DECUS
PROGRAM LIBRARY

DECUS NO.	12-62
TITLE	RUFUS
AUTHOR	David M. Stern
COMPANY	Laboratory for Atmospheric and Space Physics University of Colorado Boulder, Colorado
DATE	January 13, 1971
SOURCE LANGUAGE	PAL

RUFUS

DECUS Program Library Write-up DECUS NO. 12-62

"RUFUS" is an interactive display-oriented programmable spectral analysis system for the PDP-12. The system is heavily display oriented enabling the user to get a much better "feel" for his data than is otherwise possible. Five-hundred and twelve point spectra or vectors are the basic data item and are stored in six registers in core. These spectra may be:

added and subtracted
multiplied and divided
smoothed
edited
correlated and fitted
plotted
Fourier transformed, etc.

The "language" of RUFUS consists of a large number of basic commands that may be entered manually or run under program control. A limited number of floating point and integer variables (not more than 26) are available for arithmetic, indexing and statement labels.

Minimum configuration for "RUFUS" is a PDP-12 with 8-k of core and two Linc Tape drives, "RUFUS" will optionally support the KW-12A real time clock, a Calcomp 565 plotter, a card reader and an LP-8 line printer. All devices are run under interrupt to maximize the time the display is active.

Six spectra may be stored in memory at once in Spectrum Registers 0 to 5. Spectra may be input to registers from Linc Tape unit number one or from the card reader in binary form. Registers can be output to Linc Tape 1, the teletype, plotter or line printer.

Spectra are stored on Linc Tape Unit 1 using two blocks per spectrum beginning on each even block. 256_{10} or 450_{10} spectra may be stored on one tape, for standard linc tapes or 896_{10} block tapes respectively.

Instructions to "RUFUS" are in the form of "commands" entered from the teletype. Commands are interpreted in three modes: 1) manually - commands are executed as they are typed; 2) automatically - commands are stored in a 1024 character buffer and executed under program control with variables, loops, etc.; 3) storage mode - commands are stored in the program buffer as they are entered from the TTY and usually executed. (This is similar to "FOCAL".)

Programs may be written using the text editing features of the "DIAL" system and loaded by "RUFUS" from either a named DIAL file or from the working storage area. Also

RUFUS programs can be saved by RUFUS onto previously existing DIAL source files. These programs are completely compatible with the DIAL text editor.

The Display

Every effort has been made to make the display as interactive and informative as possible. One or two spectra may be displayed at once, selected directly by the operator or automatically by "RUFUS" as the result of a command. The x-axis can be expanded from 512 points to two points across the entire screen depending upon the setting of analog channels 1 and 5. The vertical separation between the two displayed spectra or the position of one spectra is controlled by knob 4. Full screen range vertically is 512 normally, or if sense switch 2 is depressed all y-values are divided by four before they are displayed giving a range of 2048.

The register number for each spectrum is displayed about 1½ inches above the x-axis at the left of the screen.

There are four variable features of the display, one is the vertical spacing or position controlled by knob 4. The other three are the cursor (knob 0) and the high and low expand limits, "HIEXPD and LOEXPD" (knobs 1 and 5). These three features are used by various commands for parameters or may be set under program control by other commands.

The "cursor", set by knob 0, is the one intensified point in each register at the same x-value. This x-value is determined by the setting of knob 0. The y-value of the cursor is the value of the displayed register at that point.

The two vertical dashed lines are LOEXPD and HIEXPD from left-to-right. If these ever cross, i.e. LOEXPD $\geq$ HIEXPD, the display will go blank until this is corrected. If sense switch 1 is set to one, the portion of the displayed registers between LOEXPD and HIEXPD is expanded to fill the screen horizontally. The portion between LOEXPD and HIEXPD may be varied continuously. If sense switch 1 is zero, the full 512 points will be displayed.

Sense switch 0 determines whether one or two spectra are displayed. If it is zero, two spectra are displayed. If it is one, only one spectrum is displayed with this additional feature: four numbers are displayed along the bottom of the screen. They are from left to right: 1) LOEXPD x-value, 2) cursor x-value, 3) cursor y-value and 4) HIEXPD x-value. When SNS 0 = 1, only the "UPPER DISPLAY" register is displayed. When SNS 0 = 0, both the "UPPER DISPLAY" and "LOWER DISPLAY" registers are displayed.

SNS 3 controls the input and output buffer displays which appear above the numbers on the bottom of the screen.

A summary of the display controls follows:

SENSE SWITCH 0 - dual or single register display

1 - normal or expanded display

2 - normal or y/4 scaling

3 - no buffer display or both buffers displayed (usually set to 1)

ANALOG KNOB 0 - cursor x-value

1 - LOEXPD x-value

4 - y position

5 - HIEXPD x-value

If when RUFUS is first loaded and there is no display, rotate knob 1 counter clockwise and knob 5 clockwise until the display returns.

Arithmetic

Twelve bit signed integers are used for spectrum data values, allowing a range of -2048_{10} to $+2047_{10}$. Y-values above the top of the screen or below the bottom will wrap around the screen.

Floating point variables or constants are standard DEC three-word with a range of $\pm 10^{600}$ and around 6 place precision. When a floating point variable is converted to an integer, it is truncated to a whole number unless it is outside the range of integers in which case it is set to $+2047_{10}$.

Structure

All commands are stored in the form of overlays on a file called "KILLER--". This DIAL binary file must always be on unit 0 for RUFUS to run. When first loaded, RUFUS searches the DIAL directory for "KILLER--" and for files whose name begin with the character "&" (ampersand). Those files, whose name starts with "&", may contain wavelength scales or "FRANCK-CONDON" binary information and will be called "RUFUS FILES". RUFUS finds the starting blocks of these files and saves this information on "KILLER--" eliminating an extra tape movement when RUFUS files are needed. "KILLER--" should reside as close to the middle of the tape as possible to save searching time.

Each command overlay is 3000_8 words long and there are currently 4 overlays. At the end of "KILLER--" two blocks are reserved for the current wavelength scale making "KILLER--" 4 overlays * 3000_8 words/overlay * $1/400_8$ blocks/word + 2 blocks = 32_8 blocks long.

Overlays are read only when the command requested is not in core, therefore, programs should be written to minimize excessive overlay loading. The most frequently used and most basic commands are in the first overlay.

The source of "RUFUS" consists of 9 files:

"MONST" - the display segment

"INTSRV" - interrupt service routines, general purpose subroutines used by all commands, overlay loading, etc.

"PTABLE" - the command table

"RUFINIT" - initialization routines that are used only once and then overlayed by commands

"OVER1" - overlay 1 source

"OVER2" - overlay 2 source

"OVER3" - overlay 3 source

"OVER4" - overlay 4 source

"MCRUF" - the source for the user's monitor command that writes the binary commands onto "KILLER--" from the Binary Working Storage Area on Unit 1.

