#338

VOYAGER 1 & 2

UV SPECTRAL DATA RECORDS

77-084A-04A
77-076A-04A

Table of Contents

1. Introduction

2. Errata/Change Log

3. LINKS TO RELEVANT INFORMATION IN THE ONLINE NSSDC INFORMATION SYSTEM

4. Catalog Materials

    a. Associated Documents

    b. Core Catalog Materials

---

1. INTRODUCTION:

The documentation for this data set was originally on paper, kept in NSSDC's Data Set Catalogs (DSCs). The paper documentation in the Data Set Catalogs have been made into digital images, and then collected into a single PDF file for each Data Set Catalog. The inventory information in these DSCs is current as of July 1, 2004. This inventory information is now no longer maintained in the DSCs, but is now managed in the inventory part of the NSSDC information system. The information existing in the DSCs is now not needed for locating the data files, but we did not remove that inventory information.

The offline tape datasets have now been migrated from the original magnetic tape to Archival Information Packages (AIP's).

A prior restoration may have been done on data sets, if a requestor of this data set has questions; they should send an inquiry to the request office to see if additional information exists.

2. ERRATA/CHANGE LOG:

NOTE: Changes are made in a text box, and will show up that way when displayed on screen with a PDF reader.

When printing, special settings may be required to make the text box appear on the printed output.

Version	Date	Person	Page	Description of Change
01				
02				

3 LINKS TO RELEVANT INFORMATION IN THE ONLINE NSSDC INFORMATION SYSTEM:

http://nssdc.gsfc.nasa.gov/nmc/

[NOTE: This link will take you to the main page of the NSSDC Master Catalog. There you will be able to perform searches to find additional information]

4. CATALOG MATERIALS:

a. Associated Documents To find associated documents you will need to know the document ID number and then click here.
http://nssdcftp.gsfc.nasa.gov/miscellaneous/documents/

b. Core Catalog Materials

VOYAGER 1 & 2

UV SPECTRAL DATA RECORDS

77-076A-04A PSPA-00114

77-084A-04A

This data set has been restored. There was originally one 7-track, 800 BPI tape written in Binary. There is one restored tape. The DR tape is a 3480 cartridge and the DS tape is 9-track, 6250 BPI. The original tape was created on a HP 21MX computer and the restored tape was created on an IBM 9021 computer. The DR and DS numbers along with the corresponding D number are as follows:

DR#	DS#	D#	FILES	TIME SPAN
DR004827	DS004827	D045044	1	09/12/77 - 11/15/79

Voyager 1 & 2

UV Spectral Data Records

77-076A-04A

77-084A-04A

This data set catalog contains 1 sample tape. The tape is 7-track, 800 bpi created on an Hewlett-Packard 21MX computer. The tapes may have been provided to the experimenter by JPL in 9-track format. Attempts to interpret these tapes were not very successful at the Data Center. It is possible that the PI has a complete set of these tapes and that the data center was only provided this sample tape.

DD Number	DC Number	Time-span
DD-045044	DC-024096	[redacted] - [redacted]

CENTER FOR SPACE SCIENCES
TELEPHONE: (602) 745-6767

February 3, 1983

Dr. Kent Hills
NSSDC
Goddard Space Flight Center
Code 601
Greenbelt, MD 20771

77-084A04
~~77-084A-04~~

Subject: Voyager UVS Data Set

Dear Dr. Hills:

The Voyager UVS experiment principal investigator is maintaining the data set as described in previous communications. The enclosed catalogues list the periods of available data. Please update your listing accordingly.

With best regards,

Lyle Broadfoot

A. Lyle Broadfoot
P.I. UVS Experiment

xc: Charles Stembridge
Jet Propulsion Laboratory

UNIVERSITY OF SOUTHERN CALIFORNIA, 3625 E. AJO WAY, TUCSON, ARIZONA 85713

VOYAGER

ULTRAVIOLET SPECTROMETER EXPERIMENT

Catalogue of Voyager 1 Ultraviolet Data Tapes

Period: Day 255 1977 to Day 319 1982

Number of Tapes: 492

HPEDR	FDSC1	FDSC2	SCET1					SCET2				
			YR	DAY	HR	M	S	YR	DAY	HR	M	S
1	21655	198016	77	255	10	05	22	77	314	04	46	28
2	198016	207000	77	314	04	46	52	77	317	04	33	29
3	207006	213040	77	317	04	38	53	77	319	05	06	05
4	213041	220959	77	319	05	06	17	77	321	20	33	17
5	221000	230403	77	321	20	33	29	77	324	23	48	29
6	233009	247341	77	325	20	40	41	77	330	15	30	53
7	247342	254840	77	330	15	31	05	77	333	03	30	05
8	254841	264800	77	333	03	30	17	77	336	10	58	05
9	264901	269454	77	336	10	58	17	77	338	00	29	17
10	274613	285127	77	339	17	32	29	77	343	05	43	42
11	285127	293750	77	343	05	43	42	77	346	02	49	30
12	293750	300700	77	346	02	49	30	77	348	10	09	30
13	300700	303853	77	348	10	09	30	77	349	11	39	56
14	303853	313930	77	349	11	40	00	77	352	20	10	06
15	313931	322312	77	352	20	10	18	77	355	15	07	06
16	322312	331027	77	355	15	07	06	77	358	12	55	31
17	331027	340118	77	358	12	55	31	77	361	13	36	31
18	340118	349617	77	361	13	36	31	77	364	17	35	31
19	349617	358623	77	364	17	35	31	78	002	17	39	54
20	358623	367137	78	002	17	40	06	78	005	13	51	42
21	367138	375602	78	005	13	51	54	78	008	09	23	42
22	375602	385629	78	008	09	23	42	78	011	17	44	54
23	385629	393529	78	011	17	44	42	78	014	08	56	54
24	393529	402950	78	014	08	56	54	78	017	12	15	54
25	402950	412237	78	017	12	25	54	78	020	14	39	18
26	412237	421327	78	020	14	39	06	78	023	15	19	31
27	421327	429238	78	023	15	19	31	78	026	06	40	19
28	429238	436740	78	026	06	40	19	78	028	18	42	07
29	436740	444817	78	028	18	42	07	78	031	11	11	07
30	444817	454425	78	031	11	11	07	78	034	16	05	31
31	454425	463232	78	034	16	05	31	78	037	14	35	31
32	463232	475122	78	037	14	35	31	78	041	13	39	31
33	475122	482536	78	041	13	39	31	78	044	01	02	55
34	482537	491012	78	044	01	03	07	78	046	20	43	43
35	491013	499702	78	046	20	43	55	78	049	18	11	44
36	499703	507623	78	049	18	11	56	78	052	09	40	32
37	507624	511813	78	052	09	40	44	78	053	19	08	01
38	511813	515217	78	053	19	08	05	78	054	22	23	08
39	515217	519555	78	054	22	23	20	78	056	09	17	56
40	519555	528444	78	056	09	17	56	78	059	08	20	44
41	528444	537325	78	059	08	20	44	78	062	07	17	44
42	537325	546809	78	062	07	17	44	78	065	11	04	56
43	546809	555114	78	065	11	04	56	78	068	05	32	44
44	555114	564847	78	068	05	32	44	78	071	11	35	20
45	564847	580033	78	071	11	35	20	78	076	12	59	56
46	580033	596524	78	076	12	59	56	78	082	00	53	57
47	596525	630834	78	082	00	53	31	78	093	11	32	45
48	630845	664826	78	093	11	33	33	78	104	19	18	21
49	664826	710602	78	104	19	18	21	78	120	01	23	10
50	710602	748700	78	120	01	23	10	78	132	18	09	34
51	748700	774936	78	132	18	09	34	78	141	12	14	22

52	774937	799938	78 141 12 15 10	78 149 20 15 58
53	799939	842055	78 149 20 16 46	78 163 21 17 37
54	842055	868222	78 163 21 17 41	78 172 14 27 59
55	868224	913300	78 172 14 28 47	78 187 14 58 23
56	913302	1022300	78 187 14 59 11	78 223 23 33 36
60	1022300	1097200	78 223 22 57 36	78 248 22 21 36
61	1097700	1102557	78 249 02 45 36	78 250 17 19 14
62	1102557	1125020	78 250 17 19 37	78 258 04 50 18
63	1125021	1138530	78 258 04 50 22	78 262 16 58 19
64	1138531	1160316	78 262 16 56 22	78 269 23 11 07
65	1162004	1200950	78 270 12 36 48	78 283 12 25 48
66	1200950	1202041	78 283 12 25 52	78 283 21 06 42
67	1203804	1268740	78 284 11 01 15	78 308 05 54 19
68	1268741	1371029	78 308 05 54 23	78 340 04 56 44
69	1371029	1373751	78 340 04 56 48	78 341 02 50 09
70	1373751	1384304	78 341 02 50 13	78 344 15 01 10
71	1384352	1386700	78 344 15 38 59	78 345 10 10 05
72	1386701	1389826	78 345 10 10 09	78 346 11 18 11
73	1389826	1391753	78 346 11 18 34	78 347 02 52 24
74	1391753	1394534	78 347 02 52 27	78 348 01 01 23
75	1394542	1396318	78 348 01 07 26	78 348 15 12 10
76	1396318	1406336	78 349 15 12 14	78 351 23 26 11
77	1406337	1439336	78 351 23 26 59	78 362 23 26 11
78	1439337	1460335	78 362 23 26 59	79 004 23 26 07
79	1460336	1462739	79 004 23 26 10	79 005 18 40 59
80	1462739	1463335	79 005 18 40 59	79 005 23 26 07
81	1463336	1469337	79 005 23 26 10	79 007 23 27 43
82	1469341	1474246	79 007 23 30 12	79 009 14 46 11
83	1474246	1478323	79 009 14 46 11	79 010 23 16 12
84	1478338	1482458	79 010 23 27 47	79 012 08 32 02
85	1482458	1487242	79 012 08 32 06	79 013 22 43 02
86	1467338	1492929	79 013 23 27 47	79 015 20 08 44
87	1492929	1496754	79 015 20 08 48	79 017 02 53 02
88	1496754	1502336	79 017 02 53 05	79 018 23 26 53
89	1502337	1508336	79 018 23 27 01	79 020 23 26 53
90	1508338	1513857	79 020 23 27 47	79 022 19 43 43
91	1513857	1517415	79 022 19 43 01	79 023 23 57 21
92	1517415	1523336	79 023 23 57 25	79 025 23 56 38
93	1523338	1525538	79 025 23 27 47	79 026 17 04 10
94	1525538	1527811	79 026 17 03 47	79 027 11 06 47
95	1527811	1530419	79 027 11 06 51	79 028 08 00 45
96	1530419	1532941	79 028 08 00 48	79 029 04 18 55
97	1532942	1535134	79 029 04 18 59	79 029 21 49 17
98	1535135	1537625	79 029 21 49 25	79 030 17 41 21
99	1537625	1539821	79 030 17 42 03	79 031 11 14 21
99.5	1539821	1540337	79 031 11 14 24	79 031 15 27 20
100	1540337	1542342	79 031 15 27 01	79 032 07 31 14
101	1542342	1544413	79 032 07 31 14	79 032 23 55 45
102	1544413	1546433	79 032 23 55 45	79 033 16 12 12
103	1546433	1548451	79 033 16 12 12	79 034 08 26 07
104	1548451	1550649	79 034 08 26 13	79 035 02 01 19
104.5	1550650	1551554	79 035 02 01 23	79 035 09 17 17
105	1551555	1553808	79 035 09 18 07	79 036 03 04 02

106	1553808	1556408	79 036 03 04 06	79 036 23 51 55
107	1556408	1558437	79 036 23 51 55	79 037 16 15 05
108	1558437	1560942	79 037 16 15 05	79 038 12 19 34
109	1560942	1563327	79 038 12 19 34	79 039 07 19 13
110	1563327	1566213	79 039 07 19 13	79 040 06 20 28
111	1566213	1568803	79 040 06 19 49	79 041 03 00 31
112	1568803	1571355	79 041 03 00 31	79 041 23 41 07
113	1571355	1573413	79 041 23 42 07	79 042 15 56 31
114	1573413	1575430	79 042 15 56 31	79 043 09 09 58
115	1575430	1577441	79 043 08 09 58	79 044 00 18 40
116	1577441	1579537	79 044 00 18 40	79 044 17 03 32
117	1579537	1581553	79 044 17 03 32	79 045 09 15 45
118	1581553	1584044	79 045 09 15 45	79 046 05 09 14
119	1584044	1586105	79 046 05 09 14	79 046 21 25 25
120	1586105	1588556	79 046 21 25 29	79 047 17 18 46
121	1588556	1590810	79 047 17 18 50	79 048 11 05 58
122	1590810	1592951	79 048 11 05 58	79 049 04 26 25
123	1592951	1595009	79 049 04 26 25	79 049 20 41 19
124	1595009	1597146	79 049 20 41 19	79 050 13 58 15
125	1597146	1599241	79 050 13 58 19	79 051 06 42 36
126	1599241	1601303	79 051 06 42 40	79 051 23 00 28
127	1601303	1603721	79 051 23 00 28	79 052 18 26 52
128	1603721	1605735	79 052 18 26 52	79 053 10 37 48
129	1605735	1607332	79 053 10 37 48	79 053 23 23 42
130	1607334	1609401	79 053 23 24 35	79 054 15 46 36
131	1609401	1611650	79 054 15 46 40	79 055 10 02 02
132	1611650	1613332	79 055 10 02 06	79 055 23 23 42
133	1613334	1615343	79 055 23 24 35	79 056 15 31 57
134	1615343	1617616	79 056 15 31 49	79 057 09 34 11
135	1617616	1619747	79 057 09 34 15	79 058 02 47 40
136	1619748	1621859	79 058 02 47 47	79 056 19 44 53
137	1621859	1623959	79 058 19 44 57	79 059 12 33 16
138	1623959	1625743	79 059 12 33 20	79 060 02 44 32
139	1625743	1627809	79 060 02 43 49	79 060 19 04 57
140	1627809	1630118	79 060 19 04 37	79 061 13 36 03
141	1630118	1632436	79 061 13 36 07	79 062 08 14 23
142	1632437	1634332	79 062 08 15 01	79 062 23 23 42
143	1634331	1636622	79 062 23 22 13	79 063 17 39 42
144	1636623	1638653	79 063 17 39 46	79 064 10 03 58
145	1638653	1639936	79 064 10 04 02	79 064 20 14 54
145.5	1639936	1640701	79 064 20 14 58	79 065 02 10 30
146	1640701	1642843	79 065 02 10 30	79 065 19 32 40
147	1642844	1644850	79 064 19 32 44	79 066 11 37 51
148	1644850	1647003	79 066 11 37 55	79 067 04 36 40
149	1647004	1649154	79 067 04 36 44	79 067 22 05 15
150	1649154	1651412	79 067 22 05 19	79 068 15 55 38
151	1651412	1653609	79 068 15 55 42	79 069 09 29 24
152	1653609	1655729	79 069 09 29 28	79 070 02 33 28
153	1655730	1657912	79 070 02 33 32	79 070 19 55 23
154	1657912	1659915	79 070 19 55 27	79 071 11 58 16
155	1659916	1661924	79 071 11 58 20	79 072 04 05 26
156	1661925	1664121	79 072 04 05 30	79 072 21 38 18
157	1664121	1666036	79 072 21 38 22	79 073 13 02 50

158	1666036	1668725	79 073 13 02 54	79 074 10 29 34
159	1668725	1670747	79 074 10 29 34	79 075 02 47 40
160	1670747	1672752	79 075 02 47 48	79 075 18 51 35
161	1672752	1674546	79 075 18 51 38	79 076 09 10 47
162	1674546	1676330	79 076 09 10 50	79 076 23 22 14
163	1676331	1678459	79 076 23 22 22	79 077 16 33 28
164	1678500	1680536	79 077 16 33 32	79 078 09 02 35
165	1680536	1682330	79 078 09 02 39	79 078 23 22 14
166	1682330	1684306	79 078 23 21 32	79 079 15 02 50
167	1684306	1686303	79 079 15 02 20	79 080 07 00 17
168	1686303	1688330	79 080 07 00 21	79 080 23 22 14
169	1688331	1690408	79 090 23 22 22	79 081 15 52 00
170	1690408	1692525	79 081 15 52 03	79 082 08 54 04
171	1692525	1694329	79 082 08 53 34	79 082 23 21 28
172	1694330	1696334	79 082 23 21 32	79 083 15 24 55
173	1696334	1698427	79 083 15 25 18	79 084 08 07 21
174	1698427	1700329	79 084 08 07 25	79 084 23 21 28
175	1700330	1702355	79 084 23 21 32	79 085 15 42 12
176	1702355	1706701	79 085 15 42 16	79 087 02 10 56
177	1706701	1709259	79 087 02 10 22	79 087 22 57 01
178	1709259	1711342	79 087 22 57 05	79 088 15 31 31
179	1711342	1713520	79 088 15 31 35	79 089 08 49 40
180	1713520	1715329	79 089 08 49 43	79 089 23 21 28
181	1715330	1717405	79 089 23 21 32	79 090 15 50 08
182	1717405	1719625	79 090 15 50 12	79 091 09 41 38
183	1719625	1721747	79 091 09 41 42	79 092 02 47 33
184	1721747	1723930	79 092 02 47 41	79 092 20 10 03
185	1723930	1726048	79 092 20 10 07	79 093 13 12 00
186	1726048	1728149	79 093 13 12 04	79 094 06 00 42
187	1728149	1730329	79 094 06 00 46	79 094 23 21 13
188	1730330	1733328	79 094 23 21 32	79 095 23 20 23
189	1733330	1736329	79 095 23 21 32	79 096 23 21 28
190	1736330	1738753	79 096 23 21 32	79 097 18 52 59
191	1738754	1741227	79 097 18 52 52	79 098 14 31 52
192	1741228	1743549	79 098 14 32 15	79 099 09 13 21
193	1743550	1746622	79 099 09 13 44	79 100 09 39 27
194	1746622	1748745	79 100 09 39 08	79 101 02 45 49
195	1748745	1751323	79 101 02 45 53	79 101 23 16 40
196	1751324	1753349	79 101 23 16 44	79 102 15 36 58
197	1753349	1755200	79 102 15 37 01	79 103 06 09 55
198	1755200	1757413	79 103 06 09 32	79 103 23 56 40
199	1757414	1835759	79 103 23 56 44	79 130 02 56 50
200	1835759	1928404	79 130 02 56 54	79 160 23 48 45
201	1928404	2033400	79 160 23 48 45	79 195 23 45 34
202	2033332	2111304	79 195 23 23 10	79 221 23 00 46
203	2111300	2143510	79 221 22 57 34	79 232 16 41 38
204	2146028	2162035	79 233 12 55 59	79 238 21 02 15
205	2162036	2213300	79 238 21 02 27	79 255 22 57 35
206	2213304	2252300	79 255 23 00 47	79 268 22 57 35
207	2253816	2300200	79 269 11 10 23	79 284 22 09 36
208	2300200	2313915	79 284 22 09 36	79 289 11 57 38
209	2313915	2314252	79 289 11 58 01	79 289 14 51 46
209.5	2315612	2345056	79 290 01 31 12	79 299 21 18 24

210	2346628	2355920	79 300 09 44 00	79 303 12 01 56
211	2355918	2358812	79 303 12 00 12	79 304 11 07 12
212	2358812	2363645	79 304 11 07 35	79 306 01 58 21
213	2363646	2377529	79 306 01 58 44	79 310 16 57 10
214	2377529	2417300	79 310 16 57 33	79 323 22 57 37
215	2417308	2421847	79 323 23 04 01	79 325 11 35 53
216	2421948	2435304	79 325 11 36 16	79 329 23 00 49
217	2435308	2462050	79 329 23 04 01	79 338 21 14 12
218	2462051	2484023	79 338 21 14 35	79 346 04 52 03
219	2484023	2487924	79 346 04 52 26	79 347 12 05 16
220	2487925	2552737	79 347 12 05 59	80 004 02 39 11
221	2552737	2558356	80 004 02 39 15	80 005 23 42 25
222	2558400	2586539	80 005 23 45 37	80 015 09 05 22
223	2586540	2594409	80 015 09 05 45	80 017 23 52 47
224	2594409	2608602	80 017 23 53 10	80 022 17 23 13
225	2608602	2633235	80 022 17 23 36	80 030 22 38 09
226	2633235	2654045	80 030 22 38 11	80 037 21 09 39
227	2654045	2656708	80 037 21 09 41	80 038 18 16 09
228	2656708	2693414	80 038 18 16 13	80 050 23 57 24
228.5	2693414	2696003	80 050 23 57 28	80 051 20 36 19
229	2696003	2698308	80 051 20 36 27	80 052 15 04 25
229.5	2698309	2720703	80 052 15 04 48	80 060 02 12 42
230	2720703	2741102	80 060 02 12 46	80 066 21 23 48
231	2741103	2753008	80 066 21 24 11	80 070 20 40 17
232	2753009	2763028	80 070 20 40 56	80 074 04 56 29
233	2763029	2774948	80 074 04 56 52	80 078 04 24 02
234	2775348	2781103	80 078 07 36 02	80 080 05 24 42
235	2781104	2785934	80 080 05 25 05	80 081 20 13 02
236	2785934	2801931	80 081 20 13 25	80 087 04 10 51
237	2801932	2822010	80 087 04 11 14	80 093 20 42 13
238	2822010	2824808	80 093 20 42 17	80 094 19 04 37
238.5	2824808	2825401	80 094 19 04 41	80 094 23 47 11
239	2825402	2858548	80 094 23 47 14	80 106 01 12 22
240	2858548	2861616	80 106 01 12 45	80 107 01 35 01
241	2861617	2888359	80 107 01 35 16	80 115 23 45 35
242	2888400	2923355	80 115 23 45 39	80 127 15 42 08
243	2923356	2926150	80 127 15 42 31	80 128 14 02 02
244	2926151	2929502	80 128 14 02 25	80 129 16 35 49
245	2929503	2963658	80 129 16 36 13	80 141 02 08 18
246	2963658	3005347	80 141 02 08 42	80 154 23 35 13
247	3005347	3011225	80 154 23 35 36	80 156 22 29 57
248	3011225	3023148	80 156 22 30 20	80 160 22 00 03
249	3023200	3035040	80 160 22 09 39	80 164 21 05 51
250	3035040	3048105	80 164 21 06 14	80 169 05 26 24
251	3048106	3053945	80 169 05 26 28	80 171 04 22 08
252	3053945	3070159	80 171 04 21 38	80 176 14 09 09
252.5	3070159	3071136	80 176 14 08 54	80 176 21 50 28
253	3071948	3089033	80 177 04 24 04	80 182 21 00 21
254	3089518	3091926	80 183 00 48 04	80 183 20 06 43
255	3091926	3110259	80 183 20 06 47	80 189 22 57 36
256	3110300	3114807	80 189 22 57 40	80 191 11 03 33
257	3114808	3116035	80 191 11 04 04	80 191 21 02 13
258	3117336	3134827	80 192 07 26 28	80 198 03 19 22

259	3134827	3137245	80 198 03 19 26	80 198 22 46 01
260	3137245	3155100	80 198 22 46 05	80 204 21 22 22
261	3155101	3159411	80 204 21 22 26	80 206 07 55 08
262	3159412	3173159	80 206 07 55 16	80 210 22 09 36
264	3172621	3177628	80 210 17 38 30	80 212 09 44 12
265	3177628	3179159	80 212 09 44 35	80 212 22 09 36
266	3179200	3185017	80 212 22 09 40	80 214 20 47 38
267	3185018	3195653	80 214 20 48 01	80 218 10 03 47
268	3195653	3198426	80 218 10 04 10	80 219 08 06 21
269	3198426	3200337	80 219 08 06 09	80 219 23 27 42
270	3200338	3203810	80 219 23 28 05	80 221 03 06 00
271	3203811	3207320	80 221 03 06 12	80 222 07 13 29
272	3207325	3209903	80 222 07 17 24	80 223 03 47 51
273	3209903	3213110	80 223 03 48 14	80 224 05 29 45
274	3213110	3214607	80 224 05 29 22	80 224 17 27 07
275	3214607	3217557	80 224 17 27 30	80 225 17 19 00
276	3217557	3220105	80 225 17 19 23	80 226 13 25 51
277	3220106	3222833	80 226 13 26 10	80 227 11 24 03
278	3222833	3225206	80 227 11 24 26	80 228 06 14 21
279	3225206	3227531	80 228 06 14 44	80 229 00 58 54
280	3227532	3229850	80 229 00 59 17	80 229 19 37 33
281	3229850	3232201	80 229 19 37 56	80 230 14 10 50
282	3232201	3234550	80 230 14 10 12	80 231 09 13 22
283	3234550	3236925	80 231 09 14 04	80 232 04 05 47
284	3236930	3237356	80 232 04 09 22	80 232 07 42 43
285	3237356	3239750	80 232 07 42 44	80 233 02 49 26
286	3239750	3242125	80 233 02 49 49	80 233 21 41 43
287	3242125	3244857	80 233 21 42 06	80 234 19 43 38
288	3244858	3247229	80 234 19 43 46	80 235 14 32 36
289	3247229	3249703	80 235 14 32 59	80 236 10 12 22
290	3249704	3252029	80 236 10 12 34	80 237 04 57 10
291	3252030	3254357	80 237 04 57 34	80 237 23 43 27
292	3254358	3256714	80 237 23 43 50	80 238 18 20 34
293	3256715	3259040	80 238 18 21 24	80 239 13 06 00
294	3259041	3261452	80 239 13 06 23	80 240 08 27 06
295	3261452	3263835	80 240 08 27 25	80 241 03 25 35
296	3263835	3266159	80 241 03 25 58	80 241 22 09 18
297	3266159	3268436	80 241 22 08 32	80 242 16 14 14
298	3268436	3270755	80 242 16 14 18	80 243 10 53 28
299	3270755	3273213	80 243 10 53 51	80 244 06 19 52
300	3273213	3275533	80 244 06 20 15	80 245 00 59 48
301	3275533	3277853	80 245 01 00 11	80 245 19 39 44
302	3277853	3280219	80 245 19 40 07	80 246 14 24 36
303	3280219	3282527	80 246 14 24 59	80 247 08 55 38
304	3282528	3284843	80 247 08 55 46	80 248 03 32 26
305	3284844	3287159	80 248 03 32 34	80 248 22 09 18
306	3287200	3289427	80 248 22 09 22	80 249 16 07 00
307	3289427	3292002	80 249 16 07 04	80 250 12 35 06
308	3292002	3294337	80 250 12 35 29	80 251 07 27 23
309	3294338	3296214	80 251 07 27 46	80 251 22 21 16
310	3296215	3298504	80 251 22 21 24	80 252 16 37 16
311	3298505	3300831	80 252 16 37 20	80 253 11 22 50
312	3300832	3302941	80 253 11 23 29	80 254 04 18 35

313	3302942	3305416	80 254 04 18 58	80 254 23 58 22
314	3305416	3307850	80 254 23 58 45	80 255 19 37 41
315	3307851	3310212	80 255 19 38 08	80 256 14 19 25
316	3310213	3312527	80 256 14 20 07	80 257 08 55 38
317	3312528	3314837	80 257 08 55 46	80 258 03 27 42
318	3314838	3317159	80 258 03 28 05	80 258 22 09 18
319	3317200	3319508	80 258 22 09 22	80 259 16 40 28
320	3319509	3321840	80 259 16 40 51	80 260 11 29 26
321	3321840	3324215	80 260 11 29 49	80 261 06 21 36
322	3324215	3326648	80 261 06 21 59	80 262 02 00 05
323	3326648	3328850	80 262 01 59 46	80 262 19 37 22
324	3328850	3331309	80 262 19 37 22	80 263 15 04 44
325	3331309	3333717	80 263 15 05 07	80 264 10 23 19
326	3333717	3336122	80 264 10 23 42	80 265 05 39 25
327	3336123	3338159	80 265 05 39 48	80 265 22 09 18
328	3338200	3340645	80 265 22 09 22	80 266 17 58 06
329	3340646	3343018	80 266 17 58 29	80 267 12 47 46
330	3343018	3345444	80 267 12 48 09	80 268 08 20 46
331	3345445	3347808	80 268 08 21 20	80 269 03 04 25
332	3347809	3350131	80 269 03 04 48	80 269 21 46 31
333	3350131	3352537	80 269 21 46 54	80 270 17 02 56
334	3352541	3354859	80 270 17 06 12	80 271 11 44 59
335	3354900	3357218	80 271 11 45 22	80 272 06 23 46
336	3357218	3359652	80 272 06 23 46	80 273 02 03 37
337	3359652	3362035	80 273 02 02 59	80 273 21 01 40
338	3362035	3364348	80 273 21 02 03	80 274 15 36 09
339	3364349	3366604	80 274 15 36 32	80 275 09 24 50
340	3366604	3368901	80 275 09 25 13	80 276 03 46 31
341	3368902	3371159	80 276 03 46 58	80 276 22 09 18
342	3371200	3373612	80 276 22 09 22	80 277 17 31 33
343	3373613	3375945	80 277 17 31 56	80 278 12 21 55
344	3375946	3378309	80 278 12 22 18	80 279 07 04 55
345	3378309	3380642	80 279 07 05 18	80 280 01 55 37
346	3380642	3382910	80 280 01 55 40	80 280 19 53 53
347	3382910	3385250	80 280 19 53 57	80 281 14 50 04
348	3385251	3387623	80 281 14 50 08	80 282 09 40 27
349	3387624	3389942	80 282 09 40 50	80 283 04 19 02
350	3389942	3392429	80 283 04 19 25	80 284 00 08 55
351	3392429	3395158	80 284 00 09 18	80 284 22 08 02
352	3395158	3397521	80 284 22 07 46	80 285 16 50 39
353	3397522	3399848	80 285 16 51 02	80 286 11 35 58
354	3399848	3402231	80 286 11 36 21	80 287 06 34 51
355	3402232	3404631	80 287 06 35 10	80 288 01 46 35
356	3404632	3406843	80 288 01 46 58	80 288 19 32 07
357	3406843	3408832	80 288 19 32 30	80 289 11 23 37
358	3408833	3411213	80 289 11 24 00	80 290 06 20 00
359	3411213	3413535	80 290 06 20 23	80 291 01 01 59
360	3413536	3415936	80 291 01 02 22	80 291 20 14 14
361	3415936	3418305	80 291 20 14 37	80 292 15 01 32
362	3418305	3420554	80 292 15 01 55	80 293 09 17 17
363	3420555	3422920	80 293 09 17 20	80 294 04 01 38
364	3422920	3425440	80 294 04 02 01	80 295 00 17 30
365	3425440	3427753	80 295 00 17 34	80 295 18 52 30

366	3427754	3430139	80 295 18 52 34	80 296 13 53 11
367	3430139	3432525	80 296 13 53 15	80 297 08 53 43
368	3432525	3434851	80 297 08 54 06	80 298 03 38 43
369	3434852	3437259	80 298 03 39 06	80 298 22 57 18
370	3437300	3439029	80 298 22 57 22	80 299 12 56 44
371	3439030	3441358	80 299 12 57 26	80 300 07 44 02
372	3441358	3443746	80 300 07 44 25	80 301 02 46 45
373	3443747	3446116	80 301 02 47 08	80 301 21 34 33
374	3446117	3448520	80 301 21 34 56	80 302 16 49 38
375	3448520	3450355	80 302 16 50 01	80 303 07 41 51
376	3450356	3452725	80 303 07 42 14	80 304 02 29 36
377	3452725	3455053	80 304 02 29 59	80 304 21 16 19
378	3455054	3457429	80 304 21 16 42	80 305 16 09 15
379	3457430	3459411	80 305 16 09 38	80 306 07 54 47
380	3459412	3461735	80 306 07 55 10	80 307 02 37 40
381	3461735	3463751	80 307 02 38 03	80 307 18 50 08
382	3463751	3465824	80 307 18 50 12	80 308 11 16 50
383	3465824	3468100	80 308 11 17 13	80 309 05 21 45
384	3468101	3470434	80 309 05 22 08	80 310 00 13 01
385	3470435	3472656	80 310 00 13 24	80 310 18 06 10
386	3472656	3475104	80 310 18 06 33	80 311 13 24 38
387	3475104	3478007	80 311 13 25 01	80 312 12 38 56
388	3478007	3481038	80 312 12 39 20	80 313 13 04 13
389	3481039	3483428	80 313 13 04 36	80 314 08 07 54
390	3483428	3485813	80 314 08 08 17	80 315 03 07 48
391	3485813	3488128	80 315 03 08 11	80 315 21 44 29
392	3488129	3490509	80 315 21 44 36	80 316 16 41 03
393	3490510	3492137	80 316 16 41 26	80 317 05 51 43
393.5	3492138	3493439	80 317 05 51 56	80 317 16 17 19
394	3493440	3494621	80 317 16 17 23	80 318 01 38 29
394.5	3494621	3495900	80 318 01 38 31	80 318 11 46 05
395	3494901	3498053	80 318 11 46 09	80 319 05 16 19
396	3498054	3500459	80 319 05 16 42	80 320 00 33 19
397	3500500	3502908	80 320 00 33 23	80 320 19 52 10
398	3502909	3505239	80 320 19 52 33	80 321 14 41 19
399	3505240	3507622	80 321 14 41 23	80 322 09 39 22
400	3507623	3509951	80 322 09 39 45	80 323 04 26 36
401	3509952	3512259	80 323 04 26 59	80 323 22 57 19
402	3512300	3514658	80 323 22 57 23	80 324 18 07 50
403	3514658	3517013	80 324 18 08 13	80 325 12 43 48
404	3517013	3519332	80 325 12 43 52	80 326 07 23 06
405	3519332	3521459	80 326 07 23 29	80 327 00 33 19
406	3521500	3523659	80 327 00 33 23	80 327 18 09 19
407	3523700	3525659	80 327 18 09 23	80 328 10 09 19
408	3525700	3528005	80 328 10 09 23	80 329 04 37 36
409	3528005	3530159	80 329 04 37 59	80 329 22 09 19
410	3530200	3532524	80 329 22 09 23	80 330 16 53 09
411	3532525	3534919	80 330 16 53 32	80 331 12 00 41
412	3534919	3537235	80 331 12 01 07	80 332 06 38 03
413	3537236	3539605	80 332 06 38 26	80 333 01 26 03
414	3539606	3541945	80 333 01 26 26	80 333 20 21 21
415	3541945	3544325	80 333 20 21 44	80 334 15 18 07
416	3544326	3546600	80 334 15 18 45	80 335 09 22 05

417	3546601	3548927	80 335 09 22 09	80 336 04 07 24
418	3548928	3551159	80 336 04 07 47	80 336 22 09 19
419	3551200	3553737	80 336 22 09 23	80 337 18 39 24
420	3553738	3556112	80 337 18 39 47	80 338 13 31 33
421	3556113	3558501	80 338 13 31 56	80 339 08 34 24
422	3558501	3560835	80 339 08 34 47	80 340 03 26 07
423	3560836	3563155	80 340 03 26 11	80 340 22 05 21
424	3563155	3565719	80 340 22 05 44	80 341 18 25 15
425	3565720	3568034	80 341 18 25 38	80 342 13 01 05
426	3568035	3570321	80 342 13 01 28	80 343 07 14 09
427	3570321	3572630	80 343 07 14 32	80 344 01 46 05
428	3572631	3575019	80 344 01 46 13	80 344 20 48 33
429	3575019	3577339	80 344 20 48 56	80 345 15 28 52
430	3577339	3579835	80 345 15 29 15	80 346 11 26 03
431	3579836	3582211	80 346 11 26 26	80 347 06 18 43
432	3582212	3584159	80 347 06 18 59	80 347 22 09 19
433	3584200	3586511	80 347 22 09 23	80 348 16 42 16
434	3586511	3588805	80 348 16 42 40	80 349 11 02 03
435	3588806	3591151	80 349 11 02 26	80 350 06 02 28
436	3591151	3593517	80 350 06 22 51	80 351 00 47 08
437	3593517	3597605	80 351 00 47 31	80 352 09 25 28
438	3597605	3600519	80 352 09 25 51	80 353 08 48 40
439	3600519	3602719	80 353 08 49 03	80 354 02 25 07
440	3602720	3655702	80 354 02 25 23	81 005 18 11 46
441	3655703	3659031	81 005 18 11 47	81 006 20 58 51
442	3659032	3677100	81 006 20 58 59	81 012 21 22 05
443	3677101	3680643	81 012 21 22 13	81 014 01 56 12
444	3680644	3683010	81 014 01 56 35	81 014 20 41 38
445	3683010	3725722	81 014 20 42 01	81 029 02 27 13
446	3725722	3745455	81 029 02 27 15	81 035 16 29 44
447	3745455	3747901	81 036 16 30 07	81 036 11 46 44
448	3747902	3763902	81 036 11 47 07	81 041 19 47 34
449	3763903	3766038	81 041 19 48 00	81 042 13 04 29
450	3766039	3770140	81 042 13 04 33	81 043 21 54 05
451	3770141	3823900	81 043 21 54 09	81 061 19 46 05
452	3823901	3829316	81 061 19 46 09	81 063 15 10 53
453	3829317	3846044	81 063 15 10 57	81 069 05 09 10
454	3846044	3887259	81 069 05 09 14	81 082 22 57 20
455	3887300	3977248	81 082 22 57 24	81 112 22 48 30
456	3977300	3998505	81 112 22 57 24	81 120 00 38 08
457	3998506	4040047	81 120 00 38 12	81 133 21 11 44
458	4040048	4042424	81 133 21 11 48	81 134 16 04 47
459	4042424	4045425	81 134 16 04 51	81 135 16 05 22
460	4045425	4082638	81 135 16 05 26	81 148 01 52 19
461	4082638	4220300	81 148 01 52 23	81 193 23 45 23
462	4220400	4250538	81 193 23 45 23	81 204 01 04 26
463	4250538	4446707	81 204 01 04 30	81 269 10 15 40
464	4446708	4451100	81 269 10 15 48	81 270 21 57 24
465	4451200	4523100	81 270 22 09 24	81 294 21 57 27
466	4523200	4600836	81 294 22 09 27	81 320 19 26 54
467	4600836	4603009	81 320 19 26 58	81 321 12 41 24
468	4603010	4623143	81 321 12 41 28	81 328 05 55 54
469	4623143	4625446	81 328 05 55 58	81 329 00 22 27

470	4625446	4751143	81 329 00 22 31	82 005 21 56 07
471	4751143	4797722	82 005 21 56 11	82 021 10 27 37
472	4797722	4834815	82 021 10 27 41	82 021 10 27 37
472	4797722	4834815	82 021 10 27 41	82 033 19 09 36
473	4834815	4979933	82 033 19 09 40	82 082 04 12 23
474	4979933	4986403	82 082 04 12 27	82 084 07 48 11
475	4986403	4991851	82 084 07 48 15	82 086 03 38 16
476	4991851	4998700	82 086 03 38 20	82 088 10 45 30
477	4998800	5005017	82 088 10 57 30	82 090 12 47 27
478	5005018	5026022	82 090 12 48 02	82 097 12 51 37
479	5026022	5107642	82 097 12 51 41	82 124 17 55 37
480	5107642	5175924	82 124 17 55 41	82 147 12 03 48
481	5175924	5305123	82 147 12 03 52	82 190 13 39 58
482	5305123	5323352	82 190 13 40 02	82 196 15 39 35
483	5323352	5326511	82 196 15 39 39	82 197 16 42 53
484	5326512	5389715	82 197 16 42 56	82 218 18 21 38
485	5389715	5471841	82 218 18 21 42	82 246 03 30 34
486	5471841	5476803	82 246 03 30 37	82 247 19 00 21
487	5476804	5483300	82 247 19 00 33	82 249 22 57 51
488	5483300	5489752	82 249 22 57 55	82 252 02 51 53
489	5489752	5510331	82 252 02 51 39	82 258 23 22 34
490	5510331	5521348	82 258 23 22 37	82 262 15 35 45
491	5521400	5599202	82 262 15 45 21	82 288 14 11 20
492	5599202	5692609	82 288 14 11 21	82 319 17 29 17

EARTH AND SPACE SCIENCES INSTITUTE
TELEPHONE: (602) 745-6767

April 27, 1981

Dr. Bob Vostrey
NSSDC
Code 601
Goddard Space Flight Center
Greenbelt, Maryland 20771

Dear Sir:

We are shipping separately the sample data tape which you requested by phone from Dr. Broadfoot. The tape contains 3216 spectra in GS-3 data mode (one spectrum every 3.84 seconds), from three scans of the Jupiter system. Enclosed is another copy of the description of our data tape format (which was included in the packet we mailed to you in July, 1980).

You will also find the following data products enclosed with the tape:

1) Two summary plots of the entire tape

   a) Program PSPEC sums 12 minutes of data, corrects for fixed pattern noise, and plots the resulting accumulated spectra. Four accumulated spectra are plotted per axis, successively offset vertically by ½ inch. (The first plot is at the bottom.) The labels give the FDSC, time, data mode, and pointing information for the first spectrum in the first accumulation. The Y-axis is in counts; the X-axis is labelled in both wavelengths and channel numbers. The asterisks on the right side mark accumulated spectra which include individual spectra with "bad" status words. (The status word is set to "bad" if there is not valid pointing information in the spectral record, or if any channel-to-channel count ratio is > 10:1. In GS-3 mode, the counts are usually so low that this test is not significant.)

   Note: If spectra are missing from any 12 minute interval, the sum is lower. No normalization is done, since the PSPEC plots are intended only as rough overviews of the data.

b) Program PSUM plots pointing, as well as the log of sums of counts for 12 minutes over specified channel ranges, vs FDS count and time. The sums are normalized to correct for missing spectra. There are four channel ranges plotted, as well as the azimuth and elevation of the scan platform, and limit cycle information. The X-axis is labelled in both FDS counts (FDSC) and spacecraft event times (SCET). The counts are plotted on a logarithmic scale, with 6 orders of magnitude per inch for counts from 0-1000. After 1000, the scale is 2 orders of magnitude per inch. Scan platform azimuth and elevation, along with limit cycle motion are plotted on the bottom two grids. The Y axis is labelled in degrees for scan platform azimuth and elevation. Limit cycle is plotted on a different scale, centered in the azimuth and elevation blocks. The scale for limit cycle is 1.33 degrees/inch. The asterisks along the bottom axis denote that at least one individual spectrum included in the data point has a "bad" status word (see explanation in (a), above).

PSUM's are used only as an overview of the motion of the scan platform and the rate and relative magnitude of changes in the counts. They are never used to obtain specific values.

PSPEC and PSUM plots are available in our data library for every 7-track data tape (HP-EDR) we generate.

2) HPLOG - This summary is also available for each of our HP-EDR's. It logs the JPL tape the data was generated from (JPL-EDR), the FDSC's and times, data mode, and number of spectra. A separate entry is made on the log whenever a new JPL-EDR starts, or when there is a change in data mode. HPLOG also notes the number of "zero" spectra encountered. (A record in which word number one is set to 0 to flag that the record should be ignored.)

3) Individual plots and printouts, produced by program VOY, of the first ten spectra on the tape. (Program VOY and VOY documentation were sent to you last July.) See VOY documentation for explanations of the header items. (Enclosed are two memos which describe new items in the headers.) Note that integer word #32, the HP-EDR spectrum number, should be ignored.

The counts in these particular ten GS-3 spectra are low, since the instrument is not pointed towards a bright source at the beginning of the scan.

4) A sample of the type of analysis we perform using Program VOY:

a) Printouts from positional bins performed on the spectra from the first scan. (See the VOY manual for a description of binning.)

b) Plots of the resultant binned spectra at the azimuth and elevation of the plasma torus.

I hope these items will be useful to you. If you require additional material please let me know.

Earlier, there was some discussion about our supplementary experiment data records (SEDR's). We use SEDR's only for absolute pointing information for certain special observations. For the majority of our analyses, the pointing information in the header for each spectrum is sufficient. We have a complete set of 7-track SEDR tapes generated from the 9 track tapes supplied by JPL. A dump from each 7-track SEDR of the most pertinent measurements is available in our data library. I am enclosing a sample printout. The frequency of the measurements varies, depending on the stage of the mission.

Please let me know if I can be of further assistance to you.

Sincerely,

Karen DeLay

Karen DeLay
Scientific Programmer

KD/meb

Enclosures

Item 5: Data Tape Format

The UVS data tapes are called Hewlett Packard Experiment Data Records (HPEDR's). They are 7-track, binary, written at 800 BPI. Each tape contains up to ∿ 16,000 spectral records, blocked 5 spectral records per physical tape record. They are produced on a Hewlett-Packard 21MX computer.

A spectral record is made up of 384 16-bit words. Some of the items are integers, some ASCII, and some are real. A real word contains 32 bits, i.e. it is equivalent to 2 integer words. The spectral records include words to be set by analysis programs (mainly program VOY), which log the manipulations that are performed on a spectrum during analysis. See Appendix A of VOY User's Guide for a complete description of a spectral record.

Tape Format Summary:

Type: 7-track, 800 BPI

Word Sizes: Integers-16 bits, Reals-32 bits

Physical Record: 1920 16 Bit Words = 5 logical records (spectral records)

Spectral Record: 384 Words

Header Information: 54 Words (34 Integer, 10 Real)

Engineering Data: 54 Words (27 Real)

Spectrum: 276 Words (12 shift locations---Real)

(126 data words---Real)

Number of spectral records per tape: Variable; up to ∿ 16,000

Number of files per tape: 1. End of file marks end of data on the tape.

Item 3A

PROGRAM VOY

USER'S GUIDE

D. R. MOSLEY

SEPTEMBER 1978

CAMPUS MEMO

TO Distribution

FROM Karen DeLay

DATE 22 October 1980

SUBJECT New Command in Program VOY

A command has been added to Program VOY to replace spectral data with the square root of the data, channel-by-channel.

SRT,N,M
N= STARTING LOCATION IN SPECTRAL DATA FILE
M= NUMBER OF SPECTRA

This command will replace the count in each channel with the square root of the count. The conversion will be performed on M spectra, starting with the spectrum in position N of the spectral data file. If a negative value is encountered, VOY informs the user, and sets the square root to zero for the channel.

Integer word number 34 in the spectral header (previously a spare) is incremented by 1 each time an "SRT" is performed on a spectrum. The "WRT" command has been modified to list this item under "SQRT". (A zero would indicate that no "SRT's" have been performed on the spectrum.)

Please update your VOY documentation to reflect these changes.

(Note: Unlike previously added commands, the addition of this command <u>will</u> <u>not</u> affect any programs stored in your base file.)

KITT PEAK NATIONAL OBSERVATORY

MEMORANDUM

TO Distribution Date 23 July 1979

FROM K. DeLay SUBJECT NEW COMMAND FOR PROGRAM VOY

A COMMAND HAS BEEN ADDED TO PROGRAM VOY TO REPLACE SPECTRAL DATA WITH THE LOGARITHM, BASE e, OF THE DATA.

CLN,N,M

N = STARTING LOCATION

M = NUMBER OF SPECTRA

THIS COMMAND WILL REPLACE THE COUNT IN EACH CHANNEL WITH THE NATURAL LOGARITHM OF THE COUNT. THE CONVERSION WILL BE PERFORMED ON M SPECTRA, STARTING WITH THE SPECTRUM IN POSITION N IN THE SPECTRAL DATA FILE.

INTEGER WORD NUMBER 20 IN THE SPECTRAL HEADER (PREVIOUSLY A SPARE) IS INCREMENTED BY 1 EACH TIME A "CLN" IS PERFORMED ON A SPECTRUM. THE WRT COMMAND HAS BEEN MODIFIED TO LIST THIS ITEM UNDER THE HEADING "LN." (A ZERO WOULD INDICATE THAT NO "CLN's" HAVE BEEN PERFORMED ON THE SPECTRUM.) Note: if the count in a channel is ≤0, we set that channel = -99.

PLEASE UPDATE YOUR VOY DOCUMENTATION TO REFLECT THESE CHANGES.

NOTE THAT THE ADDITION OF THIS COMMAND MAY NECESSITATE MODIFYING PROGRAMS IN YOUR BASE FILE. THE "IBN" COMMAND SHOULD BE THE ONLY COMMAND AFFECTED.

