

FRM - A PROGRAM FOR SPECTROGRAM SCAN**RECORDS REFORMATTING**

Bellás Y.

**Department of Astronomy and
National Astronomical Observatory,
Bulgarian Academy of Sciences,
72 Lenin blvd., 1784 Sofia, Bulgaria**

Abstract: A program "FRM", part of the general S-software for digital processing of stellar coude-spectrograms in the Rozhen National Astronomical Observatory (NAO) is presented. The FRM provides a program interface between the FITS format releases elaborated for spectrogram records in the NAO. The program assures tools for records reformatting from one to other release of the FITS, a number of necessary functions for recording devices (disk, tape) operating, and a set of miscellaneous help functions. The functions are activated by fixed commands the user inserts in an interactive dialogue, or the FRM gets from user-written command-files in another, indirect-command input mode. The program is created on the DEC PDP 11/34 minicomputer of the NAO, under the RSX-11/M system's FORTRAN IV PLUS language.

Keywords: software, FITS format, reformatting, spectrogram

1. INTRODUCTION

A specialized software, referred to as "S", is elaborated in the Rozhen National Astronomical Observatory (NAO) for digital processing of stellar coude-spectrograms. This is an integrated package of programs and object-modules libraries assuring spectrograms scanning/recording (Bellas, Borisova, 1988), and the records graphic-visualization and processing (Bellas, in preparation). In order to standardize the scans records input/output, a general format must be established.

For a flexible transport of spectrogram records on tapes to other institutes, a standard format has been accepted for the S-software that conforms the requirements of the well known FITS format (Wells, Greisen, 1981). On the other hand, the specific hardware, the S uses, the computer controlled JOYCE LOEBL microdensitometer for scanning/recording, the PDP 11/34 minicomputer for processing, set a problem to work out versions of the standard format having a common logical structure, but different in their physical realization. Two releases for tape and two for disk devices are created. This itself, set another problem - to develop a program interface for reformatting the scan-records.

A program, referred to as "FRM", intended to assure such an interface as part of the S-software, is created in the NAO on the PDP 11/34 minicomputer. The paper presents this program, as follow: the FITS format releases in the NAO are described in the second paragraph; the purpose and functions of the FRM are commented in the third; its realization is presented in the fourth paragraph; finally, the fifth gives a general information on its use.

The FRM is fully described in a documentation including an user's manual "FRM-MANUAL" and a listing of the program's modules "FRM-CODE". Both parts are available as text-files on the PDP as well as on an IBM-PC or compatible computer.

2. FITS FORMAT RELEASES IN THE NAO

Four versions, two for tape and two for disk devices, of the FITS format are created in the NAO for spectrogram scans records processing. They have a common logical structure, but differs in its physical realization.

2.1 FITS releases for tapes

For scans recording and flexible transport of scan-files (processed or not) to other institutes, there is realized the "STANDARD" format that follows in logical and physical structure the FITS recommendations. The scans are recorded on a tape in separate files, each ended by an EOF-mark, the last file ended by double EOF (the EOFV-mark).

Logically, each scan-file consists of a header (leading record) and scan-data. The header is a 2880-ASCII-characters text, considered as 36 rows of 80 characters, containing particular information about the spectrogram, scanning, and processing. The rows are parameters or commentary, having a fixed format. A special end-row follows the last filled up; the rest of the header, if any, is complemented with spaces. The scan-data follow immediately after the header. Each data is recorded as an integer (two-bytes, twos-complement word; leading byte - higher order), scaled in accordance with a

particular parameter of the header (another one fixes the step between the readings). An example of a real header is shown in the Fig.1.

Physically, a file in the "STANDARD" consists of blocks of 1440 words. The first is the very header, the following contain the scan-data. The file's last block is complemented with zeroes, if not filled up with data; EOF-mark follows.

In order to optimize the scanning, a version referred to as "DG - DOS" format is created. Physically, it conforms the standard for the DATA NOVA 3/12 minicomputer controlling the JOYCE LOEBL MDM-6 microdensitometer. There are two physical differences from the "STANDARD", namely: all the scans of a particular spectrogram are sequentially recorded in one and the same tape-file; the blocks are 257-words length (the last two words are system-reserved for the double-recorded file-number). The last block of each scan (header and data) in the file is complemented with zeroes, if necessary.

