06
11-30-76

WATER RESOURCES RESEARCH PROGRAM

NEARSHORE CURRENTS AND WATER TEMPERATURES IN SOUTHWESTERN LAKE MICHIGAN

(June–December 1975)

by

K. D. Saunders and L. S. Van Loon

NOTICE

PORTIONS OF THIS REPORT ARE ILLEGIBLE. It
has been reproduced from the best available
copy to permit the broadest possible avail-
ability.

GREAT LAKES PROJECT

ENERGY AND ENVIRONMENTAL
SYSTEMS DIVISION

MASTER

DIVISION OF
ENVIRONMENTAL IMPACT STUDIES

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Prepared for the U. S. ENERGY RESEARCH
AND DEVELOPMENT ADMINISTRATION
under Contract W-31-109-Eng-38

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Energy Research and Development Administration, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona	Kansas State University	The Ohio State University
Carnegie-Mellon University	The University of Kansas	Ohio University
Case Western Reserve University	Loyola University	The Pennsylvania State University
The University of Chicago	Marquette University	Purdue University
University of Cincinnati	Michigan State University	Saint Louis University
Illinois Institute of Technology	The University of Michigan	Southern Illinois University
University of Illinois	University of Minnesota	The University of Texas at Austin
Indiana University	University of Missouri	Washington University
Iowa State University	Northwestern University	Wayne State University
The University of Iowa	University of Notre Dame	The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights. Mention of commercial products, their manufacturers, or their suppliers in this publication does not imply or connote approval or disapproval of the product by Argonne National Laboratory or the U. S. Energy Research and Development Administration.

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161
Price: Printed Copy \$8.00; Microfiche \$2.25

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

WATER RESOURCES RESEARCH PROGRAM

NEARSHORE CURRENTS AND WATER TEMPERATURES
IN SOUTHWESTERN LAKE MICHIGAN

(June-December 1975)

by

K. D. Saunders and L. S. Van Loon

Energy and Environmental Systems Division
Great Lakes Project

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

May 1976

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	7
I. INTRODUCTION	9
II. DATA ACQUISITION	11
A. Layout for Current-meter Moorings	11
B. Mooring Design and Reliability	11
C. Instrumentation	14
D. Data Reduction	14
III. RESULTS	18
A. Data Return	18
B. General Water Movements	18
C. General Temperature Trends	20
REFERENCES	21
APPENDIX I. Plots of Current and Temperature Data	I.1
APPENDIX II. Listings of Data-processing Programs	II.1

LIST OF FIGURES

<u>No.</u>		<u>Page</u>
1	Locations of Current-meter Moorings	10
2	Design of a Typical Current-meter Mooring	12
3	Mechanical Wear of a Spar-buoy Mooring Ring	15
4	Damage to the Current Meter at Station IV by an SMP Propeller	16
5	Data Return from the Current Meters	19

LIST OF TABLES

<u>No.</u>	<u>Page</u>
I. Current-meter Station Positions	13
II. Mini-ranger Ranges to the Current-meter Stations	13

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

NEARSHORE CURRENTS AND WATER
TEMPERATURES IN SOUTHWESTERN LAKE MICHIGAN
(JUNE - DECEMBER, 1975)

by

K. D. Saunders

and

L. S. Van Loon

ABSTRACT

Nearshore currents and water temperatures were measured almost continuously from June 23 through December 22, at five moorings in an array centered 4 km offshore of south Chicago. The mooring array was square, each side 1.6 km long. A current-meter mooring was placed at each corner, with one mooring in the center. One Bendix Q-15 current meter and one YSI temperature sensor were fixed to each mooring line. Each meter and associated temperature sensor was placed at middepth; the water depth averaged about 12 m.

The following types of graphs are presented for current and wind observations: (1) U, V flow components versus time, (2) specific kinetic energy versus time, (3) flow speeds and directions versus time, (4) composite velocity histograms and associated U, V-component histograms, and (5) progressive-vector diagrams. Also presented are listings of the component programs used to reduce the data.

Currents in the region were dominately shore-parallel. Water temperatures reflected several episodes of upwelling and downwelling. Detailed analyses of the data will be presented in subsequent reports.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

I. INTRODUCTION

Virtually nothing is known about the detailed fine structure of coastal currents off Indiana and Illinois in extreme southwestern Lake Michigan. The present study was designed to investigate the temporal and spatial coherence of currents off South Chicago, Illinois, and East Chicago, Indiana (Fig. 1). Specific quantities of nonnuclear, energy-related pollutants enter the coastal waters of this part of the lake from the Indiana Harbor Canal (Fig. 1); lesser amounts are contributed by intermittent discharges from the Calumet River, Illinois, and Burns Ditch, Indiana. The processes of transport and dispersion of these pollutants while in the near-shore zone, and the mechanics of their mixing with the deeper-lake waters, have yet to be documented.

Previous work (Birchfield and Davidson, 1967; Snow, 1974; and Monahan and Pilgrim, 1975) on coastal currents in this region dealt with very short time series of measurements or involved measurement techniques that were incapable of generating data for studies of the temporal and spatial variability of the water motions. Thus, the goals of this study were to determine the long-period motion of the nearshore water (for periods ranging from one week to several months) and to investigate the spatial and temporal variability of these motions. The space scale was chosen to be on the order of a few kilometers and the time scale from several minutes to several days. Water-temperature fluctuations were investigated also as a related, but lower-level effort.

This report describes the methods used in obtaining the data, delineates the data-reduction techniques, and presents the original data in reduced form. Detailed analyses of the current and temperature data will appear in subsequent reports.

Fig. 1. Locations of Current-meter Moorings.

II. DATA ACQUISITION

A. Layout for Current-meter Moorings

The square-shaped array of current-meter moorings shown in Fig. 1 is suggested by the need for data that permit evaluation of the spatial coherence of currents in two main directions. The inshore and offshore sides of the square are essentially shore-parallel; the other two sides are shore-perpendicular. Thus, both the "alongshore" and the "offshore" aspects of the horizontal coherence of the current field may be evaluated by the square array. A fifth mooring (V in Fig. 1), at the center of the square, assists in interpreting trends in the structure of currents in either of the aforementioned directions, as well as in directions along the diagonals of the square. The length chosen for the square mooring array was 1.6 km (1 statute mi) and the center of the array was positioned some 4 km (2-1/2 mi) offshore. Thus, the inshore side of the array was about 3.2 km (2 mi) offshore, and the offshore side was about 4.8 km (3 mi) out into the lake. The location of the array avoided most, but not all, of the main shipping lanes in this region of Lake Michigan.

Positions for the mooring stations are listed in Table I. During field deployment of the moorings, these positions (Table I) were found by using a Motorola Mini-ranger positioning system. The Mini-ranger transponders were located on the 68th Street Crib and the end of the Calumet Harbor Breakwater (Fig. 1). Range distances, from the transponders to each mooring station, are given in Table II.

B. Moorings Design and Reliability

The moorings used were standard "U" moorings (Fig. 2). A spar buoy was taut-moored with a (2.1-m) section of chain between the buoy and the nylon mooring line. The chain was attached to the buoy by a padlock to discourage vandalism. No vandalism was apparent upon recovery, but the hardened shackle of the lock had almost worn through the connecting ring on the buoy.

The current meter, with contained temperature sensor, was moored on a separate line about 20 m from the spar line. The current-meter impeller was positioned 4-5 m below the surface.

Fig. 2. Design of a Typical Current-meter Mooring.

TABLE I. Current-meter Station Identification

Station	Current-meter Recording Package	Latitude, N	Longitude, W
I	Q3	41° 45.73'	87° 29.26'
II	Q5	41° 46.34'	87° 30.08'
III	Q6	41° 46.96'	87° 29.26'
IV	Q7	41° 46.34'	87° 28.43'
V	Q8	41° 46.34'	87° 29.26'

TABLE II. Mini-ranger Ranges to the Current-meter Stations

Station	Range from 68th St. Crib (meters)	Range from Calumet Harbor Breakwater (meters)
I	4509	3998
II	2890	5162
III	3628	6322
IV	4962	5442
V	3954	5182

Two anchors were used on this mooring (Fig. 2) and were connected by a nylon ground line. A small surface float marked the position of the current meter.

The moorings proved to be reliable, the only real evidence of wear being the abrasion of the spar-buoy's ring (Fig. 3). Ship passage was a problem at the outermost station (St. IV). On September 8, at about 2340 C.D.T., the current meter and subsurface float were apparently struck by a ship's propeller which caused considerable damage (Fig. 4) to the meter. The outer casing of the subsurface float and the current-meter head were damaged, but the electronics package and the mooring lines were unharmed.

Except for the damage at Station IV, all the mooring equipment functioned successfully throughout June-November, 1975, and remained in good condition. No equipment was lost.

C. Instrumentation

The current meters used for this study all had Bendix Q-15 R current-meter heads. Data were recorded by ANL-developed electronic current-meter instrumentation packages, described in detail in Saunders et al. (1976). At the end of the data-acquisition period, the current meters were returned to Argonne and their direction sensors checked. One of the current-meter heads, serial number 173-2526 (St. V), showed evidence of direction error. This was easily corrected by repositioning a small wiper wire in the package. Another current-meter head, serial number 173-2523 (St. III), showed evidence of slight direction errors near north, but these were within the specifications of the manufacturer. Because of the damage caused by the ship, the current-meter head associated with Station IV could not be checked.

The instrumentation worked reliably over the entire period. The only loss of data other than at Station IV was due to the disengagement of the writing head of the magnetic-tape cassette in the electronics package at Station II during June 25 - July 24, 1975.

D. Data Reduction

The computer program for converting the raw data to standard format (described in Saunders et al. [1976]) was written in November, and all of the

Fig. 3. Mechanical Wear of a Spar-buoy Mooring Ring.

Fig. 4. Damage to the Current Meter at Station IV
by an SMP Propeller.

data were reduced by this technique. The program is listed in Appendix II. The program searches for bad records and, when these are encountered, ignores them. A record is considered bad if it fails to meet the following criteria:

1. The clock reading is within 2 hr of the expected reading.
2. The speed count is less than 0.5 ms^{-1} and greater than 0.0 ms^{-1} .
3. The speed count is within three standard deviations of the average speed.
4. The temperature is between 0.0 and 30.0°C .
5. Three of the four direction readings are within 90° of each other.
6. The direction reading is between 0 and 360° .

A bad record is replaced by linear interpolation between the nearest good records. The data are then plotted by the plotting program listed in Appendix II.

III. RESULTS

A. Data Return

The data return for June-November is shown schematically in Fig. 5. The first record from Q5 (St. II) was lost due to an improper seating of the writing head on the cassette tape recorder. The direction readings from Q8 (St. V) appear to be totally unreliable during the third acquisition period (record Q8753) and perhaps somewhat doubtful during the second acquisition period. Station IV was not deployed during the third acquisition period, after being damaged by a ship.

B. General Water Movements

Between June 23 and July 11, net water motion was toward the southeast, with few reversals (Figs. I.4, I.14, and I.24). Between July 11 and July 24, there was a small net flow toward the northwest (Figs. I.9, I.19, I.29, and I.39). From July 24 to August 13, the net flow was toward the northwest (Figs. I.44, I.59, I.79, and I.94). From August 13 to September 2, the flow was toward the northwest (Figs. I.49, I.64, I.74, I.84, and I.99). The net flow between September 2 and 17 was southeastward, with several major reversals (Figs. I.54 and I.89). Between September 17 and 24, the flow was toward the northwest, and between September 24 and October 7, the net flow was toward the southeast with two major reversals (Figs. I.107, I.109, I.122, I.124, I.137, I.139, I.147, and I.149). From October 7 to 27, the net flow was toward the southwest and consisted mainly of northwest and southeast flows which had a net drift toward the southwest (Figs. I.114, I.129, and I.144). From October 27 to November 16, the net flow was toward the west, but again, the major currents were northwest and southeast (Figs. I.119 and I.134).

The currents in this region were predominantly shore parallel -- either northwest or southeast flows. The variance in current speed generally exceeded the average current speed by an order of magnitude. There was general agreement in flow direction and reversals between all pairs of current meters, except for Q8 (St. V), which had a bad contact in the direction-sensing hardware. The current speeds between the various meters showed general agreement.

Fig. 5. Data Return from the Current Meters.

The Bendix Q15 current meters were removed on November 17 for servicing before the winter deployment. To maintain a reasonably continuous record, three Braincon histogram-recording current meters were deployed at Stations II, III, and V (current meter numbers B96, B97, and B98, respectively). These remained in until December 22.

The net motion of the water over this period was toward the southeast, but at least 11 major reversals occurred during this period. The currents were particularly strong, most observations ranging from 0.15 to 0.30 m/s, though some currents as strong as 1.0 m/s were observed.

There were several periods during June-December when the net flow before reversal traveled more than 10 km toward the northwest. This order of net transport could be sufficient to carry polluted water from the Indiana Harbor Canal to the 68th Street Crib. In fact, episodes of malodorous water in November and December coincided with large northerly movements of water.

C. General Temperature Trends

Using Station I as a base, the temperatures of the water showed evidence of several downwelling and upwelling periods. From June 23 to July 4, the water temperature in the vicinity of the current meter varied several degrees in the vicinity of 15°C (Fig. I.1). On July 4, the temperature rose to about 22°C and remained warm until July 19, when the temperature dropped below 15°C (Figs. I.1 and I.6). From July 25 to 28, water temperature oscillated between 15 and 20°C, dropping below 15°C on July 28 (Fig. I.41). It remained below 15°C until August 3. The temperature then oscillated in the vicinity of 20°C until August 5 and remained above 20°C until August 9. It then dropped to about 15°C until August 13. From August 13 to August 25, there was a slow warming near 20°C, followed by a weak cooling period covering August 25 - August 30 (Fig. I.46). The temperature then rose to about 20°C and remained very steady, cooling slightly to about 18°C on September 17 (Fig. I.51). A long, steady cooling followed until November 16, when the temperature fell to about 9°C (Figs. I.106, I.111, and I.116).

REFERENCES

Birchfield, G. E., and D. R. Davidson (1967), "A Case Study of Coastal Currents in Lake Michigan," Proc. 10th Conf. on Great Lakes Res., 264-273.

