OSU: GROPE User's Manual

(Preliminary Version)

A self-learning package
for OSU Computer Graphics System
using a time-sharing terminal

ccm-71-05

Computer Center
Physics Department
Oregon State University
Corvallis, Oregon 97331

January, 1972

ACKNOWLEDGMENT

TABLE OF CONTENTS

†These commands may also be used in Display Phase.

INTRODUCTION

This manual describes the operation of an interactive computer display system which may be used to present graphic information in the classroom. The system consists of a set of computer programs which are run on-line using the time-shared OSU CDC-3300 computer and a Tektronix 4002A Computer Graphics Terminal (tekterminal). The latter consists of a keyboard for instructor input and a cathode ray tube for computer output. This enables an instructor or student to issue commands via the keyboard which result in alphanumeric or graphic information being displayed on the cathode ray tube. An unlimited number of commands can be issued so that displays may be continually developed, augmented, and modified in a classroom presentation. Since normally the computer responds to commands in seconds, the system provides a powerful tool to supplement classroom lectures. In addition to providing dynamic classroom displays, the system is appropriate for producing other visual aids such as 35 mm slides.

The system, which has been named GROPE (Graphical Representation of Parameterized Expressions), has as its underlying instructional philosophy the enhancement of the students' appreciation of various parameters which enter into a problem's formulation. Thus GROPE has been structured to develop and display parameterized families of curves. These curves may have analytic functional forms or they may involve numerically obtained solutions to differential equations.

The capability to utilize computer generated solutions to differential equations greatly expands the variety and complexity of interesting phenomena which can be studied. This ability to study complex problems is illustrated in the next section by the example of projectile motion under the influence of realistic frictional forces.

In the vast majority of cases, GROPE can be utilized effect-
ively by a prospective teacher-user who has no previous computer
experience or knowledge. Defining analytic functions and estab-
lishing differential equation sets to be solved is accomplished
by using a notation which is similar to ordinary mathematical
notation. The timing and sequential structure of graphic presen-
tations is controlled by the instructor via a set of mnemonic
commands. Provisions also exist for efficiently recovering from
the inevitable input errors.

In order to provide the flexibility required for classroom
presentation, the number of GROPE commands is necessarily large.
However, the complete contents of this manual need not be digested
before meaningful displays are generated. The reader should first
read Section I which provides enough basic information so that
he can begin to acquire a "feel" for the system. Then after a
little practice he should be able to effectively incorporate
GROPE into his lectures. If the user is not familiar with the
OSU operating system or the tekterminal, then in addition he must
read Appendix A which describes the mechanical steps necessary to
communicate with the computer via the tekterminal. After reading
Section I and Appendix A, the user should begin "playing" with
the system, for this is by far the quickest way to acquire the
needed skills.

Section II contains examples which are intended to demon-
strate some of the flexibility available in GROPE. The remainder
of the manual provides a more detailed description of each aspect
of the system and may be assimilated at a more leisurely pace
or as the need arises.

SECTION I

AN OVERVIEW OF GROPE

I.1   Four Phases of GROPE - Structure:

The GROPE system is divided into four basic phases:

A)   Function Definition and Specification (FD&S)
B)   Range and Axis Specification (R&AS)
C)   Display Phase (DP)
D)   Parameter Specification (PS)

To utilize GROPE efficiently it is necessary to understand the purpose of each phase, the relationships between phases, and how to transfer from one phase to another. A schematic representation of GROPE is shown in Figure 1. The solid lines indicate the natural flow from one phase to another and this is achieved by depressing the carriage return button (CR)[†] at the end of each phase. The natural flow of GROPE is the same as the procedure followed when producing graphs by hand. That is, the first phase, FD&S, allows functions to be defined, parameter values to be assigned, and specification of the types of plots to be made. The second phase, R&AS, requests information concerning the extremities of the axis and the location of the axis origin. When the pre-plotting information is complete, the Display Phase is entered, axes drawn, plots labelled, and the specified functions plotted.

After the display is complete, other graphs may be drawn, other functions may be plotted on the same axis set, or any number of options may be selected. GROPE provides the opportunity for the instructor to select easily and quickly the phase he wishes to execute.

[†]Parenthesis are used to enclose special non-printing keyboard buttons such as carriage return and space bar.
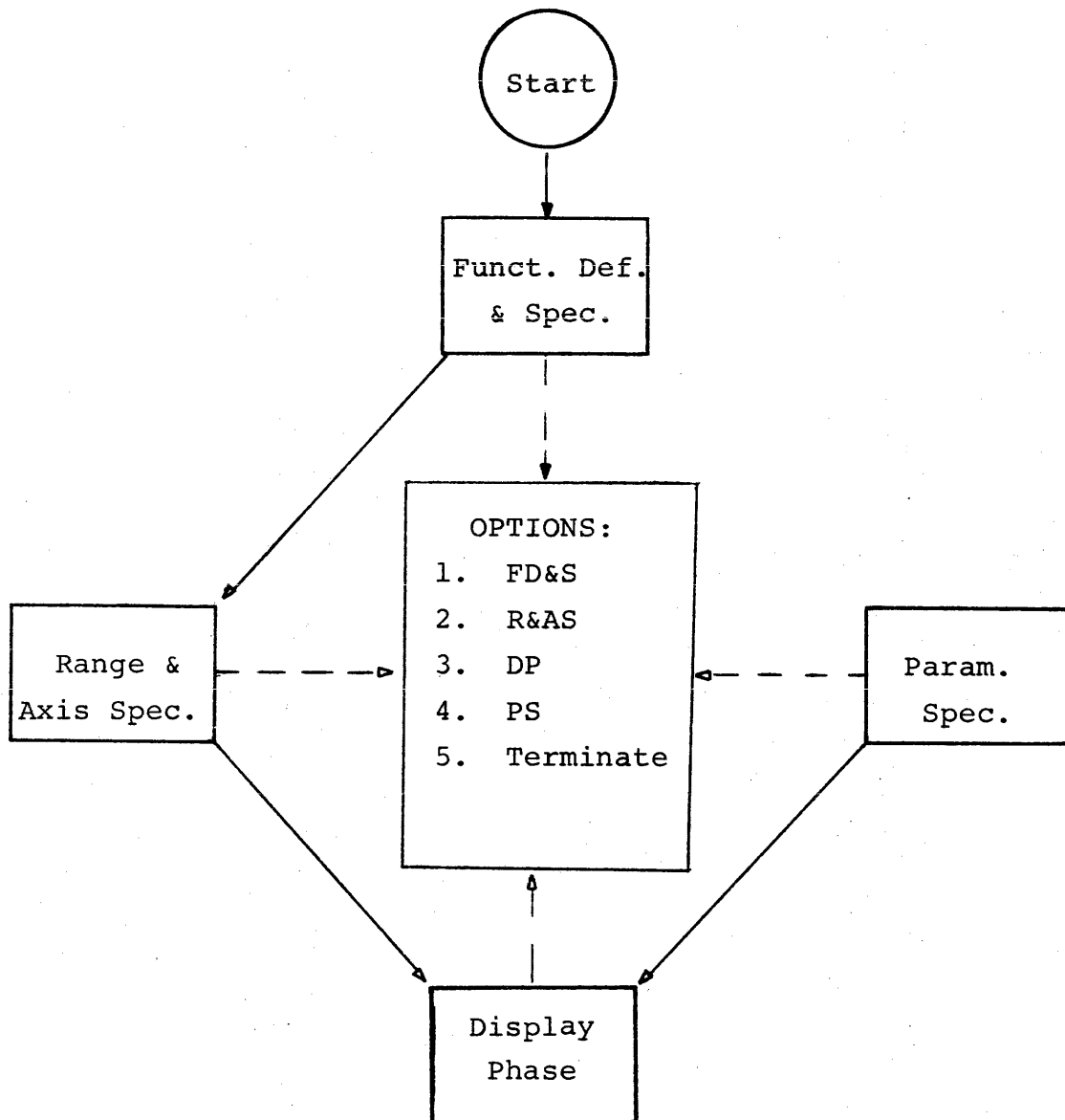
Figure 1

————————————	Exit via (CR) [Natural flow]
— — — — — — —	Exit via unmodified QUIT
Q,F	To Function Definition & Specification Phase
Q,R	To Range & Axis Specification Phase
Q,D	To Display Phase
Q,P	To Parameter Specification Phase
Q,T	Terminate Program Execution

OVERVIEW

To transfer from one phase of GROPE to another in other than the "natural flow" the QUIT command is used. This command instructs the computer to exit from the present phase and proceed to the designated phase. For example, QUIT,PS(CR) would transfer control to the Parameter Specification phase. All commands[†] and their modifiers may be abbreviated by their first letters, for example typing Q,P(CR) is equivalent to QUIT,PS(CR). Notice also that a comma (or blank) must be inserted between a command name and it's modifier.[††] If the instructor is unable to remember which options are available he may simply type an unmodified QUIT(CR) and the options which are available will be listed.

Program execution should be terminated by typing QUIT,T(CR) which is the last option shown in Figure 1. This type of orderly termination allows the GROPE staff to keep statistics on the use of the system.

Editing and error correction is achieved by using the back-slash, \ , or the @ symbol. The @ symbol deletes a whole line and the backslash typed n times deletes the last n characters. Error correction features are discussed in detail in Appendix C.

[†]Two exceptions are the manual interrupt and the ON,OFF. (See pp. 34 and 71.)

[††]There is only one exception to the comma being used to separate a command and its modifier and this is discussed on p. 48.

I.2   Function Definition and Specification Phase

After GROPE has been loaded into the computer by performing the steps outlined in Appendix A, the system is in the FD&S phase. Immediately the following message will be displayed followed by the printing of a > and the ringing of a bell. In all cases the ringing of a bell indicates input is expected.

```
+--------------------------------------------------------------+
|                                                              |
|                                                              |
|         Function Definition And Specification (FD&S)         |
|                                                              |
|                                                              |
|                                                              |
|     For information type I.   Terminate FD&S with CR or a    |
|     QUIT command.                                            |
|                                                              |
|                                                              |
|                                                              |
|     >                                                        |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
+--------------------------------------------------------------+
```
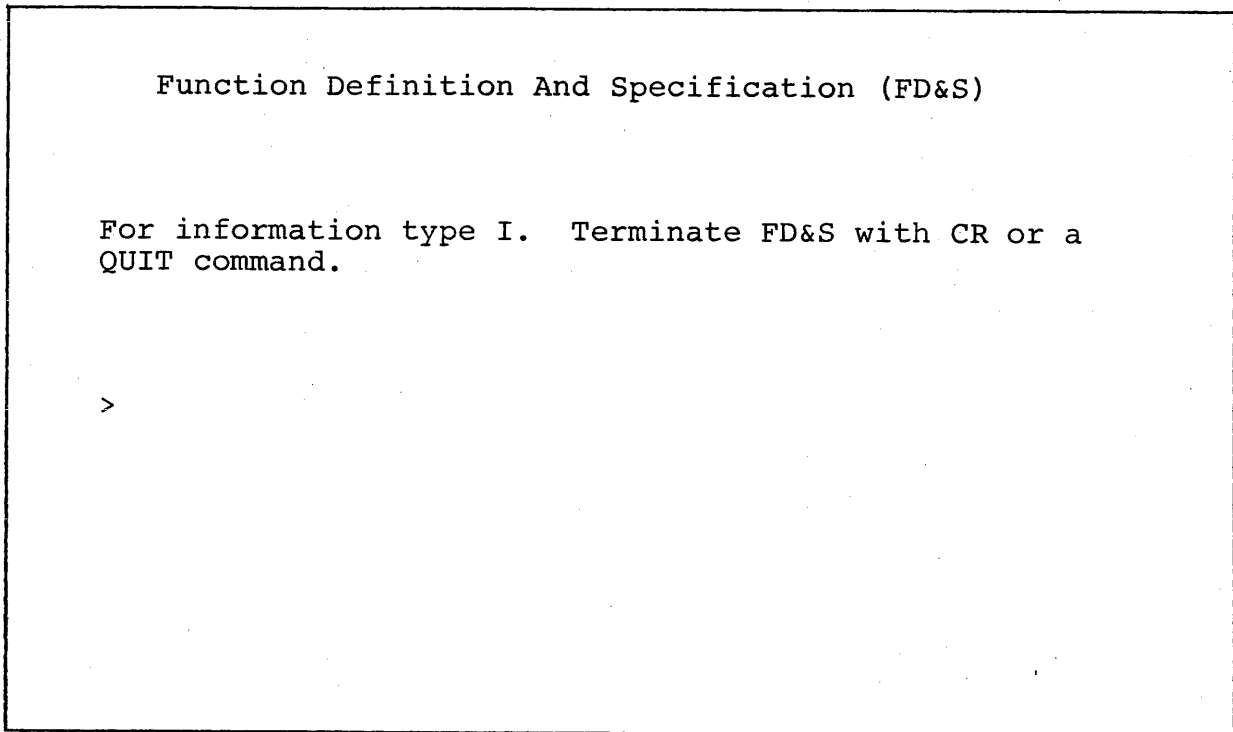
Figure 2.   In this manual information contained in rectangles
            is a duplication of the tekterminal screen.

The appearance of the "greater than" symbol (>) and/or the ringing of the bell will always indicate that GROPE is waiting for input information.[†] In this instance it is waiting for a function definition.

[†]One exception to this is when the right-hand margin of the tekterminal screen is exceeded. In this case the bell also rings in much the same way as an ordinary typewriter.

OVERVIEW

   Functions may be defined by typing in the analytic form of
the function or by typing in a differential equation set. For
example, suppose one wishes to study the vertical displacement of
a projectile with a frictional force which is proportional to dif-
ferent powers of the velocity. The appropriate set or coupled
differential equations is

$$\ddot{x} = -B\dot{x}(\sqrt{\dot{x}^2 + \dot{y}^2})^N$$

$$\ddot{y} = -G - B\dot{y}(\sqrt{\dot{x}^2 + \dot{y}^2})^N$$

The steps necessary to define the functions x(t) and y(t), assign
appropriate parameter values[†], and specify the functions to be
plotted are shown in Figure 3.

[†]GROPE treats parameter values and initial conditions the same
except in the calculational role. Therefore, the word "parameter"
unless otherwise stated will also mean initial conditions through-
out this manual.

Function Definition And Specification (FD&S)

For information type I. Terminate FD&S with CR or a
QUIT command.

>X"(T) = -B*X'*SQRT(X'^2 + Y'^2)^N   (CR)
Y"(T) = -G - B*Y'*SQRT(X'^2 + Y'^2)^N   (CR)
(CR)

X = 0   (CR)
X' = 30   (CR)
Y = 0   (CR)
Y' = 30   (CR)
B = .2   (CR)
N = 0 BY   .2   (CR)
G = 9.8   (CR)

>PLOT,Y(T)   (CR)
>(CR)

Figure 3.   In this manual information typed by the instructor
will be underlined; information typed by the computer
will not be underlined. The one-fingered typist
should not worry about typing all this information
in class since this information can be stored and re-
called in class by giving a single command. (See
p. 31.)

When entering a member of a differential equation set, the
procedure is to isolate the highest occurring derivative on the
left, and set this equal to the rest of the equation. The indepen-
dent variable T must be explicity entered and enclosed in paren-
thesis on the left side of the equal sign. Although in this example
we are only interested in plotting one function, Y(T), the other
displacement, X(T), is also completely defined by the differential
equation set as are the first and second derivatives. Therefore,
we could also plot these functions if desired.

OVERVIEW

To define analytic functions, the function name and its arguments enclosed in parenthesis are placed on the left side of the equal sign with the defining expression on the right. For example, to define a quadratic function, F(X), one would type

$$F(X) = A + B*X + C*X\wedge 2$$

followed by a (CR). It is possible to define new functions in terms of previously defined functions, their derivatives or even their integrals.

The format shown in Figure 3 and also for F(X) is closely related to the mathematical expression one would normally write. However, there are some departures and also there are certain simple rules which must be followed. These rules are discussed in detail in Section VII. For the present we list in Table 1 the special symbols and their corresponding mathematical operations.

Symbol	Operation	Example
$\wedge$	Exponentiation	$A\wedge 3 = A^3$
/	Division	$A/B = A \div B$
*	Multiplication	$A*B = A \times B$
−	Subtraction	$A - B$
+	Addition	$A + B$

Table 1

It should be noted that the square root function, SQRT, is one example of the many internally defined functions. A complete list of these functions is given in Section VII.4.

In general, every line of typed input must be terminated by depressing the carriage return. When the first differential equation is typed and terminated with a (CR), the cursor returns

to the left margin and the bell rings. The bell ringing is the
signal to enter the second differential equation. To terminate
the differential equation set, an extra (CR) must be given whether
the set has one or more equations.

Once the set has been entered, the computer scans the formulas
and begins to ask for the initial conditions and parameter values.
The instructor responds to each request by typing a number followed
by (CR). An important instructional feature illustrated by this
example is the capability to make some parameters automatically
varying. In this example (see Figure 3) the parameter N will start
at 0 and will be incremented automatically by certain commands in
steps of .2 in the Display phase. It is possible to simultaneously
increment up to 10 parameters.