2.2 FITS releases for disks

For an efficient processing of the scan records by the S-software's restoring programs, a version of the "STANDARD" is realized for the PDP minicomputer's disk-devices. This format is referred to as "DD - SPC"; it has the same logical structure (header and scan-data), but follows physically the RSX system's FILE-11 structure. The scan-file is unformatted (binary), direct-access (by record number); it is named and "residents" in the user's directory. The file's blocks have 256-words length; the file's very last data is followed by an EOF-mark, so no complementing with zeroes is performed.

The scan-data (not the header) words are "swapped" (leading byte - lower order) as this is required by the RSX-11 for integer data.

A special naming system is created, in order to maintain the sequential character of the scan-files. The name looks like "ppooss.SPC", where the "pp" fixes (for the directory) two-digit number for the spectrogram, the "oo" - such for the plate's object, and the "ss" counts the particular scan of the object. Thus, the "pp0000.SPC" names the scan-file of the very stellar-spectrum band of the "pp" spectrogram, the "pp0100.SPC" and "pp0101.SPC" of its upper and lower Fe-Ar reference-spectrum bands, the "pp02ss.SPC" - these of the photometric calibration; the "pp03ss.SPC" names the filtered scan-files, and the "pp04ss.SPC" - the restored ones.

With the sole purpose of a fast, tape-disk-tape copying (see further), another version for disk-files is accepted, referred to as "FF - SPC" format. There are two differences from the "DD - SPC", making it faster for the reformatting. Firstly, the scan-data are not swapped. Secondly, the name looks like "FFnnnn.SPC" where the variable part, the "nnnn", fixes a four-digit number being referred to the sequence of the processing only.

3. FRM FUNCTIONS

The FRM program assures the necessary interface between the FITS format releases in the NAO. It includes tools to reformat records from one to other format, to handle volumes with FITS-files on disks or tapes, to operate tapes in an universal regime (regardless the format), as well as a set

of miscellaneous functions. A particular function is to be activated by a fixed command (see 5).

3.1 Reformatting functions

The very reformatting is assured by so called T-commands for tape-to-disk copying, and by D-commands for the reverse, disk-to-tape process. These are applied to either "STANDARD" or "DG - DOS" tape-files, and to "DD - SPC" disk-files. One or more files, a group of them (one or more plates), as well as a whole volume can be processed.

On the other hand, F-commands can be used for a "fast" reformatting. These process tape-files as the above, but use "FF - SPC" format disk-files. These functions assure a full, tape-disk-tape reformatting as well as separate tape-to-disk and disk-to-tape processings of one or more files.

3.2 Disk-volume operating

This includes a disk-volume attaching (making it known to the task), necessary before any using of its files. This, as well as any changing of the disk-volumes, and the setting of "DD - SPC" file-names "limits" is assured by A-commands.

As the task fixes a "current-file" by its name, that will be the defaulted for some commands, the user can change this set up (for the "DD - SPC" files only) by N-commands.

Additionally, there can be obtained an information about the disk-volume contents, list of one or more files (for the two formats separately), of a group of them (object, plate; "DD - SPC" only), of the whole disk-volume. This is assured

by L-commands. Similarly, the user can "clear" the attached disk-volume, deleting one or more files by C-commands.

3.3 Tape-volume operating

Before any using of tape-files, like for the disk, the tape-volume must be attached; but, it should be detached in order to be changed with another (800 or 1600 BPI tapes can be used). The attaching/detaching is assured by the magtape initialization M-commands. These provide also functions for rewinding and zeroing (full initialization) the tape.

An information about a tape-volume's contents ("STANDARD" or "DG - DOS" files) can be obtained by I-commands. These list one or more files (starting with the current), one or more plates, or the whole volume; the plate's headers can be listed out, optionally.

Unlikely the disk, the desired tape-file must be really positioned on. This would be done by P-commands, fixing the target file's or plate's identifier (header's parameter) and the direction of the positioning. On the other hand, there can be skipped desired number of files or plates moving the tape in the fixed direction by S-commands.

