ORNL/TM-7383

ornl

OAK RIDGE NATIONAL LABORATORY

UNION CARBIDE

A User's Guide to the TCSTKF Software Library: A Graphics Library for Emulation of TEKTRONIX Display Images in .TKF Disk Files

W. H. Gray
R. D. Burris

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

DISCLAIMER

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A04; Microfiche A01

ORNL/TM-7383
Dist. Category UC-32

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

A USER'S GUIDE TO THE TCSTKF SOFTWARE LIBRARY: A GRAPHICS LIBRARY FOR EMULATION OF TEKTRONIX DISPLAY IMAGES IN .TKF DISK FILES

W. H. Gray

R. D. Burris

Date Published - November 1980

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

CONTENTS

PREFACE

This work was sponsored by the Oak Ridge National Laboratory (ORNL) Fusion Energy Division (FED) Large Coil Program and was done in conjunction with system development on the GIFTS 4B[1] suite of structural analysis computer programs.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ABSTRACT

This report documents the user-level subroutines of the TCSTKF software library for the Oak Ridge National Laboratory (ORNL) Fusion Energy Division (FED) DECsystem-1Ø. The TCSTKF graphics library was written and is maintained so that large-production computer programs can access a small, efficient graphics library and produce device-independent graphics files. This library is presented as an alternative to the larger graphics software libraries, such as DISSPLA. The main external difference between this software and the TCSTEK[2,3] software library is that the TCSTKF software will create .TKF formatted intermediate plot data files, as well as producing display images on the screen of a Tektronix 4ØØØ series storage tube terminal. These intermediate plot data files can be subsequently postprocessed into report-quality images on a variety of other graphics devices at ORNL.

1. INTRODUCTION

In Oak Ridge National Laboratory's (ORNL's) Fusion Energy Division (FED), there are numerous 4000 series Tektronix storage tube terminals available for use. Tektronix supplies a software library, called the Terminal Control System (TCS), which translates user program data into vector-line images for graphics display on their terminals. This document describes a new software library that will display an image on the screen of a Tektronix terminal and simultaneously create a device-independent plot data file. This plot data file can be subsequently postprocessed to reconstruct report-quality drawings of the display images on any of the graphics devices available at ORNL.

The name of this new software library is TCSTKF. The first three characters mnemonically represent the Terminal Control System, and the last three represent Tektronix character (K) file. The intermediate (device-independent) .TKF plot data file created by the TCSTKF software library is simply the same ASCII decimal equivalent (ADE) character stream used by TCS to control the electron beam or write out characters on the screen of the terminal.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

2. HOW TO USE THIS DOCUMENT

This document has the following organization. Section 3 describes how to access the TCSTKF software. In Sect. 4 there are examples of how to postprocess the .TKF file. A sample program is presented in Sect. 5. These three sections should be read by anyone using this software library. The last several sections present a detailed discussion of the internal functions of the TCSTKF software library and options available to its user. It is not necessary to read these sections before use of the library. An advanced sample problem is presented in the last section.

A microfiched listing of the TCSTKF software source is attached to the back cover of this document. Throughout this document intimate familiarity is assumed with the basic TCS software (see Refs. 2 and 3). No changes to the external functions of the TCSTEK software library were made.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

3. HOW TO ACCESS THE TCSTKF SOFTWARE

The TCSTKF software library is available to the users of the FED DECsystem-1Ø by use of a command file located on device PUB: in a manner similar to all other graphics software. Such .CMD files cause the inclusion of all the necessary relocatable object modules (.REL files) without the user's having to concern himself with the location of the .REL files or the order in which they are loaded.

3.1 STANDARD LOADING AND EXECUTION

To execute a FORTRAN program, 'yourpg' in this example, and to use the TCSTKF software library, type

```
.EXECUTE yourpg,@PUB:TCSTKF
```

Two inflexible requirements must be met by the user's FORTRAN code.

(1) There must be a call to subroutine INITT(Ø), the standard way to initialize the TCS software. (A call to subroutine INITT with an argument of zero forces TCS to determine automatically the TTY transmission speed in characters per second.) This subroutine call should be made only once and must be the first TCS subroutine call in the user's program.

(2) There must be a call to subroutine FINITT(IX,IY), the standard way to terminate the TCS software. (The arguments IX and IY represent the final position of the alphanumeric cursor in absolute screen coordinates.) This subroutine call should be made only once and must be the last TCS subroutine call in the user's program so that the last internal character buffer is written to the .TKF file.

Assuming both of the above requirements have been satisfied and the user's program executes successfully, then an intermediate .TKF plot data file will have been created in the user's disk area. This file is named FOR24.TKF and it may be postprocessed in a variety of ways. (Section 6.3.1 discusses how to create an intermediate .TKF plot data

file with an different name.)

3.2 OVERLAYED LOADING

As will be discussed further in Sect. 7., several TCS subroutines have been modified to permit automatic generation of .TKF plot data files. Therefore, the relocatable object modules containing these entry points must be loaded first. With nonoverlayed loading this is accomplished by the indirect command file PUB:TCSTKF.CMD, which contains the following two directives to program LINK:

```
REL:TCSTKF.REL

REL:TCSTEK.REL/SEARCH
```

With overlayed loading, a set of directives to program LINK that will include the TCSTKF software in the appropriate overlay segment is

```
REL:TCSTKF.REL/INCLUDE:(RESET,DATTKF,TKFDO),

REL:TCSTEK.REL/USERLIBRARY
```

The module RESET contains an entry point named RESET. The rest of the entry points discussed in Sects. 6 and 7 are contained in module TKFDO except for DATTKF, DASTKF, and DAFTKF.

4. POSTPROCESSING INTERMEDIATE .TKF PLOT DATA FILES

Once an intermediate .TKF plot data file has been created in the user's disk area, the PLOT command can be used to create display images on several types of graphics devices at ORNL. The following examples demonstrate how to postprocess a .TKF file and produce plots on the Versatec printer/plotter connected to the FED DECsystem-1Ø:

```
.PLOT VER:=FOR24.TKF
```

The same function can be accomplished on the Versatec printer/plotter connected to the FED PDP-11/45 (affectionately called the RAT) by typing

```
.PLOT V45:=FOR24.TKF
```

This postprocessing step is preferable to the previous one because the vector-to-raster conversion (a CPU intensive process) necessary to produce a Versatec image is performed on the PDP-11/45 rather than on the DECsystem-1Ø. Furthermore, the Versatec printer/plotter queue is frequently backlogged on the DECsystem-1Ø; therefore, plotting on the PDP-11/45 can result in a faster turnaround.

There are other postprocessing alternatives. For a complete description of the capabilities of the PLOT command, see Ref. 4.

To obtain a copy of the file BEST.TKF on the FR80 at K-25, a special command is available. By typing

```
.FICHE BEST.TKF
```

a request will be generated to a locally written postprocessor to spool the .TKF file to a special area. No further action is required by the user. Periodically (at present, each day at noon), the system operator creates a tape that contains all the spooled FR80 requests. This tape is sent to K-25 for postprocessing. Usually, the resulting microfiche or microfilm output will be returned to the user within 48 hours.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

5. A SIMPLE EXAMPLE

This section demonstrates a sample execution of a FORTRAN program to draw a display image on a Tektronix, create a .TKF plot data file, and subsequently postprocess the .TKF file into an image on the Versatec printer/plotter connected to the FED PDP-11/45.

