DOCUMENTATION AND USER'S GUIDE TO THE U.S. GEOLOGICAL SURVEY OILSPILL RISK ANALYSIS MODEL: OILSPILL TRAJECTORIES AND CALCULATION OF CONDITIONAL PROBABILITIES

By Kenneth J. Lanfear and William B. Samuels

U.S. GEOLOGICAL SURVEY

OPEN-FILE REPORT 81-316

1981

Contents

Page

Abstract -- 1

Introduction -- 2

Summary Description of SEDDS ---------------------- 2

Environmental data input -------------------------- 3

Currents --- 3

Winds -- 6

Targets -- 10

Land segments ------------------------------------ 12

Oilspill trajectory simulations ------------------ 12

Calculation of conditional probabilities ----------- 19

Summary and conclusions -------------------------- 21

References cited ---------------------------------- 24

Appendix, procedures and programs------------------ 25
CURVAL -- 28
CURPLOT --------------------------------------- 30
CURBAL -- 34
RAWWIND --------------------------------------- 37
LISTWIND -------------------------------------- 42
WINDTRAN -------------------------------------- 44
WINDZONE -------------------------------------- 53
DEFTGT -- 55
TGTPLOT --------------------------------------- 59
SEGMATRX -------------------------------------- 65
SPILL --- 67
HITPROB --------------------------------------- 85
LANDSEG --------------------------------------- 90

Illustrations

Figure Page

1. Flow chart for processing current data. --------- 4

2. Map showing the surface current field for December used in the Gulf of Mexico OCS Lease Sale 67 OSTA. -------------------------------- 5

3. Flow chart for processing wind data. ------------ 8

4. Map showing the ˈocation of four targets for Gulf of Mexico OCS Lease Sale 67. -------- 11

5. Map showing the division of the Mid-Atlantic shoreline into segments of approximately equal length. --------------- 13

6. Flow chart for simulating oilspill trajectories. -------------------------------- 14

7. Map showing the area launch sites for Mid-Atlantic OCS Lease Sale 59. -------------- 16

8. Map showing the transportation routes used as launch sites for Mid-Atlantic OCS Lease Sale 59. -------------------------- 17

9. Map showing two sample oilspill trajectories for Kodiak Island OCS Lease Sale 46. --------- 18

10. Flow chart for calculating conditional probabilities. ------------------------------ 20

Tables

Table Page

1. Sample output from program CURBAL. -------------- 7

2. Wind transition matrix for environmental data buoy 42002 located in the Gulf of Mexico. 9

3. Probabilities (expressed as percent chance) that an oilspill starting at a particular location will contact a certain target within 3 days (Mid-Atlantic OCS Lease Sale 59). ------------- 22

4. Probabilities (expressed as percent chance) that an oilspill starting at a particular location will contact a certain land segment within 3 days (Mid-Atlantic OCS Lease Sale 59). -------- 23

DOCUMENTATION AND USER'S GUIDE TO THE U.S. GEOLOGICAL SURVEY OILSPILL RISK ANALYSIS MODEL: OILSPILL TRAJECTORIES AND CALCULATION OF CONDITIONAL PROBABILITIES

By Kenneth J. Lanfear and William B. Samuels

Abstract

The trajectory portion of the U.S. Geological Survey Oilspill Risk Analysis model was developed to analyze the probability of an oilspill contacting coastal areas and environmental resources, given that a spill occurs. Computer programs are presented which analyze and format the oceanographic and meteorological input data to the model as well as the environmental resource data. The calculation of oilspill trajectories and the resulting conditional probabilities are also described.

INTRODUCTION

The oilspill trajectory analysis (OSTA) model was developed to help determine the oilspill risks of Outer Continental Shelf (OCS) oil development. The model analyzes the probability of oilspill occurrence for production and transportation of oil, the likely trajectories of spilled oil based on long-term oceanographic and meteorological observations, and the locations of environmental resources that may be vulnerable to oilspills. Prior descriptions of the model appear in Lanfear and others (1979) and Smith and others (1980).

Because of the large number of applications of the OSTA model (more than 15 since its development in 1976) and several hardware changes, modification of the model's procedures, programs, and subroutines are constantly being made. Thus, many programs previously documented have undergone substantial change. This paper is the second in a series documenting the latest programs used in the OSTA model. The first paper in this series (Lanfear and Nakassis, 1980) documented a system, Spatial Environmental Data Digitizing System (SEDDS), for representing large amounts of environmental data in digital form -- a key element of the OSTA model. This paper presents updated versions of the procedures, programs, and subroutines used to model oilspill trajectories and which use, as input, many of the files created by SEDDS. Readers of this paper are urged to become familiar with the SEDDS documentation before proceeding.

This paper does not emphasize modeling theory. It is intended to be a users' manual for operation of the model. A comprehensive discussion of the theory behind the OSTA model is contained in Smith and others (1980).

SUMMARY DESCRIPTION OF SEDDS

SEDDS was developed for analyzing large amounts of environmental data from a variety of sources. This system can combine data from maps of different scales and projections, place spatial data within a matrix of grid cells, and store the data in either of two storage notations using integers or individual bits, each designed for high-speed retrieval by mathematical models. The key concept in the design of SEDDS is the automatic conversion of all spatial data to a standard map projection coordinate system.

Most output files of SEDDS that are used in the OSTA model are in the form of a matrix, which corresponds to a grid system superimposed over the study area. Each grid cell is about 2 or 3 km on a side. Each element of the matrix contains information about one grid cell (for example, the presence or absence of land). Two basic types of notations are used. The first, numbered storage notation, simply assigns a number to each element. The second type, compact storage notation, uses each bit of the standard

32-bit IBM integer to represent some characteristic of the grid cell; up to 32 characteristics can be represented. Regardless of its type, a matrix is stored in 256 30- x 30-cell blocks to facilitate paging by high-speed searching algorithms.

All files produced by SEDDS for OSTA models are stored on an IBM 3330 disk[1] and prefaced with the name, OSTA.&STUDY.filename, where &STUDY is the identification of the particular analysis, and filename is the name of the particular file. SEDDS is also used to produce a base map of the study area, which is used for plotting all other maps drawn in preparing the OSTA model. This is on a file called OSRA.&STUDY.BASEMAP.

The programs and procedures described in the following section are listed in the appendix in the order of their appearence in the text. (Page numbers for these descriptions can be found in the Table of Contents).

ENVIRONMENTAL DATA INPUT

CURRENTS

Surface ocean currents are incorporated into the model in a deterministic manner in a finite element form. In most cases, geostrophic water velocities are input to the model, however, barotropic and baroclinic currents have been used. Data sources have included drift bottle studies, the output of diagnostic circulation models, and satellite observations. The current velocity data is input in meters per second and degrees. To apply this data to the model, the ocean surface is partitioned into polygons. The number of polygons is variable. By the digitizing and processing steps of SEDDS, the polygons are fit into the model's grid system in numbered storage notation; see Lanfear and Nakassis (1980, p. 14-15) for a description of the storage schemes used by SEDDS. Figure 1 shows the flow chart for processing the current data. Program CURVAL sets up the monthly surface velocity for each polygon. Input to the program is usually in the form of a card deck containing the monthly current speeds and directions for each polygon. Program CURVAL decomposes each velocity vector into its x and y components and places this information on a file called OSRA.&SALE.CURVAL (all file names in this paper use the naming conventions of IBM JCL).

Two quality control checks are applied to the current data. The first is a plot of the vector arrows for each month. Program CURPLOT reads the output of CURVAL (OSRA.&SALE.CURVAL), the current polygon centers (OSRA.&SALE.DEFPOLY.IDS.CURPOLY), and the base map file (OSRA.&SALE.BASEMAP) in order to write a complete set of plotting instructions on magnetic tape for a Gerber Model 4400 plotter. Each vector arrow, which is plotted at the center of its polygon, is examined for accuracy. The map in figure 2 shows the surface current field for December used for the Gulf of Mexico Lease Sale 67 OSTA (LaBelle and Lanfear, 1981). The second quality

[1] The use of brand names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

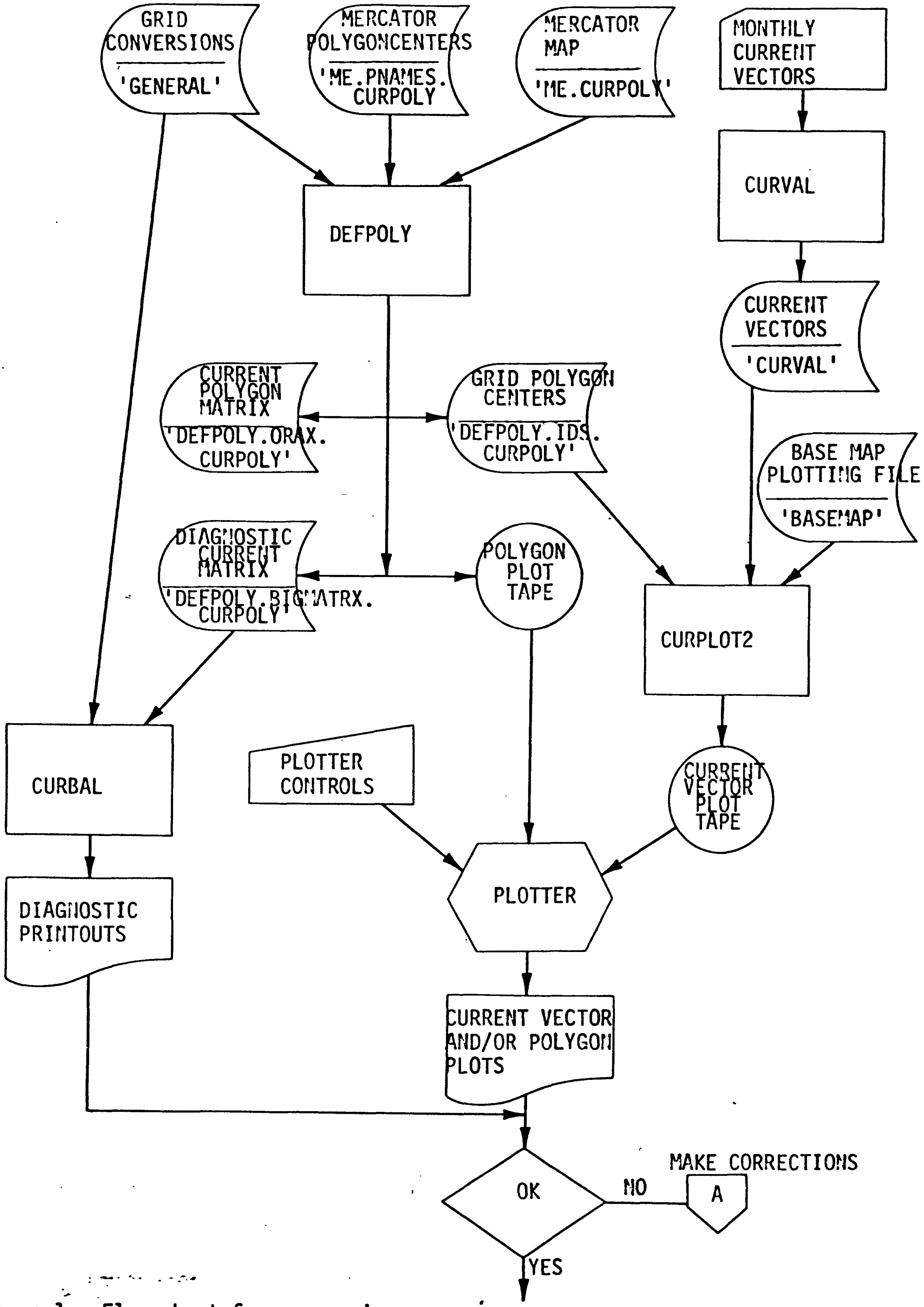


Figure 1.--Flow chart for processing current data.

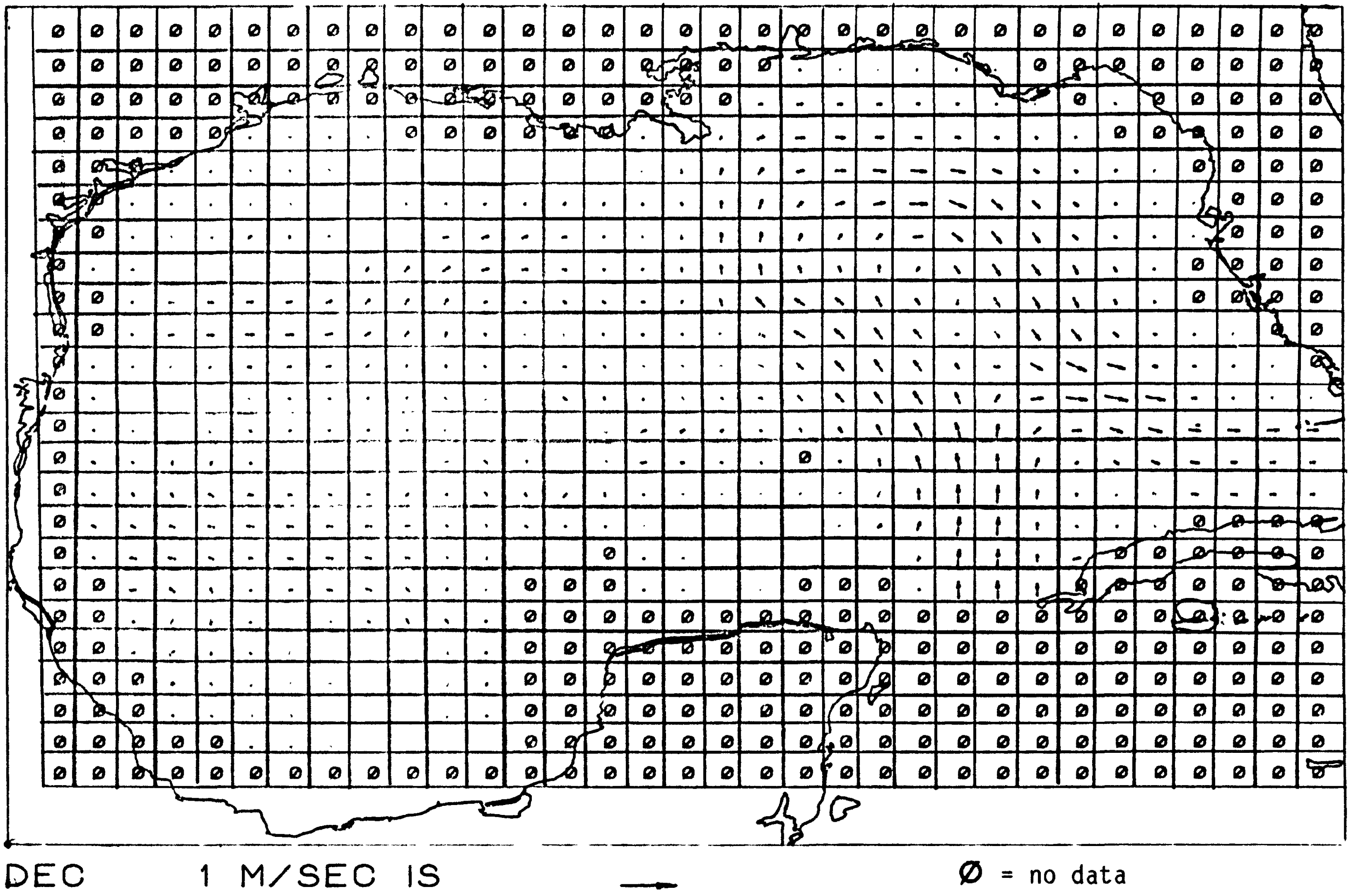


Figure 2.--Map showing the surface current field for December used in the Gulf of Mexico Lease Sale 67 OSTA.

control check is for continuity of the data. Program CURBAL checks current flow into and out of each polygon and lists, on a monthly basis, the difference between the inflow and outflow. Table 1 shows some sample output from CURBAL. If the magnitude of the difference between the inflow and outflow is large, for any given polygon, then the current vector for that polygon and its neighbors are checked for discrepancies.

WINDS

The variation in the wind is represented as a first-order Markov process, with 41 states corresponding to 8 wind directions by 5 wind speeds, plus the calm condition. A separate wind transition probability matrix is constructed for each of four seasons. These matrices are constructed from data obtained from National Oceanic and Atmospheric Administration weather tapes. Figure 3 shows the flow chart for processing the wind data.

Weather tapes may be produced in either of two formats, type TDF-14 or TDF-11. Reading a tape is cumbersome and expensive, and the tapes contain more meteorological data than is needed for calculating wind transition probabilities. Program RAWWIND reduces these data to a manageable size by reading the tapes for a station, selecting and decoding the data that are needed, and storing those data, unformatted, on a disk, in a file named OSRA.RAWWIND.&STATION, where &STATION is the identification code for the wind station. Note that wind file names do not contain a sale identifier; because a wind station may be used for more than one sale.

Over a period of years, the data collection procedures at a weather station may change. For example, the data may have been collected every 3 hours for several years, then the station downgraded to taking readings once or twice per day. Therefore, it is desirable to scan the station records to note those periods where the data are adequate for constructing wind transition probability matrices. Program LISTWIND reads the station record from the disk, and prepares a compact printout of the data. From this listing the period of useful records is determined.

With the station records on disk and the period of record chosen, the wind transition matrices can now be constructed. Program WINDTRAN reads the disk file, examines the wind speed and direction every 3 hours, and tabulates each transition. The resulting matrices are stored on a file, OSRA.WINDTRAN.&STATION. Table 2 shows the wind transition matrix formed for the winter season at environmental data buoy 42002, located in the Gulf of Mexico (LaBelle and Lanfear, 1981).

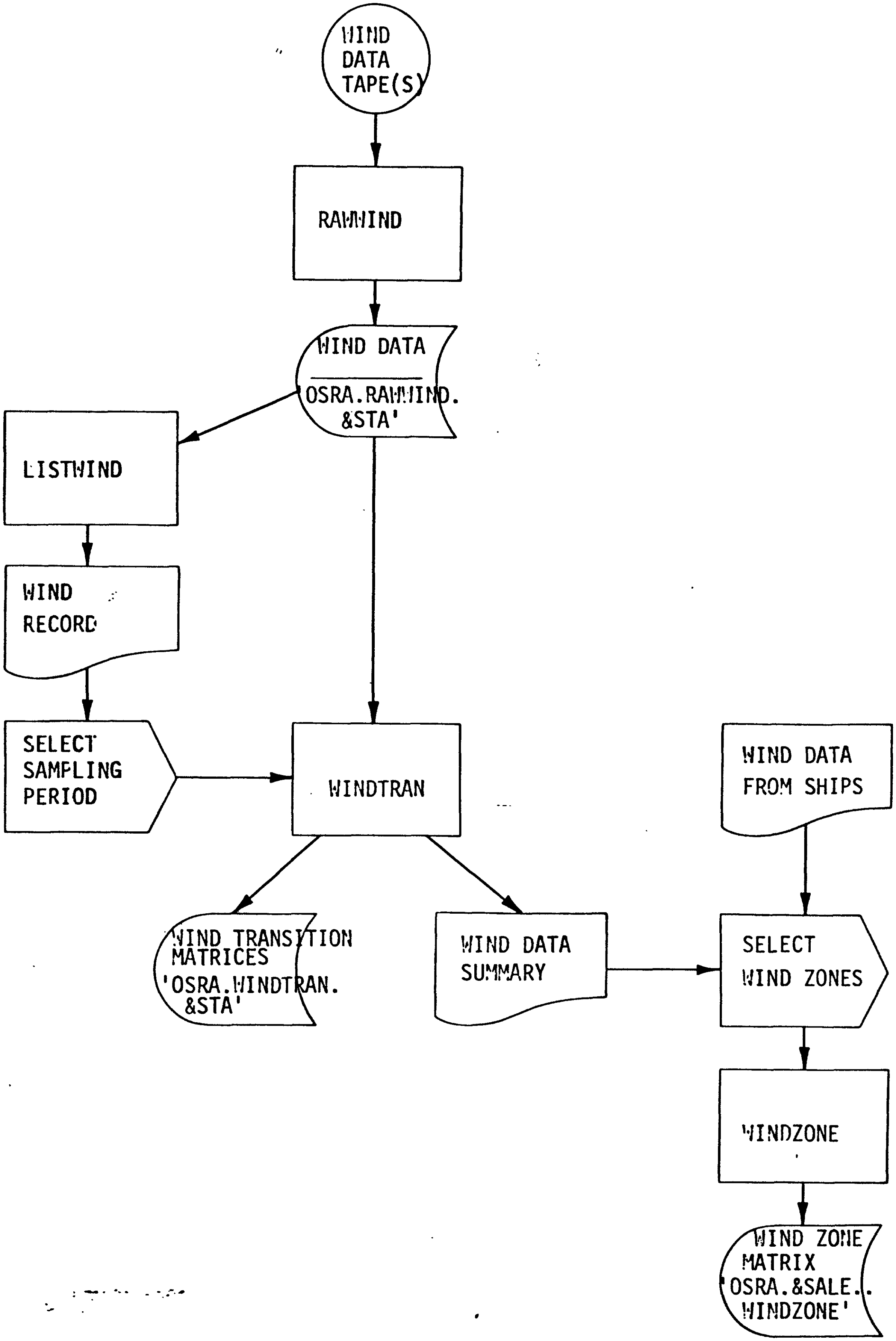
Since wind may vary over different parts of the study area, the model permits dividing the study area into a maximum of six wind zones. Each wind zone corresponds to a transition matrix for a different weather station. This operation is performed by program WINDZONE, and the resulting data file is called OSRA.&SALE.WINDZONE. Since high precision is not needed in

```
PROGRAM CURBAL

BALANCING CURRENT VECTORS FOR GULF67  CURRENTS

POLYGON    DEC   JAN   FEB   MAR   APR   MAY   JUN   JUL   AUG   SEP   OCT   NOV
    1      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
    2      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
    3      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
    4      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
    5      0.1  -0.3  -0.3  -0.3   0.0   0.0   0.0   0.0   0.0   0.0   0.1   0.1
    6      0.1  -0.2  -0.2  -0.2   0.0   0.0   0.0   0.0   0.0   0.0   0.1   0.1
    7      0.1  -0.2  -0.2  -0.2   0.1   0.1   0.1   0.0   0.0   0.0   0.1   0.1
    8     -0.0  -0.0  -0.0  -0.0   0.0   0.0   0.0   0.0   0.0   0.0  -0.0  -0.0
    9     -0.2   0.0   0.0   0.0  -0.1  -0.1  -0.1   0.0   0.0   0.0  -0.2  -0.2
   10     -0.2   0.0   0.0   0.0  -0.2  -0.2  -0.2  -0.3  -0.3  -0.3  -0.2  -0.2
   11     -0.1  -0.0  -0.0  -0.0  -0.2  -0.2  -0.2  -0.4  -0.4  -0.4  -0.1  -0.1
   12      0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.1  -0.1  -0.1   0.0   0.0
   13      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   14      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   15      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   16      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   17      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   18      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   19      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   20      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   21      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   22      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   23      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   24      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   25      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   26      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   27      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   28      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   29      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   30      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   31      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   32      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   33      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   34      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   35      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   36      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   37      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   38      0.1  -0.4  -0.4  -0.4  -0.1  -0.1  -0.1  -0.0  -0.0  -0.0   0.1   0.1
   39      0.2  -0.4  -0.4  -0.4  -0.1  -0.1  -0.1  -0.0  -0.0  -0.0   0.2   0.2
   40      0.1  -0.2  -0.2  -0.2   0.1   0.1   0.1   0.0   0.0   0.0   0.1   0.1
   41      0.1  -0.2  -0.2  -0.2   0.1   0.1   0.1   0.0   0.0   0.0   0.1   0.1
   42     -0.1  -0.0  -0.0  -0.0  -0.1  -0.1  -0.1   0.0   0.0   0.0  -0.1  -0.1
   43     -0.2   0.0   0.0   0.0  -0.2  -0.2  -0.2  -0.1  -0.1  -0.1  -0.2  -0.2
   44     -0.3   0.0   0.0   0.0  -0.2  -0.2  -0.2  -0.4  -0.4  -0.4  -0.3  -0.3
   45     -0.4   0.0   0.0   0.0  -0.2  -0.2  -0.2  -0.5  -0.5  -0.5  -0.4  -0.4
   46     -0.2  -0.0  -0.0  -0.0  -0.2  -0.2  -0.2  -0.3  -0.3  -0.3  -0.2  -0.2
   47      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   48      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   49      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   50      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   51      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   52      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   53      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   54      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   55      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   56      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
```

Table 1.--Sample output from program CURBAL.



