OCEAN SURFACE CURRENT SIMULATIONS IN THE NORTH PACIFIC OCEAN AND BERING SEA (OSCURS -- NUMERICAL MODEL)

by

W. James Ingraham, Jr.

and

Robert K. Miyahara

Resource Ecology and Fisheries Management Division
Northwest and Alaska Fisheries Center
National Marine Fisheries Service
National Oceanic and Atmospheric Administration
7600 Sand Point Way N.E.
Seattle, WA 98115

March 1988

This document is available to the public through:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

THIS PAGE INTENTIONALLY LEFT BLANK

CONTENTS

ABSTRACT

A surface ocean current model, OSCURS (Ocean Surface CURrent Simulations), has been developed as a tool in ecosystem-fisheries- oceanographic research. Wind and ocean surface drift are computed from historic daily sea level pressure fields (1946 to 1987) on a 40X104 grid over the North Pacific Ocean and Bering Sea. Total flow is the vector sum of long-term baroclinic geostrophic flow (0/2000 db) and surface wind drift. Model outputs include daily vector current fields or progressive vector drift tracks from any selected location over any selected time period. Model tuning and calibration results will appear in subsequent publications.

INTRODUCTION

Both theoretical and empirical knowledge of ocean currents have increased gradually over the past several decades. Using satellite navigation to determine latitude-longitude coordinates has significantly improved our ability to accurately define the position of ships and instruments in the ocean. Recently, for the first time, drifting buoys were deployed in the North Pacific Ocean for a long enough time (1-2 years) to illustrate and generally verify the geostrophic circulation patterns which had been derived from the long-term averaged internal density distributions (calculated from historic measurements of temperature and salinity versus pressure or depth). This has provided a basis for computer simulation models of mean surface currents on an ocean-wide scale. In an effort to gain more information about details of flow for fisheries-oceanography studies, fisheries management research, and ecosystem model inputs, the following first order approach was adopted. A numerical simulation model "OSCURS" (Ocean Surface CURrent Simulations) was developed and has shown good promise in its ability to assess historic, daily ocean circulation in the North Pacific Ocean and Bering Sea between 1946 and 1987. This post-World War II time period was chosen because consistently better gridded sea level pressure data were available.

Although ocean currents are affected by many factors, only the two major factors which determine the driving forces--the permanent, internal, thermohaline density field and the calculated local wind vectors--are used to generate surface currents in this model. Other factors such as tides, bathymetry, and atmospheric pressure effects, on sea level will be included in future model expansions as finer spatial resolution of details about flow over the shallow continental shelves is needed. Surface ocean currents considered here are composed from the vector sum of the local geostrophic flow and the local wind-induced flow. The model outputs are 1) ocean-wide, daily, vector fields of wind, wind current, geostrophic current, or total current or 2) progressive transport vectors from any selected spatial starting point or group of points over any time period from 1946 to 1987.

This report describes the detailed structure and mathematical basis for the OSCURS model including the (40X104) grid, setup, input, run instructions, the nature of computations, the daily run loop, and the output files created. The final section of the report contains the setup and run instructions for plotting model results on an ocean-wide scale or in zoom plot portions with graphics outputs on a Tektronics CRT or Calcomp Plotter. The Appendix contains listings of both Fortran programs, the model and the output graphics.

SET UP, INPUT, AND RUN INSTRUCTIONS

OSCURS Model (40X104) Grid

The (40X104) square computational grid of the model extends laterally across the North Pacific Ocean from the west coast of the U.S. (124°W) to southern Japan (130°E) and extends longitudinally southward from Bering Strait (67°N) to about latitude 30°N (Fig. 1). Note that the corner points of this grid are not necessarily the extremes of both latitude and longitude because of the nature of fitting a square, equal area grid to a portion of the nearly spherical Earth. Being a l/4 mesh subset of a portion of the standard U.S. Navy Fleet Numerical Oceanography Center (FNOC) (63X63) Northern Hemisphere grid, our (40X104) grid has the properties of the FNOC grid but only on a finer scale. The base FNOC grid was derived by centering an equally spaced, square 380 km mesh at the North Pole of a Northern Hemisphere polar stereographic map projection true at 60°N with the columns aligned parallel to longitude 170°W. Our smaller, average mesh size of about 90 km was chosen to provide a better spatial resolution while still preserving cross-ocean continuity for computing long-term progressive transport vector tracks. Again, due to the spherical geometry, the mesh size- varies slightly with latitude (from 95 km at 60°N to 83 km at 40°N). The formula used to compute the model grid length as a function of latitude in the pressure gradient calculation is

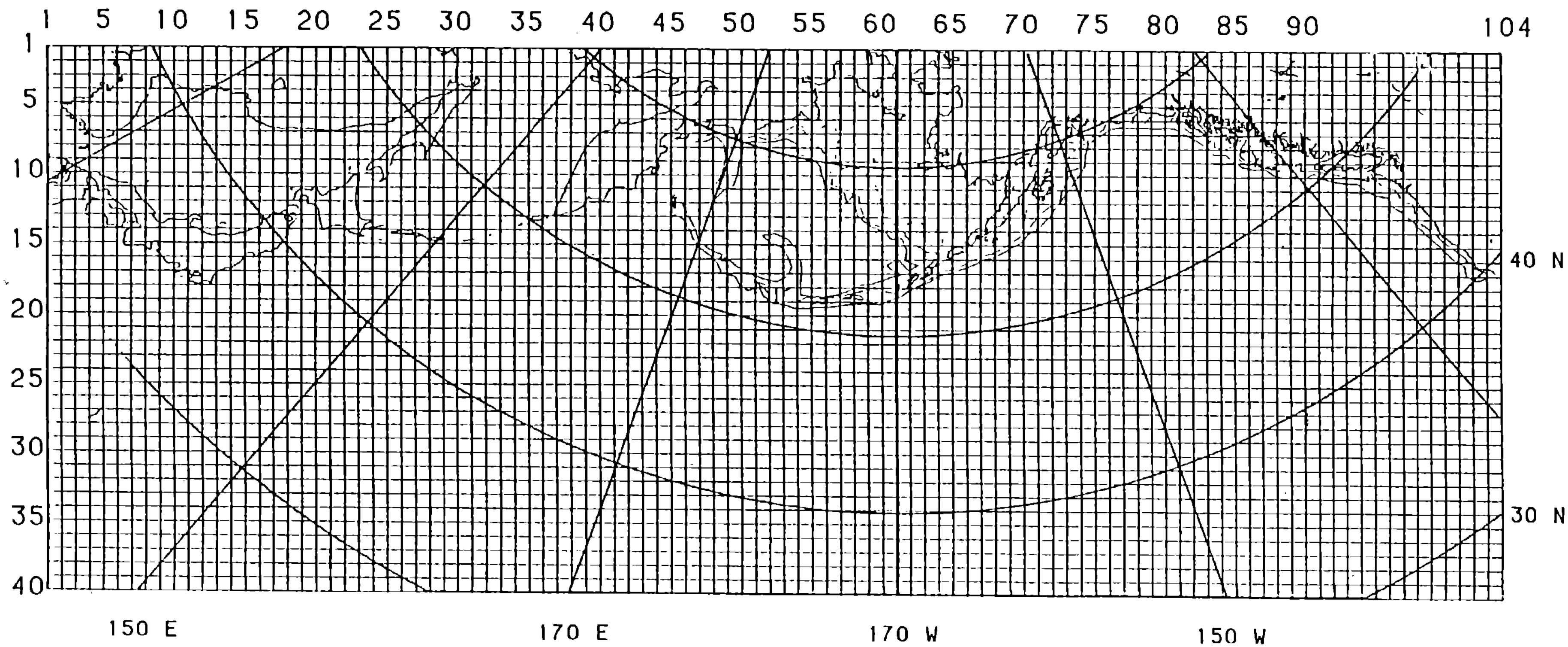

Figure 1.--Surface ocean current model (40x104) grid (95-83 km).

$$\text{Grid Length(km)} = 95.23 \left[\frac{1.0 + \sin(\text{latitude})}{1.0 + \sin(60)}\right]$$

A land-sea table (Fig. 2) is used to control the computations at different grid points related to their location over land or sea in the model. Values are assigned as follows: land=0, shallow ocean (<200m)=l, deep ocean (>200m)=2. Therefore, the land-sea table is used for tests to avoid zero points on the grid during both the computational and graphics output mode. Some land-sea values are set negative at points which are adjacent to the coastline for boundary condition calculations as described in the next section.

Input Coastline Boundary Conditions

To simulate boundary friction (zero slip at shoreline) and deflection of currents which impinge upon the coast at an angle all grid points within two grid lengths of the coastline were selected using the grid map (Fig. 1). Their grid coordinates (I,J) and the length (x-coordinate and y-coordinate) of the line normal to shore from each point were tabulated in grid space units and stored on disk Files 25 and 26. As these disk files are read into arrays, the corresponding (I,J) grid locations in the Land-Sea table are set negative to uniquely identify these particular points where rotation and deflection will be calculated in the subroutine CSTROT. The location of these points are indicated by the asterisks in Figure 3. The nature of the boundary modifications and calculations are discussed below under Total Currents with Boundary Effects.

LANDSEA 3

00000000000000000000000000000000000000000000000000000000011100000000000000000000000000000000000000000000	1
00000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000	2
00000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000	3
00000000000000000000000000000000000000000000000000000011100110000000000000000000000000000000000000000000	4
00000000000000000000000000000000000000000000000000000011111111110000000000000000000000000000000000000000	5
00000000000000000000000000000000000000000000000000000011111111000000000000000011110000000000000000000000	6
00000000000000000000000000000000000000000000012100122111111111000000000012222222100000000000000000000000	7
00000000000000000000000000000000000000000002222222222111111111000000000122222222222221000000000000000000	8
00000000000000000000000000000000000000000022222222222111111111000000011122222222222200021000000000000000	9
00000000000000000000000000000000000000000022222222222111111111110000010222222222222222222222221000000000	10
00000000000000000000000000000000000000002222222222222222111111111110000222222222222222222222222110000000	11
00000000000000000000000000000000000002222022222222222222111111111100112222222222222222222222222221000000	12
22210000000000000000000000000000000022222220222222222222211111111101122222222222222222222222222221000000	13
22222100000000000022000000000000002222222222222222222222222221111101122222222222222222222222222222100000	14
22222200000000002222222222211212222222222222222222222222222222111011222222222222222222222222222222210000	15
22222222100000002222222222222222222222222222222202222222222222221012222222222222222222222222222222221000	16
22222222200001222222222222222222222222222222222222222222222222102222222222222222222222222222222222222210	17
22222222221122222222222222222222222222222222222222222222222202222222222222222222222222222222222222222222	18
22222222222222222222222222222222222222222222222222220200222222222222222222222222222222222222222222222222	19
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	20
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	21
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	22
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	23
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	24
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	25
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	26
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	27
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	28
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	29
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	30
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	31
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	32
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	33
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	34
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	35
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	36
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	37
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	38
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	39
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222	40

Figure 2.-- Land-Sea table for 40x104 grid points (Land=0; Shelf water (<200m) =l, Deep water (>200m) =2).

LANDSEA 3

```
00000000000000000000000000000000000000000000000000 0*000000000*1*0000000000000000000000000000000000000000   1
0000000000000000000000000000000000000000000000000000*00000000*000000000000000000000000000000000000000000   2
000000000000000000000000000000000000000000000000000*000000000*000000000000000000000000000000000000000000   3
000000000000000000000000000000000000000000000000000000***00**0000000000000000000000000000000000000000000   4
00000000000000000000000000000000000000000000000*000000********000000000000000000000000000000000000000000   5
000000000000000000000000000000000000000000000000000000*11*111*000000000000000****00000000000000000000000   6
000000000000000000000000000000000000000000000***00****11111*0000000000*********0000000000000000000000000   7
0000000000000000000000000000000000000000000**********2111111**0000000000****2222****10000000000000000000   8
0000000000000000000000000000000000000000000**2222*2222111111*0000000*****222222****000210000000000000000   9
000000000000000000000000000000000000000000*222222222222221111111***0000*0**22222222************000000000  10
00000000000000000000000000000000000000000*0***222222222222111111*****00*0**222222222*************0000000  11
000000000000000000000000000000000000000000*****022222222211111111***0****222222222222222222222*****00000  12
****00000000000000000000000000000000****22202222222222222211111110**0****22222222222222222222222***000000  13
222***00000000000**00000000000000****2222222222222222222222211**0***2222222222222222222222222222***00000  14
22222*00000000000*****************22222222222222222222222222222**0***2222222222222222222222222222***0000  15
22222****0000000**22222222222222222222222222222222022222222222**0***22222222222222222222222222222****00  16
22222222*0000***222222222222222222222222222222222222222222222210*2*22222222222222222222222222222222**2*0  17
22222222******22222222222222222222222222222222222222222222222202222222222222222222222222222222222222***2  18
22222222222222222222222222222222222222222222222222202002222222222222222222222222222222222222222222222222  19
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  20
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  21
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  22
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  23
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  24
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  25
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  26
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  27
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  28
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  29
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  30
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  31
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  32
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  33
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  34
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  35
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  36
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  37
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  38
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  39
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222  40
```

Figure 3.--Land-Sea table for 40x104 grid points (Land=0; Sea=1,2,or Coastline=* or -1).

Input Permanent Geostrophic Current File

A major component of total ocean flow is the geostrophic surface current computed from internal density gradients down to an assumed level of no motion. This is an estimate of long-term permanent flow. Ideally the model will be revised in the future to have seasonal or even monthly fields of permanent geostrophic currents, but due to the lack of year round data in oceanic areas and the considerable amount of time required to prepare fields with shallow water extrapolations from deep reference levels, a constant, long-term current field was used.

To obtain geostrophic currents, anomaly of Dynamic Height fields (0/2000 db) were computed from 1 X 1 degree historical means of temperature and salinity versus depth from the Bauer-Robinson Numerical Atlas of Ocean Basins (Bauer and Robinson 1985). Horizontal numerical interpolations were performed using a I-point Bessel central difference formula fit to the 1 X 1 degree square data to obtain data at the locations (latitude-longitude) of the model grid points.

Subroutine DELTAD accesses these data through File 33 and 34 (Anomaly of Dynamic Height data,) to compute the U (east-west) and V (north-south) components of geostrophic current (cm/sec) by standard methods (LaFonde 1951) and store these data in permanent File 38. DELTAD need not be called in subsequent runs of the model as the program then reads File 50 which is the equivalent of File 38.

Input of Sea Level Pressure Data Fields

The key input variable field which drives the model is daily sea level pressure data which was obtained from files of the U.S. Navy, FNOC Monterey, California, over the portion of the standard FNOC 380 km (63X63) grid between latitude 50N and 68°N and longitude 100°W and 120°E. Data processing of these data tapes into disk files containing daily sequences of sea level pressure data by month, season, or year for the Burroughs computer was documented by Ingraham et al. 1983. Because the OSCURS model grid is a l/4-mesh, higher resolution subset, interpolations are performed using a 4-point spline central difference numerical surface fitting routine each time a new input daily field of sea level pressure is read. These data are the basis for the wind and wind current computations below.

Input Starting Locations for Progressive Vector Tracks

Before the main computation loop of the model is entered, it is necessary to choose the number of progressive vector tracks that you want to save on disk File 60 and their starting grid coordinates (I,.J). Two methods are provided, but in this version of the model the second method which saves every fourth grid point is skipped. Every grid point could be saved but storage space is economized here to show only the general capacities of the model. The first method is to put a few selected (I,J) starting grid points into File 28 (Table 1) which the model reads and counts. The counter IEND (line 412) must contain the exact number of these starting points. Although integer grid points

Table 1.--Selected starting grid points (I,J) for progressive transport vector computations on 40x104 grid (File 28, Title="P/l").

Record Number	I	J
1	10	73
2	11	73
3	11	74
4	12	74
5	12	75
6	13	75
7	14	76
8	15	77
9	16	78
10	17	79
11	18	80
12	19	81
13	20	82
14	21	83
15	22	84
16	23	85
17	24	86
18	20	12
19	28	16
20	24	20
21	20	24
22	16	28
23	32	28
24	20	36
25	28	36
26	36	36
27	20	44
28	28	52
29	36	52
30	28	64
31	36	64
32	16	73
33	28	76
34	36	76
35	8	80
36	12	84
37	24	88
38	32	88
39	16	96
40	24	96

are initially set up as starting locations, they may be reset to any decimal value or fraction of the way between grid points (see line 450-454 in the OSCURS model, Appendix 1).

Input Control Parameters and Run Instructions

Starting the model requires compiling the Fortran code and saving the object code interactively through CANDE (the standard Burroughs interactive language) followed by a "START M/FINAL/WFL" command. The Work Flow Language (WFL) program (M/FINAL/WFL) listed below (Program 1) shows the line numbers of the input and output file names and control parameters which must be edited to give the desired run conditions before the START command is given.

The output file names (lines 12-34 in M/FINAL/WFL) should include as much descriptive information about the runs they represent as possible. Although these names are long, the extra typing is quite important to clearly distinguish (now and at a later time) the many possible combinations of run parameters, especially when many runs of different time periods are to be compared.

Input File 2 is FNOC daily sea level pressure data for the desired time period. Retrieve these data from tape one year at a time in separate files of daily data fields from 1 January to 31 December. Edit into a smaller file if model starting month is greater than 1 to save reading time and disk space.

Input File 28 is a sequence of (I,J) grid point data specifying the nearest model grid point to the desired starting

Program 1. Listing of Work Flow Language (WFL) which starts the model run (File Title="M/FINAL/WFL").

```
?BEGIN JOB INGRAHAM/M/FINAL/W10DD00/S2X2/40X104;
 QUEUE=10;
 RUN OBJECT/M/FINAL;
 FILE FILE2=FNWC/DSLP/78/M7TOM12;
 FILE FILE28=P/FIG/40X104;
 FILE FILE50=CURRENTS/40X104/DYNCURR/S2X2;
 FILE FILE51=CURRENTS/40X104/DSLP/78/M7TOM12/D17START/W10DD00/S2X2
             ON SCRATCH;
 FILE FILE52=CURRENTS/40X104/WIND/78/M7TOM12/D17START/W10DD00/S2X2
             ON SCRATCH;
 FILE FILE53=CURRENTS/40X104/SFCOCN/78/M7TOM12/D17START/W10DD00/S2X
             ON SCRATCH;
 FILE FILE54=CURRENTS/40X104/W10DD00/78/M7TOM12/D17START/CSTROTATD/
             ON SCRATCH;
 FILE FILE55=CURRENTS/40X104/W10DD00/78/M7TOM12/D17START/S2X2
             ON SCRATCH;
 FILE FILE60=CURRENTS/40X104/PROGRESVECT/78/M7TOM12/D17START/W10DD0
             ON SCRATCH;
 FILE FILE5(KIND=READER);
?DATA
          MODEL CONTROL CARDS (CHANGE LEADING CONSTANTS IN CANDE)
   78=====YEAR OF DSLP  FILE
    7=====STARTING MONTH FOR RUN
   12=====STOP AT END OF THIS MONTH
  200=====NUMBER OF TOTAL DAYS IN RUN
  1.0=====WIND SPEED MULTIPLIER (WFAC)
  0.0=====GEOSTROPHIC CURRENT MULTIPLIER (DDFAC)
    0=====FLAG TO CREATE NEW GEOSTROPHIC CURRENT  FILE (0=NONE)
    1=====FLAG TO STOP PRINTOUT OF FIELDS IN LOOP
    1=====FLAG TO SUPRESS DISK STORAGE OF SLP,WIND,OC CUR PRE ROT
    1=====FLAG TO SUPRESS DISK STORAGE OF EACH TOTAL CURR FIELD
?
 COPY CURRENTS/40X104/PROGRESVECT/78/M7TOM12/D17START/W10DD00/S2X2/
      FROM SCRATCH(PACK) TO TYVO(PACK)
?END JOB
```

position for each progressive vector to be stored on disk for the final plot.

Input File 5 contains input parameters which control the run time, the value of wind and current tuning factors, the amount of printing, and whether disk files are saved. Prior to run. edit this file and set the leading constant on each line according to your desired run conditions and the following instructions.

Line 21, YEAR-1900. This value must correspond to the same year as the data in File 2. Each run is limited to data from one calendar year only (at this time).

Line 24, STARTING MONTH. Program will read through the daily SLP data file and then start the computation loop when the starting month is encountered.

Line 23, STOP MONTH. Run will stop at the end. of this month. To stop within a. month see next line.

Line 24, RUN DAYS. Total number of days in run loop is used in stopping on a specific day. Check to be sure this number is greater than or equal to the time from start month to end month if a stop within a month is not desired.

Line 25, WIND TUNING FACTOR. In numerical experimenting with output vectors, different weighting factors may be applied to wind speed which will generate a corresponding linear change in surface ocean current due to wind.

Line 26, DELTAD TUNING FACTOR. This is a direct linear multiplier of each component of the permanent geostrophic current.

Line 27, NEW DELTAD FILE FLAG. For the initial run or in case the original anomaly of dynamic height field is changed (e.g., new reference level), this flag is set to "1" which results in the overwriting of file 38 with new U and V fields of permanent geostrophic current. In any subsequent run this flag is set to "0" and the program reads the previously stored version of the same name, File 50 with the title, "CURRENTS/40X104/DYNCURR/F".

Line 28, NO PRINT. Set this flag to "0" to receive a printout of each daily SLP (Sea Level Pressure), wind speed, U and V ocean currents due to wind before rotation of near coastal vectors, and U and V total rotated current fields. For runs longer than a few weeks set the flag to "1" to avoid reams of printing.

Line 29, NO STORE. Set this flag to "1" to suppress storage of disk File numbers 51, 52, 53, and 55 leaving only the final total current fields stored in file 54. Set this flag to "0" to store all files on scratch disk.

Line 30, NO STORE FINAL. Set this flag to "1" to suppress storage of File 54, the final output daily current fields. Although these files require a lot of space these are the main results which should be archived on tape to avoid future reruns.

In output Files 51, 52, 53, 54, 55, and 60 the editing of file titles should include changes in year, months, start day, wind factor (Wl0), and delta-d factor (DDl0) before the run is started. The "10" in "W10" and "DDl0" indicates the tuning factors were both equal to "1.0" or neutral. Examples of other possible factors are "05"=0.5, "18"=1.8, and "00"=0.0. File 60 which contains the U and V components of all the transport vector starting and ending points in grid units is automatically stored on both SCRATCH disk and TYVO disk.

DAILY SEQUENCE LOOP AND OUTPUT FILES

The main function of this simulation model is to compute a surface drift current vector (cm/sec) at each grid point in the grid on a daily basis forming a daily series of velocity fields from which progressive vectors of drift tracks can be estimated. The following five steps are necessary to generate one field: 1) read in sea level pressure, 2) compute wind, 3) compute wind current, 4) tune and sum velocity components, and 5) add boundary effects. The sixth step is to compute the surface transport vectors (distance traveled per day) at each of the chosen starting points (the end points of these computed vectors become the starting points for the next day's vectors). Finally, vector velocity field components and transport vector start and end points in grid unit values are stored on separate disk files for plotting by the plot program. This procedure is repeated for the specified number of days (NDAYS) in the loop. Details of these steps and theoretical considerations are described below for one typical pass through the loop.

Sea Level Pressure

Subroutine READER (IMO, K, IYD, ITYP, SLP, SLPl) reads one 0000Z daily FNOC sea level pressure field from FILE 2 and returns the following variables, where IMO is the month, K is the year, IDY is the day and ITYP is the flag which normally equals zero but equals 999 at end of File 2. Also returned to the main program are the one-dimensional array SLP and the two-dimensional array SLPl which contain the data for processing and printing in the program. A test is made to be sure the month read is the month desired. If not, another day of data is read in time sequence until the correct month is found. Units are converted to millibars from packed units (the difference from 1000.0 in tenths of millibars).

Conversion. of data array, SLP, from the FNOC (20X44) grid to the model (40X104) grid array, SLP1, is performed by a I-point Bessel or Spline numerical curve fitting routine in two dimensions (a data surface fit). Comparisons between both interpolation routines indicate either one performs quite well. The interpolated field, array T1, is stored on disk File 51 by setting the flag NOSTOR=0.

Wind

In subroutine WIND(T1,LS1,ALAT2,WXX,WYY) three arrays are passed, where T1 is the SLP field which was just interpolated to the model grid, LS1 is a land-sea table with all values = 1 (computations are performed at each grid point even over land), and ALAT2 has the latitudes of each grid point. Two arrays (WXX and WYY) are returned which contain the computed U and V components of the wind vectors, respectively. Computations are done in two steps.

First, a pure geostrophic wind is calculated from the balance between the atmospheric pressure gradient force and the Coriolis force to the right of the wind looking in the direction of the wind. Since friction is absent in this theoretical approximation, wind vectors are parallel to the horizontal pressure isobars. Solving for wind the formula is

$$W = \frac{1}{p\ f}\left[\frac{dP}{dL}\right]$$

where W is the wind vector (m/sec), p is the density of air, 1.22 kg/m3, f is the Coriolis parameter, 1.458X10-4 X sin(latitude), $\frac{dP}{dL}$ is the atmospheric pressure change per unit grid length which is computed using a 5-gridpoint central difference numerical curve fitting routine.

In the more natural case friction acts to slow the wind in the boundary layer near the ocean surface and causes the wind to be deflected to the left of the geostrophic wind with cross-isobar flow from higher toward lower pressure. 'Since the conditions affecting friction vary considerably in time and space, there have been many experiments to determine the effect of friction on wind flow structure in the boundary layer. Results in general show that at low wind speeds (<5 m/sec) in mid-latitude the deflection is about 20 degrees to the left and the magnitude reduction factor is about 85% of the pure geostrophic wind.

In the model variations of frictional effects with latitude and wind speed are also included. Values computed from the function used in the model shown in Table 2 indicate a further reduction in speed from 85% to about 80% away from mid-latitudes (25°-45°N), increases In the angle of deflection toward the equator, and decreases in the angle of deflection at higher speeds. Next, these resultant winds are used to calculate a commensurate surface ocean flow field.

Table 2. --Reduction factor and angle of deflection of the wind to the left of pure geostrophic wind.

		Wind speed (m/sec)										
		0	2.8	5.7	8.5	11.3	14.1	17.0	19.8	22.6	25.5	28.3
Latitude	Reduction	Angle of deflection (°)										
67.0	0.801	17.5	17.4	17.2	16.9	16.4	15.8	15.0	14.1	13.0	11.8	10.5
66.0	0.803	17.6	17.5	17.3	16.9	16.4	15.8	15.0	14.1	13.1	11.9	10.5
65.0	0.806	17.6	17.6	17.4	17.0	16.5	15.9	15.1	14.2	13.1	11.9	10.5
64.0	0.808	17.7	17.6	17.4	17.1	16.6	15.9	15.1	14.2	13.1	11.9	10.5
63.0	0.810	17.8	17.7	17.5	17.1	16.6	16.0	15.2	14.2	13.1	11.9	10.5
62.0	0.812	17.9	17.8	17.6	17.2	16.7	16.0	15.2	14.3	13.2	11.9	10.5
61.0	0.814	17.9	17.9	17.6	17.3	16.8	16.1	15.3	14.3	13.2	11.9	10.5
60.0	0.817	18.0	18.0	17.7	17.4	16.8	16.2	15.3	14.4	13.2	12.0	10.5
59.0	0.819	18.1	18.0	17.8	17.4	16.9	16.2	15.4	14.4	13.3	12.0	10.6
58.0	0.821	18.2	18.1	17.9	17.5	17.0	16.3	15.5	14.5	13.3	12.0	10.6
57.0	0.823	18.3	18.2	18.0	17.6	17.1	16.4	15.5	14.5	13.4	12.0	10.6
56.0	0.826	18.4	18.3	18.1	17.7	17.1	16.4	15.6	14.6	13.4	12.1	10.6
55.0	0.828	18.5	18.4	18.2	17.8	17.2	16.5	15.7	14.6	13.4	12.1	10.6
54.0	0.830	18.6	18.5	18.3	17.9	17.3	16.6	15.7	14.7	13.5	12.1	10.6
53.0	0.832	18.7	18.6	18.4	18.0	17.4	16.7	15.8	14.8	13.5	12.2	10.6
52.0	0.834	18.8	18.7	18.5	18.1	17.5	16.8	15.9	14.8	13.6	12.2	10.7
51.0	0.837	18.9	18.8	18.6	18.2	17.6	16.9	16.0	14.9	13.7	12.2	10.7
50.0	0.839	19.0	19.0	18.7	18.3	17.7	17.0	16.0	15.0	13.7	12.3	10.7
49.0	0.841	19.2	19.1	18.8	18.4	17.8	17.1	16.1	15.0	13.8	12.3	10.7
48.0	0.843	19.3	19.2	19.0	18.5	17.9	17.2	16.2	15.1	13.8	12.4	10.8
47.0	0.846	19.4	19.3	19.1	18.7	18.0	17.3	16.3	15.2	13.9	12.4	10.8
46.0	0.848	19.6	19.5	19.2	18.8	18.2	17.4	16.4	15.3	14.0	12.5	10.8
45.0	0.850	19.7	19.6	19.4	18.9	18.3	17.5	16.5	15.4	14.0	12.5	10.8
44.0	0.850	19.8	19.8	19.5	19.0	18.4	17.6	16.6	15.5	14.1	12.6	10.9
43.0	0.850	20.0	19.9	19.6	19.2	18.6	17.8	16.8	15.6	14.2	12.7	11.0
42.0	0.850	20.2	20.1	19.8	19.3	18.7	17.9	16.9	15.7	14.4	12.8	11.1
41.0	0.850	20.3	20.2	19.9	19.5	18.9	18.0	17.0	15.8	14.5	12.9	11.2
40.0	0.850	20.5	20.4	20.1	19.6	19.0	18.2	17.2	16.0	14.6	13.0	11.3
39.0	0.850	20.6	20.6	20.3	19.8	19.2	18.3	17.3	16.1	14.7	13.1	11.4
38.0	0.850	20.8	20.7	20.4	20.0	19.3	18.5	17.5	16.2	14.8	13.2	11.5
37.0	0.850	21.0	20.9	20.6	20.1	19.5	18.6	17.6	16.4	15.0	13.4	11.6
36.0	0.850	21.2	21.1	20.8	20.3	19.7	18.8	17.8	16.5	15.1	13.5	11.7
35.0	0.850	21.4	21.3	21.0	20.5	19.8	19.0	17.9	16.7	15.2	13.6	11.8
34.0	0.850	21.6	21.5	21.2	20.7	20.0	19.1	18.1	16.8	15.4	13.7	11.9
33.0	0.850	21.8	21.7	21.4	20.9	20.2	19.3	18.3	17.0	15.5	13.8	12.0
32.0	0.850	22.0	21.9	21.6	21.1	20.4	19.5	18.4	17.1	15.7	14.0	12.1
31.0	0.850	22.2	22.1	21.8	21.3	20.6	19.7	18.6	17.3	15.8	14.1	12.2
30.0	0.850	22.4	22.3	22.0	21.5	20.8	19.9	18.8	17.5	16.0	14.3	12.3
29.0	0.850	22.7	22.6	22.2	21.7	21.0	20.1	19.0	17.7	16.1	14.4	12.5
28.0	0.850	22.9	22.8	22.5	22.0	21.2	20.3	19.2	17.8	16.3	14.6	12.6
27.0	0.850	23.1	23.0	22.7	22.2	21.5	20.5	19.4	18.0	16.5	14.7	12.7
26.0	0.850	23.4	23.3	23.0	22.4	21.7	20.8	19.6	18.2	16.7	14.9	12.9
25.0	0.850	23.6	23.5	23.2	22.7	21.9	21.0	19.8	18.4	16.8	15.0	13.0
24.0	0.842	23.9	23.8	23.5	23.0	22.2	21.3	20.1	18.7	17.2	15.4	13.4
23.0	0.834	24.2	24.1	23.8	23.2	22.5	21.6	20.4	19.1	17.5	15.7	13.7
22.0	0.826	24.5	24.4	24.1	23.5	22.8	21.9	20.7	19.4	17.8	16.1	14.1
21.0	0.818	24.8	24.7	24.4	23.8	23.1	22.2	21.1	19.7	18.2	16.4	14.5
20.0	0.810	25.1	25.0	24.7	24.1	23.4	22.5	21.4	20.1	18.5	16.8	14.8

Current Due to Wind

Computation of the surface current due to wind, like the computation of wind, is done with emperical methods in two parts. First, the pure drift current vector without friction is calculated, then the vector is corrected for frictional effects (due to the drag of the underlying water) which causes an angle of deflection of the current to the right of the wind looking in the direction of flow. With very high wind speeds the angle of deflection becomes very small and it also becomes smaller with decreasing latitude.

Although the surface current in the upper 3 m of water has been found to be generally about 2% of the average climatological (usually the monthly mean) wind speed, what is desired here is a relationship between the surface current and the instantaneous wind. For part one of our computations the formula of Witting (1909) satisfies this purpose within the limitations previously stated

$$C = k\sqrt{W}$$

where C is the speed (cm/sec) of the surface current due to wind, k is the coefficient of proportionality which equals 4.8 when it includes both the speed of the mixed layer and mass transport by waves at the surface, and W is the wind speed (m/sec).

In part two (frictional effects on wind generated surface current) the angle of deflection (BETA) is computed from a newly designed three part function which starts with the Hela (1952) relationship:

$$BETA = BETA1 - K2\sqrt{W}$$

where BETA1 is the initial constant angle of deflection of 34 degrees for wind speeds approaching zero and K2 is a constant of 7.5. In the second part of the function, BETA1 is modified to decrease linearly with latitude with a slope of 0.2933 which was determined from a graphic fit of data for a constant wind speed of 6 knots between 30°N and 70°N (Hubert and Laevastu 1967, p.11)

$$BETA1 = 34.0 - (DLAT \times 0.2933)$$

where DLAT equals 67 -latitude (ie. the difference between the top latitude of the model (67°N) and the variable latitude. The final part of the computation used the constraint that the angle of deflection (BETA) approached zero at 20.59 m/sec (40 knots) at all latitudes. With BETA equal to zero and solving for K2 this permits an adjustment of K2 for the change in BETA1 with latitude with the relationship

$$K2 = \frac{BETA1}{\sqrt{20.59}}$$

Thus, the range of values of the angle of deflection of the surface current to the right of the wind over the latitudes (20°-67°N) of the model for wind speeds 0 to 40 knots as computed by the combined function are shown in Table 3. Now that we have a permanent flow and a flow due to wind we proceed to tune and combine the components of flow.

Tuning and Vector Summation of Currents

In this experimental model, it is desirable to have tuning factors because of the remaining small uncertainties in the theoretical computations of ocean currents. Thus, any

Table 3.--Angle of deflection of surface current to the right of the wind as a function of latitude and wind speed.

Wind speed	0	5	10	15	20	25	30	35	40	(kts)
	0.00	2.57	5.15	7.72	10.30	12.87	15.44	18.02	20.59	(m/sec)
Latitude	Angle of deflection (°)									
67.0	34.00	21.98	17.00	13.18	9.96	7.12	4.56	2.20	0.00	
66.0	33.71	21.79	16.85	13.07	9.87	7.06	4.52	2.18	0.00	
65.0	33.41	21.60	16.71	12.95	9.79	7.00	4.48	2.16	0.00	
64.0	33.12	21.41	16.56	12.84	9.70	6.94	4.44	2.14	0.00	
63.0	32.83	21.22	16.41	12.72	9.61	6.87	4.40	2.12	0.00	
62.0	32.53	21.03	16.27	12.61	9.53	6.81	4.36	2.10	0.00	
61.0	32.24	20.84	16.12	12.50	9.44	6.75	4.32	2.08	0.00	
60.0	31.95	20.65	15.97	12.38	9.36	6.69	4.28	2.06	0.00	
59.0	31.65	20.46	15.83	12.27	9.27	6.63	4.24	2.04	0.00	
58.0	31.36	20.27	15.68	12.16	9.19	6.57	4.20	2.03	0.00	
57.0	31.07	20.08	15.53	12.04	9.10	6.51	4.16	2.01	0.00	
56.0	30.77	19.89	15.39	11.93	9.01	6.44	4.12	1.99	0.00	
55.0	30.48	19.70	15.24	11.82	8.93	6.38	4.08	1.97	0.00	
54.0	30.19	19.51	15.09	11.70	8.84	6.32	4.04	1.95	0.00	
53.0	29.89	19.32	14.95	11.59	8.76	6.26	4.01	1.93	0.00	
52.0	29.60	19.14	14.80	11.47	8.67	6.20	3.97	1.91	0.00	
51.0	29.31	18.95	14.65	11.36	8.58	6.14	3.93	1.89	0.00	
50.0	29.01	18.76	14.51	11.25	8.50	6.08	3.89	1.87	0.00	
49.0	28.72	18.57	14.36	11.13	8.41	6.01	3.85	1.85	0.00	
48.0	28.43	18.38	14.21	11.02	8.33	5.95	3.81	1.84	0.00	
47.0	28.13	18.19	14.07	10.91	8.24	5.89	3.77	1.82	0.00	
46.0	27.84	18.00	13.92	10.79	8.15	5.83	3.73	1.80	0.00	
45.0	27.55	17.81	13.77	10.68	8.07	5.77	3.69	1.78	0.00	
44.0	27.25	17.62	13.63	10.56	7.98	5.71	3.65	1.76	0.00	
43.0	26.96	17.43	13.48	10.45	7.90	5.65	3.61	1.74	0.00	
42.0	26.67	17.24	13.33	10.34	7.81	5.58	3.57	1.72	0.00	
41.0	26.37	17.05	13.19	10.22	7.72	5.52	3.53	1.70	0.00	
40.0	26.08	16.86	13.04	10.11	7.64	5.46	3.49	1.68	0.00	
39.0	25.79	16.67	12.89	10.00	7.55	5.40	3.45	1.67	0.00	
38.0	25.49	16.48	12.75	9.88	7.47	5.34	3.42	1.65	0.00	
37.0	25.20	16.29	12.60	9.77	7.38	5.28	3.38	1.63	0.00	
36.0	24.91	16.10	12.45	9.65	7.30	5.22	3.34	1.61	0.00	
35.0	24.61	15.91	12.31	9.54	7.21	5.16	3.30	1.59	0.00	
34.0	24.32	15.72	12.16	9.43	7.12	5.09	3.26	1.57	0.00	
33.0	24.03	15.53	12.01	9.31	7.04	5.03	3.22	1.55	0.00	
32.0	23.73	15.34	11.87	9.20	6.95	4.97	3.18	1.53	0.00	
31.0	23.44	15.15	11.72	9.09	6.87	4.91	3.14	1.51	0.00	
30.0	23.15	14.96	11.57	8.97	6.78	4.85	3.10	1.50	0.00	
29.0	22.85	14.77	11.43	8.86	6.69	4.79	3.06	1.48	0.00	
28.0	22.56	14.58	11.28	8.75	6.61	4.73	3.02	1.46	0.00	
27.0	22.27	14.40	11.13	8.63	6.52	4.66	2.98	1.44	0.00	
26.0	21.97	14.21	10.99	8.52	6.44	4.60	2.94	1.42	0.00	
25.0	21.68	14.02	10.84	8.40	6.35	4.54	2.90	1.40	0.00	
24.0	21.39	13.83	10.69	8.29	6.26	4.48	2.87	1.38	0.00	
23.0	21.09	13.64	10.55	8.18	6.18	4.42	2.83	1.36	0.00	
22.0	20.80	13.45	10.40	8.06	6.09	4.36	2.79	1.34	0.00	
21.0	20.51	13.26	10.25	7.95	6.01	4.30	2.75	1.32	0.00	
20.0	20.21	13.07	10.11	7.84	5.92	4.23	2.71	1.31	0.00	

combination of weighting factors may be selected here as a linear multiplier of either the permanent geostrophic current component or the wind current component. These adjustments will be particularly useful in computing longer term effects on progressive vector tracks for a variety of different conditions and comparing these results to obtain the best match to actual measurements from drifters that have been tracked in the ocean. Refer to Run Instructions section to set up tuning factors such as doubling the wind effect or setting wind equal zero to observe the pure permanent flow.

The total current vector field is now produced as a vector sum of the tuned geostrophic and wind current fields components of flow. This field is stored on Disk File 55 for later comparison to the field in which the vectors near shore have been adjusted for boundary conditions as described in the next section.

Total Currents With Boundary Effects

The final step in obtaining a vector field of total currents representative of this particular day in history is to call subroutine CSTROT(KYR, IMO, IDY, OCUS, OCVS, NOSTOF). This subroutine modifies the vectors within two grid lengths of the coast all along the entire coastline of the model for simulated boundary effects. These points have been preselected and read in as described above in the input section.

In the subroutine CSTROT current vectors at these points will be reduced in magnitude by the exponential function

$$\text{REDFAC (\%)} = 1.0 - (1/e^{x})$$

where x=GDIST(I,J), the normal distance from the grid point to the coastline in grid units.

Next the angle of the current vector is rotated depending upon its angle of incidence upon the coast relative to the line normal to shore. Looking toward shore, normal to the coastline trend, the current vectors which point to the left are rotated further toward the left and the vectors which point to the right are rotated farther toward the right. The maximum angle of rotation (45 degrees) is also diminished with distance from shore by the exponential factor, $1/e^{x}$, and a new function was designed to maximize the angle of rotation at an incident angle of 30 degrees. Away from 30 degrees the angle of rotation' is decreased proportionately to zero as the angle of incidence of zero degrees or 90 degrees are approached.

Transport (Progressive) Vectors

Now that a final velocity field is available a transport vector may be computed at any point on the grid, even between grid points. Transport vectors represent the distance traveled in one day proceeding at the velocity of the calculated surface current at that location. The starting point and end point of the vector are specified in grid units (I,J pairs), and the end point of this days vector is saved to be the starting point of tomorrows transport vector during the next pass through the loop.

The unique part of this routine is that the velocity at each starting point each time through the loop is interpolated spatially within the field using the spline surface fit. This gives a much greater degree of precision to progressive vector tracks formed by connecting these successive transport vectors for NDAYS compared to the tracks formed by the simpler method of connecting the transport vectors from a starting grid points which remains fixed in the field. Each day the transport vectors from each of the selected starting points are stored in the same order on Disk File 60. Thus, it is important to keep track of the number of starting points selected for later entry in the plot program. Note, that the plot program is organized to read and plot all the first day vectors then all the. second day vectors and continue in this manner, not to plot one entire progressive vector track at a time followed by the next entire track, etc.

The loop is now complete and will be entered again for another day of computations provided another day of sea level pressure data was scheduled by the control cards of the WFL deck. Output files are added onto each time through the loop and are "CLOSED" only after NDAYS when this program is stopped. Refer to the plot program instructions below for graphic output of results on a Tektronics 4014 terminal or CALCOMP Plot.

GRAPHICS OUTPUT PROGRAMS

Thanks to the availability of the "GRIDS" package of Fortran subroutines which was designed for the Burroughs 7811 computer to graphically display gridded data in geographical coordinates (latitude-longitude) on a Tektronix CRT Terminal or Calcomp Plotter (Swan 1983), the task of producing maps of our square grid data fields was greatly facilitated. The following describes our Fortran program which uses the CONTOUR and VECTOR graphics displays of GRIDS to show the model results first on an ocean-wide (40X104) grid then on a zoom (25X35) grid for the Gulf of Alaska.

Single day results of particular interest are the spatial relationships found among sea level pressure contours, wind vectors, and wind generated surface ocean current vectors. Choices of plots are 1) any combination of the five, single day vector fields or 2) one map of progressive transport vector tracks versus time. These are plotted for a specified number of days (NDAYS). A second version of the same program with more localized geographic coordinates is used for zoom plots (25X35) such as the Gulf of Alaska version presented here. Selections of which fields and the number to be plotted are made by editing your choices into the WFL code of the run job file each time the program is run. Now it is appropriate to examine the basic results of the model on an ocean-wide graphics scale.

Plots on 40X104 Grid

Ocean wide plots over the entire 40X104 grid are produced by the program "M/PLOT/AUTO" (listed in Appendix 2). If any modifications are made to this program the GRIDS subroutines must be bound again to the new object code by running the WFL job "M/PLOT/AUTO/BINDER" (Program 2). Plots are obtained by running the WFL job "M/PLOT/AUTO/WFL" (Program 3) after editing in your selections. The following describes the sequential flow of the plot program: the setup, the NDAYS plotting loop, and making your selections.

