Central Region Computer Programs and Problems
NWS CRCP - No. 1

A COMPUTER PROGRAM FOR DETERMINING AND DISPLAYING THE GEOSTROPHIC WINDS ON LAKES SUPERIOR AND MICHIGAN

Robert L. Somrek
National Weather Service Forecast Office
Chicago, IL

June 1980

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / National Weather Service

NOAA Central Region Computer Programs and Problems NWS CRCP - No. 1

A COMPUTER PROGRAM FOR DETERMINING AND DISPLAYING THE GEOSTROPHIC WINDS ON LAKES SUPERIOR AND MICHIGAN

Robert L. Somrek
National Weather Service Forecast Office
Chicago, IL

June 1980

UNITED STATES
DEPARTMENT OF COMMERCE
Philip M. Klutznick
Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Richard A. Frank, Administrator

National Weather
Service
Richard E. Hallgren, Director

CONTENTS

A COMPUTER PROGRAM FOR DETERMINING AND DISPLAYING THE GEOSTROPHIC WINDS ON LAKES SUPERIOR AND MICHIGAN

Robert L. Somrek
National Weather Service Forecast Office
Chicago, IL

I. GENERAL INFORMATION

A. Summary

The geostrophic wind has long been recognized as a useful tool for the forecaster. It is particularly useful in estimating the wind over large areas such as the Great Lakes where few or no reports are available.

A semi-graphical solution of the geostrophic wind equation as described by Johnson (1971) has been used by WSFO Chicago as an aid in analyzing and forecasting winds on Lakes Superior and Michigan. The arrival of AFOS minicomputers has presented the opportunity to abandon the graphical method and solve the equation using a numerical scheme.

A FORTRAN program is now used operationally to compute geostrophic winds at seven points on the Lakes using the sea-level pressure at ten stations as input. The program also generates an AFOS graphic which is used with a companion background to display the results of the computation.

B. Environment

The entire set of tasks associated with the program is completely integrated into the WSFO AFOS computer. An AFOS PROCEDURE oversees the orderly execution of all tasks including data input, file handling, program execution, and output display.

C. References

The technique used in the program for determining the geostrophic wind is described in the following paper:

Johnson, C.B., 1971: Practical Application of a Graphical Method of Geostrophic Wind Determination. NOAA Technical Memorandum NWS CR-47, National Weather Service, National Oceanic and Atmospheric Administration, U. S. Department of Commerce, Washington, D. C., 29pp.

The technique will be referred to as the "triangle method" in the discussion below.

A paper describing the AFOS Universal Transmission Format and a set of graphic subroutines developed by the NWS Western Region SSD were used as aids in constructing the array and subroutines necessary to produce output suitable for presentation on an AFOS GDM.

II. APPLICATION

A. Program Formulation

The simplicity of the geostrophic wind equation has lent itself to several graphical solutions. Among the disadvantages of graphical solutions, however, is the need for an accurately analyzed synoptic map. Johnson described a semi-graphical solution that eliminated the need for an analyzed map. The method, used at WSFO Chicago for many years as an aid in analyzing and forecasting winds on Lakes Superior and Michigan, retains the uncertainty inherent in any graphical method.

In a numerical scheme to solve the geostrophic wind equation, $V_g = -\frac{1}{\rho f} \nabla p \times \hat{k}$, the heart of the computation is determining the direction and magnitude of the horizontal pressure gradient force. The triangle method seeks to simplify this determination by separating the pressure gradient force into two components. Each component is chosen such that a geostrophic wind to be derived from it will be parallel to one side of a triangle circumscribing the area for which the total geostrophic wind is to be calculated. With a wind component's direction (θ) parallel to one side of the triangle, the magnitude of each pressure gradient component is equal to the difference in pressure across the other of the two chosen sides divided by the length of a line across the triangle which is perpendicular to the wind's direction and intersects the opposite vertex of the triangle. For each of the pressure gradient forces, a corresponding geostrophic wind is computed. The magnitude ($|V_g|$) of each component wind is given by the scalar form of the wind equation

$$|V_g| = \frac{1}{\rho f}\frac{\Delta p}{\Delta n} \qquad (1)$$

where ρ = the density of air
f = the Coriolis parameter
Δp = the pressure difference across the leg of the triangle
Δn = the length of a line across the triangle perpendicular to the predetermined direction of the computed wind and intersecting the opposite vertex.

The two geostrophic wind components are then resolved to the total geostrophic wind such that

$$|V_{g_T}| = \sqrt{(|V_{g_1}|\cos\theta_1 + |V_{g_2}|\cos\theta_2)^2 + (|V_{g_1}|\sin\theta_1 + |V_{g_2}|\sin\theta_2)^2} \qquad (2)$$

and

$$\theta_T = \tan^{-1}\left(\frac{|V_{g_1}|\sin\theta_1 + |V_{g_2}|\sin\theta_2}{|V_{g_1}|\cos\theta_1 + |V_{g_2}|\cos\theta_2}\right) \qquad (3)$$

where subscripts 1 and 2 denote each of the two components and subscript T indicates the resolved total geostrophic wind.

In applying the triangle method to portions of Lakes Superior and Michigan, the vertices of each triangle were chosen to be weather stations. Seven triangles and component sides were chosen as follows:

Triangle	Component Sides	
YQT-MQT-DLH	YQT-MQT	YQT-DLH
YQT-MQT-YXZ	YQT-MQT	YQT-YXZ
MQT-SSM-YXZ	MQT-SSM	MQT-YXZ
MQT-SSM-TVC	MQT-SSM	MQT-TVC
GRB-MQT-TVC	GRB-MQT	GRB-TVC
GRB- X -TVC	GRB- X	GRB-TVC
ORD-SBN- X	ORD-SBN	ORD- X

(X is a point midway between MKE and MKG)

Using the longitude and latitude for each station and some spherical trigonometry, values of Δn and β (the orientation of each side of the triangle) can be calculated. For either component in a given triangle the values of Δn and θ--a function of β ($\theta = \beta \pm 90^o$)--are known as well as the density of air and the Coriolis parameter. Equation 1 now reduces to

$$|V_g| = C\,\Delta p \tag{4}$$

where C is a constant and is equal to $\frac{1}{\rho f \Delta n}$. If points A, B, and C denote the vertices of the triangle, equations 2 and 3 may be re-written as simply

$$|V_{g_T}| = \sqrt{(C_1(p_A-p_B)\cos\theta_1 + C_2(p_A-p_C)\cos\theta_2)^2 + (C_1(p_A-p_B)\sin\theta_1 + C_2(p_A-p_C)\sin\theta_2)^2} \tag{5}$$

and

$$\theta_T = \tan^{-1}\left(\frac{C_1(p_A-p_B)\sin\theta_1 + C_2(p_A-p_C)\sin\theta_2}{C_1(p_A-p_B)\cos\theta_1 + C_2(p_A-p_C)\sin\theta_2}\right). \tag{6}$$

For the fixed geometry of each triangle, the geostrophic wind is only a function of the pressure differences among the vertices of the triangle.