When the parameter specification is complete the computer
again requests information. Although a number of options are
available, in this example the PLOT command was given which instructs
the computer to plot the function Y(T) in the Display phase. It is
possible to plot more than one function on the same display screen
and this capability is discussed on pages 36 and 38. Section III
gives a complete list of the FD&S commands and their meanings.

The final (CR) of Figure 3 terminates the FD&S phase and
control is transferred to the Range and Axis Specification phase.

OVERVIEW

I.3   Range and Axis Specification Phase (R&AS)

In the R&AS phase GROPE requests information about the ranges of the independent or dependent variables. These requests and the corresponding responses might appear as in Figure 4.

```
┌─────────────────────────────────────────────────────────────┐
│                                                             │
│         Range and Axis Specification Phase                  │
│                                                             │
│                                                             │
│    Answer the question "ORIGIN OF--- AXIS=" by typing LOW,  │
│    MIDDLE, HIGH, or a numerical value.                      │
│                                                             │
│                                                             │
│            INITIAL VALUE OF T = 0   (CR)                    │
│            FINAL VALUE OF T = 5.6   (CR)                    │
│            ORIGIN OF T-AXIS = LOW   (CR)                    │
│            NUMBER OF POINTS = 100   (CR)                    │
│                                                             │
│            MINIMUM VALUE OF Y = 0   (CR)                    │
│            MAXIMUM VALUE OF Y = 50   (CR)                   │
│            ORIGIN OF Y-AXIS = 0    (CR)                     │
│    CR  WILL TAKE YOU TO DISPLAY PHASE                       │
│    >                                                        │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

Figure 4

In specifying the number of points some restraint should be exercised for economic reasons. One should attempt to use only as many points as is necessary to produce a "pleasing" curve. The maximum number of points which may be specified is 1000, but usually 100 points will give good curves.

Provisions are in GROPE for "automatic" scaling of the dependent variable so that the last two questions may be avoided if so desired. By typing AUTO(CR) in response to any of the questions concerning the ranges of Y, automatic scaling is implemented. These provisions may be of great convenience and are discussed in more detail in

Section IV, page 45. If the automatic scaling is not used, then some homework regarding the appropriate values for Y will save time; however, it is often as easy to proceed by trial and error until a good graphic display is achieved.

In fixing the origin of either axis a numerical value may be specified, or many times it is more convenient to specify whether it is to be at the low, middle, or high position on the axis. Figure 5 shows the four possibilities when the T-origin is chosen to be L, M, H, or 1.5 and Y-origin is 0.
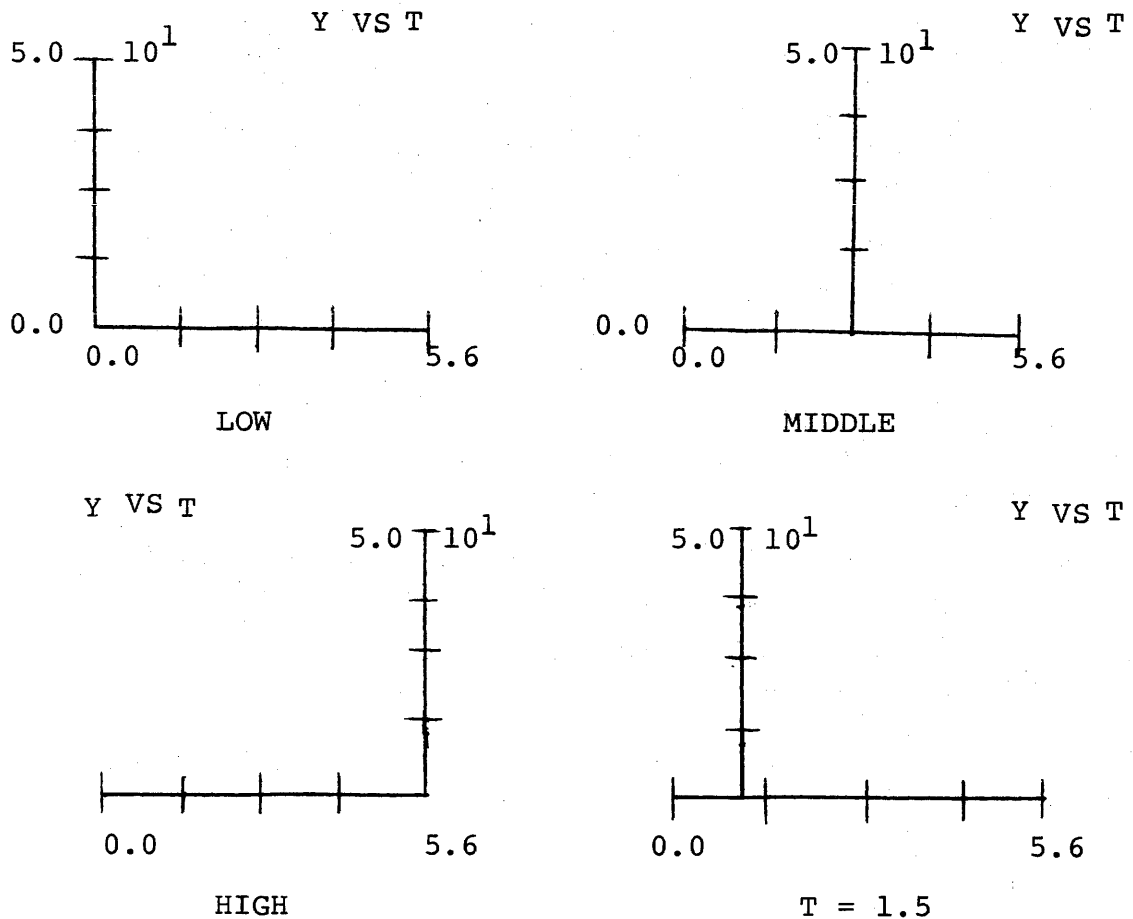

Figure 5

Pressing the (CR) button as instructed at the end of the R&AS phase transfers control to the Display phase.

I.4   Display Phase (DP)

Immediately on entering the Display phase the message shown in Figure 6 is printed.

```
DISPLAY PHASE


                    Enter Display Command:


>
```
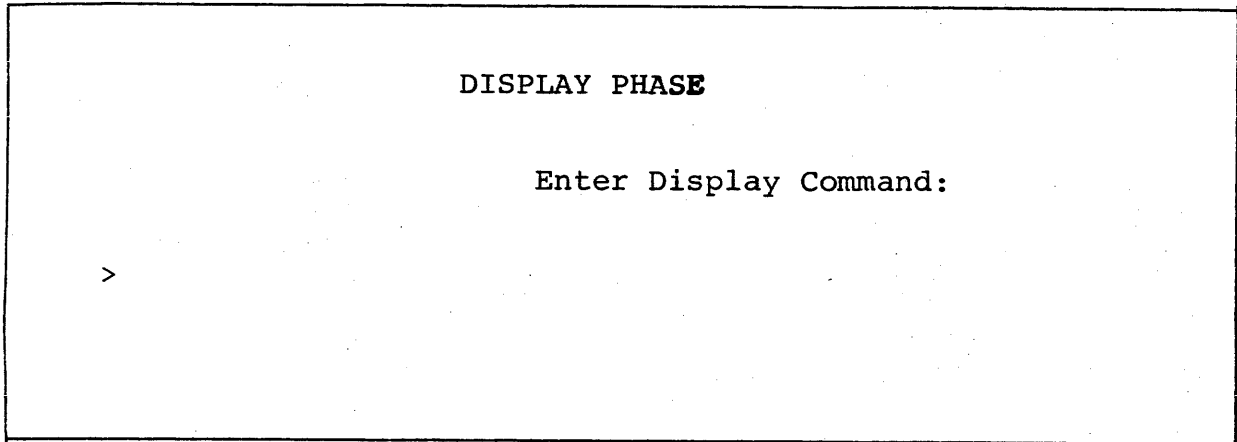
Figure 6

As in FD&S the appearance of the > symbol and the ringing of the bell is the signal that GROPE is waiting for input. A complete list of Display commands and their purpose is given in Section V.

Most likely one would like a plot of the function Y(T) using the initial parameter values as specified in FD&S. This may be accomplished by issuing the CLEAR command which has the general form CLEAR,r,m where r is _any_ real number and m is a positive integer. The CLEAR performs the following operations:

1) The display screen is erased.
2) The varying parameter values are incremented r times.
3) The axis and labels are drawn.
4) The functions specified by the PLOT command in FD&S are plotted using the incremented parameter values.
5) m members of each family of curves are plotted. Successive members of each family have parameter values which have been incremented $r, (r + 1), (r + 2),..., (r + m-1)$ times.

OVERVIEW

If there are more than one set of axes to be drawn, steps
3 through 5 are repeated. When all curves have been drawn, a >
will be typed in the upper left corner and the bell rung; signaling
that the next command is expected.

If the modifiers, r and m, are omitted then the values 0 and
1, respectively, are assumed, so that typing CLEAR(CR) and
CLEAR,0,1(CR) are entirely equivalent. In the present example either
of these forms would result in the curve and axis shown in Figure 7.
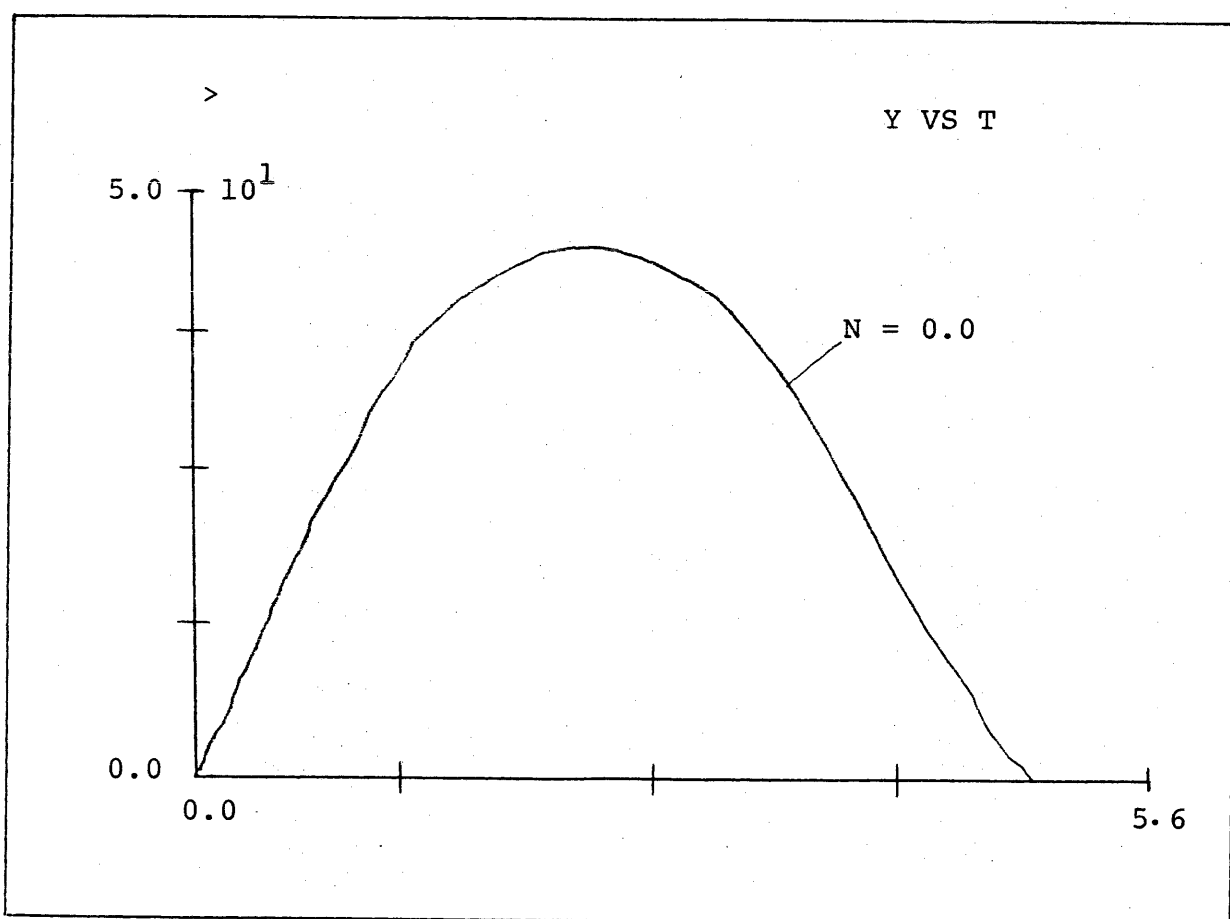

Figure 7. The curve labelling was accomplished using the (DEL)
          command.   (See p. 48.)

As another example suppose that the varying parameters are to
be incremented five times and the next three curves of the family are
to be generated. Typing CLEAR,1,2(CR) would achieve this and the

results are shown in Figure 8. Notice that there has been a screen
erasure between Figures 7 and 8, so that the curve with N=0 is no
longer present. Since the erasure may or may not be pedagogically
useful, an alternate command, called JUMP, can be employed.


```
>CLEAR,1,2(CR)
```

5.0 10¹   Y VS T

0.0

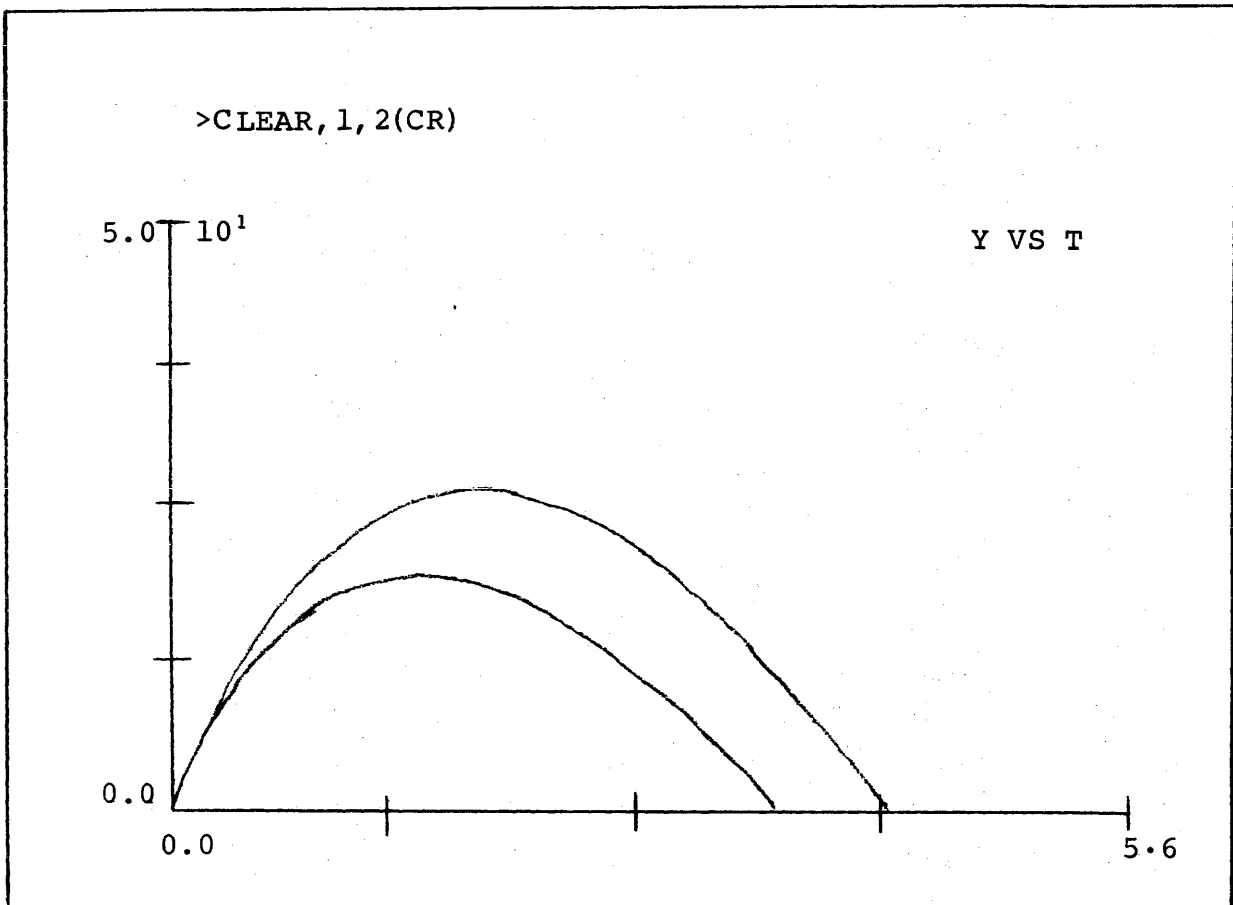0.0   5.6

Figure 8. This set of curves was generated by a CLEAR,1,2, using
the functions of the previous example.

The general form of the JUMP command is JUMP,r,m. Except for the
fact that JUMP does <u>not</u> erase the screen it is identical to the
CLEAR command, so that in the above example JUMP,1,2(CR) would
simply result in a superposition of the displays shown in Figures
7 and 8.

Although the CLEAR and JUMP provide great flexibility in curve
generation, they are not generally the most frequently used commands.
In the most frequent case, one wishes r to be a positive integer, and
more often than not r is 1. The Carriage Return command provides an

easy way to increment varying parameters an integral number of times
and generate the corresponding curves. Its general form is (n Space
Bars) m(CR) where (n + 1) is the number of times the varying para-
meters are to be incremented and m is the number of curves to be
generated. If the Space Bar is not depressed then the varying para-
meters are incremented once, and if m is not typed its assumed
value is 1. A few examples of the (CR) command and its relationship
to the JUMP command are shown below:

JUMP,1,1   Equivalent to   (CR)
JUMP,2,3   Equivalent to   (1 space bar) 3   (CR)

Table 2 contains a partial list of additional Display commands
and their purpose. This list is intended at this point to indicate
some of the graphics manipulations which are possible in the Display
phase. A complete discussion of each of these may be found on the
page listed in the right column.

COMMAND NAME	PURPOSE	PAGE REFERENCE
(DEL)	Labeling curves	48
DISPLAY	Displays function definitions	31
FILE	Files a display for future use	31
ON,OFF	Initiates special effects such as dashed curves	34
PARAMETER	Lists current parameters	35
RESET	Resets varying parameters to initial values	38
Λ	Changes parameters and/or increments	41

Table 2

When the classroom discussion of a given display is complete,
the instructor may wish to quit the display phase and proceed to
other phases to perform additional tasks. As discussed earlier in
Section I.1, this is accomplished by use of the QUIT command, for
example control can be transferred to the FD&S phase by typing
QUIT,FDS(CR). When returning to FD&S all previously defined functions
and the current parameter values are still retained, but the plotting

specifications are deleted. Once in FD&S it is possible to define
new functions and/or change the functions which are to be plotted.
Any new parameters which are introduced will be assigned values as
before.

If QUIT,PS(CR) is typed instead then control is transferred
to the parameter specification phase. In this phase all of the
current parameter values and increments are listed, and a conven-
ient format exists for altering any or all of these values and
increments. This is a useful option if there are many values to be
changed. Otherwise one should use the UP-ARROW command, "∧", which
is discussed on page 41.

QUIT,RAS(CR) will return execution to the Range and Axis Speci-
fication phase where the ranges of the dependent and/or independent
variable may be altered.

The instructor may also wish to quit and go home in which
case he would type QUIT,T(CR). The computer responds by typing a
pound sign "#" which indicates the system is in the control mode
(see Appendix A). The instructor responds by typing LOGOFF(CR),
and the computer types

        TIME XX.XX SEC, MFBLKS X COST $S.SS

IMPORTANT: If the computer fails to respond in this way,
            you have not logged off and the user's job
            number will continue to accrue charges.
            (See Appendix A.)

COMMENT: At this point you should attempt to generate some displays
of your own. The only additional information you need is contained
in Appendix A which describes the mechanical steps necessary to load
GROPE into the computer.

SECTION II

EXAMPLES

1. Projectile motion without friction (Figure 9).
    Illustrates:

    1.   Definition of analytic function.

    2.   Varying parameters.

    3.   Parametric plots (see p. 36).

    4.   CLEAR and (CR) commands (see p. 48).

    5.   Simple curve labeling (see p. 48).

    6.   Display of function definition (see p. 31).

FD&S PHASE