Monahan, E. C., and D. C. Pilgrim (1975), Coastwise Currents in the Vicinity of Chicago and Currents Elsewhere in Southern Lake Michigan, University of Michigan Technical Report.

Saunders, K. D., L. Van Loon, C. Tome, and W. Harrison (1976), Nearshore Currents at Point Beach, Wisconsin (1974-1975), ANL/WR-76-1, Energy and Environmental Systems Division, Argonne National Laboratory, Argonne, Illinois, 60439.

Snow, R. H. (1974), Water Pollution Investigation: Calumet Area of Lake Michigan, Vols. I & II, EPA Report No. EPA-905/9-74-011-A.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX I

Plots of Current and Temperature Data

Fig. I. 1. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 174-194 of 1975 (Record Q3751)

Fig. I. 2. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days
174-194 (Record Q3751)

Fig. I. 3. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
174-194 (Record Q3751)

Fig. I. 4. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 174-194 (Record Q3751)

Fig. I. 5. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 174-194 (Record Q3751)

Fig. I. 6. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 194-205 of 1975 (Record Q3751)

Fig. I. 7. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 194-205 (Record Q3751)

Fig. I. 8. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
194-205 (Record Q3751)

Fig. I. 9. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 194-205 (Record Q3751)

Fig. I. 10. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 194-205 (Record Q3751)

Fig. I. 11. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q6 for Days 174-194 of 1975 (Record Q6751)

Fig. I. 12. Current Speeds and Directions for Bendix Q-15 Meter Q6 for Days
174-194 (Record Q6751)

Fig. I. 13. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q6 for Days
174-194 (Record Q6751)

Fig. I. 14. Progressive Vector Diagram for Bendix Q-15 Meter Q6 for Days 174-194 (Record Q6751)

Fig. I. 15. Speed and Direction Histograms
for Bendix Q-15 Meter Q6 for Days 174-194 (Record Q6751)

Fig. I. 16. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q6 for Days 194-205 of 1975 (Record Q6751)

Fig. I. 17. Current Speeds and Directions for Bendix Q-15 Meter Q6 for Days 194-205 (Record Q6751)

Fig. I. 18. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q6 for Days
194-205 (Record Q6751)

Fig. I. 20. Speed and Direction Histograms
for Bendix Q-15 Meter Q6 for Days 194-205 (Record Q6751)

Fig. I. 21. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q7 for Days 175-195 of 1975 (Record Q7751)

Fig. I. 22. Current Speeds and Directions for Bendix Q-15 Meter Q7 for Days 175-195 (Record Q7751)

I. 23

Fig. I. 23. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q7 for Days
175-195 (Record Q7751)

Fig. I. 24. Progressive Vector Diagram for Bendix Q-15 Meter Q7 for Days
175-195 (Record Q7751)

Fig. I. 25. Speed and Direction Histograms
for Bendix Q-15 Meter Q7 for Days 175-195 (Record Q7751)

Fig. I. 26. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q7 for Days 195-205 of 1975 (Record Q7751)

Fig. I. 27. Current Speeds and Directions for Bendix Q-15 Meter Q7 for Days 195-205 (Record Q7751)

Fig. I. 28. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q7 for Days
195-205 (Record Q7751)

Fig. I. 29. Progressive Vector Diagram for Bendix Q-15 Meter Q7 for Days 195-205 (Record Q7751)

Fig. I. 30. Speed and Direction Histograms
for Bendix Q-15 Meter Q7 for Days 195-205 (Record Q7751)

Fig. I. 31. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 175-195 of 1975 (Record Q8751)

Fig. I. 32. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days 175-195 (Record Q8751)

Fig. I. 33. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
175-195 (Record Q8751)

Fig. I. 34. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days 175-195 (Record Q8751)

Fig. I. 35. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 175-195 (Record Q8751)

Fig. I. 36. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 195-205 of 1975 (Record Q8751)

Fig. I. 37. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days
195-205 (Record Q8751)

Fig. I. 38. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
195-205 (Record Q8751)

Fig. I. 39. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days 195-205 (Record Q8751)

Fig. I. 40. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 195-205 (Record Q8751)

Fig. I. 41. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 205-225 of 1975 (Record Q3752)

Fig. I. 42. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 205-225 (Record Q3752)

Fig. I. 43. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
205-225 (Record Q3752)

Fig. I. 44. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days
205-225 (Record Q3752)

Fig. I. 45. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 205-225 (Record Q3752)

Fig. I. 46. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 225-245 of 1975 (Record Q3752)

Fig. I. 47. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 225-245 (Record Q3752)

Fig. I. 48. Scattergram for Current Velocities and U, V, Current -
 Component Histograms for Bendix Q-15 Meter Q3 for Days
 225-245 (Record Q3752)

Fig. I. 49. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 225-245 (Record Q3752)

Fig. I. 50. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 225-245 (Record Q3752)

Fig. I. 51. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 245-260 of 1975 (Record Q3752)

Fig. I. 52. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 245-260 (Record Q3752)

Fig. I. 53. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
245-260 (Record Q3752)

Fig. I. 54. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 245-280 (Record Q3752)

Fig. I. 55. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 245-260 (Record Q3752)

Fig. I. 56. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q5 for Days 205-225 of 1975 (Record Q5752)

Fig. I. 57. Current Speeds and Directions for Bendix Q-15 Meter Q5 for Days 205-225 (Record Q5752)

Fig. I. 58. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q5 for Days
205-225 (Record Q5752)

Fig. I. 59. Progressive Vector Diagram for Bendix Q-15 Meter Q5 for Days
205-225 (Record Q5752)

Fig. I. 60. Speed and Direction Histograms
for Bendix Q-15 Meter Q5 for Days 205-225 (Record Q5752)

Fig. I. 61. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q5 for Days 225-245 of 1975 (Record Q5752)

Fig. I. 62. Current Speeds and Directions for Bendix Q-15 Meter Q5 for Days 225-245 (Record Q5752)

Fig. I. 63. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q5 for Days
225-245 (Record Q5752)

Fig. I. 64. Progressive Vector Diagram for Bendix Q-15 Meter Q5 for Days
225-245 (Record Q5752)

Fig. I. 65. Speed and Direction Histograms
for Bendix Q-15 Meter Q5 for Days 225-245 (Record Q5752)

Fig. I. 66. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q6 for Days 205-225 of 1975 (Record Q6752)

Fig. I. 67. Current Speeds and Directions for Bendix Q-15 Meter Q6 for Days 205-225 (Record Q6752)

Fig. I. 68. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q6 for Days
205-225 (Record Q6752)

Fig. I. 69. Progressive Vector Diagram for Bendix Q-15 Meter Q6 for Days
205-225 (Record Q6752)

Fig. I. 70. Speed and Direction Histograms
for Bendix Q-15 Meter Q6 for Days 205-225 (Record Q6752)

Fig. I. 71. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q6 for Days 225-235 of 1975 (Record Q6752)

Fig. I. 72. Current Speeds and Directions for Bendix Q-15 Meter Q6 for Days 225-235 (Record Q6752)

Fig. I. 73. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q6 for Days
225-235 (Record Q6752)

Fig. I. 74. Progressive Vector Diagram for Bendix Q-15 Meter Q6 for Days
225-235 (Record Q6752)

Fig. I. 75. Speed and Direction Histograms
for Bendix Q-15 Meter Q6 for Days 225-235 (Record Q6752)

Fig. I. 76. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q7 for Days 205-225 of 1975 (Record Q7752)

Fig. I. 77. Current Speeds and Directions for Bendix Q-15 Meter Q7 for Days 205-225 (Record Q7752)

Fig. I. 78. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q7 for Days
205-225 (Record Q7752)

Fig. I. 79. Progressive Vector Diagram for Bendix Q-15 Meter Q7 for Days 205-225 (Record Q7752)

Fig. I. 80. Speed and Direction Histograms
for Bendix Q-15 Meter Q7 for Days 205-225 (Record Q7752)

Fig. I. 81. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q7 for Days 225-245 of 1975 (Record Q7752)

Fig. I. 82. Current Speeds and Directions for Bendix Q-15 Meter Q7 for Days 225-245 (Record Q7752)

Fig. I. 83. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q7 for Days
225-245 (Record Q7752)

Fig. I. 84. Progressive Vector Diagram for Bendix Q-15 Meter Q7 for Days 225-245 (Record Q7752)

Fig. I. 85. Speed and Direction Histograms
for Bendix Q-15 Meter Q7 for Days 225-245 (Record Q7752)

Fig. I. 86. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q7 for Days 245-260 of 1975 (Record Q7752)

Fig. I. 87. Current Speeds and Directions for Bendix Q-15 Meter Q7 for Days 245-260 (Record Q7752)

Fig. I. 88. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q7 for Days
245-260 (Record Q7752)

Fig. I. 89. Progressive Vector Diagram for Bendix Q-15 Meter Q7 for Days 245-260 (Record Q7752)

Fig. I. 90. Speed and Direction Histograms
for Bendix Q-15 Meter Q7 for Days 245-260 (Record Q7752)

Fig. I. 91. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 205-225 of 1975 (Record Q8752)

Fig. I. 92. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days
205-225 (Record Q8752)

Fig. I. 93. Scattergram for Current Velocities and U, V, Current -
 Component Histograms for Bendix Q-15 Meter Q8 for Days
 205-225 (Record Q8752)

Fig. I. 94. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days
205-225 (Record Q8752)

Fig. I. 95. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 205-225 (Record Q8752)

Fig. I. 96. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 225-245 of 1975 (Record Q8752)

Fig. I. 97. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days
225-245 (Record Q8752)

Fig. I. 98. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
225-245 (Record Q8752)

Fig. I. 99. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days
225-245 (Record Q8752)

Fig. I.100. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 225-245 (Record Q8752)

Fig. I.101. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 245-260 of 1975 (Record Q8752)

Fig. I.102. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days 245-260 (Record Q8752)

Fig. I.103. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
245-260 (Record Q8752)

Fig. I.104. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days 245-260 (Record Q8752)

Fig. I.105. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 245-260 (Record Q8752)

Fig. I.106. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 260-280 of 1975 (Record Q3753)

Fig. I.107. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 260-280 (Record Q3753)

Fig. I.108. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
260-280 (Record Q3753)

Fig. I.109. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 260-280 (Record Q3753)

Fig. I.110. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 260-280 (Record Q3753)

Fig. I.111. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 280-300 of 1975 (Record Q3753)

Fig. I.112. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 280-300 (Record Q3753)

Fig. I.113. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
280-300 (Record Q3753)

Fig. I.114. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 280-300 (Record Q3753)

Fig. I.115. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 280-300 (Record Q3753)

Fig. I.116. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q3 for Days 300-320 of 1975 (Record Q3753)

Fig. I.117. Current Speeds and Directions for Bendix Q-15 Meter Q3 for Days 300-320 (Record Q3753)

Fig. I.118. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q3 for Days
300-320 (Record Q3753)

Fig. I.119. Progressive Vector Diagram for Bendix Q-15 Meter Q3 for Days 300-320 (Record Q3753)

Fig. I.120. Speed and Direction Histograms
for Bendix Q-15 Meter Q3 for Days 300-320 (Record Q3753)

Fig. I.121. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q5 for Days 260-280 of 1975 (Record Q5753)

Fig. I.122. Current Speeds and Directions for Bendix Q-15 Meter Q5 for Days
260-280 (Record Q5753)

Fig. I.123. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q5 for Days
260-280 (Record Q5753)

Fig. I.124. Progressive Vector Diagram for Bendix Q-15 Meter Q5 for Days 260-280 (Record Q5753)

Fig. I.125. Speed and Direction Histograms
for Bendix Q-15 Meter Q5 for Days 260-280 (Record Q5753)

Fig. I.126. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q5 for Days 280-300 of 1975 (Record Q5753)

Fig. I.127. Current Speeds and Directions for Bendix Q-15 Meter Q5 for Days 280-300 (Record Q5753)

Fig. I.128. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q5 for Days
280-300 (Record Q5753)

Fig. I.129. Progressive Vector Diagram for Bendix Q-15 Meter Q5 for Days 280-300 (Record Q5753)

Fig. I.130. Speed and Direction Histograms
for Bendix Q-15 Meter Q5 for Days 280-300 (Record Q5753)

Fig. I.131. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q5 for Days 300-320 of 1975 (Record Q5753)

Fig. I.132. Current Speeds and Directions for Bendix Q-15 Meter Q5 for Days 300-320 (Record Q5753)

Fig. I.133. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q5 for Days
300-320 (Record Q5753)

Fig. I.134. Progressive Vector Diagram for Bendix Q-15 Meter Q5 for Days 300-320 (Record Q5753)

Fig. I.135. Speed and Direction Histograms
for Bendix Q-15 Meter Q5 for Days 300-320 (Record Q5753)

Fig. I.136. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q6 for Days 260-280 of 1975 (Record Q6753)

Fig. I.137. Current Speeds and Directions for Bendix Q-15 Meter Q6 for Days 260-280 (Record Q6753)

Fig. I.138. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q6 for Days
260-280 (Record Q6753)

Fig. I.139. Progressive Vector Diagram for Bendix Q-15 Meter Q6 for Days 260-280 (Record Q6753)

Fig. I.140. Speed and Direction Histograms
for Bendix Q-15 Meter Q6 for Days 260-280 (Record Q6753)

Fig. I.141. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q6 for Days 280-295 of 1975 (Record Q6753)

Fig. I.142. Current Speeds and Directions for Bendix Q-15 Meter Q6 for Days 280-295 (Record Q6753)

Fig. I.143. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q6 for Days
280-295 (Record Q6753)

Fig. I.144. Progressive Vector Diagram for Bendix Q-15 Meter Q6 for Days 280-295 (Record Q6753)

Fig. I.145. Speed and Direction Histograms
for Bendix Q-15 Meter Q6 for Days 280-295 (Record Q6753)

Fig. I.146. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 260-280 of 1975 (Record Q8753)

Fig. I.147. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days 260-280 (Record Q8753)

Fig. I.148. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
260-280 (Record Q8753)