5.1 TEST PROGRAM

The following program uses TCS to draw an exponentially decaying cosine function. This program is internally commented and as such is self-explanatory. A listing follows.

```
      PROGRAM RNGTCS
C
      COMMON /DRWPAR/ ISW(4),RVW(4),CON
C
      DATA ISW/2ØØ,4ØØ,2ØØ,4ØØ/
      DATA RVW/Ø.,2Ø.,-1.,1./
      DATA CON/Ø.2/
C
C  INITIALIZE THE TCS SOFTWARE LIBRARY
      CALL INITT (Ø)
C
C  SET UP A SCREEN AND VIEWING WINDOW
      CALL SWINDO (ISW(1),ISW(2),ISW(3),ISW(4))
      CALL DWINDO (RVW(1),RVW(2),RVW(3),RVW(4))
C
C  DRAW RINGER
C
C  FIRST CALCULATE AND MOVE TO BEGINNING POSITION
      Y = COS(RVW(1)) * EXP(CON * -RVW(1))
      CALL MOVEA (RVW(1),Y)
C
C  NOW LOOP THROUGH ALL THE POINTS
      DO 1ØØ T=RVW(1),RVW(2),.Ø5
      Y = COS(T) * EXP(CON * -T)
      CALL DRAWA (T,Y)
1ØØ   CONTINUE
C
C  DRAW A GRID
      DO 2ØØ X = RVW(1),RVW(2),2.
      CALL MOVEA (X,RVW(3))
      CALL DRAWA (X,RVW(4))
2ØØ   CONTINUE
      DO 21Ø Y = RVW(3),RVW(4),.2
      CALL MOVEA (RVW(1),Y)
```

```
        CALL DRAWA (RVW(2),Y)
21Ø     CONTINUE
C
C  TERMINATE TCS
        CALL FINITT (Ø,78Ø)
        STOP
        END
```

5.2 SAMPLE EXECUTION

To execute the above program, the user first copies it into his disk area

```
.COPY =DOC:RNGTCS.FOR
```

then executes it by typing

```
.EXECUTE RNGTCS.FOR,@PUB:TCSTKF
```

Assuming a successful execution, the file FOR24.TKF will have been created in the user's disk area. Postprocessing on the FED PDP-11/45 yields an image that should be similar to the curve in Fig. 1.

5.3 SAMPLE BATCH EXECUTION

The following batch control file demonstrates an actual step-by-step execution of this test program. (Notice that although the program was executed in the batch mode, no ADE characters clutter the log file, as would have been the case with the TCSTEK software library. The TCSTKF software library determines that the program is being executed in the batch mode and therefore does not send any characters to the .LOG file.)

```
1Ø:ØØ:Ø5 BAJOB   BATCON version 13(1Ø71) running RNGTCS sequence
                          4351 in stream 1 for A31 DISSPLA
1Ø:ØØ:Ø5 BAFIL   Input from DSKAØ:RNGTCS.CTL[1Ø,1Ø1]
1Ø:ØØ:Ø5 BAFIL   Output to  DSKAØ:RNGTCS.LOG[1Ø,1Ø1]
1Ø:ØØ:Ø5 BASUM   Job parameters
                 Time:ØØ:Ø5:ØØ     Unique:YES        Restart:NO

1Ø:ØØ:Ø5 MONTR
```

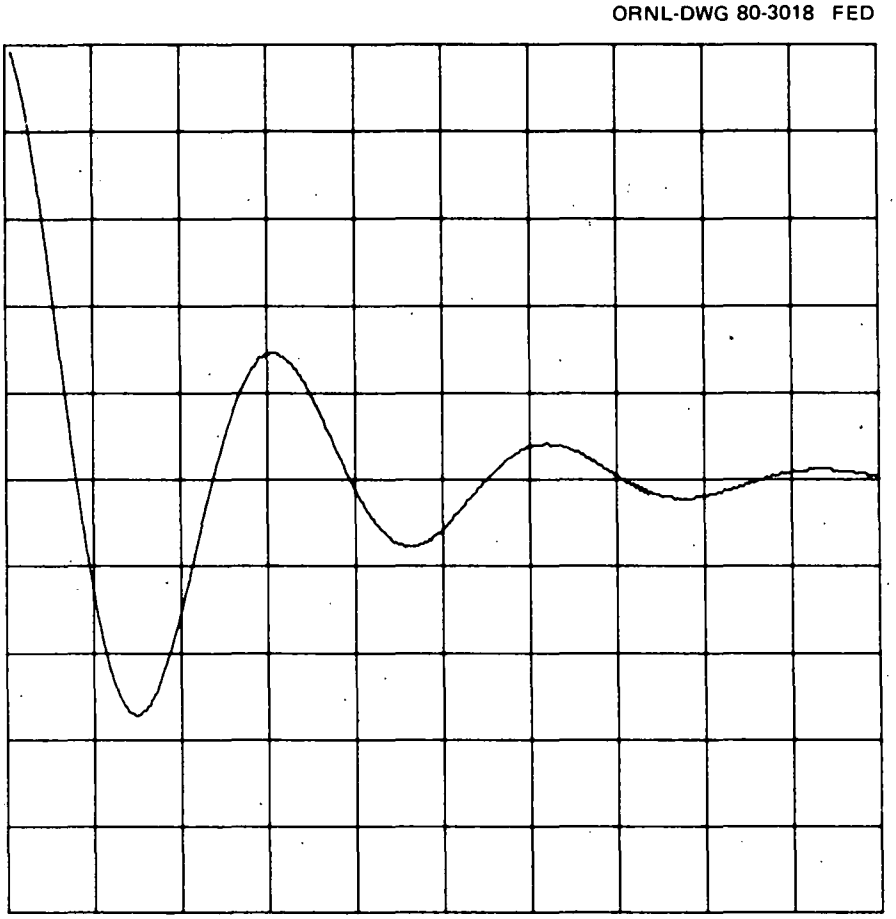

Fig. 1. The display image for the sample program in Sect. 5.

```
1Ø:ØØ:Ø5 MONTR   .LOGIN 1Ø/1Ø1
1Ø:ØØ:Ø5 USER    JOB 37 ORNL Fusion Energy 6Ø3a TTY162
1Ø:ØØ:Ø8 USER    [LGNJSP Other jobs same PPN:46]
1Ø:ØØ:Ø8 USER    1ØØØ     Ø4-Dec-79         Tue
1Ø:ØØ:1Ø MONTR
1Ø:ØØ:1Ø MONTR   .LOCATE 1
1Ø:ØØ:11 MONTR   NODE     FEDUSC(1) ORNL Fusion Energy 6Ø3a Ø9-Ø5-79
                          Located
1Ø:ØØ:11 MONTR   ..EXECUTE RNGTCS.FOR,PUB:TCSTKF
1Ø:ØØ:14 USER    LINK:    Loading
1Ø:ØØ:51 USER    [LNKXCT RNGTCS Execution]
1Ø:Ø2:28 USER
1Ø:Ø2:28 USER    STOP
1Ø:Ø2:28 USER
1Ø:Ø2:28 USER    END OF EXECUTION
1Ø:Ø2:28 USER    CPU TIME: Ø.22    ELAPSED TIME: 1:37.45
1Ø:Ø2:28 MONTR   EXIT
1Ø:Ø2:28 MONTR
1Ø:Ø2:28 MONTR   ..DIRECT FOR24.TKF
1Ø:Ø2:3Ø USER
1Ø:Ø2:3Ø USER    FOR24    TKF     2  <Ø55>    4-Dec-79 DSKA: [1Ø,1Ø1]
1Ø:Ø2:3Ø MONTR
1Ø:Ø2:3Ø MONTR
1Ø:Ø2:3Ø MONTR   .KJOB DSKAØ:RNGTCS.LOG=/W/B/Z:4/VR:1Ø/VS:4351
                         /VL:2ØØ/VP:1Ø/VD:P
1Ø:Ø4:11 K-QUE   TOTAL OF 2 BLOCKS IN 1 FILE IN LPTS1 REQUEST
1Ø:Ø4:12 KJOB    Other jobs same PPN
1Ø:Ø5:5Ø LGOUT   Job 37, User [1Ø,1Ø1]  Logged off TTY162   1ØØ5  4-Dec-79
1Ø:Ø5:5Ø LGOUT   Another job still logged in under [1Ø,1Ø1]
1Ø:Ø5:5Ø LGOUT   Runtime 2.43 s
1Ø:44:Ø7 LPMSG    LPTSPL Version 6(3441Ì)  Running on ĒPĪLØ1Ø
```

The next sections of this report are intended for the advanced user who would like to access some of the unique features of the TCSTKF software library.

6. USER-LEVEL TCSTKF SOFTWARE

The TCSTKF software library can be used without modification to a user's FORTRAN program to produce .TKF formatted plot data files automatically. With modification to a user's FORTRAN program, full control of the graphics device display is available. The Tektronix display or the .TKF file output can be selectively turned off or on. Sections 6.1 and 6.2 discuss how to control the output display on a Tektronix or in a .TKF file by using the ON/OFF switch capabilities of the TCSTKF software.

Optionally, a mode of execution is available that will allow the user to decide after a display image has been drawn on a Tektronix screen whether or not the Tektronix display image should be saved. Section 6.3 discusses how a temporary .TKF file can be created containing the current Tektronix display image from which a permanent .TKF file can be updated by transferring the display image from the temporary to the permanent .TKF file. Of course, transferring the display image to the permanent .TKF file implies that the decision to do so will be made after the display image has been drawn, thus allowing an interactive capability. Section 6.4 presents how to display a previously created .TKF file while executing a FORTRAN program. Section 6.5 discusses how to merge an ASCII file into a .TKF data file so that the resulting .TKF file will contain both plot and print output.