Distribution: Broadfoot, Ford, Forrester, Gordon, Kaiserman, McBreen, Sandel, Shemansky, Skirvin, Smith

INDEX

1.0 Introduction 1

1.1 Purpose 1
1.2 Execution 1
1.3 Program Disk Files 2
1.4 Program Modes 3

2.0 Data Storage 5

2.1 Magnetic Tape 5
2.2 Disk 6
2.3 Memory 7

3.0 Instructions 8

3.1 Data Transferring 8
3.1.01 Initialize Magnetic Tape 8
3.1.02 Data Quality Control 8
3.1.03 Tape to Disk Spectral Transfer 9
3.1.04 Inventory Primary Data File 9
3.1.05 Load Spectral Location 10
3.1.06 Store Spectral Location 10
3.1.07 Swap Spectral Locations 10
3.1.08 Write Tape Header Record 10
3.1.09 Write Spectrum 11
3.1.10 Print Flags 11
3.1.11 Read and Display Tape Header Record . . 11
3.1.12 Read and Store Spectral Record 12
3.1.13 Write End-of-File 12
3.1.14 Top-of-Form to Printer 12
3.1.15 Save Spectral Location 12
3.1.16 Purge Save Location 13
3.1.17 Recall Save Location 13
3.1.18 Create New Data File 13
3.1.19 Copy from Primary Data File to Specified File 14
3.1.20 Change Primary Data File 14

3.2 Manipulating Spectra 14
3.2.01 Add Spectra 14
3.2.02 Subtract Spectra 15
3.2.03 Multiply Spectra 15
3.2.04 Divide Spectra 15
3.2.05 Add Constant 15
3.2.06 Multiply by Constant 16
3.2.07 Smooth Spectrum by 1-2-1 Function . . . 16
3.2.08 Clear Spectral Location 16

3.2.09 Replace by Constant 16
3.2.10 Shift Spectrum 17
3.2.11 Set X-Fiducials 17
3.2.12 Integrate Between X-Fiducials 17
3.2.13 Mean Value Between X-Fiducials 17
3.2.14 Scale Spectrum N to Fit Spectrum M . . 18
3.2.15 Peak Count 18

3.3 Plotting 18
3.3.01 Set Mode to Point Plot 18
3.3.02 Set Mode to Line Plot 19
3.3.03 Set Vertical Scale 19
3.3.04 Set Horizontal Scale 19
3.3.05 Set Y-Origin 19
3.3.06 Set X-Origin 19
3.3.07 Set Vertical Height 20
3.3.08 Set Vertical Separation 20
3.3.09 List Plot Parameters 20
3.3.10 Plot Spectra 20
3.3.11 Plot Axes 21
3.3.12 Plot Offset 21
3.3.13 Plot Close 21
3.3.14 Enable Tektronix Cursor 22

3.4 Internal Programming 22
3.4.01 Interactive Mode 23
3.4.02 Programming Mode 23
3.4.03 Execution Mode 23
3.4.04 List Program Block 24
3.4.05 Input from ASCII File 24
3.4.06 Terminate ASCII File 25
3.4.07 Looping 25
3.4.08 Jump 25
3.4.09 Increment Integer Parameter 25
3.4.10 Increment Real Parameter 26
3.4.11 Reset Integer Parameter 26
3.4.12 Reset Real Parameter 26
3.4.13 Halt Execution 26
3.4.14 No-Op 27
3.4.15 Set Reference Value 27
3.4.16 Skip if Greater Than or Equal To . . . 27

3.5 Binning 28
3.5.01 Spacecraft ID Flag 28
3.5.02 Auto-Correlate Flag 28
3.5.03 Position Flag and Mode 29
3.5.04 Data Quality Flag 29
3.5.05 Bin Parameters 29
3.5.06 List Bin Parameters 30
3.5.07 Intensity Bin 30
3.5.08 Accumulate Spectra 31
3.5.09 Normalize Bin Spectra 31

3.6 Miscellaneous Instructions 32

3.6.01 Terminate Program VOY 32
3.6.02 List Instruction Log and Clear 32
3.6.03 Set Y-Fiducials 32
3.6.04 Edit Spectrum 32
3.6.05 Pile-Up Correction 33
3.6.06 Input FPN Spectrum 33
3.6.07 Input CAL Spectrum 33
3.6.08 Apply Scattering Matrix 34

Appendix A "UVS Data Record Format" A-1

Appendix B "Spectral Item List" B-1

1.0 INTRODUCTION

1.1 Purpose

Program VOY is the interactive spectral analysis program written to analyze the spectra from the UVS spectrometers on the Voyager spacecraft. The main functions of the program are as follows:

(1) Manipulating spectra—adding, subtracting, smoothing, adding constants, determining integrals, mean values, etc.

(2) Plotting spectra—determining mode, scale factors, origins, etc.

(3) Binning spectra—intensities or whole spectra.

(4) Editing spectra.

(5) Fixed pattern noise, scattering, and calibration.

Supporting these functions are various data transferring abilities and an internal programming mode.

1.2 Execution

Program VOY consists of a driver and 8 overlays. While the program is executing, it references two disk files: a program base file which contains various program values and arrays, and a spectral data file which contains the UVS spectra. To start execution of VOY, type:

RU,VOY

Note: The Multi-Terminal Monitor will create a copy of VOY which incorporates the logical unit of the user's console into the program name and runs that copy instead. The MTM, however, does not work on the system terminal.

Terminal	LU	Program Name
System Beehive	1	VOY
HP 2640A	7	VOY07
Tektronix	10	VOY10
Beehive	11	VOY11

1.3 Program Disk Files

Immediately upon execution, the program will inquire as to which files are to be used. The program will type on the user's console:

```
VOY PROGRAM BASE FILE?
                CARTRIDGE?
```

Respond with the name and cartridge number of the program base file which should be opened by the program. The user may enter any name of any file which contains a program base, or may utilize one of the default files the program can open. If the cartridge specified is 20, the default file name is VBASEn. If the cartridge is other than 20, the default file name is VBASE. The default cartridge number is 20. To choose any of the default values, enter a space/return in response to the question.

After specifying the program base file, the spectral data file is established. The program will type:

```
VOY SPECTRAL DATA FILE?
                CARTRIDGE?
```

Respond with the name and cartridge number of the spectral data file. Any file which contains VOY-formatted spectra may be specified, or the user may use one of the default files. If the cartridge specified is 20, the default file name is VDATAn. If the cartridge is other than 20, the default file name is VDATA. The default cartridge number is the number specified for the program base file.

If the default files on cartridge 20 are used, then a file access procedure is followed. The first copy of VOY executing will attach files VBASE1 and VDATA1. The second copy of VOY executing concurrently will attach files VBASE2 and VDATA2. And so on, with the third and subsequent copies of VOY.

If a user-specified file is inaccessible, the program will automatically fall back to the corresponding default file. In any case, the program will inform the user as to which files are being used.

The spectral data file which the program opened is called the primary data file. There are program commands available to create new data files, to copy spectra from the primary data file to another data file, and to change the primary data file.

1.4 Program Modes

The program operates in one of three different modes:

(1) Interactive—instructions are executed as they are entered.

(2) Programming—instructions are stored in the program base file as they are entered.

(3) Execution—instructions which were stored in the program base file are now executed.

Actions in the program are caused by entering a three-character instruction followed by up to three integer parameters, an integer and a real, or just a real. For example

CUM,1,10,20	Accumulate into spectral location 1, 20 spectra, starting from location 10.
STR,1,10	Store into spectral location 1 the spectrum from location 10.
EOF,8	Write an end-of-file mark on logical unit 8.
TOF	Write a top-of-form to the lineprinter.
ADC,1,7.5	Add to the spectrum in location 1 the constant 7.5.
YSC,80.0	Set Y-scale to 80.0 counts/inch.

Note that commas separate the parameters and the instruction.

2.0 DATA STORAGE

2.1 Magnetic Tape

JPL supplies 9-track EDR (Experiment Data Record) tapes in which the number of spectra per physical tape record varies according to the data mode. In addition the engineering record which corresponds to a given spectrum is some distance further down the tape.

The 9-track EDRs are converted into 7-track HPEDRs by Program JPLHP. In the translation process, the science records are broken up into individual spectra, given appropriate headers, and combined with the corresponding engineering data. The spectra are then reblocked into records of 5 spectra and written to the appropriate 7-track tape. The Voyager 1 data tapes are labeled 1xxxxx, and the Voyager 2 tapes are 2xxxxx.

Physical Record (7-track) = 1920 words
= 5 spectral records

Spectral Record = 384 words

header information = 54 words
34 integer words
10 real words

engineering data = 54 words
27 real words

spectrum = 276 words
12 shift locations (real)
126 data words (real)

Program VOY is capable of reading the 7-track HPEDRs and storing the spectra in the primary spectral data file.

The program is also capable of writing on tape two types of output records: a header record and a spectral record. The structure of an output tape is envisioned to be a header record followed by a number of spectral records and then an end-of-file mark. The header record is 600 words—large enough to hold the user's comment (20 lines at 60 characters/line) describing the file. The spectral records are unblocked spectra as described above. Each spectral record is 384 words. The program can also read the headers from tape and display them for the user, and can store the spectra in the primary spectral data file.

2.2 Disk

As mentioned in the introduction, the program accesses two disk files. The program base file is defined as follows:

Name: VBASE if LU ≠ 20
VBASEn if LU = 20, where n=1,2,3...

Security Code: 9850

Size: 98 ~~sectors~~

Contents: ~~Sectors~~ 1 - 10 System area
11 - 20 Program storage
21 - 22 LOG array
23 MISC array
24 - 38 Bin arrays
39 - 98 Save locations

The spectral data file is defined as follows:

Name: VDATA if LU ≠ 20
VDATAn if LU = 20, where n=1,2,3...

Security Code: 9850

Size: 3 sectors per spectrum.
Must be able to hold at least
4 spectra (12 ~~sectors~~).

Contents: UVS spectra

Note: The locations in the spectral data file are numbered from 1 to some maximum, depending on the size of the file. For example, a file 300 ~~sectors~~ long will have storage locations for 100 spectra. The locations are numbered 1, 2, 3, . . . 98, 99, 100.

The first four locations of the data file are special in that the four spectra contained in those locations are kept in memory. When the data file is opened, those locations are immediately read into memory. Execution time can be reduced substantially if as many of the manipulations as possible are done using the first four locations. When the data file is closed (and on certain other occasions), the four spectral locations on the disk are updated. Should the program abort, the contents of the disk will reflect the data from the last update and not the current data. (Hopefully, this will be a rare occurrence.)

During certain instructions, Program VOY also accesses three other disk files. These files are:

FPNSP	Fixed Pattern Noise Spectra
CALSP	Calibration Spectra
SMATRX	Scattering Matrices

2.3 Memory

Within memory, data and program values are stored in various arrays. Since the spectra and the instructions contain both integer and real values, most arrays are equivalenced between an integer type and a real type. (Note: Since real words are twice as long as integer words, the real arrays are dimensioned half of the integer arrays.) Some of these arrays are:

ISPEC	384 x 6	Integer	These two arrays contain the spectra from the first four disk locations, plus two scratch areas.
SPEC	192 x 6	Real	
LOG	64 x 4	Integer	These arrays contain a log of all instructions entered by the user.
RLOG	64 x 2	Real	
INST	32 x 4	Integer	These arrays contain the current instruction block.
RNST	32 x 2	Real	
MISC	128	Integer	This array contains various program values.
BIN	10 x 30	Real	This array contains the result from the intensity bin.

3.0 INSTRUCTIONS

3.1 DATA TRANSFERRING

The purpose of these instructions is to transfer spectra between magnetic tape and the primary data file, between locations within the data file, and to the user's console or line printer.

3.1.01 Initialize Magnetic Tape

INT,N,R

N = logical unit of tape drive (8 or 9)
R = desired position (relative or absolute)

The first parameter is the logical unit of the tape drive on which the HPEDR is mounted. It must always be specified. Various actions will occur depending on the sign/magnitude of R, the second parameter:

$R = -99$	Rewind the tape
$R < 0$	Backspace the tape R records
$0 \leq R < 100$	Forward space the tape R records
$R \geq 100$	Search for FDSC R

When forward spacing, the header of the first spectrum in each record is displayed on the user's console. After displaying the Rth record, the tape is backspaced to position it in front of that record. In other words, the last record displayed will be the record which will next be encountered. If R is zero, the current record will be displayed. If R is > 100, it is assumed to be an FDS count. The tape will be positioned to the first record containing that value.

3.1.02 Data Quality Control

DQC,R

R = data quality factor

When spectra are transferred from the HPEDR to the primary data file, the spectra are automatically scanned to determine if the ratio of counts in any two consecutive channels is greater than the factor R.

If this is the case, the spectral status bit for that spectrum is set. (Note: If the value for a particular channel is zero, it is set to 1 for the purposes of this determination.) If R is zero, the current value of the data quality factor is displayed. (Note: Program JPLHP automatically scans the spectra for channel-to-channel ratios greater than 10.0.)

3.1.03 Tape to Disk Spectral Transfer

TDK,N,M,L

N = number of spectra
M = initial location in primary data file
L = start flag

This instruction must be preceded by instruction INT to set the tape logical unit number. N spectra starting from the current tape position will be transferred to the primary data file starting at location M. L is a flag which will determine which of the five spectra in the initial tape record will begin the transfer. If L is zero, the first spectrum transferred will have the FDS count searched for by instruction INT. If L is nonzero, the first spectrum in the record will be the first spectrum transferred regardless of FDS count. Typing BR,VOY while this instruction is executing will abort the transfer.

3.1.04 Inventory Primary Data File

INV,N,M,L

N = initial location in primary data file
M = number of spectra
L = output logical unit number
(default = user's console)

A brief description of each spectrum starting at location N is listed on logical unit L. The default logical unit is the user's console.

3.1.05 Load Spectral Location

LDR,N,M

N = destination location
M = source location

The spectrum in disk location M is stored in location N.

3.1.06 Store Spectral Location

STR,N,M

N = source location
M = destination location

The spectrum in disk location N is stored in location M.

3.1.07 Swap Spectral Locations

SWP,N,M

N = location 1
M = location 2

The spectra in disk locations N and M are interchanged.

3.1.08 Write Tape Header Record

WRH,N,M

N = output logical unit
M = input logical unit
(default = user's console)

Accepts from logical unit M a comment of 20 lines with a maximum of 60 characters for each line. To terminate the comment, type /E on a new line. After the comment is entered, it is displayed for user verification. If verified, it is written to logical unit N (usually a tape drive). The header record should be written only after an end-of-file mark has been written.

TO WRITE LINES OF TEXT TO THE HEADER TAPE.

WRH,6

THIS INPUT IS TO BE 20 LINES WITH A MAXIMUM OF 60 CHARACTERS PER LINE. TYPE /E ON A NEW LINE TO END.

3.1.09 Write Spectrum

WRT,N,M

N = spectral location
M = output logical unit
(default = user's console)

Writes the spectrum from disk spectral location N to logical unit M. If M is a non-tape unit, the portion of the spectrum listed is controlled by the print flags. If M is a tape unit, the spectrum is written as a 384-word spectral record. Spectral records should be written to tape only after a header record has been written.

3.1.10 Print Flags

PRF,N

N = flag number

If N is zero, the status of the print flags is displayed on the user's console. The print flags are:

(1) Identification
(2) Characteristics
(3) History
(4) Pointing
(5) Spectrum

If N is non-zero, the status of the Nth flag is reversed, i.e., if it was on, it is turned off.

3.1.11 Read and Display Tape Header Record

RDH,N,M

N = logical unit of tape drive
M = relative file count

Reads the header record from the Mth file from the current tape position. If M equals zero, the header from the current file is read. The header record is then displayed on the user's console and the tape is left positioned behind the record.

3.1.12 Read and Store Spectral Record

RDT,N,M,L

N = spectral file location
M = logical unit of tape drive
L = record skip counter

L records from the tape are skipped and then the next record is read and stored in primary data file location N. If an EOF is encountered, the instruction aborts. Default values are M = tape drive specified by the RDH instruction, and L = zero.

3.1.13 Write End-of-File

EOF,N

N = logical unit of tape drive

Writes an end-of-file mark on the tape mounted on logical unit N.

3.1.14 Top-of-Form to Printer

TOF

Sends a top-of-form command to the line printer.

3.1.15 Save Spectral Location

SAV,N,M

N = spectral file location
M = save location number (1 - 20)

Stores the spectrum contained in primary data location N into save location M in the program base file. The save location must be purged before the transfer takes place.

3.1.16 Purge Save Location

PUR,N

N = save location number (1 - 20)

Clears save location N in the program base file.

3.1.17 Recall Save Location

RCL,N,M

N = spectral file location
M = save location number (1 - 20)

Stores the spectrum contained in save location M into primary data location N.

3.1.18 Create New Data File

CRT,N

N = number of spectra

Creates a spectral data file large enough to hold N spectra. Program VOY will inquire for the filename and the cartridge number. The filename must be unique on that particular cartridge or a filemanager error will occur. The default value for N = 100 spectra; the default for the cartridge number = 20. If the cartridge is 20, the default filename = VDATAn. If the cartridge is not 20, the default filename = VDATA. Note: The filename can be any legal filename and is not restricted to VDATA or VDATAn. This instruction does not change the primary data file.

3.1.19 Copy from Primary Data File to Specified File

COP,N,M,L

N = number of spectra
M = initial location in primary data file
L = initial location in named data file

Copies N spectra from the primary data file starting from location M to a user-specified data file starting at location L. The program will inquire for the filename and the cartridge number. The file must already exist. It will not be created by this instruction. The primary data file is unchanged.

3.1.20 Change Primary Data File

PDF

The current data file is closed and the program inquires for the name and cartridge of the next data file. The new data file must already exist.

3.2 MANIPULATING SPECTRA

The purpose of these instructions is to perform various manipulations on the spectra or between two spectra. Recall that the first four spectral locations are maintained in memory. Execution time can be reduced if those locations are utilized as much as possible.

3.2.01 Add Spectra

ADD,N,M

N = spectral location (result)
M = spectral location

The spectrum in location M is added to the spectrum in location N. The integration times are also summed.

3.2.02 Subtract Spectra

SUB,N,M

N = spectral location (result)
M = spectral location

The spectrum in location M is subtracted from the spectrum in location N.

3.2.03 Multiply Spectra

MUL,N,M

N = spectral location (result)
M = spectral location

The spectrum in location N is multiplied by the spectrum in location M. If the spectrum in M is either a FPN spectrum or a CAL spectrum, the appropriate bits are set in the status word and in the FPN/CAL ID words.

3.2.04 Divide Spectra

DIV,N,M

N = spectral location (result)
M = spectral location

The spectrum in location N is divided by the spectrum in location M. If a channel in the divisor is zero, the corresponding channel in the quotient is set to zero.

3.2.05 Add Constant

ADC,N,R

N = spectral location
R = constant value

The constant R is added to each channel in the spectrum at location N.

3.2.06 Multiply by Constant

MLC,N,R

N = spectral location
R = constant value

The constant R is used to multiply each channel in the spectrum at location N. The scale factor is also multiplied by R.

3.2.07 Smooth Spectrum by 1-2-1 Function

SMO,N,M

N = spectral location
M = repetition factor
(default = 1)

Smooths the spectrum contained in spectral location N, M times. The smoothing function is defined as follows:

SPEC(I) = [SPEC(I-1) + 2*SPEC(I) + SPEC(I+1)]/4

3.2.08 Clear Spectral Location

CLR,N,M

N = spectral location

The contents of spectral location N are completely zeroed, including the header and pointing.

3.2.09 Replace by Constant

RPC,N,R

N = spectral location
R = constant value

Every channel in the spectrum at location N is replaced by the constant R.

3.2.10 Shift Spectrum

SHF,N,M

N = spectral location
M = number of channels

The spectrum in location N is shifted by M channels. There are shift locations within the spectral array for shifting the spectrum 6 channels in each direction. Shifts greater than 6 are allowed; however, some data will be lost when it is shifted off the end.

3.2.11 Set X-Fiducials

FDX,N,M

N = fiducial number (1 or 2)
M = channel

Sets the appropriate fiducial (FX1 or FX2) to the value M. FX1 should be less than FX2. If N is zero, the current values of the fiducials are displayed on user's console.

3.2.12 Integrate Between X-Fiducials

IXX,N

N = spectral location

The channels between the x-fiducials are summed. The result is displayed on user's console and stored in the spectral header.

3.2.13 Mean Value Between X-Fiducials

MXX,N

N = spectral location

The mean value in the channels between the x-fiducials is determined. The result is displayed on the user's console and stored in the spectral header.

3.2.13a [illegible]

CLN, N, M

N = [illegible]

M = [illegible]

THE COUNT IN EACH CHANNEL IS REPLACED WITH THE NATURAL LOGARITHM OF THE [illegible] A COUNTER FOR THE [illegible] PERFORMED IS KEPT [illegible]

3.2.14 Scale Spectrum N to Fit Spectrum M

FIT,N,M

N = spectral location
M = spectral location

The spectrum in location N is scaled so that the mean value (between the x-fiducials) of spectrum N equals the mean value of spectrum M. The scaling factor is displayed on the user's console and stored in the spectral header.

3.2.15 Peak Count

PEK,N

N = spectral location

The peak count (largest count) is determined and displayed on user's console. It is also stored in the spectral header.

3.3 PLOTTING

The purpose of these instructions is to plot spectra on either the Gould plotter or the Tektronix terminal. The plotting surface on the Gould is 16.9 inches along the x-axis and 10.0 inches along the y-axis. There is adequate space to plot two pages of plots, with two sets of axes on each page. The plotting surface on the Tektronix is 10.0 inches by 10.0 inches. Only one page of plots will fit on the Tektronix screen.

Various instructions are available for setting scales, origins, etc. The plotting can be done in either a standard mode that does one page, or under direct user control as to placement of axes, number of spectra per plot, offsets, etc.

3.3.01 Set Mode to Point Plot

PNT

The plot mode is set so that a point plot will be made of the spectrum.

3.3.02 Set Mode to Line Plot

LNE

The plot mode is set so that a line plot will be made of the spectrum.

3.3.03 Set Vertical Scale

YSC,R

R = counts/inch

The scale for the y-axis is set to the value R.

3.3.04 Set Horizontal Scale

XSC,R

R = channels/inch

The scale for the x-axis is set to the value R.

3.3.05 Set Y-Origin

YOR,R

R = y-origin in counts

The origin at the y-axis is set to the value R.

3.3.06 Set X-Origin

XOR,N,M

N = lowest numbered channel to be plotted
M = highest numbered channel to be plotted

The origin of the x-axis is set to the Nth channel. If M is zero, the maximum extent of the plot is channel 126. If M is nonzero, the extent is channel M.

3.3.07 Set Vertical Height

YHT,R

R = inches

The height of both y-axes is set to the value R. If the top axis exceeds 10 inches, it is truncated.

3.3.08 Set Vertical Separation

YSP,R

R = inches

The separation between the two plots is set to the value R.

3.3.09 List Plot Parameters

LIP,N

N = logical unit number
(default = user's console)

The various parameters are displayed on logical unit N. The default logical unit is the user's console.

3.3.10 Plot Spectra

PLT,N,M,L

N = spectral location to be plotted on lower set of axes.
M = spectral location to be plotted on upper set of axes.
L = plot logical unit.

This instruction will plot two sets of axes on the same page (one above the other) according to the existing scale factors and lengths. On the lower set it will plot the spectrum in location N. On the upper set it will plot the spectrum in location M. If N, M and L are set to zero (default values), the previously

defined spectral locations are plotted on the previously defined logical unit. To suppress one of the plots, set either N or M to a negative number. The upper set of axes will not be plotted if it will overlap the lower set (although the spectrum will be plotted).

3.3.11 Plot Axes

PAX,N,M,L

N = axes set: 1 = lower, 2 = upper
M = page number 1 or 2
L = spacecraft ID: 1 = VOY1, 2 = VOY2

This instruction will plot the set of axes as defined by N and M on the previously defined logical unit. The parameter L is used to plot the wavelength scale since the two spacecraft have different offsets.

3.3.12 Plot Offset

POF,N,R

N = spectral location
R = offset in inches

The spectrum in location N is plotted on the last set of axes as defined by the PAX instruction. The value R is added to the y values before they are plotted.

3.3.13 Plot Close

PCL,N

N = plot logical unit

This instruction will close the plot file and cause the plot to be produced on logical unit N. If plotting to the Tektronix, do not use this instruction. If plotting to magnetic tape, program PASS4 will have to be run to produce the plot on the Gould.

3.3.14 Set Plot Logical Unit

PLU,N

N = PLOT LOGICAL UNIT

THE COMMAND ALLOWS YOU TO SET THE PLOT LOGICAL UNIT FOR THE PAX COMMAND.

3.3.14 Enable Tektronix Cursor

CUR

This instruction will cause the crosshairs to be displayed on the Tektronix screen. The user can position the cursor by use of the thumb wheels on the right of the keyboard. Positioning the cursor and striking a key can cause certain actions, depending upon which key was struck:

"1" Set fiducial FX1 to corresponding channel
"2" Set fiducial FX2 to corresponding channel

"L" Set fiducial FY1 to corresponding intensity
"U" Set fiducial FY2 to corresponding intensity

"W" Print wavelength
"I" Print intensity
"D" Print channel number

Any other nonblank character will disable the crosshairs and return to the program.

3.4 INTERNAL PROGRAMMING

The purpose of these instructions is to enable the user to store the instructions in blocks on the disk and then to execute them on user command. There are 10 blocks available numbered from 1 to 10. Each block is capable of holding 32 instructions numbered from 1 to 32. If an internal program exceeds 32 instructions, the next program block is allocated and the instructions are numbered from where they left off in the previous block. For example, suppose a program requires 76 instructions and it is stored beginning in block 1:

Block	Instructions
1	1 - 32
2	33 - 64
3	65 - 76

There are instructions available for skipping instructions, looping, incrementing parameters, and resetting parameters. Since these instructions actually modify other instructions, the program blocks would be modified. To prevent this, the program blocks are copied to a system area before execution of any internal program.

3.4.01 Interactive Mode

SNG

Sets the program mode to interactive. Used to terminate programming mode.

3.4.02 Programming Mode

PRG,N,M

N = block number
M = start location

Sets the program mode to internal programming. Instructions are stored in program block N starting at instruction M. No prompt is displayed while in programming mode. The instructions are echoed on the user's console as they are entered in the block. (Note: Instruction 1 is assumed to be block N, instruction 1. If instruction 35 is specified as the start location, then instructions are stored in block N + 1, instruction 3.)

3.4.03 Execution Mode

EXE,N,M,L

N = block number
M = start location
L = stop location (optional)

Sets the program mode to execute. The program blocks are copied to the system area. Execution will begin at instruction M of block N and will proceed sequentially (except for loop and jump instructions) until either (1) an instruction at a location greater than L is encountered; (2) a halt instruction is encountered; or (3) the user types

BR,VOY

on the console. In the last case, the program will display

***PROGRAM INTERRUPT.....BLOCK X INSTR XX

and set the program mode to interactive. To resume execution at the point of interruption, enter

EXE,0,0,0

Resumption is possible only if no other program was executed or internal programming performed.

3.4.04 List Program Block

LIS,N,M,L

N = block number
M = logical unit number
L = system flag

Lists program block N on logical unit M. If L is nonzero, the Nth system block is listed instead. Defaults are N, M and L all zero, which will list the current block in memory on the user's console.

3.4.05 Input from ASCII File

INP,filename

Changes the input logical unit from the user's console to the named file. The program mode is set to internal programming starting at program block 1, instruction 1. The instructions are read in from the file and stored in the program blocks. To change the destination block, use the PRG instruction within the file. The last instruction in the file should be an END.

3.4.06 Terminate ASCII File

END

Closes the ASCII file and resets the input logical unit to the user console and the programming mode to interactive. It must be the last instruction in the file.

3.4.07 Looping

LOP,N,M

N = repetition factor
M = instruction location

When the LOP instruction is encountered, an internal counter is incremented. If the counter is equal to N, control will pass to the instruction following the LOP instruction. Otherwise, control will pass to the instruction at location M. The default value for M is location 1.

3.4.08 Jump

JMP,N

N = number instructions

Control will pass to the Nth instruction from the JMP instruction. A JMP,0 will cause an infinite loop—always executing the same JMP instruction. N may be negative.

3.4.09 Increment Integer Parameter

INC,N,M,L

N = first parameter increment
M = second parameter increment
L = third parameter increment

The parameters of the previous instruction in the program block are incremented by the values N, M and L, respectively. The previous instruction must require integer parameters.

3.4.10 Increment Real Parameter

INR,N,R

N = first parameter increment
R = second parameter increment

The parameters of the previous instruction in the program block are incremented, by the values N and R, respectively. The previous instruction must require a real parameter.

3.4.11 Reset Integer Parameter

RST,N,M,L

N = instruction number
M = parameter number
L = new value

The Mth parameter of the Nth instruction in the program block is set to the value L. The instruction must require integer parameters.

3.4.12 Reset Real Parameter

RSR,N,R

N = instruction number
R = new value

The second parameter of the Nth instruction in the program block is set to the value R. The instruction must require a real parameter.

3.4.13 Halt Execution

HLT

When the halt instruction is encountered, the program mode is reset to interactive.

3.4.14 No-Op

NOP

A no-op instruction does nothing.

3.4.15 Set Reference Value

SET,N,R

N = reference parameter number (1 - 10)
R = new value

The Nth reference parameter is set to the value R. There are 10 reference parameters numbered from 1 to 10.

3.4.16 Skip if Greater Than or Equal To

SGE,N,M,L

N = spectral location
M = spectral item number
L = reference parameter number (1 - 10)

If the Mth item in the spectrum at spectral location N is greater than or equal to the Lth reference parameter, then the next instruction in the program blocked is skipped. A list of the items in the spectrum is contained in Appendix B.

3.5 BINNING

Binning is a process whereby spectra are accumulated (summed) according to what part of the sky was observed during the integration period of each spectrum. A two-dimensional array is defined. Each element of the array defines a positional "bin." If the airglow port of the UVS instrument is pointed within that "bin," then that spectrum is summed with other spectra whose pointing was similar. The results can be normalized after binning is completed.

Program VOY defines two types of binning—whole spectra and intensities. An intensity is defined to be the sum of specified channels within the spectrum. Since each spectrum yields only one intensity, it is possible to define an array to bin them. A whole-spectra bin is more cumbersome since each spectrum is 384 words long. Locations are set up in the primary data file to accumulate the spectra.

Various flags control which spectra are included in the bin. These flags must be set by the user.

3.5.01 Spacecraft ID Flag

SCF,N

N = value of flag

If N is zero, the spacecraft ID flag (ISCF) is cleared. During binning, spectra from both spacecraft are accepted. If N is 1 or 2, only spectra from the corresponding spacecraft are included in the bin.

3.5.02 Auto-Correlate Flag

ACF,N

N = value of flag

If N is zero, the flag is turned off. If N is non-zero, it is turned on. During whole-spectral bins, the spectra being accumulated are shifted before they are summed. The shifting is done to maximize the correlation of the spectra between the x-fiducials.

3.5.03 Position Flag and Mode

POS,N,M

N = value of flag
M = coordinate mode

If N is zero, the flag is cleared. If N is non-zero, the flag is set. During whole-spectral bins, if the flag is cleared, the position corresponding to a spectrum is ignored and the spectra are simply summed. If the flag is set, then the spectra are positionally binned. The coordinate system used is determined by M:

1 = L/W
2 = AZ/EL
3 = RA/DEC

3.5.04 Data Quality Flag

DQF,N,M

N = value of spectral flag
M = value of positional flag

The data quality flag has two parts—a spectral flag and a positional flag. If N is zero, the spectral flag is cleared. If N is non-zero, the spectral flag is set. Similarly for M and the positional flag. The data quality flag checks the status word in the spectral header. If the spectral bit is set and the spectral flag is on, then that spectrum is excluded from the bin. Similarly, if the position bit is set and the positional flag is on.

3.5.05 Bin Parameters

BIN,N

N = type of bin: 0 = intensity; 1 = spectral

This instruction will inquire from the user how to set up the bin array. It will inquire first for the minimums and then the increments. The first value corresponds to the L,AZ or RA direction. The second value corresponds to the W,EL or DEC direction. Then

the questions will depend upon which type of binning is chosen. For intensity bins the user will be asked about a dump of intermediate values; to define two regions within the spectrum; and to determine how the intensities should be calculated. For spectral bins the user will be asked to dimension the bin array and to specify where in the primary data file the array should start. Those spectral locations will then be cleared and made ready to accumulate spectra.

3.5.06 List Bin Parameters

LIB,N

N = logical unit number
(default = user's console)

The various bin parameters, flags, and values are displayed on logical unit N. The default logical unit is the user console.

3.5.07 Intensity Bin

IBN,N,M,L

N = number of spectra
M = initial location
L = mode

This instruction will do an intensity bin into a 10 x 30 array of N spectra starting at disk location M. The various parameters that have been set will control the binning. L determines what initialization and what output is done. If L is greater than or equal to zero, the bin arrays will be cleared before binning starts. If L is negative, the arrays are not cleared and more spectra are accumulated into them. If L is even, the normalized bin array is printed on the Gould. If L is odd, no output is done. Setting L odd, however, does not affect the dump of the intermediate values. Default for L is zero—initialize and output.

3.5.08 Accumulate Spectra

CUM,N,M,L

N = accumulate location or bin flag
M = initial location
L = number of spectra

If N is zero, binning will take place in the spectral array defined on the disk by the user. If N is non-zero, i.e. a spectral location, then accumulation will take place only in that one location. In either case, the binning is controlled by the position, spacecraft, data quality, and auto-correlation flags specified by the user.

3.5.09 Normalize Bin Spectra

NOR,N,M

N = accumulate location or bin flag
M = # TO NORMALIZE DEFAULT = L

If N is zero, all spectra in the spectral array defined on the disk by the user are normalized—that is, each spectrum is divided by the number of spectra accumulated. If N is non-zero, ~~the corresponding~~ M SPECTRA CONTAIN AT THE Nth spectral location ~~is~~ are normalized. In both cases, the header word containing the number binned is set to 1.

3.6 MISCELLANEOUS INSTRUCTIONS

3.6.01 Terminate Program VOY

STP

This instruction will terminate execution of program VOY. If the program base has been copied from the user's disk, it is copied back.

3.6.02 List Instruction Log and Clear

LOG,N

N = output logical unit

Lists on logical unit N the log which has been kept of all instructions entered by the user. After listing, the log is cleared. The default logical unit is the user's console.

3.6.03 Set Y-Fiducials

FDY,N,R

N = fiducial number (1 or 2)
R = value

Sets the corresponding fiducial (FY1 or FY2) to the value R. If N is zero, the current values of the fiducials are displayed.

3.6.04 Edit Spectrum

EDT,N,M

N = edit mode (1 or 2)
M = spectral location

The spectrum in location M is edited. Edit mode 1 allows the user to set the value of any channel in the spectrum. Edit mode 2 allows the user to set any value in the spectrum: header, position or channel. In this mode, the user must enter the item number and the new value. See Appendix B for the item list.

3.6.05 Pile-Up Correction

PUC,N,M

N = initial spectral location
M = final spectral location

The pile-up correction is applied to the spectra from location N to M. (If M is zero, only location N is corrected.) The pile-up correction is to correct for missed photons because of a fixed response time in the UVS instrument.

3.6.06 Input FPN Spectrum

FPN,N,M

N = spectral location
M = Fixed Pattern Noise location

The file FPNSP is accessed, and the Mth FPN spectrum is stored in the primary data file at location N. The FPN ID word is set to M and the spectrum is classed as a type 6 in the status word.

3.6.07 Input CAL Spectrum

CAL,N,M,L

N = spectral location
M = spacecraft ID
L = calibration factor

The file CALSP is accessed and the appropriate CAL spectrum is stored in location N. The CAL spectrum is stored in location N. The CAL spectrum is divided by the Lth calibration factor. The CAL ID word is set to 10*M+L and the spectrum is classed as a type 7. The calibration factors are:

(1) Extended Source ($R\ \mathring{A}^{-1}$)
(2) Extended Source ($Ph\ cm^{-2}\ s^{-1}\ \mathring{A}^{-1}\ ster^{-1}$)
(3) Point Source ($Ph\ cm^{-2}\ s^{-1}\ \mathring{A}^{-1}$)

3.6.08 Apply Scattering Matrix

SCT,N,M,L

N = spectral location
M = matrix number
L = spacecraft ID

The file SMATRX is accessed, and the scattering matrix defined by M and L is used to multiply the spectrum in location N. The scattering matrix ID word in the header is set to M and the appropriate bit in the spectral status word is set.

3.6.09 TAKE NATURAL LOGARITHM OF DATA

CLN,N,M

N = STARTING LOCATION
M = NUMBER OF SPECTRA

THIS COMMAND REPLACES THE COUNT IN EACH CHANNEL WITH THE NATURAL LOGARITHM OF THE COUNT. THE CONVERSION IS PERFORMED ON M SPECTRA, STARTING WITH THE SPECTRUM IN POSITION N IN THE SPECTRAL DATA FILE.

THE "LN" ENTRY IN THE SPECTRAL HEADER KEEPS TRACK OF THE NUMBER OF CLNS PERFORMED.

2.0 INSTRUCTION DECIPHERING

Whenever an overlay is entered, the value of IPARM(1) indicates whether or not an instruction is ready to be executed.

IPARM(1)	= 0	No instruction available
	≠ 0	Instruction ready for execution

If IPARM(1) = 0, the instruction routine, INSTR, is called to obtain the next instruction. If IPARM(1) ≠ 0, the overlay proceeds immediately with the execution of the instruction.

The instruction routine, INSTR, will get the next instruction from either the user's console (if program mode = 1) or the program base file (if program mode = 3). If the next instruction is from the program base file, it has already been deciphered and that section of code is skipped.

If the instruction comes from the user's console, it is read into a buffer (ABUF) under an A2 format. ABUF(1) will contain the first two characters, and ABUF(2) will contain the third character and usually a comma. The comma is immediately replaced by binary zeroes. INSTR maintains a list of all allowed instructions in the array INS, dimensioned 2 x 100. INS(1,I) contains the first two characters of the Ith instruction and INS(2,I) contains the third character and the type of the Ith instruction. A DO-LOOP compares ABUF(1) with INS (1,I) and ABUF(2) with INS(2,I) (top 8 bits only) until a match is found. The variable INST will then contain the instruction number of the instruction input by the user.

There are 4 types of instructions, depending upon the number and the type of parameters that accompany each instruction:

TYPE	PARAMETERS
1	Up to 3 integers
2	1 integer and 1 real
3	1 real
4	6 ASCII characters

The variable J is set to the type of instruction. This is done by ANDing out the type from the instruction list (INS(2,INST)) and converting to integer. Up to 7 types can be defined.

Now that the instruction number and type have been determined, the parameters are decoded. The integers are stored in variables I1, I2 and I3. The real is stored in variable R. (Note: variable R is equivalenced with I2 and I3.) The instruction and parameters are stored in the LOG array and then the appropriate action is taken.

Certain instructions (SNG, PRG, LIS, LOG, STP, INP and END) are executed immediately. These instructions cannot be stored in the program base file. (Note: EXE is handled uniquely. Study the code carefully.) If the instruction is not one of those, then the action is determined by the program mode.

If the program mode = 2 or 4, the instruction and parameters are stored in the array INSQ and eventually in the program base file. If the program mode = 1 or 3, the overlay containing the instruction is determined from the array IOVR and stored in IPARM(2). (Note: Overlay VOY0 has no instructions and therefore is not included in IOVR. The 8 zeroes in IOVR account for the instructions mentioned above.) Since within an overlay, instructions are numbered from 1, an offset must be subtracted from the instruction number depending upon which overlay the instruction is in. This offset is contained in the array IOFF. Now that the overlay and relative instruction number (JNST) have been determined, the subroutine INSTR returns to the overlay which called it.

Example:

Program Mode = 1 (Interactive)

User types: ADC,3,1.5

```
ABUF(1)   = 2HAD
ABUF(2)   = 2HC,

INST      = 47

J         = 2

I1        = 3
R         = 1.5

IPARM(2)  = IOVR(47) = 5
JNST      = 47 - IOFF(5)
          = 47 - 42
          = 5
```

Instruction ADC is the fifth instruction in overlay VOY5.

Suppose, however, that overlay VOY1 had called INSTR. It is necessary to change overlays. The value in OVRLAY(2) is set to the proper value (e.g. 2HY5) and the appropriate overlay is loaded via an EXEC call. IPARM(1) is set to a non-zero value to indicate that the instruction has already been deciphered.

3.0 INSTRUCTION EXECUTION

The following instructions are handled in a special way:

SNG	STP
PRG	INP
EXE	END
LIS	TDK
LOG	IBN

Study the code carefully to follow the sequence control for those instructions.

All other instructions follow the sequence control described below:

At statement label 10 a computed GOTO determines which instruction is executed next. Instruction labels begin at 200 and increment by 20.

Each instruction will initially determine the validity of its parameters. If the parameters are valid, the code is executed.

Upon normal completion, the instruction jumps to statement label 4 and calls INSTR to obtain the next instruction.

Upon abnormal termination, the instruction jumps to statement label 2 and prints an error message on the user console. The program mode and the input logical unit are reset. Control then passes to statement label 4. (If in VOY1 and a file manager error occurs, control will first pass to statement label 100.)

Upon returning from the INSTR subroutine, IPARM(2) is checked to determine if the instruction is in the current overlay. If so, execution continues. If not, the proper overlay is loaded.

4.0 INTERNAL PROGRAM BLOCKS

There are 10 program blocks contained in the program base file. Each is 128 words long and capable of holding up to 32 instructions. A scheme has been developed whereby it is possible to write internal programs of an arbitrary length up to a maximum of 320 instructions. This is possible even though only one block is kept in memory at one time.

The following variables are used to control internal programming:

IPC	Internal Program Counter—Indicates the number of the current instruction relative to a user-specified base. Possible values are 1 to 320.
IPS	Internal Program Stack—Indicates the number of the current stack in memory. Possible values are 1 to 20. (1-10 = system area; 11-20 = program area.)
IPCOF	Internal Program Counter Offset—Indicates the absolute location of the last instruction in the previous program block. Possible values are 0, 32, 64, 96, . . . , 288.
JPC	Relative Program Counter—Indicates the number of the current instruction within the current program block. (JPC = IPC - IPCOF.) Possible values are 1 to 32.
IPEND	Maximum Program Counter—Indicates the number of the last instruction in the program block. Usually set to 320.
INSQ	Program Block Array—Contains the current block from the program base file. Dimensioned 4 x 32.

The value of IPC which corresponds to location 1 (the base location) is determined by the user. IPC is not a totally absolute counter in that the base location is not always instruction 1 of block 1. The base location is set by the user to be in a specified block. The PRG and EXE instructions both establish the initial block. For example:

PRG,2,5	Base location is instruction 1 of block 2. First location entered will be instruction 5 of the same block.

PRG,4,35	Base location is instruction 1 of block 4. First location entered, however, will be instruction 3 of block 5 (the 35th location from the base location).
EXE,3,10,58	Base location is instruction 1 of block 3. First instruction executed will be instruction 10 of the same block. The last instruction is the 58th instruction from the base location (instruction 26 of block 4).

The current program block (that one indicated by IPS) is always in memory in the array INSQ. Whenever an EXE or PRG instruction is entered, the current block is written to the program base file, and the user-specified block brought into memory and then executed or programmed. That block will stay in memory until another block must be fetched. Then the current one is saved and the next block brought into memory.

Certain programming instructions (LOP, JMP, SGE) cause control to pass to certain locations. In each case, the location is determined and the appropriate program block is loaded. Other instructions (INC, INR, RST, RSR) actually modify instructions in the program block. The location is determined, the appropriate program block is loaded, the instruction modified, and the block written back to the program base file.

Since the programs would be affected by having their instructions modified, a system area was created. Whenever a program is to be executed, the entire program area (10 blocks) is copied to the system area. The program blocks are then loaded into memory from the system area. Therefore, when an instruction is being modified, it is located in the system area and not in the user area.

5.0 SUBROUTINES GET AND PUT

Many instructions access the spectra contained in the spectral data file. Before anything can be done with the spectra, however, they have to be brought into memory. The array ISPEC (384,6) is designed for two purposes: (1) to always maintain the spectra from the first 4 file locations in memory, and (2) to provide scratch areas to hold spectra while doing computations. Columns 1 to 4 in ISPEC contain the first 4 spectra and columns 5 and 6 are scratch areas.

To bring a spectrum into memory, subroutine GET has been provided. The call appears as follows:

```
CALL GET(ISOUR,IDEST,IERR,ISA)
```

where ISOUR = location of the spectrum (set by the user).

IDEST = location in memory of the spectrum (set by the subroutine).

IERR = File Manager error value (set by the subroutine).

ISA = scratch area in which spectrum is to be placed if not already in memory (set by the user).

Example:

Suppose the user types:

```
PEK,10
```

(determine the peak count of the spectrum in location 10.) The program will set the value of I1 to 10 and execute the code for the PEK instruction. In the instruction, a call to GET is done first:

```
CALL GET(I1,IT1,IERR,5)
```

which means fetch the spectrum specified by I1. If it is in memory, set IT1 to I1 and return. If it is not in memory, store the spectrum in the scratch area (column 5 of ISPEC) and set IT1 to 5. In this example, after the call, the spectrum in location 10 has been loaded into ISPEC column 5 and IT1 set to 5.

To write a spectrum into the primary data file, subroutine PUT has been provided. The call appears as follows:

```
CALL PUT(ISOUR,IDEST,IERR)
```

where ISOUR = column of ISPEC which contains the spectrum (set by the user).

IDEST = location in which spectrum is written (set by the user).

IERR = File Manager error value (set by the subroutine).

In the example of the PEK instruction, the following call is made at the end of the instruction:

```
CALL PUT(IT1,I1,IERR)
```

which means store the spectrum contained in column IT1 into location I1 (from column 5 to location 10).

6.0 COMMON

Each of the overlays uses the same common block. The following arrays are contained in common in the following order:

IPLOT(184)	Locations for the plot package. The first 40 locations hold various plot values. The next 144 contain the plotting Data Control Block.
IBASE(16)	Data Control Information for the program base file. The program base file must be opened first since a DCB is not provided.
IDATA(16)	Data Control Information for the primary data file.
ISPEC(384,6)	Spectral arrays.
INSQ(4,32)	Current program block.
LOG(4,64)	The LOG array.
MISC(128)	Various program values.
IPARM(5)	Parameter array for passing values between overlays.

Note that arrays ISPEC, INSQ and LOG are also equivalenced with real arrays SPEC, RNSQ, and RLOG.

7.0 PLOTTING

Included in overlay VOY4 are a number of routines from the plotting software. These routines are:

OPENG	Corrects a mistake which exists in the library version.
PUTGC	Modified so that certain parameters are stored in common. Otherwise, could not plot in a segmented program environment.
WINDO	Modifies the windowing so that it is automatically scaled.
CLOST	Modified so that the cursor is always homed when the Tektronix plot is closed.

APPENDIX A

UVS DATA RECORD FORMAT

1	SPECTRUM NUMBER	S/C IDENTIFICATION	1
3	STATUS	DATA MODE	2
5	SCET - YEAR	SCET - DAY	3
7	SCET - HOUR	SCET - MINUTE	4
9	SCET - SECOND	SCET - MILLISECOND	5
11	ERT - YEAR	ERT - DAY	6
13	ERT - HOUR	ERT - MINUTE	7
15	ERT - SECOND	ERT - MILLISECOND	8
17	FPN SPECTRUM NUMBER	CAL SPECTRUM NUMBER	9
19	SCAT MATRIX NUMBER	SPARE	10
21	COUNTING MODE	HIGH VOLTAGE LEVEL	11
23	NUMBER BINNED	NO. OF SMOOTHINGS	12
25	JPLHP VERSION	JPL-EDR TAPE LABEL	13
27	JPL-EDR TAPE LABEL	JPL-EDR TAPE LABEL	14
29	HP-EDR TAPE LABEL	HP-EDR TAPE LABEL	15
31	HP-EDR TAPE LABEL	HP-EDR SPECTRUM NO.	16
33	RA/DEC - AZ/EL FLAG	SPARE	17
35	FDS COUNT		18
37	INTEGRATION TIME		19
39	PEAK HEIGHT		20
41	INTEGRAL VALUE		21
43	MEAN VALUE		22
45	SCALE FACTOR		23
47	RA/AZ MINIMUM		24
49	RA/AZ MAXIMUM		25
51	DEC/EL MINIMUM		26
53	DEC/EL MAXIMUM		27
55	UVS HIGH VOLTAGE		28

57	UVS TEMPERATURE	29
59	AIR-GLOW AZIMUTH	30
61	AIR-GLOW ELEVATION	31
63	AIR-GLOW RIGHT ASCENSION	32
65	AIR-GLOW DECLINATION	33
67	OCCULTATION RIGHT ASCENSION	34
69	OCCULTATION DECLINATION	35
71	DELTA L 1	36
73	DELTA L 2	37
75	DELTA L 3	38
77	DELTA L 4	39
79	DELTA L 5	40
81	DELTA W 1	41
83	DELTA W 2	42
85	DELTA W 3	43
87	DELTA W 4	44
89	DELTA W 5	45
91	ROTATION MATRIX (A)	46
93	ROTATION MATRIX (B)	47
95	ROTATION MATRIX (C)	48
97	ROTATION MATRIX (D)	49
99	RIGHT ASCENSION SUN	50
101	DECLINATION SUN	51
103	PITCH BIAS ANGLE	52
105	YAW BIAS ANGLE	53
107	SPARE	54
109	SHIFT -6	55
111	SHIFT -5	56
113	SHIFT -4	57

115	SHIFT -3	58
117	SHIFT -2	59
119	SHIFT -1	60
121	UV DATA CHAN 1	61
123	UV DATA CHAN 2	62
	. . . . .	
369	UV DATA CHAN 125	185
371	UV DATA CHAN 126	186
373	SHIFT 1	187
375	SHIFT 2	188
377	SHIFT 3	189
379	SHIFT 4	190
381	SHIFT 5	191
383	SHIFT 6	192

SPECTRAL ITEM LIST

ITEM	WORD	DESCRIPTION
1	1	SPECTRUM NUMBER - Sequential count of the spectra contained in each FDS count.
2	2	SPACECRAFT IDENTIFICATION - Indicated the spacecraft which transmitted the data. 1 = Voyager 1 2 = Voyager 2
3	3	STATUS - Four octal digits: ABCD spectral flag A: 0 = Good 1 = Bad position flag B: 0 = Good 1 = Bad correction indicator C: 0 = No Correction 1 = FPN 2 = CAL 3 = CAL + FPN 4 = SCAT 5 = SCAT + FPN 6 = SCAT + CAL 7 = SCAT + CAL + FPN type indicator D: 0 = Filler 1 = Voyager Spectrum 2 = Artificial 3 = MJS Spectrum 4 = TBD 5 = TBD 6 = FPN 7 = CAL
4	4	DATA MODE - The telemetry format 1 = OC-1 5 = CR-3 2 = GS-3 6 = CR-4 3 = CR-1 7 = CR-5 4 = CR-2 8 = CR-6
		SPACECRAFT EVENT TIME - Greenwich Mean Time that corresponds to the spacecraft time of the first minor frame from which this spectrum was extracted.