Fig. I.149. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days 260-280 (Record Q8753)

Fig. I.150. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 260-280 (Record Q8753)

Fig. I.151. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 280-300 of 1975 (Record Q8753)

Fig. I.152. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days 280-300 (Record Q8753)

Fig. I.153. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
280-300 (Record Q8753)

Fig. I.154. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days 280-300 (Record Q8753)

Fig. I.155. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 280-300 (Record Q8753)

Fig. I.156. Specific Kinetic Energy and U, V Current Components for Bendix Q-15 Meter Q8 for Days 300-315 of 1975 (Record Q8753)

Fig. I.157. Current Speeds and Directions for Bendix Q-15 Meter Q8 for Days 300-315 (Record Q8753)

Fig. I.158. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Bendix Q-15 Meter Q8 for Days
300-315 (Record Q8753)

Fig. I.159. Progressive Vector Diagram for Bendix Q-15 Meter Q8 for Days 300-315 (Record Q8753)

Fig. I.160. Speed and Direction Histograms
for Bendix Q-15 Meter Q8 for Days 300-315 (Record Q8753)

Fig. I.161. Specific Kinetic Energy and U, V Current Components for Braincon Meter B6 for Days 321-341 of 1975 (Record B96758)

Fig. I.162. Current Speeds and Directions for Braincon Meter B6 for Days 321-341 (Record B96758)

Fig. I.163. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Braincon Meter B6 for Days
321-341 (Record B96758)

Fig. I.164. Progressive Vector Diagram for Braincon Meter B6 for Days 321-341 (Record B96758)

Fig. I.165. Speed and Direction Histograms
for Braincon Meter B6 for Days 321-341 (Record B96758)

Fig. I.166. Specific Kinetic Energy and U, V Current Components for Braincon Meter B6 for Days 341-353 of 1975 (Record B96758)

Fig. I.167. Current Speeds and Directions for Braincon Meter B6 for Days 341-353 (Record B96758)

Fig. I.168. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Braincon Meter B6 for Days
341-353 (Record B96758)

Fig. I.169. Progressive Vector Diagram for Braincon Meter B6 for Days
341-353 (Record B96758)

Fig. I.170. Speed and Direction Histograms
for Braincon Meter B6 for Days 341-353 (Record B96758)

Fig. I.171. Specific Kinetic Energy and U, V Current Components for Braincon Meter B7 for Days 321-341 of 1975 (Record B97758)

Fig. I.172. Current Speeds and Directions for Braincon Meter B7 for Days 321-341 (Record B97758)

Fig. I.173. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Braincon Meter B7 for Days
321-341 (Record B97758)

Fig. I.174. Progressive Vector Diagram for Braincon Meter B7 for Days 321-341 (Record B97758)

Fig. I.175. Speed and Direction Histograms
for Braincon Meter B7 for Days 321-341 (Record B97758)

Fig. I.176. Specific Kinetic Energy and U, V Current Components for Braincon Meter B7 for Days 341-349 of 1975 (Record B97758)

Fig. I.177. Current Speeds and Directions for Braincon Meter B7 for Days 341-349 (Record B97758)

Fig. I.178. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Braincon Meter B7 for Days
341-349 (Record B97758)

Fig. I.179. Progressive Vector Diagram for Braincon Meter B7 for Days 341-349 (Record B97758)

Fig. I.180. Speed and Direction Histograms
for Braincon Meter B7 for Days 341-349 (Record B97758)

Fig. I.181. Specific Kinetic Energy and U, V Current Components for Braincon Meter B8 for Days 321-341 of 1975 (Record B98758)

Fig. I.182. Current Speeds and Directions for Braincon Meter B8 for Days 321-341 (Record B98758)

I.183

Fig. I.183. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Braincon Meter B8 for Days
321-341 (Record B98758)

Fig. I.184. Progressive Vector Diagram for Braincon Meter B8 for Days 321-341 (Record B98758)

Fig. I.185. Speed and Direction Histograms
for Braincon Meter B8 for Days 321-341 (Record B98758)

Fig. I.186. Specific Kinetic Energy and U, V Current Components for Braincon Meter B8 for Days 341-361 of 1975 (Record B98758)

Fig. I.187. Current Speeds and Directions for Braincon Meter B8 for Days 341-361 (Record B98758)

Fig. I.188. Scattergram for Current Velocities and U, V, Current -
Component Histograms for Braincon Meter B8 for Days
341-361 (Record B98758)

Fig. I.189. Progressive Vector Diagram for Braincon Meter B8 for Days
341-361 (Record B98758)

Fig. I.190. Speed and Direction Histograms
for Braincon Meter B8 for Days 341-361 (Record B98758)

APPENDIX II

Listings of Data Processing Programs

```

00000100 Q15PROC: PROC OPTIONS (MAIN) ;
00000200 /*
00000300
00000400 *****
00000500 PURPOSE : TO READ RAW BENDIX Q-15 DATA , DELETE BAD DATA , INTERPOLATE TO MISSING POINTS
00000600 AND CREATE AN SDF FILE.
00000700
00000800 AUTHOR : KIM DAVID SAUNDERS
00000900 DATE : DECEMBER 1975
00001000 LANGUAGE:PL/I - IBM OPTIMIZING COMPILER
00001100
00001200 INPUT INFORMATION
00001300
00001400 1. THE RAW DATA WILL BE READ FROM A DATASET ASSOCIATED WITH DDNAME 'INFILE'.
00001500 2. THE PROGRAM CONTROL PARAMETERS ARE READ WITH A GET DATA AND ARE:
00001600 1) TIME_INTERVAL - THE SAMPLING INTERVAL IN MINUTES.
00001700 THE DEFAULT IS 8 MINUTES.
00001800 2) STARTING_DAY - THE DAY OF THE MONTH THAT THE RECORD STARTS ON.
00001900 3) MONTH - THE MONTH IN WHICH THE RECORD BEGINS.
00002000 4) HOUR - THE HOUR IN WHICH THE RECORD BEGINS.
00002100 5) MINUTE - THE MINUTE IN WHICH THE RECORD BEGINS.
00002200 6) LEAP_YEAR - IF IT IS A LEAP YEAR, ENTER 'YES', OTHERWISE OMIT OR
00002300 ENTER 'NO'.
00002400 7) RECORD_LENGTH - THE LENGTH OF THE RECORDING IN DAYS. DEFAULT IS 60.
00002500 8) X_POSITION - THE X POSITION IN METERS IN THE LOCAL COORDINATE
00002600 SYSTEM OF THE STATION.
00002700 9) Y_POSITION - SIMILARLY FOR THE Y LOCATION.
00002800 10) INSTRUMENT_CODE - THE CHARACTER INSTRUMENT CODE.
00002900 11) YEAR - THE LAST TWO DIGITS OF THE YEAR. THE DEFAULT IS 75.
00003000
00003100 3. THE INTERPOLATE DATA WILL THEN BE WRITTEN TO A DATASET WITH THE
00003200 ASSOCIATED DDNAME - OUTFILE.
00003300
00003400 *****
00003500 */
00003600
00003700 /*
00003800 *****
00003900 DATA DEFINITION COMMON TO THE WHOLE PROGRAM
00004000 *****
00004100 */
00004200
00004300 DCL INFILE FILE INPUT,
00004400 UTFILE1 FILE ,
00004500 UTFILE2 FILE,
00004600 (N_ORIGINAL_RECORDS,N_TOTAL_ERRORS,N_CURRENT_RECORDS) FIXED BIN (31) INIT (0) ,
00004700 TIME_INTERVAL FIXED BIN (31) INIT (8) ,
00004800 (STARTING_DAY,HOUR,MINUTE,MONTH) FIXED BIN (31) INIT (0) ,
00004900 LEAP_YEAR CHAR (3) VARYING INIT ('NO') ,
00005000
00005100 RECORD_LENGTH FIXED BIN (31) INIT (60) ,
00005200 PRESCALING FIXED BIN (31) INIT (32) ,
00005300 K FLOAT DEC (6) INIT (32.175) ,
00005400 SPEED_CONVERSION FLOAT DEC (6) INIT (2.145) ,
00005500 AVERAGE_SPEED FL CAT DEC (6) INIT (0) ,
00005600 MIN_DIR_COUNT FIXED BIN (31) INIT (255) ,
00005700 MAX_DIR_COUNT FIXED BIN (31) INIT (0) ,
00005800 SUM_SPEED_SQ FLOAT DEC (6) INIT (0.0) ,
00005900 STD_DEV_SPEEDS FLOAT DEC (6) INIT (0.0) ,
00006000 X_POSITION FIXED DEC (5) INIT (0) ,
00006100 Y_POSITION FIXED DEC (5) INIT (0) ,
00006200

```

```

00006300      INSTRUMENT_CODE CHAR(3) INIT('ZZZ'),
00006400      YEAR      FIXED DEC (3) INIT (75),
00006500          SUP_RAW_DATA_PRINTING CHAR(3) VARYING INIT('NO'),
00006600          SUP_DIR_PRINTING      CHAR(3) VARYING INIT('NO'),
00006700          SUP_FINAL_PRINTING    CHAR(3) VARYING INIT('NO'),
00006800      (ATAND,COSD,FLOAT,SIND,SQRT,SUM,SUBSTR,UNSPEC) BUILTIN,
00006900      SYSIN FILE INPUT STREAM,
00007000      SYSPRINT FILE OUTPUT PRINT STREAM,
00007100          FUZZ FIXED BIN (31) INIT (0) ;
00007200
00007300          ON TRANSMIT (INFILE) ;
000074001
00007500  /*
00007600  ****
00007700      PROCEDURE DECLARATIONS
00007800  ****
00007900  */
00008000
00008100  READ_RAW_DATA: PROC;
00008200
00008300  /*
00008400  ****
00008500  PURPOSE : TO READ IN THE DATA FROM THE DATASET CREATED FROM THE TRANSCRIBED TAPE, TEST
00008600            THE VALIDITY, IGNORE, IF NOT VALID, ACCEPT IF VALID, CONVERT TO SPEEDS
00008700            AND DIRECTIONS AND TO WRITE THE DATA TO A TEMPORARY FILE.
00008800  ****
00008900  */
00009000
00009100  /*
00009200  ****
00009300      DECLARATIONS LOCAL TO READ_RAW_DATA
00009400  ****
00009500  */
00009600  DCL
00009700      ANOTHER_RECORD BIT (1) INIT ('1'B),
00009800      INPUT_DATA(28) CHAR (1) BASED ,
00009900      1 RAW_RECORD,
00010000          2 DIR(4) FIXED BIN (31),
00010100          2 TEMP(2) FIXED BIN (31),
00010200          2 SPD FIXED BIN (31),
00010300          2 CLOCK FIXED BIN (31),
00010400          2 FLAG FIXED BIN (31),
00010500          P PTR,
00010600          TRK FIXED BIN (31) INIT (0),
00010700          RAW_DATA(8) BIT (32) ,
00010800          PERIOD FLOAT DEC (6) INIT (0),
00010900      1 OUT_RECORD,
00011000          2 TRACK FIXED BIN (31),
00011100          2 DIRECTION(4) FIXED BIN (31),
00011200          2 TEMPS(2) FLOAT DEC (6),
00011300          2 SPEED FIXED BIN (31),
00011400          2 CLOCK FIXED BIN (31),
00011500          2 FLAGS FIXED BIN (31);
00011600
00011700
00011800      ON ENDFILE (INFILE) ANOTHER_RECORD = '0'B;
00011900      ON ENDFILE (SYSIN ) ;
00012000      ANOTHER_RECORD = '1'B;
00012100
000122001
00012300  /*
00012400  ****
00012500      PROCEDURE DECLARATIONS LOCAL TO READ_RAW_DATA

```



```

00012600 *****
00012700 */
00012800     GET_MORE_DATA :   PROC ;
00012900 /*
00013000 *****
00013100     PURPOSE OF THIS PROCEEDURE : TO READ IN A RECORD OF DATA, INCREMENT THE
00013200     DATA COUNT, AND REJECT AN OBVIOUSLY BAD RECORD ( IF THE CLOCK COUNT IS
00013300     EITHER = 0 OR IS GREATER THEN THE CURRENT DATA COUNT.)
00013400 *****
00013500 */
00013600     DCL I FIXED BIN (31) ,
00013700           BIT_STRING BIT (16) VARYING INIT ('');
00013800     READ FILE (INFILE) SET (P);
00013900     IF ANOTHER_RECORD THEN K1: DO;
00014000     RAW_DATA = (32)'0'B;
00014100     LOOP2 : DO I = 1 TO 7;
00014200           SUBSTR(RAW_DATA(I),25,8) = SUBSTR(UNSPEC(P -> INPUT_DATA(2*I-1)),
00014300           5,4)
00014400           || SUBSTR(UNSPEC(P -> INPUT_DATA(2*I)),5,4) ;
00014500     END LOOP2;
00014600     BIT_STRING = 'B';
00014700     LOOP3 : DO I = 1 TO 4;
00014800           BIT_STRING = BIT_STRING || SUBSTR(UNSPEC(P -> INPUT_DATA(14+I)),5,4);
00014900     END LOOP3;
00015000
00015100     SUBSTR(RAW_DATA(8),17,16) = BIT_STRING;
00015200     TRK = TRK + 1;
00015300     LOOP4 : DO I = 1 TO 4;
00015400           UNSPEC(RAW_RECORD.DIR(I))=RAW_DATA(I);
00015500     END LOOP4;
00015600
00015700     UNSPEC(RAW_RECORD.TEMP(1))=RAW_DATA(5);
00015800     UNSPEC(RAW_RECORD.TEMP(2))=RAW_DATA(6);
00015900     UNSPEC(RAW_RECORD.SPD)   =RAW_DATA(7);
00016000     UNSPEC(RAW_RECORD.CLOCK) =RAW_DATA(8);
00016100     IF RAW_RECORD.CLOCK > TRK + FUZZ | RAW_RECORD.CLOCK = 0 THEN
00016200       RAW_RECORD.FLAG = 1;
00016300     ELSE RAW_RECORD.FLAG=0;
00016400     IF SUP_RAW_DATA_PRINTING ^= 'YES' THEN
00016500       PUT FILE(SYSPRINT) SKIP EDIT (TRK,RAW_RECORD)
00016600       (X(1),F(5),X(2),9(F(4),X(1)) ) ;
00016700     END K1;
00016800     RETURN;
00016900 END GET_MORE_DATA;
00017000 PROCESS_IT :   PROC ;
00017100
00017200 /*
00017300 *****
00017400     PURPOSE : TO IGNORE BAD RECORDS, DEFINED AS RECORDS WITH SPEEDS TOO HIGH,
00017500     TEMPERATURES TOO HIGH OR LOW,OR TIMES TOO SMALL OR LARGE, TO CONVERT THE
00017600     RAW DATA TO SPEEDS AND TEMPERATURES AND TO MAINTAIN COUNTS OF THE DATA
00017700     READ IN AND RUNNING SUMS OF THE SPEEDS AND SQUARES OF THE SPEEDS , AND TO
00017800     KEEP TRACK OF THE MAXIMUM AND MINIMUM DIGITAL DIRECTION READINGS FOR USE
00017900     IN DYNAMIC CALIBRATION IN THE LATER PROCESSING.
00018000 *****
00018100 */
00018200
00018300 DCL I FIXED BIN (31);
00018400     TEMPERATURE : PROC (COUNTS) RETURNS (FLOAT DEC (6) );
00018500 /*
00018600 *****
00018700