Fiqure 3.--Flow chart for processing wind data.

```
 3 HOUR WIND TRANSITION MATRIX FOR WINTER AT 42002 BUOY Y7679
 USING A DRIFT ANGLE OF  20. DEGREES.

                                   NEXT WIND IS OUT OF ... AT SPEED ... IN PERCENT.
           CALM      N            NE            E             SE            S             SW            W             NW
LAST             510152025    510152025    510152025    510152025    510152025    510152025    510152025    510152025
FROM
DIR  SPD

CALM      0 20  13 3 0 3 0   3 0 0 5 0  10 3 0 0 0  13 3 0 0 0  13 0 0 0 0   3 3 0 0 0   3 0 3 0 0   3 0 0 3 0

N      5  11  2414 0 0 0  22 5 0 0 0   5 3 0 0 0   3 3 0 0 0   0 0 0 0 0   0 0 0 0 0   8 0 0 0 0   0 3 0 0 0
N     10   0  113010 2 0  1012 4 0 0   0 2 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   2 5 2 0 0
N     15   0   0254010 1   1 7 7 1 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 1 0 0 0   0 2 3 1 0
N     20   0   0 13147 5   0 0 6 5 1   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 2 1
N     25   0   0 0 53939   0 2 0 2 0   0 2 0 0 2   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 2 7

NE     5   9   5 3 2 0 0  33 7 2 0 0  2214 0 0 0   2 2 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
NE    10   0   1 2 1 0 0   65210 1 0   611 6 0 0   0 0 1 0 0   0 0 0 0 0   0 1 0 0 0   0 0 1 0 0   0 0 0 0 0
NE    15   0   0 1 4 1 0   0134511 1   0 616 1 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
NE    20   0   0 0 1 6 3   0 0175411   0 0 1 4 1   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
NE    25   0   0 0 0 0 5   0 0 03255   0 0 0 0 9   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

E      5  10   3 0 1 0 0  11 1 0 0 0  3020 6 0 0   9 4 0 0 0   1 0 3 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
E     10   0   1 0 1 0 0   1 3 1 0 0   65013 1 0   410 4 0 0   1 1 0 0 0   0 0 1 0 0   1 0 0 0 0   0 0 0 0 0
E     15   0   0 0 0 0 0   0 1 5 0 0   0105514 0   0 3 9 3 0   0 1 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
E     20   0   0 0 0 0 0   0 0 0 3 0   0 2234811   0 0 012 2   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
E     25   0   0 0 0 0 0   0 0 0 0 0   0 0 02146   0 0 01712   0 0 0 0 0   4 0 0 0 0   0 0 0 0 0   0 0 0 0 0

SE     5  10   0 0 0 0 0   2 0 0 0 0  12 2 0 0 0  2031 6 2 0   6 4 0 0 0   2 0 0 0 0   0 2 0 0 0   0 0 0 0 0
SE    10   1   0 0 0 0 0   1 0 0 0 0   3 8 2 1 0  104615 0 0   3 9 2 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
SE    15   0   0 0 0 0 0   0 0 0 0 0   0 1 1 0 0    12053 9 0   1 3 8 3 0   0 0 0 0 0   0 0 0 0 0   0 1 0 0 0
SE    20   0   0 0 0 0 0   0 0 0 0 0   3 0 1 3 0   0 11853 8   1 1 6 4 1   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0
SE    25   0   0 0 0 0 0   0 0 0 0 0   0 0 0 013   0 0 03333   0 0 7 7 7   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

S      5  10   0 0 0 3 0   7 0 3 0 0   0 0 0 0 0   717 0 0 0  1013 3 0 0  10 3 0 0 0   7 7 0 0 0   0 0 0 0 0
S     10   5   0 0 0 0 2   0 0 2 0 0   0 0 0 0 0   3 9 3 0 0   92816 2 0   2 6 5 0 0   3 2 0 0 0   0 0 2 2 2
S     15   0   0 0 1 0 0   0 0 0 0 0   0 0 0 0 0   0 0 9 1 0   21744 9 0   1 7 2 0 0   1 1 0 1 0   0 0 0 1 1
S     20   0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 3 0   3 33138 6   0 0 3 6 0   0 3 0 0 0   0 0 0 0 3
S     25   0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 08020   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

SW     5   0   0 9 9 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   9 9 0 0 0  10 0 0 0 0  27 0 0 0 0   9 9 0 0 0
SW    10   0   0 0 0 0 7   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 7 7 0 0   7 7 7 0 0  13 7 0 0 0   7 0 02013
SW    15   0   0 0 01111   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 011 0 0   0 01111 0    033 0 0 0   0 011 0 0
SW    20   0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 025 0   0 0 050 0   0 025 0 0
SW    25   0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0

W      5   9   0 4 4 4 4   4 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0  1713 0 4 0   9 9 4 013
W     10   0   0 0 0 0 5   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0  1019 5 0 0  19 5141410
W     15   0   0 017 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   017 0 0 0    01717 017   0 017 0 0
W     20   0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 029 0   0 0291429
W     25   0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 043   0 0 01443

NW     5  12  12 0 610 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0  12 0 0 0 0  18 618 0 0
NW    10   0  1221 6 3 3   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 6 3 0 0   926 9 3 0
NW    15   0   2 414 7 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 2 0 0   0184112 0
NW    20   0   0 2 810 2   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 2 2   0 2194410
NW    25   0   0 0 0 518   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 0   0 0 0 0 3   0 0 01860

OVERALL    2   2 4 6 5 2   3 4 4 3 1   3 7 7 3 1   2 7 7 4 1   1 3 4 1 0   0 1 0 0 0   1 1 0 0 0   1 2 2 3 3
```

Table 2.--Wind transition matrix for environmental data buoy 42002 located in the Gulf of Mexico.

defining these wind zones, program WINDZONE only requires that a wind zone be defined for 10- x 10-cell blocks of the matrix. Therefore, the data for WINDZONE may be entered directly from cards, rather than digitized.

TARGETS

A maximum of 32 categories of targets can be included in the oilspill trajectory analysis. Some examples of targets are: recreational beaches, seabird foraging areas, and fishing grounds. The entire coastline of the study area is always considered to be the last target in this array. The targets are digitized and processed using SEDDS, and stored in compact storage notation. Each bit of an IBM 32-bit integer indicates the presence or absence of each target in each cell of the model's grid system (480 x maximum 480 cells). The target locations are stored in a direct access file containing 256 records (16 x 16 blocks). Each record (block) contains 900 cells of the main matrix (that is, a 30- x 30-cell block of the main matrix).

Program DEFTGT defines the targets and creates target indicator matrices. Defining targets means assigning a name to each target (maximum of 60 characters) and establishing the monthly vulnerability for each target. Monthly vulnerabilities allow an oilspill contact to be recorded only during selected months. For example, a target such as migrating birds may be present in the study area only during the spring. Therefore, this target may be deemed vulnerable to a direct oilspill contact only during the spring months. In addition to defining the targets, program DEFTGT sets up indicator matrices for the targets and land segments. These matrices indicate the presence or absence of targets or land segments in each block of 30 x 30 cells. These indicator matrices increase the efficiency of the trajectory portion of the model by requiring checking for oilspill contacts only in those blocks containing targets or land segments.

Program TGTPLOT reads the direct access target matrix and writes a complete set of plotting instructions on magnetic tape for input to a Gerber Model 4400 plotter. The plotter tape is set up in such a way that a base map of the study area with lines of latitude and longitude is written on the first file mark. The second file mark contains a base map with grid lines plotted at 10-unit divisions. The target plots are contained on succeeding file marks. Figure 4 shows the location of four targets for the Gulf of Mexico Lease Sale 67 (LaBelle and Lanfear, 1981). The presence of a target in any cell of the model's grid system is indicated by plotting either a horizontal line, a vertical line, or a combination of both lines (see SUBROUTINES section of Lanfear and Nakassis, 1980). The scale and offset of each plot is variable. This feature allows for flexibility in the number of targets to be plotted on one base map and the size of each map. Each target is superimposed on a base map and checked against the original map

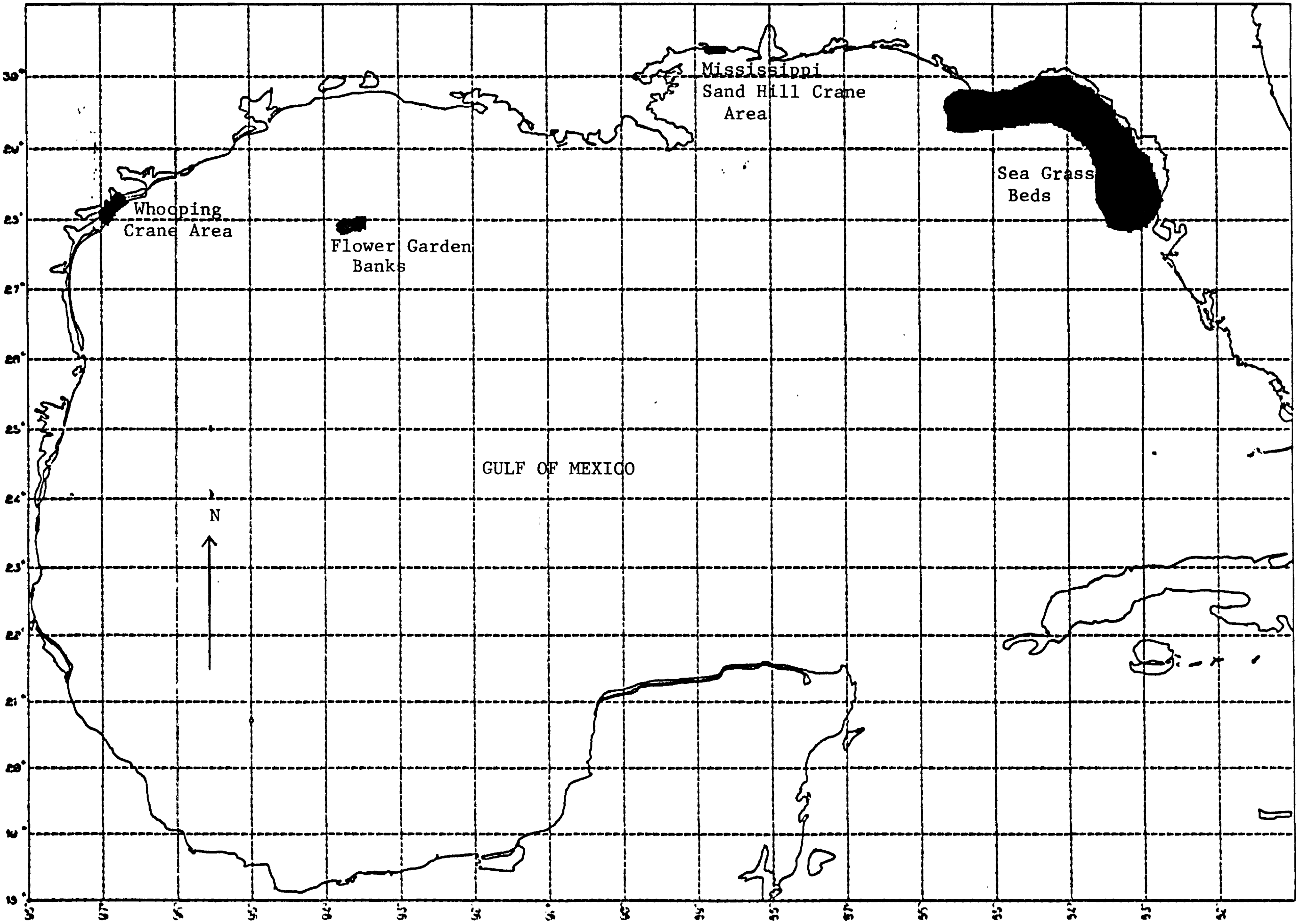


Figure 4.--Map showing the location of four targets for Gulf of Mexico OCS Lease Sale 67.

from which it was digitized.

LAND SEGMENTS

The coastline of a study area can be divided into two independent sets of land segments. A maximum of 100 segments per set is allowed. The land segments are digitized and processed as a set of polygons in numbered storage notation. The only criteria is that each coastline segment must fall within the boundary of its designated polygon. Usually, one set of segments is used to divide the coastline along equal divisions (approximately 30 km in length). The oilspill contact probabilities to this set of segments approximates the distribution function of oilspill contacts along the coastline. Figure 5 shows the divis on of the Mid-Atlantic shoreline into segments of approximately equal length (Samuels and Lanfear, 1980). A second set of segments may divide the coastline along political boundaries (for example, county boundaries), or into ecological zones. This allows for a different emphasis to be placed on the distribution of oilspill contacts. Program SEGMATRX merges the direct access files for each set into a single file by multiplying the identification numbers of set 1 by 1000, and adding them to those of set 2, so that a 2-byte integer is stored at each location in the model's grid system. The first two digits of this integer represent a land segment number for set 1, and the second two digits represent a land segment number for set 2.

OILSPILL TRAJECTORY SIMULATIONS

Oilspill trajectories are simulated by launching hypothetical oilspills from various sites which represent proposed and existing platforms, oil pipeline routes, and tanker routes. Typically 500 oilspills are launched per season in a Monte Carlo fashion from each launch site. Program SPILL, which simulates the trajectories, uses as input: (1) The general file (OSRA.&SALE.GENERAL.SPILL) created by program DEFTGT; this file contains the target names, monthly vulnerabilities, and indicator matrices; (2) the windzone matrix (OSRA.&SALE.WINDZONE); (3) the array of monthly current vectors (OSRA.&SALE.CURVAL); (4) the wind transition matrices for each wind station (OSRA.WINDTRAN.&WSTA); (5) the matrix of current polygons (OSRA.&SALE.DEFPOLY.DRAX.CURPOLY); (6) the target matrix (OSRA.&SALE.TARGETS); (7) the land segment matrix (OSRA.&SALE.SEGMATRX); and (8) the launch sites [point and line sources are read from a card deck, area sources are read from a disk file (OSRA.&SALE.LAUNCH)]. Figure 6 shows the flow chart for simulating oilspill trajectories.

The total number of launch sites and the number of trajectories simulated from each site is usually quite large. For most analyses, 20 to 100 launch sites are used, but, some studies have used as many as 300 launch sites. Typically 2,000 oilspills

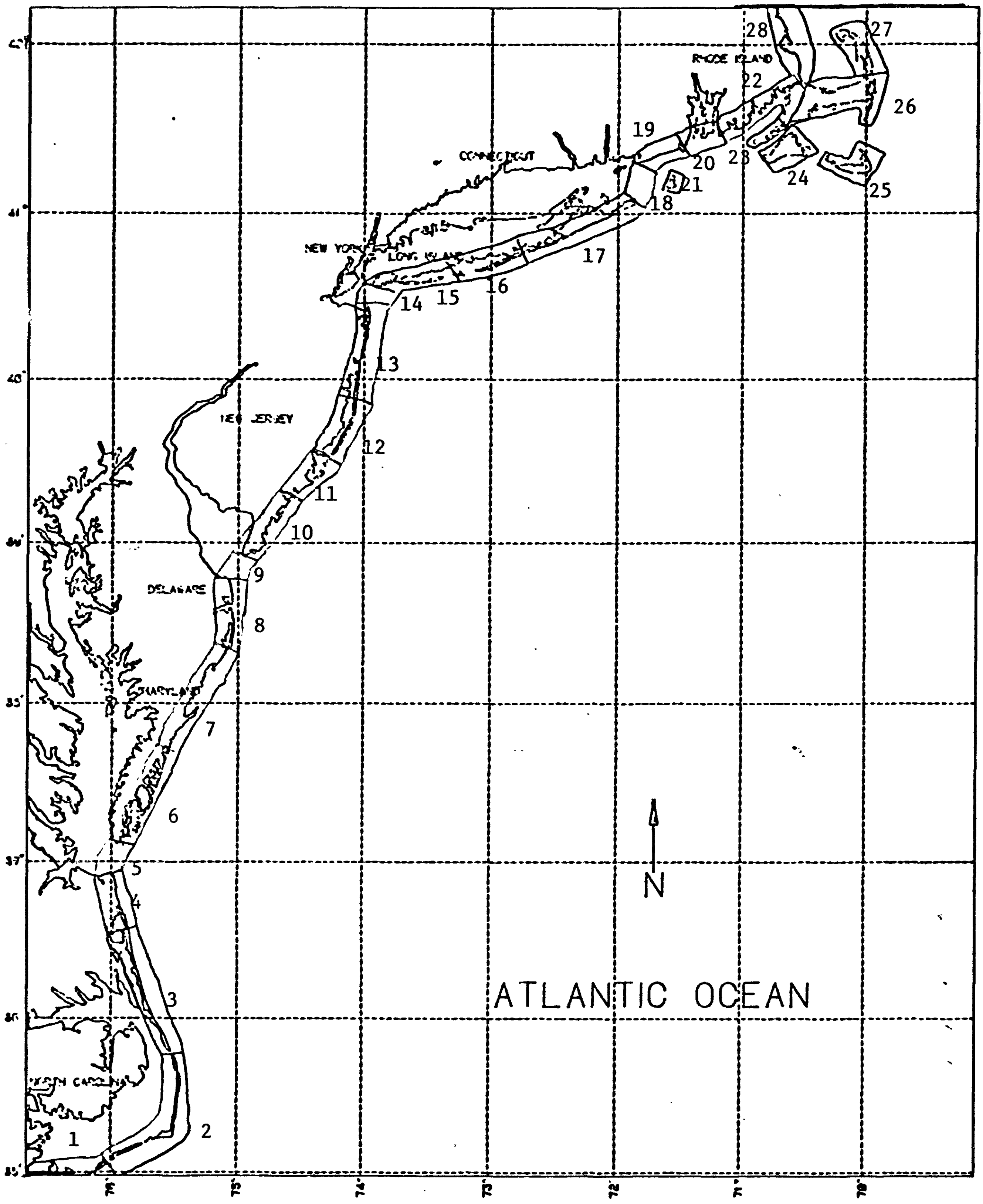
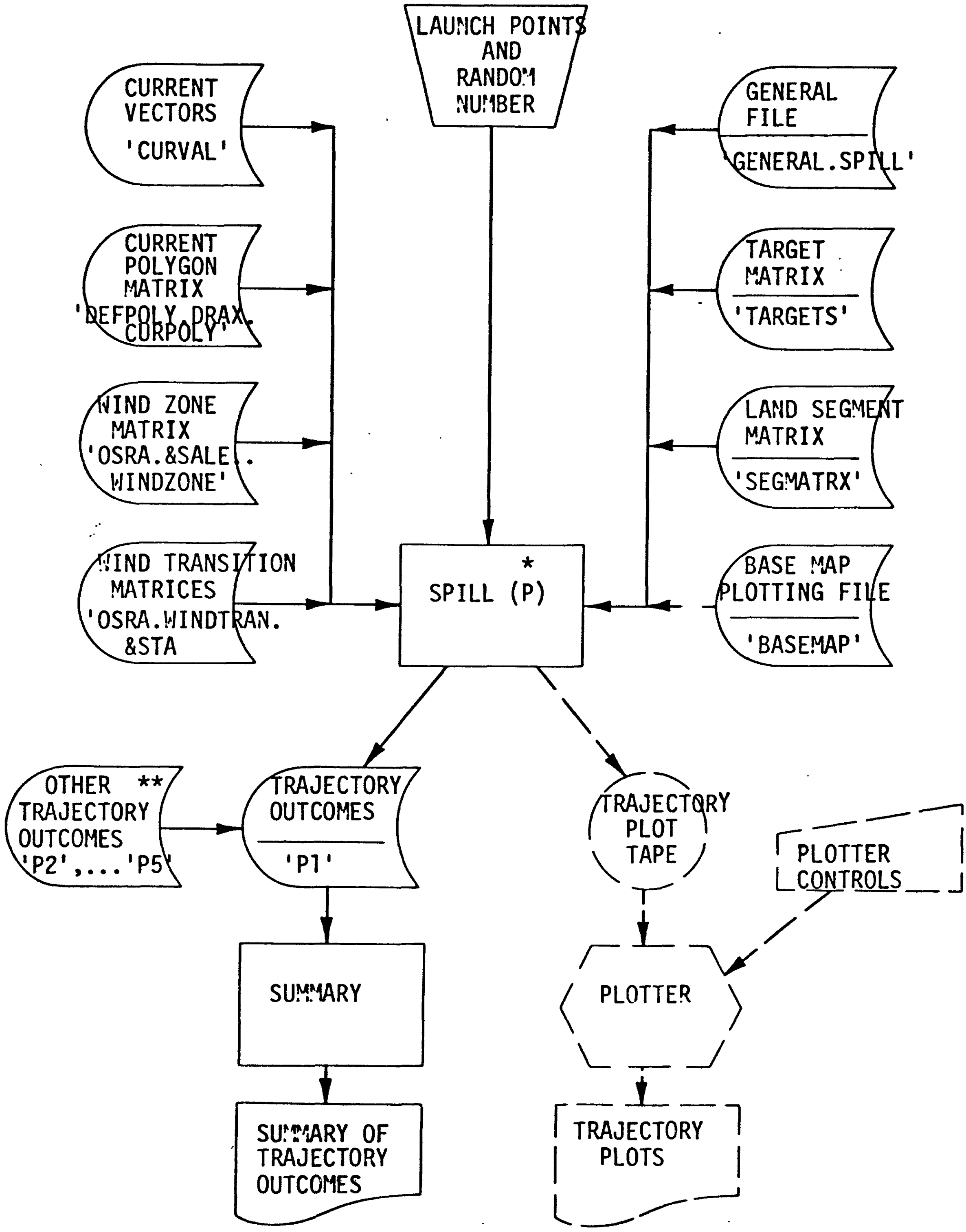


Figure 5.--Map showing the division of the Mid-Atlantic shoreline into segments of approximately equal length.


LAUNCH POINTS AND RANDOM NUMBER
CURRENT VECTORS 'CURVAL'
CURRENT POLYGON MATRIX 'DEFPOLY.DRAX. CURPOLY'
WIND ZONE MATRIX 'OSRA.&SALE.. WINDZONE'
WIND TRANSITION MATRICES 'OSRA.WINDTRAN. &STA
GENERAL FILE 'GENERAL.SPILL'
TARGET MATRIX 'TARGETS'
LAND SEGMENT MATRIX 'SEGMATRX'
BASE MAP PLOTTING FILE 'BASEMAP'
SPILL (P) *
OTHER ** TRAJECTORY OUTCOMES 'P2',...'P5'
TRAJECTORY OUTCOMES 'P1'
TRAJECTORY PLOT TAPE
PLOTTER CONTROLS
SUMMARY
PLOTTER
SUMMARY OF TRAJECTORY OUTCOMES
TRAJECTORY PLOTS

*SPILL P is the plotting version of SPILL
**SPILL is normally run in 5 sections.

Figure 6.--Flow chart for simulating oilspill trajectories.

are simulated from each launch site over a 1-year period. However, as few as 200 oilspills per year have been used in special cases; the number of spills is determined as a trade-off between computational speed to allow using a large number of trials to minimize the Monte-Carlo error, and accuracy in modeling winds and currents. Launch sites can be represented as points (all spills are launched from the same point), lines (spills are uniformly distributed along a line), or areas (spills are uniformly distributed in a designated area of any shape). Points might represent known platform locations, lines can represent pipeline and tanker routes, and areas best represent proposed and(or) existing lease tracts where the location of the platform could be anywhere within the lease tract. Figure 7 shows the area launch sites for Mid-Atlantic Lease Sale 59. Figure 8 shows the transportation routes used as launch sites for Mid-Atlantic Lease Sale 59 OSTA (Samuels and Lanfear, 1980).

The first step in simulating an oilspill from a particular launch site is to identify the starting location of the spill in the model's grid system and to load into core the starting 30 x 30 block of cells. Subroutine NEWBLK (contained in the coding of SPILL) loads the proper current and target blocks. This subroutine is designed to reduce the number of disk accesses necessary to load current and target matrices. Up to 32 current matrices and 16 target matrices can be stored in core. Those selected are the ones that were called most recently.

By a random process, a day within a particular season is selected and the wind transition matrix is initialized to the steady-state wind vector. Typically, 3-hour time steps are used for the trajectories. At each time step the wind transition matrix is randomly sampled for a new wind speed and direction. Oilspill movement caused by wind is taken as 3.5 percent of the wind speed. A 20 degree clockwise rotation (Northern Hemisphere) is applied to the wind direction to account for Coriolis effects. A current vector is also selected on the basis of the location of the spill and the simulated month. The oilspill movement is the vector sum of the wind and current vectors. Figure 9 shows two sample oilspill trajectories for the Kodiak Island, Alaska, OCS Lease Sale 46 (Samuels and others, 1980). In certain OCS areas, trajectory movement algorithms of other models have been used to simulate oilspill movement (LaBelle and others, 1980). These models are highy specific for a particular area, usually a semi-enclosed body of water, and are more appropriate for those areas than the general oilspill movement algorithm of the OSTA model.

Each grid cell that the spill passes through is checked using logical comparisons (see Lanfear and Nakassis, 1980) for the presence or absence of targets. If a target has been deemed vulnerable during the month in which it was contacted by an oilspill, then a hit is recorded to that target. Only the first contact to each target is recorded; subsequent contacts by the same spill are disregarded. If target land was contacted, then the proper land segment block is loaded into core by subroutine NEWSEG

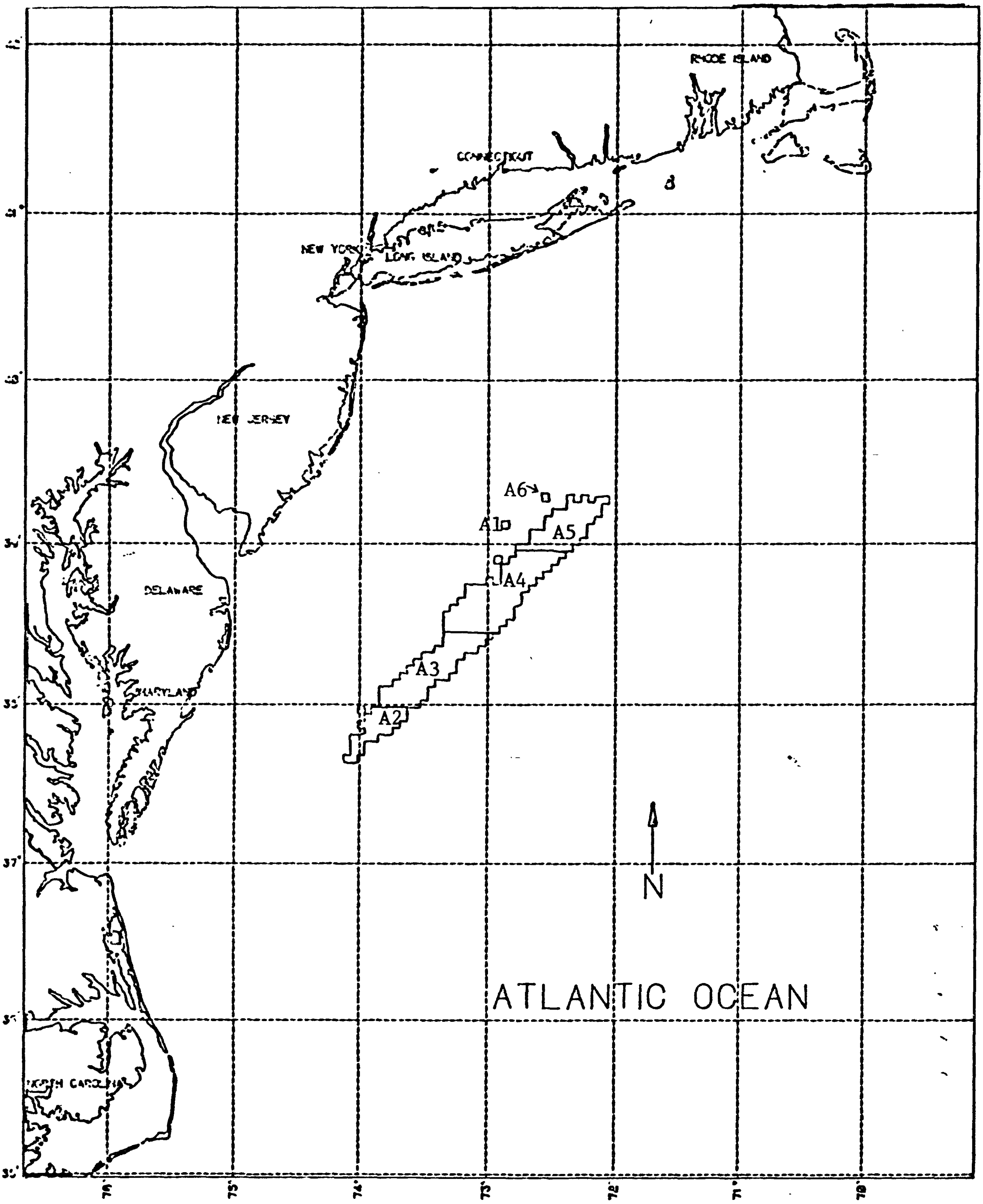


Figure 7.--Map showing the area launch sites for Mid-Atlantic OCS Lease Sale 59.