3.4 Universal magtape operating

In order to test unknown tapes, particular error-status, or the very program, there are assured functions to process free-format tapes in so called U-regime. This regime must be activated as well as it must be deactivated to return to the normal one (FITS-files only). In the U-regime, the user can

obtain an information about the current-file's contents (the header is automatically skipped if it is a FITS-file; else, it is considered as an integer-data file). Additionally, the contents (ASCII, octal, decimal) of one or more blocks can be obtained. On the other hand, there can be written one or more test-blocks (a sequence of ASCII symbols) with user-set length; they can be "arranged" in a file writing a tape-mark (either EOF or EOV). Files or blocks skipping is assured as well as the tape can be rewound or zeroed in this regime.

These functions are assured by a set of ten U-commands. In this regime, by a G-command, there can be created number of files imitating (by a standard way) a coude-spectrogram scan-records. Such files are intended only for FRM-tests.

3.5 Miscellaneous functions

The commands can be inserted in an interactive dialogue or by a command-file in an indirect-input regime (see 5). The latter must be activated by a particular @-command; the command-file's name and the command-row's pointer can be fixed, optionally. On the other hand, there can be obtained an information about the current set up name and pointer, as well as the file can be listed out by other @-commands.

A runtime help can be obtained by H-commands. The whole available help-text can be saved in a special text-file for further printing, or it can be interactively, page by page, displayed; the separate items of the help can be displayed, as well. This is done on the basis of a special help-file, the FRM program creates when it is started.

Finally, by an E-command, the user can exit from the FRM

(to stop it). This detaches still attached volumes, closes a dialogue-copy file (see 5.4) that optionally can be deleted, messages the program is completed, and stops it.

4. FRM PROGRAM REALIZATION

The FRM program is created on the PDP 11/34 minicomputer under the FORTRAN-IV-PLUS language ("PDP-11 FORTRAN", 1979). It consists of a program-dispatcher and a set of subroutines it calls to execute the particular function, as fixed by an internal flag. A general module for tape-control primitives (on basis of the system's WTQIO directive) is included, too. The general scheme of the particular modules interconnection is illustrated on the Fig.2. All the source-code modules are included in a general file FRM.FTN (the "FRM-CODE" document contains its full printing); the latter is translated by the F4P compiler ("FORTRAN IV PLUS", 1975). Then, the system's LBR utility ("RSX-11M", Utilities, 1979) is used to create, from the obtained object-modules file FRM.OBJ, a library of the modules, the FRM.OLB file, necessary to build the task.

The executable module FRM.TSK is then created by the TKB ("RSX-11M/M-PLUS", Task Builder, 1979), using an overlaying the system maintains due to memory restriction for a task's size (32 Kw). The FRM-overlaying is described in a special file FRM.ODL; the resulting task image-file is about 150 Kw, whereas the memory-resident task occupies, really, no more than 16 Kw at a time.

A special command-file FRM.CMD assures the FRM creating (merging the particular modules to create the source-code file, compiling it, creating the library, and building the

task). The compiler FTN or F77 can be used as well (the FTN limits the allowed scan's size to 32767 records; generally up to 65535 scan-data size is allowed).

5. FRM PROGRAM USING

The particular function is activated by a fixed command the user inserts during an interactive dialogue or the FRM reads in a special, user-prepared in advance, command-file in an indirect-input mode. Regardless the input-mode, the task prompts it is ready for a command, accepts it, outputs a particular information during its execution, and verifies it when successfully completed or outputs an error-message if the execution failed.

5.1 Command-string

The command-string has a standard format; it consists of three parts: a code and switch fixing the very command, and optional parameter. The command-code is an upper-case or the "@" symbol coding the command's general function, whereas the following it switch (two upper-case letters) fixes the particular one, to be executed. These two elements are the obligatory part of the command-string.

The optional parameter depends on the particular command; it can be a number, a textual one (enclosed by apostrophes), a single symbol, or a combination of them. If some allowed parameter is not included in the command, the program uses a particular defaulted value.

The allowed FRM-commands (without the parameters) and the corresponding general functions are listed bellow:

CODE	COMMANDS	GENERAL FUNCTION
@	@CO, @IN, @LI	indirect-commands processing
A	ATT	attaching a disk-device
C	OFF, CLE, CVD	clearing in the disk-volume
D	DIS, DVD	disk-to-tape reformatting
E	EXI	exiting from the "FRM" routine
F	FDT, FTD, FUL	tape-disk-tape FF-reformatting
G	GEN	generating tape-files for tests
H	HEL	help-information outputting
I	INF, IPL, IVO	informing about the tape-volume
L	LFF, LHE, LIS, LVD	listing of the disk-volume
M	MAT, MDE, MRW, MZR	magtape-device initialization
N	NAM, NVO	naming a disk-file as current
P	PBA, PFD	positioning in the tape-volume
S	SFI, SPL	skipping in the tape-volume
T	TAP, TPL, TVD	tape-to-disk reformatting
U	UAC, UBY, UFI, UGB, UPB	tape-unit U-regime functions
	URW, USB, USF, UTM, UZR	