Throughout Sects. 6 and 7 several FORTRAN logical function subprograms are described. A logical function subprogram returns a true or false value and can be used to take action depending upon the success (true) or failure (false) of a specific operation. However, all DECsystem-1Ø FORTRAN function subprograms, including logical ones, can be called as regular subroutine subprograms if the user wishes to ignore the functional value returned by the subprogram. For example, Sect. 6.4.1 describes logical function TKFTEK, which displays an existing .TKF plot data file on the user's Tektronix terminal. It can be called by either

```
IF(TKFTEK('DSK:PLOT.ME')) 100,200
```

or

```
CALL TKFTEK('DSK:PLOT.ME')
```

The first method is preferred because the file, PLOT.ME, might not be in the user's disk area. With the second method information about the presence or absence of the .TKF file is not available.

6.1 GRAPHICS DISPLAY DEVICE CONTROL

These subroutines can be used to control the graphics display either on the screen of a Tektronix storage tube or in an intermediate .TKF plot data file. The first three characters of the subroutine name are either TEK or TKF, depending upon which graphics display the user wishes to control. The last characters are either ON or OFF depending upon what a user wishes to do with his graphics display.

In all cases calling one of these subroutines will first cause the internal TCS output buffer to be sent to all graphics devices active before the call was made. After this I/O has been performed, the appropriate switch is set to control the requested graphics display. The default status of both the TEK and TKF ON/OFF switches is on. However, if a user is executing his program under the batch controller, the default setting for the TEK ON/OFF switch is off to prevent cluttering the batch .LOG file with ADE characters.

6.1.1 Subroutine TEKON

Subroutine TEKON'S function is first to clear the internal TCS buffer, if one is present, and then to turn on a switch that permits subsequent graphics to be transmitted to the user's Tektronix terminal. (This is the default status of the TCSTKF software library.)

Calling sequence:

CALL TEKON

Argument list: None

Restrictions:

The TCSTKF software will not allow the TEK ON/OFF switch to be turned on while the program is being executed from batch.

6.1.2 Subroutine TEKOFF

Subroutine TEKOFF's function is first to clear the internal TCS buffer, if one is present, and then to turn off a switch that permits graphics to be transmitted to the user's Tektronix terminal.

Calling sequence:

CALL TEKOFF

Argument list: None

6.1.3 Subroutine TKFON

Subroutine TKFON's function is first to clear the internal TCS buffer, if one is present, and then to turn on a switch that permits graphics to be transmitted to the currently open intermediate .TKF plot data file. (This is the default status of the TCSTKF software library.)

Calling sequence:

CALL TKFON

Argument list: None

6.1.4 Subroutine TKFOFF

Subroutine TKFOFF's function is first to clear the internal TCS buffer, if one is present, and then to turn off a switch that permits graphics to be transmitted to the currently open intermediate .TKF plot data file.

Calling sequence:

CALL TKFOFF

Argument list: None

6.2 GRAPHICS DISPLAY DEVICE STATUS

A facility has been provided to allow the user to examine the status of the TEK and TKF ON/OFF switches. It was determined that interactive programs need to be able to take action depending upon whether a specific graphics display has been turned on or off. For example, if the Tektronix terminal is currently being used as an output device, then a user generally would like to be able to pause to view the display image between drawing the display image and erasing the page in preparation for the next display image. If the Tektronix terminal is not being used as an output device, then a pause is generally unnecessary. The following two subprograms examine the TEK and TKF ON/OFF switches and return the corresponding logical value depending upon whether or not the display is in use.

6.2.1 Logical Function GTSTEK

Logical function GTSTEK returns the logical value of the Tektronix ON/OFF switch. If the value returned for this function is true, then graphics output can be transmitted to the user's terminal. If the value returned for this function is false, then graphics output to the user's terminal is suppressed. Tektronix ON/OFF switch is true by default, except when the user is logged into a batch job. To control the Tektronix ON/OFF switch, see the documentation on the TEKON and TEKOFF subroutines in the previous section.

Calling sequence:

IF (GTSTEK(Ø)) tekon, tekoff

where tekon and tekoff symbolically represent statement labels to which the user's program can branch depending upon the logical value returned by subprogram GTSTEK.

Argument list: A dummy argument is required.

Example:

```
        LOGICAL GTSTEK
        .
        .
        .
C
C  CHECK FOR TEKTRONIX CAPABILITY
        IF (.NOT. GTSTEK(Ø)) GO TO 1ØØ
C
C  HERE IF ON, PAUSE TO ASSIMILATE IMAGE
        CALL ADEIN(1,JUNK)
C
C  HERE IF NOT ON
1ØØ     CONTINUE
        .
        .
        .
```

6.2.2 Logical Function GTSTKF

Logical function GTSTKF returns the logical value of the intermediate .TKF plot file capability. If the value returned for this function is true, then graphics output can be transmitted to the user's .TKF file. If the value returned for this function is false, then graphics output to the user's .TKF file is suppressed. The intermediate .TKF plot data capability is true by default. To control the intermediate .TKF plot data file ON/OFF switch, see the documentation on the TKFON and TKFOFF subroutines in the previous section.

Calling sequence:

```
IF (GTSTKF(Ø)) tkfon, tkfoff
```

where tkfon and tkfoff symbolically represent statement labels to which the user's program can branch depending upon the logical value returned by subprogram GTSTKF.

Argument list: A dummy argument is required.

Example:

```
        LOGICAL GTSTKF
        .
        .
        .
C
C  CHECK FOR THE .TKF FILE OPTION
```

```
      IF (GTSTKF(Ø)) GO TO 1ØØ
C
C  HERE IF NOT ON
      CALL TKFON                    !TURN IT ON.
C
C  HERE IF ON
1ØØ   CONTINUE
      .
      .
      .
```

6.3 SOFTWARE FOR THE SELECTIVE SAVING OF .TKF PLOTS

For the interactive user another option is provided by the TCSTKF software library. Instead of having to place all display images into the .TKF file, control is possible for selectively deciding, after a display image has been drawn, whether or not to transfer the display image to the permanent .TKF file. The three subroutines described in this section implement this control.

The first subprogram, INITKF, is used to initialize the TCSTKF software library into this mode of execution. A call to this subprogram must replace the call to subroutine INITT (the TCS initialization subroutine). At the logical ending of a display image, a call to subroutine FINPAG should be made. This subroutine will ask the user whether or not the temporary .TKF file (an exact duplicate of the ADE characters that created the display image of the user's terminal) is to be transferred into the permanent .TKF file. The last subroutine described in this section provides user control over the position and composition of the prompt that queries the user whether or not to transfer the display image.

6.3.1 Logical Function INITKF

The full capabilities of the TCSTKF software library can be used by initializing the subprogram package with a call to logical function INITKF instead of the TCS subroutine INITT (among other functions, subroutine INITKF calls TCS subroutine INITT for the user). There are two arguments: an intermediate .TKF plot data file specification and a

logical variable to set the prompt switch. The file specification is the location and name of the .TKF file to be created. The prompt switch argument should be true if the user wishes to activate subprogram FINPAG for individual .TKF plot-saving option. This facility allows the user to decide after a display image is plotted on the Tektronix storage tube (but during execution of his program) whether or not to place the display image in the permanent .TKF file for subsequent postprocessing. If no interactive individual display image saving is required, then the value of the prompt switch argument should be false.

Calling sequence:

IF (INITKF('file specification', prompt)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram INITKF.

Argument list:

'file specification' is a standard DECsystem-10 file specification that can consist of a device, file name and extension, project-programmer number, and subfile directory. The default file specification is DSK:FOR24.TKF[-]. [The file specification must be passed as either an ASCIZ character string or an ASCII character string ending with a dollar sign ($) character. Blank characters are ignored. An ASCIZ character string is an ASCII character string that is guaranteed to have a null character following it.]

prompt is a logical variable indicating whether or not (true or false) the user wishes to be individually prompted during execution to save his display images. (If false, all display images will be sent to the .TKF file in accordance with the value of the TKF ON/OFF switch.)