There are 7 binary files:

RUFUS - the combination of MONST, INTSRV, PTABLE, RUFINIT and FPNT combined using the DIAL "AB" command. RUFUS starts in Linc mode at address 4020.

KILLER - overlays 1 to 4 and wavelength scale

MONST

INTSRV

PTABLE

RUFINIT

FPNT - the DEC three word floating point package modified for "RUFUS"

CORE MAP

17777	
	scratch and wavelength files
17000	
	program storage
16000	
	registers
10000	
	character patterns
7600	
	floating point package
5600	
	initialization and then overlay commands
2600	
	command table
2400	
	INTSRV
1000	
	display
200	
	linkage and variables
0	

Variables

RUFUS has 26_{10} variables ("A" to "Z") that may be integers, program labels or floating point variables. A variable may be used wherever a constant is used and may be preceeded by a minus sign. A letter may have only one of the three uses described above in any one program. Floating point variables occupy three locations starting with the name of the variable. The floating point variable "X" uses locations "X", "Y" and "Z". These locations may not be used for anything else. Caution must be used to insure that a floating point variable does not destroy integer variables or labels.

Command Format

Each command has a one or two letter name followed optionally by one or more parameters and terminated by a carriage return.

The character (RUBOUT) will erase the entire line and echo "<<". The maximum line length on input is 30 characters. Legal characters are the letters A - Z, the digits 0 - 9, the minus sign -, rubout, carriage return and the character "CNTRL/R".

If a "CNTRL/R" is typed any time the display is active, "RUFUS" will halt whatever it is doing and return to the normal manual mode. If a program was running, it may be necessary to alternately type "CNTRL/R" and rubout until the input buffer is erased.

Throughout the command descriptions, the letter R refers to a register number 0 to 5, either a variable or a constant. C refers to an integer constant or variable, either signed or unsigned. F is a floating point constant or variable. Some legal forms of floating point constants are:

+100	-1.E2
+100.	1.0E2
+.01E-03	.0032

The letter "L" refers to a label which may be any letter not used by integer or floating point variables.

Spaces are required only after the last digit of an integer or floating point constant unless it is the last character of the line. Omitted arguments are assumed zero.

Some examples follow:

1) RD 0 0

2) RD

3) RD0 0

All of these read spectrum 0 from Linc Tape 1 into Register 0.

Assume A has been set to 4 and B has been set to 39.

4) RD 4 39

5) RD4 39

6) RD A B

7) RDA B

8) RDAB

9) RD4 B

10) RDA39

All the above commands will read spectrum 39 into register 4.

11) RD439

12) RD4B

Both of the above will produce an error message. Number 11 will be interpreted as RD 439 0 and number 12 would be indeterminate.

13) ED 0 -39

14) ED 0 -B

Both number 13 and number 14 would be interpreted as edit the cursor y value in register 0 to -39.

If in the command descriptions the format R_n is used, it refers to every point in the register R_n. If $R_{n,i}$ is used, the ith point in register R_n is meant. The subscript i goes from 0 to 511_{10}.

For example, in the "AD" (add) command, AD R_1 R_2 R_3, $R_3 = R_2 + R_1$ means that $R_{3,i} = R_{2,i} + R_{1,i}$ for $i = 0$ to 511.

ARITHMETIC COMMANDS

AD $R_1\ R_2\ R_3$ Register Add

$$R_3 = R_2 + R_1$$

AV $R_1\ C_1$ Moving Average

This replaces R_1 by its smoothed result. The average is obtained by convolving a rectangle with a width of C_1 with R_1. C_1 must be odd.

$$R_{1,i} = \frac{1}{C_1} \sum_{j=k}^{L} R_{1,j} \quad \text{for } i = [C_1/2] \text{ to } 511 - [C_1/2]$$

$$k = i - [C_1/2] \quad L = i + 2[C_1/2]$$

BK $R_1\ C_1$ Background

$$R_1 = R_1 + C_1$$

CC $R_1\ R_2$ Correlation Coefficient

This finds the correlation between R_1 and R_2 between the limits, sets the floating point variable X to the correlation and prints the result if the listing switch is on.

$$r = \frac{\Sigma (R_{1,i} - \bar{R}_1)(R_{2,i} - \bar{R}_2)}{\left(\Sigma(R_{1,i} - \bar{R}_1)^2\ \Sigma(R_{2,i} - \bar{R}_2)^2\right)^{\frac{1}{2}}}$$

CN $R_1\ R_2$ Condense Scale

This command condenses the x-scale of R_1, so it fits between the limits of R_2. It is handy for making the wavelength scales of two different instruments the same.

CR Clear Registers

Zeroes registers 0 to 5.

DP R_1 R_2 Dot Product

The vector dot product of R_1 and R_2 between the limits is calculated and put into the floating point variable X.

$$X = \sum_{i=LOEXPD}^{HIEXPD} R_{1,i} * R_{2,i}$$

DS R_1 F_1 Divide Scalar

R_1 is divided by F_1 and replaced by the result.

$R_1 = R_1/F_1$

DV R_1 R_2 F_1 Divide Vector

Every point in R_1 is divided by the corresponding point in R_2 and then multiplied by the scaling factor F_1. The result replaces R_1.

$$R_{1,i} = \frac{R_{1,i}}{R_{2,i}} * F_1 \qquad \text{for } i = 0 \text{ to } 511.$$

ED R_1 C_1 Edit Point

The point on the cursor, $R_{1,\ cursor}$, is replaced by C_1.

EX R_1 R_2 Expand Scale

The portion of R_1 between the limits is expanded horizontally to a full 511 points and put into R_2. R_1 can not be equal to R_2.

FD R_1 R_2 Form Derivative

The derivative of R_1 is put into R_2.

$R_1 \neq R_2$.

$$R_{2,i} = \frac{R_{1,i+1} - R_{1,i-1}}{2}$$

FF R_1 R_2 C_1 Fast Fourier Transform

This command does a Fourier transfrom of a 512 point complex vector. R_1 is the real component, R_2 is the imaginary component - they are replaced by the complex result. For real vectors, R_2 must be zeroed. C_1 is negative for the original transform and positive for the inverse transform. The operation takes 75 seconds.

FH R_1 R_2 Form Histogram

A value versus frequency histogram of R_1 is formed in R_2. Points in R_1 are taken MODULO 511.

$R_2 = 0$

for $i = 0, 511; \; j = R_{1,i} \, \mathrm{MOD}_{511}; \; R_{2,j} = R_{2,j} + 1$

This is useful in finding noise distributions, etc.