ITEM	WORD	DESCRIPTION
5	5	YEAR
6	6	DAY
7	7	HOUR
8	8	MINUTE
9	9	SECOND
10	10	MILLISECOND
		EARTH RECEIVE TIME - Greenwich Mean Time of the reception of the first minor frame from which this spectrum was extracted.
11	11	YEAR
12	12	DAY
13	13	HOUR
14	14	MINUTE
15	15	SECOND
16	16	MILLISECOND
17	17	FPN SPECTRUM NUMBER - Number indicating which pattern noise spectrum was used on the spectrum.
18	18	CAL SPECTRUM NUMBER - Number of the calibration spectrum applied to this spectrum.
19	19	SCATTERING MATRIX NUMBER - Number of the scattering matrix applied to this spectrum.
20	20	~~SPARE~~ # OF TIMES CLN (NATURAL LOG) HAS BEEN PERFORMED.
21	21	COUNTING MODE - From the UVS command word. Indicates whether in the pulse height or pulse counting mode. 1 = Pulse mode 2 = Height mode
22	22	HIGH VOLTAGE LEVEL - From the UVS command word. Indicates the HV level: 0 - 7.
23	23	NUMBER BINNED - The number of spectra accumulated while binning.

ITEM	WORD	DESCRIPTION
24	24	NO. OF SMOOTHINGS - The number of times this spectrum was smoothed with a 1-2-1 function.
25	25	JPLHP VERSION - The version of program JPLHP which translated this spectrum.
26 27 28	26 27 28	JPLEDR TAPE - The VSN of the 9-track JPL-EDR which was the source of this spectrum. (Two chars/word)
29 30 31	29 30 31	HP-EDR TAPE - The VSN of the 7-track HP-EDR which contains this spectrum. (Two chars/word.)
32	32	HP-EDR SPECTRUM NO. - The number of this spectrum on the HP-EDR tape.
33	33	COORDINATE FLAG - Indicates which coordinate system was used during binning. 0 = Undef 1 = L/W 2 = AZ/El 3 = RA/DEC
34	34	SPARE
35	35-36	FDS COUNT - Conposed of: 100*(MOD 2**16 COUNT) + (MOD 60 COUNT) The MOD 2**16 count is incremented every 48 minutes and turns over after 5.981 years. The MOD 60 count is incremented every 48 seconds and turns over every 48 minutes.
36	37-38	INTEGRATION TIME - Time in seconds contained by the spectrum.
37	39-40	PEAK HEIGHT - The peak height of the spectrum in counts.
38	41-42	INTEGRAL VALUE - Integration of the spectrum between the X-fiducials.

ITEM	WORD	DESCRIPTION
39	43-44	MEAN VALUE - The mean value of the spectrum between the X-fiducials.
40	45-46	SCALE FACTOR - The factor used for scaling the spectrum.
41	47-48	RA/AZ MINIMUM - The minimum value in degrees of the bin interval in the RA, AZ, or L direction.
42	49-50	RA/AZ MAXIMUM - The maximum value in degrees of the bin interval in the RA,AZ, or L direction.
43	51-52	DEC/EL MINIMUM - The minimum value in degrees of the bin interval in the RA, AZ, or L direction.
44	53-54	DEC/EL MAXIMUM - The maximum value in degrees of the bin interval in the DEC, EL, or W direction.
45	55-56	UVS HIGH VOLTAGE.
46	57-58	UVS TEMPERATURE.
47	59-60	NOMINAL AZIMUTH OF THE AIR-GLOW PORT.
48	61-62	NOMINAL ELEVATION OF THE AIR-GLOW PORT.
49	63-64	NOMINAL RIGHT ASCENSION OF THE AIR-GLOW PORT.
50	65-66	NOMINAL DECLINATION OF THE AIR-GLOW PORT.
51	67-68	NOMINAL RIGHT ASCENSION OF THE OCCULATION PORT.
52	69-70	NOMINAL DECLINATION OF THE OCCULATION PORT.
53	71-72	DELTA L 1 - Correction to the pointing due to the limit cycle along the length of the slit. The initial correction corresponds to the beginning of the integration period.

ITEM	WORD	DESCRIPTION
54	73-74	DELTA L 2 - Second length correction occurs at 25% through the integration period.
55	75-76	DELTA L 3 - Third length correction occurs at 50% through the integration period.
56	77-78	DELTA L 4 - Fourth length correction occurs at 75% through the integration period.
57	79-80	DELTA L 5 - Fifth length correction occurs at the end of the integration period.
58	81-82	DELTA W 1 - Correction to the pointing due to the limit cycle along the width of the slit. Timing of the width corrections is the same as for the lenth corrections.
59	83-84	DELTA W 2 - Second width correction.
60	85-86	DELTA W 3 - Third width correction.
61	87-88	DELTA W 4 - Fourth width correction.
62	89-90	DELTA W 5 - Fifth width correction.
		TRANSLATION MATRIX - Translation from Delta L and W to Delta RA and DEC is of the form: $\begin{bmatrix} DRA \\ DDE \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} DL \\ DW \end{bmatrix}$
63	91-92	MATRIX COEFFICIENT A
64	93-94	MATRIX COEFFICIENT B
65	95-96	MATRIX COEFFICIENT C
66	97-98	MATRIX COEFFICIENT D
67	99-100	RIGHT ASCENSION OF SUN AT TIME OF SPECTRUM

ITEM	WORD	DESCRIPTION
68	101-102	DECLINATION OF SUN AT TIME OF SPECTRUM
69	103-104	PITCH BIAS ANGLE OF SPACECRAFT AT TIME OF SPECTRUM,
70	105-106	YAW BIAS ANGLE OF SPACECRAFT AT TIME OF SPECTRUM.
71	107-108	SPARE
		SHIFT WORDS - Locations for shifting the spectrum in wavelenght.
72	109-110	SHIFT -6
73	111-112	SHIFT -5
74	113-114	SHIFT -4
75	115-115	SHIFT -3
76	117-118	SHIFT -2
77	119-120	SHIFT -1
78	121-122	UV DATA CHANNEL 1
79	123-124	UV DATA CHANNEL 2
80	125-126	UV DATA CHANNEL 3
201	367-368	UV DATA CHANNEL 124
202	369-370	UV DATA CHANNEL 125
203	371-372	UV DATA CHANNEL 126
204	373-374	SHIFT 1
205	375-376	SHIFT 2
206	377-378	SHIFT 3
207	379-380	SHIFT 4
208	381-382	SHIFT 5
209	383-384	SHIFT 6

SEDR DUMP

SEDR TAPE: SUV15202 NAVIGATION RECORD TYPE: SATURN YEAR: 1980 DAY: 301 OUTPUT FROM SEDR7: 18 JANUARY 1981 13:18:27

S/C EVENT TIME DDD HH MM SS FFF	FDSC [16 /60]	SATURN R.A. (DEG)	SATURN DEC. (DEG)	SUN R.A. (DEG)	SUN DEC. (DEG)	HGA R.A. (DEG)	HGA DEC. (DEG)	A.G. BORESIGHT AZ. (DEG)	A.G. BORESIGHT EL. (DEG)	A.G. BORESIGHT R.A. (DEG)	A.G. BORESIGHT DEC. (DEG)	OCC. BORESIGHT R.A. (DEG)	OCC. BORESIGHT DEC. (DEG)
301: 0: 0:34.748	34434./19	193.99	-2.59	1.57	-1.81	4.09	-.32	249.60	174.85	193.85	-1.36	206.30	-16.53
301: 0: 1:22.748	34434./20	193.99	-2.59	1.57	-1.81	4.09	-.29	249.60	174.61	194.01	-1.58	206.48	-16.74
301: 0: 2:10.748	34434./21	193.99	-2.59	1.57	-1.81	4.09	-.30	249.60	174.37	194.15	-1.76	206.64	-16.91
301: 0: 2:58.748	34434./22	193.99	-2.59	1.57	-1.81	4.09	-.31	249.60	174.12	194.31	-1.94	206.81	-17.09
301: 0: 3:46.748	34434./23	193.99	-2.59	1.57	-1.81	4.09	-.32	249.60	173.88	194.45	-2.12	206.96	-17.27
301: 0: 4:34.748	34434./24	193.99	-2.59	1.57	-1.81	4.09	-.32	249.60	173.63	194.60	-2.31	207.13	-17.45
301: 0: 5:22.748	34434./25	193.99	-2.59	1.57	-1.81	4.09	-.33	249.60	173.39	194.75	-2.49	207.29	-17.63
301: 0: 6:10.748	34434./26	193.99	-2.59	1.57	-1.81	4.08	-.34	249.60	173.14	194.90	-2.68	207.46	-17.82
301: 0: 6:58.748	34434./27	193.99	-2.59	1.57	-1.81	4.08	-.35	249.60	172.91	195.04	-2.85	207.61	-17.98
301: 0: 7:46.748	34434./28	193.99	-2.59	1.57	-1.81	4.08	-.35	249.60	172.67	195.19	-3.04	207.78	-18.16
301: 0: 8:34.748	34434./29	193.99	-2.59	1.57	-1.81	4.08	-.35	249.60	172.48	195.31	-3.19	207.90	-18.30
301: 0: 9:22.748	34434./30	193.99	-2.59	1.57	-1.81	4.08	-.36	249.60	172.43	195.33	-3.22	207.93	-18.33
301: 0:10:10.748	34434./31	193.99	-2.59	1.57	-1.81	4.07	-.35	252.44	172.43	195.03	-3.45	206.88	-19.15
301: 0:10:58.748	34434./32	193.99	-2.59	1.57	-1.81	4.16	-.38	252.44	172.43	195.10	-3.45	206.91	-19.18
301: 0:11:46.748	34434./33	193.99	-2.59	1.57	-1.81	4.09	-.36	252.44	172.43	195.04	-3.45	206.87	-19.16
301: 0:12:34.748	34434./34	193.99	-2.59	1.57	-1.81	4.07	-.36	252.44	172.74	194.83	-3.15	206.65	-18.86
301: 0:13:22.748	34434./35	193.99	-2.59	1.57	-1.81	4.07	-.36	252.44	172.92	194.72	-3.01	206.54	-18.72
301: 0:14:10.748	34434./36	193.99	-2.59	1.57	-1.81	4.07	-.36	252.44	173.21	194.55	-2.76	206.35	-18.49
301: 0:14:58.748	34434./37	193.99	-2.59	1.57	-1.81	4.07	-.36	252.44	173.47	194.40	-2.55	206.18	-18.28
301: 0:15:46.748	34434./38	193.99	-2.59	1.57	-1.81	4.07	-.37	252.44	173.73	194.25	-2.34	206.01	-18.08
301: 0:16:34.748	34434./39	193.99	-2.59	1.57	-1.81	4.06	-.37	252.44	173.96	194.11	-2.15	205.86	-17.89
301: 0:17:22.747	34434./40	193.99	-2.59	1.57	-1.81	4.06	-.37	252.44	174.19	193.98	-1.95	205.71	-17.70
301: 0:18:10.747	34434./41	193.99	-2.59	1.57	-1.81	4.06	-.37	252.44	174.45	193.83	-1.74	205.55	-17.49
301: 0:18:58.747	34434./42	193.99	-2.59	1.57	-1.81	4.06	-.38	252.44	174.70	193.68	-1.54	205.39	-17.29
301: 0:19:46.747	34434./43	193.99	-2.59	1.57	-1.81	4.06	-.38	252.44	174.95	193.53	-1.33	205.22	-17.09
301: 0:20:34.747	34434./44	193.99	-2.59	1.57	-1.81	4.05	-.38	252.44	174.99	193.51	-1.29	205.20	-17.05
301: 0:21:22.747	34434./45	193.99	-2.59	1.57	-1.81	4.06	-.37	255.27	174.99	193.30	-1.44	204.20	-17.75
301: 0:22:10.747	34434./46	193.99	-2.59	1.57	-1.81	4.03	-.46	255.27	174.99	193.28	-1.34	204.18	-17.64
301: 0:22:58.747	34434./47	193.99	-2.59	1.57	-1.81	4.05	-.39	255.27	174.99	193.30	-1.43	204.18	-17.74
301: 0:23:46.747	34434./48	193.99	-2.59	1.57	-1.81	4.05	-.38	255.26	174.69	193.48	-1.74	204.38	-18.06
301: 0:24:34.747	34434./49	193.99	-2.59	1.57	-1.81	4.05	-.38	255.26	174.43	193.62	-1.95	204.52	-18.27
301: 0:25:22.747	34434./50	193.99	-2.59	1.57	-1.81	4.06	-.36	255.26	174.21	193.75	-2.16	204.67	-18.47
301: 0:26:10.747	34434./51	193.99	-2.59	1.57	-1.81	4.07	-.33	255.26	173.95	193.89	-2.41	204.84	-18.71
301: 0:26:58.747	34434./52	193.99	-2.59	1.57	-1.81	4.07	-.31	255.26	173.69	194.04	-2.64	205.01	-18.93
301: 0:27:46.747	34434./53	193.99	-2.59	1.57	-1.81	4.08	-.29	255.26	173.46	194.17	-2.85	205.16	-19.14
301: 0:28:34.747	34434./54	193.99	-2.59	1.57	-1.81	4.08	-.29	255.26	173.20	194.31	-3.07	205.31	-19.35
301: 0:29:22.747	34434./55	193.99	-2.59	1.57	-1.81	4.08	-.30	255.26	172.98	194.43	-3.25	205.45	-19.53
301: 0:30:10.747	34434./56	193.99	-2.59	1.57	-1.81	4.08	-.32	255.26	172.74	194.55	-3.43	205.59	-19.70
301: 0:30:58.747	34434./57	193.99	-2.59	1.57	-1.81	4.07	-.33	255.26	172.50	194.68	-3.63	205.73	-19.89
301: 0:31:46.747	34434./58	193.99	-2.59	1.57	-1.81	4.07	-.34	255.26	172.43	194.72	-3.67	205.78	-19.93
301: 0:32:34.747	34434./59	193.99	-2.59	1.57	-1.81	4.07	-.34	256.93	172.43	194.53	-3.78	205.11	-20.35
301: 0:33:22.747	34435./ 0	193.99	-2.59	1.57	-1.81	4.07	-.34	258.09	172.43	194.40	-3.86	204.64	-20.63
301: 0:34:10.747	34435./ 1	193.99	-2.59	1.57	-1.81	4.07	-.35	258.09	172.43	194.40	-3.85	204.64	-20.62
301: 0:34:58.747	34435./ 2	193.99	-2.59	1.57	-1.81	4.07	-.36	258.09	172.43	194.40	-3.84	204.63	-20.62
301: 0:35:46.747	34435./ 3	193.99	-2.59	1.57	-1.81	4.11	-.35	258.09	172.43	194.44	-3.86	204.66	-20.64