```
>X(T) = V0*COS(TH)*T    (CR)
       V0 = 18   (CR)
       TH = .34 BY .16   (CR)
>Y(T) = V0*SIN(TH)*T - .5*G*T^2    (CR)
       G = 9.8
>PLOT,Y(X)    (CR)
> (CR)
```

R&AS PHASE

```
INITIAL VALUE OF T = 0   (CR)
FINAL VALUE OF T = 3.2    (CR)
NUMBER OF POINTS = 101    (CR)

DO YOU WANT THE HORIZONTAL AND VERTICAL SCALE
FACTORS TO BE THE SAME?    YES

MINIMUM VALUE OF Y = 0   (CR)
MAXIMUM VALUE OF Y = 12    (CR)
ORIGIN OF Y-AXIS = 0    (CR)

MINIMUM VALUE OF X = 0    (CR)
MAXIMUM VALUE OF X = 40    (CR)
ORIGIN OF X-AXIS = 0    (CR)
```
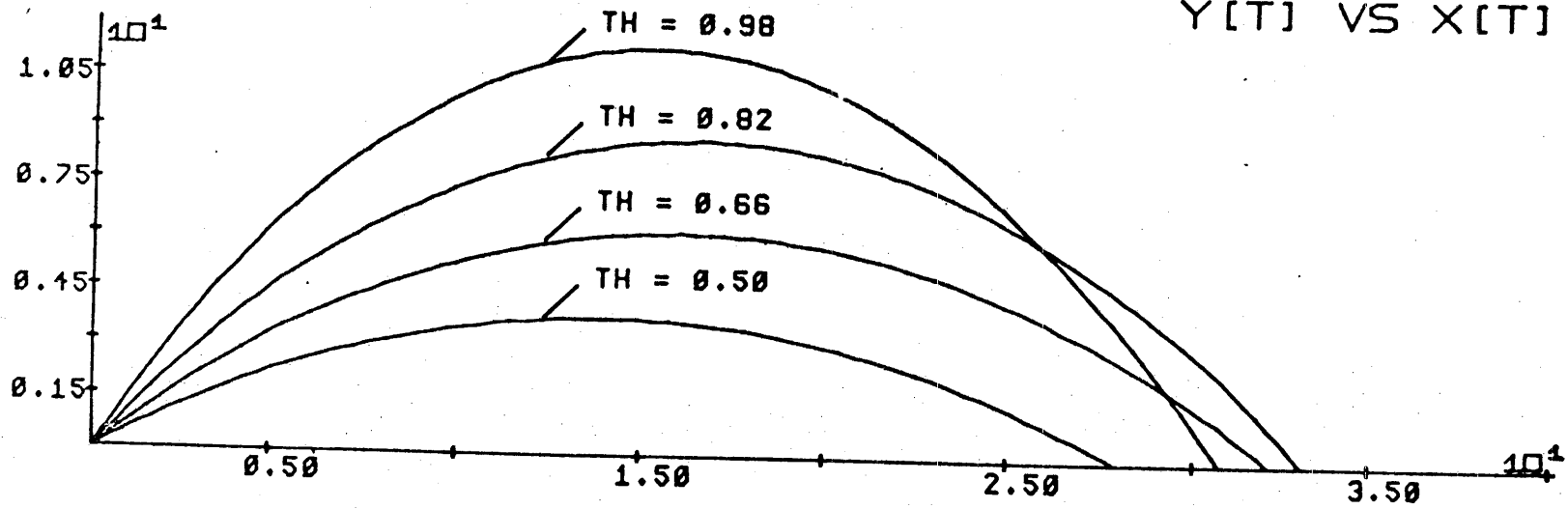
DISPLAY PHASE

```
>CLEAR,1,1    (CR)
>(DEL) (position crosshairs) (SPACE BAR)
>(CR)
>(DEL) (position crosshairs) (SPACE BAR)
>(CR)
>(DEL) (position crosshairs) (SPACE BAR)
>(CR)
>(DEL) (position crosshairs) (SPACE BAR)
>DISPLAY,X,Y    (CR)
```

Figure 9

>DISPLAY.X.Y
>

$$X(T) = VO*COS(TH)*T$$
$$Y(T) = VO*SIN(TH)*T - .5*G*T^2$$

Y[T] VS X[T]

2. Projectile motion without friction (Figure 10).
    Illustrates:

> 1. Derivative of analytic function (see p. 69).
> 2. Horizontal SPLIT plot (see p. 38).
> 3. Multiple curves on same axis set (see p. 36).
> 4. Use of ON and OFF command to selectively dash curves (see p. 34).
> 5. Use of DEL to display a message (see p. 48).

<div align="center">

FD&S PHASE
</div>

```
(X(T) and Y(T) are defined in example 1.)
>VY(T) = Y'    (CR)
>VX(T) = X'    (CR)
>PLOT,Y,VY    (CR)
>SPLIT,HORIZONTAL    (CR)
>PLOT,X,VX    (CR)
> (CR)
```

<div align="center">

R&AS PHASE
</div>

```
    INITIAL VALUE OF T = 0   (CR)
    FINAL VALUE OF T = 3.2    (CR)
    ORIGIN OF T-AXIS = 0   (CR)
    NUMBER OF POINTS = 100   (CR)

    MINIMUM VALUE OF Y = 0   (CR)
    MAXIMUM VALUE OF Y = 12   (CR)
    ORIGIN OF Y-AXIS   = 0   (CR)

    MINIMUM VALUE OF X = 0   (CR)
    MAXIMUM VALUE OF X = 40   (CR)
    ORIGIN OF X-AXIS   = 0   (CR)
```

<div align="center">

DISPLAY PHASE
</div>

```
>CLEAR, 2,2 (CR)
>(DEL)(pos. crosshairs)(DEL)(pos. crosshairs)(DEL)M  Y(T)   (CR)
>(DEL)(pos. crosshairs)(DEL)(pos. crosshairs)(DEL)M  X(T)   (CR)
>(DEL)(pos. crosshairs)(DEL)(pos. crosshairs)(DEL)M  VY(T)   (CR)
>(DEL)(pos. crosshairs)(DEL)(pos. crosshairs)(DEL)M  VX(T)   (CR)
```

Y VS T

X VS T

Figure 10

21.
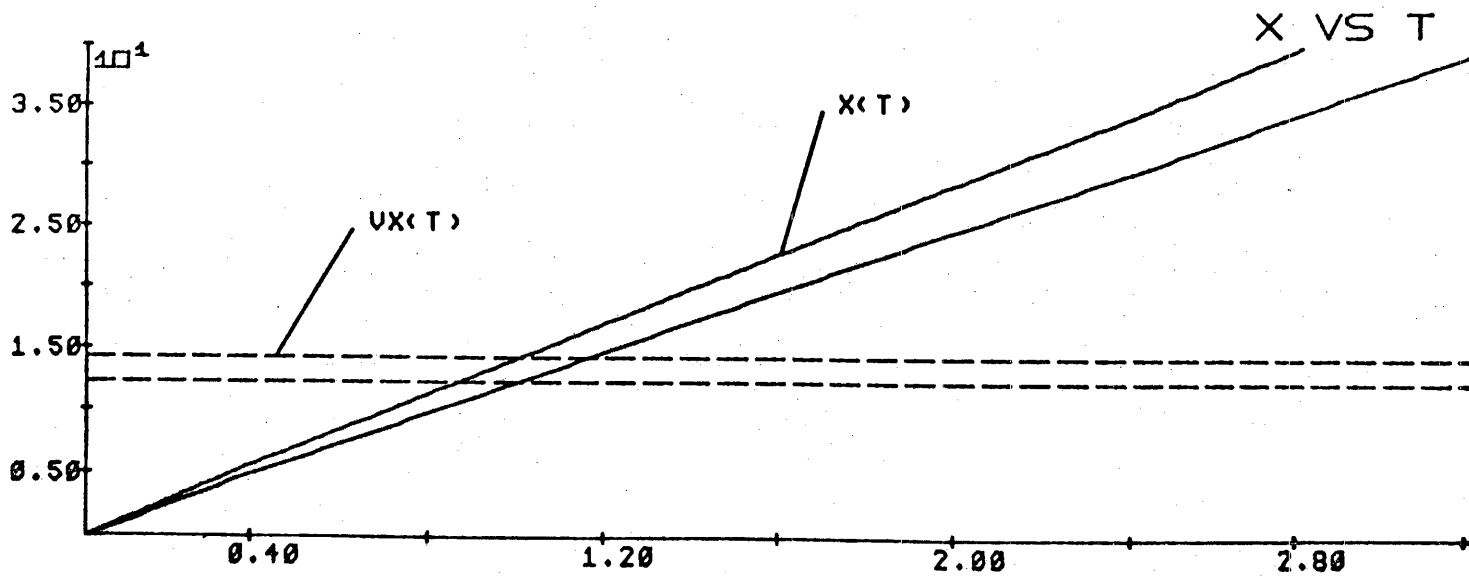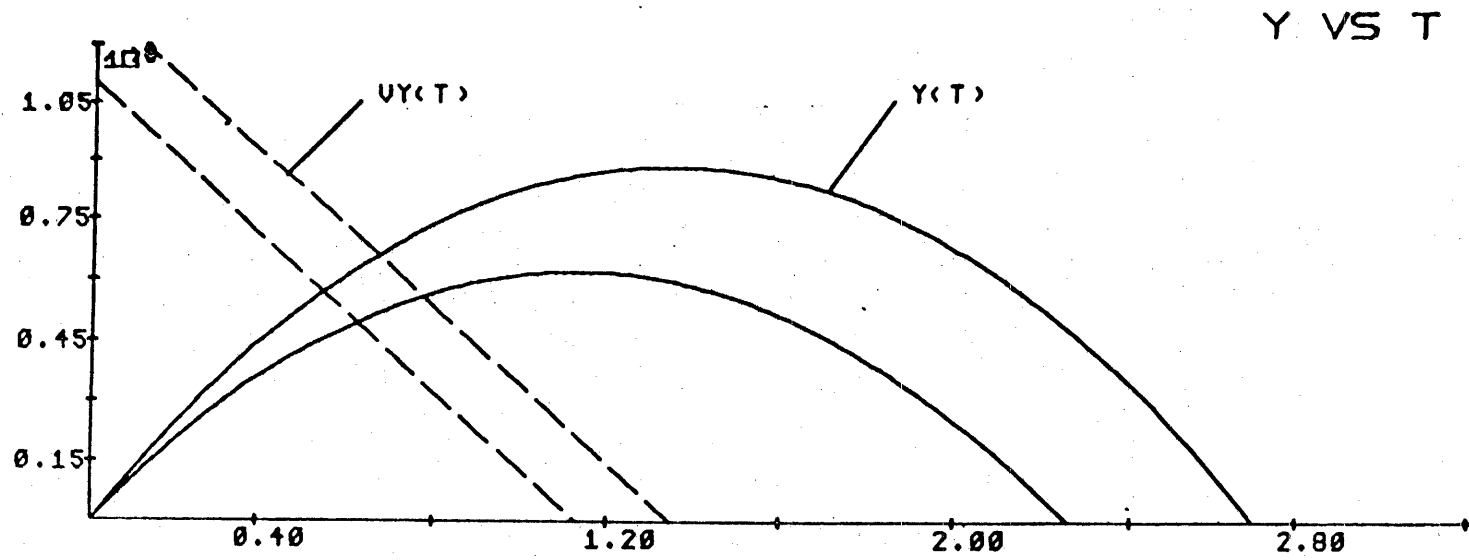
3. Linear Superposition (Figure 11).

   Illustrates:

   1. Definition of function in terms of previously defined functions (see p. 68).

   2. Horizontal SPLIT plot (see p. 38).

   3. ON,OFF - Dashes, Tics, Header (see p. 34).

   4. Display of function definition (see p. 31).

                        FD&S PHASE

   >F(X) = SIN(K*X)   (CR)
          K = 1   (CR)
   >G(X) = SIN(K*X + PH)   (CR)
          PH = 1.57   (CR)
   >S(X) = F + G   (CR)
   >PLOT,F,G   (CR)
   >SPLIT,H   (CR)
   >PLOT,S   (CR)
   >QUIT,DP,AUTO

                       DISPLAY PHASE

   INITIAL VALUE OF X = 0   (CR)
   FINAL VALUE OF X = 6.28   (CR)
   ORIGIN OF X = MIDDLE   (CR)
   NUMBER OF POINTS = 101   (CR)

   >OFF,TICS,HEADER   (CR)   (Off tics also turns off labels)
   >ON,DASHES,G
   >CLEAR   (CR)
   >DISPLAY,F   (CR) (position crosshairs) (DEL) (position
        crosshairs) (DEL) (position crosshairs) (space bar)
   >DISPLAY,G   (CR) (position crosshairs) (DEL) (position
        crosshairs) (DEL) (position crosshairs) (space bar)
   >DISPLAY,S   (CR) (position crosshairs) (DEL) (position
        crosshairs) (DEL) (position crosshairs) (space bar)

Figure 11

>DISPLAY,F
>D,G
>D,S
>

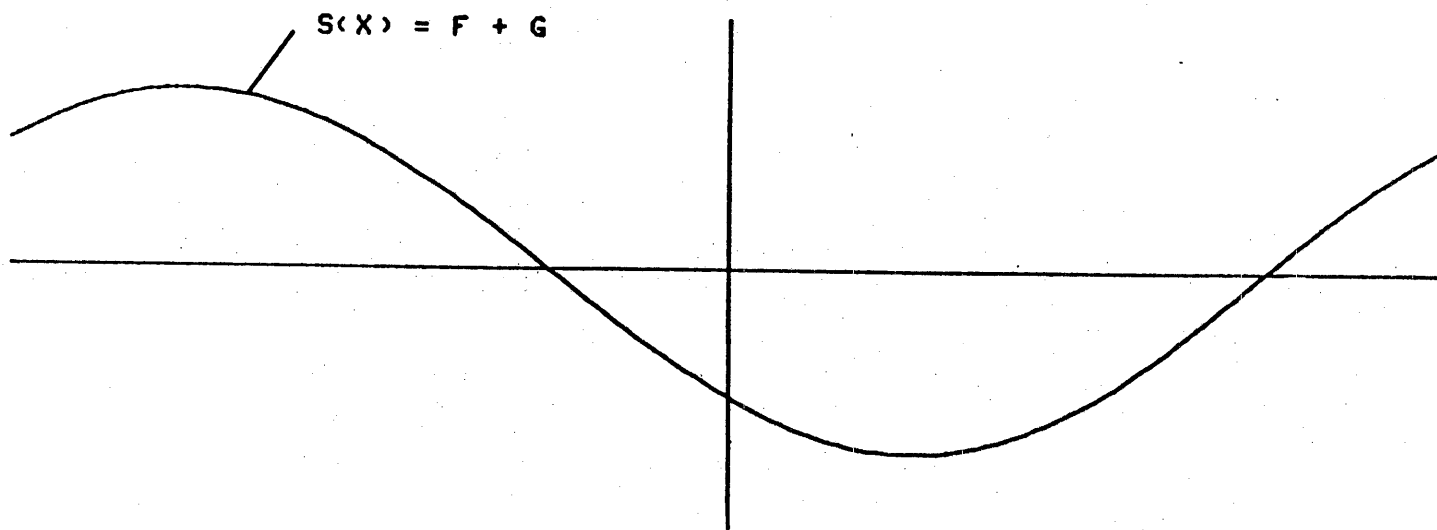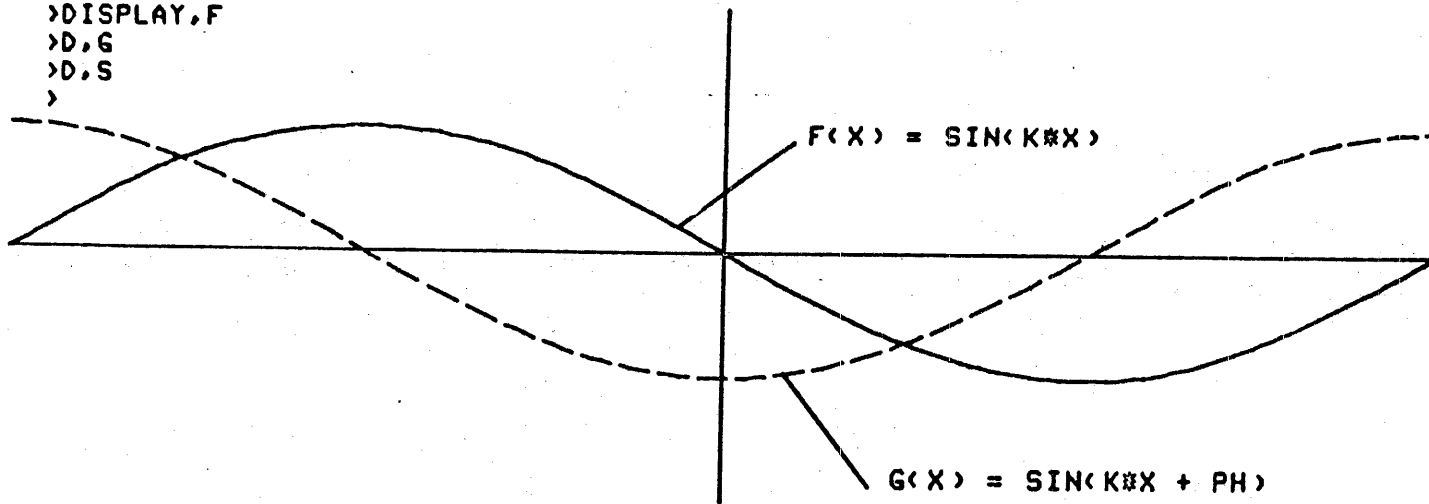F(X) = SIN(K*X)

G(X) = SIN(K*X + PH)

S(X) = F + G

23.

4. Kepler orbit (Figure 12).

   Illustrates:

   1. Polar plots (see p. 36).
   2. Changing parameter values by UP-ARROW (^) (see p. 41).
   3. Turning off axis labels (see p. 34).

<u>FD&S PHASE</u>