Setup within the Fortran code includes the standard GRIDS initialization with the reading of latitudes, longitudes (disk Files 30 and 31), and the land-sea table (disk Files 13 and 14) of the (40X104) grid and a call to STGRID to set up common arrays for all the GRIDS subroutines. Then latitude lines on the map are computed or read from previously stored disk file (set flag NOLATS=l to read disk File 36) as is the coastline file (set flag IFAST=0 to read File 71) and the 50, 100. 200, and 2000 m bathymetry files (set flag NOCONV=l to read Files 66, 67, 68, and 69). The alternative to reading the existing files is included here to allow for future versions which change the map size or area of interest and require recalculation of the map parameters. To create and store new disk files of latitude lines, coastline, and bathymetry just reverse the values set for the flags (NOLATS=0 IFAST=l, AND NOCONV=O). Next, the initial starting points for progressive vectors are read from File 28, and the permanent geostrophic ocean U and V current vector fields are read in from

Program 2.--Listing of Binder Job which binds and compiles the outputs for the 40X104 grid (File Title="M/PLOT/AUTO/BINDER").

```
?BEGIN JOB INGRAHAM/BINDER/M/PLOT/AUTO;
 QUEUE=20;
COMPILE OBJECT/M/PLOT/AUTO FORTRAN LIBRARY;
COMPILER FILE TAPE(KIND=DISK,TITLE=M/PLOT/AUTO);
?FORTRAN DATA
$SET MERGE
$RESET LIST
?
BIND M/PLOT/AUTO/BOUND BINDER LIBRARY;
BINDER FILE HOST(TITLE=OBJECT/M/PLOT/AUTO);
?BINDER EBCDIC
$RESET LIST
BIND=FROM GRIDS40104/=,(REFM0070)REFMOFF/= ON TYVO,
           (REFM0070)PLOTCOMP87/= ON TYVO;
?END JOB
```

Program 3.--listing of Word Flow Language (WFL) Program which runs the graphic outputs of the 40X104 grid (File Title="M/PLOT/AUTO/WFL").

```
?BEGIN JOB INGRAHAM/M/PLOT/PGRV/78/BOUND;
 QUEUE=20;
 RUN M/PLOT/AUTO/BOUND;
 FILE FILE10=DIR96202;
FILE FILE28=P/FIG/40X104;
FILE FILE51=CURRENTS/40X104/DSLP/78/M7TOM12/D17START;
FILE FILE52=CURRENTS/40X104/WIND/78/M7TOM12/D17START;
FILE FILE53=CURRENTS/40X104/SFCOCN/78/M7TOM12/D17START;
FILE FILE54=CURRENTS/40X104/WOODD10/78/M7TOM12/D17START/CSTROTATD;
FILE FILE55=CURRENTS/40X104/WOODD10/78/M7TOM12/D17START;
FILE FILE60=CURRENTS/40X104/PROGRESVECT/78/M7TOM12/D17START/W10DD20
        /FIG;
FILE FILE5(KIND=READER);
?DATA
   1.0=====EVERY GRID POINT PLOT SLP WITH CURRENTS
   0.0=====1. PLOT WIND VECTORS (1.0=PLOT)
   0.0=====2. PLOT WIND CURRENT VECTORS
   0.0=====3. PLOT SLP CONTOURS WITH WIND AND CURRENT VECTORS
   0.0=====4. PLOT GEOSTROPHIC CURRENT VECTORS
   0.0=====5. PLOT WIND+THERMAL+GEOSTROPHIC VECTORS
   0.0=====6. PLOT 5 WITH COASTAL BENDING
   1.0=====7. PLOT PROGRESSIVE VECTORS
   0.0=====PLOT OF 5 OVER 6 TO SEE DIFFERENCE IN BENDING (2.0=PLOT)
?
?END JOB
```

File 50. The final part of the set up is reading the card File 5 in the WFL run job (Program 3) to define the selected flags for the desired plots and the other run parameters. These flags and their respective settings to be edited into the WFL job are

EVGRPT -- On all vector plots chosen during this run, set EVGRPT =l to plot arrows at every grid point on the grid; or else, set EVGRPT=0 to plot every other arrow on the grid which reduces the visual confusion from overlapping arrows.

PLWIND -- Set PLWIND=l to plot the wind vector field in m/sec; or else set PLWIND=0.

IPLSLP -- Set IPLSLP=l to plot sea level pressure contours on top of the wind and ocean current vector fields if they are chosen; or else set IPLSLP=0.

PLOCW -- Set PLOCW=l to plot the ocean current vector field due to wind in cm/sec; or else set PLOCW=0.

PLOCDD -- Set PLOCDD=l to plot the permanent ocean current vector field (Delta-D) in cm/sec; or else set PLOCDD=0.

PLOCS -- Set PLOCS=l to plot the sum total surface ocean current vector field before adjustments for coastal boundary conditions: or else set PLOCS=0.

PLSOCR -- Set PLSOCR=l to plot the total surface ocean current vector field: or else set PLSOCR=0. This series of fields are the final daily result of the model,

NOPLT -- Set NOPLT=AN INTEGER to skip plotting THIS NUMBER MINUS ONE vector fields: or else set NOPLT=l to start the plotting at #l.

PROGV -- Set PROGV=l to plot progressive transport vectors for NDAYS; or else set PROGV=0. Only one PROGV plot may be run per job, thus all other plot selections must be zero.

IJEND -- Set (IJEND=INTEGER) the number of progressive vector starting points stored in File 60 (the maximum # available to plot). This may be different from the number read in from File 28 which are the number (IIEND) to be actually plotted.

WFAC -- Set WFAC=DECIMAL NUMBER, the wind tuning factor that was used in this model run (usually = 1.0).

DDFAC -- Set DDFAC=DECIMAL NUMBER, the Delta-D tuning factor that was used in this model run (usually= 1.0). Both WFAC and DDFAC are used only in headings to indicate which run this plot displays.

NDAYS -- Set NDAYS=INTEGER, the number of days you decide to be plotted. This controls the amount of run time.

FILE -- Edit all file names to correspond to those existing disk files you desire to be plotted.

Each time through the NDAYS (number of days to be plotted) loop there are two options. In the first option one or all of the five vector fields may be plotted in sequence on a Tektronics 4014 Terminal and a hard copy produced (names of flags to be set (1 or 0) in Program 3 are PLWIND, PLOCW, PLOCDD, PLOCS, or PLSOCR). The second option is to draw a single plot of one or more progressive transport vector tracks one arrow at a time for NDAYS (flag PROGV=l; all other plot flags=0). Examples of these model output plots on the ocean-wide (40X104) grid scale are shown below starting with the permanent flow field and then the daily variable flow fields for selected dates.

Geostrophic Currents

First, the permanent geostrophic current vector field over the whole grid is shown at every other grid point in Figure 4. On this large grid the major features of the surface ocean circulation in the North Pacific and Bering Sea are clearly evident. Notice the region of strong zonal flow of 20 to 40 cm/sec southwest of Japan in the Kuroshio Current. This flow gradually widens and diminishes to less than 5 cm/sec as it proceeds eastward generally along latitude lines in the Kuroshio Extension. The great Divergence occurs at the North American Continent (about 47°-49°N) with local acceleration to speeds of about 5 to 10 cm/sec in the two branches--southward in the coastal upwelling region of the California Current and northward in the coastal downwelling region of the Alaska Current. The next highest current speeds of 5 to 15 cm/sec are seen in the

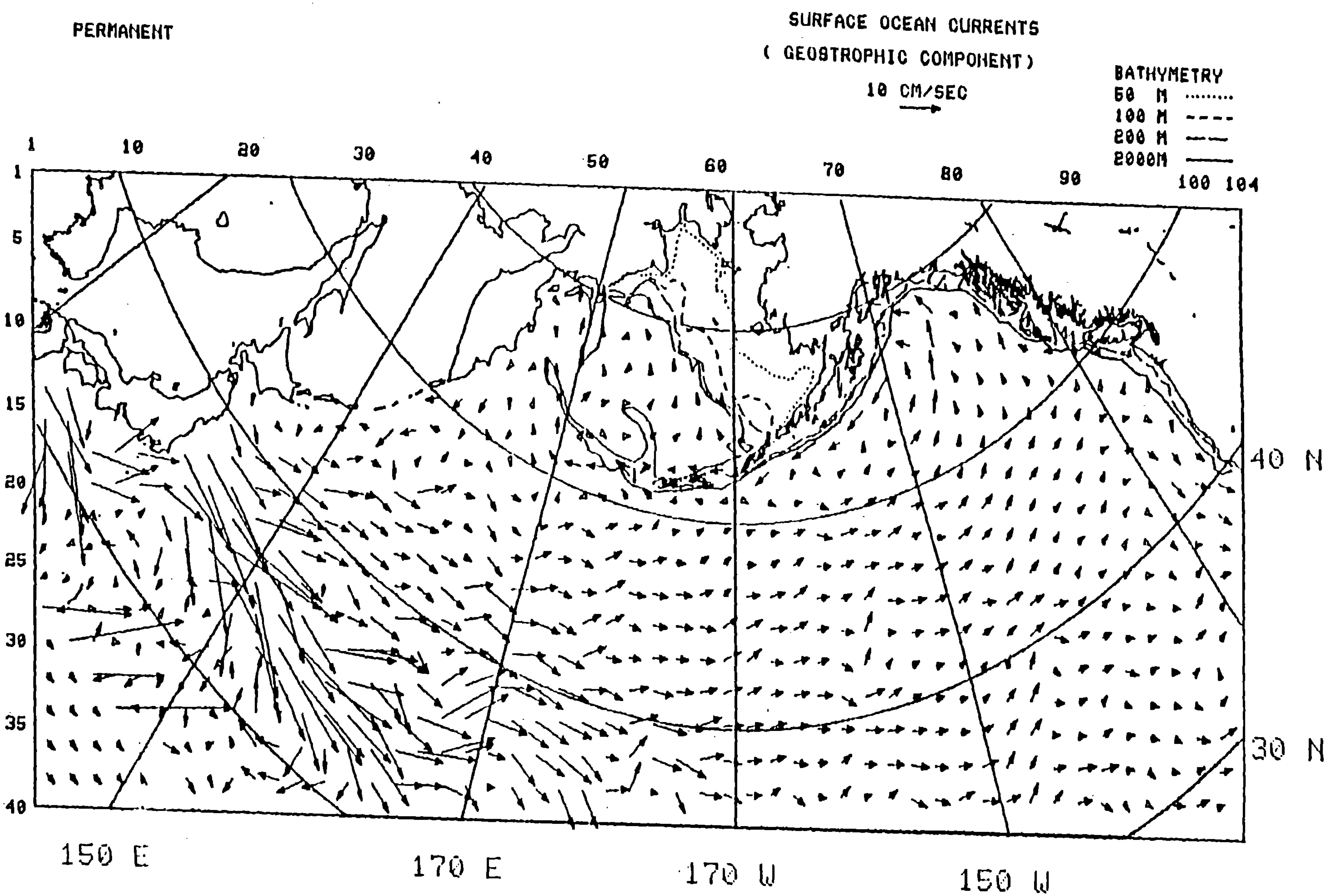

Figure 4.--Surface Geostrophic current vector field on 40x104 grid. Permanent flow component was calculated from anomaly of dynamic height data (0/2000 db).

westward return flow of the Alaskan Stream (another western boundary current) which accelerates due to the southward displacement of the westward flow along the Alaska Peninsula. The westward extension of the Alaskan Stream turns northward in the western Aleutian Islands and proceeds cyclonically (counterclockwise) around the Bering Sea Basin before joining the Oyashio Current, a third western boundary current flowing southward along the Kamchatka Peninsula and Kuril Island coasts. Thus, the appearance of a long closed loop in the mean circulation around' the Subarctic North Pacific Region is shown completed as this flow turns eastward joining the Kuroshio Extension. The accuracy of these vectors in direction is quite acceptable to show major features, but speeds are suspect particularly in regions of concentrated flow such as the Alaskan Stream in which portions of its narrow high velocity core lie unresolvable between grid points of the model.

A tuning factor constant (DDFAC) is included as a linear multiplier to adjust the magnitude of the permanent flow field. This is helpful in model calibration exercises when model outputs are compared to actual drift current measurements (see below). Particularly north of 40°N the need for augmentation of the permanent flow is expected, because at the chosen reference level for zero motion in the geostrophic calculations of 2000 decibars (db) a deeper (about 2000 m), weak flow in the same direction as the surface current may still exist locally (Ingraham and Favorite 1968). South of 40°N, however, there is also some question as to the appropriate reference level which may even be shallower than 2000 db due to the more complex water structure in the subtropical watermass. Despite the above considerations, this geostrophic current vector field is a good approximation to the basic permanent flow; but it is anticipated to improve the model by adding seasonal geostrophic flow in future versions. The other major component of flow is the wind-induced surface current which will be added to the permanent flow field to make up the final or total flow field.

Wind Currents

Surface ocean current vectors due to wind are expected to show a great diversity in daily flow patterns, generally different in both magnitude and direction from the permanent flow yet with some localized areas of strong enhancement or opposition on a short-term basis (generally 1-5 days). It takes two steps to generate a surface current field from a sea level pressure field as discussed above. Beginning with a sea level pressure field from an arbitrarily selected day in history, 17 July 1978, first a wind field is calculated (Fig. 5). Over such a wide expanse of ocean most of the common meteorologic features (cyclones, anticyclones, jets, strong shear zones, calms, etc.) will generally appear somewhere on the map. Conditions look about as expected for July with its typical pattern of summer low pressure areas over the Asian Continent which move eastward along the Aleutian Islands then northward across the Bering Sea and also a persistent Northeast Pacific High off the west coast of the United States. The key feature to notice here in

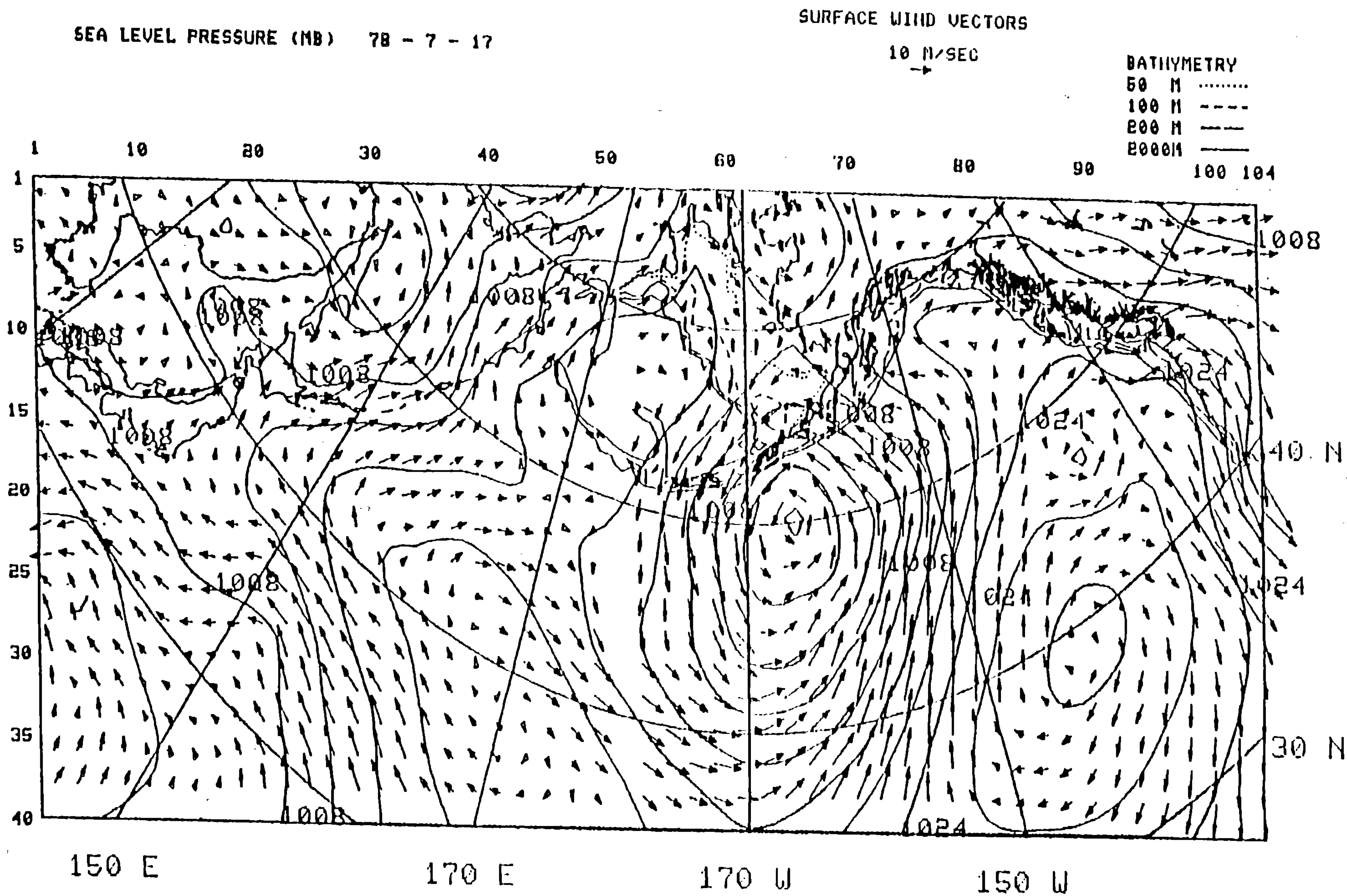

Figure 5.--Wind vector field (m/sec) calculated from arbitrarily selected sea level pressure field for 17 July 1978. Note cross-isobar flow toward lower pressure.

this one day chart is the cross-isobar wind flow. Facing with the direction of flow the angle of wind deflection to the left of the sea level pressure isobars varies in a complex manner with latitude and wind speed but is generally about 20 degrees.

By contrast in our second step the calculated surface ocean current has a deflection to the right of this wind which also varies with latitude and wind speed in a complex manner but with a slightly smaller magnitude of about 15 to 20 degrees. This gives the surface ocean current vectors (Fig. 6) an appearance of being more closely parallel to the sea level pressure isobars than the wind vectors although the speed of the ocean flow is only about 1 to 2% of the wind speed. The nature of this relationship and the bending of the ocean current vectors which impinge on the coast are more clearly seen below in zoom plots of this same data in the expanded area of the western Gulf of Alaska (see below for discussion of calculations).

Total Currents

Adding the permanent current velocity field and wind current velocity field produces the final resultant, the total current vector field for 17 July 1987 (Fig. 7). This is a one day example of the variety of complex local flow patterns that can occur in the Subarctic Region of the North Pacific Ocean. The average relatively smooth eastward flow along 40 N seen previously in the permanent component appears to be masked by very sinuous short-term variations. The effect of moving storms is seen in Figure 8 where plots of total currents for the six

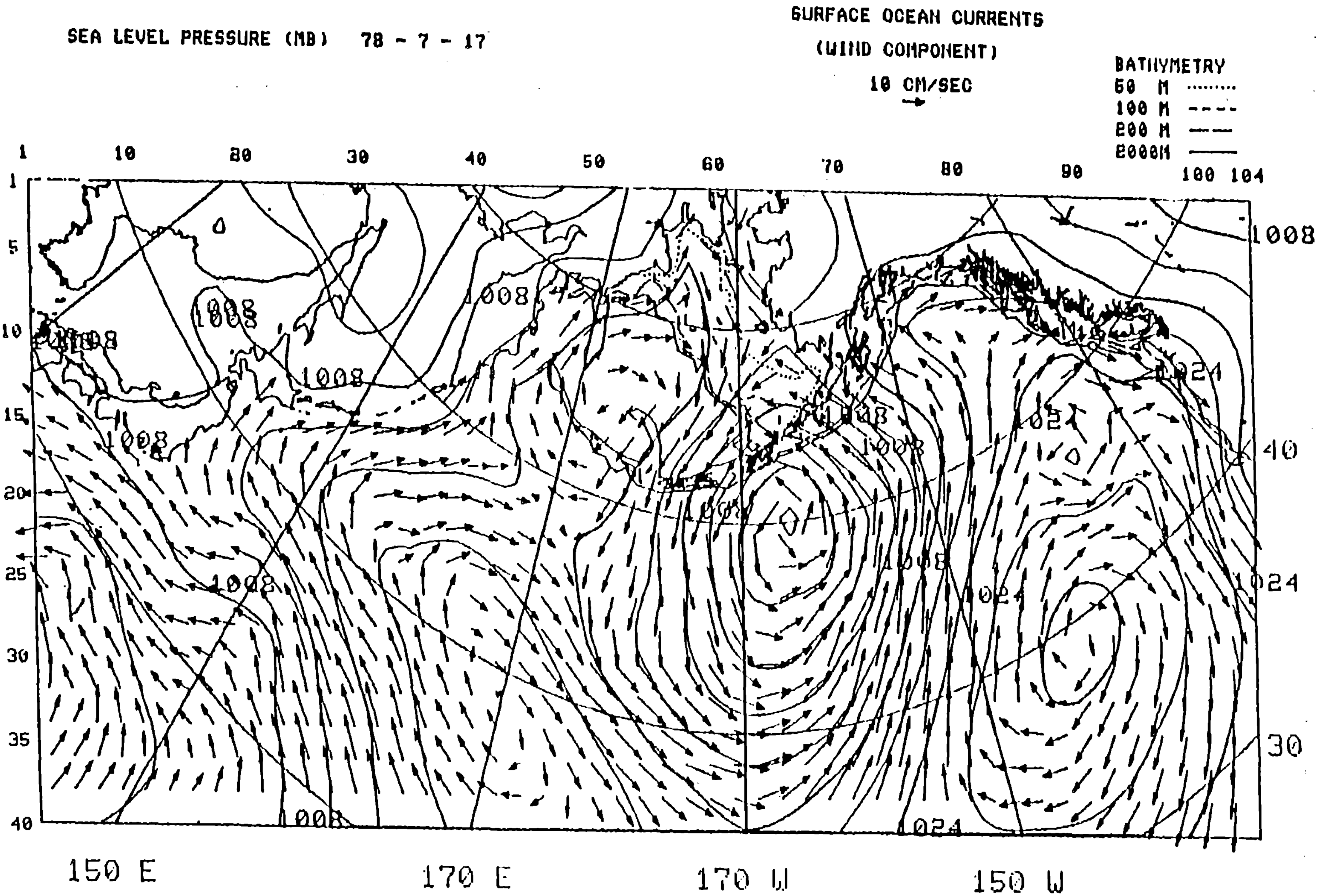

Figure 6.--Surface ocean current vector field (cm/sec) calculated from wind vector field (Fig. 5). Note: Deflection to right of wind alines currents closely with the isobars.

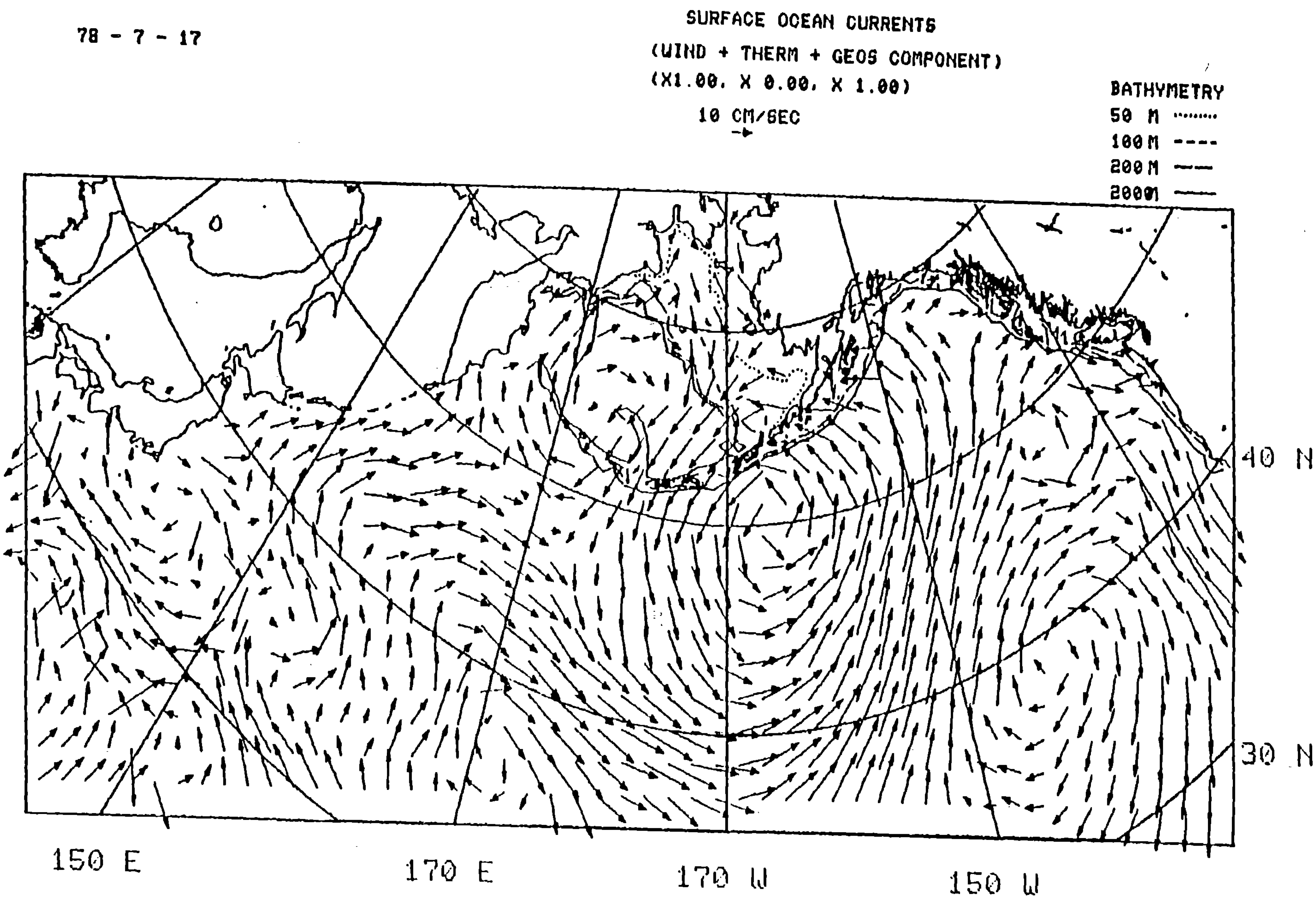

Figure 7.--Total surface ocean current vector field (sum of geostrophic plus wind current) for 17 July 1978. Final results of model computations.

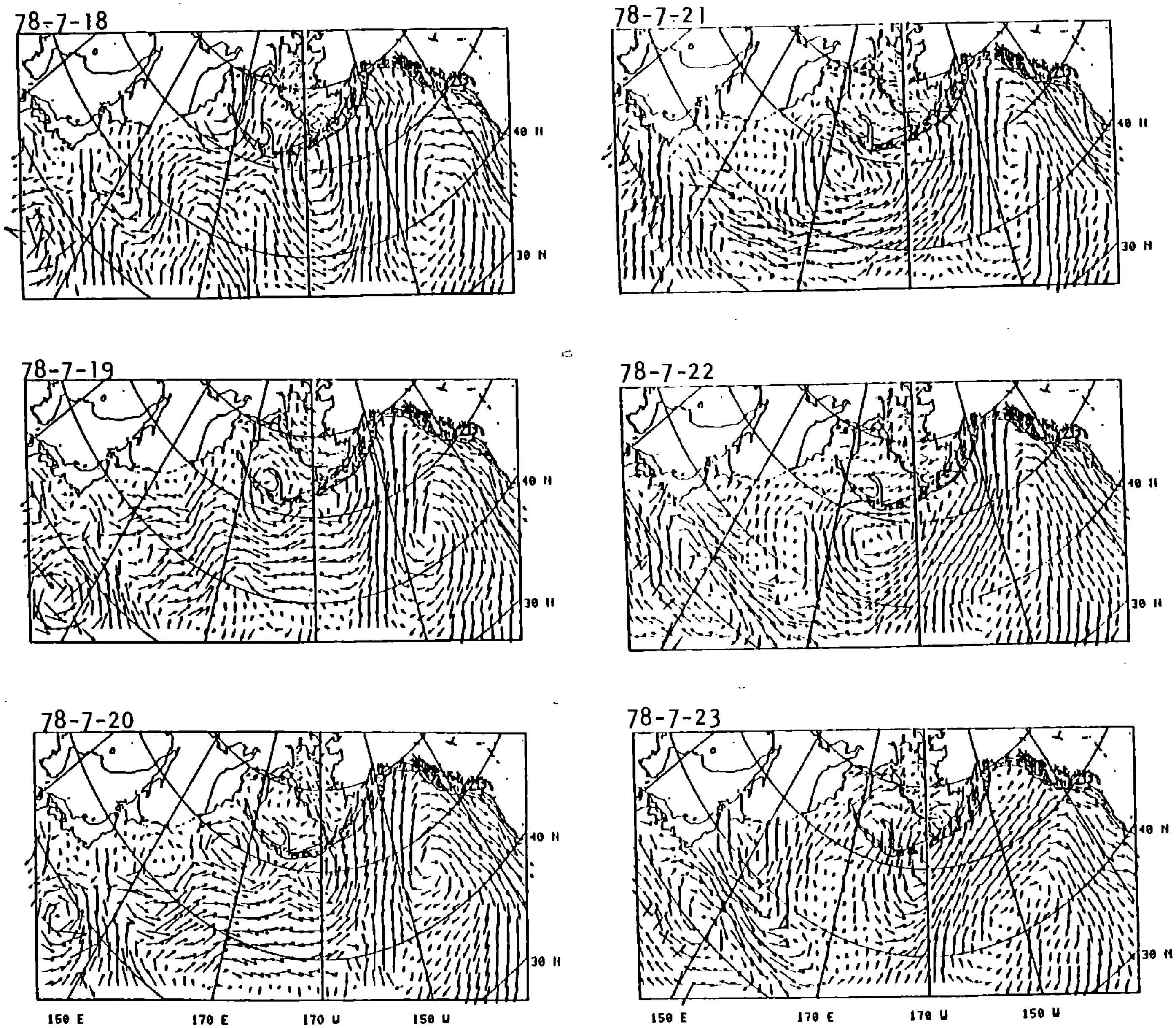

Figure 8.--Total surface currents, July 18 to 23, 1978. Model results for 6 subsequent days following 17 July 1978 (Fig. 7).

subsequent days show the nature of variability on a weekly basis. Some complete reversals of flow lead to changing conditions of convergence and divergence near shore along the coastline.

Considering the shear number of possible model output fields (wind, wind current, and total current) the task of analysis seems a bit overwhelming. Our present data base of 41 years of daily data allows about 15,000 possible charts for each field. Let us recall that the reason for making the model was to trace the effects of all this short term variability over time. Therefore, the transformation from current vector fields to progressive transport (distance moved in one day) vector strings from any point in the model gives us a good simulation of actual ocean drift conditions for as many days as we choose and the ability to tune model results to actual drifter tracks.

Progressive Vector Tracks

Progressive transport vector tracks for July through December 1978 (Fig. 9(a)) starting from arbitrarily selected starting points give us a Lagrangian perspective of flow and an estimate of the daily trajectory of a parcel of water near the surface over a 5-month period under the influence of the permanent long-term mean geostrophic flow and daily winds. Of particular interest is the band of strong zonal flow between about 35°N and 45°N which shows a rather uniform eastward drift of surface water across most of the North Pacific Ocean with short-term (daily to a few weeks) fluctuations toward the north or south or reversals superimposed on the main eastward flow.

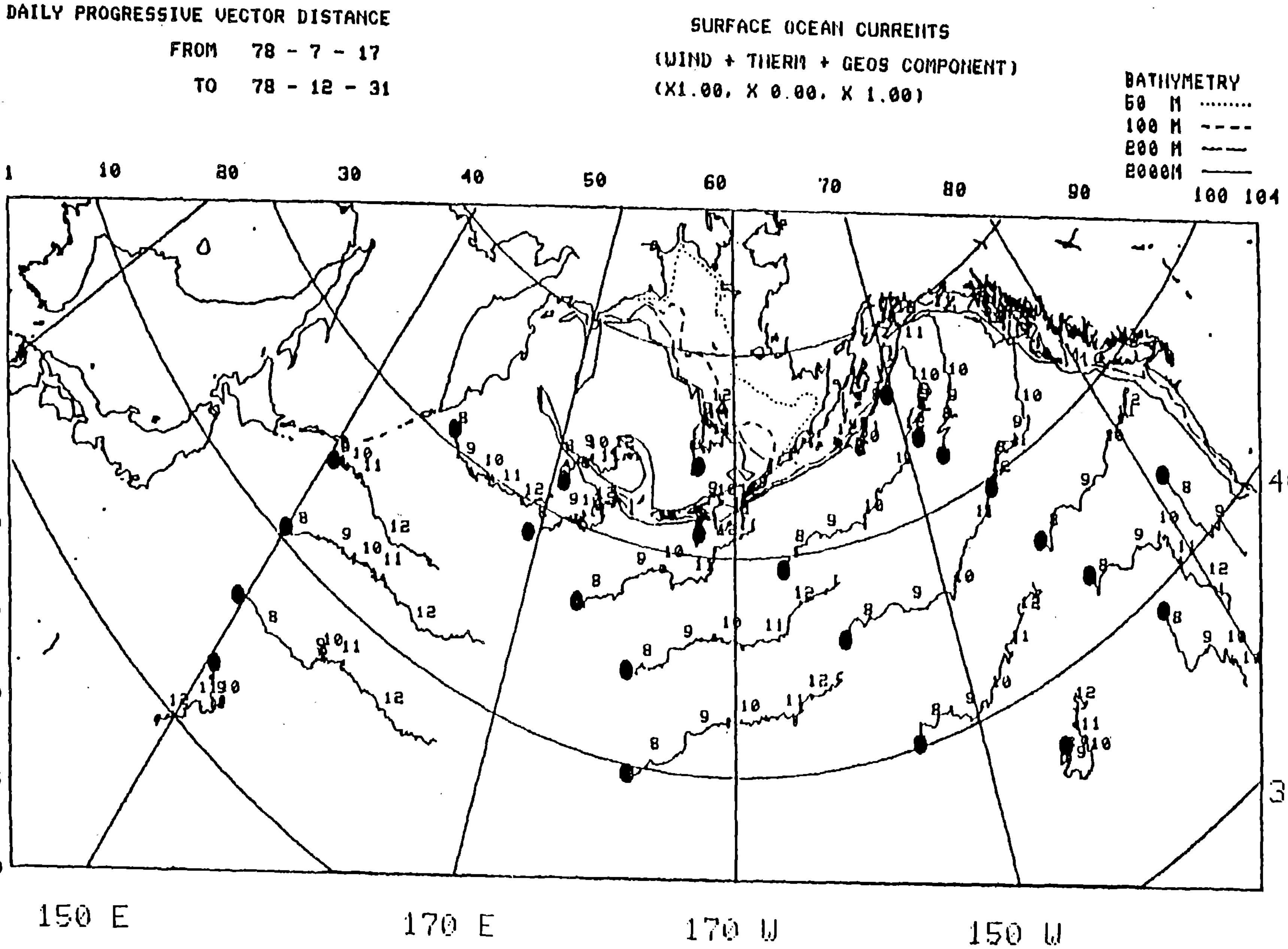

Figure 9(a). --Progressive daily transport vectors-Total surface ocean transport (17 July to 31 December 1978). Numbers along tracks correspond to the first day of the indicated month.

Many of these fluctuations seem to cancel each other in the north-south direction. Although the net surface drift in Figure 9(a) appears to be relatively uniform all across the ocean (speeds about 500-600 nautical miles per 165 days or about 6-8 cm/sec), the individual contribution from the permanent flow (Fig. 9(b)) and the wind current (Fig. 9(c)) changes considerably within this region. Off Japan at least during this summer the eastward flow is nearly all geostrophic with little wind contribution, but going eastward the geostrophic component gradually diminishes while the wind component increases. In the western Gulf of Alaska the geostrophic southwest flow of the Alaskan Stream (Fig. 9(b)) is in direct opposition to the northeastward flowing wind current (Fig. 9(c)). Some meandering and reversing appear in the overall cyclonic flow in the western Gulf of Alaska, while relatively stagnant conditions are present in the center of the oblong gyre between the westward and eastward flows. The time period for this sample model run was selected to coincide with the field data from satellite tracked drifting buoys released there in July and tracked through December 1978 (Reed 1980). These progressive transport vectors will be examined in greater detail in the next section on the zoom plot program.

Zoom Plot of Gulf of Alaska on 25X35 Grid

The purpose for drawing zoom plots, as the name indicates, is to zoom in on or expand a small area of interest and more clearly view model output. Zoom plots also assist in model tuning

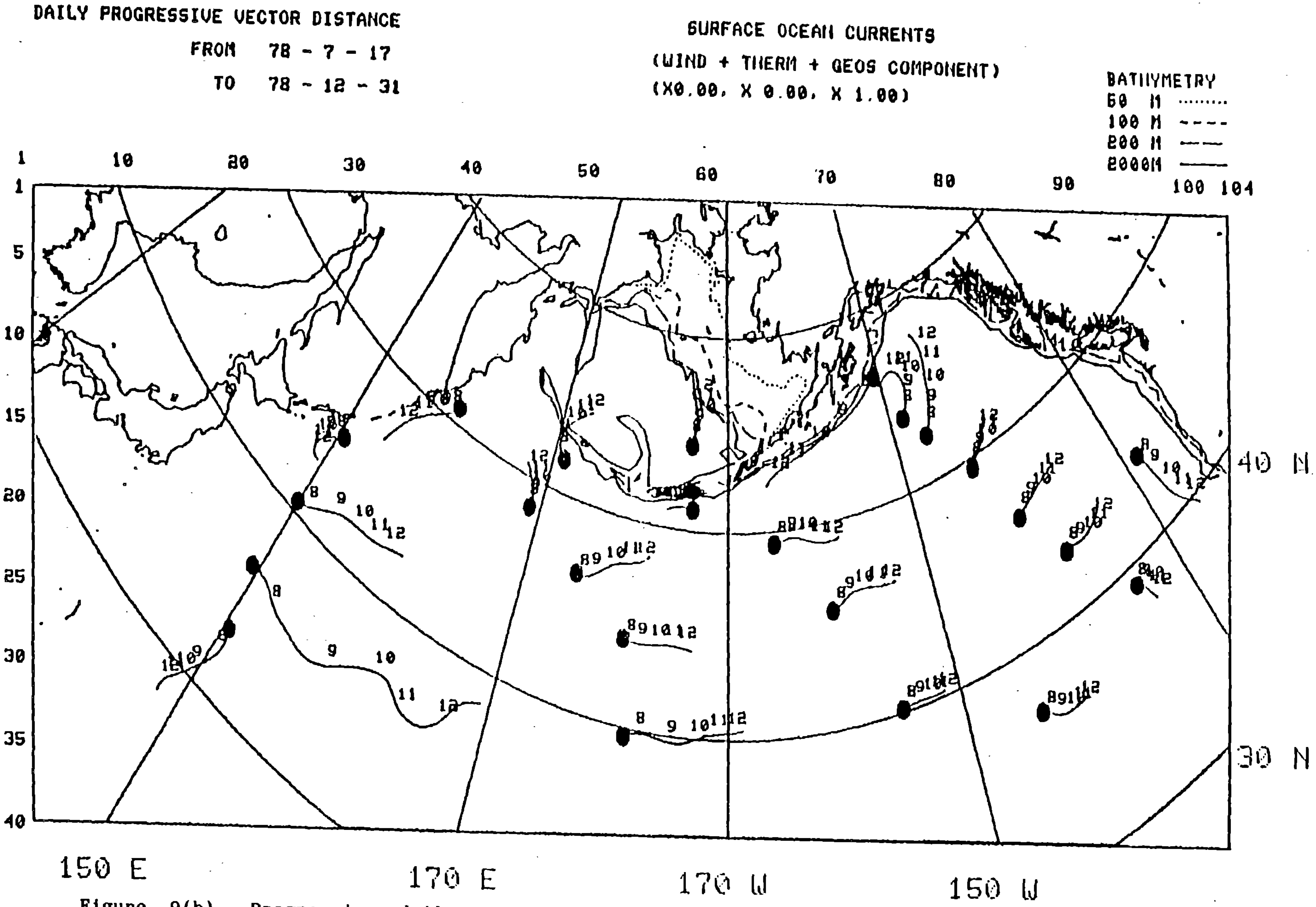

Figure 9(b).--Progressive daily transport vectors-Permanent geostrophic flow only (WFAC=0.0)(17 July to 31 December 1978).

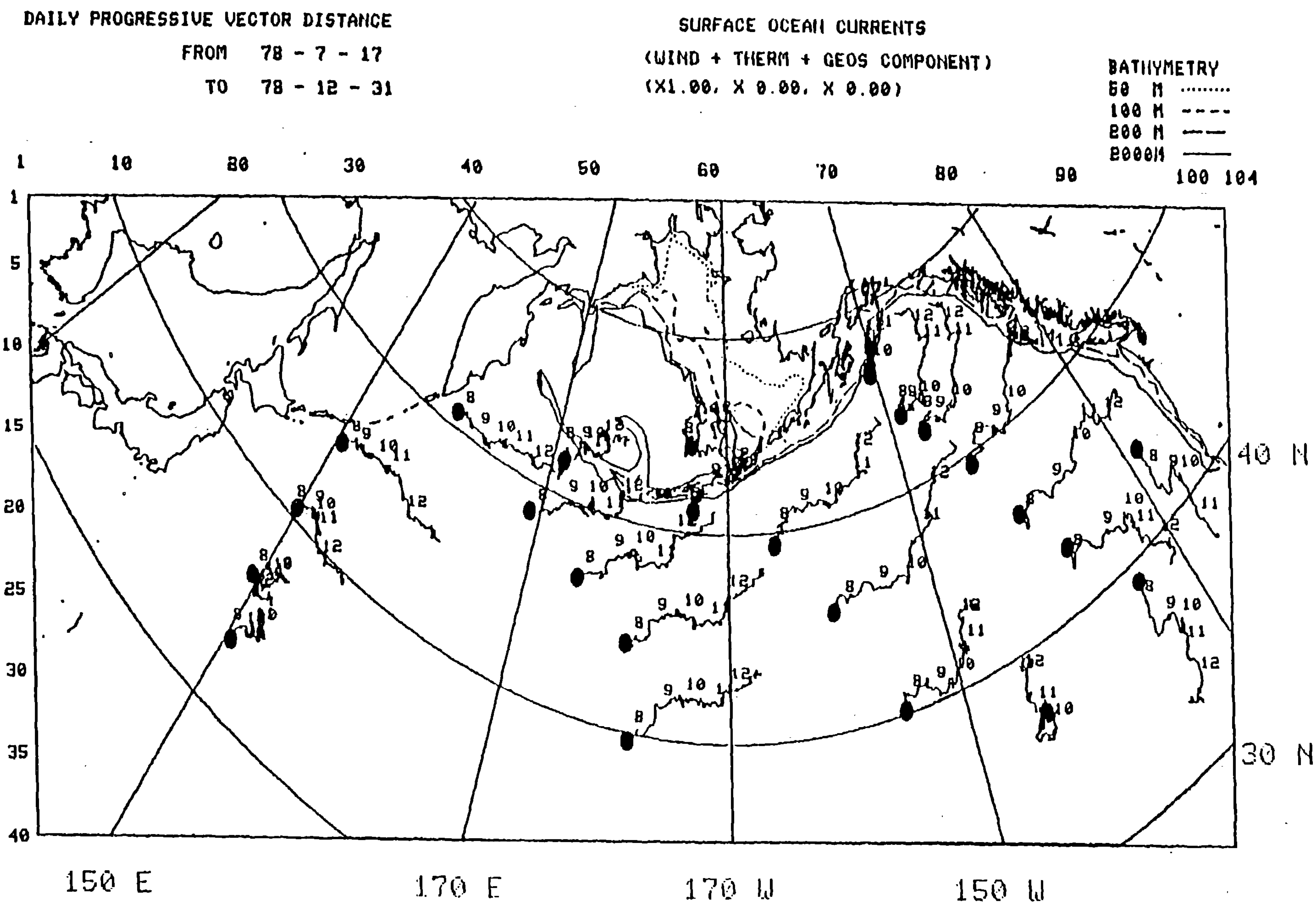

Figure 9(c). --Progressive daily vectors-Surface current due to wind only (DDFAC=0.0) (17 July to 31 December 1978).

and verification by displaying fine-scale details for comparisons between directly measured drift tracks and the progressive transport vector tracks computed by the model. Our first level of zoom is discussed here, an expansion of a 25X35 portion of the main grid (Fig. 10) in the Gulf of Alaska. The indices of the large grid, I=5-29 and J=56-90, correspond to zoom plot indices, I=1-25 and J=1-35, respectively.

Because the zoom plot program is essentially the same program as the ocean-wide plot program (already listed in Appendix 2), with some index changes and scale adjustments, the actual Fortran code is not listed here. The WFL code of the compiler-binder job (Program 4) must still be run before final execution of the model if any changes are made in the Fortran code. To run the zoom plot program, the same file name and parameter changes have to be edited into the WFL code of the run job (Program 5) as described above for the ocean-wide version.