5.2 Interactive-dialogue regime

Starting the FRM, this regime is automatically switched on. Initially, the FRM displays a title-message and writes a special help-file, if does not yet exist. Then, it messages it is ready for input, the user inputs a particular command that is accepted and executed; the ready prompt is displayed again, and so on, until the exiting is desired.

More precisely, the command-string is checked (the code, unpaired apostrophes, delimiting spaces) if allowed, and if this is the case, the particular module is activated (a flag is set depending on the code). Else, a fixed error-message is displayed and then, the ready-prompt for another command.

There is carried out another checking in the particular module for the switch and optional parameters, and for the FRM's status (an internal status-word is being continuously saved fixing the current devices and files status). If this fails, the control is returned for a fixed error-message and for another command (like above); else the fixed function is activated. During its execution, for most of the commands, there is outputted specific information. When the execution is successfully completed, a verifying message (the executed command-string and the updated status) and the ready-prompt are displayed. But if some fault occurred, the corresponding error-message (and the status), like above, is outputted.

The FRM can be stopped either by the "EXI" command or by pressing the CTRL and Z keys simultaneously. This performs appropriate final actions (see 3.5) and stops the program.

5.3 Indirect-commands-input regime

This special regime permits an automatic execution of a fixed sequence of commands. The FRM reads the commands in an user-created (in advance) text-file. The latter must contain "command-rows" (a command per row), sequentially ordered as to be executed. They would be arranged in "command-blocks", delimited by special break-rows (the "@" symbol in the first column). The file-name can contain up to six-characters name

and an optional three-characters extension (the "FRMCMD.TXT" is the initially defaulted name).

The indirect regime is activated by the "@CO" command, by the parameters of which, the user can fix the desired file's name and/or to point the command-row from where to start the run. Now, the program runs as in the dialogue regime, but after the ready prompt, it automatically reads the pointed command-row and increases the pointer, executes the command, reads the next, and so on. This cycle can be interrupted in one of the three cases: if an error occurs (command-string, status, execution); if a breaking-row is encountered (it is simply displayed); if the command-file's end is reached. In any case, the interactive regime is reset and the ready prompt is displayed.

Two additional @-commands are usable (see 5.1) to obtain an information about the current command-file and pointer as well as to list the current command-row or block. Of course, the @-commands are available only in the dialogue regime.

5.4 Program's output information

During its run, the FRM displays a particular information about its status, the ready-prompt, the command's execution, verifying or error-message; for most of the commands, there is outputted specific information during the execution. All this output is simultaneously directed to the user's display and to a special text-file "FOR004.DAT" the program creates whenever started. For some of the commands, the displaying during the execution can be suppressed (by a fixed optional parameter); nevertheless, the information will be written in

this file. On the other hand, the runtime help commands do not copy the displayed help-information, as one of them is intended to create a special, help-text file "FRMHELP.TXT" with the full help-information available.

Exiting, the FRM closes and saves the dialogue-copy file, if its deleting has not been optionally desired. Then, the file can be displayed and/or printed out. Such printings are useful to keep archives for the FRM using, for the volumes contents, etc.

6. CONCLUSIONS

The FRM program assures the reformatting of the recorded in the NAO spectrograms, to save them in format optimal for digital processing by the S-software's programs or in one, more suitable for digital images flexible transport to other institutes. The latter, the "STANDARD" format, is necessary condition to create spectrograms tape-library in the NAO.

The additional system of U-functions can be used as for testing some error-conditions in tapes handling, so to check free-format tapes. The dialogue-copy file permits to create archives of the processing and of the volumes. On the other hand, there can be performed a standard running with minimal user-control applying the indirect regime.

The runtime-help gives enough information to use the FRM by an experienced user, whereas the detailed documentation is intended for the beginners. The latter contains also the information necessary to study and update the software.

