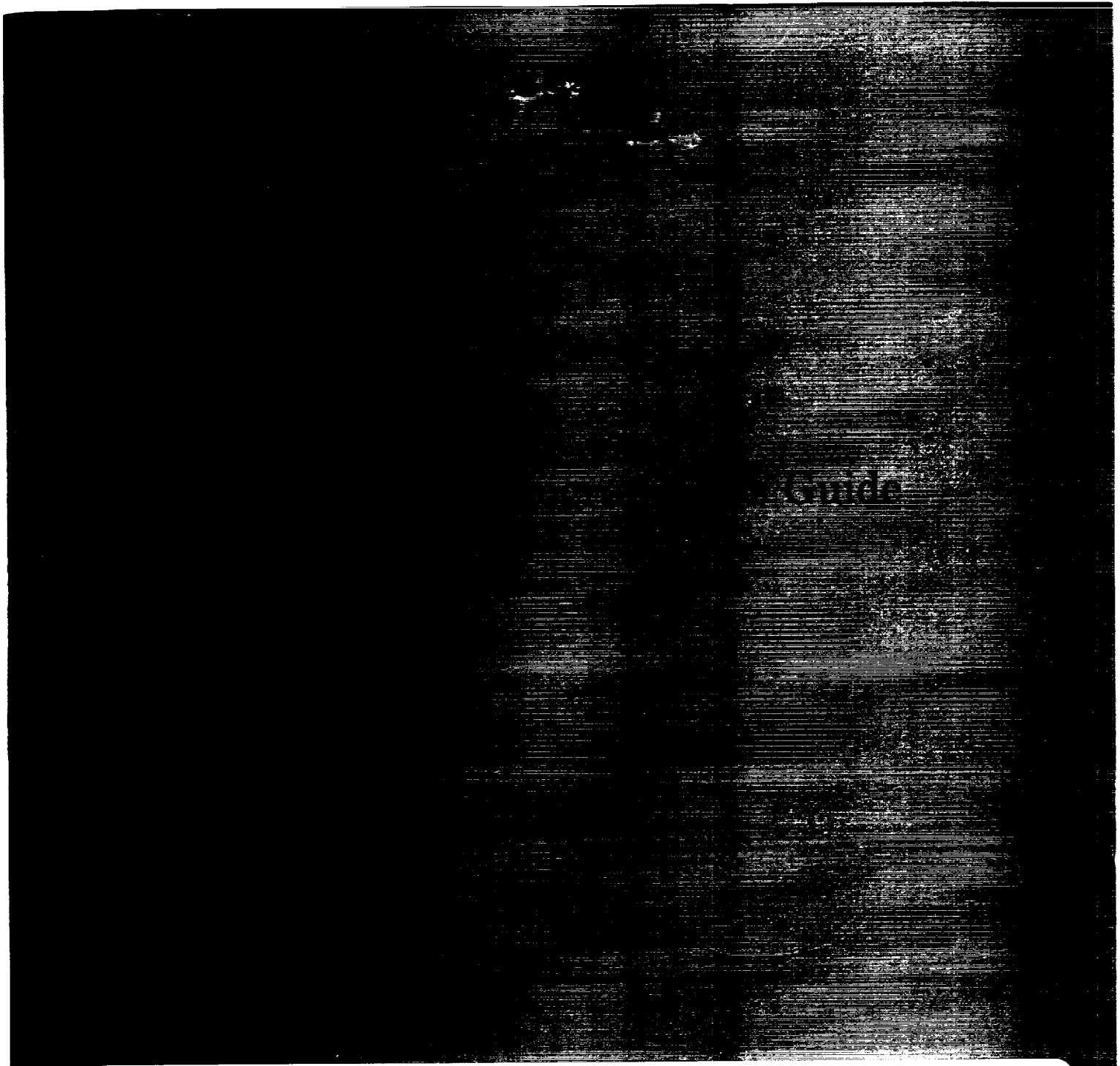
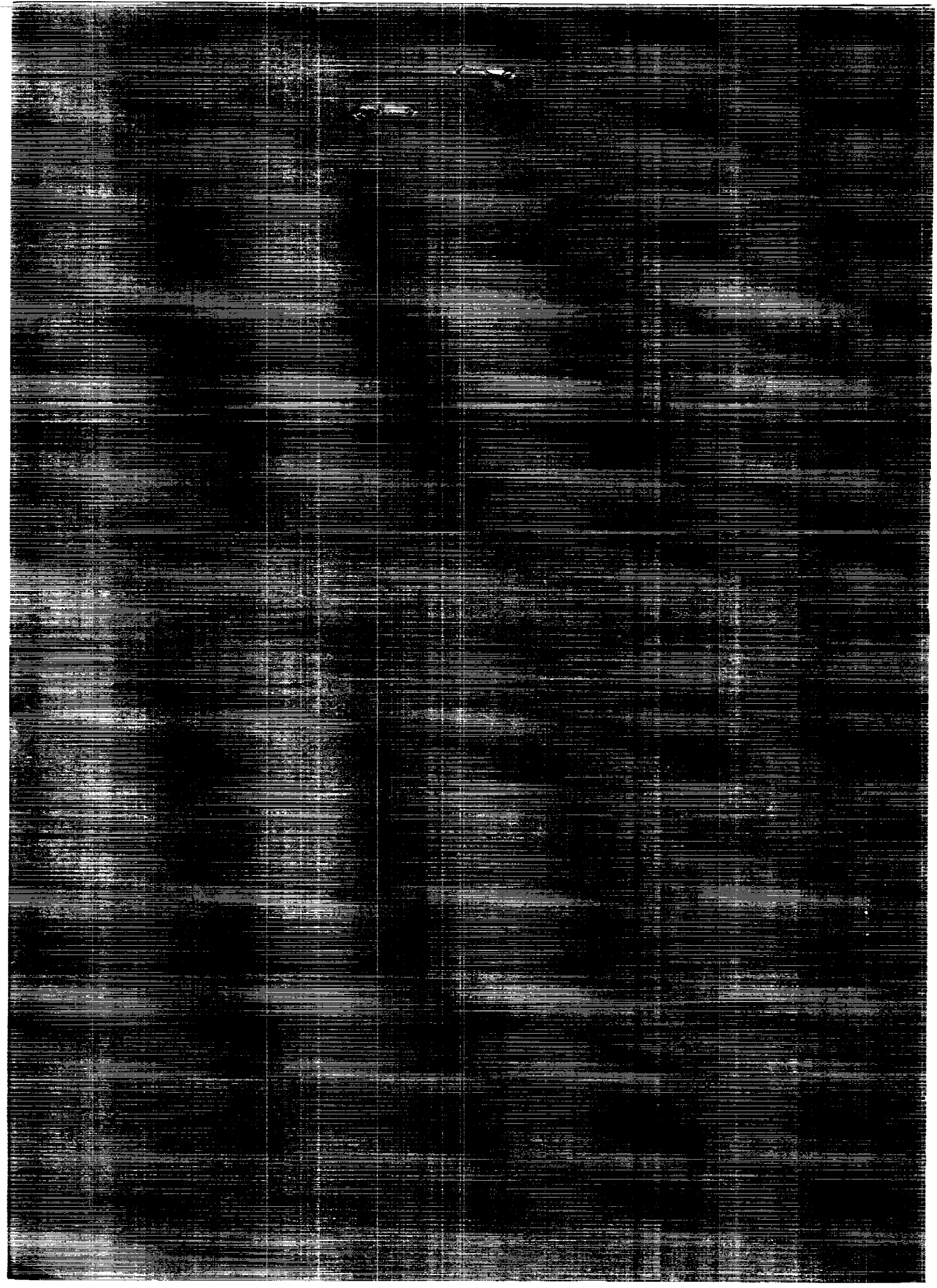




NASA Technical Memorandum 4261
Part 2

GEMPAK5
Part 2—GEMPLT Programmer's Guide

Version 5.0

Mary L. desJardins
Goddard Space Flight Center
Greenbelt, Maryland

Keith F. Brill
NOAA/NWS/NMC
Washington, D.C.

Steven S. Schotz
General Sciences Corporation
Lanham, Maryland

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1991

CHAPTER 1

GEMPLT OVERVIEW

GEMPLT is a general plotting subroutine package which provides the applications programmer with device independent line drawing capabilities. It has been designed to simplify the plotting of meteorological data. New graphic devices can be added easily.

GEMPLT allows plotting in a variety of coordinate systems. The transformations from one coordinate system to another are handled internally. Lines, text, markers, wind barbs and arrows can be plotted in any coordinate system. Utility functions to draw contours, streamlines and geopolitical maps are available.

GEMPLT has been designed to be extremely flexible for applications programmers developing either single programs or large applications systems. Programs can be written that are independent of graphics device and map projection, both of which can be selected at run time. New features, coordinate systems and graphics devices can be added easily without affecting existing applications programs.

The GEMPLT subroutines communicate with two subprocesses which are maintained as separate modules. Coordinate transformations and utility functions are included in the first subprocess called GPLT. The second subprocess is a device driver which draws lines, text and symbols. The device characteristics which are needed by GPLT are defined in the device driver. A new device driver must be written for each graphics device. Since the subprocesses can remain active after an applications program exits, plotting characteristics defined in one program may be used in other programs that follow it in the same session.

This document provides information for an applications programmer. An introduction to GEMPLT subroutine descriptions and examples are included.

CHAPTER 2

INTRODUCTION TO GEMPLT

This chapter provides a brief description of the GEMPLT functions. Required logical name assignments are described in an appendix.

2.1 INITIALIZATION AND TERMINATION

GINITP must be the first GEMPLT subroutine called by each program. The first time GINITP is called, the GPLT task is installed. A call to GINITP in later programs is required to establish communications with GPLT. When GINITP is called, a mode must be specified. Mode 1 indicates plotting will be in earth (latitude/longitude) coordinates. Mode 2 is used for drawing graph (x/y) plots. Graphs may not be defined or drawn in earth mode; map projections may not be defined or plotted in graph mode. The transformations for both earth and graph are retained in GPLT. If the mode is changed, the appropriate transformation is activated.

GENDP should be the last GEMPLT subroutine called by each application program. It will cause any data buffered internally by GEMPLT to be flushed. An option exists to stop the GPLT and device driver subprocesses.

2.2 CONTROL OF PLOTTER SPACE

The graphics device to be used for plotting is specified by a call to GSDEV, which installs the device driver as a separate process. This subroutine must be called before any plotting is done. Some transformations cannot be defined until GSDEV is called.

GSVIEW allows the user to specify a view region. The view region

is specified in fractions from 0 to 1. The point (0,0) is the lower left of the plotter space; (1,1) is the upper right corner. Thus the upper right quadrant can be selected by placing the lower left corner of the view region at (.5,.5) and the upper right corner at (1,1). Note that, except for square devices, these fractional units are not the same as the normalized coordinates defined later.

The programmer may also control the plotter space by defining margins. GSMMGN defines margins for plotting in map projections; GSGMGN defines margins for graphs. No margins will be used when plotting on satellite imagery. The margins in effect when the M coordinate system is defined will be used until the coordinate system is redefined. Later calls to the margins will have no effect until the coordinate system is redefined. Margins may be defined as fractions for the entire graphics region or as multiples of the text size.

2.3 COORDINATE SYSTEMS

GEMPLT uses four coordinate systems which are organized in a hierarchical structure. These coordinate systems are described briefly here.

- D coordinates are the actual device coordinates.

- N coordinates are normalized device coordinates. N coordinate limits range from 0.0 to 1.0 along the longest dimension and from 0.0 to a value less than or equal to 1.0 on the other axis. These limits are determined using a linear scaling of the device coordinates. Correction for the aspect ratio of the pixel is included in the transformation from D to N coordinates.

  View regions and margins may be specified in GEMPLT. Use of these parameters allows two clipped versions of N coordinates:

  - V coordinates, view coordinates, are the same as normalized coordinates but the clipping boundaries correspond to the view region.

  - P coordinates, plot coordinates, are the same as normalized coordinates but the clipping boundaries correspond to the region inside the margins.

