

NATIONAL RADIO ASTRONOMY OBSERVATORY
GREEN BANK, WEST VIRGINIA

ELECTRONICS DIVISION TECHNICAL NOTE NO. 145

Title: **FARANT 5.0 ON THE HP-320 AND HP-9816 COMPUTERS**

Author(s): John Granlund

Date: May 5, 1988

DISTRIBUTION:

GB

GB Library
R. Norrod
R. Weimer
D. Schiebel
E. Childers
R. Lacasse
C. Brockway
J. Coe
W. Albing
S. White
G. Behrens
R. Fisher
F. Crews
B. Peery

CV

ER Library
IR Library
M. Balister
S. Weinreb
C. Burgess
S-K. Pan
A. R. Kerr
Nancyjane Bailey
L. D'Addario
N. Horner
S. Srikanth
J. Granlund

TU

Library Downtown
Library Mountain
J. Payne
R. Freund
J. Lamb

VLA

VLA Library
P. Napier
J. Campbell
W. Brundage

FARANT 5.0 ON THE HP 320 AND HP 9816 COMPUTERS

John Granlund

A new computer, the Hewlett Packard model 320, has been installed at Ivy Road to run FARANT. This is faster than the HP 9816 computer that has been used with FARANT for many years [1-3]. It is planned to retain the older HP 9816, keeping its operating system, its version of FARANT, and its peripheral hardware compatible with those of the new computer. Thus, a user will be able to start developing his version of FARAFT on one computer, to finish on the other, and to run his program on either machine, without being required to modify the program depending on which computer he is using. This goal has been entirely met: The BASIC 5.0 system discs for the two machines are identical, but the versions of compiled FARANT on the two FARANT 5.0 discs are different.

The new HP 320 computer was ordered with the latest BASIC 5.0 operating system, with the HP 9122 dual disc drive for double-sided discs, and with the HP 2225A "ThinkJet" printer. Accordingly, these items have also been purchased and associated with the HP 9816 computer. Excepting a few BASIC 2.0 commands that the user will probably not have installed in his earlier versions of FARAFT, BASIC 5.0 supports all of the commands of the earlier BASICs. The user's earlier versions of FARAFT should therefore run without modification on either computer.

A major addition to BASIC is that BASIC 5.0 has been revised to deal with complex numbers and arrays of complex numbers. FARANT has not been modified to use the BASIC 5.0 complex number arithmetic, but the required COMPLEX binary program is being loaded by the AUTOSTart program, and users are invited to try it in their versions of FARAFT. The operating system binaries have been expanded to accommodate the extension to complex number arithmetic and other BASIC 5.0 additions; the compiler binaries have been correspondingly expanded. For these reasons, the first attempt to load all binaries, FARANT, FET_LIB, and a version of FARAFT, followed by a COMPILE command, produced a memory overflow error. Accordingly, the HP 98257A 1 MByte RAMs in both computers were replaced by Infotek AM220 2 MByte RAMs. Because of its built-in 1/4 MByte RAM, this leaves the HP 9816 with 2-1/4 MBytes of memory; the HP 320 now has 2 MBytes.

1. Changes Since BASIC 3.0/4.0

1.1 The Infotek Compiler

An Infotek compiler was purchased for the HP 9816 when Hewlett Packard introduced BASIC 4.0, and Infotek compilers for BASIC 5.0 have been procured for both the HP 320 and the HP 9816. To deter unscrupulous users from selling copies of its compiler, Infotek now delivers a "security module" with the compiler, which must be plugged into a computer's Hewlett Packard Interface Bus if the compiler is to be used with that computer. It is possible to use a certain security module on either the HP 320 or the

HP 9816; security modules are identical and are not associated with either the compiler binaries that were shipped with them or the computer systems that they operate in.

