ICHKNV) GO TO 51                               LA
       DO  403 I=1,6
  403  FMT(I)=FMX(I)
      SGLBL=0                                                           LA
      SGFMT = 1.0                                                       LA
      NFMT = NCF                                                        LA
       M3=INDEX(IFMT,')',26,1)
        IF (M3.EQ.0) GO TO 52
C      WRITE(6,604)(IFMT(I),I=1,M3)
  604  FORMAT(1X10A1)
       N1=DELETS(IFMT,' ',26,1)
C      WRITE(6,604)(IFMT(I),I=1,M3)
       M3=INDEX(IFMT,')',26,1)
       M1=INDEX(IFMT,'F',M3,1)
       M2=INDEX(IFMT,'.',M3,1)
       M1=M1+1
       M4=M2-1
       N=0
C      WRITE(6,605) M3,M1,M2
  605  FORMAT(1X3I5)
       DO 410 I=M1,M4
       IFMTL=IFMT(I)
        DO 400 K=1,10
       NUML=NUMBER(K)
       IF (IFMTT.EQ.NUMM) GO TO 405
  400 CONTINUE
        GO TO 410
  405 N1=K
       IF (K.EQ.10) N1=0
       IF (N.NE.0) GO TO 407
      NCOL=N1
       GO TO 409
  407 NCOL=NCOL*10+N1
  409  N=N+1
  410 CONTINUE
C       WRITE(6,606) NCOL
  606  FORMAT(1X'NCOL=',I5)
       M2=M2+1
       N=0
       DO 450 I=M2,M3
```

```
      IFMTL=IFMT(I)
      DO 420 K=1,10
      NUML=NUMBER(K)
      IF (IFMTT.EQ.NUMM) GO TO 425
  420 CONTINUE
      GO TO 450
  425 N1=K
      IF (K.EQ.10) N1=0
      IF (N.NE.0) GO TO 427
      NDIGT=N1
      GO TO 429
  427 NDIGT=NDIGT*10+N1
  429  N=N+1
  450 CONTINUE
C      WRITE(6,607) NDIGT
  607  FORMAT(1X'NDIGT=',I5)
      GO TO 4
 2     NFMT=7
    4 IF(ITEST .EQ. 1) GO TO 6
      SLBL = 0.12
      STTL = 0.16
      SSCL = 0.1
    6 M = 0
C     DRAW THE AXIS.
      CALL SYSPLT(X,Y,3)
      IF(ND .NE. 0) GO TO 8
      CALL SYSPLT(X+FL,Y,2)
      GO TO 10
    8 CALL SYSPLT(X,Y+FL,2)
   10   NI = IABS(NV)
      FNI = NI
      DX = (XMX-XMN)/FNI
C     DX1 = DX IN INCHES.
      DX1 = FL/FNI
      NI1 = NI + 1
       IF (NV .GT. 0) GO TO 11
       SGLBL = 1.0
       GO TO 21
   11 DO 12 I = 1,NI1
   12 XL(I) = XMN + FLOAT(I-1)*DX
      AX = AMAX1(ABS(XMX),ABS(XMN))
      LGX = ALOG10(AX)
C       WRITE(6,600) LGX
  600  FORMAT(1X'LGX=',I5)
      IF(SGFMT .NE. 0.0) GO TO 19
      IF(AX .GE. 1.0) GO TO 14
      M = LGX - 1
      GO TO 16
   14 IF(AX .LT. 10.0) GO TO 19
      M = LGX
```

```
   16 DIV = 10.0**M
      DO 18 I = 1,NI1
   18 XL(I) = XL(I)/DIV
   19 TL = FNI*FLOAT(NFMT)*SLBL
      IF(TL .LE. FL) GO TO 201
      WRITE (6,20)
   20 FORMAT(89H0ERROR RETURN FROM 'LABEL'--NOT ENOUGH SPACE TO PRINT AL
     1L THE LABELS(TOO MANY INTERVALS).)
      SGLBL = 1.0
C     SET UP THE CONSTANTS.
  201 C10 = 2.0*SLBL
C     IF(SGFMT .NE. 0.0) C10 = FLOAT(NCF-2)*SLBL
C     IF (SGFMT.NE.0..AND.(NCF-1).EQ.(LGX+1)) C10=FLOAT(NCF-1)*SLBL
   21 IF(NCT .LT. 0) GO TO 211
      C1 = 0.15
      C2 = SLBL  + 0.03 +SLBL/2.
      C3 = 0.03+SLBL/2.
      C4 = 0.2
      C5 = 0.2  + SSCL
      C8 = C2 + STTL + 0.1
      C9 = SLBL + 0.13+SLBL/3.
      GO TO 2110
  211 C1 = -0.15
      C2 = -0.03
      C3 = -(SLBL  + 0.03)
      C4 = -(0.2 + SSCL)
      C5 = -0.20
      C8 = -(SLBL + 0.13)
      C9 = -(SLBL + 0.13 + STTL)
 2110 CONTINUE
  212 IF(ND .NE. 0) GO TO 26
C     DRAW TICK MARKS AND LABELS ALONG X-AXIS.
      DO 24 I = 1,NI1
      NII = NI1 - I
      XT = X + FLOAT(NII)*DX1
      CALL SYSPLT(XT,Y,3)
      CALL SYSPLT(XT,Y+ C1 ,2)
      IF(SGLBL .NE. 0.0) GO TO 24
      IF(SGFMT .EQ. 0.0) GO TO 213
       CALL CONST(XL(NII+1),NCF,NCOL,NDIGT,SLBL,NI1,I,C10)
      CALL OUTCOR(BCDW,NW)
      WRITE (6,FMT) XL(NII + 1)
      CALL OUTCOR
C     IF (I.EQ.NI1) GO TO 2120
      GO TO 23
  213 CONTINUE
      CALL OUTCOR(BCDW,NW)
      WRITE (6,22) XL(NII+1)
      CALL OUTCOR
  22   FORMAT(F7.3)
```

```
      IF (I.EQ.NI1) GO TO 230
   23 CALL SYSSYM(XT-C10 ,Y- C2 ,SLBL,BCDW,NFMT,0)
      GO TO 24
  230 C10=C10-SLBL
       IF (NCT.LT.0) GO TO 231
      GO TO 23
  231 C10=C10-3./2.*SLBL
      IF (XL(1).GT.0.) C10=C10+0.5*SLBL
      GO TO 23
   24 CONTINUE
      GO TO 30
C     DRAW TICK MARKS AND LABELS ALONG Y-AXIS.
   26 DO 28 I = 1,NI1
      NII = NI1 - I
      YT = Y + FLOAT(NII)*DX1
      CALL SYSPLT(X,YT,3)
      CALL SYSPLT(X+ C1 ,YT,2)
      IF(SGLBL .NE. 0.0) GO TO 28
      IF(SGFMT .EQ. 0.0) GO TO 27
       CALL CONST(XL(NII+1),NCF,NCOL,NDIGT,SLBL,NI1,I,C10)
      CALL OUTCOR(BCDW,NW)
      WRITE (6,FMT) XL(NII+1)
      CALL OUTCOR
C     IF (I.EQ.NI1) GO TO 265
      GO TO 271
  272 C1C=C10-SLBL
      IF (NCT.LT.0) GO TO 273
      GO TO 271
  273 C10=C10-SLBL
      IF (XL(1).GT.0.) C10=C10+0.5*SLBL
      GO TO 271
   27 CONTINUE
      CALL OUTCOR(BCDW,NW)
      WRITE (6,22) XL(NII+1)
      CALL OUTCOR
      IF (I.EQ.NI1) GO TO 272
  271 CALL SYSSYM(X-C3  ,YT-C10,SLBL ,BCDW,NFMT,90.0)
   28 CONTINUE
C      WRITE(6,601) C10
  601  FORMAT(1X'C10=',E13.6)
C     PRINT THE TITLE
   30 IF(NCT .EQ. 0) GO TO 29
      IANCT = IABS(NCT)
      ANCT = IANCT
      T = 0.5*FL - 0.5*ANCT*STTL*6.0/7.0
      IF(ND .NE. 0) GO TO 38
      CALL SYSSYM(X+T,Y-C8,STTL,TITLE,IANCT,0)
      GO TO 29
   38 CALL SYSSYM(X-C9,Y+T,STTL,TITLE,IANCT,90.0)
C     PRINT SCALE FACTOR.
```

```
   29 IF(SGLBL .NE. 0.0) GO TO 40
      IF(M   .EQ. 0) GO TO 40
      C11 = FL - 2.0 + 17.0*SSCL*0.85
      SIDX = 0.8*SSCL
      IF(ND. NE. 0) GO TO 34
      CALL SYSSYM(X+FL-2.0,Y+C4 ,SSCL,SCL,17,0)
      IF(M .LT. 0) GO TO 31
      CALL OUTCOR(BCDW,NW)
      WRITE (6,301) M
      CALL OUTCOR
  301 FORMAT(I2)
      GO TO 33
   31 CONTINUE
      CALL OUTCOR(BCDW,NW)
      WRITE (6,32) M
      CALL OUTCOR
   32 FORMAT(I3)
   33 CALL SYSSYM(X+C11,Y+C5,SIDX,BCDW,3,0)
      GO TO 40
   34 CALL SYSSYM(X+C5  ,Y+FL-2.0,SSCL,SCL,17,90.0)
      IF(M .LT. 0) GO TO 35
      CALL OUTCOR(BCDW,NW)
      WRITE (6,301) M
      CALL OUTCOR
      GO TO 351
   35 CONTINUE
      CALL OUTCOR(BCDW,NW)
      WRITE (6,32) M
      CALL OUTCOR
  351 CALL SYSSYM(X+C4,Y+C11,SIDX,BCDW,3,90.0)
   40 CONTINUE
      RETURN
  50   WRITE (6,650)NV
 650   FORMAT(///5X,'VALUE OF NI (6TH ARGUMENT IN LABEL) IS TOO LARGE',
     1 1X,I20)
       STOP
  51   WRITE (6,651)NV
 651   FORMAT(///5X,'VALUE OF NI (6TH ARGUMENT IN VLABEL) IS TOC LARGE',
     1 1X, I20)
       STOP
   52  WRITE(6,652)
  652  FORMAT(//1X'ERROR FROM VLABEL--FORMAT HAS TO BE F FORMAT')
       STOP
      END
       BLOCK DATA
C LABEL
       COMMON /LBLCOM/ ILBL,SLBL,STTL,SSCL
       DATA ILBL,SLBL,STTL,SSCL /0,.12,.16,.10/
       END
       SUBROUTINE CONST(X,NCF,NC,ND,SLBL,NI1,I,C10)
```

```
C CONST                                        DATE OF OBJECT DECK 10-18-72
        IF (NCF.EQ.NC-1) GO TO 40
       IF (X.EQ.0.) GO TO 20
       ALG=ALOG10(ABS(X))
       LOGX=ALG+0.005
       LOGX=LOGX+1
       IF (ALG.LT.0.) LOGX=1
C      WRITE(6,602) LOGX
  602  FORMAT(1X'LOGX=',I5)
       NC2=NC-2
       NCT=NC2-ND-LOGX
       NCT1=NCT
       IF  (NCT.LT.0) NCT=0
       XNL=NCT
       IF (NI1.EQ.I)  GO TO 110
       NCT=NC-NCT
       NL=MOD(NCT,2)
       XNL=XNL+NCT/2
        IF (NL.NE.0) XNL=XNL+0.5
C      WRITE(6,600) NCT,NL,XNL
  600 FORMAT(1X'NCT=',I5,'NL=',I5,'XNL=',F8.3)
       GO TO 90
   20 XNL=NC -ND-2
        IF (NI1.EQ.I) GO TO 90
       XNL=XNL+0.5
       IF (ND.NE.0) XNL=XNL+0.5
C  INTEGER
  40   IF (ABS(X) .LT. 1.) X=0.
       IF (X.NE.0.) GOTO 50
        XNL=NCF-1
       IF (NI1.NE.I) XNL=XNL+0.4
       GO TO 90
   50  LOGX=   (ALOG10(ABS(X))) +0.005
       LOGX=LOGX+1
       XNL=NCF-LOGX
       IF (NI1.EQ.I)  GO TO 120
      XNL=LOGX/2 +XNL
       IF (MOD(LOGX,2).NE.0) XNL=XNL+0.4
   90 C10=XNL*SLBL*6./7.
C      WRITE(6,601) XNL
  601 FORMAT(1X'XNL=',F8.3)
  100 RETURN
C 110  XNL=NCT
  110   CONTINUE
       IF (X.GE.0..AND.NCT1.GE.0) XNL=XNL+1
       GO TO 90
  120  IF (X.LT.0.) XNL=XNL-1.
        GO TO 90
       END
```

IDENTIFICATION

MAXMIN/Subroutine* to find maximum and minimum elements of an array, either real or integer. - Assembler coded.

James Lo

September, 1971

USAGE

Calling sequence:

CALL MAXMIN (A, N, AMX, AMN)

where

A = one-dimensional array, real or integer

N = number of elements in array

= positive for real array

= negative for integer array

AMX = maximum value of array, real or integer

AMN = minimum value of array, real or integer

STORAGE

MAXMIN = $(280)_{10}$ bytes

This routine is available on the PDP-10 FORTRAN library and on the 370/155 FORTRAN library.

* Willis H. Booth Computing Center Report No. C267-214-370