FI R_1 R_2 F_1 Form Integral Function

$0 \rightarrow R_2$

for i = LOEXPD to HIEXPD

$$R_{2,i} = \frac{F_1}{\mathrm{HIEXPD} - \mathrm{LOEXPD}} \sum_{j=\mathrm{LOEXPD}}^{i} R_{1,j}$$

FL R_1 F_1 F_2 Form Line

A straight line is formed in R_1 with a slope of F_1 and an intercept of F_2 between the limits

$$0 \rightarrow R_1$$

$R_{1,i} = F_1 * i + F_2$ for i = 0 to 511.

HP R_1 C_1 C_2 High-Pass Clipper

All points in R_1 lower than C_1 are edited to C_2.

IN R_1 Integrate

R_1 is summed between the limits and the result put into floating point X. If the listing switch is on, the result is printed.

LF R_1 Least Squares Fit

A straight line is fitted to R_1 between the limits giving the parameters U and X for the equation $Y = U * i + X$. This command may be followed by the command FL R_n U X producing the line that has the best fit. If the listing switch is on, then the slope and intercept are printed. The values of U and X are stored in their corresponding floating point variables.

$$\text{Slope} = U = \frac{\sum_{i=\text{LOEXPD}}^{\text{HIEXPD}} (R_{1,i} - \overline{R}_1)(i - \overline{i})}{\sum_{i=\text{LOEXPD}}^{\text{HIEXPD}} (i - \overline{i})^2}$$

$$\text{Intercept} = X = \overline{R}_1 - U * \overline{I}$$

LP R_1 C_1 C_2 Low Pass Clipper

All points in R_1 greater than C_1 are changed to C_2.

LS R_1 C_1 Left Shift

The points in R_1 are shifted left C_1 points or sample numbers. Points shifted off the left side, sample number 0, appear on the right side, sample number 511. Hence, the shift is circular. C_1 may be any positive number < 511.

MS R_1 F_1 Multiply Scalar

The points in R_1 are multiplied by F_1.

$R_1 = R_1 * F_1$

MT R_1 F_1 F_2 Make a triangle

A triangle with an area of F_1 and a half-width of F_2, centered on the cursor, is added to R_1.

$R_1 = R_1 +$ triangle (centered on cursor)

Base of triangle $= 2 * F_2$

Area of triangle $= F_1$

Height of triangle $= F_1/F_2$

MV R_1 R_2 F_1 Multiply Vector

Every point in R_1 is multiplied by the corresponding point in R_2 and then divided by F_1. The result replaces R_1.

$$R_{1,i} = \frac{R_{1,i} * R_{2,i}}{F_1} \quad \text{for } i = 0 \text{ to } 511.$$

NM R_1 R_2 Normalize

R_1 is multiplied by a scalar so that at the cursor $R_1 = R_2$.

$$R_1 = R_1 * \frac{R_{2,cursor}}{R_{1,\ cursor}}$$

NR R_1 C_1 Noise Removal

The point in R_1 at the cursor is replaced by the average of the points C_1 from that point.

$$R_{1,cursor} = (R_{1,\ cursor - C_1} + R_{1,\ cursor + C_1})/2$$

PW R_1 R_2 R_3 Power Spectrum

The power spectrum is calculated from the results of a Fourier transform in R_1 and R_2. This puts the norm of the complex vector (R_1,R_2) into R_3.

$$R_{3,i} = \sqrt{R^2_{1,i} + R^2_{2,i}}$$

RS R_1 C_1 Right shift

The contents of R_1 are shifted right C_1 places. The shift is circular. This is the same as left shifting R_1, 511 - C_1 places.

SU R_1 R_2 R_3 Subtract

R_2 is subtracted from R_1 and the result is placed in R_3. R_3 may be the same as R_1 or R_2.

$R_3 = R_1 - R_2$

WS F_1 Wavelength Set

The cursor is set to the sample number with the wavelength closest to F_1. The current wavelength scale is used. This command has the same effects as the SC command.

XR R_1 R_2 Transfer

The contents of R_1 are put in R_2. R_1 is not affected.

ZR R_1 Zero Register

R_1 is set to all zeroes.

DISPLAY COMMANDS

RC - Restore Cursor

The control of the cursor is returned to POT 0.

RL - Restore Limits

The control of the limits is returned to POTS 1 and 5.

SC C_1 - Set Cursor

The cursor is set to sample number (x-value) C_1. POT 0 has no effect on the cursor until an RC command is given.

SH C_1 - Set High Limit

The high expand limit is set to C_1. Control of the limits is taken away from the POTS 1 and 5 until an RL command is given.

SL C_1 - Set Low Limit

The low limit is set to C_1 using the same rules as the SH command.

INPUT - OUTPUT COMMANDS

CD R_1 C_1 Input Cards

Binary cards are read and put into R_1. R_1 is zeroed first. The first C_1 data columns are skipped. The card format is the standard CDC 6400 - 6600 binary deck format. Column 1 has a 7 - 9 punch for a data card or a 7 - 8 - 9 punch for an end of record card. An end of record card terminates the reading. Column 2 is the checksum which is ignored. Columns 3 to 77 are Data Columns in 12 bit binary. Columns 78 and 79 are ignored and Column 80 is the serial number of the card in the record. If the serial number is non-zero, then it is checked and if it is out of sequence, the reading is terminated and an error message is printed. Data Columns after 512 + C_1 and before an EOR are ignored.

CW FILENAME Change Wavelength Scale

The wavelength scale is changed to the contents of "FILENAME". The file must be a RUFUS file with an & (AMPERSAND) as the first character.

FILENAME is the name of the Wavelength file with the ampersand omitted. The file should be a binary file two blocks long, with each word the wavelength of the corresponding sample number -900_{10} in 12 bit unsigned integer form. A range of 900_{10} to $900_{10} + 4095_{10}$ or 4995 is possible. The wavelengths must be monotonically increasing.

Wavelength scales available are:

G-CHL - Mariner 6 and 7 G-CHL

N-CHL - Mariner 6 and 7 N-CHL

G-CHL9 - Mariner 9 G-CHL

F-CHL9 - Mariner 9 F-CHL

LINEAR - 4 Angstroms per point

plus a number of OAO wavelength scales.

Wavelength scales that do not go over 900 + 2047 = 2947 Angstroms can be produced and saved with FOCAL-12. For scales that go over 2947 Angstroms RUFUS must be used to write the scale on a DIAL file.

DI Return to DIAL

This command will leave RUFUS and start the DIAL system.

FC Franck-Condon Commands

These two commands are used in producing synthetic ultraviolet spectra from information stored on RUFUS files. Most of the data is from Charles A. Barth, Ultraviolet Spectroscopy of Planets, NASA Technical Report No. 32-822, 1965.