Item 3C


```
C
C...PROGRAM VOY   LAST EDIT:  8 AUGUST 1978
C
FTN4
      PROGRAM VOY(3),  VERSION 080878 DRM
      DIMENSION ICOMN(3040)
      INTEGER OVRLAY(3)
      COMMON ICOMN
      DATA OVRLAY/2HVO,2HY0,2H  /
      CALL RMPAR(ICOMN(3033))
      CALL EXEC(8,OVRLAY)
      END
```

```
C
C...SUBROUTINE INSTR   LAST EDIT: 19 JUNE 1980
C
FTN4
      SUBROUTINE INSTR,  VERSION 061980 KKD
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
     +          LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10)
      LOGICAL ITTY,IFTTY,LOGOUT
      EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
     +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
      EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
     +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
     +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
     +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
     +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
     +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
     +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
     +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
     +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
     +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
     +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
     +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
     +            (MISC(37),IFX1),   (MISC(38),IFX2),   (MISC(39),IFX3),
     +            (MISC(40),IFX4),   (MISC(41),ISCF),   (MISC(42),IACF),
     +            (MISC(43),IPOSF),  (MISC(44),IDQF),   (MISC(45),MCORD),
     +            (MISC(46),IDUMP),  (MISC(47),IBIN),   (MISC(48),LBIN),
     +            (MISC(49),NROW),   (MISC(50),ITYPE),
     +                               (MISC(56),IT5),    (MISC(57),IT4),
     +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
      EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
     +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
     +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
     +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
     +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
     +            (MISC(89),RMIN1),  (MISC(91),RMIN2),  (MISC(93),RINC1),
     +            (MISC(95),RINC2),  (MISC(97),RA0),    (MISC(99),DEC0),
     +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
     +            (MISC(107),RT2),   (MISC(109),REF)
      INTEGER OVRLAY(3),ABUF(15),ISMP(3)
      DIMENSION INS(2,100),IOVR(100),IOFF(10),NBUF(15),INSPOL(16)
      EQUIVALENCE (ABUF(3),NBUF)
      DATA OVRLAY/2HVO,2HY0,2H  /
      DATA LUPRNT/6/
      DATA NINS/84/
      DATA NBUF(14),NBUF(15)/2*2H.,/
      DATA INSPOL/7*0,24B,223B,7*0/
      DATA ISMP/2HSM,2HP ,2H  /
      DATA IBEL/3400B/
      DATA INS/2HSN,2HG1,2HPR,2HG1,2HEX,2HE1,2HLI,2HS1,2HLO,2HG1,
     +         2HST,2HP1,2HIN,2HP4,2HEN,2HD1,2HIN,2HT2,2HDQ,2HC3,
     +         2HLD,2HR1,2HCU,2HM1,2HST,2HR1,2HSW,2HP1,2HWR,2HH1,
     +         2HWR,2HT1,2HRD,2HH1,2HRD,2HT1,2HEO,2HF1,2HTO,2HF1,
```

```
     +          2HSA,2HV1,2HPU,2HR1,2HRC,2HL1,2HIN,2HV1,2HPR,2HF1,
     +          2HTD,2HK1,2HPN,2HT1,2HLN,2HE1,2HYS,2HP3,2HYH,2HT3,
     +          2HYS,2HC3,2HXS,2HC3,2HYO,2HR3,2HXO,2HR1,2HFD,2HX1,
     +          2HFD,2HY2,2HLI,2HP1,2HPL,2HT1,2HPO,2HF2,2HPA,2HX1,
     +          2HPC,2HL1,2HCU,2HR1,2HPL,2HU1,2HAD,2HD1,2HSU,2HB1,
     +          2HMU,2HL1,2HDI,2HV1,2HAD,2HC2,2HML,2HC2,2HSM,2HO1,
     +          2HCL,2HR1,2HRP,2HC2,2HSH,2HF1,2HIX,2HX1,2HMX,2HX1,
     +          2HFI,2HT1,2HPE,2HK1,2HSC,2HF1,2HAC,2HF1,2HPO,2HS1,
     +          2HDQ,2HF1,2HBI,2HN1,2HNO,2HR1,2HLI,2HB1,2HSC,2HT1,
     +          2HLO,2HP1,2HJM,2HP1,2HIN,2HC1,2HIN,2HR2,2HRS,2HT1,
     +          2HRS,2HR2,2HHL,2HT1,2HNO,2HP1,2HPU,2HC1,2HED,2HT1,
     +          2HFP,2HN1,2HCA,2HL1,2HSE,2HT2,2HSG,2HE1,2HCR,2HT1,
     +          2HCO,2HP1,2HPD,2HF1,2HCL,2HN1,2HIB,2HN1/
      DATA IOVR/8*0,17*1,2,17*3,22*5,18*6,7/
      DATA IOFF/8,25,26,43,43,65,83,84/
      IF (MODE.EQ.3) GO TO 80
C
C...READ NEXT INSTRUCTION
C
    2 MODE = 1
      LUIN = LUUC
    4 WRITE(LUUC,6) IBEL
    6 FORMAT(A2,"VOY>_")
    8 READ(LUIN,10) ABUF
   10 FORMAT(15A2)
C
C...DECIPHER INSTRUCTION AND DECODE PARAMETERS
C
      ABUF(2) = IAND(ABUF(2),177400B)
      DO 14 INST = 1,NINS
      IF (ABUF(1).NE.INS(1,INST)) GO TO 14
      IF (ABUF(2).EQ.IAND(INS(2,INST),177400B)) GO TO 18
   14 CONTINUE
      WRITE(LUUC,16) IBEL,ABUF(1),ABUF(2)
   16 FORMAT(4X,2A2,A1,"?")
      GO TO (4,8,2,2), MODE
   18 J = IAND(INS(2,INST),7B)
      I1 = 0
      I2 = 0
      I3 = 0
      GO TO (20,22,24,26), J
   20 CALL CODE
      READ(NBUF,*) I1,I2,I3
      GO TO 30
   22 CALL CODE
      READ(NBUF,*) I1,R
      GO TO 30
   24 CALL CODE
      READ(NBUF,*) R
      GO TO 30
   26 I1 = NBUF(1)
      I2 = NBUF(2)
      I3 = NBUF(3)
C
```

```
C...ENTER INSTRUCTION INTO LOG
C
   30 ILC = ILC + 1
      IF (ILC.LT.1 .OR. ILC.GT.64) GO TO 34
      LOG(1,ILC) = INST
      LOG(2,ILC) = I1
      LOG(3,ILC) = I2
      LOG(4,ILC) = I3
      IF (ILC.LT.58) GO TO 40
   34 WRITE(LUUC,35) IBEL,ILC
   35 FORMAT(A2,"LOG =",I3)
C
C...DETERMINE APPROPRIATE ACTION
C
   40 GO TO (2,100,120,140,160,180,200,220,42), INST
   42 GO TO (50,60,80,60), MODE
C
C...   (1) EXECUTE INSTRUCTION
C
   50 IPARM(2) = IOVR(INST)
      JNST = INST - IOFF(IPARM(2))
      RETURN
C
C...   (2) ENTER INTO INSTRUCTION STACK
C
   60 IPC = IPC + 1
      JPC = IPC - IPCOF
      IF (JPC.LE.32) GO TO 62
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      IPS = IPS + 1
      IPCOF = IPCOF + 32
      JPC = IPC - IPCOF
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
   62 INSQ(1,JPC) = INST
      INSQ(2,JPC) = I1
      INSQ(3,JPC) = I2
      INSQ(4,JPC) = I3
      GO TO (64,66,66,64), J
   64 WRITE(LUOUT,65) IPC,INS(1,INST),INS(2,INST),I1,I2,I3
   65 FORMAT(40X,I3,2X,A2,A1,3I6)
      GO TO 68
   66 WRITE(LUOUT,67) IPC,INS(1,INST),INS(2,INST),I1,R
   67 FORMAT(40X,I3,2X,A2,A1,I6,1X,F11.3)
   68 IF (IPC.LT.IPEND) GO TO 8
      GO TO (2,2,2,220), MODE
C
C...   (3) EXECUTE STORED INSTRUCTIONS
C
   80 IF (IFBRK(IT1)) 82,86
   82 WRITE(LUUC,84) IBEL,IPS,IPC
   84 FORMAT(A2,"***PROGRAM INTERRUPT....BLOCK",I3,"  INSTR",I4)
      GO TO 2
   86 IPC = IPC + 1
      IF (IPC.GT.IPEND) GO TO 2
```

```
      JPC = IPC - IPCOF
      IF (JPC.LE.32) GO TO 88
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      IPS = IPS + 1
      IPCOF = IPCOF + 32
      JPC = IPC - IPCOF
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
   88 INST = INSQ(1,JPC)
      IF (INST.LT.1 .OR. INST.GT.NINS) INST = 72
      I1 = INSQ(2,JPC)
      I2 = INSQ(3,JPC)
      I3 = INSQ(4,JPC)
      GO TO 50
C
C...PRG, BLOCK#, STARTLOC
C
  100 IF (I1.LT.1 .OR. I1.GT.10) GO TO 300
      IF (I2.LT.1 .OR. I2.GT.320) GO TO 300
      IT1 = I1 + 10 + (I2-1)/32
      IF (IT1.LT.11 .OR. IT1.GT.20) GO TO 300
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      IPS = IT1
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
      IPC = I2 - 1
      IPCOF = 32*(IPC/32)
      IPEND = 320 - 32*(I1-1)
      IF (MODE.EQ.1) MODE = 2
      IF (MODE.EQ.2) WRITE(LUOUT,102) I1,I2
  102 FORMAT(40X,"PROGRAM BLOCK",I3,"  INSTR",I4)
      GO TO 8
C
C...EXE, BLOCK#, STARTLOC, STOPLOC
C
  120 IF (MODE.NE.1 .AND. MODE.NE.3) GO TO 60
      MODE = 3
      IF (I1.EQ.0 .AND. IPS.LE.10) GO TO 80
      IF (I1.LT.1 .OR.I1.GT.10) GO TO 300
      IF (I2.LT.1 .OR. I2.GT.320) GO TO 300
      IF (I3.NE.0 .AND. (I3.LT.I2 .OR. I3.GT.320)) GO TO 300
      IT1 = I1 + (I2-1)/32
      IF (IT1.LT.1 .OR. IT1.GT.10) GO TO 300
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      DO 122 I = 1,10,5
      CALL READF(IBASE,IERR,IBUF,640,LEN,I+10)
  122 CALL WRITF(IBASE,IERR,IBUF,640,I)
      IPS = IT1
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
      IPC = I2 - 1
      IPCOF = 32*(IPC/32)
      IPEND = I3
      IF (IPEND.EQ.0) IPEND = 320 - 32*(I1-1)
      GO TO 80
C
C...LIS, BLOCK#, LU, FLAG
```

```
C
  140 IF (I1.LT.0 .OR. I1.GT.10) GO TO 300
      IF (I2.EQ.0) I2 = LUOUT
      ITTY = IFTTY(I2)
      IF(.NOT. ITTY .AND. I2 .NE. LUPRNT) GO TO 2
      IF (I3.NE.0) I3 = 1
      IF (I1.EQ.0) GO TO 142
      IT1 = I1 + 10*(1-I3)
      IF (IT1.NE.IPS) GO TO 146
  142 DO 144 I = 1,128
  144 IBUF(I) = INSQ(I)
      IT1 = IPS
      GO TO 148
  146 CALL READF(IBASE,IERR,IBUF,128,LEN,IT1)
  148 IT2 = IT1
      IF (IT2.GT.10) IT2 = IT2 - 10
      IF (IT1.EQ.IT2) WRITE(I2,149) IT2
      IF (IT1.NE.IT2) WRITE(I2,150) IT2
  149 FORMAT(16X,"SYSTEM BLOCK",I3/)
  150 FORMAT(/16X,"PROGRAM BLOCK",I3/)
      IT1 = 16
      IT2 = 32
      LOGOUT = .TRUE.
      GO TO 166
C
C...LOG, LU
C
  160 I2 = I1
      LOGOUT = .TRUE.
      IF (I2.EQ.0) I2 = LUOUT
      IF(I2 .NE. LUPRNT .AND. I2 .NE. LUUC .AND. I2 .NE. 1)
     +  LOGOUT = .FALSE.
      DO 162 I = 1,256
  162 IBUF(I) = LOG(I)
      IT2 = ILC
      ILC = 0
      IF (IT2.GT.64) IT2 = 64
      IT1 = (IT2+1)/2
      IF (I2.EQ.LUPRNT) WRITE(LUPRNT,164)
  164 FORMAT(1H1//)
  166 IF (I2.EQ.LUPRNT) IT1 = IT2
      DO 178 I = 1,IT1
      IT3 = I
  168 IT4 = 4*(IT3-1) + 1
      IT5 = IBUF(IT4)
      IF (IT5.LT.1 .OR. IT5.GT.NINS) IT5 = 72
      J = IAND(INS(2,IT5),7B)
      GO TO (170,172,172,174), J
  170 IF(LOGOUT)WRITE(I2,171) IT3,INS(1,IT5),INS(2,IT5),
     +  (IBUF(J),J=IT4+1,IT4+3)
  171 FORMAT(5X,I2,2X,A2,A1,3I6,"  _")
      GO TO 176
  172 J = (IT4+3)/2
      IF(LOGOUT)WRITE(I2,173) IT3,INS(1,IT5),INS(2,IT5),
```

```
     +  IBUF(IT4+1),BUF(J)
  173 FORMAT(5X,I2,2X,A2,A1,I6,1X,F11.3,"  _")
      GO TO 176
  174 IF(LOGOUT)WRITE(I2,175) IT3,INS(1,IT5),INS(2,IT5),
     +  (IBUF(J),J=IT4+1,IT4+3)
  175 FORMAT(5X,I2,2X,A2,A1,2X,3A2,10X,"  _")
  176 IT3 = IT3 + IT1
      IF (IT3.LE.IT2) GO TO 168
      IF (I2.NE.LUPRNT.AND. LOGOUT) WRITE(I2,177)
  177 FORMAT(1H )
  178 CONTINUE
      GO TO (4,8,80,8), MODE
C
C...STP
C
  180 CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      MODE = 1
      IOPEN = 0
      IPARM(1) = 99
      CALL EXEC(8,OVRLAY)
C
C...INP, FILENAME
C
  200 CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      CALL GETFN(NBUF,30,INSPOL(3),INSPOL(6),INSPOL(7))
      CALL SPOPN(INSPOL,LUIN)
      IF (LUIN.GT.0) GO TO 210
      WRITE(LUUC,205) I1,I2,I3
  205 FORMAT(2X,3A2," SPOOLER ERROR")
      GO TO 300
  210 MODE = 4
      IPS = 11
      IPC = 0
      IPCOF = 0
      IPEND = 320
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
      GO TO 8
C
C...END
C
  220 IF (LUIN.GT.13) CALL EXEC(23,ISMP,4,LUIN)
      GO TO 2
C
C...ERROR MESSAGE
C
  300 WRITE(LUUC,310) IBEL
  310 FORMAT(A2,"ILLEGAL PARAMETER(S)")
      GO TO 2
      END
```

```
C
C...SUBROUTINE GETFN
C
      SUBROUTINE GETFN(LINE,LEN,NAME,ISC,ICR)
      DOUBLE PRECISION NAME,FNAME
      INTEGER IBUF(33)
      EQUIVALENCE (IBUF(2),FNAME),(IBUF(6),ISEC),(IBUF(10),ICRNO)
      DO 10 I = 1,LEN
      CALL SGET(LINE,I,K)
   10 IF (K.EQ.72B) CALL SPUT(LINE,I,54B)
      CALL PARSE(LINE,LEN,IBUF)
      NAME = FNAME
      ISC = ISEC
      ICR = ICRNO
      RETURN
      END
```

```
0336        END$
```

```
C
C...PROGRAM VOY0   LAST EDIT: 7 DECEMBER 1979 KKD
C
FTN4
      PROGRAM VOY0(5),  VERSION 120779 KKD
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
     +          LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10)
      EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
     +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
      INTEGER OVRLAY(3),VBASE(3),UBASE(3),UDATA(3)
      DATA OVRLAY/2HVO,2HY1,2H  /
      DATA VBASE/2HVB,2HAS,2HE0/
      DATA ICRB,ICRD/2*0/
      IF (IPARM(1).EQ.99) GO TO 100
C
C...INITIALIZE LOGICAL UNITS
C
      LUUC = IPARM(1)
      IF (LUUC.LE.0) LUUC = 1
      IVDATA(1) = 2HVD
      IVDATA(2) = 2HAT
      IVDATA(3) = 2HA0
      WRITE(LUUC,5)
    5 FORMAT(/"PROGRAM VOY --- VOYAGER SPECTRAL ANALYSIS PROGRAM"
     +       /"VERSION 3.2 --- RELEASED:  SEPT 1978")
      WRITE(LUUC,10)
   10 FORMAT(/"VOY PROGRAM BASE FILE? _")
      READ(LUUC,15) UBASE
   15 FORMAT(3A2)
      WRITE(LUUC,20)
   20 FORMAT(12X,"CARTRIDGE? _")
      READ(LUUC,*) ICRB
      WRITE(LUUC,25)
   25 FORMAT(/"VOY SPECTRAL DATA FILE? _")
      READ(LUUC,15) UDATA
      WRITE(LUUC,30)
   30 FORMAT(13X,"CARTRIDGE? _")
      READ(LUUC,*) ICRD
      IF (ICRB.EQ.0) ICRB = 20
      IF (ICRD.EQ.0) ICRD = ICRB
      IF (UBASE(1).NE.2H  ) GO TO 55
      IF (ICRB.NE.20) GO TO 50
C
C...OPEN PROGRAM BASE FILE
C
   40 VBASE(3) = VBASE(3) + 1
      IVDATA(3) = IVDATA(3) + 1
      CALL OPEN(IBASE,IERR,VBASE,0,9850,ICRB)
      IF (IERR.EQ.(-8)) GO TO 40
      IF (IERR.EQ.(-6)) CALL CREAT(IBASE,IERR,VBASE,98,1,9850,ICRB)
      IF (IERR.GE.0) GO TO 60
      WRITE(LUUC,45) VBASE,ICRB,IERR
```

```
   45 FORMAT(/2X,3A2," CARTRIDGE",I4," FILE MANAGER ERROR",I4)
      STOP
   50 UBASE(1) = 2HVB
      UBASE(2) = 2HAS
      UBASE(3) = 2HE
   55 CALL OPEN(IBASE,IERR,UBASE,0,9850,ICRB)
      IF (IERR.GE.0) GO TO 60
      WRITE(LUUC,45) UBASE,ICRB,IERR
      ICRB = 20
      UBASE(1) = 2H
      GO TO 40
C
C...OPEN SPECTRAL DATA FILE
C
   60 IF (UDATA(1).NE.2H  ) GO TO 75
      IF (ICRD.NE.20) GO TO 70
   62 IF (IVDATA(3).EQ.2HA0) IVDATA(3) = 2HA1
   64 CALL OPEN(IDATA,IERR,IVDATA,0,9850,ICRD)
      IF (IERR.EQ.(-8)) GO TO 66
      IF (IERR.EQ.(-6)) CALL CREAT(IDATA,IERR,IVDATA,300,1,9850,ICRD)
      IF (IERR.GE.0) GO TO 80
      WRITE(LUUC,45) IVDATA,ICRD,IERR
      CALL CLUSE(IBASE)
      STOP
   66 IVDATA(3) = IVDATA(3) + 1
      GO TO 64
   70 UDATA(1) = 2HVD
      UDATA(2) = 2HAT
      UDATA(3) = 2HA
   75 CALL OPEN(IDATA,IERR,UDATA,0,9850,ICRD)
      IF (IERR.GE.0) GO TO 80
      WRITE(LUUC,45) UDATA,ICRD,IERR
      ICRD = 20
      UDATA(1) = 2H
      GO TO 62
C
C...DECLARE FILES TO USER
C
80    CONTINUE
      IF (UBASE(1).EQ.2H  ) WRITE(LUUC,84) VBASE,ICRB
      IF (UBASE(1).NE.2H  ) WRITE(LUUC,84) UBASE,ICRB
      IF(UDATA(1) .EQ. 2H  ) GO TO 82
      DO 81 I = 1,3
81    IVDATA(I) = UDATA(I)
82    WRITE(LUUC,86)IVDATA,ICRD
   84 FORMAT(/"VOYAGER PROG BASE: FILE ",3A2," CARTRIDGE",I4)
   86 FORMAT("VOYAGER SPEC DATA: FILE ",3A2," CARTRIDGE",I4)
C
C...LOAD ARRAYS FROM PROG BASE
C
      CALL READF(IBASE,IERR,LOG,256,LEN,21)
      CALL READF(IBASE,IERR,MISC,128,LEN,23)
      CALL RWNDF(IDATA)
      DO 90 I = 1,4
```

```
0109  90     CALL READF(IDATA,IERR,ISPEC(1,I),384,LEN)
0110         IPS = MISC(8)
0111         CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
0112         MISC(12) = LUUC
0113         MISC(13) = LUUC
0114         MISC(14) = LUUC
0115         MISC(17) = ICRB
0116         MISC(18) = ICRD
0117         MISC(19) = VBASE(3)
0118         IF (MISC(19).EQ.2HE0) MISC(19) = 2HE1
0119         MISC(20) = IVDATA(3)
0120         IF (MISC(20).EQ.2HA0) MISC(20) = 2HA1
0121         IPARM(1) = 0
0122         CALL EXEC(8,OVRLAY)
0123         CALL VOY
0124  C
0125  C...TERMINATE PROGRAM
0126  C
0127    100  CALL WRITF(IDATA,IERR,ISPEC,1536,1)
0128         CALL WRITF(IBASE,IERR,MISC,128,23)
0129         CALL WRITF(IBASE,IERR,LOG,256,21)
0130         CALL CLOSE(IDATA)
0131         CALL CLOSE(IBASE)
0132         STOP
0133         END
```

```
0001  C
0002  C...PROGRAM VOY1   LAST EDIT: 6 DECEMBER 1979  K. DE LAY
0003  C
0004  FTN4
0005        PROGRAM VOY1(5),  VERSION 120679 KKD
0006        DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
0007       +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
0008       +        LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10),ITIME(14)
0009        EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
0010       +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
0011        COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
0012        EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
0013       +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
0014       +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
0015       +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
0016       +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
0017       +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
0018       +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
0019       +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
0020       +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
0021       +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
0022       +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
0023       +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
0024       +            (MISC(37),IFX1),   (MISC(38),IFX2),   (MISC(39),IFX3),
0025       +            (MISC(40),IFX4),   (MISC(41),ISCF),   (MISC(42),IACF),
0026       +            (MISC(43),IPOSF),  (MISC(44),IDQF),   (MISC(45),MCORD),
0027       +            (MISC(46),IDUMP),  (MISC(47),IBIN),   (MISC(48),LBIN),
0028       +            (MISC(49),NROW),   (MISC(50),ITYPE),
0029       +                               (MISC(56),IT5),    (MISC(57),IT4),
0030       +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
0031        EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
0032       +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
0033       +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
0034       +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
0035       +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
0036       +            (MISC(89),RMIN1),  (MISC(91),RMIN2),  (MISC(93),RINC1),
0037       +            (MISC(95),RINC2),  (MISC(97),RA0),    (MISC(99),DEC0),
0038       +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
0039       +            (MISC(107),RT2),   (MISC(109),REF)
0040        LOGICAL TSTAT,IFBRK,ITTY,IFTTY
0041        DIMENSION FPREC(8),IMASK(10)
0042        INTEGER OVRLAY(3),VOYOV(9)
0043        EQUIVALENCE(REG,IREG)
0044        DATA FPREC/0.0,0.0,1.0,4.0,4.0,19.0,59.0,114.0/
0045        DATA IMASK/77770B,77707B,77077B,70777B,07777B,
0046       +          00001B,00010B,00100B,01000B,10000B/
0047        DATA OVRLAY/2HVO,2H  ,2H  /
0048        DATA VOYOV/2HY1,2HY2,2HY3,2HY4,2HY5,2HY6,2HY7,2HY8,2HY9/
0049        DATA LUPRNT,LUTEK,LUTP1,LUTP2/6,10,8,9/
0050        DATA IBEL/3400B/
0051  C
0052  C...INITIALIZATION
0053  C
0054        IF (IPARM(1).NE.0) 10,4
```

```
2       IF (MODE.NE.1) WRITE(LUUC,3) IPS,JPC,IPC
3       FORMAT(" BLOCK NO.",I3,"   LOCATION",I3,"  ABS LOC",I4)
        MODE = 1
        LUIN = LUUC
        WRITE(LUUC,32) IBEL
32      FORMAT(A2," ILLEGAL PARAMETERS")
4       REG = LURQ(140000B,ITEMP,1)
        CALL INSTR
10      IF (IPARM(2).EQ.1) GO TO (200,220,240,260,280,300,320,340,
       +                          360,380,400,420,440,460,480,500,
       +                          520), JNST
12      OVRLAY(2) = VOYOV(IPARM(2))
        IPARM(1) = 1
        CALL EXEC(8,OVRLAY)
        CALL VOY
C
C...FILE MANAGER ERROR
C
100     WRITE(LUUC,102) IBEL,IERR
102     FORMAT(A2," FILE MANAGER ERROR",I5)
        GO TO 2
C
C...INT,N,R       INTIALIZE TAPE
C
200     IF (I1.NE.LUTP1 .AND. I1.NE.LUTP2) GO TO 2
        LUTP = I1
        REG = LURQ(140001B,LUTP,1)
        IF(IREG .NE. 0) GO TO 2
        ICONW = 100B + LUTP
        IF (R.LT.0.0) GO TO 214
        IF (R.GT.99.0) GO TO 204
        IT1 = IFIX(R)
C
C...CHECK FOR OLD FORMAT BY LOOKING AT BUF(18)
C
        DO 201 I = 1,IT1
        CALL EXEC(1,ICONW,IBUF,38)
        IFORM = 18
        IF(BUF(18) .EQ. 0.0) IFORM = 12
        IF (TSTAT(1)) GO TO 2
201     WRITE(LUOUT,202) BUF(IFORM),IBUF(1),IBUF(2),(IBUF(J),J=5,16),
       +                 IBUF(4),BUF(IFORM+1)
202     FORMAT(/" FDSC ",F7.0,"   SPEC ",I3,"   VOYAGER",I2,
       +       /" SCET ",I2,"/",2I3,":",I2,":",I2,".",I3,
       +       /" ERT  ",I2,"/",2I3,":",I2,":",I2,".",I3,
       +       /" DATA MODE"I3,"   DURATION ",F6.2," SECONDS")
        BACKSPACE LUTP
        FDSC = BUF(IFORM)
        GO TO 4
204     CALL EXEC(1,ICONW,IBUF,38)
        IFORM = 18
        IF(BUF(18) .EQ. 0.0) IFORM = 12
        IF (TSTAT(1)) GO TO 2
        IT4 = 1
```

```
205   IT1 = IBUF(4)
      IF (IT1.LT.1) IT1 = 2
      RT1 = R - FPREC(IT1)
      IF (AMOD(RT1,100.0).GT.59.0) RT1 = RT1 - 40.0
206   IF (IFBRK(IT2)) WRITE(LUUC,202) BUF(IFORM),IBUF(1),IBUF(2),
     +                (IBUF(J),J=5,16),IBUF(4),BUF(IFORM+1)
      IF (RT1.GT.BUF(IFORM)) GO TO 209
      IF (R.LT.BUF(IFORM)) GO TO 210
      IF (IT1.GT.2) GO TO 208
      IF (IBUF(1).EQ.1) GO TO 208
207   BACKSPACE LUTP
      BACKSPACE LUTP
      CALL EXEC(1,ICONW,IBUF,38)
      IFORM = 18
      IF(BUF(18) .EQ. 0.0) IFORM = 12
208   BACKSPACE LUTP
      WRITE(LUOUT,202) BUF(IFORM),IBUF(1),IBUF(2),(IBUF(J),J=5,16),
     +                 IBUF(4),BUF(IFORM+1)
      FDSC = R
      GO TO 4
209   CALL EXEC(1,ICONW,IBUF,38)
      IFORM = 18
      IF(BUF(18) .EQ. 0.0) IFORM = 12
      IF (TSTAT(1)) GO TO 2
      IT4 = 2
      IF (IBUF(4).EQ.IT1) 206,205
210   IF (IT4.EQ.1) GO TO 212
      WRITE(LUOUT,211) R
211   FORMAT(" FDSC ",F7.0," NOT ON TAPE. PREVIOUS RECORD:")
      GO TO 207
212   DO 213 I= 1,21
      BACKSPACE LUTP
      IF (TSTAT(2)) GO TO 209
213   CONTINUE
      GO TO 204
214   IT1 = IFIX(-R)
      IF (IT1.EQ.99) GO TO 217
      DO 215 I = 1,IT1
      BACKSPACE LUTP
      IF (TSTAT(2)) GO TO 216
215   CONTINUE
216   CALL EXEC(1,ICONW,IBUF,38)
      IFORM = 18
      IF(BUF(18) .EQ. 0.0) IFORM = 12
      WRITE (LUOUT,202) BUF(IFORM),IBUF(1),IBUF(2),(IBUF(J),J=5,16),
     +                  IBUF(4),BUF(IFORM+1)
      BACKSPACE LUTP
      FDSC = BUF(IFORM)
      GO TO 4
217   REWIND LUTP
      GO TO 4
C
C...DQC,R       DATA QUALITY CONTROL
C
```

```
220   IF (R.LT.0.0) GO TO 2
      IF (R.GT.0.0) DQF = R
      IF (R.EQ.0.0) WRITE(LUOUT,222) DQF
222   FORMAT(" DATA QUALITY FACTOR =",F7.3)
      GO TO 4
C
C...LDR,N,M     LOAD SPECTRUM
C
240   IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I2,I2,IERR,5)
      IF (IERR.LT.0) GO TO 100
      CALL PUT(I2,I1,IERR)
      IF (IERR.LT.0) GO TO 100
      IF (IBIN.LE.I1 .AND. I1.LE.LBIN) LBIN = 0
      GO TO 4
C
C...CUM,N,M,L   ACCUMULATE SPECTRA
C
260   IF (I1.LT.0) GO TO 2
      IF (I2.LT.1) GO TO 2
      IF (I3.LT.1) GO TO 2
      IF (I1.EQ.0 .AND. (IPOSF.EQ.0 .OR. LBIN.EQ.0)) GO TO 2
      IF (I1.NE.0) CALL GET(I1,IT2,IERR,6)
      I3 = I2 + I3 - 1
      DO 278 I = I2,I3
      CALL GET(I,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 100
      IF (ISCF.NE.0 .AND. ISPEC(2,IT1).NE.ISCF) GO TO 278
      ISTAT = IAND(ISPEC(3,IT1),7577B)
      IF (IAND(ISTAT,IDQF).NE.0) GO TO 278
      IF (IPOSF.EQ.0) GO TO 268
      GO TO (262,263,264), MCORD
262   RT2 = SPEC(46,IT1)*SPEC(49,IT1) - SPEC(47,IT1)*SPEC(48,IT1)
      IF (RT2.EQ.0.0) GO TO 278
      DRA = RA0 - SPEC(32,IT1)
      DDE = DEC0 - SPEC(33,IT1)
      RT1 = SPEC(38,IT1) - (SPEC(49,IT1)*DRA - SPEC(47,IT1)*DDE)/RT2
      RT2 = SPEC(43,IT1) - (SPEC(46,IT1)*DDE - SPEC(48,IT1)*DRA)/RT2
      GO TO 265
263   RT2 = SIN(0.01745329*SPEC(31,IT1))
      IF (RT2.EQ.0.0) GO TO 278
      RT1 = SPEC(30,IT1) + SPEC(38,IT1)/RT2
      RT2 = SPEC(31,IT1) - SPEC(43,IT1)
      GO TO 265
264   RT1 = SPEC(32,IT1) + SPEC(46,IT1)*SPEC(38,IT1)
     +                   + SPEC(47,IT1)*SPEC(43,IT1)
      RT2 = SPEC(33,IT1) + SPEC(48,IT1)*SPEC(38,IT1)
     +                   + SPEC(49,IT1)*SPEC(43,IT1)
265   IF (I1.EQ.0) GO TO 266
      IF (RT1.LT.SPEC(24,IT2) .OR. RT1.GT.SPEC(25,IT2) .OR.
     +    RT2.LT.SPEC(26,IT2) .OR. RT2.GT.SPEC(27,IT2)) GO TO 278
      GO TO 268
266   IT2 = (RT1 - RMIN1)/RINC1 + 1.0
      IF (IT2.LT.1 .OR. IT2.GT.NROW) GO TO 278
```

```
      IT3 = (RT2 - RMIN2)/RINC2 + 1.0
      IT5 = IT2 + (IT3-1)*NROW + IBIN - 1
      IF (IT5.LT.IBIN .OR. IT5.GT.LBIN) GO TO 278
      CALL GET(IT5,IT2,IERR,6)
      IF (IERR.LT.0) GO TO 100
268   IT3 = 0
      IF (IACF.EQ.0) GO TO 272
      IF (ISPEC(23,IT1).EQ.0) GO TO 272
      SUM1 = 0.0
      DO 271 J = 1,13
      SUM2 = 0.0
      DO 270 K = IFX1,IFX2
      IT4 = J + K - 7
      IF (IT4.LT.61 .OR. IT4.GT.186) GO TO 270
      SUM2 = SUM2 + SPEC(IT4,IT1)*SPEC(K,IT2)
270   CONTINUE
      IF (SUM2.LE.SUM1) GO TO 271
      SUM1 = SUM2
      IT3 = J - 7
271   CONTINUE
272   DO 273 J = 61,186
273   SPEC(J,IT2) = SPEC(J,IT2) + SPEC(J+IT3,IT1)
      IF (ISPEC(23,IT2).GT.0) GO TO 276
      DO 274 J = 1,46
274   ISPEC(J,IT2) = ISPEC(J,IT1)
      ISPEC(23,IT2) = 0
      ISPEC(33,IT2) = MCORD
      DO 275 J = 28,53
275   SPEC(J,IT2) = SPEC(J,IT1)
      SPEC(19,IT2) = 0.0
276   SPEC(19,IT2) = SPEC(19,IT2) + SPEC(19,IT1)
      NBIN1 = ISPEC(23,IT1)
      IF(NBIN1 .EQ. 0) NBIN1 = 1
      ISPEC(23,IT2) = ISPEC(23,IT2) + NBIN1
      IF (I1.EQ.0) CALL PUT(IT2,IT5,IERR)
      IF (IERR.LT.0) GO TO 100
278   CONTINUE
      IF (I1.NE.0) CALL PUT(IT2,I1,IERR)
      GO TO 4
C
C...STR,N,M     STORE SPECTRUM
C
280   IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,I1,IERR,5)
      IF (IERR.LT.0) GO TO 100
      CALL PUT (I1,I2,IERR)
      IF (IERR.LT.0) GO TO 100
      IF (IBIN.LE.I2 .AND. I2.LE.LBIN) LBIN = 0
      GO TO 4
C
C...SWP,N,M     SWAP SPECTRA
C
300   IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
```

```
      CALL GET(I2,IT2,IERR,6)
      DO 302 I = 1,384
      IT3 = ISPEC(I,IT1)
      ISPEC(I,IT1) = ISPEC(I,IT2)
302   ISPEC(I,IT2) = IT3
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 100
      IF (IBIN.LE.I1 .AND. I1.LE.LBIN) LBIN = 0
      CALL PUT(IT2,I2,IERR)
      IF (IERR.LT.0) GO TO 100
      IF (IBIN.LE.I2 .AND. I2.LE.LBIN) LBIN = 0
      GO TO 4
C
C...WRH,N,M     WRITE HEADER
C
320   IF (IHDF.NE.0 .OR. (I1 .NE. LUTP1 .AND. I1 .NE. LUTP2)) GO TO 323
      WRITE(LUUC,321)
321   FORMAT(/" EOF-MARK HAS NOT BEEN WRITTEN."
     +       /" OK TO PROCESS HEADER? _")
      READ(LUUC,322) IT1
322   FORMAT(A2)
      IF (IT1.EQ.2HOK .OR. IT1.EQ.2HYE) GO TO 323
      GO TO 2
323   WRITE(LUOUT,324)
324   FORMAT(/"60 CHAR/LINE; 20 LINES MAX. /E ON NEW LINE TO END. ")
      IF (I2.EQ.0) I2 = LUIN
      ITTY = IFTTY(I2)
      IF(I2 .NE. LUTP1 .AND. I2 .NE. LUTP2 .AND. .NOT. ITTY) GO TO 2
      IF(I1 .NE. LUTP1 .AND. I1 .NE. LUTP2 .AND. I1 .NE. LUPRNT)GO TO 2
      DO 326 I = 1,571,30
      READ(I2,325) (IBUF(J),J=I,I+29)
325   FORMAT(30A2)
      IF (IBUF(I).EQ.2H/E) GO TO 327
326   CONTINUE
      IBUF(600) = 599
      GO TO 328
327   IBUF(600) = I - 1
328   IF (MODE.NE.1) GO TO 331
      WRITE(LUOUT,329) IBEL,(IBUF(I),I=1,IBUF(600))
329   FORMAT(//A2," THE HEADER RECORD:"//(1X,30A2))
      WRITE(LUOUT,330)
330   FORMAT(10X,"-----HEADER END-----"//
     +       " IS HEADER CORRECT? _")
      READ(LUUC,322) IT1
      IF (IT1.NE.2HYE) GO TO 2
331   IHDF = 0
      IF(I1 .EQ. LUPRNT) GO TO 333
      LUTP = I1
      ICONW = 100B + I1
      CALL EXEC(2,ICONW,IBUF,600)
      IF (TSTAT(1)) GO TO 2
      GO TO 337
333   WRITE(LUPRNT,334)
334   FORMAT(1H0)
```

```
      WRITE(LUPRNT,335)(IBUF(I),I=1,IBUF(600))
335   FORMAT(1X,30A2)
      WRITE(LUPRNT,334)
337   IF (MODE.NE.1) GO TO 4
      WRITE(LUOUT,338) I1
338   FORMAT(" HEADER WRITTEN TO LOGICAL UNIT",I3)
      GO TO 4
C
C...WRT,N,M,    WRITE SPECTRUM
C
340   IF (I1.LT.1) GO TO 2
      IF (I2.LE.0) I2 = LUOUT
      CALL GET(I1,I1,IERR,5)
      IF (IERR.LT.0) GO TO 100
      IF (I2.EQ.LUTP1 .OR. I2.EQ.LUTP2) GO TO 353
      ITTY = IFTTY(I2)
      IF(I2 .NE. LUPRNT .AND. .NOT. ITTY) GO TO 2
      IF (I2.EQ.LUPRNT) WRITE(I2,341)
341   FORMAT(1H0/1H0)
      RT1 = FLOAT(ISPEC(9,I1)) + FLOAT(ISPEC(10,I1))/1000.0
      RT2 = FLOAT(ISPEC(15,I1)) + FLOAT (ISPEC(16,I1))/1000.0
      IF (IAND(IPRNT,7B).NE.0) WRITE(I2,342) SPEC(18,I1),
     +  (ISPEC(J,I1),J=1,3),(ISPEC(J,I1),J=5,8),RT1,
     +  (ISPEC(J,I1),J=13,14),RT2,ISPEC(4,I1),SPEC(19,I1)
342   FORMAT("    FDSC    SPEC VOY STAT  YR/DAY    SCET",11X,"ERT",6X,
     +        "MODE DURATION"/2X,F7.0,2I4,2X,O4,2X,I2,"/",2I3,":",I2,
     +        ":",F6.3,I4,":",I2,":",F6.3,I3,1X,F7.2/)
      IF (IAND(IPRNT,70B).EQ.0) GO TO 349
      IT2 = ISPEC(33,I1) + 1
      GO TO (343,344,345,346),IT2
343   IT2 = 2H
      IT3 = 2H
      GO TO 347
344   IT2 = 2H L
      IT3 = 2H W
      GO TO 347
345   IT2 = 2HAZ
      IT3 = 2HEL
      GO TO 347
346   IT2 = 2HRA
      IT3 = 2HDE
347   WRITE(I2,348) (ISPEC(J,I1),J=17,19),(ISPEC(J,I1),J=21,24),
     +  (SPEC(J,I1),J=20,23),ISPEC(20,I1),IT2,IT2,IT3,IT3,IT2,IT3,
     +   (SPEC(J,I1),J=24,29)
348   FORMAT("     FPN   CAL SCAT CM HVL NBIN SMTH  PEAK     INTG     MEAN",
     +       4X,"SCALE",4X,"LN"/2X,I4,2(1X,I4),I4,I3,2(1X,I4),3(1X,F7.1),
     +        1X,E10.5,1X,I2/"  COORD MODE",3X,A2," MIN",3X,A2," MAX",3X,
     +        A2," MIN",3X,A2," MAX",4X,"HV",5X,"TEMP"/5X,A2,"/",A2,2X,
     +        4(2X,F7.2),2X,F6.1,2X,F6.2/)
349   IF (IAND(IPRNT,700B).NE.0) WRITE(I2,350) (ISPEC(J,I1),J=25,32),
     +  (SPEC(J,I1),J=50,53),(SPEC(J,I1),J=46,49)
350   FORMAT(3X,"JPLHP  JPLEDR   HPEDR   SPEC  SUN RA  SUN DEC   PITCH",
     +        4X,"YAW"/2X,I6,2X,3A2,2X,3A2,2X,I5,2X,F7.2,1X,F7.2,2X,F7.3,
     +        1X,F7.3/3X,"TRANSLATION MATRIX",3X,"A",8X,"B",8X,"C",8X,
```

```
     +        "D"/18X,4(1X,F8.4)/)
      IF (IAND(IPRNT,7000B).NE.0) WRITE(I2,351) (SPEC(J,I1),J=32,33),
     +  (SPEC(J,I1),J=30,31),(SPEC(J,I1),J=36,40),(SPEC(J,I1),J=34,35),
     +  (SPEC(J,I1),J=41,45)
351   FORMAT(8X,"RA",6X,"DEC",5X,"AZ",6X,"EL",5X,"L1",5X,"L2",5X,"L3",
     +        5X,"L4",5X,"L5"/"  AG",4(1X,F7.2),5(1X,F6.3)/"  OC",
     +        2(1X,F7.2),16X,5(1X,F6.3)/)
      IF (IAND(IPRNT,70000B).NE.0) WRITE(I2,352)
     +                                  (SPEC(J,I1),J=55,192)
352   FORMAT(29X,6(1X,F6.2)/13(1X,10(1X,F6.2)/),1X,2(1X,F6.2)/)
      GO TO 4
353   IF (IHDF.EQ.1) GO TO 354
      LUTP = I2
      ICONW = 100B + I2
      CALL EXEC(2,ICONW,ISPEC(1,I1),384)
      IF (TSTAT(1)) GO TO 2
      GO TO 4
354   WRITE(LUUC,355) IBEL
355   FORMAT(A2/" HEADER HAS NOT BEEN WRITTEN."
     +          /" OK TO WRITE SPECTRUM? _")
      READ(LUUC,356) IT2
356   FORMAT(A2)
      IF (IT2.NE.2HOK .AND. IT2.NE.2HYE) GO TO 2
      IHDF = 0
      GO TO 353
C
C...RDH,N,M     READ AND DISPLAY HEADER
C
360   IF (I1.NE.LUTP1 .AND. I1.NE.LUTP2) GO TO 2
      LUTP =I1
      IF (I2) 361,361,365
361   I2 = 1 - I2
      ICONW = 1400B + LUTP
      DO 363 I = 1,I2
      CALL EXEC(3,ICONW)
362   CALL EXEC(13,LUTP,ISTAT)
      IF (IAND(ISTAT,140000B).NE.0) GO TO 362
363   CONTINUE
364   I2 = 1
365   ICONW = 1300B + LUTP
      DO 367 I = 1,I2
      CALL EXEC(3,ICONW)
366   CALL EXEC(13,LUTP,ISTAT)
      IF (IAND(ISTAT,140000B).NE.0) GO TO 366
367   CONTINUE
      ICONW = 100B + LUTP
      CALL EXEC(1,ICONW,IBUF,600)
      CALL TSTAT(1)
      WRITE(LUOUT,368) IBEL,(IBUF(I),I=1,IBUF(600))
368   FORMAT(A2/" HEADER RECORD:"/(1X,30A2))
      GO TO 4
C
C...RDT,N,M,L   READ TAPE
C
```

```
380     IF (I2.NE.0) LUTP = I2
        IF(LUTP .NE. LUTP1 .AND. LUTP .NE. LUTP2) GO TO 2
        IT1 = I1
        IF (IT1.GT.4) IT1 = 5
        IF (I3) 381,387,384
381     I3 = -I3
        ICONW = 200B + LUTP
        DO 383 I = 1,I3
        CALL EXEC(3,ICONW)
        IF (TSTAT(2)) GO TO 2
383     CONTINUE
        GO TO 387
384     ICONW = 300B + LUTP
        DO 386 I = 1,I3
        CALL EXEC(3,ICONW)
        IF (TSTAT(1)) GO TO 2
386     CONTINUE
387     ICONW = 100B + LUTP
        CALL EXEC(1,ICONW,ISPEC(1,IT1),384)
        IF (TSTAT(1)) GO TO 2
        CALL PUT(IT1,I1,IERR)
        IF (IERR.LT.0) GO TO 100
        GO TO 4
C
C...EOF,N       END-OF-FILE
C
400     IF (I1.NE.LUTP1 .AND. I1.NE.LUTP2) GO TO 2
        ENDFILE I1
        IHDF = 1
        GO TO 4
C
C...TOF         TOP-OF-FORM
C
420     CALL EXEC(3,1106B,-1)
        GO TO 4
C
C...SAV,N,M     SAVE SPECTRUM
C
440     IF (I1.LT.1) GO TO 2
        IF (I2.LT.1 .OR. I2.GT.20) GO TO 2
        IT1 = 3*I2 + 36
        CALL READF(IBASE,IERR,ISPEC(1,5),128,LEN,IT1)
        IF (IERR.LT.0) GO TO 100
        IF (ISPEC(1,5).EQ.(-1)) GO TO 444
        WRITE(LUUC,442) IBEL,I2
442     FORMAT(A2/" LOCATION",I3," HAS NOT BEEN PURGED. ")
        GO TO 2
444     CALL GET(I1,IT2,IERR,5)
        IF (IERR.LT.0) GO TO 100
        CALL WRITF(IBASE,IERR,ISPEC(1,IT2),384,IT1)
        GO TO 4
C
C...PUR,N       PURGE SAVE LOCATION
C
```

```
460   IF (I1.LT.1 .OR. I1.GT.20) GO TO 2
      I1 = 3*I1 + 36
      ISPEC(1,5) = -1
      CALL WRITF(IBASE,IERR,ISPEC(1,5),128,I1)
      IF (IERR.LT.0) GO TO 100
      GO TO 4
C
C...RCL,N,M     RECALL SPECTRUM
C
480   IF (I1.LT.1) GO TO 2
      IF (I2.LT.1 .OR. I2.GT.20) GO TO 2
      IT1 = I1
      IF (IT1.GT.4) IT1 = 5
      I2 = 3*I2 + 36
      CALL READF(IBASE,IERR,ISPEC(1,IT1),384,LEN,I2)
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 100
      GO TO 4
C
C...INV,N,M,L   INVENTORY DATA FILE
C
500   IF (I1.LT.1) GO TO 2
      IF (I2.LT.0) GO TO 2
      IF (I3.EQ.0) I3 = LUOUT
      ITTY = IFTTY(I3)
      IF(I3 .NE. LUPRNT .AND. I3 .NE. LUTP1 .AND. I3 .NE. LUTP2
     +   .AND. .NOT. ITTY) GO TO 2
      CALL GETIM(IBUF)
      CALL CLEAR(ITIME,14)
      CALL ATIME(IBUF,ITIME,ICOUNT)
      IT2 = 0
      I2 = I1 + I2 - 1
      IF(I3 .NE. LUPRNT) GO TO 501
      CALL LURQ(1,LUPRNT,1)
501   DO 512 I = I1,I2
      IF (IFBRK(IT3)) GO TO 2
      CALL GET(I,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 100
      IF (I3.EQ.LUPRNT) GO TO 506
      IF (IT2.EQ.0) WRITE(I3,502)
502   FORMAT(/5X,"  FDSC  SP V STAT YR/DAY   SCET   M   DUR    AZ",
     +       5X,"EL",5X,"RA     DEC")
      WRITE(I3,504) I,SPEC(18,IT1),(ISPEC(J,IT1),J=1,3),(ISPEC(J,IT1),
     +              J=5,9),ISPEC(4,IT1),SPEC(19,IT1),(SPEC(J,IT1),
     +              J=30,33)
504   FORMAT(I4,1X,F7.0,I3,I2,1X,O4,I3,"/",2I3,":",I2,":",2I2,
     +       5(1X,F6.2))
      IT2 = IT2 + 1
      GO TO 512
506   IF (MOD(IT2,58).EQ.0) WRITE(I3,508)IVDATA,(ITIME(K),K=1,14)
508   FORMAT(1H1//2X,"FILE: ",3A2/2X,"DATE:",14A2///
     +             9X,"FDSC  SP V STAT YR/DAY   SCET   M   DUR  FPN",2X,
     +       "CAL SCAT HVL  SCALE   AZ     EL     RA     DEC   OC RA",
     +       " OC DEC   L3     W3"/)
```

```
      WRITE(I3,510) I,SPEC(18,IT1),(ISPEC(J,IT1),J=1,3),(ISPEC(J,IT1),
     +              J=5,9),ISPEC(4,IT1),SPEC(19,IT1),(ISPEC(J,IT1),
     +              J=17,19),(ISPEC(J,IT1),J=21,22),SPEC(23,IT1),
     +              (SPEC(J,IT1),J=30,35),SPEC(38,IT1),SPEC(43,IT1)
510   FORMAT(2X,I4,1X,F7.0,I3,I2,1X,O4,I3,"/",2I3,":",I2,":",2I2,1X,
     +       F6.2,I4,2I5,2I2,9(1X,F6.2))
      IT2 = IT2 + 1
512   CONTINUE
      IF(I3 .NE. LUPRNT)GO TO 4
      CALL EXEC(3,1106B,-1)
      GO TO 4
C
C...PRF,N,M,      PRINT FLAGS
C
520   IF (I1.NE.0) GO TO 526
      WRITE (LUOUT,522)
522   FORMAT(/10X,"FLAG",10X,"STATUS")
      IT1 = 2HN
      IT2 = 2HN
      IT3 = 2HN
      IT4 = 2HN
      IT5 = 2HN
      IF (IAND(IPRNT,7B).EQ.0) IT1 = 2HFF
      IF (IAND(IPRNT,70B).EQ.0) IT2 = 2HFF
      IF (IAND(IPRNT,700B).EQ.0) IT3 = 2HFF
      IF (IAND(IPRNT,7000B).EQ.0) IT4 = 2HFF
      IF (IAND(IPRNT,70000B).EQ.0) IT5 = 2HFF
      WRITE(LUOUT,524) IT1,IT2,IT3,IT4,IT5
524   FORMAT("    1    IDENTIFICATION      O",A2/
     +       "    2    CHARACTERISTICS     O",A2/
     +       "    3    HISTORY             O",A2/
     +       "    4    POINTING            O",A2/
     +       "    5    SPECTRUM            O",A2/)
      GO TO 4
526   IF (I1.LT.1 .OR. I1.GT.5) GO TO 2
      IF (IAND(IPRNT,IMASK(I1+5)).EQ.0) GO TO 528
      IPRNT = IAND(IPRNT,IMASK(I1))
      GO TO 4
528   IPRNT = IOR(IPRNT,IMASK(I1+5))
      GO TO 4
      END
```

```
C
C...SUBROUTINE GET
C
      SUBROUTINE GET(ILOC,IRC,IERR,II)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IERR = 0
      IF (ILOC.GT.4) GO TO 10
      IRC = ILOC
      RETURN
10    LOC = 3*ILOC - 2
      IRC = II
      CALL READF(IDATA,IERR,ISPEC(1,IRC),384,LEN,LOC)
      RETURN
      END
```

```
C
C...SUBROUTINE PUT
C
      SUBROUTINE PUT(IRC,ILOC,IERR)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IF (IRC.GT.6) GO TO 30
      IERR = 0
      IF (ILOC.GT.4) GO TO 20
      IF (IRC.EQ.ILOC) RETURN
      DO 10 I = 1,384
10    ISPEC(I,ILOC) = ISPEC(I,IRC)
      RETURN
20    LOC = 3*ILOC - 2
      CALL WRITF(IDATA,IERR,ISPEC(1,IRC),384,LOC)
      RETURN
30    IERR = -12
      RETURN
      END
```

```
C
C...FUNCTION TSTAT
C
      LOGICAL FUNCTION TSTAT(I)
      DIMENSION IUP8(2)
      COMMON ICOMN(3040)
      EQUIVALENCE (ICOMN(2916),LUUC), (ICOMN(2919),LUTP)
      DATA IUP8/2HUP,2H,8/
      TSTAT = .FALSE.
10    CALL EXEC(13,LUTP,ISTAT)
      IF (IAND(ISTAT,140000B).NE.0) GO TO 10
      GO TO (20,50),I
20    IF (IAND(ISTAT,242B).EQ.0) RETURN
      CALL MESSS(IUP8,4)
      CALL EXEC(3,LUTP)
      IF (IAND(ISTAT,240B).NE.0) GO TO 30
      WRITE(LUUC,25) LUTP
25    FORMAT(" PARITY ERROR LU",I2)
      RETURN
30    TSTAT = .TRUE.
      WRITE(LUUC,35) LUTP
35    FORMAT(" END OF FILE/TAPE LU",I2)
      RETURN
50    IF (IAND(ISTAT,100B).EQ.0) RETURN
      CALL EXEC(3,LUTP)
      TSTAT = .TRUE.
      WRITE(LUUC,55) LUTP
55    FORMAT(" START OF TAPE LU",I2)
      RETURN
      END
```

```
C
C
C
C
      SUBROUTINE ATIME(IDATE,IASC,NASC), 6/26/79
      DIMENSION IDATE(6),IASC(1),IDTM(12),MONTH(6,12),ITEMP(5),
     +          IMASK(5)
      DATA IDTM/0,31,59,90,120,151,181,212,243,273,304,334/
      DATA MONTH/7,2HJA,2HNU,2HAR,2HY ,2H  ,
     +           8,2HFE,2HBR,2HUA,2HRY,2H  ,
     +           5,2HMA,2HRC,2HH ,2H  ,2H  ,
     +           5,2HAP,2HRI,2HL ,2H  ,2H  ,
     +           3,2HMA,2HY ,2H  ,2H  ,2H  ,
     +           4,2HJU,2HNE,2H  ,2H  ,2H  ,
     +           4,2HJU,2HLY,2H  ,2H  ,2H  ,
     +           6,2HAU,2HGU,2HST,2H  ,2H  ,
     +           9,2HSE,2HPT,2HEM,2HBE,2HR ,
     +           7,2HOC,2HTO,2HBE,2HR ,2H  ,
     +           8,2HNO,2HVE,2HMB,2HER,2H  ,
     +           8,2HDE,2HCE,2HMB,2HER,2H  /
      DATA IMASK/2H  ,2H00,2H:0,2H0:,2H00/
      DO 10 IM = 12,1,-1
      IF (IDTM(IM).LT.IDATE(5)) GO TO 20
   10 CONTINUE
   20 IDAY = IDATE(5) - IDTM(IM)
      IF (MOD(IDATE(6),4).NE.0) GO TO 30
      IF (IDATE(5).GT.59) IDAY = IDAY - 1
      IF (IDAY.GT.0) GO TO 30
      IM = IM - 1
      IDAY = IDATE(5) - IDTM(IM)
      IF (IDATE(5).GT.60) IDAY = IDAY - 1
   30 CALL CODE
      WRITE(IASC,40) IDAY,(MONTH(I,IM),I=2,6)
   40 FORMAT(I2,1X,5A2,1X)
      INC = 5 + MONTH(1,IM)
      INW = (INC + 1)/2
      IF (MOD(INC,2).EQ.0) GO TO 60
      CALL CODE
      WRITE(ITEMP,50) IDATE(6)
   50 FORMAT(I4,2X)
      GO TO 80
   60 CALL CODE
      WRITE(ITEMP,70) IDATE(6)
   70 FORMAT(1X,I4,1X)
   80 IASC(INW) = ITEMP(1)
      IASC(INW+1) = ITEMP(2)
      IASC(INW+2) = ITEMP(3)
      INW = INW + 2
      CALL CODE
      WRITE(ITEMP,90) IDATE(4),IDATE(3),IDATE(2)
   90 FORMAT(2X,I2,":",I2,":",I2)
      DO 100 I = 1,5
  100 IASC(INW+I) = IOR(ITEMP(I),IMASK(I))
      NASC = INW + 5
```

```
0700        RETURN
0701        END
```

```
C
C
C
C
      SUBROUTINE GETIM( BUF ), 1/15/79
C
C  GETIM RETURNS THE TIME INTO BUF
C
C  WRITTEN BY TIM MCBREEN - 1/15/79
C
C     BUF(1) :  MSEC
C     BUF(2) :  SEC
C     BUF(3) :  MIN
C     BUF(4) :  HOUR
C     BUF(5) :  DAY
C     BUF(6) :  YEAR
C
      IMPLICIT INTEGER  ( A-Z )

      DIMENSION         BUF(6)

      CALL EXEC( 11, BUF(1), BUF(6) )

      RETURN
      END
```

```
C
C
C
C
      SUBROUTINE CLEAR(IBUF,N)
C
C...CLEARS N WORDS OF IBUF
C
      DIMENSION IBUF(1)
C
      DO 10 I = 1,N
      IBUF(I) = 2H
10    CONTINUE
      RETURN
      END
```

```
C
C...PROGRAM VOY2   LAST EDIT: 27 NOVEMBER 1978
C
FTN4
      PROGRAM VOY2(5),  VERSION 112778 DRM
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
     +          LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10)
      EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
     +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
      EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
     +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
     +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
     +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
     +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
     +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
     +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
     +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
     +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
     +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
     +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
     +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
     +            (MISC(37),IFX1),   (MISC(38),IFX2),   (MISC(39),IFX3),
     +            (MISC(40),IFX4),   (MISC(41),ISCF),   (MISC(42),IACF),
     +            (MISC(43),IPOSF),  (MISC(44),IDQF),   (MISC(45),MCORD),
     +            (MISC(46),IDUMP),  (MISC(47),IBIN),   (MISC(48),LBIN),
     +            (MISC(49),NROW),   (MISC(50),ITYPE),
     +                               (MISC(56),IT5),    (MISC(57),IT4),
     +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
      EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
     +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
     +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
     +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
     +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
     +            (MISC(89),RMIN1),  (MISC(91),RMIN2),  (MISC(93),RINC1),
     +            (MISC(95),RINC2),  (MISC(97),RA0),    (MISC(99),DEC0),
     +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
     +            (MISC(107),RT2),   (MISC(109),REF)
      INTEGER OVRLAY(3),IUP8(2)
      LOGICAL IFBRK
      EQUIVALENCE (REG,IREG)
      DATA OVRLAY/2HVO,2HY1,2H  /
      DATA IUP8/2HUP,2H,8/
      DATA IBEL/3400B/
      IF (I2.LT.1) GO TO 120
      LOC = 3*I2 - 2
      CALL WRITF(IDATA,IERR,ISPEC,1536,1)
      REG = LURQ(140001B,LUTP,1)
      IF(IREG .NE. 0) GO TO 120
      ICONW = 100B + LUTP
      IF (I3.NE.0) GO TO 50
      CALL EXEC(1,ICONW,ISPEC,1920)
C
```

```
C...FIND FIRST SPECTRUM OF STARTING FDSC
C   IF RECORD IS IN OLD FORMAT, TRANSLATE TO NEW FORMAT
C
      IT1 = 0
   20 IT1 = IT1 + 1
      IF(SPEC(18,IT1).EQ.0.0)CALL FIXIT(ISPEC(18,IT1),SPEC(18,IT1),LUUC)
      IF (SPEC(18,IT1).GE.FDSC) GO TO 40
      IF (IT1.LT.5) GO TO 20
      GO TO 50
C
C...TRANSFER SPECTRA TO THE DISK
C
40    DO 45 I = IT1,5
      IF (IBIN.LE.I2 .AND. I2.LE.LBIN) LBIN = 0
      IF(SPEC(18,I).EQ.0.0) CALL FIXIT(ISPEC(1,I),SPEC(1,I),LUUC)
      IF (DQF.LT.10.0) CALL CHECK(ISPEC(3,I),SPEC(1,I),DQF)
      CALL WRITF(IDATA,IERR,ISPEC(1,I),384,LOC)
      IF (IERR.LT.0) GO TO 80
      I1 = I1 - 1
      IF (I1.LE.0) GO TO 100
      I2 = I2 + 1
      LOC = LOC + 3
45    CONTINUE
50    IT1 = 1
      IF (IFBRK(IT2)) GO TO 140
C
C...READ NEXT TAPE RECORD
C
      STAT = EXEC(1,ICONW,ISPEC,1920)
      IF (IAND(ISTAT,242B).EQ.0) GO TO 40
C
C...TAPE ERROR
C
      CALL MESSS(IUP8,4)
      CALL EXEC(3,LUTP)
      IF (IAND(ISTAT,240B).EQ.0) GO TO 40
      MODE = 1
      LUIN = LUUC
      WRITE(LUUC,60) IBEL,ISTAT,I1,I2
60    FORMAT(A2/07," END-OF-FILE.",I4," SPECTRA NOT TRANSFERRED."/
     +       " LAST LOCATION ON DISK =",I5)
      GO TO 100
C
C...FILE WRITE ERROR
C
80    MODE = 1
      LUIN = LUUC
      WRITE(LUUC,90) IBEL,IERR,I1,I2
90    FORMAT(A2/" FILE MANAGER ERROR",I4,I5," SPECTRA NOT TRANSFERRED."/
     +       " LAST LOCATION ON DISK =",I5)
C
C...TRANSFER COMPLETED
C
100   CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
```

```
110     IPARM(1) = 0
        REG = LURQ(140000B,ITEMP,1)
        CALL EXEC(8,OVRLAY)
        CALL VOY
C
C...ERROR...INPUT PARAMETERS
C
120     WRITE(LUUC,130) IBEL
130     FORMAT(A2," ILLEGAL PARAMETER"/)
        MODE = 1
        LUIN = LUUC
        GO TO 110
C
C...BREAK
C
140     WRITE(LUUC,150) IBEL,I1,I2
150     FORMAT(A2/" INTERRUPT.",I4," SPECTRA NOT TRANSFERRED."/
       +        " LAST LOCATION ON DISK =",I5)
        MODE = 1
        LUIN = LUUC
        GO TO 100
        END
```

```
C
C...SUBROUTINE CHECK
C
      SUBROUTINE CHECK(ISTAT,SPEC,DQF)
      DIMENSION SPEC(1)
      IF (ISTAT.GT.777B) RETURN
      X2 = SPEC(65) + 1.0
      DO 10 I = 66,186
      X1 = X2
      X2 = SPEC(I) + 1.0
      IF (X1.GT.(X2*DQF) .OR. X1.LT.(X2/DQF)) GO TO 20
10    CONTINUE
      RETURN
20    ISTAT = 1000B + ISTAT
      RETURN
      END
```

```
C
C...SUBROUTINE FIXIT
C
      SUBROUTINE FIXIT(IBUF,BUF,LUUC)
      DIMENSION IBUF(384),BUF(192),ITEMP(384),TEMP(192),
     +          NBUF(3),A(10,10),B(10),X(10)
      EQUIVALENCE  (ITEMP,TEMP)
      DATA DTR/0.01745329/
      DATA RTD/57.29578/
C
C...THIS SUBROUTINE TRANSLATES A RECORD FROM MJS FORMAT TO VOY FORMAT,
C   MAKING THE NECESSARY CALCULATIONS.
C
      DO 5 I = 1,17
    5 ITEMP(I) = IBUF(I)
      ITEMP(3) = 3
      ITEMP(18) = 0
      ITEMP(19) = 0
      ITEMP(20) = 0
      ITEMP(21) = IBUF(19)
      ITEMP(22) = IBUF(20)
      ITEMP(23) = 0
      ITEMP(24) = IBUF(18)
      ITEMP(25) = 0
      CALL CODE
      WRITE(NBUF,10) IBUF(382)
   10 FORMAT(I6)
      ITEMP(26) = NBUF(1)
      ITEMP(27) = NBUF(2)
      ITEMP(28) = NBUF(3)
      CALL CODE
      WRITE(NBUF,10) IBUF(381)
      ITEMP(29) = NBUF(1)
      ITEMP(30) = NBUF(2)
      ITEMP(31) = NBUF(3)
      ITEMP(32) = IBUF(384)
      ITEMP(33) = IBUF(21)
      ITEMP(34) = IBUF(22)
      DO 20 I = 18,27
   20 TEMP(I) = BUF(I-6)
      DO 30 I = 28,33
   30 TEMP(I) = BUF(I+134)
      DO 40 I = 55,186
   40 TEMP(I) = BUF(I-33)
      DO 50 I = 187,192
   50 TEMP(I) = BUF(I-31)
      SINEL = SIN(BUF(165)*DTR)
      TEMP(36) = SINEL*BUF(168)
      TEMP(37) = SINEL*BUF(172)
      TEMP(38) = SINEL*BUF(176)
      TEMP(39) = SINEL*BUF(180)
      TEMP(40) = SINEL*BUF(184)
      TEMP(41) = -BUF(169)
      TEMP(42) = -BUF(173)
```

```
      TEMP(43) = -BUF(177)
      TEMP(44) = -BUF(181)
      TEMP(45) = -BUF(185)
      TEMP(34) = 0.0
      TEMP(35) = 0.0
C
C...COMPUTE ROTATION MATRIX A AND B
C
      A(1,1) = TEMP(36)
      A(1,2) = TEMP(41)
      B(1) = BUF(170)
      A(2,1) = TEMP(40)
      A(2,2) = TEMP(45)
      B(2) = BUF(186)
      X(1) = 0.0
      X(2) = 0.0
      CALL MATRX(B,A,X,2)
      TEMP(46) = X(1)
      TEMP(47) = X(2)
C
C...COMPUTE ROTATION MATRIX C AND D
C
      B(1) = BUF(171)
      B(2) = BUF(187)
      X(1) = 0.0
      X(2) = 0.0
      CALL MATRX(B,A,X,2)
      TEMP(48) = X(1)
      TEMP(49) = X(2)
      TEMP(52) = 0.0
      TEMP(53) = 0.0
      TEMP(54) = 0.0
C
C...COMPUTE RA AND DEC OF THE SUN
C
      CALL SNSP2(C1,C2,C3,IBUF,LUUC,IFLAG)
      IF(IFLAG .LT. 0) GO TO 70
      TEMP(50) = ATAN2(C2,C1)*RTD
      IF (TEMP(50).LT.0.0) TEMP(50) = TEMP(50) + 360.0
      TEMP(51) = ATAN2(C3,SQRT(C1*C2 + C2*C2))*RTD
      DO 60 I = 1,384
   60 IBUF(I) = ITEMP(I)
      RETURN
   70 WRITE(LUUC,80)
   80 FORMAT("  RA AND DEC OF SUN CANNOT BE COMPUTED DUE",
     +       " TO ERRORS.  SET TO ZERO.")
      TEMP(50) = 0.0
      TEMP(51) = 0.0
      DO 90 I = 1,384
   90 IBUF(I) = ITEMP(I)
      RETURN
      END
```

```
C
C...SUBROUTINE MATRX
C
      SUBROUTINE MATRX(B,A,X,N)
      DIMENSION A(10,10),B(10),X(10),AA(10,10)
C
C...THIS SUBROUTINE SOLVES A SYSTEM OF SIMULTANEOUS EQUATIONS
C   OF THE FORM AX=B,  WHERE A IS AN NXN MATRIX, X IS THE VECTOR
C   OF UNKNOWNS, B IS THE SOLUTION VECTOR,  AND THE ELEMENTS OF
C   A ARE RETURNED UNCHANGED.
C
      NP1 = N + 1
      DO 100 I = 1,N
      DO 75 J = 1,N
   75 AA(I,J) = A(I,J)
  100 AA(I,N+1) = B(I)
      DO 200 I = 1,N
      ALPHA = AA(I,I)
      DO 125 J = I,NP1
  125 AA(I,J) = AA(I,J)/ALPHA
      DO 150 K = 1,N
      IF(K-I)130,150,130
  130 BETA = AA(K,I)
      DO 140 J = I,NP1
  140 AA(K,J) = AA(K,J) - BETA*AA(I,J)
  150 CONTINUE
  200 CONTINUE
      DO 250 I = 1,N
  250 X(I) = AA(I,NP1)
      RETURN
      END
```

```
C
C...SUBROUTINE SNSP2
C
      SUBROUTINE SNSP2(C1,C2,C3,IBUF,LUUC,IFLAG)
      DIMENSION LBUF(10),NBUF(10),IBUF(384),IFILE(3,2),IDCB(144)
      REAL LC1,LC2,LC3,NC1,NC2,NC3
      EQUIVALENCE (LBUF(1),LYR) , (LBUF(2),LHR) , (LBUF(3),LSEC),
     +            (LBUF(5),LC1) , (LBUF(7),LC2) , (LBUF(9),LC3),
     +            (NBUF(1),NYR) , (NBUF(2),NHR) , (NBUF(3),NSEC),
     +            (NBUF(5),NC1) , (NBUF(7),NC2) , (NBUF(9),NC3)
      DATA IFILE/2HEP,2HHE,2HM1,2HEP,2HHE,2HM2/
      DATA LYR,LHR,LSEC/77,0,0/
      DATA DT,DTMAX/0.0,1000000.0/
      DATA NUR/0/
C
C...THIS SUBROUTINE DETERMINES THE POSITION OF THE SUN GIVEN THE
C   SPACECRAFT ID, SPACECRAFT EVENT TIME ACCORDING TO EARTH MEAN
C   EQUATOR AND EQUINOX 1950.0.
C
C...DETERMINE INTERVAL FROM LOWER BOUND TO SCET
C
      IFLAG = 0
   10 DST = 8760.0*FLOAT(IBUF(5)-LYR) + FLOAT(IBUF(6)*24 + IBUF(7)-
     +      LHR) + FLOAT(IBUF(8)*60 + IBUF(9) - LSEC)/3600.0
      IF(MOD(LYR,4) .EQ. 0 .AND. IBUF(5) .GT. LYR) DST = DST + 24.0
      IF(DST .LT. (-DT) .OR. DST .GT. DTMAX) RETURN
      IF(DST .GT. DT) GO TO 20
      C1 = LC1 + DC1*DST
      C2 = LC2 + DC2*DST
      C3 = LC3 + DC3*DST
      RETURN
C
C...NEED NEXT INTERVAL
C
   20 CALL OPEN(IDCB,IERR,IFILE(1,IBUF(2)),1,-77)
      IF(IERR .GE. 0) GO TO 40
      WRITE(LUUC,30)(IFILE(I,IBUF(2)),I=1,3),IERR
   30 FORMAT(2X,3A2," FILE MANAGER ERROR",I5)
      IFLAG = -1
      RETURN
   40 DO 50 I = 1,10
   50 LBUF(I) = NBUF(I)
      NUR = NUR + 1
      CALL READF(IDCB,IERR,NBUF,10,LEN,NUR)
      IF(IBUF(5) - NYR) 80,60,40
   60 IF(IBUF(6)*24 + IBUF(7) - NHR) 80,70,40
   70 IF(IBUF(8)*60 + IBUF(9) - NSEC) 80,40,40
   80 CALL CLOSE(IDCB)
      IF(NUR .EQ. 1) GO TO 90
      IF(NYR .EQ. 100) GO TO 110
      DT = 8760.0*FLOAT(NYR-LYR) + FLOAT(NHR-LHR) + FLOAT(NSEC-LSEC)
     +     /3600.0
      IF(MOD(LYR,4) .EQ. 0 .AND. NYR .NE. LYR) DT = DT + 24.0
      DTMAX = 8760.0
```

```
      DC1 = (NC1-LC1)/DT
      DC2 = (NC2-LC2)/DT
      DC3 = (NC3-LC3)/DT
      GO TO 10
C
C...SCET OFF END OF EPHEMERIS
C
   90 IHRS = IBUF(6)*24 + IBUF(7)
      ISECS = IBUF(8)*60 + IBUF(9)
      WRITE(LUUC,100)IBUF(5),IHRS,ISECS,NYR,NHR,NSEC
  100 FORMAT("  SCET:",I3,2I5," PRECEEDS START OF EPHEMERIS:",I3,2I5)
      IFLAG = -1
      RETURN
  110 IHRS = IBUF(6)*24 + IBUF(7)
      ISECS = IBUF(8)*60 + IBUF(9)
      WRITE(LUUC,120)IBUF(5),IHRS,ISECS,LYR,LHR,LSEC
  120 FORMAT("  SCET:",I3,2I5," OFF END OF EPHEMERIS:",I3,2I5)
      IFLAG = -1
      RETURN
      END
```

```
0001  C
0002  C...PROGRAM VOY3   LAST EDIT: 4 JUNE 1980
0003  C
0004  FTN4
0005        PROGRAM VOY3(5),  VERSION 060480 KKD
0006        DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
0007       +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
0008       +         LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10),WV(2)
0009        EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
0010       +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
0011        COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
0012        EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
0013       +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
0014       +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
0015       +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
0016       +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
0017       +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
0018       +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
0019       +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
0020       +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
0021       +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
0022       +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
0023       +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
0024       +            (MISC(37),IFX1),   (MISC(38),IFX2),   (MISC(39),IFX3),
0025       +            (MISC(40),IFX4),   (MISC(41),ISCF),   (MISC(42),IACF),
0026       +            (MISC(43),IPOSF),  (MISC(44),IDQF),   (MISC(45),MCORD),
0027       +            (MISC(46),IDUMP),  (MISC(47),IBIN),   (MISC(48),LBIN),
0028       +            (MISC(49),NROW),   (MISC(50),ITYPE),
0029       +                               (MISC(56),IT5),    (MISC(57),IT4),
0030       +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
0031        EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
0032       +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
0033       +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
0034       +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
0035       +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
0036       +            (MISC(89),RMIN1),  (MISC(91),RMIN2),  (MISC(93),RINC1),
0037       +            (MISC(95),RINC2),  (MISC(97),RA0),    (MISC(99),DEC0),
0038       +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
0039       +            (MISC(107),RT2),   (MISC(109),REF),
0040       +            (MISC(110),ISCID)
0041        INTEGER OVRLAY(3),VOYOV(9),CUR(10)
0042        LOGICAL ITTY,IFTTY
0043        DATA OVRLAY/2HVO,2H  ,2H  /
0044        DATA VOYOV/2HY1,2HY2,2HY3,2HY4,2HY5,2HY6,2HY7,2HY8,2HY9/
0045        DATA CUR/1H1,1H2,1HU,1HL,1HW,1HI,1HD/
0046        DATA NCUR/7/
0047        DATA LUPRNT,LUTEK,LUTP1,LUTP2/6,10,8,9/
0048        DATA IBEL/3400B/
0049        DATA WV/534.83,513.45/
0050  C
0051  C...INITIALIZATION
0052  C
0053        IF (IPARM(1).NE.0) 10,4
0054  2     IF (MODE.NE.1) WRITE(LUUC,3) IPS,JPC,IPC
```

```
3     FORMAT(" BLOCK NO.",I3,"   LOCATION",I3,"   ABS LOC",I4)
      MODE = 1
      LUIN = LUUC
      WRITE(LUUC,32) IBEL
32    FORMAT(A2," ILLEGAL PARAMETER(S)")
4     CALL INSTR
10    IF (IPARM(2).EQ.3) GO TO (200,220,240,260,280,300,320,340,
     +                          360,380,400,420,440,460,480,500,520), JNST
12    OVRLAY(2) = VOYOV(IPARM(2))
      IPARM(1) = 1
      CALL EXEC(8,OVRLAY)
      CALL VOY
C
C...PNT          POINT PLOT
C
200   MPLOT = 0
      GO TO 4
C
C...LNE          LINE PLOT
C
220   MPLOT = 1
      GO TO 4
C
C...YSP,R        VERTICAL SEPARATION
C
240   IF (R.LT.0.0 .OR. R.GT.10.0) GO TO 2
      YSEP = R
      GO TO 4
C
C...YHT,R        VERTICAL HEIGHT
C
260   IF (R.LT.0.0 .OR. R.GT.10.0) GO TO 2
      YLEN = R
      GO TO 4
C
C...YSC,R        VERTICAL SCALE
C
280   IF (R.LE.0.0) GO TO 2
      YSC = R
      GO TO 4
C
C...XSC,R        HORIZONTAL SCALE
C
300   IF (R.LE.0.0) GO TO 2
      XSC = R
      GO TO 4
C
C...YOR,R        Y-ORGIN
C
320   YORG = R
      GO TO 4
C
C...XOR,N,M      X-RANGE
C
```

```
340     IF (I1.LT.(-6) .OR. I1.GT.132) GO TO 2
        IF (I2.NE.0 .AND. (I2.LT.I1 .OR. I2.GT.132)) GO TO 2
        IF (I1.LT.1) I1 = I1 + 1
        IXORG = 60 + I1
        IF (I2.NE.0) GO TO 342
        IXEND = 186
        GO TO 4
342     IF (I2.LT.1) I2 = I2 + 1
        IXEND = 60 + I2
        GO TO 4
C
C...FDX,N,M     SET X-FIDUCIALS
C
360     IF (I1.LT.0 .OR. I1.GT.2) GO TO 2
        IF (I1.EQ.0) GO TO 366
        IF (I2.LT.(-6) .OR. I2.GT.132) GO TO 2
        GO TO (362,364), I1
362     IFX1 = 60 + I2
        IF (I2.LT.1) IFX1 = IFX1 + 1
        GO TO 4
364     IFX2 = 60 + I2
        IF (I2.LT.1) IFX2 = IFX2 + 1
        GO TO 4
366     IT1 = IFX1 - 60
        IT2 = IFX2 - 60
        IF (IT1.LE.0) IT1 = IT1 - 1
        IF (IT2.LE.0) IT2 = IT2 - 1
        WRITE(LUOUT,368) IT1,IT2
368     FORMAT(/5X,"FX1 =",I4,3X,"FX2 =",I4/)
        GO TO 4
C
C...FDY,N,R     SET Y-FIDUCIALS
C
380     IF (I1.LT.0 .OR. I1.GT.2) GO TO 2
        IF (I1-1) 382,386,388
382     WRITE(LUOUT,384) FY1,FY2
384     FORMAT(/5X,"FY1 =",F7.2,3X,"FY2 =",F7.2/)
        GO TO 4
386     FY1 = R
        GO TO 4
388     FY2 = R
        GO TO 4
C
C...LIP,N       LIST PLOT PARAMETERS
C
400     IF (I1.EQ.0) I1 = LUOUT
        ITTY = IFTTY(I1)
        IF(.NOT. ITTY .AND. I1 .NE. LUPRNT) GO TO 2
        IT1 = IXORG - 60
        IF (IT1.LT.1) IT1 = IT1 - 1
        IT2 = IXEND - 60
        IF (IT2.LT.1) IT2 = IT2 - 1
        WRITE(I1,402) XSC,YSC,IT1,YORG,IT2,YLEN,YSEP
402     FORMAT(/5X,"X-SCALE  =",F6.2,5X,"Y-SCALE  =",F6.2
```

```
      +        /5X,"X-ORIGIN =",I4,7X,"Y-ORIGIN =",F6.2
      +        /5X,"X-MAX    =",I4,7X,"Y-HEIGHT =",F6.2