```
./  PRINT  LIST
*
* MAXMIN                                   DATE OF OBJECT DECK 11-11-68
*         THE SUBROUTINE 'MAXMIN' FINDS THE UPPER AND LOWER BOUNDARIES
*         OF AN ARRAY, EITHER REAL OR INTEGER. THE CALLING SEGUENCE IS
*         AS FOLLOWS,
*
*               CALL MAXMIN(A,N,AMX,AMN)
*
*         WHERE A = ONE DIMENSIONAL ARRAY, REAL OR INTEGER.
*               N = LENGTH OF THE ARRAY. N MUST BE GREATER OR EQUAL TO 1
*                 = POSITIVE, IF A IS A FLOATING-POINT ARRAY.
*                 = NEGATIVE, IF A IS AN INTEGER ARRAY.
*             AMX = MAXIMUM VALUE OF THE ARRAY, REAL OR INTEGER.
*             AMN = MINIMUM VALUE OF THE ARRAY, REAL OR INTEGER.
*
R0        EQU   0
R1        EQU   1
R2        EQU   2
R3        EQU   3
I         EQU   4
A         EQU   5
AMX       EQU   6
AMN       EQU   7
BASE      EQU   8
N         EQU   9
AAMX      EQU   10
AAMN      EQU   11
FMX       EQU   0
FMN       EQU   2
FR        EQU   4
          ENTRY MAXMIN
MAXMIN    SAVE  (14,12),,*
          BALR  BASE,0
          USING *,BASE
          L     AAMX,8(R1)           GET ADDR. OF AMX.
          L     AAMN,12(R1)          GET ADDR. OF AMN.
          L     R2,4(R1)
          L     N,0(R2)              GET N.
          C     N,=F'1'              IS N = 1 ?
          BNE   NNE1                 NO. BRANCH.
          L     R2,0(R1)             YES. SET
          L     R2,0(R2)             AMX = A(1),
          ST    R2,0(AAMX)           AMN = A(1).
          ST    R2,0(AAMN)           AND RETURN.
RETN      RETURN (14,12),T
NNE1      LTR   N,N
          BNZ   NNZ                  BRANCH IF N .NE. 0.
          L     15,=V(IBCOM#)        IF N = 0, PRINT
          CNOP  0,4                  AN ERROR MESSAGE
```

```
         BAL   14,4(15)           AND TERMINATE THE JOB.
         DC    A(6),AL1(1),AL3(FM)
         BAL   14,16(15)
         CALL  EXIT
NNZ      L     A,0(R1)            INITIALIZE ARRAY POINTER.
         LPR   R3,N               REG.3 = IABS(N).
         BCTR  R3,0               SET UP UPPER LIMIT OF ARRAY
         SLL   R3,2               BY SETTING REG.3 = (N-1)*4.
         LA    I,4                SET I = 4 TO START THE LOOP.
         LA    R2,4               REG.2 = 4, INCREMENT OF I.
         LTR   N,N                IS N PLUS ?
         BP    FLOATPT            YES. A IS FLOATING-PT. BRANCH.
         L     AMX,0(A)           SET AMX = A(1) .
         LR    AMN,AMX            AMN = A(1) INITIALLY.
LOOP1    L     R0,0(A,I)          GET A(I).
         CR    AMX,R0             IS AMX >= A(I) ?
         BL    SETAMX             NO. BRANCH.
         CR    AMN,R0             IS AMN <= A(I) ?
         BNH   BXLE               YES. BRANCH.
         LR    AMN,R0             NO. SET AMN = A(I)
         B     BXLE               AND BRANCH.
SETAMX   LR    AMX,R0             IF AMX < A(I), SET AMX = A(I).
BXLE     BXLE  I,R2,LOOP1         STEP I AND LOOP BACK.
         ST    AMX,0(AAMX)        STORE THE INTEGER RESULTS
         ST    AMN,0(AAMN)        AND RETURN.
         B     RETN
FLOATPT  LE    FMX,0(A)           SET FMX = A(1),
         LE    FMN,0(A)           FMN = A(1) INITIALLY.
LOOP2    LE    FR,0(A,I)          GET A(I).
         CER   FMX,FR             IS FMX >= A(I) ?
         BL    SETFMX             NO. BRANCH.
         CER   FMN,FR             IS FMN <= A(I) ?
         BNH   FBXLE              YES. BRANCH.
         LER   FMN,FR             NO. SET FMN = A(I)
         B     FBXLE              AND BRANCH.
SETFMX   LER   FMX,FR             IF FMX < A(I), SET FMX = A(I).
FBXLE    BXLE  I,R2,LOOP2         STEP I AND LOOP BACK.
         STE   FMX,0(AAMX)        STORE THE FLOATING-POINT
         STE   FMN,0(AAMN)        RESULTS AND RETURN
         B     RETN
FM       DC    C'(/'' ERROR RETURN FROM MAXMIN--N = 0.'')'
         END
```

IDENTIFICATION

OUTCOR/Output to core storage - "370"*

John Hughes

Program date (latest version) - March 18, 1972

Write-up date - April, 1973

PURPOSE

To convert data with a FORMAT and place the EBCDIC formatted line image in core storage.

METHOD

DIMENSION A(10)

The sequence...

```
CALL OUTCOR (A,N)
WRITE (6,FMT1) LIST 1
WRITE (6,FMT2) LIST 2
   ...          ...
WRITE (6,FMTn) LISTn
CALL OUTCOR
```

will cause the output of the initial WRITE statement to be placed into core starting at A. N will be set to the number of words of the array A that have been used. Each succeeding WRITE statement will begin where the last one left off. N is INTEGER*4. A must be dimensioned large enough to contain the converted information. Length of format $\leq$ 132 (number of columns per line on output

* Willis H. Booth Computing Center Report No. C169-288-370

paper). LISTn should not repeat beyond length of specified format (i.e., no more than one line of output per WRITE statement).

USER ABEND CODES

2000 - improper FIOCS module loaded.

STORAGE

$(258)_{16}$ = 600 bytes

Example of usage of OUTCOR for printing output on plot:

```
      DIMENSION A(10)
      DATA B,C,D/3.14159,6.2831,12.5462/
      CALL OUTCOR (A,N)
      WRITE (6,610) B,C,D
      CALL OUTCOR
      CALL SYSSYM (.5,5.,.2,A,N*4,0.)
      CALL SYSEND (1,1)
610   FORMAT (1X,'B = ',F7.5,1X,'C = ',F6.4,1X,'D = ',F7.4)
      STOP
      END
```

The WRITE (6,610) statement following the CALL OUTCOR (A,N) does <u>not</u> create output on unit 6. It does create N alphanumeric words starting with A(1) which contain the <u>complete</u> output created by the WRITE statement. OUTCOR sets N = 9, i.e.,

A(1)		B		=
A(2)		3	.	1
A(3)	4	1	5	9
A(4)		C		=
A(5)		6	.	2
A(6)	8	3	1	
A(7)	D		=	
A(8)	1	2	.	5
A(9)	4	6	2	

```
./  PRINT  LIST
UCOR      TITLE 'OUTCOR.  FORTRAN CORE <--> CORE WRITE.'
* OUTCOR                                DATE OF OBJECT DECK 03-18-72
          SPACE 2
***  INTERFACE AND ASSEMBLY PARAMETERS.
*
*         CALL   OUTCOR(A,N)
*         WRITE (6,F) LIST
*         CALL   OUTCOR
*
*WILL CAUSE THE EBCDIC OUTPUT OF THE WRITE TO BE PLACED INTO CORE
*STARTING AT LOCATION A.  N WILL BE SET TO THE NUMBER OF WORDS OF THE
*ARRAY A THAT HAVE BEEN USED.
*
***************
*
OUTCOR    START X'10000'
*
*
BUFLEN    EQU   133                     LINE BUFFER LENGTH.
OUTUNIT   EQU   6                       NUMBER OF OUTPUT UNIT.
FOMASK    EQU   X'FF'                   CODE FOR FORMATTED OUTPUT.
INITOPT   EQU   0                       OPCODE FOR INITIAL ENTRY.
WRITEOPT  EQU   2                       OPCODE FOR WRITE ENTRY.
*
*
XBASE     EQU   12                      PROGRAM BASE REGISTER.
XFIO      EQU   11                      PTR TO FIOCS.
XOPT      EQU   10                      PTR TO OPCODE AND OPTION BYTES.
XLEN      EQU   9                       LENGTH OF LINE HOLDER.
XNWD      EQU   8                       PTR TO WORD COUNT.
XAREA     EQU   7                       PTR TO USER'S OUTPUT AREA.
*
*
XBLEN     EQU   3                       RETURN REG FOR BUFFER LENGTH.
XBUF      EQU   2                       RETURN REG FOR BUFFER ADDRESS.
XRECL     EQU   2                       INPUT REG FOR LINE LENGTH.
XDSRN     EQU   2                       INPUT REG FOR UNIT NUMBER PTR.
*
*
          EJECT
          SPACE 2
***  SET/RESET FIOCS FOR CORE-CORE WRITE.
*
*         CALL OUTCOR<(A,M)>
*
*
OUTCOR    CSECT
          SAVE  (14,12),,*              SAVE USER'S GPR'S.
          LR    XBASE,15                SET PROGRAM BASE.
```

```
         USING OUTCOR,XBASE        #XBASE
         LA    15,SVA                    SET NEW SAVE AREA.
         ST    13,4(,15)
         ST    15,8(,13)
         LR    13,15
         L     XFIO,=V(FIOCS#)           (XFIO)= PTR TO FIOCS MODULE.
         XI    SETSW,X'FF'               INVERT OUTCOR STATUS.
         BNZ   SETFIOCS                  BRANCH IF ACTIVATING CALL.
         MVC   0(PLUGSIZE,XFIO),UNPLUG RESTORE FIOCS.
RETURN   L     13,4(,13)                 RESTORE OLD SAVE AREA PTR.
         RETURN (14,12)                  RETURN TO CALLER.
SETFIOCS LA    0,3                       GET SET TO CHECK (1).
         NR    0,1                       IS (1) A VALID PARAM LIST PTR ?
         BNZ   ERROR1                    BRANCH IF NOT.
         TM    4(1),X'80'                HAVE WE GOT 2 PARAMETERS ?
         BNO   ERROR1                    BRANCH IF NOT.
         TM    3(1),X'03'                IS 1ST PARAM A FULL-WORD ?
         BNZ   ERROR1                    BRANCH IF NOT.
         TM    7(1),X'03'                IS 2ND PARAM A FULL-WORD ?
         BNZ   ERROR1                    BRANCH IF NOT.
         MVC   PLIST(8),0(1)             MOVE PTR'S TO AREA & WORD COUNT.
         SR    0,0                       SET BYTE COUNT = 0.
         ST    0,LENGTH
         MVC   UNPLUG(PLUGSIZE),0(XFIO) SAVE OVERLAYED FIOCS CODE.
*        CLC   UNPLUG(PLUGSIZE),TEMPLATE
*        BE    PROCEED
         B     PROCEED
         ABEND 2000,DUMP,STEP
PROCEED  MVC   0(PLUGSIZE,XFIO),PLUG     OVERLAY FIOCS WITH LINKAGE TO   *
                                         THE CORE-CORE WRITER.
         B     RETURN                    RETURN TO CALLER.
ERROR1   MVI   SETSW,0                   SET FLAG 'OUTCOR INACTIVE'.
         B     RETURN                    RETURN TO CALLER.
         DROP  XBASE               #XBASE
         EJECT
         SPACE 2
***   CORE <--> CORE WRITER.
*
*        (0) = RETURN ADDR - 2
*            = PTR TO AL1(OPCODE,OPTION)
*        (1) = A(OUTCORX)
*        (2) = PTR TO UNIT NUMBER (OPCODE = INIT)
*            = OUTPUT RECORD LENGTH (OPCODE = WRITE)
*
*
OUTCORX  DS    0H
         USING OUTCORX,1           #1
         STM   0,15,SVA                  SAVE USER'S GPR'S.
         DROP  1                   #1
         LR    XBASE,1                   SET PROGRAM BASE.
```