```
>R(Z) = A/[1 + E*COS(Z-B)]  (CR)
     A = 1   (CR)
     E = .5   (CR)
     B = 0   (CR)
>PLOT,R;POLAR   (CR)
> (CR)
```

<u>R&AS PHASE</u>

```
INITIAL VALUE OF Z = 0   (CR)
FINAL VALUE OF Z = 6.28   (CR)
NUMBER OF POINTS = 101   (CR)

MAXIMUM VALUE OF R = 6   (CR)
```

<u>DISPLAY PHASE</u>

```
>OFF,LABELS   (CR)
>CLEAR   (CR)
> (DEL) (position crosshairs) (DEL) (position crosshairs)
     (DEL)M E = .5   (CR)
>^E=.75   (CR)
>JUMP   (CR)
> (DEL) (position crosshairs) (DEL) (position crosshairs)
     (DEL)M E = .75   (CR)
>^E=1   (CR)
>JUMP   (CR)
> (DEL) (position crosshairs) (DEL) (position crosshairs)
     (DEL)M E =1.0   (CR)
>DISPLAY,R   (CR)
```
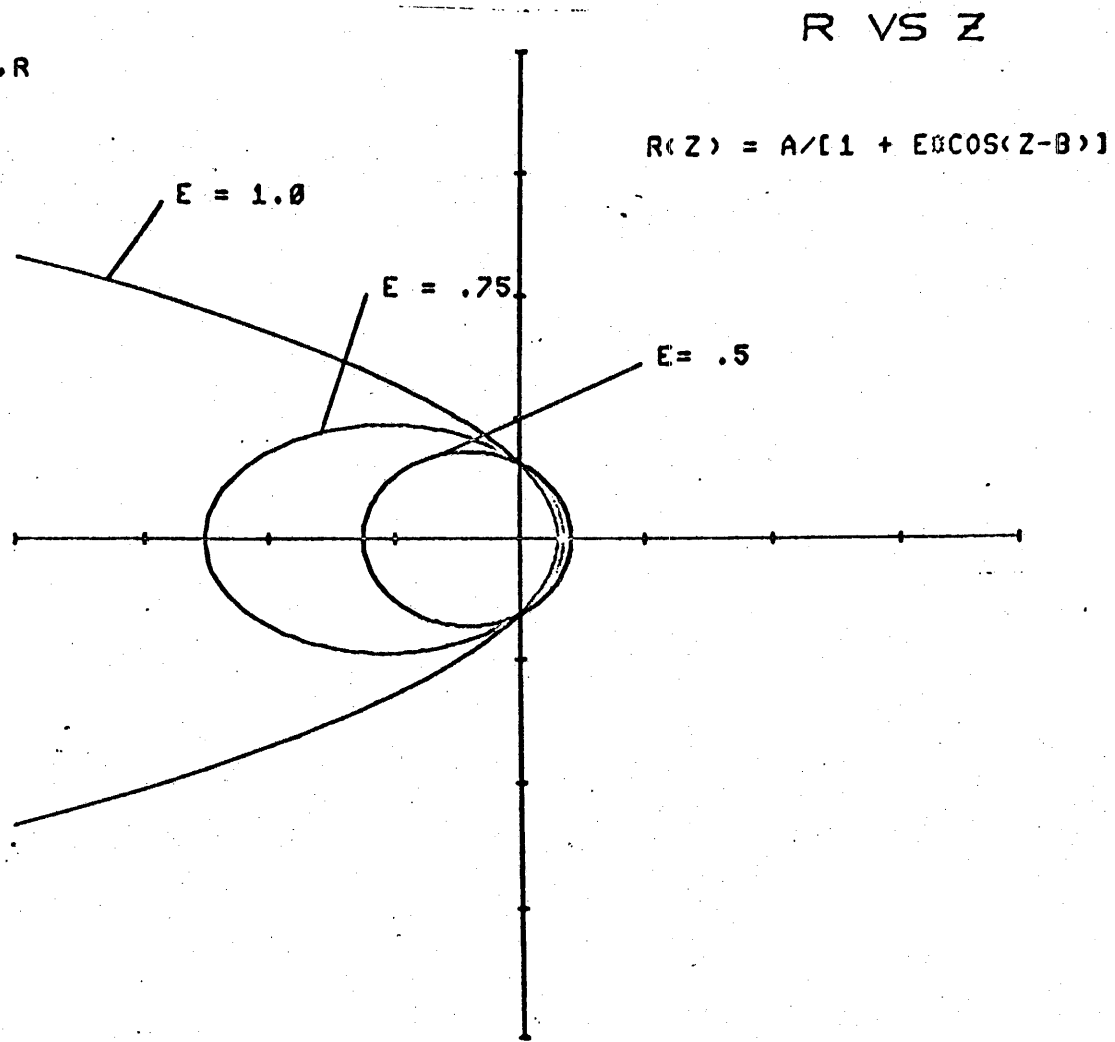
Figure 12

5. Damped driven oscillator (Figure 13).
    Illustrates:

    1. Defining differential equations (see p. 62).
    2. Split plots (see p. 38).
    3. Negative increments.

FD&S PHASE

```
>Y"(T) = -K*Y - B*Y'   (CR)
  (CR)                              (This (CR) terminates the
         Y = -1   (CR)               differential equation set.)
         Y' = 0   (CR)
         K = 1   (CR)
         B = .4   (CR)
>X"(T) = -K*X - B*X + F0*COS(W*T)   (CR)
  (CR)                              (This (CR) terminates the
         X = -1   (CR)               differential equation set.)
         X' = 0   (CR)
         FO = 5 BY -3   (CR)
         W = 3   (CR)
>PLOT,Y   (CR)
>SPLIT,H   (CR)
>PLOT,X   (CR)
> (CR)
```

R&AS PHASE

```
INITIAL VALUE OF T = 0   (CR)
FINAL VALUE OF T = 28   (CR)
ORIGIN OF T-AXIS = 0   (CR)
NUMBER OF POINTS = 200   (CR)

MINIMUM OF Y = -1   (CR)
MAXIMUM OF Y = 1   (CR)
ORIGIN OF Y-AXIS = 0   (CR)

MINIMUM OF X = -1   (CR)
MAXIMUM OF X = 1   (CR)
ORIGIN OF X-AXIS = 0   (CR)
```
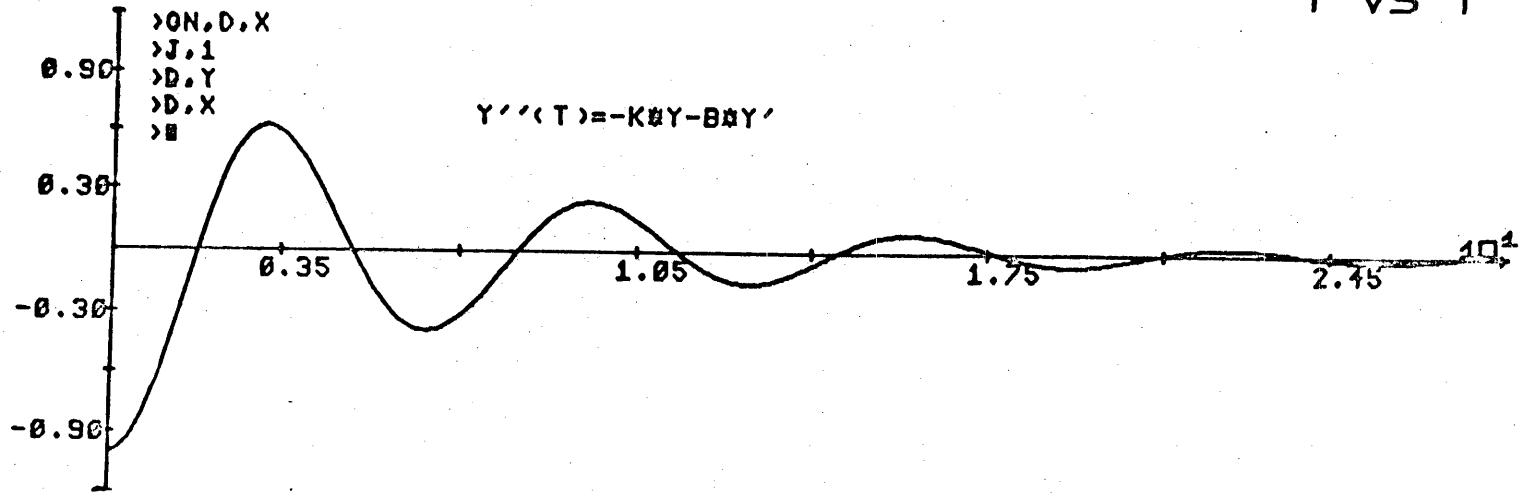
DISPLAY PHASE