The format of the Franck-Condon RUFUS files are as follows:

Word 1	Number of molecular transitions in this file
Words 2 - 4	A three word floating point number used as a scale factor for the strengths of the transitions (under variable "Q" in Barth)

Molecular Transitions (4 words)

Word 1	Upper 6 bits v' transition Lower 6 bits v" transition
Word 2	Wavelength of transition - 900 Å
Words 3 and 4	Strength of transition ("Q") expressed as a 23 bit integer.

FCM R_1 C_1 C_2 F_1 Franck-Condon Make

The spectral line or lines specified in this command are added into the register R_1. C_1 specifies v' for the line and C_2 is v". F_1 is the population for the line or lines usually in the range of 0 to 1. If either C_1 or C_2 are negative, then all FC factors found for that variable will be used. If both are negative, then all factors will be used up to a maximum of 42. If a particular line has a wavelength less than the minimum wavelength or greater than the maximum wavelength on the wavelength scale, then the line will be made at both edges of the register. Some examples follow:

To make the v' = 0, v" = 2 line for the CO Cameron band system in Register 0.

```
FCS CAMERON 5.5 1.E3
FCM 0 0 2 1.
```

(using a resolution of 5.5 and a population of 1. into Register 0.)

To make in Register 2 all lines with a v' of 1 from the N2LBH bands with a resolution of 4:

FCS N2LBH 4. 1.E3

FCM 2 1 -1 .67 (population of .67)

To make all of the CO^+ 1st negative (up to 42) in Register 5:

FCS COP1N 5.5 2.5E4

FCM 5 -1 -1 1.

In all the examples the destination registers should be all zero unless it is desired to add the lines to a previous result. As many FCM commands may be given as desired without having to give an FCS command unless the band system is changed.

FCS FILENAME F_1 F_2 Franck-Condon Set

This command is one of two Franck-Condon commands. All future Franck-Condon commands, until the next FCS command will use the data supplied by this command. "FILENAME" is the name of the RUFUS file where the FC data is contained, ordinarily the name of a band system. F_1 is the floating point resolution used in constructing the spectral lines. F_2 is a scale factor for the area of all the lines. A F_2 of around 1000. usually scales the peaks into the region of 125 - 512 for most band systems. The following Franck-Condon factor/systems are now available:

BAND SYSTEM	RUFUS FILE
N_2^+ 1st negative	N21NEG
N_2 2nd positive	N22POS
N_2 Vegard-Kaplan	N2V-K
NO Gamma	NOGAMMA
CO^+ 1st negative	CO1NEG
CO 4th positive	CO4POS
CO Cameron	CAMERON
N_2 LBH	N2LBH

More systems may be easily added. Care must be taken that the wavelength scale is the appropriate one for the band system.

F8 R_1 Receive Spectrum from the PDP-8

A 512 point spectrum is received from the PDP-8 via the serial 10 mega-herz line and put into R_1. Typing CNTRL/R will halt the command, if necessary.

Gb C_1 Get Spectrum

Spectrum number C_1 is read into Register 0 from unit 1. This is an abbreviated from of the RD command. A blank must follow the "G".

LI	List	
		This command turns on the listing switch. CNTRL/R, or a NL command will turn it off.
NL	No List	
		Turns off the listing switch.
O	Read One Spectrum Backwards	
		If spectrum 100 was read or written last, typing "0 (RETURN)" will read spectrum number 99 into Register 0.
OP	R_1 C_1 Overplot	
		See PL command
OS	R_1 C_1 Overplot Squares	
		See PL command
P	Read One Spectrum Forwards	
		This is the same as the "0" command except it reads the next spectrum.
PC	F_1 Plotting Constant	
		This command sets the wavelength axis length for the next plot command (PL, PS, OP, OS) to F_1 inches. The length is 8.5 inches if no PC command was given previously.
PI	V_1 Print Integer	
		The value of the integer variable V_1 is printed.

PL R_1 C_1 Plot

The data in R_1 is plotted according to the wavelength scale currently in effect. A CALCOMP 565 plotter with 12-inch carriage and .005 inch step size interfaced with an XY-12 is required.

The y-axis is 5 inches long with 50 sample numbers per inch giving a full scale value of 250 data numbers. Data should be less than 400_{10} and greater than -100_{10}.

The x-axis is 8.5 inches long unless a PC command is in effect. If the scope display is in the expanded mode, only those points between the limits will be plotted. The x-axis calculation automatically scales the x increments so the entire length of the axis is used.

$$\text{Plotted } x = \frac{\lambda_{(point)} - \lambda_{min}}{\lambda_{max} - \lambda_{min}} * \text{axis length}$$

The x axis is linear according to wavelength. Tic marks are drawn every 50 data numbers on the y-axis and every 100 Angstroms on the x-axis.

If C_1 is non-zero, only C_1 points will be plotted using the same scaling calculations as would be used for the full number of points.

i.e., if the display is not in the expanded mode (sense switch 1=0), 512 points will be plotted unless C_1 is non-zero, then C_1 points will be plotted.

If the display is expanded, (HIEXPD - LOEXPD + 1) points are plotted unless C_1 is non-zero. C_1 should never be greater than (HIEXPD - LOEXPD + 1).

If a PS command is given, small squares are plotted for each point. A PL command will connect each point with a straight line.

Both PL and PS will drive the pen against the left margin, roll the drum down 8.5 inches and draw new axes in addition to drawing the graph. At the end of the plot the pen will be left at the origin.

The OP and OS commands are the same as the PL and PS commands except that only the graph is drawn. The pen is assumed to be positioned at the origin at the beginning of the command and will be left there.

This scheme produces plots on 8.5 x 11 inch paper with as many graphs on the same set of axes as desired. The PL or PS commands should be given first followed by an OP or OS command for each additional graph. The pen may be changed between graphs if it is not moved. Squares and lines may both be on the same set of axes.

CNTRL/R will halt any plotting operation in progress.

PP R_1 "TEXT" Print on Printer

Register R_1 will be printed on an LP-8 printer. Each line contains the x-value for the first data number followed by 10 data numbers. Any text after R_1 on the command line will appear as the page title.

PR R_1 C_1 Print on TTY

Register R_1 is printed on the teletype, C_1 data numbers per line preceeded by the x-value of the first data number. If C_1 is zero or omitted, 5 data numbers per line will be printed. C_1 must not be greater than 6.

PS R_1 C_1 Plot Squares

See PL command.

PV R_1 Print Value

The value of the point at the cursor in R_1 is printed on the TTY.

RD R_1 C_1 Read

Spectrum C_1 is read into Register R_1.

$0 \leq C_1 \leq 255_{10}$ for standard Linc Tapes

$0 \leq C_1 \leq 450_{10}$ for 896 block Linc Tapes.

The first block of the spectrum is C_1*2.

TW Type Wavelength

The wavelength of the cursor is typed and also put into floating point variable X.

T8 R_1 Transmit to PDP-8

The spectrum in R_1 is transmitted to the PDP-8. See the F8 command.