A choice was made early in the model design to give a high priority to simulations of continuous drift tracks which could potentially cover ocean-wide space scales over annual to interannual time scales. One obvious consequence of this choice due to array size limitations was a sacrifice in resolution which could have been achieved by putting together several finer mesh submodels to make up the large grid. Avoided were the immense boundary problems of matching complex drift patterns back and forth across boundaries of several adjacent submodel size grids: and thus, we chose to rely on the graphics plot program to zoom in on results in small portions of the large grid. Al though

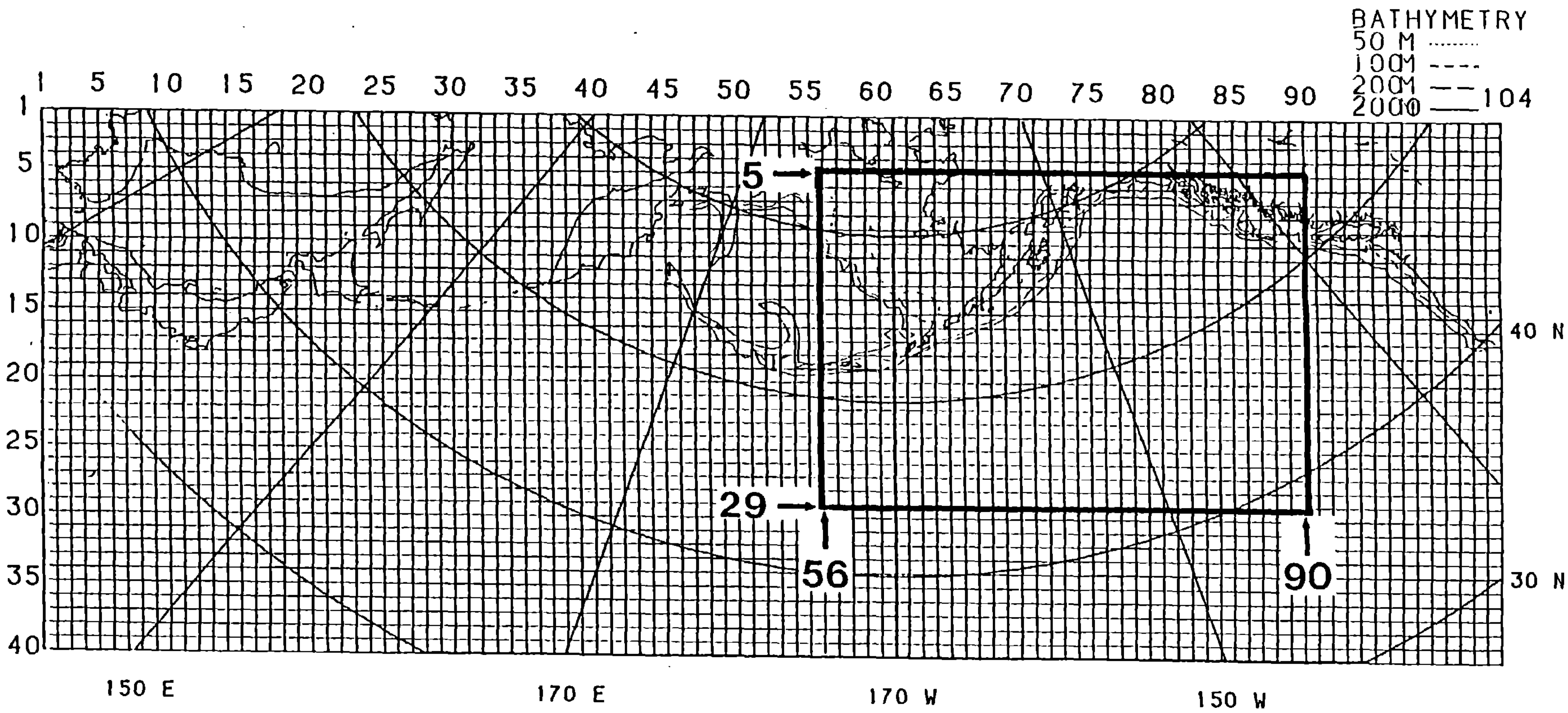

Figure 10.--Large area 40x104 grid and subset 25x35 grid for zoom graphics plots.

Program 4. --Listing of the Binder Job which binds and compiles the graphic outputs of the zoom model (File Title="ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW/BINDER").

```
?BEGIN JOB MIYAHARA/BINDER/ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW;
 QUEUE=20;
COMPILE OBJECT/ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW FORTRAN LIBRARY;
COMPILER FILE TAPE(KIND=DISK,TITLE=ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW
?FORTRAN DATA
$SET MERGE
$RESET LIST
?
BIND ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW/BOUND BINDER LIBRARY;
BINDER FILE HOST(TITLE=OBJECT/ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW);
?BINDER EBCDIC
$RESET LIST
BIND=FROM GRIDS40104/=,(REFM0070)REFMOFF/= ON TYVO,
           (REFM0070)PLOTCOMP87/= ON TYVO;
?END JOB
```

Program 5. --Listing of Word Flow Language (WFL) Program which runs the graphic outputs of the zoom model (File Title="ROB/M/PLOT/AUTO/ZOOM/WGULF ARROW/WFL").

```
?BEGIN JOB INGRAHAM/M/PLOT/PGRV/DDTEST/BOUND;
 QUEUE=20;
 RUN ROB/M/PLOT/AUTO/ZOOM/WGULF/ARROW/BOUND;
FILE FILE10=DIR96202;
FILE FILE28=P/3;
FILE FILE50=CURRENTS/40X104/DYNCURR/S2X2;
FILE FILE51=CURRENTS/40X104/DSLP/DDTEST/FM7TOM12/D17START;
FILE FILE52=CURRENTS/40X104/WIND/DDTEST/FM7TOM12/D17START;
FILE FILE53=CURRENTS/40X104/SFCOCN/DDTEST/FM7TOM12/D17START;
FILE FILE54=CURRENTS/40X104/W10DD10/DDTEST/FM7TOM12/D17START/CSTROT
FILE FILE55=CURRENTS/40X104/W10DD10/DDTEST/FM7TOM12/D17START;
FILE FILE62=CURRENTS/40X104/PROGRESVECT/76/FM7TOM12/D17START/W075DD
FILE FILE61=CURRENTS/40X104/PROGRESVECT/77/FM7TOM12/D17START/W075DD
FILE FILE60=CURRENTS/40X104/PROGRESVECT/TEST/FM7TOM12/D17START/W10D
FILE FILE63=CURRENTS/40X104/PROGRESVECT/79/FM7TOM12/D17START/W075DD
FILE FILE64=CURRENTS/40X104/PROGRESVECT/78/FFM7TOM12/D17START/W30DD
FILE FILE65=CURRENTS/40X104/PROGRESVECT/78/FFM7TOM12/D17START/W20DD
FILE FILE80=CURRENTS/40X104/PROGRESVECT/80/FM7TOM12/D17START/W075DD
FILE FILE81=CURRENTS/40X104/PROGRESVECT/81/FM7TOM12/D17START/W075DD
FILE FILE82=CURRENTS/40X104/PROGRESVECT/82/FM7TOM12/D17START/W075DD
FILE FILE83=CURRENTS/40X104/PROGRESVECT/83/FM7TOM12/D17START/W075DD
?DATA
   1.0=====EVERY GRID POINT PLOT SLP WITH CURRENTS
     5=====NUMBER OF DAYS TO PLOT
     0=====PLOT SEA LEVEL PRESSURE CONTOURS ON PLOTS
   0.0=====1. PLOT WIND VECTORS (1.0=PLOT)
   0.0=====2. PLOT WIND CURRENT VECTORS
   0.0=====3. PLOT THERMAL CURRENT VECTORS
   0.0=====4. PLOT GEOSTROPHIC CURRENT VECTORS
   0.0=====5. PLOT WIND+THERMAL+GEOSTROPHIC VECTORS
   1.0=====6. PLOT 5 WITH COASTAL BENDING
   0.0=====7. PLOT PROGRESSIVE VECTORS
   0.0=====PLOT OF 5 OVER 6 TO SEE DIFFERENCE IN BENDING (2.0=PLOT)
     1=====SET THE NUMBER TO PLOT OR NOPLT=1 TO START AT #1
     0=====THE NUMBER OF PROGRESSIVE VECTOR STARTING POINTS
   1.0=====THE WIND TUNING FACTOR
   1.0=====THE DELTA-D TUNING FACTOR
?
?END JOB
```

other zooming is planned for future versions, the present scale is adequate to describe model outputs at this stage of development. Model results which were discussed above on the broad ocean-wide scale from plots of output values at every other grid point are now shown at every grid point in an enlarged portion of the Gulf of Alaska.

Geostrophic Currents

Geostrophic current vectors at zoom scale (Fig. 11(a)) show a relatively smooth, cyclonic flow around the Gulf of Alaska Gyre. But rather than being circular as the term "gyre" implies, this gyre has a closed asymmetrical form. This oblong shape is a result of the dynamic forces which cause the formation of the Alaskan Stream western boundary current where narrowing, southward flow, and friction along the land boundary of the Alaska Peninsula require intensification of flow and the high velocity axis of the Stream to be centered on the average just seaward of the edge of the continental shelf (water depth about 200 m). Therefore, the Alaskan Stream is essentially a permanent feature. The gyre is manifested by southward flow in the Alaskan Stream, by offshore flow where the stream weakens to the west, by eastward flow of the Subarctic Current to the south, and by northward then westward flow at the head of the gulf. A long, narrow, relatively stagnant zone appears in the central part of the gyre.

The arrow lengths in Figure 11(a) indicate that velocities calculated from data with this large model grid spacing are

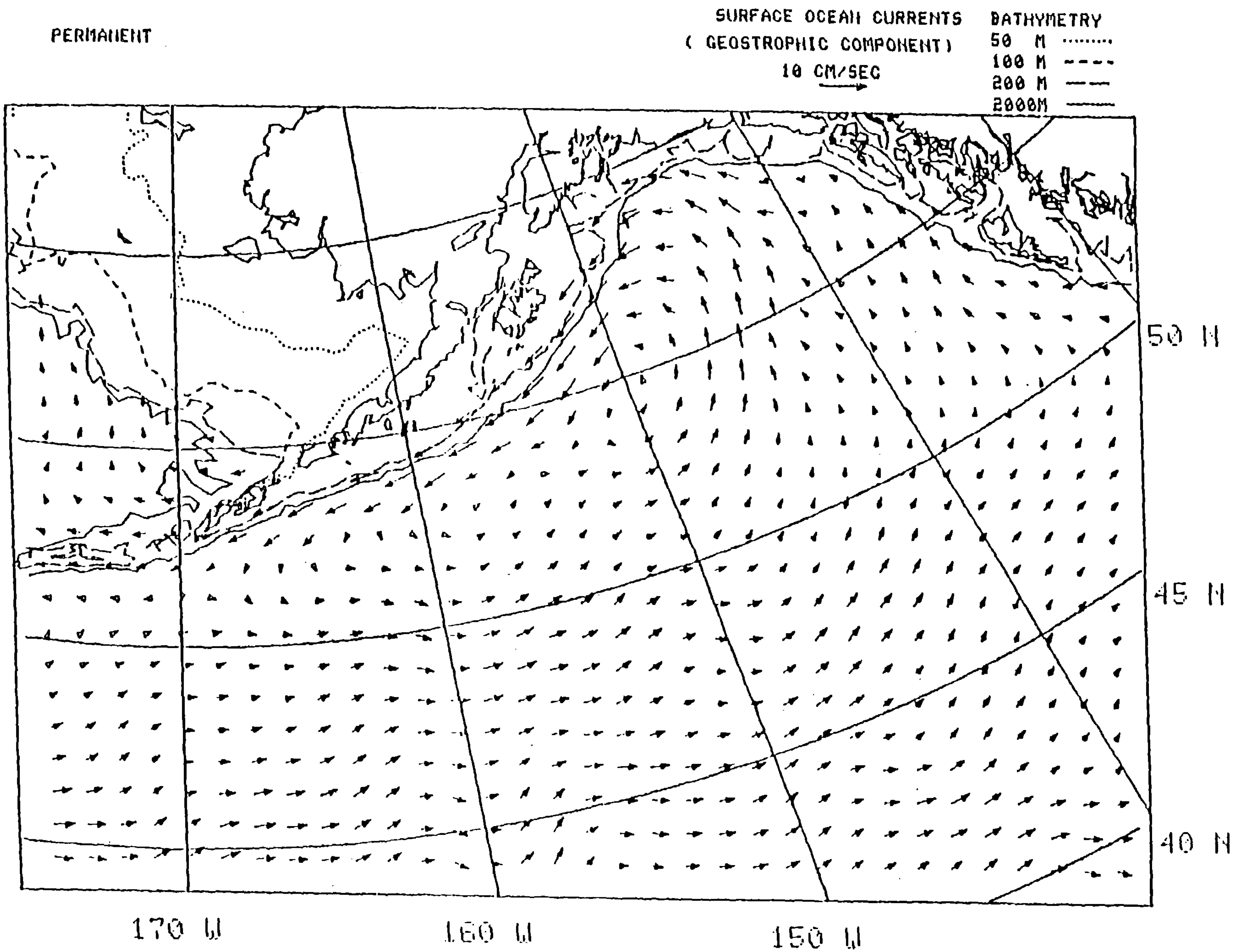

Figure 11(a). --Surface geostrophic current vector field on 25x35 grid (0/2000 db).
Permanent flow field calculated without modifiers in the Alaskan Stream.

slower than expected in the Alaskan Stream, generally less than 10 cm/sec. Closely spaced (5-15 km) conductivity, temperature and depth (CTD) station data have shown that the high velocity axis of the stream is narrow, about 15 to 40 km, much narrower than the 90 km grid of the model, and computed geostrophic speeds, may be as much as 60 cm/sec (Ingraham and Favorite 1968). To increase the reality of this simulation, velocity values calculated at certain grid points near the axis of the Alaskan Stream have been modified to reflect these higher known speeds. Modifiers used at specified grid points are shown in Table 4. Values of U and V components at these specified grid points are multiplied by 1.0 plus the corresponding modifier value. Figure 11(b) shows the geostrophic current vector field after modification for comparison with the initially calculated vectors in Figure 11(a). These modifications have greatly enhanced the simulated movement of drifters within the Alaskan Stream and will provide another mechanism for calibration and fine tuning of the model to actually measured drifter tracks from reported ocean experiments.

Wind Currents

Zoom plots of wind with sea level pressure contours superimposed (Fig. 12), currents due to wind with sea level pressure contours superimposed (Fig. 13), and total currents (Fig. 14a) for 17 July 1978, are shown to provide expanded details of the Gulf of Alaska portion of the ocean-wide plots which were discussed earlier in Figures 5, 6, and 7,

Table 4. --Modifier values for the calculation of the permanent geostrophic current field near the axis of the Alaskan Stream at specified (I,J) grid points. Model uses (l.0+modifier) times the value at the grid point to reflect higher known speeds.

I	J	Modifier
10	74	0.60
11	74	0.20
11	73	0.75
12	73	0.35
12	74	-0.15
13	73	-0.20
13	72	0.45
14	72	-0.20
14	71	0.45
15	70	0.30
15	69	0.50
16	69	-0.10
16	68	0.45
16	67	1.00
17	66	-0.10
17	65	0.95
18	64	-0.10
18	63	0.80
19	62	0.15
19	61	0.30
19	60	0.45
19	59	0.35

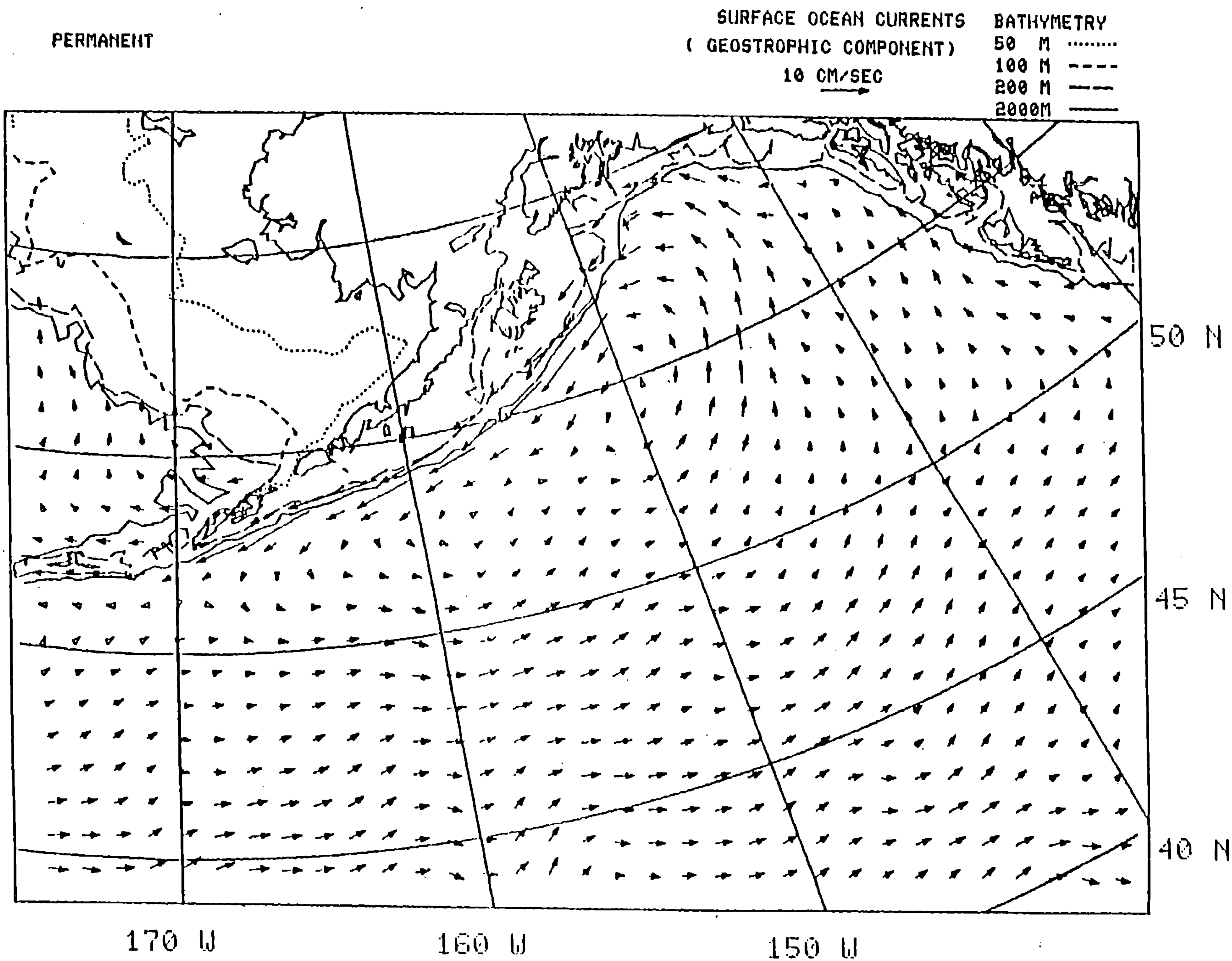

Figure 11(b). --Surface geostrophic current vector field on 25x35 grid (0/2000 db). Permanent flow field with double modifiers in the Alaskan Stream.

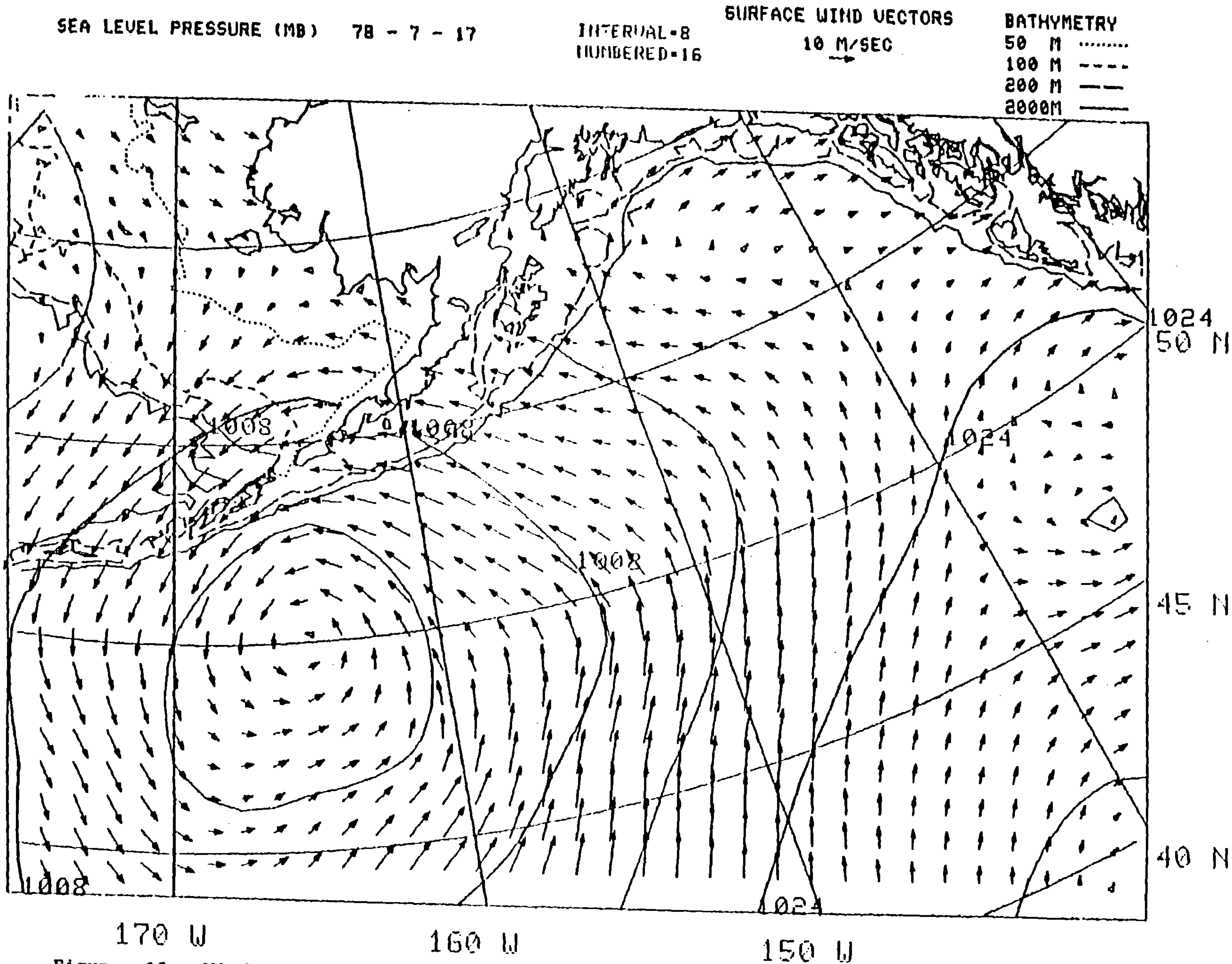

Figure 12.--Wind vector field (m/sec) calculated from sea level pressure fields for 17 July 1978. Zoom plot of Figure 5 showing cross-isobar flow into the low pressure centers.

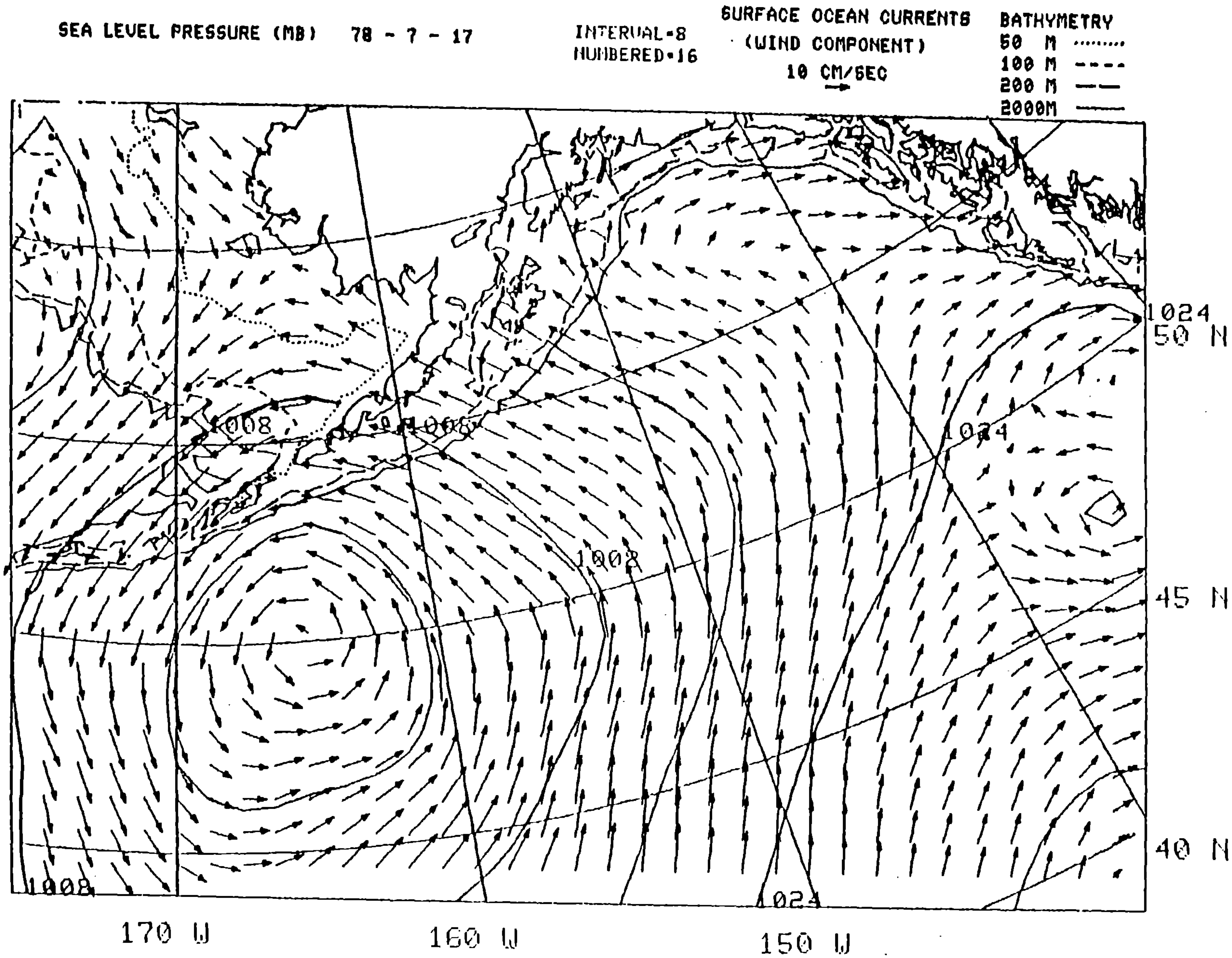

Figure 13.--Surface ocean current vector field (cm/sec) calculated from wind vector field in Figure 5 for 17 July 1978. Zoom plot of Figure 6 showing ocean currents nearly paralleling sea level pressure isobars.

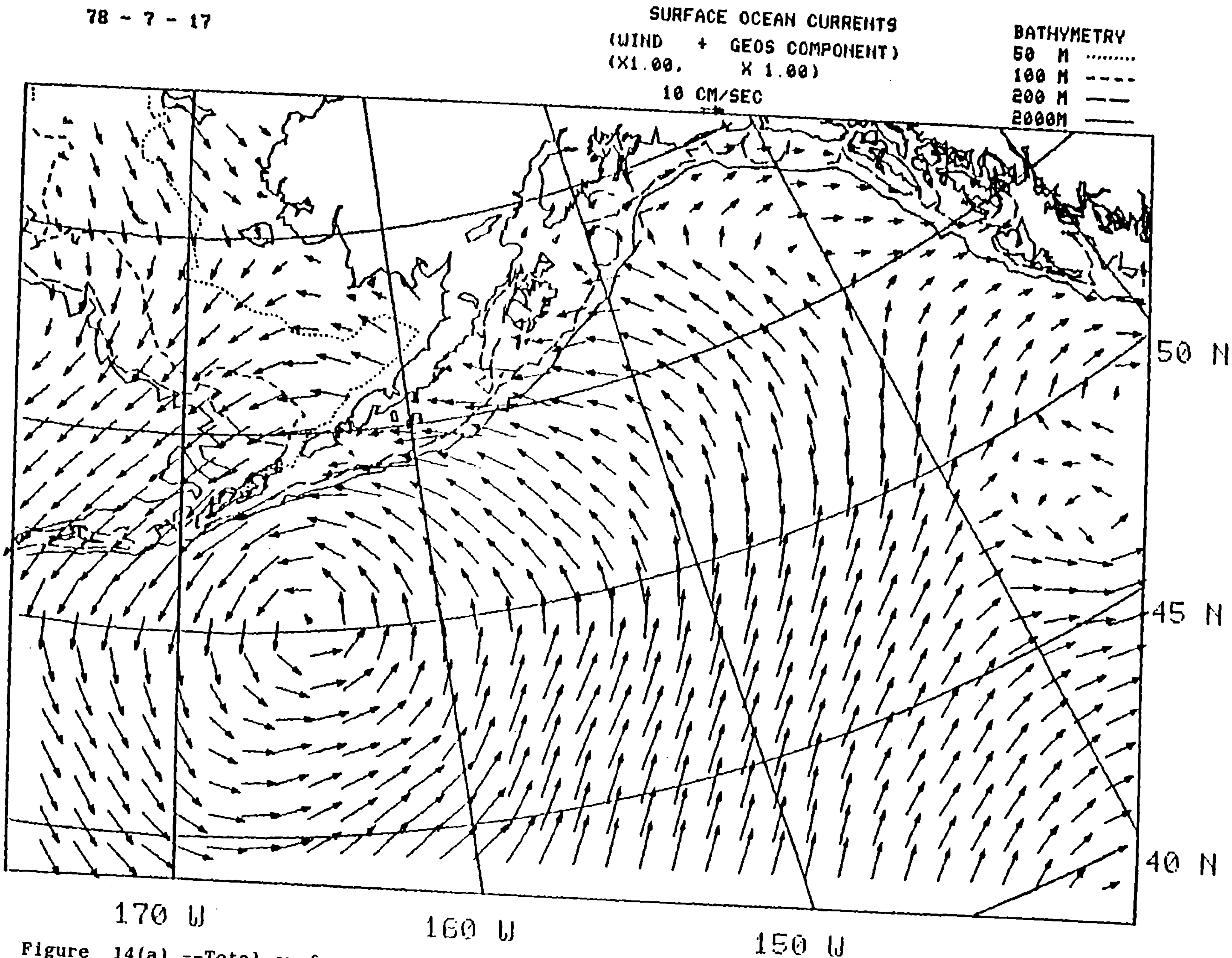

Figure 14(a).--Total surface ocean current vector field for 17 July 1978. Zoom plot of Figure 7.

respectively. Major output features of the model are more clearly seen here. Wind vectors (Fig. 12) show strong cross-isobar flow spiraling into the low pressure center at 50°N-165°W and out of the high pressure center at 40°N-142°W. Ocean surface current vectors (Fig. 13) which are computed at an angle to the right of the wind direction at each point in the field are now nearly parallel to the sea level pressure isobars. Note that some cross-isobar flow still remains in some areas such as near 47°N-153°W where winds are stronger than about 15 m/sec, making the computed angle of deflection of the currents to the right of the wind smaller. As a general character trait, currents due to wind are rarely less than 5 cm/sec or greater than 20 cm/sec, but most speeds do fall between 10 and 20 cm/sec over wide areas. Thus, wind current speeds definitely dominate over the geostrophic speeds in the overall features of the total surface currents (Fig. 14(a)).

Total Currents

A final touch is added to this total surface current vector field at each grid point which is within two grid lengths of land to account for simulated boundary effects on onshore flow (see above discussion). Here, Figure 14(b) shows a one-example comparison of the two current vectors: 1) the initial vector before bending and reduction and 2) the final resultant current vector. Along the west coast of Canada these near shore vectors are deflected to the right in this example because their angle of incidence was to the right of the line normal to shore. Along the

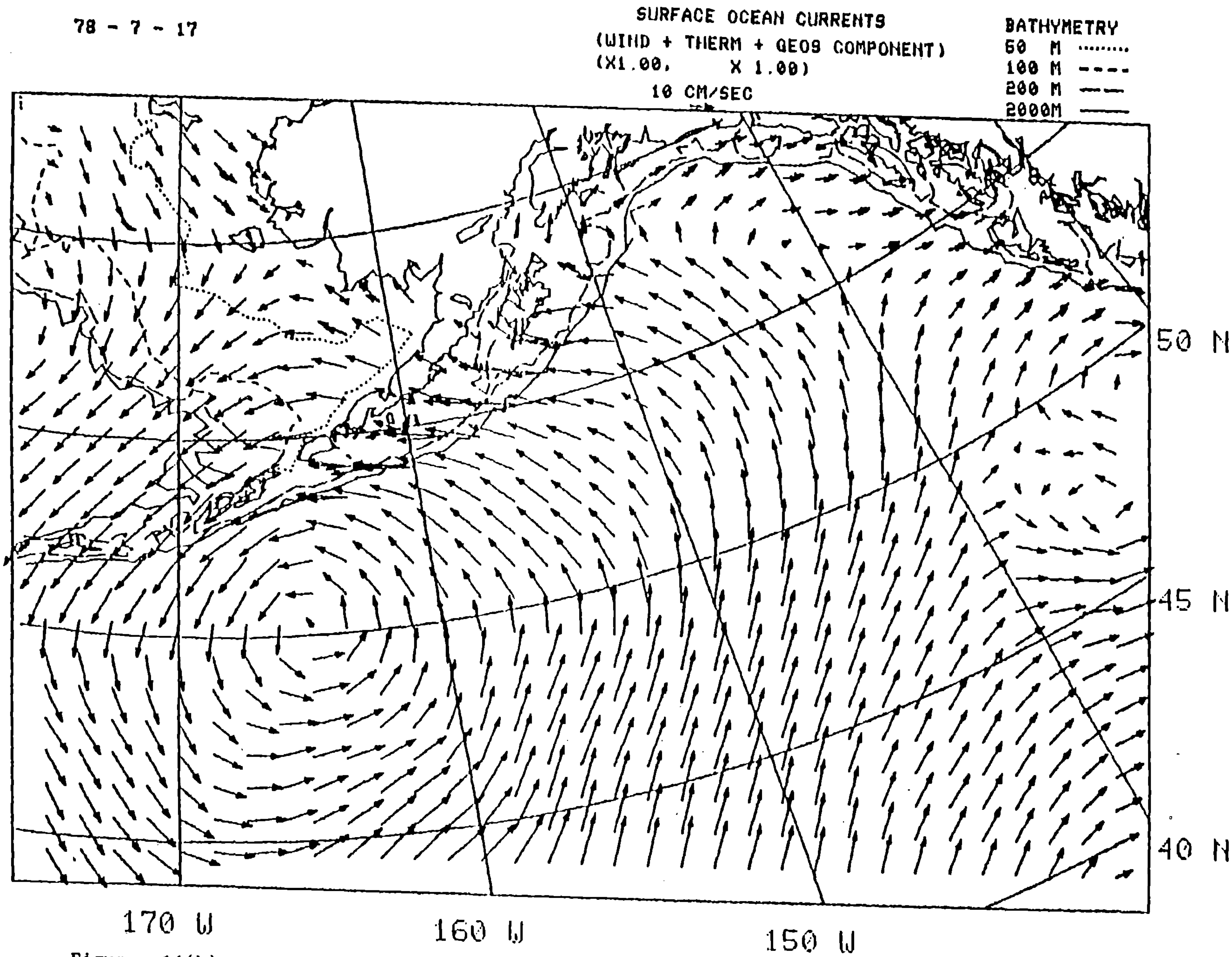

Figure 14(b).--Total surface ocean current vector field plus a second arrow near shore showing conditions before and after the costal bending routine was entered for 17 July 1978.

Alaska Peninsula the pattern shows the opposite deflection to the left. Between these two areas along the continental shelf near 150°W the flow tends to diverge at least for this one day in history. The lengths of the final total current vectors have also been diminished inversely proportional to their distance from shore. Vectors are forced mathematically to zero at shore, and no bending effects are computed for winds that are directed offshore. This completes the computation of the total current vector field for the first day of the model run.

As an example of the daily variability the next 6 days of total currents are shown in Figures 15(a-f) which are zoom plots of the six parts of Figure 8. In these six fields the overall flow pattern looks reasonably persistent with a strong 15 to 25 cm/sec current flowing northeastward towards the head of the Gulf of Alaska where this onshore flow diverges to the west and southeast. This coastal divergence area appears to shift westward from about 130°W to about 150°W and then shift back to 130°W over this 6-day period without substantial changes in the locations of the cyclonic gyre south of the Aleutian Islands at about 170°W and the anticyclonic gyre off the west coast of the United States at about 45°N-45°W. The wind current component of flow appears to continue to dominate portions of even the strongest geostrophic areas like the Alaskan Stream, either enhancing or opposing the geostrophic flow at least on a short-term basis (l-3 days). These examples show that total currents of l0-20 cm/sec still appear to be consistently active over large areas of these fields even during summer. Velocities probably would have been even

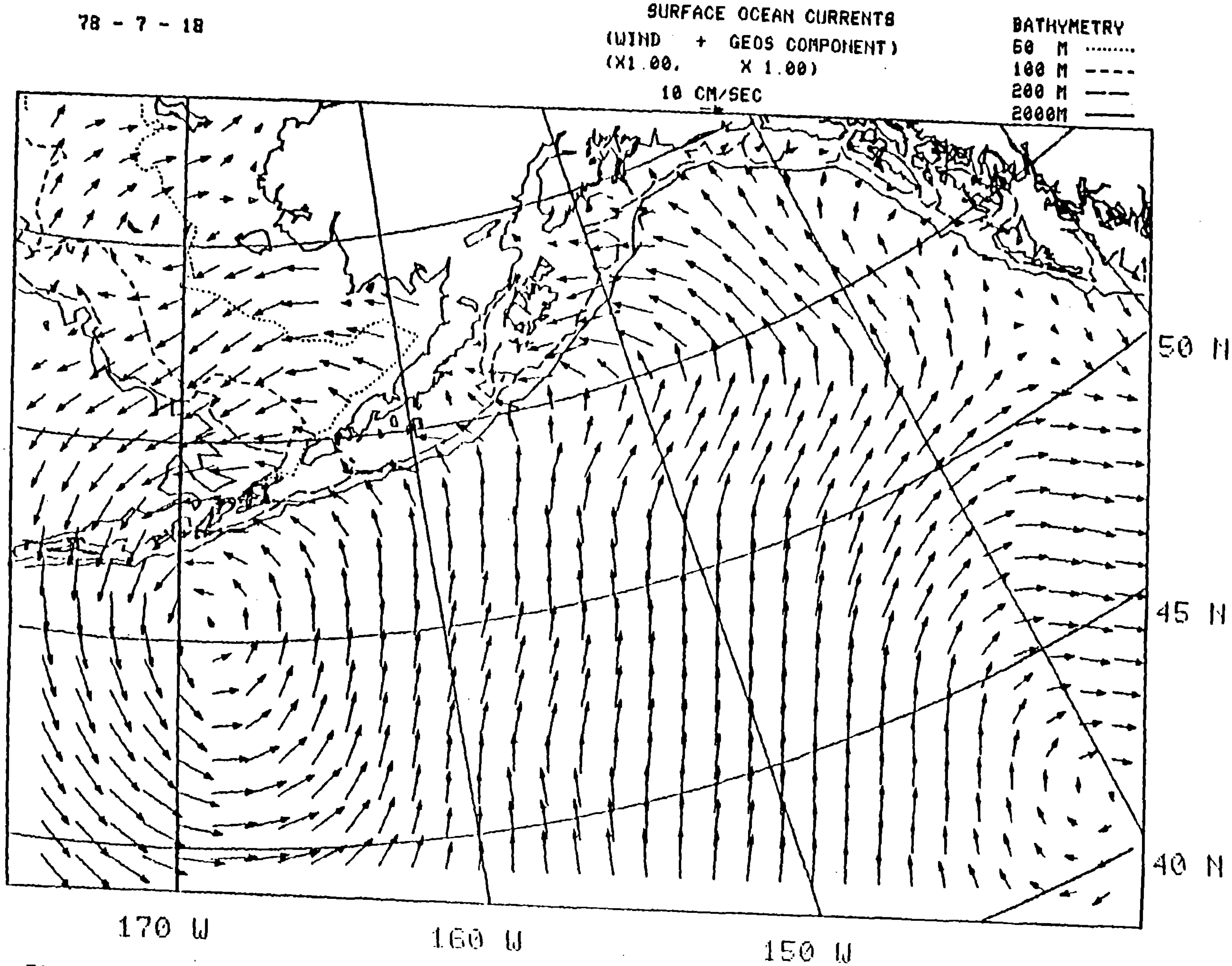

Figure 15(a).--Total surface ocean current vector field for 18 July 1978. Zoom plot of a portion of Figure 8.

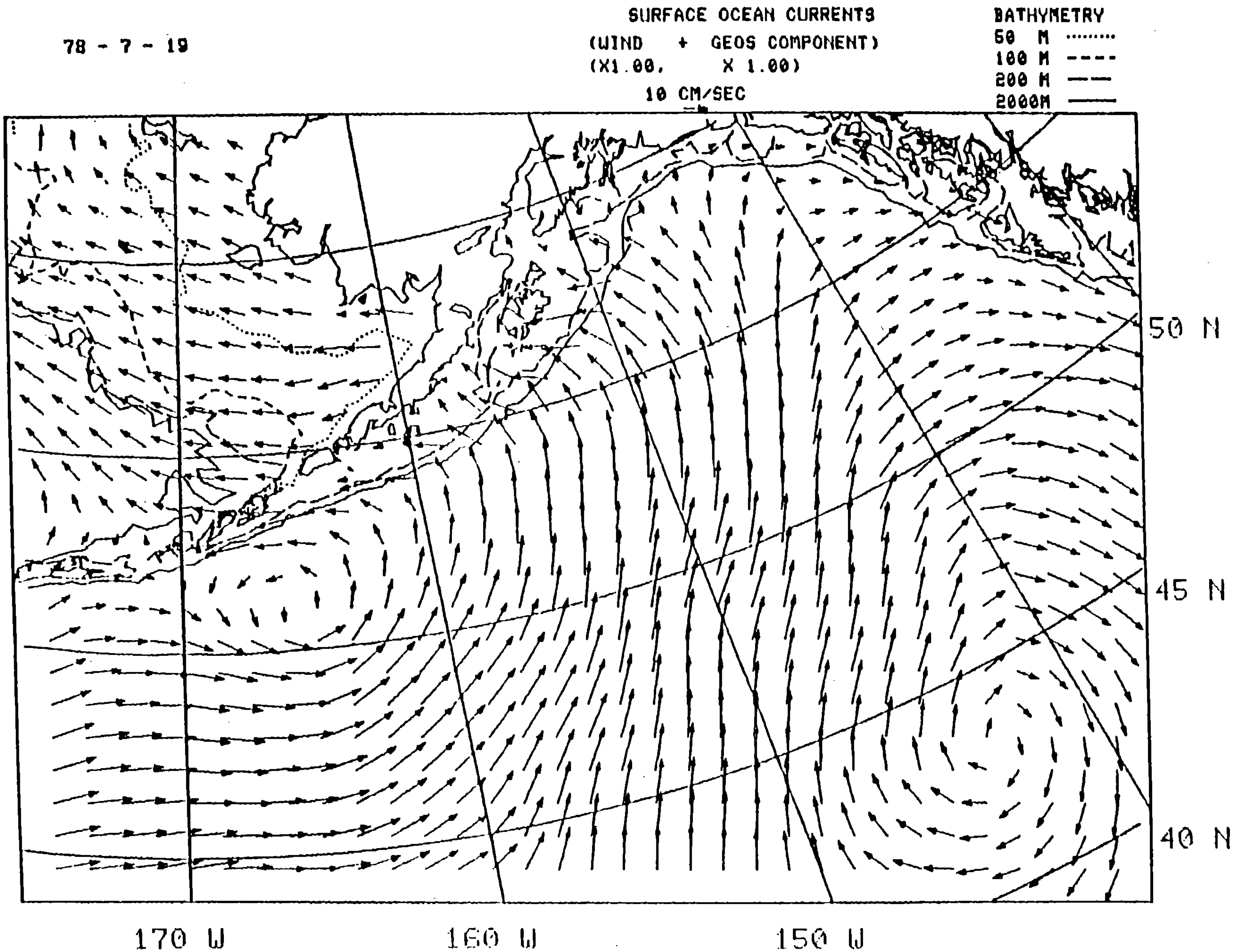

Figure 15(b).--Total surface ocean current vector field for 19 July 1978. Zoom plot of a portion of Figure 8.

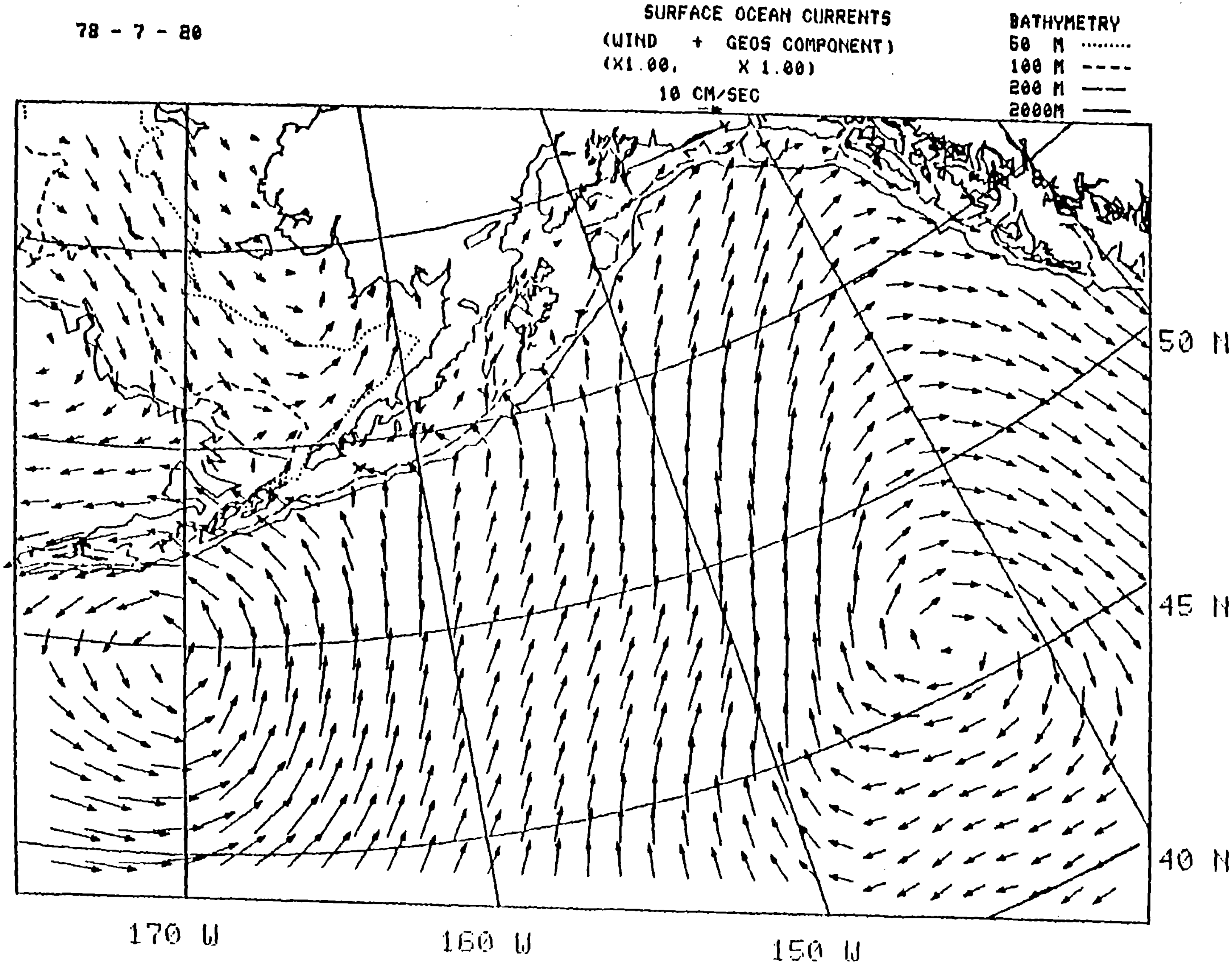

Figure 15(c).--Total surface current vector field for 20 July 1978. Zoom plot of a portion of Figure 8.