```

```

00018800      PURPOSE : TO PRODUCE THE THERMISTOR TEMPERATURE IN DEGREES CELSIUS, GIVEN
00018900      THE DIGITAL READING .
00019000      A SECOND ORDER POLYNOMIAL IS USED.
00019100      *****
00019200      */
00019300          DCL COUNTS FIXED BIN (31),
00019400          (T,X) FLOAT DEC (6);
00019500          X = COUNTS;
00019600          T = 54.428000 - X*(0.31386400 - X*0.00034178);
00019700          RETURN (T);
00019800      END TEMPERATURE;
00019900
00020000      /*
00020100      *****
00020200      MAIN CODE FOR THE PROCEDURE - PROCESS IT.
00020300      *****
00020400      */
00020500      /*
00020600      *****
00020700      TEST THE CLOCK VALUES
00020800      *****
00020900      */
00021000
00021100      IF RAW_RECORD.FLAG .NE. 0 THEN DO;
00021200          PUT FILE(SYSPRINT) SKIP LIST('***CLOCK > RECORD NUMBER OR = 0***');
00021300          PUT FILE(SYSPRINT) SKIP EDIT (TRK,RAW_RECORD) (F(5),X(2),9(F(4),X(1)));
00021400          RETURN;
00021500      END;
00021600
00021700      IF RAW_RECORD.CLOCK > 1440.0*RECORD_LENGTH THEN DO;
00021800          PUT FILE(SYSPRINT) SKIP LIST ('***CLOCK OUT OF RANGE - TOO HIGH***');
00021900          PUT FILE(SYSPRINT) SKIP EDIT (TRK,RAW_RECORD) (F(5),X(2),9(F(4),X(1)));
00022000          RETURN;
00022100      END;
00022200
00022300      /*
00022400      *****
00022500      TEST THE TEMPERATURES TO SEE IF THEY ARE IN THE ALLOWED RANGE
00022600      *****
00022700      */
00022800
00022900
00023000      OUT_RECORD.TRACK = TRK;
00023100      OUT_RECORD.DIRECTION = RAW_RECORD.DIR;
00023200
00023300      LOOP5: DO I = 1 TO 2;
00023400          OUT_RECORD.TEMPS(I) = TEMPERATURE(RAW_RECORD.TEMP(I));
00023500          IF OUT_RECORD.TEMPS(I) > 30.0 | OUT_RECORD.TEMPS(I) <= 0.0 THEN DO;
00023600              PUT FILE(SYSPRINT) SKIP LIST ('***TEMPERATURE OUT OF RANGE***');
00023700              PUT FILE(SYSPRINT) SKIP EDIT (TRK,RAW_RECORD) (F(5),X(2),9(F(4),X(1)));
00023800              RETURN;
00023900          END;
00024000
00024100      END LOOP5;
00024200      /*
00024300      *****
00024400      TEST THE SPEEDS FOR RANGE VALIDITY
00024500      *****
00024600      */
00024700
00024800
00024900      OUT_RECORD.SPEED = SPEED_CONVERSION * RAW_RECORD.SPD;
00025000      IF OUT_RECORD.SPEED > 350 | OUT_RECORD.SPEED < 0 THEN DO;

```

```

00025100          PUT FILE(SYSPRINT) SKIP LIST('***SPEED OUT OF RANGE***');
00025200          PUT FILE(SYSPRINT) SKIP EDIT (TRK,RAW_RECORD) (F(5),X(2),9(F(4),X(1)));
00025300          RETURN;
00025400      END;
00025500
00025600  /*
00025700  ****
00025800      TEST THE DIRECTIONS FOR A ZERO COUNT
00025900  ****
00026000  */
00026100      LOOP9:DO I=1 TO 4;
00026200          IF RAW_RECORD.DIR(I) = 0 THEN DO;
00026300              PUT SKIP LIST ('***DIRECTION WITH ZERO COUNT ENCOUNTERED - RECORD '
00026400              ||'IGNORED');
00026500              PUT SKIP EDIT(TRK,RAW_RECORD) (F(5),X(2),9(F(4),X(1)));
00026600              RETURN;
00026700          END;
00026800      END LOOP9;
00026900      OUT_RECORD.CLOCK = RAW_RECORD.CLOCK;
00027000      OUT_RECORD.FLAGS = 0;
00027100
00027200      WRITE FILE (UTFILE1) FROM (OUT_RECORD);
00027300
00027400      LOOP6 : DO I = 1 TO 4;
00027500          IF RAW_RECORD.DIR(I)>MAX_DIR_COUNT THEN MAX_DIR_COUNT=RAW_RECORD.DIR(I);
00027600          IF RAW_RECORD.DIR(I)<MIN_DIR_COUNT THEN MIN_DIR_COUNT=RAW_RECORD.DIR(I);
00027700      END LOOP6;
00027800
00027900      SUM_SPEED_SQ = SUM_SPEED_SQ + (OUT_RECORD.SPEED)**2;
00028000      AVERAGE_SPEED = AVERAGE_SPEED + OUT_RECORD.SPEED;
00028100      N_CURRENT_RECORDS = N_CURRENT_RECORDS + 1;
00028200
00028300      RETURN;
00028400  END PROCESS_IT;
00028500
00028600  /*
00028700  ****
00028800      READ PROGRAM CONTROL PARAMETERS
00028900  ****
00029000  */
00029100      PUT SKIP LIST('INPUT PROGRAM CONTROL PARAMETERS ');
00029200      PUT SKIP LIST('STARTING_DAY,HOUR,MINUTE, AND MONTH ARE NEEDED');
00029300      PUT SKIP LIST('OTHER PARAMETERS THAT MAY BE USED ARE:');
00029400      PUT SKIP LIST('RECORD_LENGTH (DEFAULT-60),TIME_INTERVAL (DEFAULT - 8)');
00029500      PUT SKIP LIST('X_POSITION,YPOSITION,INSTRUMENT_CODE,LEAP_YEAR');
00029600      PUT SKIP LIST('SUP_RAW_DATA_PRINTING,SUP_DIR_PRINTING,SUP_FINAL_PRINTING');
00029700      GET FILE(SYSIN) DATA (
00029800      TIME_INTERVAL,STARTING_DAY,HOUR,MINUTE,MONTH,LEAP_YEAR,RECORD_LENGTH,
00029900      X_POSITION,Y_POSITION,YEAR,INSTRUMENT_CODE,
00030000      SUP_RAW_DATA_PRINTING,SUP_DIR_PRINTING,SUP_FINAL_PRINTING,
00030100      FUZZ);
00030200
00030300  /*
00030400  ****
00030500      TEST FOR PROPER PRESCALING FACTOR AND SET A NEW ONE IF NEEDED, OR IF
00030600      THE FACTOR IS INVALID, STOP THE PROGRAM WITH AN APPROPRIATE WARNING.
00030700  ****
00030800  */
00030900
00031000      IF TIME_INTERVAL = 8 | TIME_INTERVAL = 40 THEN
00031100          LI: DO;
00031200          IF TIME_INTERVAL = 8 THEN PRESCALING = 32;

```



```

00031300      IF TIME_INTERVAL = 40 THEN PRESCALING = 128;
00031400      PERIOD = 60.0* TIME_INTERVAL ;
00031500      SPEED_CONVERSION = K*FLOAT(PRESCALING)/FLOAT(PERIOD);
00031600      PUT SKIP(2) LIST('*****SPEED_CONVERSION*****',SPEED_CONVERSION);
00031700      END L1;
00031800  ELSE
00031900      L2: DO;
00032000          PUT FILE(SYSPRINT) SKIP LIST ('***ERROR!!!! INVALID TIME INTERVAL TRIED'
00032100          '||' - PROGRAM TERMINATING !!!!! *****');
00032200          STOP;
00032300          END L2;
00032400      PUT FILE(SYSPRINT) SKIP DATA (PRESCALING, TIME_INTERVAL, K, PERIOD);
00032500      PUT FILE(SYSPRINT) PAGE;
000326001
00032700  /*
00032800  *****
00032900      MAIN PROCESSING SEQUENCE FOR READ_RAW_DATA
00033000  *****
00033100  */
00033200
00033300      OPEN FILE ( INFILE) INPUT RECORD SEQUENTIAL;
00033400      OPEN FILE (UTFILE1) OUTPUT RECORD SEQUENTIAL;
00033500
00033600      TRK = 0;
00033700      PUT FILE(SYSPRINT) PAGE LIST(' TRACK      D1      D2      D3      D4      T1      T2'||
00033800      ' SPD CLK  FLG');
00033900      LOOP1: DO WHILE ( ANOTHER_RECORD );
00034000          CALL GET_MORE_DATA;
00034100          IF ANOTHER_RECORD THEN CALL PROCESS_IT;
00034200          END LOOP1;
00034300      CLOSE FILE(INFILE);
00034400      CLOSE FILE(UTFILE1);
00034500  /*
00034600  *****
00034700      UPDATE AVERAGES, SUMS OF SQUARES OF SPEEDS,AND DIRECTION BOUNDS.
00034800  *****
00034900  */
00035000
00035100      IF N_CURRENT_RECORDS = 0 THEN DO;
00035200          PUT FILE(SYSPRINT) PAGE LIST ('*****NO RECORDS ACCEPTED - PROGRAM '||
00035300          'TERMINATING *****');
00035400          STOP;
00035500      END;
00035600
00035700      AVERAGE_SPEED = AVERAGE_SPEED /N_CURRENT_RECORDS;
00035800      STD_DEV_SPEEDS = SUM_SPEED_SQ / N_CURRENT_RECORDS;
00035900      STD_DEV_SPEEDS = SQRT ( STD_DEV_SPEEDS);
00036000
00036100      PUT FILE(SYSPRINT) SKIP DATA ( AVERAGE_SPEED, STD_DEV_SPEEDS, MIN_DIR_COUNT,
00036200      MAX_DIR_CCUNT, N_CURRENT_RECORDS,TRK);
00036300
00036400      PUT FILE(SYSPRINT) PAGE;
00036500
00036600  .END READ_RAW_DATA;
000367001
00036800  INTERMEDIATE_TESTING: PROC;
00036900
00037000  /*
00037100  *****
00037200      PURPOSE : TO COMPUTE THE DIRECTIONS FROM THE DIGITAL DATA, REJECT RECORDS
00037300      CONTAINING TWO OR MORE DIRECTIONS IN DISAGREEMENT BY MORE THAN 90 DEG,
00037400      AND TO TEST FOR SPEEDS OUT OF RANGE AND CLOCK PULSES NOT IN ORDER.
00037500  *****

```

```

00037600 */
000377001
00037800 /*
00037900 *****
00038000 DATA DECLARATIONS LOCAL TO INTERMEDIATE TESTING.
00038100 *****
00038200 */
00038300
00038400 DCL
00038500 (A,BETA,F,CF,THETA) FLOAT DEC (6) INIT (0),
00038600 (PREVIOUS_CLOCK,CURRENT_CLOCK,I,J,K,L,M) FIXED BIN (31) INIT (0),
00038700 DEGREE_ANGLE (4) FLOAT DEC (6),
00038800 1 IN_RECORD,
00038900 2 TRACK FIXED BIN (31),
00039000 2 DIR (4) FIXED BIN (31),
00039100 2 TEMPS (2) FLOAT DEC (6),
00039200 2 SPEED FIXED BIN (31),
00039300 2 CLOCK FIXED BIN (31),
00039400 2 FLAGS FIXED BIN (31),
00039500
00039600 1 OUT_RECORD,
00039700 2 TRACK FIXED BIN (31),
00039800 2 DIR FIXED BIN (31),
00039900 2 TEMP FLOAT DEC (6),
00040000 2 U FIXED BIN (31),
00040100 2 V FIXED BIN (31),
00040200 2 CLOCK FIXED BIN (31),
00040300 2 FLAGS FIXED BIN (31),
00040400
00040500 ANOTHER_RECORD BIT (1) INIT ('1'B),
00040600
00040700 SPEED_MAX FLOAT DEC (6) INIT (0);
000408001
00040900 /*
00041000 *****
00041100 SET UP PARAMETERS FOR DIRECTION CONVERSION AND INITIALIZE THE
00041200 PREVIOUS CLOCK.
00041300 *****
00041400 */
00041500
00041600 PUT FILE(SYSPRINT) PAGE DATA (MIN_DIR_COUNT,MAX_DIR_COUNT);
00041700 IF MIN_DIR_COUNT < 40 | MIN_DIR_COUNT > 70 THEN MIN_DIR_COUNT = 60;
00041800 IF MAX_DIR_COUNT < 230 | MAX_DIR_COUNT > 255 THEN MAX_DIR_COUNT = 249;
00041900
00042000 PREVIOUS_CLOCK = 0;
00042100 F = 464;
00042200 BETA = MIN_DIR_COUNT / F ;
00042300 CF = MAX_DIR_COUNT / F ;
00042400 A = ((1.00000 - CF)/(CF - BETA))*360.000 ;
00042500 SPEED_MAX = AVERAGE_SPEED + 3.00000 * STD_DEV_SPEEDS ;
000426001
00042700 ANGLE : PROC (COUNTS) RETURNS (FIXED BIN (31) );
00042800
00042900 /*
00043000 *****
00043100 PURPOSE : TO COMPUTE THE DIRECTIONS, CLOCKWISE FROM NORTH, IN DEGREES,
00043200 GIVEN THE DIGITAL DIRECTION INFORMATION.
00043300 *****
00043400 */
00043500 DCL COUNTS FIXED BIN (31);
00043600 (CF,X) FLOAT DEC (6);
00043700 CF = COUNTS/F;