```
>CLEAR   (CR)
> (DEL) (pos. crosshairs) (DEL) (pos. crosshairs) (DEL) (SPACE BAR)
>ON,D,X   (CR)
>J,1   (CR)        [or CR)]
>D,Y
>D,X
```

Y VS T

>ON,D,X
>J,1
>D,Y
>D,X
>■

$Y''(T)=-K◊Y-B◊Y'$

X VS T

$X''(T)=-K◊X-B◊X'+FO◊COS(W◊T)$

FO = 5.00

FO = 2.00

Figure 13

27.
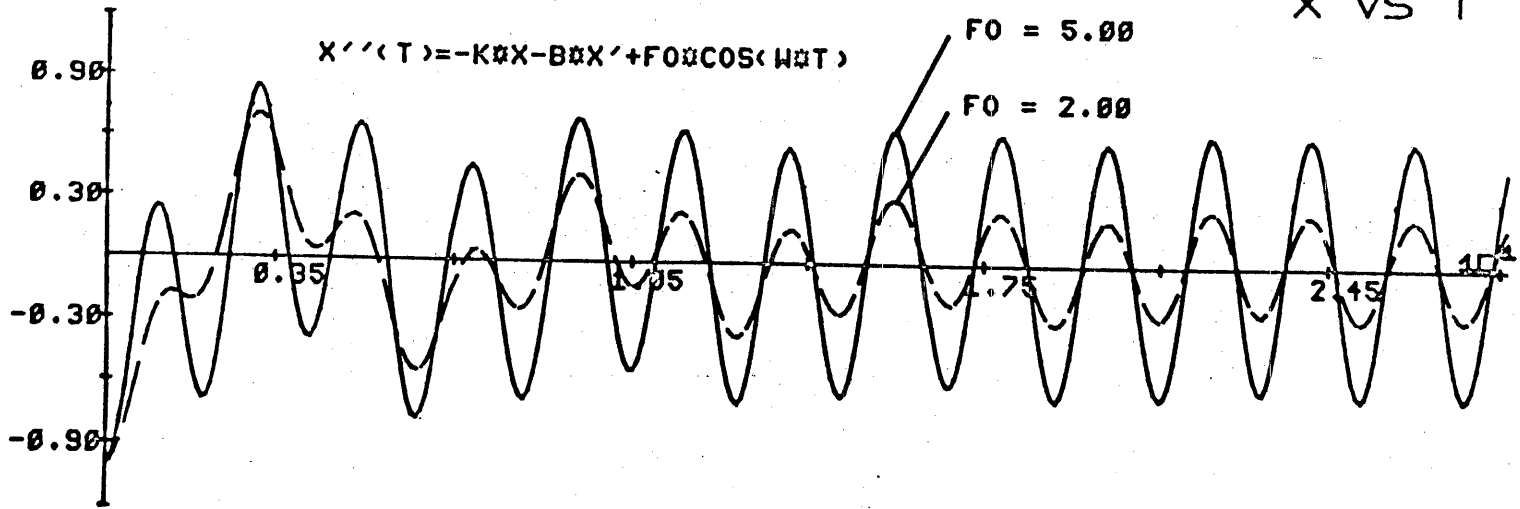
6. Fraunhofer diffraction (Figure 14).

   Illustrates:

   1. Conditionals in function definitions (see p. 59).

   2. Reserved word, PI = 3.1416.

   3. Using previously defined function in defining a new function (see p. 68).

   4. POWDER plots (see p. 35).

                        FD&S PHASE

>F(TH) = PI*SIN(TH)/LA   (CR)
       LA = 3   (CR)
>I(TH) = 1 IF F(TH) = 0; [SIN(B*F)/(B*F)]^2*COS(D*TH)^2   (CR)
       B = 5   (CR)
       D = 10   (CR)
>PLOT,I   (CR)
>QUIT,DS,AUTO

                       DISPLAY PHASE

INITIAL VALUE OF TH = -1.25   (CR)
FINAL VALUE OF TH = 1.25   (CR)
ORIGIN OF TH-AXIS = 0   (CR)
NUMBER OF POINTS = 151   (CR)

>OFF,LABELS
>CLEAR   (CR)
>ON,POWDER,3000   (CR)(Crosshairs are turned on.
>JUMP   (CR)   Define the diagonal of a
                        rectangle using (DEL).)
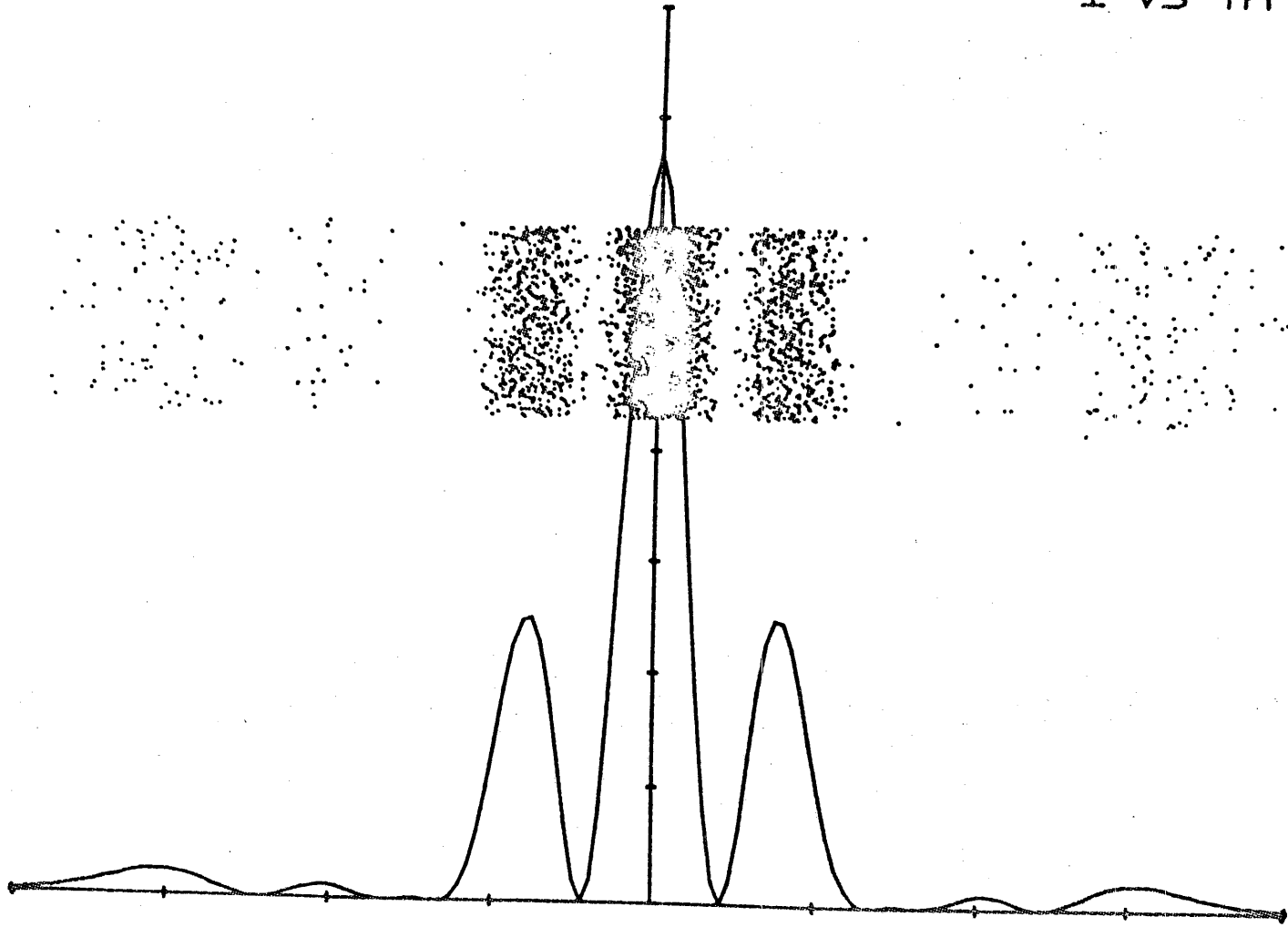
Figure 14

SECTION III

DETAILS OF FD&S COMMANDS

III.1 General Comments:

A. The signal to provide input is the greater than symbol, >, and/or the ringing of the bell.

B. Each complete line of alphanumeric input must be terminated with a Carriage Return (CR). Differential equation sets are terminated with an additional (CR).

C. Any line of input may be continued to the next line by depressing the Line Feed button (LF).

D. To exit from the FD&S phase depress (CR) or use the QUIT command.

E. Except for the ON and OFF commands, all commands and their modifiers may be abbreviated by their first letter.

F. Editing and error correction is achieved by using the backslash, \ , or the @ symbol. The @ symbol deletes a whole line and the backslash typed n times deletes the last n characters.

G. When returning to FD&S from other phases all function definitions are retained. All plotting specifications are deleted and new PLOT commands must be given.

FD&S COMMANDS

III.2   Description Of FD&S Commands

DISPLAY   D,funct1,funct2,...(CR)

The DISPLAY command provides a way of printing function definitions. This may be useful for classroom use or it may provide a way of checking to see if a function has been properly defined.

By typing DISPLAY,funct1,funct2,...(CR) the function names following the DISPLAY will be displayed. In FD&S the definitions will be printed in the normal sequence of command input. In the display phase, however, the crosshairs will be turned on, and after the crosshairs are positioned at a desired point, depressing the Space Bar will print the function definition(s) at that point. This allows the function definition(s) to be strategically placed.

An unmodified DISPLAY command will print all function definitions. In FD&S an unmodified DISPLAY command will first erase the screen and then print the functions in the upper left corner of the screen.

Examples:

        D
        D,X
        D,X,Y,Z
        D,X'

                    ****

FILE   F,name(CR)
GET   G,name(CR)

Often an instructor may not feel comfortable typing large amounts of information in front of a class or he may have spent considerable time preparing a particular display and does not wish to duplicate this effort. The FILE command provides a means for storing this information for use at a later date, and the GET command provides the means of retrieving it.

FD&S COMMANDS

The FILE command may be given in either the FD&S or the Display phase and has slightly different meanings when used in these two phases. When FILE is used in FD&S, it will store only defined functions and the <u>current</u> parameter values. When FILE is used in the Display phase, then in addition to function definitions and parameter values, the plotting specifications and the range and axis specifications are also stored.

To utilize the FILE command the user first creates a name, using the rules given in Appendix A for file names. He then simply types FILE,name(CR) and the appropriate information is stored.

To recall the stored information he issues the GET command by typing GET,name(CR). Immediately the stored function definitions and parameter values will be printed. If the information was originally filed in FD&S, then plotting specifications and range and axis information must be provided as usual. However, if the information was filed in the Display phase, then this additional information is retrieved and control is transferred to the Display phase and any desired Display command can be issued. The GET command must be given in the FD&S phase, and is illegal in the Display phase.

If functions which are defined prior to the GET command have names which are the same as the functions being retrieved by the GET, then errors will result and the command will not be executed. To correct this the KILL command (see page 33) must be issued to destroy duplicate function names. Also information up to the line causing the error must also be deleted before the GET command can be given again.

Examples:

        F,*HAROSC   (The * makes the file public.)
        G,*HAROSC
        F,^GAUSS   (The ^ makes the file semi-public.)
        F,BESS
        G,BESS

INFO   I,command (CR)

An unmodified INFO command issued in any phase will erase
the screen and list the commands which are legal in that phase.
If an INFO command is modified by a command name, then specific
information about that command will be printed. If two commands
begin with the same letter, e.g., PARAMETERS and PLOT, then infor-
mation is printed about both. Information about special keyboard
characters such as (CR), (DEL), and (Line Feed) is obtained by
modifying INFO with a typed character such as "C" for (CR).

Examples:

     INFO
     I,JUMP
     I,K

                    ****

KILL   K,functl,funct2,...(CR)

The KILL command is used to free function names for sub-
sequent definitions, or in extreme cases to free storage space
when the allocated space for function definitions is depleted.
(GROPE will store, at most, fifteen functions at one time.)

An unmodified KILL command will delete all previously defined
function and parameter names. If KILL is modified by the name
of a function or one member of a differential equation set, that
function or system of differential equations will be deleted.
When one function is defined in terms of another, as for example
F(X) = A*G(X), then GROPE will not allow the function G(X) to be
killed until F(X) is killed.

Examples:

     KILL
     K,X
     K,Y,Z
     K,X'

OFF   OFF,effect,modifier(CR)
ON   ON,effect,modifier(CR)
O   O,effect,modifier(CR)

        The ON and OFF commands are used to turn special effects
ON or OFF and the O command reverses the present status. If the
special effect is DASHES or VECTORS (see below) then the optional
modifier is a function name. If the effect is a powder plot, then
the modifier is a number specifying the number of points to be
generated in the powder plot.

        The effect names, which can be abbreviated by their first
letter, and their initial status are listed below.

EFFECT   INITIAL STATUS

AXIS   ON   OFF,A(CR) suspends the drawing of
                                   all axis sets, tic marks, and labels.

DASHES   OFF   ON,D,funct1,funct2,...(CR) causes
                                   the named functions to be dashed.
                                   If no functions are specified then
                                   all functions are dashed.

HEADER   ON   OFF,H(CR) causes the graph heading
                                   for each axis set to be turned off.

LABELS   ON   OFF,L(CR) suspends the placing of
                                   numerical values on the axis.

MARKS   OFF   ON,M(CR) causes distinguishing marks
                                   to be placed on each plotted curve.
                                   The marks are different for each
                                   curve, and the order of occurance
                                   is:

                                        1 = small x
                                        2 = small down arrow
                                        3 = small up arrow
                                        4 = small square
                                        5 = small triangle
                                        6 = small asterisk

QUARTER   ON   OFF,Q(CR) causes printing of the input
                                   and output information on the full
                                   screen. When the full screen is used
                                   the graphics information is not usable
                                   for classroom display but is appropriate
                                   for making 35 mm slides.

TICS	ON	OFF,T(CR) suppresses drawing of tic marks and printing of numerical labelling of the axis.
POWDER	OFF	ON,P,N(CR) causes a randomly generated powder plot representation of the functions specified by PLOT commands. The modifier, N, specifies the number of points to be generated for each plot.

After ON,P,N(CR) is typed all plotting is done in the powder mode. To return to normal plotting, type OFF,P(CR). Once the powder mode is entered para-meter manipulation and plotting gener-ation is performed as before. That is, the JUMP, CLEAR, and Carriage Return commands are used to generate new plots and all other Display commands maintain their usual meaning.

After a JUMP, CLEAR, or (CR) command is issued a set of crosshairs appear on the screen which allow the user to specify the area where the powder plot is to occur. The shape of the plot is rectangular and the diagonal is speci-fied by positioning the crosshairs to each end of the diagonal and depressing (DEL).

VECTORS	ON	OFF,V,funct1,funct2,...(CR) causes the named functions to be plotted as points with no vectors drawn between the points. If no functions are specified then all functions are plotted as points.

****

PARAMETERS   P(CR)

When there are many parameters in a given problem or when many incrementations have been made, it is easy to forget the current parameter values.

The PARAMETER command issued in any phase will list all current parameter values, initial conditions, and increments.

Examples:

    P

PLOT   P,funct1(T),funct2(T);system

The PLOT command specifies which functions are to be plotted, the type of coordinate system, and whether the plot is to be parametric or non-parametric. All functions which follow a given PLOT command will be plotted on the same axis set, and therefore they should have the same independent variable. If they do not, then the independent variable of the first function is assumed. Failure to give a PLOT command in FD&S will result in the error message, "PLEASE USE PLOT COMMAND BEFORE EXITING FROM FD&S". GROPE will not allow the user to proceed to R&AS or DISPLAY phase until a PLOT command is given.

GROPE has the capability of plotting functions in parametric form without requiring a formal parameter elimination between the two different functions. For example, if two functions Y(T) and X(T) are defined, then a plot of Y vs X can be obtained by the following command: P,Y(X)(CR). This is frequently a useful feature as for example in the trajectory problem discussed in SECTION I.

At present GROPE will plot in either cartesian or polar coordinates. The type of coordinate system to be used is determined by the modifier which follows the ";" in the PLOT command. For example, PLOT,R(G);POLAR(CR) will plot R(G) in a polar coordinate system. If the semicolon and coordinate system are omitted, then GROPE automatically assumes a cartesian system, so that in the majority of the cases it is <u>not</u> necessary to specify the desired system, i.e., P,Y;C(CR) and P,Y(CR) have identical meanings.

If you wish to plot the first or second derivative of an analytic function the following two sequences are equivalent:

```
>F(X) = X↑2 + COS(X)        >F(X) = X↑2 + COS(X)
>G(X) = F'(X)               >PLOT,F'(X)
>PLOT,G                     >
>                           .
```

Examples of how the plot might appear are given below.

Examples:

    PLOT,Y(T)

    P,X(T),Y(T)

    P,Y(X)

    P,R(TH);POLAR

****

QUIT   Q,phase,AUTO

In general the QUIT command is used to exit the present
phase of execution. Both modifiers are optional, and if they
are not specified, then the following set of options is presented:

       1. FD&S
       2. R&AS
       3. PS
       4. DP
       5. TERMINATE EXECUTION

The instructor can respond to these options by typing the
appropriate number (1 through 5) or by typing the first letter
of the desired option. As more familiarity with GROPE is acquired
the user will choose to jump from phase to phase by modifying the
QUIT command with the phase name to which he wishes to transfer.
For example, QUIT,PS will transfer control to the Parameter
Specification phase. With the capability of transferring from
phase to phase, it is possible for a transfer to be made to the
Display phase before a complete specification of Range and Axis
has been made. In this case control will be returned to R&AS
phase until the independent variable specification is made.

The second modifier, AUTO, is used to implement automatic
scaling when transferring to the Display phase. For example,
QUIP,DP,AUTO will transfer to the Display phase and implement
automatic scaling.

It is possible to use the QUIT command to restart the present phase of execution, and this feature may be useful in error recovery. For example, in either R&AS or PS errors may be made early in the input sequence and a fresh start of the phase is readily accomplished by using the appropriately modified QUIT command.

When QUIT is used to return to FD&S all previous plotting specifications are deleted; however, previous function definitions and current parameter values are retained.

Examples:

```
QUIT
Q,R
Q,D,Z
Q,F
Q,P
Q,T
```

****

RESET   R

The RESET command issued in any phase will set the current parameters to their initial values. If parameter values have been altered by the "^" command or in the Parameter Specification phase (see Section VI), then these altered values are considered to be the initial values.

****

SPLIT   S,type(CR)

Often, for comparison of different functions, it is advantageous to have two different axis sets with different functions on each set located in close proximity. For example the instructor may wish to have two sine functions which are out of phase on one axis set, and immediately below have the sum of the two functions plotted. The SPLIT command when used in conjunction with the PLOT command allows for this capability.

FD&S COMMANDS

The modifier TYPE specifies how the axis sets are to be arranged relative to one another and may be one of the following:

    a) Vertical
    b) Horizontal
    c) Superimposed

The Vertical or Horizontal modifiers split the display screen such that the two axis sets appear side by side or above and below each other respectively. Whereas, the superimposed modifier results in two independent axis sets being drawn on the same screen.

A PLOT command must precede and follow a SPLIT command, and each PLOT command defines a completely independent axis set and function specification. The functions specified by the first PLOT command will be located at the top of the screen or on the left-hand side depending on whether a Horizontal or Vertical split is used, and the graph specified by the PLOT following the SPLIT command will be located on the bottom or right-hand side of the screen, depending on whether a Horizontal or Vertical split is used. Since the two PLOT commands are independent a large variety of graphic displays may be generated.

To clarify these points assume that two functions have been defined by the following expressions.

$$X(T) = VO*T*COS(TH)$$
$$Y(T) = VO*T*SIN(TH) - G*T\uparrow2/2$$

The SPLIT command in FD&S might appear as follows:

      .
      .
      .

    PLOT,X(T)   (CR)
    SPLIT,H   (CR)   (or V or S)
    PLOT,Y,Y'(T)   (CR)

      .
      .
      .

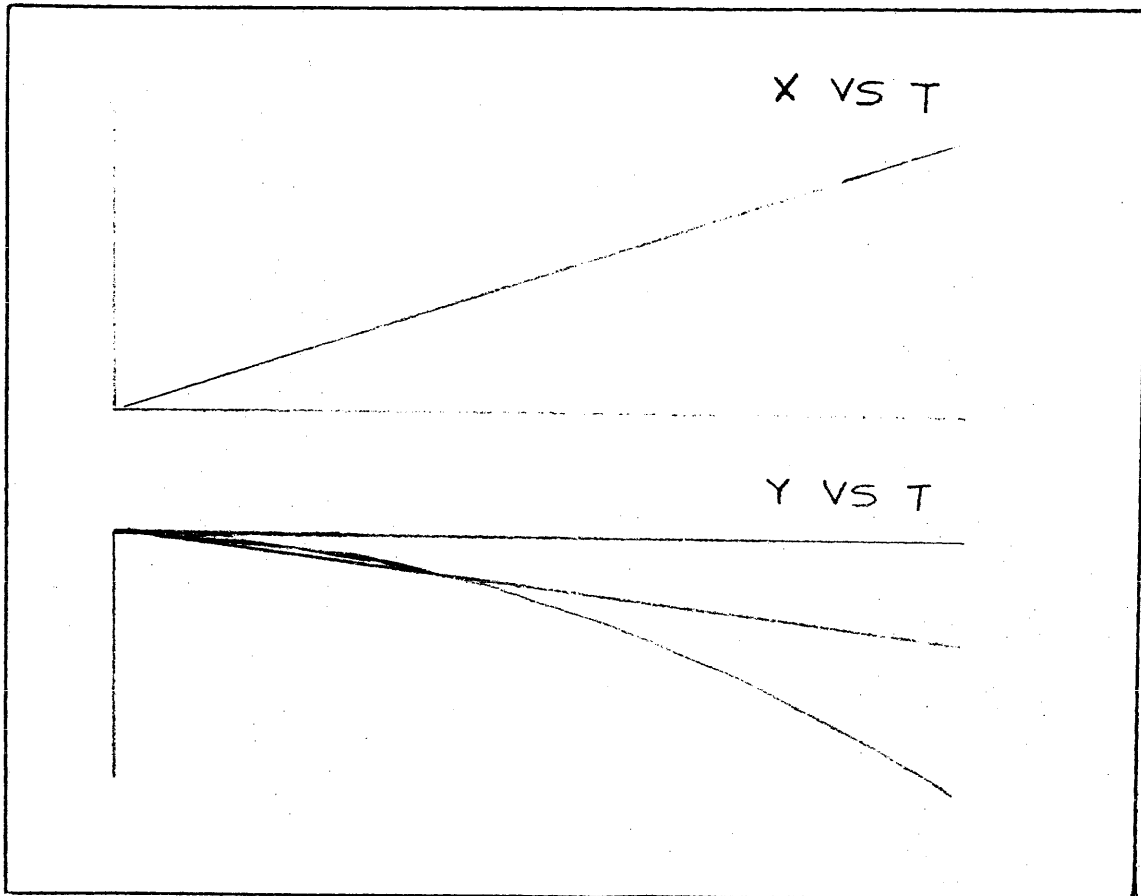The display for the horizontal split might appear as shown in Figure 15.

Figure 15

Notice that the lower graph contains both Y and Y', since both of these were specified in the second PLOT command.

Failure on the part of the instructor to give a PLOT command before and after the SPLIT will result in the system returning to FD&S until the omission is corrected.

Examples:

P,Y(X)	P,R(TH);POLAR	P,ENER(T)
S,H	S,V	S,S
P,X,Y	P,X,X'	P,MOM(T)

UP-ARROW(^)   ^pname = value1,pname2 = value2 BY ivalue,...(CR)

The ^ command is useful when the user wishes to change just a few parameter values, initial conditions, or their increments quickly without using the Parameter Specification phase. The ^ provides the following capabilities:

1) Current parameter values and increments can be altered.

2) Fixed parameters can be changed to varying parameters, and varying parameters can be changed to fixed.

To illustrate the use of this command assume that there are two parameters which are specified as

        A = 1 BY 2
        B = 3

To change the current value of A from 1 to 5, one would type ^A = 5(CR). Notice that the assigned increment, 2, is unaltered.

To change B from 3 to 6 _and_ to make it a varying parameter with an assigned increment of .5, one would type ^B = 6 BY .5(CR).

To change A from a varying parameter to a fixed parameter, one would type ^A = BY 0(CR) or ^A = 1 BY 0(CR).

It is possible to make multiple parameter and increment assignments by separating the individual assignments by commas. For example, in the above case one could type ^A = 2,B = .5 BY .01(CR).

Examples:

        ^A = 1
        ^A = BY .1
        ^A = 3 BY 0, B = 2, C = 1.2E-3 BY 7.5

SECTION IV

DETAILS OF RANGE AND AXIS SPECIFICATIONS

IV.1   General Comments:

A) Each response may be edited by using the @ or \ symbols.
    The @ symbol deletes the whole response and the back-
    slash deletes the previous character.

B) The asterisk, *, may be used to restart a sequence
    of instructions concerning a given axis. This is
    useful when an error is made in a previous line of
    input and the error is not noticed until later in the
    sequence (see Appendix C).

RANGE AND AXIS

IV.2   Types of Axis Specification Requests

At present GROPE will plot in either polar or cartesian
coordinates, as specified through the PLOT command in FD&S.
In addition to different coordinate systems, GROPE
will plot functions in parametric or non-parametric form and will
also allow for split plots, i.e., two independent axes sets. There-
fore, there are many different combinations of plots which might
result and the axis specification requests can vary considerably.
As an extreme example, we could have a non-parametric polar plot
and directly below this have a parametric cartesian plot.
Needless to say, the instructor could become quite confused as to
which axis specifications are required. However, GROPE is "smart
enough" to keep track of which questions are appropriate and it
will only ask those questions necessary for the requested display.
Usually no confusion should result.

Suppose we wish to make a single cartesian plot of a function
Y(T). The complete axis specification is accomplished by the
following questions:

           INITIAL VALUE OF T = -5   (CR)
           FINAL VALUE OF T = 0   (CR)
           ORIGIN OF T-AXIS = HIGH   (CR)
           NUMBER OF POINTS = 100   (CR)

           MINIMUM VALUE OF Y = -20.5   (CR)
           MAXIMUM VALUE OF Y = 10   (CR)
           ORIGIN OF Y-AXIS = LOW   (CR)

The above sequence is the basic unit of questions which are
asked regardless of how complicated the display might be. A pos-
sible confusing question in this sequence is the response to the
origin requests which can be answered with either LOW, MIDDLE, HIGH,
or a numerical value. In this example the origin of the T-AXIS is
placed at the high end of the T-AXIS, so that the vertical Y-AXIS
will intersect the T-AXIS at the right-hand end of the display screen.
Similarly since the Y-origin is given as LOW, the horizontal T-AXIS
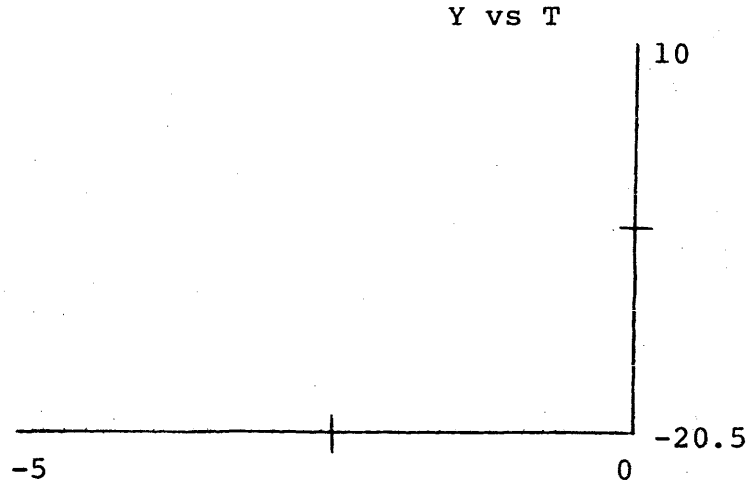is at the lower part of the display screen.

Y vs T

Figure 16

For a single polar plot of R(TH) the range requests and responses might appear as follows:

    INITIAL VALUE OF TH = 0   (CR)
    FINAL VALUE OF TH = 12.56   (CR)
    NUMBER OF POINTS = 100   (CR)

    MAXIMUM VALUE OF R-AXIS = 5   (CR)

Notice that in this case GROPE does not request information about the origin of the TH or R-AXIS, nor does it request minimum value or origin of the R-AXIS. This is typical of R&AS requests, so that spurious and meaningless questions are not asked.

For split plots there are two independent axis sets which must be specified, so the above sequence for either cartesian or Polar coordinates is requested for each axis set. However, if the functions plotted on the two axis sets have the same independent variable, then the range requests for the independent variable are requested only once and these values are used for both plots.

When parametric plots are made, e.g., for two functions X(T), Y(T) specified by PLOT Y(X), then information must be supplied for the ranges of T,Y,X, and also for the horizontal and vertical scaling. In some cases the user may want Y and X to have the same scales and in others he may want to have the scales different. So for all parametric plots GROPE asks whether the scale factors are to be the same.

IV.3   Commands in R&AS

Automatic

The automatic scaling is not intended as a cure-all since in determining the appropriate scale factors there are too many aesthetic judgements to be made. However, it can provide ballpark values for the ranges of the dependent variable when estimates would be difficult to make.

There are several different ways to utilize automatic scaling. The most direct way is to type A(CR)   in response to any dependent axis specification request. The following example illustrates this way of automatically scaling the range of the dependent variable, Y.

           INITIAL VALUE OF T = 0   (CR)
           FINAL VALUE OF T = 5   (CR)
           ORIGIN OF T-AXIS = LOW   (CR)
           NUMBER OF POINTS = 30   (CR)

           MINIMUM VALUE OF Y = A   (CR)

If in this example there are two axis sets to be specified, then GROPE will ask for the ranges of the second independent variable. The user may choose to either specify the ranges of the second dependent variable directly or to again request automatic scaling.

An alternate way of incorporating automatic scaling is to type QUIT,DP,AUTO(CR) in any phase of GROPE. This causes control to be transferred from the present phase to the Display Phase where automatic scaling will be provided.

QUIT

The QUIT command has exactly the same role as discussed on page 37. That is, it is used to change phases, terminate the program execution, or restart the entire phase.

SECTION V

DETAILS OF DISPLAY COMMANDS

V.1   General Comments:

A.   The signal to enter a command is the greater than symbol, >, and the ringing of the bell. The signal for graphic input information is the appearance of the cross-hairs.

B.   Each complete line of input must be terminated with a Carriage Return (CR).

C.   To exit from the Display Phase use the QUIT command.

D.   Except for the ON and OFF commands, all commands and modifiers may be abbreviated by their first letter.

E.   Editing and error correction is achieved by using the backslash, \, or the @ symbol. The @ symbol deletes a whole line and the backslash typed n times deletes the last n characters.

F.   Differential equations are solved using a numerical method which automatically controls the local error. When the solutions are numerically unstable the error message "HALVING TOO MANY TIMES" is printed. Sometimes it is possible to obtain stable solutions by returning to the R&AS phase and increasing the number of points.

V.2   Display Commands

<u>AUTOMATIC</u>   A(CR)

The data will automatically be scanned to determine new
ranges for the dependent variables before the next set of axis are
drawn.   New axis will be drawn after a CLEAR command or if the
screen has been erased and a command given to plot data.

****

<u>Carriage Return</u>   (n Space Bars) m(CR)

The Carriage Return is one of the most accessible keys
on the keyboard and for this reason it is designed to initiate
the most often exercised option in the Display Phase.   Usually
one wishes to increment the varying parameter values once and
generate the associated curves.   A single depression of (CR) will
accomplish this.

Depressing the Space Bar n times and then depressing
(CR) will increment the parameters (n+1) times and generate the
appropriate set of curves.   Typing an integer m followed by a
(CR) will increment the parameters once and plot the next m
members of each family of curves.   Moreover, combinations of n
Space Bars and an integer m preceding (CR) will increment the
parameters (n+1) times and generate the next m members of each
family of curves.   Because of these features the (CR) command
is a special case of the JUMP command and the following examples
illustrate the equivalence of the two commands:

          J,1,1   Equivalent to   (CR)
          J,2,1   "   "   (1 space Bar)   (CR)
          J,3,2   "   "   (2 space Bars) 2(CR)

Examples:

     (CR)

     3(CR)

     (1 Space Bar) 5(CR)

DISPLAY COMMANDS

CLEAR   C,r,m(CR)

The CLEAR command causes the screen to be erased and new axes
and curves drawn as specified by the PLOT and SPLIT commands.
The CLEAR command will increment the varying parameter values by
the assigned increment r times and then plot the next m members of
each family of curves. The real number, r, can have <u>any</u> positive
or negative value, so it is possible to increment or decrement the
varying parameters in a general manner. The modifier, m, specifies
the number of successive members of each family to be plotted, and
hence m must be a positive integer. If r and m are not specified
then the assumed values are 0 and 1, so that CLEAR(CR) uses the
current parameter values and plots one curve of each family.

As an example suppose we have the function, $F(X) = A + B*X$,
where A = 1 BY 1 and B = 2 BY 2. Then C,3,2 would plot F with the
following values:

$$\text{and} \quad \begin{array}{ll} A = 4 & B = 8 \\ A = 5 & B = 10 \end{array}$$

Examples:

C	equivalent to C,0,1
C,0,2	equivalent to C,,2
C,5,3	
C,3.2,2	
C,-2.1,3	
C,,3	equivalent to C,0,3

****

DEL   (DEL)(DEL)--------.
         (DEL)(DEL)--------(SPACE BAR)
         (DEL)(DEL)--------C (DEL)
         (DEL)(DEL)--------M message (CR)
         (DEL)(SPACE BAR)

When the (DEL) key is depressed a set of crosshairs will
appear on the screen. The crosshairs can be positioned by using
the joystick and depending on the subsequent sequence of operations,
the user can choose one of the following four features:

DISPLAY COMMANDS

1. Drawing Geometric Figures - (DEL)

   An arbitrary point on the screen can be defined by
positioning the crosshairs to that point and then
depressing the (DEL) key. Each subsequent repositioning
and depressing the (DEL) will define a new point, and
a straight line will be drawn between the last two
defined points. The sequence can be continued indefinitely
and in this way simple geometric figures which consist of
straight lines may be drawn. To terminate the sequence
and return to the usual command process a period "." is
typed.

   This capability of GROPE is useful for such features
as underlining important equations or it may be used in
conjunction with the SPACE BAR (see below) to label
curves with their varying parameter values as shown in
Figure 7, page 14.

2. Curve Labelling - (SPACE BAR)

   If at any time in the above sequence a (SPACE BAR)
is used in place of a (DEL), the current values of the
varying parameters will be printed. As many broken lines
as desired may be drawn prior to printing the parameter
values. In the usual case one line is sufficient for
clear labelling, so this case is simplified as shown by
the following sequence: (DEL)(position crosshairs to
some point on curve)(SPACE BAR). In this case a short
line (1/4") is drawn at 45° and the varying parameter
values are placed at the end of this line.

3. Coordinates of a Point - C

   If at any time in the above sequence a C is typed,
the coordinates of that point are retained and the cross-
hairs will reappear. After repositioning the crosshairs
and depressing the (DEL), a line is drawn to the position

and the retained coordinates printed. The sequence to
print the coordinates of an arbitrary point is shown by
the following:

> (DEL) (position crosshairs) C (position crosshairs) (DEL)

4. Displaying a Message - M

If at any time in the above sequence an M is typed,
an arbitrary alphanumeric message can be displayed by
the user. The message must be terminated by a (CR).

WARNING: When the crosshairs are on never respond with a (CR) as
this puts GROPE into a "wait state". If this occurs
accidentally, depress any other keys consecutively until
the crosshairs reappear and then answer properly.

****

JUMP   J,r,m(CR)

The JUMP command differs from the CLEAR command (see p.48)
in that the present screen is not erased before the functions are
plotted.

Examples:

        J   equivalent to J,0,1
        J,2   equivalent to J,2,1
        J,5,1,2
        J,-3,5
        J,,3   equivalent to J,0,3

SECTION VI

DETAILS OF PARAMETER SPECIFICATION PHASE

The Parameter Specification phase, which can be entered only
by using the QUIT command, provides the following capabilities:

1) Current parameter values and increments can
    be altered.

2) Fixed parameters can be changed to varying
    parameters, and varying parameters can be
    changed to fixed.

Each of these capabilities can be achieved by using the ∧
command (see p. 4)   in FD&S or Display phase, but it is frequently
easier to perform this task in the PS phase, particularly when
there are more than one or two parameters to change.

When entering the PS phase a message is displayed, all current
parameter values and increments are listed, and a value for the
first parameter is requested. For example, if we have defined
only one function X(T) by a differential equation, e.g.,
X"(T) = A*Y'-B*X+B, then when entering the PS phase the screen
might appear as shown in Figure 17.