Example:

```
        LOGICAL INITKF
        .
        .
        .
C
C
C  INITIALIZE THE TCSTKF AND TCS SOFTWARE
        IF (INITKF('ISXA:STORE.TKF[,,GUD,DATA,HERE]',.TRUE.)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('CALL TO INITKF FAILED.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

6.3.2 Logical Function FINPAG

The TCS software does not provide for a pause between the finish of a display image, the erasure of the screen, and the commencement of a new display image. Logical function FINPAG allows the user time to assimilate the information in a display image by pausing until the user strikes any character on the terminal keyboard (unless he has previously turned the Tektronix terminal off by calling subprogram TEKOFF or the program is being executed by the batch controller).

If the logical function INITKF has been called to set up prompting for intermediate .TKF file selective display image saving, then a call to logical function FINPAG will query the user with a prompt and wait until a single character is typed from the terminal. A response of N (upper or lower case) will cause the temporary .TKF file to be reinitialized without transfer to the permanent .TKF file. Any response other than N will be interpreted as positive, causing the temporary .TKF file to be transferred automatically to the permanent .TKF file. After the successful transfer of the display image data, the screen will be erased and program control will be returned to the calling subprogram.

Naturally, if the TKF ON/OFF switch has been turned off, then no transfer will be performed, regardless of a positive response to the prompt. Also, if the TEK ON/OFF switch has been turned off, then no prompt will be made and the temporary .TKF file (if present) will be transferred to the permanent .TKF file.

Calling sequence:

IF (FINPAG(IADE)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram FINPAG.

Argument list:

IADE — is the ADE character struck by the user in response to the prompt concerning the .TKF file transfer. IADE will be set to zero if no response was required.

Example:

```
        LOGICAL FINPAG
        .
        .
        .
C
C  TERMINATE CURRENT PLOTTING PAGE AND
C  TRANSFER IF SO DESIRED.
        IF (FINPAG(JUNK)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('CALL TO FINPAG FAILED.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

6.3.3 Logical Function TEKPMT

Logical function TEKPMT may be used to change the statement displayed during execution when a decision is made as to whether or not the display image should be transferred from the temporary .TKF file into the permanent .TKF file. Optionally, subprogram TEKPMT may be used

to turn off user interaction completely after processing a display image so that future calls to subroutine FINPAG do not request the user to decide about display image transfers. This subprogram should not be called unless the TCSTKF software was initialized by calling subprogram INITKF.

Calling sequence for three-argument call:

IF (TEKPMT(IX,IY,'yourpmt')) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram TEKPMT.

Argument list:

IX	is the X absolute Tektronix storage tube coordinate of the lower left-hand corner of the prompt character string. If the value of IX is less than zero, a switch is set to suppress prompting for display image transferring. (All display images will be transferred if IX is less than zero.)
IY	is the Y absolute Tektronix storage tube coordinate of the lower left-hand corner of the prompt character string.
'yourpmt'	is the text string to be written as a prompt to ask the question whether or not to save the display image. [This text string must be less than 80 characters long and must be terminated with either a null or a dollar sign ($) character. If not, the call is considered a failure.]

Defaults:

The position and composition of the prompt, unless otherwise overwritten by the user, are (0,785) absolute Tektronix screen coordinates and "Save this display image [Y(es) or N(o)]?", respectively.

Example:

```
      LOGICAL TEKPMT
      .
      .
      .
C
```

```
C  CHANGE THE PROMPT
       IF (TEKPMT(Ø,Ø,'HOW ABOUT IT BUB?$')) GO TO 1ØØ
C
C  HERE IF FAILURE
       TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ   FORMAT('CALL TO TEKPMT FAILED.  JOB STOPPED')
       STOP
C
C  HERE IF SUCCESSFUL
1ØØ    CONTINUE
       .
       .
       .
```

Calling sequence for two-argument call:

This form of a call to subroutine TEKPMT changes the location of the prompt but not its contents.

IF (TEKPMT(IX,IY)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram TEKPMT.

Argument list:

IX	is the X absolute Tektronix storage tube coordinate of the lower left-hand corner of the prompt character string. If the value of IX is less than zero, a switch is set to suppress prompting for display image transferring.
IY	is the Y absolute Tektronix storage tube coordinate of the lower left-hand corner of the prompt character string.

Example:

```
       LOGICAL TEKPMT
       .
       .
       .
C
C  CHANGE THE PROMPT'S LOCATION, ONLY.
       IF (TEKPMT(Ø,Ø)) GO TO 1ØØ
C
C  HERE IF FAILURE
       TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ   FORMAT('CALL TO TEKPMT FAILED.  JOB STOPPED')
       STOP
```

```
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

6.4 FORTRAN-CALLABLE SOFTWARE FOR DISPLAY OF .TKF FILES — LOGICAL FUNCTION TKFTEK

Certain users have found it convenient to pregenerate .TKF files containing complex grid labels or other display images that can then be displayed or overlayed onto a new display image. The subprogram TKFTEK provides this capability for the TCSTKF software and is described in the following section.

Logical function TKFTEK can be used to display an existing intermediate .TKF file on the user's Tektronix terminal. Optionally, this file can be simultaneously copied into another intermediate .TKF file. The subroutines TEKON, TEKOFF, TKFON, and TKFOFF can be used to control the output of the intermediate .TKF file in the same manner as they are used to control user-program Tektronix character output.

This logical function will permit overlaying several display images on the same display image. Remember — any new page instructions contained within an input file will be transferred to the output file and to the Tektronix screen if their respective switches are both turned on.

Calling sequence:

IF (TKFTEK('file specification')) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram TKFTEK.

Argument list:

'file specification' is a standard DECsystem-1Ø file specification that can consist of a device, file name and extension, project-programmer number, and subfile directory. The default file specification is DSK:SAVPLT.TKF[-]. [The file

specification must be passed as either an ASCIZ character string or an ASCII character string ending with a dollar sign ($) character. Blank characters are ignored.]

Example:

```
        LOGICAL TKFTEK
        COMMON /JUNK/ FILSPC(1Ø) , IJUNK
        .
        .
        .
C
C  READ A FILE NAME
        ACCEPT 4ØØØ, FILSPC
4ØØØ    FORMAT(1ØA5)
C
C  MAKE SURE STRING IS ASCIZ
        IJUNK = Ø
C
C  MAKE SURE THE TEKTRONIX SCREEN IS ON.
        CALL TEKON
C
C  GO READ AND PLOT THE FILE
        IF (TKFTEK(FILSPC)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('?ERROR PLOTTING .TKF FILE.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

6.5 FORTRAN-CALLABLE SOFTWARE FOR MERGING PRINT OUTPUT INTO .TKF FILES

Three subprograms are presented in this section that easily permit the inclusion of ASCII print data files directly into a .TKF file. Essentially, all three subprograms perform the same function. However, they differ with respect to the amount of control the user wishes to have over the ASCII file that is to be merged and the resulting character size for the merged, printed characters. The great advantage to using these subprograms is that the user is free to create the

included ASCII files by whatever means available instead of having to rely solely upon the limited text-string-handling capabilities of the TCS software.

6.5.1 Logical Function DATTKF

Logical function DATTKF can be used to merge the ASCII print file FOR24.DAT into a previously opened .TKF file. The steps that subprogram DATTKF uses to perform this transfer are described below.

(1) First, FORTRAN logical unit number 24 is closed.

(2) The current TCS and TKF status variables are saved.

(3) Calls to the subroutines TEKOFF and TKFON are made.

(4) The TCS subroutine ANMODE is called to place the .TKF file into the alphanumeric mode.

(5) If a new page was requested, then calls to the TCS subroutine NEWPAG and HOME are made. If not, these calls are bypassed.

(6) Subprogram TKFTEK is used to transfer the file FOR24.DAT into the currently open .TKF file.

(7) The file FOR24.DAT is deleted. This feature permits FORTRAN logical unit number 24 to be used over and over again without the user's having to concern himself about the position of the next write instruction into this file. Because this subprogram deletes FOR24.DAT, FOROTS will always OPEN a new FOR24.DAT file, initialized for sequential output, the next time FORTRAN logical unit number 24 is referenced.

(8) The final step is to restore the TCS and TKF status variables.

For a discussion of how to change the number of characters per line and number of lines per page that can be represented in a .TKF file, see the next section. This subprogram yields 74 characters per line and 35 lines per page before a TKF postprocessor begins to wrap around the printed characters. Remember — these characters are written into the .TKF file, not drawn. Therefore, the resulting .TKF file size is kept to a minimum.