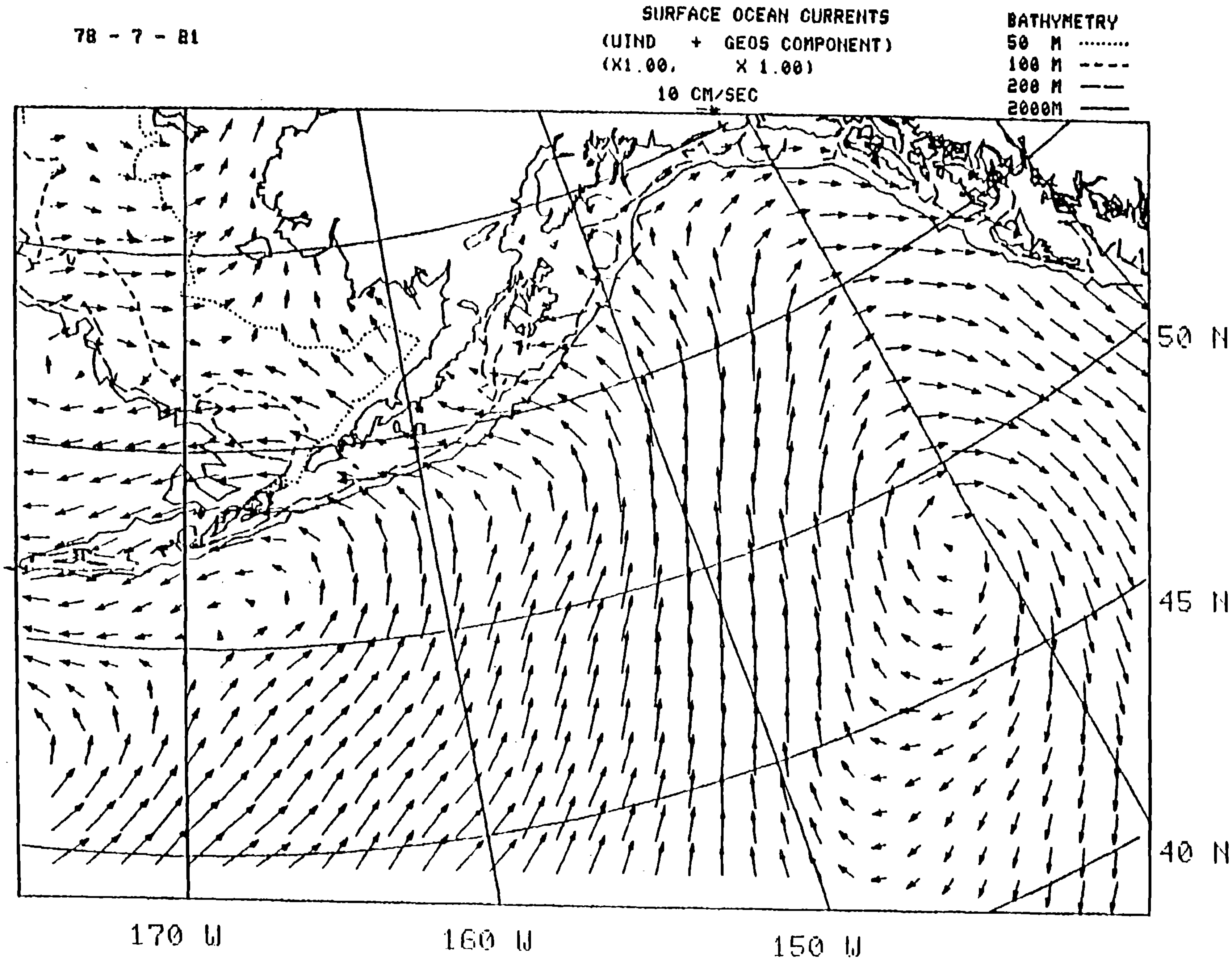

Figure 15(d).--Total surface current vector field for 21 July 1978. Zoom plot of a portion of Figure 8.

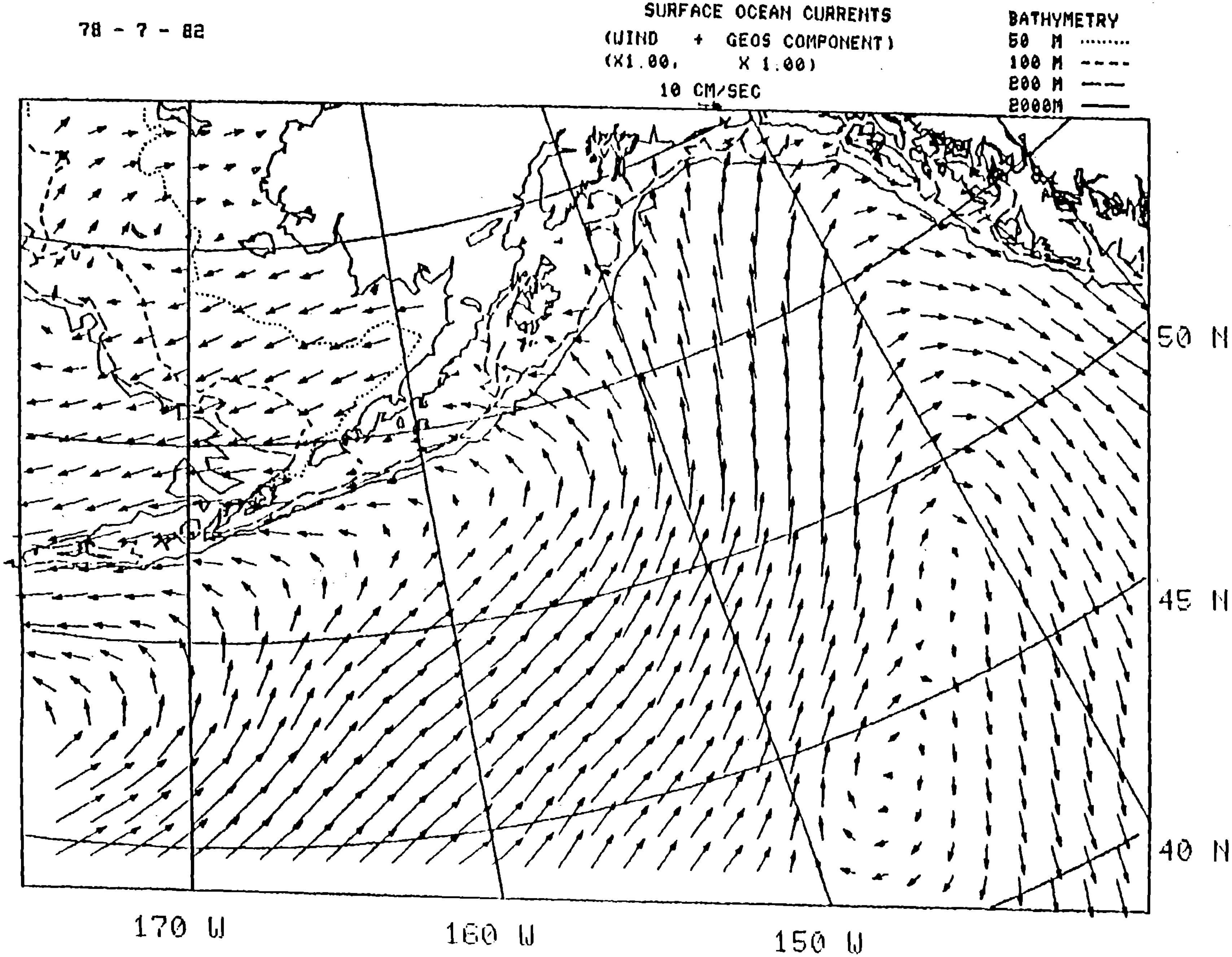

Figure 15(e).--Total surface current vector field for 22 July 1978. Zoom plot of a portion of Figure 8.

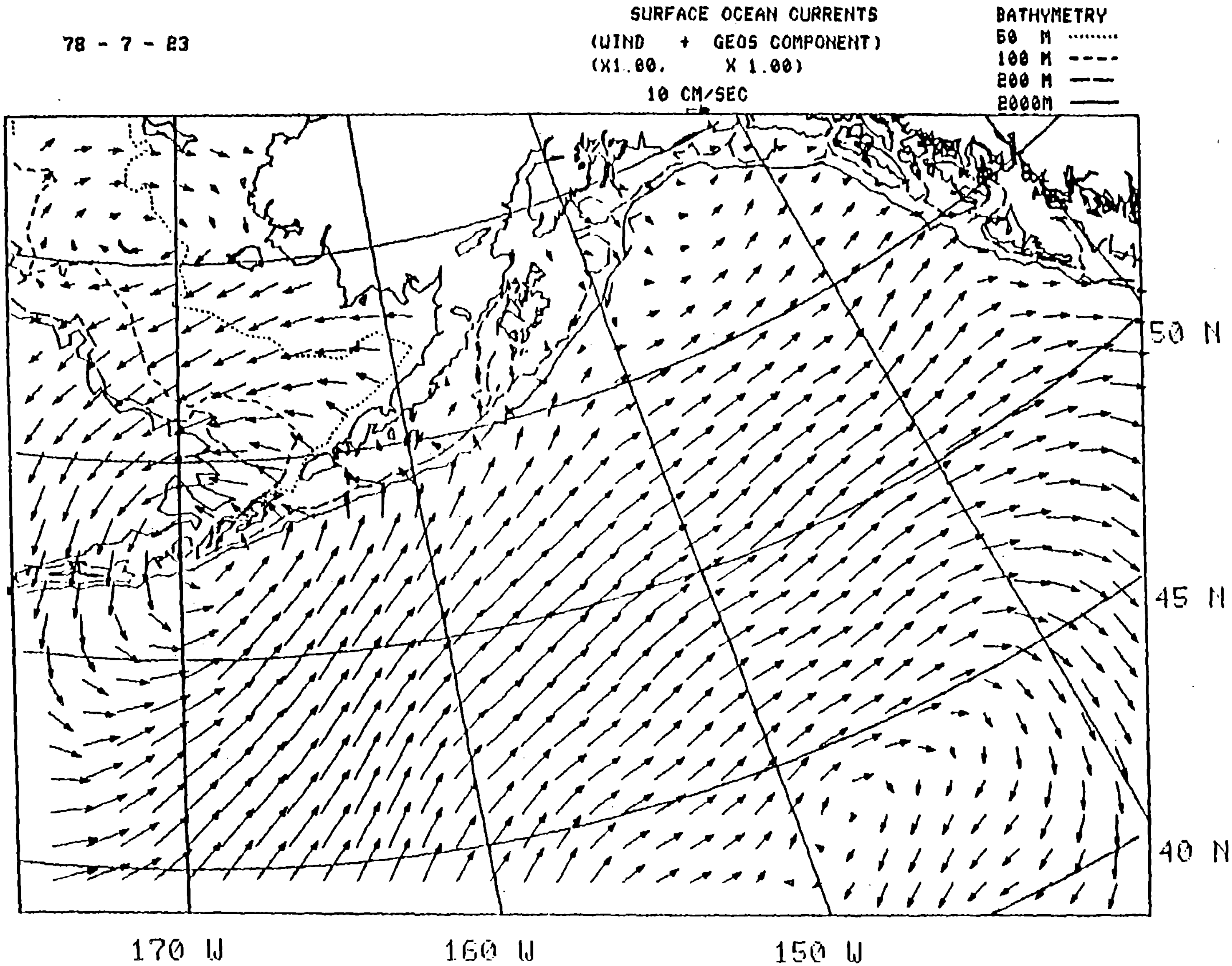

Figure 15(f). --Total surface current vector field for 23 July 1978. Zoom plot of a portion of Figure 8.

stronger if a winter series of days had been selected in this example.

Progressive Vector Tracks

Now that we have the complete total current vector field, recall that the last step in the model run loop was to compute a surface transport vector at a number of selected starting points to get the actual drift distance per day using velocities that were interpolated from this field at these points for the 24-hour average. Starting points for the next day are updated to be the end points of the previous day's transport vectors. These transport vectors have been saved on disk in a series of daily start and end point locations in grid units so that all the plot program has to do now is connect these start and end points day by day and do the map graphics to create the progressive vector tracks.

Zoom plots of progressive transport vectors are of particular interest because they summarize the model results over time from some desired historical starting date to a number of days hence. They also provide a mechanism for comparisons of model results with ocean drifter experiments to be used in calibration and tuning of the model. In Figure 16(a), a zoom plot of Figure 10(a), drifter tracks are simulated for about five and one-half months beginning in mid-July 1978, at nine starting locations which were selected to show the overall flow pattern. Indeed, near-surface ocean water drifted around the Gulf of Alaska Gyre as expected with all the tracks moving toward the east to

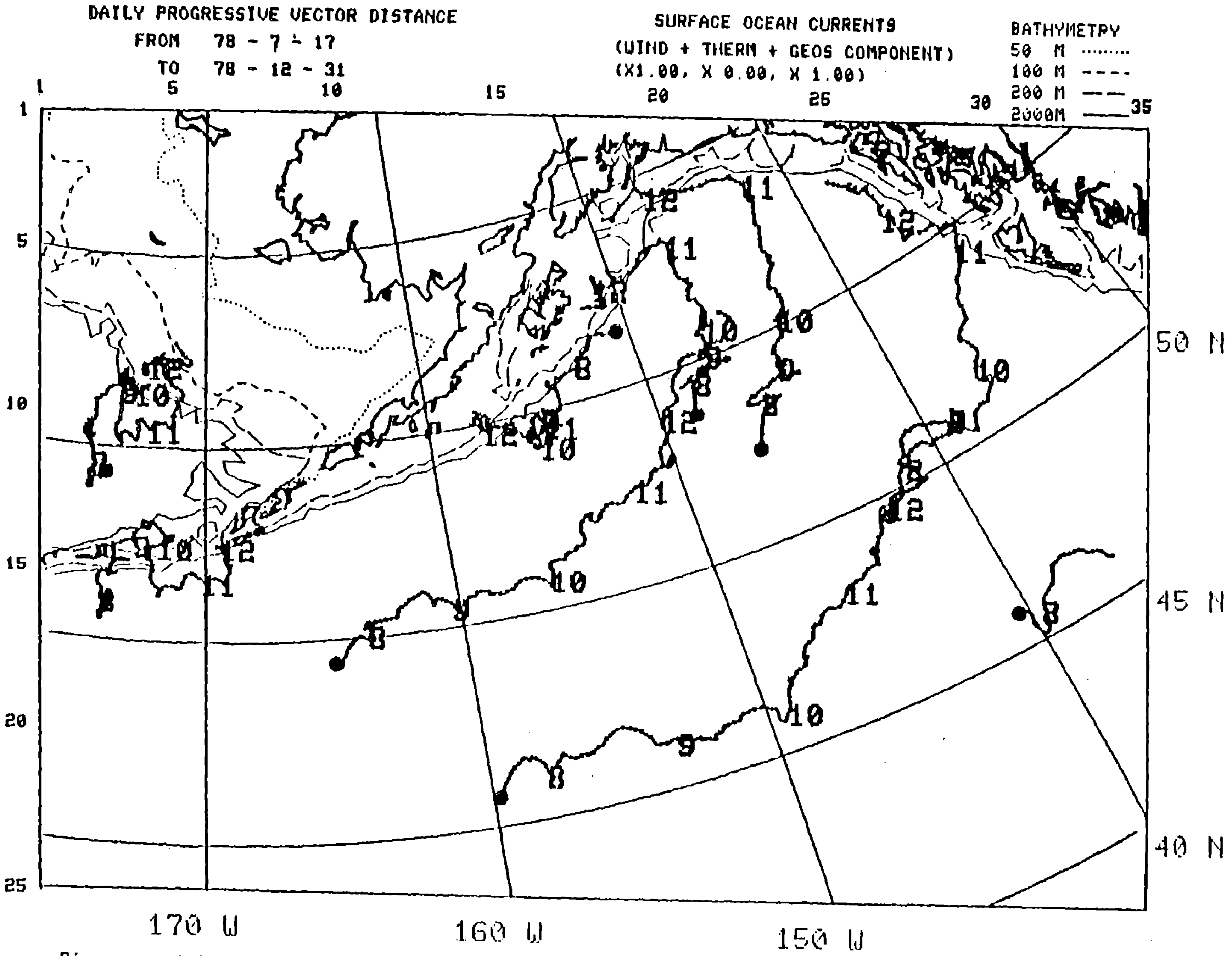

Figure 16(a).--Progressive daily transport vectors-Total surface ocean transport (17 July to 31 December 1978). Zoom plot of Figure 9(a).

northeast except those that enter the Alaskan Stream west of the top of the gulf. Speeds of drift, however, were slower than expected. The starting location of each track is marked by a large dot. Small connected arrows each represent one day of estimated distance drifted (transported), and the large numbers spaced along the tracks indicate the new month at the first day of the new month as it was encountered along the track. One feature evident in all the tracks is the acceleration in flow from the slow start in month 8 to a maximum in month 10; then relatively strong flow continued through month 12. Tracks in the southern branch of the gyre appear steadier than the others with a few small short-term loops, small meanders, and very few reversals. The longest track which started at 46°N-160°W had an average drift of 11 km/day or a speed of about 10 cm/sec. Other tracks in the Alaskan Stream and Bering Sea are shorter and display more looping and random motions. The average drift of 3.5 km/day (3 cm/sec) for the track which started at 57°N-150°W was surprisingly slow for the Alaskan Stream water. The other surprising feature was the absence of the southeastward recirculation of the gyre which has been shown to exist in the vicinity of 160°W (Reed 1980). These figures reflect the final model results at this stage of development. Some insight can be gained about the whys of the unusual behaviors by looking at tracks of individual flow components. In other words, what would drift tracks look like if the flow was purely geostrophic or purely local wind driven.

Pure geostrophic drift (WFAC=0.0) is shown in Figure 16(b), a zoom plot of Figure 10(b). This being a constant field, this flow has no time dependence, that is, tracks started at the same location may be started on any date and drift will follow the same track. Speeds around the gyre varied from 1 to 3 cm/sec in the southern part to about double, 6 cm/sec, in the Alaskan Stream.

Pure wind currents (DDFAC=0.0) are shown in Figure 16(c), a zoom plot of Figure 10(c). In contrast the flow everywhere is toward the northeast, even in the stream. These tracks give a better quantitative estimate of the dominance of wind-induced flow over geostrophic flow. The strongest wind drift track which started at 46°-160°W had an average speed of about 7 cm/sec supporting a geostrophic flow of 2-4 cm/sec in the same direction thus adding to the total flow of about 10 cm/sec. In the Alaskan Stream the wind-only drifter track (57°N-150°W) was the smallest in the whole area, only l-3 cm/sec. Despite the small speed, this consistently opposing wind current effectively subtracted from the larger geostrophic component of 6 cm/sec to the slower drift of 3-4 cm/sec seen in Figure 16(a).

In all of the above discussion, only the pure untuned (DDFAC=1.0 and WFAC=1.0) basic model equations were used. These factors must be tuned as closely as possible to conditions in nature before indices reflecting changing conditions in nature can be computed by the model. A more complete quantitative discussion of statistical averages of model outputs (net drift in km/month, monthly displacement direction in

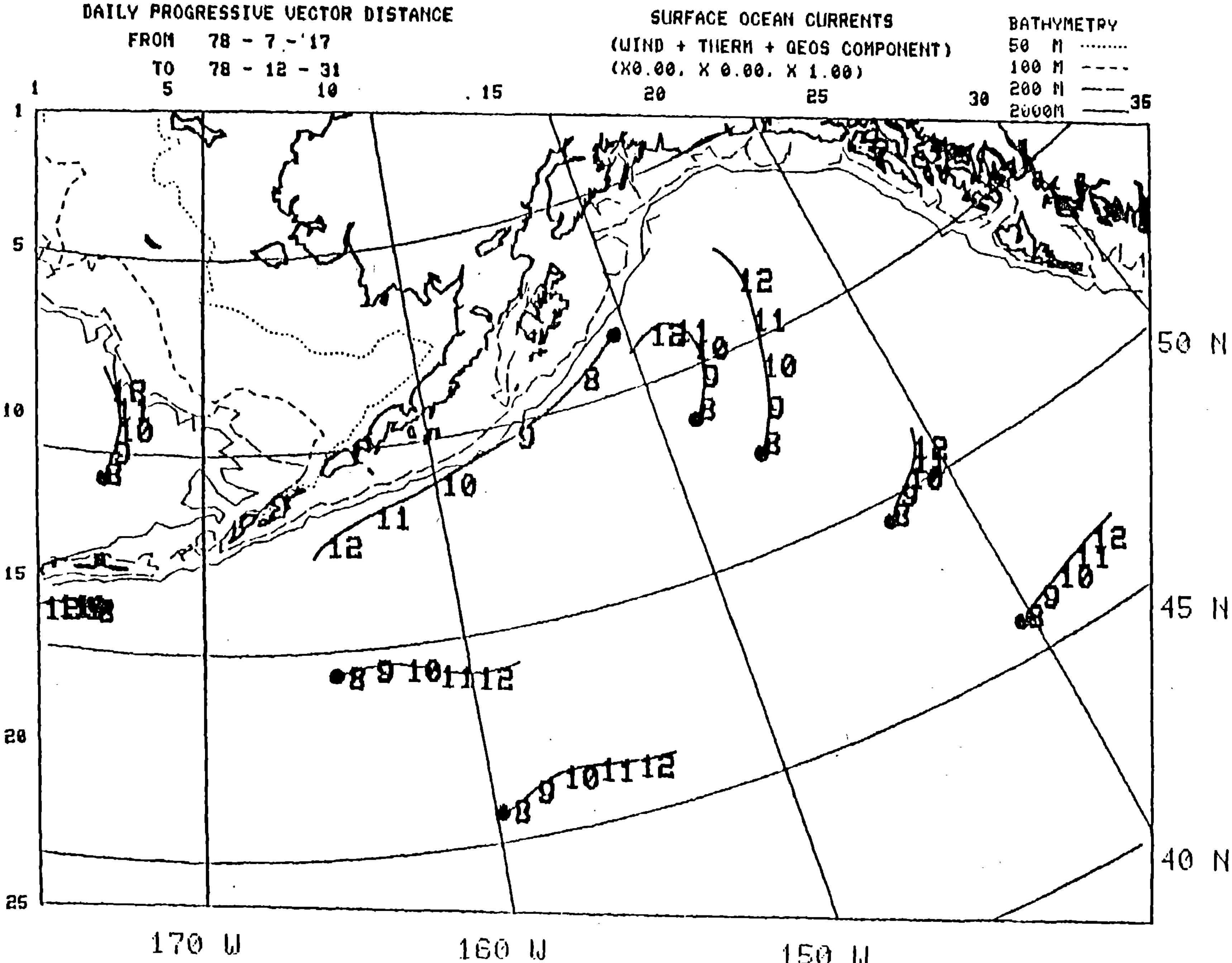

Figure 16(b). --Progressive daily transport vectors-Permanent geostrophic flow only (WFAC=0.0) (17 July to 31 December 1978). Zoom plot of Figure 9(b).

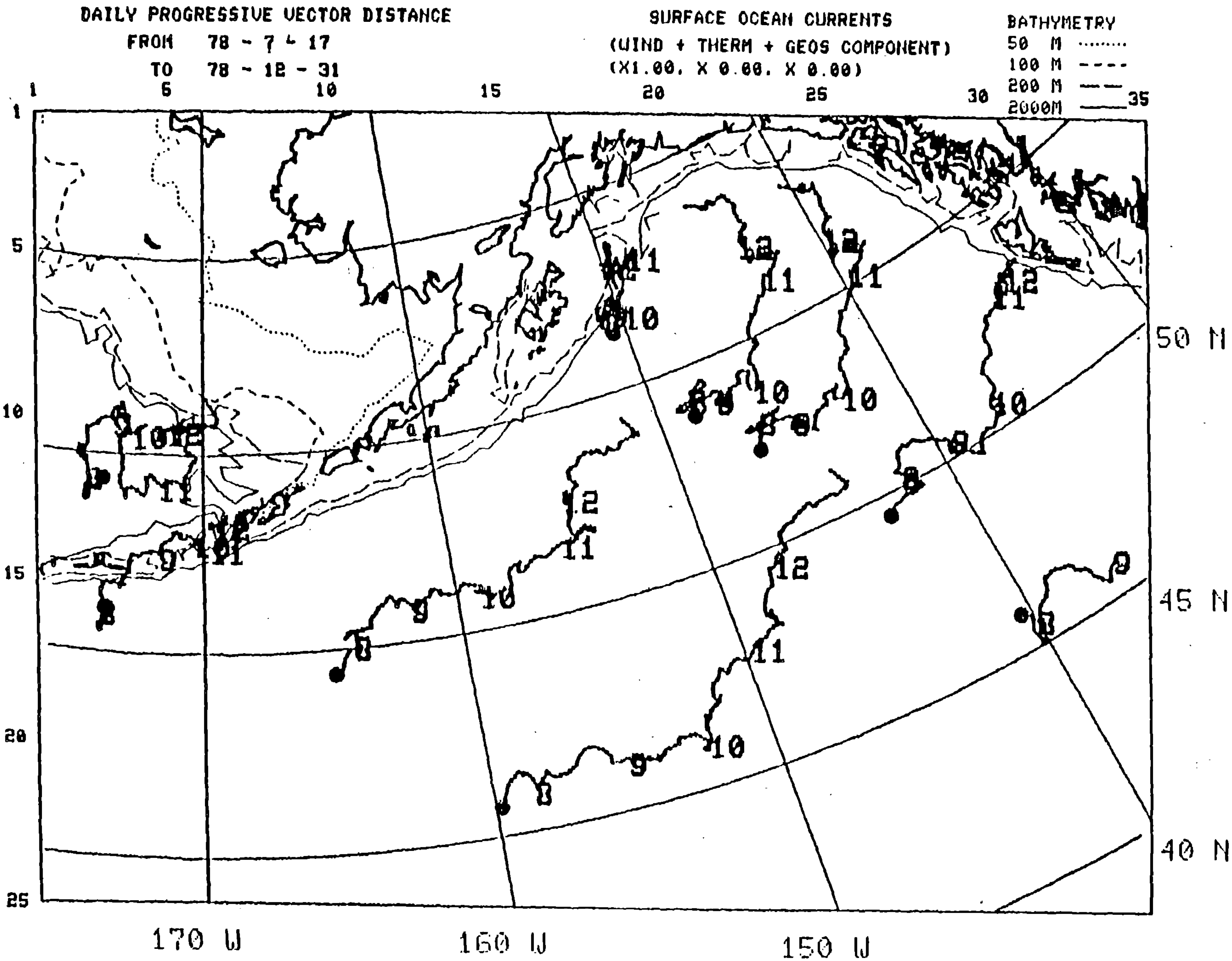

Figure 16(c). --Progressive daily transport vectors-Surface current due to wind only (DDFAC=0.0) (17 July to 31 December 1978). Zoom plot of Figure 9(c).

degrees true, etc.) will be reserved for a later date when tuning and calibration are completed. Refer to the next report in this series of model developments for a full discussion of tuning results and comparisons with actual ocean drifter track data. As a preview, the untuned model was run for one month starting at each drifter location on the first of each month, that is, three locations for months 8, 9, 10, 11, and 12 (Fig. 17). Note the lack of agreement in the Alaskan Stream area and the fair agreement in the rest of the Gulf of Alaska Gyre.

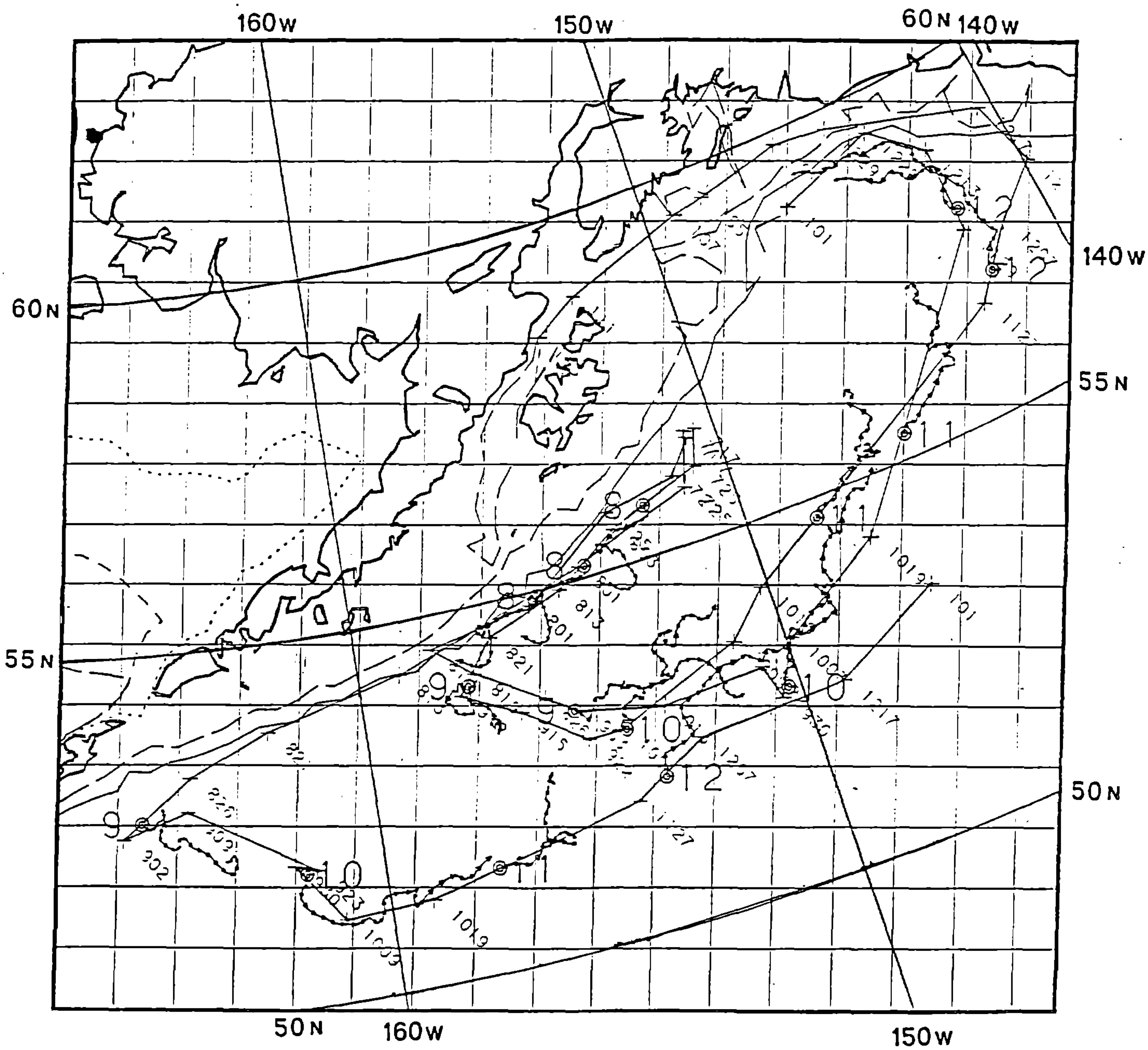

Figure 17.--Three drifter tracks from Reed (1980) and simulated drifter tracks (daily progressive transport vectors) from 30 day runs of the untuned OSCURS numerical model starting at the first day of Months 8, 9, 10, 11, and 12.

REFERENCES

Bauer, R., and M. Robinson.

1985. Description of the Bauer-Robinson numerical atlas, version VIII, February 1985. Compass Systems, Inc., 4640 Jewel1 St., #204, San Diego, CA 92109, 13 p.

Hela. I.

1952. Drift currents and permanent flow. Soc. Sci. Fennica, Comm. Phys. Math. 16(4):1-28.

Hubert, W. E., and T. Laevastu.

1985. Synoptic analysis and forecasting of surface currents. U.S. Navy Fleet Numerical Weather Facility, Monterey, CA. Technical Note 9., 47 p.

Ingraham, W. J., Jr., N. Pola Swan, Y. Lee, R. Miyahara, and M. Hayes.

1983. Processing of FNWC daily sea level pressure data on NWAFC Burroughs 7811. REFM Program Doc. 19, 101 p. Northwest and Alaska Fish. Cent., Natl. Mar. Fish. Serv., NOAA, 2725 Montlake Blvd. E., Seattle, WA 98112.

Ingraham, W. James Jr., and F. Favorite.

1968. The Alaskan Stream south of Adak Island. Deep-Sea Res. 15(4):493-496.

LaFonde, Eugene C.

1951. Processing oceanographic data. U.S. Navy Hydrographic Office, Washington D. C. Pub. 614, 114 p.

Reed, R. K.

1980. Direct Measurements of Recirculation in the Alaskan Stream. J. Phys. Ocean. l0(6): 976-978.

Swan, Nancy Pola.

1983. User's guide to plotting with the GRIDS Package. REFM Program Doc. No. 17, 52 p. Northwest and Alaska Fish. Cent., Natl. Mar. Fish. Serv., NOAA, 2725 Montlake Blvd. E., Seattle, WA 98112.

Witting, R.