- M coordinates are the earth or graph coordinates in which plot locations are generally specified. Map projection functions transform any point on the earth into a (possibly infinite)

plane, a subregion of which is scaled to the normalized coordinate system for display on the output device. Clipping boundaries are the same as for plot coordinates.

- o G coordinates are grid coordinates. This coordinate system defines the projection on which a grid was created. The grid is assumed to be evenly spaced in this projection. Using the inverse of the standard map or graph projection functions, grid point coordinates can be translated into earth coordinates. The grid projection is independent of the map coordinate projection. Clipping is the same as for plot coordinates.

In addition, the plane into which M coordinates are transformed defines an L coordinate system of linear coordinates. Since these are different for each projection, they are not generally used by the applications programmer.

GQBND can be called to query the limits of any coordinate system. The bounds of the M and G regions are estimates of the maximum and minimum values of the coordinates along each side.

GTRANS will translate a list of x,y points from any coordinate system into any other system.

2.4 ATTRIBUTES

Attribute subroutines allow plotting characteristics to be modified and queried. Subroutines are available to change line, text, marker and wind symbol attributes. Color modification subroutines are described separately.

The line attributes are set in GSLINE. Line type specifies the dashing pattern. Line width may also be specified. Flags to change from software to hardware functions for generating line types and widths are available.

GSTEXT is used to define the text font, text size multiplier, text width, and a flag specifying generation of software or hardware text.

GSMRKR sets marker characteristics including marker number, software/hardware marker flag, marker size multiplier and marker width.

GSBARB and GSARRW set the wind barb and the arrow attributes. GSWTHR sets weather symbol characteristics including size and width.

If a hardware function is specified for a device on which it is unavailable, the software function will be substituted automatically. The attributes that will be used can be obtained by calling the corresponding query subroutine. If a hardware flag is set and the hardware function is available, it will be used even if other attributes, such as size, must be modified. Note that the hardware option for many graphics devices will not draw dashed lines well when a long series of short line segments is drawn. Such short segments are generally used for drawing maps and contours.

2.5 PLOTTING FUNCTIONS

The basic GEMPLT plotting functions are the generation of lines, text, markers, wind and other meteorological symbols.

The location for each symbol is given by its x,y position and coordinate system. For plotting in earth coordinates, latitude and longitude replace x and y. The units and clipping boundaries are dependent on the coordinate system. In each case, the input coordinates are translated to device coordinates for plotting and clipping. The marker center, the first character of a text string and the base of a wind barb or arrow will be plotted at the given location.

2.6 UTILITIES

GEMPLT utility functions are implemented to perform complex plotting functions. Contours, streamlines and geopolitical maps can be plotted.

The name of the map file to be used can be changed using GSMFIL. The default map is a medium resolution map. GQMFIL is used to query the current map file name.

2.7 COLOR

GSCOLR and GQCOLR can be used to set and query the color using color numbers. Colors are specified by numbers from 1 to N where N is the number of colors defined for a particular device.

If a program requests a color number larger than available on the the current device, the color used will correspond to that of the the requested number modulo-N.

Subroutines are available to change the color components of a graphics plane on devices where this feature is available. The color components may be specified as red, green and blue intensities (RGB), as hue, lightness and saturation (HLS) or by name. Names are kept in a GEMPLT text file which may be modified by a GEMPLT systems programmer.

2.8 ERRORS

All GEMPLT subroutines send back a return code called IRET. The normal value, indicating that no error was encountered, is 0. All other error numbers are negative and indicate that the subroutine did not execute normally. The substitution of software functions for hardware functions which are requested, but not available, is not considered an error.

Programmers who prefer to check for error within programs by using parameter names should include the following file in the program declarations:

GEMINC:ERROR.PRM

The parameter name for the 0 error code is NORMAL.

Error messages may be printed using the subroutine GERROR. A list of error messages and names is given in an appendix.

CHAPTER 3

GEMPLT SUBROUTINES

GAAXIS - Draw Axes With Alphanumeric Labels

This subroutine draws a graph background with labels, tick marks, and grid lines. The graph coordinates must be defined by a call to GSGRAF before this subroutine is called.

Line attributes apply to grid lines. The axis line will be a solid line. Tick mark attributes can be set in GSTICK.

The label string (CHARY) and position (AXARY) arrays are allowed a maximum of 530 elements. Each CHARY element may contain up to 24 characters.

LBFRQ, MTFRQ and LGFRQ are specified in the form FF where FF is the frequency with which to plot items in the AXARY positions. For example, LBFRQ = 3 plots every third element of CHARY beginning with the first, that is, CHARY (1) at AXARY (1), CHARY (4) at AXARY (4), CHARY (7) at AXARY (7), and so on.

In a polar coordinate system, the axis is not drawn. When IAXIS is 1 or 3, grid lines are circles with the radii specified in AXARY. When IAXIS is 2 or 4, AXARY specifies the angle for radial lines which are drawn from the center of the circle to AXPOS. Tick marks are not drawn in polar coordinates.

Except for CHARY, GAAXIS is identical to GDAXIS.

```
GAAXIS ( IAXIS, AXPOS, LAXIS, LBFRQ, MTFRQ, LGFRQ, NP,
         AXARY, CHARY, IRET )
```

Input parameters:

IAXIS	INTEGER	Axis 1 = x axis labelled below 2 = y axis labelled left 3 = x axis labelled above 4 = y axis labelled right
AXPOS	REAL	Intersection with other axis
LAXIS	LOGICAL	Axis draw flag
LBFRQ	INTEGER	Frequency of labels
MTFRQ	INTEGER	Frequency of tick marks
LGFRQ	INTEGER	Frequency of grid lines
NDEC	INTEGER	# of decimal places in labels

NP	INTEGER	Number of values in AXARY
AXARY (NP)	REAL	Locations on other axis
CHARY (NP)	CHAR*24	Label strings

Output parameters:

IRET	INTEGER	Return code

GARRW - Plot Arrows

This subroutine draws wind arrows at points defined in any coordinate system. The length of the arrow is proportional to speed, and its orientation is relative to local north. If the arrows are not plotted on a map projection, local north is assumed to be vertical. By convention, the direction is the direction from which the wind is blowing. The arrows will be drawn using attributes defined in GSARRW.

GARRW (SYS, NP, X, Y, SPD, DIR, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of arrows
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
SPD (NP)	REAL	Wind speeds
DIR (NP)	REAL	Wind directions

Output parameters:

IRET	INTEGER	Return code

GBARB - Plot Barbs

This subroutine draws wind barbs at points defined in any coordinate system. The orientation is relative to local north. If the barbs are not plotted on a map projection, local north is assumed to be vertical. By convention, the direction is the direction from which the wind is blowing. The barbs will be drawn using attributes defined in GSBARB.

GBARB (SYS, NP, X, Y, SPD, DIR, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of barbs
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
SPD (NP)	REAL	Wind speeds
DIR (NP)	REAL	Wind directions

Output parameters:

IRET	INTEGER	Return code

GCIRCL - Draw Circle

This subroutine draws a circle centered at a point which may be defined in any coordinate system. The radius of the circle, RAD, must be given as a distance in normalized coordinates. NP is the number of points to be used in drawing the circle. If NP is zero, 10 points will be used.

GCIRCL (SYS, X, Y, RAD, NP, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
XPT	REAL	X coordinate / latitude
YPT	REAL	Y coordinate / longitude
RAD	REAL	Radius in N coordinates
NP	INTEGER	Number of points on circle

Output parameters:

IRET	INTEGER	Return code

GCLDHT - Set Cloud Height For AOIPS NAV

This subroutine sets the cloud height to be used in the AOIPS satellite projection. A call to GSATAO will reset the height to 0. Height is given in kilometers.

GCLDHT (HEIGHT, IRET)

Input parameters:

HEIGHT	REAL	Cloud height

Output parameters:

IRET	INTEGER	Return code

GCLEAR - Clear Plot Frame

This subroutine clears the current device. On a direct-access device, GCLEAR erases the screen. On a continuous-paper plotter, GCLEAR will advance to the next page. On a single-page plotter, GCLEAR will unload the paper so another sheet can be loaded.

```
GCLEAR ( IRET )
```

Output parameters:

IRET	INTEGER	Return code

GCLOSP - Close Plot File

This subroutine closes any intermediate plot file used by a sequential access type device. When plotting is continued, a new file will be opened. No action is taken if the current device does not use an intermediate plot file.

GCLOSP (IRET)

Output parameters:

IRET	INTEGER	Return code

GCONTR - Draw Contours

This subroutine draws contours through a grid of data. A grid coordinate system and a map/graph transformation must be defined before this subroutine is called. The maximum size of the subgrid is 125 x 125. The maximum number of contour levels is 50. If MISFLG is set, data will be interpolated/extrapolated from the adjacent points. Contours will be computed for a subgrid with the lower left corner at (IMINX,JMINY) and an upper right corner at (IMAXX,JMAXY). If these values are all 0, the entire grid will be used.

```
GCONTR ( KX, KY, GRID, IMINX, JMINY, IMAXX, JMAXY, NLVL, CLVL,
         LABEL, ICOLR, ILTYP, ILWID, MISFLG, IRET )
```

Input parameters:

KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
GRID (KX,KY)	REAL	Grid data array
IMINX	INTEGER	First x point of subgrid
JMINY	INTEGER	First y point of subgrid
IMAXX	INTEGER	Last x point of subgrid
JMAXY	INTEGER	Last y point of subgrid
NLVL	INTEGER	Number of contour levels
CLVL (NLVL)	REAL	Contour level values
LABEL (NLVL)	INTEGER	Contour label types
ICOLR (NLVL)	INTEGER	Contour color numbers
ILTYP (NLVL)	INTEGER	Contour line types
ILWID (NLVL)	INTEGER	Contour line widths
MISFLG	LOGICAL	Data interpolation flag

Output parameters:

IRET	INTEGER	Return code

GCTYP - Plot Cloud Type Symbols

This subroutine draws cloud type symbols defined in any coordinate system. The cloud type symbols will be drawn using attributes defined in GSCTYP.

Note that this subroutine is currently not implemented.

GCTYP (SYS, NP, CTCOD, X, Y, IXOFF, IYOFF, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of cloud symbols
CTCOD (NP)	REAL	Cloud type numeric code
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
IXOFF (NP)	INTEGER	X offsets in half characters
IYOFF (NP)	INTEGER	Y offsets in half characters

Output parameters:

IRET	INTEGER	Return code

GCYEVL - Evaluate Points On Curve

This subroutine fits a curve to a set of input points and then evaluates the curve for values of x. The input points must be in M coordinates; the current mode must be graph mode. The input points must be strictly monotonic in x:

```
xarray (1) < xarray (2) < ... < xarray (np)
                  or
xarray (1) > xarray (2) > ... > xarray (np)
```

GCYEVL (ITYPE, NP, XARRAY, YARRAY, NPOUT, XEVAL, YEVAL, IRET)

Input parameters:

ITYPE	INTEGER	Type of curve 1 = piecewise linear 2 = cubic spline
NP	INTEGER	Number of input points
XARRAY (NP)	REAL	X input coordinates
YARRAY (NP)	REAL	Y input coordinates
NPOUT	INTEGER	Number of evaluations
XEVAL (NPOUT)	REAL	X evaluation coordinates

Output parameters:

YEVAL (NPOUT)	REAL	Y evaluated coordinates
IRET	INTEGER	Return code

GDAXIS - Draw Axes With Numeric Labels

This subroutine draws a graph axis with labels, tick marks, and grid lines. The graph coordinates must be defined by a call to GSGRAF before this subroutine is called.

Line attributes apply to grid lines. The axis line will be a solid line. Tick mark attributes can be set in GSTICK.

The label position array, AXARY, is allowed a maximum of 530 elements.

If NDEC, the number of decimal places, is negative, the program will use an appropriate number.