The applied module principle for the program's software allows to update the commands, to include new functions, or

to create similar reformatting routines. On this basis, a FITS-reformatting program has been created (Bellas, Bonev, 1988) for the two-dimensional scans of the P/HALLEY comet images obtained in the NAO.

Acknowledgments: The author would like to thank Y.Borisova for her remarks on the program and this paper, as well as for her kindly assistance during the tests.

REFERENCES

Bellas Y., Bonev T., (1988, to be published)

Bellas Y., Borisova Y., Astrophysical Investigations, 1988
(in press)

"FORTRAN IV PLUS", User's Guide,
DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts,
(Dec. 1975)

"PDP-11 FORTRAN", Language Reference Manual,
DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts,
(Dec. 1979)

"RSX-11M", Utilities, chapter 14,
DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts,
(Jun. 1979)

"RSX-11M/M-PLUS", Task Builder,
DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts,
(Jun. 1979)

Wells D.C., Greisen E.W.,
"FITS: A Flexible Image Transport System",
Astronomy and Astrophysics Supplement Series,
vol. 44, No 3, 363-370, (1981)

FIG. 1 STANDARD HEADER

An example of a real scan-file's header after restoring the stellar-spectrum recorded in it. Only the filled up rows are shown; the rest of the header (rows 29-36 as well as columns 77-80) is complemented with spaces. Rows and columns numbers are given for clearness and they aren't parts of the header.

FIG. 2 FRM PROGRAM'S MODULES

- a. The modular structure of the FRM program realization, the modules functions and interconnections.
- b. functional flow-chart of the FRM program; "flag" is an internal parameter being set during command-string analysis (by the FRMCOM module) and being updated by the particular module activated accordingly to its value.