WR R_1 C_1 Write

Register R_1 is written on spectrum C_1, unit 1. See RD command.

PROGRAMMING COMMANDS

CA Clear All

All registers, variables and labels are zeroed.

CO text Comment

The text is listed on the TTY when this command is executed. A maximum of 30 characters are allowed.

CP Continue Program

This command continues running a program from the point when it was stopped or interrupted. CP is always manually entered.

CV Clear Variables

All the variables and labels are zeroed.

EP Enter Program

This command sets the program mode to "STORE" and initializes the program buffer. All commands entered after the EP and before a PE command will be stored in the buffer. Most of the commands will also be executed (see Table 2). If a mistake is made while typing commands and the carriage return is typed, then the whole program must be again entered. For this reason, it is better to write large programs with DIAL.

EQ	C_1 C_2 L_1	Jump if EQUAL

If $C_1 = C_2$, the next statement executed will be LABEL L_1. Otherwise, the next command will be executed.

FJL	F_1 F_2 L_1	Floating Point Jump - less than
FJP	F_1 L_1	Floating Point Jump - positive

For FJL, if F_1 is less than F_2, or for FJP, if F_1 is positive, label L_1 will be executed otherwise next statement.

FSA	V_1 F_1 F_2	Floating Point Set - Add
FSD	V_1 F_1 F_2	- Divide
FSI	V_1 C_1 C_2	- Integer
FSM	V_1 F_1 F_2	- Multiply
FSP	V_1	- Print
FSS	V_1 F_1	- Square Root

V_1 must be a floating point variable.

F_1 and F_2 are floating point constants or variables with or without a minus sign.

C_1 and C_2 are either integer constants or signed variables. The following actions take place:

$FSA \rightarrow V_1 = F_1 + F_2$

$FSD \rightarrow V_1 = F_1/F_2$

$FSI \rightarrow V_1 = C_1 + C_2$

$FSM \rightarrow V_1 = F_1 * F_2$

FSP → Value of V_1 is printed

$FSS \rightarrow V_1 = \sqrt{F_1}$

JP L_1 Jump Unconditionally

Label L_1 will be the next command executed.

LB L_1 Label

This command is a do-nothing command similar to a Fortran CONTINUE statement. Its purpose is to store in variable L_1 the address of the LB command in the program buffer. When a Jump L_1 command is executed, the contents of variable L_1 are put in the program pointer.

LE C_1 C_2 L_1 Jump if Less than or Equal

If $C_1 \leq C_2$, Jump to L_1, otherwise, do nothing.

LO FILENAME Load and Go

The DIAL File "FILENAME" is read into the program buffer, the labels are processed, the program pointer is initialized and the program mode is set to "RUN". If "FILENAME" is omitted or cannot be found, the source in the DIAL working area (Block 370_8 unit 0) is loaded. Note that programs can be chained with the LO command. Variables are not affected except those that are labels.

OI C_1 Output Interval

The OI command is similar to the "FOCAL" OI command. The program in execution pauses $C_1/10$ seconds, with the display active and then goes to the next statement. It is useful to put an OI command in the middle of a loop to enable the user to observe the results as they are being formed. The KW-12 is used.

PE Program End

The store mode is terminated (set by the last EP) and the manual mode is activated. This should be the last command of a program that is entered manually. Note that the PE is different than the SP command. SP halts a program that is in execution while PE has no effect when a program is running.

RP Run Program

RP starts the execution of the program buffer at the first command. Note the difference between RP and CP - RP always starts at the first line while CP starts wherever the program was stopped last.

SE V_1 C_1 C_2 Set

The integer variable V_1 is set to C_1 + C_2. Either C_1 or C_2 may be preceeded by a minus sign but must be separated by a space.

SI V_1 F_1 Set Integer to Floating Point

The integer variable V_1 is set to the value of F_1 according to the rules of floating point to integer conversion.

SP Stop Program

The program mode is changed from"RUN" to "MANUAL". Execution may be resumed with a "CP".

SS V_1 R_1 Spectrum Set

The value of $R_{1,\ cursor}$ is put into integer variable V_1. For instance, to get the value of point 200 in Register 4 into variable D, the following commands would be used:

SC 200

SS D 4

This command is the counterpart of the ED command.

SV FILENAME Save Program

The text in the program buffer is written onto the DIAL file "FILENAME". FILENAME must already exist and should be 1 or 2 blocks long, depending upon the length of the program being saved.

TABLE 1

SUMMARY OF RUFUS COMMANDS

COMMAND	DESCRIPTION	TYPE	OVERLAY
AD	Add	A	1
AV	Moving Average	A	1
BK	Background	A	1
CA	Clear All	P	1
CC	Correlation Coefficient	A	2
CD	Read Cards	I	2
CN	Condense Scale	A	3
CO	Text Comment	P	2
CP	Continue Program	P	1
CR	Clear Registers	A	1
CV	Clear Variables	P	1
CW	Change Wavelength Scale	I	3
DI	Return to DIAL System	I	1
DP	Dot Product	A	1
DS	Divide by a Scalar	A	1
DV	Divide by a Vector	A	1
ED	Edit	A	1
EP	Enter Program	P	1
EX	Expand Scale	A	3

COMMAND	DESCRIPTION	TYPE	OVERLAY
FCM	Franck-Condon Make	I	3
FCS	Franck-Condon Set	I	3
FD	Form Derivative	A	2
FF	Fourier Transform	A	4
FH	Form Histogram	A	3
FI	Form Integral Function	A	2
FJL	Floating Point Less Than Jump	P	1
FJP	Floating Point Positive Jump	P	1
FL	Form Line	A	2
FSA	Floating Set - Add	P	1
FSD	Floating Set - Divide	P	1
FSI	Floating Set - Integer	P	1
FSM	Floating Set - Multiply	P	1
FSP	Floating Set - Print	P	1
FSS	Floating Set Square Root	P	1
F8	Receive Spectrum from PDP-8	I	3
G	Get Spectrum	I	1
HP	High Pass Clipper	A	1
IN	Integrate	A	1
JP	Jump	P	1
LB	Label	P	1
LD	Lower Display	D	1

COMMAND	DESCRIPTION	TYPE	OVERLAY
LE	Jump if Less than or Equal	P	1
LF	Least Squares Fit	A	2
LI	List	P	1
LO	Load Program from DIAL (not a RUFUS file)	P	2
LP	Low Pass Clipper	A	1
LS	Left Circular Shift	A	1
MS	Multiply scalar	A	1
MT	Make Triangle	A	3
MV	Multiply Vector	A	1
NL	No List	P	1
NM	Normalize	A	1
NR	Noise Removal	A	1
O	Read Previous Spectrum	I	1
OI	Output Interval	P	1
OP	Overplot	I	4
OS	Overplot Squares	I	4
P	Read Next Spectrum	I	1
PC	Plotting Constant	I	4
PI	Print Integer	I	1
PL	Plot	I	4
PP	Print Spectrum (line printer, LP-8)	I	2
PR	Print Spectrum (TTY)	I	2
PS	Plot Squares	I	4