With its compiler for BASIC 5.0, Infotek included a copy of its floating point matrix binary, FPMAT, which is faster than Hewlett Packard's MAT binary, but uses considerably more code. FPMAT is being loaded by the AUTOSTart program in place of MAT. The sparse matrix option of FPMAT -- operations involving zero-valued matrix elements are completed without arithmetic -- is being left OFF. The options that are being selected by AUTOSTart are

- Use the fastest floating point processor on line.
- Use the faster, 80-bit extended precision, mode of calculation.

Uncompiled or compiled versions of FARANT may be selected by the user to be loaded by the CONFIG program from the FARANT 5.0 disc. Since both versions are on the same disc, different file names are required. FARANT_U and FARANT_C have been chosen, respectively. The versions of FARANT_C on the FARANT 5.0 discs for the two computers are different because the HP 320 version has been compiled to use the Motorola MC68881 floating point coprocessor resident in that machine, and the HP 9816 version uses the Infotek FP210 coprocessor installed there.

FARANT_C has been compiled with SOURCE OFF, which uses less disc storage space and computer memory. But with the original interpretive BASIC program discarded, FARANT_C cannot be UNCOMPILED, and keyboard commands pertaining to most of the original line numbers will not be executed properly. The compiler directive OPTIMIZE ON, which selects nine space-saving shortcuts described in the compiler manual, has also been used. This has cut the amount of compiled code almost in half and perhaps also halved the run time. But it can no longer be guaranteed that the uncompiled and compiled versions of a program will produce the same results, even though they are expected to. Users are asked report any such discrepancies to John Granlund.

The FET library, FEB_LIB, on the FARANT 5.0 disc has not been compiled, and neither have the various versions of FARAFT, also stored there. This leaves the user free to compile FET_LIB and his version of FARAFT with whatever compiler directives he may choose; compile time is too short to be troublesome. The recommended directives are OPTIMIZE ON and the default SOURCE ON. The latter choice makes it possible for the user to edit his program, COMPILE again, and re-run. It is also recommended that the user STORE his version of FARAFT only after UNCOMPILEing it; this will take much less disc storage capacity.

1.2 The HP 9122 Double-Sided Disc Drive

HP 9122D dual disc drives for double-sided 3-1/2-inch discs are now associated with both the HP 320 and the HP 9816 computers. The new gray double-sided discs, INITIALIZED with the HP 9122 drive, hold 2464

256-byte records, whereas the blue single-sided discs, INITIALIZED with the original HP 9121 drive, held only 1056 256-byte records. But the difference in storage capacity is almost entirely a difference between the two disc drives: A gray or a blue disc can be INITIALIZED with the new HP 9122 drive to hold 2464 records, but programs stored on such a disc cannot be retrieved by the original HP 9121 drive, even if they occupy only the first 1056 records on the disc; a blue or a gray disc can be INITIALIZED by either disc drive to hold 1056 records, and so that programs stored on the disc can be recovered by either disc drive. To format either disc with the new HP 9122 drive so that programs stored on it can be retrieved by the original HP 9121 drive, execute

```
INITIALIZE ":CS80,700,0 or 1",2,4
```

This will limit the disc capacity to 1056 records and make the disc readable by the original HP 9121 drive.

The only difference between the blue single-sided discs and the gray double-sided discs is that the blue discs are lubricated on only one side to relieve pickup head wear, whereas the gray discs are lubricated on both sides. Either disc may be used occasionally in the new HP 9122 drive, but it is recommended that a gray disc be used for programs in development or for data needed each time a program is run. To equalize pickup head wear, it is also recommended that a different default disc drive be selected each time BASIC is booted.

1.3 The HP 2225A "ThinkJet" Printer

HP 2225A "ThinkJet" Printers are now used with both the HP 320 and the HP 9816 computers. These produce an 80-column hard copy on 8-1/2 x 11 inch fanfold paper with 15/16-inch margins at both edges, so cutting the printer back to 75 columns and starting with column 6 to provide a left-edge margin is no longer appropriate, as it was for the HP 2673A thermal printer. In fact, there is no way to modify the ThinkJet's default, or power-on, operating options. The two default options that are being changed by the AUTOSTart program after power-on are

- Normal print density
- Perforation Skip OFF.