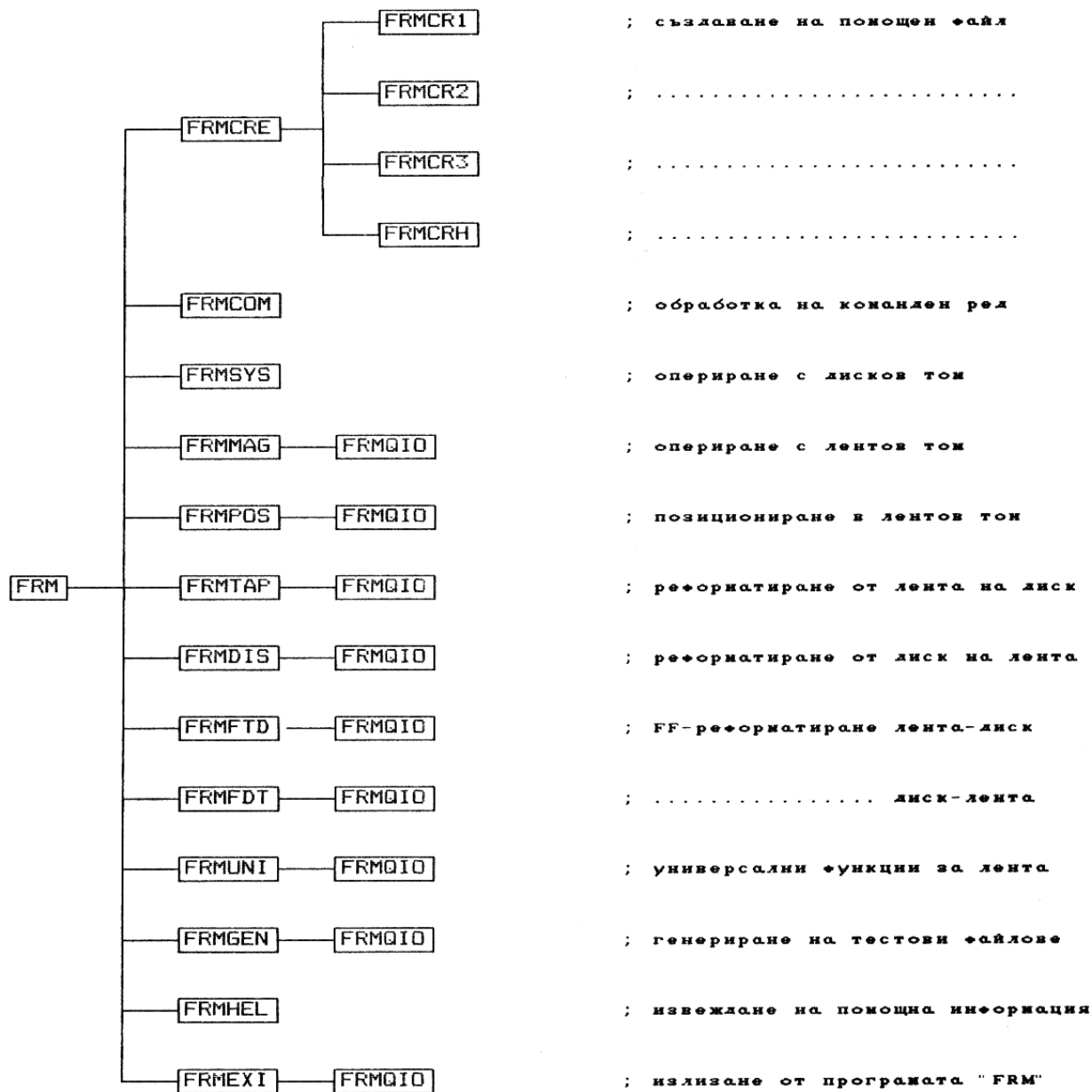
```

01 SIMPLE = T / STANDARD FITS FORMAT: "SCN" FEB-87
02 BITPIX = 16 / 2 BYTE TWOS-COMPL INTEGER: 16 BITS
03 NAXIS = 1 / 1 AXIS (SINGLE-DIMENSION-SCANNING)
04 NAXIS1 = 30068 / NUMBER OF PIXELS IN THE AXIS
05 ORIGIN = 'BNAO - ROZHEN' / BULGARIAN NATIONAL ASTRON. OBSERV.
06 TELESCOP= '2M-RCC/ COUDE' / OBSERVING INSTRUMENT
07 OBJECT = '2K0103 HD108 ' / THE IDENTIFIER (PLATE,OBJECT)
08 ALPH-OBS= ' 00/05/04 ' / OBS. RIGHT ASCENSION [HH/MM/SS]
09 DELT-OBS= '+63/34/22 ' / OBS. DECLINATION [DD/'/'/'"]
10 TIME-OBS= ' 22/00/20 ' / OBS. UT-MOMENT [HH/MM/SS]
11 DATE-OBS= ' 80/10/23 ' / OBS. UT-DATE [YY/MM/DD]
12 DISPER = 9.0 / PLATE'S ROUGH DISPERSION [A/MM]
13 INSTRUME= 'ESO: GRANT-MACHINE' / SYSTEM WHICH SCANNED THE OBJECT
14 DATE = ' 83/05/22 ' / DATE THE DATA SCANNED [YY/MM/DD]
15 BSCALE = 0.001 / LOG10(REL.INTENS.) = DATA * BSCALE
16 CDELTA1 = 0.0450 / INCREMENT ALONG X (WAVELENGTH) [A]
17 WAVELENG= 3474.585 / ORIGIN-PIXEL-CENTER WAVELENGTH [A]
18 JULDAT = 2423411.842621 / HELIOCENTRIC JULIAN-DATE [days]
19 RVCORR = 13.57 / HELIOC. RAD-VEL. CORRECTION [km/s]
20 COMMENT : OBSERV. EXPOS.= 74:40 [m:s], SP-SLIT= 6*0.25 [mm], SEEING= ~3.0"
21 COMMENT : EMULSION= 103a0 , unsensitized (MWP-2 , 10 [min])
22 COMMENT : SCANNING PIXEL= SLIT/27.5 : WIDTH= 0.275 [mm], HEIGHT= 10.0 [mm]
23 COMMENT : FEAR-HEIGHT= 2 * 15.2 [mm], HLGN-HEIGHT= 3.5 [mm]
24 HISTORY : "FRM" 08-SEP-87 11:30 (REFORMATTED FROM ESO-FITS TO STANDARD-FITS)
25 HISTORY : "FLT" 08-SEP-87 13:54 (NOISE-REDUCTION BY OPTIMAL-FILTER APPLYING)
26 HISTORY : "RST" 08-SEP-87 14:22 (RESTORED TO REL.INTENSITY VERS. WAVELENGTH)
27 COMMENT : HELIOCENTRIC WAVELENGTH SCALE: LAMBDA = LAMBDA / (1+RVCORR/3000000)
28 END