Equations 5 and 6 form the heart of the FORTRAN program. Twenty-eight steps (annotated in the program listing) perform the necessary computations for each of the seven triangles. Much of the remainder of the program is dedicated to producing the AFOS graphic to display the results of the computations.

B. Software Structure

The program is written in Data General's FORTRAN IV and runs on the standard Eclipse S/230 AFOS minicomputer.

While the primary purpose of the program is to compute geostrophic winds, the program has been structured to provide a logical flow in building the graphic array and the AFOS graphic file. The program listing (Section III E.) has been annotated to describe the program structure and flow.

Five distinct although intermingled segments are used in the program. They are:

Data Initialization
Graphic Array and AFOS Graphic File Construction
Data Input
Geostrophic Wind Computation
Wind Barb Plotting

Data Initialization: The program makes use of a large number of constants in its execution. Not only are values of C (in Equation 4) and θ initialized but also alphanumerics used in the legends, display coordinates for the legends, and coordinates for the wind barbs and shafts are initialized. Several dummy constants are also initialized to accommodate DG FORTRAN IV statement syntax.

Graphic Array and AFOS Graphic File Construction: An integer array (GPHARAY) is used to build the AFOS graphic file as various elements of the program output are computed or declared. A DATA statement initializes the array early in the program by assigning an AFOS communication header, defining the physical dimensions of the display, and instructing the GDM to frame the display area. Two subroutines--TEXT and LINE--are used to transfer data from the program into the graphic array. The subroutines insert the proper Universal Transmission Format command codes into the array before restructuring and/or accepting the data for inclusion into the array. Data are passed into the graphic array through the use of integer arrays. In the case of subroutine TEXT, the size of the alphanumerics and the beginning position are also passed to the array through the subroutine.

In calling the subroutine LINE at the end of the barb plotting program segment, use is made of the graphic feature which allows two points to be connected logically but no visible line drawn. It is enabled in the program by increasing the X-coordinate of the next barb endpoint by 5000. In the subroutine the 5000-plus coordinate sets a flag which allows the appropriate coordinate bit to be set to defeat the visible line.

A final call to subroutine UTFORMAT at the end of the program restructures the graphic array into the Universal Transmission Format and outputs the array to the RDOS file GLKWNDPLT.

Data Input: Data are input to the program as an RDOS file through the use of a SAVED AFOS database product. The entire product--data and prompting messages--is read in a block as ASCII text. The rigid, structured nature of the AFOS preformat used in constructing the product allows data in the input file to be accessed by its location in the file. No string searching routine is necessary; nor is a complicated FORMAT statement.

Portions of the data are stored across computer word boundaries and require byte masking and bit shifting to extract the necessary datum.

Since data are read as ASCII text and must be returned to the graphic array as ASCII text, several conversions to and from numerics occur throughout the program.

Geostrophic Wind Computation: Twenty-eight program steps are necessary to compute the geostrophic winds. Values of Δp are computed first. After each component magnitude is computed and each component direction is defined, the vectors are resolved to produce the total geostrophic wind magnitude and direction.

Wind Barb Plotting: A large portion of the program is dedicated to plotting wind barbs. This program segment will plot 50, 10, and 5 knot barbs. The segment also places the tens digit of the direction at the end of the wind shaft.

As described above, this segment makes use of subroutine LINE's feature which allows blank lines to be drawn. All wind barbs on one shaft are thereby plotted as a single uninterrupted line although the visible plot appears to be the appropriate number of separate barbs.

If the geostrophic wind is less than 3 knots, this program segment is skipped and the world CALM is placed in the legend and plotted in place of wind barbs in the graphic display.

C. Database

The input file used by the program is created at an ADM as an AFOS database product. The file is moved from the AFOS database to a portion of the rigid disk where it can be accessed by the program. In the move the file is renamed.

The output file is created during program execution. The file is an RDOS disk file in the AFOS graphic format. When used with a companion background, the file displays the results of the geostrophic wind computations.

D. Machine Requirements

This program runs in background memory on a WSFO minicomputer. The executable savefile requires slightly more than 16K of disk space for storage. The typical execution time from a rigid disk is about 7-8 seconds.

The program assigns data channel 25 for the input. Input data are contained in one block (512 bytes). During execution data channel 25 is closed and reassigned for output. Output data are stored in 1024 bytes of disk space.

III. PROCEDURES

A. Initiation of the Program

An AFOS PROCEDURE, "GLKWND," is used to initiate program execution. The PROCEDURE also oversees the orderly execution of all tasks associated with the running of the program including data input, file handling, and output display. The PROCEDURE is listed in Fig. 1.

The PROCEDURE first displays the pre-format used in entering the necessary data. The product is stored as an AFOS database product. After a pause of five seconds (to allow the minicomputer to complete the tasks associated with storing the product), the file is moved from the database and placed on the rigid disk as an RDOS file. In the move the file is renamed. After again allowing an appropriate pause, the command is given to execute the program. A pause is again necessary to allow the program time to be completed. The next command in the PROCEDURE requests an RDOS file containing the background to be presented. The background was produced separately and contains those items in the graphic display which do not change from run to run (i.e., map of the Lakes, various portions of legends and headings). With the background displayed, the PROCEDURE commands the display of the RDOS file GLKWNDPLT which contains the output of the program. The procedure is then terminated.

B. Input

The input used by the program is created as an AFOS database product, CHIGLKWND, at an ADM console using a pre-format (Fig. 2). The required entry to the pre-format consists of sea-level pressure (in the same format as reported in the aviation code) for DLH, GRB, MKE, MKG, MQT, ORD, SSM, TVC, YQT, and YXZ. A date/time group for the pressure data may also be included. If no entry is made, the program assumes the data were observed at the previous six-hourly synoptic time. If an entry is made and it is not the previous synoptic time, the program assumes the data are forecast values and makes several changes in the output legend.

Once all input data are entered at the ADM, the ADM ENTER key is stroked and the product is stored. The PROCEDURE then resumes control of the commands to complete the remaining tasks.