1909. Zur Kenntnie den vom Winde Erzeigtem Oberflachenstromes. Ann. Hydrogr. Marit. Met. 73, 193 p.

APPENDIX 1

Listing of Fortran Program "M/FINAL" OSCURS Model on 40X104 Grid.

```
$RESET FREE
$ SET LINEINFO

FILE  1(KIND=DISK,TITLE="FNWC/LATLON/GRID",FILETYPE=7)
FILE  2(KIND=DISK,FILETYPE=7)
FILE  5(KIND=READER)
FILE  6(KIND=PRINTER)
FILE  12(KIND=DISK,TITLE="NUMBERS/GULF",FILETYPE=7)
FILE  13(KIND=DISK,TITLE="CURRENTS/40X64/LANDSEA",FILETYPE=7)
FILE  14(KIND=DISK,TITLE="CURRENTS/40X64W/LANDSEA",FILETYPE=7)
FILE  25(KIND=DISK,TITLE="CURRENTS/40X64W/COAST/DISTANCES/NORMALS"
      ',FILETYPE=7)
FILE  26(KIND=DISK,TITLE="CURRENTS/40X64/COAST/DISTANCES/NORMALS"
      ',FILETYPE=7)
FILE  28(KIND=DISK,TITLE="CURRENTS/40X104/PROGRESVECT/IJPLOTPOINTS"
      'FILETYPE=7)
FILE  30(KIND=DISK,TITLE="CURRENTS/40X64/LATLON",FILETYPE=7)
FILE  31(KIND=DISK,TITLE="CURRENTS/40X64W/LATLON",FILETYPE=7)
FILE  33(KIND=DISK,TITLE="CURRENTS/40X64/DYNHTS/0/2000DB",FILETYPE=
FILE  34(KIND=DISK,TITLE="CURRENTS/40X64W/DYNHTS/0/2000DB",FILETYPE
FILE  38(KIND=DISK,TITLE="CURRENTS/40X104/DYNCURR/F",MAXRECSIZE=14,
      'BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  95(KIND=DISK,TITLE="RED",MAXRECSIZE=14,BLOCKSIZE=420,
      'NEWFILE=.TRUE.)
FILE  96(KIND=DISK,TITLE="ANG",MAXRECSIZE=14,BLOCKSIZE=420,
      'NEWFILE=.TRUE.)
FILE  50(KIND=DISK,TITLE="CURRENTS/40X104/DYNCURR/S2",FILETYPE=7)
FILE  51(KIND=DISK,
      'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  52(KIND=DISK,
      'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  53(KIND=DISK,
      'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  54(KIND=DISK,
      'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  55(KIND=DISK,
      'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  60(KIND=DISK,
      'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
       COMMON ANX(40,104),ANY(40,104),BB(40,104),LS3(40,104),
      'GDIST(40,104)
       COMMON/GRDCOM/CFACT,M,N,NM,XI(105),YJ(41)
       COMMON/LAT/ALAT2,LS2
       COMMON/LON/ALON2
       DIMENSION U(880),V(880),LS(40,104),ALAT(40,104),ALON(40,104),
       DIMENSION B(2)
       DIMENSION C(1)
       DIMENSION ALAT2(40,104),ALON2(40,104)
       DIMENSION LS2(40,104),LS1(40,104)
```

```
      DIMENSION OCUW(40,104),OCVW(40,104)
      DIMENSION CSPEED(40,104)
      DIMENSION WXX(40,104),WYY(40,104)
      DIMENSION CSLP(2560)
      DIMENSION T1(40,104),T2(40,104)
      DIMENSION SLP1(40,104)
      DIMENSION HWX(1),HWY(1)
      DIMENSION HSLP(1),HSFOU(1),HSFOV(1)
      DIMENSION DEPTH(4),NDATA(4),BATH(3,1000,4)
      DIMENSION OCUS(40,104),OCVS(40,104)
      DIMENSION OCUDD(40,104),OCVDD(40,104)
      DIMENSION GU1(40,104),GV1(40,104),GU2(40,104),GV2(40,104)
      DIMENSION IPV(500),JPV(500)
      DIMENSION GIPV(500),GJPV(500)
      DIMENSION ASP(10,100),AKTS(10),AMPS(10)
      DIMENSION CC(72),ANGDEF(30),SPJ(30)
      DIMENSION IAK(100), JAK(100), DDAK(100)
      DATA A/"    SLP    YEAR=  "/
      DATA B/", MONTH="/
      DATA C/", DAY="/
      DATA HWX/' WX   '/,HWY/' WY   '/
      DATA HSLP/'SLP   '/
      DATA HSFOU/'OU0   '/,HSFOV/'OV0   '/
      NN=20;MM=44; NM=NN*MM
      NN2=40
      MM2=104
C
C
C  SET CHOICES
C--------------
C  START MONTH AND NUMBER OF DAYS TO RUN
      IMSTRT=7
      IMEND=12
      NDAYS=366
C     NDAYS=121
C     NDAYS=15
      NFILE=51
C
C  SET TUNING FACTORS
      TFAC=0.0
      WFAC=2.0
      DDFAC=0.00
      AKSTRM=1.0
C
C  SET PRINT FACTOR (NO PRINT, >>>> NOPR=1)
      NOPR=1
      NOPR=0
C
C  SET STORAGE OF FILES (NOT KEEP FILES, NOSTOR=1)
      NOSTOR=1
C     NOSTOR=0
C  SET STORAGE OF FINAL COMPUTED DAILY FIELDS (DO NOT STORE =1)
C  THIS IS SUCH A LARGE FILE (58000 SECTORS FOR 4 MONTHS)
      NOSTOF=1
      NOSTOF=0
```

```
C  CALL DELTAD ONLY TO STORE A NEW GEOSTROPHIC CURRENT FILE
C     (SET NEWDD = 1 TO CREATE NEW FILE)
      NEWDD=1
C---------------
C
C  READ IN CONTROL CARD PARAMETERS FROM WFL CARD DECK INPUT
      WRITE (6,80)
   80 FORMAT ('1CONTROL CARD PARAMETERS FROM INPUT WFL CARD DECK'//
   84 FORMAT(I5,67A1)
   85 FORMAT(I5,67A1/)
   86 FORMAT(F5.1,67A1/)
   87 FORMAT(F5.1,67A1)
      READ (5,84)IYRSLP,(CC(I),I=1,67)
      WRITE(6,85)IYRSLP,(CC(I),I=1,67)
      READ (5,84)IYRSLP,(CC(I),I=1,67)
      WRITE(6,85)IYRSLP,(CC(I),I=1,67)
      WRITE(6,85)IMSTRT
      READ (5,84)IMSTRT,(CC(I),I=1,67)
      WRITE(6,85)IMSTRT,(CC(I),I=1,67)
      WRITE(6,85)IMEND
      READ (5,84)IMEND,(CC(I),I=1,67)
      WRITE(6,85)IMEND,(CC(I),I=1,67)
      WRITE(6,85)NDAYS
      READ (5,84)NDAYS,(CC(I),I=1,67)
      WRITE(6,85)NDAYS,(CC(I),I=1,67)
      WRITE(6,86)WFAC
      READ (5,87)WFAC,(CC(I),I=1,67)
      WRITE(6,86)WFAC,(CC(I),I=1,67)
      WRITE(6,86)DDFAC
      READ (5,87)DDFAC,(CC(I),I=1,67)
      WRITE(6,86)DDFAC,(CC(I),I=1,67)
      WRITE(6,86)AKSTRM
      READ (5,87)AKSTRM,(CC(I),I=1,67)
      WRITE(6,86)AKSTRM,(CC(I),I=1,67)
      WRITE(6,85)NEWDD
      READ (5,84)NEWDD,(CC(I),I=1,67)
      WRITE(6,85)NEWDD,(CC(I),I=1,67)
      WRITE(6,85)NOPR
      READ (5,84)NOPR,(CC(I),I=1,67)
      WRITE(6,85)NOPR,(CC(I),I=1,67)
      WRITE(6,85)NOSTOR
      READ (5,84)NOSTOR,(CC(I),I=1,67)
      WRITE(6,85)NOSTOR,(CC(I),I=1,67)
      WRITE(6,85)NOSTOF
      READ (5,84)NOSTOF,(CC(I),I=1,67)
      WRITE(6,85)NOSTOF,(CC(I),I=1,67)
C  SET UP FNOC GRID (LAT-LONG,LAND-SEA)
C
C  READ IN LAT-LONG OF 20X44 GRID (FNWC GRID POINTS)
      DO 300 I=1,20
      DO 300 J=1,44
  300 READ(1,1200,END=310)IDUM1,IDUM2,(ALAT(I,J),ALON(I,J))
 1200 FORMAT(2I5,2F10.5)
C
C  TEST WRITE LATS AND LONGS (20X44 GRID BY 2'S)
```

```
C     WRITE(6,311)
  311 FORMAT('1 FNWC LAT-LONGS'//)
C     DO 301 I=1,20
C     WRITE (6,304) (ALAT(I,J),J=1,44,2)
  301 CONTINUE
  304 FORMAT(23F6.1)
C     DO 302 I=1,20
C     WRITE (6,304) (ALON(I,J),J=1,44,2)
  302 CONTINUE
C
C
C  LOOK AT GRID DISTANCES NEAR 170 W (WHERE GRID IS ALONG A MERIDIA
C  TEST GRID LENGTH FUNCTION (GRIDLN) VS. LATITUDE
      WRITE(6,311)
      NBR=0
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      SUM4=0.0
      DO 308 I=2,19
      DAA=ALAT(I,22)-ALAT(I+1,22)
      DAA2=ALAT(I-1,22)-ALAT(I+1,22)
      DLAT=ALAT(I,22)-(DAA/2.0)
      DLAT2=ALAT(I,22)
      DAANM=DAA*60.0
      DAANM2=DAA2*60.0
      DAANM2=DAANM2/2.0
      DAAKM=DAANM*1.852
      DAAKM2=DAANM2*1.852
      AKM=(1.0+(SIN(DLAT*3.14159/180.0)))/1.86603
      AKM2=(1.0+(SIN(DLAT2*3.14159/180.0)))/1.86603
      GD60=DAAKM/AKM
      GD602=DAAKM2/AKM2
      WRITE (6,299) (ALAT(I,J),J=15,25)
      WRITE (6,307) I,ALAT(I,22),DLAT,DAA,DAANM,DAAKM,AKM,GD60
      WRITE (6,307) I,ALAT(I,22),DLAT2,DAA2,DAANM2,DAAKM2,AKM2,GD60
     '2
      NBR=NBR+1
      SUM1=SUM1+DAAKM
      SUM2=SUM2+GD60
      SUM3=SUM3+DAAKM2
      SUM4=SUM4+GD602
  308 CONTINUE
      AV1=SUM1/FLOAT(NBR)
      AV2=SUM2/FLOAT(NBR)
      AV3=SUM3/FLOAT(NBR)
      AV4=SUM4/FLOAT(NBR)
      WRITE(6,307)NBR,AV1,AV3,AV2,AV4
      DO 309 I=1,20
      WRITE (6,299) (ALON(I,J),J=15,25)
  309 CONTINUE
  307 FORMAT(I5,20X,7F11.6)
C
  299 FORMAT(11F11.6)
C
```

```
C
      WRITE (6,61)
      DO 65 N=1,56
      GLA=FLOAT(71-N)
      GG=GDFNOC(GLA)
      GGNM=GG/1.852
      WRITE(6,62)N,GLA,GG,GGNM
   65 CONTINUE
  310 CONTINUE
C
C
C  SET UP CURRENTS (40X104) GRID
C
C  FILL LANDSEA TABLE FOR SLP COMPUTATIONS (ALL=1)
      DO 211 I=1,40
      DO 211 J=1,104
  211 LS1(I,J)=1
C
C  READ IN LAT-LONGS OF 40X64 + 40X64W GRIDS (CURRENTS MODEL)
C  FILL RIGHT SIDE THEN LEFT SIDE TO CREATE WHOLE OCEAN ARRAY
C
      DO 630 IFILE=1,2
      DO 631 IK=1,4
      DO 632 J=41,104
      I2=IK*10
      I1=I2-9
      IF(IFILE.EQ.1)READ(30,104)(ALAT2(I,J),I=I1,I2)
      IF(IFILE.EQ.2)READ(30,104)(ALON2(I,J),I=I1,I2)
  632 CONTINUE
  631 CONTINUE
  630 CONTINUE
  104 FORMAT (10F8.3,I4)
      DO 700 IFILE=1,2
      DO 701 IK=1,4
      DO 702 J=1,64
      I2=IK*10
      I1=I2-9
      IF(IFILE.EQ.1)READ(31,104)(ALAT2(I,J),I=I1,I2)
      IF(IFILE.EQ.2)READ(31,104)(ALON2(I,J),I=I1,I2)
  702 CONTINUE
  701 CONTINUE
  700 CONTINUE
C
C  ROUGH PRINT LATS AND LONGS (EVERY 5TH COL.)
C     WRITE(6,59)
   59 FORMAT ('1 CURRENTS (40X104) LAT--LONGS (EVERY 5TH LONG)'//)
C     DO 51 I=1,40
C     WRITE(6,105) (ALAT2(I,J),J=1,104,5),ALAT2(I,104)
   51 CONTINUE
      DO 52 I=1,40
C     WRITE(6,105) (ALON2(I,J),J=1,104,5),ALON2(I,104)
   52 CONTINUE
  105 FORMAT (22F6.1)
C
C  LOOK AT GRID DISTANCES NEAR 170 W (WHERE GRID IS ALONG A MERIDIA
```

```
C  TEST GRID LENGTH FUNCTION (GRIDLN) VS. LATITUDE
      WRITE(6,59)
      NBR=0
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      SUM4=0.0
      DO 56 I=2,39
      DAA=ALAT2(I,61)-ALAT2(I+1,61)
      DAA2=ALAT2(I-1,61)-ALAT2(I+1,61)
      DLAT=ALAT2(I,61)-(DAA/2.0)
      DLAT2=ALAT2(I,61)
      DAANM=DAA*60.0
      DAANM2=DAA2*60.0
      DAANM2=DAANM2/2.0
      DAAKM=DAANM*1.852
      DAAKM2=DAANM2*1.852
      AKM=(1.0+(SIN(DLAT*3.14159/180.0)))/1.86603
      AKM2=(1.0+(SIN(DLAT2*3.14159/180.0)))/1.86603
      GD60=DAAKM/AKM
      GD602=DAAKM2/AKM2
      WRITE (6,299) (ALAT2(I,J),J=55,65)
      WRITE (6,307) I,ALAT2(I,61),DLAT,DAA,DAANM,DAAKM,AKM,GD60
      WRITE (6,307) I,ALAT2(I,61),DLAT2,DAA2,DAANM2,DAAKM2,AKM2,GD6
     '2
      NBR=NBR+1
      SUM1=SUM1+DAAKM
      SUM2=SUM2+GD60
      SUM3=SUM3+DAAKM2
      SUM4=SUM4+GD602
   56 CONTINUE
      AV1=SUM1/FLOAT(NBR)
      AV2=SUM2/FLOAT(NBR)
      AV3=SUM3/FLOAT(NBR)
      AV4=SUM4/FLOAT(NBR)
      WRITE(6,307)NBR,AV1,AV3,AV2,AV4
      DO 57 I=1,20
      WRITE (6,299) (ALON2(I,J),J=55,65)
   57 CONTINUE
C
      WRITE (6,61)
   61 FORMAT('1GRID LENGTH (KM) VS. LATITUDE'//)
      DO 60 N=1,37
      GLA=FLOAT(67-N)
      GG=GRIDLN(GLA)
      GGNM=GG/1.852
      WRITE(6,62)N,GLA,GG,GGNM
   60 CONTINUE
   62 FORMAT(I4,'LATITUDE=',F6.2,'   GRID LENGTH=',F11.6,' KM',5X,
     'F11.6,' NM')
C
C  READ LAND-SEA TABLE (LEFT AND RIGHT)
      DO 1500 I=1,NN2
      READ(14,1501,END=1503)(LS2(I,J),J=1,64)
C     WRITE(6,1505) I,LS2(I,1)
```

```
 1500 CONTINUE
 1503 CONTINUE
 1505 FORMAT (1H ,2I5)
 1501 FORMAT(64I1)
      DO 1600 I=1,NN2
      READ(13,1501,END=1603)(LS2(I,K),K=41,104)
C     WRITE(6,1505) I,LS2(I,41)
 1600 CONTINUE
 1603 CONTINUE
C  WRITE LAND-SEA
      WRITE(6,1514)
 1514 FORMAT('1 LAND-SEA 2, (40X104)'//)
      DO 1515 I=1,NN2
      WRITE(6,1516)(LS2(I,J),J=1,104)
 1515 CONTINUE
 1516 FORMAT(1H ,104I1)
      WRITE(6,110)
  110 FORMAT('1LANDSEA 3'//)
      DO 220 I=1,40
      DO 221 J=1,MM2
      LS3(I,J)=LS2(I,J)
  221 CONTINUE
      WRITE(6,1509) (LS3(I,JK),JK=1,MM2),I
  220 CONTINUE
 1509 FORMAT(1H ,104I1,I5)
C
C  TO ROTATE CURRENT VECTORS IMPINGING ON THE COAST
C    READ IN DISTANCES AND NORMALS TO THE COASTLINE
C    FOR EACH GRID POINT WHICH IS WITHIN 2 GRID LENGTHS OF THE COAS
C  DATA ARE USED IN COMPUTING CURRENT VECTOR ROTATION AND REDUCTION
C    FOR VECTORS WHICH COME WITHIN 2 GRID POINTS OF SHORE
C  AN EXPONENTIAL FUNCTION IS USED TO REDUCE THE MAGNITUDE
C    DEPENDING UPON THE DISTANCE FROM SHORE
C    AND ROTATION IS BASED ON BOTH THE DIRECTION AND MAGNITUDE
C    OF THE ANGLE OF INCIDENCE
C  AGAIN THE OCEAN WIDE ARRAY IS FILLED FROM THE TWO SMALLER
C    FILES OF THE EAST AND WEST PREVIOUS VERSIONS OF M(MODEL)
C 324 READ(25,325,END=329)I,J,GDIST(I,J),ANY(I,J),ANX(I,J)
  324 READ(25,325,END=329)I,J,V1,VV,V3
      GDIST(I,J)=V1
      ANY(I,J)=VV
      ANX(I,J)=V3
C     WRITE(6,327)I,J,GDIST(I,J),ANY(I,J),ANX(I,J)
      LS3(I,J)=-1
      GO TO 324
  325 FORMAT(2I3,F5.2,F6.2,F6.2)
  327 FORMAT(' WEST  ',2I3,F5.2,F6.2,F6.2)
  329 CONTINUE
C  READ IN DISTANCES AND NORMALS FROM FILE 26
C 424 READ(26,425,END=429)I,J,GDIST(I,J),ANY(I,J),ANX(I,J)
  424 READ(26,425,END=429)I,J,V1,VV,V3
C  OFFSET FOR EASTERN GRID
      J=J+40
      GDIST(I,J)=V1
      ANY(I,J)=VV
```

```
      ANX(I,J)=V3
C     WRITE(6,427)I,J,GDIST(I,J),ANY(I,J),ANX(I,J)
      LS3(I,J)=-1
      GO TO 424
  425 FORMAT(2I3,F5.2,F6.2,F6.2,)
  427 FORMAT(' EAST  ',2I3,F5.2,F6.2,F6.2,)
  429 CONTINUE
C
C  TEMPORARY LAND-SEA TABLE = ALL 1'S
      WRITE(6,110)
      DO 1506 I=1,40
      DO 1507 J=1,104
      LS1(I,J)=1
 1507 CONTINUE
      WRITE(6,1509) (LS3(I,JK),JK=1,MM2),I
 1506 CONTINUE
C
C
C  --------------------------------------------------------------
C  READ INITIAL I,J PLOINT TO PLOT IN PROGRESSIVE VECTOR PLOT
C  --------------------------------------------------------------
      KK=1
 1144 READ(28,1145,END=1146) IPV(KK),JPV(KK),GIPV(KK),GJPV(KK)
      WRITE(6,1147) KK,IPV(KK),JPV(KK),GIPV(KK),GJPV(KK)
      IF(IPV(KK).LE.0.OR.JPV(KK).LE.0) GO TO 1146
      KK=KK+1
      GO TO 1144
 1145 FORMAT(2I5,2F6.1)
 1147 FORMAT (3I5,2F6.1)
 1146 CONTINUE
      KK=KK-1
      WRITE(6,1145) KK
      IEND=KK
C
      LSKIP=1
      IF(LSKIP.EQ.1) GO TO 8888
C  SET STARTING GRID POINTS FOR PROGRESSIVE VECTORS
C  DO EVEN NUMBERED GRID POINTS BY 4'S
C  SAVE ONLY EVERY 4 GRID POINTS FOR PLOT AND TAPE FILES
      LC=1
C  HORIZONTAL (12 TO 96 BY 4'S)
      DO 40 L2=1,22
      LJ=12+((L2-1)*4)
C  VERTICAL (8 TO 36 BY 4'S)
      DO 41 L1=1,8
      LI=8+((L1-1)*4)
      IF (LS3(LI,LJ).EQ.0) GO TO 41
      IPV(LC)=LI
      JPV(LC)=LJ
      LC=LC+1
   41 CONTINUE
   40 CONTINUE
      KK=LC-1
      WRITE(6,1145) KK
      IEND=KK
```

```
 8888 CONTINUE
C
C
C  INITIALIZE GRID PLOT (UNITS ARE IN GRID VALUES)
C  RETAIN ACTUAL GRID UNITS IN GU AND GV ARRAYS
C  STARTING AT THE UPPER LEFT CORNER OF GRID (AS SEEN IN MAP VIEW)
C  I INCREASES IN THE MINUS Y DIRECTION FROM 1 TO NN2(40)
C  J INCREASES IN THE PLUS X DIRECTION FROM 1 TO MM2(104)
      DO 3105 I=1,NN2
      DO 3106 J=1,MM2
      GU1(I,J)=J
      GV1(I,J)=I
 3106 CONTINUE
 3105 CONTINUE
C
C  ADD SPECIAL HALF GRID POINTS TO START COMPUTING
      GU1(10,73)= 72.5
      GV1(10,73)=10.5
      GU1(11,74)=73.5
      GV1(11,74)=11.5
C     IF(1.EQ.1) GO TO 3190
      DO 3189 K=1,KK
      GU1(IPV(K),JPV(K))=GJPV(K)
      GV1(IPV(K),JPV(K))=GIPV(K)
      WRITE(6,1147)K,IPV(K),JPV(K),GU1(IPV(K),JPV(K)),GV1(IPV(K),JP
 3189 CONTINUE
 3190 CONTINUE
C
C
C -------------------------------------------------------------
C COMPUTE GEOSTROPHIC OCEAN CURRENTS 0/2000 DB
C CALL DELTAD ONLY TO STORE A NEW GEOSTROPHIC CURRENT FILE
C   (SET NEWDD = 1 TO CREATE NEW FILE)
C     NEWDD=1
C -------------------------------------------------------------
      IF (NEWDD.EQ.1) CALL DELTAD
C
C -------------------------------------------------------------
C READ PERMANENT FILE OF GEOSTROPHIC CURRENTS, U THEN V)
C -------------------------------------------------------------
      DO 38 J=1,104
      DO 38 I=1,5
      I2=I*8  ; I1=I2-7
      READ(50,100,END=218)(OCUDD(IK,J),IK=I1,I2),KYR8,MO8,KDY8
   38 CONTINUE
  218 CONTINUE
      DO 39 J=1,104
      DO 39 I=1,5
      I2=I*8  ; I1=I2-7
      READ(50,100,END=219)(OCVDD(IK,J),IK=I1,I2),KYR9,MO9,KDY9
   39 CONTINUE
  219 CONTINUE
C
C  OCEAN GEOSTROPHIC CURRENT VECTOR SPEED
      DO 954 I=1,NN2
```

```
      DO 955 J=1,MM2
      CSPEED(I,J)=SQRT((OCUDD(I,J)*OCUDD(I,J))+(OCVDD(I,J)*OCVDD(I,
  955 CONTINUE
  954 CONTINUE
C     IF (NOPR.EQ.1) GO TO 957
      WRITE(6,956) KYR,IMO,IDY
  956 FORMAT('1ORIGINAL ** GEOSTROPHIC SPEED (CM/SEC)',3I3//)
      CALL PRINT2(CSPEED,1.0)
C
C  READ IN ALASKAN STREAM AMPLIFICATION FACTOR
      K=1
  601 READ(12,602,END=609)IAK(K),JAK(K),DDAK(K)
      DDAK(K)=DDAK(K)*AKSTRM
      WRITE(6,604) K,IAK(K),JAK(K),DDAK(K)
      K=K+1
      GO TO 601
  609 CONTINUE
  602 FORMAT(2I4,F7.2)
  604 FORMAT (3I4,F7.2)
      DO 406 J=1,104
      DO 407 I=1,40
      DDAMP=1.0
      DO 611 KK=1,K
      IF(IAK(KK).EQ.I.AND.JAK(KK).EQ.J)GO TO 651
  611 CONTINUE
      GO TO 3333
  651 DDAMP=1.0+DDAK(KK)
C  MULTIPLY DD CURRENT BY DDAMP FACTOR HERE
      OCUDD(I,J)=OCUDD(I,J)*DDAMP
      OCVDD(I,J)=OCVDD(I,J)*DDAMP
 3333 CONTINUE
C  MULTIPLY DD CURRENT BY DD FACTOR HERE
      OCUDD(I,J)=OCUDD(I,J)*DDFAC
      OCVDD(I,J)=OCVDD(I,J)*DDFAC
      IF (LS3(I,J).EQ.0) OCUDD(I,J)=0.0
      IF (LS3(I,J).EQ.0) OCVDD(I,J)=0.0
  407 CONTINUE
  406 CONTINUE
C
C  WRITE OUT U AND V OF GEOSTROPHIC CURRENT FOR FINE TUNING
      WRITE(6,951)
  951 FORMAT ('1 U -- COMPONENT OF GEOSTROPHIC CURRENT'//)
      CALL PRINT2(OCUDD,1.0)
      WRITE(6,952)
  952 FORMAT ('1 V -- COMPONENT OF GEOSTROPHIC CURRENT'//)
      CALL PRINT2(OCVDD,1.0)
C
C  OCEAN GEOSTROPHIC CURRENT VECTOR SPEED
      DO 944 I=1,NN2
      DO 945 J=1,MM2
      CSPEED(I,J)=SQRT((OCUDD(I,J)*OCUDD(I,J))+(OCVDD(I,J)*OCVDD(I,
  945 CONTINUE
  944 CONTINUE
C     IF (NOPR.EQ.1) GO TO 947
      WRITE(6,946) KYR,IMO,IDY
```

```
  946 FORMAT('1GEOSTROPHIC SPEED (CM/SEC)',3I3//)
      CALL PRINT2(CSPEED,1.0)
  947 CONTINUE
C
C
C  DESCRIBE METHOD OF OBTAINING ANGLE OF DEFLECTION FOR SURFACE WIN
C  DUE TO FRICTION AT SURFACE OF OCEAN (INTO CENTER OF LOW -- TO LE
C  AS A FUNCTION OF WIND SPEED (M/SEC) AND LATITUDE
C
      WRITE(6,543)
  543 FORMAT('1TABLE 3.--FRICTIONAL EFFECTS ON COMPUTED WINDS, REDU
     ' FACTOR AND ANGLE OF DEFLECTION')
      WRITE(6,547)
  547 FORMAT(11X,' OF SURFACE WIND TO THE LEFT OF THE PURE GEOSTROP
     'IND'/)
      WRITE(6,546)
      WRITE(6,544)
  546 FORMAT(44X,'WIND SPEED (M/SEC)')
  544 FORMAT(' LAT   REDUCTION     0     2     4     6     8    10
     '     14     16     18    20')
      GLAT1=67.0
C  DECREASE LATITUDE FROM 67N TO 20N BY 1 DEGREE INCREMENTS
      KK=0
      DO 540 I=1,48
      GLAT=GLAT1-I+1
      GLATR=GLAT/57.2958
      SINLAT=SIN(GLATR)
C  COMPUTE MANGITUDE REDUCTION (H U ROLL METHOD)
      IF(GLAT.GT.45.0) RED=0.75+(0.1*((90.0-GLAT)/45.0))
      IF(GLAT.LE.45.0.AND.GLAT.GE.25.0) RED=0.85
      IF(GLAT.LT.25.0) RED=0.65+(0.2*(GLAT/25.0))
C  COMPUTE DEFLECTION ANGLE AS A FUNCTION OF WIND SPEED
      JJ=0
      DO 541 J=1,21,2
      JJ=JJ+1
      COMP=J-1
      XCOMP=RED*COMP
      YCOMP=RED*COMP
      SPEED=SQRT((XCOMP*XCOMP)+(YCOMP*YCOMP))
      SPJ(JJ)=SPEED
      ANG=(22.5-(0.0175*SPEED*SPEED))*(1.495/(SINLAT+1.0))
      ANGDEF(JJ)=ANG
  541 CONTINUE
C     IF(KK.EQ.0)WRITE(6,545) (SPJ(JK),JK=1,11)
      WRITE(95,542)GLAT,RED,(ANGDEF(JJJ),JJJ=1,11)
      WRITE(6,542)GLAT,RED,(ANGDEF(JJJ),JJJ=1,11)
      KK=KK+1
  542 FORMAT(1X,F5.1,2X,F6.3,3X,12F6.1)
  545 FORMAT (16X,11I6/)
  540 CONTINUE
      CLOSE(95,DISP=CRUNCH)
C
C
C
C  DESCRIBE METHOD OF OBTAINING THE ROTATION ANGLE FOR
```

```
C  SURFACE OCEAN CURRENTS DUE TO WIND
C  ANGLE TO THE RIGTH OF THE WIND AS A FUNCTION OF WIND SPEED AND L
C  REFERENCE FNWC TECHN. NOTE #9, 1967
C  PAGE 10
C  DEFLECTION ANGLE AS A FUNCTION OF SPEED ONLY IS:
C        ANGLE=34(DEG)-7.5XSQRT V (M/SEC)
C    A CONSTRAINT HERE = THE ANGLE APPROACHES ZERO AT 40 KTS
C  PAGE 11
C  ANGLE AS FN OF LAT ONLY BUT FOR 6 KT (3.088 M/SEC) WIND ONLY IS:
C    GRAP DETERMINED A LINEAR RELATION BETWEEN 30-60 N (MODEL LAT R
C        ANGLE=(0.2933)(LATITUDE) + 2.4
C
C  BY MAKING INITIAL ANGLE OF DEFLECTION (34.0 IN ABOVE EQN)
C    A LINEAR FN OF LATITUDE WITH THE ABOVE SLOPE (0.2933)
C     CL=(34.0-((67.0-YLAT)*0.2933))
C    AND HOLDING THE CONSTRAINT THAT THE ANGLE APPROACHES ZERO AT 4
      SP40=40.0*0.5148
C    THE NEW COEFICIENT AS A FN OF LAT AT 40 KTS IS:
C      CLV=CL/SQRT(SP40)
C  THEREFORE:
C  THE ANGLE OF DEFLECTION AS A FN OF LAT AND WIND SPEED IS:
C        ASP(I,J)
C
      YLAT1=67.0
      DO 565 I=1,10
      AI=5.0*(I-1)
      SP=AI*0.5148
      DO 566 J=1,48
      YLAT=YLAT1-((J-1))
      CL=(34.0-((YLAT1-YLAT)*0.2933))
      CLV=CL/SQRT(SP40)
      ASP(I,J)=CL-(CLV*SQRT(SP))
  566 CONTINUE
  565 CONTINUE
      DO 567 I=1,10
      AKTS(I)=(I-1)*5.0
      AMPS(I)=AKTS(I)*0.5148
  567 CONTINUE
      WRITE (6,571)
  571 FORMAT('1TABLE 4.--LE OF DEFLECTION OF SURFACE CURRENT TO THE
     'T OF WIND')
      WRITE(6,572)
  572 FORMAT('            AS A FUNCTION OF LATITUDE AND WIND SPEED'/
      WRITE(6,573) (AKTS(I),I=1,9)
  573 FORMAT(' WIND SPEED',9I7,'   (KTS)')
      WRITE(6,574) (AMPS(I),I=1,9)
  574 FORMAT(12X,9F7.2,' (M/SEC)'/)
      WRITE(6,577)
  577 FORMAT(' LATITUDE                ANGLE OF DEFLECTION (DEGREES
      DO 568 J=1,48
      YLT=YLAT1-((J-1))
      WRITE(96,575) YLT,(ASP(I,J),I=1,9)
  568 CONTINUE
  575 FORMAT (F6.1,6X,9F7.2)
      WRITE(6,576)
```

```
  576 FORMAT (1H1//)
      CLOSE(96,DISP=CRUNCH)
C
C
C  TEMPORARY STOP
C     GO TO 901
C
C
C  DO 30 DAYS OF PLOTS
      DO 10000 IDY9=1,NDAYS
C
      IF (IDY9.GT.2) NOPR=1
      WRITE(6,6002) IDY9
 6002 FORMAT(' START PROCESSING DAY NUMBER ',I4)
C
   11 CONTINUE
C
C  READ IN SLP FOR LARGE SCALE FNOC GRID
      CALL READER(IMO,K,IDY,ITYP,SLP1)
C  999=EOF
C  CLOSE FILES FOR STORED RECORDS
      IF(ITYP.EQ.999)GO TO 909
      IF (IMO.LT.IMSTRT.AND.IYRSLP.EQ.K) GO TO 11
      IF (IMO.GT.IMEND) GO TO 909
      KYR=K
      YEAR=K+1900
      DAY=IDY
C
  107 FORMAT(13F9.2)
C
      IF(WFAC.LE.0.0) GO TO 6001
C  CONVERT DSLP UNITS BACK TO MILLIBARS FROM PACKED STORAGE UNITS
      DO 25 I=1,NN
      DO 26 J=1,MM
      SLP1(I,J)=(SLP1(I,J)/10.0)+1000.0
   26 CONTINUE
   25 CONTINUE
C
C
  226 CONTINUE
C
      IF(NOPR.EQ.1) GO TO 653
C  ROUGH PRINT SLP
      DO 652 I=1,20
      WRITE(6,107) (SLP1(I,J),J=1,44,4),SLP1(I,44)
  652 CONTINUE
C
  653 CONTINUE
C
C
C
C  BESSEL 4PT. INTERPOLATION FROM FNWC GRID TO CURRENTS 40X64 GRID
C  FIND THE I,J GRID COORDINATES OF THE CURRENTS MODEL GRID
C  RELATIVE TO THE FNWC GRID FOR USE IN BESSEL INTERPOLATION
C  NOTE ;  WE KNOW THE I,J'S OF THE CURRENTS GRID
```

```
C          THEY ARE 1/4 OF THE FNWC SUBSET
      NC=1
      IOK=99
      IOK=0
      DO 401 I=1,NN2
      DO 402 J=1,MM2
      XI1=ALON2(I,J)
      XI2=XI1
      XJ1=ALAT2(I,J)
      XJ2=XJ1
C     CALL CONVIJ (XI2,XJ2,IERR,XLEN,YYLEN,MM,NN,ALAT,ALON)
C
C  COMPUTE I,J'S OF 40X64 GRID RELATIVE TO 20X44 GRID
      XIXI=2.0+(FLOAT(I-1)*0.25)
      XJ=7.0+(FLOAT(J-1)*0.25)
C     XI=XI2
C     XJ=XJ2
C
C  COMPARE BESSEL AND SPLINE INTERPOLATIONS
C     IF(I.EQ.20)CALL BESSEL(XJ,XIXI,IOK,VV,SLP1)
      CALL SPLINE(XJ,XIXI,IOK,UU,SLP1)
C     IF(I.EQ.2) GO TO 901
C     CALL BESSEL(XI,XJ,IOK,VV,V2)
      T1(I,J)=UU
C     T2(I,J)=VV
      DUUVV=UU-VV
C     IF(I.EQ.20)WRITE(6,404)I,J,XI1,XJ1,XIXI,XJ,UU,VV,IERR,IOK,DUU
  402 CONTINUE
  401 CONTINUE
  404 FORMAT(2I5,6F10.4,2I5,'   SPLINE-BESSEL',F15.11)
C
      IF(NOPR.EQ.1) GO TO 709
C  PRINT 40X64 SLP FILE
      DO 667 K=1,7
      K2=16*K
      K1=K2-15
      IF(K2.GT.104) K2=104
      WRITE(6,664) K1,K2
      WRITE(6,671)
      DO 662 I=1,NN2
      WRITE(6,663) I,(T1(I,J),J=K1,K2)
  662 CONTINUE
  663 FORMAT(I3,18F7.1)
  664 FORMAT('1SLP (40X104) BY 16S',5X,'FROM J=',I2,2X,'TO J=',I3)
  671 FORMAT('   SPLINE' /)
  669 FORMAT ('   BESSEL' /)
  667 CONTINUE
C
C
C  SMOOTH BESSEL FUNCTION INTERPOLATED SLP FIELD
C     CALL SMTH(5,0.7,T1,T2)
C
C
C
C  PRINT SMOOTHED 40X64 SLP FILE
```

```
C     WRITE(6,664)
C     WRITE (6,669)
C     DO 772 I=1,NN2
C     WRITE(6,663) I,(T2(I,J),J=1,18)
  772 CONTINUE
C
C     WRITE(6,664)
C     WRITE (6,669)
C     DO 776 I=1,NN2
C     WRITE(6,663) I,(T2(I,J),J=19,36)
  776 CONTINUE
C
C     WRITE(6,664)
C     WRITE (6,669)
C     DO 777 I=1,NN2
C     WRITE(6,663) I,(T2(I,J),J=37,50)
  777 CONTINUE
  709 CONTINUE
C
C  STORE FILE ON DISK OR NOT
      IF(NOSTOR.LT.1)CALL OUTPT(T1,NFILE,KYR,IMO,IDY,HSLP)
      IF(NOSTOR.LT.1)WRITE(6,992) KYR,IMO,IDY
  992 FORMAT(' ',3I4,'  WRITTEN TO SLP OUTPUT FILE')
C
C
C
C
C  COMPUTE WIND VECTORS FROM SLP FIELD
      CALL WIND(T1,LS1,ALAT2,WXX,WYY)
C
      IF(NOSTOR.LT.1)CALL OUTPT(WXX,52,KYR,IMO,IDY,HWX)
      IF(NOSTOR.LT.1)CALL OUTPT(WYY,52,KYR,IMO,IDY,HWY)
C
C     IF (NOPR.EQ.1) GO TO 847
C  PRESSURE CHANGE PER GRID LENGTH MAGNITUDE
      DO 27 I=1,NN2
      DO 28 J=1,MM2
      IF((I+1).GE.NN2) GO TO 28
      IF((J+1).GE.MM2) GO TO 28
      TX=T1(I,J+1)-T1(I,J)
      TY=T1(I+1,J)-T1(I,J)
      CSPEED(I,J)=SQRT((TX*TX)+(TY*TY))
   28 CONTINUE
   27 CONTINUE
      IF(IDY9.LT.2)WRITE(6,29) KYR,IMO,IDY
   29 FORMAT('1ATMOS PRESS CHANGE X 10 (MAGNITUDE/GRID) (MB)',3I3//
      IF(IDY9.LT.2) CALL PRINT2(CSPEED,10.0)
C
C  WIND VECTOR SPEED
      DO 844 I=1,NN2
      DO 845 J=1,MM2
      CSPEED(I,J)=SQRT((WXX(I,J)*WXX(I,J))+(WYY(I,J)*WYY(I,J)))
  845 CONTINUE
  844 CONTINUE
C     IF (NOPR.EQ.1) GO TO 847
```

```
      IF(IDY9.LT.2)WRITE(6,846) KYR,IMO,IDY
  846 FORMAT('1WIND SPEED (M/SEC)',3I3//)
      IF(IDY9.LT.2) CALL PRINT2(CSPEED,1.0)
C
  847 CONTINUE
C
C  COMPUTE SURFACE OCEAN CURRENT VECTOR FIELD
C  SEE FUNCTION 'WINDC' (SFC CURR DUE TO WIND)
      DO 1844 I=1,NN2
      DO 1845 J=1,MM2
      OCUW(I,J)=WINDC(WXX(I,J),WYY(I,J),1,ALAT2(I,J))
      OCVW(I,J)=WINDC(WYY(I,J),WXX(I,J),2,ALAT2(I,J))
      CSPEED(I,J)=SQRT(OCUW(I,J)*OCUW(I,J)+OCVW(I,J)*OCVW(I,J))
 1845 CONTINUE
 1844 CONTINUE
C
C     IF(NOPR.EQ.1) GO TO 1849
      IF(IDY9.LT.2)WRITE(6,1848) KYR,IMO,IDY
 1848 FORMAT('1SPEED OF SFC-OCN-CURRENT DUE TO WIND (CM/SEC)',3I3//
      IF(IDY9.LT.2) CALL PRINT2(CSPEED,1.0)
      IF(NOPR.EQ.1) GO TO 1849
      WRITE(6,1846) KYR,IMO,IDY
 1846 FORMAT('1U--COMPONENT OF SURFACE OCEAN CURRENT DUE TO WIND (C
     ')',3I3//)
      CALL PRINT2(OCUW,1.0)
      WRITE(6,1847) KYR,IMO,IDY
 1847 FORMAT('1V--COMPONENT OF SURFACE OCEAN CURRENT DUE TO WIND (C
     ')',3I3//)
      CALL PRINT2(OCVW,1.0)
C
 1849 CONTINUE
C
C
C  STORE FILE 53 (SURFACE OCEAN CURRENTS DUE TO WIND)
C
      IF(NOSTOR.LT.1)CALL OUTPT(OCUW,53,KYR,IMO,IDY,HSFOU)
      IF(NOSTOR.LT.1)CALL OUTPT(OCVW,53,KYR,IMO,IDY,HSFOV)
      IF(NOSTOR.LT.1)WRITE(6,298)KYR,IMO,IDY
  298 FORMAT('/SFCOCN CURRENTS DUE TO WIND STORED FOR',3I4)
C
 6001 CONTINUE
C
C
C
C  ----------------------------------------------------------------
C  TUNING AND SUMMATION OF CURRENT COMPONENTS
C  ----------------------------------------------------------------
C
C  MULTIPLY SFC CURRENT DUE TO WIND BY FACTOR HERE
      DO 410 J=1,104
      DO 411 I=1,40
      OCUW(I,J)=OCUW(I,J)*WFAC
      OCVW(I,J)=OCVW(I,J)*WFAC
      OCUS(I,J)=OCUW(I,J)+OCUDD(I,J)
      OCVS(I,J)=OCVW(I,J)+OCVDD(I,J)
```

```
      IF (LS3(I,J).EQ.0) OCUS(I,J)=0.0
      IF (LS3(I,J).EQ.0) OCVS(I,J)=0.0
C
  411 CONTINUE
  410 CONTINUE
C
C
C
C  WRITE FILE OF TOTAL CURRENTS BEFORE ROTATION
      IF(NOSTOR.EQ.1) GO TO 227
      DO 94 J=1,104
      DO 94 I=1,5
      I2=I*8
      I1=I2-7
      WRITE(55,113,END=224)(OCUS(IK,J),IK=I1,I2),KYR,IMO,IDY
  94  CONTINUE
  224 DO 95 J=1,104
      DO 95 I=1,5
      I2=I*8
      I1=I2-7
      WRITE(55,113,END=225)(OCVS(IK,J),IK=I1,I2),KYR,IMO,IDY
  95  CONTINUE
  225 CONTINUE
  113 FORMAT(8F8.2,3I3,A3)
C
  227 CONTINUE
C
C
C
C  -------------------------------------------------------------
C               COASTAL BOUNDARY EFFECTS
C  ROTATION AND REDUCTION OF VECTORS IMPINGING UPON THE COASTLINE
C  -------------------------------------------------------------
C  RESULTS IN FILE 54
C
C
C
      CALL CSTROT(KYR,IMO,IDY,OCUS,OCVS,NOSTOF)
C
C     IF(NOPR.EQ.1) GO TO 2151
C
      DO 2144 I=1,NN2
      DO 2145 J=1,MM2
      CSPEED(I,J)=SQRT(OCUS(I,J)*OCUS(I,J)+OCVS(I,J)*OCVS(I,J))
 2145 CONTINUE
 2144 CONTINUE
      IF(IDY9.LT.2)WRITE(6,2148) KYR,IMO,IDY
      IF(IDY9.LT.2)WRITE(6,2149) WFAC,TFAC,DDFAC
 2148 FORMAT('1SPEED OF TOTAL SFC-OCN-CURRENT (CM/SEC)',3I3)
      IF(IDY9.LT.2) CALL PRINT2(CSPEED,1.0)
C
      IF(NOPR.EQ.1) GO TO 2151
      WRITE(6,2146) KYR,IMO,IDY
 2146 FORMAT('1U--COMPONENT OF TOTAL SURFACE OCEAN CURRENT (CM/SEC)
     ',3I3)
```

```
      WRITE(6,2149) WFAC,TFAC,DDFAC
 2149 FORMAT(10X,'WIND FACTOR =',F5.2,5X,'THERMAL FACTOR =',F5.2,5X
     'STROPHIC FACTOR =',F5.2/)
      CALL PRINT2(OCUS,1.0)
      WRITE(6,2147) KYR,IMO,IDY
      WRITE(6,2149) WFAC,TFAC,DDFAC
 2147 FORMAT('1V--COMPONENT OF TOTAL SURFACE OCEAN CURRENT (CM/SEC)
     ',3I3)
      CALL PRINT2(OCVS,1.0)
C
      WRITE(6,296)KYR,IMO,IDY,WFAC,TFAC,DDFAC
  296 FORMAT('/TOTAL SFCOCN CURRENTS (40X104) STORED FOR',3I4,5X,'W
     ',F4.2,5X,'TFAC=',F4.2,5X,'DDFAC=',F4.2)
C
 2151 CONTINUE
C
C
C
C ---------------------------------------------------------------
C        TRANSPORT (PROGRESSIVE) VECTORS
C ---------------------------------------------------------------
C
C
      DO 2964 KK=1,IEND
      I=IPV(KK)
      J=JPV(KK)
      IF(LS3(I,J).EQ.0)OCUS(I,J)=0.0
      IF(LS3(I,J).EQ.0)OCVS(I,J)=0.0
      U1=OCUS(I,J)
      V1=OCVS(I,J)
C
C  HANDLE OFF MAP POINTS
      IF(GV1(I,J).GE.38.99)GV1(I,J)=38.99
      IF(GV1(I,J).LE.1.0)GV1(I,J)=1.0
      IF(GU1(I,J).GE.102.99)GU1(I,J)=102.99
      IF(GU1(I,J).LE.1.0)GU1(I,J)=1.0
C
      CALL SPLINE(GU1(I,J),GV1(I,J),IOK,UU,OCUS)
      CALL SPLINE(GU1(I,J),GV1(I,J),IOK,VV,OCVS)
C
C  COMPUTE NEW I,J GRID COORDINATES OF END POINT OF NET 24 HR. MOTI
C  APPROXIMATE GRID LENGTH IS 90 KM
C  (CM/SEC)(1KM/100000CM)(3600SEC/HR)(24HRS)(1GRDLEN/90KM)=====
C  ===== 0.009600(GRDLEN)/(CM/SEC)
C     TYPICAL FACTOR=0.009600
C  THUS, IT TAKES A CURRENT SPEED OF 104 CM/SEC
C  TO GO ONE GRID POINT OF 90 KM IN A DAY
C  A MORE TYPICAL SPEED OF 15 CM/SEC YIELDS 0.144 GRID LENGTH
C  THEREFORE ABOUT 1 % OF A GRID POINT PER CM/SEC
C
C  COMPUTE FACTOR AT APPROPRIATE GRID LENGTH AT LATITUDE OF VECTOR
C  GRID LENGTH VARIES WITH LATITUDE -- 97.6KM AT 66N TO 76.5KM AT 3
      IGV=GV1(I,J)
      JGU=GU1(I,J)
      GLA=ALAT2(IGV,JGU)
```

```
      GLN=GRIDLN(GLA)
      FACTOR=0.864000/GLN
      GU2(I,J)=GU1(I,J)+(UU*FACTOR)
      GV2(I,J)=GV1(I,J)-(VV*FACTOR)
C
C  HANDLE OFF MAP POINTS
      IF(GV2(I,J).GE.38.99)GV2(I,J)=38.99
      IF(GV2(I,J).LE.1.0)GV2(I,J)=1.0
      IF(GU2(I,J).GE.102.99)GU2(I,J)=102.99
      IF(GU2(I,J).LE.1.0)GU2(I,J)=1.0
C
C     IF (I.EQ.30.AND.J.EQ.35)WRITE(6,3198)I,J,GU1(I,J),GV1(I,J),GU
C    '),GV2(I,J),U1,V1,UU,VV
 3198 FORMAT(2I4,8F15.4)
C     WRITE(6,4104)IDY9,KK,I,J,GV1(I,J),GU1(I,J),GV2(I,J),GU2(I,J)
      CALL IJTOLL(GV1(I,J),GU1(I,J),GLA1,GLO1)
      CALL IJTOLL(GV2(I,J),GU2(I,J),GLA2,GLO2)
C
  100 FORMAT(8F8.3,3I3,A3)
C  STORE PROGRESSIVE VECTOR FILES (60)
      WRITE(60,4104)IDY9,KK,KYR,IMO,IDY,I,J,GU1(I,J),GV1(I,J),GLA1,
     'GU2(I,J),GV2(I,J),GLA2,GLO2,WFAC,TFAC,DDFAC
 4104 FORMAT(2I3,3I2,2I3,8F7.3,3F3.1)
      GU1(I,J)=GU2(I,J)
      GV1(I,J)=GV2(I,J)
 2964 CONTINUE
C
C
10000 CONTINUE
  909 CONTINUE
C
      IF(NOSTOR.LT.1)CLOSE(NFILE,DISP=CRUNCH)
      IF(NOSTOR.LT.1)CLOSE(52,DISP=CRUNCH)
      IF(NOSTOR.LT.1)CLOSE(53,DISP=CRUNCH)