LBFRQ, MTFRQ and LGFRQ are specified in the form FF where FF is the frequency with which to plot items in the AXARY positions. For example, LBFRQ = 03 plots every third element of CHARY beginning with the first, that is, CHARY (1) at AXARY (1), CHARY (4) at AXARY (4), CHARY (7) at AXARY (7), and so on.

In a polar coordinate system, the axis is not drawn. When IAXIS is 1 or 3, grid lines are circles with the radii specified in AXARY. When IAXIS is 2 or 4, AXARY specifies the angle for radial lines which are drawn from the center of the circle to AXPOS. Tick marks are not drawn in polar coordinates.

Except for CHARY, GAAXIS is identical to GDAXIS.

```
GDAXIS  ( IAXIS, AXPOS, LAXIS, LBFRQ, MTFRQ, LGFRQ, NDEC, NP,
          AXARY, IRET )
```

Input parameters:

IAXIS	INTEGER	Axis 1 = x axis labelled below 2 = y axis labelled left 3 = x axis labelled above 4 = y axis labelled right
AXPOS	REAL	Intersection with other axis
LAXIS	LOGICAL	Axis draw flag
LBFRQ	INTEGER	Frequency of labels
MTFRQ	INTEGER	Frequency of tick marks
LGFRQ	INTEGER	Frequency of grid lines

NDEC	INTEGER	# of decimal places in labels
NP	INTEGER	Number of values in AXARY
AXARY (NP)	REAL	Locations on other axis

Output parameters:

IRET	INTEGER	Return code

GDRGRD - Draw Latitude/longitude Lines

This subroutine draws a uniform latitude/longitude grid. The map projection must be defined before GDRGRD is called. The current color, line and text attributes will be used. The latitude and longitude intervals are given in degrees.

GDRGRD (DELLAT, DELLON, LBLFRQ, IRET)

Input parameters:

DELLAT	REAL	Latitude interval
DELLON	REAL	Longitude interval
LBLFRQ	INTEGER	Label frequency 0 = no grid labels 1 = every grid line 2 = every other grid line n = every n-th line

Output parameters:

IRET	INTEGER	Return code

GDRMAP - Draw Map

This subroutine draws a map. A map projection must be defined before it is called. The current color and line attributes will be used. The map file to be used may be specified in GSMFIL.

GDRMAP (IRET)

Output parameters:

IRET	INTEGER	Return code

GENDP - End Use Of GEMPLT

This subroutine must be the last subroutine called by any program that uses GEMPLT. Internal buffers will be flushed, if necessary. IEOP governs whether the GEMPLT subprocesses are retained, making the current parameter definitions available in later programs.

GENDP (IEOP, IRET)

Input parameters:

IEOP	INTEGER	End plot flag 0 = retain GEMPLT 1 = stop GEMPLT

Output parameters:

IRET	INTEGER	Return code

GEPLOT - End Current Plot

This subroutine is called when a plot sequence is completed. Internal plot buffers will be flushed by this subroutine and the terminal will be put in user mode.

GEPLOT (IRET)

Output parameters:

IRET	INTEGER	Return code

GERROR - Write Error Message

This subroutine writes an error message to the user's terminal. The output message will contain the word GEMPLT and the error number in brackets followed by the text of the error message. The error codes are listed in an appendix.

GERROR (NUMERR, IRET)

Input parameters:

NUMERR	INTEGER	GEMPLT error number

Output parameters:

IRET	INTEGER	Return code

GFLUSH - Flush Graphics Buffers

This subroutine flushes all internal plotting buffers. GCLEAR, GCLOSP, and GENDP also flush buffers. GFLUSH is only needed to force display of a partial plot during a program.

GFLUSH (IRET)

Output parameters:

IRET	INTEGER	Return code

GINITP - Initiate Use Of GEMPLT

This subroutine initializes GEMPLT for an application program. The GPLT subprocess is started if it has not been started by a previously executed program. This subroutine must be the first subroutine called by any program that uses GEMPLT. The MODE parameter determines the plotting mode for map/graph plots.

GINITP (MODE, ISTAT, IRET)

Input parameters:

MODE	INTEGER	Plotting mode 0 = no change 1 = map coordinates 2 = graph coordinates

Output parameters:

ISTAT	INTEGER	Status code 0 = GEMPLT started 1 = GEMPLT previously started
IRET	INTEGER	Return code

GLINE - Draw Line Segments

This subroutine draws line segments connecting an array of points defined in any coordinate system. The lines will be drawn using attributes defined in GSLINE.

GLINE (SYS, NP, X, Y, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of points
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes

Output parameters:

IRET	INTEGER	Return code

GMARK - Plot Markers

This subroutine draws markers at points defined in any coordinate system. The markers are centered at the specified locations. The markers will be drawn using attributes defined in GSMRKR.

GMARK (SYS, NP, X, Y, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of markers
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes

Output parameters:

IRET	INTEGER	Return code

GOUTP - Output Plot

This subroutine sends the output from an intermediate plot file to the graphics device. If a plot file is currently open, the file will be closed and sent to the plotter. If no plot file is open, the most recent plot file will be sent to the plotter. No action is taken if the current device does not use an intermediate plot file.

GOUTP (IRET)

Output parameters:

IRET	INTEGER	Return code

GPTND - Plot Pressure Tendency Symbols

This subroutine draws pressure tendency symbols defined in any coordinate system. The pressure tendency symbols will be drawn using attributes defined in GSPTND.

Note that this subroutine is currently not implemented.

GPTND (SYS, NP, PTCOD, X, Y, IXOFF, IYOFF, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of pressure tendency symbols
PTCOD (NP)	REAL	Pressure tendency numeric code
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
IXOFF (NP)	INTEGER	X offsets in half characters
IYOFF (NP)	INTEGER	Y offsets in half characters

Output parameters:

IRET	INTEGER	Return code

GPTVIS - Check For Visible Points

This subroutine returns a logical array indicating whether the input points are within the bounds of the map/graph to be plotted.

GPTVIS (SYS, NP, X, Y, VIS, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of points
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes

Output parameters:

VIS (NP)	LOGICAL	Visible flags
IRET	INTEGER	Return code

GPWTH - Plot Past Weather Symbols

This subroutine draws past weather symbols defined in any coordinate system. The past weather symbols will be drawn using attributes defined in GSPWTH.

Note that this subroutine is currently not implemented.

GPWTH (SYS, NP, IPWCOD, X, Y, IXOFF, IYOFF, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of past weather symbols
PWCOD (NP)	REAL	Past weather numeric code
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
IXOFF (NP)	INTEGER	X offsets in half characters
IYOFF (NP)	INTEGER	Y offsets in half characters

Output parameters:

IRET	INTEGER	Return code

GQARRW - Query Wind Arrow Attributes

This subroutine returns the current wind arrow size, arrow head size, line width and wind arrow type.

NOTE THAT THE CALLING SEQUENCE OF THIS SUBROUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GQARRW (SZARRW, SZARRH, IARWID, IARTYP, IRET)

Output parameters:

SZARRW	REAL	Wind arrow size multiplier
SZARRH	REAL	Wind arrow head size multiplier
IARWID	INTEGER	Wind arrow line width
IARTYP	INTEGER	Wind arrow type 1 = plot arrow for calm wind 2 = don't plot for calm wind
IRET	INTEGER	Return code

GQBARB - Query Wind Barb Attributes

This subroutine returns the current wind barb size, line width, and barb type.

NOTE THAT THE CALLING SEQUENCE OF THIS SUBROUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GQBARB (SZBARB, IBRWID, IBRTYP, IRET)

Output parameters:

SZBARB	REAL	Wind barb size multiplier
IBRWID	INTEGER	Wind barb line width
IBRTYP	INTEGER	Wind barb type
IRET	INTEGER	Return code

GQBND - Query Coordinate Bounds

This subroutine returns the boundaries of the specified coordinate system. For the linear coordinate systems (D,N,V,P), the lower left and upper right corners are returned. For M coordinates, an estimate of the minimum and maximum range of latitude and longitude in the plot are returned. For G coordinates, the minimum and maximum grid points which will be displayed in the plot area are returned.

GQBND (SYS, XL, YB, XR, YT, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates

Output parameters:

XL	REAL	Lower left x / latitude
YB	REAL	Lower left y / longitude
XR	REAL	Upper right x / latitude
YT	REAL	Upper right y / longitude
IRET	INTEGER	Return code

GQCOLR - Query Color Number

This subroutine returns the current color number.

GQCOLR (ICOLR, IRET)

Output parameters:

ICOLR	INTEGER	Color number
IRET	INTEGER	Return code

GQCOMP - Query Color Components

This subroutine returns the components corresponding to a color number. The red, green and blue components along with the hue, lightness and saturation values are returned. If the color was defined by name, the color name is also returned. All the color values are defined in the range 0 - 1.

GQCOMP (ICOLR, COLOR, RED, GREEN, BLUE, HUE, CLIGHT, SAT, IRET)

Input parameters:

ICOLR	INTEGER	Color number

Output parameters:

COLOR	CHAR*	Color name
RED	REAL	Red component value
GREEN	REAL	Green component value
BLUE	REAL	Blue component value
HUE	REAL	Hue value
CLIGHT	REAL	Lightness value
SAT	REAL	Saturation value
IRET	INTEGER	Return code

GQCTYP - Query Cloud Type Attributes

This subroutine returns the current cloud type symbol size and width multipliers.

Note that this subroutine is not currently implemented.

GQCTYP (SZCTYP, ICTWID, IRET)

Output parameters:

SZCTYP	REAL	Cloud type symbol size multiplier
ICTWID	INTEGER	Cloud type line width
IRET	INTEGER	Return code

GQDEV - Query Plot Device

This subroutine returns the current plot device identifier, unit number and access type. If no device is set, a blank is returned. DEVICE has traditionally been a 2 character name, but may now contain up to 12 characters.

GQDEV (DEVICE, IUNIT, IATYP, IRET)

Output parameters:

DEVICE	CHAR*	Plot device name
IUNIT	INTEGER	Not used
IATYP	INTEGER	Device access type 1 = direct access 2 = sequential access
IRET	INTEGER	Return code

GQGGRF - Query Grid On Graph Projection

This subroutine returns the current coordinate system definition for a grid which is evenly spaced in a graph coordinate system. The grid coordinate system is defined by GSGGRF.

GQGGRF (IXTYP, IYTYP, KX, KY, XLL, YLL, XUR, YUR, IRET)

Output parameters:

IXTYP	INTEGER	X coordinate type 1 = linear 2 = logarithmic 3 = ** KAPPA (2/7) 5 = polar (R)
IYTYP	INTEGER	Y coordinate type 1 = linear 2 = logarithmic 3 = ** KAPPA (2/7) 5 = polar (THETA)
KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
XLL	REAL	Lower left X value
YLL	REAL	Lower left Y value
XUR	REAL	Upper right X value
YUR	REAL	Upper right Y value
IRET	INTEGER	Return code

GQGMAP - Query Grid On Simple Map Projection

This subroutine returns the current definition for a grid evenly spaced in a simple map projection. The grid can be defined by GSGMAP.

GQGMAP (PROJ, KX, KY, DLATLL, DLONLL, DLATUR, DLONUR, IRET)

Output parameters:

PROJ	CHAR*	Map projection name
KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude
IRET	INTEGER	Return code

GQGMGN - Query Graph Margins

This subroutine returns the current margins for the graph mode of the map/graph coordinate system. The value returned is that originally specified as a fraction of the view region or a multiple of the text size; subsequent changes in view region or text size do not affect the value.

GQGMGN (XL, YB, XR, YT, IRET)

Output parameters:

XL	REAL	Left margin size
YB	REAL	Bottom margin size
XR	REAL	Right margin size
YT	REAL	Top margin size
IRET	INTEGER	Return code

GQGPRJ - Query Grid On Map Projection

This subroutine returns the current coordinate system definition for a grid evenly spaced on a general map projection. The grid can be defined by GSGPRJ or GSGMAP.

```
GQGPRJ ( PROJ, ANGLE1, ANGLE2, ANGLE3, KX, KY, DLATLL, DLONLL,
         DLATUR, DLONUR, IRET )
```

Output parameters:

PROJ	CHAR*	Map projection name
ANGLE1	REAL	Reference angle 1
ANGLE2	REAL	Reference angle 2
ANGLE3	REAL	Reference angle 3
KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude
IRET	INTEGER	Return code

GQGRAF - Query Graph Coordinates

This subroutine returns the current coordinate system definition for the graph plotting mode of the map/graph coordinate system.