```
Parameter Specification Phase


Press CR after each parameter request for which the
current value and increment are to be maintained,
new values and/or increments may be entered as in FD&S.
A =                     A = 1.00 BY .5
                        B = 2.00
                        C = 4.00
                        X = .5 BY .1
                        X' = 1.00
```

Figure 17

The last element printed by the computer in Figure 2 is A =.
GROPE is requesting information for the parameter A, and at this
point the instructor has the following options:

PARAMETER SPECIFICATION

1) The current parameter value and increment will remain unchanged if (CR) is depressed.

2) The current value of A can be altered by typing a number, e.g., A = 3(CR).

3) The current increment of A (.5) can be altered by typing "BY" and the new increment, e.g., A = BY 1.5(CR). If the new increment is <u>zero</u> then A is changed <u>from</u> a varying parameter <u>to</u> a fixed parameter. If A were originally a fixed parameter, then typing "BY" and an increment would make A a varying parameter.

4) Both the current value and the increment may be altered by combining options 2 and 3.

After one of these options for A has been chosen, GROPE will request the next parameter value by printing B = and ringing the bell. The above options are now available for B. This sequence is repeated until all parameter values and initial conditions are specified.

Errors made while entering a new value may be corrected by using the backslash \ and at sign @ to delete previous characters or the entire line. If one should discover an error made on a previous line, the entire sequence can be restarted by typing *(CR)   or using the QUIT command to transfer to the PS phase, i.e., Q,P (CR).

The QUIT command (see p. 37)   can be used to terminate the sequence and transfer to another phase. This will have no effect on the values of parameters already specified.

SECTION VII

RULES OF FUNCTION DEFINITION

There are three distinct classes of functions which can be made available for plotting or for use in making other function definitions. For two of these classes the function definitions are established on-line by the user himself. These are (1) functions defined through analytic expressions, and (2) functions defined as the solution to a typed-in differential equation or a differential equation set. Sections VII.1 and VII.3 describe these classes, respectively. A third category of functions are those which are defined internally by GROPE. A list and description of these internal functions is the content of VII.4.

Sometimes an equation may take one analytic form under certain conditions, and have another form under other conditions. The provisions for such definitions are described in VII.2. Finally, some formal definitions and restrictions on the quantities occurring in function-defining statements are given in VII.5.

VII.1   Analytically Defined Functions, On-Line

A function which can be expressed analytically can be defined
on-line by associating the analytic expression with a function
name. The purpose of this sub-section is to establish the rules
of construction of expressions and provide the rules for their in-
terpretation. Generally, these rules parallel those of ordinary
mathematics, so for the most part this sub-section may just serve
as a reminder as to what those rules are.

Let us start with an example. If f(x) is defined mathematically
by

$$f(x) = a + bx + cx^d/2$$

then the same definition can be provided to GROPE by

F(X) = A + B*X + C*X ∧ D/2.

F would be the function name which is associated with the expression
following the equality sign. The computer, just like the mathe-
metician, treats X as a dummy argument. A subsequent reference
to F(2.5), for example, would cause a value to be calculated for
F with X in the above expression replaced by 2.5. The quantities
A, B, C, and D are called <u>parameters</u> in GROPE. Immediately
following the function definition, values would be requested for
any parameters which had not already had values assigned to them.

In FD&S the definition would be typed in as given above,
followed by (CR). In Display Phase, function definitions must
be preceded by the ∧ character.

Expressions consist generally of operands separated by
operators, although an expression may also consist of a single
operand. The operands in the above example are A, B, X, C, D,
and 2, in the order of their appearance, but they can also be more
complicated, as described below. The allowed operators in GROPE
are +, -, *, /, and ∧, indicating addition, subtraction, multipli-
cation, division, and exponentiation, respectively. Other allowed

symbols are: parenthesis and brackets, for grouping of terms and
for enclosing the arguments of a function; and the single (') and
double (") quotes, indicating function differentiation.

When either + or - separate two operands, as in the above
definition of F, they are called binary operators, and mean addition
or subtraction. On the other hand, the - can also be used with
a single operand, such as

  -5.12 or -B,

meaning to negate the following operand before performing any other
operation with it. The + can be used in the same way, for emphasis,
even though it produces no change. When the + or - are used in
this way they are called unary operators.

The use of unary operators very seldom leads to confusion,
because their usage in GROPE is almost the same as the ordinary
mathematical usage. To be complete, however, requires a few rules
of convention:

Rule 1: Two operands can never stand adjacent to each other.
   A binary operator must always occur between them, and
   the operator must immediately follow the first of the
   operands.

Rule 2: Any operator not preceded by an operand is interpreted
   as a unary operator. The only allowed unary operators
   are + and -.

Rule 3: Every operator (both unary and binary) must be followed
   by either an operand or a unary operator.

These simple rules are a little bit further complicated by the
use of parentheses. Remember, parentheses are not operators, nor
are they operands. Rule 2 thus implies that the - in the form

  ... (-A ...)

is a unary operator. However, any valid expression when enclosed
in parentheses becomes a valid operand. Thus the form

  (... B) + (A ...)

is legal. The expression inside each pair of parentheses would be

evaluated and the resulting pair of <u>numbers</u> would be the operands for the indicated binary addition operation.

Examples:

(A)(B)   is illegal   (contrary to mathematical notation)

A*-B   means A*(-B)

--A/+B   is equivalent to A/B

A^-B   means A^(-B)

(-A^B) + C means ((-A)^B) + C   (Contrary to mathematical notation)

C - A^B means C-((A)^B)

C + (-A^B) means C + ((-A)^B)   (Contrary to mathematical notation)

(A + B-)   is illegal

Another area in which ambiguities can arise is exemplified by the expression

d + b*c.

Convention interprets this to mean d + (B*C), and the other possibility, (d + b)*c, <u>requires</u> the use of parentheses. Ordinary convention evidently establishes a "precedence of operation", whereby some operations are to be carried out before others are. These conventions are embodied in the following precedence table for <u>binary</u> operations:

<u>precedence</u>	<u>operation</u>	<u>operator</u>
highest	exponentiation	^
middle	multiplication, division	*,/
lowest	addition, subtraction	+,-

Three more rules are necessary to tidy up the conventions when there is ambiguity about which operation should take place first:

Rule 4:   Unary operations involving an operand are carried out before any binary operations involving the operand are carried out.

Rule 5: Binary operations of higher precedence are carried out before operations of lower precedence.

Rule 6: If ambiguity remains after Rules 4 or 5 are applied, the ambiguity is removed by carrying out operations from left to right.

Examples:

$-A \wedge B$	means	$(-A) \wedge B$	not	$-(A) \wedge B$
$A*B \wedge 2$	means	$A*(B \wedge 2)$	not	$(A*B) \wedge 2$
$A \wedge B*C$	means	$(A \wedge B)*C$	not	$A \wedge (B*C)$
$A + B \wedge C$	means	$A + (B \wedge C)$	not	$(A + B) \wedge C$
$A/2 + B$	means	$(A/2) + B$	not	$A/(2 + B)$
$A \wedge B \wedge C$	means	$(A \wedge B) \wedge C$	not	$A \wedge (B \wedge C)$
$A/B*C$	means	$(A/B)*C$	not	$A/B*C)$.

A good rule to follow when there is chance for confusion or ambiguity is to make the grouping explicit by the use of parentheses.

The following are examples of valid sequences of function definitions in the FD&S Phase of GROPE:

Sequence 1

$X1(T) = A1*COS(W1*T + P1)$

$X2(T) = A2*COS(W2*T + P2)$

$PE(T) = M*((W1*X1) \wedge 2 + (W2*X2) \wedge 2)/2$

$KE(T) = M*(X1' \wedge 2 = X2' \wedge 2)/2$

$E(T) = KE + PE$

(COS is the internally defined cosine function. (See p. 65 .) A1, A2, W1, W2, P1, P2, and M would be established as parameters.)

Sequence 2

$GAUS(X) = EXP(-(X/A) \wedge 2)$

$PULS(X) = GAUS(X - V*T - VO)$

(EXP is the internally defined exponential function. (See p. 65.) A, V, T, and XO would be established as parameters.)

FUNCTION DEFINITIONS

Sequence 3

F(X) = (X - V*T)/A

PULS(X) = EXP(-F(X - X0)∧2

Sequence 4

D(X1, Y1, X2, Y2) = SQRT((X1 - X2)∧2 + (Y1 - Y2)∧2)

V(X1, Y1, X2, Y2) = K/D

PE(X1, Y1, X2, Y2) = V - Z*(V(X1, Y1, 0, 0) + V(0, 0, X2, Y2))

(SQRT is the internally defined function for taking the square
root. (See p. 65 .) K and Z would be established as
parameters. Sequence 3 and Sequence 4 result in the same
function definition for PULS.)

VII.2   Functions that Require More Than One Expression; Conditional

Discontinuous functions and functions like

sin(x)/x,

for example, cannot be defined for the computer by the simple form

name(arg) = expression.

GROPE contains the capability to facilitate the definition of
such "multi-expression" functions. For example sin(x)/x could
be defined by the statement (read "else" for the ';' and the
meaning is apparent):

F(X) = SIN(X)/X IF X#0;1   (CR)

or by

F(X) = 1 IF X = 0; SIN(X)/X   (CR)

As another example, a unit step function with "discontinuity"
at A could be defined by

STEP(X) = 0 IF X<=A; 1 IF X>A   (CR)

There are two forms for the multi-expression function
definition;

name(arg) = expression IF condition;
            expression IF condition;
                .
                .
                .
            expression IF condition

and

name(arg) = expression IF condition;
            expression IF condition;
                .
                .
                .
            expression

FUNCTION DEFINITIONS

(The latter form was used for F(X), above, and the former for
STEP(X).) The characters 'IF' must be preceded and followed
by a blank. The word "condition" represents a "statement" whose
truth will be determined by the computer. If it is true the pre-
ceding expression will be evaluated as the value of the function,
if it is not true then the next condition will be examined, and
so on. If a condition is found to be false and there is no next
condition then the next expression will be evaluated and returned,
unless it is also missing in which case the value 0 is returned.
For example,

        STEP(X) = 1 IF X>0

and

        STEP(X) = 0 IF X<=0;1

would both return 0 if X is less than or equal to 0, and 1 otherwise.

   The "condition" has the general form:

expression   operator   expression   operator   expression

The three expressions are compared accordingly to the "relational
operators". The possible relational operators are:

relational operator	meaning
=	equals
<	less than
<=	less than or equals
>	greater than
>=	greater than or equals
# or <>	not equals

Example:

   A sawtooth function with amplitude A could be defined for
   N cycles and be zero elsewhere, by

FUNCTION DEFINITIONS