C. Output

The output file, GLKWNDPLT, is created during program execution. If the file already exists, it is deleted before the new one is created. The output file is an RDOS file in the AFOS Universal Transmission Format. It contains the appropriate instructions to be used by a GDM to display the output. The file is displayed by the controlling PROCEDURE using the AFOS DSP command.

When displayed with a companion background which was created independently, the output contains a legend indicating whether the computed winds are based on observed or forecast pressure values, the values of the geostrophic wind for seven locations on Lakes Superior and Michigan, a map of the two lakes with wind barbs plotted at the correct locations, and the input pressures plotted near station identifiers along the perimeter of the lakes map.

Fig. 3 contains an example of the background graphic GLKWNDGRD. Figs. 4 and 5 contain examples of the two outputs available from the program.

D. Cautions and Restrictions

The program assumes valid data have been entered in the input file. No check is made to see if non-numeric characters have been entered in place of pressure data. Due to the nature of the ASCII code and the manner in which it is converted in the program, the program will run with bad data but the output will consist of unrealistic values of the geostrophic wind. An incidental check is made of the date/time input data. Invalid entries will produce a default to the previous six-hourly synoptic time.

When synoptic or airways weather observations become reliably available on the AFOS system, the program can be rewritten to obtain the pressure values from the database. At that time the portion of the PROCEDURE which requires the input of pressure data using the pre-format can be eliminated.

E. Program Listing

(See the following pages for program and subroutine listings.)

```
MKCPCD007
WDUS00 KCHI 251857
  DISPLAY   MODE  ACC/OV    COMMAND
   (1-4)    (D/M) (R/A/O)   (ANY COMMAND; LAST LINE MUST BE END OR "NAME")
  -------   ----- -------   ---------------------------------------------
01                           M:005
02                           PAUSE 05
03                           SAVE:CHIGLKWND DP0:GLKWNDDAT
04                           PAUSE 05
05                           RUN:GLKWNDPGM.SV
06                           PAUSE 20
07   2        D      R       DSP:GLKWNDGRD
08   2        D      O       DSP:GLKWNDPLT
09                           END
10
11
12
13
14
15
16
17
18

                                                                PAGE 01
```

Figure 1

```
MKCMCP005
WOUS00 KCHI 241726

       * * * * * GEOSTROPHIC WIND COMPUTATION * * * * *

ENTER THE DATE/TIME GROUP FOR THIS DATA:
     [--/--00Z]   --THE DEFAULT IS THE PREVIOUS SYNOPTIC TIME

ENTER THE PRESSURE AT THE FOLLOWING LOCATIONS:
    DLH:  [    ]
    GRB:  [    ]
    ORD:  [    ]
    MKE:  [    ]
    MKG:  [    ]
    MQT:  [    ]
    SBN:  [    ]
    SSM:  [    ]
    TVC:  [    ]
    YQT:  [    ]
    YXZ:  [    ]    -- NOW STRIKE "ENTER"

                                                                        [ ]
                                                                   PAGE 01
```

Figure 2

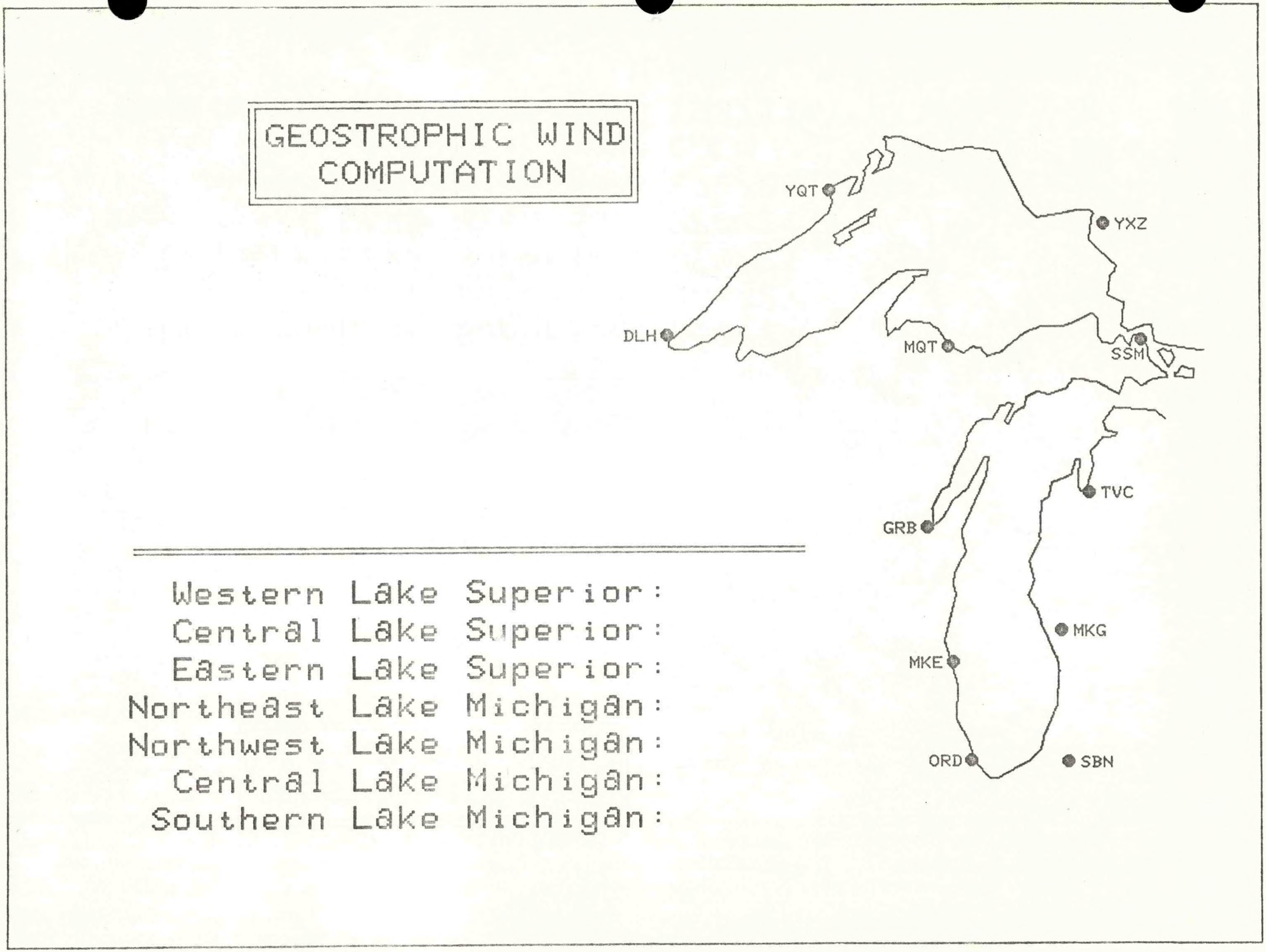
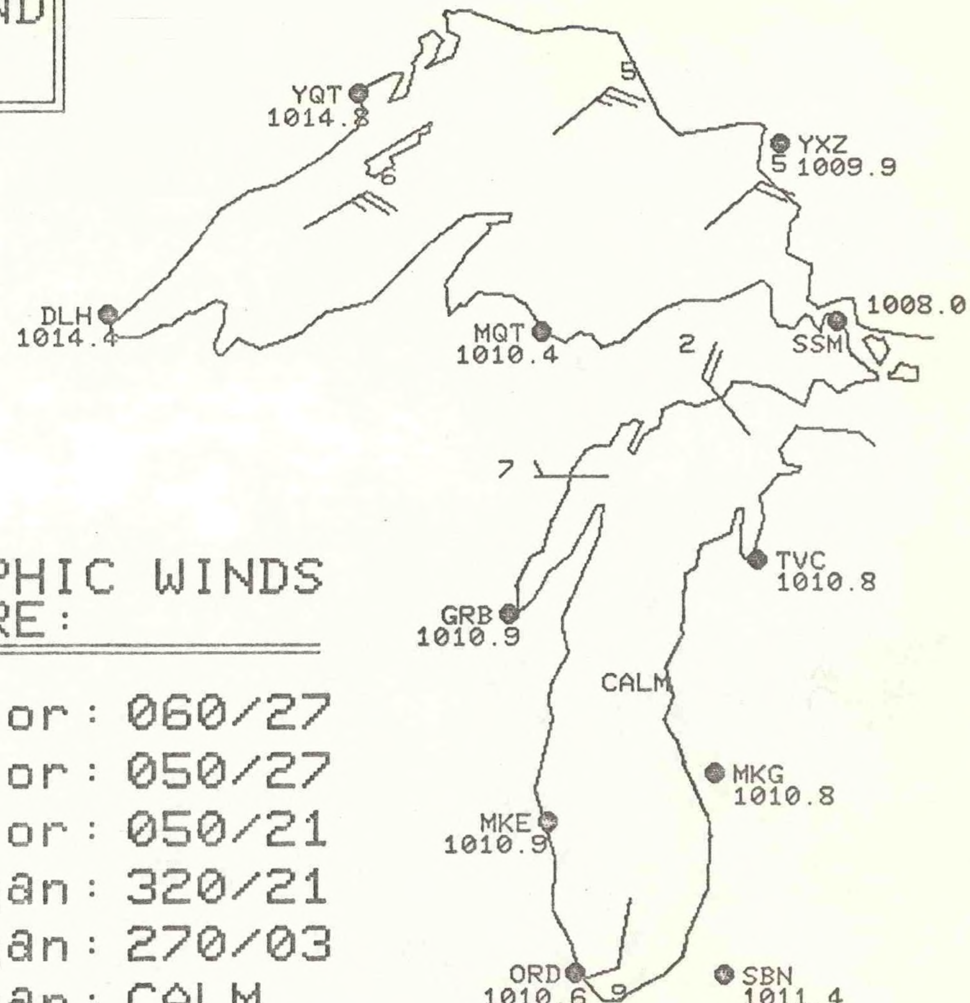