GQGRAF (IXTYP, IYTYP, YSZXSZ, XLM, YBM, XRM, YTM, IRET)

Output parameters:

IXTYP	INTEGER	X coordinate type 1 = linear 2 = logarithmic 3 = ** kappa (2 / 7) 4 = skew 5 = polar (R)
IYTYP	INTEGER	Y coordinate type 1 = linear 2 = logarithmic 3 = ** kappa (2 / 7) 5 = polar (THETA)
YSZXSZ	REAL	Height to width ratio of plot
XLM	REAL	Left limit of X axis
YBM	REAL	Bottom limit of Y axis
XRM	REAL	Right limit of X axis
YTM	REAL	Top limit of Y axis
IRET	INTEGER	Return code

GQLINE - Query Line Attributes

This subroutine returns the current line attributes including the line type number, the software/hardware line type flag, the line width size multiplier and the software/hardware line width flag.

GQLINE (ILTYP, ILTHW, IWIDTH, IWHW, IRET)

Output parameters:

ILTYP	INTEGER	Line type number 0 = no change
ILTHW	INTEGER	Line type flag 1 = software 2 = hardware
IWIDTH	INTEGER	Line width
IWHW	INTEGER	Line width flag 1 = software 2 = hardware
IRET	INTEGER	Return code

GQLPAT - Query Line Pattern

This subroutine returns the software line pattern for the current line type.

GQLPAT (ILPAT, IRET)

Output parameters:

ILPAT (8)	INTEGER	Line pattern values
IRET	INTEGER	Return code

GQMFIL - Query Map File

This subroutine returns the current map file name to be used by GDRMAP to draw a map.

GQMFIL (MAPNAM, IRET)

Output parameters:

MAPNAM	CHAR*	Map file name
IRET	INTEGER	Return code

GQMMAP - Query Simple Map Projection

This subroutine returns the current simple map projection defined by GSMMAP.

GQMMAP (PROJ, DLATLL, DLONLL, DLATUR, DLONUR, IRET)

Output parameters:

PROJ	CHAR*	Map projection name
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude
IRET	INTEGER	Return code

GQMMGN - Query Map Margins

This subroutine returns the current margins used in map mode. The value returned is that originally specified as a fraction of view region or text size multiple; subsequent changes in view region or text size do not affect the margin size.

GQMMGN (XL, YB, XR, YT, IRET)

Output parameters:

XL	REAL	Left margin size
YB	REAL	Bottom margin size
XR	REAL	Right margin size
YT	REAL	Top margin size
IRET	INTEGER	Return code

GQMODE - Query Map/graph Mode

This subroutine returns the current mode for map/graph coordinate plotting.

GQMODE (MODE, IRET)

Output parameters:

MODE	INTEGER	Plotting mode 0 = no change 1 = map coordinates 2 = graph coordinates
IRET	INTEGER	Return code

GQMPRJ - Query Full Map Projection

This subroutine returns the current map projection and bounds, which were defined by GSMPRJ or GSMMAP.

```
GQMPRJ ( PROJ, ANGLE1, ANGLE2, ANGLE3, DLATLL, DLONLL,
         DLATUR, DLONUR, IRET )
```

Output parameters:

PROJ	CHAR*	Map projection name
ANGLE1	REAL	Reference angle 1
ANGLE2	REAL	Reference angle 2
ANGLE3	REAL	Reference angle 3
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude
IRET	INTEGER	Return code

GQMRKR - Query Marker Attributes

This subroutine returns the current marker attributes including the marker number, the hardware/software flag and the marker size and line width.

NOTE THAT THE CALLING SEQUENCE OF THIS SUBOUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GQMRKR (IMARK, IMKHW, SZMARK, IMKWID, IMKWID, IRET)

Output parameters:

IMARK	INTEGER	Marker number
IMKHW	INTEGER	Hardware/software flag 1 = software 2 = hardware
SZMARK	REAL	Marker size multiplier
IMKWID	INTEGER	Marker width multiplier
IRET	INTEGER	Return code

GQPTND - Query Pressure Tendency Attributes

This subroutine returns the current pressure tendency symbol size and line width.

Note that this subroutine in not currently implemented.

GQPTND (SZPTND, IPTWID, IRET)

Output parameters:

SZPTND	REAL	Pressure tendency symbol size
IPTWID	INTEGER	Pressure tendency line width
IRET	INTEGER	Return code

GQPWTH - Query Past Weather Attributes

This subroutine returns the past weather symbol size and line width.

Note that this subroutine is not currently implemented.

GQPWTH (SZPWTH, IPWWID, IRET)

Output parameters:

SZPWTH	REAL	Past weather symbol size
IPWWID	INTEGER	Past weather symbol width
IRET	INTEGER	Return code

GQSATN - Query Satellite Navigation

This subroutine returns the current satellite navigation information.

GQSATN (NAVTYP, IMGNAM, IRET)

Output parameters:

NAVTYP	CHAR*	Satellite navigation type 'AOIPS', 'NPGS'
IMGNAM	CHAR*	Satellite image name
IRET	INTEGER	Return code

GQSKY - Query Sky Coverage Attributes

This subroutine returns the current sky coverage symbol size, line width, and sky symbol type.

Note that this subroutine is not currently implemented.

GQSKY (SZSKY, ISKTYP, ISKWID, IRET)

Output parameters:

SZSKY	REAL	Sky coverage size multiplier
ISKTYP	INTEGER	Sky coverage symbol type 1 = not filled in 2 = filled in
ISKWID	INTEGER	Sky coverage line width
IRET	INTEGER	Return code

GQSYSZ - Query Attribute Sizes

This subroutine returns the current size of text, markers and wind barbs in terms of normalized device coordinates.

GQSYSZ (RXSZMK, RYSZMK, RXSZTX, RYSZTX, RXSZWB, RYSZWB, IRET)

Output parameters:

RXSZMK	REAL	Width of markers
RYXZMK	REAL	Height of markers
RXSZTX	REAL	Width of text characters
RYSZTX	REAL	Height of text characters
RXSZWB	REAL	Length of wind barbs if oriented along x axis
RYSZWB	REAL	Length of wind barbs if oriented along y axis
IRET	INTEGER	Return code

GQTEXT - Query Text Attributes

This subroutine returns the text attributes including the font number, the text software/hardware flag, and the text size/width multipliers.

NOTE THAT THE CALLING SEQUENCE FOR THIS SUBROUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GQTEXT (ITXFN, ITXHW, SZTEXT, ITXWID, IRET)

Output parameters:

ITXFN	INTEGER	Font number
ITXHW	INTEGER	Software/hardware flag 1 = software 2 = hardware
SZTEXT	REAL	Text size multiplier
ITXWID	INTEGER	Text line width
IRET	INTEGER	Return code

GQVIEW - Query View Region

This subroutine returns the current view region boundaries.

GQVIEW (XLLF, YLLF, XURF, YURF, IRET)

Output parameters:

XLLF	REAL	Lower left x fraction
YLLF	REAL	Lower left y fraction
XURF	REAL	Upper right x fraction
YURF	REAL	Upper right y fraction
IRET	INTEGER	Return code

GQWTHR - Query Weather Attributes

This subroutine returns the weather symbol size multilplier and width.

GQWTHR (SZWTHR, IWIWID, IRET)

Output parameters:

SZWTHR	REAL	Weather code size multiplier
IWIWID	INTEGER	Weather code line width
IRET	INTEGER	Return code

GREST - Restore Graphics From A File

This subroutine restores graphics from a file. The file must have been created using GSAVE.

GREST (FILNAM, IRET)

Input parameters:

FILNAM	CHAR*	File name

Output parameters:

IRET	INTEGER	Return code

GSARRW - Set Wind Arrow Attributes

This subroutine sets the wind arrow size, arrow head size, line width, and arrow type. If these parameters are not positive, no changes are made.

NOTE THAT THE CALLING SEQUENCE OF THIS SUBROUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GSARRW (SZARRW, SZARRH, IARWID, IARTYP, IRET)

Input parameters:

SZARRW	REAL	Arrow size multiplier <=0 = no change
SZARRH	REAL	Arrow head size multiplier <=0 = no change
IARWID	INTEGER	Arrow width multiplier <=0 = no change
IARTYP	INTEGER	Arrow type <=0 = no change 1 = plot wind for calm wind 2 = don't plot for calm wind

Output parameters:

IRET	INTEGER	Return code

GSATAO - Select AOIPS Satellite Navigation

This subroutine selects AOIPS-formatted GOES satellite navigation. The navigation information will be read from the AOIPS image and navigation files. The navigation file must reside in the same subdirectory as the image.

Since GEMPLT will center the plot area in the current view region, the image must be centered in the current view region in order for the overlay capabilities to work properly. GSVIEW can be used to modify the view region.

The subroutine GCLDHT can be used to set the cloud height. GSATAO resets the height to 0.

GSATAO (IMGNAM, IRET)

Input parameters:

IMGNAM	CHAR*	Satellite image name

Output parameters:

IRET	INTEGER	Return code

GSATNP - Select NPS Satellite Navigation

This subroutine selects satellite navigation developed by the Naval Postgraduate School. Files IMGNAM.LAT and IMGNAM.LON contain the required navigation information.

GSATNP (IMGNAM, IRET)

Input parameters:

IMGNAM	CHAR*	Satellite image name

Output parameters:

IRET	INTEGER	Return code

GSAVE - Save Graphics In A File

This subroutine saves all the graphics planes in a file. The graphics may be restored to the device using GREST. This subroutine can be used only on devices which have a readback capability.

GSAVE (FILNAM, IRET)

Input parameters:

FILNAM	CHAR*	File name

Output parameters:

IRET	INTEGER	Return code

GSBARB - Set Wind Barb Attributes

This subroutine sets the wind barb size, line width, and barb type. If these parameters are not positive, no changes are made.

NOTE THAT THE CALLING SEQUENCE FOR THIS SUBOUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GSBARB (SZBARB, IBRWID, IBRTYP, IRET)

Input parameters:

SZBARB	REAL	Barb size multiplier <=0 = no change
IBRWID	INTEGER	Barb width multipier <=0 = no change
IBRTYP	INTEGER	Barb type <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GSCHLS - Set Color Components By HLS

This subroutine defines the color corresponding to a color number by specifying values of hue, lightness and saturation. These values must be in the range 0 - 1.

GSCHLS (ICOLR, HUE, CLIGHT, SAT, IRET)

Input parameters:

ICOLR	INTEGER	Color number
HUE	REAL	Hue value
CLIGHT	REAL	Lightness value
SAT	REAL	Saturation value

Output parameters:

IRET	INTEGER	Return code

GSCINT - Initialize Colors To Defaults

This subroutine initializes the colors on the current graphics device. Each device has its own default colors.

GSCINT (IRET)

Output parameters:

IRET	INTEGER	Return code

GSCNAM - Set Color By Name

This subroutine defines the color corresponding to a color number by specifying a color name. Only the first three letters of the name are checked. The currenly defined colors are listed below:

BLAck,	GREy,	WHIte,	GREen,	AVOcado,	LTGreen,	DKGreen,
CYAn,	BLUe,	AQUa,	SKY,	LTBlue,	DKBlue,	NAVy,
INDigo,	VIOlet,	PURple,	RED,	MARoon,	PINk,	DKPink,
MAGenta,	YELlow,	LEMon,	GOLd,	ORAnge,	APRicot,	DKOrange,
BROwn,	GBRown,	MUD,	TAN,	BEIge,	BLOnd,	SANd,
VANilla						

GSCNAM (ICOLR, COLOR, IRET)

Input parameters:

ICOLR	INTEGER	Color number
COLOR	CHAR*	Color name

Output parameters:

IRET	INTEGER	Return code

GSCOLR - Set Color Number

This subroutine sets the color number. Color numbers larger than the number of valid colors will be converted, via modular arithmetic, to a valid color number. If the color number is negative or zero, no change will be made.