```

фиг. 1

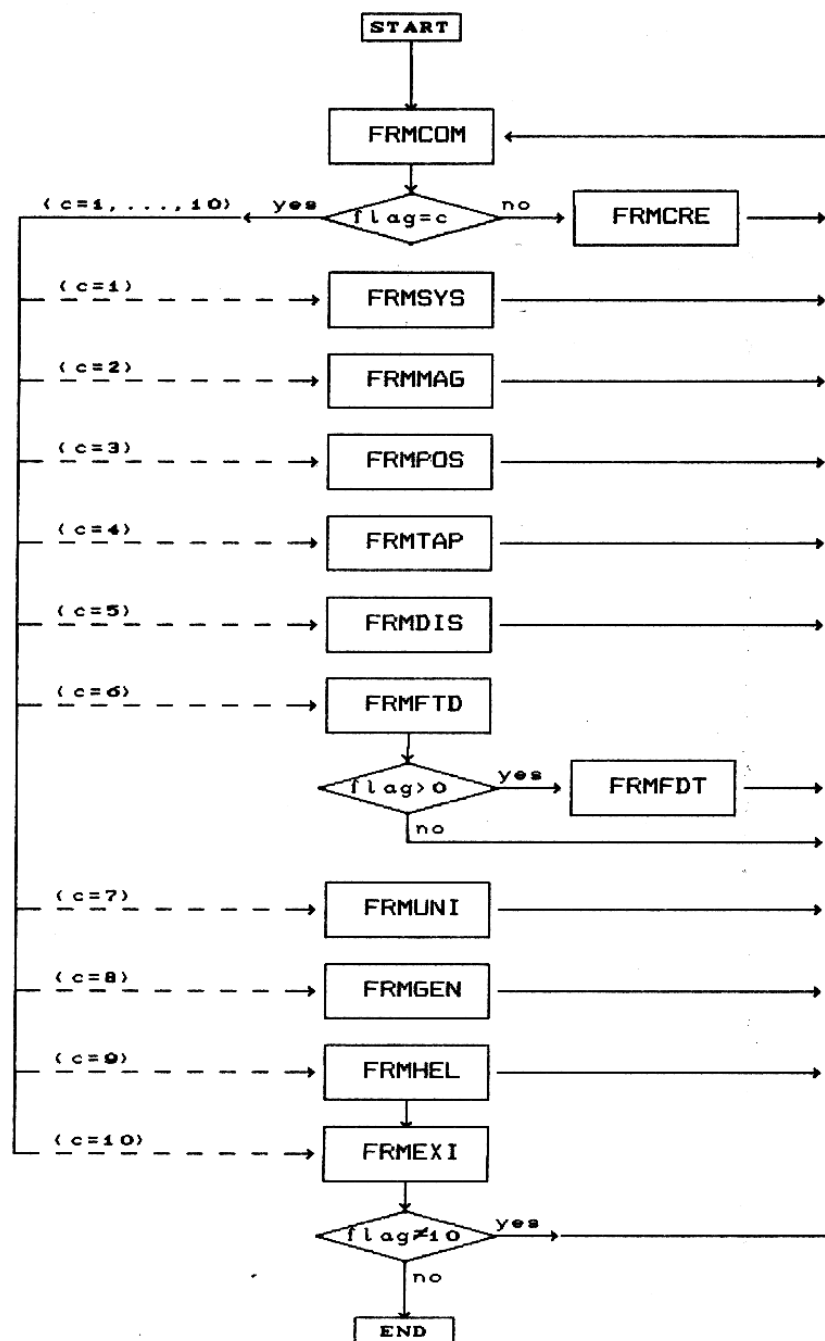
Пример на хедер на скан-файл след обработка (възстановяване) на записа на звездния спектър. Показани са само запълнените редове; останалата част на хедера (редовете 29-36, колоните 77-80) е допълнена със шпации. Номерата на редовете и колоните се дадени само за нагледност и не са част от хедера.



(модулът "FRMQIO" осигурява елементарните функции за лентовото у-во)

фиг. 2а

Модулната структура при реализацията на програмата FRM,
функциите на модулите и връзките между тях.



фиг. 2b

Функционална блок-схема на програмата FRM за реформатиране; "flag" е един вътрешен параметър за състоянието, установяван при обработката на команден ред (в модула FRMCOM) и обновяван в задействания под управлението му модул.