With the second default option in effect, the printer would produce 66 lines of print per 8-1/2 x 11 inch sheet of fanfold paper, possibly printing one line on the perforations between adjacent sheets.

Normal print density produces light gray characters that are just not dark enough. Hewlett Packard offers a special paper for the printer on which normal print density is darker, but still only marginally acceptable. Unfortunately, this special paper acts like a blotter and cannot be written on with a fountain pen. Hewlett Packard also offers a print cartridge containing ink suitable for common printer papers, but again normal print density is only marginally acceptable with this ink. Finally, the ThinkJet

offers the option of Bold Print, also called "Quality Mode Printing". This is dark enough, but certainly not too dark, and bold printing seems to take no more time than normal density printing. It has been chosen to use common printer paper, print cartridges containing the corresponding suitable ink, and Bold Print.

Even with the Bold Print option in effect, DUMP GRAPHICS produces a graph on the ThinkJet that is too light. Accordingly, SUB Dump_graphics has been added to FARANT, and this produces acceptable double-density graphs. But even though it runs print-limited, it takes about four times as much time as DUMP GRAPHICS. A user whose program produces many graphs may wish to consider using DUMP GRAPHICS for his many preliminary graphs and reserving SUB Dump_graphics for his few final graphs.

The AUTOSTart program turns Bold Print ON and Perforation Skip ON. With perforation skip enabled, the printer produces 60 lines of print on an eleven-inch page, and printer and paper can be aligned so that three of the lines skipped are at the top and three are at the bottom of the page. The printer determines where it is on the page by counting the turns of its platen. At power-on, the count is initialized to correspond to the first line at the top of the page. So to insure that perforation skip skips the perforations, rather than six lines in the middle of each page, the print head should be at the top of the page -- that is, ready to print on the fourth line below the perforations -- when the printer is turned on. This position is most easily reached by pressing the form feed (FF) button on the printer just before turning power off, and users are requested to extend this courtesy to their fellow users. The platen count can also be corrected after power-on by pressing the line feed (LF) button to bring the print head to the top of the page and then pressing the blue button marked "Aa", which initializes the platen count.

Paper will often jam in the printer when it is used with the transparent paper shield pressed down on the output paper roll after a previous output has been torn off. This can be avoided by loosening the shield until several lines of output have been printed. After a paper jam, a number of corrections are required. Printing will have ceased with the yellow attention light under the blue button flashing, the print buffer will be loaded with print instructions, and the default operating options will have been returned to the printer. The user should

- Press CLR IO -- to stop the input of print instructions.
- Clear the paper path.
- Press the blue button -- to reset the printer. This could result in the printing of many lines with the default options in effect, but it will clear the print buffer.
- Execute PRINT -- to accommodate a possible missing (carriage return)/(line feed) in the last print instruction in the print buffer.
- Press LF several times until the print head is at the top of page.

- Press the blue button -- to initialize the platen count.
- Execute PRINT CHR\$(27)&"(s1B"; -- to turn Bold Print ON.
- Execute PRINT CHR\$(27)&"&l1L"; -- to turn Perforation Skip ON.

With an 80-column printer in use, all lines of FARANT now produce an artistically acceptable and readable listing. Certain lines of FARANT have been revised to use all 80 columns. Since the ThinkJet printer does not respond to ASCII character #255, inverse video K, used, e.g., in the command to clear the CRT screen, CHR\$(255) has been substituted for it in FARANT.