GSCOLR (ICOLR, IRET)

Input parameters:

ICOLR	INTEGER	Color number <= 0 = no change

Output parameters:

IRET	INTEGER	Return code

GSCRGB - Set Color Components By RGB

This subroutine defines the color corresponding to a color number by specifying the values of the red, green, and blue color components. The color components must be in the range 0 - 1.

GSCRGB (ICOLR, RED, GREEN, BLUE, IRET)

Input parameters:

ICOLR	INTEGER	Color number
RED	REAL	Red component value
GREEN	REAL	Green component value
BLUE	REAL	Blue component value

Output parameters:

IRET	INTEGER	Return code

GSCTYP - Set Cloud Type Attributes

This subroutine sets the cloud type symbol attributes. If these parameters are not positive, no changes are made.

Note that this subroutine is not currently implemented.

GSCTYP (SZCTYP, ICTWID, IRET)

Input parameters:

SZCTYP	REAL	Cloud type symbol size
ICTWID	INTEGER	Cloud type line width <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GSDEV - Set Plot Device

This subroutine sets the plot device to be used by GEMPLT. If another device is in use when it is called, GSDEV terminates plotting on that device and starts the device subprocess for the requested device. DEVICE has traditionally been a 2-character name, but may now contain up to 12 characters.

GSDEV (DEVICE, IUNIT, IRET)

Input parameters:

DEVICE	CHAR*	Plot device name
IUNIT	INTEGER	Not used

Output parameters:

IRET	INTEGER	Return code

GSGGRF - Set Grid On Graph Projection

This subroutine defines the coordinate system for a grid which is evenly spaced in a graph coordinate system. If the grid is defined in a polar coordinates system, the grid rows correspond to constant THETA values; the grid columns correspond to constant values of R. XLL, YLL, XUR and YUR correspond to the min R, min THETA, max R and max THETA. XLL must be greater than or equal to 0.

GSGGRF (IXTYP, IYTYP, KX, KY, XLL, YLL, XUR, YUR, IRET)

Input parameters:

IXTYP	INTEGER	X coordinate type 1 = linear 2 = logarithmic 3 = ** KAPPA (2/7) 5 = polar (R)
IYTYP	INTEGER	Y coordinate type 1 = linear 2 = logarithmic 3 = ** KAPPA (2/7) 5 = polar (THETA)
KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
XLL	REAL	Lower left x value
YLL	REAL	Lower left y value
XUR	REAL	Upper right x value
YUR	REAL	Upper right y value

Output parameters:

IRET	INTEGER	Return code

GSGMAP - Set Grid On Simple Map Projection

This subroutine defines the coordinate system for a grid which is evenly spaced in a simplified map projection. It is valid for the following map projection types:

CED	Cylindrical equidistant
MCD	Modified cylindrical equidistant
MER	Mercator
NPS	North polar stereographic
SPS	South polar stereographic
LCC	Lambert conic conformal (Northern hemisphere)
SCC	Lambert conic conformal (Southern hemisphere)
NOR	North Polar Orthographic
SOR	South polar Orthographic
UTM	Universal Transverse Mercator

GSGMAP (PROJ, KX, KY, DLATLL, DLONLL, DLATUR, DLONUR, IRET)

Input parameters:

PROJ	CHAR	Map projection name
KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude

Output parameters:

IRET	INTEGER	Return code

GSGMGN - Set Graph Margins

This subroutine sets the margin size to be used in graph mode of the map/graph coordinate system. The margin sizes may be specified as either a fraction of the view region or as a multiple of the current character size. If the values entered are greater than 0 and less than 1, they are considered to be a fraction of the view region. If the values are 1 or greater, they are taken to be multiples of the character size at the time of the call to GSGMGN. The default margin size is zero.

GSGMGN should be called before any plotting is done; margin size should not be changed after plotting has begun. If size is specified in terms of character size, the text size at the time of the call is used. Later changes to the size will not affect the margins.

GSGMGN (XL, YB, XR, YT, IRET)

Input parameters:

XL	REAL	Left margin size
YB	REAL	Bottom margin size
XR	REAL	Right margin size
YT	REAL	Top margin size

Output parameters:

IRET	INTEGER	Return code

GSGPRJ - Set Grid On Map Projection

This subroutine defines the coordinate system for a grid which is evenly spaced on a general map projection. Information about map projections is given in GSMPRJ.

GSGPRJ (PROJ, ANGLE1, ANGLE2, ANGLE3, KX, KY, DLATLL, DLONLL, DLATUR, DLONUR, IRET)

Input parameters:

PROJ	CHAR*	Map projection name
ANGLE1	REAL	Reference angle 1
ANGLE2	REAL	Reference angle 2
ANGLE3	REAL	Reference angle 3
KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude

Output parameters:

IRET	INTEGER	Return code

GSGRAF - Set Graph Coordinates

This subroutine defines a coordinate system for plotting graphs. The X and Y axis coordinate types are specified independently. The user may control the height to width ratio of the plot by setting the parameter YSZXSZ to a positive value. If YSZXSZ is not positive, the plot will fill the available space.

For polar plots, X and Y are the distance and angle (R,THETA) respectively. YSZXSZ is ignored so that R will be equidistant in all directions. A centered plot with radius R may be defined by setting XL = R and YB = XR = YT = 0.

GSGRAF (IXTYP, IYTYP, YSZXSZ, XL, YB, XR, YT, IRET)

Input parameters:

IXTYP	INTEGER	X coordinate type 1 = linear 2 = logarithmic 3 = ** kappa (2/7) 4 = skew 5 = polar (R)
IYTYP	INTEGER	Y coordinate type 1 = linear 2 = logarithmic 3 = ** kappa (2/7) 5 = polar (THETA)
YSZXSZ	REAL	Height to width ratio of plot
XL	REAL	Left limit of X axis
YB	REAL	Bottom limit of Y axis
XR	REAL	Right limit of X axis
YT	REAL	Top limit of Y axis

Output parameters:

IRET	INTEGER	Return code

GSKY - Plot Sky Coverage Symbols

This subroutine draws sky coverage symbols defined in any coordinate system. The sky coverage symbols will be drawn using attributes defined in GSSKY.

Note that this subroutine is not currently implemented.

GSKY (SYS, NP, SKYCD, X, Y, IXOFF, IYOFF, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of sky coverage symbols
SKYCD (NP)	REAL	Sky coverage numeric code
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
IXOFF (NP)	INTEGER	X offsets in half characters
IYOFF (NP)	INTEGER	Y offsets in half characters

Output parameters:

IRET	INTEGER	Return code

GSLINE - Set Line Attributes

This subroutine sets the line attributes including the line type number, the software/hardware line type flag, the line width, and the software/hardware line width flag.

The line type is specified by a two-digit number. The units digit is the dash pattern; the tens digit is the scaling to be applied, with 2 as the default. The line dash patterns are:

1	solid
2	short dash
3	medium dash
4	long dash, short dash
5	long dash
6	long dash, short dash, short dash, short dash
7	long dash, dot
8	medium dash, dot, dot, dot
9	short dash, dot
10	dots

GSLINE (ILTYP, ILTHW, IWIDTH, ILWHW, IRET)

Input parameters:

ILTYP	INTEGER	Line type <=0 = no change
ILTHW	INTEGER	Sw/hw line type flag 1 = software 2 = hardware otherwise no change
IWIDTH	INTEGER	Line width size multiplier <=0 = no change
ILWHW	INTEGER	Sw/hw line width flag 1 = software 2 = hardware otherwise no change

Output parameters:

IRET	INTEGER	Return code

GSLPAT - Set Line Pattern

This subroutine sets the current line pattern to the given pattern. The current line type number will be set to zero. The values specified in the line pattern designate the length of alternating on and off segments. For example, a long-dash short-dash line pattern could be specified by LPAT values of:

10 5 5 5 0 0 0 0

This line pattern will display a line for 10 units, space for 5 units, a line for 5 units, and space for 5 units and repeat this pattern. Dots are specified using negative numbers. The absolute value of the number corresponds to the space in which the dot will be centered.

GSLPAT (LPAT, IRET)

Input parameters:

LPAT (8)	INTEGER	Line pattern values

Output parameters:

IRET	INTEGER	Return code

GSMFIL - Set Map File

This subroutine sets the name of the map file used to draw maps. If the map file cannot be found, the subroutine will check for the file in GEMMAPS. If the input file name includes directory information and cannot be found, the subroutine will check for the file name in the standard GEMPAK map file directory. If the file still cannot be found, the default map file will be used.

The map files in GEMMAPS are named by concatenating the resolution, map boundaries and area with the three letter source file type. For example, the medium political world map from GSFC is called MEPOWO.GSF.

RESOLUTION	BOUNDARIES	AREA	SOURCE
HIgh	POlitical	WOrld	GSFc
MEdium	COastline	NW quadrant	WISconsin
LOw	REgional	NE quadrant	
		SE quadrant	
		SW quadrant	
		US	
		WE hemisphere	

GSMFIL (MAPFIL, IRET)

Input parameters:

MAPFIL	CHAR*	Map file name

Output parameters:

IRET	INTEGER	Return code

GSMMAP - Set Simple Map Projection

This subroutine provides a simplified call for defining the map projection and bounds used for plotting in map coordinates. The angles necessary for defining these projections are based on the specified bounds. The following projections are valid:

CED	Cylindrical equidistant
MCD	Modified cylindrical equidistant
MER	Mercator
NPS	North polar stereographic
SPS	South polar stereographic
LCC	Lambert conic conformal (Northern hemisphere)
SCC	Lambert conic conformal (Southern hemisphere)
NOR	North Polar Orthographic
SOR	South polar Orthographic
UTM	Universal Transverse Mercator

If the projection name is DEF, the current map or satellite projection will be used. In this case, the bounds specified will not be used.

GSMMAP (PROJ, DLATLL, DLONLL, DLATUR, DLONUR, IRET)

Input parameters:

PROJ	CHAR*	Map projection
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude

Output parameters:

IRET	INTEGER	Return code

GSMMGN - Set Map Margins

This subroutine sets the margin size to be used in map mode. The margin sizes may be specified as either a fraction of the view region or as a multiple of the current character size. If the values entered are greater than 0.0 and less than 1.0, they are considered to be a fraction of the view region. If the values are 1.0 or greater, they are taken to be multiples of the character size at the time of the call to GSMMGN. The default margin size is zero.

GSMMGN should be called before any plotting is done; margin size should not be changed after plotting has begun. If size is specified in terms of character size, the size at the time of the call is used. Later changes to the text size will not affect the margins.

GSGMGN (XL, YB, XR, YT, IRET)

Input parameters:

XL	REAL	Left margin size
YB	REAL	Bottom margin size
XR	REAL	Right margin size
YT	REAL	Top margin size

Output parameters:

IRET	INTEGER	Return code

GSMODE - Set Map/graph Mode

This subroutine sets the ploting mode for map/graph coordinate plotting. It may be used to change mode within programs.

GSMODE (MODE, IRET)

Input parameters:

MODE	INTEGER	Plotting mode 0 = no change 1 = map coordinates 2 = graph coordinates

Output parameters:

IRET	INTEGER	Return code

GSMPRJ - Set Full Map Projection

This subroutine provides a general way to specify the map projection and bounds to be used when plotting in map coordinates. The subroutine now uses a character projection name rather than map class and projection numbers. The valid projection names are:

CLASS	PROJ	PROJECTION
Cylindrical	CED	Cylindrical equidistant
	MER	Mercator
	MCD	Modified cylind. equidistant
Azimuthal	AED	Azimuthal equidistant
	STR	Stereographic
	ORT	Orthographic
	LEA	Lambert equal area
	GNO	Gnomonic
Conical	LCC	North Lambert conic conformal
	SCC	South Lambert conic conformal
Oblique Mercator	TVM	Transverse Mercator
	UTM	Universal Transverse Mercator

For the UTM projection, the tangential longitude is adjusted to a standard longitude by rounding down to ..., -9, -3, 3, 9, ...