      IF(NOSTOF.LT.1)CLOSE(54,DISP=CRUNCH)
      IF(NOSTOR.LT.1)CLOSE(55,DISP=CRUNCH)
      CLOSE(60,DISP=CRUNCH)
C
C
  800 PRINT 30,IY1,LYR
   30 FORMAT(' YEARS NOT RIGHT.  FIRST=',I2,', LAST=',I2)
C
  901 CONTINUE
      END
      SUBROUTINE READER(IMO,IYR,IDY,ITYP,C)
      DIMENSION LS(40,104),A(40,104)
      DIMENSION C(40,104)
      NN=20;MM=44
      ITYP=0
  900 CONTINUE
      DO 100 J=1,44
      JJ=45-J
      READ(2,1100,END=800)IY,MM,NDY,(A(II,JJ),II=1,10)
      READ(2,1100,END=800)IY,MM,NDY,(A(II,JJ),II=11,20)
  100 CONTINUE
```

```
 1100 FORMAT(1X,3I2,8X,10F6.1)
C
      WRITE(6,1110) IY,MM,NDY
 1110 FORMAT (10X,3I3,' FNWC (20X44) FILE READ')
      IYE=IY ; MME=MM ; NDYE=NDY
      DO 210 I=1,20
      DO 210 J=1,44
      II=21-I
      C(II,J)=A(I,J)
  210 CONTINUE
      IMO=MM
      IYR=IY
      IDY=NDY
      GO TO 700
  800 PRINT 10,IYE,MME,NDYE
   10 FORMAT(" END OF FILE.  LAST RECORD FOUND FOR"3I5)
      ITYP=999
      RETURN
  600 ITYP=888
  700 RETURN

      END
      SUBROUTINE BESSEL (XJ,XI,IOK,VAL,DATA)
      DIMENSION A(4),AIJ(4)
      DIMENSION DATA(40,104)
C
      IF(IOK.NE.99) GO TO 300
      DO 310 I=1,40
      WRITE(6,305) (DATA(I,J),J=1,50,2)
  310 CONTINUE
  305 FORMAT(25F5.1)
  300 CONTINUE
C
      IOK=0
C  LIMIT CALCULATIONS WITHIN 1 INTERIOR POINT OF LEFT AND BOTTOM
C  LIMIT CALCULATIONS WITNIN 2 INTERIOR PTS OF RIGHT AND TOP
C  DO 4 POINT BELLEL INTERPOLATION FOR VALUES BETWEEN GRID POINTS
      IA=XI
      JA=XJ
C  TEST IF INDEX WITHIN RANGE
      IF (IA.LT.2.OR.JA.LT.2)   GO TO 998
      IF (IA.GT.18.OR.JA.GT.42)  GO TO 998
      XINC=XJ-FLOAT(JA)
      YINC=XI-FLOAT(IA)
      DO 105 K=1,4
      JJ=JA+(K-2)
      DO 106 L=1,4
      II=IA+(L-2)
      AIJ(L)=DATA(II,JJ)
      IF(XJ.LT.1.9)WRITE(6,450)XI,XJ,L,II,JJ,AIJ(L)
  106 CONTINUE
C  VERTICAL INTERPOLATION
      A(K)=AIJ(2)+YINC*((AIJ(3)-AIJ(2))+((YINC-1.0)/4.0)*((AIJ(4)-A
     1)+(AIJ(1)-AIJ(2))))
      IF(XJ.LT.1.9)WRITE(6,451)XI,XJ,K,A(K)
```

```
  105 CONTINUE
C  HORIZONTAL INTERPOLATION
      VAL=A(2)+XINC*((A(3)-A(2))+((XINC-1.0)/4.0)*((A(4)-A(3))+
     1(A(1)-A(2))))
      IF (XJ.LT.1.9)WRITE(6,452)XI,XJ,VAL
  450 FORMAT(2F7.3,3I4,F7.3)
  451 FORMAT(2F7.3,I4,F7.3)
  452 FORMAT(3F7.3)
      GO TO 999
  998 IOK=1
      VAL=999999
  999 RETURN
      END
      SUBROUTINE CONVIJ (XX,YY,IERR,XLEN,YYLEN,M,N,ALAT,ALON)
      DIMENSION ALAT(40,104),ALON(40,104)
      DIMENSION SLON(40,104)
      IF (XX.LT.0.0) XX=-XX
C
C
C  GIVEN A RANDOM POINT ON THE GLOBE (LONGITUDE,LATITUDE) DEGREE-DE
C  AND A GRID MESH OF DEFINED (LONG,LAT) POINTS
C  FIND THE (I,J) COORDINATES OF THE RANDOM POINT (XX,YY)
C  RELATIVE TO THE GIVEN GRID MESH
C  REJECT POINTS OUTSIDE OF THE GRID
C  THIS IS A GENERALIZED SUBPROGRAM TO FIT ANY NEARLY SQUARE
C  EQUIDISTANT GRID IN GLOBE (SPHERICAL) SPACE
C  BUT
C  GRID LINES MAY HAVE A CONSTANT SLOPE OR MAY VARY WITHIN GRID
C  IN LONGITUDE-LATITUDE SPACE
C  FOR THE GIVEN (NXM) GRID ----- N=ROW#, M=COL# -----
C  THIS RESULTS IN THE ABILITY TO MAP SQUARE OR ODD SHAPED GRIDS
C  INTO A SQUARE OR RECTANGULAR PLOT TO MATCH THE PRINTED PAGE
C  XLEN IS THE DESIRED LENGTH OF THE X-AXIS IN INCHES
C  YLEN IS THE DESIRED LENGTH OF THE Y-AXIS IN INCHES
C
C
C
C
      N1=N-1
      M1=M-1
      DXX=XLEN/M1
      DYY=YYLEN/N1
C
C  REJECT MOST POINTS OUT OF GRID EXTREMES
      IERR=0
C
C  FIND MAX AND MIN LAT AND LONG OF GRID
      AMAXLA=0.0
      AMAXLO=0.0
      AMINLA=1000.0
      AMINLO=1000.0
      DO 1201 I=1,N
      DO 1202 J=1,M
      IF (ALAT(I,J).GT.AMAXLA) AMAXLA=ALAT(I,J)
      IF (ALON(I,J).GT.AMAXLO) AMAXLO=ALON(I,J)
```

```
      IF (ALAT(I,J).LT.AMINLA) AMINLA=ALAT(I,J)
      IF (ALON(I,J).LT.AMINLO) AMINLO=ALON(I,J)
 1202 CONTINUE
 1201 CONTINUE
C
C
C  REJECT POINT IF OUT OF GRID MAX AND MIN LAT OR LONG
      IF (YY.GT.AMAXLA) IERR=1
      IF (YY.LT.AMINLA) IERR=1
      IF (XX.GT.AMAXLO) IERR=1
      IF (XX.LT.AMINLO) IERR=1
C  JUMP TO ERROR IF POINT IS OUTSIDE OF SQUARE OF BOUNDARY EXTREMES
C  AND SET BOTH COORDINATES TO MAXIMUM PLOT INCHES BEFORE RETURN
 1240 IF (IERR.EQ.1.AND.XX.GT.XLEN) XX=XLEN
      IF (IERR.EQ.1.AND.YY.GT.YYLEN) YY=YYLEN
      IF (IERR.EQ.1)  GO TO 9999
C
C
C  IN CASE THE GRID IS TILTED (ALL OR IN PART)
C  LOOK AROUND GRID BOUNDARY LINES TO REJECT EXTERIOR POINTS
C  COMPUTE EQUATION OF LINE BETWEEN 2 POINTS
C  IF XX IS BETWEEN THE POINTS TEST IF YY INSIDE OR OUTSIDE OF GRID
C  COVER THE CASES FOR EACH SIDE OF GRID SLOPING + OR - (LAT=HORIZ)
C
C  LOOK AT TOP AND BOTTOM LINES OF GRID
      I=1
 8002 DO 8001 J=1,M1
      X1=ALON(I,J)
      X2=ALON(I,J+1)
      Y1=ALAT(I,J)
      Y2=ALAT(I,J+1)
      IF (XX.GT.X1.OR.XX.LT.X2) GO TO 8001
      IF (X2.EQ.X1) GO TO 8001
      Y0=Y1+((Y2-Y1)*(XX-X1)/(X2-X1))
      IF (YY.GT.Y0.AND.I.EQ.1) IERR=1
      IF (YY.GT.Y0.AND.I.EQ.1) GO TO 1240
      IF (YY.LT.Y0.AND.I.EQ.N) IERR=1
      IF (YY.LT.Y0.AND.I.EQ.N) GO TO 1240
 8001 CONTINUE
      IF (I.EQ.N) GO TO 8025
      I=N
      GO TO 8002
 8025 CONTINUE
C
C  LOOK AT LEFT AND RIGHT BOUNDARY LINES
      J=1
 8007 DO 8006 I=1,N1
      X1=ALON(I,J)
      X2=ALON(I+1,J)
      Y1=ALAT(I,J)
      Y2=ALAT(I+1,J)
      IF (X2.EQ.X1) GO TO 8006
      IF ((X2-X1).LT.0.0) GO TO 8013
      IF (XX.LT.X1.OR.XX.GT.X2) GO TO 8006
      GO TO 8014
```

```
 8013 IF (XX.GT.X1.OR.XX.LT.X2) GO TO 8006
 8014 Y0=Y1+((Y2-Y1)*(XX-X1)/(X2-X1))
      IF (J.EQ.1.AND.(X2-X1).LT.0.0) GO TO 8011
      IF (J.EQ.M.AND.(X2-X1).GT.0.0) GO TO 8011
      IF (YY.GT.Y0) IERR=1
      IF (YY.GT.Y0) GO TO 1240
      GO TO 8006
 8011 CONTINUE
      IF (YY.LT.Y0) IERR=1
      IF (YY.LT.Y0) GO TO 1240
 8006 CONTINUE
      IF (J.EQ.M)GO TO 8035
      J=M
      GO TO 8007
 8035 CONTINUE
C
C  NOW, WE DEFINITELY KNOW (XX,YY) IS INSIDE THE GRID OR ON BOUNDAR
C
C  LOOK THROUGH ENTIRE GRID FOR CLOSEST GRID POINT TO POINT(XX,YY)
C  SEARCH USING LEAST SQUARE DISTANCE IN LINEAR 2D DEGREE SPACE
      DMIN=1000.0
      PI=3.141592
      XX=-XX
      DO 1207 I=1,N
      DO 1208 J=1,M
      SLON(I,J)=-ALON(I,J)
      X2=SLON(I,J)
      Y2=ALAT(I,J)
      DIST=SQRT(((X2-XX)*(X2-XX))+((Y2-YY)*(Y2-YY)))
      IF (DIST.LT.DMIN) NLAT=I
      IF (DIST.LT.DMIN) MLON=J
      IF (DIST.LT.DMIN) DMIN=DIST
 1208 CONTINUE
 1207 CONTINUE
C
C  NLAT,MLON ARE THE I,J COORDINATES FOR THE NEAREST GRID POINT
C
C  SET UP LAT AND LONG OF 4 SURROUNDING GRID POINTS
C
C
C        P2
C        .
C        .
C        .
C  P3...P0...P1
C        .
C        .
C        .
C        P4
C
C
C
C  COMPUTATIONS ARE NOW DONE IN THIS 4-QUADRANT SYSTEM
C  ON BOUNDARY AVOID POINTS OUTSIDE OF GRID (SET THEM = CLOSEST GRI
      C=1000.0
```

```
      P0X=SLON(NLAT,MLON)
      P0Y=ALAT(NLAT,MLON)
      P1X=P0X; P2X=P0X; P3X=P0X; P4X=P0X
      P1Y=P0Y; P2Y=P0Y; P3Y=P0Y; P4Y=P0Y
      IF (MLON.LT.M) P1X=SLON(NLAT,MLON+1)
      IF (MLON.LT.M) P1Y=ALAT(NLAT,MLON+1)
      IF (NLAT.GT.1) P2X=SLON(NLAT-1,MLON)
      IF (NLAT.GT.1) P2Y=ALAT(NLAT-1,MLON)
      IF (MLON.GT.1) P3X=SLON(NLAT,MLON-1)
      IF (MLON.GT.1) P3Y=ALAT(NLAT,MLON-1)
      IF (NLAT.LT.N) P4X=SLON(NLAT+1,MLON)
      IF (NLAT.LT.N) P4Y=ALAT(NLAT+1,MLON)
C  SPHERICAL GEOMETRY OF THE GLOBE
C    DICTATES THAT CONVERGING LONGITUDES MUST BE
C      MULTIPLIED BY COS(LAT) TO PRESERVE EQUIVALENT COORDINATE SCA
      XC=COS(ALAT(NLAT,MLON)*PI/180.0)
      DX0=(XX-P0X)*XC
      DX1=(P1X-P0X)*XC
      DX2=(P2X-P0X)*XC
      DX3=(P3X-P0X)*XC
      DX4=(P4X-P0X)*XC
      DY0=YY-P0Y
      DY1=P1Y-P0Y
      DY2=P2Y-P0Y
      DY3=P3Y-P0Y
      DY4=P4Y-P0Y
      S0=DX0*DY0
      S1=DX1*DY1
      S2=DX2*DY2
      S3=DX3*DY3
      S4=DX4*DY4
C
C  COMPUTE ANGLE (AP) BETWEEN LINE (XX,YY)-(MLON,NLAT) AND THE +X A
C  AND AVOID INFINITE TANGENTS
      AP=0.0; AE=0.0; AN=0.0; AW=0.0; AS=0.0
      IF (ABS(DX0).LE.(1.0/C)) AP=ATAN(SIGN(C,S0))
      IF (ABS(DX0).GT.(1.0/C)) AP=ATAN((DY0)/DX0)
      CALL RADCOR(DX0,DY0,ADD)
      AP=AP+ADD
C
C  COMPUTE THE ANGLE EACH OF THE 4 COORDINATES MAKES WITH THE +X AX
C  AE IS THE ANGLE THE EAST LINE (P1-P0) MAKES WITH THE +X AXIS
      IF (ABS(DX1).LE.(1.0/C)) AE=ATAN(SIGN(C,S1))
      IF (ABS(DX1).GT.(1.0/C)) AE=ATAN((DY1)/DX1)
      CALL RADCOR(DX1,DY1,ADD)
      AE=AE+ADD
C  AN IS THE ANGLE THE NORTH LINE (P2-P0) MAKES WITH THE +X AXIS
      IF (ABS(DX2).LE.(1.0/C)) AN=ATAN(SIGN(C,S2))
      IF (ABS(DX2).GT.(1.0/C)) AN=ATAN((DY2)/DX2)
      CALL RADCOR(DX2,DY2,ADD)
      AN=AN+ADD
C  AW IS THE ANGLE THE WEST LINE (P3-P0) MAKES WITH THE +X AXIS
      IF (ABS(DX3).LE.(1.0/C)) AW=ATAN(SIGN(C,S3))
      IF (ABS(DX3).GT.(1.0/C)) AW=ATAN((DY3)/DX3)
      CALL RADCOR(DX3,DY3,ADD)
```

```
      AW=AW+ADD
C  AS IS THE ANGLE THE SOUTH LINE (P4-P0) MAKES WITH THE +X AXIS
      IF (ABS(DX4).LE.(1.0/C)) AS=ATAN(SIGN(C,S4))
      IF (ABS(DX4).GT.(1.0/C)) AS=ATAN((DY4)/DX4)
      CALL RADCOR(DX4,DY4,ADD)
      AS=AS+ADD
C
C  COMPUTE LENGTH TO POINT(XX,YY)
      D0=SQRT((DX0*DX0)+(DY0*DY0))
      D1=SQRT((DX1*DX1)+(DY1*DY1))
      D2=SQRT((DX2*DX2)+(DY2*DY2))
      D3=SQRT((DX3*DX3)+(DY3*DY3))
      D4=SQRT((DX4*DX4)+(DY4*DY4))
C
C  COMPUTE (ANGLE) THE ANGLE FROM +X AXIS OF ROTATED GRID
C  AE IS THE BASE LINE USED FOR COMPUTING + ANGLES BECAUSE
C  IT CORRESPONDS TO THE +X AXIS OF THE ROTATED GRIDS
C  WARNING:  THE TRIGONOMETRY ASSUMES RIGHT ANGLES WITHIN THE GRID-
C              BUT TESTS INDICATE THIS ROUTINE PERFORMES WELL
C             FOR RELATIVELY SMALL DEVIATIONS (RT. ANGLE +- 10%)
      IF (MLON.EQ.M.AND.AW.LT.PI) AE=AW+PI
      IF (AP.LT.AE) ANGLE=(2.0*PI)+AP-AE
      IF (AP.GE.AE) ANGLE=AP-AE
C
C  COMPUTE FRACTIONAL I,J'S AWAY FROM GRID POINT
      IF (D0.EQ.0.0) XFRACT=0.0
      IF (D0.EQ.0.0) YFRACT=0.0
      IF (D0.EQ.0.0) GO TO 4198
      DD=D1
      IF (D1.EQ.0.0) DD=D3
      IF (DD.EQ.0.0)  XFRACT=0.0
      IF (DD.EQ.0.0)  GO TO 603
      XFRACT=(D0/DD)*COS(ANGLE)
  603 DD=D2
      IF (D2.EQ.0.0) DD=D4
      IF (DD.EQ.0.0)  YFRACT=0.0
      IF (DD.EQ.0.0)  GO TO 4198
      YFRACT=-(D0/DD)*SIN(ANGLE)
 4198 XXX=XFRACT + FLOAT(MLON)
 4199 YYY=YFRACT + FLOAT(NLAT)
      IF (XXX.LT.1.0) XXX=1.0
      IF (YYY.LT.1.0) YYY=1.0
      IF (XXX.GT.M) XXX=M
      IF (YYY.GT.N) YYY=N
C
C  CONVERT FROM I,J GRID VALUES TO I,J PLOT INCHES
C     XX=XLEN*((XXX-1.0)/M1)
C     YY=YYLEN-(YYLEN*((YYY-1.0)/N1))
      XX=XXX
      YY=YYY
C
 9999 CONTINUE
      RETURN
C
C
```

```
C
      END
      SUBROUTINE SMTH(N,ALPHA,FL,FL2)
      DIMENSION TDEP(40,104),TML(40,104),FL(40,104),XF(40,104),
     1TCX(40,104),TCY(40,104),XTR2(40,104),XTR(40,104),TS(40,104),
     2SLP(40,104),WCX(40,104),WCY(40,104),LS2(40,104)
      DIMENSION FL2(40,104)
      BETA=1.0-ALPHA
      DO 2 I=1,40
      DO 2 J=1,104
    2 XF(I,J)=FL(I,J)
      DO 40 M=1,N
 10   DO 20 I=2,39
      DO 20 J=2,103
      XY=FL(I,J)
      A=FL(I,J+1)
      B=FL(I+1,J)
      C=FL(I,J-1)
      D=FL(I-1,J)
      IF (A.EQ.0.OR.B.EQ.0)     GO TO 20
      IF (C.EQ.0.OR.D.EQ.0)     GO TO 20
      XF(I,J) = (ALPHA*XY) + (BETA*((A+B+C+D)/4.0))
 20   CONTINUE
      DO 30 I=2,39
      DO 30 J=2,49
      FL(I,J)=XF(I,J)
 30   CONTINUE
 40   CONTINUE
      RETURN
      END
      SUBROUTINE SPLINE (X,Y,IOK,VAL,DATA)
      DIMENSION A(4),AIJ(4)
      DIMENSION DATA(40,104)
C
C
      IOK=0
C  DO 4 PT NATURAL SPLINE INTERPOLATION FOR VALUES BETWEEN GRID PTS
      IA=Y
      JA=X
C  LIMIT CALCULATIONS WITHIN 1 INTERIOR POINT OF LEFT AND BOTTOM
C  LIMIT CALCULATIONS WITNIN 2 INTERIOR PTS OF RIGHT AND TOP
      IF (IA.LT.2.OR.JA.LT.2)   GO TO 998
C     IF (IA.GT.18.OR.JA.GT.42)  GO TO 998
      IF (IA.GT.38.OR.JA.GT.102)  GO TO 998
      DX=1.0
      X1=JA
      Y1=IA
      XP1=X1+1.0
      YP1=Y1+1.0
      DO 105 K=1,4
      JJ=JA+(K-2)
      DO 106 L=1,4
      II=IA+(L-2)
      AIJ(L)=DATA(II,JJ)
C     IF(XJ.LT.1.9)WRITE(6,450)XI,XJ,L,II,JJ,AIJ(L)
```

```
  106 CONTINUE
C  VERTICAL INTERPOLATION
      GPP=(AIJ(3)-(2.0*AIJ(2))+AIJ(1))/(DX*DX)
      GPPP1=(AIJ(4)-(2.0*AIJ(3))+AIJ(2))/(DX*DX)
      G1=((GPP/6.0)*((((YP1-Y)**(3.0))/DX)-(DX*(YP1-Y))))
      G2=((GPPP1/6.0)*((((Y-Y1)**(3.0))/DX)-(DX*(Y-Y1))))
      G3=(AIJ(2)*((YP1-Y)/DX))
      G4=(AIJ(3)*((Y-Y1)/DX))
      G=G1+G2+G3+G4
C     IF(G.LT.20.0)WRITE (66,9412) N,M,K,NK,GPP,GPPP1,G1,G2,G3,G4,G
      A(K)=G
  105 CONTINUE
C  HORIZONTAL INTERPOLATION
      GPP=(A(3)-(2.0*A(2))+A(1))/(DX*DX)
      GPPP1=(A(4)-(2.0*A(3))+A(2))/(DX*DX)
      G1=((GPP/6.0)*((((XP1-X)**(3.0))/DX)-(DX*(XP1-X))))
      G2=((GPPP1/6.0)*((((X-X1)**(3.0))/DX)-(DX*(X-X1))))
      G3=(A(2)*((XP1-X)/DX))
      G4=(A(3)*((X-X1)/DX))
      G=G1+G2+G3+G4
      VAL=G
  450 FORMAT(2F7.3,3I4,F7.3)
  451 FORMAT(2F7.3,I4,F7.3)
  452 FORMAT(3F7.3)
      GO TO 999
  998 IOK=1
      VAL=999999
  999 RETURN
      END
      SUBROUTINE PRINT1(P,NOX)
      DIMENSION P(40,104),IQ(64),PQ(40,104)
  199 FORMAT(1H1,5X,'LAND-SEA-GRID')
 200  FORMAT(1H1,5X,'TEMPERATURE AT 200 M(CX10)')
 201  FORMAT(1H1,5X,'DYNAMIC TOPOGRAPHY (DYNMX100)')
 202  FORMAT(1H1,5X,'SURFACE PRESSURE(MBX10)'//)
 203  FORMAT(1H1,5X,'SURFACE TEMPERATURE(CX10)')
  204 FORMAT(1H1,5X,'MODEL OUTPUT (CM/SEC) ')
  205 FORMAT(1H1,5X,'WEIGHTED AVERAGE OF TEMP 0-200M (C) X10.0')
  206 FORMAT (1H1,5X,'SURFACE WIND SPEED (M/SEC)'//)
  209 FORMAT(1H1,5X,'U -- COMPONENT OF WIND SPEED (M/SEC)'//)
  210 FORMAT(1H1,5X,'V -- COMPONENT OF WIND SPEED (M/SEC)'//)
  211 FORMAT(1H1,5X,'SURFACE OCEAN CURRENT SPEED (CM/SEC)'//)
  220 FORMAT (I3,32I4)
  212 FORMAT(1H1,5X,'U--DYNAMIC OCEAN CURRENT (CM/SEC)'//)
  213 FORMAT(1H1,5X,'V--DYNAMIC OCEAN CURRENT (CM/SEC)'//)
      KR=6
C     DO 500 IP=1,2
      GO TO (10,20,30,40,42,44,70,80,90,100,110,120,130)NOX
 10   WRITE(KR,200)
      DO 11 I=1,40     ; DO 11 J=1,104
   11 PQ(I,J)=P(I,J)*10.0
      GO TO 50
 20   WRITE(KR,201)
      DO 21 I=1,40     ;  DO 21 J=1,104
   21 PQ(I,J)=P(I,J)*100.0
```

```
      GO TO 50
30    WRITE(KR,202)
      DO 31 I=1,40     ;  DO 31 J=1,104
   31 PQ(I,J)=(P(I,J)-1000)*10.0
      GO TO 50
40    WRITE(KR,203)
      DO 41 I=1,40     ;  DO 41 J=1,104
   41 PQ(I,J)=P(I,J)*10.0
      GO TO 50
   42 WRITE(KR,204)
      DO 43 I=1,40
      DO 43 J=1,104
   43 PQ(I,J)=P(I,J)
      GO TO 50
   44 WRITE(KR,205)
      DO 45 I=1,40
      DO 45 J=1,104
   45 PQ(I,J)=(P(I,J))*10.0
      GO TO 50
   70 WRITE (KR,199)
      DO 71 I=1,40
      DO 71 J=1,104
   71 PQ(I,J)=P(I,J)
      GO TO 50
   80 WRITE (KR,206)
      DO 81 I=1,40
      DO 81 J=1,104
   81 PQ(I,J)=P(I,J)
      GO TO 50
   90 WRITE(KR,209)
      DO 91 I=1,40
      DO 91 J=1,104
   91 PQ(I,J)=P(I,J)
      GO TO 50
  100 WRITE(KR,210)
      DO 101 I=1,40
      DO 101 J=1,104
  101 PQ(I,J)=P(I,J)
      GO TO 50
  110 WRITE(KR,211)
      DO 111 I=1,40
      DO 111 J=1,104
  111 PQ(I,J)=P(I,J)
      GO TO 50
  120 WRITE (KR,212)
      DO 121 I=1,40
      DO 121 J=1,104
  121 PQ(I,J)=P(I,J)
      GO TO 50
  130 WRITE (KR,213)
      DO 131 I=1,40
      DO 131 J=1,104
  131 PQ(I,J)=P(I,J)
C
C
```

```
   50 CONTINUE
C
C  PRINT LEFT 32 COLS. THEN PRINT RIGHT 32 COLS.
      DO 58 KJ=1,2
      J2=KJ*32
      J1=J2-31
      IF(NOX.EQ.1.AND.KJ.EQ.2) WRITE(KR,200)
      IF(NOX.EQ.2.AND.KJ.EQ.2) WRITE(KR,201)
      IF(NOX.EQ.6.AND.KJ.EQ.2) WRITE(KR,205)
      IF(NOX.EQ.4.AND.KJ.EQ.2) WRITE(KR,203)
      IF(NOX.EQ.3.AND.KJ.EQ.2) WRITE(KR,202)
      IF(NOX.EQ.8.AND.KJ.EQ.2) WRITE(KR,206)
      IF(NOX.EQ.9.AND.KJ.EQ.2) WRITE(KR,209)
      IF(NOX.EQ.10.AND.KJ.EQ.2) WRITE(KR,210)
      IF(NOX.EQ.11.AND.KJ.EQ.2) WRITE(KR,211)
      IF(NOX.EQ.12.AND.KJ.EQ.2) WRITE(KR,212)
      IF(NOX.EQ.13.AND.KJ.EQ.2) WRITE(KR,213)
      DO 60 I=1,40
      DO 57 JJ=J1,J2
   57 IQ(JJ)=PQ(I,JJ)
      WRITE (KR,220)  I,(IQ(J),J=J1,J2)
   60 CONTINUE
   58 CONTINUE
  500 CONTINUE
      RETURN
      END
      SUBROUTINE OUTPT(OUT,NFILE,KYR,MO,KDY,H)
      DIMENSION OUT(40,104),H(1)
      DO 60 J=1,104
      DO 60 I=1,5
      I2=I*8
      I1=I2-7
      WRITE (NFILE,100)  (OUT(IK,J),IK=I1,I2),KYR,MO,KDY,H
   60 CONTINUE
  100 FORMAT (8F8.2,3I3,A3)
      RETURN
      END
      SUBROUTINE WIND(SLP5,LS2,BLAT,WX,WY)
      DIMENSION SLP5(40,104),BLAT(40,104),WX(40,104),WY(40,104)
      DIMENSION LS2(40,104)
 100  FORMAT(1X,18F8.3)
      ALPHA=0.75 ; BETA=1.0
      OMEGAV=0.80
C  COMPUTE SURFACE GEOSTROPHIC WIND IN  M/SEC
C  WIND(M./SEC) = (DP/DL)/(RHO)(CORIOLIS)
C  1 BAR = 10(6TH) DYNES/CM**2
C  WEIGHT IN KG = MG = (KG)(9.8M/SEC**2)
C 1 MBAR = 10.1971 KG/M**2 = 100.0 KG M-1 SEC-2
C  CW = (100)/(90000)(1.22)(10-4TH) = 9.11
C  1XGRIDLENGTH IS APPROX. 90000 M
C  TYPICAL WIND SPEED COMPUTATION IS FOR 1MB GRADIENT PER GRID LENG
C  W(M/S)=(1 MB)(100 KG M-1 SEC-2 MB-1)/(90000 M)(1.22 KG M-3)(0.00
C                                         58 S-1)(SIN LAT)
C     (1)(100)/(90000)(1.22)(0.0001458)(.707)  =  8.84 M/S
C
```

```
C           INCLUDES REDUCTION AND ROTATION OF WIND
C
      PI=3.1415962
      THETA=-15.0*PI/180.0    ; RR=0.7    ; A1=RR*COS(THETA)
      B2=A1    ; B1=RR*SIN(THETA)     ; A2=-B1
      DO 20 I=2,39
      DO 20 J=2,103
C  COMPUTE PURE GEOSTROPHIC WIND (WITH NO FRICTION)
      GLAT=BLAT(I,J)
      GG=GRIDLN(GLAT)
      XR=SIN(BLAT(I,J)/57.296)
      CRLS=0.0001458*XR
C  KM TO METERS
      GG=GG*1000.0
      CW=100.0/(GG*1.22*CRLS)
C     IF(LS2(I,J).EQ.0.OR.LS2(I+1,J).EQ.0.OR.LS2(I,J+1).EQ.0)  GO T
C     IF (LS2(I,J).EQ.0) GO TO 21
C  USE 5-POINT GRADIENT IN INTERIOR OF GRID
C  USE 3-POINT GRADIENT ONE GRID POINT FROM BOUNDARY
      GRADX=SLP5(I,J+1)-SLP5(I,J-1)
      GRADX=GRADX/2.0
      IF (J.EQ.2.OR.J.EQ.103) GO TO 41
      GRADX=SLP5(I,J-2)-(8.0*SLP5(I,J-1))+(8.0*SLP5(I,J+1))-SLP5(I,
      GRADX=GRADX/12.0
   41 CONTINUE
      GRADY=SLP5(I-1,J)-SLP5(I+1,J)
      GRADY=GRADY/2.0
      IF (I.EQ.2.OR.I.EQ.39) GO TO 42
      GRADY=SLP5(I-2,J)-(8.0*SLP5(I-1,J))+(8.0*SLP5(I+1,J))-SLP5(I+
      GRADY=-GRADY/12.0
   42 CONTINUE
      WU=-GRADY*CW
      WV=GRADX*CW
      GLATR=GLAT/57.2958
      SINLAT=SIN(GLATR)
C  COMPUTE MANGITUDE REDUCTION (H U ROLL METHOD)
      IF(GLAT.GT.45.0) RED=0.75+(0.1*((90.0-GLAT)/45.0))
      IF(GLAT.LE.45.0.AND.GLAT.GE.25.0) RED=0.85
      IF(GLAT.LT.25.0) RED=0.65+(0.2*(GLAT/25.0))
C  COMPUTE DEFLECTION ANGLE AS A FUNCTION OF WIND SPEED
C  USE REDUCED COMPONENTS IN FIGURING THE ANGLE OF DEFLECTION
      XCOMP=RED*WU
      YCOMP=RED*WV
      SPEED=SQRT((XCOMP*XCOMP)+(YCOMP*YCOMP))
      ANG=(22.5-(0.0175*SPEED*SPEED))*(1.495/(SINLAT+1.0))
      THETA=-ANG*PI/180.0
      A1=RED*COS(THETA)
      B2=A1
      B1=RED*SIN(THETA)
      A2=-B1
      WX(I,J)=WU*A1 + WV*B1
      WY(I,J)=WV*B2 + WU*A2
      GO TO 20
 21   WX(I,J)=0.0 ; WY(I,J)=0.0
 20   CONTINUE
```

```
      DO 22 M=1,40
      WX(M,104)=WX(M,103)
      WY(M,104)=WY(M,103)
      WX(M,1)=WX(M,2)
      WY(M,1)=WY(M,2)
   22 CONTINUE
      DO 24 N=1,104
      WX(40,N)=WX(39,N)
      WY(40,N)=WY(39,N)
      WX(1,N)=WX(2,N)
      WY(1,N)=WY(2,N)
   24 CONTINUE
C  WRITE WIND SPEED
C     CALL PRINT1(SLP5,3)
C     CALL PRINT1(WX,9)
C     CALL PRINT1(WY,10)
  395 CONTINUE
      RETURN
      END
      FUNCTION WINDC(W1,W2,M,YLAT)
C
C     CURRENT (CM/SEC) = 4.8X(SQUARE ROOT WIND SPEED M/SEC)
C
C
C  FACTOR IS BETWEEN 4 AND 5 IF MASS TRANSPORT BY WAVES IS INCLUDED
      XK=4.8
C
C
      SPDSQD=W1**2 + W2**2
      IF(SPDSQD-5.0E-04)10,10,20
   10 WINDC=0.0
      RETURN
   20 SPEED=SQRT(SPDSQD)
      SQROOT=SQRT(SPEED)
      SP40=40.0*0.5148
      CL=(34.0-((67.0-YLAT)*0.2933))
      CLV=CL/SQRT(SP40)
      ANGLD=CL-(CLV*SQRT(SPEED))
      B1=COS(ANGLD/57.296)
      B2=SIN(ANGLD/57.296)
      W10=XK*(W1/SQROOT)
      W20=XK*(W2/SQROOT)
      IF(M.EQ.2) W20=-W20
      WINDC=W10*B1 + W20*B2
C     IF (M.EQ.5) WRITE(6,88)W1,W2,A2,W10,W20,D2,B1,B2,WINDC
   88 FORMAT (1H ,9F12.5)
      RETURN
      END
      SUBROUTINE OCEANC(TML,TCX,TCY,HH)
      COMMON/LAT/ALAT2,LS2
      DIMENSION TDEP(40,104),TML(40,104),FL(40,104),XF(40,104),
     1TCX(40,104),TCY(40,104),XTR2(40,104),XTR(40,104),TS(40,104),
     2SLP(40,104),WCX(40,104),WCY(40,104),LS2(40,104)
      DIMENSION HH(1),ALAT2(40,104)
      GAMMA=1.025 ; T=0.0
```

```
      DO 70 I=1,39
      DO 70 J=1,103
C  GRIDSIZE VARIES WITH LATITUDE 83-96 KM (SEE GRIDLN FUNCTION)
      XLEN=GRIDLN(ALAT2(I,J))
C  CHANGE KM TO M
      XLEN=XLEN*1000.0
C
      XOR=SIN(ALAT2(I,J)/57.296)
      COR=0.0001458*XOR
      NP1=LS2(I,J) ; NP2=LS2(I,J+1) ; NP3=LS2(I+1,J)
C  ZERO VALUE IF ANY OF THE LAND-SEA GRID POINTS ARE ZERO
      IF (NP1.LT.2) GO TO 65
      DDI=TML(I,J+1)-TML(I,J)
      DDJ=TML(I+1,J)-TML(I,J)
 19   IF(HH(1).EQ.3HDYC) GO TO 20
      GO TO 10
 65   TCX(I,J)=0.0 ; TCY(I,J)=0.0 ; GO TO 70
   10 CONTINUE
  550 FORMAT (1H ,2I5,7F15.7)
      TCX(I,J)=THERM(DDI,DDJ,XLEN,COR,TML(I,J),T)
      TCY(I,J)=T
C     WRITE (6,550) I,J,TML(I,J),DDI,DDJ,COR,XLEN,TCX(I,J),TCY(I,J)
      GO TO 70
 20   TCX(I,J)=DYCUR(DDI,DDJ,XLEN,COR,T)
      TCY(I,J)=T
 70   CONTINUE
      DO 75 M=1,40
      TCX(M,104)=TCX(M,103)
      TCY(M,104)=TCY(M,103)
 75   CONTINUE
      DO 77 M=1,104
      TCX(40,M)=TCX(39,M)
      TCY(40,M)=TCY(39,M)
 77   CONTINUE
      RETURN
      END
      FUNCTION THERM(DDI,DDJ,XLN,COR,TML,T)
C
C  THWIND = DELTAT(G)(DZ)/(F)(TBAR)
C
C  TYPICAL THERMAL WIND CALCULATION
C  C1 = G(DZ)=(9.8M/SEC**2)(200M)=1961.2 M**2/SEC**2
C  C2 = (0.0001458)(10+273)(SIN 45)(82000) = 2392.45 M SEC-1 DEGK
C  ONE DEGREE GRADIENT GIVES TYPICAL VALUE OF CURRENT = 0.8197 M/SE
C
C
C       TIMES 0.0544 = 4.46 CM/SEC
C
      ALPHA=100.0
      C1=1961.2
      C2=TML*COR*XLN
      T=DDI*C1/C2
      THERM=DDJ*C1/C2
      T=T*ALPHA    ; THERM=THERM*ALPHA
C
```

```
C  BEST FIT CORRELATION
C  CONVERTS TO DYNAMIC HEIGHT CALCULATION WITH SALINITY CONSTANT
      T=T*0.0544
      THERM=THERM*0.0544
C
      RETURN
      END
      FUNCTION DYCUR(DDI,DDJ,XLN,COR,T)
C TYPICAL CALCULATION
C     10(0.1M**2SEC**-2) DIVIDED BY (0.0001458SEC**-1)(SIN45)(16400
C     = 0.0873 M/SEC
C  ALPHA CONVERTS TO 8.73 CM/SEC
      ALPHA=100.0
      C2=XLN*COR
      T=+ALPHA*10.0*DDI/C2
      DYCUR=+ALPHA*10.0*DDJ/C2
      RETURN
      END
      FUNCTION GRIDLN(GLAT)
C  GRID LENGTH AS A FUNCTION OF GEOGRAPHIC LATITUDE FOR 40X104 GRID
      AKM=(1.0+(SIN(GLAT*3.14159/180.0)))/1.86603
      GRIDLN=AKM*95.23
      RETURN
      END
      FUNCTION GDFNOC(GLAT)
C  GRID LENGTH AS A FUNCTION OF GEOGRAPHIC LATITUDE FOR FNOC 20X44
      AKM=(1.0+(SIN(GLAT*3.14159/180.0)))/1.86603
      GDFNOC=AKM*380.8
      RETURN
      END
      SUBROUTINE DELTAD
      COMMON/LAT/ALAT2,LS2
      DIMENSION TDEP(40,104),TML(40,104),FL(40,104),XF(40,104),
     1TCX(40,104),TCY(40,104),XTR2(40,104),XTR(40,104),TS(40,104),
     2SLP(40,104),WCX(40,104),WCY(40,104),IGD(40,104)
      DIMENSION HUDD(1),HVDD(1)
      DATA HUDD/'UDD   '/
      DATA HVDD/'VDD   '/
C*****INPUT THE DYNAMIC TOPOGRAPHY*****
C  LTM
      KYR=99
      MO=99
      KDY=99
C  READ IN WESTERN PART OF NORTH PACIFIC GRID
      DO 10 J=1,64
      DO 10 I=1,5
      I2=I*8 ; I1=I2-7
      READ(34,111) (TML(IK,J),IK=I1,I2),KYR,MO,KDY,H
  111 FORMAT (8F8.3,3I3,A3)
 10   CONTINUE
C  READ IN EASTERN PART OF NORTH PACIFIC GRID
      DO 12 J=41,104
      DO 12 I=1,5
      I2=I*8 ; I1=I2-7
      READ(33,111) (TML(IK,J),IK=I1,I2),KYR,MO,KDY,H
```

```
 12   CONTINUE
C
C
C  ADD MISSING S.W. CORNER TO MAP OF DD
      DO 20 I=27,40
      B=I-26
      FI=B/13.0
      DO 21 J=1,16
      C=16-J
      FJ=C/15.0
      IF (TML(I,J).LT.0.01) TML(I,J)=2.7+((FI+FJ)/10.0)
   21 CONTINUE
   20 CONTINUE
C
C  PRINT DYNAMIC HEIGHTS
      WRITE(6,305)
  305 FORMAT('1 DYNAMIC HEIGHTS'//)
      CALL PRINT2(TML,10.0)
C
C
C
C  PRINT DECIMAL DYN HTS FOR ALASKAN STREAM TUNING
      WRITE (6,90)
   90 FORMAT ('1DYNAMIC HEIGHTS'//)
      DO 91 J=1,104
      J2=105-J
      WRITE (6,92) J,J2,(TML(I,J2),I=1,20)
   91 CONTINUE
   92 FORMAT (2I4,20F6.3)
C
C
C     CALL PRINT1(TML,2)
C*****
C     CALL SMTH(2,0.98,TML)
C     CALL PRINT1(TML,2)
      CALL OCEANC(TML,TCX,TCY,3HDYC)
C     CALL PRINT1(TCX,12)
C     CALL PRINT1(TCY,13)
C     CALL SMTH(3,0.88,TCY)
C     CALL SMTH(3,0.88,TCX)
C     CALL PRINT1(TCX,5)
C     CALL PRINT1(TCY,5)
C
C  PRINT GEOSTROPHIC CURRENTS
      WRITE(6,300)
  300 FORMAT('1 U--COMPONENT OF GEOSTROPHIC SURFACE CURRENT (CM/SEC
      WRITE(6,303)
  303 FORMAT (10X,'LONG--TERM MEAN'/)
      CALL PRINT2(TCX,1.0)
      WRITE(6,301)
  301 FORMAT('1V  COMPONENT OF GEOSTROPHIC SURFACE CURRENT (CM/SEC)
      CALL PRINT2(TCY,1.0)
C
      CALL OUTPT(TCX,38,KYR,MO,KDY,HUDD)
      CALL OUTPT(TCY,38,KYR,MO,KDY,HVDD)
```

```
C     CLOSE(38,DISP=CRUNCH)
      RETURN
      END
      SUBROUTINE PRINT2(P,F)
      DIMENSION P(40,104),IP(40,104),I1(40)
      WRITE(6,302)
      DO 1 I=1,40
      I1(I)=I
    1 CONTINUE
      WRITE(6,301) (I1(K),K=1,40)
      WRITE(6,302)
  302 FORMAT(/)
      DO 200 I=1,40
      DO 201 JB=1,104
      J=105-JB
      IP(I,J)=P(I,JB)*F
  201 CONTINUE
  200 CONTINUE
      DO 300 J=1,104
      WRITE(6,301) (IP(K,J),K=1,40),J
  300 CONTINUE
  301 FORMAT (1X,40I3,I6)
      RETURN
      END
      SUBROUTINE CSTROT(KYR5,MO5,KDY5,U,V,NOSTOF)
      COMMON ANX(40,104),ANY(40,104),BB(40,104),LS3(40,104),
     'GDIST(40,104)
      DIMENSION GDIST1(40,104),GRDANG(40,104),U(40,104),V(40,104)
      DIMENSION U2S(40,104),V2S(40,104)
C
C  TEST TANGENT AND ARCTANGENT COMPUTATIONS
      PI=3.141596
      RAD=PI/180.0
      ANG1=-2.0*PI
      DANG=10.0*RAD
    1 TAN1=TAN(ANG1)
      ATAN1=ATAN(TAN1)
      ANG2=ATAN1/RAD
C     WRITE(6,106) ANG1,DANG,ANG2,TAN1,ATAN1
  106 FORMAT(5F11.5)
      ANG1=ANG1+DANG
      IF (ANG1.LE.(2.0*PI)) GO TO 1
C
C
C  ONSHORE COMPONENT REDUCTION FACTOR
C  IS PROPORTIONAL TO 1 OVER E TO THE DISTANCE FROM COAST
C  NOTE: E**-0=1, E**-1=0.37, E**-2=0.14, E**-3=0.05
C  THE DISTANCE UNITS ARE IN GRID LENGTHS
C
C
      DO 10 I=2,39
      DO 11 J=2,103
      U2S(I,J)=U(I,J)
      V2S(I,J)=V(I,J)
      IF (LS3(I,J).GE.0) GO TO 11
```

```
      IF(U(I,J).EQ.0.0.AND.V(I,J).EQ.0.0) GO TO 11
      CALL ANGLD(U(I,J),V(I,J),CURANG)
      CALL ANGLD(ANX(I,J),ANY(I,J),CSTANG)
      DELANG=CURANG-CSTANG
      IF (DELANG.GE.270.0) DELANG=360.0-DELANG
      IF (DELANG.LE.-270.0) DELANG=360.0+DELANG
C     WRITE(6,109)I,J,U(I,J),V(I,J),CURANG,CSTANG,DELANG
  109 FORMAT(2I5,5F15.6)
C
C
C  ROTATE FOR DELANG BETWEEN -90 AND +90 DEGREES FROM THE COAST
C  DELANG IS THE ANGLE THE CURRENT VECTOR MAKES WITH THE NORMAL TO
C  +90 IS PARALLEL TO THE COAST TO THE LEFT OF FACING THE COAST
C  -90 IS PARALLEL TO THE COAST TO THE RIGHT
C   00 IS FACING THE COAST (NORMAL)
C  DISTANCES FROM THE COAST ARE TABULATED IN ARRAY GDIST
C
      IF(DELANG.GE.90.0) GO TO 11
      IF(DELANG.LE.-90.0) GO TO 11
C  ROTATE GRID POINTS NEXT TO THE COAST IN THE DIRECTION OF INCIDEN
C  AND LOGARITHMICALLY INVERSELY PROPORTIONAL TO DISTANCE
C  REDUCING FACTOR
      REDFAC=(1.0-(1.0/EXP(GDIST(I,J))))
C  ROTATE
      ABD=ABS(DELANG)
C  MAXIMUM ROTATION IS 45 DEGREES
C  DIMINISHED BY E TO THE -DISTANCE FROM THE COAST (IN GRID LENGTHS
      DROT=45.0/EXP(GDIST(I,J))
C  SET ROTATION AT A MAXIMUM OF 30 DEGREES ANGLE OF INCIDENCE
C  DECREASING TO ZERO AT ZERO AND 90 DEGREES ANGLE OF INCIDENCE
      IF((ABD-30.0).LT.0.0) DIMFAC=SIN((ABD*90.0/30.0)/57.2958)
      IF((ABD-30.0).GE.0.0) DIMFAC=SIN(((90.0-ABD)*90.0/60.0)/57.29
      DROT=DROT*DIMFAC
      IF (DELANG.LT.0.0) DROT=-DROT
      CRANG2=CURANG+DROT
C  FIND NEW COMPONENTS OF CURRENT VECTOR AFTER ROTATION
      RLEN1=SQRT((U(I,J)*U(I,J))+(V(I,J)*V(I,J)))
      U2=RLEN1*COS(CRANG2/57.2958)
      V2=RLEN1*SIN(CRANG2/57.2958)
      IF(V(I,J).LT.0.0) V2=-ABS(V2)
      IF(U(I,J).LT.0.0) U2=-ABS(U2)
C  REDUCE MAGNITUDE OF CURRENT VECTOR
      UR=U2*REDFAC
      VR=V2*REDFAC
C     WRITE(6,333)I,J,GDIST(I,J),REDFAC,DROT,CRANG2,RLEN1,U(I,J),V(
C    'U2,V2,UR,VR
  333 FORMAT(2I4,11F10.4)
C
      U2S(I,J)=UR
      V2S(I,J)=VR
      U(I,J)=UR
      V(I,J)=VR
C
   11 CONTINUE
   10 CONTINUE
```

```
C
      IF(NOSTOF.EQ.1) RETURN
C
C  WRITE FILE OF CURRENTS ROTATED AT THE COAST
  213 DO 34 J=1,104
      DO 34 I=1,5
      I2=I*8
      I1=I2-7
      WRITE(54,100,END=214)(U2S(IK,J),IK=I1,I2),KYR5,MO5,KDY5
   34 CONTINUE
  214 DO 35 J=1,104
      DO 35 I=1,5
      I2=I*8
      I1=I2-7
      WRITE(54,100,END=215)(V2S(IK,J),IK=I1,I2),KYR5,MO5,KDY5
   35 CONTINUE
  215 CONTINUE
  100 FORMAT(8F8.2,3I3,A3)
C
C
C
      RETURN
      END
      SUBROUTINE ANGLD(X,Y,AAA)
      IQ1=0
      IQ2=0
      IQ3=0
      IQ4=0
      IF (X.GE.0.0.AND.Y.GE.0.0) IQ1=1
      IF (X.LT.0.0.AND.Y.GE.0.0) IQ2=1
      IF (X.LT.0.0.AND.Y.LT.0.0) IQ3=1
      IF (X.GE.0.0.AND.Y.LT.0.0) IQ4=1
      PI=3.141596
      RAD=PI/180.0
      IF (X.GE.0.0.AND.X.LT.0.000001) X=0.000001
      IF (X.LT.0.0.AND.X.GT.-0.000001) X=-0.000001
      AA=Y/X
      IF (IQ1.EQ.1) ANG=ATAN(AA)
      IF (IQ1.EQ.1) GO TO 50
      IF (IQ2.EQ.1) ANG=PI + ATAN(AA)
      IF (IQ2.EQ.1) GO TO 50
      IF (IQ3.EQ.1) ANG=PI + ATAN(AA)
      IF (IQ3.EQ.1) GO TO 50
      IF (IQ4.EQ.1) ANG=(2.0*PI) + ATAN(AA)
   50 CONTINUE
      AAA=ANG/RAD
C     WRITE(6,101)IQ1,IQ2,IQ3,IQ4,Y,X,AA,ANG,AAA
  101 FORMAT(4I4,5F11.5)
      RETURN
      END
      SUBROUTINE IJTOLL(Y,X,Y2,X2)
      COMMON/LAT/ALAT2,LS2
      COMMON/LON/ALON2
      DIMENSION ALAT2(40,104),ALON2(40,104)
C
```

```
C
C  CONVERT (I,J) TO LAT-LONG
C
C
C  OR FIND LAT AND LONG OF RANDOM GRID POINTS (Y,X)
C     WITHIN SLOPING GRID SQUARE   A----B
C                                  C----C
C
C
      IX=X
      DX=X-IX
      IY=Y
      DY=Y-IY
C  CORNER A = (IY,IX) INTEGERS
C  GO DOWN AC TO (YB,XB)
      YB=ALAT2(IY,IX)+(DY*(ALAT2(IY+1,IX)-ALAT2(IY,IX)))
      XB=ALON2(IY,IX)+(DY*(ALON2(IY+1,IX)-ALON2(IY,IX)))
C  COMPUTE WEIGHTED AVERAGE SLOPES FOR AB AND CD PER UNIT GRID DIST
      SLA=((1.0-DY)*(ALAT2(IY,IX+1)-ALAT2(IY,IX))) +
     '    ((     DY)*(ALAT2(IY+1,IX+1)-ALAT2(IY+1,IX)))
      SLO=((1.0-DY)*(ALON2(IY,IX+1)-ALON2(IY,IX))) +
     '    ((     DY)*(ALON2(IY+1,IX+1)-ALON2(IY+1,IX)))
C  FIND LAT-LONG OF X,Y
      Y2=YB+(DX*SLA)
      X2=XB+(DX*SLO)
C     WRITE(6,5011)ALAT2(IY,IX),ALAT2(IY,IX+1),ALAT2(IY+1,IX),ALAT2
C    ',IX+1),YB,SLA,Y,DY,Y2
 5011 FORMAT(' LAT CONV  ',9F8.3)
C     WRITE(6,5012)ALON2(IY,IX),ALON2(IY,IX+1),ALON2(IY+1,IX),ALON2
C    ',IX+1),XB,SLO,X,DX,X2
 5012 FORMAT(' LON CONV  ',9F8.3)
      RETURN
      END
```