```
         USING OUTCORX,XBASE       #XBASE
         LR    XOPT,0                   (XOPT)= A(OPTION BYTES).
         CLI   0(XOPT),INITOPT          INITIAL ENTRY ?
         BE    INIT                     BRANCH IF SO.
         CLI   0(XOPT),WRITEOPT         WRITE ENTRY ?
         BNE   ERROR2                   BRANCH IF NOT.
         LM    XAREA,XLEN,PLIST         (XAREA)= A(OUTPUT FIELD)       *
                                        (XNWD) = A(WORD COUNT)         *
                                        (XLEN) = CURRENT LENGTH.
         LTR   XRECL,XRECL              OUTPUT LENGTH > 0 ?
         BNP   NOREC                    BRANCH IF NOT.
         LA    XLEN,0(XLEN,XRECL)       (XLEN) = NEW TOTAL LENGTH.
         BCTR  XRECL,0                  GET SET TO EX MVC.
         EX    XRECL,MVC                SEND OUTPUT LINE TO USER.
         LA    XAREA,1(XAREA,XRECL)     UPDATE AREA PTR.
         STM   XAREA,XLEN,PLIST         SAVE UPDATED PARAMETERS.
NOREC    LA    1,3(,XLEN)               (1)/4 = NUMBER OF FULL WORDS.
         SRA   1,2                      (1)= NUMBER OF FULL WORDS USED.
         ST    1,0(,XNWD)               SEND USER HIS WORD COUNT.
         SLA   1,2                      (1)= BYTE COUNT TO FULL WORD.
         MVC   BUF(BUFLEN),BLANKS       CLEAR OUTPUT BUFFER.
         SR    1,XLEN                   (1)= NUMBER OF BLANKS TO FILL  *
                                              USER'S FULL WORD.
         BZ    RETX                     BRANCH IF NONE.
         BCTR  1,0                      GET SET TO EX MVC.
         EX    1,MVC                    PAD USER'S LAST FULL WORD.
RETX     L     0,SVA                    RESTORE GPR 0.
         LR    1,0                      (1)= RETURN ADDRESS - 2.
         MVC   SVA+4*XBUF(8),=A(BUF,BUFLEN) SET BUFFER PARAMETERS.
         LM    2,15,SVA+8               RESTORE REMAINING GPR'S.
         B     6(,1)                    RETURN TO CALLER. (IBCOM)
*
*
INIT     L     1,0(,XDSRN)              (1)= UNIT NUMBER OR PTR.
         TM    0(XDSRN),X'01'           (1)= PTR ?
         BZ    *+8                      SKIP IF NOT.
         L     1,0(,1)                  (1)= UNIT NUMBER.
         CH    1,=Y(OUTUNIT)            IS THIS FOR OUTPUT UNIT ?
         BNE   ERROR2                   BRANCH IF NOT.
         CLI   1(XOPT),FOMASK           FORMATTED OUTPUT ?
         BE    RETX                     RETURN IF SO.
ERROR2   MVI   SETSW,0                  SET FLAG FOR 'OUTCOR INACTIVE'.
         L     XFIO,=V(FIOCS#)          (XFIO)= FIOCS PTR.
         MVC   0(PLUGSIZE,XFIO),UNPLUG RESTORE FIOCS.
         ST    XFIO,SVA+4               SET FIOCS ADDRESS FOR RETURN.
         LM    0,15,SVA                 RESTORE GPR'S.
         BR    1                        ENTER FIOCS.
*
*
MVC      MVC   0(*-*,XAREA),BUF
```

```
         EJECT
         SPACE 2
***  CONSTANTS AND WORK AREAS.
*
*
BLANKS   DC    CL(BUFLEN+1)' '         LINE BUFFER.
BUF      EQU   BLANKS+1
         LTORG ,                       LITERAL POOL.
PLIST    DC    3A(0)                   AREA, WORD COUNT PTR, BYTE COUNT
LENGTH   EQU   PLIST+8
PLUG     DS    0F                      OVERLAY FOR FIOCS.
         L     1,8(,1)                      (1)= A(CORE-CORE WRITER).
         BR    1                            LINK TO CORE-CORE WRITER.
SETSW    DC    AL2(0)                       OUTCOR STATUS SWITCH.
         DC    A(OUTCORX)
PLUGSIZE EQU   *-PLUG                  SIZE OF FIOCS OVERLAY.
UNPLUG   DS    CL(PLUGSIZE)            OVERLAYED FIOCS CODE.
TEMPLATE DC    X'071047F01022071058101F9A'
SVA      DS    18F                     REGISTER SAVE AREA.
         SPACE 2
         END   OUTCOR
```

IDENTIFICATION/A utility program for handling multi-file tapes with one DD card for the Fortran user.

READNF* - Read multiple files on a tape with one DD card.

REWFF - Rewind a multi-file tape to the beginning of its first physical file.

BSKPF - Backspace files.

Program Revised: October 20, 1971

Documentation Revised: May, 1973

USAGE

I. CALL READNF(IU)

or

CALL READNF(IU, K)

where

IU is an integer variable which defines the tape unit number.

K is an integer variable which indicates the number of files to be skipped forward. If K = 0 (skip no files), then K can be omitted.

After READNF has been called, the next Fortran read statement will start reading at the designated new file.

II. CALL REWFF(IU)

where IU is defined as above.

III. CALL BSKPF(IU, K)

where IU is defined as above.

---

* Willis H. Booth Computing Center Report No. C968-260-370

K is an integer variable which indicates the number of files the user desires to backspace. If K = 0 the tape is repositioned at the beginning of the file he has just finished reading. If, for example, the user just finished reading the third file and wishes to back up to the first file, then K should be set equal to 2.

RESTRICTIONS

a. The first logical file must be read before any of the entry points can be called.

b. READNF and BSKPF can only be called immediately after an end of file has been encountered.

c. The DCB parameter for all of the files must be the same.

d. REWFF cannot be called immediately following a call to READNF. Once the tape has been positioned to read a new file [by a call to READNF(IU)], the tape must be processed by at least one read statement before a call to REWFF is issued.

e. The attributes on the DD card that defines the data set used by these routines cannot be DUMMY.

ERROR MESSAGE

USER CODE 1001 - an illegal unit is referenced or a READNF is called before an end of file is reached.

EXAMPLE

```
C          To read ten files (file 2-11) and rewinding to beginning
               of file 1
           DO 10I=1,10
      60   Read (20,END=50)A,B,C
           Write (6,...)A,B,C
           GO TO 60
      50   IF (I. EQ. 10) GO TO 30
      10   CALL READNF (20)
      30   CALL REWFF (20)
           .
           .
           .
           END
         //FT20F001 DD LABEL=(2,BLP,,IN),...
```

STORAGE

$(2397)_{10}$

```
./  PRINT  LIST
RDNF     TITLE 'READNF.  SET-UP TO READ NEXT TAPE FILE (FORTRAN).'
* REWFF (READNF,BSKPK)                     DATE OF OBJECT DECK 10-20-71
         SPACE 2
***  INTERFACE AND ASSEMBLY PARAMETERS.
*
*  TO READ SEVERAL FILES OF A MULTIFILE TAPE ON FORTRAN UNIT U, HAVING
*SUPPLIED ONLY THE DD-CARD FOR THE FIRST FILE:
*
*//FTUUF001 DD LABEL=(1,BLP),...
*
*PLACE THE FORTRAN STATEMENT
*
*        CALL   READNF(U)
*
*IN THE 'END=' CODE EXECUTED WHEN A READ ENCOUNTERS A TAPE MARK.  THE
*FOLLOWING READS WILL THEN REFER TO THE NEXT FILE.
*
*        CALL   REWFF(U)   TO REWIND
*
REWFF    START X'10000'
*
*
XBASE    EQU   12                          PROGRAM BASE REGISTER.
XUB      EQU   11                          PTR TO FORTRAN UNIT BLOCK.
XDCB     EQU   10                          PTR TO DCB.
XJFCB    EQU   9                           PTR TO JFCB
*
*
REWFF    CSECT
         SAVE  (14,12),,*                  SAVE CALLER'S GPR'S
         USING REWFF,15             #15
         MVI   SWITCH,X'00'                SET FOR REWIND
         LA    14,BP                       SET BRANCH POINT
         LA    XBASE,READNF                SET BASE REG
         BR    14                          BRANCH TO COMMON CODE
         DROP  15                   #15
         EJECT
         SPACE 2
         ENTRY BSKPF
BSKPF    SAVE  (14,12),,*
       USING   BSKPF,15
         MVI   SWITCH,X'11'  SET SWITCH FOR BACKWORD SKIPFL
         LA    14,BP
         LA    XBASE,READNF
         BR  14
         DROP   15
         EJECT
       SPACE   2
***  ENTRY AND VALIDITY CHECKS.
```

```
*
*
         ENTRY READNF
READNF   SAVE  (14,12),,*              SAVE CALLER'S GPR'S.
         LR    XBASE,15                SET PROGRAM BASE.
         USING READNF,XBASE       #XBASE
         MVI   SWITCH,X'01'       SET FOR NEXT FILE
BP       LA    15,SVA                  GET NEW SAVE AREA.
         ST    13,4(,15)               CHAIN SAVE AREAS.
         ST    15,8(,13)
         LR    13,15
         TM    3(1),X'03'              UNIT NUMBER IN A FULL-WORD ?
         BNZ   BADUNIT                 BRANCH IF NOT.
         SR    14,14
         TM    0(1),X'80'
         BO    BP0
         L     15,4(0,1)
         L     14,0(0,15)
BP0      STH   14,SKPCT
         L     1,0(,1)                 (1)= A(UNIT NUMBER).
         L     1,0(,1)                 (1)= UNIT NUMBER.
         LTR   1,1
         BP    BP1                .GT.0
         BZ    BADUNIT            .EQ.0
         OI    SWITCH,X'02'       MARK WRTNF
         LPR   1,1
BP1      ST    1,BS3
         L     15,=V(IHCUATBL)         (15)= A(FORTRAN UNIT TABLE).
         USING UATBL,15           #15
         LH    0,UTNUMBER              (0)= 16*(NUMBER OF UNITS).
         SRA   0,4                     (0)= NUMBER OF UNITS.
         CR    1,0                     IS GIVEN UNIT TOO LARGE ?
         BH    BADUNIT                 BRANCH IF SO.
         BCTR  1,0                     (1) = UNIT # - 1.
         SLA   1,4                     (1)= DISPL. OF UNIT TABLE ENTRY.
         LA    15,UTSTART(1)           (15)= A(UNIT TABLE ENTRY).
         DROP  15                 #15
         USING UTENTRY,15         #15
         TM    UTUBPTR+3,X'01'         UNIT IN USE ?
         BO    BADUNIT                 BRANCH IF NOT.
         L     XUB,UTUBPTR             (XUB)= A(FORTRAN UNIT BLOCK).
         DROP  15                 #15
         USING UB,XUB             #XUB
         LA    XDCB,UBDCB              (XDCB)= A(UNIT DCB).
         USING IHADCB,XDCB        #XDCB
         TM    DCBOFLGS,X'10'          IS THE DCB OPEN ?
         BZ    BADUNIT                 BRANCH IF NOT.
         TM    UBCBYTE,X'40'           CLOSED, TYPE=T ?
         BO    BADUNIT                 BRANCH IF NOT.
         DROP  XUB                #XUB
```