GEOSTROPHIC WIND
COMPUTATION
YQT
YXZ
DLH
MQT
SSM
TVC
GRB
MKG
MKE
ORD
SBN
Western Lake Superior:
Central Lake Superior:
Eastern Lake Superior:
Northeast Lake Michigan:
Northwest Lake Michigan:
Central Lake Michigan:
Southern Lake Michigan:

Figure 3

GEOSTROPHIC WIND
COMPUTATION



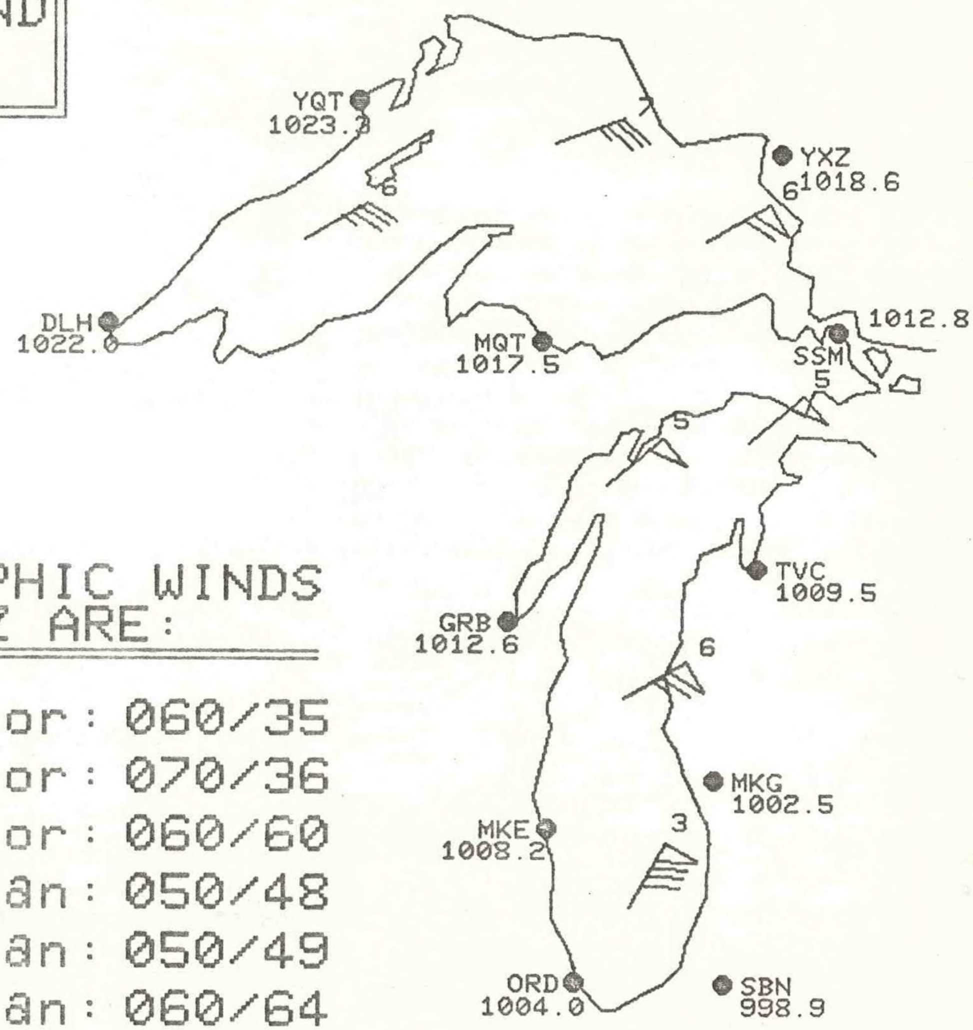
THE COMPUTED GEOSTROPHIC WINDS
AT 04/1200Z ARE:

Western Lake Superior: 060/27
Central Lake Superior: 050/27
Eastern Lake Superior: 050/21
Northeast Lake Michigan: 320/21
Northwest Lake Michigan: 270/03
Central Lake Michigan: CALM
Southern Lake Michigan: 190/11

Run time 1539Z 04Mr80

Figure 4

GEOSTROPHIC WIND
COMPUTATION



THE FORECAST GEOSTROPHIC WINDS
VALID AT 05/1200Z ARE:

Western Lake Superior:	060/35
Central Lake Superior:	070/36
Eastern Lake Superior:	060/60
Northeast Lake Michigan:	050/48
Northwest Lake Michigan:	050/49
Central Lake Michigan:	060/64
Southern Lake Michigan:	030/79

Run time: 1544Z 04Mr80

Figure 5

A COMPUTER PROGRAM FOR DETERMINING AND DISPLAYING THE GEOSTROPHIC WINDS ON LAKES SUPERIOR AND MICHIGAN

ROBERT L. SOMREK
National Weather Service Forecast Office Chicago, IL

This program computes the geostrophic wind at seven locations on Lakes Superior and Michigan. The program also generates an AFOS graphic to be used with a companion background to display the output of the program.

```
      COMPILER NOSTACK
      DIMENSION LEGEND(31),PRESSURE(11),DELTAP(14),CNSTNT(14),CMPNTMAG(14),
     1  CMPNTDIR(14),MONTH(12)
      INTEGER GPHARAY(512),WORD(0:511),CLKTIME(3),CLKDATE(3),TESTDIGIT,
     1  PREVZTIME,FCSTDATE,FCSTTIME,TXTARAY(7),PRESXY(11,2),TEXTY,GEOWNDMAG,
     2  GEOWNDDIR,SHAFTXY(7,2),SHAFT(5),TEXTX,TEXTY2,BARB50X,BARB50Y,DELTAX,
     3  DELTAY,BRBARAY(50),UNITDIGIT,BARB10X,BARB10Y,BTMLEGEND(14)
      DATA NOWORD/27/GPHARAY/'CH','IG','PH','GL','P ',0,2*-1,1280,140400K,0,   ; Initialization of the array to be used in
     1  4096,3072,2*0,141400K,2*50,8,4000,2*0,3000,010140K,2*0,012110K/        ;  building the AFOS graphic file occurs in this
                                                                              ;  DATA statement. The graphic file is given a
                                                                              ;  name, the physical dimensions of the display
                                                                              ;  are defined, and instructions are given to
                                                                              ;  frame the display.
      CALL OPEN (25,'GLKWNDDAT',2,IERR)                                       ; The input file is opened and read as a block.
      CALL WRBLK (25,0,WORD,2,IERR)                                           ;  The data appears as ASCII text.
      DATA IDUM/29/JDUM/28/LEGEND/'TH','E ','CO','MP','UT','ED',' G','EO','ST', ; The text to be used as a heading for the
     1  'RO','PH','IC',' W','IN','DS',020015K,' ','  ','  ','  ','AT','  ','  ', ;  graphic is defined. By default, a heading for
     2  '  ','00','Z ','AR','E:',3*0/                                          ;  input data valid at the previous synoptic time
                                                                              ;  is assumed.
      CALL DATE (CLKDATE,IERR)
      CALL TIME (CLKTIME,IERR)
      PREVZTIME=CLKTIME(1)/6*6
      TESTDIGIT=ISHFT(177400K.AND.WORD(113K),-8)-48                           ; If the character in this position is anything
      IF (TESTDIGIT.GE.0.AND.TESTDIGIT.LE.9)  GOTO 2                          ;  but a digit, the program assumes the time of
    1 LEGEND(22)=(8192)+(CLKDATE(2)/10+48)                                    ;  the input data is the previous synoptic time.
      LEGEND(23)=(MOD(CLKDATE(2),10)*256+12288)+(47)                          ;  If a digit is in this position, the program
      LEGEND(24)=(PREVZTIME/10*256+12288)+(MOD(PREVZTIME,10)+48)              ;  passes to statement #2 where a check is made
      GOTO 4                                                                  ;  to see whether or not the input time is the
    2 FCSTDATE=(((377K.AND.WORD(112K))-48)*10)+(TESTDIGIT)                    ;  previous synoptic time. If not, the program
      FCSTTIME=((ISHFT(177400K.AND.WORD(114K),-8)-48)*10)+                    ;  assumes the input data are forecast pressures
     1  ((377K.AND.WORD(114K))-48)                                            ;  and makes the appropriate changes in the
```

```
  IF (FCSTDATE.EQ.CLKDATE(2).AND.FCSTTIME.EQ.PREVZTIME)  GOTO 1      ; legend. The legend is transferred to the
  LEGEND(3)='FO'                                                    ; graphic array through the subroutine called
  LEGEND(4)='RE'                                                    ; at statement #4.
  LEGEND(5)='CA'
  LEGEND(6)='ST'
  DO 3  I=0,7
  LEGEND(IDUM-I)=LEGEND(JDUM-I)
3 CONTINUE
  LEGEND(17)='  '
  LEGEND(18)='  '
  LEGEND(19)='VA'
  LEGEND(20)='LI'
  LEGEND(21)='D '
  LEGEND(23)=(8192)+(377K.AND.WORD(112K))
  LEGEND(24)=(MOD(FCSTDATE,10)*256+12288)+(47)
  LEGEND(25)=WORD(114K)