```

```

00043800      X = A* (CF - BETA)/(1.000000 - CF)      ;
00043900      IF X > 360 THEN DO;
00044000          X = 360;
00044100          PUT FILE(SYSPRINT) SKIP LIST ('**ANGLE EXCEEDS 360 DEGREES**');
00044200          PUT FILE(SYSPRINT) SKIP DATA ( IN_RECORD);
00044300      END;
00044400
00044500      IF X < 0 THEN DO;
00044600          PUT FILE(SYSPRINT) SKIP LIST ('**ANGLE LESS THAN 0 DEGREES**');
00044700          PUT FILE(SYSPRINT) SKIP DATA ( IN_RECORD);
00044800      END;
00044900
00045000      RETURN(X);
00045100  END ANGLE;
00045200
00045300  INT_PROCESS : PROC;
00045400
00045500  /*
00045600  *****
00045700      PURPOSE : TO ACTUALLY PERFORM THE TESTING DESCRIBED IN THE MAIN BODY
00045800      OF INTERMEDIATE_TESTING.
00045900  *****
00046000  */
00046100
00046200      CHECK_ANGLES : PROC ;
00046300  /*
00046400  *****
00046500      PURPOSE : TO
00046600          1 TEST TO SEE IF THREE OR MORE DIRECTIONS ARE WITHIN 90 DEGREES OF ONE
00046700          ANOTHER. IF SO , ACCEPT THE RECORD.
00046800          2 IF NOT - IGNORE IT.
00046900          3 AVERAGE THE 3 OR 4 BEST DIRECTIONS PER RECORD AND RETURN.
00047000  *****
00047100  */
00047200
00047300  /*
00047400  *****
00047500      DECLARATIONS LOCAL TO CHECK_ANGLES
00047600  *****
00047700  */
00047800      DCL
00047900          (NORTH(8),EAST(8),LENGTH(4),LENGE,MAX_LENGTH,N,E,SN,SE)
00048000          FLOAT DEC (6) INIT (0),
00048100          (I,J,K,INDEX_MAX) FIXED BIN (31) INIT (0);
00048200
00048300  /*
00048400  *****
00048500      CHECK TO SEE IF ALL FOUR ANGLES AGREE
00048600  *****
00048700  */
00048800      K = 0;
00048900      L1: DO I = 1 TO 2;
00049000          L2: DO J = 1 TO 4;
00049100              K = K + 1;
00049200              NORTH(K)=COSD(DEGREE_ANGLE(J));
00049300              EAST(K) =SIND(DEGREE_ANGLE(J));
00049400          END L2;
00049500      END L1;
00049600
00049700      LENGE = SQRT((SUM(NORTH)/8.0)**2+(SUM(EAST)/8.0)**2);
00049800
00049900  /*
00050000  *****

```



```

00050100      IF ALL FOUR ANGLES AGREE THEN UPDATE THE DIRECTION AND ESCAPE
00050200      *****
00050300      */
00050400          IF LENG >= 0.7071 THEN DO ;
00050500              N = SUM(NORTH)/8.0;
00050600              E = SUM(EAST)/8.0;
00050700              THETA = ATAND(E,N);
00050800              OUT_RECORD.DIR = THETA;
00050900              RETURN;
00051000          END;
00051100
00051200      /*
00051300      *****
00051400      TEST TO SEE IF THREE CUT OF THE FOUR ANGLES ARE IN AGREEMENT
00051500      *****
00051600      */
00051700
00051800          MAX_LENGTH = 0;
00051900          L3 : DO I = 1 TO 4 ;
00052000              LENGTH (I) = 0;
00052100              SN = 0;
00052200              SE = 0;
00052300              L4 : DO J = I TO I + 2;
00052400                  SN = SN + NORTH(J);
00052500                  SE = SE + EAST(J) ;
00052600              END L4;
00052700              SN = SN / 3.0 ;
00052800              SE = SE / 3.0 ;
00052900              LENGTH (I) = SQRT(SN*SN + SE*SE );
00053000              IF LENGTH (I) > MAX_LENGTH THEN DO;
00053100                  MAX_LENGTH = LENGTH(I);
00053200                  INDEX_MAX = I;
00053300              END;
00053400          END L3;
00053500
00053600          IF MAX_LENGTH < 0.7071 THEN DO ;
00053700              PUT FILE(SYSPRINT) SKIP LIST ('*** DIRECTIONS DO NOT '||
00053800              ' AGREE ***');
00053900              PUT FILE(SYSPRINT) LIST(IN_RECORD);
00054000              PUT FILE(SYSPRINT) DATA (DEGREE_ANGLE,NORTH,EAST);
00054100              RETURN;
00054200          END;
00054300
00054400      /*
00054500      *****
00054600      IF THREE OUT OF THE FOUR ANGLES ARE ACCEPTABLE, THEN CALCULATE THE BEST
00054700      DIRECTION AND ESCAPE.
00054800      *****
00054900      */
00055000
00055100          N = 0.0 ;
00055200          E = 0.0;
00055300          L5: DO I = INDEX_MAX TO INDEX_MAX+ 2 ;
00055400              N = N + NORTH(I);
00055500              E = E + EAST(I);
00055600          END L5;
00055700
00055800          N = N / 3.0;
00055900          E = E / 3.0;
00056000          THETA = ATAND(E,N);
00056100          OUT_RECORD.DIR = THETA;
00056200          RETURN;

```

```

00056300      END CHECK_ANGLES;
000564001
00056500      /*
00056600      *****
00056700      MAIN BODY CF CODE FOR INT_PROCESS
00056800      *****
00056900      */
00057000      CURRENT_CLOCK = IN_RECORD.CLOCK ;
00057100      IF PREVIOUS_CLOCK >= CURRENT_CLOCK THEN DO;
00057200          PUT FILE(SYSPRINT) SKIP LIST
00057300              ('*** CLOCK ERROR - SEQUENCE ***',IN_RECORD);
00057400          RETURN;
00057500          END;
00057600
00057700      PREVIOUS_CLOCK = CURRENT_CLOCK;
00057800
00057900      IF IN_RECORD.SPEED > SPEED_MAX THEN DO;
00058000          PUT FILE(SYSPRINT) SKIP LIST('***SPEED ERROR - SPEED EXCEEDS'||
00058100              ' 3 SIGMA OF THE AVERAGE SPEED',IN_RECORD);
00058200          RETURN;
00058300          END;
00058400
00058500      DO M = 1 TO 4 ;
00058600          DEGREE_ANGLE(M) = ANGLE ( IN_RECORD.DIR(M) ) ;
00058700      END;
00058800
00058900      CUT_RECORD.DIR = -9999;
00059000      CALL CHECK_ANGLES ;
00059100      IF CUT_RECORD.DIR < -400 THEN RETURN;
00059200
00059300      CUT_RECORD.V = IN_RECORD.SPEED * SIND ( THETA) ;
00059400      CUT_RECORD.U = IN_RECORD.SPEED * COSD ( THETA) ;
00059500      OUT_RECORD.CLOCK = IN_RECORD.CLOCK;
00059600      CUT_RECORD.TRACK = IN_RECORD.TRACK;
00059700      OUT_RECORD.TEMP = 0.5*( SUM (IN_RECORD.TEMPS));
00059800      CUT_RECORD.FLAGS = 0.0;
00059900
00060000
00060100      WRITE FILE (UTFILE2) FROM (OUT_RECORD) ;
00060200      END INT_PROCESS;
000603001
00060400      ON ENDFILE (UTFILE1) ANOTHER_RECORD = '0'B;
00060500
00060600      OPEN FILE (UTFILE1) INPUT RECORD SEQUENTIAL ;
00060700      OPEN FILE (UTFILE2) OUTPUT RECORD SEQUENTIAL;
00060800      /*
00060900      *****
00061000      MAIN BODY CF INTERMEDIATE TESTING
00061100      *****
00061200      */
00061300      LOOP7 : DO WHILE (ANOTHER_RECORD) ;
00061400          READ FILE (UTFILE1) INTO (IN_RECORD);
00061500          IF ANOTHER_RECORD THEN CALL INT_PROCESS;
00061600      END LOOP7;
00061700
00061800      CLOSE FILE (UTFILE1);
00061900      CLOSE FILE (UTFILE2);
00062000
00062100      END INTERMEDIATE_TESTING;
000622001
00062300      INTERPOLATIONS: PROC;
00062400
00062500      /*

```

```

00062600 *****
00062700 DECLARATIONS LOCAL TO INTERPOLATIONS
00062800 *****
00062900 */
00063000
00063100 DCL OUTFILE FILE OUTPUT RECORD SEQUENTIAL,
00063200 1 IN_RECORD,
00063300     2 TRACK FIXED BIN (31),
00063400     2 DIR   FIXED BIN (31),
00063500     2 TEMP  FLOAT DEC (6),
00063600     2 U     FIXED BIN (31),
00063700     2 V     FIXED BIN (31),
00063800     2 CLOCK FIXED BIN (31),
00063900     2 FLAGS FIXED BIN (31),
00064000
00064100 1 OUT_RECORD,
00064200     2 YEAR  PIC '99' INIT ('75'),
00064300     2 DAY   PIC '999' INIT ('000'),
00064400     2 HR    PIC 'ZZ9' INIT ('000'),
00064500     2 MIN   PIC '99' INIT ('00'),
00064600     2 XPOS  PIC '----9' INIT ('0'),
00064700     2 YPOS  PIC '----9' INIT ('0'),
00064800     2 NORTH PIC '-----9' INIT ('0'),
00064900     2 EAST  PIC '-----9' INIT ('0'),
00065000     2 TEMP  PIC '--9V.9' INIT ('0.0'),
00065100     2 CODE  CHAR (3) INIT ('ZZZ'),
00065200     2 BLANKS CHAR (12) INIT ( (12) ' '),
00065300     2 TRACK PIC 'ZZZ9' INIT ('0'),
00065400     2 BLKS  CHAR (24) INIT ( (24) ' '),
00065500
00065600 ANOTHER_RECORD BIT (1) INIT ('1'B) ,
00065700
00065800 1 CURRENT_RECORD,
00065900     2 U FLOAT DEC (6),
00066000     2 V FLOAT DEC (6),
00066100     2 T FLOAT DEC (6),
00066200     2 CLOCK FIXED BIN (31),
00066300
00066400 1 NEXT_RECORD LIKE CURRENT_RECORD;
00066500
00066600 /*
00066700 *****
00066800 DECLARATIONS OF PROCEDURES INTERNAL TO INTERPOLATIONS
00066900 *****
00067000 */
00067100
00067200 INITIALIZE : PROC ;
00067300
00067400 /*
00067500 *****
00067600 PURPOSE : TO PROVIDE INITIALIZATION FOR FILES AND VARIABLES IN
00067700 INTERPOLATION. THIS ALSO SETS THE VALUES OF THE PARTS OF OUT_RECORD
00067800 WHICH DO NOT CHANGE.
00067900 *****
00068000 */
00068100
00068200 OPEN FILE (UTFILE2) INPUT RECORD SEQUENTIAL;
00068300 OPEN FILE (OUTFILE) OUTPUT RECORD SEQUENTIAL;
00068400
00068500 READ FILE (UTFILE2) INTO (IN_RECORD);
00068600
00068700 CURRENT_RECORD.CLOCK = IN_RECORD.CLOCK;

```

```

00068800      CURRENT_RECORD.U      = IN_RECORD.U;
00068900      CURRENT_RECORD.V      = IN_RECORD.V;
00069000      CURRENT_RECORD.T      = IN_RECORD.TEMP;
00069100
00069200      OUT_RECORD.YEAR = YEAR;
00069300      CALL TIME;
00069400      OUT_RECORD.XPOS = X_POSITION;
00069500      OUT_RECORD.YPCS = Y_POSITION;
00069600      OUT_RECORD.NORTH = CURRENT_RECORD.U;
00069700      OUT_RECORD.EAST  = CURRENT_RECORD.V;
00069800      OUT_RECORD.TEMP  = CURRENT_RECORD.T;
00069900      OUT_RECORD.CODE = INSTRUMENT_CODE;
00070000      OUT_RECORD.TRACK = IN_RECORD.TRACK;
00070100
00070200
00070300      WRITE FILE (OUTFILE) FROM (OUT_RECORD);
00070400      IF SUP_FINAL_PRINTING ^= 'YES' THEN
00070500      PUT FILE (SYSPRINT) SKIP EDIT (OUT_RECORD)
00070600      (P'99',P'999',P'ZZ9',P'99',2 P'-----9',2 P'-----9',
00070700      P'--9V.9',A(3),A(12),P'ZZ9',A(24) );
00070800
00070900      END INITIALIZE;
00071000
00071100      TIME      : PROC;
00071200
00071300      /*
00071400      *****
00071500      PURPOSE : TO COMPUTE THE DAY OF THE YEAR, THE HOUR AND MINUTE OF EACH
00071600      RECORD.
00071700      THE RESULTS ARE PLACED IN OUT_RECORD. THE DATA FOR THE COMPUTATION ARE
00071800      TAKEN FROM CURRENT_RECORD.
00071900      *****
00072000      */
00072100
00072200      DCL (T,T0,DT,XX,YY,ZZ) FLOAT DEC (14),
00072300      MONTHS (12) FIXED BIN (31) INIT (31,28,31,30,31,30,31,31,30,31,30,31),
00072400      (I,J,K) FIXED BIN (31),
00072500      FLAG BIT (1) INIT ('1'B);
00072600
00072700      IF FLAG THEN L0:DO;
00072800          FLAG = '0'B;
00072900          IF LEAP_YEAR = 'YES' THEN MONTHS(2) = 29;
00073000          T0 = 0;
00073100          L1 : DO I = 1 TO MONTH - 1;
00073200              T0 = T0 + MONTHS(I) ;
00073300          END L1;
00073400          YY=HOUR;
00073500          ZZ=MINUTE;
00073600          T0 = T0 + (YY)/24.0000 + (ZZ)/1440.000+ STARTING_DAY;
00073700          END L0;
00073800
00073900          XX=TIME_INTERVAL;
00074000          DT = ( CURRENT_RECORD.CLOCK*XX)/1440.000000 ;
00074100          T = T0 + DT;
00074200          OUT_RECORD.DAY = T;
00074300          OUT_RECORD.HR  = (T - OUT_RECORD.DAY)*24.00000;
00074400          OUT_RECORD.MIN  = ((T - OUT_RECORD.DAY)*24.000 - OUT_RECORD.HR)*60.00;
00074500
00074600      END TIME ;
00074700
00074800      INTERP_AND_WRITE_SDF : PROC;
00074900
00075000      DCL