Calling sequence:

IF (DATTKF(PAGE)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram DATTKF.

Argument list:

PAGE is a logical variable used to indicate whether (true) or not (false) the user requires a new page instruction to be included in his .TKF file before the requested file is merged.

Example:

```
      LOGICAL DATTKF
      DIMENSION ARRAY(15Ø)
      .
      .
      .
C
C  INITIALIZE THE TCSTKF SOFTWARE
      CALL INITT(Ø)
      .
      .
      .
C
C  BUILD AN ASCII TEXT FILE
      WRITE(24,1ØØØ) ARRAY
1ØØØ  FORMAT('FOR THE LAST PLOT THE ARRAY WAS',/
     1  25(6E12.5/))
C
C  MERGE THE ASCII FILE INTO THE .TKF FILE
      IF (DATTKF(.TRUE.)) GO TO 1ØØ
C
C  HERE IF FAILURE
      TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ  FORMAT('?ERROR MERGING .TKF FILE.  JOB STOPPED')
      STOP
C
C  HERE IF SUCCESSFUL
1ØØ   CONTINUE
      .
      .
      .
```

6.5.2 Logical Function DASTKF

This logical function will merge ASCII text files into .TKF files. The only difference between logical functions DATTKF and DASTKF is that with the latter the character size for the merged ASCII characters can be specified, thus emulating the variable-character-size capability of a Tektronix 4Ø14/15 series storage tube terminal.

Calling sequence:

IF (DASTKF(PAGE,ICHAR)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram DASTKF.

Argument list:

PAGE is a logical variable used to indicate whether (true) or not (false) the user requires a new page instruction to be included in his .TKF file before the requested file is merged.

ICHAR is an integer variable that contains the Tektronix character size with which the user wishes his file to be merged (see Table 1).

Table 1. Tektronix character sizes

ICHAR	Characters per line	Lines per page
1	74	35
2	81	38
3	121	58
4	133	65

Example:

```
        LOGICAL DASTKF
        DIMENSION ARRAY(15Ø)
        .
        .
        .
C
C  INITIALIZE THE TCSTKF SOFTWARE
        CALL INITT(Ø)
        .
        .
        .
C
C  BUILD AN ASCII TEXT FILE
        WRITE(24,1ØØØ) ARRAY
1ØØØ    FORMAT('FOR THE LAST PLOT THE ARRAY WAS',/
     1  25(6E12.5/))
C
C  MERGE THE ASCII FILE INTO THE .TKF FILE
        IF (DASTKF(.TRUE.,3)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('?ERROR MERGING .TKF FILE.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

6.5.3 Logical Function DAFTKF

This logical function differs from DATTKF and DASTKF with regard to specification of the ASCII file to be merged. With the previous two logical functions it was implicitly assumed that the file FOR24.DAT was to be merged. This logical function determines the file to be merged from the arguments to the subprogram.

Calling sequence:

IF (DAFTKF(IUNIT,DIALOG,DELETE,PAGE,ICHAR)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram DAFTKF.

Argument list:

IUNIT is an integer variable representing the FORTRAN logical unit number to be used during the transfer. It can be 24.

DIALOG is a standard DECsystem-1Ø file specification that can consist of a device, file name and extension, project-programmer number, and subfile directory. The default file specification is DSK:FOR24.DAT[-]. [The file specification must be passed as either an ASCIZ character string or an ASCII character string ending with a dollar sign ($) character. Blank characters are ignored.]

DELETE is a logical variable that indicates whether (true) or not (false) to delete the file described by DIALOG after the transfer is complete.

PAGE is a logical variable used to indicate whether (true) or not (false) the user requires a new page instruction to be included in his .TKF file before the requested file is merged.

ICHAR is an integer variable that contains the Tektronix character size with which the user wishes his file to be merged (see Table 1).

Example:

```
      LOGICAL DAFTKF
      DIMENSION ARRAY(15Ø)
      .
      .
      .
C
C  INITIALIZE THE TCSTKF SOFTWARE
      CALL INITT(Ø)
      .
      .
      .
C
C  BUILD AN ASCII TEXT FILE
      WRITE(1Ø,1ØØØ) ARRAY
1ØØØ  FORMAT('FOR THE LAST PLOT THE ARRAY WAS',/
     1  25(6E12.5/))
C
C  MERGE THE ASCII FILE INTO THE .TKF FILE
      IF (DAFTKF(1Ø,'FOR1Ø.DAT',.FALSE.,.TRUE.,1)) GO TO 1ØØ
```

```
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('?ERROR MERGING .TKF FILE.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

7. MODIFIED TCS SOFTWARE AND MISCELLANEOUS SUBPROGRAMS

This section describes several modified TCS subroutines as well as several miscellaneous subprograms used by the TCSTKF software library. It is not intended for the casual user. Modifications required to implement this software into the TCS suite of subroutines involved changing the initialization, termination, output, and timing subroutines of TCSTEK. Other subroutines in this section deal with requesting TCS to send its internal character buffer to a graphics device. Also, a subprogram is presented that initializes the TEK ON/OFF switch to on except when the batch controller (BATCON) is controlling the user's job (in which case the TEK ON/OFF switch is turned off).

7.1 MODIFICATIONS TO TCS INITIALIZATION AND TERMINATION SOFTWARE

Two subroutines of the TCSTEK software library have been augmented to allow automatic creation of .TKF plot data files without requiring modifications to a user's FORTRAN source code. These are subroutines RESET and FINITT. Subroutine RESET is called by the TCS subroutine INITT to reset several of the important variables in common block /TKTRNX/. Subroutine FINITT is called to terminate a Tektronix graphics display.

The TCS subroutine RESET has been augmented to include a call to logical function OPNTKF (see Sect. 7.2.1). This call is made the first time subroutine RESET is called, usually by subroutine INITT. Subsequent calls to subroutine RESET do not generate calls to logical function OPNTKF. Therefore, a user can interchange the TCSTKF and the TCSTEK software library without changing a single line of FORTRAN code. (The default file name used by subroutine RESET when it calls logical function OPNTKF is FOR24.TKF. This name was chosen for consistency with other graphics packages on the FED DECsystem-1Ø, but this does not imply that FORTRAN logical unit number 24 is used by the TCSTKF software library.)

The TCS subroutine FINITT has been augmented to make a call to logical function CLSTKF (see Sect. 7.2.2) to close the .TKF file. Therefore, assuming the user has included both a call to subroutine INITT and FINITT in his source code, he can switch between the TCSTEK and TCSTKF software libraries without any alterations to his FORTRAN source code. A .TKF file either will or will not be created depending upon which software library the user loads with his program.

7.2 THE .TKF FILE-OPENING AND CLOSING SOFTWARE

So that the TCSTKF software library does not interfere with a user's FORTRAN logical unit number assignments, the FOROTS subroutines ALCHN. and DECHN.[5] are used to obtain and release a disk I/O channel number, respectively. It is not necessary to preallocate a FORTRAN logical unit number (such as 24, used by the DISSPLA software)[6] when using the TCSTKF library. This will be done automatically — just as FOROTS does for a user's FORTRAN source code.

7.2.1 Logical Function OPNTKF

The function of subroutine OPNTKF is to open an intermediate .TKF plot data file. If an intermediate .TKF plot data file is already open, then a warning message is printed out and the error branch is taken.

Calling sequence:

IF (OPNTKF('file specification')) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram OPNTKF.

Argument list:

'file specification' is a standard DECsystem-10 file specification that can consist of a device, file name and extension, project-programmer number, subfile directory, and one switch. The default file specification is DSK:FOR24.TKF[-]. [The file specification must be passed as either an

ASCIZ character string or an ASCII character string ending with a dollar sign ($) character. Blank characters are ignored.]

If a /A or /APPEND switch is encountered at the end of a file specification, then the file append switch is turned ON and the file will be opened in the append mode. (This is equivalent to calling subroutine APNTKF prior to this logical function, as seen in Sect. 7.4.1.)