4 CALL TEXT (3,GPHARAY,408,1375,LEGEND,NOWORD)
  DO 5  I=1,11                                                      ; * * * * * * * * * * * * * * * *
  J=I*8+125
  PRESSURE(I)=900+(((377K.AND.WORD(J))-48)*10)+                     ; In this loop the pressure (in tens, units, and
 1 (ISHFT(177400K.AND.WORD(J+1),-8)-48)+(FLOAT((377K.AND.WORD(J+1))-48)/10) ; tenths) is read from the input tile and con-
  IF (PRESSURE(I).LT.950)  PRESSURE(I)=PRESSURE(I)+100              ; verted from ASCII equivalents to numerics.
  TXTARAY(1)=030460K                                                ; Since the format of the input file is fixed
  IF (PRESSURE(I).LT.1000)  TXTARAY(1)=000071K                      ; specific words can be accessed and the appro-
  TXTARAY(2)=ISHFT(377K.AND.WORD(J),8).OR.ISHFT(177400K.AND.WORD(J+1),-8) ; priate byte masked; no string search is ne-
  TXTARAY(3)=(11776)+(377K.AND.WORD(J+1))                           ; cessary. The last 7 steps of the loop take the
  DATA TXTARAY(4)/0/PRESXY/1918,2747,2884,2804,3399,2829,3421,3678,3487, ; ASCII equivalents of the pressure and transfer
 1 2438,3531,1887,1267,520,838,938,1851,519,1952,1378,2344,2237/    ; them to the graphic array to be plotted.
  CALL TEXT (1,GPHARAY,PRESXY(I,1),PRESXY(I,2),TXTARAY,NOWORD)
5 CONTINUE                                                          ; * * * * * * * * * * * * * * * *
  CALL CLOSE (25,IERR)
  CALL DFILW ('GLKWNDDAT',IERR)
```

```
  DELTAP(1)=PRESSURE(10)-PRESSURE(6)                              ; * * * * * * * * * * * * * * * *
  DELTAP(2)=PRESSURE(10)-PRESSURE(1)
  DELTAP(3)=DELTAP(1)
  DELTAP(4)=PRESSURE(10)-PRESSURE(11)
  DELTAP(5)=PRESSURE(6)-PRESSURE(8)
  DELTAP(6)=PRESSURE(6)-PRESSURE(11)
  DELTAP(7)=DELTAP(5)
  DELTAP(8)=PRESSURE(6)-PRESSURE(9)
  DELTAP(9)=PRESSURE(2)-PRESSURE(6)
  DELTAP(10)=PRESSURE(2)-PRESSURE(9)
  DELTAP(11)=PRESSURE(2)-(PRESSURE(4)+PRESSURE(5))/2              ; The geostrophic  wind is  computed  in these 28
  DELTAP(12)=DELTAP(10)                                           ;  steps. Each component magnitude is computed in
  DELTAP(13)=PRESSURE(3)-PRESSURE(7)                              ;  the loop ending at statement #6. The direction
  DELTAP(14)=PRESSURE(3)-(PRESSURE(4)+PRESSURE(5))/2              ;  of  each  component  is pre-determined by  the
  DATA KDUM/0/CNSTNT/6.1909,5.4358,8.4715,5.9804,10.8387,10.5927,7.9407,  ;  geometry of  the triangles  used in this tech-
 1  7.8167,7.0374,8.1430,9.4328,8.1778,14.4795,11.1173/           ;  nique and is given  in the  DATA statement  as
  DO 6  I=1,14                                                    ;  CMPNTDIR.  The paired components  are resolved
  CMPNTMAG(I)=CNSTNT(I)*DELTAP(I)                                 ;  to  produce the total  geostrophic wind magni-
6 CONTINUE                                                        ;  tude  (GEOWNDMAG)  and direction  (GEOWNDDIR).
  DATA TEXTY/1215/CMPNTDIR/3.8304,2.1553,2.9677,5.2968,3.7384,6.2731,5.3813, ;  The values are rounded to the nearest knot and
 1  3.1316,0.1525,4.5245,3.2941,5.2167,4.3418,6.2136/             ;  ten degrees respectively.
  DO 17  I=1,13,2
  GEO1=CMPNTMAG(I)*COS(CMPNTDIR(I))+CMPNTMAG(I+1)*COS(CMPNTDIR(I+1))
  GEO2=CMPNTMAG(I)*SIN(CMPNTDIR(I))+CMPNTMAG(I+1)*SIN(CMPNTDIR(I+1))
  GEOWNDMAG=SQRT(GEO1**2+GEO2**2)
  IF (GEO1.GE.0)  GOTO 7
  GEOWNDDIR=IFIX((90-ATAN(GEO2/GEO1)*57.29577)*0.1+0.5)*10
  GOTO 8
7 GEOWNDDIR=IFIX((270-ATAN(GEO2/GEO1)*57.29577)*0.1+0.5)*10
8 IF (GEOWNDDIR.EQ.0)  GEOWNDDIR=360                              ; * * * * * * * * * * * * * * * *
```

```
      IF (GEOWNDMAG.LT.3)  GOTO 15
      TXTARAY(1)=(GEOWNDDIR/100*256+12288)+(MOD(GEOWNDDIR/10,10)+48)
      TXTARAY(2)=030057K
      J=3                                                        ; The values of the geostrophic wind magnitude
      IF (GEOWNDMAG.LT.100)  GOTO 9                              ;  and direction are coverted to their ASCII
      TXTARAY(3)=GEOWNDMAG/100*256+12288                         ;  equivalents and transferred to the graphic
      J=4                                                        ;  array for plotting in the output display.
    9 TXTARAY(J)=(MOD(GEOWNDMAG/10,10)*256+12288)+(MOD(GEOWNDMAG,10)+48)
      TXTARAY(J+1)=0
      TEXTY=TEXTY-120
      CALL TEXT (3,GPHARAY,2150,TEXTY,TXTARAY,NOWORD)            ; * * * * * * * * * * * * * * * *
      DATA K/1/SHAFT(5)/9999/SHAFTXY/2515,3028,3341,3428,3136,3171,3185,2129,   ; From this point  to statement #17,  coordinates
     1  2322,2124,1705,1617,1172,741/                            ;  for plotting wind barbs appropriate to each of
      SHAFT(1)=SHAFTXY(K,1)                                      ;  the seven computed  geostrophic winds are cal-