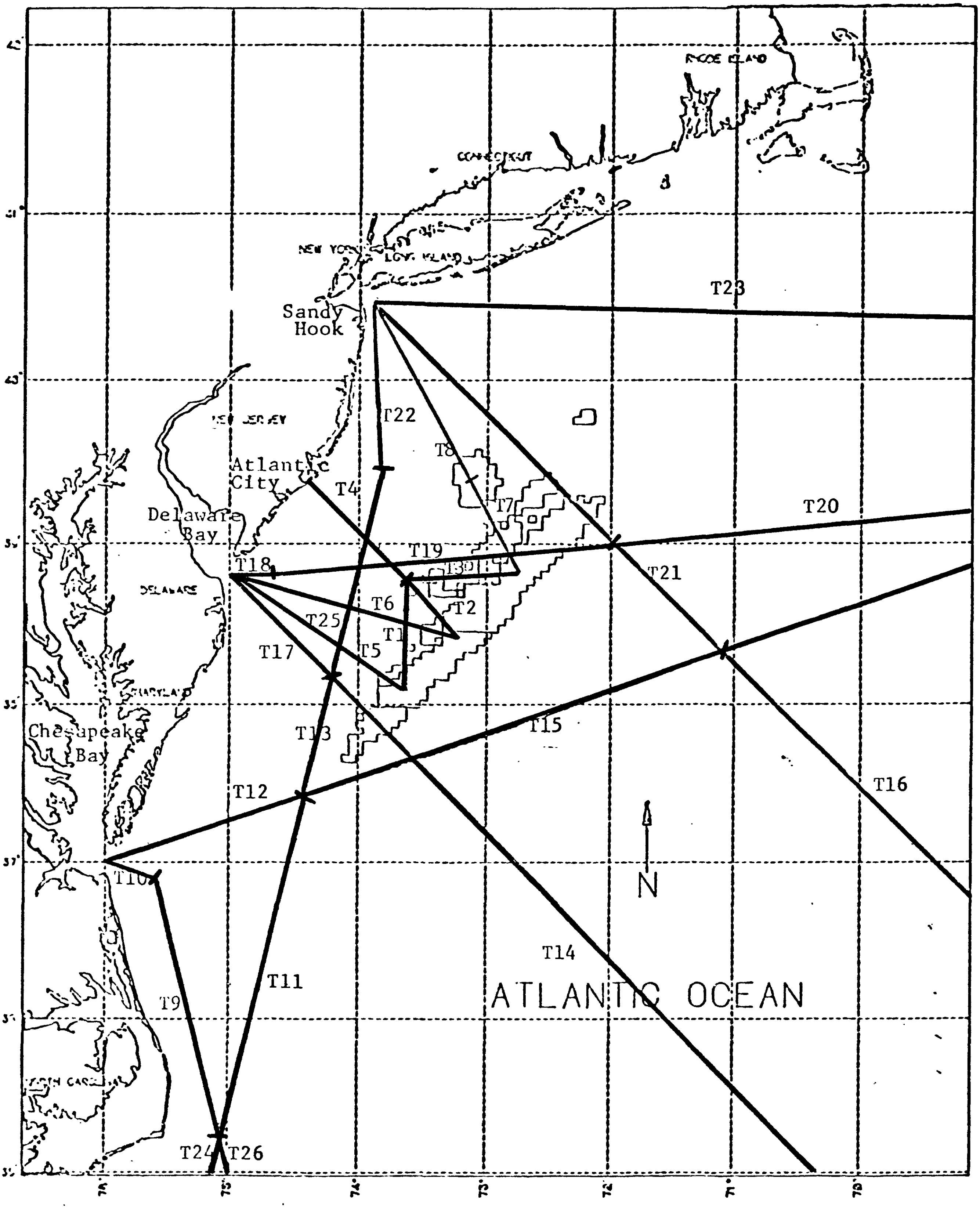


Figure 8.--Map showing the transportation·routes used as launch sites for Mid-Atlantic OCS Lease Sale 59.

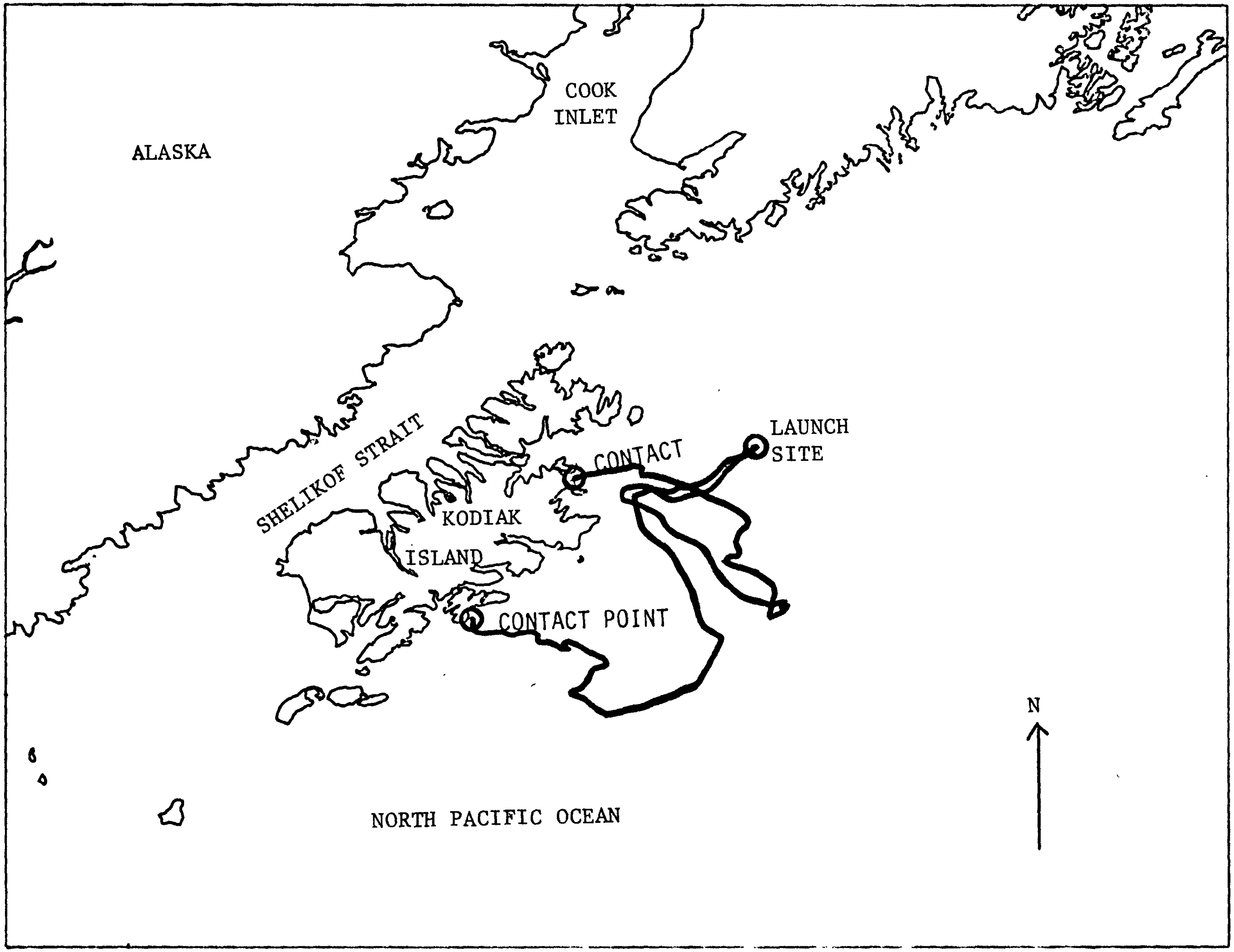


Figure 9.--Map showing two sample oilspill trajectories for the Kodiak Island, Alaska, OCS Lease Sale 46.

(see coding of SPILL). The land segment matrix is searched to determine which segment in each set was contacted.

Oilspill trajectories are simulated for a maximum of 30 days. Spill movement can end in one of three ways:

1. the spill contacts land
2. the time limit exceeds 30 days (spill decayed)
3. the spill moves beyond the boundaries of the study area.

For the last case, the direction of each spill leaving the study area is recorded. The following statistics are recorded for each season:

1. the number of hits on each target
2. the number of spills that went off the map
3. the number of spills that hit land
4. the number of spills that decayed
5. the minimum, maximum and average travel times (days) before spills hit land
6. the number of spills that hit land for each time period (3, 10, or 30 days).

For each spill, the following statistics are also recorded:

1. the time of travel before the spill first hit each target
2. the final location of the spill
3. the land segment(s) in each set hit by the spill
4. the month in which each target, if vulnerable was first hit.

CALCULATION OF CONDITIONAL PROBABILITIES

The disk files produced by program SPILL contain data about the trajectories of 2,000 hypothetical oilspills from each launch point, and the contacts made by these trajectories to targets and land segments. SPILL does not perform any analysis or interpretation of these data; summations and statistical analyses are performed by the programs which follow SPILL. The first set of these programs, HITPROB, LANDSEG1, and LANDSEG2, determine the probabilities that, if a spill occurs at a given launch site, it will contact certain targets or land segments within 3, 10, or 30 days. This information, in turn, becomes the basis for all subsequent analyses. Figure 10 shows the flow chart for calculating conditional probabilities.

HITPROB, LANDSEG1, and LANDSEG2 all operate in a similar manner. For operational reasons, SPILL usually produces output in five groups, with each group containing data on 400 spills (100 per season) per launch site; all five groups are subsequently processed at the same time. If the logistics of running SPILL required the groups to be further subdivided, the files must be concatenated so that each group contains a complete set of launch sites; the launch sites must also be in the same order for each group.

HITPROB produces a matrix, [C], of dimensions (number of targets) x (number of launch points), where each element, c(i,j) represents the probability that, if a spill occurs at launch point

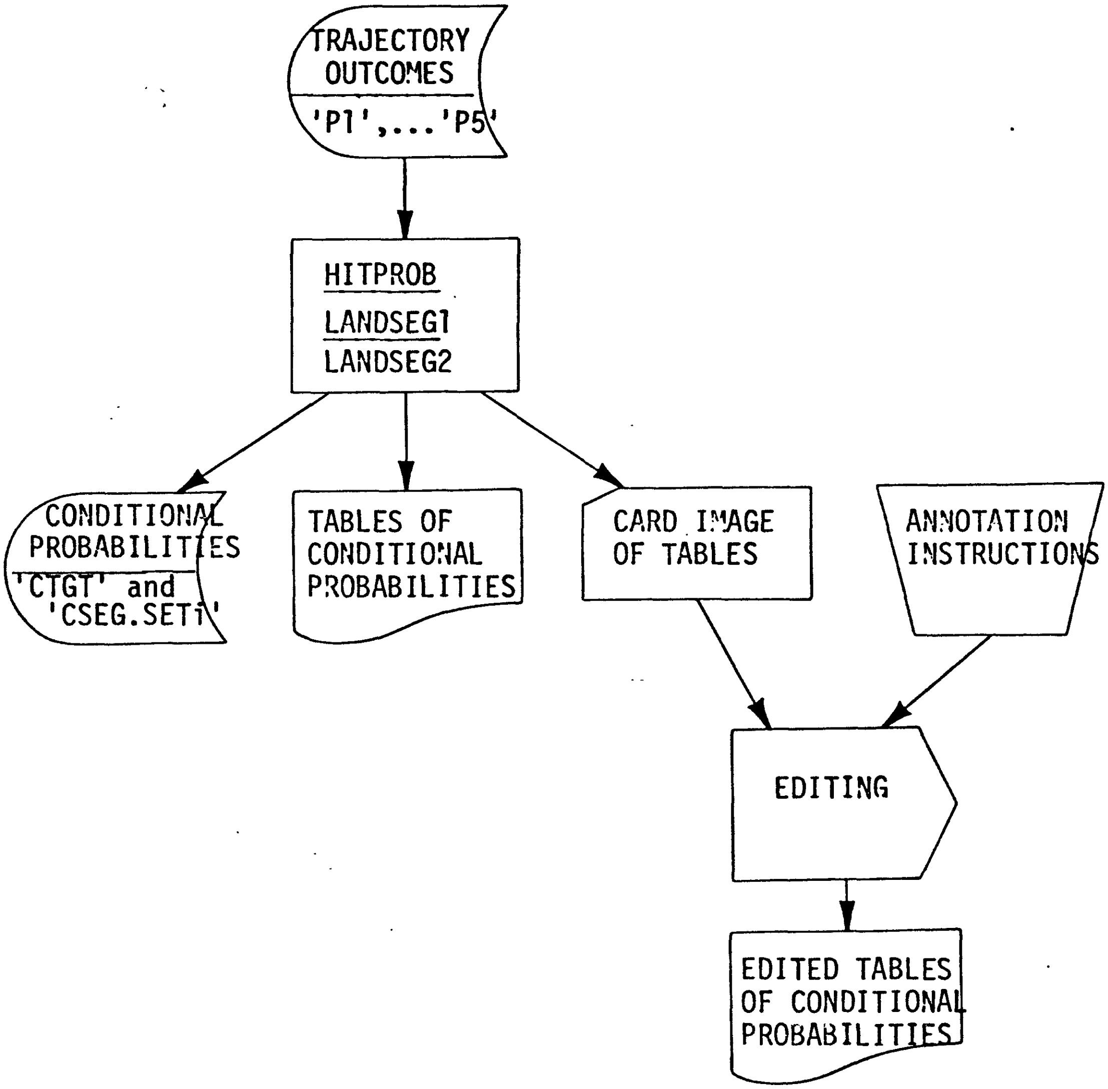


Figure 10.--Flow chart for calculating conditional probabilities.

j, it will contact target i within a specified period of time. Matrices are constructed for travel times of 3, 10, and 30 days. The output of HITPROB is called OSRA.&SALE.CTGT. LANDSEG1 and LANDSEG2 do exactly the same calculations for land segments sets 1 and 2 respectively. The only difference is in decoding the land segment code. The output files of LANDSEG1 and LANDSEG2 are named OSRA.&SALE.CSEG.SET1 and OSRA.&SALE.CSEG.SET2. Table 3 shows some sample output from HITPROB for Mid-Atlantic OCS Lease Sale 59. Table 4 shows some sample output from LANDSEG2 for Mid-Atlantic OCS Lease Sale 59 (Samuels and Lanfear, 1980). All three programs produce printouts of the matrices. LANDSEG1 and LANDSEG2 are listed as LANDSEG in the appendix.

SUMMARY AND CONCLUSIONS

The trajectory portion of the U.S. Geological Survey Oilspill Risk Analysis model was developed to analyze the probability of oilspills contacting coastal areas and environmental resources. The latest versions of the procedures and programs used to calculate oilspill trajectories and conditional probabilities have been documented. In this paper, operation of the model is emphasized over modeling theory.

Table 3. -- Probabilities (expressed as percent chance) that an oilspill starting at a particular location will contact a certain target within 3 days, (Mid-Atlantic OCS Lease Sale 59).

Target	Hypothetical Spill Location																					
	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	T1	T2	T3	T4
Land	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	6
Wading bird colonies	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	5
Sea duck wintering	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	1
Osprey nesting areas	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	1
Bald eagle nesting	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Peregr. falcon nest.	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	4
Peregr. falcon migr.	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	2
Sea turtle nesting	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Grey seal area	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Hard clam grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	1
Soft clam grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Bay scallop areas	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Oyster grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Blue crab grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Lobster grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Sandy beaches	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	3
Coastal marshes	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	1
Natl. Pks.,Shr.,Rec.	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
St. Pks. and Shores	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Natl. Wildlife Ref.	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	6
St. Wild.,Natr. Area	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	1
St. mar. sanctuaries	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Non-Govt. Wld., Nat.	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Narragansett Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Long Island Sound	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Delaware Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Chesapeake Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Raritan Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Sewage dumping site	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26
Land	3	3	n	7	13	25	n	10	n	n	n	n	4	11	n	n	2	15	2	n	n	n
Wading bird colonies	1	1	n	n	7	7	n	5	n	n	n	n	1	1	n	n	n	1	n	n	n	n
Sea duck wintering	12	10	n	5	n	n	n	6	n	n	n	n	19	56	1	n	2	11	6	n	1	n
Osprey nesting areas	2	2	n	3	n	10	n	6	n	n	n	n	3	7	n	n	1	6	1	n	n	n
Bald eagle nesting	1	1	n	n	n	4	n	1	n	n	n	n	1	1	n	n	n	n	n	n	n	n
Peregr. falcon nest.	n	n	n	1	n	n	n	n	n	n	n	n	n	n	n	n	n	4	n	n	n	n
Peregr. falcon migr.	1	1	n	n	4	7	n	1	n	n	n	n	2	3	n	n	n	1	n	n	n	n
Sea turtle nesting	n	n	n	n	9	16	n	4	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Grey seal area	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Hard clam grounds	1	1	n	n	n	n	n	n	n	n	n	n	1	3	n	n	n	1	n	n	n	n
Soft clam grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Bay scallop areas	n	n	n	1	n	n	n	n	n	n	n	n	n	n	n	n	n	4	n	n	n	n
Oyster grounds	n	n	n	n	n	n	n	2	n	n	n	n	1	1	n	n	n	n	n	n	n	n
Blue crab grounds	n	n	n	n	n	16	n	4	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Lobster grounds	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Sandy beaches	4	3	n	9	19	35	n	8	n	n	n	n	6	15	n	n	3	21	3	n	n	n
Coastal marshes	1	1	n	2	1	4	n	1	n	n	n	n	1	2	n	n	n	3	n	n	n	n
Natl. Pks.,Shr.,Rec.	1	1	n	1	9	3	n	n	n	n	n	n	2	2	n	n	n	1	n	n	n	n
St. Pks. and Shores	1	2	n	1	n	3	n	n	n	n	n	n	2	8	n	n	n	4	1	n	n	n
Natl. Wildlife Ref.	1	n	n	1	5	12	n	5	n	n	n	n	1	1	n	n	n	3	n	n	n	n
St. Wild.,Natr. Area	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
St. mar. sanctuaries	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Non-Govt. Wld., Nat.	n	n	n	n	n	1	n	5	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Narragansett Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Long Island Sound	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Delaware Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	1	n	n	n	n	n	n	n	n
Chesapeake Bay	n	n	n	n	n	1	n	2	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Raritan Bay	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Sewage dumping site	n	n	n	28	n	n	n	n	n	n	n	n	n	n	n	n	12	24	8	n	n	n

Note: n = less than ~~0.5 percent~~
A1 to A6 are proposed lease tract subdivisions
B1 to B12 are existing lease tract subdivisions
T1 to T26 are transportation routes

Table 4. -- Probabilities (expressed as percent chance) that an oilspill starting at a particular location will contact a certain land segment within 3 days, (Mid-Atlantic OCS Lease Sale 59).

Land Segment	Hypothetical Spill Location A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	T1	T2	T3	T4
2	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
3	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
4	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
5	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
6	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
7	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
8	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
9	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
10	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	2
11	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	4
12	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
13	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
15	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

Land Segment	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26
2	n	n	n	n	5	1	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
3	n	n	n	n	8	13	n	2	n	n	n	n	n	n	n	n	n	n	n	n	n	n
4	n	n	n	n	n	10	n	1	n	n	n	n	n	n	n	n	n	n	n	n	n	n
5	n	n	n	n	n	1	n	3	n	n	n	n	n	n	n	n	n	n	n	n	n	n
6	n	n	n	n	n	n	n	3	n	n	n	n	n	n	n	n	n	n	n	n	n	n
7	1	1	n	n	n	n	n	n	n	n	n	n	1	1	n	n	n	n	n	n	n	n
8	2	2	n	n	n	n	n	n	n	n	n	n	3	10	n	n	n	n	n	n	n	n
9	n	n	n	n	n	n	n	n	n	n	n	n	n	1	n	n	n	n	n	n	n	n
10	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
11	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
12	n	n	n	1	n	n	n	n	n	n	n	n	n	n	n	n	n	4	n	n	n	n
13	n	n	n	5	n	n	n	n	n	n	n	n	n	n	n	n	1	10	1	n	n	n
15	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	1	n	n	n

Note: n = less than 0.5 percent.
Land segments for which all probabilities are less than 0.5 percent are not shown.

REFERENCES CITED

LaBelle, R.P., Samuels, W.B. and Lanfear, K.J., 1980, An oilspill risk analysis for the Cook Inlet-Shelikof Strait (Proposed Sale 60) Outer Continental Shelf lease area. U.S. Geological Survey Open-File Report No. 80-863, 83 p.

LaBelle, R.P., and Lanfear, K.J., 1981, An oilspill risk analysis for the Gulf of Mexico (Proposed Sale 67) Outer Continental Shelf lease area, U.S. Geological Survey Open-File Report No. 81-306

Lanfear, K.J., Smith, R.A., and Slack, J.R., 1979. An introduction to the oilspill risk analysis model: Proceedings of the Offshore Technology Conference, 11th, Houston, Tex., 1979, OTC 3607, p. 2173-2175.

Lanfear, K.J., and Nakassis, A., 1980, Documentation and user's guide to the spatial environmental data digitizing system, SEDDS: U.S. Geological Survey Open-File Report No. 80-871, 129 p.

Smith, R.A., Slack, J.R., Wyant, T., and Lanfear, K.J., 1980. The oilspill risk analysis model of the U.S. Geological Survey: U.S. Geological Survey Open-File Report No. 80-687, 107 p.

Samuels, W.B., Lanfear, K.J. and Nakassis, A., 1980. An oilspill risk analysis for Kodiak Island (Proposed Sale 46) Outer Continental Shelf Lease Area: U.S. Geological Survey Open-File Report No. 80-175, 66 p.

Samuels, W.B. and Lanfear, K.J., 1980. An oilspill risk analysis for the Mid-Atlantic (Proposed Sale 59) Outer Continental Shelf Lease Area: U.S. Geological Survey Open-File Report No. 80-2026, 82 p.

APPENDIX
Procedures and Programs

Disclaimer for Distribution of Programs

Although these programs have been tested by the Geological Survey, United States Department of the Interior, no warranty, expressed or implied, is made by the Geological Survey as to the accuracy and functioning of the programs and related program material, nor shall the distribution constitute any such warranty, and no responsibility is assumed by the Geological Survey in connection therewith.

A Note on Program Authorship

The computer programs of the OSTA model have been modified many times since they were first written in 1976. Since a number of programmers were involved, it is difficult to attribute specific programs to individuals. The following USGS employees contributed to the programs in this appendix:

James R. Slack, Water Resources Division, Systems Analysis Group
Richard A. Smith, Water Resources Division, Systems Analysis Group
Timothy Wyant, Water Resources Division, Systems Analysis Group
Kenneth J. Lanfear, OESA, EAO, Environmental Modeling Group
Anastase Nakassis, OESA, EAO, Environmental Modeling Group
William B. Samuels, OESA, EAO, Environmental Modeling Group

Procedure: CURVAL
Executes program: CURVAL
Purpose: To set up the matrix of monthly current vectors.
Reads: Card deck containing monthly current vectors for each
       current polygon.
Writes files: OSRA.&SALE.CURVAL
Text page reference: 3

Procedure listing:

```
//CURVAL PROC U=CCDXXX
//C1 EXEC PGM=IEFBR14
//C1 DD UNIT=3330,VOL=SER=&U,DISP=(OLD,DELETE),
// DSN=OSRA.&SALE.CURVAL
//G EXEC PGM=CURVAL,REGION=200K,TIME=1
//STEPLIB DD UNIT=3330,VOL=SER=&U,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&U,DISP=(NEW,KEEP),
//  DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6444),
//  SPACE=(TRK,(5,5),RLSE),DSN=OSRA.&SALE.CURVAL
```

Program listing:

```
C     PROGRAM CURVAL
C     SETS UP THE CURRENT VALUE FILE
C     CURRENT VELOCITIES ARE IN M/SEC AND MUST BE INPUT AS SUCH
C     DECEMBER IS CONSIDERED THE FIRST MONTH
C
C     CARD INPUT:
C       CARD 1:
C         COL. 01-60 CURRENT FILE NAME (15A4)
C       CARDS 2-N:
C         COL. 01-04 POLYGON NUMBER (I4)
C         COL. 5-52 AZIMUTH IN DEGREES FOR EACH MONTH (12I4)
C         CONTINUATION CARD
C         COL. 01-60 VELOCITY IN M/SEC FOR EACH MONTH (12F5.2)
C
C     OUTPUT:
C       DISK: FILE OF X AND Y CURRENT VECTORS
C       PRINT: CONTROL MESSAGES
C
      REAL MNAME(12)/'DEC','JAN','FEB','MAR','APR','MAY','JUN','JUL',
     1   'AUG','SEP','OCT','NOV'/
      REAL XCUR(1000,12),YCUR(1000,12),CURNAM(15)
      REAL RADDEG/0.017453293/
      INTEGER D(12)
      REAL V(12)
      INTEGER CARD/5/,PRINT/6/,UVAL/10/
C     ZERO OUT THE VECTORS
      CALL ZERO4(XCUR,1000*12)
      CALL ZERO4(YCUR,1000*12)
      READ(CARD,5001)CURNAM
 5001 FORMAT(15A4)
      WRITE(PRINT,6001)CURNAM,MNAME
 6001 FORMAT('1PROGRAM CURVAL.'//' CURRENT DATA FOR ',15A4//
     1  ' POLYGON ',12(1X,A4))
C     PROCESS EACH POLYGON
      DO 20 I=1,1000
      READ(30,END=30)IP,D,V
      WRITE(PRINT,6002)IP,D,V
 6002 FORMAT(/I5,1X,12I4/6X,12F5.2)
C     DO EACH MONTH
      DO 10 IM=1,12
      A=D(IM)*RADDEG
      Q=V(IM)
      XCUR(IP,IM)=SIN(A)*Q
      YCUR(IP,IM)=COS(A)*Q
   10 CONTINUE
   20 CONTINUE
C     THIS IS AN ERROR SINCE ONLY 1000 VALUES ARE ALLOWED
      WRITE(PRINT,3001)
 3001 FORMAT(///'***** MORE THAN 1000 VALUES HAVE BEEN ENCOUNTERED'/
     1  ' EXECUTION TERMINATED.')
      STOP 16
C     THIS IS THE NORMAL LOOP EXIT
   30 CONTINUE
      WRITE(UVAL) CURNAM
      WRITE(UVAL) XCUR
      WRITE(UVAL) YCUR
      WRITE(PRINT,6003)
 6003 FORMAT(//' NORMAL TERMINATION.')
      STOP
      END
```

Procedure: CURPLOT
Executes program: CURPLOT
Purpose: Plot the monthly current vectors on a base map.
Reads files: OSRA.&SALE.CURVAL
OSRA.&SALE.DEFPOLY.IDS.&FILEOUT
OSRA.&SALE.BASEMAP
Writes files: Magnetic tape containing plotting instructions
for the current vector arrows and the base map.
Text page reference: 3

Procedure listing:

```
//CURPLOT PROC U=CCDXXX
//G EXEC PGM=CURPLOT,REGION=250K,TIME=3
//STEPLIB DD UNIT=3330,VOL=SER=&U,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&U,DISP=SHR,
//  DSN=OSRA.&SALE.CURVAL
//FT13F001 DD UNIT=TAPELO,VOL=SER=&PTAPE,
//  DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=72,BLKSIZE=1008,DEN=2),
// LABEL=(1,SL)
//FT13F002 DD UNIT=TAPELO,VOL=SER=&PTAPE,
// DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=72,BLKSIZE=1008,DEN=2),
// LABEL=(2,SL)
//FT20F001 DD UNIT=3330,VOL=SER=&U,DISP=SHR,
// DSN=OSRA.&SALE.DEFPOLY.IDS.&FILEOUT
//FT90F001 DD UNIT=3330,VOL=SER=&U,DISP=SHR,
// DSN=OSRA.&SALE.BASEMAP
```