Example:

```
        LOGICAL OPNTKF
        .
        .
        .
C
C  OPEN A .TKF FILE
        IF (OPNTKF('DSKD:PRETTY.TKF[274,23Ø4,WHERE,ITS,AT]/APPEND'))
     1  GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('CALL TO OPNTKF FAILED.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

7.2.2 Logical Function CLSTKF

The purpose of logical function CLSTKF is to close the currently open intermediate .TKF plot data file. If there are characters in the internal TCS buffer, they are output before the file is closed.

Calling sequence:

IF (CLSTKF(Ø)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram CLSTKF.

Argument list: A dummy argument is required.

Example:

```
        LOGICAL CLSTKF
        .
        .
        .
C
C  CLOSE THE .TKF FILE
        IF (CLSTKF(Ø)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('CALL TO CLSTKF FAILED.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

7.3 REWRITTEN TCS OUTPUT AND TIMING SOFTWARE

The essence of the TCSTKF software library capability is the augmented functioning of subroutine ADEOUT, an installation-dependent subprogram for writing ADE characters to the user's terminal. (For a description of the version of subroutine ADEOUT currently used by the TCSTEK software library, see Ref. 3.) This subprogram has been augmented to transmit the same ADE characters to a disk file and to the terminal. Also, subprogram SLP, the TCS subroutine that controls the page refresh timing delay, has been modified to prevent I/O waits under certain conditions.

7.3.1 Subroutine ADEOUT

The function of subroutine ADEOUT is to optionally send ADE characters to the Tektronix terminal and/or intermediate .TKF plot data file, depending upon true values of the TEK and TKF ON/OFF switches, respectively. This subroutine replaces a similarly named subroutine in the TCSTEK software library.

Calling sequence:

```
CALL ADEOUT(NADE,IADE)
```

Argument list:

NADE is the number of ADE characters to be transmitted.

IADE is a single-dimension array of ADE characters to be transmitted stored one right-justified ADE character per word.

Example:

```
DIMENSION IERASE (2)
          .
          .
          .
IERASE(1) = "33            !ADE code for escape
IERASE(2) = "14            !ADE code for form feed
CALL ADEOUT (2,IERASE)
          .
          .
          .
```

7.3.2 Subroutine SLP

The Tektronix storage tube must have time to refresh after a new page operation so that graphics output will not be lost. This is accomplished by putting the program to sleep. The version of subroutine SLP in the TCSTEK software library unconditionally puts the user's program to sleep. However, the version of subroutine SLP in the TCSTKF library will conditionally put the user's program to sleep only if the TEK ON/OFF switch is on. If the TEK ON/OFF switch is off, then Tektronix terminal display is not available, the timing problem does not exist, and therefore there is no reason to delay the program's execution.

Calling sequence:

```
CALL SLP (ISEC)
```

Argument list:

ISEC is the integer number of seconds the program should go to sleep. Two seconds is generally sufficient to pause for screen erasure. Seven seconds is generally sufficient to pause for a Tektronix hard-copy device to make a plot.

7.4 OTHER MISCELLANEOUS SUBROUTINES

The following five subprograms are used internally by the TCSTKF software; however, they may be called (with caution) by a user to achieve his desired result.

7.4.1 Subroutine APNTKF

A call to subroutine APNTKF will cause subsequent display images to be appended to an existing .TKF file instead of creating a completely new .TKF file. If the specified .TKF file does not exist, it is not an error. In this case the subroutine call is ignored and a new .TKF file is created.

Calling sequence:

```
CALL APNTKF
```

Argument list: None

Restrictions:

So that this subroutine can function correctly, it is necessary to call subroutine APNTKF before any of the actual file-opening subroutines (OPNTKF and INITKF) are accessed.

7.4.2 Subroutine NOTKF

Subroutine NOTKF will prevent an intermediate .TKF plot data file from being opened by any of the file-opening routines. By calling this subroutine before any other graphics subroutines, the TCSTKF software library becomes equivalent in external function to the TCSTEK software library.

Calling sequence:

```
CALL NOTKF
```

Argument list: None

7.4.3 Logical Function SNDTKF

Subroutine SNDTKF's function is to force out to disk the current intermediate .TKF plot data file character buffer. In general, this routine should never be called by the user.

Calling sequence:

IF (SNDTKF(Ø)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram SNDTKF.

Argument list: A dummy argument is required.

Example:

```
        LOGICAL SNDTKF
        .
        .
        .
C
C  FORCE OUT A CHARACTER BUFFER
        IF (SNDTKF(Ø)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('?ERROR WRITING .TKF FILE.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

7.4.4 Subroutine CTSEND

Subroutine CTSEND's function is to check the presence of ADE characters remaining in the internal TCS character buffer. If there are any ADE characters remaining, then the TCS subroutine TSEND is called to send these characters to the graphics devices and reinitialize the internal TCS character buffer.

Calling sequence:

IF (CTSEND(Ø)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram CTSEND.

Argument list: A dummy argument is required.

Example:

```
        LOGICAL CTSEND
        .
        .
        .
C
C  SEND CHARACTERS TO THE GRAPHICS DEVICES
        IF (CTSEND(Ø)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('?ERROR SENDING .TKF CHARACTERS.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

7.4.5 Subroutine TEKBAT

Subroutine TEKBAT's function is to determine if BATCON is controlling the user's job. If so, then the TEK ON/OFF switch is turned off. This subroutine effectively prevents the user from inadvertently turning on the TEK ON/OFF switch even if a call to subroutine TEKON is made while the program is being executed under BATCON.

Calling sequence:

IF (TEKBAT(Ø)) success, failure

where success and failure symbolically represent statement labels to which a user's program can branch depending upon the logical value returned by subprogram TEKBAT.

Argument list: A dummy argument is required.

Example:

```
        LOGICAL TEKBAT
        .
        .
        .
C
C  CHECK FOR BATCH STATUS
        IF (TEKBAT(Ø)) GO TO 1ØØ
C
C  HERE IF FAILURE
        TYPE 5ØØØ                    !PRINT ERROR MESSAGE
5ØØØ    FORMAT('?ERROR CHECKING BATCH STATUS.  JOB STOPPED')
        STOP
C
C  HERE IF SUCCESSFUL
1ØØ     CONTINUE
        .
        .
        .
```

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

8. THE .TKF FILE FORMAT

The TCS software accepts higher-level subprogram calls and converts them into graphics device instructions that are output to the user's terminal via subroutine ADEOUT. These device primitives (machine instructions) are actually special sequences of ADE characters. For example, the following FORTRAN code demonstrates how to erase the screen of a Tektronix storage tube.

```
      DIMENSION IERASE (2)
                .
                .
                .
      IERASE(1) = "33           !ADE code for escape
      IERASE(2) = "14           !ADE code for form feed
                .
                .
                .
      !WRITE (5,1ØØ) IERASE     !WRITE OUT IERASE
1ØØ   FORMAT(2RI)
```

By replacing the WRITE and FORMAT statement, the same function can be performed with subroutine ADEOUT.

```
      IERASE(1) = "33
      IERASE(2) = "14
      CALL ADEOUT (2,IERASE)
```

Before the ADE characters that TCS sends to subroutine ADEOUT are output to disk, they are packed left-justified five characters per word, as is any other ASCII file. This modification was made in the interest of reducing the file size. There are no other differences. For a description of the special sequences of ADE control characters and the effect they have on a user's terminal, see Ref. 7.

Because .TKF files are ASCII, they can be typed to the user's terminal by the instruction

```
.TYPE BLETCH.TKF
```

However, there is one annoyance with the above instruction. If the

display image contains the ASCII rubout character as a coordinate location, then part of the vector that would have been generated will be destroyed (an inconvenience caused by system program PIP). The result will be glitches in the display image, even though the .TKF file is perfectly correct. Because it is very probable that some of the vectors on a display image will contain a rubout character, the PLOT command should be used to view .TKF files on a Tektronix; e.g.,

```
.PLOT TEK:=BLETCH.TKF
```

Also, the same software that creates the rubout character problem will prevent a successful combination of two .TKF files for the same reason. This can be eliminated by using the /I switch when using the COPY command. For example,

```
.COPY COMBIN.TKF=FIRST.TKF/I,LAST.TKF/I
```

Remember — in this example if LAST.TKF does not begin with a new page instruction (or if FIRST.TKF does not end with one), then the last display image contained in FIRST.TKF will be overlayed on the first display image contained in LAST.TKF. This may or may not be desirable.

9. ADVANCED SAMPLE PROBLEM

The purpose of this sample program is to demonstrate how to use several of the advanced features of the TCSTKF software library. The main program, called CMDTCS in this example, is a small, interactive driver that controls the value of an exponentially decaying cosine function and its position on a Tektronix screen. The actual drawing of the display image is done with subroutine DRWRNG (see lines 145 through 18Ø). This subroutine is an edited version of the program discussed in Sect. 5.