APPENDIX 2

Listing of Fortran Program "M/PLOT/AUTO" Graphics Outputs on 40X104 Grid.

```
$RESET FREE
FILE  1(KIND=DISK,TITLE="FNWC/LATLON/GRID",FILETYPE=7)
FILE  2(KIND=DISK,FILETYPE=7)
FILE  6(KIND=PRINTER)
FILE  13(KIND=DISK,TITLE="CURRENTS/40X64/LANDSEA",FILETYPE=7)
FILE  14(KIND=DISK,TITLE="CURRENTS/40X64W/LANDSEA",FILETYPE=7)
FILE  25(KIND=DISK,TITLE="CURRENTS/40X64W/COAST/DISTANCES/NORMALS"
     ',FILETYPE=7)
FILE  26(KIND=DISK,TITLE="CURRENTS/40X64/COAST/DISTANCES/NORMALS"
     ',FILETYPE=7)
FILE  28(KIND=DISK,TITLE="CURRENTS/40X104/PROGRESVECT/IJPLOTPOINTS"
     'FILETYPE=7)
FILE  30(KIND=DISK,TITLE="CURRENTS/40X64/LATLON",FILETYPE=7)
FILE  31(KIND=DISK,TITLE="CURRENTS/40X64W/LATLON",FILETYPE=7)
FILE  35(KIND=DISK,TITLE="CURRENTS/40X104/FASTLATITUDES/CCSQ/FIG",
     'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  36(KIND=DISK,TITLE="CURRENTS/40X104/FASTLATITUDES/CCSQ/FIG",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  40(KIND=DISK,TITLE="AF",MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=
FILE  41(KIND=DISK,TITLE="PRINT/MORE/LAND",FILETYPE=7)
FILE  51(KIND=DISK,
     'TITLE="CURRENTS/40X104/DSLP/78/FM7",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  52(KIND=DISK,
     'TITLE="CURRENTS/40X104/WIND/78/FM7/D17TOD31",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  49(KIND=DISK,TITLE="CURRENTS/40X104/TCURR/M1TOM12",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  50(KIND=DISK,TITLE="CURRENTS/40X104/DYNCURR/S",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  53(KIND=DISK,
     'TITLE="CURRENTS/40X104/SFCOCN/78/FM7/D17TOD31",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  54(KIND=DISK,
     'TITLE="CURRENTS/40X104/W10DD10/CURR/78/FM7/D17TOD31/CSTROTATD
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  55(KIND=DISK,
     'TITLE="CURRENTS/40X104/W10DD10/CURR/78/FM7/D17TOD31",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  60(KIND=DISK,
     'TITLE="CURRENTS/40X104/PROGRESVECT/78/FM7/D17TOD31",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
FILE  16(KIND=DISK,TITLE="RACE/DEPTHCONTOURS/CHAR/50M",FILETYPE=7)
FILE  17(KIND=DISK,TITLE="RACE/DEPTHCONTOURS/CHAR/100M",FILETYPE=7)
FILE  18(KIND=DISK,TITLE="RACE/DEPTHCONTOURS/CHAR/200M",FILETYPE=7)
FILE  19(KIND=DISK,TITLE="RACE/DEPTHCONTOURS/CHAR/2000M",FILETYPE=7
FILE  56(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/50M",
     'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
FILE  57(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/100M",
     'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
```

```
      FILE  58(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/200M",
     'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
      FILE  59(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/2000M",
     'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
      FILE  66(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/50M",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
      FILE  67(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/100M",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
      FILE  68(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/200M",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
      FILE  69(KIND=DISK,TITLE="CURRENTS/40X104/FASTBATH/CCSQ/FIG/2000M",
     'MAXRECSIZE=14,BLOCKSIZE=420,FILETYPE=7)
      FILE  70(KIND=DISK,TITLE="RACE/COAST/CHAR/FNWC",FILETYPE=7)
      FILE  71(KIND=DISK,TITLE="CURRENTS/40X104/FASTCOAST/CCSQ/FIG",
     'MAXRECSIZE=14,BLOCKSIZE=420,NEWFILE=.TRUE.)
      FILE  73(KIND=DISK,TITLE="CURRENTS/40X104/FASTCOAST/CCSQ/FIG",
     'FILETYPE=7)
      COMMON ANX(40,104),ANY(40,104),BB(40,104),LS3(40,104),
     'GDIST(40,104)
      COMMON/GRDCOM/CFACT,M,N,NM,XI(105),YJ(41)
      DIMENSION ALAT2(40,104),ALON2(40,104)
      DIMENSION LS1(40,104),LS2(40,104)
      DIMENSION U2(40,104),V2(40,104),OCUW(40,104),OCVW(40,104)
      DIMENSION POCU(40,104),POCV(40,104),OCUU(4160),OCVV(4160),
     'CSPEED(40,104)
      DIMENSION OCUT(40,104),OCVT(40,104)
      DIMENSION WXX(40,104),WYY(40,104),PWXX(40,104),PWYY(40,104)
      DIMENSION CSLP(4160),BSV2(576)
      DIMENSION T1(40,104),T2(40,104)
      DIMENSION IA(80)
      DIMENSION SLP(4160),SLP1(40,104),SLP2(40,104)
      DIMENSION HWX(1),HWY(1),WXXU(4160),WYYV(4160),WSPEED(40,104)
      DIMENSION HSLP(1),HSFOU(1),HSFOV(1)
      DIMENSION DEPTH(4),NDATA(4),BATH(3,1000,4)
      DIMENSION A(3),B(2),C(1)
      DIMENSION C1(25000),C2(25000),C3(25000)
      DIMENSION OCUS(40,104),OCVS(40,104)
      DIMENSION OCUDD(40,104),OCVDD(40,104)
      DIMENSION GU1(200),GV1(200),GU2(200),GV2(200)
      DIMENSION A2(3)
      DIMENSION IPV(200),JPV(200)
      DIMENSION ALATSF(14)
      DIMENSION PLO(2000),PLA(2000),IPEN(2000)
      DIMENSION YMAX(100),IOFFMP(200)
C     DATA ALATSF/31.9,46.5,22.3,60.7,32.7,65.9,35.8,67.4/
      DATA ALATSF/47.8,46.2,60.5,33.9,65.7,32.9,
     '67.3,35.8,65.9,32.8,60.5,22.5,46.3,32.0/
      DATA A/"    FROM   YEAR=  "/
      DATA A2/"      TO    YEAR=  "/
      DATA B/", MONTH="/
      DATA C/", DAY="/
      DATA IAA/'      '/
      DATA IAB/'*****'/
      DATA HWX/' WX   '/,HWY/' WY   '/
      DATA HSLP/'SLP   '/
```

```
      DATA HSFOU/'OU0   '/,HSFOV/'OV0   '/
      NN2=40
      MM2=104
C
C
C-----------------------------------------------------------------
C  PROGRAM TO PLOT CURRENT FIELDS ON 40X64W GRID
C  1. PLOT WIND VECTORS
C  2. PLOT WIND CURRENT VECTORS
C  3. PLOT THERMAL CURRENT VECTORS
C  4. PLOT GEOSTROPHIC CURRENT VECTORS
C  5. PLOT WIND + THERMAL + GEOSTROPHIC CURRENT VECTORS
C  6. ADD COASTAL BENDING OF CURRENTS TO 5. AND PLOT
C  7. PROGRESSIVE VECTORS (AS STORED IN FILE)
C
C
C
C-----------------------------------------------------------------
C
C
C################################################################
C################################################################
C  LABEL LONG'S
C  CHOOZE WHICH PLOTS TO DO
      EVGRPT=0.0
      PLWIND=0.0
      PLOCW=0.0
      PLOCT=0.0
      PLOCDD=0.0
      PLOCS=0.0
      PLSOCR=0.0
C  PLOT SLP WITH CURRENTS (IPLSLP=1)
      IPLSLP=0
C  PROGRESIVE VECTORS
      PROGV=1.0
      IFCOP=1
C   IF NPLOT=2 THE PLOT OF WIND+THERMAL+GEO CURRENT VECTORS
C   (#5)WILL BE PLOTTED WITH THE COASTAL BENDING (#6) OVERLAYED
C   ON THE PLOT.
C   MAKE SURE THAT PLOCS AND PLSOCR IS BOTH EQUAL TO 1.0
      NPLOT=0
      WRITE(6,80)
   80 FORMAT('1CONTROL CARD PARAMETERS FROM WFL CARD DECK'//)
   81 FORMAT(F5.1,67A1)
   82 FORMAT(F5.1,67A1/)
      READ(5,81)EVGRPT,(CC(I),I=1,67)
      WRITE(6,82)EVGRPT,(CC(I),I=1,67)
      READ(5,81)PLWIND,(CC(I),I=1,67)
      WRITE(6,82)PLWIND,(CC(I),I=1,67)
      READ(5,81)PLOCW,(CC(I),I=1,67)
      WRITE(6,82)PLOCW,(CC(I),I=1,67)
      READ(5,81)IPLSLP,(CC(I),I=1,67)
      WRITE(6,82)IPLSLP,(CC(I),I=1,67)
      READ(5,81)PLOCDD,(CC(I),I=1,67)
      WRITE(6,82)PLOCDD,(CC(I),I=1,67)
```

```
      READ(5,81)PLOCS,(CC(I),I=1,67)
      WRITE(6,82)PLOCS,(CC(I),I=1,67)
      READ(5,81)PLSOCR,(CC(I),I=1,67)
      WRITE(6,82)PLSOCR,(CC(I),I=1,67)
      READ(5,81)PROGV,(CC(I),I=1,67)
      WRITE(6,82)PROGV,(CC(I),I=1,67)
      READ(5,81)NPLOT,(CC(I),I=1,67)
      WRITE(6,82)NPLOT,(CC(I),I=1,67)
C  SKIP PLOTS 1 UP TO 1
      NOPLT=1
C  DO NDAYS OF PLOTS
C     NDAYS=12
      NDAYS=180
C     NDAYS=10
      NDAYS=500
C################################################################
      WFAC=1.0
      TFAC=0.0
      DDFAC=1.0
C
C
C
C  READ IN LAT-LONGS OF 40X64 + 40X64W GRIDS (CURRENTS MODEL)
C
      DO 630 IFILE=1,2
      DO 631 IK=1,4
      DO 632 J=41,104
      I2=IK*10
      I1=I2-9
      IF(IFILE.EQ.1)READ(30,104)(ALAT2(I,J),I=I1,I2)
      IF(IFILE.EQ.2)READ(30,104)(ALON2(I,J),I=I1,I2)
  632 CONTINUE
  631 CONTINUE
  630 CONTINUE
  104 FORMAT (10F8.3,I4)
      DO 680 IFILE=1,2
      DO 681 IK=1,4
      DO 682 J=1,64
      I2=IK*10
      I1=I2-9
      IF(IFILE.EQ.1)READ(31,104)(ALAT2(I,J),I=I1,I2)
      IF(IFILE.EQ.2)READ(31,104)(ALON2(I,J),I=I1,I2)
  682 CONTINUE
  681 CONTINUE
  680 CONTINUE
C
C  ROUGH PRINT LATS AND LONGS
      DO 51 I=1,40
C     WRITE(6,105) (ALAT2(I,J),J=1,104,5),ALAT2(I,104)
   51 CONTINUE
      DO 52 I=1,40
C     WRITE(6,105) (ALON2(I,J),J=1,104,5),ALON2(I,104)
   52 CONTINUE
  105 FORMAT (22F6.1)
C
```

```
C
C  READ LAND-SEA TABLE
C
      DO 1500 I=1,NN2
      READ(14,1501,END=1503)(LS3(I,J),J=1,64)
C     WRITE(6,1505) I,LS3(I,1)
 1500 CONTINUE
 1503 CONTINUE
 1505 FORMAT (1H ,2I5)
 1501 FORMAT(64I1)
      DO 1600 I=1,NN2
      READ(13,1501,END=1603)(LS3(I,K),K=41,104)
C     WRITE(6,1505) I,LS3(I,41)
 1600 CONTINUE
 1603 CONTINUE
C
C  WRITE LAND-SEA
      WRITE(6,1514)
 1514 FORMAT('1 LAND-SEA 2, (40X104)'//)
      DO 1515 I=1,NN2
      WRITE(6,1516)(LS3(I,J),J=1,104)
 1515 CONTINUE
 1516 FORMAT(1H ,104I1)
C
C
C
      DO 8001 I=1,40
      DO 8002 J=1,104
      LS1(I,J)=1
 8002 CONTINUE
 8001 CONTINUE
C
C
C  READ INITIAL I,J PLOINT TO PLOT IN PROGRESSIVE VECTOR PLOT
      KK=1
 1144 READ(28,1145,END=1146) IPV(KK),JPV(KK)
      WRITE(6,1145) KK,IPV(KK),JPV(KK)
      IF(IPV(KK).LE.0.OR.JPV(KK).LE.0) GO TO 1146
      KK=KK+1
      GO TO 1144
 1145 FORMAT(3I5)
 1146 CONTINUE
      KK=KK-1
      WRITE(6,1145) KK
      IIEND=KK
C
C
C
      XLN=12.
      YLN=9.0
      YYLEN=8.75
      YYLEN=9.25
      CALL RTIME(101)
C  FOR SQUARE GRID --- XLN=12.0, YLN SHOULD BE =4.54369 PLUS 1 INCH
C  FOR SQUARE GRID --- XLN=15.0, YLN SHOULD BE =5.67961 PLUS 1 INCH
```

```
      XLN=15.0
      YLN=6.67961
      YYLEN=YLN+0.5
      CALL PLOTS(960,3,4096,0.,17.0,0.,9.0)
C     CALL PLOTS(960,3,4096,0.,14.5,0.,10.9)
C     CALL PLOTS(480,0,1024,0.,14.5,0.,10.9)
      CALL RTIME(102)
      CALL STGRID(NN2,MM2,ALAT2,ALON2,LS1,XLN,YLN)
C
C
C  SKIP COMPUTATION OF PLOT LATITUDE LINES
      NOLATS=1
C     NOLATS=0
      IF (NOLATS.EQ.1) GO TO 2222
C  STORE 30, 40, 50, AND 60 DEGREE LATITUDE LINES
      NCC=1
      DLON=0.5
      DO 1131 NY=3,6
      CALL RTIME(1)
      YY=FLOAT(NY)*10.0
      ALOMAX=250.0
      ALOMIN=90.0
      XX=ALOMAX
      NC=1
      YY1=YY
 1132 CONTINUE
      XX1=XX
      CALL CONVRT(XX,YY,NEND)
      IF (NEND.EQ.1) XX=XX1-DLON
      IF (NEND.EQ.1) YY=YY1
C     IF(NEND.EQ.1)WRITE (6,1151) XX1,XX,XLN,YY1,YY,YLN,NEND,NC,NCC
      IF (NEND.EQ.1.AND.XX1.LT.110.0) GO TO 1131
      IF (NEND.EQ.1.) GO TO 1132
 1151 FORMAT (6F10.4,3I6)
      NPEN=2
      IF (NC.EQ.1) NPEN=3
      PLO(NCC)=XX
      PLA(NCC)=YY
      IPEN(NCC)=NPEN
      WRITE(35,1159) XX,YY,NPEN,NC,NCC
 1159 FORMAT(2F10.4,3I10)
      XX=XX1-DLON
      YY=YY1
      IF (XX.LT.ALOMIN) GO TO 1131
      NC=NC+1
      NCC=NCC+1
      GO TO 1132
 1131 CONTINUE
      CLOSE(35,DISP=CRUNCH)
C  TEST WRITE
      DO 1133 I=1,104
      II=I
      IF (II.GT.41) II=41
      WRITE (6,1134)CFACT,M,N,NM,XI(I),YJ(II),I
 1133 CONTINUE
```

```
 1134 FORMAT (4I8,2F8.4,I8)
C
C
      NAROW=1
      IF (NAROW.EQ.1) GO TO 2222
C  TEST ARROWS
      XX1=0.0
      YY1=0.0
      DXX1=0.25
      DYY1=0.1
      DDXX1=0.05
      DDYY1=0.1
      NNN=1
 1156 X1=XX1*(XLN/103.0)
      Y1=(YLN-1.0)-(YY1*((YLN-1.0)/39.0))
      X2=XX1+DXX1
      Y2=YY1+DYY1
      X2=X2*(XLN/103.0)
      Y2=(YLN-1.0)-(Y2*((YLN-1.0)/39.0))
      X1=X1+1.0
      Y1=Y1+1.0
      X2=X2+1.0
      Y2=Y2+1.0
      CALL PVECTR(X1,Y1,X2,Y2,10.0,10.0)
      WRITE(6,1155)XX1,YY1,NNN,X1,Y1,X2,Y2
 1155 FORMAT (1H ,2F10.3,I5,4F10.3)
      XX1=XX1+DXX1
      DYY1=DYY1+DDYY1
      YY1=YY1+DYY1
      IF (YY1.GT.15.0.AND.DXX1.GE.0.0) DXX1=-DXX1
      NNN=NNN+1
      IF (YY1.LT.40.0) GO TO 1156
C
C  STOP HERE
      GO TO 11111
 2222 CONTINUE
C
C  READ FASTLATS
      IF(NOLATS.LT.1) GO TO 2223
      NCC=1
 2225 READ(36,1159,END=2223)PLO(NCC),PLA(NCC),IPEN(NCC)
      NCC=NCC+1
      GO TO 2225
 2223 CONTINUE
C
      NCC=NCC-1
      IPEN(NCC)=3
C
C
      CALL RTIME(2)
C
C  MAKE FASTCOASTLINE FILE
      ISKIP=1
      IFAST=1
      IFAST=0
```

```
      IF (IFAST.EQ.1) CALL FASTCO(IFAST,ISKIP,IC)
      IF (IFAST.LT.20) CALL CSET(73,IC,C1,C2,C3)
C     CALL COAST(0,1,IC,C1,C2,C3)
      CALL RTIME(3)
C*
C*
C*  READ BATHYMETRY FILE
C*
      DEPTH(1)=50.
      DEPTH(2)=100.
      DEPTH(3)=200.0
      DEPTH(4)=2000.0
      NDATA(1)=1000
      NDATA(2)=1000
      NDATA(3)=1000
      NDATA(4)=1000
      NDATA(2)=0
      NDATA(4)=0
C
C
C  SKIP CONVERSION OF BATHYMETRY IF NOCONV=1
      NOCONV=1
C     NOCONV=0
      IF(NOCONV.EQ.1) GO TO 951
C
      IB1=0
      IB2=0
      IEND1=0
      IEND2=0
      IEND3=0
      IEND4=0
C
  910 CONTINUE
C
      NDATA(1)=1000
      DO 911 II1=1,1000
      IF (IEND1.EQ.1) NDATA(1)=0
      IF (IEND1.EQ.1) GO TO 913
      READ(16,9900,END=912) (BATH(J,II1,1),J=1,3),NRECS1
      IF (II1.EQ.1) BATH(3,1,1)=3
  911 CONTINUE
      GO TO 913
  912 IEND1=1
      NDATA(1)=II1-2
      WRITE(6,9911) II1,NRECS1
 9911 FORMAT(' END OF FILE 16, LAST READ =',2I6)
  913 CONTINUE
C
      NDATA(2)=0
      DO 921 II2=1,1000
      IF (IEND2.EQ.1) NDATA(2)=0
      IF (IEND2.EQ.1) GO TO 923
      READ(17,9900,END=922) (BATH(J,II2,2),J=1,3),NRECS2
      IF (II2.EQ.1) BATH(3,1,2)=3
  921 CONTINUE
```

```
      GO TO 923
  922 IEND2=1
      NDATA(2)=II2-2
      WRITE(6,9912) II2,NRECS2
 9912 FORMAT(' END OF FILE 17, LAST READ =',2I6)
  923 CONTINUE
C
      NDATA(3)=1000
      DO 931 II3=1,1000
      IF (IEND3.EQ.1) NDATA(3)=0
      IF (IEND3.EQ.1) GO TO 933
      READ(18,9900,END=932) (BATH(J,II3,3),J=1,3),NRECS3
      IF (II3.EQ.1) BATH(3,1,3)=3
  931 CONTINUE
      GO TO 933
  932 IEND3=1
      NDATA(3)=II3-2
      WRITE(6,9913) II3,NRECS3
 9913 FORMAT(' END OF FILE 18, LAST READ =',2I6)
  933 CONTINUE
C
      NDATA(4)=0
      DO 941 II4=1,1000
      IF (IEND4.EQ.1) NDATA(4)=0
      IF (IEND4.EQ.1) GO TO 943
      READ(19,9900,END=942) (BATH(J,II4,4),J=1,3),NRECS4
      IF (II4.EQ.1) BATH(3,1,4)=3
  941 CONTINUE
      GO TO 943
  942 IEND4=1
      WRITE(6,9914) II4,NRECS4
 9914 FORMAT(' END OF FILE 19, LAST READ =',2I6)
      NDATA(4)=II4-2
  943 CONTINUE
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
 9900 FORMAT(2F10.4,2I10)
C
C  CONVERT BATHYMETRY
      CALL BATHY(1,0,0,BATH,NDATA,DEPTH)
C
C
C  STORE FILES OF FAST BATHYMETRY CONVERTED
      DO 915 NB=1,4
      IFILE=55+NB
      IF (NDATA(NB).EQ.0) GO TO 915
      DO 916 NK=1,NDATA(NB)
      WRITE(IFILE,9900) (BATH(J,NK,NB),J=1,3)
  916 CONTINUE
  915 CONTINUE
      CLOSE(56,DISP=CRUNCH)
      CLOSE(57,DISP=CRUNCH)
      CLOSE(58,DISP=CRUNCH)
      CLOSE(59,DISP=CRUNCH)
C
C
```

```
  951 CONTINUE
C
      CALL RTIME(4)
C  READ CONVERTED FAST BATHYMETRY FILES
      IF (NOCONV.NE.1) GO TO 960
      DO 952 NK=1,1000
      READ(66,9900,END=953) (BATH(J,NK,1),J=1,3)
  952 CONTINUE
  953 CONTINUE
      NDATA(1)=NK-1
      WRITE(6,9926) NDATA(1)
 9926 FORMAT ('0END OF FILE 66  ',I5,' RECORDS READ'/)
C
      DO 954 NK=1,1000
      READ(67,9900,END=955) (BATH(J,NK,2),J=1,3)
  954 CONTINUE
  955 CONTINUE
      NDATA(2)=NK-1
      WRITE(6,9927) NDATA(2)
 9927 FORMAT ('0END OF FILE 67  ',I5,' RECORDS READ'/)
C
      DO 956 NK=1,1000
      READ(68,9900,END=957) (BATH(J,NK,3),J=1,3)
  956 CONTINUE
  957 CONTINUE
      NDATA(3)=NK-1
      WRITE(6,9928) NDATA(3)
 9928 FORMAT ('0END OF FILE 68  ',I5,' RECORDS READ'/)
C
      DO 958 NK=1,1000
      READ(69,9900,END=959) (BATH(J,NK,4),J=1,3)
  958 CONTINUE
  959 CONTINUE
      NDATA(4)=NK-1
      WRITE(6,9929) NDATA(4)
 9929 FORMAT ('0END OF FILE 69  ',I5,' RECORDS READ'/)
  960 CONTINUE
      CALL RTIME(5)
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
 9901 FORMAT(8I10)
      IFLBLR=1
      CALL RTIME(6)
C
C
C  GET MORE BATHYMETRY DATA
C     IF(IEND1.EQ.1.AND.IEND2.EQ.1) IB1=1
C     IF(IEND3.EQ.1.AND.IEND4.EQ.1) IB2=1
C     IF(IB1.NE.1.OR.IB2.NE.1) GO TO 910
C
C     DO 36 J=1,104
C     DO 36 I=1,5
      I2=I*8
```

```
      I1=I2-7
C     READ (49,100,END=216)(OCUT(IK,J),IK=I1,I2),KYR6,MO6,KDY6
   36 CONTINUE
C 216 DO 37 J=1,104
C     DO 37 I=1,5
      I2=I*8
      I1=I2-7
C     READ(49,100,END=217)(OCVT(IK,J),IK=I1,I2),KYR7,MO7,KDY7
   37 CONTINUE
  217 CONTINUE
      DO 38 J=1,104
      DO 38 I=1,5
      I2=I*8  ; I1=I2-7
      READ(50,100,END=218)(OCUDD(IK,J),IK=I1,I2),KYR8,MO8,KDY8
   38 CONTINUE
  218 CONTINUE
      DO 39 J=1,104
      DO 39 I=1,5
      I2=I*8  ; I1=I2-7
      READ(50,100,END=219)(OCVDD(IK,J),IK=I1,I2),KYR9,MO9,KDY9
   39 CONTINUE
  219 CONTINUE
C
C  MULTIPLY BY TEMP AND DD FACTORS HERE
      DO 406 J=1,104
      DO 407 I=1,40
      OCUT(I,J)=OCUT(I,J)*TFAC
      OCVT(I,J)=OCVT(I,J)*TFAC
      OCUDD(I,J)=OCUDD(I,J)*DDFAC
      OCVDD(I,J)=OCVDD(I,J)*DDFAC
  407 CONTINUE
  406 CONTINUE
C
C
C  DO 1 SETS OF PLOTS
      DO 9000 KPLT=1,NDAYS
C
C   READ IN SLP, WIND, AND OCNC FILES
C
      IF(IPLSLP.LT.1.0.AND.PLOCW.LT.1.0.AND.PLWIND.LT.1.0) GO TO 21
      DO 31 J=1,104
      DO 31 I=1,5
      I2=I*8
      I1=I2-7
      READ(51,100,END=211)(SLP1(IK,J),IK=I1,I2),KYR1,MO1,KDY1
   31 CONTINUE
  211 DO 32 J=1,104
      DO 32 I=1,5
      I2=I*8
      I1=I2-7
      READ(52,100,END=212)(WXX(IK,J),IK=I1,I2),KYR2,MO2,KDY2
   32 CONTINUE
  212 DO 33 J=1,104
      DO 33 I=1,5
      I2=I*8
```

```
      I1=I2-7
      READ(52,100,END=213)(WYY(IK,J),IK=I1,I2),KYR3,MO3,KDY3
   33 CONTINUE
  213 DO 34 J=1,104
      DO 34 I=1,5
      I2=I*8
      I1=I2-7
      READ (53,100,END=214)(OCUW(IK,J),IK=I1,I2),KYR4,MO4,KDY4
   34 CONTINUE
  214 DO 35 J=1,104
      DO 35 I=1,5
      I2=I*8
      I1=I2-7
      READ(53,100,END=9000)(OCVW(IK,J),IK=I1,I2),KYR5,MO5,KDY5
   35 CONTINUE
      CALL DATFIX(SLP1,CSLP)
  215 CONTINUE
  100 FORMAT(8F8.3,3I3,A3)
C
C     WRITE(6,231) KYR1,MO1,KDY1
C     WRITE(6,231) KYR2,MO2,KDY2
C     WRITE(6,231) KYR3,MO3,KDY3
C     WRITE(6,231) KYR4,MO4,KDY4
C     WRITE(6,231) KYR5,MO5,KDY5
C     WRITE(6,231) KYR6,MO6,KDY6
C     WRITE(6,231) KYR7,MO7,KDY7
C     WRITE(6,231) KYR8,MO8,KDY8
C     WRITE(6,231) KYR9,MO9,KDY9
  231 FORMAT(3I10)
      YEAR=KYR1
      AMO=MO1
      DAY=KDY1
C     CALL RTIME(7)
      FYR=KYR1
      FMO=MO1
      FDY=KDY1
C
C
C  SKIP PLOTS FROM 1 TO NOPLT
      IF (KPLT.LT.NOPLT) GO TO 965
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
C  DO PLOT NUMBER 1 (SLP CONTOURS + WIND VECTORS)
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
      IF (PLWIND.LT.1.0) GO TO 961
C
C  PRINT SLP (40X64W) CONTOUR ARRAY
C     WRITE (6,664)
  664 FORMAT('1 SLP CONTOURS'/)
C     DO 175 KJ=1,416
C     I2=KJ*10
```

```
C     I1=I2-9
C     WRITE (6,176) (CSLP(IK),IK=I1,I2),I1,I2,KJ
  175 CONTINUE
  176 FORMAT (10F10.2,3I5)
      CALL PLOT (1.0,1.0,-3)
      CALL NEWPEN(2)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
      CALL NUMMER
C
C  PLOT LATITUDE LINES
      DO 9141 IPK=1,NCC
      CALL PLOT (PLO(IPK),PLA(IPK),IPEN(IPK))
 9141 CONTINUE
C
      CALL NEWPEN(2)
C
C  PLOT LONGITUDES/CCSQ/FIG FROM 910 (+20) TO 230 DEGREES
      CHSZ=0.307
      DO 9161 LO=1,7
      IF(LO.EQ.1) GO TO 9161
      XX=110.0+((LO-1)*20.0)
      XXL=XX
      NLA=LO*2
      YY=ALATSF(NLA-1)
      XX1=XX
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,3)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      YY=ALATSF(NLA)
      XX=XX1
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,2)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      IF(LO.EQ.2) GO TO 9161
      IF (LO.EQ.7) GO TO 9161
      XA=XX-(2.0*CHSZ)
      YA=YY - 0.5
      IF (LO.EQ.5) XXL=170.0
      IF (LO.EQ.6) XXL=150.0
      CALL NUMBER(XA,YA,CHSZ,XXL,0.0,-1)
      XAW=999.0
      YAW=999.0
      IF(LO.EQ.3.OR.LO.EQ.4)CALL SYMBOL(XAW,YAW,CHSZ,' W',0.0,2)
      IF(LO.GE.5.OR.LO.EQ.6)CALL SYMBOL(XAW,YAW,CHSZ,' E',0.0,2)
 9161 CONTINUE
C  LABEL LAT'S
      ALO=170.0
      DO 9176 NL=1,2
      ALA=30.0 + ((NL-1)*10)
      ALA1=ALA
      IF(NL.EQ.1) CLO=133.5
      IF(NL.EQ.2) CLO=123.0
```

```
      CALL CONVRT(CLO,ALA,IEND)
      WRITE(6,6072)CLO,CLA,ALA1
      CLA=ALA-(0.5*CHSZ)
      CLO=XLN+0.1
      CALL NUMBER(CLO,CLA,CHSZ,ALA1,0.0,-1)
      WRITE(6,6072)CLO,CLA,ALA1
      IF (NL.LT.3)CALL SYMBOL(999.0,999.0,CHSZ,' N',0.0,2)
 9176 CONTINUE
C
      XX=0.5
      YY=YYLEN-0.3
      HH=0.2
      CALL SYMBOL(XX,YY,HH,'SEA LEVEL PRESSURE (MB)    ',0.0,26)
      CALL NUMBER(999.0,999.0,HH,FYR,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FMO,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FDY,0.0,-1)
C
C
      CALL NEWPEN(1)
      CALL CONTOR(0,1,900.,1600.,4.,8.,16.,0,CSLP)
C
C  SCALE DOWN ARROWS IN WIND VECTOR PLOT
      FACTOR=0.25
      DO 844 I=1,NN2
      DO 845 J=1,MM2
      WSPEED(I,J)=SQRT((WXX(I,J)*WXX(I,J))+(WYY(I,J)*WYY(I,J)))
      PWXX(I,J)=WXX(I,J)*FACTOR
      PWYY(I,J)=WYY(I,J)*FACTOR
  845 CONTINUE
  844 CONTINUE
C
      CALL PRINT1(WSPEED,8)
C
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
      IF (EVGRPT.EQ.1.0) GO TO 971
      DO 765 JR=1,104,2
      DO 765 IR=1,40
      PWXX(IR,JR)=0.0
      PWYY(IR,JR)=0.0
  765 CONTINUE
      DO 766 IR=1,40,2
      DO 766 JR=1,104
      PWXX(IR,JR)=0.0
      PWYY(IR,JR)=0.0
  766 CONTINUE
  971 CONTINUE
      DO 767 IR=1,40
      PWXX(IR,50)=0.0
      PWYY(IR,50)=0.0
  767 CONTINUE
      DO 768 JR=1,104
```

```
      PWXX(40,JR)=0.0
      PWYY(40,JR)=0.0
  768 CONTINUE
C
C
      CALL DATFIX(PWXX,WXXU)
      CALL DATFIX(PWYY,WYYV)
      CALL VECTOR(WXXU,WYYV)
C
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
      XP=XLN*0.72
      YP=YYLEN-0.4
      UVAL=10.0*FACTOR
      CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YYLEN-0.3
      CALL SYMBOL(XP,YP,0.2,9H 10 M/SEC  ,0.,9)
      XP=XLN*0.63
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,20HSURFACE WIND VECTORS ,0.0,20)
C
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
      CALL FIXER(IFCOP)
  961 CONTINUE
C     CALL RTIME(8)
      IF (PLOCW.LT.1.0) GO TO 962
C
C------------------------------------------------------------------
C------------------------------------------------------------------
C
C  DO PLOT NUMBER 2 (SLP CONTOURS + WIND COMP. OCEAN SURFACE CURREN
C
C------------------------------------------------------------------
C------------------------------------------------------------------
C
C  COMPUTE AND SCALE DOWN ARROWS IN OCEAN CURRENT VECTOR PLOT
C  SEE FUNCTION "WINDC" (SFC CURR DUE TO WIND)
      FACTOR=0.25
      DO 1844 I=1,NN2
      DO 1845 J=1,MM2
      CSPEED(I,J)=SQRT(OCUW(I,J)*OCUW(I,J)+OCVW(I,J)*OCVW(I,J))
      IF(LS3(I,J).EQ.0)OCUW(I,J)=0.0
      IF(LS3(I,J).EQ.0)OCVW(I,J)=0.0
      POCU(I,J)=OCUW(I,J)*FACTOR
      POCV(I,J)=OCVW(I,J)*FACTOR
 1845 CONTINUE
 1844 CONTINUE
C
      CALL PRINT1(CSPEED,11)
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
```

```
      IF (EVGRPT.EQ.1.0) GO TO 972
      DO 865 JR=1,104,2
      DO 865 IR=1,40
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  865 CONTINUE
      DO 866 IR=1,40,2
      DO 866 JR=1,104
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  866 CONTINUE
  972 CONTINUE
      DO 867 IR=1,40
      POCU(IR,104)=0.0
      POCV(IR,104)=0.0
  867 CONTINUE
      DO 868 JR=1,104
      POCU(40,JR)=0.0
      POCV(40,JR)=0.0
  868 CONTINUE
C
C
C
      CALL PLOT(1.,1.,-3)
      CALL DATFIX(POCU,OCUU)
      CALL DATFIX(POCV,OCVV)
      CALL NEWPEN(1)
      CALL VECTOR(OCUU,OCVV)
      CALL NEWPEN(2)
C
C     CALL CONTOR(0,1,900.,1600.,8.,16.,16.,0,CSLP)
      CALL CONTOR(0,1,900.,1600.,4.,8.,16.,0,CSLP)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
      CALL NUMMER
C
C  PLOT LATITUDE LINES
      DO 8141 IPK=1,NCC
      CALL PLOT (PLO(IPK),PLA(IPK),IPEN(IPK))
 8141 CONTINUE
C
      CALL NEWPEN(2)
C
C  PLOT LONGITUDES/CCSQ/FIG FROM 810 (+20) TO 230 DEGREES
      CHSZ=0.307
      DO 8161 LO=1,7
      IF(LO.EQ.1) GO TO 8161
      XX=110.0+((LO-1)*20.0)
      XXL=XX
      NLA=LO*2
      YY=ALATSF(NLA-1)
      XX1=XX
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,3)
```

```
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      YY=ALATSF(NLA)
      XX=XX1
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,2)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      IF(LO.EQ.2) GO TO 8161
      IF (LO.EQ.7) GO TO 8161
      XA=XX-(2.0*CHSZ)
      YA=YY - 0.5
      IF (LO.EQ.5) XXL=170.0
      IF (LO.EQ.6) XXL=150.0
      CALL NUMBER(XA,YA,CHSZ,XXL,0.0,-1)
      XAW=999.0
      YAW=999.0
      IF(LO.EQ.3.OR.LO.EQ.4)CALL SYMBOL(XAW,YAW,CHSZ,' W',0.0,2)
      IF(LO.GE.5.OR.LO.EQ.6)CALL SYMBOL(XAW,YAW,CHSZ,' E',0.0,2)
 8161 CONTINUE
C  LABEL LAT'S
      ALO=170.0
      DO 8176 NL=1,2
      ALA=30.0 + ((NL-1)*10)
      ALA1=ALA
      IF(NL.EQ.1) CLO=133.5
      IF(NL.EQ.2) CLO=123.0
      CALL CONVRT(CLO,ALA,IEND)
      WRITE(6,6072)CLO,CLA,ALA1
      CLA=ALA-(0.5*CHSZ)
      CLO=XLN+0.1
      CALL NUMBER(CLO,CLA,CHSZ,ALA1,0.0,-1)
      WRITE(6,6072)CLO,CLA,ALA1
      IF (NL.LT.3)CALL SYMBOL(999.0,999.0,CHSZ,' N',0.0,2)
 8176 CONTINUE
C
      XX=0.5
      YY=YYLEN-0.3
      HH=0.2
      CALL SYMBOL(XX,YY,HH,'SEA LEVEL PRESSURE (MB)    ',0.0,26)
      CALL NUMBER(999.0,999.0,HH,FYR,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FMO,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FDY,0.0,-1)
C
C
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
      IFLBLR=1
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
```

```
      XP=XLN*0.72
      YP=YYLEN-0.7
      UVAL=10.0*FACTOR
      CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YP+0.1
      CALL SYMBOL(XP,YP,0.2,10H10 CM/SEC  ,0.,10)
      XP=XLN*0.63
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,22HSURFACE OCEAN CURRENTS ,0.0,22)
      YP=YP-0.3
      CALL SYMBOL(XP,YP,0.2,19H  (WIND COMPONENT)  ,0.0,19)
C
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL FIXER(IFCOP)
  962 CONTINUE
C     CALL RTIME(9)
      IF (PLOCT.LT.1.0) GO TO 963
C
C------------------------------------------------------------------
C------------------------------------------------------------------
C
C  DO PLOT NUMBER 3 (THERMAL COMPONENT OF OCEAN SURFACE CURRENTS-0/
C
C------------------------------------------------------------------
C------------------------------------------------------------------
C
C  COMPUTE AND SCALE DOWN ARROWS IN OCEAN CURRENT VECTOR PLOT
      FACTOR=1.25
      DO 2844 I=1,NN2
      DO 2845 J=1,MM2
      CSPEED(I,J)=SQRT(OCUT(I,J)*OCUT(I,J)+OCVT(I,J)*OCVT(I,J))
      IF(LS3(I,J).EQ.0)OCUT(I,J)=0.0
      IF(LS3(I,J).EQ.0)OCVT(I,J)=0.0
      POCU(I,J)=OCUT(I,J)*FACTOR
      POCV(I,J)=OCVT(I,J)*FACTOR
 2845 CONTINUE
 2844 CONTINUE
C
      CALL PRINT1(CSPEED,11)
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
      IF (EVGRPT.EQ.1.0) GO TO 973
      DO 875 JR=1,104,2
      DO 875 IR=1,40
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  875 CONTINUE
      DO 876 IR=1,40,2
      DO 876 JR=1,104
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  876 CONTINUE
      DO 877 IR=1,40
```

```
      POCU(IR,104)=0.0
      POCV(IR,104)=0.0
  877 CONTINUE
      DO 878 JR=1,104
      POCU(40,JR)=0.0
      POCV(40,JR)=0.0
  878 CONTINUE
  973 CONTINUE
C
C
C
      CALL PLOT(1.,1.,-3)
      CALL DATFIX(POCU,OCUU)
      CALL DATFIX(POCV,OCVV)
      CALL NEWPEN(1)
      CALL VECTOR(OCUU,OCVV)
      CALL NEWPEN(2)
C
C     CALL CONTOR(0,1,900.,1600.,8.,16.,16.,1,CSLP)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
      CALL NUMMER
      CALL SYMBOL(0.,YYLEN,.2,A,0.,16)
      XX=2.0
      CALL NUMBER(XX,YYLEN,.2,YEAR,0.,-1)
      XX=XX+.6
      CALL SYMBOL(XX,YYLEN,.2,B,0.,9)
      XX=XX+1.0
      CALL NUMBER(XX,YYLEN,.2,AMO,0.,-1)
      XX=XX+0.4
      CALL SYMBOL(XX,YYLEN,0.2,C,0.0,6)
      XX=XX+0.75
      CALL NUMBER(XX,YYLEN,0.2,DAY,0.0,-1)
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
      IFLBLR=1
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
      XP=XLN*0.72
      YP=YYLEN-0.7
      UVAL=10.0*FACTOR
      CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YP+0.1
      CALL SYMBOL(XP,YP,0.2,10H10 CM/SEC  ,0.,10)
      XP=XLN*0.63
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,22HSURFACE OCEAN CURRENTS ,0.0,22)
      YP=YP-0.3
      CALL SYMBOL(XP,YP,0.2,22H (THERMAL COMPONENT)   ,0.0,22)
C
```

```
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL FIXER(IFCOP)
  963 CONTINUE
C     CALL RTIME(10)
      IF (PLOCDO.LT.1.0) GO TO 964
C
C-------------------------------------------------------------------
C-------------------------------------------------------------------
C
C  DO PLOT NUMBER 4 (GEOSTROPHIC  OCEAN SURFACE CURRENTS-0/2000 DB)
C
C-------------------------------------------------------------------
C-------------------------------------------------------------------
C  COMPUTE AND SCALE DOWN ARROWS IN OCEAN CURRENT VECTOR PLOT
      FACTOR=.25
      DO 2944 I=1,NN2
      DO 2945 J=1,MM2
      OCUS(I,J)=OCUDD(I,J)
      OCVS(I,J)=OCVDD(I,J)
      CSPEED(I,J)=SQRT(OCUS(I,J)*OCUS(I,J)+OCVS(I,J)*OCVS(I,J))
      IF(LS3(I,J).EQ.0)OCUS(I,J)=0.0
      IF(LS3(I,J).EQ.0)OCVS(I,J)=0.0
      POCU(I,J)=OCUS(I,J)*FACTOR
      POCV(I,J)=OCVS(I,J)*FACTOR
 2945 CONTINUE
 2944 CONTINUE
C
      CALL PRINT1(CSPEED,11)
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
      IF (EVGRPT.EQ.1.0) GO TO 974
      DO 885 JR=1,104,2
      DO 885 IR=1,40
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  885 CONTINUE
      DO 886 IR=1,40,2
      DO 886 JR=1,104
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  886 CONTINUE
      DO 887 IR=1,40
      POCU(IR,104)=0.0
      POCV(IR,104)=0.0
  887 CONTINUE
      DO 888 JR=1,104
      POCU(40,JR)=0.0
      POCV(40,JR)=0.0
  888 CONTINUE
  974 CONTINUE
C
C
C
      CALL PLOT(1.,1.,-3)
```

```
      CALL DATFIX(POCU,OCUU)
      CALL DATFIX(POCV,OCVV)
      CALL NEWPEN(1)
      CALL VECTOR(OCUU,OCVV)
      CALL NEWPEN(2)
C
C     CALL CONTOR(0,1,900.,1600.,8.,16.,16.,1,CSLP)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
      CALL NUMMER
C
C  PLOT LATITUDE LINES
      DO 7141 IPK=1,NCC
      CALL PLOT (PLO(IPK),PLA(IPK),IPEN(IPK))
 7141 CONTINUE
C
      CALL NEWPEN(2)
C
C  PLOT LONGITUDES/CCSQ/FIG FROM 710 (+20) TO 230 DEGREES
      CHSZ=0.307
      DO 7161 LO=1,7
      IF(LO.EQ.1) GO TO 7161
      XX=110.0+((LO-1)*20.0)
      XXL=XX
      NLA=LO*2
      YY=ALATSF(NLA-1)
      XX1=XX
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,3)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      YY=ALATSF(NLA)
      XX=XX1
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,2)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      IF(LO.EQ.2) GO TO 7161
      IF (LO.EQ.7) GO TO 7161