      +        //5X,"VERTICAL SEPARATION =",F6.2)
       IF (MPLOT.EQ.0) WRITE(I1,403)
       IF (MPLOT.NE.0) WRITE(I1,404)
403    FORMAT(5X,"MODE = POINT PLOT")
404    FORMAT(5X,"MODE = LINE PLOT")
       WRITE(I1,406) LUPLT,IRBOT,IRTOP
406    FORMAT(5X,"PLOT LOGICAL UNIT =",I3
      +       /5X,"LOWER SPECTRUM =",I4
      +       /5X,"UPPER SPECTRUM =",I4/)
       GO TO 4
C
C...PLT,N,M,L    PLOT
C
420    IF (I1.LT.0) IRBOT = 0
       IF (I1.GT.0) IRBOT = I1
       CALL GET(IRBOT,IT1,IERR,5)
       IF(IRBOT .NE. 0 .AND. IERR .LT. 0) GO TO 2
       IF (I2.LT.0) IRTOP = 0
       IF (I2.GT.0) IRTOP = I2
       CALL GET(IRTOP,IT1,IERR,6)
       IF(IRTOP .NE. 0 .AND. IERR .LT. 0) GO TO 2
       IF (I3.NE.0) LUPLT = I3
       IF(LUPLT .NE. LUPRNT .AND. LUPLT .NE. LUTEK
      +   .AND. LUPLT .NE. LUTP1 .AND. LUPLT .NE. LUTP2) GO TO 2
       INST = 1
       I1 = 1
       I2 = 1
       I3 = LUPLT
       IF (I3.EQ.LUPRNT) I3 = 0
       OFSET = 0.0
       OVRLAY(2) = VOYOV(4)
       CALL EXEC(8,OVRLAY)
C
C...POF,N,R      PLOT OFFSET
C
440    IRBOT = I1
       CALL GET(IRBOT,IT1,IERR,5)
       IF(IERR .LT. 0) GO TO 2
       OFSET = R
       INST = 2
       I1 = 1
       OVRLAY(2) = VOYOV(4)
       CALL EXEC(8,OVRLAY)
C
C...PAX,N,M,L    PLOT AXIS
C
460    INST = 3
       IF(I1 .EQ. 0 .OR. I2 .EQ. 0 .OR. I3 .EQ. 0) GO TO 2
       OVRLAY(2) = VOYOV(4)
       CALL EXEC(8,OVRLAY)
C
C...PCL,N        PLOT CLOSE
```

```
C
480     IF (I1.EQ.LUPRNT) I1 = 0
        IF (I1.NE.0 .AND. I1.NE. LUTP1 .AND. I1.NE.LUTP2) GO TO 2
        I3 = I1
        INST = 4
        OVRLAY(2) = VOYOV(4)
        CALL EXEC(8,OVRLAY)
C
C...CUR          ENABLE CURSOR
C
500     CONTINUE
501     CALL CURSI(I1,I2,I3)
        DO 502 I = 1,NCUR
        IF (I1.EQ.CUR(I)) GO TO (503,504,505,506,507,507,507), I
502     CONTINUE
        IF (I1.EQ.1H ) GO TO 501
        GO TO 4
C
C...LEFT FIDUCIAL
C
503     IFX1 = IXORG + IFIX((FLOAT(I2)-100.0)*XSC/100.0)
        GO TO 501
C
C...RIGHT FIDUCIAL
C
504     IFX2 = IXORG + IFIX((FLOAT(I2)-100.0)*XSC/100.0)
        GO TO 501
C
C...TOP FIDUCIAL
C
505     FY2 = YORG + ((FLOAT(I3)-22.0)/70.0 - YSEP)*YSC
        GO TO 501
C
C...BOTTOM FIDUCIAL
C
506     FY1 = YORG + ((FLOAT(I3)-22.0)/70.0)*YSC
        GO TO 501
C
C...WAVELENGTH
C
507     CH = (FLOAT(I2) - 100.0)*XSC/100.0
        ICH = IXORG - 60 + IFIX(CH)
        RINT = YORG + ((FLOAT(I3) -22.0)/70.0)*YSC
        WVL = WV(ISCID) + (9.26)*(ICH - 1) + (CH - IFIX(CH))*9.26
        IF(I .EQ. 5)WRITE(LUUC,510)WVL
510     FORMAT(" WAVELENGTH = ",F8.0)
        IF(I .EQ. 6) WRITE(LUUC,511)RINT
511     FORMAT(" INTENSITY = ",F10.0)
        IF(I .EQ. 7) WRITE(LUUC,512)ICH
512     FORMAT(" CHANNEL NUMBER: ",I3)
        GO TO 501
C
C...PLU COMMAND
C
```

```
520   IF(I1 .NE. LUPRNT .AND. I1 .NE. LUTP1 .AND. I1 .NE. LUTP2
     +   .AND. I1 .NE. LUTEK) GO TO 2
      LUPLT = I1
      GO TO 4
      END
```

```
C
C...SUBROUTINE GET
C
      SUBROUTINE GET(ILOC,IRC,IERR,II)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IERR = 0
      IF (ILOC.GT.4) GO TO 10
      IRC = ILOC
      RETURN
10    LOC = 3*ILOC - 2
      IRC = II
      CALL READF(IDATA,IERR,ISPEC(1,IRC),384,LEN,LOC)
      RETURN
      END
```

```
C
C...PROGRAM VOY4   LAST EDIT: 10 JUNE 1980
C
FTN4
      PROGRAM VOY4(5),VERSION 061080 KKD
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
     +          LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10)
      EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
     +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
      EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
     +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
     +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
     +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
     +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
     +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
     +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
     +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
     +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
     +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
     +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
     +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
     +            (MISC(37),IFX1),   (MISC(38),IFX2),   (MISC(39),IFX3),
     +            (MISC(40),IFX4),   (MISC(41),ISCF),   (MISC(42),IACF),
     +            (MISC(43),IPOSF),  (MISC(44),IDQF),   (MISC(45),MCORD),
     +            (MISC(46),IDUMP),  (MISC(47),IBIN),   (MISC(48),LBIN),
     +            (MISC(49),NROW),   (MISC(50),ITYPE),
     +                               (MISC(56),IT5),    (MISC(57),IT4),
     +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
      EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
     +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
     +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
     +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
     +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
     +            (MISC(89),RMIN1),  (MISC(91),RMIN2),  (MISC(93),RINC1),
     +            (MISC(95),RINC2),  (MISC(97),RA0),    (MISC(99),DEC0),
     +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
     +            (MISC(107),RT2),   (MISC(109),REF),
     +            (MISC(110),ISCID)
      INTEGER OVRLAY(3)
      LOGICAL IOPEN,IAXIS,IFRST,ITEK
      DATA OVRLAY/2HVO,2HY3,2H  /
      DATA LUTEK/10/
C
C...INITIALIZATION
C
      ITEK = LUPLT.EQ.LUTEK
      GO TO (10,50,10,70), INST
C
C...OPEN PLOT FILE AND DRAW AXES
C
10    IF (IOPEN) GO TO 25
      CALL PLOTS(16.9,10.49)
```

```
      IPLOT(36) = 1
      IPLOT(35) = 192
      IPLOT(38) = -1
      IF (ITEK) GO TO 15
            CALL OPENG
      IF(IPLOT(37) .NE. 0) GO TO 150
            IOPEN = .TRUE.
            GO TO 20
C
15          CALL PAGE
            CALL OPENT
            CALL FCTXY(1.0,0.7)
C
20    CALL PLOT(1.0,0.31,-3)
      IF(IPLOT(37) .NE. 0) GO TO 150
C
C...DETERMINE AXES
C
25    XLEN = FLOAT(IXEND - IXORG + 1)/XSC
      XMIN = FLOAT(I2-1)*8.6*AINT(XLEN/8.5 + 1.0)
      XMAX = XMIN + XLEN
      YMIN = FLOAT(I1-1)*YSEP
      YMAX = YMIN + YLEN
      IF (YMAX.GT.10.0) YMAX = 10.0
      IAXIS = .TRUE.
C
C...DETERMINE IF AXES SHOULD BE PLOTTED
C
      IF (INST.NE.1) GO TO 30
      IT1 = MISC(31+I1)
      IF (IT1.LE.0) GO TO 60
      IF (IT1.GT.4) IT1 = 4 + I1
      IT2 = ISPEC(2,IT1)
      IF (I1.EQ.1) GO TO 40
      IF (IRBOT.LE.0) GO TO 40
      IF (YSEP.GE.(YLEN+0.5)) GO TO 40
      GO TO 55
30    IT2 = I3
40    CALL PAXIS(IT2,ITEK)
      IF(IPLOT(37) .NE. 0) GO TO 150
      ISCID = IT2
      IFRST = .TRUE.
      GO TO (55,50,90,70),INST
C
C...PLOT SPECTRUM
C
50    IF (ITEK) CALL OPENT
      IF (.NOT.IAXIS) GO TO 100
      IT1 = MISC(31+I1)
      IF (IT1.LE.0) GO TO 90
      IF (IT1.GT.4) IT1 = 4 + I1
55    IF (MPLOT.EQ.0) CALL PPLOT(IT1)
      IF (MPLOT.NE.0) CALL LPLOT(IT1)
      IF(IPLOT(37) .NE. 0) GO TO 150
```

```
      GO TO (60,90,90,70), INST
60    IF (I1.EQ.2) GO TO 70
      I1 = 2
      GO TO 25
C
C...CLOSE PLOT
C
70    CALL EXEC(11,IPARM,IT1)
      CALL CODE
      WRITE(IBUF,75) IT1,IPARM(5),IPARM(4),IPARM(3),IPARM(2)
75    FORMAT(I4,"/",2I3,":",I2,":",I2,1X)
      CALL JUST(0)
      CALL SYMB(0.2,10.05,0.10,IBUF,0.0,17)
      IF(IPLOT(37) .NE. 0) GO TO 150
      IF (ITEK) GO TO 80
      CALL CLOSG
      IF(IPLOT(37) .NE. 0) GO TO 150
      CALL PASS2(1,0,1,I3)
80    IOPEN = .FALSE.
      IAXIS = .FALSE.
90    IF (ITEK) CALL CLOST
      IF(ITEK) CALL HOME
      IPARM(1) = 0
      CALL EXEC(8,OVRLAY)
      CALL VOY
100   WRITE(LUUC,110)
110   FORMAT(" AXIS NOT DEFINED FOR PLOT")
      MODE = 1
      LUIN = LUUC
      GO TO 90
C
150   WRITE(LUUC,160) IPLOT(37)
160   FORMAT(" PLOT PACKAGE ERROR NUMBER ",I4/)
C
C...PURGE VECTOR FILE AND RESET FLAGS
C
      CALL PASS2(-1,0,0,0)
      IOPEN = .FALSE.
      IAXIS = .FALSE.
      MODE = 1
      LUIN = LUUC
      GO TO 90
      END
```

```
C
C...SUBROUTINE PAXIS
C
      SUBROUTINE PAXIS(ISCID,ITEK)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          INSQ(384),MISC(128)
      EQUIVALENCE (ISPEC,SPEC)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,MISC
      EQUIVALENCE (MISC(30),IXORG), (MISC(31),IXEND), (MISC(36),IFRST)
      EQUIVALENCE (MISC(67),YLEN),  (MISC(69),YSC),   (MISC(71),XSC),
     +            (MISC(73),YORG),  (MISC(75),OFSET), (MISC(77),XMIN),
     +            (MISC(79),XMAX),  (MISC(81),YMIN),  (MISC(83),YMAX)
      LOGICAL IFRST,ITEK
      DIMENSION WV(2)
      DATA WV/534.83,513.45/
      CALL PLOT(XMIN,YMIN,3)
      CALL PLOT(XMIN,YMAX,2)
      CALL PLOT(XMAX,YMAX,2)
      CALL PLOT(XMAX,YMIN,2)
      CALL PLOT(XMIN,YMIN,2)
      X = XMIN
      Y = YMIN - 0.12
      DX = 10.0/XSC
      AN = IXORG - 60
      IF (AN.LE.0.0) AN = AN - 1.0
      CALL JUST(1)
10    CALL NUMB(X,Y,0.10,AN,0.0,0)
      IF (AN.LT.0.0) AN = AN + 1.0
      AN = AN + 10.0
      X = X + DX
      IF (X.LT.XMAX) GO TO 10
      X = XMIN
      Y = YMIN - 0.17
      CALL PLOT(X,Y,3)
      DX = (200.0/9.26)/XSC
      AN = 600.0
      X = X + ((600.0 - WV(ISCID))/9.26 - IXORG + 61)/XSC
      IF(X .LT. XMIN) GO TO 24
20    CALL SYMB(X,Y,0.08,3,0.0,-2)
      CALL NUMB(X,Y-0.13,0.10,AN,0.0,0)
      CALL PLOT(X,Y,3)
24    CONTINUE
      AN = AN + 200.0
      X = X + DX
      IF (X.LT.XMAX) GO TO 20
      CALL PLOT(XMAX,Y,2)
      X = XMIN - 0.04
      Y = YMIN - 0.05
      CALL JUST(2)
      AN = YORG
      AY = 0.5*YSC
      IF(YSC .GT. 10) GO TO 30
28    CALL CODE
      WRITE(ITEMP,29)AN
```

```
29    FORMAT(F6.2)
      CALL SYMB(X,Y,0.10,ITEMP,0.0,6)
      AN = AN + AY
      Y = Y + 0.5
      IF(Y .LT. YMAX)GO TO 28
      GO TO 32
30    CALL NUMB(X,Y,0.10,AN,0.0,0)
      AN = AN + AY
      Y = Y + 0.5
      IF (Y.LT.YMAX) GO TO 30
32    IF (ITEK) RETURN
      CALL PLOT(XMIN,YMIN,-3)
      FX = 20.0/XSC
      CALL WINDO(XMIN+1.0,YMIN+0.31,XMAX+1.0,YMAX+0.31)
      CALL FCTXY(FX,1.0)
      CALL FORM(2)
      CALL FCTXY(1.0,1.0)
      CALL WINDO(0.0,0.0,16.9,10.49)
      CALL PLOT(-XMIN,-YMIN,-3)
      RETURN
      END
```

```
C
C...SUBROUTINE PPLOT
C
      SUBROUTINE PPLOT(IR)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          INSQ(384),MISC(128),IBUF(50)
      EQUIVALENCE (ISPEC,SPEC)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,MISC
      EQUIVALENCE (MISC(30),IXORG), (MISC(31),IXEND), (MISC(36),IFRST)
      EQUIVALENCE (MISC(67),YLEN),  (MISC(69),YSC),   (MISC(71),XSC),
     +            (MISC(73),YORG),  (MISC(75),OFSET), (MISC(77),XMIN),
     +            (MISC(79),XMAX),  (MISC(81),YMIN),  (MISC(83),YMAX)
      LOGICAL IFRST
      CALL WINDO(XMIN+1.0,YMIN+0.31,XMAX+1.0,YMAX+0.31)
      DO 10 I = IXORG,IXEND
      X = FLOAT(I-IXORG)/XSC + XMIN
      Y = (SPEC(I,IR) - YORG)/YSC + YMIN + OFSET
      IF(X .LT. XMIN .OR. X .GT. XMAX .OR. Y .LT. YMIN .OR. Y .GT. YMAX)
     +    GO TO 10
      CALL SYMB(X,Y,0.07,1,0.0,-1)
10    CONTINUE
      CALL WINDO(0.0,0.0,16.9,10.49)
      IF (.NOT.IFRST) RETURN
      AZ = 0.0
      IF(SPEC(31,IR) .EQ. 0.0) GO TO 12
      SINEL = SIN(0.01745329*SPEC(31,IR))
      AZ = SPEC(30,IR) + SPEC(38,IR)/SINEL
12    EL = SPEC(31,IR) - SPEC(43,IR)
      RA = SPEC(32,IR) + SPEC(46,IR)*SPEC(38,IR)
     +                 + SPEC(47,IR)*SPEC(43,IR)
      DEC = SPEC(33,IR) + SPEC(48,IR)*SPEC(38,IR)
     +                  + SPEC(49,IR)*SPEC(43,IR)
      CALL JUST(0)
      CALL CODE
      WRITE(IBUF,20) ISPEC(2,IR),SPEC(18,IR),ISPEC(1,IR),ISPEC(4,IR),
     +               (ISPEC(I,IR),I=5,10),AZ,EL,RA,DEC
C
C...NOTE CONTROL U IS USED HERE FOR MARGIN SET, AND CONTROL V
C...FOR MARGIN RETURN
C
20    FORMAT("■VOY",I2," FDSC ",F8.0,I4," MODE",I2,
     +       "■SCET",I3,"/",2I3,":",I2,":",I2,".",I3,
     +       "■AZ",F7.2,"  EL",F7.2,
     +       "■RA",F7.2," DEC",F7.2)
      IBUF(10) = IOR(IBUF(10),1B)
      X = XMIN + 0.2
      Y = YMAX - 0.23
      CALL SYMB(X,Y,0.10,IBUF,0.0,98)
      IFRST = .FALSE.
      RETURN
      END
```

```
C
C...SUBROUTINE LPLOT
C
      SUBROUTINE LPLOT(IR)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
     +          INSQ(384),MISC(128),IBUF(50)
      EQUIVALENCE (ISPEC,SPEC)
      COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,MISC
      EQUIVALENCE (MISC(30),IXORG), (MISC(31),IXEND), (MISC(36),IFRST)
      EQUIVALENCE (MISC(67),YLEN),  (MISC(69),YSC),   (MISC(71),XSC),
     +            (MISC(73),YORG),  (MISC(75),OFSET), (MISC(77),XMIN),
     +            (MISC(79),XMAX),  (MISC(81),YMIN),  (MISC(83),YMAX)
      LOGICAL IFRST
      CALL WINDO(XMIN+1.0,YMIN+0.31,XMAX+1.0,YMAX+0.31)
      X = XMIN
      Y = (SPEC(IXORG,IR) - YORG)/YSC + YMIN + OFSET
      CALL PLOT(X,Y,3)
      DO 10 I = IXORG+1,IXEND
      X = FLOAT(I-IXORG)/XSC + XMIN
      CALL PLOT(X,Y,2)
      Y = (SPEC(I,IR) - YORG)/YSC + YMIN + OFSET
      CALL PLOT(X,Y,2)
10    CONTINUE
      CALL PLOT(XMAX,Y,2)
      CALL WINDO(0.0,0.0,16.9,10.49)
      IF (.NOT.IFRST) RETURN
      AZ = 0.0
      IF(SPEC(31,IR) .EQ. 0.0) GO TO 12
      SINEL = SIN(0.01745329*SPEC(31,IR))
      AZ = SPEC(30,IR) + SPEC(38,IR)/SINEL
12    EL = SPEC(31,IR) - SPEC(43,IR)
      RA = SPEC(32,IR) + SPEC(46,IR)*SPEC(38,IR)
     +                 + SPEC(47,IR)*SPEC(43,IR)
      DEC = SPEC(33,IR) + SPEC(48,IR)*SPEC(38,IR)
     +                  + SPEC(49,IR)*SPEC(43,IR)
      CALL JUST(0)
      CALL CODE
      WRITE(IBUF,20) ISPEC(2,IR),SPEC(18,IR),ISPEC(1,IR),ISPEC(4,IR),
     +               (ISPEC(I,IR),I=5,10),AZ,EL,RA,DEC
C.
C...NOTE, CONTROL U IS USED HERE FOR MARGIN SET, AND CONTROL V
C...FOR MARGIN RETURN
C
20    FORMAT("■VOY",I2," FDSC ",F8.0,I4," MODE",I2,
     +       "■SCET",I3,"/",2I3,":",I2,":",I2,".",I3,
     +       "■AZ",F7.2,"  EL",F7.2,
     +       "■RA",F7.2," DEC",F7.2)
      IBUF(10) = IOR(IBUF(10),1B)
      X = XMIN + 0.2
      Y = YMAX - 0.23
      CALL SYMB(X,Y,0.10,IBUF,0.0,98)
      IFRST = .FALSE.
      RETURN
      END
```

```
C
C...SUBROUTINE WINDO
C
      SUBROUTINE WINDO(XA,YA,XX,YX)
      COMMON XN,YN,XF,YF,XOFF,YOFF,HST,RATIO,TOPEN,GOPEN,
     +       JCODE,XMIN,YMIN,XMAX,YMAX
      LOGICAL TOPEN,GOPEN
      XMIN = XA*XF
      YMIN = YA*YF
      XMAX = XX*XF
      YMAX = YX*YF
      CALL PLOT(XN,YN,3)
      RETURN
      END
```

```
0001  C
0002  C...PROGRAM VOY5   LAST EDIT: 27 NOVEMBER 1979  KKD
0003  C
0004  FTN4
0005
0006        PROGRAM VOY5(5),  VERSION 112779 KKD
0007        DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
0008       +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
0009       +          LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10)
0010        EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
0011       +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
0012        COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
0013        EQUIVALENCE (MISC(1),INST),   (MISC(2),I1),    (MISC(3),I2,R),
0014       +            (MISC(4),I3),     (MISC(5),MODE),  (MISC(6),JNST),
0015       +            (MISC(7),IPC),    (MISC(8),IPS),   (MISC(9),IPCOF),
0016       +            (MISC(10),JPC),   (MISC(11),IPEND), (MISC(12),LUUC),
0017       +            (MISC(13),LUIN),  (MISC(14),LUOUT), (MISC(15),LUTP),
0018       +            (MISC(16),LUPLT), (MISC(17),ICRB),  (MISC(18),ICRD),
0019       +            (MISC(19),IVBAS), (MISC(20),IVDAT), (MISC(21),ILC),
0020       +            (MISC(22),IHDF),  (MISC(23),ICONW), (MISC(24),ISTAT),
0021       +            (MISC(25),ITLOG), (MISC(26),IERR),  (MISC(27),LEN),
0022       +            (MISC(28),IPRNT), (MISC(29),MPLOT), (MISC(30),IXORG),
0023       +            (MISC(31),IXEND), (MISC(32),IRBOT), (MISC(33),IRTOP),
0024       +            (MISC(34),IOPEN), (MISC(35),IAXIS), (MISC(36),IFRST),
0025       +            (MISC(37),IFX1),  (MISC(38),IFX2),  (MISC(39),IFX3),
0026       +            (MISC(40),IFX4),  (MISC(41),ISCF),  (MISC(42),IACF),
0027       +            (MISC(43),IPOSF), (MISC(44),IDQF),  (MISC(45),MCORD),
0028       +            (MISC(46),IDUMP), (MISC(47),IBIN),  (MISC(48),LBIN),
0029       +            (MISC(49),NROW),  (MISC(50),ITYPE),
0030       +                              (MISC(56),IT5),
0031        EQUIVALENCE (MISC(58),IT3),   (MISC(59),IT2),   (MISC(57),IT4),
0032       +            (MISC(24),STAT),  (MISC(61),FDSC),  (MISC(60),IT1)
0033       +            (MISC(65),YSEP),  (MISC(67),YLEN),  (MISC(63),DQF),
0034       +            (MISC(71),XSC),   (MISC(73),YORG),  (MISC(69),YSC),
0035       +            (MISC(77),XMIN),  (MISC(79),XMAX),  (MISC(75),OFSET),
0036       +            (MISC(83),YMAX),  (MISC(85),FY1),   (MISC(81),YMIN),
0037       +            (MISC(89),RMIN1), (MISC(91),RMIN2), (MISC(87),FY2),
0038       +            (MISC(95),RINC2), (MISC(97),RA0),   (MISC(93),RINC1),
0039       +            (MISC(101),SUM1), (MISC(103),SUM2), (MISC(99),DEC0),
0040       +            (MISC(107),RT2),  (MISC(109),REF)   (MISC(105),RT1),
0041        DOUBLE PRECISION DSUM
0042        INTEGER SMATRX(3),OVRLAY(3),VOYOV(9)
0043        LOGICAL ITTY,IFTTY
0044        DATA SMATRX/2HSM,2HAT,2HRX/
0045        DATA OVRLAY/2HVO,2H  ,2H  /
0046        DATA VOYOV/2HY1,2HY2,2HY3,2HY4,2HY5,2HY6,2HY7,2HY8,2HY9/
0047        DATA IBEL/3400B/
0048        DATA LUPRNT/6/
0049  C
0050  C...INITIALIZATION
0051  C
0052        IF (IPARM(1).NE.0) 10,4
0053      2 IF (MODE.NE.1) WRITE(LUUC,3) IPS,JPC,IPC
0054      3 FORMAT (" BLOCK NO.",I3,"   LOCATION",I3,"   ABS LOC",I4)
          MODE = 1
```

```
      LUIN = LUUC
      WRITE(LUUC,32) IBEL
   32 FORMAT(A2," ILLEGAL PARAMETER(S)")
    4 CALL INSTR
   10 IF (IPARM(2).EQ.5) GO TO (200,220,240,260,280,300,
     +                          320,340,360,380,400,420,
     +                          440,460,480,500,520,540,
     +                          560,580,600,620), JNST
   12 OVRLAY(2) = VOYOV(IPARM(2))
      IPARM(1) = 1
      CALL EXEC(8,OVRLAY)
      CALL VOY
C
C...ADD,N,M       ADD SPECTRA
C
  200 IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      CALL GET(I2,I2,IERR,6)
      IF (IERR.LT.0) GO TO 2
      DO 202 I = 55,192
  202 SPEC(I,IT1) = SPEC(I,IT1) + SPEC(I,I2)
      SPEC(19,IT1) = SPEC(19,IT1) + SPEC(19,I2)
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...SUB,N,M       SUBTRACT SPECTRA
C
  220 IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      CALL GET(I2,I2,IERR,6)
      IF (IERR.LT.0) GO TO 2
      DO 222 I = 55,192
  222 SPEC(I,IT1) = SPEC(I,IT1) - SPEC(I,I2)
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...MUL,N,M       MULTIPLY SPECTRA
C
  240 IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      CALL GET(I2,I2,IERR,6)
      IF (IERR.LT.0) GO TO 2
      DO 242 I = 55,192
  242 SPEC(I,IT1) = SPEC(I,IT1)*SPEC(I,I2)
      IF (ISPEC(3,I2).EQ.6) GO TO 244
      IF (ISPEC(3,I2).EQ.7) GO TO 246
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
```

```
  244 ISPEC(3,IT1) = IOR(ISPEC(3,IT1),10B)
      ISPEC(17,IT1) = ISPEC(17,I2)
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
  246 ISPEC(3,IT1) = IOR(ISPEC(3,IT1),20B)
      ISPEC(18,IT1) = ISPEC(18,I2)
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...DIV,N,M       DIVIDE SPECTRA
C
  260 IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      CALL GET(I2,I2,IERR,6)
      IF (IERR.LT.0) GO TO 2
      DO 266 I = 55,192
      IF (SPEC(I,I2).EQ.0.0) 262,264
  262 SPEC(I,IT1) = 0.0
      GO TO 266
  264 SPEC(I,IT1) = SPEC(I,IT1)/SPEC(I,I2)
  266 CONTINUE
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...ADC,N,R       ADD CONSTANT TO SPECTRA
C
280   IF(I1.LT.1)GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      DO 282 I = 55,192
  282 SPEC(I,IT1) = SPEC(I,IT1) + R
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...MLC,N,R       MULTIPLY BY CONSTANT
C
  300 IF (I1.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      DO 302 I = 55,192
  302 SPEC(I,IT1) = SPEC(I,IT1)*R
      SPEC(23,IT1) = SPEC(23,IT1)*R
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...SMO,N,M       SMOOTH SPECTRUM BY 1-2-1 FUNCTION
C
  320 IF (I1.LT.1) GO TO 2
```

```
      IF (I2.LE.0) I2 = 1
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      DO 324 I = 1,I2
      RT1 = 2.0*SPEC(55,IT1) +  SPEC(56,IT1)
      RT1 = RT1/3.0
      DO 322 J = 56,191
      RT2 = SPEC(J-1,IT1)+2.0*SPEC(J,IT1)+SPEC(J+1,IT1)
      SPEC(J-1,IT1) = RT1
  322 RT1 = RT2/4.0
      RT2 = SPEC(191,IT1) + 2.0*SPEC(192,IT1)
      SPEC(191,IT1) = RT1
  324 SPEC(192,IT1) = RT2/3.0
      ISPEC(24,IT1) = ISPEC(24,IT1) + I2
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...CLR,N,M       CLEAR SPECTRUM
C
  340 IF (I1.LT.1) GO TO 2
      IF(I2 .LE. 0)I2 = 1
      DO 346 J = I1,I1+I2-1
      CALL GET(J,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      DO 342 I = 1,384
  342 ISPEC(I,IT1) = 0
      DO 344 I = 26,31
  344 ISPEC(I,IT1) = 2H
      SPEC(23,IT1) = 1.0
      CALL PUT(IT1,J,IERR)
      IF (IERR.LT.0) GO TO 2
346   CONTINUE
      GO TO 4
C
C...RPC,N,R       REPLACE BY CONSTANT
C
  360 IF (I1.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      DO 362 I = 55,192
  362 SPEC(I,IT1) = R
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...SHF,N,M       SHIFT SPECTRUM
C
  380 IF (I1.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      IF (I2.LT.0) GO TO 386
      DO 382 I = 192,55,-1
      IT2 = I + I2
```

```
      IF (IT2.GT.192) GO TO 382
      SPEC(IT2,IT1) = SPEC(I,IT1)
  382 CONTINUE
      IT2 = IT2 - 1
      IF (IT2.LT.55) GO TO 392
      DO 384 I = 55,IT2
  384 SPEC(I,IT1) = 0.0
      GO TO 392
  386 DO 388 I = 55,192
      IT2 = I + I2
      IF (IT2.LT.55) GO TO 388
      SPEC(IT2,IT1) = SPEC(I,IT1)
  388 CONTINUE
      IT2 = IT2 + 1
      IF (IT2.GT.192) GO TO 392
      DO 390 I = IT2,192
  390 SPEC(I,IT1) = 0.0
  392 CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...IXX,N          INTEGRATE BETWEEN X-FIDUCIALS
C
  400 IF (I1.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      SUM1 = 0.0
      DO 402 I = IFX1,IFX2
  402 SUM1 = SUM1 + SPEC(I,IT1)
      IT2 = IFX1 - 60
      IT3 = IFX2 - 60
      IF (MODE.NE.3) WRITE(LUOUT,403) I1,IT2,IT3,SUM1
  403 FORMAT(/" INTEGRAL SPEC",I4," BETWEEN DIODES",I4,
     +       " -",I4," =",F12.6/)
      SPEC(21,IT1) = SUM1
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...MXX,N          MEAN VALUE BETWEEN X-FIDUCIALS
C
  420 IF (I1.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      SUM1 = 0.0
      DO 422 I = IFX1,IFX2
  422 SUM1 = SUM1 + SPEC(I,IT1)
      SUM1 = SUM1/(IFX2 - IFX1 + 1)
      IT2 = IFX1 - 60
      IT3 = IFX2 - 60
      IF (MODE.NE.3) WRITE(LUOUT,423) I1,IT2,IT3,SUM1
  423 FORMAT(/"MEAN VALUE SPEC",I4," BETWEEN DIODES",I4,
     +       " -",I4," =",F12.6/)
      SPEC(22,IT1) = SUM1
```

```
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...FIT,N,M       SCALE SPEC N TO FIT SPEC M
C
  440 IF (I1.LT.1 .OR. I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      CALL GET(I2,I2,IERR,6)
      IF (IERR.LT.0) GO TO 2
      SUM1 = 0.0
      SUM2 = 0.0
      DO 442 I = IFX1,IFX2
      SUM1 = SUM1 + SPEC(I,IT1)
  442 SUM2 = SUM2 + SPEC(I,I2)
      RT1 = SUM2/SUM1
      DO 444 I = 55,192
  444 SPEC(I,IT1) = SPEC(I,IT1)*RT1
      IF (MODE.NE.3) WRITE(LUOUT,445) I1,RT1
  445 FORMAT(/"SCALE FACTOR SPEC",I4," =",F12.6/)
      SPEC(23,IT1) = SPEC(23,IT1)*RT1
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...PEK,N         PEAK HEIGHT
C
  460 IF (I1.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      RT1 = 0.0
      DO 462 I = 55,192
      IF (SPEC(I,IT1).LE.RT1) GO TO 462
      RT1 = SPEC(I,IT1)
  462 CONTINUE
      IF (MODE.NE.3) WRITE(LUOUT,463) I1,RT1
  463 FORMAT(/"PEAK HEIGHT SPEC",I4," =",F12.6/)
      SPEC(20,IT1) = RT1
      CALL PUT(IT1,I1,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...SCF,N         SPACECRAFT ID FLAG
C
  480 ISCF = I1
      GO TO 4
C
C...ACF,N         AUTO-CORRELATE FLAG
C
  500 IACF = I1
      GO TO 4
C
C...POS,N,M       POSITION FLAG AND MODE
```

```
C
  520 IPOSF = I1
      IF (IPOSF.NE.1) GO TO 4
      IF (I2.LT.1 .OR. I2.GT.3) GO TO 2
      MCORD = I2
      GO TO 4
C
C...DQF,N,M       DATA QUALITY FLAG
C
  540 IF (I1.EQ.0) IDQF = IAND(IDQF,700B)
      IF (I1.NE.0) IDQF = IOR(IDQF,7000B)
      IF (I2.EQ.0) IDQF = IAND(IDQF,7000B)
      IF (I2.NE.0) IDQF = IOR(IDQF,700B)
      GO TO 4
C
C...BIN,N         BIN PARAMETERS
C
  560 WRITE(LUUC,561) MCORD
  561 FORMAT(/"WHICH COORDINATE SYSTEM?",I2/5X,"1=L/W"/
     +       5X,"2=AZ/EL"/5X,"3=RA/DEC _")
      READ(LUIN,*) MCORD
      IF (MCORD.NE.1) GO TO 564
      WRITE(LUUC,562) RA0,DEC0
  562 FORMAT(/"WHAT IS FIXED RA AND DEC?",2F10.3,"  _")
      READ(LUIN,*) RA0,DEC0
  564 WRITE(LUUC,565) RMIN1,RMIN2
  565 FORMAT(/"WHAT ARE THE MINIMUMS?",2F10.3,"  _")
      READ(LUIN,*) RMIN1,RMIN2
      WRITE(LUUC,566) RINC1,RINC2
  566 FORMAT(/"WHAT ARE THE INCREMENTS?",2F10.3,"  _")
      READ(LUIN,*) RINC1,RINC2
      IF (I1.EQ.1) GO TO 572
      WRITE(LUUC,567) IDUMP
  567 FORMAT(/"DO YOU WANT A DUMP OF INTERMEDIATE VALUES?",I3/
     +       5X,"0=NO DUMP"/5X,"LU=DUMP TO THAT LU  _")
      READ(LUIN,*) IDUMP
      IT1 = IFX1 - 60
      IT2 = IFX2 - 60
      IT3 = IFX3 - 60
      IT4 = IFX4 - 60
      WRITE(LUUC,568) IT1,IT2
  568 FORMAT(/"DEFINE REGION 1 - CHANNEL A TO CHANNEL B",I4,
     +          I4,"  _")
      READ(LUIN,*) IT1,IT2
      IFX1 = IT1 + 60
      IFX2 = IT2 + 60
      WRITE(LUUC,569) IT3,IT4
  569 FORMAT(/"DEFINE REGION 2 - CHANNEL A TO CHANNEL B",I4,
     +       I4,"  _")
      READ(LUIN,*) IT3,IT4
      IFX3 = IT3 + 60
      IFX4 = IT4 + 60
      WRITE(LUUC,570) ITYPE
  570 FORMAT(/"DEFINE BIN TYPE",I2/5X,"1=SUM CHANNELS ON REGION 1"/
```

```
     +        5X,"2=SUM CHANNELS ON REGION 2"/5X,"3=SUM CHANNELS ON BOTH"
     +        " REGIONS"/5X,"4=DIFFERENCE BETWEEN REGIONS (R1-R2)"/5X,
     +        "5=RATIO OF REGIONS (R1/R2)     _")
      READ(LUIN,*) ITYPE
      GO TO 4
  572 NCOL= (LBIN - IBIN + 1)/NROW
      WRITE(LUUC,573) NROW,NCOL
  573 FORMAT(/"WHAT ARE THE SPECTRAL ARRAY DIMENSIONS? (ROWS,COLS)",
     +        2I5,"   _")
      READ(LUIN,*) NROW,NCOL
      WRITE(LUUC,574) IBIN
  574 FORMAT(/"WHAT IS THE STARTING LOCATION?",I5,"   _")
      READ(LUIN,*) IBIN
      LBIN = NROW*NCOL + IBIN - 1
      DO 575 I = 1,384
  575 ISPEC(I,5) = 0
      ISPEC(33,5) = MCORD
      RT1 = RMIN1
      RT2 = RMIN2
      IT1 = NROW + IBIN
      DO 576 I = IBIN,LBIN
      SPEC(24,5) = RT1
      SPEC(25,5) = RT1 + RINC1
      SPEC(26,5) = RT2
      SPEC(27,5) = RT2 + RINC2
      CALL PUT(5,I,IERR)
      IF (IERR.LT.0) GO TO 2
      RT1 = RT1 + RINC1
      IF((I+1).LT.IT1) GO TO 576
      RT1 = RMIN1
      RT2 = RT2 + RINC2
      IT1 = IT1 + NROW
  576 CONTINUE
      GO TO 4
C
C...NOR,N,M       NORMALIZE SPECTRA
C
  580 IF (I1) 2,582,590
  582 IF (LBIN.LT.IBIN) GO TO 2
      IF(I2 .NE. 0) GO TO 2
      DO 586 I = IBIN,LBIN
      CALL GET(I,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      DO 584 J = 55,192
      IF (ISPEC(23,IT1).LE.1) GO TO 586
  584 SPEC(J,IT1) = SPEC(J,IT1)/ISPEC(23,IT1)
      ISPEC(23,IT1) = 1
      CALL PUT(IT1,I,IERR)
      IF (IERR.LT.0) GO TO 2
  586 CONTINUE
      GO TO 4
590   IF(I2 .LE. 0)I2 = 1
      DO 594 J = I1,I1+I2-1
      CALL GET(J,IT1,IERR,5)
```

```
      IF (IERR.LT.0) GO TO 2
      IF (ISPEC(23,IT1).LE.1) GO TO 594
      DO 592 I = 55,192
  592 SPEC(I,IT1) = SPEC(I,IT1)/ISPEC(23,IT1)
      ISPEC(23,IT1) = 1
      CALL PUT(IT1,J,IERR)
      IF (IERR.LT.0) GO TO 2
594   CONTINUE
      GO TO 4
C
C...LIB,N       LIST BIN PARAMETERS
C
  600 IF (I1.EQ.0) I1 = LUOUT
      ITTY = IFTTY(I1)
      IF(.NOT. ITTY .AND. I1 .NE. LUPRNT)GO TO 2
      WRITE(I1,601) ISCF,IACF,IPOSF,MCORD
  601 FORMAT(/" S/C FLAG  AUTO-CORRELATE FLAG  POSITION FLAG",
     +        "  COORD MODE"/4X,I1,15X,I1,16X,I1,13X,I1)
      IT1 = 2HN
      IT2 = 2HN
      IF (IAND(IDQF,7000B).EQ.0) IT1 = 2HFF
      IF (IAND(IDQF,700B).EQ.0) IT2 = 2HFF
      WRITE(I1,602) IT1,IT2
  602 FORMAT(/" DATA QUALITY FLAG:   SPECTRAL    O",A2
     +        /22X,"POSITIONAL  O",A2)
      WRITE(I1,603) RA0,DEC0
  603 FORMAT(/" FIXED RA AND DEC FOR L/W BIN:",2F10.3)
      WRITE(I1,604) RMIN1,RMIN2,RINC1,RINC2
  604 FORMAT(/" MINIMUMS:",2F10.3//" INCREMENTS:",2F10.3)
      IT1 = IFX1 - 60
      IT2 = IFX2 - 60
      IT3 = IFX3 - 60
      IT4 = IFX4 - 60
      WRITE(I1,605) IT1,IT2,IT3,IT4,ITYPE
  605 FORMAT(/" INTENSITY BIN:   REGION 1:",I4," TO",I4
     +        /18X,"REGION 2:",I4," TO",I4
     +        /18X,"BIN TYPE:",I3)
      NCOL = (LBIN - IBIN + 1)/NROW
      WRITE(I1,606) NROW,NCOL,IBIN,LBIN
  606 FORMAT(/" SPECTRAL BIN:   ARRAY DIMENSIONS:",I4," X",I4
     +        /17X,"FILE LOCATIONS:",I4," TO",I4)
      GO TO 4
C
C...SCT,N,M,L   APPLY SCATTERING MATRIX
C
  620 IF (I1.LT.1) GO TO 2
      IF (I2.LT.1) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (I3.EQ.0) I3 = ISPEC(2,IT1)
      CALL OPEN(IDCB,IERR,SMATRX,2,9850)
      IF (IERR.GE.0) GO TO 622
      WRITE(LUUC,621) IBEL,IERR
  621 FORMAT(A2/" SMATRX FILE MANAGER ERROR",I5)
      GO TO 2
```

```
  622 CALL READF(IDCB,IERR,IBUF,60,LEN,1)
      DO 623 I = 1,60,3
      IF (I3.NE.IBUF(I)) GO TO 623
      IF (I2.EQ.IBUF(I+1)) GO TO 625
  623 CONTINUE
      WRITE(LUUC,624) IBEL,I2,I3,(IBUF(I),I=1,60)
  624 FORMAT(A2/"MATRIX",I3," FOR VOYAGER",I2," NOT AVAILABLE"
     +       //"FOLLOWING MATRICES CURRENTLY ON FILE:"
     +       /" VOYAGER  MATRIX  HEADER"/(4X,I2,5X,I4,5X,I3))
      CALL CLOSE(IDCB)
      GO TO 2
  625 I = I/3
      IT1 = 252*I + 2
      CALL WRITF(IDATA,IERR,ISPEC,1536,1)
      LOC = 3*I1 - 2
      CALL READF(IDATA,IERR,ISPEC(1,3),384,LEN,LOC)
      DO 626 I = 1,384
  626 ISPEC(I,4) = ISPEC(I,3)
      IT2 = 61
      DO 632 I = 1,64
      IT3 = 0
      CALL READF(IDCB,IERR,BUF,512,LEN,IT1)
      IT1 = IT1 + 4
  628 DSUM = 0.0D0
      DO 630 J = 61,186
      IT3 = IT3 + 1
  630 DSUM = DSUM + DBLE(SPEC(J,3))*DBLE(BUF(IT3))
      SPEC(IT2,4) = SNGL(DSUM)
      IT3 = IT3 + 2
      IT2 = IT2 + 1
      IF (IT2.GT.186) GO TO 634
      IF (IT3.LT.256) GO TO 628
  632 CONTINUE
  634 ISPEC(19,4) = I2
      ISPEC(3,4) = IOR(ISPEC(3,4),40B)
      CALL CLOSE(IDCB)
      CALL WRITF(IDATA,IERR,ISPEC(1,4),384,LOC)
      CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
      GO TO 4
      END
```

```
C
C...SUBROUTINE GET
C
      SUBROUTINE GET(ILOC,IRC,IERR,II)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IERR = 0
      IF (ILOC.GT.4) GO TO 10
      IRC = ILOC
      RETURN
10    LOC = 3*ILOC - 2
      IRC = II
      CALL READF(IDATA,IERR,ISPEC(1,IRC),384,LEN,LOC)
      RETURN
      END
```

```
C
C...SUBROUTINE PUT
C
      SUBROUTINE PUT(IRC,ILOC,IERR)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IF (IRC.GT.6) GO TO 30
      IERR = 0
      IF (ILOC.GT.4) GO TO 20
      IF (IRC.EQ.ILOC) RETURN
      DO 10 I = 1,384
10    ISPEC(I,ILOC) = ISPEC(I,IRC)
      RETURN
20    LOC = 3*ILOC - 2
      CALL WRITF(IDATA,IERR,ISPEC(1,IRC),384,LOC)
      RETURN
30    IERR = -12
      RETURN
      END
```

```
0001  C
0002  C...PROGRAM VOY6   LAST EDIT: 26 NOVEMBER 1979
0003  C
0004  FTN4
0005        PROGRAM VOY6(5),  VERSION 112679 KKD
0006        DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6),SPEC(192,6),
0007       +          IDCB(16),IBUF(600),BUF(300),INSQ(4,32),RNSQ(2,32),
0008       +          LOG(4,64),RLOG(2,64),MISC(128),IPARM(5),REF(10)
0009        EQUIVALENCE (ISPEC,SPEC),  (IBUF,BUF),  (INSQ,RNSQ),
0010       +            (LOG,RLOG),  (ISPEC(1,5),IDCB),  (ISPEC(17,5),IBUF)
0011        COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
0012        EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
0013       +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
0014       +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
0015       +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
0016       +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
0017       +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
0018       +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
0019       +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
0020       +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
0021       +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
0022       +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
0023       +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
0024       +            (MISC(37),IFX1),   (MISC(38),IFX2),   (MISC(39),IFX3),
0025       +            (MISC(40),IFX4),   (MISC(41),ISCF),   (MISC(42),IACF),
0026       +            (MISC(43),IPOSF),  (MISC(44),IDQF),   (MISC(45),MCORD),
0027       +            (MISC(46),IDUMP),  (MISC(47),IBIN),   (MISC(48),LBIN),
0028       +            (MISC(49),NROW),   (MISC(50),ITYPE),
0029       +                               (MISC(56),IT5),    (MISC(57),IT4),
0030       +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
0031        EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
0032       +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
0033       +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
0034       +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
0035       +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
0036       +            (MISC(89),RMIN1),  (MISC(91),RMIN2),  (MISC(93),RINC1),
0037       +            (MISC(95),RINC2),  (MISC(97),RA0),    (MISC(99),DEC0),
0038       +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
0039       +            (MISC(107),RT2),   (MISC(109),REF)
0040        INTEGER OVRLAY(3),VOYOV(9),FFPN(3),FCAL(3)
0041        DIMENSION NAME(3),CALF(3)
0042        DATA OVRLAY/2HVO,2H  ,2H  /
0043        DATA VOYOV/2HY1,2HY2,2HY3,2HY4,2HY5,2HY6,2HY7,2HY8,2HY9/
0044        DATA FFPN/2HFP,2HNS,2HP /
0045        DATA FCAL/2HCA,2HLS,2HP /
0046        DATA CALF/9.26,1.16E-04,4.39/
0047        DATA IBEL/3400B/
0048  C
0049  C...INITIALIZATION
0050  C
0051        IF (IPARM(1).NE.0) 10,4
0052      2 IF (MODE.NE.1) WRITE(LUUC,3) IPS,JPC,IPC
0053      3 FORMAT(" BLOCK NO.",I3,"   LOCATION",I3,"   ABS LOC",I4)
0054        MODE = 1
```

```
      LUIN = LUUC
      WRITE(LUUC,32) IBEL
   32 FORMAT(A2," ILLEGAL PARAMETER")
    4 CALL INSTR
   10 IF (IPARM(2).EQ.6) GO TO (200,220,240,260,280,300,
     +                          320,340,360,380,400,420,
     +                          440,460,480,500,480,540) JNST
   12 OVRLAY(2) = VOYOV(IPARM(2))
      IPARM(1) = 1
      CALL EXEC(8,OVRLAY)
      CALL VOY
C
C...LOP,N,M       LOOP
C
  200 IF (I3.EQ.0) I3 = 1
      IF (I1.EQ.I3) 204,202
  202 INSQ(4,JPC) = I3 + 1
      IT1 = I2 - 1
      IF (IT1.GE.IPEND) GO TO 2
      IPC = IT1
      IF (IPC.LT.0) IPC = 0
      JPC = IPC - IPCOF
      IF (JPC.GE.0 .AND. JPC.LT.32) GO TO 4
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      IF (JPC.LT.0) JPC = JPC - 31
      IPS = IPS + JPC/32
      IPCOF = 32*(IPC/32)
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
      GO TO 4
  204 INSQ(4,JPC) = 0
      GO TO 4
C
C...JMP,N         JUMP
C
  220 IT1 = IPC + I1 - 1
      IF (IT1.LT.0 .OR. IT1.GE.IPEND) GO TO 2
      IPC = IT1
      JPC = IPC - IPCOF
      IF (JPC.GE.0 .AND. JPC.LT.32) GO TO 4
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      IF (JPC.LT.0) JPC = JPC - 31
      IPS = IPS + JPC/32
      IPCOF = 32*(IPC/32)
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
      GO TO 4
C
C...INC,N,M,L     INCREMENT (INTEGER)
C
  240 IF ((JPC-1).LT.1) GO TO 242
      INSQ(2,JPC-1) = INSQ(2,JPC-1) + I1
      INSQ(3,JPC-1) = INSQ(3,JPC-1) + I2
      INSQ(4,JPC-1) = INSQ(4,JPC-1) + I3
      GO TO 4
  242 CALL READF(IBASE,IERR,IBUF,128,LEN,IPS-1)
```

```
      IBUF(126) = IBUF(126) + I1
      IBUF(127) = IBUF(127) + I2
      IBUF(128) = IBUF(128) + I3
      CALL WRITF(IBASE,IERR,IBUF,128,IPS-1)
      GO TO 4
C
C...INR,N,R       INCREMENT (REAL)
C
  260 IF ((JPC-1).LT.1) GO TO 262
      INSQ(2,JPC-1) = INSQ(2,JPC-1) + I1
      RNSQ(2,JPC-1) = RNSQ(2,JPC-1) + R
      GO TO 4
  262 CALL READF(IBASE,IERR,IBUF,128,LEN,IPS-1)
      IBUF(126) = IBUF(126) + I1
      BUF(64) = BUF(64) + R
      CALL WRITF(IBASE,IERR,IBUF,128,IPS-1)
      GO TO 4
C
C...RST,N,M,L     RESET (INTEGER)
C
  280 IF (I2.LT.1 .OR. I2.GT.3) GO TO 2
      IF (I1.LT.1) GO TO 2
      IT1 = 32*(IP3-1) - IPCOF
      IF (I1.GT.(320-IT1)) GO TO 2
      JPC = I1 - IPCOF
      IF (JPC.LT.1 .OR. JPC.GT.32) GO TO 282
      INSQ(I2+1,JPC) = I3
      GO TO 4
  282 IT1 = 32*(IPS-1) + I1 - IPCOF
      IT2 = (IT1-1)/32 + 1
      IT1 = IT1 - 32*(IT2-1)
      CALL READF(IBASE,IERR,IBUF,128,LEN,IT2)
      IBUF(I2+1+4*(IT1-1)) = I3
      CALL WRITF(IBASE,IERR,IBUF,128,IT2)
      GO TO 4
C
C...RSR,N,R       RESET (REAL)
C
  300 IF (I1.LT.1) GO TO 2
      IT1 = 32*(IPS-1) - IPCOF
      IF (I1.GT.(320-IT1)) GO TO 2
      JPC = I1 - IPCOF
      IF (JPC.LT.1 .OR. JPC.GT.32) GO TO 302
      RNSQ(2,JPC) = R
      GO TO 4
  302 IT1 = 32*(IPS-1) + I1 - IPCOF
      IT2 = (IT1-1)/32 + 1
      IT1 = IT1 - 32*(IT2-1)
      CALL READF(IBASE,IERR,IBUF,128,LEN,IT2)
      BUF(2*IT1) = R
      CALL WRITF(IBASE,IERR,IBUF,128,IT2)
      GO TO 4
C
C...HLT           HALT
```

```
C
  320 MODE = 1
      GO TO 4
C
C...NOP          NO-OP  SPARE LOCATION
C
  340 GO TO 4
C
C...PUC,N,M     PILE-UP CORRECTION
C
  360 IF (I1.LT.1) GO TO 2
      IF (I2.LE.0) I2 = I1
      IF (I2.LT.I1) GO TO 2
      DO 364 I = I1,I2
      CALL GET(I,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      RT1 = SPEC(19,IT1)
      IF (RT1.LE.0.0) GO TO 364
      DO 362 J = 55,192
  362 SPEC(J,IT1) = -3125.0*RT1*ALOG(1.0-(SPEC(J,IT1)/RT1/3125.0))
      CALL PUT(IT1,I,IERR)
      IF (IERR.LT.0) GO TO 2
  364 CONTINUE
      GO TO 4
C
C...EDT,N,M      EDIT SPECTRUM
C
  380 IF (I1.LT.1 .OR. I1.GT.2 .OR. I2.LT.1) GO TO 2
      CALL GET(I2,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      CALL EDIT(I1,LUUC,ISPEC(1,IT1),SPEC(1,IT1))
      CALL PUT(IT1,I2,IERR)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...FPN,N,M      INPUT FPN SPECTRUM
C
  400 IF (I1.LT.1) GO TO 2
      IF (I2.LT.1 .OR. I2.GT.128) GO TO 2
      CALL OPEN(IDCB,IERR,FFPN,1)
      IF (IERR.LT.0) GO TO 408
      IT1 = 2*I2 - 1
      CALL READF(IDCB,IERR,ISPEC(121,6),256,LEN,IT1)
      CALL CLOSE(IDCB)
      DO 402 I = 1,120
  402 ISPEC(I,6) = 0
      ISPEC(3,6) = 6
      ISPEC(17,6) = I2
      ISPEC(26,6) = 2HFP
      ISPEC(27,6) = 2HNS
      ISPEC(28,6) = 2HP
      RT1 = SPEC(188,6)
      DO 404 I = 61,186
  404 SPEC(I,6) = RT1*SPEC(I,6)
```

```
      SPEC(22,6) = RT1
      SPEC(23,6) = 1.0
      DO 406 I = 187,192
  406 SPEC(I,6) = 0.0
      CALL PUT(6,I1,IERR)
      GO TO 4
  408 WRITE(LUUC,409) IBEL
  409 FORMAT(A2/"FILE FPNSP NOT AVAILABLE."/)
      GO TO 2
C
C...CAL,N,M,L     INPUT CALIBRATION SPECTRUM
C
  420 IF (I1.LT.1) GO TO 2
      IF (I2.LT.1 .OR. I2.GT.2) GO TO 2
      IF (I3.LT.1 .OR. I3.GT.3) GO TO 2
      CALL OPEN(IDCB,IERR,FCAL,1)
      IF (IERR.LT.0) GO TO 426
      IT1 = 2*I2 - 1
      CALL READF(IDCB,IERR,ISPEC(121,6),256,LEN,IT1)
      CALL CLOSE(IDCB)
      RT1 = CALF(I3)
      DO 421 I = 61,186
  421 SPEC(I,6) = SPEC(I,6)/RT1
      DO 422 I = 1,120
  422 ISPEC(I,6) = 0
      ISPEC(3,6) = 7
      ISPEC(18,6) = 10*I2 + I3
      ISPEC(26,6) = 2HCA
      ISPEC(27,6) = 2HLS
      ISPEC(28,6) = 2HP
      DO 424 I = 187,192
  424 SPEC(I,6) = 0.0
      SPEC(23,6) = 1.0
      CALL PUT(6,I1,IERR)
      GO TO 4
  426 WRITE(LUUC,427) IBEL
  427 FORMAT(A2/"FILE CALSP NOT AVAILABLE."/)
      GO TO 2
C
C...SET,N,R       SET REFERENCE VALUE
C
  440 IF (I1.LT.1 .OR. I1.GT.10) GO TO 2
      REF(I1) = R
      GO TO 4
C
C...SGE,N,M,L     SKIP IF GREATER THAN OR EQUAL
C
  460 IF (I1.LT.1) GO TO 2
      IF (I2.LT.1 .OR. I2.GT.209) GO  TO 2
      IF (I3.LT.1 .OR. I3.GT.10) GO TO 2
      CALL GET(I1,IT1,IERR,5)
      IF (IERR.LT.0) GO TO 2
      IF (I2.GT.34) GO TO 462
      IF (ISPEC(I2,IT1).GE.REF(I3)) GO TO 464
```

```
      GO TO 4
  462 I2 = I2 - 17
      IF (SPEC(I2,IT1).GE.REF(I3)) GO TO 464
      GO TO 4
  464 IF (IPC.EQ.IPEND) GO TO 2
      IPC = IPC + 1
      JPC = IPC - IPCOF
      IF (JPC.LE.32) GO TO 4
      CALL WRITF(IBASE,IERR,INSQ,128,IPS)
      IF (IERR.LT.0) GO TO 2
      IPS = IPS + 1
      IPCOF = IPCOF + 32
      JPC = IPC - IPCOF
      CALL READF(IBASE,IERR,INSQ,128,LEN,IPS)
      IF (IERR.LT.0) GO TO 2
      GO TO 4
C
C...CRT,N       CREATE NEW DATA FILE
C
  480 WRITE(LUUC,482)
  482 FORMAT(/5X,"FILE NAME? _")
      READ(LUUC,484) NAME
  484 FORMAT(3A2)
      WRITE(LUUC,486)
  486 FORMAT(5X,"CARTRIDGE? _")
      IT1 = 20
      READ(LUUC,*) IT1
      IF (IT1.EQ.2 .OR. IT1.EQ.3) GO TO 2
      IF (NAME(1).NE.2H  ) GO TO 488
      NAME(1) = 2HVD
      NAME(2) = 2HAT
      NAME(3) = IVDAT
      IF (IT1.EQ.20) GO TO 488
      NAME(3) = 2HA
  488 IF (JNST.EQ.16) GO TO 501
      IF (JNST.EQ.17) GO TO 520
      IF (I1.LE.0) I1 = 100
      I1 = 3*I1
      CALL CREAT(IDCB,IERR,NAME,I1,1,9850,IT1)
      IF (IERR.LT.0) GO TO 490
      CALL CLOSE(IDCB)
      GO TO 4
  490 WRITE(LUUC,492) IBEL,NAME,IT1,IERR
  492 FORMAT(2X,4A2," CARTRIDGE",I4," FILE MANAGER ERROR",I4)
      GO TO 2
C
C...COP,N,M,L   COPY FROM PRIMARY TO NAMED DATA FILE
C
  500 IF (I1.LT.1 .OR. I2.LT.1 .OR. I3.LT.1) GO TO 2
      GO TO 480
  501 CALL OPEN(IDCB,IERR,NAME,0,9850,IT1)
      IF (IERR.LT.0) GO TO 490
      CALL WRITF(IDATA,IERR,ISPEC,1536,1)
      I1 = I2 + I1 - 1
```

```
      I3 = 3*I3 - 14
      DO 502 I = I2,I1,4
      LOC = 3*I - 2
      IT1 = I + 3
      IF (IT1.GT.I1) IT1 = I1
      IT1 = 384*(IT1-I+1)
      CALL READF(IDATA,IERR,ISPEC,IT1,LEN,LOC)
      I3 = I3 + 12
      CALL WRITF(IDCB,IERR,ISPEC,LEN,I3)
      IF (LEN.LT.IT1) GO TO 504
      IF (IERR.LT.0) GO TO 508
  502 CONTINUE
      CALL CLOSE(IDCB)
      CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
      GO TO 4
  504 I = I + LEN/384 - 1
      WRITE(LUUC,506) I
  506 FORMAT("  FILE MANAGER ERROR -12. LAST LOC TRANSFERRED =",I5)
      CALL CLOSE(IDCB)
      CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
      GO TO 2
  508 I = I - 1
      WRITE(LUUC,510) NAME,IERR,I
  510 FORMAT(2X,3A2," FILE MANAGER ERROR",I4,". LAST LOC TRANSFERRED",
     +       " =",I5)
      CALL CLOSE(IDCB)
      CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
      GO TO 2
C
C...PDF      CHANGE PRIMARY DATA FILE
C
  520 CALL OPEN(IDCB,IERR,NAME,0,9850,IT1)
      IF (IERR.LT.0) GO TO 490
      CALL WRITF(IDATA,IERR,ISPEC,1536,1)
      CALL CLOSE(IDATA)
      DO 522 I = 1,16
  522 IDATA(I) = IDCB(I)
      DO 525 I = 1,3
525   IVDATA(I) = NAME(I)
      CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
      ICRD = IT1
      GO TO 4
C
C...CLN,N,M    CONVERT COUNTS TO NATURAL LOG OF COUNTS
C
540   IF(I1 .LE. 0)GO TO 2
      IF(I2.LE.0)I2 = 1
      DO 546 J = I1,I1+I2-1
      CALL GET(J,IT1,IERR,5)
      IF(IERR .LT. 0)GO TO 2
      DO 544 I = 55,192
      IF(SPEC(I,IT1) .LE. 0.0)GO TO 542
      SPEC(I,IT1) = ALOG(SPEC(I,IT1))
      GO TO 544
```

```
542   CONTINUE
      SPEC(I,IT1) = -99
544   CONTINUE
      ISPEC(20,IT1) = ISPEC(20,IT1) + 1
      CALL PUT(IT1,J,IERR)
      IF(IERR .LT. 0)GO TO 2
546   CONTINUE
      GO TO 4
C
      END
```

```
C
C...SUBROUTINE EDIT
C
      SUBROUTINE EDIT(IGO,LUUC,IBUF,BUF)
      DIMENSION IBUF(1),BUF(1)
      GO TO (10,70), IGO
   10 WRITE(LUUC,20) (I,I = 1,10)
   20 FORMAT(/2X,10(5X,I2))
      DO 30 I = 0,120,10
      IT1 = I + 61
      IT2 = IT1 + 9
      IF (IT2.GT.186) IT2 = 186
   30 WRITE(LUUC,40) I,(BUF(J),J = IT1,IT2)
   40 FORMAT(I3,10(1X,F6.2))
      WRITE(LUUC,50)
   50 FORMAT(" CORRECTIONS--ENTER CHANNEL, NEW VALUE",
     +        "--BLANK TO EXIT"//"EDT>_")
      I = 0
      READ(LUUC,*) I,R
      IF (I.EQ.0) RETURN
   60 BUF(I + 60) = R
      WRITE(LUUC,62)
62    FORMAT("EDT>_")
      I = 0
      READ(LUUC,*) I,R
      IF (I.EQ.0) 10,60
   70 WRITE(LUUC,80)
   80 FORMAT(/"EDIT MODE 2--USE WITH CAUTION"//
     +" CORRECTIONS--ENTER ITEM NUMBER, NEW VALUE",
     +"--BLANK TO EXIT."//"EDT>_")
   90 I = 0
      READ(LUUC,*) I,R
      IF (I.EQ.0) RETURN
      IF (I.GE.26 .AND. I.LE.31) GO TO 110
      IF (I.GT.34) GO TO 100
      IBUF(I) = IFIX(R)
      WRITE(LUUC,62)
      GO TO 90
  100 I = I - 17
      BUF(I) = R
      WRITE(LUUC,62)
      GO TO 90
  110 WRITE(LUUC,120)
  120 FORMAT(/"EDITING LABELS NOT ALLOWED")
      RETURN
      END
```

```
C
C...SUBROUTINE GET
C
      SUBROUTINE GET(ILOC,IRC,IERR,II)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IERR = 0
      IF (ILOC.GT.4) GO TO 10
      IRC = ILOC
      RETURN
10    LOC = 3*ILOC - 2
      IRC = II
      CALL READF(IDATA,IERR,ISPEC(1,IRC),384,LEN,LOC)
      RETURN
      END
```

```
C
C...SUBROUTINE PUT
C
      SUBROUTINE PUT(IRC,ILOC,IERR)
      DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(384,6)
      COMMON IPLOT,IBASE,IDATA,ISPEC
      IF (IRC.GT.6) GO TO 30
      IERR = 0
      IF (ILOC.GT.4) GO TO 20
      IF (IRC.EQ.ILOC) RETURN
      DO 10 I = 1,384
10    ISPEC(I,ILOC) = ISPEC(I,IRC)
      RETURN
20    LOC = 3*ILOC - 2
      CALL WRITF(IDATA,IERR,ISPEC(1,IRC),384,LOC)
      RETURN
30    IERR = -12
      RETURN
      END
```

```
0001  C
0002  C...PROGRAM VOY7   LAST EDIT: 5 JUNE 1980 KKD
0003  C
0004  FTN4
0005        PROGRAM VOY7(5), VERSION 060580 KKD
0006        DIMENSION IPLOT(184),IBASE(16),IDATA(16),ISPEC(2304),SPEC(1152),
0007       +          INSQ(4,32),LOG(4,64),MISC(128),IPARM(5),LAB1(3,5),
0008       +          LAB2(5,3),LAB3(3),LAB4(3)
0009        EQUIVALENCE (ISPEC,SPEC)
0010        COMMON IPLOT,IBASE,IDATA,ISPEC,INSQ,LOG,MISC,IPARM,IVDATA(3)
0011        EQUIVALENCE (MISC(1),INST),    (MISC(2),I1),      (MISC(3),I2,R),
0012       +            (MISC(4),I3),      (MISC(5),MODE),    (MISC(6),JNST),
0013       +            (MISC(7),IPC),     (MISC(8),IPS),     (MISC(9),IPCOF),
0014       +            (MISC(10),JPC),    (MISC(11),IPEND),  (MISC(12),LUUC),
0015       +            (MISC(13),LUIN),   (MISC(14),LUOUT),  (MISC(15),LUTP),
0016       +            (MISC(16),LUPLT),  (MISC(17),ICRB),   (MISC(18),ICRD),
0017       +            (MISC(19),IVBAS),  (MISC(20),IVDAT),  (MISC(21),ILC),
0018       +            (MISC(22),IHDF),   (MISC(23),ICONW),  (MISC(24),ISTAT),
0019       +            (MISC(25),ITLOG),  (MISC(26),IERR),   (MISC(27),LEN),
0020       +            (MISC(28),IPRNT),  (MISC(29),MPLOT),  (MISC(30),IXORG),
0021       +            (MISC(31),IXEND),  (MISC(32),IRBOT),  (MISC(33),IRTOP),
0022       +            (MISC(34),IOPEN),  (MISC(35),IAXIS),  (MISC(36),IFRST),
0023       +            (IHDR(1),IFX1),    (IHDR(2),IFX2),    (IHDR(3),IFX3),
0024       +            (IHDR(4),IFX4),    (IHDR(5),ISCF),    (IHDR(6),IACF),
0025       +            (IHDR(7),IPOSF),   (IHDR(8),IDQF),    (IHDR(9),MCORD),
0026       +            (IHDR(10),IDUMP),  (IHDR(11),IIBN),   (IHDR(12),LBIN),
0027       +            (IHDR(13),MROW),   (IHDR(14),ITYPE),
0028       +                               (MISC(56),IT5),    (MISC(57),IT4),
0029       +            (MISC(58),IT3),    (MISC(59),IT2),    (MISC(60),IT1)
0030        EQUIVALENCE (MISC(24),STAT),   (MISC(61),FDSC),   (MISC(63),DQF),
0031       +            (MISC(65),YSEP),   (MISC(67),YLEN),   (MISC(69),YSC),
0032       +            (MISC(71),XSC),    (MISC(73),YORG),   (MISC(75),OFSET),
0033       +            (MISC(77),XMIN),   (MISC(79),XMAX),   (MISC(81),YMIN),
0034       +            (MISC(83),YMAX),   (MISC(85),FY1),    (MISC(87),FY2),
0035       +            (IHDR(15),RMIN1),  (IHDR(17),RMIN2),  (IHDR(19),RINC1),
0036       +            (IHDR(21),RINC2),  (IHDR(23),RA0),    (IHDR(25),DEC0),
0037       +            (MISC(101),SUM1),  (MISC(103),SUM2),  (MISC(105),RT1),
0038       +            (MISC(107),RT2),   (MISC(109),REF)
0039        EQUIVALENCE (IHDR(27),NTOTL),  (IHDR(28),NB)
0040        DIMENSION BIN(10,30),ROW(10),COL(30),NBIN(10,30),NROW(10),
0041       +          NCOL(30),IHDR(100),IBIN(1152),TOP(10)
0042        EQUIVALENCE (IBIN(1),BIN),     (IBIN(601),NBIN),  (IBIN(901),ROW),
0043       +            (IBIN(921),NROW),  (IBIN(931),COL),   (IBIN(991),NCOL),
0044       +            (IBIN(1021),IHDR)
0045        EQUIVALENCE (ISPEC(385),IBIN)
0046        EQUIVALENCE (SPEC(30),AZ),    (SPEC(31),EL),   (SPEC(32),RA),
0047       +            (SPEC(33),DEC),   (SPEC(38),L),    (SPEC(43),W),
0048       +            (SPEC(46),A),     (SPEC(47),B),    (SPEC(48),C),
0049       +            (SPEC(49),D)
0050        REAL L
0051        INTEGER OVRLAY(3),TOTBIN
0052        LOGICAL INIT,OUT,LOCK
0053        DATA LUPRNT/6/
0054        DATA OVRLAY/2HVO,2HY5,2H  /
```

```
      DATA LOCK/.FALSE./
      DATA LAB1/2H S,2HUM,2H1 ,
     +          2H S,2HUM,2H2 ,
     +          2HSU,2HM1,2H+2,
     +          2HDI,2HF1,2H-2,
     +          2HRA,2HT1,2H-2/
      DATA LAB2/5*2H  ,
     +          2HCO,2HRR,2HEC,2HTE,2HD ,
     +          2HCO,2HRR,2HEC,2HTE,2HD /
      DATA LAB3/2H L,2HAZ,2HRA/
      DATA LAB4/2H W,2HEL,2HDE/
C
C...INITIALIZATION
C
      INIT = I3.GE.0
      OUT = MOD(I3,2).EQ.0
      CALL WRITF(IDATA,IERR,ISPEC,1536,1)
      I1 = I1 + I2 - 1
      IF (INIT) GO TO 40
      CALL READF(IBASE,IERR,IBIN,1152,LEN,24)
      GO TO 100
C
C...CLEAR ARRAYS
C
40    DO 50 I = 1,1020
50    IBIN(I) = 0
      DO 60 I = 37,50
60    IHDR(I-36) = MISC(I)
      DO 70 I = 89,100
70    IHDR(I-74) = MISC(I)
      NTOTL = 0
      NB = 0
C
C...DUMP HEADER
C
100   IT1 = IFX1 - 60
      IT2 = IFX2 - 60
      IT3 = IFX3 - 60
      IT4 = IFX4 - 60
      IT5 = 2HN
      IT6 = 2HN
      IF (IAND(IDQF,7000B).EQ.0) IT5 = 2HFF
      IF (IAND(IDQF,700B).EQ.0) IT6 = 2HFF
      IF (IDUMP.EQ.0) GO TO 200
      CALL LURQ(1,LUPRNT,1)
      LOCK = .TRUE.
      IF (IDUMP.EQ.LUPRNT) WRITE(LUPRNT,105)
105   FORMAT(1H1///)
      WRITE(IDUMP,110) MCORD
110   FORMAT(/"  COORDINATE SYSTEM =",I2/7X,"1=L/W"/7X,"2=AZ/EL"
     +       /7X,"3=RA/DEC")
      IF (MCORD.EQ.1) WRITE(IDUMP,115) RA0,DEC0
115   FORMAT(/"  FIXED RA AND DEC =",2F10.3)
      WRITE(IDUMP,120) RMIN1,RMIN2,RINC1,RINC2,IT1,IT2,IT3,IT4,ITYPE
```

```
0109  120   FORMAT(/"  MINIMUMS =", 2F10.3
0110       +       //"  INCREMENTS =",2F10.3
0111       +       //"  REGION 1 =  CHANNEL",I4," TO CHANNEL",I4
0112       +        /"  REGION 2 =  CHANNEL",I4," TO CHANNEL",I4
0113       +       //"  BIN TYPE =",I2,"  WHERE:"
0114       +        /7X,"1=SUM CHANNELS ON REGION 1"
0115       +        /7X,"2=SUM CHANNELS ON REGION 2"
0116       +        /7X,"3=SUM CHANNELS ON REG1 AND REG2"
0117       +        /7X,"4=DIFFERENCE OF REG1 MINUS REG2"
0118       +        /7X,"5=RATIO OF REG1 TO REG2")
0119        WRITE(IDUMP,130) ISCF,IT5,IT6
0120  130   FORMAT(/"  SPACECRAFT ID FLAG =",I3
0121       +       //"  DATA QUALITY FLAG:  SPECTRAL     = 0",A2
0122       +        /22X,"POSITIONAL  = 0",A2/)
0123  C
0124  C...BIN
0125  C
0126  200   IF (I2.LT.1 .OR. I1.LT.I2) GO TO 302
0127        IF(IDUMP .NE. 0)WRITE(IDUMP,202)(LAB2(I,MCORD),I=1,5),
0128       +                                 (LAB1(I,ITYPE),I=1,3),
0129       +                                 LAB3(MCORD),LAB4(MCORD)
0130  202   FORMAT(95X,"DELTAS",8X,5A2,6X,"BIN"/2X,"NO.",5X,"FDSC",
0131       +        3X,"#",2X,"STAT",5X,"SUM1",5X,
0132       +        "SUM2",6X,3A2,4X,"AZ",7X,"EL",6X,"RA",5X,"DEC",
0133       +        7X,"L",7X,"W",7X,A2,5X,A2,5X,"COORDS"//)
0134        DO 300 I = I2,I1
0135        LOC = 3*I - 2
0136        CALL READF(IDATA,IERR,ISPEC,384,LEN,LOC)
0137        IF (IERR.LT.0) GO TO 500
0138        SUM1 = 0.0
0139        DO 205 J = IFX1,IFX2
0140  205   SUM1 = SUM1 + SPEC(J)
0141        SUM2 = 0.0
0142        DO 210 J = IFX3,IFX4
0143  210   SUM2 = SUM2 + SPEC(J)
0144        GO TO (212,214,216,218,220), ITYPE
0145  212   RB = SUM1
0146        GO TO 225
0147  214   RB = SUM2
0148        GO TO 225
0149  216   RB = SUM1 + SUM2
0150        GO TO 225
0151  218   RB = SUM1 - SUM2
0152        GO TO 225
0153  220   IF (SUM2.EQ.0.0) GO TO 222
0154        RB = SUM1/SUM2
0155        GO TO 225
0156  222   RB = 0.0
0157  C
0158  C...DETERMINE BIN
0159  C
0160  225   GO TO (230,232,234), MCORD
0161  230   DET = A*D - B*C
0162        IF (DET.EQ.0.0) GO TO 300
```

```
      DRA = RA - RA0
      DDEC = DEC - DEC0
      R1 = L + (D*DRA - B*DDEC)/DET
      R2 = W + (A*DDEC - C*DRA)/DET
      GO TO 240
232   SINEL = SIN(0.01745329*EL)
      IF (SINEL.EQ.0.0) GO TO 300
      R1 = AZ + L/SINEL
      R2 = EL - W
      GO TO 240
234   R1 = RA + A*L + B*W
      R2 = DEC + C*L + D*W
240   IB1 = (R1 - RMIN1)/RINC1 + 1.0
      IB2 = (R2 - RMIN2)/RINC2 + 1.0
      IF (IDUMP.EQ.0) GO TO 260
      WRITE(IDUMP,250) I,SPEC(18),ISPEC(1),ISPEC(3),SUM1,SUM2,RB,
     +                 AZ,EL,RA,DEC,L,W,R1,R2,IB1,IB2
250   FORMAT(I6,F10.0,I2,2X,O4,3(2X,F8.3),8F8.2,2I5)
C
C...FORM SUMS
C
260   IF (ISCF.NE.0 .AND. ISPEC(2).NE.ISCF) GO TO 300
      ISTAT = IAND(ISPEC(3),7577B)
      IF (IAND(ISTAT,IDQF).NE.0) GO TO 300
      IF (IB1.LT.1 .OR. IB1.GT.10) GO TO 300
      IF (IB2.LT.1 .OR. IB2.GT.30) GO TO 300
      NBIN(IB1,IB2) = NBIN(IB1,IB2) + 1
      NROW(IB1) = NROW(IB1) + 1
      NCOL(IB2) = NCOL(IB2) + 1
      NB = NB + 1
      BIN(IB1,IB2) = BIN(IB1,IB2) + RB
      ROW(IB1) = ROW(IB1) + RB
      COL(IB2) = COL(IB2) + RB
300   NTOTL = NTOTL + 1
302   CALL WRITF(IBASE,IERR,IBIN,1152,24)
      IF (.NOT.OUT) GO TO 400
C
C...COMPUTE OVERALL AVERAGE
C
      TOTBIN = 0
      TSUM = 0.0
      DO 305 I = 1,300
      TOTBIN = NBIN(I) + TOTBIN
305   TSUM = BIN(I) + TSUM
      AVG = TSUM/TOTBIN
C
C...NORMALIZE BINS
C
      IF(.NOT. LOCK)CALL LURQ(1,LUPRNT,1)
      LOCK = .TRUE.
      DO 310 I = 1,300
      IF (NBIN(I).EQ.0) GO TO 310
      BIN(I) = BIN(I)/NBIN(I)
310   CONTINUE
```

```
      DO 320 I = 1,10
      IF (NROW(I).EQ.0) GO TO 320
      ROW(I) = ROW(I)/NROW(I)
320   CONTINUE
      DO 330 I = 1,30
      IF (NCOL(I).EQ.0) GO TO 330
      COL(I) = COL(I)/NCOL(I)
330   CONTINUE
      RT1 = RMIN1 + RINC1/2.0
      DO 340 I = 1,10
      TOP(I) = RT1
340   RT1 = RT1 + RINC1
C
C...OUTPUT RESULTS
C
      WRITE(LUPRNT,105)
      WRITE(LUPRNT,110) MCORD
      IF (MCORD.EQ.1) WRITE(LUPRNT,115) RA0,DEC0
      WRITE(LUPRNT,120) RMIN1,RMIN2,RINC1,RINC2,IT1,IT2,IT3,IT4,ITYPE
      WRITE(LUPRNT,130) ISCF,IT5,IT6
      WRITE(LUPRNT,105)
      GO TO (350,352,354), MCORD
350   IT1 = 2H L
      IT2 = 2H W
      GO TO 360
352   IT1 = 2HAZ
      IT2 = 2HEL
      GO TO 360
354   IT1 = 2HRA
      IT2 = 2HDE
360   WRITE(LUPRNT,365) NTOTL,NB,IT1,TOP,IT2,(I,I=1,10)
365   FORMAT(/10X,"TOTAL SPECTRA =",I6,5X,"TOTAL BINNED =",I6
     +       ///13X,A2,10F10.3//7X,A2,3X,10I10/)
      RT2 = RMIN2 + RINC2/2.0
      DO 370 I = 1,30
      WRITE(LUPRNT,368) RT2,I,(NBIN(J,I),J=1,10),NCOL(I)
368   FORMAT(1X,F10.3,I4,I9,10I10)
370   RT2 = RT2 + RINC2
      WRITE(LUPRNT,375) NROW
375   FORMAT(/14X,10I10)
      WRITE(LUPRNT,105)
      WRITE(LUPRNT,365) NTOTL,NB,IT1,TOP,IT2,(I,I=1,10)
      RT2 = RMIN2 + RINC2/2.0
      DO 380 I = 1,30
      WRITE(LUPRNT,378) RT2,I,(BIN(J,I),J=1,10),COL(I)
378   FORMAT(1X,F10.3,I4,11(1X,F9.3))
380   RT2 = RT2 + RINC2
      WRITE(LUPRNT,385) ROW
385   FORMAT(/15X,10(1X,F9.3))
      WRITE(LUPRNT,390)AVG
390   FORMAT(//10X,"OVERALL AVERAGE = ",F9.3)
      WRITE(LUPRNT,105)
C
C...EXIT FROM OVERLAY
```

```
C
400     CALL READF(IDATA,IERR,ISPEC,1536,LEN,1)
        IF(LOCK)CALL LURQ(0,LUPRNT,1)
        LOCK = .FALSE.
        IPARM(1) = 0
        CALL EXEC(8,OVRLAY)
        CALL VOY
C
C...FMGR ERROR
C
500     WRITE(LUUC,510) IERR,I
510     FORMAT("  FILE MANAGER ERROR",I4,"  AT LOCATION",I5)
        LUIN = LUUC
        MODE = 1
        GO TO 302
        END
```