Program listing:

```
C     PROGRAM CURPLOT
C
C
C     PLOTS THE MONTHLY CURRENT VECTORS
C     USING GERBER SUBROUTINES
C     NOTE: DECEMBER IS THE FIRST MONTH
C
C     XCUR, YCUR - ARRAY OF CURRENT VECTORS
C
C     INPUT:
C       CARD NUMBER 1
C         COL. 01-08 SALE (2A4)
C         COL. 09 BLANK
C         COL. 10-14 SCALE LENGTH OF CURRENT VECTOR (F5.2)
C         COL. 15 BLANK
C         COL. 16-20 UNITS PER INCH (F5.2)
```

```
C
C       CARD NUMBERS 2-N
C         COL. 01-04 MONTH (A4)
C         COL. 05 BLANK
C         COL. 06-07 MONTH NUMBER (I2)
C         COL. 08 BLANK
C         COL. 09-18 X ORIGIN (F10.2)
C         COL. 19 BLANK
C         COL. 20-29 Y ORIGIN (F10.2)
C         COL. 30 BLANK
C         COL. 31 0=NO CURRENTS, 1=CURRENTS (I1)
C         COL. 32 BLANK
C         COL. 33 0=NO WAIT, 1=WAIT
C
C       DISK:
C         FILE OF CURRENT VECTORS
C
C     OUTPUT
C       PRINT: CONTROL MESSAGES
C
C       PLOT: MAP OF MONTHLY CURRENT VECTORS
C
      DIMENSION CURNAM(15)
      DIMENSION X(100),Y(100),NUMB(100)
      INTEGER CARD/5/,PRINT/6/,UVAL/10/,UNUMB/20/,UBASE/90/
      REAL XCUR(1000,12),YCUR(1000,12)
      REAL MSPK
      REAL SALE(2)
      REAL RADDEG/0.0174533/
      REAL A/0.0/
C
      READ(CARD,5001)SALE,MSPK,UPI
 5001 FORMAT(2A4,2(1X,F5.2))
C     MSPK=SCALE LENGTH OF A 1 M/SEC CURRENT VECTOR IN METERS
C     DEFAULT IS 50 METERS
      IF(MSPK.LT.0.1)MSPK=50.0
      WRITE(PRINT,6009)SALE,MSPK,UPI
 6009 FORMAT(' ','SALE= ',2A4,' MSPK= ',F5.2,' UPI= ',F5.2)
C
C     READ CURRENT DATA
      READ(UVAL)CURNAM
      READ(UVAL)XCUR
      READ(UVAL)YCUR
C
C     PRINT A MESSAGE
      WRITE(PRINT,6000)CURNAM
 6000 FORMAT('1PROGRAM CURPLOT2.'//' PLOTTING THE MONTHLY CURRENT',
     1   ' VECTOR MAPS FOR: ',15A4)
C
```

```
      CALL INIT(1)
      CALL SCAL ((1.0/UPI),(1.0/UPI))
C     PLOT A BASEMAP ON FILEMARK 1
      CALL PLOTMP(SALE,40.0,40.0,0,0,UBASE)
      WRITE(PRINT,6004)
 6004 FORMAT(' ','BASEMAP IS ON FILEMARK 1')
      CALL DONE
C
      CALL INIT(1)
      CALL SCAL((1.0/UPI),(1.0/UPI))
C     READ THE MONTHS TO PLOT
  100 READ(CARD,5000,END=999)MNAME,MNUM,XORIG,YORIG,IC,IW
 5000 FORMAT(A4,1X,I2,2(1X,F10.2),2(1X,I1))
      WRITE(PRINT,6001)MNAME,MNUM,XORIG,YORIG,IC,IW
 6001 FORMAT(' MONTH: ',A4,' MNUM= ',I2,' XORIG= ',F10.2,
     1 ' YORIG= ',F10.2,' CURRENTS= ',I1,' WAIT= ',I1)
      IF(IW.EQ.1) GO TO 1000
  101 CONTINUE
      CALL ZER((XORIG/UPI),(YORIG/UPI))
C
C     WRITE A TITLE ON THE PLOT
      CALL ALPHA2(0.0,-20.0,10.0,0.0,4,MNAME)
      CALL ALPHA2(60.0,-20.0,10.0,0.0,12,' 1 M/SEC IS ')
      CALL ARROW2(240.0,-20.0,0.0,MSPK,1)
C
C     PLOT THE CURRENT VECTOR
C     PLOT THE CURRENT VECTOR ONLY IF IC=1
      IF(IC.NE.1)GO TO 103
  500 READ(UNUMB,END=103)J,X,Y,NUMB
      DO 90 I=1,J
      IF(NUMB(I).LE.0.OR.NUMB(I).GT.1000)GO TO 90
      IF(YCUR(NUMB(I),MNUM).EQ.0.0.AND.XCUR(NUMB(I),MNUM).EQ.0.0)
     1  GO TO 80
      Z=SQRT(XCUR(NUMB(I),MNUM)**2+YCUR(NUMB(I),MNUM)**2)*MSPK
      IF(YCUR(NUMB(I),MNUM).NE.0.0.OR.XCUR(NUMB(I),MNUM).NE.0.0)
     1  A=ATAN2(YCUR(NUMB(I),MNUM),XCUR(NUMB(I),MNUM))/RADDEG
      CALL ARROW2(X(I),Y(I),A,Z,2)
      GO TO 90
   80 CONTINUE
      CALL ALPHA2(X(I),Y(I),5.0,0.0,1,'0')
   90 CONTINUE
      GO TO 500
C
  103 CONTINUE
C     WRITE A MESSAGE THAT THE MONTH IS FINISHED
      WRITE(PRINT,6002)MNAME
 6002 FORMAT(' FINISHED ',A4)
      REWIND 20
      GO TO 100
```

```
C
C     THE WAIT COMMAND WAS INDICATED
 1000 CONTINUE
      CALL WAIT
      GO TO 101
C
C     ALL DONE
  999 CONTINUE
C     RELEASE THE PLOTTER
      CALL DONE
      STOP
      END
```

CURBAL

Procedure: CURBAL
Executes program: CURBAL
Purpose: Checks the current flow into and out of each current
         polygon.
Reads files: OSRA.&SALE.CURVAL
             OSRA.&SALE.GENERAL
             OSRA.&SALE.DEFPOLY.BIGMATRX.&FILEOUT
Text page reference: 6

Procedure listing:

```
//CURBAL PROC U=CCDXXX
//G EXEC PGM=CURBAL,REGION=600K,TIME=2
//STEPLIB DD UNIT=3330,VOL=SER=&U,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&U,DISP=SHR,
// DSN=OSRA.&SALE.CURVAL
//FT20F001 DD UNIT=3330,VOL=SER=&U,DISP=SHR,
// DSN=OSRA.&SALE.GENERAL
//FT30F001 DD UNIT=3330,VOL=SER=&U,DISP=SHR,
// DSN=OSRA.&SALE.DEFPOLY.BIGMATRX.&FILEOUT
```

Program listing:

```
C     PROGRAM CURBAL
C     CHECKS CURRENT FLOW INTO AND OUT OF
C     EACH POLYGON.
C
C
C     INPUT:
C       CARD NUMBER 1
C         COL. 01-05 NUMBER OF POLYGONS (I5)
C
C
      REAL F(1000,12)
      DIMENSION CURNAM(15)
      DIMENSION XCUR(1000,12),YCUR(1000,12)
      INTEGER*2 NUM(500,500)
      INTEGER UVAL/10/,UGEN/20/,UMAT/30/
      REAL MNAME(12)/'DEC','JAN','FEB','MAR','APR','MAY','JUN','JUL',
     1  'AUG','SEP','OCT','NOV'/
      REAL SALE(2)
C
      READ(5,5001)NP
 5001 FORMAT(I5)
      READ(UVAL)CURNAM
      READ(UVAL)XCUR
      READ(UVAL)YCUR
      READ(UGEN)SALE,IX,IY
      M=IX
      N=IY
      WRITE(6,6010)SALE,IX,IY,M,N
 6010 FORMAT(' ',' SALE= ',2A4,' IX= ',I8,' IY= ',I8,' M= ',I8,' N=',I8)
      READ(UMAT)NUM
      WRITE(6,6000)CURNAM,MNAME
 6000 FORMAT('1PROGRAM CURBAL '//' BALANCING CURRENT VECTORS FOR ',
     1  15A4//' POLYGON ',12(2X,A4))
      CALL ZERO4(F,1000*12)
      DO 600 IM=1,12
      L=M-1
C     DO THE HORIZONTAL VECTOR
      DO 200 I=1,L
      DO 100 J=1,N
      IF(NUM(I,J).LE.0.OR.NUM(I,J).GT.600) GO TO 100
      IF(NUM(I+1,J).LE.0.OR.NUM(I+1,J).GT.600) GO TO 100
      IP=NUM(I,J)
      IQ=NUM(I+1,J)
      IF(IP.EQ.IQ)GO TO 100
      VX=(XCUR(IP,IM)+XCUR(IQ,IM))/2.0
      F(IP,IM)=F(IP,IM)-VX
      F(IQ,IM)=F(IQ,IM)+VX
  100 CONTINUE
  200 CONTINUE
C     DO THE VERTICAL VECTOR
      L=N-1
      DO 500 J=1,L
      DO 400 I=1,M
      IF(NUM(I,J).LE.0.OR.NUM(I,J).GT.600) GO TO 400
      IF(NUM(I,J+1).LE.0.OR.NUM(I,J+1).GT.600) GO TO 400
      IP=NUM(I,J)
      IQ=NUM(I,J+1)
      IF(IP.EQ.IQ)GO TO 400
      VY=(YCUR(IP,IM)+YCUR(IQ,IM))/2.0
      F(IP,IM)=F(IP,IM)-VY
      F(IQ,IM)=F(IQ,IM)+VY
  400 CONTINUE
  500 CONTINUE
      IN=IQ
C     DO THE BOUNDARIES
      DO 450 I=1,M
      IF(NUM(I,1).LE.0.OR.NUM(I,1).GT.600) GO TO 450
      IF(NUM(I,N).LE.0.OR.NUM(I,N).GT.600)GO TO 450
      IP=NUM(I,1)
      IQ=NUM(I,N)
      F(IP,IM)=F(IP,IM)+YCUR(IP,IM)
      F(IQ,IM)=F(IQ,IM)-YCUR(IQ,IM)
  450 CONTINUE
```

```
      DO 550 J=1,N
      IF(NUM(1,J).LE.0.OR.NUM(1,J).GT.600) GO TO 550
      IF(NUM(M,J).LE.0.OR.NUM(M,J).GT.600) GO TO 550
      IP=NUM(1,J)
      IQ=NUM(M,J)
      F(IP,IM)=F(IP,IM)+XCUR(IP,IM)
      F(IQ,IM)=F(IQ,IM)-XCUR(IQ,IM)
  550 CONTINUE
  600 CONTINUE
      DO 800 I=1,NP
      WRITE(6,6001)I,(F(I,J),J=1,12)
 6001 FORMAT(' ',2X,I3,3X,12F6.1)
  800 CONTINUE
      STOP
      END
```

RAWWIND

Procedure: RAWWIND
Executes program: RAWWIND
Purpose: Read wind data from a NOAA TDF-11 or TDF-14 weather tape. Decodes the wind direction and speed.
Reads tape: NOAA Weather tape
Writes files: OSRA.RAWWIND.&STA
Text page reference: 6

Procedure listing:

```
//RAWWIND PROC UVOL=CCDXXX
//*
//* RAWWIND
//* EXECUTES PROGRAM RAWWIND, TO READ A NOAA WIND TAPE, AND STORE
//*   THE EDITED DATA ON A DISK FILE.
//* REQUIRED: STA=
//*
//*
//* CLEAR THE OLD FILE (IF ANY).
//C EXEC PGM=IEFBR14
//OLD DD UNIT=3330,VOL=SER=&UVOL,DISP=(OLD,DELETE),
// DSN=OSRA.RAWWIND.&STA
//*
//* EXECUTE PROGRAM RAWWIND, FROM OSRA.PGMLIB
//* TIME=15 IS SUFFICIENT FOR ONE TAPE.
//G EXEC PGM=RAWWIND,TIME=15
//STEPLIB DD UNIT=3330,VOL=SER=&UVOL,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=TAPE9,LABEL=(1,SL,,IN),VOL=SER=&TAPE,DSN=&NAME,
// DCB=(RECFM=F,BLKSIZE=495)
//FT20F001 DD UNIT=3330,VOL=SER=&UVOL,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=6440,BLKSIZE=6444),SPACE=(TRK,(10,10),RLSE),
// DSN=OSRA.RAWWIND.&STA
```

Program listing:

```
C     PROGRAM RAWWIND -- VERSION 2.0--18 MAR 77.
C     READS WIND DATA FROM A NOAA TDF-11 OR TDF-14 TAPE.
C     DECODES THE WIND DIRECTION AND SPEED FIELDS.
C     PRODUCES A COMPACT FILE OF WIND VELOCITY AND A LISTING
C        OF NUMBER OF OBSERVATIONS PER DAY.
      REAL STANAM(10)
      INTEGER CARD/5/,PRINT/6/,UOUT/20/
C     READ THE STATION NUMBER AND NAME AND TAPE FORMAT.
      READ (CARD,5001) ITDF,ISTA,STANAM
 5001 FORMAT (I2,1X,I5,1X,10A4)
C     WRITE THE HEADING.
```

```
      WRITE (PRINT,6001) ISTA,STANAM,ITDF
 6001 FORMAT ('1 RAWWIND FOR STATION',I6,1X,10A4/
     1   ' USING THE TDF-',I2,' FORMAT.')
C     WRITE THE FIRST HEADER RECORD ON THE OUTPUT FILE.
      WRITE (UOUT) ITDF,ISTA,STANAM
C     GO TO CORRECT FORMAT ROUTINE.
      IF (ITDF.EQ.11) CALL TDF11(ISTA)
      IF (ITDF.EQ.14) CALL TDF14(ISTA)
C     NEITHER CHOSEN. REPORT AND TERMINATE.
      WRITE (PRINT,3001)
 3001 FORMAT (' FORMAT SPECIFIED IS NEITHER 11 OR 14.'/
     1   ' EXECUTION TERMINATED.')
      STOP 16
      END
      SUBROUTINE TDF11(ISTA)
      INTEGER PRINT/6/,UIN/10/,UOUT/20/
      INTEGER*2 LYR,LMO,LDA
      INTEGER*2 DIR(24)/24*-1/,SPD(24)/24*-1/
      INTEGER*2 KNT(31,12)/372*0/,BLANK/' '/,A0/'0'/,A1/'1'/,A2/'2'/
      INTEGER*2 IYR,IMO,IDA,IDI,IDC,ISI,ISV
C     READ FIRST RECORD.
      READ (UIN,1001) IYR,IMO,IDA,IHR,IDI,IDC,ISI,ISV,JSTA
 1001 FORMAT (18X,I2,3I2,A1,I2,A1,I3,T90,I4)
      LYR=IYR
      LMO=IMO
      LDA=IDA
C     SET DIRECTION CODE.
      IDW=0
      IF (IDI.EQ.BLANK) IDW=2
      IF (IDI.EQ.A0) IDW=3
      IF (IDI.EQ.A1) IDW=4
      IF (IDI.EQ.A2) IDW=5
C     WRITE THE SECOND HEADER RECORD.
      WRITE (UOUT) IDW
      WRITE (6,6001) IDW
 6001 FORMAT (' THE KEY IS',I2/)
C     VERIFY THE STATION NUMBER.
      IF (JSTA.EQ.ISTA) GO TO 40
C     THE STATION NUMBER IS INCORRECT.
C     REPORT AND TERMINATE.
      WRITE (PRINT,3001) JSTA
 3001 FORMAT (' THE STATION NUMBER ON THE TAPE IS',I5/
     1   ' EXECUTION TERMINATED.')
      STOP 16
C     COME HERE FOR A NEW RECORD.
   10 CONTINUE
      READ (UIN,1002,END=70) IYR,IMO,IDA,IHR,IDC,ISI,ISV
 1002 FORMAT (18X,I2,3I2,1X,I2,A1,I3)
C     STILL SAME YEAR?
```

```
      IF (IYR.EQ.LYR) GO TO 20
C     NO. REPORT LAST YEAR.
      WRITE (PRINT,6002) LYR,KNT
 6002 FORMAT (/' NUMBER OF OBSERVATIONS PER DAY FOR 19',I2/(31I3))
C     ZERO OUT COUNTER.
      CALL ZERO2(KNT,372)
   20 CONTINUE
C     IS THIS A NEW DAY?
      IF (IDA.EQ.LDA) GO TO 40
C     YES. WRITE OUT THE PREVIOUS DAY.
      WRITE (UOUT) LYR,LMO,LDA,DIR,SPD
C     SET VELOCITY TO -1 TO INDICATE MISSING DATA.
      DO 30 I=1,24
      DIR(I)=-1
      SPD(I)=-1
   30 CONTINUE
      LYR=IYR
      LMO=IMO
      LDA=IDA
   40 CONTINUE
C     PROCESS THE DATA.
      IHR=IHR+1
C     IS THIS A VALID VALUE?
      IF (ISI.EQ.BLANK.OR.IDC.GT.36) GO TO 10
C     YES. STORE THE SPEED.
      SPD(IHR)=ISV
C     CONVERT THE DIRECTION CODE.
C     THIS IS THE "OUT OF" DIRECTION CODE.
      DIR(IHR)=IDC+1
C     INCREMENT THE COUNTER.
      KNT(IDA,IMO)=KNT(IDA,IMO)+1
      GO TO 10
C     INPUT EXHAUSTED.
   70 CONTINUE
C     WRITE OUT THE PRESENT DAY AND REPORT THIS YEAR.
      WRITE (UOUT) LYR,LMO,LDA,DIR,SPD
      WRITE (PRINT,6002) LYR,KNT
      STOP
      END
      SUBROUTINE TDF14(ISTA)
      INTEGER PRINT/6/,UIN/10/,UOUT/20/
      INTEGER*2 LYR,LMO,LDA
      INTEGER*2 DIR(24)/24*-1/,SPD(24)/24*-1/,D
      INTEGER*2 KNT(31,12)/372*0/,BLANK/' '/,ASTER/'*'/
      INTEGER*2 IDR(6,2),ISPD(6),JSPD(2,6),IYR,IMO,IDA
      INTEGER KSPD(6)/6*0/,IHR(6)
      EQUIVALENCE(JSPD(1),KSPD(1))
C     READ FIRST RECORD.
      READ (UIN,1001) JSTA,IYR,IMO,IDA,(IHR(I),IDR(I,1),IDR(I,2),
     1   ISPD(I),JSPD(2,I),I=1,6)
 1001 FORMAT (4X,I5,3I2,6(I2,8X,2I1,I2,A1,65X))
      LYR=IYR
      LMO=IMO
      LDA=IDA
C     WRITE THE SECOND HEADER RECORD.
      IDW=1
      WRITE (UOUT) IDW
      WRITE (6,6001) IDW
 6001 FORMAT (' THE KEY IS',I2/)
C     VERIFY THE STATION NUMBER.
      IF (JSTA.EQ.ISTA) GO TO 40
C     THE STATION NUMBER IS INCORRECT.
C     REPORT AND TERMINATE.
      WRITE (PRINT,3001) JSTA
 3001 FORMAT (' THE STATION NUMBER ON THE TAPE IS',I6/
     1   ' EXECUTION TERMINATED.')
      STOP 16
C     COME HERE FOR A NEW RECORD.
   10 CONTINUE
      READ (UIN,1002,END=70) IYR,IMO,IDA,(IHR(I),IDR(I,1),IDR(I,2),
     1   ISPD(I),JSPD(2,I),I=1,6)
 1002 FORMAT (9X,3I2,6(I2,8X,2I1,I2,A1,65X))
C     STILL SAME YEAR?
      IF (IYR.EQ.LYR) GO TO 20
C     NO. REPORT LAST YEAR.
      WRITE (PRINT,6002) LYR,KNT
 6002 FORMAT (/' NUMBER OF OBSERVATIONS PER DAY FOR 19',I2/(31I3))
C     ZERO OUT COUNTER.
      CALL ZERO2(KNT,372)
   20 CONTINUE
C     IS THIS A NEW DAY?
      IF (IDA.EQ.LDA) GO TO 40
C     YES. WRITE OUT THE PREVIOUS DAY.
      WRITE (UOUT) LYR,LMO,LDA,DIR,SPD
C     SET VELOCITY TO -1 TO INDICATE MISSING DATA.
      DO 30 I=1,24
      DIR(I)=-1
      SPD(I)=-1
   30 CONTINUE
      LYR=IYR
      LMO=IMO
      LDA=IDA
   40 CONTINUE
C     PROCESS THE SIX HOURS OF DATA
      DO 60 I=1,6
      JHR=IHR(I)+1
C     IS THIS A VALID VALUE?
      IF (JSPD(2,I).EQ.BLANK.OR.JSPD(2,I).EQ.ASTER) GO TO 50
```

```
C     YES. DO BIT-BUSTING TO EXTRACT THE SPEED.
C     THIS IS IBM-370 SPECIFIC CODE.
      SPD(JHR)=ISPD(I)*10+KSPD(I)/256-192
C     CONVERT THE DIRECTION CODE.
C     CALM=1,N=2,NNW=17
C     THIS IS THE "OUT OF" DIRECTION CODE.
      D=IDR(I,1)*2-1
      IF (IDR(I,2).EQ.IDR(I,1)) D=D+1
      IF (IDR(I,2).EQ.IDR(I,1)+1) D=D+2
      IF (D.EQ.1) D=17
      IF (D.EQ.0) D=1
      DIR(JHR)=D
C     INCREMENT THE COUNTER.
      KNT(IDA,IMO)=KNT(IDA,IMO)+1
   50 CONTINUE
   60 CONTINUE
      GO TO 10
C     INPUT EXHAUSTED.
   70 CONTINUE
C     WRITE OUT THE PRESENT DAY AND REPORT THIS YEAR.
      WRITE (UOUT) LYR,LMO,LDA,DIR,SPD
      WRITE (PRINT,6002) LYR,KNT
      STOP
      END
```

LISTWIND

Procedure: LISTWIND
Executes program: LISTWIND
Purpose: Lists output of RAWWIND
Reads files: OSRA.RAWWIND.&STA
Writes: prints output of RAWWIND
Text page reference: 6

Procedure listing:

```
//LISTWIND PROC UVOL=CCDXXX
//*
//* LISTWIND
//* PRINTS .HE DATA ON THE RAWWIND FILE.
//* REQUIRED: STA=
//* REQUEST EXTRA LINES OF PRINTOUT, IF FULL FILE.
//G EXEC PGM=LISTWIND
//STEPLIB DD UNIT=3330,VOL=SER=&UVOL,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&UVOL,DISP=SHR,
// DSN=OSRA.RAWWIND.&STA
```

Program listing:

```
C     PROGRAM LISTWIND -- VERSION 2.0-- 18 MAR 77.
C     LISTS THE OUTPUT OF RAWWIND.
      INTEGER*2 DIR(24),SPD(24),IYR,IMO,IDA
      INTEGER CARD/5/,PRINT/6/,UIN/10/
      REAL STANAM(10)
      NDAYS=100
C     READ A CARD TO DETERMINE HOW MANY LINES (DAYS) TO LIST.
C     IF NO CARD IS PRESENT, THE DEFAULT VALUE
C        OF NDAYS (SET ABOVE) IS USED.
      READ (CARD,5001,END=10) NDAYS
 5001 FORMAT (I6)
   10 CONTINUE
C     READ THE TWO HEADER RECORDS AND REPORT THEM.
      READ (UIN) ITDF,ISTA,STANAM
      WRITE (PRINT,6001) ISTA,STANAM,ITDF
 6001 FORMAT ('1 LISTWIND FOR STATION',I6,1X,10A4/
     1   ' TAPE FORMAT WAS TDF-',I2)
      READ (UIN) IWD
      WRITE (PRINT,6002) IWD
 6002 FORMAT (' THE WIND CODE IS',I2//
     1   ' YR MON DAY<',16('-'),'DIRECTION CODE',17('-'),
     2   '><',32('-'),'SPEED',32('-'),'>'/)
C     READ AND PRINT THE RECORDS.
      DO 20 I=1,NDAYS
      READ (UIN,END=999) IYR,IMO,IDA,DIR,SPD
      WRITE (PRINT,6003) IYR,IMO,IDA,DIR,SPD
 6003 FORMAT (I5,2I3,24I2,24I3)
   20 CONTINUE
  999 STOP
      END
```

WINDTRAN

Procedure: WINDTRAN
Executes program: WINDTRAN
Purpose: To calculate wind transition probability matrices for
         a wind station. A matrix is calculated for each of
         4 seasons.
Reads files: OSRA.RAWWIND.&STATION
Writes file: OSRA.WINDTRAN.&STATION
Text page reference: 6

Procedure listing:

```
//WINDTRAN PROC UVOL=CCDXXX
//*
//* WINDTRAN
//* EXECUTES PROGRAM WINDTRAN TO CONSTRUCT WIND TRANSITION MATRICES.
//* REQUIRED: STA=
//*
//* CLEAR THE OLD FILE (IF ANY).
//C EXEC PGM=IEFBR14
//OLD DD UNIT=3330,VOL=SER=&UVOL,DISP=(OLD,DELETE),
// DSN=OSRA.WINDTRAN.&STA
//*
//* EXECUTE PROGRAM WINDTRAN FROM OSRA.PGMLIB
//G EXEC PGM=WINDTRAN,REGION=120K
//STEPLIB DD UNIT=3330,VOL=SER=&UVOL,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&UVOL,DISP=SHR,
// DSN=OSRA.RAWWIND.&STA
//FT20F001 DD UNIT=3330,VOL=SER=&UVOL,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6444),SPACE=(TRK,(10,5),RLSE),
// DSN=OSRA.WINDTRAN.&STA
//FT30F001 DD UNIT=3330,VOL=SER=&UVOL,DISP=SHR,
// DSN=OSRA.WINDKEYS
```