```
F(X,N) = K*X - N*A
SAW(X) = 0 IF X 0;   (Line Feed)
         F(X,0) IF F(X,1) < = 0;   (Line Feed)
         F(X,1) IF F(X,2) < = 0;   (Line Feed)
            .
            .
            .
         F(X,N - 1)  IF F(X,N) < = 0 (CR)
```

FUNCTION DEFINITIONS

VII.3   Differential Equation Sets

Functions can be defined in GROPE by specifying an ordinary differential equation or differential equation set that the function satisfies.* The set must be arranged as a set of algebraic expressions for the highest derivative of each of the dependent variables occurring in the set. That is, the set must be solved <u>algebraically</u> for these highest derivatives. For example,

$$L_1 q_1'' + M q_2'' + R_1 q_1' = E(t)$$

$$L_2 q_2'' + M q_1'' + R_2 q_2' = 0$$

would be specified to GROPE as

    Q1"(T) = [L2*(E - R1*Q1') + M*R2*Q2']/(L1*L2 - M*M)

    Q2"(T) = [M*(E - R1*Q1') + L1*R2*Q2']/(M*M - L1*L2)

Thus a differential equation set, as seen by **GROPE**, consists of a set of definitions for the highest derivatives of each dependent variable. Another restriction is that the independent variables indicated as arguments of these highest derivatives must agree in name and number for all members of the set.

The rules for constructing the expressions on the right-hand side of the equality are almost identical to those for analytic function definitions. (See Section VII.1.) The only difference is that <u>any</u> of the dependent variables and their derivatives can be referrenced in <u>any</u> of the members of the set. This is illustrated by the example above, where Q2' appears in the expression Q1" before the definition of Q2" has been entered.

---

*Functions thus defined are not actual solutions, but are numerical approximations to the solutions. In the vast majority of cases, they are accurate approximations graphically undistinguishable from the true solutions.

After each differential equation has been entered, followed by (CR) as usual, GROPE will sound the bell, indicating it is waiting for another member of the set. Notice that the usual '>' is not printed, because GROPE is not waiting for a command, it is waiting for a continuation of the set. To signal the end of the set, the (CR) key must be depressed after the bell has rung. After the entire set has been entered, and terminated by the additional (CR), the values will be requested for the initial conditions on the dependent variables and for the parameters that were defined. In the above example the latter would consist of Q1, Q1', Q2, and Q2'. Their values will correspond to initial conditions at the initial value of the independent variable, as specified in R&AS phase. The "value" may be an expression.

Considerable typing can sometimes be avoided by using an abbreviation for a combination of the dependent variables. For example, the set for Q1 and Q2 above might be entered as

$$D(T) = L1*L2 - M*M$$

$$Q1''(T) = [L2*F + M*R2Q2']/D$$
$$Q2''(T) = -[M*F + L1*R2*Q2']/D$$
$$F(T) = E -R1*Q1'$$

(Diff. Eq. Set)

Note that the abbreviation F could not have been entered prior to the Q1" equation, as D was, because GROPE would not have recognized it as part of a differential equation set, and would therefore have considered Q1 to be undefined. It is more economical to enter abbreviations prior to entering the set whenever this is possible.

GROPE waits until the entire set of equations has been entered before processing any of it. Consequently any errors will not be discovered until then. If it does find errors, it will print an appropriate error message and ask if it is entirely possible that the error did not really occur there. You will be asked which equation(s) you wish to change. If the first and third equation require changing, for example, the reply would be

1,3 (CR)

and the corrected versions of equations 1 and 3 should then be typed. An example of error correction for differential equations is given in Appendix C.

A commonly made error is that the final (CR), signifying the end of the set, is forgotten. The subsequent commands will then incorrectly be considered to be part of the set and so will be discovered to be in error by GROPE. These "equations" must then be deleted. This is accomplished by indicating their equation numbers as ones to be re-entered, and then depressing only the (CR) key as their "corrected" version.

FUNCTION DEFINITIONS

VII.4   Internally Defined Functions

GROPE provides certain mathematical functions which are commonly used in scientific work, such as sine, arctangent, absolute value, and integral. Each function has a preassigned name which consists of from 2 to 4 characters and an argument which determines the value of the function. The argument(s) must be enclosed in parentheses and immediately follow the function name, e.g., SIN(X). The argument(s) may consist of any valid analytic expression (see VII.1).

Table 3 gives a complete list of internally defined functions and their equivalent computation.

NAME IN GROPE	COMPUTATION
SIN(X)	Sine of X radians.
COS(X)	Cosine of X radians.
TAN(X)	Tangent of X radians.
SQRT(X)	Square root of X.
ABS(X)	Absolute value of X.
COSH(X)	Hyperbolic cosine of X.
SINH(X)	Hyperbolic sine of X.
TANH(X)	Hyperbolic tangent of X.
ASIN(X)	Arcsine of X.
ACOS(X)	Arccosine of X.
ATAN(X)	Arctangent of X.
EXP(X)	e raised to the power X.
LN(X)	Natural logarithm of X.
LOG(X)	Base 10 logarithm of X.
INT(F,A,B,N)	Integral of the function F between the limits A and B. F may not be related to a differential equation set. The technique used involves subdividing the interval [A,B] into subintervals computing the integral by a quadrature formula and summing these to get the value of the integral of F. N is the number of subintervals to be used and should be chosen carefully.

Table 3

FUNCTION DEFINITIONS

The functions listed in Table 3 can be used as an operand in any analytic expression. Thus, in the expression

    1.5*X + EXP(-3*X)

the function e^{-3X} would be computed and the result added to 1.5X.

The following example illustrates the use of the sine and integral in defining functions.

    VY(T) = V0*SIN(TH)-G*T
    Y(T) = INT(VY,0,T,4)

VII.5   Operand Restrictions

An operand can be any one of: numerical constant, parameter name, dummy argument name, function name, function derivative name, or an otherwise valid expression enclosed in parentheses, (), or brackets, [].

Numerical Constants: These are numbers like 5, 5., 55.5, .55, etc. In addition, multiplication by an integer power of ten, useful for very small and very large numbers, can be indicated by adding an "E" followed by the desired 1 or 2 digits integer exponent. Examples are:

5E6	meaning	5×10^6
4.5E+6	meaning	4.5×10^6
.5E-6	meaning	$.5 \times 10^{-6}$

Examples of illegal forms are 5E.5, E3, and 1.0 E 3. The latter is illegal since no spaces may preceed or follow "E".

Parameter Names and Dummy Argument Names: These two types are very similar. They both correspond to the mathematical concept of an algebraic variable. The names themselves can consist of 1 or 2 characters, the first of which must be a letter (A - Z), with the second either a letter or a number (0 - 9). The difference between them is a formal one. A dummy argument name occurs as one of the arguments on the left hand side of the function definition. Parameter names do not. For example, in the definition

F(X,Y) = A*X^2 + B*Y^2,

X and Y are dummy argument names, and A and B are parameter names. Dummy argument names have no meaning outside the function definition in which they occur. (Except they may appear as graph labels-- See the PLOT command, p. 36 .) The parameter names are established "permanently", on the other hand, and their values, which are immediately requested by the computer, will be the same for any other function in whose definition they occur. Parameter names can only be removed by a KILL command (see p. 33).

FUNCTION DEFINITIONS

A <u>function name</u> serves to identify a defined function. It may consist of from 1 to 4 letters or numbers, with the first being a letter. Each defined function must have its own distinct identifying name. A function name can be used as an operand anytime after that function has been defined. (See VII.3 for details about differential equation sets.) As an operand, it may occur either alone or with arguments following, enclosed in parentheses and separated by commas. If the arguments are omitted, then the dummy arguments of the function being defined are assumed, so their number must agree. For example the F(X,Y), defined above, could be used in the following ways:

        G(X,Y) = F^2

        G(X,Y) = F(X,Y)^2
        (These two examples are equivalent.)

        G(X) = F(A,B)*X
        (Values for A and B would be requested if they
        had not been defined previously.)

The forms

        G(X) = F

        G(X,Y,Z) = F

would be illegal however. Any valid expression can specify the argument of a function used as an operand except that the name of the function itself, or its derivatives, may not occur in it.

Examples:

        G(X) = F(X, X^2 + Y^2)

is legitimate.

Examples of function names:

        A
        AB
        A12
        ABCD

but not ABCDE (this will be interpreted as **ABCD**), 3A1, or 2.

FUNCTION DEFINITIONS

A <u>Function Derivative Name</u> is used to refer to derivatives of a <u>previously defined</u> function. A single (') or double (") quote is appended to the function name to indicate the first or second derivative of the function, respectively. Combinations of these symbols can also be used. The highest derivative that can be taken is 2, unless the function is defined via a differential equation set (see VII.3).

Function derivatives can be used in just the same ways as functions themselves, with the following two restrictions:

1) Derivatives may not be subsequently taken of a function that is itself defined in terms of function derivatives. I.e., a combination like

    F(X) = G'(X)
    H(X) = F'(X)

   is not allowed.

2) If a function derivative appears on the left hand side of an equality sign it indicates the equation is a differential equation.

Examples of function derivative names:

    F'
    F'(X)
    F"

APPENDIX A

APPENDIX A

USING THE TEKTERMINAL AND OS-3

A.1   Logging On From The Tekterminal:

To communicate with the computer and to load GROPE using the tekterminal a series of four simple steps is necessary.

Step 1:   Set Control Indicators on Terminal

a) Power OFF/ON - This key should be left in the ON position even when the terminal is not in use. To turn the terminal on turn the key clockwise 90° and wait 15 seconds until the screen appears bright green. Then depress the ERASE indicator.

b) ON-LINE/LOCAL - Must be ON-LINE for communication with the operating system.

c) ASCII/TTY - Set to TTY.

d) KEYBOARD/AUX - Set to both KEYBOARD and AUX.

e) DIRECT/COMPOSE - Set to DIRECT.

f) The settings of other indicators are arbitrary.

Step 2:   Depress ERASE indicator to clear the screen.

Step 3:   Logging On

a) Depress the SOH key located in the upper part of the right-hand keyboard. The computer should respond by displaying a pound sign (#) in the upper left corner of the screen. If it fails to do this after several attempts, recheck the indicator settings and depress the HOME button. If this also fails then most probably the computer is malfunctioning. To verify that this is the situation call the computer center operations desk, ext. 2033.

b) After the computer types the pound sign it is in the control mode and you should respond within 20 seconds by typing your user's job number and the four letter validity code followed by a carriage return (CR).

#123456 ZAPO   (CR)

If the numbers and letters are valid the tek-terminal will overprint these so that they are unreadable, print out the current date and time, the terminal number, and then type another # symbol. The computer is then waiting another command.

If after the carriage return nothing happens, repeat steps 3a and 3b.

IMPORTANT: If at any subsequent time you wish to quit it is important that you LOGOFF. (See A.2 below.)

Step 4: Loading GROPE

When you have successfully logged ON and the computer has typed the second # symbol you should type *GROPE (CR). After GROPE has been loaded, the screen will be erased and the message shown on page 6 will be displayed. The system is then in the FD&S phase and you are ready to proceed with execution.

A.2   Logging Off From The Tekterminal:

If at any time during the execution of GROPE you wish to quit, you should always LOGOFF. The preferred way of doing this is to type QUIT,T(CR) wait for the # symbol to appear, and then type LOGOFF (CR). The computer will respond by typing

TIME XX.XX SECONDS, MFBLKS X COST $X.XX

If the computer fails to respond in this way, you have not logged off, and the user's job number will continue to accrue charges. In the event that the above message is not printed, you should depress the SOH button and type LOGOFF (CR) again. If after several attempts you cannot succeed, the computer is probably malfunctioning. In this case telephone the operations desk, ext. 2033, and verbally request to be logged off.

A.3   Useful OS-3 Control Mode Statements

FP,<file name>   (File Protect)

This command will protect the file specified so that it cannot be inadvertantly written on and destroyed.

GO

Allows the program to continue execution at the point it was interrupted. The interruption may have occurred because of a time cut or hitting the break or SOH keys.

MI

The MI will return control to the GROPE command mode. One may want to use this command if he has intentionally gone to control mode due to some error.

RFP<file name>   (Remove File Protection)

This command removes file protection from the file named. The file can then be written over.

TIME=N

The TIME command is used to set a limit on the number of seconds of computer time to be used. If a time cut has occurred the new limit must be greater than the old one.

A.4   OS-3 File Name Conventions

There are three kinds of saved files: a) private, b) semi-public, or c) public. A file name consists of 1-8 characters with the first character determining the domain of the file. If the first character is a) alphabetic--the file is private and can be accessed only from this job user code; b) up-arrow,↑,--the file is semi-public and can be referenced by anyone having the job number; or c) an asterisk, *,--the file is public and can be referenced by anyone.

A.5   Miscellaneous Comments

1) Time Cuts

The user is normally allotted 60 seconds of computer time when he logs on the system. Frequently, during the course of execution this allotted time is used. The message TIME CUT will be printed and the user put into control mode. He can then use the TIME command described above to increase his time and resume execution by using the GO or MI command described above.

If after increasing the time the computer repeatedly gives a time cut then this usually indicates that the job number has insufficient funds, and the user must request more time from the computer center office.

2) Interrupt Button

Occassionally the user may wish to interrupt GROPE during the middle of execution. This can be accomplished by pushing the INTERRUPT button which will interrupt execution and then return control to control mode. By typing ⓂⒹ(CR) or ⒼⓄ(CR) control will be transferred back to GROPE.

APPENDIX B

APPENDIX B

SUMMARY OF FD&S AND DISPLAY COMMANDS

* - Denotes commands legal in FD&S

† - Denotes commands legal in Display phase

COMMAND NAME	DESCRIPTION

AUTOMATIC†

A(CR)

Causes automatic scaling to be used when the next CLEAR command is issued.

CARRIAGE RETURN†

(n Space Bars) m(CR)

Increments the varying parameters (n + 1) times and plots the next m members of each family of curves.
Examples: (CR) (3 blanks)2(CR)

CLEAR†

C,r,m(CR)

Clears the screen, draws axes, increments the varying parameters r times and plots the next m members of each family of curves.
Examples: C C,3,2 C,-3.1,4

DEL KEY†

Each of the following is <u>initiated</u> by depressing (DEL) and then positioning crosshairs.

1. Drawing Geometric Figures:

Alternatly depressing (DEL) and positioning crosshairs will draw a series of straight lines. To terminate the series depress Period or Space Bar. The latter will print varying parameter values.

2. Curve Labeling:

Depressing Space Bar will draw a line and print the varying parameter values. (See item 1 above for more elaborate labeling.)

3. Coordinates of a Point:

Type C when crosshairs have been positioned to the point whose coordinates are desired. Repositioning crosshairs and depressing (DEL) will draw a line and print the coordinates.

4. Message:

Type M after crosshairs have been positioned. Then type the desired message and terminate with (CR).

SUMMARY OF COMMANDS

DISPLAY*[†]

D,functl,funct2,...(CR)

Displays definitions of the named functions or displays all function definitions if no functions are specified. In Display phase position crosshairs to a point where definitions are to be printed and depress Space Bar.
Examples: D   D,X,Y   D,X"

FILE*[†]

F,name(CR)

Stores information on a file which is identified by "name". In FD&S, function definitions and current parameter values are stored. In Display phase, function definitions, current parameter values, plotting specifications, and range specifications, and range specifications are stored.
Examples: F,PHYS521   F,*PHYS211

GET*

G,name(CR)

Retrieves information which was stored by the FILE command.
Examples: G,PHYS521   G,*PHYS211

JUMP[†]

J,r,m(CR)

Increments the varying parameters r times and plots the next m members of each family of curves.
Examples: J   J,2,1   J,-4.5,3

KILL*[†]

K,functl,funct2,...(CR)

Destroys the named functions and their associated parameters. An unmodified KILL destroys all functions.
Examples: K   K,X,Y   K,X"

ON,OFF*[†]

ON,effect,modifier(CR)
OFF,effect,modifier(CR)
O,effect,modifier(CR)

Turns special effects ON or OFF. The effects present status is altered by O.

EFFECT	INITIAL STATUS	Examples
AXIS	ON	OFF,A
DASHES	OFF	ON,D,X,Y,X'
HEADER	ON	OFF,H
LABELS	ON	OFF,L
MARKS	OFF	ON,M
POWDER	OFF	ON,P,1000
QUARTER	ON	OFF,Q
TICS	ON	OFF,T
VECTORS	ON	OFF,V,X,Y,X'

SUMMARY OF COMMANDS

PARAMETERS*[†]

P(CR)

Lists the current values and increments
of all parameters.

PLOT*

P,functl,funct2,...;coord. sys.   (CR)

Specifies the names of functions to be
plotted and coord. system. The functions
named in a given PLOT command are drawn
on the same axis set.
Examples: P,X   P,Y,Z   P,Y(X)

QUIT*[†]

Q,phase,A(CR)

Quits the present phase and transfers
execution to the named phase. The
modifier A will incorporate automatic
scaling.
Examples: Q   Q,R   Q,D,A

RESET*[†]

R(CR)

Sets the current parameter values to
their initial values. If parameter values
have been altered in parameter specification
phase or by an UP-ARROW command then the
altered values are the initial values.

SPLIT*

S,type(CR)

Specifies that two independent axis sets
are to be used for plotting. SPLIT must
be preceeded and followed by a PLOT command.
All functions specified by a given PLOT
command will be plotted on the same axis
set. The modifier, type, may be vertical,
horizontal, or superimposed.
Examples: PLOT,X   PLOT,Y(X)
           S,H   S,V
           PLOT,Y   PLOT,X'

UP-ARROW*[†]

∧pnamel=valuel, pname2=value2 BY ivalue,...(CR)

Alters the values and/or increments of the
designated parameters.
Examples: ∧A = 1   ∧B = 2 BY 1
           ∧C = BY .5

APPENDIX C

APPENDIX C

EDITING, ERROR CORRECTION, AND ERROR MESSAGES

The editing capabilities of GROPE are designed to allow easy correction of typing errors. Before a line of input in any phase of GROPE is terminated with a (CR), the line may be corrected by using either the backslash "\" or the "@". The @ symbol deletes a whole line of input and allows a fresh start to be made. The backslash may be used to delete one or more characters. This symbol applied once deletes the last character; applied twice deletes the last two characters, etc.

As an example suppose in defining the function

$$y(t) = v_o t \sin\theta - \tfrac{1}{2}gt^2$$

the following is typed:

$$Y(T9 = VO*T \qquad .$$

After the T is typed, the user realizes that the fourth character, "9", should have been a ")". The @ symbol could be used to correct this error as follows:

$$Y(T9 = VO*T@ \; Y(T) = VO*T*SIN(TH)-G*T^2/2 \quad (CR) \qquad .$$

This error could also have been corrected by typing

$$Y(T9 = VO*T\backslash\backslash\backslash\backslash\backslash\backslash\backslash\backslash) = VO*T*SIN(TH)-G*T^2/2 \quad (CR) \qquad .$$

Notice that blanks count as a character and where necessary they must also be deleted with a backslash.

Editing differential equation sets is more involved since several lines of input are inner related. The backslash and the @ still retain the same meaning. That is, they can be used to edit a given member of the set as it is being typed. After a set has been entered GROPE will check to see if there are any errors. If GROPE discovers an error it immediately prints an appropriate error message and follows this by typing

ENTER NUMBER OF EQ. YOU WISH TO
RETYPE ("OR ALL"):

ERROR CORRECTION

The user may type numbers corresponding to the equations
which have errors. These numbers must be in ascending order and
must be separated by commas. GROPE responds by moving the cursor
to the left margin ringing the bell, and waits for the corrected
differential equations to be typed. After the first erroneous
equation is corrected, GROPE is ready to receive the next corrected
equation, and this process continues until the user indicates the
end of the set by depressing the Carriage Return. The following
example indicates how this procedure is used.

	Equation Number
>X"(T9 = -(X + Y + Z) (CR)	1
Y"(T) = -Y (CR)	2
Z"(T) = X*(Y + Z (CR)	3
(CR)	

UNMATCHED PARENTHESIS
ENTER NUMBER OF EQ. YOU WISH TO
RETYPE ("OR ALL"):   1,3   (CR)

```
    X"(T) = -(X + Y + Z)    (CR)
    Z"(T) = X*(Y + Z)    (CR)
    (CR)
```

GROPE will make the corrections by deleting the erroneous equations
and appending the corrected equations to the differential equation
set, so that if additional errors are made in correcting the equations,
the numbering sequence is altered. In the above example this means
that the new numbering sequence is

	Equation Number
Y"(T) = -Y	1
X"(T) = -(X + Y + Z)	2
Z"(T) = X*(Y + Z)	3

The asterisk, *, is another error recovery feature which
allows corrections of previous lines of numerical input. This
can be used anytime as series of numerical values are being requested
such as for parameter requests (see Figure 3, page   ) or for range
requests (see Figure 4, page   ). For example, suppose a parameter
A was to have a value 1 but was erroneously assigned a value 2 as
shown in the following sequence:

```
 > F(X)  = A + B*X + C*X^2    (CR)
     A = 2    (CR)
     B = 1 BY 1    (CR)
     C =
```

ERROR CORRECTION

If the error is recognized when a value for C is requested, then
the * can be used to correct the error as shown below.

```
>F(X) = A + B*X + C*X^2   (CR)
    A = 2    (CR)
    B = 1 BY 1   (CR)
    C = *    (CR)
    A = 1    (CR)
    B = 1 BY 1   (CR)
    C = 3    (CR)
>
```

In general typing *(CR) in response to any numerical request
will restart the sequence of requests.

Grope Error Messages

ERROR MESSAGE	DESCRIPTION
TOO MANY FUNCTIONS	The number of user defined functions has been exceeded. To continue use either the selective or total KILL.
TOO MANY INDEPENDENT VARIABLES	More than 15 different independent variable names have been used. Either use old names or use a total KILL.
MEMORY OVERFLOW, RESTART	The available storage for function code has been exceeded. To recover use the KILL command.
EXPRESSION TOO COMPLICATED	The expression contains more than 30 operators or contains too many nested parenthetical expressions. Simplify and re-enter the function.
TOO MANY DIFF. EQUATIONS	One should use the selective or total KILL to make room for the equation.
TOO MANY PARAMETERS	More than 30 parameters have been defined. To continue use the selective or total KILL.
TOO MANY CONSTANTS	A total KILL will allow new constants to be defined.
TOO MANY INITIAL CONDITIONS	The array containing initial conditions is full. A total KILL will allow more to be defined.
TOO MANY DIFF. EQUATIONS IN SYSTEM	A system of differential equations has been created in which the number of equations plus the sum of the orders is greater than 15. A simplified equation is the only solution.
ILLEGAL FUNCTION NAME	The function should be re-entered with a name which is composed of an alphabetic character followed by 0-3 alphanumeric characters.
FUNCTION ALREADY DEFINED	The function name has already been used. The old function must be KILLED or the new function re-entered with a new name.
ILLEGAL PARAMETER ELEMENT	A number or special character appears in the list of independent variables in the function definition. Re-enter the corrected function.

ERROR CORRECTION

ERROR MESSAGE	DESCRIPTION
TWO OPERATORS OR OPERANDS IN A ROW	Check that all operands are separated by an operator. If an undefined function is referenced with a parameter list, this error occurs. Re-enter the corrected function.
UNMATCHED PARENTHESIS	Re-enter the function definition with the number of right parenthesis equal to the number of left parenthesis.
SYSTEM ERROR	Record the function definition that gave the error and give it to one of the GROPE staff.
DIFFERENTIATING MORE THAN TWICE	The number of primes on a non-differential equation is more than two. Re-define the function. Note that a function being differentiated cannot involve a function to be differentiated or a differential function.
ILLEGAL COMMAND	This error occurs when an illegal command is given or if an illegal form of the command is given.
TOO MANY VARYING PARAMETERS	Varying parameters can be made fixed by making the increment zero. There is room for 10 varying parameters.
MODIFIER ERROR	The modifiers following the command are not in the correct format. Re-enter the command after checking the definition.
CANNOT KILL FUNCTION USED BY ANOTHER	The function referencing this function must be KILLED.

EVALUATION FORM
GROPE USER'S MANUAL

TO THE USER:

   The GROPE staff intend to revise this manual, and would appreciate your help in ascertaining the needs of the users. Would you please fill out this form and return it to the Computer Center. Thank you for your cooperation.

                    The Computer Center Publications
                    Committee

NAME _____(Optional)

DEPT _____   INSTITUTION _____

DATE _____   STUDENT /_7 INSTRUCTOR /_7 OTHER /_7

How would you classify yourself in regard to your computing background?

/_7   No knowledge of any computer language.

/_7   Simple knowledge of perhaps one language (FORTRAN, OSCAR, BASIC) and/or the ability to run a "canned" program on the machine.

/_7   6-18 months programming experience.

/_7   18 months or more of programming experience and ability to write fairly sophisticated FORTRAN programs.

How long did you use GROPE before you felt comfortable using it in class?

/_7   1 to 3 hours

/_7   3 to 12 hours

/_7   12 or more hours

Is this manual: ☐ too simple

☐ just right

☐ too difficult

☐ too repetitive

☐ too verbose

(Please circle the appropriate number):

		very poor	poor	average	good	excellent
1.	How does this manual meet your own personal needs?	1	2	3	4	5
2.	Clarity of material	1	2	3	4	5
3.	Organization of material	1	2	3	4	5
4.	Accuracy of information in manual	1	2	3	4	5
5.	Clarity of examples	1	2	3	4	5
6.	Relevancy of examples	1	2	3	4	5
7.	Adequacy of examples	1	2	3	4	5
8.	Usefulness of manual	1	2	3	4	5
9.	Personal opinion of manual	1	2	3	4	5

Additional comments:

Please include any suggestions you might have for increasing the
effectiveness of this manual or improving it, including eliminating
or adding topics or changing format.

What do you feel the major shortcomings of the manual are?

What type of examples would you like to see?

Please return to: Publications Committee
                      OSU Computer Center
                      Corvallis, Oregon 97331