COMMAND	DESCRIPTION	TYPE	OVERLAY
PV	Print Value	I	1
PW	Power Spectrum	I	1
RC	Restore Cursor	D	1
RD	Read Spectrum	I	1
RL	Restore Limits	D	1
RP	Run Program	P	1
RS	Right Circular Shift	A	1
SC	Set Cursor	D	1
SE	Set Variable	P	1
SH	Set High Expand Limit	D	1
SI	Set Integer to Floating Point	P	1
SL	Set Low Expand Limit	D	1
SP	Stop Program	P	1
SS	Spectrum Set	P	1
SU	Subtract	A	1
SV	Save RUFUS Program on DIAL File	P	2
TW	Type Wavelength	I	3
T8	Transmit Spectrum to PDP-8	I	3
UD	Upper Display	D	1
WR	Write Spectrum	I	1
WS	Wavelength Set	A	3

COMMAND	DESCRIPTION	TYPE	OVERLAY
XR	Transfer	A	1
ZR	Zero Register	A	1

TYPES OF COMMANDS

A - Arithmetic

D - Display

I - Input/Output

P - Programming

TABLE 2

COMMAND ACTION IN DIFFERENT MODES

COMMAND	MANUAL MODE	STORE MODE	RUN MODE
CP	A	I	I
EP	A	A	I
EQ	I	I	A
JP	I	I	A
LB	I	A	I
LE	I	I	A
LO	A	I	A
RE	I	A	I
RP	A	A	A
SP	I	I	A

"A" means command works or is active

"I" means command has no effect or is inactive

All other commands are active in all three modes.

ENTERING RUFUS PROGRAMS WITH THE DIAL TEXT EDITOR

To use DIAL in writing RUFUS programs, type the program into the DIAL working storage area on unit 0. Use exactly the same format as is used when entering through RUFUS except that no EP or PE commands are required. Also a slash may be used to preceed a comment using the same rules as PAL. RUFUS will ignore the slash and the following text.

Also if a card reader is available, the programs may be read from cards into a DIAL source file with "PIP" and then loaded by RUFUS.

Error Messages

If RUFUS detects an error in a command, parameter or file, it will type two question marks and a signed decimal number and return to the manual mode. This decimal number is the address +1 of the location where the error was detected. Convert the number to octal and consult the listing if the cause of the error is in question.

Use of the Listing Switch

The Listing Switch is a software switch that is set with the LI or NL commands. If it is on, all commands will be printed as they are executed; the values of the variables will be printed before they are used by each command and all results will be printed. This feature is useful for debugging programs.

If the switch is off, only certain results will be printed.

The User's Monitor Command, "MCRUF"

The RUFUS system tape has a DIAL User's Monitor Command, "MCRUF", that writes the binary of the assembled overlays from the working storage area on unit 1 to the correct blocks of "KILLER--". This resides on block 270_8 of the system tape. After an overlay has been assembled, type the DIAL command "MCAn,0". Where n is the overlay number. This will load MCRUF and type "?n". If n is correct, type a CTRL/P. MCRUF will copy the binary and return to DIAL. If n is incorrect, type a space and MCRUF will only return to DIAL. "KILLER--" must be on unit 0.

DIRECTORY OF RUFUS SYSTEM TAPE

PX

NAME	SOURCE BN	SOURCE BLKS	BINARY BN	BINARY BLKS
KILLEH--			236	32
PIP			215	21
RUFUS			47Ø	16
SUPERSUM	213	2		
SLOWREAD	212	1		
DATASTRP	21Ø	2		
&CAMERON			2Ø6	2
&N-CHL			5Ø6	2
&G-CHL			2Ø4	2
&NOGAMMA			51Ø	2
&CO4POS			2Ø1	3
STARSRCH	512	1		
RAYLEIGH	513	2		
SHIFTY	177	2		
RAYPLOT	515	2		
&G-CHL9			175	2
&F-CHL9			517	2
&N21NEG			174	1
&N22POS			173	1
&N2LBH			521	4
&N2V-K			165	6
&CO1NEG			525	1
&LINEAR			526	2
CROSS	164	1		
OVR2	53Ø	32		
INTSRV	14Ø	24	133	5
OVR1	1ØØ	33		
MONST	562	14	576	5
PTABLE	6Ø3	5	61Ø	2
FLTPT			612	7
RUFINIT	73	5	621	3
MCRUF	7Ø	3		
OVR3	624	26		
OVR4	36	32		
NOISEOUT	652	1		

```
        EJECT
/
/
/       PTABLE (COMMAND TABLE)
/
/
/
/       PTABLE AS OF MAY 12, 1971
/
        PMODE
        *2400
        1               /OVERLAY 1
        -0104           /ADD
        -2325           /SU SUBTRACT
        -0213           /BK BACKGROUND
        -3222           /ZR ZERO
        -0504           /ED EDIT
        -2305           /SE SET
        -0520           /EP ENTER PROG
        -2220           /RP RUN PROG
        -2320           /SP STOP PROGRAM
        -1402           /LB LABLE
        -0521           /EQ  JP IF EQUAL
        -1405           /LE  JP IF LE
        -1220           / JP JUMP UNCOND
        -2303           /SC SET CURSON
        -2203           /RC RESTORE CURS
        -2270           /R8 READ 8BIT TA
        -2204           /RD READ LINC
        -2722           /WR WRITE LINC
        -1411           /LI LIST
        -1614           /NL NO LIST
        -2005           /PE END PROGRAM
        -2504           /UD UPPER DISP
        -1404           /LD LOWER DISP
        -0423           /DS SCALAR DIV
        -1523           /MS SCALAR MPY
        -0426           /DV VECTOR DIV
        -1526           /MV VECTOR MPY
        -2026           /PV PRINT VALUE
        -1116           /IN INTEGRATE
        -2310           /SH SET HI LIM
        -2314           /SL SET LO LIM
        -2214           /RL RESTORE LIM
        -1423           /LS LEFT SHIFT
        -2223           /RS RIGHT SHIFT
        -1420           /LP LOW PASS
        -1020           /HP HIGH PASS
        -0126           /AV MOVING AVG
        -1622           /NR NOISE REMOV
```

```
        -Ø322           /CR CLEAR REG
        -Ø326           /CV CLEAR VARIAB
        -Ø3Ø1           /CA CLEAR ALL
        -Ø411           /DI GET DIAL SYS
        -Ø32Ø           /CP CONTINUE PRO
        -3Ø22           /XR TRANSFER
        -2323           /SS SPECTRUM SET

        -17ØØ           /O GET LAST SP
        -2ØØØ           /P GET NEXT SP
        -Ø74Ø           /G GET N SP
        -1615           /NM NORMALIZE
        -Ø612           /FJ FLT PT JUMP
        -Ø623           /FS FLT PT SET
        -2311           /SI SET IN TO F
        -1711           /OI WAIT
        -Ø42Ø           /DP DOT PRODUCT
        -2Ø11           /PI  PRINT INTEGER
/
/
/
        2               /OVERLAY 2
        -2Ø22           /PR PRINT SPECTR
        -1417           /LO LOAD FILE
        -Ø3Ø3           /CC CORREL COEFF
        -Ø614           /FL FORM LINE
        -14Ø6           /LF LEAST SQUAR
        -2326           /SV SAVE PROGRAM
        -Ø317           /CO COMMENT
        -2Ø2Ø           /PP LIST ON PETN
        -Ø6Ø4           /FD FORM DERIVAT
        -Ø611           /FI FORM INTEGRA
/
/
/
        3               /OVERLAY 3
        -Ø67Ø           /F8 FROM PDP-8
        -247Ø           /I8 TO PDP-8
        -Ø61Ø           /FH FORM HISTOGR
        -Ø316           /CN CONDENSE SC
        -Ø53Ø           /EX EXPAND SCALE
        -1524           /MT MAKE TRIANGL
        -2723           /WS SET TO WVLGT
        -2427           /TW TYPE WVLGT
        -Ø6Ø3           /FC FRANCK-CONDO
        -Ø327           /CW CHNG WVLGTH
/
/
```

```
4
-2ØØ3        /PC PLOT CHRCTRS
-2Ø14        /PL PLOT SPECTRM
-172Ø        /OP OVER PLOT SP
-2Ø23        /PS PLOT SQUARES
-1723        /OS OVPL SQUARES
-Ø6Ø6        /FF FOURIER TRAN.
-2Ø27        /PW PWER SPECT
Ø
Ø
```