Program listing:

```
C     PROGRAM WINDTRAN -- VERSION 3.0 -- 21 MAR 77.
C     CONSTRUCTS WIND TRANSITION PROBABILITY MATRICES FROM
C        THE OUTPUT OF RAWWIND.
C
      REAL WSPEED(2,41,4)
      INTEGER*2 WTRANS(41,41,4),IWTRNS(41,41,4)
      INTEGER*2 STEADY(41,4),IWSTDY(41,4)
      REAL STANAM(10)
      REAL TOXD(41,4),TOYD(41,4),TOSP(41,4)
      INTEGER NFROM(41,4),NTO(41,4)
      INTEGER*2 IYR,IMO,IDA,DIR(24),SPD(24)
      INTEGER*2 IYRB,IMOB,IYRE,IMOE
      INTEGER CARD/5/,PRINT/6/,UIN/10/,UOUT/20/,UANGLE/30/
      INTEGER IFROM(4)/4*0/
      REAL XV(37),YV(37)
      INTEGER IDIR(37,5)
      REAL ANGLE(37,5)
      REAL RADDEG/0.0174533/
      REAL*8 SEASON(4)/'WINTER','SPRING','SUMMER','AUTUMN'/
      REAL MONTH(12)/'JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG',
     1   'SEP','OCT','NOV','DEC'/
      INTEGER IPCNT(41)
      REAL ADIR(9)/'CALM','N','NE','E','SE','S','SW','W','NW'/
      INTEGER NSEA(4)
      REAL SEAROS(15,4),STAROS(15)
C
C     CLEAR MATRICES.
      CALL ZERO2(WTRANS,6724)
      CALL ZERO4(WSPEED,328)
      CALL ZERO4(NFROM,164)
      CALL ZERO4(NTO,164)
      CALL ZERO4(TOXD,164)
      CALL ZERO4(TOYD,164)
      CALL ZERO4(TOSP,164)
C
C     READ WIND DIRECTION CODE SYSTEM FILE.
      READ (UANGLE) IDIR,ANGLE
C     READ LIMITS OF RECORD TO BE USED AND DRIFT ANGLE.
      READ (CARD,5001) IYRB,IMOB,IYRE,IMOE,NHOUR,NUSE,NOFF,DRFTAN
 5001 FORMAT (7I3,F6.0)
C     IYRB,IMOB = STARTING YEAR AND MONTH
C     IYRE,IMOE = ENDING YEAR AND MONTH
C     NHOUR = SAMPLING INTERVAL, USED FOR PRINTOUT ONLY
C     NUSE = ACTUAL SAMPLING INTERVAL
C     NOFF = STARTING HOUR, 1 = MIDNIGHT.
C            IT IS IMPORTANT THAT 'NOFF' MATCHES THE ACTUAL SAMPLING
C            TIMES OF THE RAWWIND FILE, OR GOOD DATA WILL BE IGNORED.
C            CHECKING THE FULL OUTPUT OF LISTWIIND IS RECOMMENDED.
C     DRIFTAN = DRIFT ANGLE
C
C     READ HEADER RECORDS FROM RAWWIND FILE.
      READ (UIN) ITDF,ISTA,STANAM
      READ (UIN) KEY
C
C     PRINT THE HEADING.
      WRITE (PRINT,6001) ISTA,STANAM,MONTH(IMOB),IYRB,MONTH(IMOE),
     1   IYRE,DRFTAN,NHOUR,NUSE,NOFF,ITDF,KEY
 6001 FORMAT ('1PROGRAM WINDTRAN FOR',I6,1X,10A4/' USING THE RECORD FROM
     1 ',A3,I3,' TO ',A3,I3/' WITH A DRIFT ANGLE OF',F5.0,' DEGREES.'/
     2' NHOUR=',I2,' NUSE=',I2,' NOFF=',I2,' ITDF=',I2,' KEY=',I1)
C
```

```
C     SET UP THE PROPER VELOCITY CONVERSION VECTOR.
C     THIS VECTOR INCLUDES THE DRIFT ANGLE
      XV(1)=0.0
      YV(1)=0.0
      DO 10 I=2,37
      A=(ANGLE(I,KEY)+DRFTAN)*RADDEG
      XV(I)=SIN(A)
      YV(I)=COS(A)
   10 CONTINUE
C
C     SET UP THE MONTH INDICATORS.
      IMONTH=IMOB+12*IYRB
      LMONTH=IMOE+12*IYRE
C
C     DISCARD DATA UNTIL THE SPECIFIED STARTING TIME IS FOUND.
   20 CONTINUE
      READ (UIN) IYR,IMO,IDA,DIR,SPD
      IF (IMO+12*IYR.LT.IMONTH) GO TO 20
C
C     COME HERE WITH A NEW DAY.
   30 CONTINUE
C     DECEMBER IS THE FIRST MONTH OF THE FIRST SEASON.
      ISEA=(IMO+3)/3
      IF (ISEA.EQ.5) ISEA=1
C     NOTE THAT THE TRANSITION IS CALCULATED FROM THE LAST DAY OF
C        ONE SEASON TO THE FIRST DAY OF THE SAME SEASON IN THE
C        FOLLOWING YEAR.
      IFRM=IFROM(ISEA)
C
C     PROCESS THIS DAY.
C     SEE EARLIER COMMENT REGARDING NOFF
      DO 50 IHR=NOFF,24,NUSE
C     DISCARD INVALID VELOCITIES.
      IF (SPD(IHR).LT.0) GO TO 50
C     CONVERT SPEED TO 5 KNOT CLASSES.
      IS=(SPD(IHR)+2)/5
C     CLASSES ABOVE CALM ARE 5, 10, 15, 20, AND 25 KNOTS OR GREATER.
      IF (IS.GT.5) IS=5
C     DIRECTION CLASS IS DETERMINED FROM THE ACTUAL DIRECTION, AND
C        REPORTED AS SUCH, WITHOUT CONSIDERING THE DRIFT ANGLE.
C        HOWEVER, THE AVERAGES FOR THE CLASS INCLUDE THE DRIFT ANGLE.
C     CONVERT DIRECTION TO EIGHT DIRECTION CLASSES PLUS CALM.
      ID=IDIR(DIR(IHR),KEY)
C     CALCULATE THE VELOCITY CLASS.
      ITO=1
      IF (ID.GT.1.AND.IS.GT.0) ITO=ID*5+IS-9
C     IS THERE A TRANSITION?
      IF (IFRM.LE.0) GO TO 40
C
```

```
C     YES. RECORD IT.
      WTRANS(ITO,IFRM,ISEA)=WTRANS(ITO,IFRM,ISEA)+1
      NFROM(IFRM,ISEA)=NFROM(IFRM,ISEA)+1
      NTO(ITO,ISEA)=NTO(ITO,ISEA)+1
C     COLLECT THE AVERAGE VECTOR VELOCITY
      WSPEED(1,ITO,ISEA)=WSPEED(1,ITO,ISEA)+XV(DIR(IHR))*SPD(IHR)
      WSPEED(2,ITO,ISEA)=WSPEED(2,ITO,ISEA)+YV(DIR(IHR))*SPD(IHR)
C     COLLECT THE GROSS AVERAGE SPEED
      TOSP(ITO,ISEA)=TOSP(ITO,ISEA)+SPD(IHR)
C     COLLECT AVERAGE DIRECTION.
C     DIRECTION IS AVERAGED AS THE SUM OF UNIT VECTORS
      TOXD(ITO,ISEA)=TOXD(ITO,ISEA)+XV(DIR(IHR))
      TOYD(ITO,ISEA)=TOYD(ITO,ISEA)+YV(DIR(IHR))
   40 CONTINUE
      IFRM=ITO
   50 CONTINUE
C
C     READ NEXT DAY.
      IFROM(ISEA)=IFRM
      READ (UIN,END=60) IYR,IMO,IDA,DIR,SPD
C     IS THIS STILL IN THE DESIRED PORTION OF THE RECORD?
      IF (IMO+12*IYR.LE.LMONTH) GO TO 30
C
C     NO. ALL OF THE DESIRED RECORD HAS BEEN COLLECTED.
C
C     REPORT THE RESULTS BY SEASON.
   60 CONTINUE
      DO 140 ISEA=1,4
C
C     THE FOLLOWING ROUTINE CHECKS FOR A 'DEAD END' CONDITION, I.E.,
C        THE PROBABILITY OF ARRIVING AT A STATE IS NONZERO, BUT THE
C        PROBABILITY OF LEAVING THAT STATE IS ZERO. THIS COULD OCCUR
C        ONLY IF THE WIND RECORD FOR A SEASON ENDS IN A STATE WHICH IT
C        HAS NOT PREVIOUSLY ENCOUNTERED.
   61 CONTINUE
      DO 63 I=1,41
      IF(NFROM(I,ISEA).EQ.0.AND.NTO(I,ISEA).EQ.1) GO TO 64
   63 CONTINUE
C     DEAD END NOT ENCOUNTERED
      GO TO 65
C     DEAD END CONDITION ENCOUNTERED. CORRECT THE STATISTICS BY
C        DELETING THE LAST CONDITION.
   64 CONTINUE
      DO 62 J=1,41
      IF(WTRANS(I,J,ISEA).NE.0) NFROM(J,ISEA)=NFROM(J,ISEA)-1
      WTRANS(I,J,ISEA)=0
   62 CONTINUE
      NTO(I,ISEA)=0
      WSPEED(1,I,ISEA)=0.0
      WSPEED(2,I,ISEA)=0.0
      TOSP(I,ISEA)=0.0
      TOXD(I,ISEA)=0.0
      TOYD(I,ISEA)=0.0
C     REPEAT THE ROUTINE TO BE SURE THAT THE SECOND-TO-LAST RECORD HAS
C     NOT BECOME A DEAD END
      GO TO 61
C
C     REPORT THE AVERAGE WINDS
C     NOTE THAT THE DRIFT ANGLE DOES NOT AFFECT THE GROUPING OF DATA
C        BY CLASS, BUT DOES AFFECT THE AVERAGES WITHIN EACH CLASS.
   65 CONTINUE
      WRITE (PRINT,6002) NHOUR,SEASON(ISEA),ISTA,STANAM,DRFTAN
 6002 FORMAT ('1AVERAGE',I3,' HOUR WINDS FOR ',A6,' AT',I6,1X,10A4/
     1   ' USING A DRIFT ANGLE OF',F5.0,' DEGREES.'/
     2   ' ALL AVERAGES ARE BASED ON THE RAW DATA.'/
     3   ' SPEEDS ARE IN KNOTS AND DIRECTIONS ARE IN DEGREES.'//
     4   16X,'AVERAGE OF VECTORS',4X,'AVE. OF SCALARS',1X,
     5   'NO. OF POINTS'/
     6   '  DIR SPD X-COMP Y-COMP  SPEED DEGREE   SPEED DEGREE     TO',
     7   '   FROM'/)
C
C     ZERO THE SEASON COUNTERS.
      NF=0
      NT=0
      AXV=0.0
      AYV=0.0
      ASP=0.0
      AXD=0.0
      AYD=0.0
C
C     CALCULATE THE STATISTICS FOR EACH CLASS
      DO 80 ITO=1,41
C     EXTRACT THE SPEED AND DIRECTION CODES.
      ID=(ITO+8)/5
      IS=0
      IF (ITO.NE.1) IS=(ITO-ID*5+9)*5
C     ACCUMULATE SEASON TOTALS.
      N=NTO(ITO,ISEA)
      NT=NT+N
      NF=NF+NFROM(ITO,ISEA)
      AXV=AXV+WSPEED(1,ITO,ISEA)
      AYV=AYV+WSPEED(2,ITO,ISEA)
      ASP=ASP+TOSP(ITO,ISEA)
      AXD=AXD+TOXD(ITO,ISEA)
      AYD=AYD+TOYD(ITO,ISEA)
C     CALCULATE THE STATISTICS FOR THIS CLASS.
      TOSPD=0.0
      ADEG=0.0
      VSPD=0.0
      VDEG=0.0
      IF (N.EQ.0) GO TO 70
      WSPEED(1,ITO,ISEA)=WSPEED(1,ITO,ISEA)/N
      WSPEED(2,ITO,ISEA)=WSPEED(2,ITO,ISEA)/N
      VDEG=ATAN2(WSPEED(1,ITO,ISEA),WSPEED(2,ITO,ISEA))/RADDEG
      IF (VDEG.LT.0.0) VDEG=VDEG+360.0
      VSPD=SQRT(WSPEED(1,ITO,ISEA)**2+WSPEED(2,ITO,ISEA)**2)
      ADEG=ATAN2(TOXD(ITO,ISEA),TOYD(ITO,ISEA))/RADDEG
      IF (ADEG.LT.0.0) ADEG=ADEG+360.0
      TOSPD=TOSP(ITO,ISEA)/N
   70 CONTINUE
      WRITE (PRINT,6003) ADIR(ID),IS,WSPEED(1,ITO,ISEA),
     1   WSPEED(2,ITO,ISEA),VSPD,VDEG,TOSPD,ADEG,N,NFROM(ITO,ISEA)
 6003 FORMAT (1X,A4,I4,   2F7.0,F7.1,F7.0,F8.1,F7.0,2I7)
   80 CONTINUE
C
C     REPORT TOTALS.
      AXV=AXV/NT
      AYV=AYV/NT
      VDEG=ATAN2(AXV,AYV)/RADDEG
      IF (VDEG.LT.0.0) VDEG=VDEG+360.0
      VSPD=SQRT(AXV**2+AYV**2)
      ADEG=ATAN2(AXD,AYD)/RADDEG
      IF (ADEG.LT.0.0) ADEG=ADEG+360.0
      ASP=ASP/NT
      WRITE (PRINT,6004) AXV,AYV,VSPD,VDEG,ASP,ADEG,NT,NF
 6004 FORMAT (/' OVERALL ',2F7.0,F7.1,F7.0,F8.1,F7.0,2I7)
C
C     REPORT THE TRANSITION MATRIX.
      WRITE (PRINT,6005) NHOUR,SEASON(ISEA),ISTA,STANAM,DRFTAN,
     1   ADIR,((I,I=5,25,5),J=1,8)
 6005 FORMAT ('1',I2,' HOUR WIND TRANSITION MATRIX FOR ',A6,' AT',
     1   I6,1X,10A4/ ' USING A DRIFT ANGLE OF',F5.0,' DEGREES.'//
     2   30X,'NEXT WIND IS OUT OF ... AT SPEED ... IN PERCENT.'/
     3   9X,A4,8(4X,A3,4X)/' LAST',7X,8(1X,5I2)/
     4   ' FROM'/' DIR  SPD'/)
C
C     REPORT THE PERCENTAGES FOR EACH CLASS
      DO 100 IFRM=1,41
C     EXTRACT THE SPEED AND DIRECTION CODES.
      ID=(IFRM+8)/5
      IS=0
      IF (IFRM.NE.1) IS=(IFRM-ID*5+9)*5
C     CONVERT THE CLASS COUNTER TO PER CENT.
      D=0.0
      IF (NFROM(IFRM,ISEA).GT.0) D=100.0/NFROM(IFRM,ISEA)
      DO 90 ITO=1,41
      IPCNT(ITO)=WTRANS(ITO,IFRM,ISEA)*D+0.5
   90 CONTINUE
      WRITE (PRINT,6006) ADIR(ID),IS,IPCNT
 6006 FORMAT (1X,A4,I4,I3,8(1X,5I2))
      IF (IS.EQ.25.OR.IS.EQ.0) WRITE (PRINT,6007)
 6007 FORMAT (' ')
  100 CONTINUE
C     REPORT OVERALL
      D=100.0/NT
      DO 110 ITO=1,41
      IPCNT(ITO)=NTO(ITO,ISEA)*D+0.5
  110 CONTINUE
      WRITE (PRINT,6008) IPCNT
 6008 FORMAT (' OVERALL',I4,8(1X,5I2))
C
C     CONSTRUCT AND SORT THE CUMULATIVE TRANSITION MATRIX
C     SINCE THE WIND TRANSITION MATRIX MUST BE SAMPLED MANY TIMES IN
C        THE PROGRAM 'SPILL', IT IS DESIRABLE TO ARRANGE THIS MATRIX
C        SO THAT THE SEARCH ROUTINE OF 'SPILL' WILL ENCOUNTER THE MOST
C        FREQUENTLY OCCURRING CLASSES FIRST. THEREFORE, 'WTRANS' IS
C        FIRST SORTED IN ORDER OF DECREASING FREQUENCY BEFORE IT IS
C        CONVERTED INTO THE CUMULATIVE TRANSITION PROBABILITY MATRIX.
C        THE MATRIX 'IWTRANS' IS NECESSARY TO KEEP TRACK OF THE ORDER
C        IN WHICH THE CLASSES ARE SORTED.
      DO 120 IFRM=1,41
      CALL SORTW(WTRANS(1,IFRM,ISEA),IWTRNS(1,IFRM,ISEA),
     1   NFROM(IFRM,ISEA))
  120 CONTINUE
C     USE THE FREQUENCY COUNT OF "FROM" FOR THE STEADY STATE VECTOR.
      DO 130 IFRM=1,41
      STEADY(IFRM,ISEA)=NFROM(IFRM,ISEA)
  130 CONTINUE
      CALL SORTW(STEADY(1,ISEA),IWSTDY(1,ISEA),NF)
      NSEA(ISEA)=NF
  140 CONTINUE
C
C     WRITE OUT THE RESULTS TO THE FILE.
      WRITE (UOUT) ITDF,ISTA,STANAM,KEY,IYRB,IMOB,IYRE,IMOE,
     1   DRFTAN,NHOUR,NUSE,NOFF
      WRITE (UOUT) WTRANS
      WRITE (UOUT) IWTRNS
      WRITE (UOUT) WSPEED,STEADY,IWSTDY
C
C     COMPUTE AND REPORT WIND ROSE SUMMARIES BY SEASON AND STATION.
      CALL ZERO4(STAROS,15)
      CALL ZERO4(SEAROS,60)
      NSTA=0
      DO 160 ISEA=1,4
      CALM=NFROM(1,ISEA)
      SEAROS(1,ISEA)=CALM
      SEAROS(10,ISEA)=CALM
      STAROS(1)=STAROS(1)+CALM
      STAROS(10)=STAROS(10)+CALM
      NSTA=NSTA+NSEA(ISEA)
      DO 150 IFRM=2,41
      ID=(IFRM+8)/5
      IS=(IFRM-ID*5+9)+10
      SEAROS(ID,ISEA)=SEAROS(ID,ISEA)+NFROM(IFRM,ISEA)
      SEAROS(IS,ISEA)=SEAROS(IS,ISEA)+NFROM(IFRM,ISEA)
      STAROS(ID)=STAROS(ID)+NFROM(IFRM,ISEA)
      STAROS(IS)=STAROS(IS)+NFROM(IFRM,ISEA)
  150 CONTINUE
  160 CONTINUE
      DSTA=100.0/NSTA
      DO 190 IR=1,15
      STAROS(IR)=STAROS(IR)*DSTA
      DO 180 ISEA=1,4
      DSEA=100.0/NSEA(ISEA)
      SEAROS(IR,ISEA)=SEAROS(IR,ISEA)*DSEA
  180 CONTINUE
  190 CONTINUE
      WRITE (PRINT,6009) ADIR,ADIR(1),(I,I=5,25,5),SEAROS,STAROS
 6009 FORMAT ('1',11X,'DIRECTIONS IN PER CENT BY SEASON',16X,'SPEEDS',
     1   ' IN PER CENT BY SEASON'/20X,'WIND FROM (DEGREES)',27X,
     2'SPEED CLASS (KNOTS)'/' SEASON',1X,A4,1X,8(1X,A4),7X,A4,5I5/
     3' WINTER',9F5.0,8X,6F5.0/
     4' SPRING',9F5.0,8X,6F5.0/
     5' SUMMER',9F5.0,8X,6F5.0/
     6' AUTUMN',9F5.0,8X,6F5.0/
     7'  TOTAL',9F5.0,8X,6F5.0/)
      STOP
      END
C
C     SEE EARLIER COMMENTS REGARDING SORTING THE TRANSITION MATRIX
      SUBROUTINE SORTW(W,IT,N)
      INTEGER*2 W(1),IT(1),IH
      DO 10 I=1,41
      IT(I)=I
   10 CONTINUE
      IF (N.EQ.0) RETURN
      DO 30 I=1,40
      ITOP=42-I
      IMIN=1
      DO 20 J=2,ITOP
      IF (W(J).LE.W(IMIN)) IMIN=J
   20 CONTINUE
      IH=W(IMIN)
      W(IMIN)=W(ITOP)
      W(ITOP)=IH
      IH=IT(IMIN)
      IT(IMIN)=IT(ITOP)
      IT(ITOP)=IH
   30 CONTINUE
      DO 40 I=2,41
      W(I)=W(I)+W(I-1)
   40 CONTINUE
C     FREQUENCIES ARE NORMALIZED BETWEEN 0 AND 10000, FOR SAMPLING
C        WITH A FOUR DIGIT RANDOM NUMBER.
      D=10000.0/N
      DO 50 I=1,41
      W(I)=W(I)*D
   50 CONTINUE
      RETURN
      END
```

Procedure: WINDZONE
Executes program: WINDZONE
Purpose: Define the zones for each wind station
Reads cards: wind station numbers and names
             zone boundaries
Writes files: OSRA.&SALE.WINDZONE
Text page reference: 6

Procedure listing:

```
//WINDZONE PROC D=CCDXXX
//*
//* PROCEDURE WINDZONE
//* DEFINES THE ZONES OF EACH WIND STATION.
//* REQUIRED: &SALE
//*
//CLR EXEC PGM=IEFBR14
//DA DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),
// DSN=OSRA.&SALE.WINDZONE
//G EXEC PGM=WINDZONE
//FT10F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=6440,BLKSIZE=6444),SPACE=(TRK,(1,1),RLSE),
// DSN=OSRA.&SALE.WINDZONE
```

Program listing:

```
      REAL ZFAC(6)/6*1.0/
      INTEGER*2 IWIND(48,48)
      INTEGER NOWST(6),ZNAME(10)
      INTEGER CARD/5/,PRINT/6/,UWND/10/
      READ(CARD,5001)NSTA,NOWST,ZNAME
 5001 FORMAT(I2,6I7/10A4)
      WRITE(PRINT,6001) NSTA,(I,NOWST(I),I=1,NSTA)
 6001 FORMAT(' PROGRAM WINDZONE'/' THE',I2,' WIND STATIONS ARE:',
     1  I2,'-',I7,5(',',I1,'-',I7))
      WRITE(PRINT,6002)ZNAME
 6002 FORMAT(/' THIS IS ZONE PATTERN ',10A4//' THE ZONES ARE:')
  200 READ(5,5002,END=90)L1,L2,NUMB
 5002 FORMAT(3I5)
      DO 10 I=L1,L2
      DO 100 J=1,48
      IWIND(I,J)=NUMB
  100 CONTINUE
   10 CONTINUE
      GO TO 200
   90 CONTINUE
      DO 500 K=1,48
      J=49-K
      WRITE(PRINT,6003)J,(IWIND(I,J),I=1,48)
```

```
6003 FORMAT(I3,1X,48I1)
 500 CONTINUE
     WRITE(UWND)NSTA,NOWST,IWIND,ZNAME,ZFAC
     STOP
     END
```

Procedure: DEFTGT
Executes program: DEFTGT
Purpose: Defines the targets and creates target indicator matrices.
         Establishes the general file for program spill.
         The general file can be written in an incomplete form.
Reads files: OSRA.&SALE.TARGETS
Writes files: OSRA.&SALE.GENERAL.SPILL
Text page reference: 10

Procedure listing:

```
//DEFTGT PROC D=CCDXXX
//* PROCEDURE DEFTGT
//* EXECUTES PROGRAM DEFTGT TO DEFINE THE TARGETS.
//*
//* REQUIRED: &SALE
//*
//C EXEC PGM=IEFBR14
//C DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),
// DSN=OSRA.&SALE.GENERAL.SPILL
//G EXEC PGM=DEFTGT,REGION=200K,TIME=10
//STEPLIB DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.PGMLIB
//* FILE 20: THE TARGET MATRIX, STORED AS 30 X 30 DIRECT ACCESS BLOCKS.
//FT20F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.TARGETS
//* FILE 30: AN EXPANDED GENERAL FILE, FOR PROGRAM SPILL.
//*          THIS EXPANDED FILE CONTAINS THE INDICATOR MATRICES FOR
//*          THE TARGET BLOCKS.
//FT30F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// SPACE=(TRK,(10,5),RLSE),
// DCB=(RECFM=VBS,LRECL=6440,BLKSIZE=6444),
// DSN=OSRA.&SALE.GENERAL.SPILL
```

Program listing:

```
C
C     PROGRAM DEFTGT
C
C
C
C     INPUT:
C        CARD NUMBER 1:
C           COL. 01-08 SALE
C           COL. 09-10 BLANK
C           COL. 11-15 NUMBER OF TARGETS. (I5)
C           COL. 16-20 NUMBER OF LAND SEGMENTS, SET A. (I5)
C           COL. 21-25 NUMBER OF LAND SEGMENTS, SET B. (I5)
C           COL. 26-29 BLANK
C           COL. 30-30 NONZERO IF THIS IS A PARTIAL RUN.
```

```
C        CARD NUMBER 2:
C           COL. 01-02 TARGET NUMBER. (I2)
C           COL. 03-14 1 IF VULNERABLE IN MONTH DJFMAMJJASON. (12I1)
C           COL. 15-20 BLANK
C           COL. 21-80 TARGET NAME. (15A4)
C        REPEAT CARD NUMBER 2 FOR ALL TARGETS.
C
C
      INTEGER UIN/5/,UOUT/6/,UMAP/10/,UTGT/20/,UGENSP/30/
      INTEGER A(30,30),B(900)
      INTEGER DUM(12)
      INTEGER NSPBLK(16,16)/256*0/
      INTEGER FTGT(31)/31*0/
      INTEGER ITGT(12)
      REAL SALEG(2),SALE(2)
      INTEGER LHOME(12)/12*0/
      LOGICAL*1 HOME(12,31)/372*.FALSE./
      LOGICAL*1 LSGBLK(16,16)/256*.FALSE./
      INTEGER BITP(31)
      REAL TGTNAM(15,31)
      INTEGER TGTNO
      EQUIVALENCE (A(1,1),B(1))
      DEFINE FILE 20 (256,900,U,IP20)
C
C     SET THE BIT CORRESPONDING TO THE TARGET NUMBER.
      DO 10 IB=1,31
      BITP(IB)=2**(31-IB)
   10 CONTINUE
C
C     READ THE MAP PARAMETERS FROM THE GENERAL FILE.
C     THESE WILL BE TRANSFERRED TO THE GENERAL FILE FOR SPILL.
      READ (UMAP) SALEG,DUM
C
C     READ A HEADER CARD.
      READ (UIN,5001) SALE,NSPEC,NSEGA,NSEGB,IND
 5001 FORMAT (2A4,2X,3I5,4X,I1)
C
C     WRITE HEADINGS.
      WRITE (UOUT,6001) SALE,NSPEC,NSEGA,NSEGB
 6001 FORMAT ('OTARGETS FOR ',2A4//
     1   ' THERE ARE',I3,' TARGETS.'/
     2   ' LAND SEGMENT SET A HAS',I4,' SEGMENTS.'/
     3   ' LAND SEGMENT SET B HAS',I4,' SEGMENTS.'//
     4   'OTARGET    NAME',60X,'D J F M A M J J A S O N'/
     5   '  NO.'/)
C
C     CHECK FOR VALUES WITHIN RANGES.
      IF(NSPEC.GT.0.AND.NSPEC.LT.33.AND.NSEGA.GE.0.AND.NSEGA.LT.100.
     1   AND.NSEGB.GE.0.AND.NSEGB.LT.100) GO TO 15
```

```
      WRITE (UOUT,9003)
 9003 FORMAT ('OERROR - INVALID VALUE(S) FOR NSPEC, NSEGA, OR NSEGB.')
      GO TO 999
C
C     CHECK THE SALE NAMES ON THE HEADER RECORDS.
   15 CONTINUE
      IF (SALE(1).EQ.SALEG(1).AND.SALE(2).EQ.SALEG(2)) GO TO 20
C     THE HEADERS DO NOT MATCH.
      WRITE (UOUT,9001) SALE,SALEG
 9001 FORMAT ('OERROR - CARD SPECIFIES SALE ',2A4,', BUT THE MAP IS',
     1   ' FOR SALE ',2A4,'.')
      GO TO 999
C
C
C     READ A CARD IDENTIFYING A TARGET.
   20 CONTINUE
      READ(UIN,5002,END=40) TGTNO,ITGT,(TGTNAM(I,TGTNO),I=1,15)
 5002 FORMAT (I2,12I1,6X,15A4)
C     WRITE THE CARD CONTENTS.
      WRITE(UOUT,6002) TGTNO,(TGTNAM(I,TGTNO),I=1,15),ITGT
 6002 FORMAT ('0',I4,6X,15A4,4X,12I2)
C     CHECK FOR A VALID TARGET NUMBER.
      IF (TGTNO.LE.0.OR.TGTNO.GT.31) GO TO 22
C     SEE IF THIS TARGET HAS BEEN ENTERED BEFORE.
      IF (FTGT(TGTNO).EQ.1) GO TO 22
      FTGT(TGTNO)=1
      GO TO 30
C     IMPROPER TARGET NUMBER.
   22 CONTINUE
      WRITE (UOUT,9002)
 9002 FORMAT (' IMPROPER TARGET NUMBER')
      GO TO 20
C
C     PROCESS A TARGET.
   30 CONTINUE
      DO 31 I=1,12
      IF (ITGT(I).NE.1) GO TO 31
C     THE TARGET IS HOME.
      HOME(I,TGTNO)=.TRUE.
C     SET THE BIT.
      LHOME(I)=LOR(LHOME(I),BITP(TGTNO))
   31 CONTINUE
C     GO BACK FOR ANOTHER CARD.
      GO TO 20
C
C
C     READ EACH OF THE TARGET MATRIX BLOCKS TO CREATE AN INDICATOR
C        MATRIX, NSPBLK, WHICH INDICATES PRESENSE OR ABSENCE OF EACH
C        TARGET IN EACH 30 X 30 BLOCK.
```

```
   40 CONTINUE
C     SEE IF THE INDICATOR MATRICES ARE TO BE CREATED.
      IF (IND.EQ.0) GO TO 999
      DO 41 IBX=1,16
      DO 42 IBY=1,16
C     FIND AND READ THE BLOCK OF CELLS.
      IP20=(IBY-1)*16+IBX
      READ (UTGT'IP20) A
C     WORK AREA TO AVOID INDEXING WITHIN THE INNER LOOP.
      MA=0
C     CHECK ALL 900 CELLS OF THE BLOCK.
      DO 43 I=1,900
      MA=LOR(MA,B(I))
   43 CONTINUE
      NSPBLK(IBX,IBY)=MA
   42 CONTINUE
   41 CONTINUE
C
C     CREATE LSGBLK TO INDICATE THE POSSIBLE PRESENCE OF LAND SEGMENTS.
C     PROGRAM SPILL USES LSGBLK TO DETERMINE WHETHER OR NOT TO LOAD
C        LAND SEGMENT MATRICES.
C     SINCE THE TARGET 'LAND' IS THE ZEROTH BIT, A NEGATIVE VALUE
C        FOR NSPBLK(I,J) INDICATES THAT LAND IS PRESENT IN THE BLOCK.
C     NOTE THAT THIS ROUTINE WILL CAUSE PROGRAM SPILL TO CHECK FOR
C        LAND SEGMENTS ONLY WHEN LAND IS ACTUALLY PRESENT IN THE
C        BLOCK, AND THAT NON-LAND CELLS MARKED WITH LAND SEGMENT
C        CODES WILL, IN EFFECT, BE IGNORED.
      DO 51 IBX=1,16
      DO 52 IBY=1,16
      LSGBLK(IBX,IBY)=.FALSE.
      IF (NSPBLK(IBX,IBY).LT.0) LSGBLK(IBX,IBY)=.TRUE.
   52 CONTINUE
   51 CONTINUE
C
C
C     WRITE THE GENERAL FILE FOR PROGRAM SPILL.
C     EXCEPT FOR THE FIRST RECORD, WHICH CONTAINS THE MAP PARAMETERS,
C        THIS FILE IS COMPATIBLE WITH SPILL 6 TO 8.
   90 CONTINUE
      WRITE (UGENSP) SALE,DUM
      WRITE (UGENSP) NSPEC,NSPBLK,HOME,LHOME,TGTNAM
      WRITE (UGENSP) LSGBLK,NSEGA,NSEGB
C
C
C     END OF JOB.
  999 CONTINUE
      STOP
      END
```