```
         CLOSE ((XDCB),LEAVE)          CLOSE THE DCB.
         DEVTYPE DCBDDNAM,TEMP         (TEMP)= UCB TYPE FIELD.
         CLI   TEMP+2,X'80'            IS THE UNIT A TAPE ?
         BNE   BADUNIT                 BRANCH IF NOT.
*        FALL THROUGH FOR NEXT FILE SET-UP.
         EJECT
         SPACE 2
***   SET-UP TO READ THE NEXT FILE.
*
*
READJFCB L     1,DCBEXLST              SAVE OLD EXIT-LIST ADDR.
         ST    1,SVEXLST
         MVC   DCBEXLST+1(3),=AL3(EXLIST)   PLANT NEW EXIT-LIST ADDR.
         LA    0,JFCBLGTH              GET CORE FOR JFCB.
         GETMAIN R,LV=(0)
         ST    1,EXLIST                STORE PTR TO JFCB.
         MVI   EXLIST,X'87'            SET TO READ JFCB.
         LR    XJFCB,1                 SET JFCB BASE.
         RDJFCB ((XDCB),)              READ THE JFCB.
         USING JFCB,XJFCB        #XJFCB
         LH    1,JFCBFLSQ              (1)= OLD FILE SEQ. NUMBER.
         TM    SWITCH,X'10'
         BZ    BS5
         SH    1,SKPCT
         B     BS6
BS5      EQU   *
         LA    1,1(,1)                 BUMP IT TO NEXT.
         AH    1,SKPCT
BS6      EQU   *
         TM    SWITCH,X'01'      TEST OPTION
         BO    BS1                     NEXT FILE OPTION
         LA    1,1                     REWIND OPTION
BS1      STH   1,JFCBFLSQ              UPDATE FILE SEQ. NUMBER.
         OI    JFCBMASK+4,X'80'        SET FLAG TO REWRITE JFCB.
         OI    JFCBMASK+6,X'80'  TREAT INOUT AS INPUT ONLY
         DROP  XJFCB             #XJFCB
         PACK  TEMP(2),DCBDDNAM+5(3)   GET OLD SEQ.NO. FROM DDNAME.
         SP    TEMP(2),=PL1'1'         MINUS 1,IBCOM# WILL ADD ONE
         TM    SWITCH,X'01'      TEST OPTION
         BO    BS2                     NEXT FILE OPTION
         MVC   TEMP(2),ONE             REWIND OPTION
BS2      UNPK  TEMP+5(3),TEMP(2)       TRANSLATE IT TO ZONED FORMAT.
         OI    TEMP+7,X'F0'            SET LAST ZONE.
         MVC   TEMP(5),DCBDDNAM        (TEMP)= NEW DDNAME.
         OPEN  ((XDCB),INPUT),TYPE=J   REWRITE THE JFCB.
         CLOSE ((XDCB),REREAD)
*
***   TAPE IS NOW POSITIONED AT START OF NEXT FILE.
*
         MVC   DCBDDNAM(8),TEMP        (DCBDDNAM)= OLD DDNAME.
```

```
         L     1,SVEXLST                RESTORE OLD EXIT-LIST ADDR.
         ST    1,DCBEXLST
         LA    0,JFCBLGTH               FREE JFCB CORE.
         L     1,EXLIST
         FREEMAIN R,LV=(0),A=(1)
         TM    SWITCH,X'01'      TEST OPTION
         BO    RETURN                   NEXT FILE OPTION
         MVC   DCBDDNAM+4(4),=C'F001'
         MVC   DCBEXLST+1(3),=AL3(EXLIST)   PLANT NEW EXIT-LIST ADDR.
         MVI   EXLIST,X'80'
         OPEN  ((XDCB),INPUT)
         L     15,=V(IBCOM#)            REWIND OPTION
         CNOP  0,4
         BAL   14,44(0,15)              ISSUE A REWIND
BS3      DC    A(0)
         L     1,SVEXLST                RESTORE OLD EXIT-LIST ADDR.
         ST    1,DCBEXLST
RETURN   L     13,4(,13)                RETURN TO CALLER.
         RETURN (14,12)
         EJECT
         SPACE 2
***  ERROR PROCESSORS.
*
*
BADUNIT  LA    1,1001
         TM    SWITCH,X'02'
         BZ    BUABE
         LA    1,3001
BUABE    ABEND (1),DUMP,STEP
         EJECT
         SPACE 2
***  CONSTANTS AND WORK AREAS.
*
*
SVA      DC    18A(0)                   REGISTER SAVE AREA.
TEMP     DC    2A(0)                    8-BYTE WORK AREA.
EXLIST   DC    A(0)                     DCB EXIT LIST.
SVEXLST  DC    A(0)                     SAVE FOR OLD EXIT-LIST PTR.
ONE      DC    PL2'1'
SKPCT    DS    H
SWITCH   DS    CL1                      =0 -->  REWIND                 X
                                        ¬0 -->  NEXT FILE
         LTORG ,                        LITERAL POOL.
         EJECT
         SPACE 2
***  DCB DSECT.
*
*
         DCBD  DSORG=(PS),DEVD=(TA)
         EJECT
```

```
         SPACE 2
***   JOB FILE CONTROL BLOCK DSECT.
*
*
JFCB     DSECT
         IEFJFCBN
         EJECT
         SPACE 2
***   FORTRAN UNIT BLOCK DSECTS.
*
*
UATBL    DSECT ,                      UNIT TABLE.
         DS    H
UTNUMBER DS    H                      LENGTH OF UNIT TABLE BODY.
UTBYTES  DS    4X                     STANDARD UNIT NUMBERS.
UTSTART  EQU   *                      START OF UNIT TABLE BODY.
*
*
UTENTRY  DSECT ,                      UNIT TABLE ENTRY.
UTUBPTR  DS    A                      PTR TO UNIT BLOCK OR 1.
         DS    3A
*
*
UB       DSECT ,                      UNIT BLOCK.
         DS    2X
UBCBYTE  DS    X                      3RD FLAG BYTE.
BUFPTR   DS    6F                     BUFFERING INFORMATION.
UBDECB1  DS    5F                     DECB1
UBLIVC1  DS    F
UBDECB2  DS    5F                     DECB2
UBLIVC2  DS    F
UBDCB    DS    F                      START OF 88 BYTE DCB
         SPACE 2
         END   READNF
```

IDENTIFICATION

SYSEND*/"370" Subroutine to terminate a plot with or without plotting of the page number at the lower-right corner of the graph sheet

James Lo

April, 1967

Revised March, 1971

USAGE

CALL SYSEND (LAB, FLAG)

where:

LAB = page control

= 0, the pen will return to the current origin without plotting a page number.

> 0, the page number is plotted. Then the pen is moved to the next origin.

< 0, the pen is moved to the next origin without plotting the page number.

FLAG ≠ 0, the statements "ONE SHEET PLOTTED" and "ENDING PLOT LABELED..." will be printed on user's output.

= 0, the above printing will be suppressed.

NOTE:

SYSEND is used in the following six subroutines to terminate a plot.

---

* Willis H. Booth Computing Center Report No. C467-221-370

CPLOT	XYPLT
CPLT	PLOTY
XYPLOT	PLOTXY

DD CARD

Users calling any plot subroutines must provide the appropriate DD card as follows:

```
//SYSPLTDN   DD   SYSOUT=N     (for narrow paper plotting)
      or
//SYSPLTDW   DD   SYSOUT=W     (for wide paper plotting)
```

For general plotting information, see "CIT User's Guide to Calcomp Plotter."

```
./  PRINT  LIST
C      THE SUBROUTINE 'SYSEND' TERMINATES THE PLOT ON THE SHEET OF PAPER
C SYSEND                                DATE OF OBJECT DECK 11-21-69
C      . 'SYSEND' CALLS THE FOLLOWING SUBROUTINES,
C           SYSPLT,SYSWHR,OUTCOR.
C      THE DETAILED DESCRIPTION OF THE CALLING SEQUENCE , SEE THE
C      WRITE-UP.
C
       SUBROUTINE SYSEND(LAB,FLAG)
       DIMENSION BCDW(3)
       INTEGER PAGENO
       DATA PAGENO /0/
C      LAB = 0, MORE PLOT ON SAME PAGE. RETURN WITH NO ACTION.
       IF(LAB .EQ. 0) RETURN
       PAGENO = PAGENO + 1
       IF (LAB .GT. 0) GO TO 10
C      LAB .LT. 0, CLOSE PLOT WITHOUT PLOTTING PAGE NO.
       CALL SYSPLT(0,0,999)
       RETURN
C      LAB .GT. 0, PLOT THE PAGE NO., THEN CLOSE THE PLOT.
   10  CALL SYSWHR(PX,PY,PAT)
       Y = -0.4
       X = AINT((PX + PAT)/PAT)*PAT - 2.5
   14  CALL OUTCOR(BCDW,NW)
       WRITE (6,16) PAGENO
   16  FORMAT('PAGE ' I2)
       CALL OUTCOR
       CALL SYSSYM(X,Y,0.2,BCDW,7,0)
       CALL SYSPLT(0,0,999)
       IF(FLAG .EQ.0.0) RETURN
       WRITE (6,18) PAGENO
   18  FORMAT(/' ONE SHEET PLOTTED. PLOT ENDED ON PAGE ' I2)
       RETURN
       END
```

IDENTIFICATION

SYSPLT*/Basic Calcomp plotter subroutine. It moves the pen from its current position to a specified point with pen either up or down. - "370"

James Lo

Program date (latest revision) - February 21, 1973

USAGE

CALL SYSPLT (X, Y, IPN)

where:

X, Y = coordinates in inches of a given point to which the pen is moved

IPN = an integer

= 2 pen down

= 12 pen down. See *IPN Note*

= 3 pen up

= 13 pen up. See *IPN Note*

IPN Note

Before moving the pen, a new X and new Y are computed with offset and scale factor applied to X and Y, respectively, using the appropriate formula:

X = X * XFACT + XOFF

Y = Y * YFACT + YOFF

See calling sequence of SYSOFF later in the write-up.

---

* Willis H. Booth Computing Center Report No. C467-222-370

Note: To have a better understanding of the use of SYSPLT, you should consult "User's Guide to Calcomp Plotter." SYSPLT also provides the following 4 entries:

1. CALL SYSWHR (X, Y, PASIZE)

   To provide the user with the current position of the pen and the length of the graph sheet.

   X, Y = X, Y coordinates in inches of current pen position.

   PASIZE = current paper length in inches.

2. CALL SYSPSZ (I)

   To set a flag I

   where:

   I = 0, narrow paper is to be used (11 inches wide, perforation to perforation).

   I = 1, wide paper is to be used (29.6 inches wide, perforation to perforation).*

   If SYSPSZ is not called, I = 0 is the default value.

   * In addition, the user must also specify <u>wide</u> paper on the "plot request" slip.

3. CALL SYSXMX (XMAX)

   To reset a new right-hand boundary, where

   XMAX = maximum X in inches to which the plotter is allowed to move (XMAX was set to 50.0 internally).

   The subroutines CPLOT, CPLT, XYPLOT, XYPLT, PLOTY and PLOTXY allow a user to alter the limits along the X direction or XLNGTH, normally set to 15.0 inches. If XLNGTH is reset to more than 50.0, SYSXMX should be called to change XMAX accordingly.

4. CALL SYSOFF (XOFF, XFACT, YOFF, YFACT)

To reset the offset and scale factor for X and Y.

XOFF = offset for X in inches

XFACT = scale factor for X in inches

YOFF = offset for Y in inches

YFACT = scale factor for Y in inches

With the use of SYSOFF combined with IPN = 12 or 13 in call to SYSPLT (see previous page), the user can reset the origin to any place in the plotting area at any moment and use any unit other than inches. The Subroutines CPLOT, CPLT, XYPLOT, XYPLT, PLOTY and PLOTXY treat the 'lower-left corner' of the graph paper as the origin. For use of SYSOFF, see "User's Guide to Calcomp Plotter."

DD Card

Users calling any plot subroutines must provide the appropriate DD Card (for the new monitor):

```
//SYSPLTDN  DD  SYSOUT=N   (for narrow paper plotting)
                    or
//SYSPLTDW  DD  SYSOUT=W   (for wide paper plotting)
```

If the standard FORTGCLG procedure is used, the GO. step must be included.

NOTE: A call to SYSEND is required to terminate the plot. For general plotting information, see "CIT User's Guide to Calcomp Plotter."