ITEMS 4A & B

CALIBRATION SPECTRUM FOR VOYAGER I INSTRUMENT.

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
0.	0	0	0007	0/ 0	0: 0: 0.000	0: 0: 0.000	0	0.00

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	11	0	0	0	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP
		0.00	0.00	0.00	0.00	0.0	0.00

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
0	CALSP		0	0.00	0.00	0.000	0.000

TRANSLATION MATRIX

A	B	C	D
0.0000	0.0000	0.0000	0.0000

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	0.00	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000
OC	0.00	0.00			0.000	0.000	0.000	0.000	0.000

				0.00	0.00	0.00	0.00	0.00	0.00
1.33	1.36	1.38	1.39	1.42	1.44	1.46	1.50	1.54	1.57
1.63	1.66	1.71	1.80	1.89	1.98	2.08	2.20	2.34	2.45
2.57	2.67	2.78	2.90	2.92	2.99	3.01	3.06	3.08	3.09
3.10	3.13	3.14	3.18	3.19	3.19	3.21	3.22	3.23	3.25
3.30	3.37	3.40	3.46	3.54	3.69	3.79	3.93	4.15	4.29
4.44	4.65	4.86	5.07	5.37	5.54	5.87	6.17	6.63	7.10
7.71	8.31	9.00	9.82	10.59	11.37	12.41	13.33	14.69	16.61
18.30	20.45	22.88	24.83	26.34	28.42	30.42	25.71	21.82	21.90
21.95	21.95	21.99	22.04	22.04	22.04	22.04	22.04	22.08	22.13
22.27	22.36	22.45	22.50	22.59	22.64	22.69	22.78	22.98	23.08
23.48	23.73	24.00	24.54	25.06	25.59	26.21	26.73	27.13	27.98
28.42	29.19	30.00	30.85	31.76	32.72	33.96	34.84	36.00	37.50
38.57	40.00	41.54	43.20	44.62	46.35	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
0.	0	0	0007	0/ 0	0: 0: 0.000	0: 0: 0.000	0	0.00

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	21	0	0	0	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP
	/	0.00	0.00	0.00	0.00	0.0	0.00

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
0	CALSP		0	0.00	0.00	0.000	0.000

TRANSLATION MATRIX	A	B	C	D
	0.0000	0.0000	0.0000	0.0000

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	0.00	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000
OC	0.00	0.00			0.000	0.000	0.000	0.000	0.000

				0.00	0.00	0.00	0.00	0.00	0.00
1.09	1.10	1.12	1.14	1.17	1.20	1.23	1.27	1.30	1.34
1.39	1.45	1.52	1.60	1.66	1.77	1.89	2.02	2.14	2.29
2.45	2.65	2.83	3.07	3.26	3.47	3.62	3.82	3.86	3.80
3.75	3.70	3.62	3.60	3.56	3.52	3.48	3.46	3.43	3.41
3.39	3.37	3.37	3.37	3.37	3.39	3.40	3.46	3.52	3.60
3.74	3.86	4.08	4.29	4.54	4.86	5.19	5.62	6.00	6.35
6.63	6.92	7.30	7.55	8.12	8.85	9.73	10.69	11.61	12.78
14.02	15.21	16.49	18.00	19.46	21.05	22.50	24.27	26.02	23.84
20.57	20.69	20.73	20.77	20.77	20.85	20.89	20.97	21.09	21.09
21.17	21.17	21.26	21.30	21.34	21.51	21.60	21.60	21.60	21.69
21.82	22.04	22.27	22.59	22.98	23.43	23.89	24.11	24.71	25.11
25.71	26.47	27.34	28.27	29.19	30.00	31.03	32.43	33.23	34.61
35.64	36.73	39.27	40.75	42.35	43.54	0.00	0.00	0.00	0.00
0.00	0.00								

CALIBRATION SPECTRUM FOR VOYAGER 2 INSTRUMENT.

Fixed Pattern Noise

Voyager 1, Pre-encounter

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
0.	0	0	0006	0/ 0	0: 0: 0.000	0: 0: 0.000	0	0.00

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
47	0	0	0	0	0	0	0.0	0.0	1.7	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP
/	0.00	0.00	0.00	0.00	0.0	0.00

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
0	FPNSP		0	0.00	0.00	0.000	0.000

TRANSLATION MATRIX

A	B	C	D
0.0000	0.0000	0.0000	0.0000

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	0.00	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000
OC	0.00	0.00			0.000	0.000	0.000	0.000	0.000

				0.00	0.00	0.00	0.00	0.00	0.00
1.03	1.75	8.74	10.20	1.69	1.73	1.88	1.66	1.96	1.47
2.34	1.76	1.62	1.96	1.98	1.79	1.38	1.13	1.80	1.80
1.83	1.65	1.84	1.72	1.99	1.58	2.20	1.88	1.98	1.80
1.58	1.91	1.47	1.02	1.76	2.00	1.59	1.68	1.88	1.69
1.92	1.69	2.18	1.76	1.91	1.79	1.86	1.96	1.39	.96
1.93	2.04	1.64	1.75	1.70	1.73	2.23	1.62	2.23	1.88
1.79	1.89	1.75	1.67	1.61	.96	1.80	2.08	1.62	1.65
2.02	1.70	1.82	1.57	2.07	1.88	2.11	1.68	1.76	1.73
1.35	1.20	1.91	1.93	1.57	1.81	1.87	1.68	1.91	1.58
1.89	1.78	1.85	1.78	1.97	1.89	1.63	1.13	1.89	2.00
1.54	1.70	1.90	1.73	1.78	1.72	1.98	1.76	1.69	1.94
1.80	1.67	1.42	1.13	1.70	1.91	1.72	1.60	1.65	1.60
1.82	1.23	1.98	1.51	1.42	1.29	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
0.	0	0	0006	0/ 0	0: 0: 0.000	0: 0: 0.000	0	0.00

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
49	0	0	0	0	0	0	0.0	0.0	1.8	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP
	0.00	0.00	0.00	0.00	0.0	0.00

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
0	FPNSP		0	0.00	0.00	0.000	0.000

TRANSLATION MATRIX

A	B	C	D
0.0000	0.0000	0.0000	0.0000

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	0.00	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000
OC	0.00	0.00			0.000	0.000	0.000	0.000	0.000

				0.00	0.00	0.00	0.00	0.00	0.00
.75	1.73	6.31	6.55	2.30	1.55	1.96	1.31	1.87	1.18
2.05	1.47	1.61	1.74	1.87	1.55	1.53	1.09	1.98	1.80
2.34	1.60	2.29	1.53	2.05	1.68	2.35	1.72	2.19	1.73
1.59	1.83	1.69	1.17	2.02	1.87	1.81	1.67	2.15	1.56
2.19	1.66	1.97	1.76	2.11	1.53	1.64	1.69	1.52	.94
1.89	1.69	1.53	1.59	1.80	1.56	2.52	1.50	2.31	1.83
1.89	2.02	1.72	1.37	1.80	.80	1.39	1.44	1.28	1.19
2.20	1.65	2.47	1.55	2.03	1.49	1.58	.66	.88	1.00
1.31	.95	1.81	1.51	1.29	1.25	1.46	1.29	2.16	1.51
2.05	1.78	2.02	1.90	2.02	1.92	1.84	1.19	2.06	2.04
1.96	2.04	2.27	1.79	1.81	1.82	1.87	1.73	1.88	1.93
1.83	1.37	1.52	1.01	1.83	1.79	1.94	1.49	1.53	1.35
1.97	1.13	1.89	1.41	1.64	1.26	0.00	0.00	0.00	0.00
0.00	0.00								

Fixed Pattern Noise
Voyager 1, Post - Encounter

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
0.	0	0	0006	0/ 0	0: 0: 0.000	0: 0: 0.000	0	0.00

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
76	0	0	0	0	0	0	0.0	0.0	1.5	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP
/	0.00	0.00	0.00	0.00	0.0	0.00

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
0	FPNSP		0	0.00	0.00	0.000	0.000

TRANSLATION MATRIX	A	B	C	D
	0.0000	0.0000	0.0000	0.0000

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	0.00	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000
OC	0.00	0.00			0.000	0.000	0.000	0.000	0.000

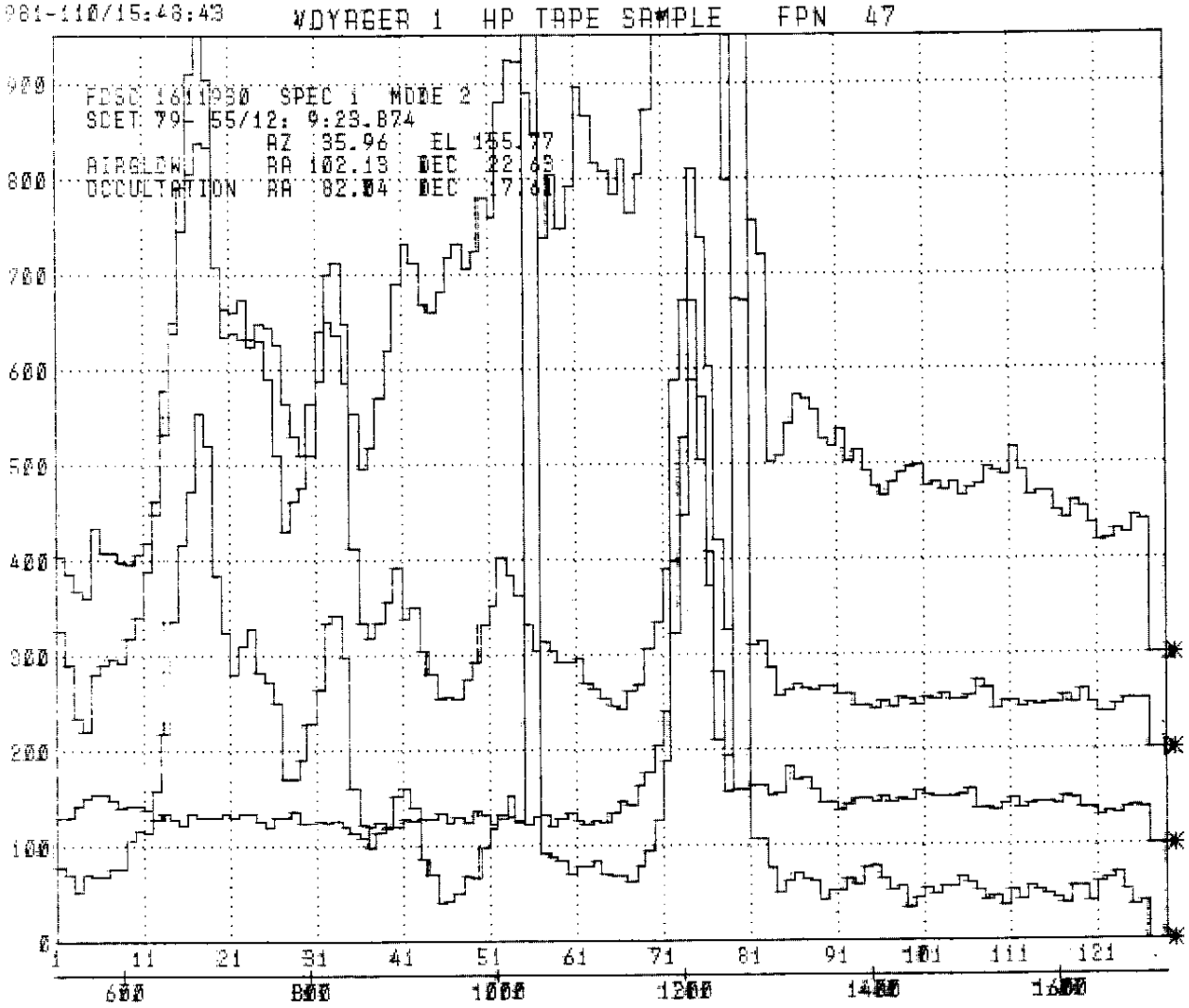
				0.00	0.00	0.00	0.00	0.00	0.00
1.22	1.23	7.06	8.25	1.62	1.66	1.40	1.45	1.60	1.40
1.83	1.54	1.53	1.67	1.55	1.77	1.50	1.07	1.53	1.67
1.41	1.66	1.52	1.54	1.65	1.51	1.83	1.64	1.67	1.77
1.67	1.70	1.14	1.00	1.55	1.74	1.46	1.55	1.48	1.54
1.50	1.50	1.79	1.65	1.51	1.57	1.62	1.77	1.12	.89
1.57	1.86	1.39	1.39	1.62	1.46	1.49	1.50	1.68	1.58
1.53	1.63	1.42	1.75	1.39	1.17	1.54	1.58	1.40	1.56
1.48	1.45	1.55	1.49	1.51	1.54	1.44	1.64	1.52	1.52
1.14	1.15	1.54	1.88	1.46	1.44	1.43	1.51	1.46	1.52
1.56	1.40	1.57	1.65	1.44	1.80	1.30	1.00	1.47	1.78
1.31	1.55	1.52	1.49	1.61	1.41	1.58	1.83	1.62	1.42
1.50	1.58	1.32	1.10	1.55	1.60	1.34	1.69	1.56	1.66
1.61	1.47	1.56	1.47	1.29	1.47	0.00	0.00	0.00	0.00
0.00	0.00								