It has long been clear that typing-aid key labels should not be allowed to clutter the CRT at the point where the user is presented with his graph, ready for labeling. So SCRATCH KEY was used to remove the key labels. But since the user could have removed the disc containing the FARKEY labels from the default disc drive, it did not make sense to LOAD KEY "FARKEY" after DUMP GRAPHICS. The user was then left without typing-aid keys. Instead, FARANT has been revised to use, and the user is advised also to incorporate in his version of FARAFT, the equivalents of the commands KEY LABELS OFF and KEY LABELS ON, which do not require inputs from the disc drive. These two commands are supported by the binary file CRTX, which is not needed for any other purpose, and is therefore not loaded. Instead, the following equivalent commands should be used:

Command	Equivalent Command
KEY LABELS ON	CONTROL 1,12;0
KEY LABELS OFF	CONTROL 1,12;1

At this time, SUB Smith in FARANT turns KEY LABELS OFF, SUB Mplot in FET_LIB turns KEY LABELS OFF and ON, and SUB Web in FET_LIB turns KEY LABELS ON. Among the versions of FARAFT, SLICE1 turns KEY LABELS ON and 673A_FIT and LENS_GROOV turn KEY LABELS OFF and ON. The user should note that, after the command KEY LABELS OFF has been given, his efforts to recall the typing-aid key labels by executing LOAD KEY "FARKEY" will not succeed. Instead, he, or his program, must execute CONTROL 1,12;0.

2. SUB Equalrip(P(*),Tmin,Tmax,INTEGER N1,N2)

This SUBprogram has been added to FARANT. Its purpose is to adjust the parameters contained in the vector P(*) of the non-negative objective function $y(t;P]$ so that the maxima of y in the range $Tmin \leq t \leq Tmax$ of the independent variable are made equal. In words, this is an equal-ripple program. What is really needed is a minimax program: one that would adjust the parameters to minimize the maximum value of $y(t;P]$ within the specified range of t . Fortunately, equal-ripple peak values often agree with minimax peak values; however, sometimes the minimax peak values are smaller. But at the time of writing, a suitable minimax program had not been found. So Equalrip must be regarded as a temporary substitute for a minimax program.

For this reason, only a bare minimum description will be given here. Users who need more detail -- or who are aware of a suitable minimax program! -- should see John Granlund.

Like SUB Optimize, SUB Equalrip is CALLED with initial guesses for the parameter values installed in P(*), and with values for Tmin, Tmax, and INTEGERS N1 and N2. It calls SUB Cktnalysis -- with Opt = 1, Fvalue = t, and with current parameters installed in P(*) -- for each evaluation of the objective function, which SUB Cktnalysis must install in Fvalue without changing P(*). When it is done, SUB Equalrip PRINTs -- on the printer in service when it was CALLED -- the final mean peak height and the "MEASURE" of peak height inequality, the initial parameter values, and the final parameter values. Then it CALLs SUB Cktnalysis with the final parameter values in P(*), Fvalue = final mean peak height, and Opt = 0. Before it quits, SUB Equalrip also sets Tmin = MEASURE and Tmax = final mean peak height, so these and the final parameters are also available to the CALLing context. If max = maximum peak height and min = minimum peak height,

$$\text{MEASURE} = \frac{\text{max} - \text{min}}{\text{max} + \text{min}} .$$

SUB Equalrip uses MEASURE to decide when to stop iterating. Only if the number of peaks found is greater than the number of parameters will SUB Equalrip quit after MEASURE has been driven to a minimum value, has risen, fallen, and risen again in the next several iterations, and has started back down once more. This deters the program from quitting until MEASURE is so small that it is primarily determined by roundoff errors.

Unlike SUB Optimize, SUB Equalrip evaluates all of the required derivatives numerically, and it does not use second derivatives, obtained by updating previous second derivative estimates, as SUB Optimize does. SUB Equalrip uses a complex, but rather efficient, two-step procedure to locate and evaluate the peaks of the objective function. In the first step, the objective function is sampled coarsely at N1 samples per peak spacing: If there are N parameters, NN1 + 1 samples uniformly spaced throughout the specified range of t are taken. Peaks are located approximately by considering all triples of adjacent sample values, selecting those for which

$$y_{k-1} < y_k \quad \text{and} \quad y_k \geq y_{k+1} ,$$

fitting parabolas to the selected point triples, and recording the abscissas of the peaks of these parabolas. The left edge sample is considered to be a peak if $y_0 \geq y_1$, and the right edge sample y_{NN1} is considered to be a peak if $y_{NN1} > y_{NN1-1}$.