The angles are defined for the various map classes.

Cylindrical	No angles needed; polon is set to midway between DLONLL and DLONUR.
Azimuthal	angle1 -- polat (the pole latitude; must be +90. or -90.) angle2 -- polon (the central longitude) angle3 -- not used
Conical	angle1 -- latitude 1 angle2 -- polon (the central longitude) angle3 -- latitude 2
Oblique	angle1 -- tangential longitude angle2 -- not used angle3 -- not used

Azimuthal and conic projections which include the pole, can be defined by specifying the lower left and upper right corners. If DLATLL and DLATUR are equal and DLONLL and DLONUR are also equal, a map area will be defined which includes the area from the pole

to the given latitude in each direction with DLONLL as the central longitude.

GSMPRJ (PROJ, ANGLE1, ANGLE2, ANGLE3, DLATLL, DLONLL, DLATUR, DLATUR, DLONUR, IRET)

Input parameters:

PROJ	CHAR*	Map projection name
ANGLE1	REAL	Reference angle 1
ANGLE2	REAL	Reference angle 2
ANGLE3	REAL	Reference angle 3
DLATLL	REAL	Lower left latitude
DLONLL	REAL	Lower left longitude
DLATUR	REAL	Upper right latitude
DLONUR	REAL	Upper right longitude

Output parameters:

IRET	INTEGER	Return code

GSMRKR - Set Marker Attributes

This subroutine sets the marker attributes including the marker number, the hardware/software flag and the marker size/width multipliers.

NOTE THAT THE CALLING SEQUENCE FOR THIS SUBROUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GSMRKR (IMARK, IMHMW, SZMARK, IMKWID, IRET)

Input parameters:

IMARK	INTEGER	Marker number <=0 = no change
IMMHW	INTEGER	Sw/hw marker flag 1 = software 2 = hardware
SZMARK	REAL	Marker size multiplier
IMKWID	INTEGER	Marker line width <= 0 = no change

Output parameters:

IRET	INTEGER	Return code

GSPTND - Set Pressure Tendency Attributes

This subroutine sets the pressure tendency symbols size and line width.

Note that this subroutine is not currently implemented.

GSPTND (SZPTND, IPTWID, IRET)

Input parameters:

SZPTND	REAL	Pressure tendency symbol size <=0 = no change
IPTWID	INTEGER	Pressure tendency line width <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GSPWTH - Set Past Weather Attributes

This subroutine sets the past weather symbol size and line width.

Note that this subroutine is not currently implemented.

GSPWTH (SZPWTH, IPWWID, IRET)

Input parameters:

SZPWTH	REAL	Past weather symbol size <=0 = no change
IPWWID	INTEGER	Past weather line width <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GSSKY - Set Sky Coverage Attributes

This subroutine sets the sky coverage symbol size, line width and the symbol type.

Note that this subroutine is not currently implemented.

GSSKY (SZSKY, ISKTYP, ISKWID, IRET)

Input parameters:

SZSKY	REAL	Sky coverage size multiplier <=0 = no change
ISKTYP	INTEGER	Sky coverage symbol type 1 = not filled in 2 = filled in
ISKWID	INTEGER	Sky coverage line width <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GSTEXT - Set Text Attributes

This subroutine sets the text attributes including the font number, the text software/hardware flag, and the text size/width multipliers.

NOTE THAT THE CALLING SEQUENCE OF THIS SUBROUTINE HAS CHANGED FROM PREVIOUS VERSIONS OF GEMPAK.

GSTEXT (ITXFN, ITXHW, SZTEXT, ITXWID, IRET)

Input parameters:

ITXFN	INTEGER	Text font <=0 = no change
ITXHW	INTEGER	Text sw/hw flag 1 = software 2 = hardware otherwise no change
SZTEXT	REAL	Text size multiplier <=0 = no change
ITXWID	INTEGER	Text line width <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GSTICK - Set Tick Attributes

This subroutine sets the tick attributes including the tick type and size.

GSTICK (ITKTYP, SZTICK, IRET)

Input parameters:

ITKTYP	INTEGER	Tick type 0 = no change 1 = major tick outside 2 = major tick inside 3 = minor tick outside 4 = minor tick inside 5 = major tick out and inside 6 = minor tick out and inside
SZTICK	REAL	Tick size multiplier

Output parameters:

IRET	INTEGER	Return code

GSTRML - Draw Streamlines

This subroutine draws streamlines through a gridded vector field. Streamlines will be computed for a subgrid with the lower left corner at (IMINX,JMINY) and the upper right corner at (IMAXX,JMAXY). The maximum number of points in the subgrid is 8000. A grid coordinate transformation and a map / graph coordinate transformation must be defined before GSTRML is called. The current line and arrow attributes apply to the streamlines. This subroutine is an adaptation of the NCAR streamline subroutine STRMLN.

GSTRML (KX, KY, U, V, IMINX, JMINY, IMAXX, JMAXY, MISFLG, IRET)

Input parameters:

KX	INTEGER	Number of x grid points
KY	INTEGER	Number of y grid points
U (KX,KY)	REAL	U - component grid
V (KX,KY)	REAL	V - component grid
IMINX	INTEGER	First x point of subgrid
JMINY	INTEGER	First y point of subgrid
IMAXX	INTEGER	Last x point of subgrid
JMAXY	INTEGER	Last y point of subgrid
MISFLG	LOGICAL	Interpolate missing data flag

Output parameters:

IRET	INTEGER	Return code

GSVIEW - Set View Region

This subroutine sets the boundaries of the view region to be used to display the plot. The view region is specified using fractions of the available area on the plot device. The point (0., 0.) is the lower left corner of the device; (1., 1.) is the upper right corner of the device. For example:

```
CALL GSVIEW ( .5, .5, 1., 1., iret )
```

will display plots in the upper right quadrant of the device.

Note that the fractions describing the view region are not equal to coordinate values, except for a square device.

GSVIEW (XLLF, YLLF, XURF, YURF, IRET)

Input parameters:

XLLF	REAL	Lower left x fraction
YLLF	REAL	Lower left y fraction
XURF	REAL	Upper right x fraction
YURF	REAL	Upper right y fraction

Output parameters:

IRET	INTEGER	Return code

GSWTHR - Set Weather Symbol Attributes

This subroutine sets the weather symbol size and line width.

GSWTHR (SZWTHR, IWTWID, IRET)

Input parameters:

SZWTHR	REAL	Weather symbol size <=0 = no change
IWTWID	INTEGER	Weather symbol line width <=0 = no change

Output parameters:

IRET	INTEGER	Return code

GTEXT - Plot Text

This subroutine plots text starting at a point defined in any coordinate system. The point determines the center of the first character. The text string may be rotated from horizontal at the reference point and offset along the rotated X and Y coordinates. Positive X offsets are toward the right; positive Y offsets are toward the top. The text will be drawn using attributes defined in GSTEXT.

GTEXT (SYS, XPT, YPT, CCHAR, ROTAT, IXOFF, IYOFF, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
XPT	REAL	X coordinate / latitude
YPT	REAL	Y coordinate / longitude
CCHAR	CHAR*	Text
ROTAT	REAL	Rotation angle in degrees
IXOFF	INTEGER	X offset in half characters
IYOFF	INTEGER	Y offset in half characters

Output parameters:

IRET	INTEGER	Return code

GTRANS - Transform Points To New Coordinates

This subroutine transforms an array of points in any input coordinate system into the specified output coordinate system. In the 'M' coordinate system, X represents latitude, and Y represents longitude.

GTRANS (SYSIN, SYSOUT, NP, XIN, YIN, XOUT, YOUT, IRET)

Input parameters:

SYSIN	CHAR*	Input coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
SYSOUT	CHAR*	Output coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of points
XIN (NP)	REAL	X input coordinates/latitudes
YIN (NP)	REAL	Y input coordinates/longitudes

Output parameters:

XOUT (NP)	REAL	X output coordinates/latitudes
YOUT (NP)	REAL	Y output coordinates/longitudes
IRET	INTEGER	Return code

GWTHR - Plot Weather Symbols

This subroutine draws weather symbols defined in any coordinate system. The weather symbols will be drawn using attributes defined in GSWTHR.

GWTHR (SYS, NP, WTCOD, X, Y, IXOFF, IYOFF, IRET)

Input parameters:

SYS	CHAR*	Coordinate system 'D' = device coordinates 'N' = normalized coordinates 'V' = view coordinates 'P' = plot coordinates 'M' = map coordinates 'G' = grid coordinates
NP	INTEGER	Number of weather symbols
WTCOD (NP)	REAL	Weather numeric code
X (NP)	REAL	X coordinates / latitudes
Y (NP)	REAL	Y coordinates / longitudes
IXOFF (NP)	INTEGER	X offsets in half characters
IYOFF (NP)	INTEGER	Y offsets in half characters

Output parameters:

IRET	INTEGER	Return code

CHAPTER 4

WRITING DEVICE DRIVERS

This section describes the procedure for writing a new GEMPLT device driver. It assumes the programmer will use the dummy device driver, GN. In many cases, however, it will be easier to start with an existing device driver written for a device which is similar to the programmer's device. For example, the BL device driver is a pen plotter driver which writes output to an intermediate plot file and can be used to create a driver for different brand pen plotter.

To begin, create a device driver subdirectory and copy all the files in the GN library into it. Delete the .OLB file. Rename GN.COM to XX.COM and GN.SUB to XX.SUB where XX is the two letter identifier for the driver to be written. Edit these two files, replacing references to GN with XX.

All of the FORTRAN subroutines must be rewritten for the specific device. If certain hardware capabilities are not available, the subroutines need not be changed from the GN versions. A guide to changing these subroutines follows.

INITIALIZATION AND TERMINATION SUBROUTINES

HINIT - HINIT defines the characteristics of the device to GEMPLT. Each of the assignment statements should be edited. This subroutine is called once when the device driver is installed.

HINITD - HINITD contains device-dependent initialization values. It is called once when the device driver is installed. Intermediate plot file names, a flag indicating that the file is not open, hardware sizes in device units and other device-specific parameters are assigned here and saved in DEVICE.CMN. A typical DEVICE.CMN is included in each device driver. Any specific initialization

commands for the graphics device are included here.

HINITP - HINITP is called each time the user calls GINITP at the start of each program. Generally, this subroutine does nothing. Occasionally, some device-dependent functions must be performed at the start of each program in addition to functions performed when the device driver is started.

HOPEN - HOPEN is used, when necessary, to open intermediate plot files. The other H subroutines should check the open file flag when they are ready to add plotting information to the file. If the file is not open, this subroutine is called. If no intermediate plot file is to be used, the subroutine will do nothing.

HENDD - HENDD is called whenever GENDP is called or a new device is requested. Unless some special functions are to be performed at the end of each applications program, this subroutine will not do anything if IEOP = 0, i.e., the device driver task is to be retained. When IEOP = 1, the device driver is requested to exit and some special action may have to be taken.

REQUIRED SUBROUTINE TO DRAW LINES

HLINE - HLINE must be written in each device driver. This is the subroutine that actually draws lines on the graphics device. If it is necessary to rasterize the lines being drawn, both the II and LA device drivers provide examples of such code.

DEVICE CONTROL SUBROUTINES

HEPLOT - HEPLOT can be used to return a terminal from graphics to text mode.

HFLUSH - HFLUSH will cause any buffers internal to the device driver process to be flushed. These are buffers which are filled by H level subroutines.

HCLEAR - HCLEAR causes the screen on direct access devices to be cleared. On continuous feed paper devices, a form feed should be issued. On single sheet paper devices, the plotter should pause for the user to load a new piece of paper.

HCLOSP - HCLOSP closes the intermediate plot file, if any, and

resets the open flag to false. A new file should not be opened until plotting commands are received.