Procedure: TGTPLOT
Executes program: TGTPLOT
Purpose: To plot each of the targets from the target matrix. In the normal sequence, the first file is a base map with latitude and longitude, the second file is a base map with the grid, and the other targets follow in order.
Reads files: OSRA.&SALE.GENERAL.SPILL
OSRA.&SALE.TARGETS
OSRA.&SALE.BASEMAP
Writes: Plotting instructions on magnetic tape for input to a Gerber Model 4400 Plotter.
Text page reference: 10

Procedure listing:

```
//TGTPLOT PROC D=CCDXXX,PTAPE=DUMMY
//*
//* PROCEDURE TGTPLOT
//G EXEC PGM=TGTPLOT,REGION=200K,TIME=30
//STEPLIB DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.GENERAL.SPILL
//* FILE 13: GERBER 4400 PLOT TAPE.
//FT13F001 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(NEW,PASS),
// DCB=(RECFM=FB,LRECL=72,BLKSIZE=1008,DEN=2),LABEL=(1,SL)
//FT13F002 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(2,SL)
//FT13F003 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(3,SL)
//FT13F004 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(4,SL)
//FT13F005 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(5,SL)
//FT13F006 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(6,SL)
//FT13F007 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(7,SL)
//FT13F008 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(8,SL)
//FT13F009 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(9,SL)
//FT13F010 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(10,SL)
//FT13F011 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(11,SL)
//FT13F012 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(12,SL)
```

```
//FT13F013 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(13,SL)
//FT13F014 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(14,SL)
//FT13F015 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(15,SL)
//FT13F016 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(16,SL)
//FT13F017 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(17,SL)
//FT13F018 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(18,SL)
//FT13F019 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=* FT13F001,
// LABEL=(19,SL)
//FT13F020 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(20,SL)
//FT13F021 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(21,SL)
//FT13F022 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(22,SL)
//FT13F023 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(23,SL)
//FT13F024 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(24,SL)
//FT13F025 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(25,SL)
//FT13F026 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(26,SL)
//FT13F027 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(27,SL)
//FT13F028 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(28,SL)
//FT13F029 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(29,SL)
//FT13F030 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
//   LABEL=(30,SL)
//FT13F031 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(31,SL)
//FT13F032 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(32,SL)
//FT13F033 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(33,SL)
//FT13F034 DD UNIT=TAPELO,VOL=SER=&PTAPE,DISP=(,PASS),DCB=*.FT13F001,
// LABEL=(34,SL)
//* FILE 20: THE TARGET MATRIX, STORED AS 30 X 30 DIRECT ACCESS BLOCKS.
//FT20F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.TARGETS
//* FILE 90: THE BASE MAP PLOT INSTRUCTIONS.
//FT90F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
```

// DSN=OSRA.&SALE.BASEMAP

Program listing:

```
C
C     PROGRAM TGTPLOT
C     PLOTS THE TARGETS FROM THE TARGET MATRIX.
C     EACH TARGET IS PLOTTED ON A SEPARATE FILE.
C     BASE MAPS WITH LAT/LON AND GRID ARE ALSO PLOTTED.
C     THE NOMINAL ORIGIN OF THE MAP IS (50,50) UNITS.
C     OPTIONAL PATTERNS ARE PROVIDED.
C
C     INPUT:
C        CARD:
C           CARD NUMBER 1:
C              COL. 01-08 SALE (2A4)
C              COL. 09-10 BLANK
C              COL. 11-15 MAP SCALE, UNITS PER INCH. (DEFAULT = 20.0)
C              COL. 16-18 BLANK
C              COL. 19-19 NONZERO TO DELETE BASE MAP WITH LAT/LON.
C              COL. 20-20 NONZERO TO DELETE BASE MAP WITH GRID.
C
C           CARD NUMBER 2:
C              COL. 01-32 PLOT PATTERN INDICATOR FOR EACH TARGET.
C                         TARGET 32 IS LAND.
C                         THE PATTERNS ARE:
C                            0 = NO PLOT.
C                            1 = HORIZONTAL LINE.
C                            2 = VERTICAL LINE.
C                            3 = COMBINE 1 AND 2.
C
C        DISK:
C           FT10F001 GENERAL FILE
C           FT20F001 TARGET FILE (DA)
C           FT90F001 BASE MAP FILE.
C
C     OUTPUT:
C        PRINT: LISTING OF FILES PLOTTED, INCLUDING THEIR
C               POSITIONS ON THE PLOT TAPE.
C
C        TAPE: FT13F001 GERBER 4400 PLOTTER TAPE.
C
C
      INTEGER UIN/5/,UOUT/6/,UGEN/10/,UTGT/20/,UBASE/90/
      REAL SALEG(2),SALE(2)
      INTEGER KFM/0/
      INTEGER A(30,30)
      INTEGER NSPBLK(16,16)
      INTEGER BITP(32)
      REAL TGTNAM(15,31)
      LOGICAL*1 HOME(12,31)
      INTEGER LHOME(12)
      INTEGER KPLOT(32)
      COMMON /PAT/A
      DEFINE FILE 20 (256,900,U,IP20)
C
C
C     SET THE BIT PATTERNS.
      DO 10 IB=1,31
      BITP(IB)=2**(31-IB)
   10 CONTINUE
C     TURN THE ZEROTH BIT ON.
      BITP(32)=2**31
C
C     READ THE CONTROL CARDS.
      READ (UIN,5001) SALE,UPI,IBL,IBG,KPLOT
 5001 FORMAT (2A4,2X,F5.2,3X,2I2/32I1)
C     DEFAULT VALUE FOR UPI IS 20.0
      IF (UPI.LE.0.0) UPI=20.0
C
C     READ THE GENERAL FILE.
      READ (UGEN) SALEG,IX,IY
      READ (UGEN)NSPEC,NSPBLK,HOME,LHOME,TGTNAM
C
C     CHECK THAT THE HEADERS ARE THE SAME.
      IF (SALE(1).EQ.SALEG(1).AND.SALE(2).EQ.SALEG(2)) GO TO 20
C     SALE NAMES DO NOT MATCH.
      WRITE (UOUT,9001) SALE,SALEG
 9001 FORMAT ('OERROR - CARD SPECIFIES SALE ',2A4,', BUT THE GENERAL ',
     1   'FILE IS FOR SALE ',2A4,'.')
      STOP 16
C
C     WRITE TITLE ON OUTPUT.
   20 CONTINUE
      WRITE (UOUT,6003) SALE,UPI
 6003 FORMAT ('1PROGRAM TARGPLOT FOR SALE ',2A4/
     1   ' THE SCALE IS ',F10.2,' UNITS PER INCH.')
C
C     PLOT A BASE MAP WITH LATITUDE AND LONGITUDE.
      IF (IBL.NE.0) GO TO 30
      CALL INIT(1)
      CALL SCAL ((1.0/UPI),(1.0/UPI))
      KFM=KFM+1
      CALL PLOTMP (SALE,50.0,50.0,0,1,UBASE)
      WRITE (UOUT,6004) KFM
 6004 FORMAT ('OBASE MAP WITH LAT/LON IS FILE MARK NUMBER',I4)
      CALL DONE
C
```

```
      REWIND 90
C     PLOT A BASE MAP WITH A GRID.
   30 CONTINUE
      IF (IBG.NE.0) GO TO 40
      CALL INIT(1)
      CALL SCAL ((1.0/UPI),(1.0/UPI))
      KFM=KFM+1
      CALL PLOTMP (SALE,50.0,50.0,10,0,UBASE)
      WRITE (UOUT,6005) KFM
 6005 FORMAT ('0BASE MAP WITH GRID IS FILE MARK NUMBER',I4)
      CALL DONE
C
C
C     PLOT THE TARGETS.
   40 CONTINUE
C
      DO 50 NTGT=1,32
C     LAND IS TARGET NUMBER 32.
C     BYPASS TARGETS WITH NO PLOT CODE.
      IF (KPLOT(NTGT).EQ.0) GO TO 50
C
C     TARGET TO BE PLOTTED.
C     RESET COUNTER FOR THE NUMBER OF BLOCKS THAT CONTAIN A TARGET.
      KBLK=0
C     INITIALIZE AND SCALE THE PLOTTER.
      CALL INIT(1)
      CALL SCAL ((1.0/UPI),(1.0/UPI))
C     INCREMENT THE FILE COUNTER.
      KFM=KFM+1
C
C     PROCESS EACH BLOCK.
      DO 60 IBY=1,16
      DO 62 IBX=1,16
C
C     BYPASS BLOCKS THAT DO NOT CONTAIN THE TARGET.
      IF (LAND(NSPBLK(IBX,IBY),BITP(NTGT)).EQ.0) GO TO 62
C
C     COUNT THE BLOCKS CONTAINING THE TARGET.
      KBLK=KBLK+1
C
C     READ THE TARGET MATRIX.
      IP20=(IBY-1)*16+IBX
      READ (UTGT'IP20) A
C
C     SELECT A SUBROUTINE TO PLOT THE PATTERN IN THE BLOCK.
      IF (KPLOT(NTGT).EQ.3)
     1   CALL PAT3 (50.0,50.0,IBX,IBY,BITP(NTGT))
      IF (KPLOT(NTGT).EQ.2)
     1   CALL PAT2 (50.0,50.0,IBX,IBY,BITP(NTGT))
```

```
      IF (KPLOT(NTGT).EQ.1)
     1   CALL PAT1 (50.0,50.0,IBX,IBY,BITP(NTGT))
C
   62 CONTINUE
   60 CONTINUE
C
C     FINISHED PLOTTING A TARGET. WRITE A MESSAGE.
      IF(NTGT.EQ.32) WRITE(UOUT,6011) NTGT,KBLK,KPLOT(NTGT)
 6011 FORMAT('OTARGET NUMBER ',I3/
     1 1X,'LAND'/
     2 ' PRESENT IN',I4,' BLOCKS.'/
     3 ' PLOTTING PATTERN NUMBER ',I2)
      IF(NTGT.EQ.32)GO TO 90
      WRITE (UOUT,6002) NTGT,(TGTNAM(I,NTGT),I=1,15),KBLK,KPLOT(NTGT)
 6002 FORMAT ('OTARGET NUMBER ',I3/
     1   1X,15A4/
     2   ' PRESENT IN',I4,' BLOCKS.'/
     3   ' PLOTTING PATTERN NUMBER',I2)
   90 CONTINUE
      WRITE (UOUT,6006) KFM
 6006 FORMAT (' FILE MARK NUMBER',I4)
C
C     ANNOTATE THE TARGET PLOT.
      CALL ALPHA2 (50.0,25.0,5.0,0.0,8,SALE)
C     CHECK IF THE TARGET IS LAND.
      IF (NTGT.EQ.32) GO TO 80
      CALL ALPHA2 (125.0,25.0,5.0,0.0,10,'TARGET NO.')
      CALL FXPLT (180.0,25.0,0.0,NTGT)
      CALL ALPHA2 (50.0,15.0,5.0,0.0,60,TGTNAM(1,NTGT))
      GO TO 52
C
C     ANNOTATION FOR LAND.
   80 CONTINUE
      CALL ALPHA2 (50.0,15.0,5.0,0.0,31,
     1   'INTERNAL REPRESENTATION OF LAND')
C
C     COMPLETED THIS TARGET PLOT.
   52 CONTINUE
      CALL DONE
C
C     GO BACK FOR ANOTHER TARGET.
   50 CONTINUE
      STOP
      END
```

SEGMATRX

Procedure: SEGMATRX
Executes program: SEGMATRX
Purpose: Merges two direct access files, land segments 1 and 2.
Reads files: OSRA.&SALE.DEFPOLY.DRAX.&FILEA
             OSRA.&SALE.DEFPOLY.DRAX.&FILEB
Writes files: OSRA.&SALE.SEGMATRX
Text page reference: 12

Procedure listing:

```
//SEGMATRX  PROC  U=3330,UVOL=CCDXXX,SALE=ANY,
//   FILEA=SMITH,FILEB=JONES
//*                 MERGES FILES 30 AND 40 INTO FILE 50.
//CLEAR  EXEC  PGM=IEFBR14
//T   DD  UNIT=&U,VOL=SER=&UVOL,DISP=(OLD,DELETE),
//   DSN=OSRA.&SALE.SEGMATRX
//A  EXEC  PGM=SEGMATRX
//STEPLIB  DD  UNIT=&U,VOL=SER=&UVOL,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001  DD  DDNAME=SYSIN
//FT06F001  DD  SYSOUT=A
//FT30F001  DD  UNIT=&U,VOL=SER=&UVOL,
//   DISP=SHR,DCB=DSORG=DA,SPACE=(1800,(256)),
//   DSN=OSRA.&SALE.DEFPOLY.DRAX.&FILEA
//FT40F001  DD  UNIT=&U,VOL=SER=CCD909,
//   DISP=SHR,DCB=DSORG=DA,SPACE=(1800,(256)),
//   DSN=OSRA.&SALE.DEFPOLY.DRAX.&FILEB
//FT50F001  DD  UNIT=&U,VOL=SER=CCD909,
//   DISP=(NEW,KEEP),DCB=DSORG=DA,SPACE=(1800,(256)),
//   DSN=OSRA.&SALE.SEGMATRX
```

Program listing:

```
C     PROGRAM SEGMATRX
C     MERGES TWO DIRECT ACCESS FILES
C     LAND SEGMENTS 1 AND 2
C
      INTEGER*2 LMAPA(900),LMAPB(900),LMAP(900)
      DEFINE FILE30(256,450,U,IP30)
      DEFINE FILE40(256,450,U,IP40)
      DEFINE FILE50(256,450,U,IP50)
      DO 200 K=1,256
      READ(30'K) LMAPA
      READ(40'K) LMAPB
      DO 100 J=1,900
      LMAP(J) = 100*LMAPA(J)+LMAPB(J)
  100 CONTINUE
      WRITE(50'K) LMAP
  200 CONTINUE
      STOP
```

```
END
```

Procedure: SPILL
Executes program: SPILL
Purpose: To simulate oilspill trajectories in a Monte Carlo fashion and to record the number of oilspill contacts to targets and land segments.
Reads files: OSRA.&SALE.GENERAL.SPILL
OSRA.&SALE.WINDZONE
OSRA.&SALE.CURVAL
OSRA.&SALE.&WSTA1
OSRA.&SALE.&WSTA2
OSRA.&SALE.&WSTA3
OSRA.&SALE.&WSTA4
OSRA.&SALE.&WSTA5
OSRA.&SALE.&WSTA6
OSRA.&SALE.CURMATRX
OSRA.&SALE.TARGETS
OSRA.&SALE.SEGMATRX
OSRA.&SALE.LAUNCH.POINTS
Writes files: OSRA.&SALE.&RUN
Text page reference: 12

Procedure listing:

```
//SPILL PROC D=CCDXXX,WSTA2=DUMMY,WSTA3=DUMMY,WSTA4=DUMMY,WSTA5=DUMMY,
// WSTA6=DUMMY
//C EXEC PGM=IEFBR14
//CLEAR DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),DSN=OSRA.&SALE.&RUN
//G EXEC PGM=&PROG,REGION=510K,TIME=5
//STEPLIB DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT10F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.GENERAL.SPILL
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.&SALE.WINDZONE
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.&SALE.CURVAL
//FT20F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.WINDTRAN.&WSTA1
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.WINDTRAN.&WSTA2
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.WINDTRAN.&WSTA3
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.WINDTRAN.&WSTA4
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.WINDTRAN.&WSTA5
// DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.WINDTRAN.&WSTA6
//FT30F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=6440,BLKSIZE=6444),SPACE=(TRK,(50,10),RLSE),
// DSN=OSRA.&SALE.&RUN
//FT40F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.&SALE.CURMATRX
//FT50F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.&SALE.TARGETS
//FT60F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.&SALE.SEGMATRX
//FT70F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.&SALE.LAUNCH.POINTS
```