      SHAFT(2)=SHAFTXY(K,2)                                      ;  culated. The computations plot the wind shaft,
      THETA=(450-GEOWNDDIR)/57.29577                             ;  place  the tens digit of the direction  at the
      SHAFT(3)=SHAFT(1)+IFIX(COS(THETA)*150)                     ;  end  of the  shaft,  and  plot the appropriate
      SHAFT(4)=SHAFT(2)+IFIX(SIN(THETA)*150)                     ;  number of 50, 10, and 5 Knot barbs.
      CALL LINE (GPHARAY,SHAFT,NOWORD)
      TEXTX=SHAFT(3)+((GEOWNDDIR/180+1)/2+1)*IFIX(COS(THETA)*35)
      TEXTY2=SHAFT(4)+((GEOWNDDIR/180+1)/2+1)*IFIX(SIN(THETA)*35)
      TXTARAY(1)=377K.AND.TXTARAY(1)
      TXTARAY(2)=0
      CALL TEXT (1,GPHARAY,TEXTX,TEXTY2,TXTARAY,NOWORD)
      BARB50X=COS(THETA-1.57079)*75
      BARB50Y=SIN(THETA-1.57079)*75
      DELTAX=-COS(THETA)*20
      DELTAY=-SIN(THETA)*20
      BRBARAY(1)=SHAFT(3)
      BRBARAY(2)=SHAFT(4)
      J=3
   10 IF(GEOWNDMAG.LE.47)  GOTO 11
      BRBARAY(J)=BRBARAY(J-2)+BARB50X                            ; - - - - - - - - - - - - - - -
      IF (BRBARAY(J).GT.5000)  BRBARAY(J)=BRBARAY(J)-5000
      BRBARAY(J+1)=BRBARAY(J-1)+BARB50Y
      BRBARAY(J+2)=BRBARAY(J)-BARB50X+2*DELTAX                   ;  This  portion of  the barb  plotting
      BRBARAY(J+3)=BRBARAY(J-1)+2*DELTAY                         ;  routine  computes  graphic   display
      BRBARAY(J+4)=BRBARAY(J+2)+DELTAX+5000                      ;  coordinates for plotting 50 Kt barbs
      BRBARAY(J+5)=BRBARAY(J+3)+DELTAY
      J=J+6
      GEOWNDMAG=GEOWNDMAG-50
      GOTO 10                                                    ; - - - - - - - - - - - - - - -
```

```
11 UNITDIGIT=MOD(GEOWNDMAG,10)                                          ;  - - - - - - - - - - - - - - -
   IF (UNITDIGIT.GT.7)  GEOWNDMAG=(GEOWNDMAG/10+1)*10
   BARB10X=COS(THETA-1.09129)*75
   BARB10Y=SIN(THETA-1.09129)*75
   I3=GEOWNDMAG/10
   DO 12  I2=1,I3                                                       ;  Graphic  coordinates  for  ten  Knot
   BRBARAY(J)=BRBARAY(J-2)+BARB10X                                      ;  barbs are calculated in this section
   IF (BRBARAY(J).GT.5000)  BRBARAY(J)=BRBARAY(J)-5000                  ;  of the program.
   BRBARAY(J+1)=BRBARAY(J-1)+BARB10Y
   BRBARAY(J+2)=BRBARAY(J)-BARB10X+DELTAX+5000
   BRBARAY(J+3)=BRBARAY(J-1)+DELTAY
   J=J+4
12 CONTINUE                                                             ;  - - - - - - - - - - - - - - -
   IF (UNITDIGIT.GT.7.OR.UNITDIGIT.LT.3)  GOTO 14
   IF (GEOWNDMAG.GT.7)  GOTO 13
   BRBARAY(1)=SHAFT(3)+DELTAX
   BRBARAY(2)=SHAFT(4)+DELTAY                                           ;  Five Knot  barb coordinates are cal-
13 BRBARAY(J)=BRBARAY(J-2)+BARB10X/2                                    ;  culated in these eight statements.
   IF (BRBARAY(J).GT.5000)  BRBARAY(J)=BRBARAY(J)-5000
   BRBARAY(J+1)=BRBARAY(J-1)+BARB10Y/2
   J=J+2                                                                ;  - - - - - - - - - - - - - - -
14 BRBARAY(J)=9999
   CALL LINE (GPHARAY,BRBARAY,NOWORD)                                   ;  Finally,  the entire set of coordin-
                                                                        ;  ates is transferred to  the  graphic
                                                                        ;  array through this subroutine.
   GOTO 16                                                              ;  - - - - - - - - - - - - - - -
15 TXTARAY(1)=041501K                                                   ;  If the geostrophic wind is less than
   TXTARAY(2)=046115K                                                   ;  3 Knots, flow  of  the  program  was
   TXTARAY(3)=0                                                         ;  transferred to these six statements.
   TEXTY=TEXTY-120                                                      ;  They  combine  to plot  CALM in  the
   CALL TEXT (3,GPHARAY,2150,TEXTY,TXTARAY,NOWORD)                      ;  graphic  legend and in place of wind
   CALL TEXT (1,GPHARAY,SHAFTXY(K,1)-45,SHAFTXY(K,2),TXTARAY,NOWORD)    ;  barbs on the display.
16 K=K+1
17 CONTINUE
```

```
      DATA LDUM/0/BTMLEGEND/'Ru','n ','ti','me',': ',' ',2*0,'Z ','  ',4*0/
      DATA MDUM/0/MONTH/'Ja','Fe','Mr','Ap','My','Jn','Jl','Au','Se','Oc','No',
     1 'De'/
      DO 18  I=1,2
      BTMLEGEND(I+6)=(CLKTIME(I)/10*256+12288)+(MOD(CLKTIME(I),10)+48)
   18 CONTINUE
      BTMLEGEND(11)=(CLKDATE(2)/10*256+12288)+(MOD(CLKDATE(2),10)+48)
      BTMLEGEND(12)=MONTH(CLKDATE(1))
      BTMLEGEND(13)=(CLKDATE(3)/10*256+12288)+(MOD(CLKDATE(3),10)+48)
      CALL TEXT (1,GPHARAY,100,100,BTMLEGEND,NOWORD)
      CALL UTFORMAT (GPHARAY)

      STOP
      END
```