HOUTP - HOUTP sends output stored in an intermediate plot file to the device.

WRITE HARDWARE SYMBOLS

HMARK - HMARK is called to write hardware markers to the device. If hardware markers are not to be implemented, this subroutine will do nothing.

HTEXT - HTEXT is called to write hardware text to the device. If hardware text is not to be implemented, this subroutine will do nothing.

SET HARDWARE ATTRIBUTES

HSLTYP - HSLTYP saves the necessary information in DRIVER.CMN to use hardware line types. If hardware line types are not to be implemented, this subroutine does nothing.

HSLWID - HSLWID sets the hardware graphics line width. If hardware line width is not implemented, this subroutine does nothing.

HSMRKR - HSMRKR saves the necessary information in DRIVER.CMN to use hardware markers. If hardware markers are not implemented, this subroutine does nothing.

HSTEXT - HSTEXT saves the necessary information in DRIVER.CMN to use hardware text. If hardware text is not implemented, this subroutine does nothing.

COLOR SUBROUTINES

HSCOLR - HSCOLR sets a new color number on color devices. On a monochromatic device, this subroutine does nothing.

HSCRGB - HSCRGB sets the color components of a graphics plane using red, green and blue color components. These values will be in the range 0 - 1, and may need to be scaled to a different range.

HSCINT - HSCINT initializes the colors on a color graphics device to some default colors. It is used only on color devices on which color components can be changed. It should specify a default color for each number.

GRAPHICS SAVE/RESTORE FUNCTIONS

HSAVE - HSAVE saves the graphics planes in a file using the readback function of the terminal, if available.

HREST - HREST restores the graphics planes to the terminal from a file saved using HSAVE.

After all the H level subroutines have been written, they should be compiled and added to an object library called XX.OLB. The command procedure BLDLIB may be used to compile the subroutines and add them to the library. Enter the following command from DCL:

```
BLDLIB
```

Answer the questions as follows:

```
[]
XX
```

After the library is created, link the device driver using the command:

```
@XX
```

The command file will link the H-level subroutines just written with general device driver code (the D-level subroutines) which are common to all devices. Note that control of the device driver and all the software functions, along with line clipping, are performed in the D-level subroutines. None of these subroutines should have to be modified for a particular device.

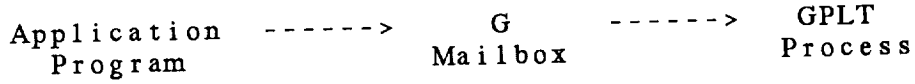
The executable module produced by @XX will be put in GEMEXE, which is where all the GEMPLT executable files are found. The easiest way to test the device driver is to run the test program, ATEST . This program allows the user to enter GEMPLT subroutine names, executing the functions one at a time. Set the device to XX using GSDEV and test each function individually, by drawing lines, markers, etc. Don't forget that GPLT will buffer many plotting commands. Call GEPLOT to send output to the device.

APPENDIX A

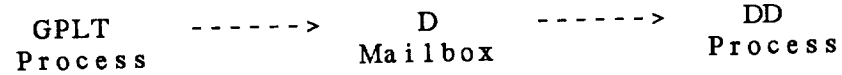
SOFTWARE LIBRARIES

The GEMPLT software commands are executed in two subprocesses. The first process, GPLT, is created the first time GINITP is called during a session. The second process is a device driver, DD, which is created when GSDEV is called. These processes remain active after the program which initiated them exits and are then used by later programs making GEMPLT calls.

Because the GPLT subprocess is independent of the application program requesting GEMPLT, the calling parameters for each subroutine call are put in a mailbox that is read by the GPLT process. Thus, information is passed in the following steps:



Similarly, GPLT passes requests to the device driver:



The application program must be linked with the GEMPLT object library, GEMOLB:APPL . These routines pass a request through the mailbox to GPLT. GPLT is linked with library GEMOLB:GPLT . Each device driver must be linked with the device code which resides in GEMOLB:DEVICE.

The following example traces a call from an application program to GLINE to draw line segments.

SUBROUTINE	LIBRARY	PROCESS	FUNCTION
GLINE	GEMOLB:APPL	APPLICATION	Pass request to G mailbox
GLINE	GEMOLB:GPLT	GPLT	Transform points to device coordinates
DLINE	GEMOLB:GPLT	GPLT	Pass request to D mailbox
DLINE	GEMOLB:DEVICE	DD	Control clipping, dashing, ...
HLINE	DD	DD	Draw line segments on graphics device

If desired, the application program can be linked with GEMOLB:GPLT . This will eliminate the GPLT subprocess and mailbox, but GPLT will need to be initialized in each program, and information defined in one program will not be remembered in later programs.

Similarly, GEMOLB:DEVICE can replace the calls to the D-level subroutines in GEMOLB:GPLT . In this case only one device can be supported in a linked, executable module.

APPENDIX B

CREATING MAP FILES

GEMPAK MAP FILES

GEMPAK map files are stored as direct access, packed binary files in a GEMPAK Standard Format (GSF). This form was chosen for compactness and relatively rapid access.

In addition, a Sequential Standard Format (SSF) is defined for storing maps as ASCII sequential files. SSF files are used to create new maps, edit existing map files and transfer map files to non-VMS computer systems. Since these are ASCII files, they may be created or changed using a text editor. The SSF files use the same format as the NCAR sequential map file.

In both the GSF and SSF formats, latitude and longitude are given in degrees. West longitude is negative; all longitudes must be between -180 and +180 degrees. No map segment should cross the International Date Line.

FORMAT OF THE SSF FILE:

An SSF file contains map segments, each of which has a series of latitude / longitude points to be connected when the map is drawn. Each segment contains the following information:

NP2	INTEGER	Number of points (np) * 2
RMXLT	REAL	Maximum latitude in segment
RMNLT	REAL	Minimum latitude in segment
RMXLN	REAL	Maximum longitude in segment
RMNLN	REAL	Minimum longitude in segment
PTS (NP2)	REAL	LAT1 LON1 ... LATnp LONnp

These segments are written and read with the FORTRAN format statment:

```
FORMAT  ( I4, 14X, 6F9.3, 8X, / ( 8F9.3, 8X ) )
```

The maximum number of points which may be contained in any segment is 3000.

FORMAT OF THE GSF FILE:

These files are direct access file with a recordsize of 128 words. The data is written in blocks of two records or 256 words. The first block is the file header, followed by data blocks. Map segments are packed into the data blocks; no map segment spans blocks.

The file header has 181 INTEGER*2 words containing:

NMBLK	INTEGER*2	Total number of blocks
LATPTR1	INTEGER*2	Latitude pointer for 90 degrees
LATPTR2	INTEGER*2	Latitude pointer for 89 degrees
...		
LATPTR180	INTEGER*2	Latitude pointer for -89 degrees

The GSF files are sorted by maximum latitude. The LATPTR array contains the first block to use given the maximum latitude on the plot.

These INTEGER*2 values may be changed to INTEGER*4 if required. However, the changes must be made in the programs GSFSSF and SSFGSF described below, as well as in GDRMAP which is the GEMPLT subroutine used to draw maps.

The data blocks contain only 4-byte integer or real words. Each data block contains:

NMSEG	INTEGER	Number of segments
-----	------	Unused
NP1	INTEGER	Number of points in segment 1
RMNLT1	REAL	Minimum latitude
RMNLN1	REAL	Minimum longitude
RMXLT1	REAL	Maximum latitude
RMXLN1	REAL	Maximum longitude
-----	------	Unused
RLAT1 (1)	REAL	Latitude of point 1
RLON1 (1)	REAL	Longitude of point 1
...		
RLAT1 (np1)	REAL	Latitude of point np1
RLON1 (np1)	REAL	Longitude of point np1
NP2	INTEGER	Number of points in segment 2
RMNLT1	REAL	Minimum latitude
...		

CREATING MAP FILES

CONVERTING A GSF TO AN SSF FILE:

The program GSFSSF, which is in GEMMAPS, can be used to convert any GSF file to an SSF file. The user is prompted for the existing GSF file name and a name for the SSF file to be created.

CONVERTING AN SSF FILE TO A GSF FILE:

The program SSFGSF, also in GEMMAPS, can be used to convert an SSF file to a GSF file. The user is prompted for the existing SSF file name and a name for the GSF file to be created. The user can also enter a latitude / longitude window to subset the data. Any points outside the window will be eliminated from the output file. However, no clipping is done. The minimum and maximum latitude and longitude stored with each segment will be recomputed internally, so the values in the SSF file need not be correct. The intermediate files, DAFIL.INT and SEQFIL.INT, are used by the program and may be deleted after completion.

NAMING CONVENTIONS FOR GEMPAK MAP FILES:

The map files are named by concatenating the resolution, map boundaries, and area with the three-letter source file type. For example, the medium-resolution political world map from GSFC is called MEPOWO.GSF

RESOLUTION	BOUNDARIES	AREA	SOURCE
HIgh	POlitical	WOrld	GSFc
MEdium	COastline	NW quadrant	WISconsin
LOw	REgional	NE quadrant	
	CouNty	SE quadrant	
		SW quadrant	
		WEst hemisphere	
		North Hemisphere	
		South Hemisphere	
		US	
		MarylanD	

The maps from Goddard were obtained for GEMPAK Version 1.0. The history of these maps is unknown. The maps from the University of Wisconsin were digitized there.

APPENDIX C

GEMPLT ERROR MESSAGES

Following is a list of the error messages available when calling the GEMPLT error message subroutine. The list includes the error number, the mnemonic code used internally by GEMPLT and the error message.

```
!  This is the GEMPLT error file.
!
!  The file must have the error number in the first four columns.
!  The message to be printed is found after the !.
!
   0       ! NORMAL - Normal.
  -1       ! NMBRER - Mailbox read.
  -2       ! NMBWER - Mailbox write.
  -3       ! NEXQUO - Exceed quota.
  -4       ! NFNCCD - Invalid function code.
  -5       ! NOMFIL - Invalid map file defined -- default used.
  -6       ! NOCORD - Invalid coordinate system.
  -7       ! NOBUFF - G buffer length exceeded.
  -8       ! NDVICE - Device not set.
  -9       ! NDCHAR - No device characteristics.
 -10       ! NIVIEW - Invalid view region.
 -11       ! NOLUN  - Invalid logical unit number.
 -12       ! NIMODE - Invalid mode.
 -13       ! NINVAL - Invalid input parameter.
 -14       ! NOGRAF - No graph defined.
 -15       ! NIPROJ - Invalid projection specified.
 -16       ! NIPBND - Invalid projection bounds.
 -17       ! NIPOSN - Invalid position in projection.
 -18       ! NIGDSZ - Invalid grid size.
 -19       ! NOPFIL - No plot file.
 -20       ! NGINIT - GINITP has not been initialized.
 -21       ! NODEVC - Invalid device selected.
 -22       ! NICOLR - Invalid color name selected.
 -23       ! NOCOLR - Color component cannot be set.
 -24       ! NOCTBL - No color table.
 -25       ! NICNUM - Invalid color number.
```

-26	! NOGFIL	- No graphics file.
-27	! NSATNV	- Satellite navigation not defined.
-28	! NOPNTS	- Too few points to fit curve.
-29	! NIPNTS	- Too many points to fit curve.
-30	! NOMONO	- Points are not monotonic.
-31	! NOIMGF	- Invalid satellite image file.
-32	! NONAVF	- Invalid satellite navigation file.
-33	! NOMAPP	- No plotable points in map file.
-34	! NOBNDS	- No grids points in graphics area.
-35	! NMAPFR	- Map file read error.
-36	! NLBLEX	- Number of label positions exceeds 50.
-37	! NILOGP	- Invalid LOG axis defined.
-101	! NOPROC	- Nonexistent executable.
-102	! NSYSTM	- System service error.
-103	! NSTRER	- String error.
-104	! NNODEV	- Nonexistent device.
-105	! NDEVNA	- Device is not allocated.
-106	! NNOOWN	- Device allocated by another process.