It will be noted from the previous paragraph that peaks of the objective function, considered as a function of the independent variable t, are expected to be more-or-less equally-spaced. The user must select t so that this condition is met. For example, peaks of the squared error of a multi-decade 90° phase shifter are more-or-less uniformly distributed in log f

over its useful range of frequencies, so $t = \log f$ is an appropriate choice for the 90° phase shifter. Again, loss peaks of a Chebyshev-like bandpass filter, constructed of reasonable, unequal lengths of transmission line, are more-or-less uniformly distributed in $t = \cos^{-1} \frac{2(f - f_0)}{w}$, where f_0 = center frequency and w = bandwidth, so such a choice is appropriate for this problem. Note that t becomes complex out of band for this choice! It should also be noted that loss minimization is not enough in this problem; constraints must be included to force the filter to cut off outside of $f_0 - w/2 < f < f_0 + w/2$. In summary, the choices of the independent variable t and of the objective function are left to the user.

In the second peak-location step, the objective function is selectively sampled more finely at $N2$ samples per peak spacing. If all samples were taken, it would take $NN2 + 1$ samples to cover the specified range of t , and with $N2 \gg N1$, this would take much time. But in this second step, the only samples taken are at the abscissas of the peaks internal to the specified range of t found in step one, and one sample either side of each such peak location. A parabola is fitted to the point triple corresponding to each internal peak, the peak location of this parabola is recorded, and the peak height is obtained by a CALL to SUB Cktnalysis.

It is expected that INTEGER $N2$ will be determined by the user so that final peak location errors due to roundoff and due to non-zero sample spacing are comparable. INTEGER $N1$ should be chosen so that the central sample of each triple used in final peak location is the largest. Equalrip objects with a BEEP whenever this condition is not met, and the user should then increase $N1$. In the 90° phase shifter problem, $N2$ was calculated to be 1600, and $N1$ was estimated at 20, but an excess of BEEPs occurred when two pairs of peaks began to merge, so $N1$ was increased to 40.

If more than $N + 1$ peaks are found, the lesser peaks are discarded until there are just $N + 1$. $N + 1$ linearized equations in the N parameters and the final common peak value are solved to make all peak values equal. If only $M < N + 1$ peaks are found, the final common peak value

$$y_{00} = \frac{M}{N + 1} \bar{y} = \frac{1}{N + 1} \sum_{i=1}^M y_i$$

is chosen, where y_i are the heights of the peaks that have been found. The parameters are adjusted in the gradient direction -- to minimize the sum of squares of their fractional changes -- to reach this goal. So if only a few peaks have been found, large changes will be called for, but if most of the expected $N + 1$ peaks have been found, only small changes will be required. If the required change in any parameter would have it increase or decrease by more than 2:1, Equalrip emits a low-pitched BEEP and reduces the fractional changes of all parameters so that none changes by more than 2:1.

In closing, the user is to be reminded once again that Equalrip will not quit until it has found a number of peaks at least one greater than the

number of parameters. Using a short piece of open-circuited transmission line as a coupling capacitor and specifying both its line length and characteristic impedance as parameters is one very likely way to prevent Equalrip from finishing.

REFERENCES

- [1] Dan L. Fenstermacher, "A Computer-Aided Analysis Routine Including Optimization for Microwave Circuits and Their Noise," NRAO Electronics Division Internal Report No. 217, July 1981.
- [2] John Granlund, "FARANT on the HP9816 Computer," NRAO Electronics Division Internal Report No. 250, July 1984.
- [3] John Granlund, "FARANT 3.0 on the HP9816 Computer," NRAO Electronics Division Technical Note No. 124, November 1984.