SOME EXAMPLES OF PROGRAMS

SLOWREAD

This program is a mini-MAGSPY that will show the spectra on tape 1, numbers A to B at a rate determined by C.

The display controls (POTS 0, 1, 4, 5 and sense switches 0,1, 2, 3) may be changed while the program is running.

```
CO        Set A to First Spectrum
CO        Set B to Last Spectrum
CO        Set C to Delay
SP            / Stop and allow user to set A, B and C manually
              and then type CP to resume execution.
SE E A        / Save value of A
LB M          /Jump here for Main Loop
SC E          /Set the cursor x to spectrum number to show
              which spectrum is being displayed.
RD Ø E        /Read into RO from spectrum number E.
OI C          /Wait C/10 seconds
SE E E 1      /Increment E by 1
LE E B M      /If E≤ B,  do it again
SP            /Done so stop
SE E A        /User can type CP to see the whole thing again
JP M          /Go back to Main Loop
```

STARSRCH

STARSRCH will read spectra from numbers A to B, display the data, integrate each spectrum between X values J and K and store the average data value in Register 1. This gives a band strength versus time graph of all the data.

```
CO          Set A to start
CO          Set B to stop
CO          Set J to low limit
CO          Set K to high limit
SP
ZR 1            /Clear Register 1
SL J            /Set the limits
SH K
SE L K -J       /L = number of points in intergration
FSI U L         /Float L into floating point variable U.
SE E Ø          /E is the point counter
LB M            /Main Loop
RD O A          /Read spectrum A into RO
IN O            /Integrate puts sum into x
FSD X X U       /Divide by number of points
SI D X          /Convert to integer
SC E            /Set cursor for graph
ED 1 D          /Edit average value into R 1
```

```
SE E E 1        /Increment E
SE A A 1        /Increment A
LE A B M        /If A≤B, do it again
SL Ø            /Done so set the limits so graph occupies entire
                 screen
SH E
CO          To plot Type PL 1
SP              /Stop
```

NOISEOUT

NOISEOUT removes the noise from the spectrum in Register 0 using the second derivative technique. If the absolute value of the second derivative is greater than "C", the point in RO will be replaced by the average of its two neighbors. More than one pass may be necessary.

```
CO        Set C to Limit
SP
LB M           /Start of pass
FD Ø 1         /First derivative in R1
FD 1 2         /Second derivative in R2
SE A Ø         /Index on A
LB N           /Search Loop
SC A           /Set cursor on point
SS B 2         /Get value of second derivative in B
LE B -C O      /Second derivative is too small
LE C B O       /Second derivative is too large
LB P           /Comes here after removing noise
SE A A 1       /Increment A
LE A 511 N     /Go back to N if A ≤ 511
CO        Type CP for another pass
SP
JP M           /Go do another pass
LB O           /Come here to remove noise
```

```
UD Ø          /Show Register 0

OI 1

NR Ø 1        /Remove noise (the "1" may be made larger)

OI 1          /Show without noise

JP P          /Go back to loop
```

If it is desired to set the noisy points to zero, the following sequence of commands will do the job much faster than the above sequence.

```
FD 0 1         /First derivative

FD 1 2         /Second derivative

LP 2 C 1000    /Edit all points >C to 1000

HP 2 -C 1000   /Edit all points <-C to 1000

HP 2 999 0     /Change all good points to 0

BK 2 -1000     /Register 2 is all 0's for bad points or -1000
                for good points

MV 0 2 -1000   /Mask out good points
```

The chief advantage of doing it this way is that all of the commands work on the whole spectrum instead of only one point at a time.

RAYLEIGH

This is a more complex program that was designed to do a signal strength and error estimate for the 1216 Å Lyman α hydrogen signal from the Mariner Mars 1969 and 1971 project. The variable "C" is the center of the signal to be analyzed which may actually be anywhere in the spectrum. The instrument is assumed to have a triangular window with a half-width of 4.5 sample numbers. The calibration function of the instrument should be in Register 5 in the form of Response *100. The program first finds the background centered on the sample number B, subtracts it, calibrates the spectrum and then does a least squares fit of the signal using a triangle. After obtaining a triangle fit to the data, the statistical error is calculated and printed. The least square fit uses the formula

$$\alpha = \frac{\Sigma t_i d_i}{\Sigma t_i^2} \quad \begin{cases} t_i = \text{triangle} \\ d_i = \text{data} \end{cases}$$

where α is the correction factor for the triangle.