Program listing:

```
C     TRAJECTORY PROGRAM SPILL.
C     SPILL USES MERCATOR NAVIGATION
C     SPILL INCORPORATES THE 'LINE SOURCE' FEATURE
C     SPILL INCORPORATES THE "AREA SOURCE" FEATURE
C     SPILL IS DESIGNED TO BE COMPATIBLE WITH THE M&S DIGITIZER
C        ROUTINES. THESE ROUTINES INCORPORATE A MERCATOR GRID, AND
C        BY EXPRESSING GRID COORDINATES AS A LINEAR TRANSFORMATION OF
C        MERCATOR COORDINATES, ELIMINATE THE NEED FOR REFERENCE
C        POINTS.
C     THE CELL COORDINATES OF POINT (X,Y) ARE EXPRESSED AS INTEGERS
C        (X+1,Y+1). THIS IS A SLIGHT CHANGE FROM EARLIER TRAJECTORY
C        PROGRAMS.
C     THIS VERSION RECORDS THE MONTH IN WHICH EACH SPILL FIRST HITS
C        THE TARGETS.
C     SPILL ALSO USES CURRENT DATA IN METERS PER SECOND, RATHER THAN
C        KNOTS, AS A STEP TOWARDS METRIFICATION. THE WIND VELOCITIES,
C        HOWEVER, ARE STILL IN KNOTS.
C     THIS VERSION INCORPORATES A FASTER MERCATOR NAVIGATION ROUTINE
C        THAN EARLIER VERSIONS.
C     THE MAXIMUM NUMBER OF CURRENT POLYGONS IS 1000.
C     550 KILOBYTES OF STORAGE IS REQUIRED FOR THIS VERSION.
C      THIS VERSION INCORPORATES A VARIABLE WIND FACTOR
C      FOR EACH WIND STATION
C
C
C
      REAL TNAMES(10)
      INTEGER*2 XLOC(100),YLOC(100)
      DEFINE FILE 70(100,110,U,IP70)
      REAL XCUR(1000,12),YCUR(1000,12)
      REAL VERSIN(4)/'9.3','12','FEB','80'/
      REAL ZFAC(6)
      REAL WHEN(7)
      REAL SALE(2)
      REAL SALEG(2)
      COMMON /BLOCK/ OBJ,ICUR,NSPBLK,LSGBLK,LSEG
      INTEGER OBJ(30,30)
      INTEGER NSPBLK(16,16)
      INTEGER*2 ICUR(30,30),LSEG(30,30)
      LOGICAL*1 LSGBLK(16,16)
      COMMON /SEED/ ISEED,IUSED
      LOGICAL*1 HOME(12,31)
      INTEGER LHOME(12)
      REAL WSPEED(2,41,4,6)
      INTEGER*2 WTRANS(41,41,4,6),STEADY(41,4,6)
      INTEGER*2 IWTRNS(41,41,4,6),IWSTDY(41,4,6)
      INTEGER*2 IWIND(48,48)
      INTEGER NOWST(6)
      REAL OBJNAM(15,31),WSTNAM(10)
      REAL RUNAME(10),ZNAME(10)
      REAL CURNAM(10),MPC
      INTEGER OFFIDX(10)/3,7,0,1,2,8,0,5,4,6/
      INTEGER OFFMAP(8)
      REAL OMN(8)/' N','NE',' E','SE',' S','SW',' W','NW'/
      REAL*8 SEANAM(4)/'WINTER','SPRING','SUMMER','AUTUMN'/
      INTEGER*2 HTIME(32,100)
      INTEGER*2 HMONTH(32,100)
      INTEGER*2 SAVEXY(3,100)
      INTEGER TOLAND(31)
      INTEGER CHIT(31)
      INTEGER BITP(31)
C     MERCATOR NAVIGATION CONSTANTS.
      REAL EPSQ/6.768657E-3/
      REAL SECRAD/4.848137E-6/
      REAL ASU/30.71313/
      REAL SECMET/30.92241/
      REAL EPSIL/8.227185E-2/,PION4/0.7853982/,SECRA2/2.424068E-6/,
     1   EPSIL2/4.113593E-2/
      REAL A/6378206.4/
C     PROGRAM PERFORMANCE COUNTERS.
      COMMON /COUNT/KNCB,KNCBR,KNOB,KNOBR,KNSB,KNSBR
      INTEGER KSM/0/
C     I/O DEVICES.
      INTEGER CARD/5/,PRINT/6/,UGEN/10/,UWIND/20/,UOUT/30/
C
C     ZERO THE PROGRAM PERFORMANCE COUNTERS.
      KNCB=0
      KNCBR=0
      KNOB=0
      KNOBR=0
      KNSB=0
      KNSBR=0
C
C     WHAT TIME IS IT?
      CALL NOW(WHEN)
C
C     SET THE BIT CORRESPONDING TO THE OBJECT NUMBER.
      DO 10 IB=1,31
      KB=31-IB
      BITP(IB)=2**KB
   10 CONTINUE
C
C     DEFINE THE MODEL'S CONTROL PARAMETERS.
C     NC=NUMBER OF CELLS PER BLOCK.
      NC=30
C     MAXDAY MAXIMUM NUMBER OF DAYS TO TRACK A SPILL (<=90).
```

```
      MAXDAY=30
C     NTP=NUMBER OF TIME PERIODS PER DAY.
C     SPILL USES 3-HOUR TIME PERIODS.
      NTP=4
      TP=24.0/NTP
      MAXTP=MAXDAY*NTP
C     THE WIND SPEED FACTOR IS 3.5 PERCENT.
C     (THE DRIFT ANGLE IS INCLUDED IN THE WIND TRANSITION MATRICES.)
      WNDFAC=0.035
C     WFAC CONVERTS KNOTS TO METERS PER TIME PERIOD
      WFAC=WNDFAC*TP*1852.0
C     CURFAC CONVERTS METERS PER SECOND TO METERS PER TIME PERIOD.
      CURFAC=TP*3600.0
C
C     READ THE RUN PARAMETERS.
C     'SALE' IS A UNIQUE 8-CHARACTER IDENTIFIER FOR EACH LEASE SALE.
C     NTRY IS THE NUMBER OF TRAJECTORIES,PER SEASON, THAT ARE
C        STARTED AT EACH LAUNCH POINT.
C     IXS IS THE RANDUM NUMBER SEED.
C        SINCE THE TRAJECTORY PROGRAM IS USUALLY RUN IN SEVERAL
C        SECTIONS, IT IS IMPORTANT TO SELECT THE RANDUM NUMBER
C        SEEDS FOR EACH SECTION SUFFICIENTLY FAR APART IN THE TABLE
C        OF RANDUM NUMBERS SO THAT THE TRAJECTORIES WILL NOT
C        REPEAT ONE ANOTHER. CONSULT THE TABLE OF RANDUM NUMBERS
C        GENERATED BY THIS PARTICULAR RANDUM NUMBER FUNCTION TO
C        SELECT PROPER VALUES FOR IXS
      READ (CARD,5001) SALE,NTRY,IXS
 5001 FORMAT (2A4,I3,I10)
C
C     READ BASE MAP AND GRID INDICATORS.
      READ (UGEN) SALEG,KXMAX,KYMAX,A1,B1,C1,A2,B2,C2
C     SPILL USES A GRID WITH LOWER LIMITS OF ZERO.
      KXMIN=1
      KYMIN=1
C     CHECK FOR MATCHING FILES.
C     THIS CHECK AUTOMATICALLY INTERLOCKS THE TRAJECTORY RUN WITH
C        THE PROPER GENERAL FILE. IT DOES NOT ASSURE USE OF THE
C        CORRECT WIND, CURRENT, OR TARGET FILES. THESE MUST BE CHECKED
C        BY INSPECTING THE JOB CONTROL MESSAGES. SINCE PROGRAM SPILL
C        IS USUALLY RUN FROM A CATALOGUED PROCEDURE FOR EACH SALE,
C        CHECKING THE GENERAL FILE IS USUALLY SUFFICIENT.
      IF (SALE(1).NE.SALEG(1).OR.SALE(2).NE.SALEG(2)) GO TO 3020
C
C     READ OBJECT AND LAND SEGMENT INDICATOR MATRICES.
C        THE ACTUAL DATA MATRICES ARE READ INTO CORE LATER,
C        AS NEEDED, NY SUBROUTINES.
      READ (UGEN) NSPEC,NSPBLK,HOME,LHOME,OBJNAM
      READ (UGEN) LSGBLK,NSEGA,NSEGB
C
```

```
C     WRITE THE HEADINGS TO IDENTIFY THE RUN AND THE OBJECTS.
      WRITE (PRINT,6000) SALE,NTRY,IXS,WNDFAC,VERSIN,WHEN,
     1   NSPEC,(IS,(OBJNAM(K,IS),K=1,15),(HOME(IMO,IS),IMO=1,12),
     2   IS=1,NSPEC)
 6000 FORMAT ('1',2A4,' TRAJECTORY SIMULATION:',I5,' STARTS USING',
     1   ' THE SEED',I10,' AND WIND FACTOR',F6.3/
     2   ' THIS IS VERSION ',A3,' OF ',A3,A4,A2,4X,
     3   ' THE TIME OF THIS RUN IS ',I2,1X,A3,I3,' @ ',2(I2,':'),I2,
     *   1X,A2
     4   /' THE',I3,' OBJECTS OF INTEREST ARE'/(I5,1X,15A4,4X,12L2))
C
C     READ AND STORE THE WIND TRANSITION DATA.
C     THE WIND TRANSITION MATRIX IS DESIGNED FOR A FAST SEARCH ROUTINE.
C     SEE THE LISTING OF PROGRAM 'WINDTRAN' FOR AN EXPLANATION OF
C        HOW THE WIND TRANSITION DATA IS ARRANGED.
C     READ THE WIND STATION PARAMETERS.
      READ(UGEN) NWST,NOWST,IWIND,ZNAME,ZFAC
      WRITE (PRINT,6001) ZNAME,NWST
 6001 FORMAT (/' THE WIND ZONE PATTERN IS ',10A4/
     1   ' THE',I2,' WIND STATIONS ARE:'/
     2   ' NO DRIFT TIME    STA  STATION NAME'/
     3   '     ANGLE INCR     NO')
C     READ THE WIND TRANSITION MATRICES.
      DO 20 IWST=1,NWST
      READ (UWIND) ITDF,ISTA,WSTNAM,KEY,DUM1,DUM2,DRFTAN,NHOUR
C     CHECK FOR THE CORRECT WIND STATIONS. THE ORDER OF WIND STATIONS
C        CALLED FOR IN THE JCL MUST MATCH THE ORDER SPECIFIED
C        WHEN THE WIND ZONE MATRIX WAS CREATED.
      IF (ISTA.NE.NOWST(IWST)) GO TO 3010
C     WTRANS CONTAINS THE TRANSITION PROBABILITIES.
C     IWTRNS IS A SORT KEY FOR WTRANS.
C     WSPEED CONTAINS THE WIND VELOCITIES FOR EACH CLASS.
C     DRIFT ANGLE EFFECTS ARE INCLUDED IN WSPEED.
C     STEADY CONTAINS THE OVERALL CLASS PROBABILITIES, FOR STARTING.
C     IWSTDY IS A SORT KEY FOR STEADY.
      READ (UWIND) (WTRANS(I,1,1,IWST),I=1,6724)
      READ (UWIND) (IWTRNS(I,1,1,IWST),I=1,6724)
      READ (UWIND) (WSPEED(I,1,1,IWST),I=1,328),
     1   (STEADY(I,1,IWST),I=1,164),(IWSTDY(I,1,IWST),I=1,164)
      WRITE(PRINT,6002) IWST,DRFTAN,NHOUR,ISTA,WSTNAM,ZFAC(IWST)
 6002 FORMAT (I3,F6.1,I5,I7,2X,10A4,2X,F5.2)
   20 CONTINUE
C
C     READ CURRENT.
C     CURRENTS ARE IN METERS PER SECOND.
      READ (UGEN) CURNAM
      READ (UGEN) XCUR
      READ (UGEN) YCUR
      WRITE (PRINT,6003) CURNAM,TP,MAXDAY
 6003 FORMAT (' CURRENT DATA ARE FROM: ',10A4/
     1   ' USING',F4.0,' HOUR TIME PERIODS FOR AT MOST',I3,' DAYS.'//
     2   ' THE RESULTS ARE:')
C
C
C
C     COME HERE TO START A SIMULATION FROM A LOCATION.
   30 CONTINUE
C
C     READ THE STARTING LOCATION.
      READ(CARD,5002,END=999)JFLAG,X0,Y0,XT0,YT0,RUNAME
 5002 FORMAT(I5,4F5.0,10A4)
      IF(JFLAG.LE.0)GO TO 32
C     DISTRIBUTED LAUNCH POINTS - READ FROM DISK
      READ(70'JFLAG)TNAMES,XLOC,YLOC
C     CHECK FOR LAUNCH POINT MISMATCH
      IF(RUNAME(1).NE.TNAMES(1))GO TO 3040
      GO TO 33
C
C     *****************************************************************
C
C                                 WARNING
C
C     THIS UNREFRENCED CONTINUE STATEMENT IS NECESSARY TO BYPASS AN
C        ERROR IN THE FORTRAN H COMPILER, OPTIMIZATION LEVEL 2.
C        WITHOUT THE LABEL, THE OPTIMIZER INCORRECTLY RECOGNIZES A
C        LOOP, AND PRODUCES INCORRECT CODE. THIS RESULTS IN AN '0C4'
C        ERROR.
   31 CONTINUE
C
C
C     *****************************************************************
C     IF ENDING POINTS ARE BLANK, USE A SINGLE LAUNCH POINT.
C     THE LAUNCH POINT MOVES IN EQUAL INCREMENTS BETWEEN (X0,Y0)
C        AND (XT0,YT0).
   32 CONTINUE
      IF (XT0.EQ.0.0) XT0=X0
      DX0=(XT0-X0)/(NTRY-1)
      IF (YT0.EQ.0.0) YT0=Y0
      DY0=(YT0-Y0)/(NTRY-1)
C
C     X0,Y0 ARE OVERALL GRID LOCATIONS FRO THE STARTING POINT.
C     XT0,YT0 ARE THE OVERALL GRID LOCATIONS FROM THE ENDING POINT.
C     IX,IY ARE OVERALL CURRENT CELL LOCATIONS.
C     LX,LY ARE OVERALL PREVIOUS CELL LOCATIONS.
C     IXB,IYB ARE CURRENT (NEW) BLOCK INDICIES.
C     IXC,IYC ARE CURRENT LOCATIONS (CELLS) IN A BLOCK.
C     LXB,LYB ARE LOADED BLOCK INDICIES.
C     XB,YB ARE THE PRESENT GRID LOCATION.
```

```
C     XBL,YBL ARE THE PREVIOUS GRID LOCATION.
C
C     INITIATE THE RANDUM NUMBER SELECTOR.
   33 CONTINUE
      ISEED=IXS
      IUSED=100
C
C     WRITE A HEADER RECORD DESCRIBING THE RUN.
      WRITE (UOUT) NTRY,X0,Y0,VERSIN,WHEN,WNDFAC,OBJNAM,NSPEC,IXS,
     1   RUNAME,NTP,NSEGA,NSEGB,SALE,IXT0,IYT0
C
C     PRINT THE DESCRIPTION OF THE RUN.
      WRITE (PRINT,6004) X0,Y0,XT0,YT0,RUNAME,(IS,IS=1,NSPEC)
 6004 FORMAT (/' FROM ',2F8.2,' TO ',2F8.2,'  RUN - ',10A4/
     3   ' SEASON HIT DE- OFF DAYS-TO-LAND ',
     4   'NUMBER-OF-HITS-ON-EACH-OBJECT',T112,'NUMBER-OFF-MAP'/
     5   '        LAND CAY MAP MIN AVE MAX',31I3)
C
C
C
C     SIMULATE FOR EACH SEASON.
      DO 190 ISEA=1,4
C
C     INITIALIZE OVERALL COUNTERS.
      CALL ZERO4(CHIT,31)
      CALL ZERO4(OFFMAP,8)
      CALL ZERO4(TOLAND,31)
      CALL ZERO2(HTIME,32*100)
      CALL ZERO2 (HMONTH,32*100)
      CALL ZERO2(SAVEXY,300)
      LANDED=0
      IDK=0
      NOFF=0
      MEANTL=0
      MAXTL=0
      MINTL=MAXTP
      ISDAY=(ISEA-1)*90
C
C
C
C     SIMULATE ONE SPILL.
      DO 180 IB=1,NTRY
C
C     INITIALIZE LOCATION.
C     INTEGERS ARE USED TO IDENTIFY GRID CELLS, BUT THE ACTUAL GRID
C        LOCATION IS A FLOATING POINT NUMBER.
      IF(JFLAG.GT.0)XB=XLOC(IB)
      IF(JFLAG.GT.0)YB=YLOC(IB)
      IF(JFLAG.LE.0)XB=X0+DX0*(IB-1)
      IF(JFLAG.LE.0)YB=YO+DYO*(IB-1)
      IX=XB+1
      IY=YB+1
      XBL=XB
      YBL=YB
      LX=IX
      LY=IY
C     STARTING LATITUDE AND LONGITUDE.
      XM=(XB-A1)/B1
      YM=(YB-A2)/C2
      CALL CMELL (XM,YM,SLAT,SLON,IERR)
      IF (IERR.NE.0) GO TO 3030
      RSLAT=SLAT*SECRAD
      SINRS=SIN(RSLAT)
C
C     LOAD STARTING BLOCK DATA.
      IXB=(IX-1)/NC+1
      IYB=(IY-1)/NC+1
      CALL NEWBLK(IXB,IYB,LXB,LYB)
      IXC=IX-(IXB-1)*NC
      IYC=IY-(IYB-1)*NC
C
C     SELECT STARTING DAY WITHIN SEASON.
C     DECEMBER 1 IS THE FIRST DAY OF THE YEAR.
      IDAY=90*RANDUB(ISEED)+ISDAY
C
C     SELECT STARTING WIND.
      IWST=IWIND((IX+9)/10,(IY+9)/10)
      IW=IPICK(STEADY(1,ISEA,IWST),IWSTDY(1,ISEA,IWST))
      JSEA=ISEA
C
C     ZERO HIT INDICATORS.
      KHIT=0
      LHIT=0
C
C
C     SPILL MOVEMENT ROUTINE.
      DO 130 IT=1,MAXTP
C
C     COUNT THE NUMBER OF SPILL MOVEMENTS, TO MONITOR
C        PROGRAM PERFORMANCE.
      KSM=KSM+1
C
C     NEW SEASON?
      JDAY=IDAY+IT/NTP
      IMO=MOD(JDAY/30,12)+1
      JSEA=(IMO+2)/3
C
C     PICK WIND AND CURRENT SPEED.
```

```
      IWST=IWIND((IX+9)/10,(IY+9)/10)
      IW=IPICK(WTRANS(1,IW,JSEA,IWST),IWTRNS(1,IW,JSEA,IWST))
      IC=ICUR(IXC,IYC)
C
C     MOVE THE SPILL.
C     MOVEMENT IN METERS THIS TIME PERIOD.
      XWIND=WSPEED(1,IW,JSEA,IWST)
      YWIND=WSPEED(2,IW,JSEA,IWST)
      DELTAX=ZFAC(IWST)*WFAC*XWIND+CURFAC*XCUR(IC,IMO)
      DELTAY=ZFAC(IWST)*WFAC*YWIND+CURFAC*YCUR(IC,IMO)
C     LATITUDE AND LONGITUDE MOVED.
      SSQPHI=SINRS**2
      UMESSQ=1.0-EPSQ*SSQPHI
      SLON=SLON-DELTAX*SQRT(UMESSQ/(1.0-SSQPHI))/SECMET
      SLAT=SLAT+DELTAY*UMESSQ**1.5/ASU
      XM=SECMET*SLON
      ASLAT=ABS(SLAT)
      RSLAT=ASLAT*SECRAD
      SINRS=SIN(RSLAT)
      ESINPH=EPSIL*SINRS
      YM=A*ISGN(SLAT)*ALOG(TAN(PION4+ASLAT*SECRA2)*
     1   ((1.0-ESINPH)/(1.0+ESINPH))**EPSIL2)
C     NEW GRID LOCATION.
      XB=A1+B1*XM
      YB=A2+C2*YM
      IX=XB+1
      IY=YB+1
C
C     IS THE SPILL OFF THE MAP?
      IF (IX.LT.KXMIN.OR.IX.GT.KXMAX.OR.IY.LT.KYMIN.OR.IY.GT.KYMAX)
     1   GO TO 140

C     CHECK EACH CELL PASSED THRU.
      IF (IX.EQ.LX) GO TO 80
C
C     NON-VERTICAL PATH.
      SLOPE=(YB-YBL)/(XB-XBL)
      IXDO=IABS(IX-LX)+1
      IXADD=ISGN(IX-LX)
      XADD=IXADD
      IYADD=ISGN(IY-LY)
      XN=LX-1
      IF (IXADD.LT.0) XN=XN-XADD
      LX=LX-IXADD
      DO 70 JX=1,IXDO
      LX=LX+IXADD
      IXB=(LX-1)/NC+1
      IXC=LX-(IXB-1)*NC
      XN=XN+XADD
      IF (JX.EQ.IXDO) XN=XB
      KY=((XN-XBL)*SLOPE+YBL)+1
      IYDO=IABS(KY-LY)+1
      LY=LY-IYADD
      DO 60 JY=1,IYDO
      LY=LY+IYADD
      IYB=(LY-1)/NC+1
      IYC=LY-(IYB-1)*NC
C     CHECK THAT THE CORRECT OBJECT BLOCK IS IN PLACE.
      IF (IYB.NE.LYB.OR.IXB.NE.LXB) CALL NEWBLK(IXB,IYB,LXB,LYB)
C     SEE IF A HIT HAS OCCURRED.
      LHIT=LOR(LHIT,LAND(OBJ(IXC,IYC),LHOME(IMO)))
      IF (LHIT.EQ.KHIT) GO TO 50
C     A HIT HAS OCCURRED. WHAT WAS HIT?
      JHIT=LXOR(KHIT,LHIT)
      DO 40 IS=1,NSPEC
C     RECORD THE HIT TIME FOR EACH OBJECT HIT.
      IF (LAND(JHIT,BITP(IS)).EQ.0) GO TO 40
      HTIME(IS,IB)=IT
C     RECORD MONTH TARGET IS FIRST HIT.
      HMONTH(IS,IB)=IMO
   40 CONTINUE
      KHIT=LHIT
   50 CONTINUE
      IF (OBJ(IXC,IYC).LT.0) GO TO 150
   60 CONTINUE
   70 CONTINUE
      GO TO 120
C
C     VERTICAL PATH.
   80 CONTINUE
      IYADD=ISGN(IY-LY)
      IYDO=IABS(IY-LY)+1
      LY=LY-IYADD
      DO 110 JY=1,IYDO
      LY=LY+IYADD
      IYB=(LY-1)/NC+1
      IYC=LY-(IYB-1)*NC
C     SEE THAT THE CORRECT OBJECT MATRIX IS IN PLACE.
      IF (IYB.NE.LYB) CALL NEWBLK(IXB,IYB,LXB,LYB)
C     SEE IF A HIT HAS OCCURRED.
      LHIT=LOR(LHIT,LAND(OBJ(IXC,IYC),LHOME(IMO)))
      IF (LHIT.EQ.KHIT) GO TO 100
C     A HIT HAS OCCURRED. WHAT WAS HIT?
      JHIT=LXOR(KHIT,LHIT)
      DO 90 IS=1,NSPEC
C     RECORD HIT TIME FOR EACH OBJECT HIT.
      IF (LAND(JHIT,BITP(IS)).EQ.0) GO TO 90
      HTIME(IS,IB)=IT
```

```
C     RECORD MONTH TARGET IS FIRST HIT.
      HMONTH(IS,IB)=IMO
   90 CONTINUE
      KHIT=LHIT
  100 CONTINUE
      IF (OBJ(IXC,IYC).LT.0) GO TO 150
  110 CONTINUE
  120 CONTINUE
      XBL=XB
      YBL=YB
      LY=IY
  130 CONTINUE
C
C
C     TIME LIMIT PASSED -- SPILL ASSUMED DECAYED.
C     HIT TIME EQUAL TO -1 INDICATES SPILL DECAYED.
      HTIME(32,IB)=-1
      IDK=IDK+1
      GO TO 160
C
C
C     SPILL WENT OFF THE MAP.
  140 CONTINUE
C     RECORD THE DIRECTION.
      IDX=0
      IF (IX.GT.KXMAX) IDX=1
      IF (IX.GT.KXMAX) LX=KXMAX
      IF (IX.LT.KXMIN) IDX=IDX+2
      IF (IX.LT.KXMIN) LX=KXMIN
      IF (IY.GT.KYMAX) IDX=IDX+4
      IF (IY.GT.KYMAX) LY=KYMAX
      IF (IY.LT.KYMIN) IDX=IDX+8
      IF (IY.LT.KYMIN) LY=KYMIN
      IDX=OFFIDX(IDX)
      OFFMAP(IDX)=OFFMAP(IDX)+1
      NOFF=NOFF+1
      GO TO 160
C
C
C     THE SPILL HAS HIT LAND.
  150 CONTINUE
      HTIME(32,IB)=IT
      MEANTL=MEANTL+IT
      MAXTL=MAX0(MAXTL,IT)
      MINTL=MIN0(MINTL,IT)
      IDX=(IT*TP/24.0+3.5)/3
      TOLAND(IDX)=TOLAND(IDX)+1
      LANDED=LANDED+1
C     CHECK FOR LAND SEGMENTS.
```

```
      CALL NEWSEG(IXB,IYB)
C     DETERMINE THE LAND SEGMENT CODE.
      ISEG=LSEG(IXC,IYC)
      SAVEXY(3,IB)=ISEG
C
C
C     ACCUMULATE STATISTICS.
  160 CONTINUE
C     SAVE THE END POINT OF EACH SPILL.
      SAVEXY(1,IB)=LX
      SAVEXY(2,IB)=LY
      DO 170 IS=1,NSPEC
      IF (LAND(LHIT,BITP(IS)).NE.0) CHIT(IS)=CHIT(IS)+1
  170 CONTINUE
C
C
C     DONE THIS SPILL.
  180 CONTINUE
C
C
C
C     DONE THIS SEASON -- REPORT THE STATISTICS.
      IF (LANDED.NE.0) MEANTL=MEANTL*TP/(24.0*LANDED)+0.5
      MAX TL=MAX TL*TP/24.0+0.5
      MIN TL=MIN TL*TP/24.0+0.5
      WRITE (PRINT,6005) SEANAM(ISEA),LANDED,IDK,NOFF,MINTL,MEANTL,
     1   MAXTL,(CHIT(IS),IS=1,NSPEC)
 6005 FORMAT (1X,A6,6I4,31I3)
      IF (NOFF.NE.0) WRITE (PRINT,6006) OMN,OFFMAP
 6006 FORMAT (110X,7A3,A2/109X,8I3)
C
C     SAVE STATISTICS.
C
C     THE FOLLOWING IS RECORDED FOR EACH SEASON -
C        1. THE NUMBER OF HITS ON EACH OBJECT.
C        2. THE NUMBER OF SPILLS WHICH WENT OFF OF THE MAP.
C        3. THE NUMBER OF SPILLS WHICH HIT LAND.
C        4. THE NUMBER OF SPILLS WHICH DECAYED.
C        5. THE MINIMUM, MAXIMUM, AND AVERAGE TIMES TO LAND.
C        6. THE NUMBER OF SPILLS WHICH LANDED WITHIN EACH 3 DAYS.
C     FOR EACH SPILL, THE FOLLOWING STATISTICS ARE SAVED -
C        1. THE TIME AT WHICH THE SPILL FIRST HIT EACH OBJECT.
C        2. THE FINAL LOCATION OF THE SPILL.
C        3. THE LAND SEGMENT CODE, IF ANY, HIT BY THE SPILL.
C        4. THE MONTH (IF ANY) IN WHICH EACH TARGET WAS FIRST HIT.
      WRITE (UOUT) CHIT,OFFMAP,LANDED,IDK,MINTL,MEANTL,MAXTL,TOLAND,
     1   HTIME,SAVEXY,HMONTH
C
C
```

```
C     END SEASONS.
  190 CONTINUE
      GO TO 30
C
C     NORMAL END-OF-JOB.
  999 CONTINUE
C     REPORT ON PROGRAM PERFORMANCE.
      WRITE (6,6007) KSM,KNCB,KNCBR,KNOB,KNOBR,KNSB,KNSBR
 6007 FORMAT (///' NORMAL END-OF-JOB.'/
     1   ' SPILL PERFORMANCE STATISTICS:'/
     2   1X,I8,' SPILL MOVEMENTS.'/
     3   1X,I8,' CALLS FOR A NEW CURRENT BLOCK, WITH',
     4   I8,' DISK ACCESSES.'/
     5   1X,I8,' CALLS FOR A NEW TARGET BLOCK, WITH ',
     6   I8,' DISK ACCESSES.'/
     7   1X,I8,' CALLS FOR A NEW LAND SEGMENT BLOCK, WITH ',
     8   I8,' DISK ACCESSES.')
      STOP
C
C
C     ERROR MESSAGES.
 3010 CONTINUE
      WRITE (PRINT,3011) ISTA,NOWST(IWST)
 3011 FORMAT (///' ***** WIND STATION INPUT MISMATCH.'/
     1   ' WANTED',I8,'   GOT',I8/' EXECUTION TERMINATED.')
      STOP 16
 3020 CONTINUE
      WRITE (PRINT,3621) SALE,SALEG
 3621 FORMAT (///' ***ERROR. CARD ASKS FOR SALE ',2A4,' BUT THE ',
     1   'GENERAL FILE IS FOR SALE ',2A4)
      STOP 16
 3030 CONTINUE
      WRITE (PRINT,3631) IERR
 3631 FORMAT (///' ***ERROR. MERCATOR CONVERSION SUBROUTINE.')
      STOP 16
 3040 CONTINUE
      WRITE(PRINT,3721)RUNAME,TNAMES
 3721 FORMAT(///' ***ERROR. CARD ASKS FOR LAUNCH POINT ',10A4/
     1 ' BUT THE DISK FILE IS FOR LAUNCH POINT ',10A4)
      STOP 16
      END
C
C
C
C
      SUBROUTINE NEWBLK(IXB,IYB,LXB,LYB)
C
C     SEE THAT THE PROPER CURRENT AND OBJECT BLOCKS ARE LOADED.
C
```

```
C     THIS SUBROUTINE IS DESIGNED TO REDUCE THE NUMBER OF DISK
C        ACCESSES NECESSARY TO LOAD CURRENT AND OBJECT MATRICES.
C        UP TO 32 CURRENT MATRICES AND 16 OBJECT MATRICES CAN BE
C        STORED IN CORE. THOSE SELECTED ARE THE ONES WHICH WERE
C        CALLED MOST RECENTLY.
C
C     FILES ARE: 40-CURRENT,50-LAND & OBJECTS.
      DEFINE FILE 40(256,450,U,IP40)
      DEFINE FILE 50(256,900,U,IP50)
C
      INTEGER PRINT/6/,UCUR/40/,UOBJ/50/
      COMMON /BLOCK/ OBJ,CUR,NSPBLK
      INTEGER NSPBLK(16,16)
      INTEGER OBJ(30,30)
      INTEGER HOBJ(30,30,16)
      INTEGER*2 CUR(30,30)
      INTEGER*2 HCUR(30,30,32)
      INTEGER IAMC(32)/32*0/,IAMO(16)/16*0/
      INTEGER CTIME(32)/32*0/,OTIME(16)/16*0/
      INTEGER CIN(256)/256*0/,OIN(256)/256*0/
      INTEGER ITIME/0/
      INTEGER NB/16/,NC2/900/
      COMMON /COUNT/ KNCB,KNCBR,KNOB,KNOBR,KNSB,KNSBR
C
C     CHECK FOR LOCATION WITHIN RANGE
      IF (IXB.LT.1.OR.IXB.GT.NB.OR.IYB.LT.1.OR.IYB.GT.NB) GO TO 3010
      IP=(IYB-1)*NB+IXB
C     TIME COUNTER TO RECORD WHEN MATRICES ARE LOADED INTO CORE.
      ITIME=ITIME+1
      LXB=IXB
      LYB=IYB
C     COUNT THE CALLS FOR A NEW CURRENT BLOCK.
      KNCB=KNCB+1
C
C     IS NEEDED CURRENT BLOCK IN CORE?
      IF (CIN(IP).NE.0) GO TO 20
C
C     NO. LOAD INTO OLDEST LOCATION.
      IOLD=1
      DO 10 I=2,32
      IF (CTIME(I).LT.CTIME(IOLD)) IOLD=I
   10 CONTINUE
C     COUNT THE DISK ACCESSES FOR CURRENT BLOCKS.
      KNCBR=KNCBR+1
      READ (UCUR'IP) CUR
      CALL MOVE(CUR,HCUR(1,1,IOLD),NC2,2)
      CIN(IAMC(IOLD))=0
      CIN(IP)=IOLD
C     RECORD THE TIME THE MATRIX WAS STORED.
```

```
      CTIME(IOLD)=ITIME
      IAMC(IOLD)=IP
      GO TO 30
C
C     YES. MOVE TO CUR.
   20 CONTINUE
      CALL MOVE(HCUR(1,1,CIN(IP)),CUR,NC2,2)
C     RECORD THE TIME THE MATRIX WAS STORED.
      CTIME(CIN(IP))=ITIME
   30 CONTINUE
C
C     IS AN OBJECT BLOCK NEEDED?
C
C     NSPBLK IS A MATRIX WHICH IDENTIFIES THOSE OBJECT BLOCKS
C        WHICH CONTAIN INFORMATION, AND THOSE WHICH CONTAIN ALL ZEROS.
C     VALUES FOR NSPBLK ARE:
C         0 FOR A ZERO BLOCK.
C         NON-ZERO FOR A BLOCK ON THE FILE (HOLDS LOGICAL PATTERN).
      IF (NSPBLK(IXB,IYB).NE.0) GO TO 40
C
C     ZERO OBJECT BLOCK.
      CALL ZERO4(OBJ,NC2)
      RETURN
C
C     NON-ZERO BLOCK.
C     IS IT IN CORE?
   40 CONTINUE
C     COUNT THE CALLS FOR A NEW TARGET BLOCK.
      KNOB=KNOB+1
      IF (OIN(IP).NE.0) GO TO 60
C
C     NO. LOAD INTO OLDEST LOCATION.
      IOLD=1
      DO 50 I=2,16
      IF (OTIME(I).LT.OTIME(IOLD)) IOLD=I
   50 CONTINUE
C     COUNT THE DISK ACCESSES FOR NEW TARGET BLOCKS.
      KNOBR=KNOBR+1
      READ (UOBJ'IP) OBJ
      CALL MOVE(OBJ,HOBJ(1,1,IOLD),NC2,4)
      OIN(IAMO(IOLD))=0
      OIN(IP)=IOLD
      OTIME(IOLD)=ITIME
      IAMO(IOLD)=IP
      RETURN
C
C     YES. MOVE TO OBJ.
   60 CONTINUE
      CALL MOVE(HOBJ(1,1,OIN(IP)),OBJ,NC2,4)
```

```
C     RECORD THE TIME THE MATRIX WAS STORED.
      OTIME(OIN(IP))=ITIME
      RETURN
C
C     ERROR ROUTINE.
 3010 WRITE (PRINT,3011) IXB,IYB
 3011 FORMAT (/' ***** SUBROUTINE NEWBLK CALLED WITH IXB,IYB=',2I4/
     1   ' EXECUTION TERMINATED.')
      STOP 16
      END
C
C
      SUBROUTINE NEWSEG(IXB,IYB)
C
C     SEE THAT THE PROPER LAND SEGMENT BLOCK IS LOADED.
C
C     THIS SUBROUTINE IS DESIGNED TO REDUCE THE NUMBER OF DISK
C        ACCESSES NECESSARY TO LOAD LAND SEGMENT MATRICES.
C        UP TO 16 LAND SEGMENT BLOCKS CAN BE STORED IN CORE.
C        THOSE SELECTED ARE THE ONES WHICH WERE CALLED MOST RECENTLY.
      DEFINE FILE 60(256,450,U,IP60)
      INTEGER PRINT/6/,USEG/60/
      COMMON /BLOCK/ OBJ,ICUR,NSPBLK,LSGBLK,LSEG
      INTEGER NSPBLK(16,16)
      INTEGER OBJ(30,30)
      INTEGER*2 ICUR(30,30),LSEG(30,30)
      LOGICAL*1 LSGBLK(16,16)
      INTEGER*2 HSEG(30,30,16)
      INTEGER IAMS(16)/16*0/,STIME(16)/16*0/,SIN(255)/255*0/
      INTEGER ITIME/0/,LXB/0/,LYB/0/
      INTEGER NB/16/,NC2/900/
      COMMON /COUNT/ KNCB,KNCBR,KNOB,KNOBR,KNSB,KNSBR
C
C     DO NOTHING IF THE CURRENTLY ACTIVE BLOCK IS REQUESTED.
      IF (IXB.EQ.LXB.AND.IYB.EQ.LYB) RETURN
C     COUNT THE NUMBER OF CALLS FOR A NEW LAND SEGMENT BLOCK.
      KNSB=KNSB+1
C
C     CHECK FOR LOCATION WITHIN RANGE.
      IF (IXB.LT.1.OR.IXB.GT.NB.OR.IYB.LT.1.OR.IYB.GT.NB) GO TO 3010
      IP=(IYB-1)*NB+IXB
      LXB=IXB
      LYB=IYB
C     TIME COUNTER TO RECORD WHEN MATRICES ARE LOADED INTO CORE.
      ITIME=ITIME+1
C
C     CHECK FOR AN EMPTY BLOCK.
C     VALUES FOR LSGBLK ARE:
C        .FALSE. FOR AN EMPTY BLOCK.
```

```
C        .TRUE. FOR A VALID BLOCK ON THE FILE.
      IF (LSGBLK(IXB,IYB)) GO TO 10
C     ZERO THE SEGMENT BLOCK.
      CALL ZERO2(LSEG,NC2)
      RETURN
C
C     NON-ZERO BLOCK REQUESTED.
C     IS IT IN CORE?
   10 CONTINUE
      IF (SIN(IP).NE.0) GO TO 30
C
C     NO. LOAD IT INTO THE OLDEST LOCATION.
      IOLD=1
      DO 20 I=2,16
      IF (STIME(I).LT.STIME(IOLD)) IOLD=I
   20 CONTINUE
C     COUNT THE NUMBER OF DISK ACCESSES FOR NEW LAND SEGMENT BLOCKS.
      KNSBR=KNSBR+1
      READ (USEG'IP) LSEG
      CALL MOVE(LSEG,HSEG(1,1,IOLD),NC2,2)
      SIN(IAMS(IOLD))=0
      SIN(IP)=IOLD
      STIME(IOLD)=ITIME
      IAMS(IOLD)=IP
      RETURN
C
C     YES. MOVE IT TO THE ACTIVE AREA.
   30 CONTINUE
      CALL MOVE(HSEG(1,1,SIN(IP)),LSEG,NC2,2)
      STIME(SIN(IP))=ITIME
      RETURN
C     ERROR ROUTINE.
 3010 WRITE (PRINT,3011) IXB,IYB
 3011 FORMAT (/' ***** SUBROUTINE NEWSEG HAS BEEN CALLED WITH IXB,IYB=',
     1   2I4/' EXECUTION TERMINATED.')
      STOP 16
      END
C
C
      FUNCTION IPICK(C,IW)
C
C     IPICK SELECTS THE NEXT STATE IN THE MARKOV PROCESS.
C     THE WINDTRAN MATRIX IS PROVIDED IN SORTED FORM, SO THAT THE
C        MOST FREQUENT STATES ARE CHECKED FIRST.
C
      COMMON /SEED/ IX,IUSED
      INTEGER*2 C(1),IW(1),P
      REAL E(100)
      IF (IUSED.EQ.100) CALL RANDUV(IX,E,100)
      IF (IUSED.EQ.100) IUSED=0
      IUSED=IUSED+1
      P=E(IUSED)*10000.0
      DO 10 I=1,40
      IF (C(I).GT.P) GO TO 20
   10 CONTINUE
      I=41
   20 CONTINUE
      IPICK=IW(I)
      RETURN
      END
```