```



```

00075100      (NUMBER_OF_LOOPS ,I,J ) FIXED BIN (31) INIT (1) ,
00075200      (DU,DV,DT) FLOAT DEC (6) INIT (0);
00075300
00075400  /*
00075500  ****
00075600      DECLARATION OF PROCEDURE LOCAL TO INTERP_AND_WRITE_SDF
00075700  ****
00075800  */
00075900
00076000      WRITE_SDF_OUT: PROC;
00076100          OUT_RECORD.NORTH = CURRENT_RECORD.U;
00076200          OUT_RECORD.EAST  = CURRENT_RECORD.V;
00076300          OUT_RECORD.TEMP. = CURRENT_RECORD.T;
00076400          IF I = NUMBER_OF_LOOPS THEN
00076500              OUT_RECORD.TRACK=IN_RECORD.TRACK;
00076600              CALL TIME;
00076700              WRITE FILE (OUTFILE) FROM (OUT_RECORD);
00076800          IF SUP_FINAL_PRINTING -='YES' THEN
00076900              PUT FILE(SYSPRINT) SKIP EDIT (OUT_RECORD)
00077000              (P'99',P'999',P'ZZ9',P'99',2 P'-----9',2 P'-----9',
00077100               P'--9V.9',A(3),A(12),P'ZZZ9',A(24) );
00077200          END WRITE_SDF_OUT;
00077300
00077400
00077500
00077600      NEXT_RECORD.CLOCK = IN_RECORD.CLOCK;
00077700      NEXT_RECORD.U      = IN_RECORD.U;
00077800      NEXT_RECORD.V      = IN_RECORD.V;
00077900      NEXT_RECORD.T      = IN_RECORD.TEMP;
00078000
00078100      IF NEXT_RECORD.CLOCK = CURRENT_RECORD.CLOCK + 1 THEN
00078200          L1 : DO ;
00078300              CURRENT_RECORD = NEXT_RECORD;
00078400              I=1;
00078500              NUMBER_OF_LOOPS=1;
00078600              CALL WRITE_SDF_OUT;
00078700          END L1;
00078800
00078900      ELSE
00079000          L2 : DO;
00079100              NUMBER_OF_LOOPS=NEXT_RECORD.CLOCK - CURRENT_RECORD.CLOCK;
00079200              DU = FLOAT(NEXT_RECORD.U-CURRENT_RECORD.U)/
00079300                  FLOAT(NUMBER_OF_LOOPS);
00079400
00079500              DV = FLOAT(NEXT_RECORD.V-CURRENT_RECORD.V)/
00079600                  FLOAT(NUMBER_OF_LOOPS);
00079700
00079800              DT = FLOAT(NEXT_RECORD.T-CURRENT_RECORD.T)/
00079900                  FLOAT(NUMBER_OF_LOOPS);
00080000
00080100              L3: DO I = 1 TO NUMBER_OF_LOOPS;
00080200                  CURRENT_RECORD.CLOCK=CURRENT_RECORD.CLOCK+1;
00080300                  CURRENT_RECORD.U=CURRENT_RECORD.U+DU;
00080400                  CURRENT_RECORD.V=CURRENT_RECORD.V+DV;
00080500                  CURRENT_RECORD.T=CURRENT_RECORD.T+DT;
00080600                  IF I < NUMBER_OF_LOOPS THEN
00080700                      OUT_RECORD.TRACK=OUT_RECORD.TRACK+1;
00080800                      CALL WRITE_SDF_OUT;
00080900                  END L3;
00081000              END L2;
00081100      END INTERP_AND_WRITE_SDF;
00081200

```

```

00081300  /*
00081400  *****
00081500  MAIN BODY OF CODE FOR INTERPOLATIONS
00081600  *****
00081700  */
00081800
00081900      ON ENDFILE (UTFILE2)  ANOTHER_RECORD = '0'B;
00082000
00082100      CALL INITIALIZE;
00082200
00082300      LOOP8 : DO WHILE (ANOTHER_RECORD);
00082400          READ FILE (UTFILE2) INTO (IN_RECORD);
00082500          IF ANOTHER_RECORD THEN CALL INTERP_AND_WRITE_SDF;
00082600      END LOOP8;
00082700
00082800      CLOSE FILE (UTFILE2);
00082900      CLOSE FILE (OUTFILE);
00083000
00083100  END INTERPOLATIONS;
00083200
00083300  /*
00083400  *****
00083500      MAIN BODY OF THE PROGRAM
00083600  *****
00083700  */
00083800
00083900
00084000      CALL READ_RAW_DATA;
00084100
00084200
00084300      CALL INTERMEDIATE_TESTING;
00084400
00084500      CALL INTERPOLATIONS;
00084600
00084700
00084800  END Q15PROC;

```

```

C*****00000010
C00000020
C00000030
C    PROGRAM      VECPUNCH      00000040
C00000050
C    MODIFIED BY K D SAUNDERS    00000060
C    THE ORIGINAL VERSION OF THIS PROGRAM HAS BEEN CHANGED TO    00000070
C    SEND THE CARD IMAGE INFORMATION FROM VECPUNCH TO A SEQUENTIAL 00000080
C    FILE, DESIGNATED BY LOGICAL UNIT NUMBER 21 IN THE EARLIER FORMAT 00000090
C    USED IN THE ARGONNE GREAT LAKES RESEARCH PROJECT.            00000100
C00000110
C    THE DATA IS WRITTEN IN THE STANDARD DATA FORMAT THAT IS TO BE USED 00000120
C    FROM AUGUST 1974 FOR RECORDING ALL LAKE DATA ON LOGICAL UNIT 22. 00000130
C00000140
C00000150
C    INPUT      00000160
C00000170
C    CARD ONE    00000180
C                FORMAT : I4,I2,2X,18A4 00000190
C                COLUMNS  CONTENTS      00000200
C                01-04      (I4) NUMBER OF POINTS TO BE PUNCHED 00000210
C                05-08      (I2) LOGICAL TAPE UNIT NUMBER FOR READING 00000220
C                09-80      (18A4) TITLE 00000230
C00000240
C    CARD TWO    00000250
C    FORMAT : 3((I4,I4,F5.2)) 00000260
C                COLUMNS  CONTENTS      00000270
C                01-04      (I4) IC (1)   00000280
C                05-08      (I4) NDIR(1)   00000290
C                09-13      (F5.2) SPD(1)   00000300
C                14-17      (I4) IC (2)   00000310
C                18-21      (I4) NDIR(2)   00000320
C                22-26      (F5.2) SPD(2)   00000330
C                27-30      (I4) IC (3)   00000340
C                31-34      (I4) NDIR(3)   00000350
C                35-39      (F5.2) SPD(3)   00000360
C00000370
C00000380
C    WHERE      00000390
C                IC      = THE TRACK NUMBER 00000400
C                NDIR    = THE ANGLE        00000410
C                SPD     = THE SPEED        00000420
C00000430
C00000440
C00000450
C    CARD THREE  FORMAT:I4,I3,I4,I2,A3 00000460
C                COLUMN    CONTENTS      00000470
C                01-04      (I4)          YEAR 00000480
C                05-07      (I3)          DAY OF THE YEAR 00000490
C                08-11      (I4)          TIME OF THE FIRST RECORD ON FILM 00000500
C                                (HOUR PORTION) 00000510
C                12-13      BLANK          00000520
C                14-16      (A3)          INSTRUMENT CODE 00000530
C00000540
C    WHERE      00000550
C    THE INSTRUMENT CODE MUST BE B96,B97,OR B98 , I.E. ONE OF THE 00000560
C    BRAINCON CURRENT METEERS. NOTE THAT B96 IS NOT INVERTED, WHILE 00000570
C    BOTH B97 AND B98 ARE INVERTED. 00000580
C00000590
C*****00000600
C00000610
C    DIMENSION  TITLE(18),IC(4000),NDIR(4000),SPD(4000),SPEDCO(4000), 00000620
C    1ITEST(4000),IC1(3),NDIR1(3),SPD1(3),SPEDCA(4000) 00000630

```

```

SUBROUTINE DRIVER(I PLOT,NPTS,U,V,TIME,TEMPS,KEDEN,UHIST,VHIST,X,Y 00000100
1,AVTEMP,MAXKE,XX,XXN,YY,YYN,TITLE1,SPEED,ANGLE) 00000200
REAL*4 U(1),V(1),TIME(1),KEDEN(1),UHIST(1),VHIST(1),X(1),Y(1) 00000300
REAL*4 TEMPS(1),IUX(201) 00000400
REAL*4 MAXKE,XX,XXN,YY,YYN,SPEED(1),ANGLE(1) 00000500
INTEGER I PLOT,NPTS,TITLE1(1) 00000600
C 00000700
C ***** 00000800
C 00000900
C DRIVER FOR THE PLOTING SECTION OF THE PLI DATA REDUCTION 00001000
C PROGRAM. 00001100
C ***** 00001200
C 00001300
C IF(I PLOT .EQ. 0) CALL STRTPL 00001400
C 00001500
C ***** 00001600
C 00001700
C 00001800
C PLOT THE FIRST GRAPH - VELOCITY COMPONENTS,KINETIC ENERGIES, 00001900
C AND TEMPERATURES IF THERE ARE ANY. IF NOT, ADJUST THE PLOT 00002000
C ACCORDINGLY. 00002100
C ***** 00002200
C 00002300
C 00002400
C NUM=I PLOT + 1 00002500
C CALL BGNPL(NUM) 00002600
C CALL SCMPX 00002700
C IF(AVTEMP .LT. 0.5) CALL PHYSOR(1.0,2.0) 00002800
C IF(AVTEMP .GE. 0.5) CALL PHYSOR(1.0,1.0) 00002900
C CALL HEIGHT(0.09) 00003000
C CALL TITLE(' ',1,'TIME (DAYS) $',100,'U (MM/SEC) $', 00003100
1100,6.5,1.5) 00003200
C IKTM = TIME(1) 00003300
C TKTM = IKTM 00003400
C ITP = TIME(NPTS) 00003500
C TP = ITP+1 00003600
C CALL GRAF(TKTM,1.0,TP,-500.0,500.0,500.0) 00003700
C CALL GRID(1,1) 00003800
C CALL FRAME 00003900
C CALL CURVE(TIME,U,NPTS,0) 00004000
C CALL ENDGR(1) 00004100
C CALL OREL(0.0,2.0) 00004200
C CALL HEIGHT(0.09) 00004300
C CALL TITLE(' ',1,' ',1,'V (MM/SEC) $',100,6.5,1.5) 00004400
C CALL REGR 00004500
C CALL GRID(1,1) 00004600
C CALL FRAME 00004700
C CALL CURVE(TIME,V,NPTS,0) 00004800
C CALL ENDGR(2) 00004900
C CALL OREL(0.0,2.0) 00005000
C CALL HEIGHT(0.09) 00005100
C CALL TITLE(' ',1,' ',1,'SPECIFIC K.E. $',100,6.5,1.5) 00005200
C CALL GRAF(TKTM,1.0,TP,0.0,'SCALE',MAXKE) 00005300
C CALL GRID(1,1) 00005400
C CALL FRAME 00005500
C CALL CURVE(TIME,KEDEN,NPTS,0) 00005600
C CALL ENDGR(3) 00005700
C IF(AVTEMP .LT. 0.5) GO TO 1000 00005800
C CALL OREL(0.0,2.0) 00005900
C CALL HEIGHT(0.09) 00006000
C CALL TITLE(' ',1,' ',1,'TEMP (DEG C) $',100,6.5,1.5) 00006100
C CALL FRAME 00006200