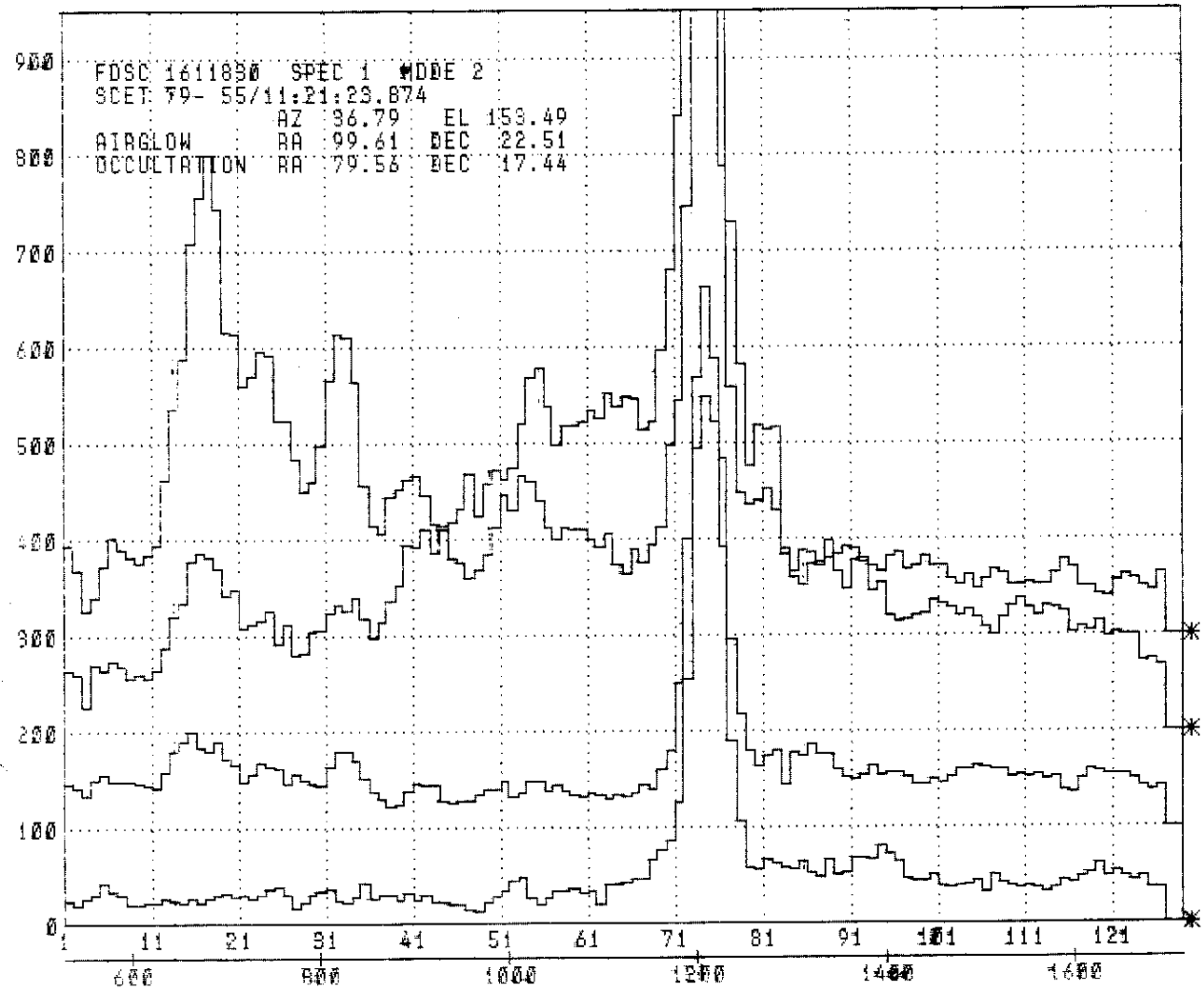
FIXED PATTERN NOISE
VOYAGER 2

HP EDR SAMPLE

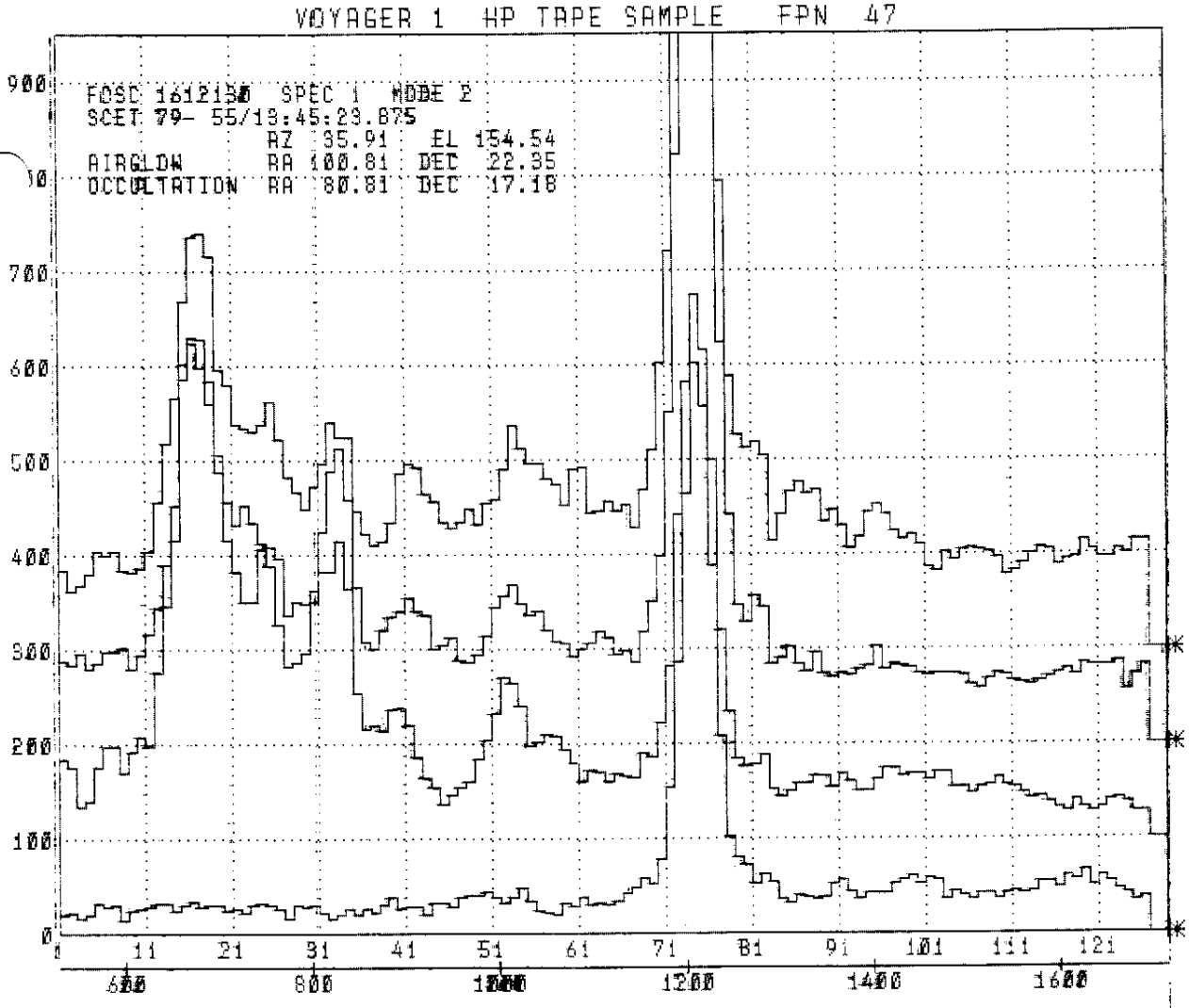
PSPEC

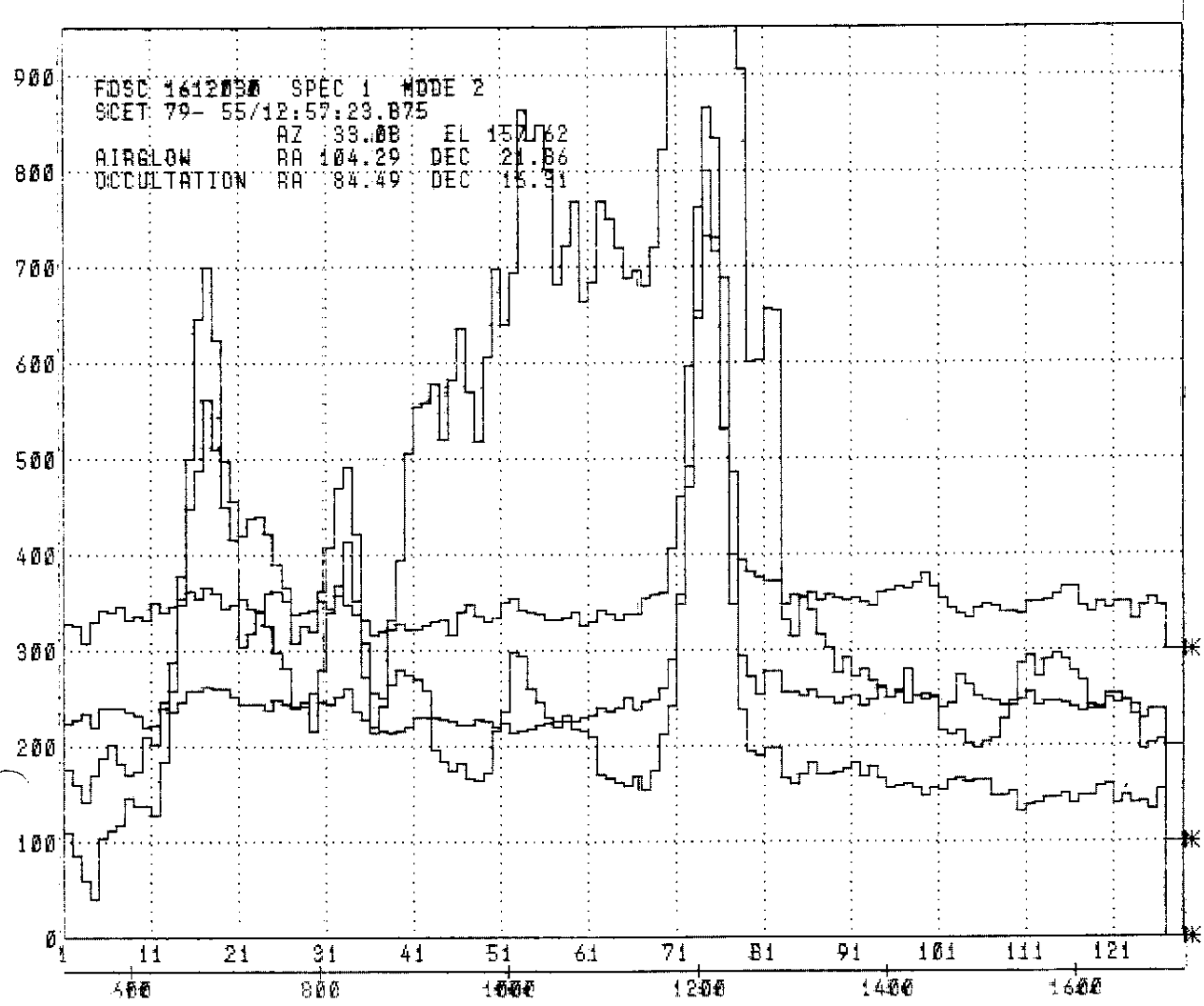
1981-110/15:48:43 VOYAGER 1 HP TAPE SAMPLE FPN 47



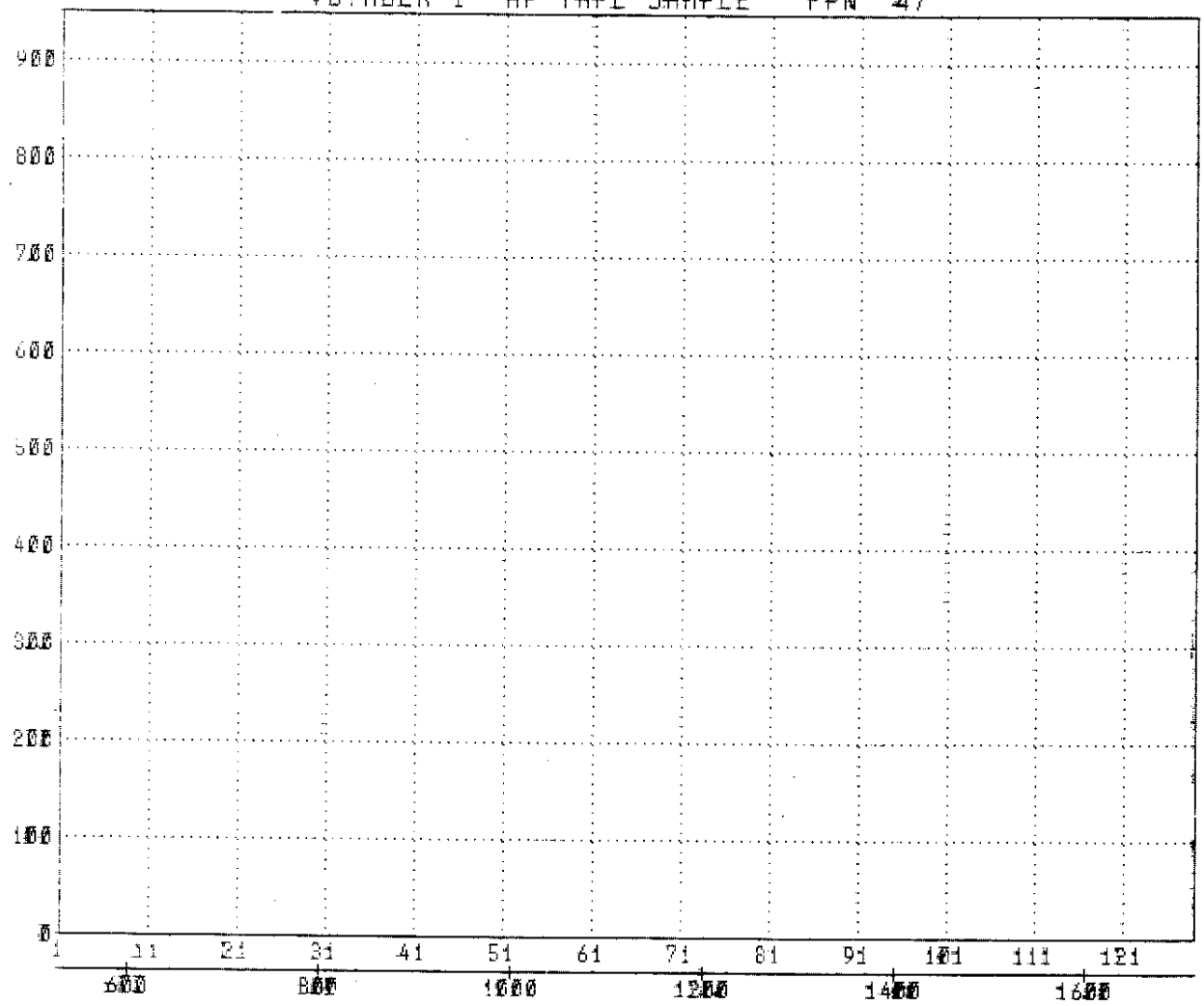


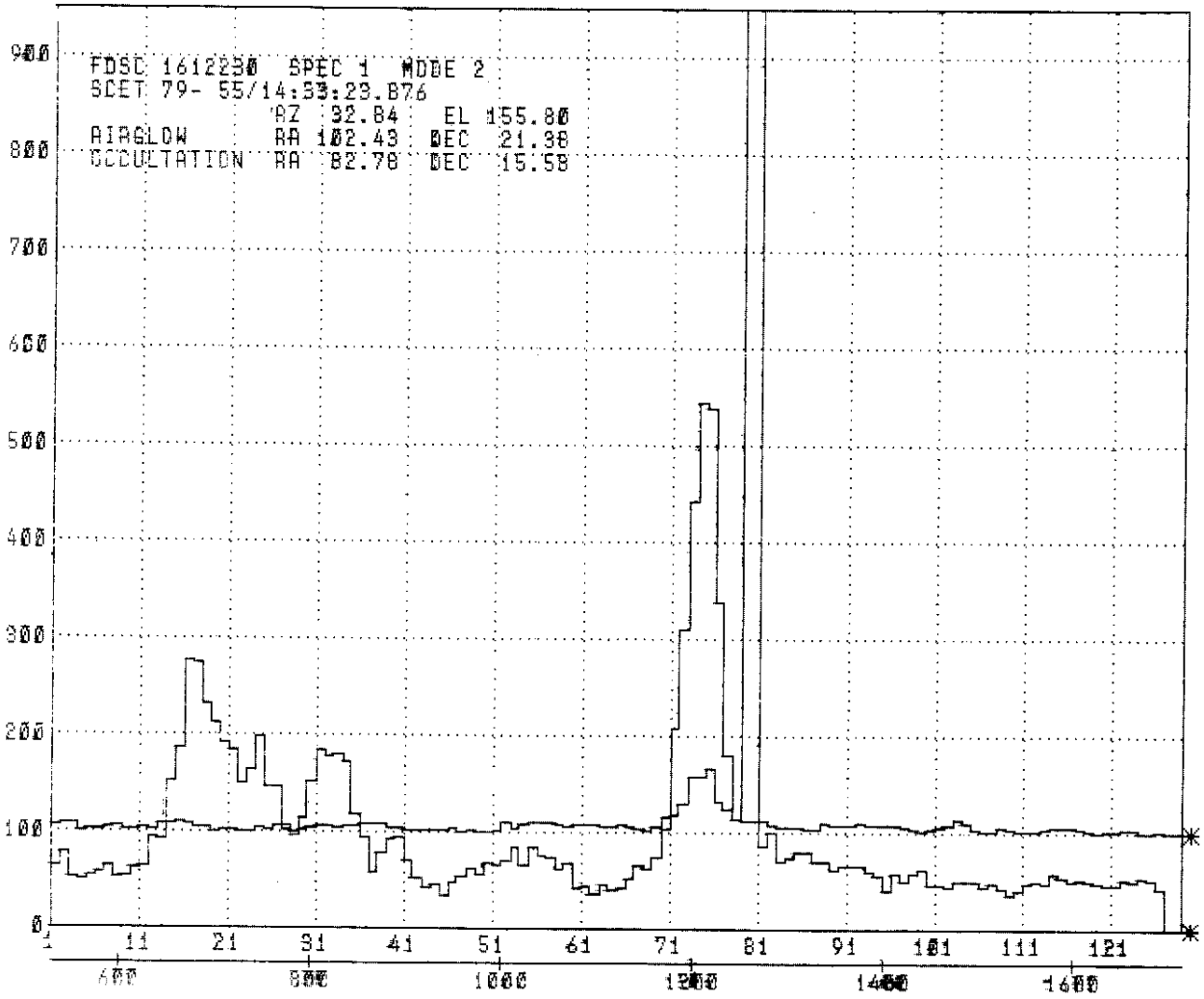
VOYAGER 1 HP TAPE SAMPLE FPN 47





1981-110/15:48:49 VOYAGER 1 HP TAPE SAMPLE FPN 47

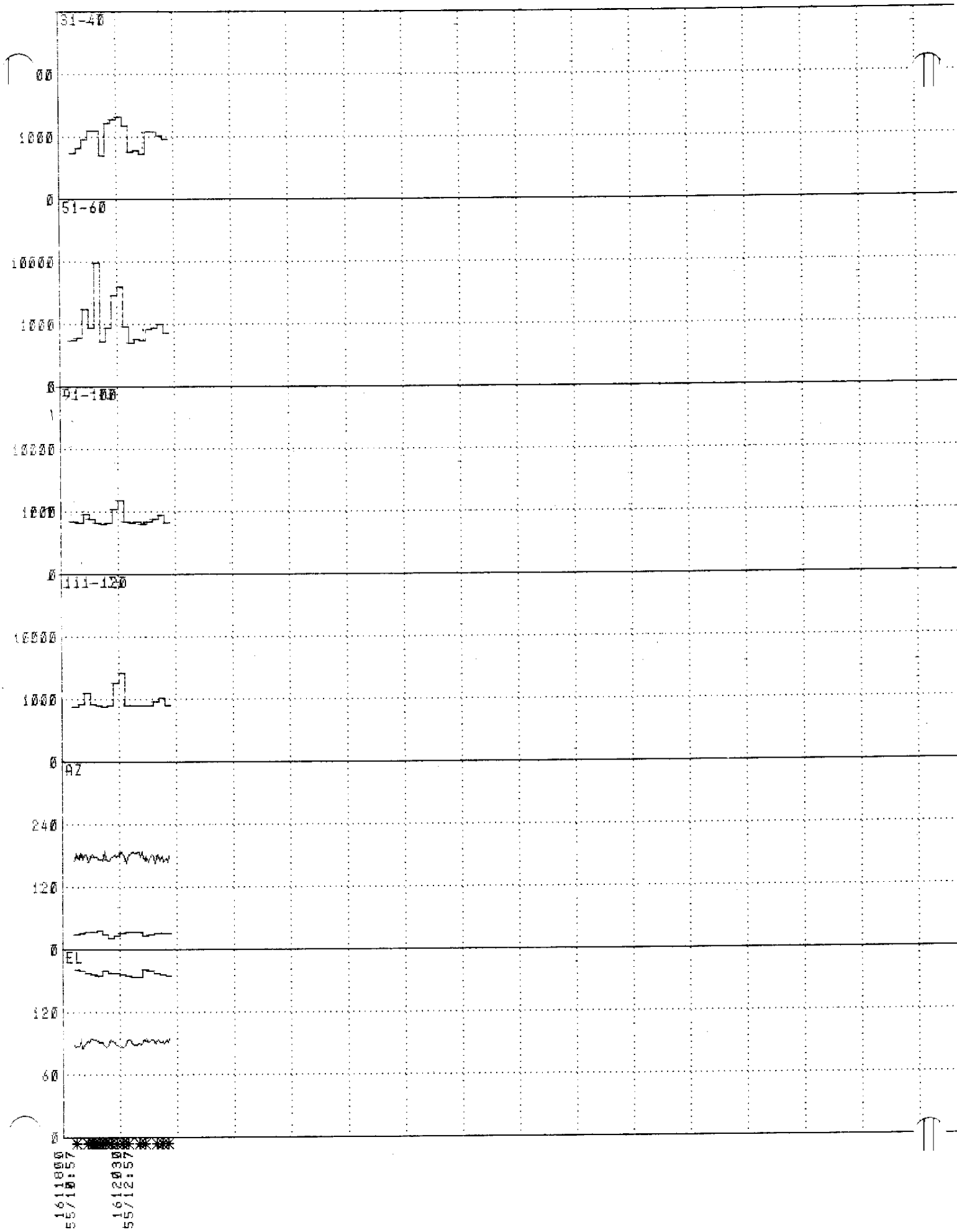




HPEDR SAMPLE PSUM

1981-110/15:43:48

VOYAGER 1 HP TAPE SAMPLE



HPLOG 4:05 PM MON., 20 APR., 1981

HP-EDR	SAMPLE (SOURCE TAPE #)	BEGINNING FDSC + SPECTRUM #	ENDING FDSC + SPECTRUM #	BEGINNING SCET	ENDING SCET	DATA MODE	# OF SPECTRA
	JPL-EDR 021158	1611830 1	1612246 12	79/055 11:21:23	79/055 14:46:54	GS-3	3216

TOTAL # SPECTRA = 3216

ZERO SPECTRA = 4

Printouts of first Ten spectra. (From Program VOY.)

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611330	1	1	0101	79/ 55	11:21:23.874	11:57:49.320	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/		0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1226	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1980	-1.0594	.9832	-.2022

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	99.61	22.51	36.79	153.49	-.041	-.041	-.040	-.040	-.040
OC	79.56	17.44			-.001	.001	.003	.005	.007

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	2.00	2.00	2.00
1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	1.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00	2.00	0.00	1.00	2.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

6 shift locations

Spectrum

6 shift locations

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	2	1	0101	79/ 55	11:21:27.714	11:57:53.160	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/		0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1227	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1926	-1.0614	.9841	-.1977

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	100.31	22.64	36.79	154.15	-.040	-.040	-.039	-.040	-.040
OC	80.21	17.64			.007	.009	.011	.012	.014

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.00	1.00	2.00	3.00	1.00	0.00	1.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	3	1	0101	79/ 55	11:21:31.554	11:57:57.000	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/	0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1228	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX

A	B	C	D
-.1872	-1.0633	.9850	-.1931

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	101.00	22.77	36.79	154.80	-.040	-.041	-.042	-.042	-.042
OC	80.86	17.84			.014	.015	.016	.018	.020

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
1.00	2.00	1.00	0.00	0.00	0.00	1.00	1.00	2.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	2.00	1.00	0.00	1.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00	0.00
0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	4	1	0101	79/ 55	11:21:35.394	11:58: .840	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/	0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1229	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX

A	B	C	D
-.1770	-1.0669	.9867	-.1846

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	102.28	22.99	36.79	155.99	-.042	-.042	-.041	-.041	-.041
OC	82.06	18.21			.020	.022	.024	.026	.028

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	2.00	2.00	2.00	2.00	1.00	2.00	1.00	0.00
0.00	0.00	0.00	3.00	2.00	2.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

2.00	[illegible]	4.00	5.00	4.00	5.00	0.00	0.00	1.00	1.00
0.00	[illegible]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	5	1	0101	79/ 55	11:21:39.234	11:58: 4.680	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/		0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1230	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1658	-1.0705	.9883	-.1755

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	103.65	23.21	36.79	157.28	-.041	-.041	-.040	-.040	-.038
OC	83.35	18.59			.028	.030	.032	.033	.034

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	2.00	2.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
2.00	2.00	2.00	4.00	2.00	1.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
1.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	6	1	0101	79/ 55	11:21:43.074	11:58: 8.520	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/		0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1231	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1543	-1.0740	.9899	-.1661

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	105.02	23.43	36.79	158.56	-.038	-.037	-.036	-.035	-.036
OC	84.64	18.97			.034	.035	.035	.037	.038

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	1.00	2.00	1.00	0.00	1.00	2.00	1.00	0.00	1.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	7	1	0101	79/ 55	11:21:46.914	11:58:12.360	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/	0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1232	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1557	-1.0730	.9897	-.1688

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	106.28	23.37	36.09	159.69	-.036	-.037	-.037	-.038	-.038
OC	85.92	18.86			.039	.040	.041	.042	.044

				0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00
1.00	0.00	1.00	1.00	2.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	8	1	0101	79/ 55	11:21:50.754	11:58:16.200	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
/	0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1233	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1662	-1.0691	.9882	-.1800

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	107.52	23.15	34.88	160.77	-.037	-.036	-.034	-.032	-.030
OC	87.25	18.45			.045	.046	.048	.049	.0[illegible]

				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00
0.00	0.00	0.00	1.00	0.00	1.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	9	1	0101	79/ 55	11:21:54.594	11:58:20.040	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
	/	0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1234	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1768	-1.0652	.9866	-.1913

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	108.74	22.96	33.67	161.85	-.029	-.029	-.028	-.028	-.027
OC	88.57	18.06			.051	.052	.053	.054	.055

				0.00	0.00	0.00	0.00	0.00	0.00
2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	2.00	1.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	3.00	2.00	2.00	1.00	1.00	0.00	1.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00								

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611830	10	1	0101	79/ 55	11:21:58.434	11:58:23.880	2	3.84

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
0	0	0	1	3	0	0	0.0	0.0	0.0	.10000E+01	0

COORD	MODE	MIN	MAX	MIN	MAX	HV	TEMP	SQRT
	/	0.00	0.00	0.00	0.00	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1235	308.96	-19.25	3.860	-4.992

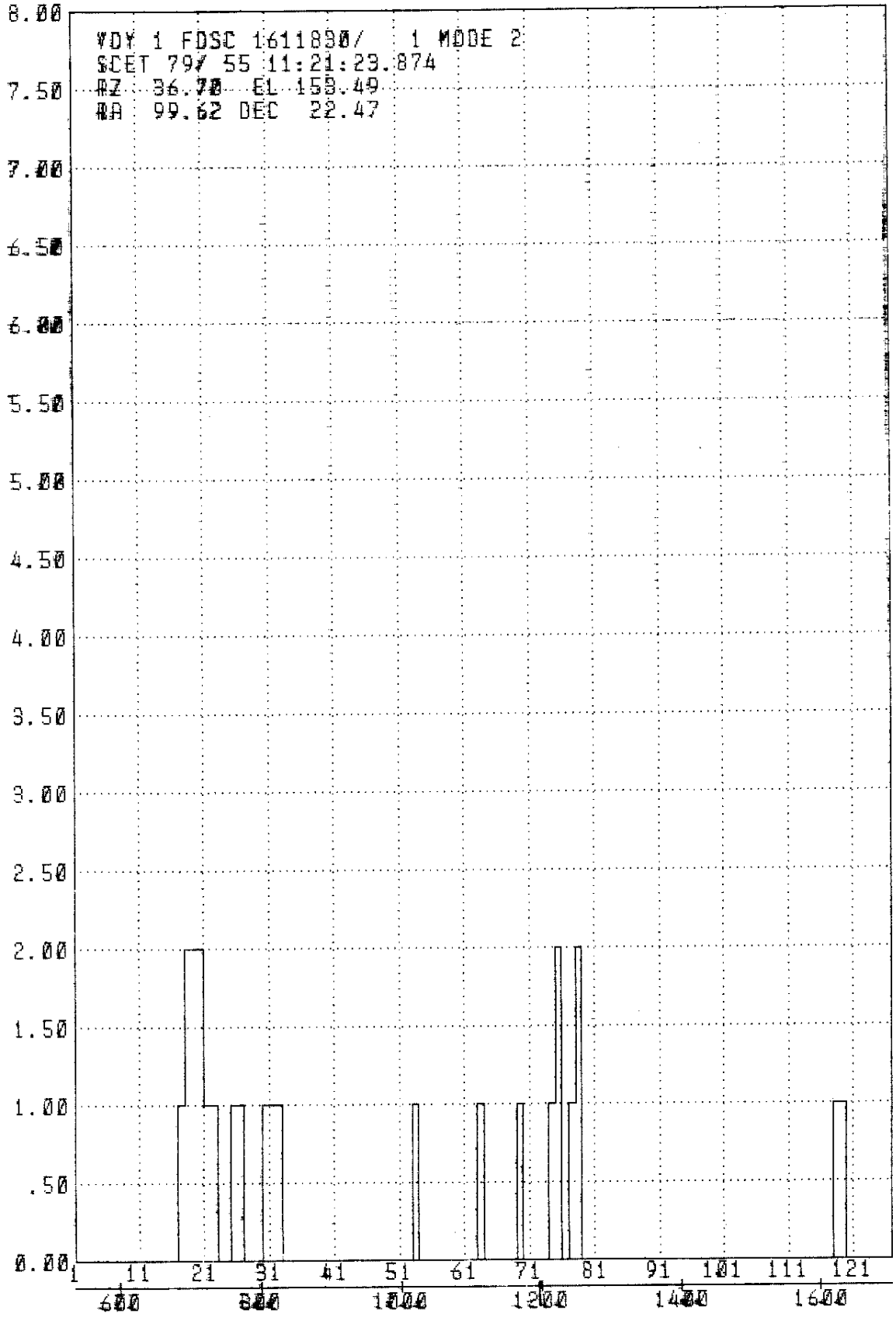
TRANSLATION MATRIX	A	B	C	D
	-.1846	-1.0625	.9854	-.1992

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	109.2	22.83	32.98	162.26	-.027	-.026	-.025	-.025	.022
OC	89.12	17.79			.055	.056	.057	.057	.057

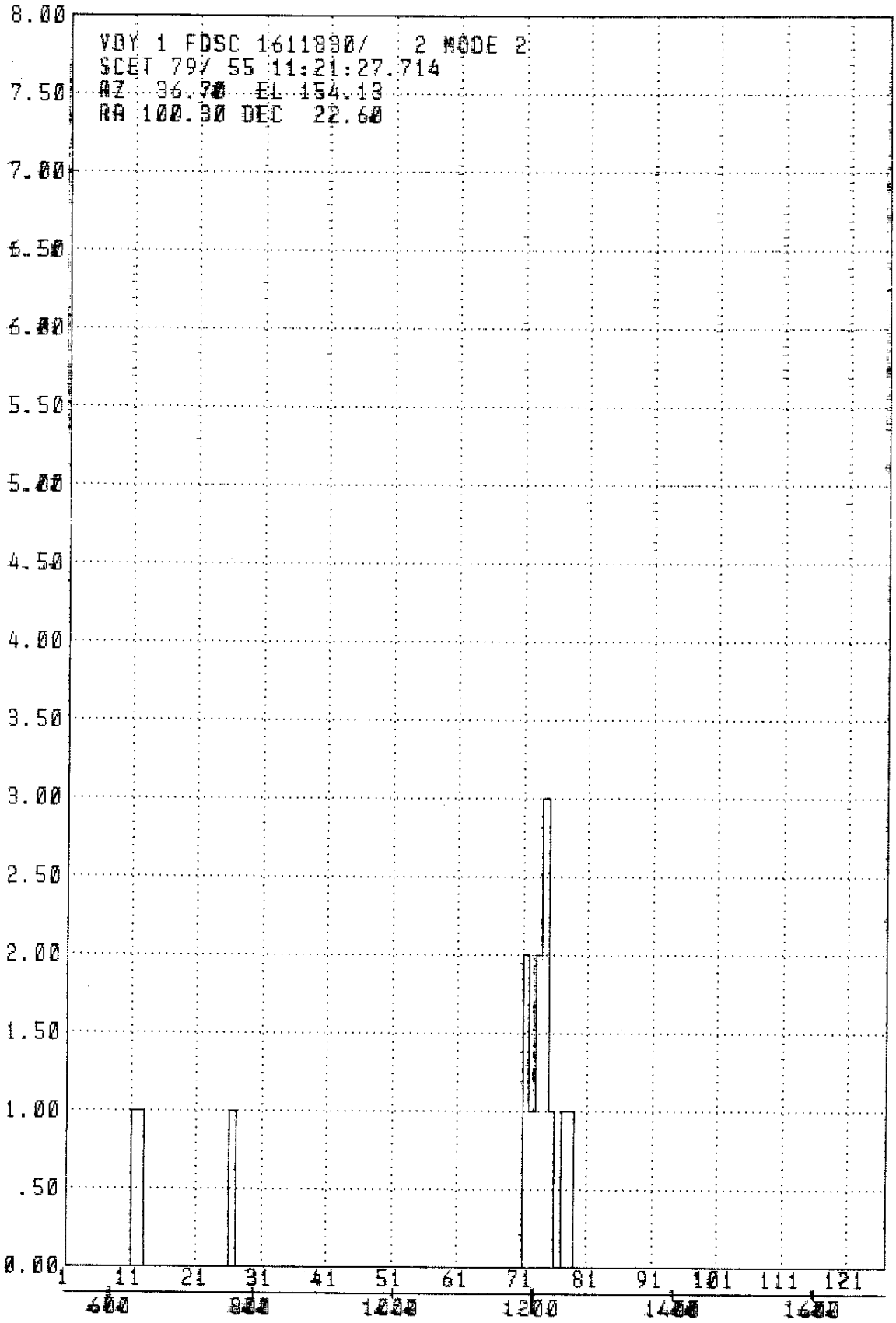
				0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
1.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	2.00	2.00	1.00	2.00	1.00	0.00	0.00	0.00
0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.00	1.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
0.00	0.00								

1981/111 10:58: 5

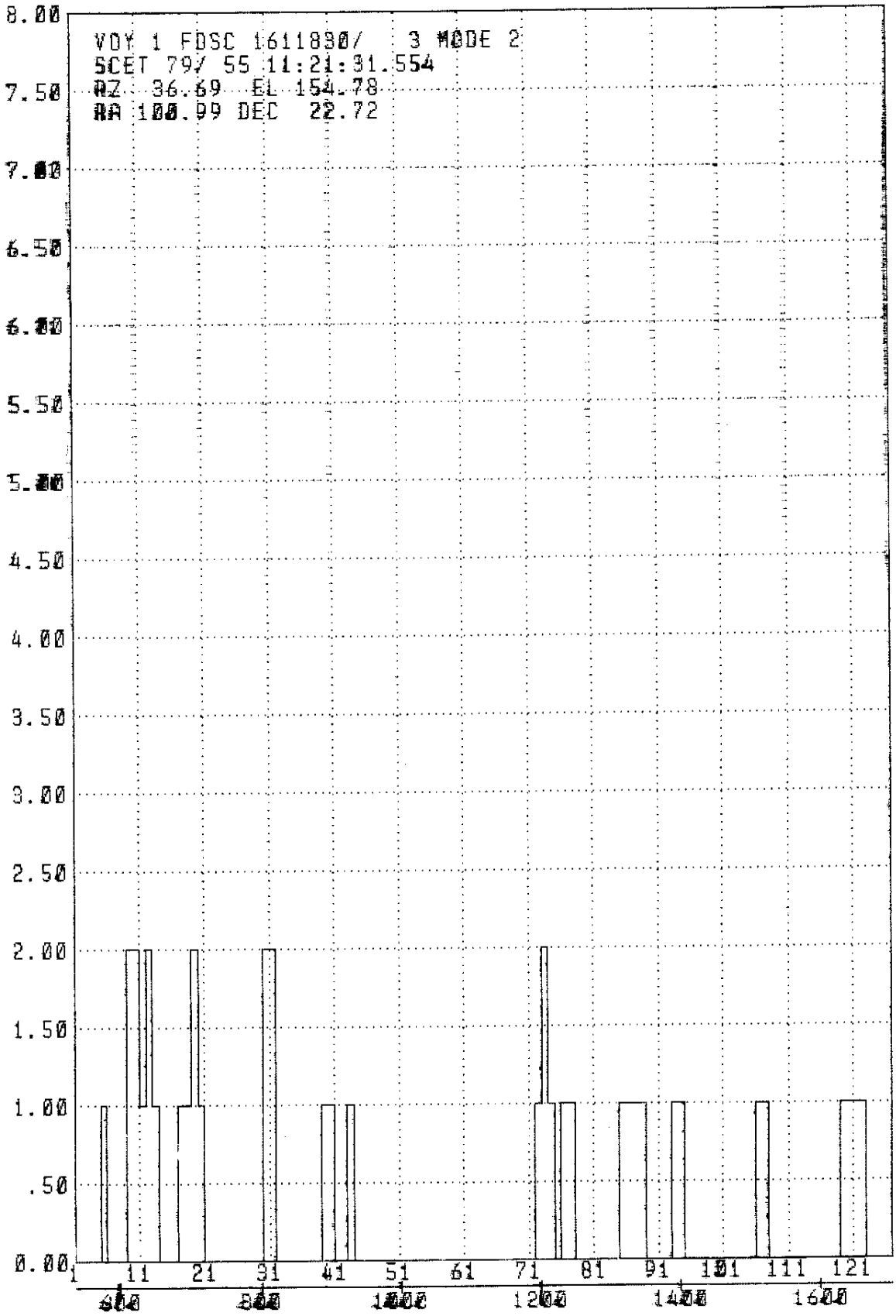
Plots from Program VOY
(First ten spectra)