      XA=XX-(2.0*CHSZ)
      YA=YY - 0.5
      IF (LO.EQ.5) XXL=170.0
      IF (LO.EQ.6) XXL=150.0
      CALL NUMBER(XA,YA,CHSZ,XXL,0.0,-1)
      XAW=999.0
      YAW=999.0
      IF(LO.EQ.3.OR.LO.EQ.4)CALL SYMBOL(XAW,YAW,CHSZ,' W',0.0,2)
      IF(LO.GE.5.OR.LO.EQ.6)CALL SYMBOL(XAW,YAW,CHSZ,' E',0.0,2)
 7161 CONTINUE
C  LABEL LAT'S
      ALO=170.0
      DO 7176 NL=1,2
      ALA=30.0 + ((NL-1)*10)
      ALA1=ALA
      IF(NL.EQ.1) CLO=133.5
```

```
      IF(NL.EQ.2) CLO=123.0
      CALL CONVRT(CLO,ALA,IEND)
      WRITE(6,6072)CLO,CLA,ALA1
      CLA=ALA-(0.5*CHSZ)
      CLO=XLN+0.1
      CALL NUMBER(CLO,CLA,CHSZ,ALA1,0.0,-1)
      WRITE(6,6072)CLO,CLA,ALA1
      IF (NL.LT.3)CALL SYMBOL(999.0,999.0,CHSZ,' N',0.0,2)
 7176 CONTINUE
C
      XX=0.5
      YY=YYLEN-0.3
      HH=0.2
      CALL SYMBOL(XX,YY,HH,'PERMANENT    ',0.0,10)
C
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
      IFLBLR=1
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
      XP=XLN*0.72
      YP=YYLEN-0.7
      UVAL=10.0*FACTOR
      CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YP+0.1
      CALL SYMBOL(XP,YP,0.2,10H10 CM/SEC  ,0.,10)
      XP=XLN*0.63
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,22HSURFACE OCEAN CURRENTS ,0.0,22)
      YP=YP-0.3
      XP=XP-0.35
      CALL SYMBOL(XP,YP,0.2,27H( GEOSTROPHIC COMPONENT)   ,0.0,27)
C
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL FIXER(IFCOP)
  964 CONTINUE
C     CALL RTIME(11)
C
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
C  DO PLOT NUMBER 5 (WIND+THERM+GEOS  OCEAN SURFACE CURRENTS-0/200M
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
      IF(PLOCS.LT.1.0) GO TO 965
C
C  COMPUTE AND SCALE DOWN ARROWS IN OCEAN CURRENT VECTOR PLOT
```

```
      FACTOR=.25
      DO 2954 I=1,NN2
      DO 2955 J=1,MM2
      CSPEED(I,J)=SQRT(OCUS(I,J)*OCUS(I,J)+OCVS(I,J)*OCVS(I,J))
      IF(LS3(I,J).EQ.0)OCUS(I,J)=0.0
      IF(LS3(I,J).EQ.0)OCVS(I,J)=0.0
      POCU(I,J)=OCUS(I,J)*FACTOR
      POCV(I,J)=OCVS(I,J)*FACTOR
 2955 CONTINUE
 2954 CONTINUE
C
      CALL PRINT1(CSPEED,11)
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
      IF (EVGRPT.EQ.1.0) GO TO 975
      DO 995 JR=1,104,2
      DO 995 IR=1,40
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  995 CONTINUE
      DO 996 IR=1,40,2
      DO 996 JR=1,104
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  996 CONTINUE
      DO 997 IR=1,40
      POCU(IR,104)=0.0
      POCV(IR,104)=0.0
  997 CONTINUE
      DO 998 JR=1,104
      POCU(40,JR)=0.0
      POCV(40,JR)=0.0
  998 CONTINUE
  975 CONTINUE
C
      CALL PLOT(1.,1.,-3)
      CALL DATFIX(POCU,OCUU)
      CALL DATFIX(POCV,OCVV)
      CALL NEWPEN(1)
      CALL VECTOR(OCUU,OCVV)
      CALL NEWPEN(2)
C
C     CALL CONTOR(0,1,900.,1600.,8.,16.,16.,1,CSLP)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
      CALL NUMMER
C
C  PLOT LATITUDE LINES
      DO 6141 IPK=1,NCC
      CALL PLOT (PLO(IPK),PLA(IPK),IPEN(IPK))
 6141 CONTINUE
C
      CALL NEWPEN(2)
C
```

```
C  PLOT LONGITUDES/CCSQ/FIG FROM 610 (+20) TO 230 DEGREES
       CHSZ=0.307
       DO 6161 LO=1,7
       IF(LO.EQ.1) GO TO 6161
       XX=110.0+((LO-1)*20.0)
       XXL=XX
       NLA=LO*2
       YY=ALATSF(NLA-1)
       XX1=XX
       YY1=YY
       CALL CONVRT(XX,YY,IEND)
       CALL PLOT (XX,YY,3)
       WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
       YY=ALATSF(NLA)
       XX=XX1
       YY1=YY
       CALL CONVRT(XX,YY,IEND)
       CALL PLOT (XX,YY,2)
       WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
       IF(LO.EQ.2) GO TO 6161
       IF (LO.EQ.7) GO TO 6161
       XA=XX-(2.0*CHSZ)
       YA=YY - 0.5
       IF (LO.EQ.5) XXL=170.0
       IF (LO.EQ.6) XXL=150.0
       CALL NUMBER(XA,YA,CHSZ,XXL,0.0,-1)
       XAW=999.0
       YAW=999.0
       IF(LO.EQ.3.OR.LO.EQ.4)CALL SYMBOL(XAW,YAW,CHSZ,' W',0.0,2)
       IF(LO.GE.5.OR.LO.EQ.6)CALL SYMBOL(XAW,YAW,CHSZ,' E',0.0,2)
 6161  CONTINUE
C  LABEL LAT'S
       ALO=170.0
       DO 6176 NL=1,2
       ALA=30.0 + ((NL-1)*10)
       ALA1=ALA
       IF(NL.EQ.1) CLO=133.5
       IF(NL.EQ.2) CLO=123.0
       CALL CONVRT(CLO,ALA,IEND)
       WRITE(6,6072)CLO,CLA,ALA1
       CLA=ALA-(0.5*CHSZ)
       CLO=XLN+0.1
       CALL NUMBER(CLO,CLA,CHSZ,ALA1,0.0,-1)
       WRITE(6,6072)CLO,CLA,ALA1
       IF (NL.LT.3)CALL SYMBOL(999.0,999.0,CHSZ,' N',0.0,2)
 6176  CONTINUE
C
       XX=0.5
       YY=YYLEN-0.3
       HH=0.2
       CALL SYMBOL(XX,YY,HH,'SEA LEVEL PRESSURE (MB)    ',0.0,26)
       CALL NUMBER(999.0,999.0,HH,FYR,0.0,-1)
       CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
       CALL NUMBER(999.0,999.0,HH,FMO,0.0,-1)
       CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
```

```
      CALL NUMBER(999.0,999.0,HH,FDY,0.0,-1)
C
C
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
      IFLBLR=1
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
      XP=XLN*0.72
      YP=YYLEN-0.7
      UVAL=10.0*FACTOR
      CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YP+0.1
      CALL SYMBOL(XP,YP,0.2,10H10 CM/SEC  ,0.,10)
      XP=XLN*0.63
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,22HSURFACE OCEAN CURRENTS ,0.0,22)
      YP=YP-0.3
      XP=XP-0.35
      CALL SYMBOL(XP,YP,0.2,27H(WIND+THERM+GEOS COMPONENT)   ,0.0,2
C
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL FIXER(IFCOP)
C
C
  965 CONTINUE
      IF(PLSOCR.LT.1.0) GO TO 9000
C
C  READ FILE OF TOTAL CURRENTS AFTER ROTATION
      DO 94 J=1,104
      DO 94 I=1,5
      I2=I*8
      I1=I2-7
      READ(54,113,END=3110)(OCUS(IK,J),IK=I1,I2),KYR5,MO5,KDY5
   94 CONTINUE
  224 DO 95 J=1,104
      DO 95 I=1,5
      I2=I*8
      I1=I2-7
      READ(54,113,END=3110)(OCVS(IK,J),IK=I1,I2),KYR5,MO5,KDY5
   95 CONTINUE
  225 CONTINUE
      IF(NDYPL.EQ.1) DY1=KDY5
      IF(NDYPL.EQ.1) AMO1=MO5
      IF(NDYPL.EQ.1) YR1=KYR5
      FDY=KDY5
      FMO=MO5
      FYR=KYR5
      CALL RTIME(12)
  113 FORMAT(8F8.2,3I3,A3)
```

```
C
C
      IF (KPLT.LT.NOPLT) GO TO 9001
C
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
C  DO PLOT NUMBER 6 (WIND+THERM+GEOS  OCEAN SURFACE CURRENTS-0/200M
C                    WITH NEAR COAST EFFECT VECTOR ROTATION
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
 3110 CONTINUE
C
C
C  COMPUTE AND SCALE DOWN ARROWS IN OCEAN CURRENT VECTOR PLOT
      FACTOR=.25
      DO 3954 I=1,NN2
      DO 3955 J=1,MM2
      CSPEED(I,J)=SQRT(OCUS(I,J)*OCUS(I,J)+OCVS(I,J)*OCVS(I,J))
      IF(LS3(I,J).EQ.0)OCUS(I,J)=0.0
      IF(LS3(I,J).EQ.0)OCVS(I,J)=0.0
      POCU(I,J)=OCUS(I,J)*FACTOR
      POCV(I,J)=OCVS(I,J)*FACTOR
 3955 CONTINUE
 3954 CONTINUE
C
C
C
      CALL PRINT1(CSPEED,11)
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
      IF (EVGRPT.EQ.1.0) GO TO 976
      DO 994 JR=1,104,2
      DO 994 IR=1,40
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  994 CONTINUE
      DO 993 IR=1,40,2
      DO 993 JR=1,104
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  993 CONTINUE
  976 CONTINUE
      DO 992 IR=1,40
      POCU(IR,104)=0.0
      POCV(IR,104)=0.0
  992 CONTINUE
      DO 991 JR=1,104
      POCU(40,JR)=0.0
      POCV(40,JR)=0.0
  991 CONTINUE
```

```
C
C
C
      CALL PLOT(1.,1.,-3)
      CALL DATFIX(POCU,OCUU)
      CALL DATFIX(POCV,OCVV)
      CALL NEWPEN(1)
      CALL VECTOR(OCUU,OCVV)
      CALL NEWPEN(2)
C
      IF(IPLSLP.EQ.1)CALL CONTOR(0,1,900.,1600.,8.,16.,16.,0,CSLP)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
C
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
      IFLBLR=1
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
C
C
C  PLOT LATITUDE LINES
      DO 5141 IPK=1,NCC
      CALL PLOT (PLO(IPK),PLA(IPK),IPEN(IPK))
 5141 CONTINUE
C
      CALL NEWPEN(2)
C
C  PLOT LONGITUDES/CCSQ/FIG FROM 510 (+20) TO 230 DEGREES
      CHSZ=0.307
      DO 5151 LO=1,7
      IF(LO.EQ.1) GO TO 5151
      XX=110.0+((LO-1)*20.0)
      XXL=XX
      NLA=LO*2
      YY=ALATSF(NLA-1)
      XX1=XX
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,3)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      YY=ALATSF(NLA)
      XX=XX1
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,2)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      IF(LO.EQ.2) GO TO 5151
      IF (LO.EQ.7) GO TO 5151
      XA=XX-(2.0*CHSZ)
      YA=YY - 0.5
      IF (LO.EQ.5) XXL=170.0
      IF (LO.EQ.6) XXL=150.0
      CALL NUMBER(XA,YA,CHSZ,XXL,0.0,-1)
```

```
      XAW=999.0
      YAW=999.0
      IF(LO.EQ.3.OR.LO.EQ.4)CALL SYMBOL(XAW,YAW,CHSZ,' W',0.0,2)
      IF(LO.GE.5.OR.LO.EQ.6)CALL SYMBOL(XAW,YAW,CHSZ,' E',0.0,2)
 5151 CONTINUE
C  LABEL LAT'S
      ALO=170.0
      DO 5176 NL=1,2
      ALA=30.0 + ((NL-1)*10)
      ALA1=ALA
      IF(NL.EQ.1) CLO=133.5
      IF(NL.EQ.2) CLO=123.0
      CALL CONVRT(CLO,ALA,IEND)
      WRITE(6,6072)CLO,CLA,ALA1
      CLA=ALA-(0.5*CHSZ)
      CLO=XLN+0.1
      CALL NUMBER(CLO,CLA,CHSZ,ALA1,0.0,-1)
      WRITE(6,6072)CLO,CLA,ALA1
      IF (NL.LT.3)CALL SYMBOL(999.0,999.0,CHSZ,' N',0.0,2)
 5176 CONTINUE
C
      XX=0.5
      YY=YYLEN-0.3
      HH=0.2
      CALL SYMBOL(XX,YY,HH,' ',0.0,1)
      IF(IPLSLP.EQ.1)CALL SYMBOL(XX,YY,HH,'SEA LEVEL PRESSURE (MB)
     '0.0,26)
      CALL NUMBER(999.0,999.0,HH,FYR,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FMO,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FDY,0.0,-1)
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
      XP=XLN*0.55
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,22HSURFACE OCEAN CURRENTS ,0.0,22)
      YP=YP-0.3
      XP=XP-0.45
      CALL SYMBOL(XP,YP,0.2,31H(WIND + THERM + GEOS COMPONENT),0.0,
      YP=YP-0.25
      CALL SYMBOL(XP,YP,0.2,'(X ',0.0,2)
      CALL NUMBER(999.0,999.0,0.2,WFAC,0.0,2)
      CALL SYMBOL(999.0,999.0,0.2,', X ',0.0,4)
      CALL NUMBER(999.0,999.0,0.2,TFAC,0.0,2)
      CALL SYMBOL(999.0,999.0,0.2,', X ',0.0,4)
      CALL NUMBER(999.0,999.0,0.2,DDFAC,0.0,2)
      CALL SYMBOL(999.0,999.0,0.2,')',0.0,1)
      XP=XP+1.0
      YP=YP-0.4
      UVAL=10.0*FACTOR
      CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YP+0.1
```

```
      CALL SYMBOL(XP,YP,0.2,10H10 CM/SEC  ,0.,10)
C
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL FIXER(IFCOP)
      CALL RTIME(14)
 9000 CONTINUE
C
C
C
C
 9001 CONTINUE
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
C  DO PLOT NUMBER 7 (PROGRESSIVE VECTORS)
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C
      IF (PROGV.LT.1.0) GO TO 11111
C  DO PLOT OF PROGRESSIVE VECTORS
C
      NDYPL=0
  966 CONTINUE
      NDYPL=NDYPL+1
C
C  READ FILES OF PROGRESSIVE VECTORS
      IJEND=152
      IJEND=24
      IJEND=66
      IJEND=33
      IJEND=1
      DO 3115 L=1,IJEND
      READ(60,4104,END=967)IDY9,KK,KYR,IMO,IDY,I,J,GU1(L),GV1(L),GL
     'O1,GU2(L),GV2(L),GLA2,GLO2,WFAC,TFAC,DDFAC
 4104 FORMAT(2I3,3I2,2I3,8F7.3,3F3.1)
      IF (IDY9.GT.2.AND.IDY9.LT.180) GO TO 4118
      WRITE(6,4105)IDY9,KK,KYR,IMO,IDY,I,J,GU1(L),GV1(L),GLA1,GLO1,
     'GU2(L),GV2(L),GLA2,GLO2,WFAC,TFAC,DDFAC
 4118 CONTINUE
C
C
C  HANDLE OFF MAP POINTS
      IF(GV1(L).GE.38.99)IOFFMP(L)=1
      IF(GV1(L).LE.1.01)IOFFMP(L)=1
      IF(GU1(L).GE.102.99)IOFFMP(L)=1
      IF(GU1(L).LE.1.01)IOFFMP(L)=1
      IF(GV2(L).GE.38.99)IOFFMP(L)=1
      IF(GV2(L).LE.1.01)IOFFMP(L)=1
      IF(GU2(L).GE.102.99)IOFFMP(L)=1
      IF(GU2(L).LE.1.01)IOFFMP(L)=1
C
      IF(IOFFMP(L).EQ.1) GO TO 3115
      IPLALL=1
      IPLALL=0
```

```
      IF(IDY9.GT.NDAYS) GO TO 967
C     DO 3127 NXK=16,96,8
C     IF(J.EQ.NXK) GO TO 3115
 3127 CONTINUE
      IF(IPLALL.EQ.1) GO TO 4129
      DO 4120 LK=1,IIEND
      IF (IDY9.GT.0.AND.IDY9.LT.180) GO TO 4119
C     WRITE(6,4108)LK,IIENO,I,J,IPV(LK),JPV(LK)
 4119 CONTINUE
      IF(I.EQ.IPV(LK).AND.J.EQ.JPV(LK)) GO TO 4129
 4108 FORMAT(' TEST I J  ',6I5)
 4120 CONTINUE
      GO TO 3115
 3125 IOTHER=0
      GO TO 3115
 4129 CONTINUE
C
      IF (IDY9.GT.2.AND.IDY9.LT.180) GO TO 4106
      WRITE(6,4105)IDY9,KK,KYR,IMO,IDY,I,J,GU1(L),GV1(L),GLA1,GLO1,
     'GU2(L),GV2(L),GLA2,GLO2,WFAC,TFAC,DDFAC
 4105 FORMAT (2I4,3I2,2I4,8F10.3,3F5.1)
 4106 CONTINUE
      X1=(GU1(L)-1.0)*(XLN/103.0)
      IF (IDY9.EQ.1) YR1=KYR
      IF (IDY9.EQ.1) MO1=IMO
      IF (IDY9.EQ.1) DY1=IDY
      FYR=KYR
      FMO=IMO
      FDY=IDY
      Y1=(YLN-1.0)-((GV1(L)-1.0)*((YLN-1.0)/39.0))
      X2=(GU2(L)-1.0)*(XLN/103.0)
      Y2=(YLN-1.0)-((GV2(L)-1.0)*((YLN-1.0)/39.0))
      X1=X1+1.0
      Y1=Y1+1.0
      X2=X2+1.0
      Y2=Y2+1.0
C     IF(LS3(I,J).EQ.0) GO TO 3119
C     IF(IDY9.EQ.1)CALL SPCSYM(X1,Y1,0.307,42,0.0,-1)
      IF(IDY9.EQ.1)CALL DOT(X1,Y1,0.050)
 3119 CONTINUE
C     IF(IDY9.LE.2)WRITE(6,6062)IDY9,L,X1,Y1,X2,Y2
C     WRITE(6,6062)IDY9,L,X1,Y1,X2,Y2
 6062 FORMAT(2I5,4F10.3)
C     CALL PVECTR(X1,Y1,X2,Y2,OCUS(I,J),OCVS(I,J))
      CALL PVECTR(X1,Y1,X2,Y2,2.0,2.0)
      IF(FDY.EQ.1.0)CALL NUMBER(X2,(Y2+0.1),CHSZ,FMO,0.0,-1)
 3115 CONTINUE
C
      IF (IDY9.GT.2.AND.IDY9.LT.175) GO TO 4117
C     WRITE(6,6032)NDYPL,IDY9
 6032 FORMAT(' PROGRESSIVE VECTOR PLOT DAY NO. ',2I5)
 4117 CONTINUE
      GO TO 966
  967 CONTINUE
C
```

```
C
C
C     CALL PRINT1(CSPEED,11)
C
C  ZERO OUT EVERY OTHER ROW IND COL.
C  TO REDUCE THE NUMBER OF ARROWS IN PLOT
      IF (EVGRPT.EQ.1.0) GO TO 776
      DO 884 JR=1,104,2
      DO 884 IR=1,40
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  884 CONTINUE
      DO 883 IR=1,40,2
      DO 883 JR=1,104
      POCU(IR,JR)=0.0
      POCV(IR,JR)=0.0
  883 CONTINUE
      DO 882 IR=1,40
      POCU(IR,104)=0.0
      POCV(IR,104)=0.0
  882 CONTINUE
      DO 881 JR=1,104
      POCU(40,JR)=0.0
      POCV(40,JR)=0.0
  881 CONTINUE
  776 CONTINUE
C
C
C
      CALL PLOT(1.,1.,-3)
C     CALL DATFIX(POCU,OCUU)
C     CALL DATFIX(POCV,OCVV)
      CALL NEWPEN(1)
C     CALL VECTOR(OCUU,OCVV)
      CALL NEWPEN(2)
C
C     CALL CONTOR(0,1,900.,1600.,8.,16.,16.,1,CSLP)
      CALL GRIDER(102,38)
C     CALL GRIDER(0,0)
C
C
C  PLOT LATITUDE LINES
      DO 1141 IPK=1,NCC
      CALL PLOT (PLO(IPK),PLA(IPK),IPEN(IPK))
 1141 CONTINUE
C
      CALL NEWPEN(2)
C
C  PLOT LONGITUDES/CCSQ/FIG FROM 110 (+20) TO 230 DEGREES
      CHSZ=0.307
      DO 1161 LO=1,7
      IF(LO.EQ.1) GO TO 1161
      XX=110.0+((LO-1)*20.0)
      XXL=XX
      NLA=LO*2
```

```
      YY=ALATSF(NLA-1)
      XX1=XX
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,3)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
      YY=ALATSF(NLA)
      XX=XX1
      YY1=YY
      CALL CONVRT(XX,YY,IEND)
      CALL PLOT (XX,YY,2)
      WRITE(6,4007)LO,NLA,IEND,XX1,YY1,XX,YY
 4007 FORMAT(3I5,4F10.4)
      IF(LO.EQ.2) GO TO 1161
      IF (LO.EQ.7) GO TO 1161
      XA=XX-(2.0*CHSZ)
      YA=YY - 0.5
      IF (LO.EQ.5) XXL=170.0
      IF (LO.EQ.6) XXL=150.0
      CALL NUMBER(XA,YA,CHSZ,XXL,0.0,-1)
      XAW=999.0
      YAW=999.0
      IF(LO.EQ.3.OR.LO.EQ.4)CALL SYMBOL(XAW,YAW,CHSZ,' W',0.0,2)
      IF(LO.GE.5.OR.LO.EQ.6)CALL SYMBOL(XAW,YAW,CHSZ,' E',0.0,2)
 1161 CONTINUE
C  LABEL LAT'S
      ALO=170.0
      DO 1176 NL=1,2
      ALA=30.0 + ((NL-1)*10)
      ALA1=ALA
      IF(NL.EQ.1) CLO=133.5
      IF(NL.EQ.2) CLO=123.0
      CALL CONVRT(CLO,ALA,IEND)
      WRITE(6,6072)CLO,CLA,ALA1
      CLA=ALA-(0.5*CHSZ)
      CLO=XLN+0.1
      CALL NUMBER(CLO,CLA,CHSZ,ALA1,0.0,-1)
      WRITE(6,6072)CLO,CLA,ALA1
 6072 FORMAT (' LABEL LATS   ',3F10.4)
      IF (NL.LT.3)CALL SYMBOL(999.0,999.0,CHSZ,' N',0.0,2)
 1176 CONTINUE
C
      CALL NUMMER
      CALL SYMBOL(0.,YYLEN,.2,'DAILY PROGRESSIVE VECTOR DISTANCE',0
      XX=2.0
      YY=YYLEN-0.3
      HH=0.2
      CALL SYMBOL(XX,YY,HH,'FROM   ',0.0,7)
      CALL NUMBER(999.0,999.0,HH,YR1,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,MO1,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,DY1,0.0,-1)
      YY=YY-0.3
      CALL SYMBOL(XX,YY,HH,'  TO   ',0.0,7)
```

```
      CALL NUMBER(999.0,999.0,HH,FYR,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FMO,0.0,-1)
      CALL SYMBOL(999.0,999.0,HH,' - ',0.0,3)
      CALL NUMBER(999.0,999.0,HH,FDY,0.0,-1)
C
C*  DRAW BATHYMETRY
C
      WRITE(6,9901)NRECS1,NRECS2,NRECS3,NRECS4,(NDATA(NN),NN=1,4)
      IFLBLR=1
      CALL BATHY(0,1,IFLBLR,BATH,NDATA,DEPTH)
C
C
C  PLOT SCALE ARROW (1 INCH = 10CM/SEC)
      XP=XLN*0.72
      YP=YYLEN-0.7
      UVAL=10.0*FACTOR
C     CALL AVECTR(XP,YP,UVAL,0.0)
      XP=XP-0.4
      YP=YP+0.1
C     CALL SYMBOL(XP,YP,0.2,10H10 CM/SEC  ,0.,10)
      XP=XLN*0.55
      YP=YYLEN
      CALL SYMBOL(XP,YP,0.2,22HSURFACE OCEAN CURRENTS ,0.0,22)
      YP=YP-0.3
      XP=XP-0.45
      CALL SYMBOL(XP,YP,0.2,31H(WIND + THERM + GEOS COMPONENT),0.0,
      YP=YP-0.25
      CALL SYMBOL(XP,YP,0.2,'(X ',0.0,2)
      CALL NUMBER(999.0,999.0,0.2,WFAC,0.0,2)
      CALL SYMBOL(999.0,999.0,0.2,', X ',0.0,4)
      CALL NUMBER(999.0,999.0,0.2,TFAC,0.0,2)
      CALL SYMBOL(999.0,999.0,0.2,', X ',0.0,4)
      CALL NUMBER(999.0,999.0,0.2,DDFAC,0.0,2)
      CALL SYMBOL(999.0,999.0,0.2,')',0.0,1)
C
      CALL COAST(0,1,IC,C1,C2,C3)
      CALL FIXER(IFCOP)
C
C
C
11111 CONTINUE
C
C
C
C
C
      CALL PLOT(0.,0.,999)
      END
      SUBROUTINE AVECTR(XI,YJ,U,V)
      DATA VFAC/10.0/
      CALL PLOT(XI,YJ,3)
      IF(U.EQ.0. .AND. V.EQ.0.)GO TO 200
      X=XI+U/VFAC
      Y=YJ+V/VFAC
```

```
      CALL PLOT(X,Y,2)
      Z=.09/SQRT(U*U+V*V)
      SI=V*Z
      CO=U*Z
      XX=SI*.3
      YY=CO*.3
      CO=X-CO
      SI=Y-SI
      CALL PLOT(CO-XX,SI+YY,2)
      CALL PLOT(CO+XX,SI-YY,2)
      CALL PLOT(X,Y,2)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE PVECTR(X1,Y1,X2,Y2,U,V)
      DATA VFAC/10.0/
      CALL PLOT(X1,Y1,3)
C     IF(U.EQ.0. .AND. V.EQ.0.)GO TO 200
C     X=XI+U/VFAC
C     Y=YJ+V/VFAC
      CALL PLOT(X2,Y2,2)
      DX=X2-X1
      DY=Y1-Y2
      DL=SQRT((DX*DX)+(DY*DY))
C     DL2=DL*0.3
      Z=6.0*DL
      Z=9.0*DL
C     Z=4.0*DL
      IF (Z.GT.1.0) Z=1.0
      IF (Z.GT.0.2) Z=0.2
      SPEED=SQRT((U*U)+(V*V))
C     IF (SPEED.LT.2.0) GO TO 200
      SI=-DY*Z
      CO=DX*Z
      SI2=SI
      CO2=CO
C     IF (SI.GT.DL2) SI=DL2
C     IF (CO.GT.DL2) CO=DL2
C     IF (SI.LT.-DL2) SI=-DL2
C     IF (CO.LT.-DL2) CO=-DL2
C      XX=SI*0.3
C      YY=CO*0.3
      XX=SI*.5
      YY=CO*.5
      CO=X2-CO
      SI=Y2-SI
C     IF(U.EQ.10.0.AND.V.EQ.10.0)WRITE(6,222)X1,Y1,X2,Y2,DL,Z,SI,CO
C    'Y,CO2,SI2
  222 FORMAT(12F10.3/)
      CALL PLOT(CO-XX,SI+YY,2)
      CALL PLOT(CO+XX,SI-YY,2)
      CALL PLOT(X2,Y2,2)
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRIAFP(FLD,KM)
      COMMON/BLAND/AL(24,22)
      COMMON/LNDSEA/SEAA(40,104)
      DIMENSION LAB(6),LISARY(14),NAME1(3),NAME2(3),NAME3(3)
      DIMENSION BLON(11,11),BLAT(11,11),XII(11,11),YJJ(11,11)
      DIMENSION CLON(24,24),CLAT(24,24)
      DIMENSION XXXX(11,11),YYYY(11,11)
      DIMENSION IFISH(18)
      DIMENSION T1(24,24),SST(24,24,12),BOTT(24,24,12)
      DIMENSION IA(80)
      DIMENSION FLD(24,24),IFLD(24,24),PRF(24,24),IAF(24,24)
      DATA IAP/'  .  '/
      DATA IAZP/' 0.00'/
C
C
C  KM=0, FULL UNITS, NO DECIMAL PRINTED
C  KM=1, DIVIDE BY 100
C  KM=2, DIVIDE BY 1000
C  KM=3, DIVIDE BY 10
C  KM=4, DIVIDE BY 100, PRINT 1 DECIMAL PLACE
C  KM=5, FULL UNITS, PRINT 1 DECIMAL PLACE
C  KM=6, FULL UNITS, PRINT 2 DECIMAL PLACES
C  KM=10, F12.2 FORMAT ON GRIDS2 DISK
C
      IF (IFIRST.EQ.88)  GO TO 9998
C
C
      NOUT=24
      MOUT=24
C
C  GET RID OF ZERO IN PRINT (SET UP A-FORMAT ARRAY)
      REWIND 40
      DO 200 I=1,NOUT
      DO 200 J=1,MOUT
      WRITE (40,201) I,J,FLD(I,J)
  200 CONTINUE
  201 FORMAT (2I5,F5.2)
      REWIND 40
      DO 205 I=1,NOUT
      DO 205 J=1,MOUT
      READ (40,202) I,J,IAF(I,J)
      IF (IAF(I,J).EQ.IAZP)  IAF(I,J)=IAP
  205 CONTINUE
  202 FORMAT (2I5,A5)
C
C
C
C
 9998 CONTINUE
      IF (KM-1)  1,2,3
    3 IF (KM-3)  2,2,6
    6 IF (KM-5)  7,8,999
    1 PRINT 10
C  10 FORMAT (/,5X,10HORIG.UNITS)
   10 FORMAT (1H )
```

```
      PRINT 500,(N,N=1,24)
  500 FORMAT (/,5X,25I5)
      PRINT 511,(N,(FLD(N,M),M=1,24),N=1,24)
  511 FORMAT (/I3,2X,24F5.0)
      GO TO 100
    7 PRINT 11
   11 FORMAT (/,5X,17HUNITS DIV. BY 100)
      PRINT 500,(N,N=1,24)
      DO 12 N=1,24
      DO 12 M=1,24
      PRF(N,M)=FLD(N,M)/100.
   12 CONTINUE
      PRINT 512,(N,(PRF(N,M),M=1,24),N=1,24)
  512 FORMAT (/I3,2X,24F5.1)
      GO TO 100
    8 PRINT 10
      PRINT 500,(N,N=1,24)
      PRINT 512,(N,(FLD(N,M),M=1,24),N=1,24)
      GO TO 100
  999 IF (KM.NE.10) GO TO 9
      IDSK=71
      CALL RDGRID(FLD)
      GO TO 100
    9 PRINT 10
      PRINT 500,(N,N=1,24)
      WRITE (6,7083)
 7083 FORMAT (4X,124H***************************************************
     '******************************************************************
     '*******        )
      DO 591 N=1,24
      WRITE (6,1265) (AL(N,LL),LL=1,22)
 1265 FORMAT (22A6)
      PRINT 513,(N,(IAF(N,M),M=1,24))
      PRINT 525,(SEAA(N,M),M=1,24)
  525 FORMAT (1H+,5X,24A5)
  591 CONTINUE
      WRITE (6,7083)
  513 FORMAT(1H ,I2,3H **,24A5,2H**)
      GO TO 100
    2 IF (KM-2)  20,21,22
   20 DIL=100.
      PRINT 11
      GO TO 30
   21 DIL=1000.
      PRINT 24
   24 FORMAT(/,5X,18HUNITS DIV. BY 1000)
      GO TO 30
   22 DIL=10.
      PRINT 25
   25 FORMAT(/,5X,16HUNITS DIV. BY 10)
   30 DO 40 N=1,24
      DO 40 M=1,24
      IFLD(N,M)=FLD(N,M)/DIL
   40 CONTINUE
      PRINT 500,(N,N=1,24)
```

```
      PRINT 514,(N,(IFLD(N,M),M=1,24),N=1,24)
  514 FORMAT (/I3,2X,24I5)
  100 RETURN
      END
      SUBROUTINE SMTH(N,ALPHA,FL,FL2)
      DIMENSION TDEP(40,104),TML(40,104),FL(40,104),XF(40,104),
     1TCX(40,104),TCY(40,104),XTR2(40,104),XTR(40,104),TS(40,104),
     2SLP(40,104),WCX(40,104),WCY(40,104),IGD(40,104)
      DIMENSION FL2(40,104)
      BETA=1.0-ALPHA
      DO 2 I=1,40
      DO 2 J=1,50
    2 XF(I,J)=FL(I,J)
      DO 40 M=1,N
 10   DO 20 I=2,39
      DO 20 J=2,49
      XY=FL(I,J)
      A=FL(I,J+1)
      B=FL(I+1,J)
      C=FL(I,J-1)
      D=FL(I-1,J)
      IF (A.EQ.0.OR.B.EQ.0)     GO TO 20
      IF (C.EQ.0.OR.D.EQ.0)     GO TO 20
      XF(I,J) = (ALPHA*XY) + (BETA*((A+B+C+D)/4.0))
 20   CONTINUE
      DO 30 I=2,39
      DO 30 J=2,49
      FL(I,J)=XF(I,J)
 30   CONTINUE
 40   CONTINUE
      RETURN
      END
      SUBROUTINE PRINT1(P,NOX)
      DIMENSION P(40,104),IQ(50),PQ(40,104)
  199 FORMAT(1H1,5X,'LAND-SEA-GRID')
 200  FORMAT(1H1,5X,'TEMPERATURE AT DEPTH(CX10)')
 201  FORMAT(1H1,5X,'GEOSTROPHIC TOPOGRAPHY (DYNMX100)')
 202  FORMAT(1H1,5X,'SURFACE PRESSURE(MBX10)'//)
 203  FORMAT(1H1,5X,'SURFACE TEMPERATURE(CX10)')
  204 FORMAT(1H1,5X,'MODEL OUTPUT (CM/SEC) ')
  205 FORMAT(1H1,5X,'WEIGHTED AVERAGE OF TEMP 0-200M (C) X10.0')
  206 FORMAT (1H1,5X,'SURFACE WIND SPEED (M/SEC)'//)
  209 FORMAT(1H1,5X,'U -- COMPONENT OF WIND SPEED (M/SEC)'//)
  210 FORMAT(1H1,5X,'V -- COMPONENT OF WIND SPEED (M/SEC)'//)
  211 FORMAT(1H1,5X,'SURFACE OCEAN CURRENT SPEED (CM/SEC)'//)
  220 FORMAT (1X,26I5)
 212  FORMAT(4X,18(1X,F6.1))
      KR=6
C     DO 500 IP=1,2
      GO TO (10,20,30,40,42,44,70,80,90,100,110)NOX
 10   WRITE(KR,200)
      DO 11 I=1,40     ; DO 11 J=1,50
   11 PQ(I,J)=P(I,J)*10.0
      GO TO 50
 20   WRITE(KR,201)
```

```
      DO 21 I=1,40     ;  DO 21 J=1,50
   21 PQ(I,J)=P(I,J)*100.0
      GO TO 50
 30   WRITE(KR,202)
      DO 31 I=1,40     ;  DO 31 J=1,50
   31 PQ(I,J)=(P(I,J)-1000)*10.0
      GO TO 50
 40   WRITE(KR,203)
      DO 41 I=1,40     ;  DO 41 J=1,50
   41 PQ(I,J)=P(I,J)*10.0
      GO TO 50
   42 WRITE(KR,204)
      DO 43 I=1,40
      DO 43 J=1,50
   43 PQ(I,J)=P(I,J)
      GO TO 50
   44 WRITE(KR,205)
      DO 45 I=1,40
      DO 45 J=1,50
   45 PQ(I,J)=(P(I,J))*10.0
      GO TO 50
   70 WRITE (KR,199)
      DO 71 I=1,40
      DO 71 J=1,50
   71 PQ(I,J)=P(I,J)
      GO TO 50
   80 WRITE (KR,206)
      DO 81 I=1,40
      DO 81 J=1,50
   81 PQ(I,J)=P(I,J)
      GO TO 50
   90 WRITE(KR,209)
      DO 91 I=1,40
      DO 91 J=1,50
   91 PQ(I,J)=P(I,J)
      GO TO 50
  100 WRITE(KR,210)
      DO 101 I=1,40
      DO 101 J=1,50
  101 PQ(I,J)=P(I,J)
      GO TO 50
  110 WRITE(KR,211)
      DO 111 I=1,40
      DO 111 J=1,50
  111 PQ(I,J)=P(I,J)
C
C
   50 CONTINUE
C
C  PRINT LEFT 25 COLS. THEN PRINT RIGHT 25 COLS.
      DO 58 KJ=1,2
      J2=KJ*25
      J1=J2-24
      IF(NOX.EQ.3.AND.KJ.EQ.2) WRITE(KR,202)
      IF(NOX.EQ.8.AND.KJ.EQ.2) WRITE(KR,206)
```

```
      IF(NOX.EQ.9.AND.KJ.EQ.2) WRITE(KR,209)
      IF(NOX.EQ.10.AND.KJ.EQ.2) WRITE(KR,210)
      IF(NOX.EQ.11.AND.KJ.EQ.2) WRITE(KR,211)
      DO 60 I=1,40
      DO 57 JJ=J1,J2
   57 IQ(JJ)=PQ(I,JJ)
      WRITE (KR,220)  I,(IQ(J),J=J1,J2)
   60 CONTINUE
   58 CONTINUE
  500 CONTINUE
      RETURN
      END
C***********************************************
C*        SUBROUTINE FASTCOAST
C*
C***********************************************
      SUBROUTINE FASTCO(IFAST,ISKIP,NREC)
      NREC=0
      NIE=0
      NSKIP=0
    1 READ(70,107,END=49) XC,YC,IPEN
      NSKIP=NSKIP+1
      IF (IPEN.EQ.3.OR.NSKIP.EQ.ISKIP) GO TO 2
      GO TO 1
    2 CONTINUE
      NSKIP=0
  107 FORMAT (2F10.4,I10)
C     XC=-XC
      CALL CONVRT(XC,YC,IE)
      IF (IE.GT.0) NIE=NIE+1
      IF (NIE.GT.0) IPEN=3
      IF (IE.EQ.0) NIE=0
      IF (NIE.GT.2) GO TO 1
      IF (IE.GT.0) IPEN=3
      IF (NREC.EQ.0) IPEN=3
      CALL PLOT(XC,YC,IPEN)
      NREC=NREC+1
      WRITE (71,108) XC,YC,IPEN
  108 FORMAT (2F10.4,I3)
      GO TO 1
   49 CLOSE(71,DISP=CRUNCH)
      WRITE(6,110) NREC
      RETURN
  110 FORMAT (1H1,'NUMBER OF FASTCOAST RECORDS =',I10)
      END
C*****************************************************
C*
C*           SUBROUTINE CSET
C*
C*****************************************************
      SUBROUTINE CSET(NFILE,IC,C1,C2,C3)
      DIMENSION C1(25000),C2(25000),C3(25000)
      N1=1
      N2=0
    1 READ(NFILE,100,END=999)C1(N1),C2(N1),C3(N1)
```

```
  100 FORMAT(2F10.4,I3)
      N1=N1+1
      N2=N2+1
      GO TO 1
  999 CONTINUE
      IC=N2
      RETURN
      END
      SUBROUTINE ANGLD(X,Y,AAA)
      IQ1=0
      IQ2=0
      IQ3=0
      IQ4=0
      IF (X.GE.0.0.AND.Y.GE.0.0) IQ1=1
      IF (X.LT.0.0.AND.Y.GE.0.0) IQ2=1
      IF (X.LT.0.0.AND.Y.LT.0.0) IQ3=1
      IF (X.GE.0.0.AND.Y.LT.0.0) IQ4=1
      PI=3.141596
      RAD=PI/180.0
      IF (X.GE.0.0.AND.X.LT.0.000001) X=0.000001
      IF (X.LT.0.0.AND.X.GT.-0.000001) X=-0.000001
      AA=Y/X
      IF (IQ1.EQ.1) ANG=ATAN(AA)
      IF (IQ1.EQ.1) GO TO 50
      IF (IQ2.EQ.1) ANG=PI + ATAN(AA)
      IF (IQ2.EQ.1) GO TO 50
      IF (IQ3.EQ.1) ANG=PI + ATAN(AA)
      IF (IQ3.EQ.1) GO TO 50
      IF (IQ4.EQ.1) ANG=(2.0*PI) + ATAN(AA)
   50 CONTINUE
      AAA=ANG/RAD
C     WRITE(6,101)IQ1,IQ2,IQ3,IQ4,Y,X,AA,ANG,AAA
  101 FORMAT(4I4,5F11.5)
      RETURN
      END
      SUBROUTINE OUTPT(OUT,NFILE,KYR,MO,KDY,H)
      DIMENSION OUT(40,104),H(1)
      DO 60 J=1,104
      DO 60 I=1,5
      I2=I*8
      I1=I2-7
      WRITE (NFILE,100)  (OUT(IK,J),IK=I1,I2),KYR,MO,KDY,H
   60 CONTINUE
  100 FORMAT (8F8.2,3I3,A3)
      ENDFILE 17
      RETURN
      END
      SUBROUTINE RTIME(N)
      IPC=TIME(2)
      IO=TIME(3)
      WRITE(6,120) N,IPC,IO
  120 FORMAT (' TIME NO.',I4,5X,'CPU TIME =',I10,5X,'I/O TIME =',I1
      RETURN
      END
C***********************************************
```

```
C*       SUBROUTINE STGRID                     *
C*                                             *
C* SETS UP COMMON BLOCKS TO BE USED BY ROUTINES*
C* NN,MM ARE NUMBER ROWS, NUMBER COLS IN GRID  *
C* ALA,ALO ARE LON,LAT ARRAYS OF GRID POINTS   *
C* ILS IS THE LAND-SEA ARRAY                   *
C* XL,YL ARE THE WIDTH AND HEIGHT OF THE PLOT  *
C*********************************************
      SUBROUTINE STGRID(NN,MM,ALA,ALO,ILS,XL,YL)
      DIMENSION ALA(40,104),ALO(40,104),ILS(40,104)
      DIMENSION SLAT(40,104),SLON(40,104)
      COMMON/GRDCOM/CFACT,M,N,NM,XI(105),YJ(41)
      COMMON/SIZES/XLEN,YLEN,YYLEN,SIZ1H,SIZ1V,SIZ2H,SIZ2V,
     &SIZ3H,SIZ3V,SIZ4H,SIZ4V
      COMMON/LSEA/LS(40,104)
      COMMON/LATLON/ALAT(40,104),ALON(40,104)
      COMMON/SSYM/ISYM(5)
      M=MM
      N=NN
      NM=N*M
      XLEN=XL
      YLEN=YL
      YYLEN=YLEN-1.0
      SIZ1H=0.198
      SIZ1V=0.307
      SIZ2H=0.181
      SIZ2V=0.290
      SIZ3H=0.120
      SIZ3V=0.185
      SIZ4H=0.110
      SIZ4V=0.168
      DO 10 I=1,N
      DO 10 J=1,M
      ALAT(I,J)=ALA(I,J)
      ALON(I,J)=ALO(I,J)
   10 LS(I,J)=ILS(I,J)
  930 FORMAT (I5,15F8.3)
      DO 60 I=1,N
   60 CONTINUE
      ISYM(1)=40
      ISYM(2)=44
      ISYM(3)=41
      ISYM(4)=42
      ISYM(5)=32
      DO 20 I=1,N
      DO 21 J=1,M
      IM1=(N+1)-I
      IF (J.GT.1.AND.I.LT.N) GO TO 21
      SLON(I,J)=ALON(I,J)
      SLAT(I,J)=ALAT(I,J)
  944 FORMAT (4I5,2F10.4,I10)
      CALL CONVRT(SLON(I,J),SLAT(I,J),IE)
      YJ(IM1)=SLAT(I,J)
      XI(J)=SLON(I,J)
   21 CONTINUE
```

```
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SPLINE (X,Y,IOK,VAL,DATA)
      DIMENSION A(4),AIJ(4)
      DIMENSION DATA(40,104)
C
C
      IOK=0
C  DO 4 PT NATURAL SPLINE INTERPOLATION FOR VALUES BETWEEN GRID PTS
      IA=Y
      JA=X
C  LIMIT CALCULATIONS WITHIN 1 INTERIOR POINT OF LEFT AND BOTTOM
C  LIMIT CALCULATIONS WITNIN 2 INTERIOR PTS OF RIGHT AND TOP
      IF (IA.LT.2.OR.JA.LT.2)   GO TO 998
      IF (IA.GT.38.OR.JA.GT.62)  GO TO 998
      DX=1.0
      X1=JA
      Y1=IA
      XP1=X1+1.0
      YP1=Y1+1.0
      DO 105 K=1,4
      JJ=JA+(K-2)
      DO 106 L=1,4
      II=IA+(L-2)
      AIJ(L)=DATA(II,JJ)
C     IF(XJ.LT.1.9)WRITE(6,450)XI,XJ,L,II,JJ,AIJ(L)
  106 CONTINUE
C  VERTICAL INTERPOLATION
      GPP=(AIJ(3)-(2.0*AIJ(2))+AIJ(1))/(DX*DX)
      GPPP1=(AIJ(4)-(2.0*AIJ(3))+AIJ(2))/(DX*DX)
      G1=((GPP/6.0)*((((YP1-Y)**(3.0))/DX)-(DX*(YP1-Y))))
      G2=((GPPP1/6.0)*((((Y-Y1)**(3.0))/DX)-(DX*(Y-Y1))))
      G3=(AIJ(2)*((YP1-Y)/DX))
      G4=(AIJ(3)*((Y-Y1)/DX))
      G=G1+G2+G3+G4
C     IF(G.LT.20.0)WRITE (66,9412) N,M,K,NK,GPP,GPPP1,G1,G2,G3,G4,G
      A(K)=G
  105 CONTINUE
C  HORIZONTAL INTERPOLATION
      GPP=(A(3)-(2.0*A(2))+A(1))/(DX*DX)
      GPPP1=(A(4)-(2.0*A(3))+A(2))/(DX*DX)
      G1=((GPP/6.0)*((((XP1-X)**(3.0))/DX)-(DX*(XP1-X))))
      G2=((GPPP1/6.0)*((((X-X1)**(3.0))/DX)-(DX*(X-X1))))
      G3=(A(2)*((XP1-X)/DX))
      G4=(A(3)*((X-X1)/DX))
      G=G1+G2+G3+G4
      VAL=G
  450 FORMAT(2F7.3,3I4,F7.3)
  451 FORMAT(2F7.3,I4,F7.3)
  452 FORMAT(3F7.3)
      GO TO 999
  998 IOK=1
      VAL=0.0
  999 RETURN
```

```
      END
      SUBROUTINE DOT(X0,Y0,SIZE)
      DR=0.02
      R=DR
      T=0.0
      DT=30.0*(3.1416/180.0)
      T2=360.0*(3.1416/180.0)
      CALL PLOT(X0,Y0,3)
    1 CONTINUE
      X=R*COS(T)
      Y=R*SIN(T)
      CALL PLOT((X0+X),(Y0+Y),2)
      T=T+DT
      IF (T.LE.T2)GO TO 1
      T=0.0
      R=R+DR
      IF (R.LT.SIZE) GO TO 1
      RETURN
      END
```