```



```

CALL GRAF(TKTM,1.0,TP,0.0,5.0,25.0) 00006300
CALL GRID(1,1) 00006400
CALL CURVE(TIME,TEMPS,NPTS,0) 00006500
CALL ENDGR(4) 00006600
1000 CONTINUE 00006700
CALL OREL(0.0,1.0) 00006800
CALL HEIGHT(0.09) 00006900
IF( AVTEMP .GE. 0.5) 00007000
1CALL TITLE('VELOCITIES,K.E.,TEMPERATURES $',100,' ',1,' ',1,6.5, 00007100
11.5) 00007200
IF (AVTEMP .LT. 0.5 ) CALL TITLE('VELOCITIES AND KINETIC ENERGIES 00007300
1 $',100,' ',1,' ',1,6.5,1.5) 00007400
CALL ENDGR(15) 00007500
CALL OREL(0.0,0.5) 00007600
CALL HEIGHT(0.09) 00007700
CALL TITLE(TITLE1,100,' ',1,' ',1,6.5,1.5) 00007800
CALL ENDPL(NUM) 00007900
C 00008000
C ***** 00008100
C 00008200
C PLOT THE HISTOGRAMS OF THE CURRENTS 00008300
C 00008400
C ***** 00008500
C 00008600
UHMx = 0.0 00008700
VHMx = 0.0 00008800
DO 4000 I=1,201 00008900
IF(UHMx .LT. UHIST(I)) UHMx=UHIST(I) 00009000
IF(VHMx .LT. VHIST(I)) VHMx=VHIST(I) 00009100
4000 IUX(I) = I - 101 00009200
CALL BGNPL(2) 00009300
CALL SCMPX 00009400
CALL CROSS 00009500
CALL HEIGHT(0.09) 00009600
CALL PHYSOR(2.75,1.0) 00009700
CALL TITLE(' ',1,'U - VELOCITY (CM/SEC) $',100,'PROBABILITY$',100, 00009800
13.0,1.5) 00009900
CALL GRAF(-100.0,50.0,100.0,0.0,'SCALE',UHMx) 00010000
CALL FRAME 00010100
CALL XTICKS(5) 00010200
CALL CURVE(IUX,UHIST,201,0) 00010300
CALL ENDGR(1) 00010400
CALL OREL(0.0,2.5) 00010500
CALL CROSS 00010600
CALL HEIGHT(0.09) 00010700
CALL TITLE(' ',1,'V - VELOCITY (CM/SEC) $',100,'PROBABILITY$',100, 00010800
13.0,1.5) 00010900
CALL GRAF(-100.0,50.0,100.0,0.0,'SCALE',VHMx) 00011000
CALL FRAME 00011100
CALL XTICKS(5) 00011200
CALL CURVE(IUX,VHIST,201,0) 00011300
CALL ENDGR(2) 00011400
CALL CROSS 00011500
CALL HEIGHT(0.09) 00011600
CALL OREL(0.0,2.5) 00011700
CALL TITLE(' ',1,' ',1, 00011800
1' ',1,3.0,3.0) 00011900
CALL GRAF(-500.0,500.0,500.0,-500.0,500.0,500.0) 00012000
CALL XTICKS(5) 00012100
CALL YTICKS(5) 00012200
CALL FRAME 00012300
CALL MARKER(3) 00012400
CALL CURVE(V,U,NPTS,-1) 00012500

```

```

CALL ENDGR(3)                                00012600
CALL OREL(0.0,0.5)                          00012700
CALL HEIGHT(0.09)                          00012800
CALL TITLE('VELOCITY HISTOGRAMS $',100,' ',1,' ',1,3.0,3.0) 00012900
CALL ENDGR(4)                                00013000
CALL OREL(0.0,0.2)                          00013100
CALL HEIGHT(0.09)                          00013200
CALL TITLE(TITLE1,100,' ',1,' ',1,3.0,3.0) 00013300
CALL ENDPL(2)                               00013400
C                                           00013500
C *****                                00013600
C                                           00013700
C PLOT OF ANGLES AND SPEEDS OF THE CURRENTS. 00013800
C THE ANGLES ARE IN DEGREES FROM NORTH, POSITIVE TAKEN CLOCKWISE. 00013900
C THE SPEEDS ARE IN MM/SEC.                00014000
C                                           00014100
C *****                                00014200
C                                           00014300
CALL BGNPL(4)                               00014400
CALL SCMPX                                  00014500
CALL PHYSOR(1.0,3.0)                       00014600
CALL HEIGHT(0.09)                          00014700
CALL TITLE(' ',1,'TIME (DAYS) $',100,'SPEED (MM/SEC) $',100,6.5,1.5) 00014800
CALL FRAME                                00014900
CALL GRAF(TKTM,1.0,TP,0.0,250.0,500.0)    00015000
CALL GRID(1,1)                             00015100
CALL YTICKS(5)                             00015200
CALL CURVE(TIME,SPEED,NPTS,0)              00015300
CALL ENDGR(41)                             00015400
CALL OREL(0.0,2.5)                         00015500
CALL HEIGHT(0.09)                         00015600
CALL TITLE(' ',1,' ',1,'ANGLE (DEG FROM N) $',100,6.5,1.5) 00015700
CALL FRAME                                00015800
CALL GRAF(TKTM,1.0,TP,-180.0,90.0,180.0) 00015900
CALL GRID(1,1)                             00016000
CALL CURVE(TIME,ANGLE,NPTS,0)             00016100
CALL ENDGR(42)                             00016200
CALL OREL(0.0,0.5)                         00016300
CALL HEIGHT(0.09)                         00016400
CALL TITLE('SPEEDS AND DIRECTIONS$',100,' ',1,' ',1,6.5,1.5) 00016500
CALL ENDGR(45)                             00016600
CALL OREL(0.0,0.5)                         00016700
CALL HEIGHT(0.09)                         00016800
CALL TITLE(TITLE1,100,' ',1,' ',1,6.5,1.5) 00016900
CALL ENDPL(4)                               00017000
C                                           00017100
C *****                                00017200
C                                           00017300
C PLOT THE LAST GRAPH - THE PLOT OF THE PROGRESSIVE VECTOR 00017400
C DIAGRAM.                                00017500
C                                           00017600
C *****                                00017700
C                                           00017800
CALL BGNPL(3)                               00017900
CALL SCMPX                                  00018000
CALL HEIGHT(0.09)                          00018100
CALL TITLE('PROGRESSIVE VECTOR DIAGRAM $',100,'DISTANCE (M) $',100 00018200
1,' ',1,7.0,7.0)                          00018300
DMAX = XMX                                  00018400
IF(DMAX.LT.YMX) DMAX=YMX                   00018500
DMIN=XMN                                    00018600
IF(DMIN.GT.YMN) DMIN=YMN                   00018700

```

	WIDTH = DMAX -DMIN	00018800
	STEPSZ = 500.0	00018900
2000	NSTEPS = WIDTH/STEPSZ	00019000
	IF (NSTEPS .LT.15) GO TO 3000	00019100
	STEPSZ = STEPSZ*2.	00019200
	GO TO 2000	00019300
3000	CALL GRAF(DMIN,STEPSZ,DMAX,DMIN,STEPSZ,DMAX)	00019400
	CALL GRID(1,1)	00019500
	CALL FRAME	00019600
	CALL CURVE(X,Y,NPTS,180)	00019700
	CALL RLREAL (TIME(1),106,0.0,0.0)	00019800
	CALL ENDGR(55)	00019900
	CALL OREL(0.0,0.5)	00020000
	CALL HEIGHT(0.09)	00020100
	CALL TITLE(TITLE1,100,' ',1,' ',1,7.0,7.0)	00020200
	CALL ENDPL(3)	00020300
	RETURN	00020400
	END	00020500

```

00000020 MURPHY: PROCEDURE OPTIONS (MAIN);
00000030 /*
00000040
00000050 *****
00000060
00000070 PROGRAM : CURRENT DISPLAY
00000080
00000090 PURPOSE : TO GRAPHICALLY DISPLAY THE RESULTS FROM CURRENT MEASUREMENTS
00000100 OF THE ARGONNE NATIONAL LABORATORY'S GREAT LAKES RESEARCH
00000110 PROJECT.
00000120
00000130 AUTHOR : DR. K. D. SAUNDERS
00000140 ENVIRONMENTAL AND ENERGY SYSTEMS DIVISION
00000150 ARGONNE NATIONAL LABORATORY
00000160 9700 S CASS AVE.
00000170 ARGONNE, ILLNOIS 60439
00000180
00000190 DATE : AUGUST 6, 1975
00000200
00000210
00000220 LANGUAGE : PL/I
00000230
00000240 USAGE : THE INPUT IS FREE STREAM. THE ORDER OF ITEM IS IMPORTANT.
00000250 NO. ITEM
00000260 1 THE STARTING DAY OF THE RECORD.
00000270 2 MONTH
00000280 3 HOUR
00000290 4 MINUTE
00000300 5 THE NUMBER OF DAYS PER PLOT DESIRED.
00000310 6 THE SAMPLING INTERVAL ( MINUTES).
00000320 7 A TITLE WHICH WILL BE PLOTTED ON ALL PLOTS
00000330
00000340 THE INPUT DATA IS LOCATED IN A FILE NAMED 'INFILE'. THIS MUST
00000350 HAVE A DD CARD WITH THE DDNAME=GO.INFILE.
00000360 *****
00000370
00000380
00000390 */
00000400 ON ERROR BEGIN;
00000410 PUT DATA;
00000420 CALL PLIDUMP('TFCHAB');
00000430 END;
00000440 DECLARE
00000450 CONVCTR FIXED BIN (15) INITIAL (0),
00000460 (DAY0,MONTH0,HOUR0,MINUTE0,NUMBEROFDAYS PERPLOT) FIXED BIN (15),
00000470 I999 FIXED BIN (31) INITIAL (999),
00000480 IO FIXED BIN (31) INITIAL ( 0 ),
00000490 MONTH (12) CHAR (9) INITIAL
00000500 ('JANUARY','FEBRUARY','MARCH','APRIL','MAY','JUNE','JULY','AUGUST',
00000510 'SEPTEMBER','OCTOBER','NOVEMBER','DECEMBER'),
00000520 JULIANDAY(12) FIXED BIN (15) INITIAL
00000530 (0,31,59,90,120,151,181,212,243,273,304,334),
00000540 (SIZE,INTERVAL) FIXED BIN (15),
00000550 INFILE FILE RECORD SEQUENTIAL,
00000560 FLAG FIXED BIN (15),
00000570 (STARTDAY,STOPDAY) FLOAT DEC (6),
00000580 (PLOT,DRIVER) EXTERNAL OPTIONS (FORTRAN INTER),
00000590 TITLE1 CHAR (40) ALIGNED ;
00000600 GET LIST (DAY0,MONTH0,HOUR0,MINUTE0,NUMBEROFDAYS PERPLOT,INTERVAL,TITLE1);
00000610 PUT DATA (DAY0,MONTH0,HOUR0,MINUTE0,NUMBEROFDAYS PERPLOT,INTERVAL,TITLE1);
00000620 ARRAYLOAD: PROCEDURE (A,S);
00000630 DECLARE (I,L,A(*)) FIXED BIN (31),
00000640 S CHAR (*) ;

```



```

00000650      L = LENGTH(S);
00000660      INBLOCK: BEGIN;
00000670          DECLARE XS CHAR (L+4);
00000680          XS = S||'  ';
00000690          LOADLOOP: DO I = 1 TO CEIL((L-.0001)/4.0);
00000700              UNSPEC(A(I)) = UNSPEC(SUBSTR(XS,4*(I-1)+1,4));
00000710          END LOADLOOP;
00000720      END INBLOCK;
00000730      END ARRAYLOAD;
00000740      JDAY: PROCEDURE RETURNS (FIXED BIN (15));
00000750          RETURN (JULIANDAY(MONTHO) + DAYO);
00000760      END JDAY;
00000770      CORL: PROCEDURE (U,V,C,N);
00000780          DECLARE
00000790              (U(*),V(*),C(*),DU(N),DV(N)) FLOAT DEC (6),
00000800              (I,J,N) FIXED BIN (31);
00000810              DU=U*U;
00000820              DV=V*V;
00000830          LOOP1: DO I = 0 TO N/10 - 1;
00000840              C(I+1) = 0.0;
00000850              LOOP2: DO J = 1 TO N-I;
00000860                  C(I+1) = C(I+1) + U(J)*V(J+I);
00000870              END LOOP2;
00000880              C(I+1) = C(I+1)/SQRT(SUM(DU)*SUM(DV));
00000890          END LOOP1;
00000900      END CORL;
00000910      OUTBLOCK: BEGIN;
00000920          FLAG = 0;
00000930          SIZE = (1440/INTERVAL)*NUMBEROFDAYS PERPLOT;
00000940          BLOCK2: BEGIN;
00000950              DECLARE
00000960                  (U(SIZE),V(SIZE),TIME(SIZE),TEMPS(SIZE),KEDEN(SIZE),AVTEMP,MAXKE,SPEED(SIZE)
00000970                   ,ANGLE(SIZE)) FLOAT DEC (6),
00000980                  (CORUU(SIZE/10),CORVV(SIZE/10),CORUV(SIZE/10),CORVU(SIZE/10),DUM(SIZE))
00000990                  FLOAT DEC (6),
00001000                  (X(SIZE+1),Y(SIZE+1),XMAX,YMAX,XMIN,YMIN) FLOAT DEC (6),
00001010                  (UHIST(-100:100),VHIST(-100:100),DT) FLOAT DEC (6),
00001020                  (I,J) FIXED BIN (15),
00001030              1 DATALIST,
00001040                  2 YEAR      CHAR (2),
00001050                  2 FDAY      CHAR (3),
00001060                  2 HR        CHAR (3),
00001070                  2 MIN       CHAR (2),
00001080                  2 XPOS      CHAR (5),
00001090                  2 YPOS      CHAR (5),
00001100                  2 PU        CHAR (6),
00001110                  2 PV        CHAR (6),
00001120                  2 PT        CHAR (5),
00001130                  2 DPTH      CHAR (3),
00001140                  2 CODE      CHAR (3),
00001150                  2 STUFF     CHAR (37),
00001160                  (NPOINTS,IPLPOTFLAG) FIXED BIN (31),
00001170                  (XHR,PMIN) FLOAT DEC (6),
00001180                  FTIME FLOAT DEC (6) INITIAL (0.0);
00001190                  STARTDAY = JDAY;
00001200                  STOPDAY = NUMBEROFDAYS PERPLOT + STARTDAY;
00001210                  FDAY = '000';
00001220                  FTIME = STARTDAY + .001;
00001230          OUTLOOP: DO IPLPOTFLAG = 0 TO 5 WHILE (FLAG = 1);
00001240              NPOINTS = 0;
00001250              KEDEN = 0.0;
00001260              U = 0.001;