Control is achieved via ten commands (see lines 14 through 23). The command prompt is the terminal bell. A list of the commands and their functions follows.

Command	Action taken
SWINDO	This command changes the location and extent of the TCS screen window. The user will be asked for MINX, LENX, MINY, and LENY, the four arguments to subroutine SWINDO.
DWINDO	This command changes the location and extent of the TCS virtual window. The user will be asked for XMIN, XMAX, YMIN, and YMAX, the four arguments to subroutine DWINDO.
CONSTANT	This command changes the value of the exponential power factor, CON, for the function $F(x) = \cos(x) * e^{-(CON * x)}$
DRAW	This command draws an exponentially decaying cosine function in the current screen and virtual windows by calling subroutine DRWRNG.
NEWPAG	This command calls subroutine FINPAG to transfer the temporary .TKF file into the permanent .TKF file, TEST.TKF.

TEKON	This command calls subroutine TEKON to turn on the TEK ON/OFF switch.
TEKOFF	This command calls subroutine TEKOFF to turn off the TEK ON/OFF switch.
TKFON	This command calls subroutine TKFON to turn on the TKF ON/OFF switch.
TKFOFF	This command calls subroutine TKFOFF to turn off the TKF ON/OFF switch.
STOP	This command terminates execution of the program, updating the intermediate .TKF file as necessary.

The following is a listing of the program.

```
1            PROGRAM CMDTCS
2     C
3            PARAMETER NCMD = 1Ø
4     C
5            DOUBLE PRECISION CMD(NCMD),TTYC
6     C
7            LOGICAL TRANF
8     C
9            DIMENSION TCSVAR(6Ø)
1Ø    C
11           COMMON /DRWPAR/ ISW(4),RVW(4),CON
12    C
13    C  LIST OF VALID COMMANDS
14           DATA CMD/          'SWINDO',
15           1                  'DWINDO',
16           3                  'CONSTANT',
17           4                  'DRAW    ',
18           5                  'NEWPAG',
19           6                  'TEKON ',
2Ø           7                  'TEKOFF',
21           8                  'TKFON ',
22           9                  'TKFOFF',
23           1                  'STOP  '/
24    C
25    C  INITIALIZE THE TCS SOFTWARE LIBRARY
26           IF (.NOT. INITKF('TEST',-1)) GO TO 999
27           TRANF = .FALSE.
28    C
29    C  LOOP FOR COMMAND SCANNER
3Ø    1ØØ    CONTINUE
31           CALL HOME
32    14Ø    CONTINUE
```

```
C
C  SAVE CURRENT TERMINAL STATUS
      CALL SVSTAT (TCSVAR)
13Ø   CONTINUE
C
C  RESTORE PREVIOUS TERMINAL STATUS
      CALL RESTAT (TCSVAR)
C
C  RING BELL FOR A PROMPT
      CALL BELL
      CALL NEWLIN
      CALL ANMODE
      CALL TSEND
C
C  INPUT A COMMAND
      ACCEPT 1ØØØ,TTYC
C
C  LOOP THROUGH THE VALID COMMANDS FOR A MATCH.
      DO 11Ø I = 1,NCMD
      IF (TTYC .EQ. CMD(I)) GO TO 12Ø
11Ø   CONTINUE
      TYPE 1Ø1Ø,TTYC
C
C  INPUT COMMAND WAS INVALID. TYPE ERROR AND TRY AGAIN.
      GO TO 14Ø
12Ø   CONTINUE
C
C  INPUT COMMAND WAS VALID.  BRANCH TO APPROPRIATE LOCATION
      GO TO (21Ø,22Ø,23Ø,24Ø,25Ø,26Ø,27Ø,28Ø,29Ø,3ØØ) , I
21Ø   CONTINUE
C
C  *** SWINDO
C
C  INPUT THE SCREEN WINDOW PARAMETERS
      TYPE 121Ø
      ACCEPT *,ISW
      CALL NEWLIN
      GO TO 14Ø
22Ø   CONTINUE
C
C  *** DWINDO
C
C  INPUT THE VIRTUAL WINDOW PARAMETERS
      TYPE 122Ø
      ACCEPT *,RVW
      CALL NEWLIN
      GO TO 14Ø
23Ø   CONTINUE
C
C  *** CONSTANT
C
C  INPUT THE EXPONENTIAL POWER FACTOR
```

```
85              TYPE 123Ø
86              ACCEPT *,CON
87              CALL NEWLIN
88              GO TO 14Ø
89        24Ø   CONTINUE
9Ø        C
91        C  *** DRAW
92        C
93        C  SAVE CURRENT TERMINAL STATUS
94              CALL NEWLIN
95              CALL SVSTAT (TCSVAR)
96        C
97        C  CALL DRWRNG TO DRAW AN EXPONENTIALLY DECAYING COSINE
98              CALL DRWRNG
99              TRANF = .TRUE.
1ØØ             GO TO 13Ø
1Ø1       25Ø   CONTINUE
1Ø2       C
1Ø3       C  *** NEWPAG
1Ø4       C
1Ø5       C  CALL FINPAG TO SEE ABOUT DISPLAY IMAGE TRANSFER
1Ø6             CALL TEKPMT (Ø,Ø)
1Ø7             CALL FINPAG(JUNK)
1Ø8             TRANF = .FALSE.
1Ø9             GO TO 14Ø
11Ø       26Ø   CONTINUE
111       C
112       C  *** TEKON
113             IF (.NOT. GTSTEK(Ø)) CALL TEKON
114             GO TO 13Ø
115       27Ø   CONTINUE
116       C
117       C  *** TEKOFF
118             IF (GTSTEK(Ø)) CALL TEKOFF
119             GO TO 13Ø
12Ø       28Ø   CONTINUE
121       C
122       C  *** TKFON
123             IF (.NOT. GTSTKF(Ø)) CALL TKFON
124             GO TO 13Ø
125       29Ø   CONTINUE
126       C
127       C  *** TKFOFF
128             IF (GTSTKF(Ø)) CALL TKFOFF
129             GO TO 13Ø
13Ø       3ØØ   CONTINUE
131       C
132       C  *** STOP
133       C
134       C  ASK ABOUT LAST PAGE, THEN QUIT.
135             IF (TRANF) CALL FINPAG(JUNK)
136             CALL FINITT (Ø,78Ø)
```

```
137     999     CONTINUE
138             STOP
139     1ØØØ    FORMAT(A1Ø)
14Ø     1Ø1Ø    FORMAT(' BAD COMMAND? ',A1Ø)
141     121Ø    FORMAT(' INPUT MINX, LENX, MINY,  LENY '$)
142     122Ø    FORMAT(' INPUT XMIN, XMAX, YMIN,  YMAX '$)
143     123Ø    FORMAT(' INPUT CONSTANT '$)
144             END
145     C
146             SUBROUTINE DRWRNG
147     C
148             COMMON /DRWPAR/ ISW(4),RVW(4),CON
149     C
15Ø             DATA ISW/2ØØ,4ØØ,2ØØ,4ØØ/
151             DATA RVW/Ø.,2Ø.,-1.,1./
152             DATA CON/Ø.2/
153     C
154     C  SET UP A SCREEN AND VIEWING WINDOW
155             CALL SWINDO (ISW(1),ISW(2),ISW(3),ISW(4))
156             CALL DWINDO (RVW(1),RVW(2),RVW(3),RVW(4))
157     C
158     C  DRAW RINGER
159     C
16Ø     C  FIRST CALCULATE AND MOVE TO BEGINNING POSITION
161             Y = COS(RVW(1)) * EXP(CON * -RVW(1))
162             CALL MOVEA (RVW(1),Y)
163     C
164     C  NOW LOOP THROUGH ALL THE POINTS
165             DO 1ØØ T=RVW(1),RVW(2),.Ø5
166             Y = COS(T) * EXP(CON * -T)
167             CALL DRAWA (T,Y)
168     1ØØ     CONTINUE
169     C
17Ø     C  DRAW A GRID
171             DO 2ØØ X = RVW(1),RVW(2),2.
172             CALL MOVEA (X,RVW(3))
173             CALL DRAWA (X,RVW(4))
174     2ØØ     CONTINUE
175             DO 21Ø Y = RVW(3),RVW(4),.2
176             CALL MOVEA (RVW(1),Y)
177             CALL DRAWA (RVW(2),Y)
178     21Ø     CONTINUE
179             RETURN
18Ø             END
```