Also the error $e_i = d_i - \alpha t_i$

the fractional error is:

$$\beta = \sqrt{\frac{\Sigma e_i^2}{(n-1)\ \Sigma(\alpha t_i)^2}}\ .$$

and the error in Rayleighs is

$$= \beta\Sigma\alpha t_i$$

while the true signal is:

$$\Sigma\alpha t_i.$$

```
CO          Set B to BKGRND CENTER

CO          Set C to SIGNAL CENTER

CO          Set F to POINT SPREAD

CO          Read Calibration into 5.

CO          Read Spectrum into 0.

SP

LB M            /For restarting

SE D B -F       /Lower background limit

SL D

SE D B F        /Upper background limit

SH D

IN 0            /Get sum of background in X

SE G F F        /Get number of points (2*F + 1)

SE G G 1

FSI R G         /Float into R

FSD X X R       /Average background

FSA X X .5      /Round up

SI A X          /Convert to integer

CO              /Space 1 line

CO BACKGROUND
```

```
PI A            /Print background
BK 0 -A         /Subtract it
MV 0 5 100.     /Calibrate it
SE D C -F
SL D            /Lower limit for signal
SE D C F
SH D            /Upper limit for signal
SC C            /Set cursor on signal
ZR 1            /Clear R1
MT 1 1000. 4.5    /Make any triangle, 4.5 half width
DP 0 1          /Σt_i d_i
FSA R X         /Save it in R
DP 1 1          /Σt_i^2
FSD U R X       /α
FSM R U 1000.   /α*1000(area of new triangle)
ZR 1            /Clear R1
MT 1 R 4.5      /Best fit triangle
SU 0 1 2        /Errors into R2
DP 2 2          /Σ error^2
FSA U X         /Save it
DP 1 1          /Σt_i^2
FSD X U X       /Σerror^2/Σt_i^2
FSI U F F       /(n-1) degrees of freedom
FSD X X U       /Divide by degrees
FSS X X         /Take square root
```

```
CO
CO EMISSION RATE IN RAYLEIGHS
FSP R         /Print R
FSM R X R     /Error fraction *signal
CO STANDARD ERROR IN RAYLEIGHS
FSP R
CO
CO FRACTIONAL STANDARD ERROR
FSP X
CO
CO
CO
SP
JP M          /Restart for another spectrum
```

SHIFTY

"SHIFTY" reads in a spectrum, subtracts the background and shifts it left or right until it correlates best with a triangle produced at sample number C. Then the shifted spectrum is written on top of the old spectrum. This is used to get all of the wavelengths aligned. The triangle is placed so it coincides with a known signal (Lyman α).

"SHIFTY" is not listed in this document although it is on the RUFUS system tape.

SUPERSUM

SUPERSUM obtains an average spectrum for spectra A through B. All data points higher than C are considered noise and are set to 0. It is assumed that the maximum data value is 255_{10} so that 8 spectra may be added before overflow is possible. The technique is to add 8 spectra to Register 1, divide it by 8, add this to Register 2 and accumulate 8 sums of 8 divided by 8 in Register 3. When the last spectrum has been processed, the program corrects for spectra that thave not been added to Register 3 and then displays the result.

"SUPERSUM" is not listed here but may be found on the RUFUS system tape.

DOING THINGS WITH FOURIER TRANSFORMS

Let the Fourier Transfrom of the vector "x" be $S_x(f)$.

Then the power spectrum of x is $S_x(f) \cdot S_x(f)^*$.

where $S_x(f)^*$ is the complex conjugate of $S_x(f)$.

Autocorrelation is $S^{-1}\left(S_x(f) \cdot S_x(f)^*\right)$ where S^{-1} is the inverse transform.

The cross power spectrum of X and Y is:

$S_y(f) \cdot S_x(f)^*$.

The cross-correlation of X and Y is:

$S^{-1}\left(S_y(f) . S_x(f)^*\right)$.

And the transfer function (which is complex) is:

$H(f) = S_y(f)/S_x(f)$.

To find the power spectrum of R_1:

(R_2 must be 0.)

FF R_1 R_2 -1

PW R_1 R_2 R_3 /The power spectrum will be in R_3

To find the auto-correlation of R_1

(R_2 must be zero)

FF R_1 R_2 -1

PW R_1 R_2 R_3

MV R_3 R_3 X /see note below

ZR R_2

FF R_3 R_2 1 /auto-correlation is in R_3

The floating point constant 'X' should be equal to the largest value in R_3 squared divided by 2000_{10} to prevent overflow or underflow in the MV instruction.

To find the cross-power spectrum of R_1 and R_3 (R_2 and R_4 are zero)

$= S_y(f) \cdot S_x(f)^*$

FF R_1 R_2 -1

FF R_3 R_4 -1

XR R_1 R_5

MV R_5 R_3 X

XR R_2 R_6

MV R_6 R_4 X

AD R_5 R_6 R_5

MV R_1 R_4 X

MV R_2 R_3 X

SU R_1 R_2 R_6

XR R_6 R_4

XR R_5 R_3

PW R_4 R_3 R_1 /Cross-power spectrum in R_1

The value of 'X' is the same as that given for finding the auto-correlation.

To find the cross-correlation of R_1 and R_3, follow the procedure given in the cross-power spectrum derivation and put a

FF R_5 R_6 1 command

at the end. The cross-correlation will be in Register R_5.

Here is a RUFUS program that finds the transfer function of data in Registers 0 and 2:

$H(F) = S_y(F)/S_x(F)$

SQRT(-1) = i

$S_x(F) = xR + i * xI$

$S_y(F) = yR + i * yI$

$H(F) = (yR + i * yI)/(xR + i * xI)$

$= (yR + i * yI) * (xR - i * xI)/(xR↑2 + xI↑2)$

$= ((xR * yR + xI * yI + (xR * yI - xI * yR) * i)/$

$(xR↑2 + xI↑2)$

The program gives the real part in Register 4 and the imaginary part in 5. The real part looks better.

```
FF Ø 1 -1            /S(x)
FF 2 3 -1            /S(y)
SE A Ø               /Set CNTR
LB C                 / STRT of Loop
SC A                 /Set Cursor
SS B Ø
FSI O B              /xR
```

```
SS B 1
FSI R B         /xI
SS B 2
FSI U B         /yR
SS B 3
FSI X B         /yI
FSM L O O       /xR * xR
FSM F U U       /yR * yR
FSA F F L       /xR 2 + yR 2
FSM I U O       /xR * yR
FSM L X R       /xI * yI
FSA I I L       /xR * yR + xI * yI
FSD I I F       /(xR * yR + xI * yI)/(xR↑2 + yR↑2)
FSM I I 1ØØ.    /Scale it
ED 4 B          /Edit in Real Part
OI 1            /Look at it
FSM L O X       /xR * yI
FSM I U R       /xI * YR
FSA L L -I      /xR * yI - xI * yR
FSD L L F       /(xR * yI - xI * yR)/(xR 2 + yR 2)
FSM L L 1ØØ.
SI B L          /Scale it
ED 5 B
OI 1            /Look at it
SE A A 1        /INCR CNTR
```