; These final statements provide the time and
; date the program was run. This information
; serves as a reference, particularly when the
; input data consists of forecast pressures.

; The call to this subroutine rearranges the
; graphics array into the Universal Transmission
; Format and transfers it to the RDOS file
; GLKWNDPLT. When displayed with a companion
; background, this RDOS file produces the AFOS
; graphic display.

* *

```
      SUBROUTINE TEXT (CHRSIZ,GPHARAY,X,Y,TXTARAY,NOWORD)
      INTEGER CHRSIZ,GPHARAY(1),X,Y,TXTARAY(1)
      GPHARAY(NOWORD+1)=142400K
      GPHARAY(NOWORD+2)=X+50
      CALL ISET (GPHARAY(NOWORD+2),14)
      IF (CHRSIZ.EQ.3)  CALL ISET (GPHARAY(NOWORD+2),15)
      GPHARAY(NOWORD+3)=Y+50
      NOWORD=NOWORD+3
      DO 1  I=1,100
      IF (TXTARAY(I).EQ.0)  GOTO 2
      GPHARAY(NOWORD+I)=TXTARAY(I)
    1 CONTINUE
    2 NOWORD=NOWORD+(I-1)
      RETURN
      END
```

; This subroutine allows alphanumerics to be in-
; cluded in the graphic array. The text is
; passed two characters per computer word
; through the integer array TXTARAY. The text is
; plotted in the graphic display such that the
; lower left-hand corner of the first character
; is at location X,Y (in pixels). A choice of
; two character sizes is passed through the
; argument CHRSIZ.

```
      SUBROUTINE LINE (GPHARAY,LINARAY,NOWORD)
      DIMENSION LINARAY(1)
      INTEGER GPHARAY(1),DELTAX,DELTAY
      GPHARAY(NOWORD+1)=141400K
      GPHARAY(NOWORD+2)=50
      GPHARAY(NOWORD+3)=50
      NOWORD=NOWORD+4
      NTEMP=NOWORD
      KOUNT=0
      LASTX=0
      LASTY=0
      IFLAG=0
      DO 6 I=1,499,2
      IF (LINARAY(I).EQ.9999)  GOTO 7
      IF (LINARAY(I).LT.5000)  GOTO 1
      LINARAY(I)=LINARAY(I)-5000
      IFLAG=1


    1 NOWORD=NOWORD+2
      KOUNT=KOUNT+2
      DELTAX=LINARAY(I)-LASTX
      DELTAY=LINARAY(I+1)-LASTY
      GPHARAY(NOWORD-1)=DELTAX
      GPHARAY(NOWORD)=DELTAY
      LASTX=LINARAY(I)
      LASTY=LINARAY(I+1)
      IF (DELTAX.GE.0) GOTO 3
      DO 2 J=13,15
      CALL ICLR (GPHARAY(NOWORD-1),J)
    2 CONTINUE
    3 IF (DELTAY.GE.0)  GOTO 5
      DO 4 J=13,15
      CALL ICLR (GPHARAY(NOWORD),J)
    4 CONTINUE
    5 CALL ICLR (GPHARAY(NOWORD),13)
      IF (IFLAG.EQ.1.OR.I.EQ.1)  CALL ISET (GPHARAY(NOWORD),13)
      IFLAG=0
    6 CONTINUE
    7 GPHARAY(NTEMP)=KOUNT
      RETURN
      END
```

```
; This subroutine allows the construction of
;  straight lines in the graphic display.  The
;  endpoints (x,y coordinates) of each line or
;  line segment are passed as paired points
;  through the integer array LINARAY.

; The subroutine is constructed such that if the
;  x-coordinate passed through the array exceeds
;  the display dimensions it is understood that
;  no visible line will be drawn between the pre-
;  vious paired coordinates and the current pair.
;  This feature is used in the wind barb plotting
;  program segment.

; The Universal Transmission Format does not
;  accept line endpoints for drawing lines.
;  Rather the graphic array must be constructed
;  such that incremental values of x and y from
;  the current cursor position define the
;  direction and magnitude of the line to be
;  drawn.
```

```
      SUBROUTINE UTFORMAT (GPHARAY)          ; This subroutine puts the graphic array into the
      INTEGER GPHARAY(1),TMPARAY(1024),BLOCKS  ; Universal Transmission Format and outputs the
      IFLAG=0                                ; restructured array to the RDOS file GLKWNDPLT.
      DO 1  I=1,512
      TMPARAY(I*2-1)=ISHFT(GPHARAY(I),-8)
      TMPARAY(I*2)=ISHFT(ISHFT(GPHARAY(I),8),-8)
    1 CONTINUE
      DO 6  I=1,1024
      IF (TMPARAY(I).EQ.203K)  GOTO 2
      IF (TMPARAY(I).NE.20K)  GOTO 6
      IF (TMPARAY(I-1).EQ.20K)  GOTO 6
      IFLAG=1
    2 J=1025
    3 J=J-1
      IF (J.LT.I+2)  GOTO 4
      TMPARAY(J)=TMPARAY(J-1)
      GOTO 3
    4 IF (IFLAG.EQ.1)  GOTO 5
      TMPARAY(I)=20K
    5 TMPARAY(I+1)=14K
      IF (IFLAG.EQ.1)  TMPARAY(I+1)=20K
      IFLAG=0
    6 CONTINUE
      DO 7  I=1,512
      GPHARAY(I)=(ISHFT(TMPARAY(I*2-1),8))+(TMPARAY(I*2))
    7 CONTINUE
      DO 8  I=1,512
      IF (GPHARAY(513-I).NE.0)  GOTO 9
    8 CONTINUE
    9 GPHARAY(514-I)=203K
      CALL DFILW ('GLKWNDPLT',IERR)
      BLOCKS=(514-I)/256+1
      CALL CFILW ('GLKWNDPLT',3,BLOCKS,IERR)
      CALL OPEN (25,'GLKWNDPLT',0,IERR)
      CALL RDBLK (25,0,GPHARAY,BLOCKS,IERR)
      CALL CLOSE (25,IERR)
      RETURN
      END
```

```
CHIGLKWND
WOUS00 KCHI 112010

       * * * * * GEOSTROPHIC WIND COMPUTATION * * * * *

ENTER THE DATE/TIME GROUP FOR THIS DATA:
     [11/2000Z]   --THE DEFAULT IS THE PREVIOUS SYNOPTIC TIME

ENTER THE PRESSURE AT THE FOLLOWING LOCATIONS:
    DLH:  [241]
    GRB:  [213]
    ORD:  [226]
    MKE:  [224]
    MKG:  [194]
    MQT:  [186]
    SBN:  [215]
    SSM:  [132]
    TVC:  [184]
    YQT:  [219]
    YXZ:  [138]    -- NOW STRIKE "ENTER"
```

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

NOAA, the *National Oceanic and Atmospheric Administration*, was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth, and to assess the socioeconomic impact of natural and technological changes in the environment.

The six Major Line Components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS — Important definitive research results, major techniques, and special investigations.

TECHNICAL REPORTS—Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS — Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.

CONTRACT AND GRANT REPORTS—Reports prepared by contractors or grantees under NOAA sponsorship.

TECHNICAL SERVICE PUBLICATIONS—These are publications containing data, observations, instructions, etc. A partial listing: Data serials; Prediction and outlook periodicals; Technical manuals, training papers, planning reports, and information serials; and Miscellaneous technical publications.

ATLAS—Analysed data generally presented in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER
ENVIRONMENTAL DATA SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

3300 Whitehaven Street, N.W.
Washington, D.C. 20235