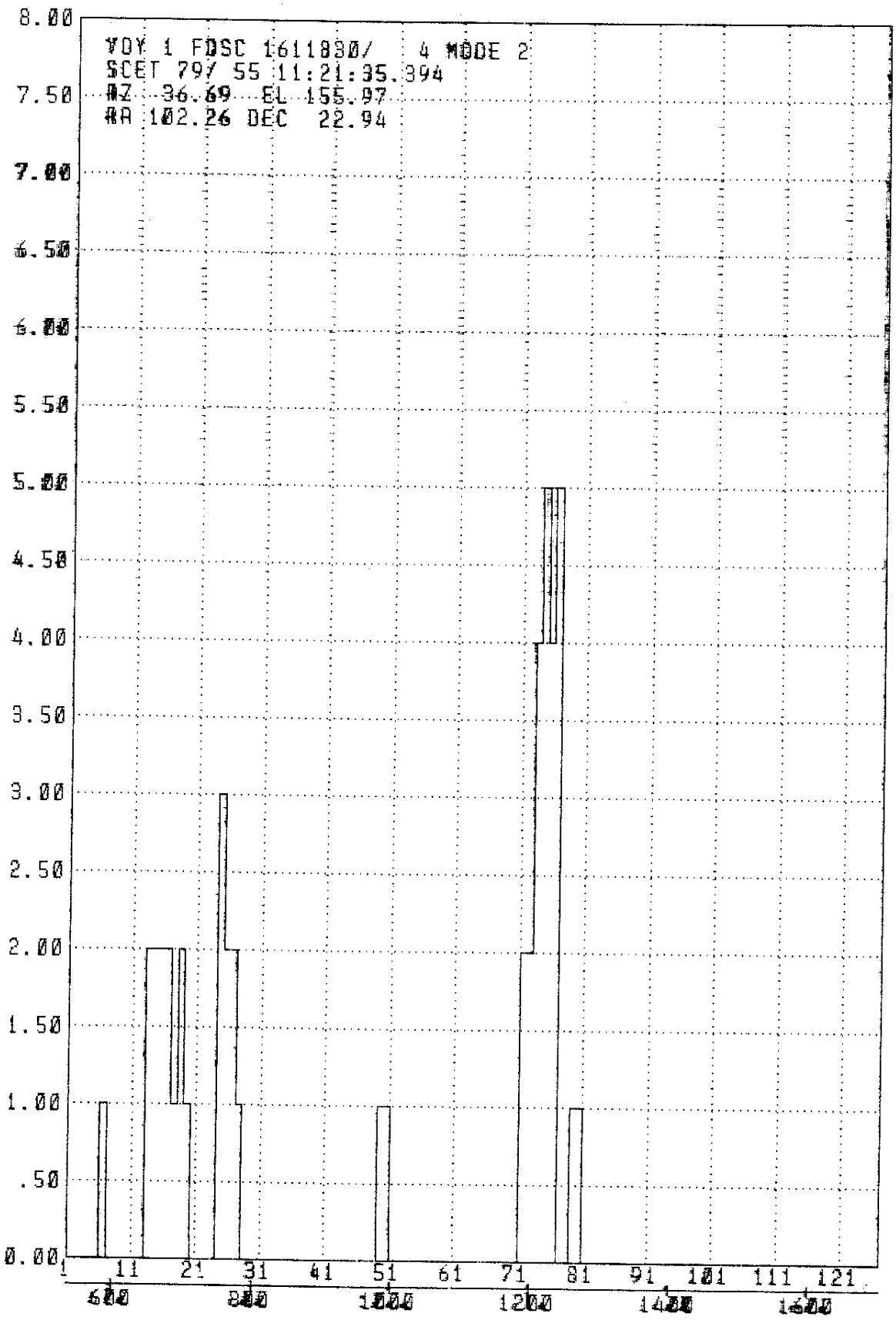
1981/111 10:58:34



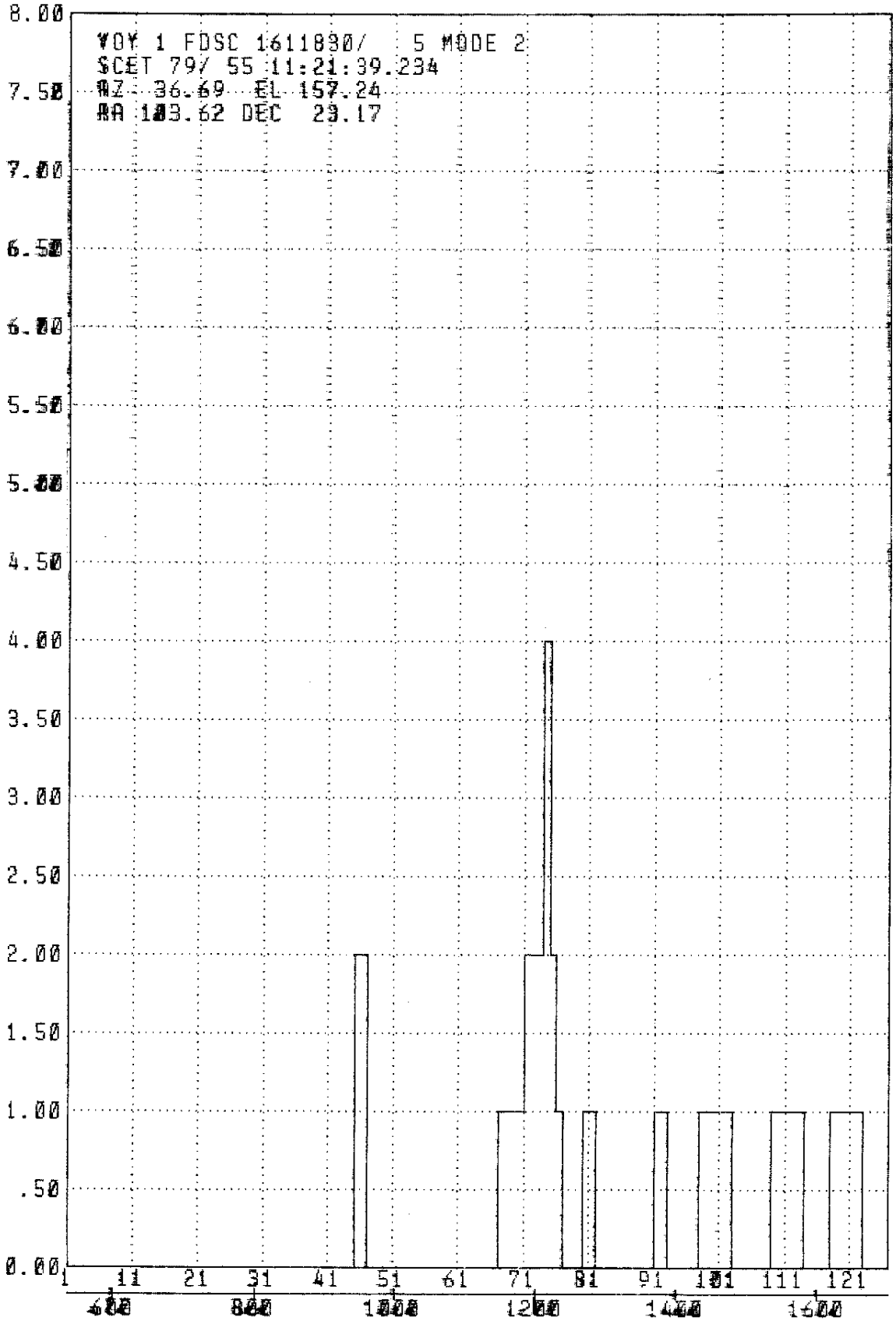
1981/111 10:59: 3



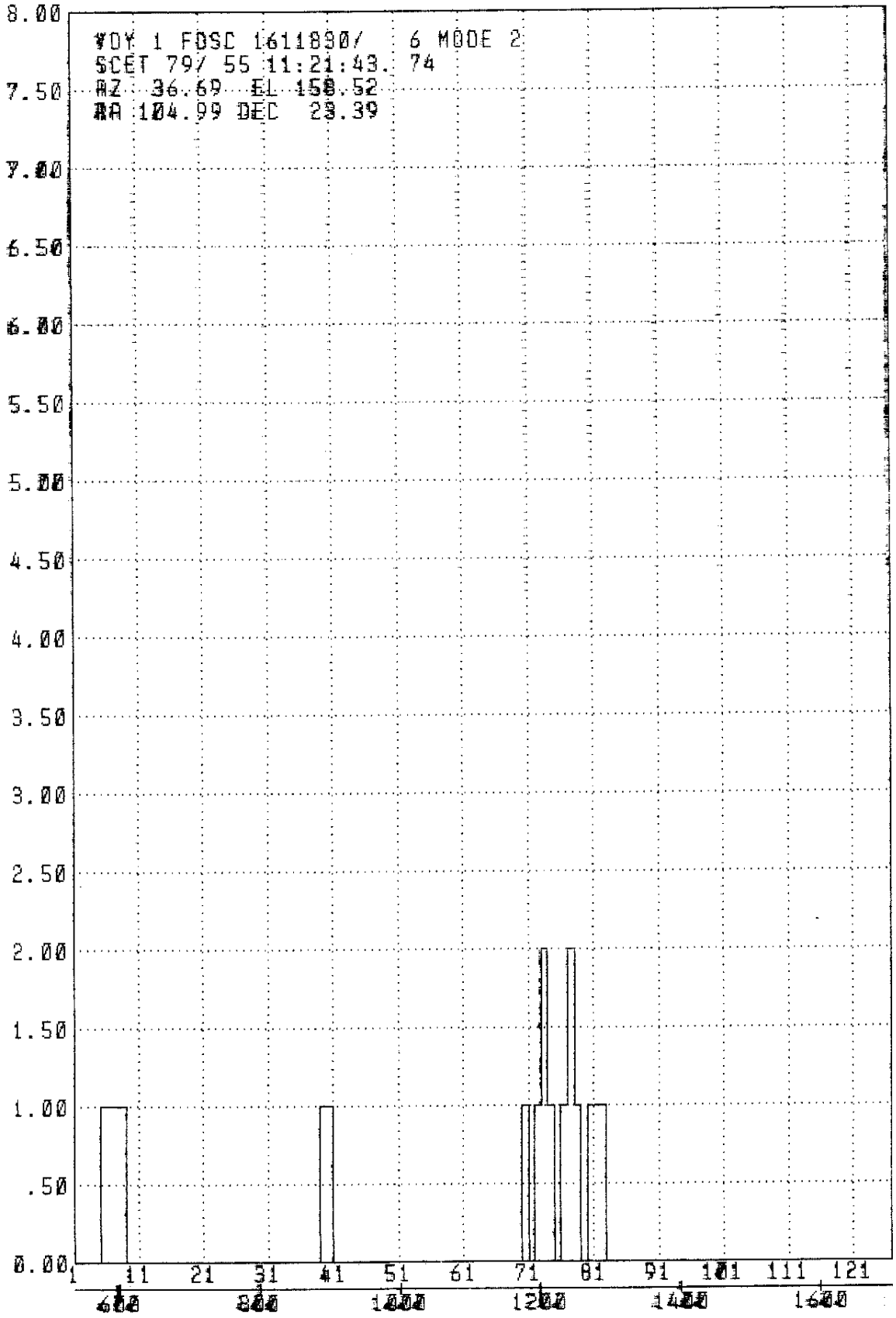
1981/111 10:59:32



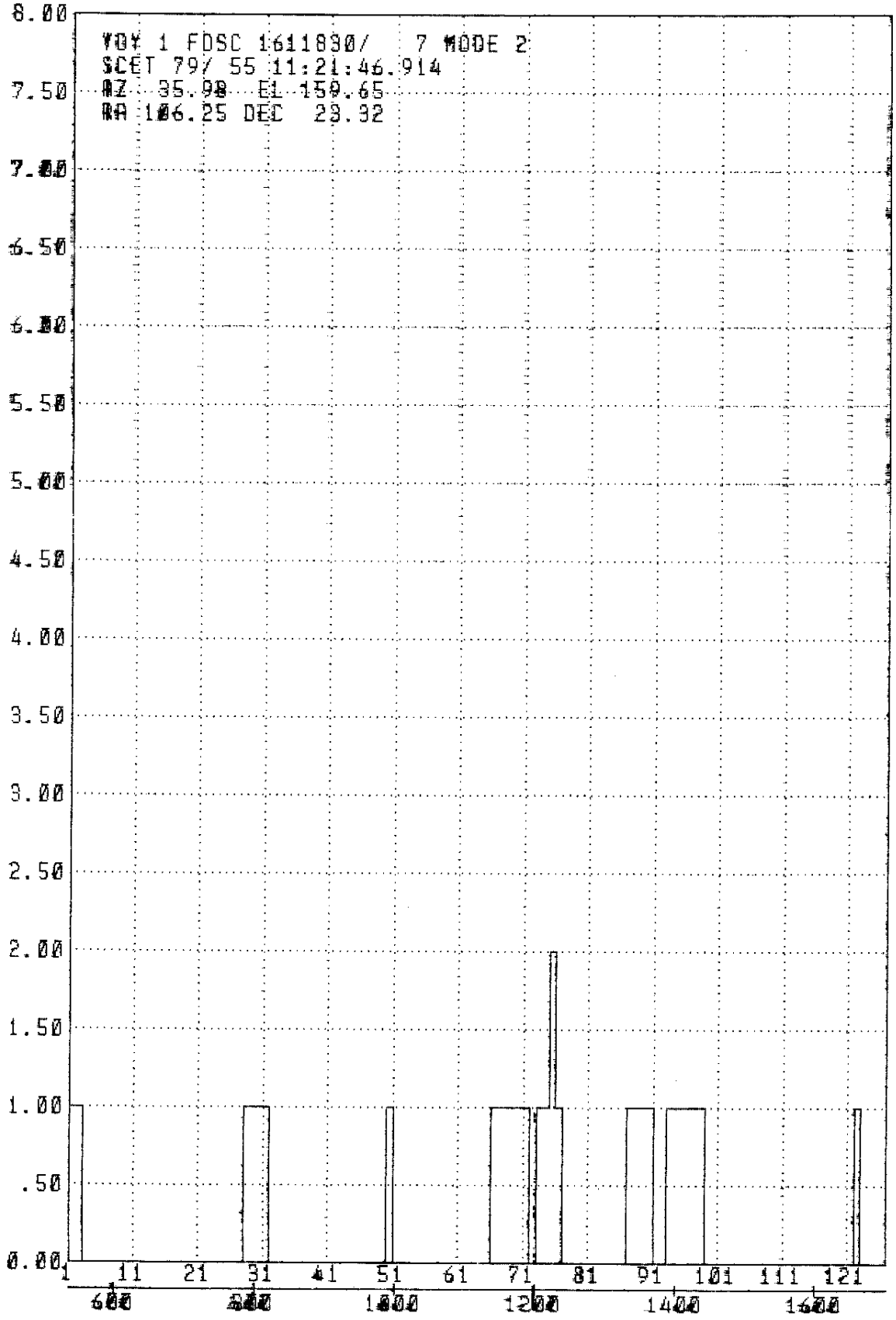
1981/111 11: 0: 1



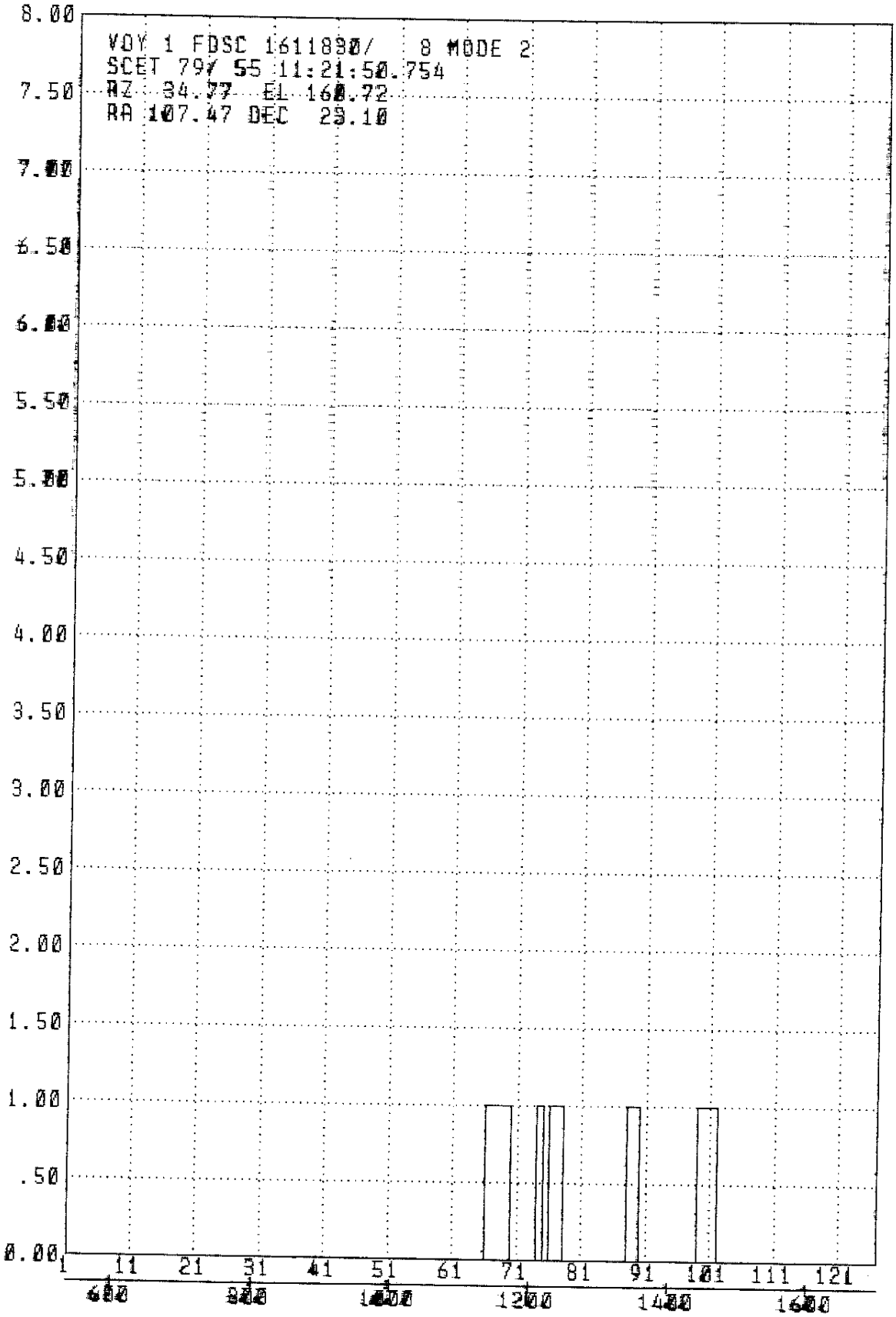
1981/111 11: 0:30



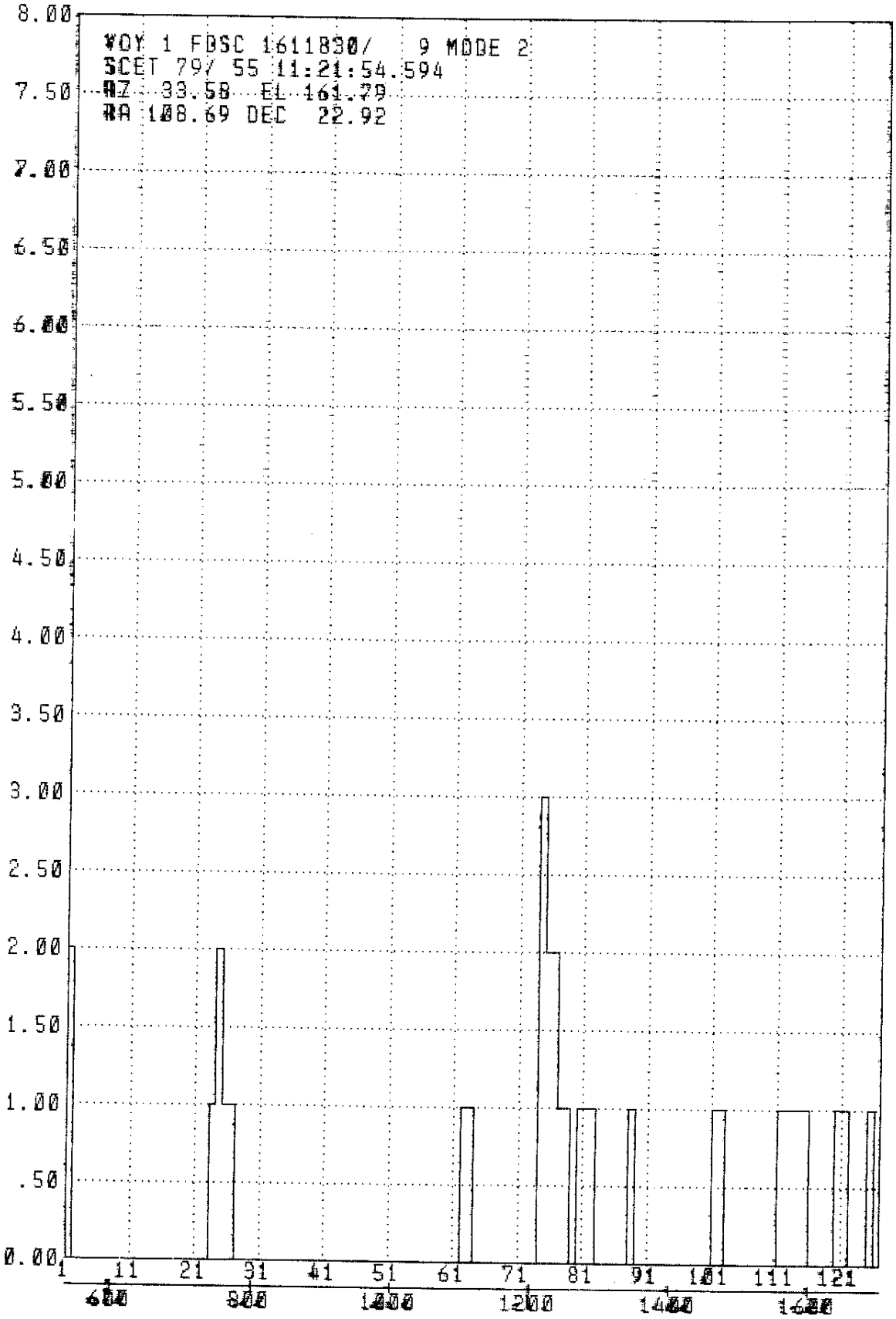
1981/111 11: 1: 0



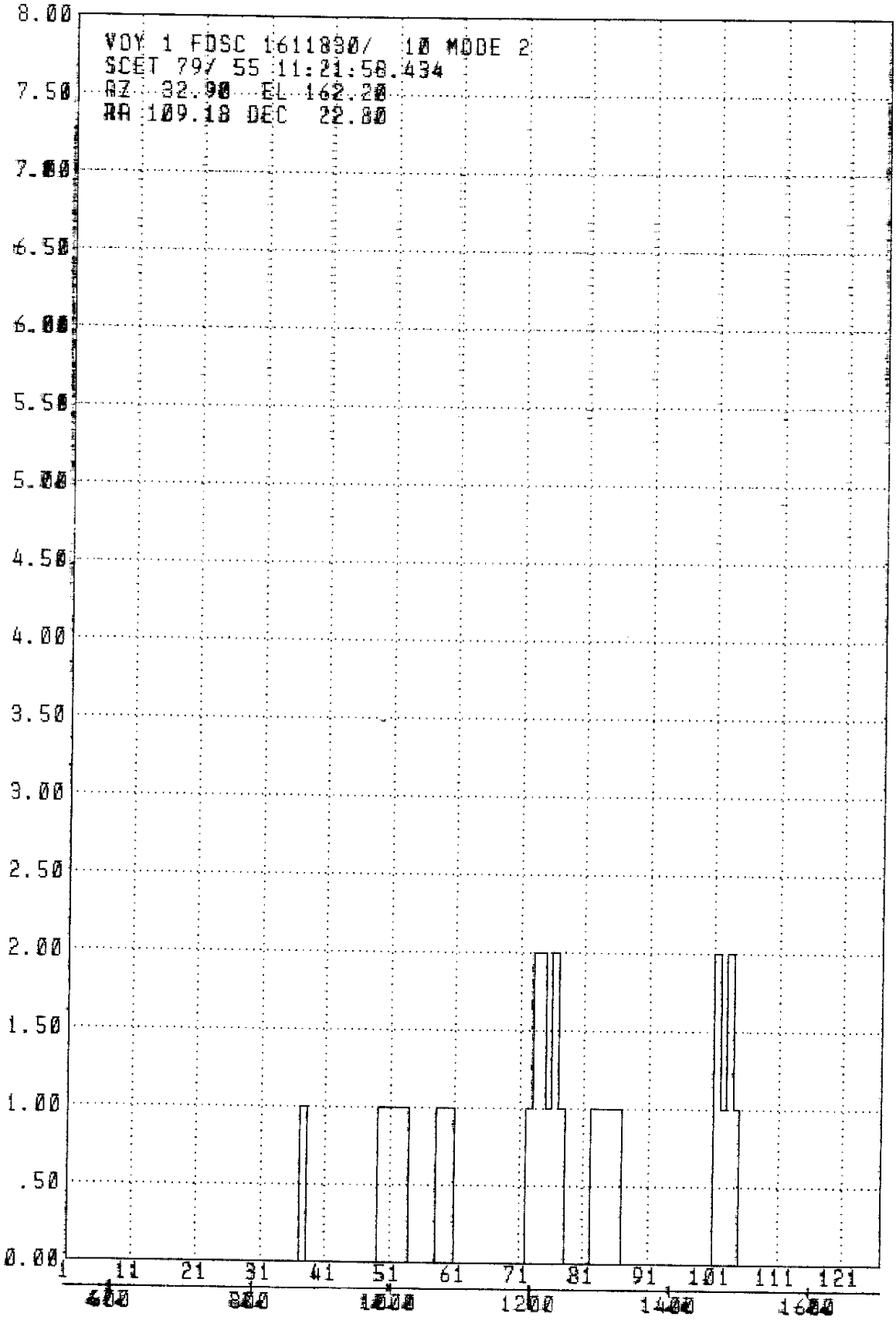
1981/111 11: 1:30



1981/111 11: 2: 9



1981/111 11: 2:40



```
COORDINATE SYSTEM = 2
     1=L/W
     2=AZ/EL
     3=RA/DEC

MINIMUMS =     29.000    153.600

INCREMENTS =      1.000       .300

REGION 1 =  CHANNEL  70 TO CHANNEL  78
REGION 2 =  CHANNEL  13 TO CHANNEL  21

BIN TYPE = 2  WHERE:
     1=SUM CHANNELS ON REGION 1
     2=SUM CHANNELS ON REGION 2
     3=SUM CHANNELS ON REG1 AND REG2
     4=DIFFERENCE OF REG1 MINUS REG2
     5=RATIO OF REG1 TO REG2

SPACECRAFT ID FLAG =  1

DATA QUALITY FLAG:  SPECTRAL     = ON
                    POSITIONAL   = ON
```

AZ- bin
of spectra 1-950
ch (13-21)
685Å feature

TOTAL SPECTRA = 950 TOTAL BINNED = 869

EL	AZ / Bin #	29.500 / 1	30.500 / 2	31.500 / 3	32.500 / 4	33.500 / 5	34.500 / 6	35.500 / 7	36.500 / 8	37.500 / 9	38.500 / 10	
153.750	1	0	0	0	0	0	0	0	0	3	0	3
154.050	2	0	0	0	0	0	0	0	0	23	0	23
154.350	3	0	0	0	0	0	0	0	20	10	0	30
154.650	4	0	0	0	0	0	0	0	22	0	0	22
154.950	5	0	0	0	0	0	0	0	35	0	0	35
155.250	6	0	0	0	0	0	0	0	39	0	0	39
155.550	7	0	0	0	0	0	0	0	14	0	0	14
155.850	8	0	0	0	0	0	0	16	12	0	0	28
156.150	9	0	0	0	0	0	0	35	0	0	0	35
156.450	10	0	0	0	0	0	0	34	0	0	0	34
156.750	11	0	0	0	0	0	8	27	0	0	0	35
157.050	12	0	0	0	0	0	34	0	0	0	0	34
157.350	13	0	0	0	0	0	13	0	0	0	0	13
157.650	14	0	0	0	0	0	61	0	0	0	0	61
157.950	15	0	0	0	0	17	14	0	0	0	0	31
158.250	16	0	0	0	0	38	0	0	0	0	0	38
158.550	17	0	0	0	0	36	0	0	0	0	0	36
158.850	18	0	0	0	3	33	0	0	0	0	0	36
159.150	19	0	0	0	36	0	0	0	0	0	0	36
159.450	20	0	0	0	35	0	0	0	0	0	0	35
159.750	21	0	0	11	32	0	0	0	0	0	0	43
160.050	22	0	0	37	0	0	0	0	0	0	0	37
160.350	23	0	0	33	0	0	0	0	0	0	0	33
160.650	24	0	11	26	0	0	0	0	0	0	0	37
160.950	25	0	39	0	0	0	0	0	0	0	0	39
161.250	26	0	31	0	0	0	0	0	0	0	0	31
161.550	27	0	25	0	0	0	0	0	0	0	0	25
161.850	28	0	5	0	0	0	0	0	0	0	0	5
162.150	29	0	0	1	0	0	0	0	0	0	0	1
162.450	30	0	0	0	0	0	0	0	0	0	0	0
		0	111	108	106	124	130	112	142	36	0	

TOTAL SPECTRA = 950 TOTAL BINNED = 869

EL	AZ	29.500	30.500	31.500	32.500	33.500	34.500	35.500	36.500	37.500	38.500	
		1	2	3	4	5	6	7	8	9	10	
153.750	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.333	0.000	3.333
154.050	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.043	0.000	1.043
154.350	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	2.200	0.000	1.400
154.650	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4.409	0.000	0.000	4.409
154.950	5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	11.029	0.000	0.000	11.029
155.250	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	24.026	0.000	0.000	24.026
155.550	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	26.857	0.000	0.000	26.857
155.850	8	0.000	0.000	0.000	0.000	0.000	0.000	14.938	19.083	0.000	0.000	16.714
156.150	9	0.000	0.000	0.000	0.000	0.000	0.000	15.057	0.000	0.000	0.000	15.057
156.450	10	0.000	0.000	0.000	0.000	0.000	0.000	12.706	0.000	0.000	0.000	12.706
156.750	11	0.000	0.000	0.000	0.000	0.000	11.125	7.333	0.000	0.000	0.000	8.200
157.050	12	0.000	0.000	0.000	0.000	0.000	7.206	0.000	0.000	0.000	0.000	7.206
157.350	13	0.000	0.000	0.000	0.000	0.000	6.231	0.000	0.000	0.000	0.000	6.231
157.650	14	0.000	0.000	0.000	0.000	0.000	4.131	0.000	0.000	0.000	0.000	4.131
157.950	15	0.000	0.000	0.000	0.000	5.647	4.786	0.000	0.000	0.000	0.000	5.258
158.250	16	0.000	0.000	0.000	0.000	4.816	0.000	0.000	0.000	0.000	0.000	4.816
158.550	17	0.000	0.000	0.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000	5.000
158.850	18	0.000	0.000	0.000	4.000	3.000	0.000	0.000	0.000	0.000	0.000	3.083
159.150	19	0.000	0.000	0.000	3.333	0.000	0.000	0.000	0.000	0.000	0.000	3.333
159.450	20	0.000	0.000	0.000	4.000	0.000	[illegible]	[illegible]	0.000	0.000	0.000	4.000
159.750	21	0.000	0.000	2.818	1.656	0.000	[illegible]	[illegible]	0.000	0.000	0.000	1.953
160.050	22	0.000	0.000	1.270	0.000	0.000	[illegible]	[illegible]	0.000	0.000	0.000	1.270
160.350	23	0.000	0.000	1.091	0.000	0.000	[illegible]	[illegible]	0.000	0.000	0.000	1.091
160.650	24	0.000	.818	.885	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.865
160.950	25	0.000	.641	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.641
161.250	26	0.000	.710	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.710
161.550	27	0.000	.480	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.480
161.850	28	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
162.150	29	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
162.450	30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	.658	1.269	3.066	4.500	5.646	12.464	14.401	1.556	0.000	

OVERALL AVERAGE = 6.127

AZ-EL bin
of spectra 1-950
ch (70-78)
Lyα

```
COORDINATE SYSTEM = 2
     1=L/W
     2=AZ/EL
     3=RA/DEC

MINIMUMS =     29.000    153.600

INCREMENTS =      1.000       .300

REGION 1 =  CHANNEL  70 TO CHANNEL  78
REGION 2 =  CHANNEL  13 TO CHANNEL  21

BIN TYPE = 1  WHERE:
     1=SUM CHANNELS ON REGION 1
     2=SUM CHANNELS ON REGION 2
     3=SUM CHANNELS ON REG1 AND REG2
     4=DIFFERENCE OF REG1 MINUS REG2
     5=RATIO OF REG1 TO REG2

SPACECRAFT ID FLAG =  1

DATA QUALITY FLAG:  SPECTRAL     = ON
                    POSITIONAL   = ON
```

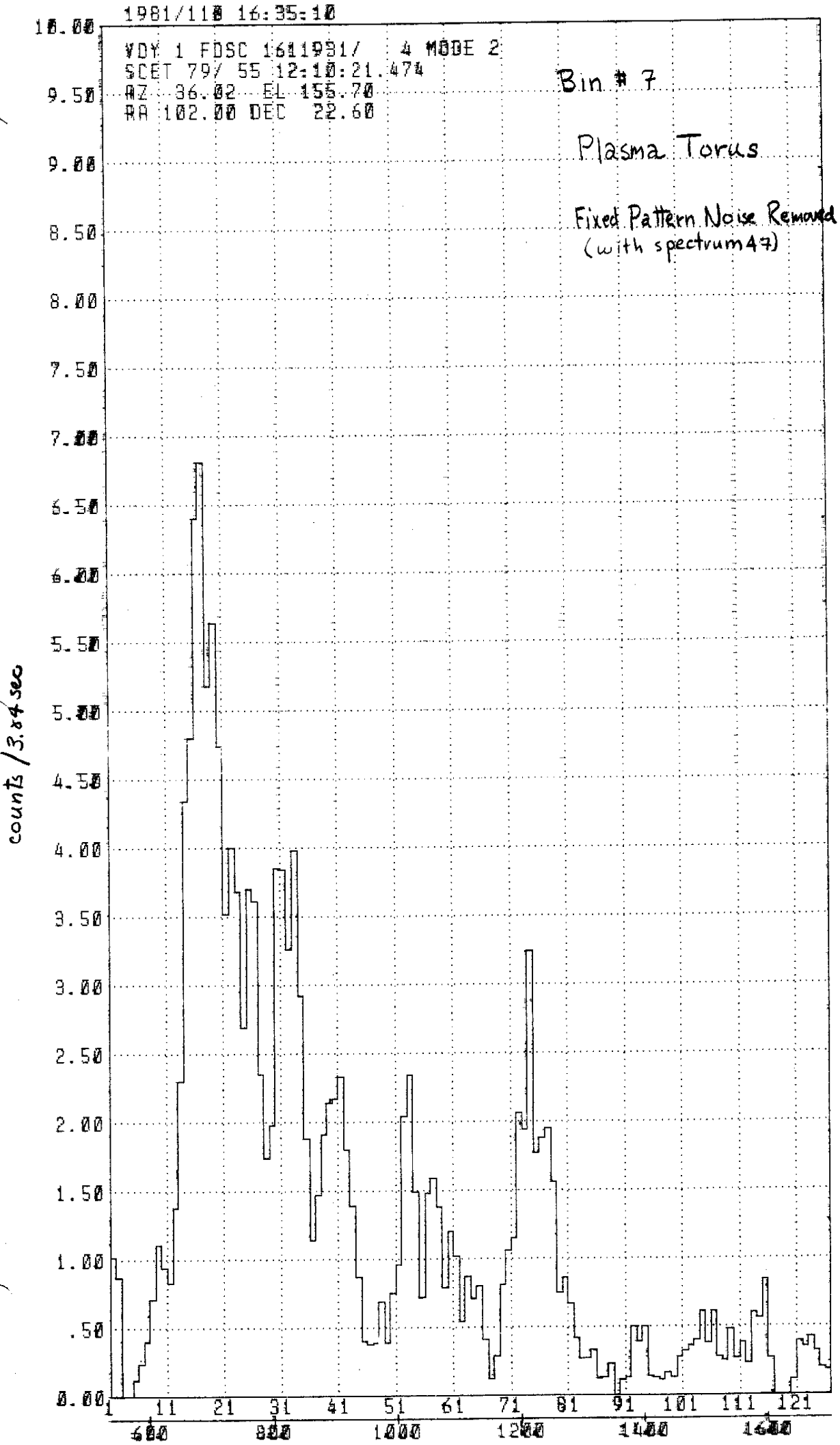
TOTAL SPECTRA = 950 TOTAL BINNED = 869

EL	bin #	AZ 29.500 (1)	30.500 (2)	31.500 (3)	32.500 (4)	33.500 (5)	34.500 (6)	35.500 (7)	36.500 (8)	37.500 (9)	38.500 (10)	#spectra in bin
153.750	1	0	0	0	0	0	0	0	0	3	0	3
154.050	2	0	0	0	0	0	0	0	0	23	0	23
154.350	3	0	0	0	0	0	0	0	0	10	0	30
154.650	4	0	0	0	0	0	0	0	[illegible]	0	0	22
154.950	5	0	0	0	0	0	0	0	[illegible]	0	0	35
155.250	6	0	0	0	0	0	0	0	[illegible]	0	0	39
155.550	7	0	0	0	0	0	0	0	14	0	0	14
155.850	8	0	0	0	0	0	0	16	12	0	0	28
156.150	9	0	0	0	0	0	0	35	0	0	0	35
156.450	10	0	0	0	0	0	0	34	0	0	0	34
156.750	11	0	0	0	0	0	8	27	0	0	0	35
157.050	12	0	0	0	0	0	34	0	0	0	0	34
157.350	13	0	0	0	0	0	0	0	0	0	0	13
157.650	14	0	0	0	0	0	61	0	0	0	0	61
157.950	15	0	0	0	0	17	14	0	0	0	0	31
158.250	16	0	0	0	0	38	0	0	0	0	0	38
158.550	17	0	0	0	0	36	0	0	0	0	0	36
158.850	18	0	0	0	0	36	0	0	0	0	0	36
159.150	19	0	0	0	36	0	0	0	0	0	0	36
159.450	20	0	0	0	35	0	0	0	0	0	0	35
159.750	21	0	0	11	32	0	0	0	0	0	0	43
160.050	22	0	0	37	0	0	0	0	0	0	0	37
160.350	23	0	0	33	0	0	0	0	0	0	0	33
160.650	24	0	11	26	0	0	0	0	0	0	0	37
160.950	25	0	39	0	0	0	0	0	0	0	0	39
161.250	26	0	31	0	0	0	0	0	0	0	0	31
161.550	27	0	25	0	0	0	0	0	0	0	0	25
161.850	28	0	5	0	0	0	0	0	0	0	0	5
162.150	29	0	0	1	0	0	0	0	0	0	0	1
162.450	30	0	0	0	0	0	0	0	0	0	0	0
		0	111	108	106	124	130	112	142	36	0	

TOTAL SPECTRA = 950 TOTAL BINNED = 869

EL	AZ	29.500	30.500	31.500	32.500	33.500	34.500	35.500	36.500	37.500	38.500	
		1	2	3	4	5	6	7	8	9	10	
153.750	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	6.333	0.000	6.333
154.050	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	8.652	0.000	8.652
154.350	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7.150	9.000	0.000	7.767
154.650	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7.455	0.000	0.000	7.455
154.950	5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7.571	0.000	0.000	7.571
155.250	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	9.590	0.000	0.000	9.590
155.550	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	9.286	0.000	0.000	9.286
155.850	8	0.000	0.000	0.000	0.000	0.000	0.000	8.938	[illegible].333	0.000	0.000	9.107
156.150	9	0.000	0.000	0.000	0.000	0.000	0.000	10.029	0.000	0.000	0.000	10.029
156.450	10	0.000	0.000	0.000	0.000	0.000	0.000	[illegible]	0.000	0.000	0.000	8.265
156.750	11	0.000	0.000	0.000	0.000	0.000	6.250	[illegible]	0.000	0.000	0.000	9.314
157.050	12	0.000	0.000	0.000	0.000	0.000	8.618	[illegible]	0.000	0.000	0.000	8.618
157.350	13	0.000	0.000	0.000	0.000	0.000	25.000	0.000	0.000	0.000	0.000	25.000
157.650	14	0.000	0.000	0.000	0.000	0.000	51.033	0.000	0.000	0.000	0.000	51.033
157.950	15	0.000	0.000	0.000	0.000	15.588	22.206	0.000	0.000	0.000	0.000	18.613
158.250	16	0.000	0.000	0.000	0.000	10.026	0.000	0.000	0.000	0.000	0.000	10.026
158.550	17	0.000	0.000	0.000	0.000	7.778	0.000	0.000	0.000	0.000	0.000	7.778
158.850	18	0.000	0.000	0.000	8.333	8.636	0.000	0.000	0.000	0.000	0.000	8.611
159.150	19	0.000	0.000	0.000	8.083	0.000	0.000	0.000	0.000	0.000	0.000	8.083
159.450	20	0.000	0.000	0.000	8.029	0.000	0.000	0.000	0.000	0.000	0.000	8.029
159.750	21	0.000	0.000	9.091	7.938	0.000	0.000	0.000	0.000	0.000	0.000	8.233
160.050	22	0.000	0.000	7.622	0.000	0.000	0.000	0.000	[illegible]	0.000	0.000	7.622
160.350	23	0.000	0.000	8.727	0.000	0.000	0.000	0.000	[illegible]	0.000	0.000	8.727
160.650	24	0.000	9.455	8.154	0.000	0.000	0.000	0.000	[illegible]	0.000	0.000	8.541
160.950	25	0.000	7.949	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7.949
161.250	26	0.000	7.839	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7.839
161.550	27	0.000	6.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	6.600
161.850	28	0.000	9.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	9.400
162.150	29	0.000	0.000	11.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	11.000
162.450	30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	7.829	8.269	8.028	9.766	31.485	9.384	8.366	8.556	0.000	

OVERALL AVERAGE = 12.041


1981/110 16:35:10
VDY 1 FDSC 1611931/ 4 MODE 2
SCET 79/ 55 12:10:21.474
AZ 36.02 EL 155.70
RA 102.00 DEC 22.60
Bin # 7
Plasma Torus
Fixed Pattern Noise Removed
(with spectrum 47)
counts /3.84 sec
10.00
9.50
9.00
8.50
8.00
7.50
7.00
6.50
6.00
5.50
5.00
4.50
4.00
3.50
3.00
2.50
2.00
1.50
1.00
.50
0.00
1
11
21
31
41
51
61
71
81
91
101
111
121
600
800
1000
1200
1400
1600

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611931	4	1	0011	79/ 55	12:10:21.474	12:46:47.176	2	53.76

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
47	0	0	1	3	1	0	0.0	0.0	0.0	.10000E+01	0

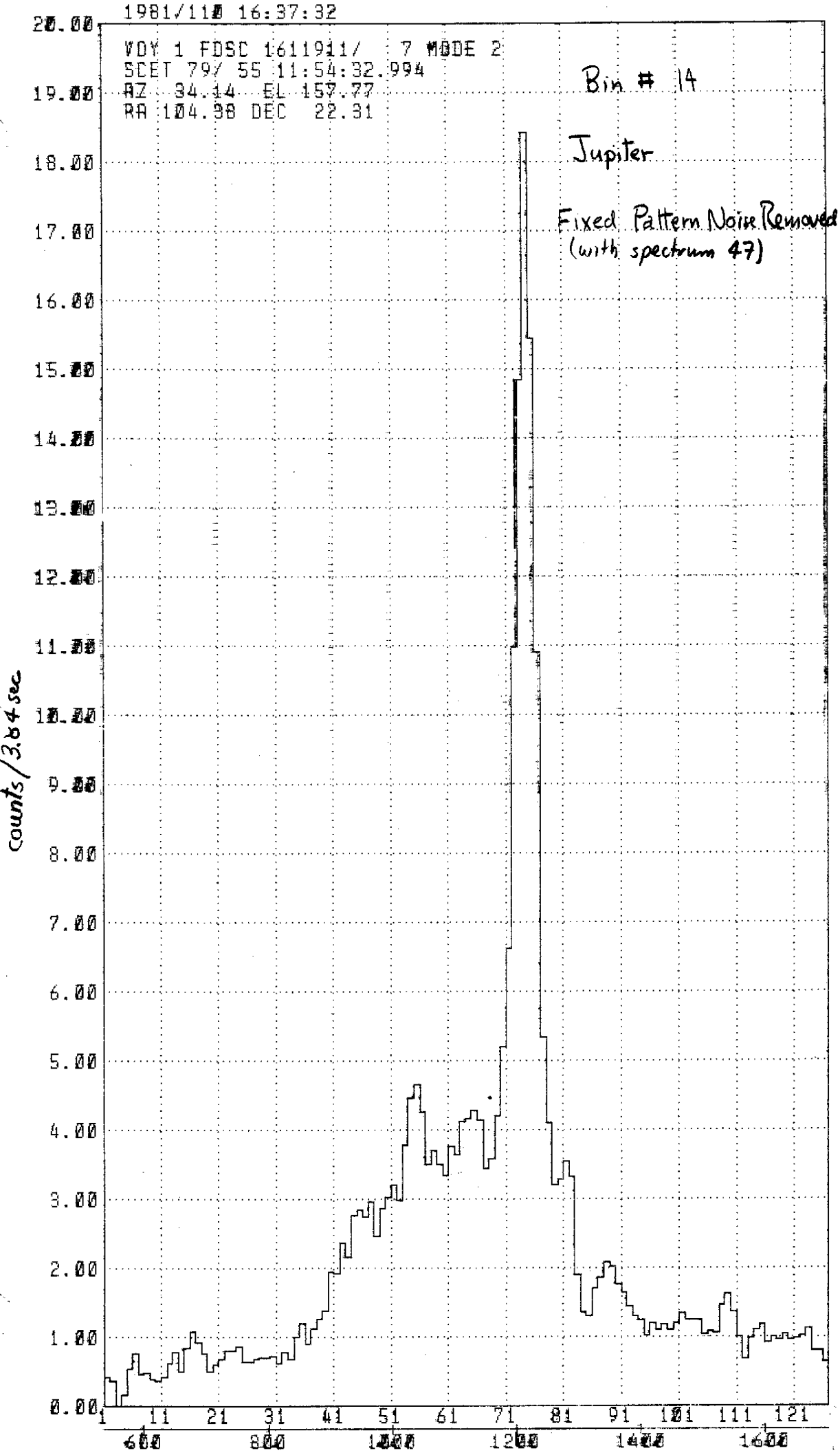
COORD MODE	AZ MIN	AZ MAX	EL MIN	EL MAX	HV	TEMP	SQRT
AZ/EL	29.00	39.00	155.40	155.70	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1991	308.96	-19.25	3.889	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.1911	-1.0612	.9844	-.1976

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	102.04	22.62	36.05	155.74	-.015	-.016	-.015	-.014	-.013
OC	81.95	17.62			.038	.038	.038	.037	.037

				0.00	0.00	0.00	0.00	0.00	0.00
1.03	.87	0.00	0.00	.12	.25	.40	.71	1.12	.97
.83	1.38	2.31	4.34	4.81	6.40	6.82	5.19	5.65	4.74
3.53	4.00	3.68	2.70	3.70	3.62	2.35	1.75	1.98	3.85
3.84	3.27	3.99	2.92	1.88	1.14	1.48	1.92	2.14	2.18
2.33	1.81	1.40	.88	.41	.38	.40	.70	.40	.75
.97	2.04	2.34	1.50	.73	1.48	1.59	1.38	.80	1.21
1.02	.54	.88	.72	.81	.41	.13	.30	.81	1.06
1.16	2.07	1.95	3.25	1.77	1.88	1.96	1.56	.75	.87
.67	.43	.27	.28	.34	.13	.13	.24	0.00	.11
.14	.51	.40	.51	.14	.13	.12	.16	.14	.29
.33	.36	.41	.62	.38	.61	.28	.25	.48	.28
.38	.24	.61	.57	.85	.27	0.00	0.00	0.00	.11
.39	.35	.42	.32	.20	.18	0.00	0.00	0.00	0.00
0.00	0.00								


1981/110 16:37:32
VOY 1 FDSC 1611911/ 7 MODE 2
SCET 79/ 55 11:54:32.994
AZ 34.14 EL 157.77
RA 104.38 DEC 22.31
Bin # 14
Jupiter
Fixed Pattern Noise Removed
(with spectrum 47)
counts/3.84 sec
20.00
19.00
18.00
17.00
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
1 11 21 31 41 51 61 71 81 91 101 111 121
600 800 1000 1200 1400 1600

FDSC	SPEC	VOY	STAT	YR/DAY	SCET	ERT	MODE	DURATION
1611911	7	1	0011	79/ 55	11:54:32.994	12:30:58.609	2	234.24

FPN	CAL	SCAT	CM	HVL	NBIN	SMTH	PEAK	INTG	MEAN	SCALE	LN
47	0	0	1	3	1	0	0.0	0.0	0.0	.10000E+01	0

COORD MODE	AZ MIN	AZ MAX	EL MIN	EL MAX	HV	TEMP	SQRT
AZ/EL	29.00	39.00	157.50	157.80	2080.9	-2.21	0

JPLHP	JPLEDR	HPEDR	SPEC	SUN RA	SUN DEC	PITCH	YAW
9000	021158	101320	1744	308.96	-19.25	3.860	-4.992

TRANSLATION MATRIX	A	B	C	D
	-.2020	-1.0565	.9825	-.2098

	RA	DEC	AZ	EL	L1	L2	L3	L4	L5
AG	104.39	22.37	34.30	157.79	-.058	-.059	-.060	-.061	-.061
OC	84.40	17.16			.021	.021	.021	.020	.019

				0.00	0.00	0.00	0.00	0.00	0.00
.44	.37	0.00	.17	.55	.77	.46	.49	.39	.36
.42	.64	.80	.51	.84	1.09	.93	.76	.50	.62
.69	.81	.81	.87	.65	.65	.68	.71	.72	.74
.62	.78	.70	1.01	1.21	.92	1.12	1.27	1.39	1.94
1.92	2.38	2.18	2.77	2.84	2.75	2.96	2.47	2.86	3.03
3.20	2.98	3.79	4.47	4.66	4.28	3.51	3.71	3.51	3.36
3.78	3.66	4.14	4.17	4.28	4.16	3.46	3.58	4.21	5.20
6.63	10.98	14.85	18.43	15.45	10.90	5.35	4.11	3.20	3.29
3.56	3.33	1.91	1.36	1.32	1.72	1.87	2.09	2.04	1.76
1.64	1.46	1.30	1.25	1.03	1.21	1.12	1.19	1.12	1.21
1.36	1.25	1.25	1.25	1.05	1.10	1.07	1.47	1.63	1.37
1.00	.68	1.00	1.12	1.20	.94	1.01	.97	1.06	.97
.99	1.03	1.14	.82	.81	.66	0.00	0.00	0.00	0.00
0.00	0.00								

DUMP OF TAPE D45044

INPUT TAPE D45044 ON MT5
DATA INPUT 07 FL 1 1 1

FILE	RECORD	LENGTH	5760 BYTES							
(0)	000001000001	000201000002	000217000167	000013000025	000027003352	000217000167	000013000171	000161001200
(48)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000001
(96)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(144)	000000000000	000000000000	000000000000	201007256030	561360500004	221425343014	234577074020	303434502016
(192)	262014243012	237416657016	215576015012	523764041771	534425557771	535167276771	535631015771	536372523771
(240)	537763301757	321150434755	274432456761	237365417743	340435620763	462545757775	574163321002	375631537000
(288)	460372722775	235175315022	542774723012	363410702004	540140555006	000000000000	000000000000	000000000000
(336)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(384)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(432)	000000000000	000000000000	000000000000	000000000000	200000000002	200000000004	200000000004	200000000004
(480)	200000000002	200000000002	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000
(528)	000000000000	200000000002	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000
(576)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(624)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(672)	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(720)	000000000000	000000000000	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000
(768)	200000000002	000000000000	000000000000	000000000000	000000000000	200000000002	200000000004	000000000000
(816)	200000000002	200000000004	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(864)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(912)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(960)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1008)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1056)	000000000000	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000
(1104)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1152)	000002000001	000201000002	000217000157	000013000025	000033002612	000217000167	000013000171	000165000540
(1200)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000002
(1248)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(1296)	000000000000	000000000000	000000000000	201007256030	561360500004	221425343014	235022522020	314216056016
(1344)	262417516012	240165177016	216417562012	536345157771	537006475771	537550003771	537021434771	535363154771
(1392)	340435620763	220743003765	261367101765	316206202765	337744660765	475342034775	574022563002	375770261000
(1440)	462613441775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(1488)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1536)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002
(1584)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1632)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	000000000000	000000000000
(1680)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1728)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1776)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1824)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1872)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1920)	000000000000	000000000000	200000000004	200000000002	200000000004	300000000004	200000000002	000000000000
(1968)	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2016)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2064)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2112)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2160)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2208)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2256)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2304)	000003000001	000201000002	000217000167	000013000025	000037002152	000217000167	000013000171	000171000000
(2352)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000003
(2400)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(2448)	000000000000	000000000000	000000000000	201007256030	561360500004	221425354014	235346157020	314777663016
(2496)	263017722012	240634020016	217237214012	535234234771	523573541771	522033057771	521223103771	521764203771
(2544)	337744642765	361403323765	201421000767	217556256767	237770316767	500150700775	563762401002	376026075000
(2592)	475142022775	235175315022	542774723012	363410702004	540140555006	000000000000	000000000000	000000000000
(2640)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2688)	200000000002	000000000000	000000000000	000000000000	200000000004	200000000004	200000000002	200000000004
(2736)	200000000002	000000000000	000000000000	000000000000	200000000002	200000000002	200000000004	200000000002

DUMP OF TAPE GIN7

INPUT TAPE GIN7 ON MT5
DATA INPUT 07 NF 1 FL 1 2 2

FDSC= 16118:30

FILE	1 RECORD	1 LENGTH	5760BYTES		Word 25-26	27-28	29-30	31-32
(0)	000001000001	000201000002	000217000167	000013000025	000027003352	000217000167	000013000171	000161001200
(48)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000001
(96)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(144)	000000000000	000000000000	000000000000	201007256030	561360500004	221425343014	234577074020	303434502016
(192)	262014243012	237416657016	215576015012	523764041771	534425557771	535167276771	535631015771	536372523771
(240)	537763301757	321150434755	274432456761	237365417763	340435620763	462545757775	574163321002	375631537000
(288)	460372722775	235175315022	542774723012	363410702004	540140555006	000000000000	000000000000	000000000000
(336)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(384)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(432)	000000000000	000000000000	000000000000	000000000000	200000000002	200000000004	200000000004	200000000004
(480)	200000000002	200000000002	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000
(528)	000000000000	200000000002	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000
(576)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(624)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(672)	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(720)	000000000000	000000000000	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000
(768)	200000000002	000000000000	000000000000	000000000000	000000000000	200000000002	200000000004	000000000000
(816)	200000000002	200000000004	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(864)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(912)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(960)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1008)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1056)	000000000000	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000
(1104)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1152)	000002000001	000201000002	000217000167	000013000025	000033002612	000217000167	000013000171	000165000540
(1200)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000002
(1248)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(1296)	000000000000	000000000000	000000000000	201007256030	561360500004	221425343014	235022522020	314216056016
(1344)	262417516012	240165177016	216417562012	536345157771	537006475771	537550003771	537021434771	535363154771
(1392)	340435620763	220743003765	261367101765	316206202765	337744660765	475342034775	574022563002	375770261000
(1440)	462613441775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(1488)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1536)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002
(1584)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1632)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	000000000000	000000000000
(1680)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1728)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1776)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1824)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1872)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1920)	000000000000	000000000000	200000000004	200000000002	200000000004	300000000004	200000000002	000000000000
(1968)	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2016)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2064)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2112)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2160)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2208)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2256)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2304)	000003000001	000201000002	000217000167	000013000025	000037002152	000217000167	000013000171	000171000000
(2352)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000003
(2400)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(2448)	000000000000	000000000000	000000000000	201007256030	561360500004	221425354014	235346157020	314777663016
(2496)	263017722012	240634020016	217237214012	535234234771	523573541771	522033057771	521223103771	521764203771
(2544)	337744642765	361403323765	201421000767	217556256767	237770316767	500150700775	563762401002	376026075000
(2592)	475142022775	235175315022	542774723012	363410702004	540140555006	000000000000	000000000000	000000000000
(2640)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2688)	200000000002	000000000000	000000000000	000000000000	200000000004	200000000002	200000000002	200000000004
(2736)	200000000002	000000000000	000000000000	000000000000	200000000002	200000000002	200000000004	200000000002

(2784)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2832)	000000000000	200000000004	200000000004	000000000000	000000000000	000000000000	000000000000	000000000000
(2880)	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000	200000000002	000000000000
(2928)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2976)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3024)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3072)	000000000000	000000000000	000000000000	200000000002	200000000004	200000000002	000000000000	200000000002
(3120)	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3168)	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000
(3216)	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3264)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002	000000000000
(3312)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3360)	000000000000	000000000000	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000
(3408)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3456)	000004000001	000201000002	000217000167	000013000025	000143001412	000217000167	000013000172	000000003210
(3504)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000004
(3552)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(3600)	000000000000	000000000000	000000000000	201007256030	561360500004	221425354014	235777257020	316206400016
(3648)	263762757012	242016522016	220625637012	521606102771	522346542771	523007202771	523547642771	534210302771
(3696)	237770315767	260202355767	300414416767	320626457767	341140517767	515233156775	563570421002	376213120000
(3744)	501365720775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(3792)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3840)	000000000000	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3888)	200000000004	200000000004	200000000004	200000000004	200000000002	200000000004	200000000002	000000000000
(3936)	000000000000	000000000000	000000000000	300000000004	200000000004	200000000004	200000000002	000000000000
(3984)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4032)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4080)	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000
(4128)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4176)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4224)	000000000000	000000000000	200000000004	200000000004	200000000006	240000000006	200000000006	240000000006
(4272)	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000
(4320)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4368)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4416)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4464)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4512)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4560)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4608)	000005000001	000201000002	000217000167	000013000025	000147000752	000217000167	000013000172	000004002550
(4656)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000005
(4704)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(4752)	000000000000	000000000000	000000000000	201007256030	561360500004	221425354014	236543276020	317545435016
(4800)	274632417012	243230435016	222236417012	534036754771	534576656771	535236560771	537031563771	541342236771
(4848)	341140514767	361352555767	200632301771	203344762771	215142274771	523027322775	563374436002	376401423000
(4896)	516152003775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(4944)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(4992)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5040)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5088)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5136)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5184)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5232)	200000000004	200000000004	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5280)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5328)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002
(5376)	200000000002	200000000002	200000000004	200000000004	200000000004	200000000006	200000000004	200000000002
(5424)	000000000000	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000	000000000000
(5472)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002
(5520)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002	200000000002
(5568)	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(5616)	200000000002	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000	000000000000
(5664)	000000000000	200000000002	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000
(5712)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000

FILE 1 RECORD 2 LENGTH 5760BYTES

(0)	000006000001	000201000002	000217000167	000013000025	000153000212	000217000167	000013000172	000010002010
(48)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000006

(96)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(144)	000000000000	000000000000	000000000000	201007256030	561360500004	221425354014	237207303020	321005577016
(192)	275565677012	254543714016	223636163012	541207022771	543422562771	555636323771	557223654771	555760562771
(240)	215142275771	216637576771	220435077771	223416254771	236155515771	541000302775	563203202002	376565454000
(288)	522760016775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(336)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(384)	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000
(432)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(480)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(528)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(576)	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000
(624)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(672)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(720)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(768)	000000000000	200000000002	000000000000	200000000002	200000000004	200000000002	000000000000	200000000002
(816)	200000000004	200000000002	000000000000	200000000002	200000000002	200000000002	000000000000	000000000000
(864)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(912)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(960)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1008)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1056)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1104)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1152)	000007000001	000201000002	000217000167	000013000025	000156003422	000217000167	000013000172	000014001350
(1200)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000007
(1248)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(1296)	000000000000	000000000000	000000000000	201007256030	561360500004	220154342014	237630140020	322210402016
(1344)	275365643012	255754555016	223356341012	556614200771	555143456771	543372736771	541622215771	542011760771
(1392)	237011021771	241366235771	243743442771	256220656771	261743517771	540204301775	563224215002	376556762000
(1440)	521415402775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(1488)	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000
(1536)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1584)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1632)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002
(1680)	200000000002	200000000002	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000
(1728)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1776)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	000000000000	000000000000
(1824)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(1872)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002	200000000002
(1920)	200000000002	200000000002	200000000002	000000000000	200000000002	200000000002	200000000004	200000000002
(1968)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2016)	000000000000	000000000000	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000
(2064)	200000000002	200000000002	200000000002	200000000002	200000000002	200000000002	000000000000	000000000000
(2112)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2160)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2208)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	000000000000	000000000000
(2256)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2304)	000010000001	000201000002	000217000167	000013000025	000162002762	000217000167	000013000172	000020000610
(2352)	000000000000	000000000000	000001000003	000000000000	103150140162	141161155170	243201235220	234205000010
(2400)	000000000000	302340660152	362741220004	000000000000	000000000000	000000000000	200000000002	000000000000
(2448)	000000000000	000000000000	000000000000	201007256030	561360500004	215601456014	240342161020	323404016016
(2496)	274425676012	257200074016	221607660012	543606336771	557017040771	562227542771	575745403771	415432337767
(2544)	263361543771	276616754771	302154156771	315140261771	316006735771	522747042775	563424217002	376376321000
(2592)	503630014775	235175315022	542774723012	363410700004	540140555006	000000000000	000000000000	000000000000
(2640)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2688)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2736)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2784)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2832)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2880)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2928)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(2976)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3024)	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002	200000000002
(3072)	200000000002	000000000000	000000000000	000000000000	000000000000	200000000002	000000000000	200000000002
(3120)	200000000002	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
(3168)	000000000000	000000000000	000000000000	200000000002	200000000002	000000000000	000000000000	000000000000
(3216)	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	200000000002	200000000002

```
( 4224)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4272)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4320)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4368)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4416)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4464)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4512)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4560)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4608)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4656)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4704)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4752)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4800)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4848)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4896)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4944)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 4992)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5040)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5088)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5136)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5184)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5232)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5280)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5328)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5376)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5424)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5472)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5520)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5568)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5616)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5664)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
( 5712)  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000

      FILE   INPUT    DATA RECORDS    MAX.          READ ERROR SUMMARY          INPUT RETRIES
             RECS.      INPUT         SIZE       PERM  ZERO B   SHORT  UNDEF.  #RECS.  TOTAL#
        1     644      646            5760         0       0       0       0      86      99

 EOJ        DUMP STOPPED AFTER FILE   1          # OF PERMANENT READ ERRORS  0

START TIME 08/09/84  09:16:52        STOP TIME 08/09/84  09:18:19

$$
$REW LO
```