Procedure: HITPROB

Executes program: HITPROB

Purpose: To calculate the conditional probabilities of oilspill contact to targets.

Reads files: OSRA.&SALE.P1
OSRA.&SALE.P2
OSRA.&SALE.P3
OSRA.&SALE.P4
OSRA.&SALE.P5

Writes files: OSRA.&SALE.CTGT

Text page reference: 19

Procedure listing:

```
//HITPROB PROC D=CCDXXX
//*
//* PROCEDURE HITPROB
//* COMPUTES MATRIX CTGT
//* REQUIRED: &SALE
//*
//C EXEC PGM=IEFBR14
//CLEAR DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),
// DSN=OSRA.&SALE.CTGT
//CLEAR2 DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),
// DSN=OSRA.CARDCOPY.&SALE.CTGT
//G EXEC PGM=HITPROB,REGION=200K
//STEPLIB DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//* PUNCH OUTPUT IS SAVED ON DISK IN CARD IMAGE FORM, FOR RECOVERY
//* BY THE PROCEDURE COPYDC.
//FT07F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),SPACE=(TRK,(1,1),RLSE),
// DSN=OSRA.CARDCOPY.&SALE.CTGT
//FT10F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=6440,BLKSIZE=6444),SPACE=(TRK,(1,1),RLSE),
// DSN=OSRA.&SALE.CTGT
//FT51F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P1
//FT52F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P2
//FT53F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P3
//FT54F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P4
//FT55F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P5
```

Program listing:

```
C*******************************DOCUMENTATION*******************************
C
C   PROGRAM NAME: HITPROB (HP)
C
C   OPERATIONS PERFORMED: CALCULATES THE CONDITIONAL PROBABILITY OF A HIT ON
C      EACH OBJECT BY A SPILL FROM EACH LAUNCH POINT.
C
C   INPUT:
C      CARD: EXECUTED FROM THE LIBRARY PROCEDURE HITPROB. CERTAIN CONTROL
C         INFORMATION MUST BE SPECIFIED ON DATA CARDS.
C      DISK: TRAJECTORY OUTPUT FROM PROGRAM SPILL.
C
C   OUTPUT:
C      PRINT: CONDITIONAL PROBABILITY OF A HIT ON EACH OBJECT (WHEN VULNERABLE)
C         FROM EACH LAUNCH POINT, FOR PERIODS OF 3, 10, AND 30 DAYS.
C      PUNCH: THE PRINTED OUTPUT IS ALSO PUNCHED ON CARDS, FOR PRODUCING
C         REPORT TABLES ON THE MULTICS SYSTEM.
C      DISK: THE PRINTED OUTPUT IS ALSO STORED ON A DISK, AS
C         "OSRA.&SALE.TGT".
C
C   USER INSTRUCTIONS:
C      //HP EXEC HITPROB,SALE=XXXXXXXX
C      //G.SYSIN DD *
C      DATA CARD NUMBER 1
C         COL. 1-3 NUMBER OF TRAJECTORY RUN SETS. (I3)
C         COL. 4-6 NUMBER OF LAUNCH POINTS. (I3)
C
C   OPTIONS AND MODIFICATIONS:
C
C      1. CONCATENATING TRAJECTORY OUTPUT SETS.
C         SEE THE DOCUMENTATION FOR PROGRAM SPILL FOR THE CIRCUMSTANCES
C         UNDER WHICH TRAJECTORY OUTPUT SETS MAY NEED TO BE CONCATENATED.
C         THESE FILES CONTAIN NO HEADER OR TRAILER RECORDS, AND CAN EASILY BE
C         JOINED TO THOSE NAMED IN THE LIBRARY PROCEDURE. SEE THE LISTING
C         OF PROCEDURE HITPROB FOR THE CORRECT ORDER AND FILE NAMES.
C
C      2. NUMBER OF TRAJECTORY OUTPUT SETS.
C         HITPROB USUALLY RUNS WITH FIVE (5) TRAJECTORY OUTPUT SETS. IF A
C         DIFFERENT NUMBER IS USED, PORTIONS OF PROCEDURE HITPROB MUST BE
C         OVERRIDDEN. SEE THE PROCEDURE LISTING.
C
C      3. DUMMY OUTPUT.
C         IN TESTING, IT MAY BE DESIRABLE TO SUPPRESS DISK AND/OR PUNCHED
C         OUTPUT. THE DISK OUTPUT FILE IS G.FT10F001 AND THE PUNCH OUTPUT
C         IS G.FT07F001.
C
C      4. CLEARING PREVIOUS OUTPUT FILES.
```

```
C         PROCEDURE HITPROB AUTOMATICALLY CLEARS PREVIOUS OUTPUT FILES
C         NAMED "OSRA.&SALE.TGT".
C
C   TIMING AND STORAGE: PROCEDURE HITPROB PROVIDES SUFFICIENT TIME AND
C      STORAGE FOR NORMAL OPERATIONS:
C         TIME=5
C         REGION=140K
C
C*********************************************************************************
C
C
C     PROGRAM HITPROB
C     MODIF ED BY K. J. LANFEAR, USGS, 7 MAR 78.
C
C     CALCULATES THE CONDITIONAL PROBABILITY OF A HIT OCCURRING ON EACH
C        OBJECT, FOR PERIODS OF 3, 10, AND 30 DAYS.
C        PRODUCES A PRINTOUT, A DISK RECORD, AND PUNCHED OUTPUT.
C
C
      INTEGER*2 HTIME(32,100)
      REAL RUNAME(10),OBJNAM(15,31),VERSIN(4),WHEN(7)
      INTEGER OFFMAP(8),CHIT(31),TOLAND(31)
      INTEGER IP(32)
      INTEGER KDA(5)/3,10,30,60,90/
      INTEGER SNTRY/0/
      REAL P(32)
      REAL SALE(2)
      LOGICAL FIRST/.TRUE./,FIRSTP/.TRUE./
C
C     READ NUMBER OF TRAJECTORY SETS AND NUMBER OF LAUNCH POINTS.
      READ (5,1001) NR,NSITE
 1001 FORMAT (2I3)
C
C
C     DO FOR 3, 10, AND 30 DAYS.
      DO 120 JDA=1,3
      IDA=KDA(JDA)
C
C     REWIND THE FILES.
      DO 10 IR=1,NR
      IU=50+IR
      REWIND IU
   10 CONTINUE
C     'FIRST IS USED TO CONTROL PRINTING HEADINGS.
      FIRST=.TRUE.
C
C
C     DO FOR EACH LAUNCH POINT.
      DO 110 ISITE=1,NSITE
      CALL ZERO(P,32)
C
C
C     DO FOR EACH TRAJECTORY RUN SET.
      DO 70 IR=1,NR
      IU=50+IR
C     READ THE HEADER RECORD DESCRIBING THE RUN SET.
      READ (IU) NTRY,IX0,IY0,VERSIN,WHEN,WDRIFT,OBJNAM,NSPEC,IXS,
     1   RUNAME,NTP,NSEGA,NSEGB,SALE
C     PRINT A HEADING THE FIRST TIME THROUGH.
      IF (FIRST) WRITE (6,6001) SALE,IDA,VERSIN,
     1   (WHEN(I),I=1,3),(IS,IS=1,NSPEC)
 6001 FORMAT ('1 ',2A4,3X,I3,' DAY PASSAGE PROBABILITIES'/
     1   ' USING VERSION ',A3,' OF ',A3,A4,A2, ' ON ',I2,1X,A3,I3//
     3   '0*** NOTICE - THIS PRINTOUT IS A USGS WORKING DRAFT.'/
     4   '      IT IS SUBJECT TO CHANGE.'//
     2   ' LOC LAND',31I4)
      FIRST=.FALSE.
C     PUNCH HEADER TO IDENTIFY THE CARD OUTPUT.
      IF (FIRSTP) WRITE (7,7002) SALE,VERSIN,(WHEN(I),I=1,3)
 7002 FORMAT (2A4,3X,'HITPROB, VERSION ',A3,' OF ',A3,A4,A2,' ON ',I2,
     1   1X,A3,I3)
      FIRSTP=.FALSE.
C     SUM THE NUMBER OF TRAJECTORIES.
      SNTRY=SNTRY+NTRY
      LTP=NTP*IDA
C
C     READ TRAJECTORY STATISTICS AND SUM FOR EACH SEASON.
      DO 60 ISEA=1,4
      READ (IU) CHIT,OFFMAP,LANDED,IDK,MINTL,MEANTL,MAXTL,
     1   TOLAND,HTIME
C
C     SUM THE HITS ON EACH OBJECT.
      DO 50 IS=1,32
      IF (IS.GT.NSPEC.AND.IS.NE.32) GO TO 40
      N=0
      DO 30 ITRY=1,NTRY
      IF (HTIME(IS,ITRY).GT.0.AND.HTIME(IS,ITRY).LE.LTP) N=N+1
   30 CONTINUE
      P(IS)=P(IS)+N
   40 CONTINUE
   50 CONTINUE
C
   60 CONTINUE
C
   70 CONTINUE
C
C     CALCULATE THE CONDITIONAL PROBABILITIES.
      DA=SNTRY*4
      DB=DA/100.0
      SNTRY=0
      DO 80 IS=1,32
      IP(IS)=P(IS)/DB+0.5
      P(IS)=P(IS)/DA
   80 CONTINUE
C
C
C     PRINT THE RESULTS.
      WRITE (6,6002) RUNAME(1),IP(32),(IP(I),I=1,NSPEC)
 6002 FORMAT (1X,A3,I5,31I4)
C     PUNCH THE RESULTS.
C     THE PUNCHED OUTPUT CAN BE USED FOR PRINTING ON OTHER MACHINES.
      WRITE (7,7001) RUNAME(1),IDA,NSPEC,IP
 7001 FORMAT (A4,2I3,32I2)
C     WRITE RESULTS ON A DISK.
      WRITE (10) RUNAME(1),IDA,NSPEC,P
C
  110 CONTINUE
C
  120 CONTINUE
      STOP
      END
```

Procedure: LANDSEG
Executes program: LANDSEG
Purpose: To calculate the conditional probabilities
of oilspill contact to land segments.
Reads files: OSRA.&SALE.P1
OSRA.&SALE.P2
OSRA.&SALE.P3
OSRA.&SALE.P4
OSRA.&SALE.P5
Writes files: OSRA.&SALE.CSEG.SET2
Text page reference: 19

Procedure listing:

```
//LANDSEG2 PROC D=CCDXXX
//*
//* PROCEDURE LANDSEG 2
//* COMPUTES MATRIX CSEG FOR LAND SEGMENT SET NUMBER 2.
//* REQUIRED: &SALE
//*
//C EXEC PGM=IEFBR14
//CLEAR DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),
// DSN=OSRA.&SALE.CSEG.SET2
//CLEAR2 DD UNIT=3330,VOL=SER=&D,DISP=(OLD,DELETE),
// DSN=OSRA.CARDCOPY.&SALE.CSEG.SET2
//G EXEC PGM=LANDSEG2,REGION=280K
//STEPLIB DD UNIT=3330,VOL=SER=&D,DISP=SHR,DSN=OSRA.PGMLIB
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//* PUNCH OUTPUT IS SAVED ON DISK IN CARD IMAGE FORM.
//FT07F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),SPACE=(TRK,(10,5),RLSE),
// DSN=OSRA.CARDCOPY.&SALE.CSEG.SET2
//FT10F001 DD UNIT=3330,VOL=SER=&D,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=6440,BLKSIZE=6444),SPACE=(TRK,(10,5),RLSE),
// DSN=OSRA.&SALE.CSEG.SET2
//FT51F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P1
//FT52F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P2
//FT53F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P3
//FT54F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P4
//FT55F001 DD UNIT=3330,VOL=SER=&D,DISP=SHR,
// DSN=OSRA.&SALE.P5
```

Program listing:

```
C********************************DOCUMENTATION********************************
C
C     PROGRAM NAME: LANDSEG2
C   OPERATIONS PERFORMED: CALCULATES THE CONDITIONAL PROBABILITY OF A HIT
C     ON EACH LAND SEGMENT IN SET NUMBER 2, BY A SPILL FROM EACH OF THE
C      LAUNCH POINTS.
C
C   INPUT:
C     CARD: EXECUTED FROM THE LIBRARY PROCEDURE LANDSEG2. CERTAIN CONTROL
C         INFORMATION MUST BE SPECIFIED ON DATA CARDS.
C      DISK: TRAJECTORY OUTPUT FROM PROGRAM SPILL.
C
C   OUTPUT
C      PRINT: CONDITIONAL PROBABILITY OF A HIT ON EACH LAND SEGMENT IN SET
C        NUMBER 2, BY A SPILL FROM EACH OF THE LAUNCH POINTS, FOR TRAVEL
C        TIMES OF 3, 10, AND 30 DAYS.
C      PUNCH: THE PRINTED OUTPUT IS ALSO PUNCHED ON CARDS, FOR PRODUCING
C         REPORT TABLES ON THE MULTICS SYSTEM.
C        THE PUNCH OUTPUT IS STORED ON DISK IN CARD IMAGE FORM, FOR
C        PUNCHING AT A LATER TIME, WITH PROCEDURE COPYDC.
C      DISK: THE PRINTED OUTPUT IS ALSO STORED ON A DISK, AS
C        'OSRA.&SALE.CSEG.SET2'
C
C   USER INSTRUCTIONS:
C     //L2 EXEC LANDSEG2,SALE=XXXXXXXX
C      //G.SYSIN DD *
C      DATA CARD NUMBER 1
C         COL. 01-03 NUMBER OF TRAJECTORY RUN SETS. (I3)
C         COL. 04-06 NUMBER OF LAUNCH POINTS. (I3)
C      /*
C
C   OPTIONS AND MODIFICATIONS:
C
C      1. CONCATENATING TRAJECTORY OOUTPUT SETS.
C         SEE THE DOCUMENTATION FROM PROGRAM SPILL FOR THE CIRCUMSTANCES
C         UNDER WHICH TRAJECTORY OUTPUT SETS MAY NEED TO BE CONCATENATED.
C         THESE FILES CONTAIN NO HEADER OR TRAILER RECORDS, AND CAN EASILY BE
C         JOINED TO THOSE NAMED IN THE LIBRARY PROCEDURE. SEE THE LISTING
C         OF PROCEDURE LANDSEG1 FOR THE CORRECT ORDER AND FILE NAMES.
C
C      2. NUMBER OF TRAJECTORY OUTPUT SETS.
C        LANDSEG2 USUALLY RUNS WITH FIVE (5) TRAJECTORY OUTPUT SETS. IF A
C        DIFFERENT NUMBER IS USED, PORTIONS OF PROCEDURE LANDSEG2 MUST BE
C         OVERRIDDEN. SEE THE PROCEDURE LISTING.
C
C      3. DUMMY OUTPUT.
C         IN TESTING, IT MAY BE DESIRABLE TO SUPPRESS DISK AND/OR PUNCHED
C         OUTPUT. THE DISK OUTPUT FILE IS G.FT10F001 AND THE PUNCH OUTPUT
C         IS G.FT07F001.
```

```
C
C      4. CLEARING PREVIOUS OUTPUT FILES.
C        PROCEDURE LANDSEG2 AUTOMATICALLY CLEARS PREVIOUS OUTPUT FILES
C        NAMED "OSRA.&SALE.CSEG.SET2"
C
C   TIMING AND STORAGE: PROCEDURE LANDSEG2 PROVIDES SUFFICIENT TIME AND
C      STORAGE FFOR NORMAL OPERATIONS:
C         TIME=3
C         REGION=180K
C
C*******************************************************************************
C     PROGRAM LANDSEG2
C     MODIFIED BY K. J. LANFEAR, USGS, 24 APR 79.
C
C     CALCULATES THE CONDITIONAL PROBABILITY OF A HIT ON EACH LAND
C        SEGMENT, FOR THE FIRST SET OF LAND SEGMENTS.
C
C     LANDSEG2 ACCEPTS UP TO 100 LAND SEGMENTS, AND 100 LAUNCH POINTS.
C
C
C
C
      INTEGER*2 HTIME(32,100),SAVEXY(3,100)
      REAL RUNAME(10),OBJNAM(15,31),VERSIN(4),WHEN(7)
      INTEGER OFFMAP(8),CHIT(31),TOLAND(31)
      INTEGER IP(100,100)
      INTEGER KDA(5)/3,10,30,60,90/
      REAL RN(100),P(100)
      REAL SALE(2)
      LOGICAL FIRST/.TRUE./
      INTEGER SNTRY/0/
C
C     READ THE NUMBER OF RUN SETS, AND THE NUMBER OF LAUNCH POINTS.
      READ (5,5001) NR,NSITE
 5001 FORMAT (2I3)
C
C
C     DO FOR 3, 10, AND 30 DAYS.
      DO 120 JDA=1,3
      IDA=KDA(JDA)
C
C     REWIND THE FILES.
      DO 10 IR=1,NR
      IU=50+IR
      REWIND IU
   10 CONTINUE
      CALL ZERO(IP,10000)
C
C
```

```
C     DO FOR EACH LAUNCH POINT
      DO 110 ISITE=1,NSITE
C     INITIALIZE THE PROBABILITIES FOR THIS LAUNCH POINT.
      CALL ZERO(P,100)
C
C
C     DO FOR EACH SPILL DATA SET.
      DO 70 IR=1,NR
      IU=50+IR
C     READ THE HEADER RECORD.
      READ (IU) NTRY,IXO,IYO,VERSIN,WHEN,WDRIFT,OBJNAM,NSPEC,IXS,
     1   RUNAME,NTP,NSEGA,NSEGB,SALE
C     SUM THE NUMBER OF TRAJECTORIES.
      SNTRY=SNTRY+NTRY
      LTP=NTP*IDA
C
C
C     DO FOR EACH SEASON
      DO 60 ISEA=1,4
      READ (IU) CHIT,OFFMAP,LANDED,IDK,MINTL,MEANTL,MAXTL,
     1   TOLAND,HTIME,SAVEXY
C
C     TABULATE HITS FOR EACH TRAJECTORY.
      DO 30 ITRY=1,NTRY
C     LAND SEGMENTS ARE CODED IN TWO PARTS. THE FIRST TWO DIGITS
C        ARE USED FOR THE FIRST SET OF LAND SEGMENTS.
C        THE LAST TWO DIGITS ARE USED FOR THE SECOND SET.
      ISEG=0
      IF (HTIME(32,ITRY).GT.0.AND.HTIME(32,ITRY).LE.LTP) ISEG=SAVEXY(3,
     1   ITRY)-(SAVEXY(3,ITRY)/100)*100
C     CHECK FOR ALLOWABLE LAND SEGMENT CODE.
      IF (ISEG.LT.0.OR.ISEG.GT.NSEGB) WRITE (6,6003) IR,ISITE,ISEA,
     1   ITRY,ISEG
 6003 FORMAT (' SEGMENT NUMBER ERROR',5I5)
      IF (ISEG.GT.0) P(ISEG)=P(ISEG)+1.0
   30 CONTINUE
C
   60 CONTINUE
C
   70 CONTINUE
C
C     ONLY THE FIRST FOUR CHARACTERS OF THE LAUNCH POINT NAME WILL BE
C        USED FOR PRINTOUT, BECAUSE OF SPACE LIMITATIONS.
      RN(ISITE)=RUNAME(1)
C
C     CALCULATE THE PROBABILITIES.
      DA=SNTRY*4
      DB=DA/100.0
      SNTRY=0
      SN=0.0
      DO 80 ISEG=1,NSEGB
      SN=SN+P(ISEG)
      IP(ISITE,ISEG)=P(ISEG)/DB+0.5
      P(ISEG)=P(ISEG)/DA
   80 CONTINUE
C     OVERALL TOTAL FOR THIS LAUNCH POINT
      IP(ISITE,100)=SN/DB+0.5
      P(100)=SN/DA
C     WRITE THE CONDITIONAL PROBABILITIES ON A DISK.
      WRITE(10)RUNAME(1),IDA,NSEGB,P,SALE
C
  110 CONTINUE
C
C
C     PRINT THE RESULTS.
C     BECAUSE OF SPACE LIMITATIONS, IT MAY BE NECESSARY TO BREAK THE
C        PRINTOUT INTO AS MANY AS FOUR SECTIONS.
C
C     PRINT A HEADING THE FIRST TIME THROUGH.
      IF (FIRST) WRITE (6,6005) SALE,NR,NSITE,VERSIN,
     1   (WHEN(I),I=1,3)
 6005 FORMAT ('1PROGRAM LANDSEG2, FOR SALE ',2A4/
     1   ' THE ',I4,' DATA SETS, EACH WITH ',I4,' LAUNCH POINTS,',
     2   ' WERE CREATED USING SPILL VERSION ',A3,' OF ',A3,A4,A2,
     3   ' ON ',I2,A3,I3)
C
C     PUNCH A HEADER THE FIRST TIME THROUGH,
C        TO IDENTIFY THE PUNCHED OUTPUT.
      IF (FIRST) WRITE (7,7002) SALE,VERSIN,(WHEN(I),I=1,3)
 7002 FORMAT (2A4,3X,'LANDSEG1, VERSION ',A3,' OF ',A3,A4,A2,' ON ',I2,
     1   1X,A3,I3)
      FIRST=.FALSE.
C
C     PRINTOUT IN GROUPS OF 25 LAUNCH POINTS.
      IPRT=(NSITE-1)/25+1
      DO 41 J=1,IPRT
      IPRTA=25*(J-1)+1
      IPRTB=IPRTA+24
      IF (NSITE.LT.IPRTB) IPRTB=NSITE
      WRITE (6,6001) SALE,IDA,(RN(I),I=IPRTA,IPRTB)
 6001 FORMAT ('1',2A4,' LAND SEGMENT PROBABILITIES FOR',I3,' DAYS.'//
     2   ' *** NOTICE - THIS PRINTOUT IS A USGS WORKING DRAFT. ***'/
     3   ' *** IT IS SUBJECT TO CHANGE.                        ***'//
     1   ' SEG',2X,25A4)
      DO 40 ISEG=1,NSEGB
      WRITE (6,6002) ISEG,(IP(ISITE,ISEG),ISITE=IPRTA,IPRTB)
 6002 FORMAT (26I4)
   40 CONTINUE
```

```
      WRITE (6,6004) (IP(ISITE,100),ISITE=IPRTA,IPRTB)
 6004 FORMAT (' ALL',25I4)
   41 CONTINUE
C
C     PUNCH THE RESULTS, FOR PRINTOUT ON OTHER MACHINES.
      DO 42 ISEG=1,NSEGB
      WRITE (7,7001) IDA,ISEG,(IP(ISITE,ISEG),ISITE=1,NSITE)
 7001 FORMAT (40I2/4X,38I2/4X,24I2)
   42 CONTINUE
C
  120 CONTINUE
      STOP
      END
```