The following sequence of commands produced the display image in Fig. 2.

```
SWINDO
Ø,35Ø,Ø,35Ø
```

```
DRAW
SWINDO
400,350,0,350
CONSTANT
.1
DRAW
SWINDO
400,350,400,350
CONSTANT
.05
DRAW
SWINDO
0,350,400,350
CONSTANT
.01
DRAW
STOP
Y
```

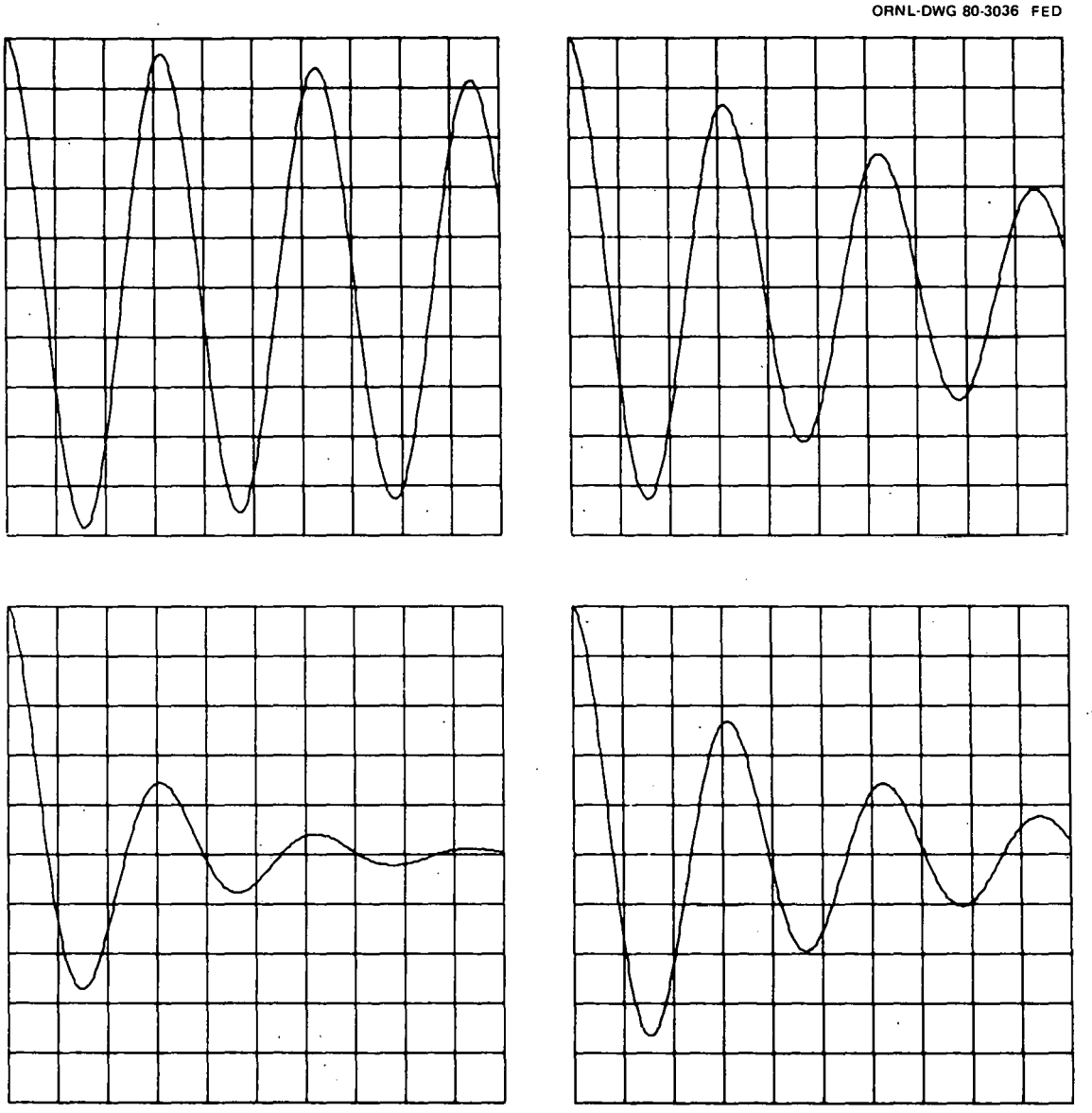

Fig. 2. The display image for the advanced example problem in Sect. 9.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

1Ø. CONCLUSIONS

A new software library, called TCSTKF, has been described. This software library will create display images on the screen of a Tektronix 4ØØØ series storage tube terminal and will create display images in an intermediate .TKF plot data file. This library makes a convenient way for users of TCS software to generate report-quality drawings of the same display image that is drawn on their Tektronix terminals.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ACKNOWLEDGMENTS

We would like to thank D. W. Swain and G. H. Neilson, Jr., for their critical debugging of the TCSTKF software library. Also, thanks go to K. H. Carpenter for his thoughtful suggestions. Finally, thanks (we think) go to J. K. Ballou for relentlessly prodding us to finish this documentation.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

REFERENCES

1. H. A. Kamel and M. W. McCabe, "The GIFTS System: Version 4B User's Manual," University of Arizona, Tucson, Arizona (March 1978).

2. "Tektronix PLOT 1Ø Terminal Control System User Manual," Tektronix, Inc., Beaverton, Oregon (February 1977).

3. W. H. Gray, "User and System Considerations for the TCSTEK Software Library," ORNL/TM-6918, Oak Ridge, Tennessee (August 1979).

4. R. D. Burris and W. H. Gray, "A Generalized Plotting Facility," Proc. of the Digital Equipment Computer Users Society, San Francisco, California (November 1978).

5. Appendix D in "DECsystem-1Ø FORTRAN Programmers Reference Manual," Digital Equipment Corporation, Maynard, Massachusetts (June 1977).

6. W. H. Gray and D. N. Clark, "A User's Guide to the Installation Dependencies of the FED DISSPLA Software," in preparation.

7. "Tektronix 4Ø15 and 4Ø15-1 Computer Display Terminal Instruction Manual," Tektronix, Inc., Beaverton, Oregon (August 1974).

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ORNL/TM-7383
Dist. Category UC-32

INTERNAL DISTRIBUTION

1. F. W. Baity
2. T. V. Baudry
3. J. D. Bell
4. F. M. Bieniosek
5. W. B. Bigney
6. T. G. Brown
7-11. R. D. Burris
12. B. A. Clark
13. D. N. Clark
14. J. A. Cobble
15. R. J. Colchin
16. W. A. Davis
17. R. A. Dory
18. L. C. Emerson
19. R. H. Fowler
20. J. C. Glowienka
21. G. E. Gorker
22-26. W. H. Gray
27. D. E. Greenwood
28. C. E. Hammons
29. W. R. Hendrich
30. D. L. Hillis
31. R. J. Hooper
32. L. M. Jenkins
33. T. C. Jernigan
34. H. E. Ketterer
35. P. W. King
36. A. Komori
37. R. A. Langley
38. E. A. Lazarus
39. M. S. Lubell
40. L. B. Maddox
41. B. F. Maskewitz
42. D. H. McCullough
43. R. W. McGaffey
44. G. S. McNeilley
45. J. R. Moore
46. M. Murakami
47-48. Laboratory Records Department
49-50. Central Research Library
51. Document Reference Section
52. Laboratory Records, ORNL-RC
53. ORNL Patent Office
54-55. Fusion Energy Division Library
56. Fusion Energy Division Reports Office

EXTERNAL DISTRIBUTION

57. D. J. Anthony, General Electric Company, 1 River Road, Building 23, Room 290, Schenectady, NY 12345
58. S. O. Dean, Science Applications, Inc., 2 Professional Drive, Suite 249, Gaithersburg, MD 20760
59. H. K. Forsen, Exxon Nuclear Company, Inc., 777 106th Avenue, NE, Bellevue, WA 98009
60. H. P. Furth, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08540
61. R. L. Hirsch, Exxon Research and Engineering Company, P.O. Box 101, Florham Park, NJ 07932
62. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830

63-245. Given distribution as shown in TID-4500, Mathematics and Computers (Distribution Category UC-32)

ORNL/TM-7383

A User's Guide to the TCSTKF Software Library: A Graphics Library for Emulation of TEKTRONIX Display Images in .TKF Disk Files

W. H. Gray R. D. Burris

MICROFICHE ENCLOSED