```
./ PRINT LIST
SPIO     TITLE 'SYSPIO -- CALCOMP PLOTTER I/O ROUTINES.'
*    REVISED  7-2-73
         MACRO                     MACRO USED IN ENTRY 'SYSOFF'.
         MSET  &D,&LOC
         L     2,&D.(,1)
         LE    0,0(,2)
         STE   0,&LOC
         MEND
         EJECT
SYSPIO   CSECT
         DC    C'SYSWHR'
         DC    H'0'
BASE     EQU   11
*
         ENTRY SYSWHR              CALL SYSWHR(PENX,PENY,PATLENG)
         USING *,15
SYSWHR   EQU   *
         ST    2,SAVEA+28              THE ENTRY 'SYSWHR' RETURNS TO THE
         LE    0,PENX                  CALLING PROGRAM THE POSITION
         L     2,0(,1)             OF THE PEN IN INCHES WITH
         STE   0,0(,2)             RESPECT THE LOWER LEFT
         LE    0,PENY                  CORNER OF THE PLOT SHEET.
         L     2,4(,1)             LENGTH IN INCHES OF THE
         STE   0,0(,2)             PLOT PAPER BEING USED.
         LE    0,PATLENG
         L     2,8(,1)
         STE   0,0(,2)
         L     2,SAVEA+28
         BR    14
         SPACE 2
*
         ENTRY SYSPSZ              CALL SYSPSZ(SIZE)
         USING *,15
SYSPSZ   EQU   *
         STM   14,12,SAVEA+12
         BALR  BASE,0
         USING *,BASE
         L     1,0(,1)                 THE ENTRY 'SYSPSZ'
         L     0,0(,1)             RECEIVES  A FLAG FROM THE
*                                      CALLING PROGRAM. IF
         LTR   0,0                 SIZE = 0, NARROW PAPER
         BNZ   WPAPER              IS TO BE USED.
         BR    14                  OTHERWISE WIDE PAPER IS TO BE USED.
WPAPER   LA    0,1                      SET CODE = 1
         ST    0,CODE                  AND PASS THE INFORMATION TO SYSTEM.
         ST    0,SIZE                   SET SIZE = 1.
         USING IHADCB,10
         LA    10,DCBA
         CLOSE (DCBA,REREAD)
```

```
         MVC   DCBDDNAM(8),WPAPERDD  CHANGE DDNAME FOR WIDE PAPER.
         L     12,=A(SYSPLT)      INITIALIZE BASE REG. OF SYSPLT.
         NI    OPENTEST+1,X'00'    RESET SW-OPEN.
         BAL   14,OPENTEST
         PUT   DCBA,X
         LM    14,12,SAVEA+12
         BR    14              SET ABSTOP TO 29.5, THE WIDE PAPER WIDTH.
         DROP  BASE
         SPACE 2
*
         ENTRY SYSPAT          CALL SYSPAT(PATLENG)
         USING *,15
SYSPAT   EQU   *
         STM   14,12,SAVEA+12
         BALR  BASE,0
         USING *,BASE
         L     1,0(,1)         THE ENTRY 'SYSPAT' CHANGES
         LE    0,0(,1)             THE PATTERN LENGTH.
         STE   0,PATLENG
         STE   0,X                 SET X = PATLENG AND
         SR    0,0
         ST    0,CODE              SET CODE = 0 AND
         ST    14,SAVEA+12         PASS THE INFORMATION TO SYSTEM.
         L     12,=A(SYSPLT)       INITIALIZE BASE REG. OF SYSPLT.
         BAL   14,OPENTEST
         PUT   DCBA,X
         LM    14,12,SAVEA+12
         BR    14
         DROP  BASE
         SPACE 2
*
         ENTRY SYSXMX          CALL SYSXMX(XMAX)
         USING *,15
SYSXMX   L     1,0(,1)         THE ENTRY 'SYSXMX'
         LE    0,0(,1)         CHANGES THE RIGHT-HAND
         STE   0,ABSXMX        SIDE BOUNDARY, WHICH WAS
         BR    14              INTERNALLY SET TO 50.0 INCHES.
         SPACE 2
*
         ENTRY SYSOFF          CALL SYSOFF(XOF,XFAC,YOF,YFAC)
         USING *,15
SYSOFF   EQU   *
         ST    2,SAVEA+28          THE ENTRY 'SYSOFF'
         MSET  0,XOFF          ENTERS OFFSET FACTORS
         MSET  4,XFACT         INTO 'SYSPLT'.
         MSET  8,YOFF
         MSET  12,YFACT
         L     2,SAVEA+28
         BR    14
         EJECT
```

```
*        THE SUBROUTINE 'SYSPLT' GENERATES CODES TO MOVE THE PLOTTER
*        PEN FROM CURRENT POSITION TO (X,Y), WHERE X,Y ARE FLOATING-
*        POINT NUMBERS IN INCHES RELATIVE TO THE LOWER LEFT CORNER OF
*        THE PLOTTING PAPER. THE CALLING SEQUENCE IS AS FOLLOWS,
*
*             CALL SYSPLT(X,Y,IPEN)
*
*        WHERE X, Y = COORDINATES IN INCHES OF THE INTENDED PEN POSI.
*             IPEN = 2, PEN DOWN
*                  = 3, PEN UP
*                  = 12,13, MOVE PEN WITH OFFSET FACTORS APPLIED
*                  = 888, MOVE PEN TO OLD ORIGIN.
*                  = 999, MOVE PEN TO NEXT ORIGIN.
*        NOTE. 1. THE DISTANCE OF A SINGLE MOVE SHOULD NOT EXCEED
*                 30 INCHES.
*              2. THE PLOTTING CODES ARE NOT GENERATED IN 'SYSPLT'. IN-
*                 STEAD, THE INFORMATION IS PASSED ON TO THE SYSTEM
*                 THRU 'PUT' MACRO IN THE FOLLOWING FORM,
*
*                 CALL SYSPLTS(X,Y,CODE)
*
*        WHERE X,Y WERE DEFINED ABOVE
*             CODE = 0, X IS SET TO PATLENG AND Y IS A DUMMY.
*                  = 1, WIDE PAPER IS TO BE USED, X,Y ARE DUMMIES.
*                  = 2, PEN DOWN.
*                  = 3, PEN UP.
*                  = 4, MOVE PEN TO NEXT ORIGIN. X,Y ARE DUMMIES.
*                  =5, MOVE PEN TO OLD ORIGIN. X,Y ARE DUMMIES.
*             3. EACH SINGLE MOVE IS LIMITED TO 30 INCHES.
*
         ENTRY SYSPLT
SYSPLT   SAVE  (14,12),,*
         LR    12,15
         USING SYSPLT,12       GPR12 = NEW BASE REG.
         LR    11,13
         LA    13,SAVEA        LOAD THE SAVE REG.
         ST    13,8(,11)       AND CHAIN THE
         ST    11,4(,13)       SAVE AREAS.
START    BAL   14,OPENTEST
         OI    XOFFTEST+1,X'F0'    AT THE BEGINNING OF EACH PLOT,
         OI    YOFFTEST+1,X'F0'    RESET THE OFFSET SWITCHES.
ENDSW    B     LINPARM              HAS SYSPLT(X,Y,999) BEEN CALLED ?
         L     0,SIZE               YES. TEST FOR SIZE.
         LTR   0,0                  IF SIZE = 0, IT IS NARROW PAPER.
         BZ    RESETESW             BRANCH TO RESET END SW.
         LA    0,1                  IF SIZE = 1, CALL SYSPLTS(X,Y,1)
         ST    0,CODE               TO RESET THE WIDE PAPER FLAG
         PUT   DCBA,X               FOR PLOTTING A NEW SHEET.
RESETESW OI    ENDSW+1,X'F0'        RESET END SW FOR NORMAL SKIP.
LINPARM  L     1,24(11)             RELOAD INPUT PARA. LIST PTR.
```

```
         L       2,8(,1)
         L       2,0(,2)          LOAD PEN.
         C       2,=F'3'          IS PEN > 3 ?
         BH      PENGT3           YES. BRANCH.
         C       2,=F'2'
         BL      ERROR
STORPEN  ST      2,PEN            NO. STORE AWAY PEN.
         L       2,0(,1)
         LE      0,0(,2)          GET X.
XOFFTEST NOP     TESTX1
         ME      0,XFACT          IF PEN = 12,13 SET
         AE      0,XOFF           X = X*XFAC + XOFF
         B TESTX1
PENGT3   C       2,=F'13'
         BH      PENGT13
         SR      0,0              IF PEN = 12 OR 13,
         STC     0,XOFFTEST+1     TURN OFF MASK AT XOFFTEST
         STC     0,YOFFTEST+1     AND YOFFTEST SO THAT OFFSET AND
         S       2,=F'10'         FACTOR WILL BE APPLIED TO X AND Y.
         B       STORPEN
PENGT13  C       2,=F'999'        IS PEN = 999 ?
         BE      SETEND           YES. BRANCH.
         C       2,=F'888'              IS PEN = 888 ?
         BE      PEN888                 YES. BRANCH.
ERROR    EQU     *
         L       15,=V(IBCOM#)    IF PEN IS NOT 2,3
         CNOP    0,4              12,13 OR 999. IT IS
         BAL     14,4(15)         AN ERROR. WRITE A
         DC      A(6),AL1(1),AL3(FM1)  MESSAGE AND TERMINATE
         BAL     14,16(15)        THE JOB.
         CALL    EXIT
PEN888   LA      0,5                    IF PEN = 888
         B       SETEND+4               SET CODE = 5 AND BRANCH.
SETEND   LA      0,4                   IF PEN = 999,
         ST      0,CODE                SET CODE = 4 AND
         SR      0,0                    SET X = 0.
         ST      0,X                    SET Y = 0.
         ST      0,Y                    ALSO SET END-OF-PLOT
         NI      ENDSW+1,X'00'          SWITCH THEN
         B       OVERTEST              PASS INFORMATION TO SYSTEM.
TESTX1   CE      0,ABSXMX               IS X > ABSXMX ?
         BNH     STOREX                 NO. BRANCH TO STORE X.
         NI      OVERTEST+1,X'00'       OTHERWISE RESET SW AT OVERTEST
         B       RETN                   AND RETURN.
STOREX   ME      0,=E'200.'    STORE X COORDINATE
         STE     0,TEMPX
         SDR      0,0
         LE      0,TEMPX
         AW      0,MASK        CONVERT TO  INTEGER
         STD     0,TEMP
```

```
         L        2,TEMP+4
         O        2,MASK1
         ST       2,TEMP1
         LE       2,TEMP1
         AE       2,=E'0.0'
         LTDR     0,0
         BH       PASS
         LCER     2,2   CHANGE  SIGN
PASS     EQU       *
         ME       2,=E'0.005'
         STE      2,X
         L        2,4(,1)
         LE       0,0(,2)         GET Y.
YOFFTEST NOP      STOREY                IF PEN = 12,13
         ME       0,YFACT         SET Y = Y*FACT+ YOFF
         AE       0,YOFF
STOREY   CE       0,=E'-0.48'
         BNL      TESTUY
         FSET
         FWRF     (6),'(1X''PLOT IS EXCEEDED LOWER BOUNDARY IN Y DIRECTION*
                  ,POINT IS SKIPPED'')'
         FENDF
         NI       OVERTEST+1,X'00'
         B        RETN
TESTUY   L        2,SIZE
         C        2,=F'0'
         BE       NARROWP
         CE       0,=E'29.0'
ERRORP   BNH      SAVEY
         FSET
       FWRF       (6),'(1X''PLOT IS EXCEEDED UPPER BOUNDARY IN Y DIRECTION*
                  ,POINT IS SKIPPED'')'
         FENDF
         NI       OVERTEST+1,X'00'
         B        RETN
NARROWP    CE 0,=E'10.48'
         B        ERRORP
SAVEY    ME       0,=E'200.'       NUMBER OF STEPS
         STE      0,TEMPX
         SDR      0,0
         LE       0,TEMPX
         AW       0,MASK
         STD      0,TEMP
         L        2,TEMP+4
         O        2,MASK1
         ST       2,TEMP1
         LE       2,TEMP1     LOAD FLOATING  REG. 2
         AE       2,=E'0.0'
         LTDR        0,0
         BH        PASSY
```