```

```

00001270      V = 0.001;
00001280      TEMPS = 0.0;
00001290      DUM = 0.0;
00001300      TIME = 0.0;
00001310  /*
00001320  ****
00001330  THE BLOCK - GRABDATA - IS USED TO GET A BLOCK OF DATA FROM INFILE,
00001340  PERFORM THE NECESSARY CONVERSIONS TO INTERNAL REPRESENTATION, AND
00001350  UPDATE THE TIME POINTERS.
00001360  ****
00001370  */
00001380  GRABDATA: BEGIN;
00001390      INLOOP: DO ILOOP = 1 TO SIZE - 1 WHILE ( FTIME <= STOPDAY );
00001400          ON ENDFILE (INFILE) BEGIN;
00001410              FLAG = 1;
00001420              GO TO OUT;
00001430          END;
00001440          ON CONVERSION BEGIN;
00001450              PUT DATA (DATALIST, CONVCTR, NPOINTS);
00001460              CONVCTR = CONVCTR + 1;
00001470              IF CONVCTR > 10 THEN STOP;
00001480              REVERT CONVERSION;
00001490          END;
00001500          READ FILE (INFILE) INTO (DATALIST);
00001510          NPOINTS = NPOINTS + 1;
00001520          IF NPOINTS > SIZE THEN GO TO OUT;
00001530          TIME(NPOINTS) = FDAY;
00001540          IF HR = ' ' THEN XHR = 0;
00001550          ELSE XHR = HR;
00001560          IF MIN = ' ' THEN PMIN = 0;
00001570          ELSE PMIN = MIN;
00001580          TIME(NPOINTS) = TIME(NPOINTS) + XHR/24.0 + PMIN/1440.;
00001590          FTIME = TIME(NPOINTS);
00001600          U(NPOINTS) = FU;
00001610          V(NPOINTS) = FV;
00001620          IF U(NPOINTS) = 0.0 THEN U(NPOINTS) = 0.01;
00001630          IF V(NPOINTS) = 0.0 THEN V(NPOINTS) = 0.01;
00001640          IF SUBSTR(FT, 5, 1) = '*' | SUBSTR(FT, 5, 1) = ' ' THEN FT = '00000';
00001650          TEMPS(NPOINTS) = FT;
00001660      END INLOOP;
00001670      OUT: STARTDAY = TIME(NPOINTS) + 1.0E00/1440.000000;
00001680      STOPDAY = STARTDAY + NUMBEROFDAYS PERPLOT;
00001690  END GRABDATA;
00001700  PUT DATA (NPOINTS, STARTDAY, STOPDAY, FTIME);
00001710  KEDEN = 0.5*(U*U + V*V);
00001720  SPEED = SQRT(2.0*KEDEN);
00001730  ANGLE = ATAND(V, U);
00001740  MAXKE = 0.0;
00001750  DO LL = 1 TO NPOINTS;
00001760      IF KEDEN(LL) > MAXKE THEN MAXKE = KEDEN(LL);
00001770  END;
00001780  /*
00001790  ****
00001800  THE BLOCK - STATISTICS - IS USED FOR COMPUTING THE CURRENT STATISTICS
00001810  FOR THE BLOCK OF DATA OBTAINED BY - GRABDATA. AVERAGES, STANDARD
00001820  DEVIATIONS, SKEWNESS AND KURTOSSES ARE COMPUTED AS WELL AS REYNOLDS
00001830  STRESSES FOR THE PERIOD.
00001840  ****
00001850
00001860
00001870
00001880
00001890

```

```

00001900
00001910 *****
00001920
00001930 */
00001940 STATISTICS: BEGIN;
00001950 DECLARE
00001960 (SIGU,SIGV,KURTU,KURTV,SKEWU,SKEWV,AVU,AVV,REYN)
00001970 FLOAT DEC (6);
00001980 AVU = SUM(U)/NPOINTS;
00001990 AVV = SUM(V)/NPOINTS;
00002000 PUT DATA (AVU,AVV);
00002010 DUM = U*U;
00002020 SIGU=SUM(DUM)/NPOINTS - AVU*AVU;
00002030 DUM = V*V;
00002040 SIGV=SUM(DUM)/NPOINTS - AVV*AVV;
00002050 SIGU=SQRT(SIGU);
00002060 SIGV=SQRT(SIGV);
00002070 PUT DATA (SIGU,SIGV);
00002080 DUM = (U-AVU)**3;
00002090 SKEWU=SUM(DUM) / (NPOINTS*SIGU**3) ;
00002100 DUM = (V-AVV)**3;
00002110 SKEWV=SUM(DUM) / (NPOINTS*SIGV**3);
00002120 DUM = (U-AVU)**4;
00002130 KURTU=SUM(DUM) / (NPOINTS*SIGU**4);
00002140 DUM = (V-AVV)**4;
00002150 KURTV=SUM(DUM) / (NPOINTS*SIGV**4);
00002160 PUT SKIP DATA (SKEWU,SKEWV,KURTU,KURTV);
00002170 REYN= SUM((U-AVU)*(V-AVV))/NPOINTS;
00002180 PUT PAGE EDIT ('STATISTICAL SUMMARY') (X(10),A(20));
00002190 PUT SKIP EDIT (TITLE1,TIME(1),TIME(NPOINTS))
00002200 (X(10),A(40),SKIP(2),X(10),2 F(10,5));
00002210 PUT SKIP EDIT ('NUMBER OF POINTS', NPOINTS) (X(10),A(16),X(5),F(5));
00002220 PUT SKIP EDIT ('AVERAGE-U',AVU,'AVERAGE-V',AVV)
00002230 (X(10),A(9),X(5),F(10,3),SKIP,X(10),A(9),X(5),F(10,3));
00002240 PUT SKIP EDIT ('SIGMA','SKEWNESS','KURTOSIS')
00002250 (COLUMN(20),A(5),COLUMN(40),A(8),COLUMN(60),A(8));
00002260 PUT SKIP(2) EDIT ('U',SIGU,SKEWU,KURTU,'V',SIGV,SKEWV,KURTV)
00002270 (2(X(10),A(1),COLUMN(20),F(10,5),COLUMN(40),F(10,5)
00002280 ,COLUMN(60),F(10,5),SKIP));
00002290 PUT SKIP EDIT ('AVERAGE REYNOLDS STRESS',REYN)
00002300 (X(10),A(24),X(5),F(20,5));
00002310 PUT PAGE;
00002320 END STATISTICS;
00002330 PUT SKIP LIST ('END STATISTICS');
00002340 /*
00002350
00002360 *****
00002370
00002380 THE BLOCK -CORRELLATIONS- IS USED FOR COMPUTING THE AUTO AND CROSS
00002390 CORRELLATIONS OF THE TWO VELOCITY COMPONENTS FOR THE PERIOD
00002400 OF DAYS DETERMINED FROM THE INPUT. THE NUMBER OF LAGS IS PROPORTIONAL
00002410 TO THE LENGTH OF THE RECORD.
00002420
00002430 *****
00002440
00002450 */
00002460
00002470 CORRELLATIONS: BEGIN;
00002480 CORUU = 0.0;
00002490 CORVV = 0.0;
00002500 CORUV = 0.0;
00002510 CORVU = 0.0;

```

```

00002520      CALL CORL(U,U,CORUU,NPOINTS);
00002530      CALL CORL(V,V,CORVV,NPOINTS);
00002540      CALL CORL(U,V,CORUV,NPOINTS);
00002550      CALL CORL(V,U,CORVU,NPOINTS);
00002560      PUT SKIP EDIT ('CORRELATIONS') (X(10),A(13));
00002570      PUT SKIP EDIT ('LAG','UU','VV','UV','VU')
00002580          (COLUMN(10),A(3),COLUMN(20),A(2),COLUMN(30),A(2),COLUMN(40),A(2)
00002590          ,COLUMN(50),A(2));
00002600      PRINTLOOP:DO I = 0 TO NPOINTS/10;
00002610          PUT SKIP EDIT (I,CORUU(I+1),CORVV(I+1),CORUV(I+1),CORVU(I+1))
00002620              (X(9),F(5),COLUMN(20),4 F(10,5));
00002630      END PRINTLOOP;
00002640      END CORRELATIONS;
00002650      PUT SKIP LIST ('END CORRELATIONS');
/*
00002660
00002670
00002680      *****
00002690
00002700      THE BLOCK -PERSISTENCE- CALCULATES THE AVERAGE PERIOD THE CURRENT TRAVELS
00002710      IN ONE DIRECTION FOR EACH COMPONENT OF THE CURRENT.
00002720
00002730      *****
00002740
00002750      */
00002760      PERSISTENCE:BEGIN;
00002770      DECLARE
00002780          (ULEN(NPOINTS),VLEN(NPOINTS),UK,VK) FIXED BIN (15),
00002790          XK FLOAT DEC (6),
00002800          (ULTOT,VLTOT,PERU,PERV,DELTAT) FLOAT DEC (6);
00002810      ULEN= 0;
00002820      VLEN=0;
00002830      UK = 1;
00002840      VK = 1;
00002850      LP1: DO I = 1 TO NPOINTS - 1;
00002860          IF SIGN(U(I)) = SIGN(U(I+1)) THEN ULEN(UK) = ULEN(UK) + 1;
00002870          ELSE
00002880              DO;
00002890                  UK = UK + 1;
00002900                  ULEN(UK) = 1;
00002910              END;
00002920          IF SIGN(V(I)) = SIGN(V(I+1)) THEN VLEN(VK) = VLEN(VK) + 1;
00002930          ELSE
00002940              DO;
00002950                  VK = VK + 1;
00002960                  VLEN(VK) = 1;
00002970              END;
00002980      END LP1;
00002990      XK = UK;
00003000      ULTOT = SUM(ULEN)/XK;
00003010      VK = VK;
00003020      VLTOT = SUM(VLEN)/VK;
00003030      DELTAT = TIME(NPOINTS) - TIME(1);
00003040      PERU = (ULTOT/NPOINTS)*DELTAT;
00003050      PERV = (VLTOT/NPOINTS)*DELTAT;
00003060      PUT SKIP EDIT ('U-PERSISTENCE',PERU) (X(10),A(13),X(3),F(10,5));
00003070      PUT SKIP EDIT ('V-PERSISTENCE',PERV) (X(10),A(13),X(3),F(10,5));
00003080      PUT PAGE;
00003090      END PERSISTENCE;
00003100      PUT SKIP LIST ('END PERSISTENCE');
/*
00003110
00003120
00003130      *****
00003140

```



```

00003150 THE BLOCK -TWODINHISTOGRAM- SETS UP THE ARRAYS FOR PLOTTING THE ONE
00003160 AND TWO DIMENSIONAL HISTOGRAMS OF THE CURRENT COMPONENTS.
00003170 *****
00003180
00003190
00003200 */
00003210 TWODINHISTOGRAM:BEGIN;
00003220 DECLARE (TESTU,TESTV) FLOAT DEC (6);
00003230 UHIST = 0;
00003240 VHIST= 0;
00003250 HISTLOOP:DO I = 1 TO NPOINTS;
00003260 J=U(I)/10.0 + 0.5;
00003270 UHIST(J) = UHIST(J) + 1;
00003280 J=V(I)/10.0 + 0.5;
00003290 VHIST(J) = VHIST(J) + 1;
00003300 END HISTLOOP;
00003310 UHIST=UHIST/NPOINTS;
00003320 VHIST=VHIST/NPOINTS;
00003330 TESTU=SUM(UHIST);
00003340 TESTV=SUM(VHIST);
00003350 PUT LIST ('TESTU',TESTU,'TESTV',TESTV);
00003360 END TWODINHISTOGRAM;
00003370 PROGVECTORPOS: BEGIN;
00003380 /* THIS BLOCK COMPUTES THE POSITIONS FOR USE IN PLOTTING
00003390 THE PROGRESSIVE VECTOR DIAGRAM. */
00003400 X = 0;
00003410 Y = 0;
00003420 XMAX = -100000.;
00003430 YMAX = -100000.;
00003440 XMIN = 100000.;
00003450 YMIN = 100000.;
00003460 PUT SKIP DATA (XMAX,YMAX,XMIN,YMIN);
00003470 /* X AND Y WILL BE EXPRESSED IN METERS */
00003480 PROGLOOP: DO I = 1 TO NPOINTS;
00003490 IF I = 1 THEN
00003500 DO;
00003510 X(I) = 0;
00003520 Y(I) = 0;
00003530 END;
00003540 ELSE
00003550 DO;
00003560 DT = 86.4*(TIME(I)-TIME(I-1));
00003570 X(I)=X(I-1)+V(I)*DT;
00003580 Y(I)=Y(I-1)+U(I)*DT;
00003590 END;
00003600 IF X(I)>XMAX THEN XMAX=X(I);
00003610 IF Y(I)>YMAX THEN YMAX=Y(I);
00003620 IF X(I)<XMIN THEN XMIN=X(I);
00003630 IF Y(I)<YMIN THEN YMIN=Y(I);
00003640 END PROGLOOP;
00003650 PUT SKIP DATA (XMAX,YMAX,XMIN,YMIN);
00003660 PUT SKIP LIST ('END OF PROGVECTOR LOOP');
00003670 END PROGVECTORPOS;
00003680 AVTEMP= SUM(TEMPS)/NPOINTS;
00003690 */
00003700 *****
00003710
00003720 THE FOLLOWING CALL TO DRIVER IS ONE OF THE TWO CALLS TO A FORTRAN
00003730 ROUTINE IN THIS PROGRAM.
00003740
00003750 DRIVER TAKES CARE OF THE PLOTTING OF ALL THE DATA AND IS WRITTEN IN
00003760

```

```

00003770      FORTRAN TO OVERCOME SOME PROBLEMS WITH GETMAINS THAT OCCURRED WHEN
00003780      THE DISPLAY ROUTINES WERE CALLED DIRECTLY FROM THE PL/I PROGRAM.
00003790      THIS RUSE SEEMS TO HAVE WORKED. BUT AS OF AUGUST 8, 1975 THE PROBLEM
00003800      STILL HAS NOT BEEN SOLVED.
00003810
00003820      *****
00003830
00003840      */
00003850          CALL DRIVER(I PLOTFLAG,NPOINTS,U,V,TIME,TEMPS,
00003860          KEDEN,UHIST(-100),VHIST(-100),X,Y,AVTEMP,MAXKE,XMAX,XMIN,YMAX,YMIN
00003870          ,TITLE1,SPEED,ANGLE);
00003880      END OUTLOOP;
00003890      END BLOCK2;
00003900  END OUTBLOCK;
00003910  CALL PLOT(I0,I0,I999);
00003920  END MURPHY;

```