APPENDIX D

CHANGES FROM GEMPAK4 TO GEMPAK5

This appendix documents the changes from GEMPAK4 to GEMPAK5. Some of the major changes include:

- The contouring and streamline packages have been rewritten.
- Improvements have been made to the map and grid transformations when the pole or International Date Line (IDL) is included in the display or grid area.
- Line dashing patterns have been expanded to include scaling and the use of dots.
- Plotting of weather symbols has been added. Subroutines to plot cloud type, sky coverage, pressure tendency and past weather have been defined but not implemented.
- The width for text, wind arrows, barbs, markers and weather symbols can be set. A size parameter for wind arrow heads has been added.
- A county map for the US has been added.
- GDRGRD, which draws latitude / longitude lines, has been improved, especially when plotting across the IDL.
- The tick mark subroutines have been changed and labelling graphs has been improved.
- Spurious lines have been removed from map files.
- A default map file is used if an invalid map file is specified.
- Clipping has been added to hardware text and marker generation.

CHANGES FROM GEMPAK4 TO GEMPAK5

- GKS device driver has been added.

- Additional grid lines are plotted on Skew - T plots.

- A type has been added to wind barb and arrow specification. Setting the arrow type to 2 will prevent plotting of calm (wind speed = 0) winds.

The calling sequences of the following subroutines have been changed from GEMPAK4:

GCONTR	:	MISLBL has been deleted
GQARRW	:	SZARRH, IARWID, IARTYP added
GQBARB	:	IBRWID, IBRTYP added
GSARRW	:	SZARRH, IARWID, IARTYP added
GSBARB	:	IBRWID, IBRTYP added
GSMRKR	:	IMKWID added
GSTEXT	:	ITXWID added
GQLPAT	:	ILTYP is no longer an input

The meanings of some parameters in the calling sequence have been changed:

GAAXIS and GDAXIS : LBFRQ, MTFRQ and LGFRQ refer to the frequency rather than the start and frequency combined.

The following subroutines are new:

GWTHR
GQWTHR
GSWTHR
GSTICK

The following subroutines are new but are not implemented:

GCTYP	GPTND	GPWTH	GSKY
GQCTYP	GQPTND	GQPWTH	GQSKY
GSCTYP	GSPTND	GSPWTH	GSSKY

CHANGES FROM GEMPAK4 TO GEMPAK5

The following subroutine has been deleted:

GSMCON

APPENDIX E

EXAMPLES

This appendix contains a few simple examples of the use of GEMPLT subroutines to create map and graph output. These sample programs do not check for non-zero return codes. Real programs should check for non-zero return codes using GERROR and take the appropriate actions.

EXAMPLE 1: Draw a mercator map of the continental United States.

```
      PROGRAM MAP

      CHARACTER device*2, cprj*3
C-----------------------------------------------------------------------
C*    Initialize GEMPLT.  Set the mode to map coordinates.

      mode = 1
      CALL GINITP ( mode, istat, iret )

C*    Set the device to a VT100 Retro-graphics terminal.

      device = 'VT'
      iunit  = 0

      CALL GSDEV ( device, iunit, iret )

C*    Set the map projection for mercator map of the US.

      cprj  = 'MER'
      dlats =   25.
      dlonw = -125.
      dlatn =   50.
      dlone =  -65.

      CALL GSMMAP ( cprj, dlats, dlonw, dlatn, dlone, iret )

C*    Draw the map.

      CALL GDRMAP ( iret )

C*    End the GEMPLT plotting session.  Do not retain the
C*    GEMPLT subprocess.

      ieop = 1
      CALL GENDP  ( ieop, iret )
C*
C*    Exit the program.

      END
```

EXAMPLE 2: Draw a cartesian graph.

```
      PROGRAM GRAPH

      CHARACTER device*2
      REAL      xaxis (10), yaxis (10), xpts (20), ypts (20)
      LOGICAL   laxis
C-----------------------------------------------------------------
C*    Initialize GEMPLT.  Set the mode to graph coordinates.

      mode = 2
      CALL GINITP ( mode, istat, iret )

C*    Set the device to a Bausch and Lomb pen plotter.  The
C*    intermediate plot file is automatically opened.

      device = 'BL'
      iunit  = 0

      CALL GSDEV ( device, iunit, iret )

C*    Set the graph margins on the bottom to be twice the
C*    height of the character size.

      xl = 0.
      yb = 2.
      xr = 0.
      yt = 0.

      CALL GSGMGN ( xl, yb, xr, yt, iret )
C*
C*    Set the X and Y axis to linear coordinates with limits
C*    along each axis from 0 to 100.  YSZXSZ set to 1. will
C*    make the plotting area square.

      ixtyp  = 1
      iytyp  = 1
      yszxsz = 1.
      xlm    = 0.
      ybm    = 0.
      xrm    = 100.
      ytm    = 100.

      CALL GSGRAF ( ixtyp, iytyp, yszxsz, xlm, ybm, xrm, ytm,
     +              iret )
C
C*    Draw the X axis with labels below the line, intercepting
C*    the Y axis at the point Y = 0., with tick marks and
C*    labeling frequency of every other point.  The labels
C*    will be integer numbers and no grid lines will be drawn.
```

```
            iaxis  = 1
            axpos  = 0.
            laxis  = .true.
            lbstfq = 102
            mtstfq = 102
            lgstfq = 0
            ndec   = 0
            np     = 10

C*          Fill the X axis coordinate array.

            DO i = 1, np
               xaxis (i) = 10. * i
            END DO

            CALL GDAXIS ( iaxis, axpos, laxis, lbstfq, mtstfq,
     +                    lgstfq, ndec, np, xaxis, iret )

C*          Now draw the Y axis with labels to the left of the line.
C*          The Y axis will intercept the X axis at the point X = 0.
C*          Label and tick every point along the Y axis with integer
C*          labels. No grid lines will be drawn.

            iaxis  = 2
            axpos  = 0.
            laxis  = .true.
            lbstfq = 101
            mtstfq = 101
            lgstfq = 0
            ndec   = 0
            np     = 10

C*          Fill the Y axis coordinate array.

            DO i = 1, np
               yaxis (i) = 10. * i
            END DO

            CALL GDAXIS ( iaxis, axpos, laxis, lbstfq, mtstfq,
     +                    lgstfq, ndec, np, yaxis, iret )

C*          Draw the array of points on the graph.
C*          The arrays XPTS and YPTS in a real program would be
C*          filled with the your data.
C*          'M' indicates the x,y points are in units of the map
C*          coordinate system defined by GSGRAF.

            npts = 20
            CALL GLINE  ( 'M', npts, xpts, ypts, iret )
```

EXAMPLES

```
C*          Close the plot file and send it to the Bausch and Lomb
C*          pen plotter.

            CALL GOUTP  ( iret )

C*          End the GEMPLT plotting session.  Do not retain the
C*          GEMPLT subprocess.

            ieop = 1
            CALL GENDP  ( ieop, iret )

C*          Exit the program.

            END
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-4261, Part 2		

4. Title and Subtitle	5. Report Date
GEMPAK5 Part 2--GEMPLT Programmer's Guide Version 5.0	February 1991
	6. Performing Organization Code 973
7. Author(s) Mary L. desJardins, Keith F. Brill, and Steven S. Schotz	8. Performing Organization Report No. 91B00038
	10. Work Unit No.
9. Performing Organization Name and Address Laboratory for Hydrosperic Processes NASA/Goddard Space Flight Center Greenbelt, Maryland 20771	11. Contract or Grant No.
	13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Mary L. desJardins: NASA/GSFC, Greenbelt, MD;
Keith F. Brill: NOAA/NWS/NMC, Washington, DC;
Steven S. Schotz: General Sciences Corporation, Lanham, MD.

16. Abstract

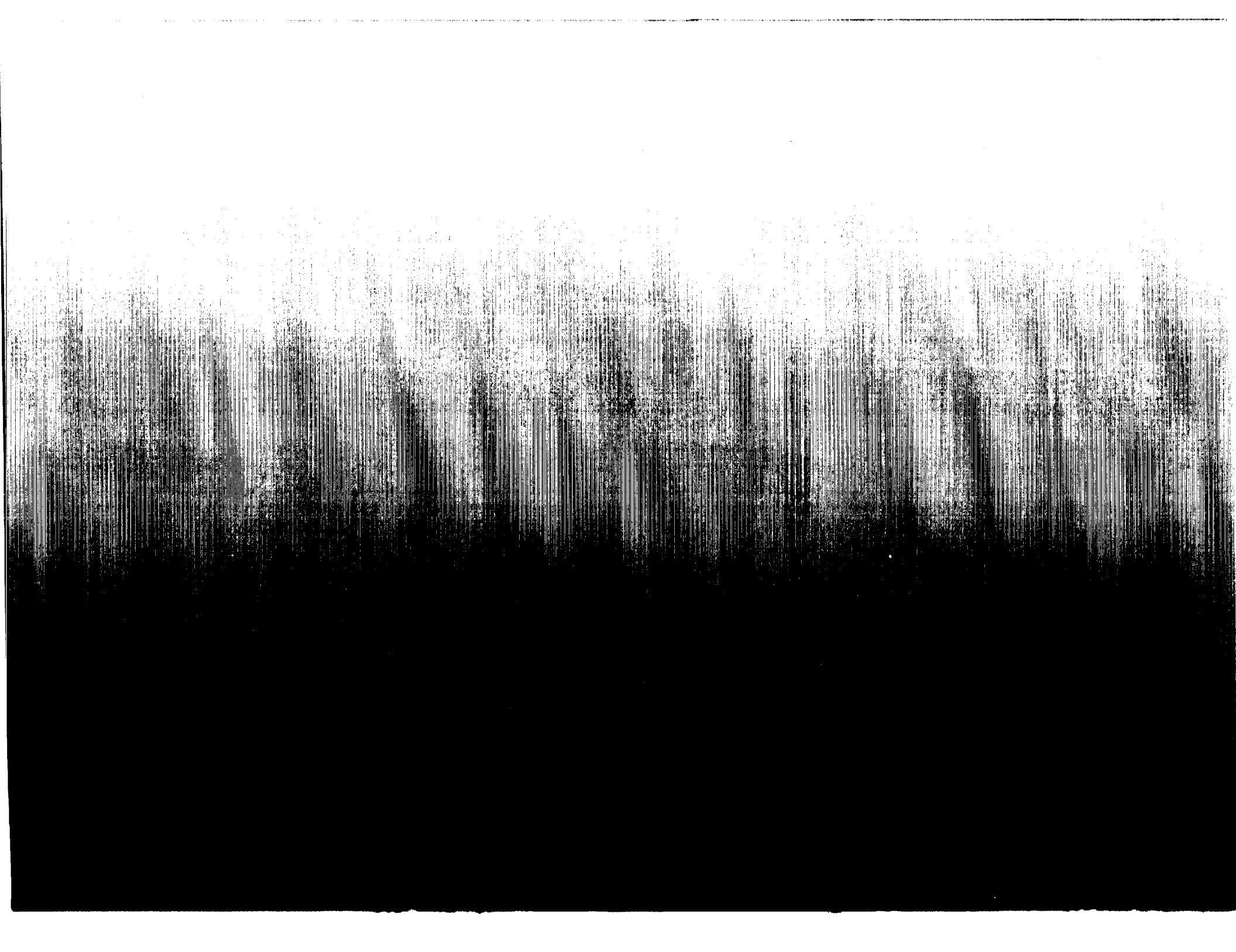
GEMPAK is a general meteorological software package used to analyze and display conventional meteorological data as well as satellite derived parameters. The GEMPLT Programmer's Guide describes the subroutines which can be used in the GEMPAK graphics and transformation subsystem, GEMPLT. (Part 1 contains GEMPAK subroutines.)

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Meteorology Software systems	Unclassified - Unlimited Subject Category 47

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	128	A07

NASA FORM 1626 OCT 86

NASA-Langley, 1991



National Aeronautics and
Space Administration
Code NTT-4

Washington, D.C.
20546-0001

NASA