```
         LCER  2,2
PASSY    EQU   *
         ME    2,=E'0.005'
         STE   2,Y                STORE Y COORDINATE.
OVERTEST NOP   CALLSYSP          IF NO OVERFLOW OCCURRED, BRANCH.
         OI    OVERTEST+1,X'F0'  OTHERWISE RESET MASK IN
         CLI   PEN+3,X'02'        'OVERTEST'.
         BNE   CALLSYSP           IF PEN .NE. 2 BRANCH.
         LA    0,3                ELSE SET PEN = 3
         ST    0,PEN             FOR MOVING TO X,Y WITH PEN UP.
CALLSYSP EQU   *
         PUT   DCBA,X
RETN     EQU   *
         L     13,SAVEA+4
         RETURN (14,12)
*
*        THIS SECTION OPENS THE OUTPUT FILE. THE ACTION IS BYPASSED
*        IF IT IS ALREADY OPEN. IT ALSO INITIALIZES THE 'OVERTEST'
*        SWITCH FOR CHECKING PLOTTING BOUNDARIES.
*
OPENTEST NOP   SKIPOPEN
         OI    OPENTEST+1,X'F0'
         OPEN  (DCBA,(OUTPUT))
         LA    10,DCBA
         USING IHADCB,10
         TM    DCBOFLGS,X'10'     IS OPEN SUCCESSFUL ?
         DROP  10
         BNZ   OPENOK             YES. BRANCH.
         FSET                     IF NOT, PRINT A MESSAGE AND STOP.
         FWRF  (6),'(//'' ERROR RETURN FROM SYSPLT--PLOT DD CARD MISSIN&
               G. SEE WRITE-UP.'')'
         FENDF
         CALL  EXIT
OPENOK   EQU   *
         OI    OVERTEST+1,X'F0'
SKIPOPEN BR    14
         SPACE 3
ABSXMX   DC    E'50.0'
PATLENG  DC    E'17.0'           INITIAL PATTERN LENGTH.
XOFF     DS    1F
YOFF     DS    1F
XFACT    DS    1F
YFACT    DS    1F
SAVEA    DS    18F
FM1      DC    C'(/72H ERROR RETURN FROM SYSPLT--IPEN IS NOT 2,3,12,13 &
               OR 999. JOB TERMINATED.)'
WPAPERDD DC    C'SYSPLTDW'
X        DS    1F
Y        DS    1F
CODE     DS    1F
```

```
SIZE     DC    F'0'                          SIZE = 0 INITIALLY FOR NARROW         &
                                               PAPER PLOTTING.
         DS    0D
MASK     DC    X'4E00000000000000'
TEMP     DS    2F
TEMPX    DC    F'0'
TEMP1    DC    F'0'
MASK1    DC    X'46000000'
*
PEN      EQU   CODE
PENX     EQU   X
PENY     EQU   Y
*
         LTORG
DCBA     DCB   DEVD=DA,DSORG=PS,EROPT=ACC,DDNAME=SYSPLTDN,MACRF=(PM),          &
               RECFM=FB,LRECL=12,BLKSIZE=600
         DCBD  DEVD=DA,DSORG=PS
         END
```

IDENTIFICATION

SYSSYM*/Plots alphanumeric symbols on Calcomp plotter - 370

James Lo

April, 1972

USAGE

Calling sequence

CALL SYSSYM(X, Y, SIZE, BCD, N, THETA)

where:

X, Y = starting position, in inches, of the symbols to be plotted, measured from the lower left-hand corner of the grid lines.

SIZE = height in inches of the symbol.

BCD = a one-dimensional array where the alphameric information is stored.

N = number of alphameric characters in EBCDIC. (See Note 1.)

THETA = angle of lettering with respect to X-axis (in degrees).

NOTE:

1. Normally N is a positive integer, which indicates that (X, Y) is to be the lower left corner of first character in 'BCD'. However, N may be negative. In this case 'BCD' will be treated as an integer constant of 0 to 14,

* Willis H. Booth Computing Center Report No. C367-218-370

denoting a special symbol to be plotted with (X, Y) as the center of the symbol. (See chart on page 343.)

2. By using SYSSYM along with OUTCOR, a subroutine on the system, the user can print on the plotting paper any output he would normally obtain through a Fortran WRITE statement. Thus the use of SYSSYM is greatly extended.

Example:

```
CALL SYSSYM (5.0,5.0,0.2,'SYMBOL TABLE',12,0.0)
ISYM = 12
CALL SYSSYM (1.0,2.0,0.1,ISYM,-1,0.0)
```

The above calling sequences would result in the plotting of:

SYMBOL TABLE   starting from (5.0,5.0):

and the symbol ⧖   at (1.0,2.0).

3. SYSSYM only plots the following symbols. (For each undefined symbol, SYSSYM plots a '?').

   a. Numerals 0 - 9.

   b. Alphabets A - Z and blanks.

   c. Special symbols [+], [-], [*], [/], [=], [(], [)], ['], [·], [±]*, [?], [%], [>], [<].

   d. Symbols for point plot. See page 343.

* Since there is no ± symbol on the keypunch machine, a # is treated as ±.

DD Card

Users calling any plot subroutines must provide the appropriate DD Card as follows:

```
//SYSPLTDN   DD   SYSOUT=N   (for narrow paper plotting)
                or
//SYSPLTDW   DD   SYSOUT=W   (for wide paper plotting)
```

This routine uses subprogram SYSPLT.

For general plotting information, see "CIT User's Guide to Calcomp Plotter."

NOTE:

A call to SYSSYM should precede calls to XYPLT, XYPLOT, CPLT, CPLOT, PLOTXY, or PLOTY, because the latter six routines have 'LAB' as an argument which is an option to terminate the plotting page. Otherwise, a call to SYSEND is required.

ISYM	SYMBOL
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

```
./  PRINT  LIST
*        THE SUBROUTINE 'SYSSYM' GENERATES A SERIES OF CALLS TO
*        'SYSPLT' TO PLOT AN ALPHANUMERIC SYMBOL. THE CALLING SEQUENCE
*        IS AS FOLLOWS,
*   REVISED 11-17-72
*
*        CALL SYSSYM(X,Y,H,STRING,N,THETA)
*
*        WHERE X,Y = FLOATING-PT NUMBERS IN INCHES WHICH SPECIFY
*        THE LOCATION OF THE LOWER-LEFT CORNER OF THE FIRST SYMBOL TO
*        BE PLOTTED.
*                H = HEIGHT OF THE SYMBOL IN INCHES.
*           STRING = STARTING ADDRESS OF THE SYMBOL STRING.
*                N = POSITIVE, DENOTES NO. OF SYMBOLS TO BE PLOTTED.
*                  = NEGATIVE, ONLY ONE SPECIAL SYMBOL WILL BE PLOTTED
*                    . 'STRING' IN THIS CASE MUST BE A FULL WORD INTE-
*                    GER FROM 0 TO 14, SPECIFYING A PARTICULAR SYMBOL.
*                  = 0, NO SYMBOL WILL BE PLOTTED.
*            THETA = THE ANGLE IN DEGREES BETWEEN THE BASE OF THE SYM-
*                  BOL AND THE HORIZONTAL AXIS.
*
*
*        THE GENERAL PURPOSE REGISTERS ARE ASSIGNED AS FOLLOWS,
*          GPR1--POINTER TO SYMBOL ADDRESS TABLE.
*          GPR2,3--X,Y OFFSET REGISTERS.
*          GPR4--COUNTS 1 TO NS, NO. OF STROKES FOR EACH SYMBOL.
*          GPR5--BASE REGISTER.
*          GPR6--POINTER TO INPUT BCD LETTERS.
*          GPR7--POINTER TO SYMBOL OFFSET TABLES.
*          GPR8--WORK REGISTER.
*          GPR9--COUNTS 1 TO N, THE NO. OF SYMBOLS TO BE PLOTTED.
*
NQUES    EQU   69
*
         ENTRY SYSSYM
         EXTRN IBCOM#
SYSSYM   CSECT
         SAVE  (14,12),,*
         LR    5,15            GPR5 = NEW BASE REG.
         USING SYSSYM,5
         LR    4,13            LOAD THE SAVE
         LA    13,SAVEA        REG.  AND CHAIN
         ST    13,8(4)         THE SAVE AREAS.
         ST    4,4(13)         GPR4 NOW CONTAINS THE ADDR. OF THE SAVE  X
                               AREA IN THE CALLING PROGRAM.
         LA    0,3                 SET PEN = 3 SO THAT THE
         ST    0,PEN           PLOTTING WILL START WITH PEN UP.
         L     2,0(,1)
         LE    0,0(,2)
         STE   0,X             STORE X.
```

```
         L      2,4(,1)
         LE     0,0(,2)
         STE    0,Y              STORE Y.
         L      2,8(,1)
         LE     0,0(,2)          GET HEIGHT OF THE SYMBOL.
         DE     0,=E'7.0'
         STE    0,FACTOR         FACTOR = HEIGHT/7.0
         L      6,12(,1)         GPR6 = A(BCD).
         L      2,16(,1)
         L      0,0(,2)          GET N.
         ST     0,N              STORE IT AWAY.
         LTR    0,0              TEST FOR N.
         BNM    NPLUSA               IF N IS NOT MINUS, BRANCH.
         L      2,8(,1)          OTHERWISE IT  IS A SPECIAL
         LE     0,0(,2)          SYMBOL. RECOMPUTE
         DE     0,=E'4.0'        FACTOR = HEIGHT/4.0
         STE    0,FACTOR
NPLUSA   L      2,20(,1)
         LE     0,0(,2)          GET THE ANGLE THETA.
         LTER   0,0                  TEST IF THETA = 0.
         BNZ    THETANZ              NO. BRANCH.
         SER    0,0              YES. SET
         STE    0,INCSIN         INCSIN = 0.
         STE    0,YT             YT = 0.
         LE     0,FACTOR
         STE    0,INCCOS         INCCOS = FACTOR.
         ME     0,=E'6.0'
         STE    0,XT             XT = FACTOR*6.
         B      ARRAYU           BRANCH TO COMPUTE UNITSIN, UNITCOS.
THETANZ  ME     0,=E'0.0174533'      IF THETA IS NOT ZERO,
         STE    0,THETA              CONVERT TO RADIANS AND
         LA     13,SAVEA         COMPUTE THE PARAMETERS
         CALL   SIN,(THETA)      AS FOLLOWS.
         ME     0,FACTOR
         STE    0,INCSIN         INCSIN = SIN(THETA)*FACTOR.
         ME     0,=E'6.0'
         STE    0,YT             YT = INCSIN*6.
         CALL   COS,(THETA)
         ME     0,FACTOR
         STE    0,INCCOS         INCCOS = COS(THETA)*FACTOR.
         ME     0,=E'6.0'
         STE    0,XT             XT = INCCOS*6.
*        THIS SECTION COMPUTES THE ARRAYS INITSIN(I) AND UNITCOS(I),
*        I = 1,7.
ARRAYU   SR     1,1              GPR1 = I IN UNITSIN(I).
         LA     2,4              GPR2 = 4, INCREMENT OF BYTES.
         LA     3,24             GPR3 = UPPER LIMIT OF I, WHICH IS 7.
         SER    2,2                  CLEAR FPR2.
LOOPU    AE     2,=E'1.0'        FPR2 = 1.0,2.0,....,7.0.
         LER    0,2
```

```
         ME    0,INCSIN
         STE   0,UNITSIN(1)     UNITSIN(I) = I*INCSIN.
         LER   0,2
         ME    0,INCCOS
         STE   0,UNITCOS(1)     INITCOS(I) = I*INCCOS.
         BXLE  1,2,LOOPU        I = 1,7.
         L     9,N
         LTR   9,9              TEST FOR N AGAIN.
         BZ    RETN                  IF N = 0, SKIP PLOTTING.
NNOTZRO  BP    CLEAR1           N = PLUS, BRANCH.
         LE    0,INCSIN         N = MINUS. PLOT ONE
         SE    0,INCCOS         SPECIAL SYMBOL.
         AER   0,0              MOVE (X,Y) FROM LOWER
         AE    0,X              LEFT CORNER TO THE CENTER OF
         STE   0,X              THE SYMBOL.
         LE    0,INCSIN         X = X + 2(INCSIN - INCCOS).
         AE    0,INCCOS
         AER   0,0
         LCER  0,0
         AE    0,Y
         STE   0,Y              Y = Y - 2(INCSIN + INCCOS).
         LA    9,1              SET N = 1 AND
         L     1,0(6)           LOAD GPR1 WITH SPECIAL SYMBOL
         LTR   1,1                   IS IT LESS THAN 0 ?
         BM    ILLEC                 YES. BRANCH TO PLOT SYMBOL '?'.
         C     1,=F'14'              OTHERWISE TEST AGAIN.
         BNH   FETCH                 BRANCH IF IT IS <= 14.
ILLEC    L     1,NNQUES              IF > 14, PLOT SYMBOL '?'.
         B     FETCH            CONSTANT AND BRANCH.
CLEAR1   SR    1,1
         IC    1,0(,6)          LOAD ONE BCD BYTE IN GPR1.
         IC    1,TRTABLE(1)     TRANSLATE IT TO A POINTER.
         LTR   1,1                   IS THE SYMBOL DEFINED ?
         BNZ   FETCH                 YES. BRANCH.
         L     1,NNQUES              OTHERWISE SET REG.1 TO PLOT '?'.
FETCH    SLA   1,2              CONVERT POINTER TO BYTES BY MULTIPLYING 4
         LH    4,TABLE(1)       LOAD NS(NO. OF STROKES) IN GPR4.
         LTR   4,4              IF NS = 0, IT IS A BLANK.
         BZ    NEXTSYM          SKIP PLOTTING. OTHERWISE
         LH    7,TABLE+2(1)
         STH   7,*+6            LOAD GPR7 WITH THE
         LA    7,0              POINTER TO THE OFFSET TABLE.
CLEAR2   SR    2,2              CLEAR GPR2.
         IC    2,0(7)           FETCH ONE PAIR OF OFFSETS.
         SRDL  2,4              SHIFT Y-OFFSET INTO GPR3.
         LE    0,X
         STE   0,XX             SET XX = X, YY = Y.
         LE    0,Y
         STE   0,YY
         LTR   2,2              IS X-OFFSET = 0?
```

```
         BZ    GETYOFF         YES. BRANCH.
         C     2,=F'7'         NO. IS IT A 7?
         BL    XLT7            NO. BRANCH.
         LA    8,3             YES. SET PEN = 3 FOR
         ST    8,PEN           MOVING TO NEXT POINT
         B     TEST1           WITH PEN UP.
XLT7     LE    0,X             IF XOFF IS NOT 0 AND LESS THAN 7,
         BCTR  2,0             COMPUTE XX, YY AS FOLLOWS,
         SLA   2,2             FIRST CONVERT GPR2 TO
         AE    0,UNITCOS(2)    POINTER IN BYTES. THEN COMPUTE
         STE   0,XX            XX = X + UNITCOS(XOFF).
         LE    0,Y
         AE    0,UNITSIN(2)
         STE   0,YY            YY = Y + UNITSIN(XOFF).
GETYOFF  SR    2,2
         SLDL  2,4             GET Y-OFFSET.
         LTR   2,2             IS Y-OFFSET = 0?
         BZ    PLOT            YES. BRANCH.
         LE    0,XX            NO. COMPUTE
         BCTR  2,0             XX,YY BY,
         SLA   2,2
         SE    0,UNITSIN(2)
         STE   0,XX            XX = XX - UNITSIN(YOFF).
         LE    0,YY
         AE    0,UNITCOS(2)
         STE   0,YY            YY = YY + UNITCOS(YOFF).
PLOT     LA    13,SAVEA
         CALL  SYSPLT,(XX,YY,PEN)  MOVE PEN ONE STROKE.
         LA    8,2             SET PEN TO DOWN POSITION.
         ST    8,PEN           AFTER EACH MOVE.
TEST1    BCT   4,NEXTMOVE      ARE THERE MORE STROKES OF THE SYMBOL?
         B     NEXTSYM         NO. BRANCH FOR NEXT SYMBOL.
NEXTMOVE LA    7,1(,7)         YES. MOVE THE POINTER TO NEXT
         B     CLEAR2          PAIR OF OFFSET DATA AND BRANCH BACK.
NEXTSYM  LA    8,3             SET PEN= 3 AT THE
         ST    8,PEN           BEGINNING OF NEXT SYMBOL.
         LE    0,X             ALSO SET
         AE    0,XT            X = X + XT.
         STE   0,X
         LE    0,Y             Y = Y + YT.
         AE    0,YT
         STE   0,Y
         BCT   9,MORESYM       ARE THERE MORE SYMBOL TO BE PLOTTED?
RETN     EQU   *
         L     13,SAVEA+4      NO. RESTORE REGISTERS
         RETURN (14,12),T,RC=0 AND RETURN.
         LM    14,12,12(13) RESTORE THE REGISTERS
         MVI   12(13),X'FF' SET RETURN INDICATION
         LA    15,0(0,0) LOAD RETURN CODE
         BR    14 RETURN
```

```
MORESYM  LA    6,1(,6)          YES. MOVE THE POINTER OF
         B     CLEAR1           BCD ARRAY AND BRANCH.
*
*        THIS IS THE TRANSLATION TABLE WHICH CONVERTS THE INPUT SYMBOLS
*          INTO A SET OF POINTERS.
*
TRTABLE  DC    256X'00'
         ORG   TRTABLE+C'+'
         DC    AL1(51)          PLUS SIGN.
         ORG   TRTABLE+C'-'
         DC    AL1(52)          MINUS SIGN.
         ORG   TRTABLE+C' '
         DC    AL1(53)          BLANK.
         ORG   TRTABLE+C'A'
         DC    AL1(15,16,17,18,19,20,21,22,23)  A THRU I.
         ORG   TRTABLE+C'J'
         DC    AL1(24,25,26,27,28,29,30,31,32)  J THRU R.
         ORG   TRTABLE+C'S'
         DC    AL1(33,34,35,36,37,38,39,40)  S THRU Z.
         ORG   TRTABLE+C'0'
         DC    AL1(41,42,43,44,45,46,47,48,49,50)  0 THRU 9.
         ORG   TRTABLE+C'*'
         DC    AL1(54)
         ORG   TRTABLE+C'/'
         DC    AL1(55)
         ORG   TRTABLE+C'='
         DC    AL1(56)
         ORG   TRTABLE+C'('
         DC    AL1(57)
         ORG   TRTABLE+C')'
         DC    AL1(58)
         ORG   TRTABLE+C','
         DC    AL1(59)
         ORG   TRTABLE+C'.'
         DC    AL1(60)
         ORG   TRTABLE+C'#'
         DC    AL1(61)
         ORG   TRTABLE+C'<'
         DC    AL1(62)
         ORG   TRTABLE+C'>'
         DC    AL1(63)
         ORG   TRTABLE+C'%'
         DC    AL1(64)
         ORG   TRTABLE+C''''
         DC    AL1(65)
         ORG   TRTABLE+C'$'
         DC    AL1(66)
         ORG   TRTABLE+C':'
         DC    AL1(67)
         ORG   TRTABLE+C'&&'
```

```
         DC    AL1(68)
*
*         NOTE. WHENEVER THE TABLE IS EXPANDED, THE SYMBOL 'NQUES'
*               MUST BE RE-EQUATED.
*
         ORG
*
*        EACH ENTRY OF 'TABLE' CONTAINS THE FOLLOWING INFORMATION FOR A
*          PARTICULAR SYMBOL,
*           (1) NO. OF MOVES (OR STROKES).
*           (2) ADDRESS OF THE OFFSET DATA FOR EACH MOVE.
*
TABLE    DS    0F
         DC    AL2(8),S(S0)
         DC    AL2(12),S(S1)
         DC    AL2(6),S(S2)
         DC    AL2(7),S(S3)
         DC    AL2(7),S(S4)
         DC    AL2(7),S(S5)
         DC    AL2(7),S(S6)
         DC    AL2(6),S(S7)
         DC    AL2(8),S(S8)
         DC    AL2(7),S(S9)
         DC    AL2(14),S(S10)
         DC    AL2(13),S(S11)
         DC    AL2(6),S(S12)
         DC    AL2(4),S(S13)
         DC    AL2(8),S(S14)
         DC    AL2(9),S(SA)
         DC    AL2(12),S(SB)
         DC    AL2(8),S(SC)
         DC    AL2(7),S(SD)
         DC    AL2(7),S(SE)
         DC    AL2(6),S(SF)
         DC    AL2(12),S(SG)
         DC    AL2(6),S(SH)
         DC    AL2(6),S(SI)
         DC    AL2(5),S(SJ)
         DC    AL2(6),S(SK)
         DC    AL2(3),S(SL)
         DC    AL2(5),S(SM)
         DC    AL2(4),S(SN)
         DC    AL2(11),S(SO)
         DC    AL2(7),S(SP)
         DC    AL2(11),S(SQ)
         DC    AL2(10),S(SR)
         DC    AL2(12),S(SS)
         DC    AL2(4),S(ST)
         DC    AL2(6),S(SU)
         DC    AL2(3),S(SV)
```

```
         DC    AL2(5),S(SW)
         DC    AL2(5),S(SX)
         DC    AL2(5),S(SY)
         DC    AL2(8),S(SZ)
         DC    AL2(9),S(N0)
         DC    AL2(5),S(N1)
         DC    AL2(8),S(N2)
         DC    AL2(13),S(N3)
         DC    AL2(9),S(N4)
         DC    AL2(9),S(N5)
         DC    AL2(11),S(N6)
         DC    AL2(6),S(N7)
         DC    AL2(17),S(N8)
         DC    AL2(12),S(N9)
         DC    AL2(5),S(NPLUS)
         DC    AL2(2),S(NMINUS)
         DC    F'0'
         DC    AL2(8),S(MULTI)
         DC    AL2(2),S(DIVIDE)
         DC    AL2(5),S(EQUAL)
         DC    AL2(4),S(LEFTPARN)
         DC    AL2(4),S(RIGTPARN)
         DC    AL2(6),S(COMMA)
         DC    AL2(5),S(PERIOD)
         DC    AL2(8),S(PLSMNS)
         DC    AL2(3),S(LSTHAN)
         DC    AL2(3),S(GRTHAN)
         DC    AL2(13),S(PERCENT)
         DC    AL2(6),S(QUOTE)
         DC    AL2(14),S(DOLLAR)
         DC    AL2(11),S(COLON)
         DC    AL2(11),S(AMPER)
         DC    AL2(11),S(QUESMARK)
*
*        'SYMBOL' CONTAINS THE OFFSET DATA FOR EACH SYMBOL TO BE
*          PLOTTED.
*
* ADDING EBCDIC FIGURES TO SYSSYM:
*NOTE: QUESMARK MUST ALWAYS BE THE LAST ENTRY IN 'TABLE.'
*
*OFFSET DATA HAS THE FOLLOWING INTERPRETATION:
*EACH PAIR OF HEX DIGITS DEFINES INTEGER COORDINATES (X,Y)
*OF A STROKE.  0<=X<=6      0<=Y<=7.  I.E. A SYMBOL IS
*DESIGNED ON A 6X7 GRID.
*THE PEN IS ALWAYS UP WHEN MOVING TO THE FIRST COORDINATES.
*IT IS DOWN THEREAFTER UNLESS A 7 IS SPECIFIED FOR THE X
*COORDINATE. IN THIS CASE, THE PEN IS UP WHEN MOVING
*TO THE FOLLOWING PAIR OF COORDINATES.
*EXAMPLE: THE MINUS SIGN IS X'0343'
*START AT COORDINATES (0,3) AND DRAW A STROKE TO COORDINATES (4,3)
```

```
*
SYMBOL   EQU   *
S0       DC    X'2224040040442422'
S1       DC    X'22241403011030414334 2422'
S2       DC    X'222401412422'
S3       DC    X'22420222202422'
S4       DC    X'22440022044022'
S5       DC    X'22240220422422'
S6       DC    X'22202402422422'
S7       DC    X'220044044022'
S8       DC    X'2244044400400022'
S9       DC    X'22042244222022'
S10      DC    X'224433130413110011314031 3322'
S11      DC    X'224202222024224400220 44022'
S12      DC    X'22440440 0022'
S13      DC    X'22202422'
S14      DC    X'0043034024007022'
SA       DC    X'0003430306173746 40'
SB       DC    X'413000073746453404344341'
SC       DC    X'4637170601103041'
SD       DC    X'37464130000737'
SE       DC    X'40000747070434'
SF       DC    X'000747070434'
SG       DC    X'434130100106173746703353'
SH       DC    X'000704444740'
SI       DC    X'103020271737'
SJ       DC    X'0110304147'
SK       DC    X'000703472540'
SL       DC    X'400007'
SM       DC    X'4047230700'
SN       DC    X'47400700'
SO       DC    X'2547463717060110304146'
SP       DC    X'00073746453404'
SQ       DC    X'4637170601103041464022'
SR       DC    X'00073746453404344340'
SS       DC    X'4637170605143443 41301001'
ST       DC    X'47072720'
SU       DC    X'070110304147'
SV       DC    X'072047'
SW       DC    X'4740240007'
SX       DC    X'4700704007'
SY       DC    X'2024072447'
SZ       DC    X'4000241434244707'
N0       DC    X'4637170601103041 46'
N1       DC    X'1627201030'
N2       DC    X'0617374644010040'
N3       DC    X'0617374645341434434 1301001'
N4       DC    X'3730204030324202 07'
N5       DC    X'470704344341301001'
N6       DC    X'1434434130100106173746'
```

```
N7        DC    X'060747462120'
N8        DC    X'06173746453414344341301001031405O6'
N9        DC    X'4433130406173746413O1001'
NPLUS     DC    X'2125230343'
NMINUS    DC    X'0343'
MULTI     DC    X'0244704204702125'
DIVIDE    DC    X'0047'
EQUAL     DC    X'0242704404'
LEFTPARN  DC    X'27050220'
RIGTPARN  DC    X'27454220'
COMMA     DC    X'203132222131'
PERIOD    DC    X'2021313020'
PLSMNS    DC    X'0141700444242622'
LSTHAN    DC    X'460340'
GRTHAN    DC    X'064300'
PERCENT   DC    X'0047261615252670212232312l'
QUOTE     DC    X'253637272636'
DOLLAR    DC    X'021131424304051636453626272O'
COLON     DC    X'2223333222702425353424'
AMPER     DC    X'5014152635341211203052'
QUESMARK  DC    X'0506173746452322702120'
NNQUES    DC    A(NQUES)
PEN       DS    1F
X         DS    1F
Y         DS    1F
N         DS    1F
XT        DS    1F
YT        DS    1F
XX        DS    1F
YY        DS    1F
FACTOR    DS    1F
THETA     DS    1F
INCSIN    DS    1F
INCCOS    DS    1F
UNITSIN   DS    7F
UNITCOS   DS    7F
SAVEA     DS    18F
FM1       DC    C'(//'' ERROR RETURN FROM SYSSYM--N .LE. 0. JOB TERMINAT&
                ED.'')'
          LTORG
          END
```

IDENTIFICATION

WRTNF*/Write Multiple Files With One DD Card.

Program Revised June 30, 1970

Documentation Revised January, 1973

USAGE

To write several files on magnetic tape on FORTRAN unit UU, having supplied only the DD card for the first file as follows:

```
//FTUUF001   DD   LABEL=(1,BLP),.....
```

Whenever the last record of any file has been written and an end-of-file mark is desired, then use

```
CALL WRTNF (UU)
```

The above procedure can be repeated for as many files as you wish to write. To rewind you MUST use

```
CALL ENDMF (UU)
```

This creates an extra end-of-file before rewinding.

RESTRICTIONS

The attributes on the DD card that defines the dataset used by these routines cannot be DUMMY.

Rewinding positions the tape at the beginning of its first physical file.

NOTES:

a) The DCB parameter for all files will be the same.

---

* Willis H. Booth Computing Center Report No. C470-336-370

b) The tape can only be rewound by a call to ENDMF. Do not use the FORTRAN statements END FILE or REWIND.

c) One file mark is placed between files. The system automatically creates an end-of-file at the end.

ABENDS

3001 - Illegal usage of WRTNF

LENGTH

$(ICE)_{16}$ = 462 bytes

EXAMPLE

```
C       WRITE 10 FILES ON TAPE 17
        DO 1 I = 1,10
        IF (I.NE.1) CALL WRTNF (17)
        .
        .
        .
  1     WRITE (17) List
        CALL ENDMF (17)
        STOP
        END
  //FT17F001  DD  LABEL=(1,BLP),....
```

```
./  PRINT  LIST
      SUBROUTINE WRTNF (FILE)
C WRTNF (ENDMF)                          DATE OF OBJECT DECK 06-30-70
      INTEGER*4 FILE,F
      END FILE FILE
      IX=-FILE
      CALL READNF (IX)
      RETURN
      ENTRY ENDMF (F)
      END FILE F
      IX=-F
      CALL REWFF (IX)
      RETURN
      END
```

IDENTIFICATION

XYPLOT*/370 Subroutine for continuous line plots specified by their coordinates X(I), Y(I). (Plotting area is 15" wide by 10" high.)

James Lo

Program date (latest version) - September 26, 1969

Write-up date - May, 1973

USAGE

CALL XYPLOT(N, X, Y, XMIN, XMAX, YMIN, YMAX, DD, LAB)

where:

N = Total number of points to be plotted ($\leq$ size of the X, Y arrays)

X = One-dimensional array containing X-coordinate

Y = One-dimensional array containing Y-coordinate.

XMIN = X-value for left-hand edge of the plotting area

XMAX = X-value for right-hand edge of the plotting area

YMIN = Y-value for bottom edge of graph paper

YMAX = Y-value for top edge of the plotting area

DD = One-dimensional array of length 3. If DD (1) $\neq$ 0, DD(1) and DD(2) will be treated as 8 EBCDIC characters and plotted on the upper right corner of graph paper at the end of each plot (when LAB $\neq$ 0). If DD(1) = 0, such plotting will be suppressed.

---

* Willis H. Booth Computing Center Report No. C167-201-370

DD(3) $\neq$ 0, plotter information such as scale, label, etc., will be printed on the user's output after each plot. DD(3) = 0, such printing will be suppressed.

LAB = Page control

= 0, this plot is not the last plot on the graph sheet.

> 0, last plot on the graph sheet. Page number will be plotted on the lower right corner.

< 0, last plot on graph sheet. Page number will not be plotted.

NOTE:

1. Any values outside the designated range of MIN and MAX will not be plotted.
2. Normally XYPLOT plots on an area of 15 inches by 10 inches, where 15 inches is the length along the X-direction (measured from the left-hand edge of the paper) and 10 inches the width along the Y-direction. If the user wishes to alter these dimensions, he can enter the information through a labeled COMMON in the calling program as follows:

```
COMMON/COMPLO/ITEST, XLNGTH, YLNGTH
```

where,

ITEST must be set = 1

XLNGTH = altered X-length in inches

YLNGTH = altered Y-length in inches

NOTE:

Both values must be provided even if only one is to be altered. If YLNGTH > 10., the user must initiate his program with

CALL SYSPSZ (1) and specify wide paper on the "Plot Request" slip.

DD Cards

Users calling any plot subroutines must provide the appropriate DD card:

//SYSPLTDN DD SYSOUT=N

//SYSPLTDW DD SYSOUT=W

If the FORTGCLG standard procedure is used, the GO. step should be used in the DD card.

H Compilation

Storage = $(4BE)_{16}$ + System plot routines

Timing = 167 ms for 1440 points - 15" span

For general plotting information, see "CIT User's Guide to Calcomp Plotter." This routine uses subprograms: SYSEND, SYSPLT, SYSSYM, BLOCK DATA.

```
./  PRINT  LIST
      SUBROUTINE XYPLOT(N,X,Y,XMN,XMX,YMN,YMX,DD,LAB)
C XYPLOT                                    DATE OF OBJECT DECK 09-26-69
C
C     N = TOTAL NO. OF POINTS TO BE PLOTTED.
C     X = ARRAY OF ABSCISSA.
C     Y = ARRAY OF ORDINATES.
C     XMN, XMX = RANGE OF X
C     YMN, YMX = RANGE OF Y.
C     LAB = 0, PLOT ON SAME SHEET OF PAPER.
C     LAB .GT. 0, PLOT TERMINATES CURRENT SHEET OF PAPER.
C     LAB = -1, PRINTING OF JOB SEQUENCE NUMBER IS SUPPRESSED.
C     DD(1),DD(2) = BCD TITLE TO BE PLOTTED ON THE UPPER RIGHT CORNER
C                   AT THE END OF EACH PLOT AS AN IDENTIFICATION. IF
C                   DD(1) = 0 THIS PLOTTING WILL BE SUPPRESSED.
C     DD(3) = 0, PRINTING OF PLOTTER INFORMATION LIKE SCALE, LABEL ETC
C               ON USER'S OUTPUT WILL BE SUPPRESSED.
C     DD(3) .NE. 0, THE ABOVE INFORMATION WILL BE PRINTED.
C
      COMMON/COMPLO/ITEST,XLNG,YLNG
       REAL LBOUND
      DIMENSION X(1),Y(1),DD(1)
      INTEGER PEN
      DATA XC,YC/1HX,1HY/
       RBOUND = XMX
       IF (XMN .LE. XMX) GO TO 5
       RBOUND = XMN
    5  CONTINUE
       IF (YMN.LE.YMX) GO TO 10
       TBOUND = YMN
       BBOUND = YMX
   10  CONTINUE
      IF(ITEST .EQ. 1) GO TO 12
      XLNGTH = 15.0
      YLNGTH = 10.0
      GO TO 13
   12 XLNGTH = XLNG
      YLNGTH = YLNG
   13 IF(N .GT. 0) GO TO 18
   14 WRITE (6,16) N,XMN,XMX,YMN,YMX
   16 FORMAT(//' ERROR RETURN FROM ''XYPLOT''--ONE OF THE FOLLOWING ARGU
     1MENTS HAS WRONG VALUE'/10X,'N,XMN,XMX,YMN,YMX ='I10,4E18.6)
      RETURN
   18 IF(XMX .EQ. XMN .OR. YMX .EQ. YMN) GO TO 14
      SX = XLNGTH/(XMX-XMN)
      SY = YLNGTH/(YMX-YMN)
      PEN = 3
      DO 30 I = 1,N
       IF (X(I).GT.RBOUND)GO TO 28
      XX = (X(I) - XMN)*SX
```

```
      YY = (Y(I) - YMN)*SY
      CALL SYSPLT(XX,YY,PEN)
      PEN = 2
      GO TO 30
   28 PEN = 3
   30 CONTINUE
      IF(DD(3) .EQ. 0.0) GO TO 35
      WRITE (6,331)
  331 FORMAT('0XYPLOT COMPLETED.')
      WRITE (6,34) XC,XMN,XMX,YC,YMN,YMX
   34 FORMAT(5X5H THE A1,27H COORDINATE IS SCALED FROM 1PE10.3,4H TO
     1 E10.3)
   35 IF(LAB .EQ. 0) RETURN
      IF(DD(1) .EQ. 0.0) GO TO 32
      CALL SYSSYM(13.0,9.8,0.2,DD,8,0)
   32 CALL SYSEND(LAB,DD(3))
      RETURN
       END
       BLOCK DATA
C XYPLOT
       COMMON /COMPLO/ IPLO,XPLO,YPLO
       DATA IPLO,XPLO,YPLO /0,15.,10./
C
       END
```